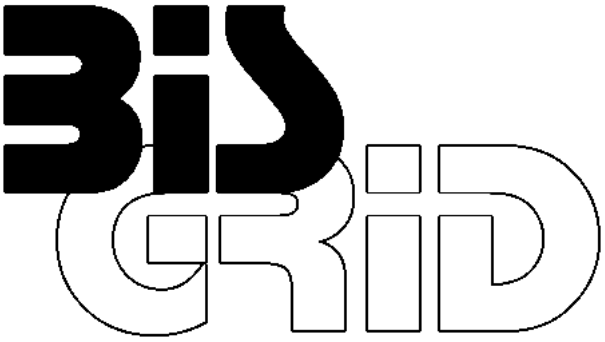

Betriebliche Informationssysteme:
Grid-basierte Integration und Orchestrierung

Deliverable 6.1

Arbeitspaket 6

Projektglossar

30. April , 2010

GEFÖRDERT VOM

Promotional Reference: 01IG07005

Autoren:

Stefan Gudenkauf, **Guido Scherp**
OFFIS Institute for Information Technology
Technology Cluster Software Engineering and Enterprise Architecture
R&D-Division Energy

André Höing
Technische Universität Berlin
Faculty of Information Technologies
Complex and Distributed IT Systems

This work is supported by the German Federal Ministry of Education and Research (BMBF) under grant No. 01IG07005 as part of the D-Grid initiative.

Inhaltsverzeichnis

1 Information

Dieses Dokument repräsentiert Deliverable 6.1 “Projektglossar” des Arbeitspaketes 6 des Projekts BIS-Grid[1], einem vom BMBF geförderten Projekt der deutschen D-Grid Initiative[2]. Inhalt des Dokuments ist die Definition der wichtigsten Begriffe aus dem Projektumfeld der Grid-basierten Integration von Diensten. Bitte beachten Sie, das das Dokument als Gegenstand laufender Arbeit durch höhere Dokumentversionen ersetzt oder erweitert werden kann.

2 Allgemeine Definitionen

Software-Architektur

Eine Software-Architektur ist die Struktur der Komponenten eines Systems, deren Beziehungen zueinander, und die Prinzipien und Richtlinien die ihre Entwicklung im Zeitverlauf steuern (“the structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time”) [GP95, IEE00].

3 Grid Computing

Grid

Als Grid bezeichnet man einen Zusammenschluss verschiedener, organisatorisch und geographisch verteilter Ressourcen, um diese über standardisierte Protokolle und Schnittstellen gemeinschaftlich zu nutzen. Nach Ian Foster muss ein Grid folgende drei Anforderungen erfüllen [FKNT02]:

- Koordination von Ressourcen, die keiner zentralen Kontrolle unterstehen
- Verwendung von standardisierten, offenen und generischen Protokollen und Schnittstellen
- Bereitstellung nicht-trivialer Dienstqualitäten

Zusätzlich besitzen Grids – in Analogie zu Stromnetzen – idealerweise die folgenden drei Eigenschaften:

- Reliability: Ein Grid ist *immer* verfügbar
- Consistency: Der Zugriff auf ein Grid ist überall *gleich*
- Pervasiveness: Ein Grid ist nahezu *überall* verfügbar

[1] http://www.bisgrid.de
[2] http://www.d-grid.de

Resource

Im Grid-Kontext werden alle verfügbaren Systeme als *Resource* bezeichnet. Dazu gehören neben den klassischen Rechner- oder Speichersystemen auch Software oder Sensoren. Im Allgemeinen bezeichnet der Begriff jedoch Betriebsmittel, welche nach folgenden Aspekten klassifiziert werden können:

- Hardware- und/oder Softwarekomponenten
- Exklusive und/oder gleichzeitige Nutzbarkeit
- Einmalige und/oder mehrmalige Benutzbarkeit
- Entziehbarkeit und/oder Nicht-Entziehbarkeit

Grid Middleware

Eine Software-basierte Mediatorschicht, die homogenen Zugriff auf Ressourcen bereit stellt, wobei die Ressourcen selbst lokal mit verschiedenen Zugriffsmethoden verwaltet werden können. Die bekanntesten Vertreter sind das Globus Toolkit[3] und das in BIS-Grid verwendete UNICORE[4], sowie das im EGEE-Kontext[5] eingesetzte gLite[6].

GSI

Die Grid Security Infrastructure (GSI)[7], vormals auch Globus Security Infrastructure genannt, ist die Spezifikation einer vertraulichen, manipulationssicheren und delegierbaren Kommunikation zwischen verschiedener Software innerhalb eines Grids, was über eine so genannte *public key infrastructure* (PKI) sichergestellt wird.

Grid Service

Nach OGSA (s. u.) ist ein Grid Service ein Web Service, der eine Menge wohldefinierter Schnittstellen bereitstellt und besonderen Konventionen folgt. Diese Schnittstellen adressieren die folgenden Punkte:

- Auffinden von Diensten (discovery)
- Dynamische Diensteerstellung (dynamic service creation)
- Lebenszeit-Management (lifetime management)
- Benachrichtigungen (notification)

[3] http://www.globus.org/
[4] http://www.unicore.eu/
[5] http://public.eu-egee.org/
[6] http://glite.web.cern.ch/glite/
[7] http://www.globus.org/security/overview.html

- Handhabbarkeit (manageability)

Die für Grid Services geltenden Konventionen betreffen die folenden Punkte:

- Namensvergabe (naming)
- Erweiterbarkeit (upgradeability)
- Zukünftige Autorisierung (authorization)
- Nebenläufigkeitskontrolle (concurrency control)

Im allgemeinen Sprachgebrauch der Grid-Welt bezeichnen Grid Services zustandbasierte Web Services, die von oder über Grid Middlewares bereit gestellt werden, worunter auch WSRF-konforme Web Services fallen.

Grid Service Protocol Binding

Die Aspekte Authentisierung (authentification) und vertrauenswürdige Dienstaufrufe (reliable invocation) werden von OGSA als Dienst-Protokoll-Bindung (service protocol binding) angesehen und sind explizit nicht Teil der Definition eines Grid Service [FKNT02].

OGSA

Die Open Grid Services Architecture (OGSA)[8] spezifiziert die Anforderungen an die Architektur eines Grid, die sich an der Veröffentlichung "The Physiology of the Grid" [FKNT02] orientieren. Eine Besonderheit in OGSA ist, dass alle Dienste, Elemente etc. als Grid Services aufgefasst werden. Neben der Unterstützung verschiedener Dienste (Infrastructure services, Execution Management services, Data services, Resource Management services, Security services, Self-management services und Information services), muss jede Ressource bzw. jeder Dienst als ein sogenannter Grid Service, ein zustandsbehafteter(Web) Service unter der Einhaltung bestimmter Regeln, angeboten werden.

OGSA wurde 2002 auf dem Treffen des Global Grid Forum (GGF, mittlerweile Open Grid Forum (OGF))[9], einer globalen Grid-Initiative, definiert. Zukünftige Standards im Bereich Grid sollen OGSA-konform sein. Die Entwicklung der OGSA wird bis heute fortgesetzt, wobei beispielsweise die Schnittstellen bestimmter Dienste weiter spezifiziert werden. Auf die OGSA aufbauende Standards werden durch diese Entwicklung immer wieder beeinflusst.

OGSI

Die Open Grid Services Infrastructure (OGSI) stellt den ersten OGSA-konformen Standard für Grid Services aus dem Jahr 2003 dar, der in Globus Toolkit 3 implementiert wurde. Allerdings ist dieser Standard nicht Web Services-konform, weil die verwendeten Web Services-Schnittstellen auf WSDL 2.0 aufbauen, der noch nicht als Standard

[8] http://www.globus.org/ogsa/
[9] http://www.ogf.org/

verabschiedet wurde und für den es auch noch keine Werkzeugunterstützung gibt. Des weiteren wird die OGSI auch als zu objektorientiert kritisiert. OGSI bildet nicht die gesamte OGSA-Architektur ab, sondern lediglich die Forderung nach Grid Services. Globus Toolkit 3 bietet aber nahezu alle gefordeten Dienste an.

WSRF

Um die Schwachstellen von OGSI auszugleichen, wurde mit dem Web Services Resource Framework (WSRF)[Ban06] ein weiterer OGSA-konformer Standard entwickelt. Als Referenzimplementierung wurde das Globus Toolkit 4 entwickelt. Der Standard beinhaltet mehrere Web Services-konforme Schnittstellen, die von einer Ressource implementiert werde müssen, um als Grid Service angeboten werden zu können. Dabei werden WSRF-konforme Dienste als generell zustandsbasiert definiert. Analog zu OGSI deckt WSRF nur den OGSA-Teil der Grid Services ab, die meisten geforderten Dienste werden aber auch in Globus Toolkit 4 implementiert. Der WSRF-Standard umfasst die folgenden Spezifikationen:

- *WS-Resource* definiert eine WS-Resource als die Komposition einer Ressource mit einem Web Service, über den auf die Ressource zugegriffen werden kann.
- *WS-ResourceProperties* definiert eine Schnittstelle, um eine Menge typisierter Werte mit einer WS-Ressource zu verbinden, die über die Schnittstelle in standardisierter Weise gelesen und manipuliert werden können.
- *WS-ResourceLifetime* definiert eine Schnittstelle zur Verwaltung des Lebenszyklus einer WS-Resource.
- *WS-BaseFaults* definiert einen Erweiterungsmechanismus für auf höherer Ebene semantisch bedeutungsvolle SOAPFaults.

Shibboleth

Das im Shibboleth-Projekt[10] entwickelte Shibboleth-System ist eine standardbasierte Open-Source-Software für das Web-basierte Single-Sign-On (SSO) über organisatorische Grenzen hinweg. Das System ermöglicht, automatisiert Autorisationsenscheidungen für anfragende Individuen zu treffen, ohne die Datensicherheit der betreffenden Online-Ressource (z. B. ein Internetportal) zu gefährden. Shibboleth sieht dabei die Unterstützung des SAML-Standards[RHP+07] als Anforderung an, so dass die Unterstützung von SAML V2.0 als Nahziel auf der Entwicklungs-Roadmap von Shibboleth angekündigt ist. Das Shibboleth-Projekt ist eine Initiative des Internet2-Konsortiums[11]. Ziel des Projekts ist die Entwicklung Policy-basierter Open-Source-Systeme zur Kontrolle des Zugriffs auf Online-Ressourcen.

[10] http://shibboleth.internet2.edu/
[11] http://www.internet2.edu/

GridShib

GridShib[12] ist ein System, um eine verteilte Autorisierungsinfrastruktur auf Basis des Shibboleth-Systems mit Grid-Technologien wie dem Globus Toolkit zu integrieren. Das GridShib-System fungiert in einem Shibboleth-System als Service-Provider und führt so eine Authentifizierung/Autorisierung auf Basis des Shibboleth-Systems durch. Nach erfolgreicher Authentifizierung wird ein bis zu circa 11 Tage (1 Million Sekunden) gültiges (kurzlebiges) Zertifikat (short-lived certificate, SLC) generiert, durch das ein Zugriff auf Grid-Resourcen möglich ist. Das GridShib-Projekt ist eine Kooperation zwischen dem National Center for Supercomputing Applications (NCSA)[13] der Universität Illinois und der Universität von Chicago, und wird durch die National Science Foundation (NSF)[14] gefördert.

4 Cloud Computing

Cloud

Basierend auf dem SOA-Gedanken und der rapiden Entwicklung von Internet-Technologien in den letzten Jahren sind verschiedene Dienstanbieter entstanden, die spezielle Dienste anbieten, die gemeinhin als Cloud bekannt geworden sind. Eine Cloud stellt Dienste als so genannte Services bedarfsgesteuert (on demand) und auf Basis eines Geschäftsmodells an, das diese Dienste nach der tatsächlichen Nutzung abrechnet (pay-per-use). Da die von Clouds erbrachten Dienste sehr verschieden sein können, hat sich eine Klassifizierung nach dem Muster "*-as-a-Service" durchgesetzt, wobei derzeit ein Konsenz auf die drei Kategorien *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) und *Software as a Service* (SaaS) [VRMCL09] zu erkennen ist. Die einzelnen Kategorien werden dabei als aufeinander aufbauende Abstraktionen der tatsächlichen ausführenden Infrastruktur angesehen.

IaaS

Infrastructure as a Service (IaaS) bezeichnet den Service-basierten Zugriff auf (häufig virtualisierten) Rechner-Ressourcen wie Speicherplatz und Rechenzeit.

PaaS

Platform as a Service (PaaS) bezeichnet den Service-basierten Zugriff auf eine Software-Plattform, die dem Kunden die maßgeschneiderte Entwicklung skalierbarer Anwendungen ermöglicht. Diese werden dabei häufig in virtualisierten Intfrastrukturen betrieben.

[12] http://gridshib.globus.org/
[13] http://www.ncsa.illinois.edu/
[14] http://www.nsf.gov/

SaaS

Software as a Service (SaaS) bezeichnet den Service-basierten Zugriff auf ein spezifisches Software-Produkt (z. B. ERP-Software) oder eine bestimmte Funktionalität (z. B. Prüfung auf Kreditwürdigkeit).

5 Service-orientierte Architekturen

SOA

Eine Service-orientierte Architektur (SOA) bezeichnet ein Architekturkonzept für Anwendungslandschaften auf der obersten Abstraktionsebene, das maßgeblich auf dem Service-Konzept, dem Service-basierten Zusammenbringen von Geschäftswelt und IT (Business-IT-Alignment), und der Referenzarchitekturbildung basiert[RH08].

Service

Laut [RH08] ist ein *Service* ein Element geschäftlichen Verhaltens. Er stellt eine geschäftliche Leistung dar, die ein Service-Anbieter gegenüber Dienstnehmern erbringt. Jedem Service liegt dabei ein Vertrag zu Grunde. Dieser legt die ein- und ausgehenden Informationen und Güter fest. Services oder Teile von Services können mittels IT durch die Anwendungslandschaft erbracht werden. In der Grid-Welt wird der Begriff Service auch als Netzwerk-aktive Entität definiert, die eine gewisse Leistung (capability) durch den Austausch von Nachrichten anbietet[FKNT02].

Service Level Agreement

Ein Service Level Agreement (SLA) ist eine *"formale, schriftlich dokumentierte, für einen bestimmten Zeitraum abgeschlossene Vereinbarung zwischen einem Leistungsnehmer (Kunde) und einem Leistungsanbieter (Dienstleister), in der der Leistungsnehmer die Erbringung gewisser inhaltlich und qualitativ definierter Dienstleistungen und der Kunde hierfür die Leistung definierter finanzieller Ausgleichszahlungen zusagt"* [Ber05]. Dabei erfolgt die Festlegung der durch den Dienstleister zu erbringende Qualität der Dienstleistung durch die Vereinbarung von einzuhaltenden *Service Levels*. Diese werden durch gemeinsam definierte und quantifizierbare Dienstleistungsmerkmale in Form von Kennzahlen definiert. Ferner definiert ein SLA *"Verfahren, die den Nachweis der Einhaltung der Service-Levels regeln, sowie Konsequenzen für den Fall der Abweichung von den vereinbarten Service-Levels"* [Ber05]. Folgende Merkmale sind charakteristisch für SLAs (vgl. [Ber05]):

- Formalität. Eine SLA ist formal strukturiert und schriftlich dokumentiert.
- Bilateralität. Eine SLA wird zwischen zwei Parteien geschlossen.
- Dienstleistungsbezug. Eine SLA bezieht sich auf eine Dienstleistung als Wirtschaftsgut.

- Verpflichtungsbezug. Eine SLA beinhaltet die Verpflichtung der Erbringung einer Dienstleistung eines Partners (Dienstleister) und die Verpflichtung der Erbringung einer Gegenleistung eines anderen Partners (Kunde).

- Zeitbezug. Eine SLA bezieht sich stehts auf einen bestimmten Zeitraum.

- Inhaltsbezug. Eine SLA beinhaltet die inhaltliche Beschreibung der zu erbringenden Dienstleistung sowie die Beschreibung von Regelprozessen zur Dienstleistungserbringung.

- Qualitätzbezug. Eine SLA definiert die Qualität der zu erbringenden Dienstleistung. Dies geschieht durch die Quantifizierung relevanter Merkmale der Dienstleistung mit Hilfe der Definition von *Kennzahlen*[15] und so genannten *Service Levels*[16].

- Preisbezug. Eine SLA nennt die Verrechnungspreise der Dienstleistung (z. B. auch Kompensationszahlungen) in Bezug auf die definierten Service Levels.

- Abweichungsregelung. Eine SLA enthält Regelungen für Abweichungen von vereinbarten Service Levels (z. B. Konventionalstrafen).

Generell kann zwischen SLAs im engeren Sinne und SLAs im weiteren Sinne unterschieden werden. Erstere definieren ausschließlich die zu erbringende Dienstleistungsqualität, letztere neben der Qualität der Dienstleistung ebenfalls den konkreten Inhalt und die Kosten der Dienstleistung. Eine weitere Klassifizierung kann nach der Art/Beschaffenheit der Beziehungen der beiden Partner eines SLA vorgenommen werden. Bzgl. der rechtlichen wirtschaftlichen Beziehungen kann unterschieden werden, ob Dienstleister und Partner ein und derselben Organisation angehören oder ob sie organisatorisch und wirtschaftlich unabhängig sind. Im ersten Falle spricht man von *internen SLAs*, in letzterem von *externen SLAs*. Weiterhin können die internen SLAs in *unternehmensinterne und konzerninterne SLAs* unterschieden werden, je nachdem, ob auch innerhalb der Organisationsform bei den internen SLAs wirtschaftliche oder rechtliche Aspekte bestehen.

Ein weiteres Unterscheidungsmerkmal ist der Grad des Anwendungsbezugs des Leistungsnehmers in der SLA. Danach lassen sich SLAs in anwendungsbezogene SLAs und technische SLAs unterscheiden. *Anwendungsbezogene SLAs* gehen dabei auf die Belange von Endanwendern ein. Diese betrachten Vereinbarungen, die den direkten fachlichen Nutzen der IT-Dienstleistung für deren betriebliche Aufgabe zum Gegenstand haben. *Technische SLAs* beziehen sich dagegen ausschließlich auf spezifische technische Anforderungen und setzen insbesondere auf Seiten des Leistungsnehmers technisches Verständnis voraus.

[15] *"Eine Kennzahl bezeichnet eine in numerischer Weise ausgedrückte Information über einen bestimmten quantifizierbaren Tatbestand"* [Ber05].

[16] "Ein Service-Level stellt den Wert einer bestimmten, in einem SLA vereinbarten Kennzahl zur Beurteilung der Qualität einer Dienstleistung dar"[Ber05].

ESB

Ein Enterprise Service Bus (ESB) ist eine konkrete Umsetzung einer SOA und deren Betrieb[RH08]. Ein ESB gewährleistet hierbei primär die Kommunikation zwischen den Services, sowie weitere Funktionalitäten für das Management und den Betrieb der SOA.

6 Workflows

Geschäftsprozess

Ein Geschäftsprozess besteht aus einer oder mehreren verbundenen Prozeduren oder Aktivitäten, die kollektiv und innerhalb eines bestimmten organisationalen Kontexts ein Geschäftsziel oder Richtlinie realisieren, wobei der organisationale Kontext die möglichen beteiligten Rollen und deren Verhältnisse zueinander definiert[WfM99].

Workflow

Ein Workflow ist die vollständige oder teilweise Automatisierung eines Geschäftsprozesses, während dessen Ausführung Dokumente, Informationen und/oder Aufgaben zwischen den Beteiligten des Workflows anhand einer Menge von prozeduralen Regeln zu deren Bearbeitung hin- und hergereicht werden[WfM99].

Workflow Management System

Ein Workflow Management System (WfMS) definiert, erstellt und verwaltet die Ausführung von Workflows über geeignete Software, die die Beschreibung von Workflows in so genannten Prozessdefinitionen interpretieren, mit den Workflow-Beteiligten interagieren, und bei Bedarf weitere IT-Werkzeuge und Anwendungen starten kann[WfM99].

Workflow Engine

Eine Workflow Engine ist die Laufzeit-Ausführungsumgebung für Prozessinstanzen. Eine Workflow Engine ist häufig Kernbestandteil eines Workflow Management Systems.

Prozess

Ein Prozess ist eine als geordnete und koordinierte Menge von Aktivitäten formalisierte Sicht auf einen Geschäftsprozess, wobei die Aktivitäten wiederum als untergeordnete Prozesse formalisiert sein können.

Prozessdefinition

Eine Prozessdefinition ist die Repräsentation eines Geschäftsprozesses in einer von einem Workflow Management System ausführbaren Form, vgl. [WfM99].

Prozessinstanz

Eine Prozessinstanz ist die Repräsentation einer einzigen Inkraftsetzung oder Durchführung einer Prozessdefinition, vgl. [WfM99]. Der Begriff ist im Kontext des Projektes synonym zu dem Begriff Workflow-Instanz.

Workflow-Beteiligter

Ein Workflow-Beteiligter ist eine Resource (z. B. Service, Grid Service oder eine Person), die zur Ausführung einer bestimmten Aktivität als Teil eines Workflows vorgesehen ist.

BPMN

Die Business Process Modeling Notation (BPMN)[17] ist eine graphische Spezifikationssprache zur Modellierung von Geschäftsprozessen. Die BPMN wurde 2002 von Stephen A. White (IBM) erarbeitet und durch die Business Process Management Initiative (BPMI)[18] veröffentlicht. Sie wurde im Juni 2005 durch die Object Management Group (OMG)[19] zur weiteren Pflege übernommen. Seit 2006 ist BPMN ein offzieller OMG-Standard. Der Schwerpunkt der BPMN liegt auf der Notation (graphische Darstellung) von Geschäftsprozessen. Teil des BPMN-Standards ist ein Vorschlag, wie BPMN-Diagramme in die Sprache WS-BPEL abgebildet werden könnten.

WS-BPEL

Die *Web Service Business Process Execution Language* (WS-BPEL [OAS07]) ist ein XML-basierter OASIS-Standard zur programmatischen Beschreibung von Geschäftsprozessen, deren Aktivitäten als externe Dienste, so genannte (Web) Services, implementiert sind. Daher kann WS-BPEL zur so genannten Service-Orchestrierung eingesetzt werden. Die von einem derartigen Service bereit gestellten Funktionen werden durch eine Schnittstelle beschrieben, die mit Hilfe der *Web Services Description Language* (WSDL) definiert sind. Mit WS-BPEL definierte Geschäftsprozesse werden von entsprechenden Workflow Engines interpretiert und ausgeführt.

Der WS-BPEL-Standard kann als de-facto-Standard für die Integration betrieblicher Informationssysteme betrachet werden, und genießt breite Unterstützung von Werkzeugherstellern und einsetzenden Unternehmen. Die Verwendung von WS-BPEL geht häufig mit der Verwendung weiterer existierender Standards einher, z. B. XML Schema [W3C], WS-Addressing [Web] und XPATH [W3C07]. Ursprünglich wurde WS-BPEL als BPEL4WS (Business Process Execution Language for Web Services) [BPE06] im Jahr 2002 von den Unternehmen IBM, BEA Systems, Microsoft, SAP AG, und Siebel Systems eingeführt. Zur Implementierung von Geschäftsprozessen stellt der WS-BPEL-Standard verschiedene Elemente bereit. Es existieren z. B.

[17] http://www.bpmn.org/
[18] http://www.bpmi.org/
[19] http://www.omg.org/

- Kontrollflusselemente (Sequenzen, Schleifen, parallele Ausführungsstränge etc.)
- Elemente zur Datenmanipulationen (es werden XPATH und XSLT unterstützt)
- Elemente zur Fehler- und Ausnahmebehandlung, Kompensation und transaktionalem Verhalten

In einer Geschäftsprozessmodellierung mit WS-BPEL nur die abstrakten Teile einer WSDL-Schnittstelle verwendet, um den Nachrichtenaustausch der Services zu beschreiben. Die tatsächliche Kopplung der abstrakten Schnittstellen an konkrete, bereitgestellte Dienste über so genannte *service endpoints* ist die Aufgabe der WS-BPEL-Ausführungsumgebung. Generell wird dabei zwischen der statischen und der dynamischen Bindung unterschieden, wobei bei der dynamischen Bindung die service endpoints durch WS-Addressing-Konstrukte beschrieben werden. Abbildung 1 zeigt einen Überblick über die kommunikationsspezifischen Elemente von WS-BPEL. Das Element `partnerLinkType` ist eine WSDL-Erweiterung, die eine Kommunikationsbeziehung zwischen zwei Partnern (Services) darstellt. Dazu gibt das `partnerLinkType`-Element alle WSDL Porttypen an, die an der Kommunikation teilnehmen. Dabei ist jedem `portType` eine Rolle zugeordnet. Ein `partnerLink` kann als Instanz eines `partnerLinkType` angesehen werden und legt fest, welche Rolle der eigene Geschäftsprozess und welche der externe Partner besitzt, d. h. welcher Partner welche Web Service-Schnittstellen zu Verfügung stellen muss. Dabei weist ein `partnerLink` bei einer asynchronen Kommunikation dem Geschäftsprozess eine Rolle zu (Attribut *myRole*) und dem externen Partner ebenfalls eine Rolle (Attribut *partnerRole*). Im Falle der synchronen Kommunikation wird dagegen nur jeweils eine Rolle im `partnerLink` spezifiziert.

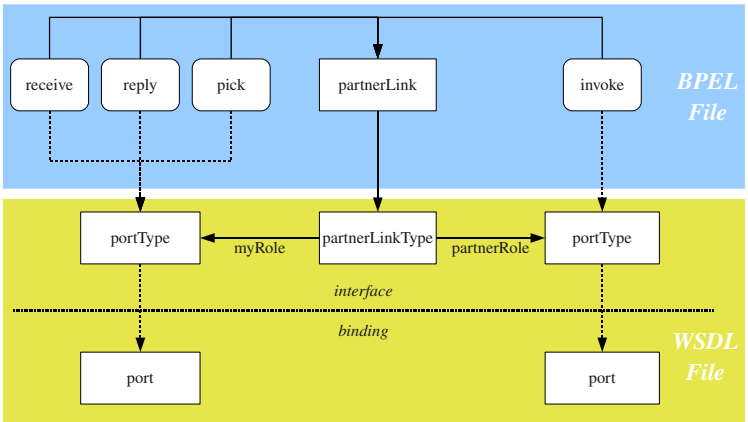

Abbildung 1: Kommunikationselemente in WS-BPEL.

WSDL

Die WSDL (Web Services Description Language) ist eine plattform-, programmiersprachen- und protokollunabhängige XML-Spezifikation zur Beschreibung von Netzwerkdiensten (Web Services) für den Nachrichtenaustausch. WSDL beschreibt die angebotenen Funktionen, Daten, Datentypen und Austauschprotokolle eines Web Service. Ein WSDL-Dokument enthält die folgenden funktionellen Angaben:

- Schnittstelle des Dienstes
- Zugangsprotokoll
- Details zum Deployment
- Allen notwendigen Informationen zum Zugriff auf den beschriebenen Dienst

WSDL benutzt dabei die folgenden Hauptelemente zur Beschreibung eines Dienstes:

- Datentypen (types)
- Nachrichten (message)
- Port-Typen (portType), unterteilt in die Typen *One-way*, *Request-response*, *Solicit-response* und *Notification*
- Bindung (binding)
- Ports (port)
- Services (service)

Endpoint Reference

Über eine Endpoint Reference (EPR) wird der Zugriffspunkt (z. B. URL) für einen Web Service beschrieben. Dafür gibt es den XML-Standard WS-Addressing, was z. B. auch in WSRF genutzt wird, um eine Ressource eindeutig zu identifizieren.

SOAP

SOAP ist ein leichtgewichtiges Netzwerkprotokoll zum Austausch XML-basierter Nachrichten und für Remote Procedure Calls (RPC) über ein Computernetzwerk. Es stellt Regeln für das Nachrichtendesign auf, regelt, wie Daten in der Nachricht abzubilden und zu interpretieren sind, und gibt eine Konvention für entfernte Prozeduraufrufe mittels SOAP-Nachrichten vor. SOAP macht keine Vorschriften zur Semantik applikationsspezifischer Daten, sondern stellt ein Rahmenwerk zur Verfügung, welches es erlaubt, dass beliebige applikationsspezifische Informationen übertragen werden können.

SOAP stützt sich auf die Dienste anderer Standards: XML zur Repräsentation der Daten und Internet-Protokolle der Transport- und Anwendungsschicht (vgl. TCP/IP-Referenzmodell) zur Übertragung der Nachrichten. Die gängigste Kombination ist SOAP über HTTP und TCP. SOAP hat den Status einer W3C-Empfehlung.

7 UNICORE

UNICORE

UNICORE[20] (UNiform Interface to COmputing REsources) ist eine Grid Middleware zum nahtlosen und sicheren Zugriff auf verteilte Grid Ressourcen wie Supercomputer, Cluster-Systeme und Datenbanken. UNICORE wurde ursprünglich in zwei vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekten geschaffen und wurde in verschiedenen von der Europäischen Union geförderten Projekten zu einer ausgewachsenen und wohlgetesteten Grid Middleware weiterentwickelt. UNICORE wird produktiv in verschiedenen Supercomputer-Einrichtungen weltweit eingesetzt und dient als Basis verschiedener europäischer und internationaler Foprschungsprojekte. UNICORE ist Open Source unter einer BSD-Lizenz auf SourceForge verfügbar und wird in regelmäßigen Abständen aktualisiert.

UVOS

UNICORE VO Service[21] (UVOS) ist ein Client-Server System zur Unterstützung großer verteilter Systeme – insbesondere solcher die auf UNICORE aufbauen – durch die Verwaltung so genannter Virtueller Organisationen. Die grundlegenden Funktionalitäten von UVOS sind das Vorhalten der Identitäten von Grid-Nutzern und -Ressourcen, das Gruppieren und Attributieren solcher Identitäten, die Abwicklung von Registrierungsanfragen, sowie die Authentifizierung von Web-Browser-basierten Grid Clients. Das UVOS-System basiert auf verbreiteten Standards wie z. B.

- SAML Attribute Query Deployment Profile for X.509 Subjects
- SAML Attribute Self-Query Deployment Profile for X.509 Subjects
- OGSA Attribute Exchange Profile Version 1.2
- XACML Attribute Profile

USite

Eine USite ist eine Gruppe von Ressourcen incl. dem Unicore Gateway als so genannter Single Point of Entry.

VSite

Eine VSite ist eine einzelne Ressource hinter einem Unicore Gateway. Mehrere USites können zu einer VSite zusammengefasst werden.

[20] http://www.unicore.eu/
[21] http://uvos.chemomentum.org/

SAML

Die Security Assertion Markup Language (SAML) ist in der Version 2.0 von der OASIS als Standard angenommen worden und wurde von der Liberty Alliance, einem Zusammenschluss von über 130 Firmen, übernommen. SAML ist ein auf XML aufsetzender Standard zum Austausch von Autorisierung-, Authentisierungs- und Attributinformationen zwischen verschiedenen Sicherheitsbereichen (*Security Domains*), z. B. verschiedenen Firmen. Dabei werden Sicherheitsinformationen über einen Nutzer (*Principle*) in Form von *Assertions* zwischen einem *Identity Provider* (IdP , Aussteller von "Assertions") und einem *Service Provider* (SP, Nutzer von Assertions) über Web Service-Schnittstellen ausgetauscht. Das SAML-Protokoll und die SAML-*Bindings* definieren, wie einzelne SAML-Elemente in Anfragen (*request*) und Antworten (*response*) verpackt werden.

Der SAML-Standard beinhaltet SAML-Profile für bestimmte Anwendungsszenarien, wie z. B. für die einmalige Anmeldung über einen Web Browser (Single-Sign-On, *Web Browser SSO Profile*). In diesen SAML-Profilen ist detailliert das Zusammenspiel zwischen Identity Provider, Service Provider und Principle, sowie den benötigten Teilen des SAML Standards beschrieben. Es gibt einige Annahmen die dem SAML Standard zu Grunde liegen.

- Es existiert ein Vertrauensverhältnis zwischen dem (einem oder mehreren) Identity Provider und allen Service Provider. Dieses Vertrauensverhältnis wird meist technisch durch eine Public-Key-Infrastruktur (PKI) umgesetzt, in der über Zertifikate Nachrichten signiert und/oder verschlüsselt werden.
- Ein Principle ist bei mindestens einem Identity Provider registriert und der Identity Provider garantiert die Identität eines Principle. Bei einem menschlichen Principle (Nutzer) wird dessen Identität vor der Registrierung mittels des Lichtbildausweises überprüft.
- Der Identity Provider bietet dem Principle eine Schnittstelle zur Authentifizierung, wobei der SAML-Standard die Art der Authentifizierung nicht vorschreibt. Diese kann über Zertifikate, Username/Passwort, eToken, usw. erfolgen.
- Der Service Provider trifft seine Zugriffsentscheidung für eine Resource/Dienstleistung aufgrund einer SAML-Assertion, die durch den Principle beim Identity Provider veranlasst wird.

XACML

Die eXtensible Access Control Markup Language (XACML) liegt in der Version 2.0 vor und ist durch die OASIS standardisiert. XACML ist ein feingranulares Zugriffskontrollsystem für Subjekte, die ihre Identität in XML ausdrücken. Dabei können die Subjekte sehr unterschiedlich sein, z. B. so genannte *Principles* wie sie in SAML definiert sind. Der XACML-Standard beschreibt ein XML-Schema und einen Namensraum, mit dem die Zugriffsrechte als *Policies* modelliert werden können. Alle Policies sind nach dem folgenden Muster modelliert: ein Subjekt möchte eine Aktion auf einer Ressource unter

bestimmten Bedingungen durchführen. Um solche Policies durchzusetzen benötigt man zwei Komponenten, einen *Policy Enforcement Point* (PEP) und einen *Policy Decison Point* (PDP).

Die beiden Standards XACML V2.0 und SAML V2.0 wurden dahingehend entwickelt, sich gegenseitig zu ergänzen. XACML-basierte Attribute können z. B. in SAML beschrieben werden. So ist es möglich, dass mittels XACML-Policies spezifiziert wird, was beim Eintreffen einer SAML-Assertion bei einem Provider geschehen soll.

8 BIS-Grid

BIS-Grid Workflow Engine

Die BIS-Grid Workflow Engine ist eine Workflow Engine, die auf dem industriellen de facto-Standard für Dienstorchestration, WS-BPEL, basiert und in der Lage ist, sowohl herkömmliche Web Services als auch Grid und Cloud Services zu orchestrieren. Sie basiert maßgeblich auf Diensterweiterungen der Grid-Middleware UNICORE 6 und einer frei wählbaren, herkömmlichen WS-BPEL-Engine. Dienstorchestrierungen werden dabei von der BIS-Grid Workflow Engine als (zustandsbehaftete) Grid Services angeboten. Die BIS-Grid Workflow Engine besteht aus zwei Komponenten: Der Grid-Middleware UNICORE 6 und einer frei wählbaren WS-BPEL-Engine im Backend, die nur über die UNICORE 6-Middleware zu erreichen ist (cp. [GHH+08]). Dabei überbrückt die UNICORE-Schicht die technische Kluft zwischen Grid-Umgebungen und WS-BPEL. Jede Nachricht, die zwischen der WS-BPEL-Egine und einem externen Dienst oder Endklienten ausgetauscht wird, muss die UNICORE-Schicht passieren. Dort werden z. B. Nachrichten um Sicherheitszeugnisse (sog. *credentials*) wie etwa SAML Assertions ergänzt[22]. Durch die Auslagerung von WS-BPEL außerhalb der UNICORE-Middleware als Frontend kann diese separat deployt werden, um beispielsweise Lastbalancierung zu unterstützen[23].
Die UNICORE-Schicht der BIS-Grid-Engine basiert auf Diensterweiterungen für den Service-Container von UNICORE 6 (vgl. Abbildung 2). Diese bestehen im Wesentlichen aus einem *Workflow Management Service* und einem generischen *Workflow Service*. Der Workflow Management Service stellt Funktionalitäten wie Prozess-Deployment, -Redeployment und -Undeployment bereit. Während des Deployments eines Prozesses erstellt der Workflow Management Service eine neue spezielle Instanz des Workflow Service für den betreffenden Prozess. Unter anderem bietet dieser Workflow Service dann die vollständige Web Service-Schnittstelle des originalen WS-BPEL-Prozesses an. Somit besitzt jeder über die UNICORE-Schicht in der WS-BPEL-Engine deployter WS-BPEL-Prozess eine zugehörige Instanz des Workflow Service. Ein *Proxy Service* fängt dabei alle Nachrichten ab, die von einer Prozessinstanz in der WS-BPEL-Engine gesendet werden und übermittelt sie an die zugehörige Workflow Service-Instanz zur weiteren Bearbeitung, bevor sie an den eigentlichen Empfänger versendet wird. Weitere Informationen

[22] SAML Assertions werden derzeit nicht in WS-BPEL unterstützt

[23] Weitere Details zum Thema Lastbalancierung mit der BIS-Grid Workflow Engine finden sich in [GHH+08]

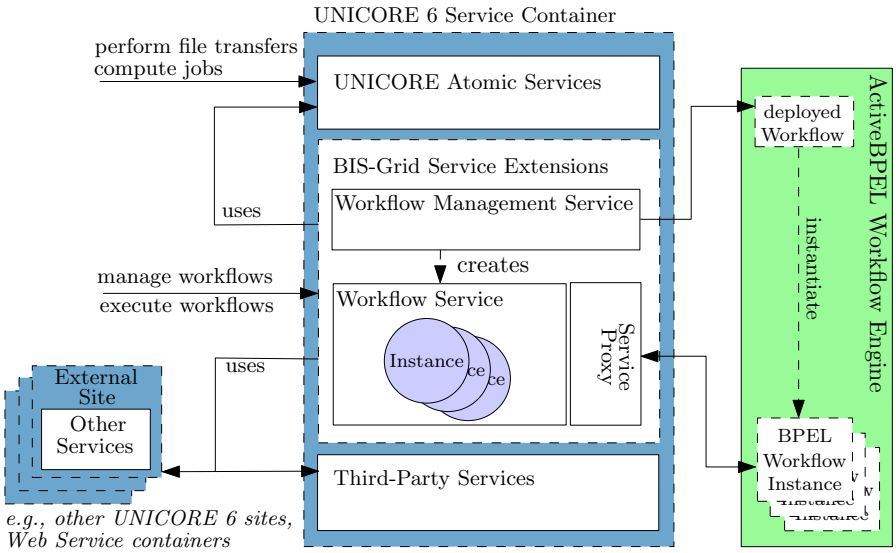

Abbildung 2: Architektur der BIS-Grid-Engine

zur BIS-Grid Workflow Engine finden sich in Deliverable 3.1, der Spezifikation der BIS-Grid Workflow Engine [HHG+07] und Deliverable 3.2, der zugehörigen Dokumentation [HGS08].

OaaS

Orchestration as a Service (OaaS) ist eine innerhalb des BIS-Grid Projekts entwickelte bestimmte Ausprägung von PaaS, die den Betrieb einer Dienstorchestrierungsplattform, typischerweise realisiert als Workflow Engine oder als ausgewachsenes Workflow Management System, durch einen externen Provider vorsieht, siehe [HSG+09]. Als Vorteile (+) und Nachteile (-) von OaaS werden die folgenden Punkte angesehen:

+ Pay-per-Workflow Kostenmodell möglich

+ Kombinierbare Kostenmodelle, z. B. Pay-per-Workflow aufbauend auf Grundanforderungen wie Verfügbarkeit, Ausfallsicherheit und Einhaltung vereinbarter Antwortzeiten

+ Betrieb und Wartung der Orchestrierungslösung durch die IT-Experten des Provider

+ Hohe Verfügbarkeit

+ Grundlage zur Auslagerung weiterer IT-Dienste

+ Skalierbare Dienstinanspruchnahme durch Virtualisierung, z. B. Umzug der Orchestrierungslösung auf dedizierte Knoten

- Notwendigkeit der Datensicherheit und Datenübertragungssicherheit
- Daten verlassen die unternehmenseigene administrative Domäne
- Unsichere Datenübertragung über das Internet
- Vertrauensbeziehung zum Provider als Grundvoraussetzung

Literatur

[Ban06] Tim Banks. Web Services Resource Framework (WSRF) - Primer v1.2. http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf, May 2006.

[Ber05] Thomas G. Berger. *Konzeption und Management von Service-Level-Agreements für IT-Dienstleistungen*. PhD thesis, Fachbereich Rechts- und Wirtschaftswissenschaften der Technischen Universität Darmstadt, Darmstadt, April 2005.

[BPE06] Business Process Execution Language for Web Services Version 1.1. `ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf`, 2006. Letzter Zugriff: 2006-11-15.

[FKNT02] Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. 2002.

[GHH+08] Stefan Gudenkauf, Wilhelm Hasselbring, Felix Heine, André Höing, Guido Scherp, and Odej Kao. Bis-Grid: Business Workflows for the Grid. In Marian Bubak, Michal Turala, and Kazimierz Wiatr, editors, *CGW'07 Proceedings*, pages 86–94, Krakow, Poland, 2008. ACC CYFRONET AGH.

[GP95] David Garlan and Dewayne E. Perry. Introduction to the Special Issue on Software Architecture. *IEEE Trans. Softw. Eng.*, 21(4):269–274, 1995.

[HGS08] Andre Höing, Stefan Gudenkauf, and Guido Scherp. BIS-Grid Deliverable 3.2: Documentation - WS-BPEL Engine. Technical report, BIS-Grid, August 2008.

[HHG+07] Felix Heine, Andre Höing, Stefan Gudenkauf, Guido Scherp, Holger Nitsche, and Jens Lischka. BIS-Grid Deliverable 3.1: Specification. Technical report, BIS-Grid, December 2007.

[HSG+09] André Höing, Guido Scherp, Stefan Gudenkauf, Dirk Meister, and André Brinkmann. An Orchestration as a Service Infrastructure Using Grid Technologies and WS-BPEL. In *ICSOC/ServiceWave*, pages 301–315, 2009.

[IEE00] IEEE Computer Society. IEEE 1471-2000 Recommended Practice for Architectural Description for Software-Intensive Systems. Technical report, IEEE Computer Society, http://ieeexplore.ieee.org/servlet/opac?punumber=7040, 2000.

[OAS07] OASIS WSBPEL Technical Committee. Web Services Business Process Execution Language (WSBPEL) Primer. May 2007.

[RH08] Ralf Reussner and Wilhelm Hasselbring, editors. *Handbuch der Software-Architektur*. dpunkt.verlag, 2., überarb. und erw. Auflage edition, Dezember 2008.

[RHP+07] Nick Ragouzis, John Hughes, Rob Philpott, Eve Maler, Paul Madsen, and Tom Scavo. Security Assertion Markup Language (SAML) V2.0 Technical Overview. `http://www.oasis-open.org/committees/download.php/22553/sstc-saml-tech-overview-2%200-draft-13.pdf`, February 2007. Working Draft.

[VRMCL09] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39(1):50–55, 2009.

[W3C] W3C Working group. XML Schema 1.1. `http://www.w3.org/XML/Schema`.

[W3C07] W3C Working group. XML Path Language (XPath) 2.0. `http://www.w3.org/TR/xpath20/`, January 2007. W3C Recommendation.

[Web] Web Services Addressing Working Group. Web Services Addressing 1.0. `http://www.w3.org/2002/ws/addr/`. W3C Recommendation.

[WfM99] Workflow Management Coalition Terminology & Glossary. Technical report, Workflow Management Coalition, Winchester, Hampshire SO23 8BB, United Kingdom, February 1999.