5.– 8. September 2011
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Leichter ohne Bus

Alternative Ansätze für Integration

Eberhard Wolff

adesso AG

adesso business. people. technology.

Leichter ohne Bus

Alternative Ansätze für Integration

Eberhard Wolff

Architecture and Technology Manager

adesso AG

08.09.11

About me

- Eberhard Wolff
- Architecture & Technology Manager at adesso
- adesso is a leading IT consultancy in Germany
- Speaker
- Author (i.e. first German Spring book)

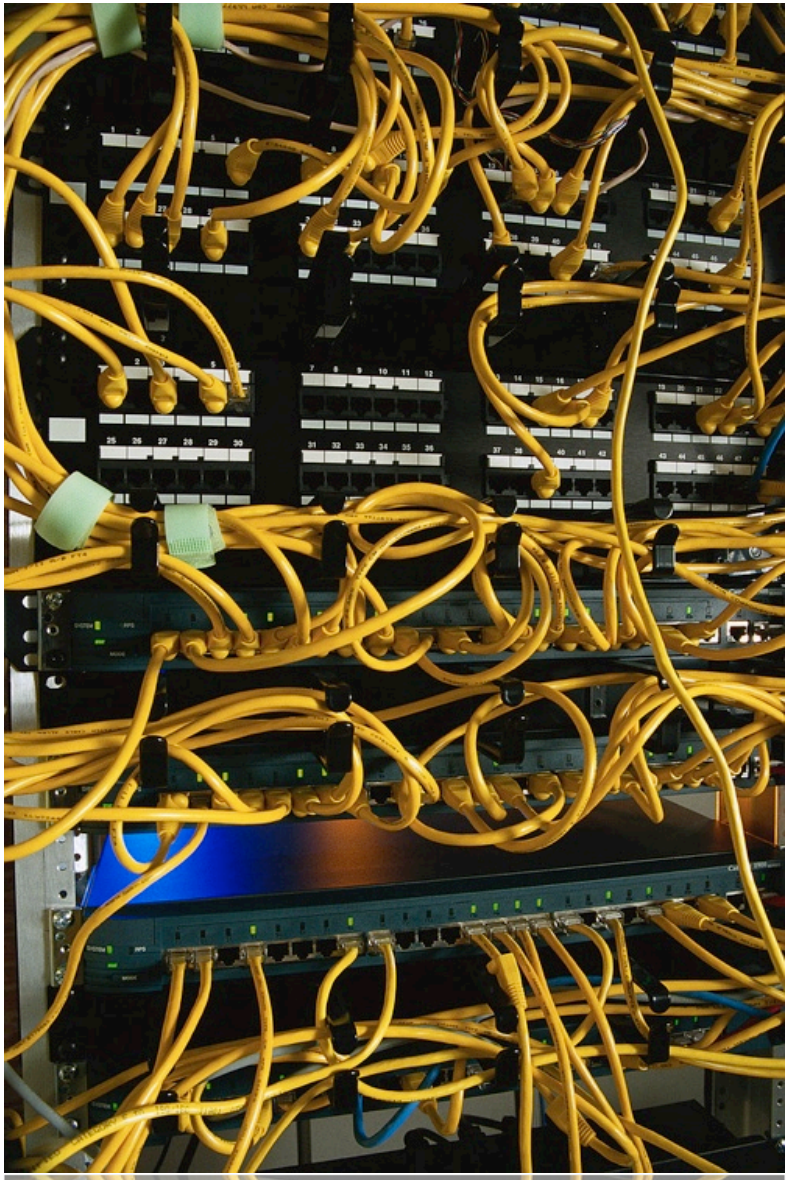
- Blog: http://ewolff.com
- Twitter: @ewolff
- http://www.slideshare.net/ewolff
- eberhard.wolff@adesso.de

- We are hiring

adesso business. people. technology.

Enterprise Service Bus

- A piece of software
- To integrate several systems
- Talks several network protocols and application protocol
- Data transformation
- Orchestration (Process Engine)
- Service Registry
- The Final Solution For All Problems Integration Related ™

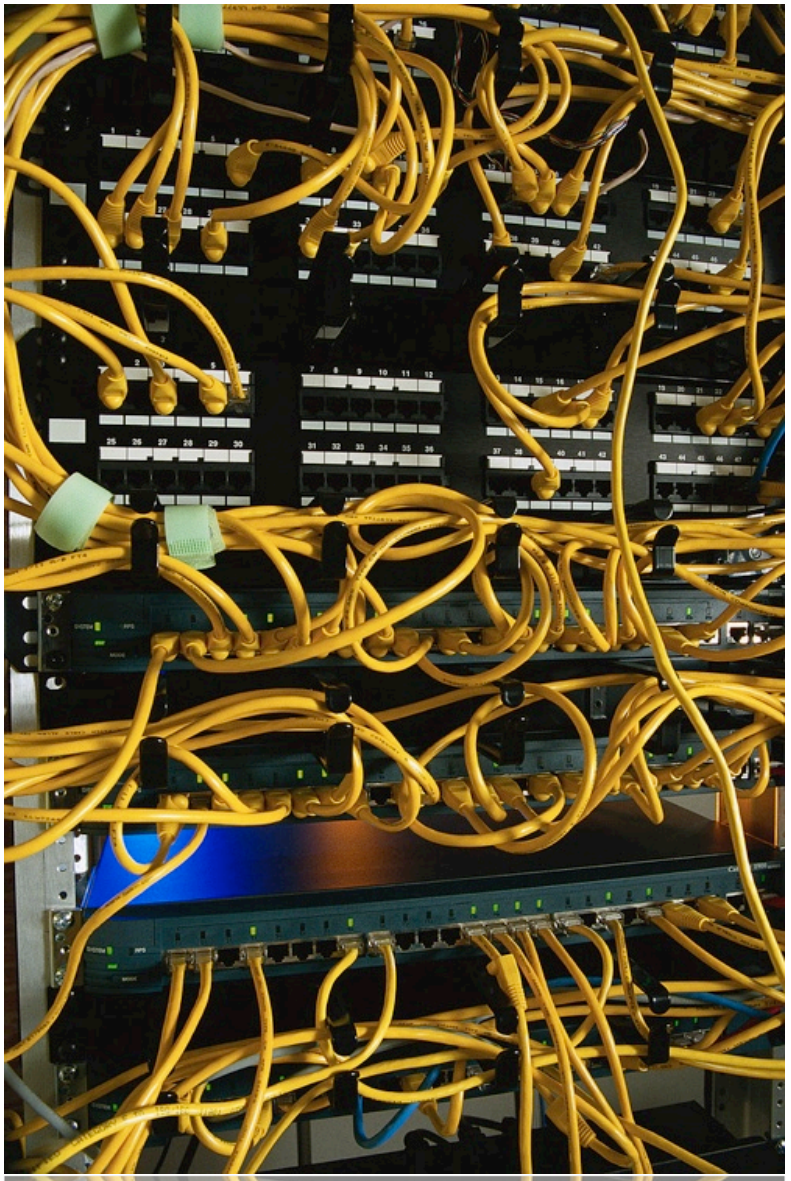


Part

- Asynchronous communication
- Decoupling: Events, not methods
- i.e. “order arrived”, not “create invoice”
- Decoupling concerning time: Can / will be handled later
- High reliability: Store-and-forward

Why Enterprise Service Bus?

- The Final Solution For All Problems Integration Related ™
- Sometimes a strategic Decision
- That is:
  A decision that has no justification by itself
- Seemingly easy solution to a complex problem
- Safe bet: lots of large providers
- The other CIOs have one, too!



Challenges: Common Data Model

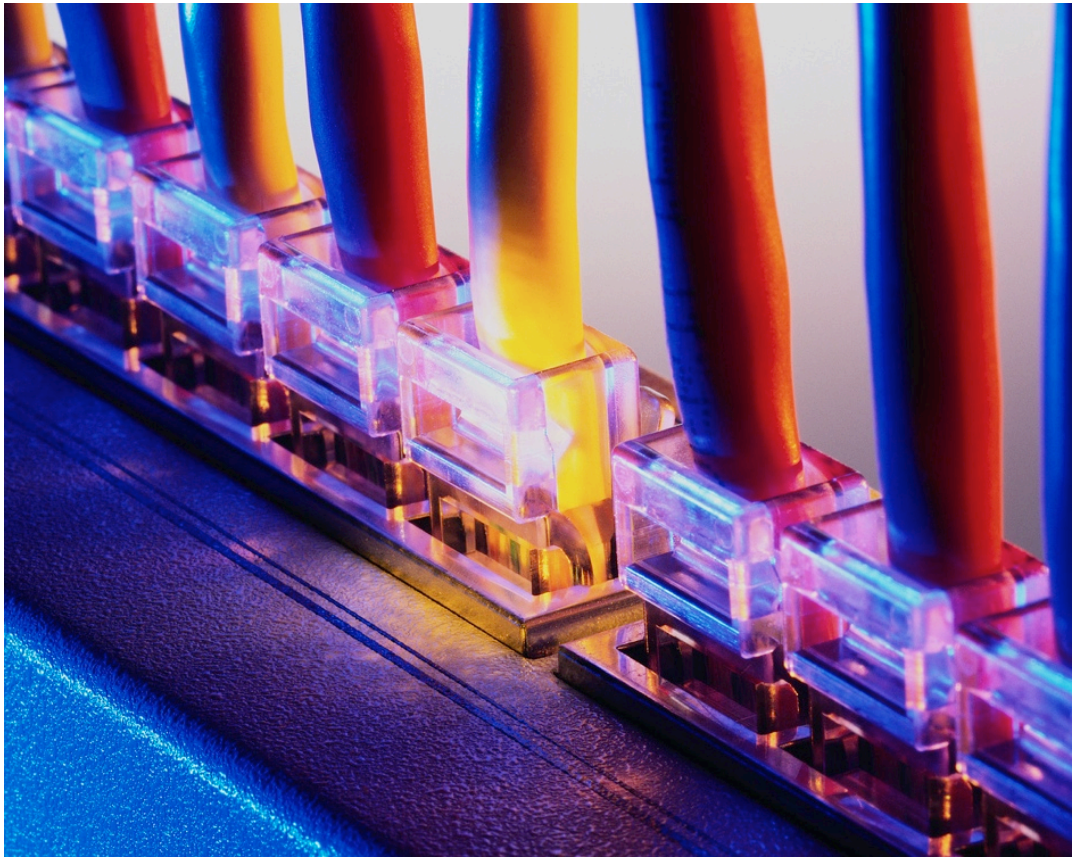
- Services must have a common data model
- But: Services might have different views on common data
- …and need different parts

- Common model might not even make sense

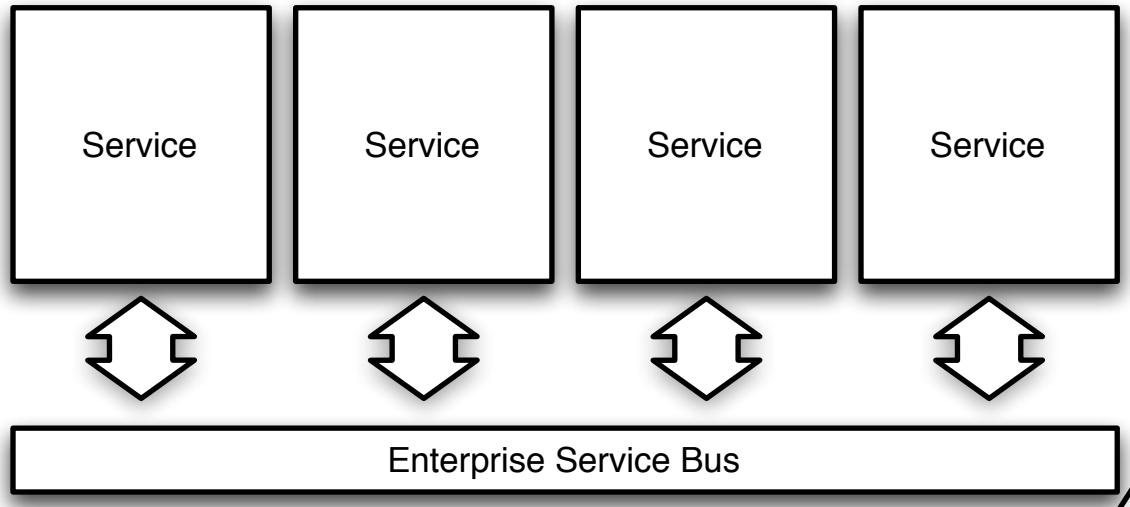
- Example:
  - Customer data for the delivery of an order
  - vs. customer data for the payment

Challenges: Network

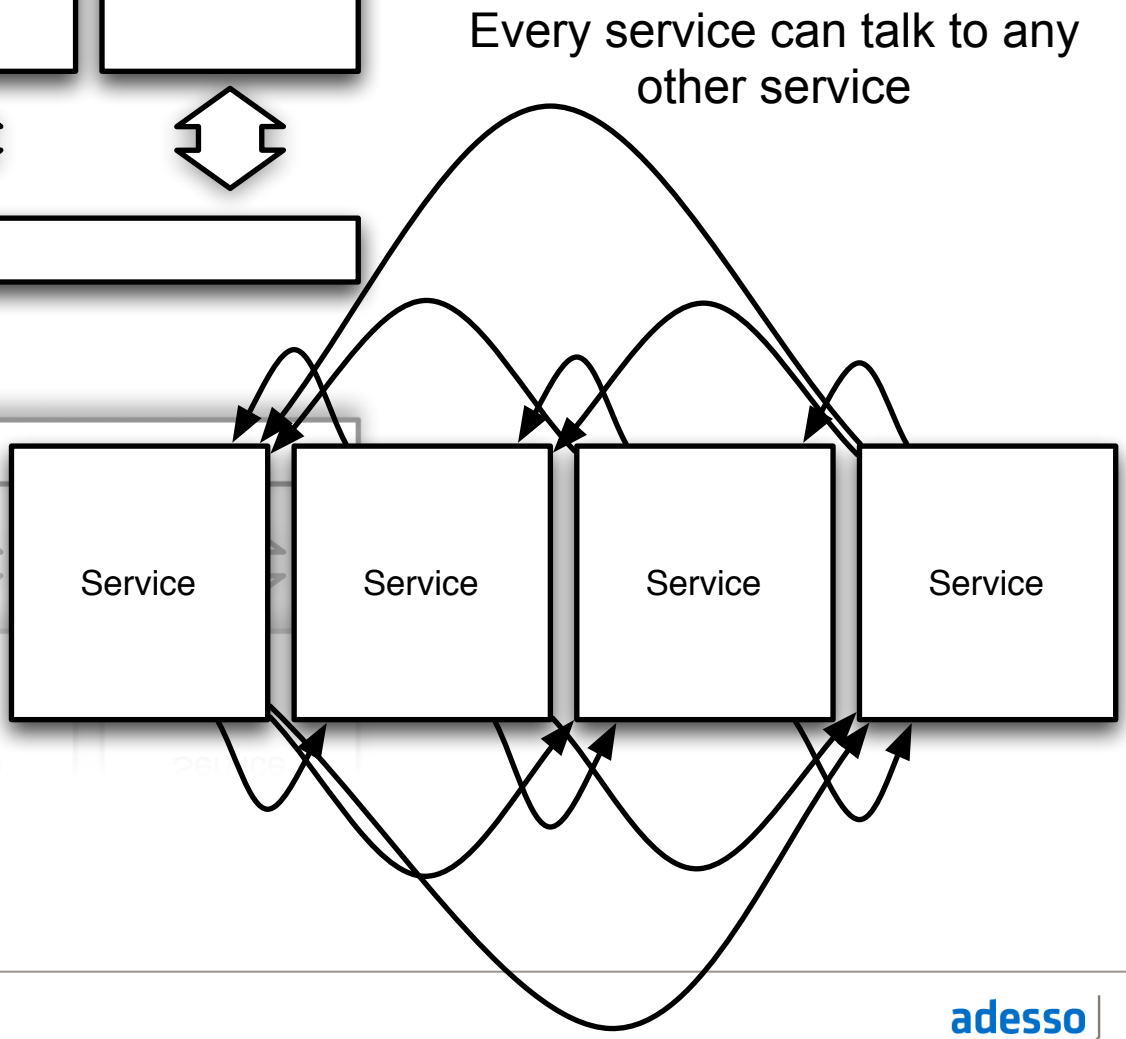
- Typically a networked services
- Doesn’t conform to the First Rule of Distributed Objects
  “Don’t Distribute Your Objects!”
- Latency / throughput
- Centralized communication infrastructure – scaling?



Challenge: Dependency Chaos



Technical view



adesso

Solution To the Dependency Chaos

- Note: Dependency chaos makes it actually harder to change anything!
- Note: Dependency management is the foundation of all software architecture!

- Dependencies on Business Events, not services

- I.e. “New order arrived”

- Component might create an invoice, start the fulfillment etc.

- Easy to add new functions (e.g. bonus program)

Challenges: Architecture

- Monolithic
  - Many services
  - All paid for
- Complex
  - 40 different Web Services standards
  - Countless application protocols

http://www.thinkgeek.com/geektoys/collectibles/e1e0/

adesso

ESB And SOA

- ESB are considered an enabler for SOA

- Service Oriented Architecture
- Deconstruct IT into several services
- Enable easier customization and creation of new services by orchestration

- But:
  - SOA creates no value by itself, just better agility
  - Hard to do: completely change the IT in your enterprise
  - So: Low and slow ROI

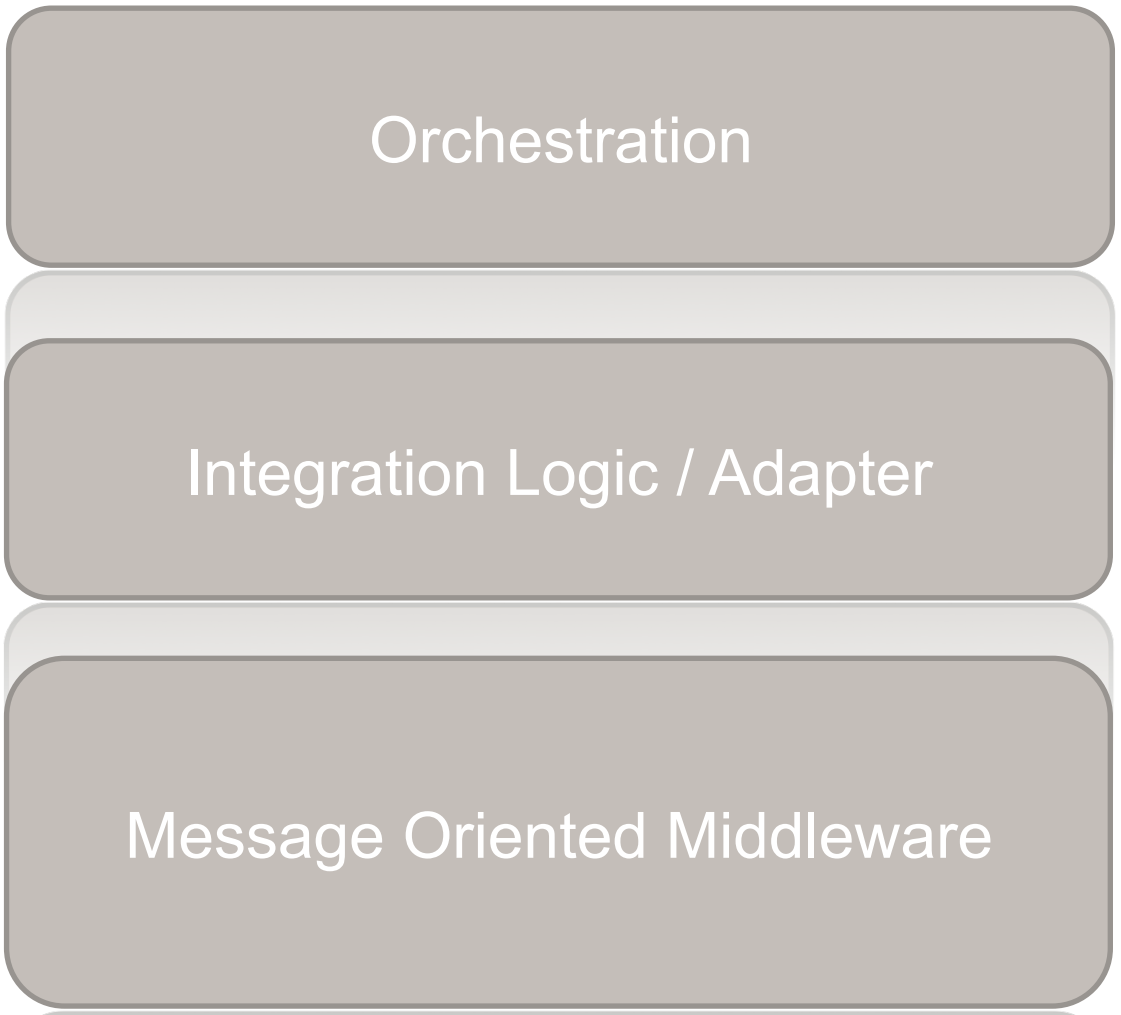
- SOA as an architectural pattern still useful
- But: Stay away from completely redoing your IT!
- http://apsblog.burtongroup.com/2009/01/soa-is-dead-long-live-services.html

Enterprise Service Bus Is No Technology

- ESB is an integration *architecture*
- Not a specific product!

- I.e. create your own technology stack!

My Own ESB!



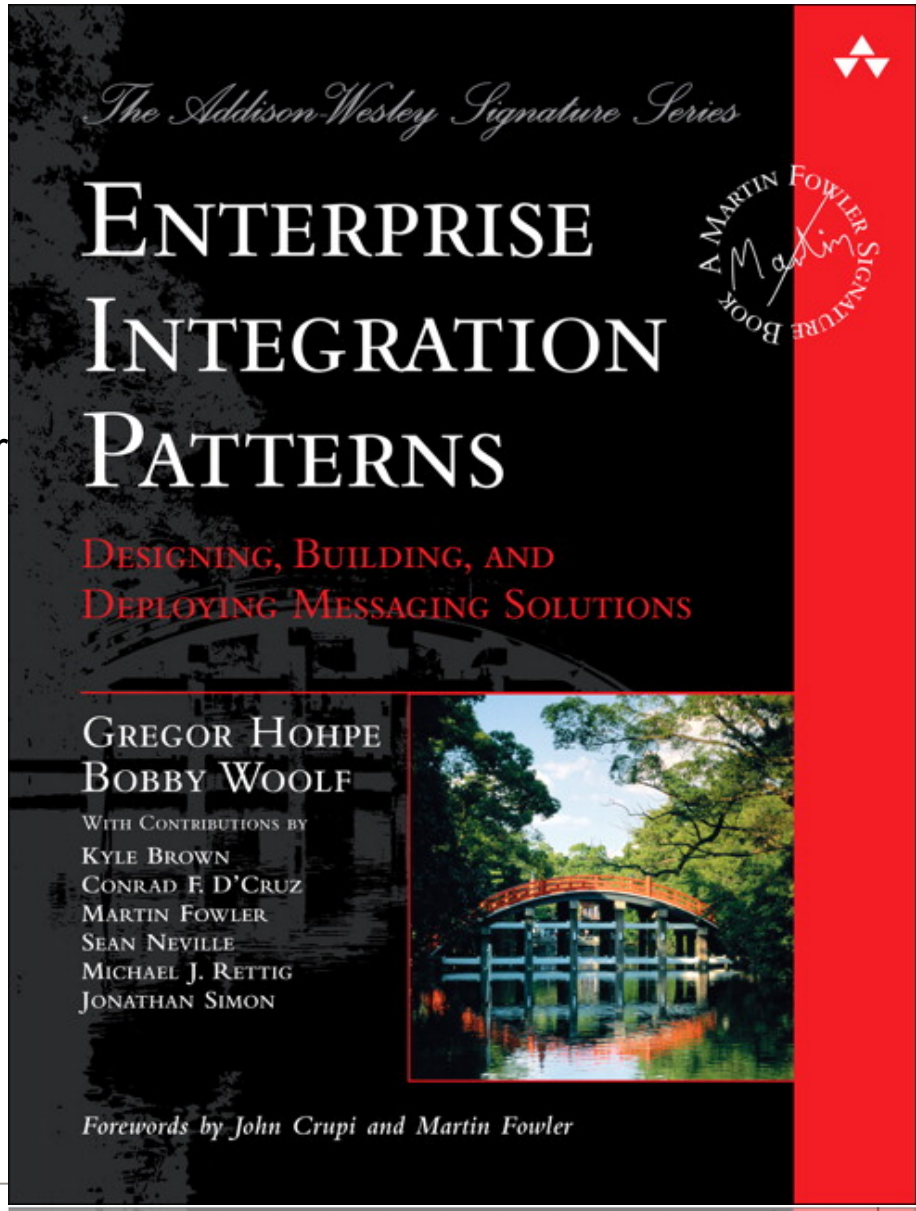
- jBPM
- Activiti
- Scripting (Groovy) (?)

- Spring Integration
- Apache Camel

- Apache ActiveMQ
- IBM WebSphere MQ
- SonicMQ
- HornetQ (JBoss)
- RabbitMQ
- Apache Qpid

Nice Guideline

- Enterprise Integration Patterns
- By Gregor Hohpe und Bobby Woolf
- http://www.eaipatterns.com/
- Contains Patterns like Router, Translator or Adapter
- Direct support in Spring Integration and Apache Camel
- Good reference for more inspiration



Integration Is Asynchronous?

- Enterprise Integration Patterns are really asynchronous patterns
- Sometimes integration is really a batch
- I.e. transfer today's order for fulfillment tomorrow
- Might use Spring Batch instead
- + cron / Quartz as scheduler
- What about completely different approaches for integration?

REST

- Integration using completely different approaches
- REST: Representational State Transfer
- Clients and server
- Request and Responses
- Representation of the state of a resource
- i.e. a document (XML, JSON…)

- Example: HTTP
- Resources addressed by URLs
  http://adesso.de/customer/42
- Requests: GET, POST, PUT, DELETE, HEAD
- Representation with accept header or
  file extension http://adesso.de/customer/42.xml

REST & Reliability

- Safe i.e. no side effects:
  GET, OPTIONS
- Idempotent i.e. can be done multiple times without changing the state:
  PUT, DELETE, GET, HEAD
- Only problem: POST
- Multiple request might result in multiple new resources
- But why bother with REST for integration?

REST & Integration

- Built in support for multiple representations
- i.e. each client can request its own data format

- Libraries for any programming language and platform

- URLs allow globally unique identifier
- Easy to refer to data stored on a different system
- i.e. http://adesso.de/order/42 might link to http://adesso.de/customer/17
- Easy to have different system handle different URLs

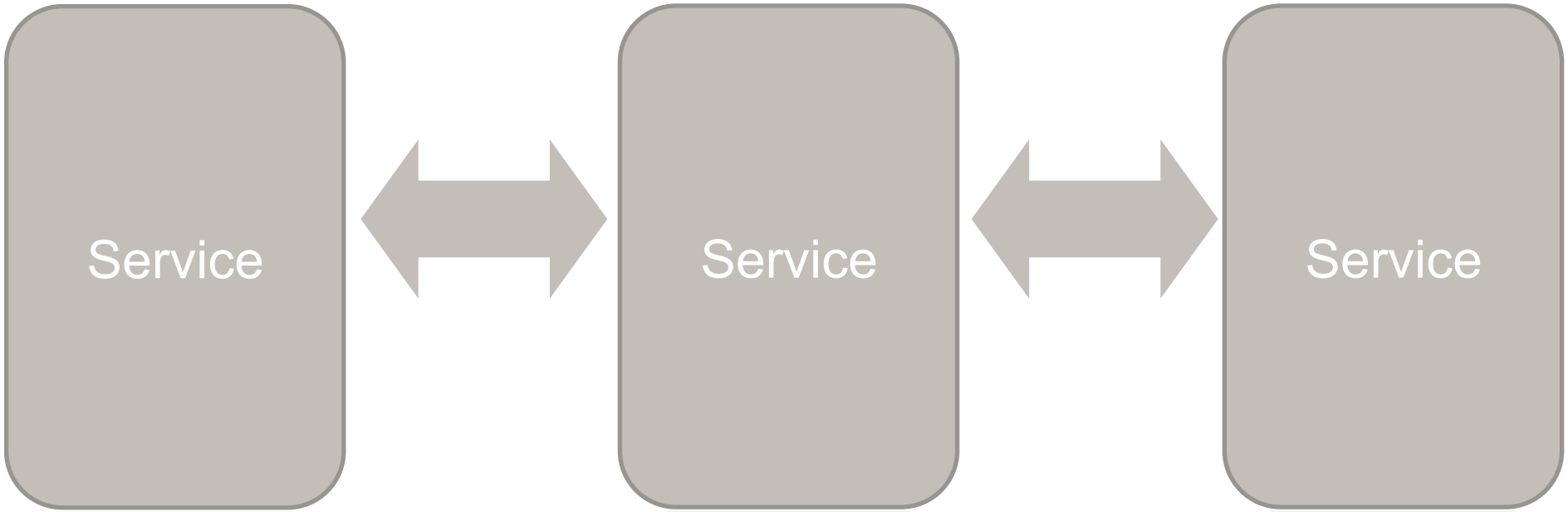
- Foundation for largest integration project known (i.e. World Wide Web)

- Lots of optimizations (i.e. caching)

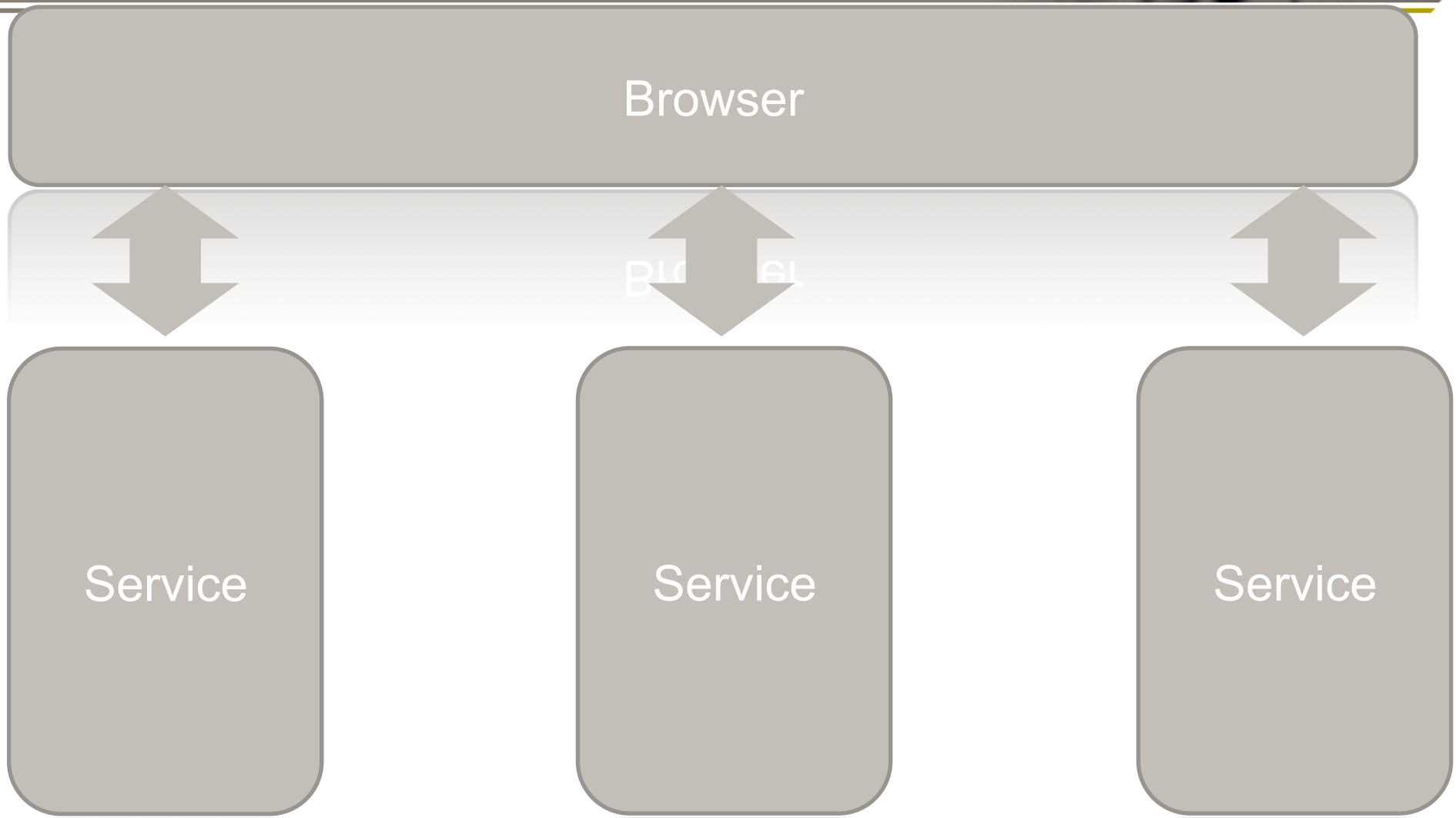
Ein Buch!

- Stefan Tilkov
- REST und HTTP
  Einsatz der Architektur des Web für Integrationsszenarien
- dpunkt

So Far



Client Side Integration



Client Side Integration

- Mash Ups
  - Pictures from Flickr
  - Maps from Google Maps
  - Videos from YouTube

- Reuses ideas from the Web for Enterprise applications

- i.e. show orders for the customer selected in a different application

- Easiest way: Pass in the customer id in the link
- Might consider REST
- Link handled by different application

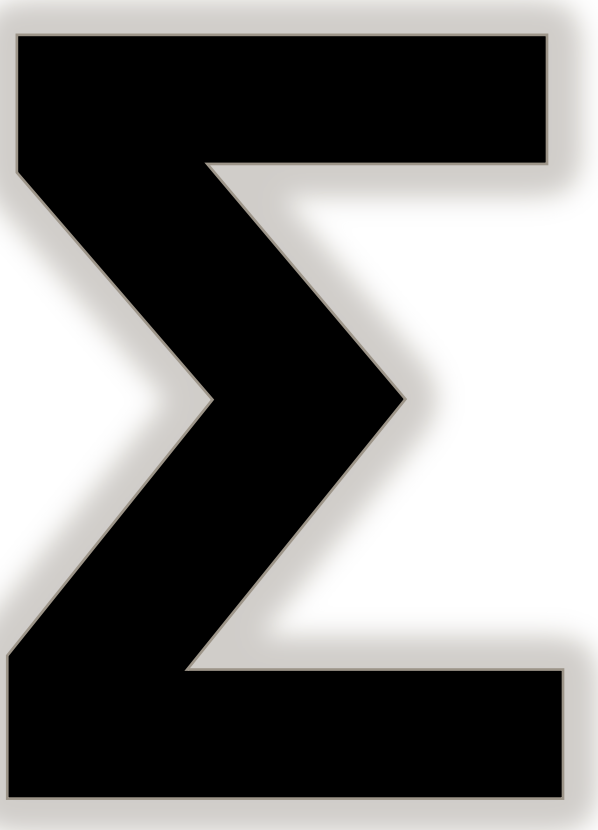
Client Side Integration

- More complex: Integration logic in JavaScript
- I.e. store the customer id and access it using JavaScript
- Automate across applications

- More examples
  - Google Apps Scripts allows for automation of Google App
  - Views and queries in CouchDB (NoSQL)
- More and more services offer JSON (JavaScript Object Notation)
- JavaScript can talk directly to a backend (Web Sockets)

Sum Up

- Enterprise Bus is a Pattern, not a product
- Use with care
  - Performance considerations
  - Dependency hell
  - Consider your creating a customized stack
  - SOA's promises are hard to reach
- REST and HTTP might be a lightweight and practical alternative
- Client side integration allow a completely different perspective
- Learn from the cool Web kids!
- Know the alternatives and use the best tool for the job!



adesso business. people. technology.

Wir suchen Sie als

- Software Architekt (m/w)
- Projektleiter (m/w)
- Senior Software Engineer (m/w)

jobs@adesso.de
www.AAAjobs.de