Enterprise Application Integration patterns for Java EE cloud applications

JavaOne 2012

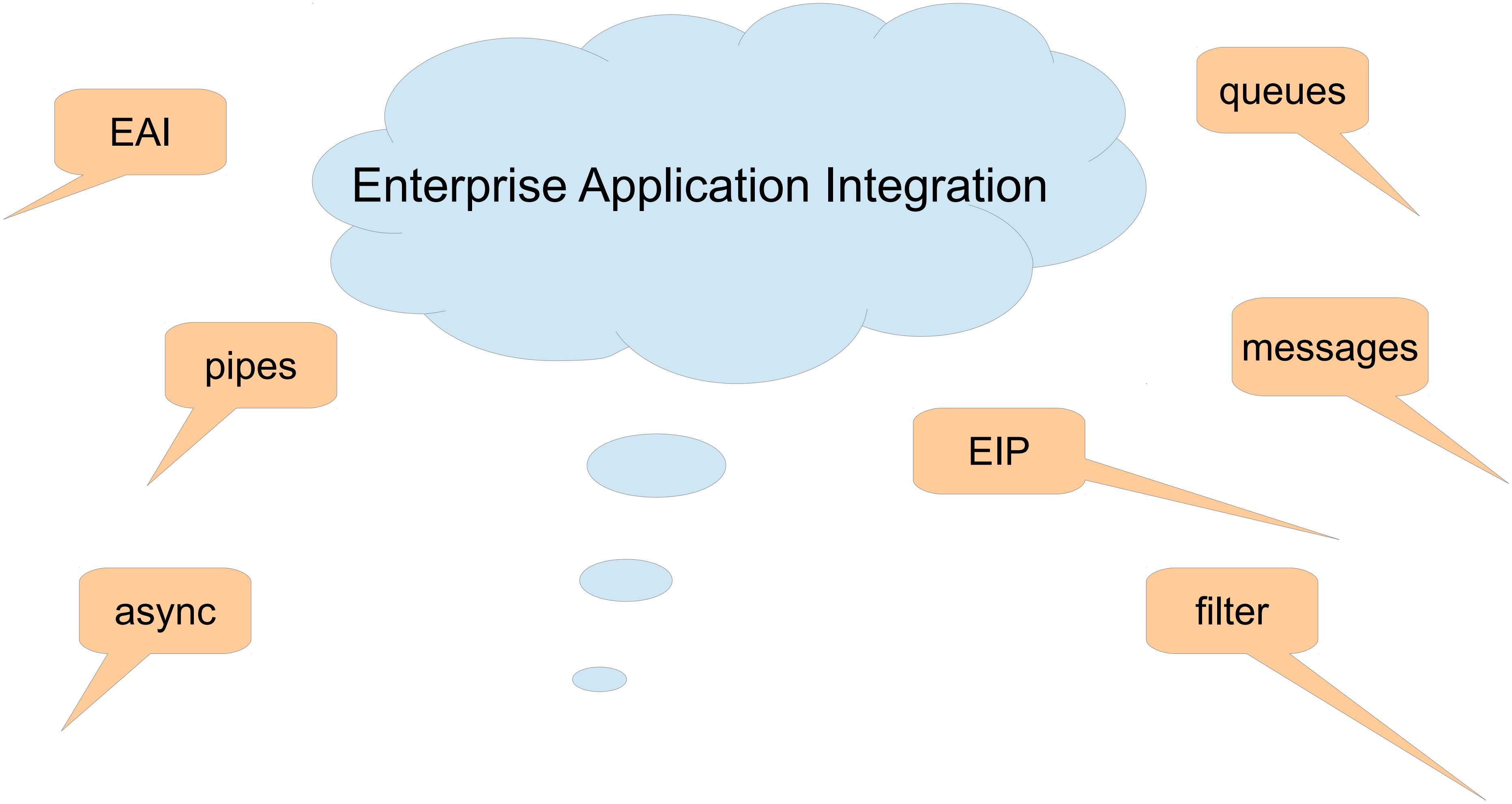
Stefan Reuter & Alexander Heusingfeld

Status: draft

Speakers

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

Clearing the context



Enterprise Application Integration

EAI

queues

pipes

messages

EIP

async

filter

What is Enterprise Application Integration?

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

Ok, it's all about messaging

Like that?

Is there EAI in the real world?

As every application has a kind of mailbox … imagine a

reliable postal service

Benefits of using messaging

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

What are EAI patterns?

Pattern

= proven solution for a common problem

EAI patterns

= Swiss-army knife of reliable integration solutions

How to use EAI in JavaEE 6

Multiple approaches possible

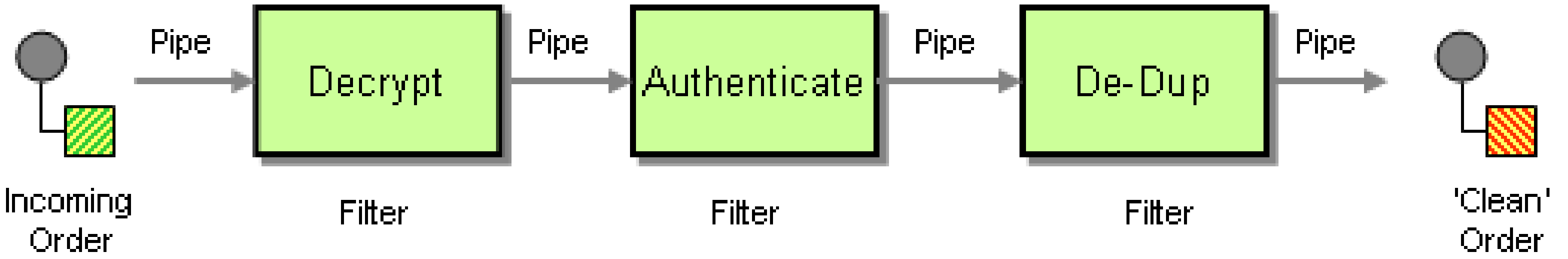
- Do-it-yourself by leveraging the JEE 6 APIs
- Use a mediation framework
  - Apache Camel
  - Spring Integration

EAI patterns in Spring Integration

- Pipes and filters → Code
- Router → Code
- Transformer → Code
- Splitter & Aggregator → Code

EAI Pattern: Pipes and Filters

- scenario: new 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)

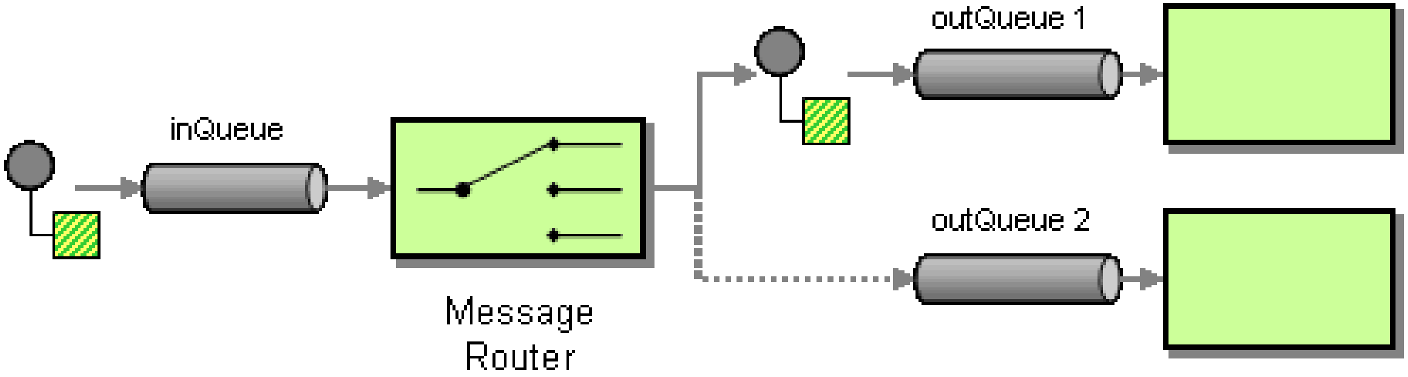
http://www.enterpriseintegrationpatterns.com/PipesAndFilters.html

EAI Pattern: Pipes and Filters

- code sample

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

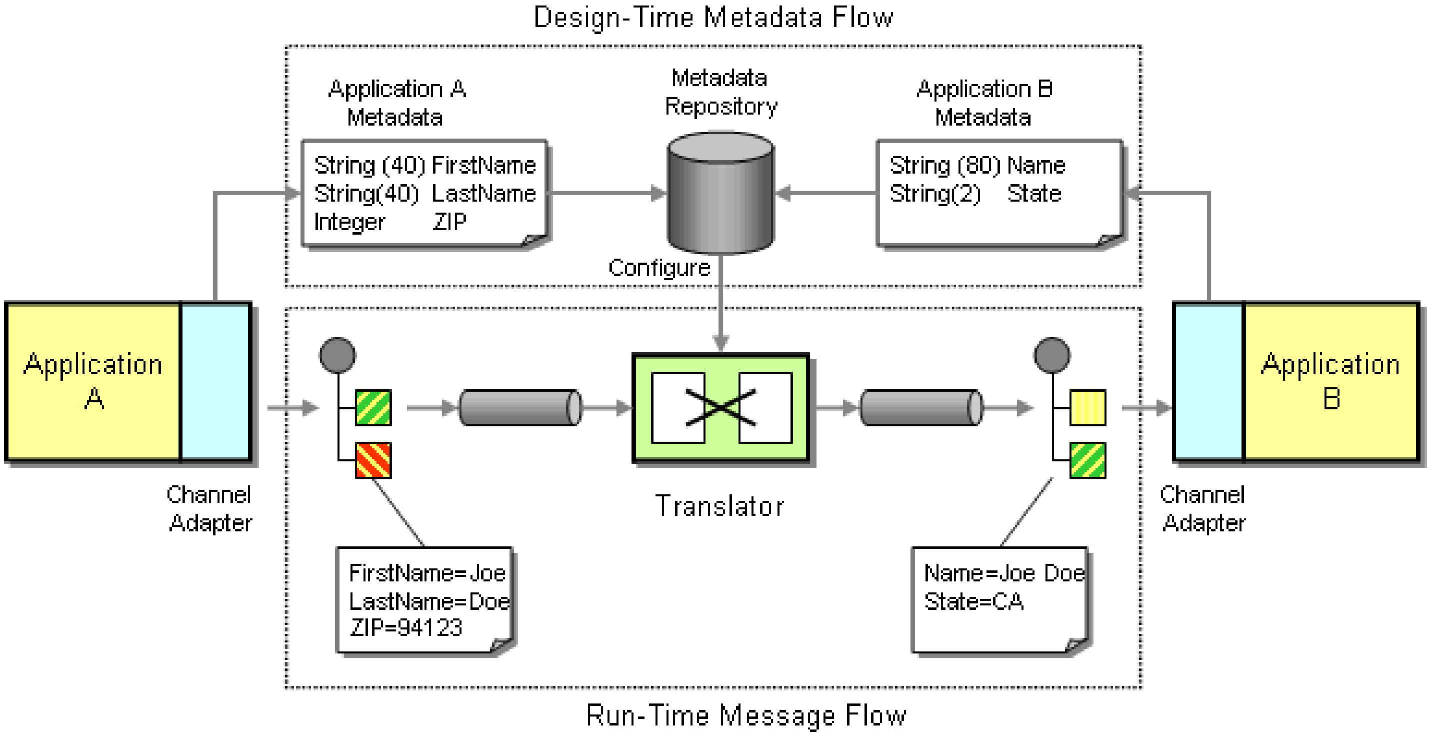
http://www.enterpriseintegrationpatterns.com/MessageRouter.html

EAI Pattern: Message Router

- code sample

EAI Pattern: Transformer

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



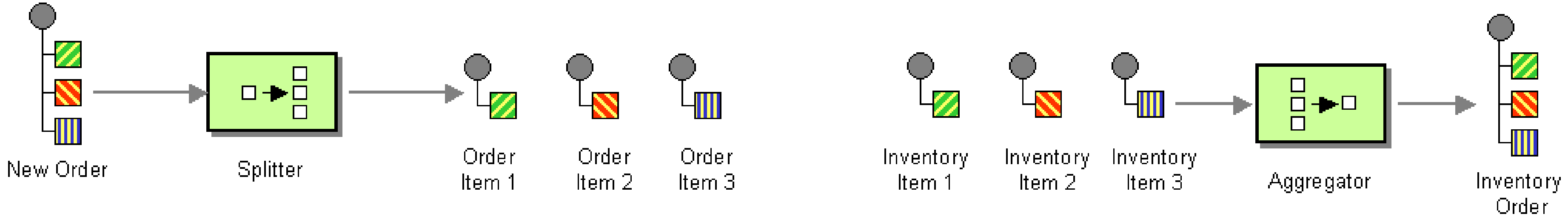
http://www.enterpriseintegrationpatterns.com/MessageTransformationIntro.html

EAI Pattern: Transformer

- code sample

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 Pattern: Splitter and Aggregator

- code sample

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

Q & A

- Feel free to
- ask questions now
- 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
-