SQL vs. NoSQL

Frage 1

Was sind die Vor- und Nachteile von 2 bzw. 3 Tier-Architekturen? Wo können Skalierungsprobleme auftreten und wie lassen sie sich lösen?

Antwort

-

Frage 2

Was sind die Stärken und Schwaächen von RDBMS?

Antwort

-

Frage 3

Welche Aspekte werden durch NoSQL-Datenbanken adressiert?

Antwort

-

Frage 4

Was ist der Unterschied zwischen Scale-Up und Scale-Out?

Antwort

- vertikale Skalierbarkeit Scale UP
- horizontale Skalierbarkeit Scale Out

Frage 5

Erläutern Sie die Unterschiede zwischen Shared Memory, Shared Disk und Shared Nothing Systemen? Was sind die Vor- und Nachteile?

Antwort

logisch

Frage 6

Was bedeutet ßchemafreie Datenmodellierung"?

Antwort

kaum restriktionen , einfache key-value stores oder dokument stores

Frage 7

Was füer Arten von NoSQL-DBS gibt es? Charakterisieren Sie diese kurz und nennen Sie jeweils Beispiele für konkrete Systeme!

Antwort

Die Mischung macht's: Kombination aus verschiedenen Technologien, um Optimum zu erreichen, Sehr Komplexe Infrastruktur!

Frage 8

Was bedeutet Polyglot Persistence? Bewerten Sie den Ansatz!

Antwort

-

Frage 9

Was versteht man unter Big Data? Wofür stehen die 3 bzw. 4 V's? Welche der V's adressieren die meisten NoSQL-Datenbanken?

Antwort

- Volume (Quantität der Daten)
- Variety (Art der strukturierung der Daten)
- Veracity (unglaubwüdige Daten)
- Velocity (daten werden schnell, exponential erzeugt)

JDBC und DAO

JDBC und DAO

Frage 1

Was sind die Gemeinsamkeiten und Unterschiede der Klassen Statement, PreparedStatement und CallableStatement?

Antwort

-

Frage 2

Nennen Sie jeweils einen Anwendungsfall, bei dem Sie die Klasse Statement bzw. PreparedStatement einsetzen würden und begründen Sie Ihre Entscheidung!

Antwort

-

Frage 3

Nennen Sie Gründe, warum PreparedStatements in der Regel einfachen Statements vorzuziehen sind!

Antwort

-

Frage 4

Warum sollten JDBC-Aufrufe immer in try-catch-finally-Blöcke eingeschlossen werden (bzw. try-with- resources)? Was passiert in jedem der Blöcke?

Antwort

-

Frage 5

Nennen Sie drei Maßnahmen/Richtlinien, um beim Datenbankzugriff über ein API wie z.B. JDBC eine hohe Performance zu erreichen!

Antwort

-

Frage 6

Was versteht man unter SQL-Injection? Geben Sie ein konkretes Beispiel für eine SQL-Injection! Was können Sie tun, um Ihre Anwendung sicher zu machen?

Antwort

-

Frage 7

Was sind Business- oder Domänen-Klassen (bzw. Domain Objects) und warum ist die direkte Einbettung von SQL in Domänen-Klassen problematisch, d.h. was für Probleme können potenziell auftreten?

Antwort

-

Frage 8

Was versteht man unter Data Access Objects? Erläutern Sie, welche Klassen bzw. Interfaces beim Einsatz von DAOs erstellt werden müssen und wie sie zusammen spielen! Welche Aufgaben übernehmen sie jeweils? Welche Vorteile bietet der Einsatz von DAOs?

Antwort

-

Frage 9

Wann sollte das Abstract Factory Pattern für DAOs eingesetzt werden und warum?

Antwort

-

Frage 10

Was ist eine DAOFactory und wofür wird sie eingesetzt?

Antwort

-

Architekturmuster

Architekturmuster

Frage 1

Welche Ziele adressiert die Software-Architektur?

Antwort

-

Frage 2

Was versteht man unter Flexibilität? Mit welchen Methoden kann man sie erreichen? Ist "maximale"Flexibilität ohne Nachteile zu haben?

Antwort

Funktionalität (hinzufügen und modifizieren von Funktionen), kopplung, kohäsion, verantwortlichkeiten trennen

- Nein kostet Performance

Frage 3

Erläutern Sie das Architekturmuster SSchichten". Wie sollten die Schichten konzipiert werden? Was für Vor- und Nachteile bringt eine mehrschichtige Architektur

Antwort

- Hierarchische Strukturierung
- Vermeidung von Layer-Bridges
- Information Hiding / Kapselung

Frage 4

Was versteht man unter Loser Kopplung und starker Kohäsion?

Antwort

weiß ich

Frage 5

Was ist das DRY-Prinzip und warum wird dadurch die Abhängigkeit zwischen Komponenten verstärkt?

Antwort

- Don’t Repeat Yourself
- vermeidung von Redundanz, Code-Duplizierung
- wiederverwendung von code / generic / kopplung

Frage 6

Nennen und beschreiben Sie die SOLID-Prinzipien.

Antwort

- SRP (nur eine verantwortlichkeit)
- Open Close
- Liskov
- mehrere interfaces anstatt ein einzelnes
- abhänigkeiten von interfaces nicht von konkreten implementierungen

Frage 7

Erläutern Sie, wie eine Dependency Inversion implementiert werden kann! Sollten alle Abhängigkeiten höherer von niederen Modulen über Dependency Inversion aufgelöst werden?

Antwort

-

Frage 8

Was ist eine Dependency Injection? Was ist der Unterschied zu Dependency Inversion? Welche Einsatzbeispiele haben Sie kennengelernt?

Antwort

-

Frage 9

Erläutern Sie die Architekturmuster Factory, Singleton und Adapter. Wofür werden sie eingesetzt? Welche Einsatzbeispiele haben Sie kennengelernt?

Antwort

-

Verteilte Datenverarbeitung

Verteilte Datenverarbeitung

Frage 1

Was ist das 2 PC –Protokoll? Wie funktioniert es und wofür wird es benötigt?

Antwort

- nur wenn mehrere die schreibvorgänge erfolgreich abgeschlossen haben, war es erfolgreich

Frage 2

Welchen Einfluss hat das 2PC Protokoll auf die Verfügbarkeit von verteilten DBMS

Antwort

-

Frage 3

Was versteht man unter Sharding und Auto-Sharding?

Antwort

-

Frage 4

Erläutern Sie das Konzept der Replikation. Welche Arten gibt es? Welche Probleme kann man damit lösen und welche bekommt man dafür

Antwort

-

Frage 5

Wie kann man Konsistenz bei replizierten Daten sicherstellen? Und bei partitionierten Daten?

Antwort

-

Frage 6

Was besagt das CAP-Theorem? Geben Sie eine anschauliche Begründung!

Antwort

- Konsistenz
- verfügbarkeit
- ausfallsicher

Frage 7

Was sind jeweils die Vorteile von CP und AP-Systemen und was die Einschränkungen?

Antwort

- cp = konsisteznz
- ap = verfügbar und ausfallsicher

Frage 8

Stellen Sie die Konsistenzmodelle ACID und BASE gegenüber!

Antwort

- ACID schwerpunkt auf konsistenz (2 Phasen commit)
- BASE schwerpunkt verfügbarkeit, späte konflikt erkennung

Java Web

Java Web

Frage 1

Welche Arten von HTTP-Requests gibt es?

Antwort

- GET, POST, UPDATE, DELETE, PUT, HEAD, OPTION

Frage 2

Wie ist der prinzipielle Ablauf eines Servlet-Requests?

Antwort

Request an den Webcontainer, Übersetzung in Methodenaufruf, http Response an Client

Frage 3

Erläutern Sie das MVC-Muster!

Antwort

- Model : Daten speichern, informiert view
- View : Präsentaton, holt änderung aus model
- Control : Anwendungslogik, erfragt und ändert modeldaten

Cassandra

Cassandra

Frage 1

Nennen Sie die Hauptbestandteile des Datenmodells von Cassandra. Wie kann man auf einen Wert einer Zelle zugreifen?

Antwort

Hauptbestandsteile sind: Keyspaces (vergleichbar mit der Datenbank in RDBMS), Columns Families (Tables in CQL), Rows und Columns, Zugriff auf einen Wert der Zelle erfolgt über row_id, Columns Family Name und Column Name.

Frage 2

Wozu wird ein Clustering Key verwendet?

Antwort

Clustering Key wird für die Vorgruppierung und Sortierung der Daten verwendet, um diese effektiv mit einer Abfrage zu ermitteln

Frage 3

Wie kann die Konsistenz im Cassandra Cluster gesteuert werden und welche grundsätzlichen Level gibt es?

Antwort

Konsistenz in einem Cassandra Cluster wird durch das „Consistency Level“ gesteuert. Dies gibt an wie viele Knoten einen Lese- bzw. Schreibzugriff bestätigen müssen, damit dieser als erfolgreich gilt; Die gängigsten Level sind: ANY, ONE, TWO, THREE, QUORUM, ALL

Frage 4

Was versteht man unter Consistent-Hashing und was wird dadurch ermöglicht?

Antwort

Unter Consistent-Hashing versteht man die Organisation der Daten in einem Cluster durch einen TokenRing. Jeder Knoten ist für einen bestimmten Zahlenraum zuständig. Zur Zuordnung der Daten werden die PrimaryKeys durch einen Hashfunktion in einen eindeutigen Token umgewandelt. Anhand dieses INT-Wertes erfolgt die Zuordnung der Daten zu den jeweiligen Knoten im Cluster. Consistent-Hashing ermöglicht die Verteilung von Daten über mehrere Knoten hinweg und minimiert die Re-Organisation von Daten bei hinzukommenden oder wegfallenden Knoten.

Frage 5

Welche Unterschiede gibt es zwischen CQL und SQL?

Antwort

Bei CQL gibt es keine JOINS und keine FOREIGN KEYS. Wildcards gibt es ebenfalls nicht

Frage 6

Welche verschiedenen Collections gibt es in CQL und wie unterscheiden sie sich?

Antwort

- List [item1, item2], Angehängte Liste von Werten, unsortiert
- Set,item1, item2, Alphabetisch Sortierte Liste
- Map, key1:value1, key2:value2, Key-Value-Paare, nach Key alphabetisch sortiert

Frage 7

Wie kann man erreichen, dass ein Datensatz nach einer bestimmten Zeit automatisch gelöscht wird?

Antwort

INSERT INTO tabelle(id, name) VALUES (1, ‚Bla’) USING TTL 120; **JPA**

JPA

Frage 1

Was ist ein OR-Mapper und was versteht man unter „object relational impedance mismatch“?

Antwort

OR-Mapper: Objekt-Relatonaler-Mapper (Abbildung von Datenbankschema auf Programmierschema), Object relational impedance: Die Unverträglichkeit zwischen der objektorientierten Programmiersprache und einer relationalen Datenbank bei z. B. Vererbungen

Frage 2

Was sind die Vorteile von JPA gegenüber DAO?

Antwort

- SQL-Abfragen müssen nicht vom Programmierer geschrieben werden (Automatische Generierung von SQL-Abfragen)
- Unabhängigkeit zur Datenbank
- Standardisierte Schnittstelle zur Datenbank
- Objekte werden durch Funktionsaufrufe persistiert

Frage 3

Welche Komponenten von JPA gibt es und wie arbeiten sie zusammen?

Antwort

- Enitity-Manager: zentrale Programmierschnittstelle zur Datenbanksteuerung
- Persistenz-Kontext: Zwischenspeicher des Entity-Managers
- Life Cycle von Objekten:

Frage 4

Was sind die Stärken und Schwächen von OR-Mappern

Antwort

Frage 5

Welche Grundvoraussetzungen müssen erfüllt sein, sodass ein Java-Objekt als JPA Entität erkannt wird?

Antwort

- muss class oder abstract class sein
- default konstruktor
- Interface implementieren java.io.Serializable
- Keine Kennzeichnung der JPA-Entities mit final
- Vorhandensein von Primärschlüssel für die gemappte Datenbanktabelle
- @Entity @Id

Stärken	Schwächen
Keine Low-Level-Programmierung	Datenbank know how geht verloren (Praxis)
Sprachkonsistenz	JPQL ist stark an SQL angelehnt
Typsicherheit von Java	Abstraktion ist nicht durchsichtig (Beispiel: Fetching)
Unabhängigkeit zur Datenbank	Performance ist fragwürdig
Zugriffsvereinfachung auf die Datenbank	Erzeugung sinnvoller Querys?
Stabilität (geringe Wartung)	Fremdzugriffe auf die Datenbank können nicht verarbeitet werden (Caching)
Austauschmöglichkeit der Datenbank	Fähigkeiten und Stärken einer relationalen Datenbank wird nicht ausgenutzt.

Abbildung 0.1

CRUD	SQL	Entity-Manager
Create	INSERT	persist()
Read	SELECT	find()
Update	UPDATE	automatisch
Delete	DELETE	remove()

Abbildung 0.2

Frage 6

Was ist der Unterschied zwischen einer uni- und einer bidirektionalen Assoziation?

Antwort

Unidirektional: Wenn Objekte einer Seite (Klasse A) auf zugeordnete Objekte der anderen Seite (Klasse B) zugreifen kann. Datenübertragung in eine Richtung Bidirektional: Wenn auch der umgekehrte Fall gilt; also, dass von Objekten der Klasse B auch auf Objekte der Klasse A zugegriffen werden kann Datenübertragung in beide Richtungen

Frage 7

Was sind die grundlegenden Funktionen vom Entity-Manager?

Antwort

Frage 8

Welchen Nachteil von JPQL gleicht die Criteria API aus?

Antwort

JPQL-Anfragen basieren zum großen Teil auf Strings. Dadurch kann die Anfrage nicht auf syntaktische Korrektheit überprüft werden. Etwaige Tippfehler werden erst zur Laufzeit bemerkt und lösen Laufzeitfehler aus. Die Criteria API verwendet ein komplexes Klassenmodell um Strings soweit wie möglich aus der Anfrage-Erstellung herauszuhalten. Dafür hat der Programmierer keine Kontrolle mehr über die Performanz der erzeugten SQL-Abfragen. **MongoDB**

MongoDB

Frage 1

Was sind die wesentlichen Vor- und Nachteile dokumentenorientierter zur relationalen Datenbanken?

Antwort

Vorteile

- Zugriffe und Abfragen direkt auf Dokumente Möglich
- keine Schemarestriktion
- hohe Flexibilität mit großen Datenmengen
- Geschwindigkeitsvorteil gegenüber RDBMS
- bessere horizontale Skalierung

Nachteile

- geringe Verbreitung
- damit auch geringe Unterstützung von Schnittstellen und Tools
- keine Dokumentenübergreifenden Operationen
- darstellen komplexer Beziehungen eher schwierig
- durch Schemafreiheit müssen eventuelle Wertprüfungen nicht von DB übernommen müssen selbst programmiert werden

Frage 2

Für welche Anwendungsbereiche bietet sich die Nutzung von MongoDB an?

Antwort

- Anwendungen ohne festes Datenbankschema
- Anwendungen in denen Daten in nicht tabellarischer Form gespeichert werden
- Webanwendungen, da JSON auch zur Serialisierung verwendet wird
- Anwendungen mit sehr großen Datenmengen
- Anwendungen bei denen viele Nutzer versch. Daten erzeugen und diese schnell gespeichert / verarbeitet werden müssen

Frage 3

Erläutern Sie den grundlegenden Aufbau von MongoDB?

Antwort

MongoDb speichert als Grundeinheit JSON Objekte und besteht aus den folgenden Bestandteilen:

Datenbanken

- MongoDB-Prozess kann mehrere Datenbanken verwalten
- Datenbank besteht aus beliebig vielen Collections
- DatenbankName.CollectionName stellt Namespace dar

Collections

- speichert Dokumente
- Datenbank besteht aus beliebig vielen Collections
- vergleichbar mit Tabelle einer relationalen DB
- Dokumente können völlig verschieden sein (Schemalos)

Capped Collections

- größenbeschränkte Dokumentensammlung
- Größe und Anzahl an Dokumenten kann beim Anlegen festgelegt werden
- Verhält sich wie digitaler Ringspeicher nach Erreichen der Größe
- kann tailable Cursors verwenden

System Collections

- automatisch von MongoDb erzeugt
- Verwalten Indizes, Namespaces, JavaScript und Benutzer

Frage 4

Wie kommuniziert eine Anwendung mit Shards im Sharding Cluster und warum findet die Kommunikation so statt?

Antwort

Eine Anwendung kommuniziert über Router (Schnittstellen) mit den Shards. Dabei werden in sogenannten Config Servers Metadaten und Konfigurationseinstellungen gespeichert. Die Kommunikation soll/muss so stattfinden, da die Anwendung sonst nicht weiß, wie Datenbankoperationen ausgeführt werden. Zum Beispiel: Wie interpretiere ich Datenbankoperationen? Auf welchem Shard befindet sich welcher Datensatz? (s. Folie 41/42 für Veranschaulichung)

Frage 5

Wie funktioniert Replikation in MongoDB? Welches Konsistenzmodell kommt dabei zum Einsatz?

Antwort

MongoDB implementiert eine Master-Slave Replikation. Es gibt nur einen Master (Primary) und nur auf dem Master können Write-Operationen ausgeführt werden. Auf den Slaves (Secondaries) kann nur gelesen werden. Bei einem Write im Master werden die Daten asynchron auf die Slaves aktualisiert. (s. Folie 36 für Veranschaulichung)

Frage 6

Wann wird bei Replica-Sets ein neuer Primary Server gewählt?

Antwort

Ein Primary Server wird dann neu gewählt, wenn...

- der Primary Server ausfällt.
- ein neues Replica-Set (Secondary) zum System hinzufügt wird.
- ein Replica-Set die Verbindung zum Primary verliert.
- ein Primary zurücktritt und zu einem Replica-Set wird.

Eine Wahl beginnt, indem sich ein Replica-Set als neuen Primary vorschlägt. Jedes andere Replica-Set hat dann genau eine Stimme, um dafür oder dagegen zu stimmen. Die Wichtigsten Kriterien damit ein Replica-Set zum Primary gewählt wird sind folgende:

- Antwortzeit von anderen Replica-Sets, nach dem man sie angepingt hat (Replica-Set mit der schnellsten Antwortzeit bekommt die Stimme)
- Priorität (muss manuell festgelegt werden, Replica-Set mit der höchsten Priorität leitet als erstes eine Wahl für sich selbst ein)
- Aktuellster Stand (letzte Datenbankoperation ist ausgeführt worden)
- Aufrechte Verbindung zu mindestens der Hälfte aller Replica-Sets

Frage 7

Wie funktioniert prinzipiell das Map-Reduce-Verfahren?

Antwort

Mit Hilfe von Map-Reduce soll erreicht werden, dass die Gesamtlast auf verschiedene Rechner verteilt wird, um so die Performance zu stärken. Vergleichbar mit einer großen Pizza: Je mehr Leute von der Pizza essen, desto schneller ist diese vertilgt.

Neo4j

Neo4j

Frage 1

Was versteht man unter einem Labeled Property Graphen?

Antwort

Hierbei handelt es sich um einen Graphen, in dem sowohl die Knoten als auch die Beziehungen zwischen den Knoten typisiert (labeled) und den Knoten und Beziehungen Eigenschaften (Properties) zugewiesen werden können.

Frage 2

Wodurch entstehen bei wachsendem Datenbestand Performanceverluste in einem RDBS im Vergleich zur Graph-DB und wodurch wird der Performanceverlust bei Graphdatenbanken verhindert?

Antwort

Durch die viele Anzahl an Joins, durch welche Beziehungen abgebildet werden. In GDB sind Beziehungen first-class-citizens und Anfragen müssen nur auf einen Teilgraphen angewandt werden => Ausführungszeit ist nur von Teilgraph abhängig

Frage 3

Wie unterscheiden sich RDBMS und Graph-Datenbanken bezüglich der Modellierung von Beziehungen?

Antwort

Bei RDMS benötigt man Zwischentabellen (Join-Table), um die Beziehungen zu 1-n-Beziehungen herunter zu stufen und darstellen zu können. Graphen haben dieses Problem nicht, da sie auf keinem Schema beruhen

Frage 4

Welche Problemstellungen können mit Graph-Datenbanken besser abgebildet werden als mit RDBMS und warum?

Antwort

Durch Graphendatenbanken lassen sich stark vernetzte Probleme besser Darstellen als mit RDBMS. Vor allem Probleme bei denen Beziehungen zwischen den Daten eine große Rolle spielen, da diese in Graphdatenbanken "first-class-citizensßind und im gegensatz zu RDBMS nicht umständlich abgebildet werden müssen.

Frage 5

Weshalb ist es bei Graphen sinnvoller mehr Verbindungen zu erstellen, anstatt Knoten weitere Eigenschaften zuzuweisen?

Antwort

Es ist „günstiger“ Verbindungen zu durchlaufen, anstatt Properties auszulesen. Dennoch kann dies auch zur Unübersichtlichkeit führen.

Frage 6

Was sind typische Anwendungsbeispiele für Graphdatenbanken im Vergleich zu anderen NoSQL und relationalen Datenbanken?

Antwort

Soziale Netzwerke, Master Data Management, Empfehlungen (zB in Online Shops), Network and Data Center Management, Authorisierung und Zugriffskontrolle, Fraud Detection **PaaS**

PaaS

Frage 1

Nennen und erläutern Sie wesentliche Punkte, die für oder gegen Cloud Computing in Unternehmen sprechen!

Antwort

Fehlenden Standards, Datenschutz usw.

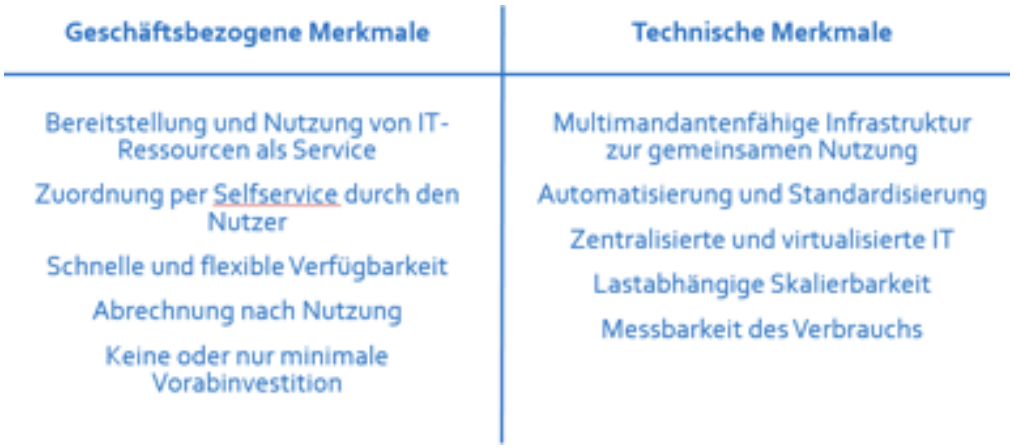

Abbildung 0.3

Frage 2

Was sind die wesentlichen Merkmale von Cloud Computing?

Antwort

Frage 3

Nennen und erläutern Sie die 3 Betriebsarten von Cloud Computing!

Antwort

- Public Cloud
- Private Cloud
- Hybrid Cloud (Mischung aus P und P)

Frage 4

Nennen Sie die 3 Arten von Cloud-Angeboten (3-Ebenen-Modell)!

Antwort

- IaaS
- PaaS
- SaaS

Frage 5

Was sind die wesentlichen Vorteile von PaaS im Vergleich zur klassischen Entwicklung von Anwendungen?

Antwort

- Beschleunigung von Geschäftsprozessen
- Schnelleres und flexibleres Testen und Entwickeln
- Ideales Werkzeug für agile Entwicklung / DevOps

Frage 6

Welche Möglichkeiten zur Skalierung gibt es bei IBM Bluemix?

Antwort

- Manuelle Skalierung
- Automatische Skalierung der Instanzen anhand von festgelegten Regeln

SOLR

SOLR

Frage 1

Was ist der Unterschied zwischen Suche-Systeme und Relationale Datenbanken?

Antwort

Relationale Datenbanken sind für Transaktionssicherheit und Konsistenz konzipiert

Suche Systeme für Indizes und viele Gleichzeitige Abfragen

Frage 2

Was ist ein inverted Index?

Antwort

arbeitet rückwärts, zu den einzelnen Tokens wird festgehalten, in welchem Feld und in welchem Dokument es vorkommt

Frage 3

Aus welchen drei Schritten besteht der Analyse Prozess?

Antwort

- Charfilter (optional)
- Tokenizer (es kann und muss nur einen geben)
- TokenFilter

Frage 4

Wofür steht TF?

Antwort

Term Frequenzy **DB-basierte Web-Services**

Web-Services

Frage 1

Was ist die Motivation für Service-orientierte Architekturen?

Antwort

- Wiederverwendung
- Plattformübergreifende Nutzung
- Integration
- Information Hiding

Frage 2

Was sind die Komponenten einer service-orientierten Architektur? Wie ist der theoretische Ablauf eines Dienstaufrufs?

Antwort

- Dienstverzeichnis (UDDI), Dienstanbieter, Dienstnutzer
- Dienstanbieter veröffentlicht seinen Service im UDDI
- Dienstnutzer durchsucht UDDI nach passenden Service
- Dienstverzeichnis gibt dem Dienstnutzer einen Verweis auf den Dienst (WSDL)
- Interaktion zwischen Dienstanbieter und Dienstnutzer über SOAP-Nachrichten

Frage 3

Welche Herausforderungen gibt es in Bezug auf service-orientierte Architekturen?

Antwort

- Skalierbarkeit
- Sicherheit
- Strukturierung
- Organisationstruktur
- Qualität/Verfügbarkeit
- Konkurrierende Spezifikationen

Frage 4

Welche Vor- und Nachteil hat REST gegenüber SOAP?

Antwort

Vorteile

- Geringere Datenmenge (kein XML)
- Menschenlesbare Ergebnisse
- Höhere Performance

- De-/Serialisieren aufwendiger
- Keine Schnittstellenvertrag (WSDL)

Frage 5

Was sind die 6 Prinzipien von REST?

Antwort

- Einheitliche Schnittstelle (Uniform interface)
- Client-Server
- Zustandslosigkeit (Stateless)
- Cachbar
- Mehrschichtige Systeme (Layered System)
- Code auf Anfrage (optional) (Code on demand(optional))

Frage 6

Was ist der Grundgedanke eines MicroService?

Antwort

Abgeschlossene Fachliche Einheit mit GUI, Logik und Datehaltung. Anwendung setzt sich aus Sammlung von MicroServices zusammen.

Frage 7

Nennen Sie die Vorteile von MicroSerivces gegenüber monolithischen Architekturen.

Antwort

- Kommunikation und Koordination vereinfacht
- Test einer fachlichen Einheit einfacher
- Laufzeitumgebung an Voraussetzungen angepasst
- Langfristige Wartbarkeit
- Vorteile in der Skalierbarkeit

Frage 8

In welchen Punkten unterscheidet sich SOA von MicroServices?

Antwort

- Anwendung als Sammlung von MicroServices - Orchestrierung
- Neue Anwendungsfälle durch neue MicroServices – Neue Anwendungen durch Komposition
- MicroServices haben eigene UI, Businesslogik und Datenhaltung – Mehrere Dienste können an einer UI und Datenhaltung beteiligt sein.
- MicroServices beeinflussen ein Team – SOA beeinflusst mehrere Teams
- MicroServices sind unabhängig – SOAs haben einen single point of failure: Enterprise Service Bus
- MicroServices bilden eine angeschlossene fachliche Einheit – SOA liegt eine Abstraktionsebene über MicroServices