Toward Plug and Play Medical Cyber-Physical Systems (Part 3)

Insup Lee
PRECISE Center
School of Engineering and Applied Science
University of Pennsylvania

EMSIG Autumn School
Copenhagen, Denmark
November 10, 2015

Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Virtual Medical Device (VMD)

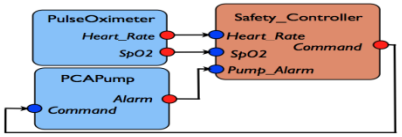
- **MD PnP** enables the concept of **Virtual Medical Device:**
  - A set of medical devices coordinating over a network for clinical scenario.

Device Coordination Algorithm

+

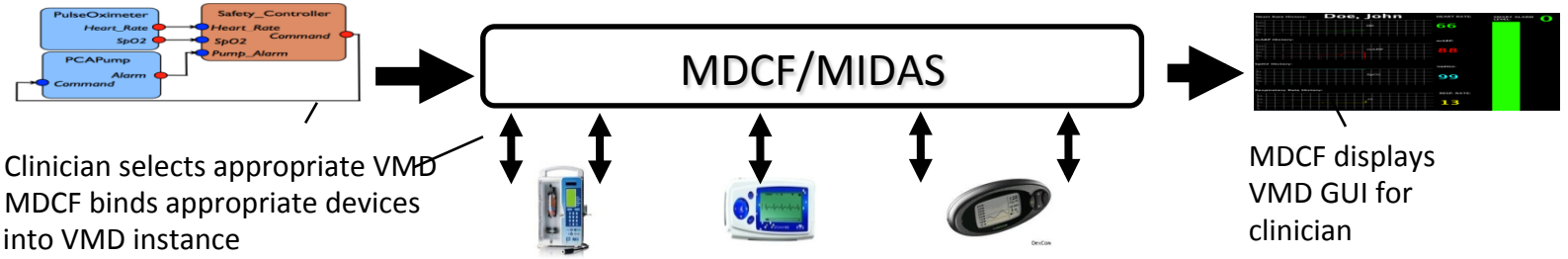
Medical Device Types

=



Virtual Medical Device (VMD)

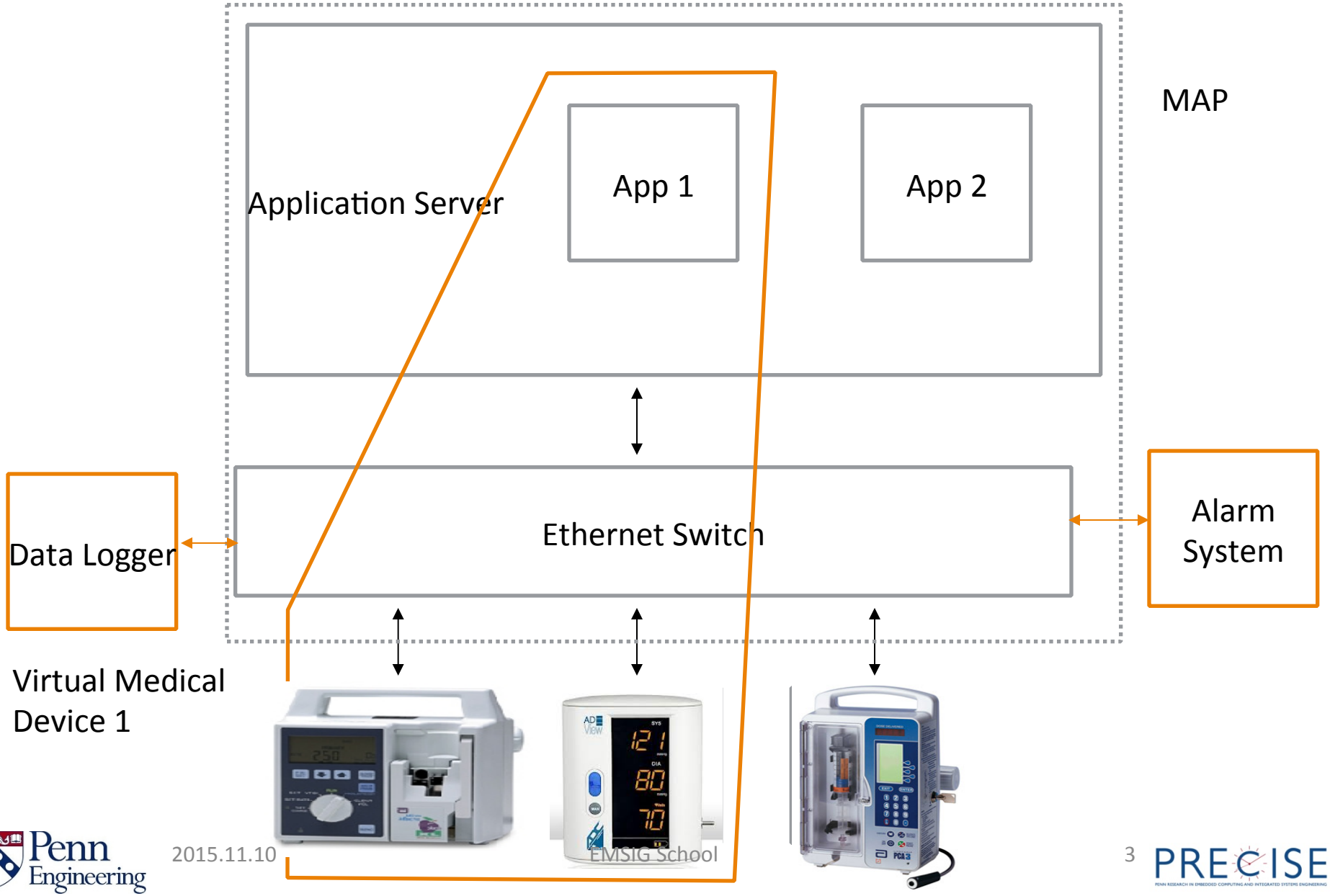
- VMD does not physically exist until instantiated at a hospital.
- The Medical Device Coordination Framework (MDCF) is prototype middleware for managing the correct composition of medical devices into VMD.



- Clinician selects appropriate VMD
- MDCF binds appropriate devices into VMD instance

MDCF displays VMD GUI for clinician

Architecture (Impl.)



SAFETY ASSURANCE FOR PLUG & PLAY MEDICAL SYSTEMS

Need New Paradigm

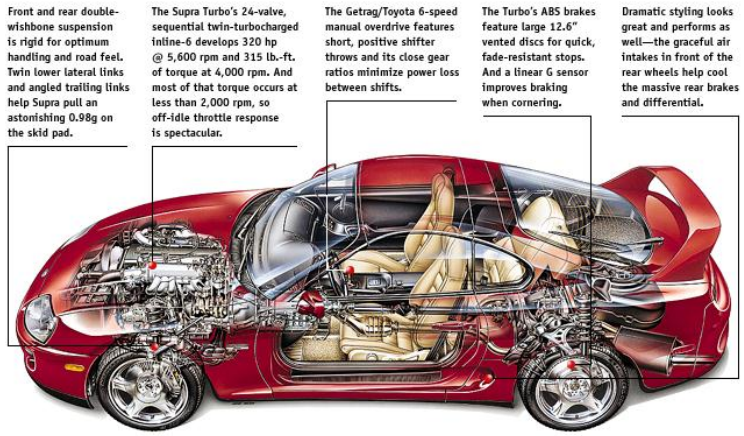
- Why traditional approach won't scale
  - Medical devices need to work as stand alone as well as an integrated system

- Certify VMD app based on abstract interfaces
  - System can use components (including medical devices) that satisfy their specs

Traditional safety critical systems...

Aerospace

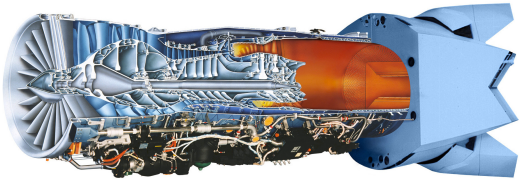
Nuclear

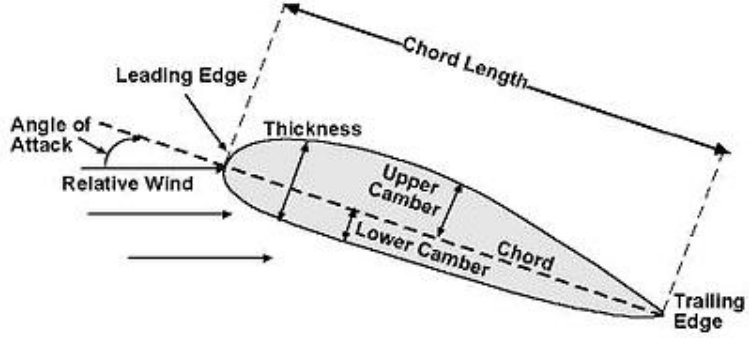


Automotive

Emergent Properties

Example: Top-Speed of an airplane



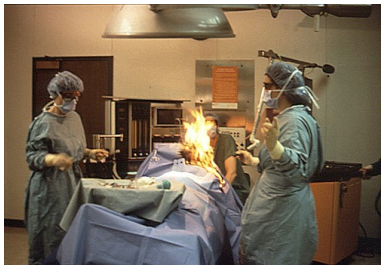


Top-speed is a function of engines + fuselage + wings + flight control software (FCS)

Does it make sense to talk about the top-speed of the FCS?

Emergent Properties

Example: Safety (Laser / Ventilator Interlock)



Safe system

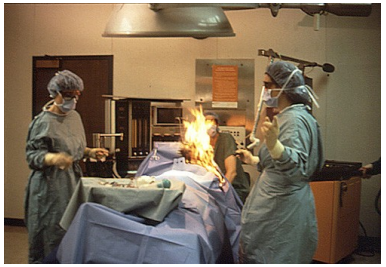
(Emergent) Behavior of Integrated System

Laser is on
&
Ventilator is on

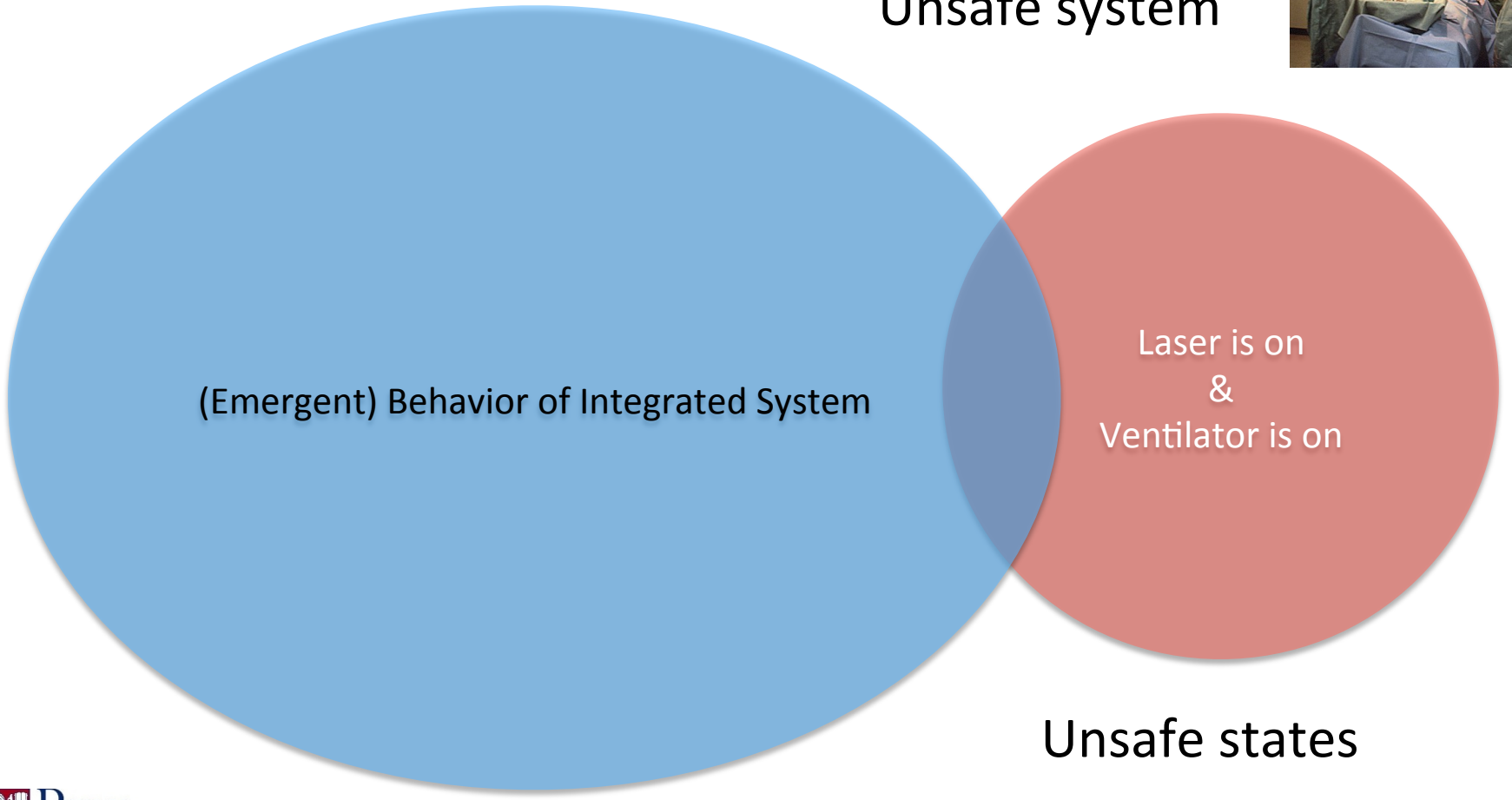
Unsafe states

Emergent Properties

Example: Safety (Laser / Ventilator Interlock)



Unsafe system



Unsafe states

System Integration

In other safety critical domains, there is typically a prime contractor that is responsible for integration and system-level verification and validation.

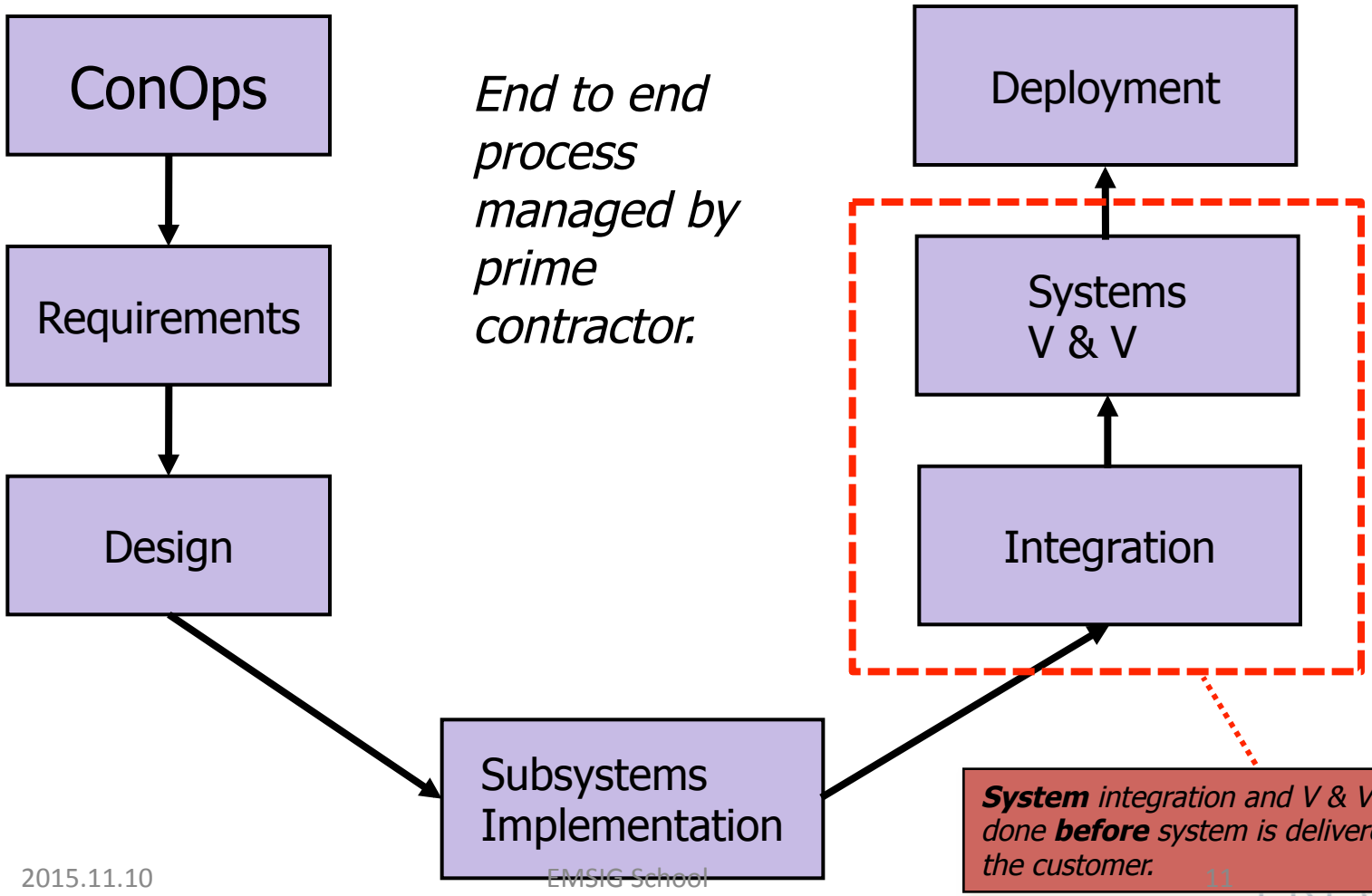
- Integration is performed *before* deployment with full knowledge and behavior of components being integrated
- Integrator has expert-level technical knowledge of components & system behavior
- Responsible for overall system
  - Verification & Validation
  - Safety arguments
  - Certification

787 Final Assembly Integrator - The Boeing Company

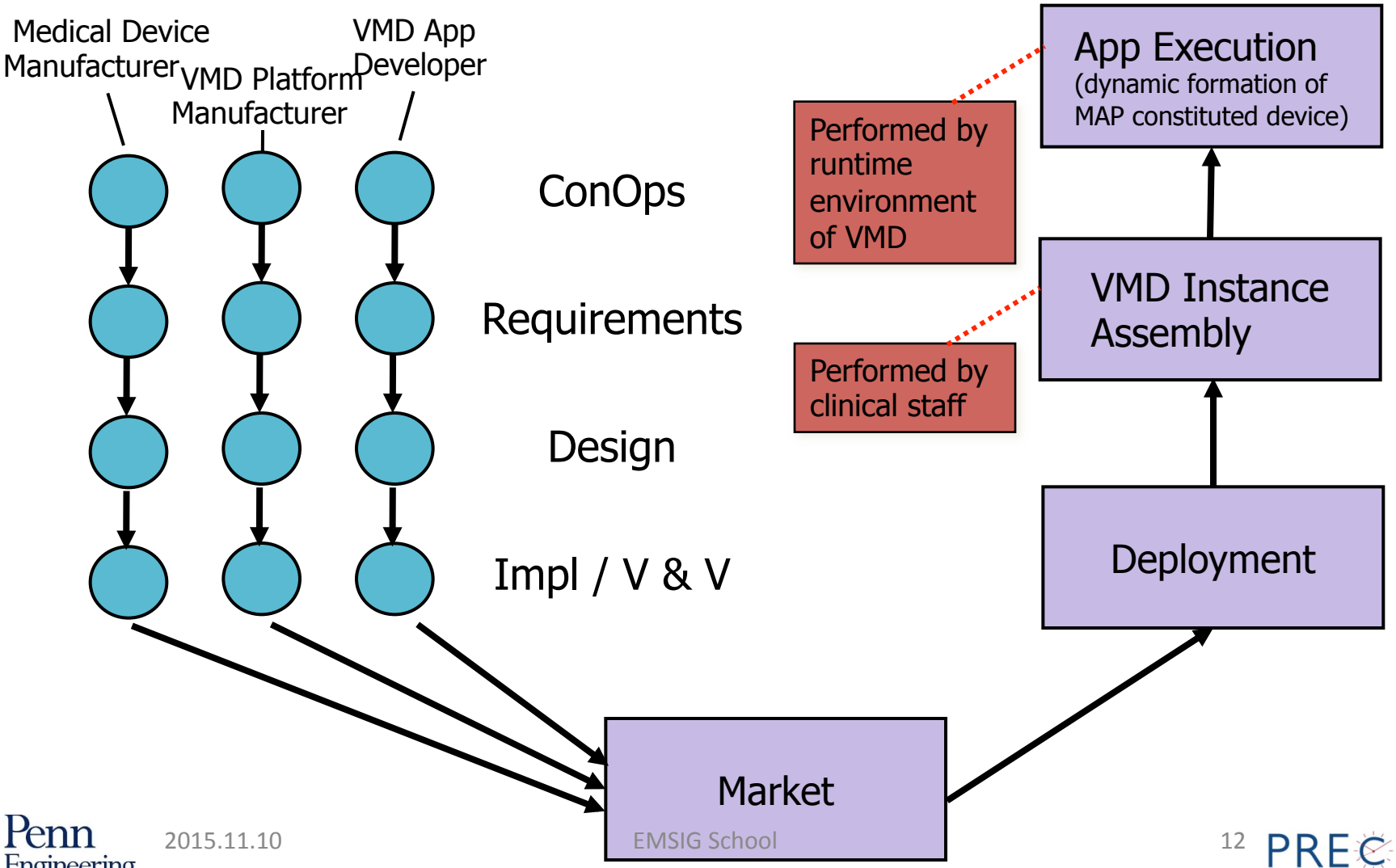
As Prime Contractor/Integrator for the final assembly of the composite 787 Dreamliner in Everett, WA,

System Integration

In other safety critical domains, there is a typically a prime contractor that is responsible for integration and system-level verification and validation.



VMD Development & Assembly



VMD Characteristics

In other safety critical domains, there is a typically a prime contractor that is responsible for integration and system-level verification and validation.

- Integration is performed *before* deployment with full knowledge and behavior of components being integrated
- Integrator has expert-level technical knowledge of components & system behavior
- Responsible for overall system
  - Verification & Validation
  - Safety arguments
  - Certification

With VMDs, there is **no** prime contractor that is responsible for integration and system-level verification and validation.

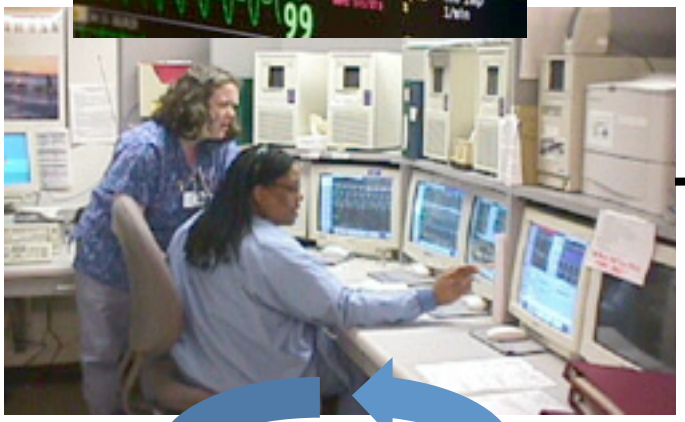
- Assembly is performed *after* deployment
- Assembler (hospital staff) **does not have** expert-level technical knowledge of components & system behavior
- **App developer** is responsible for overall system safety arguments
- Platform services (compatibility checks) assist in determining **at app launch time** if platform and attached devices satisfy requirements of app
- The app's directions for assembly of the platform constituted device are stated **only in terms of properties/ capabilities that are exposed on the interfaces** of the platform and devices.

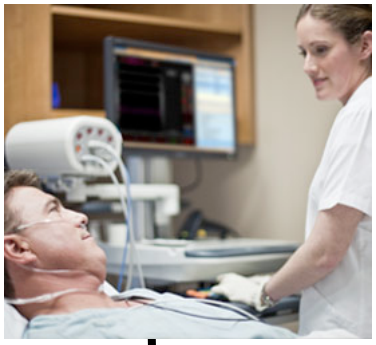
Certification

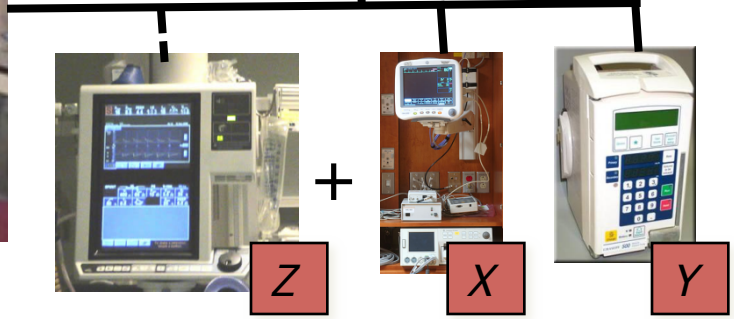
- In the U.S., FDA approves medical devices for specific use
  - Safety and effectiveness are assessed
  - Evaluation is process-based: ISO 9001 (quality management) and ISO 14971 (risk management)
  - Hazard analysis is key to approval
  - FDA's 510(k) requires "substantially equivalent" to devices on the market
- No certification of interconnectable MCPS
  - Currently, each collection of interconnected devices is a new medical device to be approved.

Current Regulatory Approach

Current regulation of integrated systems (e.g., central station monitors) requires "**pair-wise**" **clearance**: whenever a new type of device is added to the monitoring platform, the entire infrastructure must be re-cleared.







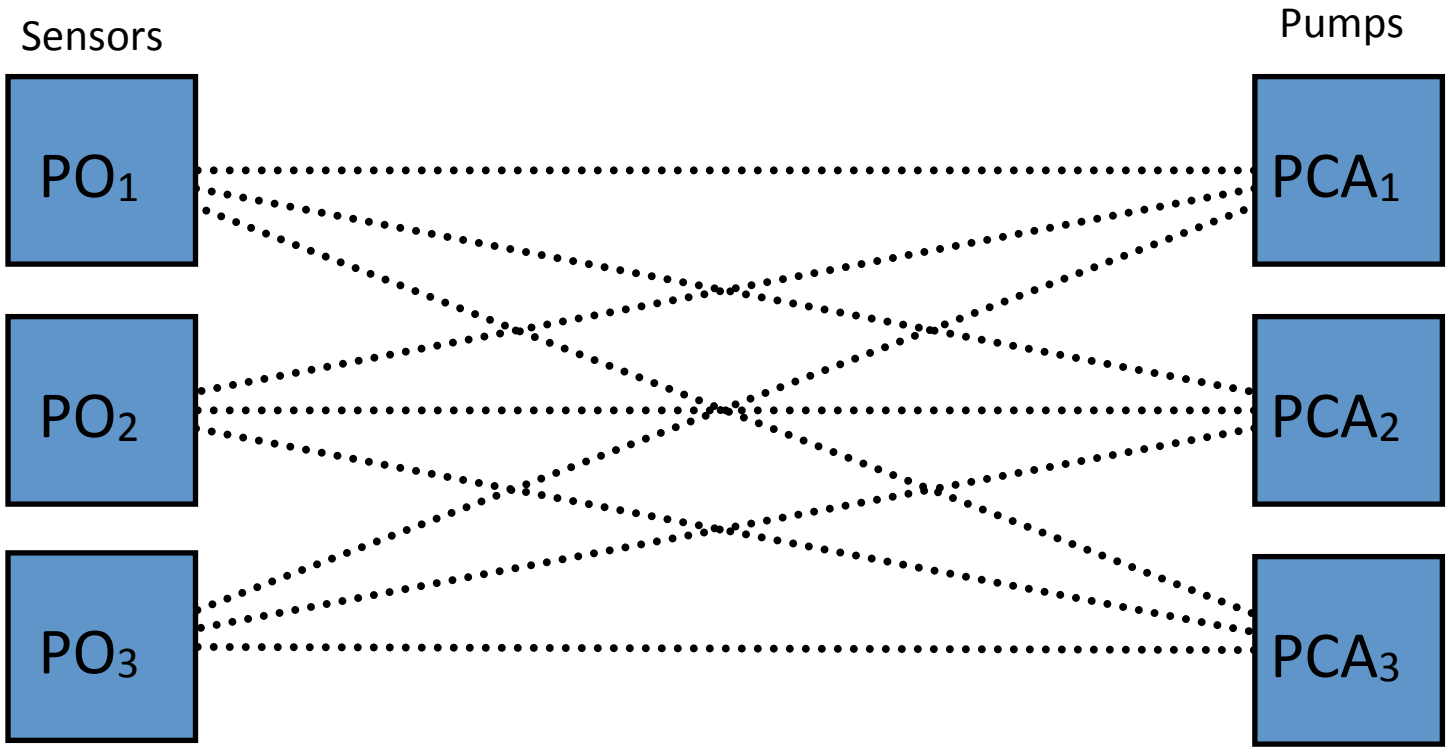
Regulatory Clearance

Assume monitoring system was originally developed, verified, and received regulatory clearance for devices of type X & Y.

In current regulatory approach, adding a new type of device (e.g., Z) typically causes the entire system to be re-submitted for regulatory clearance.

Pairwise Certification Complexity

Example "interoperable" device ecosystem 3 different (model/manufacturer) blood oxygen sensors, 3 different (model/manufacturer) PCA pumps:

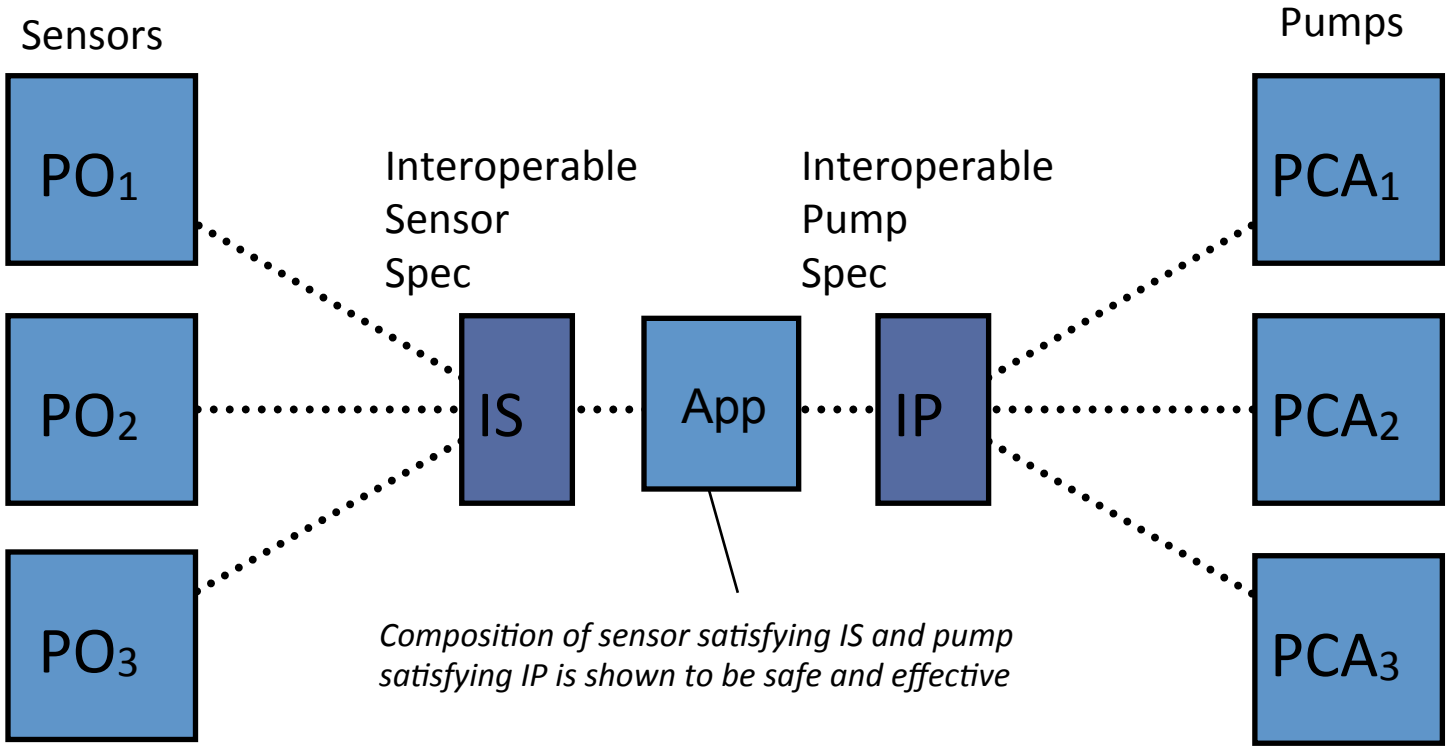


Each sensor must be approved or certified for use with each pump and vice versa. This is burdensome for manufacturers and regulators

Certification or approval relationship

Interface-based certification

Example "interoperable" device ecosystem 3 different (model/manufacturer) blood sugar sensors, 3 different (model/manufacturer) insulin pumps:



Each sensor (or pump) only needs certification or approval w.r.t. the interface spec. Additionally, the ecosystem can grow without forcing recertification (or re-approval) of previously analyzed devices

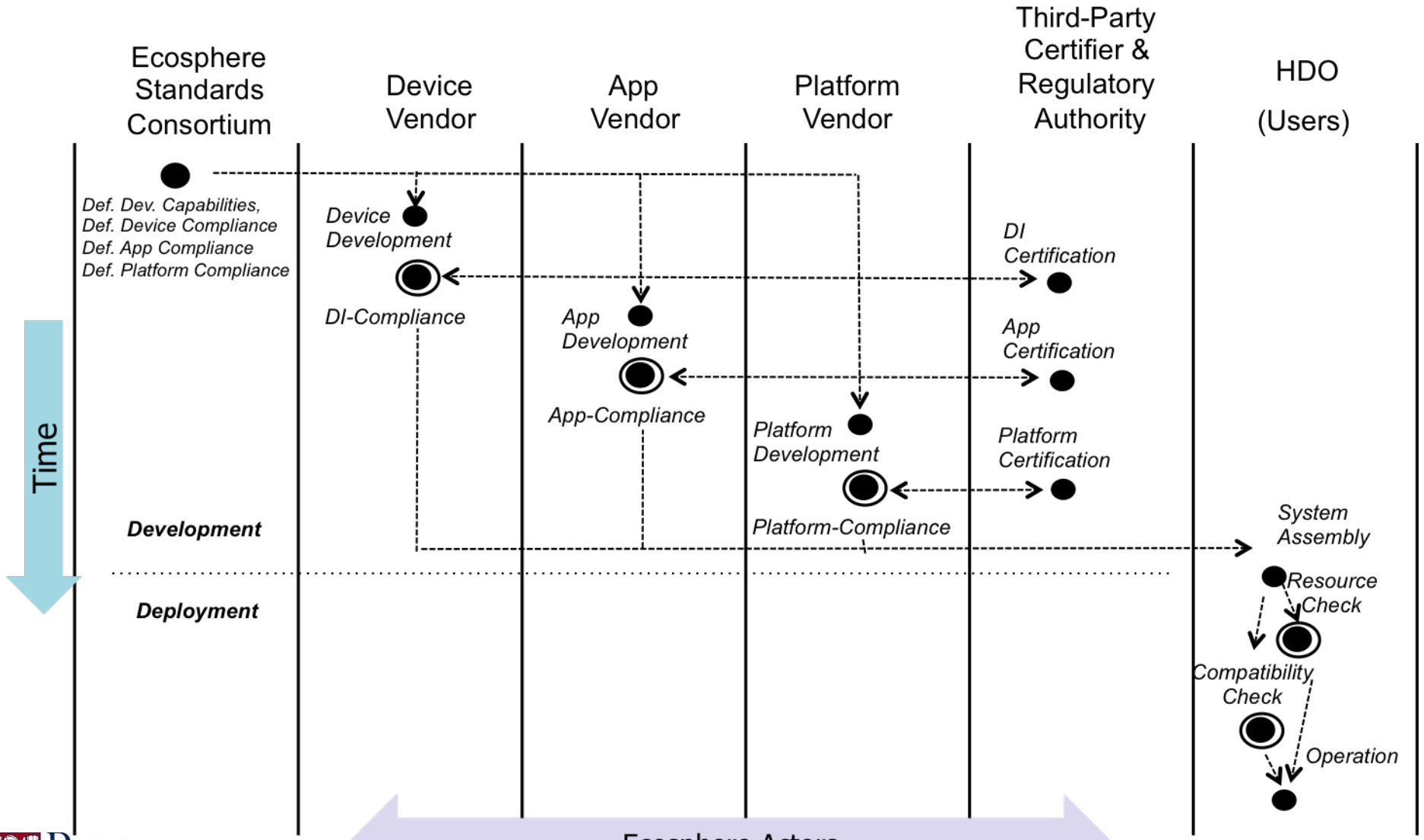
Certification or approval relationship

MEDICAL APP PLATFORM APPROACH

The Proposed Platform Approach

- Maintain a curated ecosphere of Devices, Apps, and Platforms
  - Apps define "the system":
    - Implement the clinical scenario algorithm
    - Specify required devices and their required behavior
    - App can be analyzed for safety using "models" as proxies for concrete devices and environment
  - Devices carry out required functions
    - Its (formal) capabilities model is captured by its "interface"
    - Adherence of a device to its capabilities needs to be "certified"
  - Platforms run the applications and facilitate system composition:
    - Ensures apps are only composed with compatible devices
    - Ensures app QoS requirements are met
- How does the ecosphere work?

VMD Ecosphere

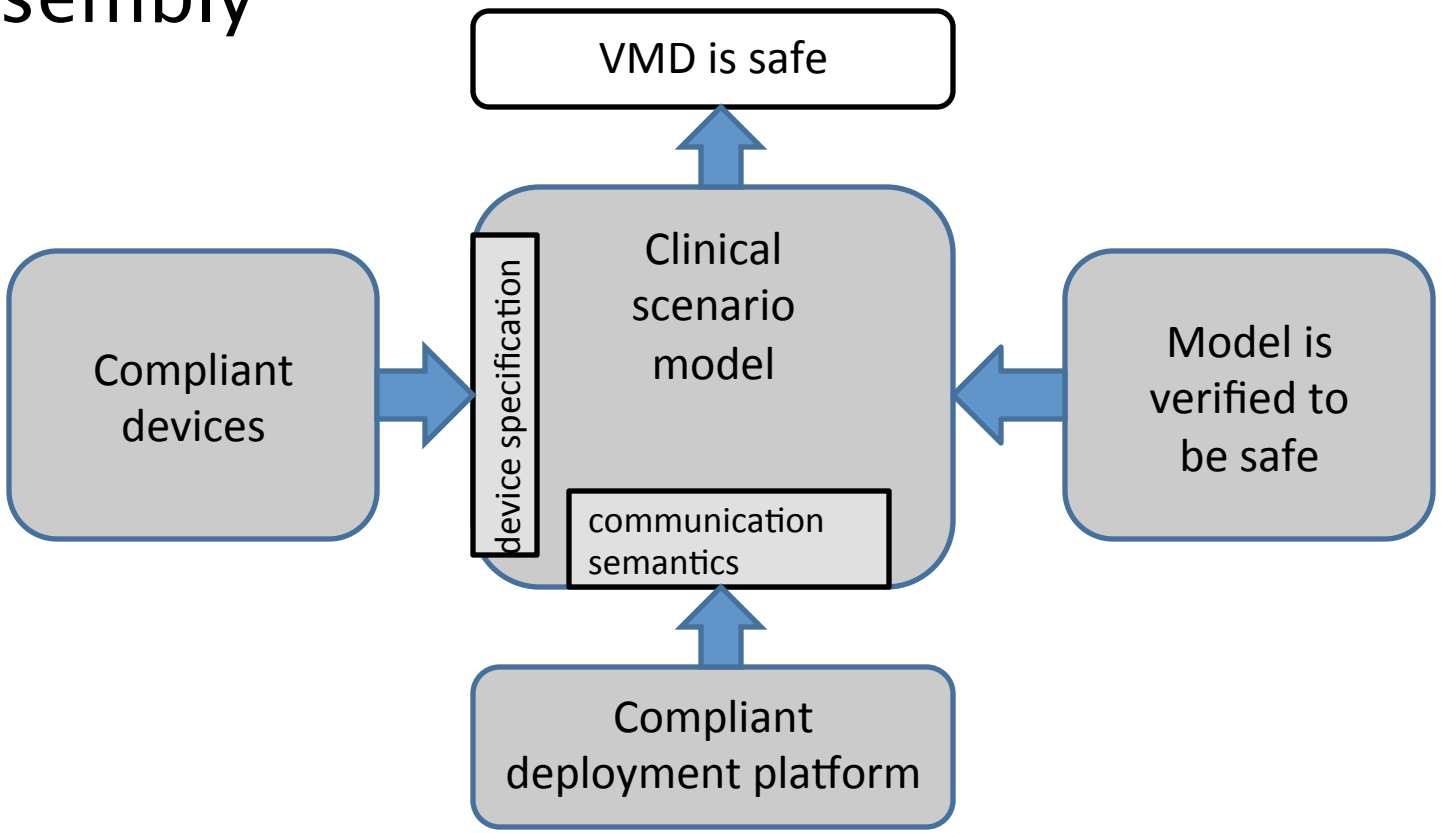


Model-based Safety Reasoning

- Why model-based reasoning (MBR)?
  - Each App defines a set of possible systems, each of which is an allowed combination of medical devices and platforms
  - App vendors would not be able to analyze all possible systems directly since
    - The number of device/platform combinations may be huge
    - New devices may be admitted after the App is certified
- What type of models?
  - Models must capture all the relevant behavior of allowed system combinations
  - The suitability of models and their analysis is dependent on:
    - Ecosphere certification/assurance processes
    - Platform quality / capabilities
    - Ecosphere notion of device / app compatibility
    - Intended use of the system
    - The safety properties being checked

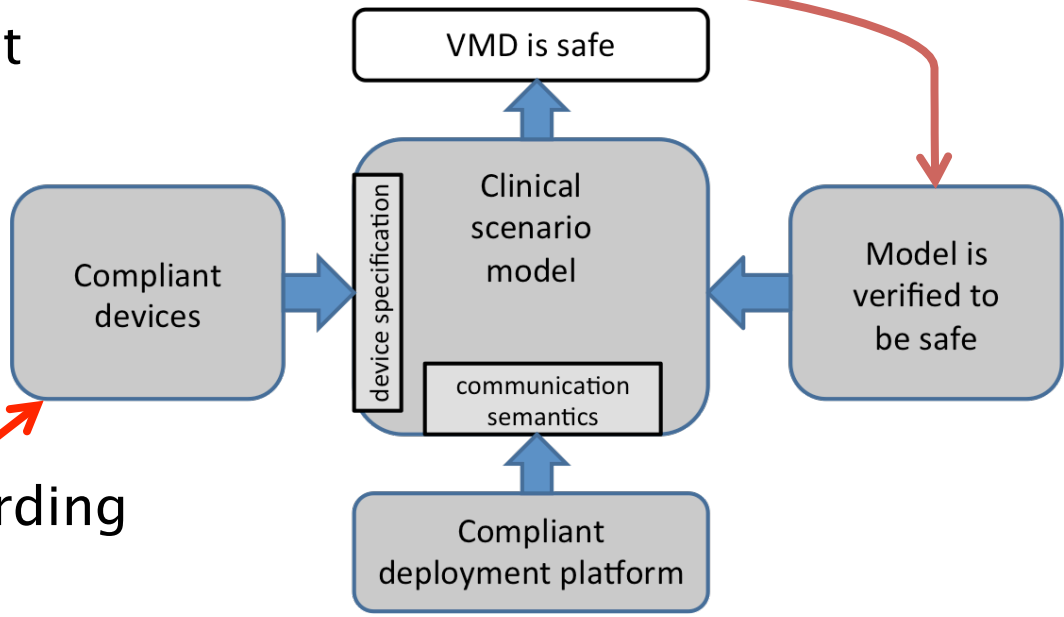
Safety Assurance for VMD

- Model-based analysis at design time
- Validation of modeling assumptions during assembly



Development and Instantiation

- Model the VMD and verify its safety properties
  - Models of constituent devices
  - Scenario logic
- Two assumptions:
  - Devices behave according to their models
  - Execution and communication semantics are guaranteed by the deployment platform



VMD App Modeling Language

- Modular
  - Clearly separate device specs from scenario logic
- Formal
  - Support verification
- Modal
  - Support alternative medical device/ implementations
- On-Demand Checking
  - Support checking devices at instantiation

Example Architectural Specification

vmd ClosedLoopPCA

devices

pcaPump : PCA

po : PulseOximeter

logicmodules

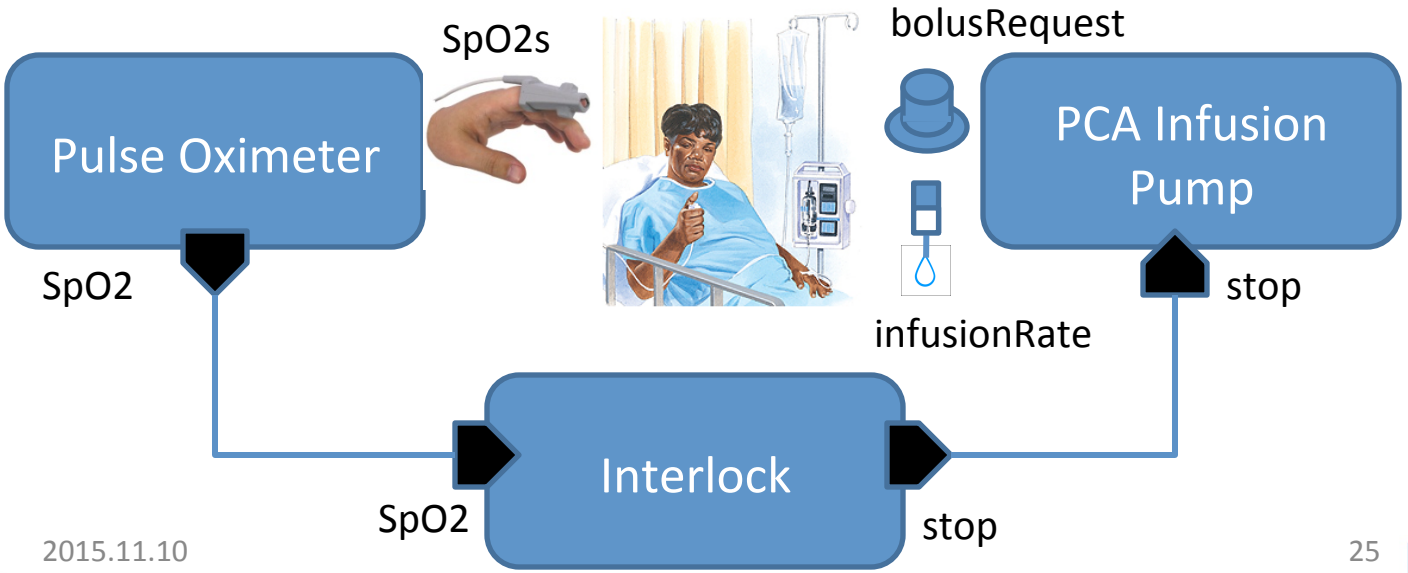
controller : PCATicketGenerator

dataflows

po.SpO2 →^{50ms} controller.SpO2

controller.ticket → ^{100ms} pcaPump.ticket

- Devices are specified separately from scenario logic
- Flows can be decorated with quality of service parameters



Modal Behavior Specification

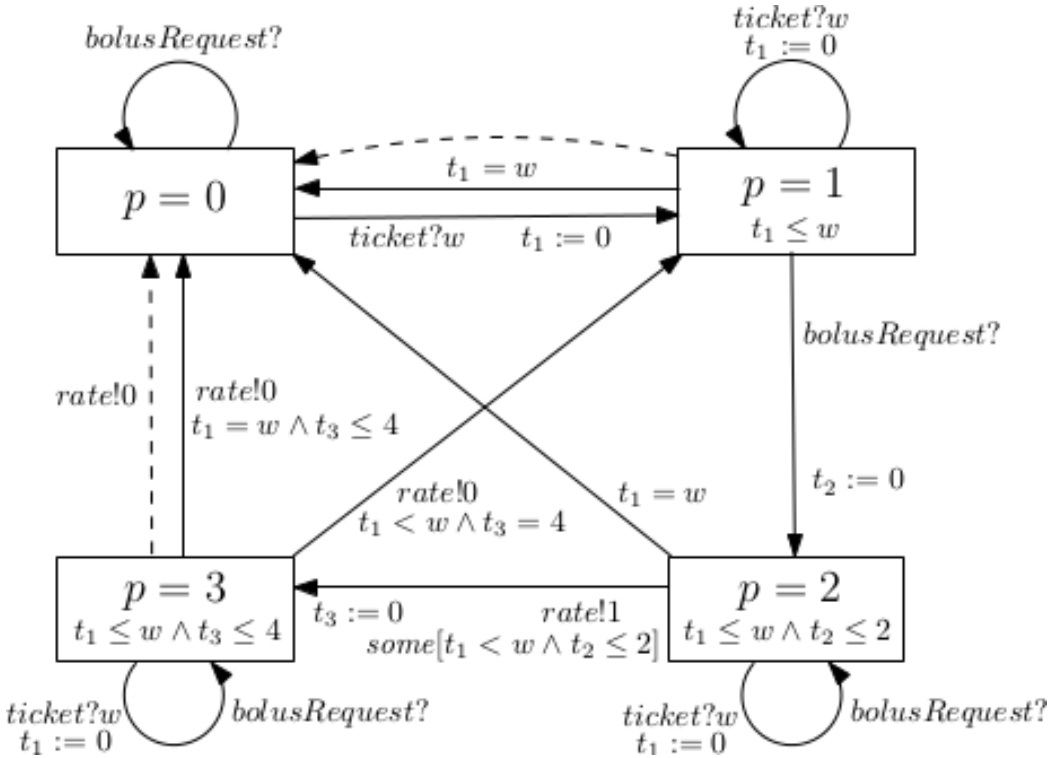
module PCA: **device**

interface

patient input: bolusRequest *event*

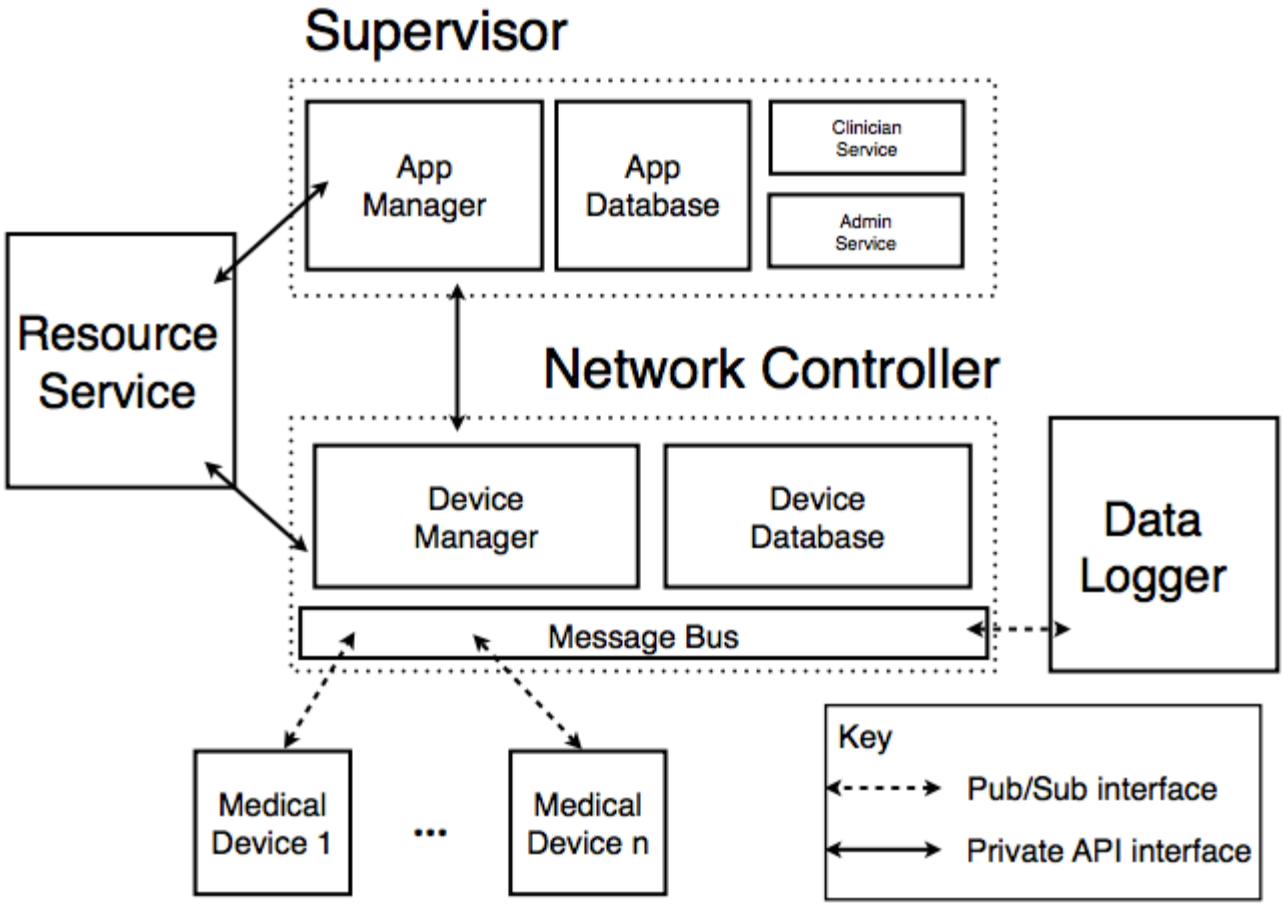
patient output: *rate continuous infusionRate*[0..2]

network input: ticket *event integer*[0..300]



MAP Architecture

- Builds on ICE ASTM 2761 and IEEE/ISO 11073 manager-mediated communication



MAP Design (Supervisor)

- Execution environment for the logic of a VMD
  - Checks compatibility between VMD requirements and devices
  - Orchestrates VMD lifecycle
    - Device / VMD Coupling
    - Operation
    - Exceptions
    - Shutdown
- MDCF, OpenICE
  - Prototype implementations at KSU and MGH/ CIMIT

MAP Design (Network Controller)

- Provides communications abstraction for VMD
  - Pub / Sub with timing guarantees (end to end latency)
  - Isolation (data/time) between different data-flows
- Admits / Tracks Devices onto network
  - Per device authentication
  - Records device capabilities
- Real-Time Message Bus
  - Prototype at Penn using OpenFlow

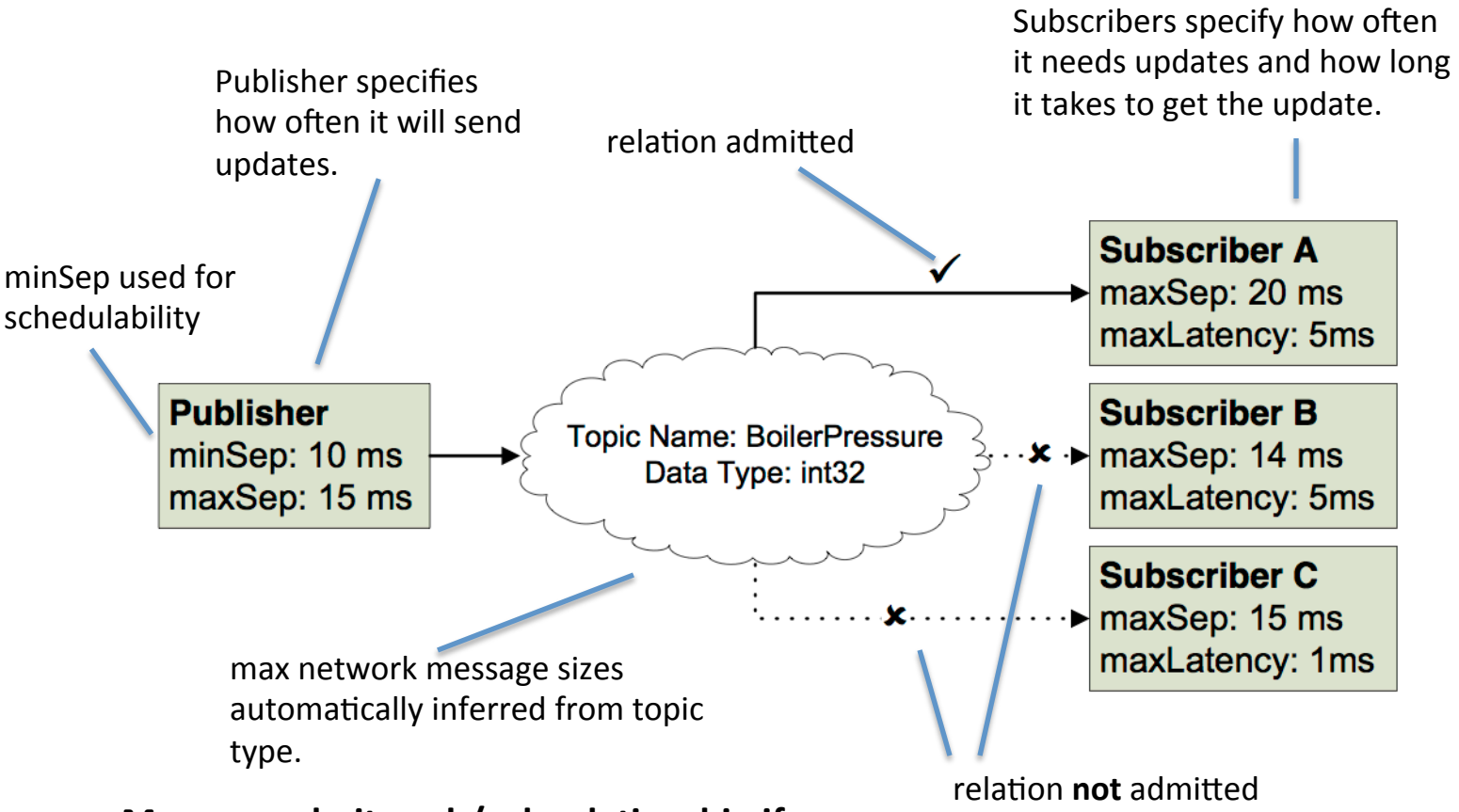
MAP Design (Resource Manager)

- VMD's timing characteristics depend on underlying platform.
  - E.g. logical message passing latency between device / app depends on network transmission time and processing time on the supervisor
  - The resource manager must orchestrate resource scheduling (network, CPU, etc.) to ensure the logical timing requirements of the VMD are met.

MIDAS: MIDdleware ASsurance Substrate

- Requirements
  - Dynamic reconfiguration
    - plug in and out of medical devices
    - addition and deletion of clinical Apps
  - QoS isolation, real-time guarantee
  - Security
  - Implementation using Openflow switches

QoS Example



Resource Manager admits pub/sub relationship if:

- **The publisher *maxSep* <= subscriber *maxSep*.**
- **The resource manager can guarantee the end to end latency.**

Other Challenges

- Data Logger, Integrated Alarm System
- Security and Privacy
- Third-party Certification
- Human-in-the-loop
- Assurance Cases

Assume-Guarantee Safety Assurance

- **Goal**: guarantee that $P(A) (||_{j=1...n} D_j) || E \models \phi$

> The execution of App A on the platform P, denoted by P(A), together with the assembly of medical devices D_1, …, D_n in the environment E satisfies the safety property ϕ.

- Entities in the assume-guarantee reasoning rule

	Model	Software / Specification	Physical Embodiment
App	A^m	A	P(A)
Interface	AI_j^m (j=1…n)	AI_j (j=1…n)	
Devices		DI_j (j=1…n)	D_j (j=1…n)
Platform	P^m		P
Environment	E^m		E

Assume-Guarantee Reasoning Rule

	Model	Software / Specification	Physical Embodiment
App	A^m	A	P(A)
Interface	AI_j^m (j=1…n)	AI_j (j=1…n)	
Devices		DI_j (j=1…n)	D_j (j=1…n)
Platform	P^m		P
Environment	E^m		E

① $A^m \simeq A$

② $AI_j^m \simeq AI_j$

③ $P^m \simeq P$

④ $E^m \simeq E$

App developers need to assure that models are faithful to the implementation/platform/environment.

Assume-Guarantee Reasoning Rule

	Model	Software / Specification	Physical Embodiment
App	A^m	A	$P(A)$
Interface	AI_j^m $(j=1…n)$	AI_j $(j=1…n)$	
Devices		DI_j $(j=1…n)$	D_j $(j=1…n)$
Platform	P^m		P
Environment	E^m		E

① $A^m \simeq A$

② $AI_j^m \simeq AI_j$

③ $P^m \simeq P$

④ $E^m \simeq E$

⑤ $A^m (||_{j=1…n} AI_j^m) \,||\, P^m \,||\, E^m \models \phi$

App developers use model checking to verify that the composed system model satisfies the safety property.

Assume-Guarantee Reasoning Rule

	Model	Software / Specification	Physical Embodiment
App	A^m	A	$P(A)$
Interface	AI_j^m $(j=1…n)$	AI_j $(j=1…n)$	
Devices		DI_j $(j=1…n)$	D_j $(j=1…n)$
Platform	P^m		P
Environment	E^m		E

① $A^m \simeq A$

② $AI_j^m \simeq AI_j$

③ $P^m \simeq P$

④ $E^m \simeq E$

⑤ $A^m (||_{j=1…n} AI_j^m) || P^m || E^m \vDash \phi$

---

①-⑤ $A (||_{j=1…n} AI_j) || P || E \vDash \phi$

Assume-Guarantee Reasoning Rule

	Model	Software / Specification	Physical Embodiment
App	A^m	A	$P(A)$
Interface	AI_j^m $(j=1…n)$	AI_j $(j=1…n)$	
Devices		DI_j $(j=1…n)$	D_j $(j=1…n)$
Platform	P^m		P
Environment	E^m		E

①-⑤ $A (||_{j=1…n} AI_j) || P || E \vDash \phi$

⑥ $DI_j \simeq D_j$

Device manufacturers need to assure that a device's capability specification conforms to its actual behavior.

Assume-Guarantee Reasoning Rule

	Model	Software / Specification	Physical Embodiment
App	A^m	A	$P(A)$
Interface	AI_j^m $(j=1…n)$	AI_j $(j=1…n)$	
Devices		DI_j $(j=1…n)$	D_j $(j=1…n)$
Platform	P^m		P
Environment	E^m		E

①-⑤ $A (||_{j=1…n} AI_j) || P || E \vDash \phi$

⑥ $DI_j \simeq D_j$

⑦ $AI_j \simeq DI_j$ (or DI_j refines AI_j)

The compatibility between the App's interface about the required device specification and the actual devices' capability needs to be checked, e.g. by third-party certifiers.

Assume-Guarantee Reasoning Rule

	Model	Software / Specification	Physical Embodiment
App	A^m	A	$P(A)$
Interface	AI_j^m $(j=1…n)$	AI_j $(j=1…n)$	
Devices		DI_j $(j=1…n)$	D_j $(j=1…n)$
Platform	P^m		P
Environment	E^m		E

①-⑤ $A\ (||_{j=1…n}\ AI_j)\ ||\ P\ ||\ E \vDash \phi$

⑥ $DI_j \simeq D_j$

⑦ $AI_j \simeq DI_j$

---

①-⑦ $A\ (||_{j=1…n}\ D_j)\ ||\ P\ ||\ E \vDash \phi$

Assume-Guarantee Reasoning Rule

	Model	Software / Specification	Physical Embodiment
App	A^m	A	P(A)
Interface	AI_j^m (j=1…n)	AI_j (j=1…n)	
Devices		DI_j (j=1…n)	D_j (j=1…n)
Platform	P^m		P
Environment	E^m		E

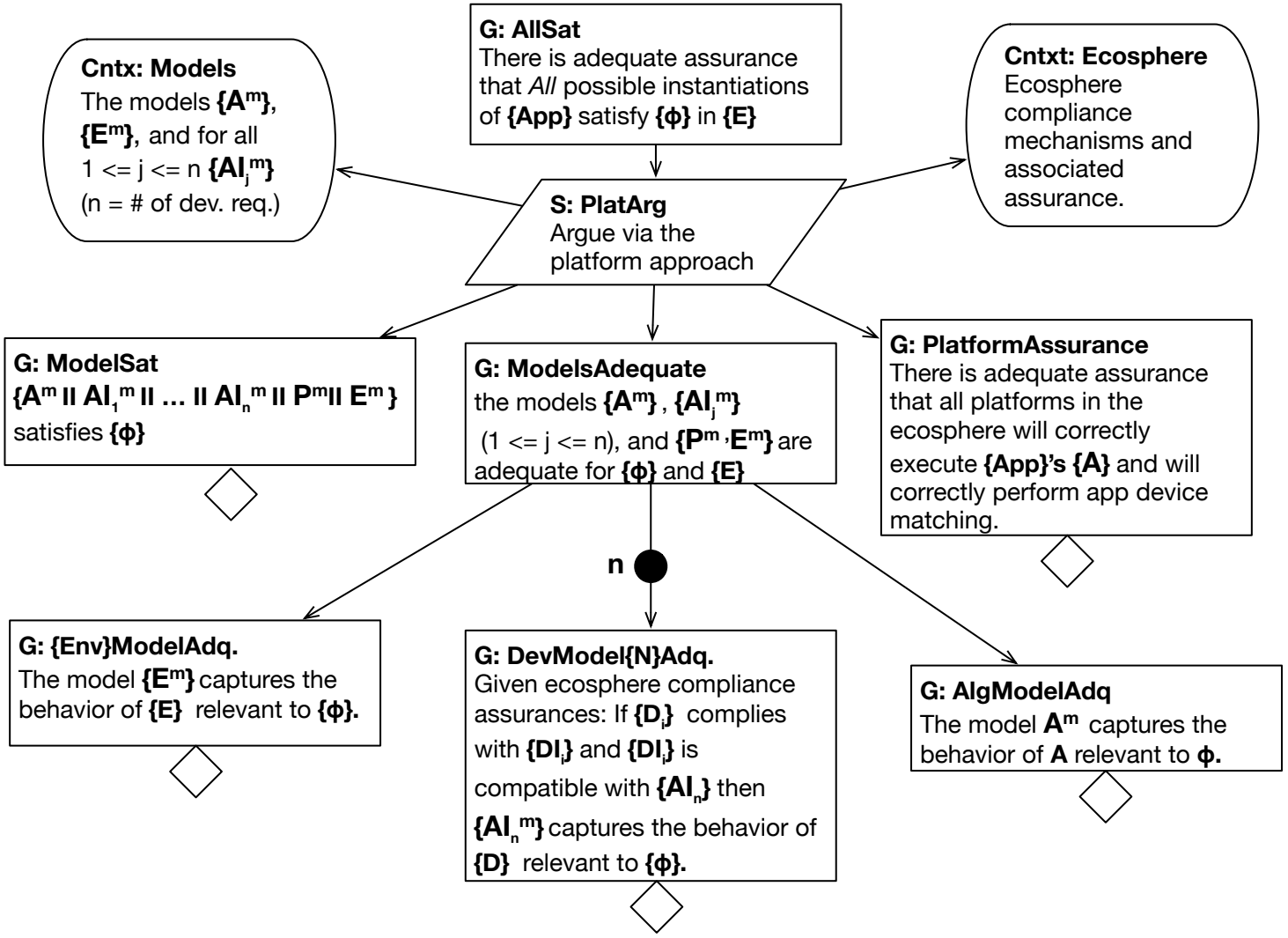
①–⑦ $A (||_{j=1\ldots n} D_j) || P || E \vDash \phi$

⑧ $A || P \simeq P(A)$ /* P(A) means Dj's are compatible for A */

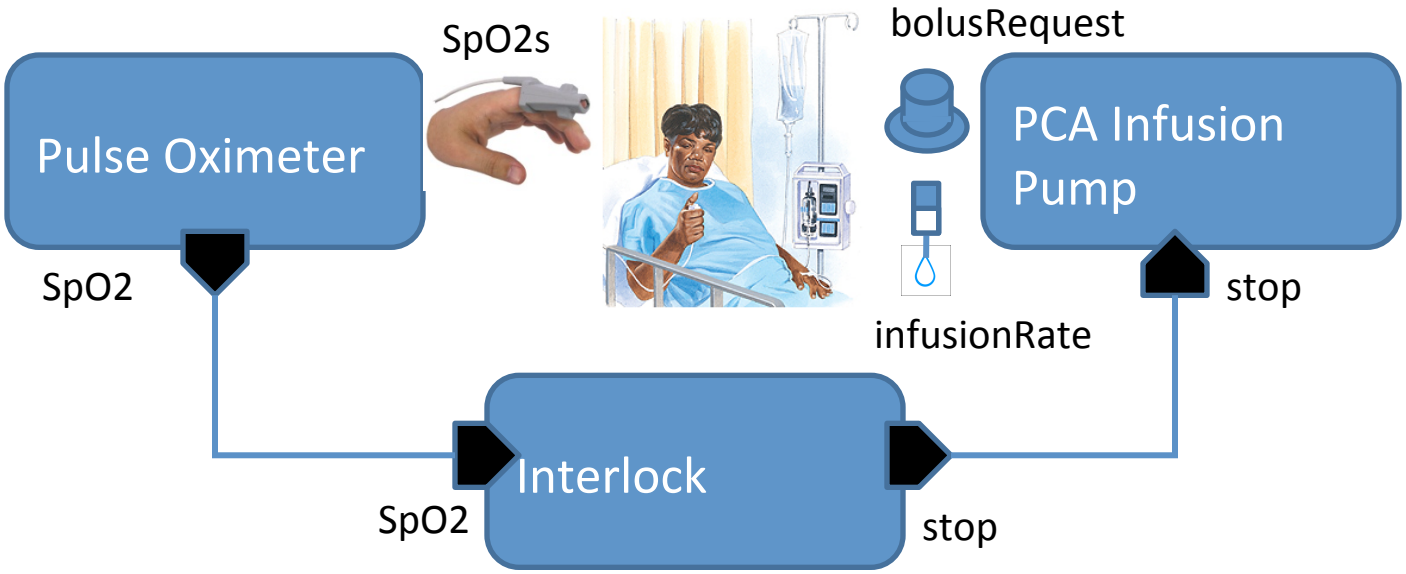
①-⑧ $P(A) (||_{j=1\ldots n} D_j) || E \vDash \phi$

The execution of App A on the platform P, denoted by P(A), together with the assembly of medical devices D_1, …, D_n in the environment E satisfies the safety property ϕ.

Proposed Assurance Argument Pattern

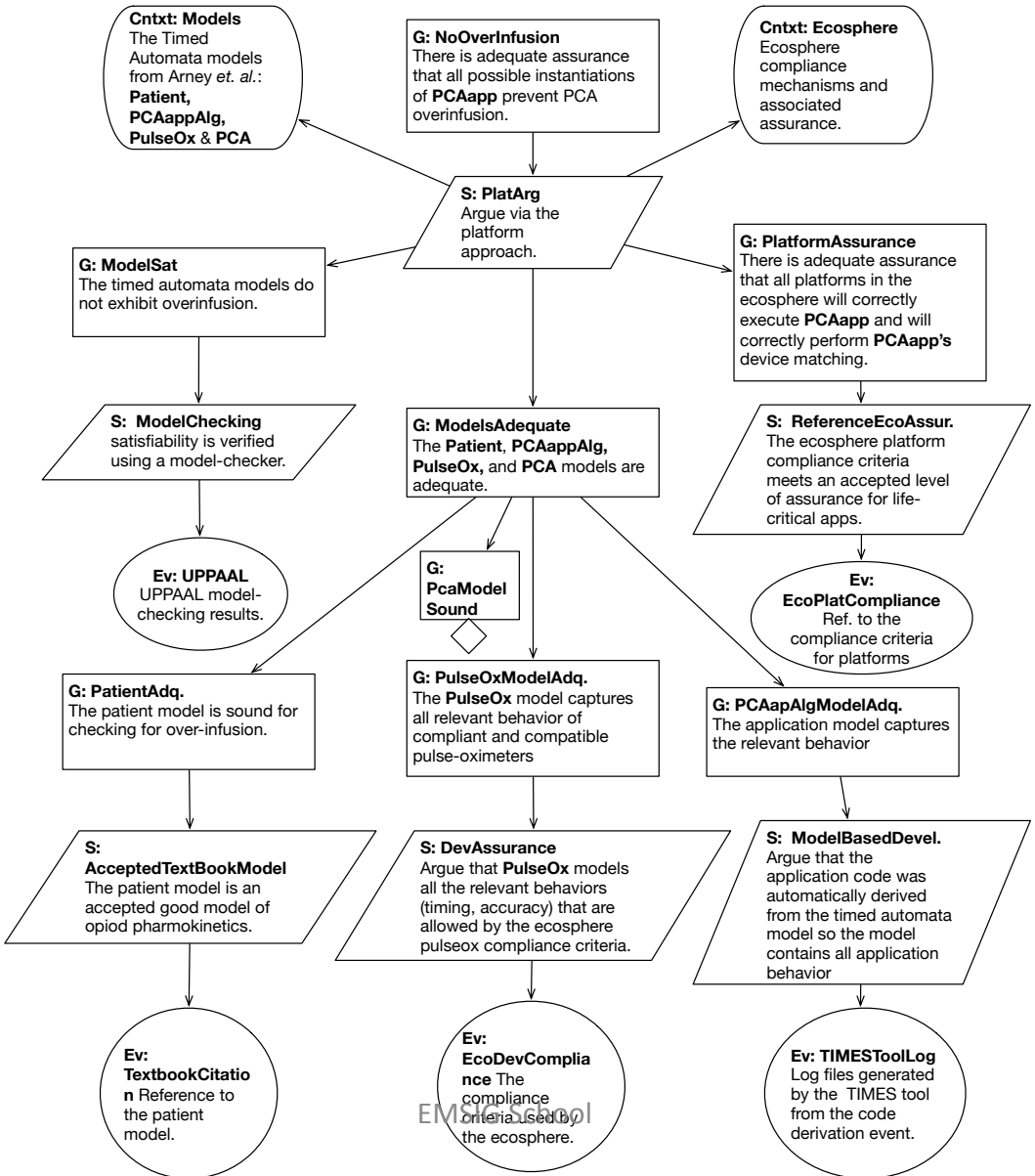


Case Study: PCA Control App



vmd ClosedLoopPCA
- **devices**
  - pcaPump : PCA
  - po : PulseOximeter
- **logicmodules**
  - controller : PCATicketGenerator
- **dataflows**
  - po.SpO2 →^{50ms} controller.SpO2
  - controller.ticket → ^{100ms} pcaPump.ticket

Example Assurance Case



Summary

- Propose an assurance argument pattern to assist the safety analysis of plug & play MCPS that consist of
  - a set of medical devices
  - an App (i.e., a software component that coordinates the medical devices for a specific clinical scenario),
  - and a platform that runs the App
- Present an assume-guarantee compositional proof rule/ framework for plug & play MCPS and show how it can be used to as a logical basis for the proposed pattern
  - model-based analysis at design time
  - validation of modeling assumptions during assembly
- [SAFECOMP 2015] Towards Assurance for Plug & Play Medical Systems, Andrew L. King, Lu Feng, Sam Procter, Sanjian Chen, Oleg Sokolsky, John Hatcliff, Insup Lee, SAFECOMP, Sept 2015.

Community Efforts

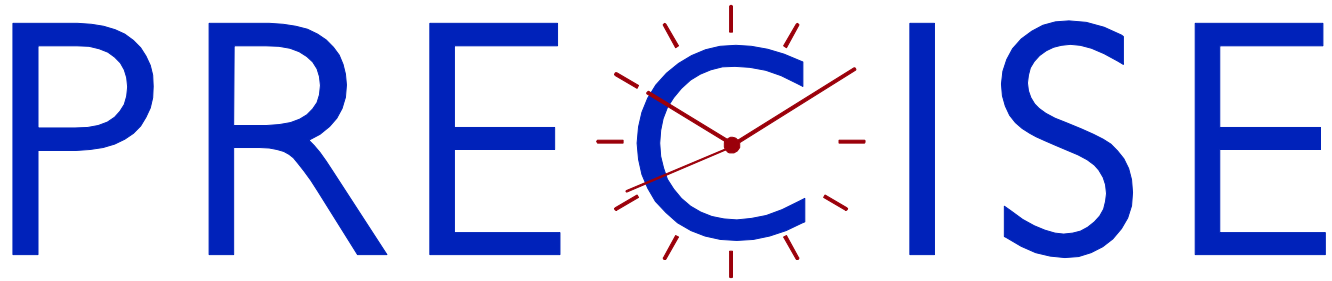
- MDPnP Program
  - ICE standard, clinical scenarios
  - www.mdpnp.org
- MDISWG (Medical Device Interoperability Safety Workhing Group)
  - Pre-Ide to FDA
  - http://mdpnp.org/MD_PnP_Program___MDISWG.html
- AAMI/UL 2800 Joint Committee Working Groups
- Middleware for MCPS
  - OpenICE, MGH/CIMIT
  - MDCF, KSU/UPenn
  - MIDAS, UPenn
  - DDS based, Docbox

Selected References

- Platform-Specific Code Generation from Platform-Independent Timed Models, BaekGyu Kim, Lu Feng, Oleg Sokolsky and Insup Lee, IEEE RTSS, Dec 2015. (to appear)
- A Data-Driven Behavior Modeling and Analysis Framework for Diabetic Patients. Sanjian Chen, Lu Feng, Michael Rickels, Amy Peleckis, Oleg Sokolsky and Insup Lee. IEEE Int. Conf. on Healthcare Informatics (ICHI), Oct. 2015.
- An Intraoperative Glucose Control Benchmark for Formal Verification. Sanjian Chen, Matthew O'Kelly, James Weimer, Oleg Sokolsky, Insup Lee. ADHS (IFAC Conf. on Analysis and Design of Hybrid Systems), Oct 14-16, 2015.
- Towards Assurance for Plug & Play Medical Systems, Andrew L. King, Lu Feng, Sam Procter, Sanjian Chen, Oleg Sokolsky, John Hatcliff, Insup Lee, SAFECOMP, Sept 2015.
- A Safety Argument Strategy for PCA Closed-Loop Systems: A Preliminary Proposal, Lu Feng, et al., MedCPS Workshop, April 14, 2014.
- Distributed aspects of the artificial pancreas, S.D. Patek, S. Chen, P. Keith-Hynes, I. Lee, *51st Annual Allerton Conference on Communication, Control, and Computing*, Oct 2013.
- Model-Driven Safety Analysis of Closed-loop Medical Systems, Miroslav Pajic, et al., In *IEEE Transactions on Industrial Informatics*, 2013.
- A Model-Based I/O Interface Synthesis Framework for the Cross-Platform Software Modeling, BaekGyu Kim, et al., *IEEE RSP,* October 2012.
- Evaluation of a Smart Alarm for Intensive Care using Clinical Data. Andrew King, et al. In *34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'12)*, September, 2012.
- A Safety Case Pattern for Model-Based Development Approach, Anaheed Ayoub, Baek-Gyu Kim, Insup Lee and Oleg Sokolsky. *NASA Formal Methods Symposium (NFM)*, April 2012.
- Challenges and Research Directions in Medical Cyber-Physical Systems, Insup Lee, et al. In *Special Issue on Cyber-Physical Systems, Proceedings of the IEEE*, Volume 100, Issue 1, pp.75-90, January 2012.
- Safety-Assured Development of the GPCA Infusion Pump Software, BaekGyu Kim, et al., *EMSOFT 2011*, October 2011.

- Additional related papers at www.cis.upenn.edu/~lee/home/publications/

Thank You!
Questions?



http://precise.seas.upenn.edu