



Probeklausur Integrations- und Migrationstechnologien

2015/2016

1. Motivation & Heterogenität

1a) Beschreiben sie das Konzept einer vertikal organisierten IT. Begründen Sie, warum dieses Konzept nicht geeignet ist, um sich schnell ändernde Unternehmensprozesse abzubilden. Punkte 4

- Große IT-Systeme sind im Laufe der Zeit aus kleineren Systemen entstanden
- Die ursprünglichen Systeme waren für einen dedizierten Zweck entwickelt
- Durch zusätzliche (fachliche) Anforderungen wurden die Systeme weiterentwickelt
- Die Funktionalität orientiert sich dabei an genau einer Fachanforderung
- Jedes System hat seine eigene Oberfläche und Datenbank
- **Vorteile**
 - Ausgereifte Architektur
 - Gute Unterstützung der implementierten Aufgaben
 - Unauffällig gegen Ausfälle
 - Hohe Stabilität aufgrund der vorhandenen Redundanz
 - **Diese Landschaft ist ausreichend wenn:**
 - Die Arbeit sequentiell durch die Systeme läuft
 - Die betrieblichen Prozesse stabil sind

1b) Nennen/Beschreiben Sie die drei wesentlichen Eigenschaften eines föderierten Systems, welche die Integration erschweren/notwendig machen. Geben Sie für jede Eigenschaft ein Beispiel an. Punkte 4

- Autonomie: Entwurfsautonomie
- Heterogenität: **Syntaktische Heterogenität**
- Verteilung: Daten(quellen)

1c) Beschreiben Sie die sog. „datenmodellbasierte Heterogenität“. Geben Sie dafür ein Beispiel an. Punkte 4

Unterschiede in der Datenmodellierung, ausgelöst durch Anwendung unterschiedlicher Modellierungskonzepte oder –Sprachen (z.B. relationales vs. objektorientiertes Datenmodell).

Objektorientiertes Modell vs. Relationales Modell

2. Grundlagen

2a) Beschreiben Sie die Integrationsart „Prozessintegration“ und definieren Sie die wesentlichen Bestandteile und Rollen. Punkte 4

- Verschiedene Systeme nehmen an einem übergreifenden Geschäftsprozess teil
- Geschäftsprozesse müssen permanent an neue Gegebenheiten angepasst werden
- Problem: Wie können lang laufende Prozesse über mehrere Applikationen hinweg gesteuert werden?
- **Process Modell: Definiert den logischen Ablauf einer Geschäftsprozesses**
- Process manager: Interagiert mit den verschiedenen Applikationen und steuert für jede Prozessinstanz die einzelnen Verarbeitungsschritte
- Application: Führt eine spezifische Businessfunktion gesteuert durch den Prozessmanager aus

2b) Beschreiben Sie drei Vorteile, welche sich aus dem Einsatz der Prozessintegration ergeben. Punkte 4

Maintainability. Creating a separate process integration layer allows users to define and maintain the business process independent from the implementation of the individual functions. This increases maintainability and reduces the skill set requirements for the personnel who maintain the process definition.

Reusability. Because the existing applications are not dependent on the process management layer, the functions inside these applications can be reused in multiple process definitions.

Flexibility. The process manager can support a variety of configurations that would be difficult to implement in many traditional programming models. For example, parallel execution of multiple tasks, synchronization points, timeouts, and escalations are all configurations that would be difficult to implement in a traditional programming model. Supporting a variety of configurations gives the process manager the flexibility to adapt to many different business requirements.

2c) Beschreiben Sie zwei mögliche Nachteile, welche bei dem Einsatz der Prozessintegration zu berücksichtigen sind. Punkte 4

Potential bottleneck. Managing a large number of process instances in a central process manager component can present a run-time bottleneck. However, in most cases, the parallel execution of multiple process manager instances can alleviate this threat while maintaining the benefit of central configurability.

Temptation to overuse. This liability is the flip side of the flexibility inherent in Process Integration solutions. Because Process Integration can be used to solve nearly any integration problem, some people take this as an indication that each and every integration problem should be solved using Process Integration. This is not true. Using Process Integration frivolously can lead to overarchitected and sometimes inefficient solutions—for example, in cases where a Message Broker would be perfectly appropriate to address the requirements.

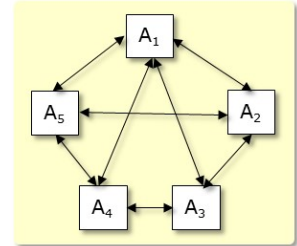
Complexity. A generic process manager can be difficult to build because it has to deal with concurrency, checkpoints, and recoverability. For these reasons, a commercial process manager should be used whenever possible. If a commercial component is not available, the cost of building the process manager should be carefully weighed against the cost of making changes to the process model.

3. Kopplung & Architektur

3a) Beschreiben Sie die Integrationsarchitektur „Point to Point“. Gehen Sie auf Vorteile und Nachteile dieser Integrationsarchitektur ein. Skizzieren Sie diese Integrationsarchitektur. Punkte 4

Point-To-Point:

- Verbund unabhängiger Systeme, die durch ein Netzwerk miteinander verbunden sind.
- Alle Systeme sind gleichberechtigt und können sowohl Dienste konsumieren als auch bereitstellen.
- Keine zentrale Datenhaltung, jedes System hat seinen eigenen Datenbestand
- Komplexität: $n * (n - 1) / 2$



Stärken:

- Geringe Start-/Infrastrukturkosten
- Autonomes System

Schwächen:

- Nur bei wenigen Systemen und wenigen Verbindungen praktikabel
- Einzelne Systeme nur mit hohem Aufwand austauschbar, sehr unflexibel
- Wiederverwendbarkeit von Komponenten ist beschränkt, aufwändiger Betrieb

3a) Beschreiben Sie das Konzept der losen Kopplung und begründen Sie, warum die lose Kopplung als ein wichtiges Designprinzip in einer Integrationsumgebung gilt. Punkte 4

In langfristig beherrschbaren Anwendungslandschaften sollten die Komponenten tendenziell lose gekoppelt sein.

Lose Kopplung hilft um komplexe Anwendungslandschaften beherrschbar zu machen

Vorteile von loser Kopplung

- Erhöht die Unabhängigkeit zwischen den AL-Komponenten
- Erhöht die Stabilität im Betrieb
- Unterstützt die Wartbarkeit und Austauschbarkeit von AL-Komponenten

3b) Beschreiben Sie die Kopplungsart „Kopplung durch Datentypen und Schnittstellen“. Begründen Sie das Problem und beschreiben Sie mögliche Problemfelder. Begründen Sie, wie das Problem in Form „Loser Kopplung“ gelöst werden kann. Punkte 4

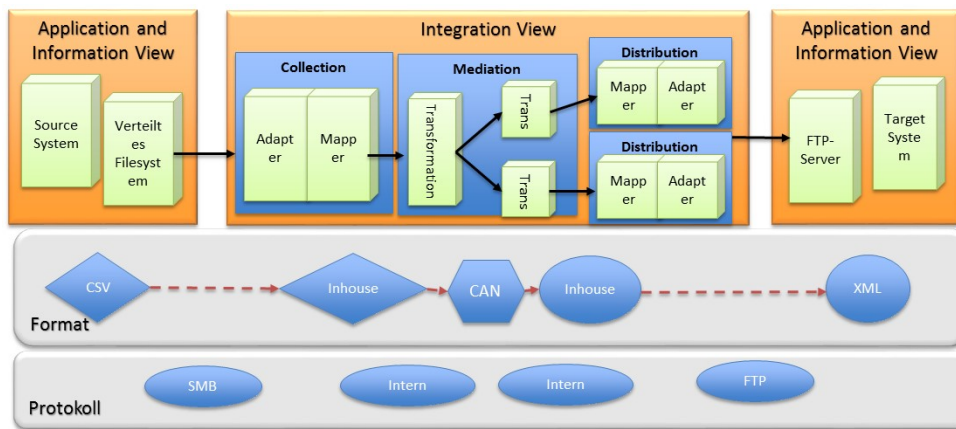
These: Verteilte Systeme können einfacher implementiert werden, wenn überall gemeinsame Datentypen verwendet werden

In dieser Annahme steckt eine Quelle für häufige Fehler:

- Einigung auf gemeinsame Datentypen ist ein extrem schwieriger Prozess (Gremienabstimmungen sind lähmend und langwierig.)
- Unterschiedliche Entwicklungsgeschwindigkeiten, Änderungen an gemeinsamen Datentypen betreffen alle, die sie verwenden.

4. Integration Architecture Blueprint

Analysieren Sie das nachfolgende Beispiel einer möglichen Stammdatenverteilung.



Punkte
8

4a) Erläutern Sie folgende Konzepte anhand des gezeigten Beispiels:

Inhouse-Format:

Das Datenformat in welchem die Integrationslösung die empfangenen Nachrichten intern darstellt. (Häufig XML)

Mapper: Wandelt die externen Formate in das Inhouse-Format um

Adapter: Kommuniziert über Protokolle mit den externen System z.B. Dateisystem

Transformation: Wandelt eine Nachricht von Struktur X in Struktur Y um

Kanonisches Datenformat: Allgemeines Datenformat welches als Ausgangspunkt für die Verteilung auf verschiedene andere Systeme genutzt wird. Integrationsaufwand wird verringert weil Quell- und Zielsysteme unabhängig voneinander werden

4c) Erläutern Sie die Aufgabe der Komponente „Mediation“ innerhalb des Architecture Blueprints. Nennen Sie einen möglichen Standard wie diese Komponente realisiert werden kann.

Punkte
4

- Sichergestellte Weiterleitung von Informationen (reliable forwarding)
- Weiterleitung kann an Komponenten der Process Layer oder an Collection/Distribution
- Komplexere Filterketten können durch weitere Transformation und Routingschritte ergänzt werden
- Standards XSLT, XPATH

5. Enterprise Integration Patterns

- 5a) Erläutern Sie das Grundprinzip eines Message-Channels, Message und Endpoint. Wie können diese Grundelemente für den Bau einer Integrationslösung benutzt werden?

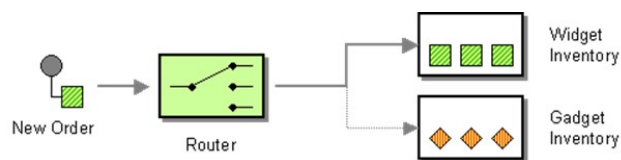
Punkte
7

Connect the applications using a *Message Channel*, where one application writes information to the channel and the other one reads that information from the channel.

When an application has information to communicate, it doesn't just fling the information into the messaging system, it adds the information to a particular *Message Channel*. An application receiving information doesn't just pick it up at random from the messaging system; it retrieves the information from a particular *Message Channel*.

- b) Erläutern Sie das Muster „Content-Based-Router“. Geben Sie ein mögliches Beispiel an, welcher den sinnvollen Einsatz verdeutlicht.

Punkte
5

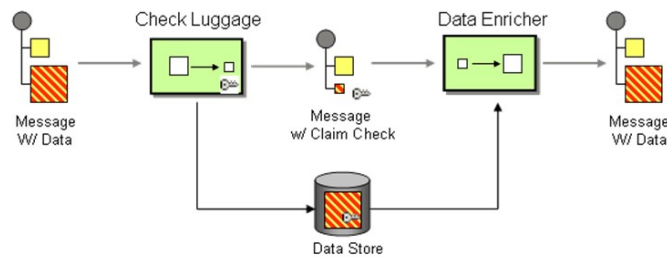


Use a *Content-Based Router* to route each message to the correct recipient based on message content.

The *Content-Based Router* examines the message content and routes the message onto a different channel based on data contained in the message. The routing can be based on a number of criteria such as existence of fields, specific field values etc. When implementing a *Content-Based Router*, special caution should be taken to make the routing function easy to maintain as the router can become a point of frequent maintenance. In more sophisticated integration scenarios, the *Content-Based Router* can take on the form of a configurable rules engine that computes the destination channel based on a set of configurable rules.

Beispiel

- 5c) Erläutern Sie das Muster „Claim Check“. Begründen Sie warum dieses Muster besonders bei großen Nachrichten hilfreich sein kann.

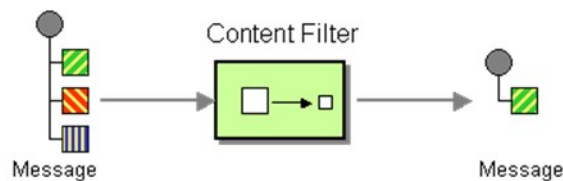


Punkte
5

Claim Check: How can we reduce the data volume of message sent across the system without sacrificing information content?

Store message data in a persistent store and pass a *Claim Check* to subsequent components. These components can use the *Claim Check* to retrieve the stored information.

- 5d) Erläutern Sie das Muster „Content-Filter“. Geben Sie ein mögliches Beispiel an, welcher den sinnvollen Einsatz verdeutlicht.



Punkte
5

Content Filter: How do you simplify dealing with a large message, when you are interested only in a few data items?

Use a *Content Filter* to remove unimportant data items from a message leaving only important items.

The *Content Filter* does not necessarily just remove data elements. A *Content Filter* is also useful to simplify the structure of the message. Often times, messages are represented as tree structures. Many messages originating from external systems or packaged applications contain many levels of nested, repeating groups because they are modeled after generic, normalized database structures. Frequently, known constraints and assumptions make this level of nesting superfluous and a *Content Filter* can be used to 'flatten' the hierarchy into a simple list of elements than can be more easily understood and processed by other systems.

6 Transaktionen

- 6a) Beschreiben Sie das Grundprinzip einer Kompensations-Transaktion. Warum sollte die Transaktionsart bevorzugt eingesetzt werden? Stellt diese Transaktionsart eine lose oder hohe Kopplung dar? Punkte 6

Ausgangspunkt:

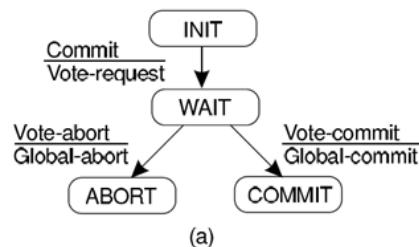
- Es stehen keine verteilten Transaktionen zur Verfügung (z.B. Kopplung von Legacy Systemen)

Grundidee:

- Jeder Funktionsaufruf wird in einer lokalen Transaktion ausgeführt
- Für jede aufgerufene Methode wird eine Compensation-Aktion aufgezeichnet
- Tritt ein Fehler auf, werden alle Compensations-Aktionen in umgekehrter Reihenfolge ausgeführt.
- Dadurch wird das Gesamtsystem wieder in einen konsistenten Systemzustand erreicht
- Um Stabilität über Abstürze hinaus zu gewährleisten sollten die Compensations-Aktionen persistent gespeichert werden

- 6b) Erläutern Sie das nachfolgend dargestellte Zustandsübergangs-diagramm für einen Transaktionskoordinator (2-Phase-Commit). Wie muss sich der Koordinator verhalten, wenn Netzwerkfehler die Kommunikation zwischen Teilnehmern verhindern.

Punkte 6



- Die Transaktion wird gestartet
- Wenn der Koordinator zum Commit aufgefordert wird dann sendet er einen „vote-request“ an alle Teilnehmer
- Stimmt ein Teilnehmer für vote-abort wird die Transaktion abgebrochen und alle anderen Teilnehmer erhalten vote-abort
- Stimmen aller Teilnehmer für commit wird die Gesamttransaktion positiv abgeschlossen

7 XML-Technologien für die Integration

7a) Definieren Sie das Konzept einer idempotenten Funktion. Begründen Sie warum dieses Designprinzip sehr hilfreich im Zusammenhang mit Netzwerkfehlern ist. Punkte 4
Was kann ein Client im Falle eines Netzwerkfehlers tun?

- Als idempotent bezeichnet man Funktionsaufrufe, die immer zu den gleichen Ergebnissen führen, unabhängig davon, wie oft sie mit den gleichen Daten wiederholt werden. Idempotente Arbeitsgänge können zufällig oder absichtlich wiederholt werden, ohne dass sie nachteilige Auswirkungen auf den Computer haben.

7b) Beschreiben Sie die grundsätzliche Idee des Standards „WS-Reliable-Messaging“. Wie löste der Standard das Problem von Netzwerkfehlern und welche Bestandteile existieren? Punkte 4

- Vor der Übertragung der Nutzdaten wird eine Protokoll-Sequence etabliert welche den Verlauf der Kommunikation auf basis von „fortlaufenden Nummern“ verfolgbar macht.

The primary goal of WS-Reliable-Messaging is to create a modular mechanism for reliable transfer of messages. It defines a messaging protocol to identify, track, and manage the reliable transfer of messages between a source and a destination. It also defines a SOAP binding that is required for interoperability. Additional bindings can be defined.

Wiederholung von Nachrichten. Standard verhindert Duplikate

7b) Erläutern Sie den Quality-Of-Service „ExactlyOnce“ innerhalb des Standards WS-Reliable-Messaging. Was muss das Framework tun um dies sicherzustellen? Punkte 4

Each message is to be delivered exactly once; if a message cannot be delivered then an error MUST be raised by the RM Source and/or RM Destination. The requirement on an RM Source is that it SHOULD retry transmission of every message sent by the Application Source until it receives an acknowledgement from the RM Destination. The requirement on the RM Destination is that it SHOULD retry the transfer to the Application Destination of any message that it accepts from the RM Source until that message has been successfully delivered, and that it MUST NOT deliver a duplicate of a message that has already been delivered.

8 XML-Security

8a) Begründen Sie die Notwendigkeit des Standards „XML-Encryption“. Warum ist eine Verschlüsselung auf Ebene der Transport-Layer nicht ausreichend? Punkte 6

- Definiert Standards zur Verschlüsselung von XML-Dokumenten oder Teilen davon
- Ziel: Vertraulichkeit sichern

Grundlegende Vorgehensweise:

- Notwendige Randbedingungen definieren
- Verschlüsselungsverfahren, Initialisierungsparameter
- Verschlüsselungsgranularität (elementweise, ...), Schlüssel
- Generierung des für Dritte nicht mehr lesbaren Dokuments aus dem Eingangsdokument

Notwendig weil:

- Payload unabhängig von Netzwerkprotokoll geschützt wird
- Dadurch ist eine End-2-End-Verschlüsselung möglich

----- END DER PRÜFUNG -----