Reliable Web Service Execution and Deployment in Dynamic Environments*

Markus Keidl, *Stefan Seltzsam*, and Alfons Kemper

Universität Passau
94030 Passau, Germany
`<lastname>@db.fmi.uni-passau.de`

* This research is done in cooperation with the Advanced Infrastructure Program (AIP) group of SAP.

Outline

- Architecture of ServiceGlobe
- Dynamic Service Selection
- Load Balancing and Service Replication
- Current Utilization of the Techniques
- Conclusion

Outline

- Architecture of ServiceGlobe
- Dynamic Service Selection
- Load Balancing and Service Replication
- Current Utilization of the Techniques
- Conclusion

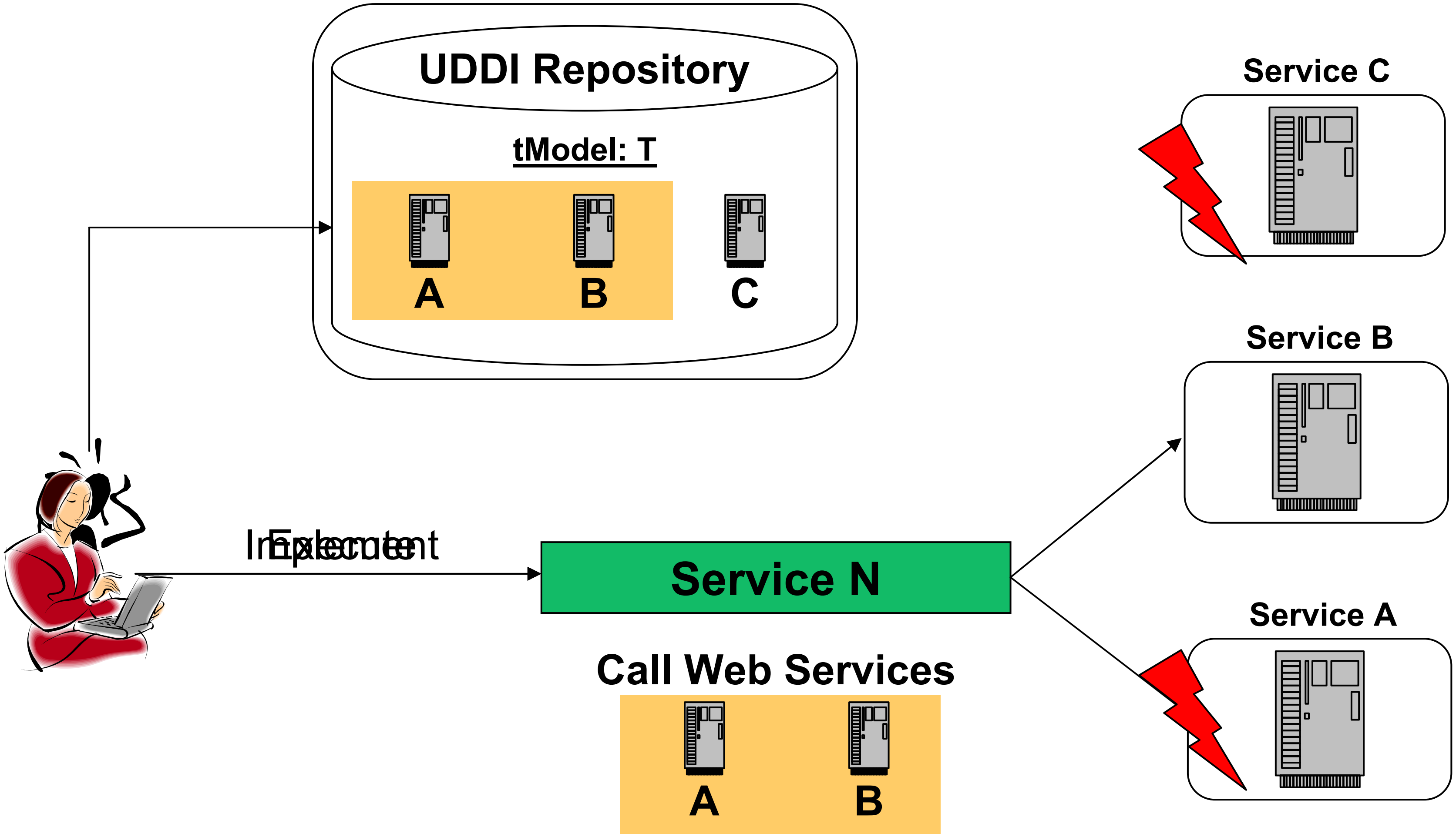
The ServiceGlobe System

- ServiceGlobe is an open, distributed, and extensible service platform developed for research purposes
- Fully implemented in Java 2
- Based on standards like XML, SOAP, UDDI, WSDL,…
- Offers standard functionality of a service platform
  - □ secure communication
  - □ transaction system
  - □ security system
- Supports mobile code, i.e., services can be distributed and instantiated during runtime on demand
  - → needed for automatic service replication

Outline

- Architecture of ServiceGlobe
- Dynamic Service Selection
- Load Balancing and Service Replication
- Current Utilization of the Techniques
- Conclusion

Motivation Dynamic Service Selection



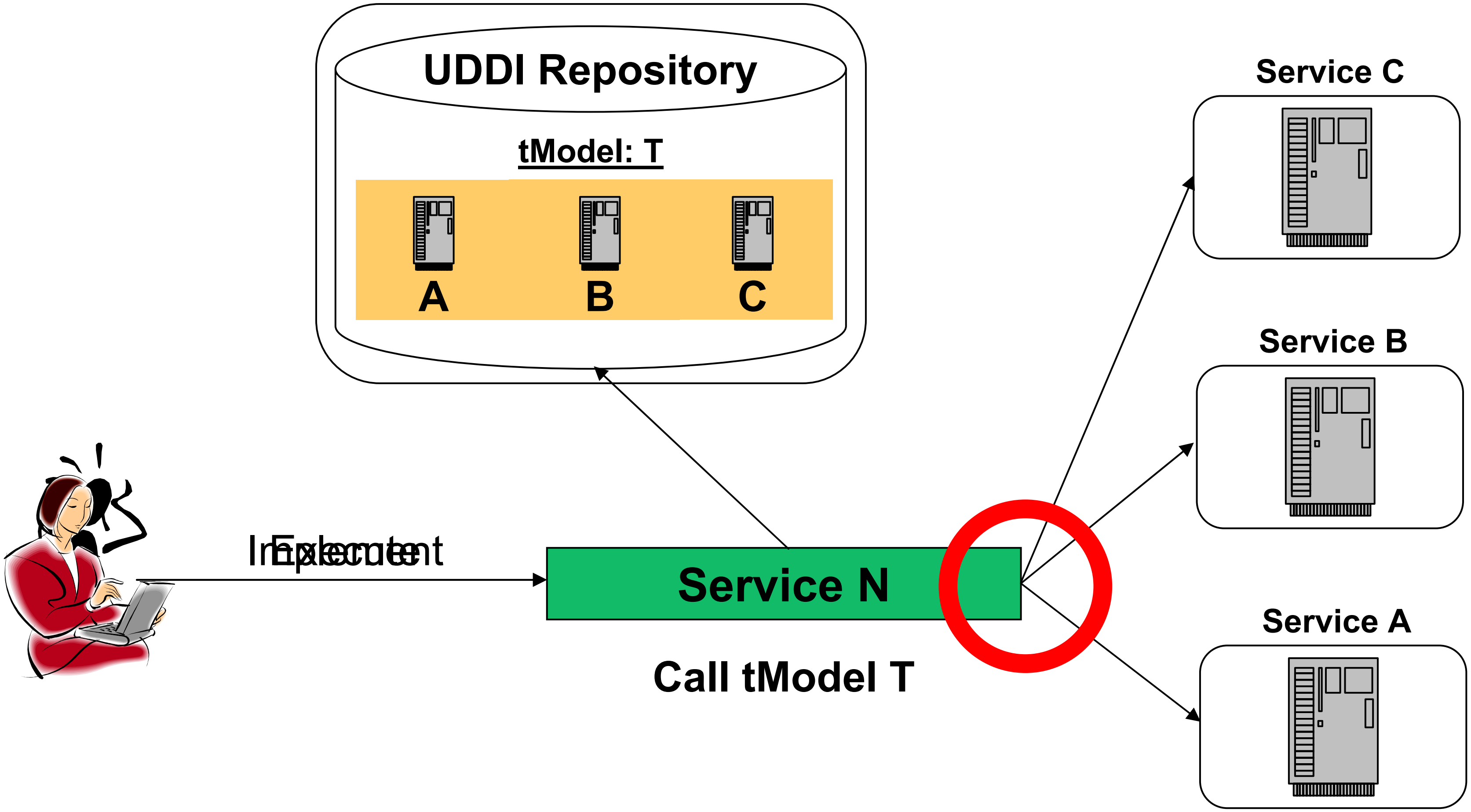
Dynamic Service Selection

- DSS uses UDDI:
  - □ UDDI assigns Web services to tModels
  - □ tModel: semantic classification of functionality and formal description of interface
- DSS is ‘Calling a tModel' instead of the traditional ‘Calling a Web service‘
- Web services are selected at runtime not at development time.
- Constraints can be used to influence Web service selection and invocation

→ Dynamic fusion of Web services

→ Reliable Web service execution

Survey of Dynamic Service Selection



Classification of Constraints

	Metadata Constraints	Location Constraints	Mode Constraints	Reply Constraints		Result Constraints	
				Selection	Property	Timeout	First-N
Preferences			X			X	X
Conditions							

Apply to Metadata
(UDDI, …)

Apply to SOAP
Response

Constraints

- Types of constraints:
  - □ Metadata Constraints
  - □ Location Constraints
  - □ Mode Constraints
  - □ Reply Constraints
  - □ Result Constraints
- Constraints are either Preferences or Conditions
- Constraints can be combined using logical operators

Metadata Constraints

- XPath queries applied to service metadata
- Metadata: UDDI + additional metadata
- Examples:

```
<metadataPreference>
  /businessEntity/name= "Company"
</metadataPreference>

<metadataCondition>
  /ServiceMetadata/CostsPerCall="0"
</metadataCondition>
```

Location Constraints

- Selection based on the location of services (stored in UDDI)
- Example:

```
<locationCondition
     addressType="Geographical"
     serviceType="All">
  <center>DE-BY-PAS</center>
  <maxDistance>50km</maxDistance>
</locationCondition>
```

Mode Constraints

- Specify the number of Web services to invoke
- Modes: One, Some, All
- Example:

```
<modeCondition modeType="Some"
               number="5%"/>
```

Reply Constraints

- Property Constraints:
  - Query properties of replies (signature, age of data, encryption,…)
  - Insertion of properties by Web services, service platforms
- Selection Constraints: XPath queries applied to SOAP response
- Example:

```
<propertyCondition>
   <maxAgeOfData unit="hours">2</maxAgeOfData>
</propertyCondition>
```

Result Constraints

- Timeout Constraints
- First-N Constraints
- Examples:

```
<timeoutCondition value="100"
                  valueUnit="Seconds"/>

<firstNCondition number="10"
                 numberType="Percentage"/>
```

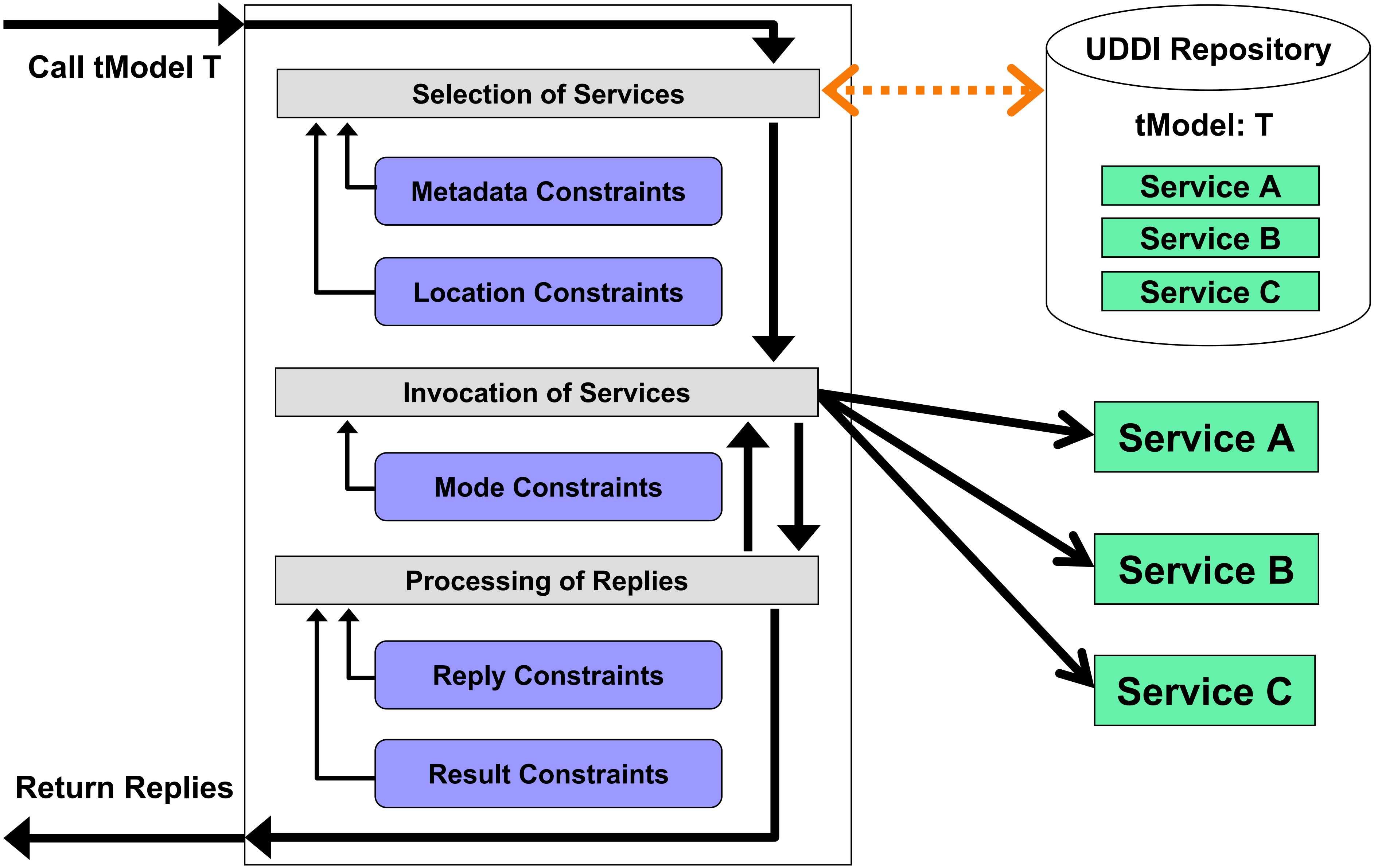
Combination of Constraints

- Operators: AND, OR
- Example:

```
<andGroup>
  <metadataCondition>
    /businessEntity/name="Company"
  </metadataCondition>
  <timeoutCondition
     value="100"
     valueUnit="Seconds" />
</andGroup>
```

Evaluation of Constraints

- Combine all constraints for a single tModel call conjunctively
- Transformation into DNF
- Resolve conflicts based on priorities
- Parallel evaluation of disjunctively combined AND terms


Call tModel T
Selection of Services
Metadata Constraints
Location Constraints
Invocation of Services
Mode Constraints
Processing of Replies
Reply Constraints
Result Constraints
Return Replies
UDDI Repository
tModel: T
Service A
Service B
Service C
Service A
Service B
Service C

Outline

- Architecture of ServiceGlobe
- Dynamic Service Selection
- Load Balancing and Service Replication
- Current Utilization of the Techniques
- Conclusion

Motivation

- A single service host is often not sufficient to provide low response times.
- Downtime can be costly.
- Integration of load balancing into services is error-prone and expensive.

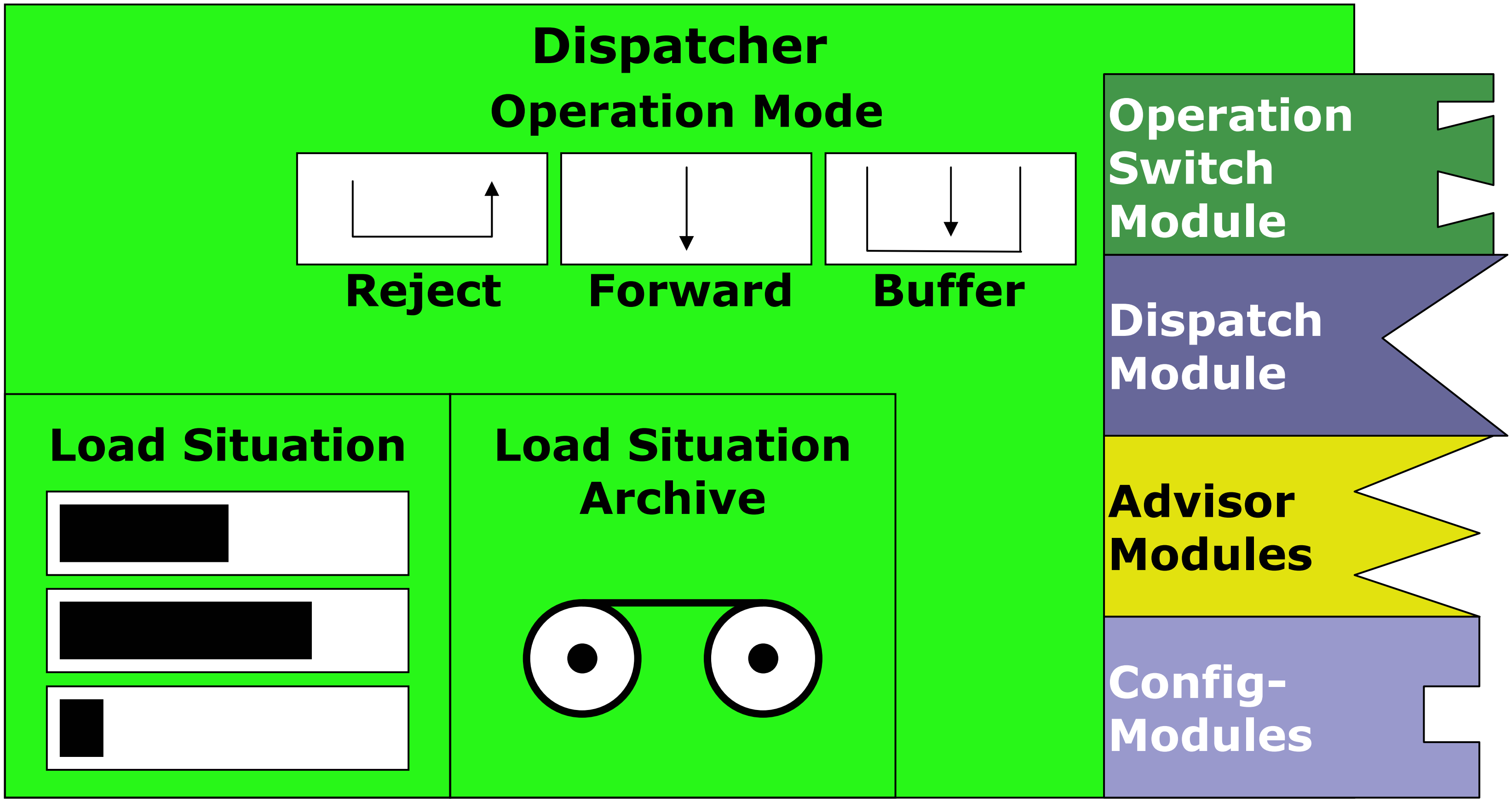
➔Modular dispatcher service

Advantages of a Modular Dispatcher Service

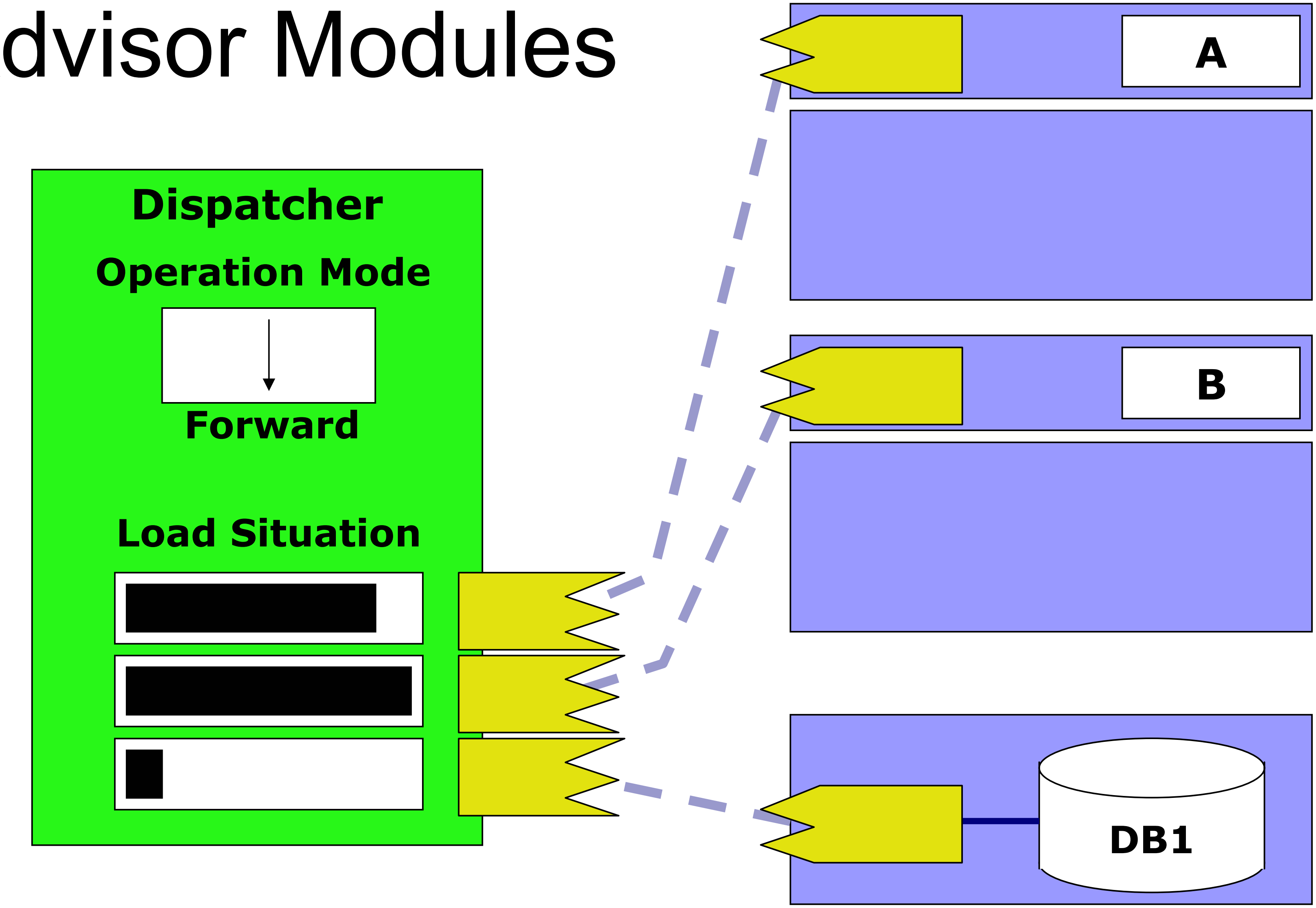
- Can act as proxy for arbitrary services
- Software-based layer-7 switch realized as a regular service
  - seamlessly integrated into the platform
- It is possible to enhance many existing services or develop new services with load balancing and high availability features
  - as long as Concurrency control mechanisms are used (e.g., DBMS as back end)
- Additional feature: automatic service replication

→Reliable Web service deployment

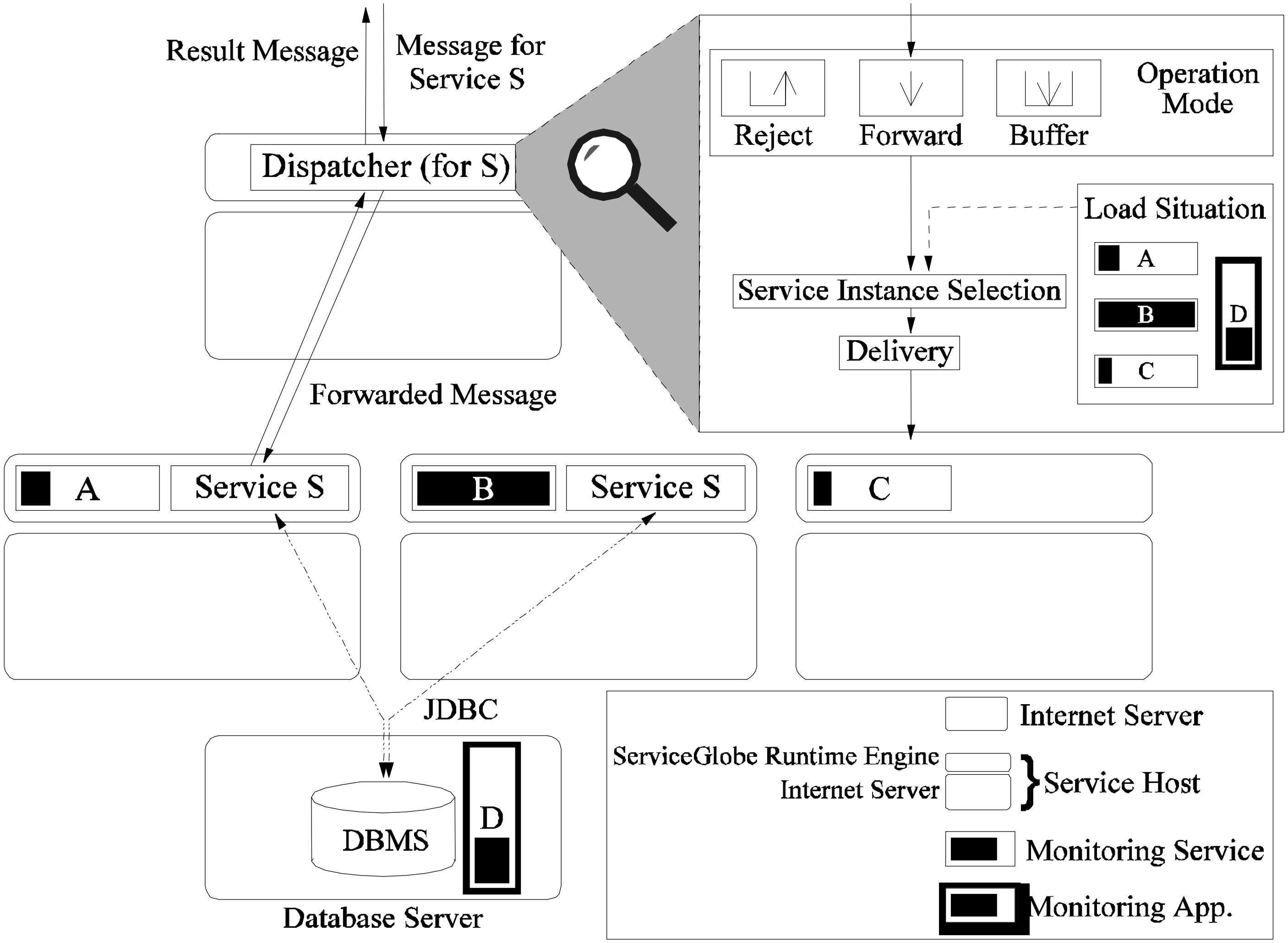
Architecture of the Dispatcher



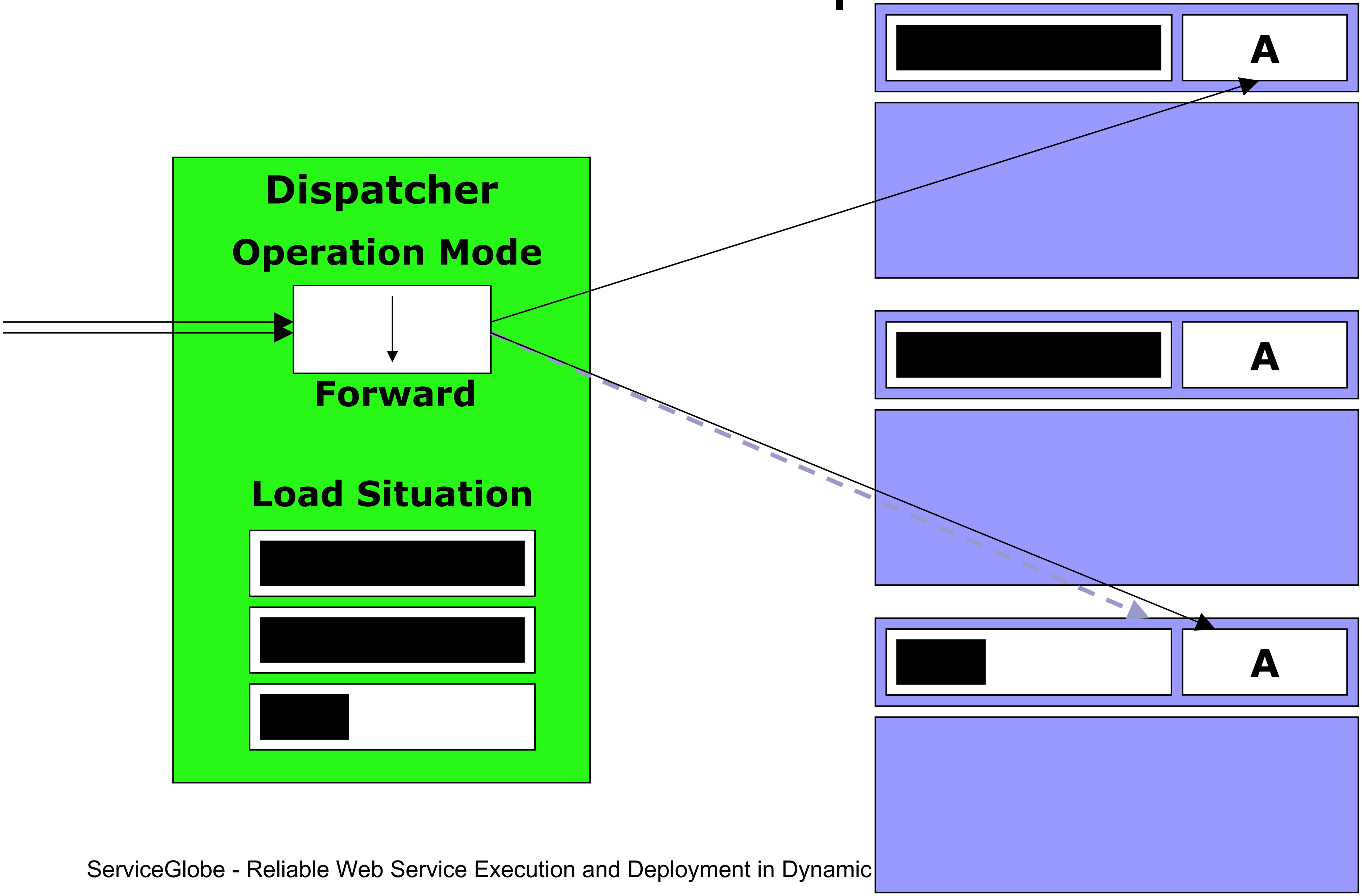
Advisor Modules



Survey of the Load Balancing System



Automatic Service Replication



Dispatcher

Operation Mode

Forward

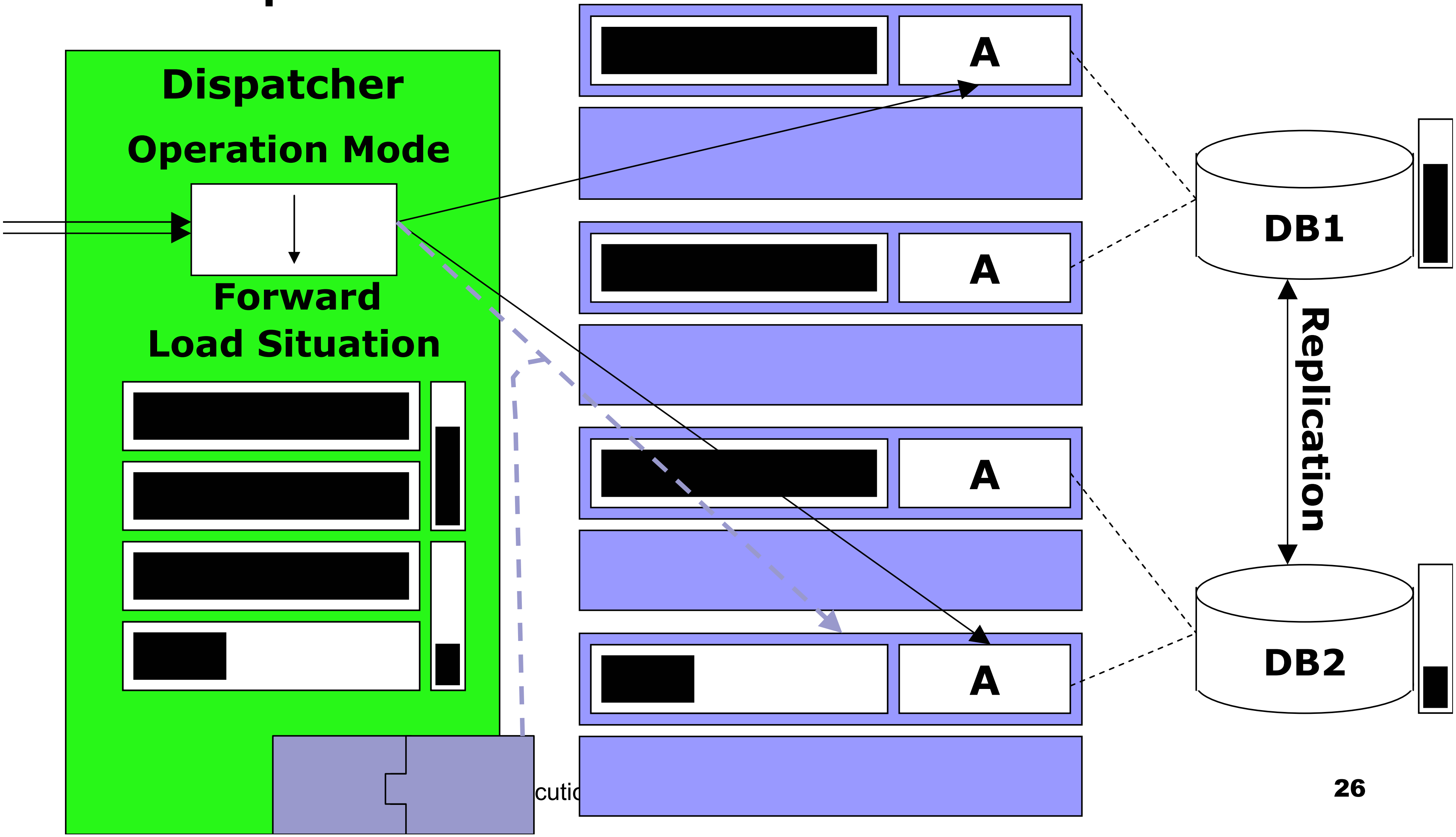
Load Situation

A

A

A

Automatic Service Replication in Complex Environments



Availability using a Dispatcher

$$\alpha = \alpha_{ServiceHost} = \frac{\mathrm{MTBF}}{\mathrm{MTBF} + \mathrm{MTTR}}$$

$$\alpha_{pool} = \sum_{i=1}^{N} \alpha^{i} (1-\alpha)^{(N-i)} = 1 - (1-\alpha)^{N}$$

- Example: MTBF = 48h, MTTR = 12h, 8 hosts

➔ 1.5 minutes unavailable a year

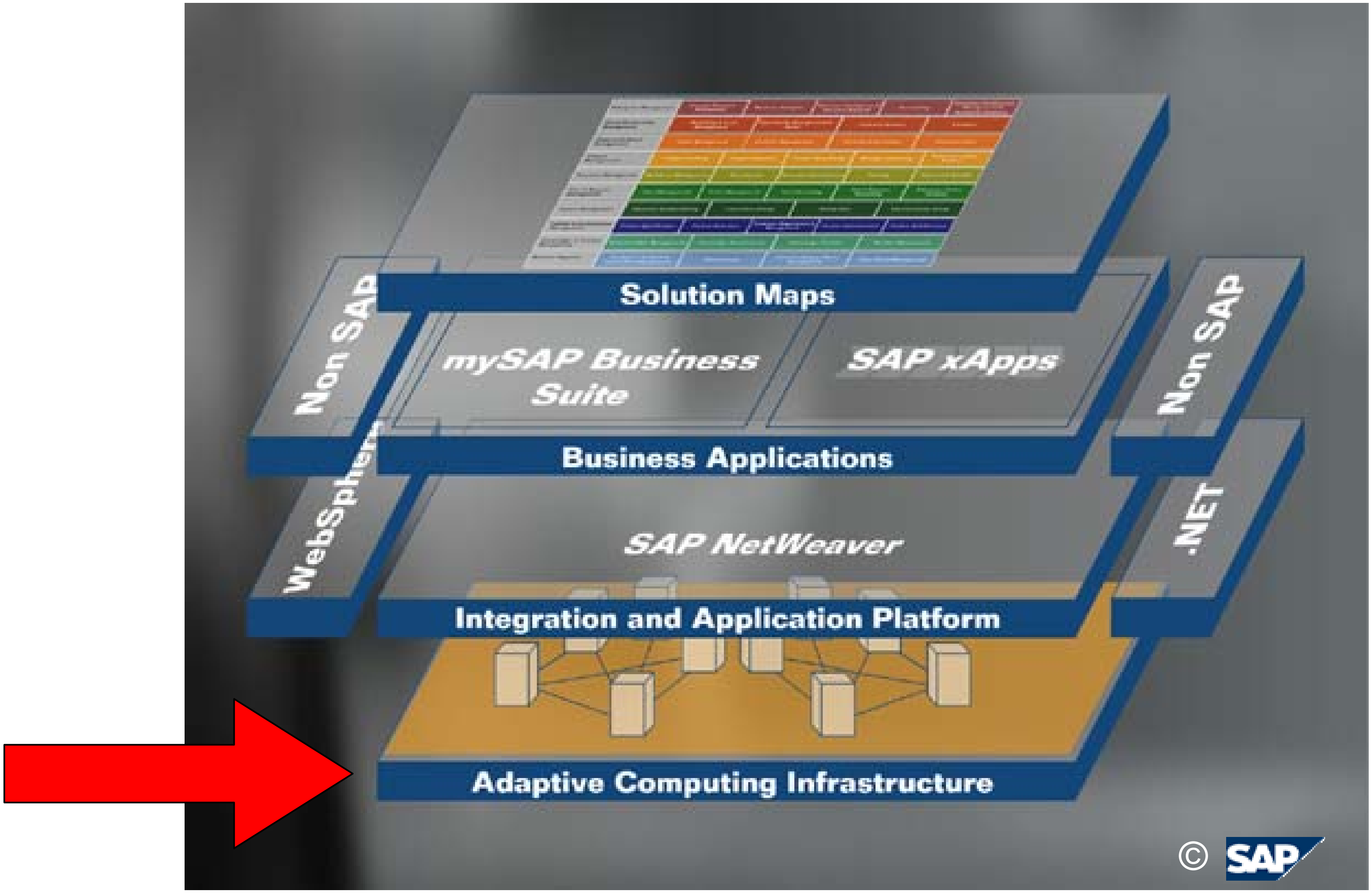
Outline

- Architecture of ServiceGlobe
- Dynamic Service Selection
- Load Balancing and Service Replication
- Current Utilization of the Techniques
- Conclusion

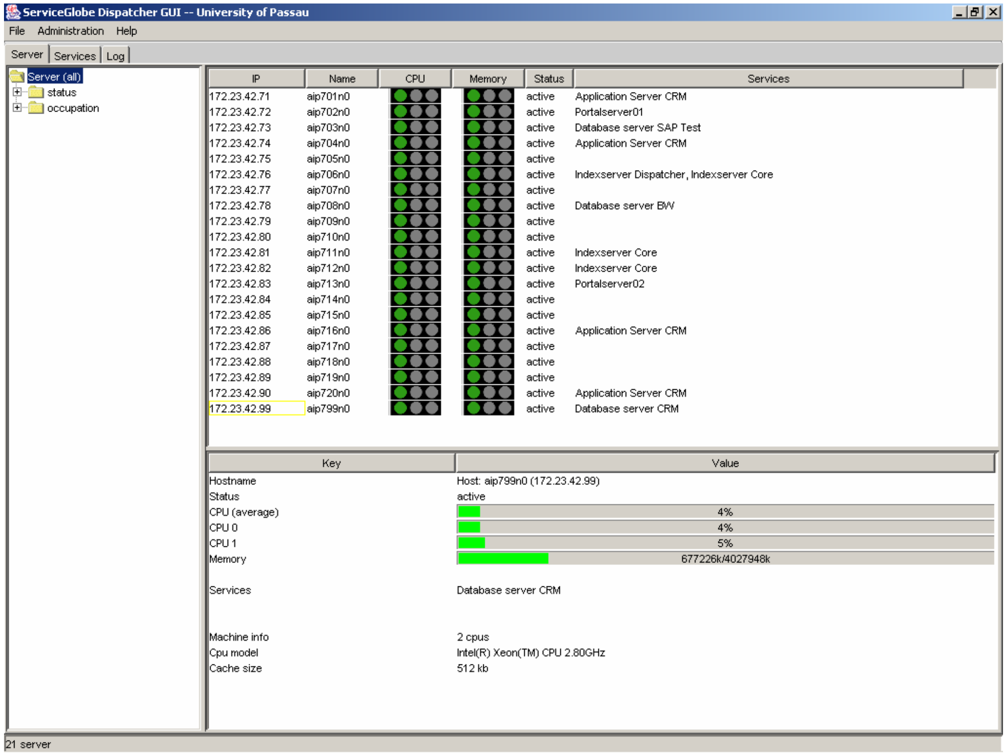
Current Utilization of the Techniques

- The presented technologies are currently integrated into SAP NetWeaver platform to supplement its service virtualization capabilities.
- A demonstration was shown at the SAPphire 2003, SAP's user conference.

SAP NetWeaver



Application Service Manager



IP	Name	CPU	Memory	Status	Services
172.23.42.71	aip701n0	🟢⚫⚫	🟢⚫⚫	active	Application Server CRM
172.23.42.72	aip702n0	🟢⚫⚫	🟢⚫⚫	active	Portalserver01
172.23.42.73	aip703n0	🟢⚫⚫	🟢⚫⚫	active	Database server SAP Test
172.23.42.74	aip704n0	🟢⚫⚫	🟢⚫⚫	active	Application Server CRM
172.23.42.75	aip705n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.76	aip706n0	🟢⚫⚫	🟢⚫⚫	active	Indexserver Dispatcher, Indexserver Core
172.23.42.77	aip707n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.78	aip708n0	🟢⚫⚫	🟢⚫⚫	active	Database server BW
172.23.42.79	aip709n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.80	aip710n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.81	aip711n0	🟢⚫⚫	🟢⚫⚫	active	Indexserver Core
172.23.42.82	aip712n0	🟢⚫⚫	🟢⚫⚫	active	Indexserver Core
172.23.42.83	aip713n0	🟢⚫⚫	🟢⚫⚫	active	Portalserver02
172.23.42.84	aip714n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.85	aip715n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.86	aip716n0	🟢⚫⚫	🟢⚫⚫	active	Application Server CRM
172.23.42.87	aip717n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.88	aip718n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.89	aip719n0	🟢⚫⚫	🟢⚫⚫	active	
172.23.42.90	aip720n0	🟢⚫⚫	🟢⚫⚫	active	Application Server CRM
172.23.42.99	aip799n0	🟢⚫⚫	🟢⚫⚫	active	Database server CRM

Key	Value
Hostname	Host: aip799n0 (172.23.42.99)
Status	active
CPU (average)	4%
CPU 0	4%
CPU 1	5%
Memory	677226k/4027948k
Services	Database server CRM
Machine info	2 cpus
Cpu model	Intel(R) Xeon(TM) CPU 2.80GHz
Cache size	512 kb

21 server

- Single Point of Control
- **Server** Landscape and Application Service Management
- Dynamic Demand driven Resource Assignment
- Change Management

Outline

- Architecture of ServiceGlobe
- Dynamic Service Selection
- Load Balancing and Service Replication
- Current Utilization of the Techniques
- Conclusion

Conclusion

- We presented techniques for reliable Web service execution and deployment
  - Dynamic Service Selection/Fusion and
  - Load Balancing and Service Replication
- in the Context of ServiceGlobe

Thank You for Your Attention!

… any questions???