

Current Status and Future Direction

Open Grid Services Architecture

Hiro Kishimoto, Ph.D.
OGF OGSA-WG, co-chair
GFSG and OGF Board member
Research Fellow, Fujitsu
Visiting Professor, National Institute of Informatics

Agenda



- Open Grid Services Architecture Overview
- OGSA specification schedule
- OGSA and EGA reference model
- OGSA information/data modeling

The Open Grid Services Architecture



- An open, service-oriented architecture (SOA)
 - Resources as first-class entities
 - Dynamic service/resource creation and destruction
- Built on a Web services infrastructure
- Resource virtualization at the core
- Build grids from small number of standards-based components
 - Replaceable, coarse-grained
 - e.g. brokers
- Customizable
 - Support for dynamic, domain-specific content...
 - ...within the same standardized framework

SOA & Web Services: Key Benefits

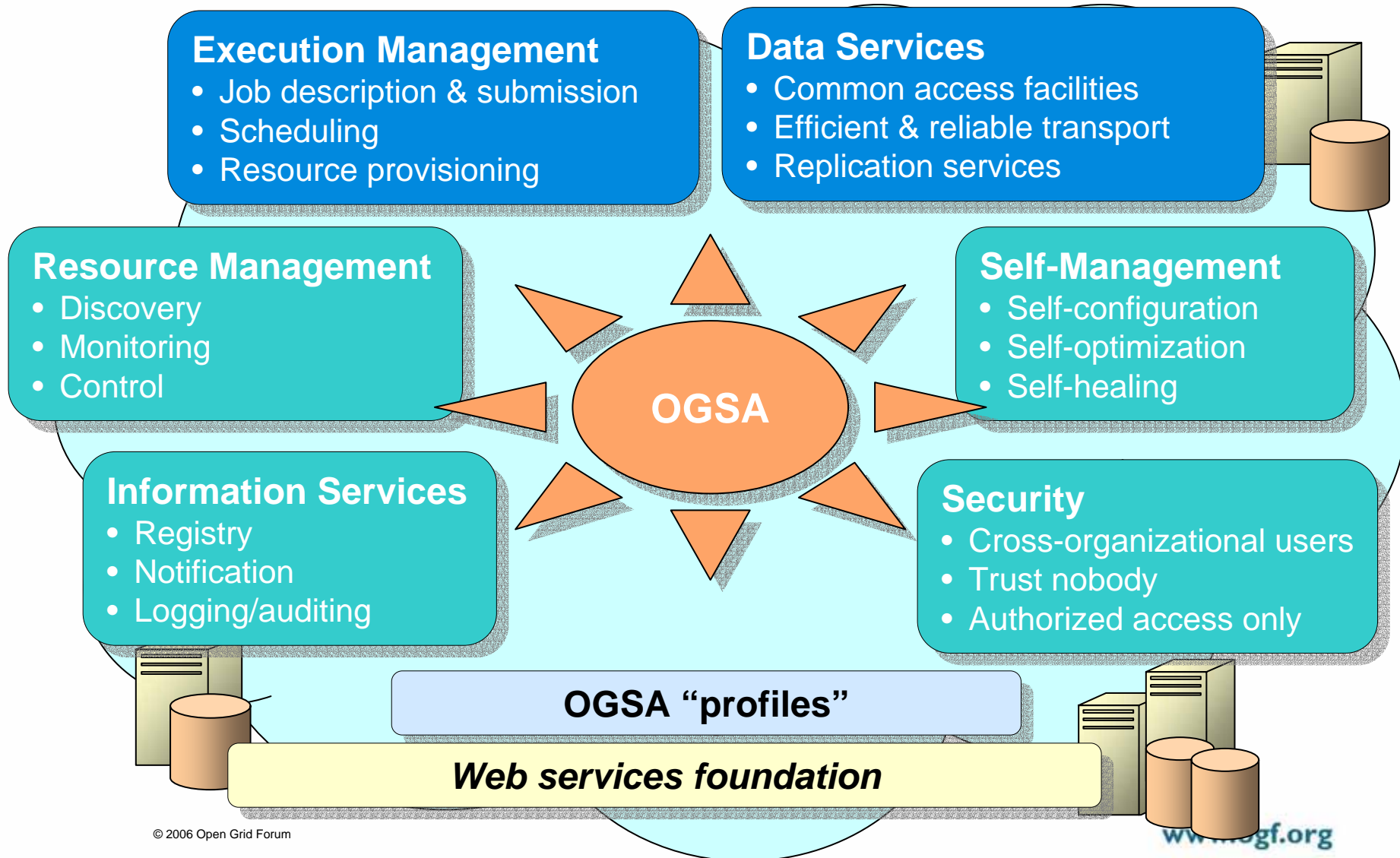
SOA

- Flexible
 - Locate services on any server
 - Relocate as necessary
 - Prospective clients find services using registries
- Scalable
 - Add & remove services as demand varies
- Replaceable
 - Update implementations without disruption to users
- Fault-tolerant
 - On failure, clients query registry for alternate services

Web Services

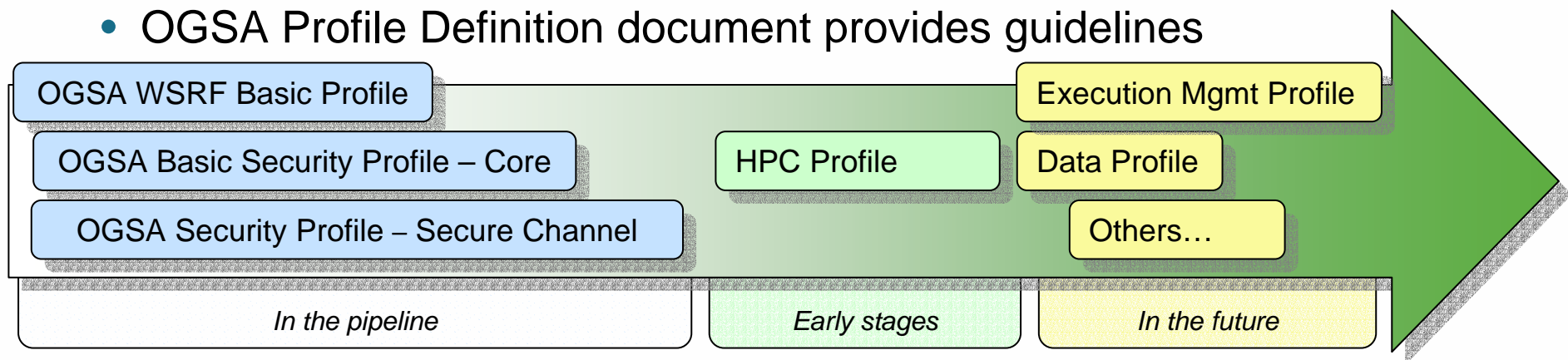
- Interoperable
 - Growing number of industry standards
- Strong industry support
- Reduce time-to-value
 - Harness robust development tools for Web services
 - Decrease learning & implementation time
- Embrace and extend
 - Leverage effort in developing and driving consensus on standards
 - Focus limited resources on augmenting & adding standards as needed

OGSA Capabilities

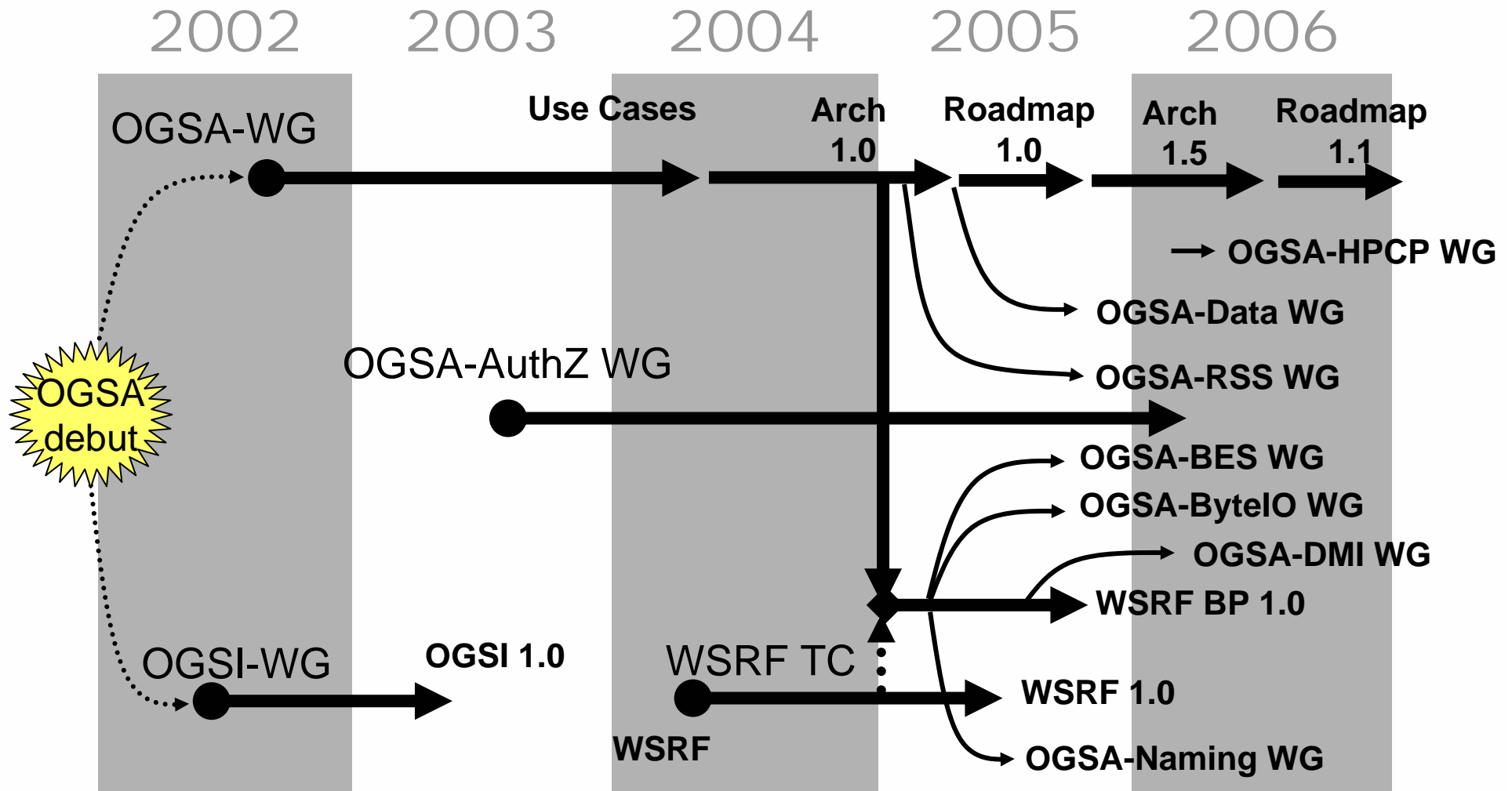


OGSA Profiles

- The normative definition of OGSA
 - Styled on WS-I profiles to promote interoperability
 - Define specific usage patterns
 - e.g. execution management
 - Basis for claims of conformance
 - *“My scheduler conforms to the OGSA Execution Management Profile...”*
- Include specifications developed by OGF **and** by other bodies
 - Issue: How mature and widely adopted?
 - OGSA Profile Definition document provides guidelines



OGSA and Fellow WGs

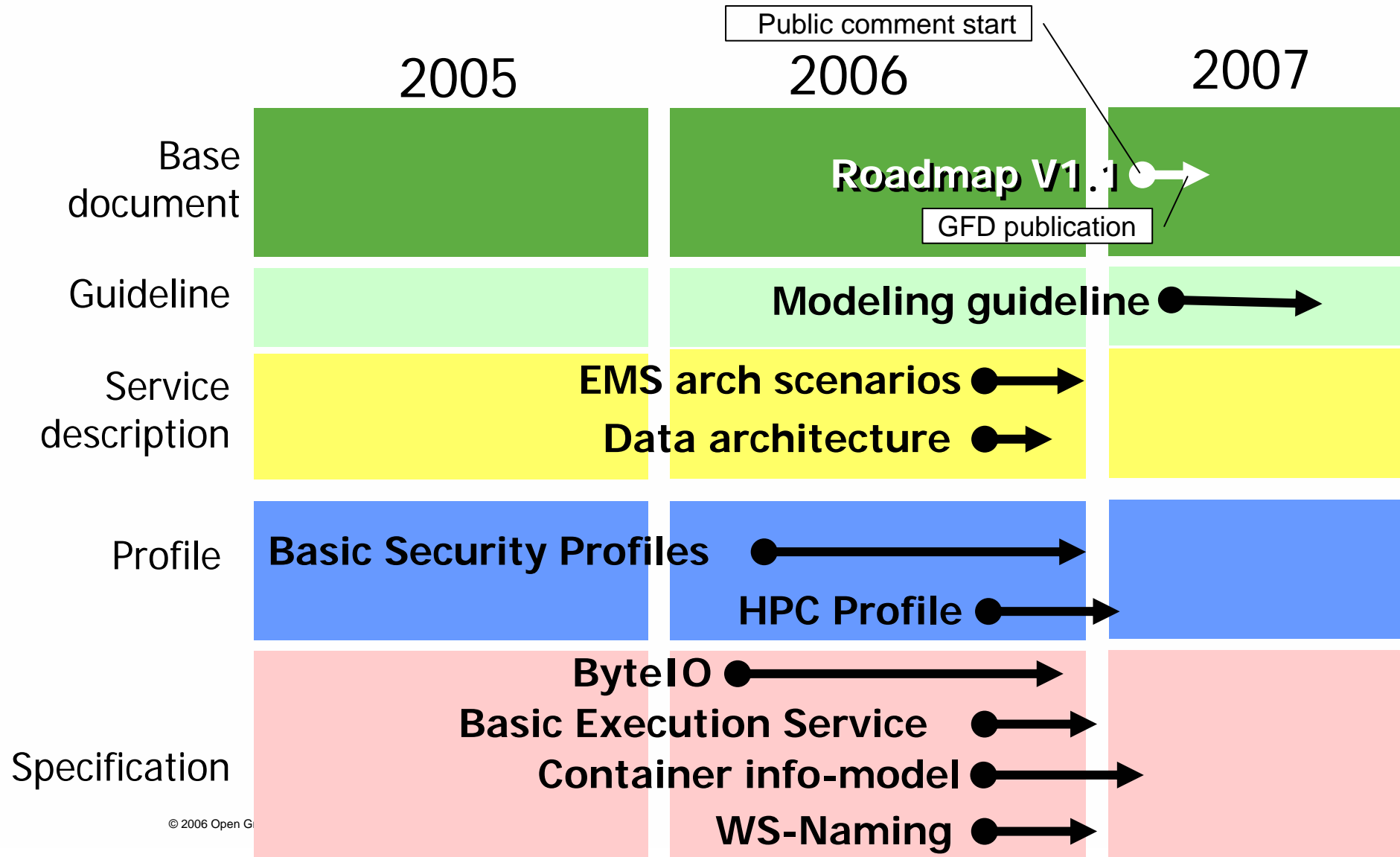


Published OGSA Documents



| | 2004 | 2005 | 2006 |
|---------------------|------------------------|--|----------------------|
| Base document | ✓ Use Cases (GFD-I.29) | ✓ Architecture V1.0 (GFD-I.30) V1.5 (GFD-I.80) ✓ ✓ Glossary V1.0 (GFD-I.44) V1.5 (GFD-I.81) ✓ ✓ Resource Management (GFD-I.45) | Roadmap (GFD-I.53) ✓ |
| Guideline | | Profile Definition (GFD-I.59) ✓ | |
| Service description | | | |
| Profile | | OGSA WSRF BP (GFD-R-P.72) ✓ | |
| Specification | | JSDL (GFD-R-P.56) ✓ | |

OGSA Schedule



Document schedule (1)



| Document name | First draft available | Ready for PC | GFD publication |
|--|--------------------------|--------------------------|------------------------|
| OGSA WSRF Basic Profile 1.0 | May 2005 May 2005 | Sept. 2005 Sept. 2005 | June 2006 Sept 2006 |
| OGSA Architecture & Glossary 1.5 | Oct. 2005 Oct. 2005 | Nov. 2005 Mar. 2006 | July 2006 Sept 2006 |
| OGSA Security Profile 1.0 – Secure Channel | Sept. 2005 Sept. 2005 | Nov. 2005 Feb. 2006 | Aug. 2006 Oct. 2006 |
| OGSA Basic Security Profile 1.0 – Core | Sept. 2005 Sept. 2005 | Nov. 2005 Feb. 2006 | Aug. 2006 Oct. 2006 |

Plan date
Actual date

completed

delayed

Document schedule (2)



| Document name | First draft available | Ready for PC | GFD publication |
|--|------------------------|-------------------------|------------------------|
| OGSA EMS Architecture scenarios 1.0 | July 2006 July 2006 | Sept. 2006 Oct. 2006 | Nov. 2006 Dec. 2006 |
| OGSA EMS Architecture scenarios capability | TBD (3~6 month) | | |
| OGSA EMS Architecture | Sept. 2006 ? | Nov. 2006 ? | Feb. 2007 ? |
| OGSA roadmap | Sept. 2006 | Nov. 2006 | Jan. 2007 |

Plan date
Actual date

completed

delayed

Document schedule (3)



| Document name | First draft available | Ready for PC | GFD publication |
|---|-------------------------|-------------------------|-------------------------|
| Guidelines for Information Modeling for OGSA Entities (explain necessary process) GFD-I | Jan. 2006 Jan. 2006 | Sept. 2006 Feb. 2007 | Jan. 2007 April 2007 |
| container information model (appendix of BES spec) GFD-R.P | Jan. 2006 Jan. 2006 | Sept. 2006 Oct. 2006 | Jan. 2007 Feb. 2007 |
| Information Modeling in OGSA XML schema and semantics Best practices GFD-R.P Also wiki page | Sept. 2006 Nov. 2006 | Sept. 2006 Feb. 2007 | Jan. 2007 April 2007 |

These document should go together!!

Plan date
Actual date

completed

delayed

OGSA and EGA reference model



- Glossary comparison
 - Identify similarities and differences between terminology.
- ERG Ref Model and OGSA
 - Establish mutual understanding
 - Working hypothesis: “OGSA is the SOA and Web service based instantiation of Reference Model”

Format

4 Glossary of Terms

| Term | Definition |
|-------------|--|
| Abstraction | Abstraction involves changing the interface of an object and exposing it in a more useful or appropriate form. It provides a layer of abstraction of objects, making them easier to use. |

2. Terms

Note: In the table.

| Term | Ref's |
|---------------|--|
| A | [7] |
| AAA | Authentication, authorization and accounting. |
| Abstract name | See name. |
| ACID | Four properties that must generally apply to stateful resources used within the context of a transactional unit of work within a traditional, two-phase-commit-enabled transaction system. Briefly: <ul style="list-style-type: none">• Atomicity: Updates must be made in an all-or-nothing fashion.• Consistency: Resources must be left in a consistent state, even in the event of failure. |

Not a problem!

© 2000

14

www.ogf.org

Statistics

| | EGA | OGSA | Note |
|---------|-----|------|--|
| Terms | 42 | 156 | |
| Trivial | 8 | 41 | Expands initials or refers to another definition |
| Overlap | 11 | | |

Overlapping Terms (1)

| EGA | OGSA | Notes |
|------------------------------------|--------------|---|
| CIM | CIM | Both trivial. |
| Enterprise grid, Grid computing | Grid | OGSA definition is formal; EGA is more explanatory. |
| GGF | GGF | Both trivial – deprecate! :0) |
| Grid component | Component | OGSA definition is brief & formal; EGA explanatory, wider scope. |
| OGSA | OGSA | Both trivial! |
| Provisioning | Provisioning | Not identical, but no real conflict. OGSA was considering revising. |

Overlapping Terms (2)

| EGA | OGSA | Notes |
|----------------|------------|---|
| Resource | Resource | Similar. |
| Service | Service | OGSA is formal & succinct; EGA more explanatory. |
| SLA | SLA | OGSA is a little more explanatory; OGSA mentions “level of service” vs. EGA’s “properties & attributes.” |
| SLO | SLO | No conflict. |
| Virtualization | Virtualize | No conflict; EGA is more explanatory. |

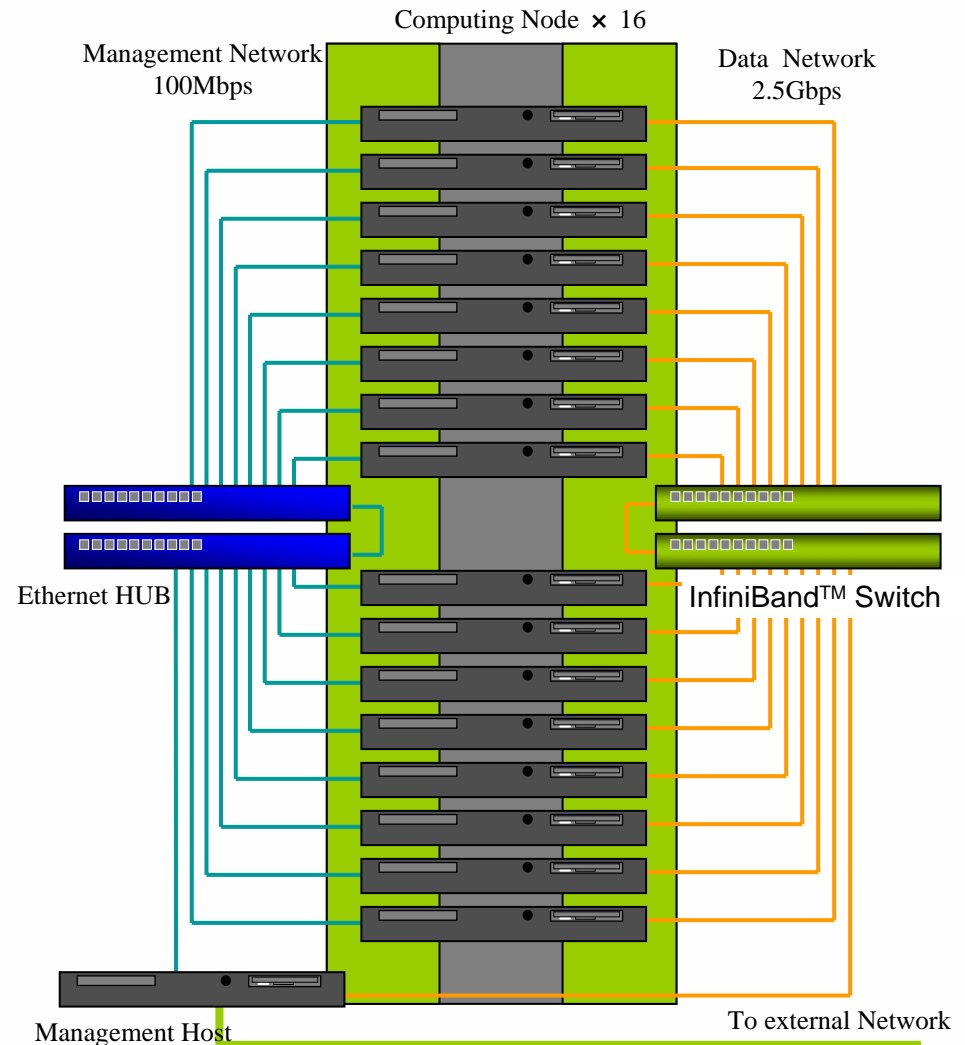
Observations

- No violent disagreement
- EGA definitions tend to be longer and more explanatory
 - A few of the OGSA definitions are similar in style
 - Consider if styles should be merged
- Need to look for non-obvious overlaps
 - There might be some hidden conflicts
- May need to define some new terms
 - e.g., “enterprise”

OGSA HPC cluster usecase



- HPC cluster is set of computational servers connected high-speed network
 - It is not grid in general sense
- Administrator deploy multiple applications on computing nodes
- Users submit computational jobs to cluster (management host)
 - Each job runs on one or more computing nodes exclusively
 - More than one job can run on each node in turns
- EPS (scheduler) prioritize submitted jobs based on administers policy
- EPS monitors and logs note usage for accounting and billing

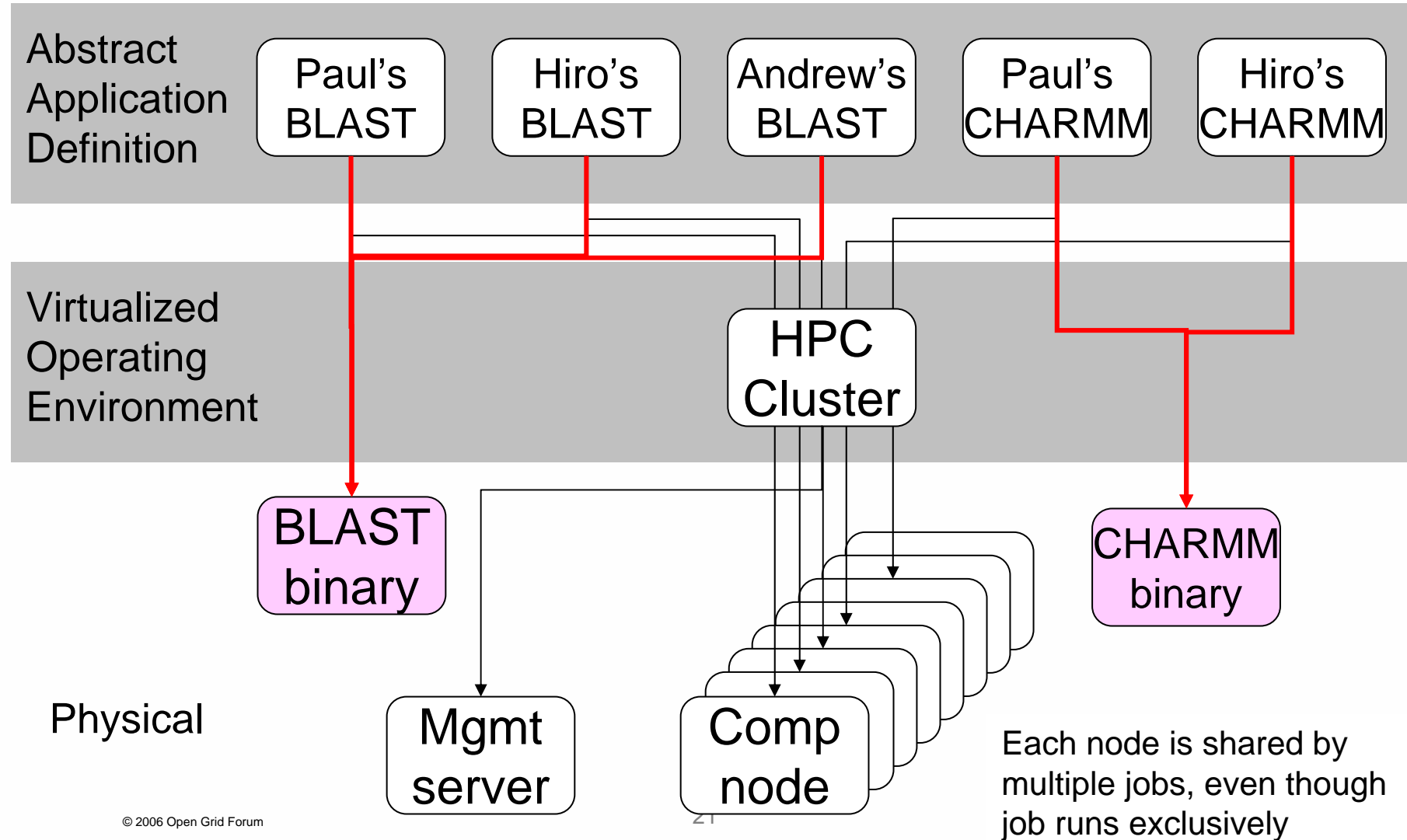


Workload

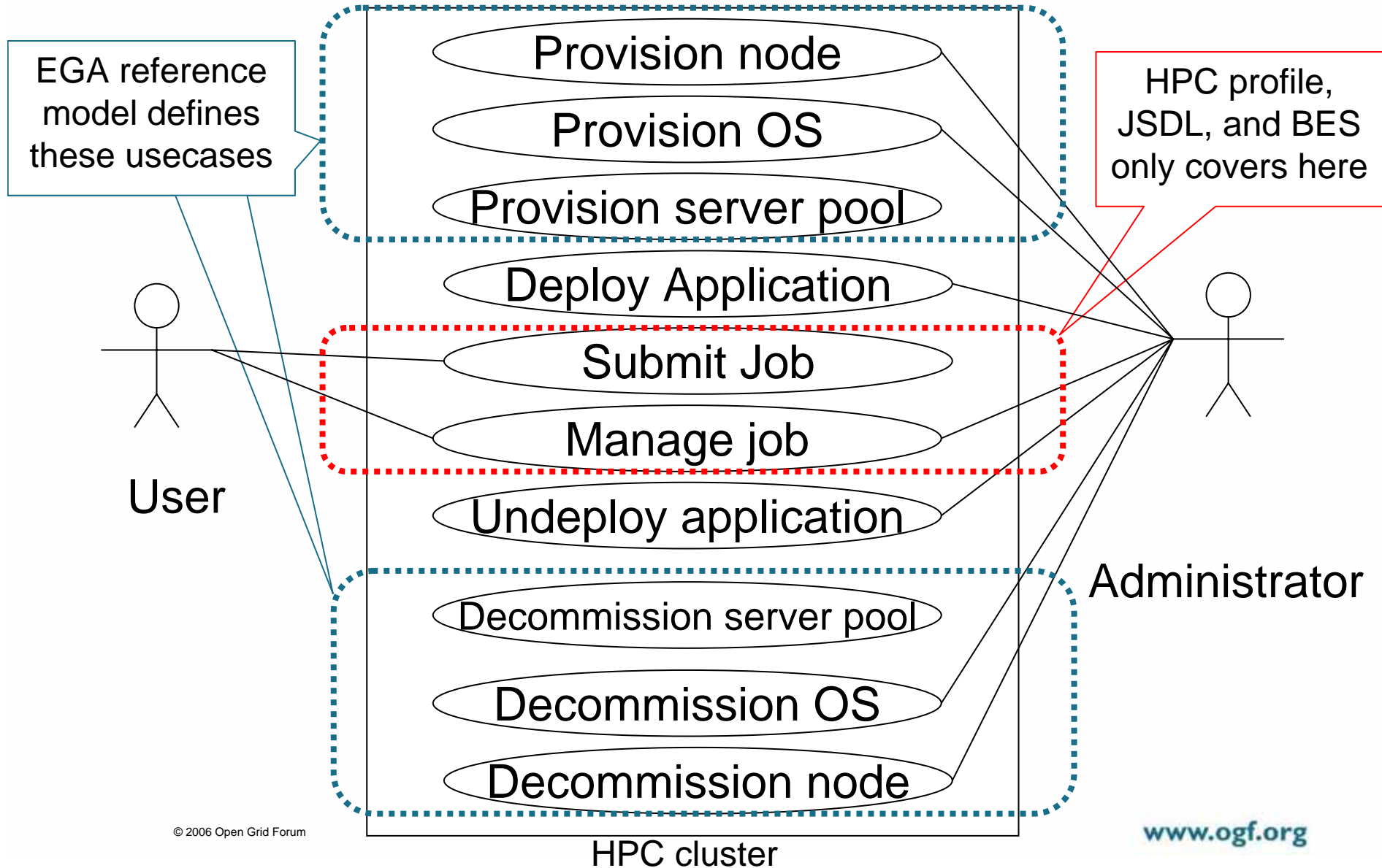


- Non interactive batch workload
- User may submit multiple jobs
 - E.g. workload consists of 100 jobs
 - Each job last several seconds to several days
 - Single job (run on single node) or parallel job (run on multiple nodes)
- Each job is “abstract application”: top level entity of Grid Component tree diagram
 - Job is different form “transactions” of online shopping application

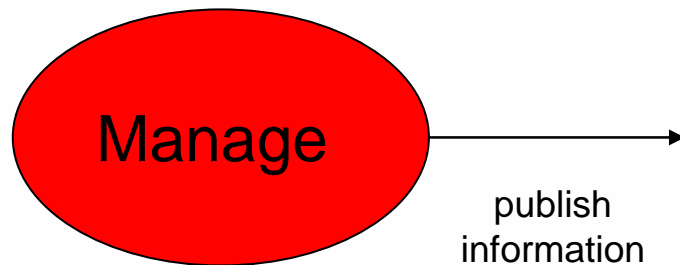
Batch job: grid component mapping



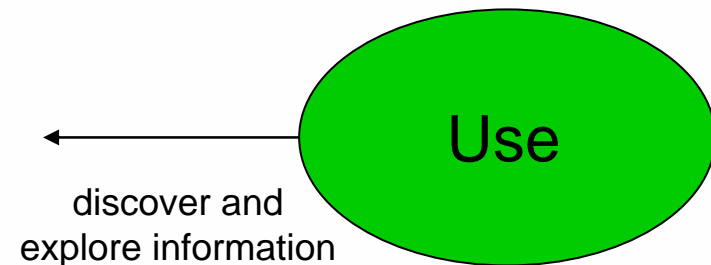
OGSA HPC usecase diagram



Resource Usage



- monitor
- provision & configure
- workload management
- info / storage mgmt
- ...



- execute workload/jobs
- access information
- exploit special capabilities
- ...

Resource Information Usage

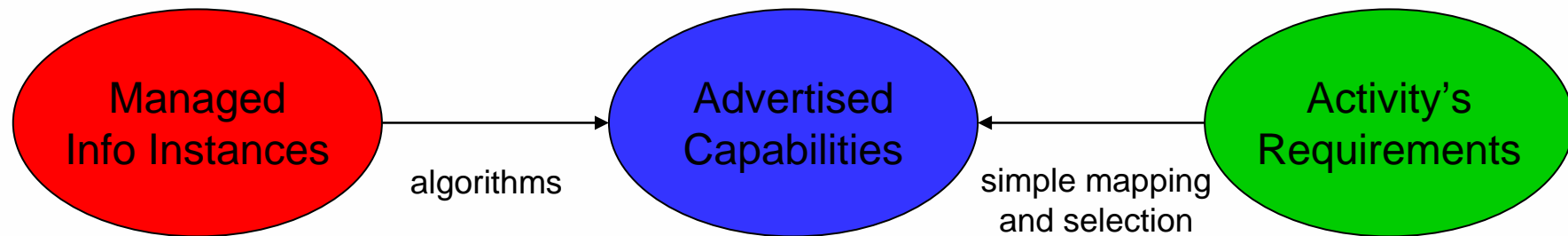
(An Observation)



- Resources are modeled for 3 purposes
 - Managing resources/information in the system/network
 - Advertising existence & capabilities of resources in the system/network
 - Expressing (requesting) set of requirements to accomplish an activity in the system/network
- System/Network management is focused on managing.....Grid is focused on advertising and requesting
 - System/Network management: models are granular & detailed; examples are CIM, GLUE, Unicore, SDD, CMDB, TPM DCM, ...
 - Grid: models similar but different focus...higher level, user friendly, simpler, capabilities, requirements, more abstract
- But Grid modeled information can and should build on and re-use System/Network information...
 - For example, to route workload successfully, you don't need to know/specify lots of requirements

OGSA Information Model Concept

Capabilities and Requirements have same language (name-value pairs) with ability to express range values for easy mapping and selection



- Instances info, e.g. CIM, GLUE
- Granular
- Expresses managed information in the environment
- May be virtualized

- Abstracted and generated from managed information via algorithms
- Simple, meaningful, and user friendly
- Expresses the capabilities of an entity, e.g. container
- May be virtualized
- May be refreshed as managed information changes

- Simple, meaningful, and user friendly
- Expresses an activity's needs
- May be virtualized

Moving from Concept to Concrete...



- ❑ *Relationship of EMS model to systems management (“detailed”) model*
- ❑ *Advertisement/Requirement (ClassAd-like) model*
- ❑ *Representation of resource capabilities*
- ❑ *Representation of requirements specification*
- ❑ *Basic set of resource capabilities and properties from which extend*

OGSA Information Model Relationship

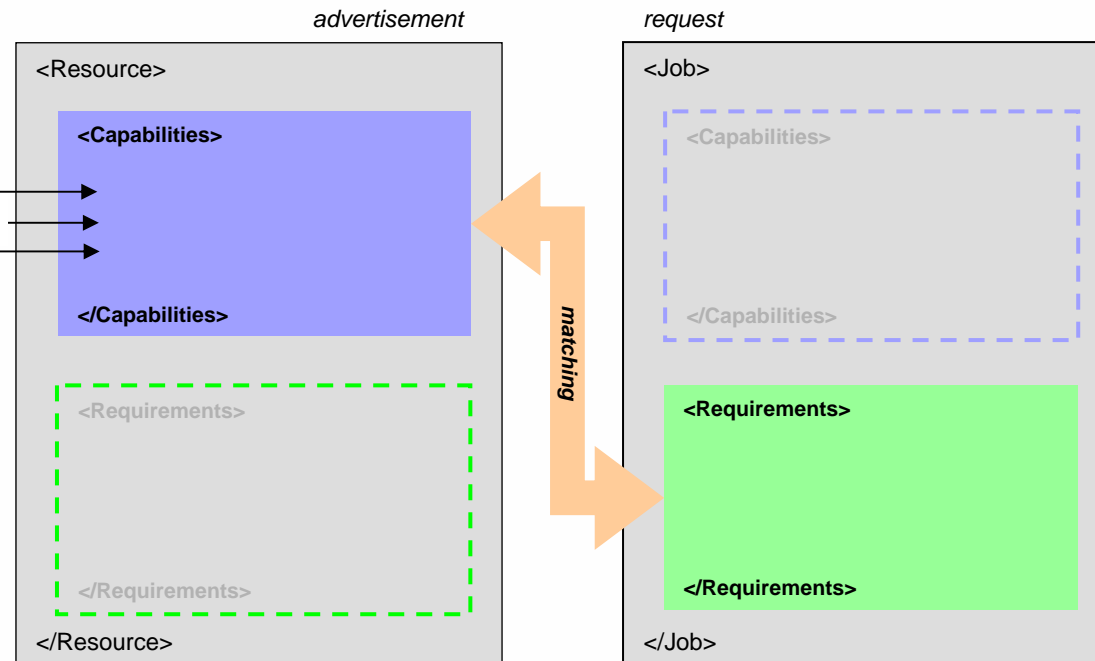
Various "detailed" information models including Glue, CIM, SDD/IUDD, SDM and others



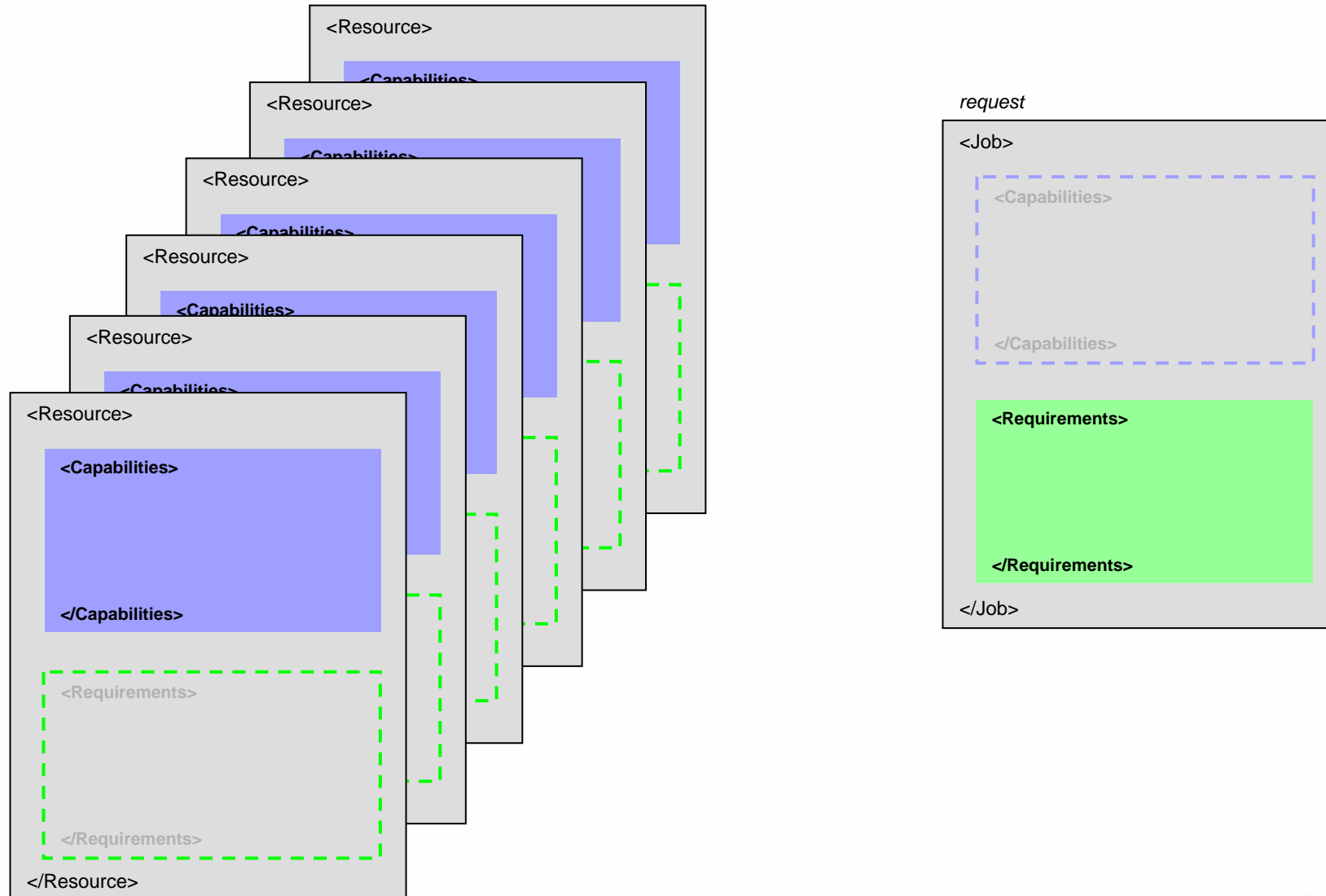
Monitoring Instrumentation



Actual devices and components

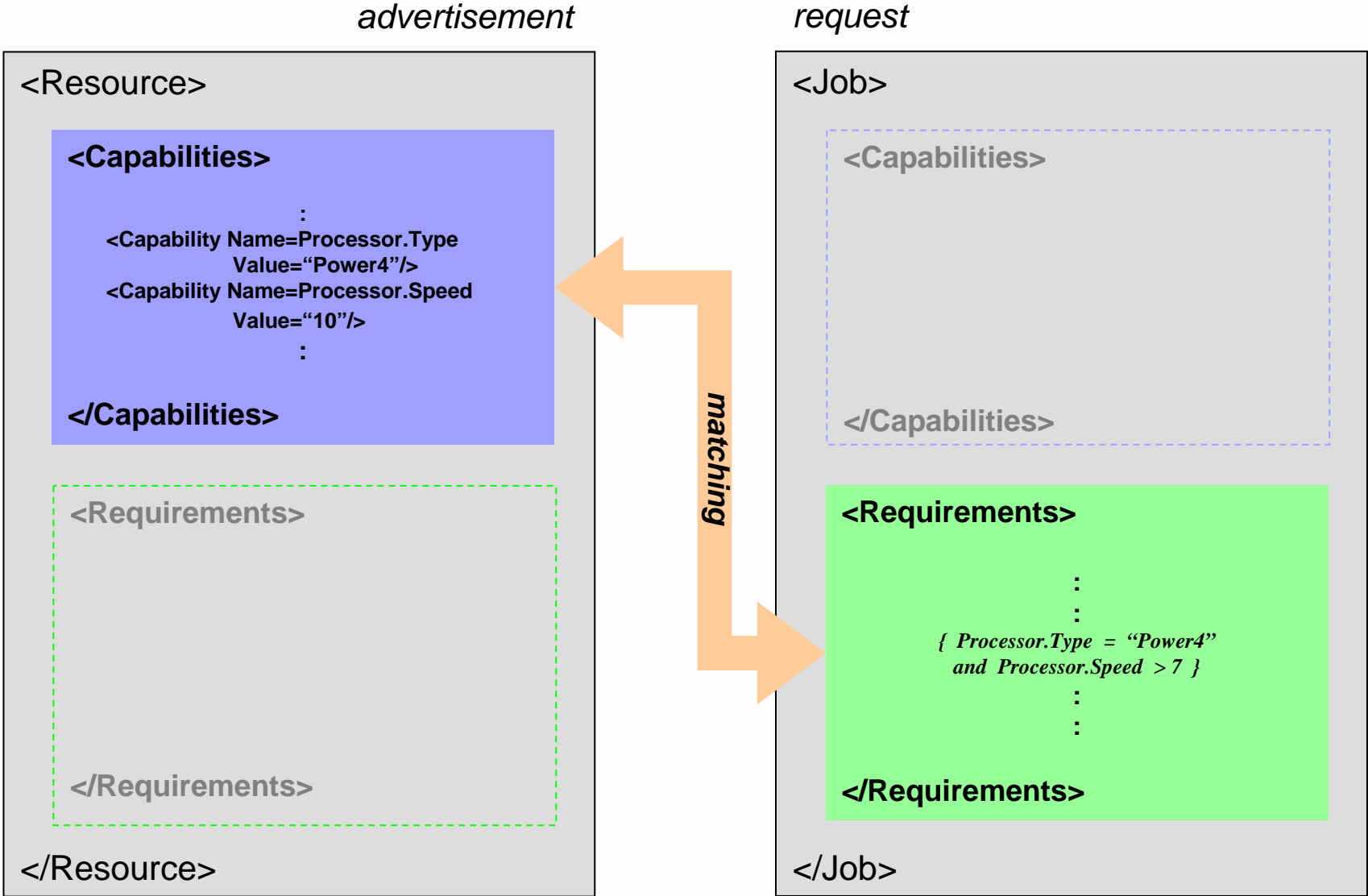


Advertisement/Requirement Matching



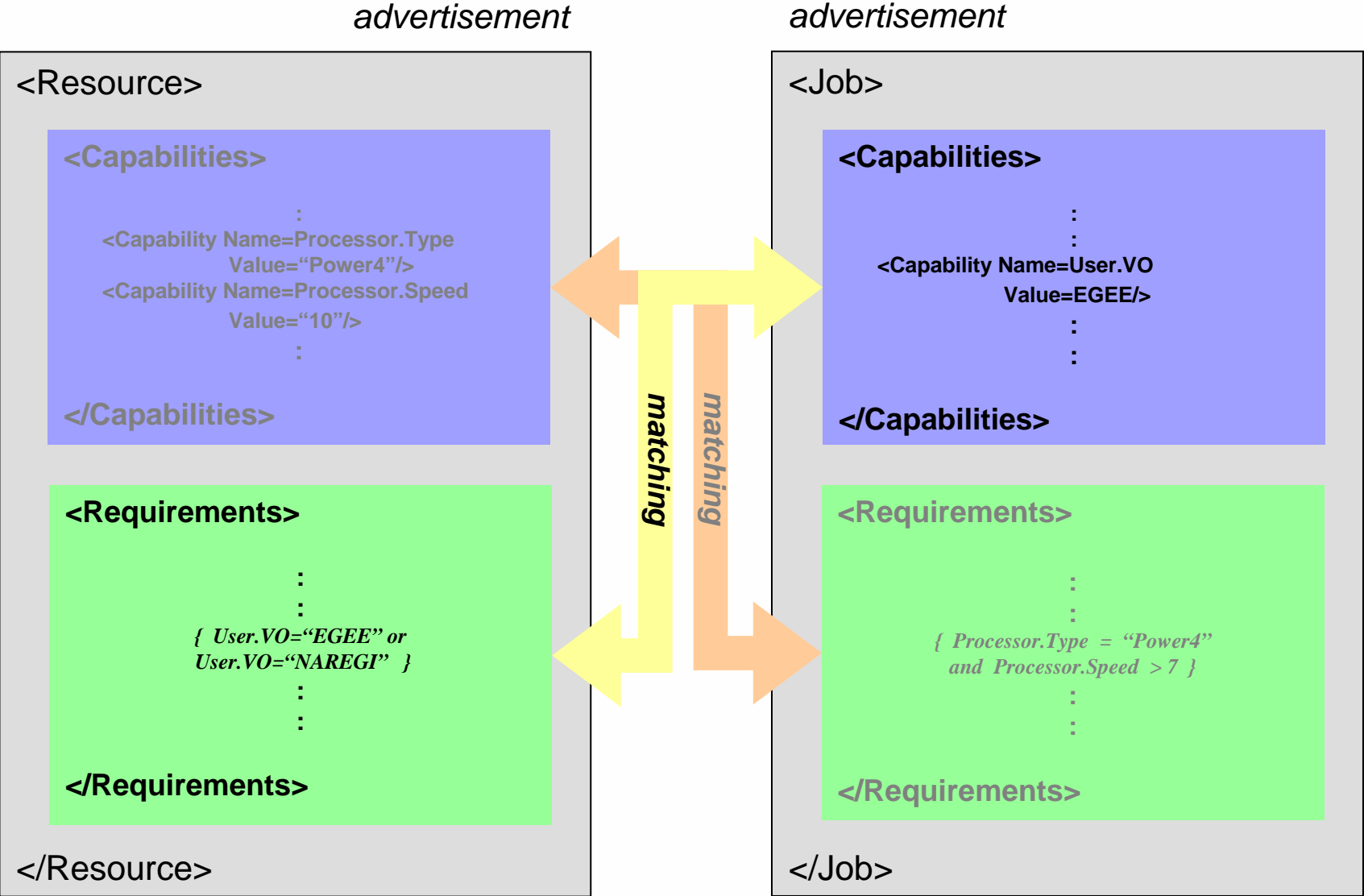
Advertisement/Requirement Matching

document syntax is exemplary & not rigorous on these concept charts 



Advertisement/Requirement Matching

document syntax is exemplary & not rigorous on these concept charts



Advertisement/Requirement Rendering



- **Proposal:** Use XML Document as Capability Specification
 - Declarative format
 - Name-value (capabilities-property) pair “basic model”
 - Limited (2-level) hierarchy
 - Can be expressed as XML document and either stored natively or in a relational table.
 - Concrete rendering uses XSD schema and XML typing

XML Rendering of Resource Capabilities



Rendering: "Concrete"

```
<Resource Name="BESNode3">
  <Capabilities>
    <Processor>
      <Architecture>Power4</Architecture>
      <Speed>10</Speed>
    </Processor>
    <OS>
      <OSType>Linux</OSType>
      <PhysicalMemory>3000000</PhysicalMemory>
      <MaxProcessesPerUser>32</MaxProcessesPerUser>
    </OS>
    <SoftwareLicense>
      <SAP/>
    </SoftwareLicense>
  </Capabilities>
</Resource>
```

- Capitalize on tooling for generation and validation
- XML typing used; reuse of element names requires use of namespaces
- XSD can be used but must be open for extensibility, i.e. <any>
- Extensions can be derived from existing models
- Existing XML models can be included

| Resource | Capability | Property | Value |
|----------|-----------------|---------------------|----------|
| BESNode3 | Processor | Architecture | Power4 |
| BESNode3 | Processor | Speed | 10 |
| BESNode3 | OS | OSType | Linux |
| BESNode3 | OS | PhysicalMemory | 3000000 |
| BESNode3 | OS | MaxProcessesPerUser | 32 |
| BESNode3 | SoftwareLicense | SAP | 1 (True) |

Advertisement/Requirement Rendering



- **Proposal:** Use XQuery (in some form) as a requirements specification grammar.
 - XQuery Status:
 - Based on XPath 2.0
 - Spec is a Candidate Recommendation; implementation test suite exists
 - “Candidate Set Generation” (CSG) becomes an XQuery over all the “advertised” capabilities/requirements between execution endpoints and a job.
 - XQuery language is a fairly complete “algebra” with many comparison, arithmetic, operators.
 - *XQuery is really an XML programming language including FLWOR [for, let, while, order-by, return] blocks, and IF/THEN logic etc.)*
 - *Suggest development of a profile establish reasonable use subset (spec already lists optional parts...can/should we limit further?)*
 - User defined functions and prologues can be exploited to make XQuery “requirements” specifications simpler.
 - Implementation of “resource repository” may either be native XML document store or data base (*ODBC/JDBC*)

Full Copyright Notice



Copyright (C) Open Grid Forum (2005, 2006). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.