

Enterprise Application Integration patterns for Java EE cloud applications

JavaOne 2012

Stefan Reuter & Alexander Heusingfeld

Speakers

- Stefan Reuter
 - Senior Consultant, freelancer
 - [@stefanreuter](#), stefan.reuter(at)reucon.com
- Alexander Heusingfeld
 - Senior consultant, Cyber:con GmbH
 - [@goldstift](#), alex(at)firstpoint.de

Clearing the context

Enterprise Application Integration?

EAI?

EIP?

What is Enterprise Application Integration?

- most applications need to communicate with others
- Basically: EAI means communicate via “messaging”

Ok, it's all about messaging infrastructure

Like that?



Is there EAI in the real world?

As every application has a kind of mailbox ... imagine a

reliable postal service



What does it mean to use messaging at all?

There are at least three different levels for messaging

- programming model
 - perfect examples are Erlang and Go
- event-based applications
 - see <http://martinfowler.com/eaDev/EventSourcing.html>
- transport mechanism

Benefits of using messaging

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

What are EAI patterns?

- Set of proven solutions for common problems
- Swiss-army knife of adapters



How to use EAI in JavaEE 6

Multiple approaches possible

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

EAI patterns in Spring Integration

- Pipes and filters
- Transformer
- Splitter and Aggregator

Gotchas of EAI in the cloud

- limited number of I/O gateways → e.g. no filesystem
- To preserve scalability, scalable but unique endpoints are needed → HTTP
-

Q & A

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>
-