

The background of the slide is a photograph of a spiral staircase. The staircase is viewed from a low angle, looking up. The railings are dark, and the steps are light-colored. A prominent yellow stripe runs along the inner edge of the railings, creating a strong visual contrast. The sky is visible through the center of the spiral, showing some clouds.

# Meet Mike, the SOA Architect

## Erfahrungen aus SOA Projekten

# About me...

Eduard Hildebrandt

*IT Consultant*



EAI



SOA



MDA



+49 (0711) 72846627



+49 (0160) 8870983



[eduard.hildebrandt@logica.com](mailto:eduard.hildebrandt@logica.com)



Meet **Mike**...

Mike **lives** in...

Munich, Germany



he **drives**...



he **loves**...





Mike **cares** for...

he **cheers** for...





and **makes** time for...





Mike's job involves **integration**...  
...of different systems.



He heard a lot **buzz words**...





But finding the **right path**...  
...in the integration labyrinth...  
...is not an easy task!



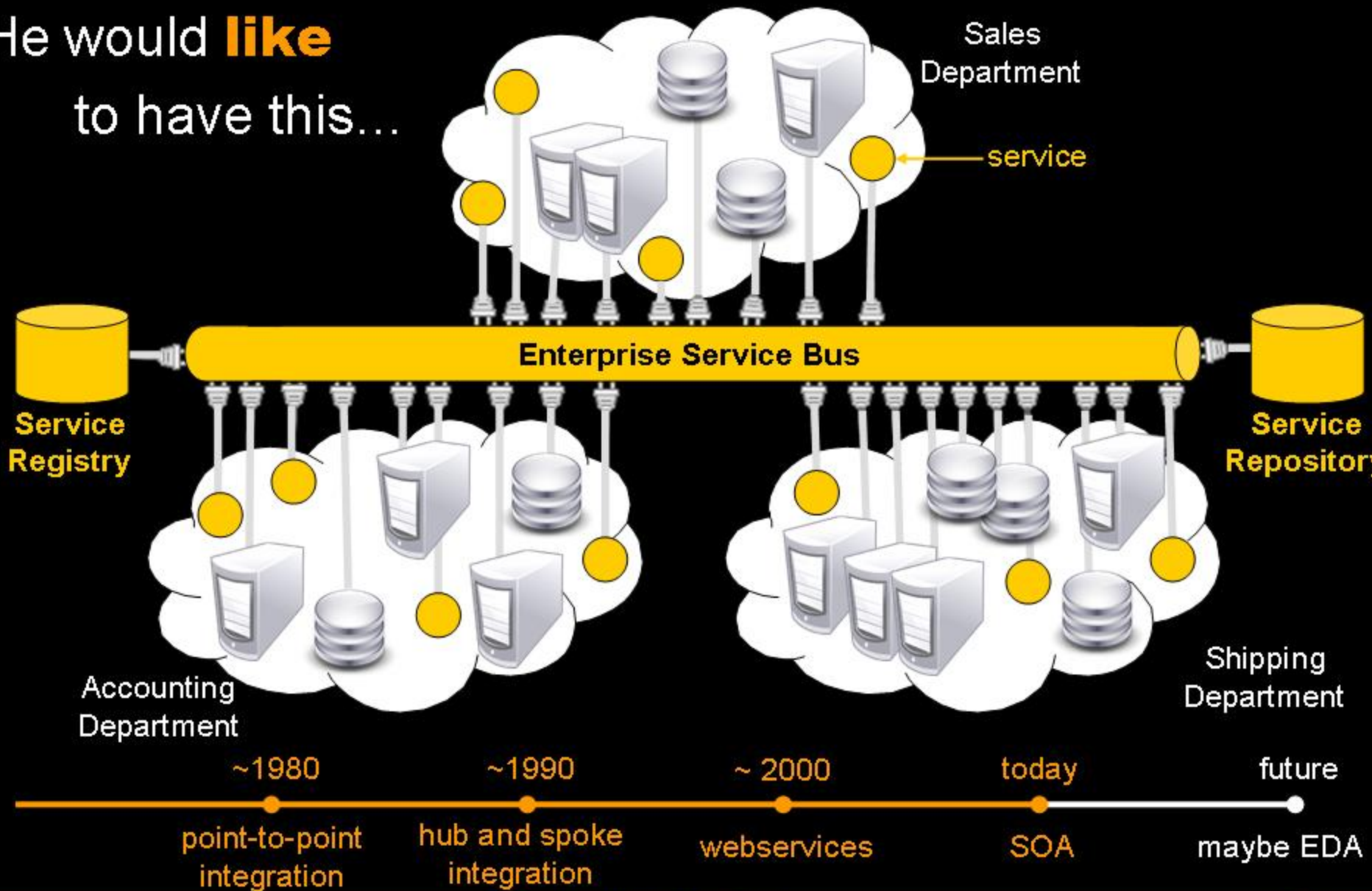
Where to start?

Which path to go?

What to do?



He would **like**  
to have this...





One day Mike  
**meets** Tom...

The background of the slide is a photograph of a spiral staircase. The staircase is viewed from a low angle, looking up, creating a sense of depth and rotation. The railings are dark, and the steps are light-colored. The sky is visible through the central opening of the spiral, showing a cloudy sky. The overall color palette is dominated by greys, blacks, and whites, with a prominent yellow highlight on the railings.

Tom **works** for...  
... a consulting company.



He is an **evangelist**...  
...for **SOA** and **BPM**.





... **ask** Tom what **SOA** is?



He'd give it some thought...  
...and say it is a **trick question**

„The discussion about SOA is like...  
some **blind men**...



... **touching an elephant.**“



# Everybody touches just a **small part**...

It is like...

a rope



a pillar



a big hand fan



a solid pipe



a huge wall



a branch of a tree

# Everybody understands **something different**..

SOA is all about...

loose coupling



webservices



process management



governance



integration



agility

SOA is a **way of thinking**...  
...you have to **understand** and **live** it.

You might call it:

- concept
- vision
- philosophy
- values system
- style
- paradigm





SOA is not  
a concrete architecture.

It **leads** to  
a concrete architecture!



SOA is a **concept** for...  
...large, distributed,  
heterogenic systems...

...consisting of **3 elements**:

- services
- loose coupling
- governance

A low-angle photograph of a space shuttle launching, viewed from below. The shuttle is ascending vertically, leaving a massive, bright white plume of exhaust and smoke that fills the lower half of the frame. The sky is a clear, deep blue. The shuttle's orange external tank and white boosters are visible, along with the white orbiter. The lighting is bright, suggesting a clear day.

With some help from Tom...  
...Mike **initialized** a SOA initiative...  
...in his company.



The core of their  
architecture was a  
**messaging system!**





Using a messaging system...  
... **reduces** dependencies.





location  
dependencies



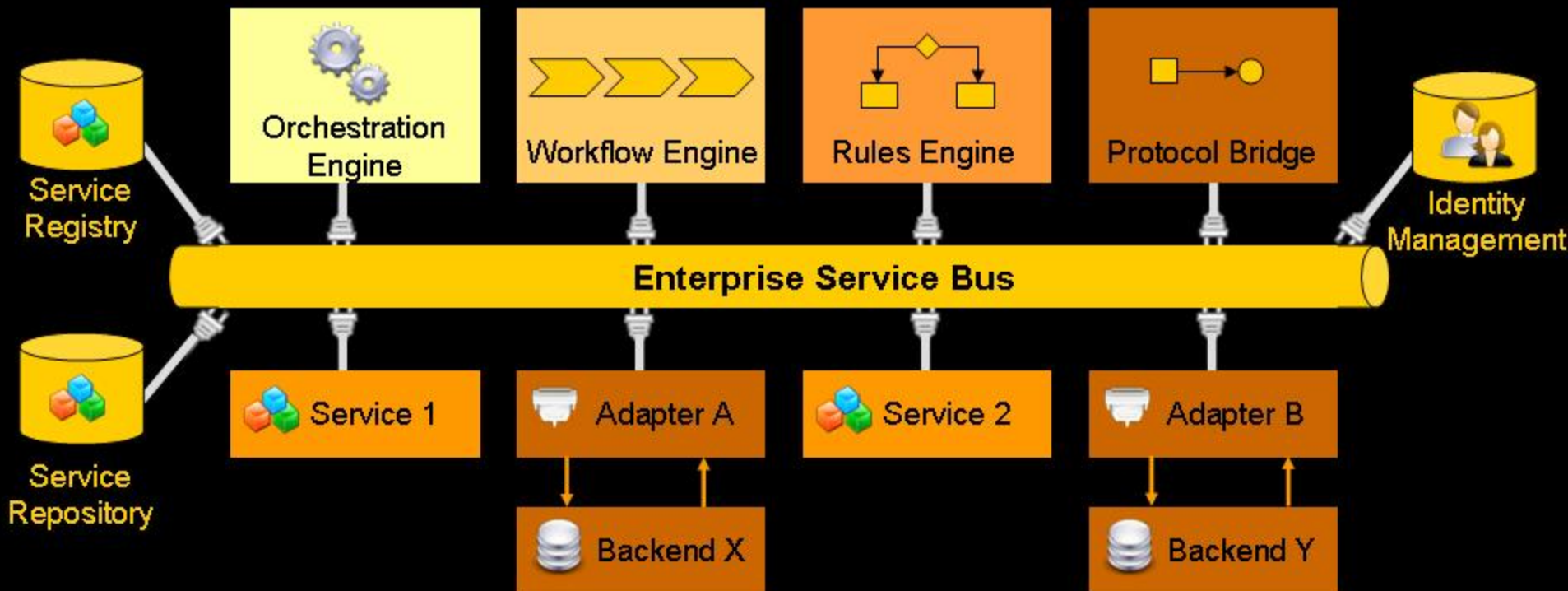
data format  
dependencies



temporal  
dependencies

Using a messaging system...  
... **reduces** dependencies.

Here is an overview of their **architecture**:



But you don't have SOA just by...  
...buying an ESB...  
and using Webservices!

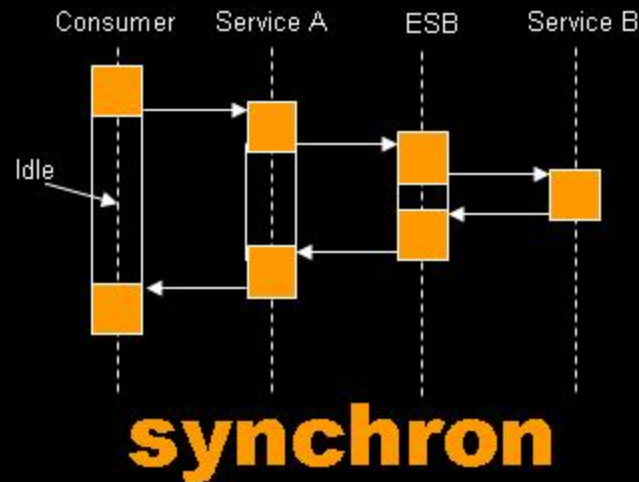
People are not  
easy to change...



They often keep their  
**bad habits.**

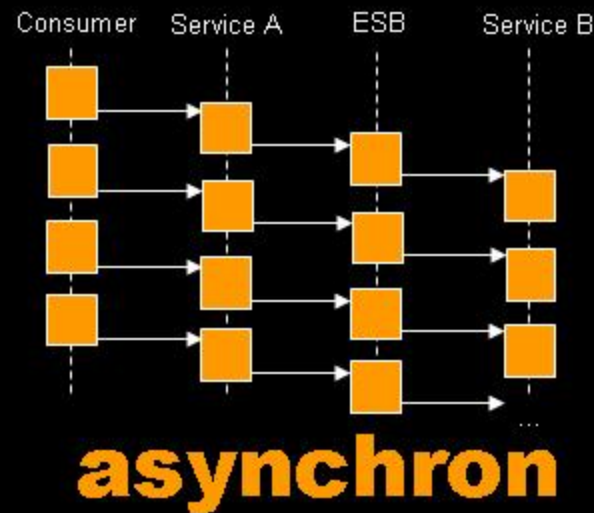


# Developers like to think...



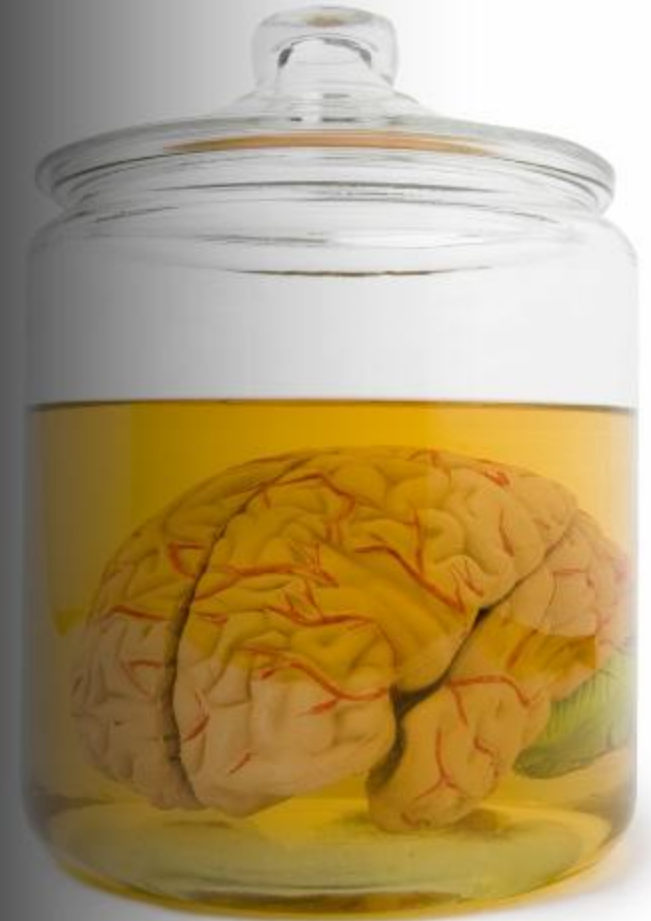
- ✘ Consumer and services are still tightly coupled.
- ✘ Bad throuput because consumer has to wait.
- ✘ Consumer must implement time-out-handling.
- ✘ Message must have a time-out.

# ...instead of...



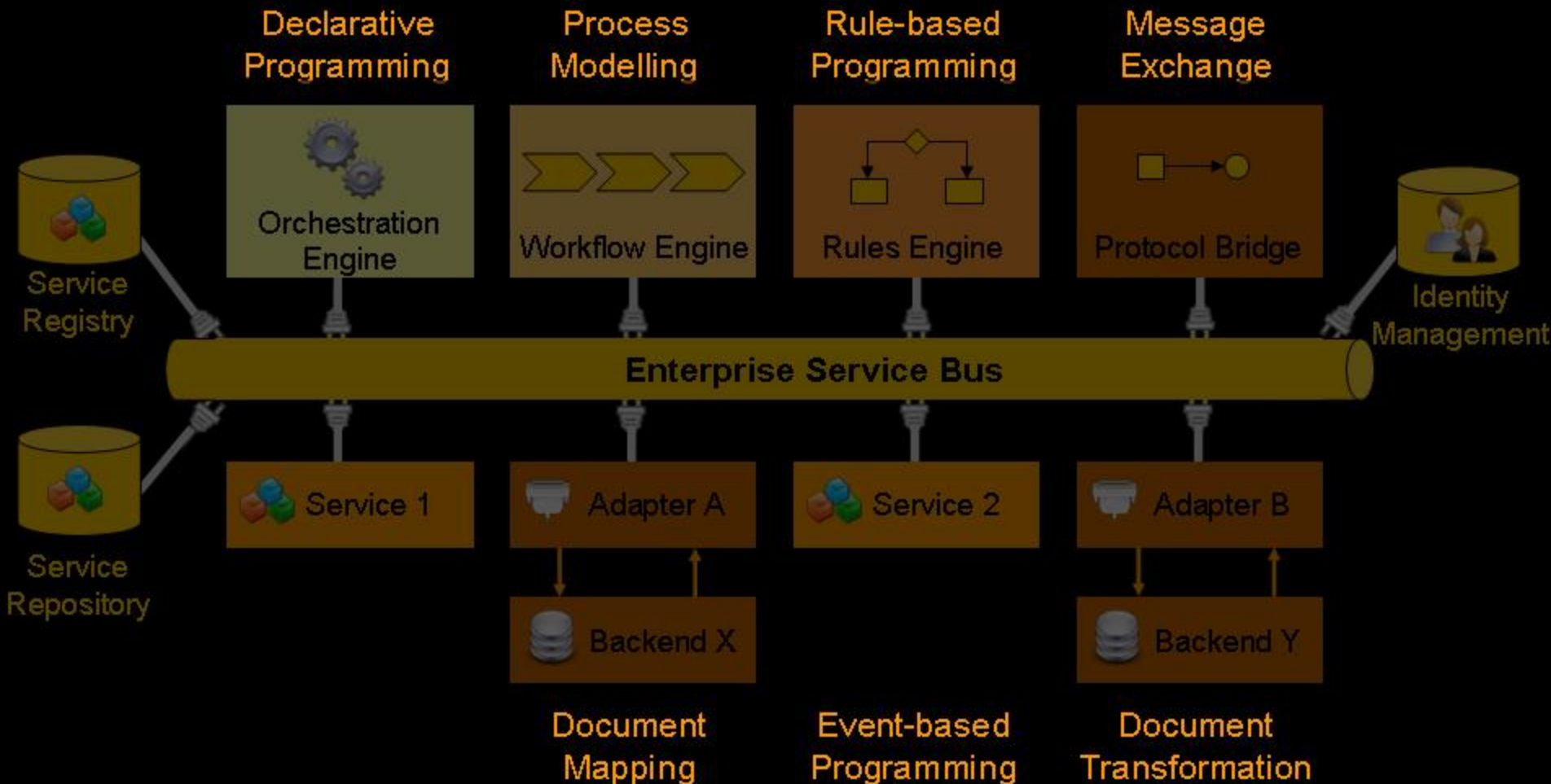
- ✓ Consumer and services are loosely coupled.
- ✓ Better throuput because nobody has to wait.
- ✓ Time-out-handling is not necessary.

And this is just one  
of many examples...



SOA brings many  
**paradigm changes!**

Let's **take a look** at the reference architecture:





Mike and Tom asked themselves:  
**How can we communicate  
good solutions to the people?**





They **marketed** it as...



# The Teddy Bear Concept



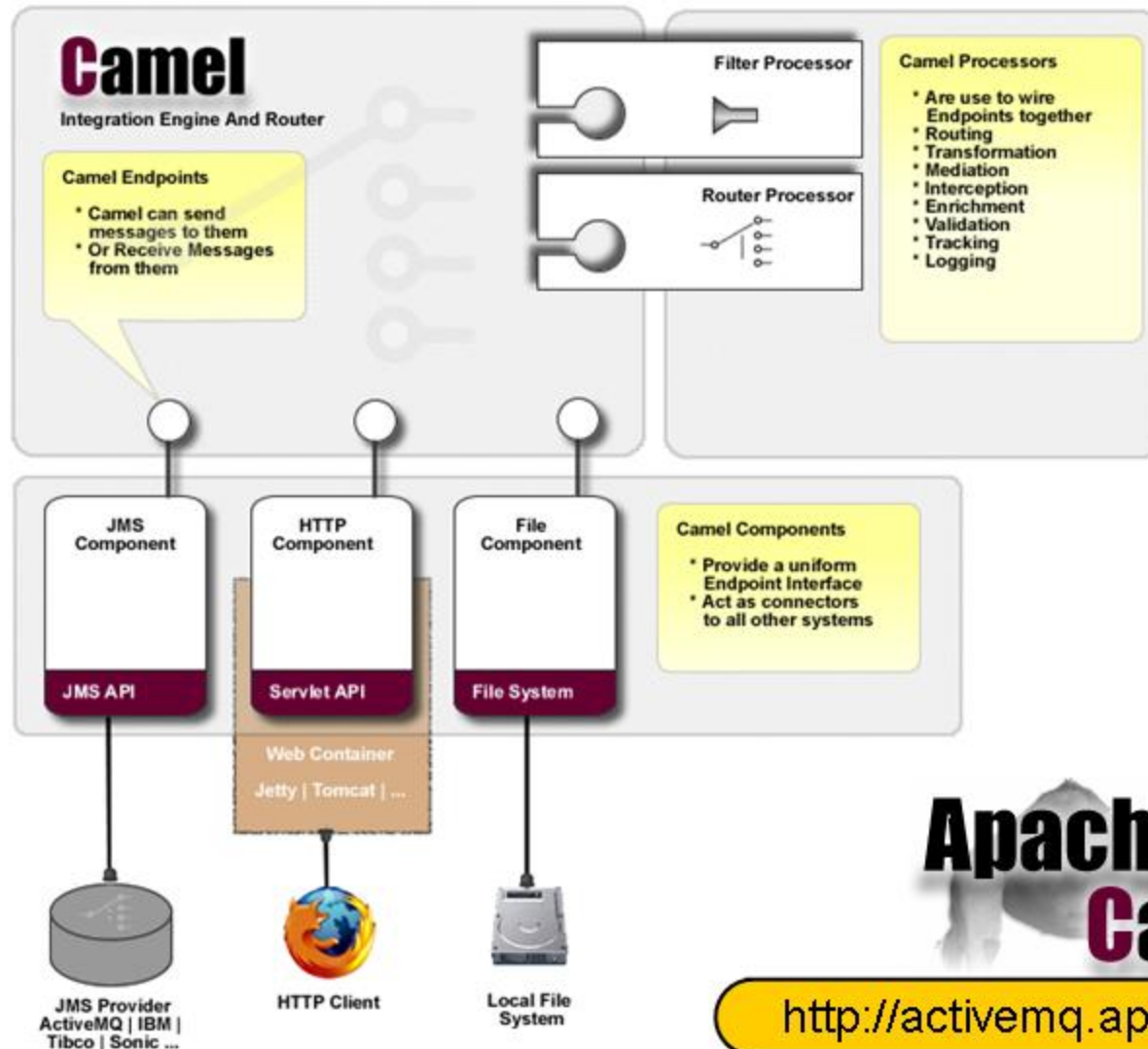
„Every developer gets a personal mentor (teddy bear) sitting near his desk helping him to implement his first service.“



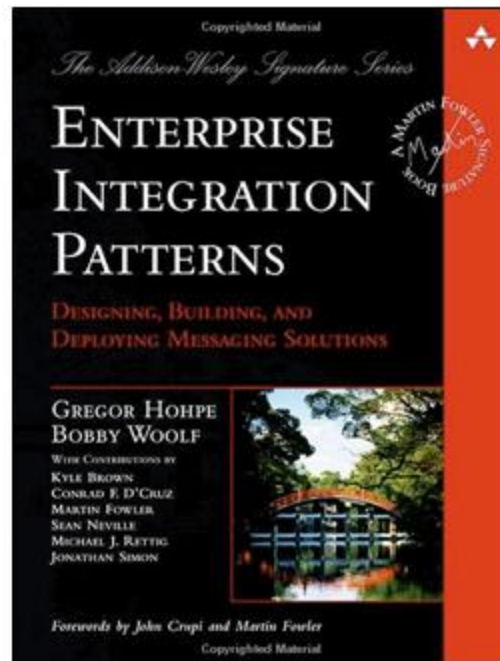


Implementing their integration solutions Tom & Mike made great experience with **Apache Camel**.

# Apache Camel architecture



You can find a **detailed description** of 65 patterns in:



**Titel:** Enterprise Integration Patterns  
**Autor:** Gregor Hohpe und Bobby Woolf  
**Verlag:** Addison-Wesley  
**ISBN:** 0321200683

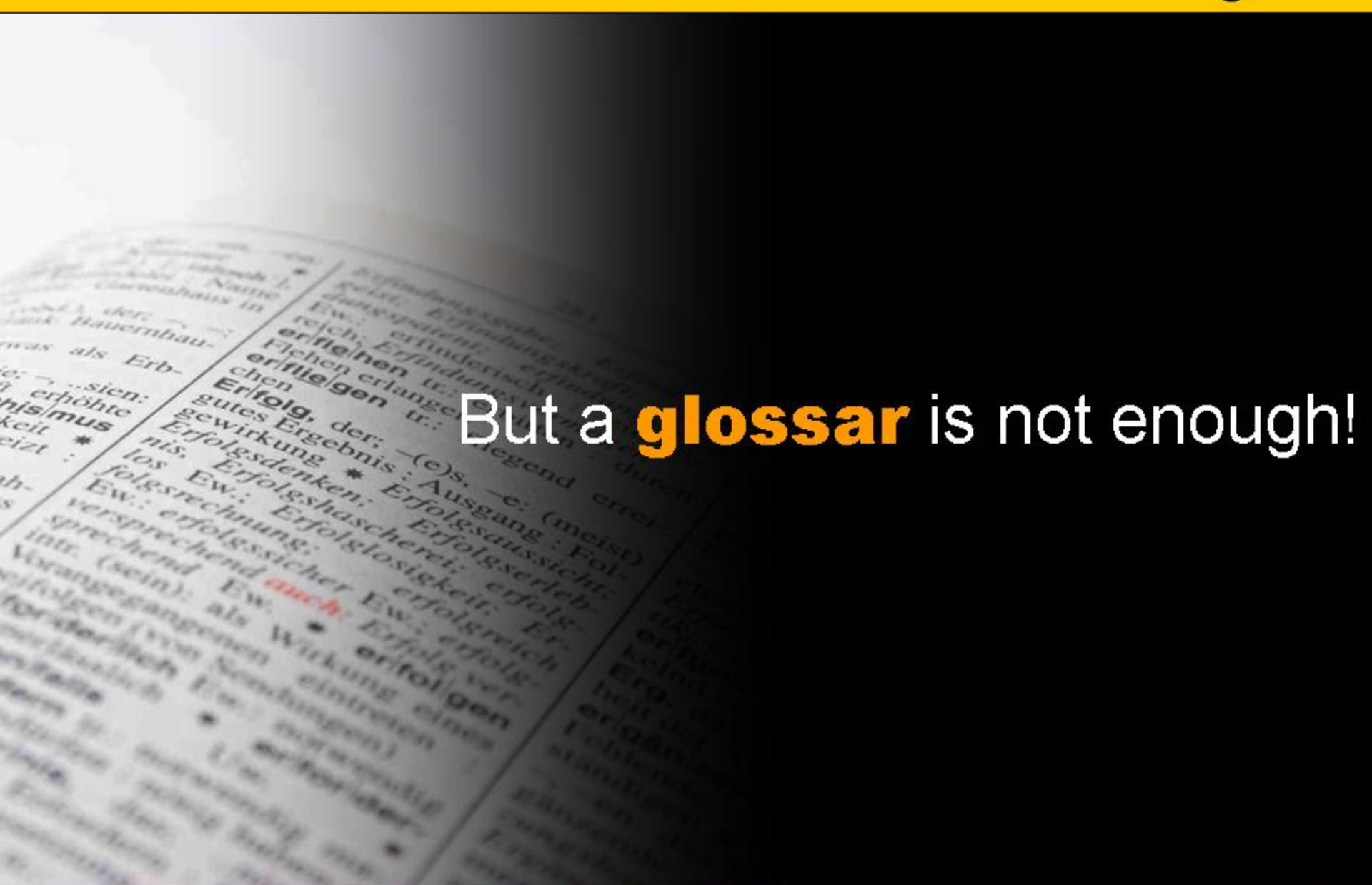


Max understands that...  
to be **successful**...



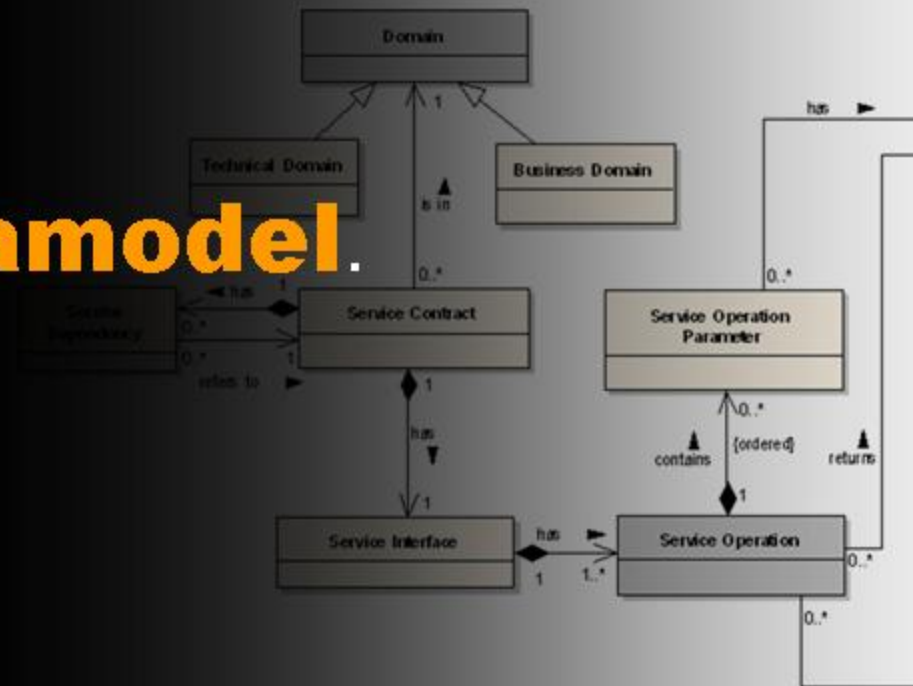
...the team needs to speak...  
... a **common language!**





But a **glossar** is not enough!

# Tom suggested to... ...develop a **metamodel**.



A metamodel specifies:

- ✓ all important entities (artifacts, roles, ...)
- ✓ associations between entities
- ✓ different views of the model



Mike & Tom discussed

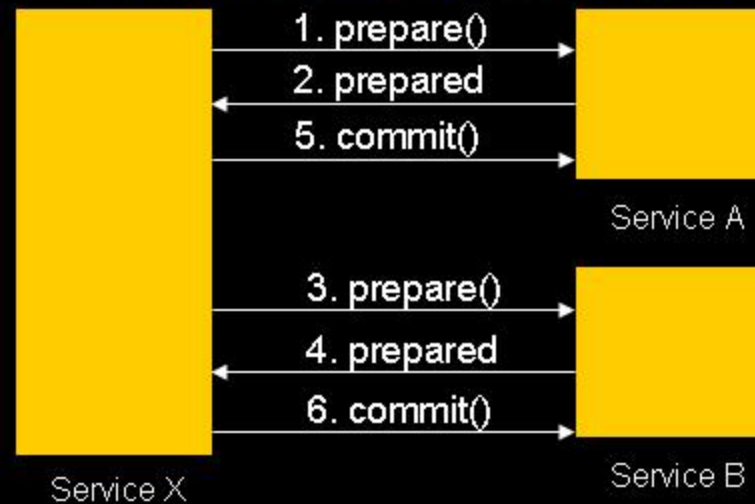
**Transactions**

versus

**Compensation**

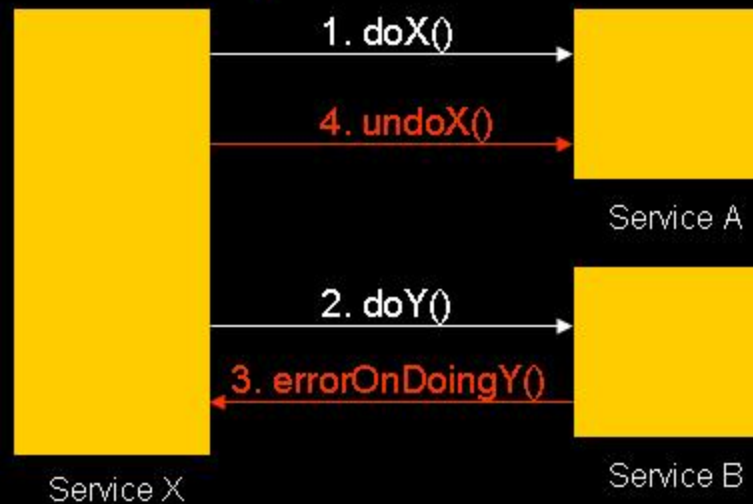


# Transactions



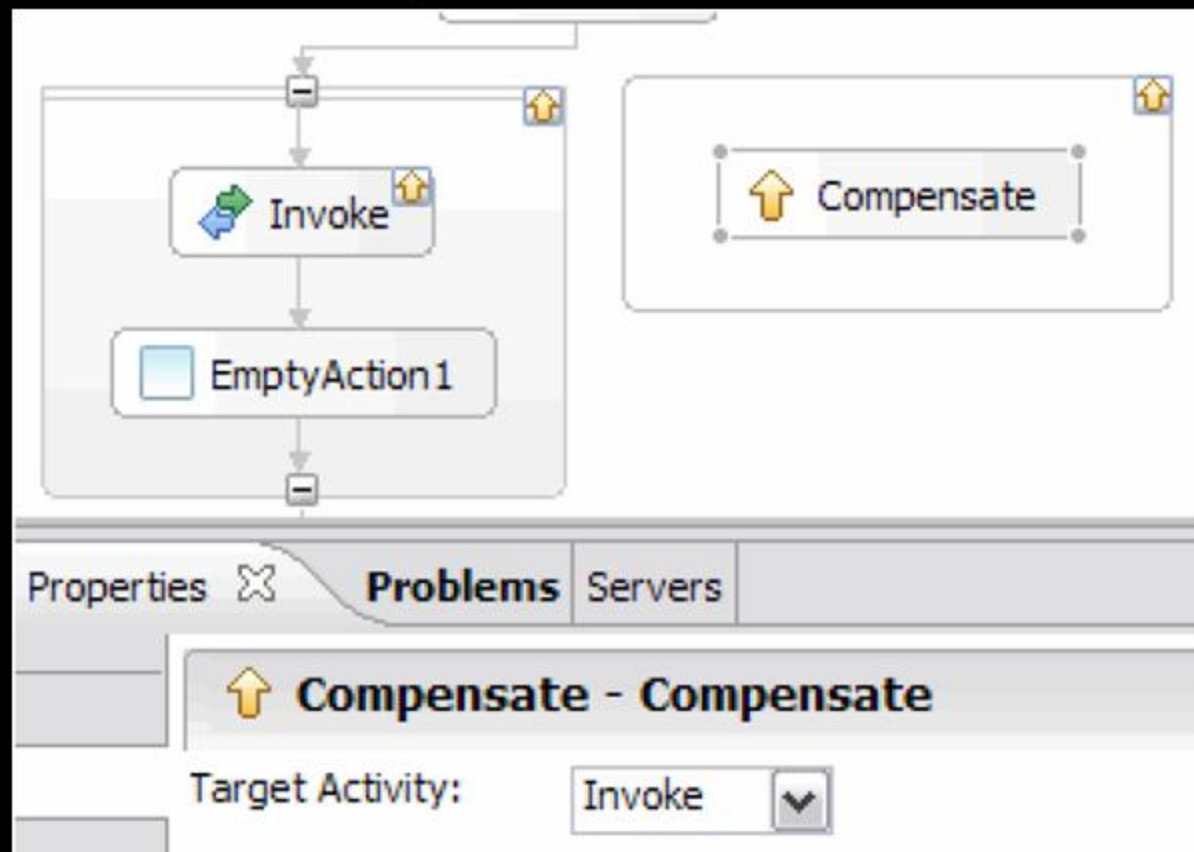
- tight coupling of services
- locks data and limit throuput
- can produce dead locks
- coupling of services
- many very complex webservice standards (WS-\*)

# Compensation



- + loose coupling of services
- + better throuput because no locks
- some operation can not be undone
- compensation behavior must be specified

# An **example** for compensation implementation:



(IBM WebSphere Integration Developer)



Mike & Tom  
discussed

**Choreography**

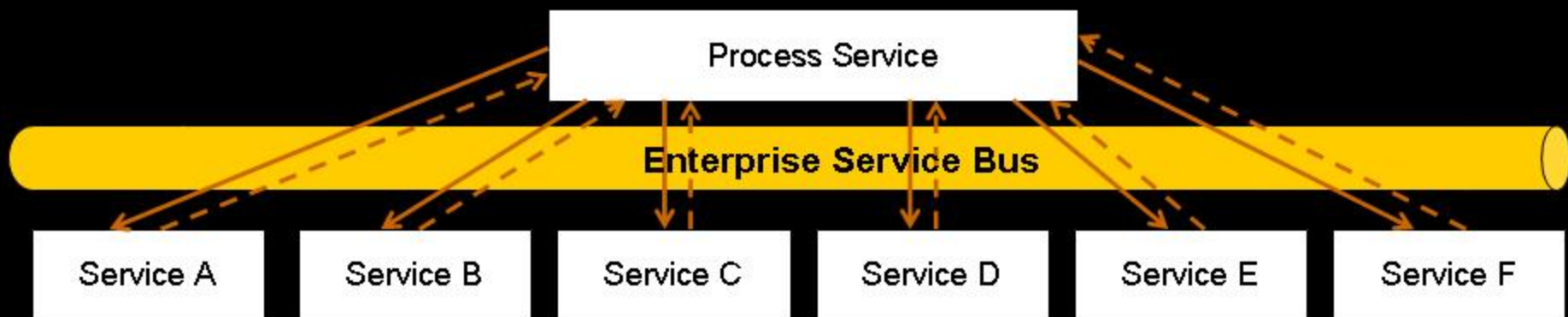
versus



**Orchestration**

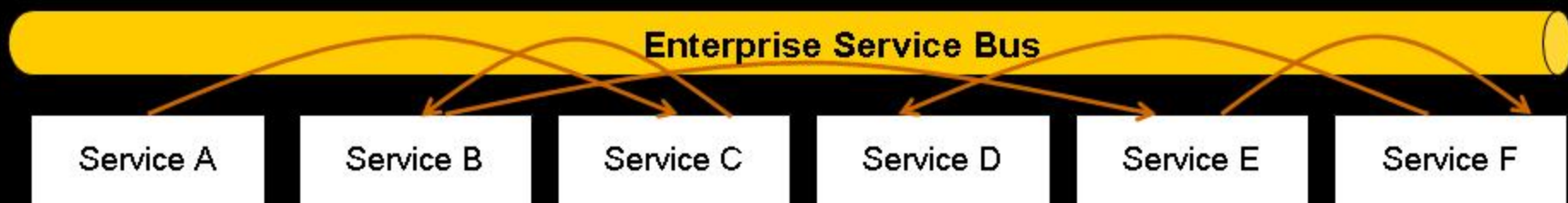


# Orchestration



- + status always available
- + central point for process logic and monitoring
- coupling of services
- single point of failure

# Choreography

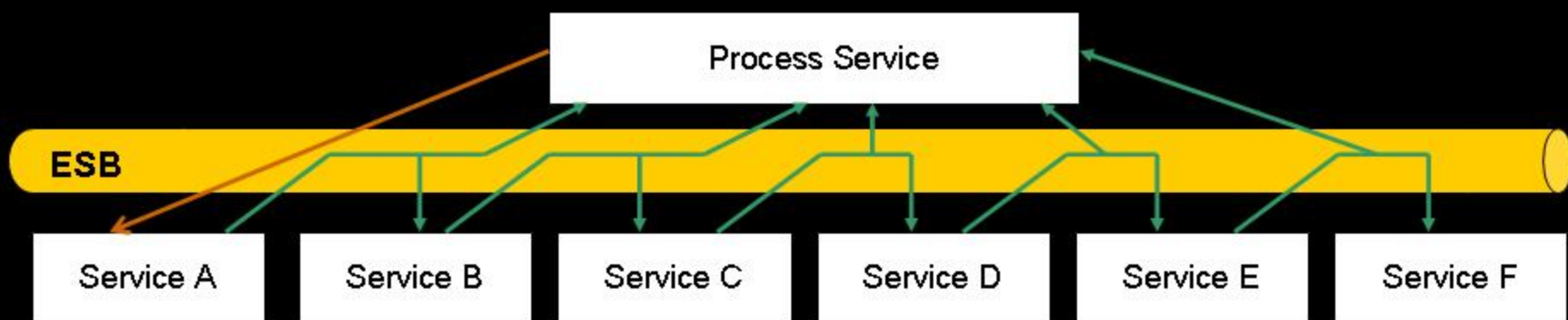


- + loose coupling of services
- + better fault tolerance
- no status information
- spreading of process logic

# Orchestration or Choreography?

It's not an easy decision!

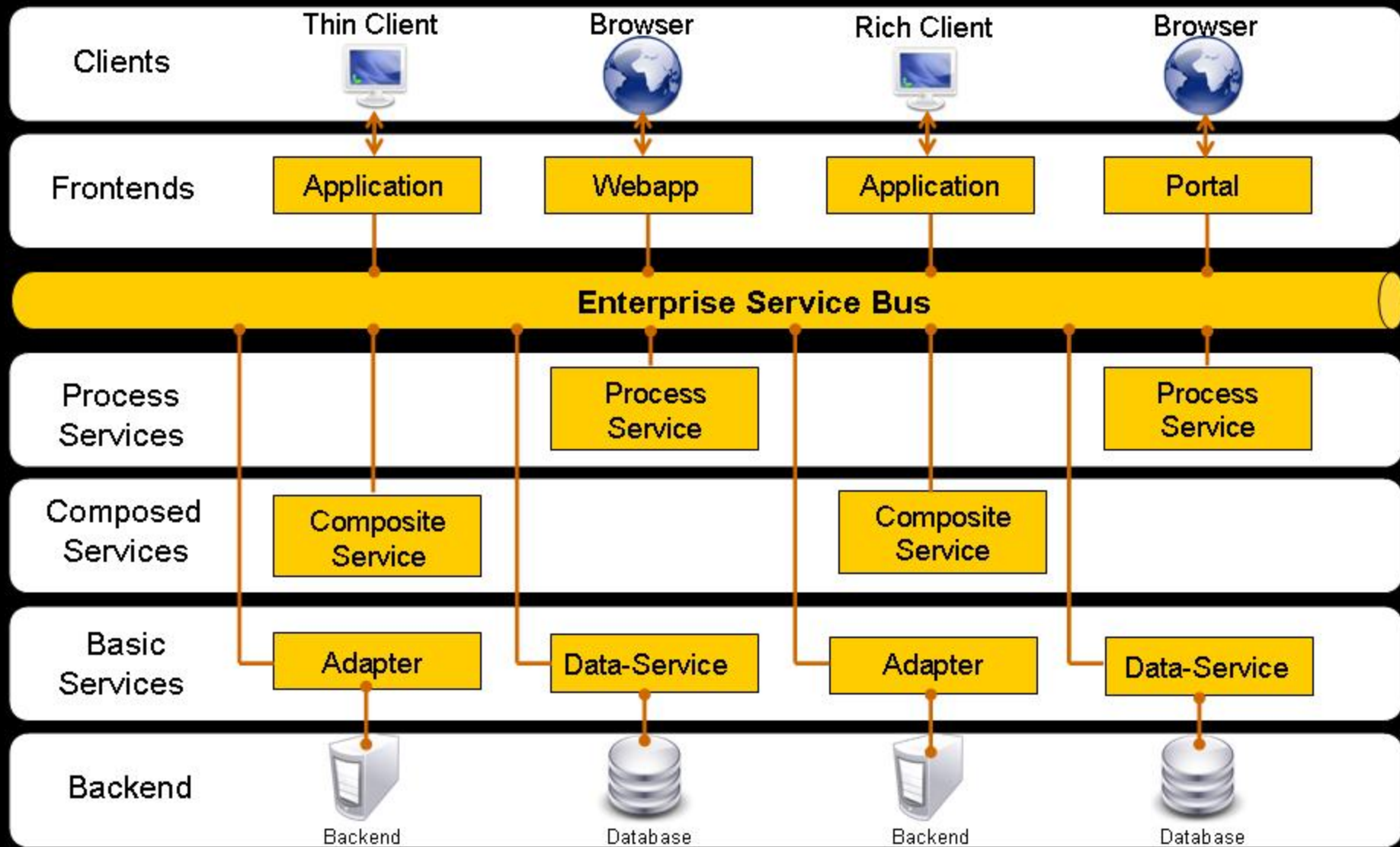
Think about a **comibination**...





Mike & Tom tried to **adapt**...  
...good architecture concepts...  
like...

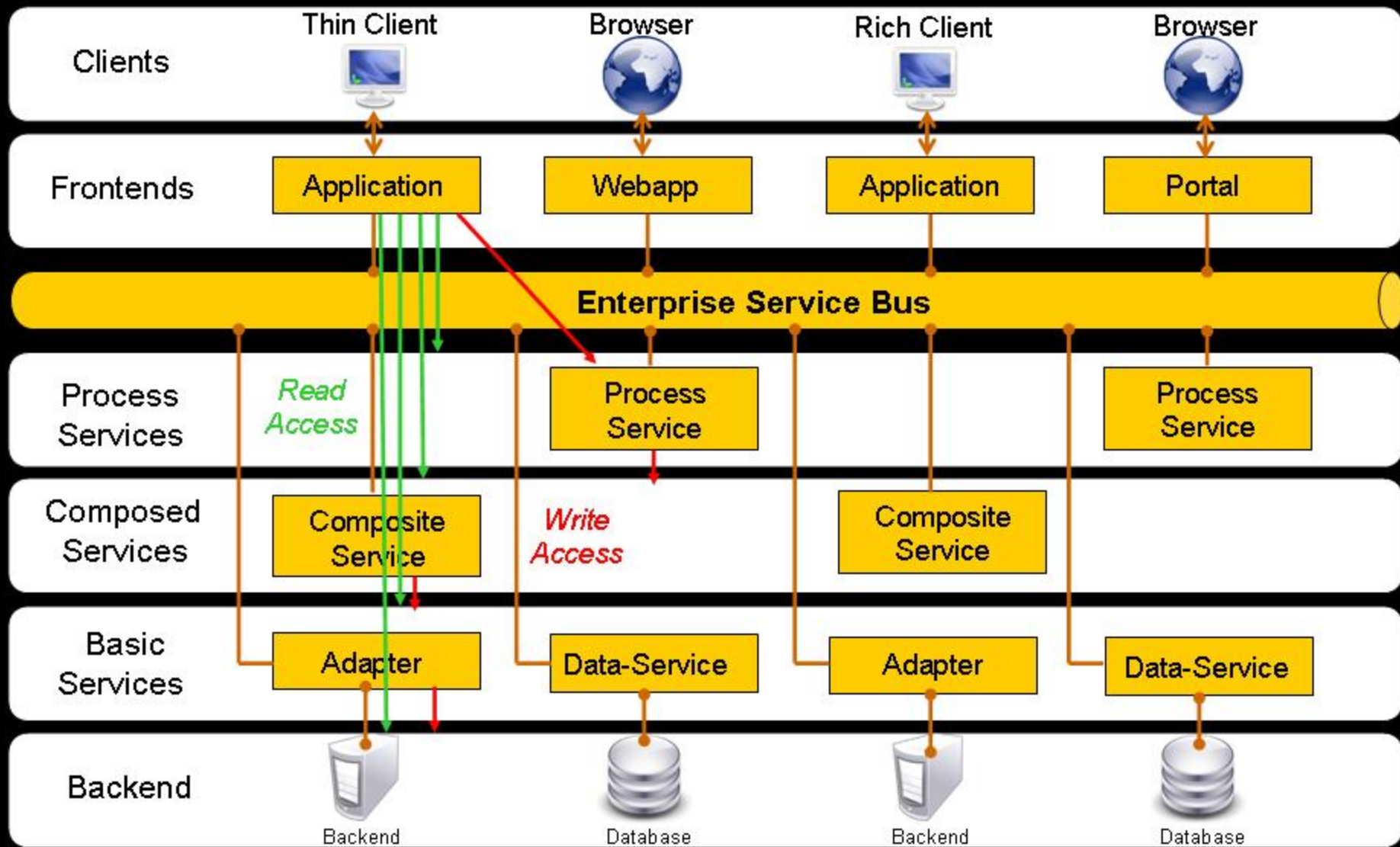
# Layering of services



That way it's possible to:

- ✓ define communication rules
- ✓ automatically validate the architecture
- ✓ define different views
- ✓ manage services by category
- ✓ etc.

for example...





From the Hub-and-Spoke EAI projects  
Mike and Tom learned:

**The idea of a common  
business object model  
is not working!**

- ✗ very complex
- ✗ many different views
- ✗ many stakeholders
- ✗ big communication overhead

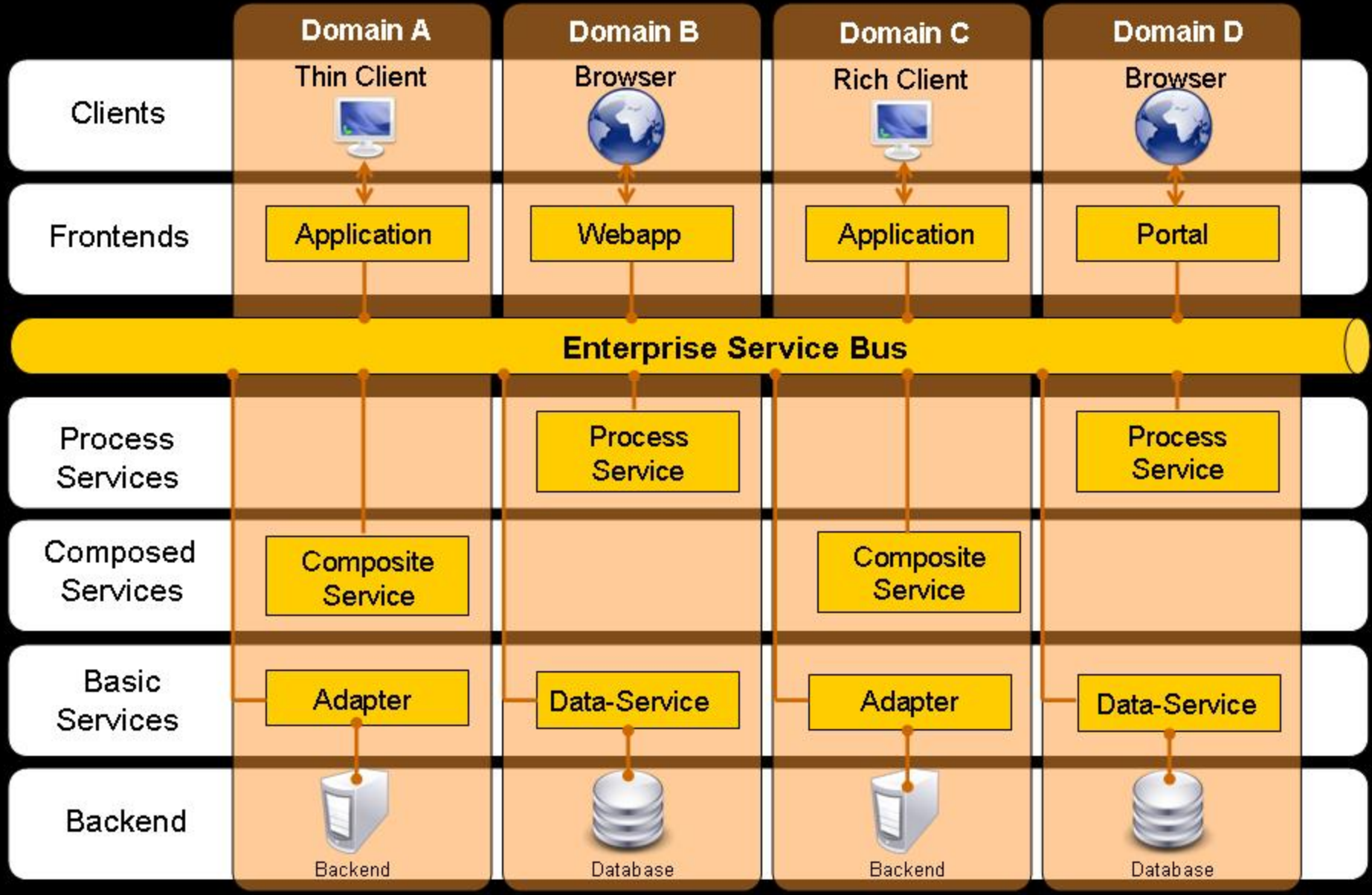
But, ...



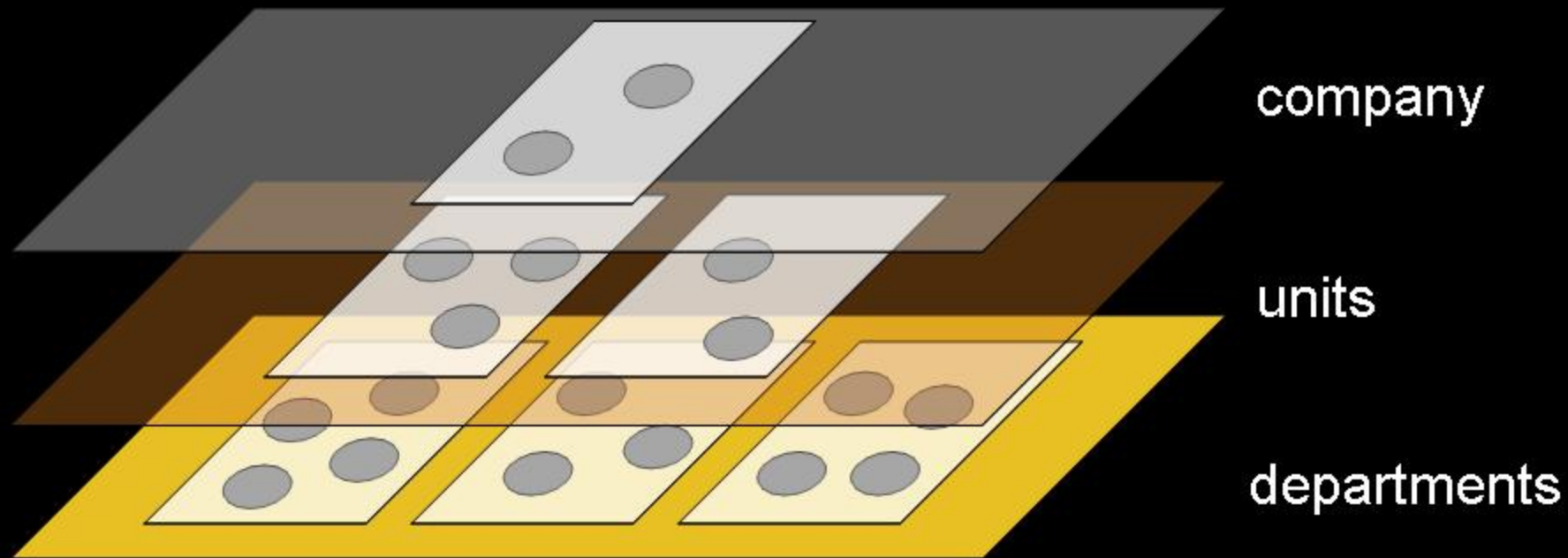
*“Divide and conquer!”*

-- King Ludwig XI.

# Structure the services in domains:



You can structure domains **hierarchically**...



...and delegate the **responsibilities**...

...according to your **organisation structure**.





Another important issue in SOA is  
**versioning of services**

## Simplest versioning strategy:

**„Every change is a new service!“**

- + easy implementation for service provider
- quantity of services grows rapidly
- change on consumer side with every new version (even on compatible changes)

## compatible changes

- ✓ adding an operation
- ✓ adding an optional parameter
- ✓ adding a new datatype



**Minor  
Release**

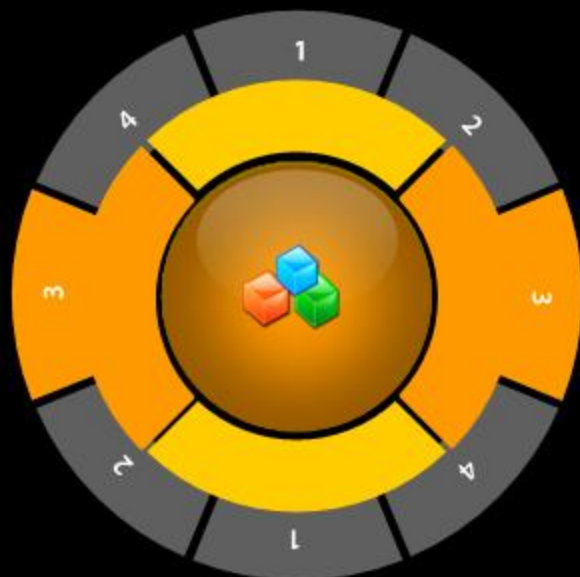
## incompatible changes

- ✗ removing an operation, parameter or datatype
- ✗ renaming an operation, parameter or datatype
- ✗ changing order in a sequence of parameters  
(and anything else...)



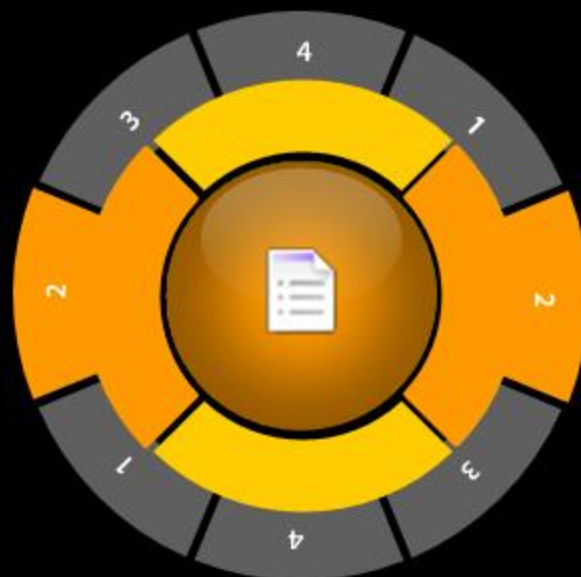
**Major  
Release**

Mike and Tom defined a **versioning strategy** for the services:



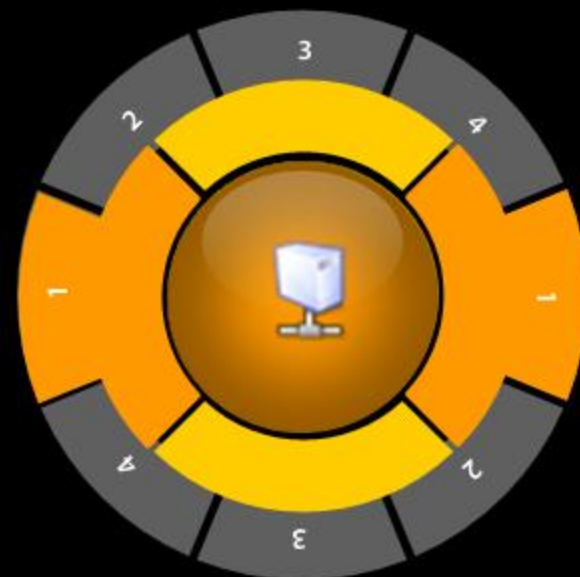
### Granularity

- 1 = no versioning
- 2 = on operation level
- 3 = on service level



### Version Info

- 1 = as parameter
- 2 = in element name
- 3 = in namespace



### Deployment

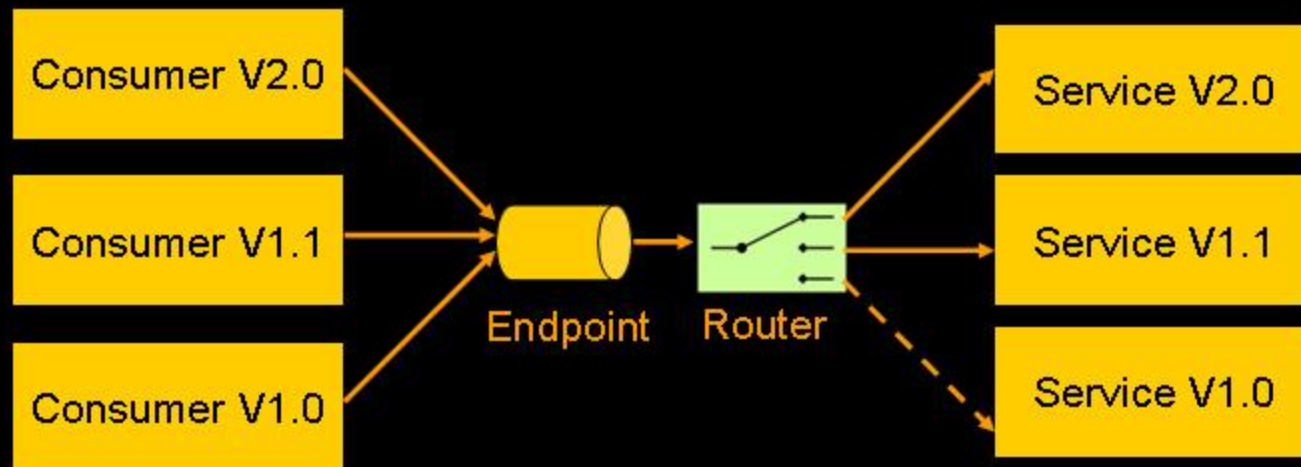
- 1 = one endpoint for every major version
- 2 = one endpoint for every new version
- 3 = single endpoint for all versions



Let's take a closer look on...

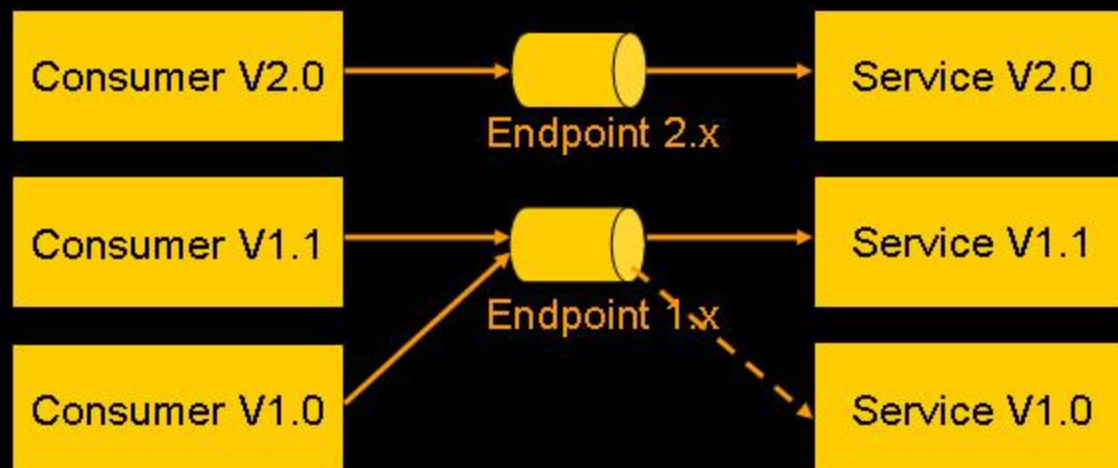
... **deployment aspects.**

# One single endpoint for **all** versions...



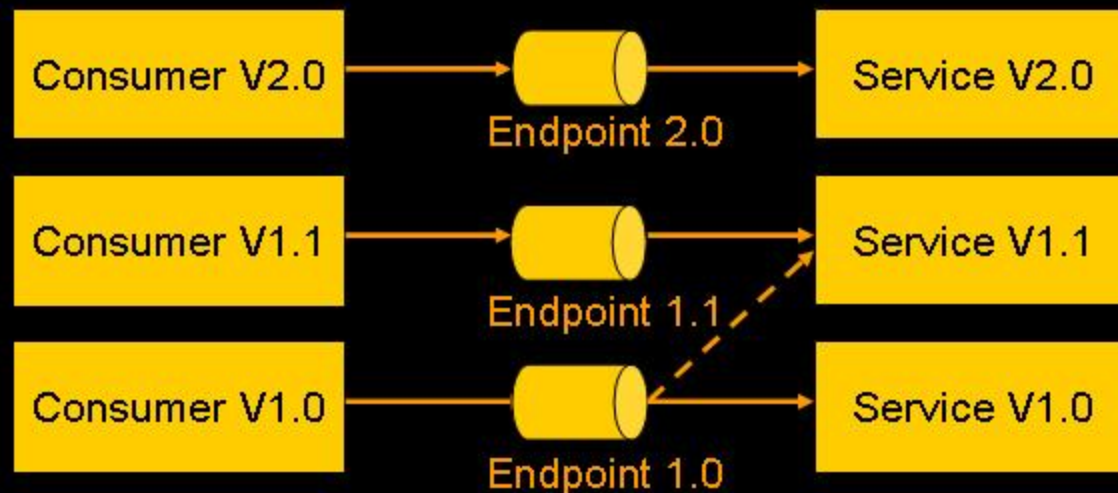
- + consumer knows only one endpoint
- implementation of a router necessary
- errors in new versions or router can affect old versions

# One endpoint for **every major** service version...



- + no modifications on old incompatible services
- + implementation of a router not necessary
- errors in minor versions can affect old versions

# One endpoint for **every new** service version...



- + no modifications on old services
- + implementation of a router not necessary
- management of many endpoints
- framework must support multiple bindings





# **ObjectSPEKTRUM 02/2009**

article about service versioning

## Best Practices on one slide:

- ✔ Provide a vision and roadmap.
- ✔ Develop a reference architecture.
- ✔ Provide an messaging middleware and reduce dependencies between services.
- ✔ Keep in mind that people are not easy to change.
- ✔ Diskuss pradigm changes and talk about „bad habbits“.
- ✔ Establish a SOA team and advertise about it in other projects.
- ✔ Collect patterns, best practices and develop a meta model.
- ✔ Define service lifecycle, roles and responsibilities.
- ✔ Diskuss choreography vs. orchestration and transaction vs. compensation.
- ✔ Adapt good architecture concepts from other technologies.
- ✔ Layer and categorize your services.
- ✔ Define and structure your domain model.
- ✔ Create and publish a versioning strategy for your services.

**Two more things...**



Maybe the most important  
thing Mike and Tom learned is:

**Integrating business processes  
is all about people!**



**People fear change!**





People need  
**motivation**

People need  
**inspiration**



People need  
**leadership**



Meet **Bob**...



...he is Mike's **boss**!





Bob is happy!



Bob is happy!  
**Why?**

Thanks to Mike and Tom...

...his business is **running great!**





... **ask** Bob why SOA?



He will say he hates **buzz words**.

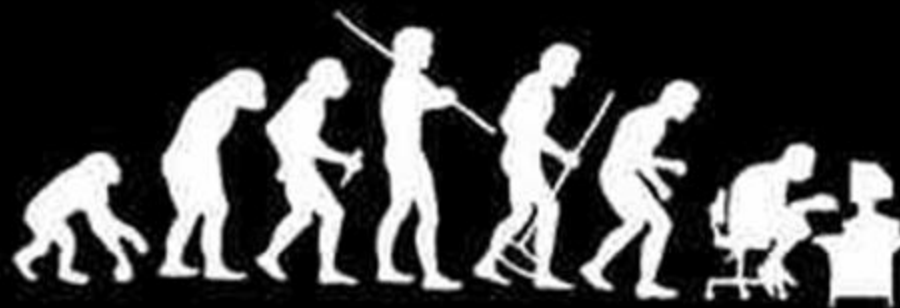


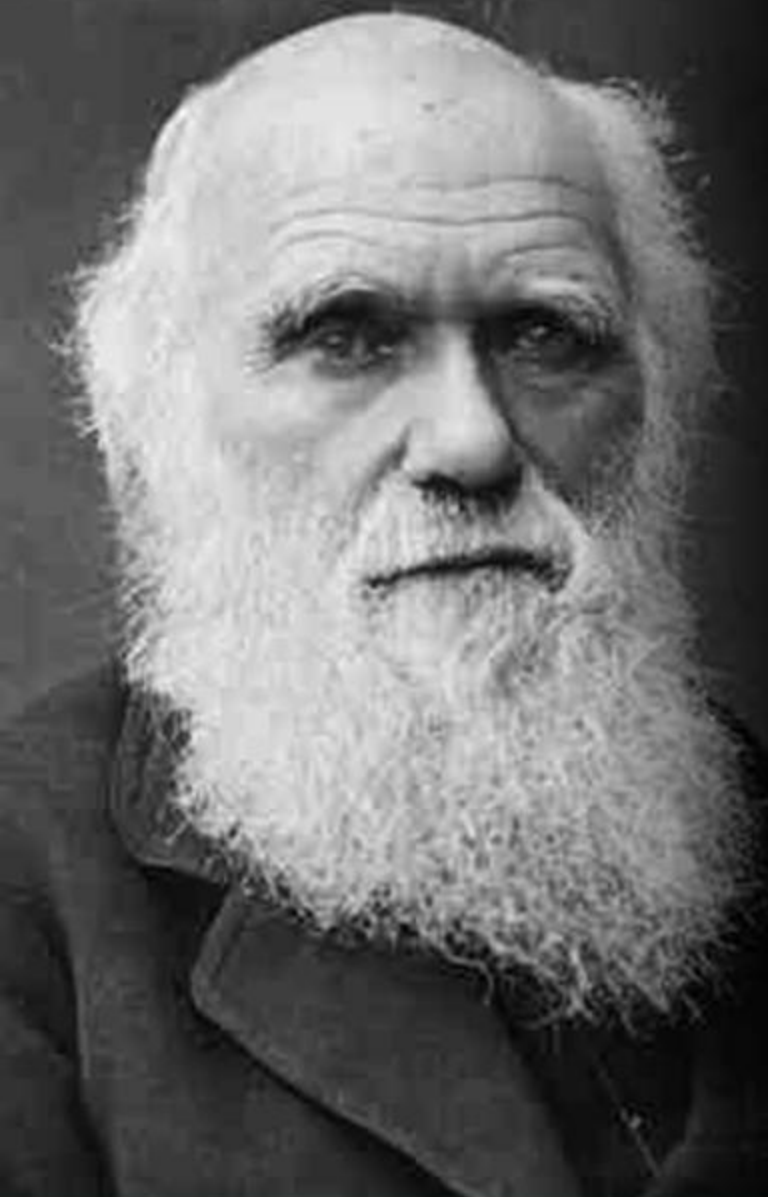
„Some people may say... ROI

...it's all about...

...**maximizing ROI**.“

„But Bob is investing in SOA because...  
...he believes in **evolution theory**.“





“ It is not the strongest of the species that *survive*, nor the most intelligent, but the one most *responsive to change*. ”

-- Charles Dawin



This species was not flexible enough!

His company **invests** in SOA to keep **flexible**.





Business process integration...  
... is a **long journey!**




Many companies decided...  
...to make their journey...



... **with us!**

THANK YOU!

Slide design was inspired by:  
<http://www.ethos3.com>  
<http://www.presentationzen.com>



Blog

<http://www.eduard-hildebrandt.de>



[www.logica.com/de](http://www.logica.com/de)

Releasing your potential

# Logica

Logica is a leading IT and business services company, employing 39,000 people across 36 countries. It provides business consulting, systems integration, and IT and business process outsourcing services. Logica works closely with its customers to release their potential – enabling change that increases their efficiency, accelerates growth and manages risk. It applies its deep industry knowledge, technical excellence and global delivery expertise to help its customers build leadership positions in their markets. Logica is listed on both the London Stock Exchange and Euronext (Amsterdam) (LSE: LOG; Euronext: LOG). More information is available at [www.logica.com](http://www.logica.com).