

Adaptive Context Aware Security

Adapt and Survive?



Dr Naranker Dulay

Department of Computing
Imperial College London

UK-UbiNet (5-7 May 2004)

Brave New World



■ Technology View

- Small computing devices **everywhere** – fridges, washing machines, door locks, cars, furniture, plants, animals, people, etc
- **Battery** powered
- Capable of **wireless** communication

■ User View

- Devices mostly **invisible** – devices interact implicit with each other and environment
- Augment human abilities in performance of tasks



Will need to be “secure” for wide-scale acceptance



SECURITY Wishlist

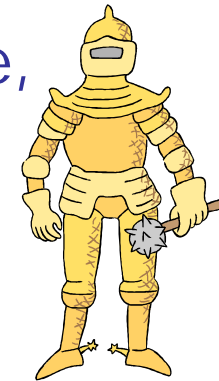


- SECURITY solutions that are proactive, minimally intrusive, easy to use
- Ability to control who/what has access to “my” data (stored, communicated, inferred), ability to define levels of privacy, trust etc
- Devices that recognise/respond to “owners” only

SECURITY Wishlist 2

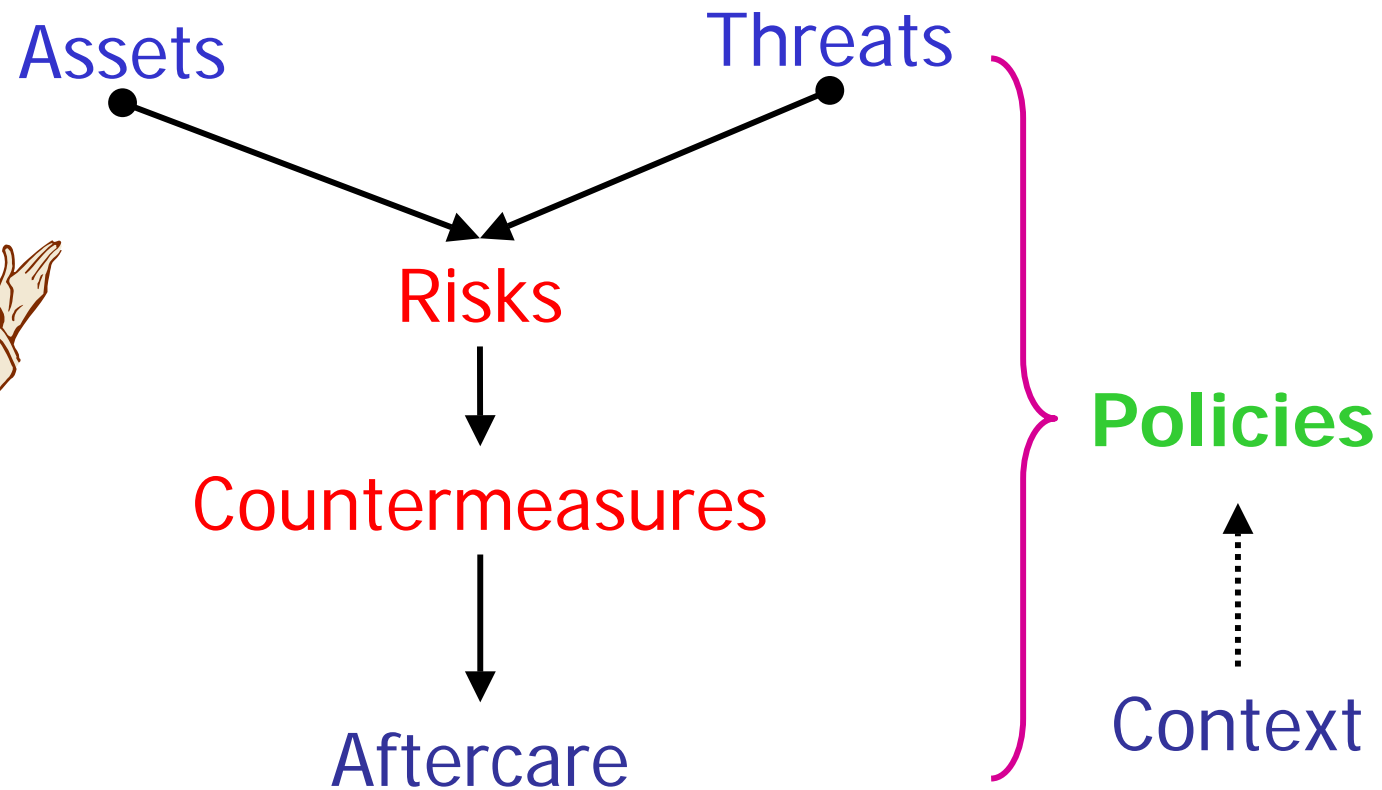


- Means of tracing stolen devices, proving transactions
- Ability to be invisible or anonymous when needed
- Protection from spam, viruses, denial of service, identity theft
- Etc....



SECURITY solutions that are adaptive and context-aware

Security Management

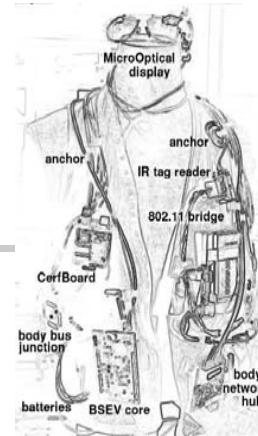


Contextual Security Policies

- SECURITY POLICIES that use CONTEXT
- CHANGES in context, TRIGGER changes in SECURITY
- Ability to UPDATE/EVOLVE security policies



Context



■ Current State

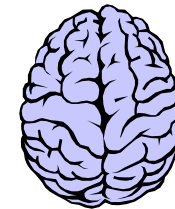
- Current location, time, role, activity, neighbours, physiological state, room temperature, noise, battery-life, free memory, CPU speed, network connectivity/QoS/cost, available services, etc..

■ User Preferences & Relationships

- Preferences, relationships, recommendations from others

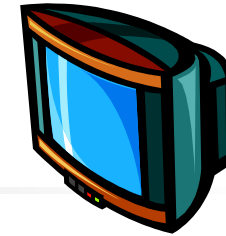
■ History

- Readings and outcomes, accumulated wisdom

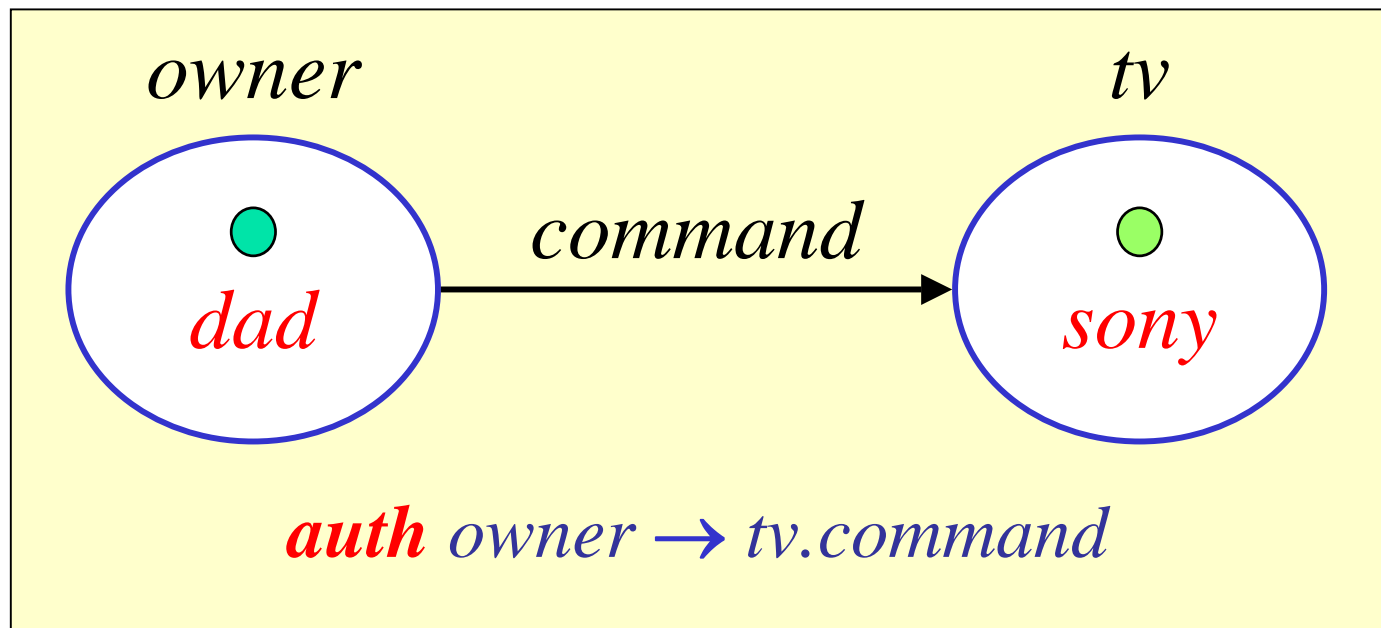


- ☞ Degree of certainty that should be placed in such information?
- ☞ Need for high-level models of context

Example 1

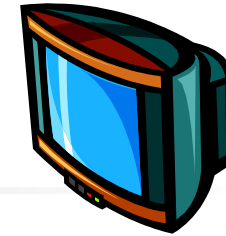


- Only operate for "owner"

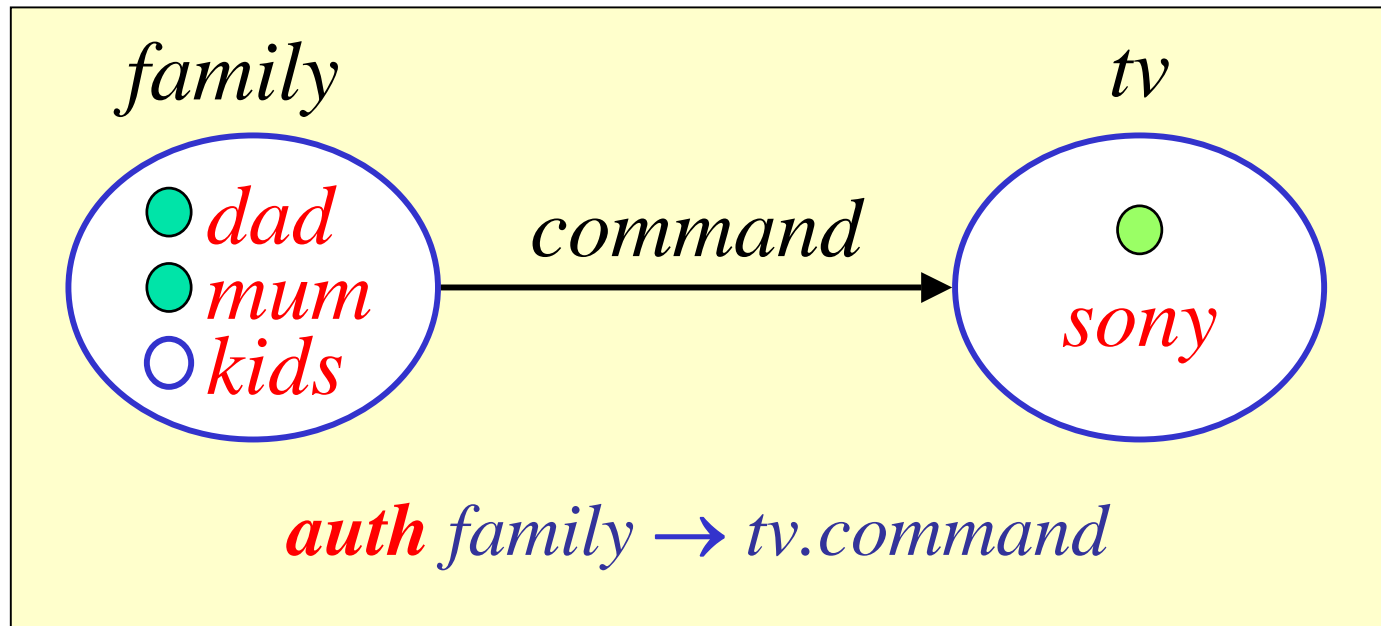


- On purchase, add *dad* to *owner* domain, and *sony* to *tv* domain

Example 2

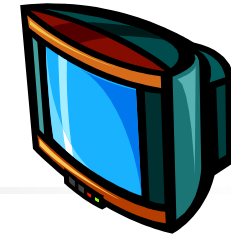


- Allow members of owner's **family** to operate **tv**

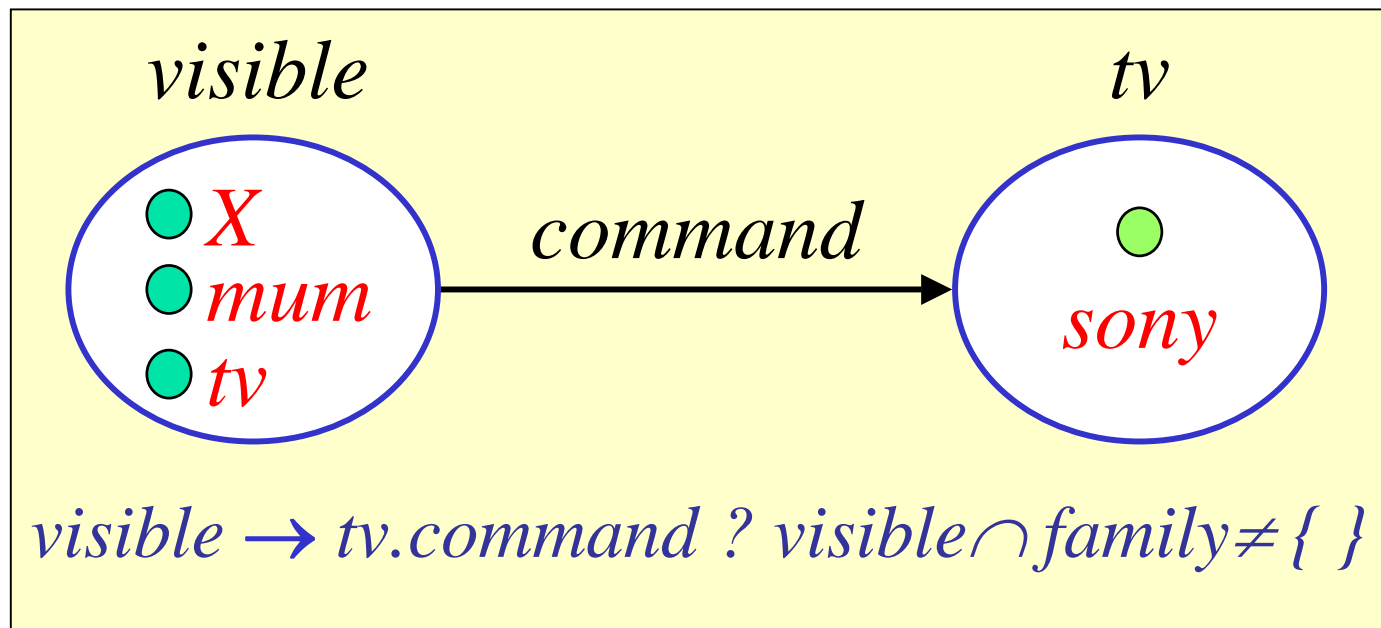


- Domains are organised hierarchically

Example 3

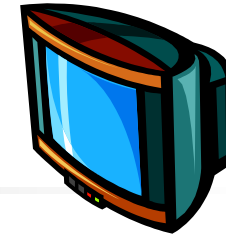


- Allow those that are **co-located** to **family** to operate **tv**



- Member's can be people, devices, processes etc.

Example 4

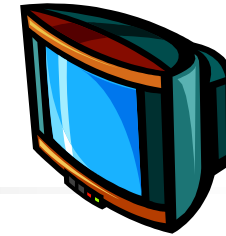


- How is *visible* formed?
- Ideally maintained by a Discovery service
- Could be formulated as ECA (Event-Condition-Action) policies, e.g.:

```
on detect(X) do  
  visible = visible + {X}
```

```
every secs (60) do  
  visible = { X for X in visible if now-X.seen < 30 }
```

Example 5



- Go into SOS mode after 20 denied attempts

```
on 20*deny(tv.command) do  
    disable familyPolicy, coLocatedPolicy  
    enable SOSpolicy
```

- SOSpolicy could periodically broadcast STOLEN messages, or then after awhile perhaps COMPLETELY disable the TV?



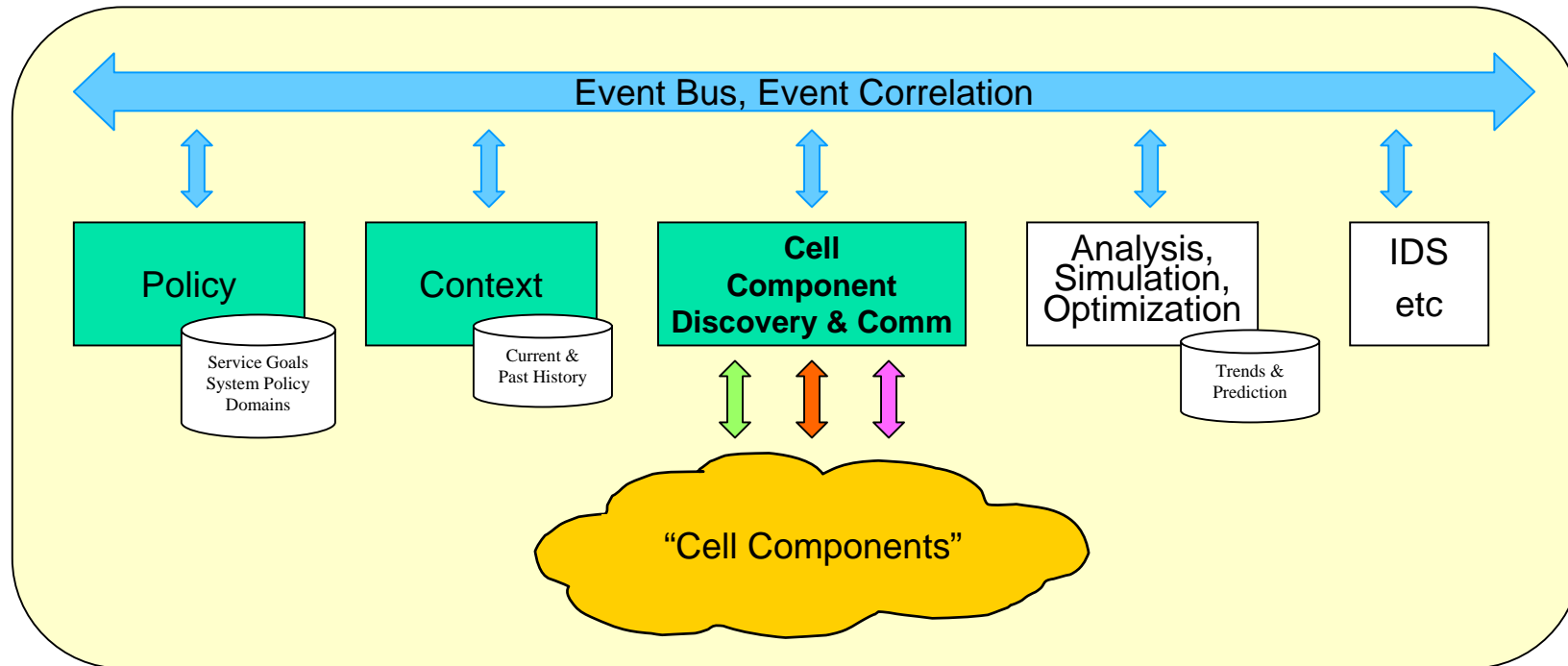
AMUSE Project



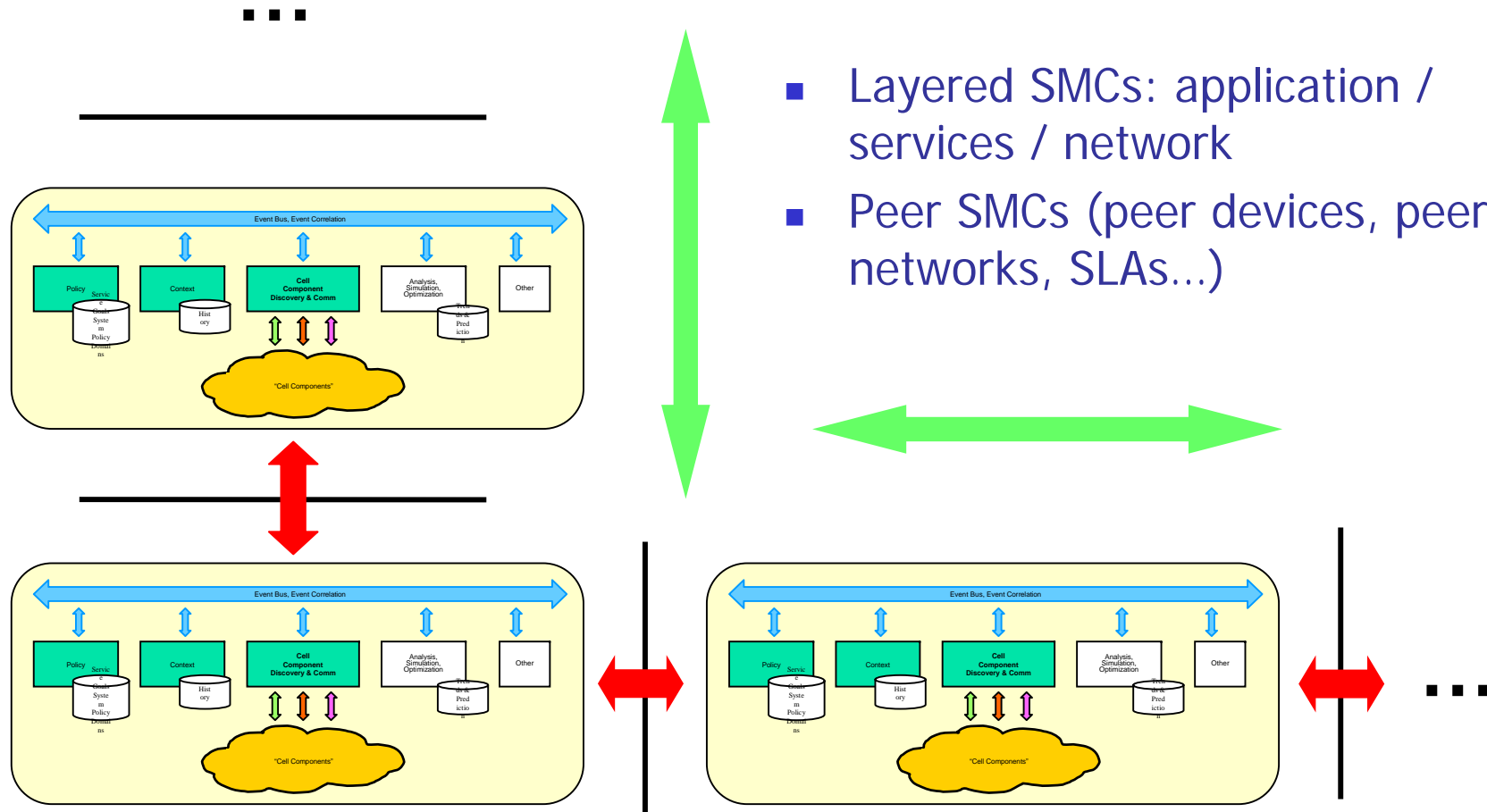
- **Autonomic Management of Ubiquitous Systems for E-health**
- Imperial College & Glasgow University (Joe Sventek)
- EPSRC Funded: 2004 for 36 months
- <http://www.dcs.gla.ac.uk/amuse/>

- ☞ Integrates ideas on **closed-loop** management with policy-based management for distributed systems big and SMALL
- ☞ Key idea => Self-Managed Cell (SMC)

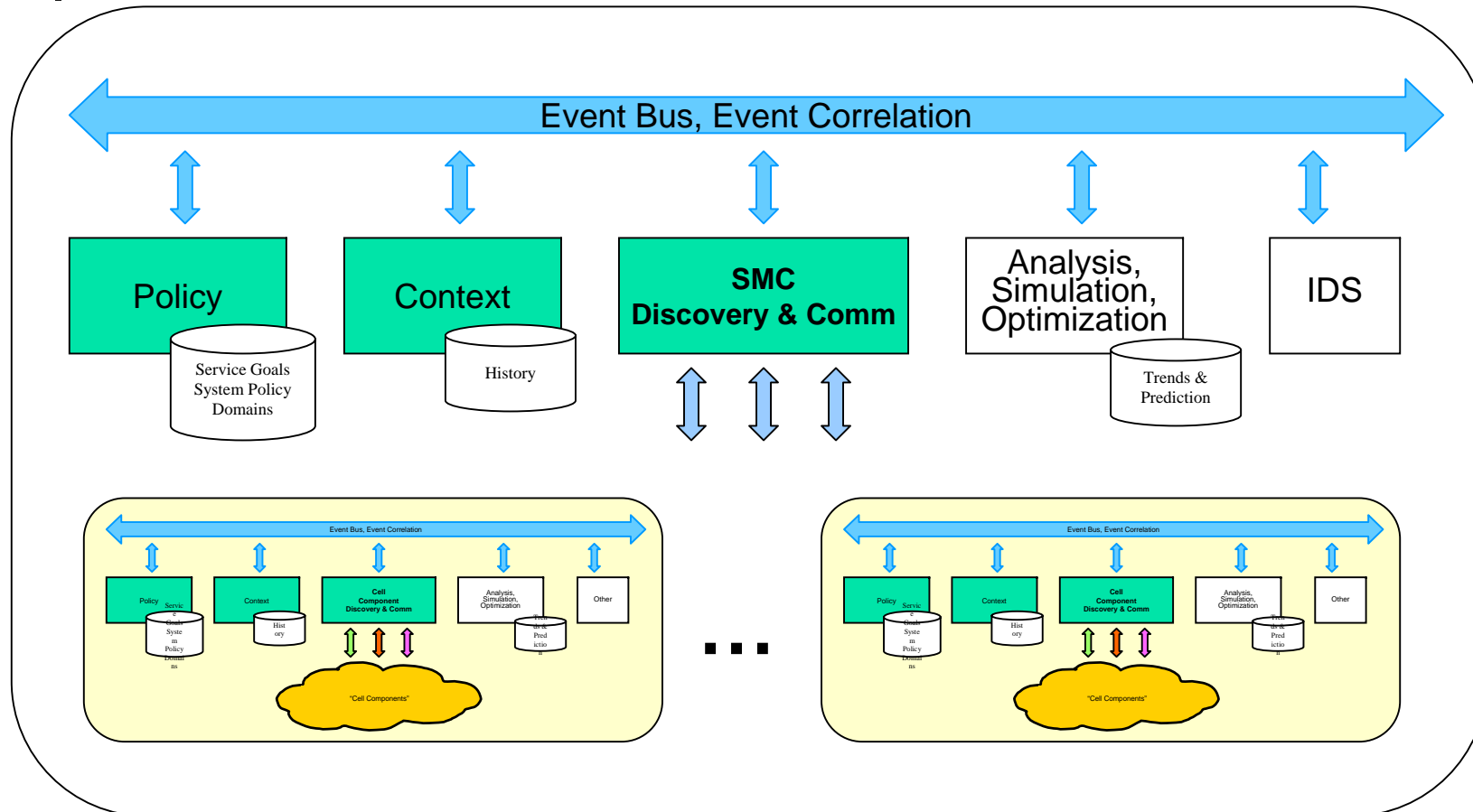
Self-Managed Cell (SMC)



Layered and Federated SMCs



SMC Composition



- Enclosing SMC "programs" the nested SMCs



Summary



- **Policy-driven security management**
 - Policies for Authorisation, Delegation, Event-Condition-Actions. Policies applied over Domains (Sets) of objects
- **Contextual policies:**
 - Context conditions, context events, context history
- **Policies are 1st class:**
 - Policies on policies, e.g. can enable, disable, replace policies
- **Need Models of:**
 - Context
 - Privacy and Trust
- **The Future:**
 - Systems that Self-Manage, Policies that “emerge”

The Future:

