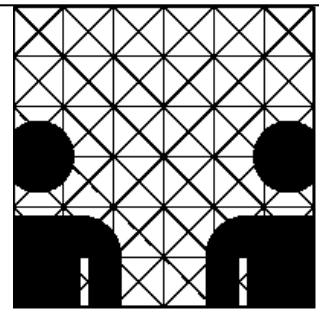


Servicekomposition mit eFlow

Savas Cetin

Inhalt

- Einführung und Motivation
- Überblick über eFlow
  - Composite E-Services
  - Nodes
  - Engine
- Adaptive Service-Prozesse
  - Dynamic Service Discovery
  - Multiservice Nodes
  - Generic Service Nodes
- Modifikationen
  - Ad-hoc
  - Bulk

Einführung und Motivation

- Über das Internet angebotene E-Services müssen mit einer sich ständig verändernden, also stark dynamischen Geschäftsumwelt fertig werden
- Durch ständige Weiterentwicklung der Technologie und dadurch, dass viele Geräte internetfähig werden, wächst die Zahl und Art der Services bzw. Dienste und Anbieter täglich
- Der Wettbewerb wächst und die Anbieter sind gezwungen kundenspezifische Services anzubieten um wettbewerbsfähig zu bleiben
- Diese Entwicklung stellt hohe Anforderungen an ein System, welches die Entwicklung und Zustellung von „***composite e-services“*** *(„zusammengefaßte Dienste“)* unterstützen soll.

Einführung und Motivation (2)

- eFlow ist eine ***Plattform*** um composite e-services zu spezifizieren, in Gang zu setzen und zu überwachen
- Composite e-services sind ***Prozesse*** die andere composite oder Basis-Services zu einem Service zusammenfügen
- Bietet Vielzahl von Funktionen, welche die Service-Prozess-Spezifikation und das Service-Prozess Management unterstützen
- Besitzt mächtige, trotzdem einfache „*service composition language*“
- Event und Exception-Handling
- ACID Service- Level Transaktionen
- Security-Management
- Überwachungsfunktionen

Einführung und Motivation (3)

- Über das Internet angebotene E-Services müssen also mit einer sich ständig verändernden Geschäftsumwelt fertig werden
- Natürlich ist es nicht möglich den Prozess kontinuierlich den Änderungen der Marktbedingungen anzupassen

  =>Änderungen würden zu häufig auftreten

  =>Prozessänderung ist zeitaufwendige Angelegenheit
- Es sollte aber möglich sein Prozessänderungen mit minimalsten Benutzereingriff oder mit keinem Eingriff durchzuführen

Einführung und Motivation (4)

- Es sollte auch möglich sein Prozess-Definitionen oder Änderungen auf einfache und effektive Art durchzuführen, wo Benutzereingriff erforderlich
- Bildung von Konsistenzregeln
- Bildung von Autorisationsregeln

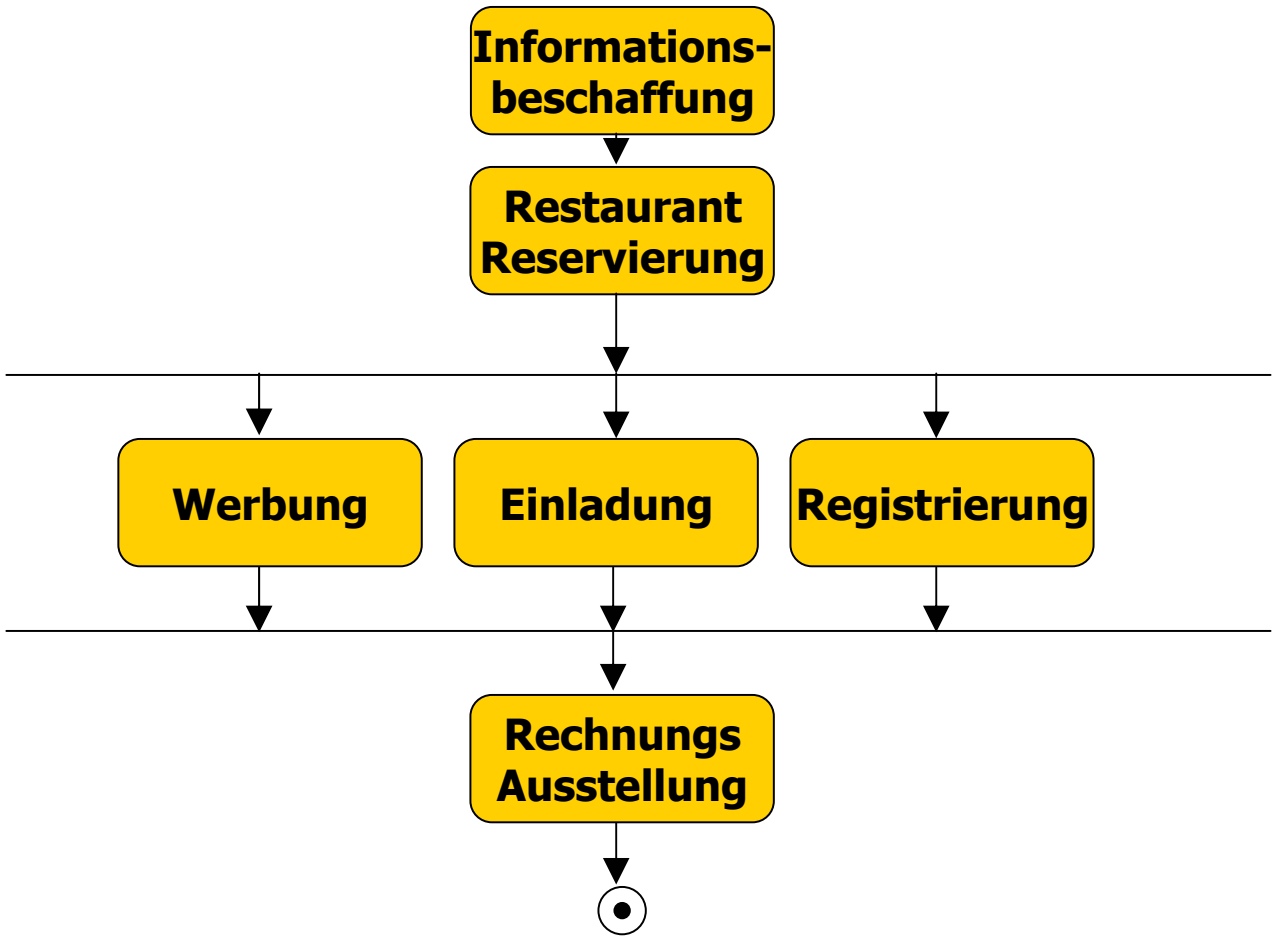
Composite E-Services

- ***„Composite e-services“*** sind Prozesse die andere composite oder Basis-Services zu einem Service zusammenfügen
- Composite services werden als Geschäftsprozesse modelliert und durch ein Prozessschema beschrieben
- Sie werden vom „***Service-Process-Engine“*** ausgeführt
- Ein Service wird durch einen Graphen dargestellt, welche die Ausführungsordnung zwischen den einzelnen Nodes angibt.

Nodes

- **Service Nodes**: stellen den Aufruf eines Basis/Composite-Services dar
- **Decision Nodes**: spezifizieren die Alternativen und Regeln, welche den Ausführungsfluss kontrollieren
- **Event Nodes**: ermöglichen es den Service-Prozessen verschiedene Eventtypen zu senden und zu empfangen

Prozess-Instanz-Schema



Service-Node-Spezifikation

- Die Service-Node-Spezifikation beinhaltet
  - welche Daten der Service Node zu lesen und ändern berechtigt ist,
  - die Beschreibung des Services, der aufgerufen werden soll
  - Die vom Kunden gestellten Anforderungen,
  - Einen letzten Ausführungstermin, im Falle eines Verzugs

Engine

- Prozessinstanzen werden vom Engine in Kraft gesetzt
- Der Engine kontrolliert den Zugang zu den Fall-Packet-Daten
- Seine Hauptaufgabe ist es Nachrichten zu bearbeiten
  - Nachricht über Zustand des Services
  - Mitteilungen eingetretener Events

Anpassungsfähige Service Prozesse

- Um mit den ständigen Veränderungen zurecht zu kommen und um daraus einen Vorteil zu ziehen, müssen Service-Prozesse anpassungsfähig („*adaptive*“) sein,
- d.h. Änderungen unter minimalsten Eingriff der Benutzer oder durch keinen Eingriff von außen durchzuführen.
- eFlow bietet verschiedene Funktionen an:
  - Dynamic Service Discovery
  - Multiservice Nodes
  - Generic Nodes

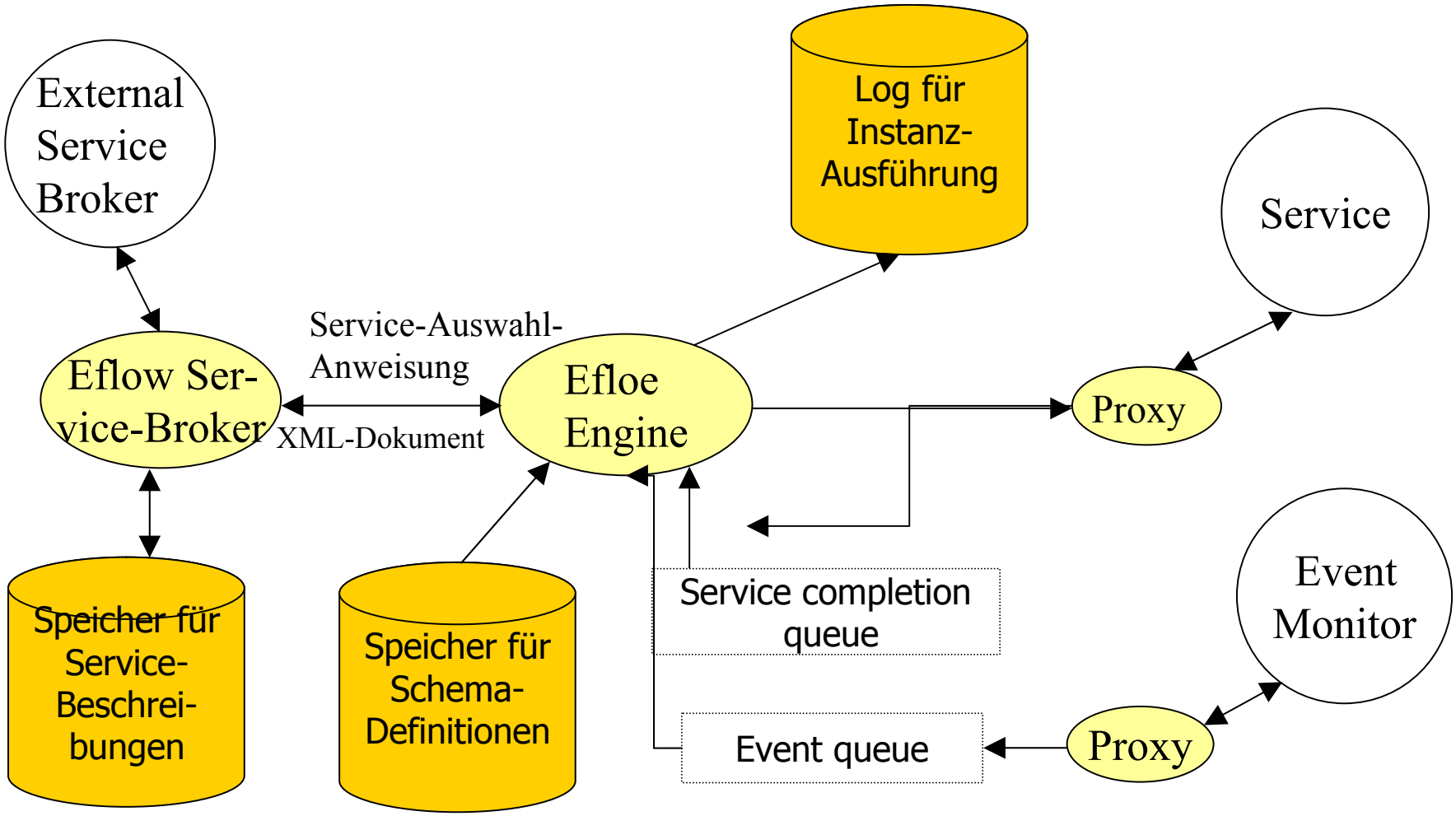
Dynamic Service Discovery

- Um mit den Charakteristiken der Internetumgebung klarzukommen bietet eFlow eine offene und dynamische Vorgehensweise für die Service-Auswahl
- Statische Service-Bindung ist oft zu starr, ermöglicht nicht:
  - Auswählen des passenden Services abhängig von Kundenwünschen
  - Dynamische Entdeckung der verfügbaren Services, welche am besten die Wünsche eines spezifischen Kunden erfüllen

Dynamic Service Discovery (2)

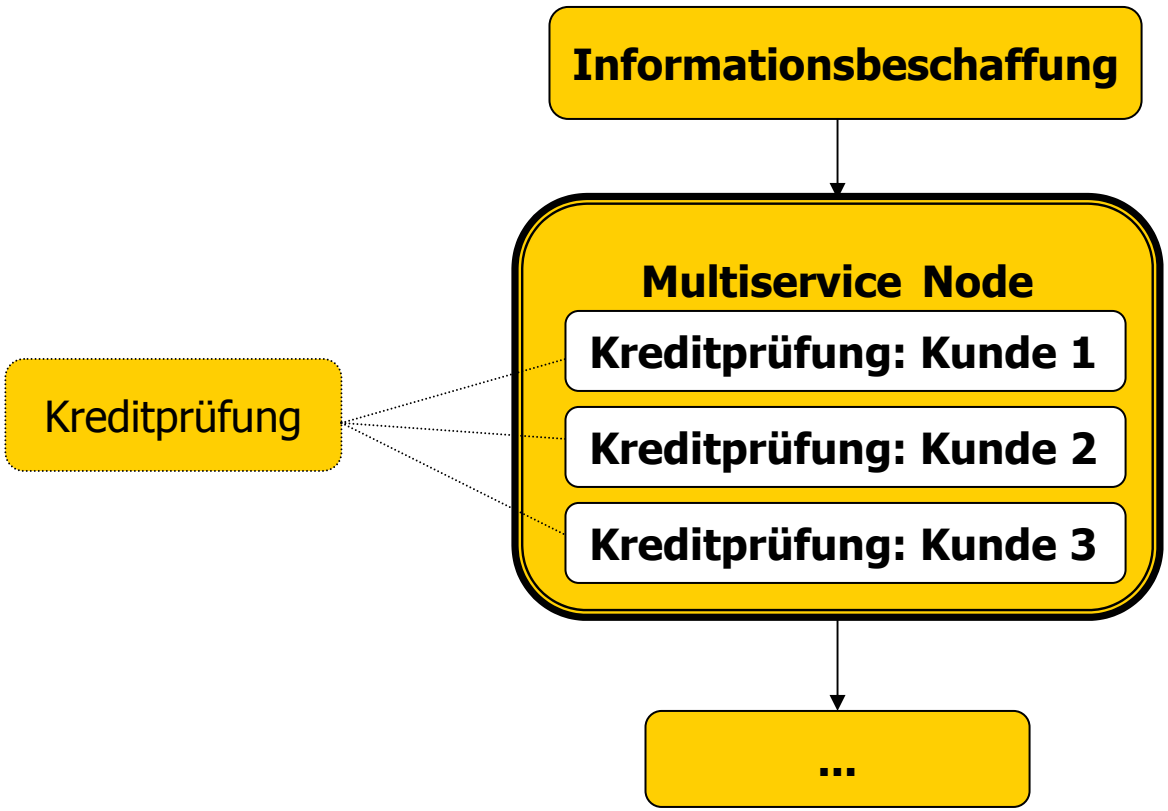
- Ist ein Service-Node gestartet, ruft der engine eine Service-Broker, einen Vermittler auf
- Benutzer können aber einen eigenen Vermittler auswählen: Plugged-in Broker
- Dieser führt die spezifische Regel aus und liefert den entsprechenden Service zurück
- Diese Service-Auswahl-Anweisung sind in einer Broker-spezifischen Sprache definiert, z.B. XQL für e-speak
- eFlow fordert nur, dass die Regel ein XML-Dokument zurück liefert

Dynamic Service Discovery (3)



Multiservice Nodes

Knoten zur **mehrfachen**, **parallelen** Aktivierung **des selben** e-Services.



Multiservice Nodes (2)

- Bestimmungs-Faktoren (zur Laufzeit) für die Anzahl der zu aktivierenden Instanzen hängt ab:
  - Anzahl der Service-Anbieter
  - Inputparameter
- Terminierungs-Bedingung:
  - Abschluss aller Services
  - Erhalten eines bestimmten Ergebnisses

Multiservice Nodes (3)

- Spezifikation eines Multiservice-Nodes

...

<MULTISERVICE_NODE id="check_customer_credit">

<NAME> check Customers´credit </NAME>

<SERVICE_NODE id="check_single_customer_credit" />

<DESCRIPTION> Multiservice der die Kredit Historie

verschiedener Kunden parallel prüft

</DESCRIPTION>

<ACTIVATION mode="by_variable"
varref="custermers_list" />

<TERMINATION> rejections.length > 0 </TERMINATION>

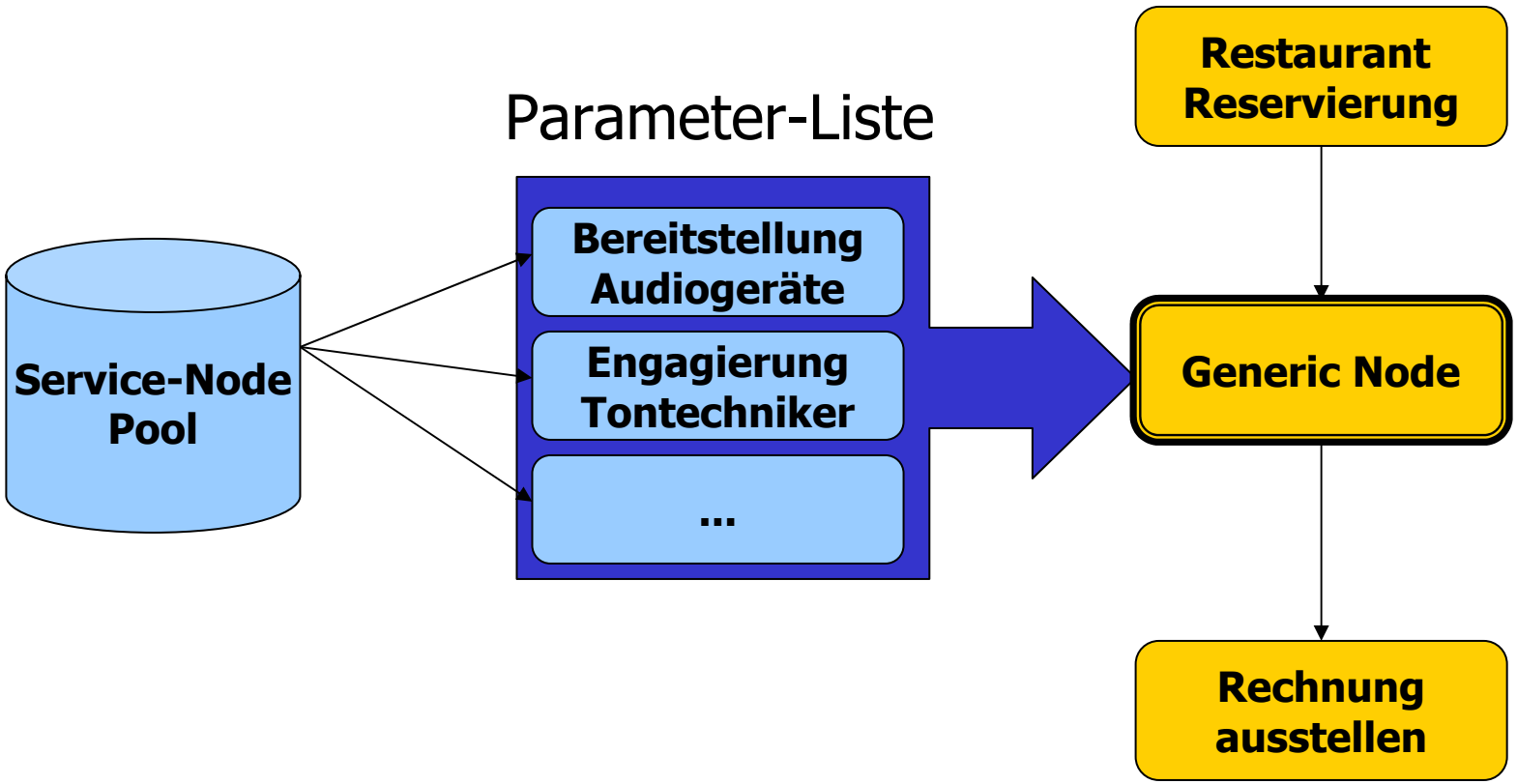
<MULTISERVICE_NODE>

...

Generic Service Nodes

- Knoten zur Aktivierung **mehrerer**, **unterschiedlicher** e-Services.
- Flexible Gestaltung der e-Services zur Befriedigung individueller Kundenwünsche

Generic Service Nodes (2)



Generic Service Nodes (3)

- Enthält einen Parameter für die Services
  - Parameter ist vom Typ `ListOf(Service_Node)`
- Ausführungsweise wird im Attribut *executionMode* festgelegt:
  - sequential (Schleife)
  - parallel

Generic Service Nodes (4)

- Spezifikation eines Generic Service Nodes

...

```
<GENERIC_NODE id="award_ceremony_services>
        <NAME> Award Ceremony Service </NAME>
        <SERVICE_NODE_POOL> Ceremony Service Pool
        </SERVICE_NODE_POOL>
        <DESCRIPTION> Platzhalter für Service-Knoten, die
                      sich auf einen Ceremony-Service
                      beziehen und parallel ausgeführt
                      werden
        </DESCRIPTION>
        <SERVICE_SELECTION_VAR> SelectedServices
        </SERVICE_SELECTION_VAR>
        <EXECUTION_MODE mode="parallel" />
</GENERIC_NODE>
```

...

Modifikationen
Service Prozess

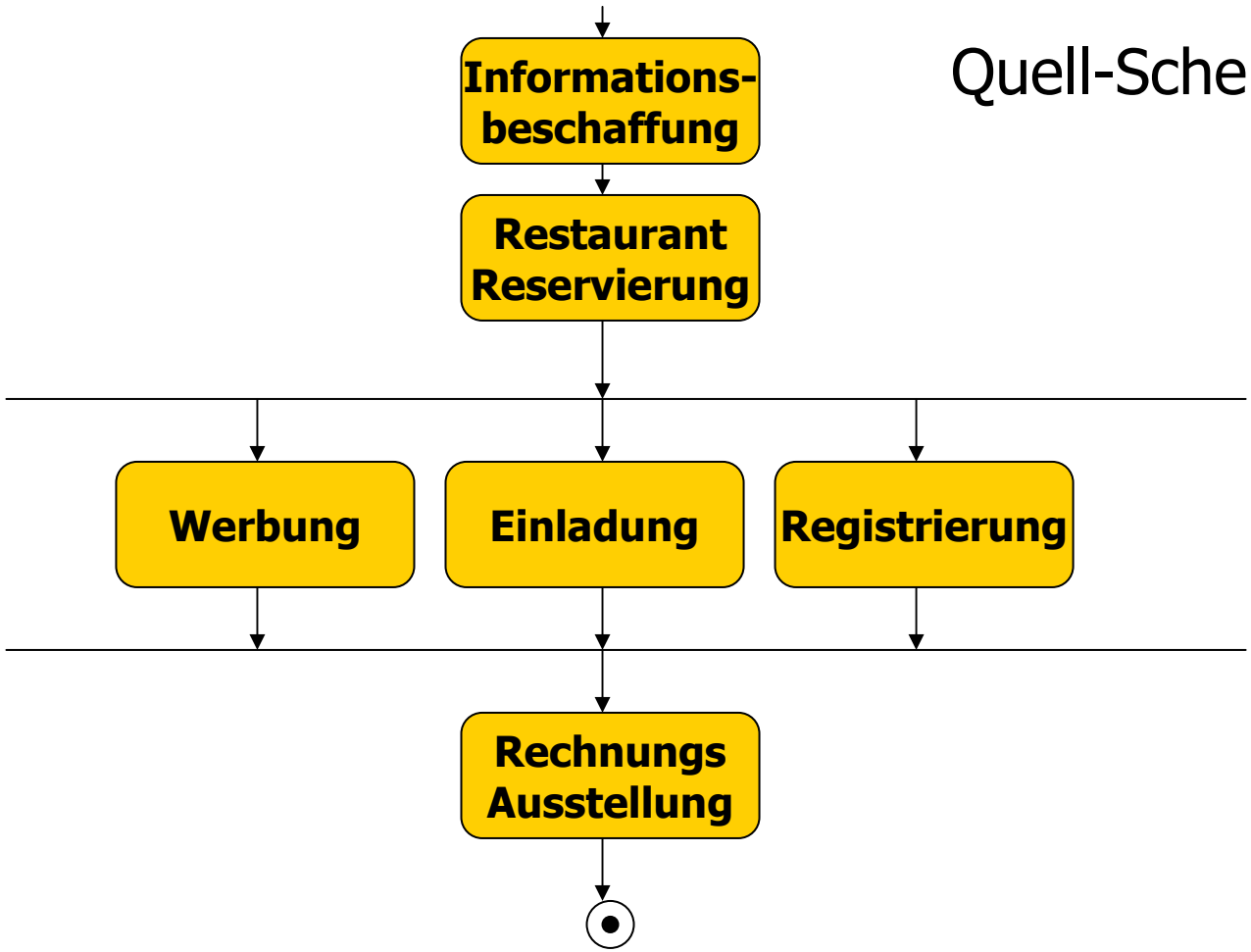
- Ursachen
  - neue Gesetzeslage oder Geschäftsstrategie
  - Prozessoptimierung
  - Fehlerkorrektur
  - Mangelhaftigkeit der aktuellen Definition
- Modifikationsarten:
  - Ad-hoc Modifikation
  - Bulk-Modifikation

Ad-hoc-Modifikation

- Modifikation **einer einzelnen**, laufenden Prozess-Instanz

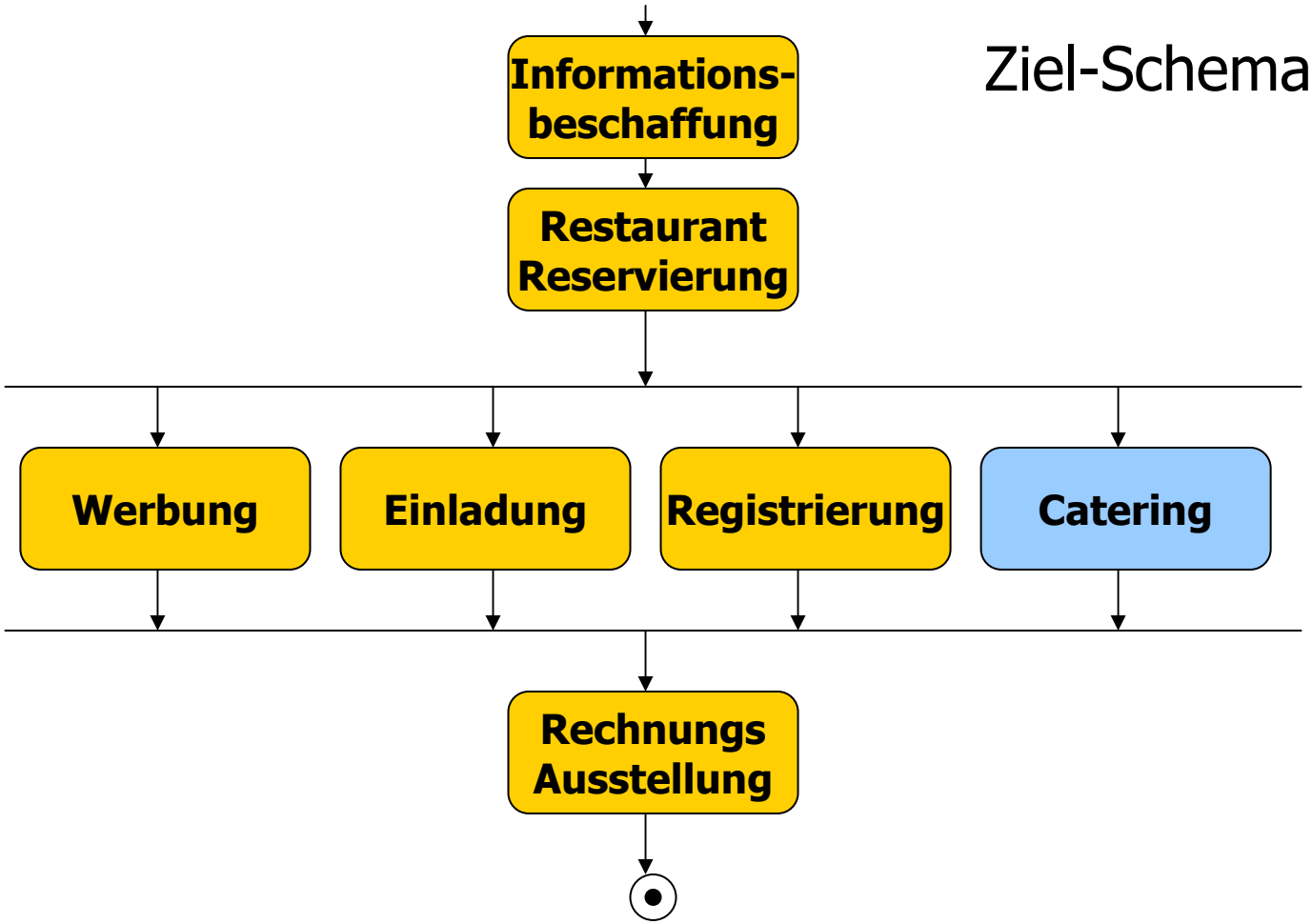
- Zwei Arten:
  - Änderung des Prozess-Instanz-Schemas
  - Änderung des Prozess-Instanz-Status

Ad-hoc-Modifikation (2)
Prozess-Instanz-Schema



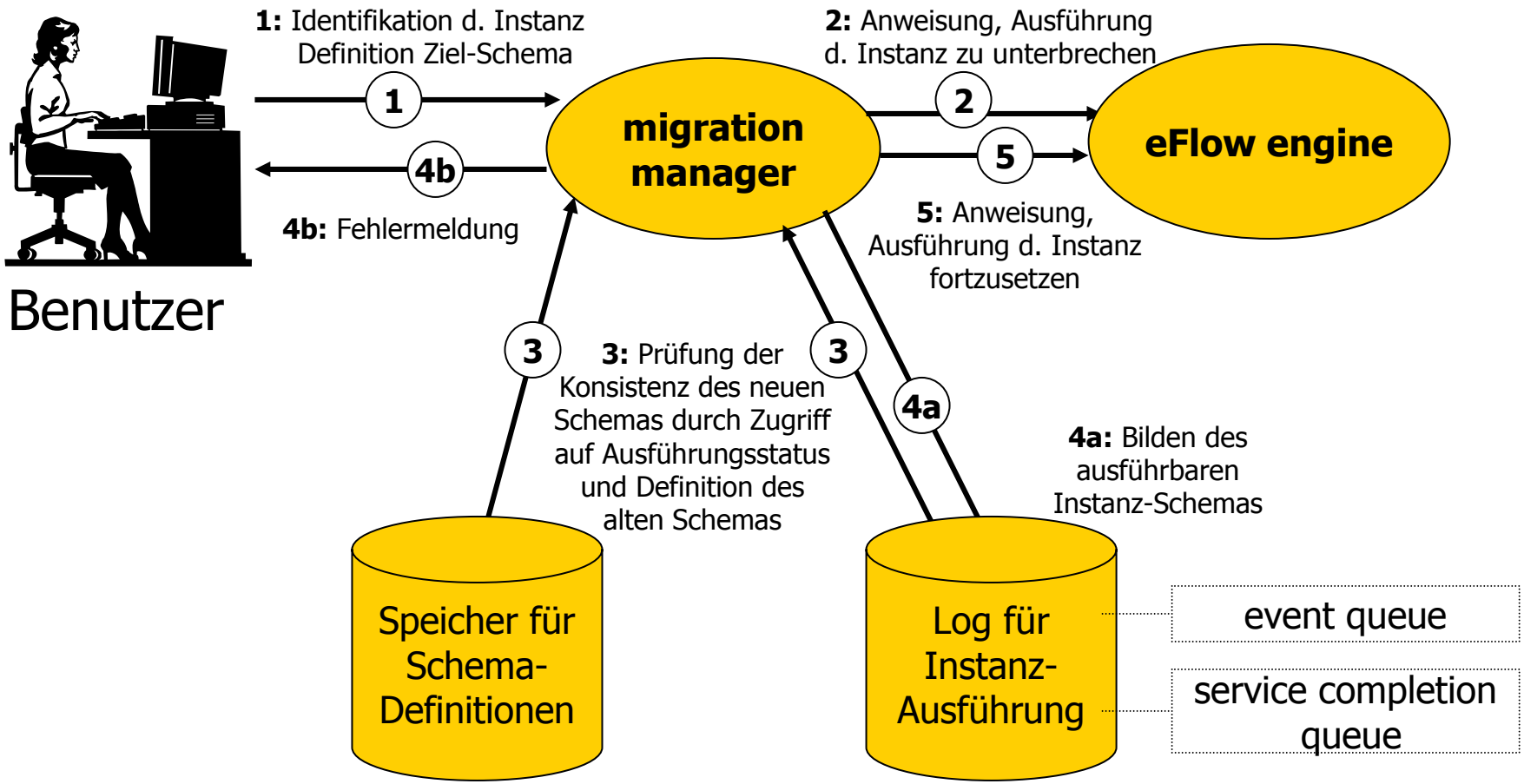
Quell-Schema

Ad-hoc-Modifikation (3)
Prozess-Instanz-Schema



Ziel-Schema

Ad-hoc-Modifikation (4)
Vorgehen



Ad-hoc-Modifikation (5)
Konsistenz-Regeln

- Gerade aktive Knoten müssen im Ziel-Schema enthalten sein
- Variable, die im Quell- und Zielschema enthalten sind, müssen vom gleichen Typ sein, damit die Werte nach einer Migration erhalten bleiben

Bulk-Modifikation

- Modifikation **mehrerer**, laufender Prozess-**Instanzen des selben Prozesses** mit gleichen Eigenschaften

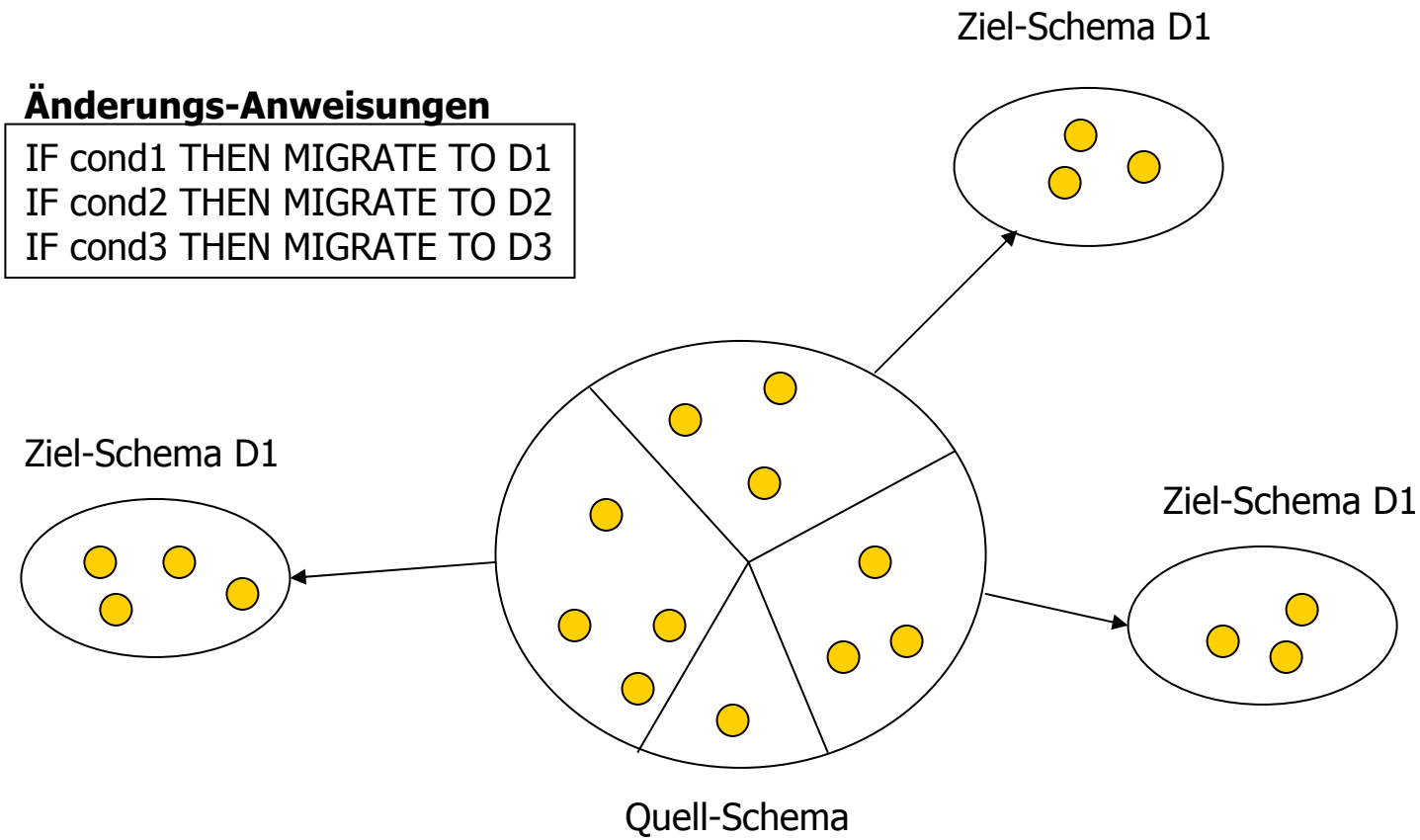
- Beispiel: Änderungs-Anweisung

```
IF (guests > 100) THEN MIGRATE TO "Security_Ceremony_Service"
```

Bulk-Modifikation (2)
Beispiel

Änderungs-Anweisungen

IF cond1 THEN MIGRATE TO D1
IF cond2 THEN MIGRATE TO D2
IF cond3 THEN MIGRATE TO D3



Bulk-Modifikation (3)
Vorgehen

1. Definition, Compilierung und Überprüfung der Überführungs-Vorschrift
2. Unterbrechung aller laufender Instanzen des Prozesses
3. Prüfung der Erfüllung der Bedingung der Änderungs-Anweisung
4. Prüfung der Konsistenz-Regeln und der Autorisierung
5. Durchführung der Änderung
6. Fortsetzen der Ausführung

Sicherheits-Regeln

- Berechtigungen in Abhängigkeit vom Ausführungs-Status
- Berechtigungen:
  - *Authorized_State_Modifiers*
  - *Authorized_Node_Modificators*
  - *Authorized_Flow_Modificators*
  - *Authorized_Initiators*

Implementierung

- Prototyp von HP (F. Casati, M. Shan) basierend auf e-speak und Process Manager
  http://www.research.microsoft.com/research/db/debull/A01mar/issue.htm

- Ansatz der Universität Saarland
  (G. Shegalov, M. Gillmann, G. Weikum) Java-basiert und XML-basiert
  http://www-dbs.cs.uni-sb.de/~gillmann/Publications/XML-TES.pdf

Literatur

- F. Casati, S Ilnicki, L. Jin, V. Krishnamoorthy, M. Shan, Adaptive and Dynamic Service Composition in eFlow, Technical Report HPL-2000-39, HP Software Technology Laboratory, März 2000: http://www.hpl.hp.com/techreports/2000/HPL-2000-39.pdf

- F. Casati, S Ilnicki, L Jin, V. Krishnamoorthy, M Shan, eFlow: a Platform for Developing and Managing Composite e-Services, Technical Report HPL-2000-36, HP Software Technology Laboratory, März 2000: http://www.hpl.hp.com/techreports/2000/HPL-2000-36.pdf

- F.Casati, M. Shan, Definition, Execution, Analysis, and Optimization of Composite E-Services, HP Laboratories, 2001: http://www.research.microsoft.com/research/db/debull/A01mar/issue.htm