Towards a Logical Foundation for Assurance Arguments for Plug & Play Systems

Lu Feng, Andrew King, Insup Lee, Oleg Sokolsky

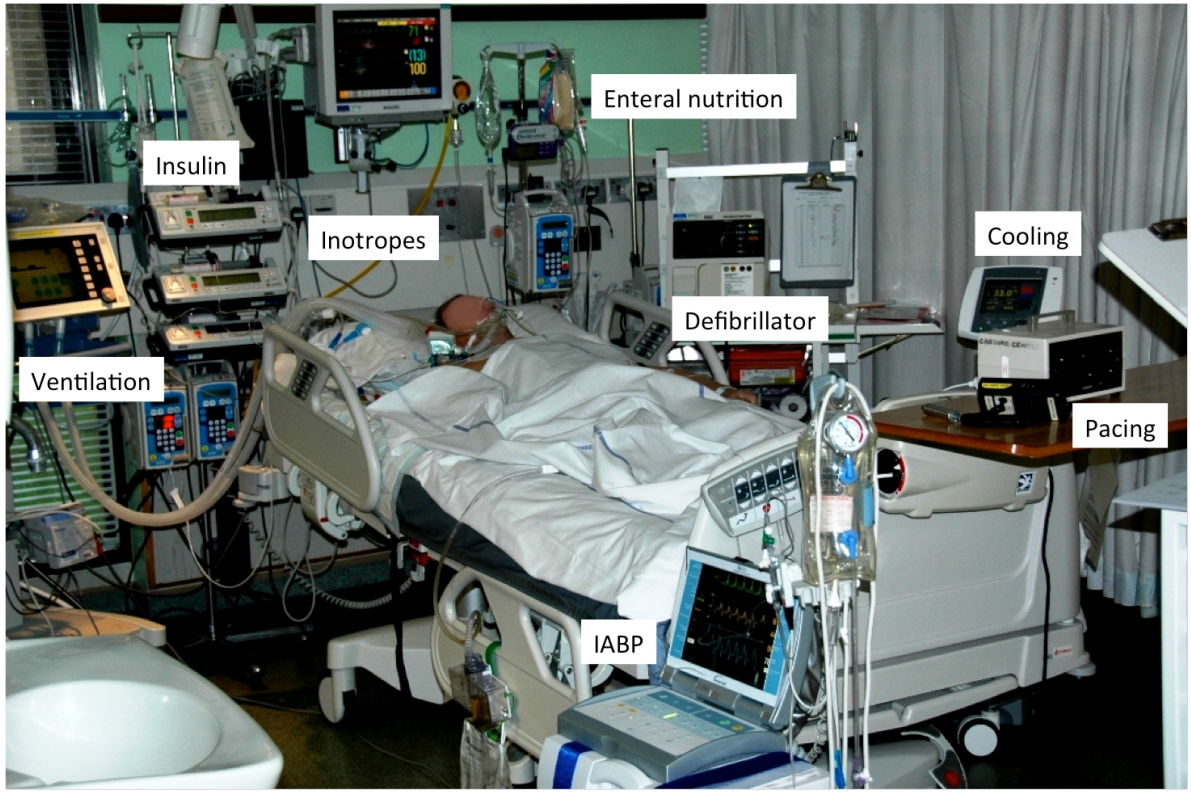
PRECISE Center
School of Engineering and Applied Science
University of Pennsylvania

VeriSure: Verification and Assurance Workshop at CAV 2015
San Francisco, 18 July 2015

Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Medical Device Interoperability



Problem: little to no integration of devices with each other

- Humans must automate even simple clinical workflows
- Unnecessary burdens placed on human caregivers
- Few opportunities for "sensor fusion" (better alarms and diagnostics)

Clinical Scenario: Laser Surgery / Ventilator

- Doctors enforce the following invariant
  - If laser = on then oxygen = off
  - If patient's SpO2 < 95 then oxygen = on
- Systems of Systems approach
  - Let devices communicate and automate safety invariant enforcement

Benefits of Medical Device Interoperability

- Interoperable medical devices can self-coordinate
  - Provide continuous monitoring
  - Handle routine tasks and respond to obvious problems
  - Alert caregivers in more serious cases; reduce alarm fatigue
  - Physiological closed-loop control in many cases

Current

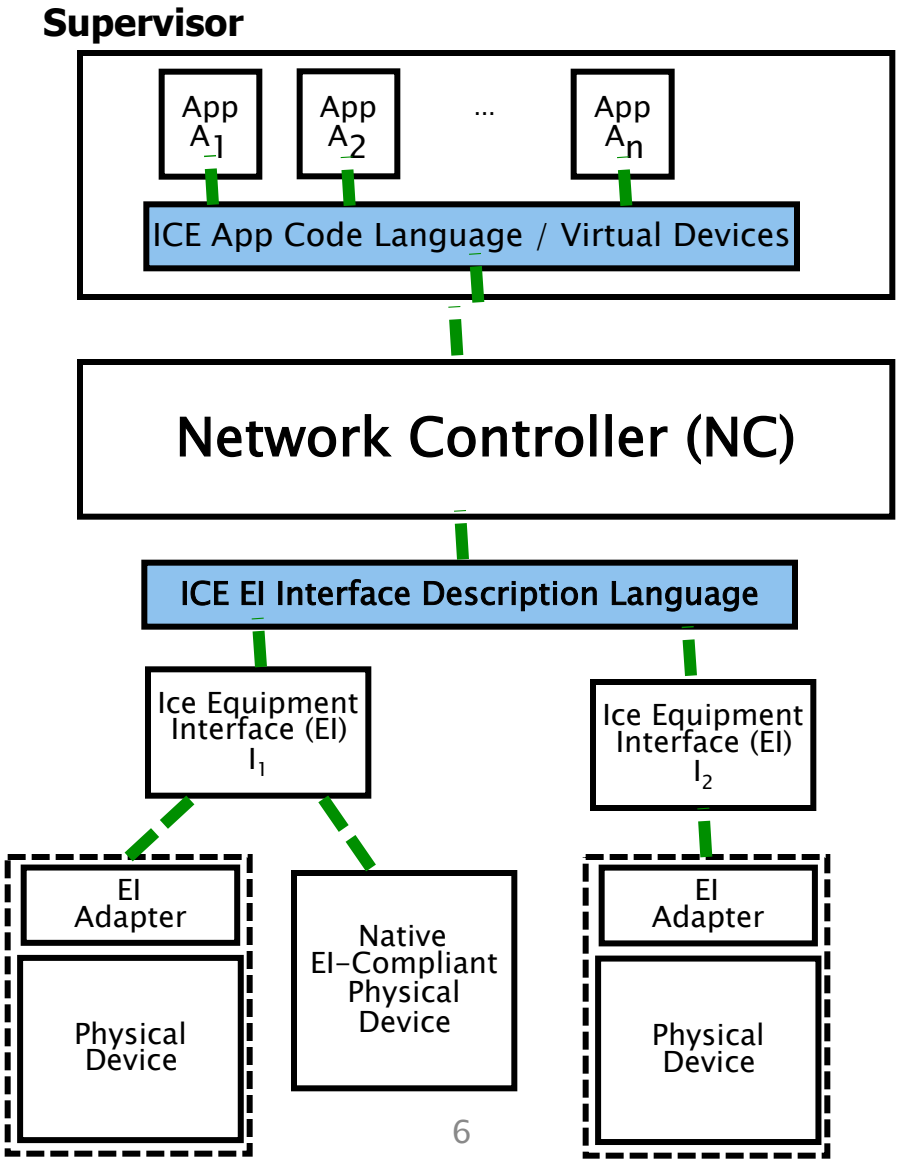
Future

Medical Device Plug-and-Play Open Systems

- Medical Device Plug-and-Play (MD PnP)
  - Interoperable medical devices based on plug-and-play
  - Vender neutrality based on open medical device interfaces
  - www.mdpnp.org

- Emerging Interoperability Standards
  - ASTM Standard F2761-2009 for Integrated Clinical Environment (ICE) defines a high-level architecture and functional concept
  - The ICE architecture standard is the focal point for FDA's evaluation of MAP (Medical App Platform) concepts in future medical systems

ICE Architecture



Virtual Medical Device (VMD)

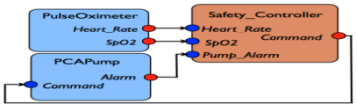
- **MD PnP** enables the concept of **VMD**
  - A set of medical devices coordinating over a network for clinical scenario

Device Coordination Algorithm

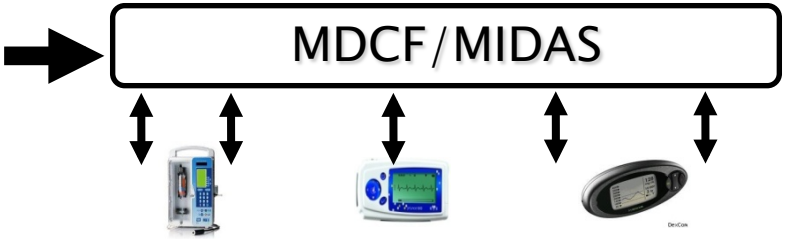
Medical Device Types

Virtual Medical Device (VMD)

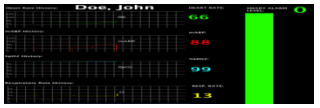
- VMD does not physically exist until instantiated at hospitals
- The Medical Device Coordination Framework (MDCF)
  - Our prototype middleware for managing the correct composition of medical devices into VMD.



Clinician selects appropriate VMD

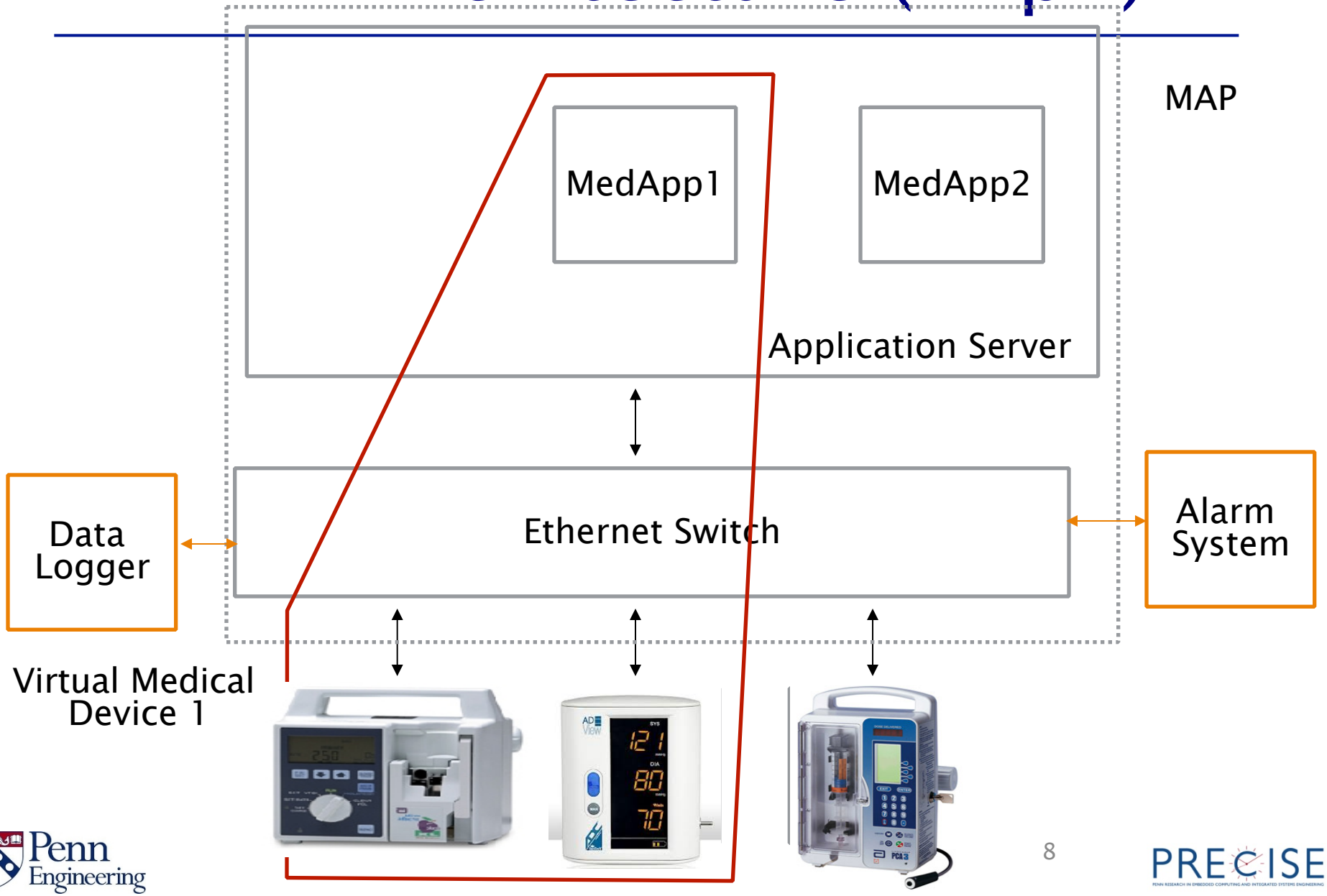


MDCF binds appropriate devices into VMD instance

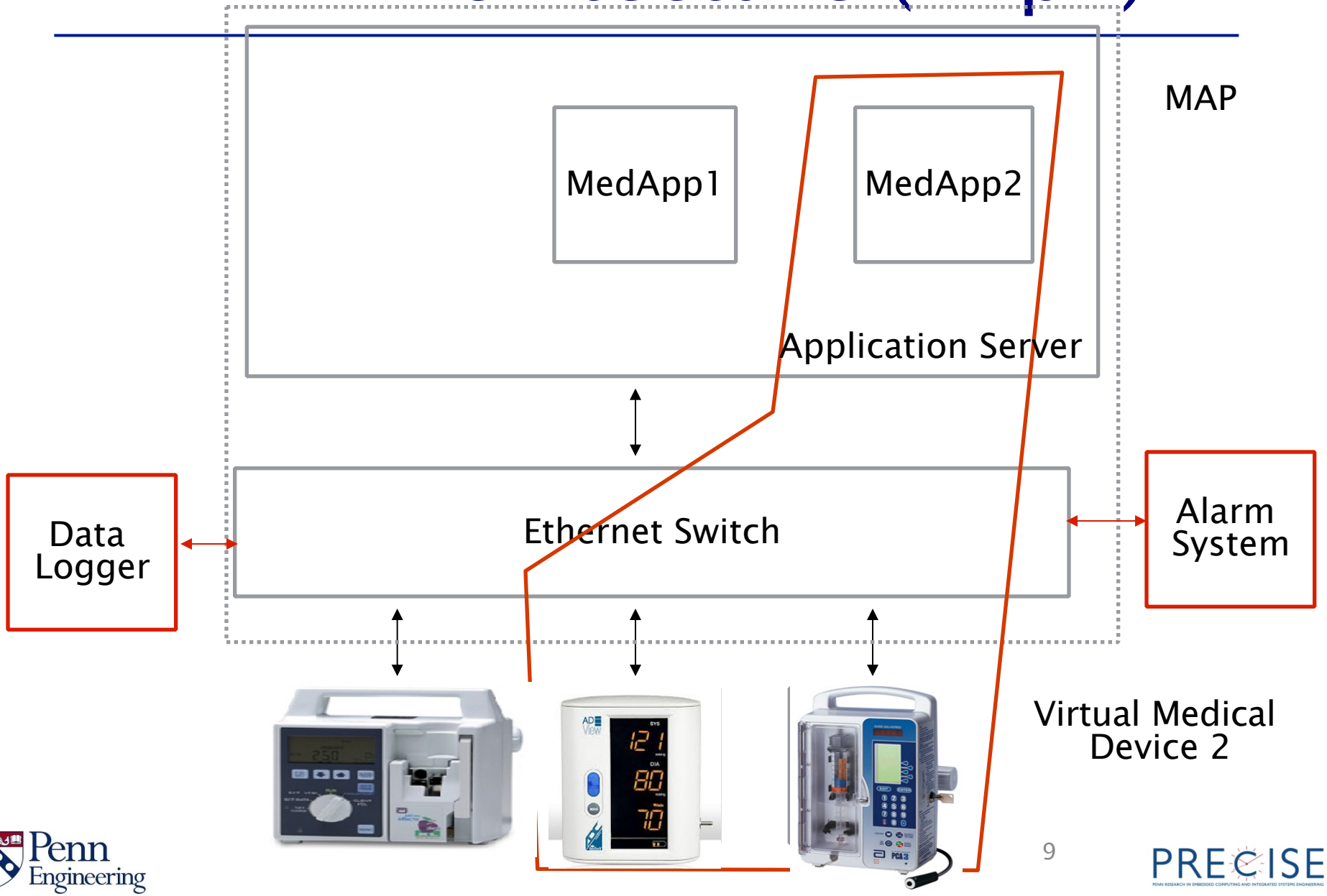


MDCF displays VMD GUI for clinician

VMD Architecture (Impl.)



VMD Architecture (Impl.)

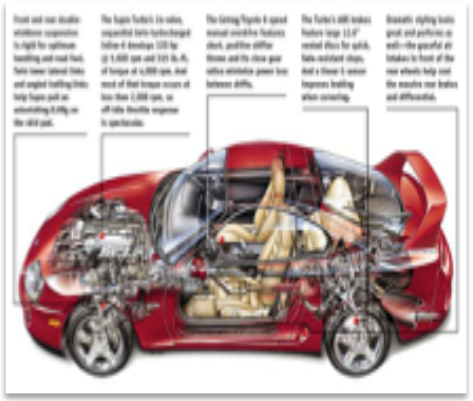


Safety Assurance Challenge for VMD

- The new system integration paradigm of VMD has serious implications for safety assurance, where the traditional approach won't scale

- Traditional safety critical systems
  - fixed function
  - designed and integrated by a single system integrator

Aerospace

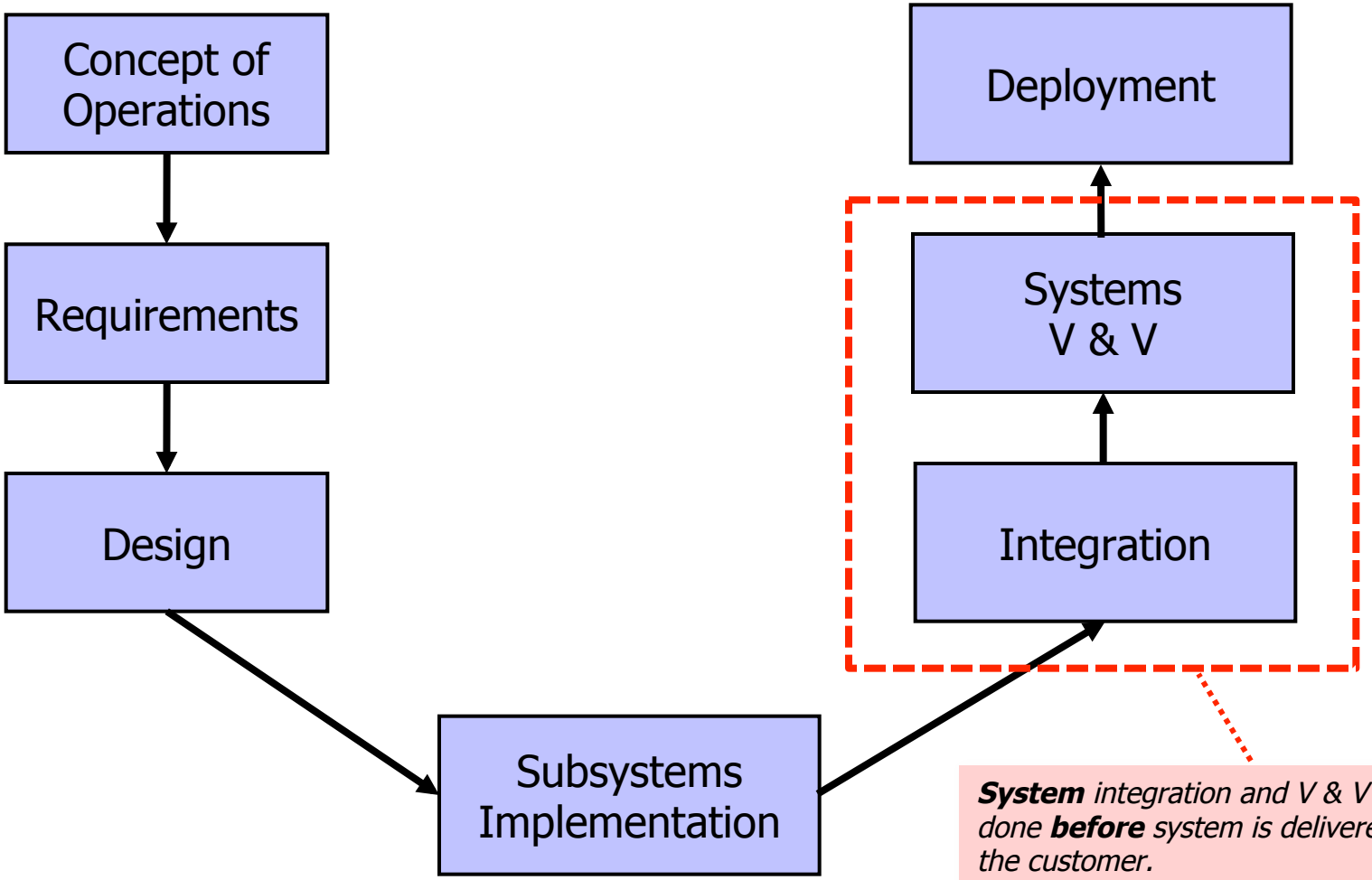


Automotive

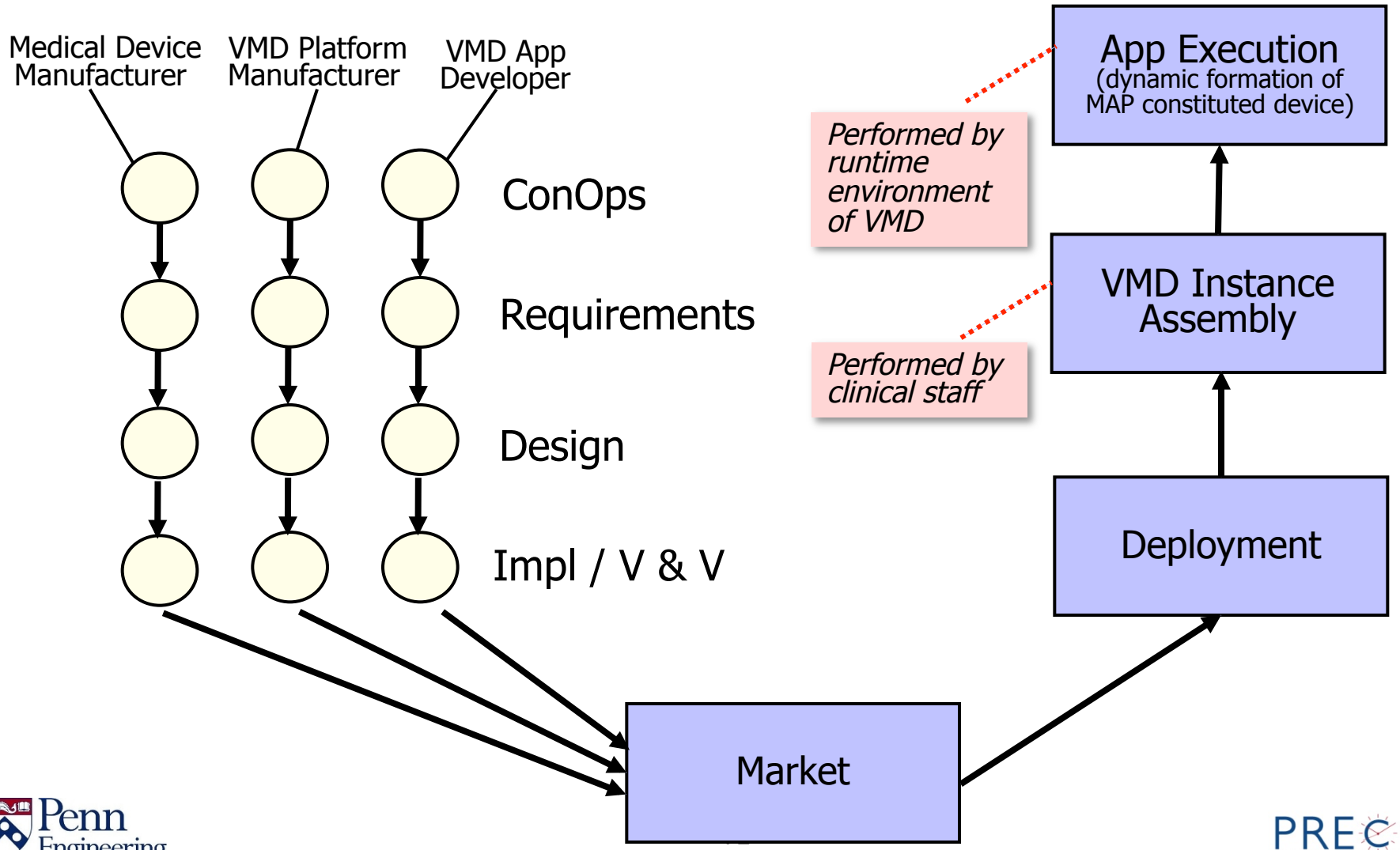
Nuclear

Traditional System Integration

- End to end process managed by prime contractor



VMD Development & Assembly



VMD Characteristics

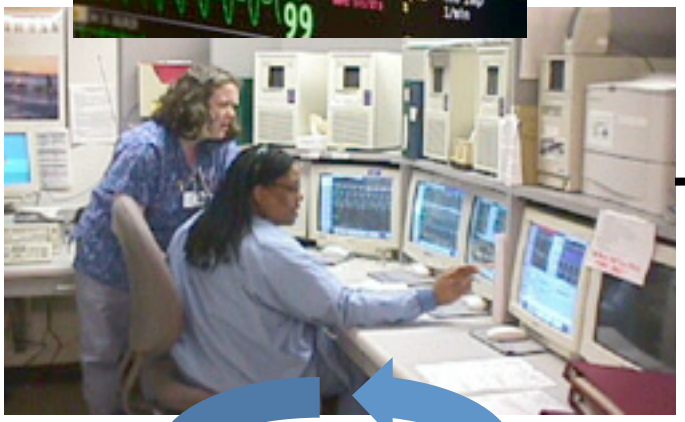
- There is no prime contractor that is responsible for VMD integration and system-level V&V
  - Assembly is performed after deployment
  - Assembler (hospital staff) **does not have** expert-level technical knowledge of components & system behavior
  - App developer is responsible for overall system 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

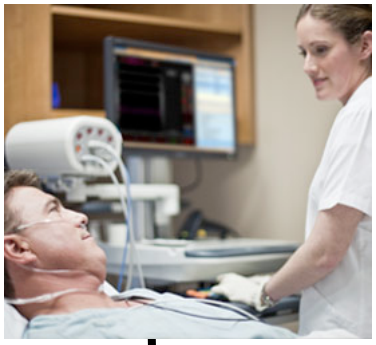
Medical Device Certification

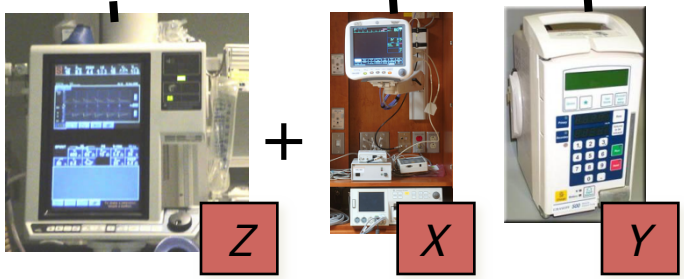
- In the U.S., FDA approves medical devices for specific use
  - Safety and effectiveness are assessed
  - Evaluation is process-based: ISO 9001 (quality management) and ISO 14971 (risk management)
  - Hazard analysis is key to approval
  - FDA's 510(k) requires "substantially equivalent" to devices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







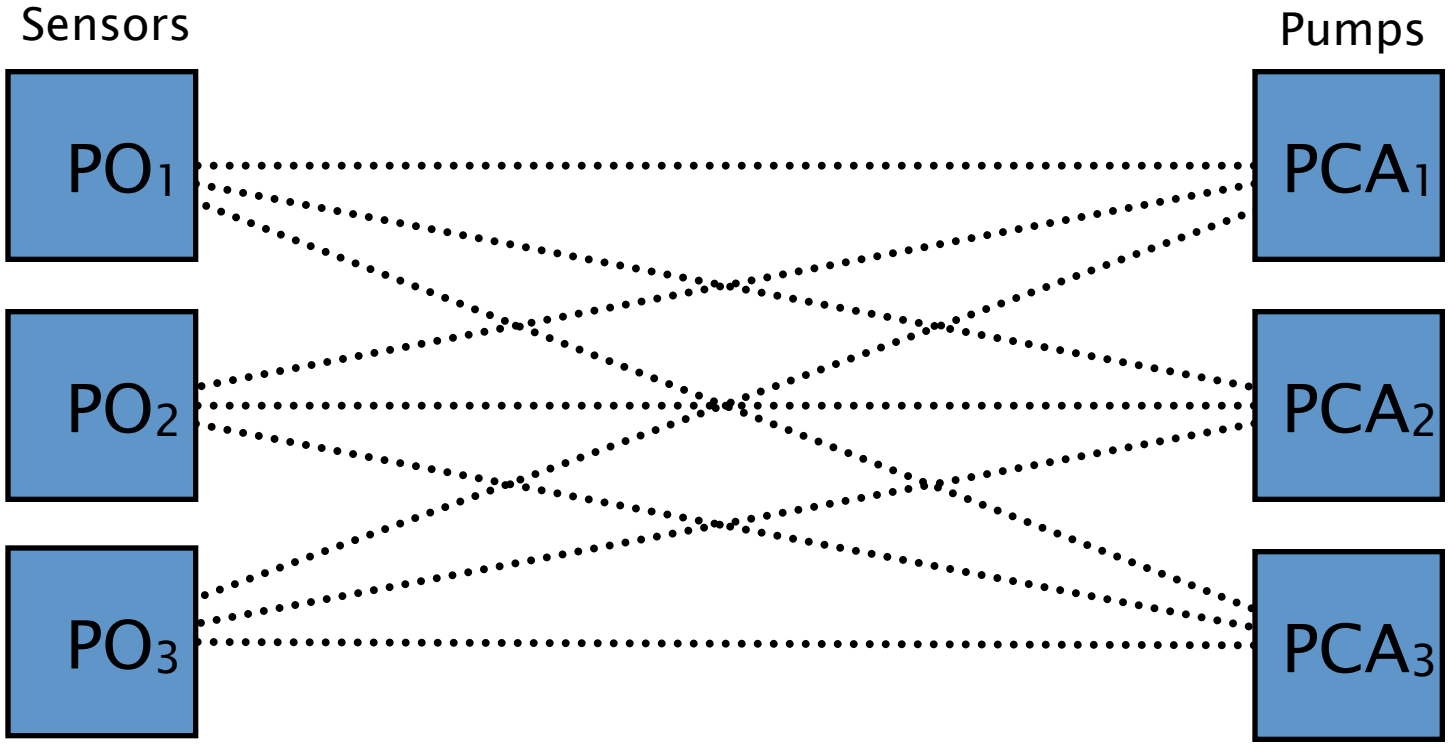
Regulatory Clearance

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:

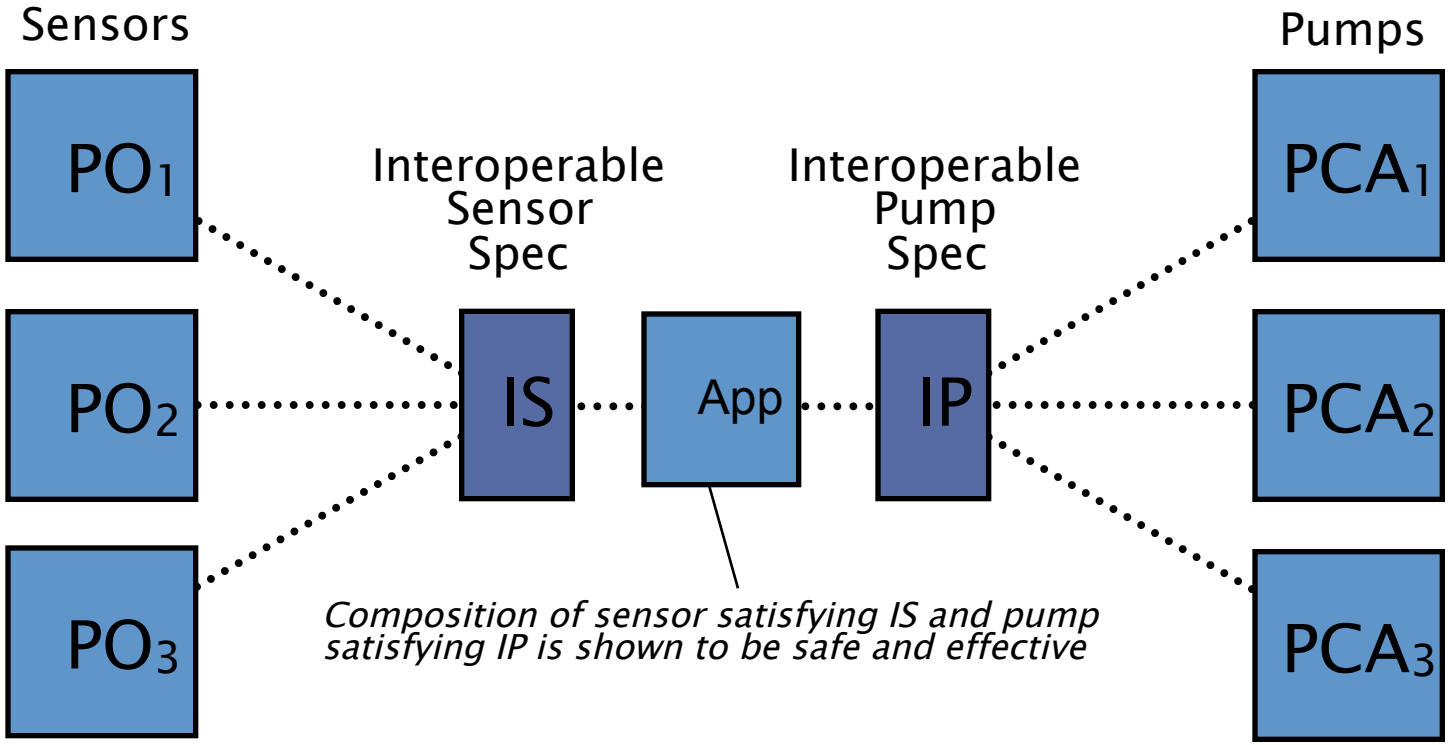


Each sensor must be approved or certified for use with each pump and vice versa. This is burdensome for manufacturers and regulators.

..... Certification or approval relationship

Interface-based Certification

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

Some Observations …

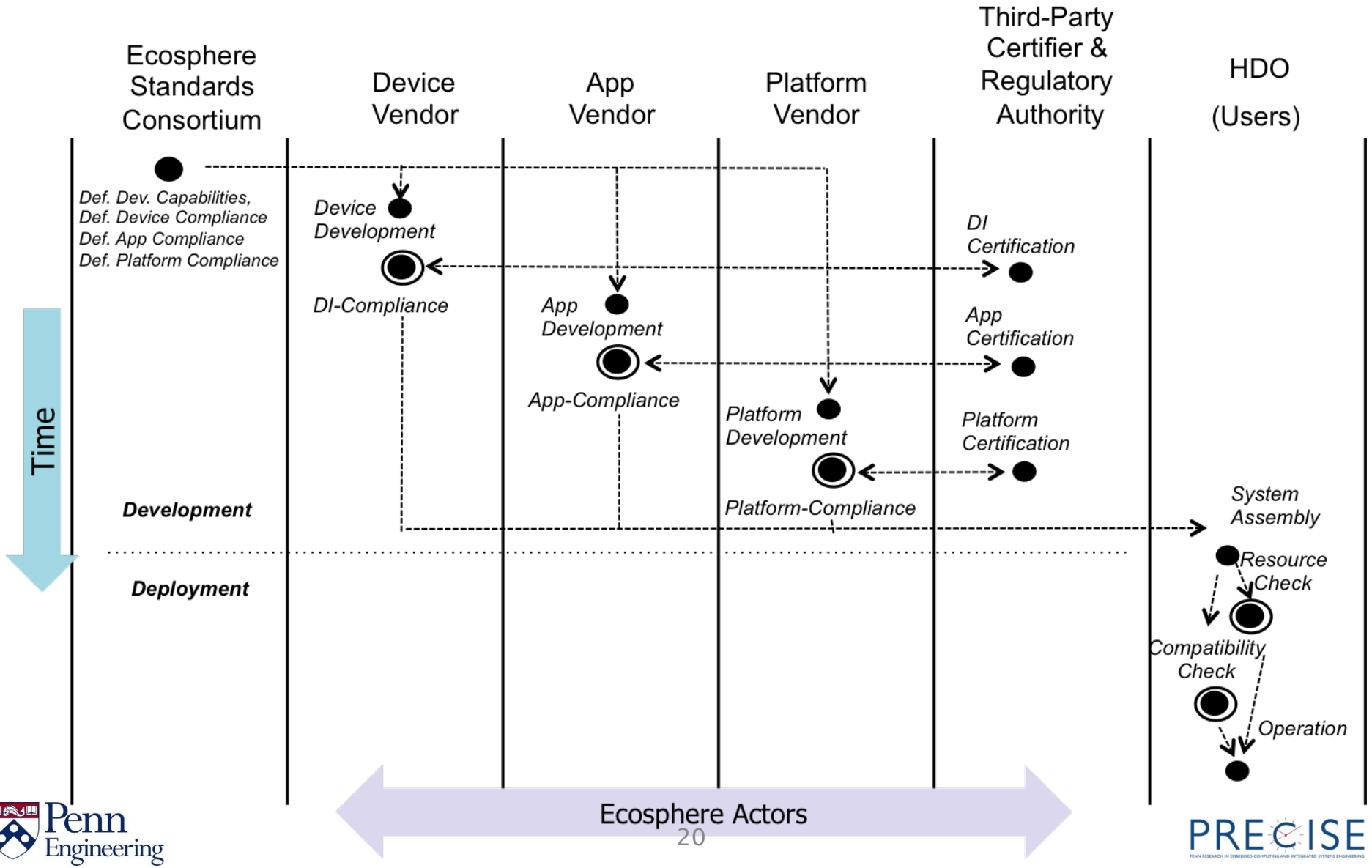
- Safety can only be assured by predicting the emergent system behavior
  - Vendors cannot use traditional methods to directly predict a VMD's behavior, because the system does not exit until assembled by hospital users

- Safety requirements for specific clinical scenarios
  - Devices can interact in unexpected ways, creating new hazards for the patient
  - Manufacturers unlikely anticipate safety hazards for all possible clinical scenarios

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

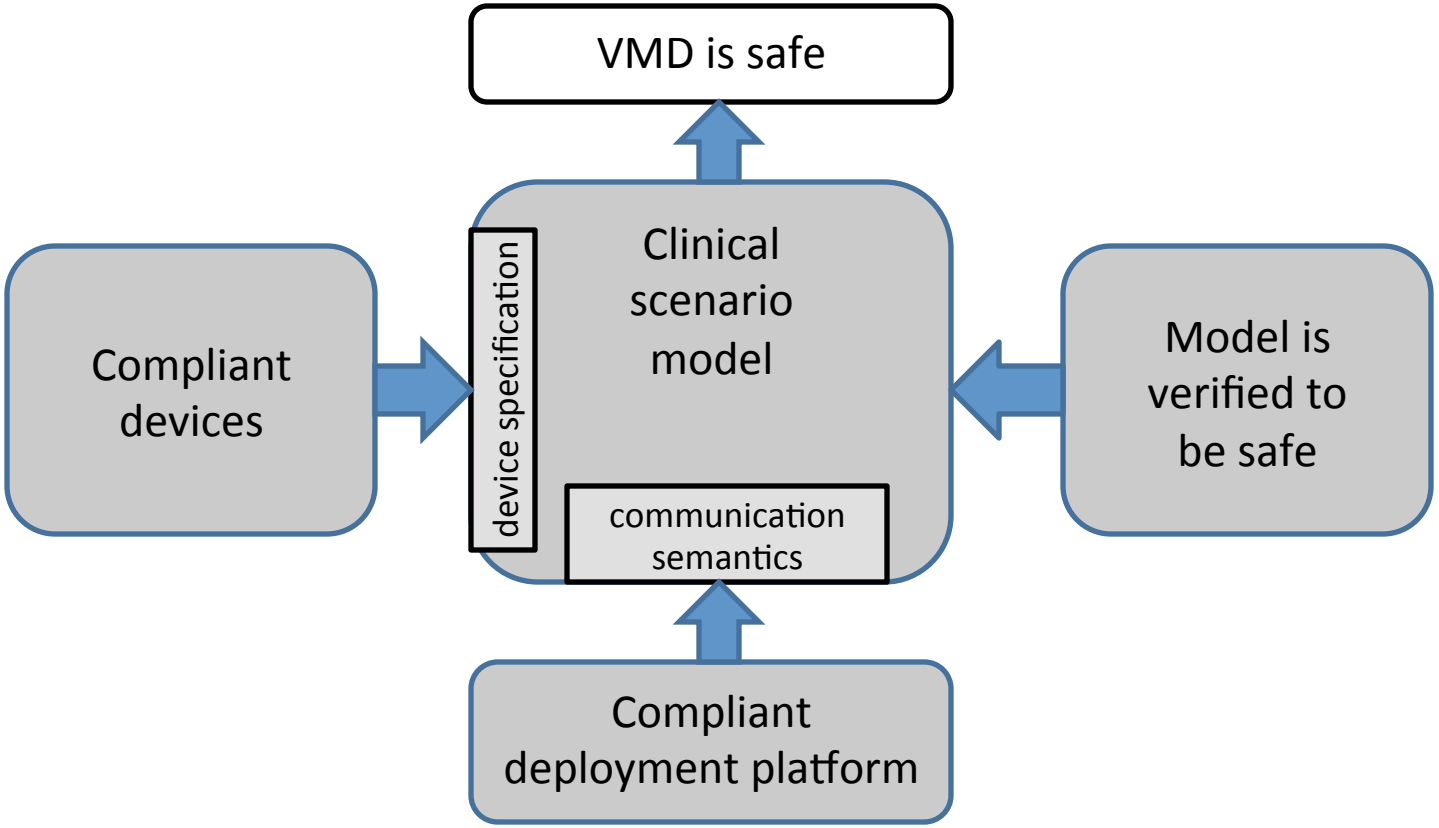


Model-based Safety Reasoning

- 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

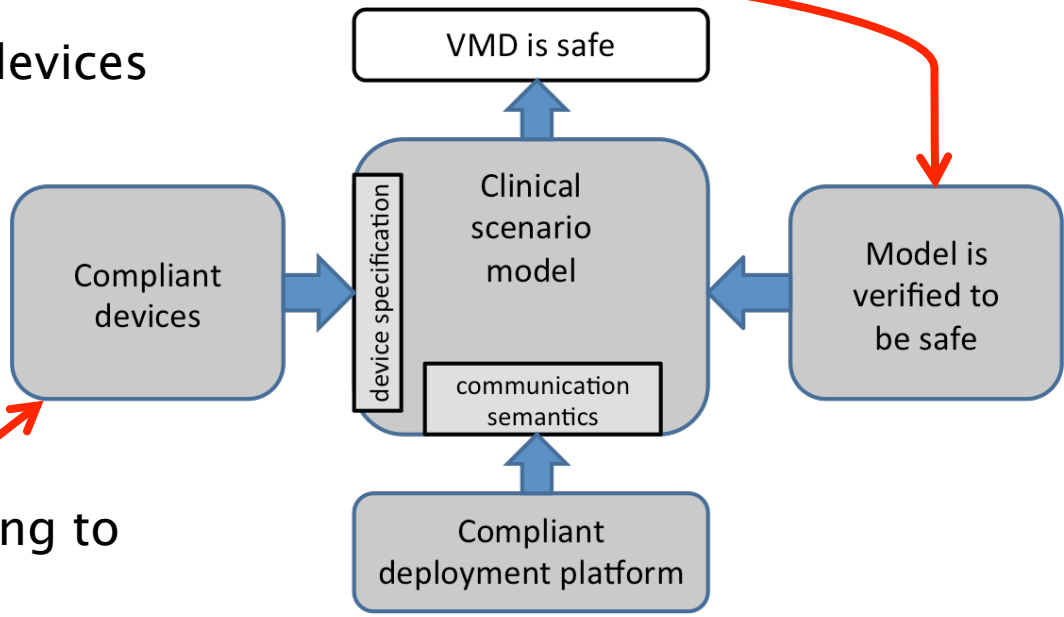
Safety Assurance for VMD

- 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



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 $(j=1\ldots n)$	AI_j $(j=1\ldots n)$	
Devices		DI_j $(j=1\ldots n)$	D_j $(j=1\ldots n)$
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 $(j=1\ldots n)$	AI_j $(j=1\ldots n)$	
Devices		DI_j $(j=1\ldots n)$	D_j $(j=1\ldots n)$
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 $A^m \,(||_{j=1\ldots n}\, AI_j^m) \,||\, P^m \,||\, E^m \models \phi$

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 $(j=1\ldots n)$	AI_j $(j=1\ldots n)$	
Devices		DI_j $(j=1\ldots n)$	D_j $(j=1\ldots n)$
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 $A^m \,(||_{j=1\ldots n}\, AI_j^m) \,||\, P^m \,||\, E^m \models \phi$

---

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

Assume-Guarantee Reasoning Rule

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

①–⑤ $A\ (||_{j=1\ldots n}\ AI_j)\ ||\ P\ ||\ E \models \phi$

⑥ $DI_j \simeq D_j$

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

Assume-Guarantee Reasoning Rule

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

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

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 $(j=1\ldots n)$	AI_j $(j=1\ldots n)$	
Devices		DI_j $(j=1\ldots n)$	D_j $(j=1\ldots n)$
Platform	P^m		P
Environment	E^m		E

①–⑤ $A\ (||_{j=1\ldots n}\ AI_j)\ ||\ P\ ||\ E \models \phi$

⑥ $DI_j \simeq D_j$

⑦ $AI_j \simeq DI_j$

---

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

Assume-Guarantee Reasoning Rule

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

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

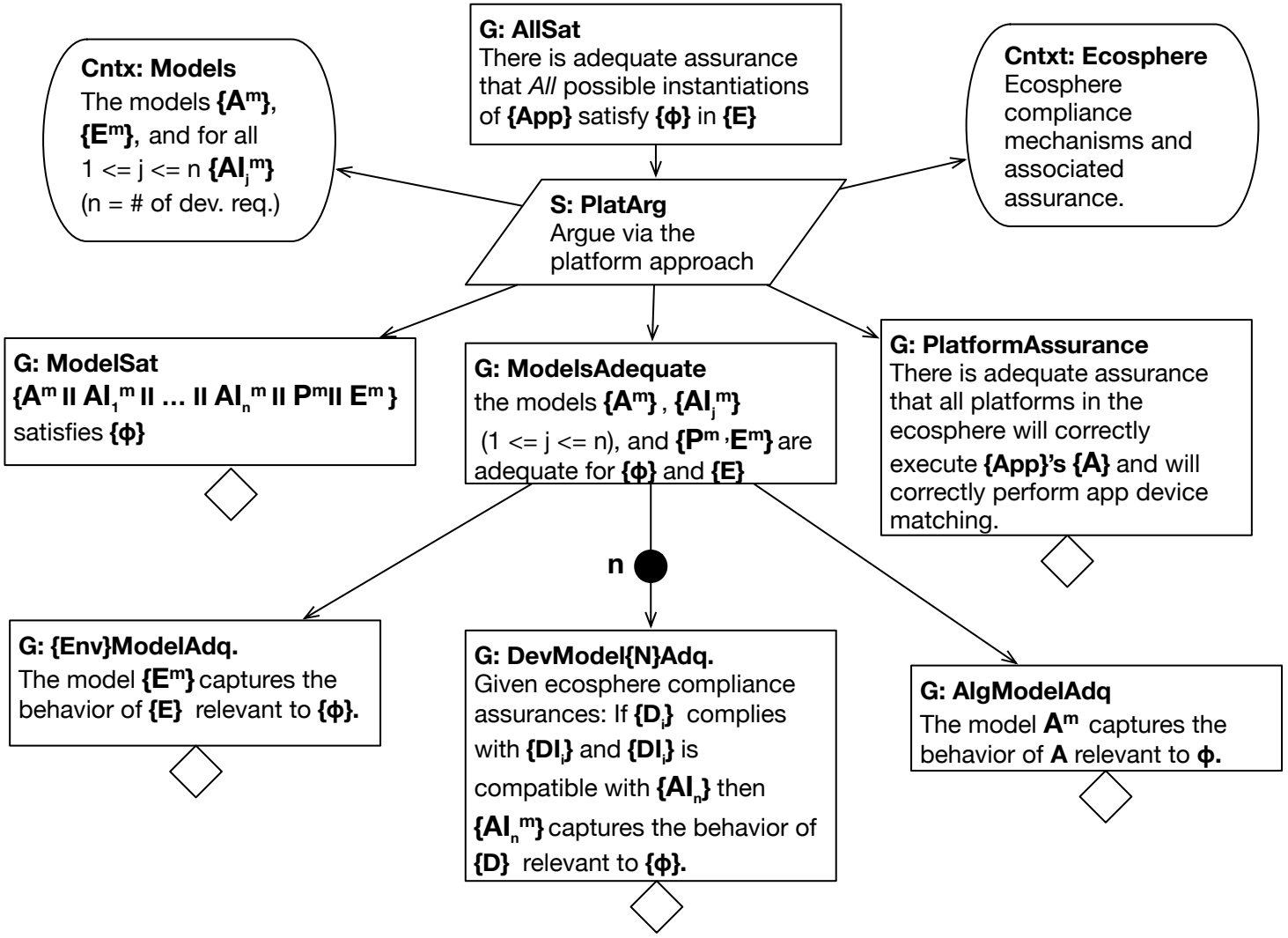
⑧ $A\ ||\ P \simeq P(A)$ /* P(A) means Dj's are compatible for A */

---

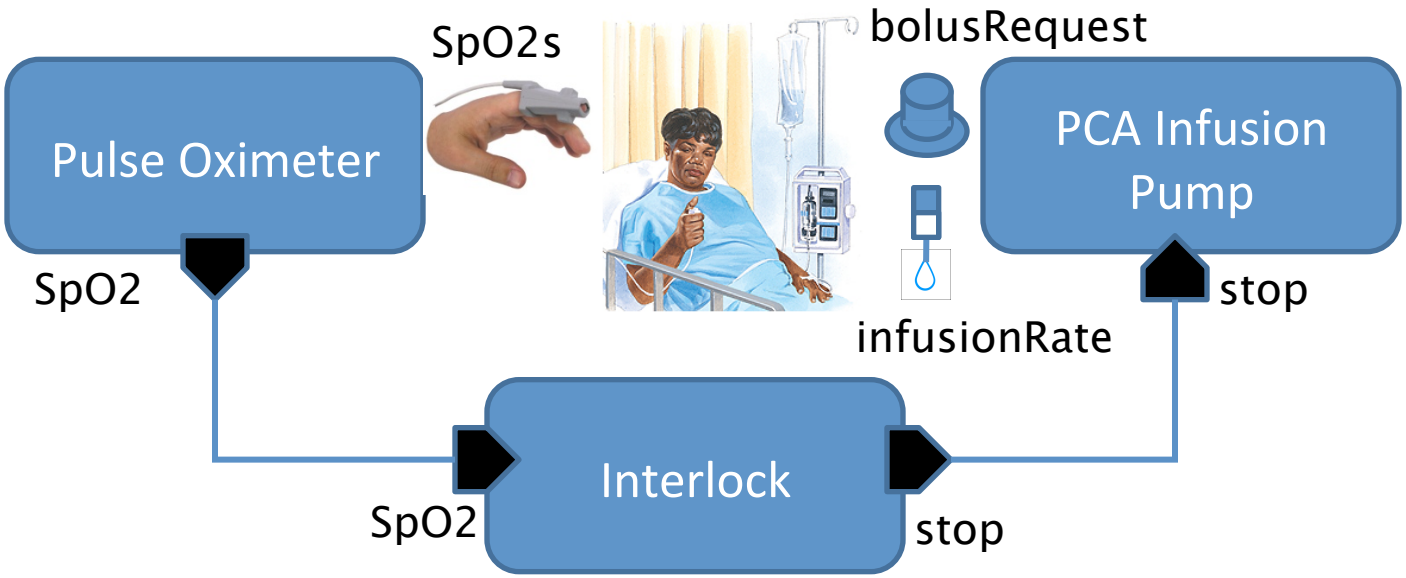
①-⑧ $P(A)\ (||_{j=1\ldots n}\ D_j)\ ||\ E \models \phi$

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.

Proposed Assurance Argument Pattern

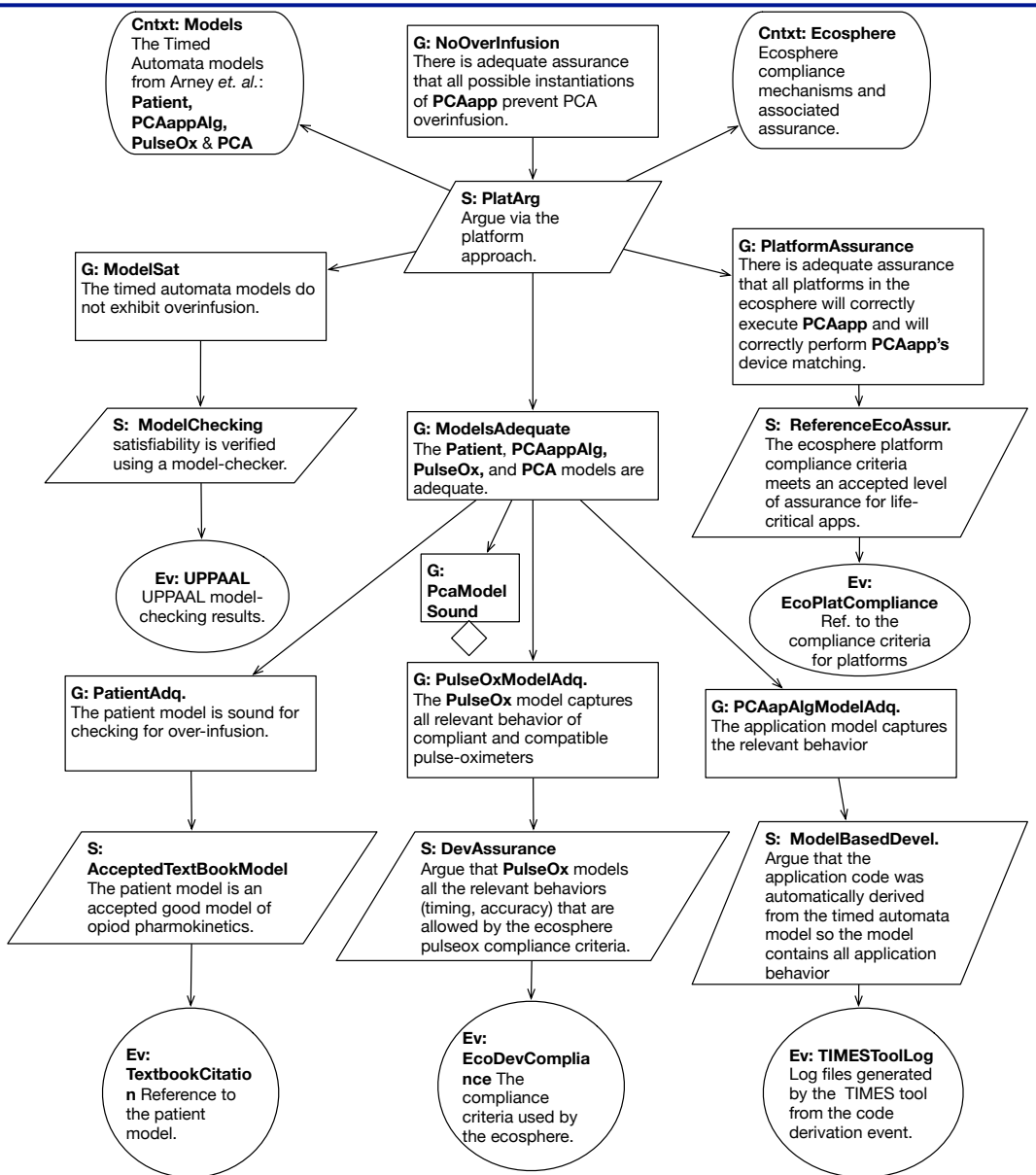


Case Study: PCA Control App



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

Example Assurance Case

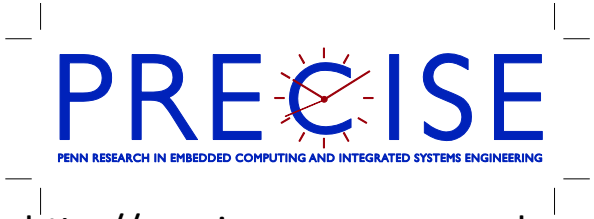


Summary

- 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

Thank You!
Questions?



http://precise.seas.upenn.edu