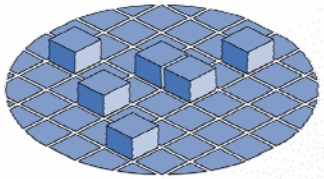
FUJITSU THE POSSIBILITIES ARE INFINITE



UniGrids

OGSA, WSRF, and the Foundations of Grid

Dr. David Snelling
Fujitsu Laboratories of Europe
David.Snelling@ UK.Fujitsu.com

Advert!

- **This is UniGrids**
- **Based on Unicore**
  - A complete Grid environment started in 1997
- **Unicore has been a major contributor to**
  - OGSA
    - And components: BES, ByteIO, Naming, RSS, JSDL,
  - OASIS:
    - WSRF
    - WSN
    - WSDM

UniGrids

What do Grids do?

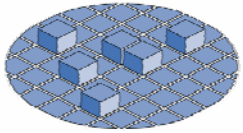
- **Resource Sharing**
  - Compute and data
- **Collaboration**
- **Security**
  - Delegation
  - Authentication
- **TCO Reduction for throughput systems**
- **More Categories**
  - More Services

UniGrids

What do Grids do?

FUJITSU

- **Security**
  - Secure connections
  - Authorization control
  - Delegation
- **Virtual Organizations**
  - Shared goals
  - Authorization Sharing
  - Resource Sharing
- **Data**
  - Transport
  - Virtualization
  - Federation
  - Replica Management
  - Streaming Data
- **Execution**
  - Jobs
  - Services
  - Scheduling
- **Service Composition**
  - Workflow
  - Subcontracting
- **Discovery**
  - Services
  - Data Sets
  - Resources
  - Registration

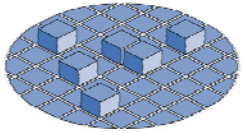


UniGrids

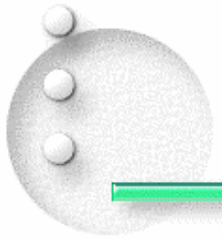
Is OGSA up to it?

What is OGSA?

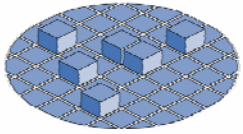
- **Open Grid Services Architecture**
- **Standard for building Grids.**
  - Assumes some things about what a Grid is.
  - Proposes a process for making Grids standard.
- **The “Service” in OGSA**
  - OGSA is based on Service Oriented Architecture manifested in Web Services.
- **The “Open” in OGSA**
  - The process by which the architecture is defined is open to all and transparent.
  - This means Standards Development Organizations
    - GGF, OASIS, W3C, IETF, ...



UniGrids

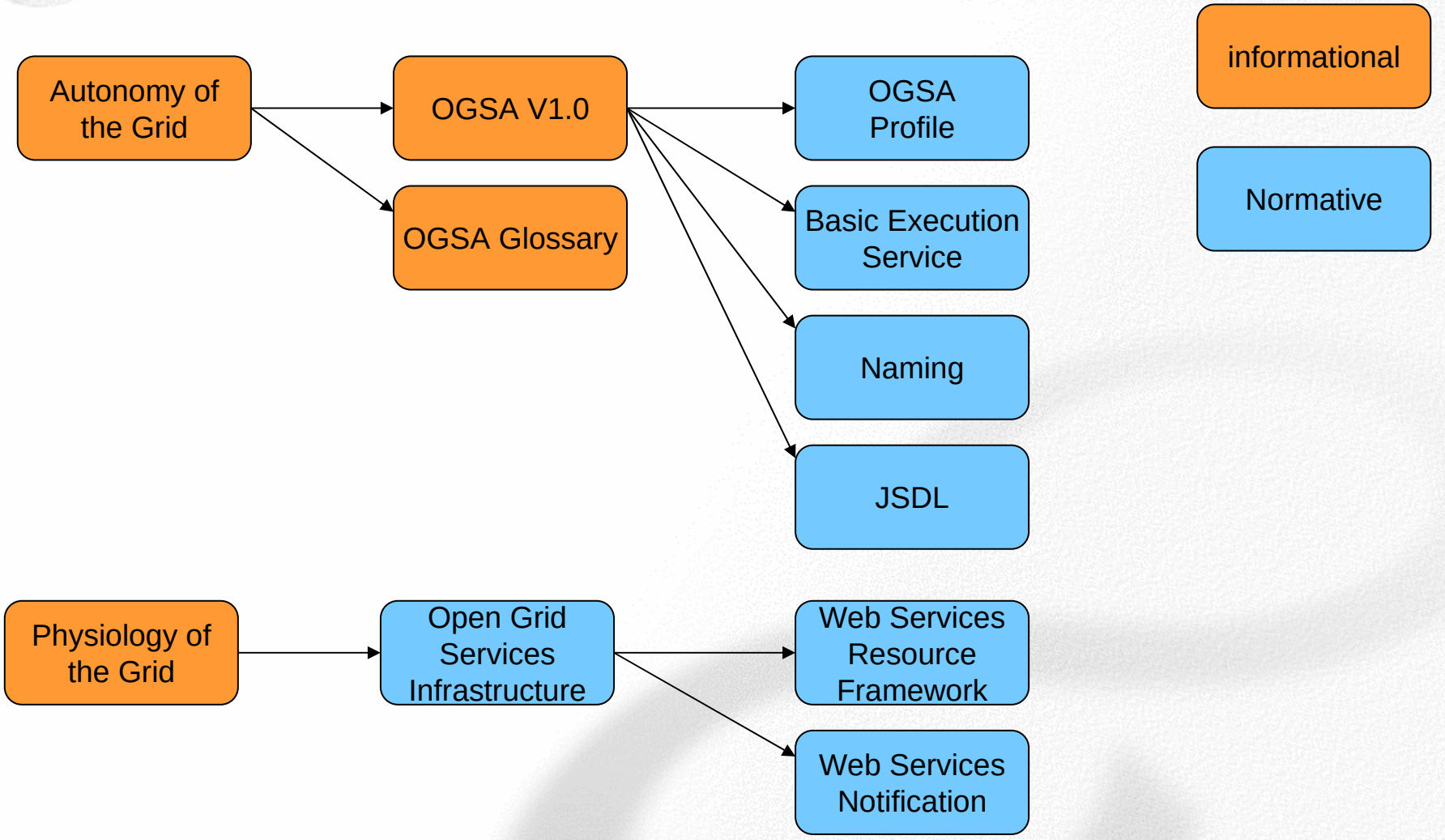


FUJITSU



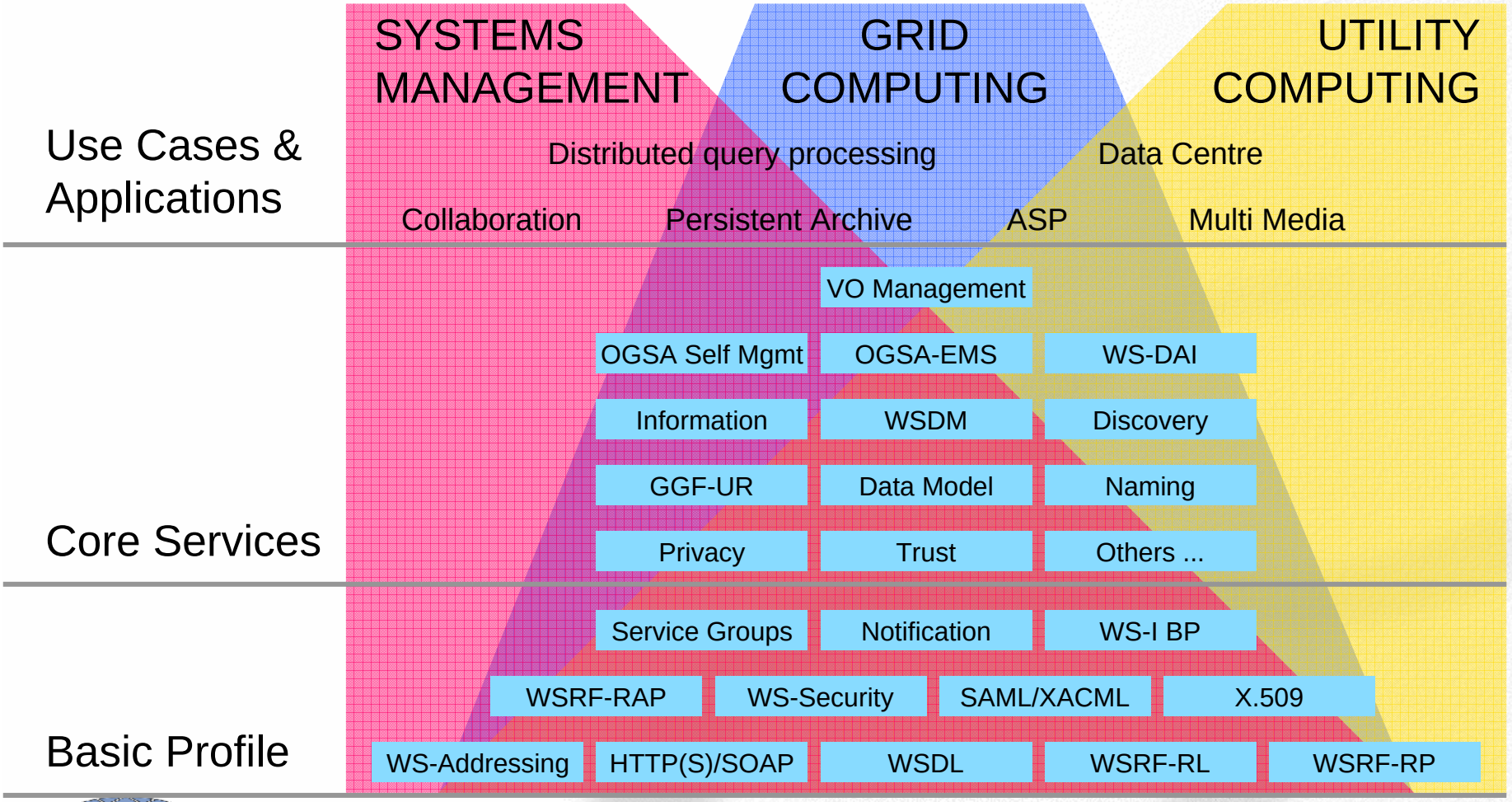
UniGrids

OGSA Evolution



UniGrids

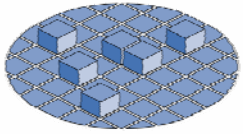
OGSA Specifications Landscape



UniGrids

Basic WSRF/WSN Concepts

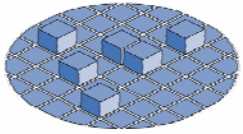
- State as a Grid Fundamental
- WSRF 101
- Basic Properties
- Lifetime
- Notification



UniGrids

Credits

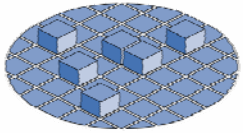
- Most of the material is the result of collaborative effort from:
- Steve Graham, Jeff Frey, Jay Unger, Steve Tuecke and Tom Maguire.



UniGrids

State: A Grid Fundamental

- **Stateful Entities Exist**
  - Data in a purchase order
  - Schema for a data base or structured data file
  - Current usage agreement for resources
  - Metrics associated with work load on a server
- **Hitherto: No WS Standards for State Management**
  - Each system does it in an “idiosyncratic way”
  - Integration impediment
- **OGSA Needs to**
  - Formalize a mechanism to represent “state”
  - WSRF specifications provide this



UniGrids

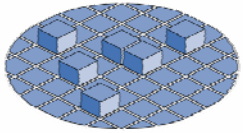
Reconciling State and SOA

- **Web Services:**
  - Operation execution available at an endpoint address
    - Service defined in terms of the operations it implements
  - An operation is defined in terms of a message exchange
    - WSDL, SOAP, and all that...
  - Accessible using WS-Addressing Endpoint Reference
  - Lifecycle of a Web service is described in terms of deployment
    - It is there or it isn't. No factory pattern, only discovery.
- **The Web Service itself is typically Stateless**
- **Grids Need Access to Stateful Resources**

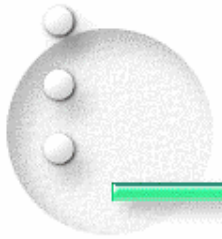
UniGrids

Factory Pattern

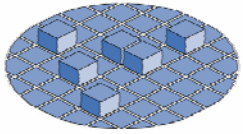
- **When you need a Resource you need it**
  - For example:
    - Start a Job to do this...
    - Create a result data set combining these...
    - Get me management interface for this experiment...
  - OGSA provides for dynamic creation of resources.
- **Find it or Create it?**
  - The difference is moot:
    - FindMeAThing (Description) -> HandleToThing
    - MakeMeAThing (Description) -> HandleToThing



UniGrids



FUJITSU

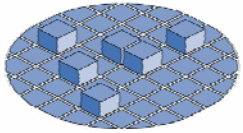


UniGrids

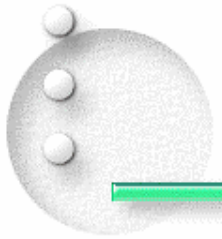
WSRF 101: Stateful Resource

FUJITSU

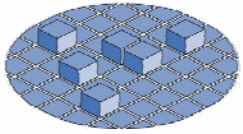
- **A Resource:**
  - A specific set of state data expressible as an XML document
    - This is not typically all of the resource's state!
  - Has a well-defined identity and lifecycle
  - Known to, and acted upon, by one or more Web services.
- **Many Possible Instances**
  - Files, Database tables, EJB Entities, XML documents, Compositions of multiple data sources, Virtualized executions of applications, etc.
- **A WS-Resource has:**
  - Identity: Can be uniquely identified/referenced
  - Lifetime: Often created & destroyed by clients
  - State: Part of the state can be projected as XML
  - Type: Its Web service interface



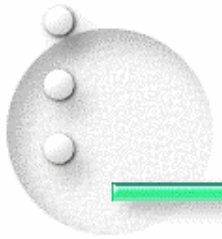
UniGrids



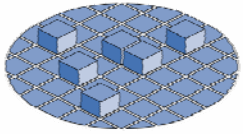
FUJITSU



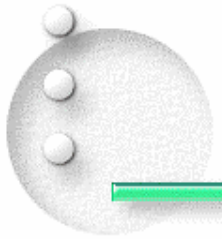
UniGrids



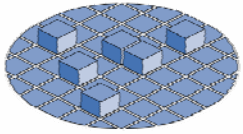
FUJITSU



UniGrids



FUJITSU



UniGrids

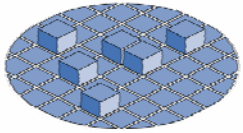
Advert!

QuickTime™ and a
GIF decompressor
are needed to see this picture.

UniGrids

Resource Properties

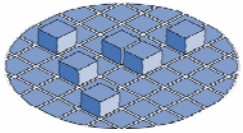
- **The Specification Defines How to:**
  - Use XML to model elements of resource state
  - Associate state with a WSDL portType
  - Use standard operations for getting, setting, querying,
  - Use standard mechanisms to subscribe to state changes
- **Why:**
  - Basis for standard resource inspection, monitoring, and state management
- **Intuition:**
  - Think of Resource Properties as an XML projection of the actual state of the resource



UniGrids

Modeling State in XML

- **A WS-Resource's "Resource Properties Document"**
  - Elements of state are modeled as child elements of the root
  - <job:JobProperties>
    <job:JobName>Daves Job<job:JobName>
    <job:executionState>Submitted</job:executionState>
    <job:handle> 2824 </job:handle>
    <wsrl:CurrentTime> 23 Apr 2004 19:45:29 GMT </ … >
    <wsrl:TerminationTime xsi:nil="true" />
    </job:JobProperties>
- **PortType Attribute Declares the RPD**
  - <wsdl:portType …
    wsrf-rp:ResourceProperties="xsd:QName"? …> …
- **XSD:ref is used to "combine" Resource Properties from various definitions**
  - Each element references by QName.

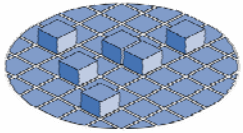


UniGrids

Resource Property Operations

FUJITSU

- **GetResourceProperty**
  - Simple single resource property element getter
  - May return multiple instances of the named RP.
  - <wsrf-rp:GetResourceProperty>
    job:handle
    </wsrf-rp:GetResourceProperty>
  - <wsrf-rp:GetResourcePropertyResponse>
    <job:handle> 1577 </job:handle>
    </wsrf-rp:GetResourceProperty>



UniGrids

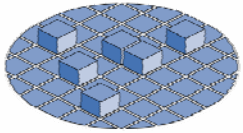
Resource Property Operations

- **GetMultipleResourceProperties**
  - More sophisticated multiple property value retrieval
  - ```
    <wsrf-rp:GetMultipleResourceProperties>
      <wsrf-rp:ResourceProperty>job:handle </job:handle>
      <wsrf-rp:ResourceProperty>job:executionState </ ...>
      <wsrf-rp:ResourceProperty>job:JobName </job:JobName>
    </wsrf-rp:GetMultipleResourceProperties>
    ```
  - ```
    <wsrf-rp:GetMultipleResourcePropertiesResponse>
      <job:handle> 2824 </job:handle>
      <job:executionState> Submitted </job:executionState>
      <job:JobName> xclock </job:JobName>
    </wsrf-rp:GetMultipleResourcePropertiesResponse>
    ```

UniGrids

Resource Property Operations

- **QueryResourceProperties**
  - Execute an expression against the resource properties document
  - ```
    <wsrf-rp:QueryResourceProperties>
      <wsrf-rp:QueryExpression Dialect="URI">
       xsd:any
      </wsrf-rp:QueryExpression>
    </wsrf-rp:QueryResourceProperties>
    ```
- **QueryExpression defines dialect by URI**
  - XPath 1.0, 2.0
  - XQuery
  - SQL
  - SPARQL



UniGrids
    ```

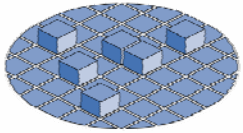
Set Resource Properties

- **Three modes can be applied many times in any order in a single message.**
  - Insert: Add a new resource property
  - Update: Replace all properties with a given name.
  - Delete: Remove all properties with a given name.
- **However they must appear to happen in order.**
- **The final version of the RPD must validate.**
- **Faults:**
  - On partial completion, final state of RPD is implementation dependent.
  - Failure to validate final RPD.
- **There are also individual Insert, Update, and Delete operations**

UniGrids

Whole Document Operations

- **GetResourcePropertyDocument**
  - Get the entire resource property document
- **PutResourcePropertyDocument**
  - Replace the entire RPD with a new one.
  - The semantics are service specific, and very loose
  - If the resulting document is different than the provided document, the resulting document is returned.
- **Virtually Identical Semantics to WS-Transfer**
  - The document type in WS-Transfer less restricted.



UniGrids

Lifetime

- **Defines:**
  - Immediate, synchronous destruction operation
  - Time-based, scheduled destruction operation
  - “Soft-state” or “leased” lifetime management
  - Termination time not required to increase monotonically
- **Resource properties:**
  - CurrentTime: Can be used to determine clock skew
  - TerminationTime: Current scheduled termination time
- **Notification of resource termination**
- **Why:**
  - Define clear means by which resources can be destroyed
  - Allow the Grid to “Garbage Collect” itself automatically

Lifetime Operations

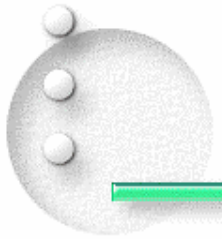
FUJITSU

- **Immediate Destruction**
  - <wsrf-rl:Destroy/>
- **Scheduled Termination**
  - <wsrf-rl:SetTerminationTime>
    [
    <wsrf-rl:RequestedTerminationTime xsi:nil=”xsd:boolean”?>
    xsd:dateTime
    </wsrf-rl:RequestedTerminationTime>
    ] | [
    <wsrf-rl:RequestedLifetimeDuration>
    xsd:duration
    </wsrf-rl:RequestedLifetimeDuration>
    ]
    </wsrf-rl:SetTerminationTime>

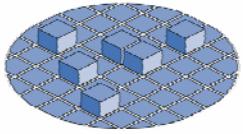
UniGrids

Notification

- **WS-Notification**
  - Brings publish and subscribe messaging to Web services
  - Loosely coupled, asynchronous messaging in WSs
  - WS Notification composes with WSRF and other WSs
- **Brokered notification**
  - Support for intermediates, queuing, aggregation, distribution, filtering, ...
- **Topics and Topic Spaces**
  - Define a mechanism to advertise topics for subscription.
- **Use of WS-Notification in WS-RF**
  - Receive notification changes to the Resource Properties



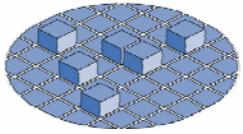
FUJITSU



UniGrids

Advanced Topics

- **Advanced Resource Properties**
  - Dynamic
  - Coherence issues and relationship to service operations
- **Service Groups**
- **Base Faults**

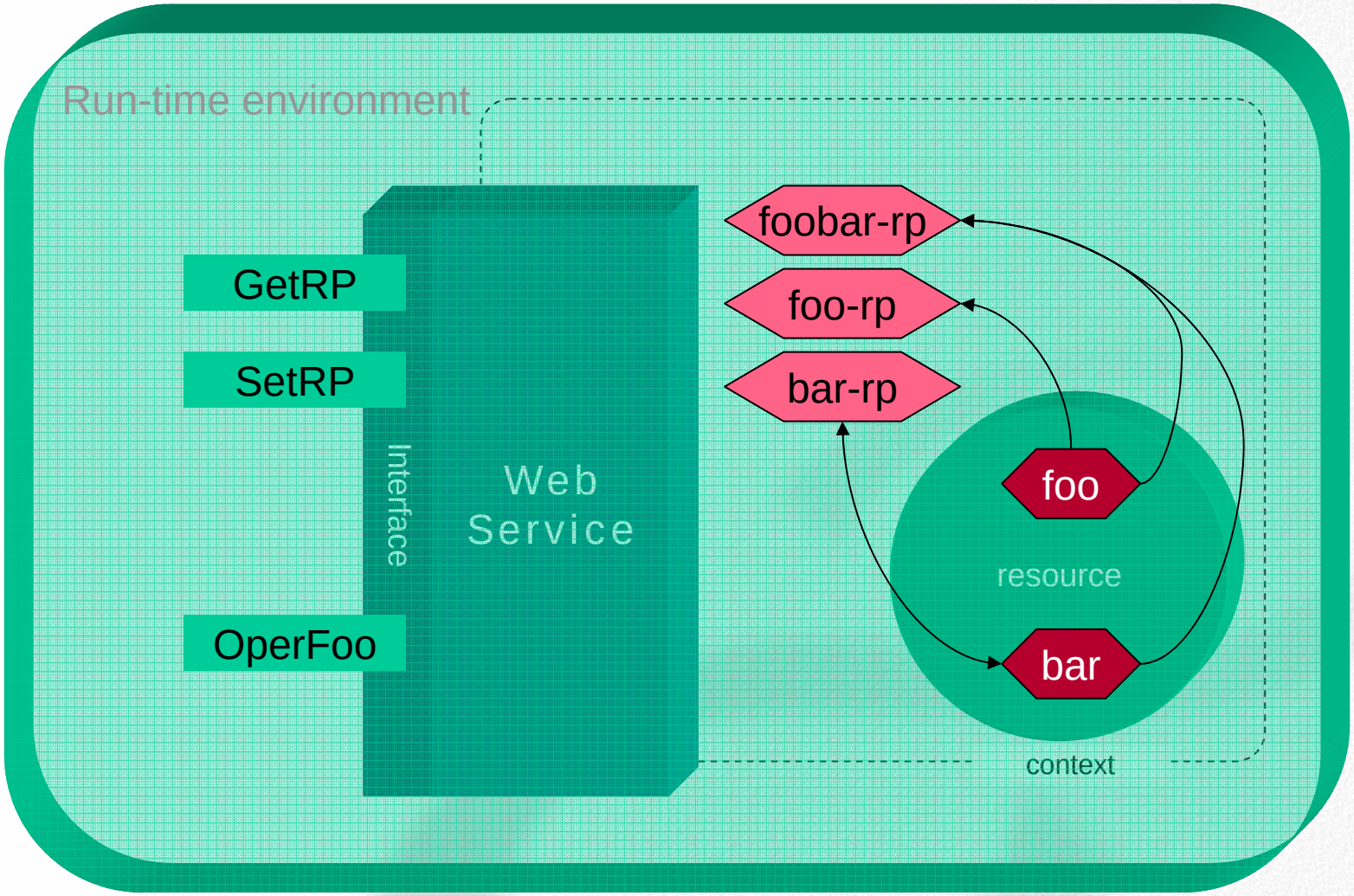


UniGrids

Dynamic Resource Properties

- **Recall that Resource Properties are**
  - An XML representation of a resource's state
  - Should be thought of as a projection of actual state
- **Dynamic Properties**
  - May not actually exist until their value is requested
  - For example:
    - Queue Length on a Batch Subsystem
    - Current Time on any resource
  - These RPs' XMP representations can be created on the fly
- **This has interesting implications for the Query Operations and for the Resource Property Document**

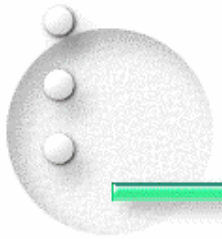
Coherency



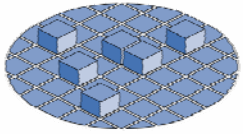
UniGrids

Service Groups

- **Defines:**
  - Web service interfaces for representing collections of Web services or WS-Resources, referenced by EPRs
  - Each SG entry includes member EPR + associated content
    - WSRF-RP used for representing the entries
    - Members may be homogenous or heterogeneous
    - Can have rules constraining membership and content
  - Has a registration interface for adding entries
  - WSRF-RL used for removing entries
- **Why:**
  - Myriad of reasons for groups: E.g. Registries, collective operations, federated services, etc.



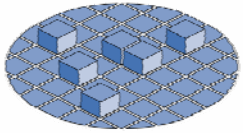
FUJITSU



UniGrids

Model Continued

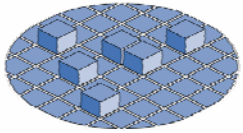
- **ServiceGroup**
  - A collection of Web services and the information that pertains to them.
  - Purpose is application domain specific
- **ServiceGroupEntry**
  - A WS-Resource that models the membership relationship
  - Includes associated content
- **ServiceGroupRegistration**
  - Message exchange for adding new members to s SG
  - Membership may occur in other ways
- **Member**
  - Web service, WS-Resource, Something identified by an EPR



UniGrids

Service Group Characteristics

- **ServiceGroup Destruction**
  - ServiceGroupEntries should also be destroyed, i.e. the membership relationship id destroyed.
  - However, the members are not affected.
- **Membership Largely Unrestricted**
  - A member may belong to several ServiceGroups.
  - A member may belong to the same ServiceGroup more then once.
  - The member of a ServiceGroup may implement message exchanges from various interfaces.

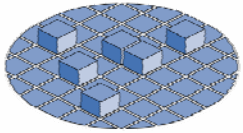


UniGrids

Resource Properties Relationship

FUJITSU

- **MembershipContentRule**
  - These elements specify the constraints on membership of the service group
    - For example, the supported interfaces
  - Mandates the presence of specified Resource Properties in the Entry's Resource Property Document
- **Entry Resource Properties**
  - These elements may represent a projection of the Resource Properties of Member WS-Resources
  - Other content is also permitted.



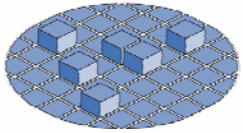
UniGrids

Base Faults

- **Defines**
  - Set of “common” properties of a fault
  - Convention for specializing “common” fault
  - How “common” fault type is used in WSDL
- **Why?**
  - Increases the likelihood that service requestors to automatically (without human intervention) understand and/or adapt to faults requires interface designers to define rich, structured fault messages
  - Standard fault messages encourage tooling that can assist interface designers, service implementers, and client implementers

Mini Workshop

- **Each Group Takes a Topic**
- **Design How to Implement the Service**
  - The only operation you can add is a ‘create’ operation
  - All others must be from WS-Addressing, WSRF, or WSN.
- **See how sophisticated you can make the service.**
- **Comment on your design**
  - Is this really such a good idea?
  - What are possible problems with it?
  - What constraints could you add to the environment to make it better?
  - Do standards play and further role in this process? What?
  - .....etc?



UniGrids

Mini Workshops

- **Group 1: Job Submit Service**
  - Given a JSDL Document for a job, execute the job on a resource
  - Obtain a reference (EPR) to the job
  - Kill the job facility and Maximum runtime facility
  - Send a message when the job finished.
  - Obtain output from the job.
- **Group 2: Registry**
  - Advertise
    - Services supporting a number of interfaces
    - Services have standard properties for load and policy
    - Services are public or “members only” services
  - Obtain references to the services
  - Obtain the availability of the services (directly and indirectly)
  - Discuss mechanisms for keeping the registry up to date.

UniGrids

Mini Workshops

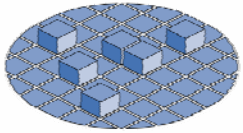
- **Group 3: Data Streaming Service**
  - A facility that allow a service to provide a continuous supply of data to a client
  - Data should not be provided more than once.
  - Status information should be provided.
  - Metadata about the stream should be provided.
  - Can the data be made available to clients behind the firewall.
- **Group 4: A VO Manager**
  - Create a new VO
  - Add and remove members.
  - Members join for a set time and extend their membership if necessary
  - There must be advertised and enforced membership rules

UniGrids

Mini Workshops

FUJITSU

- **Group 5: Virtualized Data Source**
  - Provide access to the data from an experiment or the results of some computation
  - The schema/format of the data source must be discoverable
  - The data is always available once created, until the service is no longer needed.
- **Group 6: Identity Mapping Service**
  - All users have an X509 Certificate
  - They may also have several other identities in various formats. The service maps to one of these
  - A facility to advertise what mappings are possible.
  - A facility to add a new mapping for a user.
  - Discuss the security implications, if this is a remote service.

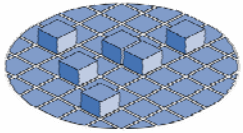


UniGrids

Mini Workshops

FUJITSU

- **Group 7: Third Party File Transfer Service**
  - Given two URI's for two files
  - A service that will transfer the source to the destination.
  - Provide information on the status of the transfer
  - The service should have an option to tell the user when it is finished.
  - Cancel the transfer or give it longer to finish.
- **Group 8: Application Steering**
  - Assume an application is running and supports WSRF
  - How would you expose a steering interface?
  - Design a few simple steering controls.
  - You need a facility to obtain data from the application and provide for a feedback mechanism.



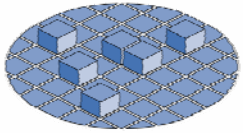
UniGrids

Mini Workshops

FUJITSU

Group 9: Authorization Service

- Given a user identity (X509 cert), the EPR for a service, and maximum compute time.
- Provide a service to authorize users for particular requests
- Registration and removal of authorized users is also needed
- Discuss the security implications, if this is a remote service.
- [Extra Credit: Include accounting based authorization, e.g. “No you can’t run it. You spent all your money already.”]



UniGrids