Using Co-Simulation for Model Reuse and Experiment Reproducibility

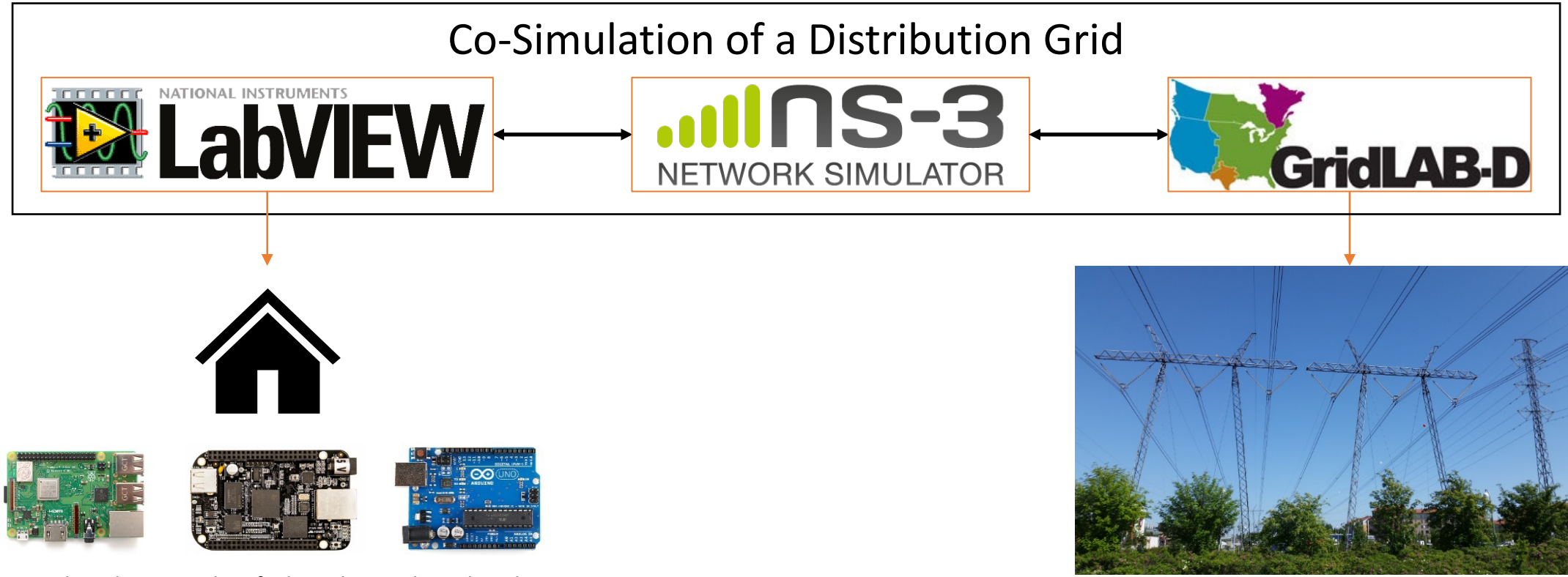
National Institute of Standards and Technology
U.S. Department of Commerce

Thomas Roth

December 7, 2021

Co-Simulation Definition

A **co-simulation** is a simulation with multiple subsystems that are executed simultaneously and exchange data at runtime.



Certain commercial products are identified to adequately explain the concept; this does not imply endorsement or recommendation by NIST, nor does it imply that such products are necessarily the best available for the purpose.

Applications of Co-Simulation

- **Human-in-the-loop training**
- **Model reuse across simulators**
- **Shared or unique physical resources**
- **Simulation of complex systems (CPS/IoT)**
  - Often safety-critical systems
  - Often cannot run experiments on the live systems
  - Often require a combination of expertise to understand

Approaches to Co-Simulation

ad hoc

- MATLAB S-Functions, TCP/IP Sockets, …

frameworks

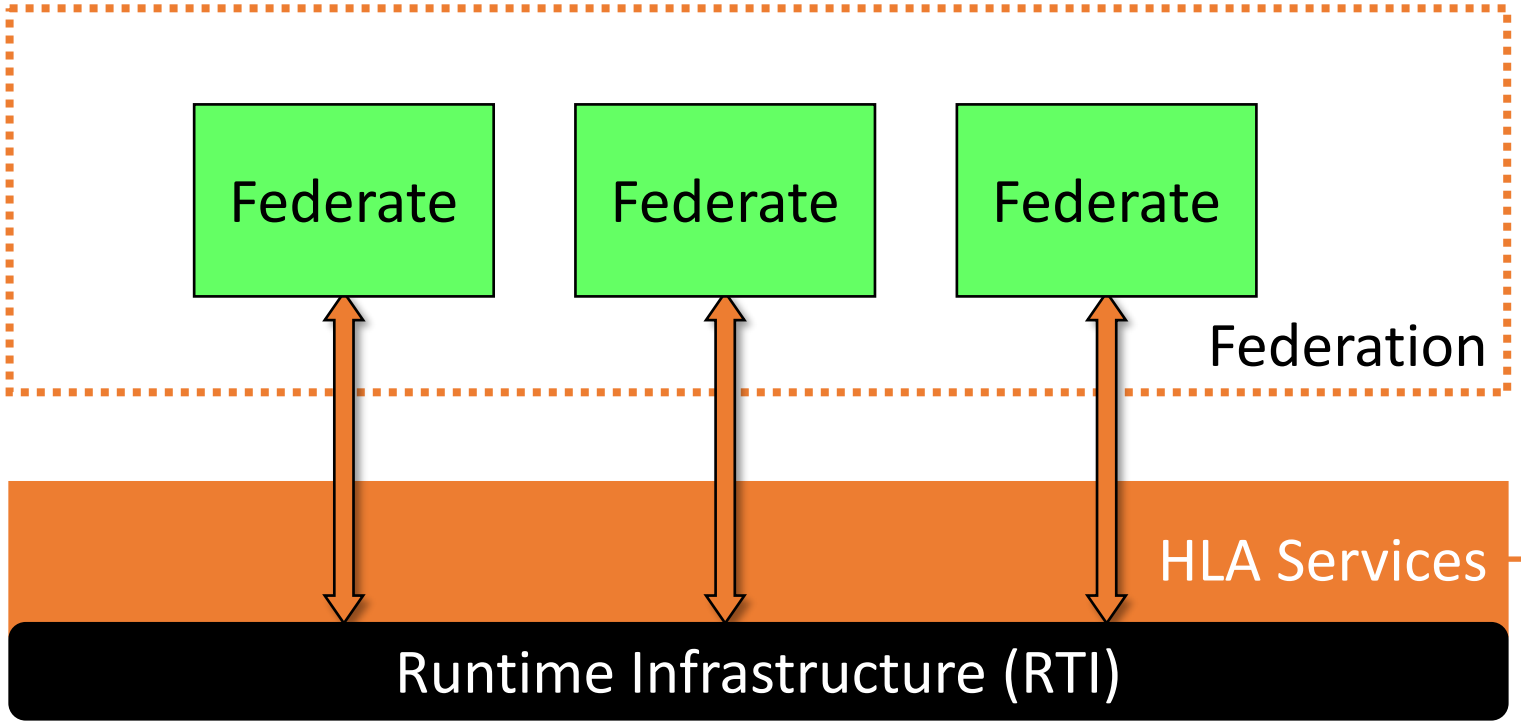
- HELICS, mosaik, ...

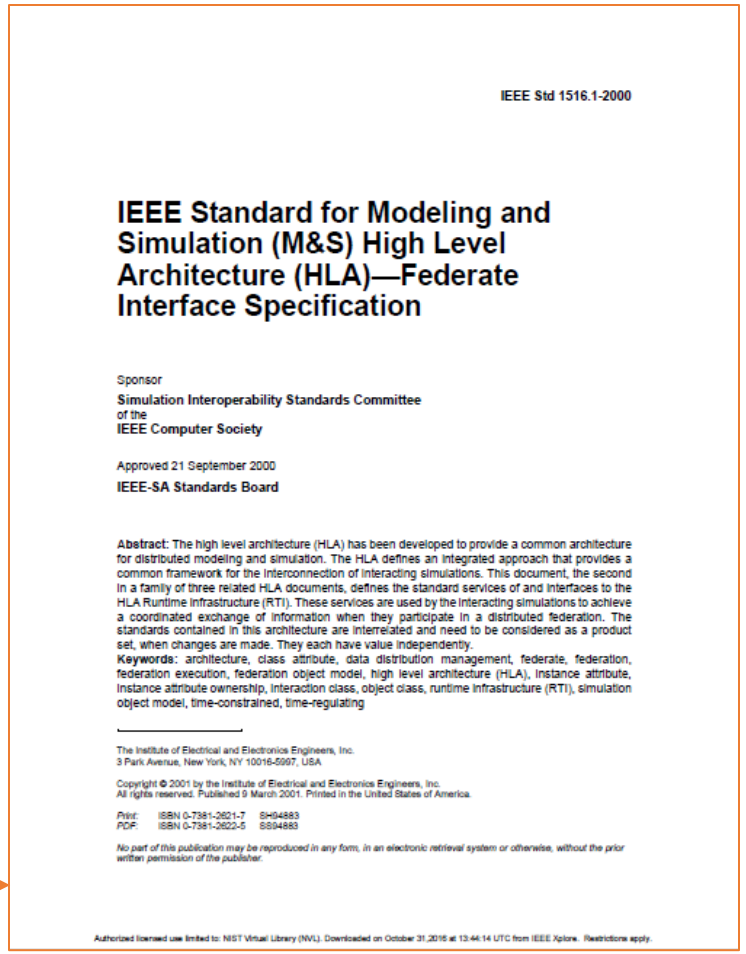
standards

- High Level Architecture (HLA)
- Functional Mock-up Interface (FMI)
- …

High Level Architecture (HLA)

The IEEE **High Level Architecture (HLA)** is a co-simulation standard defining the services a set of ***federates*** can use in a ***federation***.





IEEE Std 1516.1-2000

IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Federate Interface Specification

Sponsor
Simulation Interoperability Standards Committee
of the
IEEE Computer Society

Approved 21 September 2000
IEEE-SA Standards Board

Abstract: The high level architecture (HLA) has been developed to provide a common architecture for distributed modeling and simulation. The HLA defines an integrated approach that provides a common framework for the interconnection of interacting simulations. This document, the second in a family of three related HLA documents, defines the standard services of and interfaces to the HLA Runtime Infrastructure (RTI). These services are used by the interacting simulations to achieve a coordinated exchange of information when they participate in a distributed federation. The standards contained in this architecture are interrelated and need to be considered as a product set, when changes are made. They each have value independently.
Keywords: architecture, class attribute, data distribution management, federate, federation, federation execution, federation object model, high level architecture (HLA), instance attribute, instance attribute ownership, interaction class, object class, runtime infrastructure (RTI), simulation object model, time-constrained, time-regulating

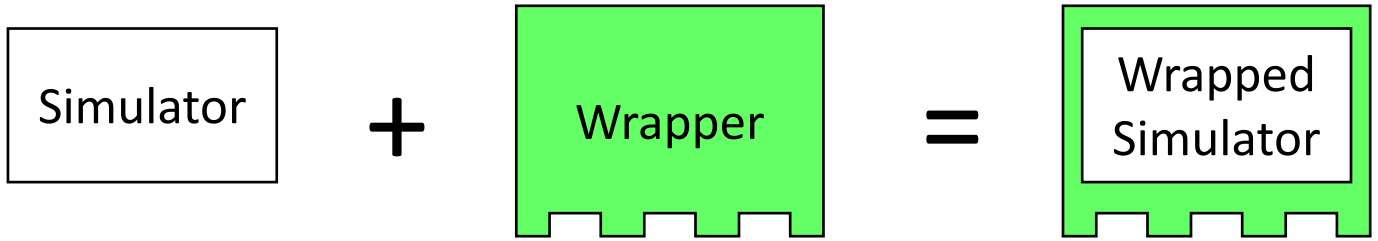
The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 0-7381-2621-7 SH94883
PDF: ISBN 0-7381-2622-5 SS94883

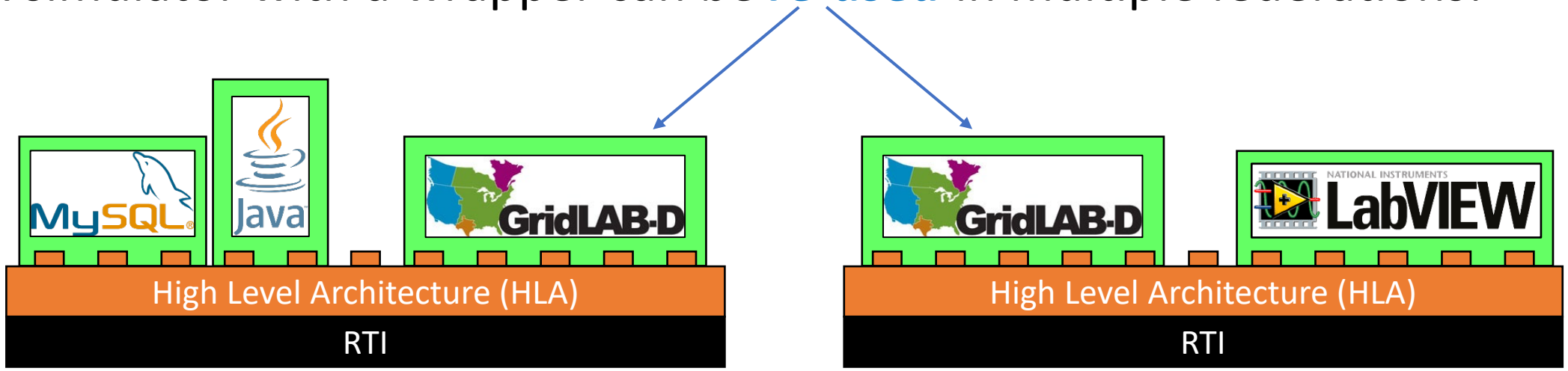
IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Federate Interface Specification (DOI 10.1109/IEEESTD.2010.5557728)

Simulation Integration

- A **wrapper** is software that defines the method of ***time synchronization*** and ***data exchange*** used for a simulator in a federation.



- A simulator with a wrapper can be ***re-used*** in multiple federations:



Federate Interoperability

Connectivity

- Able to share data (protocols)

Semantic Interoperability

- Able to ***understand*** shared data (data models)

'Functional' Interoperability

- Able to ***effectively use*** shared data

NIST Co-Simulation Platform

Distributed as a virtual machine

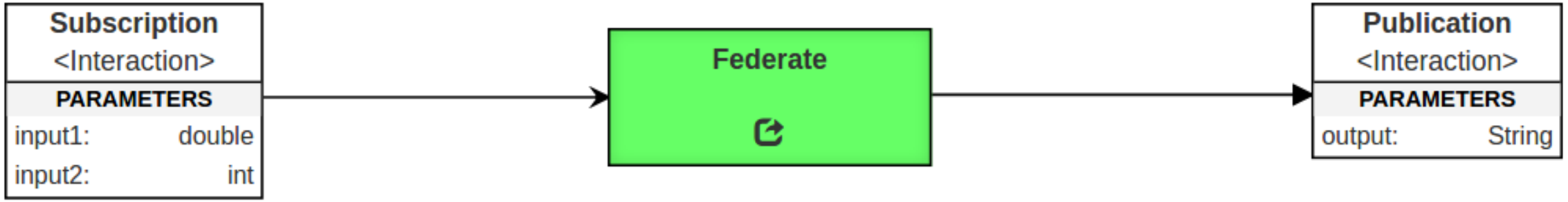
Contains a graphical experiment and federate design environment

Uses code generation to turn models into executable code

available at https://github.com/usnistgov/ucef

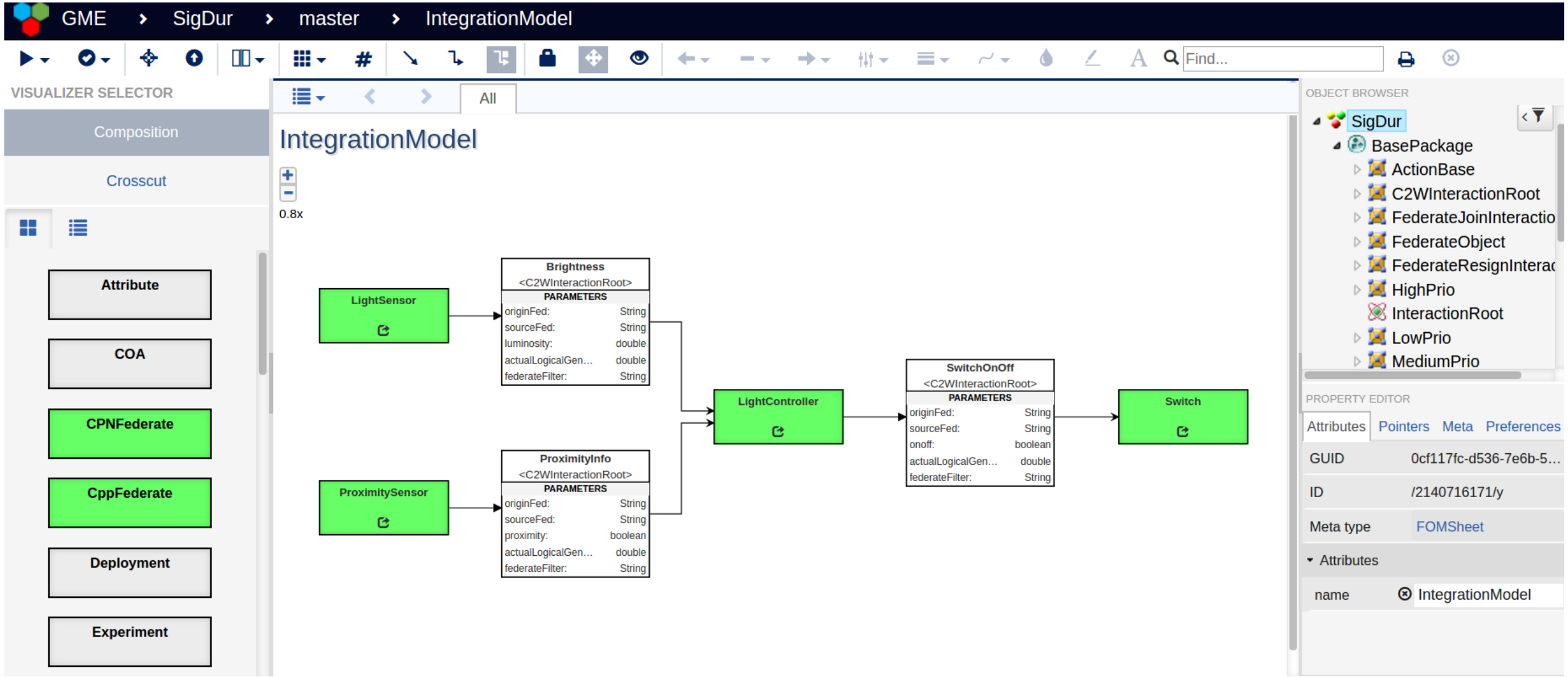
HLA Modeling Language

In UCEF a graphical language is used to design federates:

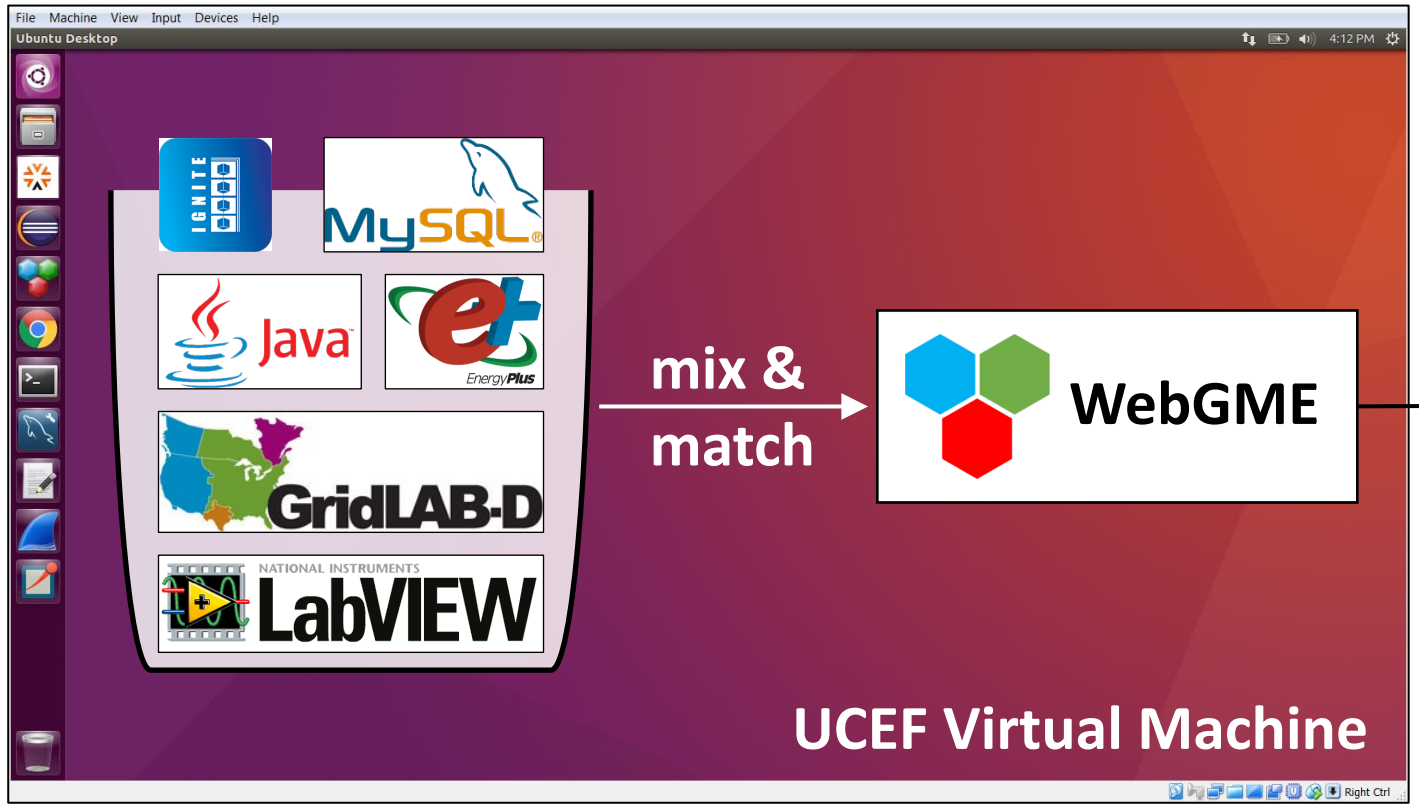


The federates modeled in this language can be transformed into executable code / simulation models using code generation

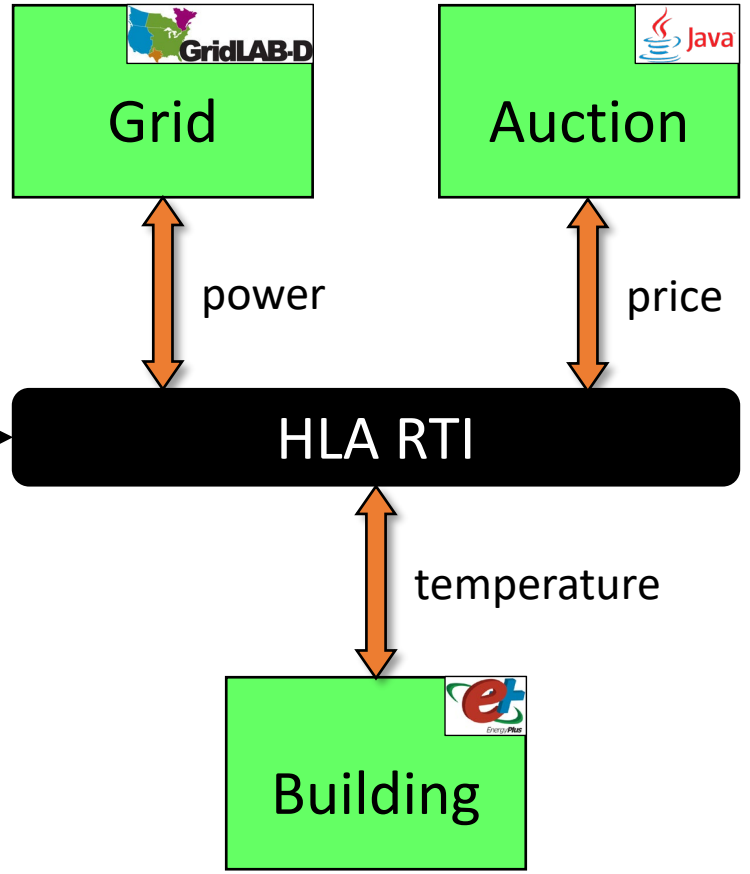
Web-based Generic Modeling Environment (WebGME)



UCEF: Portable HLA Development Kit



generates



Federate Interoperability

Connectivity

- Able to share data (protocols)

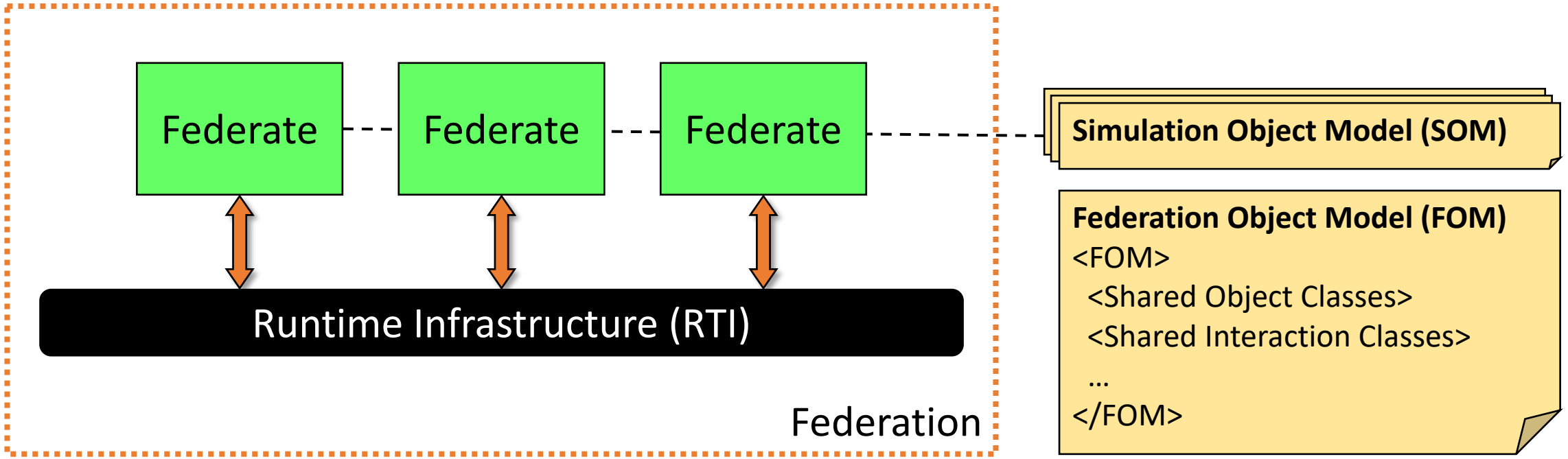
Semantic Interoperability

- Able to ***understand*** shared data (data models)

'Functional' Interoperability

- Able to ***effectively use*** shared data

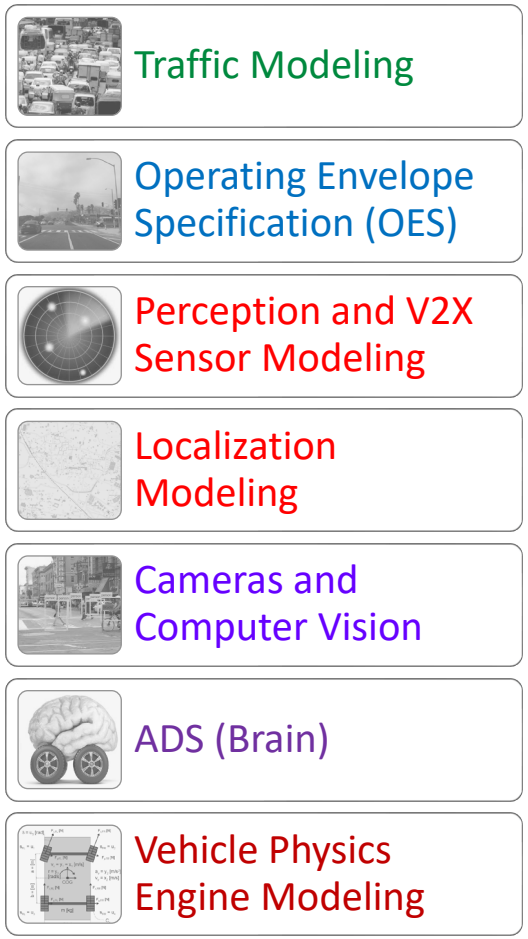
Federation Object Model (FOM)

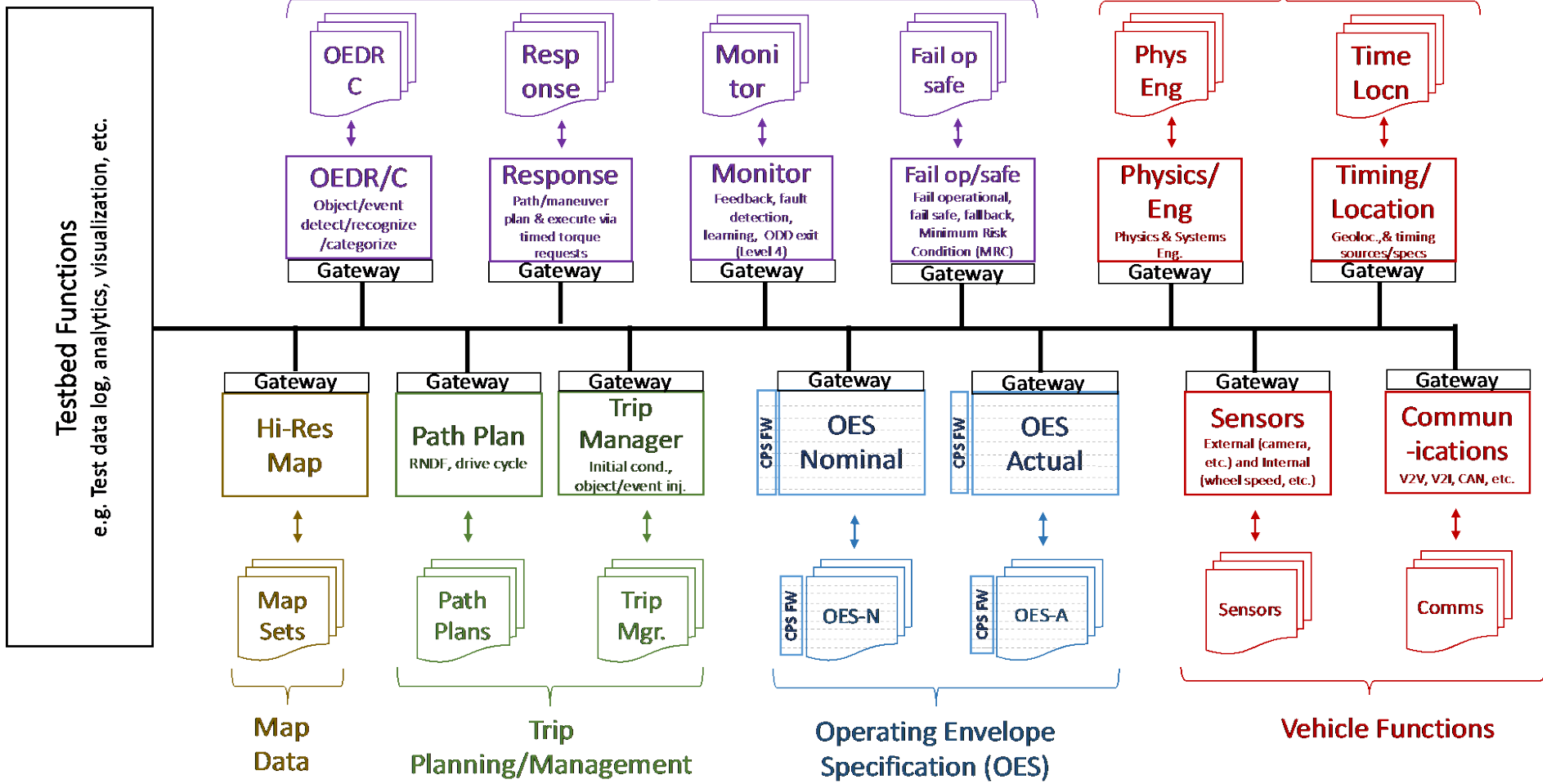


- **Examples:**
  - SISO-STD-001.1-2015
    Real-time Platform Reference Federation Object Model (RPR FOM)
  - SISO-STD-018-00-2020
    Space Reference Federation Object Model (SpaceFOM)

Federation Schematic

AV Functions include:





Federate Interoperability

Connectivity

- Able to share data (protocols)

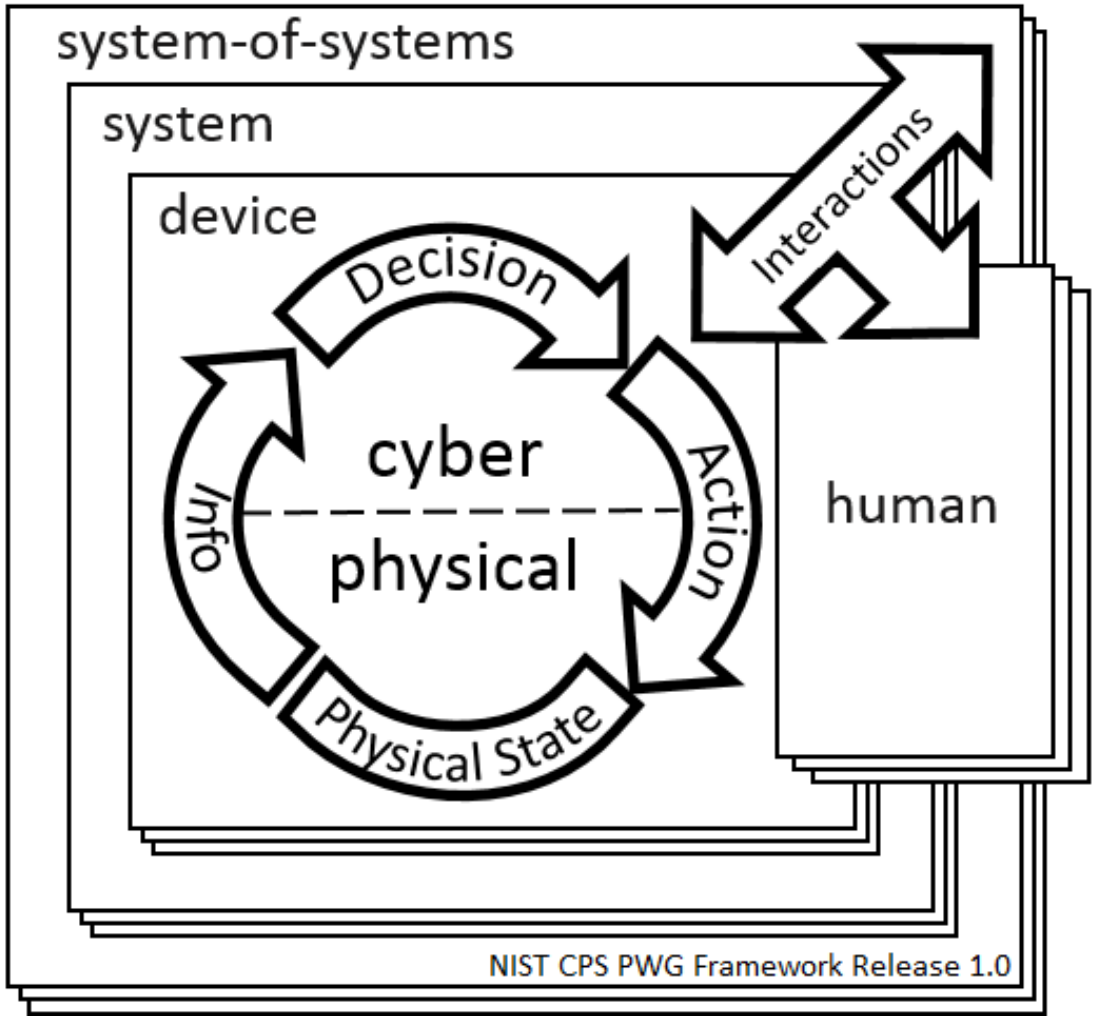
Semantic Interoperability

- Able to ***understand*** shared data (data models)

‘Functional’ Interoperability

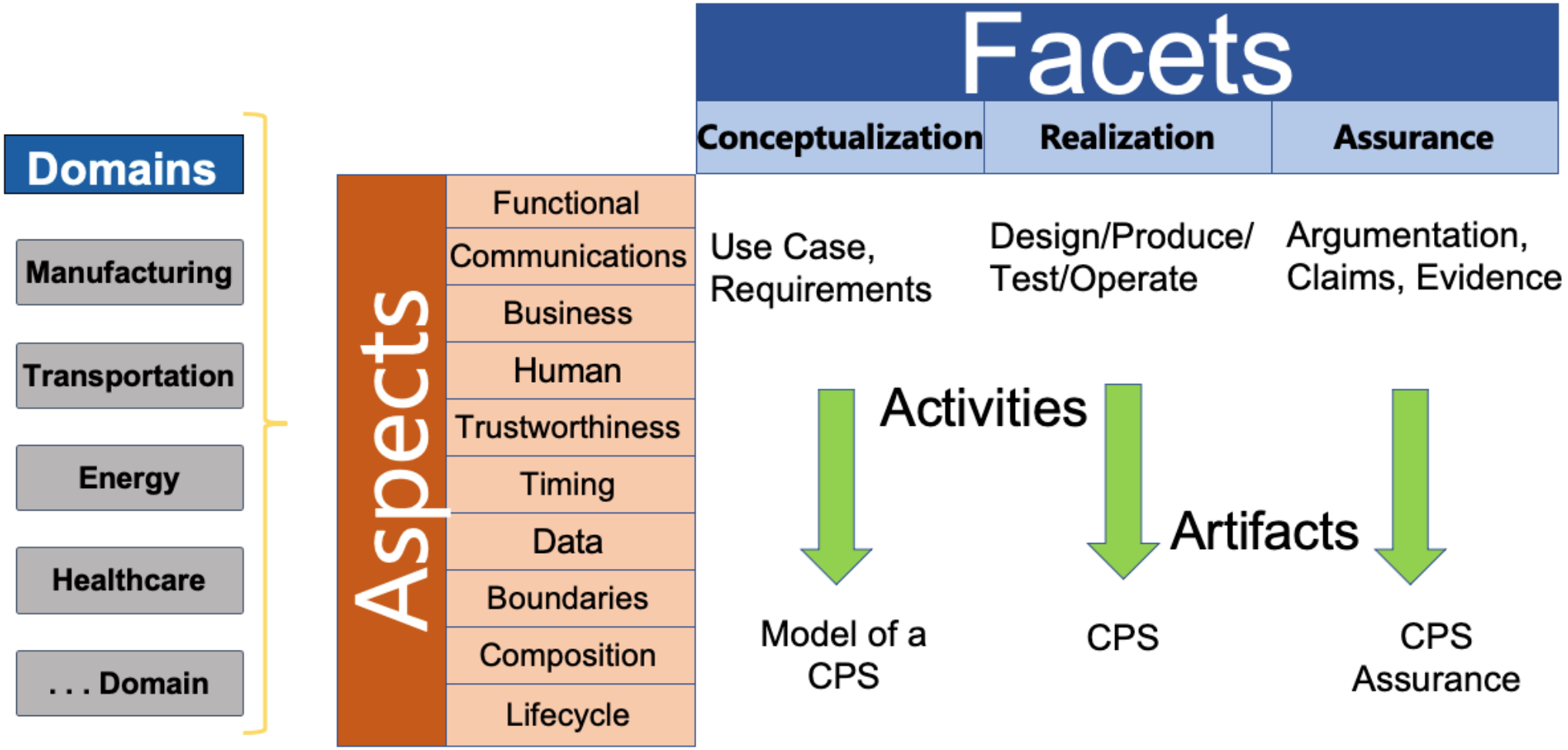
- Able to ***effectively use*** shared data

Cyber-Physical Systems (CPS)



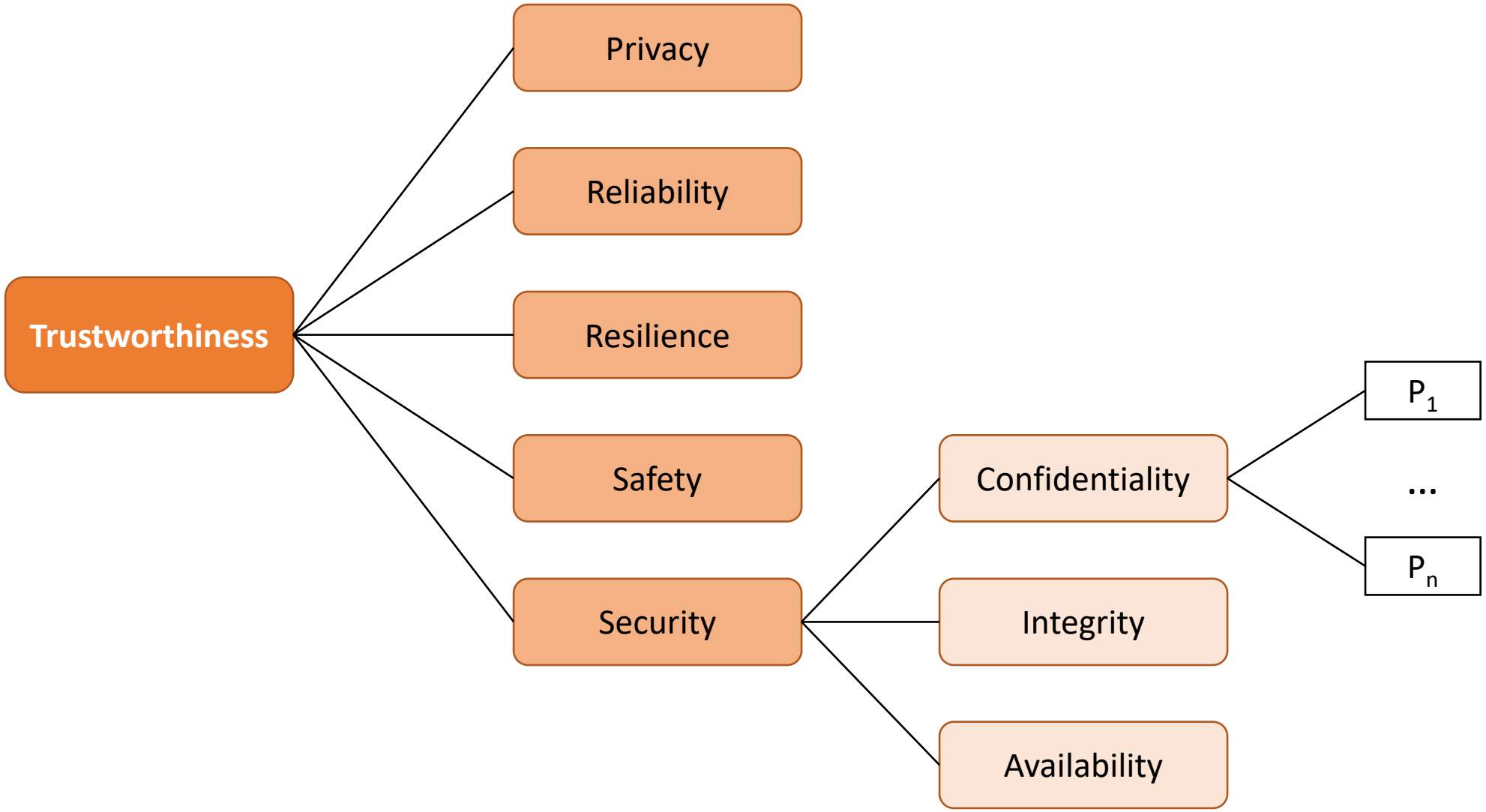
Framework for Cyber-Physical Systems: Volume 1, Overview (10.6028/NIST.SP.1500-201)

NIST CPS Framework

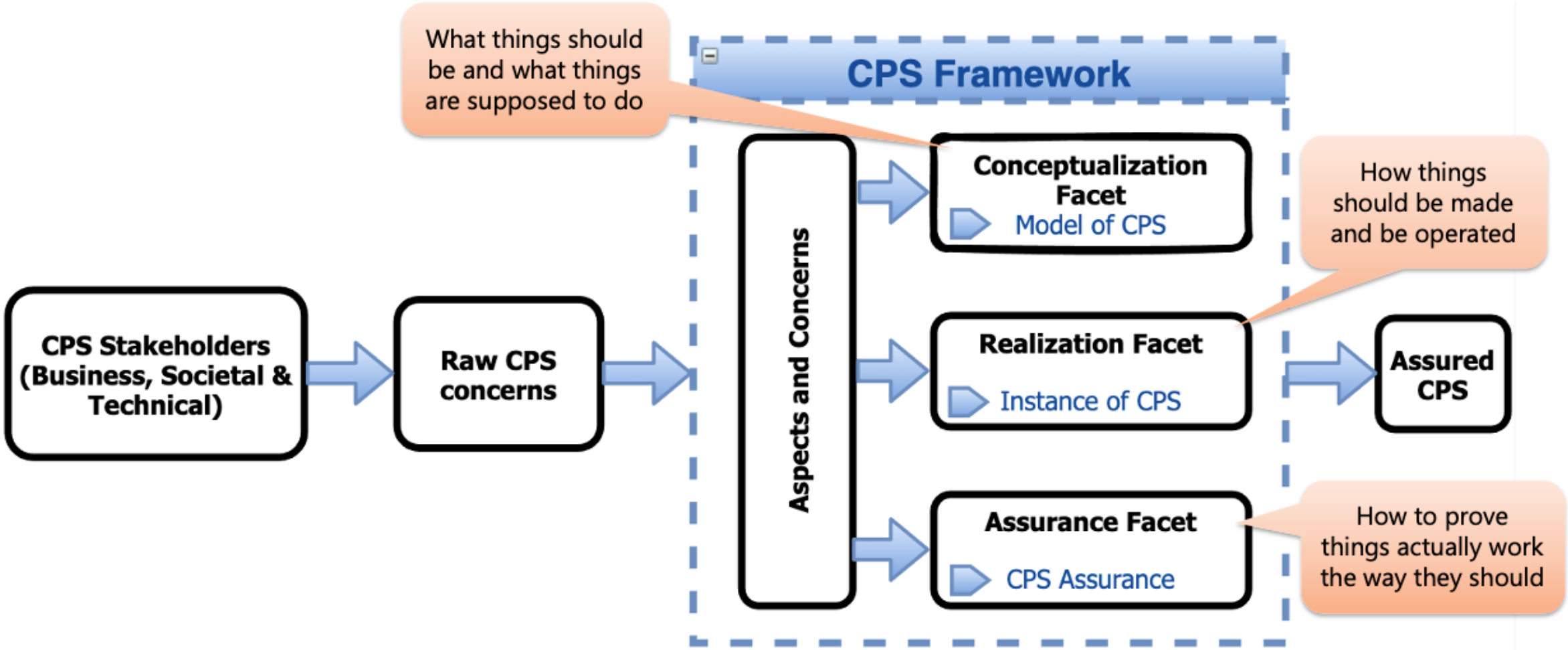


Framework for Cyber-Physical Systems: Volume 1, Overview (10.6028/NIST.SP.1500-201)

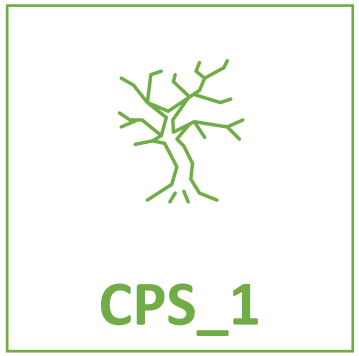
CPS Framework - Property Trees



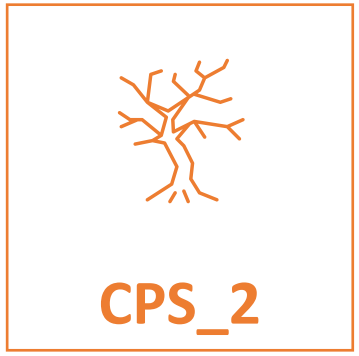
Generation of Property Trees



Composition of CPS



+



Will these systems work together?

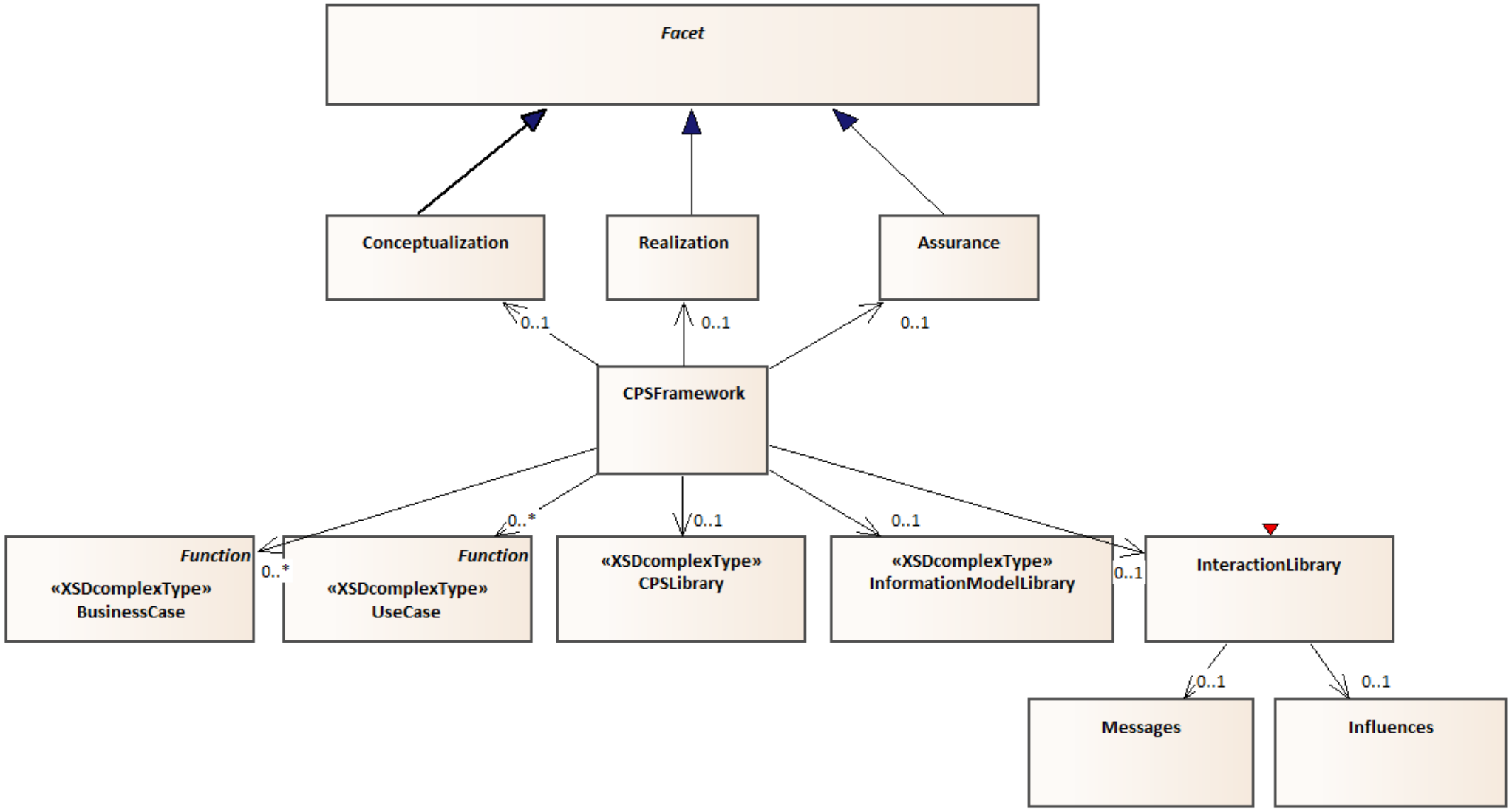
CPS_1.Trustworthiness.Security.Confidentiality.P1

+ CPS_2.Trustworthiness.Security.Confidentiality.P1

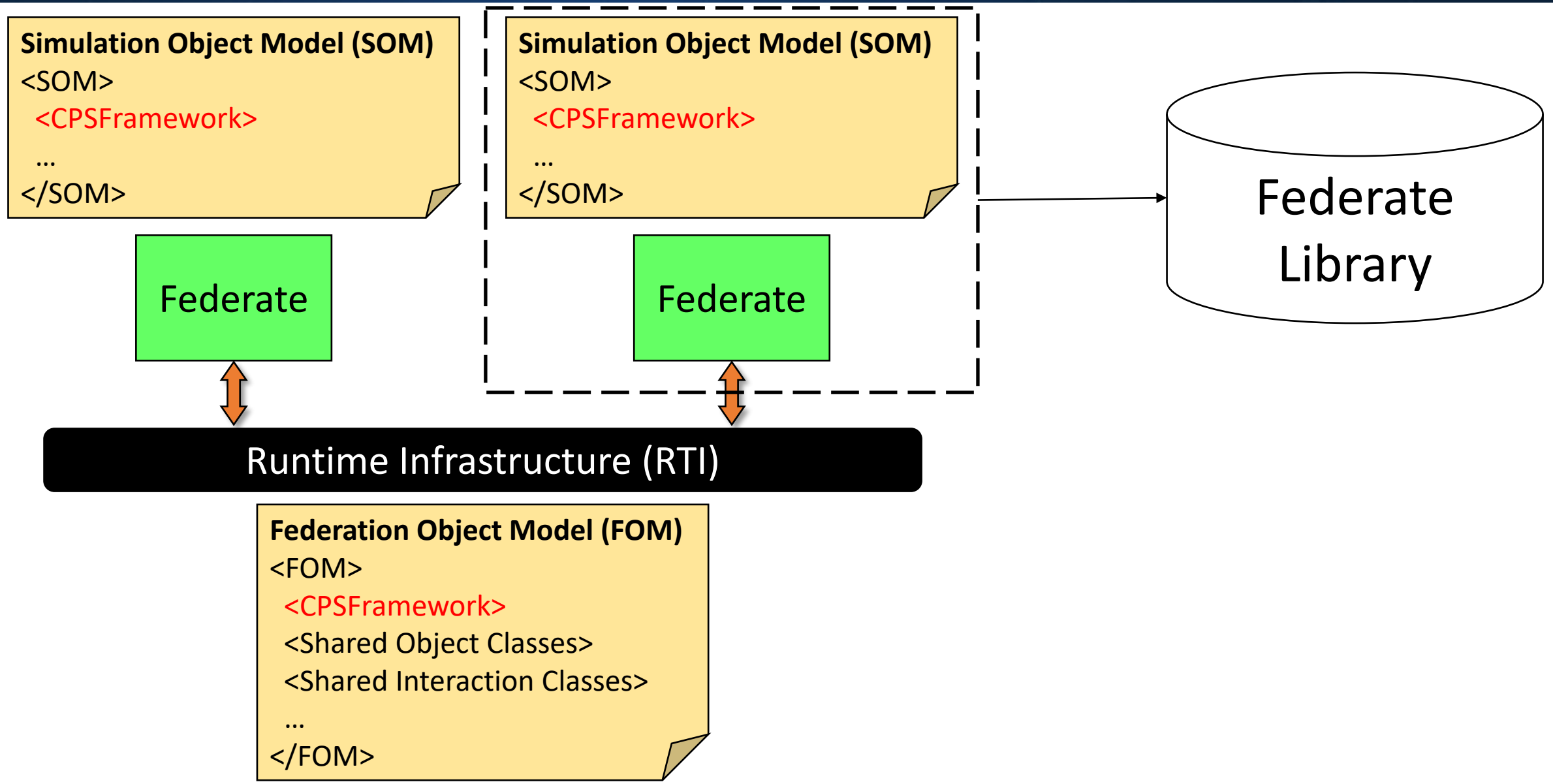
+ CPS_2.Trustworthiness.Security.Confidentiality.P2

= **VALID SYSTEM**

CPS Descriptor



Federate Metadata



Questions

Thomas Roth

NIST Communications Technology Laboratory

IoT Devices and Infrastructures Group

thomas.roth@nist.gov