Future of Grid and Globus™

Steve Tuecke (tuecke@mcs.anl.gov)
Globus Project™

CERN

August 26, 2003

the globus project™
www.globus.org

The Globus Project™

Making Grid computing a reality

- Close collaboration with real Grid projects in science and industry
- Development and promotion of standard Grid protocols (e.g. OGSA) to enable interoperability and shared infrastructure
- Development and promotion of standard Grid software APIs and SDKs to enable portability and code sharing
- The Globus Toolkit™: Open source, reference software base for building Grid infrastructure and applications
- Global Grid Forum: Development of standard protocols and APIs for Grid computing

GlobusWORLD™

- Annual Globus conference
- Next event:
  - January 20-23, 2004
  - San Francisco, California USA
- Open Call For Participation
  - Abstract submissions for talks, panels, and workshops
- Early bird registration through Sept 30
- www.globusworld.org

Overview

- Where we are today
- Standards landscape & Globus Toolkit plans
- Transitioning from GT2 to GT3

Overview

- Where we are today
  - What is a Grid?
  - OGSI
  - Globus Toolkit® v3.0
- Standards landscape & Globus Toolkit plans
- Transitioning from GT2 to GT3

Is the Grid …

a) A collaboration & resource sharing infrastructure for scientific applications?

b) A standards-based distributed service integration & management technology?

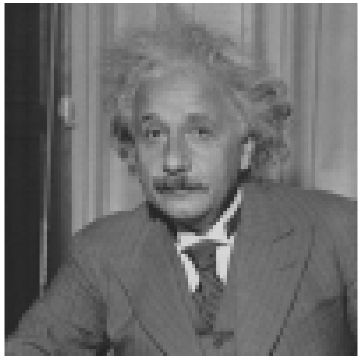
c) A disruptive technology that enables a virtualized, collaborative, distributed world?

d) An open source technology & community?

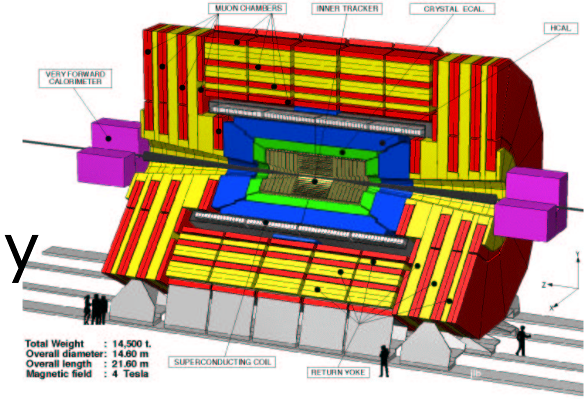
e) An over-used marketing slogan?

f) All of the above?

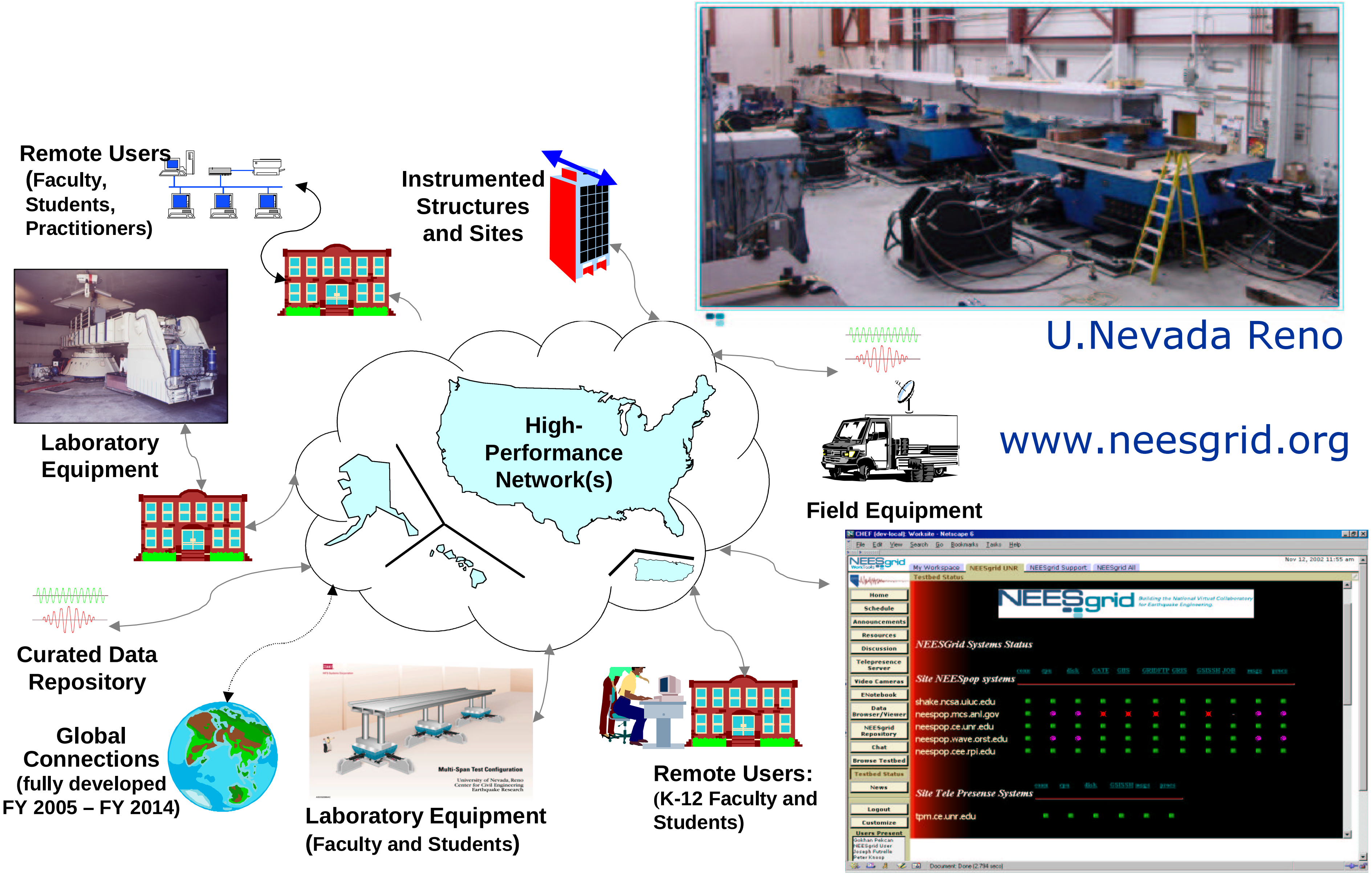
Why the Grid?
Origins: Revolution in Science



- Pre-Internet
  - Theorize &/or experiment, alone or in small teams; publish paper
- Post-Internet
  - Construct and mine large databases of observational or simulation data
  - Develop simulations & analyses
  - Access specialized devices remotely
  - Exchange information within distributed multidisciplinary teams



NEESgrid Earthquake Engineering Collaboratory



Remote Users (Faculty, Students, Practitioners)

Instrumented Structures and Sites

U.Nevada Reno

Laboratory Equipment

High-Performance Network(s)

www.neesgrid.org

Field Equipment

Curated Data Repository

Global Connections (fully developed FY 2005 – FY 2014)

Laboratory Equipment (Faculty and Students)

Remote Users: (K-12 Faculty and Students)

Multi-Span Test Configuration
University of Nevada, Reno
Center for Civil Engineering
Earthquake Research

CHEF (dev-local): Worksite - Netscape 6

Nov 12, 2002 11:55 am

Testbed Status

NEESgrid Building the National Virtual Collaboratory for Earthquake Engineering.

NEESGrid Systems Status

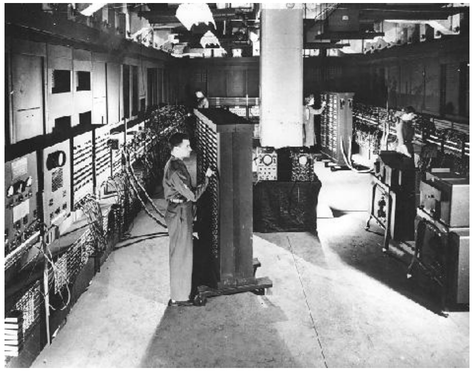
Site NEESpop systems	conn	cpu	disk	GATE	GIIS	GRIDFTP	GRIS	GSISSH	JOB	msgs	procs
shake.ncsa.uiuc.edu	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
neespop.mcs.anl.gov	🟩	●	●	✗	✗	✗	🟩	✗	-	●	●
neespop.ce.unr.edu	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
neespop.wave.orst.edu	🟩	●	●	🟩	🟩	🟩	🟩	🟩	🟩	●	●
neespop.cee.rpi.edu	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Site Tele Presense Systems	conn	cpu	disk	GSISSH	msgs	procs
tpm.ce.unr.edu	🟩	🟩	🟩	🟩	🟩	🟩

Users Present
Gokhan Pekcan
NEESgrid User
Joseph Futrelle
Peter Knoop

Document: Done (2.794 secs)

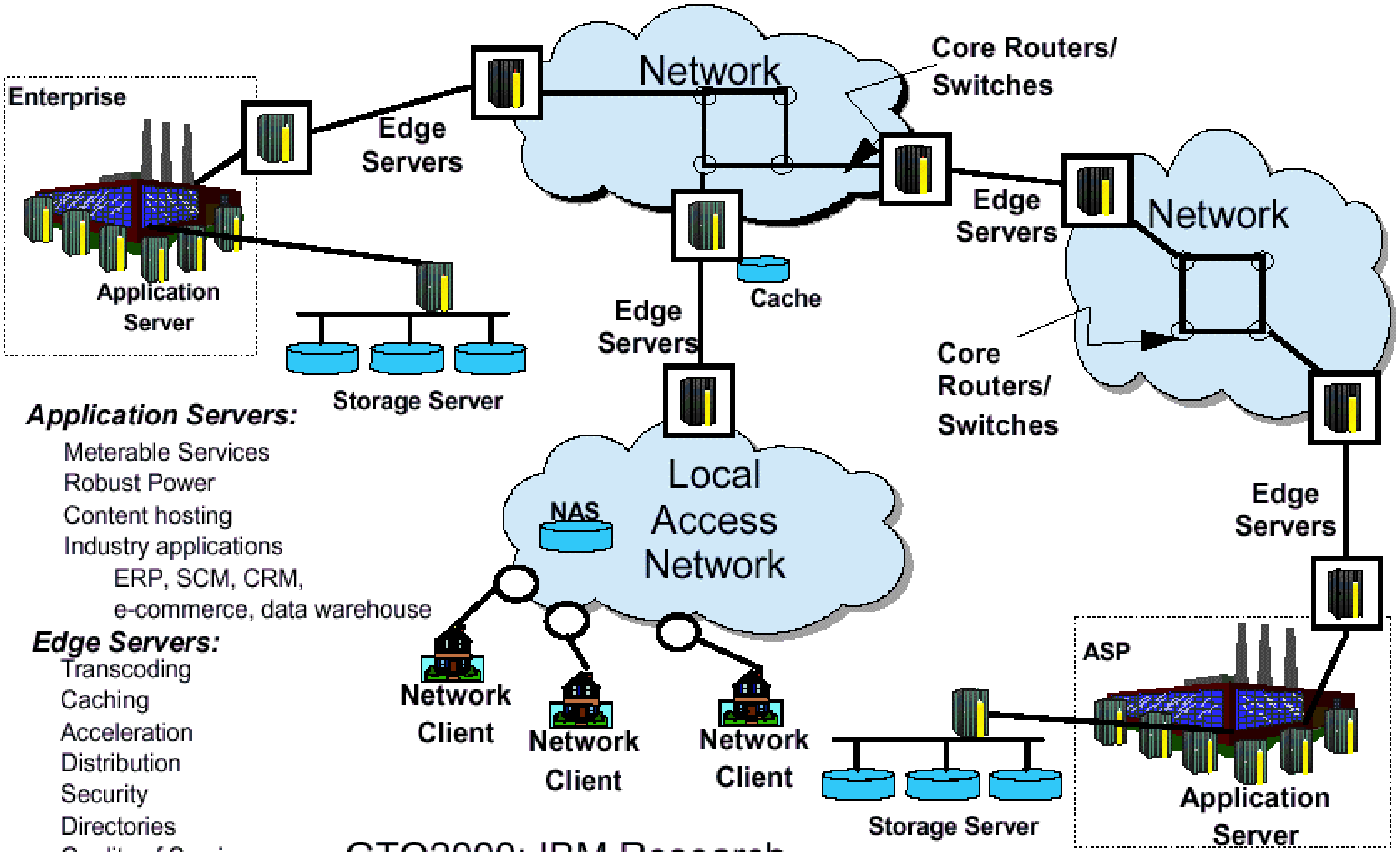
Why the Grid? New Driver: Revolution in Business



- Pre-Internet
  - Central data processing facility
- Post-Internet
  - Enterprise computing is highly distributed, heterogeneous, inter-enterprise (B2B)
  - Business processes increasingly computing- & data-rich
  - Outsourcing becomes feasible => service providers of various sorts
  - Growing complexity & need for more efficient management

Current Installations

The New Enterprise Computing Environment



Application Servers:

- Meterable Services
- Robust Power
- Content hosting
- Industry applications
  - ERP, SCM, CRM,
  - e-commerce, data warehouse

Edge Servers:

- Transcoding
- Caching
- Acceleration
- Distribution
- Security
- Directories
- Quality of Service

GTO2000: IBM Research

Common eScience/eBusiness Requirements

- Dynamically link resources/services
  - From collaborators, customers, eUtilities, … (members of evolving “virtual organization”)
- Into a “virtual computing system”
  - Dynamic, multi-faceted system spanning institutions and industries
  - Configured to meet instantaneous needs, for:
- Multi-faceted QoX for demanding workloads
  - Security, performance, reliability, …

What is a Grid?

- We believe there are three key criteria:
  - Coordinates resources that are not subject to centralized control …
  - using standard, open, general-purpose protocols and interfaces …
  - to deliver non-trivial qualities of service.
- What is not a Grid?
  - A cluster, a network attached storage device, a scientific instrument, a network, etc.
  - Each is an important component of a Grid, but by itself does not constitute a Grid

Why Now?

- Moore’s law improvements in computing produce highly functional end-systems
- The Internet and burgeoning wired and wireless provide universal connectivity
- Changing modes of working and problem solving emphasize teamwork, computation
- Network exponentials produce dramatic changes in geometry and geography

Open Grid Service Infrastructure (OGSI) Version 1.0

- GGF “proposed recommendation” (GFD-R.15)
  - ◆ Equivalent to IETF draft standard (RFC)
- Editors:
  - ◆ Tuecke (ANL), Czajkowski (USC/ISI), Foster (ANL), Frey (IBM), Graham (IBM), Kesselman (USC/ISI), Maguire (IBM), Sandholm (ANL), Snelling (Fujitsu Labs), Vanderbilt (NASA Ames)
- Contributors:
  - ◆ Butler, Ferguson, Grimshaw, Finkelstein, Leymann, Nally, Nick, Rofrano, Stokes, Storey, Unger, Weerawarana

OGSI History

- Jun 2001: Steve Tuecke (Globus) wrote initial internal draft OGSI specification
- Sep 2001: IBM joined effort, and substantially ramped up the pace
- Feb 2002: Globus & IBM introduced draft OGSI specification at GGF4, proposed wg
- Mar 2002: Globus OGSI Tech Preview v1
- Sep 2002, Nov 2002, Jan 2003: Meetings
- Apr 2003: Enter final public comment period
- July 2003: GGF "proposed recommendation"

What Is OGSI?

- Useful, general purpose plumbing to make it easier to build Web services relevant to Grids
  - OGSI came about because we started trying to define Globus Toolkit® functionality using WSDL, and found there were common, base behaviors that I wanted to define once and reuse in all of our services.
- But there is nothing Grid specific about OGSI!
- Perhaps it should have been better named:

WS-UsefulInterfacesToBuildAnInterestingClassOfWebServices

Open Grid Services Architecture (OGSA)

- A standard substrate: the Grid service
  - OGSI = Open Grid Service Infrastructure
  - Web services interfaces and behaviors that address key distributed system issues
- … supports standard service specifications
  - Resource mgt, dbms, workflow, security, …
  - Target of current & planned GGF efforts
  - OGSA wg defines "OGSA compliance"
- … and arbitrary application-specific services based on these & other definitions

"Web Services"

- For OGSI, Web Services = WSDL
  - OGSI is defined in terms of WSDL portTypes, messages, and XML Schema types
  - OGSI is largely silent on WSDL binding and service
- SOAP is important insofar as it defines a standard, inter-operable binding under WSDL. But OGSI is silent on this.
  - Ditto for WS-Security, etc.
- UDDI is registry function that could potentially be used with OGSI.
  - But OGSI also defines primitives for building custom or domain-specific registries

WSDL

- Web Service Description Language
- XML-based language for:
  - Abstractly describing message exchanges between clients and services
    - Types defined using XML Schema
    - Message comprising one or more parts of XML Schema types/elements
    - Operation = input/output or input only messages
    - Interface = named group of operations
  - Binding the interfaces to concrete protocols
    - E.g. Soap/http
- It says nothing about what messages are sent

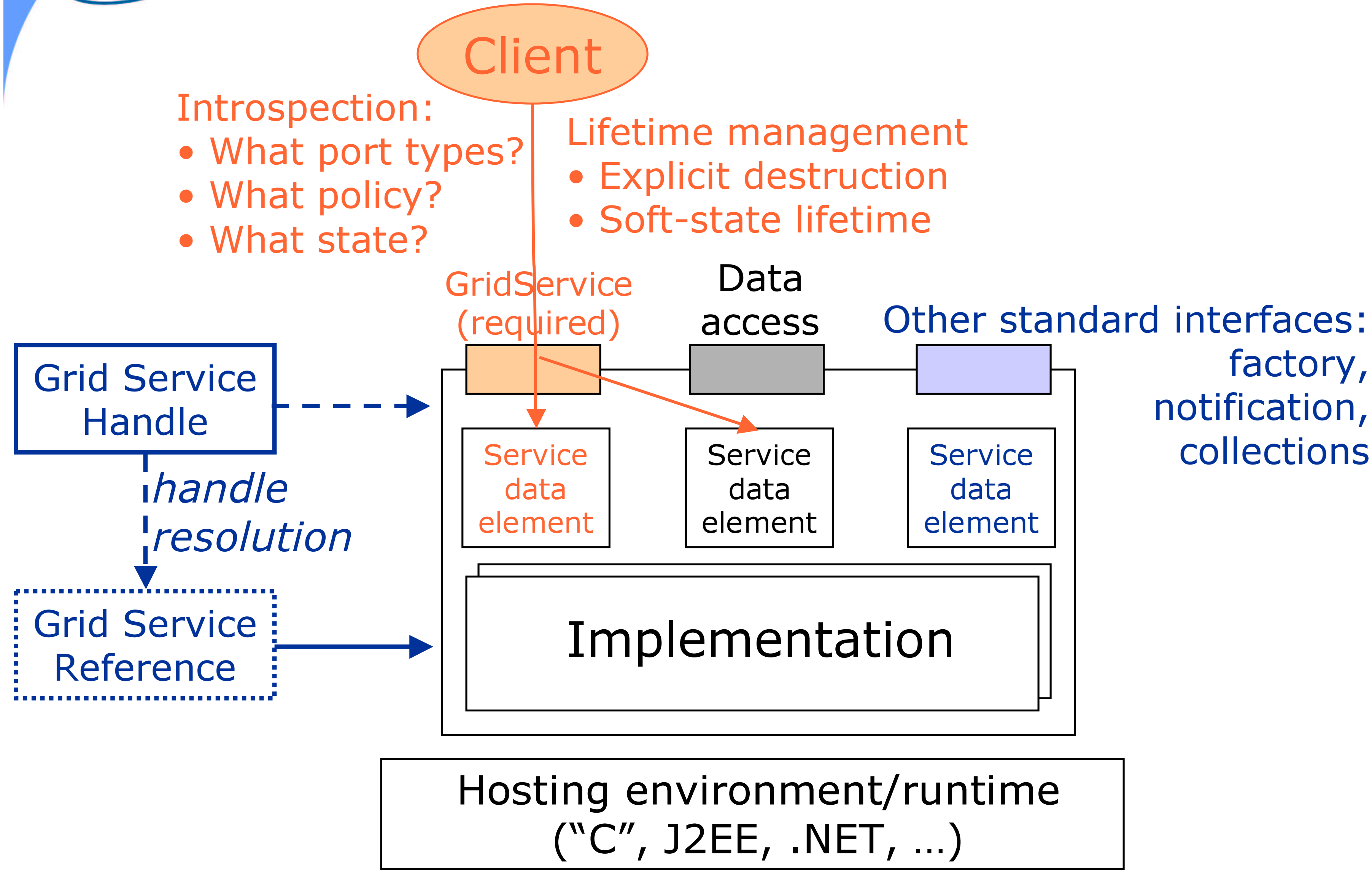
WSDL Example

```
<wsdl:definitions targetNamespace=“…”>
  <wsdl:types>
    <schema>
      <xsd:element name=“fooInput” …/>
      <xsd:element name=“fooOutput” …/>
    </schema>
  </wsdl:types>
  <wsdl:message name=“fooInputMessage”>
    <part name=“parameters” element=“fooInput”/>
  </wsdl:message>
  <wsdl:message name=“fooOutputMessage”>
    <part name=“parameters” element=“fooOutput”/>
  </wsdl:message>
  <wsdl:portType name=“fooInterface”>
    <wsdl:operation name=“foo”>
      <input message=“fooInput”/>
      <output message = “fooOutput”/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

OGSI Specification

- Defines WSDL conventions and extensions
  - For describing and naming services
  - Working with W3C WSDL working group to drive OGSI extensions into WSDL 1.2
- Defines fundamental interfaces (using extended WSDL) and behaviors that define a Grid Service
  - A unifying framework for interoperability & establishment of total system properties
- http://www.ggf.org/ogsi-wg

Open Grid Services Infrastructure



GWSDL

- OGSI requires interface extension/composition
- We worked within W3C WSDL working group to define standard interface extension in WSDL 1.2 that meets OGSI requirements
- But could not wait for WSDL 1.2
- So defined gwsdl:portType that extends WSDL 1.1 portType with:
  - WSDL 1.2 portType extension
  - WSDL 1.2 open content model
- Define GWSDL → WSDL 1.1 & 1.2 mappings

GWSDL Example

```
<wsdl:definitions>
  <wsdl:types>…</wsdl:types>
  <wsdl:message>…</wsdl:message>
  …
  <gwsdl:portType name=“foo”
                    extends=“ns:bar ogsi:GridService”>

    <wsdl:operation name=“op1”>…</wsdl:operation>
    <wsdl:operation name=“op2”>…</wsdl:operation>

    <ogsi:serviceData … />

  </gwsdl:portType>
  …
</wsdl:definitions>
```

Fundamental Interfaces & Behaviors

- OGSI defines basic patterns of interaction, which can be combined with each other and with custom patterns in a myriad of ways
- OGSI Specification focuses on:
  - Atomic, composable patterns in the form of portTypes/interfaces
    - Define operations & associated service data elements
  - A model for how these are composed
    - Compatible with WSDL 1.2
- Complete service descriptions are left to other groups that are defining real services

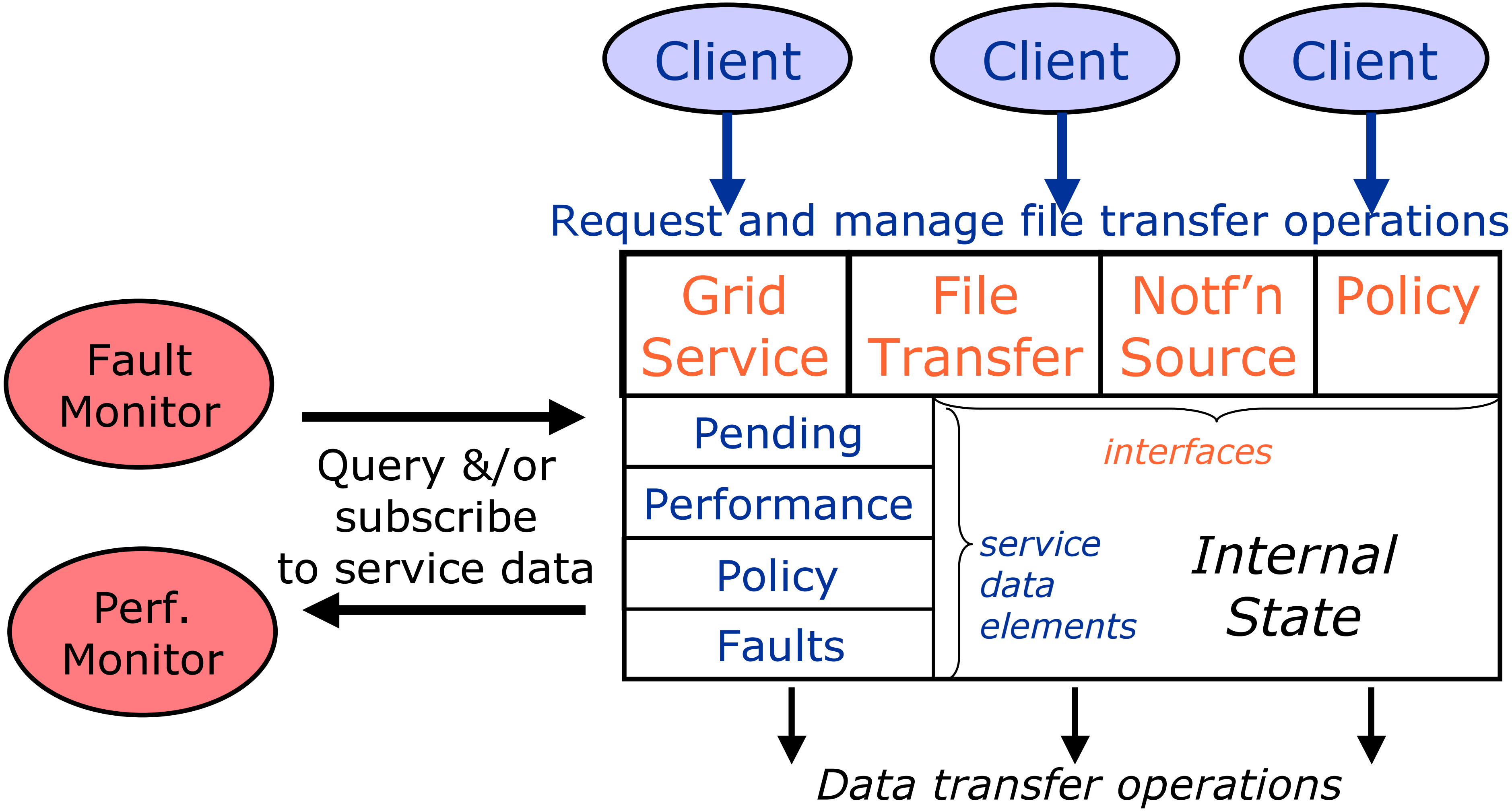
OGSI: Standard Web Services Interfaces & Behaviors

- Naming and bindings (basis for virtualization)
  - Every service instance has a unique name, from which can discover supported bindings
- Lifecycle (basis for fault resilient state management)
  - Service instances created by factories
  - Destroyed explicitly or via soft state
- Information model (basis for monitoring & discovery)
  - Service data (attributes) associated with GS instances
  - Operations for querying and setting this info
  - Asynchronous notification of changes to service date
- Service Groups (basis for registries & collective svcs)
  - Group membership rules & membership management
- Base Fault type

OGSI Service Data

- Attributes: Publicly visible state of the service
- Want to bring full power of XML to attributes
  - getXXX/setXXX is too limiting
    - How to get/set multiple?
    - Want richer queries across attributes (e.g. join).
  - Use XML Schema, XPath, XQuery, XSLT, etc.
  - OGSI service data:
    - Attributes defined using XML Schema
    - Attributes combined into a single (logical) document within the service
    - Rich pull/push/set operations against service data document
- Should declare attributes in WSDL interface

Example: File Transfer Service



Globus Toolkit® v3.0

- All of the GT v2.4 services and clients
- Complete Java implementation of OGSI v1.0
  - Rich, container-based implementation
  - Built on Apache Axis
- Globus “proprietary” services built on OGSI:
  - Managed Jobs (akin to GT2 GRAM)
  - Reliable File Transfer (RFT)
  - Index Services (akin to GT2 GIIS)
- Some services not yet OGSI-fied:
  - GridFTP, Replica Location Services (RLS)

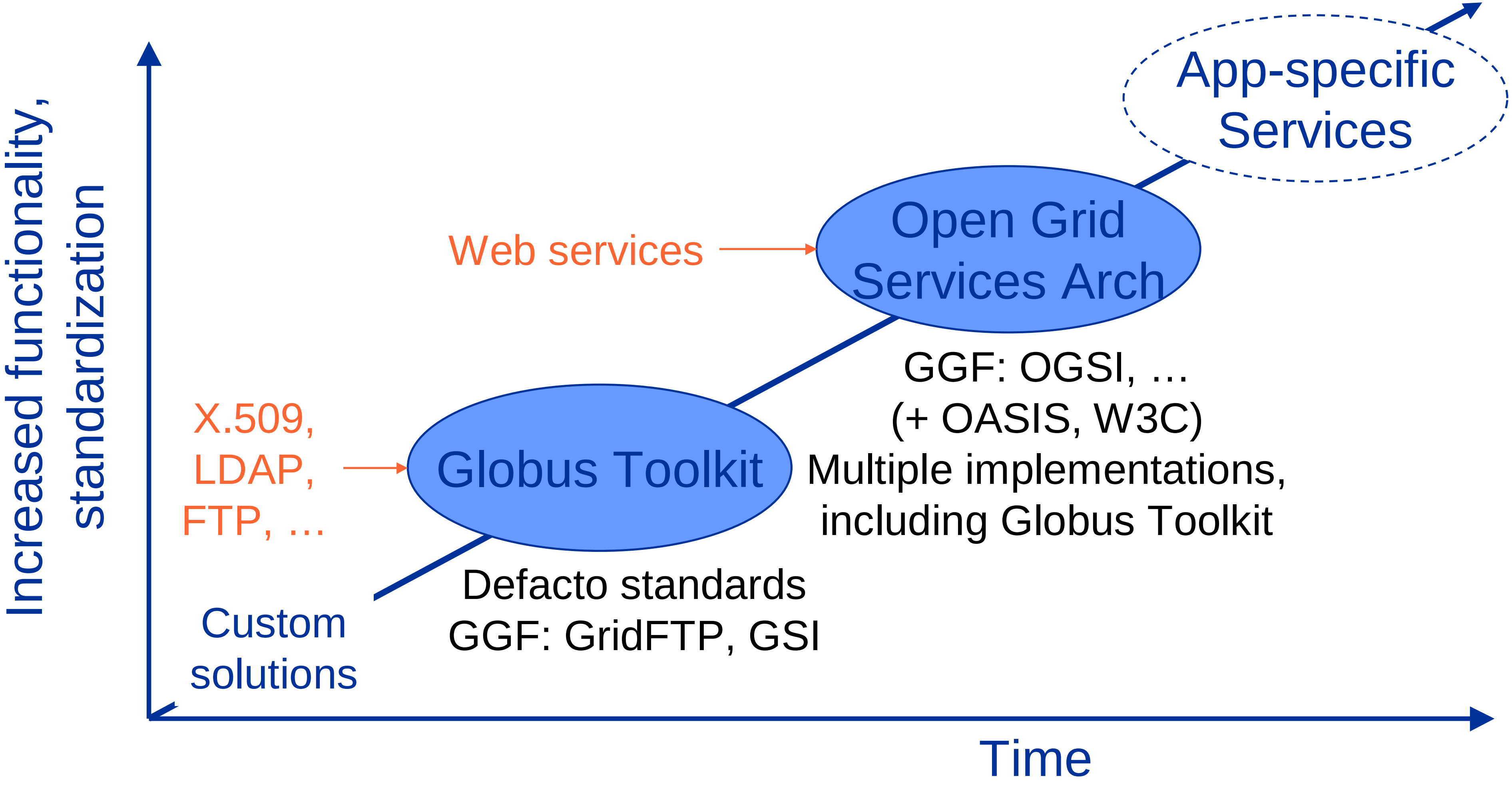
OGSI Implementations

- Globus Toolkit version 3.0 (Java, C client)
- U Virginia OGSI.NET (.NET)
- LBNL pyGlobus (Python)
- U Edinburgh (.NET)
- U Manchester (PERL)
- Fujitsu Unicore (Java)

Overview

- Where we are today
- Standards landscape & Globus Toolkit plans
  - Introduction
  - Plumbing
  - Agreements
  - Management
  - Core Services
  - Workflow
  - Summary
- Transitioning from GT2 to GT3

Grids and Open Standards



Increased functionality, standardization

Time

Custom solutions

X.509, LDAP, FTP, …

Globus Toolkit

Defacto standards
GGF: GridFTP, GSI

Web services

Open Grid Services Arch

GGF: OGSI, …
(+ OASIS, W3C)
Multiple implementations, including Globus Toolkit

App-specific Services

OGSA
Open Grid Services *Architecture*

- OGSA refers to the collection of specifications that together define a complete architecture
- GGF OGSA WG is defining OGSA
  - Services must be OGSI-compliant
  - Coordination group: Specifications for the services will come from other working group
  - Will define requirements, scope activities, …
  - This effort is just ramping up

GWD-R (draft-ggf-ogsa-platform-3) Open Grid Services Architecture Platform http://www.ggf.org/ogsa-wg	**Editors:** I. Foster, Argonne & U.Chicago D. Gannon, Indiana U.

Why Standards Matter

- Ubiquitous adoption demands open, standard protocols
  - Standard protocols enable *interoperability*
  - Avoid product/vendor lock-in
  - Enables innovation/competition on end points
- Further aided by open, standard APIs
  - Standard APIs enable *portability*
  - Allow implementations to port to different vendor platforms
- Internet and Web as exemplars

Relevant Standards Organizations

- GGF: Grid services: OGSI/A, WS-Agreement
- W3C: Web services: WSDL, SOAP
- OASIS: Web services security, WSDM, SAML
- IETF: Internet protocols and security
- Project Liberty Alliance: Identity federation
- DMTF: Common Information Model (CIM)

Intellectual Property issues

- Ubiquitous adoption is likely only to happen if IP is licensed royalty free (RF)
  - ◆ Core specifications must be RF
  - ◆ Higher level service specifications may be RAND (Reasonable and Non-Discriminatory) or even proprietary
- OGSI authors have made RF commitment
  - ◆ Many of the key IBM/Microsoft WS-* specs are not (currently) RF, though IBM/MS has stated intent to do so with some
  - ◆ But WS-Security is RF, so hopefully…

Globus Toolkit and Standards

- GT implements open standards as they emerge
- GT 2.x
  - X.509 (Proxy) Certs, GridFTP, LDAP, GSS-API
- GT 3.0 (June 2003)
  - GT2 + WSDL, SOAP, OGSI, WS-Security, etc.
- GT 3.2 (1Q2004)
  - Maintenance release + new GridFTP code
- GT 3.x (3-4Q2004)
  - First implementation of many of the standards to be discussed next…

Plumbing

WSDL

- WSDL 1.1 is a W3C “tech note”
  - ◆ Lesson: “standard” = result, not process
  - ◆ But process can affect resulting adoption
- WSDL 1.2 (2.0) is in progress in W3C
  - ◆ Clean up, clear up ambiguities, etc.
  - ◆ Target SOAP 1.2 (standardized SOAP)
  - ◆ Add interface inheritance + open content
    - OGSI requirements & people were critical to these additions
- Unsure of completion date
  - ◆ Original target: Fall 2002, Now: end of 2003

OGSI

- OGSI 1.0 completed in GGF in July 2003
- Standard interfaces for common patterns
  - Naming, lifetime, inspection, grouping, etc.
- GWSDL = WSDL 1.1 + WSDL 1.2 interface extension + open content
  - Draft spec for GWSDL → WSDL 1.1 mapping
  - GWSDL → WSDL 1.2 expected
- Start OGSI v2 in 2004 (perhaps split it up)
  - Build on WSDL 1.2 and other emerging specs
  - 12-24 month process

Transactions & Contexts

- WS-Coordination & WS-Transaction
  - IBM/MS (not in standards org)
- WS-CAF (Coordinated Application Framework)
  - Sun/Oracle/Arjuna/Fujitsu (not in standards org)
  - WS-CTX (Context)
  - WS-CF (Coordination Framework)
  - WS-TXM (Transaction Management)
- Both take a “contextualization” approach
  - Context (id) threaded through SOAP header
  - OGSI for context creation, naming & lifecycle???

State in a Service Oriented Architecture

- Don’t confuse:
  - Stateful/less connection
    - Web Services Architecture (WSA) requires stateless connections
  - Stateful/less interaction
  - Stateful services
- Two complimentary patterns to managing state, both valid within WSA & WSDL:
  - Encapsulation into a service (OGSI)
  - Contextualization (WS-Transaction, WS CAF)

WS-Addressing

- IBM/MS (not in standards org)
- Endpoint Reference
  - To convey information about a Web services endpoint
    - Address, reference properties, WSDL interfaces, WSDL service, policy
- Message Information Headers
  - SOAP header for flowing endpoint reference with a message
- Could be useful in future for OGSI GSR

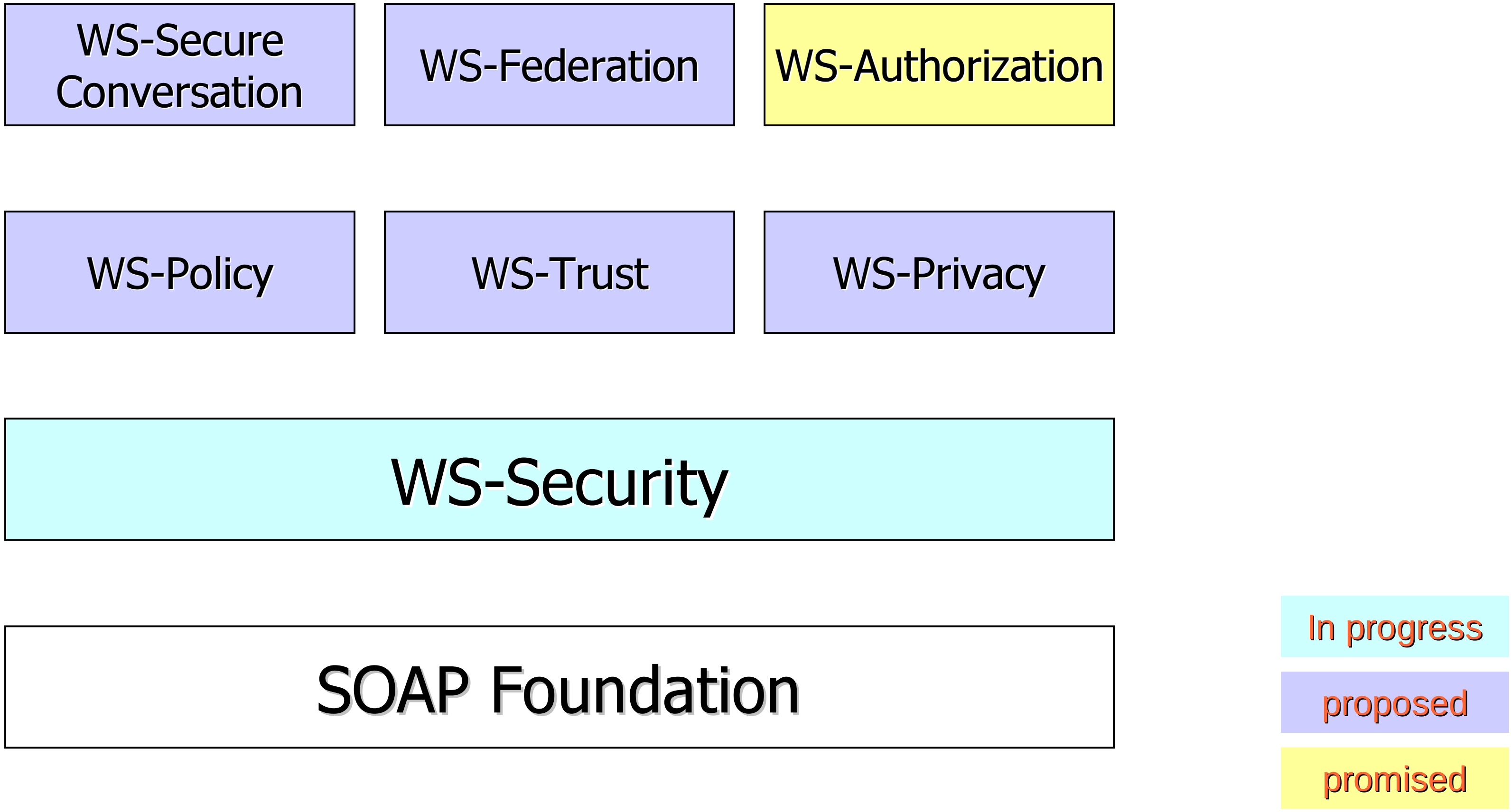
Security Standards

- Many core security standards are from IETF
  - X.509, Kerberos, etc.
  - X.509 Proxy Certificates (RFC soon hopefully)
    - Used by Globus Toolkit GSI
- OASIS appears to be leader in Web services security standards
  - WS-Security: SOAP message security
  - SAML: signed assertions using XML
  - XACML: access control lists using XML
- GGF OGSA Security WG evaluating security specifications for applicability to OGSA

IBM/Microsoft WS Security Architecture

- Large set of specifications for doing Web services security, most of which should be appropriate for OGSA
- Announced April 2002
- Initial spec in July 2002 (WS-Security)
  - Submitted to OASIS
- New crops of specs arrive periodically
  - WS-Policy*, WS-Trust, WS-Federation, etc.
  - But… Not yet in any standards organization

WS Security
Current/Proposed WSS-specs



WS-Secure Conversation

WS-Federation

WS-Authorization

WS-Policy

WS-Trust

WS-Privacy

WS-Security

SOAP Foundation

In progress

proposed

promised

OASIS SAML & XACML

- SAML: Security Assertion Markup Language
  - Good for asserting properties such as group membership, etc
- XACML: eXtensible Access Control Markup Language
  - For defining access control policies
- These are gaining considerable momentum, but WS-Policy* leaves these in question

Project Liberty Alliance

- V1.x specifications for identity federation
  - Allows cross-organization identification
  - Privacy preserving model
- Based on SAML

Accounting & Billing

- A couple GGF activities, but their traction is uncertain
  - ◆ GGF resource usage record schema
  - ◆ GGF resource usage protocol
- This is area that needs work

Agreements

(OGSI-) WS-Agreement

- Recall key criteria of a Grid:
  - *Coordinates resources that are not subject to centralized control* …
  - using standard, open, general-purpose protocols and interfaces …
  - *to deliver non-trivial qualities of service*.
- Implies need to express and negotiate agreements that govern the delivery of services to clients
  - Agreement = what will be done, QoS, billing, compliance monitoring

WS-Agreement Contents

- Standard agreement *language*
  - A composition of a set of terms that govern a service’s behavior with respect to clients
  - Agreement language uses WS-Policy (currently)
  - Standard attributes for terms that express current state of negotiation
  - Other groups define specific terms
- Standard agreement negotiation *protocol*
  - Establish, monitor, re-negotiate agreement
  - Expressed using OGSI GWSDL interfaces
  - Each agreement represented by a service

WS-Agreement Applicability

- All interesting Web/Grid services interactions will be governed by agreements!
- WS-Agreement (language and interfaces) should be used by specifications that define domain specific services
  - Data services
  - Job submission
  - Specialized services
  - Etc.

Management

WSDM / WSMF / CMM

- OASIS Web Services Distributed Management (WSDM) technical committee
  - Management using/of Web Services
  - HP submitted its Web Services Management Framework (WSMF) to WSDM in July 2003
    - WS-Events: event schema, subscription, message queues
    - WSMF-Foundation: management *using* Web services
    - WSM: management *of* Web services
- GGF Common Management Model (CMM) WG
  - IBM submission overlaps WSMF-Foundation
- Working to bring WSDM & CMM together

WSMF-Foundation

- Defines base constructs of ManagedObject:
  - Identity, references, relationships, states, and faults
- Defines six WSDL interfaces
  - ManagedObjectIdentity (required)
  - ManagedObjectConfiguration
  - ManagedObjectMonitoring
  - ManagedObjectDiscovery
  - ManagedObjectControl
  - ManagedobjectCollection

WSMF & OGSI

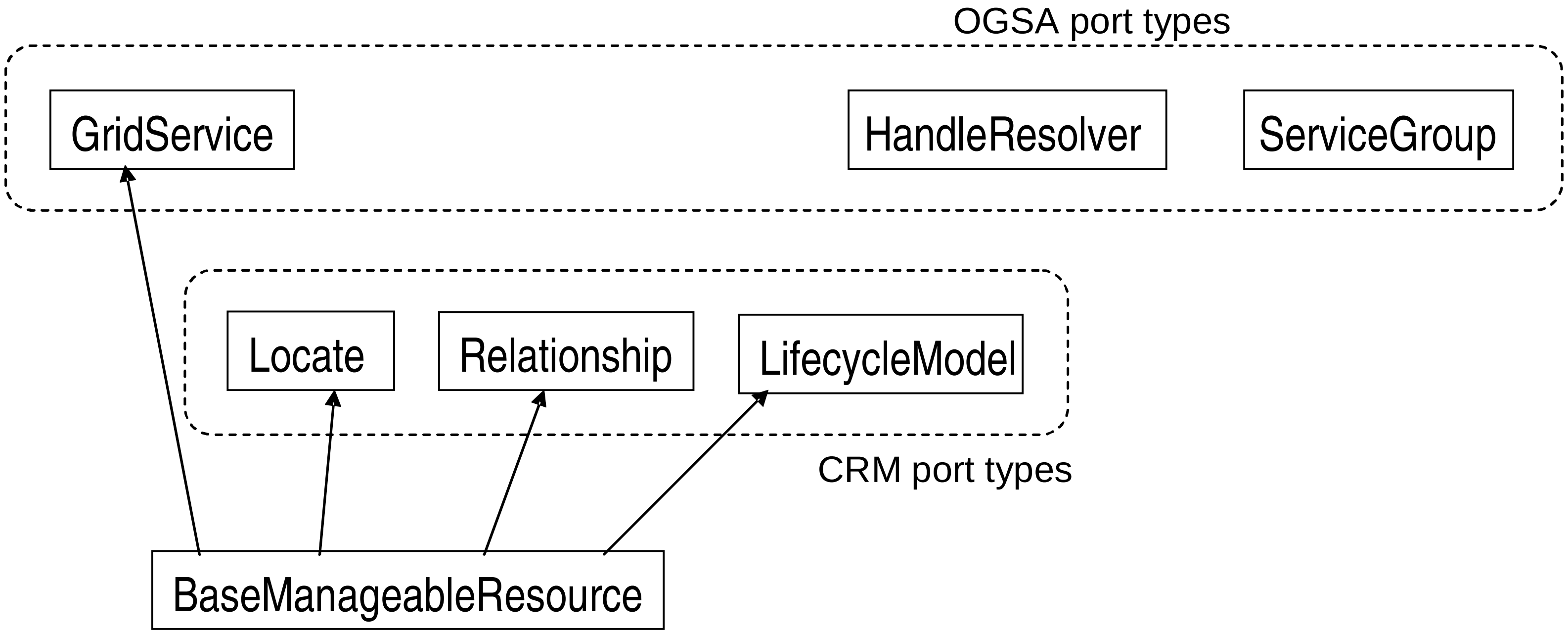
- WSMF re-defines most of OGSI. E.g.:
  - ManagedObject identity & refs → GSH, GSR
  - Attributes, events → Service Data Elements
  - Collection → ServiceGroup
  - Faults
- Globus and HP are working together to re-factor WSMF-Foundation and WS-Events specifications to exploit OGSI to achieve a more powerful expression of a generalized Web services framework for building management interfaces

Common Management Model

- A manageable resource is a Grid service, thus
  - ◆ Global resource names: Grid service handles
  - ◆ State data modeling + access: SDEs
  - ◆ Lifetime management
  - ◆ Service Group for grouping resources
  - ◆ Interface definition language: WSDL
- Plus additional schema & operations
  - ◆ Standard manageable resource SDE schema
  - ◆ Interfaces for extensible lifecycle and relationship management

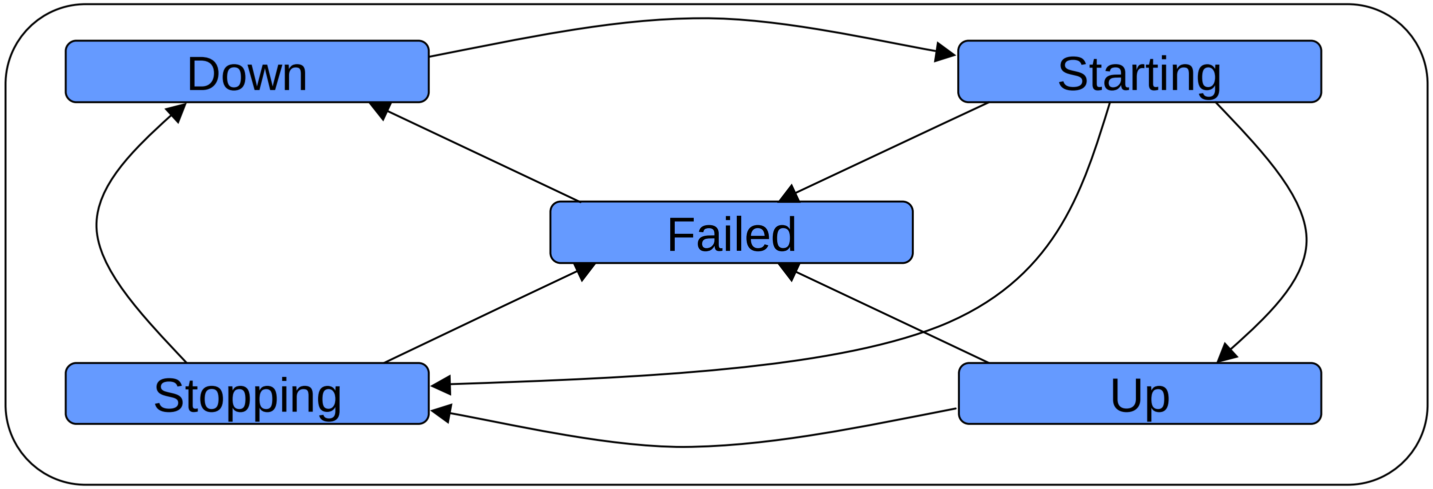
⇒ BaseManageableResource interface

Base Manageable Port Types



LifecycleModel Port Type: A Container for Lifecycle States

- There may be multiple models, but only one for a given resource’s port type
- Example: Get/set resource’s lifecycle state
  - down, starting, up, stopping, failed
  - Each state has additional info, e.g.,
    - up state: idle, busy, degraded



WSMF & CMM

- Once you re-factor WSMF-Foundation to use OGSI, what you are left with looks (in scope and concept at least) similar to CMM
  - Service lifecycle (state) models
  - Service relationships
- These efforts should be able to come together into a single standards effort that meets both Web Services & Grid communities needs
  - *If* they base on OGSI (discussions in process)

Event Schema

- A standard schema for representing (containing) events
  - ◆ Standard ways of representing time information, parties involved, etc.
- Proposals:
  - ◆ HP WS-Events includes a simple event schema
  - ◆ IBM presented richer event schema WSDM f2f
- These should be able to come together in WSDM

Message Queues

- Aka channels, aka buffered notification
- OGSI has simple notification model
  - SDE represents visible, typed, changeable state/attributes of a service
  - Extensible subscription to notification of changes of SDE (SDEs name = event source)
- What is missing?
  - Buffering of messages
    - Allows richer QoS, pull delivery, batched push delivery, …
  - Hierarchical topic names

WS-Events & OGSI

- HP’s WSMF includes WS-Events
  - Non-hierarchical event source names
  - Buffered delivery w/ push & pull
- HP & Globus re-factoring WS-Events for OGSI
  - SDEs are the event source names
  - Extending OGSI subscription interfaces to supported “buffered notification”
- Hopefully this will come together soon with IBM work on hierarchical topic based approach

Core Sevices

Data Services

- GGF Data Access and Integration Svcs (DAIS)
  - OGSI-compliant interfaces to access relational and XML databases
  - Needs to be generalized to encompass other data sources (see next slide…)
- Generalized DAIS becomes the foundation for:
  - Replication: Data located in multiple locations
  - Federation: Composition of multiple sources
  - Provenance: How was data generated?

"OGSA Data Services" (Foster, Tuecke, Unger, eds.)

- Describes conceptual model for representing all manner of data sources as Web services
  - Database, filesystems, devices, programs, …
  - Integrates WS-Agreement
- Data service is an OGSI-compliant Web service that implements one or more of base data interfaces:
  - DataDescription, DataAccess, DataFactory, DataManagement
  - These would be extended and combined for specific domains (including DAIS)

Job Submission / Service Deployment

- Globus Toolkit’s GRAM/ManagedJob service allows for “job” submission and management
  - This will be standardized in terms of:
    - WS-Agreement: Base protocol and language for submission
    - JSDL (maybe): WS-Agreement terms for job submission
    - WSDM: Base manageability interfaces, states, etc.
  - Also add WS-Agreement-based services for advance reservation and account management
- More generally, evolve to a standard for service deployment
  - E.g. Linux image, app server, services

Workflow

- Two competing efforts
  - OASIS BPEL
    - Led by IBM & MS
  - W3C WS-Choreography
    - Led by Sun, etc.
- Both take a similar approach
  - XML-based programming language for specifying workflows comprising Web services invocations
  - Thread contexts through a series of invocations
- What changes are needed for OGSI?

Standards Summary

- Standards are critical to Grid success
- Grid and Web Services are merging
  - Grid is an aggressive use case of Web Services
- Web Services standards landscape is in great flux, and OGSI will need to evolve with it
- Grid Services standards landscape heating up
- W3C, OASIS, GGF are key standards orgs
- Uncertain status of security & policy standards continues to be a big source of concern
- Open source software important for adoption

Overview

- Where we are today
- Standards landscape & Globus Toolkit plans
- Transitioning from GT2 to GT3
  - GRAM
  - MDS
  - GridFTP
  - Reliable File Transfer (RFT)
  - Replica Location Service (RLS)

Transition Message

- Do it incrementally!!!
  - GT3 combines:
    - All of the services that you are familiar with from GT2
    - Additional OGSI-compliant services
      - Some are new (e.g. RFT)
      - Some are supersets of GT2 functionality (e.g. ManagedJob)
  - Eventually all of the GT2 functionality will be covered in OGSI-compliant services
    - Start testing them as they arrive, and migrate when they meet your needs
    - OGSI services is where most new functionality will appear

GT 3.2

- Targeted for 1Q2004
- Contains:
  - Bug fixes and performance improvements
  - New GridFTP code (client & server)
    - Not wu-ftpd based
- Message:
  - Start developing and testing now so that any problems you run into can be fixed and delivered in 3.2
  - Production rollout on 3.2

GRAM

- GT 3.0 includes OGSI-compliant ManagedJob service
  - Uses same backend submission scripts as GT2-compatible services (job manager)
  - Functionality is almost identical
  - Still working on improving scalability and performance (3.2)
- Working on standards now that will replace ManageJob and add new functionality (e.g. advance reservations)

Monitoring & Discovery System (MDS)

- Conceptually identical to GT2 MDS
- But factoring and capabilities have changed considerably with OGSI
  - No longer a separate GRIS server – now every OGSI service is its own GRIS, through service data
  - Index service (like GIIS) still available, but with Xpath query support
- Working on richer queuing, logging, and archiving support for future releases

GridFTP

- GT 3.0 is same as 2.4 (not OGSI-compliant), just with some bug fixes
- GT 3.2 will contain new GridFTP
  - Built from scratch (no more wu-ftpd)
  - Will grow to include striped server in 3.4
- GGF DAIS working group will probably include filesystem data services
  - "OGSA Data Services" paper lays the foundation
  - Filesystem interface specs will be coming

Reliable File Transfer (RFT)

- New OGSI-compliant service in GT 3.0
  - Accepts requests from clients to manage a third party transfer between two GridFTP servers
- This is probably the most natural place to start with OGSI-compliant GT3 services
- Will evolve to support DAIS filesystems when available

Replica Location Service (RLS)

- Not OGSI-compliant in 3.0
- Working on OGSI-compliant services with RLS capabilities, for future GT 3.x
  - Exploits “OGSA Data Services”

Conclusion

- OGSI v1.0 is done, and ready for use today
- Globus Toolkit future is intimately tied to the evolution of Web services and other Grid standards
- There is a lot of standards activity, but the 12-18 month picture is starting to clarify
- Transition from GT2 → GT3 should be incremental