



# ServiceGlobe: Flexible and Reliable Web Service Execution

*Markus Keidl, Stefan Seltzsam und Alfons Kemper*

**Universität Passau**

**Fakultät für Mathematik und Informatik**

**94030 Passau**

**<nachname>@db.fmi.uni-passau.de**

# [ Gliederung ]

- Motivation
- Das ServiceGlobe-System
- Dynamic Service Selection
- Automatic Service Replication
- Zusammenfassung und Ausblick

# [ Motivation ]

- Ausführung von Web Services
  - Verteilte Ausführung: Nutzung der Rechenzeit im Internet
  - Parallelisierung
- Robuste Ausführung

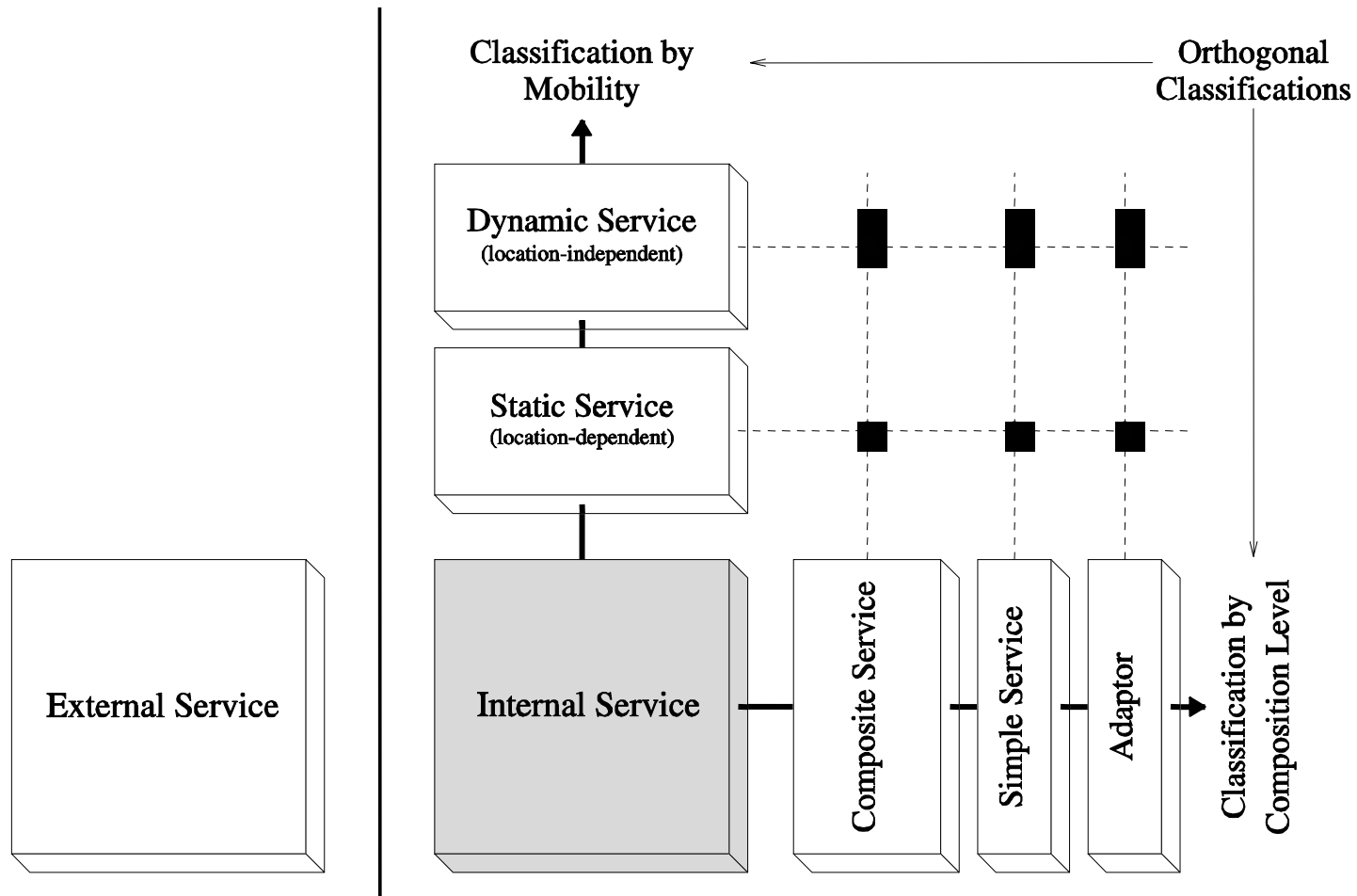
# Was ist ServiceGlobe?

- Forschungsplattform für Dienste
  - Implementiert in Java
  - Basierend auf Standards (XML, SOAP, UDDI, ...)
- Entwicklung von neuen Möglichkeiten für die Ausführung von Diensten

# [Eigenschaften von ServiceGlobe]

- Dienste sind mobiler Code
  - Laden des Codes von Diensten zur Laufzeit
  - Verteilung von Diensten  
⇒ *Runtime Service Loading*
- Sicherheitssystem
- Transaktionssystem

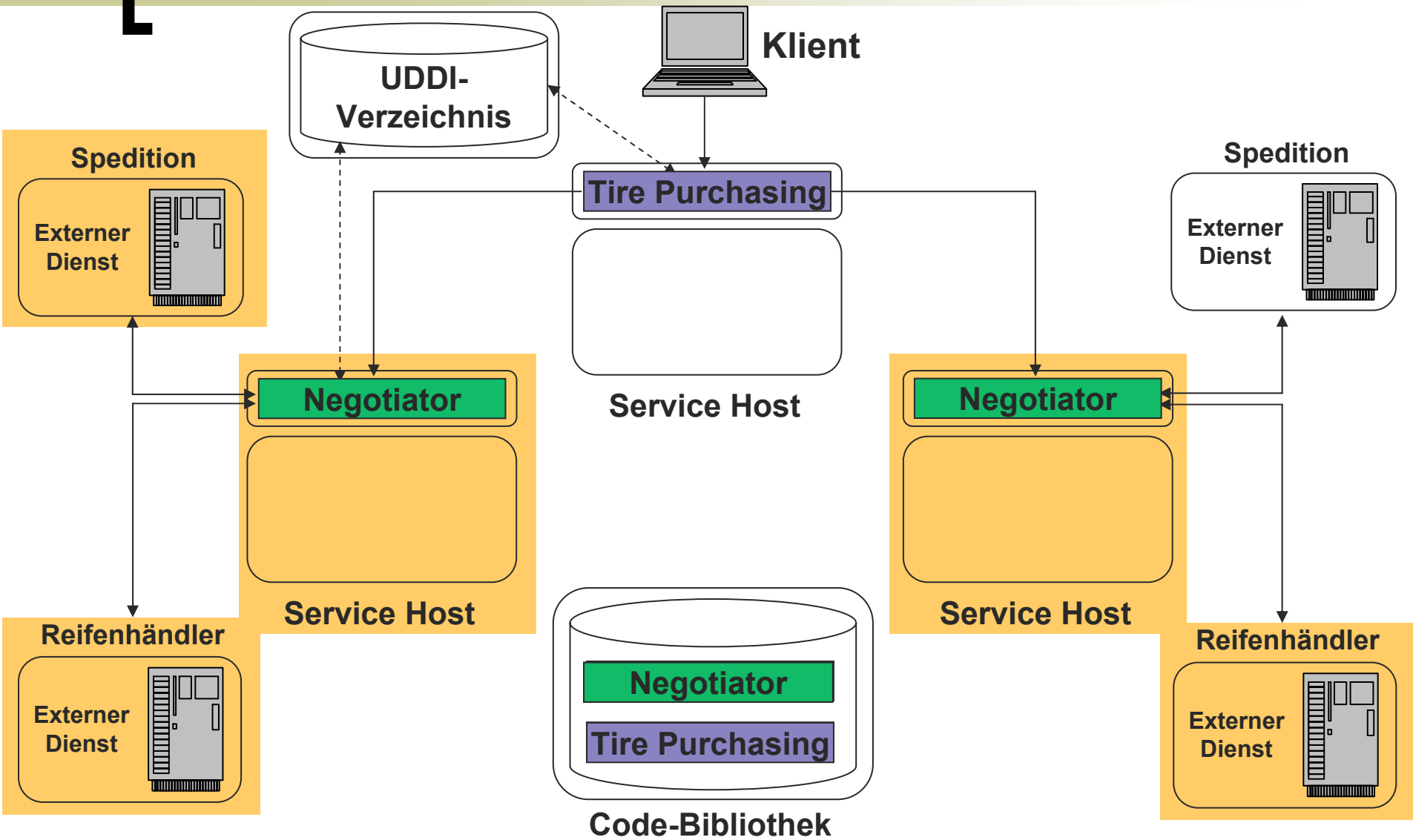
# Klassifikation von Diensten



# Beispielszenario: E-Procurement

- Aufgabe: Einkauf von Reifen und Beauftragung einer Spedition für die Lieferung
- Details:
  - Angebote von Reifenhändlern einholen
  - Angebote von Speditionen für Lieferung einholen
  - Billigstes kombiniertes Angebot berechnen
  - Auftrag vergeben
- Aufteilung in zwei Diensten:
  - *Tire Purchasing* und *Negotiator*
  - Ausführung der *Negotiator*-Dienste auf Rechnern, die nahe bei den Reifenhändlern liegen
  - Parallele Ausführung aller *Negotiator*-Dienste

# Architektur von ServiceGlobe



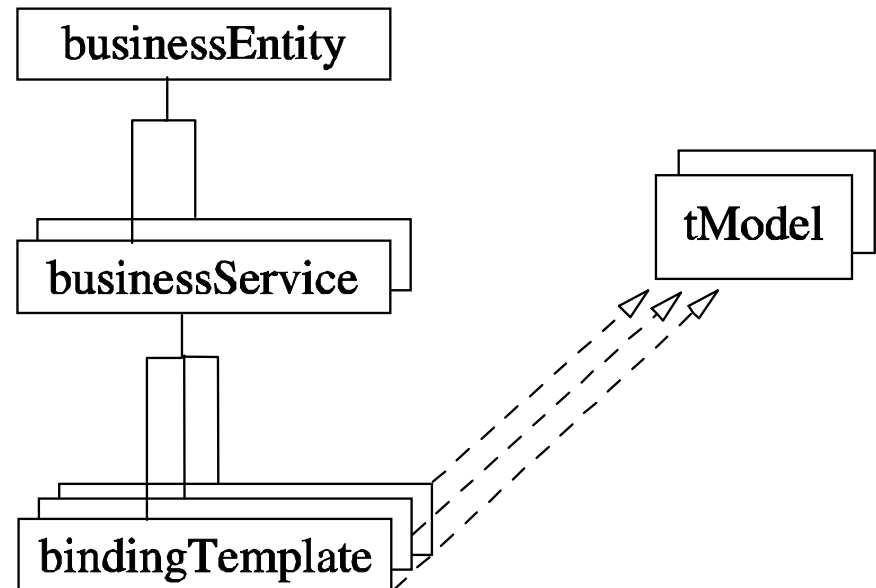


# [ Gliederung ]

- Motivation
- Das ServiceGlobe-System
- **Dynamic Service Selection**
- Automatic Service Replication
- Zusammenfassung und Ausblick

# [ UDDI – Eine kurze Einführung ]

- Datenstrukturen:
  - businessEntity
  - businessService
  - bindingTemplate
  - tModel  
(technical model)



# [ Dynamic Service Selection ]

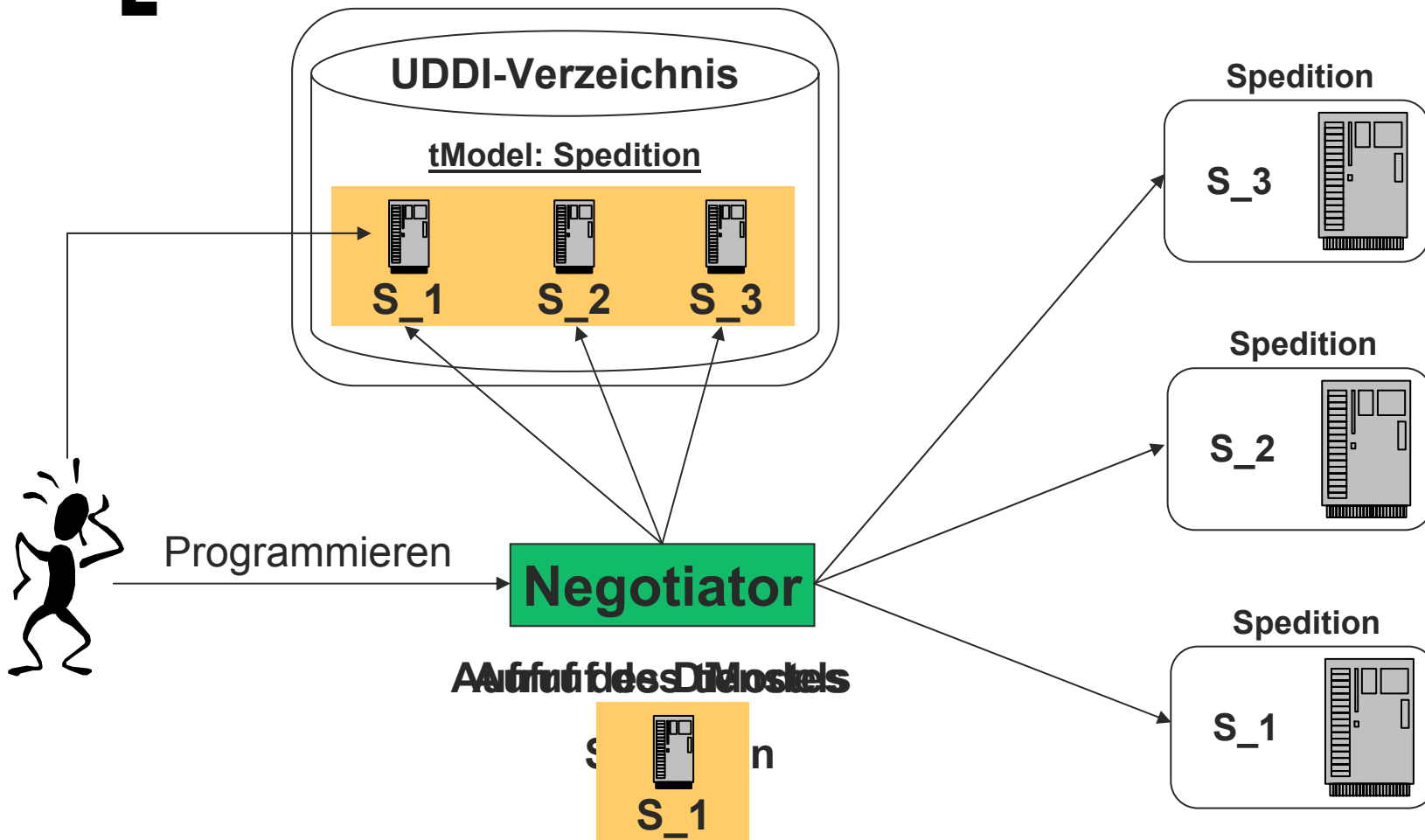
- UDDI: Zuordnung von Diensten zu tModels (Beschreibung der Funktionalität und des Interfaces)
- 'Aufruf eines tModels' anstatt 'Aufruf eines Dienstes'
- Beeinflussung von Auswahl und Aufruf:
  - Aufrufmodi: one/some/all
  - Vorgaben

# [ Vorgaben ]

- Präferenzen, Bedingungen
- Typen: Metadaten, Rechner, Antwort, Aufruf
- Beispiele:

```
<metadataPreference>  
  /businessEntity/name="Spedition Schnell"  
</metadataPreference>  
  
<metadataCondition>  
  /serviceMetadata/costsPerCall="0"  
</metadataCondition>
```

# Beispiel zur Dienstauswahl



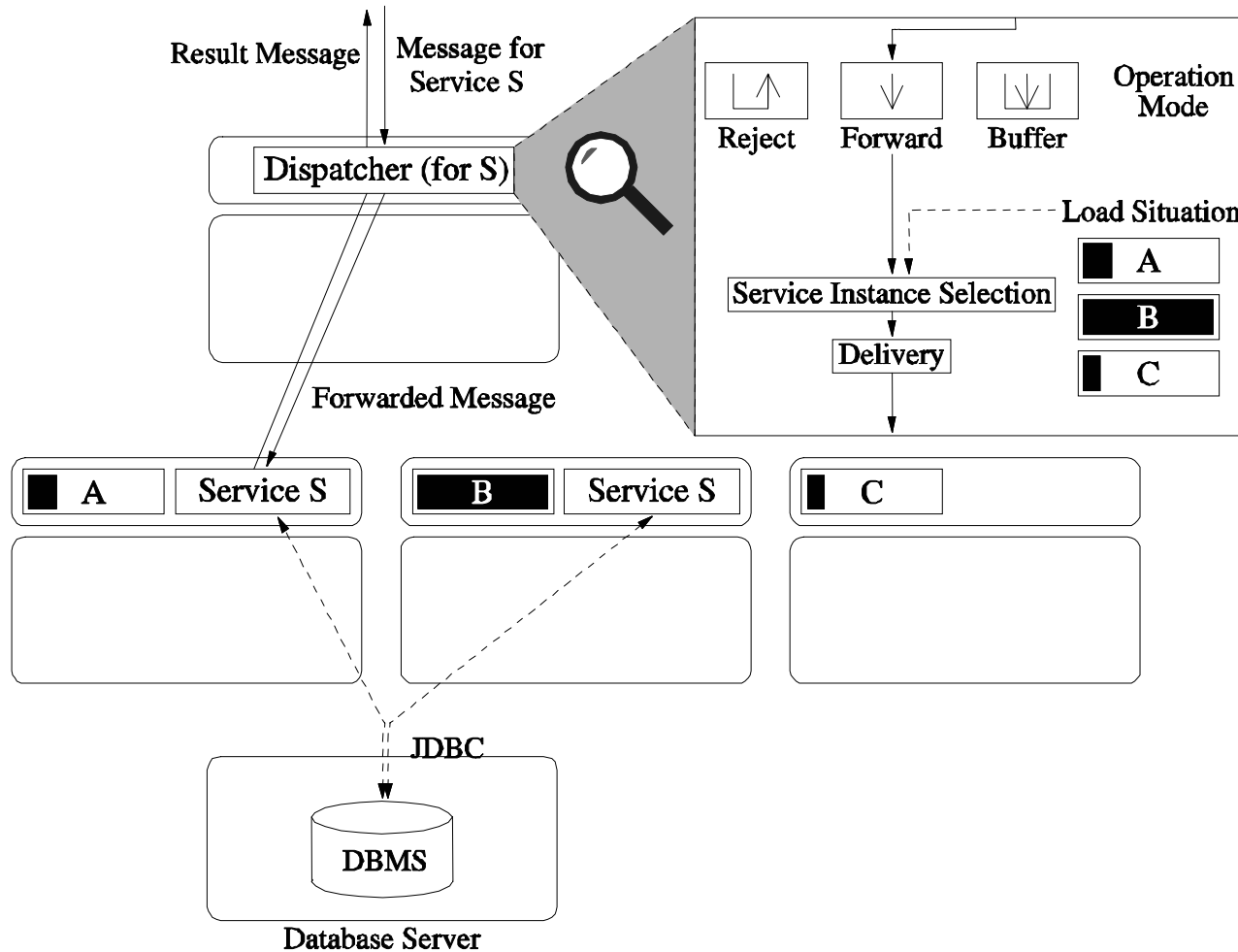
# [ Gliederung ]

- Motivation
- Das ServiceGlobe-System
- Dynamic Service Selection
- **Automatic Service Replication**
- Zusammenfassung und Ausblick

# Automatic Service Replication

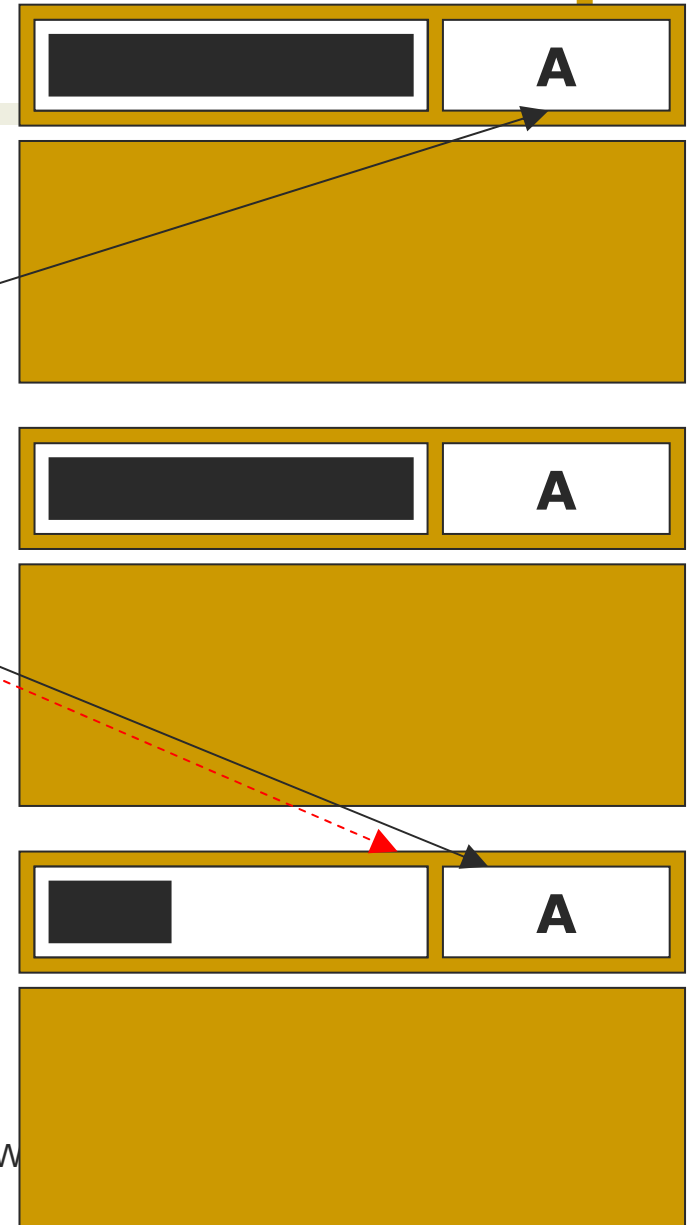
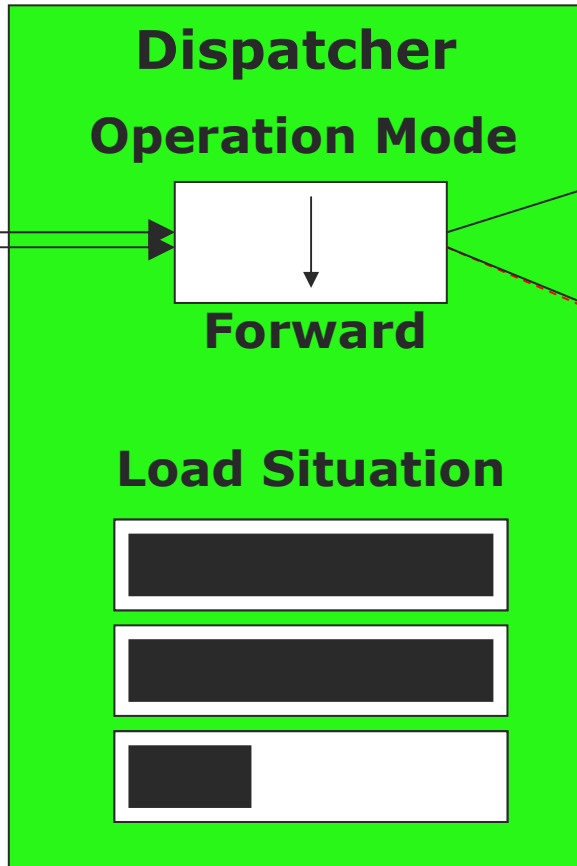
- Ziel: Lastbalancierung und hohe Verfügbarkeit ohne Änderung der Implementierung eines Dienstes
- Generischer Dispatcher-Dienst:
  - Proxy zwischen Aufrufendem und Dienst(en)
  - Überwachung der Service Hosts
  - Verteilung von Anfragen auf Service Hosts und Dienste
  - Instantiierung von zusätzlichen Diensten

# Dispatcher-Dienst

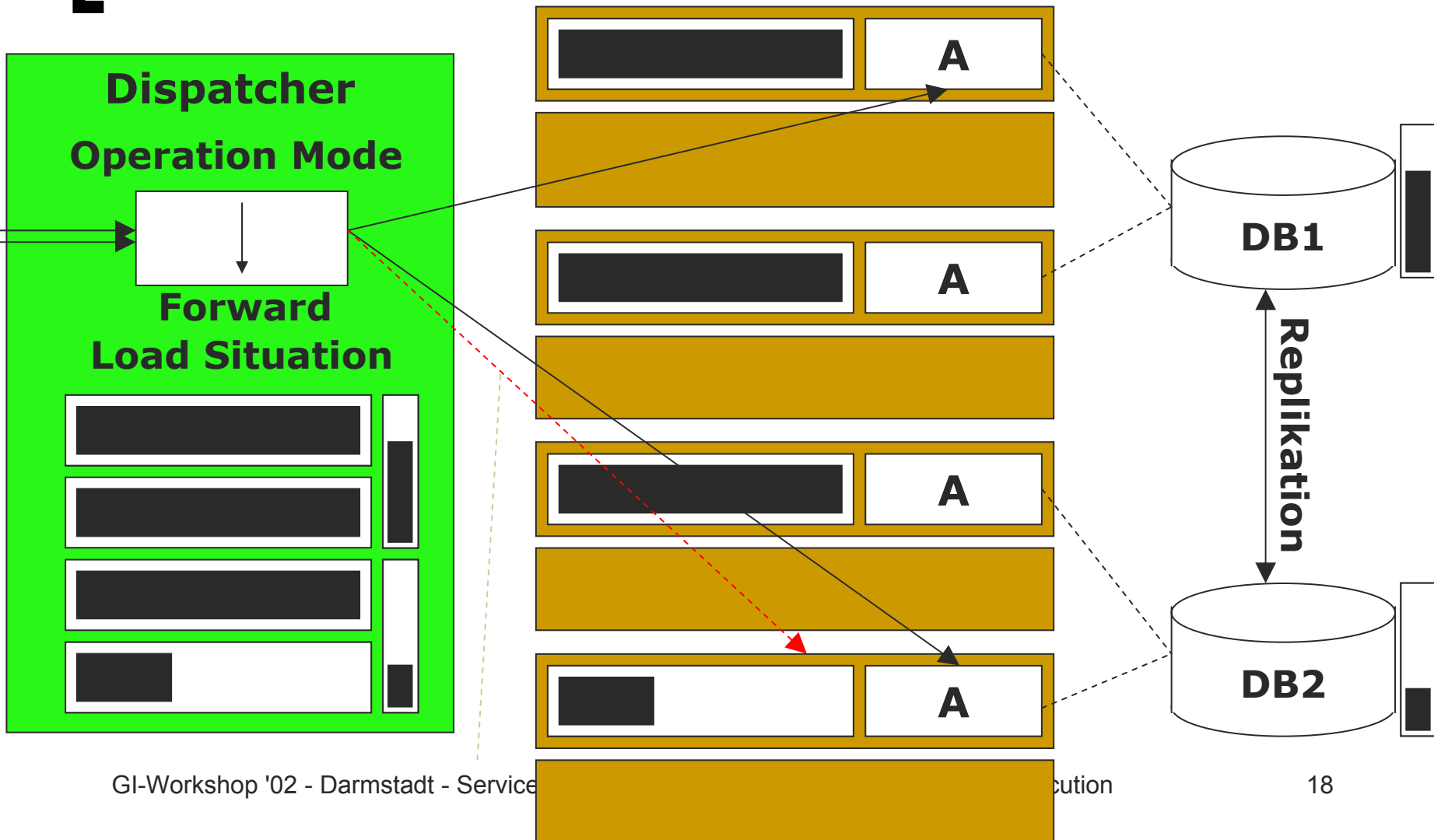




# Einsatz der Automatic Service Replication



# Automatic Service Replication in komplexen Szenarien



# [ Zusammenfassung ]

- Dienstplattform: ServiceGlobe
  - Architektur
  - Schlüsseleigenschaften
- Dynamic Service Selection
- Automatic Service Replication
- ⇒ flexible und robuste Ausführung von Diensten

# Ausblick

- Automatic Service Replication:
  - Konfiguration von neu gestarteten Diensten
  - Integration von Caching
- Kontext für Dienste
- Integration von Vorgaben in den Kontext eines Dienstes
- Weitere Informationen:
  - <http://www.db.fmi.uni-passau.de/projects/sg>
  - Demo auf der VLDB'02 in Hong Kong