Agenda

- The following three short contributions address the needs, solutions, and effects of service orientation in applications as diverse as mobile telecommunications, e-government, and logistics.
  1. Toward Mobile Services: Three Approaches
  2. New paradigms for Next-generation E-Government Projects
  3. Creating Business Value through Flexible IT Architecture

- This presentation illustrates how vital a realm engineering-driven approach is by highlighting the aspects of each contributions:
  - Challenges
  - Solutions
  - approaches

- Lessons learned from these three contribution
For most companies, software products are considerably more valuable if integrated into a product suite from which each customer can pick and choose a unique configuration.

Similarly, in an increasingly networked world, a company can greatly benefit from uniformly integrating its products with those of other companies in the same ecosystem.

A number of tasks can be carried out more efficiently when stationary other can be executed with acceptable or even superior efficiency using mobile devices. This capability usually evolves in a three-stage process.

- A given task cannot be performed using a general-purpose mobile device.
- The task can be performed in mobile contexts, which remains secondary to a stationary solution.
- Users prefer to carry out the task with a mobile device even if a stationary option is available.

Encounter software product companies:
- To provide a uniform and integrated user experience across the desktop, Web, and mobile platforms.
- To maximize the natural architectural of the software system common to all three platforms while minimizing the platform-specific part.
- To simplify and stably the interface between the generic and platform-specific parts.

Impediment of the mobile platform implementation:
- Different mobile device types offer various subsets of mechanical, hardware, and software capabilities that are difficult to hide from the application software running on the device.
- Each device type tends to be on the market for a limited time, typically months rather than years, and the speed of evolution is significant.
- Any device-side software must be tested and validated on each device type as the differences among devices can affect both the software’s ability to execute properly and the user experience.

Solution
- Four requirements needed on mobile Services Oriented Architectures (SOAs)
  - Any service should be provided on a wide variety of mobile devices to maximize its market potential.
  - The service should make full use of the device’s native features.
  - The architecture should facilitate rapid time to market of new services.
  - Services should enjoy a high degree of forward compatibility by allowing the use of the service while a mobile device is updated to a successor model.
Toward Mobile Services

Approaches

Table 1. Three basic options the SOA requirements leave to software developers

<table>
<thead>
<tr>
<th>Approach</th>
<th>Business factors</th>
<th>Technical factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client-server scheme using native client at the mobile device</td>
<td>- High R&amp;D investment for reaching broad customer base&lt;br&gt; - Low product agility</td>
<td>+ Superior user experience as all device capabilities are exploited&lt;br&gt; + Every device type needs a version. Portability, deployability, and business scalability are difficult to achieve.</td>
</tr>
<tr>
<td>Client-server architecture using a Java client</td>
<td>+/- Medium R&amp;D investment&lt;br&gt; +/- Medium product agility</td>
<td>+/- User experience partially exploits device capabilities&lt;br&gt; +/- Offers better portability, easier deployment, and scalability&lt;br&gt; +/- A decreased ability to employ device-specific features impairs the user experience&lt;br&gt; +/- Multiple versions needed</td>
</tr>
<tr>
<td>To develop a thin browser-based client for the device</td>
<td>+/- Low R&amp;D investment&lt;br&gt; +/- High product agility</td>
<td>Low user experience for several classes of applications&lt;br&gt; +/- Single version of product offers superior portability, deployment, and scalability</td>
</tr>
</tbody>
</table>

Software product evolution

Moving from a stand-alone offering to a company- or ecosystem-integration-driven offering has significant consequences for how software products are architected and built

Table 2. Software product evolution

<table>
<thead>
<tr>
<th>Stand-alone product</th>
<th>A product is created by combining, at development time, external infrastructure components, internal shared software assets, and product-specific functionality.</th>
<th>Although the interfaces can be expressed using SOA principles, the main advantage is easier maintenance of software components due to lower coupling.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product suite</td>
<td>A product can, at runtime, integrate with other products in the company’s product suite, but only with those available at development time.</td>
<td>SOA principles can be used, but the advantages mainly lie in decoupling the company’s products.</td>
</tr>
<tr>
<td>Ecosystem integration</td>
<td>A product or service can at runtime uniformly integrate with other products and services developed by any organization in the ecosystem, including those developed after the product’s release.</td>
<td>SOA principles are employed both for decoupling and for runtime discovery and interface negotiation.</td>
</tr>
</tbody>
</table>
Toward Mobile Services

The transition from a stand-alone product to an ecosystem-centric approach requires moving from early integration to a compositional approach to creating products and services, as Figure 1 illustrates. On Figure 1, Architectural guidelines guarantee composability, while components and subsystems guarantee quality.

Figure 1. Transitioning from an early integration to a compositional approach to creating products and services

New Paradigms for Next-Generation E-Government Projects

In e-government

- Three emerging IT trends are defining new rules for the public sector. These paradigms require next-generation e-government projects to shift the emphasis from building on proper technology to gaining a more profound understanding of the nature of organizational transformation and business process management.
- The public services sector entered a digital revolution era in late 1990s to better connect with their people and rebuild public trust in civic institutions.
- Relied on Internet-based technology to support transformative initiatives and improve service provisioning on a short-term basis.

Challenges

- The gap between service provisioning and customer value has unintentionally widened.
- Technological advances have led to a focus on the service front end and neglected the back-end infrastructure resulting in lower customer satisfaction.
- The amount of money expended on failed IT projects in the public services sector is staggering, which made analysts and public officials skeptical about large-scale e-government initiatives.
- Only 40 % of major initiatives are deemed successful, and these often fail to deliver their promised benefits.
- Many officials charged with implementing e-government projects lack the skills required to deliver the promised service improvements.

Solution

- Three IT trends defined new rules to shift the emphasis from building on proper technology to gaining a more profound understanding of the nature of organizational transformation and business process management.
### New Paradigms for Next-Generation E-Government Projects

**Approaches**

<table>
<thead>
<tr>
<th>Isolated functions</th>
<th>To interoperable applications</th>
<th>To integrated process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lighthouse projects</td>
<td>Interoperability applications</td>
<td>Integrated process</td>
</tr>
</tbody>
</table>

- **Table 3. Three emerging IT trends defining new rules for the public sector**
  - Beginning around 2004, the shift concentrated on IT infrastructure such as hardware, networks, and specific applications which led to a more intensive exchange of data among public institutions, private companies, and citizens.
  - Value chain integration and automatic synchronization of databases and business processes provide efficiencies that have the potential to generate tremendous economic benefits.
  - Interoperability can get the most out of reorganization, collaboration, and standardization.
  - An example of this trend is the ambitious goal of creating a single market for services in the EU.
  - Because digitizing existing processes alone will not produce the desired effect, this paradigm stresses government institutions' organizational framework.
  - Improving an organization's efficiency and effectiveness by optimizing and whenever possible integrating its business processes.
  - Authorities in the UK, Germany, and Belgium use the LoG-IN Generic Information Infrastructure (www.login-project.net) to build and deploy powerful Web services using only a browser.
  - Businesses can incorporate information from the system, which is based on XML/GML open standard specifications, into their own Web sites or processes to improve efficiency and more effectively support human-human interaction.

---

### Creating business value through flexible IT architecture

#### In creating business value

- Globalization and e-commerce have led to rapid growth and constant change in the logistics market. Modern supply chains span many tiers of logistics service providers, suppliers, retailers, and customers. Continuously unbundling value chains increases the breadth and complexity of those networks.
- Therefore, mastering IT management and establishing a flexible architecture is a key prerequisite for excellent IT delivery.
- In addition to mastering IT management, a flexible architecture is a key prerequisite for excellent IT delivery. In an evolutionary approach to architecture transformation, Deutsche Post developed a coordinated set of projects that implemented and reused business services, thus both creating business value and contributing to a flexible IT landscape.

#### Challenges

- IT system landscapes at most large enterprises resemble spaghetti bowls instead of clear structures.
- Years ago, Deutsche mail division faced massive competition, deregulation, and globalization challenges. Its IT architecture was hampering, rather than fostering, change.

#### Solution

- SOA program was business-driven, rather than IT-driven.
- Deutsche post started an architecture initiative intended to achieve higher business flexibility through more adaptive IT systems, make it more effective through increased IT ownership by businesspeople, provide a baseline for effective IT governance, focus on the creation of differentiating business functionality, while reusing existing business services, and develop the application landscape in a “managed evolution” manner.
Creating business value through flexible IT architecture

### Approaches

<table>
<thead>
<tr>
<th>Business factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Through reuse of business services, as well as hiding legacy applications and infrastructure behind the business service facades, the logical service architecture helped reduce the side effects of business and IT change, facilitating effective reuse of IT assets.</td>
</tr>
<tr>
<td>The core idea of this managed evolution was to strike a balance between short-term business value and long-term IT integrity.</td>
</tr>
<tr>
<td>A coordinated set of projects implemented and reused business services, thus both creating business value and contributing to a flexible IT landscape.</td>
</tr>
</tbody>
</table>

#### Table 1. Logical services architectures

<table>
<thead>
<tr>
<th>Approach</th>
<th>Business factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business logic was hard coded in applications that were integrated point-to-point and featured redundant data.</td>
<td></td>
</tr>
<tr>
<td>The postal provider initially defined a logical-service-architecture layer bridging the process-architecture layer and the application-architecture layer.</td>
<td></td>
</tr>
<tr>
<td>Logical-service-architecture layer consisted of business domains, business objects, and business services and was the core of SOA.</td>
<td></td>
</tr>
</tbody>
</table>

Creating business value through flexible IT architecture

### Approaches

<table>
<thead>
<tr>
<th>Federal IT governance</th>
<th>Service design</th>
</tr>
</thead>
<tbody>
<tr>
<td>The logical service architecture let Deutsche Post pursue a federal, rather than central, paradigm for IT organization and governance.</td>
<td>Service design is a key ingredient to implementing SOA.</td>
</tr>
<tr>
<td>It reflects the fundamental conviction that to increase IT effectiveness, responsibility should lie closest to the business units and their management where the value originates.</td>
<td>Two comprehensive service-design processes “service portfolio management and service standards and infrastructure management” ensure that services can interact and contribute to the evolution of the architecture blueprint.</td>
</tr>
<tr>
<td>To foster individual management ownership for specific business services, Deutsche Post’s business-management and enterprise architects together factored its business system into domains by thorough analysis.</td>
<td>For each service, the processes of service specification, realization, and life-cycle management govern the life cycle of each service.</td>
</tr>
<tr>
<td>Domains are coarse-grained and loosely coupled units in a logical service architecture, bundling responsibility for a cohesive set of business logic, functionality, and data.</td>
<td>The first step in creating and managing a business-service portfolio is target-service discovery. A thorough semantic harmonization of business concepts is a key prerequisite to the definition of a reusable service portfolio.</td>
</tr>
<tr>
<td>Deutsche Post’s domain landscape features 13 domains (relationship, operations, or finance) that businesspeople own, thus establishing a reference grid for federal business-driven IT governance.</td>
<td>The managed-evolution approach aims to create a rich service portfolio, but it involves solid, long-term strategy.</td>
</tr>
</tbody>
</table>
During several years of implementing services, early SOA adopters reported declines of 60-70% in integration costs and 10-20% in development and maintenance costs, as well as a 30% increase in time to market.

Besides implementing projects more quickly, Deutsche Post flattened its 15 percent annual increase in IT costs. The quality level of customer data soared, and SOA at Deutsche Post became mission critical.

From IT architecture flexibility through its SOA, Deutsche Post learned to
- Focus on semantic integration, not technology, and harmonize business terms, since no technology platform will help solve these fundamental issues;
- Let business, not IT, drive the integration effort, and use SOA to provide a common language between IT and businesspeople;
- Actively manage the evolution of the service portfolio to gain long-term architectural stability;
- Protect existing assets, evolve to SOA, and exploit a business domain structure to decouple modification cycle times in the application landscape;
- Employ enterprise architecture as the essential IT governance lever, and enjoy the de-bottlenecking power of decentralized business initiatives without ending up in a mess.

Lessons learned from these three contributions

- Today, companies aiming to offer their software products and services through desktop, Web, and mobile platforms might need to compromise certain quality requirements while selecting the appropriate mobile SOA.
- Internet browsers are increasingly becoming standards-based application containers, leading to convergence also for mobile browsers.
- Local capabilities on the mobile device might well become services themselves, simplifying the use of device-specific features and thereby improving the user experience.
- By using SOA, the mobile platform is bound to become the platform of choice for an increased set of everyday tasks.

SOA = Semantic integration + Loose coupling + Managed evolution.
- Semantic integration is the major prerequisite and challenge.
- Loose coupling is the distinct feature of SOA.
- Managed evolution represents both a purpose and implementation approach.