

Grobentwurf

Version 1.0

29. Oktober 2009



Dok-Status: neu

QS-Status: nicht QS-geprüft

Prüf-Status: nicht geprüft

Review-Status: kein Review durchgeführt

End-Status: -

Inhaltsverzeichnis

1	Einleitung	5
1.1	Zweck des Dokuments	5
1.2	Das SIMPL Rahmenwerk	5
1.3	Anforderungen	5
1.3.1	Einfache Modellierung	5
1.3.2	Große Datenmengen	6
1.3.3	Auditing von Prozessen	6
1.3.4	Late Binding von Datenquellen	6
1.3.5	Authentifizierung und Autorisierung	6
1.3.6	Registry	6
1.3.7	Admin-Konsole	6
1.3.8	Erweiterbarkeit	7
1.3.9	Verwendbarkeit	7
1.4	Entwurfsprinzipien	7
1.4.1	Offenes Rahmenwerk	7
1.4.2	Modularisierung	7
1.4.3	Kopplung und Zusammenhalt	7
1.4.4	Entwicklungsrichtung	8
1.4.5	Plug-In	8
1.4.6	Adapter	8
1.5	Überblick über den Grobentwurf	8
2	Architektur	8
2.1	Apache Tomcat	8
2.2	Eclipse	8
3	Komponenten	9
3.1	Eclipse	9
3.2	Apache ODE	9
3.3	SIMPL Core	9
3.3.1	Security Service	9
3.3.2	Reference Service	9
3.3.3	Strategy Service	11
3.3.4	Transformation Service	11
3.3.5	Administration Service	11
3.3.6	Datasource Service	11
3.3.7	Registry Service	11
3.3.8	Storage Service	11
3.4	Apache jUDDI	11
4	Schnittstellen	12
4.1	Apache ODE	12
4.1.1	DatasourceService	13
4.1.2	StrategyService	13
4.1.3	ReferenceService	13
4.1.4	SecurityService	13
4.2	Eclipse	13
4.2.1	AdministrationService	13
4.2.2	TransformationService	13
4.2.3	StorageService	13

4.2.4	RegistryService	13
4.3	SIMPL Core	13

Änderungsgeschichte

Version	Datum	Autor	Änderungen
0.1	29.10.2009	schneimi	Erstellung des Dokuments
0.2	30.10.2009	schneimi	Kapitel 1
0.3	04.11.2009	schneimi	Kapitel 2, Kapitel 3.3
0.4	07.11.2009	schneimi	Kapitel 4

1 Einleitung

Dieses Kapitel soll dem Leser einen Überblick über das SIMPL Rahmenwerk geben und Zweck, Anforderungen und die eingesetzten Entwurfsprinzipien erläutern. Die Struktur und der Aufbau des Dokuments orientieren sich dabei an der Entwurfsvorlage [1] von Markus Knauf.

1.1 Zweck des Dokuments

Der Grobentwurf beschreibt das Rahmenwerk auf Komponentenebene und bildet die Grundlage für den späteren Feinentwurf. Es werden alle wichtigen Komponenten sowie Schnittstellen identifiziert und ihr Zusammenspiel beschrieben. Damit soll herausgestellt werden, dass das resultierende System funktionieren kann und den Anforderungen gerecht wird.

1.2 Das SIMPL Rahmenwerk

Das SIMPL Rahmenwerk soll dem Benutzer eine generische und erweiterbare Umgebung bieten, die eine einfache Modellierung von BPEL-Geschäftsprozessen mit Zugriff auf beliebige Datenquellen ermöglicht. Bei den Datenquellen kann es sich beispielsweise um Datenbanken, Sensornetze oder Dateisysteme handeln. Die Modellierung der Prozesse findet in Eclipse mit dem Eclipse BPEL Designer Plug-In statt, das für diesen Zweck um zusätzliche Aktivitäten, für den Zugriff auf Datenquellen, erweitert wird. Die Ausführung der Prozesse erfolgt durch die Apache ODE Workflow Engine, bei der das bestehende Event-Modell und das Auditing der Prozessdaten für diese Aktivitäten angepasst werden. Dienste, die für die Ausführung der Aktivitäten von der Workflow Engine benötigt werden, werden in Form von Web Services bereitgestellt. Diese Aktivitäten werden im folgenden Verlauf des Dokuments als Data-Management-Aktivitäten bzw. DM-Aktivitäten bezeichnet.

1.3 Anforderungen

In diesem Abschnitt werden die Anforderungen des SIMPL Rahmenwerks beschrieben.

1.3.1 Einfache Modellierung

Bei der Modellierung von Prozessen wiederholen sich in der Regeln häufig längere Statements in Anfragesprachen wie z.B SQL und XQuery oder auch längere Namen von Datenquellen oder Datencontainern (Tabellen, Dateien, XML-Dokumente, etc). Damit diese vom Prozess-Modellierer nicht jedes mal vollständig angegeben werden müssen, soll es die Möglichkeit geben diese in BPEL-Variablen zu hinterlegen, die anschließend als Referenzen in anderen Statements verwendet werden können.

1.3.2 Große Datenmengen

Der Schwerpunkt des Rahmenwerks liegt bei der Modellierung von wissenschaftlichen Prozessen, bei denen überwiegend mit großen Datenmengen gearbeitet wird. Damit diese Datenmengen nicht innerhalb des Prozess gehalten werden müssen, wird ein Reference Resolution System (RRS) realisiert, das es ermöglicht Daten zu referenzieren, die nur bei Bedarf aufgelöst werden und somit sehr schnell weitergegeben werden können.

1.3.3 Auditing von Prozessen

Bei dem Auditing von Prozessen handelt es sich um das Protokollieren von Daten, die in Prozessen anfallen und ein Monitoring der Prozesse ermöglichen. Die Erfassung der Daten muss auf die zusätzlichen DM-Aktivitäten angepasst werden und dem Benutzer die Möglichkeit gegeben werden, die Granularität der Daten zu bestimmen. Zusätzlich soll das interne Auditing von Apache ODE, auf eine frei wählbare Datenbank umgeleitet werden können.

1.3.4 Late Binding von Datenquellen

Da bei der Modellierung nicht immer unmittelbar feststeht, auf welche Datenquelle zugegriffen werden soll, beispielsweise beim Ablegen von Daten, soll ein Late Binding von Datenquellen unterstützt werden. Damit kann durch die Formulierung von Anforderungen an die Datenquelle und das Wählen einer Auswahlstrategie durch den Prozess-Modellierer, eine passende Datenquelle zur Laufzeit bestimmt werden.

1.3.5 Authentifizierung und Autorisierung

Datenquellen erfordern in der Regel eine Authentifizierung und Autorisierung des Benutzers bei einem Zugriff. Das Rahmenwerk soll dem Prozess-Modellierer diesen Vorgang vereinfachen, so dass die dafür benötigten Informationen wie z.B. Name und Passwort nicht bei jedem wiederholten Zugriff erneut angegeben werden müssen. Dazu soll das Konzept des Single Sign On angeboten bzw. unterstützt werden.

Der Single Sign On (SSO) ist ein Konzept der Authentifizierung, bei dem sich der Benutzer nur einmal erfolgreich bei einem System authentifizieren muss, und auch bei weiteren Zugriffen, evtl. auch auf andere Komponenten des Systems, kein erneuter Authentifizierungsvorgang benötigt wird. Das SIMPL Rahmenwerk soll dabei verschiedene Authentifizierungsverfahren unterstützen. Bei der Modellierung soll der Prozess-Modellierer zudem dahingehend unterstützt werden, dass er Authentifizierungsinformationen nicht zu jeder DM-Aktivität, einer bestimmten Datenquelle, wiederholt angeben muss.

1.3.6 Registry

Für die Verwaltung der für den Prozess-Modellierer zur Verfügung stehenden Datenquellen, soll eine Registry bereitgestellt werden, in der die Datenquellen vom Workflow-Administrator zentral definiert werden können. Dort werden auch die entsprechenden Eigenschaften der Datenquellen hinterlegt, die für das Late Binding benötigt werden.

1.3.7 Admin-Konsole

Für alle wichtigen Einstellungen des Rahmenwerks soll eine Admin-Konsole bereitgestellt werden, über die der Workflow-Administrator Einstellungen treffen kann. Dies betrifft vor allem alle Einstellungen die auch zur Laufzeit getätigt werden können, wie z.B. das Einbinden einer zusätzlicher Datenquelle.

1.3.8 Erweiterbarkeit

Das Rahmenwerk stellt folgende Anforderungen an die Erweiterbarkeit und soll dafür entsprechende “Hot Spots” bereitstellen:

- Weitere Typen von Datenquellen
- Unterstützung weiterer Anfragesprachen
- Erweiterung um neue DM-Aktivitäten
- Erweiterung um neue Events für das Auditing
- Unterstützung weiterer Authentifizierungs- und Autorisierungsverfahren
- Unterstützung weiterer Registries
- Austauschbare GUI der Admin-Konsole

1.3.9 Verwendbarkeit

Die Aktivitäten, die für SIMPL entwickelt werden, sollen auf beliebigen anderen BPEL Workflow-Engines ausgeführt werden können. Dazu muss der von den Aktivitäten bei der Modellierung erzeugte, erweiterte Code, in Standard-BPEL transformiert werden. Eine zusätzliche Anforderung ist, dass die von SIMPL bereitgestellten Web Services, auf verschiedenen Web Containern lauffähig sein müssen.

1.4 Entwurfsprinzipien

In diesem Abschnitt werden die Prinzipien beschrieben, die bei dem Entwurf angewendet werden. (vgl. [2], Kapitel 17)

1.4.1 Offenes Rahmenwerk

Bei dem SIMPL Rahmenwerk handelt es sich größtenteils um ein offenes Rahmenwerk. Daher können viele Erweiterungen nicht ohne technisches Verständnis und Wissen über die Mechanismen und Abläufe des Rahmenwerks realisiert werden. Die Erweiterungsmöglichkeiten werden daher ausführlich dokumentiert und Beispiele erstellt, anhand denen eigene Erweiterungen umgesetzt werden können. Bereiche des Rahmenwerks bei denen im Laufe des Projekts ausreichend Erfahrung gesammelt wurde, werden wenn möglich in geschlossener Form für die Entwicklung bereitgestellt.

1.4.2 Modularisierung

Durch die Modularisierung werden die Komponenten in einfache und leicht verständliche Teile gegliedert. Die Realisierungsdetails eines Moduls werden nach dem Prinzip des Information Hiding versteckt und die Dienste nur über eine Schnittstelle angeboten. Ziel ist es, später Module ändern oder austauschen zu können, möglichst ohne dabei die Schnittstellen ändern zu müssen und damit Auswirkungen auf andere Module zu verursachen.

1.4.3 Kopplung und Zusammenhalt

Bei dem Entwurf der Module wird darauf geachtet, dass die Kopplung zu anderen Modulen möglichst gering bleibt und der Zusammenhalt innerhalb des Moduls möglichst hoch wird. Durch dieses Vorgehen wird eine hohe Lokalität und damit gute Wartbarkeit erreicht, da sich Fehler die bei Änderungen entstehen, nicht im System fortpflanzen können.

1.4.4 Entwicklungsrichtung

Bei der Entwicklung wird Top-down vorgegangen. Dabei wird die Aufgabe des Rahmenwerks rekursiv bis zur elementaren Ebene (der Programmiersprache) in Teilaufgaben zerlegt und damit schrittweise verfeinert.

1.4.5 Plug-In

Plug-Ins sind externe Software-Einheiten, durch die das Rahmenwerk um zusätzliche Funktionalität erweitert werden kann. Das Rahmenwerk bietet dafür entsprechende “Hot Spots” an, an denen die Plug-Ins angeschlossen werden können.

1.4.6 Adapter

Adapter bzw. Konnektoren sind interne Verbindungsstücke, die dort entwickelt werden, wo Komponenten angebunden werden sollen, deren Schnittstellen nicht zu vorhandenen Schnittstellen des Rahmenwerks passen.

1.5 Überblick über den Grobentwurf

- Kapitel 2 “Architektur” beschreibt die Architektur des Rahmenwerks. Das Rahmenwerk wird in überschaubare Komponenten gegliedert, die jeweils genau definierte Funktionen erfüllen.
- Kapitel 3 “Komponenten” beschreibt die im Kapitel “Architektur” beschriebenen Komponenten im Detail. Dabei werden Schnittstellen, Protokolle und Verhalten definiert.
- Kapitel 4 “Schnittstellen” beschreibt die in Kapitel “Komponenten” definierten Schnittstellen und beschreibt wie diese genutzt werden.

2 Architektur

Abbildung 1 zeigt eine Übersicht des SIMPL Rahmenwerks. Die übergeordneten Komponenten und ihr Zusammenspiel werden im Folgenden kurz beschrieben. Auf die elementaren Komponenten des Rahmenwerks wird dann in Kapitel 3 eingegangen.

2.1 Apache Tomcat

Apache Tomcat ist die Laufzeitumgebung für Apache ODE, sowie den SIMPL Core und die Registry Apache jUDDI. Apache ODE ist für die Ausführung der Prozesse nach der Modellierung zuständig und benötigt, wie bereits in Kapitel 1.2 erwähnt, bestimmte Dienste für die Ausführung der zusätzlichen DM-Aktivitäten in Prozessen. Diese Dienste werden von dem SIMPL Core bereitgestellt, der sich aus verschiedenen, nach Aufgaben eingeteilten, Web Services zusammensetzt. Mit der Registry Apache jUDDI werden die im Rahmenwerk zur Verfügung stehenden Datenquellen vom Workflow-Administrator zentral bereitgestellt.

2.2 Eclipse

Die Entwicklungsumgebung Eclipse bildet, mit den entsprechenden Plug-Ins, die GUI für den Prozess-Modellierer, als auch den Workflow-Administrator. Mit dem Eclipse BPEL Designer Plug-In kann der Prozess-Modellierer bereits BPEL Prozesse erstellen und auf dem Apache ODE zum Einsatz bringen (deployen). Das SIMPL Eclipse Plug-In erweitert nun zum Einen die bestehenden Aktivitäten des Eclipse BPEL Designer Plug-Ins um die DM-Aktivitäten und bietet zum Anderen die GUI mit den Einstellungen für den Workflow-Administrator, wie z.B die Admin-Konsole und die Globalen Einstellungen (siehe [3], Kapitel 4).

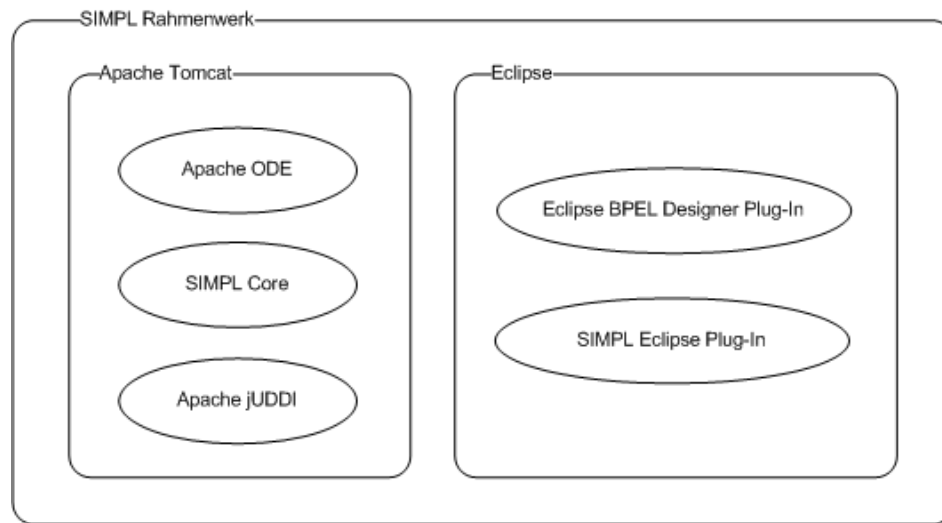


Abbildung 1: Architektur des SIMPL Rahmenwerks

3 Komponenten

In Abbildung 2 werden die für das SIMPL Rahmenwerk wichtigen Komponenten und Abhängigkeiten gezeigt, die in den folgenden Abschnitten näher beschrieben werden.

3.1 Eclipse

3.2 Apache ODE

3.3 SIMPL Core

Der SIMPL Core stellt alle Funktionalität zur Verfügung, die während der Modellierung und der Ausführung von Prozessen benötigt werden. Er besteht aus mehreren Web Services (siehe 3), die jeweils fest definierte Aufgaben innerhalb des SIMPL Rahmenwerks haben. Die Web Services werden dabei innerhalb einer Apache Axis2 Installation bereitgestellt und mit der Java API JAX-WS erstellt, damit ein Einsatz auch mit anderen Web Containern, außer Apache Tomcat, gewährleistet ist. (siehe Anforderung 1.3.9)

Jeder Web Service bietet nach Außen eine WSDL-Schnittstelle und als Übertragungsprotokoll zur Kommunikation wird SOAP verwendet. Wie diese Kommunikation im Detail aussieht ist dann Thema von Kapitel 4. Im den folgenden Abschnitten werden die Web Services und ihre Aufgaben näher beschrieben.

3.3.1 Security Service

Der Security Service ist zuständig für Aufgaben, die die Authentifizierung und Autorisierung gegenüber Datenquellen betreffen. Mit diesem Service können Authentifizierungs- und Autorisierungsinformationen verarbeitet und ggf. transformiert werden. Die Informationen können außerdem über den Storage Service zwischengespeichert werden um das Konzept des SSO (siehe 1.3.5) zu realisieren.

3.3.2 Reference Service

Der Reference Service erfüllt die Anforderung in Abschnitt 1.3.2, große Datenmengen in BPEL referenzieren zu können. Über diesen Service lassen sich die Referenzen verwalten und natürlich auflösen,

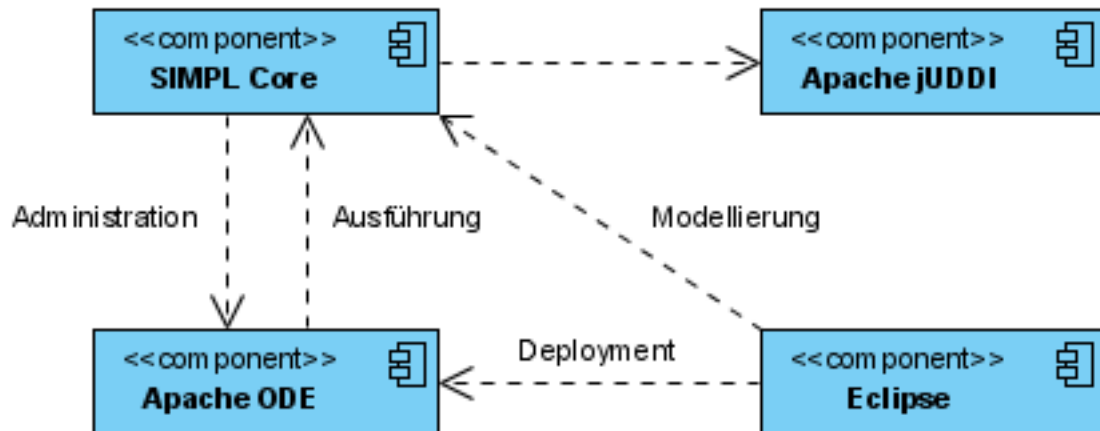


Abbildung 2: Komponenten

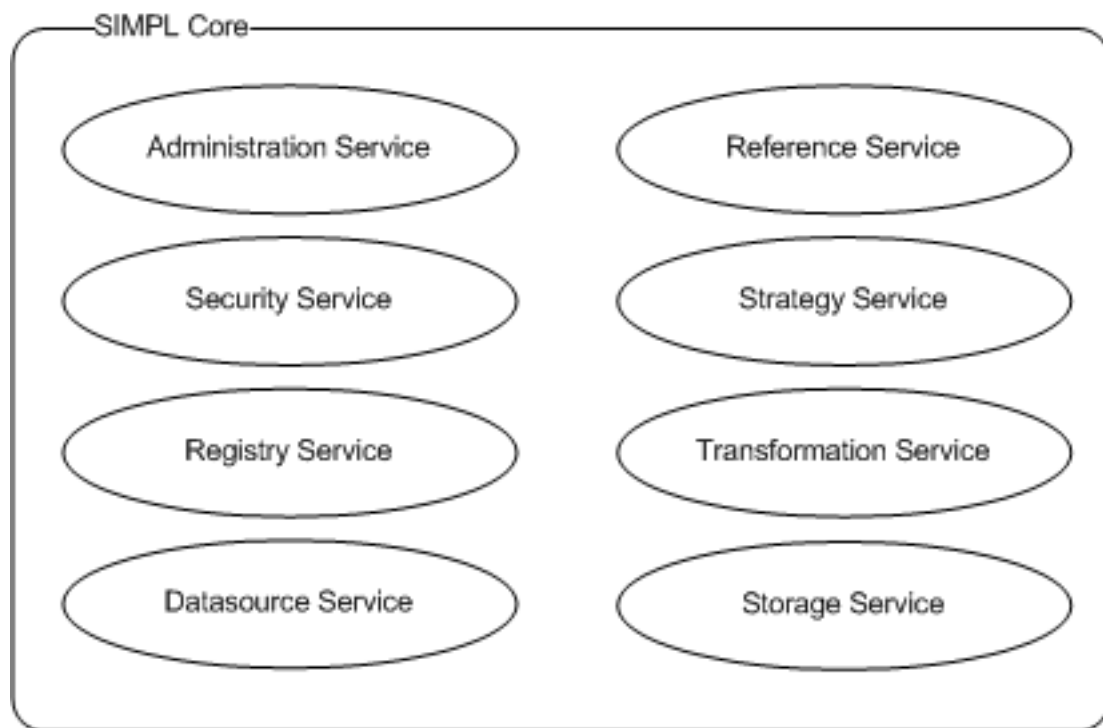


Abbildung 3: SIMPL Core

um die referenzierten Daten bei Bedarf in einen Prozess zu holen.

3.3.3 Strategy Service

Mit dem Strategy Service wird das Late Binding (siehe 1.3.4) ermöglicht, dort stehen verschiedene Auswahlstrategien bzw. -algorithmen zur Verfügung, um mit den im Prozess formulierten Anforderungen eine passende Datenquelle ausfindig zu machen. Um die vorhandenen Datenquellen abzurufen und die Anforderungen mit den Eigenschaften der Datenquellen zu vergleichen, wird der Registry Service in Anspruch genommen.

3.3.4 Transformation Service

Damit modellierte Prozesse, die DM-Aktivitäten enthalten, auch auf anderen Workflow Engines ausgeführt werden können (siehe Anforderung 1.3.9), wird der Transformation Service bereitgestellt, mit dem der erweiterte BPEL Code in Standard-BPEL-Code transformiert werden kann.

3.3.5 Administration Service

Über den Administration Service werden alle Einstellungen des Rahmenwerks verwaltet und die Funktionalität für die Admin-Konsole (siehe 1.3.7) bereitgestellt. Für Einstellungen, mit denen zur Laufzeit Einfluss auf Apache ODE genommen wird, wie z.B. die Granularität des Auditing, werden die vorhandenen Schnittstellen von Apache ODE verwendet und, falls nötig, neue Schnittstellen geschaffen. Die in Abschnitt 1.3.8 geforderte austauschbare GUI, wird durch die unabhängige WSDL-Schnittstelle erreicht.

3.3.6 Datasource Service

Der Datasource Service ist für alle Aufgaben zuständig, die den Zugriff auf die Datenquellen betreffen. Dort werden entsprechende Adapter (siehe 1.4.6) in Form von Direct Access Services (DAS) implementiert, die den Zugriff auf verschiedene Typen von Datenquellen ermöglichen und die Erweiterbarkeit für weitere Typen und Anfragesprachen (siehe Anforderungen 1.3.8) garantieren. Des weiteren werden Plug-In-Schnittstellen (siehe 1.4.5) für zusätzliche funktionale Erweiterungen geschaffen, wie z.B. die Unterstützung verschiedener Dateitypen bei Dateisystemen.

3.3.7 Registry Service

Über den Registry Service kann auf die Registry des Rahmenwerks zugegriffen werden, in der die vorhandenen Datenquellen mit ihren Eigenschaften zentral verwaltet werden. (siehe 1.3.6)

3.3.8 Storage Service

Der Storage Service bildet den zentralen Speicherort von Daten des SIMPL Rahmenwerks, dort laufen alle Einstellungen, Daten und Informationen zusammen und werden persistent über den Datasource Service in einer Datenquelle gespeichert. Beispielsweise werden dort die Referenzen gespeichert, die über den Reference Service hinzugefügt werden.

3.4 Apache jUDDI

Das SIMPL Rahmenwerk stellt mit Apache jUDDI bereits eine UDDI-Registry zur Verfügung, die von dem Registry Service genutzt wird. Über die WSDL Schnittstelle des Registry Service kann aber auch jede andere Implementierung einer Registry realisiert werden (siehe Anforderungen 1.3.8).

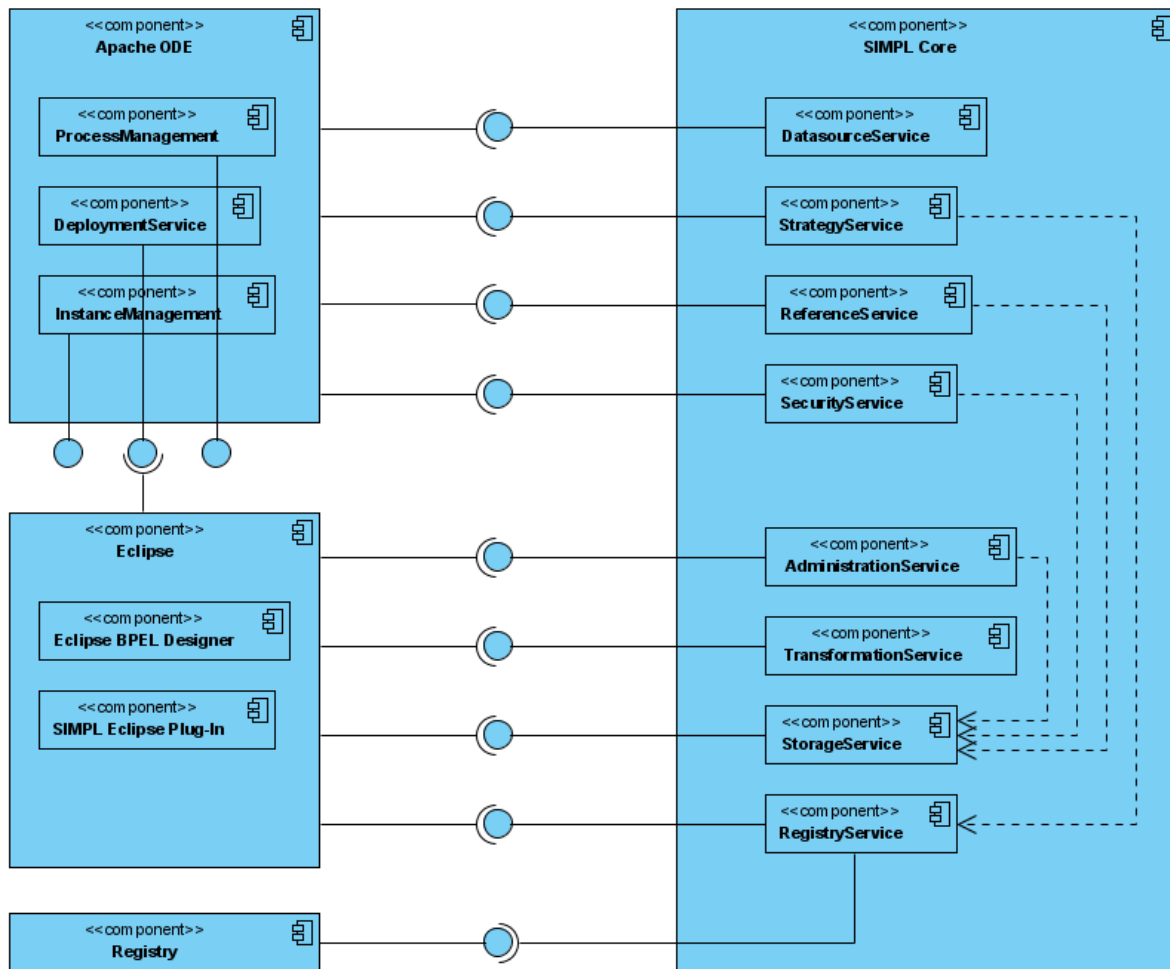


Abbildung 4: Schnittstellen

4 Schnittstellen

In diesem Kapitel wird die Kommunikation der Komponenten über die Schnittstellen beschrieben und interne und externe Abhängigkeiten, sowie eingesetzte Protokolle und übertragene Objekte erläutert. Abbildung 4 zeigt dazu die Schnittstellen in einer Übersicht, die in den folgenden Abschnitten näher beschrieben werden.

4.1 Apache ODE

Folgende Schnittstellen werden von Apache ODE, in einem Prozess mit DM-Aktivitäten, zur Laufzeit verwendet.

4.1.1 DatasourceService

4.1.2 StrategyService

4.1.3 ReferenceService

4.1.4 SecurityService

4.2 Eclipse

Während der Modellierung in Eclipse, werden folgende Schnittstellen des SIMPL Core bzw. von Apache ODE verwendet.

4.2.1 AdministrationService

4.2.2 TransformationService

4.2.3 StorageService

4.2.4 RegistryService

4.3 SIMPL Core

Literatur

- [1] Knauß, Markus (März 2008): *Entwurfsvorlage*, <http://www.iste.uni-stuttgart.de/se/>.
- [2] Ludewig, Jochen; Lichter, Horst (2007): *Software Engineering: Grundlagen, Menschen, Prozesse und Techniken*, punkt.verlag GmbH.
- [3] Stupro-A SIMPL: *Spezifikation* (2009), <http://code.google.com/p/simpl09/>.

Abkürzungsverzeichnis

API	Application Programming Interface
BPEL	Business Process Execution Language
DAS	Data Access Service
DM	Data-Management
GUI	Graphical User Interface
JAX-WS	Java API for XML - Web Services
ODE	Orchestration Director Engine
RRS	Reference Resolution System
SAML	Security Assertion Markup Language
SDA	Service Data Object
SIMPL	SimTech: Information Management, Processes and Languages
SOAP	ehem. Simple Object Access Protocol
SQL	Structured Query Language
SSO	Single Sign On
UDDI	Universal Description, Discovery and Integration
WS	Web Service
WSDL	Web Service Description Language
XACML	eXtensible Access Control Markup Language
XQuery	XML Query Language

Abbildungsverzeichnis

1	Architektur des SIMPL Rahmenwerks	9
2	Komponenten	10
3	SIMPL Core	10
4	Schnittstellen	12