Enterprise Application Integration patterns for Java EE cloud applications

JavaOne 2012

Alexander Heusingfeld

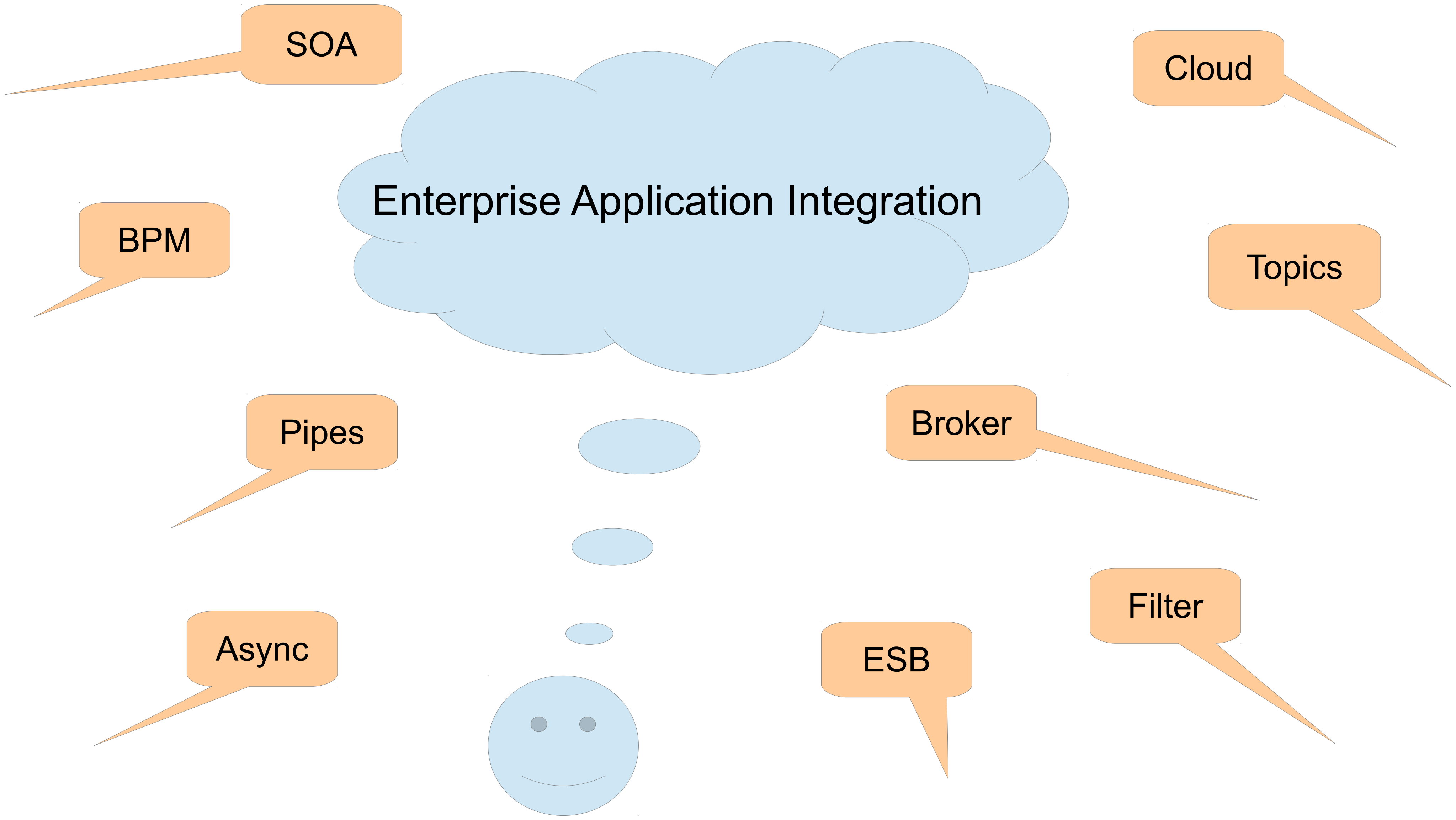
Stefan Reuter

Speakers

- Alexander Heusingfeld
  - Senior Consultant, Cyber:con GmbH
  - @goldstift, aheusingfeld(at)firstpoint.de

- Stefan Reuter
  - Software Architect, Freelancer
  - @stefanreuter, stefan.reuter(at)reucon.com

Handling the buzz



Enterprise Application Integration

"Enterprise application integration (EAI) is def ned as the use of software and computer systems architectural principles to integrate a set of enterprise computer applications."

- Wikipedia

Enterprise Application Integration

- Process of linking 'information silos' via
  - Mediation – between multiple applications
  - Federation – providing access to the 'outside world'

- Best Practice: asynchronous messaging architectures

Ok, it's all about asynchronous messaging

- one message per delivery
- messages can be queued
- exchangeable transport

EAI in the real world

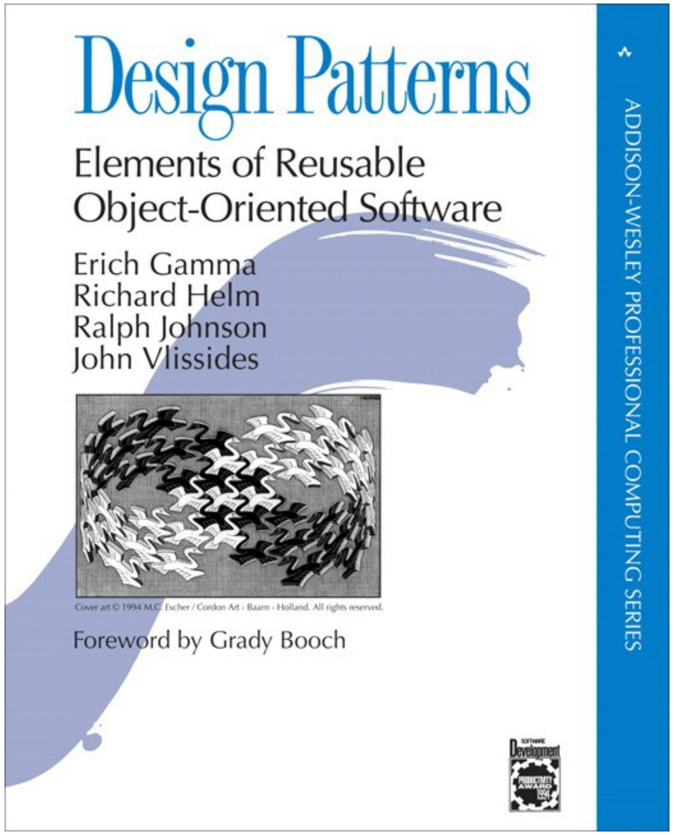
As every application has a kind of mailbox imagine a …

… **reliable** postal service

Benefits of using messaging

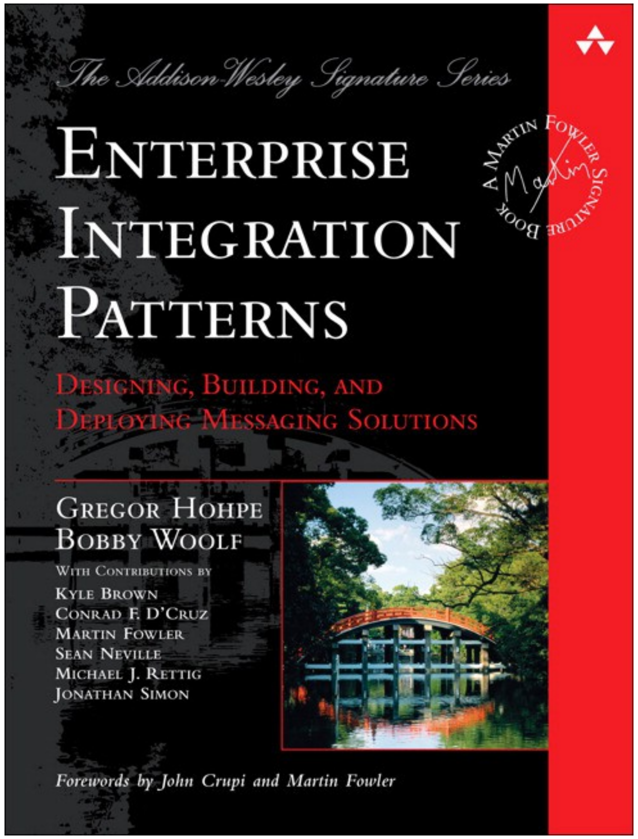
- message-based communication allows decoupling
- integrate heterogenous platforms/ languages
- variable timing & throttling – every app at its pace
- reliable communication
- disconnected operation

Patterns applied to EAI


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Design Patterns (Gamma et al)

Proven solutions for common problems


The Addison Wesley Signature Series
Enterprise Integration Patterns
Designing, Building, and Deploying Messaging Solutions
Gregor Hohpe
Bobby Woolf
With Contributions by
Kyle Brown
Conrad F. D'Cruz
Martin Fowler
Sean Neville
Michael J. Rettig
Jonathan Simon
Forewords by John Crupi and Martin Fowler
A Martin Fowler Signature Book

EAI patterns

Swiss-army knife of reliable integration solutions

How to use EAI patterns in JavaEE

Multiple approaches possible

- Do-it-yourself by leveraging the JEE 6 APIs
- Use a mediation framework
    - Apache Camel
    - Spring Integration

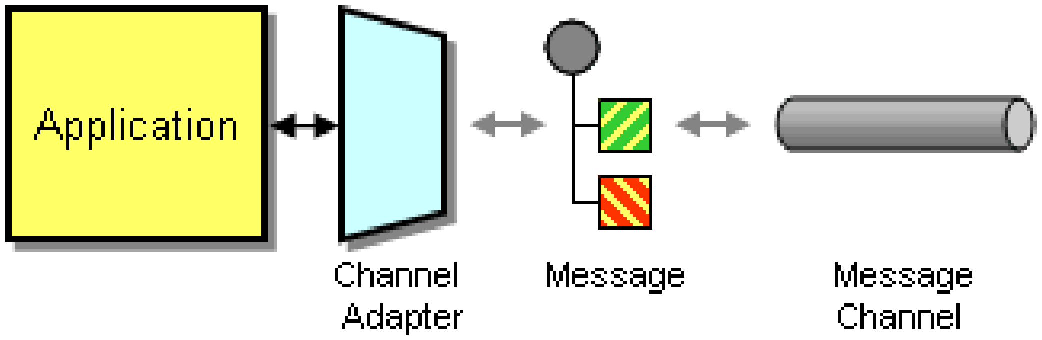
Samples for common EAI patterns

- Adapters and Gateways
- Pipes and filters
- Router
- Transformer
- Splitter & Aggregator

...and many more

EAI Pattern: Inbound- and Outbound-Adapter

- **scenario:** get application's data to another application
  - CRM receives data via HTTP POST → HTTPOutboundAdapter
  - Logistics system sends via Mail → SMTPInboundAdapter

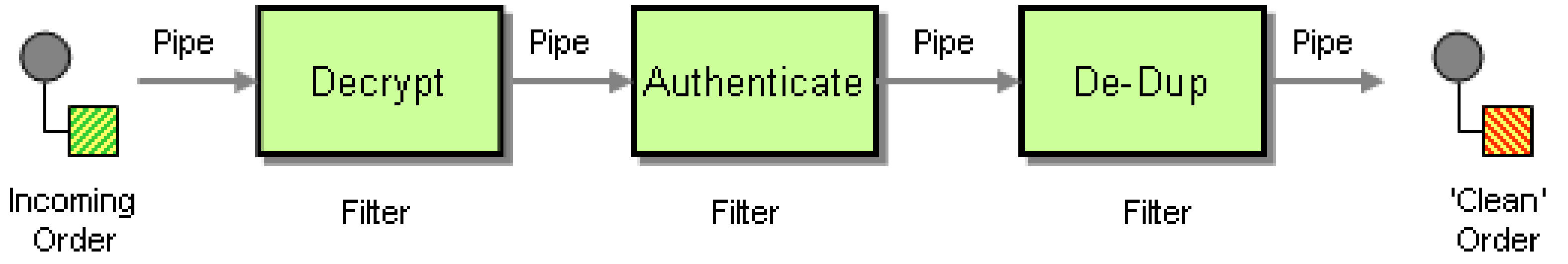


- endpoint fitting the specific capabilities of the communication applications
- NOTE: for synchronous communication use a **gateway**!

http://www.eaipatterns.com/ChannelAdapter.html

EAI Pattern: Pipes and Filters

- **scenario:** new purchase order arrives as a message
  - decrypt encrypted order
  - check for trusted customer
  - check for duplicate order

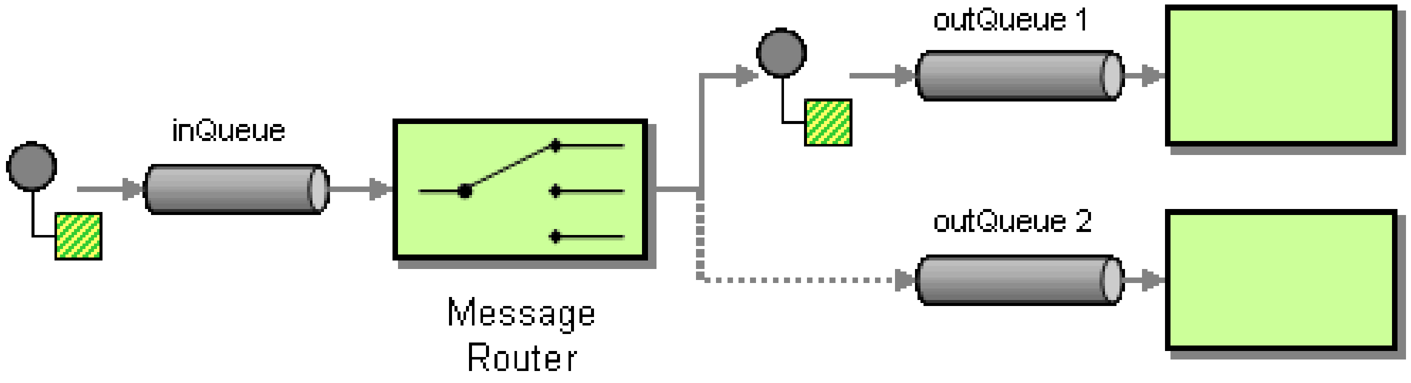


- divide complex processing tasks into smaller steps
- steps (Filters) are connected by channels (Pipes)

http://www.enterpriseintegrationpatterns.com/PipesAndFilters.html

EAI Pattern: Message Router

- **scenario:** decouple processing depending on conditions
  - product type: downloadable software cannot be shipped
  - payment method: credit card transactions vs. direct debit

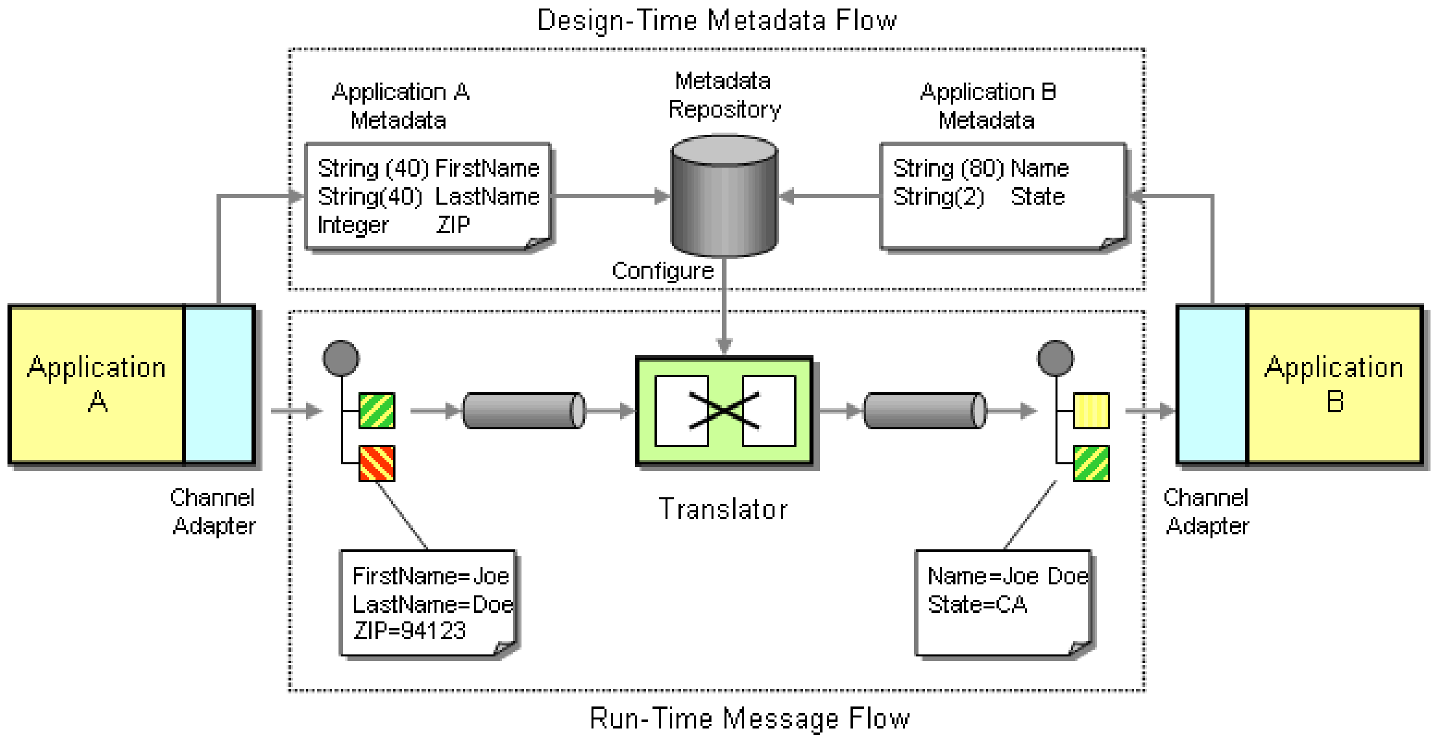


- MessageRouter is a special “filter” which routes but doesn't modify message
- decoupled: surrounding components are unaware of its existence

http://www.enterpriseintegrationpatterns.com/MessageRouter.html

EAI Pattern: Transformer

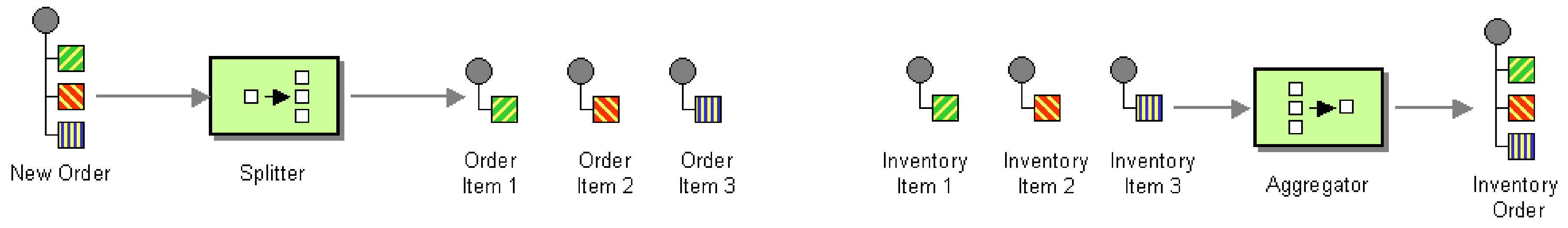
- **scenario:** transfer customer data of the order to CRM
  - fields have different meaning and content
  - fields have different length



http://www.enterpriseintegrationpatterns.com/MessageTransformationIntro.html

EAI Pattern: Splitter and Aggregator

- **scenario:** process order positions separately, aggregate state later
  - send positions with physical products to the warehouse system
  - aggregate order state depending on shipping state of order positions



- Splitter publishes one message for each item of the original message

http://www.enterpriseintegrationpatterns.com/Sequencer.html

- Aggregator collects messages until a set of related messages is complete

http://www.enterpriseintegrationpatterns.com/Aggregator.html

EAI patterns in Spring integration

```
<gateway id="sender" service-
interface="com.github.aheusingfeld.javaone2012.eai.gateways.Sender"
default-request-channel="sample-channel"/>

<channel id="sample-channel"/>

<filter id="simple-filter" input-channel="sample-channel" output-
channel="matching-msg-channel" discard-channel="non-matching-msg-
channel" expression="payload['ORDER_NUMBER'] > 0"/>

<channel id="matching-msg-channel"/>

<outbound-channel-adapter id=”erp-out” channel=”matching-msg-channel”
ref=”erpService” method=”sendOrder”/>

<channel id="non-matching-msg-channel"/>

<logging-channel-adapter channel="non-matching-msg-channel" log-full-
message="true"/>
```

EAI in the cloud

- limited number of I/O gateways → e.g. no filesystem
  - No FTP, no Hot Folder
- mostly no open ports → no basic TCP to custom ports
- scalable but unique endpoints are needed → everything via HTTP
- keep an eye on traffic and network I/O

Q & A

Feel free to

- ask questions now
- contact us on twitter @goldstift & @stefanreuter
- post issues on github:
  https://github.com/aheusingfeld/javaone2012/issues

Thanks for your attention

If you have any questions or need advice afterwards

- contact us on twitter @goldstift & @stefanreuter
- post issues on github: https://github.com/aheusingfeld/javaone2012/issues

Image copyrights

- USB Power adapter courtesy of ChinBuye Limited. http://bit.ly/xBKwGw
- 3newmessages.jpg from http://www.photooutpost.com
- Deutsche Post postboxes by http://bit.ly/PtrhWy
- all pattern graphics are courtesy of Addison Wesley

Backup

Gotchas of EAI in the cloud

- limited number of I/O gateways → e.g. no filesystem
    - No FTP, no Hot Folder
- mostly no open ports → no basic TCP to custom ports
- scalable but unique endpoints are needed → everything via HTTP
- keep an eye on traffic and network I/O