Assuring Medical Device Interoperability Plug-and-Play Open Systems

Lu Feng, Andrew King, Insup Lee, Oleg Sokolsky

PRECISE Center
School of Engineering and Applied Science
University of Pennsylvania

SCC Meeting, Annapolis, May 2015

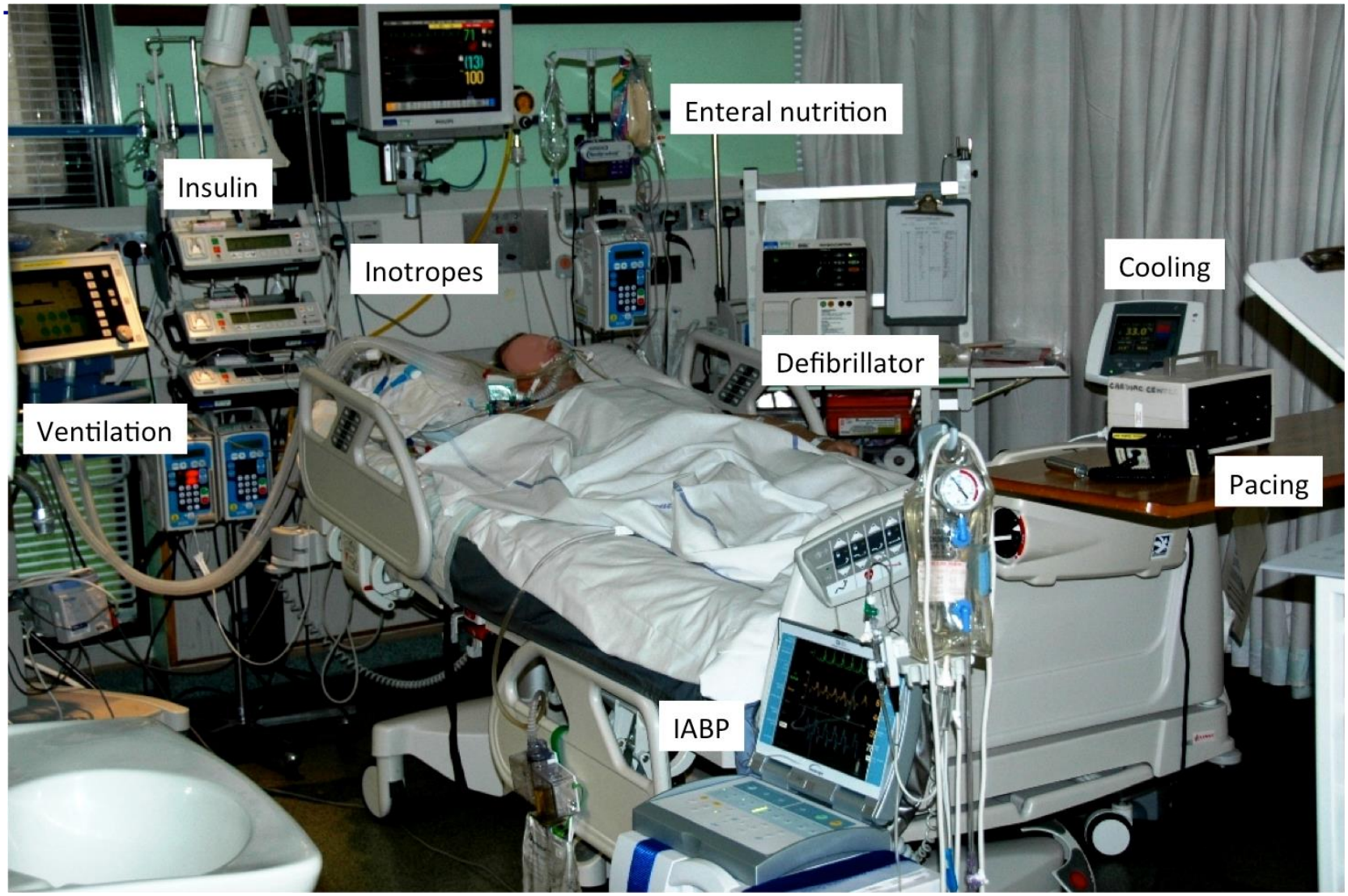
Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Outline

1. Medical Device Interoperability
2. Safety Assurance for MD PnP Systems
3. Medical Application Platform Design

Example: ICU


Enteral nutrition
Insulin
Inotropes
Cooling
Defibrillator
Ventilation
Pacing
IABP

Current 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)

- Potential safety hazards
  - Clinical scenario 1: *laser surgery / ventilator*
  - Clinical scenario 2: *x-ray / ventilator*

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

Clinical Scenario: X-Ray / Ventilator

"With the advent of sophisticated anesthesia machines incorporating comprehensive monitoring, it is easy to forget that serious anesthesia mishaps still can and do occur."

APSF Newsletter Winter 2005

Anesthesia Machine

Surgeons

Portable x-ray machine

Why 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
  - Physiological closed-loop control in many cases

Current

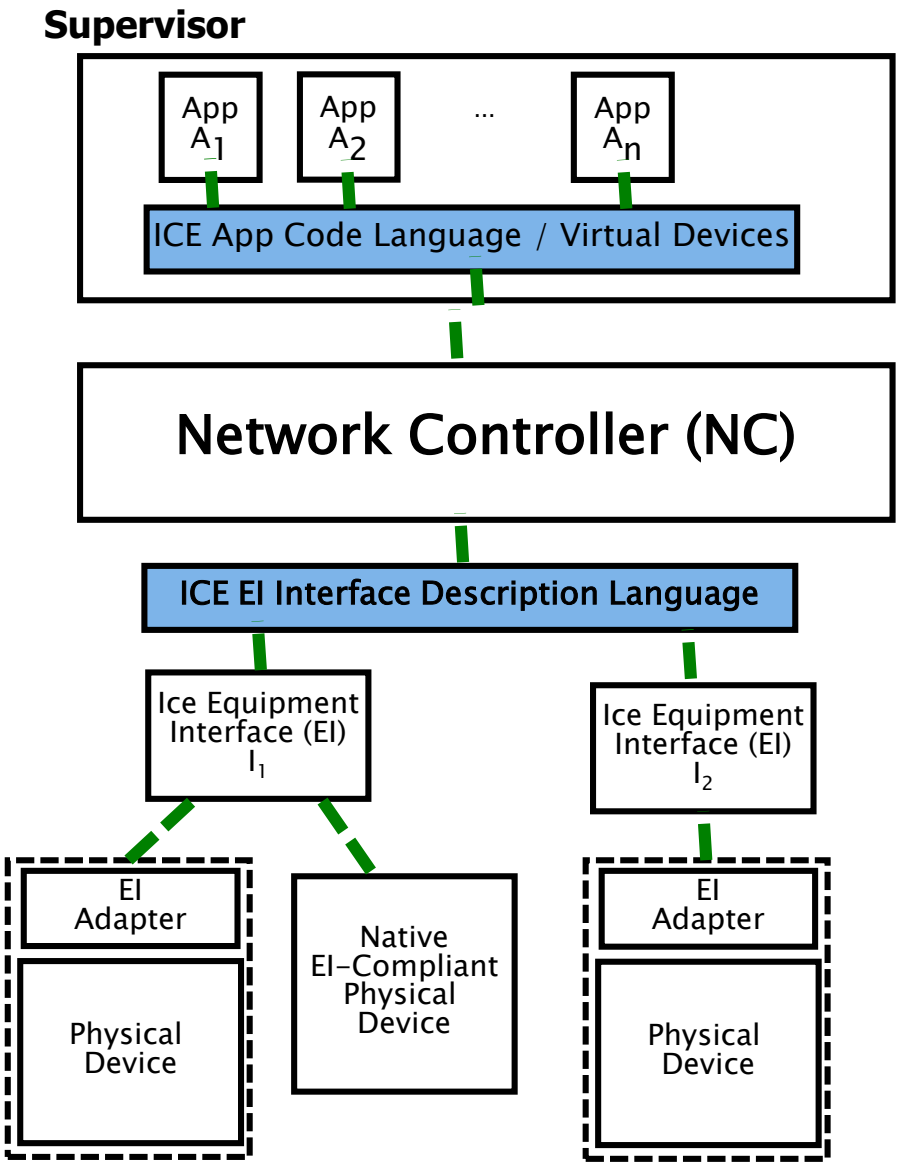
Future

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)

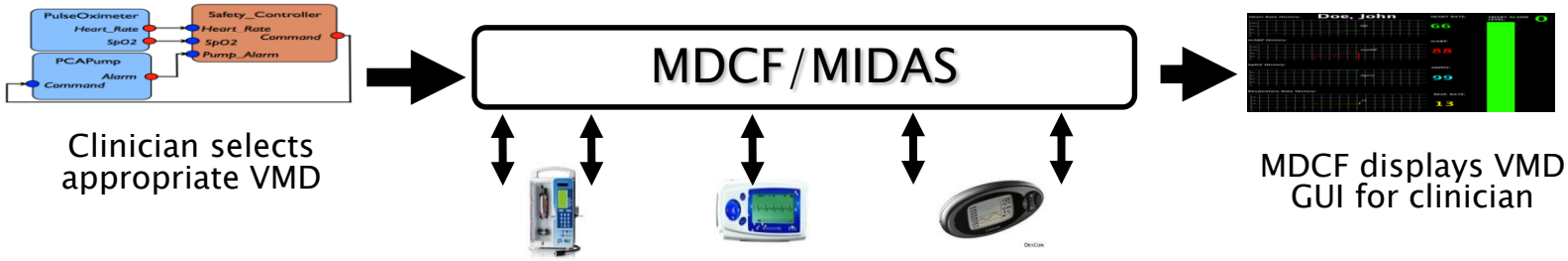
- **MD PnP** enables the concept of **VMD**
  - A set of medical devices coordinating over a network for clinical scenario

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 displays VMD GUI for clinician

MDCF binds appropriate devices into VMD instance

Outline

1. Medical Device Interoperability
2. Safety Assurance for MD PnP Systems
3. Medical Application Platform Design

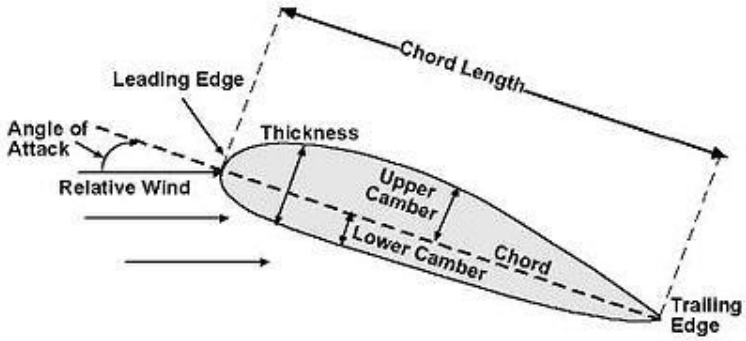
VMD Safety Assurance Challenge

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

Emergent Property

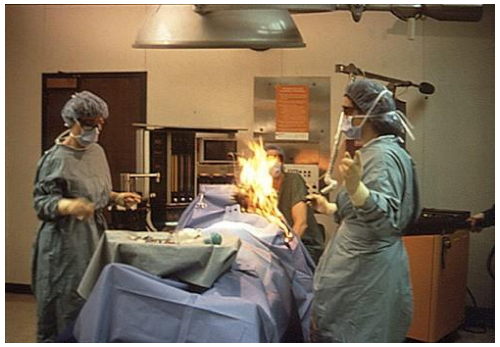
- Example: **Top-speed of an airplane**
  - The top-speed is a function of

    engines + fuselage + wings + flight control software (FCS)
  - Does it make sense to talk about the top-speed of FCS?



Safety is an Emergent Property

- Example: **Laser / Ventilator Safety Interlock**



Safe system

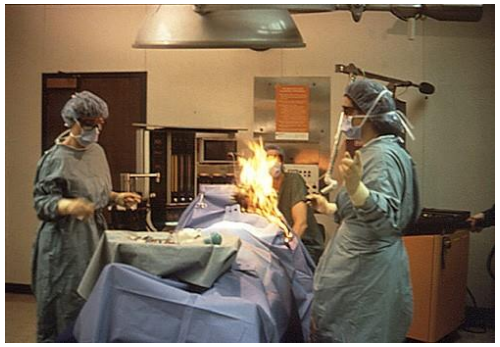
(Emergent) Behavior of Integrated System

Unsafe states:

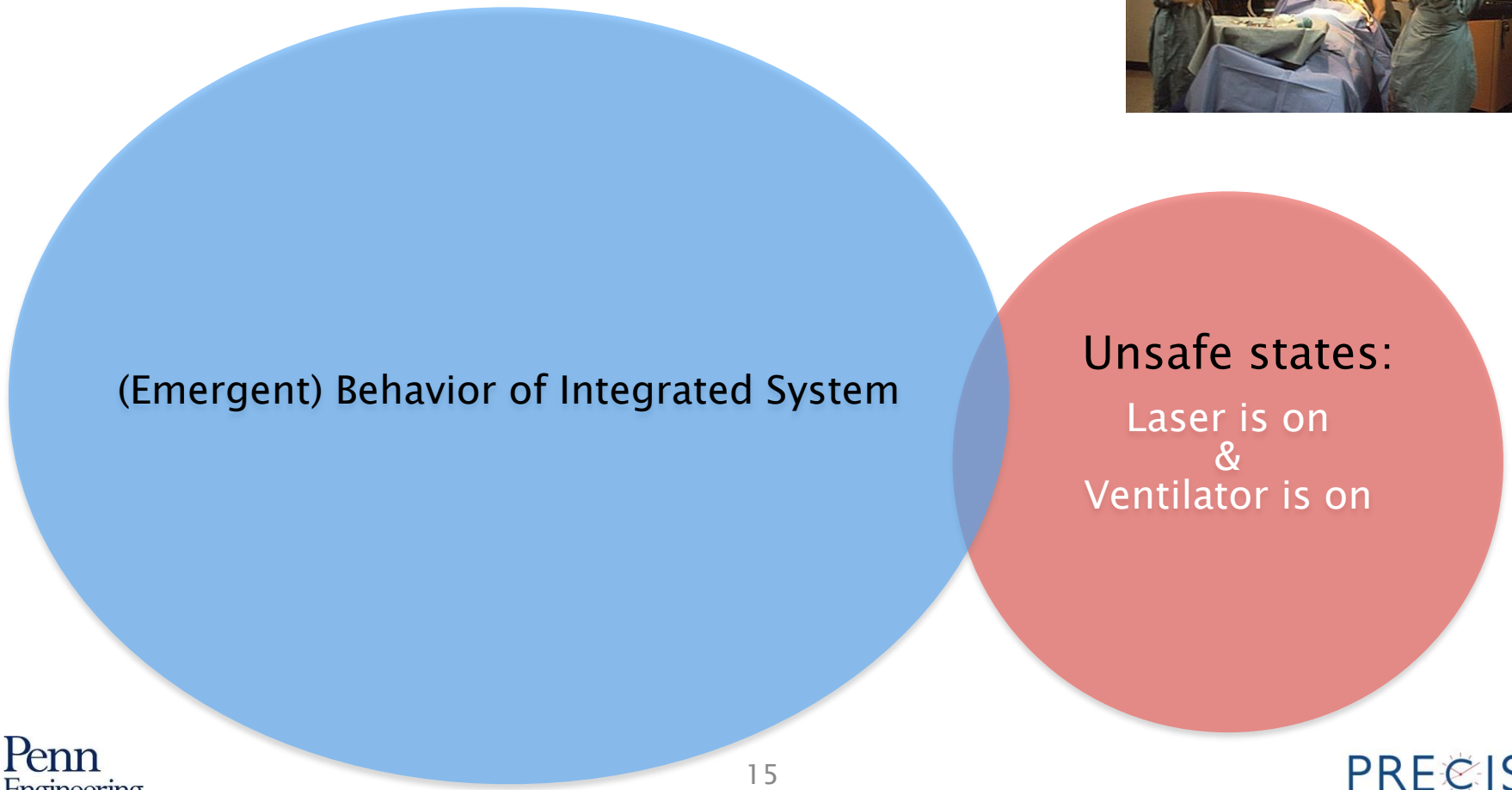
Laser is on
&
Ventilator is on

Safety is an Emergent Property

- Example: **Laser / Ventilator Safety Interlock**



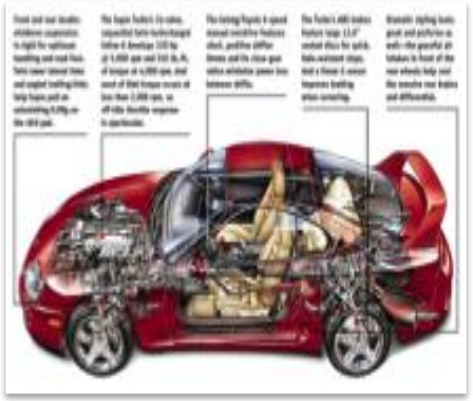
Unsafe system



Traditional Safety Critical Systems

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

Aerospace

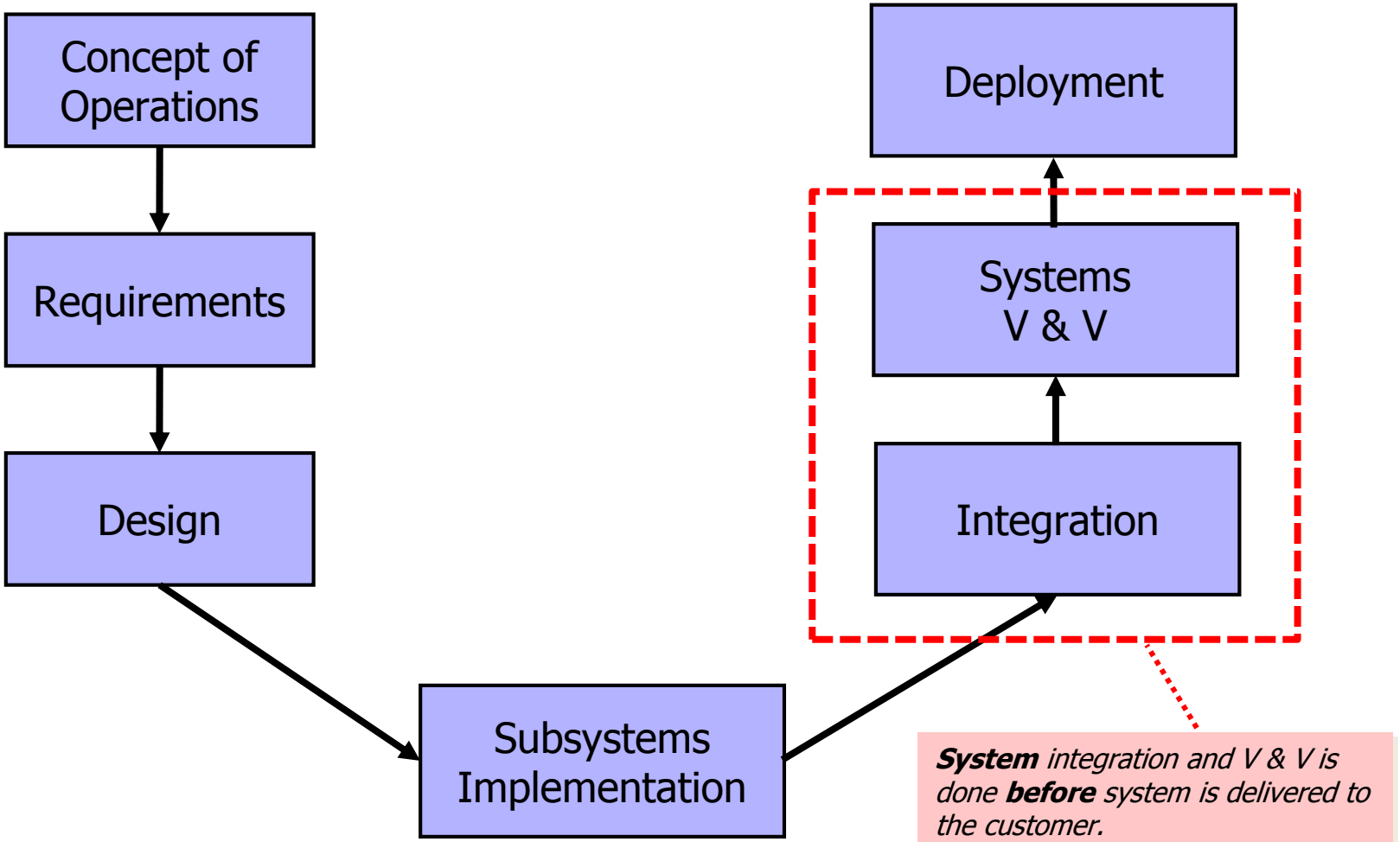


Automotive

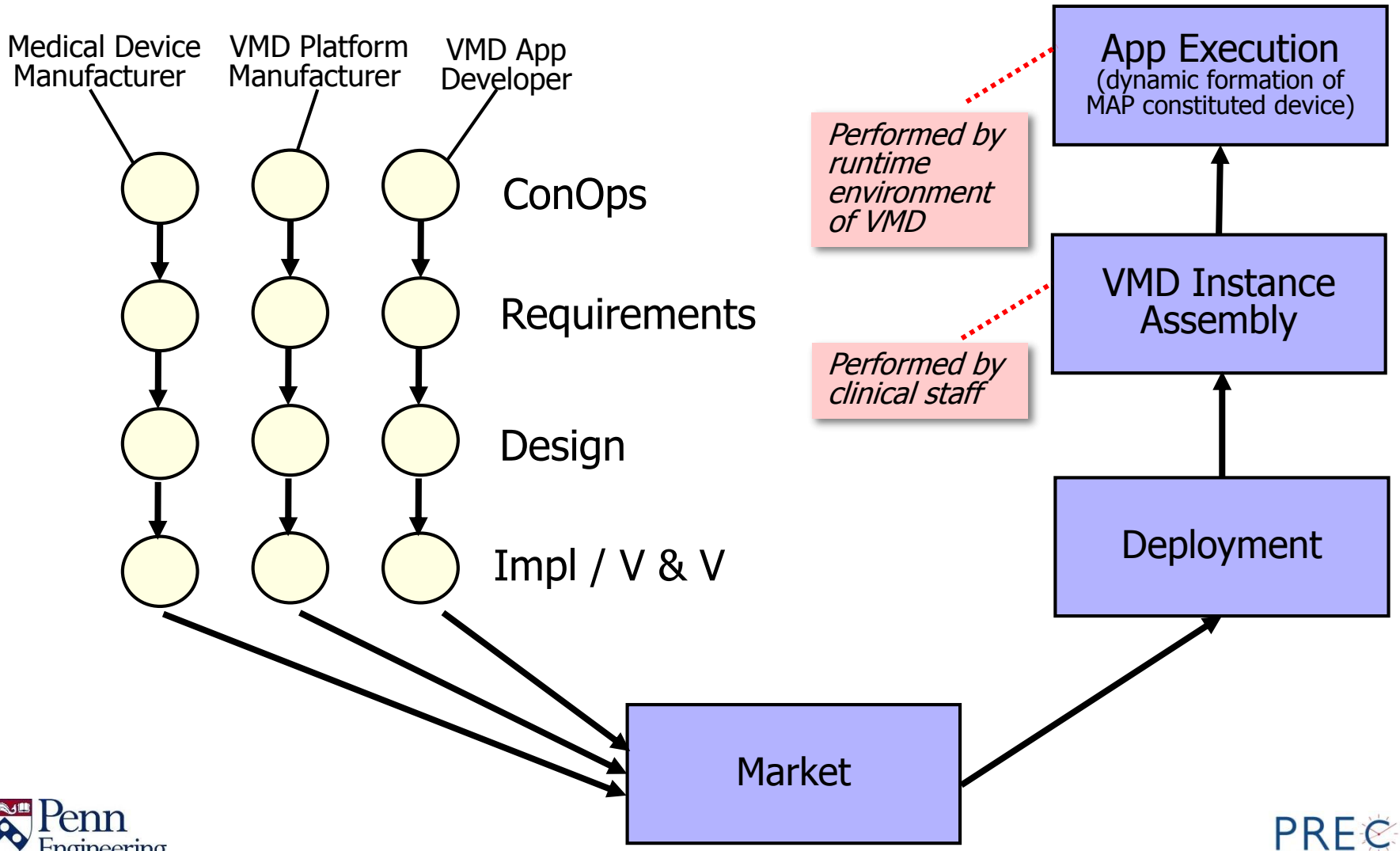
Nuclear

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

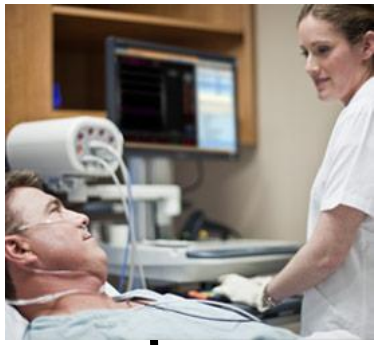
Medical Device Certification

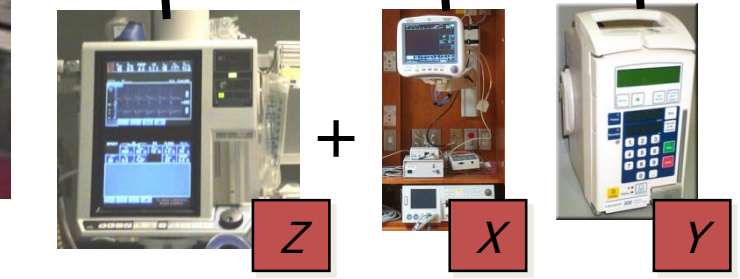
- In the U.S., FDA approves medical devices for specific use
  - Safety and effectiveness are assessed
  - Evaluation is process-based: ISO 9001 (quality management) and ISO 14971 (risk management)
  - Hazard analysis is key to approval
  - FDA's 510(k) requires "substantially equivalent" to devices 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.

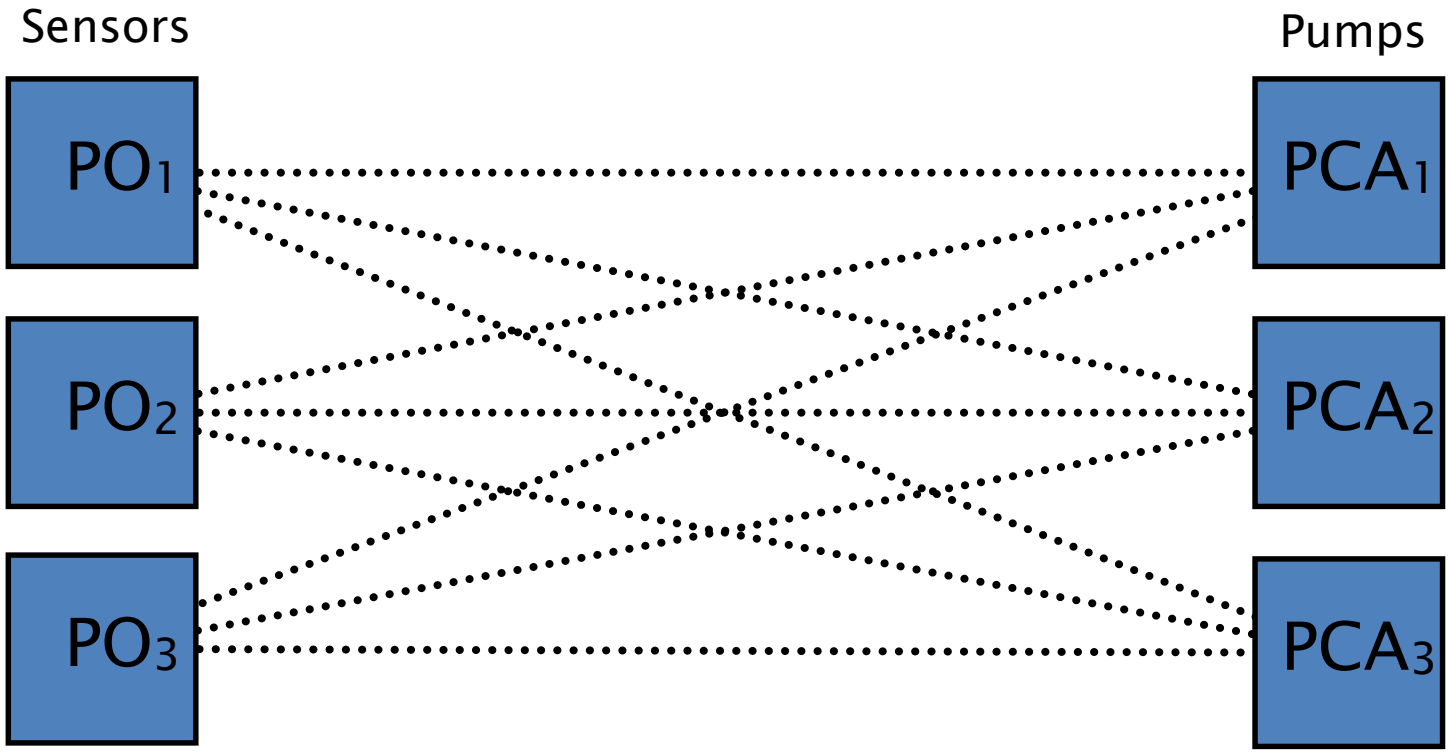
In current regulatory approach, adding a new type of device (e.g., Z) typically causes the entire system to be re-submitted for regulatory clearance.

Penn Engineering

PRECISE

Pairwise Certification Complexity

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

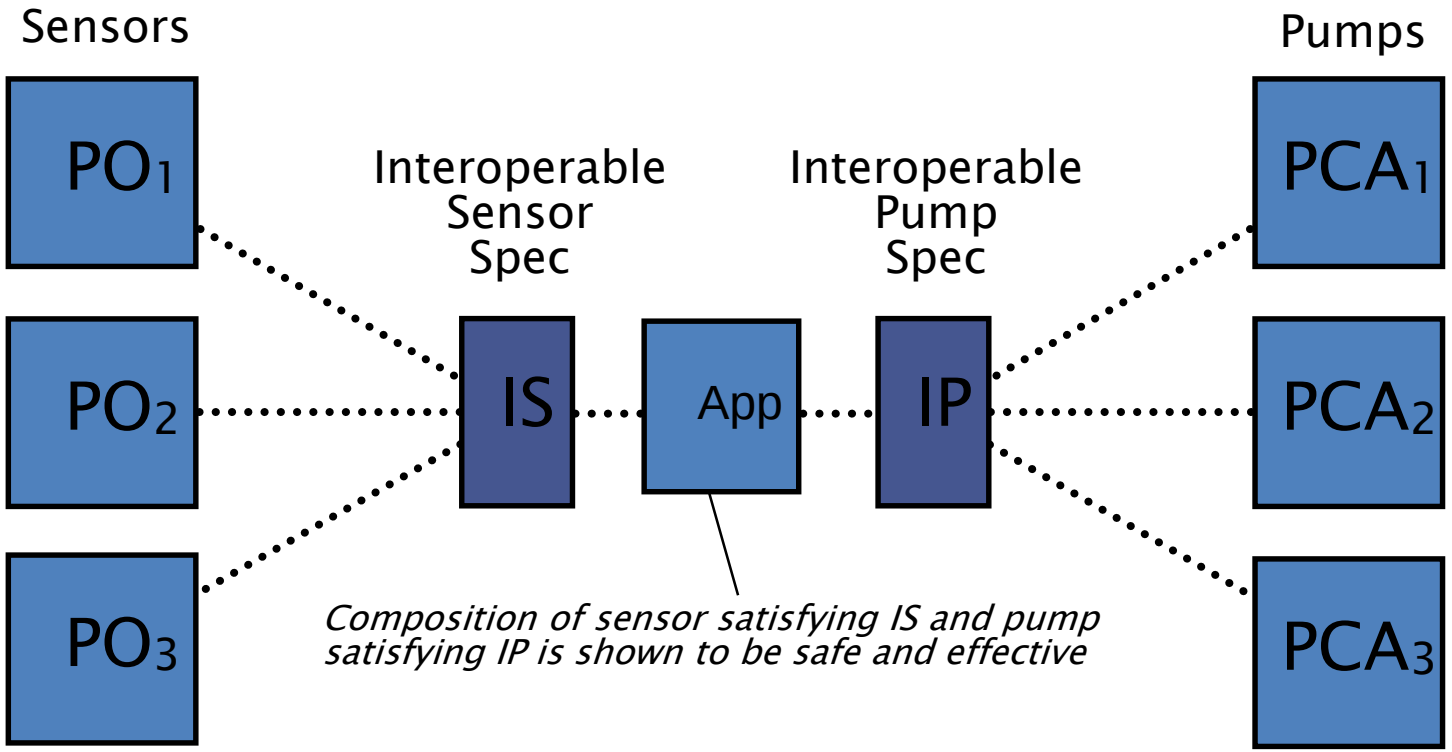


Each sensor must be approved or certified for use with each pump and vice versa. This is burdensome for manufacturers and regulators.

..... 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

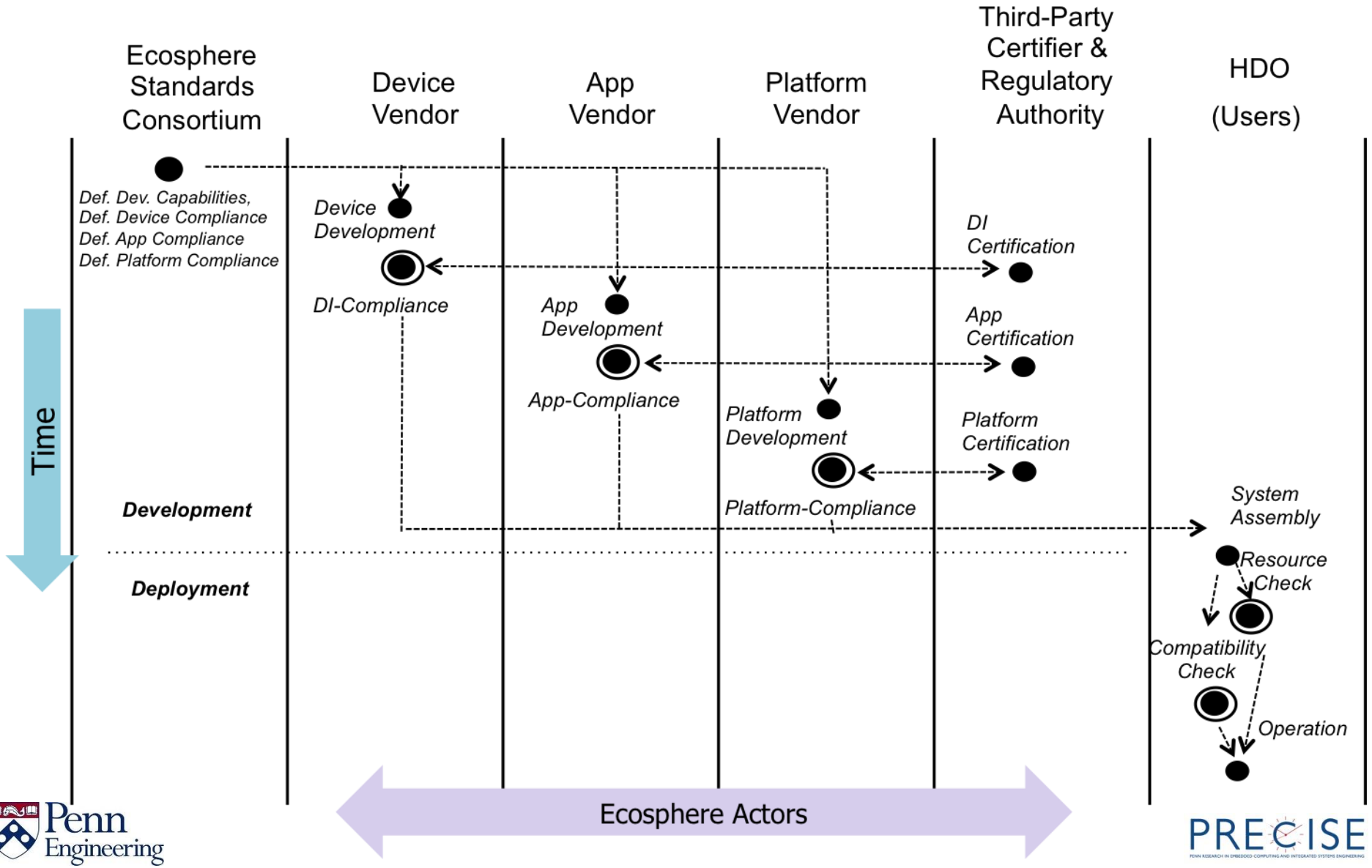
- Safety requirements for specific clinical scenarios
  - Devices can interact in unexpected ways, creating new hazards for the patient
  - Manufacturers unlikely anticipate safety hazards for all possible clinical scenarios

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
  - Devices specify a formal capabilities model
  - 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

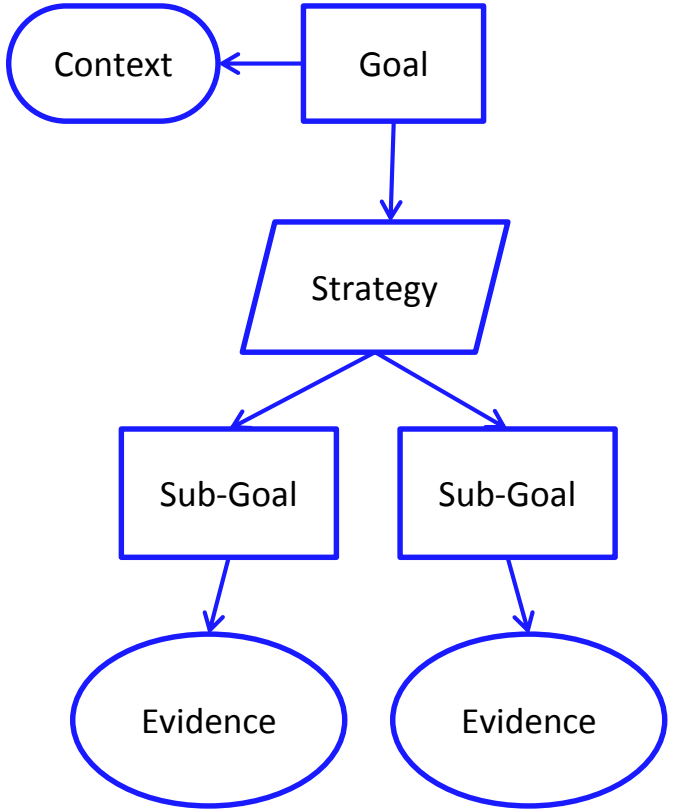


Model-based Safety Reasoning

- 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

Assurance Cases: Motivation

- GSN pattern specification
- Makes App developer justify the models used in the MBR given:
  - The intended use of the application
  - The ecosphere certification and assurance processes
  - Platform capabilities
  - etc.
- Help both regulators and developers identify assurance deficits



Incremental Assurance Composition

- App vendors need to leverage both
  - safety of individual medical devices, and
  - the assurances provided by the platform eco-system

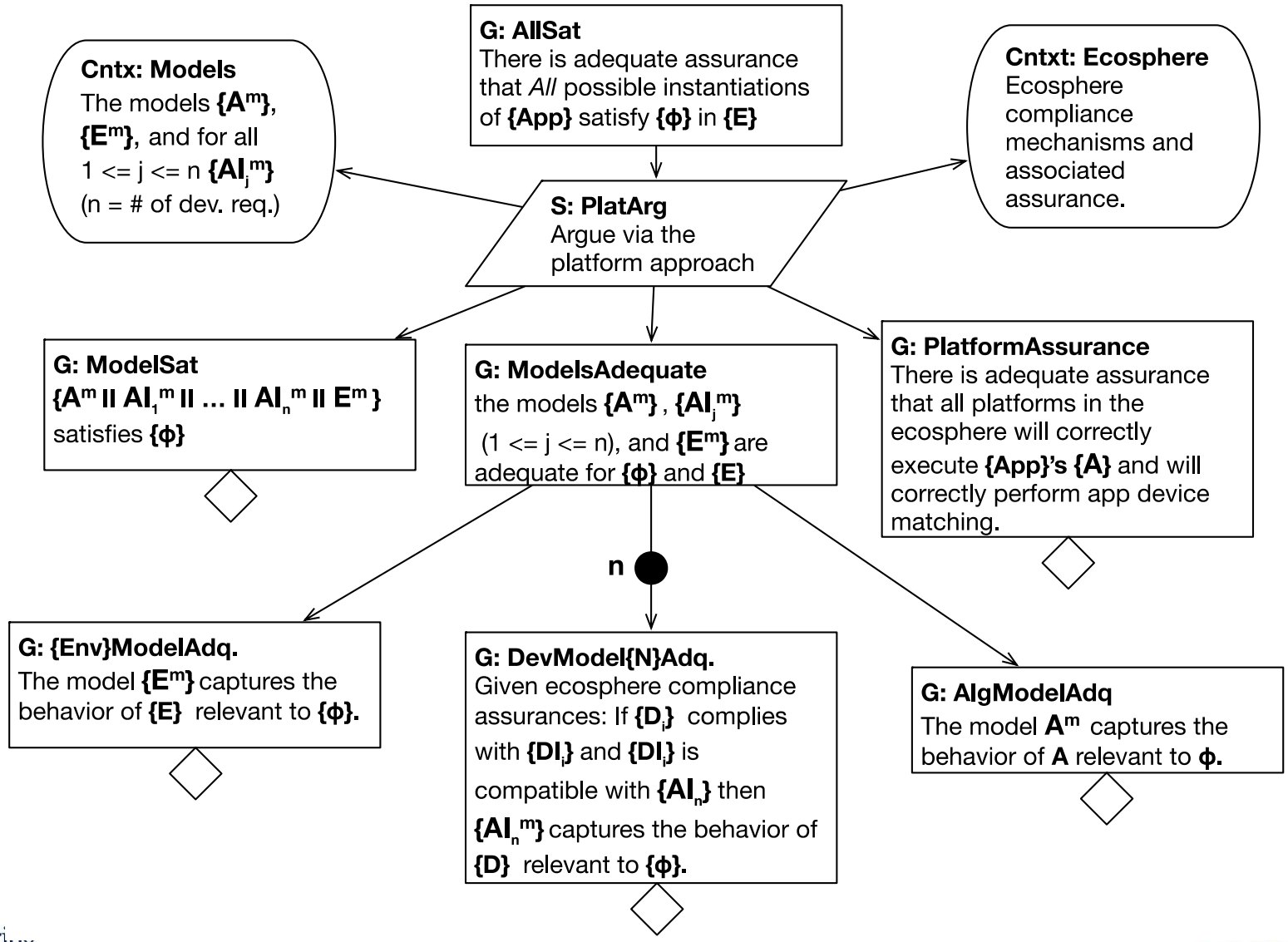
- ***Goal***: establish a sound way of combining these safety assurances into an assurance case for the App

- Platform Approach based Argument Strategy
  - Step 1: model-based reasoning

    $$A^m \,||\, AI_1^m \,||\, \ldots \,||\, Ai_n^m \,||\, E^m \models \phi$$

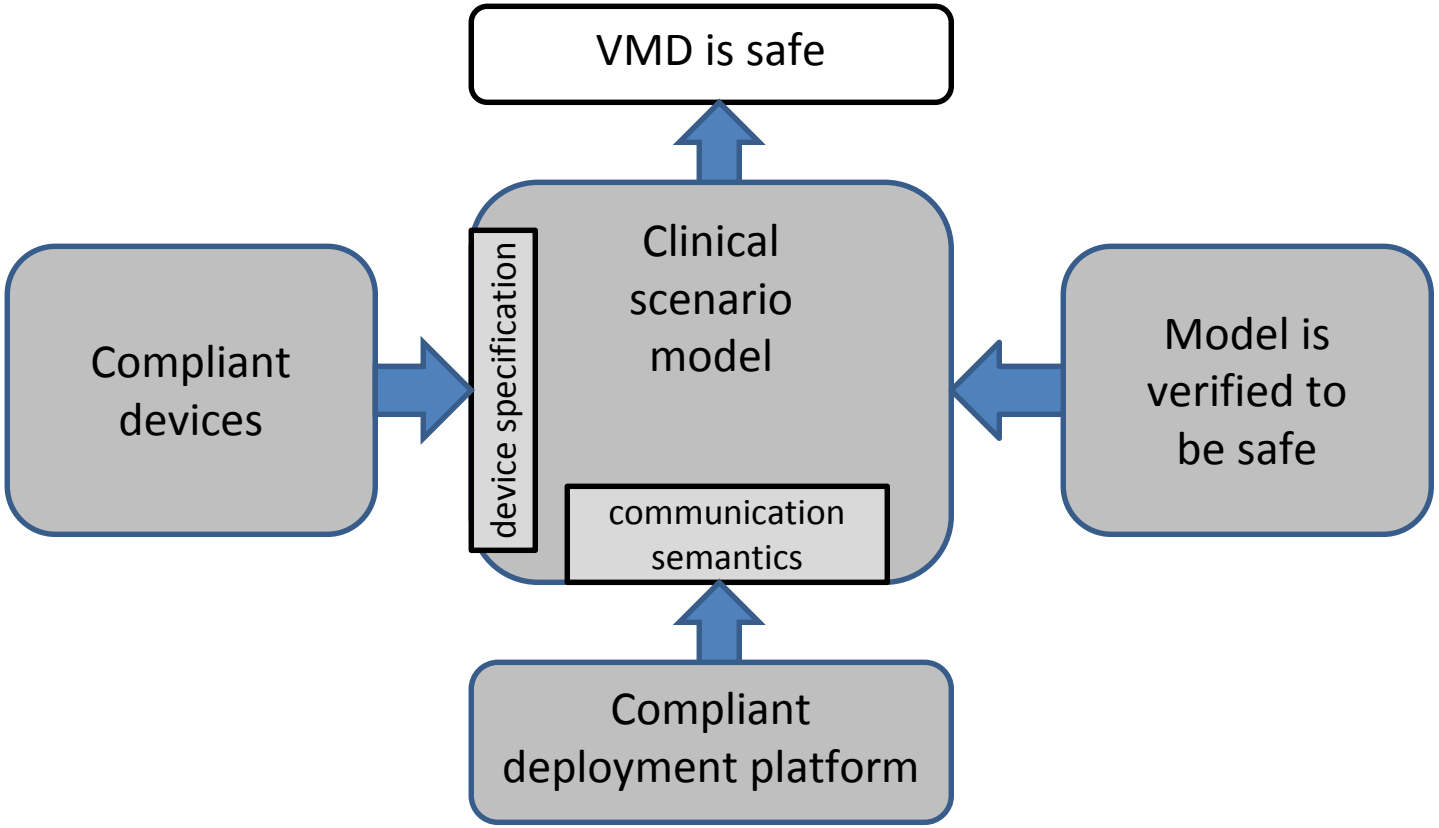
  - Step 2: argue why models (for App, devices, and environment) used in Step 1 are adequate
  - Step 3: argue why the assurance provided by any ecosphere compliant platform is sufficient to support the App

Safety Assurance Argument Pattern



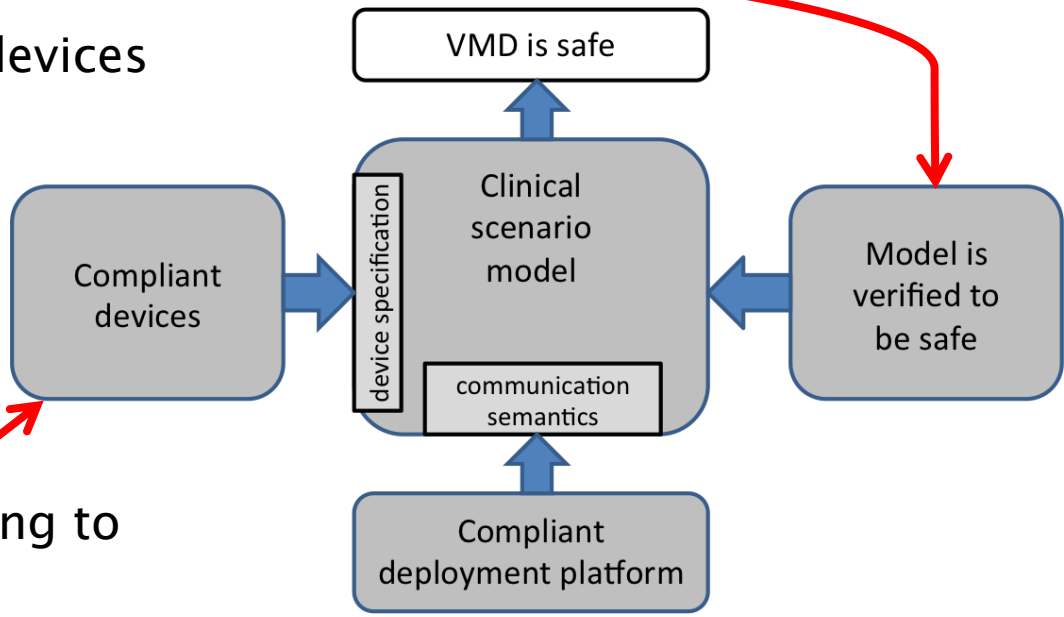
Safety Assurance for Platform Approach

- 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

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

vmd ClosedLoopPCA

devices

pcaPump : PCA

po : PulseOximeter

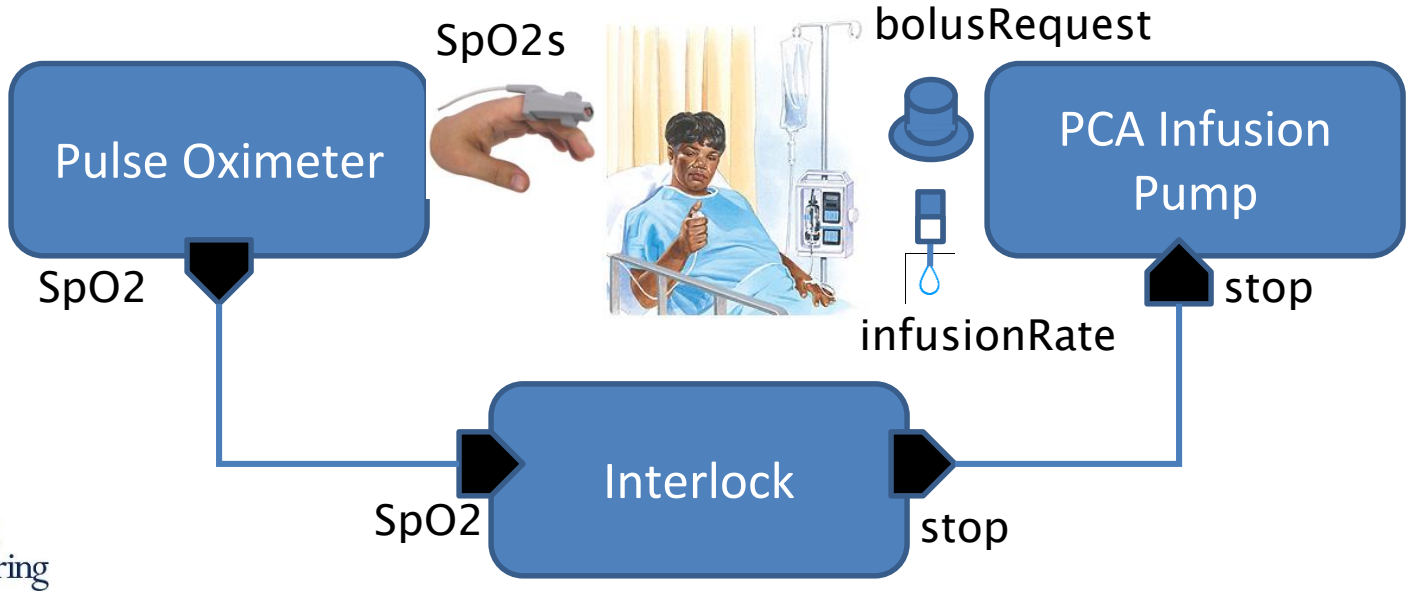
logicmodules

controller : PCATicketGenerator

dataflows

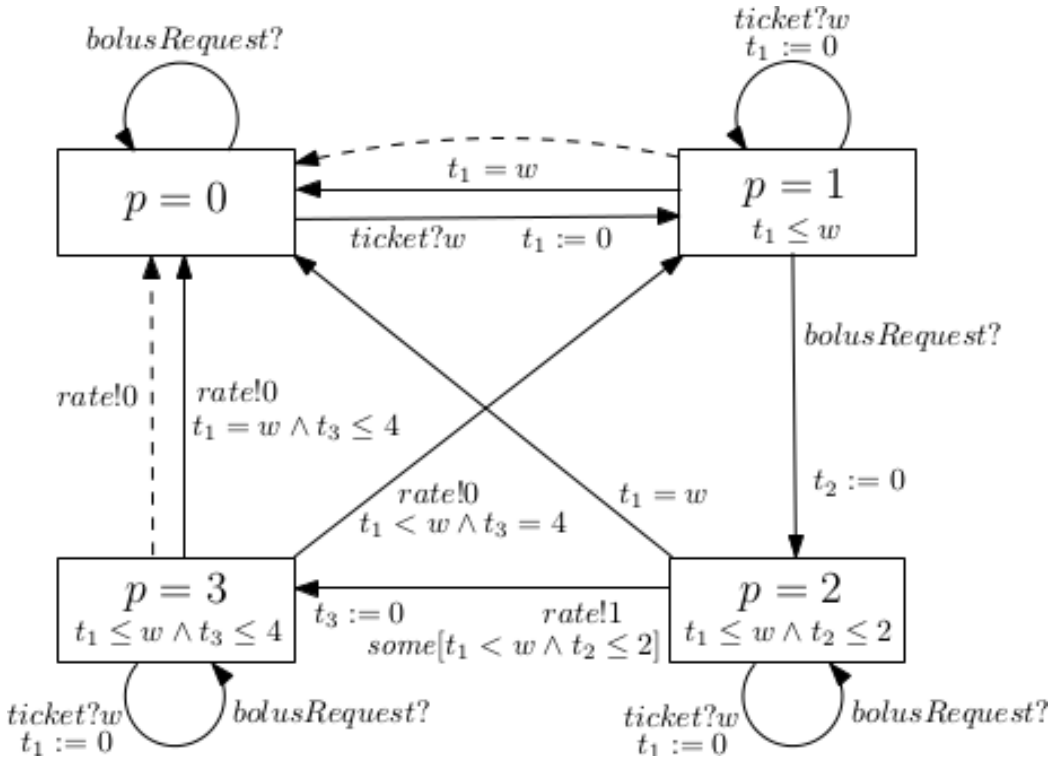
po.SpO2 →^{50ms} controller.SpO2

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

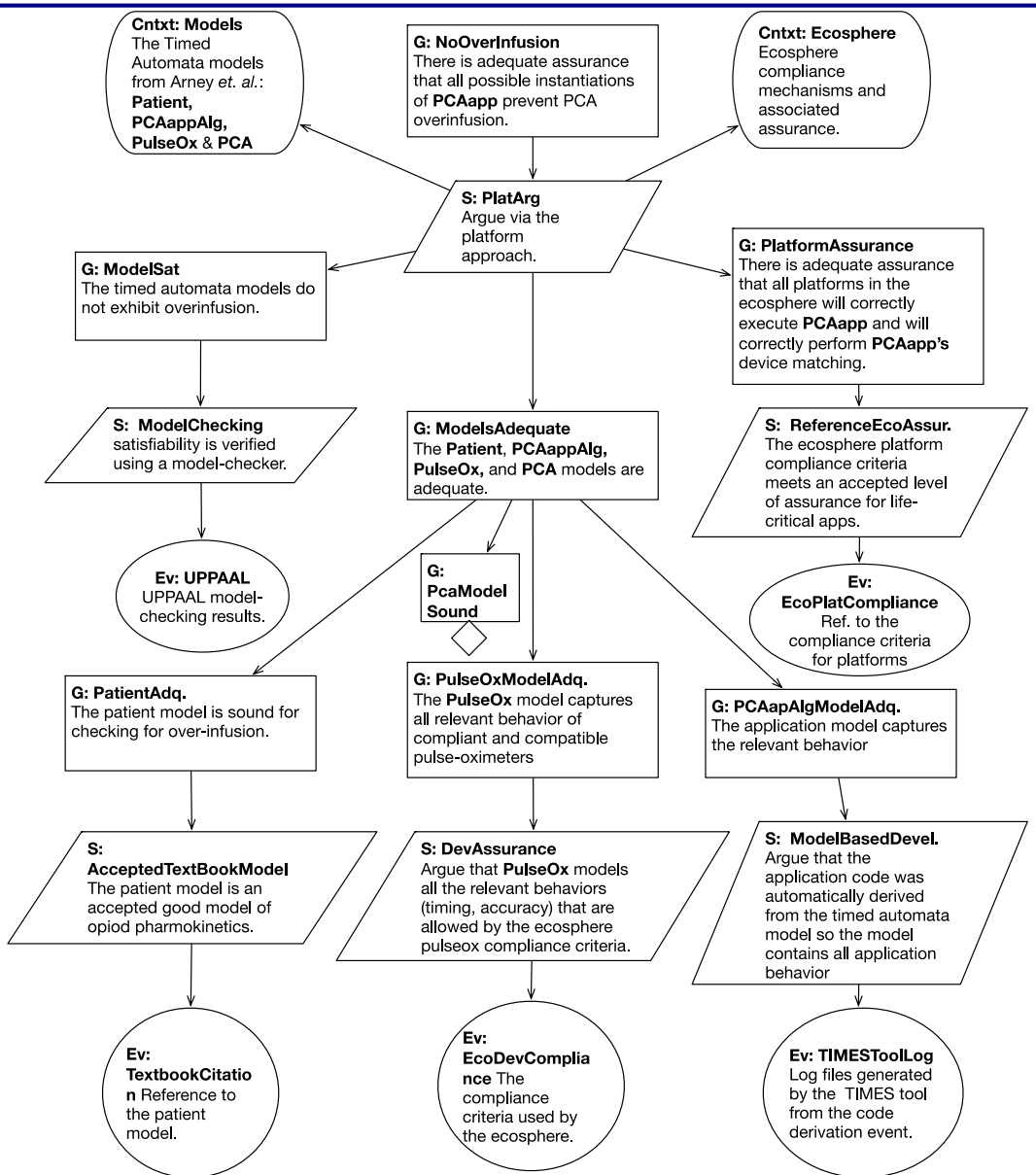


Example Modal Behavior Specification

module PCA: **device**
  interface
    patient input: bolusRequest *event*
    patient output: *rate continuous infusionRate*[0..2]
    network input: ticket *event integer*[0..300]



Example Assurance Case



Penn Engineering

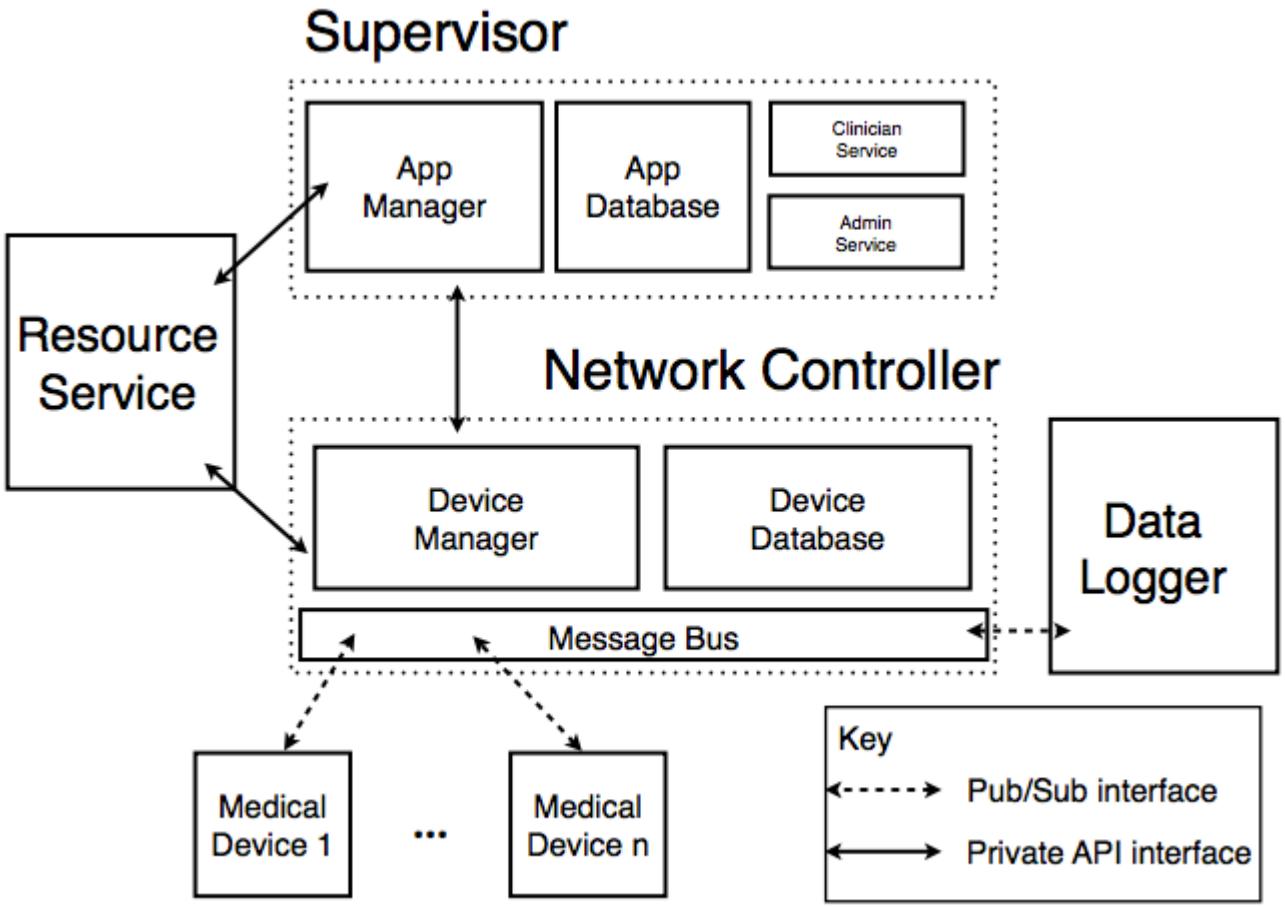
PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Outline

1. Medical Device Interoperability
2. Safety Assurance for MD PnP Systems
3. Medical Application Platform Design

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

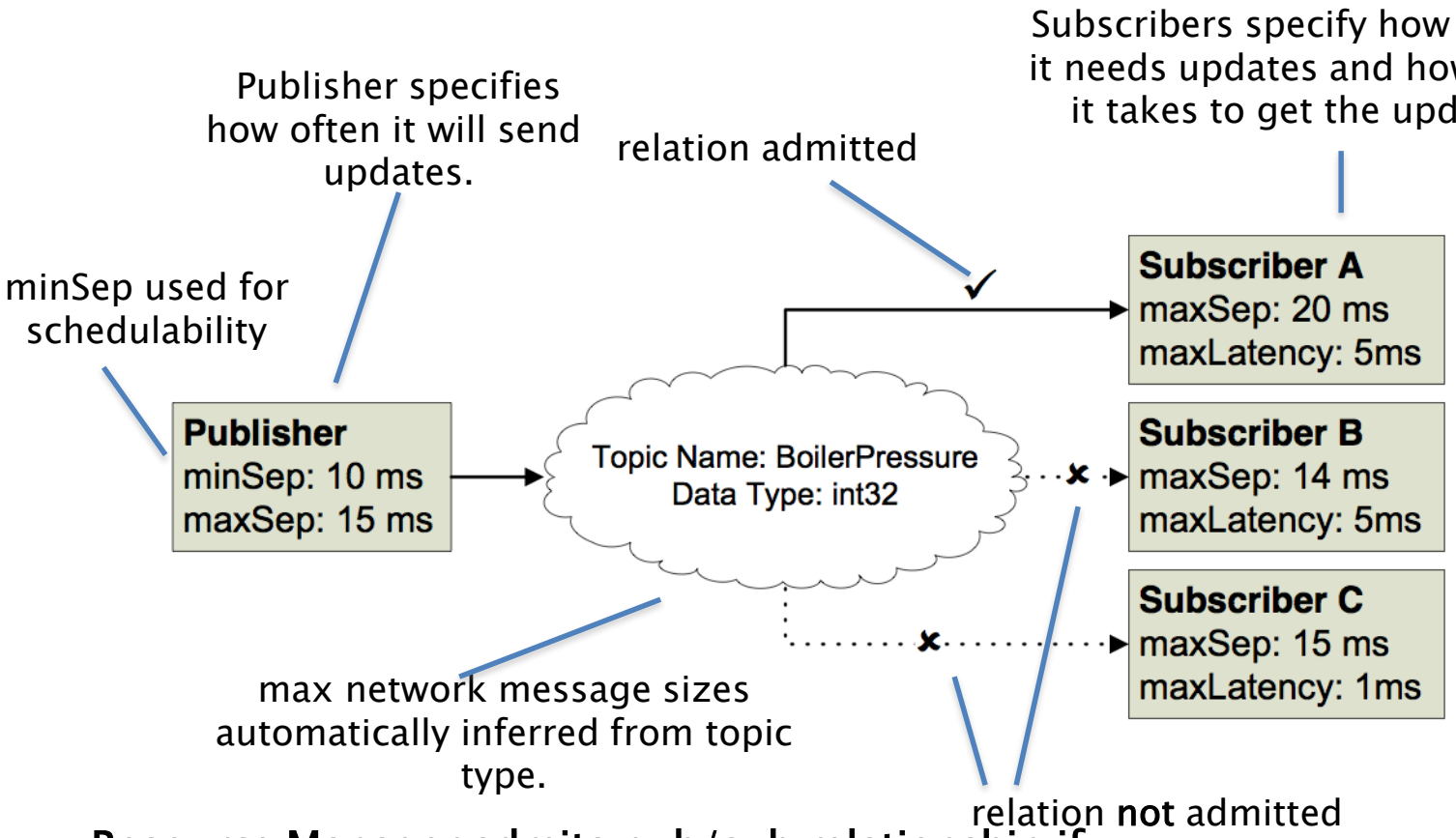
MAP Design: Resource Manager

- 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: MIDleware ASurance 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



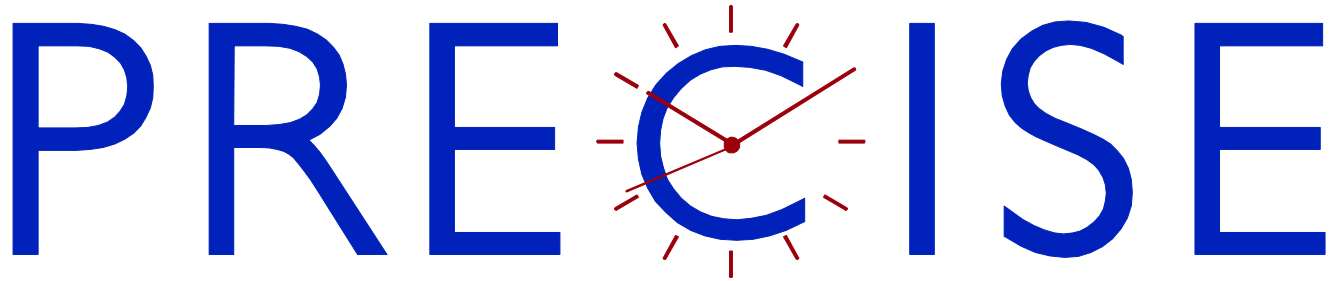
Resource Manager admits pub/sub relationship if:

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

Summary

- The need for medical device interoperability
  - Improve patient care and safety
  - Increase caregivers' productivity
- Medical device plug-and-play open systems
  - Vender neutrality based on open medical device interfaces
  - Integrated Clinical Environment (ICE) standard
- Safety assurance for MD PnP systems
  - A new system integration paradigm where traditional safety assurance approaches won't scale
  - Our solution: The Platform Approach
    - Maintain a curated ecosphere of Devices, Apps, and Platforms
    - Propose a new ***Safety Assurance Argument Pattern***
    - Define a new App modeling language
    - Prototype implementation of Medical App Platform

Thank You!
Questions?



PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

http://precise.seas.upenn.edu