

Where is the state in Web Services?

Caltech – Pasadena – 12-15 July 2004

Savas Parastatidis
School of Computing Science
University of Newcastle upon Tyne

savas@parastatidis.name
<http://savas.parastatidis.name>

- The Grid, Service-Orientation, and Web Services
- Choosing WS specifications
- Our approach
- Conclusions & future plans

- “What is the Grid?”
 - *“Neo, the Grid is everything you would like it to be”*
- IBM: on-demand computing
- HP: utility computing
- Microsoft: seamless computing
- ORACLE: 10g
- Sun: Sun Grid Engine
- Intel: Seti@home or whatever makes money
- HPC community: Interconnected supercomputers
- ...

- The Newcastle team: Internet-scale distributed computing (using Web Services as the infrastructure)



Grid

- Build applications that span organisations
- Create virtual organisations
- Seamless integration
- Hide (virtualise) or share use of resources, network, infrastructure

Web Services

- Glue for heterogeneous platforms/applications/systems
- Cross- and intra-organisation integration
- Standards-based distributed computing
- Interoperability
- Composability

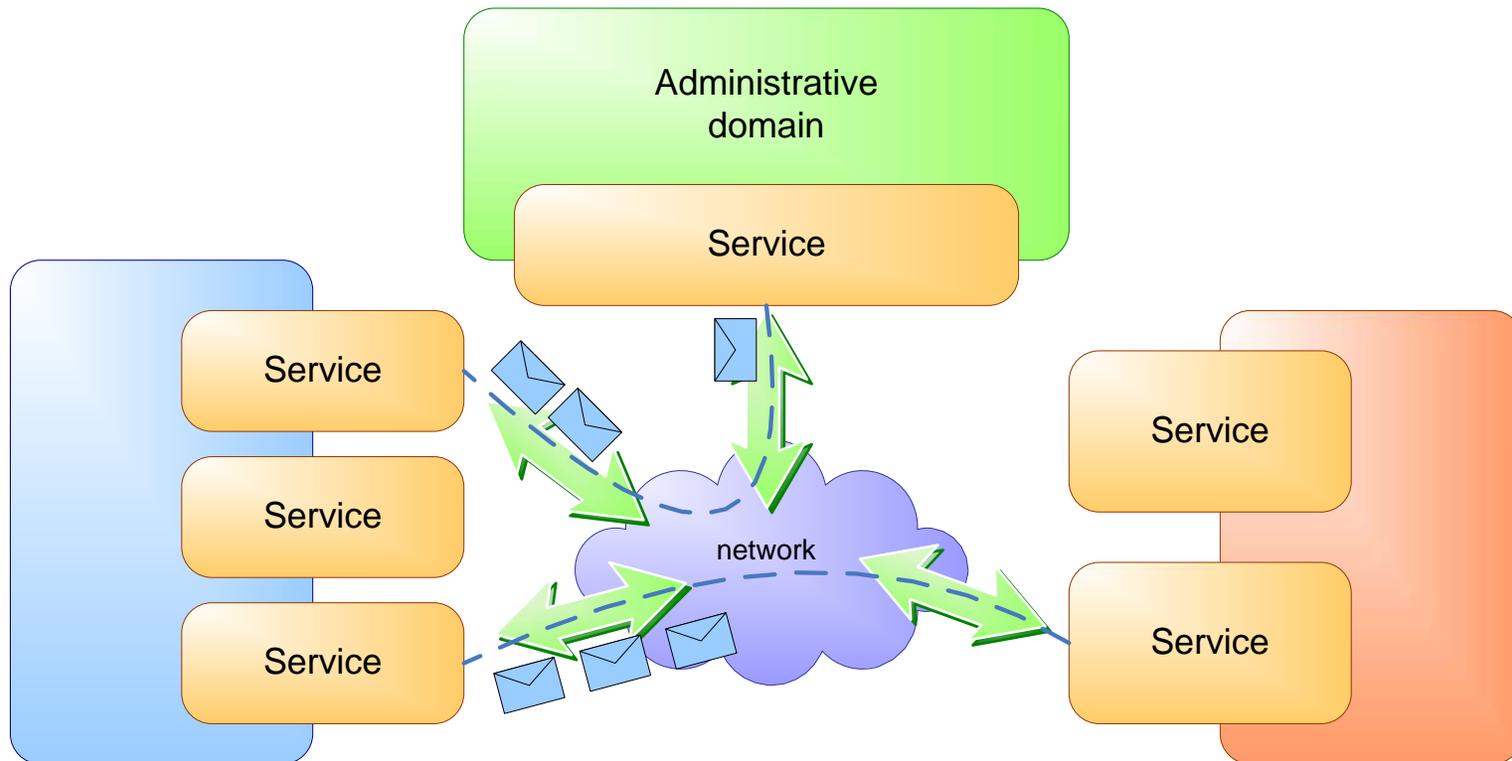
- ...Based on the concepts of Service Orientation



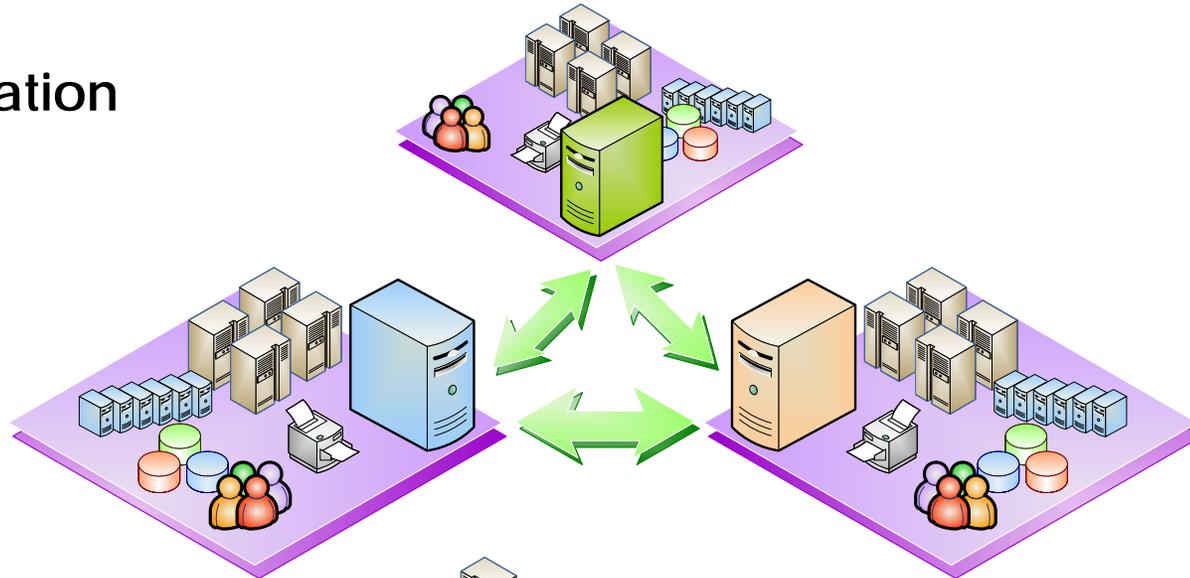
- Built around the concepts of service and message
- *A service is the logical manifestation of some physical/logical resources (like databases, programs, devices, humans, etc.) and/or some application logic that is exposed to the network and*
- *Service interaction is facilitated by exchanging messages*
- A service adheres to a contract
 - Describes the format of the messages exchanged
 - Defines the message exchange patterns in which a service is prepared to participate
- Services are governed by policy
 - Declaratively describe service interaction requirements, quality of service, security, etc



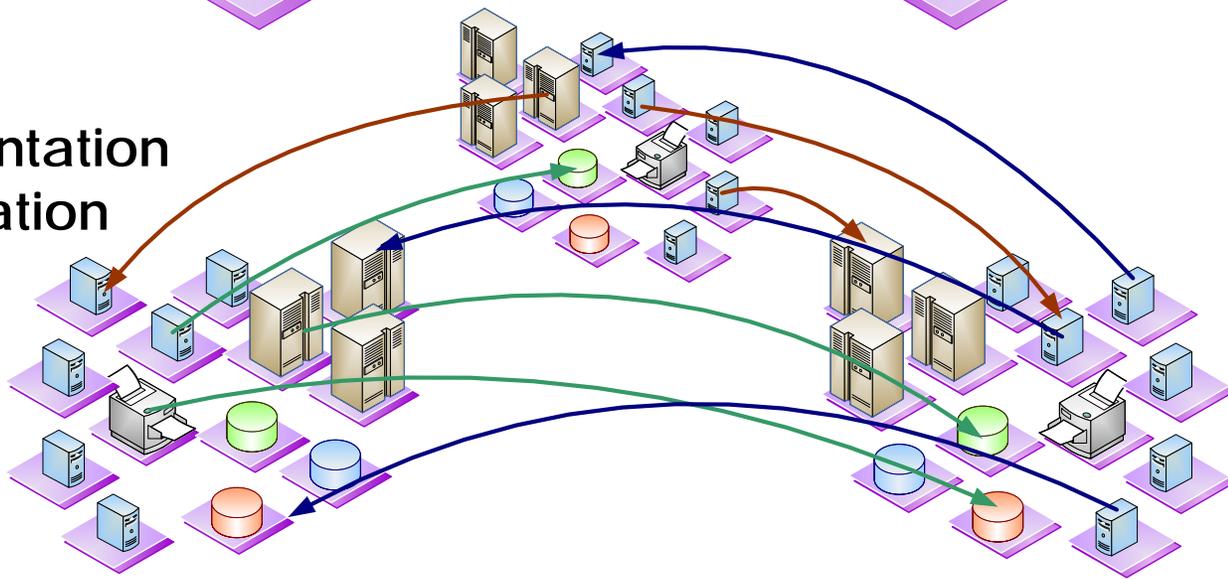
- Service-orientation (and Web Services) helps architects achieve the following properties (but do not guarantee them)
 - Scalability, encapsulation, maintenance, re-use, composability, loose coupling, etc.



Service-orientation



Resource-orientation Object-orientation



- Is it a good idea to base grid application design on the idea of exposing the internal service “resources” to clients?
 - traditionally discouraged
- Breaks encapsulation?
- Can lead to brittle applications?
- Is the CORBA experience applicable?
 - “fine for LANs, but not for Internet-scale applications”?
- Is it better for system management within an enterprise, rather than at internet-scale grid applications?
- The unusual, resource-based conceptual model gives rise to issues of composability with existing Web Services specifications
 - e.g., issues with WSRF and current BPEL (workflow) specification



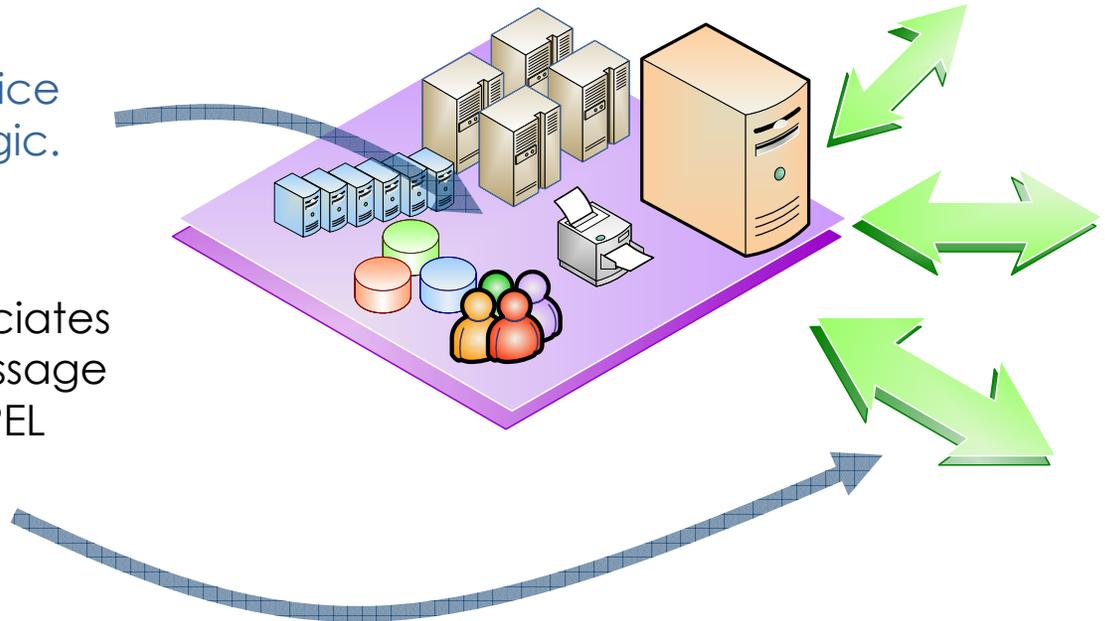
Two types of state

Service internal state

Should not be our concern. Service state is managed by business logic.

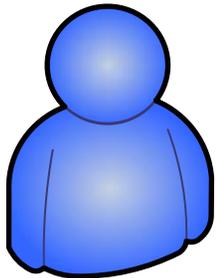
Interaction state

State that the service logic associates with a particular interaction/message correlation (e.g., WS-Context, BPEL properties, service-specific information, etc.)

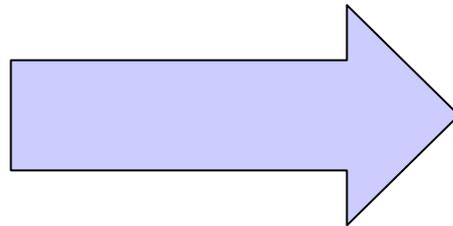


- Stateless interaction

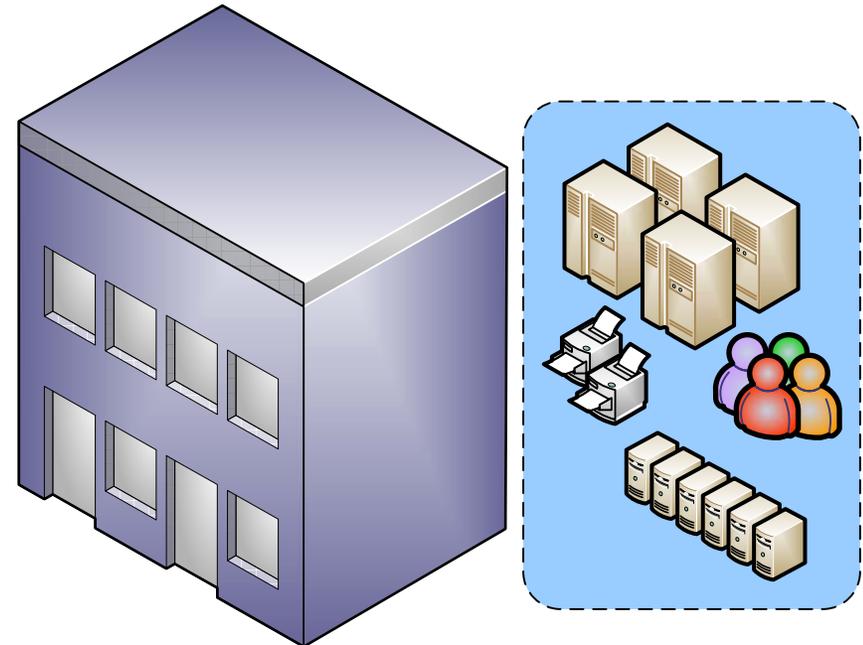
Consumer



Hey, I am savas

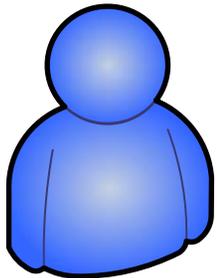


Bank Service

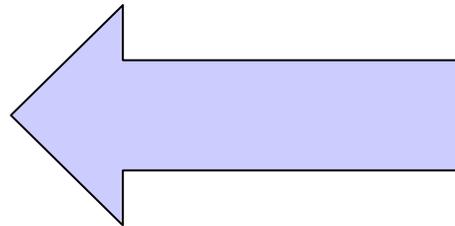


- Stateless interaction

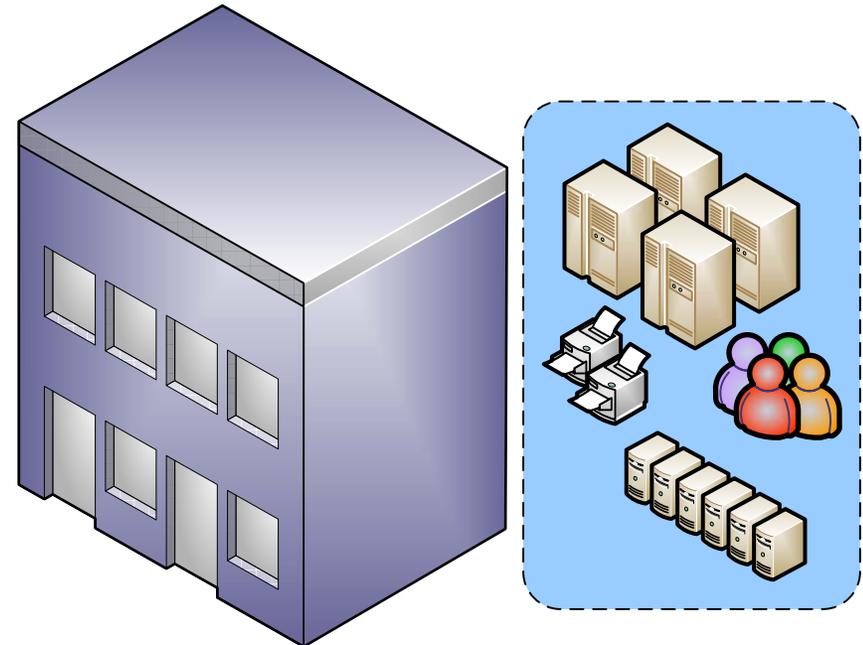
Consumer



Welcome Savas

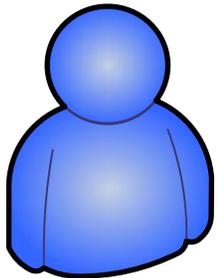


Bank Service

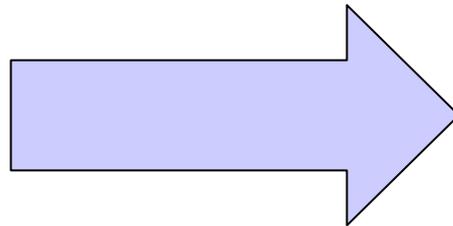


- Stateless interaction

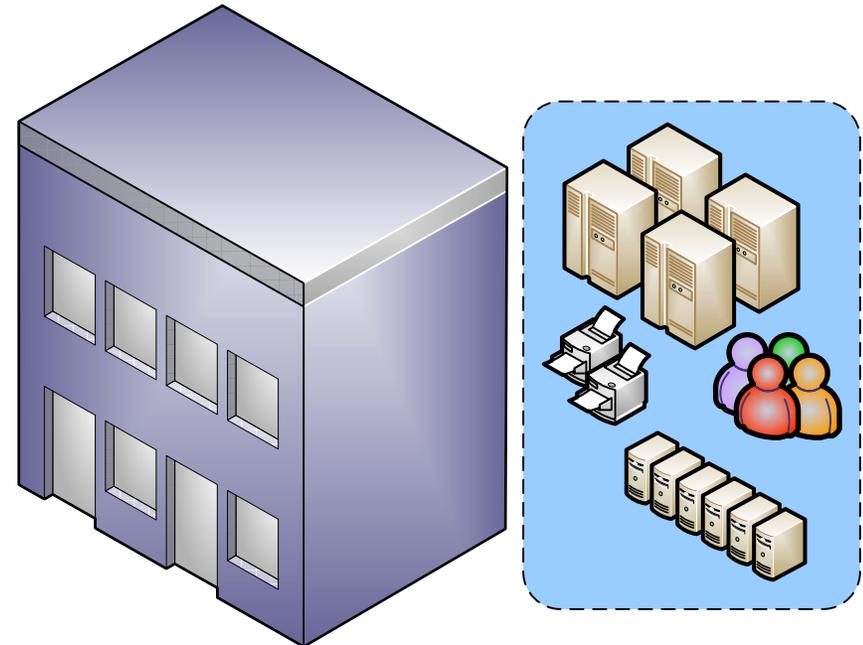
Consumer



I would like £100, please

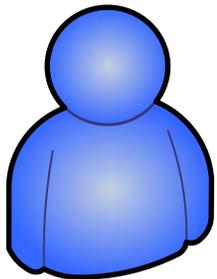


Bank Service

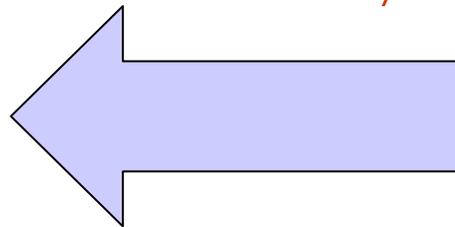


- Stateless interaction

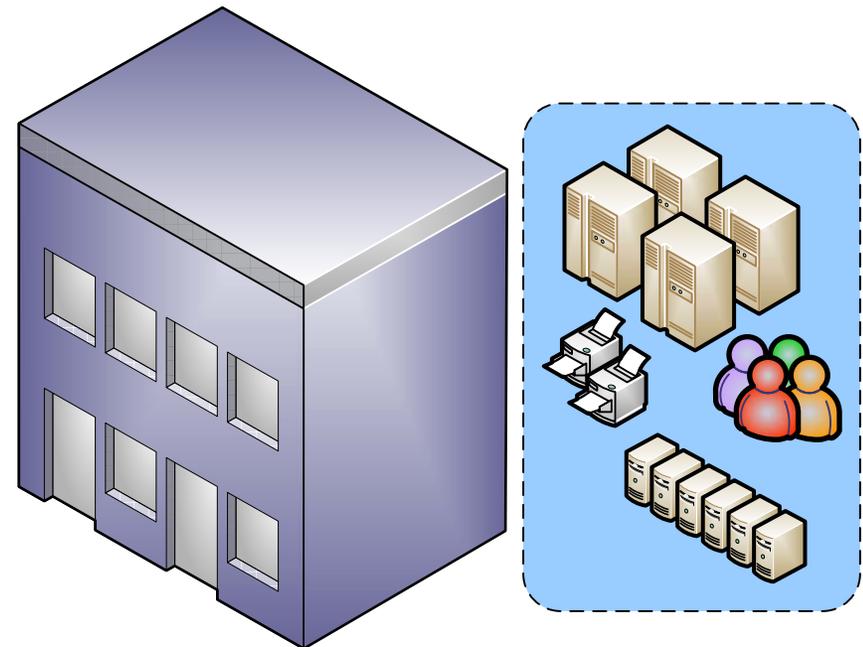
Consumer



No can't do
Don't know you

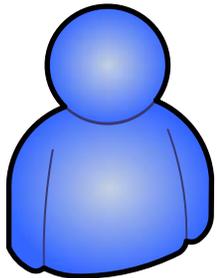


Bank Service

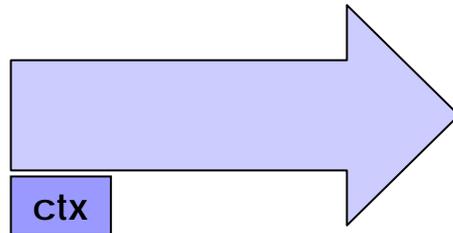


- Stateful interaction

Consumer



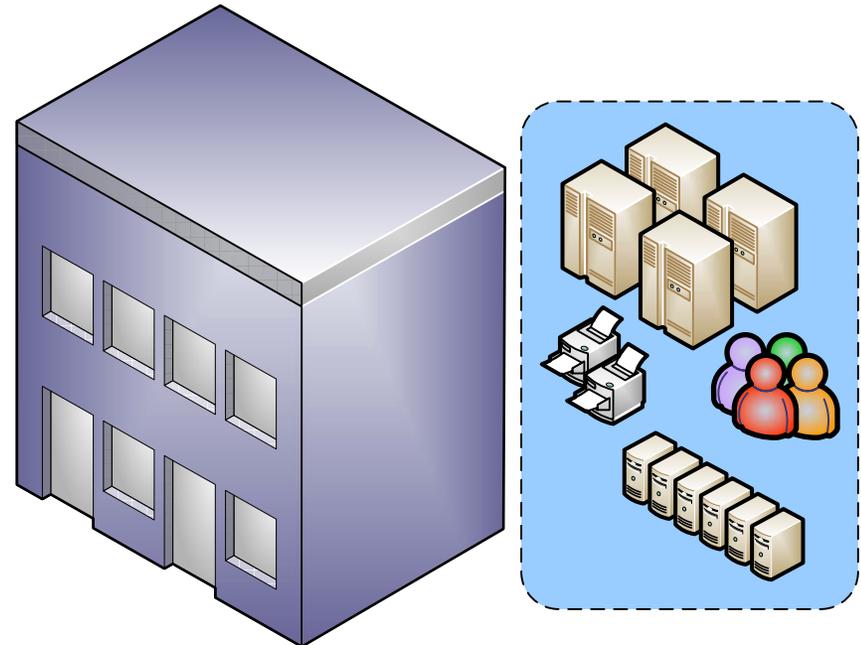
Hey, I am savas



context

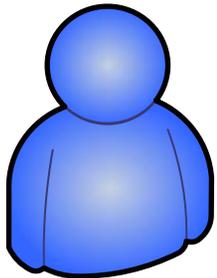


Bank Service

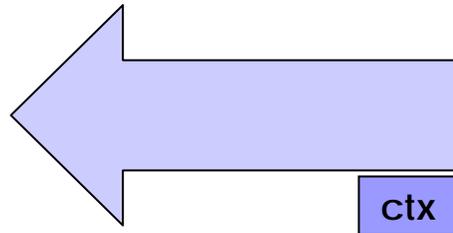


- Stateful interaction

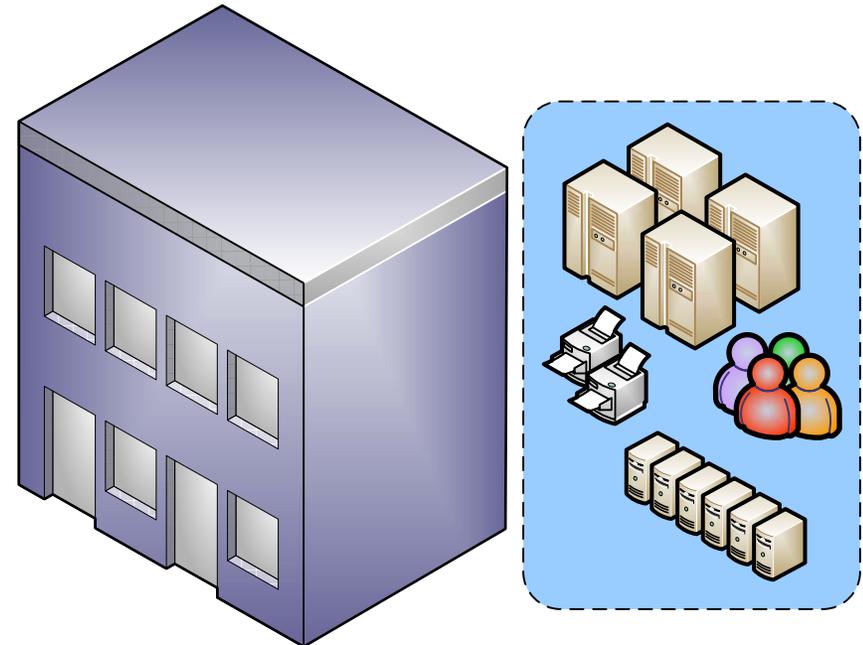
Consumer



Welcome Savas

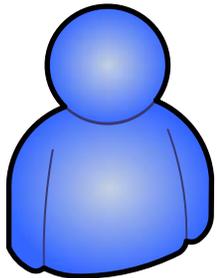


Bank Service

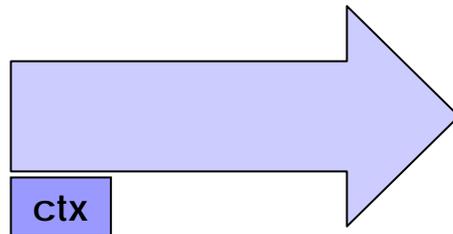


- Stateful interaction

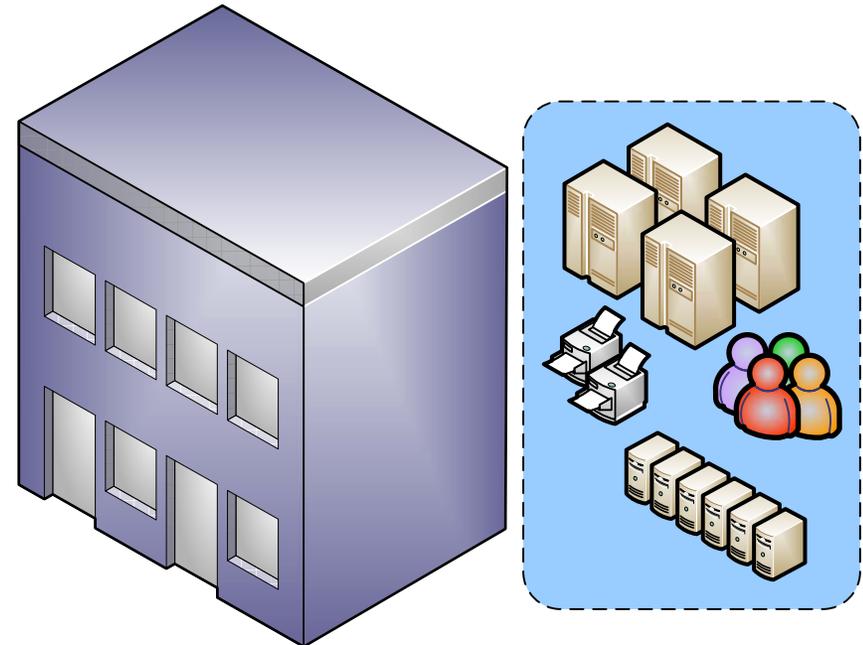
Consumer



I would like £100, please



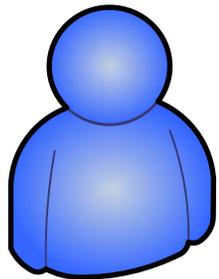
Bank Service



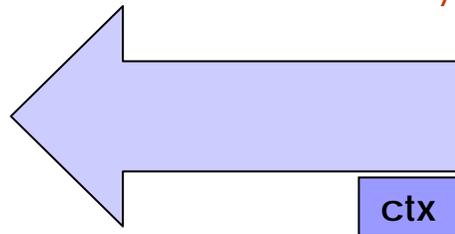
- Stateful interaction



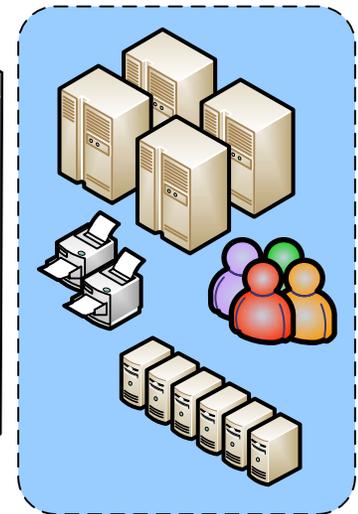
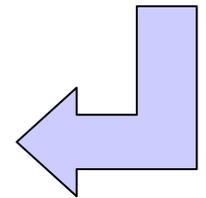
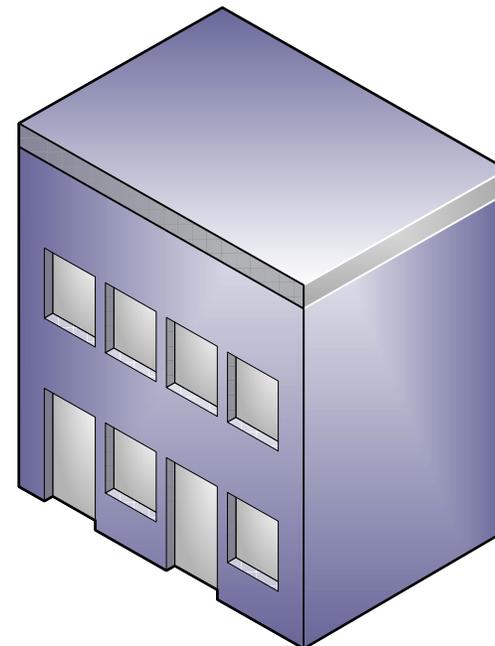
Consumer



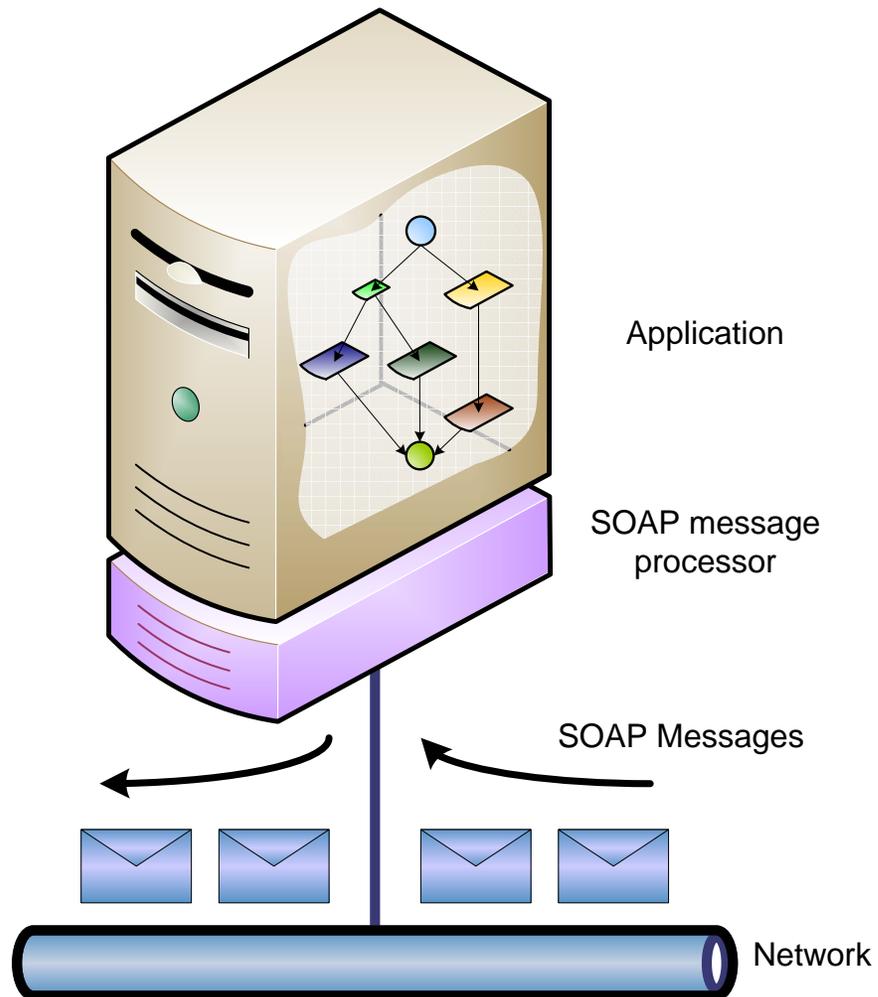
Your £100
Have a nice day!



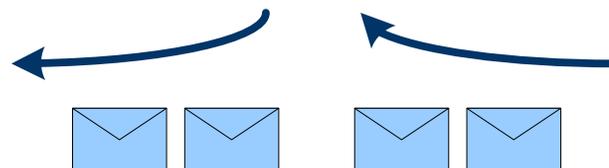
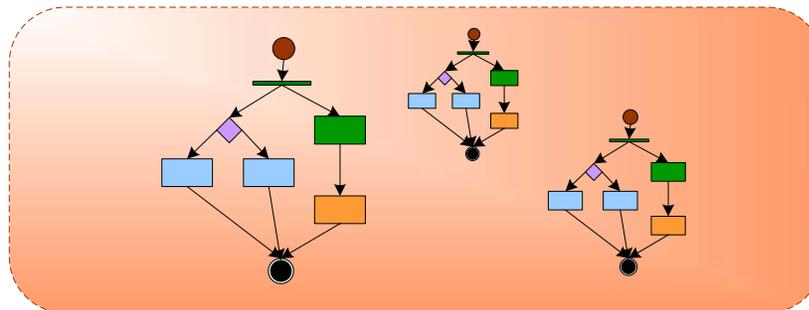
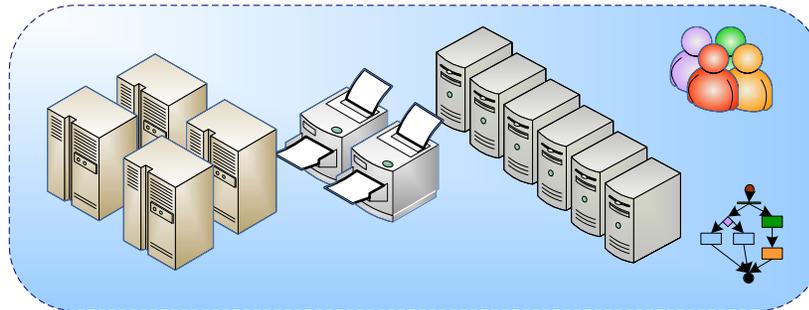
Bank Service



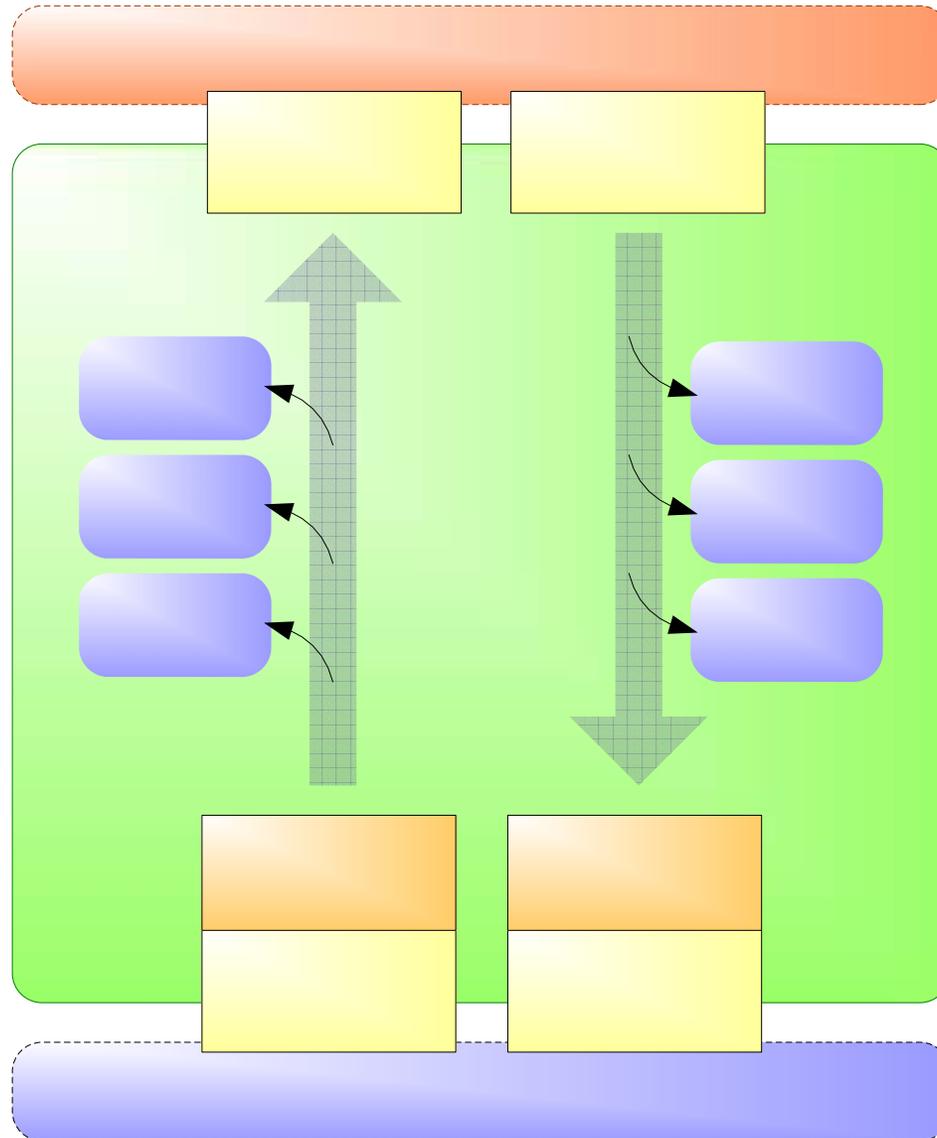
A canonical Web Service



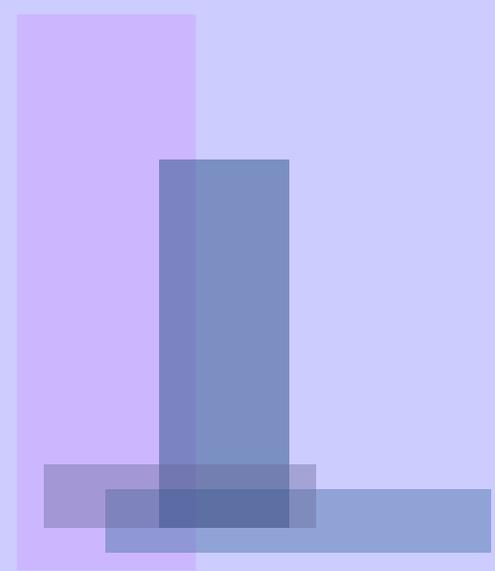
The Anatomy of a Web Service



A canonical message processor

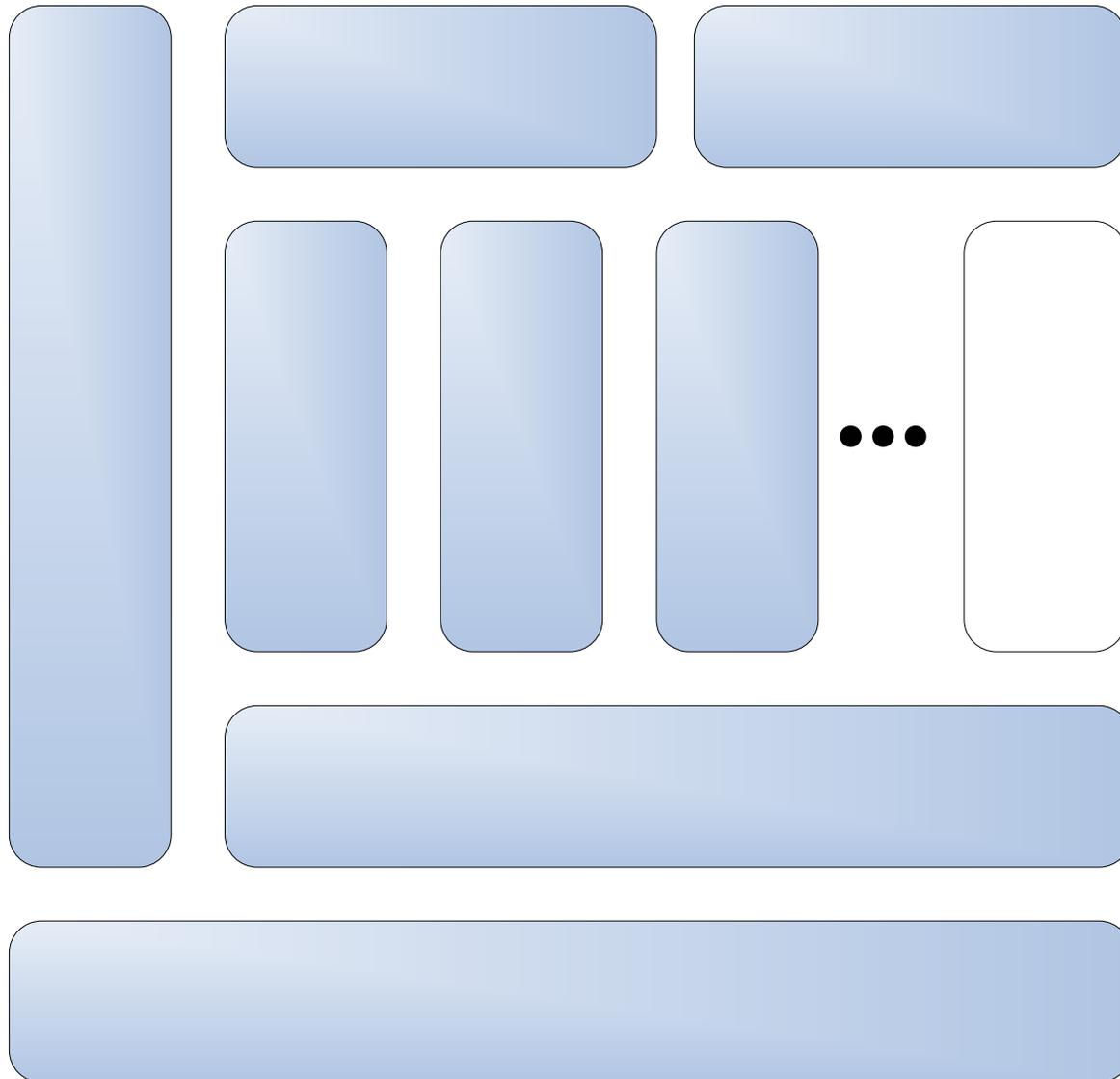


Web Services Grid Application Framework (WS-GAF)



- Milestones
 - OGSi release
 - WS-GAF paper
 - WS-RF release
 - Community concerns over WS specification instability
- We now focus on creating applications and demonstrating ideas
 - Feedback from the community
 - Decided to create risk-based profiles for building Grid applications based on risk/value assessment

The WS-* space



- Application-specific requirements
- Stability of infrastructure
 - WS-I Profile 1.0a
 - WS-I Security Profile (draft)
- Interoperability and adoption
- Composability
- Tool support
- Previous experience
- Documentation and training
- Vendor support
 - IBM **AND** Microsoft?
 - Microsoft?
 - IBM?
 - Sun/Oracle?
 - Others?

Risk/value
assessment

The specification with regards to a characteristic is considered to be of:
“no risk”, “low risk”, “medium risk”, “high risk”

The specification with regards to a characteristic is considered to be of:
“high value”, “medium value”, “low value”, “no value”

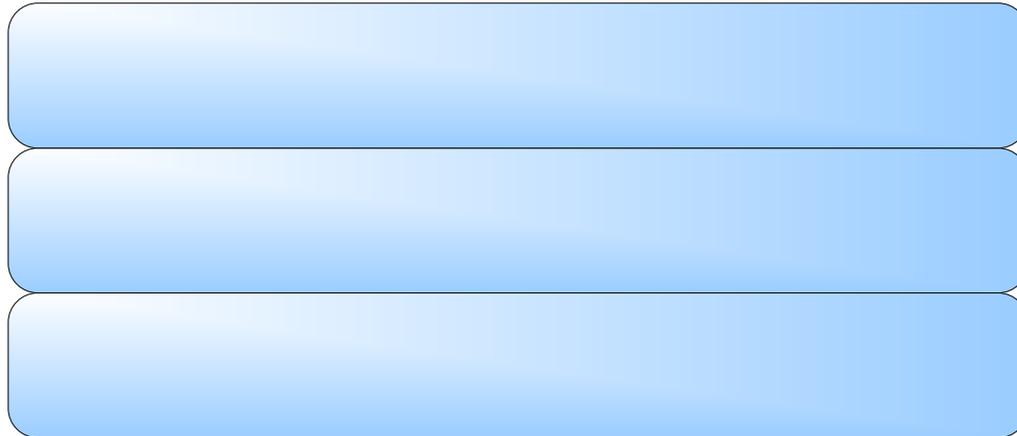
From “Assessing the Risk and Value of Adopting Emerging and Unstable Web Services Specifications”, Savas Parastatidis, Jim Webber (to be presented at IEEE SCC 2004)



- Do not treat scientists as “lab rats”
- Aims for “Production Grids”
 - Low-risk – longevity
 - Interoperability
 - Developer productivity
 - focus on the science and not on taming the technology
 - Focus on stability, wide adoption, tool support where possible
 - Assess specifications before adoption
 - Adopt less stable specs only if definitely required
 - functionality needed
 - implementation(s) available
 - better alternative than building our own
 - design as architecture rather than for particular specification

- Investigations into new approaches, different conceptual models, emerging specifications, etc.
 - “Experimental Grids”
 - But...understand risks of early adoption
 - be prepared for instability or (worst case) abandonment
 - interoperability hurdles

- Our goals
 - Meet Grid requirements
 - Propose a solution based on current WS specifications and practices
 - Emphasize the importance of high-level services
 - Build using specs in WS-I profiles (SOAP, WSDL, UDDI, WS-Security)



- Stateful interactions
 - Contextualisation

BPEL (message correlation), service-specific means (explicit context propagation), etc. etc. etc.

- Resource identification

URN: Uniform **R**esource Names, URIs, or any other logical name

- Metadata
 - Grid Resource Specification

An XML document

- Lifetime management of resources

Application domain or service specific

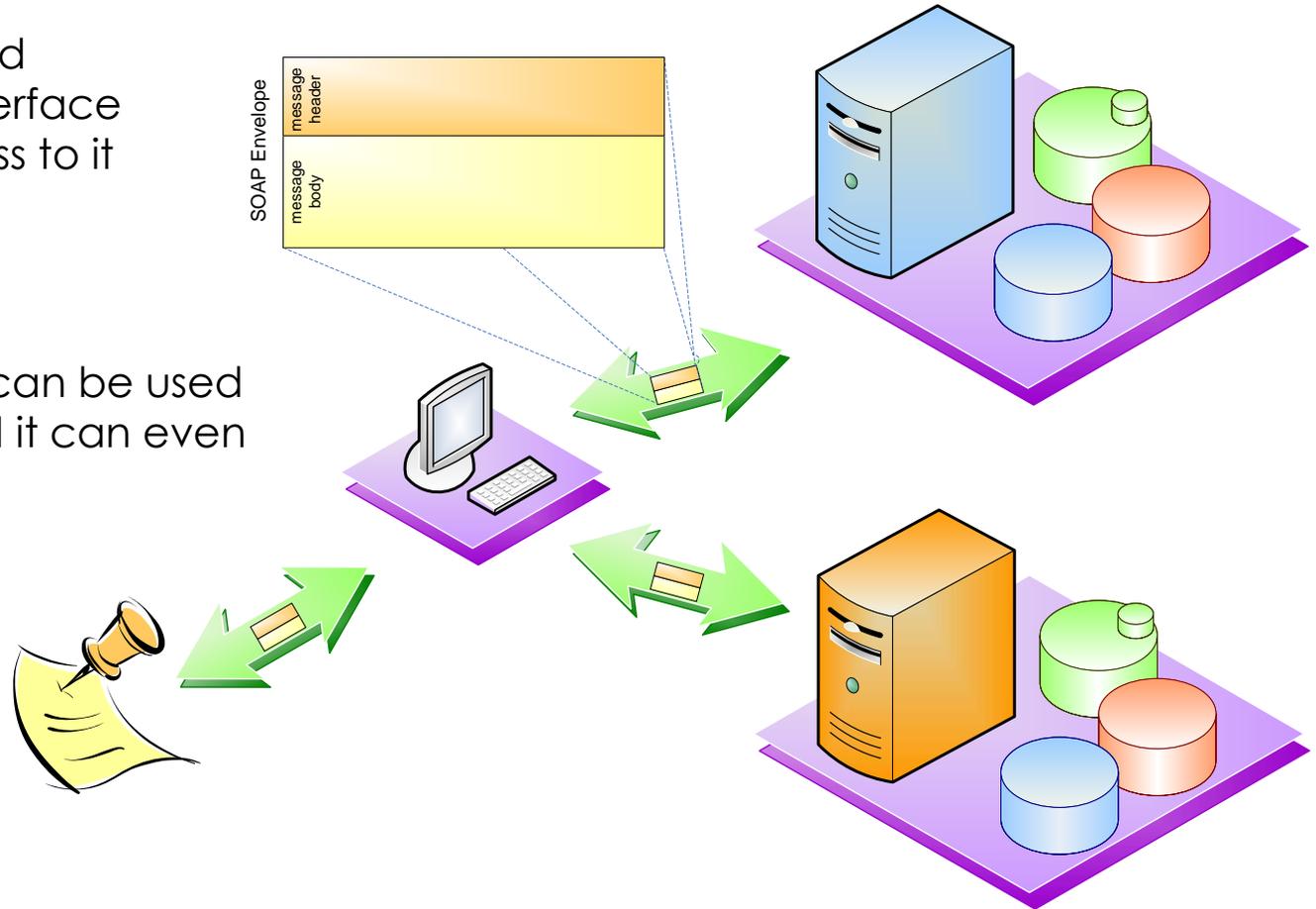
- Security

WS-Security

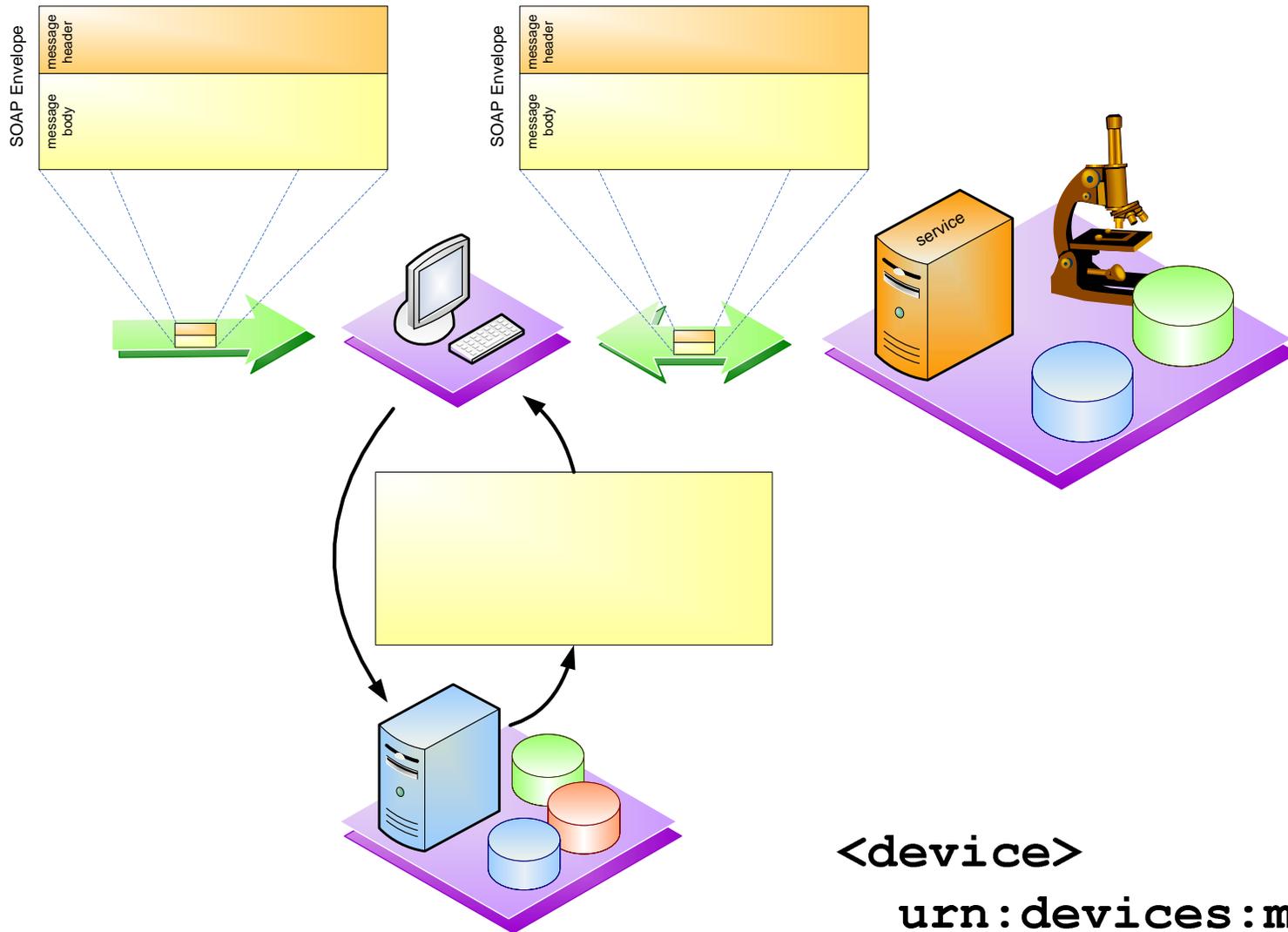
- Resources are encapsulated by services
- There are cases where resources need to be identifiable outside an organisation's boundaries
- Logical names
 - Everlasting, uniform resource identifier (Uniform Resource Name, URN)
 - Can be stored in a database or printed in a journal
 - Decoupling of identity from interface

The resource is identified separately from the interface that can provide access to it

The resource identifier can be used with many services and it can even be printed on a journal



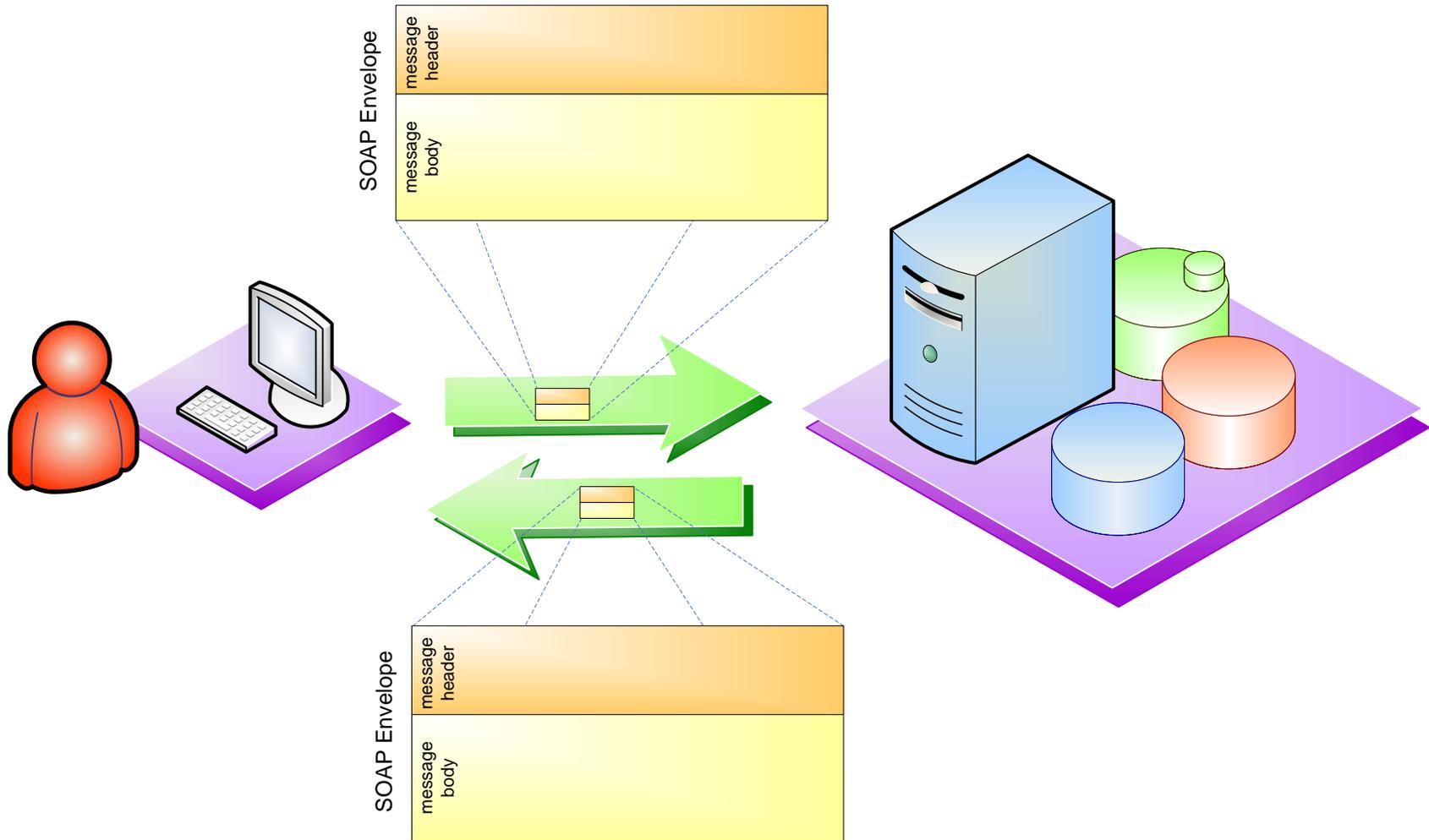
Example: Using a registry



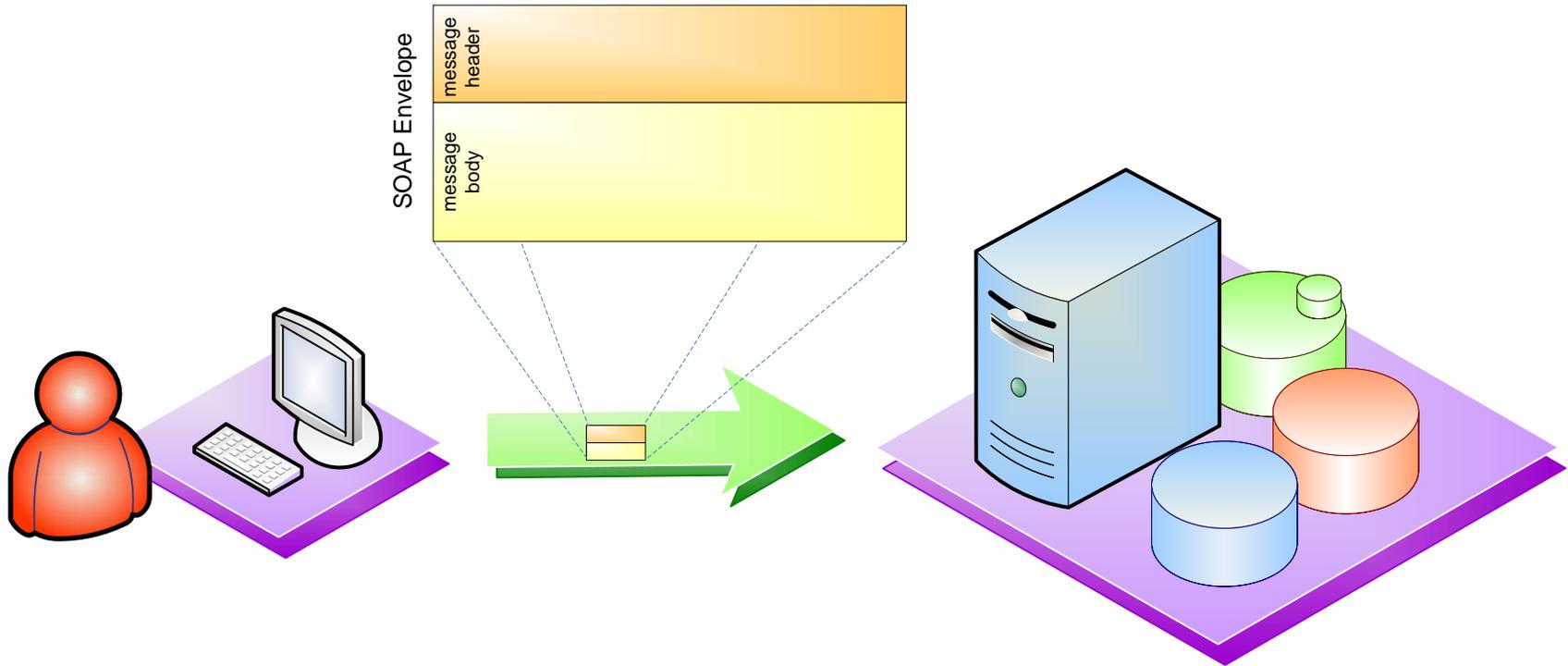
```
<device>  
  urn:devices:microscopy  
</device>
```



Examples (using URNs)



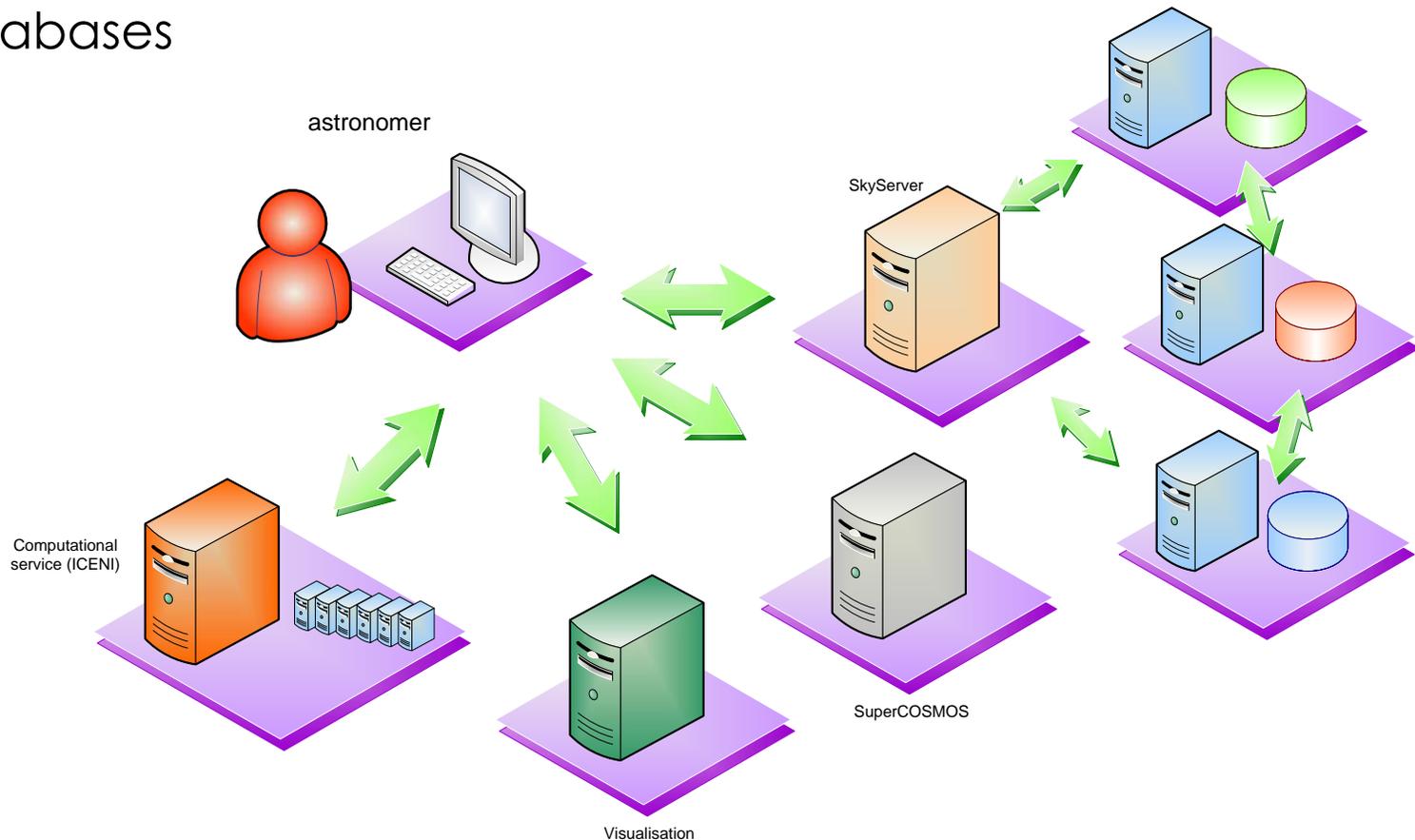
Examples (using URNs)



- We need to build large-scale Grid applications using Web Services in order to find out what is actually required
- Aims
 - Define the characteristics of a “typical” Grid application
 - Demonstrate the applicability of the WS-GAF approach in building Grid applications
 - Learn from the challenges of constructing a truly global, distributed, scalable, loosely-coupled application
- Working on a “typical”, global-scale Grid application with international partners

Searching for "White Dwarfs"

- SkyServer and SuperCOSMOS archive
- Combine info from other databases
- Utilise computational resources (with security in mind)
- Visualise



- GOLD UK e-Science pilot project
- Use RFIDs to identify bottles
- Bottles move between organisations
- Security/trust
- Databases – continuous queries – large amount of information

- WS are a good technology for building Grid applications
- The WS space will become clearer over time
 - initially only low-level infrastructure specifications standardised
 - later, high-level specs will stabilise (notification, workflow...)
- We see benefits in building services from basic, stable Web Service specifications
 - building on a stable infrastructure is of key importance to us
- Investigation on new technologies is not discouraged
- Try to avoid over-dependence on specific technologies

- We should spend our time on high-level services and science
 - industry will sort out the lower level infrastructure for us



- Paul Watson (Paul.Watson@newcastle.ac.uk)
- Savas Parastatidis (Savas.Parastatidis@newcastle.ac.uk)
- Jim Webber (Jim.Webber@newcastle.ac.uk)

Web Services Grid Application Framework (WS-GAF)
<http://www.neresc.ac.uk/ws-gaf>

Mailing list (>90 people from all over the world)
ws-gaf@newcastle.ac.uk

Join by sending a message to mailbase@newcastle.ac.uk including the following line in the body

```
join ws-gaf YourFirstName YourLastName
```

- DTI
- JISC
- UK e-Science Core programme