

The PCA Interlock App in AADL

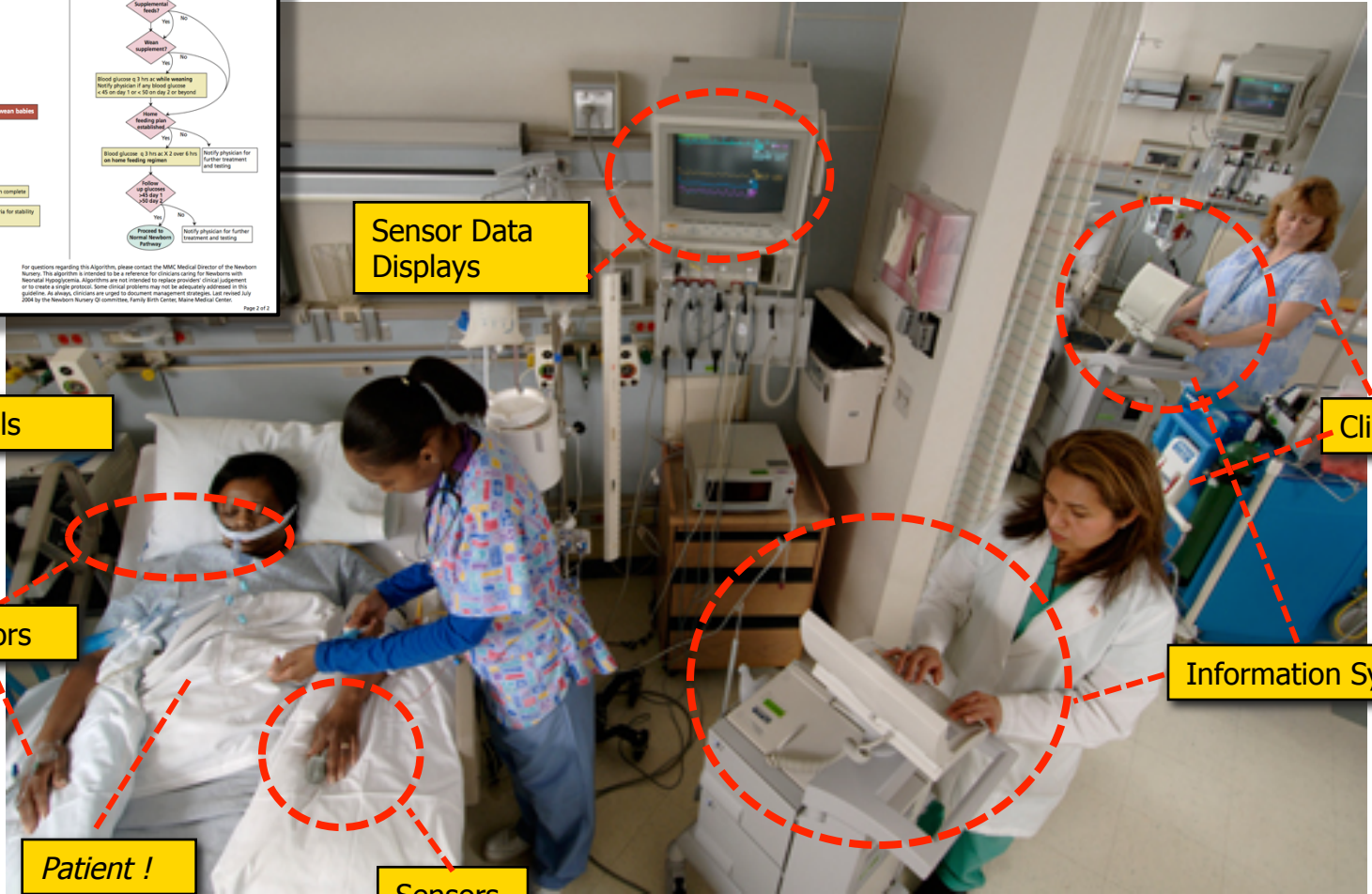
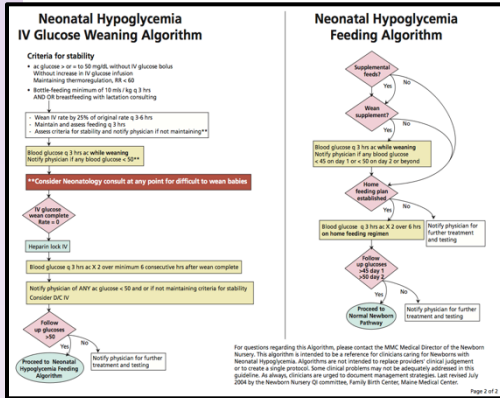
<http://cis.ksu.edu/~samprocter>

Sam Procter, John Hatcliff, Stephen Barret,
and Robby
SAnToS Lab
Kansas State University

Support:

This work is supported in part by the US National Science Foundation (NSF) (#1239543), the NSF US Food and Drug Administration Scholar-in-Residence Program (#1355778) and the National Institutes of Health / NIBIB Quantum Program.

Health Care Involves A Variety of System Components



Sensor Data Displays

Clinical Protocols

Clinicians

Information Systems

Actuators

Patient !

Sensors

Motivation

- What are the types of things we could do with device integration?
 - Information forwarding
 - Automation of clinical workflows
 - Closed loop control between devices
- Unlike personal computing, medical devices are not designed to work together
- Integrating medical devices would bring myriad benefits
- ... how can we do so safely?

Outline

- Background
 - PCA Interlock Scenario
 - Medical Application Platforms
- Language

PCA Interlock Scenario

- Patients are commonly given patient-controlled analgesics after surgery
- Crucial to care, but numerous issues related to safety
- Data for disabling the pump exists now (just a system invariant) -- we just need to integrate it



Clinically Supported

Motivating Clinical Problem: PCA Overdose



NEWSLETTER

The Official Journal of the Anesthesia Patient Safety Foundation

Volume 21, No. 4, 61-88

Circulation 80,350

Winter 2006-2007

Dangers of Postoperative Opioids

APSF Workshop and White Paper Address Prevention of Postoperative Respiratory Complications

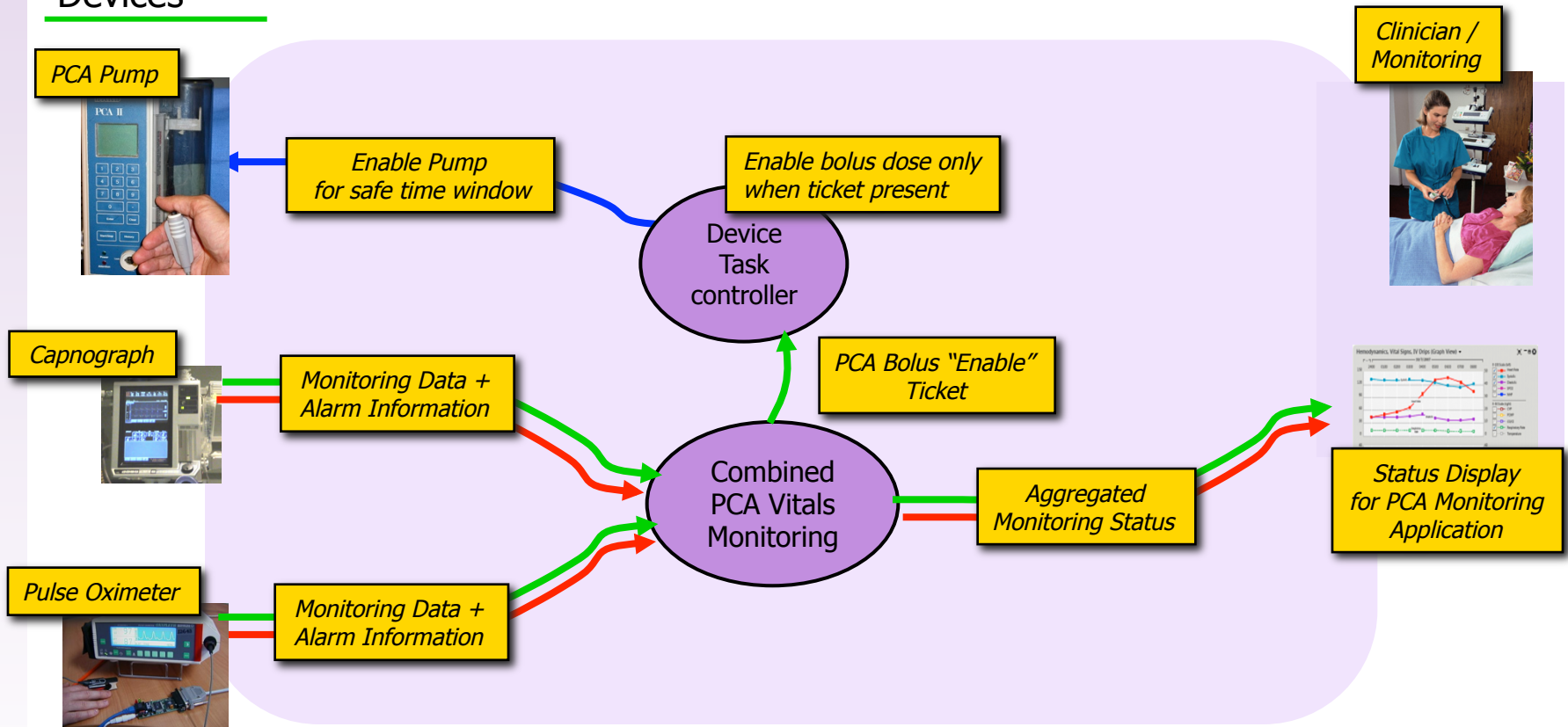
- “A particularly attractive feature may be the ability to automatically terminate or reduce PCA (or PCEA) infusions when monitoring technology suggests the presence of opioid-induced respiratory depression. To facilitate such capabilities, we strongly endorse the efforts to develop international standards for device interoperability and device-device communication.
- It is critical that any monitoring system be linked to a reliable process to summon a competent health care professional to the patient's bedside in a timely manner. “



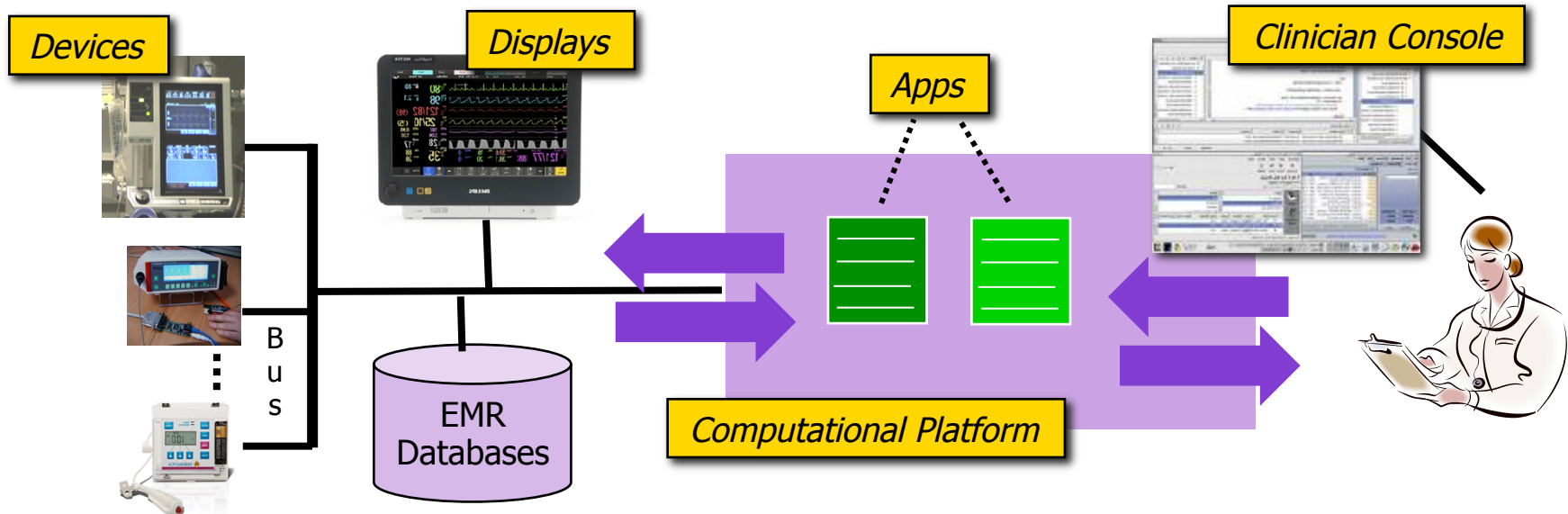
PCA Pump Safety Interlock

Fully leverage device data streams and the ability to *control* devices

Devices



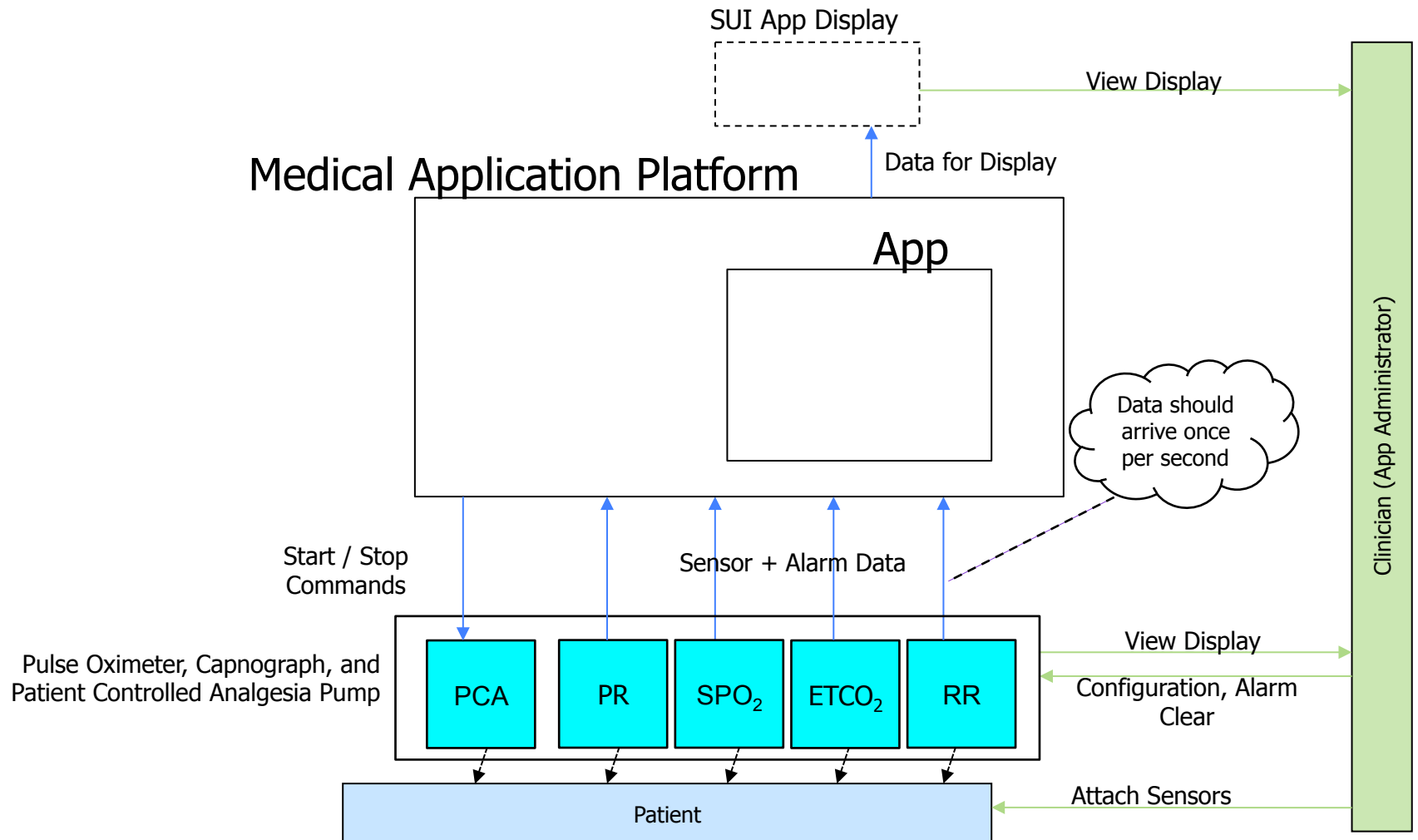
Medical Application Platforms



- A *Medical Application Platform* is a safety- and security-critical real-time computing platform for...
 - Integrating heterogeneous devices, medical IT systems, and information displays via communications infrastructure, and
 - Hosting applications (“apps”) that provide medical utility via the ability to acquire information from and update/control integrated devices, IT systems, and displays

Background

PCA Pump Interlock Architecture

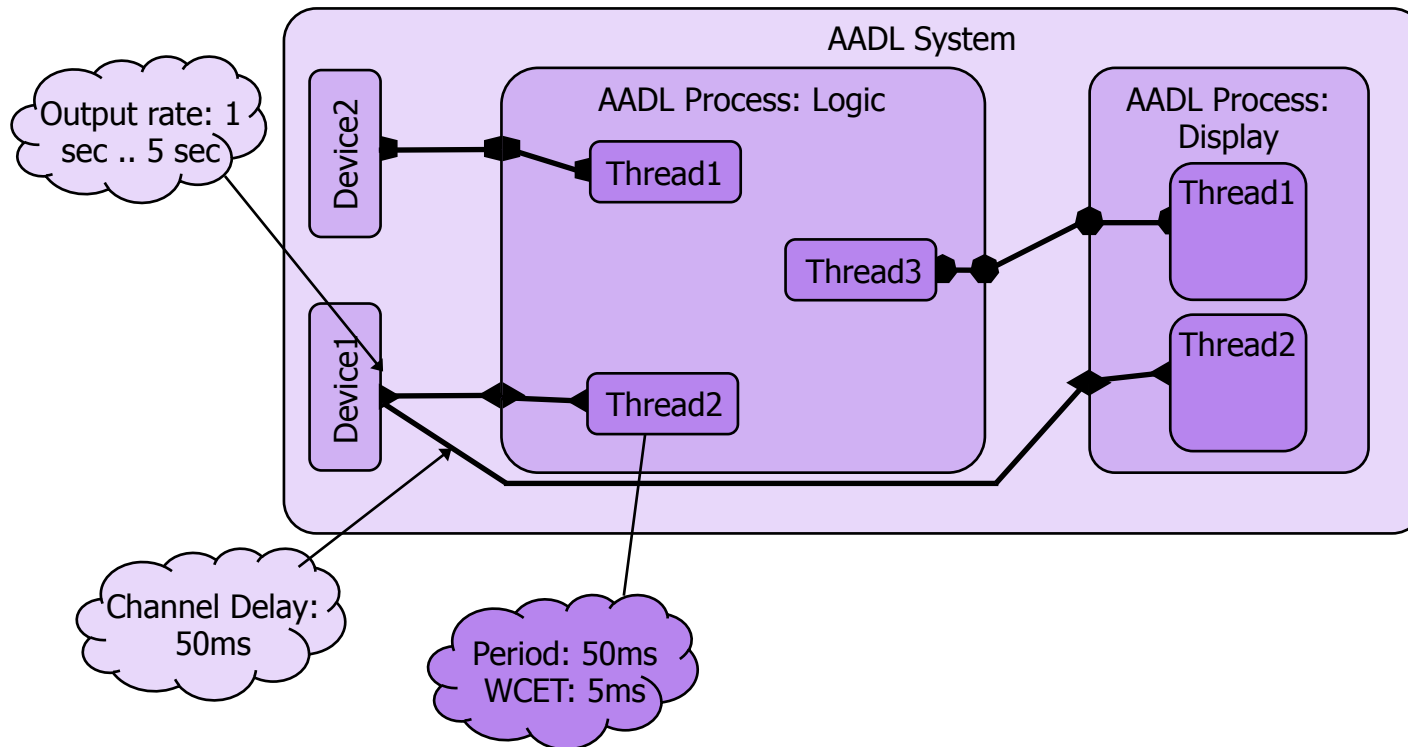


Outline

- Background
- Language

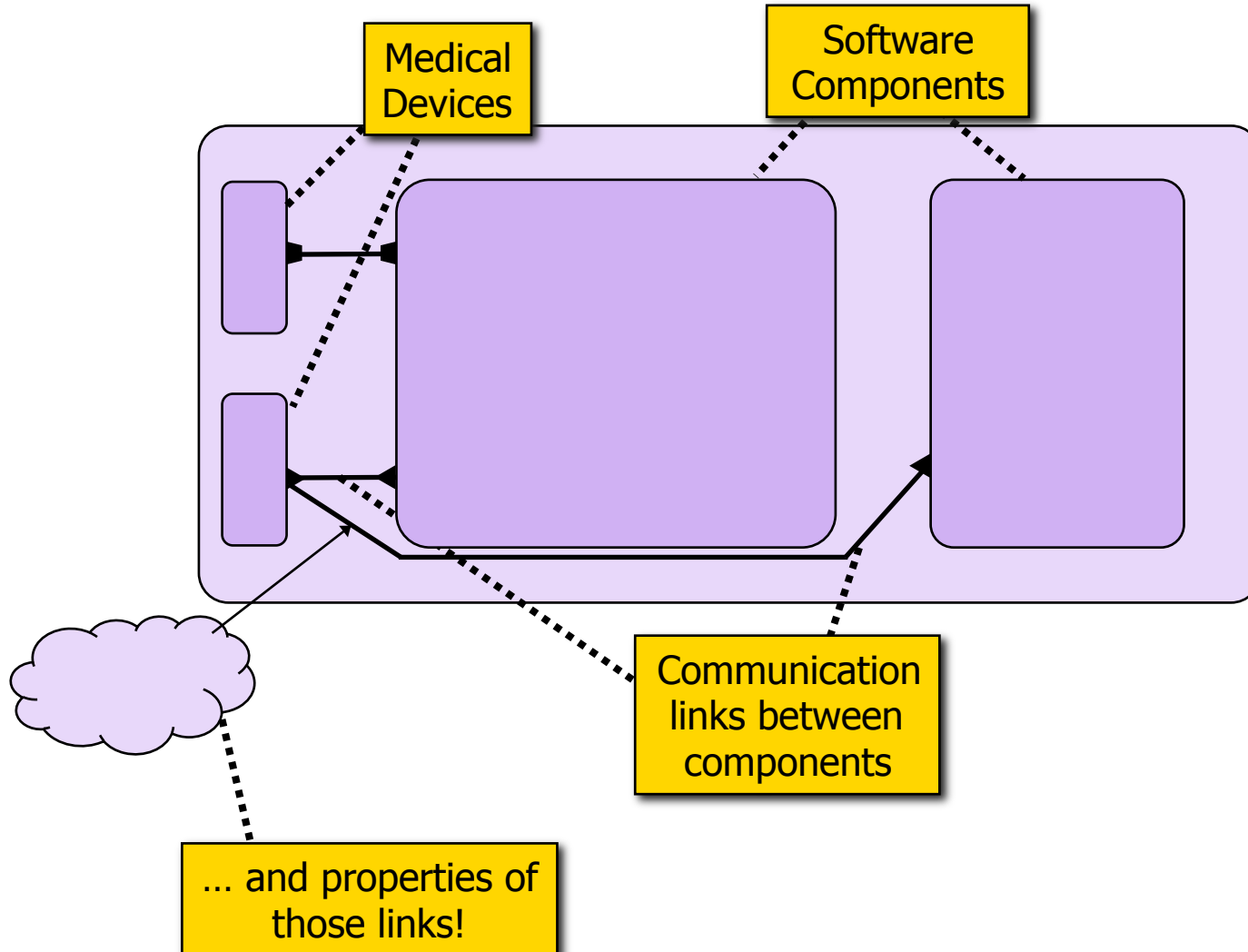
Language

Model



Language

System



Language

System

```
package PCA_Shutoff
public
with PulseOx_Interface, PCAPump_Interface, PCA_Shutoff_Logic,
    PCA_Shutoff_Properties, MAP_Error_Properties, PCA_Shutoff_Display,
    PCA_Shutoff_Errors, Capnograph_Interface, MAP_Errors,
    PCA_Shutoff_Error_Properties;

system PCA_Shutoff_System
end PCA_Shutoff_System;

system implementation PCA_Shutoff_System.imp
subcomponents
    -- Physiological inputs
    capnograph : device Capnograph_Interface::ICEcapnographInterface.imp;

    -- App Logic
    appLogic : process PCA_Shutoff_Logic::ICEpcaShutoffProcess.imp;
    appDisplay : process PCA_Shutoff_Display::ICEpcaDisplayProcess.imp;

    -- Controlled device
    pcaPump : device PCAPump_Interface::ICEpcaInterface.imp;
connections
    -- From components to Logic
    respiratoryrate_logic : port capnograph.RespiratoryRate -> appLogic.RespiratoryRate;
    pumpcommand_logic : port appLogic.CommandPumpNormal -> pcaPump.PumpNormally;
    etco2_logic : port capnograph.ETCO2 -> appLogic.ETCO2
    {MAP_Properties::Channel_Delay => 50 ms;};

    -- From components to display
    pumpcommand_disp : port appLogic.CommandPumpNormal -> appDisplay;
end PCA_Shutoff_System.imp;
end PCA_Shutoff;
```

Medical
Devices

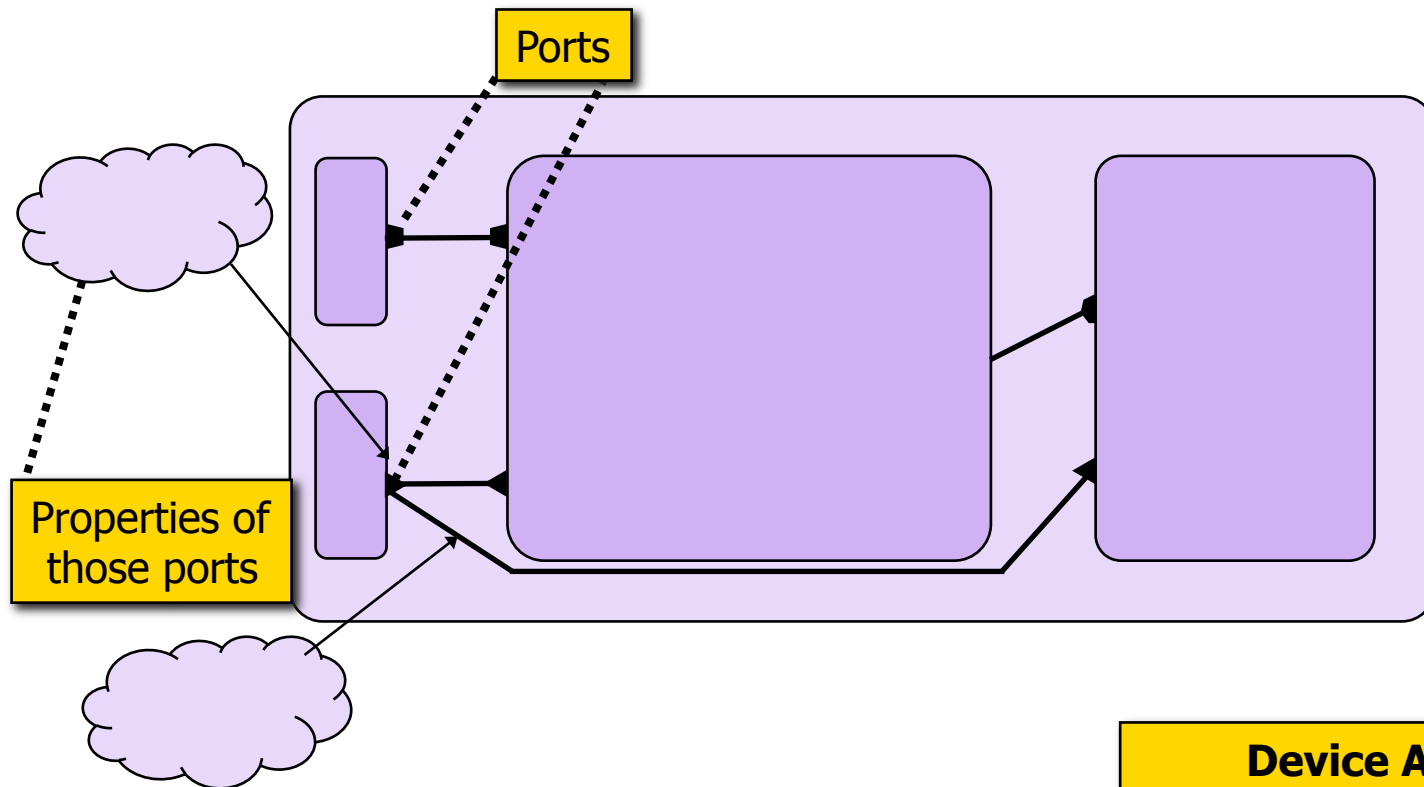
Software
Components

Communication
links between
Components

... and properties of
those links!

Language

Device Interface Specification



Device API Only --

Presents the app's view of the required device capabilities, *not the full device capabilities*

Language

Device Interface Specification

```
package Capnograph_Interface
public
with PCA_Shutoff_Types;
device ICEcapnographInterface
features
    ETCO2 : out event data port PCA_Shutoff_Types::ETCO2;
    RespiratoryRate : out event data port PCA_Shutoff_Types::RespiratoryRate;
end ICEcapnographInterface;

device implementation ICEcapnographInterface.imp
end ICEcapnographInterface.imp;
end Capnograph_Interface;
```

Device API Only --
Presents the app's view of
the required device capabilities,
not the full device capabilities

Ports

```
package PCAPump_Interface
public
with PCA_Shutoff_Types;
device ICEpcaInterface
features
    PumpNormally : in event data port PCA_Shutoff_Types::PumpNormalCommand
    {MAP_Properties::Output_Rate => 50 ms .. 75 ms};
end ICEpcaInterface;

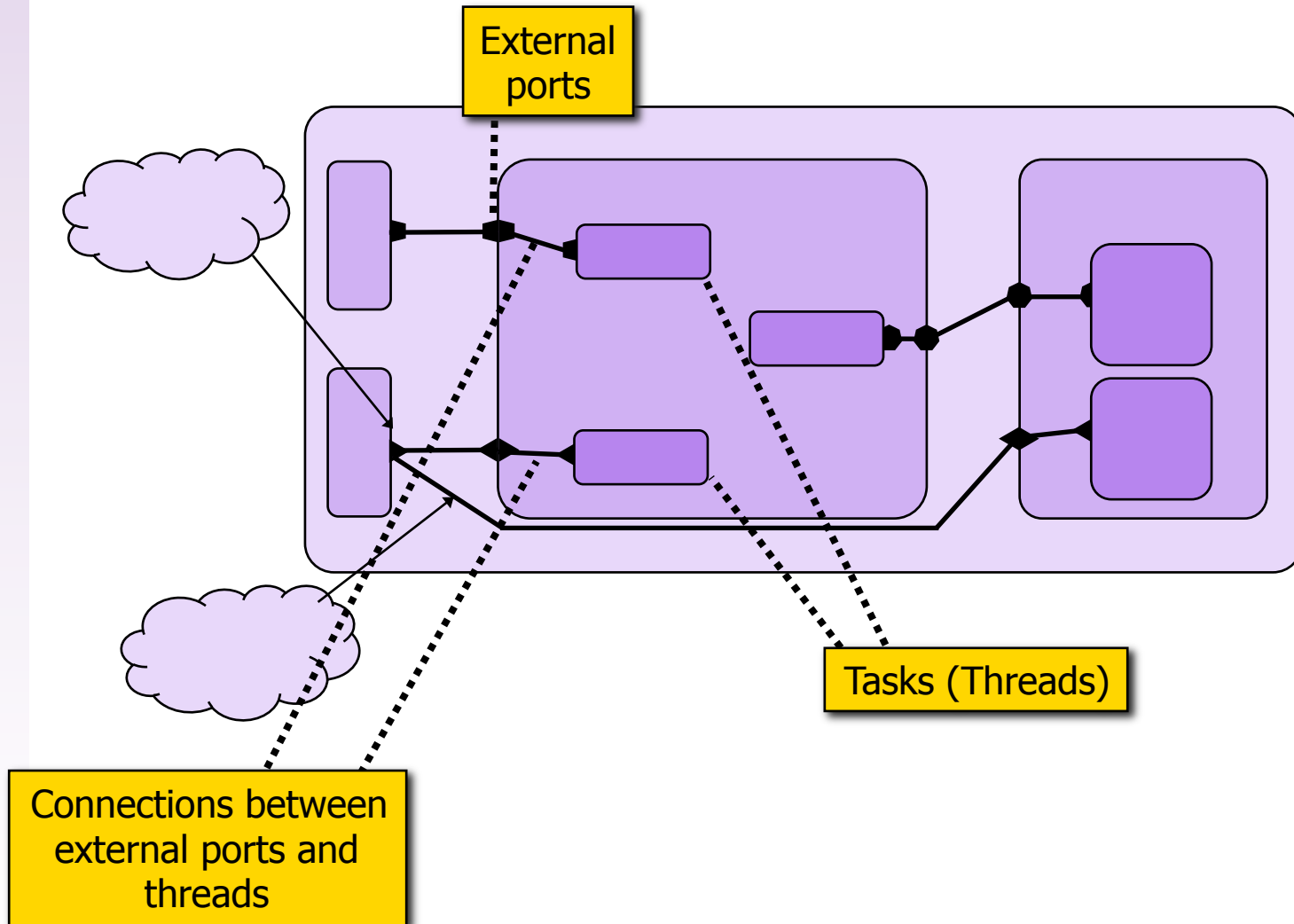
device implementation ICEpcaInterface.imp
end ICEpcaInterface.imp;

end PCAPump_Interface;
```

Properties on
those ports

Language

Process Specification



Language

Process Specification

External ports

```
PCA_Shutoff_Logic
PCA_Shutoff_Types, PCA_Shutoff_Properties, MAP_Properties;

process ICEpcaShutoffProcess
features
  -- No ETCO2 thread / connection because it's a data port
  ETCO2 : in data port PCA_Shutoff_Types::ETCO2;
  RespiratoryRate : in event data port PCA_Shutoff_Types::RespiratoryRate;

  CommandPumpNormal : out event data port PCA_Shutoff_Types::PumpNormalCommand;
properties
  MAP_Properties::Component_Type => logic;
end ICEpcaShutoffProcess;

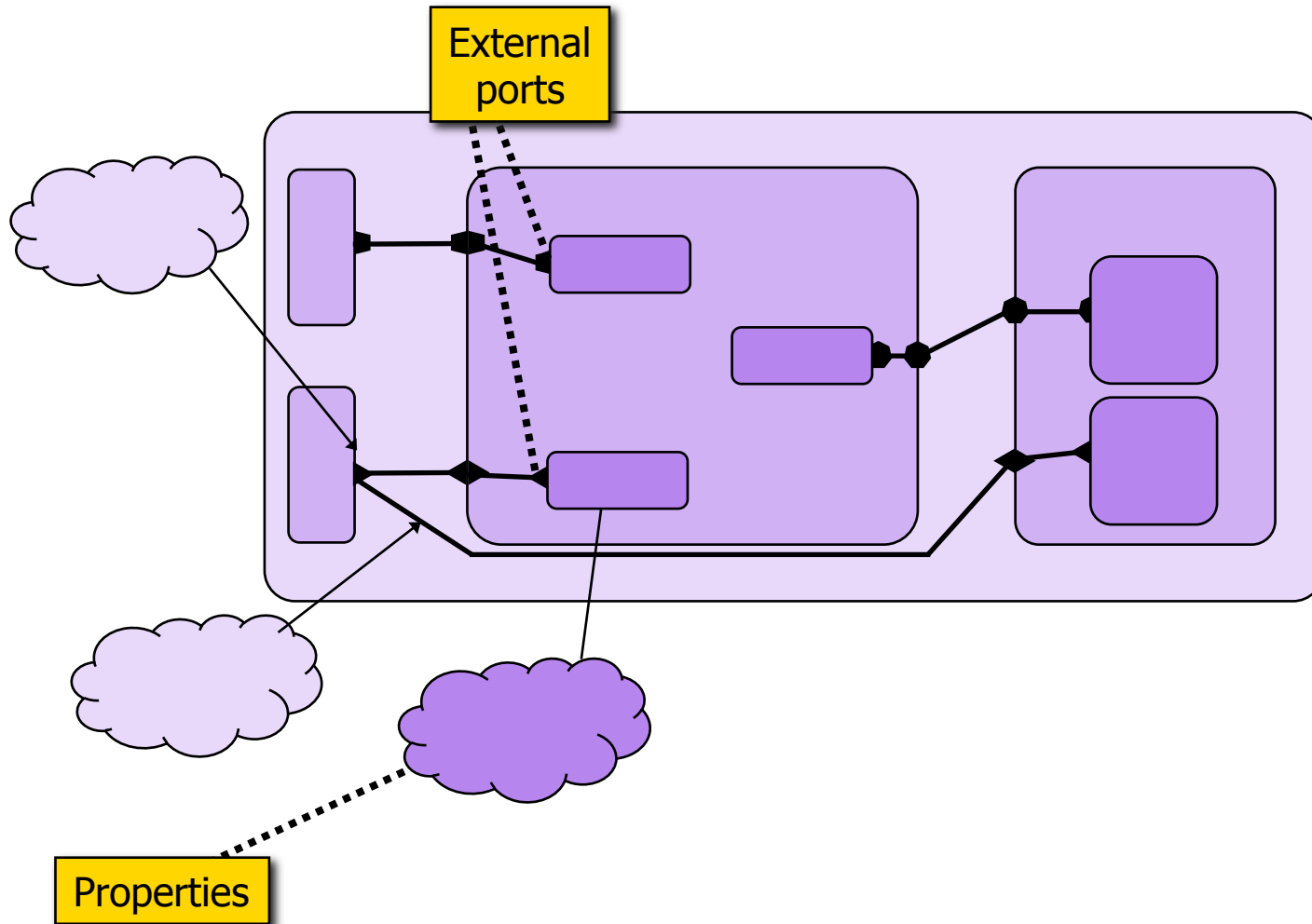
process implementation ICEpcaShutoffProcess.imp
subcomponents
  UpdateRespiratoryRateThread : thread UpdateRespiratoryRateThread.imp;
  PumpControlThread : thread PumpControlThread.imp;
connections
  incoming_rr : port RespiratoryRate -> UpdateRespiratoryRateThread.RespiratoryRate;
  outgoing_pump_command : port PumpControlThread.PumpNormal -> CommandPumpNormal;
end ICEpcaShutoffProcess.imp;
```

Tasks (Threads)

Connections between external ports and threads

Language

Thread Specification



Language

Thread Specification

External
ports

```
thread UpdateRespiratoryRateThread
features
  RespiratoryRate : in event data port PCA_Shutoff_Types::RespiratoryRate;
properties
  Timing_Properties::Deadline => 75 ms;
  Timing_Properties::Period => 95 ms;
  MAP_Properties::Worst_Case_Execution_Time => 7 ms;
  Thread_Properties::Dispatch_Protocol => Sporadic;
end UpdateRespiratoryRateThread;

thread implementation UpdateRespiratoryRateThread.imp
end UpdateRespiratoryRateThread.imp;

thread PumpControlThread
features
  PumpNormal : out event data port PCA_Shutoff_Types::PumpNormalCommand;
properties
  Timing_Properties::Deadline => 50 ms;
  Timing_Properties::Period => 105 ms;
  MAP_Properties::Worst_Case_Execution_Time => 10 ms;
  Thread_Properties::Dispatch_Protocol => Periodic;
end PumpControlThread;

thread implementation PumpControlThread.imp
end PumpControlThread.imp;
```

Properties

Any necessary architectural
annotations can be created!

Language Subset

AADL Constructs Used

AADL Construct	MAP Concept
Components	
System	Layout
Device	Medical Device API for App
Process	Software Component
Thread	Task
Connections	
System-level port connection	Channel
Process-level port connection	Task Trigger
Process implementation-level port connection	Task-Port Communication