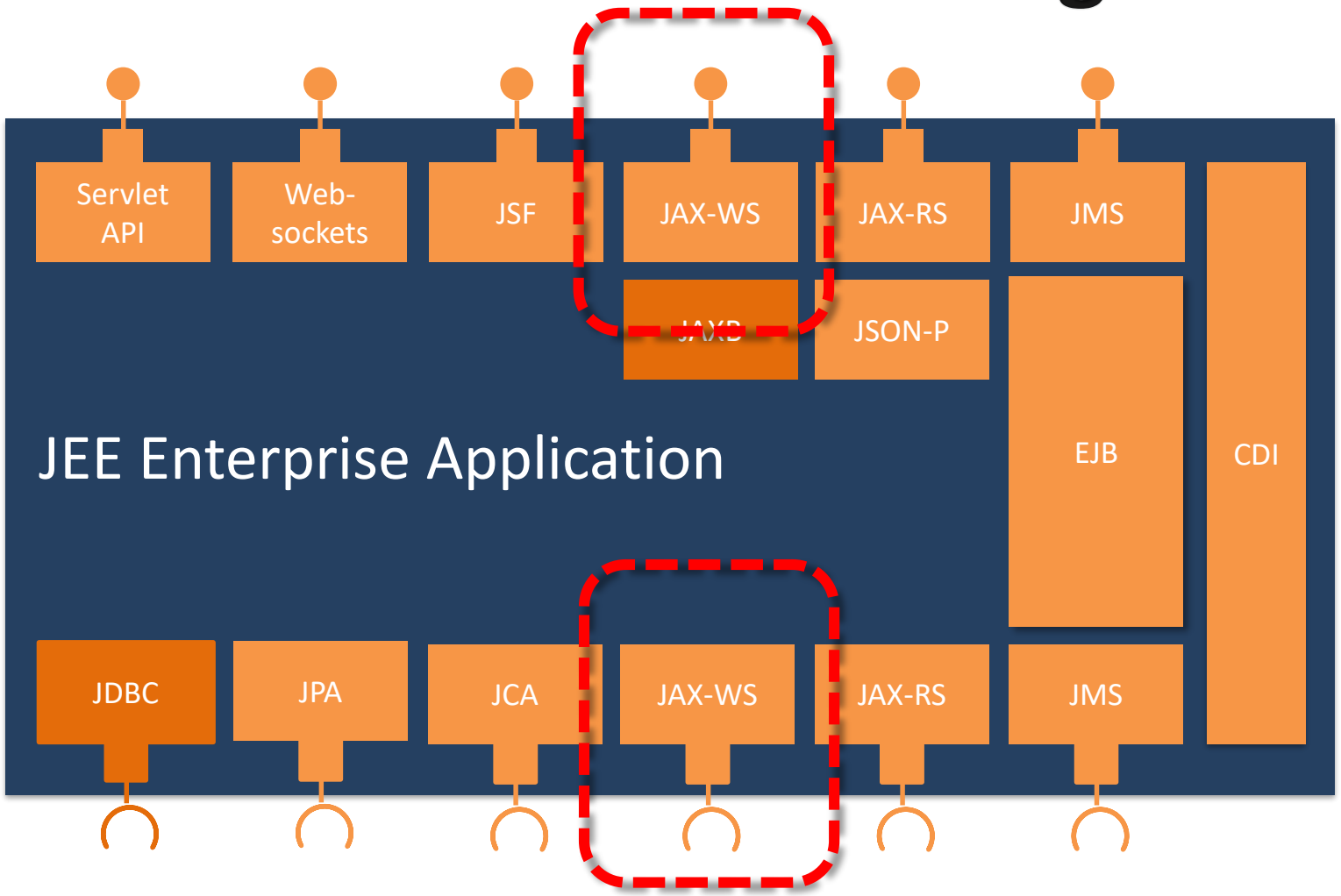
SOAP-Webservices mit JAX-WS

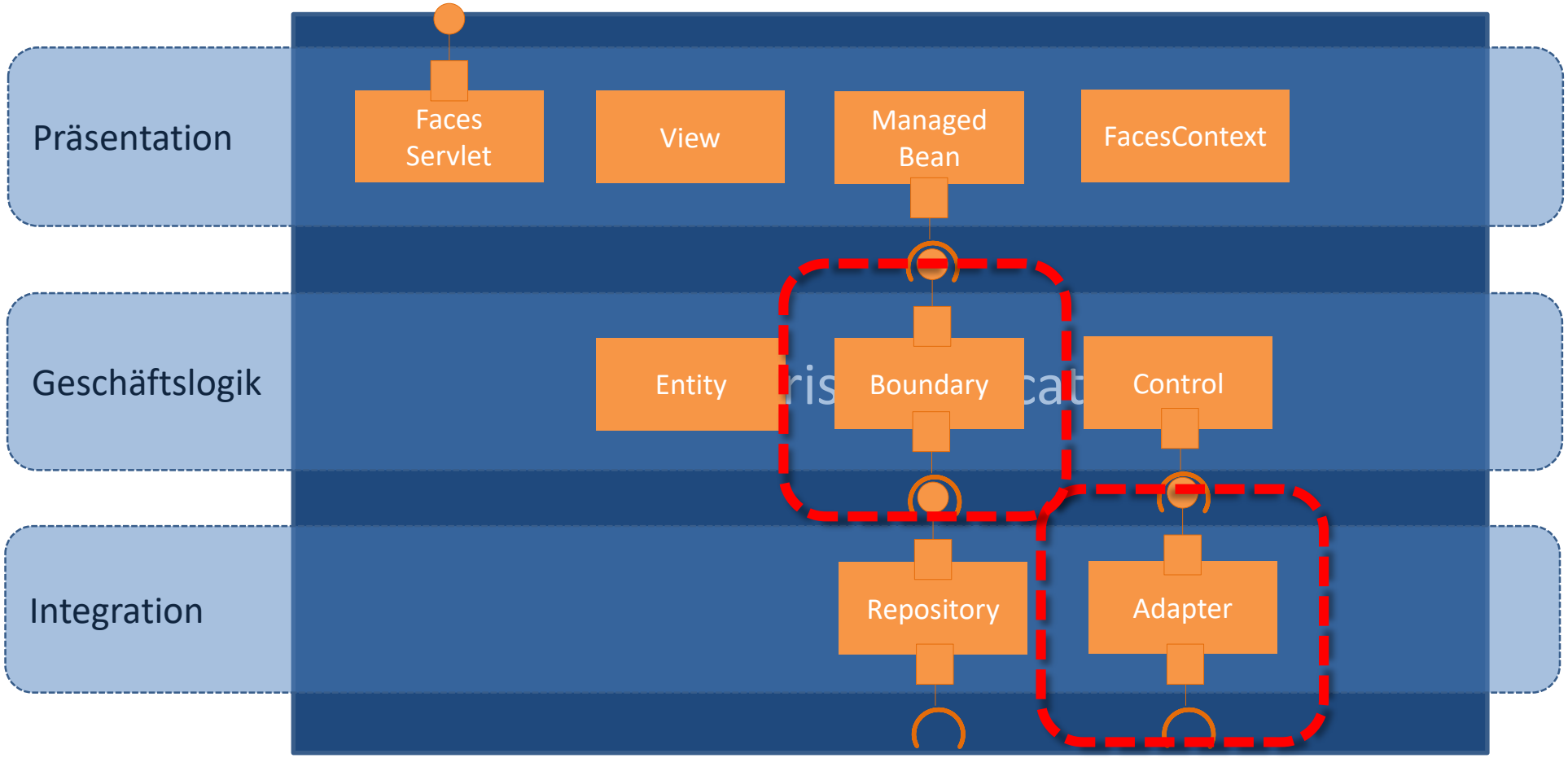
FWP Aktuelle Technologien zur Entwicklung verteilter Java-Anwendungen

GRUNDLAGEN VON SOAP-BASIERTEN WEBSERVICES

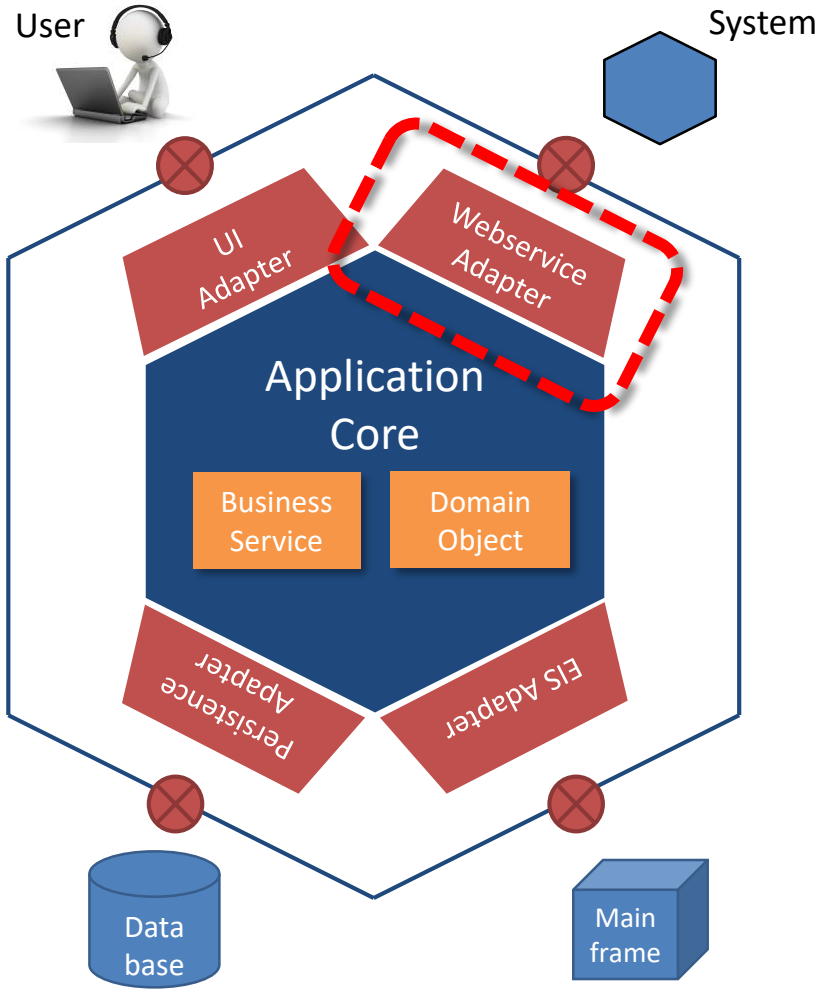
Kontext: JEE Technologien



Kontext: Einheitliches Schichtenmodell



Kontext: Hexagonale Architektur



Die Macht von Webservices

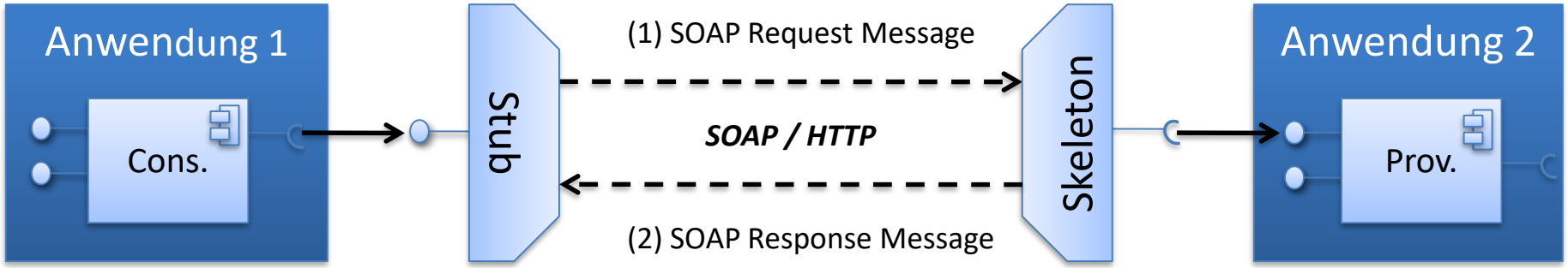
- Webservices helfen bei der Integration von Applikationen aus unterschiedlichen technologischen Plattformen
- Webservices sind der Defacto-Standard für lose gekoppelten Applikationen, aus denen SOA-basierte IT-Landschaften aufgebaut werden können
- Kommunikation basiert auf standardisierten Nachrichten (SOAP)
- Schnittstelle ist durch einen formellen Vertrag (WSDL) definiert

WS*-Standards: mehr Fluch als Segen

- Kommunikation über SOAP wird durch eine Vielzahl von Standards mit weiteren Features ergänzt
  - Security
  - Transaktionen
  - Zuverlässige Zustellung
  - …
- Vielzahl, mangelnde Interoperabilität und Unübersichtlichkeit haben zur sinkenden Bedeutung von SOAP-WS beigetragen

Kommunikation über SOAP-Nachrichten

- Service Consumer generiert aus WSDL spezifischen Stub
- Service Provider generiert aus WSDL spezifischen Skeleton
  ODER
  Service Provider generiert aus Klassen und Interfaces WSDL
- Stub und Skeletons übersetzen Methodenaufrufe in Nachrichten-Kommunikation (RPC)



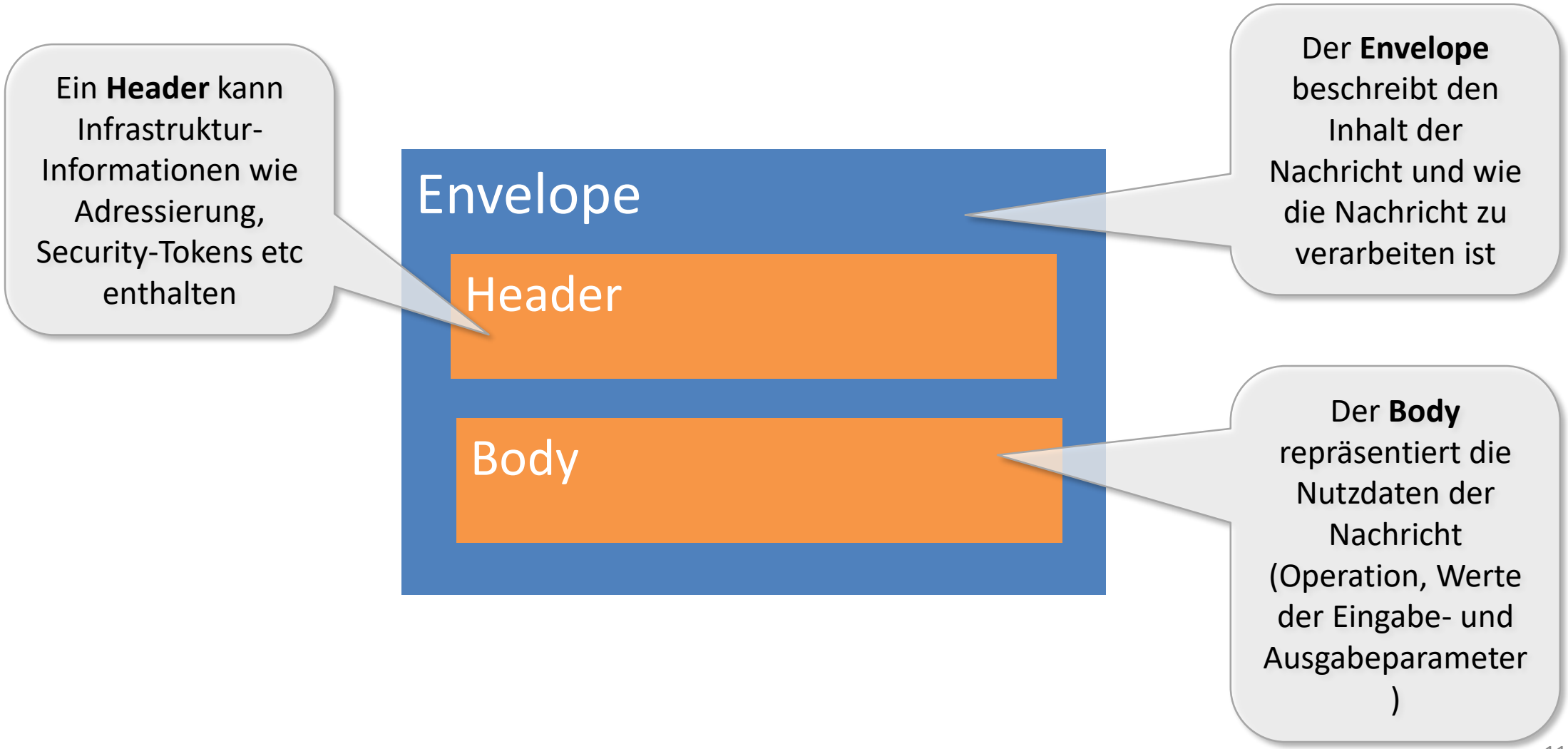
Webservice Definition Language (WSDL)

- Standardisiertes XML-Dokument, welches den formalen Vertrag zwischen Service Provider und Service Consumer definiert
  - Interfaces and Operationen
  - Struktur der SOAP-Request- und SOAP-Response-Nachrichten (header/bodies)
  - Struktur der Nutzdaten in den SOAP-Nachrichten
  - Webservice Stil
  - Endpunktadresse

Simple Object Access Protocol (SOAP)

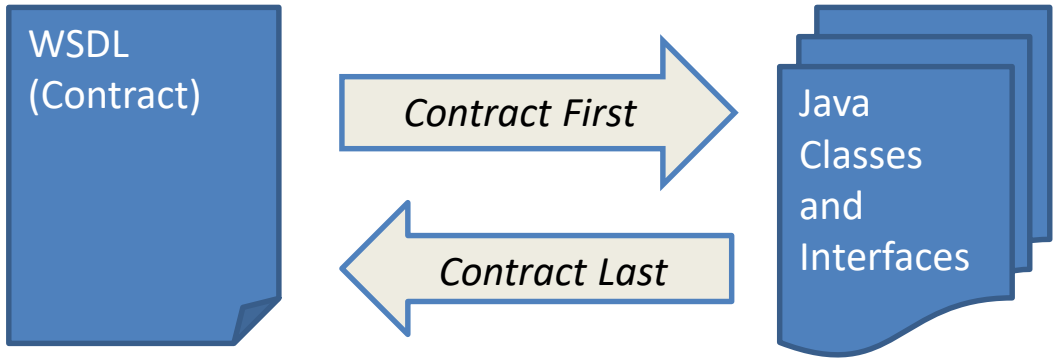
- Request- und Response-Nachrichten werden als standardisierte SOAP-konforme XML-Dokumente übertragen
- In den meisten Fällen werden SOAP-Nachrichten über eine HTTP(S)-Verbindung ausgetauscht
  - Andere Verbindungstypen sind aber möglich (SOAP/JMS, SOAP/RMI)
- Jede SOAP-Nachricht ist intern gleich aufgebaut

Aufbau von SOAP-Nachrichten



Zwei Strategien zur Implementierung

- Contract First: Beginne mit einer WSDL und generiere daraus Java-Klassen und –Interfaces
- Contract Last: Beginne mit annotierten Java-Klassen und –Interfaces und generiere daraus die WSDL



Contract First vs Contract Last

Contract First

- Stellt die höchste Kompatibilität sicher
- Ermöglicht kontrollierte Weiter-entwicklung des Webservices
- Webservices auf die harte Tour
  - Manuelle Erstellung von WSDLs ist nicht leicht und erfordert umfangreiche Kenntnisse

Contract Last

- Änderungen im Java-Code können unerwünschte Auswirkungen auf die WSDL haben
  - Verletzung der Kompatibilität zu Vorgängerversionen
  - Konsumenten mit einer älteren Version des Webservices können diesen plötzlich nicht mehr aufrufen
- Webservices auf die leichte Tour
  - Alle Laufzeitumgebungen für Webservices können WSDLs on-the-fly generieren

WS-Stil Document Literal Wrapped

- Garantiert höchstmögliche Interoperabilität
- Regeln sind ziemlich einfach
  - Request- und Response dürfen jeweils nur einen Message Part enthalten
  - Jeder Message Part muss sich auf ein Wrapper-Element beziehen
  - Ein Wrapper muss als Complex Type bestehend aus einer Sequenz von Elementen definiert werden; jedes Element entspricht einem Parameter
  - Der Name des Request-Wrappers muss dem Name der aufzurufenden Operation entsprechen; der Name des Response-Wrappers muss aus dem Namen der aufgerufenen Operation plus dem Anhang “Response” bestehen
  - Der Response-Wrapper sollte nur ein Element beinhalten, das den Rückgabewert der Operation repräsentiert
  - Im soap:binding muss style=“document” und im soap:body muss use=“literal” angegeben werden

WEBSERVICES IN JAVA MIT JAX-WS

Java API for XML Web Services (JAX-WS)

- Standard für SOAP-basierte Webservices in Java
- Seit Java 6 Bestandteil der JavaSE Runtime
- Übernimmt Kommunikation über SOAP-Nachrichten
- Führt das Object-XML-Mapping über JAXB durch
- Java-Klassen können mit Annotationen zu SOAP-Endpunkten gemacht werden
- Unterstützt RPC- und nachrichten-orientierte WS-Schnittstellen
- Bietet Werkzeuge fürs Generieren von Java-Code und WSDLs

Webservice-Binding mit Annotationen

Annotation	Beschreibung
@WebService	Bietet jede beliebige Klasse als Webservice an
@SOAPBinding	Definiert das SOAP-Binding (document literal wrapped als Voreinstellung)
@WebMethod	Stellt eine Methode als eine Webservice-Operation zur Verfügung (was bei einer mit @WebService annotierten Klasse automatisch für alle Methoden gilt)
@WebParam	Definiert einen Parameter einer Webservice-Operation (was automatisch für alle Parameter einer Operation gilt, wobei allerdings kryptische Namen wie arg0 verwendet werden)
@WebResult	Definiert den Rückgabewert (optional)

wsgen zur Generierung von WSDL

- Tool der Wahl, um aus annotierten Java-Klassen eine WSDL zu generieren
- Liest eine Webservice-Endpunkt-Klasse und generiert alle Artefakte, die für das Deployment und für den Aufruf des Webservices erforderlich sind

wsimport zur Generierung von Java-Code

- Tool der Wahl, wenn aus einer WSDL einen Webservice-Client zu generieren
- Erzeugt portable Artefakte wie
  - Service Endpoint Interface (SEI) = Java-Ausprägung des Webservice-Interfaces
  - Service Klasse = Factory für die Erzeugung von Webservice-Stubs
  - Exception-Klasse aus wsdl:faults
  - Klassen für den asynchronen Aufruf von Operationen
  - JAXB-basierte Java-Klassen für alle komplexen Parametertypen

Wie exportiere ich SOAP-Webservices in einer JEE-Applikation mit JAX-WS?

BEREITSTELLEN VON SOAP-WEBSERVICES

Wie rufe ich in einer JEE-Applikation SOAP-Webservices auf?

KONSUMIEREN VON WEBSERVICES

Fragen?

ANHANG

Quellen

- Beispiel-Code auf GitHub unter https://github.com/mikeT92/jeetrain
  MAVEN-Projekt **jeedemo** MAVEN-Modul **jeedemo-jaxws**
- Eric Jendrock et. al.: **The Java EE 7 Tutorial Part VI Webservices Chapter 28**
  http://docs.oracle.com/javaee/7/tutorial/jaxws.htm
  Oracle September 2014

Kontakt

Michael Theis
Lehrbeauftragter Hochschule München

email michael.theis@hm.edu
mobile + 49 170 5403805
web http://www.tschutschu.de/Lehrauftrag.html