

Grid Components and Tools

Philipp Wieder
Research Centre Jülich
COMDEX Greece 2004
Athens, November 5, 2004



Contents

- Setting the scene:
Why do we need Grid components & tools?
- Components:
Which components do we need?
- Architecture:
What is state-of-the-art?
- Tools
What to look for (links e.g.)?
- Future:
What to expect (& to read)?



Grid: Vision

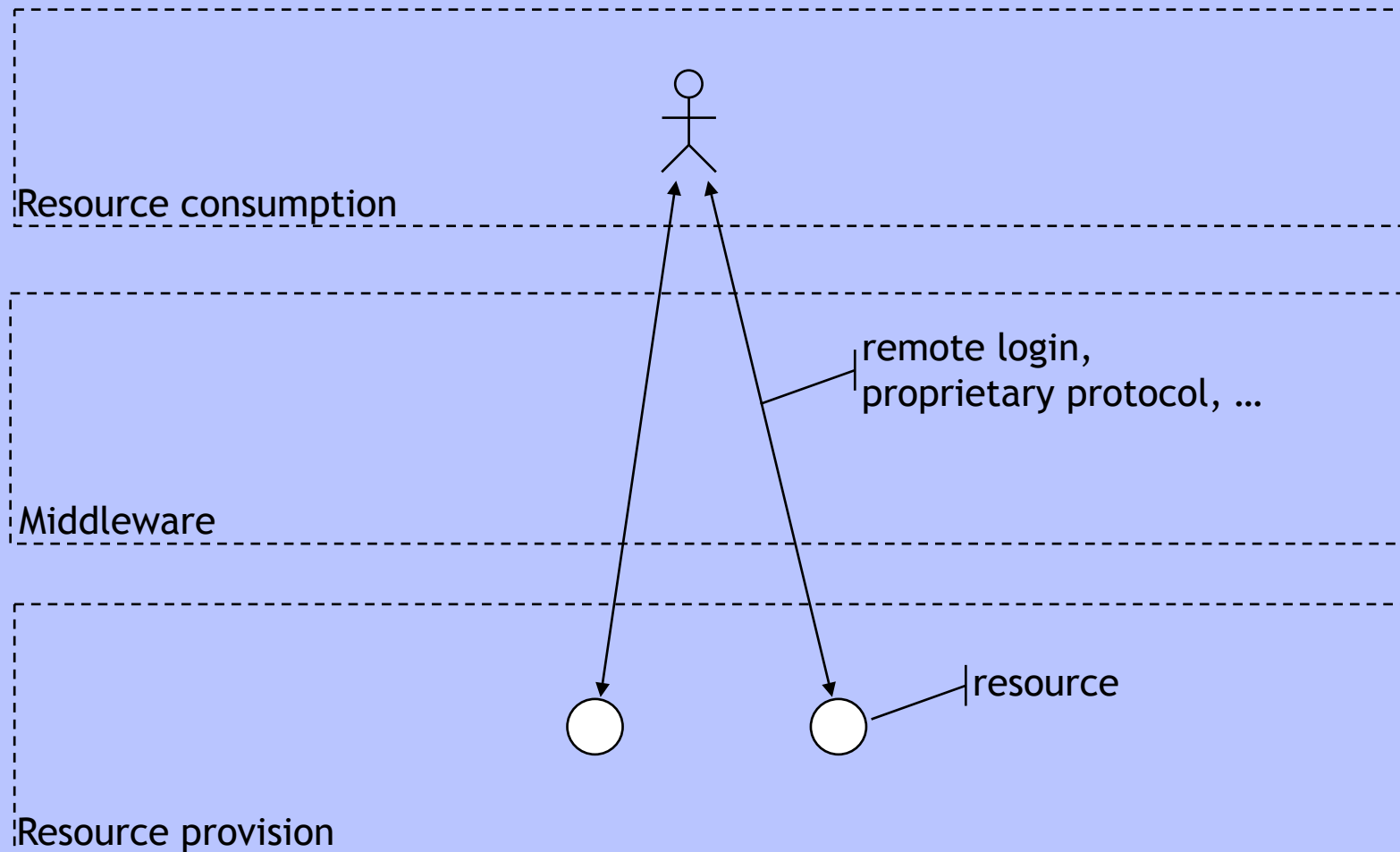
“... as today’s enterprise systems are transformed from separate computing resource islands to integrated, multitiered distributed systems, service components can be integrated dynamically and flexibly, both within and across various organizational boundaries.” Foster et.al., “The Physiology of the Grid”



- Co-ordinated sharing of resources
- Dynamic virtual organisations
- Interoperability & standards

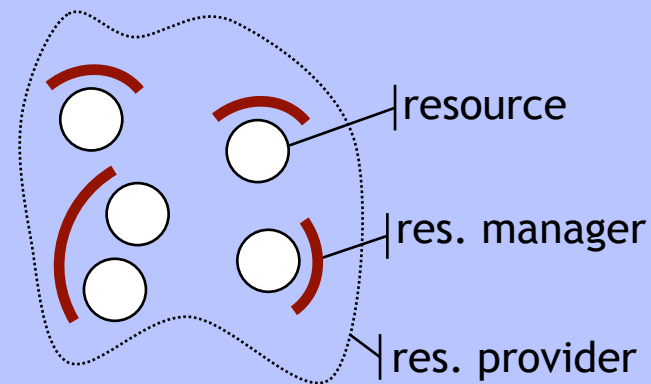


Grid: Landscape



Components I: Resource provision

- Resource types:
 - Computing, data, network, software, ...
 - ... but also: sensors, instruments, humans
- Resource managers:
 - Provide abstraction layer
 - Access point for middleware
- Resource providers:
 - Autonomy
 - Max. utilisation/profit/...



In general: Resources not exclusively for Grid use



Components II: Middleware

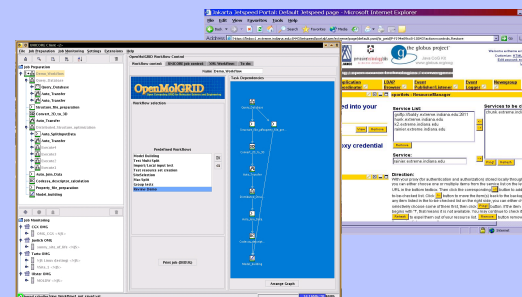
- Provide information
 - Dynamic resource/user/... information
- Support work flow execution
 - Orchestration & co-ordination of Grid components
 - Control flow of work
- Manage resources
 - Broker, scheduler, ...
- Enable Virtual Organisations
 - Ad-hoc creation of Grids

Middleware should be transparent



Components III: Resource consumption

- APIs & SDKs
 - Integration of Grid functions into applications
 - Standard API for user interaction desirable
- Work flow components
 - Providing Grid functions to end-user
 - Work flow submission, monitoring, control
- Portals
 - Grids for communities



Decrease user effort to Grid utilisation



Common properties

A Grid has to be

- **Easy-to-use:** end-users, but also developers
- **Open & pervasive:** see WWW
- **Interoperable:** protocols, languages, semantics
- **Secure:** AAA, trust-models, integrity, ...
- **Scalable:** increase of Grid “nodes” transparent
- **Reliable:** complex Grid \neq complex failure modes
- **Persistent:** personal, local & global



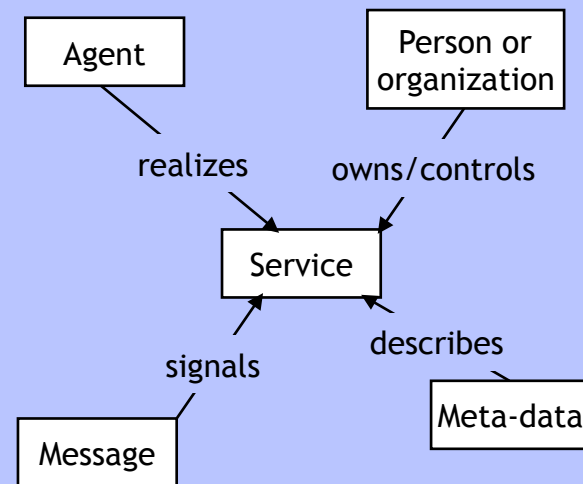
The Service Approach

➤ Web Services

- Technological foundation of many Grid developments (WSDL, SOAP, ...)
- Inheritance of pro.s (tooling, ..) and con.s (performance, ...)

➤ Service Oriented Architecture

- Encapsulation of existing functionality as services
- Component architectures easy portable



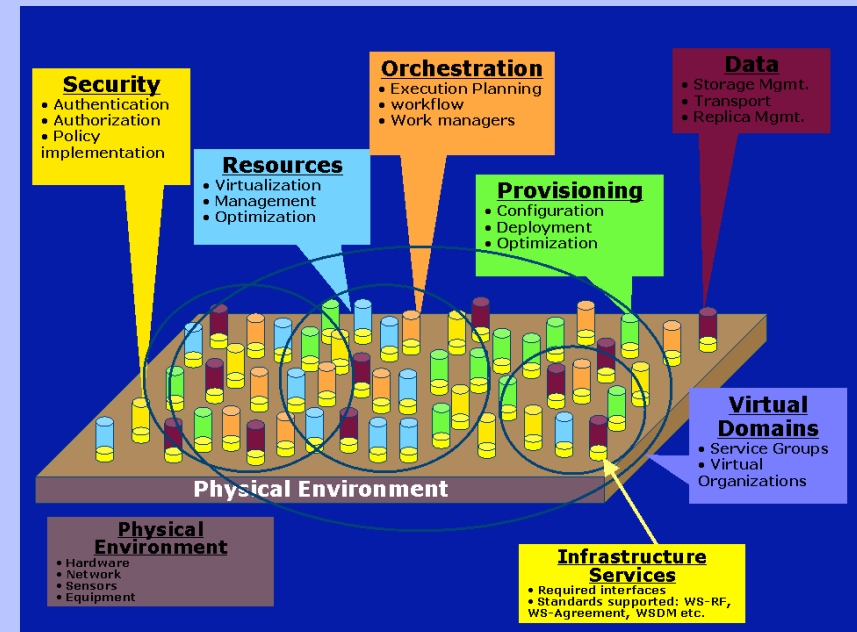
Service oriented model, Source: W3C, WS Architecture

Grid + WS + SOA = ?



The Service-Oriented Grid

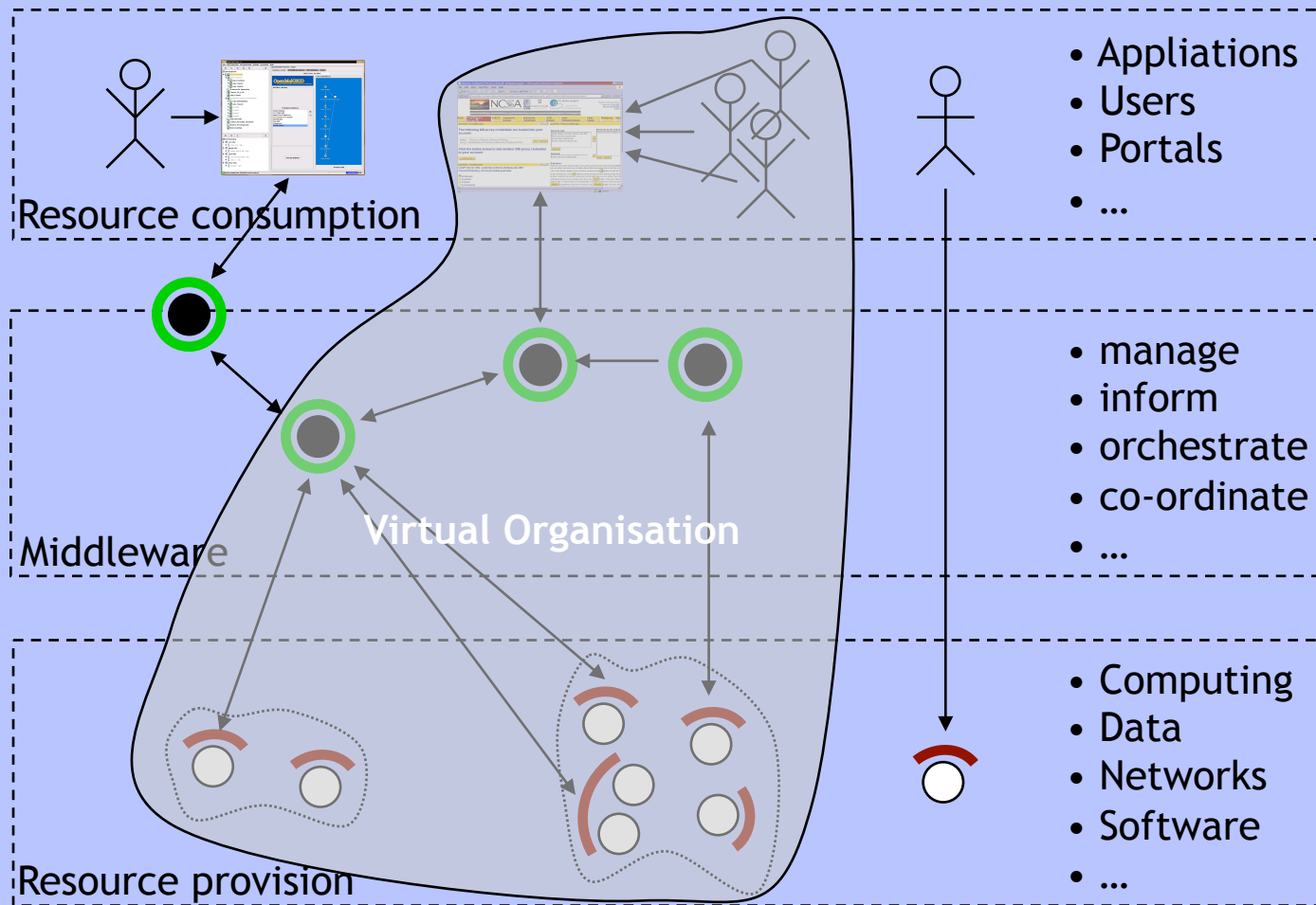
- **Grid Service**
 - An extended Web Service?
 - A service providing Grid functionality?
- **Open Grid Services Architecture (OGSA)**
 - Framework for integration, virtualisation & management of distributed systems
 - Definition of a core set of services (+ resource models, interaction, ...)



“OGSA framework”, OGSA version 1.0, © Global Grid Forum



Putting it together ...



Tool selection: criteria

- Extensibility
- Which components are provided?
- Is all necessary functionality exposed?
- Standards compliance
- Interoperability with other tools
- As well as:
 - License (open source, commercial, ...)
 - Support (community, professional, ...)
 - Documentation
 - Roadmap



Selected tools (academic)

- Condor-G
<http://www.cs.wisc.edu/condor/condorg>
- EGEE gLite
<http://glite.web.cern.ch/glite>
- GridLab
<http://www.gridlab.org>
- Globus & CoG Kit
<http://www.globus.org> & <http://www-unix.globus.org/cog/java/>
- UNICORE
<http://unicore.sourceforge.net>



Selected tools (commercial)

- Avaki
<http://www.avaki.com>
- GridSystems
<http://www.gridsystems.com>
- Platform LSF
<http://www.platform.com/products/LSFfamily>
- Sun N1
<http://www.sun.com/software/n1gridsystem>



What to expect in future?

- Present Grid systems:
 - Static, closed Grid installations often with one specific focus
 - Non-standard, monolithic, hard-to-install systems
 - Little interoperability by design
- Future Grid systems:
 - OGSA-compliance
 - Support for standards (GGF, OASIS, W3C)
 - Built-in interoperability
 - Semantic Grids
 - A Grid programming language



Information sources

- Next Generation Grids 1 & 2
<http://www.cordis.lu/ist/grids/pub-report.htm>
- National Science Foundation Middleware Initiative (NMI)
<http://www.nsf-middleware.org>
- Baker et.al., “Grids and Grid technologies for wide-area distributed computing”
www.buyya.com/papers/gridtech.pdf
- Philipp Wieder
ph.wieder@fz-juelich.de

