

Course Export - Service orientierte Architekturen

University

Semester

Created By

vabene1111

01 Einführung

Was ist ein Dienst (klassisch und in der IT) ?

Klassisch: Eine durchgängige Handlung on Mensch zu Mensch

IT:

- Eine Abstraktion für das Anbieten von Soft- und Hardware über ein Netzwerk
- Unabhängig von Hardware, Programmiersprache oder Betriebssystem

Was ist ein Service ?

- Ein Paradigma der Machine-to-Machine Kommunikation
- sind plattformunabhängige, wiederverwendbare und gekapselte Software Komponenten
- bieten lokal und/oder über ein Netzwerk nutzbare Schnittstellen

Nennen und Erklären sie die Merkmale eines Services!

- **M2M:** Services sind ein Paradigma der Machine-to-Machine Kommunikation
- **Wiederverwendbarkeit:** Services können nach der Nutzung freigegeben oder von Anfang an Zustandslos sein
- **Kapselung:** Implementierung ist hinter Schnittstelle verborgen, für den Nutzer irrelevante Informationen werden verborgen
- **Nutzung über Netzwerk:** Kann in (lokalem) Netzwerk genutzt werden
- **Plattformunabhängigkeit:** in unterschiedlichen Laufzeiten, unabhängig von spezifischen Implementierungen nutzbar

Was ist eine Service orientierte Architektur ?

Eine **Systemarchitektur** die vielfältige, verschiedene und ggf. inkompatible Methoden oder Anwendungen als wiederverwendbare und offen zugreifbare **Services** repräsentiert. Dadurch werden alle **Merkmale** eines Services erfüllt.

Beschreiben sie den SoA Tempel und erläutern sie die wichtigsten Merkmale!

Merkmale (Säulen)

- **Prozessorientiert:** kapselt Service Implementierung, Fokus auf Funktionalität
- **Verteilte Anwendungen:** mehrere Netzwerkteilnehmer können Services nutzen und anbieten
- **Dynamische Bindung:** Consumer können, über einen Broker, Provider (und damit Services) finden
- **Lose Kopplung:** einbinden von Services zur Laufzeit dank klarer Schnittstellenbeschreibungen

Erklären sie die Komponenten des einfachen SoA Dreiecks!

Was ist eine Service Beschreibung und woraus besteht sie ?

- **funktionale** Beschreibung: Signatur(Eingabeparameter, was kommt zurück)
- **nicht-funktionale** Beschreibung: Service Level Agreement (z.B. Antwortzeit, Kosten)

Was ist ein ESB und welche Funktionen erfüllt er ?

Keine Eindeutige definition, quasi die Eierlegendewollmilchsau als Vermittler zwischen Entitäten die nicht die gleichen Protokolle sprechen.

Kann als **Bestandteil** einer SOA (einfacher Vermittler) oder als **Weiterentwicklung** von SOA (Funktionen wie Autorisierung oder Transformation) gesehen werden.

Aufgaben sind u.a.:

- Routing und Transformation von Nachrichten
- Protokolltransformation und Integration mithilfe von Standards
- Autorisierung und Authentifizierung
- Quality of Service

Erklären sie den Unterschied zwischen Programmieren im Kleinen und im Großen

Programmieren im Kleinen

- Überschaubare Komponenten mit exakten Schnittstellen- & Funktionsdefinitionen
- beliebige Programmiersprache
- Outsourcing möglich/sinnvoll

Programmieren im Großen

- Zusammenwirken der Komponenten
- abstrakte Beschreibung des Gesamtsystems
- realistisches Modell der Geschäftsprozesse
- kein Outsourcing da Systemkenntnis erforderlich

Nennen sie ****Vorteile**** von SOA!

- Nutzung von **Standards** und damit einhergehende Interoperabilität und einfache Entwicklung
- Unabhängigkeit von Plattformen und Sprachen
- Code Wiederverwendbarkeit
- Schnelle Anpassbarkeit
- einfache Bereitstellung, Wartung und Entwicklung von Anwendungen
- ermöglicht Enterprise Application Integration durch M2M Kommunikation

Nennen sie ****Nachteile**** von SOA!

- dynamische Architektur kann Instabilität verursachen
- wenig Homogenität, ggf. Vermittler (ESB) notwendig
- Einführung sehr Aufwändig
- Extrem schwierige Echtzeitfähigkeit

Unter welchen Umständen ist das Einführen einer SOA ****nicht**** notwendig/sinnvoll ?

- wenn **bereits Homogene** IT vorhanden ist
- wenn **keine Flexibilität** in der IT notwendig ist
- wenn **enge Kopplung** benötigt wird
- bei harten **Echtzeitanforderungen**
- wenn die **organisatorischen Voraussetzungen** nicht gegeben sind

Was sind Service Level Agreements (SLA) ?

Im SOA/IT Kontext: Festlegung von Leistungseigenschaften wie z.B. Verfügbarkeit, Antwortzeiten, Ausfallraten, ... und die klare Vereinbarung/Bekanntmachung dieser.

Definition recht frei, gerne Ergänzen

Wie funktionieren die folgenden Architekturprinzipien - Peer-to-Peer - Publisher-Subscriber - Client-Server

- Peer-to-Peer: Direkte Kommunikation zwischen mehreren Geräten ohne Zentrale Instanz (z.B. Torrent)
- Publisher-Subscriber: Client pflegt Subscriptions die Regelmäßig bei Publishern abgeholt werden (üblicherweise RSS Feeds/Podcasts)
- Client-Server: Client('s) stellen Anfragen an Zentralen Server und erhalten entsprechende Antworten (z.B. Website)

02 Softwarearchitekturen

Was sind häufig Probleme und deren Ursachen bei klassischer Softwareentwicklung ?

Durch u.a.

- iterative Anforderungsänderungen
- Zeitdruck
- Mangelnde Dokumentation

entsteht ein "Big Ball of Mud" mit

- unkontrollierbarer Komplexität
- wenig Wiederverwendbarkeit und Flexibilität

- schwieriger Wartbarkeit und Planbarkeit
- schlechter Performance und schwieriger Integration

Softwarearchitekturen helfen genau das zu verhindern!

Was ist eine **Softwarearchitektur** und welche Grundlegenden Fragen beantwortet sie ?

Eine Softwarearchitektur legt die "tragenden Säulen" eines IT-Systems fest ohne ausführliche Details zu geben. Typische Fragen sind

- Welche Anforderungen führen zu bestimmten Strukturen und Entscheidungen ?
- Welche wesentlichen logischen und physikalischen Bausteine gibt es ?
- Welche Schnittstellen und Abhängigkeiten gibt es ?
- Nach welchen Kriterien wird das System in Bausteine aufgegliedert ?

Was sind die 6 **Dimensionen** der Softwarearchitektur und wie hängen sie zusammen ?

1. Was: Architekturen und Disziplinen
2. Wo: Perspektiven
3. Womit: Mittel
4. Warum: Anforderungen
5. Wer: Organisationen/Individuen
6. Wie: Vorgehen

Beschreiben sie die **"Was" Dimension von Softwarearchitekturen**

Legt die Grundlagen und Definitionen für die Arbeit als Architekt. Ordnet die Architektur in 7 Disziplinen.

Architektur ist die Summe aus Struktur und Herangehensweise

- **Struktur:** welche Anforderungen und Mittel beeinflussen die Architektur und damit die Struktur des Produktes
- **Herangehensweise:** Welchen Anforderungen beeinflussen den Architektonischen Prozess

Welche Disziplinen der Architektur gibt es neben der Softwarearchitektur noch ?

- **Netzwerkarchitektur:** Planung und Entwurf von Netzwerkfunktionen, -Diensten, ...
- **Integrationsarchitektur:** Planung und Realisierung der Verbindung von (heterogenen) Systemen
- **Datenarchitektur:** Entwurf logischer und physischer Datenmodelle, Auswahl von Systemen
- **Sicherheitsarchitektur:** Gewährleistung aller Sicherheitsaspekte (Vertraulichkeit, Integrität, Verfügbarkeit), alles rund um Auth und Autorisierung (SSO, PKI, ...)
- **System-Management-Architektur:** operationale Aspekte entwerfen (Betriebsstrategien, SLA's)
- **Enterprise-Architektur:** Eingliederung in Geschäftsprozesse, organisatorische und rechtliche Anforderungen, Change Management

Beschreiben sie die **"Wo" Dimension von Softwarearchitekturen!**

"Wo" beschreibt die **Perspektiven**, also die unterschiedlichen Ebenen und Sichtweisen ein Architektur.

Ebenen sind abhängig vom Use-Case, z.B.

- Organisationsebene
- Systemebene (betrachtet IT-Systeme der Organisation)
- Bausteinebene (betrachtet Bausteine einzelner Systeme)

Verschiedene Sichten für unterschiedliche Zwecke/Betrachter

- Anforderungssicht (Architektur Anforderungen)
- Logische Sicht (logische Strukturen des Systems)
- Datensicht (Datenstrukturen und -flüsse)
- Umsetzungssicht (Umstellungsstruktur und Umsetzungsinfrastruktur)
- Prozesssicht (Steuerung und Koordination der Bausteine)
- Verteilungssicht (physikalische Verteilung der Software Bausteine)

Beschreiben sie die **"Warum" Dimension von Softwarearchitekturen!**

Softwarearchitekturen werden auf basis und Grund von Anforderungen durch Kunden, Gesetze, Märkte, technischer Machbarkeit, usw. entwickelt.

Wichtige Anforderungen an Anforderungen in der Softwarearchitektur sind:

- **Korrektheit:** Inhaltlich sinnvolle, mit allen Stakeholdern besprochene Anforderung

- **Machbarkeit:** unter gegebenen Rahmenbedingungen realisierbar
- **Eindeutigkeit:** Einfache und unmissverständliche Formulierung
- **Nachprüfbarkeit:** Erfüllung der Anforderung muss anhand von Kriterien prüfbar sein
- **Vollständigkeit:** Alle Anforderungen und Fehlerfälle aufgenommen, ggf. von dritten überprüft
- **Konsistenz:** Anforderungen widersprechen sich nicht

Beschreiben sie das "Womit" Dimension von Softwarearchitekturen!

Es gibt verschiedene Architektonische Werkzeuge/Referenzmodelle, beispielsweise Service orientierte Architekturen, mit denen Softwarearchitekturen entwickelt werden können.

Es gibt verschiedene Einflussfaktoren (von wichtig zu weniger wichtig):

1. Referenzarchitekturen: Implementierungen von Basisarchitekturen für eine konkrete Domäne
2. Basisarchitekturen: Konkrete Architekturen (SOA, Client/Server, ..)
3. Konzepte: Ansätze zur Einhaltung von Prinzipien (Objektorientierung, Code Generierung, ..)
4. Prinzipien: Bewährte Grundlagen (loose Kopplung, Modularität, ...)

Beschreiben sie das "Wer" Dimension von Softwarearchitekturen!

Wie interagiert der Architekt mit anderen beteiligten Parteien ?
Befasst sich mit den sozialen und organisatorischen Aspekten.

Beschreiben sie die "Wie" Dimension von Softwarearchitekturen!

Strukturierung des Architektonischen Vorgehens z.B. anhand von Vorgehensmodellen.

Was ist ein IT-System ?

Ein (IT-)System ist eine Gesamtheit von Elementen (Systembausteine) die auf eine Weise Wechselwirken das sie als Einheit angesehen werden kann. Gemeinsam erfüllen sie eine Aufgabe, Zweck oder Sinn.

Systeme haben Ziele und ihre Bausteine können wiederum Systeme sein. Sie haben offene oder geschlossene Grenzen zur Umwelt.

Ein System wird durch eine Softwarearchitektur (vor)strukturiert!

Auf welchen Architekturebenen liegen die einzelnen Architektursichten?

- Anforderungssicht (Architektur-Anforderungen): auf **Organisationsebene**
- Logische Sicht (logische Struktur des Systems): auf **Systemebene**
- Datensicht (Datenstrukturen und -Flüsse) : auf **Organisationsebene**
- Umsetzungssicht (Umsetzungsstruktur und Umsetzungsinfrastruktur): auf **Bausteinebene**
- Prozesssicht (Steuerung und Koordination) : auf **Bausteinebene/Systemebene**
- Verteilungssicht (physikalische Verteilung der Software-Bausteine): **alle 3 Ebenen**

03 Entwurfsprinzipien

Nennen sie die 8 Entwurfsprinzipien!

- Abstraktion
- Autonomie
- Auffindbarkeit
- Kompositionsfähigkeit
- Kopplung von Services
- Service Verträge
- Wiederverwendbarkeit
- Zustandslosigkeit

Was sind Service Verträge und welche Schwierigkeiten ergeben sich ?

Service Verträge sind formale und standardisierte Beschreibungen des Zwecks und der Fähigkeiten eines Services. Sie werden üblicherweise mit dem Service ausgeliefert.

Enthalten semantische Informationen (z.B. Signaturen, WSDL, ..) und nicht technische Informationen (z.B. SLA).

Sie erhöhen Interoperabilität, sorgen für einfache Verständlichkeit und können Organisationsweit standardisierte Datentypen und Policies enthalten.

Schwierigkeiten sind

- Versionierung: Langlebige Services müssen auf Änderungen reagieren
- Technologieabhängigkeiten
- Unausgereifte Werkzeuge verursachen viele nicht standardisierte Verträge

Was versteht man unter dem Entwurfsprinzip ****Kopplung von Services**** ?

Kopplung zwischen Service und Kontext (Systemen, Umgebung, anderen Komponenten) sowohl uni- als auch Bidirektional sollte bei SOA minimiert werden.

Auch die Abhängigkeit von Service Verträgen zu ihrer Konkreten Implementierung sollte minimiert werden.

Kopplung kann zwischen verschiedenen Entitäten existieren

- Logik an Vertrag
- Vertrag and Implementierungstechnologie
- Vertrag an externe Logik
- Consumer an direkte Implementierung eines Services (am Vertrag vorbei)
- Consumer an einen Vertrag

Lose Kopplung erhöht Laufzeitaufwand und muss mit anderen Entwurfsprinzipien in Balance gehalten werden.

Was versteht man unter dem Entwurfsprinzip ****Abstraktion****?

Unnötige Informationen verbergen, notwendige Informationen veröffentlichen.

Typen von Informationen die, je nach Sicht, abstrahiert werden können

- technologische Informationen
- funktionale Informationen
- Informationen zur Programmlogik
- Informationen zur Service Qualität

Richtiges Maß an Abstraktion ist die wesentliche Schwierigkeit.

Was versteht man unter dem Entwurfsprinzip ****Wiederverwendbarkeit****?

Nutzbarkeit eines Services für mehrere Zwecke erhöht den Mehrwert erschwert jedoch die Weiterentwicklung und erhöht die Komplexität von Verträgen, Versionierung und Laufzeitumgebung.

Grade von Wiederverwendbarkeit

- **taktisch**: nur notwendiges implementiert und auch nur das Wiederverwendbar
- **zielgerichtet**: absehbare Erweiterungen mit hohem Potential provisorisch mitentwickelt
- **vollständig**: alle denkbaren Funktionen eines Services implementiert für maximale Wiederverwendbarkeit

Was versteht man unter dem Entwurfsprinzip ****Autonomie****?

Erhöht Zuverlässigkeit, Performance und Berechenbarkeit eines Services wenn keine anderen Einflüsse existieren.

Zwei Dimensionen Umfassen Autonomie

- zur Laufzeit: konsistent gute Performance, berechenbares Verhalten, Isolationsmöglichkeiten
- beim Entwurf: Skalierbarkeit, einfache Wartung und Weiterentwicklung

Was versteht man unter dem Entwurfsprinzip ****Zustandslosigkeit****?

Durch Zustandslosigkeit erhöhte Wiederverwendbarkeit, Schonung von Ressourcen und erleichtert Skalierung.

Verschiedene Grade von Zustandslosigkeit umfassen

- geringe/keine: Lange Laufzeit, Kontextbasierte Service Regeln
- reduziert: lagert ggf. Zustandsdaten aus oder behält Zustandsdaten nur bei langer Laufzeit
- mittlere: Service versucht im Leerlauf Zustandslos zu sein
- hohe: vollständige Verschiebung der Zustandsverwaltung, Service nutzt jede Möglichkeit um Zustandslos zu werden

Komplexität und Kosten steigen mit Grad der Zustandslosigkeit und sind schwer abschätzbar.

Was versteht man unter dem Entwurfsprinzip ****Auffindbarkeit****?

Discovery Prozess um zu verhindern das existierende Services übersehen werden. Möglichkeit die Funktion eines gefunden Services zu interpretieren.

Services können vom Entwickler beim Entwurf eines Consumers (**Entwurfszeit-Discovery**) oder durch den Consumer direkt zur Laufzeit durch ein Verzeichnis gefunden werden. Die Anforderungen an den Service Vertrag und seinen Standardisierung sind bei **Laufzeit-Discovery** sehr hoch.

Was versteht man unter dem Entwurfsprinzip ****Kompositionsfähigkeit****, was sind Schwierigkeiten ?

Optimale Wiederverwendbarkeit durch Zerlegung und Komposition von Anwendungslogik in kleinere Bestandteile.

Schwierigkeiten

- hohe Anforderungen an Laufzeitumgebung (Skalierbarkeit, Zuverlässigkeit)
- Jedes Element einer Komposition ist eine Quelle für Fehler oder für Performance Engpässe
- Komplexe Abhängigkeiten in Kompositionen machen Weiterentwicklung schwierig

Nennen sie ein Beispiel für jedes der 8 Entwurfsprinzipien!

Mögliche Beispiele:

- Service-Verträge: Web Services Service Vertrag (WSDL + SLA)
- Kopplung von Services: Client wurde direkt für Service implementiert und ist nur mit ihm kompatibel
- Abstraktion: Taschenrechner Service abstrahiert Programmiersprache aber nicht welche Verfahren verfügbar sind
- Wiederverwendbarkeit: Taschenrechner Service der von Lagerverwaltung und Rechnungswesen genutzt wird
- Autonomie: Drucker Service der Autonom auf dem Drucker läuft und nur eine klare Schnittstelle nach außen hat
- Zustandslosigkeit: Taschenrechner Service der nur Zahlen zwischenspeichert und danach wieder Zustandslos wird
- Auffindbarkeit: SOA Dreieck, Provider published im Verzeichnis, Consumer findet
- Kompositionsfähigkeit: Taschenrechner Service nutzt andere Services für Visualisierung oder Authentifizierung

04 Service orientierte Organisationen

Beschreiben sie Ziele und das Grundlegende vorgehen von Service orientierten Organisationen!

Unternehmen sind Hauptzielgruppe für SOA. Die Flexibilität soll gesteigert und die Umsetzung von IT Anforderungen beschleunigt werden. Legacy Anwendungssysteme ("Insellösungen") werden als Service gekapselt um flexibel in Prozessen genutzt werden zu können.

Ein gutes Anforderungsmanagement ist essenziell und Kompromisse und Entscheidungen darüber was Services werden sollen und wie viele Services notwendig sind notwendig.

Beschreiben sie die Phasen des SOA Lebenszyklus

Design

- Benötigte Services definieren, implementieren bzw. legacy systeme kapseln und zuletzt die neue Anwendung entwickeln
- Wahl eines realisierbaren aber auch gewinnbringenden Abdeckungsbereichs

Betriebs und Modifikationsphase

- Inbetriebnahme und täglicher Betrieb
- Anpassung an neue Anforderungen oder Technologien

Service Sundown

- Abhängigkeiten frühzeitig beachten
- Services üblicherweise 30 Jahre in Betrieb

Was versteht man unter ****SOA-Readiness**** und wieso ist es wichtig ?

SOA Readiness bewertet ob die Voraussetzungen für ein Unternehmen für die Einführung oder Weiterentwicklung von SOA gegeben sind.

Change Management Tätigkeiten wie Schulungen, Participation, Erklärung der Vorteile, etc. müssen gegeben sein. Es muss außerdem Geschäftsprozessmanagement betrieben werden.

Kommunikation wird als wesentliches Erfolgskriterium angesehen, wichtig ist auch das das ganze Unternehmen nach **einem globalen optimum** strebt.

Zur Analyse kann z.B. das SOA Maturity Modell benutzt werden.

Erklären sie das Vorgehensmodell bei der SOA-Einführung

Definition Projektziele und Vorgehensweise

- Projektplanung
- Grunderklärung SOA

Bestimmung geeigneter Technologien

- Was kann wiederverwendet werden was schon da ist
- Einsatz ESB ?

Aufbau von SOA Know-How

- Schulung von Beteiligten aus allen Unternehmensbereichen (IT, Spezialisten, Management)
- Simulation von Diensten und Abläufen aus verschiedenen Perspektiven

Sukzessive Annäherung an die Ziel-Architektur

- Stück für Stück, Bottom-Up System aufbauen
- Entwicklung aus IT Perspektive unter Berücksichtigung von Business Anforderungen

Wofür ist eine Return on Investment ROI Analyse Sinnvoll und wieso ist sie schwierig im SOA Kontext ?

ROI Analysen helfen Ausgaben zu rechtfertigen indem sie aussagen darüber treffen wann sie sich finanziell lohnen.

Schwierig bei SOA da sie Stabilisierung erfordern und SOA's sich üblicherweise ständig entwickeln. Diverse andere Methoden können einzelne Aspekte bewerten jedoch keine Ganzheitliche Aussage treffen.

Erläutern sie den Unterschied zwischen Säulen und Balken in IT Landschaften

Säulen sind gewachsene Landschaften in denen jedes System eine Säule darstellt die aufwendig über Schnittstellen verbunden ist. Häufig überlappen Inhalte und Daten ohne jedoch ihre implementation/Speicherung zu teilen.

Balken Fokussieren die Systeme auf den eigentlichen Geschäftsprozess, Systeme sind nur einmal implementiert und bieten ihre Funktionalität als Service.

Welche Sichtbarkeitskriterien gibt es bei einer SOA Einführung

- Transparenz & Kommunikation: SOA als Teil der Unternehmensstrategie erkennbar
- Change Management: Informationen und Schulungen anbieten, Einbeziehung fördern
- Nutzen Verständnis: Darstellung der Vorteile durch SOA

Abgrenzung zu Kompetenzen schwierig, siehe VL 04, Folie 27

Welche Potentiale und Herausforderungen bietet SOA im Hinblick auf die Strategische Ebene, die Prozessebene und die IT-Ebene

	Potentiale	Herausforderungen
Strategisch	Agilität Supply-Chain Integration IT/Business besser verknüpfen	uneinheitliches Verständniss von SOA aufwändige Governance langfristigkeit
Prozess	einfacher neue Prozessfunktionalität Prozessautomatisierung Prozessoptimierung	fehlende semantische Darstellung fehlende Standards/Methoden zur modellierung
IT	weniger Wartungsaufwand Mehrfachverwendung weniger Schnittstellenpflege	Performance/Security Probleme aufwändiges Design mangelnde Interoperabilität

Was besagt Conways Gesetz ?

Conways gesetz besagt das eine Organisation die ein System designed etwas produziert was ein Abbild der organisatorischen Kommunikationsstruktur ist.

Durch eine bessere Kommunikationsstruktur, also z.B. dem streben nach einem globalen Optimum, können Systeme dementsprechend verbessert werden.

05 SoA Implementierungen

Welche Basiskonzepte liegen allen SOA Implementierungen zugrunde ?

- Service **Discovery** durch Verzeichnisse
- Service **Beschreibung** durch Bekanntgabe von Schnittstellen (z.B. WSDL)
- Service **Nutzung**
- **Abmelden** eines Services durch Angabe der Gültigkeit von Services
- **Ereignisse** über Benachrichtigungen oder Subscription Services
- **Zusätzliche Komponenten** wie Authentication, Accounting, Privacy, Availability

Was sind Beispiele für SOA Implementierungen?

- Web Services
- UPnP
- Jini
- DNS-SD / Bonjour
- RESTful Web Services
- OSGi

Erklären sie die Folgenden Begriffe im Kontext von XML - Namespace - Parser - DTD - XML Schema - SGML - Grammatik - DOM - SAX

- **Namespace:** Namensräume werden benutzt um Elemente und Attribute eindeutig zu identifizieren.
- **Parser:** Ein Programm welches eine XML Datei einlesen und für die Weiterverarbeitung aufbereiten kann
- **DTD:** Document Type Definition, definiert die verwendeten Datentypen im Dokument
- **XML Schema:** Beschreibt den (standardisierten) Aufbau eines XML Dokuments
- **SGML:** Standard Generalized Markup Language, allgemeine Standard Sprache/Architektur, XML ist eine vereinfachte Version von SGML
- **Grammatik:** Welcher Schreibweise gefolgt wird (XML, SGML, HTML, ..)
- **DOM:** Document Object Model ist ein standardisiertes Verfahren um mit Elementen eines XML (oder HTML) zu interagieren
- **SAX:** Simple API for XML, das de-facto Standard Interface zum Parsen von XML

06 Web Services

Welche Technologien sind typisch für Web-Services eingesetzt ?

- **SOAP:** XML basiertes Nachrichtenformat
- **WSDL:** XML basierte Beschreibung von Services
- **UDDI:** Verzeichnisdienst zur Verwaltung von Webservice Metadaten
- **WSI:** XML basierte Auflistung von Web Service Metadaten

Was sind Web Services ?

Web Services sind Brücken zwischen IT Systemen die von anderer Software (also M2M) üblicherweise über das IP genutzt werden können.

Bezeichnen sie das SOA Dreieck für Web Services mit typischen Technologien ?

Beschreiben sie die Komponenten des Web Service Stack (Schichten)

Von unten nach oben

1. **Transport:** Häufig HTTP/SMTP
2. **XML Spezifikationen:** Gemeinsame XML Datenmodelle
3. **WS Kommunikationsprotokoll:** Eigentlich immer SOAP
4. **Sicherheit:** Verschlüsselung, Authentifizierung und Autorisierung als Erweiterung (WS Security, WS Policy, SAML)
5. **Föderation & Routing:** WS Routing routete XML Nachrichte, WS federation definiert Domänen in denen Dinge gelten (z.B. Sicherheitsregeln)
6. **Integration & Kooperation:** Prozessmodellierung / wie sind Services verbunden
7. **Enterprise Grid:** Integration über geografische Grenzen hinweg

Parallel zu Schicht 1-6 verlaufen

- **Metadaten:** Beschreibung der Services und seiner Eigenschaften (WSDL, UDDI, ..)
- **QoS Schicht:** Sicherheit, Performance, Verfügbarkeit, Nutzung eines Services abhängig von Gewährleistung von QoS

Wie ist eine SOAP Nachricht aufgebaut und was sind die Funktionen der einzelnen Bestandteile ?

XML Dokument mit folgenden Bestandteilen

- **Envelope:** Container außen drum, enthält quasi nur die Version.
- **Header:** Optional, enthält Informationen zum Routing, Authentifizierung und Autorisierung.
- **Body:** Eigentlichen Inhalt der Nachricht

Wie ist ein SOAP Fehler aufgebaut ?

Ein SOAP Fault Block wird im Body einer Nachricht angegeben. Er enthält folgende Elemente

- Code (z.B. VersionMismatch, MustUnderstand, DataEncodingUnknown)
- Reason

- Node (optional)
- Role (optional)
- Detail (optional)

Was sind RPC und wie funktionieren sie ?

Remote Procedure Calls können mithilfe eines SOAP requests an einem Web Service ausgeführt werden.

Grundlage hierfür ist die Beschreibung des Funktionsaufrufs z.B. in WSDL. Benutzung erfolgt nach Request-Response Verfahren.

07 WSDL & Verzeichnisse

Welche Bestandteile hat eine WSDL Datei und welche Bedeutung haben sie ?

Abstrakte Elemente

- `description`: Äußeres Element zur Kapselung der verschiedenen Abschnitte
- `types`: Zentrale Definition aller Datentypen (gemäß XML Spezifikation)
- `interface`: Abstrakte Beschreibung der Schnittstellen (vergleichbar mit Java Interface)
- `operation`: XML Nachrichten (input/output) die beim Funktionsaufruf zwischen Nutzer und Service ausgetauscht werden

Konkrete Elemente

- `binding`: Protokolle (üblicherweise SOAP) und Transport (RPC, document-style, ...)
- `service`: Verbindung zwischen Abstrakter Schnittstelle (binding) und konkreten Schnittstelle (Service Adresse/URL)
- `documentation`: Optionale, freie textuelle Beschreibung des Services

Was kann mit `include` und `import` bei WSDL Dokumenten erreicht werden ?

Mit `include` können Komponenten des gleichen Namensraums eingebunden werden. Mit `import` können sogar Komponenten aus anderen Namensräumen eingebunden werden.

Der Namensraum wird in der `description` definiert und muss einzigartig sein.

Wie ist eine WSDL Datei als Baum aufgebaut (Zusammenfassung aller Elemente und ihre Hierarchie)?

Was sind MEP und welche gibt es ?

MEP sind Message Exchange Patterns. Sie können

Für die Propagierung von Fehlern gibt es 3 verschiedene Regeln

- No Faults: Fehler werden ignoriert
- Message Triggers Fault: Eine Fehlermeldung wird gesendet wenn ein Fehler auftritt
- Fault Replaces Message: Eine Nachricht kann durch einen Fehler ersetzt werden

Der Nachrichtenaustausch kann folgenden Mustern folgen:

- Unidirektional: In-Only/Out-Only (Regel: No Faults)
- Robust Unidirektional: Wie Unidirektional (Regel: Message Triggers Fault)
- Bidirektional: In-Out/Out-In (Regel: Fault Replaces Message)
- optional Bidirektional: Out-opt-in (Regel: Message Triggers Fault)

Letzere ist nicht in der [Spezifikation](#) aufgeführt aber in den Folien (VL 07 Folie 26) enthalten.

Was ist WS-Inspection und wie funktioniert es ?

WS-Inspection ist ein Verzeichnisdienst für WebServices. Jeder Provider hat ein eigenes Verzeichnis welches nur die Services von einem Anbieter beschreibt in Form einer Web Service Inspection Language Datei (WSIL).

Das Verzeichnis kann bei jedem Anbieter an einer Standardisierten Stelle gefunden werden (z.B. www.example.com/inspection.wsil).

Die Verwaltung (publis/change/delete) der Services in einem WSIL Verzeichnis kann über einen Web Service stattfinden.

Was ist UDDI und wofür ist es gut ?

UDDI steht für Universal Description, Discovery and Integration und ist ein Verzeichnisdienst für WebServices. Die im UDDI Verzeichnis hinterlegten Informationen erlauben das erstellen einer WSDL Datei. UDDI bietet eine API zum verwalten und finden von Services/Providern.

Wird quasi nirgends genutzt außer manchmal Firmenintern.

Welche Sichten/Tabellen gibt es bei UDDI ?

- **White Pages:** (Business)Informationen über alle Anbieter
- **Yellow Pages:** Dienste aller Anbieter (sortiert nach Kategorie)
- **Green Pages:** Beschreibung einzelner Dienste für Menschen (als Website)
- **Service Type Registration:** Maschinenlesbare Green Pages

Was ist der Unterschied zwischen dem abstrakten und konkretem Teil einer WSDL ?

Abstrakt: Rein Funktionale Beschreibung des Services, beschreibt Methoden, Nachrichtentypen, Datentypen, Parameter etc. Alles notwendige um den Service zu nutzen.

Konkret: Konkrete Informationen über einen bestimmten Service, gibt Endpoint (URL/IP) und alles nötige an um den Bind durchzuführen.

Vorteil: Bei gleichem abstraktem Teil der Beschreibung kann der konkrete gewechselt werden ohne das Anpassungen am Consumer nötig sind (Bei Bind während der Laufzeit) z.B. für Wechsel auf eine andere IP oder Load-Balancing.

08 Web Service Security

Was sind allgemeine Sicherheitsmaßnahmen ?

- Stellenwert der Sicherheit in der Organisation erhöhen
- Einführung von Sicherheitsvorgaben
- Rechte Management
- Risikoanalyse Daten und situationsgerechte Sicherheitskonzepte
- Aufklärung, Schulung
- Absicherung gegen Elementarschäden, Diebstahl und Vandalismus

Beschreiben sie kurz die Eigenschaften von Symmetrischer bzw. Asymmetrischer Kryptografie und Digitalen Signaturen ? Welche Eigenschaften gewährleisten sie jeweils ?

- Symmetrische Verschlüsselung: Ein Schlüssel ver/entschlüsselt Daten. Schnell, simple und wenig Rechenleistung dafür problem des Schlüsselaustauschs
- Asymmetrische Verschlüsselung: Public/Private Key, Komplex und Rechen aufwändig dafür kein geheimer Schlüsselaustausch
- Digitale Signaturen: Nachrichten Hash erzeugt mit privatem Schlüssel wird der Nachricht beigefügt

Die Verfahren gewährleisten

- Vertraulichkeit: Niemand kann Mitlesen (durch Verschlüsselung)
- Integrität: Es wurde nichts verändert (durch Signatur)
- Authentizität: Die Nachricht kommt vom angegebene Absender (mit Identitäts-Zertifikaten durch Signatur)

Welche spezifischen Sicherheitsrisiken gibt es bei Web Services ?

- Replay Attacken (durch Zustandslosigkeit begünstigt)
- XML spezifische Attacken
- WSDL und Service Scanning
- Kompromittierung eines Services
- unberechtigte Service Nutzung
- Ausnutzung von Organisationsschwächen (durch komplexe Struktur begünstigt)

Was ist die kanonische Repräsentation einer XML Datei und wofür ist sie gut ?

Die Kanonische Repräsentation einer XML Datei ist die Überführung der Datei in eine exakt spezifizierte Formatierung ohne Änderung der Semantik/Bedeutung der Inhalte. Da bei Hash Verfahren auch kleine Unterschiede (wie Leerzeichen und Zeilenumbrüche) den kompletten Hash verändern und hierdurch Unterschiede entstehen können die Signaturen ungültig machen würden auch wenn sie eigentlich gültig sein sollten.

Was ist/kann WS-Security und wie funktioniert es ?

WS Security ist eine Erweiterung von SOAP um Sicherheitsbezogene Merkmale im SOAP Header zu integrieren.

Es kann

- Signieren: Umwandlung des XML Body in Kanonische Repräsentation, anschließend Signatur
- Verschlüsselung: Verschlüsselung beliebiger Teilbäume der XML, Übermittlung der genutzten Verfahren im SOAP Header
- Security Tokens: Erstellung zeitlich begrenzt gültiger Login Token zur Authentifizierung

- Basis für weitere Sicherheits Erweiterungen

Was sind Transaktionen und welche Bedeutung haben die ACID Eigenschaften ?

Transaktionen sind folgen von Operationen deren Effekt entweder ganz oder gar nicht zum tragen kommt. Dieses verhalten wird durch die ACID Eigenschaften beschrieben

- **Atomarität:** Ganz oder garnicht
- **Konsistenz:** Von einem konkreten zustand in einen anderen konkreten zustand
- **Isoliertheit:** Transaktion unbeeinflusst von parallelen Transaktionen
- **Dauerhaftigkeit:** Das Ergebnis wird gespeichert oder der Ursprungszustand wiederhergestellt bei Fehlern

Was ist WS-TX und welche Typen gibt es ?

Das **WebService Transaction Framework** erlaubt WebServices Transaktions-basiert zu arbeiten.

WS-Coordination erzeugt einen Transaktionscontext. Teilnehmer können der Transaktion hinzugefügt werden.

Bei WS-AtomicTransaction sammelt und propagiert der Koordinator das Ergebnis der Transaktion zentral (synchron). Vorteil: Sicherer und besser für kleine Transaktionen mit wenigen Teilnehmern. Einfacher Rollback Nachteil: Potenziell lange Locks

WS-BusinessActivity wie Atomic nur ohne Locks mit Attributen zur Rücknahme von Änderungen (asynchron). Vorteil: Besser für große Transaktionen mit vielen Teilnehmern die möglicherweise lang laufen. Nachteil: Rollback kann sehr anspruchsvoll werden

Benutzt eigentlich kaum einer weil hart kompliziert und schlechte Tools. ESB als Alternative.

Welche Sicherheitsanforderungen erfüllen Verschlüsselung und digitale Signaturen einzeln und Zusammen ?

11 Sonstige SoA Implementierungen

Was ist DNS ?

DNS steht für Domain Name System und ist ein System zum auffinden von Ressourcen.

Ist gibt verschiedene Arten von Records (A, MX, PTR, NS, TXT) die verschiedene Funktionen Bereitstellen (Name->Adresse, Adresse->Name, Mail, Text, andere Name Server).

Provider veröffentlichen Dienst in einem Verzeichnis (Name Server), Consumer finden Dienste und nutzen sie dann.

Was ist DNS-SD ?

DNS SD ist ein DNS Basiertes Service Discovery Protokoll. Wichtigste Implementierung ist **bonjour** von Apple.

DNS-SD Nutzt drei Records

- PTR für Übersetzung von Service Typ zu Service Name
- SRV für Services und ihre Priorität, Gewichtung, Port und ihren jeweiligen Provider (quasi Beschreibung)
- TXT für allgemeine Attribute

Service suche passiert wie DNS Anfrage. Häufig Multicast anstelle von konkreten Name Server wo jeder Service Antwortet als wäre er ein Name Server.

Wie sieht das DNS-SD SOA Dreieck aus ?

Wie sieht der Prozess beim finden eines Services mithilfe von DNS-SD aus ?

- Suche nach Service Typ
- Antwort mit Liste aller Services jenes Typs
- Abfrage genauer Informationen zu Service
- Antwort mit SRV Record und ggf. TXT Record

Was ist OSGi und wie funktioniert es ?

OSGi steht für Open Services Gateway initiative. Es ist die Spezifikation und Referenzimplementierung einer Java basierten Softwareplattform mit dem Ziel Anwendungen zu Modularisieren (in sogenannten Bundles). Diese Bundles, in Form von Java Objekten, werden über einen Broker zwischen Bundles ausgetauscht und genutzt.

Wie sieht das SOA Dreieck für OSGi aus ?

Wie sieht der Lifecycle von OSGi Bundles aus ?

1. Installation: Bundle wird im lokalen OSGi Framework persistiert
2. Resolving: Framework löst alle Abhängigkeiten auf
3. Aktivität: Angebot und Nutzung von Services, `start()` und `stop()` Methoden
4. Deinstallation: Bundle wird entfernt außer Abhängigkeiten andere Bundles existieren

Was sind RESTful Web Services und welche Eigenschaften haben sie ?

Representational State Transfer Architektur Beschreibt den Einsatz von Web Standards zum Transfer von Web Ressourcen. Ressourcen besitzen Adressen und werden vom Client abgerufen.

Verschiedene Methoden für Verschiedene Funktionen (GET, POST, PUT, PATCH, DELETE, ...).

Erfüllen alle Eigenschaften eines Services sind jedoch meist nicht mit Verzeichnis, daher kein SOA Dreieck. Sind im wesentlichen Ressourcen und nicht Funktionen.

REST vs. WS ist wie Apfel vs. Birnen. Alles seine Berechtigung.

Was sind Microservices ?

Kleine Teams die Vollständig für ihren Service verantwortlich sind. Größere Applikationen werden als Sammlung von unabhängigen Services erstellt.

Die Größe der Services ist irrelevant, wichtig ist vor allem ihre Abgeschlossenheit und Unabhängigkeit. Kommunikation erfolgt häufig über REST Services und Microservices haben oft eigene Datenbanken.

Wie kann Interoperabilität in einer heterogenen SOA Landschaft hergestellt werden ?

Brücken

Zwischen A und B. Bedeutet ggf. Umwege wenn über mehrere Brücken übersetzt werden muss bzw. Anzahl der Brücken Steigt enorm wenn alle Services verbunden werden.

Abstrakte Provider

Services werden in abstrakter Sprache angeboten und durch Übersetzer in die jeweiligen SOA Protokolle übersetzt. Nachteil ist das bestehende SOA Fähige Geräte ausgeschlossen sind.

Abstrakte Consumer

Spezielle Consumer die Services in Abstrakter Form konsumieren. Wie bei den Abstrakten Providern können hier die bereits SOA Fähigen Consumer nicht weiter genutzt werden.

Zentrale Vereinheitlichung

Eine Zentrale Stelle Übersetzt zwischen Providern und Consumern. Hierdurch ist optimale Skalierbarkeit gegeben, es kann jedoch sein das beim Umwandeln semantische Verluste auftreten und die Zentrale Übersetzungsinstanz ist sehr komplex.

Was sind die wesentlichen Unterschiede zwischen WebServices und REST ?

Web Service	REST
Aufruf von Funktionen in verteilten Systemen	Zugriff und Manipulation von Ressourcen
umfangreiche Erweiterungsmöglichkeiten	keine nennenswerten Erweiterungen
Bindung meist zur Laufzeit	Bindung meist zur Entwicklungszeit
Schwergewichtig	Leichtgewichtig
Sinnvoll bei besonderen Sicherheitsanforderungen	Sinnvoll wenn geringe Security Anforderungen und
und wenn der Aufruf von Funktionen besonder wichtig ist	Fokus auf Ressourcen

12 Prozessmodellierung

Was ist ein Geschäftsprozess ?

Ein Geschäftsprozess ist eine Folge von Aktivitäten um ein Geschäftliches Ziel zu erreichen. Ein Geschäftsprozess hat Kunden, besteht aus einer Menge Zielgerichteter Aktivitäten und wird durch Personen oder Systeme ausgeführt. Er berührt in der Regel mehrere Unternehmensbereiche.

Wieso und wofür ist Modellierung im SOA Kontext relevant ?

Welche Services werden Benötigt und wie lassen sie sich aus den Geschäftsprozessen ableiten.

Welche grundsätzlichen Arten der Notation gibt es für Modelle und was sind ihre Vor-/Nachteile ?

- Text: Einfach zu erstellen aber nicht standardisiert oder automatisch prüfbar. Schnell unübersichtlich.

- Tabellen: Keine vernünftige Darstellung von Kontrollflüssen und zusammenhängen. Kaum Standardisiert und schnell Unübersichtlich.
- Tabellen mit Diagrammen: Können Kontrollflüsse Darstellen.
- Graphische Modellierung: Sehr mächtige Standardisierte Notationen. Teilweise schwierig zu erlernen aber dann vorteilhaft.

Beschreibe und benenne die Grundlegenden Elemente von BPMN.

Beschreibe den Grundlegenden aufbau eines UML Aktivitätsdiagramm!

Was sind die Elemente einer EPK ?

Was sind die Elemente von Programmablaufplänen ?

Was ist der Unterschied zwischen Orchestrierung und Choreographie im Bezug auf Service Komposition ?

Orchestrierung:

Über ein **centralized workflow management**, das den orchestrator darstellt wird der Ablauf einer Aktivität gesteuert. Der orchestrator entscheidet unter welchen Bedingungen welcher Service verwendet wird, nicht der Service selbst. -> Zentralisierter Ansatz für Service Komposition

(Hier centralized workflow management als "Composite Service")

Choreographie:

Hier wird dokumentiert wie Services direkt miteinander kommunizieren. Das Format des Datenaustausches und der Interaktion zwischen Services liegt im Fokus. -> Dezentralisierter Ansatz für Service Komposition

13 SoA für Pervasive Computing

Was ist Pervasive Computing und wie ist die Entstehungsgeschichte ?

Pervasive Computing beschreibt die unsichtbare Unterstützung des Nutzers durch allgegenwärtige Informationsverarbeitung (z.B. Smart Home).

Entwicklung von Mainframe Ära (ein Computer, viele Anwender) zur PC Ära (ein Anwender ein Computer) hin zur Ubicomp Ära (ein Anwender, viele Computer).

Welche Herausforderungen von Pervasive Computing können mithilfe von SOA gelöst werden ?

- Abstraktion von Hard- und Softwareunterschieden
- einfache spontane Einbettung und Anpassung von Komponenten
- Interaktivität und Skalierbarkeit

Wie kann SOA für die Geräteinteraktion bei Pervasive Computing verwendet werden ?

Zu Geräten bei Pervasive Computing gehören u.a.

- Computer, Smartphones, Kameras, Lautsprecher, Mikrofone
- Steuerung von Heizung, Licht, Klimaanlage und Sensoren für Helligkeit, Temperatur, ...
- RFID und andere Tags

SOA Verfahren wie Web Services, Jini/UPnP, DNS-SD, usw. können genutzt werden um die Kommunikation zwischen den Geräten zu ermöglichen.

Was sind formen von Kontext und wie kann SOA Helfen Kontext zu nutzen ?

- Roher Kontext: Rohdaten direkt von Sensoren oder Messgeräten
- Normalisierter Kontext: Roher Kontext in einheitlichen Datenstrukturen, erlaubt vergleich verschiedener Kontextdaten
- Komplexer Kontext: Bedeutung des Normalen Kontexts im Rahmen eines Szenarios

SOA's können Kontext

- Anbieten: Als Provider für z.B. normalisierte oder komplexe Kontext Messdaten
- Transformieren: Als Übersetzer für Informationen (z.B. Einheiten)
- Ableiten: Aus mehreren Kontextdaten neue Informationen Ableiten

Welche Dimensionen/Arten von Kontext gibt es und welche Bedeutung haben sie ?

Die wichtigsten Kontext Dimensionen/Arten sind

- location: Wo ist ein Nutzer ?
- identity: Wer Interagiert/macht gerade etwas ?
- activity: Was passiert gerade ?

- time: Wann wird etwas gemacht ?

Diese vier Kontext Kategorien können von einem Anwendungsentwickler genutzt werden um zu verstehen **Warum** etwas gerade passiert und entsprechend das Verhalten der Anwendung anpassen.

Wie könnte man eine Kontextbewusste Applikation beschreiben/definieren ?

Die Definition für Kontext Bewusst lautet wie folgt

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.

Anwendungen kennen verschiedene Abstufungen sind aber Prinzipiell Kontextbewusst wenn sie diesem Schema folgen.

14 Cloud Computing

Was sind Eigenschaften von Cloud Computing ?

Eigenschaften

- Dienstbringung auf Anforderung ohne Menschliche Interaktion
- Netzbasierter Zugang über standardisierte, maschinenlesbare Schnittstellen
- Ressourcen in Pools für mehrere Nutzer parallel verfügbar
- Ressourcenverfügbarkeit on demand und gefühlt unendlich
- Messbare Dienstqualität und Nutzungsabhängige Abrechnung

Welche Virtualisierungsebenen gibt es ?

- Betriebssystem: Ein Host System mit verschiedenen Anwendungen (vRoot Server)
- Plattform: Eine Plattform mit Gast OS und Anwendungen (VM's)
- Speicher: Storage Area Networks die eine Schnittstelle zu selbst verwaltetem Speicher bieten
- Netzwerk: vLAN's bieten Netze die Virtuellt zusammengefasst sind
- Anwendung: Anwendung in der Cloud und Streaming zum Kunden (RDP)

Was ist der Unterschied zwischen Public, Private und Hybrid Cloud

Public (jeder kann Service beziehen) und Private Cloud (Service nur für eine Organisation). Hybrid Cloud nutzt beides je nach Bedarf.

Welche Formen von Cloud Services (?aaS) gibt es ?

- IaaS: Virtuelle Cloud Infrastruktur (EC2, AWS)
- PaaS: Virtuelle Entwicklungsumgebungen mit PaaS spezifischen Bibliotheken (App Engine, Azure)
- SaaS: Konkrete Software für Endkunden in der Cloud (Google Docs)
- HaaS: Einbinden von Menschen in IT Systeme (Mechanical Turk)

Welche Wirtschaftliche Bedeutung haben Cloud Angebote für Nutzer und Anbieter ?

Kostenreduzierung durch Pay per Use und hohe Flexibilität und Elastizität für Anwender.

Cloud Computing ist ein Markt mit sehr hohem Wirtschaftlichen Potenzialen für Anbieter

Welche Risiken gibt es bei der Nutzung von Cloud Angeboten ?

- Zuverlässigkeit eines Anbieters
- Vendor Lock-In bei PaaS und SaaS
- ggf. Vertraulichkeitsprobleme und Rechtliche Unsicherheit
- Kostenfallen

Welche Arten von Hypervisor bzw. Virtuellen Maschinen Monitoren gibt es und wie können diese virtualisieren ?

Arten von Hypervisoren

Typ-1: Setzt auf Hardware direkt auf, sprich ist speziell für diese entwickelt.

Typ-2: Setzt auf einem Basis-Betriebssystem für Virtualisierung auf (Höherer Overhead-Verlust, aber günstiger da nicht speziell)

Virtualisierungsarten

Vollständige Virtualisierung: Keine Anpassungen auf Seiten des Gast-Betriebssystems notwendig, volle Hardware-Emulierung.

Para-Virtualisierung: Anpassungen auf Seiten des Gast-Betriebssystems notwendig, da spezielle Calls statt Hardware

Was sind Vorteile und Nachteile von Cloud Computing ?

Vorteile:

- Effizienzsteigerung durch Hardwarereduzierung und höhere Auslastung
- Effizientes Notfallmanagement durch Verschiebung ganzer Virtueller Maschinen bzw. Container
- Schnelle und einfache Erhöhung von verfügbaren Ressourcen für Virtueller Maschine bzw. Container

Nachteile:

- Zeit und Kostenintensiver Umstieg besonders für Traditionell gewachsene Systeme
- Kontrollverlust als Nutzer, speziell über Location im Sinne von Rechtslage und Ping
- Abstraktionsschicht kostet selbst Ressourcen

09 UPnP

Was ist UPnP und welche Eigenschaften hat es ?

Universal Plug and Play wurde eingeführt um verschiedene Geräte einfach über unterschiedliche Transportmedien (Ethernet, Bluetooth, WLAN, ..) zu verbinden.

Eigenschaften sind

- Klar definierte Protokolle zum Beitritt und Verlassen eines (ad-hoc) Netzwerks
- Zero-Configuration durch Discovery/Announcement Prozesse
- Sicherheit für Einfachheit vernachlässigt
- Geräte verfügen über Services (Funktionen) und Service Beschreibungen
- Control Points können als Event Empfänger oder Steuerung eines Geräts registriert werden

Was ist Multicast ?

Multicast ist das Senden einer Nachricht an eine Gruppe von Empfängern.

In einem IP Netz gibt es Vordefinierte Adressen für bestimmte Zwecke die von Geräten "abboniert" werden können. Ist im Netzwerk ein Gateway vorhanden muss dieser Multicast unterstützen/weiterleiten).

Was ist DHCP und Auto-IP ?

DHCP ist das Dynamic Host Configuration Protocol. Es vergibt IP Adressen an Netzwerkteilnehmer. Entweder Automatisch permanent, manuell oder dynamisch Temporär.

Ist kein DHCP Server im Netzwerk zu finden kann Auto-IP per "trial-and-error" eine freie IP finden. Hierbei wird üblicherweise weiterhin nach einem DHCP Server gesucht und umgestellt sobald einer auftaucht.

Was sind definierte Features für UPnP ?

- Gerätebindung: Annoncement, Discovery, Protokolle zur Interaktion, Beitritt, Austritt
- Ad-Hoc: Spontane Vernetzung ohne Infrastrukturabhängigkeit
- Zero Configuration: Nutzer muss nichts konfigurieren
- Standard basierte Architektur: Alles Standards, keine eigenen API's
- Plattformunabhängig: durch unabhängige Protokolle
- Medien/Geräte unabhängig: durch IP-Basis
- Mangelhafte Sicherheit für einfache Nutzung

Wie sieht das SOA Dreieck für UPnP aus ?

Welche Bedeutung haben URL und URN ?

- URL = Uniform Resource Locators: Identifikation von Ressourcen anhand von Adresse mit Angabe des Protokolls (z.B. `http://www.example.com/ressource.html`)
- URN = Uniform Resource Names: Eindeutiger Name einer Ressource (z.B. `urn:isbn:0-9717861-1-9`)

URN's werden zur Identifikation von Device und Service Typen und Instanzen genutzt während URL's der Adressierung dienen.

Erläutern sie die Phasen von UPnP!

Nacheinander

- Addressing: DHCP oder Auto-IP finden IP für Gerät
- Discovery: Per SSDP werden Ressourcen im Netzwerk gesucht.
- Description: XML Dokumente Beschreiben Geräte und Services in UPnP Template Language

Als letztes Parallel

- Control: Steuerung von Geräten über SOAP
- Eventing: GENA erlaubt Control Points Events der Geräte zu abonnieren.
- Presentation: Präsentation durch Web Interfaces oder Interfaces auf Geräten

Was ist und wie funktioniert SSDP bei der UPnP Discovery ?

SSDP steht für Simple Service Discovery Protocol. Es gibt zwei Verfahren

- Discovery: Der Control Point schickt einen Discovery (multicast) request und erhält Antworten von Geräten
- Announcements: Geräte schicken Announcements das sie da sind (per multicast), Control Points empfangen diese

Was ist und wie funktioniert GENA bei UPnP Eventing?

GenA steht für General Event Notification Architecture.

- Ein Control Point kann per subscribe Nachrichten Abonnieren.
- Per Notify erhält er dann events bis zum unsubscribe
- Control Points können alle Variablen der Service Beschreibung abonnieren
- Geräte/Services bestimmen rate Limits (maximumRate) und minimale Differenzen für Variablen bei Notify (minimumDelta)

10 Jini

Was ist Jini und Apache River

Jini == Apache River. Alles ist ein Service, Fokus liegt auf der einfachen Zusammenarbeit von Anwendungen/Anwendungsfunktionen untereinander. Theoretisch Plattformunabhängig, defacto aber nur in Java benutzt.

Erläutern sie die Basiskonzepte von Jini!

- Service: Die Service Beschreibung ist ein Java interface welches über das Netzwerk übertragen und mittels Stubs scheinbar lokal genutzt werden kann.
- Lookup Service: Multicast basiertes Verzeichnis welches auf Anfrage seine Adresse zurückgibt. Registrierung von Services mit Lease (Gültigkeitszeitraum). Ist selbst ein Service.
- Service Registrierung: Services registrieren sich selbstständig beim Lookup Service
- Service Discovery: Anfragen beim Lookup Service geben Services gefiltert nach bestimmten Kriterien zurück

Wie sieht das SOA Dreieck für Jini aus ?

Welche Sicherheitsaspekte sind bei Jini zu beachten ?

Bei Jini wird fremder Code in der eigenen Umgebung ausgeführt. Ein **Security Manager** verwaltet rechte für bestimmte Domains oder Hosts.

Ab Jini 2.0 gibt es zusätzlich Signaturen, Authentifizierung und Verschlüsselung mithilfe von Plugins.

Was ist eine Jini-Surrogate-Architektur ?

Wird genutzt um nicht Jini fähige Geräte in eine Jini Architektur zu integrieren.

Ein Host verwaltet alle Surrogate Hosts, diese verwalten jeweils ihre Geräte über ein Surrogate (jar) welches die Gerätespezifische Kommunikation übernimmt.

Wie sieht der Jini-Surrogate Lifecycle aus ?

1. **Discovery** über Push durch Gerät oder Pull durch Surrogate Host
2. **Binding** Upload durch- oder Download vom Gerät durch den Surrogate Host
3. **Ausführung** Host bietet Nutzung über Jini Service an
4. **Freigabe** bei Verlust der Geräteverbindung oder bei Abmeldung eines Geräts