SOA with Web Services in Practice

Nicolai M. Josuttis
IT-communication.com

03/09

Nicolai Josuttis

- **Independent consultant**
  - continuously learning since 1962
- **Systems Architect, Technical Manager**
  - finance, manufacturing, automobile, telecommunication
- **„SOA“ experience for multiple years**
  - Focus: bringing SOA into operation
    - Telco:
      - >50 international systems
      - >500 services
      - >10 million service calls per day
    - Automotive:
      - Business SOA (top-down)
      - Project SOA (bottom-up)

  …
- **http://www.soa-in-practice.com**

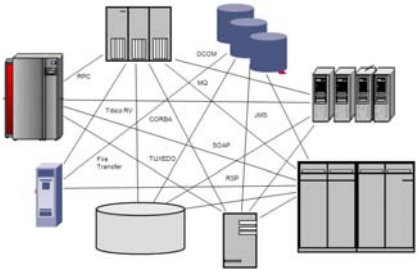
SOA

- **Service Oriented Architecture:**
  - An architectural **paradigm**
    - way of thinking, style, value system
    - e.g.: „*use lose coupling where appropriate*“
  - for **distributed processes**
    - multiple companies, business units, departments, systems involved
  - over **heterogeneous systems**
    - multiple platforms, programming languages, ...
  - that may be under control of **different owners**
    - different schedules, budgets, priorities, ...

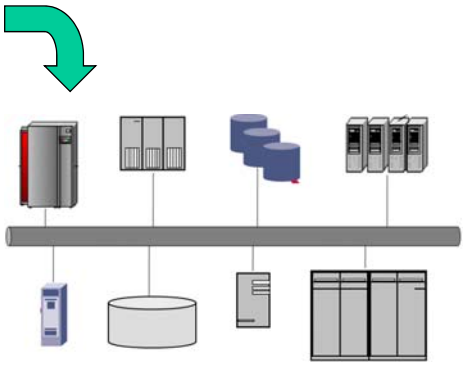
SOA Key Concepts

- **Loose Coupling**
  - Decouple dependencies to gain flexibility and fault tolerance
- **Services**
  - Integrate business‘ language into IT
- **High interoperability**
  - Make it easy to connect and reorganize systems and capabilities

Motivation for a Service Bus



- up to 25 individual connections
- no central management

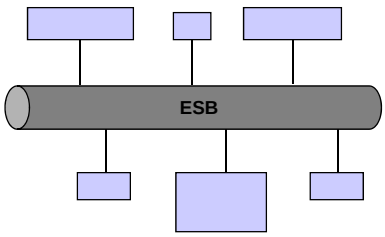


- 8 connections to the bus
- central management possible

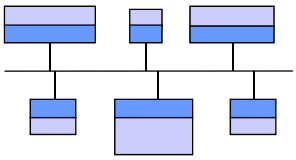
Enterprise Service Bus (ESB)

- **Infrastructure**
  - to provide high interoperability
  - in a (heterogeneous) distributed environment

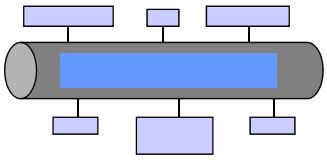
- **Tasks:**
  - mandatory:
    - Providing connectivity
    - Data transformation
    - (Intelligent) routing
    - Monitoring and logging
  - optional (“value-added services”):
    - Dealing with security
    - Dealing with reliability
    - Service management
    - …



ESB Approaches



– Stupid network
– Providers and consumers care for „intelligence“



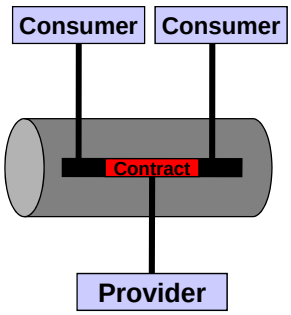
– Intelligent „network“
– Providers and consumers with simple adapters

- **Different ESB approaches for:**
  – Level of decoupling, (intelligent) routing,
  – Logging, monitoring, security, reliability, …
  – Message exchange patterns (synchronous or asynchronous)

  which has different technical and organizational consequences

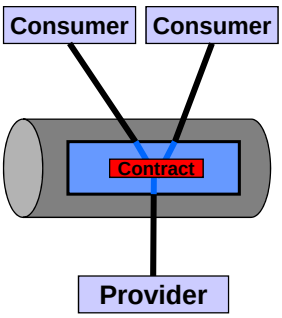
ESB Approaches in Practice

just a common protocol:

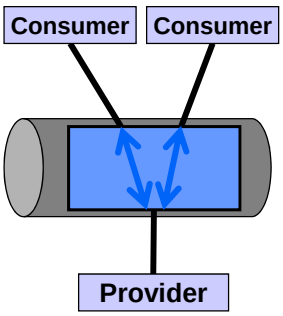


native Web Services

contracts with value-added services:



individual integration of interfaces:



possible EAI approach

Web Services

Web Services

- **SOA infrastructure using Internet technology**
  - Web Servers provide services
  - XML as interface language
- **Driven by Microsoft and IBM and others**
- **Core Standards:**
  - **WSDL** (Web Services Description Language)
    - as service interface description
  - **SOAP** (~~Simple Object Access~~ Protocol)
    - as message format
  - **UDDI** (Universal Description, Discovery and Integration)
    - as registry/broker („yellow pages“)

WSDL and SOAP (Extract)

WSDL

```
<types>
 <element name="getAddress">
  <complexType><sequence>
   <element name="customerID" type="long"/>
  </sequence></complexType>
 </element>
  …
</types>

<message name="getAddressInput">
 <part name="params" element="getAddress"/>
</message>
<message name="getAddressOutput">
  <part name="params" element="getAddressResponse"/>
</message>
…

<portType name="CustomerInterface" >
  <operation name="getAddress">
    <input message="getAddressInput"/>
    <output message="getAddressOutput"/>
    <fault message="customerNotFound"/>
  </operation>
</portType>
…
```

SOAP Request

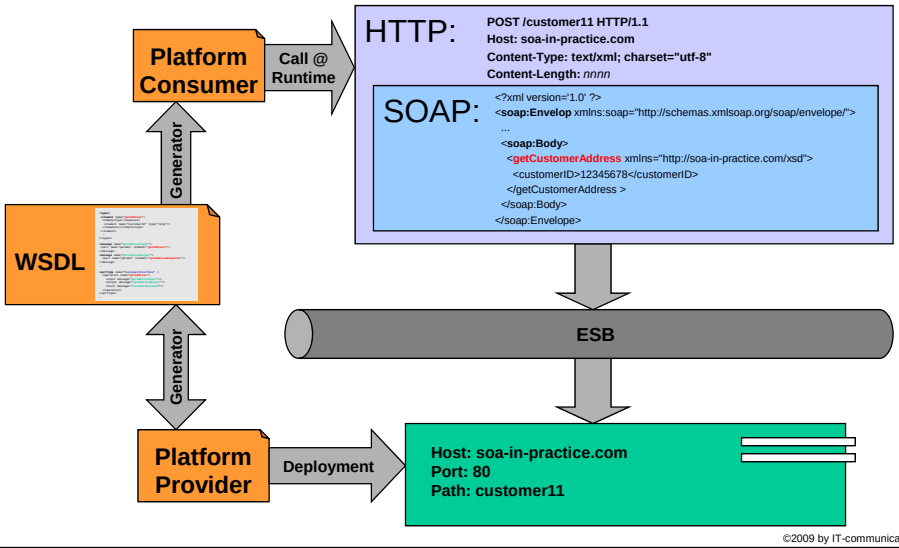
```
<Body>
  <getAddress>
    <customerID>12345678</customerID>
  </getAddress>
<Body>
```

SOAP Response

```
<Body>
  <getAddressResponse>
    <address>
      <street>Gaussstr. 29</street>
      <city>Braunschweig</city>
      <zipCode>D-38106</zipCode>
    </address>
  </getAddressResponse>
<Body>
```

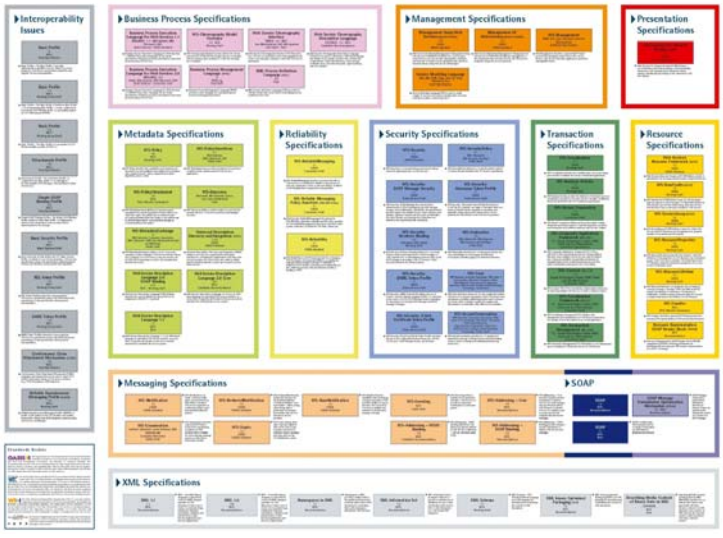
+ WSDL <binding>
+ WSDL <service>
+ Namespaces

Web-Services Overview



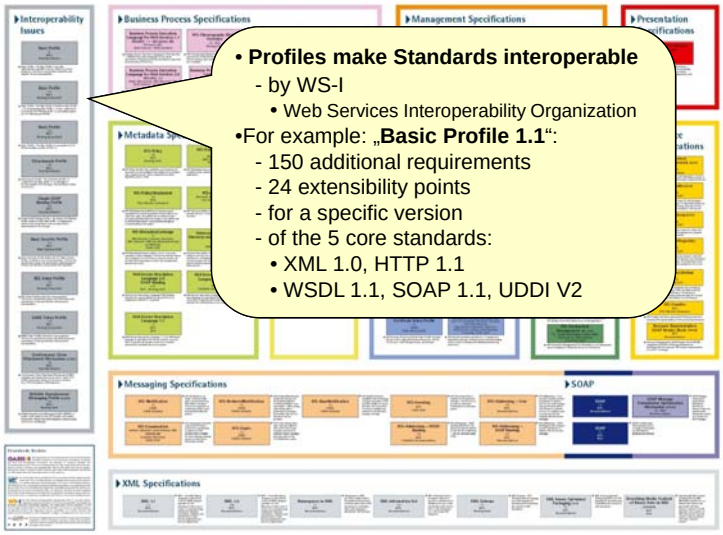
Web Services in Practice

Web Services Standards



see http://www.innoq.com/resources/ws-standards-poster/

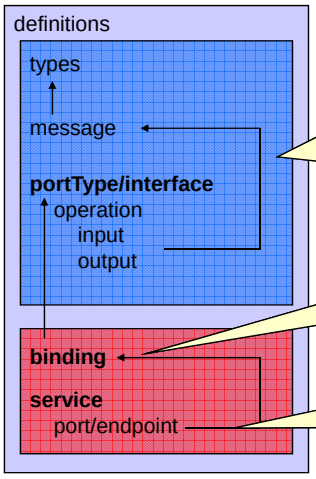
Web Services Standards



- **Profiles make Standards interoperable**
  - by WS-I
    - Web Services Interoperability Organization
- For example: „**Basic Profile 1.1**“:
  - 150 additional requirements
  - 24 extensibility points
  - for a specific version
  - of the 5 core standards:
    - XML 1.0, HTTP 1.1
    - WSDL 1.1, SOAP 1.1, UDDI V2

see http://www.innoq.com/resources/ws-standards-poster/

Native WSDL



- **“What”**
- **Technical Interface**
  - **Operations**
  - **Data-Types**
- **May be partially specified in multiple files**
  - **e.g data types in xsd files**

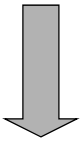
- **“How”**
- **Technical Protocol**
  - **e.g. SOAP with “document” style**

- **“Where”**
- **Hardwired IP Address**
  - **e.g. http://josuttis.de/myservice**

WSDL Lifecycle

1. **Design team specifies technical interface**
   - operations
   - data types
   - exceptions
2. **Infrastructure team specifies binding (protocol details)**
   - e.g. SOAP “document literal/wrapped”
3. **Operation team specifies locations/endpoints**
   - Test&Integration
   - QA
   - Production

“abstract” WSDL file:



“concrete” WSDL file:

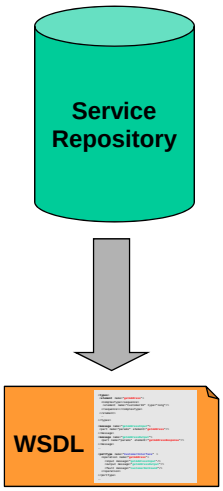
Service Contract

- **Description of a service**
- **more or less formal**
- **usually between**
  - a specific service **provider** and
  - a specific service **consumer**

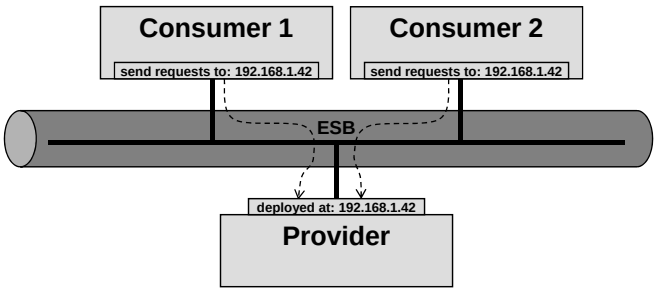
- **Includes:**
  - Description what a service does (the **semantics**)
  - Description how to use a service (the **interface**)
  - The quality of a service (**service level agreement, SLA**)
    - Availability (e.g. “24x7”)
    - Costs and limits (e.g. “up to 1.000 usages per day”)
    - Performance (e.g. “80% of results in 0.5 seconds”)

WSDL and Service Contracts

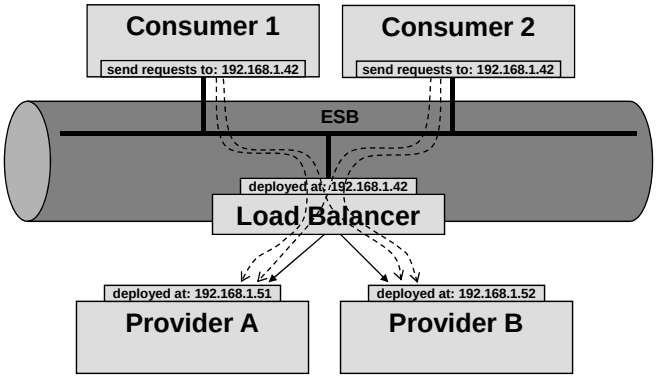
- **WSDL files are inherently limited**
  - No semantic information
  - No service level agreements (SLAs)
  - Format modifications
    - WSDL 1.1 and WSDL 2.0 are different
- **There are workaround/extensions**
  - Comments
  - WS policies
- **More sustainable:**
  - Specify services independent from infrastructure (in repositories)
  - Generate WSDL files out of repositories



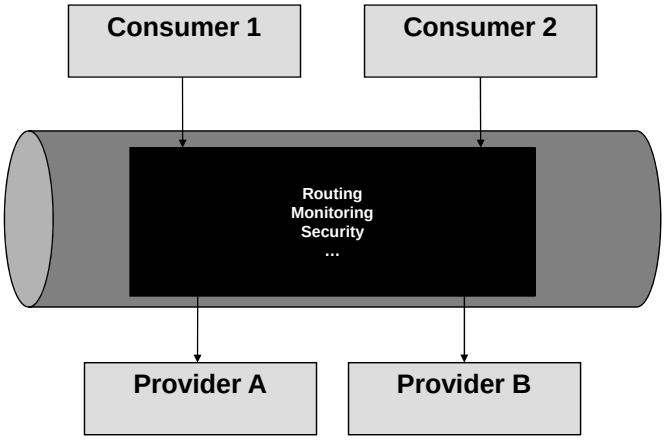
Web Services are Inherently Point-To-Point Connections



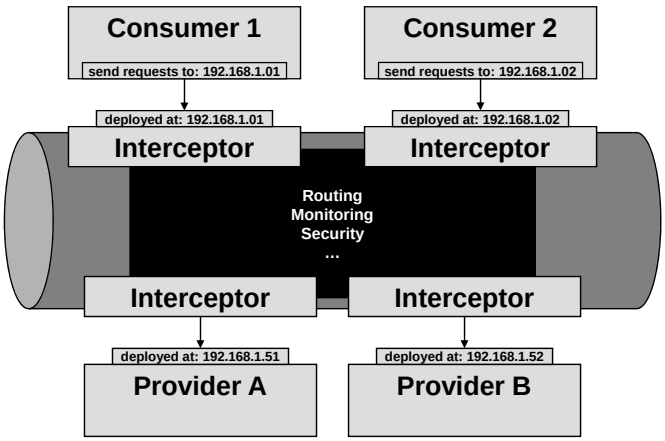
Load-Balanced Providers



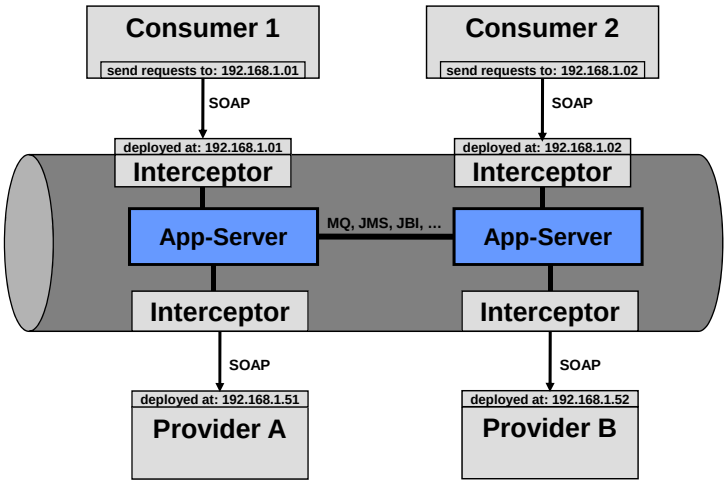
ESB as Central Black Box



Central Intelligence Effects Service Endpoints

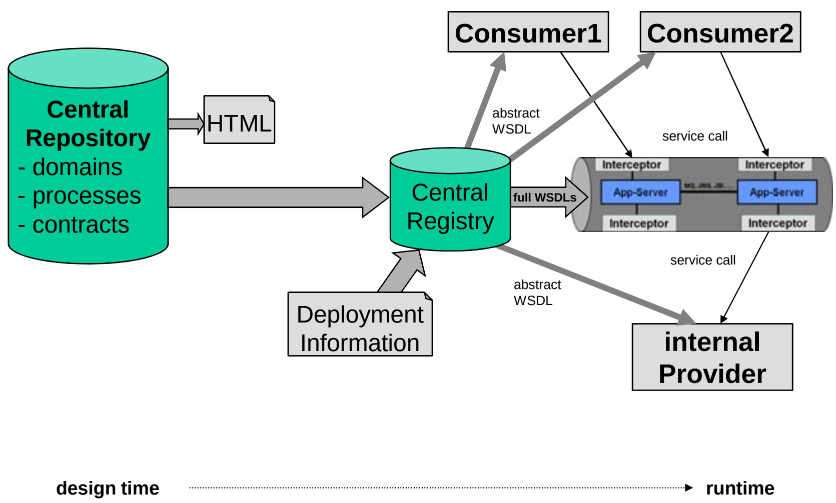


Web Services ESB's Might Internally Use Other Middleware

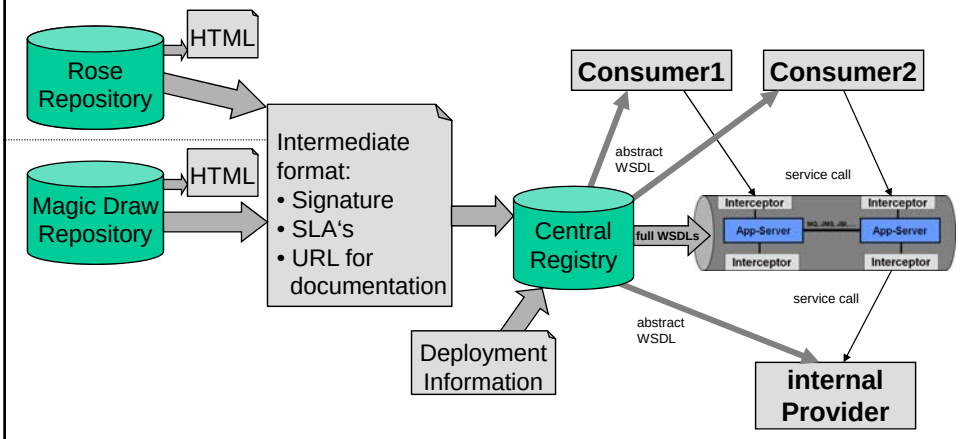


Bringing it all together

Service Development Lifecycle

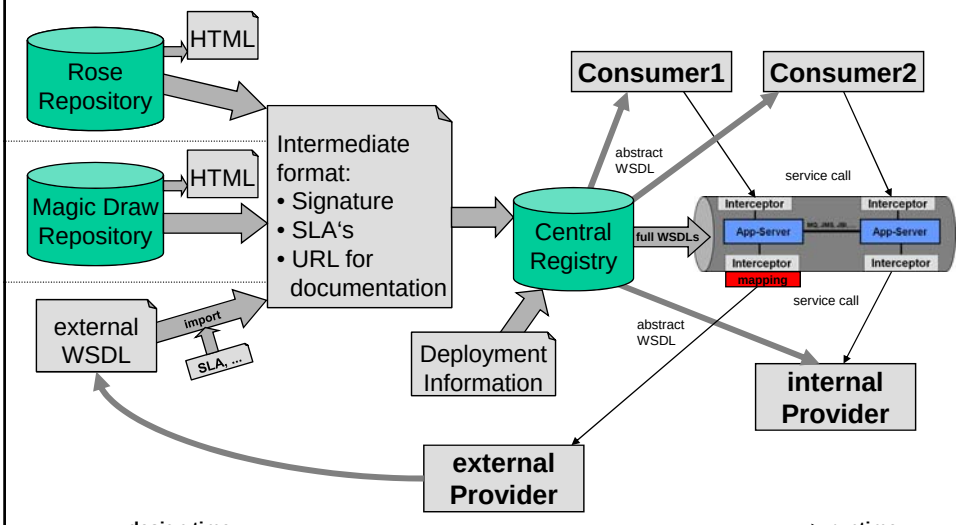


Service Development Lifecycle



design time ⟶ runtime

Service Development Lifecycle

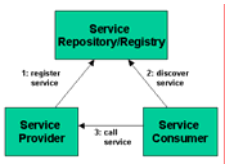


design time ⟶ runtime

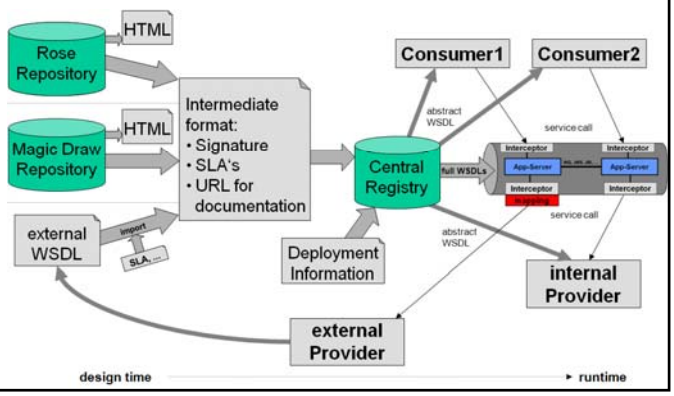
Summary

Web Services in Practice

Model:



Practice:



Problems with Web Services

- **Open Standardization**
  - Leads to Standards that are not interoperable
- **Low-level infrastructure**
  - For intelligent routing and management of services you need more
- **Focus on technical aspects**
  - No inherent support for non-functional attributes
- **Danger of harmonization trap**
  - There will be other infrastructures
  - There will be multiple repositories

Web Services FAQ

- **Should we use Web Services?**
  - What else:
    - nothing (if you can avoid distribution, avoid it)
    - proprietary solution
    - self made
- **What should we bear in mind using Web Services?**
  - Follow WS-I Basic Profile 1.1
  - Use SOAP protocol “document/literal wrapped”
  - Don’t expect everything to work, so
    - be careful with new/special features
    - provide resources to solve certain problems
  - Don’t use WSDL to specify service contracts
- **Are we doing SOA when we use Web Services?**
  - You might only do “JaBoWS” (**J**ust **a** **B**unch **o**f **W**eb **S**ervices)

Web Services Summary

- **Web Services are a de-facto standard**
  - Use it to provide standard interfaces
- **Web Services only care for technical aspects**
  - Don‘t use it as a starting point for service modeling
- **Web Services operate on a pretty low level**
  - Needs effort and/or tools to provide „intelligent“ ESBs
- **Web Services are just a current technology**
  - Be prepared for other technologies

Q&A

Nicolai Josuttis

www.it-communication.com
josuttis@it-communication.com

Gaussstr. 29
D - 38106 Braunschweig
Germany

Tel.: +49 531 / 129 88 86
+49 700 / 5678 8888
+49 700 / JOSUTTIS