LANCASTER UNIVERSITY

From Mobility to Ubiquity and Beyond: Challenges to Middleware

Gordon S. Blair

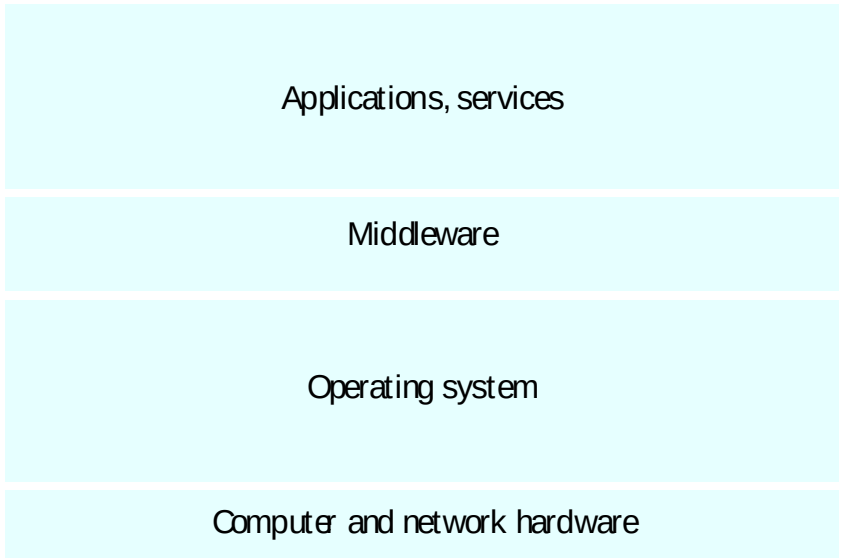
InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

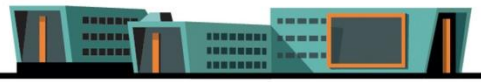
InfoLab21
Computing Department

Context for this talk: design of middleware

Middleware:

- Offering a suitable programming abstraction
- Masking out complexity, including heterogeneity
- Offering interoperability, portability and E2E QoS





Middleware: a definition?

Middleware:
The platform that is left behind when the train departs the station at high speed for its destination.

A Few Words from Danny Cohen

In the beginning ARPA created ARPANET.

And the ARPANET was without form and void.

And darkness was upon the deep.

And the spirit of ARPA moved upon the face of the network and ARPA said, 'Let there be a protocol,' and there was a protocol. And ARPA saw that it was good.

And ARPA said, 'Let there be more protocols,' and it was so. And ARPA saw that it was good.

And ARPA said, 'Let there be more networks,' and it was so.

Gordon's distributed systems version

In the beginning there was small scale experimentation.

And the experiments were without abstraction or openness.

And darkness was upon the deep.

And the spirit of the OMG moved upon the face of the distributed system and said, 'Let there be a middleware standard,' and there was a standard. And OMG saw that it was good.

And Microsoft said, 'Let there be more standards,' and it was so. And Microsoft saw that it was good.

And the community said, 'Let there be more networks and of course also mobility, ubiquity and cloud computing for good measure,' and it was so…..

…. but is it good?

Early distributed systems

- Limited in scale and heterogeneity, issues such as openness, and support for QoS not a big issue

Internet-scale distributed systems

- Large scale and significant levels of heterogeneity (platforms, languages and middleware), significant advances in supporting openness and QoS

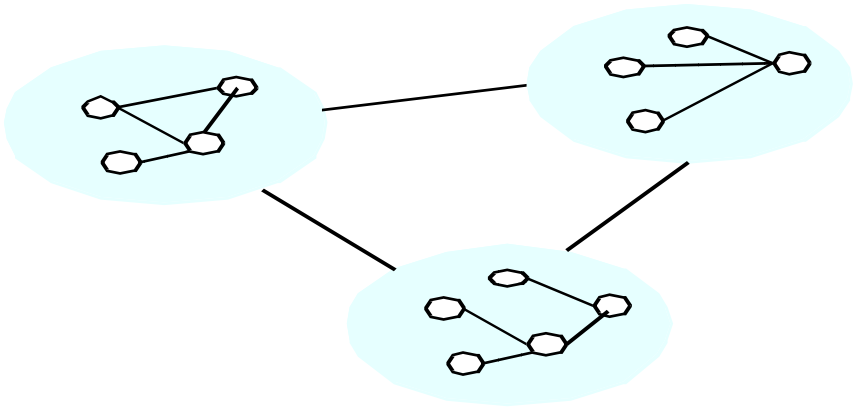
The complex distributed systems of tomorrow

- Significant increases in scale and also heterogeneity in *all* its dimensions; major research questions concerning openness and QoS

From distributed systems to systems of systems

Three key dimensions:

- Mobility
- Ubiquity
- Utility (cf. cloud computing)



See also Ultra-Large-Scale Systems (http://www.sei.cmu.edu/uls/)

The mobility dimension

Key developments include:

- Supporting variable connection and disconnected operation
- Support for spontaneous interoperation
- Context-awareness and adaptation
- Tailored communication paradigms
  - Queued RPC, tuple spaces, publish-subscribe, etc
- Tailored security and privacy models
- Etc.

The ubiquity dimension

Key developments include:

- Low footprint operating systems for sensor devices
- Power awareness
- Emphasis on suitable overlay structures
  - WSNs, MANETs, VANETs, DTNs, etc
- Tailored communication paradigms
  - Tuple spaces, publish-subscribe, etc
- Tailored security and privacy models
- Etc.

The utility dimension

Key developments include:

- Emphasis on everything as a service (infrastructure, platform, software)
- Techniques for large scale search, storage, distributed consensus and processing
- Supporting developments on virtualization and associated infrastructure, e.g. see Xeno Servers
- Proprietary infrastructure for cloud computing
  - Google's AppEngine, Amazon's EC2, Microsoft's Azure, Yahoo's HADOOP, etc.

A statement of challenges

What are the theories and systems principles that underpin distributed systems of systems, in particular:

- How do we achieve basic systems properties including interoperability in such systems?
- How do we ensure that end-to-end quality of service properties can be achieved in such systems?
- How do we manage such complex systems-of-systems?

Avoiding the issues……

In this paper, we have presented a really super new approach to programming in mobile systems ….. In future work, we will consider how to extend our approach to interoperate with the fixed Internet and to achieve key end to end properties across such composite systems ….

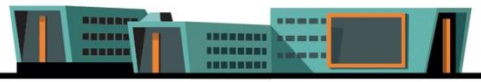
Illustrating the challenge: Towards environmental observatories

Motivation

- Many well-known challenges to the environment
  - Global warming, pollution, diminishing of natural resources, threats to bio-diversity, etc.
- The environment decade [Al Gore, Earth in the Balance]
  - Book also focuses on the potential role of technology in addressing these problems

What is environmental observatories (the Lancaster perspective)?

- Investigating the role of *contemporary computing technologies*, particularly where pervasive technologies meet distributed systems in supporting the (real-time) *monitoring and management* of the natural environment



Example areas of application

Sustainable water management

- Too much/ too little
- Water quality
- From source to point of delivery

Sustainable agriculture

- Precision agriculture
- Traceability in food production

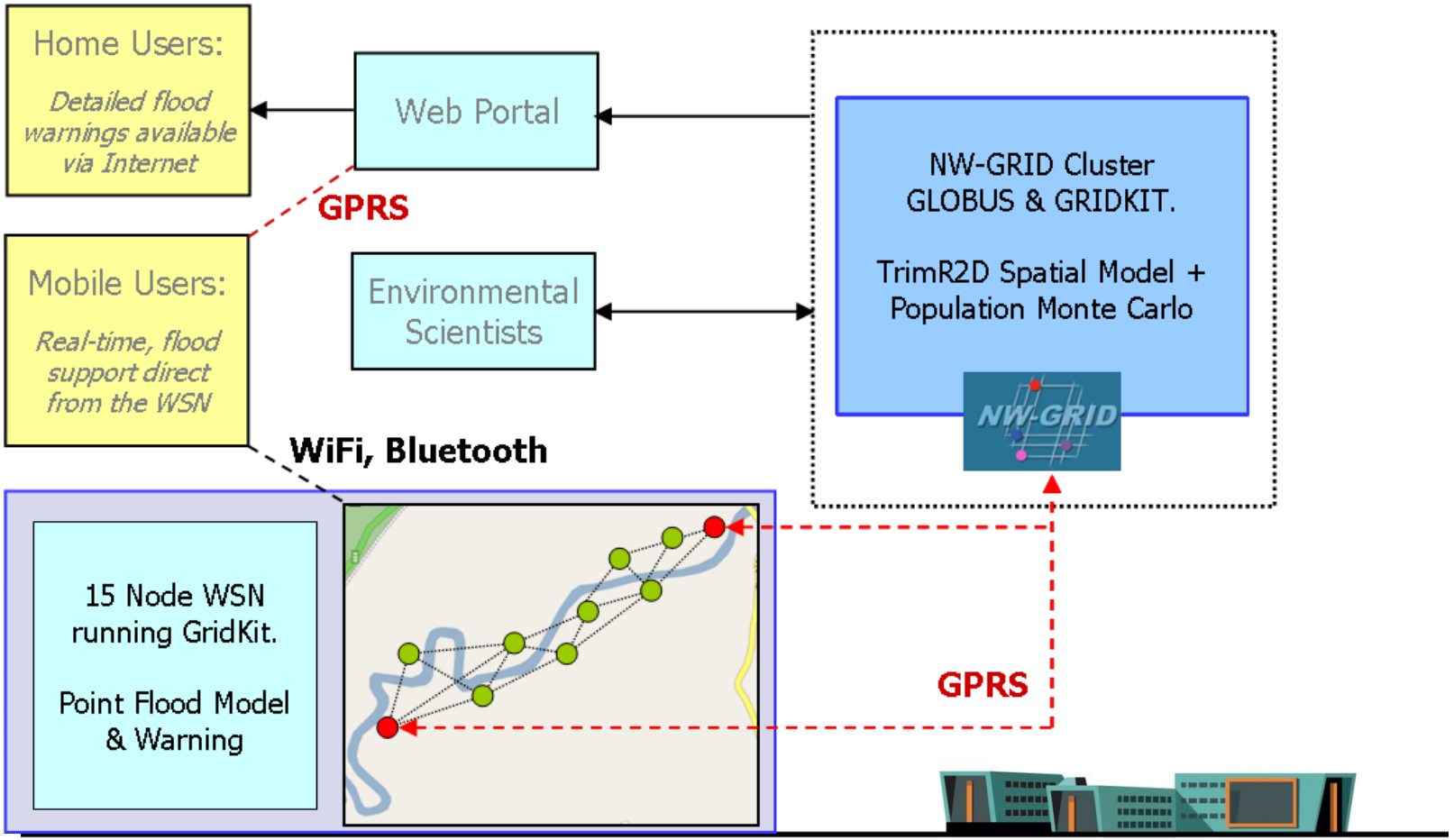
Sustainable energy

- Monitoring and intelligent management of energy use

Sustainable chemical management

- Safety management

An experiment in environmental observation: flood prediction



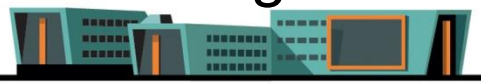
The solution space: a personal view

The need for a strong *architectural representation* of system structures, preferably applied consistently throughout the system architecture

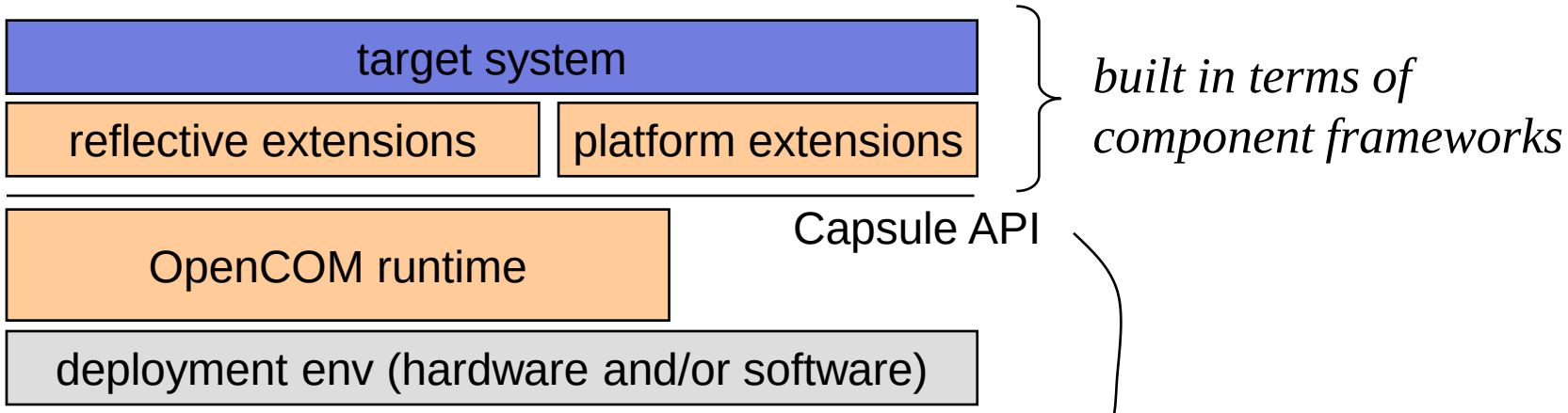
The need for associated *meta-representation* and meta-information to aid reasoning about such system structures

The need to step from syntactic structures to semantic structures (cf. *semantic middleware*)

The need to *reify the goals and intent* of the system and to (dynamically) provide structures to realise these goals



Realising the first two steps through OpenCom

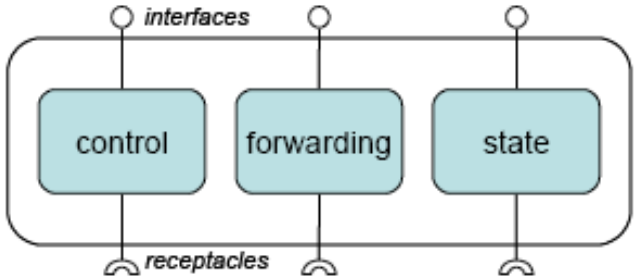



```
template     load(comp_type name);
comp_inst    instantiate(template t);
status       unload(template t);
status       destroy(comp_inst comp);
comp_inst    bind(interface i, receptacle r);
status       putprop(entity e, key k, opaque value);
opaque       getprop(entity e, key k);
```

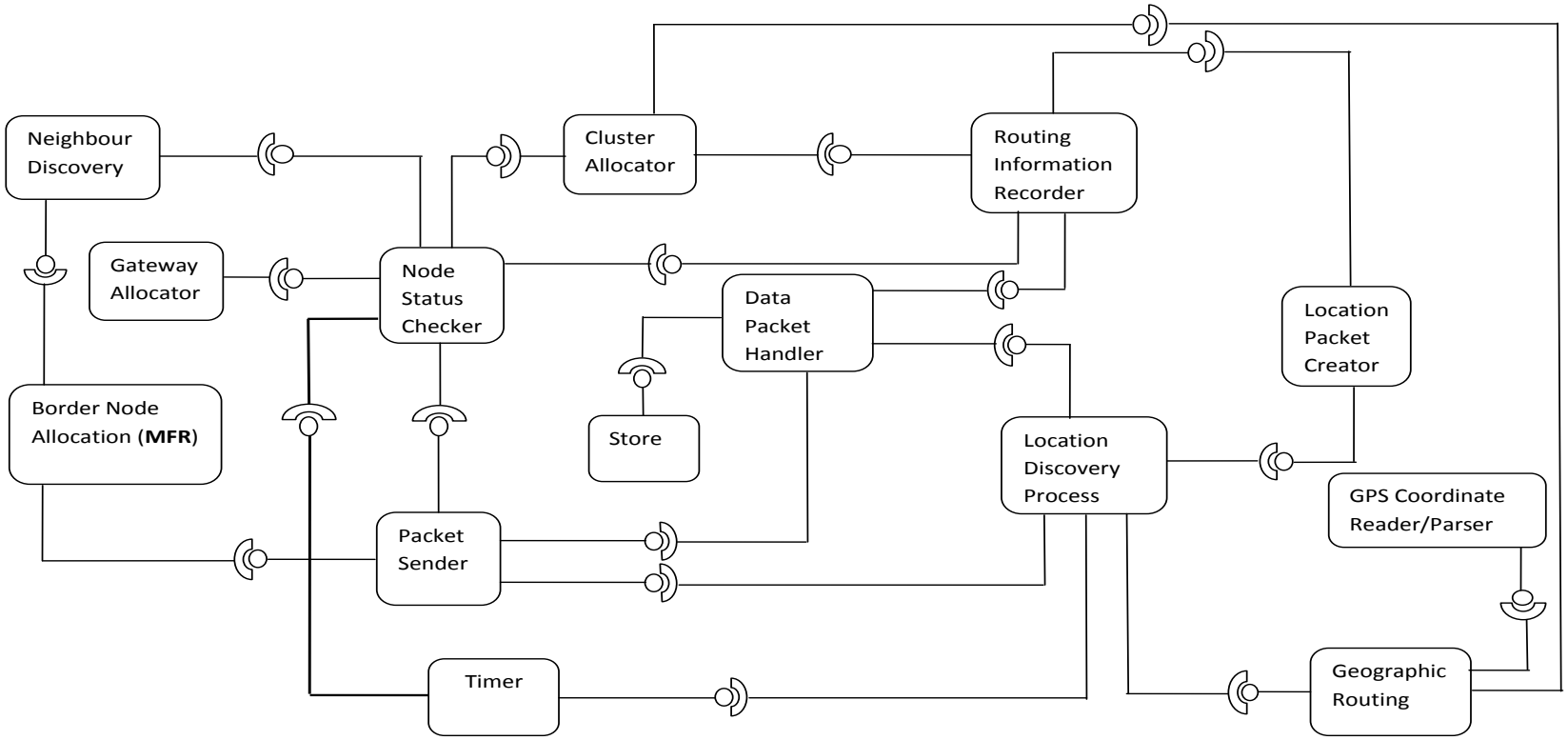
Preferably applied consistently ….

Reclaiming the network (the open overlays approach)

- Important trend towards network virtualisation
- In open overlays, such virtualizations represented by (open) component frameworks
- Overlays can be layered on top of each other and can co-exist
- Key part of the experimental GridKit platform:
  - http://sourceforge.net/projects/gridkit



An example: VANETs



The LORA_CBF component architecture

The next parts are more difficult!

A few pointers

- Sonia Ben Mokhtar , in her PhD thesis, proposes interesting ideas in the area of semantic middleware:
  - http://sonia.bm.googlepages.com/Thesis.pdf
- Also worth looking at
  - http://www.dvs1.informatik.tu-darmstadt.de/publications/pdf/Concept-based04.pdf
- The models@run.time community are investigating the potential role of models to direct the dynamic adaptation of systems
  - See for example the forthcoming special issue of IEEE Computer
  - See also Finkelstein's requirements reflection
    - http://www.cs.ucl.ac.uk/staff/a.finkelstein/talks/reqtsreflection.pdf

Introducing the Connect Project: Towards emergent middleware

LANCASTER UNIVERSITY

Resolve interoperability barriers through on the fly CONNECTor synthesis.

Elicit supporting formal foundation for CONNECTors

Synthesize CONNECTors that are dependable, unobtrusive, and evolvable

connect-forever.eu

SEVENTH FRAMEWORK PROGRAMME

The Connect consortium

Summary

Distributed systems are becoming increasingly complex, driven by trends such as mobility, ubiquity and utility

Most research in distributed systems focuses on one specific area within this space, e.g. middleware for mobile computing

There is a pressing need to re-consider the fundamentals of distributed systems:

- seeking theories and systems techniques to achieve key properties such as interoperability and end-to-end QoS in systems of systems

I hope I can stimulate some of you to join me in this quest

Acknowledgements

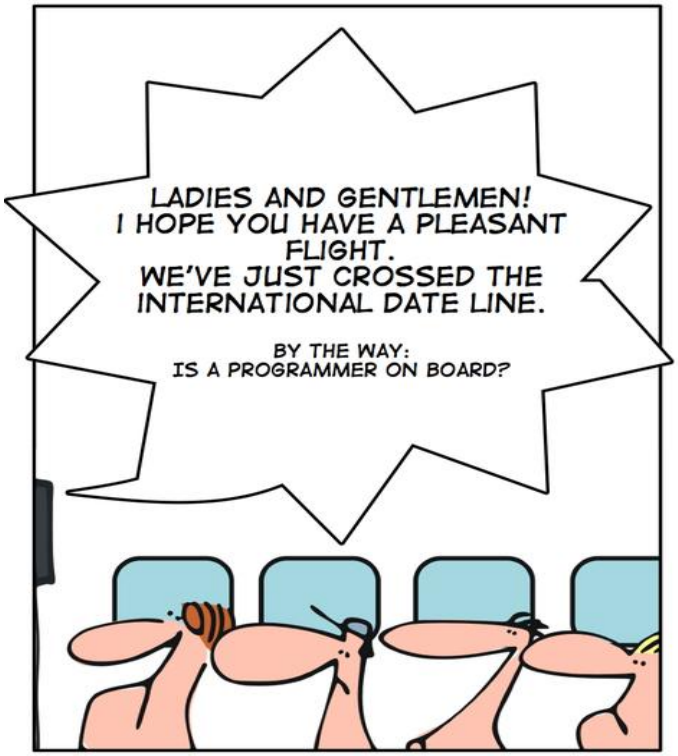
To the *middleware team* at Lancaster:

- Nelly Bencomo, Carlos Flores Cortes, Geoff Coulson, Diego Garcia, Paul Grace, Danny Hughes, Sirak Kaewjamnong, Shen Lin, Lei Liu, Vatsala Nundloll, Barry Porter, Rajiv Ramdhany, Bholanath Surajbali, Francois Taiani, Nirmal Weerasinghe

My many colleagues in the *Lancaster Environment Centre* (LEC) who have provided me with rather stimulating distributed systems problems over the last few years

And also all the collaborators in *CONNECT*, and my colleagues and friends at the *University of Tromsø* who are also working on related issues.

Any questions?



HOW DO YOU KNOW WHEN IT'S TIME TO GET WORRIED