

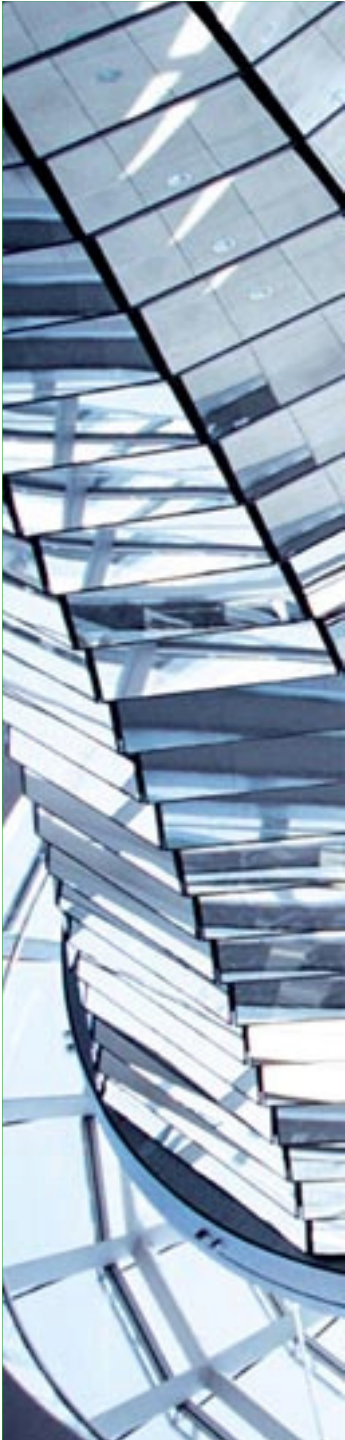
10. Service Oriented Architecture Reference Architectures and Patterns

Winter Semester 2008 / 2009
Prof. Dr. Bernhard Humm
Darmstadt University of Applied Sciences
Department of Computer Science



The lecture in the context of the entire course

1. Introduction
2. A reference architecture for business information systems
3. Application kernel
4. Persistence and transaction
5. Authorization
6. Client architecture
7. Exception handling
8. Business Intelligence
9. Systems integration
10. Service-oriented architecture
11. Selected design patterns
12. Design for testability



Agenda

→ SOA myths and reality

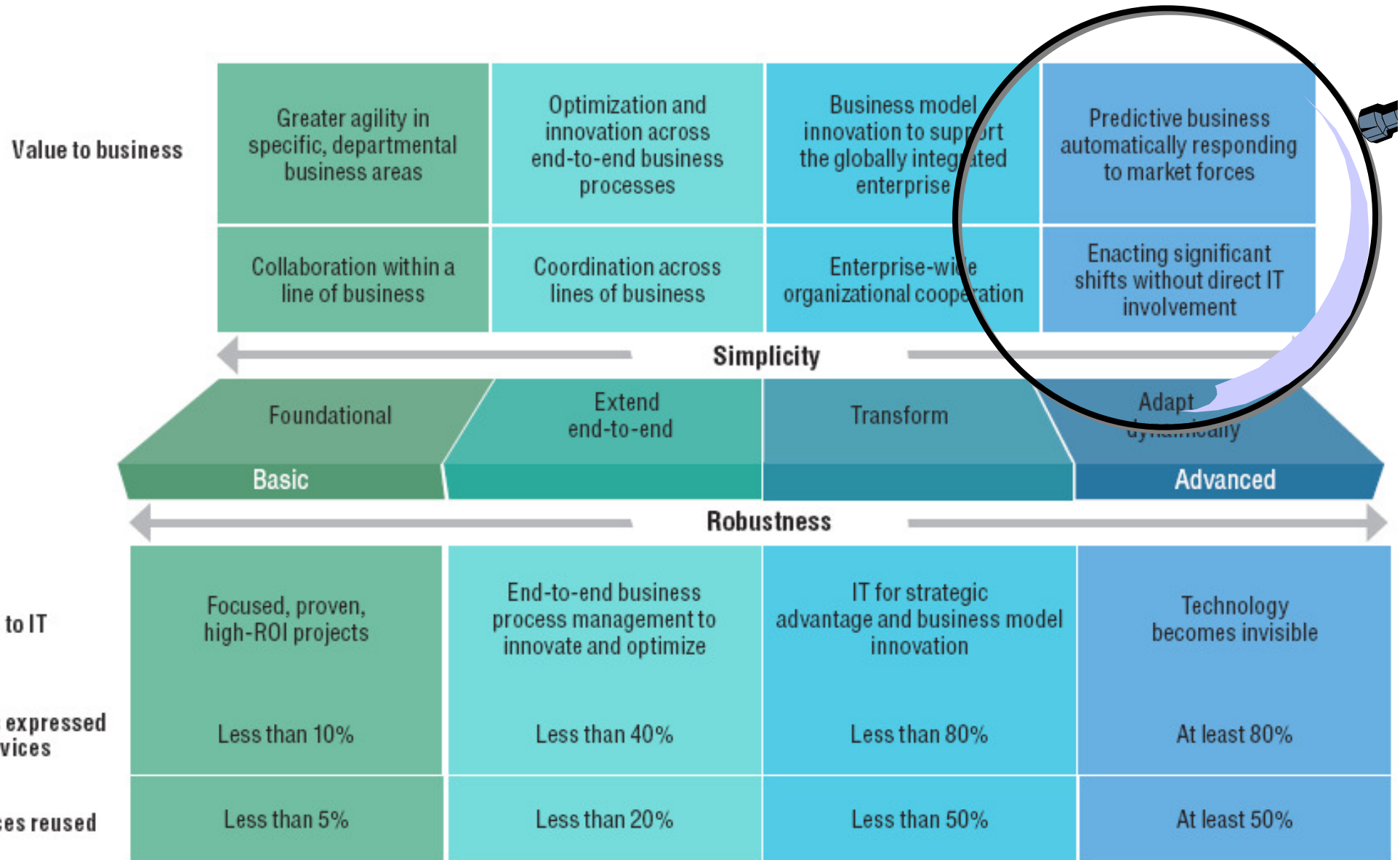
Cities and application landscapes

Reference architecture

Example

Technology

Literature



Source: IBM white paper: "Smart SOA: Best practices for agile innovation and optimization"

Software AG technology can increase your visibility across a global supply chain and extend services to your business partners using **B2B Integration**.

Run your organization more efficiently and cost-effectively. Build new, highly competitive processes, or improve current ones with our **Business Process Management (BPM)** software.

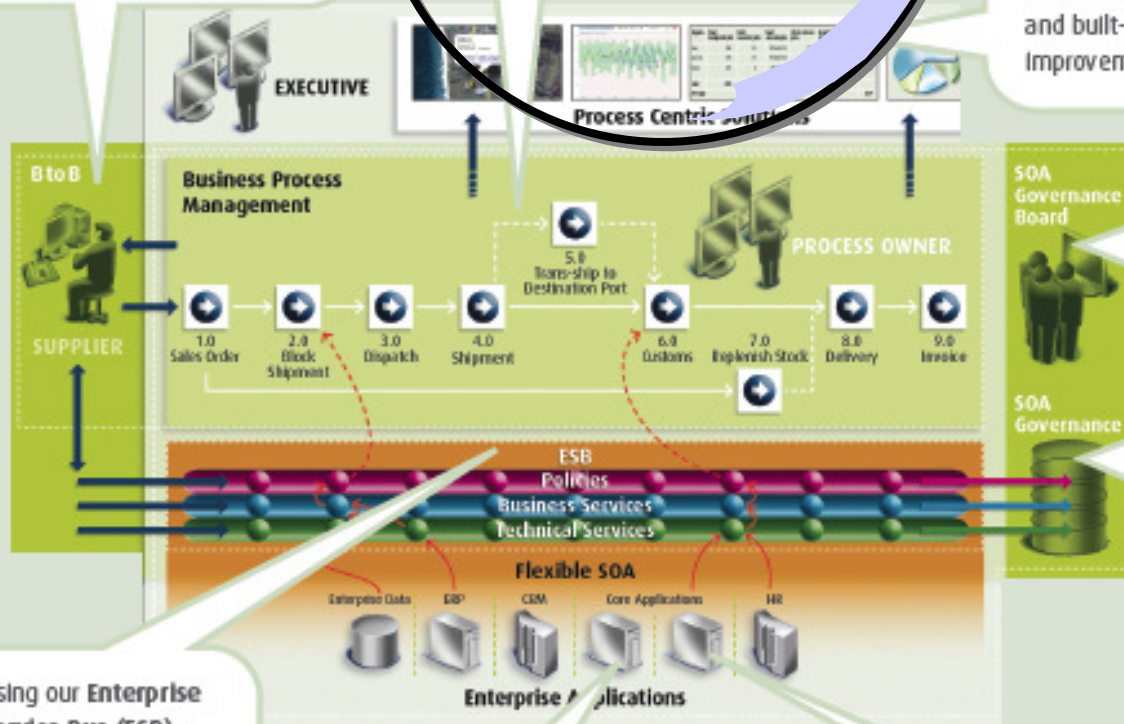
With **Business Activity Monitoring (BAM)**, you can accurately measure and improve business performance in real time. Customizable dashboards provide managers with unique views into activity, alerts to problems and built-in recommendations for improvement.

With **SOA Governance**, you can support rapid and frequent business change by easily managing, locating and re-using IT assets and enforcing policies that ensure consistently high quality.

Using our **Enterprise Service Bus (ESB)**, you can integrate your heterogeneous systems and extract features, data and services from them so they can be re-used easily in an SOA and composite applications.

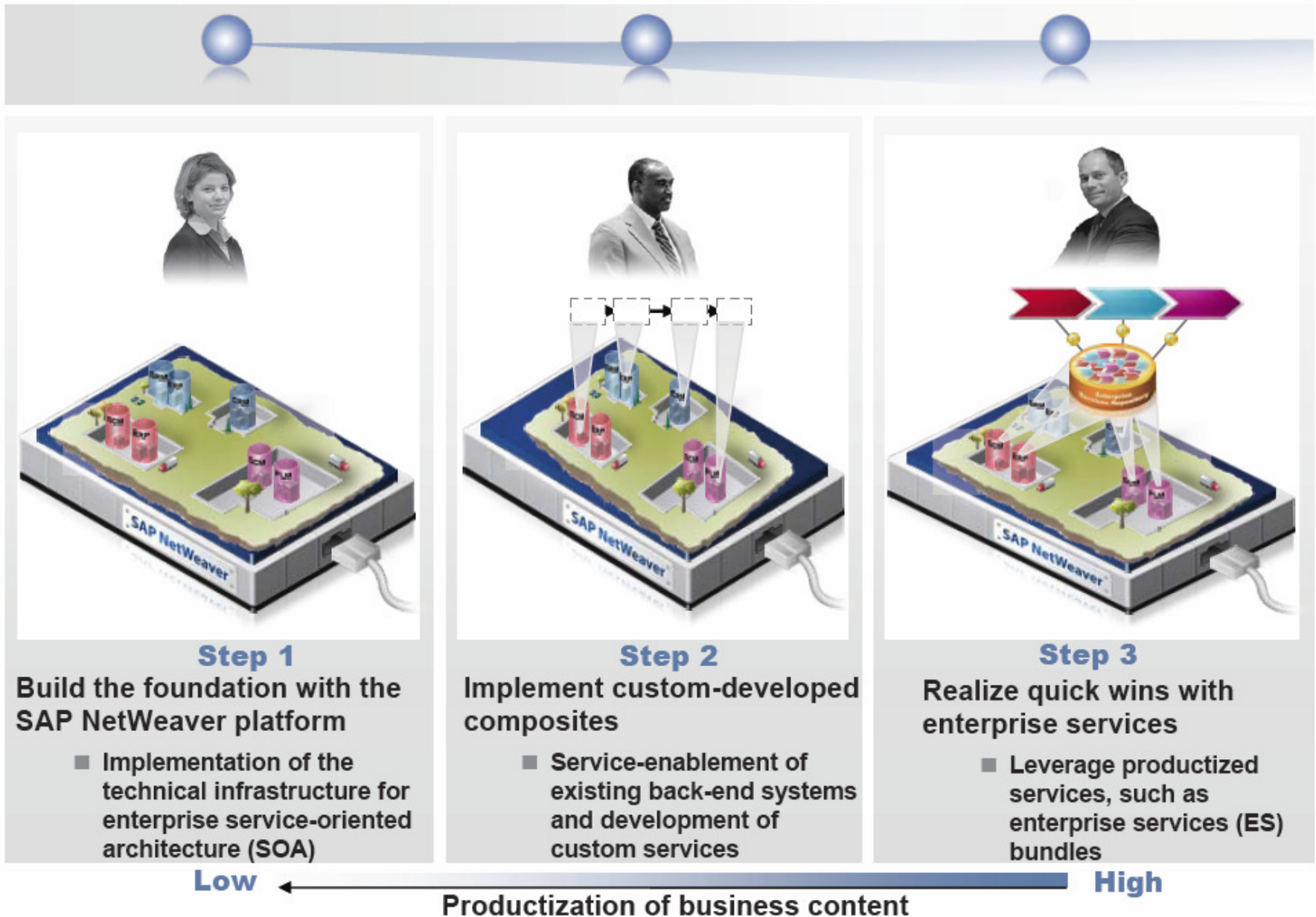
With our **Application Modernization Suite**, you can achieve rapid service enablement to extend core applications to an SOA and modern Web environments without extensive programming. You can also build new high-value business services that provide greater alignment between business and IT.

Using **Natural** and **Adabas**, you can rapidly develop and efficiently maintain mission-critical business applications. Natural delivers 50 percent productivity over alternatives such as Java or .NET. And with Adabas you'll achieve peak performance in even the most demanding high-transaction environments while benefiting from a low Total Cost of Ownership (TCO).

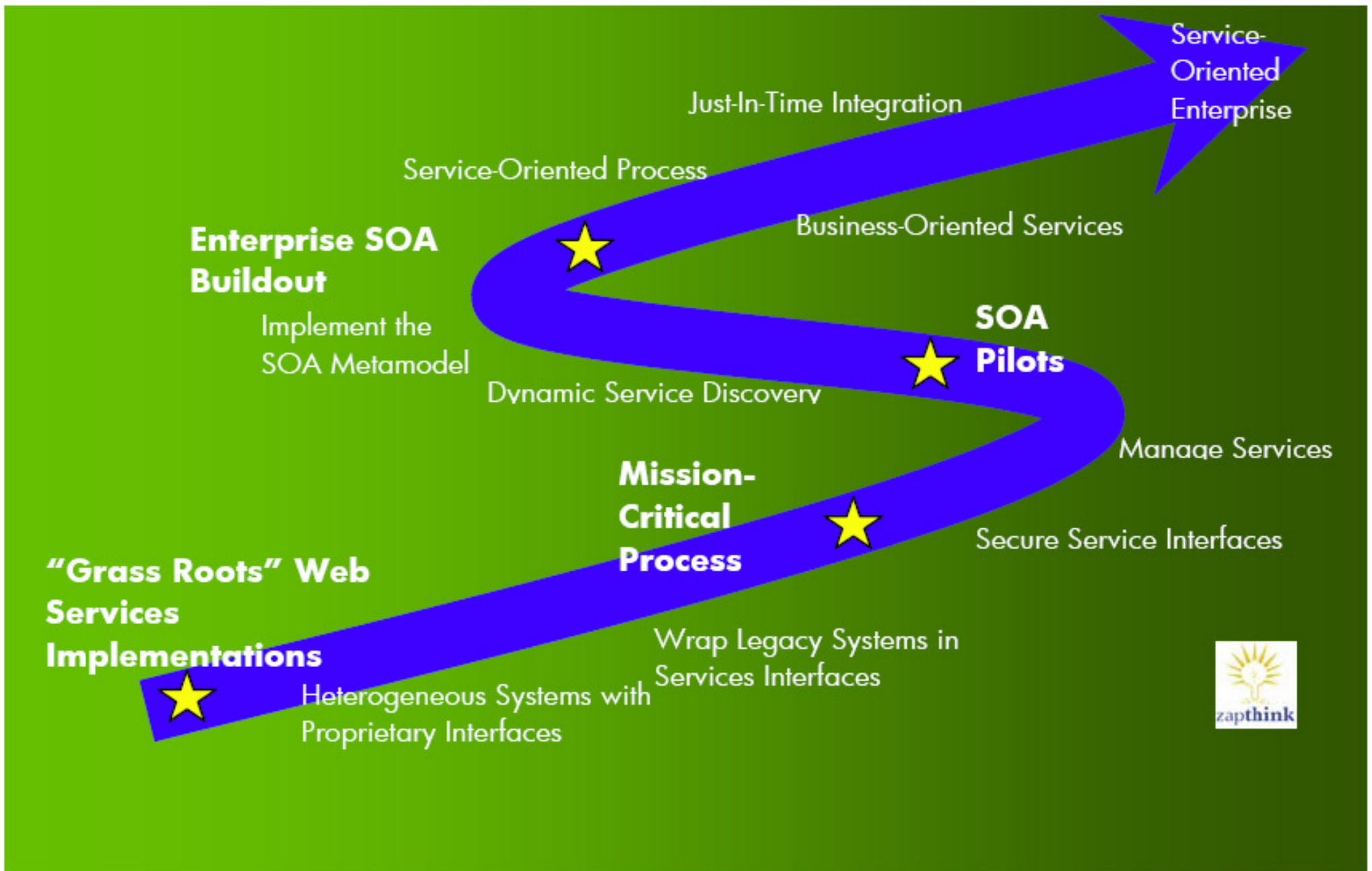


Source:
Software AG
white paper
"Business Infrastructure
Software"

The Technical Path to Enterprise Service-Oriented Architecture: Stepwise Adoption of New Technologies



Source: SAP „Enterprise Service-Oriented Architecture made Easy“



Source: HP white paper „Service Oriented Architecture“

Myths and reality

Myth	Example	Reality
<ul style="list-style-type: none"> ■ SOA is a technology (standards and products) 	<ul style="list-style-type: none"> ■ "If you're using Web services (such as SOAP over HTTP), you've achieved SOA" 	<ul style="list-style-type: none"> ■ SOA is paradigm for structuring the business of an enterprise which then drives the IT application landscape
<ul style="list-style-type: none"> ■ SOA is easy 	<ul style="list-style-type: none"> ■ "You can buy SOA out of the box" 	<ul style="list-style-type: none"> ■ Structuring application landscapes towards a true SOA takes a long time, needs high expertise and is expensive
<ul style="list-style-type: none"> ■ Technology drives business 	<ul style="list-style-type: none"> ■ "Business automatically responds to market forces, once SOA technology is implemented" 	<ul style="list-style-type: none"> ■ Business must drive technology ■ Technology can only <i>support</i> business
<ul style="list-style-type: none"> ■ Architecture means technical infrastructure 	<ul style="list-style-type: none"> ■ "Enterprise Service bus is the core of an SOA" 	<ul style="list-style-type: none"> ■ The structure of applications is the primary focus of SOA ■ Technical infrastructure is secondary



Agenda

SOA myths and reality

→ **Cities and application landscapes**

Reference architecture

Example

Technology

Literature

An aerial photograph of a dense urban skyline, likely New York City, showing numerous skyscrapers and buildings. A semi-transparent white box with a halftone pattern is overlaid at the top, containing the title. The sky is clear and blue.

A Megacity

www.t-migge.science-explorer.de



Agenda

SOA myths and reality

Cities and application landscapes

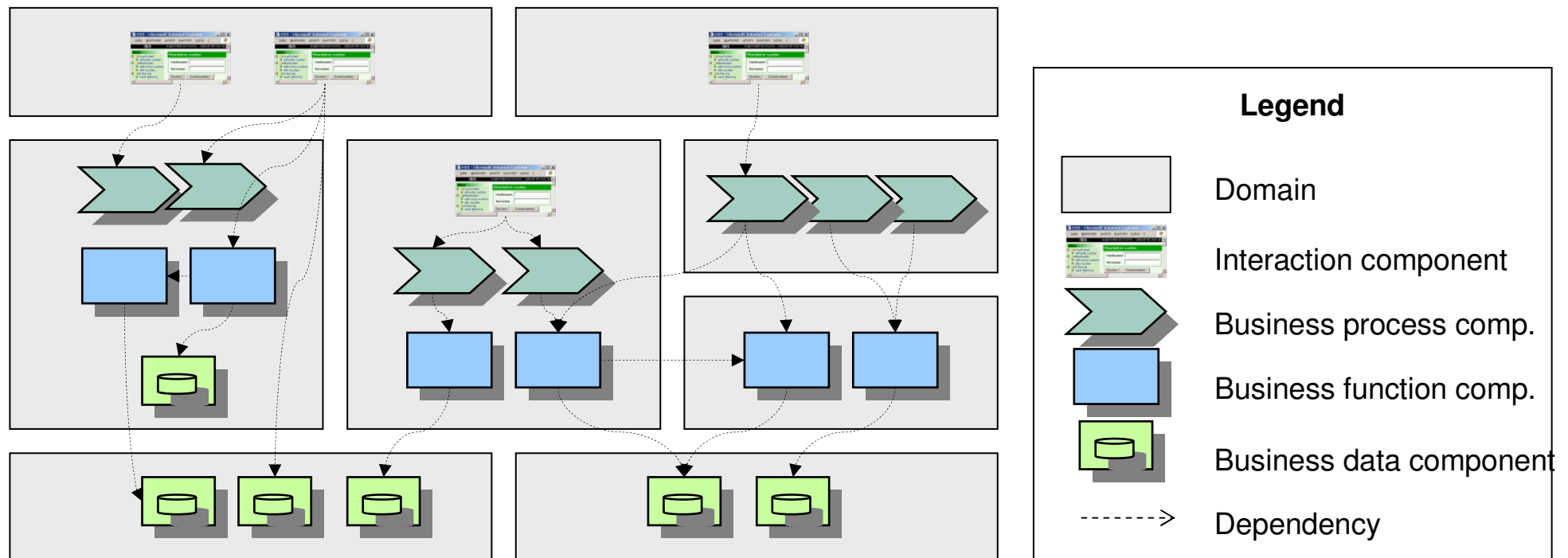
→ **Reference architecture**

Example

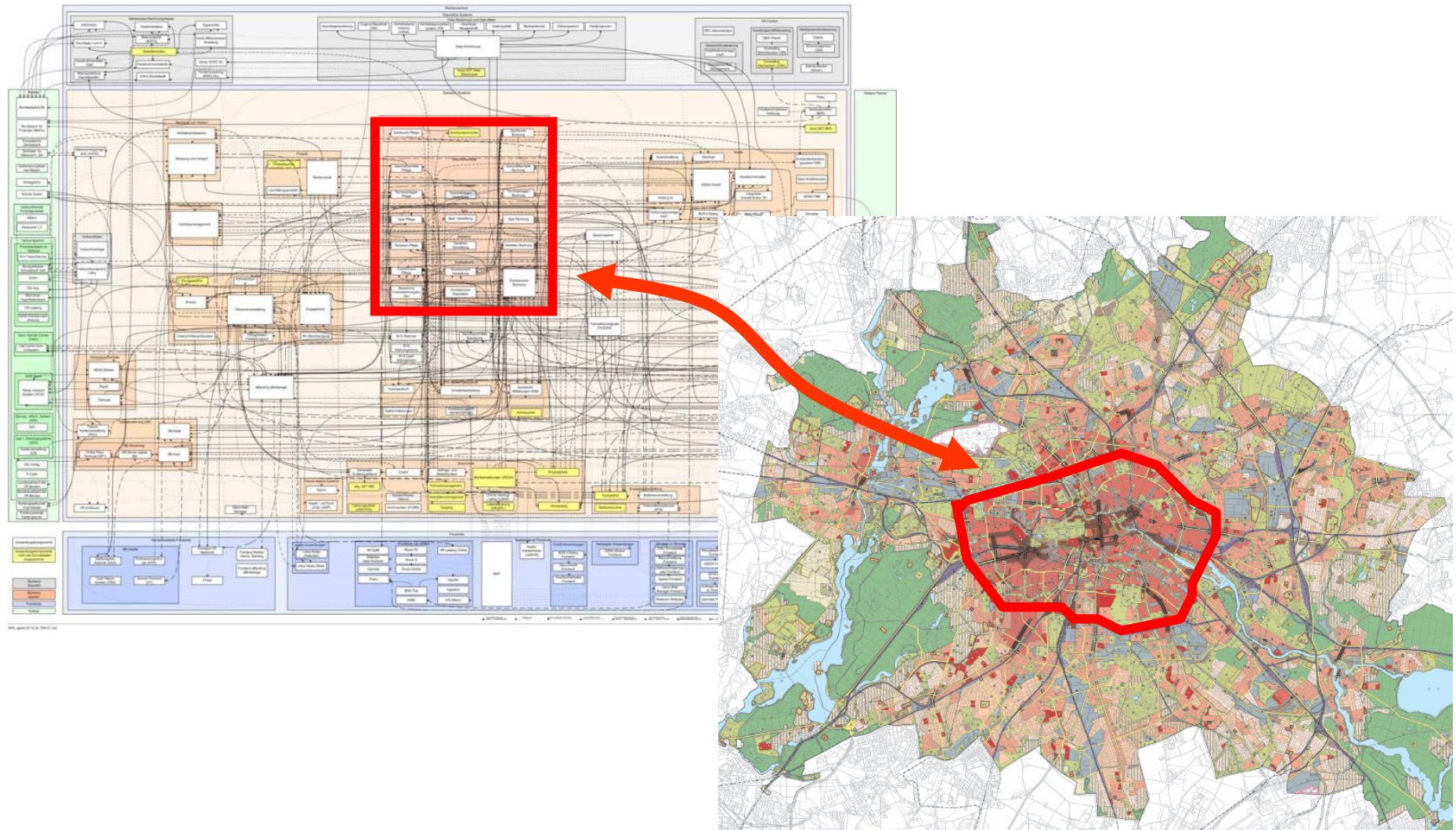
Technology

Literature

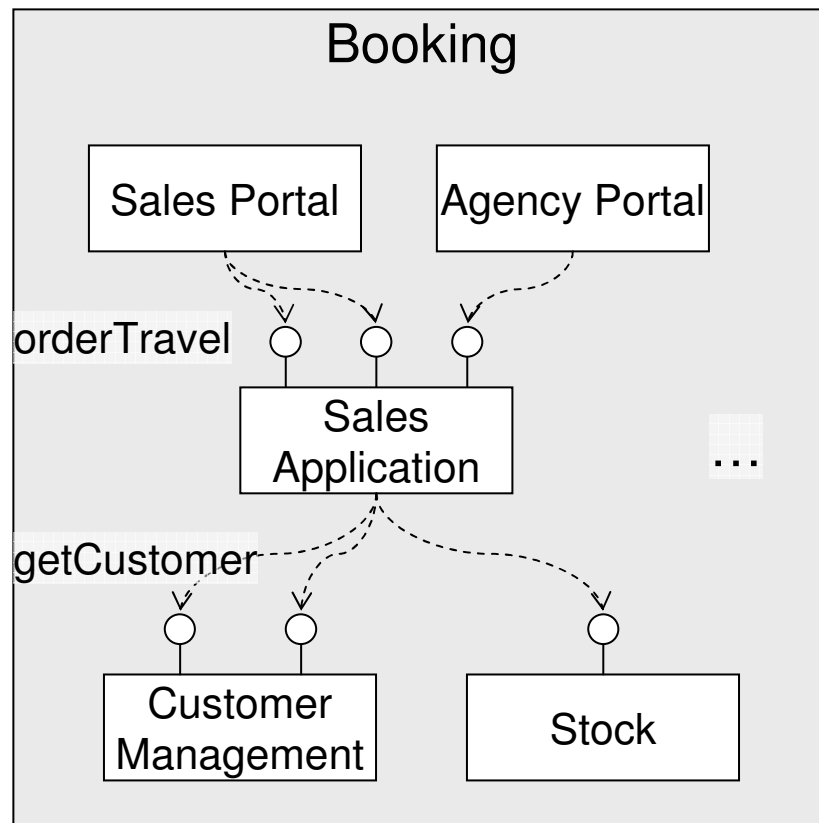
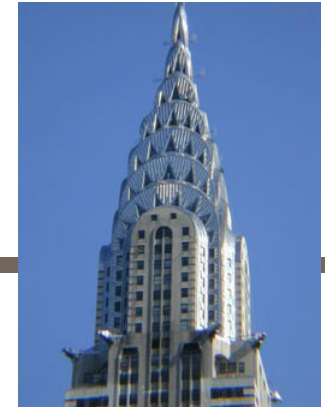
Reference architecture for application landscapes / Service Oriented Architecture



Domains = the suburbs of the software city (IT application landscape)

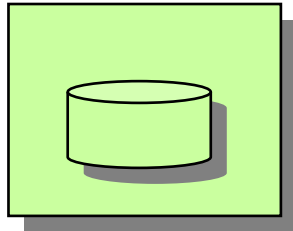


Enterprise application components = Buildings of the software city

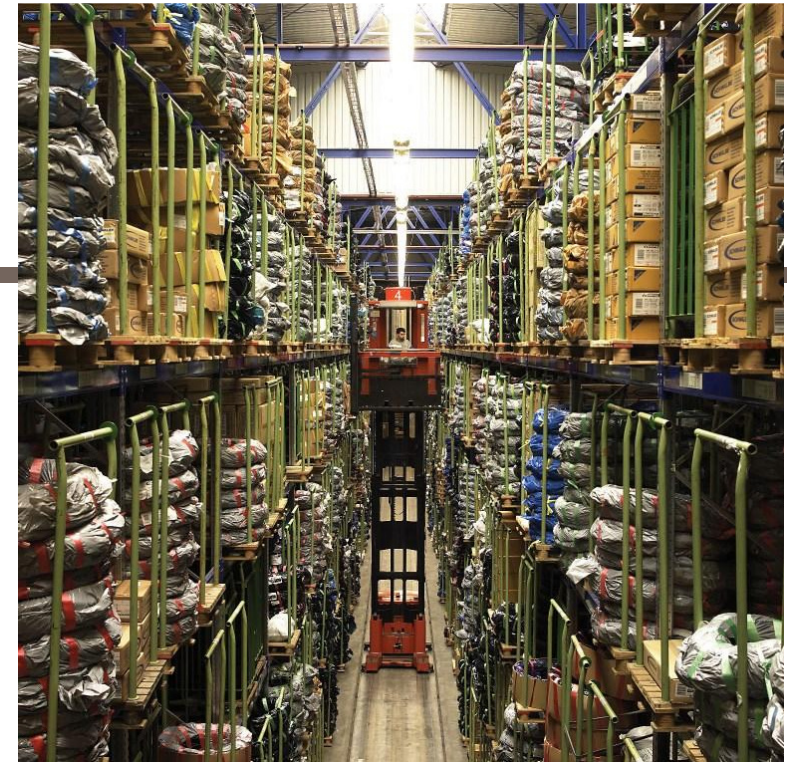


- Enterprise application components belong to domains
- Enterprise components should be internally constructed according to the reference architecture for business information systems.
But: those internals are not relevant for SOA
- Components export interfaces they provide
- Components import interfaces they require
- Components can be composed
- Different categories of enterprise application components: data, function, process, and interaction

Business data components = Storehouses in the software city



- Control business objects that are managed by applications
- Provide CRUD operations: Create, Read, Update, Delete
- Provide consistency rules
- Provide higher-level operations for writing data, e.g., data history
- Provide various views on data (reading)
- Examples: CustomerManagement, ProductManagement, ...



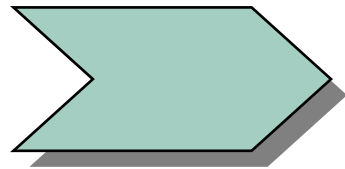
Business function components = Factories in the software city



- Implement complex business logic
- Usually the largest components in an enterprise architecture
- Use business data components
- Examples: Pricing, Forecasting, CapacityControl, Ranking, ...



Business process components = Control centres in the software city



- Implement business processes: process control
- Do not implement complex business logic but use business function components instead
- Use business data components
- Are often implemented using workflow management systems
- BUT: A workflow management system is part of the technical infrastructure, NOT an enterprise component



Interaction components = Shopping malls of the software city

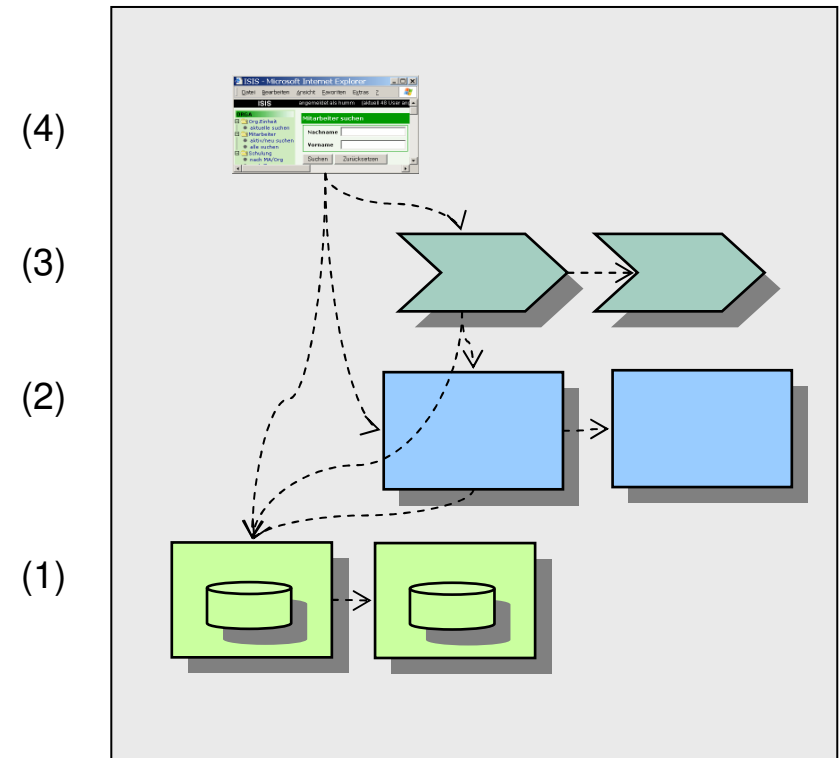


- Allow users to interact with enterprise components
- Often: Internet or Intranet portals
- Uniform layout
- Single sign on
- Examples: SalesPortal, EmployeePortal, ...



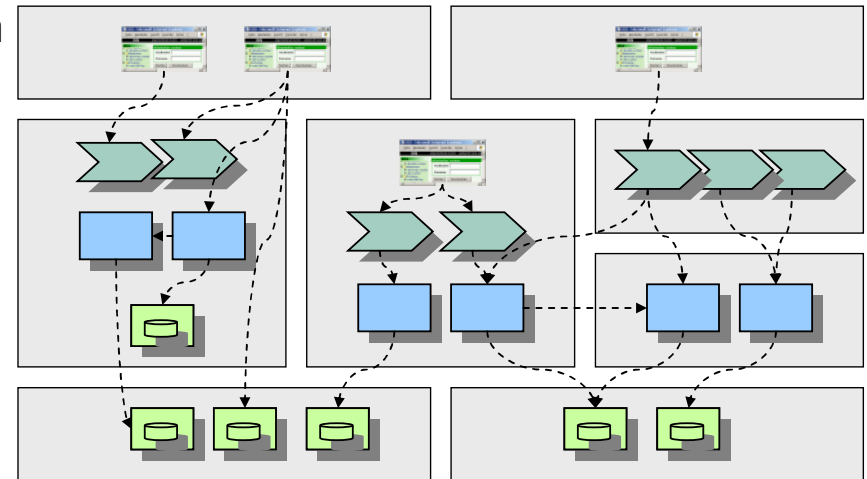
Dependencies

- Dependency: knows / calls / receives data from
- Components of layer n may only have dependencies to components of layer $\leq n$
(Layer 1 = business data component ...
Layer 4 = interaction component)
- i.e.:
 - Business data components (1) may have dependencies to other business data components (1) only
 - Business function components (2) may have dependencies to other business function (2) and business data components (1) only
 - Business process components (3) may have dependencies to other business process components (3), business function components (2) and business data components (1) only
 - Interaction components (4) may have dependencies to components of all categories (1-4)



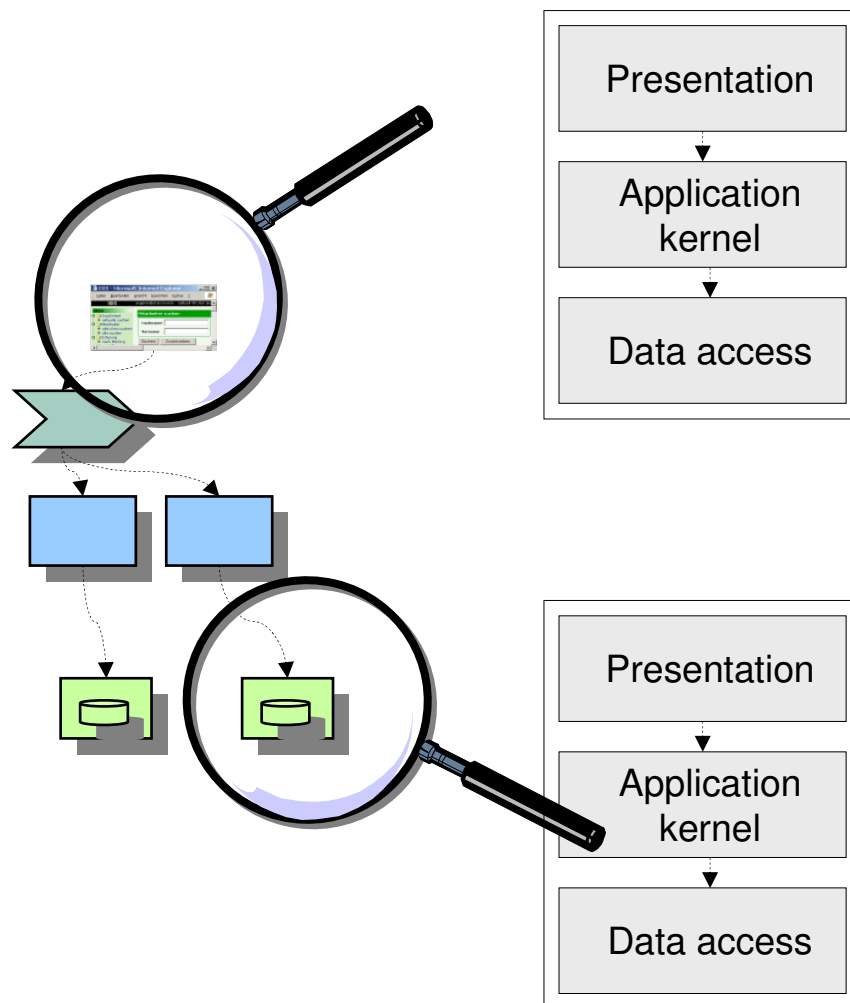
Key statements of the reference architecture

1. Every component belongs to exactly one domain
2. Every component is of exactly one category: data, function, process, interaction
3. Dependencies follow a layering of the components according to their categories



- In practise, IT application landscapes rarely comply completely to the reference architecture.
- However, the reference architecture is a guideline for developing the IT application landscape

Component categories are NOT 3-Layer-Architecture (reference architecture of business information systems)!



- Components of every category may consist of three layers internally
- Interaction component
 - Presentation: Portal
 - Application kernel: dialog control
 - Data access: content management
- Business data component
 - Presentation: Dialogs for data management
 - Application kernel: consistency rules
 - Data access: data proper
- Analogous for business process components and business function components



Agenda

SOA myths and reality

Cities and application landscapes

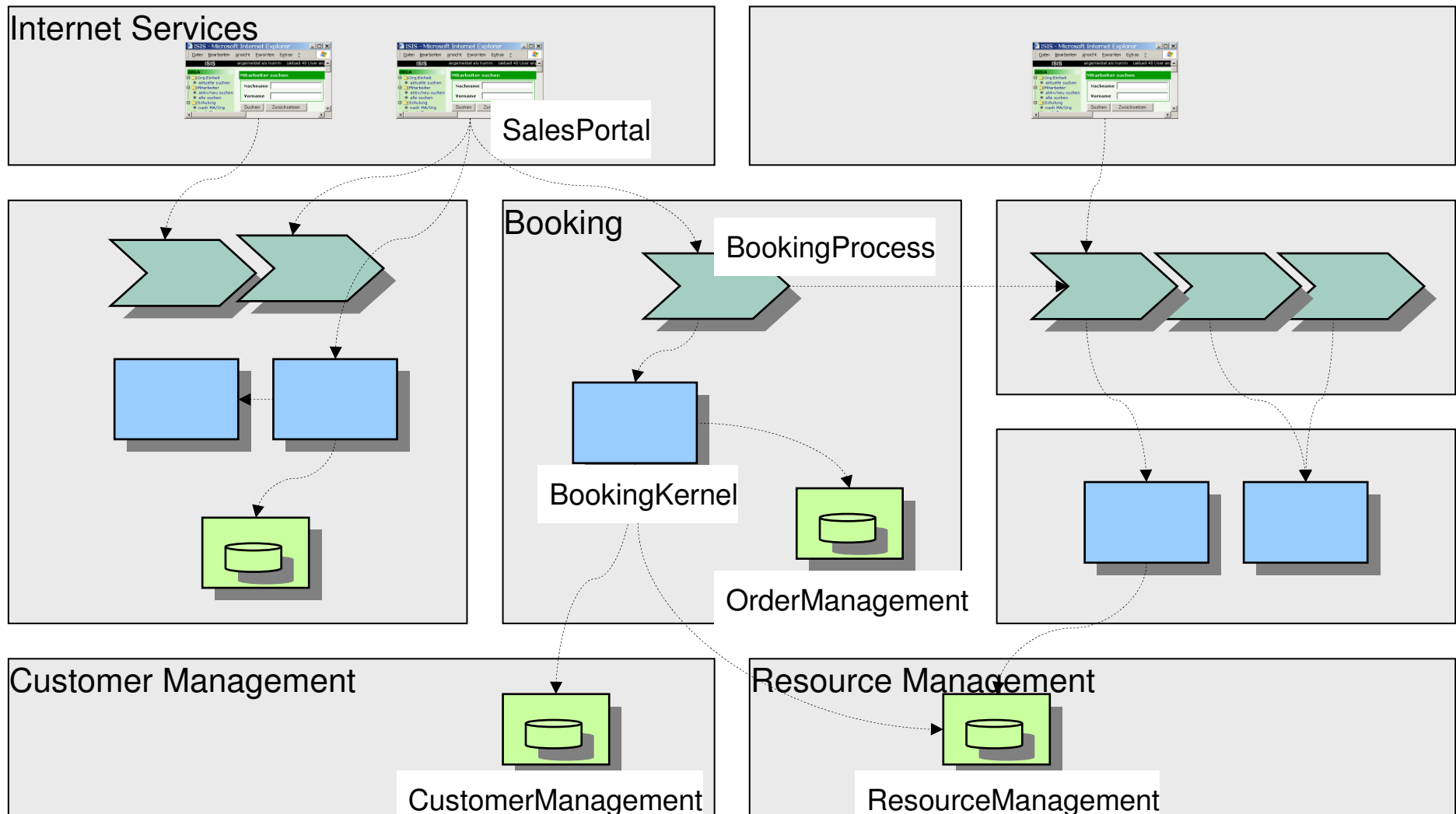
→ **Example**

Example

Technology

Literature

Example: Components for travel booking via the Internet





Agenda

SOA myths and reality

Cities and application landscapes

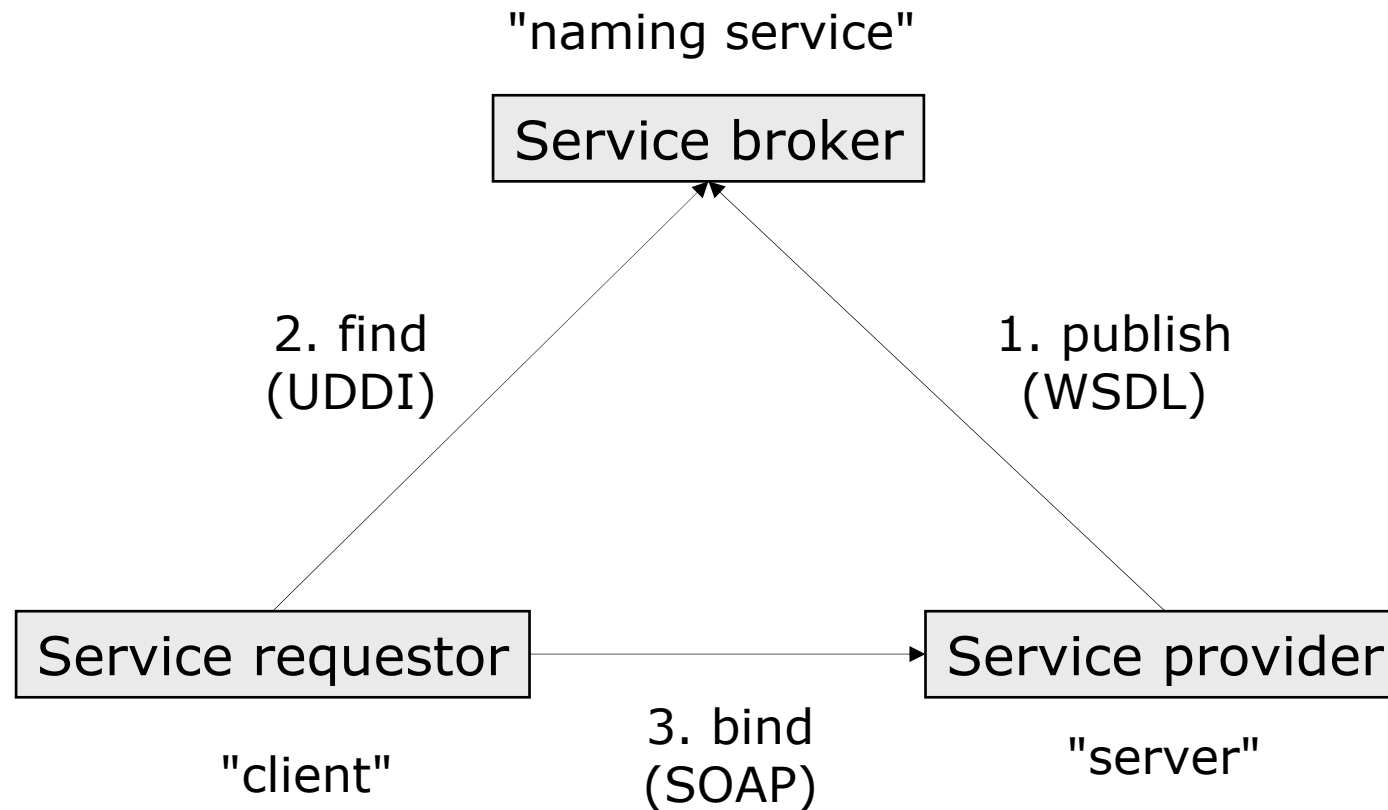
Reference architecture

Example

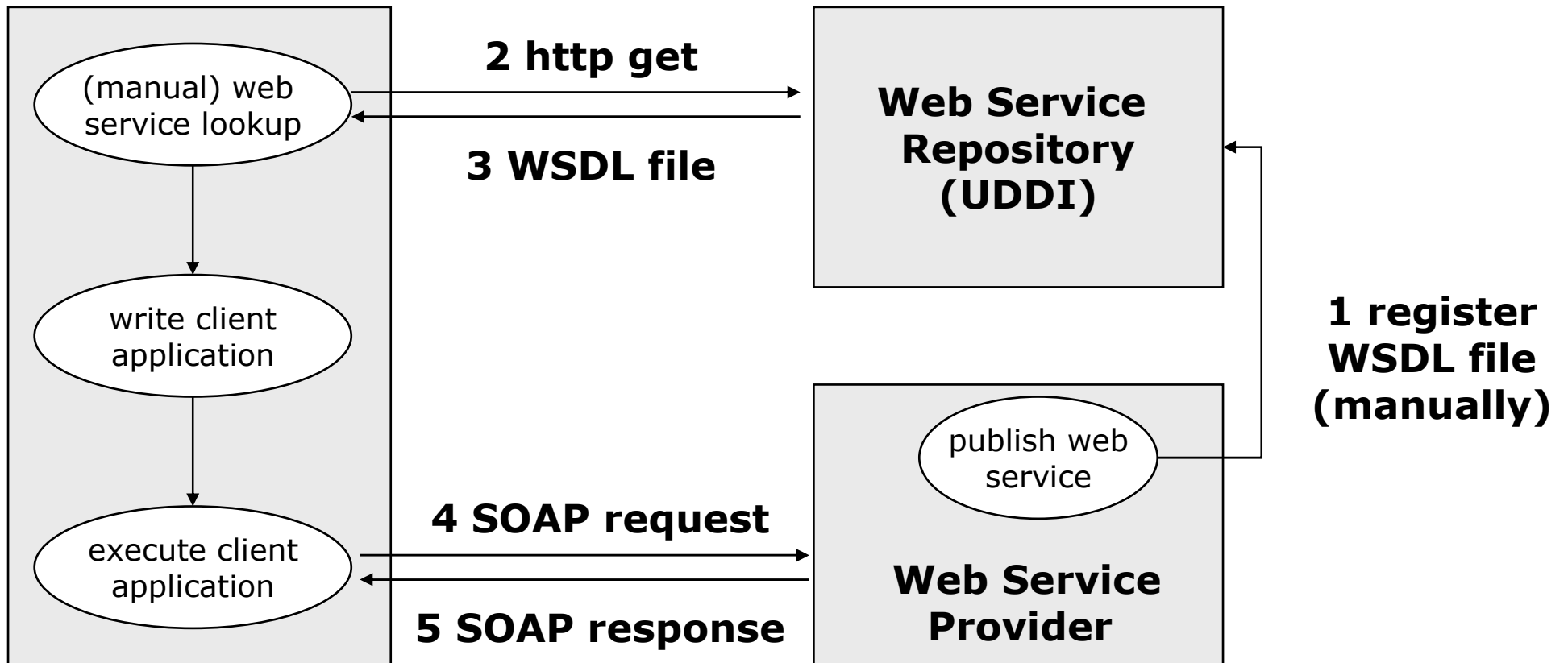
→ **Technology**

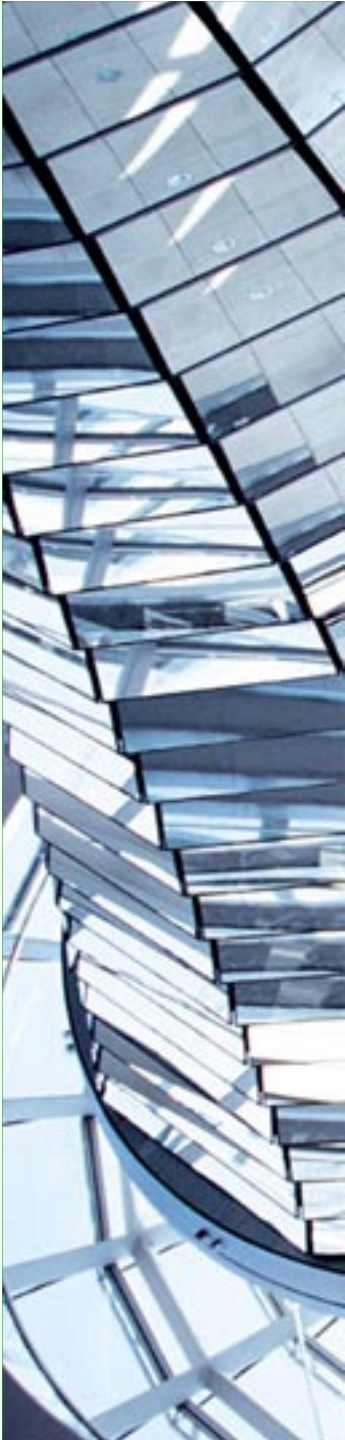
Literature

Web Services Technology: Naming Service



Basic Web Service Usage Scenario





Agenda

SOA myths and reality

Cities and application landscapes

Reference architecture

Example

Technology

→ **Literature**

Literature (download from my homepage)

- Andreas Hess, Bernhard Humm, Markus Voß:
Regeln für serviceorientierte Architekturen hoher Qualität.
Hauptbeitrag Informatik Spektrum 6/2006, Springer Verlag.
December 2006.
- Andreas Hess, Bernhard Humm, Markus Voß, Gregor Engels: *Structuring Software Cities - A Multidimensional Approach.* Proceedings of the 11th IEEE International EDOC Enterprise Computing Conference, Annapolis, Maryland, USA, pp. 122-129. IEEE Press, 2007.
- Markus Voß, Andreas Hess, Bernhard Humm: *Towards a Framework for Large Scale Quality Architecture.* In: Hofmeister, Ch., et.al. (Eds.): *Perspectives in Software Quality - Short Papers of the 2nd International Conference on the Quality of Software Architectures (QoSA)*, Interner Bericht 2006-10, Universität Karlsruhe, Fakultät für Informatik, ISSN 1432-7864, 2006.
- Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: *Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten.* dpunkt-Verlag 2008.
- Bernhard Humm: Was ist eigentlich ein Service? GI-Workshop Software Architektur, Mai 2008.

