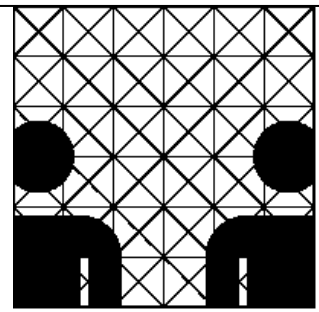


OGSA

Open Grid Services Architecture

Gliederung

1. Kurzer Rückblick in die betriebliche IT – Nutzung
2. Motivation für OGSA
3. Grundlagen für OGSA
4. OGSA-Aufbau
5. Beispiel: Anwendung der OGSA-Spezifikation

Rückblick in die betriebliche IT-Nutzung

EDV-Landschaft früher:

- zentralistisch
- stark integrierte Komponenten
- statisch
- Systeme gingen in der Regel nicht über Unternehmensgrenzen hinweg

Aufkommende Veränderungen

- B2B-Relationships erfordern eine Vernetzung zwischen Unternehmen
- Kostenersparnisse durch die Einbindung fremder Dienstleistungen
- Komplexere Berechnungsaufgaben müssen gelöst werden, die von den bestehenden IT-Systemen der Unternehmen nicht erfüllt werden können

Entstandene Anforderungen

- Integration von
  - verteilten (Global)
  - heterogenen
  - dynamischen Systemen
- Zugriff auf dezentrale und fremde Ressourcen
- effektivere Ausnutzung der Ressourcen
- Standardisierung der Schnittstellen

Was ist Grid-Computing?

- steht für „verteiltes“ Rechnen
- effektivere Ausnutzung vorhandener IT-Kapazitäten (Daten, Speicher, Anwendungen, Server und Rechenleistung)
- Bewältigung komplexer Berechnungsaufgaben durch ihre Verteilung
- erfolgreicher Einsatz in der Wissenschaft (SETI-Projekt, Entschlüsselung von Genen usw.)

Was sind Web-Services?

- sind modulare Anwendungen, die beschrieben, veröffentlicht, lokalisiert und über ein Netzwerk aufgerufen werden können, meistens über das Internet
- sind plattform- und implementierungsunabhängig
- komplexe Web-Services lassen sich aus einfacheren schachteln

Was ist OGSA?

- vorgestellt von IBM und den Forschern vom Globus Project
- eine Sammlung von Spezifikationen und Standards, das die Vorteile von Grid-Computing mit denen von Web-Services kombinieren soll
- Ziel ist es, eine Plattform zu schaffen, die eine gemeinsame Nutzung von Anwendungs und Computer-Ressourcen über das Internet im kommerziellen Bereich interessant macht

O G S A

Open — Anbieterunabhängigkeit und Erweiterbarkeit

Grid

Services

Architecture — Wohldefinierte Sammlung von Basis-Schnittstellen

Eine Alternative zu OGSA

...kommt von Sun.

Sie basiert auf:

- der Sun „Grid Engine“, die die ungenutzten Ressourcen, der im Netz eingebundenen Linux und Solaris Rechner einsammelt
- dem „Portal Server“ von Iplanet, der die Web-Service-Schnittstellen WSDL, UDDI, SOAP enthält

Worauf basiert OGSA?

- **Globus Toolkit**

  die wichtigsten Komponenten:

  - GRAM Grid Resource Allocation and Management protocol
  - MDS-2 Meta Directory Service
  - GSI Grid Security Infrastructure
- **Web-Services**

  insbesondere SOAP,WSDL und WS-Inspection

Was ist die Zielsetzung von OGSA?

Einen Standard zu entwickeln, der es ermöglicht Grid-Funktionalitäten als WebService anzubieten.

Der Hauptfokus liegt dabei auf GridServices...

Was ist ein GridService?

Ein GridService ist ein Service,

- der wohldefinierte Schnittstellen anbietet
- der spezielle Konventionen erfüllt
- der Grid Funktionalitäten bereitstellt. [Im Sinne von OGSA]

Zum besseren Verständnis: GridServices verhalten sich ähnlich wie die Objekte aus objektorientierten Programmiersprachen.

Anforderungen an GridServices

[Im Sinne von OGSA]

1. Erzeugung
2. Lifetime Management
3. Service Discovery
4. Notification
5. Platform/Protokollunabhängigkeit
6. Upgradable

1.Erzeugung

- Die GridServices müssen dynamisch erzeugt werden können
- Die konkreten GridServices müssen nach ihrer Erzeugung referenzierbar sein (GridService Instanz)

2.Lifetimemanagement

Da Ressourcen möglichst effektiv ausgenutzt werden sollen, besteht ein Bedarf für Lifetimemanagement

Also: Ressourcen sollen schnellst möglich wieder freigegeben werden, wenn sie von einen GridService nicht benötigt werden

Problem: Woher weiß man das ein GridService nicht mehr benötig wird? $\Rightarrow$ *Reference-Counting??*

3.Service Discovery

- GridServices müssen einfach gefunden werden können
- ihre spezifischen Eigenschaften müssen festgestellt werden können

4.Notification

GridServices müssen die Möglichkeit haben, Meldungen auszutauschen (z.B Fehlermeldungen, Versionsänderungen, usw.)

5.Platform-/Protokollunabhänigkeit

- Damit GridServices eine möglichst große Verbreitung und Akzeptanz bekommen können, müssen sie Platform-/Protokollunabhänigkeit sein

6.Upgradability

Auch GridServices in komplexen, verteilten Systemen müssen aufrüstbar sein, daher gilt

- Versionen und ihre Kompatibilität müssen verwaltet werden

- Services müssen aufrüstbar sein, ohne die Operation der Clients zu trennen

So stellt sich OGSA den Anforderungen

Dazu werden zwei Punkte genauer spezifiziert:

- GridService Aufbau
- Virtual Organization-Struktur

GridService Aufbau (1)

- Die Implementierung eines GridService wird hinter einer WSDL Schnittstelle gekapselt

  → Platformunabhängigkeit
- Jede GridService Instanz muss eindeutig gekennzeichnet werden

  → Grid Service Handle (GSH)

  (keine Instanz-spezifischen Informationen)
- Daher hat jede GridService Instanz eine Referenz

  → Grid Service Reference (GSR)

GridService Aufbau (2)

- GridServices können für den Nachrichtenaustausch zwei Schnittstellen implementieren:

NotificationSource: Dieses Schnittstelle muß vom GridService implementiert werden, der Meldungen mitteilen möchte

NotificationSink: Diese Schnittstelle von dem, der Meldungen empfangen will (asynchronous delivery).

GridService Aufbau (3)

- jeder GridService hat eine festgelegte Anfangslebenszeit, nach deren Ablauf er selbstständig terminiert
- der Client (GridService-Nutzer) kann eine Anfrage auf Verlängerung seiner Lebenszeit stellen
- der Client muß den GridService permanent mitteilen, daß er noch Bedarf an ihn hat

  (keepalive Messages).
- der Anbieter kann eine Anfrage auf Lebenszeit-

  verlängerung ablehnen

GridService Aufbau (4)

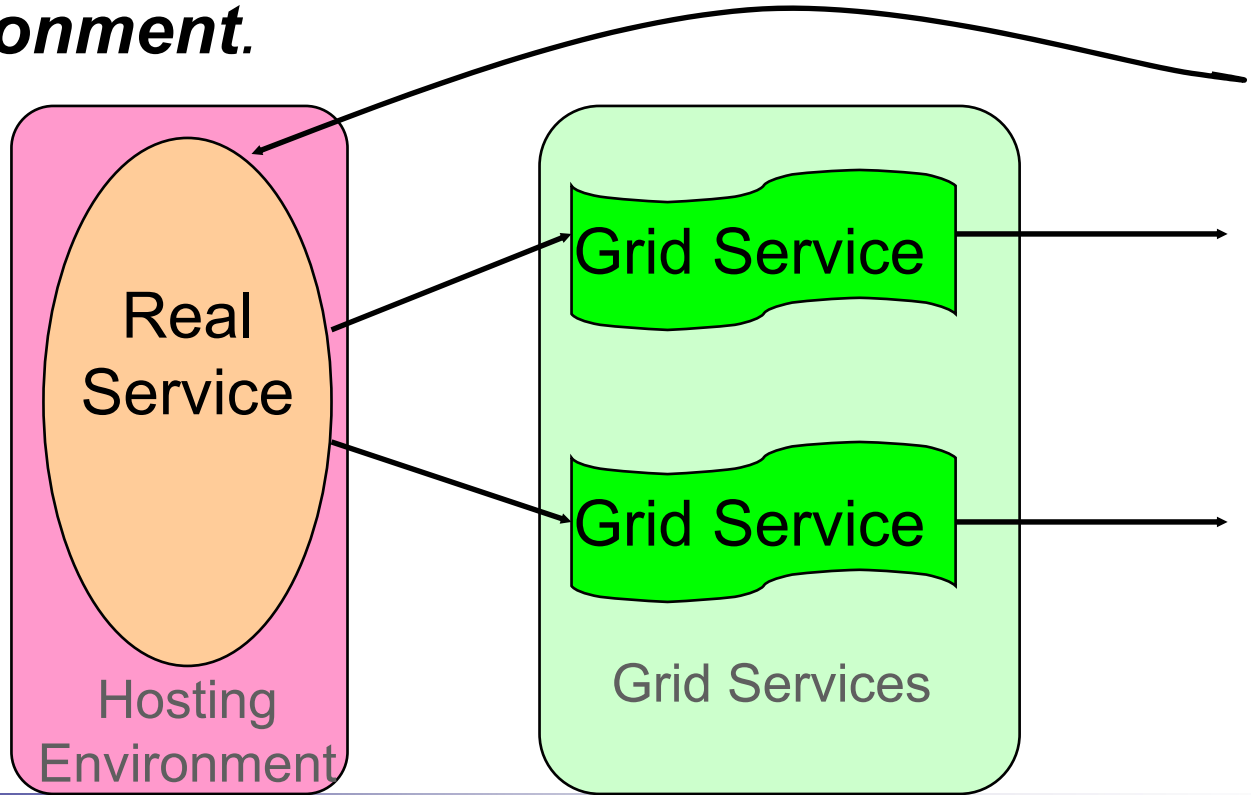
Wichtig: OGSA legt nicht fest, ***wie*** ein GridService implementiert wird

Das wird von der sog. ***Hosting Environment*** bestimmt. Sie besteht aus:

- der Plattform (Linux, NT, usw.) auf der der GridService aufgesetzt wird
- einer konkreten Programmiersprache (C++,Java, usw.) mit der er implementiert wird

GridService Aufbau (5)

- Die Erzeugung eines GridService führt zur Erzeugung eines neuen Prozesses in der ***Hosting Environment***.



So stellt sich OGSA den Anforderungen

Dazu werden zwei Punkte genauer spezifiziert:

- GridService Aufbau
- Virtual Organization-Struktur

Virtual Organization - Struktur (1)

- um die genannten Anforderungen erfüllen zu können, bedarf es mehr als nur einen GridService

- deshalb spezifiert OGSA den Aufbau einer „Virtual Organization” (VO) zur Anbietung von GridServices

- VO ist ein Gebilde, das aus mehreren organisatorischen Einheiten besteht

Virtual Organization - Struktur (2)

- Es gibt 3 vordefinierte Einheiten zum Aufbau einer VO-Struktur :
  - factory
  - registry
  - handleMap
- bieten bestimmte Operationen an
- es sind GridServices, die bestimmte Schnittstellen implementieren

factory

- Der ***factory*** GridService implementiert die ***Factory*** Schnittstelle
- Diese Schnittstelle erzeugt mit der ***CreateService*** Operation die angefordete GridService Instanz und gibt den ***GSH*** zurück
- Sie beschreibt aber nicht **wie** eine Service Instanz erzeugt wird (Hosting Environment)
- Jeder factory service muss sich bei einem registry service registrieren

registry

- Implementiert die Schnittstelle ***Registry***
- Im ***registry*** service werden ***factory*** services, Service Instanzen registriert
- ***registry*** service unterstützt die Auffindung von Services und Service Instanzen und liefert den ***GSH*** zurück

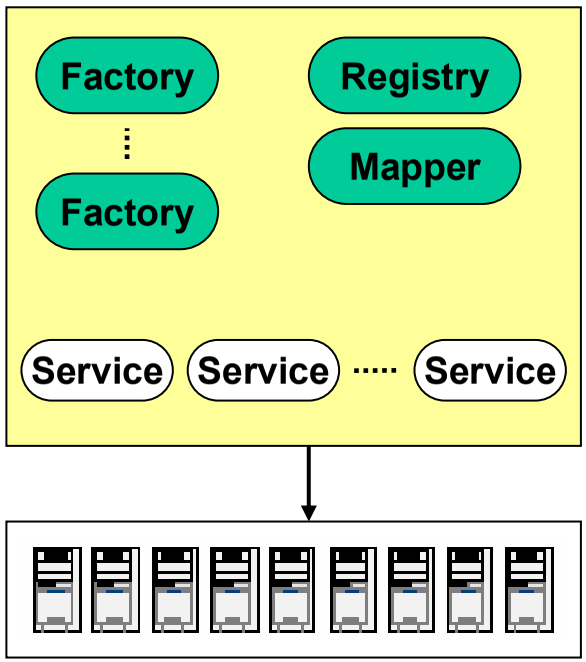
handleMap

- Implementiert die Schnittstelle ***HandleMap***
- Liefert zur einer übergebenen ***GSH*** einen gültigen/aktuellen ***GSR*** zurück

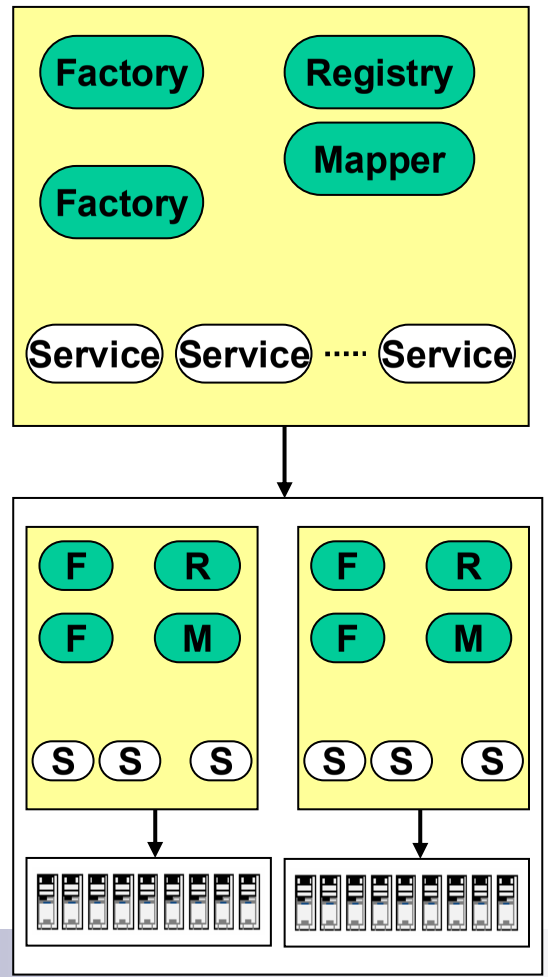
- Jede GridService Instanz hat eine ***homeHandleMap***, über seine GSH bekommt man einen Verweis auf sie

Beispiel: 2 verschieden VO Strukturen

Simple Hosting Environment



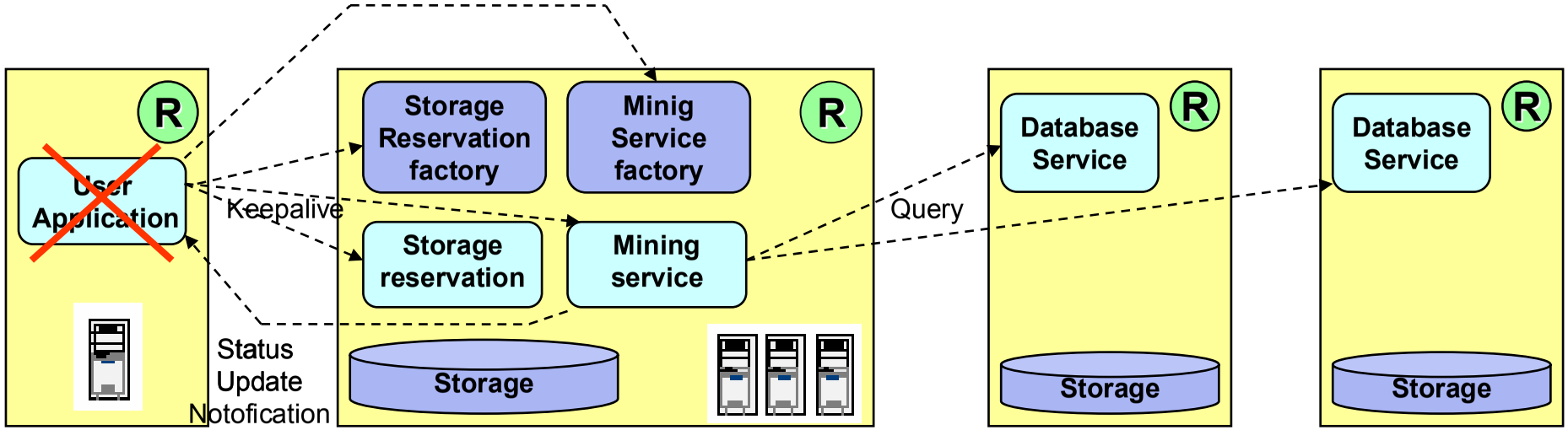
Virtual Hosting Environment



Bespiel für Grid services : Data mining computation

Was ist Data mining?

Data Mining analysiert die Daten der Vergangenheit, um komplexe Zusammenhänge herauszuarbeiten, um daraus dann Informationen für zukünftige Entscheidungen gewinnen zu können.



- Der Client sendet „keepalive“ Nachrichten an die Grid service Instanzen
- Da keine keepalive Nachrichten mehr empfangen werden, werden die 2 Grid services beendet

Literatur-Hinweise

- http://www.globus.org/research/papers/ogsa.pdf
- http://www.globus.org/research/papers/anatomy.pdf
- http://www-1.ibm.com/grid/
- http://www.golem.de/