Framework for Cyber-Physical Systems

Release 1.0

May 2016

Cyber Physical Systems Public Working Group

Revision Tracking

Version	Date	Editor	Changes
0.7	20150107	MJB	First integrated version
0.7	20150407	MJB	Release for CPS PWG F2F discussion
0.7	20150806	MJB	Draft prior to editors' resolution
0.7	20150817	KAS/MJB	Draft for CPS PWG Review
0.8	20150918	MJB	Public Review Draft
0.9	20160516	MJB	Resolved Comments Draft
0.9	20160517	ERG	Draft for CPS PWG Review
1.0	20160526	ERG	First Release Version

Release versioning:

0.7 + {date-time} Working drafts

0.8 Draft for public review

0.9 Draft for final review

1.0 First release version

Table of Contents

Table of Figures

Table of Tables

Disclaimer

This document has been prepared by the Cyber-Physical Systems Public Working Group (CPS PWG), an open public forum established by the National Institute of Standards and Technology (NIST) to support stakeholder discussions and development of a framework for cyber-physical systems. This document is a freely available contribution of the CPS PWG and is published in the public domain.

Certain commercial entities, equipment, or materials may be identified in this document in order to describe a concept adequately. Such identification is not intended to imply recommendation or endorsement by the CPS PWG (or NIST), nor is it intended to imply that these entities, materials, or equipment are necessarily the best available for the purpose. All registered trademarks or trademarks belong to their respective organizations.

Acknowledgement

We would like to acknowledge the valuable leadership of the following individuals who helped guide the participants through the framework activity:

NIST Leadership:

Edward Griffor, David Wollman, Christopher Greer

Reference Architecture:

Industry Cochairs:	Stephen Mellor, Shi-Wan Lin
Academic Cochairs:	Janos Sztipanovits
NIST Cochairs:	Abdella Battou

Security:

Industry Cochairs:	Claire Vishik, Larry John
Academic Cochairs:	Bill Sanders
NIST Cochairs:	Victoria Pillitteri, Stephen Quinn

Use Cases:

Industry Cochairs:	Stephen Mellor
Academic Cochairs:	John Baras
NIST Cochairs:	Eric Simmon

Data Interoperability:

Industry Cochairs:	Peggy Irelan, Eve Schooler
Academic Cochairs:	Larry Lannom
NIST Cochairs:	Martin Burns

Timing:

Industry Cochairs:	Sundeep Chandhoke
Academic Cochairs:	Hugh Melvin
NIST Cochairs:	Marc Weiss

Additionally, we would like to express our appreciation to all the participants from industry, government and academia that, through their generous donation of their intellect, experience and precious time, made this effort a success.

Executive Summary

Cyber-physical systems (CPS) are smart systems that include engineered interacting networks of physical and computational components. CPS and related systems (including the Internet of Things (IoT) and the Industrial Internet) are widely recognized as having great potential to enable innovative applications and impact multiple economic sectors in the worldwide economy.

In mid-2014, NIST established the CPS Public Working Group (CPS PWG) to bring together a broad range of CPS experts in an open public forum to help define and shape key characteristics of CPS, so as to better manage development and implementation within and across multiple "smart" application domains, including smart manufacturing, transportation, energy, and healthcare.

The objective of the CPS PWG is to develop a shared understanding of CPS and its foundational concepts and unique dimensions (as described in this "CPS Framework") to promote progress through exchanging ideas and integrating research across sectors and to support development of CPS with new functionalities. While in principle there are multiple audiences for this work, a key audience is the group of CPS experts, architects, and practitioners who would benefit from an organized presentation of a CPS analysis methodology based on the CPS Framework presented as *facets* and *aspects* in this document. The identified key concepts and issues are informed by the perspective of the five expert subgroups in the CPS PWG: Vocabulary and Reference Architecture, Cybersecurity and Privacy, Timing and Synchronization, Data Interoperability, and Use Cases. The CPS analysis methodology is designed as a framework to support the understanding and development of new and existing CPS, including those designed to interact with other CPS and function in multiple interconnected infrastructure environments. This foundation also enables further use of these principles to develop a comprehensive standards and metrics base for CPS to support commerce and innovation. As an example, the CPS Framework could support identification of the commonalities and opportunities for interoperability in complex CPS at scale. The broader audience for this work includes all CPS stakeholders, who may be interested in broadening individual domain perspectives to consider CPS in a holistic, multi-domain context.

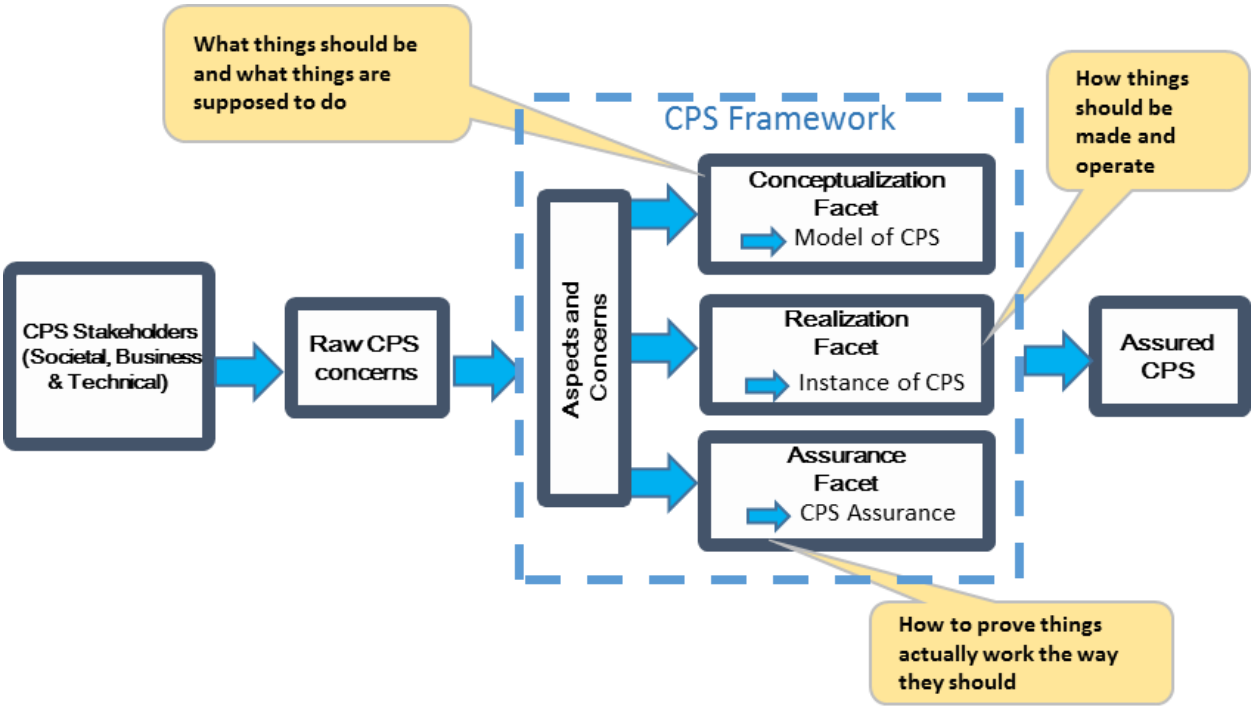
The first stage in the three-stage work plan of the CPS PWG was to develop initial "Framework Element" documents in each of the five expert subgroups. In the second stage, these documents were combined into an initial draft CPS Framework and then revised and improved to create this draft document. The documented discussions of the subgroups have been included here as Appendices A through C. After public review and finalization of the CPS Framework Release 1, the applicability of this approach will be assessed in selected CPS domains, leading to a planned future road mapping activity to both improve the CPS Framework and develop understanding and action plans to support its use in multiple CPS domains.

With respect to this draft CPS Framework, the first goal was to derive a unifying framework that covers, to the extent understood by the CPS PWG participants, the range of unique dimensions

of CPS. The second goal is to populate a significant portion of the CPS Framework with detail, drawing upon content produced by the CPS PWG subgroups and leadership team. It is important to note that there are sections of this draft CPS Framework that are not fully developed at this time. It is anticipated that additional content will be included in the future revisions to this document.

The diagram below shows this analysis proceeding in a series of steps as undertaken within the reference architecture activity:

1. **Identify** domains of CPS; these are the areas of deployment of CPS in which stakeholders may have domain-specific and cross-domain concerns.
2. **Identify** cross-cutting concerns, like societal, business, technical, etc. Stakeholders can have concerns that overlap or are instances of broader conceptual concerns.
3. **Analyze** cross-cutting concerns to produce aspects, or grouping of conceptually equivalent or related concerns.
4. **Address** concerns (aspects) through activities and artifacts organized within three fundamental facets of conceptualization, realization, and assurance.



Two iterations of integration and analysis produced the following Framework elements:

Domains. It is intended that the Framework can be applied to concrete CPS application domains, e.g., manufacturing, transportation, and energy, as both a specialization of these common conceptions and descriptions and a means for integrating domains for coordinated functions. Conversely, these specializations may validate and help to enhance the overarching CPS conceptions and descriptions. Within and across these domains, stakeholders have a variety of concerns or interests.

Facets. Facets are views on CPS encompassing identified responsibilities in the system engineering process. They contain well-defined activities and artifacts (outputs) for addressing concerns. There are three identified facets:

- The conceptualization facet captures activities related to the high-level goals, functional requirements, and organization of CPS as they pertain to what a CPS or its components should be and what they are supposed to do. It provides as its overarching output a conceptual model of the CPS.
- The realization facet captures the activities surrounding the detailed engineering design, production, implementation, and operation of the desired systems. These activities include tradeoff analyses, detailed engineering design and simulation(s), and more, that drive towards and are responsible for realization of a CPS.
- The assurance facet deals with obtaining confidence that the CPS built in the realization facet satisfies the model developed in the conceptualization facet. Its activities include evaluating the claims, argumentation, and evidence as required to address important (and sometimes mandatory) requirements of design, policy, law, and regulation.

Aspects. Aspects are high-level groupings of cross-cutting concerns. Concerns are interests in a system relevant to one or more stakeholders. The identified aspects are listed below:

- Functional
- Business
- Human
- Trustworthiness[1]
- Timing
- Data
- Boundaries
- Composability
- Lifecycle

Using the concepts of facets and aspects, this draft CPS Framework describes a CPS analysis methodology, in which the activities identified in the facets are implemented in a coordinated approach to address concerns throughout the entire life cycle, using a range of development approaches, such as waterfall, agile, spiral and iterative.

[1] Trustworthiness includes security, privacy, safety, reliability, and resilience.

In summary, this draft CPS Framework takes the foundation-building work done within the CPS PWG and integrates and reorganizes that work to form a cohesive document based on the identified concepts of facets and aspects.

It is hoped that this Framework will satisfy the need for a reference CPS description language on which tools, standards, and documented applications can be based.

Further input and comments from a broad audience will inform CPS PWG efforts to build out and improve this CPS Framework.

1 Introduction

This section provides an introduction for the document. It comprises the following:

- Section 1.1 provides a brief overview of cyber-physical systems.
- Section 1.2 defines the purpose of the document.
- Section 1.3 explains the scope of the document.
- Section 1.4 explains the organization of the rest of the document.

1.1 Overview

1.1.1 Background

Cyber-physical systems (CPS) are smart systems that include engineered interacting networks of physical and computational components.[2] These highly interconnected and integrated systems provide new functionalities to improve quality of life and enable technological advances in critical areas, such as personalized health care, emergency response, traffic flow management, smart manufacturing, defense and homeland security, and energy supply and use. In addition to CPS, there are many words and phrases (Industrial Internet, Internet of Things (IoT), machine-to-machine (M2M), smart cities, and others)[3] that describe similar or related systems and concepts.[4] There is significant overlap between these concepts, in particular CPS and IoT, such that CPS and IoT are sometimes used interchangeably; therefore, the approach described in this CPS Framework should be considered to be equally applicable to IoT.

The impacts of CPS will be revolutionary and pervasive; this is evident today in emerging autonomous vehicles, intelligent buildings, smart energy systems, robots, and smart medical devices. Realizing the full promise of CPS will require interoperability among heterogeneous components and systems, supported by new reference architectures using shared vocabularies and definitions. Addressing the challenges and opportunities of CPS requires broad consensus

[2] A technical definition of CPS is provided in Section 2.1, and for convenience, CPS may be considered to be either singular or plural.

[3] Some of these terms are defined in Appendix A.

[4] CPS will be the focus of this document; however, terminology distinctions may be introduced to aid the reader where beneficial or informative.

in foundational concepts, and a shared understanding of the essential new capabilities and technologies unique to CPS. NIST has established the CPS Public Working Group (CPS PWG) as an open forum to foster and capture inputs from those involved in CPS, both nationally and globally.

The CPS PWG was launched in mid-2014 with the establishment of five subgroups (Vocabulary and Reference Architecture, Use Cases, Cybersecurity and Privacy, Timing and Synchronization, and Data Interoperability.)[5] Initial CPS PWG "Framework Element" documents were produced by each of the subgroups in December 2014, then integrated, reorganized, and refined to create this draft CPS Framework (see the Revision Tracking page at the beginning of the document). The CPS Framework is intended to be a living document and will be revised over time to address stakeholder community input and public comments; some sections of the document are incomplete and will be developed and extended over time.

The core of the CPS Framework is a common vocabulary, structure, and analysis methodology. As a process method, the CPS analysis methodology should enable and facilitate CPS systems engineering using a range of development approaches, such as waterfall, agile, spiral, and iterative. There are many well-documented system engineering process documents and flows – e.g., TOGAF [4] and CMMI [5]. This Framework, however, focuses on the detailed scope of CPS and the specific concerns implementers and analysts have in designing and understanding them. The concepts described in this document map cleanly to the more general methods and therefore are complementary to them rather than competitive.

The purpose of this CPS Framework is to allow for a comprehensive analysis of CPS. The CPS Framework captures the generic functionalities that CPS provide, and the activities and artifacts needed to support conceptualization, realization, and assurance of CPS. This analysis methodology includes addressing concerns that are specific to CPS and those that are applicable to any device or system. By this means, a complete methodology is proposed with common vocabulary and structure, which emphasizes CPS concerns but not to the exclusion of others.

1.1.2 What Is Different about CPS

CPS go beyond conventional product, system, and application design traditionally conducted in the absence of significant or pervasive interconnectedness. There are many reasons for this, including the following:

[5] Additional information on the NIST CPS PWG is available at www.cpspwg.org and http://www.nist.gov/cps/.

- **The combination of the cyber and the physical, and their connectedness, is essential to CPS.** A CPS generally involves sensing, computation and actuation. CPS involve traditional information technology (IT) as in the passage of data from sensors to the processing of those data in computation. CPS also involve traditional operational technology (OT) for control aspects and actuation. The combination of these IT and OT worlds along with associated timing constraints is a particularly new feature of CPS.

- **A CPS may be a System of Systems (SoS).** As such, it may bridge multiple purposes, as well as time and data domains, hence requiring methods of translation or accommodation among these domains. For example, different time domains may reference different time scales or have different granularities or accuracies.

- **Emergent behaviors are to be expected of CPS, due the open nature of CPS composition**. Understanding a behavior that cannot be reduced to a single CPS subsystem, but comes about through the interaction of possibly many CPS subsystems, is one of the key analysis challenges. For example, a traffic jam is a detrimental emergent behavior; optimal energy distribution by the smart grid where power consumers and producers work together is a desirable positive emergent effect.

- **CPS need a methodology to ensure interoperability, managing evolution, and dealing with emergent effects.** Especially in large scale CPS such as smart grid and smart city, many of the subsystems are the responsibility of different manufacturers.

- **CPS may be repurposed beyond applications that were their basis of design.** For example, a cell phone in a car may be used as a mobile traffic sensor, or energy usage information may be used to diagnose equipment faults.

- **CPS are noted for enabling cross-domain applications.** As an example, consider the intersection of the domains: manufacturing and energy distribution systems, smart cities, and consumer-based sensing.

- **CPS potential impact on the physical world and their connectedness bring with them heightened concern about trustworthiness.** There is a more urgent need for emphasis on security, privacy, safety, reliability, and resilience, and corresponding assurance for pervasive interconnected devices and infrastructures. As an example, CPS networks may have “brokers” and other infrastructure-based devices and aggregators that are owned and managed by third parties, resulting in potential trust issues – e.g., publish and subscribe messaging, certificate authorities, type and object registries.

- **CPS should be freely composable.** Components are available that may be combined into a system dynamically, and the system architecture may be modified during runtime to address changing concerns. There are challenges, however. For example, timing composability may be particularly difficult. Also, it may not always be necessary or

desirable to purchase assets to build a system; instead, services can be purchased on a per-use basis, with users only paying for using the resources needed for a specific application and at the specific time of usage.

- **CPS must be able to accommodate a variety of computational models.** Each CPS application has computational and physical components and the range of platform and algorithm complexity is broad.

- **CPS must also support a variety of modes of communication.** CPS comprise systems that range from standalone to highly networked. They may use legacy protocols or anything up to more object exchange protocols. And they may be anywhere from power constrained to resource rich.

- **The heterogeneity of CPS leads them to display a wide range of complexity.** The complexity associated with the sensing and control loop(s) with feedback that are central to CPS must be well addressed in any design. This complexity must be accommodated by any framework for CPS, including sensors that range from basic to smart; static and adaptive sensors and control; single-mode and multi-faceted sensors; control schemes that can be local, distributed, federated, or centralized; control loops that rely on a single data source and those that fuse inputs; and so on. Interactions can be loosely coupled, as in repurposing of distributed sensing that is part of an existing CPS, as well as tightly coupled, as in telemedicine or smart grid operations. Coupling is both an opportunity to fulfill the vision of CPS and a challenge to CPS assurance. Emergent behaviors can become part of the intent of new services or may be unwanted. To mitigate complexity, CPS may be a product of co-design. In co-design, the design of the hardware and the software are considered jointly to inform tradeoffs between the cyber and physical components of the system.

- **There is typically a time-sensitive component to CPS, and timing is a central architectural concern.** A bound may be required on a time interval, i.e., the latency between when a sensor measurement event actually occurred and the time at which the data was made available to the CPS. The accuracy of event timestamps is a constraint on a time value, in this case between the actual time of the event and the value of the timestamp. Accurate time intervals are useful for coordinating actions in CPS of large spatial extent. Accurate timestamps in CPS are typically needed to facilitate better root cause analysis, and sometimes also for legal or regulatory reasons.

- **CPS are characterized by their interaction with their operating environment** (as indicated by the sensing and control loop(s) discussed above). CPS, together or individually, ‘measure’ and sense and then calculate and act upon their environment, typically changing one or more of the observed properties (thus providing closed loop control). The CPS environment typically includes humans, and humans function in a different way than the other components of a CPS. The architecture must support a

variety of modes of human interaction with CPS to include: human as CPS controller or partner in control; human as CPS user; human as the consumer of CPS output; and human as the direct object of CPS to be measured and acted upon.

1.2 Purpose

The success of this CPS Framework can be assessed by its usefulness as guidance in designing, building, and verifying CPS and as a tool for analyzing complex CPS. It should aid users in determining the properties of a CPS, and provide guidance such that two CPS instance architectures, independently derived or tailored from this Framework, are in substantial alignment. That is, they can be mutually understood through the organizational and descriptive means of this Framework, and in doing so their real or potential interactions can be more easily understood.

By providing a framework for discussion of, design of, and reasoning about CPS, a common foundation will be established from which a myriad of interoperable CPS can be developed, safely and securely combined, verified, and delivered to the public, government, and researchers. If broadly adopted, this Framework will serve to enable activity in research and development that will produce reliable, well-designed, easily-integrated CPS-based products and services.

The framework uses many terms that have been defined in many other documents. We have provided definitions in Appendix E to indicate how we have used language in this document. Where possible we have drawn on documents in other standards, however some words are used differently in different standards and in different industries. There are also some words that are commonly used that we have not defined here. An example is the word precision or precise. This word is used in many contexts and in some cases with different meanings. It is hoped that such words are made clear by the context in which they are used.

This document defines a CPS as follows: Cyber-physical systems integrate computation, communication, sensing, and actuation with physical systems to fulfill time-sensitive functions with varying degrees of interaction with the environment, including human interaction.

A CPS conceptual model is shown in Figure 1. This figure is presented here to highlight the potential interactions of devices and systems in a system of systems (SoS) (e.g., a CPS infrastructure). A CPS may be as simple as an individual device, or a CPS can consist of one or more cyber-physical devices that form a system or can be a SoS, consisting of multiple systems that consist of multiple devices.

This pattern is recursive and depends on one's perspective (i.e., a device from one perspective may be a system from another perspective). Ultimately, a CPS must contain the decision flow together with at least one of the flows for information or action. The information flow represents digitally the measurement of the physical state of the physical world, while the

action flow impacts the physical state of the physical world. This allows for collaborations from small and medium scale up to city/nation/world scale.

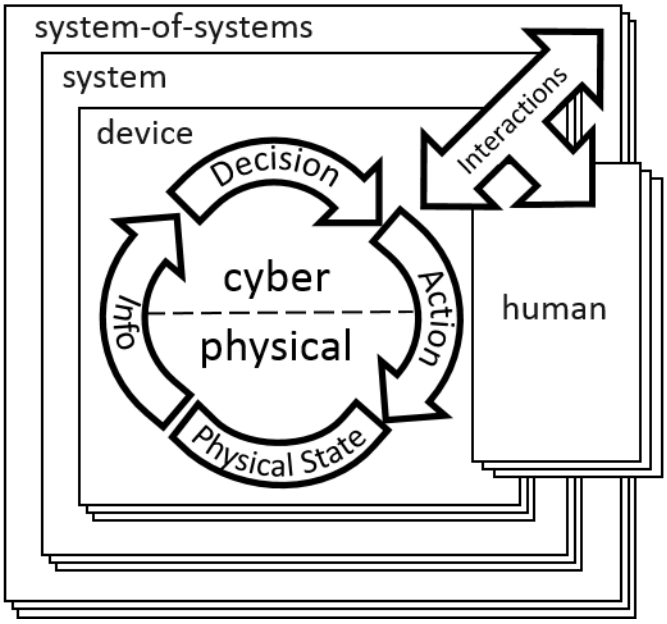


Figure 1: CPS Conceptual Model

1.3 Scope

The scope of CPS is very broad by nature, as demonstrated in the M2M sector map in Figure 2. There are large number and variety of domains, services, applications, and devices. This figure displays CPS focused on the IoT.[6] This broad CPS scope includes cross-cutting functions (i.e., functions that are derived from critical and overriding CPS concerns) that are likely to impact multiple interacting CPS domains. The CPS Framework will facilitate users' understanding of cross-cutting functions. Examples include safety, security, and interoperability.

[6] Note that the inclusion of Figure 2 is designed only to illustrate the scope of deployed, commercial CPS, but not a particular or preferred architecture for studying it.

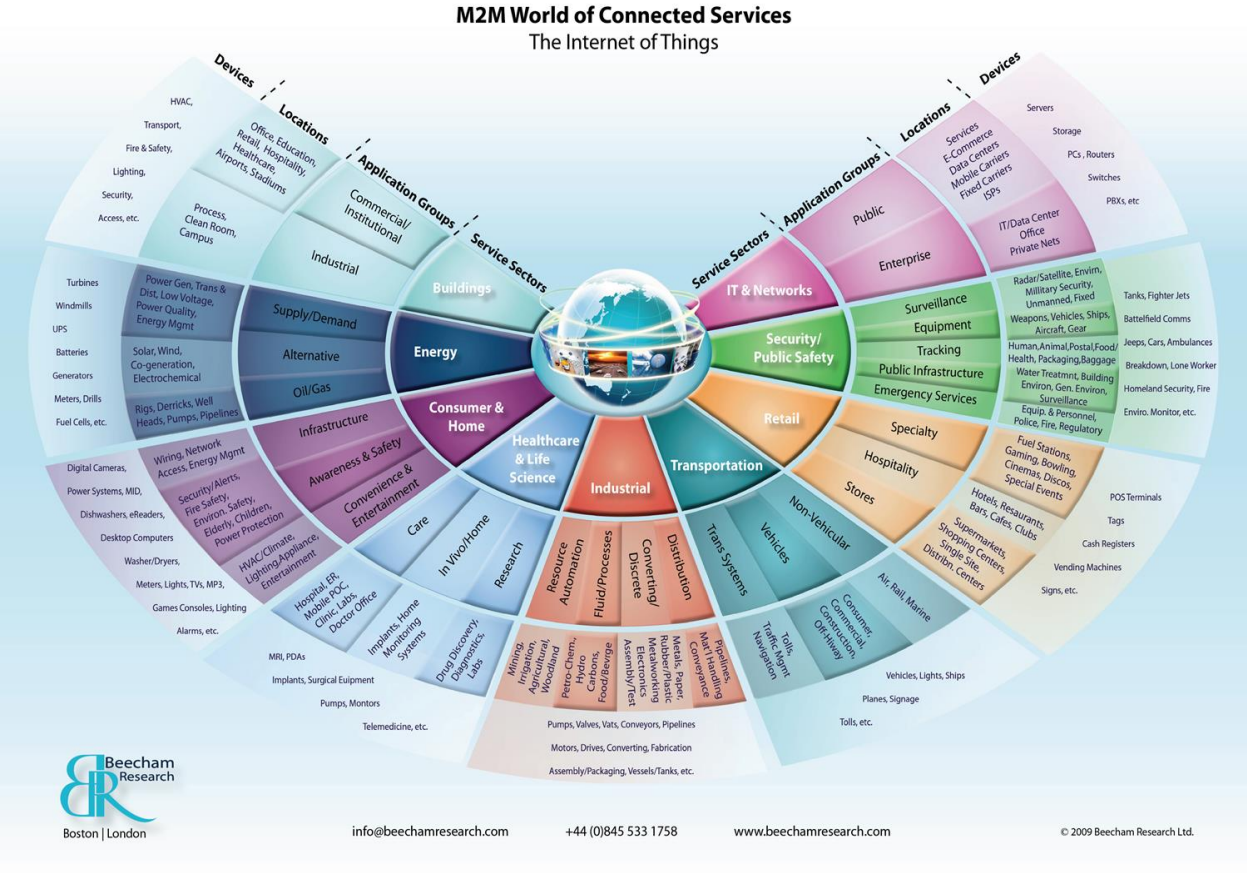


Figure 2: Segmentation of M2M Market[7]

The scope of the CPS Framework includes the full dimensionality of CPS and the entire systems engineering process. That is, rather than address "selected topics," it intends to provide a comprehensive tool for the analysis and description of CPS. It is quite possible that this initial draft CPS Framework will have missed some important elements of CPS. With input from stakeholders including through a public review process, these gaps will be addressed in future releases.

As an example of the scope of CPS, consider the case of smart traffic systems. Smart traffic systems consist of smart traffic monitoring and control infrastructure, advanced traffic control centers powered by predictive analytics on real-time traffic data, autonomous vehicles

[7] Courtesy of Beecham Research, used by permission, http://www.beechamresearch.com/article.aspx?id=4, 2009

interacting with peer vehicles in proximity, and traffic control systems. This section provides examples of how the use of CPS can impact a smart traffic system. This section also provides context for Appendix B, in which a simplified traffic-related emergency response scenario is used as an analysis example to demonstrate how to exercise the CPS analysis methodology of this CPS Framework.

Figure 3: Smart Transportation[8]

CPS controls have a variety of levels of complexity ranging from automatic to autonomic. A prominent example of CPS in smart traffic is autonomous vehicles, which are themselves systems of CPS. The functions of CPS within an autonomous vehicle are orchestrated, collaborated, and coordinated to achieve the overall autonomous functions of the vehicle.

[8] ETSI http://www.etsi.org/technologies-clusters/technologies/intelligent-transport, used with permission

Another smart traffic CPS is the on-location smart traffic control systems. They are installed in street intersections to sense and measure local traffic patterns and conditions so they can apply commands to the traffic signals to orchestrate the movement of vehicles passing the intersections based on prescribed objectives. These on-location smart traffic control systems may be orchestrated by regional traffic control centers to optimize overall traffic flows.

CPS can collaborate with each other to produce effects that are greater than the sum of the parts. An example of collaboration of CPS is the collaboration of vehicles in proximity to avoid collisions. These vehicles communicate with each other in the cyber space, dynamically forming ad hoc communities to inform others of the actions each of them is taking that may affect the communities of vehicles. Examples of such actions include applying a brake or changing lanes. They also interact, albeit indirectly, in the physical space by continuously sensing and measuring the movement and trajectory of neighboring vehicles. The information gathered from both the cyber and physical spaces is then synthesized to gain an understanding of the state and intent of the vehicles in proximity. From this understanding, and based on prescribed objectives (e.g., to avoid collision, a physical effect), control decisions are continuously made to produce the desired physical effects in the vehicle in question, e.g., to slow down, stop, accelerate, or change course in order to avoid the undesired effects, such as collisions between vehicles or between vehicles and other objects.

CPS can be orchestrated by a cyber system that communicates logically with them. An example of this orchestration is the computational unit in an autonomous vehicle strongly orchestrating the activities between the steering, braking, and powertrain CPS. Another example of this is a traffic control unit using wireless signaling to orchestrate autonomous vehicles passing through a street intersection.

The SoS domain enables the complex management of CPS and supports emerging behavior. In smart traffic, traffic monitoring systems send data to the on-location traffic control units and to their respective regional traffic control centers. Vehicles also report driving data to the traffic Internet, which can in turn be routed to the relevant traffic control centers. The information component for the regional traffic control centers analyzes these data to understand the traffic conditions and patterns. The application component synthesizes this information with other information such as traffic patterns in the neighboring regions, current and forecast weather conditions, current and pending large public events, and road accident reports. It takes into account in its model the constraints imposed by the objectives, such as minimizing traffic delay, minimizing air and noise pollution, increasing safety and enhancing security, and reducing energy consumption. It optimizes the traffic routing patterns and sends high-level instructions to on-location traffic control units to orchestrate regional traffic patterns. It coordinates vehicle traffic flows by broadcasting advice to vehicles to suggest alternative routes. The application component may assist emergency response in locating accident sites for rescue and recovery. It may interact with the business component to plan road or facility repairs accounting for either or both material or work crews. It may interact with the business component to schedule

predictive maintenance or repairs on the traffic control infrastructure based on information provided by the information and entity management component that manages the CPS in the traffic control infrastructure.

Furthermore, sensory data gathered from the vehicles correlated with geolocation, climate, and season data, as well as road construction and maintenance records, can be analyzed to derive information on road and bridge conditions at precise locations, and their relations to the interworking of climate, season, patterns of usage, construction materials and procedures, and maintenance frequency. Optimal preventive maintenance can be planned in relation to usage patterns, season, and cost. New materials and optimal procedures can be developed for specific usage patterns and climates.

1.4 Organization of This Document

Beyond the introductory material in this section, this Framework document is organized as follows:

Section 2: The CPS Framework – This section describes the CPS environment and stakeholder concerns, and provides an overview of the CPS Framework analysis methodology with its core concepts of *facets* (components of the systems engineering process with associated activities and artifacts) and *aspects* (groupings of cross-cutting concerns). The Framework itself is derived from the CPS PWG 's analysis of Sections 3 and 4, which elaborate the extent of CPS relative to their topics of facets and aspects.

Appendix A Facets of the CPS Framework – This section describes the conceptualization, realization, and assurance facets, their primary results (model of CPS, instance of CPS, and assurance of CPS), and their associated activities and artifacts.

Appendix B: Aspects of the CPS Framework – This section describes groups of cross-cutting concerns organized into aspects. For each aspect, the dimensionality of CPS with regard to the concerns is discussed in a comprehensive, coherent, and coordinated manner. Note that several aspects were identified in the analysis by CPS PWG participants and subgroups but are not covered in depth in this draft Framework; these aspects are anticipated to be further developed in the future.

Appendix C: Use Case Analysis – This section describes the role of use cases in the CPS Framework and their importance to understanding the functional requirements for these systems.

Appendix D: References – This section provides references to a variety of CPS-related articles, standards, and other material.

Appendix E: Definitions and Acronyms – This appendix provides a set of acronyms and definitions applicable to this document.

Appendix F: Applying the CPS Framework – This appendix uses a simplified "Emergency Response" scenario involving multiple CPS to illustrate how owners, designers, engineers, and operators can apply the Framework to analyze CPS in an operational context.

2 The CPS Framework

This section defines the CPS Framework at a high level. The components of the section are:

- Section 2.1 provides an overview of CPS and key CPS concepts.
- Section 2.2 explains the derivation of the CPS Framework.
- Section 2.3 describes potential uses of the CPS Framework.
- Section 2.4 contains the complete description of the CPS Framework.
- Section 2.5 discusses related standards and activities.
- Section 2.6 summarizes Section 2.

2.1 Overview

The focus of this Framework is to develop a CPS analysis methodology and a vocabulary that describes it. It includes the identification of CPS domains, facets, aspects, concerns, activities, and artifacts. These terms are defined in the context[9] of this Framework and are introduced later in this section.

To appreciate the scope of coverage that the CPS Framework addresses, this section briefly discusses the dimensionality of CPS. This presentation covers the entire scope of CPS as opposed to section 1.1.2 which emphasized unique differences of highly connected systems versus conventional systems.

- CPS are frequently systems of systems (SoS), and the architectural constructs should be able to be applied recursively or iteratively to support this nested nature of CPS. The sensing/control and computational nature of CPS generally leads to emergent higher levels of behavior and system intelligence.

- CPS should be characterized by well-defined components. They should provide components with well-known characteristics described using standardized semantics and syntax. Components should use standardized component/service definitions, descriptions, and component catalogs.

- CPS should support application and domain flexibility. To do this, the definition of the components should be flexible and open ended. The architecture should support the provision of accurate descriptions of things to allow for flexibility in virtual system creation and adaption and to promote innovation. It should also support a large range

[9] In a technology space as broad as CPS, any term will have more than one meaning in practice by different audiences. Therefore, the definitions and usage in this Framework are intended to have the scope of this Framework and are not proposed for universal meaning.

of application size, complexity, and workload. The same components that are used in a very simple application should also be usable in a very large, complex, distributed system. Ideally the components can be assembled and scaled quickly, even during runtime. CPS architecture should allow composition from independent, decoupled components for flexibility, robustness, and resilience to changing situations. Decoupling should also exist between architectural layers, allowing each layer to be modified and replaced without unwittingly affecting the other layers. In order for the system to integrate different components, the interfaces to these components should be based on well-defined, interpretable, and unambiguous standards. Further, standardization of interfaces will allow for easy provisioning of various components by any systems envisioned today and into the future. By allowing internal component flexibility while providing external interoperability through standardized interfaces, customization can be achieved. This supports desirable diversity of application and scalability.

- CPS frequently perform critical applications, so the CPS architecture must support the level of reliability needed to meet requirements. It should provide the ability to resist change due to external perturbations or to respond to those changes in ways that preserve the correct operation of the critical application.

- Security is a necessary feature of the CPS architecture to ensure that CPS capabilities are not compromised by malicious agents, and that the information used, processed, stored and transferred has its integrity preserved and is kept confidential where needed. The nature of CPS not only increases the consequences of a breach but also introduces additional types of vulnerabilities. For example, timing in a CPS has unique vulnerabilities different from traditional data vulnerabilities considered in cybersecurity. Security needs to be built into CPS by design in order to be sufficiently flexible to support a diverse set of applications. This security should include component security; access control; as well as timing, data and communications security. Security must be considered in combination with other prioritized and potentially conflicting concerns, such as privacy, safety, reliability, and resilience, in a comprehensive risk management framework.

- Data exchange is a prominent dimension of CPS operation. The nature of data and its reliability, type, identity, and discovery are all key attributes that allow for a common understanding of data conveyed through communications in and among CPS. Often, data are “fused” or combined with other data to anonymize or enrich it or to summarize it for the benefit of users. Access to data is often constrained by “rights” or “privileges.”

- Components that contain sensors and/or actuators should have an appropriate level of awareness of physical location and time. For example, the accuracy requirement for location will change based upon the application. To support such applications,

components may need the ability to access and/or report both location and the associated uncertainty of the location.

- Additionally, CPS architectures should support legacy component integration and migration. Legacy devices have physical artifacts, software, protocols, syntax, and semantics that exist due to past design decisions, and they may be inconsistent with the current architectural requirements. New components and systems should be designed so that present or legacy devices do not unnecessarily limit future system evolution. As even new components will become legacy in the future, a plan for adaptation and migration of legacy systems and standards should be created to avoid stranded investments, if possible. Legacy components should be integrated in a way that ensures that security and other essential performance and functional requirements are met.

2.2 Derivation of the Framework

A useful reference for the terminological and definitional conventions relating to systems architecture and systems architecture frameworks is ISO/IEC/IEEE 42010 [2]. For the purposes of this section, here are a few of these conventions:

- An *architecture framework* consists of the "conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders."
- A *concern* is an "interest in a system relevant to one or more of its stakeholders."
- An *architecture view* consists of "work product expressing the architecture of a system from the perspective of specific system concerns."
- An *architecture viewpoint* consists of "work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns."

Another key reference relevant to the construction of this Framework is ISO/IEC/IEEE 25288 on System Life Cycle processes [3], which describes a number of processes and outcomes to guide system engineering.

Building on these two references, this CPS Framework derives the core notions of facets, activities, artifacts, aspects, and concerns. Note that while these are two key references to general systems engineering principles, the Framework emphasizes the nature and function of CPS specifically.

2.2.1 Key Elements of the Framework

The discussions in Appendix A on Facets and Appendix B on Aspects are designed to elaborate the respective topics relative to CPS. This section distills from this content the CPS Framework. Here are the key elements of the Framework:

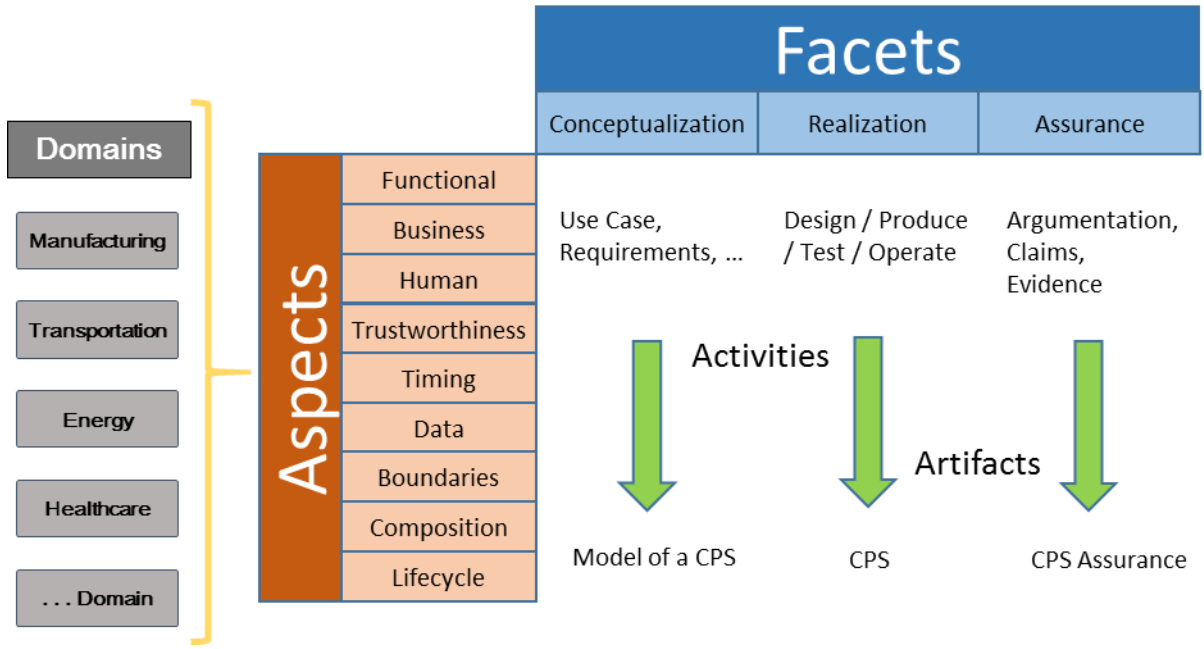


Figure 4: CPS Framework – Domains, Facets, Aspects

- *Domains* represent the different application areas of CPS as shown in Figure 4.
- *Concerns,* as expressed by many different stakeholders in their unique and collective viewpoints, are a fundamental concept that drives the CPS Framework methodology. They are addressed throughout the CPS development cycle. Concerns that are conceptually equivalent or related are grouped into Aspects, which are addressed by activities within the facets.
- *Properties* are the concrete assertions that address the concerns. They include requirements, design elements, tests, and judgments.
- *Aspects* consist of groupings of conceptually equivalent or related concerns.[10] A listing of aspects is provided in this Framework in Appendix B; there may be modifications or

[10] Aspects are sometimes called cross-cutting concerns.

other valid groupings of concerns that may benefit a particular application of the CPS Framework in a specific context. Note that aspects and concerns are not considered orthogonal. There are nine defined aspects: functional, business, human, trustworthiness, timing, data, boundaries, composition, and lifecycle.

- *Facets* are views on CPS encompassing identified responsibilities in the systems engineering process. Each facet contains a set of well-defined activities and artifacts (outputs) for addressing concerns. There are three identified facets: conceptualization, realization, and assurance.

The Framework was developed through an analysis process that followed a defined sequence of steps. This diagram shows this analysis, working in the context of identified CPS domain(s):

1. **Identify** domains of CPS; these are the areas of deployment of CPS in which stakeholders may have domain-specific and cross-domain concerns.
2. **Identify** cross-cutting concerns, like societal, business, technical, etc. Stakeholders can have concerns that overlap or are instances of broader conceptual concerns.
3. **Analyze** cross-cutting concerns to produce aspects, or grouping of conceptually equivalent or related concerns.
4. **Address** concerns (aspects) through activities and artifacts organized within three fundamental facets of conceptualization, realization, and assurance.

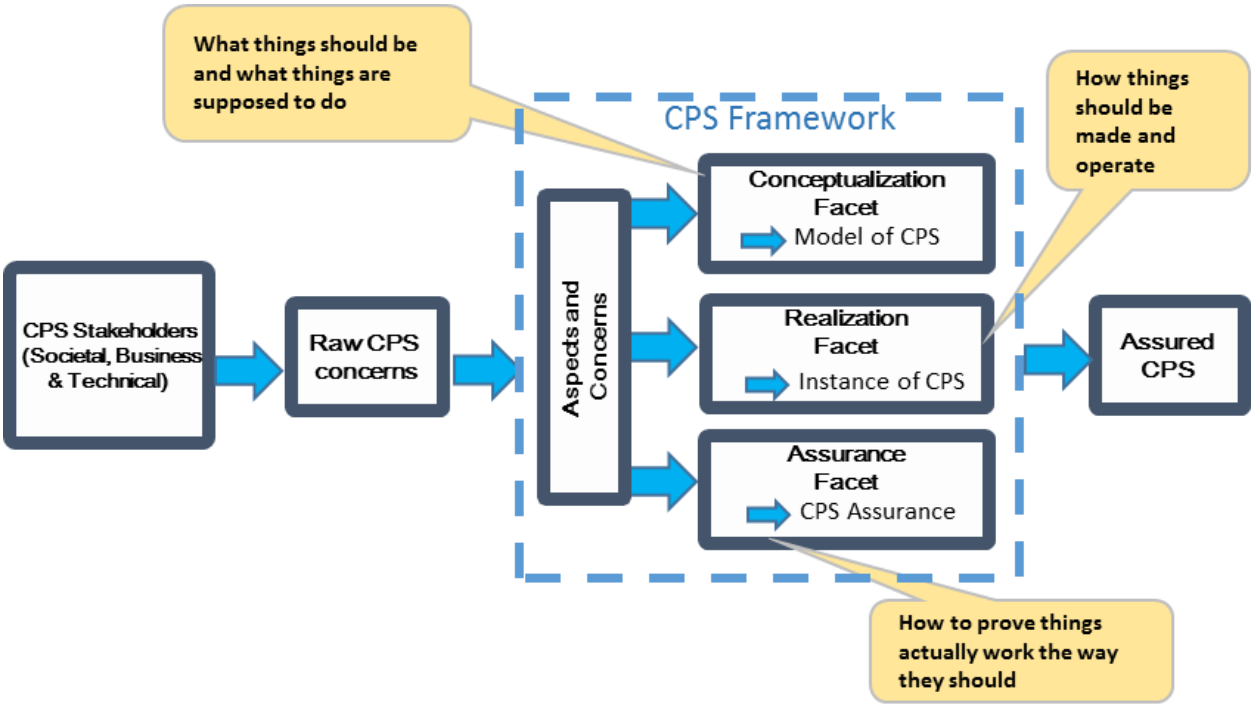


Figure 5: Analysis of CPS and Derivation of Framework

It is intended that the identification and description of the activities, methods, and artifacts in each of the facets can be applied within concrete CPS application domains (e.g., manufacturing, transportation, energy) as a specialization of these common conceptions and descriptions. Conversely, these specializations may validate and help to enhance these conceptions and descriptions.

2.2.2 The Facet as an Activity-Organized Analysis of Concerns

It is a primary goal of the CPS Framework to be *actionable*: to be useful to perform analyses of CPS. With that concern in mind, the prototypical model of a facet is shown in Figure 6:

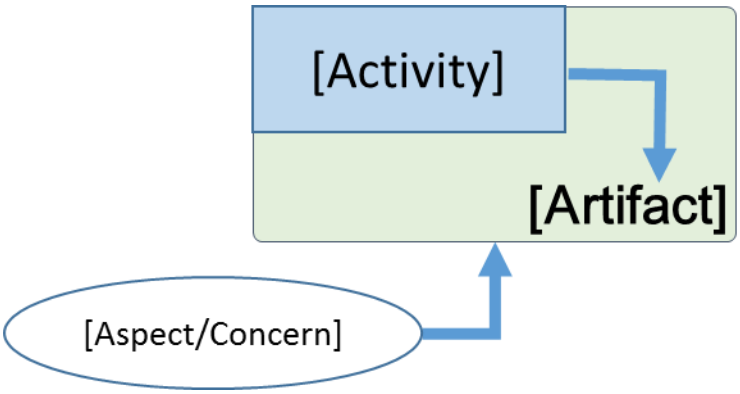


Figure 6: Model of a Facet

A *facet*, therefore, is a collection of activities that produce artifacts that are driven by aspects and their concerns for a CPS.

From this simple model, the three Framework facets are derived as shown in Figure 7:

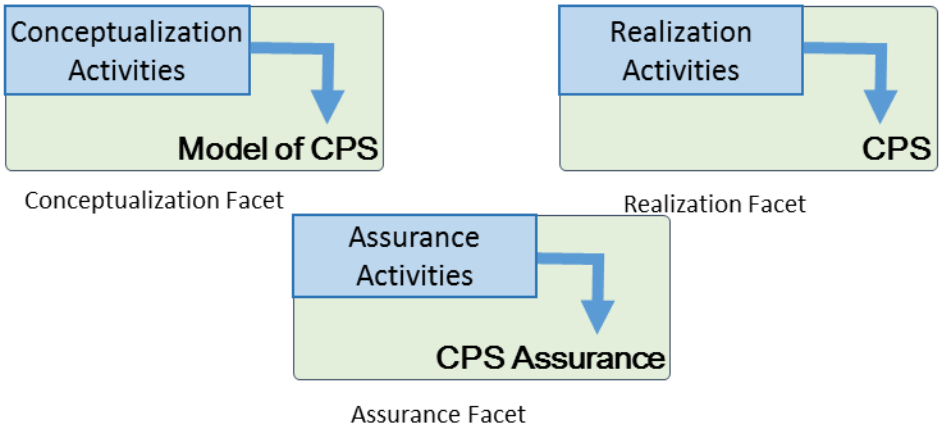


Figure 7: Three Primary Framework Facets

The three facets comprise the traditional systems engineering process (typified by the "VEE" model [8].) By analyzing each aspect within all facets, all cross-cutting concerns can be addressed at every stage of the design, creation and operation of a CPS. Figure 8 illustrates this concept:

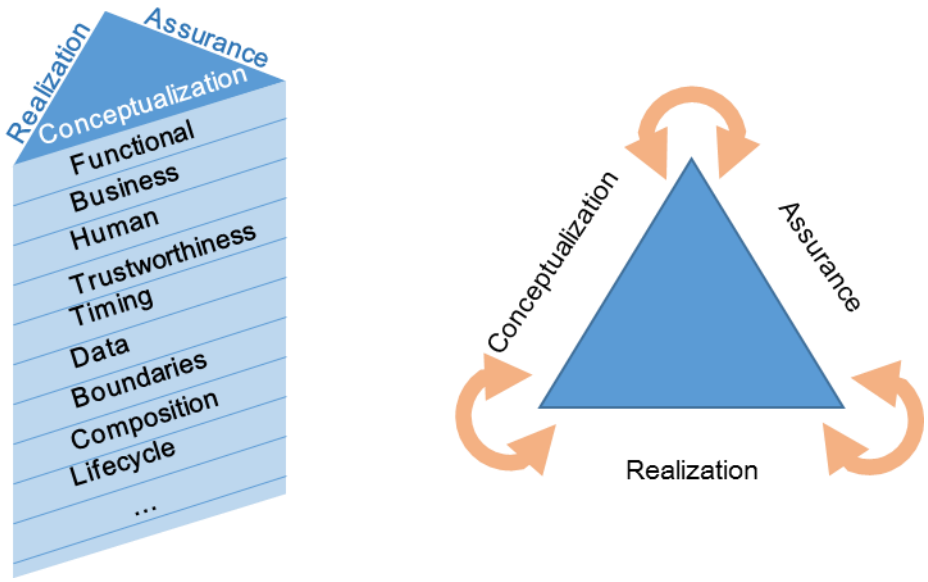


Figure 8: Facets and Aspects

On the left of the figure, the prism illustrates that aspects must be viewed through each face of the prism – the facets. Note that analysis of CPS can be viewed from any face and assembled by navigating one or more times through the faces to obtain the complete view of the CPS as shown on the right.

To emphasize that analysis of CPS need not (and is often not) a waterfall process, the facets should be considered as modes of analysis where transitions are valid at any time during the lifecycle of CPS concept.

2.2.3 Properties

The *conceptualization facet* comprises the set of activities and artifacts that produce a model of a CPS. This model is made up of distinct *properties of the CPS*. These properties are expressions of concerns held by the CPS stakeholders. There can be different kinds of such properties, for example, requirements and model elements. These properties put requirements or constraints on functions and behaviors of the CPS. They represent as well other attributes of the CPS associated with design and build practices and include properties of operation and disposal, i.e., the properties of a CPS span the entire lifecycle of a CPS.

A realized and assembled "CPS model" is an *instance of a CPS*. The *CPS model* is the theoretical ideal of the CPS. The *realization facet* and its activities strive to quantitatively satisfy the aspirational properties of the conceptualization facet. The *assurance facet* then provides the assurance that the conceptualization was realized *as intended*. The properties of the realization facet are made up of design elements and test elements.

CPS can be seen as extensions of human capabilities, including sensing, decision-making, and action. Many times human beings are more than aware of the limitation of their abilities, so assurance methodologies frequently provide both an extension of those abilities and an estimate of the uncertainty inherent in using these extensions. Humans maintain a certain level of situational awareness and many times need to be protected from errors in judgment.

Human beings' capabilities are enhanced through CPS, however CPS assurance and estimates of CPS assurance levels will be important to the success and adoption of CPS and will increase their benefit to mankind.

High on the list of CPS challenges are topics related to *human factors*. The assurance facet is intended to provide a methodology for understanding the scope and limits CPS capabilities. In doing this the interaction between operator and CPS may also be improved. Closer consideration of Figure 9 suggests that there is much research required to better understand the relationship between the cognitive cycle of a human operator and that of the CPS conceived, built, and operated by humans.

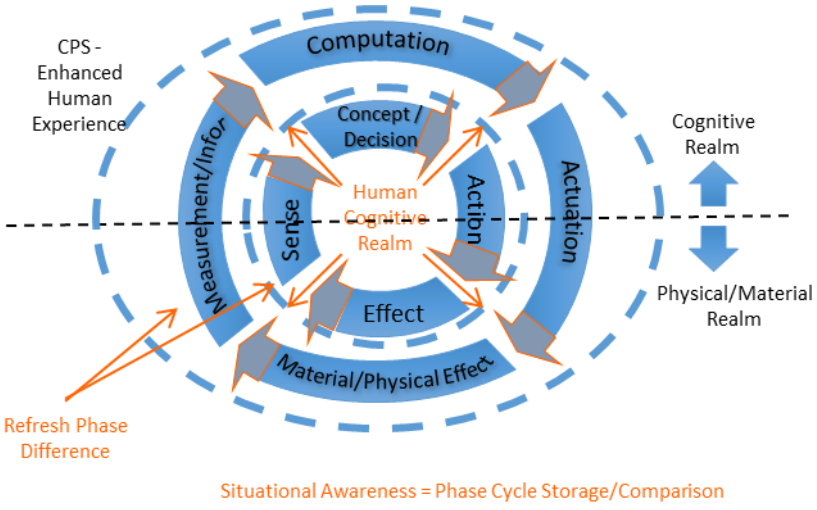


Figure 9: CPS Enhanced Cognitive Cycle

Elements of the *assurance case* of a CPS, developed using this Framework, consists of statements built from data produced during the activities of the first two facets of the framework, conceptualization and realization. The elements are:

- Claims

- Evidence
- Argumentation
- Estimate of confidence

The typical statement of assurance takes the form:

> "The [Evidence] is sufficient to conclude that the [Claims] are true based on the [Argumentation] with this [Estimate of Confidence]."

This is an *assurance judgment.* Judgments are properties of the assurance facet. Ultimately this relationship between evidence, claims, argumentation, and estimate of confidence can be formalized. In this formalization a judgment will have assumptions that are themselves judgments. *Derivation rules* can be used for deriving new judgments from given ones, i.e., one can apply formal reasoning to derive assurance judgments that themselves provide a justification for accepting the derived judgment. As an example, these rules may simply capture the reasoning suggested or dictated by a standard.

An added value of this approach is that such a derivation contains a mapping of all of the evidence used in deriving the judgment. It also provides guidance for how to re-construct the evidence used to conclude that a CPS has the desired properties.

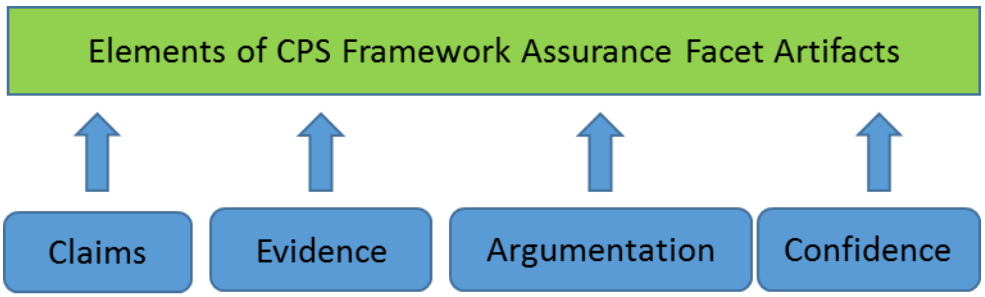


Figure 10: Elements of Assurance

The claims in the assurance facet are formed using the properties of the CPS developed during the conceptualization facet, i.e., the CPS Model. The CPS Model consists of the properties of the intended CPS. The claims in the assurance facet are the assertions that the CPS in question has or satisfies each of these properties. The CPS is said to *satisfy* the CPS Model if it satisfies or has each of the CPS Model properties. In the transportation domain, with ISO 26262 [23] examples, the high-level statement or judgement is that the CPS meets the requirements of the functional safety standard or that the processes of the organization that developed the CPS are ISO 26262 compliant.

The *evidence* in the assurance facet is formed from the artifacts of the realization facet, such as process documentation, design artifacts, test plans, and results, as depicted in Figure 11. They

are determined by the specialization of the realization facet activities and artifacts to their domain and the applicable aspects.

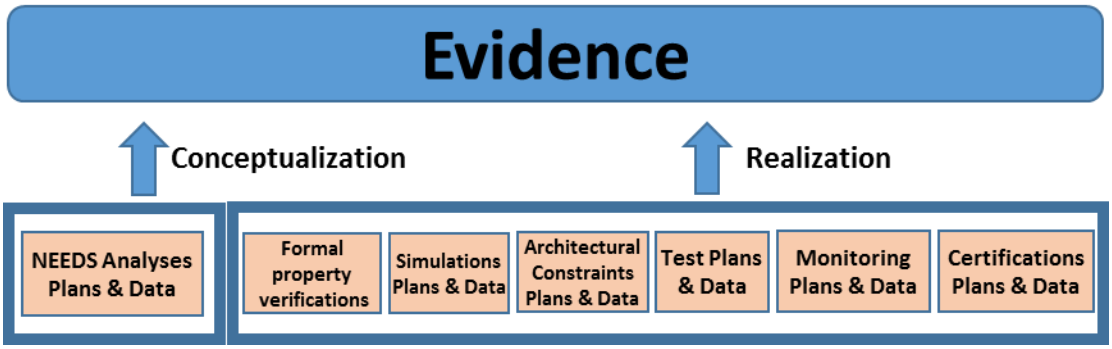


Figure 11: Evidence

As shown in Figure 12, the *argumentation* of the assurance facet is formed from a variety of things, including appeal to:

- Standards
- Best practices/consensus
- Formal methods
- Regulation (proscribed practices)
- Expert judgment (including criteria for being an expert in a domain)

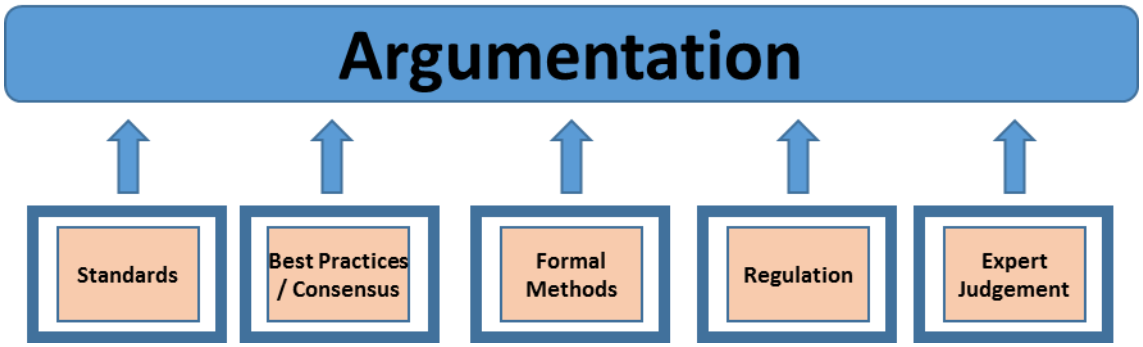


Figure 12: Argumentation

Application of the CPS Framework develops this information when each of the facet activities reviews each of the aspects of the Framework for impact on that activity. The output of that review is an updating of that activity and its artifacts. It is intended to provide criteria for evidence and supporting argumentation in the assurance facet to assure that the concerns in that aspect have been adequately addressed in the activity.

To facilitate addressing of the assurance for any of the properties in a CPS Model, we document the properties of the CPS developed during the conceptualization facet in the form of a tree of properties. Formally a tree is a partially ordered set with a unique root (all nodes trace back ultimately to the same node) and no *cycles* (a cycle corresponds to a node that can be reached from itself following a non-trivial path in the tree). The *property tree of a CPS*, consists of the properties of the CPS Model, ordered under the traceability ordering. The root is the property $P_{M/BC}$ with the successor relation outlined above.

Graphically this tree has the appearance shown in Figure 13:

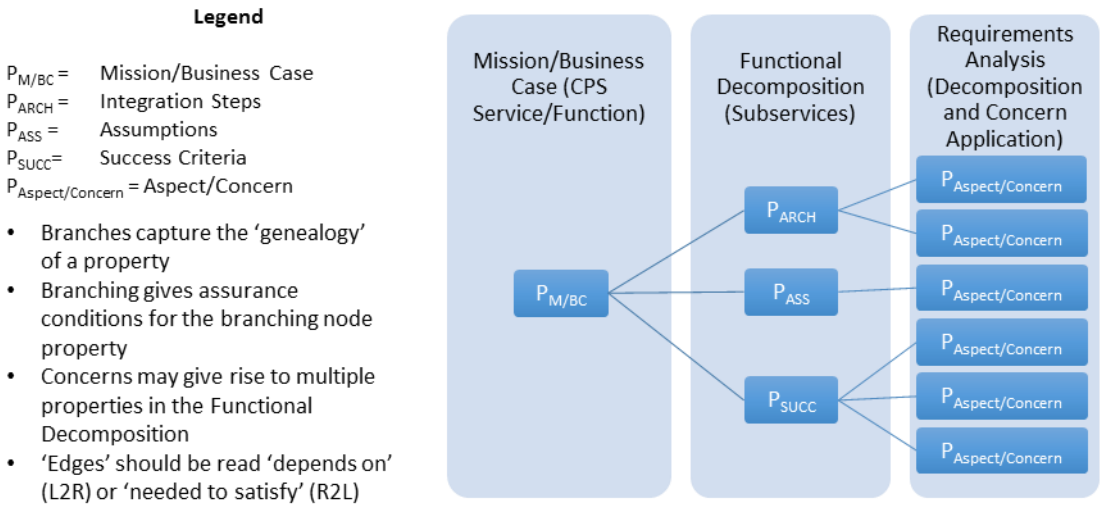


Figure 13: The Property Tree of a CPS

There are two types of assurance arguments, structural and empirical. Which one is applied depends on the types or sources of properties to be assured.

The 'branching type' assurance argument (1) itself has a couple of different flavors, one for assurance of a logically compound property (containing propositional connectives) and one for properties that are compound due to the componentry of the CPS and its interactions.

The 'leaf type' assurance argument (2) is one that relates to the design D and test T put in place in order to achieve a property of the CPS.

(1) $H_{Branching}$

(2) H_{Leaf}

The argument, "A", in this case may take the form that the test itself, the setup for the test, and the way in which test results are stored and managed is in compliance with a standard, and the argument would make reference to the standard, as an example.

(1) Structural (logical and architectural) for branching properties, P and Q:

$A(P*Q) =_{Def} H_{Branching} (A(P), A(Q))$,

for logically compound or architecture properties

(2) Empirical for the terminating properties or 'leaves' of the tree:

$A(P, D, T) =_{Def} H_{Leaf} (P, D, T)$

Each leaf represents the argumentation that "the design D and test T are sufficient to conclude that the property P is met." The argumentation H_{Leaf} makes reference to a certification, standard, or regulation where the test T is recommended or required to establish that property as well as provide an estimate of level of confidence.

The *assurance case* of a CPS consists of all of the assurance judgments for every property in the CPS Model.

2.2.4 Concerns to Aspects

The concerns are identified and further analyzed, producing a set of cross-cutting concern groupings called *aspects*. These aspects are "factored" from the work of the various working groups that produced this Framework – namely, the Vocabulary and Reference Architecture, Use Case, Cybersecurity and Privacy, Timing and Synchronization, and Data Interoperability subgroups.

Concerns and *aspects* are not orthogonal. That is, within the analysis of a given concern, consideration must also be given to related concerns. For example, in considering the trustworthiness aspect, the trustworthiness of timing should be considered.

2.2.5 Activities and Artifacts

In using the Framework to analyze and document CPS, a series of *activities* is performed. For example, a typical waterfall process includes use case development, functional decomposition, requirements analysis, design, etc.

These are generic activities and are identified for each facet. These activities can be considered activity groups which may be tailored during analyses of the aspects and concerns. For example, a Conceptualization facet activity "Requirements Analysis" may include a Trustworthiness requirements analysis, a Timing requirements analysis, etc.… In this Framework, "activities" may refer to individual activities or activity groups.

Each activity produces one or more *artifacts*, which are the concrete technical components used to document the results.

2.3 Uses of the CPS Framework

As described in 2.2.2, the CPS Framework consists of *aspects* and *facets*. The *aspects* are categories of concerns. Each aspect represents a set of similar concerns and this is reflected in the name of the aspect. For example, the trustworthiness aspect includes concerns of security, privacy, safety, reliability and resilience. A list of the CPS Framework aspects and the concerns that belong to them is in Section 2.4.3. The CPS Framework facets are described in Section 2.4.2.

Having clarity about the elements of each facet, the activities/artifacts lists, and how they interrelate is critically important to understanding the approach of this document. It is important to understand how the activities and their artifacts address concerns and aspects in all three facets.

Facet activities and artifacts are at the outset very general and relate to a generic high-level process needed to understand CPS conceptualization, realization, and assurance. Hence these activities are in essence a template and need to be specialized to a CPS domain. The specialization of facet activities to a CPS domain involves the following:

- **Defining which of the CPS aspects apply to that domain:** People are often subject matter experts about a certain concern as it relates to a certain domain. Over time, they have built a consensus that a specific set of processes and tools must be applied in a specific way to adequately *address the concern.*
- **Updating the facet activities and artifacts for each applicable aspect:** this should be based on a review of the best practices for addressing each concern in the aspect.

The result of specializing facet activities to a CPS domain, and the attendant concerns, is a set of activities and artifacts that address the concerns that apply to that domain. For example, if the CPS performs or delivers safety-critical functions in the transportation domain, then there are multiple safety processes and test regimens that have become standards. For example, in the area of transportation that has to do with ground vehicles, the ISO/IEC 26262 document [23], has become a standard for shaping the approach of the commercial ground vehicle industry to establishing the software system safety of the vehicle systems.

Thus, the Framework can be used in different processes, depths, and scopes:

Processes:

- Waterfall (analyze conceptualization, then realization, then assurance.) This traditional system engineering flow allows for a requirements-driven process that leads to assured and verified function. Note that although this indicates a linear sequence through the facets, the ability to iterate and propagate changes discovered in one facet to the others is typically observed.
- Reverse engineering (analyze realization, then conceptualization, then assurance.) To

understand a deployed CPS and perhaps to extend or enhance it, reverse engineering analyzes the realized CPS for its properties and observes its documentation to determine assurances. Once it is analyzed, modifications and enhancements can be made starting at any facet.

- Agile (do some conceptualization, then realization, then assurance, then iterate to greater depths of detail.) Analysis alone sometimes results in a reality other than what was originally envisioned at a high level, so an agile process seeks to take a minimal or "core" conceptualization to rapid realization and assurance. Once confirming initial assumptions about the CPS, the agile development process fills in additional detail in each facet to iteratively arrive at the completed set of artifacts.
- Service-based (analyze conceptualization, identify/fit advertised realizations, then assurance.) Dynamic services can be envisioned and deployed on top of existing CPS.
- Gap-analysis (analyze a set of CPS including systems of CPS and compare to discover gaps and overlaps for Pivotal Points of Interoperability (PPI)). Understanding the opportunities for integration or gap-filling informs holistic tradeoff decisions about integrating systems and capabilities.

Depths:

- Critical tightly-coupled CPS: For critical infrastructures such as the energy grid, a deep and detailed process would be developed using the Framework. Emphasis on hard requirements and assurances, along with constraints from most aspects, would be evidenced.
- Loosely-coupled CPS: Especially appropriate for applications of CPS that repurpose capabilities of existing CPS and integrate them in new and novel ways, a lighter emphasis on hard requirements and a greater weight on functional goals are sought.
- Shallow analysis: For presenting concepts or talking about alternative approaches to CPS problems, small subsets of the Framework might be used. The use of the Framework structure and terminology allows the substance of the concept to be readily understood because the Framework sets a context for the discussion.

Scopes:

- Single CPS device: A device such as a video camera, robot, or thermostat. The focus of the analysis would emphasize the robustness of the design to enable it to become a valued component of a CPS.
- System or subsystem: A system of individual cyber, physical, and cyber-physical devices such as an HVAC system, which might consist of thermostat, air handler, compressor, and furnace.
- SoS: A system of interconnected systems, such as a power company demand response program interacting with individual HVAC systems to achieve a balanced energy system.

2.4 The Description of the CPS Framework

This section presents a detailed description of the CPS Framework. The Framework provides a taxonomy and organization of analysis that allow the complex process of studying, designing, and evolving CPS to be orderly and sufficiently encompassing.

A visual representation of the Framework was previously shown in Figure 4 in terms of domains, facets, and aspects.

The rest of this section presents the elements of the Framework in tabular form, providing only the taxonomy.

2.4.1 Domains

The domains of CPS are the areas of deployment of CPS in which stakeholders may have domain-specific and cross-domain concerns. Table 1 provides the initial listing.[11]

Table 1: Domains of CPS

Domains	
Advertising	Entertainment/sports
Aerospace	Environmental monitoring
Agriculture	Financial services
Buildings	Healthcare
Cities	Infrastructure (communications, power, water)
Communities	Leisure
Consumer	Manufacturing
Defense	Science
Disaster resilience	Social networks
Education	Supply chain/retail
Emergency response	Transportation
Energy	Weather
… perhaps others.	

[11] This list is expected to expand.

2.4.2 Facets

Table 2 lists and defines the facets.

Table 2: Facets

Facet	Description
Conceptualization	What things should be and what things are supposed to do: the set of activities that produce a model of a CPS (includes functional decomposition, requirements, and logical models.)
Realization	How things should be made and operate: the set of activities that produce, deploy, and operate a CPS (includes engineering tradeoffs and detailed designs in the critical path to the creation of a CPS instance.)
Assurance	How to achieve a desired level of confidence that things will work the way they should: the set of activities that provide confidence that a CPS performs as specified (includes claims, evidence, and argumentation.)

2.4.3 Aspects and Concerns

Table 3 lists and defines the aspects.

Table 3: Aspects

Aspect	Description
Functional	Concerns about function including sensing, actuation, control, communications, physicality, etc.
Business	Concerns about enterprise, time to market, environment, regulation, cost, etc.
Human	Concerns about human interaction with and as part of a CPS.
Trustworthiness	Concerns about trustworthiness of CPS including security, privacy, safety, reliability, and resilience.
Timing	Concerns about time and frequency in CPS, including the generation and transport of time and frequency signals, timestamping, managing latency, timing composability, etc.
Data	Concerns about data interoperability including fusion, metadata, type, identity, etc.
Boundaries	Concerns related to demarcations of topological, functional, organizational, or other forms of interactions.

Aspect	Description
Composition	Concerns related to the ability to compute selected properties of a component assembly from the properties of its components. Compositionality requires components that are composable: they do not change their properties in an assembly. Timing composability is particularly difficult.
Lifecycle	Concerns about the lifecycle of CPS including its components.

Table 4 lists and defines the concerns.

Table 4: Concerns

Aspect	Concern	Description
Functional	**actuation**	Concerns related to the ability of the CPS to effect change in the physical world.
Functional	**communication**	Concerns related to the exchange of information internal to the CPS and between the CPS and other entities.
Functional	**controllability**	Ability of a CPS to control a property of a physical thing. There are many challenges to implementing control systems with CPS including the non-determinism of cyber systems, the uncertainty of location, time and observations or actions, their reliability and security, and complexity. Concerns related to the ability to modify a CPS or its function, if necessary.
Functional	**functionality**	Concerns related to the function that a CPS provides.
Functional	**manageability**	Concerns related to the management of CPS function. For example, Managing Timing in complex CPS or SoS is a new issue with CPS that did not exist before. It is being developed with new standards
Functional	**measurability**	Concerns related to the ability to measure the characteristics of the CPS.
Functional	**monitorability**	Concerns related to the ease and reliability with which authorized entities can gain and maintain awareness of the state of a CPS and its operations. Includes logging and audit functionality.
Functional	**performance**	Concerns related to the ability of a CPS to meet required operational targets.
Functional	**physical**	Concerns about purely physical properties of CPS including seals, locks, safety, and EMI.

Aspect	Concern	Description
Functional	**physical context**	Concerns relating to the need to understand a specific observation or a desired action relative to its physical position (and uncertainty.) While this information is often implied and not explicit in traditional physical systems, the distributed, mobile nature of CPS makes this a critical concern.
Functional	**sensing**	Concerns related to the ability of a CPS to develop the situational awareness required to perform its function.
Functional	**states**	Concerns related to the states of a CPS. For example, the functional state of a CPS is frequently used to allow for variation in the CPS response to the same set of inputs. Variation in response based on state is sometimes referred to as functional modes.
Functional	**uncertainty**	Managing the effects of uncertainties is a fundamental challenge in CPS. Sources of uncertainty in CPS can be grouped into statistical (aleatoric), lack of knowledge (epistemic) uncertainty, or systematic uncertainty. In CPS, statistical uncertainty is caused by randomness of accuracy of sensing and actuation, often caused by uncertainty of manufacturing processes. Systematic uncertainty is caused by incomplete knowledge either due to limits of acquired knowledge or due to simplification in modeling. Typical manifestations of epistemic uncertainty are limited validity of models of physical processes or limits of computability of properties of mathematical models.
Business	**enterprise**	Concerns related to the economic aspects of CPS throughout their lifecycle.
Business	**cost**	Concerns related to the direct and indirect investment or monetary flow or other resources required by the CPS throughout its lifecycle.
Business	**environment**	Concerns related to the impacts of the engineering and operation of a CPS on the physical world.
Business	**policy**	Concerns related to the impacts of treaties, statutes, and doctrines on a CPS throughout its lifecycle.
Business	**quality**	Concerns related to the ease and reliability of assessing whether a CPS meets stakeholder (especially customer) expectations.

Aspect	Concern	Description
Business	**regulatory**	Concerns related to regulatory requirements and certifications.
Business	**time to market**	Concerns related to the time period required to bring a CPS from need realization through deployment.
Business	**utility**	Concerns related to the ability of a CPS to provide benefit or satisfaction through its operation. Utility reflects a business concern, especially when considered as the numerator when computing value, which equals utility divided by costs.
Human	**human factors**	Concern about the characteristics of CPS with respect to how they are used by humans.
Human	**usability**	Concerns related to the ability of CPS to be used to achieve its functional objectives effectively, efficiently, and to the satisfaction of users (adapted from ISO 9241-210.) The combination of physical and cyber into complex systems creates challenges in meeting usability goals. Complexity is a major issue. The diversity of interfaces creates a significant learning curve for human interaction.
Trustworthiness	**privacy**	Concerns related to the ability of the CPS to prevent entities (people, machines) from gaining access to data stored in, created by, or transiting a CPS or its components such that individuals or groups cannot seclude themselves or information about themselves from others. Privacy is a condition that results from the establishment and maintenance of a collection of methods to support the mitigation of risks to individuals arising from the processing of their personal information within or among systems or through the manipulation of physical environments.
Trustworthiness	**reliability**	Concerns related to the ability of the CPS to deliver stable and predictable performance in expected conditions.
Trustworthiness	**resilience**	Concerns related to the ability of the CPS to withstand instability, unexpected conditions, and gracefully return to predictable, but possibly degraded, performance.
Trustworthiness	**safety**	Concerns related to the ability of the CPS to ensure the absence of catastrophic consequences on the life, health, property, or data of CPS stakeholders and the physical environment.

Aspect	Concern	Description
Trustworthiness	**security**	Concerns related to the ability of the CPS to ensure that all of its processes, mechanisms, both physical and cyber, and services are afforded internal or external protection from unintended and unauthorized access, change, damage, destruction, or use. Confidentiality: Preserving authorized restrictions on access and disclosure. Integrity: Guarding against improper modification or destruction of system, and includes ensuring non-repudiation and authenticity Availability: Ensuring timely and reliable access to and use of a system.
Timing	**logical time**	Concerns related to the order in which things happen (causal order relation) or event driven.
Timing	**synchronization**	Concerns for synchronization are that all associated nodes have timing signals traceable to the same time scale with accuracies as required. There are three kinds of synchronization that might be required: time, phase, and frequency synchronization, although frequency synchronization is also called syntonization.
Timing	**time awareness**	Concerns that allow time correctness by design. The presence or absence of time explicitly in the models used to describe, analyze, and design CPS and in the actual operation of the components. This is a life-cycle concern as well as a concern for the ability to build devices without the need for extensive calibration of the timing properties.
Timing	**time-interval and latency**	Specifying requirements for timing generally involves requirements for time-intervals between pairs of events. A time-interval is the duration between two instants read on the same timescale. CPS timing requirements are generally expressed as constraints on the time intervals (TI) between pairs of system significant events. These can be categorized in terms of bounded TIs or latency, deterministic TIs, and accurate TIs.
Data	**data semantics**	Concerns related to the agreed and shared meaning(s) of data held within, generated by, and transiting a system.
Data	**identity**	Concerns related to the ability to accurately recognize entities (people, machines, and data) when interacting with or being leveraged by a CPS.

Aspect	Concern	Description
Data	**operations on data**	Concerns related to the ability to create/read/update/delete system data and how the integrity of CPS data and behaviors may be affected.
Data	**relationship between data**	Concerns related to how and why sets of data must, may, or may not be associated with each other and the value or harm that can be derived from those associations.
Data	**data velocity**	Concerns related to the speed with which data operations are executed.
Data	**data volume**	Concerns related to the volume or quantity of data associated with a CPS' operation.
Boundaries	**behavioral**	Concerns related to interdependence among behavioral domains. Concerns related to the ability to successfully operate a CPS in multiple application areas.
Boundaries	**networkability**	Concerns related to the ease and reliability with which a CPS can be incorporated within a (new or existing) network of other systems.
Boundaries	**responsibility**	Concerns related to the ability to identify the entity or entities authorized to control the operation of a CPS.
Composition	**adaptability**	Concerns related to the ability of the CPS to achieve an intended purpose in the face of changing external conditions such as the need to upgrade or otherwise reconfigure a CPS to meet new conditions, needs, or objectives.
Composition	**complexity**	Concerns related to our understanding of the behavior of CPS due to the richness and heterogeneity of interactions among its components, such as existence of legacy components and the variety of interfaces.
Composition	**constructivity**	Concerns related to the ability to combine CPS modular components (hardware, software, and data) to satisfy user requirements.
Composition	**discoverability**	Concerns related to the ease and reliability with which a CPS component can be observed and understood (for purposes of leveraging the component's functionality) by an entity (human, machines). Concerns related to the ease and reliability with which a CPS component's functions can be ascertained (for purposes of leveraging that functionality) by an entity (human, machines).

Aspect	Concern	Description
Lifecycle	**deployability**	Concerns related to the ease and reliability with which a CPS can be brought into productive use.
Lifecycle	**disposability**	Concerns related to the impacts that may occur when the CPS is taken physically out of service.
Lifecycle	**engineerability**	Concerns related to the ease and reliability with which a CPS design concept can successfully be realized via a structured engineering process.
Lifecycle	**maintainability**	Concerns related to the ease and reliability with which the CPS can be kept in working order.
Lifecycle	**operability**	Concerns related to the operation of the CPS when deployed.
Lifecycle	**procureability**	Concerns related to the ease and reliability with which a CPS can be obtained.
Lifecycle	**producibility**	Concerns related to the ease and reliability with which a CPS design can be successfully manufactured.

2.4.4 Composition of Concerns

Concerns are applied in the CPS Framework in general to all of the activities of all of the facets. This is one sense in which they are potentially 'cross-cutting'. For a particular CPS one may decide to apply certain of the concerns and may view others as not being relevant. This is one of the ways that the CPS Framework can be tailored to the development of a CPS. At the same time, once the set of relevant concerns has been determined, the application of a concern must take into account its interactions with other relevant concerns. For example, action taken in a design to address the cyber-security concern may adversely affect the safety of the CPS. Corrective action then taken to bolster its safety may then reduce the effectiveness of the actions for cyber-security, resilience or reliability. In other words, there will be trade-offs between concerns.

Thus one needs to explain how a set of more than one concern, deemed as relevant to a CPS, is applied to the CPS in question. This is referred to as the *composition of concerns* and explains how it is to be understood and used.

The effect of applying a concern to a CPS depends on the facet and activity being considered. Generally, that application can be associated with the *set of properties or requirements* that the concern requires of the CPS. Hence applying two or more concerns amounts to requiring *all of the properties required by the set of concerns*.

If one denotes formally the set of properties of a CPS required by a concern as:

$$\bar{C}^{CPS} = \{properties\ of\ the\ CPS\ required\ by\ the\ concern\ C\}$$

then the composition of concerns C_1 and C_2 can be expressed as follows:

$$\overline{C_1 * C_2}^{CPS} = \overline{C_1}^{CPS} \cup \overline{C_2}^{CPS}.$$

The interpretation of the composition of multiple concerns is defined in terms of binary composition. The composition of a set of concerns is interpreted as the *union of the properties required by each concern in the set*. This notion of composition is clearly *commutative* and *associative*.

This *set-theoretic semantics* of the CPS Framework can be extended to all of the concepts of the framework and will be worked out in detail in future works.

An example of composition of concerns is *timing security*. The composition of *timing* and *security* results in the collection of all the properties of CPS that are required by the timing and the security concerns. Resolving 'conflicts' between properties is one of the tasks of *requirements analysis*.

Consider since timing requires both a physical signal and data about that signal, timing security includes the security of the data in much the same way as traditional cyber-security, plus the security of the physical signal. Many CPS will require timing reliability, both for the local system and for the traceability of the timing. For example, GPS jamming is the timing equivalent of a cyber denial-of-service attack. Resilience will appear as fault-tolerance in timing, whether the fault is intentional or unintentional. Timing safety will depend on the CPS. Certainly in some systems a timing failure can lead to a lack of safety. Privacy will not be a timing concern in many systems, because timing is generally intended to be public information. However, there may well be cases where the timing requires privacy.

So assume that it is desired to present the concerns that apply to the exchange of GNSS ("GPS") timing. You would simultaneously have to satisfy concerns of the form:

- Trustworthiness.Reliability – "message delivery shall be reliable"
- Trustworthiness.Security – "availability shall not be interfered with through Denial Of Service"
- Trustworthiness.Resilience - "the system shall be fault-tolerant"
- Trustworthiness.Safety – "message exchange failure shall not lead to hazard or harm"
- Trustworthiness.Confidentiality – "message exchange shall only be understood by the intended recipient"
- Trustworthiness.Privacy – "message shall not contain PII"
- Data.DataSemantics – "shall have a representation of time"

- Trustworthiness.Security.Cybersecurity – “message exchange must not be tampered with”

This expresses the above outlined example as a set of concerns which taken together corresponds to the union of the properties given by each concern.

2.4.5 Activities and Artifacts

Table 5 lists the activities (groups) and artifacts related to the conceptualization facet.

Table 5: Conceptualization Facet: Activities and Artifacts

Activity and Artifacts
Mission and Business Case Development **Artifact: Business use cases**
Functional Decomposition **Artifact: Detailed use cases, actors, information exchanges**
Requirements Analysis **Artifact: Functional and non-functional requirements**
Requirements Allocation **Artifact: HW/SW configuration Items**
Interface Requirements Analysis **Artifact: Interface requirements**

Table 6 lists the activities (groups) and artifacts related to the realization facet.

Table 6: Realization Facet: Activities and Artifacts

Activity and Artifacts
Business Case Analysis **Artifact: Trade studies, lifecycle cost analysis, return on investment, and interdependencies with requirements, regulations, and incentives**
Lifecycle Management **Artifact: Lifecycle management and sustainability plan, integrated lifecycle management monitoring**
Design **Artifact: Design documentation, tradeoff analyses, requirement verification, virtual prototypes**
Manufacturing/Implementation **Artifact: Manufactured, integrated products, testing plans, and test results**
Operations **Artifact: Performance, quality, and product evolution tracking**
Disposal **Artifact: Reuse, sustainability and energy recovery assessments, disposal manifests**
Cyber-Physical Abstraction Layer Formation **Artifact: Domain (and product)-specific ontologies, modeling languages, and semantics specifications used in all phases of the lifecycle**
Physical Layer Realization **Artifact: Physical substrates of the CPS used in all phases of the lifecycle.**

Table 7 lists the activities (groups) and artifacts related to the assurance facet.

Table 7: Assurance Facet: Activities and Artifacts

Activity and Artifacts
Identify Assurance Objectives **Artifact: Assurance objectives/analysis report**
Define Assurance Strategy **Artifact: Strategy document/plan**
Control Assurance Evidence **Artifact: Control documentation**
Analyze Evidence **Artifact: Analysis report**
Provide Assurance Argument **Artifact: Assurance argument report**
Provide Estimate of Confidence **Artifact: Confidence estimate**
Configuration Audit **Artifacts: Product configuration assessment**
Requirements Verification **Artifact: Requirements and test results assessment**

Activity and Artifacts
Product Certification and Regulatory Compliance Testing **Artifact: Certifications**

2.5 Related Standards and Activities

The purpose of this section is to highlight some, though far from all, related standards, organizations and working groups that are relevant to the NIST CPS PWG effort.

From 2010 to 2013, the European Lighthouse Integrated Project "Internet of Things – Architecture" (IoT-A) developed and proposed an architectural reference model for the IoT, referred to as the IoT Architectural Reference Model (IoT ARM) [1]. The goal of the project was to introduce a common language for fostering the interoperability between vertical "silos" (domains) in emerging IoT applications. The IoT ARM introduces top-down architectural principles and design guidelines.

IoT-A explicitly separates itself in scope from CPS. The IoT-ARM's functional view is organized in service layers (including communication, services, management, and security) on top of CPS. CPS, in IoT-A's terminology, are IoT devices (devices) and IoT resources (software), and their architecting guidelines are not covered by the IoT ARM. It is important for the NIST CPS PWG Vocabulary and Reference Architecture subgroup to determine possible interactions with the IoT ARM.

The IEEE P2413 working group [4] was formed in 2014 to promote cross-domain interaction, aid system interoperability, and provide functional compatibility in the IoT. The IEEE P2413 also defines an architectural framework for the IoT, including abstractions and a common vocabulary. It emphasizes a "blueprint for data abstraction and the quality quadruple (protection, security, privacy, and safety.)"

The IoT ARM and IEEE P2413 share a few important characteristics that are worth noting. Both initiatives adhere to the ISO/IEC/IEEE 42010 standard, their functional models are inspired by the OSI reference model, and they explicitly take into consideration architecture divergence. Also, both identify architecture divergence as a major topic. It is important for the NIST CPS PWG to find similarities and key differences between the scopes of IoT-related activities and CPS. This will help readers of this document to distinguish between CPS and IoT and use the NIST CPS Reference Architecture to define CPS-specific architectures that may be compatible with IoT services and standards.

oneM2M [11] is intended to be an interoperability enabler for the entire CPS, M2M and IoT Ecosystem. The purpose and goal of oneM2M is to develop Technical Specifications and Technical Reports, which address the need for a common IoT Service Layer that can be readily realized through an API embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with IoT application servers worldwide. A critical

objective of oneM2M is to enable users to build platforms, regardless of existing sector or industry solutions, to enable wider integration and cross-system value to be derived than is currently possible. oneM2M aims to attract and actively involve a wide variety of organizations from IoT-related business domains such as: telematics and intelligent transportation, healthcare, utilities, industrial automation, smart homes, etc.

Cybersecurity Research Alliance (CSRA) [55] is an industry-led, non-profit consortium focused on research and development strategy to address evolving cybersecurity environment through partnerships between government, industry, and academia. This effort was established in response to the growing need for increased public-private collaboration to address R&D issues in cybersecurity.

CPS Voluntary Organization (supported by the National Science Foundation) [56] is an online site to foster collaboration among CPS professionals in academia, government, and industry.

The Networking and Information Technology Research and Development (NITRD) CPS Senior Steering Committee [57] coordinates programs, budgets, and policy recommendations for CPS research and development (R&D). This includes identifying and integrating requirements, conducting joint program planning, and developing joint strategies for the CPS R&D programs conducted by agency members of the NITRD Subcommittee. CPS includes fundamental research, applied R&D, technology development and engineering, demonstrations, testing and evaluation, technology transfer, and education and training; and "agencies" refers to Federal departments, agencies, directorates, foundations, institutes, and other organizational entities.

NIST Privacy Engineering [58] focuses on providing guidance that can be used to decrease privacy risks, and to enable organizations to make purposeful decisions about resource allocation and effective implementation of controls in information systems. This NIST privacy engineering work targets specifically how government agencies are to address privacy and may not be adequate for the private sector.

The goal of making time an integral part of networks is being advanced in foundational standards that define both wired and wireless networks. IEEE 802.1 [59][59] [12, 13] has a time

[12]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards/ieee.org/).

[13] The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

sensitive networking (TSN) working group defining various standards that will enable determinism in local area wired networks using synchronized clocks. The method of synchronizing clocks is based on the IEEE 1588 standards that have invented the Precision Time Protocol (PTP) [211]. The Internet Engineering Task Force (IETF) is planning to leverage the building blocks defined by IEEE 802.1 and IEEE 1588 to enable determinism in wide area networks (routable wired networks). Similar initiatives in the IEEE 802.11 [60] standards body have resulted in the development of Timing Measurement and Fine Timing Measurement protocols that enable precise clock synchronization in WiFi networks. International Telecommunications Union (see ITU-T standard G.8265.1-July 2014[14]) has also leveraged the work done in 1588 and applied it to telecommunication networks. Enhancements to the IEEE 802.15.4 standards have resulted in the development of a time-slotted communication model for low power personal area networks. This work has been created by the IETF task group called 6TISCH, RFC 7554 [61].

AVnu Alliance [62] is a community for creating an interoperable ecosystem servicing precise timing and low latency requirements of diverse applications using open standards like Time-Sensitive Networking (TSN). This alliance focuses on creating interoperability tests and certification for products used in applications requiring bounded latency, reserved bandwidth, and synchronized time.

Industrial Internet Consortium (IIC) [63] brings together the organizations and technologies necessary to accelerate growth of the Industrial Internet by identifying, assembling, and promoting best practices. This goal of the IIC is to drive innovation through the creation of new industry use cases and testbeds for real-world applications; define and develop the reference architecture and frameworks necessary for interoperability; influence the global development standards process for Internet and industrial systems; facilitate open forums to share and exchange real-world ideas, practices, lessons, and insights; and build confidence around new and innovative approaches to security.

National Security Telecommunications Advisory Committee (NSTAC) [64] brings together up to 30 industry chief executives from major telecommunications companies, network service providers, information technology, finance, and aerospace companies. These industry leaders provide the President with collaborative advice and expertise, as well as robust reviews and recommendations. The NSTAC's goal is to develop recommendations to the President to assure vital telecommunications links through any event or crisis, and to help the US Government maintain a reliable, secure, and resilient national communications posture.

[14] ITU-R publications are available from the International Telecommunications Union, Place des Nations, 1211 Geneva 20, Switzerland (http://www.itu.in/).

The European Telecommunications Standards Institute's (ETSI's) standardization group dedicated to Low Throughput Networks (LTN) [52] technology has released the first three specifications of an Internet of Things (IoT) network dedicated to low throughput communications. These new requirements provide a breakthrough in the machine to machine business, allowing object connection for a few euros per year, with a few milliwatts for transmission and a modem costing less than 1 euro. The key to the success of IoT standardization and implementation, these assumptions are the basis for many new and innovative applications. Low Throughput Network (LTN) technology is a wide area bidirectional wireless network with key differentiators compared to existing networks. It enables long-range data transmission (distances around 40 km in open field) and/or communication with buried underground equipment and operates with minimal power consumption allowing several years of operation even with standard batteries. This technology also implements advanced signal processing that provides effective protection against interference.

The Internet of Things (IoT) is one of the new, convergent technologies addressed by Open Platform 3.0™ [53]. The Open Group IoT standards aim to do for the IoT what HTML/HTTP did for the Web, enabling everything to be connected on the fly. Vendors will be able to collect information from products in the field throughout their lifecycle. This will allow the optimization of maintenance operations, providing increased safety at lower cost. Enterprises will be able to monitor and control installed equipment, and integrate it into intelligent solutions, for example, to ensure the health of buildings and machinery, or to improve energy efficiency.

2.6 Summary

The CPS Framework presents a set of high-level concepts, their relationships, and a vocabulary for clear communication among stakeholders (e.g., architects, engineers, users). The ultimate goal of the CPS Framework is to provide a common language for describing interoperable CPS architectures in various domains so that these CPS can interoperate within and across domains and form systems of systems.

The CPS Framework includes the identification of foundational goals, characteristics, common roles, and features across CPS domains, while considering cybersecurity, privacy, and other cross-cutting concerns. The CPS Framework is an abstract framework, or meta-model, for understanding and deriving application domain-specific CPS architectures. Work remains to be done to further specify this high-level architecture independent from specific application domains, problems, standards, technologies, protocols, and implementations, and to identify interfaces to facilitate cross-sector CPS interoperability.

The CPS Framework consists of three facets – conceptualization, realization, and assurance. Each facet is presented and understood from its set of activities and artifacts. The activities in turn address aspects and concerns throughout the CPS development cycle.

The artifacts consist of properties discovered and modeled in the conceptualization facet, implemented and deployed in the realization facet, and verified and validated in the assurance facet.

Appendix A. Facets of the CPS Framework

This section defines and describes the three facets of the CPS Framework: conceptualization, realization, and assurance.

A.1 Conceptualization Facet

This section examines the conceptualization facet. It comprises the following sections:

- Section A.1.1 provides an overview of the conceptualization facet.
- Section A.1.2 discusses the conceptualization facet's activities and artifacts.
- Section A.1.3 explains mission or business analysis.
- Section A.1.4 explains requirement analysis.
- Section A.1.5 explains functional analysis.
- Section A.1.6 provides a conceptual functional view of SoS.
- Section A.1.7 gives a logical functional decomposition of the CPS.

A.1.1 Overview

The conceptualization facet is responsible for defining the stakeholders and user-oriented view of the CPS desired capabilities, and for providing the (aspirational) model of the CPS. It includes activities that address (1) the problems (gaps or deficiencies) or the opportunity with respect to the organization's business, mission, vision, strategic goals, and objectives; (2) the needs and requirements of the major stakeholders (customers, users, administrators, owners, and regulators) who have an interest in the CPS throughout its lifecycle, and (3) the analysis and decomposition into abstract functional elements and their logical rather than their engineering interactions, which are addressed in the realization facet.

A.1.2 Activities / Artifacts

The conceptualization facet is described in terms of activities and artifacts as shown in Table 8.

Table 8: Conceptualization Activities and Artifacts

Activities and Artifacts
Mission and Business Case Development **Artifact: Business use cases**
Functional Decomposition **Artifact: Detailed use cases, actors, information exchanges**
Requirements Analysis **Artifact: Functional and non-functional requirements**
Requirements Allocation **Artifact: HW/SW configuration items**

Activities and Artifacts
Interface Requirements Analysis **Artifact: Interface requirements**

A.1.3 Mission or Business Analysis

The mission and business case development activity identifies the problems (gaps or deficiencies) or the opportunity with respect to the organization's business, mission, vision, strategic goals, and objectives and produces high-level business use cases relevant to the CPS. The output of this analysis is part of the portfolio management decisions of the organization.

A.1.4 Requirements Analysis

The requirements analysis activity identifies the major stakeholders (customers, users, administrators, owners, and regulators) who have an interest in the CPS throughout its lifecycle, then proceeds to assess their needs and analyze their requirements. Outputs of this phase include a prioritized list of needs, capabilities of the CPS, and interactions between the CPS and its users. The analysis includes checks of whether each requirement is necessary, consistent, feasible, traceable, verifiable, and affordable. The requirements are allocated to each building block following the function analysis. Since these requirements address concerns of multiple stakeholders, it becomes necessary to manage them through agreements with the stakeholders.

A.1.5 Functional Analysis

This activity develops the rationale for the CPS functional decomposition and provides the building blocks to functionally derive domain-specific CPS architectures from the CPS Framework. This section highlights the thought process and one outcome of this activity. Clearly, this activity may yield very different outcomes depending on its guiding principles, approaches, objectives, constraints, and context of the intended use case. The goal for this exercise, however, is to arrive at a CPS functional decomposition as the outcome or artifact of this activity, possessing values beyond a goals of this activity. It should be useful in guiding the system conceptualization and design of many CPS systems. It is intended to be adaptable to many industry sectors. For this objective, the generality of the CPS Framework is emphasized, without imposing unnecessary constraints to its wide applicability. At the same time, a balance is struck between its general applicability and the usefulness of the CPS Framework in guiding the design for any specific concrete implementation.

This activity divides the overall system functions into key constituent building blocks (or functional components) and describes the structures in which these building blocks are assembled to form the whole system. It also describes the relationships and interactions between the building blocks to provide the intended system-wide functions.

The functional components are recursively decomposable. As the decomposition progresses, it is expected that the resulting functional decomposition will be specific and consequently less adaptable. It is foreseeable that domain-specific CPS architectures developed using this framework will contain functional structures that meet their specific use case requirements.

This section describes the functional components at an abstract level but does not constrain them to any specific technologies or implementations. Furthermore, it does not make a distinction between whether a cyber function is implemented in hardware or software. This is left to the implementation to make the best choice based on the functional requirements described in this general framework and those drawn from the specific use cases. It does, however, make a distinction between the cyber and physical functions where it is appropriate and to highlight the cyber-physical co-design requirements where it is important from the functional point of view.

Some technical requirements that can be met entirely within the functional space while others cannot. For example, security requires functional components such as those that implement cryptography. It also requires best practice processes, governance, and even regulations on design, development, testing, and certification across the cyber-physical boundary of a system.

Certain capabilities are commonly required in many functional components. To realize these capabilities, it often requires different functional components to act consistently and cohesively as a whole. For example, system security or safety cannot be achieved by functional components in isolation, and any weak link in the system would render the whole system vulnerable. Also, it may not be possible with current methodologies to decompose function and preserve or enable timing specification or realization. Consequently, these capabilities are categorized and described in the aspects material in Section Appendix B.

The goal is to provide a common and accessible framework for dealing with complex CPS in a way that ensures that most of the functional components identified can be implemented as interoperable, composable, and interchangeable building blocks, whether they are products, hardware, software, or services. Leveraging the advantages of the efficiency from specialization and the economy of scale, it should be possible to build large and complex CPS at lower cost by employing proven off-the-shelf system building blocks.

A.1.6 Conceptual Functional View: SoS

This section explores a broad concept that CPS are SoS, which are engineered products with integrated computational and physical capabilities for automatic and, increasingly, autonomous operations, in interaction with physical entities/environment and humans, to produce the desired physical outcomes. At a simpler level, a CPS may be deployed to sense and measure the states and conditions of the physical world for a better understanding of the world and the impacts that people cause. This better understanding would enable better decision-making in the human interest. More often, on the other hand, a CPS may be deployed for changing the

states of the physical entities or environment to bring about physical effects desirable by humans. Figure 14 illustrates the functional composition of CPS as a SoS.

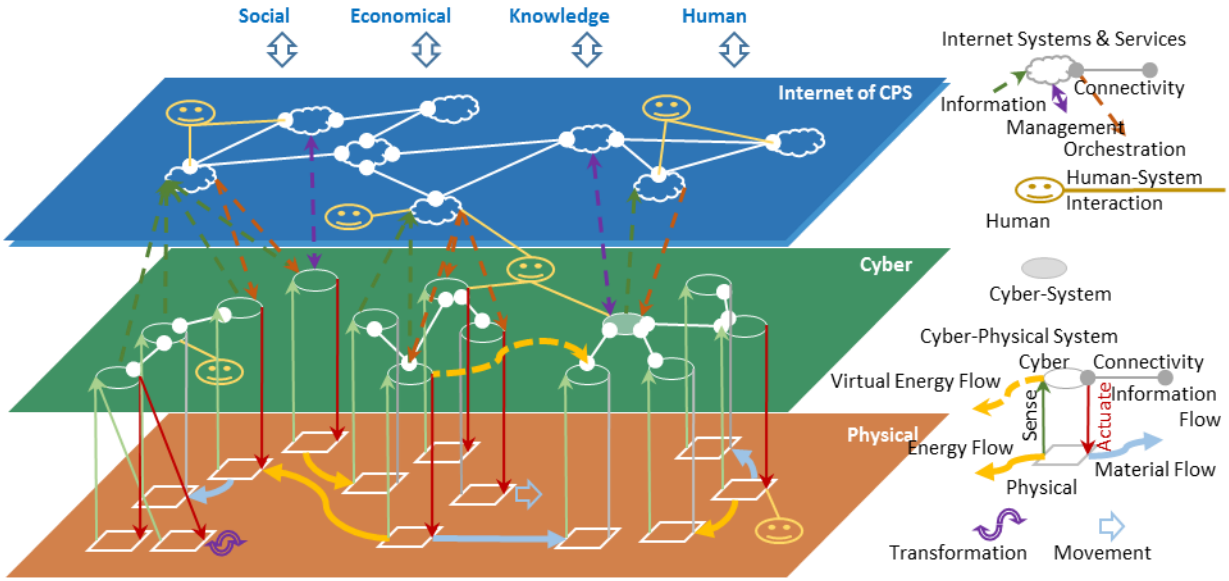


Figure 14: A CPS View: Systems of Systems

At an abstract level, CPS may be deployed to enable and control:

- the flow of energies (e.g., electric grid)
- the flow of materials (e.g., oil pipeline and freight transportation)
- the transformation from materials to objects to goods (e.g., mining, fabrication, chemical refinery and production, manufacturing, farming, generic engineering)
- the movement of objects (e.g., autonomous vehicle, robots, traffic control)
- the flow of signals (e.g., air traffic control)
- the conversion of energies, material, and signals

While some CPS may operate in isolation, many others may be required to operate in concert to produce the desired physical effects at large scale. To enable concerted action, the CPS are connected into clusters of systems. The CPS in such clusters communicate with each other in the cyber space. They may also interact in the physical space. Some of the connectivity may be statically configured while others may be dynamically established.

To orchestrate the operations of the CPS at a global level for a given use case, the clusters of CPS are increasingly brought online with broader systems, predominately the vast computation and communication infrastructure and business processes that have been established in the past decades, forming systems of CPS. This is a defining concept that directly influences the consideration of the scope and structure of this functional decomposition.

With the global technology trends in advanced computing and manufacturing, pervasive sensing, and ubiquitous network connectivity, CPS will likely advance in two major directions:

- CPS are rapidly shifting from programmed automation to an autonomous mode of operations – in other words, becoming more intelligent.
- CPS are increasingly connected horizontally with each other and vertically with the broader systems. The horizontal connectivity paves the way for CPS to collaborate directly. The vertical connectivity brings about the possibility of realizing a global view of the states of the vast network of the CPS and the opportunity to coordinate or orchestrate their operations to achieve optimization at a global level.

These new capabilities in the CPS, fusing with other important evolutions of technologies such as social media, mobile computing, cloud computing, and big data analytics are expected to bring transformational changes to economies, societies, our knowledge of the world, and ultimately the way people live. It is important that the CPS Framework should foresee and accommodate the engagements and interactions between the CPS and these important technological developments.

A.1.7 A Logical Functional Decomposition of the CPS

With the general SoS view of the CPS and their basic characteristics as outlined in the previous section, a CPS functional architecture can be naturally divided into two major domains, the core cyber-physical domain and the SoS domain, as shown in Figure 15.

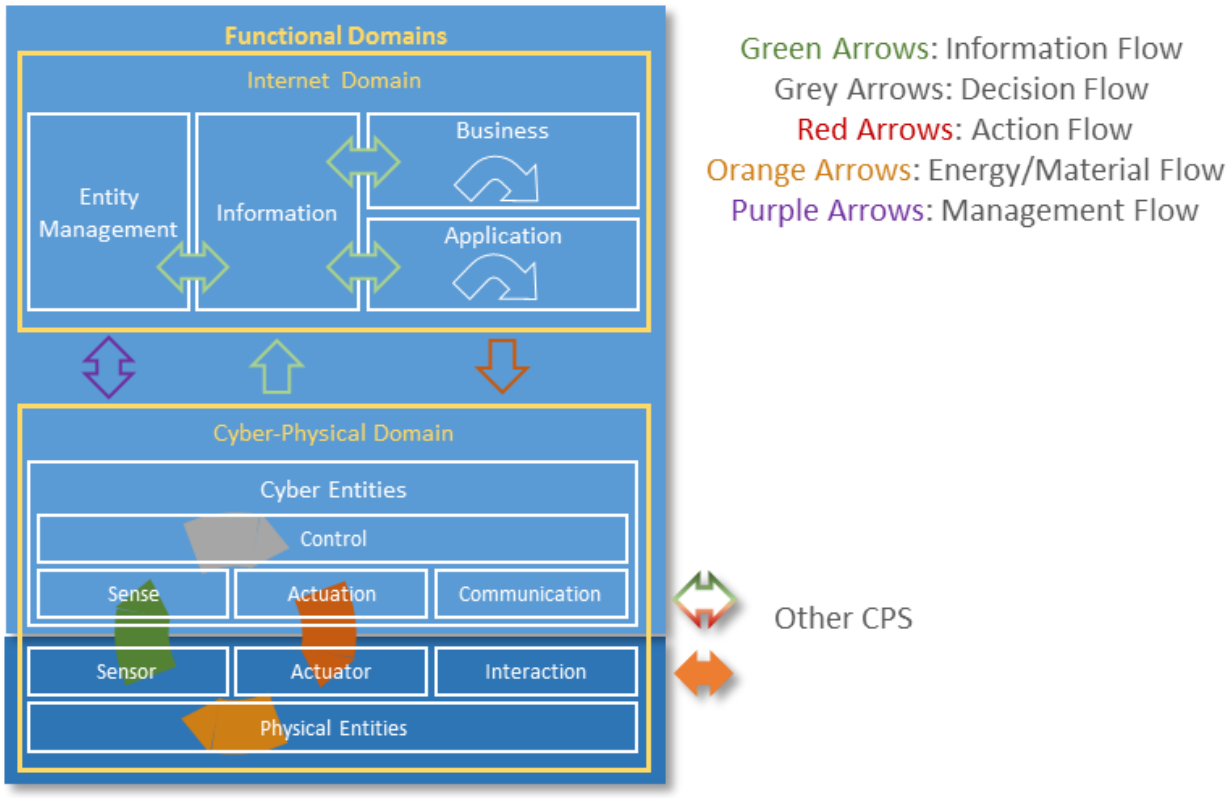


Figure 15: CPS Functional Domains

A.1.7.1 The core cyber-physical domain

The core cyber-physical domain (illustrated by the lower gold rectangle in Figure 15) consists of functional components that contribute to or are involved in the designed functions of the CPS. These functions include sensing the physical condition and state of physical entities, executing control logic, and exercising actuation to produce the desired physical effects. Some CPS may perform only parts of these high-level functions, such as sensing and reporting of the observed physical properties. A complete CPS typically includes all four high-level functions with the combination of the full cycle of sensing, control, actuation, and the physical process forming closed-loop control to produce the desired physical effects.

This domain includes physical entities that carry out functions in the physical world; sensors, actuators, and interactions that mediate between the cyber and physical entities; and cyber entities that exert control on physical entities through sense, actuation, and communication. The sense/actuate control loop is a key feature of CPS.

The CPS may have different levels of sophistication in performing the closed-loop control functions. The control logic may be fully programmed in some systems. In others it may be more flexible and open-ended, allowing intelligent response based on prescribed objectives and

situation awareness. Some systems are merely automatic while others are more autonomous. Some systems may only handle a single input-output stream, but others may be able to synthesize inputs from multiple sources and respond with multiple concerted outputs.

To complete complex tasks, many CPS may connect to and interact with each other, forming a community or a SoS either by configuration or dynamically. The interactions between the CPS can be realized either through logical communication between their respective cyber components or through the physical interaction between their physical counterparts, or both. They can even be relayed across the cyber-physical boundary. Which path of communication or interaction to take is specific to the systems in question and the context in which they are operating; it is in the domain of cyber-physical co-design. The result of co-design should be a coherent model of concerted cyber communications and physical interactions among the CPS to produce the desired physical effects.

In some scenarios, the activities of CPS may be orchestrated by a cyber system that communicates logically with the CPS. These orchestrating cyber systems produce no direct physical effort themselves but are required to maintain the operations of a system of CPS. These orchestration functions depended on by the ongoing operations of the CPS are considered within the core cyber-physical functional domain.

While connectivity is important for many systems of CPS to operate, it is important to note that connectivity should be by design a non-deterministic factor in maintaining the operations of CPS, at least for most of the cases. In the event that connectivity becomes unavailable, the CPS should be able to continue to operate using locally-based programmed logic or autonomous smart control, albeit in a non-optimal or even degraded mode of operations.

A.1.7.2 The SoS domain

Functions in the SoS domain (illustrated by the upper gold rectangle in Figure 15) are responsible for connecting to the CPS, gathering data from these systems, transforming the data into information, and performing analyses on the information to gain insights on a global scale about the operational states of the CPS or the environments that the CPS are monitoring or interacting with. The information can be synthesized with the information from other CPS as well as the information about the environment, business, economy, social, and government for better decision-making. They can also be used to achieve better effectiveness and efficiency in operations by automatically or autonomously orchestrating or coordinating the activities of the CPS at a global scale.

The system of systems domain consists of four major functional components: information, application, business, and entity management.

The information component provides functions for gathering data from the CPS, transforming and persisting them where it is required, and analyzing them to provide information on the

operational states of the CPS, synthesizing information from other sources to inform the business components and to aid the application component in its orchestration or coordination of activities of the CPS.

The application component provides functions that take in information from the information component and process this information based on prescribed objectives, rules, and/or models to orchestrate or coordinate the activities of the CPS to achieve better effectiveness and efficiency in operations. It also interacts with the business component to complete the activities that are required to maintain the optimal operation of the CPS.

The business component provides functions that enable the end-to-end operations of the CPS, including business processes and procedural activities. Examples of these include enterprise resource management (ERM), customer relationship management (CRM), payment systems, order systems, and work planning and scheduling systems.

The entity management component provides manageability functions to the CPS, including identifying, provisioning, configuration, monitoring, updating decommissioning, diagnosis, and more advanced activities such as predictive and prognostic maintenances.

A.2 Realization Facet

This section examines the realization facet. It comprises the following subsections:

- Section A.2.1 provides an overview of the realization facet including the realization facet's activities and artifacts
- Section A.2.2 explores the business layer
- Section A.2.3 explores the lifecycle management layer
- Section A.2.4 explores the design, manufacturing, operations, and disposal layers
- Section A.2.4.1explores the design layer
- Section A.2.4.2 explores the manufacturing/implementation layer
- Section A.2.4.3 explores the operations layer
- Section A.2.4.4 explores the disposal layer
- Section A.2.5 explores the cyber-physical abstraction layers
- Section A.2.6 explores the physical layer

The 'layers' above, and in Figure 16, are used to decompose the activities in the realization facet and are not related to the aspects or concerns with similar names.

A.2.1 Overview

The realization facet is responsible for transforming the outputs of the conceptualization facet into an effective CPS, to enable consistent reproduction of the CPS, to operate and maintain the CPS while providing the required services, and to dispose of the CPS when its services are no longer needed. Its primary output is an instance of the CPS.

CPS are often engineered systems. CPS are differentiated from other types of engineered systems in that they are constructed via the integration of cyber and physical component types and not by the specific functionalities they jointly deliver, the services they provide, or the application domain where they are used. While various definitions create stronger or weaker expectations regarding the characteristics of interactions among cyber and physical components, there is agreement that CPS functionalities are the result of the tight integration of the cyber and physical sides.

The realization facet of the CPS Framework focuses on how CPS are made (see Figure 16). As in other engineered systems, the "make" process can be described using layers typical for engineered systems, such as business, lifecycle, operation, and physical. However, CPS have unique characteristics, specific combinations of concerns, and domain-specific architectures that span a wide range of technology and application domains from Industrial Internet systems to sector-specific product categories to societal scale infrastructures. These areas must be understood and then developed and supported by new foundations, methods, technologies, and standards.

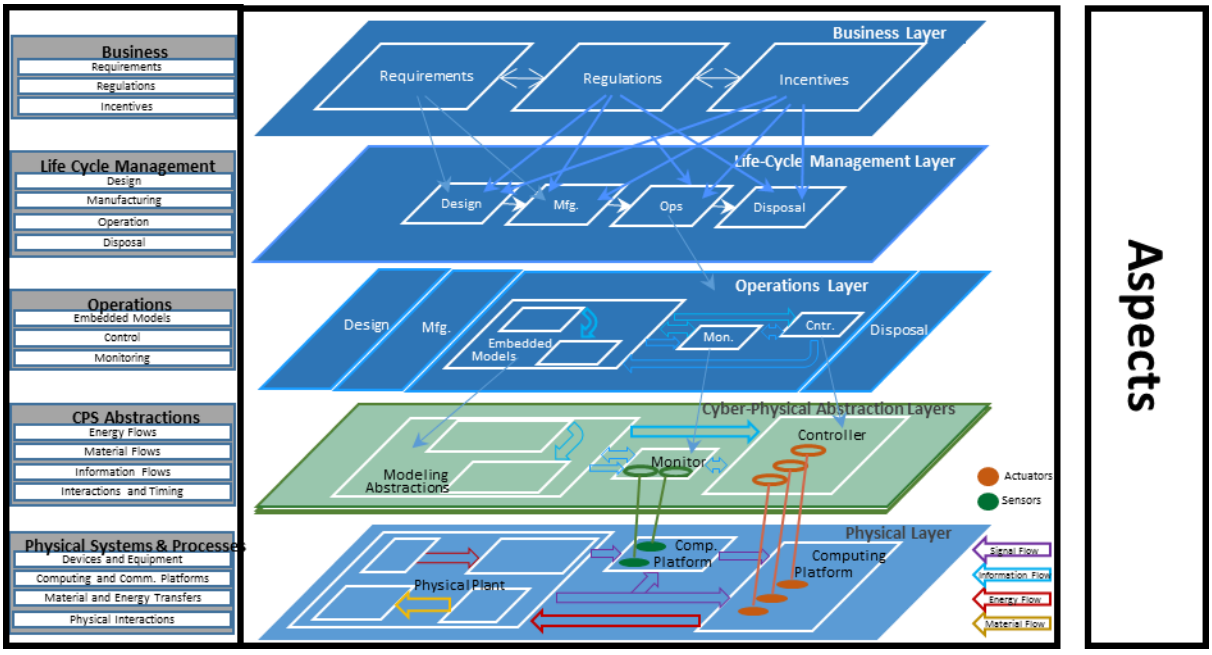


Figure 16: Realization Facet

Figure 16 captures key conceptual layers of the realization facet. Each layer is associated with concepts, components, and notional architectures that can be instantiated into layer- and domain-specific CPS architectures.

Table 9 provides a short summary of activities and artifacts in the realization facet and their relationship to the individual layers.

Table 9: Realization Activities and Artifacts

Activities and Artifacts
Business Case Analysis **Artifact: Trade studies, lifecycle cost analysis, return on investment, and interdependencies with requirements, regulations, and incentives**
Lifecycle Management **Artifact: Lifecycle management and sustainability plan, integrated lifecycle management monitoring**
Design **Artifact: Design documentation, tradeoff analyses, requirement verification, virtual prototypes**
Manufacturing/Implementation **Artifact: Manufactured, integrated products, testing plans, and test results**
Operations **Artifact: Performance, quality, and product evolution tracking**
Disposal **Artifact: Reuse, sustainability and energy recovery assessments, disposal manifests**
Cyber-Physical Abstraction Layer Formation **Artifact: Domain (and product)-specific ontologies, modeling languages, and semantics specifications used in all phases of the lifecycle**
Physical Layer Realization **Artifact: Physical substrates of the CPS used in all phases of the lifecycle**

A.2.2 Business Layer

The role of the business layer is to analyze and formalize business considerations for the realization process of a CPS. Evolution of CPS is driven by societal, business, and individual needs, which are the source of requirements to which business enterprises respond. A unique aspect of CPS is that in many industrial sectors, CPS products are safety critical, environmentally sensitive, or subject to privacy restrictions. In these areas, existing and emerging government regulations translate into a wide range of technical constraints that, in addition to the functional requirements, should be addressed in the realization facet. In industrial sectors such as medical devices, aerospace, and defense, regulations require certification processes, which deeply influence system design and have significant implications on the business case. Frequently, existing certification methods designed for previous generation systems create technical challenges that are not yet answered. An additional element of the business layer in the realization facet is incentives. Incentives are important tools for coupling the business layer to all phases of the CPS lifecycle. The emerging field of incentives engineering views the design of incentives and market mechanisms as a tool for optimizing the operation of large, distributed CPS with many conflicting operational objectives.

A.2.3 Lifecycle Management Layer

As with other engineered products, the CPS lifecycle covers phases from engineering design through manufacturing/implementation, to operation and to disposal of products. Operation includes configuration, updates, upgrades, diagnosis associated with commissioning and deployment, maintenance, and decommissioning. CPS construction has strong impacts on the overall structure and individual phases of the lifecycle.

- **Integration between design and manufacturing/implementation**. In cyber components and systems there is a well-known tight relationship between design and implementation. In physical systems the design and manufacturing steps are traditionally more isolated. The strong co-design aspect of CPS stimulates searching for new tradeoffs between cyber and physical boundaries driven by manufacturing constraints and lifecycle cost considerations of physical components. In addition, new research directions focus on co-design of products and their manufacturing processes.
- **Evolving CPS**. Cyber and physical components have very different lifecycles. This brings up the need for partial system updates and the opportunity for creating CPS that evolve – increasingly blurring the boundaries between the phases of the lifecycle.

While each lifecycle phase could be further elaborated to show CPS impact, for simplicity's sake within the scope of this document, Figure 16 elaborates only the operations layer portion of the horizontal plane. The reader should refer back to the treatment of composite concerns in Chapter 2.

A.2.4 Design, Manufacturing, Operations and Disposal Layers

These layers, shown as segments of one horizontal plane in Figure 11, are comprised of the realization activities for Design, Manufacturing, Operations, and Disposal of the CPS.

A.2.4.1 Design Layer

Current engineering design flows are clustered into isolated, discipline-specific verticals, such as CAD, thermal, fluid, electrical, electronic control, and others. Heterogeneity and cross-cutting design concerns motivate the need for establishing horizontal integration layers in CPS design flows. This need can be answered only through the development of new standards enabling model and tool integration across traditionally isolated design disciplines.

An important challenge in CPS design is the simultaneous satisfaction of cross-cutting design concerns. While system complexity largely depends on the extent and richness of interactions among components, design complexity is strongly influenced by the number of, and interdependence among, design concerns. Just as restricting and controlling interactions in systems is a key to decrease behavioral complexity, "separation of concerns" is the most frequently applied engineering principle to mitigate design complexity. Cross-cutting concerns

are essential in all phases of the lifecycle because they limit the effectiveness of the separation of concerns principle in correcting problems in manufacturing/implementation, operation, and retiring CPS.

See Appendix B for an elaborated discussion of concerns organized by aspects. In the realization facet, activities (listed in Table 9) address these concerns and aspects. Some prominent concerns are discussed below relative to the realization facet:

Performance is the primary driver of creating a CPS. It captures concerns that carry values for users, and is usually expressed as delivered functionalities, related capabilities, and performance metrics. Many design tradeoffs in the realization facet are expressed in terms of compromises between utility and some other category of concerns.

Safety properties of CPS express their capabilities for mitigating and avoiding hazards. In many CPS domains, safety considerations are key factors that influence decisions in all system layers. For example, in safety-critical CPS, regulations may require certification of safety properties that in turn motivate the selection of architectures and design methods for verifiability; exert influence on manufacturing, testing, and system operation; determine the level of abstractions used for modeling physical components and processes; and impose restrictions on acceptable physical architectures.

Security and privacy have emerged as major concerns in CPS. As opposed to IT cybersecurity, which focuses only on mitigating the impact of cyber-attacks, CPS security and privacy consider the coordinated exploitation of both physical and cyber vulnerabilities. Impacts of security and privacy considerations are pervasive on multiple layers of a CPS instance.

Time and synchronization are fundamental concerns due to the inherent role of time in the physical side of CPS. This category of concerns leads to services and protocols necessary to:

- Ensure that the temporal aspects of data that are common to more system components are based on a common understanding of reference timescales so that logical operations and computations on these data are meaningful.
- Ensure that ordering of system-wide operations based on some defined temporal relationships are correct.
- Ensure that time interval requirements including latencies are met.
- Enable the explicit use of timing and synchronization abstractions in complex, distributed CPS. Timing abstractions for correctness by design have not been generally available.

Interoperability and compositionality are key concepts in both engineering systems from components and developing SoS. For systems that leverage information, *interoperability* means that system components are able to exchange data based on a shared interpretation and able to interact to coordinate operations. *Compositionality* means that properties of composed systems can be computed from properties of the systems' components. Compositionality is

crucial in integrating large systems. Satisfying the conditions for compositionality ensures "correct by construction" (i.e., the elimination of design/manufacture/build/test/re-design iterations). There are many open challenges to achieving interoperability and compositionality in CPS due to the impacts of heterogeneity and the difficulty in specifying and enforcing timing constraints and specifications.

A.2.4.2 Manufacturing Layer

CPS manufacturing incorporates both physical and cyber components as well as their integration. As product complexity is increasingly migrating toward software components, industries with dominantly physical product lines need to change. This transformation is frequently disruptive and requires the adoption of new manufacturing platforms, design methods, and tools, and tighter integration of product and manufacturing process design.

A.2.4.3 Operations Layer

CPS operations deliver the utility for users. Accordingly, the operations layer extends to functionalities and services implemented by the networked interaction of cyber and physical components. While the functional architecture of CPS is domain-specific, there are common functionalities incorporated by many systems, which can be captured in the CPS Framework. The common elements include physical and cyber entities, information flows among them, functionalities such as hierarchical control layers, monitoring, anomaly detection, self-diagnostics and contingency management systems, and models that support operation and human operators.

CPS operations cover the phase of the life-cycle where benefits of new technologies are manifested in terms of better performance, increased autonomy, new services, dependability, evolvability, and other characteristics. Adaptability and autonomy create new challenges in assurance, since the behavior of advanced CPS products will evolve as a result of operation-time learning and reconfiguration.

A.2.4.4 Disposal Layer

Disposal of physical components (and associated costs) is an integral part of the overall lifecycle management process. Rising concerns over sustainability and environmental friendliness create pressure to move the cyber-physical boundary to decrease the physical footprint.

A.2.5 Cyber-Physical Abstraction Layers

The cyber-physical abstraction layer(s) forms a suite of structural and behavior models of systems that span both cyber and physical aspects. The abstraction layers and related ontologies and modeling languages are selected according to the essential properties to be verified and tested during design and monitored during operation. Some of these models (for

example, lumped-parameter physical dynamics of controllers of physical processes) represent behaviors that are refined during implementation to software and to physical computation platforms. Similarly, physical interactions may also be virtualized by mapping them to information flows connected to the physical world through sensors and actuators. Timing is an essential component in many CPS that rely on precisely coordinated interactions between physical and computational processes. In these systems, challenges go well beyond the introduction of physical time abstractions in computing (which has a rich history in real-time computing). New challenges and opportunities emerge from integrating the rich concurrency models in computing with time abstractions in physical systems and finding solutions for managing timing uncertainties.

Abstraction layers are usually defined by modeling languages that capture the concepts, relations, and well-formedness rules that each model must satisfy. In other words, modeling languages introduce invariants that all design (captured in the modeling language) satisfies. An important role in selecting modeling languages (i.e., abstraction layers) is to ensure that essential properties (such as stability or timing) are guaranteed by the introduced invariants.

A.2.6 Physical Layer

All CPS incorporate physical systems and interactions implementing some forms of energy and material transfer processes. Physical systems include plants, computation and communication platforms, devices, and equipment. CPS abstraction layers explicitly model the structure and behavior of these physical processes and express their relations to cyber models. They link information flows to physical variables via sensors and actuators and modeling the deployment of computations and information flows to platforms. Consequently, CPS design flows do not abstract out physicality in computations, but consider the implementation side effects of computations and networking on abstracted behaviors.

A.3 Assurance Facet

This section examines the assurance facet. It comprises the following sections:

- Section A.3.1 provides an overview of the assurance facet.
- Section A.3.2 discusses the assurance facet's activities and artifacts.
- Section A.3.3 explores the foundations of CPS assurance.
- Section A.3.4 describes patterns of assurance.
- Section A.3.5 discusses assurance and uncertainty.
- Section A.3.6 examines composability and compositionality.
- Section A.3.7 describes the need for assurance.
- Section A.3.8 describes Evaluation Assurance Levels.

A.3.1 Overview

The assurance facet complements and uses the artifacts of the conceptualization and realization facets. Its purpose is to capture the judgments made that the evidence accumulated in the facets is sufficient to conclude that the properties gathered in conceptualization are satisfied, with a defined level of confidence, and to conclude that the overall CPS model has been realized as intended. Assurance efforts appear throughout the execution of CPS conceptualization and realization.

Figure 17 captures how the activities of the other facets come together to support the activities of the assurance facet.

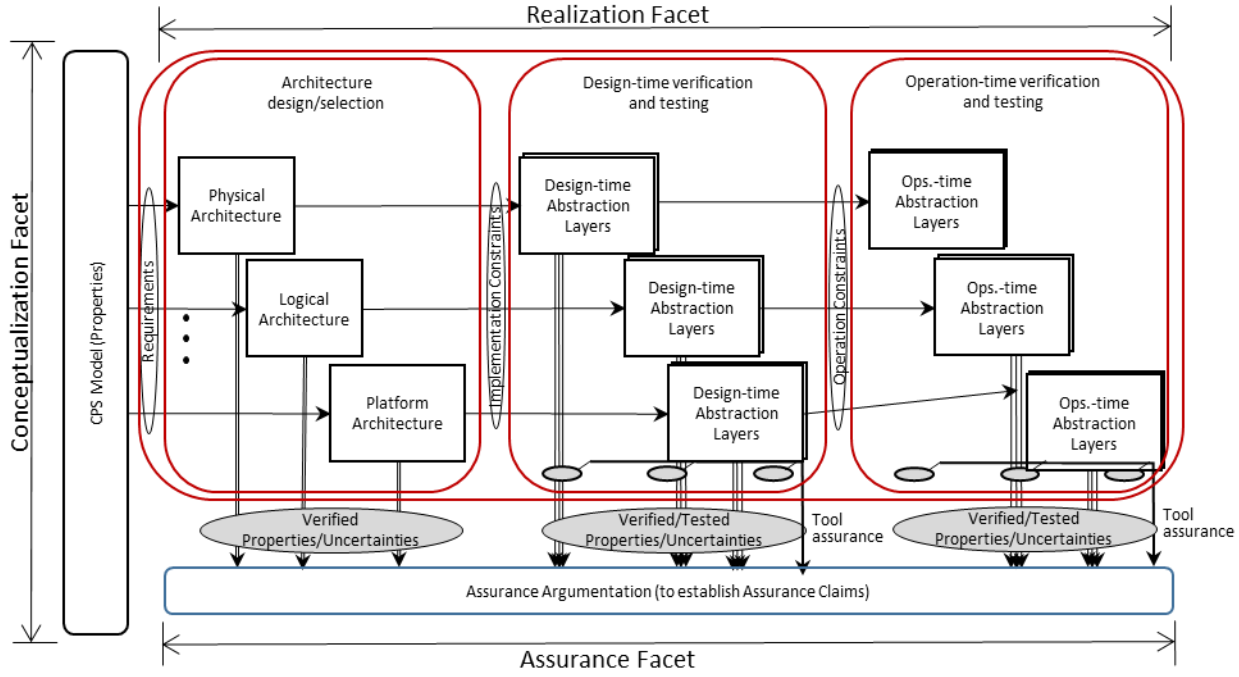


Figure 17: Assurance Facet

A.3.2 Activities / Artifacts

Table 10 contains a list of typical assurance activities and the artifacts resulting from these activities. There are activities associated with *assurance planning*, such as identifying the assurance objectives, defining the strategy to achieve those objectives, managing/controlling the assurance evidence, analyzing that evidence, forming the assurance argument, and providing the estimate of confidence that assurance objectives have been met. There are also activities specific to the domain, such as assessing the product configuration for completeness and integrity, assessing the requirements and test results, and performing certification and compliance testing.

Table 10: Assurance Activities and Artifacts

Activities and Artifacts
Identify Assurance Objectives **Artifact: Assurance objectives/analysis report**
Define Assurance Strategy **Artifact: Strategy document/plan**
Control Assurance Evidence **Artifact: Control documentation**
Analyze Evidence **Artifact: Analysis report**
Provide Assurance Argument **Artifact: Assurance argument report**
Provide Estimate of Confidence **Artifact: Confidence estimate**
Configuration Audit **Artifacts: Product configuration assessment**
Requirements Verification **Artifact: Requirements and test results assessment**
Product Certification and Regulatory Compliance Testing **Artifact: Certifications**

A.3.3 The Foundations of CPS Assurance

The assurance facet is a set of *activities* and *artifacts* with outcomes that are *judgments* that relate in a concrete way to the *activities* and *artifacts* of the conceptualization and realization facets of the CPS Framework. The assurance facet's artifacts, as depicted in Figure 18, take a form consisting of:

- Claims
- Evidence
- Argumentation
- Estimate of confidence

The typical statement of assurance takes the form:

> "The [Evidence] is sufficient to conclude that the [Claims] are true based on the [Argumentation] with this [Estimate of Confidence]."

This is the form of an *assurance judgment.* Ultimately this relationship between evidence, claims, argumentation, and estimate of confidence can be formalized. In this formalization a judgment will have assumptions that are themselves judgments. *Derivation rules* can be used for deriving new judgments from given ones, i.e., one can apply formal reasoning to derive assurance judgments that themselves provide a justification for accepting the derived

judgment. As an example, these rules may simply capture the reasoning suggested or dictated by a standard.

The derivation of a judgement contains all of the evidence associated with the intermediate judgments.

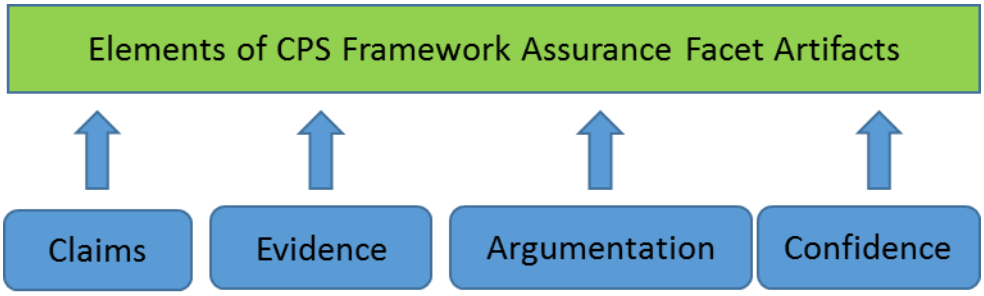


Figure 18: Elements of Assurance

The *claims* in the assurance facet are formed from the elements of the artifact of the conceptualization facet, the CPS Model. The CPS Model consists of the properties of the intended CPS. The claims in the assurance facet are the assertions that the CPS in question has or satisfies each of these properties. The CPS is said to *satisfy* the CPS Model if it satisfies or has each of the CPS Model properties. In the transportation and software safety (ISO 26262) examples above, the high-level assertion would be that the processes of the organization that developed the CPS are ISO 26262 compliant.

The *evidence* of the assurance facet is formed from the artifacts of the realization facet, such as process documentation, design artifacts, test plans, and results, as depicted in Figure 19. They are determined by the specialization of the realization facet activities and artifacts to their domain and the applicable aspects.

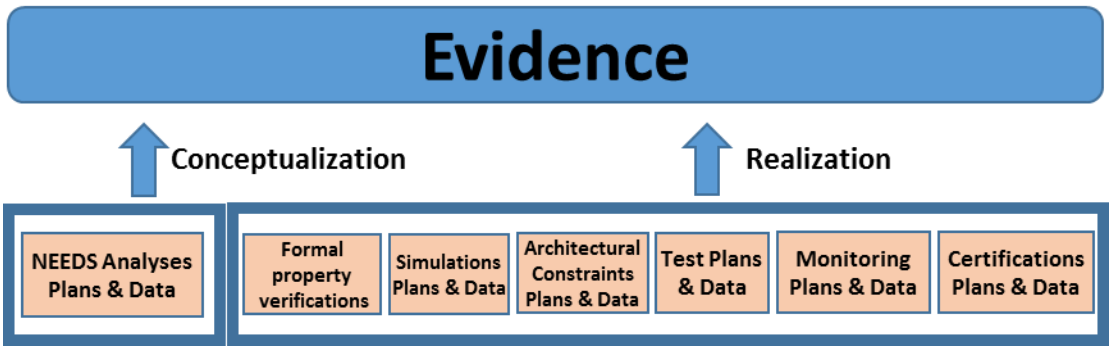


Figure 19: Evidence

As shown in Figure 20, the *argumentation* of the assurance facet is formed from a variety of things, including appeal to:

- Standards
- Best practices/consensus
- Formal methods
- Regulation (proscribed practices)
- Expert judgment (including criteria for being an expert in a domain)

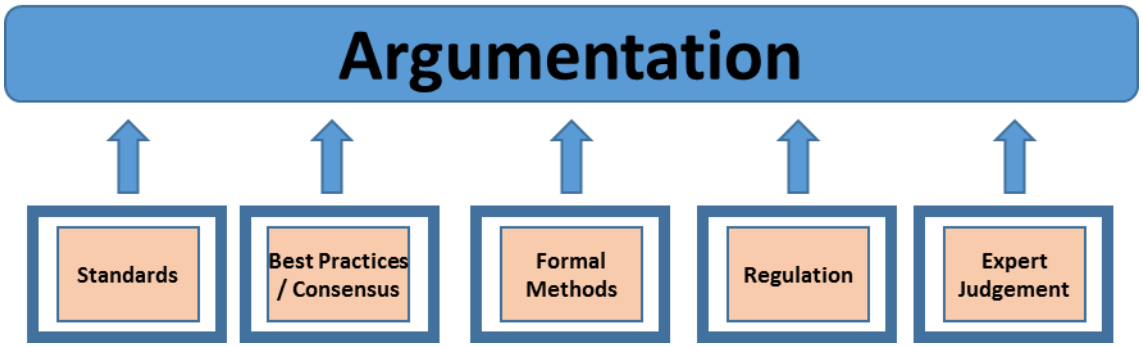


Figure 20: Argumentation

This information is provided during the application of the CPS Framework, when each of the facet activities reviews each of the aspects of the Framework for impact on that activity. The output of that review is an updating of that activity and its artifacts. It is intended to provide criteria for evidence and supporting argumentation in the assurance facet to ensure that the concerns in that aspect have been adequately addressed in the activity.

A.3.4 Patterns of Assurance: Executing the Framework

The Model of the CPS (or CPS Model) is the artifact of the conceptualization facet and consists of a structured set of properties that the CPS under analysis or construction must satisfy. It is structured in the sense that there are dependencies between these properties as they evolve throughout the activities of this CPS framework facet.

A CPS provides one or more *functions,* and the CPS Model are properties of those functions. Some of these properties are imposed by the Aspects/Concerns that apply to the CPS and others result from requirements analysis:

- Mission/Business case, describing the functions the CPS provides: $P_{M/BC}$
- Functional Decomposition, describing the other functions required to realize the CPS function, together with the 'steps' or interactions between them required to enable the CPS function: P_{ARCH}; in addition, there are properties corresponding to any assumptions P_{ASS} and the success criteria P_{SUCC}

- Requirements Analysis and properties that result from applying any Aspect/Concern that applies to the CPS: $P_{Aspect/Concern}$

To facilitate addressing the assurance for any of the properties in the CPS Model, we document the properties of the CPS developed during the conceptualization facet in the form of a tree of properties. Formally, a tree is a partially ordered set with a unique root (all nodes trace back ultimately to the same node) and no *cycles* (a cycle corresponds to a node that can be reached from itself following a non-trivial path in the tree). The *property tree of a CPS*, consists of the properties of the CPS Model, ordered under the traceability ordering. The root is the property $P_{M/BC}$ with the successor relation outlined above.

Graphically this tree has the appearance shown in Figure 21:

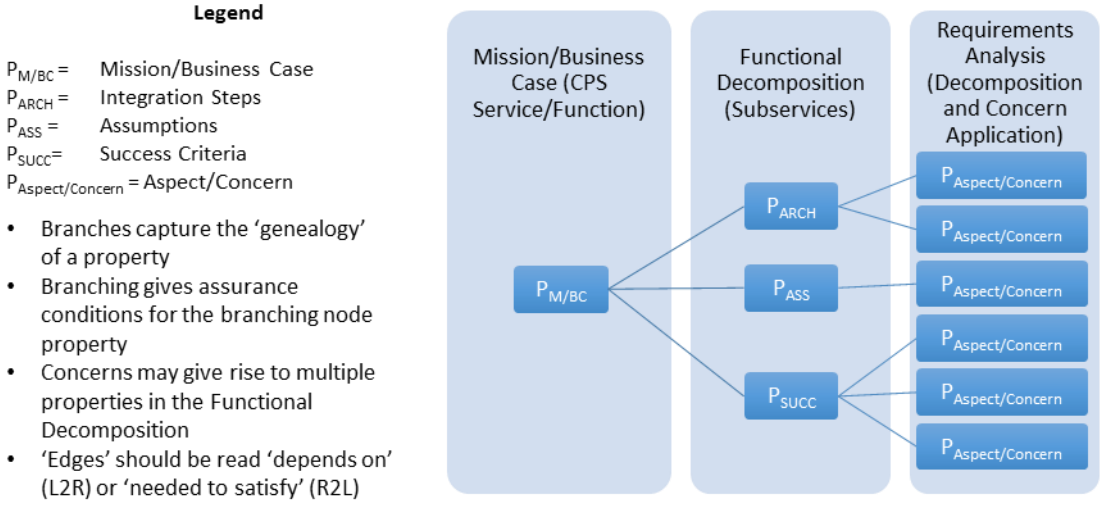


Figure 21: The Property Tree of a CPS

There are two types of assurance arguments, structural and empirical. Which one is applied depends on the types or source of properties to be assured.

The 'branching type' assurance argument (1) itself has a couple of different flavors, one for assurance of a logically compound property (containing propositional connectives) and one for properties that are compound due to the componentry of the CPS and its interactions.

The 'leaf type' assurance argument (2) is one that relates to the design D and test T put in place in order to achieve a property of the CPS.

(1) $H_{Branching}$

(2) H_{Leaf}

The argument, "A", in this case may take the form that the test itself, the setup for the test, and the way in which test results are stored and managed complies with a standard. The argument would make reference to the standard as an example.

(1) Structural (logical and architectural) for branching properties, P and Q:

$A(P*Q) =_{Def} H_{Branching} (A(P), A(Q))$,

for logically compound or architecture properties

(2) Empirical for the terminating properties or 'leaves' of the tree:

$A(P, D, T) =_{Def} H_{Leaf} (P, D, T)$

Where each leaf is the argumentation that the design, test, and tracing to property are sufficient to conclude that the property P is met.

In specific instances, H_{Leaf} may make reference to a certification, standard, or regulation where this test is recommended or required to establish the property in question with a certain level of confidence.

A.3.5 Assurance and Uncertainty

There is almost always an element of uncertainty associated with the judgments in the assurance facet. The *assurance claims* typically contain information about the level of confidence that the CPS in question will satisfy the properties that make up the CPS Model. Confidence can be achieved in different ways. It may be a number or another form of estimate, but its representation should be easily understood and the actions needed to update that estimate, in case of design changes, should be clear and executable.

The activities associated with assurance have as their objective to provide an independent assessment of the overall process of conceiving and realizing a CPS. The artifacts of these two facets include models/analyses, engineering drawings, specifications needed to physically and functionally describe the intended CPS, documentation required to support procurement, manufacturing, test, delivery, use, maintenance, and disposal of the CPS.

A.3.6 Composability and Compositionality

Composability and compositionality are concepts critical to an understanding and successful application of the CPS Framework. Analysis of these concerns is central to the topic of CPS as the science of the IoT.

Composability is a property of two or more systems that relates to our ability to successfully construct new CPS from given ones or to compose given ones to obtain new CPS. It means that system components can be composed in a meaningful way, i.e., they are able to exchange data

based on a shared interpretation of that data and they are able to interact and, in so doing, to coordinate operations. We say that a system is a *subsystem* of a second system if the second is the result of successive compositions of the first system with one or more other systems. This is one approach to the subsystem concept and will be explored in follow-on work to this Framework.

Compositionality, on the other hand, is the property of the composition of systems. It states that the status of a property of composite systems can be determined entirely from the status of that property of the systems of which it is composed. Restated, the truth of a property of a CPS – the fact that a CPS satisfies a property – depends only on whether the CPS of which it is composed satisfies the property. Usually the properties that are considered are limited to a specific set of CPS. A set of properties of CPS is said to be compositional if each property's truth is completely determined by its truth for the constituent CPS.

Since the CPS Model, the artifact of the conceptualization facet, consists of CPS properties, compositionality is crucial to the successful integration of large systems of CPS, i.e., integration that preserves the properties common to those CPS. Conversely, if it can be demonstrated that the component systems of a system all have a critical property, then compositionality for that property would entail that the system at hand satisfies that property.

Hence compositionality can be seen as a kind of proof by induction on the construction of a CPS of a property that is compositional with respect to the CPS construction principles used to obtain that system. Another way to put this is: satisfying the conditions for compositionality for a set of properties ensures "correctness with respect to those properties by construction" (i.e., the elimination of design/manufacture/build/test/redesign iterations.) There are many open challenges to achieving composability and compositionality in CPS due to the systemic impacts of heterogeneity. Timing has specific difficulties with compositionality. Timing correctness by construction is not generally available, particularly in cyber-physical systems today.

A.3.7 CPS and the Need for Assurance

CPS can be seen as extensions of human capabilities, including sensing, decision-making, and action. Many times human beings are more than aware of the limitation of their abilities, so assurance methodologies frequently provide both an extension of those abilities and an estimate of the uncertainty inherent in using these extensions. Humans maintain a certain level of situational awareness and many times need to be protected from errors in judgment.

In a world of human beings with CPS-enhanced abilities, assurance and estimates of the assurance level will increase the likelihood of success in CPS usage and increase their benefit to mankind.

High on the list of CPS challenges are topics related to *human factors*. The assurance facet should provide us with a clearer understanding of our newly found capabilities. In doing this the

interaction between operator and CPS may also be improved. Closer consideration of Figure 22 suggests that there is much research required to better understand the relationship between the cognitive cycle of a human operator and that of the CPS conceived, built, and operated by humans.

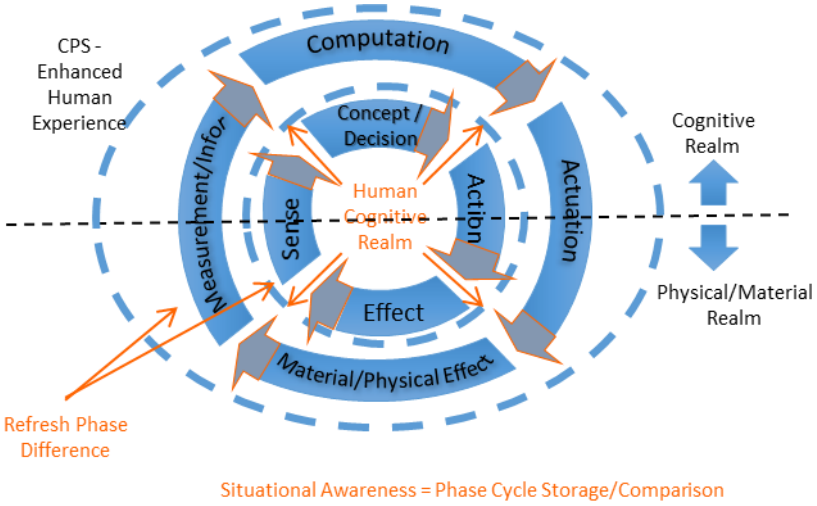


Figure 22: CPS Enhanced Cognitive Cycle

A.3.8 Related Work: Evaluation Assurance Levels

As noted earlier, the elements of assurance of a CPS consist of argumentation to the effect that the evidence produced during the activities of the CPS Framework facets is sufficient to conclude that the CPS satisfies the CPS Model produced in its conceptualization facet. To be useful, this CPS assurance should include a measure of our level of confidence in the conclusion.

As an example of such a measure of confidence used to evaluate the security of an IT product or system, we mention the Evaluation Assurance Levels (EAL) [24]. EAL is a discrete numerical grade (from EAL1 through EAL7) assigned to the IT product or system following the completion of a Common Criteria security evaluation, which is an international standard in effect since 1999. These increasing levels of assurance reflect the fact that incremental assurance requirements must be met to achieve Common Criteria certification. The intent of the higher assurance levels is a higher level of confidence that the system's security features have been reliably implemented. The EAL level does not measure the security of the system; rather it rather simply states at what level the system was tested.

- EAL 1 Functionally Tested
- EAL 2 Structurally Tested
- EAL 3 Methodically Tested and Checked
- EAL 4 Methodically Designed, Tested and Reviewed

- EAL 5 Semiformally Designed and Tested
- EAL 6 Semiformally Verified Design and Tested
- EAL 7 Formally Verified Design and Tested

For our purposes, the mention of the EAL levels are sufficiently explanatory, the detailed description of the rigorous methodology applied to design, test, and reviews can be found in the reference material on the subject. One comment on the use of 'semi-formally' and 'formally' in EAL 5 through EAL 7 is in order: This is a reference to the extensive formal analysis requirements for tightly focused security functionality, which is amenable to such analysis due to its structure and specificity.

Appendix B. Aspects of the CPS Framework

This section discusses the aspects and the dimensionality of their scope with respect to CPS. The nine aspects are:

- Functional (Section B.1)
- Business (Section B.2)
- Human (Section B.3)
- Trustworthiness (Section B.4)
- Data (Section B.5)
- Timing (Section B.6)
- Boundaries (Section B.7)
- Composition (Section B.8)
- Lifecycle (Section B.9)

Note that some of these aspects are fully elaborated in this draft (drafted by the subgroups: Trustworthiness, Data, & Timing). However, several other aspects discovered during the course of the framework development activity will need to be more fully developed after the initial public review. The aspects above are listed in a useful order for the benefit of organizing facet activities.

B.1 Functional Aspect

Concerns about function including sensing, actuation, control, communications, physicality, etc.:

actuation Concerns related to the ability of the CPS to effect change in the physical world.

communication Concerns related to the exchange of information internal to the CPS and between the CPS and other entities.

controllability Ability of a CPS to control a property of a physical thing. There are many challenges to implementing control systems with CPS, including the non-determinism of cyber systems, the uncertainty of location, time and observations or actions, their reliability and security, and complexity. Concerns related to the ability to monitor and, if necessary, modify a CPS or its function.

functionality Concerns related to the function that a CPS provides.

measurability Concerns related to the ability to measure the characteristics of the CPS.

monitorability	Concerns related to the ease and reliability with which authorized entities can gain and maintain awareness of the state of a CPS and its operations. Includes logging and audit data.
performance	Concerns related to whether a CPS can meet required operational targets.
physical	Concerns about purely physical properties of CPS including size, weight, volume, seals, locks, safety, EMI resilience, etc.
physical context	Concerns relating to the need to understand a specific observation or a desired action relative to its physical position (and uncertainty.) While this information is often implied and not explicit in traditional physical systems, the distributed, mobile nature of CPS makes this a critical concern.
sensing	Concerns related to the ability of a CPS to develop the situational awareness required to perform to its function.
uncertainty	Managing the effects of uncertainties is a fundamental challenge in CPS. Sources of uncertainty in CPS can be grouped into statistical (aleatoric) or lack of knowledge (epistemic) uncertainty. In CPS statistical uncertainty is caused by randomness of accuracy of sensing and actuation, often caused by uncertainty of manufacturing processes. Systematic uncertainty is caused by incomplete knowledge either due to limits of acquired knowledge or due to simplification in modeling. Typical manifestations of epistemic uncertainty are limited validity of models of physical processes or limits of computability of properties of mathematical models.

A more comprehensive treatment of this Aspect is anticipated in the future.

B.2 Business Aspect

Concerns about enterprise, regulation, cost, etc.:

enterprise	Concerns related to the economic aspects of CPS throughout their lifecycle.
cost	Concerns related to the direct and indirect monetary outflow or other resources required by the CPS throughout its lifecycle.
environment	Concerns related to the impacts of the engineering and operation of a CPS on the physical world and vice versa.

policy	Concerns related to the impacts of treaties, statutes, regulations, and doctrines on a CPS throughout its lifecycle.
quality	Concerns related to the ease and reliability of assessing whether a CPS meets stakeholder (especially customer) expectations.
regulatory	Concerns related to regulatory requirements and certifications.
time to market	Concerns related to the period required to bring a CPS from need realization through deployment.
utility	Concerns related to the ability of a CPS to provide benefit or satisfaction through its operation. Utility reflects a business concern, especially when considered as the numerator when computing value, which equals utility divided by costs.

A more comprehensive treatment of this Aspect is anticipated in the future.

B.3 Human Aspect

Concerns about human interaction with and as part of a CPS:

human factors	Concern about the characteristics of CPS with respect to how they are used by humans.
usability	Concerns related to the ability of CPS to be used to achieve its functional objectives effectively, efficiently, and to the satisfaction of users (adapted from ISO 9241-210.) The combination of physical and cyber into complex systems creates challenges in meeting usability goals. Complexity is a major issue. The diversity of interfaces creates a huge learning curve.

A more comprehensive treatment of this Aspect is anticipated in the future.

B.4 Trustworthiness Aspect

Trustworthiness is the demonstrable likelihood that the system performs according to designed behavior under any set of conditions as evidenced by characteristics including, but not limited to, safety, security, privacy, reliability and resilience.

In computer security, a chain of trust is established by validating each component of hardware and software from the bottom up. It is intended to ensure that only trusted software and hardware can be used while still retaining some level of flexibility (adapted from Wikipedia). The notion of the chain of trust is essential for cyber-physical environments that contain diverse hardware and software systems and need to preserve integrity to perform mission-critical tasks. Roots of trust in CPS represents an important topic, but is still an area under

development. Trust anchors in CPS are not addressed in a consistent way, and the approaches are fragmented. More research is required to ensure integrity of CPS.

This section describes the trustworthiness aspect of the CPS Framework. Components of this section are as follows:

- Section B.4.1 provides an overview of the trustworthiness aspect.
- Section B.4.2 discusses CPS cybersecurity and privacy risks.
- Section B.4.3 looks at moving from classic cybersecurity properties to cross-property risk management.
- Section B.4.4 introduces safety concerns
- Section B.4.5 introduces reliability concerns
- Section B.4.6 introduces resilience concerns

B.4.1 Overview

Emerging generations of CPS will extend the functionality and capabilities of existing information technology (IT), operational technology (OT)/industrial control systems (ICS), and embedded systems. They will provide an opportunity to leverage multi-disciplinary approaches as technologies converge to shape continued and future innovation across countless sectors of national and international economies. Designing these CPS will require international, cross-sector collaboration to produce desired benefits. These efforts will be influenced by common business and technical drivers, such as interoperability and standards-based platforms, a need for common reference architectures, and growing consumer/user needs.

CPS will integrate many traditional vertical applications/systems. As an illustrative example, a home energy management system may comprise temperature sensors to facilitate control of a heating and cooling system. Similarly, a fire alarm system may comprise smoke detectors for fire detection. The co-engineering and cross-coupling of information between these distinct applications may provide more accurate intelligence to be gleaned. If a fire is detected, then the readings from the temperature sensors can validate the existence of a fire if the temperature sensors also indicate high temperature readings. Alternatively, the temperature sensors may raise awareness of a potential false alarm if the temperature readings are normal.

New CPS will provide the next generation of “smart,” co-engineered interacting components connected over diverse networks. Composed of heterogeneous, potentially distributed, components and systems, CPS bridge the digital and physical worlds. Assuring that these systems are trustworthy in the broadest sense (e.g., reliable, resilient, secure, private and safe) poses unique cybersecurity challenges. Traditional approaches to cybersecurity, privacy, reliability, resilience, and safety may not be sufficient to address the risks to CPS. This produces a need for a cross-property risk management [25] approach that leverages and extends the risk

management approaches from historically disparate areas of expertise. To support the co-design aspect of CPS, a deeper understanding of the relative significance of, and interactions among, these properties is necessary to ensure the functionality of the CPS is not compromised such that a system produces unintended outcomes. This cross-property understanding will enable appropriate CPS design trade-offs and complementary cross-property design decisions.

Together in the context of CPS, the risk management properties defined above support the trustworthiness of the system – "the system does what is required despite environmental disruption, human user and operator error, and attacks by hostile parties and not other things" (Fred B. Schneider, Trust in cyberspace). To achieve trustworthiness of a system is greater than the sum of trustworthy parts.

The following sub-sections highlight the unique elements of the trustworthiness properties of CPS and how they relate to, and impact, the other properties in the context of CPS: cybersecurity and privacy (Sections B.4.2), and full dimensions of Trustworthiness in section B.4.3.

B.4.2 CPS Cybersecurity and Privacy Risk

In its broadest sense, providing CPS cybersecurity will require significant changes that must reflect how systems and applications are designed, deployed, and applied across both legacy and new systems. New standards affecting design, engineering configuration, automation, and communication must be instituted to ensure desirable outcomes. When considering cybersecurity for CPS, it is important to focus on CPS physicality and the operational constraints it may place on CPS cybersecurity strategies. Certainly, many of the cybersecurity challenges that apply to IT systems also apply to CPS. However, some challenges may not have the same criticality in the CPS space as they do in IT systems, and CPS may pose additional challenges not present in the IT space. Further, the mechanisms used to address IT challenges may not be viable in the world of CPS. The physicality of CPS also presents opportunities for cybersecurity solutions that are not available to IT solution providers.

In addition to required reliability, as society becomes more aware of the risks associated with lack of privacy, challenges emerge as to how to use information in CPS while preserving and perhaps enhancing privacy.

B.4.2.1 Cybersecurity challenges

B.4.2.1.1 Overarching issues

Perhaps the most significant challenge in providing cybersecurity for CPS is addressing the requirement for resilience. For engineered systems, *reliability* engineering is intended to ensure predictable system performance (the system behaves as it was intended to do) and safety in sets of predetermined conditions—the system's expected operating environment. It addresses

risk, situations where the distribution of possible outcomes (system behavior and the impacts that result) produced by the interaction of the system and its environment are known. *Resilience*, on the other hand, is intended to address *uncertainty*, situations where the distribution of possible outcomes produced by the interaction of the system with its environment are NOT known, often because the environmental conditions that produce the impacts are unknown or not well understood. Much resiliency engineering focuses on situations where the environmental conditions have deliberately and intentionally been manipulated by malefactors. Using these definitions, many key cybersecurity and privacy challenges lie not in the domain of reliability, but in the domain of resilience.

CPS cybersecurity must protect operational goals from the impacts of malicious cyber-attack, enabling continuing safe operations even in compromised conditions. Cybersecurity for CPS must address how a system can continue to function correctly when under attack, provide mechanisms that support fault-tolerance and/or graceful degradation in accordance with mission- or business-driven priorities, and enable the system to fail-safe in those circumstances in which resilience cannot be provided in the face of threat.

Providing cybersecurity for CPS is further complicated by the fact that an ever-expanding array of CPS will be required to operate in a wide range of operational conditions, and could be threatened by a plethora of cyber-attack mechanisms and processes. Security concepts, processes and solutions must encompass that breadth. When thinking about this issue, it can be helpful to visualize the set of CPS as a continuum. On one end are the safety-critical systems. Safety-critical systems are those systems whose failure could result in loss of life, significant property damage, or damage to the environment. There are many well-known examples in application areas such as medical devices, aircraft flight control, weapons, and nuclear systems.[15] These systems are often highly regulated and physically protected, and are the product of careful design and significant capital investment. On the other end of the continuum are consumer convenience or entertainment devices. These systems may assume no limits on access, and are produced in a variety of development environments (some of which are relatively unstructured) at a sufficiently low cost that they are considered disposable commodities. Cybersecurity and privacy professionals must recognize that because the capabilities of these systems are converging, cybersecurity efforts must be prepared to address the entire continuum. Consider wearable or implantable medical devices: they are safety-critical and somewhat regulated, but exhibit limited physical protection, are almost always accessible, and are produced and used in environments similar to the consumer goods

[15] J. C. Knight, Safety Critical Systems: Challenges and Directions, http://users.encs.concordia.ca/~ymzhang/courses/reliability/ICSE02Knight.pdf

environment. Yet security and privacy considerations are as critical to their safety and integrity as for industrial controls or critical elements of the power grid.

The system-of-systems (SoS) nature of many CPS introduces another challenge to providing cybersecurity for CPS. A SoS is not necessarily designed as a coherent system—it can emerge as the result of opportune connections among systems that may have never been designed to interact with each other. It can be difficult to pin down the boundaries of a SoS. Analysis and design of cybersecurity capabilities for CPS are complicated by the need to understand and address the upstream and downstream dependencies of the component systems. Where the SoS consists of systems owned by multiple entities, there is also the issue of determining responsibility for the security of the whole CPS and how responsibility is shared or trust relationships are established among responsible entities to assure global protection.

The extreme scalability of CPS also presents challenges. The emergence of the IoT increases the number of connected entities on a scale that dwarfs current IT networks. Huge networks of small sensors are becoming more commonplace. Security mechanisms and the infrastructure to manage them must be able to scale up to accommodate these structures.

The next generation of CPS should have a threat-aware based approach that supports development of systems that are resilient by nature. As such, the ability to not only detect one or more threats, but also correlate those threats with their impact on system behavior is a necessary capability. What follows is a synopsis of the current complexity challenges that will need to be addressed in future CPS designs. First, current automation environments are the result of organic interconnection of CPS and the inability to anticipate, recognize, and prevent resulting faults. Secondly, benign human error as the result of data overload and lack of information is an ongoing issue. For the malicious human, current perimeter protections are insufficient and not designed to adapt rapidly to attacks in order to prevent compromise. Finally, current CPS have multiple performance goals, but without the necessary identification and prioritization, the goals can lead to undesirable response from both the human operation and the automation design.

The following are the aspects that shape resilience in understanding the multi-disciplinary nature of these issues, which clearly require human, control, and cyber systems to address holistically[16]:

- Unexpected condition adaptation

[16] http://ieee-ies.org/resources/media/members/committees/resia/challenges/MultidisciplinaryProjectforReSia.pdf

- Achievable hierarchy with semi-autonomous echelons: The ability to have large-scale, integrated supervisory control methodologies that implement graceful degradation.
- Complex interdependencies and latency: Widely distributed, dynamic control system elements organized to prevent destabilization of the controlled system.

- Human interaction challenges
  - Human performance prediction: Humans possess great capability based upon knowledge and skill, but are not always operating at the same performance level.
  - Cyber awareness and intelligent adversary: The ability to recognize and mitigate cyber-attacks is necessary to ensure the integrity of the control system.

- Goal conflicts
  - Potentially conflicting goals and flawed understanding of the factors affecting system behavior: Besides stability, security, efficiency and other factors influence the overall criteria for performance of the control system.
  - Lack of state awareness: Raw data must be translated to information on the condition of the process and the control system components.

B.4.2.1.2 Challenges due to interaction with physical world

Another set of CPS cybersecurity challenges stems from the fact that CPS are designed to interact with the physical world. Perhaps the most obvious of these is that the impact of attacks on a CPS can be physically catastrophic. In addition to threatening intellectual property (a problem that is common to all IT systems), attacks on CPS can adversely impact product quality, operational safety, and product performance. When compared with IT systems, this means there may be a different level of tolerance for threats against CPS, and a different level of urgency in addressing attacks. A denial of service attack against a website produces loss of access to data, loss of revenue, or even damage to a server, but if the attack is addressed in minutes, recovery may not be difficult. By contrast, a denial of service attack against the system that regulates the safe operation of a power generation facility or an industrial plant can lead to irreparable damage to capital equipment that could take months or years to replace. For systems like these, the time scale for addressing the attack cannot be minutes. In addition, CPS are sometimes deployed in ways that preclude physically securing all of their components. This increases the likelihood that cybersecurity processes will be operating in a compromised environment.

Because CPS interact with the physical world, they are subject to the time constraints of the physical process they are executing. These processes are generally time-aware and deadline-

sensitive. As a result, security processes must fit within the time constraints of the application. Current IT cybersecurity controls may need to be modified significantly, or be completely replaced, because those solutions cannot meet the timing criteria required by CPS. Further, the tight time constraints on addressing attacks largely rule out human-in-the-loop solutions. This drives the need for continuous, autonomous, real-time monitoring, detection, and response.

B.4.2.1.3 Challenges due to operational constraints

The operational settings of CPS are often very different from those of IT systems, particularly enterprise systems. This challenges the application of existing cybersecurity paradigms to CPS. Moreover, the operational settings and requirements vary greatly across the range of CPS, so the challenges are not uniform for all CPS. Thus, it is useful to consider a variety of operational implications for CPS cybersecurity.

CPS often exist on resource-constrained platforms. As a result, security mechanisms must be lightweight in terms of storage space, memory use, processor use, network connectivity, and electrical power consumption. Furthermore, these platforms are often distributed; the individual components must perform global tasks using local information exchange and limited computation at the nodes.

Cybersecurity for CPS generally must accommodate the in-place business processes. Access controls and authentication and authorization mechanisms must accommodate the fact that CPS are often deployed in operational situations that require immediate access to control systems or access by any member of a group. “Strong” passwords, passwords that are lengthy or complicated to enter, or passwords that require frequent updates are often inappropriate for such environments. On the shop floor, passwords are often shared among all the individuals holding a particular role to eliminate potential discontinuity between shifts and provide rapid emergency access to the system. New mechanisms to establish trust between machines and people are needed for these conditions.

CPS often have "always on" requirements. This makes rebooting and patching non-viable strategies for many systems. Furthermore, the software that executes processes in many of these systems is often old and has required extensive analysis and testing to meet safety requirements; it cannot be easily changed because the “downtime” cost of implementing changes is prohibitive.

In several CPS sectors (including, but not limited to, transportation and emergency response), the domain of use is dynamic. Actors, be they people or machines, come and go. The set of valid users is constantly changing at an ever-quickening pace. Traditional key management is ineffective over large “accidental” populations of this type. For example, consider the impact of providing keys to all the driver-assisted or autonomous vehicles on any major road during peak traffic. Encryption mechanisms are not likely to work under such dynamic conditions without new keying mechanisms and protocols. The dynamism of system configuration is increased by

two other facts: in many use cases, nodes are intermittently unavailable; in others, nodes may change context (and the attendant security requirements) depending on the task at hand. The variable reliability of human participants also adds to the level of system dynamism.

B.4.2.1.4 Lifecycle issues

A number of lifecycle issues also complicate the cybersecurity of CPS. Some operational technology and infrastructure CPS have very long lifetimes (30 years or more). These systems are difficult to change; industry needs strategies that both "future-proof" designs and allow for integration with systems. In some cases, the verification cost of these systems locks owners into old technology; they need methods that enable rapid reassessment and conjoined maintenance of new and legacy systems. This raises challenges associated with composability; therefore, new system designs should be informed by the need to accommodate existing devices.

CPS owners and operators must also consider the potential cybersecurity effects of the "orphaned devices and code" they may have. Orphaned devices are devices for which no firm provides the required support (e. g., operating system upgrades and cybersecurity patches). Orphaned code is code that the system no longer uses but may be still present in the CPS. Even on devices that are not orphans themselves, patches may not address issues that spring from orphaned code. This equipment or code cannot be made resistant to emerging threats; rather, it poses a risk to any network to which it is connected. Additional challenges can be introduced by inappropriate use of throwaway systems, which have a limited lifespan by design, but which are never removed from the environment and can be co-opted in an attack. In both the static and the dynamic environments, there is a need to understand lifecycle threats and take a systems engineering approach to address the security of the manufacturing process, supply chain, and the commissioning, operation, and decommissioning of devices.

B.4.2.2 Privacy challenges

Any analysis of privacy risks in CPS must consider the processing of personal information throughout the entire data lifecycle, from data creation/collection through to disposal. Such analysis provides the foundation for understanding the context of the system, the identification of potential privacy risks, and the definition of appropriate privacy controls as part of the system design specifications. It is therefore critical that this analysis takes care to fully comprehend the system, data sensitivity, and associated data lifecycle before determining privacy risks and requirements. It should be noted that there are likely to be many situations where privacy may not be implicated at all, for example, in the monitoring of an industrial control system.

The single most significant factor impacting privacy as Cyber-Physical Systems proliferate throughout the global infrastructure, is the sheer volume of data generated, collected and subsequently analyzed. This data may not be considered to be personal information initially,

but may be correlated with other data and then attributed to an identifiable individual. The challenge will therefore center on how best to apply existing regulatory and legal obligations, internationally-recognized privacy principles, and privacy engineering and risk management processes in a world where more and more systems are interconnected and generate a vast volume of data, some of which may have significant privacy implications.

Another important factor to consider is the impact of a CPS privacy violation may be quite different from that of a traditional information privacy violation and may have a greater likelihood to result in risks leading to physical harm to individuals and property than in traditional IT systems. The operation of CPS may manipulate or modify individuals' behavior by constraining choices or opportunities and limiting their action in the physical world, either as a by-product of the system functionality or as the result of a malicious actor.

Examples of issues:

Large volumes of data, once correlated and analyzed, may provide significant insight into an individual's life, thoughts, and beliefs and may potentially be used to discriminate against or cause harm to individuals. This data will need to be protected from unauthorized access, modification, misuse, and loss.

There are cases in which certain types of data in a CPS may have little or no privacy implications in isolation, but when combined with other types of data could be privacy intrusive. Data collected in one context may be repurposed and reused without the knowledge or consent of the data subject.

Individuals may suffer physical harm as a result of privacy violations in CPS. These may include, for example, through the generation of inaccurate medical device sensor readings, the automated delivery of incorrect medication dosages via a compromised insulin pump, or the malfunctioning of critical smart car controls, such as braking and acceleration.

The system-of-systems nature of CPS produces highly complex interrelationships among systems that make it more difficult for users to understand what is happening with their information and where it is going. Traditionally, organizations have relied on privacy notices and publicly posted policies in attempts to provide transparency about their use of data and to gain consent from individuals. Many now question the efficacy of such methods even in traditional systems, but in a CPS environment, the opportunity to provide such forms of transparency may be all the more limited.

Currently, there is no clear understanding of the privacy tipping point for aggregation of data elements across many interconnected systems. In addition, CPS data are often collected for the sake of the management of the system, not for any user-driven purpose, and the individual may not have the opportunity to control or derive appropriate value from their data. In short, the division of rights between users and system owners to manage and use personal information is unclear. In these cases, professional designers guided by best practices, standards, regulations,

and norms in this area are responsible for characterizing the tradeoffs between the gains made by the collection of such data (forecasting, non-technical losses/revenue protection, etc.) versus the privacy costs/losses experienced by operators and consumers.

Finally, there is the consideration of data leakage or "exhaust" – information that is leaked as a consequence of using a system or operating in a CPS-enabled environment. For example, Non-Intrusive Load Monitoring (NILM) leaks device usage information through the power line. The simple act of turning on an automobile leaks information en route. Water and gas flow changes leak information about control structures. Smart meters communicate energy usage information to utilities over the network infrastructure. However, the frequency with which energy usage data is collected has increased from once a month or so to once every few minutes. While this certainly captures energy usage data with greater resolution than ever before, it also provides insight into many behavioral patterns and profiles of the individuals who reside at that residence, such as whether anyone is at home, if individuals are awake or asleep, cooking, or watching TV. The signatures of specific appliances, including medical equipment, may also become apparent. In essence, high-frequency data collected for the purpose of energy monitoring unintentionally leaks confidential information well outside the context of energy. There also exists the potential to infer information from an individuals' behaviors, such as their search of information repositories and their general day-to day-interactions with CPS.

Complete consideration of privacy risks, along with the other top level trustworthiness management properties of cybersecurity, safety, reliability and resilience, should help to proactively address these concerns and provide a strategy for appropriate risk management in CPS.

Privacy Recommendations

- Research should be directed at new methods for enabling individuals and system operators to effectively and appropriately manage information and have reliable assumptions about the collection and processing of their information in CPS.
- Research is needed for technical measures that can enable the processing of information without association to individuals or their devices except within certain narrowly-scoped operational requirements.
- Research is needed to determine how best to apply existing internationally-recognized privacy principles and best practices and develop complementary approaches that scale in line with the proliferation of CPS.
- Efforts should be made to establish and professionalize the discipline of Privacy Engineering.

- Organizations and practitioners are encouraged to maintain awareness and, as they are capable, to participate in the development and evolution of CPS privacy norms and standards.
- Research needs to be focused on developing technical standards that can enable the functional benefits of the system while mitigating the privacy risks to the maximum extent.

B.4.2.3 Opportunities

Though the nature of CPS introduces many cybersecurity and privacy challenges, it also presents some opportunities that may enable use of novel approaches to securing these systems or make viable some approaches that are difficult to implement in the more open world of IT systems. The laws of physics often constrain the operations of CPS. As a result, the normal behavior range of a given CPS is often well understood (the province of reliability engineering). These features may make anomaly detection and control easier (the province of resilience engineering, especially when the anomalies were the result of someone's desire to produce adverse effects). CPS have comparatively well-defined network dynamics: servers rarely change, the topology is often but not always fixed (i.e. mobile devices), the user population is relatively stable, communication patterns are often regular, and the number of protocols is limited. These parameters can be modeled, and the model of the dynamics of the system can be used to detect a compromised node or identify out-of-norm behavior. Because of these more limited dynamics, it is possible to consider use of models that can adjust the connectivity of a system based on its criticality and known business needs. Such a process would eliminate or limit connectivity that does not address some mission or business need. However, the current drive to create smart systems relies on increased connectivity and information fusion; thus, security professionals' desire to limit connectivity will constantly be in tension with the potential for improved cost-effectiveness that additional connectivity will enable.

The deployment strategies used for CPS present several possibilities for novel protection strategies. CPS are often highly distributed and provide multiple observations of the same, or highly related, phenomena. This multiplicity could be used to devise new means of providing data integrity by leveraging the multiple viewpoints. Although the challenges associated with upgrading legacy CPS are discussed above in section B.4.2.1.4, the addition of new systems into the legacy environment also provides opportunities. The new components can monitor or protect their older comrades, or serve as wrappers that enable the old technology to participate in new protection strategies. As more "smarts," processing power, capability, and control move to the system edges, additional protection nodes can be added that are robust enough to protect themselves and the system of which they are a part.

The fact that many CPS are safety-critical systems also provides some opportunity for improved cybersecurity. Systems that often undergo rigorous analysis for safety and cybersecurity may be

able to leverage those analyses in the context of threat models to devise protections. Some of the safety controls already in place in CPS can mitigate the effects of some types of cyber-attack, thus providing mechanical and non-cyber solutions to cybersecurity problems. To provide required reliability, safety-critical systems are also often designed with redundancy, which cybersecurity engineers can leverage to provide resilience. In contrast, low power systems are often not designed to provide either reliability or resilience. This opens the door for resilience strategies that rely on redundancy in infrastructure rather than at the endpoints, but it may be challenging to design and implement such strategies.

Identifying the specific properties of CPS that are different from those of IT systems can help system designers and cybersecurity professionals tailor existing cybersecurity and privacy solutions or identify new ones that are well suited to this domain.

B.4.2.4 The design response

The special characteristics of CPS must be considered when designing and developing secure CPS. Trustworthy CPS architectures must be based on a detailed understanding of the physical properties and constraints of the system. Analysis in support of design activities must include creation and simulation of up-to-date adversary models. These activities should be based upon principles in four key areas:

- Threats to Resilience
    - Threat vectors cannot all be known, so system design should incorporate diverse methods for gaining and maintaining awareness and adapting the system's configuration to transform system behaviors as needed.
    - Cyber threats are co-adaptive and intelligent, requiring constantly evolving methodologies to predict the nature of potential threats, their specific objectives, and their specific effects on each system.
    - It is difficult to assign probabilities of behaviors to complex systems that include technology and human interactions (ability for multiple controls to fail, impact of falsified indicators, etc.) We must also address the fact that human-in-the-loop systems may not be capable of sufficiently timely response for some threats, but those humans should still be kept aware of the system's state.
    - The supply chain for digital technology is global, complex, and not under any single organization's control. Adversaries may have many opportunities to compromise CPS.
    - Increased design obfuscation provides resilience to a malicious man-made threat, but increasing complexity adds brittleness (reduces resilience) to non-malicious, man-made, and natural threats.
- Cyber-Physical
    - Minimize shared interdependencies between networked systems to reduce potential for unrecognized interactions that lead to cascading failures.

    - Localize process and security dynamics of system operation to stabilize any cyber-physical disturbance and prevent cascading failures.
- Cognitive
    - Target and tailor CPS information to the benign human-based role and expected action to ensure a reproducible response irrespective of background or experience.
    - Gain cognitive understanding of expected operational behavior based on a thorough understanding of the difference between normal or intended patterns of behavior and those that are not.
    - Obfuscate and abstract CPS information available to or accessible by the potentially malicious human, integrating active transformation of system behaviors in response to a perceived threat.
- Cyber-Physical-Cognitive
    - CPS cyber-physical degradation must be quantified within the distributed architecture to ensure an effective response and that effects from threats are localized.
    - Concepts of trust are not granted, but they are earned, based upon conformance metrics associated with the design, with the highest degree of correlation associated with the highest consequence.

Adaptive capacity for the cyber resilience that will prevent serious consequences can take many forms, including adding new physical controls, such as manual valves; removing non-essential alternatives to remote operation that can directly lead to poor consequences if used inappropriately; or requiring a two-person rule when accessing a location to prevent insider threats. The desired process includes holistic consideration of the cognitive, cyber-physical aspects that provide barriers to malicious operation, but do not necessarily impede effective, benign operation. When considering protection capabilities, designers and system owners must understand the cost-effectiveness associated with the implementation of advanced, diverse analytics technologies that include physical indicators of cyber anomalies, manually-implemented interlocks, and other features that may not be characteristically cyber-based but provide operability assurance and degradation protection in the face of CPS compromise.

There is also a need to design proactive, real-time, autonomic algorithms and architectures that can defend dynamically against changing adversary models. Incorporating dynamic models of the systems to be controlled can help increase understanding of the impacts of attacks and leverage this understanding to reason about what the attacker might do should he or she gain access. To address privacy protection, CPS owners and operators need purpose-aware collection of data, which enables system owners to collect only what is needed, at intervals that are tuned to the needs of the application. System designers should explicitly consider privacy risk and trade operational gain versus privacy loss when creating their designs.

B.4.3 Moving from Classic Cybersecurity Properties to Cross-Property Risk Management[17]

This section defines the top-level properties of systems that risk managers should consider when performing risk management and explains their relevance to CPS.

B.4.3.1 Trustworthiness Concerns

The top-level Trustworthiness management properties are:

- **Cybersecurity (or security):** A condition that results from the establishment and maintenance of protective measures that enable a system to perform its mission or critical functions despite risks posed by threats to its use. Protection measures may involve a combination of deterrence, avoidance, prevention, detection, recovery, and correction that should form part of the enterprise's risk management approach [CNSSI 4009].
- **Privacy:** A condition that results from the establishment and maintenance of a collection of methods to support the mitigation of risks to individuals arising from the processing of their personal information within or among systems or through the manipulation of physical environments. Risk mitigation controls may involve a combination of administrative, policy, and technical measures directed at maintaining individuals' autonomy and their physical, financial, and psychological well-being.
- **Safety:** The absence of catastrophic consequences on the user(s) and freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment [81].
- **Reliability**: The ability to provide a consistent level of service to end users [82] or continuity of correct service [81].
- **Resilience:** The ability to prepare for and adapt to changing conditions and withstand and recover rapidly from disruptions. Resilience includes the ability to withstand and recover from deliberate attacks, accidents, or naturally occurring threats or incidents [83].

Given the scope of CPS, traditional enterprise IT approaches and solutions cannot adequately address the relevant cybersecurity, safety, and privacy needs. CPS owners and operators may need to consider additional risk management properties. These will vary based on system functionality and operational needs. The Working Group's analysis of illustrative examples led

[17] For the purposes of this document, risk managers address both risk and uncertainty, as defined in Section B.4.2.1.1, so the term "risk management" covers both sets of activities.

to the conclusion that the above five properties applied most broadly across the diverse breadth of CPS.

B.4.3.2 Cross-property nature of the threat

CPS owners and operators, who have traditionally been concerned with system risk in terms of safety, reliability, resilience, physical security, and privacy, have good reason to also be concerned about cybersecurity. Users need systems that will behave as expected, even under stress due to attacks [65]. Confidence that the system will perform as expected is especially critical to CPS because they have the potential to cause harmful effects in the physical world. To gain that confidence, an integrated risk management approach is needed that considers cybersecurity, safety, reliability, resilience, and privacy. The case of the Stuxnet worm [66] illustrates the importance of cross-property risk analysis for CPS:

> [Stuxnet] was a 500-kilobyte computer worm that infected the software of at least 14 industrial sites in Iran, including a uranium-enrichment plant. It targeted Microsoft Windows machines and networks, repeatedly replicating itself. Then, it sought out Siemens Step7 software, which is also Windows-based and used to program industrial control systems that operate equipment, such as centrifuges. Finally, it compromised the programmable logic controllers.
>
> The key compromise was that Stuxnet placed itself in a critical path where it could not only disrupt the plant process, but also disrupt/manipulate the information flow to the system operator. In this particular instance of Stuxnet, it caused the fast-spinning centrifuges to tear themselves apart, while fabricating monitoring signals to the human operators at the plant to indicate processes were functioning normally.
>
> Stuxnet could spread stealthily between computers running Windows—even those not connected to the Internet [via infected USB drives]. It exploits vulnerabilities associated with privilege escalation, designed to gain system-level privileges even when computers have been thoroughly locked down. That malware is now out in the public spaces and can be reverse engineered and used again against CPS.
>
> *Excerpted from [66]*

Stuxnet used the cyber interface to the target system to impact its physical operation and cause safety and reliability concerns. In concept, malware with capabilities similar to those displayed by Stuxnet could maliciously alter the operational state of any CPS by compromising cyber subsystems (e.g., digital data feeds from sensors, digital files used by cybernetic control systems to control machine operation, and digital data storage used to record system state information) in ways that adversely affect safety, reliability, resilience, privacy, and financial bottom lines. Such malware could also collect and exfiltrate intellectual capital that could inform attackers' future attempts to threaten system performance. Managing risk associated with CPS

cybersecurity, therefore, requires consideration of these properties along with classic IT security concerns.

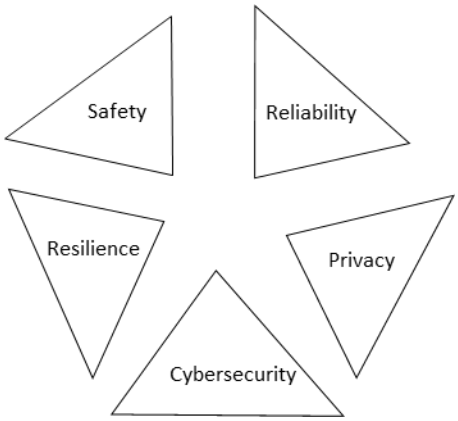


Figure 23: Evolution of Systems Design Property Silos

The properties of safety, reliability, privacy, cybersecurity, and resilience have, for the most part, evolved within distinct silos (see Figure 23). Historically, systems design has occurred within disparate disciplines. Large systems engineering and integration projects often have property-specific leads, who represent discrete viewpoints within the trade-off process overseen by the chief systems engineer/integrator. Functional requirements often have caused engineers and designers to prioritize each property differently, based on domain-specific (energy, manufacturing, transportation, etc.) requirements and perspectives, but achieving a certain level of success in each property typically is vital to the overall success of the system. Likewise, risk management activities have often been conducted within each silo, rather than across them. As the prior entries in this aspect make clear, the future of CPS design, integration, and risk management, however, appears to be evolving toward a multi-disciplinary approach where systems designers and integrators will increasingly be required to work across properties, with the growing imperative to provide cybersecurity becoming a common requirement for all. Ideally, personnel responsible for each property will consider the interdependencies among all five properties throughout the system lifecycle.

Stuxnet illustrates how the continuing integration of cyber technology into traditional systems is breaking down silo walls. “Cyber technology” exploited by Stuxnet included the data interfaces, digital data pathways, and digital sensors used to compromise the PLCs associated with centrifuge control. Machines built with locally isolated controls were “connected” by a USB interface designed to offer greater convenience to workers. The interface unwittingly permitted transfer of cyber-attack payloads across an air gap. The operational systems used to deliver services in many critical infrastructure sectors and in plants that manufacture goods, including national security systems, use similar configurations.

Stuxnet's principal objective appears to have been to cause physical damage to centrifuges. Its developers determined that a cyber payload could use digital data to manipulate the mechanical and digital components of the centrifuge system such that the centrifuges would damage or destroy themselves. Having designed the payload, the individuals behind Stuxnet only needed a way around the cyber protections to achieve harmful effects that were typically the concern of other risk management properties. Stuxnet used the cyber interface to effectively overcome the safety, reliability, privacy, security, and resilience provisions of the target systems.

Figure 24: Recommended Interdisciplinary Design Approach to CPS Engineering

Industry trends suggest that discrete systems engineering disciplines are converging toward increased interdependency [68] as illustrated in Figure 24. This is particularly important for CPS, in which systems-based holistic thinking will be critical to supporting objectives such as safety, reliability, resilience, privacy, and security. The relative importance and interaction of the various risk-related properties must be considered so that problems arising with respect to one property, or protections inserted to address one dimension of concern, do not compromise other primary system objectives or cause deleterious unintended effects. An interdisciplinary approach to systems design and integration is, therefore, required to establish an overall SoS design objective and support appropriate trade-offs in the service of that objective, if possible.

Because earlier CPS were custom-designed over time and mostly isolated, it was believed there were few common processes or software systems by which a cybersecurity incident could affect CPS, let alone spread through multiple systems. Due to the implementation of commonly used software and communication protocols, increasing interconnections between different systems, and connection to the Internet, CPS cybersecurity is becoming increasingly important

to CPS owners and operators. We have already seen that cyber-attacks can now affect CPS operations in a variety of ways, some with potentially significant adverse effects.

The development of trustworthy [69], networked CPS requires a deep understanding of potential impacts resulting from intentional and unintentional cyber-attacks or incidents on both the cyber and the physical aspects of the system. While the operational conditions embodied in such attacks and incidents exist in the domain of resilience engineering, their potential impacts span all of the risk-related properties. So, too, must efforts to plan, prepare, respond, mitigate, and recover.

B.4.3.3 The need for cross-property risk analysis for CPS

By their nature, CPS are subject to physical, cyber, and hybrid (cyber-physical[18]) attacks. These attacks seek to use one or more compromised cyber subsystems of the CPS to interrupt or damage the physical object (or physical process) of control. As the Stuxnet example shows, such attacks can have devastating cyber, business, and physical effects. Corporate executives, risk managers, and CPS operators must understand how different subsystems (cyber and physical) work and how they interface with each other. This understanding will help them select and implement appropriate security controls and algorithms; make meaningful risk evaluation decisions; create and execute useful risk mitigation strategies; and recognize ongoing attacks. This section shows that, as with systems engineering, it is critical that risk management planning and operations be conducted holistically, rather than within discipline-specific silos.

Modern systems, including CPS, often include physical, analog and cyber elements[19]. Cyber components are proliferating because they often provide a favorable combination of lifecycle cost, capability, supportability, and in many cases flexibility. But use of cyber components, especially in CPS that demand high reliability, resilience, and safety (and, therefore, high levels of cybersecurity, privacy protection, and trustworthiness) also presents a significant challenge. Unlike the behaviors of physical and analog components, which can be subjected to rigorous tests with results obtained via direct observation, the potential behaviors of cyber components can be difficult and costly to test in the wide variety of configurations and operating conditions to which they may be subjected. This means that cyber components are likely to consume a far greater share of the overall "system risk budget" than either analog or physical components—a

[18] http://www.utdallas.edu/~alvaro.cardenas/papers/NordSec2013.pdf

[19] Physical elements rely on materials sciences and well-understood physical properties; analog elements rely on mechanical, electrical, and kinetic principles, often in sensor systems that convert physical data (pressure, rotational speed, voltage, etc.) into data which can be represented digitally and vice versa; cyber components rely on logical, mathematical and computational principles and constructs. Physical and analog components are both represented within the physical layer of the Framework.

fact that engineers, owners, and even operators may need to recognize when assessing risk and developing mitigation plans.

Privacy represents a particular challenge as the field currently lacks common terminology to describe privacy risk and objectives that can facilitate system design and risk mitigation control selections. Organizations have been attempting to use principle-based approaches like the Fair Information Practice Principles or Privacy by Design to address privacy in information systems. These principles have helped organizations consider various aspects of handling personal information, but they have not offered organizations a consistent, repeatable methodology for understanding how to identify specific privacy risks across different systems and in the face of rapidly evolving technologies. As a result, organizations like NIST, MITRE, and others have begun to research and develop methodologies for expanding understanding of privacy risk management and privacy engineering.[20]

An objective of CPS is to achieve optimum behavior through the correct allocation of requirements to each of the three elements (physical, analog and cyber elements) through a process of co-design. "Optimum" in this context involves determination of the desired balance point for cost, benefit, and risk. Systems designers and integrators often assign a 'risk budget' to manage the degree of allowable impact measures taken to ensure security, safety, reliability, privacy and resilience may have on system performance. With the co-design of risk-relevant properties, this budget should not be meted out with a separate share to each concern, but viewed as a common resource on which each property can draw. System designers must develop a risk model that indicates the level of protection required for each of the properties and must determine in which portions of the system protections are best provided. Since this budget is fixed, designers need to determine the allocation that best achieves the overall objective. Tradeoffs will be required if the budget is not adequate to address all concerns. Obviously, determination of specific priorities will be situation-dependent and the risk budget need not be apportioned equally.

[20] MITRE Privacy Engineering: http://www.mitre.org/publications/technical-papers/privacy-engineering-framework

NIST Privacy Engineering: http://csrc.nist.gov/projects/privacy_engineering/index.html

Centre for Information Policy Leadership Privacy Risk Framework: http://www.informationpolicycentre.com/

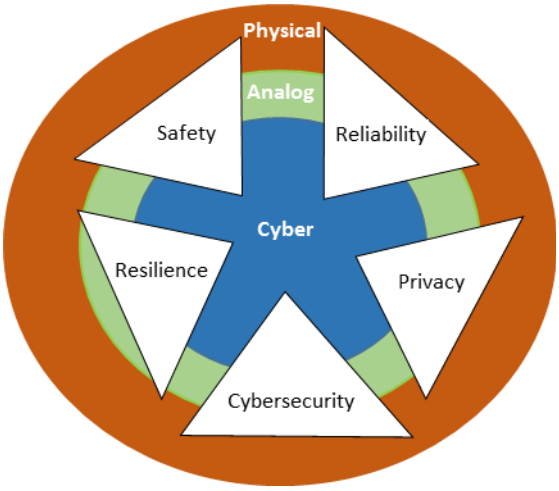


Figure 25: Physical, Analog, and Cyber Components of CPS

As Figure 25 illustrates, CPS may include physical, analog, and cyber components. When considering solutions involving cyber, physical, or analog components, engineers must determine how to evaluate the effect of their choices in terms of multi-level trade-off metrics. In simplistic terms, security considers operational and reputational risk, safety considers error rates, reliability considers failure rates, privacy considers unwanted disclosure rates, and resilience considers recovery rates. The complexity, interconnectivity, and dynamism typical of CPS argue for a more holistic process that spans all risk-relevant properties and types of components.

B.4.3.4 Cybersecurity as a CPS risk management property

It is interesting to consider how cybersecurity interacts with the other risk-relevant properties to provide confidence that the system will work as expected in the face of changing internal and external conditions, including threats that may cause faults or threaten the security of critical data. By adding cyber components to systems, we are introducing new loci of faults and new vectors of threat, as well as a more complex environment. This leads to new challenges in providing safety, resilience, reliability and privacy. However, by adding a cyber component to the system and considering cybersecurity as an integral part of that component, we are also adding a new locus of protections and protection mechanisms (“smarts”) that could not be instantiated in the physical domain.

Cybersecurity is a key component of achieving privacy, but cybersecurity breaches are not the only cause of privacy intrusions. Therefore, cybersecurity risk management programs alone will not sufficiently account for privacy risk. Cybersecurity risk assessments require an analysis for potential exploitation of vulnerabilities by a malicious actor. In contrast, understanding privacy risk requires organizations to account for adverse impacts on individuals that may be the by-product of system operations that involve the processing of personal information. Indeed, the

very implementation of cybersecurity measures can create risks to privacy (e.g., log retention, system monitoring). A key aspect of CPS design will be the application of controls that balance the achievement of positive outcomes among the five properties.

Safety and resilience may be the area's most critically affected by the addition of a cyber component to the system. Safety is the absence of catastrophic consequences on the user(s) and the environment [70]. The primary focus of any system safety program is to implement a comprehensive process to systematically predict or identify the operational behavior of each safety-critical failure condition, fault condition, or human error that could lead to a hazard and potential mishap. This process is used to influence requirements that drive control strategies and safety attributes in the form of safety design features or safety devices to prevent, eliminate, and mitigate unsafe conditions and behaviors. The cyber component greatly increases the complexity of the set of possible behaviors and so greatly complicates this analysis. Modern system safety processes are comprehensive. They are risk-based, requirements-based, function-based, and criteria-based. They include specific objectives aimed at producing engineering evidence to verify whether safety functionality will always function as intended and provides acceptable risk in the actual operating environment. However, safety system analyses must evolve to consider the design threats posed by cyber vulnerabilities and modern cyber-attack techniques.

Cyber components that command, control, and monitor the safety-critical functions of physical systems require extensive system/software safety analyses to inform detail design requirements, especially in relatively autonomous or robotic systems that require little or no operator intervention. Cybersecurity capabilities must accommodate system complexity, and system designers and engineers must consider cybersecurity principles that support separation of functions and assured composition.

The safety of a CPS also depends on its resilience, which includes fault tolerance, ability to degrade gracefully, and pre-defined fail-safe states (and the triggers for each state). Resilience gives a system "tolerance to degraded and failed conditions that permits continued performance of all or at least critical functions [72]." In the event of significant system failure that could compromise safety, a resilient system must provide a highly reliable way to achieve pre-defined fail-safe status. Alternatively, the system may reconfigure process streams and control parameters to meet new functional objectives, including establishing new operational priorities such as shutting down low-priority processes in order to direct remaining resources to higher-priority ones (graceful degradation). Cybersecurity protections can also support the identification of the more critical portions of the system and the processes it supports, and provide additional protections to those system components and processes.

We have already noted that system reliability can be a critical requirement of CPS. An unreliable CPS can produce malfunctions in the greater systems of systems, service disruptions, poor-quality products, financial losses, damage critical equipment beyond repair and even endanger human life and the environment. Each component (and component system) of the

CPS must provide a sufficiently low failure rate to enable the CPS as a whole to achieve sufficient aggregate system-level reliability. Resilience gained through redundancy and synchronization (fault-tolerant approach) among different CPS components, in combination with high-confidence detection of failures, are the major means used to provide required level of reliability and availability of a system [73]. Cybersecurity practices and mechanisms can be used to provide software assurance and to improve failure detection.

As discussed in Section B.4.2.1.1, reliability shares goals with cybersecurity. The major difference is that reliability has traditionally addressed expected potential issues. Resilience, on the other hand, is impacted by cybersecurity that aims first to protect against and then to mitigate the effects of unexpected disruptions caused by attacks that may target:

- **System and data availability**—the ability to provide required functions/data (including control functions, specifications and state indicators)
- **System and data integrity**—the ability to execute the correct instructions using the correct data at the correct time. It is important to recognize that attacking the cyber subsystem can disrupt proper functioning of the physical subsystem(s) of the CPS or cause the system to function in accordance with an improper set of instructions
- **Data confidentiality**—the ability to protect system data (including internal programs) from disclosure to unauthorized individuals or use of data for unauthorized purposes

Traditionally, reliability mechanisms concentrate on detection, protection, and mitigation of CPS component failures (fault tolerance) in a predicted set of operational conditions. Cybersecurity, on the other hand, concentrates on detection, prevention, and mitigation of attacks and compromises (threat tolerance), the full extent of which cannot be predicted. Enabling the seamless convergence of reliability and cybersecurity will help provide CPS resilience and the required level of safety.

B.4.3.5 CPS trends and risk analysis

Traditional IT cybersecurity provides information protection (integrity, confidentiality) and readiness for correct services (availability). CPS cybersecurity has the same goals as traditional IT cybersecurity--though perhaps with different priorities--but should also be focused on how to protect physical components from the results of cyber-attacks. Two challenges are typical for CPS cybersecurity:

- Detection and prevention of deception attacks (e.g., attacks on sensors that can lead them to input malicious data to the cyber component and, as a result, to provide wrong, or even dangerous, output from the cyber component)
- Detection of compromised cyber components and prevention of incorrect cyber functioning (or failure to function)

These challenges are not unique to CPS; rather, their consequences are potentially more severe because they impact the physical world. More importantly, the means to prevent these problems include not only cybersecurity controls, but also safety and reliability controls that are not applicable to IT systems.

Thus, CPS cybersecurity requirements should be determined in conjunction with safety, reliability, and privacy requirements. In its turn, CPS resilience should provide ways and means to continue not just IT services, but also critical CPS operations in case of a failure or a cyber-attack, ideally with full CPS recovery. This can be done only through co-design of CPS cybersecurity, including privacy, with safety, reliability, and resilience. As a result, consideration of the traditional tenets of confidentiality, integrity, and availability is no longer the sole focus of cybersecurity for CPS. Nor is providing CPS cybersecurity simply a matter of prioritization and application of existing controls. Rather, it involves the tradeoff of risks. This process of risk management becomes even more critical when considering the potential impact of cybersecurity failures on the ability to deliver capability across the disciplines.

In addition, to develop effective CPS cyber protection and mitigation actions, the nature, functions, and interactions of all three types of components of CPS – cyber, analog, and physical – must be understood.

CPS designers and integrators should consider both the intended and unintended effects resulting from the combination of properties where the goals of each may contradict or be complimentary to their counterparts. Trade-off decisions should be considered in light of the system-of-systems objective, if known. This is much more challenging than it sounds.

Figure 26: CPS Risk Properties

A SoS design or integration approach for CPS may benefit from 'risk model' analysis that considers both the impact to each system objective individually and the SoS objective as a

whole; see Figure 26. For example, a SoS with the highest priority goal being to deliver safety should have a risk model that favors safety. Risk models may also aid in placing emphasis on the most appropriate type of component—physical, analog, or cyber. System risk analysis may provide helpful context when considering how best to apply desired CPS risk-related properties. While their specific equities and priorities may be different, CPS owners and operators should use a similar process when evaluating risk in operational situations. This requires a detailed understating of the strengths and weaknesses of the system in place, the role of each architectural layer, and the interactions among the layers.

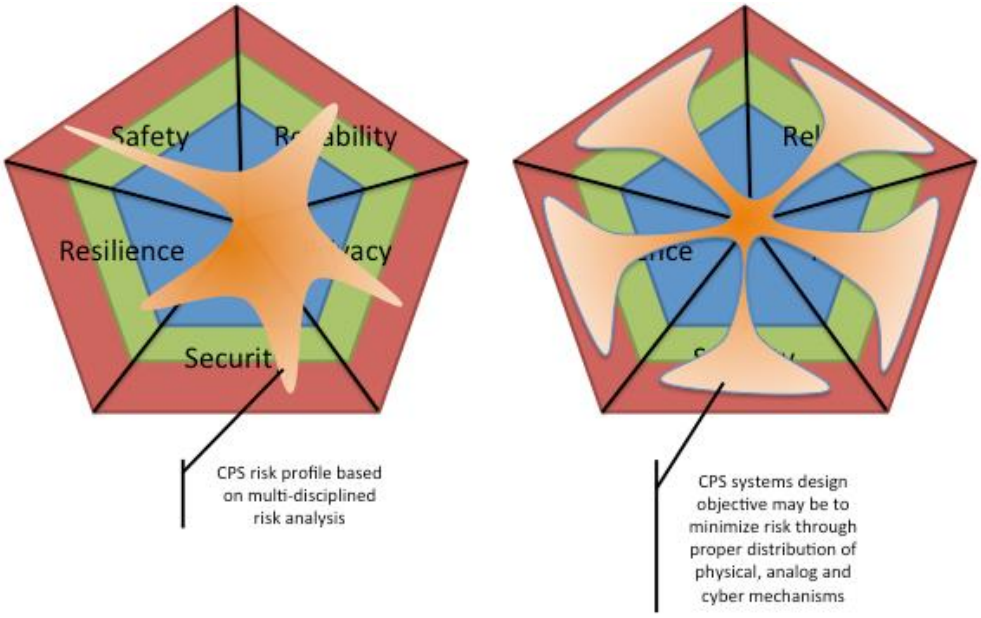


Figure 27: Applying Risk Analysis to CPS

It is useful to look at a few illustrative examples of risk models. Figure 27 shows two conceptual examples of risk models. The shape of the risk "blob" represents where risks are incurred (the figure on the left) and where risk may be assigned within the risk budget. These assignments must be based upon thorough knowledge of the nature and relevant capabilities or tolerance of each type of component. The next section introduces high-level examples of such a process, which is necessarily implementation-dependent, and describes the relationship among the risk-related properties in a number of example systems.

It is important to recognize that the different types of components have different tolerances for the various sources and impacts of risk. This fact can help risk managers devise designs that limit risk. In the Stuxnet example, for instance, if the centrifuges had been equipped with physical or analog controls that physically prevented the centrifuges from spinning faster than their design limit no matter what the digital system commanded, the cyber-attack would have failed.

B.4.3.5.1 Implanted medical device

An implanted medical device has high requirements for safety because incorrect operation could cause direct harm to patients. It also has high reliability requirements because the patient's welfare depends on the continued predictable operation of the device. Privacy requirements may be important; patients have a legitimate concern that their health metrics remain private, and in this example it is assumed that there is personally identifiable information (PII) associated with the device – who the patient is, what the device is for, how it is configured. If a key piece of information is separated from that available from the device that is required for this to become personal—the unit number as it is related not to the name but to the Medical Record Number (ostensibly an obfuscated identifier and one that cannot be traced back to the patient reference). This becomes a risk only if a malefactor intends to directly implant false values in the system. Otherwise, any implanted device with wireless capability could be compromised. This brings to light that there are high requirements for cybersecurity protections on the command and control paths of implanted devices, but probably lesser requirements on their reporting paths (unless they can provide access to the command and control paths). In fact, the privacy requirements might more than cover the cybersecurity requirements on the data reporting paths. Given the high reliability requirement, one might think resilience is critical, but the small size and low power typical of implanted devices make the usual methods for providing resilience (e.g., redundancy, fail over) impractical and lead us to think about alternative strategies such as frequent monitoring, scheduled replacement, or early detection of degradation.

B.4.3.5.2 Chemical manufacturing plant

A chemical manufacturing plant has high requirements for safety that refer to two concerns. One is process safety itself, to prevent unwanted or uncontrolled chemical reactions. The other is equipment safety, which seeks to prevent equipment failure or damage. An example would be preventing pressure in the reactor from exceeding safety limits to stave off reactor burst [77]. Today, more than 100 million Americans live close enough to one of the more than 470 chemical facilities across the country that they could be at risk if there were a deliberate or accidental release of chemicals at those sites [78]. Safety of chemical plants relies on reliability and security. High reliability can compensate for possible failures by minimizing defects and using one or more alternative control structures in parallel. But in case of cyber-attacks, such as integrity attacks (sensor manipulation attacks), denial of service (DoS) attacks, and attacks on situational awareness (attack on a Human Machine Interface console), only cybersecurity can provide the necessary detection and protection. Control processes must be highly resilient because they inherently require high reliability and strong cybersecurity protection. Resilience provided by improving the length of time the process can withstand an attack can give operators the time they need to intervene. Privacy requirements are low, because there is likely to be little personally identifiable information associated with the chemical process or plant equipment.

B.4.3.5.3 Wearable computing and IoT

Wearable computing is the use of a miniature, body-borne computer or sensory device worn on, over, under or integrated within, clothing. Constant interaction between the user and the computer, where the computer "learns" what the user is experiencing at the time he or she is experiencing it and superimposes on that experience additional information, is an objective of current wearable computing design [79]. According to a 2013 market research report [80], there are four main segments in the wearable technology marketplace:

- Fitness, wellness, and life tracking applications (e.g., smart clothing and smart sports glasses, activity monitors, sleep sensors) which are gaining popular appeal for those inclined to track many aspects of their lives
- Infotainment (e.g., smart watches, augmented reality headsets, smart glasses)
- Healthcare and medical (e.g., continuous glucose monitors, wearable biosensor patches)
- Industrial, police and military (e.g., hand worn terminals, body-mounted cameras, augmented reality headsets)

Security and privacy issues should be considered very seriously, as wearable devices work through an IoT that deals not only with a huge amount of sensitive data (personal data, business data, etc.) but also has the power to affect the physical environment through its control abilities. CPS like these must, therefore, be protected from different kind of malicious attacks. Security, privacy, resilience and safety requirements depend on the particular application. For example, fitness tracking applications have low requirements for risk-related CPS properties, but may have significant privacy risks. Police or military applications should have high safety, security, and resilience requirements based on their mission.

B.4.3.6 Recommended Next Steps

CPS that address a more complete set of tenets will be more complete and hence will present less risk to the greater system-of-systems envisioned by concepts like the IoT. Safe, reliable, or resilient systems that lack attention to security or privacy may increase these risks when connected to other systems with a primary objective of security or privacy. CPS cybersecurity is concerned with managing risk for the entire SoS as well as for sub-systems. Development of a common approach to cybersecurity design, integration, and operation is an important next step. In particular, CPS designers need to consider the following when addressing cybersecurity controls:

1. Proactive mechanisms in sensor network security have focused on integrity and availability from a communication network point of view. They have not considered how deception and DoS attacks affect the application layer service, i.e., how successful attacks affect estimation and control algorithms – and ultimately, how they affect the physical world. Novel robust control and estimation algorithms should be designed that

consider realistic attack models from a security point-of-view. These attack models should simulate deception and DoS attacks.

2. Cybersecurity controls have not considered algorithms for detecting deception attacks against estimation and control algorithms. In particular, previous detection of deception attacks launched by compromised sensor nodes assumes a large number of redundant sensors; they have not considered the dynamics of the physical system and how this model can be used to detect a compromised node. Furthermore, there has not been any detection algorithm to identify deception attacks launched by compromised controllers.
3. Many cybersecurity controls involve a human in the loop. Because CPS use autonomous, real-time decision-making algorithms for controlling the physical world, they introduce new challenges for the design and analysis of secure systems: a response by a human may impose time delays that may compromise the safety of the system. Therefore, autonomous and real-time detection and response algorithms should be designed for safety-critical applications.
4. CPS security should be defined with respect to an adversary model. Previous research has not studied rational adversary models against CPS. The field of automatic control is more mature in comparison to information security. However, despite great achievements in the field of nonlinear and hybrid systems theory, robust, adaptive, game-theoretic, and fault-tolerant control, much more needs to be done for design of secure control algorithms to ensure survivability of CPS.
5. In addition to the state of the system to be controlled, the state of the communication network should be jointly estimated. Approaches to estimate the indicators of performance and integrity of the communication network based on available network data should be developed. The estimated state of the network should be used to design transmission policies for sensors and actuators as well as scheduling policies for controllers to optimize performance.
6. Physical and analytical redundancies should be combined with security principles (e.g., diversity and separation of duty) to adapt or reschedule its operation during attacks. For example, under sensor faults or when only intermittent sensory information is available, the system should be able to operate using open-loop control for a sufficient amount of time.

A notion of trustworthiness should be associated with different components of CPS, and trust management schemes should be designed when the above redundancies are in place.

B.4.4 Safety

Safety concerns relate to the ability of the CPS to ensure the absence of catastrophic consequences on the life, health, property, or data of CPS stakeholders and the physical environment.

A more comprehensive treatment of this Concern is anticipated in the future.

B.4.5 Reliability

Reliability concerns relate to the ability of the CPS to deliver predictable performance in expected conditions.

A more comprehensive treatment of this Concern is anticipated in the future.

B.4.6 Resilience

Resilience concerns relate to the ability of the CPS to withstand instability, unexpected conditions, and gracefully return to predictable, but possibly degraded, performance.

A more comprehensive treatment of this Concern is anticipated in the future.

B.5 Data Aspect

This section describes the data aspect of the CPS Framework.

This section presents key topics about data interoperability from the CPS viewpoint. Each of these topics in turn has an overview to discuss the topic and an example of what the topic is about in order to give it some context. Then a section summarizes the critical dimensions of data interoperability and provides for a detailed discussion of data and metadata, identification, data quality and provenance, governance, privacy and cybersecurity, and verifiability and assurance.

This framework cites a significant number of references. However, the scope of data interoperability is broad, and a more exhaustive study could include many more substantive references. Further, there are mentions of specific references that are helpful in illustrating the concepts presented. However, these descriptions are intended to be exemplary rather than prescriptive.

It comprises the following sections:

- Section B.5.1, which provides an overview of the data aspect
- Section B.5.2, which discusses data interoperability topics from the CPS viewpoint
- Section B.5.3, which examines traditional data interoperability issues, and discusses the difference between data versus information models.

B.5.1 Overview

Data may be created, maintained, exchanged, and stored in many domains. Each datum has a lifecycle and can be moved among and stored at any number of systems and components. Each domain naturally defines its own data semantics and exchange protocols. But both humans and systems can find it difficult to process, understand, and manage data that has been moved across domains and ownership boundaries. In an ever more connected world, processing and

understanding data is a growing necessity. A CPS is a system of collaborating computing elements that monitor and control physical entities. Understanding data exchanged among independent computing elements is as much, if not more, important than it is in other data management domains. Note that this section only focuses on communications among computing elements via data. Communications may occur among CPS elements via changes of state of an environment and observations of those changes, through what is referred in the literature as stigmergic channels [85]. Such channels are not part of the data aspect, and thus are not in scope for this section.

Data of concern to CPS components fall into two major categories: archival data and real-time data. Archival data informs about observations of the CPS and its environment taken at a past instant for archival purposes. Real-time data is used to control the environment and has a limited temporal validity—think of the state of a traffic light. The progression of time can invalidate real-time data. A valid real-time data element that is put into a queue might be invalid when taken out of the queue.

CPS components collect, process, share, and examine data to provide actionable inputs to other CPS components. Data are acquired, shared, and examined at multiple levels within scales. A *scale* is a spatial, temporal, quantitative, or analytical dimension used to measure and examine the data. A *level* is a unit of analysis on a scale. For example, temporal scale can be thought of as divided into different levels (timeframes) related to rates, durations, or frequencies.

The dynamics of cross-scale and cross-level interactions are affected by the interactions among collaborating computing elements and entities at multiple levels and scales. Addressing these complexity issues in an efficient and effective manner will require new approaches to managing data integration, and all boundaries (ownership, scales, and levels) need to be more widely understood and used.

The challenges of data integration complexity and CPS boundaries include:

- Data fusion (see section B.5.2.1) that is done at any time from multiple sensor or source types, or use of a single data stream for diverse purposes
- Data fusion of streaming data and predictive analytics capabilities
- Complex data paths that cross-scale and cross-level connecting architectural layers, dedicated systems, connected infrastructure, systems of systems, and networks
- Data-driven interactions between dependent and independent CPS
- Privacy-protecting data policies and procedures in light of the ubiquitous nature of IoT
- Data interoperability issues including metadata, identification of type and instance, data quality and provenance, timing, governance, and privacy and cybersecurity

The goal of this data aspect is to provide a sound underlying description and standards base that simplifies and streamlines the task of understanding cross-domain data interactions.

B.5.1.1 What is Data interoperability?

The concept of *data interoperability* involves how and to what extent systems and devices can exchange and interpret data. It assumes a requirement to understand the exchanged data to realize the intended benefits of the exchange. The *dimensions of data interoperability* describe the extent to which exchanged data can be understood. Note that data interoperability is but a subset of all dimensions of interoperability necessary to establish an interoperable architecture of exchange. However, this section focuses only on the data dimensions – syntactical, semantic, and contextual.

- *Syntactical interoperability* defines the structure or format of data exchange, where there is uniform movement of data from one system to another such that the purpose and meaning of the data is preserved and unaltered. Syntactical interoperability defines the syntax of the data – organization of the bits and bytes – and certain structural descriptions of intermediate processing such as processing for storage, describing what data is provided, data descriptions, and pipelining. It ensures that data exchanges between systems can be interpreted at the individual data field level.
- *Semantic interoperability* provides the ability of two or more information systems or elements to exchange information and to enable the use of the information that has been exchanged, processed, interpreted, or otherwise used, independent of the syntax by which it was exchanged. Semantic interoperability is about a shared, common interpretation of data. This degree of interoperability supports the exchange and other operations on data among authorized parties via potentially dependent and independent systems, if required. The semantics include metadata about the data such as the relationship of timing to instances of data.
- *Contextual interoperability* includes business rules about the validation and authorization of data. As with any interaction between systems, the data exchanged will be driven by how the data are used. The content and format of data exchanges is driven by the intended purpose of the exchange—specifically, where, when, how, and why the receiving system will use the exchanged data.

In addition to physical connectivity that permits data movement, use of data across disparate systems often requires translation of data objects from the syntax of the sender's data into a form that is compatible with the receiver's syntax. For systems that require integration, the exchange of data between systems is done through data models and data objects that describe the data semantics. The receiving system must understand the context, for example metadata that describe the nature and constraints on the data, in which the data were created to properly apply the semantics to its purpose.

In practice, data exchange requires the interoperability framework to encompass the physical connection of sensors and system components accounting for transmission of data through

various interfaces. These data are then processed through system software data ingest functions according to specified rules and procedures.

B.5.1.2 Canonical models and adaptors

Many CPS are composed, at least in part, of legacy components and data implementations. These legacy components may not implement current best practices and protocols. A descriptive semantic model relies on the data types and the relationships between the data types within a given data model. Redesigning applications to use a given semantic model may not be straightforward or even feasible. This means that the source system's data model must be transformed into each destination system's data model for integration.

A set of common canonical data models that can be mapped to a set of disparate semantic models can reduce complexity in these cases. The models can be maintained for critical systems within each infrastructure and, at the highest level, between infrastructures. The use of common canonical models reduces the number of transformations between systems required from "n(n-1)" to "n" (where n is the number of disparate systems that must ultimately exchange data), because in the more complex case, each pairwise exchange domain must have its own bilateral transformation.

Data related to time, privacy, and security are also important within the context of data exchanges between applications. The integration of time-series data should express time information in a manner that can be traceable to an international time scale, including drift. This is similar to how GNSS can be used for geo-level data integration to enable consistent understanding across system boundaries. Privacy, security, and authentication data are also essential to the contextual understanding of information because they embody essential trustworthiness requirements.

Adaptors can minimize the impact on cost and complexity of interoperability achieved. In traversing many network segments and protocols, a standard interface can be inserted at any point in the data flow, rendering data upstream from it "interoperable" per the canonical model.

The higher degrees of interoperability achieved have implications for reducing the complexity of the data exchange and use. Data exchange adaptors between systems should be strategically located for maximum effect and minimum cost. This will reduce the risk to these systems as they evolve and expand.

B.5.1.3 CPS data interoperability and SoS

A CPS is a cyber-physical system, and every system must have clearly identified boundaries. When data crosses a system's boundary, it may flow to another system. The movement of data may be to an *actor* (e.g., person, component, device or system) that (by definition) closely involved with the operation of the CPS, or it may be to an actor having no direct connection to

the original one. From the perspective of the first CPS, some systems may appear to passively consume data. When other systems exist outside the CPS boundary, it is possible that a collection of such systems could interact, with new behaviors emerging from this interaction. In this way, the original CPS may become part of a SoS CPS. This is an example of composability of CPS. Whether or not the CPS interacts at this scale may be of little or no import to the individual CPS. Ideally, well-crafted interfaces from the CPS to other systems will permit the circulation of data among systems, while limiting data use to authorized users and purposes. From the data interoperability perspective, the challenge lies in the design of the CPS data interface. The focus of this subsection is to raise data interoperability issues and discuss how they may be addressed in practice. These issues include:

- The identity of the sender
- The identity of the data
- The integrity of the data
- The time sensitivity of the data
- The semantic meaning (including context) of the data
- The authorization to acquire and use the data (for specified purposes)

Whether a particular CPS is able to interact with other systems to become part of a SoS is perhaps a test of the quality of the handling of these issues. When a CPS is designed, it may be expected to occupy a particular position in a large and well-defined ecosystem. Or, it may be part of a small collection of systems, or even stand alone. Ideally, such matters would be immaterial to the interface. However, interfaces that support exchanges among multiple stakeholder systems are difficult to realize. In information systems, the very nature of "identity" and "meaning" are usually arrived at by mutual agreement. There is no global authority to certify all identities and all semantic meaning for all applications. It is thus left to the technical community to arrive at useful solutions to some of these issues. These arrangements must be balanced by other practical concerns such as:

- The costs associated with communication (and thus the degree of implicit versus explicit semantic content)
- Safety concerns, and the risks associated with data errors to the application or other actors
- The extent and reliability of security required by the application
- The provision of version control and the support of newer/older versions of an interface

B.5.2 Data Interoperability Topics from the CPS Viewpoint

This section describes dimensions of data interoperability that are critically important in the evolution of CPS. The topics covered in this section are scenario-driven and address issues from the CPS point-of-view. The following section, B.5.3, approaches issues from a traditional data interoperability point-of-view.

B.5.2.1 Data fusion

Readers must recognize that researchers and practitioners have offered several strong definitions of the term *data fusion,* each of which is informed by their unique perspectives. The difference in perspective is driven by what matters to decision makers in the writer's domain of interest. However, the diversity of definitions can present important challenges to analysts, engineers and decision makers trying to design, develop, deploy and operate CPS that rely on data fusion capabilities.

For example, thinking from the perspective of military intelligence and operations, the US Department of Defense's Joint Director of Laboratories Workshop [102] defined data fusion as a "… multi-level process dealing with the association, correlation, combination of data and information from single and multiple sources to achieve refined position, identify estimates and complete and timely assessments of situations, threats and their significance." Outside of issues revolving around system trustworthiness (see Section B.4), it is reasonable to expect that most CPS applications will not be required to alert intelligence collectors or control weapon systems—but the needs, solutions and lessons learned from military and intelligence applications can be instructive.

Hall and Llinas [100] synthesized prior research to offer a more abstract definition of data fusion as "… techniques [that] combine data from multiple [sources], and related information from associated databases, to achieve improved accuracies and more specific inferences than could be achieved by the use of a single [source] alone." CPS designers, developers, and owners who are skilled at extrapolating from an abstract model to their own applications may find this definition more palatable. However, they may face challenges finding useful real world models that offer lessons that are both authoritative and immediately useful.

Taking a much narrower view for their Linked Data effort, Bizer, Heath and Berners-Lee [101] defined data fusion as "… the process of integrating multiple data items representing the same real-world object into a single, consistent, and clean representation." This definition appears to apply best to applications involving the requirement to resolve potential discrepancies between inputs from multiple data sources.

Approaching the problem from a very different direction, Castanedo [103] groups data fusion techniques into "three nonexclusive categories: (i) data association, (ii) state estimation, and (iii) decision fusion." Each category conveys its own requirements, attributes, constraints, and methods of application. As with the DoD characterization, the value of Castanedo's narrow definition for other types of CPS may lie in the ability to offer lessons learned from real-world applications that can be transferred to other domains.

To further illustrate the concept, the next few sections discuss several general applications of some of the different definitions of data fusion.

B.5.2.1.1 Data fusion from multiple sensor or source types or use of such data for diverse purposes

CPS are increasingly leveraging capabilities provided by improved sensors, processing techniques, and computing power to monitor, analyze (sometimes in near-real time), and control increasingly sophisticated systems and processes in domains as diverse as manufacturing, robotics, the operation of medical devices (both free-standing and implanted), environmental control, energy generation and distribution, and transportation. As the desire for additional data fusion grows, CPS users are likely to rely on data fusion in the sense of all of the definitions provided above.

Efforts to fuse data from multiple sources face significant data interoperability challenges. These challenges include, but are not limited to: identifying and resolving differences in vocabulary, context and semantic meaning; structuring of data (schema); attributing data to their source and maintaining an accurate "trail of provenance" (with attendant issues in identity management); resolving differences among different data formats; and detecting and resolving issues of accuracy versus timeliness.

An international standard, Recommendation ITU-T X.1255 [20], was approved in September 2013. The recommendation adopts a fundamental approach toward defining core concepts for purposes of interoperability across heterogeneous information systems. It describes a digital entity data model that provides a uniform means to represent metadata records as digital entities, and can also be used to represent other types of information as digital entities (whether also referred to as data, data item, data fusion, or other terminology). It is a logical model that allows for multiple forms of encoding and storage, and enables a single point of reference (i.e., the identifier) for many types of information that may be available in the Internet.

B.5.2.1.1.1 Example

A typical air traffic control system is a CPS that leverages data fusion. Each air traffic controller is the man-in-the-loop in a control system that directs aircraft to certain flight paths and altitudes at specific speeds. Controllers also advise pilots of potentially hazardous traffic and weather. The air traffic control system combines data from two types of sensors to provide an annotated image used by air traffic controllers to monitor and control the flight of thousands of aircraft a day. The first type of sensor is fixed-site surveillance radar. The surveillance radar provides bearing and slant range from a known point (the radar antenna's location) and can detect some forms of hazardous weather. The displayed aircraft geographic position (the "blip" or "primary return" on a radar screen) is a function of slant range and the known geographic location of the radar antenna. The second sensor is one of a pair of redundant "Identification Friend or Foe" (IFF) transponders on each aircraft. The transponder collects altitude data from the aircraft's flight instruments and combines this data with the aircraft's identification code, then transmits this data to a receiver mounted on top of the surveillance radar. The system that

displays the images on the controller's radar screen must merge and continuously update the primary and secondary data to present an accurate and integrated picture over time to enable controllers to help ensure proper routing and safe separation of aircraft from each other and possible hazards.

B.5.2.1.2 Data interoperability dimensions for data fusion

There is a need for a common interpretation of data to support the exchange of information. Data from today's CPS in various domains are collected separately; each domain exhibits its own data structure and may use different protocols. Data fusion techniques are needed if a user wishes to combine data from various systems.

Among the protocols that seek to help federate data, so that data from multiple sources can be acquired and fused, is OPC Unified Architecture[21] (OPC UA) [148]. Supervisory Control and Data Acquisition (SCADA) systems are examples that, when using OPC UA, combine the data into a common structured dataset accessible via web services. Software like Hadoop [149] enables distributed processing of large data sets across clusters of computers. However, obtaining and harmonizing the data can be a challenge due to the differences in format and variance in protocols. Identification is often also an issue in today's systems, as many systems may offer no identity other than a tag name, which may not provide the required level of assurance. While some modern systems tag data with Internet Protocol (IP) or Media Access Control (MAC) addresses, these are insufficient for a positive determination of device type, device owner, device operator, and device trustworthiness. Realistic projections indicate solutions to similar requirements must scale to trillions of devices.

CPS today are beginning to transition to a "semantic" form whereby metadata information can be used to describe the device and related information. This metadata can include guidance on how to handle the information. Also gaining popularity is use of identifiers that can be captured in the form of a Quick Response (QR) Code [150].

CPS have begun to use IPv6 and 6LoWPAN [151] to be able to capture sensor data and represent unique identifiers for the source of data. Widespread use of this identifier within CPS is a few years out, and faces considerable challenges using the IPv6 address as the primary identification. A client of the data must be configured to use the sensor device address to represent its identity. This has proven useful on a small scale (e.g., in smart phones and some sensor systems deployed in homes and buildings). However, obtaining the information across a

[21] The acronym OPC was borne from OLE (object linking and embedding) for Process Control was a legacy term carried forward by the standards group.

backhaul where there have been many local network segments using different protocols from Wi-Fi to Broadband over Power lines (BPL) remains an outstanding challenge.

Additionally, varied approaches to information exchange protocols exist (e.g., SOAP [152] and Representational State Transfer (REST) [153]). One is service oriented – SOAP, and the other data oriented – REST. Thus, a challenge still exists to move the information in a common format that would facilitate data fusion easily.

For the immediate future, data collection and fusion for data analytics are also complicated by security concerns, particularly the confidentiality of the information. Today, data mining is often achieved through access to databases and/or data sets that have been exposed to the public via web pages. CPS used in healthcare, by utilities, and elsewhere are often maintained on closed networks with understandable reluctance to share the information with third parties.

Currently, a migration to Web Application Programming Interfaces (APIs) based on SOAP and REST provides a flexible means of serving up data in loosely coupled systems allowing "mashups" of data from multiple sources into analytic services, which fuse the data for predictive and other purposes.

B.5.2.1.2.1 Example

Figure 28, which shows the merger of different data sources (often from distinct databases), depicts the model that is generally used today for obtaining information from data and integrating them into a common data source.

This approach, though commonly used, may be inadequate to handle the scale of the IoT. Many of the systems require collection of the data on an intermediate server. These servers cannot be federated. Traffic is also open to man-in-the-middle attack. The data which is in binary does not expose the binary structure without metadata that provide a definition of the data types, engineering units, and of course where this data is permitted to be used or shared with others. This must be accomplished via initial provisioning of the devices and authorization by the owner. The scale of the data needs this definition which is better handled in XML. Binary data can be used but this is better relegated to the end points. Many require polling which increases data traffic and can be impractical when using the Internet due to significant latency, and, that the routing of information is via best efforts through unknown intermediaries.

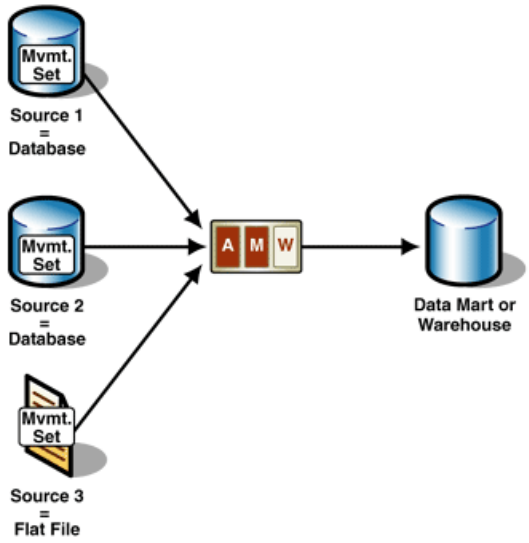


Figure 28: Merger of Different Sources of Data

Figure 29 depicts an example of data fusion today.

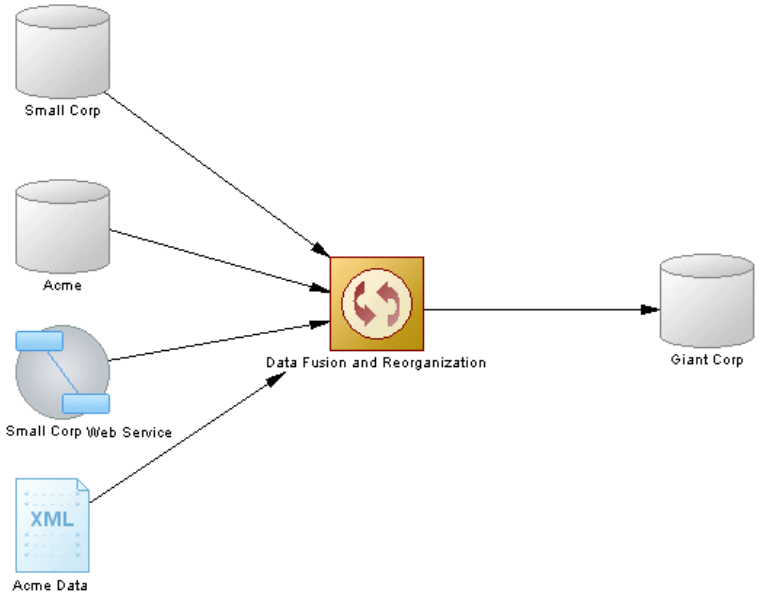


Figure 29: Data Fusion Today

Today's profusion of data sources and uses imposes an additional requirement in that the data flows may need to be shared with multiple locations simultaneously. This drives a requirement for multicast capabilities with extended trustworthiness that preserves the data's integrity and rights.

In the case of a sensor device, the endpoint in the second diagram could be a sensor or group of sensors collecting information, but there would still be a need for data concentration and forwarding to an endpoint collection system. System owners must decide whether to disseminate the information directly from the endpoint via a local or regional server/concentrator or use a federated cloud repository that contains the information. Distributing the information is more practical as long as an effective trust engagement is used to assure integrity of the devices with a data sharing capability.

B.5.2.1.2.2 Discussion of relevant standards

This section discusses a sampling of standards used which are exemplars of what is needed to make end to end data interoperability work. Other standards exist which perform similar services. These are intended to provide understanding of the scope of the problems they address.

There are systems such as Metadata Access Points (IF-Map) that have a data binding using SOAP [153][154]. There many standards that use SOAP, which was developed by the OASIS Foundation [156]. Most protocols today are either binary or utilize REST, which offers hypertext interfaces originally used for web page exchange and utilizes SSL/TLS security [239]. The use of REST has grown. See section B.5.3.5 on Data Service Patterns for a more detailed discussion of exchange mechanisms.

There is a joint effort known as ISO/IEC/IEEE P21451-1-4 (also known as Sensei-IoT*) [115] that has defined a common transport language with built-in security. It offers the data in a common form utilizing eXtensible Markup Language (XML) constructs known as IoT XEPs (Extensions) to the eXtensible Messaging and Presence Protocol (XMPP). This approach has security built into the protocol using Transport Layer Security (TLS) and makes use of trust engagement whereby all devices must be registered to participate in a network. Assuming the root of trust is reliable, this trust relationship allows the data to be trusted and shared with other domains under the control of the owner of a participating device.

The data expressed in the form of XML makes merging information with other systems much easier. Moreover, an additional benefit is that during the transition of the original protocol data representation from one intermediate form to another, it offers metadata isolation and the ability to apply policy for the particular data, which in turn provides the ability to apply control on a more granular basis. Textually serialized data using XML is often expressed in JavaScript Object Notation (JSON) [128], which is an equivalent format that conserves size of the data while improving compatibility with programming languages handling messages.

The transition, which may take a binary protocol form into XML or return to another form, provides metadata isolation benefits which can benefit the use of data in a common form expressed in XML. This provides a transformation but also offers cyber security benefits for the

user of the data. It is among the first Semantic Web 3.0 standards to address the complexities of the IoT [128].

The XMPP protocol is used extensively in social networks such as Skype™, Yahoo™, MSN™ and data sharing systems such as GotoMeeting™, WebEx™, gaming systems, and now Software Defined Networks (SDN). However, while they use XMPP to set up the security session, they often use other protocols to secure the exchange of data.

B.5.2.1.2.3 Summary analysis

Trustworthy data fusion will continue to be a challenge until systems can ensure the integrity and confidentiality of the data, non-repudiable identification of relevant actors and devices, and creation of justified trust among users, devices, and applications. CPS present a challenge if the Internet is to be used as a vehicle to transport the information. Each of the technologies presented in this subsection have deficiencies noted in one aspect or another. New approaches are needed to provide the assurance that data fusion results in integrity and that the information from those systems is interoperable across different domains of use.

B.5.2.2 Complex data exchange and other management issues for interoperability across heterogeneous systems

When IP was being developed in the mid-70's and early 80's, most computers were large, stationary, and expensive to own, and generally had limited interaction with other computing environments. The foundations of both computing and internetworking, including use of the Domain Name System (DNS) to facilitate translating between IP addresses and host names, have therefore been rooted in a location-centric mindset; data and other information in digital form is counted on to be accessible at a location and, for the most part, immobile. Thus, the broadly accepted view is that such information cannot be addressed directly through a persistent and unique address but must instead be referenced via a computer address followed by a data pathway within that computational environment.

This method of naming, storing, and moving digital information has become increasingly problematic in the face of trends such as mobile computing, data-producing smart 'things', increasing size and volume of data files, and decreasing costs for both bandwidth and storage. More data are being stored, in more formats, for more widely varied uses than ever before. Information and analytics have become commonly traded commodities and are often moved across trust and privacy boundaries, touching multiple administrative domains. Data pathways are becoming increasingly complex and increasingly vulnerable to loss of availability, integrity, and confidentiality.

Additionally, the role of a client of data may determine the nature of access. For example, in manufacturing precision and control are critical, so access to read and write data are highly constrained. The relationship between controller and actuator nodes is often termed *tightly*

coupled. On the other hand, such a control system may have access to measurement data that might be of value to other CPS clients outside the control system or even outside the CPS domain. This client relationship can be termed *loosely coupled*. From this example, the tightness or looseness of the coupling between communicating parties may be based on their respective roles in CPS.

An example of this complexity occurs in the manufacturing domain. Many of today's medium-to-large manufacturing enterprises have multiple lines of business, each with multiple plants, each of which contains multiple communication networks that are logically layered. Giving decision makers access to information produced by these plants in a timely manner and in a form normalized for useful understanding is quite a challenge.

The communications networks within a plant often have a hierarchical topology where lower layers become increasingly specialized to meet requirements of the manufacturing functions and systems they support and the conditions in which they operate. The communications equipment in these lower layers is considered manufacturing equipment, which has a long lifecycle and is expensive to take offline; it is rarely replaced or upgraded. Figure 30 (from ChemicalProcessing.com) shows a simplified view of a topology of networks for a process plant in the continuous process industry. Many of the production equipment and sensors that produce manufacturing data reside at the bottom of this hierarchy. This equipment is infrequently replaced, leading to a set of equipment that is diverse in type, era, and technology.

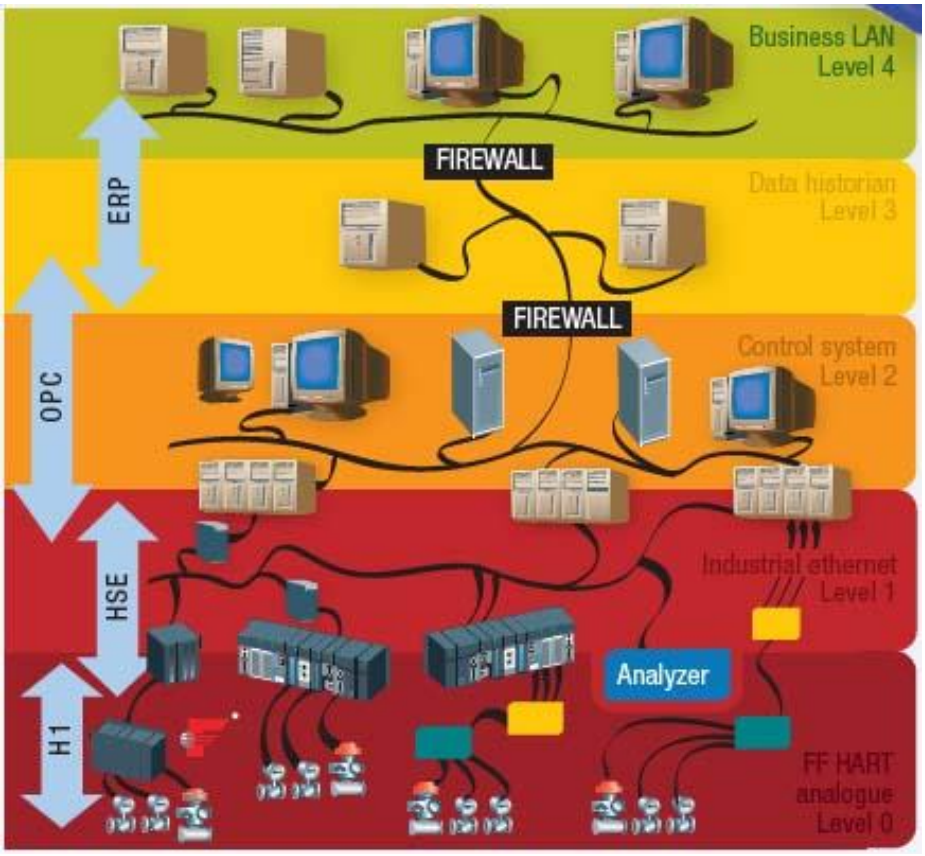


Figure 30: Simplified Topology of Networks for a Chemical Plant

Typically, data from production equipment must flow through its supporting specialized networks upward to reach the enterprise network where business applications support corporate decision-making. Such data are typically refined and digested to produce a smaller aggregate result. The raw data itself, however, is being increasingly found to be important for manufacturing and business intelligence, once characterized and transferred. Various approaches are being investigated to achieve more timely and easy access to this data. These approaches include: (1) using machine-to-machine technologies and standards to connect equipment or specialized equipment networks directly to corporate clouds and (2) adapting elements of ubiquitous network technologies to factory networks while maintaining performance characteristics such as determinism, availability, security, and robustness that are needed to ensure safe and proper operations. While hierarchies will not disappear, plant architectures will slowly become more homogeneous and provide a common means for collection of data from lower layers. Challenges lie in avoiding adverse impacts on the performance of production systems and networks, in providing confidentiality of data (at and after collection), and in providing means to normalize and merge diverse data such that it provides correct views of an entire portion of an enterprise. The expected lifespan of capital

assets, issues of safety and availability, and many characteristics required for manufacturing control networks also apply in other domains.

Approaches, technologies, or architectural elements that address data integration problems in many or all domains of CPS will have a broader and longer impact than those that apply narrowly. Two standards that seek to provide a comprehensive solution to data integration are the Digital Object (DO) Architecture [105] and Recommendation ITU-T X.1255 [104]. They represent a basic architectural foundation whereby mobile programs, smart applications and services, and devices of various kinds involved in managing information in digital form can exchange information on the location and provenance of data. Also of note is the recent establishment of an infrastructure to manage the evolution and deployment of this DO Architecture globally [144].

B.5.2.2.1 Example

An embedded-control boiler system that has been in service for decades is being migrated into an IT infrastructure through a new capability. Previously, the data generated from this system was generated, stored, and accessed by known parties using locally known infrastructure through set data paths. Now this data must be made accessible globally, for use in unknown and potentially complex systems, through unknown infrastructure. A tool is required that would enable such transactions or operations.

The simplest method of storing and locating data in this scenario employs a repository that is part of a secure cloud computing service that can be uniformly accessed by any number of authorized third parties. This may present challenges to data privacy and ownership, because once the data moves outside of the originating entity's infrastructure, it becomes subject to the cloud computing service provider's trust framework. In addition, if the originator wants to move the data from one service to another, the data pathway changes and must be changed with all accessing parties as well. Credentials may also have to change.

The originating entity might instead choose to host the data in its own infrastructure for better privacy; however, this introduces the same kind of complexities as described above, and may increase security and privacy concerns. As the data ages, originators might need to move old data into storage or destroy it altogether. Network locations and naming conventions may change over time as the originator's system evolves. Abstraction can be used to limit the amount of manual work required to maintain data in such a scenario; however, this increases the complexity of initial setup. All these factors increase the complexity of maintaining persistent data pathways for accessing parties and present major challenges to efficiently realizing value from the data. The resulting inability of users to store and manage their own data is a challenge for maintaining an open, competitive, secure, and privacy-enhancing data marketplace.

B.5.2.2.2 Discussion of relevant standards

Modern web standards and practices provide many tools for describing, fusing, sharing and accessing distributed heterogeneous data (see Christian Bizer, Tom Heath, and Tim Berners-Lee, “Linked Data – The Story So Far” [108]). The standard web infrastructure and protocols [109] provide a means for accessing and sharing distributed data. Any kind of element can be considered a resource and named using an internationalized resource identifier (IRI) following guidelines in the standards and practices associated with linked data. These standards include the Resource Description Framework (RDF) [110] and the Web Ontology Language (OWL) [111] for describing the data (i.e., these are languages for metadata), formats for encoding the data and related metadata for sharing and fusing [112], and a protocol and language called the SPARQL Protocol and RDF Query Language [113] for merging (via SPARQL endpoints) and querying the data. These standards and approaches have been used to integrate industrial data in the electric power industry, oil drilling industry, and manufacturing shop floors, among others.

The digital entity data model is a standardized approach that makes use of components of an infrastructure that are distributed and interoperable with each other in practice [104] [105]. It is compatible with existing Internet standards.

Another approach to data exchange is being developed in the IETF/IRTF ICNRG working group [114] that is focused on Information Centric Networking (ICNRG). This concept, which proposes the notion of uniquely named data as a core Internet principle is being explored by several organizations. In this approach, data is treated as being independent of location, application, storage means, and underlying means of communication. Data is no longer routed by address but rather routed by name. Thus data names at the application level can be mapped to names at the routing level. Additionally, ICN moves away from the model where we protect the channel across data flows and instead the data is self-securing (it is signed at the time of creation, to ensure integrity and authentication). Finally, it also supports native caching in the network.

B.5.2.3 Data-driven interactions between dependent and independent CPS

For effective and controlled data interaction to occur between the various elements of a particular CPS system, roles, procedures, rights, and permissions of the humans who create and manage each system must be defined. These humans will ultimately be responsible to set up, manage, and maintain both the cyber and physical components of such systems.

The primary challenge, then, is to develop a set of definitions that is comprehensive and unambiguous, so that interactions between systems can be appropriately described and standardized. The multiple dimensions involved are discussed herein.

As it regards data-driven interactions between dependent and independent CPS, for clarity three groups (actors, roles, and permissions) are identified.

ACTORS: While any particular CPS instance may involve different actors, they will generally fall into four categories:

1. Those who manage the data elements (Data Managers)
2. Operations/production personnel who interact at some level with a CPS element (Operations Staff)
3. Governance, Risk, and Compliance (GRC) personnel who manage the various security and governance elements that may be required (GRC Staff)
4. Devices and subsystems that operate on behalf of personnel

These actors will interact with each other based upon their defined *roles* (see below), with each role consisting of a series of *permissions* (see below) that will govern such interaction.

ROLES of actors:

- *Data managers* will be responsible for creating the processes that will manage all data elements that initiate an action to, or are the result of an action from, a CPS device. Data managers' roles will include program development, testing, and deployment; database management; and data analysis management.
- *Operations staff* will be responsible for the physical devices that are employed as well as those that perform the actual human tasks that may be included in any set of managed processes.
- *GRC staff* will be responsible for defining and managing all processes and rules that may be required to meet governance and oversight standards that apply to certain processes.

PERMISSIONS: Permissions will be established for each role of each actor and will govern the actions that each actor will be responsible for. The following permissions and their associated definitions will be present in most CPS systems:

- Define interaction points between devices
- Initiate specific interaction points between devices
- Monitor interaction points between devices
- View data
- Modify data
- Create new workflows
- Import data from other devices

- Export data to other devices

Control Processes and Procedures - The actual control processes and procedures must be clearly defined. Examples of these processes and procedures include:

- Define interaction points between devices - CPS devices, whether dependent or independent, will need precise parameters by which they can interconnect. This may vary even with the same device, based on what other input/output is being employed for any particular instance.
- Initiate specific interaction points between devices - Once the interaction points have been established, there must be a trigger, or event, that initiates the ensuing process(es). In a dependent CPS device, the triggered data event will most likely begin once the output of its dependent source begins transmission. In an independent CPS device, that initiation will range from simple 'human' kick-off to timing devices that auto-start the independent device.[22]
- Monitor interaction points between devices - It might be argued that this is a combination of 'Presence' and other factors defined below, nonetheless procedures must be clearly defined that continuously monitor these interaction points to ensure that they are reporting and functioning as required for each specific interaction.
- View data – This can define which actors are given access to specific data sets.
- Modify data – This can define which actors have the authority to modify data once transmitted. [23]
- Import data from other sources – This can define which actors have the authority to import data from other systems or databases.
- Export data to other sources – This can define which actors have the authority to export data to other systems or databases.
- Create new workflows for each component process – The fact that there will be differences in precisely how each component process occurs or interacts necessitates clearly defined workflows so that consistency is maintained regardless of the origin of any particular process. A critical companion of each 'flow' must be an audit trail that is never allowed to be modified or deleted. Unless such an absolute audit trail is in place, it will be impossible to determine with certainty what may have occurred should any such component procedure fail or be compromised in any way.

[22] From a pure definition standpoint it must be determined if such an action makes the 'independent' device a 'dependent' device, as its function is 'dependent' on the stated action.

[23] Precise audit trails must be in place for compliance and regulatory oversight permission to be granted to modify viewed or transmitted data in any way.

- Sanitize data to conform with regulatory and privacy requirements – Owing to the magnitude of 'big data' that may be produced by CPS devices, there may be a need to sanitize data to remove extraneous elements that result during specific operations. This also includes removing combinations of data elements that cannot be shared to avoid privacy or rights leakage. Any such sanitizing must be strictly controlled, including which actors/devices may perform such action and a requirement that all actions must be instantly and permanently archived in a way that prevents tampering after the fact.
- Interact with other data sources – CPS devices may need to interact with other non-CPS sources of data, such as an on-line security check of personnel. Such interaction may be automated, or conducted by humans. The procedures and methodologies must be clearly defined and data-maps must be pre-established for consistency between such sources.
- Report outcomes to stakeholders/actors – As cited in the example below in Section B.5.2.3.1, there must be defined procedures and processes by which each actor/stakeholder interacts with the output data. In certain instances, it may be simple reporting for archiving purposes, while in other instances notification may need to be immediate and redundant if mission-critical actions are required.
- Request permission to modify/delete data – This can define the process by which individual users may initiate a request for their ability to modify/delete data, including to which specific sets of data the permission applies. Any such action must be strictly controlled and a record instantly and permanently archived for future auditability.
- Define rights and permissions – Strict controls must be in place that determine the rights and permissions that each actor may be granted or restricted to. These determinations must include not only audit trails, but multi-level redundancy in managing these processes and procedures to ensure compliance and enable regulatory oversight. Rights will include, but not be limited to: View Data Only; View and Suggest Data Modification; and View and Modify Data.

Finally, various mechanisms must be developed and specified to ensure that these processes are implemented correctly and reliably. Examples of these mechanisms include:

- Ensuring that all required CPS devices are functional
- Ensuring that all required CPS devices are in place to monitor their intended functions
- Ensuring that data are being transmitted in the form and format needed
- Establishing trust factors

B.5.2.3.1 Example

The following illustrative example discusses how various CPS devices will interact to improve security to help manage the procedures and processes to control the safety of the more than six million shipping containers that enter US ports annually [130].

While this example represents a potential comprehensive end-to-end solution, it is impossible to 'boil the ocean' in an attempt to reach all of the stated goals as a single project.

Therefore, any single project may be divided into smaller subsets that will be effective on their own, while leading to a full implementation over some undefined period of time as the new features are deployed and integrated.

A suggested roadmap of some of these iterative steps will follow the example detail below:

- There exist multiple vulnerability points from the time that a shipment originates in a foreign country until that shipment arrives at its final US destination.
- In this case, the manufacturing point of origin is the primary point of vulnerability where goods can be tampered with, or hazardous materials can be packaged and concealed as the product is being prepared for shipment.
- RFID tags could be placed on each item and locator tags then placed on each pallet used to load a container to track the movement of all items. An assigned freight supervisor will monitor the loading of the shipping container, ensuring that each item has a RFID tag. As each RFID tag is attached a resultant scan will transmit the data to a secure storage system.
- Finally, the supervisor will place a Digital GPS Tracking device within the container and secure that container with a digital seal that will instantly report any tampering to the secure storage system cited above. All associated and/or resultant actions will result in that data being transmitted to a cloud database or other monitoring system.
- Standard screening inspections of the shipping container at port of exit are then performed using devices such as a gas chromatograph or mass spectrometer working through container air vents to ensure that no explosives or harmful chemicals are present.
- These CPS devices will instantly upload data to a central data repository that will alarm should any negative feedback result using the Digital GPS Tracking device mounted inside or attached to the container.
- If final packaging and consolidation was not performed in the factory as described above, it will usually take place at a warehouse or staging area that prepares the product shipment for truck or rail transport to the port. At this stage, illicit activity can occur while products are being consolidated into larger shipping loads, and while being trucked or railed to their maritime port of debarkation. Constant surveillance within the warehouse facility, final load inspection, and employee background checks for both warehouse and transport personnel are effective to improve security. As a prepared load is being transported, a truck can easily be diverted from its given route, providing the opportunity to tamper with the shipment. The use of GPS technology gives transportation management the ability to better track adherence to routes. Truck drivers often have broad discretion over their routes, and are subject to last-minute changes.

- Freight dock supervisors will constantly monitor the RFID tags of each piece of freight or pallet to ensure that it remains on its proper path.
- These RFID devices will instantly upload data to a central data repository that will alarm should deviation from established routes occur for any tagged piece of freight.
- Once the container is at sea, procedures must be in place to prevent tampering. Containers typically do not have a uniform seal or any way to exhibit obvious signs of tampering. Ocean carrier personnel may not routinely check containers for seals or signs of container tampering while onboard. Container ships often stop at various seaports to unload and load containers. The container ship's transits through various routes and ports pose different levels of security risks.
- A digital tampering device combines a covert Assisted GPS tracking & sensing device with a reusable electronic seal that can be affixed to a conveyance door. The GPS tracker can be hidden within a pallet. This system is web based, so when a seal is compromised, the GPS device sends the event and location information to the stakeholder for immediate action. This system can be used for cross border or domestic trailer tracking using cellular and web-based technology.
- As above, this CPS device will instantly upload data to a central data repository that will alarm should any tampering occur. If the upload is thwarted, through either oversight or malicious activity, the vulnerability can be remediated through adequate message auditing.

Once the container arrives at the port of entry, it may be at risk of tampering, especially if it must sit for extended periods of time before being staged and loaded onto a cargo ship. Terminal operators may not routinely check containers for seals or signs of container tampering, so a device such as described above will help to further ensure the integrity of each container. Alarms should be logged by default with the logs subject to integrity checks and audit.

Re-conceptualizing basic legal documentation in the maritime industry, in particular bills of lading, may also serve to enhance security and reliability across various related industries such as shipping, banking, and insurance. Where unique persistent identifiers are associated with information structured as digital entities (aka digital objects), it is possible to move beyond static information and create more dynamic data structures. As an example, if a storm occurs at sea and a container is swept into the ocean, video information captured at the time of its lading when compared to conditions at the time the container broke loose may be used to identify possible negligence in strapping down the cargo; and the relevant insurance companies may be notified as appropriate [106][137].

B.5.2.3.1.1 Suggested order of the first two iterative projects

PHASE 1

A. Standard screening inspections of the shipping container at the port of exit are performed using devices such as a gas chromatograph or mass spectrometer working through container air vents to ensure that no explosives or harmful chemicals are present.

B. Place a digital GPS tracking device within the container that will track that container's movements so that any diversion of previously specified routes will cause an instant notification to appropriate authorities.

C. Once the inspection is completed, secure that container with a digital seal that will instantly report any tampering to the secure storage system. All associated and/or resultant actions will result in that data being transmitted to a cloud database or other monitoring system.

PHASE 2

A. RFID tags could be placed on each item and locator tags then placed on each pallet used to load a container to track the movement of all items. An assigned freight supervisor will monitor the loading of the shipping container, assuring that each item has an RFID tag. As each RFID tag is attached, a resultant scan will transmit the data to secure storage system.

B. As in Phase 1.C above, once the inspection is completed, secure that container with a digital seal that will instantly report any tampering to the secure storage system. All associated and/or resultant actions will result in that data being transmitted to a cloud database or other monitoring system.

Instituting these two phases will dramatically improve the possibility of spotting or preventing a major adverse event before it becomes a disaster.

B.5.2.3.1.2 Discussion of relevant standards

There are few relevant standards that apply to this aspect of CPS data interoperability. However, the above-referenced example [130] was required to comply with a variety of other applicable standards. These standards include:

- Department of Homeland Security, US Customs and Border Protection, Container Security Initiative (CSI) program [130]
- Department of Homeland Security, US Customs and Border Protection, C-TPAT (Customs Trade Partnership Against Terrorism) program [131]
- Department of Homeland Security, US Customs and Border Protection, Bonded Warehouse Manual for CBP Officers and Bonded Warehouse Proprietors [132]
- Department of Homeland Security, US Customs and Border Protection, *Amendment to the Current Reporting Requirements for the Ultimate Consignee at the Time of Entry or Release*, [133]
- Department of Homeland Security, US Customs and Border Protection, *International Carrier Bonds for Non-Vessel Operating Common Carriers (NVOCCs)* [138]

B.5.2.3.2 Summary analysis

Data-driven interactions between dependent and interdependent CPS require a precise unambiguous set of definitions to describe and regulate these interactions. Some of the needed definitions include roles of actors, control processes and procedures, and monitoring mechanisms. There may be an opportunity to describe and standardize these definitions to enable robust interactions between dependent and interdependent CPS.

B.5.2.4 Privacy-protecting data infrastructures

The ubiquitous nature of IoT/CPS creates the potential for data in these environments to be intrusive. Protecting the privacy of the humans, businesses, nation states, non-profit institutions, and other entities involved in a complex CPS is an increasingly difficult proposition because data are being produced in greater volumes, from a greater variety of sources. Complex proprietary data infrastructures have combined to make the overall data infrastructure more opaque, and data access controls vary dramatically as the number of vendors and products that produce data in a CPS grow. Data are often mined in ways that do not currently require a user's explicit permission. Data storage is increasingly moving away from the users that own the data and is being centralized in third-party cloud servers. Movement of data often includes multiple third party brokers or aggregators. Data leakage is often a side effect of data collection (e.g., an observer can use appliance data to determine if a user is at home). Ironically, attempting to impose access control and integrity protections can actually serve to decrease user privacy as the authenticating information, which is stockpiled with increasing numbers of security administrators, grows.

Lack of a uniform way to identify, secure, store, and access data across proprietary system boundaries has made it difficult for users and institutions to effectively manage privacy. Indeed, companies such as Google have recently made it clear to regulators in places such as the EU that, given today's infrastructure, it is exceedingly difficult to give a user the ability to be 'forgotten' in the Internet.

The release of personal information, even to support the normal functioning of a system (e.g., the provision of services at an individual's request) can still raise privacy risks. These risks could include stigmatization of the individual or loss of trust from the unanticipated revelation of personal information or from the release of inaccurate information. Thus, any standard or implementation needs to incorporate design requirements and privacy-enhancing controls to support the protection of privacy and civil liberties in the developing CPS ecosystem. User management of the release of attributes is one such control.

Although user control is important, individuals are not always in the best position to mitigate all privacy risks. Therefore, any potential approach should include design requirements and controls that do not rely solely on user management. Requirements that provide the capability for claims to be derived instead of releasing actual values can limit the unnecessary disclosure

of personal information. For example, if an online credential can get a teenager into a movie theater, the only exposure necessary is that the teenager is older than seventeen. Full birth date, even birth year, is not needed. Metadata should also have privacy-enhancing controls. For example, if 'over 17' is asserted, the implementation should consider that a 'valid DMV' asserted that fact, not that 'the Virginia DMV' asserted it, causing unnecessary data leakage. The objective is to consider the full range of privacy risks and appropriate mitigation strategies that can be incorporated into executable, implemented systems, and not just rely on manual management policies.

B.5.2.4.1 Example

An advanced utility grid is using data from millions of synchrophasers[24], heat sensors, vibration sensors, and other data production points to balance power generation against system load through "sense, actuate, and control" CPS. Sources of data generation in this environment include power generation assets owned by a variety of vendors: Independent Service Operators (ISOs), public distribution infrastructures, local municipal infrastructures, and the industrial/commercial/consumer's premises.

The information collected comes from a variety of different sources, through a variety of infrastructures, and via a variety of different market pathways. Consumer data such as power consumption information from appliance vendors may be used to estimate potential load on the grid but can also leak information such as when a person is at home, what specific electricity-consuming activities the person is engaging in, and even what media a person is consuming on their devices. Asset operators may expose proprietary operational information, such as which assets are utilized in certain scenarios and how assets are being utilized and managed, just by providing data to central aggregation/analytics points. Even public information may be collected and analyzed. For example, social media surrounding popular sporting events may give a hint of load spikes to the grid, but may also reveal information about individual participants in the aggregated data.

A user - whether institutional or individual - who wishes to protect their privacy in such a system of systems may have a very difficult time simply locating all the different collection points and data stores that track usage patterns, and may not even be aware of the individual data collection practices of the vendors involved. A user in such a scenario has very little expectation of privacy and very little capability to control what information of his or hers is being shared with whom, and for what purpose.

[24] Synchrophasers are data acquisitions systems that measure the phase of electric power whose measurements are time synchronized to a reference. traceable to an international time scale, such as UTC or TAI.

B.5.2.4.2 Discussion of relevant standards

To truly enhance privacy in the conduct of online transactions, the Fair Information Practice Principles (FIPP)s [159] must be universally and consistently adopted and applied in the CPS ecosystem. The FIPPs are the widely accepted framework of defining principles to be used in the evaluation and consideration of systems, processes, or programs that affect individual privacy.

However, the FIPPs may not be enough when engineering automated systems. As such, NIST, in a public and private partnership, is exploring privacy engineering methodologies to integrate privacy-preserving controls directly into systems as opposed to depending solely on documented paper policy. As illustrated in Figure 31, the FIPPs provides the baseline input to an overall privacy engineering methodology, but is not the sole tool used to impact effective privacy management.

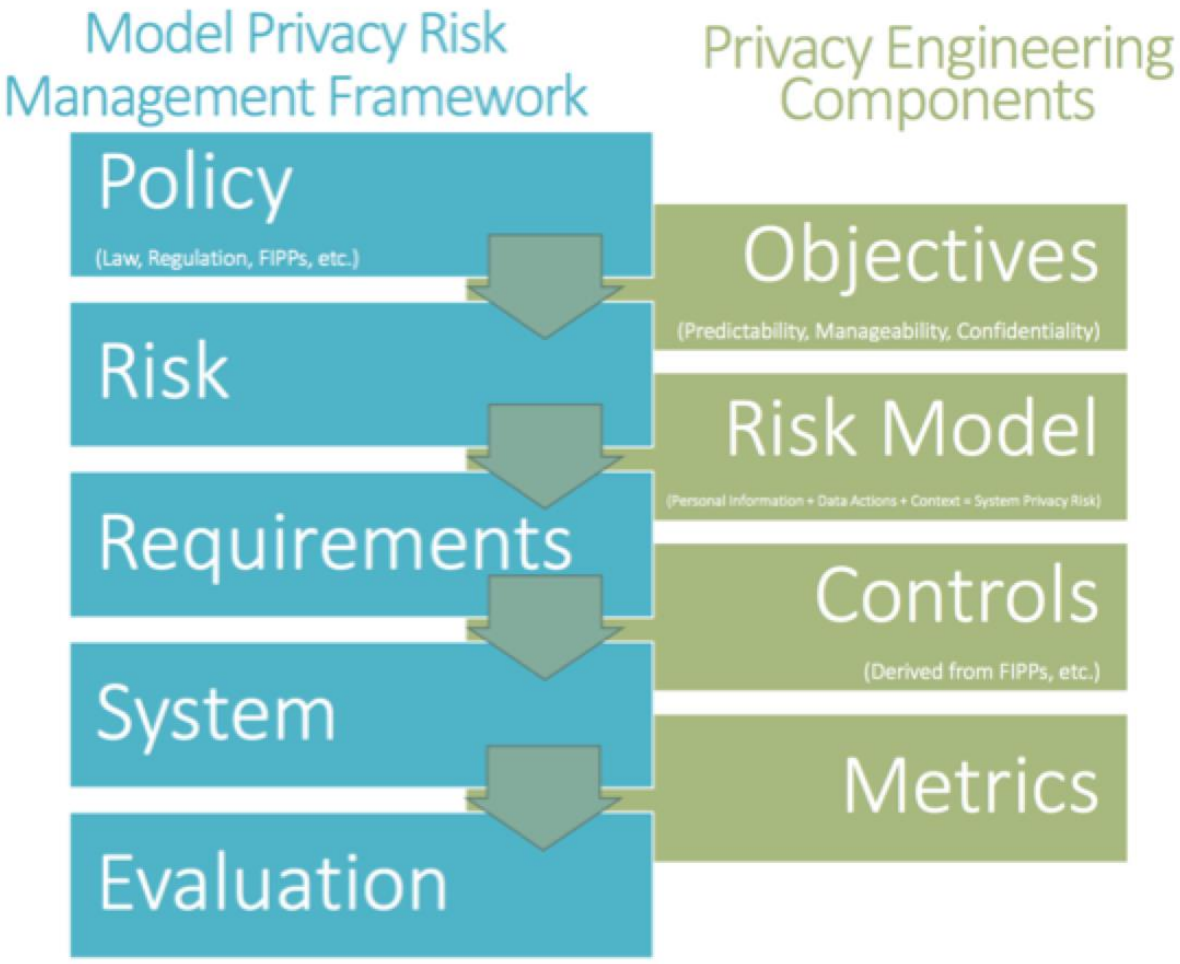


Figure 31: Continuous Refinement of Privacy Risk Management

These concepts are under continuous refinement, but could serve as another data point in CPS efforts to engineer privacy directly into systems that potentially handle personal information.

Specifications like OAuth, OpenID Connect, and User Managed Access (UMA) allow explicit user control over information release. During transactions governed by these specifications, where a third party is requesting information, the user is required to consent prior to disclosure. Fine-grained user controls are possible that allow individuals to manage consent in a myriad of ways. For example, a user can allow one-time release, whitelist entities where release does not require consent, turn consent on/off for an individual datum, or revoke consent for any or all previously authorized entities. Emerging concepts such as Personal Data Stores (PDS) can and should influence attribute standards and should be built upon existing standards that give users explicit control and choice over the information they share.

Other approaches include, but are not limited to, cryptographic profiles that include zero-knowledge assertions such that intermediaries or brokers cannot see attribute values, and design requirements that limit the building of user profiles by preventing identity providers from knowing the consuming relying parties. Commonly known as double or triple blind, this latter approach is not codified in any singular standard, but is becoming a de facto implementation technique to limit traceability of users online. Figure 32 is a data instance diagram of a possible double-blind scheme.

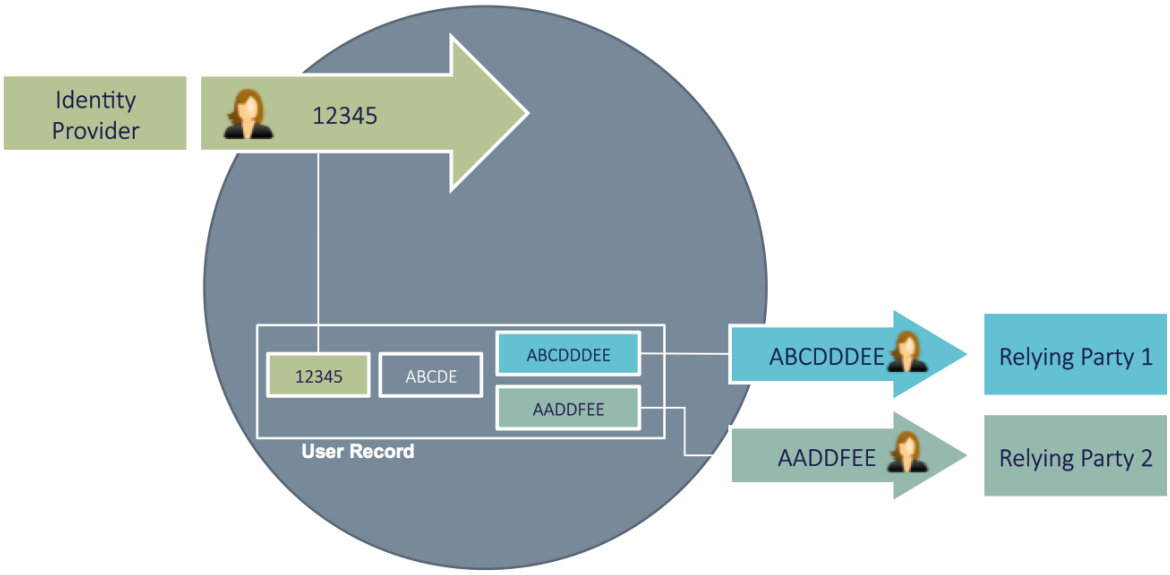


Figure 32: Double-Blind Authentication Scheme

This model is designed specifically to ensure that privacy requirements of anonymity, unlinkability, and unobservability are built in from the start. However, without the appropriate cryptography, this model allows user information to flow freely through the broker depicted by

the gray circle. Although great care is taken to generate pseudonymous identifiers throughout the system, any personal information provided by the identity provider needs to be encrypted in a manner that keeps the broker from viewing information. This is simple in traditional Public Key Infrastructure (PKI) systems where the source system, the Identity Provider (IdP) encrypts the data for the destination system, the Relying Party (RP), using the RP public key. Yet, traditional PKI breaks the design requirements of anonymity, unlinkability, and unobservability because knowing which public key to use means the IDP knows where the user is going. Open, tested, and approved cryptographic algorithms must be used to keep attributes encrypted without exposing the user destination to the IDP. Such cryptographic techniques are not yet available in common use. Finally, the broker is in an extreme position of power, as well as being a prime attack vector for those who wish to do harm. Automated compensating controls, in addition to paper policy (contracts, laws, regulations, etc.), are still under development to reduce or eliminate the vulnerabilities of the double-blind, broker-centric architecture.

B.5.3 Data Interoperability Issues

This section describes traditional data interoperability issues that are critical to all data exchange methodologies, in CPS and non-CPS.

B.5.3.1 Data models, relationships between data and data type

Terminology has evolved from the ANSI notion of data modeling that described three types of data schema (or model): conceptual, logical, and physical. Often, the key distinction now is between data models and information models. The discussion below is largely derived from a presentation by Ed Barkmeyer [142] to the Ontolog Forum in 2007, though there are other sources that similarly distinguish data models from information models such as RFC3444 [143].

Data models and information models differ both in nature and purpose.

Data models relate data to data. They support software implementations and organize data for access, encoding, or processing. Their classifiers (i.e., primary language constructs) describe the structure and type of the data.

Information models relate things to other things, as well as things to information about those things. These models are used to support a set of business processes or describe a domain and organize information for human comprehension. They use classifiers to collect properties. Transformation rules often exist for information modeling formalisms to data modeling formalisms to enable generation of data models from information models.

Semantic models (many of which are called *ontologies*) are information models that are meant for machine "comprehension". These models use information to classify things. Semantic models are often constructed using knowledge representation methods, languages, and technologies. Such languages are sufficiently formal to support machine reasoning that provides this comprehension. Examples of inferences this can support include: revealing

relationships between elements of independently authored ontologies or data sets (classifying both types and things), determining the logical consistency of a model, and determining the satisfiability of particular elements of a model (i.e., whether or not it is possible for any instance to exist that satisfies all the constraints of its type).

A way to distinguish these different kinds of models is by what their classifiers classify and how they do it. If the main classifier in a modeling language describes a data structure (such as an Element in XML Schema) then it is a data modeling language; if it describes properties associated with an entity (such as attributes and associations for a class in UML or relations to an entity in ER diagrams) then it is an information modeling language.

As one moves up this spectrum, the models become less prescriptive and more descriptive. Semantic models have flexibility that is quite useful for integrating information, but data models have the specificity needed for insuring their integrity for use in implementations of critical systems. Thus, both are useful for data integration in CPS.

An obvious goal of data exchange is conveyance of understanding from the data source to a destination user of the data. There has been much work on defining interoperability and understanding; it has been developed from very theoretical first principles to quite practical terms. Some examples can be found in the Web Ontology standards from the W3C [139][140].

This section describes the three key dimensions that allow conveyance of understanding. Note that other aspects of data interoperability are covered in other parts of Section B.5.3, but this one deals with the data itself.

The first subsection, Section B.5.3.1.1, describes the concept of data models (and the higher abstraction called semantic models or information models) and how they are typically scoped and described.

The second subsection, Section B.5.3.1.2, describes metadata as data related to other data; outlines the major kinds of metadata used in the library community and how these kinds relate to our concerns; describes the importance of metadata to data interoperability for CPS; and enumerates some things that may need to be done with respect to metadata standards to enable data interoperability across CPS.

The third subsection, Section B.5.3.1.3, describes data type and structure.

B.5.3.1.1 Data models

"A message to mapmakers: Highways are not painted red, rivers don't have county lines running down the middle, and you can't see the contour lines on a mountain." [Kent, William, updated by Steve Hoberman. "Data & Reality: A Timeless Perspective on Perceiving and Managing Information in Our Imprecise World." Westfield, NJ: Technics Publications, 2012. Print]

The above tongue-in-cheek quote begins the 1978 preface to William Kent's classic book on data modeling, *Data and Reality*, and shows that everyone understands data modeling to a certain degree. Reducing, for the moment, the nice distinctions made above among data, information, and semantic modeling to a single concept, we can address the general challenge with modeling, which is the difficulty of mapping some subset of the real world, including CPS, onto a conceptual structure that allows us to more easily understand and/or manipulate that real world subset, within certain constraints. Those constraints include the limits of the modeling language used, i.e., what can and cannot be expressed using the language, and the difficulty of capturing all of the relevant information. Furthermore, even using the same modeling language, multiple individuals can easily create variant conceptual structures describing the same real world subset. With this in mind, the relevance of data modeling to data interoperability is quite clear. Data captured from a given CPS will be structured according to a certain model, and that model will be constrained by the modeling language used, by the level of granularity of the data collected, and, now going back to the distinctions among data, information, and semantic models described above, the basic type of modeling being done. Combining data streams from multiple CPS at multiple times structured according to multiple data models using multiple approaches to structuring the data is a specific and challenging subset of the general and well-known problem of making sense of heterogeneous data sets.

Approaching specific data interoperability problems in CPS will require understanding the data modeling, or even lack of modeling, that has resulted in the available data structured or presented as it is. As noted elsewhere in this document, a clear requirement for data interoperability among CPS is that many CPS are legacy systems that must be accommodated in any data interoperability scenario and that clean slate solutions ignoring that legacy are unacceptable.

It is tempting to compare modeling approaches to each other and to favor one over another, but that ignores both the issue of legacy systems and the even more basic fact that different situations and different points of view require different approaches to modeling and no single solution fits all cases. Contrast, for example, OMG's Unified Modeling Language (UML) and W3C's Web Ontology Language (OWL). Both are widely used, historically by separate communities for different purposes, both are appropriate to those purposes, and both can be used synergistically within the same domain. UML comes out of the software engineering and more traditional data modeling community while OWL comes more out of the artificial intelligence community and looks at knowledge representation. One cannot be favored over the other in general, but each is appropriate to and solidly in place in its own community. It is beyond the scope of this document to compare modeling approaches, but furthering the work of data interoperability in CPS will require understanding those approaches and the tools that can help in mapping from one to another.

One issue that will come up over and over in data modeling is the issue of metadata, which is further discussed below. Data, including data relevant to CPS, goes through a lifecycle. At each

stage the difference between data and metadata is not in the kind of data but in the relationship of that data to other data. Thus, what is considered primary data and what is considered metadata can vary through the lifecycle.

Here are some examples of typical names of data sets where this consideration could apply. These may not be orthogonal depending on the detailed definitions:

- Status – often derived states from other data categories
- Control – actuators and supervisory control points
- Measurements – sensor data
- Settings – set points, including ranges and frequencies, for algorithms and alarms
- Documentation – manufacturer information, schema references
- Configuration – parameters that bind the device to its system
- Capability – possible degrees of freedom for settings and configuration
- Faults – logs of significant events and problems and their management
- Access management – authorization and authentication
- Identification – identifiers both traceable and opaque (people, processes, devices, and systems), as well as identifiers associated with the digital entities in which such pre-existing identifiers are incorporated for operational purposes.

Note that typically, the ability to communicate these values is often regulated by access rights that include authentication as well as authorization. These access rights are themselves a type of metadata.

Going back to Bill Kent, in his introduction to Entities:

"As a schoolteacher might say, before we start writing data descriptions let's pause a minute and get our thoughts in order. Before we go charging off to design or use a data structure, let's think about the information we want to represent. Do we have a very clear idea of what that information is like? Do we have a good grasp of the semantic problems involved?"

Paraphrasing that for purposes of thinking about the interoperability of data coming out of different CPS, we might ask if we have a very clear idea of the data we are trying to integrate and a good grasp of the semantic problems involved.

B.5.3.1.2 Relationships between data

In his 1968 dissertation, Philip Bagley may have coined the term “metadata” as data about data. In his Extension of Programming Language Concepts [116], Bagley says: "As important as being able to combine data elements to make composite data elements is the ability to associate explicitly with a data element a second data element which represents data 'about' the first data element. This second data element we might term a 'metadata element'."

The way that a "metadata element" in Bagley's definition relates to the data element it describes can be thought of as a role of the metadata element with respect to the described data element. All it means, then, to say that something is metadata is that it relates to other data in a particular way. However, communities differ on which relationships constitute a metadata role. In some communities, everything but raw measurements are considered metadata, while in others complex data structures may capture many of the important relationships among data with metadata only providing data about the entire collection.

Types of metadata correspond to different ways that data can relate to other data. The library community makes heavy use of metadata to describe information resources. NISO, the National Information Standards Organization, describes three main types of metadata [141] used in this community that are also important in the information technology realm. These three types are structural, descriptive, and administrative.

According to NISO, "*structural metadata* indicates how compound objects are put together, for example, how pages are ordered to form chapters." In the IT realm this type of metadata can include data models, data type identifiers and descriptions, and models used to describe structural metadata (aka metamodels). In other words, structural metadata are data about the containers of data.

NISO asserts that "*descriptive metadata* describes a resource for purposes such as discovery and identification. It can include elements such as title, abstract, author, and keywords." This kind of metadata relates to the nature and identity of the data or the thing the data are describing.

Finally, NISO asserts that *administrative metadata* provides information to help manage a resource, such as when and how it was created, file type and other technical information, and who can access it. There are several subsets of administrative data; two that are sometimes listed as separate metadata types are:

- *Rights management metadata*, which deals with intellectual property rights, and
- *Preservation metadata*, which contains information needed to archive and preserve a resource.

In the IT realm administrative metadata will include provenance data as well as data on who may access which information and how.

Metadata may be structured or freeform (e.g., freeform text tags assigned by users to web links, files or services). Metadata describing metadata are also important to evaluating its use.

Metadata are critical to integrating data across diverse systems and having confidence in the implications of the results. Structural metadata provides a means to agree on common forms for exchange or determine common forms for aggregation. It also provides information on how to parse the data and assess its integrity (e.g., by its conformity to the structure and rules

specified in its data model). Descriptive metadata supports finding data relevant to a particular purpose, assessing its veracity, and assessing its compatibility with other data. Administrative metadata supports assessing freshness, trust, and availability of data, as well as the means of access and use.

There are many standards for these different kinds of metadata. For data interoperability to work quickly and safely in CPS, one must assess what is needed from each type of metadata, which metadata standards are in use in different CPS domains, how they relate, and how they should be extended or narrowed to meet time, availability, and safety requirements for data interoperability for cooperating CPS.

On the other hand, Bagley recognizes that metadata represents the need to be able to associate explicitly one data set with another. For example, for a control application, the data might be temperature or energy or relay state. The metadata might be units of measure, scaling, uncertainty, precision, etc. Additional metadata might include make/manufacturer/model/serial number for the sensor monitoring temperature or energy or for the device having the state or attribute being monitored such as the relay. Yet to an asset management application the make/manufacturer/model/serial number is the data.

The use of the term *metadata* may have evolved beyond Bagley's original usage to include analogous types of data about things such as devices and processes. A device data sheet typically describes characteristics of a class of device or machine and may be referred to as device metadata. This is analogous to the role of data type and data models with respect to the data it describes. Additionally, there may be calibration data associated with a particular device that is analogous to provenance information on the source and history of data instances. Since it may be useful to apply the same mechanisms used for managing data about data to these analogous kinds of data about other types of things, it may be wise to broaden the CPS interpretation of metadata to include these other uses of the term.

B.5.3.1.3 Data type

Automated processing of large amounts of data, especially across domains, requires that the data can be parsed without human intervention. Within a given domain that functionality can simply be built into the software, e.g., the piece of information that appears in this location is always a temperature reading in centigrade or, at a different level of granularity, this data set is structured according to Domain Standard A including base types X, Y, and Z where the base types are things like temperature readings in centigrade. This knowledge, easily available within a given domain or a set of closely related organizational groups, can be built into processing workflows. But outside of that domain or environment the 'local knowledge' approach can begin to fail and more precision in associating data with the information needed to process it is required. This also applies across time as well as domains. What is well known today may be less well known twenty years hence, but age will not necessarily reduce the value of a data set and indeed may increase it.

We are using the term ‘type’ here as the characterization of data structure at multiple levels of granularity, from individual observations up to and including large data sets. Optimizing the interactions among all of the producers and consumers of digital data requires that those types be defined and permanently associated with the data they describe. Further, the utility of those types requires that they be standardized, unique, and discoverable.

Simply listing and describing types in human readable form, say in one or more open access wikis, is certainly better than nothing. But full realization of the potential of types in automated data processing requires a common form of machine readable description of types, i.e., a data model and common expression of that data model. This would not only aid in discoverability, but also in the analysis of relations among types and evaluation of overlap and duplication as well as possible bootstrapping of data processing in some cases.

Types will be at different levels of granularity, e.g., individual observation, a set of observations composed into a time series, a set of time series describing a complex phenomenon, and so forth. The ease of composing lower level, or base, types into more complex composite types would be an advantage of a well-managed type system.

An immediate and compelling use case for a managed system of types comes directly out of persistent identifiers for data sets. Accessing a piece of data via a persistent identifier, either as a direct reference or as the result of a search, requires resolving the identifier to get the information needed to access the data. This information must be understandable by the client, whether that client is a human or a machine, in order for the client to act on it. For a machine, it must be explicitly typed. A type registry for persistent identifier information types would appear to be an early requirement for coherent management of scientific data.

Finally, assigning persistent identifiers to types would aid in their management and use. All of the arguments for using persistent identifiers for important digital information that must remain accessible over long periods of time will apply equally well to whatever form of records are kept for data types.

A recent effort to codify types, still very much in development, is a Research Data Alliance (RDA) Working Group on Data Type Registries [157].

B.5.3.2 Identification of type and instance

How does one know what a piece of metadata is referencing? How can one find the metadata for a given digital entity? How can one understand the basic type of an entity? What ties all of these things together? And, finally, because we want people and processes that did not create the data to understand and reuse it, how does one understand them, and which are key to data interoperability?

Unique, persistent, and resolvable identifiers are essential to managing distributed data in the Internet and other computational environments. A digital entity that is referenced from outside

its local domain must be uniquely identified, and that identifier must be resolvable to allow for access to relevant and timely state information about the entity, e.g., current location or access conditions. This allows the identifier for a digital entity to persist over changes in the state of the entity, i.e., the identifier itself remains constant while the returned state data from a resolution request can change as needed.

Allotting a persistent identifier for a digital entity and maintaining that identifier for at least as long as the identified entity exists is a commitment, the success of which depends in the end on the organization or process that mints and maintains the identifier. Not all entities require this level of identification. However, an entity that is never referenced from outside of its local context would still require an identifier for local management purposes, subject only to local policies and procedures.

The conditions under which the changes to an existing digital entity are judged to be sufficient to declare it to be a new entity, and thus requiring a new identifier, are application and domain-dependent. Moving a data set from one location to another, for example, clearly seems not to be essential to its identity, as it is still the same data set. Moving a sensor, however, from one location to another might be seen as sufficient, as the core identity of a sensor might be seen as sensor type plus location. An assertion that two things are or are not the same must be made in the context of 'same for what purpose'.

An identifier may serve as a single point of reference to access a service that provides the required current state information as part of its service, including perhaps the digital entity itself. An identifier resolution system can be used as a late binding mechanism to connect current attributes to entities, e.g., current public key for a person or process.

Such an identifier system needs a method for dealing with fragments or subsets of identified entities, e.g., seconds N through M of a given video in digital form, where it would be impractical or impossible to assign unique identifiers for each potential fragment or subset.

Further, such an identifier system needs a method for associating related datasets to each other, for example, in the CPS/IoT, when data migrates from the edges of the network upstream toward the cloud and is aggregated/transcoded or when analytics is performed on the data resulting in a series of derived datasets.

Trust is a key issue in identifier resolution and takes multiple forms. On what basis do I trust that the resolution response received is indeed the response that was sent? On what basis do I trust that the resolution response reflects the data that was entered in the system by the party responsible for the identifier? And do I trust the information itself, i.e., on what basis do I trust the party that stands behind it? In a CPS context this includes the need for the identified data to have come from or be sent to an authenticated device.

The structure of the identifier string itself is of some importance. Experience has shown that building semantics into the string, while perhaps useful for minting and administering

identifiers, can be dangerous in that it can tempt people and processes to make unjustifiable assumptions about the identified entity. Any changeable attribute baked into the identifier itself, as opposed to the changeable record to which it resolves, results in a brittle identifier, e.g., identifying an entity by its location or ownership when either may change.

Although the TCP protocol was implemented to provide a virtual circuit mechanism, the notion of end-to-end in the Internet was never a requirement of the early protocol design work undertaken by Robert Kahn and Vint Cerf. As the Internet moves forward to embrace the IoT, however, substantiation of a data "endpoint" is still of some interest in a scalable, unified data identification system. In particular, temporal relations between elements become extremely important in CPS. Also problematic is a location-centric or owning-entity-centric structure. The core of many challenges in sharing and managing data lies in our treatment of data entities as second-class entities, existing without continuous and credentialed identification. This means that we have a paradigm of securing servers, and then managing access to those servers. A key weakness in today's technological landscape is PKI-based credentialing systems that do not allow for interoperability across trust domains. The method of credentialing is therefore an important issue in data interoperability.

There are two distinct classifications of identifiers – traceable and untraceable. The discussion above provides clear rationales for where traceable and navigable identification schemes are valuable. The Universally Unique Identifier (UUID) typifies a second class of identifier [147]. A UUID may be necessary when anonymity is required, often for privacy purposes. Application requirements must dictate which and when identifiers of each kind, or both, are required.

Finally, naming of data is an emergent topic. You should be able to name data in multiple name spaces. Namespaces are used to resolve (i.e. disambiguate) names that might otherwise appear the same. Naming schemes have to scale so that unnatural limits are not placed on the ability to name data. Names should also be human readable and logical to convey context. Naming of data should not tie data to the location where it originates unless this is part of the data itself. This latter point is critical for mobility of data.

B.5.3.3 The Impact of Data Volume and Velocity on Data Interoperability

As described in the sections on Data Fusion B.5.2.1, with CPS, a growing volume and velocity of data creation and transfer is occurring. This presents unique issues to data management. This section introduces these concepts.

B.5.3.3.1 Volume

With the ever-increasing volume of data being created by CPS and the IOT, there is an even more critical need to name, catalog, and describe data (e.g., through meta-data) in a manner that enables its easy discovery, stewardship, and combination or correlation with other data. This section describes primarily issues with handling volume of data. Note that issues of

Migration of Functionality and Data Processing and Transformation presented in the section on B.5.3.3.2 Velocity, apply equally to Volume.

Storage of data

A first concern however is that CPS may or may not have the ability to permanently store all the data generated. Thus, when there is a tidal wave of data, there is an attendant need for policies on if and how to store the data locally, when to expire the data, as well as if, how, and where to migrate the data. Additionally, the creation of large amounts of data underscores a need to keep separate (i) the registries that logically describe data and (ii) the actual physical repositories and/or caches where the data is stored.

Transmission of data

If data needs to be migrated elsewhere, for example for archival purposes or for sharing with other CPS, services or applications, transmission may be hampered by a mismatch between the volume of data being generated and the available network bandwidth. The implication is that the CPS will be unable to transmit the data in its original form. Therefore, data may need to be transformed (e.g., transcoded, sub-sampled, aggregated, compressed) to meet the constraints of the network. After this transformation, the newly derived data should remain associated or linked in some manner with its original self. For instance the original data and derived data could be managed as related objects in a Smart object framework, or in the Digital Object Architecture's Handle system, or a pointer to the derived data could be stored as meta-data in the original data handle [104][105]. Any policies associated with the original data (e.g., relating to access control, operating requirements, system constraints, SLAs, QoS), should remain intact. A real-world concern is that when data is copied to somewhere other than where it originated, it becomes difficult to impossible to ensure policies are being upheld or that data remains safeguarded. Witness the regular breaches to the databases of retailers, the UC system, banking institutions, etc. [173][171][172][174]. Consequently, there is growing interest in mechanisms, such as ABE (attribute-based encryption) that embed the access control policies in the data itself (through encryption), such that regardless of data movement (e.g., to a remote Cloud), access is prohibited (e.g., decryption fails) unless the policy rules are met [168].

Note that the transmission of CPS/IOT data upstream, from edge-of-the-network devices to a back-end cloud, may warrant multiple stages of transformation and storage. The high volume of data that is created either at the outset or when data is aggregated en route elsewhere may cause an N-to-1 implosion of data over the network, which the network and system components may not have been provisioned to meet. In fact, this mass data "inversion" and migration, sometimes called Reverse or Inverse CDN (content distribution network) may cause several generations worth of derived data, which may necessitate preserving the data lineage, continued association, and possibly a description of the function (and or inverse function) that captures the relationship between the original and derived data. It remains to be seen if certain more common data models might warrant standard types of transformations, which might

make the aggregation of disparate data streams more immediately interoperable or combinable. These same issues surrounding data stewardship over the data lifecycle are shared with other forms of processing that act on the data, lead to derivative data sets, and enable other CPS to take advantage of the results, e.g., aggregating/combining (similar) data into a single data set, performing data fusion to blend disparate datasets, transcoding of data to meet system constraints, etc.

Naming of Data

One consideration with data volume is the granularity at which data is named or to what level of data an identifier is attached, particularly for streaming data, which may grow ad infinitum, yet needs to support access and manipulation at a variety of levels relative to anchors in time or to other data features. While high volume data may initially warrant naming the data at a coarse grain, post-processing and analytics may be warranted to identify interesting events in the data stream, to tag the data and to improve its utility.

B.5.3.3.2 Velocity

High-velocity data presents its own set of challenges for data interoperability. When we refer to data velocity we mean data of a time-sensitive nature (e.g., requires delivery within a deadline) or the data is part of a time-sensitive control loop.

Migration of Functionality (vs Data)

A key disruption underway is that, despite the popularity of the Cloud, CPS and IOT systems sometimes have requirements that render Cloud solutions in a back-end data center unusable. Use cases that generate high-velocity data at the edges of the network are just such examples. They may not have the luxury of waiting for the Cloud to respond, because the Cloud may be too far away to meet the time-sensitivity requirements. The implication is that functionality (e.g., compute/analytics, storage, networking) that is normally offered in a back-end Cloud must be migrated to be more proximate to wherever the data is generated. For instance, instead of moving the data to be processed by an analytics engine in the Cloud, it may be quicker and result in less overhead to distribute the analytics algorithms (as executables) to where the data resides. The technique – of keeping the data stationary and moving the functions to the data - is also useful for protecting trust-sensitive data, e.g., data that is prohibited by law from being moved, as with certain kinds of healthcare data. The distribution of Cloud functionality and services to the network edge is referred to as Fog computing [170].

Note that when all data owners/managers deem their data immovable, there may be a need for brokers or arbiters to mediate fusion operations on the data. Although brokers may be a necessity, there is the potentially larger overhead time incurred when using them as 3rd parties to broker agreement and interaction between the data and the data processing among multiple entities, e.g., to preserve anonymity of interacting group members and their data.

Data Processing and Transformation

A side effect of the mere act of processing data – whether it is for aggregation, combination, transcoding, encryption, compression, analytics or fusion of data - is typically longer handling delays, and in some instances results in greater data overhead as well, both of which are further exacerbated by sheer data volume.

Take for example encryption. Although encryption is an important weapon in our arsenal to protect CPS/IOT data, the side effect of its use is typically longer data processing delays, particularly for decryption, and also greater data overhead. These concerns underscore the need to find algorithms and hardware to accelerate encryption/decryption and also to investigate other forms of processing that enable data to remain in the encrypted realm [169], bypassing decryption altogether.

Delays introduced by processing data can affect the CPS' ability to meet timing and/or synchronization requirements, especially for distributed analytics, where distributed components may require a synchronization checkpoint before agreeing to continue on with a task together. Processing of high-volume data may also be at higher risk of violating either timing requirements or time synchronization requirements. Fortunately, as mentioned in section 2.5 on Related Standards and Activities, standardization efforts are underway to try to ensure at least end-to-end awareness, if not guarantees, on overall time delays focused on Time Sensitive Networks (TSN) and Time Coordinated Computing (TCC), which will enable finer-grain time management for data when transmitted over networks and also aims to solve the "last inch" problem of time management within device platforms, respectively.

B.5.3.4 Data quality and provenance

The availability and exchange of data is of no practical use if its quality cannot be determined, and, if the source is not known or trusted. This section is a limited introduction to the standards which define data quality and provenance.

ISO/IEC 2382-1 [120] differentiates information from data through the following definitions:

- Information: Knowledge concerning objects, such as facts, events, things, processes or ideas, including concepts, that within a certain context has a particular meaning
- Data: Re-interpretable representation of information in a formalized manner suitable for communication, interpretation, or processing

ISO 9000 [122] defines quality as the degree to which a set of inherent characteristics fulfills requirements.

ISO 8000 [123], the international standard for data quality, defines quality data as data that: (1) references a syntax, (2) is semantically explicit, and (3) meets stated requirements. By its very definition quality data are portable data (explicit syntax and explicit semantic encoding).

ISO 22745-30 [124] is the international standard for stating requirements for data in a computer-processable form using an open technical dictionary.

ISO 22745-40 is the international standard for the exchange of characteristic data in a computer-processable form using an open technical dictionary.

ISO 8000 data quality can automatically be assessed by comparing ISO 22745-40 data to an ISO 22745-30 data requirement.

ISO 8000-120, the international standards for quality data with provenance, requires that provenance be provided for all characteristic values. Provenance is the identifier of the organization that provided the data, and the date and time the data was extracted. Provenance must be provided at the data element level, and not at the record or exchange level.

Quality data relies on a concept dictionary for semantics. A concept dictionary will contain the explicit definition of all encoded concepts to include metadata and code lists (reference data). A metadata registry typically only includes attributes (name of the characteristic) and their definitions, but a concept dictionary also includes code lists.

An example of a code list is a state code – CA would be a possible value. It needs to be defined in a dictionary as CA=California.

B.5.3.5 Data Service Patterns

CPS interact typically through communications of some sort. The interaction is described in terms of an interface. Data Services are those interfaces specifically focused on interacting with or exchanging data. In this section, the predominant information exchange patterns used in CPS interactions are introduced. For any given CPS to CPS interaction, designers might specify one or more of these interaction patterns based on their understanding of the complete set of relevant aspects and concerns described in this framework. Various protocols in common use for CPS make use of one or more of these patterns. Refer to the following figure, Figure 33, which illustrates the exchange patterns described in this section.

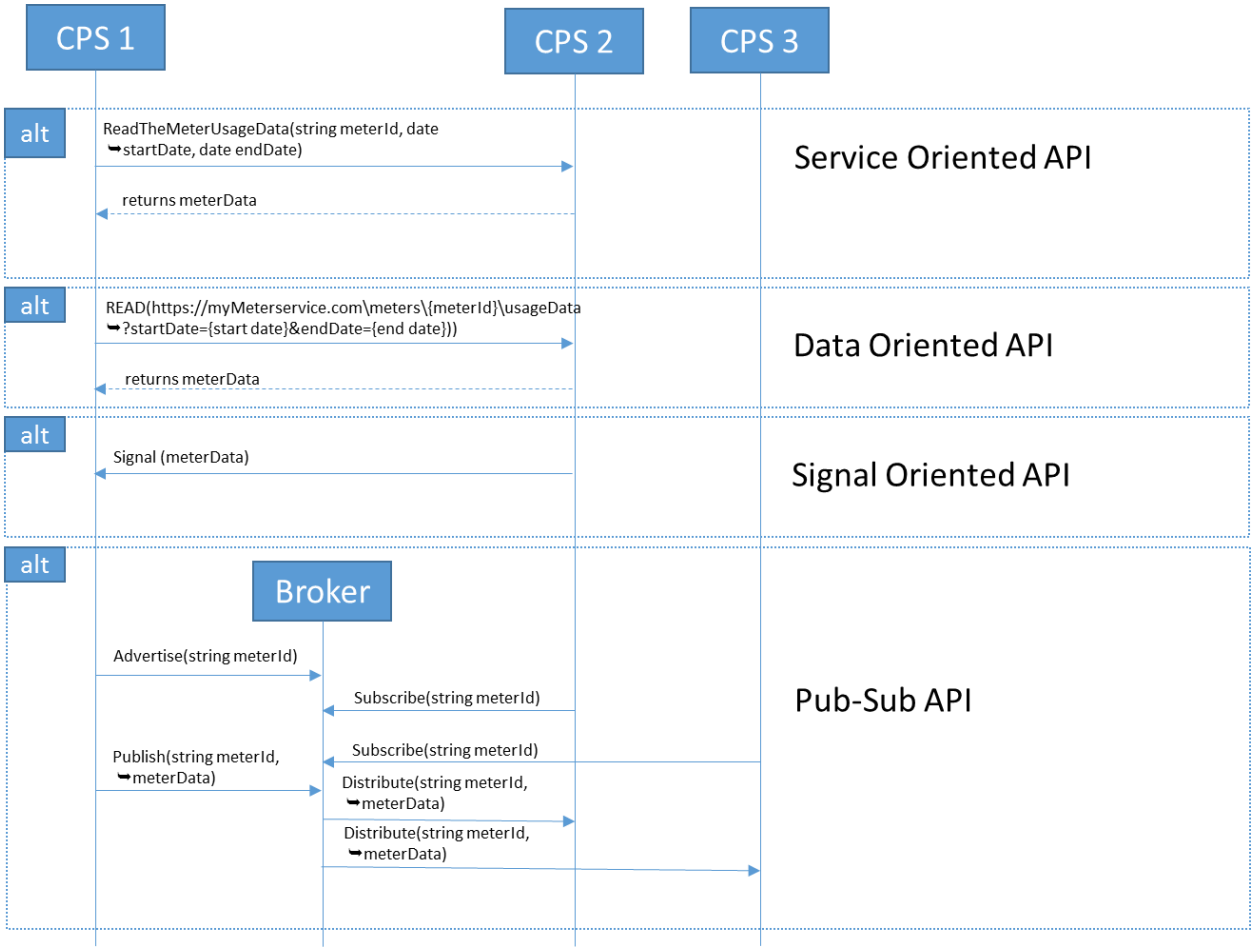


Figure 33: Common Data Services

The figure illustrates four alternative service models by which essentially the same data may be exchanged among CPS. In all four examples it is the goal of the service to provide “meter UsageData” to the recipient.

Data services are typically provided using Application Program Interfaces (API)s. There are many kinds of APIs. This section describes two variants of the Request-Reply model, an event driven data exchange model, and a publish-subscribe model.

A simple request-reply service provides an endpoint and query parameters. Together, this results in a “remote procedure call” typical of service oriented architectures [161]. The endpoint identifies the service, and the query parameters represent the arguments, or signature, of the service. An example of such a service provided by a CPS that is responsible for utility meter data management might be:

ReadTheMeterUsageData(string meterId, date startDate, date endDate)

Readily observed from the service description implied by the example, this service will read the usage data from a meter given a meterId, and a start and end date. The attractiveness of service oriented architectures is that data structures for both requests and responses are "strongly typed." That is, data structures must be known at design-time or discovered at run-time. Strongly typed interfaces are less prone to improper use but are less flexible.

Another kind of API is a data oriented architecture [161] where complex data structures can be navigated via references to data, and query parameters are used to filter results returned as a data set. Additionally, they use "common data services" that allow for a limited "reduced instruction set" of service methods. The attractiveness of data oriented architectures over service oriented architectures is that once the data structures are understood, the API can be more easily used for purposes not envisioned by the originators of the API. With a service oriented approach, the limits of the service signature can constrain the access to the data. This is especially the case for complex data (highly structured and nested data sets). An example of such a service might be:

> READ(https://myMeterservice.com\meters\{meterId}\usageData?startDate={start date}&endDate={end date})

In this example, a path to the UsageData is provided using the generic READ service. The constraints of start and end date are provided as "query parameters" which are general arguments that can be applied to virtually any GET service.

The complexity of implementing a data-oriented architecture will be comparable to a service-oriented architecture if the services are designed to allow for data filtering based on arguments to the service definition. For very simple services that are tightly targeted at accessing only a specific data set, the service approach will be simpler, although not extensible without defining new services or service arguments.

The potential benefits of data oriented architecture stem from the use of "reduced instruction set computing" similar to that in microprocessors. These include a deterministic set of message handlers due to the distinct data-oriented nature of the services – typically Create/Read/Update/Delete (CRUD). Of course once message handlers have validated the messaging, interpretation of what to do remains. On the other hand, the "what to do" of a service is explicit in the design of the service.

Request-reply messaging may be stateless or stateful. That is, the message exchange may occur in a single transaction or in a series of transactions. Another dimension of several implementations of data exchange patterns is the ability to discover the availability of data by type or instance and the ability to provide the type description of the data that can be acquired.

Service oriented services are commonly, although not exclusively, presented using Web Service Description Language (WSDL) [163] and implemented via Service Oriented Application Protocol (SOAP) [165].

Resource oriented services are commonly, although not exclusively, presented using Representational State Transfer (REST) [164].

A third pattern of data exchange is via unidirectional signals. In this case, data (or alternately a stream of data) is transferred from source to destination. Since the target of the data simply needs to receive the data provided, it doesn't need to ask for it or when event driven the data becomes immediately available when it is ready.

A fourth common service model for data exchange is the "publish-subscribe" model. In publish and subscribe a source of data first registers with a broker and "advertises" data sets that it will provide. It then "publishes" the data to a "broker" service where the data is known by a unique identifier tag. Nodes that are interested in the data behind the tag "subscribe" to it with the broker. When the data provider publishes the data to the broker, the broker in turn transfers the data to those who have subscribed to it. There are variations on how the availability of data is advertised or discovered and how subscription and delivery occurs.

There are two primary approaches to publish-subscribe messaging – using a broker as a middleman [166], or, implementing the broker as a feature of each publishing node [167]. The advantage of the former is that publishers and subscribers have minimal responsibilities. The advantage of the latter is that no external and trusted broker is required. There are benefits and tradeoffs to each approach.

There are many application layer protocols that can be used to implement the variations on common data services enumerated in this section. Alternatives will vary in both style and syntax by which the messages are encoded. This section provides an overview of the different types of data exchanges in common practice.

B.5.3.6 Governance

Data governance[25] is the collection of stated rules, regulations, and policies that govern data. Data governance is associated with a system of decision rights and accountabilities for information-related processes, executed according to agreed-upon models that describe who can take what actions with what information, and when, under what circumstances, and using what methods.

Data governance covers all data, as shown in Figure 34.

[25] Note that the term "data governance" has little to do with legal and regulatory issues and is mainly concerned with enterprise-level policies and procedures.

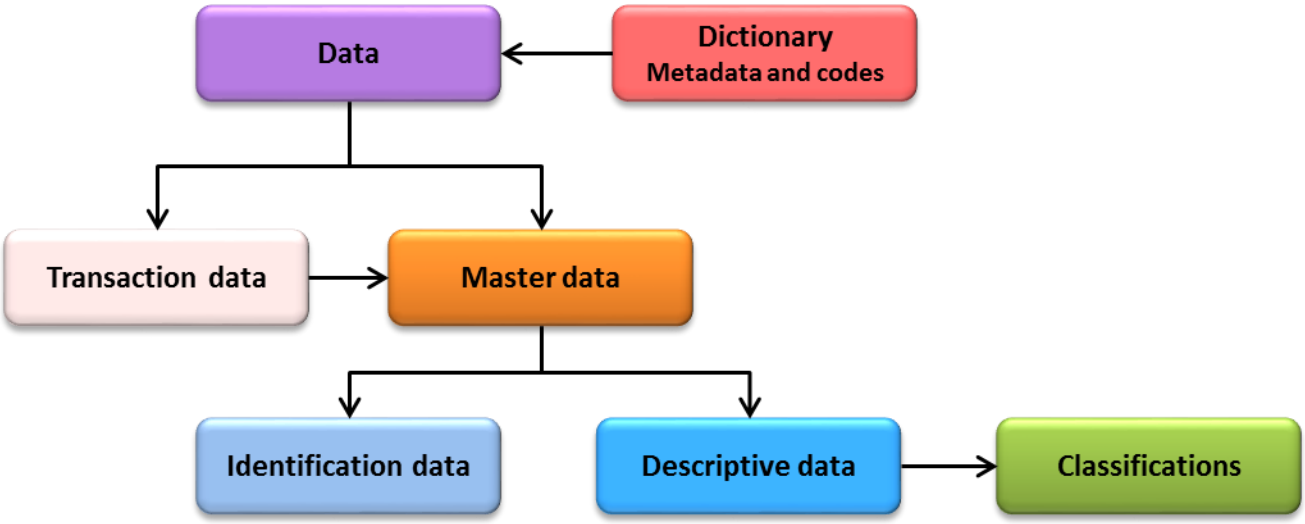


Figure 34: Taxonomy of Data

Master data are defined as "data held by an organization that describes the entities that are both independent and fundamental for that organization, and that it needs to reference in order to perform its transactions." [122] Examples of master data include records that describe customers, products, employees, materials, suppliers, services, shareholders, facilities, equipment, and rules and regulations.

For CPS and data interoperability, the information exchange by the CPS is described as *transaction data* that is dependent upon the quality of the master data. A key requirement of data quality for CPS, in addition to syntactic and semantic definitions, is the notion that the data are portable; the data are application independent.

B.5.3.7 Cybersecurity and privacy

This section discusses the relationship between cybersecurity, privacy, and data interoperability.

Cybersecurity and privacy are often discussed using measurements of confidentiality, integrity, and availability, each holding more or less importance depending on the environment. Without comparing value, this section uses these anchor points to address traditional data interoperability issues with cybersecurity and privacy.

Confidentiality is obviously vital for privacy, as well as for control of information and the system itself. Control of information can make certain attacks (physical and cyber) on an entity more difficult to plan and execute successfully. Control of the system itself is vital for data integrity, which we'll talk about next. Standard solutions to confidentiality involve encryption. Once a CPS platform is compromised, data in transit protections are circumvented. Therefore, it

essential to protect the confidentiality of data at rest (i.e. where it is stored) as well as in transit.

Encryption is only as good as the implementation of its algorithm, key exchange between parties, and key data storage. If any of these is poorly implemented, an attacker may be able to compromise the encryption, potentially leading to breach of privacy and/or control of the system.

Integrity of a given system is vital for trusting any of the data or behaviors the system provides. Attacks (e.g., credential compromise, memory corruption exploit, man-in-the-middle attack) that allow for unauthorized modification of the information maintained by the system, or control of the system, jeopardize the value and trustworthiness of the system. For instance, if a system generates, transports, or interprets sensor data from power equipment in the field to a control center, modifying that information along the path could lead to disastrous decisions by the people consuming the information. Likewise, if information about a crop report is intercepted and modified before being delivered to the agricultural market, decisions would be made that could destroy an entire portion of our society's food chain.

Typically, authentication and authorization are used to ensure correct controls over a system, and cryptographic integrity checks (aka digital signatures) ensure data has not been altered since creation. In addition, most networking layers provide integrity checks, but these are intended to identify accidental bit errors, not to keep an attacker from modifying the data. Authentication is the art of ensuring the identity of an actor on a system. Several common methods are used to verify the identity of an actor, including passwords/shared keys and multi-factor authentication, which attempts to make impersonation more difficult. Passwords/shared keys mean that both sides have some type of pre-shared data. These passwords can be stolen if stored on a compromised device, and in many cases, they can be guessed and/or cracked offline. Multi-factor authentication attempts to ensure that the entity has at least two of the following: knowledge of some pre-shared key, some offline device, or some biometric evaluation. Multi-factor is currently only good at identifying human entities since it relies on the interpretation of something that is not network-attached (thus more difficult to compromise), but the key value of multi-factor is that an attacker must overcome multiple hurdles to impersonate an entity on the network. Best practices for each of these involve cryptographic means to verify the identity of a given entity, such that information is not immediately compromised over a network by an attacker who may be capturing and analyzing the data, and verifying that data actually comes from who the system says it comes from.

Authorization is ensuring that a particular entity is allowed to be performing an activity. This verification allows a system to have many verifiable entities, each only allowed to perform certain tasks under certain conditions. This concept of constraining information on a “need to know” basis is also known as the *principle of least privilege*. For data authorization this might mean that access control policies have been associated with the data and the policies specify

who is authorized to gain access to the data under what circumstances (e.g., when directly attached to the corporate network, but not when connected through a corporate VPN).

There are numerous methods of verifying that data has not been modified in transit, including cyclic redundancy check (CRC), checksums, and any given hash (MD5/SHA256/etc.) of the data. However, these methods only provide protection from accidental modification. An attacker need simply re-<method> their modified data and pass all checks. For this reason, cryptographic integrity checks (aka digital signatures) were created to ensure that the calculation of any integrity check was based on information only maintained by the original sender. This type of check has been integrated into most common encryption schemes to ensure both confidentiality and integrity of the data – assuming no compromise of the information used to sign/encrypt the data.

Availability means that a system or data are accessible as needed or desired. This data or system may provide important information for a given process or may be part of a designed system of trust. For example, TLS, as used in HTTPS and other encrypted services, uses cryptographic certificates and a PKI. This PKI uses a Certificate Revocation List (CRL), which is often just a web page with a list of certificates that are no longer trusted. If that CRL is not available when a TLS-enabled service is accessed, known compromised keys will still be considered valid because the mechanism required to verify that a certificate has not been compromised is unavailable. From a process control standpoint, if a system is unavailable during manufacturing, chemical mixing, power drains, and a myriad of other physical events, products can be destroyed (or simply not produced), chemicals may explode, electrical components can be damaged, and otherwise "bad things" can happen. For this reason, control systems engineers tend to favor availability over anything else, whereas common IT engineers tend to favor confidentiality and integrity primarily and consider availability more valuable when money and reputation are involved.

Availability is ensured through careful design and use of redundancy. Poor design can leave many single points of failure that lead to services and data being unavailable when needed. Proper design of a system includes sufficiently redundant network connectivity, identifier name resolution (if necessary), and in many cases, redundant services and data. Services themselves may be provided behind a load balancer or use some other failover method (which itself then has redundancy). Data may be served by one of these redundant services, and be mirrored between different storage media, providing further redundancy and availability. These are potentially complex solutions that require deep knowledge and understanding of their technology, which also has to be considered in proper design. Many OT devices do not have the luxury of redundancy because they were designed before redundancy technology was cost-effective. The measures that provide redundancy in these legacy systems tend to be nonstandard and difficult to work with.

Data interoperability and cybersecurity are significantly intertwined. Cybersecurity requires that both sides of communications understand and agree upon the security and privacy

protocols in use for communications to take place. This communication is a key part of availability. When CPS are composed as systems of systems, there is the need to align and/or broker the data across the interfaces between components, particularly when the data crosses a private-public boundary, transits between different domains under different administrative oversight or flows between components with different owners. Thus, components that exchange data must have access control or usage policies that are compatible.

What good is data if you cannot trust it? And why is data trustworthiness so important to CPS? In CPS, the physical world may be actuated in response to data generated or analyzed. Thus, to achieve trusted actuation, trusted decisions are needed, which in turn depend on trusted analytics, which in turn require trusted data. Ensuring trusted data begins as a function of the trustworthiness of the physical device that created the data and then continues as a function of one's ability to ensure the security of the data throughout its lifecycle. In fact, data interoperability becomes meaningless if the data are not transmitted, used, and stored securely. Data trustworthiness also may be impacted by the aggregation, transcoding, compressing, sub-sampling, or any form of alteration of the data. Data terms related to cybersecurity discussed include:

- Certificate
- Certificate Revocation List (CRL)
- Checksum and CRC
- Credential
- Cryptographic certificate
- Cryptographic hash
- Cryptographic key
- Digital signature
- Hash
- Key data storage
- Password
- Pre-shared key
- Signature

B.5.3.8 Data about timing and timestamps

Many data require timestamps reflecting when the data were created. For example, a sensor of a moving part in a motor might need to take data at a regular rate, and each data point would need a timestamp with enough accuracy with respect to the appropriate reference time scale to make the data useful. There are several issues here:

1. The short-term stability of the timestamping clock is determined by the local oscillator. For improved longer-term performance, this oscillator may be locked to an external reference. With an external reference, requisite stability up to the loop time constant is the requirement and the loop time constant is, in turn, influenced by the level of noise (such as packet delay variation) in the external reference as received. Without a sufficiently accurate and sufficiently stable external reference, the local oscillator needs both accuracy and stability; note that these two are rather independent requirements. A significant trade-off here is that the better the oscillator, generally, the more size, weight, power, and cost it may demand.
2. The quantization error of the timestamp is determined by the least-significant-bit (LSB) of the counter and the impact of the measurement front end that feeds it. This, along with clock instability, is the source of stochastic noise on the timestamps. In some cases, the quantization error can be synthetically reduced by adjusting the sampling phase.
3. A stable but inaccurate timestamping oscillator produces a deterministic offset in the data collection rate. If this can be measured, it can be removed. This measurement generally requires an external reference.
4. Traceability of the oscillator is a function of the time-transfer accuracy from the reference timescale. If data need to be correlated between nodes, a common reference timescale is required. Often this is best done using an international timescale such as UTC or TAI.
5. Missing data need to be accounted for. If the user of the data is expecting data at a certain rate, there needs to be a method of acknowledging missing data for the user to maintain the correct data rate.
6. Formats used to write or create timestamps can cause serious issues. Consider in a networked system of possibly dissimilar nodes, the potential for different timestamp formats (e.g., 48 bits versus 64 bits or the order of significance reversed, high to low versus low to high) as well as varying granularity of timestamp clocks. One system might generate timestamps at 40 MHz and another at 250 MHz. The period of the slower clock allows for greater local oscillator error influence.
7. Translation among reference timescales in any networked system of shared timestamps is mandatory.

These issues suggest the need for the following parameters for data timestamps:

1. The nominal data rate
2. An indication if data are missing at a regular measurement time
3. Enough significant digits in the data and timestamp to meet requirements
4. The stochastic uncertainty of the timestamps
5. The deterministic uncertainty of the timestamps
6. The traceability accuracy and reference timescale
7. A formalism for resolving differences among timestamp formats
8. Perhaps a period of validity and/or expiration date of the data

Timing data can contribute to security and monitoring issues, for example, knowing that a user cannot be in two places at the same time. Accurate timestamping can contribute to root-cause-analysis of when a failure or incursion happened somewhere in a network.

B.5.3.9 Safety and configuration assurance

Design and implementation assurance is an important part of CPS with regard to safety, reliability, and resilience. It is essential that, for any given CPS component, it can be verified to some level of certainty that the system conforms to required levels of safety assurance.

The Assurance Facet, section A.3, describes the nature and importance of assurance for CPS. An assurance case is met prior to the commissioning of a CPS or continually through surveillance. Maintaining the state of CPS is highly dependent on the ability to verify the configuration and the detection of tampering or damage to the data that govern proper operation.

There are two key dimensions to this that pertain to data interoperability:

1. Determining that the software running on the CPS device is indeed that which is believed to be running, and,

2. Determining that the running configuration is as established by authorized configuration management software, policies, and procedures.

Software images are typically verified through secure hash checksums that ensure that the code in firmware is as expected by design.

Changes to CPS device configuration can be managed through event recording of changes and the maintenance of a change history. This ensures that a record is built. One may need near-real-time monitoring and control to actually manage changes

ANSI C12.19 [146] is a standard used throughout North America for automated meter reading. This standard tackles these issues from the perspective of data interoperability with a function they called “Event Logger.” The principle used is that configuration changes that can be made to what is essentially the cash register of the utility must be tracked and auditable. Further, since communications can be intermittent, and changes can be imparted locally or remotely to

such devices, a persistent record of some depth must reside within the CPS device itself, with a larger, less limited record "spooled up" to the owner – typically the utility. A series of secure hashes and timestamped event records are performed which guarantee that any current state of the CPS device can be re-created by executing the logged sequence of changes and only in the order that they were recorded.

Many CPS devices provide for configuration management through communications interfaces. Inadvertent, incorrect, or malicious changes can cause havoc in a CPS, depending on the role of such a device in the system. Therefore, best practices on the version and state control need to be specified for many components of CPS. A future CPS Framework User's Guide should include more specific procedures and examples of the best practices, as in which types of components to protect in different types of CPS.

B.6 Timing Aspect

This section describes the timing aspect of the CPS Framework. The components of this section are as follows:

- Section B.6.1, which provides an overview of the timing aspect, discussing fundamental concepts needed for understanding the subsections that follow.
- Section B.6.2, which presents the current status of, and needs for, time awareness in system elements of a CPS.
- Section B.6.3, which discusses timing and latency in CPS. Latency is a core concept for timing in CPS. Latency is a critical issue in all CPS, but is especially critical where control systems span several nodes with significant spatial separation, and especially in SoS or any systems that include cloud computing or virtualization technologies in the control system. Also, the temporal relationships between acquired data (e.g., simultaneity) are of paramount importance. The challenges of predictability in software are increased by the non-determinism of the layers of software managing data transfer and non-determinism of the network connecting these nodes.
- Section B.6.4, which discusses special security issues that arise with timing. General trust disciplines relating to CPS include security, resilience, safety, reliability, and privacy. Timing plays a key role in many of these and thus the provision of secure timing raises specific challenges relating to security and resilience. Security of a timing signal requires security of both the physical signal and the data associated with the signal. Security of the data in a timing signal is similar to other cybersecurity problems. Security of the physical signal brings in a number of aspects unique to timing. The user is typically remote from the source of the timing signal representing the particular system timescale. For security, the user needs to know both that the physical signal came from the correct source, and that the transmission delay has not been tampered with. In addition to these two aspects, denial-of-service can be created for timing signals in a number of ways.

B.6.1 Introduction

There are many aspects to timing, but, fundamentally, all timing includes a physical signal. The physical signal may be accompanied by data, which describes it or is meant to be used with the physical signal. The physical nature of timing is at odds with the way data systems work, leading to core difficulties in CPS. Data systems, computer hardware, software, and networking have been optimized by abstracting away the timing properties of the physical layer. These systems all isolate timing processes, allowing the data to be processed with maximum efficiency due in part to asynchrony. However, coordination of processes, timestamping of events, latency measurement, and real-time control are enabled and enhanced by a strong sense of timing. Positioning and timing are strongly interrelated. CPS involve a marriage of the cyber and the physical: a marriage of data networking and processing systems with systems that live within the laws of physics. Generally speaking, CPS currently overcome this fundamental conflict of modern system design by using dedicated hardware and customized software for timing-critical systems. Things that require strong temporal determinism are processed as much as possible with systems that do little or no data processing. However, in many cases CPS must include significant data processing. Here, both software and hardware must be reliably shown to ensure agreement with timing specifications. Changes or upgrades to hardware or software may create a need for re-calibration of the entire system.

This section (B.6) of the document discusses the status of current systems and points out problems and new directions that are currently in development. A later document will more fully show a roadmap for future timing systems.

B.6.1.1 Types of timing and timing requirements

There are three different types of timing signals for synchronization: frequency, phase, and time. Accurate frequency can be supplied by an individual clock, a cesium standard, though practicality drives the use of oscillators that require calibration and active reference signals. By contrast, phase and time synchronization *always* require transport of signals and perhaps data. Unlike the transfer of data, the transfer of time and phase requires compensation for the transmission delay of these timing signals to the required synchronization accuracy. For example, GPS provides positioning by sending synchronized time signals from known locations in space. The transmission delay is on the order of 70 ms. To provide ranging accurate to 1 m, the true delay must be removed to better than 3 ns, a factor of about 1 part in 20 million.

Data often accompany physical timing signals, though phase synchronization may not need it. The simplest timing data are for time, sometimes called *time-of-day*, where the signal indicates when the time information is correct, but the actual date and time-of-day of that time signal must be transferred as data. In this case, the time signal is sometimes called the *on-time marker*. The time *data* can be transferred with significant noise and latency, as long as when it arrives it is clear which on-time marker the data refer to. Depending on the applications, many

other data may be associated with timing signals. For example, a quality level of the source clock is often required with timing data.

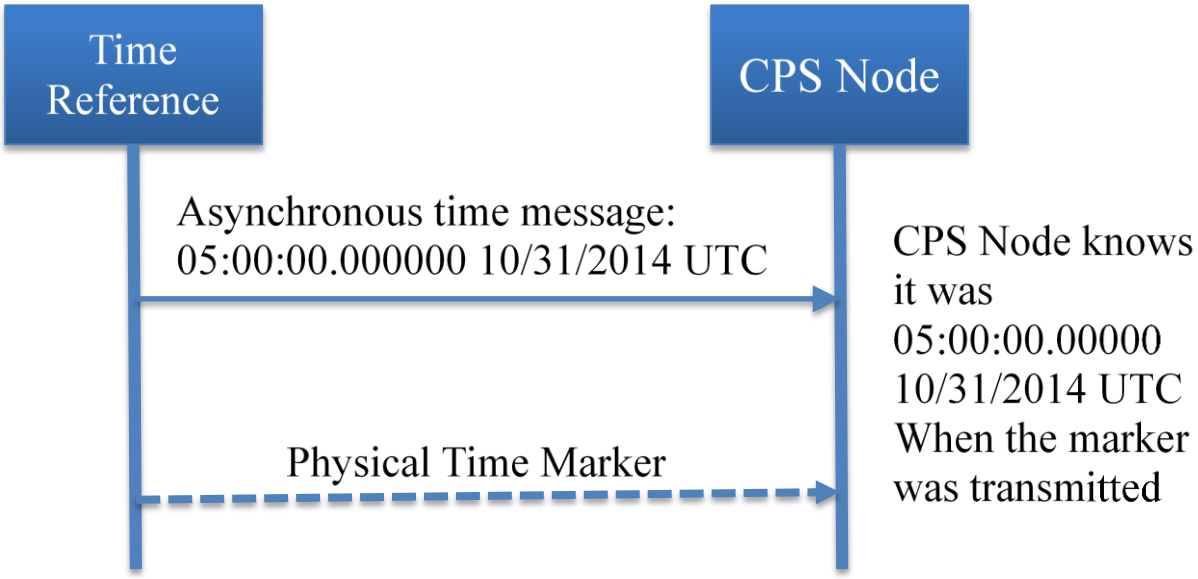


Figure 35: On-Time Marker

Figure 35 is an illustration of the relationship between the physical time signal and the associated data, which is an asynchronous time message in this case. Note that the time of arrival of the marker is the transmission time plus the delay. The CPS node will need to either know or cancel the transmission delay commensurate with its time accuracy requirements.

Synchronization through networks will generally involve the transmission of such time markers and data using a two-way time protocol to cancel the delay through the network. Two-way time transfer is discussed in the accompanying Draft Timing Framework for Cyber Physical Systems Technical Annex, Release 0.8 (Timing Framework Annex) Section 1.1 [178]. Common protocols for this are the Network Time Protocol (NTP) [217] and the Precise Time Protocol (PTP) [184] [185] [186] [187]. Other protocols are discussed later, in Section B.6.2. Systems with timing requirements that are coarse enough that the time-transfer delay is negligible will not need to cancel or remove the transmission delay.

A specific set of CPS nodes will be synchronized against a single reference timescale forming a CPS synchronization domain, some of which are the CPS domains as described in Section B.6.3. Section B.6.3 also discusses how timescales will need to be synchronized across domains if they need to coordinate functions such as timestamps of data or control. This will apply to all forms of synchronization depending on what is needed for the specific CPS function: time, phase, or frequency synchronization. Synchronization across domains can require more care if the systems are connected through a cloud or across a network with virtualization. The impact of

new networking paradigms such as Software Defined Networking (SDN) on timing performance needs to be carefully considered as does the role of Network Function Virtualization (NFV), as discussed in Section B.6.2.4.

CPS timing requirements can be specified in terms of the time interval between significant events. The concept of a time interval specification implies that the system supports a timescale against which intervals can be measured (timescale is defined in [175]). A timescale is characterized by two features: the epoch (which marks the origin, i.e., time zero) and the rate at which time advances (typically the definition of the second).

The concept of a "second" is defined in the International System of Units (*Système International d'unités*, SI) developed and maintained by the International Bureau of Weights and Measures (*Bureau International des Poids et Mesures*, BIPM), in terms of energy levels of Cesium atoms. Thus, a clock is accurate (in frequency) to the extent its rate agrees with the definition of the second. Time is accurate if it is traceable to UTC or TAI. TAI is the timescale called International Atomic Time (*Temps Atomique International*), which is generated by the BIPM with the rate that best realizes the SI second, and the time origin determined by the transition to atomic time from astronomical time in 1958. UTC is considered "discontinuous" due to leap second adjustments. These are inserted into UTC to keep it within 0.9 seconds of UT1, the time scale linked with the Earth time. Note that any real-time UTC or TAI signal is only a prediction of the exact value, since UTC and TAI are post-processed time scales [176]. A table in the timing appendix [178] identifies some of the timescales in use and the choice of time origin (epoch).

In many CPS systems, the timescale need only be self-consistent, with no requirement to agree with an international timescale such as UTC. However, due to the inherent communication infrastructure of the IoT, some level of accuracy of time that is traceable (traceable is defined in [175]) to an international scale such as UTC [176] will often be available, though perhaps not at the accuracy the system requires. Thus, in many systems, the precision timing of the epoch is an application-specific event (e.g., when the power was turned on), and the rate is typically a count of the oscillations of a local oscillator in one of the nodes. In other systems the timescale is required to agree with an internationally defined timescale, e.g., UTC or TAI [176]. In this case the rate must be the SI second. The Timing Framework Annex Section 1.1 [178] contains a detailed discussion of timescale issues and metrics.

Equally important aspects of CPS timing are predictability and determinism. There are two aspects to determinism. The first, and the typical computer science meaning, is that a system is deterministic if for the same set of input values and system state (ignoring timing) the resulting output values and system state are always the same. Thus for example, 2+2 is *always* 4 and the command "initialize" *always* puts the system into a defined initial state. This is clearly a requisite property for CPS systems. However, CPS systems often require *temporal determinism*, i.e., identical or at least very similar timing behavior. Due to inherent variability of execution time on modern high-performance architectures, system significant time intervals can only be identical (deterministic) if identical input, identical initial architectural state, and the absence of

external interference can be guaranteed. Issues of temporal determinism are discussed in Section B.6.2.

Timing predictability means that the timing behavior can be predicted within appropriate parameters that a specific system requires. This is discussed in more detail in the Timing Framework Annex Section 1.1 [178]. To the extent the timing is predictable, it can be predicted at any future time, given the initial values of input and state. The BIPM has developed a standard method for determining uncertainty by breaking it into type A, typically the statistical uncertainty, and type B, typically a deterministic uncertainty, or an uncertainty of how large a bias there may be in the data [176]. Thus, uncertainty is in a sense the opposite of accuracy, i.e., uncertainty is the amount of inaccuracy. An example of this is in the IEEE 1588 protocol, or PTP. Short-term noise is caused by packet delay variation (PDV), also called jitter. This would be a type A uncertainty, i.e., it is a statistical uncertainty. Asymmetry in the delay between the two directions of timing packet transfer causes a constant time error in the resultant time transfer. This would be a type B error; it cannot be seen in the measurements, even with a very small standard deviation in the stochastic effects. Thus, an estimate of the magnitude of the asymmetry would be part of the type B uncertainty. Timing uncertainty is discussed in detail in the Timing Framework Annex Section 1.1 [178].

B.6.1.2 Event versus time-triggered measurements

Two common execution models are event-driven and time-triggered measurements. Both cases require that the number of interactions with the physical world be bounded so that controller computations can be completed by application specified deadlines. In the time-triggered architecture a defined set of interactions with the physical world is initiated by the controller, generally using a periodic cycle of sense, compute, and actuate – hence the name time-triggered. In a distributed system, communications between nodes are also scheduled. Scheduling is based on a common timescale usually implemented via IEEE 1588, NTP, or similar, perhaps proprietary, protocols.

In an event-driven system, external events or controller-initiated interactions with external physics, i.e., the plant, are permitted. In this case constraints on the number and frequency of external events must be imposed by application-specific methods. This is required to prevent these events from overwhelming resources. In a distributed system, a common timescale is used to timestamp external events. The determination of whether time-triggered, event-driven, or possibly some other model is used is highly application-dependent. Some systems may use a combination of event and time-triggered measurements.

B.6.1.3 Ordering of timestamps

Caution must be used when ordering events based on their timestamps. If the two timestamps A and B are within the timestamp resolutions or within the synchronization error of the time scales, then it is not possible to determine with confidence which event happened first. This is a

general problem of comparing any two physical measurements. The situation is also complicated if the two time scales do not have the same resolution. If ordering cannot be resolved using timestamps with sufficient difference, then another method must be used, typically an arbitrary order such as lexical order based on variable name. There are methods to avoid this situation, at the expense of confounding otherwise separate closely spaced events. If it is possible to implement a sufficiently fine granularity to timestamps, this confounding can be avoided, e.g., see the sparse time proposal by Kopetz [179].

B.6.1.4 Position and time are coupled

Since time and phase require compensation for delay, the position of the end device is intimately connected to time transfer. On the other hand, position and navigation are generally accomplished today using time signals that learn known locations from synchronized clocks. Thus, position and navigation are mutually dependent on synchronization. GPS and GNSS are commonly used for obtaining both time and navigation or position, but there are many limitations to these systems. GNSS signals are very weak. They cannot penetrate buildings well. They are vulnerable to intentional and unintentional interference. Many CPS will need to receive timing signals through networks. For example, PTP can provide synchronization with a timing source (such as GPS or a rubidium clock) through either a wired or wireless connection to the physical and MAC layers. A key issue is to make these timing signals available through all of the layers of the network stack including the application layer.

Similarly, location of nodes could be determined independent of GNSS by determining time-of-flight of a signal between the target and each of at least three sensors, and then using trilateration to estimate the target's 3D coordinates. For a sensor, it is necessary to know the spatial position and the relative phase (time) of timestamp generation with respect to a common reference. To achieve trilateration, these sensors need to have their coordinates known each with respect to the same reference coordinate system, and their time stamps within the same time scale.

In most existing CPS, location information is specified by proxy, such as a logical location (e.g., room 21,) or a network attribute, such as IP address. In traditional navigation systems, location is determined by services such as GNSS and specified by coordinates such as longitude and latitude. There are strengths and weaknesses to both methods depending on the application. Future applications, particularly in the IoT, will likely require a seamless method of handling both proxy and explicit location specification very analogous to the issue of combining timing domains as discussed in Section B.6.3.

Establishing time and position would enable spatiotemporal reasoning necessary in intelligent distributed applications. Examples include self-driving cars, collaborating car highway interaction, better inventory and delivery control, navigation applications, threat location, 3D camera, etc. An application could specify a time-space region for a set of cooperating CPS nodes. The application could establish the time-space region by the enumeration, discovery, or

presence of the set of collaborating CPS nodes and by rejecting those outside the boundaries. Once a time-space region is established, applications and possibly higher authorities could exclude other applications from interfering within this region perhaps for security or for managing resource allocation. Finally, a time-space region could also specify latency guarantees among cooperating CPS nodes in the region. For example, collaborating CPS nodes could measure communication latency in a region, which could then be advertised and used as a basis for adjusting applications.

B.6.1.5 Benefits introduced from timing

Timing is inherent in CPS. Precise timing capability in a CPS can enable better control and provide more robust correlation of acquired data, which may in turn permit CPS that have large spatial extent and/or higher degrees of complexity, such as the telecommunications network, the power grid, or future distributed systems.

Perhaps more significantly, the increasing use of time in both networks and the nodes themselves, holds the possibility of designing CPS that are correct by construction. In the future, the presence of appropriate support for time will lead to new and more robust designs for the applications themselves. Both these points are discussed in Section B.6.2.

Accuracy in timing may mean many different things. Besides the different types of timing (frequency, phase, and time), there are many orders of magnitude of variation in timing requirements. These are illustrated in the Timing Framework Annex Section 1.4 [178].

In the absence of a time-aware CPS architecture that infuses appropriate timing into the components on which applications are built, today's CPS are increasingly being rolled out with many limitations due to the lack of availability of precise time. For example, there are inabilities to update software or hardware in systems that require accurate timing without extensive recertification of timing. Another significant limitation is the inability to correlate data, such as the significant difficulty to determine event sequences after the 2003 northeast North America power blackout. Emerging CPS application domains that may benefit from enhanced timing include smart systems (grid, cities, buildings, transportation systems), location-based systems, medical devices, environmental monitoring, and entertainment.

There is an urgent need to revisit conventional Information and communications technology paradigms so they maintain appropriate time awareness, such that next generation CPS will not be held back by design and engineering constraints. This will signal an era whereby CPS will have the potential to transform lives by facilitating huge performance leaps in existing application domains and setting a foundation block for as-of-yet unheard of domains.

B.6.2 Time Awareness in CPS

This section examines the components of a CPS from the perspective of the presence or absence of time in the models used to describe, analyze, and design CPS and in the actual operation of the components.

Such systems take many forms and have diverse timing requirements as indicated in the Timing Framework Annex Section 1.4 [178]. Timing requirements are generally expressed as constraints on the time intervals (TI) (the duration between two instants read on the same timescale) between pairs of system significant events. For example, the TI between the acquisition of a sensor reading and the time at which an actuator is set as a result of that reading may be *specified* to be 100 µs±1µs. Similarly, a bound may be required on the TI (i.e., the *latency*) between when a sensor measurement event actually occurred and the time at which the data was made available to the CPS. Latency can vary in time and also vary by system layer. Latency specifications are generally time limits on deadlines, though there can be other requirements such as jitter limits. Likewise, the accuracy of event timestamps is a constraint on a TI, in this case between the actual time of the event and the value of the timestamp.

Constraints on TIs can be categorized based on their degree of time awareness in terms of bounded, deterministic, and accurate TIs. *Bounded TIs* are required for CPS with timing behavior based on deadlines. *Deterministic TIs* (meaning temporal determinism as discussed in SectionB.6.1.1) specify the interval between two significant events, but allow for a specified deviation. Deterministic TIs are necessary for CPS where repeatable and precise timing relative to the system timescale is required. *Accurate TIs* are deterministic TIs where the system timescale is TAI or UTC. Accurate TIs are useful for coordinating actions in CPS of large spatial extent, where accessing a traceable timescale is often more convenient than propagating a system-specific one. Accurate TIs are sometimes required due to legal or regulatory requirements.

B.6.2.1 Bounded TI

A bounded TI is always less than some stated value Δ_{MAX} (and sometimes always greater than some stated value Δ_{MIN}), i.e., $\Delta_{MIN} < TI < \Delta_{MAX}$. To be useful $\Delta_{MAX} < \Delta_{REQ}$, where Δ_{REQ} is an application-specific requirement on the bound.

Bounded TIs are the basis for operation in deadline-oriented CPS. For example, in an airplane the TI between the pilot's signal that the landing gear should be lowered and the gear being in place and locked must have a predictable bound but need not be deterministic. Failure occurs if the bound is exceeded, but there are no issues if the operation completes earlier.

Similarly, in a power plant the TI between a loss of load and shutting off the energy input to the generator turbine must have a predictable bound to prevent damage to turbines or other equipment that must dissipate the energy. In all such cases Δ_{MAX} must be small enough to meet

the application requirements. The verification of such bounds is a major task in designing and certifying CPS in many industrial and safety-critical applications.

B.6.2.2 Deterministic TI

In contrast to a bounded TI, a deterministic TI is always within some stated error ε of the application specification Δ_{REQ} on the TI, i.e., $|TI- \Delta_{REQ}| \leq \varepsilon$. In most CPS the attributes Δ_{REQ} and ε are specified in terms of a system-defined timescale rather than on international standards.

For example, smart highway designs require that cars be able to determine the distance to the car in front. Acoustic or electromagnetic ranging can be used to determine the TI between the transmitted signal and the signal returned from the other car. For acoustic-based ranging, and assuming the allowed error is one foot, a reasonable value for ε is one millisecond. That is, the difference between the actual and the measured time interval is the error of one foot divided by the speed of sound. If electromagnetic ranging is used, a reasonable value for ε is one nanosecond. Here ε is the required precision of the measurement (i.e., the CPS must be able to measure the ranging time with a resolution of ε). However, the accuracy requirement is much less severe (i.e., the second defined by the system timescale can differ from the SI second). In this case 0.1%, (e.g., allowing an error of one foot in 1000 feet), is probably more than adequate and would easily be met by a timescale governed by a quartz crystal oscillator with no need for calibration against international standards.

Engine control units are another example where the TIs must be deterministic rather than simply bounded. The intervals between fuel injections must have a precise timing relationship to the sensed position of the shaft. Again the timescale is local, since consistency within the engine is required, but it is not required for function that timing be based on the SI second.

B.6.2.3 Accurate TI

An accurate TI is a deterministic TI with the added requirement that the timescale be traceable to international standards. These are discussed in section B.6.1.1. Accurate TIs based on a timescale traceable to international standards are often needed to meet regulatory or legal requirements. For example, it is quite common in the medical industry for CPS specifications, including time, to be certified based on metrics defined by international standards.

However, the use of accurate, as opposed to merely deterministic, TIs often provides a simpler and more robust solution for a CPS. This is particularly true where the CPS is sufficiently large spatially that it is difficult to establish a deterministic timescale. For example, in North America, power systems often need to be coordinated over distances of thousands of miles. Synchrophasor technology is likely to be a critical part of the smart grid and will need to function over these distances. Synchrophasor technology requires the determination of the phase angles between the voltage waveforms at various parts of the grid.

The only realistic way this can be done on a continental scale is to make local measurement of phase with respect to a 60 (or 50) Hz cosine waveform synchronous with TAI. In principle, one could establish a consistent continental timescale by distributing time, frequency, and phase from a central location, but the effort would far exceed that of simply using GPS or GNSS. Power systems and telecommunication systems are similar in that both are continental-scale and both are implemented by independent companies rather than by a monolithic organization. So, for example, in North America prior to the breakup of the Bell System, a continental frequency standard was established by Bell based on distribution from a central location. Consistent frequency-not necessarily based on the SI second-was all that was required. Since the breakup, the only practical way to achieve the continental agreement on frequency is for each of the operating companies to implement their frequency distribution based on the SI second, again typically relying on GNSS. More recent protocols require time as well as frequency agreement, which has led the ITU-T to publish standards on the use of protocols such as IEEE 1588 in combination with GNSS for this purpose.

B.6.2.4 CPS nodes

A CPS node typically samples the physical world via one or more sensors, performs some computation based on the sensed values-often along with data obtained from other CPS nodes, possibly including the time of sensing-and delivers the computed results either to another CPS node or as an instruction to an actuator. In the case of a bounded TI, there need not be any explicit reference to the time of a timescale; while in the case of an accurate TI, the time is not only explicit but traceable to international standards.

To dispel any doubt about the central role that time awareness plays in CPS, look at the measures currently used in industry to achieve such awareness: time-triggered architectures [180], TDMA network protocols, and architectures such as PROFINET [181], IRIG-B [182], GNSS [183], IEEE 1588 [184] [185] [186] [187], FPGAs for critical local timing control, and finally analysis and reasoning techniques to determine code execution bounds, i.e., worst-case execution time (WCET) [188] [189] [190] [191], and the correctness of programs in meeting timing requirements [192] [193]. Conspicuously absent is timing-correctness by design, a term discussed later in this section.

Next consider how the architecture of typical CPS devices supports, or fails to support, timing. Figure 36 is a block diagram of a typical networked node of a CPS. Note that a CPS need not be networked, but may consist of one or more autonomous nodes. At the other end of the spectrum, very large scale CPS may form SoS, which introduce further challenges. Furthermore, many CPS nodes have multiple network interfaces to permit daisy-chained or more complex topologies.

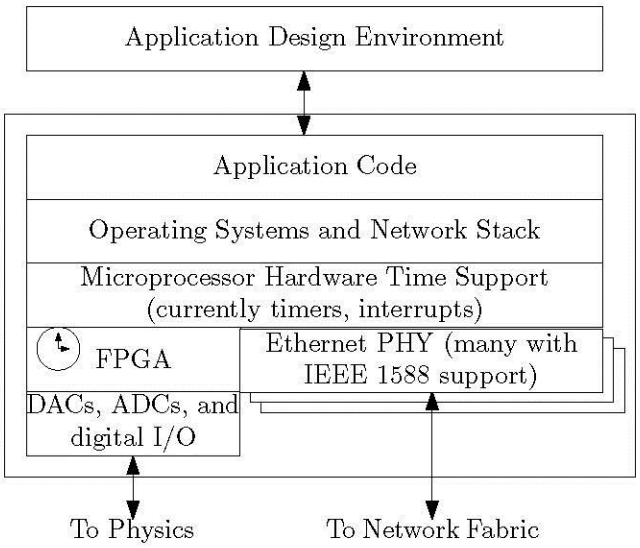


Figure 36: Architecture of a CPS Node and Environment

Consider the "P" or "physical" part of a CPS node, which includes physical things such as biological, electrical, thermodynamic, and chemical processes. For the most part, CPS physics models for natural and many man-made target devices include time explicitly (e.g., Maxwell's and Newton's equations, the diffusions equation). However, there are definitely targets of interest where time is not explicit in our physics models (e.g., radioactivity, Ethernet network traffic). Here the models are more likely to be state or statistical models. In these types of models, precise and accurate time is relevant to produce good models and particularly in establishing data provenance.

Considering the CPS microprocessor of Figure 36, timers and interrupts are the principal explicit means for supporting time constraints in modern microprocessors. With very few exceptions, it is not possible to specify or control the actual execution time of a code segment or the time to react to an interrupt. Furthermore, these times are often not even repeatable given the same inputs and code due to process scheduling, memory caches, pipelining, speculative execution, and similar features that have been introduced to increase the performance of modern microprocessors. In effect, modern general-purpose microprocessor operation is no longer time-aware; execution time is at best construed as a performance metric rather than as a correctness criterion. The result is that operating systems and commonly used programming languages also lack time awareness. It is clear that modern microprocessors cannot by themselves support deterministic or accurate TI requirements [194].

Under some restrictions, particularly on processors with no operating system or operating systems with non-preemptive scheduling, it is possible, albeit difficult, to analyze code execution timing and predict safe upper bounds [190]. Many safety-critical systems are based on these timing analysis techniques. For example, the aviation/aerospace industry uses these

techniques, but only uses qualified and certified processors and in applications that are deadline based, or use timing support hardware that can add determinism.

Time-triggered architectures illustrate how the separation of timing at the boundary between the cyber and physical parts of a CPS allow deterministic or, if needed, accurate timing at this interface, while requiring only bounded TIs on the computation phase [180]. This is a general principle not fully explored in today's design practices, CPS architectures, and applications.

Next consider the network interface. With the exception of TDMA protocols, network latency between two microprocessors is as unpredictable as code execution within the microprocessors. A lower bound can be set on latency, but that is the extent of network time awareness.

Where explicit and accurate time constraints are required within a CPS node, timing constraints are typically implemented in FPGAs, ASICs, or custom hardware logic where time is explicit, as opposed to depending on microprocessor code execution timing. If the CPS is distributed, it is possible to order events by means of messages passing over the network, but the enforcement of accurate timing requirements requires system-wide explicit time (i.e., a clock synchronized to its peers). In some cases, frequency and (relative) phase will suffice (e.g., ensuring that all converters between analog and digital (and vice versa) in a system use a common sampling rate, and/or a common sampling phase/time). In safety-critical systems, system-wide time is used to establish time-triggered architectures where applicable sampling, code execution, actuation, and network traffic are all based on schedules, generally periodic, and enforced by special hardware such as application-specific integrated circuits (ASIC) or field programmable gate array (FPGA) logic based on the node's synchronized clock.

Synchronized clocks are readily, but not universally, implemented in a CPS node. NTP can be made available at the application level, but this is of little help for accurate timing at the interface to "physics." As shown in Figure 36, newer physical layer network interface chips (e.g., Ethernet PHYs), typically contain hardware support for implementing synchronized clocks using protocols such as IEEE 1588, which enables the establishment of system-wide time to levels of accuracy and stability appropriate to the majority of CPS applications [195]. GNSS (e.g., GPS) technology is often used to provide a source of time for synchronizing clocks in a distributed CPS. However, to be truly useful, the time from the clock needs to be a key and explicit feature of timing support in microprocessors. This is not the case at present. At a minimum, standardized interfaces for time-sensitive operations should be inherent in the microprocessor architecture itself.

If time from synchronized clocks was inherent in microprocessor timing support, it would be possible to conceive of operating systems and languages that could enforce designers' timing requirements to a high degree of accuracy and determinism. It should be noted that if time were made explicit throughout the CPS along the lines outlined, the way designers conceive applications would change. The best example is the Google Spanner project [196], a worldwide

database that replaces the usual message passing logic for commits with logic based on reasoning about timestamps associated with transactions. The timestamps are generated by a worldwide time scale implemented by synchronized clocks. While not a CPS, Spanner does illustrate the change in design philosophy possible given the presence of system-wide time. The use of an adequately granular and accurate timescale allowed Google to revolutionize the management of database synchronization. This follows an observation of Barbara Liskov who some years earlier noted that NTP spurred interest in using time to improve mainstream computer science algorithms and protocols [197]

"Time correctness by design" includes the concept of designers including accurate timing in designs, independent of hardware [197] [199]. Designers need to be able to specify timing in a CPS as an abstraction, much as most modern systems are designed as abstractions, without reference to specific hardware. This is necessary to allow a design to persist through upgrades in the hardware and software. There is much work to be done to realize time correctness by design in full. In its ideal realization, a designer could include timing as an abstraction in a GUI design system. Upon choosing the target hardware, the system determines if that hardware can support the timing, and if so, generates the code and implementations to support the design.

B.6.2.5 Logical versus physical time

"Time correctness by design" involves a mapping of logical time to physical time. In engineering, timing is a physical signal. Before the physical implementation of a CPS, modeling may be employed to ensure correctness of design. Models use the concept of logical time. The use of time in models is not to indicate when things happen, but in what order they happen (causal order relation). The logical clock is a software counting register monotonically increasing its value. It is not related to a physical clock. Each process employs its logical clock to apply a timestamp to events. The ordering relation is obtained through this logical timestamping. While this decoupling of logical time from physical time may be useful in modeling, certainly at run-time, what may have been logical time must be mapped to a physical time, in the sense that the system is run on hardware that can enforce the correct timing relationships.

Certain newer distributed programming modeling tools like PTIDES [200][209] are better suited to designing CPS because they include the potential to map the logical time model into appropriate hardware. They assume the concepts of synchronized time and provide timing semantics in the programming language used to create the model. In these modeling tools, the interaction of the model with physical inputs (sensors) and outputs (actuators) uses delays that explicitly indicate the latency desired between reading a physical input and writing to a physical output (based on application requirement) in an implementation agnostic way. When mapping models created by such tools to physical implementations of CPS, the "logically synchronized" clock is mapped to the physical synchronized clock of the CPS, thereby enabling the latency specified in the model to be easily verified for a specific physical implementation. Used in

conjunction with the Y-chart [201], multiple architectural designs can be explored by mapping the model to a plurality of physical CPS implementations.

B.6.2.6 Recommendations

Finally, we present some recommendations for the design of future CPS:

- Incorporate time awareness at the lower levels (e.g., network and hardware) of the systems.
- As they become available, use microprocessors and other commercial off-the-shelf hardware that provide explicit support for time.
- Use networks with on-path support for clock synchronization. There are numerous examples of bridges and routers for Ethernet that incorporate such support.
- Explore ways in which the use of time, particularly in distributed systems, can be used to improve application designs.

From an architectural viewpoint, CPS nodes rarely exist in isolation and will typically form part of large-scale, geographically distributed systems. The concept of SoS illustrates the potential scalability of CPS. In such cases cloud computing will play increasingly important roles in CPS. The networks that support such systems will also see adoption of SDN and NFV technologies. This raises a range of timing-related challenges:

- Cloud – The role of the cloud in CPS will dictate the degree of time awareness that is necessary. At a minimum, data analytics will require synchronization, and a mapping from local to traceable time scale. Any comparison of data from remote locations will require consistent timing. For example, to achieve efforts such as root-cause analysis of events, the measurements of sensors must be time-stamped with an appropriate accuracy, referenced to a common time-scale. If the cloud plays a more time-sensitive role, then requirements similar to those discussed above concerning execution time must be met. Such challenges are made more difficult by virtualization, which is a foundation block of cloud computing.
- Network – The impact of SDN on timing performance needs to be carefully considered as does the role of NFV. While both technologies may reduce complexity and cost, and increase flexibility, their abstracted architectures may degrade timing performance.

Finally, CPS exist to fulfill business needs, and as shown in the CPS Framework, the timing requirements at this ‘aspect’ need to be met. Timing relevant to business requirements may be much coarser than for the operation of systems (e.g. seconds, months, and years versus milliseconds and microseconds), but business timing must be considered in .

B.6.3 Managing Timing and Latency in CPS

This section addresses the use of time to provide bounded latency in a CPS. The aim is to provide reference architectures/frameworks that enable the building of time-aware CPS to solve control and measurement applications.

Given the diversity in CPS applications and scale, it is not surprising that temporal considerations vary considerably over the range. For example, in small closed systems such as a packaging machine, the primary temporal concern is that all components respect a self-consistent timing design. In such systems, networking temporal considerations (e.g., design of a TDMA scheme) are part of the design itself. However, in large scale and, more critically, in environments characterized as SoS, timing issues are more difficult, as outlined above. For example, "smart highways" will involve many different systems, some in the vehicle, some in the infrastructure, some in a traffic management center, etc. Each will have its own temporal requirements that must be met while sharing network bandwidth and in some cases computation bandwidth on servers. Today the technology for managing the timing in such systems is still a work in progress. The remainder of this subsection discusses both the general issues as well as some of the current thinking on these issues. Some of these can be applied to smaller systems. There is no doubt that the work on larger systems will result in improvements (e.g., in time-sensitive network technology) that will make small system temporal design much easier and more robust.

CPS are used in both control and measurement applications. The requirement of bounded latency is obvious in control systems where latency from when a physical input is read to when a physical output is written has to be proven by timing and schedulability analysis. In large-scale control systems, this requirement becomes even more challenging because the input, computation, and output may be occurring on different, spatially-distributed nodes. The challenges of predictability in software are added to by the non-determinism provided by layers of software managing data transfer on the network connecting these nodes. As described above, the impact on timing of cloud computing and networking concepts such as SDN and NFV need to be carefully considered.

In CPS-based measurement systems, the deterministic relationship between acquired data (e.g., simultaneity) is of paramount importance. However, what is typically overlooked is the efficiency and complexity of transferring the acquired data from thousands of nodes to one or more aggregating units, where analytics or logging is being performed. Misaligned data can result in faulty conclusions. In many CPS-based applications, the data measurements are used for asset or structural-health monitoring, and in many cases a timely response based on real-time analytics is required. Time, when applied to data transfer, can enable bandwidth reservation in networks used in these measurement applications, thereby enabling faster analytics, a smaller memory footprint, and increased efficiency in data-reduction techniques

(for logging). Moreover, bounded latency is extremely useful in distributing triggers to multiple nodes inside a CPS.

Similar to CPUs, computer networking has traditionally been optimized for "best effort delivery;" that has worked extremely well in the past and will continue to do so in the future for many uses. However, it is not good enough when the same networking technology is used for time-sensitive applications that are served by CPS. Time-based CPS can be built using standard Ethernet technologies to enable seamless integration with the Internet. Time awareness in standard Ethernet is paving the way to enable time-sensitive (bounded latency) traffic to coexist on the same network as traditional best-effort (no latency guarantees) traffic. There are several standards being developed in the IEEE and other standards development organizations for this purpose.

A time-aware CPS should guarantee bounds on latency of data delivery and guarantees on synchronization accuracy as it applies to timing correlation of physical I/O. To build such large-scale systems with these guarantees, the following two concepts of CPS time domain and CPS network manager are defined.

CPS Time Domain: A CPS time domain is a logical group of CPS nodes and bridges that form a network with its own timing master. It may be a time-space region as discussed in section B.6.1.4. The master may synchronize to a globally traceable time source (e.g., GPS). Each CPS time domain has its own primary (or locally synchronized) timescale. This timescale provides a strong monotonically increasing clock to applications for performing input/output functions and time-based scheduling. The timing master of a CPS time domain should not produce a discontinuity of time once time-sensitive data transfer within the domain has commenced, even if the master sporadically loses connectivity to its global source (e.g., GPS).

If a global traceable time is required inside a CPS node, then the node can implement a second timescale called the *Global Traceable Timescale*. This timescale can be managed independently of the CPS's primary timescale. To correlate the CPS's primary timescale to the Global Traceable Timescale, the offset of the primary timescale from the Global Traceable Timescale can be maintained at all times by the CPS node. The Global Traceable Timescale can be used to correlate CPS timescales from multiple CPS time domains.

Many CPS will be small enough that they do not need an external timescale, and the primary timescale will suffice. However, with many things becoming networked, some level of traceable timing may be available, though perhaps not at the needed precision. For example, a power plant controller may use its local time scale for operations, but be referenced to UTC from GNSS for time-stamping events.

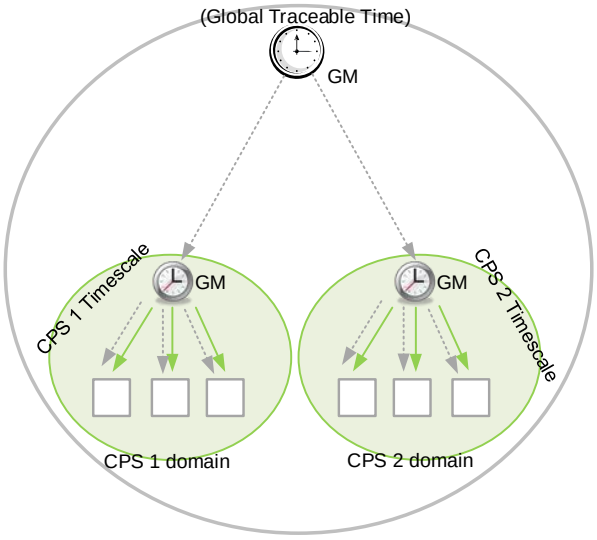


Figure 37: Domains and Multiple Timescales in Time-Aware CPSs[26]

CPS Network Manager (CNM): A CNM is a workstation or CPS node connected to a CPS time domain that manages and monitors the state and configuration of all CPS nodes in one or more CPS time domains, or in a more scalable SoS. The CNM interfaces with a schedule generator and path computation engine to generate the schedule for the CPS. This may be done by interfacing with a centralized network controller. For performance, reliability, and/or scalability reasons, functions of a CNM may be distributed among multiple devices. For example, a large SoS may require a distributed CNM, much like in SDN. There are currently efforts underway to standardize the role of the CNM with the name Centralized User Configuration (CUC). References to these efforts are:

a) Time Sensitive Networking, "802.1Qcc - Stream Reservation Protocol (SRP) Enhancements and Performance Improvements" (this is still in draft stage currently)

b) AVnu Industrial , "Interoperability and Conformance Standards about Centralized User Configuration and Centralized Network Configuration", www.avnu.org/industrial. (This work is in incubation and very early stages of discussion).

All fieldbuses that are used in Industrial Automation support a centralized configuration entity. Examples include RSLogix from Rockwell Automation and STEP 7 from Siemens.

[26] **Source: Sundeep Chandhoke, National Instruments**

The functions of a CNM vary depending on the size of the system. These functions include:

- Control and manage the state of all CPS nodes in a CPS time domain
- Coordinate with a centralized network controller to configure bridges in a CPS domain
- Configure transmission schedules on CPS nodes
- Monitor the health of the CPS time domain (for handling errors, changing schedules, bringing new CPS nodes online, etc.)
- Configure application and I/O timing on each CPS node
- Configure any static timing requirements for time-based synchronization

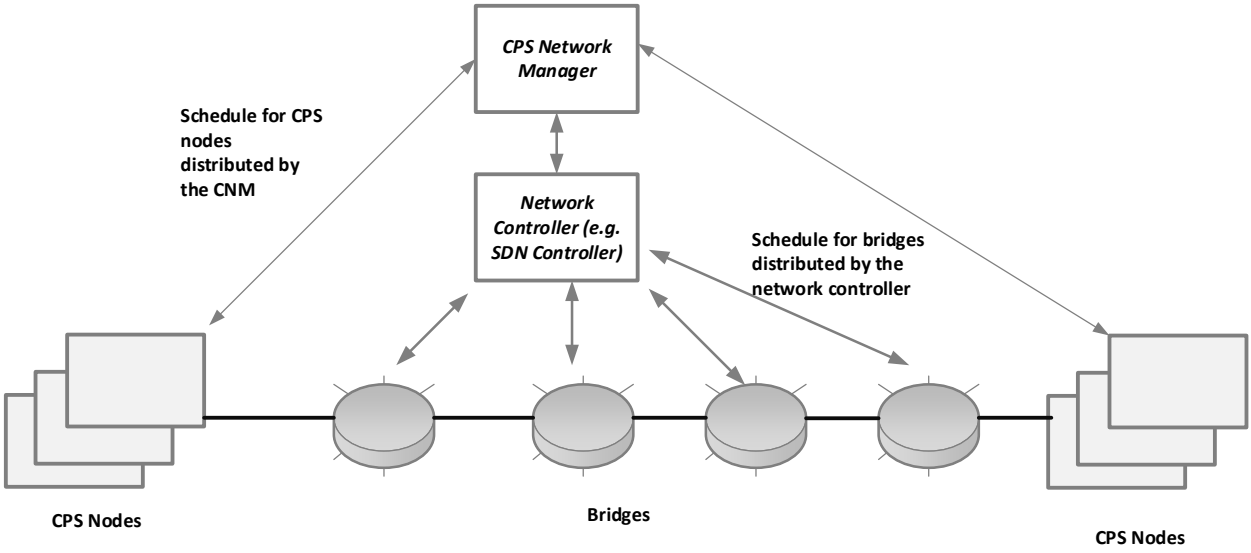


Figure 38: CPS Network Manager Configuring a CPS [27]

Either the CNM or the centralized network controller has to gather performance metrics and determine the topology of CPS nodes in a CPS time domain in order to create a schedule. The relevant performance metrics include bridge delays, propagation delays, and forwarding/transmission delays. There are multiple ways to detect topology. For example, one approach to SDN defines a "Packet-In" "Packet-Out" protocol that uses Openflow [202] with Link Layer Discovery Protocol (LLDP) [203]. Some other protocols like PROFINET [204] use Simple Network Management Protocol (SNMP) [205] along with LLDP. The CNM computes the topology for the CPS time domain using these mechanisms, and determines the bandwidth

[27] Source: Sundeep Chandhoke, National Instruments

requirements for each time-sensitive stream based on application requirements. The bandwidth can be specified by the period and the size of the frame. Optionally the application can also specify a range <min, max> for the offset from start of a period. This information is provided to the Centralized Network Controller. The Centralized Network Controller computes the path for the streams and gathers performance metrics for the stream (latency through the path and through the bridges). This information is then used to compute the schedule for the transmission time of each time-sensitive stream and the bridge shaper/gate events to ensure that each time-sensitive stream has guaranteed latency through each bridge. Additionally, queues in bridges are reserved for each stream to guarantee bandwidth for zero congestion loss. It should be noted that schedule computation is the subject of continuing research as the problem becomes intractable for large systems.

It should also be noted that there is considerable activity in the IEEE 802.1 and other standards communities aimed at providing additional tools for controlling network temporal properties. See the Timing Framework Annex Section 1.2 [178] for additional details.

An illustration of a possible device model for a time-aware CPS node is shown in Figure 39.

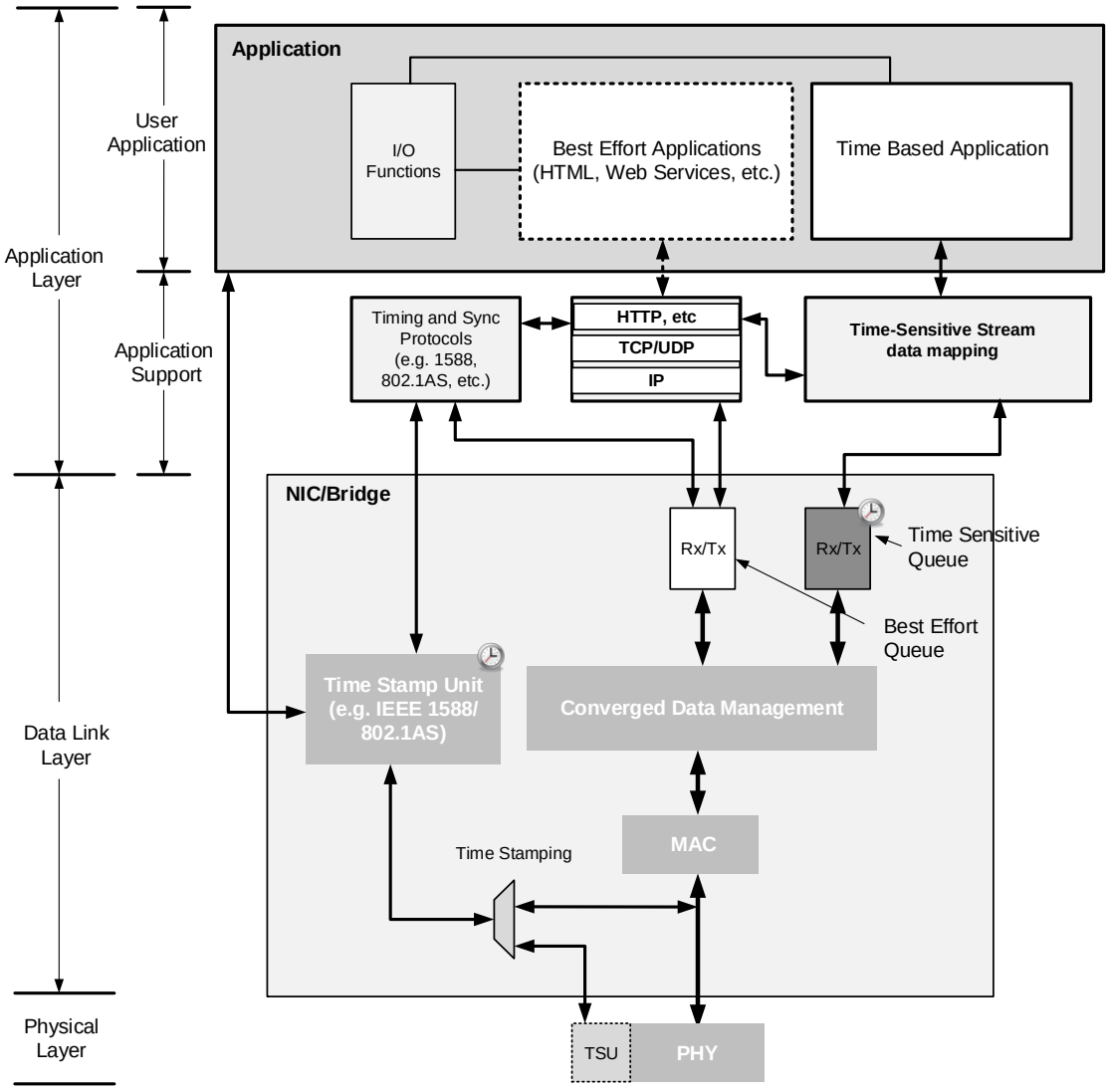


Figure 39: Time-Aware CPS Device Model[28]

The physical layer receives data units from the data link layer, encodes the bits into signals, and transmits the resulting physical signals to the transmission medium connected to the CPS node. If the physical layer supports a timestamp unit (TSU) then its management interface should be connected to the data link layer so that a timestamp can be retrieved if required by the timing and synchronization protocol (e.g., IEEE Std. 1588™).

[28] Source: Sundeep Chandhoke, National Instruments

The data link layer provides time-sensitive data communication among devices in a CPS time domain. The data link layer implements a set of dedicated buffer pairs (Tx and Rx queues) for time-sensitive data. At a minimum, two pairs of buffers are required so that time-sensitive data can be managed independently from best-effort data. The time-sensitive transmit buffer is connected to a scheduled (time-triggered) transmit unit. This unit uses a schedule provided by the CNM, reads data from the application, copies it into the time-sensitive transmit frame, and transmits the frame on to the CPS time domain.

The application layer consists of these parts:

- Application-support protocols: These are the protocols that support the conveyance of time sensitive data at the user's application level.
- Time-sensitive data mapping: Protocol to manage the mapping of application data to time-sensitive data exchange frames between devices. An example is CANopen [206], which is used as a data-mapping protocol by multiple industrial protocols.
- Best-effort protocols: Used for standard Internet access, non-time-sensitive streams.
- Timing and sync protocols: These include protocols that propagate synchronized time from the network to the application (including I/O functions). Some examples of such protocols are IEEE 1588 and IEEE 802.1AS.
- User application: User-defined applications accessing time-sensitive and best-effort data, and time-sensitive I/O interfaces to allow decoupling of logical and physical time with enforcement only at the boundary to physics. An example of a realization of this capability of mapping the boundary, logical time to physics, is inherent in the design of the Texas Instruments DP8360 Ethernet PHY.

Currently time in CPUs is implemented via timestamp counters (TSC) that increment time using the local clock driving the CPU. This clock is not locked to time from the network. The TSC can be disciplined via software to slave it to time as received from the network. However, this leads to significant loss of precision and accuracy. For CPS nodes that synchronize to a single external clock source, it may be desirable to have the TSC driven directly by the time from the network. This may be implemented by linking the registers of the TSC with the timekeeper in the network interface or by providing a common time-base that can be atomically captured by the network interface before propagating the network time to the CPU or any peripheral device. More generally, CPS applications may choose to maintain offset/PPM state for each derived clock and translate on-the-fly as needed without physically disciplining the TSC. This is especially useful in cases where the applications care about multiple time sources.

Modern CPU architectures and their memory-mapped interconnects are becoming time-aware. The PCI-SIG group introduced the concept of time into PCI-Express in March 2013 (called Precision Time Measurement [206]). PCIe-PTM allows a common reference time for the entire CPU peripheral subsystem, making it possible to more precisely correlate time in the peripherals with time being propagated in the network. This is key for CPS since the peripherals

implement the interfaces to the physics (digital and analog inputs and outputs). More recently Intel has also introduced better time integration for CPUs so that logic executing in the CPU can be tightly coordinated with synchronized time. Intel added a new instruction to the CPUs that will allow higher precision in measuring the offset between the timestamp counter (TSC) clock (which is now no longer spread spectrum) and the source of synchronized time (1588, GPS etc.). The instruction is called Invariant Timekeeping (17.13.4 in Intel's software development manual [207]).

Languages used for modeling and programming of time-aware CPS need time as a fundamental programming semantic. Time in the language is required when interfacing to physical I/O and the network. Functions that take future time events to read physical inputs and write physical outputs can enable coordination of physical I/O with scheduled data on the network. Additionally, time-triggered loops can enable coordination of logic execution with schedule of transmission of data. PTIDES [209] and LabVIEW [210] are two examples of system design tools that implement these time-based programming semantics.

CPS can employ operating systems with a wide range of complexities, from a simple application-level infinite loop (e.g., the Arduino platform) to a virtual machine hypervisor running several instances of virtualized systems on a multi-blade, multi-core hardware platform. The issues that arise throughout these systems with respect to time awareness are how to get time to the application with a bounded latency and with accuracy, and how to schedule tasks with time accuracy and bounded latency. Greater detail on these issues in CPS can be found in the Timing Framework Annex Section 1.2 [178].

At the application layer, the introduction of time awareness will have a profound impact on the conception, design, execution, and robustness of CPS applications. This is a very active area of research, but there are hints of things to come. For example, the concept of decoupling of logical and physical time with enforcement only at the boundary to physics mentioned above has yet to be fully exploited. In some cases, tradeoffs made between message passing, which consumes network bandwidth, and reasoning about timestamps can be exploited by applications.

Building CPS using the above-mentioned techniques will make it easier to analyze systems, which is a key requirement of safety-critical systems. CPS with scheduled converged networks built with FPGAs and time-aware CPUs will provide static guarantees and always satisfy timing requirements. Architecture-specific analysis tools can derive these guarantees in the form of upper and sometimes also lower bounds on all execution times, since time is foundational in all elements of the CPS.

B.6.4 Secure and Resilient Time

Requirements for secure and resilient time exist at all layers of the network from the physical to the application layer. While time is physical, its abstraction into networks and complex

information systems transform its security into cyber and physical concerns. Therefore, time affects both cyber and physical security architectures. As described in the Timing Framework Annex Section 1.3.2 [178], timing may be vulnerable to unintentional (interference, space weather impacts, network anomalies, etc.) or intentional threats such as jamming and spoofing (counterfeiting via RF signal injection or cyber attack).

The ability to meet timing performance requirements in CPS is also susceptible to vulnerabilities either related to time protocols in use or introduced by cybersecurity measures.

For example, the use of network firewalls may isolate time in a CPS for protection from the external network at large. Network firewalls enable centralized control of perimeter data security added as a user-provided assurance. However, with time isolated, clock drift may occur between the internal and external networks, resulting in performance degradation and in some cases failure at one or more levels.

Similarly, when time reference sources from GNSS or from networks are compromised causing synchronization errors, attempting to normalize or restore time services can run a high risk of timing discontinuities and other alignment issues.

Due to the increasingly wide range of timing-dependent applications in critical infrastructure domains, secure time must be designed into the system to *detect* timing anomalies before performance degradation of the system occurs and to seamlessly ensure sufficient time accuracy and precision can be maintained in the overall system during a compromise. This section describes the elements that constitute secure and resilient time and how time can be compromised, as well as methods for ensuring access to secure and resilient time.

B.6.4.1 Elements of secure timing

There are several prevalent methods to distribute time over a CPS. For example, a CPS in a closed system might need a locally synchronized time that can be achieved via a local implementation of PTP. Other CPS might need to be synchronized globally to UTC and depend on GNSS or a GNSS-derived network timing source. Each of these timing sources enters the CPS from a different network layer and hardware chain.

Wherever possible and viable, timing distribution systems should provide some level of data and channel assurance. This source-provided timing assurance provides a baseline of security that individual CPS may or may not choose to enhance on an application-specific basis. If the time transfer medium does not include source or data assurance, a user may augment the security with user-added security, such that the CPS would be able to verify the integrity and authenticity of the time. These topics are summarized in Table 11.

In many applications, such as critical infrastructure, the CPS must also tolerate both permanent and transient faults. Fault tolerant distributed systems must be able to synchronize the non-faulty processes in the presence of incorrect or conflicting temporal information from correct

and erroneous sources as well as missing information from failed components. Collectively, such arbitrary behaviors in temporal information are known as Byzantine faults. In order to achieve Byzantine fault-tolerant timing, one must include not only the information security elements to ensure integrity and authenticity, but also ensure predictable failure as well as diverse and redundant traceable timing sources and paths.

Achieving predictable failure in the event of intentional and unintentional system timing errors requires prompt, precise, and accurate *fault detection*. The properties of fault detection include (a) timeliness and (b) percentage of true or false positives or negatives. Furthermore, once a fault is detected, the system can enable *fault containment*. A *temporal firewall* [211] would contain the temporal faults of a component or subsystem to ensure other components in the system are not compromised. The interface of the temporal firewall would communicate the temporal properties of *time accuracy*, the *(global) time base*, and the *time-bounded validity* (timeliness) of the information being propagated from within the firewall. Ideally, a predictable failure would also include *self-stabilization* where in bounded time, the system in any state, fault or recovery, should converge to a correct state without external intervention [212]. The self-stabilization process would need to ensure the *slew rate* or *step size* are bounded to prevent temporal alignment errors in the system. The challenges for ensuring fault tolerance in system clock synchronization include managing complexity, efficient convergence time, and the ability to prove correctness of fault tolerance [213].

Ensuring fault tolerance in reaching synchronization among erroneous and failed nodes requires the system to have diverse and multiple sources and nodes. The topology of the system must also ensure each node has multiple paths to each source. The literature includes many theories and methods to achieving temporal fault tolerance [211], [214], [215].

The fault tolerant system should specify (a) the number of faulty nodes/links tolerated (b) the properties of fault detection (c) the topology of the system including number of end nodes and bridges (d) duration of the changeover (e) failure modes supported and (f) timing uncertainty bounds when all other assumptions are met.

A CPS with fully secured time must possess the necessary assurance and resilience attributes described in Table 11.

Table 11: Elements of Secure Timing

Attribute	Description
Source channel assurance	Opportunities to verify that timing information is delivered via an undistorted channel whose expected behavior is well characterized to ensure any deviations can be quickly detected. Distortion of the time-transfer channel may be driven by natural events (e.g., solar weather), unintentional actions (e.g., physically bumping an antenna), or intentional manipulation (e.g., introducing a time delay via spoofing). The data carried by a time-transfer channel may assist in verifying the channel itself. Enablers of channel verification may include unpredictable bits of a digital signature, or a symmetrically-encrypted channel.

Source data assurance	Verification mechanisms to prove timing data are not forged. These may include digital signatures or symmetrically encrypted packets.
User-provided assurance	User-implemented security to verify unassured timing information. This may include anti-spoof GNSS receiver techniques or additional layers of network security.
Predictable failure	Known CPS failure modes that account for timing anomalies, such as denial. The ability to achieve predictability includes consistent, accurate, and precise *fault detection*, *fault containment*, and *self-stabilization*.
Availability and Diversity	Availability and diversity work together. Availability: Some timing signal and its associated data as required by a CPS can always be received and used by the CPS. Diversity: Multiple sources and paths of secure time are available to a CPS; if a commonly used source of time such as GNSS is denied to a CPS, other equally accurate and secure sources of time are available. Where possible, sources are verified against each other, and in the event of a denial or spoofing attack on one source or other timing anomaly, the diverse signals may permit defeat of the attack, or a mechanism to switch to a redundant source is available.

When a timing source does not make assured time available, the CPS should implement timing assurance methods appropriate for the level of protection they need. Table 12 provides a survey of timing distribution methods and whether or not they provide any level of source channel or data assurance. Different levels of timing assurance are appropriate for different applications. For example, a car's timing network may require more security than a networked household appliance. Table 12 indicates whether *any* elements of assured time are present in these distribution methods or whether they remain open to a trivial attack. Current timing distribution systems are generally lacking in source-provided assurance and rely on users to implement their own security measures; however, opportunities may exist to enhance their security.

Table 12: Survey of Time Distribution Methods

	Order of Timing	**Source Channel Assurance Provided Today**	**Source Data Assurance Provided Today**	**Source Channel Assurance Possible via Enhancement**	**Source Data Assurance Possible via Enhancement**
GPS L1 C/A	nanoseconds	No	No	No	No
GPS L2C/L5	nanoseconds	No	No	Yes	Yes
Galileo	nanoseconds	No	No	Yes*	Yes*
PTP [211]	nanoseconds	No	No	Yes	Yes
NTP [217]	milliseconds	No	No	Yes	Yes
eLoran [218]	nanoseconds	No	No	Yes	Yes
WWVB [219]	microseconds	No	No	Yes	Yes

*Galileo is not yet a fully operational GNSS constellation, but has indicated strong support for source channel and data assurance via navigation message authentication.

To safely and reliably operate in today's threat environment, a CPS should implement as many elements of secure timing as possible. Ideally, every CPS in a safety-critical application should have multiple, independent, assured, and traceable sources of time with safe and predictable failure modes should time be denied or perceptibly manipulated. Where a mix of secure and unsecure timing sources are available, and traceability to a common time standard exists between them, the unsecure timing sources may be validated against the secure timing sources.

Secured time signals and measurements should be assured for a CPS that uses well-defined performance metrics including phase accuracy, frequency stability, holdover capability, mean detection time, traceability, and switchover time. Addressing the research needs for a fully-secured time in safety-critical CPS remains a high priority.

The Timing Framework Annex Section 1.3.8 [178] describes two possible use cases in the power system domain where secure time is necessary. The first use case describes how GNSS vulnerabilities can lead to synchrophasor measurement errors. To enable Phasor Measurement Units for real-time control, the power industry must ascertain the measurements are accurate and reliable. Erroneous measurements could appear as instabilities in the grid. Automatic protection schemes relying on the compromised measurements could trip generators. Tripping generators unnecessarily can cause blackouts and/or significant damage to power systems' equipment. The use case illustrates how elements of secure time implemented on top of GNSS timing led to a hypothetical detection of the GNSS compromise. Subsequently, a predictable failover to an equally precise redundant timing distribution system would ensure access to assured time.

Similarly, the second use case describes how digital substation automation can be compromised by network timing protocol attacks such as spoofing and DoS. Again, both attacks can lead to erroneous measurements of synchrophasors, leading to inability to accurately monitor the state of the grid, and potentially impacting control decisions. distribution through networks that implement the secure time elements--including source channel and data assurance, user-provided assurance, predictable failure, and diversity and redundancy--would minimize any compromise's impact on system timing performance.

Without assured time, critical infrastructure systems that people depend upon daily (power distribution, telecom, transportation, the Internet, etc.) are vulnerable to disruption. As Table 12 illustrates, time distribution methods available today require user or system enhancements to meet source channel and data assurance requirements. If there are conventional security measures built into the time distribution method, these measures have known vulnerabilities

that are readily compromised by an attacker. Additionally, most end-use timing equipment is vulnerable to the disruption caused by source channel and source data disruption.

B.6.4.2 Current security in distributed timing systems

Timing is generally distributed to CPS via GNSS constellations or a network timing protocol. This section surveys the security mechanisms and vulnerabilities inherent in these two distribution methods.

B.6.4.2.1 GNSS timing directly to devices/equipment

Civil GNSS signals are the primary worldwide timing distribution mechanism, and are inherently vulnerable to jamming and spoofing.

Jamming refers to the denial of the signals-in-space by illegally broadcasting energy in the radio navigation spectrum. Low-power (<1W) jammers are widely available to consumers and are marketed and used as "personal privacy devices." High-power jammers are generally used to intentionally deny GNSS receivers over a wide area. Though the effects of denial can be damaging, robust timing receivers should enter into pre-defined holdover, mitigation, or failure modes when jamming is detected.

GNSS spoofing is the RF injection of counterfeit or recorded GNSS signals into a receiver. Spoofing attacks may be data (e.g., replace the navigation data on the GNSS signal) or timing oriented (e.g., induce a delay). Jamming may be intentional or incidental. Generally spoofing is intentional, though it may be possible for incidental spoofing to occur (e.g., through legal GNSS repeaters). Unlike incidental jamming, many straightforward mitigations exist to incidental spoofing. Though spoofing is not yet as commoditized as jamming, publicly available research into spoofing techniques has been significantly increasing, and software-defined spoofers have been appearing in multiple independent research universities.

As the majority of critical infrastructures rely on GNSS as a reference source, GNSS jamming and spoofing are known critical infrastructure vulnerabilities (due to reliance on GNSS-provided timing), and awareness of their consequences has been increasing significantly. Current areas of research include source channel and data assurance, anomaly detection before clocks are significantly impacted, and redundant distribution sources.

There has been significant work done on receiver-side techniques to mitigate spoofing and jamming. Some GNSS providers (Galileo) have advanced toward securing the signal-in-space via *navigation message authentication (NMA*) – that is, digitally signing the data transmitted by the satellites). An NMA implementation scheme that could be implemented on the modernized civil GPS signals is being considered [220]. A signal-side security scheme such as NMA provides an affordable and backwards-compatible baseline of protection for civil GNSS receivers against spoofing, and would provide globally available time that is "source assured." Receivers could choose to ignore NMA, adopt it, or adopt it and implement additional measures of assurance.

Asymmetric cryptography schemes can also be added to other timing signals and protocols (e.g., possibly WVVB or PTP) for source channel and data assurance.

The development of other methods for national-level reference time distribution to backup and augment GNSS in the event of a failure has become another active area of research. The Timing Framework Annex Section 1.3.3 [178] describes some currently available or researched alternatives to distribution of time traceable to a national reference. WWVB and eLORAN[218][221] are two alternatives that have been able to achieve wide area synchronization. Research efforts in alternative methods include achieving a timing accuracy comparable to GNSS as well as ensuring secure time in the alternative methods. Communication sector timing distribution methods, such as time distribution protocols over dedicated optical networks or a combination of SyncE and PTP, can serve as an alternative source of national reference time. Another area of research is in Assisted Partial Timing Support (APTS) [222], which provides active monitoring and detection of synchronization deviations as well as automatic switchover to an alternative time distribution source in the event the GNSS is deemed unreliable.

B.6.4.2.2 Network timing

Network timing distribution leverages a packet-based protocol (e.g., PTP or NTP) to distribute timing information via a hierarchy of receivers. At the top of the hierarchy is a timing source that often derives a traceable national reference time from a satellite constellation (e.g., GNSS) or another time transfer source (e.g., eLORAN[218][221], WWVB[219], etc.). Network timing distribution has a different set of security considerations than GNSS-based timing. Network-based distribution methods are prone to common network vulnerabilities. The threats can compromise the integrity and availability of time in a CPS network. Securing network time distribution methods includes assurance for authenticity to a traceable time reference, integrity of the timestamps and other metadata exchanged in the synchronization packets, and availability through redundant and diverse paths. Another key requirement to secure time in networks is the ability to detect the intrusion or other forms of anomaly in the network before the threat has had an impact on the network time. When anomalies in the timing distribution network are detected, the CPS would have the means to fail predictably with minimal impact on the function of the system. Ideally, the system would have diverse and redundant paths for timing distribution where the system can switch over readily once an anomaly is detected while maintaining the necessary timing accuracy and precision in the CPS.

B.6.4.2.2.1 Attack vectors in time networks

Network timing distribution methods are susceptible to attacks characterized by an unauthorized third party, known as Man in the Middle (MitM) or interceptor, which can be manifested as different threat types. Table 13 outlines different principal threat vectors [223][224] and their impact on time networks. The impacts of the threats include limiting the

availability of time distribution in the network, distributing completely erroneous time, or distributing time with reduced accuracy. The threats can be passive (message interception) or active (message interruption, insertion, or modification). Passive attacks tend to be the prerequisite to other attacks. Therefore, detecting passive attacks is one method to preventing an attack from having impact on the timing accuracy of the CPS.

Both external and internal perpetrators must be considered in a network security threat analysis. While external attackers do not have access to the network's security credentials, internal attackers do. The Timing Framework Annex Section 1.3.4 [178] provides more in-depth definition of terms for describing time compromises in networks, and The Timing Framework Annex Section 1.3.5 [178] provides detailed external and internal threat analyses for network time distribution protocols.

Table 13: Principal Threat Vectors in an Unsecured Time Network

Threat Type	Threat Characteristic	Impact	Example
Packet Manipulation	Modification (MitM)	False time	In-flight manipulation of time protocol packets
Replay Attack	Insertion / Modification (MitM or injector)	False time	Insertion of previously recorded time protocol packets
Spoofing	Insertion (MitM or injector)	False time	Impersonation of legitimate master or clock
Rogue Master (or Byzantine Master) Attack	Insertion (MitM or injector)	False time	Rogue master manipulates the master clock election process using malicious control packets, i.e., manipulates the best master clock algorithm
Interception and Removal	Interruption (MitM)	Reduced accuracy, depending on precision of local clock	Time control packets are selectively filtered by attacker
Packet Delay Manipulation	Modification (MitM)	Reduced accuracy, depending on precision of local clock	Intermediate / transparent clock relays packets with non-deterministic delay
Flooding-Based General DoS or Time Protocol DoS	Insertion (MitM or injector)	• Impairment of entire (low-bandwidth) network • Limited or no availability of target (service)	• Rogue node floods 802.15.4 network with packets • Rogue node overwhelms single victim with time protocol packets

Interruption-Based General DoS or Time Protocol DoS[29]	Interruption (MitM or possibly injector)	• Impairment of entire network communication • Limited or no availability of target	• Rogue node jams network • Rogue node jams selectively certain time protocol packets
Master Time Source Attack	• Interruption (MitM or injector) • Insertion (MitM or injector)	• Reduced accuracy • False time	• GNSS jamming • GNSS spoofing
Cryptographic. Performance Attack	Insertion (MitM or injector)	Limited or no availability of target	Rogue node submits packets to master that trigger execution of computational expensive cryptographic algorithm (e.g., digital certificate validation)[30]

Current mitigation strategies for addressing network time distribution vulnerabilities include authentication of the synchronization source and integrity verification. NTP uses the AutoKey protocol to achieve end-to-end authentication, message integrity, and replay protection. NTP is an end-to-end synchronization protocol, whereas PTP is a hop-by-hop synchronization protocol using transparent/boundary clocks to achieve higher synchronization precision. The ability to secure a hop-by-hop protocol presents a unique security challenge. PTP has an experimental Annex K, which provides group source authentication, message integrity, and replay protection. The Timing Framework Annex Section 1.3.6 [178] describes some of the network timing distribution protocols' security extensions. With the increasing demand for security, existing security protocols such as MACsec and IPsec can be used to complement PTP. MACsec provides hop-by-hop integrity protection, whereas IPsec provides end-to-end integrity protection. The Timing Framework Annex Section 1.3.7 [178] details current countermeasures for achieving authentication and integrity.

As with GNSS, research continues with respect to detection of anomalies and the ability to maintain resilience of the clock synchronization network while maintaining the increasingly stringent precision and accuracy requirements. In large scale and dynamic networks, key management is a challenge in ensuring hop-by-hop timing protocol (e.g., PTP) security. Furthermore, there is a continuous need to improve countermeasures as new vulnerabilities

[29] This attack is blunter than the "Interception and Removal" attack above, as here all time-protocol related packets are omitted.

[30] The exchange and validation of a certificate as part of the authentication and authorization of a node can be the building block of such an attack.

arise. There is currently a demand on the network time distribution protocol standards efforts for guidance in achieving secure timing, while minimizing impact on time distribution performance. Current security extensions are susceptible to certain threats such as cryptographic spoofing and a variety of internal attacks. Standards efforts are currently underway to define optional security specifications for meeting source channel and source data assurance in NTP[225][226] and PTP[227].

B.6.4.3 Achieving secure time

Timing security in critical systems requires more than the availability of secured timing sources. Secure time requires the inclusion of timing security in the CPS system architecture from its design in such a way that when the system detects potential timing compromises, it can failover to a redundant timing source (either internal or external to the system). Existing technologies use redundancy and diversity of routes to time and frequency sources as well as holdover capabilities of high-stability oscillators. There continue to be research needs in the areas of timing compromise detection, alternative sources to traceable national standard reference time, timing network topologies to support diverse and redundant paths, and cybersecurity measures that minimize impact on timing performance. In addition, practical testing and validation of experimental results would ensure safety and predictability in failure modes.

Due to the lack of secured timing sources globally available today, a reasonable approach to securing time is to ensure systems can maintain timing within the tolerance of their application for the duration of a timing compromise. The future vision of secure time is to ensure timing compromises can be detected sufficiently early such that systems dependent on accurate and precise timing can seamlessly function under compromised conditions without any performance impact to the CPS.

B.7 Boundaries Aspect

Concerns related to demarcations of physical, functional, organizational, or other forms of interactions:

behavioral	Concerns related to interdependence among behavioral domains. Concerns related to the ability to successfully operate a CPS in multiple application areas.
networkability	Concerns related to the ease and reliability with which a CPS can be incorporated within a (new or existing) network of other systems.
responsibility	Concerns related to the ability to identify the entity or entities authorized to control the operation of a CPS.

CPS consist of systems that include devices. There are many kinds of devices. Each device falls under one or more organizations that has responsibility for its configuration, lifecycle

maintenance, and access rules to interact with it. Additionally, there is a network topology overlaying the organizational topology.

In some cross-CPS-domain use cases, access to data by client applications far removed from the actual administration of devices may be desirable.

The following are general classes of logical devices that can potentially interact:

- Sensor and actuator: Sensors and actuators provide the simplest functionality that allows the interaction between cyber and physical.
- Controller: Controllers combine data from sensors and produce control actions via actuators.
- Gateway: Gateways provide the ability to forward information exchange between local devices within a proprietary network and a remote network (often the Internet). Gateways are often, but not always, the boundary between private and public networks.
- Aggregators and concentrators: Aggregators and concentrators provide for data fusion and allow for managing the forwarding of information obtained from resource-constrained networks to more capable ones.
- Broker: Message brokers supporting publish and subscribe message routing and certificate assurance services are examples of infrastructure components that enhance the function and security of information exchange.
- Cloud-based analytics: “Big data” and other cloud-based services provide for the exploitation of large collections of data from many sources.

A more comprehensive treatment of this Aspect is anticipated in the future.

B.8 Composition Aspect

Concerns related to the ability to compute selected properties of a component assembly from the properties of its components. Compositionality requires components that are composable: they do not change their properties in an assembly. Timing composability is particularly difficult. Specific concerns include:

adaptability Concerns related to the ability of the CPS to achieve an intended purpose in the face of changing external conditions such as the need to upgrade or otherwise reconfigure a CPS to meet new conditions, needs, or objectives.

complexity Concerns related to our understanding of the behavior of CPS due to the richness and heterogeneity of interactions among its components, such as existence of legacy components and the variety of interfaces.

constructivity Concerns related to the ability to combine CPS modular components (hardware, software, and data) to satisfy user requirements.

discoverability Concerns related to the ease and reliability with which a CPS component can be observed and understood (for purposes of leveraging the component's functionality) by an entity (human, machines). Concerns related to the ease and reliability with which a CPS component's functions can be ascertained (for purposes of leveraging that functionality) by an entity (human, machines).

A more comprehensive treatment of this Aspect is anticipated in the future.

B.9 Lifecycle Aspect

Concerns about the lifecycle of CPS components:

deployability Concerns related to the ease and reliability with which a CPS can be brought into productive use.

disposability Concerns related to the impacts that may occur when the CPS is taken physically out of service.

engineerability Concerns related to the ease and reliability with which a CPS design concept can successfully be realized via a structured engineering process.

maintainability Concerns related to the ease and reliability with which the CPS can be kept in working order.

operability Concerns related to the operation of the CPS when deployed.

procureability Concerns related to the ease and reliability with which a CPS can be obtained.

producibility Concerns related to the ease and reliability with which a CPS design can be successfully manufactured.

A more comprehensive treatment of this Aspect is anticipated in the future.

Appendix C. Use Case Analysis

This section discusses the use case analysis. It comprises the following sections:

- Section C.1 provides background on the use case analysis.
- Section C.2 discusses the analysis method.
- Section C.3 examines supporting use case examples.

C.1 Background

This section provides an overview of use cases as they are used in the NIST CPS PWG. It serves to orient the reader and guide them through the remainder of the Use Case Analysis section. It is not intended to serve as a treatise on use cases (there are plenty of references on that), nor as a (necessarily incomplete) list of use cases for CPS systems. This section does, however, describe how to better understand the functional requirements for these systems by examining functional examples and use cases describing CPS systems. This will help to validate the reference architecture being developed by the CPS PWG, guide standards development organizations in the development of supporting standards, and assist software and hardware developers in the creation of supporting products.

C.1.1 Requirements

To understand how to design a system, it is important to understand what the goals of the system are, and what the requirements are that must be satisfied to achieve those goals. Developing use cases is one method of gathering functional requirements for a system based on the known ways the system will be used. Non-functional requirements are not typically captured in the use cases (but sometimes may be inferred from them). In the specific case of CPS, the CPS environment may provide additional value by supporting not just the known functions any component is designed for, but also promoting innovation and providing the flexibility to develop the new functionality that will accompany this innovation. The use cases find only those requirements driven directly by known uses of the systems, so the output of the Use Cases subgroup must be used with other methods of gathering requirements.

CPS use cases exhibit certain system *properties*. The collection of these *properties* distinguishes a system that expresses them as the model of the CPS. These *properties* can be derived from analysis of CPS *aspects* and *concerns*, such as timing, security, and data interoperability. Other types of systems can have *properties* in these areas, but these system *properties* must be fulfilled by any realized CPS architecture, and so become requirements placed upon the CPS Framework.

C.1.2 Relationship with Other CPS PWG Subgroups

Because use cases provide a link between each user's goals and the system properties as described above, there is a tight coupling between the use cases and the system or infrastructure architecture. This implies the need for tight coupling between the CPS PWG Use Cases and the Vocabulary and Reference Architecture subgroups.

The use cases are used both to check the scope of the CPS definition created by the Vocabulary and Reference Architecture subgroup and to derive a set of requirements that the CPS Framework must support. In this way the output of the Use Cases subgroup functions as input to the development of the CPS definition and the development of the CPS Framework. Once the CPS definition and architecture are complete, the use cases and requirements will be used to validate the definition and architecture.

The other subgroups are also linked together with the Use Cases subgroup. Each use case may have specific timing, security, and data interoperability requirements. Once these requirements are identified by the Use Cases subgroup, they will be fed to the appropriate subgroup for investigation. Additionally, any specific timing, security, or data interoperability use cases that are generated within the three subgroups will be fed into the Use Cases subgroup and included in the CPS PWG use case repository.

The interactions are bidirectional and started at the beginning of the PWG process to ensure that there will not be any major gaps at the end of the process.

C.1.3 Overview of CPS Use Cases

Use cases are a common technique for gathering requirements in systems of many sorts, including CPS. Each use case describes how an actor (the user) interacts with a system to achieve a goal. Use cases are used to elucidate functional behavior, with an emphasis on the value delivered to the users of the system. Each use case captures a function, or range of functions, required by the user, and acts as a guide to engineers responsible for developing the hardware and software that will make up the system.

A *user* refers to the actor that interacts with a system. A user can be a human or a constructed system. More generally, and especially in CPS, a user may be a person, machine, another system, or even the system itself, which may respond to an internally generated trigger. The *actor* concept represents a role that interacts with the system to cause it to carry out some function. Capturing the "real" requirements, however, requires a step back from the actors to also consider the constellation of entities affected by the system, such as regulators, corporate strategists, society, or the environment at large, collectively known as *stakeholders*.

In the case of a single system, a complete collection of identified use cases should comprise a complete set of functional requirements for that system. Experienced engineers then scan the collection of use cases for common aspects that can be implemented once and used in multiple

places. For example, a control system for a chemical plant might need to control both temperature and pressure with a deadband. We might invent, or take off our mental shelf, an implementation of a PID loop, or, more broadly, a control loop. The same implementation can be used in multiple contexts. From the other direction, the concept of an acceleration profile can be applied for an elevator, a robot arm, or a tape drive. Even though the specific application domains are different, the same pattern can be applied.

Because this process abstracts away from the specifics of a particular application, it is possible to go one step further and observe *collections* of interlocking patterns that often appear in similar *types* of systems, such as batch, event-driven, service-oriented, or CPS. Such collections of interlocking patterns of the elements of a (type of) system, what they are, and how they connect, are part of what is called the system's *architecture*.

Colloquially, however, architecture does not require a careful definition. For this document's purposes, it is a convenient term to refer to the abstract organization of the elements of a system and how they connect one to another. This is why use cases are being gathered: to identify the kinds of elements that comprise a CPS and how they are related, and from that to identify requirements and gaps in the architectures of CPS.

Broadly speaking, the process is to:

- Identify stakeholders
- Identify application categories
- Identify and elaborate CPS examples and use cases
- Identify architectural dimensions (high-level view)
- Identify primitive requirements for CPS architecture

However, the number of potential CPS use cases is practically infinite, and will continue to expand as CPS systems are applied in new ways and unimagined markets. For this reason, it is not possible to find all use cases. Instead a method (described in Section C.2) has been developed to enable analysis of sets of use cases with some repeatability at a high level and use of the analysis to decide whether they need further elaboration. The method is based on clustering use cases based on a set of characteristics particular to the architectures of CPS. These characteristics can be broadly grouped together (shown in Table 15) and include groups such as functional concerns (e.g., device control or analytics) and cross-cutting concerns (e.g., security or timing). Each use case can then be categorized as to whether or not it imposes requirements on timing, for example. Additional categorization can be done based on actors, application types, and systems, each aspect providing a different view into the system.

This structure is reflected in the structure of the report, which begins with the identified stakeholders, then the application types, and finally the requirement categories, showing relationship of the example/use-case to all the relevant requirements.

The following subsection describes the method used to evaluate and classify use cases, and how the requirements are then identified. The subsection after that describes just a few supporting use case examples.

Finally, the requirements identified on the architecture are listed. They are divided into requirements placed on the functional architecture and then the cross-cutting concerns of cybersecurity, timing, and data management.

C.1.4 Stakeholders

The stakeholders of a system are by definition "a person or group that has an investment, share, or interest in something, as a business or industry [260]." The users are usually perceived as the key stakeholders, but often the primary focus is on the usability of the system and the system performance in meeting the user goals. The secondary stakeholders are also important, and understanding them and their needs will provide better understanding of the system requirements. Table 14 lists the stakeholder groups identified by the Use Cases subgroup as important to the success of a system.

Table 14: List of Stakeholders

Classes of Stakeholders	**Who Are They?**
Creators	The builder, system integrator, project manager, etc. of the CPS
Owners	Those who own the CPS
Operators	Those who operate the CPS
Customers/users	Those who benefit from the function performed by the system
Supply chain providers	Third-party suppliers of components anywhere in the supply chain that end up in the CPS product
Service providers	Consultants, contractors, lawyers, bankers, etc.
Insurers	Insurance companies
Regulators	Mostly state and federal agencies responsible for developing and monitoring regulations.
Competitors	Companies in the same market as the entity that experienced a failure
Government	Representatives of the three branches of government. Includes local, state, and federal

C.1.5 Application Domains

The application domains or types describe the different business areas in which CPS are predicted to be used. Some of the core application areas include emergency response, where a CPS needs to be quickly assembled from an assorted set of (possibly not fully functional) components; manufacturing, where systems integration and maintainability can lead to cost savings and improved safety; defense systems with important reliability and security

requirements; and even advertising that is linked into events in the physical world. These are only a few of the exciting possibilities; the entire list of application domains is shown in Table 15. This list will be updated as new domains are uncovered. The most up-to-date list can be found on the www.cpspwg.org website.

Table 15: Application Domains of CPS

Domain	
Advertising	Entertainment/sports
Aerospace	Environmental monitoring
Agriculture	Financial services
Buildings	Healthcare
Cities	Infrastructure (communications, power, water)
Communities	Leisure
Consumer	Manufacturing
Defense	Science
Disaster resilience	Social networks
Education	Supply chain/retail
Emergency response	Transportation
Energy	Weather
…	

C.2 Analysis Method

The pool of potential use cases is infinite. This makes filtering the examples and use cases to a set that effectively covers the requirements a daunting task. Additionally, the degree of similarly between use cases can vary greatly, making it even more difficult to process examples and use cases. To overcome this problem, there must be a thorough evaluation of each use case to identify common properties. This process will allow the use cases to be clustered based on architectural characteristics so as to get coverage where there are gaps in requirements for the reference architecture. For example, if the collection of use cases exhibited only loose timing requirements, another use case with stringent timing requirements might be solicited. For the evaluation process to be effective, it is imperative that each example and use case is evaluated in a consistent manner. To this end, the Use Cases subgroup developed a standard approach to use case evaluation.

This method provides an approach to identify patterns of use of CPS-based solutions from a set of use cases corresponding to different types of applications. These patterns of use will determine the specific architectural requirements that can be organized and described in a CPS Framework. The patterns also illustrate the capabilities needed to run the processes of the applications of interest. In general, the methodology is intended to help a CPS-based solution stakeholder to describe the requirements of an application, i.e., the problem description. These requirements are inputs to the CPS-based solution providers both directly – as a set of requirements needed for a specific system or type of system — and indirectly through the CPS PWG reference architecture.

For this effort, the Use Cases subgroup will use a two-stage process designed to support differing uses for this information. The first step is to collect and analyze high-level CPS scenarios (which are referred to as “CPS Examples” to prevent confusion with how scenarios are used in use case terminology). These examples can describe complex interactions between several systems and may cross one or more application category boundaries. The examples will help indicate which requirements areas are important for that example and what the different actors and systems are (actors are a type of system, but in this case they are specific types of systems acting on another system). The CPS Example analysis phase will help provide valuable knowledge about the types of actors, systems, and their interactions along with a general understanding of the types of requirements needed for each example. This first stage will not provide the specific simple requirements that will be needed to thoroughly validate the architecture (and can also be used to validate any systems designed to meet the full set or a subset of the requirements). Phase two will fill that need.

To gather the more detailed, specific requirements necessary to validate the CPS architecture, the Use Cases subgroup will deconstruct CPS examples into a set of specific use cases. This process will include both black box use cases describing the specific interaction between an actor and a system with no knowledge of what goes on within the system; and white box use cases going into detail of the internal workings of the system. These specific use cases will then be analyzed using a set of primitive requirements which may be associated either with a use case or with a specific step within the use case. These primitive requirements will provide specific singular requirements that are mapped to specific steps within a use case (and therefore are associated with a specific actor and system). By looking at a set of these functional requirements for a specific instance of a system, organizations can then a) build a system based on these requirements or b) test a system based on these requirements.

The primitive requirements for CPS are being generated using a set of smart grid primitive requirements as the starting point. A model of the decomposition of requirements into primitives is shown in Figure 40. The thousand-plus requirements developed as part of the EPRI IntelliGrid project [261] are being modified and expanded to fit the more general needs of the CPS environment. The Use Cases subgroup will map the primitive requirements to high-level requirements categories.

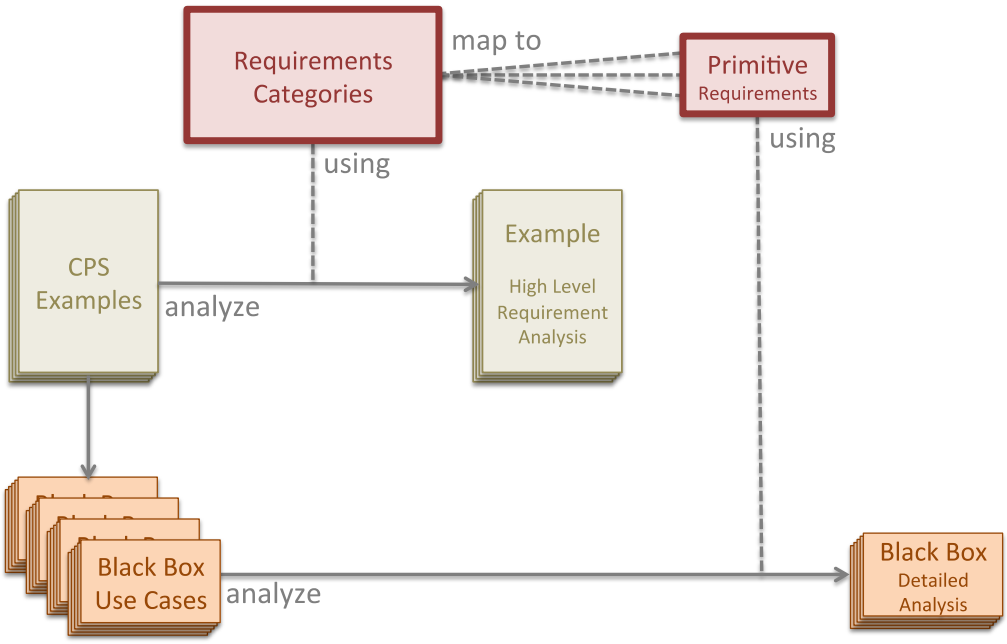


Figure 40: Requirements Decomposition into Primitives

The output of the Use Cases subgroup will be the requirement analyses of the set of CPS Examples and a set of primitive requirements for the set of specific use cases. While at first the output will only cover a selected set of important examples and use cases, over time it is desirable to cover all the requirements categories (see Section C.2.2).

C.2.1 Method for describing a CPS Example

The *CPS Example* is a use case summary describing a set of actors and systems that interact to achieve a variety of goals (not always the same goals). It contains information on the actors and systems. Systems can be actors as well; in this case a system is something that is acted upon, and an actor is the entity doing the acting on the system. The CPS Example differs in one major way from the specific use cases used in the second phase of this project – the example has multiple systems, actors, and interactions, while the specific use cases have only one. Table 16 provides a template for a CPS Example.

Table 16: CPS Example Template

CPS Example Template
CPS Example Name - phrase describes interaction between actor and system
Description - brief description
Notes – any relevant notes that help in understanding the use case
Goals – what goals do the stakeholder want to see achieved?
Use Case Source Organization - who developed the use case

Actors - the actor that interact with the systems described in the example
Systems - the systems being acted on by the actors described in the example

C.2.2 Requirement Categories

Once the CPS examples have been collected, the next step is to evaluate them in terms of their architectural characteristics. These characteristics cover questions like the volume and velocity of data, variability in data sizes, confidentiality, timing constraints, and computational effort. Since these characteristics are quite heterogeneous, they are grouped into two levels of categories, as shown in the first two columns in Table 17.

The architectural characteristics are directly related to the system properties described above. If a use case is part of a system that exhibits a need to collect data continuously (e.g., avionics that determine aircraft position), then this implies styles of implementation that can realize continuous behavior (e.g., an analog subsystem that must be integrated with the rest of the system), or a digital system that operates periodically. A *reference architecture* should be able to cater to both architectural characteristics.

As each use case is evaluated, after it is compared against the known characteristics, unique characteristics not covered by the standard form must be looked for. If there are such characteristics, the form will be modified to address the additional needs of the use case. The modified form will also be retroactively applied to previously processed and future use cases. This iterative approach will ensure that the methodology for evaluating use cases is comprehensive and adaptable to changing needs.

Table 17 below, therefore, is a starting point, rather than comprehensive. Architectural characteristics may be added based on known properties of CPS systems that are not reflected in the current set of use cases.

Table 17: Requirements Categories

Aspect	Requirement Category	Description
Boundaries	**Application Areas**	Does the use case require a system that crosses multiple application areas? If so, which application areas are included?
Composability	**Composition**	Does the use case require the interaction of heterogeneous subsystems?
Composability	**Intersystem Interaction**	Are there specific requirements caused by the use case interacting with legacy systems?
Human	**Human Interaction**	Are humans an important part of the system?
Functional	**Physical Properties**	What physical properties are being monitored?
Functional		What physical properties are being acted upon?

Data	**Volume and Velocity**	Describe the size of the datasets being processed and the speed at which they come into/out of the system.
Functional	**Computation**	Describe the computation effort and processing required to achieve the use case goals.
Data	**Aggregation**	Describe the requirements to aggregate different data types
Data	**Variability**	Is the size of data being generated/used consistent or is there a growth/shrinkage trend?
Functional	**Error Sensitivity**	Describe the sensitivity of the system to errors in the data.
Functional	**Certainty**	What is the level of uncertainty in the data being generated/processed and the assurance of the resulting actions taken by the system?
Timing	**Timeliness**	What are the use case timing constraints?
Timing	**Time Synchronization**	What are the use case time synchronization requirements?
Boundaries	**Physical Location**	What are the location requirements of the use case?
Trustworthiness	**Robustness**	What are the robustness requirements? (preventing a fault)
Trustworthiness	**Resilience**	What are the resiliency requirements? (recovering from a fault or sub-fault)
Trustworthiness	**Confidentiality**	What happens if information within the system leaks (or is pulled) out?
Trustworthiness	**Integrity**	What happens if the system acts on incorrect data (including software)?
Trustworthiness	**Availability**	What happens if the system or data it generates is not accessible and prepared to function properly when and where needed?

C.2.3 Method for describing a Specific CPS Use Case

Once a CPS Example has been identified, along with the associated systems/actors, it will be broken down into a set of specific use cases describing specific interactions between an actor and a system. The resulting use cases will be described using a template based on traditional use case design, focusing on the actor, the system, pre and post conditions, and the steps between the two. The full use case template is shown in Table 18.

Table 18: Black Box Use Case Template

SPECIFIC USE CASE TEMPLATE
Use Case Name - Phrase describing interaction between actor and system
Use Case Description - Brief description
Notes – Any relevant notes that help in understanding the use case
Goal – What goal performing the use case achieves
Use Case Source Organization - Who developed the use case

Actor - The actor that performs the steps in the use case
System - The system being acted on in the use case
Pre-Conditions - A list of true conditions before the use case starts
Steps - A list of steps to perform the use case
Post-Conditions - A list of true conditions when the use case ends

Since this effort focuses on deriving CPS requirements from the use cases, a list of simple (primitive) requirements will be used to associate each step of a black box use case with a set of requirements. The primitive requirements will be developed using a set of simple requirement statements numbering in the thousands. These simple requirements will be generalized (as is appropriate for CPS covering a wide range of application types) and mapped to the requirement categories used in the high-level requirements analysis.

As new simple requirements are identified during the use case analysis, they will be added to the set of requirements. The set of primitive requirements will be used to validate the CPS against a set of known CPS functions as the analysis effort approaches completion. The effort can never be finished, as more examples and use cases will be added as they are discovered. In fact this trend might increase as the new capabilities drive the Use Cases subgroup's imaginations.

Not only can these simple requirements be used to test the CPS reference architecture, but they can also be used to describe and test any specific instance of a CPS. If these requirements are used in the development of CPS components, it will become easier to assemble systems and efficiently make use of available resources. The primitive requirements can be used in different ways:

- By grouping the set of requirements together for a use case, the specific use case can be tested.
- By grouping the set of requirements for a specific system, the system can be designed and tested.
- By grouping all the requirements together, the architecture can be validated (this is described in the next section).

C.2.4 Procedure for Identifying Reference Architecture Requirements

Once all the identified use cases have been processed using this method, the outcome will be a set of characteristics for the use case that the supporting system must be able to meet. While the specifics of these characteristics will be specific to each individual use case, the collection will represent a comprehensive set of use case needs. The next step is to translate the needs into requirement statements that will be levied against the Vocabulary and Reference Architecture, Timing and Synchronization, and Cybersecurity and Privacy subgroups. The Use Cases subgroup will analyze and abstract each characteristic away from its corresponding use

case, grouping the characteristics based upon similarity and removing any duplicates. The result of this process will be a generalized set of needs that will serve as requirements for the other subgroups.

C.3 Supporting CPS Use Case Examples with Evaluation

Following are two CPS Examples that have been submitted and then analyzed by the Use Cases subgroup for an initial high-level analysis based on the requirement categories.

C.3.1 CPS Example – Monitoring Manufacturing System Energy Efficiency

In this example, the energy efficiency index of a manufacturing system is needed for reconfiguration and rescheduling, in a run-to-run basis.

Example Description –Figure 41 illustrates a manufacturing system architecture. Level 3 manufacturing operations management obtains a set of production Key Performance Indicators (KPIs) based on Level 2 and Level 1 operational data about the process, equipment, and product. The energy efficiency indices are derived from the production KPIs and used to generate the new manufacturing system parameters for reconfiguration and adjustments to scheduling before the next set of production orders are done.

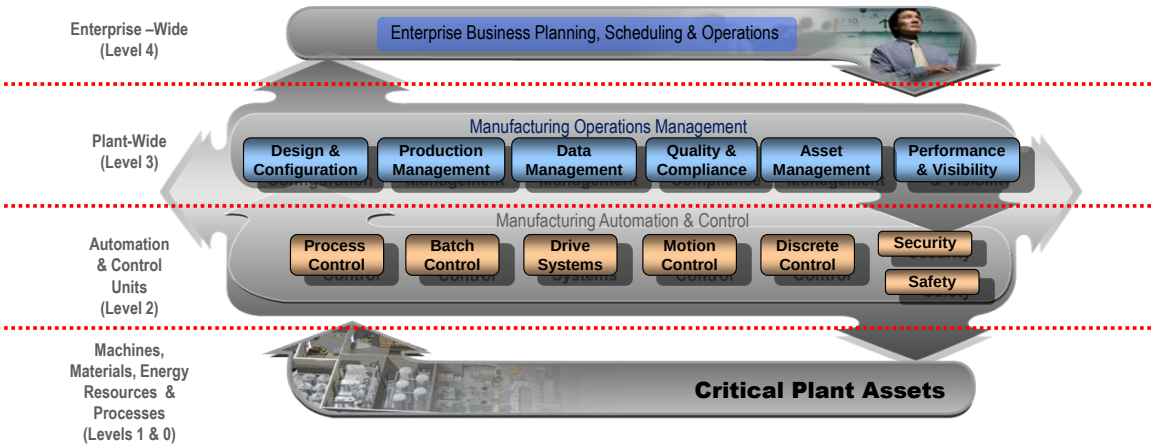


Figure 41: Example of Reference Architecture Model of "Manufacturing" System-of-Interest

Details - A production order prepared at Level 4 of the enterprise has been scheduled for execution at Level 3 with a set of manufacturing resources allocated, configured, validated, and dispatched to process the provisioned materials and energy flows and output the required finished goods, at the lower levels (2, 1, 0), in a work request with detailed workflows.

A work request is sent by a Level 3 MOM application to Level 2 manufacturing control and automation application. A sequence of procedural automation steps is performed by Level 2 automation units to direct the Level 1 sensing, control, and actuation units that conduct the

production processes and machines (at Level 0) required to produce the desired outputs of the manufacturing system. A combination of data acquisition units collects real time data about the process, materials, energy flows, equipment, and personnel that provide the basis for generating the relevant KPIs for evaluating the energy efficiency index of the manufacturing system. A Level 4 production performance tracking application evaluates the energy efficiency index of the current production run and estimates any needed changes to the configuration and scheduling parameters in order for the next production run to achieve the production objectives in quality, cost, timeliness, and safety.

The architectural characteristics of this example use case are shown below in Table 19, without the first column used to group them, so as to save space.

Table 19: Analysis of Use Case

Application Areas	Does the use case require a system that crosses multiple application areas? If so, how many application areas are included?	Across several domains of an enterprise; among functional and resource levels
Composition Intersystem Interaction	Does the use case require the interaction of heterogeneous subsystems?	Systems of processes, resources, and organizational units
	Are there specific requirements caused by the CPS-based solution interacting with legacy systems?	Many of the identified heterogeneous subsystems can be considered as “legacy” types
Human Interaction	Are humans an important part of the system?	Critical to the objectives of an enterprise, e.g., in task prioritization, fault recognition & recovery
Physical Properties	What physical properties are being monitored?	Wide range of physical variables involved in the material and energy conversions plus equipment and personnel coordination to make a product
	What physical properties are being acted upon?	Process, product, equipment personnel properties to be set at target values needed to complete production
Volume and Velocity	Describe the size of the datasets being processed and the speed at which it comes into/out of the system.	PLC “I/O data tables” for control loops closed in millisecond cycles up to MOM KPI targets and results composed and conveyed in seconds
Computation	Describe the computation effort and processing required to achieve the use case goals.	Processing efforts scales according to size of enterprise and required throughput of products

Aggregation	Describe the requirements to aggregate different data types.	Both composition and decomposition tasks performed on signals, data and information that are at and cross multiple levels and domains [hundreds of megabytes per run or job]
Variability	Is the size of data being generated/used consistent or is there a growth/shrinkage trend?	"Data" associated with various forms, e.g., text, graphics, audio, video, or encoded/compressed bit streams typically span tens of bytes up to tens of MBs per transaction (more in future)
Error Sensitivity	Describe the sensitivity of the system to errors in the data.	Critical product tolerances have to be maintained at parts per billion [with or without fault tolerance mechanisms] [exception reporting capabilities to mitigate]
Certainty	What is the level of uncertainty in the data being generated/processed and the assurance of the resulting actions taken by the system?	Very wide range; floating point and 64-bit integer computation mostly a starting point
Timeliness	What are the use case timing constraints?	See above (Volume and Velocity)
Time Synchronization	What are the use case time synchronization requirements?	Tens of processing lines with 10K I/O points per line and job cycles up to 1800 items/hr per line
Physical Location	What are the location requirements of the use case?	Manufacturing and production sites occupy 1-2M square feet per site, with multiple sites in different regional locations;
Robustness	What are the robustness requirements? (preventing a fault)	For example, MTBF is 5K hours
Resilience	What are the resiliency requirements? (recovering from a fault or sub-fault)	Fault recovery is acceptable if it does not affect production
Confidentiality	What happens if information within the system leaks (or is pulled) out?	Intellectual property losses. Recommended encryption: 128-bit and higher (AES)
Integrity	What happens if the system acts on incorrect data (including software)?	Loss in productivity and work safety on the order of >$1M/month
Availability	What happens if the system or data it generates is not accessible and prepared to function properly when and where needed?	Fault causes loss in productivity

For the next production run, a new work request and associated workflow have been prepared with a set of resource configurations and schedules. The variances in the previous production

run denoted in the KPIs and the energy efficiency index have been converted into a set of target production drivers for the next production run.

Notes – Obtaining information about the real time manufacturing system's capabilities and controlling the behavior of the automation units throughout the multiple physical, cyber, and cyber-physical domains involve the use of human interface units, advanced sensing units, actuation units, and control and optimization units.

Example Goals – highly energy efficient manufacturing with high quality and timely delivered products

Systems/Actors

- MOM application
- Control and automation system
- Production equipment
- Materials, personnel, and energy handling units

C.3.2 CPS Example – Grain/Produce Monitoring and Delivery

Ingredients with specific characteristics are required for the production of a food product. Food producers and ingredient vendors collaborate to get appropriate ingredients delivered for production. Before shipment, vendors send ingredient samples to a lab for analysis and have the results sent to the food producer. The food producer uses the analysis results to adjust manufacturing plans. The adjustments may include stopping shipments of unacceptable ingredients, determining which food product batch is best to use the ingredients in, and/or modifying the production process for the food production batch that is to use the ingredients.

Since the properties of ingredients can change during transit, they may be monitored via sensors during the shipment. Manufacturing planning may make use of the sensor information if it exists.

The systems that need to interact include supply chain and production systems. The interactions involve multiple layers of communication systems – sensor communication over mobile network, business-to-business communication, and application-to-application communication. The communication topology may be peer-to-peer or hub/intermediary-based. Sensors may need to be able to regularly join and adjourn different food producers' networks because trucks used for transporting ingredients likely do not belong to the food producer (e.g., may belong to a third-party logistics service provider or the grain vendor or farmer).

2.6.1.1 Example goals

What goals does performing the use case achieve?

Information about variations in the characteristics of input ingredients is available in time for a) the food producer to reject unacceptable ingredients before shipment and b) production planning to modify the food production process to account for ingredient variations.

2.6.1.2 Systems/actors

- Farmer
- Testing lab
- Trucker/truck
- Container
- Customer

2.6.1.3 High-level review

Table 20 summarizes the analysis for this Use Case.

Table 20: High-Level Review - Grain/Produce Analysis and Monitoring

Application Areas	Does the use case require a system that crosses multiple application areas? If so, how many application areas are included?	YES - Supply chain, manufacturing, transportation, agriculture
Composition	Does the use case require the interaction of heterogeneous subsystems?	YES
Intersystem Interaction	Are there specific requirements caused by the use case interacting with legacy systems?	YES, but not explicit (example – existing lab often can only send hardcopy of the data)
Human Interaction	Are humans an important part of the system?	The lab may employ humans in critical roles. The manufacturer will employ humans in decision-making roles.
Physical Properties	What physical properties are being monitored?	Temperature, humidity/moisture, light levels, time, location, biological, grain/produce properties
	What physical properties are being acted upon?	The produce/grain (location, manufacturing process, shipment acceptance)
Volume and Velocity	Describe the size of the datasets being processed and the speed at which they come into/out of the system.	Data needs to go through multiple heterogeneous systems. Truck monitoring data could get large.
Computation	Describe the computation effort and processing required to achieve the use case goals	Some on the laboratory (measurement/calculation) side, maybe some on the process reformulation side
Aggregation	Describe the requirements to aggregate different data types	Test data needs to be combined. ID and other metadata needs to be combined. Customer

		specification (ingredient spec) may be created from multiple data sources.
Variability	Is the size of data being generated/used consistent or is there a growth/shrinkage trend?	Consistent
Error Sensitivity	Describe the sensitivity of the system to errors in the data	Depends on property being measured. Can be HIGH – error can cause large monetary cost. If contaminated could lead to sickness or loss of life.
Certainty	What is the level of uncertainty in the data being generated/processed and the assurance of the resulting actions taken by the system?	Unknown. See error sensitivity. Predictive modeling causes additional uncertainties.
Timeliness	What are the use case timing constraints?	Truck monitoring data – minutes (resolution and latency). Lab turnaround – time to send grain/produce to the lab + time for analysis and data transmission. Analysis and data transmission time – minutes to hours
Time synchronization	What are the use case time synchronization requirements?	Truck lab and farm data needs to be synchronized, but requirements are not very hard to meet. Need timestamps.
Physical Location	What are the location requirements of the use case?	Multiple locations. Supplier and OEM customer are possibly separated by large distances. Suppliers might not have good communication access. A distribution truck is mobile and has a dynamic location. Location data for specific produce is important.
Robustness	What are the robustness requirements? (preventing a fault)	Cost of lack of production. High liability cost if something goes wrong (and not monitored). Failure is better than error.
Resilience	What are the resiliency requirements? (recovering from a fault or sub-fault)	Resilience is possible if it meets the other requirements of the use case (especially timing requirements)
Confidentiality	What happens if information within the system leaks (or is pulled) out?	The confidentiality of data is important to protect the manufacturer's secret recipe. Sensors as well as data streams need to be protected. Data about produce may be authorized for specific actors. Devices, farmers, lab staff, truckers/trucking staff, and manufacturer staff all have different access needs.
Integrity	What happens if the system acts on incorrect data (including software)?	Misinformation could cause the customer large amounts of harm if the recipe used is dependent on the data from produce/grain measurement results.

Availability	What happens if the system or data it generates is not accessible and prepared to function properly when and where needed?	The manufacturer might not receive critical information about the produce shipment being purchased resulting in additional costs and time delays.

C.4 Specific Use Cases

The specific use cases (both black box and white box) will be developed from the CPS Examples as key examples are identified.

C.4.1 Detailed Analysis

Detailed analysis will be done on carefully selected specific use cases.

C.5 Current CPS Examples and Black Box Use Cases

The CPS Examples and Black Box Use Cases will be available from the CPS PWG website as they are developed: http://www.cpspwg.org

Appendix D. References

This section provides references to a variety of CPS-related articles, standards, and other material.

D.1 Reference Architecture

[1] Internet of Things – Architecture (IoT-A), "Final architectural reference model for the IoT v3.0", http://www.iot-a.eu/public/public-documents/d3.1, 2013.

[2] ISO/IEC/IEEE 42010, "Systems and software engineering – architecture description", 2011, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6129467

[3] ISO/IEC/IEEE FDIS 15288:2014(E), "Systems and software engineering - System life cycle processes", http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6994196

[4] TOGAF Version 9.1, 2011, http://pubs.opengroup.org/architecture/togaf9-doc/arch/

[5] CMMI Version 1.3, http://www.sei.cmu.edu/cmmi/tools/cmmiv1-3/

[6] Svetlana Nikitina, "Translating qualitative requirements into design choices – evaluating the method proposed in the architectural reference model for the internet of things", ERCIS, 2014.

[7] IEEE P2314, "Standard for an Architectural Framework for the Internet of Things (IoT)", Webinar, June 13, 2014.

[8] Kevin Forsberg and Harold Mooz, "The Relationship of System Engineering to the Project Cycle," in Proceedings of the First Annual Symposium of National Council on System Engineering, link, October 1991.

[9] [IOT-A], EU IOT-A Terminology http://www.iot-a.eu/public/terminology/copy_of_term

[10] IHMC, http://www.ihmc.us/groups/datkinson/wiki/fcb0e/intelligent_system_autonomy_automation_robots_and_agents.html

[11] Standards for M2M and the Internet of Things, OneM2M, http://www.onem2m.org/technical/published-documents

[12] ISO 7498-2:1989, Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture

[13] ISO TS 19104:2008, Geographic information – Terminology

[14] ISO/IEC 14814:2006, Road transport and traffic telematics — Automatic vehicle and equipment identification — Reference architecture and terminology

[15] ISO/IEC 24760-1:2011 , Information technology — Security techniques — A framework for identity management — Part 1: Terminology and concepts

[16] ISO/IEC 24791-1:2010, Information technology — Radio frequency identification (RFID) for item management — Software system infrastructure — Part 1: Architecture

[17] ISO/IEC 27000:2014, Information technology — Security techniques — Information security management systems — Overview and vocabulary http://standards.iso.org/ittf/PubliclyAvailableStandards/c063411_ISO_IEC_27000_2014.zip .

[18] ISO/IEC DIS 18834-1, RA SOA – Terminology and Concepts

[19] ISO/TS 19129:2009, Geographic information — Imagery, gridded and coverage data framework

[20] ISO/TR 14252:1996, Information technology -- Guide to the POSIX Open System Environment (OSE)

[21] OED, Oxford Dictionary of English, 2nd Edition

[22] Industrial Internet Consortium, http://www.industrialinternetconsortium.org/

[23] ISO 26262-1:2011(en), Road vehicles – Functional safety, https://www.iso.org/obp/ui/#iso:std:iso:26262:-1:ed-1:v1:en

[24] Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 4, September 2012. Part 3: Security assurance requirements, CCMB-2012-09-003.

D.2 Trustworthiness

[25] The Framework for Improving Critical Infrastructure Cybersecurity, http://nist.gov/cyberframework/upload/cybersecurity-framework-021214.pdf , Use to guide what components need to be included in a framework intended to address multiple domains, Multiple

[26] NIST Interagency Report 7628 Rev. 1, Guidelines for Smart Grid Cybersecurity, http://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf , Use to guide what considerations need to be included in a framework intended to address multiple stakeholders within a single domain, Energy

[27] NIST SP 800-53 Rev. 4, Security and Privacy Controls for Federal Information Systems and Organizations, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-

53r4.pdf , Use security controls to guide what CPS framework elements need to be considered and factored in, Multiple

[28] NIST SP 800-82 Rev. 1, Guide to Industrial Control Systems (ICS) Security, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r1.pdf , Use to further bolster the list of differences between IT and CPS systems, Multiple

[29] ISO/IEC 2700x- Information security management, http://www.iso.org/iso/home/standards/management-standards/iso27001.htm, Use similar to how NIST 800-53 is used, but includes a more international perspective (see NIST 800-53 for a mapping between 800-53 and ISO controls), Multiple

[30] ISA/IEC 62443 Series, Industrial Automation and Control Systems Security, http://isa99.isa.org/ISA99%20Wiki/Master-Glossary.aspx, NOTE: Only the IEC62443-3-3 System Security Requirements and Levels is published (final), but requires membership to access the document. Use Master Glossary to determine common terminology being used across international community regarding industrial automation and control systems, Multiple

[31] Electric Sector Failure Scenarios and Impact Analyses, http://smartgrid.epri.com/doc/NESCOR%20failure%20scenarios09-13%20finalc.pdf, Use to guide creation of additional CPS cybersecurity use cases, Energy

[32] National Infrastructure Protection Plan (NIPP) 2013 - Partnering for Critical Infrastructure Security and Resilience, http://www.dhs.gov/sites/default/files/publications/NIPP%202013_Partnering%20for%20Critical%20Infrastructure%20Security%20and%20Resilience_508_0.pdf

[33] Health Insurance Portability and Accountability Act (HIPAA) Security Rule

[34] HIPAA Privacy Rule

[35] Health Information Technology for Economic and Clinical Health (HITECH) Act

[36] 45 CFR Parts 160 and 164 - Modifications to the HIPAA Privacy, Security, Enforcement, and Breach Notification Rules Under the Health Information Technology for Economic and Clinical Health Act and the Genetic Information Nondiscrimination Act; Other Modifications to the HIPAA Rules; Final Rule, http://www.hhs.gov/ocr/privacy/hipaa/understanding/srsummary.html

[37] http://www.hhs.gov/ocr/privacy/hipaa/understanding/summary/index.html

[38] http://www.healthit.gov/sites/default/files/hitech_act_excerpt_from_arra_with_index.pdf (see Title XIII)

[39] http://www.gpo.gov/fdsys/pkg/FR-2013-01-25/pdf/2013-01073.pdf, Use to inform health information and information exchange security and privacy considerations that need to be reflected in the CPS framework, Healthcare and Public Health

[40] Modeling and Simulation for Cyber-Physical System Security Research, Development and Applications, http://prod.sandia.gov/techlib/access-control.cgi/2010/100568.pdf, The Virtual Control System Environment (VCSE) Framework and Architecture (p. 11 of 27) diagrams and descriptions assist in understanding a basic framework that is intended to describe the portions of a CPS that are similar across domains. Can use this to help guide our framework that must cover multiple CPS domains., Multiple

[41] NFPA 3: Recommended Practice for Commissioning of Fire Protection and Life Safety Systems, 2015 Edition, 2015 Edition, http://www.nfpa.org/catalog/category.asp?category_name=Codes+and+Standards&Page=1&src=catalog , Note: Documents require payment - cannot determine relevance without reading, Multiple

[42] NFPA 4: Standard for Integrated Fire Protection and Life Safety System Testing, 2015 Edition, http://www.nfpa.org/catalog/category.asp?category_name=Codes+and+Standards&Page=1&src=catalog , Note: Documents require payment - cannot determine relevance without reading, Multiple

[43] NFPA 1600®: Standard on Disaster/Emergency Management and Business Continuity Programs, 2013 Edition, http://www.nfpa.org/catalog/category.asp?category_name=Codes+and+Standards&Page=1&src=catalog , Note: Documents require payment - cannot determine relevance without reading, Multiple

[44] Whole Building Design Guide - Cybersecurity, http://www.wbdg.org/resources/cybersecurity.php, Use to see an example of how cybersecurity is applied to operational technology and industrial control systems; leverage concepts to guide establishment of cybersecurity framework for CPS broadly, Multiple

[45] Securing government assets through combined traditional security and information technology, http://www.dhs.gov/interagency-security-committee-standards-and-best-practices, Use to see how operational technology (OT) and information technology (IT) security considerations overlap and diverge to help understand unique security considerations for CPS Note: The link is only to the Interagency website; the actual report that is listed in Column A "Title" is not publicly available., Multiple

[46] Basic Concepts and Taxonomy of Dependable and Secure Computing, http://www.landwehr.org/2004-aviz-laprie-randell.pdf, Use to determine common terminology that can be referenced in our Framework publication, Multiple

[47] Homeland Security President Directive - 12 (HSPD-12) Implementation Standards and Testing, http://www.gsa.gov/portal/content/105233

[48] http://www.idmanagement.gov/ficam-testing-program , Use to inform efforts to incorporate credentialing into the Internet of Things (IOT) concept within CPS, Multiple

[49] Cyber Security Research Alliance - Roots of Trust for Cyber Physical Systems, http://cybersecurityresearch.org/

[50] http://cybersecurityresearch.org/documents/Roots_of_Trust_for_Cyber_Physical_Systems_Abstract_-_November_2014.pdf, Use full report (must request such via website; Abstract only is directly available) to leverage CPS taxonomy for CPS PWG report, Multiple

[51] Object Management Group (OMG) Industrial Internet of Things (IIOT), http://www.omg.org/hot-topics/iot-standards.htm, Acknowledge OMG's work in CPS PWG conclusions to express awareness of relevant parallel activity to further the credibility and usefulness of CPS PWG report , Multiple

[52] ETSI specification for Internet of Things and Machine to Machine Low Throughput Networks, http://www.etsi.org/news-events/news/827-2014-09-news-etsi-new-specification-for-internet-of-things-and-machine-to-machine-low-throughput-networks

[53] The Open Group Internet of Things (IoT) Work Group, http://www.opengroup.org/getinvolved/workgroups/iot

[54] “An Immunity Based Network Security Risk Estimation”, Li Tao, Ser. F Information Sciences 2005 Vol.48 No.5 ff7-578.

[55] http://www.cybersecurityresearch.org/about_us.html

[56] http://cps-vo.org/

[57] https://www.nitrd.gov/nitrdgroups/index.php?title=Cyber_Physical_Systems_(CPS_SSG)#title

[58] http://csrc.nist.gov/projects/privacy_engineering/index.html

[59] IEEE 802.1 Time Sensitive Networking Group, http://www.ieee802.org/1/pages/tsn.html.[31, 32]

[60] IEEE 802.11, "Wireless Local Area Networks", http://www.ieee802.org/11/

[61] 6TISH, "RFC 7554 on Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", IETF, 14th May 2015.

[62] AVnu, www.avnu.org.

[63] http://www.industrialinternetconsortium.org/

[64] http://www.dhs.gov/nstac

[65] "Trusted Computing Group: Where Trust Begins," Trusted Computing Group briefing Sept. 24, 2014, slide 7, http://www.trustedcomputinggroup.org/files/resroucefiles/.

[66] Kushner, David (2013). "The Real Story of Stuxnet" Spectrum.IEEE.Org (North American), Mar 2013, pp 49-53. http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet#

[67] http://arstechnica.com/security/2014/11/stuxnet-worm-infected-high-profile-targets-before-hitting-iran-nukes/

[68] Amit, "The Convergence of Engineering Disciplines in Modern Product Development," accessed 1 December 2014 at https://www.ibm.com/developerworks/community/blogs/invisiblethread/entry/the_convergence_of_engineering_disciplines_in_modern_product_development?lang=en_us, Sept 18, 2013.

[69] Fred B. Schneider, Trust in cyberspace, http://www.nap.edu/catalog/6161/trust-in-cyberspace

[70] Avizienis A., Laprie J.-C., Randell B., Landwehr C., Basic concepts and taxonomy of dependable and secure computing, Dependable and Secure Computing, IEEE Transactions on (Volume:1 , Issue: 1), http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1335465&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1335465

[31] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards/ieee.org/).

[32] The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[71] http://www.inl.gov/technicalpublications/documents/4680346.pdf, page 16

[72] http://web.ornl.gov/sci/electricdelivery/pdfs/ORNL_Cybersecurity_Through_Real-Time_Distributed_Control_Systems.pdf

[73] S. M. Amin, "U.S. electrical grid gets less reliable," IEEE Spectrum, p. 80, January 2011, http://dl.acm.org/citation.cfm?id=2244627

[74] http://tools.ietf.org/pdf/rfc4949.pdf

[75] http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4577833&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4577833

[76] http://www.deviceauthority.com/solutions/m2m-authentication-for-government-and-iot

[77] http://www.utdallas.edu/~alvaro.cardenas/papers/NordSec2013.pdf

[78] http://www.usatoday.com/story/opinion/2014/02/20/christine-todd-whitman-chemicals/5612695/

[79] https://www.priv.gc.ca/information/research-recherche/2014/wc_201401_e.pdf

[80] http://www.marketresearchreports.biz/analysis-details/wearable-technology-market-global-scenario-trends-industry-analysis-size-share-and-forecast-2012-2018

[81] IEEE Transactions on Dependable and Secure Computing

[82] Disaster Resilience Framework, 50% draft

[83] Presidential Policy Directive (PPD)-21: Critical Infrastructure Security and Resilience

D.3 Data Interoperability

[84] ISO/IEC CD 11179-1 Information Technology -- Metadata Registries (MDR) - Part 1: Framework Ed 3

[85] Kopetz, H et al. Direct versus Stigmergic Information Flow in System-of- Systems. Proc. of SoSE 2015. San Antonio, TX, pp. 36-41. pp.36-41.IEEE Press

[86] Satisfiability. (n.d.). Retrieved October, 2014, from http://en.wikipedia.org/wiki/Satisfiability

[87] Hodges, Wilfrid, "Model Theory", The Stanford Encyclopedia of Philosophy (Fall 2013 Edition), Edward N. Zalta (ed.). Retrieved October, 2014, from http://plato.stanford.edu/archives/fall2013/entries/model-theory.

[88] ISO 42010:2011

[89] http://www.rickmurphy.org/gag-modest.zip

[90] Package Java Util. Function. (2014). Retrieved October, 2014, from http://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html

[91] Class LambdaMetafactory. (2014). Retrieved October, 2014, from http://docs.oracle.com/javase/8/docs/api/java/lang/invoke/LambdaMetafactory.html

[92] http://www.rickmurphy.org/gag-modest.zip

[93] OCaml. (2014). Retrieved October, 2014, from http://ocaml.org/

[94] The Scala Programming Language. (2014, January 1). Retrieved October, 2014, from http://www.scala-lang.org/

[95] Duggal, D. (2014, August 14). Semantic SOA makes Sense! Retrieved October, 2014, from http://www.dataversity.net/semantic-soa-makes-sense/

[96] Kahn, R., & Wilensky, R. (2006). A Framework for Distributed Digital Object Services. International Journal on Digital Libraries, 6(2), 115-123. Retrieved October, 2014, from http://link.springer.com/article/10.1007/s00799-005-0128-x?no-access=true

[97] Institute of Electrical and Electronics Engineers, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, New York, NY: 1990.

[98] http://en.wikipedia.org/wiki/Software_assurance Software assurance. (n.d.). Retrieved October 7, 2014, from http://en.wikipedia.org/wiki/Software_assurance

[99] http://en.wikipedia.org/wiki/Evaluation_Assurance_Level Evaluation Assurance Level. (n.d.). Retrieved October 7, 2014, from http://en.wikipedia.org/wiki/Evaluation_Assurance_Level

[100] http://en.wikipedia.org/wiki/National_Information_Assurance_Glossary Hall, David L. and James Llinas (1997). "An introduction to multisensory data fusion," Proceedings of the IEEE, vol. 8, no. 1, pp. 6-23.

[101] Bizer, Christian, Tom Heath and Tim Berners-Lee (2009). "Linked data – The story so far," in Heath, T., Hepp, M., and Bizer, C. (eds.). Special Issue on Linked Data, International Journal on Semantic Web and Information Systems (IJSWIS), vol. 5, no. 2, pp. 1-22.

[102] JDL (1991). "Data Fusion Lexicon," Technical Panel For C3, F.E. White, San Diego, Calif: Code 4.

[103] Castanedo, Federico (2013). “A Review of Data Fusion Techniques,” The Scientific World Journal, Volume 2013, Article ID 704504, accessed 3 October 2014 at http://www.hindawi.com/journals/tswj/2013/704504/#B39.

[104] "Framework for discovery of identity management information" [available free of charge at http://www.itu.int/rec/T-REC-X.1255-201309-I; ITU-T announcement: http://newslog.itu.int/archives/137] was approved at an International Telecommunication Union (ITU) meeting in Geneva (ITU-T Study Group 17 (Security)) on September 4, 2013.

[105] “Overview of the Digital Object Architecture,” http://www.cnri.reston.va.us/papers/OverviewDigitalObjectArchitecture.pdf.

[106] Lyons, Patrice A. and Kahn, Robert E., "The Handle System and its Application to RFID and the Internet of Things,” RFIDs, Near-Field Communications and Mobile Payments; A Guide for Lawyers, edited by Sarah Jane Hughes, ABA Cyberspace Law Committee, 2013, pp. 257-270 (http://hdl.handle.net/4263537/5046).

[107] Berners-Lee, T., James Hendler, and O. Lassila, 2001. “The Semantic Web.” Scientific American, May, 29-37

[108] Berners-Lee, Tim, Christian Bizer, and Tom Heath. "Linked data-the story so far." International Journal on Semantic Web and Information Systems 5.3 (2009): 1-22.

[109] Ian Jacobs; Norman Walsh. Architecture of the World Wide Web, Volume One. 15 December 2004. W3C Recommendation. URL: http://www.w3.org/TR/webarch/

[110] Richard Cyganiak, David Wood, Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, 25 February 2014. URL: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. The latest edition is available at http://www.w3.org/TR/rdf11-concepts/

[111] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition). 11 December 2012. W3C Recommendation. URL: http://www.w3.org/TR/owl2-overview/ www.w3.org/TR/2012/REC-owl2-overview-20121211/

[112] Fabien Gandon; Guus Schreiber. RDF 1.1 XML Syntax. 9 January 2014. W3C Proposed Edited Recommendation. URL: http://www.w3.org/TR/rdf-syntax-grammar/

[113] W3C SPARQL Working Group. SPARQL 1.1 Overview. 21 March 2013. W3C Recommendation. URL: http://www.w3.org/TR/sparql11-overview/

[114] Information-Centric Networking (icnrg), IETF, https://datatracker.ietf.org/rg/icnrg/documents/

[115] ISO/IEC/IEEE P21451-1-4

[116] http://www.dtic.mil/dtic/tr/fulltext/u2/680815.pdf

[117] http://en.wikipedia.org/wiki/Evaluation_Assurance_Level

[118] https://www.niap-ccevs.org/

[119] http://en.wikipedia.org/wiki/Common_Criteria

[120] ISO/IEC 2382-1:1993, definition 01.01.02

[121] Jamshidi, M. 2005, Theme of the IEEE SMC 2005, Waikoloa, Hawaii, USA, http://ieeesmc2005.unm.edu, Oct 2005.

[122] ISO 9000:2015, Quality management systems – Fundamentals and vocabulary

[123] ISO/TS 8000 Data Quality
- ISO/TS 8000-1:2011, Data quality — Part 1: Overview
- ISO 8000-2:2012, Data quality — Part 2: Vocabulary
- ISO/TS 8000-100:2009, Data quality — Part 100: Master data: Exchange of characteristic data: Overview
- ISO 8000-102:2009, Data quality — Part 102: Master data: Exchange of characteristic data: Vocabulary
- ISO 8000-110:2009, Data quality — Part 110: Master data: Exchange of characteristic data: Syntax, semantic encoding, and conformance to data specification
- ISO/TS 8000-120:2009, Data quality — Part 120: Master data: Exchange of characteristic data: Provenance
- ISO/TS 8000-130:2009, Data quality — Part 130: Master data: Exchange of characteristic data: Accuracy
- ISO/TS 8000-140:2009, Data quality — Part 140: Master data: Exchange of characteristic data: Completeness
- ISO/TS 8000-150:2011, Data quality — Part 150: Master data: Quality management framework

[124] ISO 22745, Open technical dictionaries and their application to master data
- Part 1: Overview and fundamental principles
- Part 2: Vocabulary
- Part 10: Dictionary representation
- Part 11: Guidelines for the formulation of terminology
- Part 13: Identification of concepts and terminology
- Part 14: Dictionary query interface
- Part 20: Procedures for the maintenance of an open technical dictionary
- Part 30: Identification guide representation (data specification)
- Part 35: Query for characteristic data

- Part 40: Master data representation

[125] ISO 29002, Exchange of characteristic data

[126] ISO 3534-2

[127] http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

[128] The JavaScript Object Notation (JSON) Data Interchange Format, RFC7159, March 2014, https://tools.ietf.org/html/rfc7159

[129] P21451-1-4, Standard for a Smart Transducer Interface for Sensors, Actuators, and Devices - eXtensible Messaging and Presence Protocol (XMPP) for Networked Device Communication, active project

[130] Department of Homeland Security, CUSTOMS AND BOARDER PROTECTION, Container Security Initiative (CSI), http://www.cbp.gov/border-security/ports-entry/cargo-security/csi/csi-brief

[131] C-TPAT (Customs Trade Partnership Against Terrorism), http://www.cbp.gov/border-security/ports-entry/cargo-security/c-tpat-customs-trade-partnership-against-terrorism

[132] Bonded Warehouse Manual for CBP Officers and Bonded Warehouse Proprietors, http://www.cbp.gov/document/guidance/bonded-warehouse-manual-cbp-officers-and-bonded-warehouse-proprietors

[133] Amendment to the Current Reporting Requirements for the Ultimate Consignee at the Time of Entry or Release, http://www.cbp.gov/border-security/ports-entry/cargo-security/cargo-control/ult-consignee

[134] EN ISO 19115-1:2014, Geographic information -- Metadata -- Part 1: Fundamentals

[135] Richard C. Murphy, “NIST Cyber Physical Systems Working Group Data Interop Contributions”,http://www.rickmurphy.org/data-interop.html

[136] ISO/IEC 16500-8:1999, Information technology -- Generic digital audio-visual systems -- Part 8: Management architecture and protocols

[137] Kahn, Robert E. and Lyons, Patrice A., "Representing Value as Digital Objects: A Discussion of Transferability and Anonymity". *Journal on Telecommunications & High Technology Law*, Vol. 5, Issue 1, 189 (2006).

[138] International Carrier Bonds for Non-Vessel Operating Common Carriers (NVOCCs), http://www.cbp.gov/border-security/ports-entry/cargo-security/cargo-control/carrier-bonds

[139] http://en.wikipedia.org/wiki/Semantic_Web_Stack

[140] Turnitsa, C.D. (2005). Extending the Levels of Conceptual Interoperability Model. Proceedings IEEE Summer Computer Simulation Conference, IEEE CS Press, see http://en.wikipedia.org/wiki/Conceptual_interoperability

[141] NISO, (2004)" Understanding metadata", Bethesda, MD: NISO Press, p1, http://www.niso.org/standards/resources/UnderstandingMetadata.pdf

[142] Models as a Basis for Ontologies, Ed Barkmeyer, NIST, Ontolog Forum, April, 2007. http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2007_04_12

[143] RFC3444 On the Difference between Information Models and Data Models, http://www.rfc-editor.org/rfc/rfc3444.txt

[144] http://www.dona.net

[145] http://en.wikipedia.org/wiki/Satisfiability

[146] ANSI C12.19-2008 American National Standard For Utility Industry End Device Data Tables, American National Standards Institute, Inc., February 24, 2009

[147] A Universally Unique IDentifier (UUID) URN Namespace, Leach, Mealling & Salz, Julu 2005, http://www.ietf.org/rfc/rfc4122.txt

[148] IEC 62541 Series-OPC Unified Architecture (OPC-UA), Version 1.01, Released,2009-02-09, http://www.opcfoundation.org/ua

[149] Hadoop, http://hadoop.apache.org/

[150] ISO/IEC 18004:2006 Information technology – Automatic identification and data capture techniques – QR code 2005 bar code symbology specification, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43655

[151] RFC 6282, Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, Hui & Thubert, September 2011, https://tools.ietf.org/html/rfc6282

[152] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), WC3 Recommendation 27 April 2007, http://www.w3.org/TR/soap12-part1/

[153] Architectural Styles and the Design of Network-based Software Architectures, Fielding, 2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[154] Trusted Computing Group, TNC Architecture for Interoperability, Revision 1.5 Rev 3, May 2012, http://www.trustedcomputinggroup.org/files/resource_files/2884F884-1A4B-B294-D001FAE2E17EA3EB/TNC_Architecture_v1_5_r3-1.pdf

[155] TNC IF-MAP Binding for SOAP, Specification version 2.2 rev 9, 26 Mar 2014, http://www.trustedcomputinggroup.org/files/static_page_files/FF3CB868-1A4B-B294-D093D8383D733B8A/TNC_IFMAP_v2_2r9.pdf

[156] OASIS Web Services Security (WSS), https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss-m

[157] https://rd-alliance.org/groups/data-type-registries-wg.html

[158] PRIVACY POLICY GUIDANCE MEMORANDUM, Memorandum Number: 2008-01, December 29, 2008, Department of Homeland Security, http://www.dhs.gov/xlibrary/assets/privacy/privacy_policyguide_2008-01.pdf

[159] FIPPs - http://www.dhs.gov/xlibrary/assets/privacy/privacy_policyguide_2008-01.pdf.

[160] Kent, William, updated by Steve Hoberman. "Data & Reality: A Timeless Perspective on Perceiving and Managing Information in Our Imprecise World." Westfield, NJ: Technics Publications, 2012. Print

[161] Data-Oriented Architecture: Loosely Coupling Systems into “Systems of Systems”, http://www.rtcmagazine.com/articles/view/100926

[162] Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI), http://www.oracle.com/technetwork/articles/javase/soa-142870.html

[163] Web Service Description Language (WSDL) 1.1, https://www.w3.org/TR/wsdl

[164] Architectural Styles and Design of Network-based Software Architectures, Dissertation University of California Irvine, Roy Thomas Fielding, 2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[165] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), https://www.w3.org/TR/soap12-part1/

[166] XEP-0060: Publish-Subscribe, http://xmpp.org/extensions/xep-0060.html

[167] Data Distribution Service (DDS), V1.4, OMG, April 2015, http://www.omg.org/spec/DDS/1.4/PDF

[168] John Bethencourt, Amit Sahai, Brent Waters, “Ciphertext-Policy Attribute-Based Encryption”, Proc 2007 IEEE Symposium on Security and Privacy, p.321-334 (May 2007).

[169] Justine Sherry, Chang Lan, Raluca Ada Popa and Sylvia Ratnasamy, “BlindBox: Deep Packet Inspection over Encrypted Traffic”, Proc ACM SIGCOMM 2015, pp. 213-226 (Aug 2015).

[170] Bonomi, F., Milito, R., Zhu, J., and Addepalli,S., "Fog Computing and its Role in the Internet of Things", Proc of ACM SIGCOMM 2012, Workshop on Mobile Cloud Computing (MCC 2012), pp. 13-16 (Aug 2012).

[171] "HVAC Vendor Confirms Link to Target Breach", Stephanie Mlot, PC Magazine (Feb 2015), http://www.pcmag.com/article2/0,2817,2430505,00.asp

[172] "UC Berkeley: 80,000 Staff, Students Compromised by Hack", Brittany A. Roston, SlashGear (Feb 2016). http://www.slashgear.com/uc-berkeley-80000-staff-students-compromised-by-hack-26429316/

[173] "Biggest-ever US data breach hits 100 million people with bank accounts", Jason Abbrusezze (Nov 2016). http://mashable.com/2015/11/10/bank-data-breach-100-million/#ccFPY1uW5Oqb

[174] "Healthcare Industry Tops 2015 Data Breach List", HIPPA Journal (Sept 2015). http://www.hipaajournal.com/healthcare-industry-tops-2015-data-breach-list-8108/

D.4 Timing

[175] References from **ITU-R Recommendation TF,686-3 (12/2013) Glossary and Definitions of Time and** Frequency Terms available from http://www.itu.int/rec/R-REC-TF.686-3-201312-I/en Note: this document contains references to additional glossary and definition material published by NIST, BIPM, IEC and the ISO.

[176] The time scales UTC and TAI and the International System of Units, SI, are defined and maintained by the International Bureau of Weights and Measures (Bureau International des Poids et Mesures, BIPM),. See http://www.bipm.org

[177] D.B. Sullivan, D.W. Allan, D.A. Howe, and F.L. Walls, "Characterization of Clocks and Oscillators," NIST Tech. Note 1337, June 1, 1999, available from: http://tf.boulder.nist.gov/general/pdf/868.pdf

[178] DRAFT Timing Framework for Cyber Physical Systems Technical Annex, Release 0.8, September 2015, NIST Cyber-Physical System Public Working Group, available at www.cpspwg.org

[179] Kopetz, Hermann. Real-time systems: design principles for distributed embedded applications. Springer Science & Business Media, 2011.

[180] H. Kopetz and G. Bauer. The time-triggered architecture. Proceedings of the IEEE, 91(1):112–126, 2003.

[181] Jasperneite, J.; Feld, J., "PROFINET: an integration platform for heterogeneous industrial communication systems," Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on , vol.1, no., pp.8 pp.,822, 19-22 Sept. 2005

[182] Timing Committee Telecommunications and Timing Group- Range Commanders Council, "IRIG Serial time code formats," September, 2004. [Online]. Available: http://www.irigb.com/pdf/wp-irig-200-04.pdf

[183] Kaplan, Elliott D., and Christopher J. Hegarty, eds. Understanding GPS: principles and applications. Artech house, 2005.

[184] IEEE Instrumentation and Measurement Society, "1588: IEEE standard for a precision clock synchronization protocol for networked measurement and control sytems" IEEE, Standar Specification, July 24, 2008

[185] K. Harris, "An application of IEEE 1588 to industrial automation," in Precision Clock Synchronization for Measurement, Control and Communication, 2008, ISPCS. IEEE International Symposium on. IEEE, 2008, pp 71-76

[186] M. Shepard, D. Fowley, R. Jackson, and D. King, "Implementation of IEEE Std-1588 on a Networked I/O Node, " in Proceedigns of the 2003 Workshop on IEEE-1588, NIST publication NISTIR 7070, Gaithersburg, MD, 2003.

[187] F. Steinhauser, C. Riesch, and M. Ridigier, "IEEE 1588 for time synchronization of devices in the electric power industry," in Precision Clock Synchronization for Measurement, Control and Communication, 2010, ISPCS. IEEE International Symposium on. IEEE, 2010, pp 1-6

[188] Giorgio C. Buttazzo: Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, Third Edition. Real-Time Systems Series 24, Springer 2011, ISBN 978-1-4614-0675-4, pp. 1-521

[189] R. Wilhelm, D. Grund: Computation takes time, but how much? Commun. ACM 57(2): 94-103 (2014)

[190] P. Axer, R. Ernst, H. Falk, A. Girault, D. Grund, N. Guan, B. Jonsson, P. Marwedel, J. Reineke, C. Rochange, M. Sebastian, R. von Hanxleden, R. Wilhelm, W. Yi: Building timing predictable embedded systems. ACM Trans. Embedded Comput. Syst. 13(4): 82 (2014)

[191] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D.B. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, P. Stenström: The worst-case execution-time problem - overview of methods and survey of tools. ACM Trans. Embedded Comput. Syst. 7(3) (2008)

[192] J. Rushby and W. Steiner, “TTA and PALS: Formally verified design patterns for distributed cyber-physical systems,” in 30th IEEE/AIAA Digital Avionics Systems Conference (DASC), Seattle, WA, 2011.

[193] ARINC, “ARINC 653 family of standards,” November, 2010. [Online]. Available: https://www.arinc.com/cf/store/

[194] Edward A. Lee: Computing needs time. Commun. ACM 52(5): 70-79 (2009)

[195] PHYTER, DP83640 Precision. "IEEE 1588 precision time protocol transceiver." (2008).

[196] Corbett, James C., et al. "Spanner: Google’s globally distributed database."ACM Transactions on Computer Systems (TOCS) 31.3 (2013): 8.

[197] B. Liskov, “Practical uses of synchronized clocks in distributed systems,” Distributed Computing, vol. 6, no. 4, pp. 211–219, 1993.

[198] Y. Zhao, E.A. Lee, and J. Liu. A programming model for time-synchronized distributed real-time systems. In Real-Time and Embedded Technology and Applications Symposium (RTAS), Bellevue, WA, USA, April 3-6 2007. IEEE.

[199] Broman, David, Patricia Derler, and John Eidson. "Temporal issues in cyber-physical systems." Journal of the Indian Institute of Science 93.3 (2013): 389-402.

[200] P. Derler, E. A. Lee, M. Zimmer, “Logically Synchronous Models of Distributed Systems with Explicit Timing Specifications”, 20th IMEKO TC4 International Symposium and 18th International Workshop on ADC Modelling and Testing. September 15-17, 2014.

[201] B. Keinhuis, E. Deprettere, P. Wolf, K. Vissers, “A Methodology to Design Programmable Embedded Systems. The Y-chart approach”, LNCS series vol. 2268, pg. 18-37, by Springer-Verlag © 2001.

[202] Open Networking Foundation, “OpenFlow Switch Specification,” October 14, 2013. [Online]Available from: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

[203] Paul Congdon, “Link Layer Discovery Protocol Overview (LLDP),” March 8, 2003. [Online] Available from: http://www.ieee802.org/1/files/public/docs2002/LLDP%20Overview.pdf

[204] PROFINET. [Online] presentation at: http://www.profibus.com/technology/profinet/

[205] IETF Network Working Group, “Simple Network Management Protocol (SNMP)”. [Online] available at: https://www.ietf.org/rfc/rfc1157.txt

[206] PCI-SIG Precision Time Measurement Revision 1.0a, [Online] available at: https://pcisig.com/sites/default/files/specification_documents/ECN_PTM_Revision1a_31_Mar_2013.pdf

[207] Intel® 64 and IA-32 Architectures Software Developer's Manual, [Online] available at: http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf

[208] CANopen. [Online] available from: http://www.can-cia.org/index.php?id=canopen

[209] Center for Hybrid and Embedded Software (CHESS), UC Berkeley, "PTIDES". [Online] see: http://chess.eecs.berkeley.edu/ptides/

[210] National Instruments, "LabVIEW System Design Software". [Online] see: http://www.ni.com/labview/

[211] Kopetz, H. "Real-Time Systems: Design Principles for Distributed Embedded Applications (Second Edition)", Springer, 2011.

[212] Dijkstra, E. "Self-Stabilizing Systems In Spite of Distributed Control," *Communications of ACM*, 17(11), 1974.

[213] Malekpour, Mahyar. "A Byzantine Fault-Tolerant Self-Stabilizing Protocol for Distributed Clock Synchronization," Proceedings of the 8th international conference on Stabilization, safety, and security of distributed systems

[214] Lamport, L., Shostak, R., and Pease, M. "The Byzantine General's Problem." *ACM Transactions on Programming Languages and Systems*, 4(3), July 1982, pp. 382-401.

[215] Mills, D. *Computer Network Time Synchronization: The Network Time Protocol on Earth and in Space, November 2010.*

[216] IEEE Instrumentation and Measurement Society, IEEE 1588-2008 IEEE Standard for Precision Clock Synchronization Protocol for Measurement and Control Systems, 24 July 2008.

[217] NTP: The Network Time Protocol. [Online] http://www.ntp.org/

[218] (1994). SPECIFICATION OF THE TRANSMITTED LORAN-C SIGNAL. U.S Department of Transportation.

[219] ATIS COAST Standards Body, document SYNC-2014-00052R000 from NIST, "CONTRIBUTION TO STANDARDS PROJECT — COAST-SYNC: WWVB for Assisted Timing." John Lowe; Marc Weiss. October 2013.

[220] A.J. Kerns, K.D. Wesson, and T.E. Humphreys, "A Blueprint for Civil GPS Navigation Message Authentication," IEEE/ION PLANS, Monterey, CA, May 2014. [Online] available from: http://radionavlab.ae.utexas.edu/images/stories/files/papers/nmaimpPLANS2014.pdf

[221] Johnson, G. S. (2007). An Evaluation of eLoran as a Backup to GPS. *Technologies for Homeland Security, 2007 IEEE Conference on* (pp. 95-100). Woburn, MA: IEEE.Satisfiability. (n.d.). Retrieved October, 2014, from http://en.wikipedia.org/wiki/Satisfiability

[222] A. Pearson and K. Shenoi. "A Case for Assisted Partial Timing Support Using Precision Timing Protocol Packet Synchronization for LTE-A," *IEEE Communications Magazine,* August 2014, pp. 136-143.

[223] T. Mizrahi, RFC 7384: Security Requirements of Time Protocols in Packet-Switched Networks. https://www.rfc-editor.org/rfc/rfc7384.txt

[224] T. Mizrahi, Time synchronization security using IPsec and MACsec, International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2011

[225] *D. Sibold, S. Roettger, K. Teichel*, "Network Time Security", October 2014. https://tools.ietf.org/html/draft-ietf-ntp-network-time-security-05

[226] *D. Sibold et al.*, "Protecting Network Time Security Messages with the Cryptographic Message Syntax (CMS)", October 2014, https://tools.ietf.org/html/draft-ietf-ntp-cms-for-nts-message-00

[227] *IEEE 1588 Working Group Website.* https://ieee-sa.centraldesktop.com/1588public/ 20 Nov. 2013.

[228] Caverly, R.J. "GPS Critical Infrastructure: Usage/Loss Impacts/Backups/Mitigation", April 27, 2011. http://www.swpc.noaa.gov/sww/sww11/SWW_2011_Presentations/Wed_830/GPS-PNTTimingStudy-SpaceWeather4-27.pptx

[229] Kappenman, J. "Geomagnetic Storms and Their Impacts on the U.S. Power Grid." January 2010. http://web.ornl.gov/sci/ees/etsd/pes/pubs/ferc_Meta-R-319.pdf

[230] The MITRE Corporation. "Detection, Localization, and Mitigation Technologies for Global Positioning System (GPS) Jamming and Spoofing (Final)". *Redacted for Public Release.* February 2014.

[231] Jaldehag, K., Ebenhag, S., Hedekvist, P., Rieck, C., and Lothberg, P. "Time and Frequency Transfer Using Asynchronous Fiber Optical Networks: Progress Report," *Proceedings of 41st Annual Precise Time and Time Interval (PTTI) Meeting*, 2009.

[232] R. Cohen, PTP Security Tutorial, International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS), 2007

[233] A. Treytl, G. Gaderer, B. Hirschler, Traps and pitfalls in secure clock synchronization, International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS), 2007

[234] A. Treytl, B. Hirschler, Validation and Verification of IEEE 1588 Annex K, International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS), 2011

[235] C. Önal and H. Kirrmann, Security improvements for IEEE 1588 Annex K, International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS), 2012

[236] S. Röttger, Analysis of the NTP Autokey Extension (in German), University of Braunschweig and Physikalisch-Technische Bundesanstalt Braunschweig, 2011

[237] A. Treytl, B. Hirschler, Security Flaws and Workarounds for IEEE 1588 (Transparent) Clocks, International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS), 2009

[238] A. Treytl, B. Hirschler, Practical Application of 1588 Security, International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS), 2008

[239] RFC 6066. Internet Engineering Task Force (IETF) Transport Layer Security (TLS) Extensions: Extension Definitions. January 2011. https://tools.ietf.org/html/rfc6066

[240] J. Tournier, O. Goerlitz, Strategies to Secure the IEEE 1588 Protocol in Digital Substation Automation, Fourth International Conference on Critical Infrastructures (CRIS), 2009

[241] Daniel P. Shepard, D.P.; Humphreys, T.E., Fansler, A.A. "Going Up Against Time: The Power Grid's Vulnerability to GPS Spoofing Attacks." GPS World, August 2012.

[242] D. P. Shepard, J. A. Bhatti, T. E. Humphreys, Evaluation of Smart Grid and Civilian UAV Vulnerability to GPS Spoofing Attacks, slide no. 11, September 21, 2012.

[243] "Time Anomaly Detection Applique", 2013. http://www.mitre.org/research/technology-transfer/technology-licensing/time-anomaly-detection-appliqu%C3%A9-tada

[244] Langley, R.B. “Innovation: GNSS Spoofing Detection.” GPS World. http://gpsworld.com/innovation-gnss-spoofing-detection-correlating-carrier-phase-with-rapid-antenna-motion/

[245] Pearson, T. and Shenoi, K. “A Case for Assisted Partial Timing Support Using Precision Timing Protocol Packet Synchronization for LTE-A.” *IEEE Communications Magazine,* 52 (8), August 2014, pp. 135-143.

[246] Amelot, J., Li-Baboud, Y., Vasseur, C., Fletcher, J., Anand, D., and Moyne, J. “An IEEE 1588 Performance Testing Dashboard for Power Industry Requirements,” *Proceedings of International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS),* pp. 132-137, 12-16 Sept. 2011.

[247] Crain, A. and Sistrunk, C. Advisory (ICSA-13-210-01). https://ics-cert.us-cert.gov/advisories/ICSA-13-219-01

[248] E. O. Schweitzer, E.O; Guzmán, A. “Real-Time Synchrophasor Applications for Wide-Area Protection, Control, and Monitoring.” © 2009 by Schweitzer Engineering Laboratories, Inc.

[249] http://www.bpa.gov/news/newsroom/Pages/Synchrophasor-success-lands-B...

[250] Goldstein, A. Email to the CPS_Sync list dated 16 SEPT 2014.

[251] http://www.microsemi.com/products/timing-synchronization-systems/time-frequency-distribution/gps-instruments/xli-saasm ; also Symmetricom’s GPS Disciplined Master Timing Reference (ATS 6501B). Warriner, J., private communication on 21 February 2014.

[252] *I. Fernández Hernández, “*Design Drivers, Solutions and Robustness Assessment of Navigation Message Authentication for the Galileo Open Service,” ION GNSS+ 2014, Tampa, FL, September 2014.

[253] *J.T. Curran, M. Paonni, J. Bishop, “*Securing GNSS: An End-to-end Feasibility Analysis for the Galileo Open-service,” ION GNSS+ 2014, Tampa, FL, September 2014.

[254] *Factsheet: National Risk Estimate: Risks to U.S. Critical Infrastructure from Global Position System Disruptions*, June 2013. http://www.gps.gov/news/2013/06/2013-06-NRE-fact-sheet.pdf

[255] ITU-R Recommendation TF,686-3 (12/2013) Glossary and Definitions of Time and Frequency Terms available from http://www.itu.int/rec/R-REC-TF.686-3-201312-I/en Note: this document contains references to additional glossary and definition material published by NIST, BIPM, IEC and the ISO.

[256] All ITU-T published recommendations can be downloaded from: http://www.itu.int/rec/T-REC-G/e

The following lists ITU-T Published Recommendations associated with timing in telecom networks.

[257] ITU-T Published Recommendations (PDH/SDH)

- ITU-T Recommendation G.803, Architecture of transport networks based on the synchronous digital hierarchy (SDH).
- ITU T Recommendation G.810, Definitions and terminology for synchronization networks.
- ITU T Recommendation G.811, Timing characteristics of primary reference clocks.
- ITU T Recommendation G.812, Timing requirements of slave clocks suitable for use as node clocks in synchronization networks.
- ITU T Recommendation G.813, Timing characteristics of SDH equipment slave clocks (SEC).
- ITU-T Recommendation G.823, The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy
- ITU-T Recommendation G.824, The control of jitter and wander within digital networks which are based on the 1544 kbit/s hierarchy
- Recommendation ITU-T G.825, The control of jitter and wander within digital networks which are based on the synchronous digital hierarchy (SDH)

[258] ITU-T Published Recommendations (Packet Sync - Frequency)

- ITU T Recommendation G.8261, Timing and synchronization aspects in packet networks.
- ITU T Recommendation G.8262, Timing characteristics of Synchronous Ethernet Equipment slave clock (EEC).
- ITU T Recommendation G.8264, Distribution of timing through packet networks
- Recommendation ITU-T G.8261.1, Packet Delay Variation Network Limits applicable to Packet Based Methods (Frequency Synchronization).
- Recommendation ITU-T G.8263, Timing Characteristics of Packet based Equipment Clocks (PEC) and Packet based Service Clocks (PSC)
- ITU-T Recommendation G.8265), Architecture and requirements for packet based frequency delivery
- ITU-T Recommendation G.8265.1, Precision time protocol telecom profile for frequency sync
- ITU-T Recommendation G.8260, Definitions and terminology for synchronization in packet networks

[259] ITU-T Consented Recommendations (Packet Sync – Phase/Time)

- ITU T Recommendation G.8271, Time and phase synchronization aspects of packet

networks

- ITU T Recommendation G.8272, Timing characteristics of Primary reference time clock
- ITU T Recommendation G.8271.1 , Network limits
- ITU T Recommendation G.8272, Primary Reference Timing Clock (PRTC) specification
- ITU T Recommendation G.8273, Clock General Requirements
- ITU T Recommendation G.8273.2 , Telecom Boundary Clock specification
- ITU T Recommendation G.8275 , Architecture for time transport
- ITU T Recommendation G.8275.1 , IEEE-1588 profile for time with full support from the network

D.5 Use Case Analysis

[260] Dictionary.com

[261] Customer Communications Architecture Development: Metrics for Standards and Product Assessment. EPRI, Palo Alto, CA 94303, 20-Dec-2011, Product 1021945

Appendix E. Definitions and Acronyms

The following definitions and acronyms are presented as a ready reference to the intended meaning of their use in the text of this document. It is recognized that within various technical domains, many of these terms and acronyms have multiple meanings. The intent is to provide clarity for the interpretation of this framework and not to make a definitive statement about the "universal" definition of the terms and acronyms. In some cases, canonical references were not identified and the "source" column lists "this document" as the context for the definition.

E.1 Selected terms used in this document are defined below.

Term	Definition	Source
access control	A means to ensure that access to assets is authorized and restricted based on business and security requirements Note: Access control requires both authentication and authorization	[17]
accuracy	Closeness of the agreement between the result of a measurement and the true value of the measurand.	ITU-R Rec. TF.686
actors	A person or system component who interacts with the system as a whole and who provides stimulus which invoke actions.	[136]
actuator	A device which conveys digital information to effect a change of some property of a physical entity.	[9]++
ageing	The systematic change in frequency with time due to internal changes in the oscillator. NOTE 1 – It is the frequency change with time when factors external to the oscillator (environment, power supply, etc.) are kept constant.	ITU-R Rec. TF.686
architecture view	An 'architecture view' consists of 'work product expressing the architecture of a system from the perspective of specific system concerns'.	[2]
architecture viewpoint	An 'architecture viewpoint' consists of work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns'.	[2]
aspect	Conceptually equivalent concerns, or major categories of concerns. Sometimes called "cross-cutting" concerns.	This document
assurance	The level of confidence that a CPS is free from vulnerabilities, either intentionally designed into it or accidentally inserted during its lifecycle, and that the CPS functions in the intended manner.	This document
assurance level	The Evaluation Assurance Level (EAL1 through EAL7) of an IT product or system is a numerical grade assigned following the completion of a Common Criteria security evaluation, an international standard in effect since 1999. The increasing assurance levels reflect added assurance requirements that must be met to achieve Common Criteria certification. The intent of the higher levels is to provide	This document

Term	Definition	Source
	higher confidence that the system's principal security features are reliably implemented. The EAL level does not measure the security of the system itself, it simply states at what level the system was tested.	
assured time	Time derived from a known good time reference in a secure manner.	This document
attribute	A characteristic or property of an entity that can be used to describe its state, appearance, or other aspects.	[15]
authenticated identity	Identity information for an entity created to record the result of identity authentication.	[15]
authentication	Provision of assurance that a claimed characteristic of an entity is correct.	[17]
authorization	Granting of rights, which includes the granting of access based on access rights.	[12]
automatic	Working by itself with little or no direct human control.	[21]
automation	The use or introduction of automatic equipment in a manufacturing or other process or facility. Note: Automation emphasizes efficiency, productivity, quality, and reliability, focusing on systems that operate without direct control, often in structured environments over extended periods, and on the explicit structuring of such environments.	[21]
calibration	The process of identifying and measuring offsets between the indicated value and the value of a reference standard used as the test object to some determined level of uncertainty. NOTE 1 – In many cases, e.g., in a frequency generator, the calibration is related to the stability of the device and therefore its result is a function of time and of the measurement averaging time.	ITU-R Rec. TF.686
certificate	A set of data that uniquely identifies an entity, contains the entity's public key and possibly other information, and is digitally signed by a trusted party, thereby binding the public key to the entity. Additional information in the certificate could specify how the key is used and its cryptoperiod.	SP 800-21
certificate revocation list (CRL)	A list of revoked public key certificates created and digitally signed by a Certification Authority	SP 800-63; FIPS 201
checksum	Value computed on data to detect error or manipulation	CNSSI-4009
choreography	Type of composition whose elements interact in a non-directed fashion with each autonomous part knowing and following an observable predefined pattern of behavior for the entire (global) composition	[18]

Term	Definition	Source
clock	A device that generates periodic signals for synchronization. Note: Other definitions are provided in different references that are tailored to particular applications. Suitable references include ITU-T Rec. G.810, ITU-R Rec. TF.686 and IEEE Std. 1377-1997.	IEEE Std. 1377-1997
collaboration	Type of composition whose elements interact in a non-directed fashion, each according to their own plans and purposes without a predefined pattern of behavior.	[18]
component	Modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces.	[13]
composition	Result of assembling a collection of elements for a particular purpose.	[18]
concern	Category of analysis by which a CPS can be considered	Framework
controller	A user that interacts across a network to affect a physical entity.	[9] ++
CPS architecture	A concrete realization of a reference CPS architecture designed to satisfy use-case-specific constraints.	This document
CPS Framework	Abstract framework and analysis methodology for understanding and deriving application-domain-specific CPS architectures. Activities and outputs to support engineering of CPS.	This document
CPS network manager	A work-station or CPS node connected to a CPS domain that manages and monitors the state and configuration of all CPS nodes in one or more CPS domains.	This document
CPS time domain	A CPS time domain is a logical group of CPS nodes and bridges which form a network with their own timing master.	This document
credential (electronic)	Digital documents used in authentication that bind an identity or an attribute to a subscriber's token	CNSSI-4009
cross-cutting concern	See aspect	This document
cryptographic (encryption) certificate	A certificate containing a public key that is used to encrypt electronic messages, files, documents, or data transmissions, or to establish or exchange a session key for these same purposes	SP 800-32

Term	Definition	Source
cryptographic hash (function)	A function that maps a bit string of arbitrary length to a fixed length bit string. Approved hash functions satisfy the following properties: 1) (One-way) It is computationally infeasible to find any input which maps to any pre-specified output, and 2) (Collision resistant) It is computationally infeasible to find any two distinct inputs that map to the same output.	SP 800-21
cryptographic key	A value used to control cryptographic operations, such as decryption, encryption, signature generation, or signature verification	SP 800-63
cyber-physical device	A device that has an element of computation and interacts with the physical world through sensing and actuation.	This document
cyclical redundancy check (CRC)	A method to ensure data has not been altered after being sent through a communication channel	SP 800-72
data	Re-interpretable representation of information in a formalized manner suitable for communication, interpretation, or processing. NOTE Data can be processed by humans or by automatic means.	[122]
data accuracy	Closeness of agreement between a property value and the true value. NOTE 1: In practice, the accepted reference value is substituted for the true value.	[122]
device	A physical entity embedded inside, or attached to, another physical entity in its vicinity, with capabilities to convey digital information from or to that physical entity.	[22]
device endpoint	An endpoint that enables access to a device and thus to the related physical entity.	[22]
digital entity	An entity represented as, or converted to, a machine-independent data structure consisting of one or more elements in digital form that can be parsed by different information systems; and the essential fixed attribute of a digital entity is its associated unique persistent identifier, which can be resolved to current state information about the digital entity, including its location(s), access controls, and validation, by submitting a resolution request to the resolution system.	This document
digital signature	An asymmetric key operation where the private key is used to digitally sign data and the public key is used to verify the signature. Digital signatures provide authenticity protection, integrity protection, and non-repudiation	SP 800-63
element	Unit that is indivisible at a given level of abstraction and has a clearly defined boundary.	[18]

Term	Definition	Source
	Note: An element can be any type of entity	
endpoint	One of two components that either implements and exposes an interface to other components or uses the interface of another component.	[16]
endpoint address	Data element designating the originating source or destination of data being transmitted.	[14]
entity	Item inside or outside an information and communication technology system, such as a person, an organization, a device, a subsystem, or a group of such items that has recognizably distinct existence	[15]
epoch	Epoch signifies the beginning of an era (or event) or the reference date of a system of measurements.	ITU-R Rec. TF.686
facet	Facets are perspectives on CPS that each express a distinct set of well-defined processes, methods and tools to support the CPS development process and for expressing the architecture of a system. The Framework identified facets are conceptualization, realization and assurance.	This document
formal syntax	Specification of the valid sentences of a formal language using a formal grammar. NOTE 1 A formal language is computer-interpretable. NOTE 2 Formal grammars are usually Chomsky context-free grammars. NOTE 3 Variants of Backus-Naur Form (BNF) such as Augmented Backus-Naur Form (ABNF) and Wirth Syntax Notation (WSN) are often used to specify the syntax of computer programming languages and data languages. EXAMPLE 1 An XML document type definition (DTD) is a formal syntax. EXAMPLE 2 ISO 10303-21, contains a formal syntax in WSN for ISO 10303 physical files.	This document

Term	Definition	Source
fractional frequency deviation	The difference between the actual frequency of a signal and a specified nominal frequency, divided by the nominal frequency.	ITU-T Rec. G.810
frequency	If T is the period of a repetitive phenomenon, then the frequency f = 1/T. In SI units the period is expressed in seconds, and the frequency is expressed in hertz (Hz).	ITU-R Rec. TF.686
frequency drift	A systematic undesired change in frequency of an oscillator over time. Drift is due to ageing plus changes in the environment and other factors external to the oscillator. See “ageing”.	ITU-R Rec. TF.686
frequency instability	The spontaneous and/or environmentally caused frequency change of a signal within a given time interval. NOTE 1 – Generally, there is a distinction between systematic effects such as frequency drift and stochastic frequency fluctuations. Special variances have been developed for the characterization of these fluctuations. Systematic instabilities may be caused by radiation, pressure, temperature, and humidity. Random or stochastic instabilities are typically characterized in the time domain or frequency domain. They are typically dependent on the measurement system bandwidth or on the sample time or integration time. See Recommendation ITU-R TF.538.	ITU-R Rec. TF.686
frequency offset **(see also fractional frequency deviation)**	The frequency difference between the realized value and the reference frequency value. NOTE 1 – The reference frequency may or may not be the nominal frequency value.	ITU-R Rec. TF.686
frequency standard	An accurate stable oscillator generating a fundamental frequency used in calibration and/or reference applications. See Recommendation ITU-T G.810.	ITU-R Rec. TF.686
functional requirement	Functional requirements define specific behavior (functions) or particular results of a system and its components, what the system is supposed to accomplish.	This document
gateway	A forwarding component, enabling various networks to be connected.	[9] ++

Term	Definition	Source
hash	Value computed on data to detect error or manipulation. See Checksum.	CNSSI-4009
identification	A process of recognizing an entity in a particular identity domain as distinct from other entities.	[15]
identifier	Identity information that unambiguously distinguishes one entity from another one in a given identity domain.	[15]
identity	The characteristics determining who or what a person or thing is.	[21]
identity authentication	Formalized process of identity verification that, if successful, results in an authenticated identity for an entity.	[15]
identity domain	An environment where an entity can use a set of attributes for identification and other purposes.	[15]
identity information	A set of values of attributes optionally with any associated metadata in an identity. Note: In an information and communication technology system an identity is present as identity information.	[15]
identity management	Processes and policies involved in managing the lifecycle and value, type and optional metadata of attributes in identities known in a particular identity domain.	[15]
identity verification	A process to determine that presented identity information associated with a particular entity is applicable for the entity to be recognized in a particular identity domain at some point in time.	[15]
industrial internet	An Internet of things, machines, computers and people, enabling intelligent industrial operations using advanced data analytics for transformational business outcomes.	[22]
information	Knowledge concerning objects, such as facts, events, things, processes or ideas, including concepts, that within a certain context has a particular meaning	[120]

Term	Definition	Source
infrastructure services	Specific services that are essential for a CPS/Internet of Things (IoT) implementation to work properly. Such services provide support for essential features of the IoT.	[9]
interface	Named set of operations that characterize the behavior of an entity.	[9]
Internet	A global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.	[21]
IP endpoint	An endpoint which has an IP address.	[22]
jitter	The short-term phase variations of the significant instants of a timing signal from their ideal position in time (where short-term implies here that these variations are of frequency greater than or equal to 10 Hz). See also “wander”.	ITU-R Rec. TF.686
key data storage (key escrow)	A deposit of the private key of a subscriber and other pertinent information pursuant to an escrow agreement or similar contract binding upon the subscriber, the terms of which require one or more agents to hold the subscriber's private key for the benefit of the subscriber, an employer, or other party, upon provisions set forth in the agreement.	SP 800-32
latency	The latency of a device or process is the time delay introduced by the device or process.	This document
master data	Data held by an organization that describes the entities that are both independent and fundamental for that organization, and that it needs to reference in order to perform its transactions.	[122]

Term	Definition	Source
network synchronization	A generic concept that depicts the way of distributing a common time and/or frequency to all elements in a network.	ITU-T Rec. G.810
network time protocol (NTP)	The network time protocol (NTP) is used to synchronize the time of a computer client or server to another server or reference time source, such as a terrestrial or satellite broadcast service or modem. NTP provides distributed time accuracies on the order of one millisecond on local area networks (LANs) and tens of milliseconds on wide area networks (WANs). NTP is widely used over the Internet to synchronize network devices to national time references. See www.ntp.org. See also IETF documents (e.g., RFC 5905).	ITU-R Rec. TF.686
non-functional requirement	Non-functional requirements specify criteria useful to evaluate the qualities, goals or operations of a system, rather than specific behaviors or functions of a system.	This document
observer	A user that interacts across a network to monitor a physical entity.	[9] ++
orchestration	The type of composition where one particular element is used by the composition to oversee and direct the other elements. Note: the element that directs an orchestration is not part of the orchestration.	[18]
oscillator	An electronic device producing a repetitive electronic signal, usually a sine wave or a square wave.	ITU-R Rec. TF.686
password	A secret that a Claimant memorizes and uses to authenticate his or her identity. Passwords are typically character strings	SP 800-63

Term	Definition	Source
phase coherence	Phase coherence exists if two periodic signals of frequency M and N resume the same phase difference after M cycles of the first and N cycles of the second, where M/N is a rational number, obtained through multiplication and/or division from the same fundamental frequency.	ITU-R Rec. TF.686
phase synchronization	The term phase synchronization implies that all associated nodes have access to reference timing signals whose significant events occur at the same instant (within the relevant phase accuracy requirement). In other words, the term phase synchronization refers to the process of aligning clocks with respect to phase (phase alignment). NOTE 1 – Phase synchronization includes compensation for delay between the (common) source and the associated nodes. NOTE 2 – This term might also include the notion of frame timing (that is, the point in time when the timeslot of an outgoing frame is to be generated). NOTE 3 – The concept of phase synchronization (phase alignment) should not be confused with the concept of phase-locking where a fixed phase offset is allowed to be arbitrary and unknown. Phase alignment implies that this phase offset is nominally zero. Two signals which are phase-locked are implicitly frequency synchronized. Phase-alignment and phase-lock both imply that the time error between any pair of associated nodes is bounded	ITU-T Rec. G.8260
physical entity	An entity that is the subject of monitoring and control actions.	[9] ++
PID loops	Proportional, integrative, derivative loop used in automation.	SP 800-82
policy	A course or principle of action adopted or proposed by an organization or individual.	[21]
precision time protocol (PTP)	A time protocol originally designed for use in instrument LANs now finding its way into WAN and packet based Ethernet network applications. PTP performance can exceed NTP by several orders of magnitude depending on the network environment. See IEEE 1588.	ITU-R Rec. TF.686
pre-shared key (symmetric key)	A cryptographic key that is used to perform both the cryptographic operation and its inverse, for example to encrypt and decrypt, or create a message authentication code and to verify the code.	SP 800-63; CNSSI-4009

Term	Definition	Source
reference timing signal	A timing signal of specified performance that can be used as a timing source for a slave clock.	ITU-T Rec. G.810
repeatability	Closeness of agreement between the results of successive measurements of the same measurand carried out under the same conditions as follows: • with respect to a single device when specified parameters are independently adjusted to a stated set of conditions of use, it is the standard deviation of the values produced by this device. It could also be termed “resettability”; • with respect to a single device put into operation repeatedly without readjustment, it is the standard deviation of the values produced by this device; • with respect to a set of independent devices of the same design, it is the standard deviation of the values produced by these devices used under the same conditions.	ITU-R Rec. TF.686
reproducibility	With respect to a set of independent devices of the same design, it is the ability of these devices to produce the same value. With respect to a single device, put into operation repeatedly, it is the ability to produce the same value without adjustments. NOTE 1 – The standard deviation of the values produced by the device(s) under test is the usual measure of reproducibility.	ITU-R Rec. TF.686
satisfiability	In mathematical logic, a formula is satisfiable if it is possible to find an interpretation that makes the formula true.	[145]

Term	Definition	Source
second	The SI unit of time, one of the seven SI base units. The second is equal to the duration of 9 192 631 770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium-133 atom. Note: The symbol for second, the SI unit of time, is s.	found in IEEE Std 270-2006 (Revision of IEEE Std 270-1966); IEEE Standard Definitions for Selected Quantities, Units, and Related ... |
sensor	A sensor is a special device that perceives certain characteristics of the real world and transfers them into a digital representation.	[9]
service	A distinct part of the functionality that is provided by an entity through interfaces.	[20]
signature	See digital signature	SP 800-63
stability	Property of a measuring instrument or standard, whereby its metrological properties remain constant in time.	ITU-R Rec. TF.686
subsystem	A discrete part of a system that groups some functionality that is part of the whole.	
syntonization	The relative adjustment of two or more frequency sources with the purpose of cancelling their frequency differences but not necessarily their phase difference.	ITU-R Rec. TF.686
system	A system is a composite set of logical components that together satisfy a concrete set of Use Cases.	This document
system function	What the system does. Formalized requirements.	This document

Term	Definition	Source
system of systems	Systems of systems exist when there is a presence of a majority of the following five characteristics: operational and managerial independence, geographic distribution, emergent behavior, and evolutionary development.	[121]

Term	Definition	Source
TAI : international atomic time	The timescale established and maintained by the BIPM on the basis of data from atomic clocks operating in a number of establishments around the world. Its epoch was set so that TAI was in approximate agreement with UT1 on 1 January 1958. The rate of TAI is explicitly related to the definition of the SI second as measured on the geoid. See “second”, “universal time”, “UT1” and SI Brochure.	ITU-R Rec. TF.686
temporal determinism	Property of a device or process whereby the latency introduced is known a priori.	This document
thing	Generally speaking, any physical object. In the term ‘Internet of Things’ however, it denotes the same concept as a physical entity.	[9]
time awareness	The extent to which a device or system has an appropriate ability to sense and response to timing signals and information. Also the extent to which a model can appropriately use time accurately for design, including time semantics, visual design, and time correctness once applied to operational systems.	This document
time interval	The duration between two instants read on the same timescale.	ITU-R Rec. TF.686
time scale (timescale; time-scale)	A system of unambiguous ordering of events. NOTE – This could be a succession of equal time intervals, with accurate references of the limits of these time intervals, which follow each other without any interruption since a well-defined origin. A time scale allows to date any event. For example, calendars are time scales. A frequency signal is not a time scale (every period is not marked and dated). For this reason "UTC frequency" must be used instead of "UTC".	ITU-T Rec. G.810
time stamp (timestamp; time-stamp)	An unambiguous time code value registered to a particular event using a specified clock.	ITU-R Rec. TF.686
time standard	A device used for the realization of the time unit. A continuously operating device used for the realization of a timescale in accordance with the definition of the second and with an appropriately chosen origin.	ITU-R Rec. TF.686

Term	Definition	Source
time synchronization:	Time synchronization is the distribution of a time reference to the real-time clocks of a telecommunication network. All the associated nodes have access to information about time (in other words, each period of the reference timing signal is marked and dated) and share a common timescale and related epoch (within the relevant time accuracy requirement. Examples of timescales are: • UTC • TAI • UTC + offset (e.g., local time) • GPS • PTP • local arbitrary time Note that distributing time synchronization is one way of achieving phase synchronization	ITU-T Rec. G.8260
timescales in synchronization	Two timescales are in synchronization when they, within the uncertainties inherent in each, assign the same date to an event and have the same timescale unit. NOTE 1 – If the timescales are produced in spatially separated locations, the propagation time of transmitted time signals and relativistic effects are to be taken into account.	ITU-R Rec. TF.686
timing	A general term for the field or discipline, including time and frequency sources, signals, measurement methods, timestamp methods, specification methods, and metrics.	This document
timing signal	A nominally periodic signal, generated by a clock, used to control the timing of operations in digital equipment and networks. Due to unavoidable disturbances, such as oscillator phase fluctuations, actual timing signals are pseudo-periodic ones, i.e., time intervals between successive equal phase instants show slight variations.	ITU-T Rec. G.810

Term	Definition	Source
traceability	The property of a result of a measurement whereby it can be related to appropriate standards, generally international or national standards, through an unbroken chain of comparisons. (ISO/IEC 17025:2005). Ability to compare a calibration device to a standard of even higher accuracy. That standard is compared to another, until eventually a comparison is made to a national standards laboratory. This process is referred to as a chain of traceability.	found in IEEE Std 1159-1995; IEEE Recommended Practice for Monitoring Electric Power Quality; also ITU-R Rec. TF.686
universal time (UT)	Universal time is a measure of time that conforms, within a close approximation, to the mean diurnal motion of the sun as observed on the prime meridian. UT is formally defined by a mathematical formula as a function of Greenwich mean sidereal time. Thus UT is determined from observations of the diurnal motions of the stars. The timescale determined directly from such observations is designated UT0; it is slightly dependent on the place of observation See Recommendation ITU-R TF.460. UT0: UT0 is a direct measure of universal time as observed at a given point on the Earth's surface. In practice, the observer's meridian (position on Earth) varies slightly because of polar motion, and so observers at different locations will measure different values of UT0. Other forms of universal time, UT1 and UT2, apply corrections to UT0 in order to establish more uniform timescales. See "universal time", "UT1" and "UT2" and Recommendation ITU-R TF.460. UT1: UT1 is a form of universal time that accounts for polar motion and is proportional to the rotation of the Earth in space. See "universal time" and Recommendation ITU-R TF.460. UT2: UT2 is a form of universal time that accounts both for polar motion and is further corrected empirically for annual and semi-annual variations in the rotation rate of the Earth to provide a more uniform timescale. The seasonal variations are primarily caused by meteorological effects. See "universal time" and Recommendation ITU-R TF.460. NOTE 1 – The UT2 timescale is no longer determined in practice.	ITU-R Rec. TF.686

Term	Definition	Source
user	An entity that is interested in interacting with a particular physical entity.	[9] ++
user endpoint	An endpoint used by a user to interact.	[22] proposed
UTC : coordinated universal time	The time scale, maintained by the Bureau International des Poids et Mesures (BIPM) and the International Earth Rotation Service (IERS), which forms the basis of a coordinated dissemination of standard frequencies and time signals. See Recommendation ITU R TF.460. It corresponds exactly in rate with TAI, but differs from it by an integer number of seconds. The UTC scale is adjusted by the insertion or deletion of seconds (positive or negative leap seconds) to ensure approximate agreement with UT1. See “universal time” and Recommendation ITU R TF.460.	ITU-T Rec. G.810 and ITU-R Rec. TF.686
virtual entity	Computational or data element representing a physical entity.	[9]
wander	The long-term phase variations of the significant instants of a timing signal from their ideal position in time (where long-term implies here that these variations are of frequency less than 10 Hz). See “jitter”. Note: there is work in ITU-T SG15/Q13 to address wander/jitter associated with time signals such as 1PPS where the 10Hz breakpoint is not meaningful.	ITU-R Rec. TF.686

E.2 Selected acronyms used in this document are defined below.

Acronym	Expansion
3D	Three dimensional
6LoWPAN	IPv6 over low-power personal area networks
ACM	Association for Computing Machinery
AES	Advanced Encryption Standard
AIAA	American Institute of Aeronautics and Astronautics
ANSI	American National Standards Institute
API	Application programming interface
APTS	Assisted Partial Timing Support
ARINC	Aeronautical Radio, Incorporated

Acronym	Expansion
ASIC	Application-specific integrated circuit
ATIS	Alliance for Telecommunications Industry Solutions
BIPM	Bureau International des Poids et Mesures
C-TPAT	Customs Trade Partnership Against Terrorism
CAD	Computer-aided design
CBP	Customs and Border Protection
CHESS	Center for Hybrid and Embedded Software
CMS	Cryptographic Message Syntax
CNM	CPS Network Manager
COAST	Copper/Optical Access, Synchronization, and Transport Committee
CPS PWG	Cyber-Physical Systems Public Working Group
CRC	Cyclic redundancy check
CRIS	Critical Infrastructures
CRL	Certificate Revocation List
CRM	Customer relationship management
CSI	Container Security Initiative
CSRA	Cybersecurity Research Alliance
DIS	Draft International Standard
DMV	Department of Motor Vehicles
DNS	Domain Name System
DO	Digital Object
DoS	Denial of service
EEC	Synchronous Ethernet equipment slave clock
EMI	Electromagnetic interference
EPRI	Electric Power Research Institute
ERM	Enterprise resource management
EU	European Union
FDIS	Final Draft International Standard
FIPP	Fair Information Practice Principles
FPGA	Field-programmable gate array
GNSS	Global navigation satellite system
GPS	Global positioning system
GRC	Governance, Risk, and Compliance
GUI	Graphical user interface
HIPAA	Health Insurance Portability and Accountability Act

Acronym	Expansion
HITECH	Health Information Technology for Economic and Clinical Health Act
HSPD-12	Homeland Security Presidential Directive 12
HTTPS	Hypertext Transfer Protocol over TLS
HVAC	Heating, ventilating, and air conditioning
HW	Hardware
I/O	Input/output
ICNRG	Information Centric Networking
ICS	Industrial control systems
IdP	Identity provider
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IF-Map	Interface for Metadata Access Points
IFF	Identification Friend or Foe
IHMC	Florida Institute for Human and Machine Cognition
IIC	Industrial Internet Consortium
IIOT	Industrial Internet of Things
IJSWIS	International Journal on Semantic Web and Information Systems
IoT	Internet of Things
IoT ARM	Internet of Things Architectural Reference Model
IoT-A	Internet of Things – Architecture
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv6	Internet Protocol version 6
IRIG-B	Inter-Range Instrumentation Group timecode B
ISA	Instrumentation, Systems, and Automation Society
ISO	Independent Service Operator
ISO	International Organization for Standardization
ISPCS	International IEEE Symposium on Precision Clock Synchronization for Measurement, Control, and Communication
IT	Information technology
ITU	International Telecommunication Union
ITU-R	International Telecommuncation Union – Radiocommunication Sector
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
JDL	Joint Director of Laboratories

Acronym	Expansion
KPI	Key performance indicator
LLDP	Link Layer Discovery Protocol
LNCS	Lecture Notes in Computer Science
LSB	Least-significant-bit
LTE-A	Long Term Evolution Advanced
M2M	Machine-to-machine
MAC	Media Access Control
MACsec	Media Access Control Security
MD5	Message Digest
MDR	Metadata Registries
MitM	Man in the middle
MOM	Manufacturing operations management
MTBF	Mean time between failures
NFPA	National Fire Protection Association
NFV	Network Function Virtualization
NILM	Non-intrusive load monitoring
NIPP	National Infrastructure Protection Plan
NISO	National Information Standards Organization
NIST	National Institute of Standards and Technology
NITRD	Networking and Information Technology Research and Development
NMA	Navigation message authentication
NSTAC	National Security Telecommunications Advisory Committee
NTP	Network Time Protocol
NVOCC	Non-Vessel Operating Common Carrier
OED	Oxford English Dictionary
OEM	Original equipment manufacturer
OMG	Object Management Group
OPC UA	OPC Unified Architecture
OSE	Open System Environment
OT	Operational technology
OWL	Web Ontology Language
PALS	Physically-Asynchronous Logically-Synchronous
PDH	Plesiochronous digital hierarchy
PDV	Packet delay variation
PEC	Packet-based equipment clock

Acronym	Expansion
PID	Persistent identifier
PII	Personally identifiable information
PKI	Public key infrastructure
POSIX	Portable Operating System Interface
PPD	Presidential Policy Directive
PPM	Parts per million
PROFINET	Process Field Net
PRTC	Primary reference timing clock
PSC	Packet-based service clock
PTIDES	Programming Temporally Integrated Distributed Embedded Systems
PTP	Precise Time Protocol
QR	Quick Response
R&D	Research and development
RA	Reference architecture
RDA	Research Data Alliance
RDF	Resource Description Framework
REST	Representational State Transfer
RF	Radio frequency
RFC	Request for Comments
RFID	Radio-frequency identification
RP	Relying party
RTAS	Real-Time and Embedded Technology and Applications Symposium
SCADA	Supervisory Control and Data Acquisition
SDH	Synchronous digital hierarchy
SDN	Software Defined Networking
SEC	SDH equipment slave clock
SHA256	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SOA	Service-oriented architecture
SoS	System-of-systems
SPARQL	SPARQL Protocol and RDF Query Language
SW	Software
TAI	International Atomic Time (Temps Atomique International)
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access

Acronym	Expansion
TI	Time interval
TLS	Transport Layer Security
TNC	Trusted Network Communications
TOCS	Transactions on Computer Systems
TS	Technical Specification
TSC	Timestamp counter
TSU	Timestamp unit
TTA	Time-Triggered Architecture
UAV	Unmanned Aerial Vehicle
UMA	User Managed Access
UML	Unified Modeling Language
URL	Universal Resource Locator
US	United States
USB	Universal Serial Bus
UTC	Coordinated Universal Time
UUID	Universally Unique Identifier
VCSE	Virtual Control System Environment
W3C	World Wide Web Consortium
WCET	Worst-case execution time
WSS	Web Services Security
XEP	XMPP Extension Protocol
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

Appendix F. Applying the CPS Framework: An Emergency Response Use Case Analysis

In this appendix, as a simplified example to illustrate the use of the CPS Framework concepts, the CPS Framework is applied to analyze an example CPS use case for Emergency Response. The use case has been limited in scope in order to make this example use of the Framework more clearly expressed.

F.1 Perspective for Applying the Framework

There are many variations, alternative scenarios, and critical features that would make this a comprehensive use case. However, the purpose of the exercise is to demonstrate how to use the Framework, as opposed to how to design emergency response.

The activity, therefore, limited the scope to the initial use case with no elaborations beyond a refinement of the success criteria. It can be expected that this constrained scope will result in significant limitations to the actual value of the analysis to offer insight into emergency response. On the other hand, the concepts presented will be readily recognized by the reader and should enhance the understanding of how the Framework was applied.

The goals for analyzing this problem are to convey an understanding of the *mechanics* of the CPS Framework. To this end the discussion is further limited to:

- The interfaces to the CPS devices and systems involved in the scenario and not the architecting of these systems and devices.
- The *properties* exposed during the Framework analysis of this CPS. For the complete analysis the reader should refer to section 2 of the CPS Framework. A complete analysis of this and its component systems would include all the properties, full design, and full assurance cases of the component systems and this system.
- As offered in section 2.3 Uses of the CPS Framework, this analysis will be of “shallow depth.”

F.2 Workflow for Analyzing the Emergency Response Use Case

The Framework section 2.4, The Description of the CPS Framework, suggests a workflow that starts with the development of a template containing the Domains/Facets/Activities/Aspects/Concerns and allows for a tailoring based on the identified use of the framework from section 2.3.

With a decision to follow a waterfall process, the work in analyzing a use case using the CPS Framework has the following high-level steps:

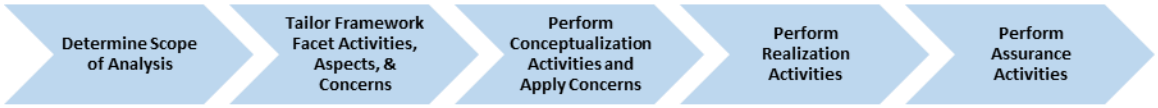


Figure 42: Workflow for Framework Application Sample

F.3 Emergency Response Use Case Original

The following is the use case that was the basis of the exercise. The input provided to the exercise as a starting point is as follows:

"Injured person needs help – 1st responders on the way"

A person has been injured. The injured person sends a text to e-911 for help. An ambulance is dispatched. A smart GPS combines map data with traffic flow data to route the ambulance. Traffic signals are triggered to assist ambulance in navigation (or negotiating) the route.

Systems

- Smart phone
- E-911 system (includes dispatch system)
- Ambulance (includes smart GPS subsystem)
- Traffic Control System
- GPS System
- Cellular Phone Network

Steps

- A person becomes injured.
- Person uses cell phone to text for help.
- The E-911 system gets the person's location from the GPS if available and Cell Tower if not.
- The cell phone provides the location through the cellular system to the E-911 system.
- A request is sent to the closest ambulance.
- The ambulance uses map data + traffic flow data to determine best route.
- The route is sent to the traffic control system.
- Traffic control system changes the lights to green as the ambulance proceeds towards the destination (based on ambulance GPS).

- Light status is fed back to the ambulance (including intersections with no lights).
- The ambulance progress is sent by text to the injured person and the dispatch system (i.e. E-911 system)

Success

- Ambulance arrives at injured person in a timely fashion and in line with the urgency indicated by the injury information.

Variations (not analyzed in this Appendix but noted for future work)

- A power line falls while the ambulance is on route, and the ambulance needs to take a different route.
- A UAV drone support system is used to augment the "smart GPS" of the ambulance.

F.4 Determine Scope of Analysis

Section 2.3 provides for the tailoring of the overall analysis. The following table was used to choose among the possibilities (note that an 'x' in the left column selects an item for inclusion):

Table 21: Tailoring the Analysis

What kind of analysis is this?	
	Processes
x	Waterfall
	Reverse Engineer
	Agile
	Service-Based
	Depth
	Critical-tightly coupled
	Loosely coupled
x	Shallow analysis
	Scopes
	Single CPS Device
	System or subsystem
x	System of systems

Then, the CPS Application Domains directly related to the use case are identified:

Table 22: CPS Application Domains Relevant to Use Case

	Domain
	Advertising
	Aerospace
	Agriculture
	Buildings
x	Cities
	Communities
x	Consumer
	Defense
	Disaster resilience (includes preparedness and crisis management activities)
	Education
x	Emergency response
	Energy (included in “infrastructure”, but this is a very broad category)
	Entertainment/sports
	Environmental monitoring (e.g., weather, greenhouse gas emission tracking)
	Financial services
	Healthcare
x	Infrastructure (communications, power, water)
	Leisure
	Manufacturing
	Science
	Social networks
	Supply chain/retail
x	Transportation

F.5 Tailor Framework Facet Activities, Aspects & Concerns

The Conceptualization Facet was tailored to three activities:

Table 23: Tailoring the Conceptualization Facet

Conceptualization Facet	
	Activities and Artifacts
x	Mission and Business Case Development Artifact: Business use cases

Conceptualization Facet	
x	Functional Decomposition Artifact: Detailed use cases, actors, information exchanges
x	Requirements Analysis Artifact: Functional and non-functional requirements
	Interface Requirements Analysis Artifact: Interface requirements

The Realization Facet was tailored to a single activity. Also note that this activity was limited to resolving two Conceptualization properties that arose from concerns.

Table 24: Tailoring the Realization Facet

Realization Facet	
	Activities and Artifacts
	Business Case Analysis Artifact: Trade studies, lifecycle cost analysis, return on investment, and interdependencies with requirements, regulations, and incentives
	Lifecycle Management Artifact: Lifecycle management and sustainability plan, integrated lifecycle management monitoring
x	Design Artifact: Design documentation, requirement verification, virtual prototypes
	Manufacturing/Implementation Artifact: Manufactured, integrated products, testing plans, and test results
	Operations Artifact: Performance, quality, and product evolution tracking
	Disposal Artifact: Reuse, sustainability and energy recovery assessments, disposal manifests
	Cyber-Physical Abstraction Layer Formation Artifact: Domain (and product)-specific ontologies, modeling languages, and semantics specifications used in all phases of the lifecycle
	Physical Layer Realization Artifact: Physical substrates of the CPS used in all phases of the lifecycle.

The Assurance Facet was tailored to two activities:

Table 25: Tailoring the Assurance Facet

Assurance Facet	
	Activities and Artifacts
	Configuration Audit Artifacts: Product configuration assessment
	Requirements Verification Artifact: Requirements and test results assessment
	Product Certification and Regulatory Compliance Testing Artifact: Certifications
x	Identify Assurance Objectives Artifact: Assurance objectives/analysis report
x	Define Assurance Strategy Artifact: Strategy document/plan
	Control Assurance Evidence Artifact: Control documentation
	Analyze Evidence Artifact: Analysis report
	Provide Assurance Argument Artifact: Assurance argument report
	Provide Estimate of Confidence Artifact: Confidence estimate

The Aspects were tailored as follows:

Table 26: Tailoring of Aspects

	Aspects
x	Functional
	Business
x	Human
x	Trustworthiness
x	Timing
x	Data
x	Boundaries
x	Composition
	Lifecycle

F.6 Perform Conceptualization Activities and Apply Concerns

The Conceptualization Facet has as its artifact the CPS Model which consists of the properties of the intended CPS.

F.6.1 Conceptualization Activity 1: Mission and Business Case Development

This activity involved the analysis of the Use Case and the derivation of an overarching business case and key assumptions and success metrics. The materials started with were broken down into "properties" for further use in the functional decomposition and the requirements analysis.

Note that each of the following could be further broken down into more primitive components. However, that will be left to a future activity.

P_{BC} Business Case:

The goal of this service is to provide medical attention to an injured person. It is assumed for this exercise that the value of human life justifies the expenditures needed to make this service viable.

P_{UC} Use Case:

A person has been injured. The injured person sends a text to E-911 for help. An ambulance is dispatched. A smart GPS combines map data with traffic flow data to route the ambulance. Traffic signals are triggered to assist the ambulance in navigation (or negotiating) the route.

P_{ASS} Assumptions:

The existence of a set of system components is assumed (see below). They are assumed to be functioning as expected. Organizational responsibilities are pre-existing and functioning as expected. No other extraordinary event is occurring at the same time.

P_{SUCC} Success Metric:

The ambulance arrives at the injured person in a timely fashion and in line with the urgency indicated by the injury information.

F.6.2 Conceptualization Activity 2: Functional Decomposition

The use case is analyzed to identify additional properties: system components, information exchanges, and general information about the networks they utilize.

P_{SC} System Components

- Smart phone [cell]
- E-911 system (includes dispatch system) [E911]

- Ambulance (includes smart GPS subsystem) [ambulance]
- Traffic Control System [TCS]
- GPS System [GPS]

P_{NW} Assumptions

- Person's cell phone and ambulance are on a cellular network.
- The TCS and E911 are on a high speed enterprise network.

P_{ARCH} Use Case Steps (how the system should function)

Table 27: Emergency Response Use Case Steps

Data Exchange Messaging for Use Case				
	Step	from Actor	to Actor	data
1	A person becomes injured			
2	Person uses cell phone to text for help	cell	E911	text help message
3	The cell phone gets the person's location from the GPS if available and Cell Tower if not	GPS	cell	location
4	The cell phone provides the location through the cellular system to the E-911 system	cell	E911	location
5	A request is sent to the response (closest) ambulance	E911	ambulance	dispatch
6	The ambulance uses map data + traffic flow data to determine best route	TCS	ambulance	TCS status
7	The route is sent to the traffic control system	ambulance	TCS	route
8	Traffic control system changes the lights to green as the ambulance proceeds towards the destination (based on ambulance GPS).	ambulance	TCS	location
9	Other vehicles move out of the way	TCS	other vehicles	emergency status
10	Light status is fed back to the ambulance (including intersections with no lights)	TCS	ambulance	light status
11	The ambulance progress is sent by text to the injured person and the dispatch system (i.e. E-911 system)	ambulance	cell, TCS, E911	progress

F.6.3 Conceptualization Activity 3: Requirements Analysis

The results of Activity 1 and 2 were studied with respect to each Aspect and their subsidiary Concerns to identify the properties that would comprise the CPS Model. Each property discovered is listed in the corresponding cell for an aspect/concern. Multiple properties are separated by semicolon/line feeds. Aspects that were profiled out (see earlier section) or had no elucidated properties will be blank.

It is likely additional property elaboration would be appropriate to be a complete result to the depth of this CPS analysis. However, the properties identified provide a good guide to the nature and abstraction of such properties for this kind of effort.

Table 28: Emergency Response Requirements Analysis

Aspect	Concern	Requirements Analysis
Functional	actuation	ambulance gets sent; ambulance proceeds unimpeded; vehicles move out of the way;
Functional	communication	deliver text message to E911; location delivered; texting (cell to E911); E911 to ambulance; GPS identification of cell location; GPS sends cell location to ambulance; ambulance sends route to TCS; TCS to all vehicles;
Functional	controllability	E911 identify and dispatch ambulance; TCS light control; E911 monitors progress; optimal route;
Functional	functionality	Use Cases; Business Cases; success criteria; assumptions;
Functional	measurability	successful arrivals of the ambulances; average time to get to person;
Functional	monitorability	timestamped sequence of events; status of all systems;
Functional	performance	ambulance arrives within target time; vehicles are informed in time to move;
Functional	physical	

Aspect	Concern	Requirements Analysis
Functional	physical context	location of ambulance relative to traffic, intersections and destination at a given time; location of person;
Functional	sensing	location of ambulance, person; time for stamping; traffic flows;
Functional	uncertainty	route uncertainty better less than road dimension; location uncertainty small enough to determine location of person and ambulance; systems are time synchronized to establish reliable sequence of events;
Business	enterprise	
Business	cost	
Business	environment	
Business	policy	
Business	quality	
Business	regulatory	
Business	time to market	
Human	human factors	
Human	usability	emergency text should be sent from a simple unambiguous behavior (no dialog/navigation/typing); other vehicle interpretation of guidance to get out of ambulance way should be unambiguous;
Human	utility	
Trustworthiness	privacy	personally identifiable information (PII) from the emergency response is protected in flight;

Aspect	Concern	Requirements Analysis
Trustworthiness	reliability	ambulance, cell, TCS, E911, and GPS have an acceptable combined reliability (e.g. 95% assurance that the ambulance arrives in the timely fashion).
Trustworthiness	resilience	failure of ambulance is detected and another ambulance dispatched; in order to maintain the acceptable combined performance, redundant or backup systems are available to maintain timely response for any emergency response; the ambulance, TCS, E911 timing physical and messaging signals have resilience;
Trustworthiness	safety	TCS avoids creating hazardous conditions in managing lights with respect to cross streets; E911, TCS, GPS are designed to fail functional; Directions to the ambulance does not create hazard to the ambulance operation; Route should convey the ambulance safely;
Trustworthiness	security	Messaging is not confidential; Records of the emergency response are protected at rest; Source and destination of messages are validated; Messages received have not been tampered with; All messaging with guaranteed delivery; All components protect against physical tamper; The ambulance, TCS, and E911 timing, physical and messaging signals have integrity; The ambulance, TCS, and E911 timing, physical and messaging signals have availability;
Timing	logical time	The sequence of events is as described in the Use Case;
Timing	managing timing and latency	cell network delivers text message in a timely manner (e.g. <10 seconds); TCS/E911 network have minimum message latency (e.g. <1 seconds);
Timing	synchronization	TCS, E911, ambulance must have a common time scale (e.g. UTC)
Timing	time awareness	TCS, E911, ambulance can give a timely response;

Aspect	Concern	Requirements Analysis
Timing	time-interval and latency control	time interval from sending text message to E911 and ambulance arrival is timely (e.g. <6 minutes); timing of TCS must perform relevant to the movement of the ambulance, and other vehicles and cross-traffic to effect rapid progress of ambulance and minimize impact to cross-traffic (e.g. predicted progress of the ambulance accurate to 1s);
Data	data semantics	text help message; location; dispatch; TCS status; route; emergency status; light status; progress; text help message is encoded as a "text message";
Data	identity	personal phone, ambulance, TCS system(s), E911 system, intersections, response event;
Data	operations on data	fuse data from various sources to determine best route; evaluate ambulance characteristics and availability to optimize allocation;
Data	relationship between data	locations of ambulances, person, route, traffic must be analyzed and correlated;
Boundaries	cross-domain	emergency response interacting with traffic control and, … see domain list
Boundaries	connectivity	cell phone can connect with cell phone towers -- one hop to cell tower, GPS network receive broadcast
Boundaries	responsibility	TCS is the responsibility of municipal government traffic management; The holder of the phone has the ability to participate in the scenario; The E911 is the responsibility of the government e-response; The ambulance is part of the emergency response function and may be fire/police/private;
Composition	adaptability	work with different cell phone technologies; use cell towers or GPS for location;

Aspect	Concern	Requirements Analysis
Composition	complexity	work with older (flip phones); deal with different kinds and managements of ambulance services; "green lighting" can cause impact on other existing flows;
Composition	constructivity	Emergency response requires E911 system, the diversity of cell phones, cell phone networks, and ambulance services; coordination between neighbor TCS;
Composition	discoverability	ambulance location and capability; cellphone location; traffic need to be determined; pertinent TCS identity and capability;
Lifecycle	deployability	
Lifecycle	disposability	
Lifecycle	engineerability	
Lifecycle	maintainability	
Lifecycle	operatability	
Lifecycle	procureability	
Lifecycle	producibility	

F.7 Perform Realization Activities

This exercising of the realization activities produced two example design/test pairs.

During this analysis, an accelerated design process reviewed some on-line literature and derived a first level design for two properties enumerated in the Conceptualization activities.

These designs were provided with hypothetical test plans that could verify the successful performance of the design.

Table 29: Realization Activity

Realization		
Aspect/Concern: Property	**Design**	**Test Format: Test ID (TID) Test Description**
Functional/Performance: 2.6.1.4 ambulance arrives within target time	$D_{scenario\ timing}$ Steps[33] 1 – start 2 – 10 s high confidence that SMS gets through to target 3 – 60 s maximum GPS location acquisition time 4 – 10 s same as 2 5 – 10 s E911 has situational awareness of all ambulances locations and metrics 6 – 3 s ambulance is enabled to rapid routing 7 – 3 s high quality of service to TCS from ambulance 8 – 4.5 m drive time to cell phone 9 10 11 6 minutes target response time location accuracy 50-300 m maximum distance for ambulance to travel 5, 35 mph	$T_{scenario\ timing}$ TID 1. Measure SMS propagation over 1000 messages and verify <10 s TID 2. Measure GPS location acquisition time from a selected set of locations and cell phone start conditions and verify <60 s in all cases TID 3. Measure ambulance routing capability and verify < 3 s over 100 random locations within 4 mile radius TID 4. Measure transit time of message from ambulance to TCS and verify <3 s over 100 locations throughout territory. TID 5. From a set of 100 test locations and different traffic conditions, verify that test response driving times are < 4.5 minutes
Composition/Adaptability 2.6.1.5 work with different cell phone technologies	$D_{cell\ phone\ technologies}$ Rely on SMS and cellular location as a minimum requirement	$T_{cell\ phone\ technologies}$ TID1. Measure SMS transfer and locational accuracy for each available cell phone on the market and several legacy phones including flip phones. Verify 10 s SMS. Verify 30 m location accuracy

[33] Some metrics from this article were used in the design analysis:
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4288957/

F.8 Perform Assurance Activities

The conceptualization facet produces the CPS Model that consists of properties of the CPS. Some of these properties result from interpreting the initial business case and the impact of the relevant aspects and concerns on the business case.

Two of the activities of the Assurance Facet were undertaken. Refer to section A.3.4 for definition of terminology.

To identify Assurance Objectives, we turn back to the properties that were defined as the artifacts of the Conceptualization Facet and we identify as an objective the assurance of those properties. There were two properties defined in the requirements analysis activity for which all three facets were exercised – corresponding to the *performance concern of the functional aspect* and the *adaptability concern of the composition aspect*:

Functional Aspect/Performance Concern driven property:

$P_{\text{Ambulance shall arrive within target time}}$

Composition Aspect/Adaptability Concern driven property:

$P_{\text{Shall function with different cell phone technologies}}$

These two properties comprise the assurance objective.

The assurance strategy for these two properties makes use of the design and test artifacts called out in the Realization Facet. The strategy is to provide argumentation to the effect that the successful execution of the test suffices to make the judgment that the properties are met:

H_{Leaf} is the argumentation that says that the test, design, and tracing to the property is sufficient to conclude that the property in question has been met:

$A(P, D, T) =_{Def} H_{Leaf} (P_{\text{Ambulance shall arrive within target time}}, D_{\text{scenario timing}}, T_{\text{scenario timing}})$

$A(P, D, T) =_{Def} H_{Leaf} (P_{\text{Shall function with different cell phone technologies}}, D_{\text{cell phone technologies}}, T_{\text{cell phone technologies}})$