Self-Managed Cells and their Federation

Prof. J. Sventek
University of Glasgow
joe@dcs.gla.ac.uk

In collaboration with M. Sloman, E. Lupu, and N. Dulay of Imperial College London

Outline

- Healthcare application scenario
- Basic architectural building block - SMC
- Body-area network (SMC in the small)
- SMC federation
- Integration into health information systems (SMC in the large)
- Summary/conclusions

Patient monitoring

- Average age of population in most countries is shifting to higher values (baby boomers)
- Consumption of health care is strongly correlated with age
- Patient recovery from major health events is substantially more successful if this recovery can take place in familiar environs
- Critical care delivery is the most expensive form of health care
- Out-patient monitoring of stable critical care patients should lead to substantially better outcomes while substantially reducing costs.
- Any such system must be integrated into the overall IT infrastructure supporting health care delivery.

Ubiquitous e-Health

Healthcare Everywhere

Wireless
Video Camera Pill

Applications

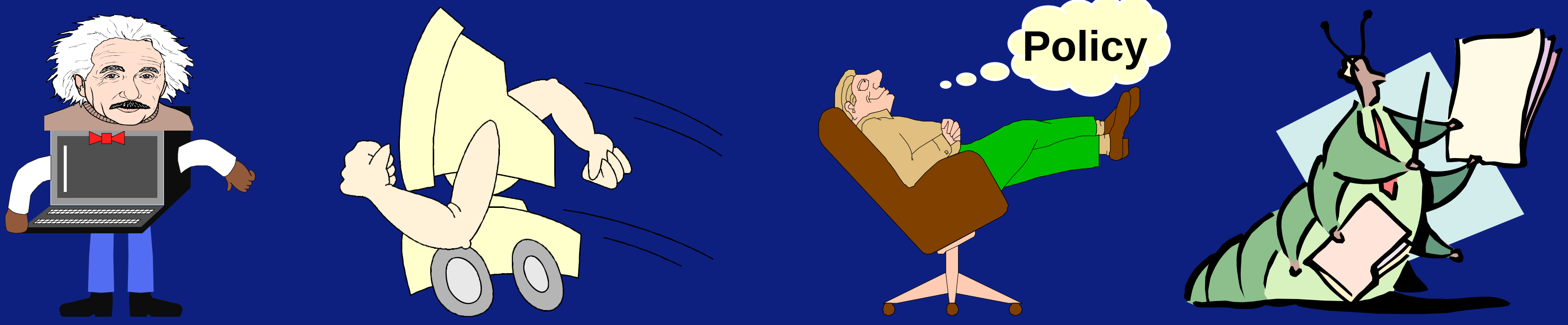
- Automated monitoring
  - Implanted devices
  - Smart clothing
  - Swallow/inject intelligent sensors and actuators
  - Reaction to complex drug regimes

Benefits

- High → lower risk monitoring
- Mobility for chronically ill
- Greater out-of-hospital patient management
- Mass data & analysis
- Emergency feedback or response

Autonomic Management

- **Autonomic** – self-organising, self-configuring, self-healing, self-optimising, *adaptive* management
- Remove human from the loop
- “Intelligent” agents, mobile agents, policy, genetic algorithms?



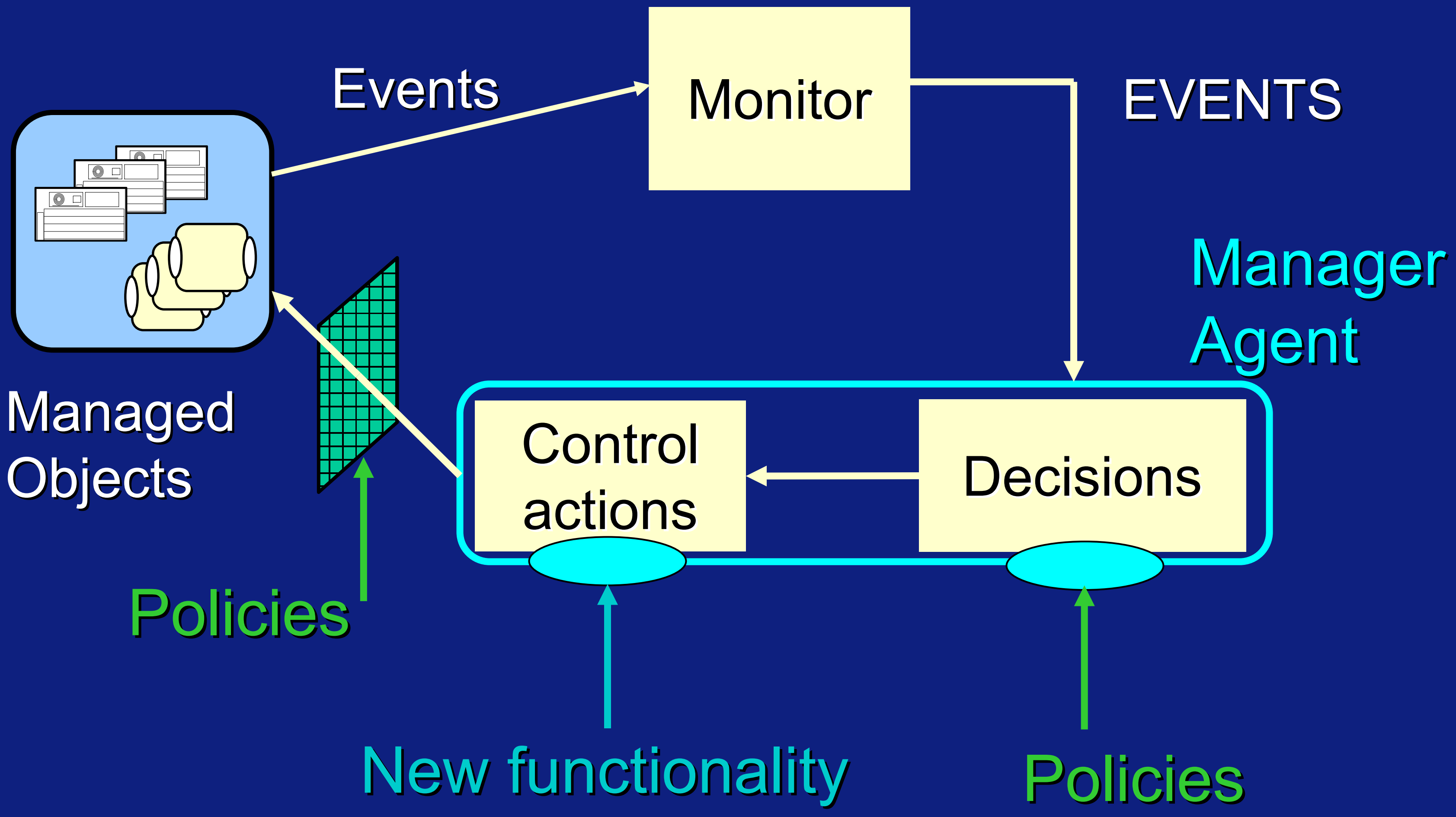
☞ Policy-based control loops

☞ Self-Managed Cell (SMC) architecture

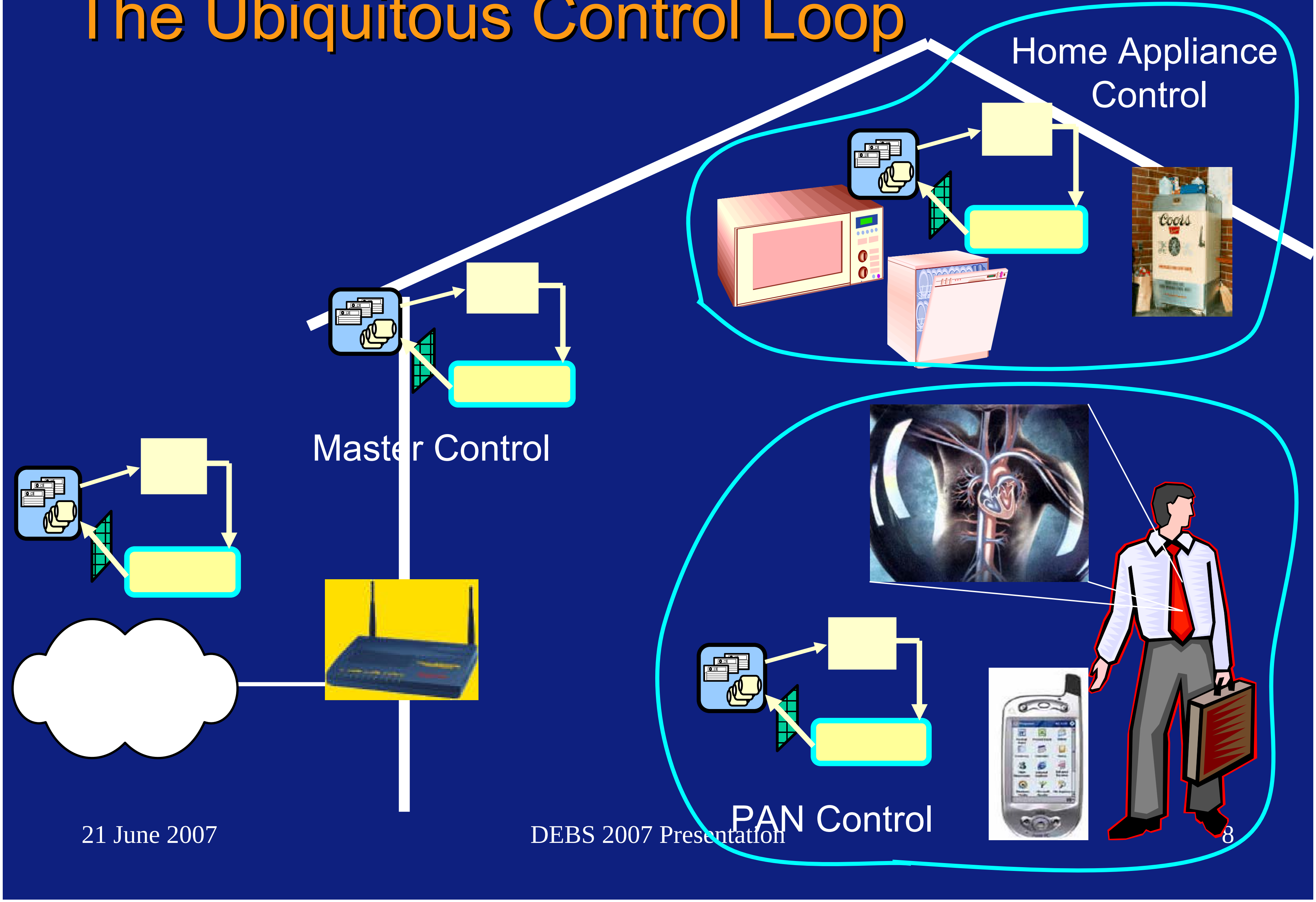
Basic Model

- Patient wears non-invasive sensors as specified by his/her physician (same measurements that would be taken in the critical care unit)
- These sensors are knitted into an autonomic, body-area, distributed system
- Each such autonomic system (a self-managed cell [SMC]) federates with appropriate other SMCs; in particular, each federates with the SMC that represents the IT infrastructure supporting the healthcare delivery system
- The connectivity that enables this federation (in particular, connectivity back to the responsible care giver) must be spectrum and protocol agnostic

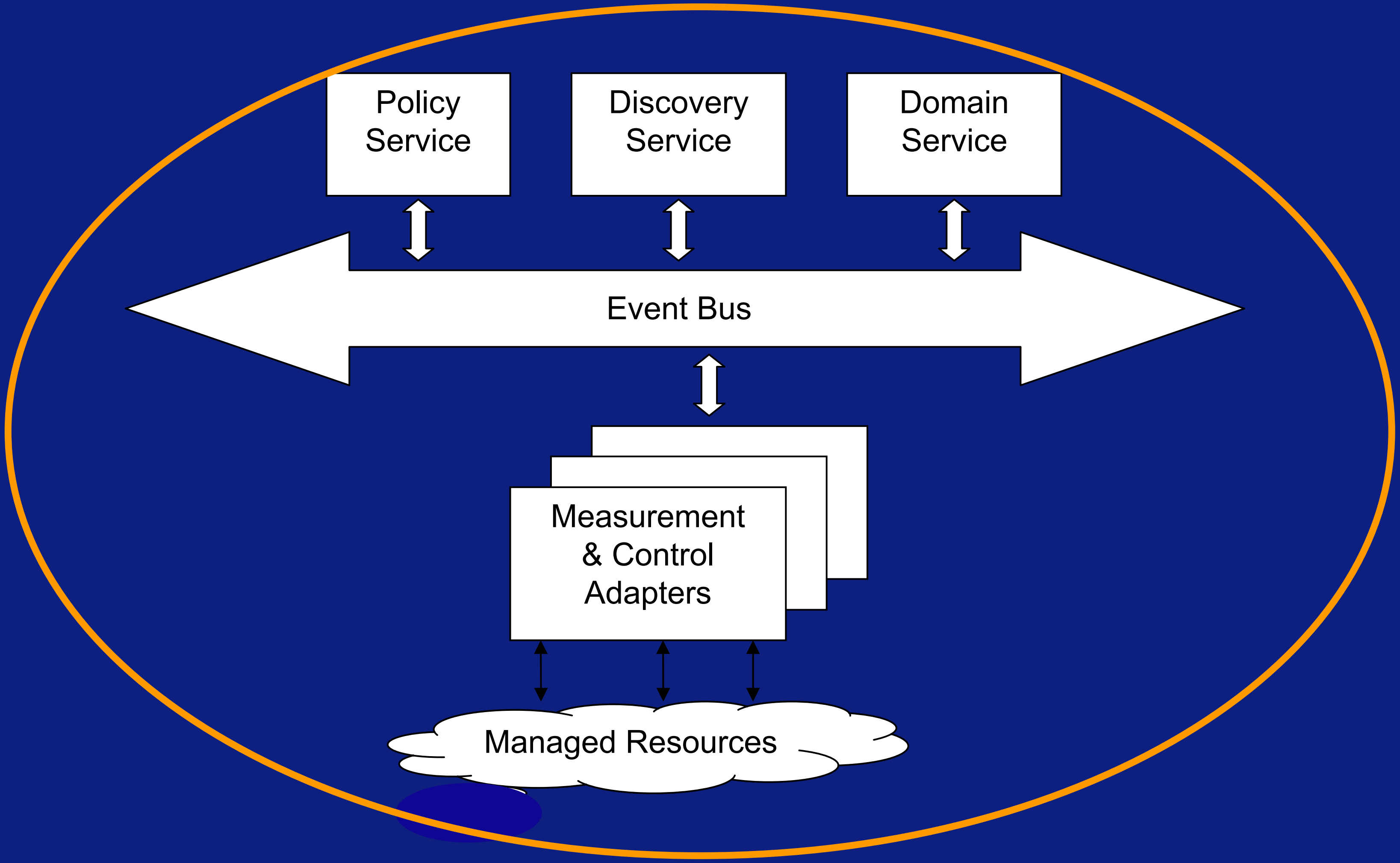
Policy-Based Management



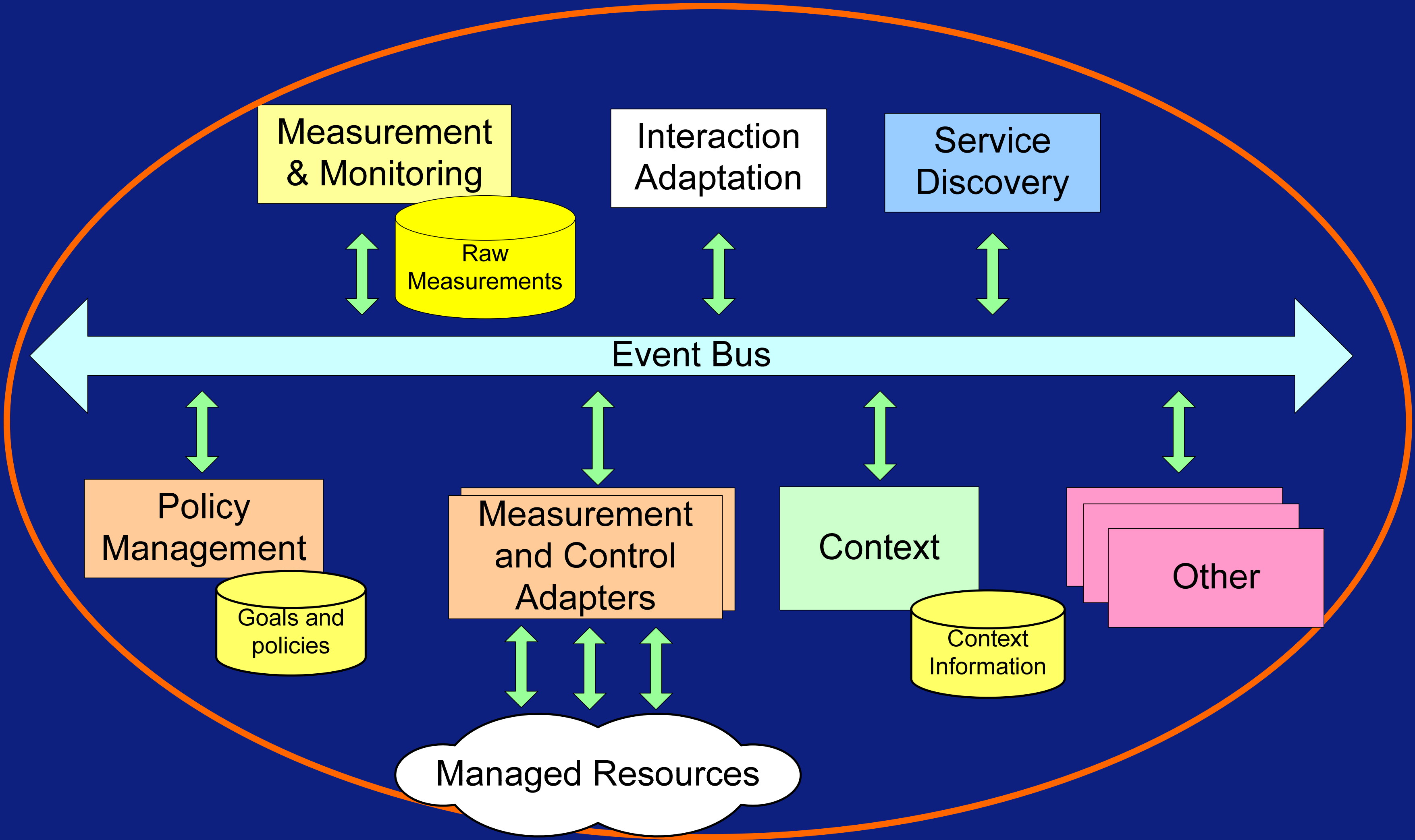
The Ubiquitous Control Loop



Basic Self-Managed Cell



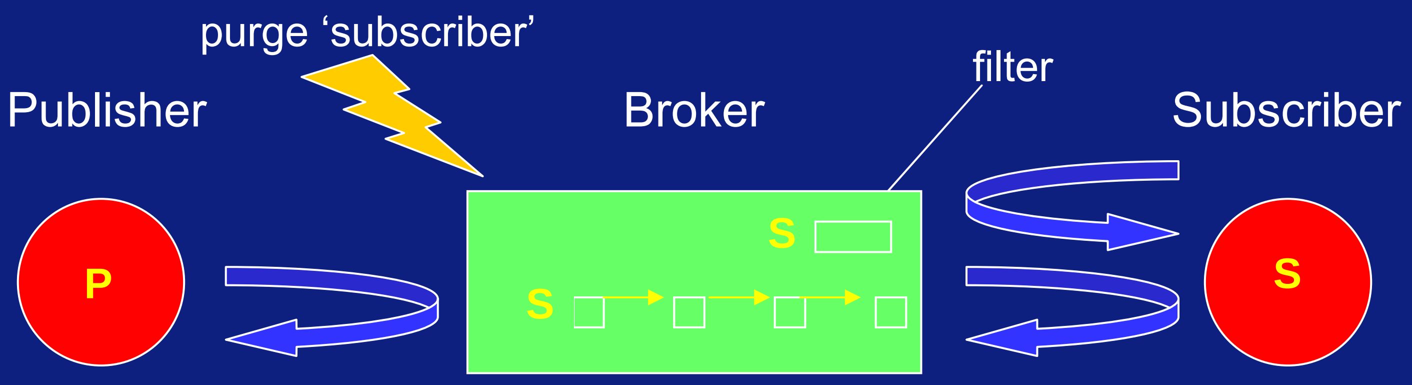
General Self-Managed Cell



Body-area Architecture

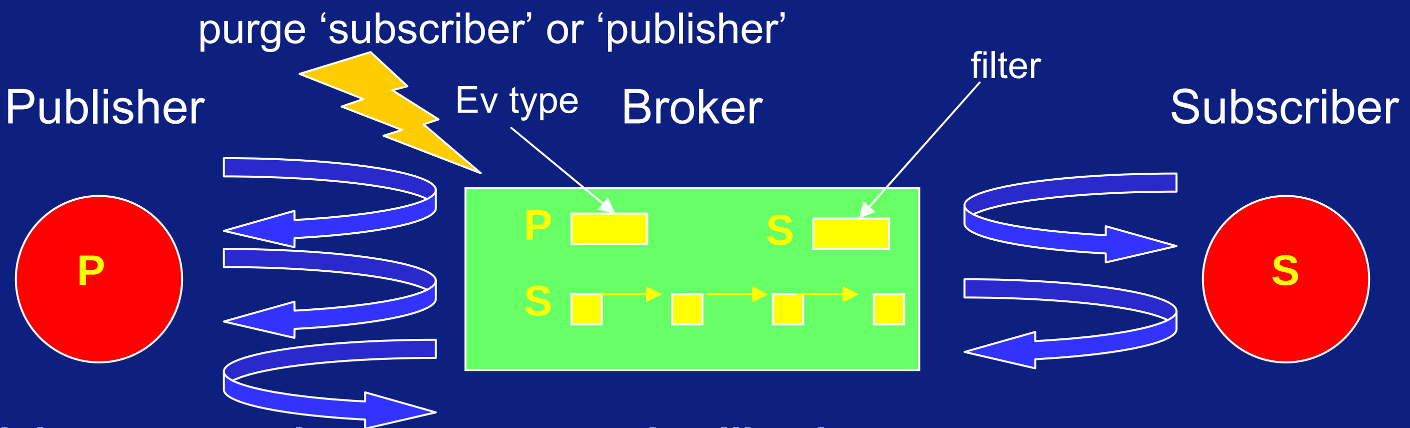
- Event bus is publish/subscribe using a broker
- The broker is content-based
- A discovery/membership service is concerned with keeping track of which devices and services are “in” a self-managed cell
- Each device has a unique identifier (e.g. 802.* MAC address of one of the communication interfaces)

At-most-once, persistent event delivery



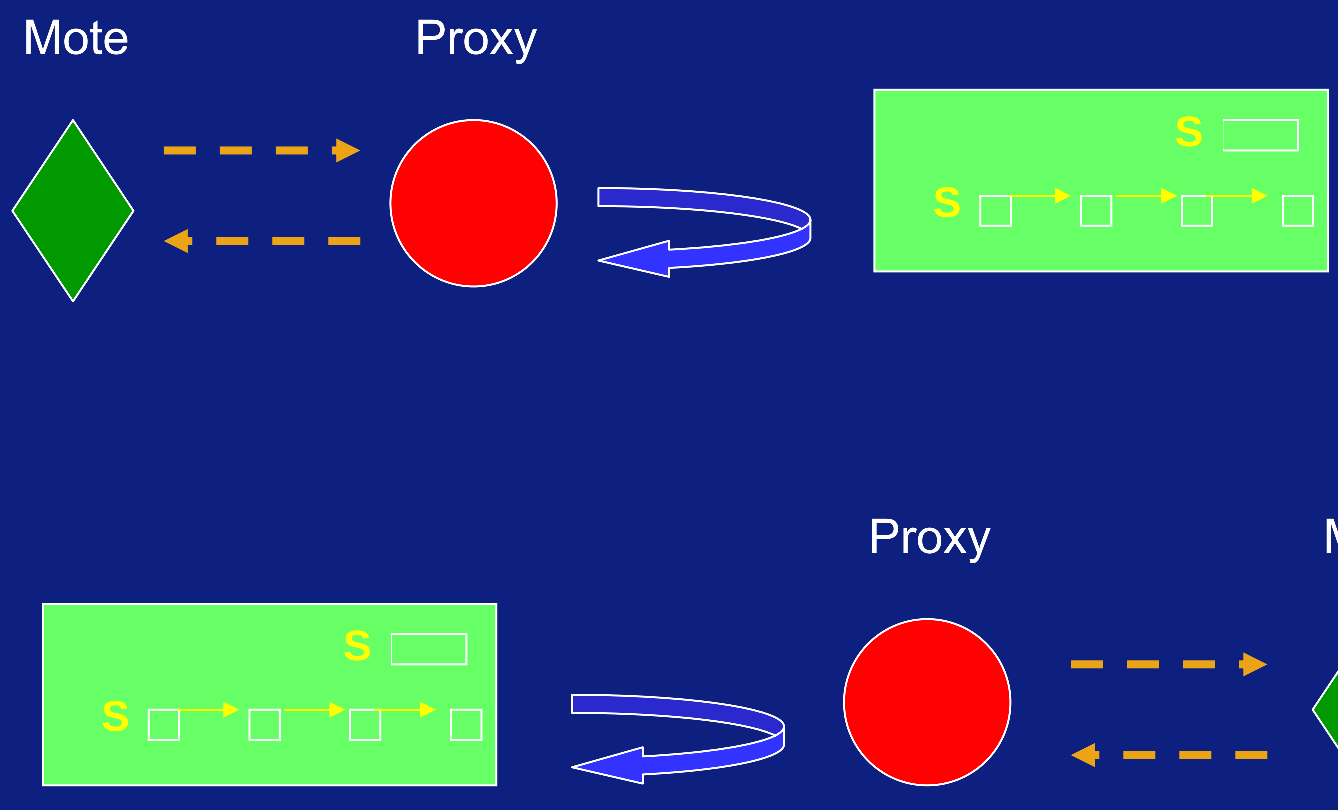
- No session establishment for Publisher
- Subscriber must register ‘filter’ and callback
- Push of event from Publisher to Broker (and Broker to Subscriber) is synchronous – i.e. exception condition is returned to sender if unsuccessful
- Broker attempts to deliver a message until it knows that a Subscriber is no longer a member of the SMC
- When purge event received, removes ‘filter’ and any queued messages associated with that Subscriber
- Each Subscriber is guaranteed to receive all messages from a particular publisher in the same order as received by the Broker

At-most-once, persistent, quenchable event delivery



- Publisher must register 'Ev type' and callback
- Subscriber must register 'filter' and callback
- Push of event from Publisher to Broker (and Broker to Subscriber) is synchronous – i.e. exception condition is returned to sender if unsuccessful
- Broker attempts to deliver a message until it knows that a Subscriber is no longer a member of the SMC
- When purge event received
  - If for a subscriber, removes 'filter' and any queued messages associated with that Subscriber
  - If for a publisher, removes 'Ev type'
- Each Subscriber is guaranteed to receive all messages from a particular publisher in the same order as received by the Broker
- Quench/unquench messages sent to Publisher if the number of subscribers matching event type is zero/non-zero.

How to incorporate a sensor/mote into this structure?



Authentication

- performed SMC wide (device/service is a member of the SMC)
- integrity/confidentiality are required in health-care scenarios
- access control – component-specific, done through policies (authorization policies)

Discovery/Membership

- Detect new devices within communication range
- Vette device for membership
  - obtain device profile
  - perform any required authentication
- Generate new cell member event
- Determine when device leaves cell
  - Generate cell member left event
- Discovery protocol DOES NOT use the event system to discover and negotiate with devices; the discovery service DOES use the event service to announce the addition/removal of a member

Where do the new device/service events go?

- The system must be primed with obligation policies that listen for these events
- Upon receipt of one of these events, the action enters the device/service into appropriate domain[s]
- A particular obligation policy will be interested only in particular types of devices or services; new device/service events may trigger several such obligation policies
  - if can specify event type and filter expression upon subscription, then only the particular obligation policy that is interested in that particular device/service type will be notified
  - if cannot specify filter expression to event bus, than all such policies will be invoked; only those for which the condition is true will perform actions

Discovery protocol

- Cell is centred around event bus broker
- Device that contains the broker broadcasts its identity message at frequency ω_B (the identity message has the form
  “id; type[; extra]”)
- Other devices respond to broker identity message with unicast device identity message
- Broker device and other device carry on vetting protocol (obtain profile[; authenticate])

Discovery protocol (cont)

- After other device knows that it has been granted membership, it unicasts its identity message at frequency ω_D
- If broker device misses n_D successive device identity messages, it declares the device to have forfeited its membership in the cell
- If the other device misses n_B successive broker device identity messages, it infers that it is no longer a member of that cell
- Stability of system requires $n_B \bullet \omega_B = n_D \bullet \omega_D$
- Currently exploring ramifications of $\omega_B \neq \omega_D$

Communication primitives required

- Event bus is only used for communications between cell management elements
- Basic communication primitives are required to implement the event bus communications, required protocols, and general communication between application components
  - broadcast, asynchronous messaging
  - multicast, asynchronous messaging
  - unicast, asynchronous messaging
  - remote method invocation
  - reliable, flow-controlled streams

What about services?

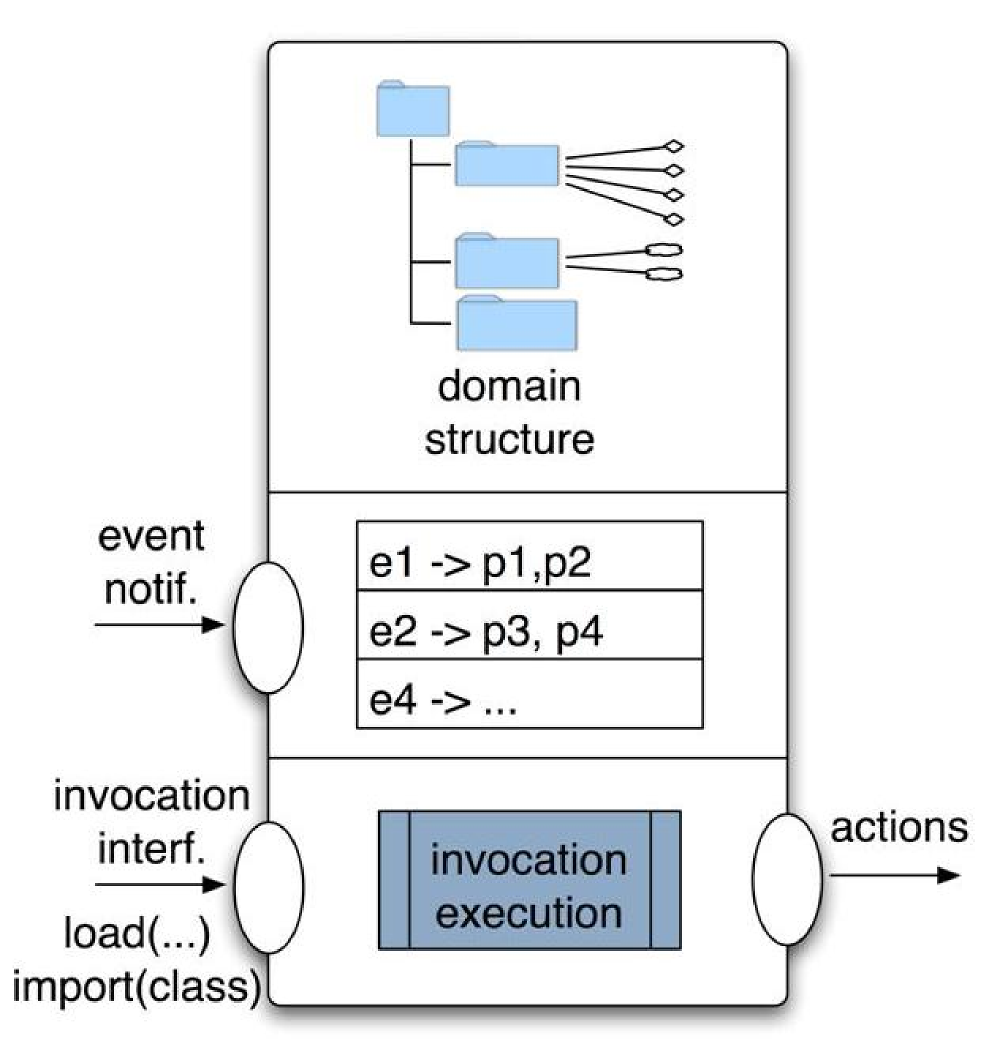
- Devices are discovered by the discovery service.
- When a device becomes part of the cell, it generates events announcing active services that it provides/hosts
- While a member of the cell, each device generates an event whenever another service that it provides/hosts becomes active or if such a service is deactivated

The Policy Service

- two types of policies
  - authorisation policies define what actions are permitted under given circumstances
  - obligation policies define what actions to carry out when specific events occur if certain conditions are fulfilled (ECA rules)
- The general format of an obligation policy follows that of traditional ECAs:

```
on <event> do
    if <condition> then
        <action sequence>
```

Policy Service Architecture



domain structure

event notif.

e1 -> p1,p2

e2 -> p3, p4

e4 -> ...

invocation interf.

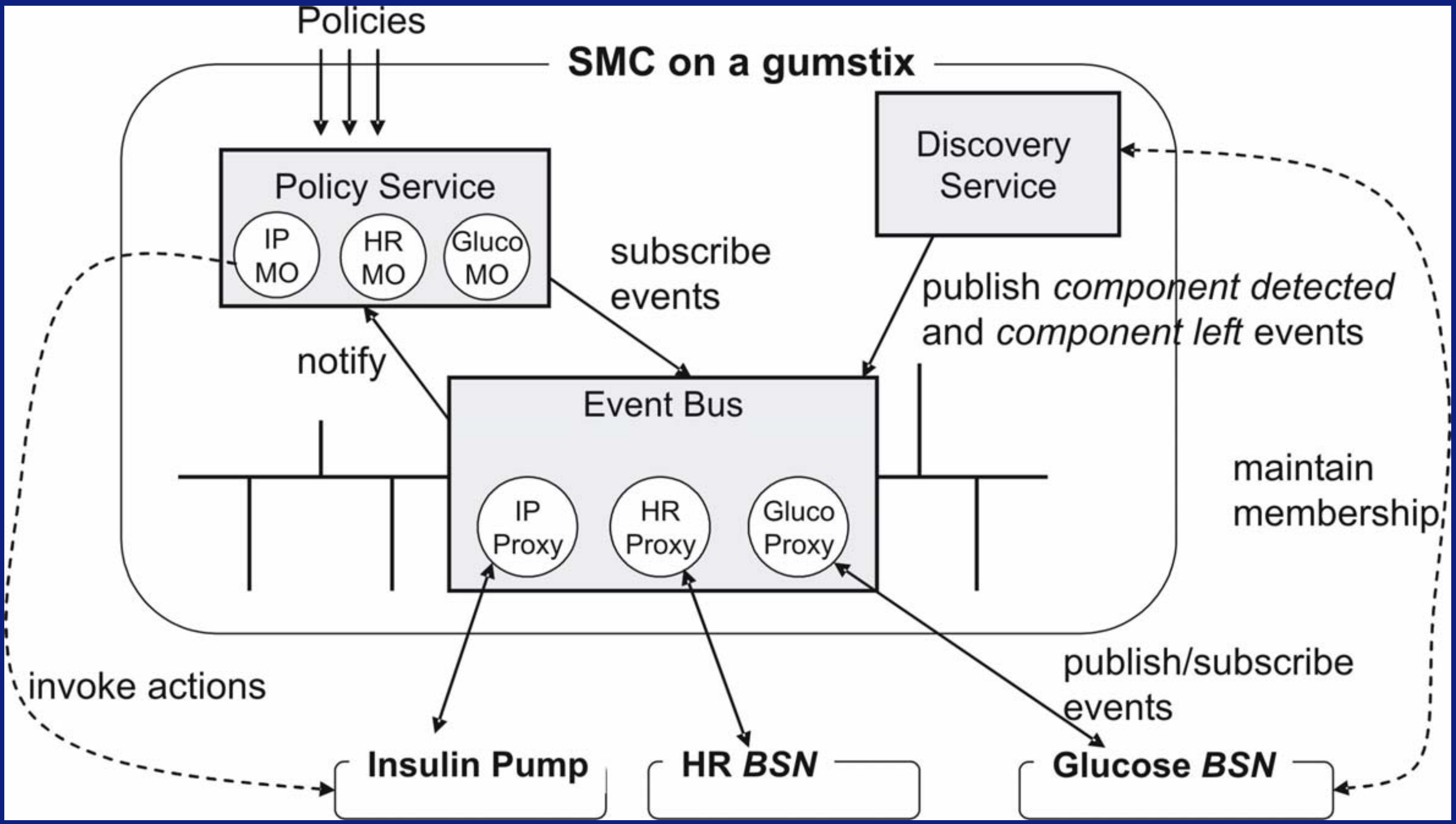
load(...)

import(class)

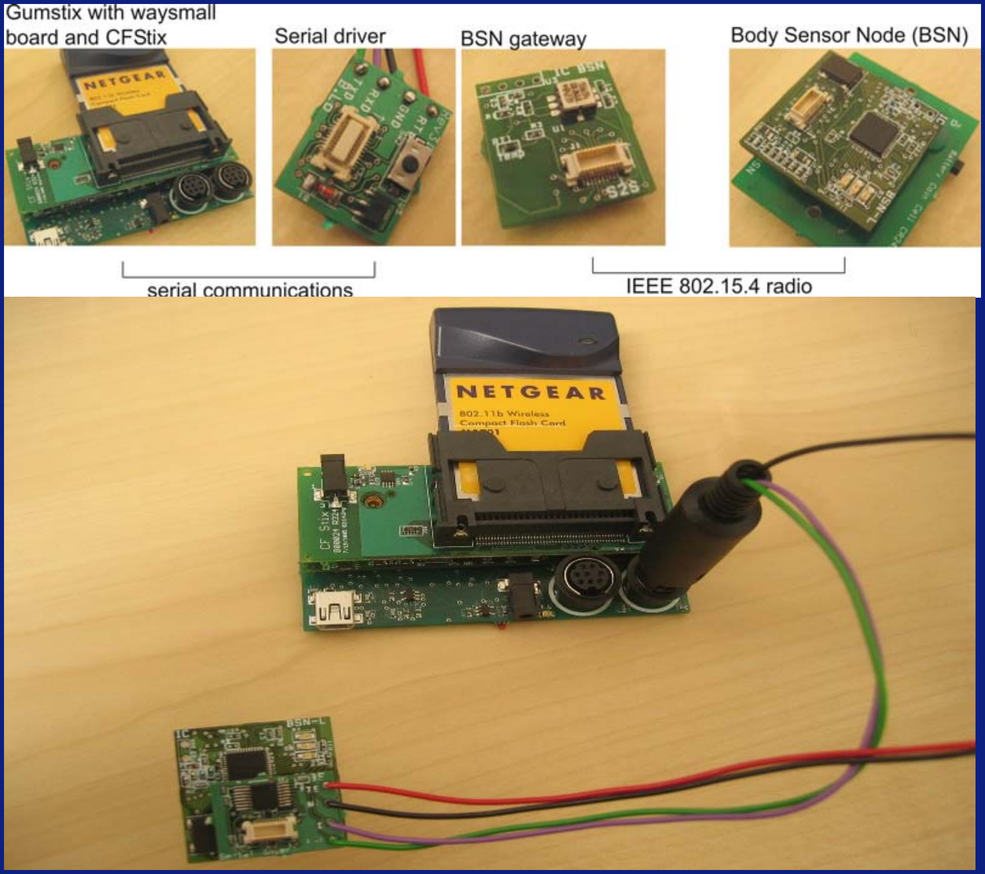
invocation execution

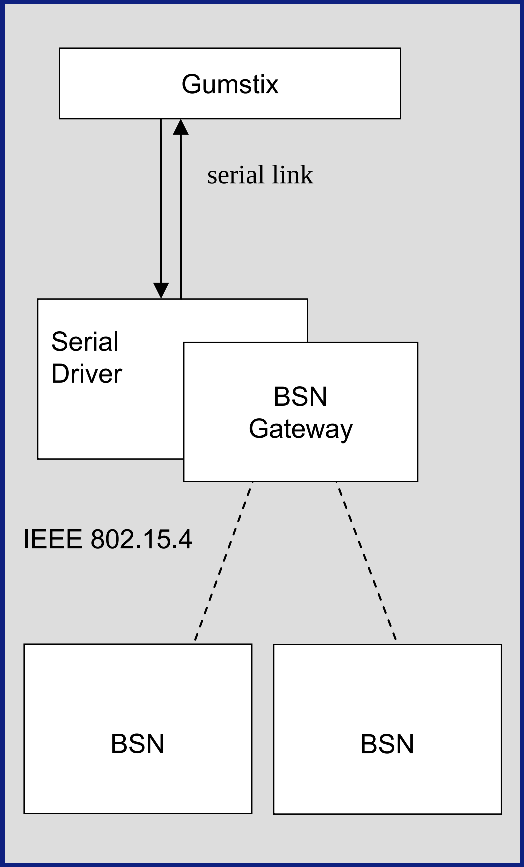
actions

SMC on a gumstix

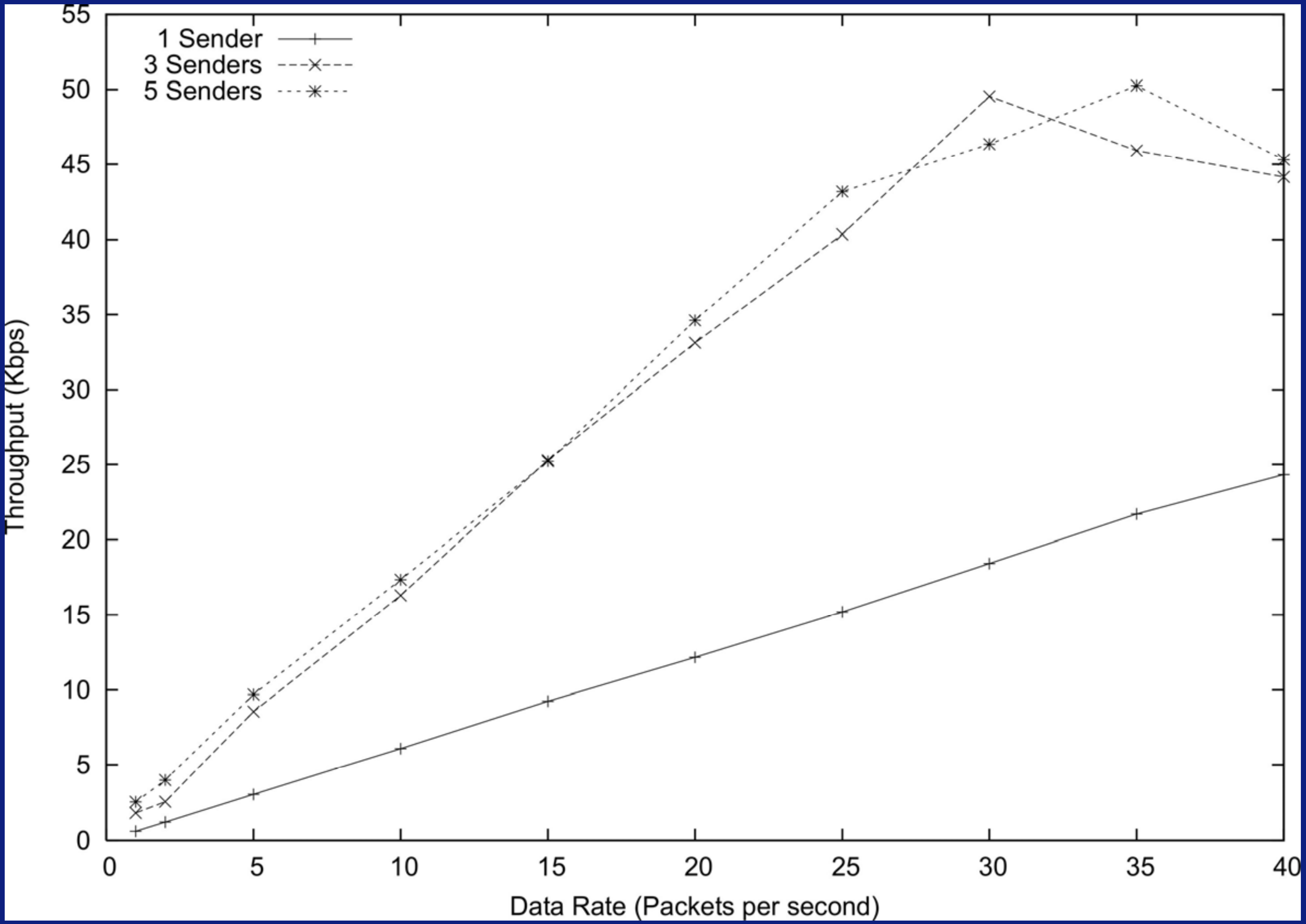


Hardware Configuration





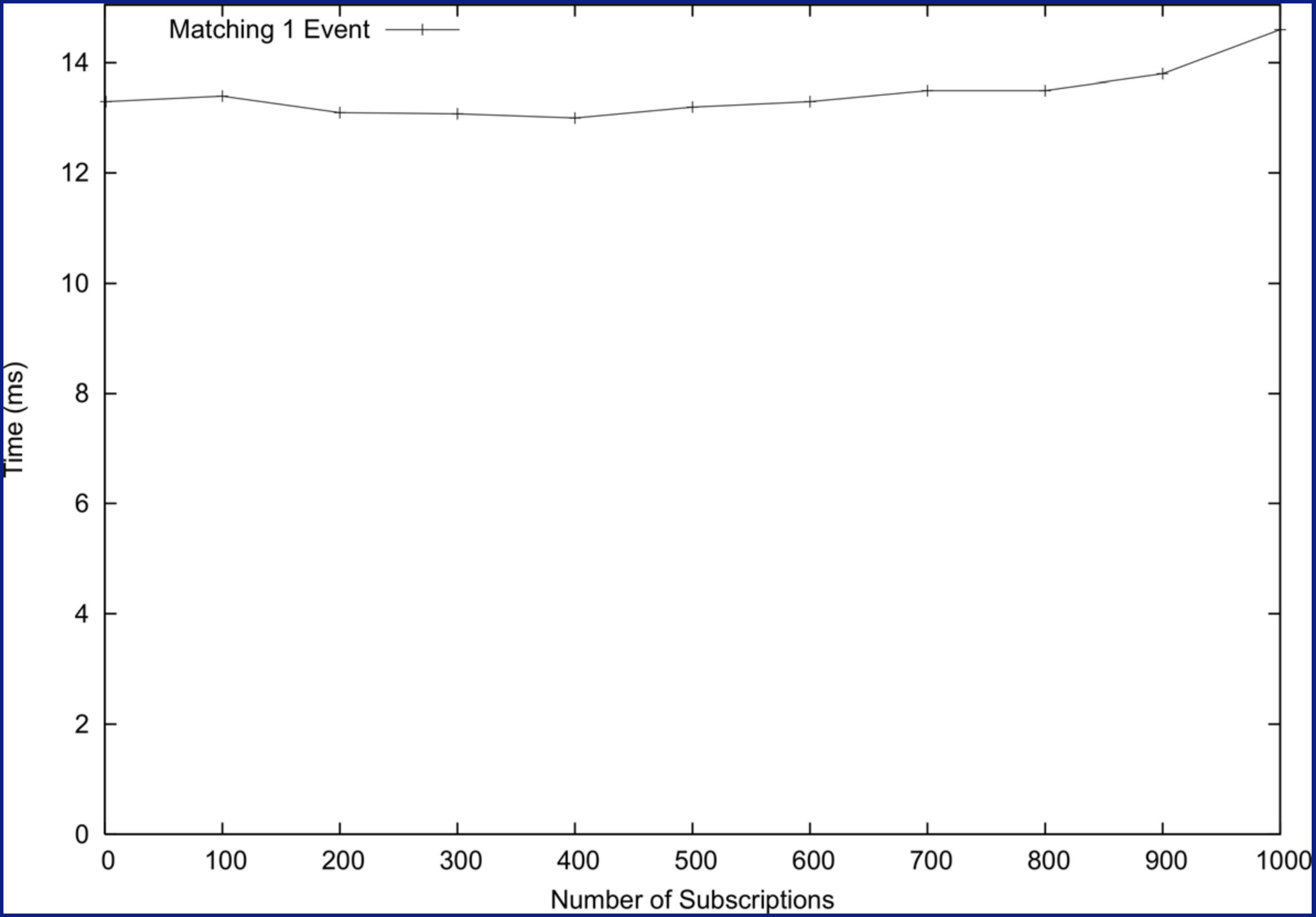
Throuput of BNS’s



Performance of the Policy Service

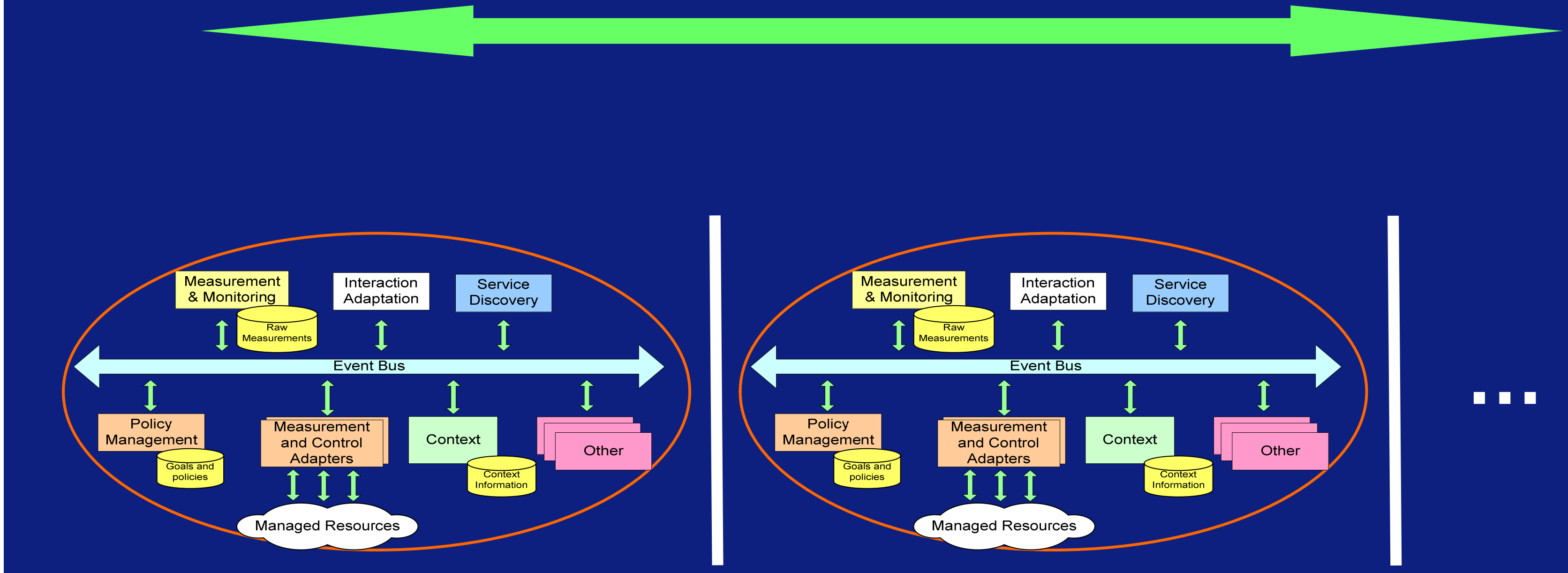
	Task	*Time (ms)*
1	Executing a policy (no condition, empty action)	13.6
2	Executing a policy (no condition, an action to issue a command to BSN via IEEE 802.15.4)	48.2
3	Executing a policy (a condition, an action to publish a new event)	136
4	Executing a policy (a condition, an action to create a managed object upon discovery of a new BSN)	168

Performance of Event Service



Federated SMCs

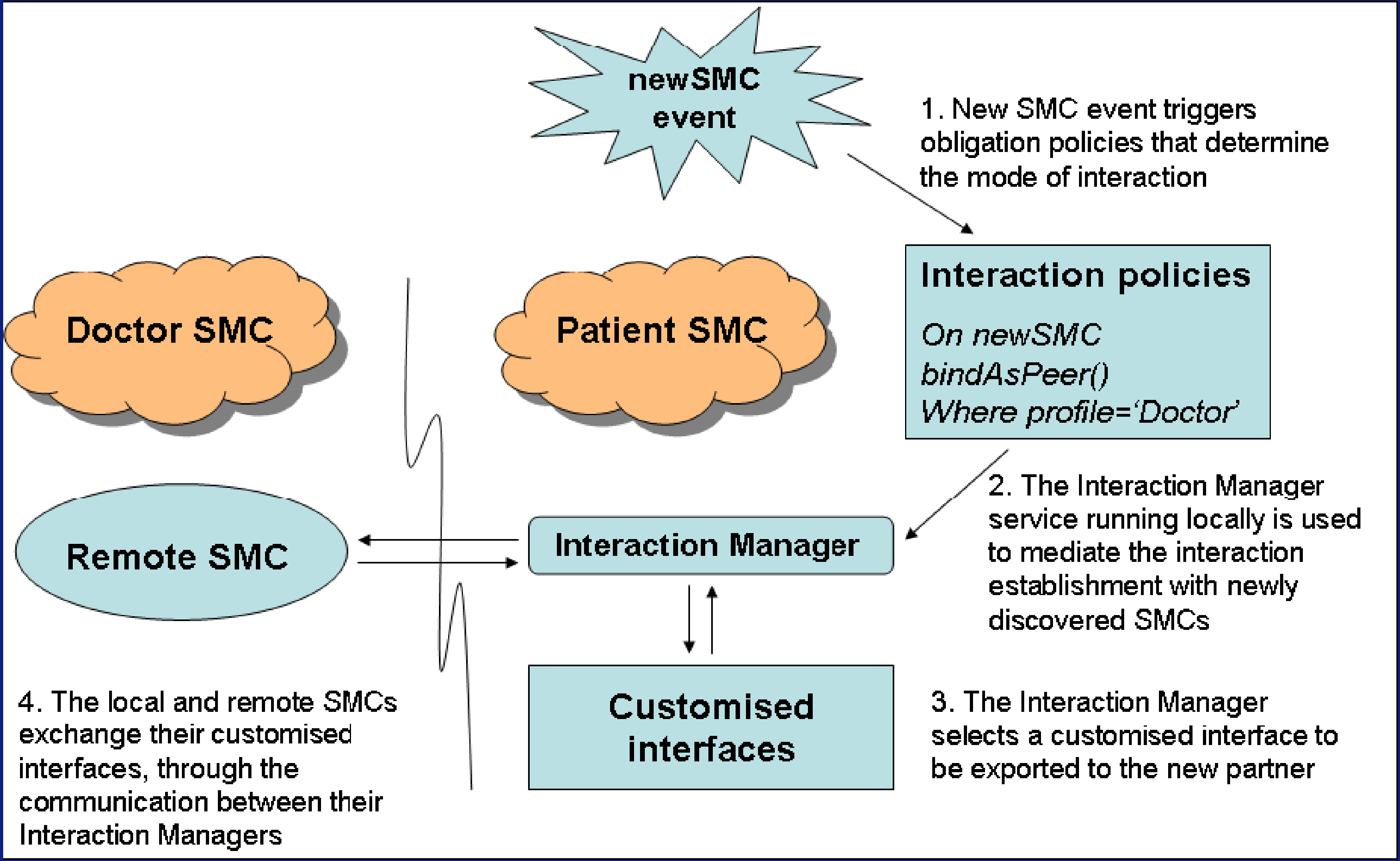
- Peer SMCs (peer devices, peer networks, SLAs…)



Federation Essentials

- Architecture – traditional flat, one-dimensional architecture vs. hierarchical, multi-tiered architecture
- Ontology – federates must possess an agreed vocabulary of common terms and their meanings
- Security and privacy – as the level of integration increases between autonomous managed resources, protecting the security and privacy of these resources also increases; it is critical not to assume that every federate has access to all distributed resources; there may be a natural precedence among federates
- Negotiation – given the potentially ephemeral nature of these federations, negotiation protocols between SMC's to create these federations are essential

SMC Federation Establishment



Interaction policies

On newSMC
bindAsPeer()
Where profile='Doctor'

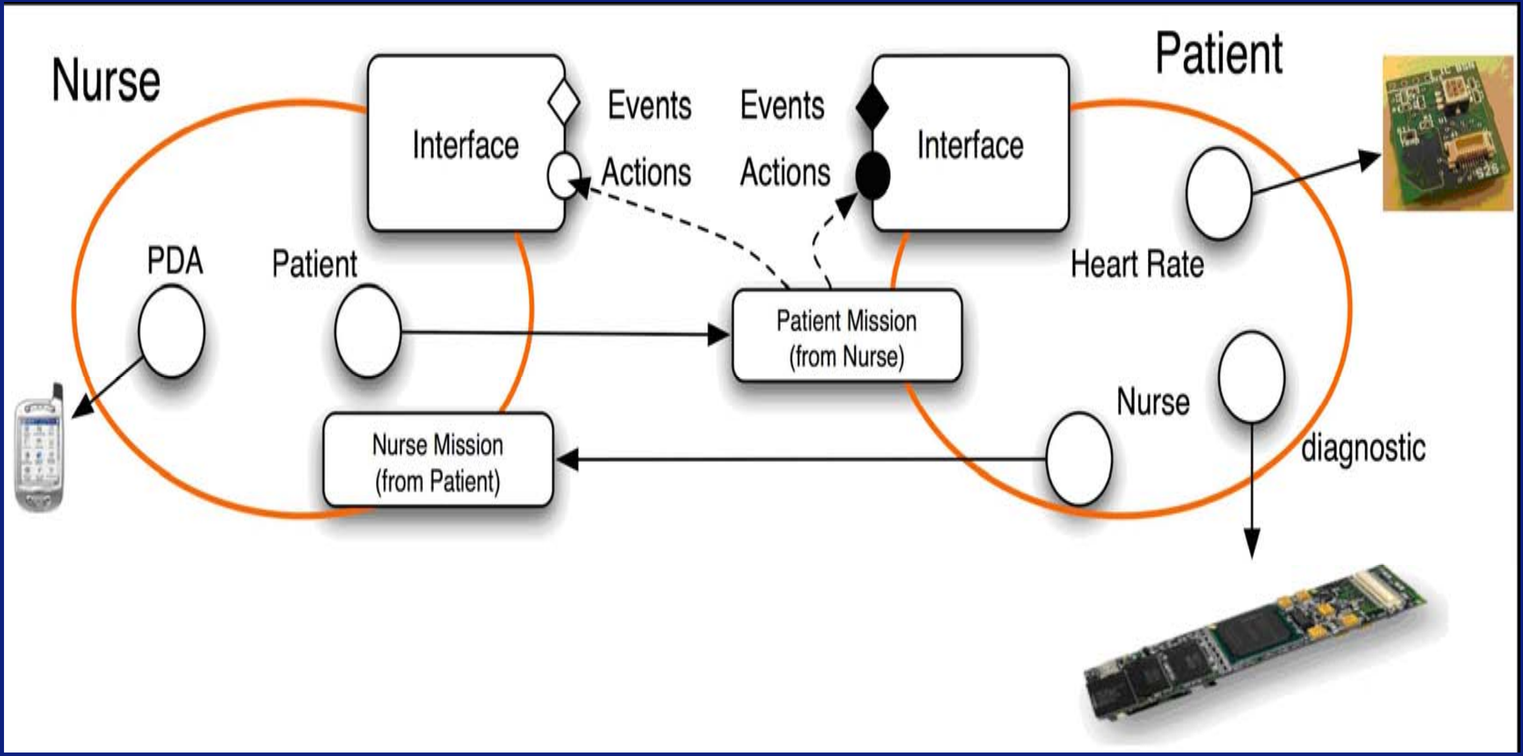
1. New SMC event triggers obligation policies that determine the mode of interaction

2. The Interaction Manager service running locally is used to mediate the interaction establishment with newly discovered SMCs

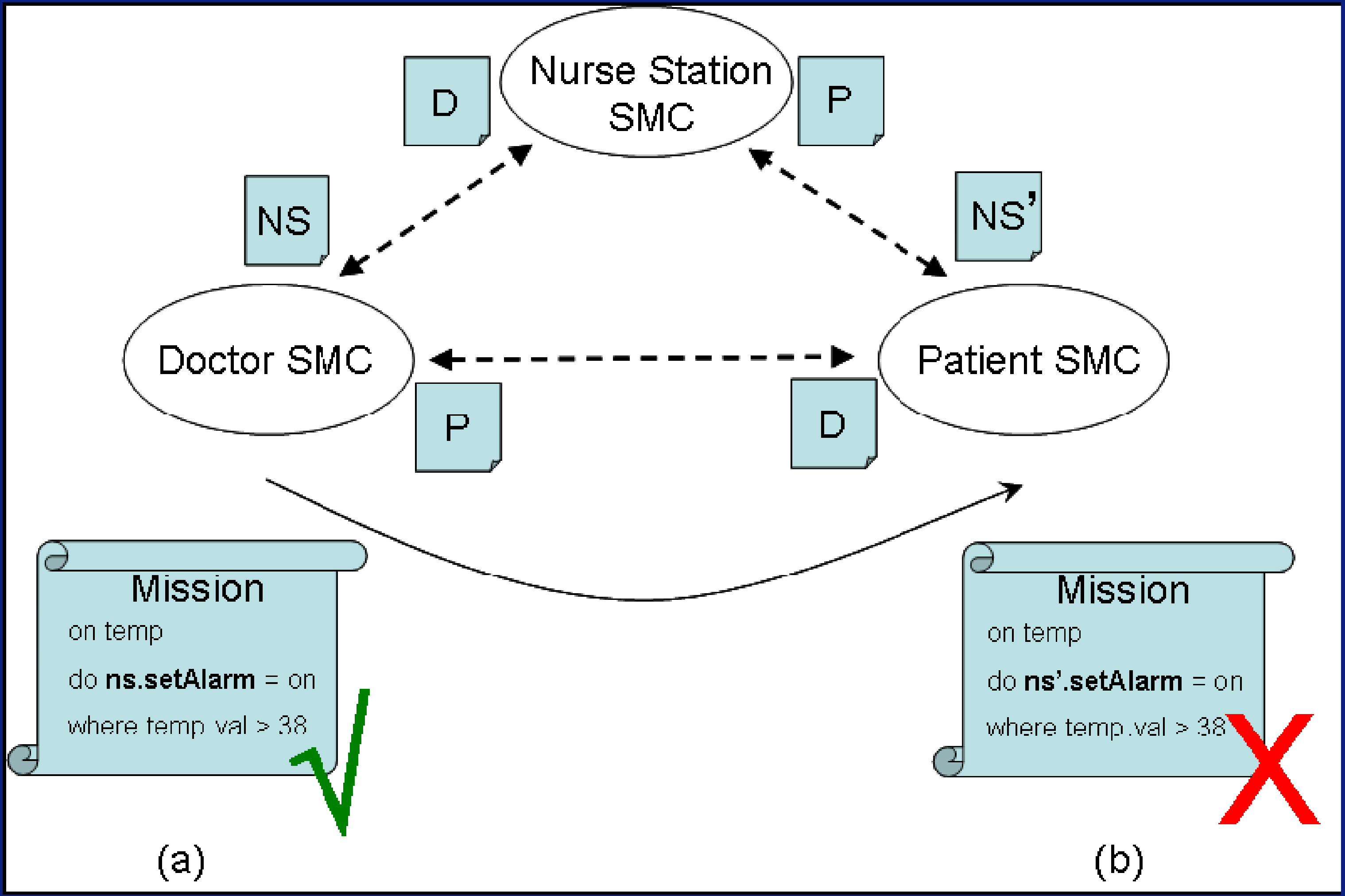
3. The Interaction Manager selects a customised interface to be exported to the new partner

4. The local and remote SMCs exchange their customised interfaces, through the communication between their Interaction Managers

Missions across SMCs



Validation of missions



SMCs in the large

- As we saw earlier, the basic features of an SMC are an event bus, a discovery service, and a policy service
- The implementation environment for the body-area distributed system is particularly simple
- How does one translate these concepts to a wide-area context?
- How does one exploit the ephemeral federation of mobile SMCs with fixed SMCs?

Wide-area SMC implementation

- Body-area content-based event bus modelled after Siena formats and features, without broker network; therefore, event bus implementation in the wide area is simply Siena
- The discovery service is a combination of Service Location Protocol (SLP) and active registration within directory services (e.g. LDAP)
- Ponder2 (the implementation in the body-area system) was designed for use at all levels of scale

Communication of information over ephemeral federations

- The federation mechanism permits two SMCs to interact subject to the defined interaction policies
- The usual reason for federation is to transmit information from mobile, constrained environments for storage and analysis in less-constrained, fixed environments.
- As indicated earlier, different applications will require different interaction styles to meet their needs
- In addition to the urgent delivery of critical events, we have explored more relaxed forms of data transfer using delay tolerant networking techniques

DTN scenario

- Track the use of asthma inhalers across a geographically-distributed population
- Patient is equipped with an inhaler that simply counts the number of uses
- Inhaler contains the battery, storage space, networking capabilities, and processing power required of an SMC.
- Goal is to monitor inhaler usage and gather statistics on seasonal and geographical variation in inhaler usage

Specifics

- Prior to use, an inhaler is loaded with enough patient-specific information to allow for data collection to take place
- Inhaler count data, together with identifying and location information, finds its way to the patients general practitioner’s surgery (doctor’s office)
- Statistical queries can be made of the distributed asthma data within the wireline environment
- The count data for a patient should be uploaded whenever the inhaler SMC is able to federate with other SMCs that are part of the NHS fabric

SMCs involved

- Inhaler (many, mobile)
- Ambulance (fewer, mobile)
- Doctor’s surgery (even fewer, fixed)
- Hospital (small number, fixed)

DTN

- Whenever the inhaler can federate with an instance of one of the other three SMC types, it issues a custody transfer request for a bundle containing its data
- The SMC that accepts the custody transfer associates its current location with the bundle and then forwards it onto the final destination (doctor’s office/surgery)
- If the transfer has been to an ambulance, then it will eventually transfer custody of the now augmented packet to the next hospital with which it can federate
- Once received by a fixed SMC, the data will be transferred to its final destination.

Conclusions

- Prototype implementation has demonstrated that the SMC pattern can be applied to e-Health applications
- Event bus provides sufficient performance, modularity, and scale to adequately address e-Health management traffic
- ECA policy-based management provides a simple and effective strategy for encoding the necessary adaptation strategy for e-Health applications
- The SMC concept can be extended to larger scale environments
- The federation techniques that were developed for mobile SMCs also work when federating with larger scale, fixed environments

Future Work

- Security and trust in body area networks
- Negotiation protocols required for peer SMC's to federate (Glasgow PhD dissertation)
- Augmentation of the basic system with AI inference engines (e.g. support vector machines)

- For more information
  http://www.dcs.gla.ac.uk/amuse/