The Open Grid Services Architecture (OGSA)

Seminar Distributed Computing
Wintersemester 2003/2004

Till Adam (till@adam-lilienthal.de)

Overview

What is OGSA and what problem does it solve?

The Service Architecture Approach

Web Services Technologies

Open Grid Service Infrastructure (OGSI)

Higher level models, interfaces and behaviors (OGSA)

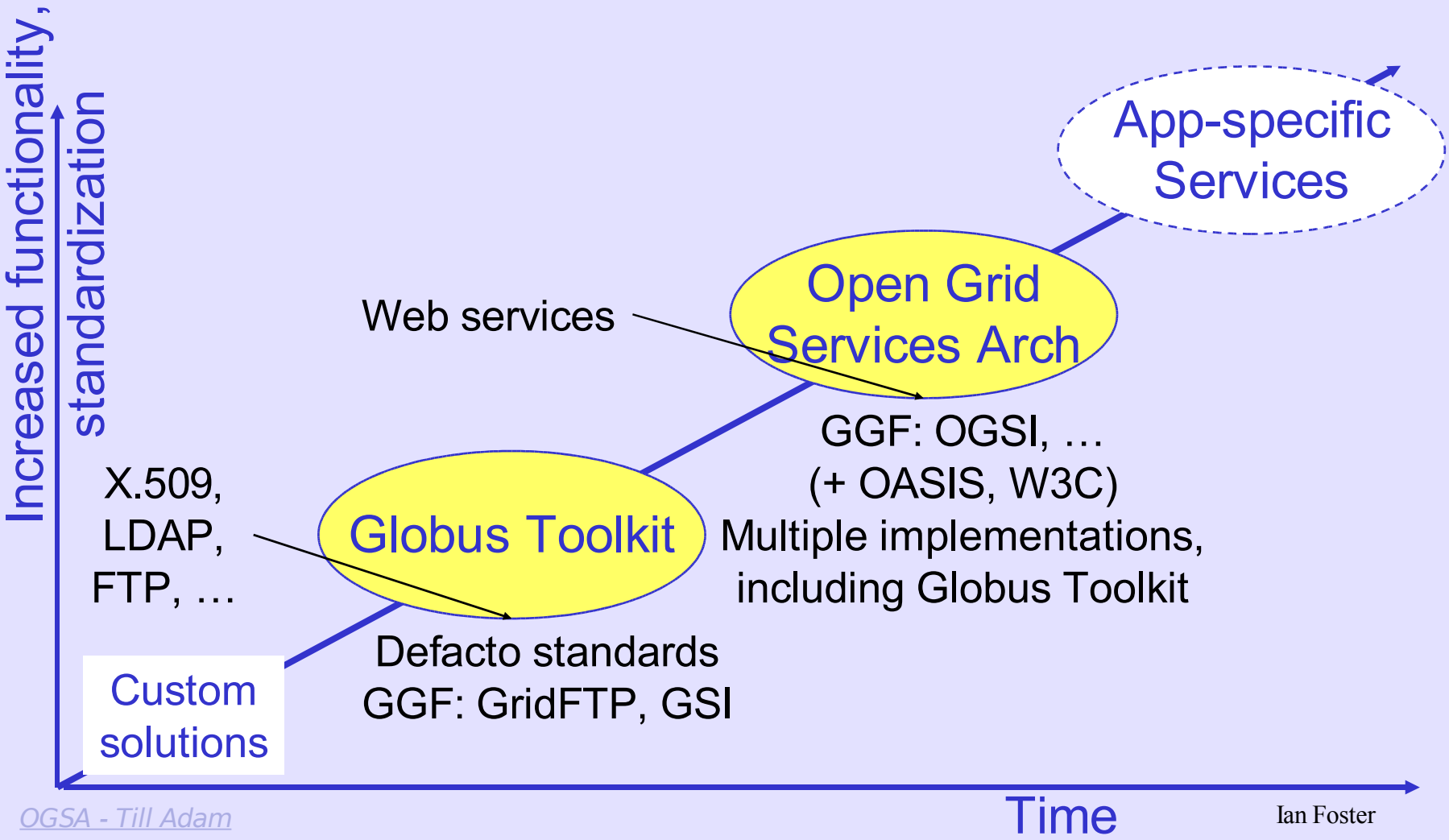
Status Quo and Perspective

The Problem

- services need to be integrated across distributed, heterogeneous, dynamic "virtual organizations"
- internal and external ressources
- service providers
- various qualities of service (QoS) need to be achieved
- various different native platforms

=> standardized architecture is needed to address issues of safety, efficiency, naming, discovery, and much more

Historic Perspective



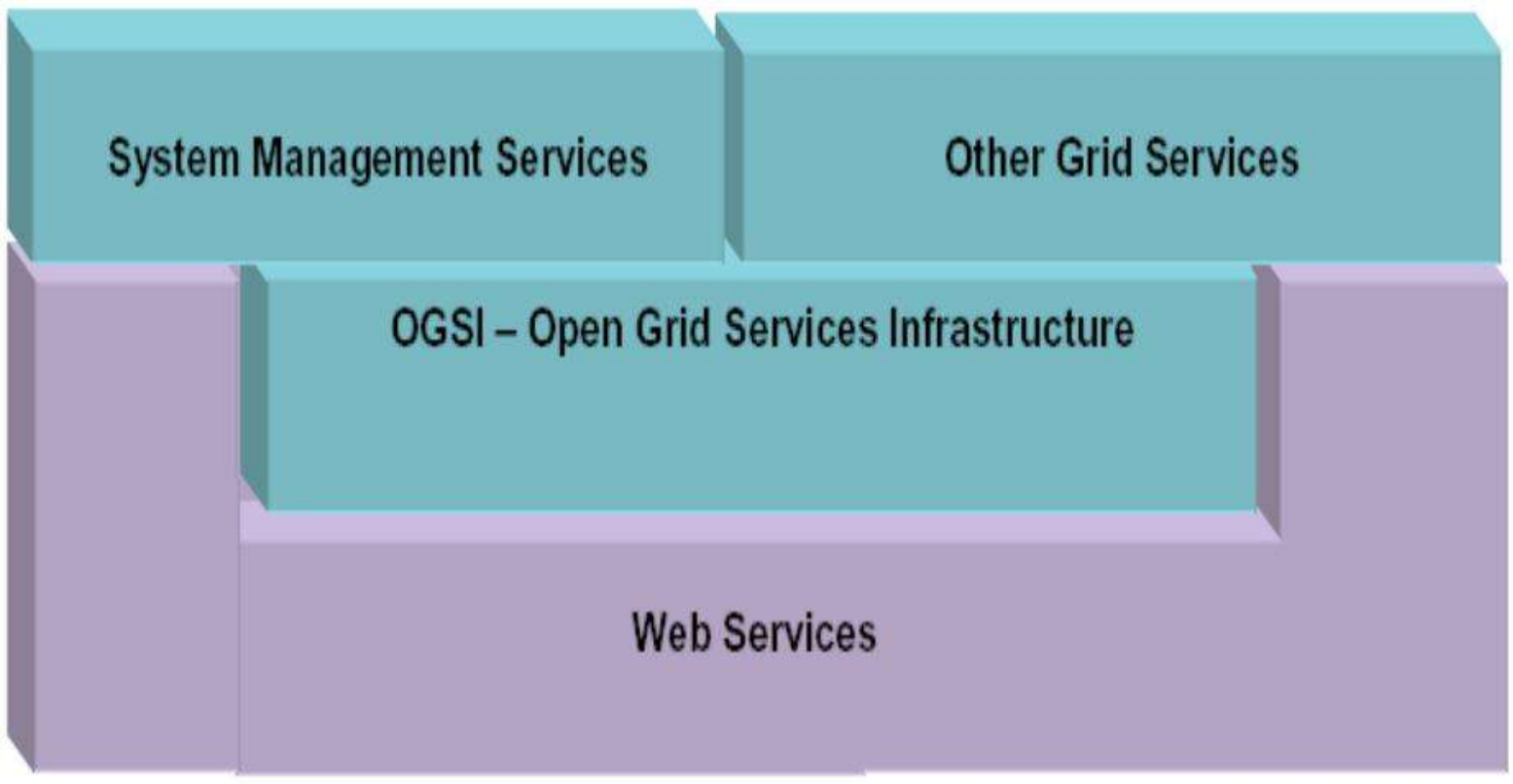
OGSA

- defines a uniform exposed service semantics (the Grid service)
- defines standard mechanisms for creating, naming, and discovering (potentially transient) Grid service instances
- provides location transparency and multiple protocol bindings for service instances
- supports integration with underlying native platform facilities

OGSA

- defines, in terms of the Web Services Description Language (WSDL) interfaces and associated conventions
- defines mechanisms required for creating and composing sophisticated distributed systems, including lifetime management, change management, and notification
- Service bindings can support reliable invocation, authentication, authorization, and delegation, if required

OGSA Structure



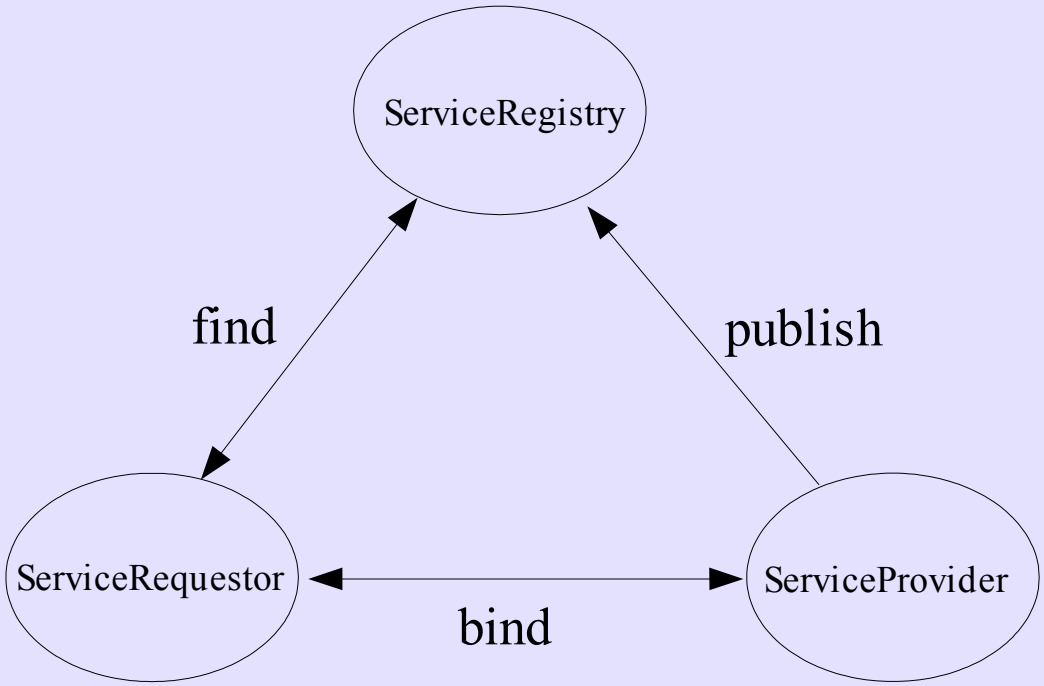
Services

A service is:

- an entity that provides some capability to its clients by exchanging messages
- only defined by identifying sequences of specific message exchanges that cause the service to perform some operation

=> location and implementation agnostic
=> very flexible

Services


ServiceRegistry
find
publish
ServiceRequestor
ServiceProvider
bind

Service Examples

- Storage Service
  - storing, retrieving data
  - reserving space
  - access policy
- Data Transfer Service
  - transfer of data between services
- Troubleshooting Service
  - monitoring the status of other services
  - initiating appropriate actions

Service Architecture

- All entities in a service-oriented architecture are services.
- All operations visible to the architecture are the result of message exchange.
- Services are composed into more complex services.

Interfaces

- Abstract concepts can be expressed as a grouping of operations into a service interface.
- Interfaces can be combined to specify a service with a certain behavior.
- Services are *virtualized.*
- Implementation and Interface are de-coupled.

Interface Description

- Standard interface description language (IDL) is used to describe service interfaces.
- IDL defines only the messages a service produces and consumes, not the behavior itself.

Service Architecture based on interfaces

allows for:

- service discovery
- service composition
- specialization
- interface extension

Web Services

- Web Services solve many of the problems
- Service Based Architecture
- XML based infrastructure for standardized access to information across the internet
- IDL already exists (WSDL)
- Remote Procedure Call mechanisms (SOAP)
- Transport bindings (HTTP/SMTP)
- Discovery mechanisms (UDDI: Universal Description, Discovery and Integration)

WSDL

"an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages"

- **Types** - a container for data type definitions using some type system (such as XSD).
- **Message** - an abstract, typed definition of the data being communicated.
- **Operation** - an abstract description of an action supported by the service.
- **Port Type** - an abstract set of operations supported by one or more endpoints.
- **Binding** - a concrete protocol and data format specification for a particular port type.
- **Port** - a single endpoint defined as a combination of a binding and a network address.
- **Service** - a collection of related endpoints.

SOAP

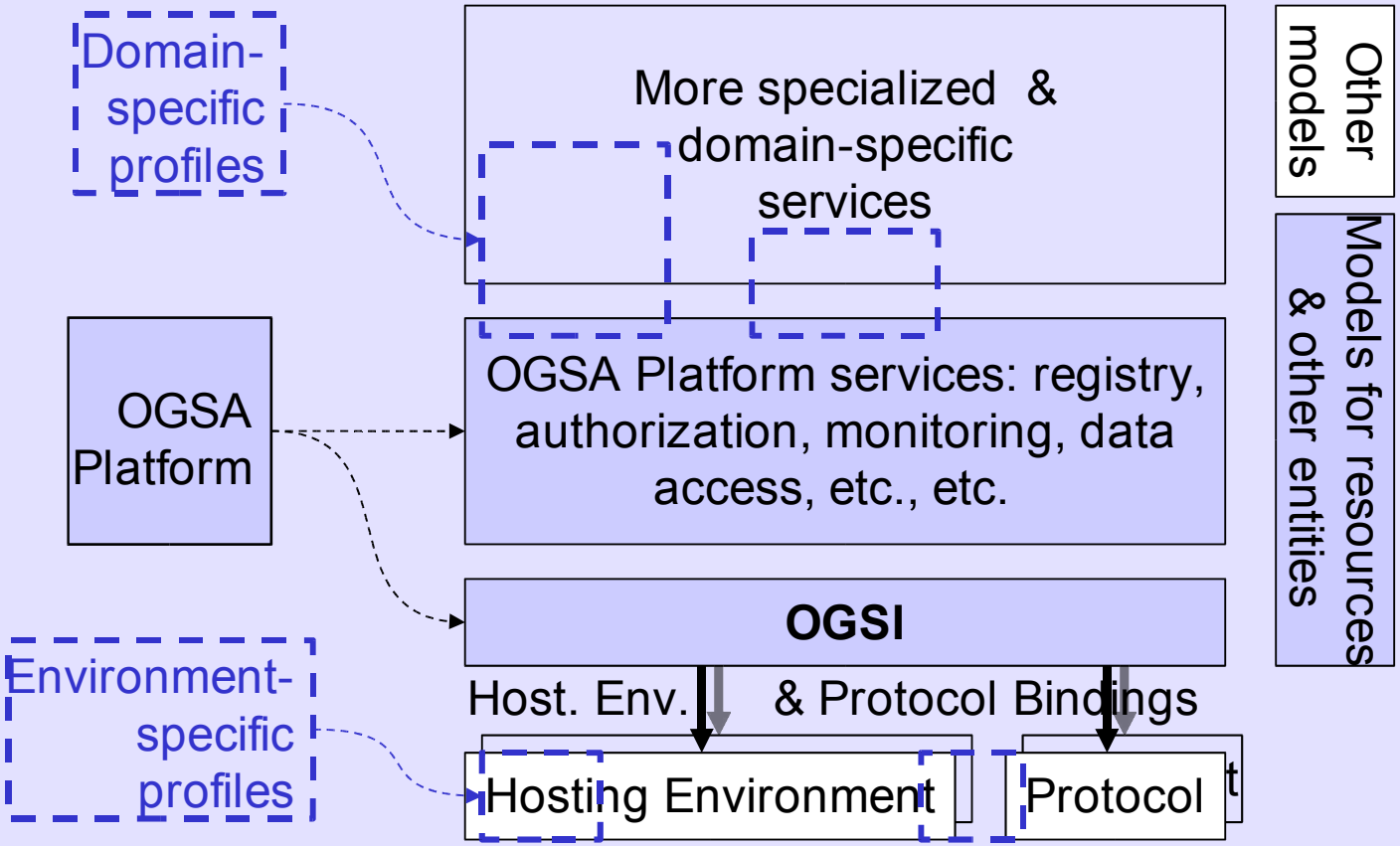
“SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses.”

What is missing?

- Web Services address discovery & invocation of **persistent services**
- Grids must also support **transient service** instances, created/destroyed dynamically
- Web services have no notion of state

Grid service

- A (potentially transient) Web Service with specified interfaces & behaviors
  - Creation
  - Global naming & references
  - Lifetime management
  - Registration & Discovery
  - Authorization
  - Notification
  - Concurrency
  - Manageability



GWD-R (draft-ggf-ogsa-platform-3)
Open Grid Services Architecture Platform
U.Chicago
http://www.ggf.org/ogsa-wg

Editors:
I. Foster, Argonne &

D. Gannon, Indiana U.

OGSI

- standard interface definition and associated semantics for common service interactions
- needed to build interoperable, reusable components
- combination of wsdl and human readable specifications
- defines mechanisms for creating, naming, managing lifetime, monitoring, grouping, and exchanging information among entities called Grid Services
- standard factory and group registration interfaces for creating and discovering grid services

OGSI Interfaces

- **GridService** – must be implemented
- **HandleResolver** – mapping GSH to GSR
- **NotificationSource** – allow subscription to notif.
- **NotificationSubscription** – manage subscriptions
- **NotificationSink** – single operation for delivery
- **Factory** – creation of Grid service instances
- **ServiceGroup** – grouping and associated policies
- **ServiceGroupeRegistration** – addition/removal
- **ServiceGroupEntry** – lifetime and properties

Service Data

- wsdl has no notion of state
- service data element (SDE)
- named, typed XML elements with access control properties
- Service Data Element Schema
- pull mode, push mode

Service Data

```
<wsdl:definitions xmlns:tns="…" targetNamespace="…">
  <gwsdl:portType name="StorageService" extends="ogsi:GridService">
    <wsdl:operation name=getFile>
      …
    </wsdl:operation>

    <sd:serviceData name="capacity" type="xsd:integer" />
    <sd:serviceData name="location" type="xsd:string" />
    <sd:serviceData name="speed" type="xsd:integer" />
    <sd:serviceData name="freeSpace" type="xsd:integer" />
    <sd:serviceData name="load" type="xsd:integer" />
    <sd:serviceData name="outOfSpaceError" type="ogsi:FaultType"
                    minOccurs="0" />
    <sd:serviceData name="accessControlPolicy"
                    type="tns:PolicyType"/>
    <sd:serviceData name="activeTransfer"
                    type="tns:ActiveTransferType"
                    minOccurs="0" maxOccurs="unbounded"/>
  </gwsdl:portType>
</wsdl:definitions>
```

Service Data

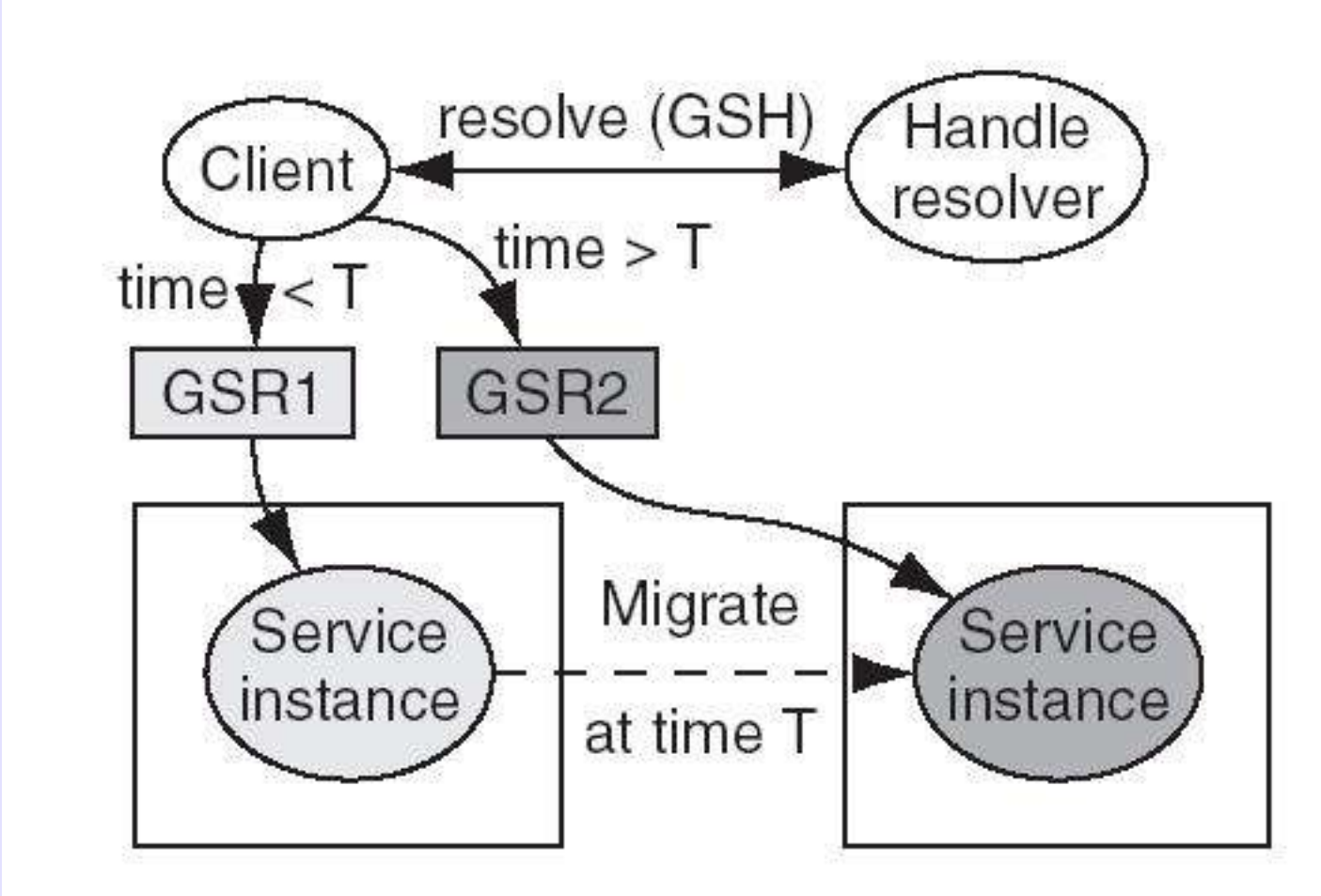
- Pull mode:
  - client queries the service
  - *findServiceData* Operation of the **GridService** interface
- Push mode:
  - via (optional) **NotificationSource**, **NotificationSubscription** and **NotificationSink** interfaces
  - operations such as *subscribe* with sink and initial lifetime arguments
  - periodic keepalive messages

Naming

- Grid Service Handler (GSH)
  - globally unique name
  - long lived, abstract
  - used for lookup
  - described by a Uniform Rersource Identifier (URI)

- Grid Service Reference (GSR)
  - concrete but potentially shorter-lived
  - returned by a handle resolver (service implementing HandleResolver interface)

Naming


Client
resolve (GSH)
Handle resolver
time < T
time > T
GSR1
GSR2
Service instance
Migrate
at time T
Service instance

Service Life Cycle

- creating transient services: factories
  - *createService* operation
  - returns GSH and initial GSR
  - should register the instance with handle resolver

- service lifetime management
  - soft-state approach
  - on creation earliest and latest acceptable termination times are indicated
  - factory selects initial termination time
  - Infinity disables soft-state management
  - *destroy* – explicit termination
  - repeated *requestTerminationAfter* messages

Fault Model

- consistent error handling across OGSI
- standard XSD type: ogsi:FaultType
- two elements: originating service, timestamp
- optionally plain language description
- nested faults
- similar to programming language exceptions

Service Groups

- ServiceGroup
  - optional constraints on membership with
  - membership-ContentRule SDE
- ServiceGroupEntry
  - lifetime management functions for individual entries
- ServiceGroupRegistration
  - add and remove operations

OGSA Services

- those grid components that are sufficiently broadly applicable that we can expect to see them in any grid system
- service discovery, service management, monitoring, security, data access and messaging
- work in progress

Core Services

- Name resolution and discovery (GSHs -> GSRs)
- Service Domains
- Security
- Policy
- Messaging queuing and logging
- Events
- Metering and accounting

Data and Information Services

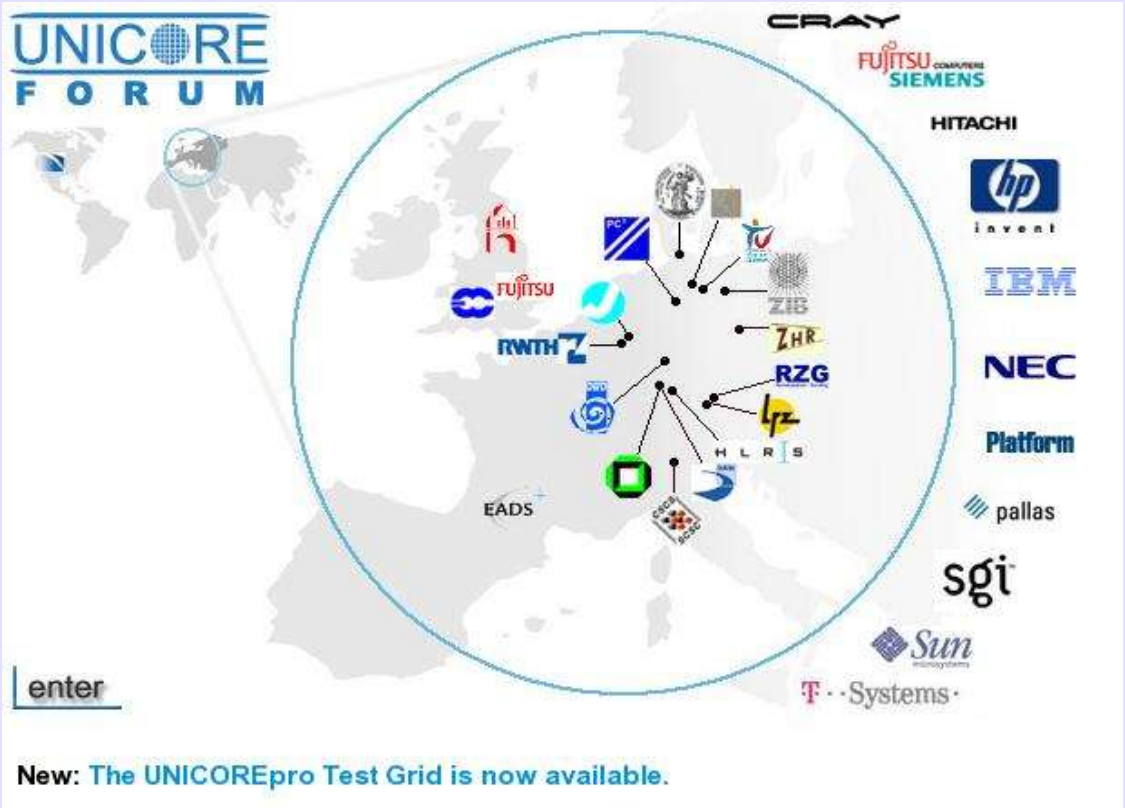
- Data naming and access
- Replication
- Metadata and Provenence

Ressource and Service Management

- Provisioning and ressource management
- Service orchestration
- Transactions
- Administration and Deployment

Status Quo

- Globus Toolkit Version 3.0.2
- Unicore (www.unicore.org) java implementation



Status Quo

- OGSA Evaluation project at the Department of Computer Science of the University College London (http://sse.cs.ucl.ac.uk/UK-OGSA/)
- Grid service enhancements to web services are intended to become part of the web services architecture
- Pending integration into major platforms such as:
  - J2EE
  - Microsoft .NET
  - IBM e-business on demand

Questions?