

# **Versteckte Schätze in BPM & SOA Suite 11g - gesammelte Projekterfahrungen**

**Danilo Schmiedel**  
**OPITZ CONSULTING Berlin GmbH**  
**Berlin**

**Dr. Jens Hündling**  
**ORACLE Deutschland B.V. & Co. KG**  
**Potsdam**

## **Schlüsselworte**

Oracle Fusion Middleware, Oracle SOA Suite, Oracle BPM Suite, Erfahrungsbericht, BPMN.

## **Einleitung**

Es existieren eine Reihe von Features in der aktuellen SOA/BPM Suite, die selten genutzt und wenig beworben werden. Darüber hinaus gibt es in verschiedensten Kundenprojekten häufig auch Anforderungen, die zunächst Implementierungs- oder Anpassungsaufwand bedeuten, sich dann aber doch mit Bord-Mitteln realisieren lassen. In diesem Vortrag wollen wir auf ausgewählte Features eingehen und diese im Kontext von Kundenprojekten und Erfahrungen präsentieren. Neben einem durchgängigen Beispiel zur Verwendung des Metadata Stores, dem Fault Management Framework, der OWSM-Policies, dem Dynamic Binding und der Datenvalidierung via Schematron sollen auch die sogenannten Domain Value Maps näher betrachtet werden. Die Betrachtung neuer technischer Features, die sich erst seit 11g adäquat umsetzen lassen sowie die Verbindung zwischen BPMN/BPEL und Java runden den Beitrag ab.

Ferner sollen die vielen Berührungspunkte zwischen einzelnen Technologien anhand der folgenden Beispiele demonstriert werden:

- Integration von Java Code in der Service Component Architecture (SCA)
- Erweiterung der Plattformfunktionalitäten für Errorhandling und Security
- Embedded Java Komponenten für ein verbessertes Monitoring von Prozessinstanzen
- Customizing von Oberflächen für zentrale Postkorbapplikationen (Aufgabenverwaltung, Workflow)

## **Rent your legacy car (RYLC) – wie sich ein Unternehmen auf den erfolgreichen Weg zu SOA macht.**

Um die Funktionalitäten möglichst anschaulich präsentieren zu können, wollen wir ein durchgängiges Beispiel mit dem Titel „Rent your legacy car“ einführen. Dies soll ein Appetit-Anreger sein, der sich aus den Projekterfahrungen der OPITZ CONSULTING GmbH und ORACLE speist und leicht auf andere Unternehmen übertragen werden kann.

Rent Your Legacy Car (RYLC) ist ein Autoverleihunternehmen mit der ambitionierten Vision, in den nächsten Jahren zu den drei umsatzstärksten Unternehmen des Car-Rental-Segments zu gehören. Seit längerem ist das RYLC-Management mehr und mehr gezwungen, sich mit einer aggressiven Marktsituation auseinanderzusetzen. In den letzten Jahren hat RYLC immer wieder Marktanteile verloren. Die Konkurrenz konnte neue Produkte und besseren Service schneller bieten, während RYLC aufgrund von IT Integrationsproblemen nicht nachkam. Die Branche ist allgemein im Aufwind. Im gesamten Jahr 2010 meldeten die Beherbergungsbetriebe 380,3 Millionen Gästeübernachtungen. Dies ist ein Anstieg von 3,2% gegenüber dem Vorjahr (Quelle: Statistisches Bundesamt). Die zentralen Kundengruppen von RYLC sind Urlaubs- & Geschäftsreisen. In diesem Segment soll stark auf eine IT Unterstützung und eine effiziente Online-Buchung gesetzt werden.

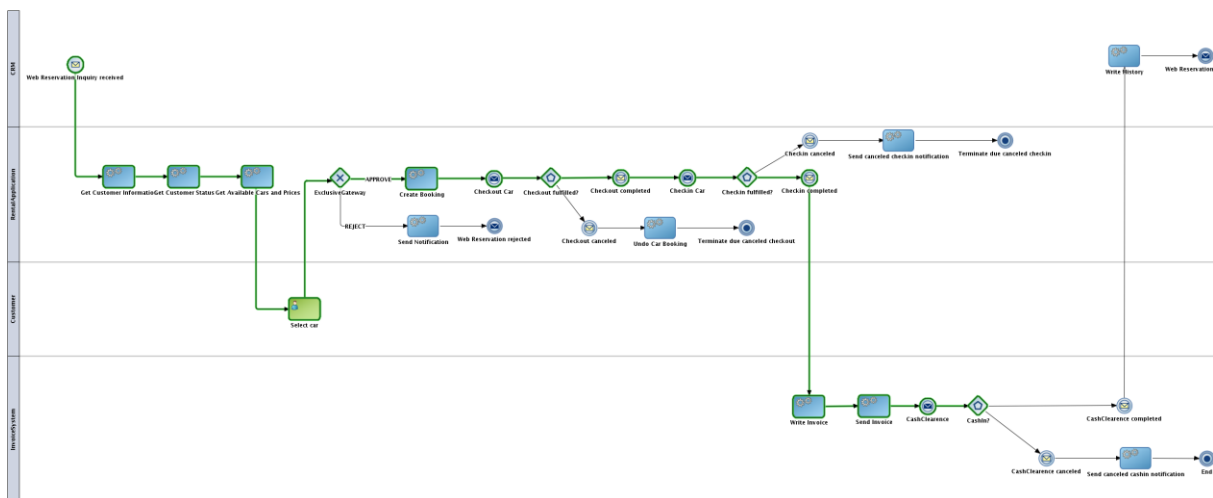


Abb. 1: Hauptprozess „RYLC Car reservation“ in der Notation BPMN (modelliert in der Oracle BPM Suite)

Weitere Hintergründe und Rahmenbedingungen zu dem RYLC Szenario sind in dem Sonderheft Software und Support mit dem Titel „SOA Spezial - Ready for Change“ erschienen (November 2009, Berthold Maier, Hajo Normann, Bernd Trops, Clemens Utschig-Utschig, Torsten Winterberg). Neben grundlegenden Architektur-Prinzipien wie lose Kopplung, Service-Kategorisierung und kanonische Datenmodelle werden darin auch die Themen Governance, Security und Benutzeroberflächen behandelt.

Da es sich bei RYLC um eine Art „Pet Store for SOA“ handeln soll, wurden im Rahmen der Implementierung auch vielerlei versteckte Schätze und Features in der Oracle BPM bzw. SOA Suite deutlich. Einen Überblick hierzu soll dieser Beitrag liefern.

### Wiederverwendung von Daten und Services mittels MDS

Ein wichtiger Aspekt in einer Serviceorientierten Architektur ist die Veröffentlichung von Metadaten potenziell wiederverwendbarer SOA-Artefakte. Mit den Metadata Services (MDS) bietet die Oracle SOA Suite eine solche Komponente an, um die zentrale Verwaltung und Bereitstellung von Schnittstellen (WSDL), Daten (XSD), Geschäftsregeln (Rules), FaultPolicies (XML) und Geschäftsereignissen (EDL) zu ermöglichen. Somit können Daten und Dokumente zur Design- und Laufzeit applikationsübergreifend angeboten werden und die Duplizierung von Code wird reduziert.

Folgende Erkenntnisse aus Projekten und Best Practices der Entwicklung werden im Vortrag anhand von Demos vertieft:

- 1) Versionierung von Services und Daten ist ein für den Betrieb einer SOA wichtiger Aspekt und muss bereits im Rahmen des Service-Designs berücksichtigt werden (Service Lifecycle Management). Andernfalls bedeuten Änderungen in den Schnittstellen einen erhöhten Entwicklungsaufwand und wenig Potenzial für Wiederverwendung.
- 2) Als Best Practice hat sich zur Entwicklungszeit die Verwendung eines lokalen MDS Repositories bewährt. Dieses wird bereits mit einer herkömmlichen Installation des Oracle JDevelopers bereitgestellt. Auf performante Art und Weise kann somit die zukünftige Speicherung in einem zentralen Repository von Beginn an berücksichtigt werden. Besonders vorteilhaft ist dabei auch, dass nicht jeder Entwicklungsschritt der MDS-Inhalte sofort deployed werden muss.
- 3) Zur Laufzeit stellt die Oracle SOA Suite ein zentrales MDS-Repository auf dem Server bereit, in welches die verteilten Schnittstellen, Daten, Event-Definitionen und Fault-Policies deployed werden.
- 4) Unabhängig ob das lokale oder das Server-MDS verwendet werden soll, so erfolgt die Referenzierung der Inhalte aus einem SCA Composite mit Hilfe folgender URL:  
oramds:/apps/<ORDNER>/<Service>.wsdl
- 5) Die Administration des MDS Repositories zum Export, Import oder Löschen der Metadaten kann über den Enterprise Manager, ANT oder das Weblogic Scripting Tool (WLST) erfolgen.

### Einheitliche Behandlung von Fehlern mit Hilfe des Fault Management Frameworks

Die SOA Suite 11g bietet einen einheitlichen Ansatz für die Behandlung von Fehlern auch über die Grenzen der einzelnen Komponenten eines SCA-Composites hinweg. Wichtig ist hierbei, dass zwischen fachlichen Fehlern, die Bestandteil der Business Logik sind und technischen Fehlern (beispielsweise aufgrund von fehlerhaften Datentransformationen oder nicht verfügbaren Service-Endpunkten) unterschieden wird. Neben der Möglichkeit das Fehlerhandling beispielsweise direkt in einem BPEL- oder BPMN-Prozess zu verorten, stellt die Oracle Plattform auch das sogenannte Fault Management Framework bereit. Tritt etwa ein Fehler beim Aufruf einer Prozessaktivität auf, fängt das Framework diesen ab und führt eine konfigurierte Aktion durch, welche zuvor in einem assoziierten Fault-Policy-File definiert wurde. Um etwa die Wiederaufnahme einer fehlerhaften Prozessinstanz von der Administrationsoberfläche zu ermöglichen, ist es im Rahmen dessen auch möglich mit manuellen Aktionen in den Lebenszyklus einer Instanz einzugreifen (siehe Abb. 2).

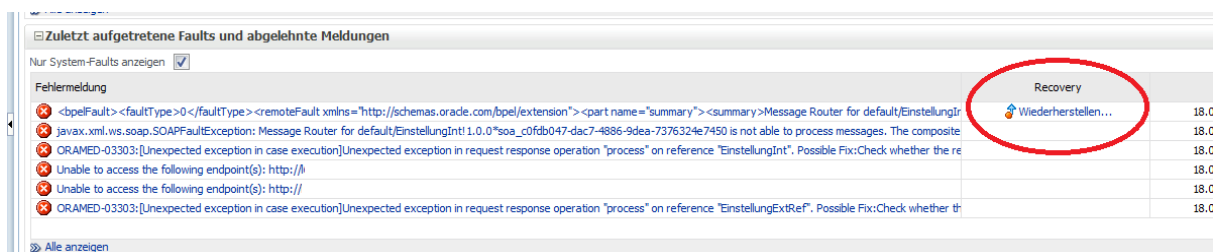


Abb. 2: Wiederherstellen einer fehlerhaften Prozessinstanz via Fault Management Framework

Im Vortrag werden wir verschiedene Komponenten und Funktionalitäten des Fault Management Frameworks am Beispiel demonstrieren.

## Component Properties: zur Laufzeit veränderbare Konfigurationsparameter

Für die Auslagerung von Konfigurationswerten - wie beispielsweise ein umgebungsabhängiger Pfad zu einem Dateiablageort - bietet sich die Verwendung sogenannter Component-Properties an. Die Definition der Properties erfolgt in der Hauptdatei einer Composite Applikation (composite.xml) als Child-Element des <component>-Knotens.

```
<component name="TestProcess" version="1.1">
  <implementation.bpel src="BPEL_PROZESS_NAME.bpel"/>
  <property name="bpel.preference.testPreference">TestValue</property>
</component>
```

Mit der Funktion `ora:getPreference(„testPreference“)` können derartige Properties innerhalb eines BPEL-Prozesses ausgelesen und verwendet werden. Zusätzlich zur umgebungsspezifischen Anpassung der Werte für unterschiedliche Deployments (mittels Config-Plan) können diese auch zur Laufzeit verändert werden. Dies geschieht im MBean-Browser des Enterprise Managers, welcher unter den folgenden Menüoptionen erreicht werden kann: Menü Administration => System-MBean Browser => Anwendungsdefinierte MBeans => `oracle.soa.config` => `COMPOSITE_NAME` => `SCAComposite.SCAComponent` => `PROCESS_NAME`. Im Rahmen des Vortrages werden wir die Definition, Verwendung und Anpassung der Component Properties darstellen.

## Dehydration von BPEL-Prozessen – was lässt sich dies verhindern?

Im Lebenszyklus eines BPEL-Prozesses werden dessen Prozessinstanzen mit Statusinformationen im Repository der Oracle SOA Suite gespeichert. Diesen Speichervorgang nennt man im BPEL-Umfeld auch Dehydration und die dazugehörige Datenbank ist der Dehydration Store. Sobald eine laufende Instanz (an bestimmten Aktivitäten im Prozess) dehydriert wurde, entfernt der BPEL Server diese aus dem Memory. Wenn daraufhin etwa ein Callback eines asynchronen Service-Aufrufes eintrifft oder ein abgelaufener Timer „feuert“, holt sich der BPEL Server die passende Instanz aus dem Dehydration Store zurück in den Memory und fährt mit der Ausführung fort. Einerseits sichert dieser Vorgang die Ausfallsicherheit, da laufende Instanzen nach einem Neustart des BPEL Servers automatisch am letzten Dehydrierungspunkt fortgesetzt werden können. Andererseits kann die Dehydrierung in einem Umfeld mit High Performance Anforderungen und einer enormen Anzahl an Prozessinstanzen kritisch sein.

Wann und wie die Dehydration ausgeführt wird, hängt vom gewählten Prozessstyp ab. Die Oracle BPEL Engine unterscheidet hierbei zwischen „transient“ und „durable“. Transiente Prozesse verursachen keine Dehydration während der Ausführung der Instanz. In Summe werden transiente Prozesse nur einmal am Ende des Prozesses dehydriert. Beispiele hierfür sind kurzlaufende synchrone Request-Response Szenarios. Prozesse vom Typ „durable“ werden mehr als einmal – jeweils an bestimmten Aktivitäten (z.B. Receive, OnMessage, Wait) - dehydriert.

Prozesse, für die im Betrieb eine sehr hohe Anzahl an Instanzen zu erwarten ist, sollten daher als transienter Prozessstyp umgesetzt werden. Im Allgemeinen sind hierfür die folgenden Anforderungen zu erfüllen:

- Keine Receive-Aktivitäten innerhalb des Prozessflusses
- Keine Wait-Aktivitäten
- Keine OnMessage Aktivitäten

- Eine synchrone Schnittstellenbeschreibung

Sind diese Voraussetzungen erfüllt, müssen auf Composite Ebene weitere Attribute gesetzt werden, um eine vollständige Ausführung des Prozesses im Memory zu gewährleisten. Diese werden nachfolgend aufgelistet und kurz beschrieben.

Konfigurationsparameter	Beschreibung
inMemoryOptimization=true	Der Standardwert dieses Parameters ist „false“. Die Eigenschaft kann nur gesetzt werden, wenn keine der o.g. Aktivitäten im Prozess enthalten ist. Steht der Wert auf „true“, wird für Instanzen dieses Prozesses eine Speicheroptimierung bei Wertzuweisungen vorgenommen.
completionPersistPolicy=faultet	Diese Property definiert, welche Prozessinstanzen gespeichert werden. Mit dem Setzen des Wertes auf „faultet“ werden nur fehlgeschlagene Instanzen gespeichert.
oneWayDeliveryPolicy=sync	Der Parameter setzt die Persistenz-Eigenschaften des Prozesses im Delivery Layer. Mit der Zuweisung des Wertes „sync“ wird erreicht, dass die Instanz-erzeugende Nachricht nicht temporär im Delivery Layer gespeichert wird. Die Engine benutzt den Speicher für das Erzeugen der Nachricht.

Im Verlauf des Vortrages soll das Konzept der Dehydrierung und die Unterschiede beider Prozessstypen zur Laufzeit näher erläutert werden.

### Wiederverwendung und Integration von Java-Code in BPEL/BPMN

Im Kontext der Service Component Architecture (SCA) besitzen BPMN und BPEL auch Berührungspunkte mit der weit verbreiteten Programmiersprache Java. So kann vorhandener Java-Code beispielsweise über die Spring Context Komponente in eine Composite Applikation eingebunden werden. Ferner besteht in BPEL die Möglichkeit, auf Basis sogenannter Embedded-Java Aktivitäten den Titel einer Prozessinstanz in der Instanzenansicht des Enterprise Managers zu modifizieren – eine für die Fehleranalyse und das Monitoring enorm hilfreiche Funktionalität, da die Auflistung mit fachlichen Informationen angereichert werden kann (siehe Abb. 3). In der Praxis sorgt dies für eine effizientere Auswahl relevanter Durchläufe.

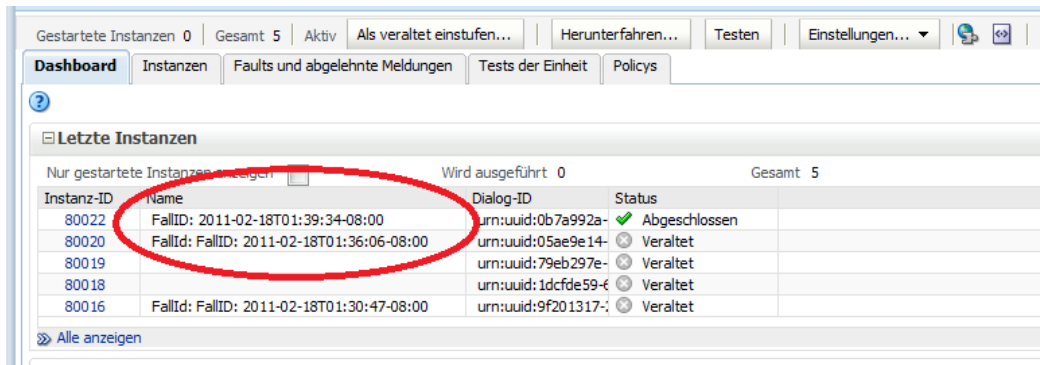


Abb. 3: Setzen des Titels einer Composite-Instanz im Enterprise Manager

Zur Integration eines LDAP-Verzeichnisses bietet die Oracle BPM&SOA Suite eine Reihe von sog. Identity Providern an. Für seltene und nicht standardisierte Lösungen ist es auch möglich, einen eigenen Java-basierten Identity Provider zu integrieren und somit die Plattform um wichtige Security-Aspekte erweiterbar zu machen. Schließlich können aus dem oben erläuterten Fault Management Framework ebenfalls Aufrufe zu Java-Klassen initiiert werden.

### Dynamisches Endpunkt-Binding

In einer SOA findet sich in Praxisprojekten häufig ein spezielles Problem: die dynamische Adressierung von Services. Es sind typischerweise Services, die zwar die gleiche Schnittstelle besitzen, aber auf unterschiedlichen Maschinen gehostet sind. Zur Laufzeit muss nun ermittelt werden, welcher konkrete Endpunkt auf welchem Server respektive in welcher Lokation angesprochen werden kann. Durch die Verbindung mit Oracle Business Rules, Domain Value Maps (DVM) oder Component Properties ist es möglich, die Adresse von Service-Endpunkten zur Laufzeit anzupassen. Das dynamische Routing erfolgt mit Hilfe der Property „endpointURI“ auf der Invoke-Aktivität des Service-Calls.

Konzeptionell lässt sich Ähnliches natürlich auch mit einem Service Bus oder der Mediator-Komponente realisieren. Die im Vortrag gezeigte Methode eignet sich allerdings besser in speziellen Szenarien bzw. wenn kein Service Bus in der vorhandenen Architektur existiert. Nur so kann zur Laufzeit der tatsächliche physikalische Endpunkt eines Services beeinflusst werden. Der Vorteil gegenüber einem Mediator oder Service Bus ist hierbei, dass kein Routing und keine Transformation für die Integration eines zusätzlichen Endpunktes konfiguriert bzw. geändert werden muss und somit eine dynamische Skalierung möglich ist.

### Customizing von Oberflächen für zentrale Postkorbapplikationen (Aufgabenverwaltung, Workflow)

Im Vortrag zeigen wir dieses unter Verwendung der Human Workflow Services und der zugrundeliegenden API, die von der Oracle BPM bzw. SOA Suite zur Verfügung gestellt wird. Die folgende Abbildung skizziert die einzelnen Teilkomponenten.

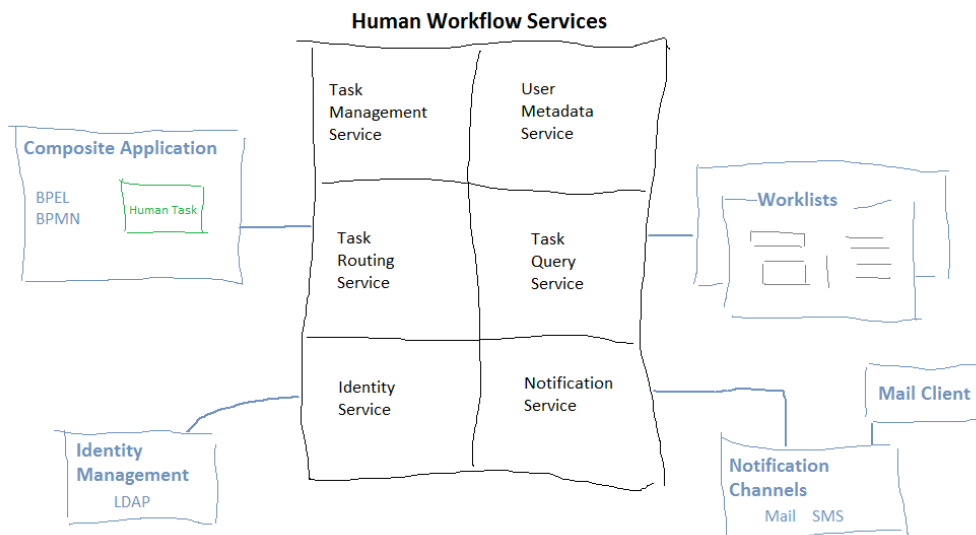


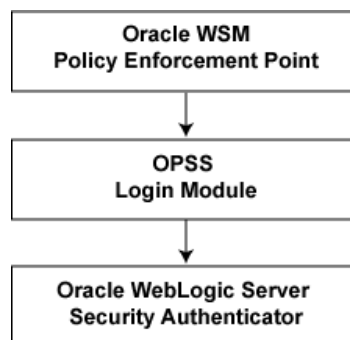
Abb. 4: Teilkomponenten der Human Workflow Services

- Task Management Service: Speicherung von Ausführungsdaten von Human Tasks, Formularen und Attachments; dient der Versionsverfolgung und hält den AuditTrail vor

- Task Routing Service: Zuweisung von Tasks zu den entsprechenden Benutzern mit Rücksicht auf das ausgewählte Pattern
- Identity Service: Authentifizierung von Benutzern sowie Lookup von Benutzer-Properties, Rollen und Berechtigungen im angebundenen LDAP-System
- User Metadata Service: Verwaltet Metadaten, die mit Human Workflow Benutzern in Verbindung stehen (z.B. Benutzereigenschaften, Urlaubsvertretungs- und Delegation-Rules)
- Task Query Service: Bietet Operationen für die Abfrage von Task-Listen bzw. Task-Details
- Notification Service: Dient der Benachrichtigung von Benutzern über verschiedene Kanäle (Mail, SMS, Fax und Voice)

### **OWSM-Policies**

Oracle Web Services Manager (OWSM) stellt ein Policy Framework zur Verfügung, mit dem Web Services konsistent gesichert und gemanaged werden können. Das Policy Framework basiert auf dem WS-Policy Standard. Die grundlegende Architektur wird in der folgenden Abbildung skizziert. Dabei verwendet ein Oracle WSM Policy Enforcement Point (PEP) das Oracle Plattform Security Service (OPSS) Login Module und den Oracle WebLogic Server Security Authenticator zur Authentifizierung und Authorisierung.



*Abb. 5: OWSM, OPSS Login Module und WLS Security Authenticator*

Im Wesentlichen stellt der OWSM die folgenden Funktionalitäten zur Verfügung:

- Handle WS-Security (for example, encryption, decryption, signing, signature validation, ...)
- Define authentication and authorization policies against an LDAP directory.
- Generate standard security tokens (such as SAML tokens) to propagate identities across multiple Web services used in a single transaction.
- Segment policies into different namespaces by creating policies within different folders.
- Examine log files.

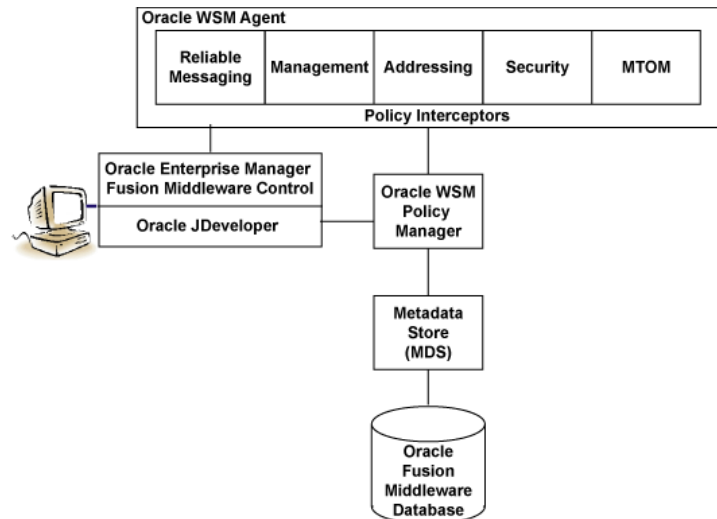


Abb.6:Komponenten der Oracle WSM Architektur

Die praktische Verwendung des OWSM werden wir in der Live Demo zeigen. Weitere Informationen zu dem Produkt finden Sie im Oracle Fusion Middleware Security and Administrator's Guide for Web Service unter: [http://download.oracle.com/docs/cd/E12839\\_01/web.1111/b32511/toc.htm](http://download.oracle.com/docs/cd/E12839_01/web.1111/b32511/toc.htm).

#### Kontaktadressen:

Danilo Schmiedel  
 OPITZ CONSULTING Berlin GmbH  
 Tempelhofer Weg 64  
 D-12347 Berlin

Telefon: +49 (0)30 6298889-1632  
 Mobil: +49 (0)173 7279001  
 E-Mail: [danilo.schmiedel@opitz-consulting.com](mailto:danilo.schmiedel@opitz-consulting.com)  
 Internet: [www.opitz-consulting.com](http://www.opitz-consulting.com)

Dr. Jens Hündling  
 ORACLE Deutschland B.V. & Co. KG  
 Schiffbauergasse 14  
 D-14467 Potsdam

Telefon: +49 (0) 331-200 72 09  
 Mobil: +49 (0) 172-441 26 91  
 E-Mail: [jens.huendling@oracle.com](mailto:jens.huendling@oracle.com)  
 Internet: [www.oracle.com](http://www.oracle.com)