Monolith to Microservices: A Best Practices Approach

Jasper van der Hoek / Enterprise Architect
Architecture Trends
Architecture Trends

Traditional Large System
Internal Modular Design

GUI
Logic
Monolith System
Database
Process
Architecture Trends

**SOA “System”**
Layered Re-Used Components

- **UI Layer**
- **Logic Layer**
- **Process Layer**
- **Data Layer**
  - database
  - Back-end systems

**ESB**
Recent Architecture Trends

Microservices System
Independent Specific Components
Recent Architecture Trends

**Traditional Large System**
- Internal Modular Design
- GUI
- Logic
- Database
- Process

**SOA “System”**
- Layered Re-Used Components
- UI Layer
- Logic Layer
- Process Layer
- Data Layer
- ESB

**Microservices System**
- Independent Specific Components
- Micro-Service
- Micro-Service
- Micro-Service
- Micro-Service
- Database
- Back-end systems

- Process Layer
Recent Architecture Trends

Traditional Large System
Internal Modular Design

SOA “System”
Layered Re-Used Components

Microservices System
Independent Specific Components

<table>
<thead>
<tr>
<th>Time p. Feature</th>
<th>App Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GUI
Logic
Database
Process

Monolith System

<table>
<thead>
<tr>
<th>Time p. Feature</th>
<th>App Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UI Layer
Logic Layer
Process Layer
Data Layer
ESB

<table>
<thead>
<tr>
<th>Time p. Feature</th>
<th>App Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Micro-service
Back-end systems

<table>
<thead>
<tr>
<th>Time p. Feature</th>
<th>App Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Microservices Search Trends
Microservices Search Trends vs SOA
Why Microservices?

- Increase speed of development
- Improve scalability
- Improve Quality
- (Long-term) Architectural Flexibility
- Independent Release Management
How?

- Maximize re-use
- Reduce dependencies
- Use single database
- Scale individual services
- Reduce regression testing
- Standardize technology
Re-use

Maximize re-use

Standardize technology

Use single database
Re-use versus Efficiency

Maximize re-use

Standardize technology

Use single database

Autonomous Functional Services

Best Technology for the Job

Services own their data
How to choose the right Microservices
Choosing the Right Microservices

- Increase speed of development
- Improve scalability
- Improve Quality
- (Long-term) Architectural Flexibility
- Independent Release Management
Choosing the Right Microservices

- Identify unique business domains by leveraging the ‘Bounded context’
- Design for clear separation in data and ownership
Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.

- M. Conway
Aligning Organization and Systems

<table>
<thead>
<tr>
<th>Business Processes</th>
<th>IT Solutions (Microservices)</th>
</tr>
</thead>
</table>
Aligning Business with IT

Business

- Business Processes

IT

- IT Solutions (Microservices)
Aligning Business with IT... and technology
Aligning Business with IT... and technology

<table>
<thead>
<tr>
<th>Business Processes</th>
<th>IT Solutions (Microservices)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Breaking down the Monolith

A value driven approach
Choosing the *Right* Microservices

### Level 1
- Differentiator
- Differentiator
- Differentiator
- Differentiator
- Differentiator
- Differentiator
- Differentiator

### Level 2
- Value Driver
- Value Driver
- Value Driver

### Level 3
- Business Line
- Business Line
- Business Line
- Business Line
- Business Line

### Company Mission
- Value Driver
- Value Driver
- Value Driver

### Process
- Process
- Process
- Process
Choosing the *Right* Microservices

- Increase speed of development
- Improve scalability
- Improve Quality
- (Long-term) Architectural Flexibility
- Independent Release Management
Validate Your Microservices

Service Lifecycle & Responsibility
• Start of the service lifecycle
• End of the service lifecycle

Business Events (incoming/outgoing)
• Describe the events in a single sentence

Process description
• Identification of major processes (non-exhaustive)

Stakeholders
• Roles of users of the service
Thank you!

To learn more, connect with Expert Services and ask them about our approach to Microservices.

Mendix Makes Microservices easier
Checkout these track tomorrow:

**Wed — 2:15 pm**
The aha! Moment: How eXp Realty reached the decision to build a microservices architecture

**Wed — 3:15 pm**
Microservices: What, Why, When and How

**Wed — 4:15 pm**
OData! Bringing structure and syntax to accessing your data