EAI War Stories

Praxisbeispiele zu EAI-Pattern und Lessons Learned

Alexander Heusingfeld, @goldstift
Martin Huber, @waterback

We take care of it - personally!

innoQ

EAI Pattern in 2013?

Nobody uses them anymore!

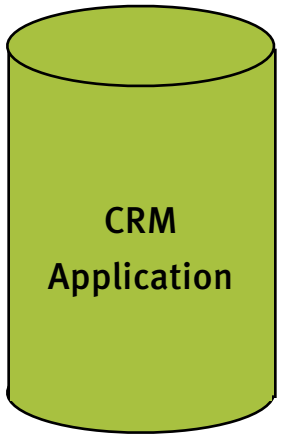
innoQ

That’s THE problem!

innoQ

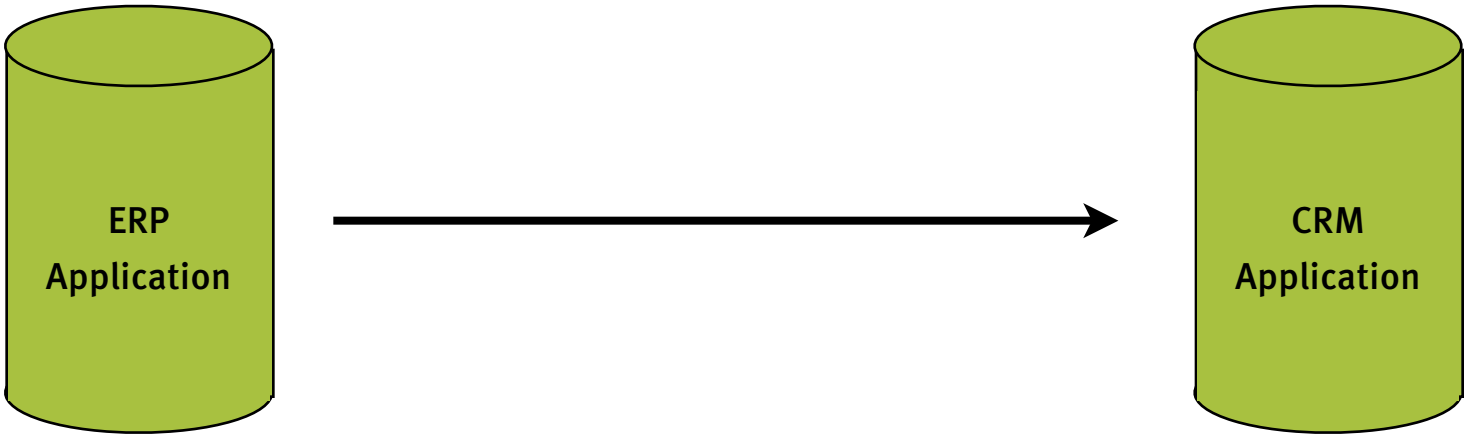
Integration is ubiquitous

Multitude of potential service consumers



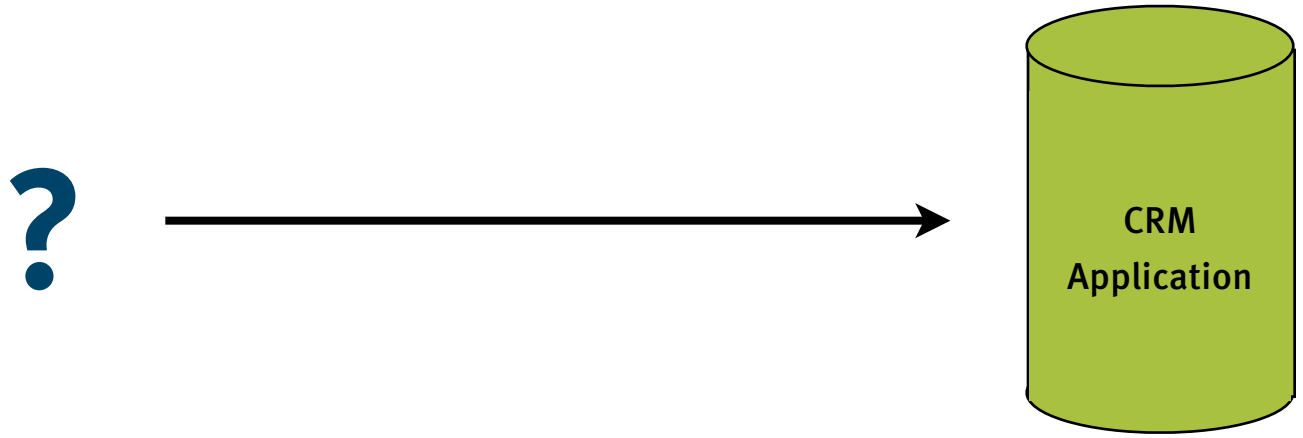
Integration is ubiquitous

Multitude of potential service consumers



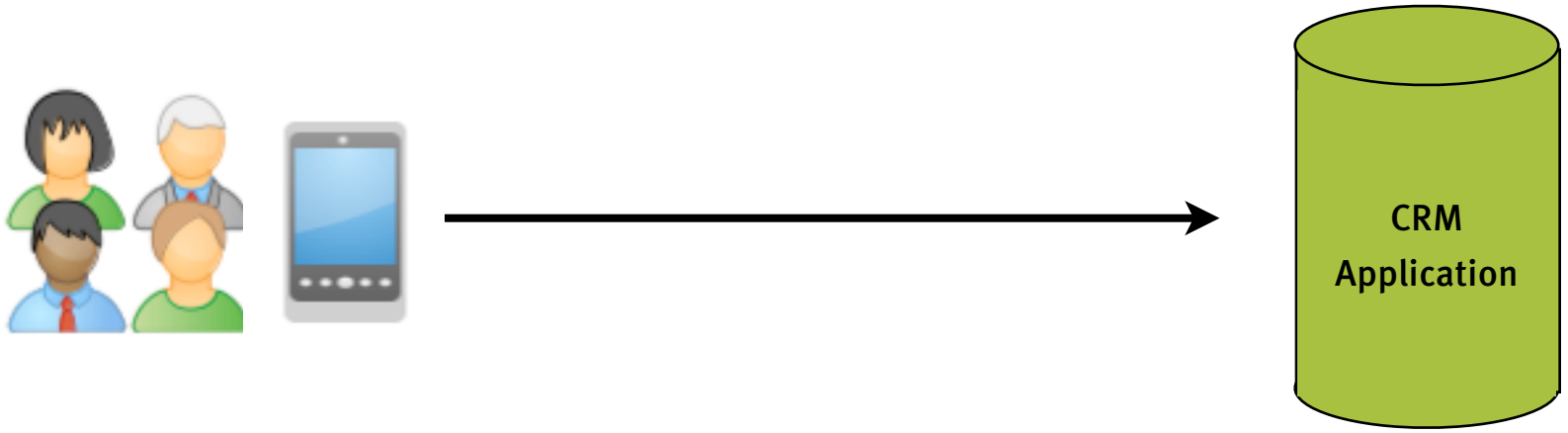
Integration is ubiquitous

Multitude of potential service consumers



Integration is ubiquitous

Multitude of potential service consumers

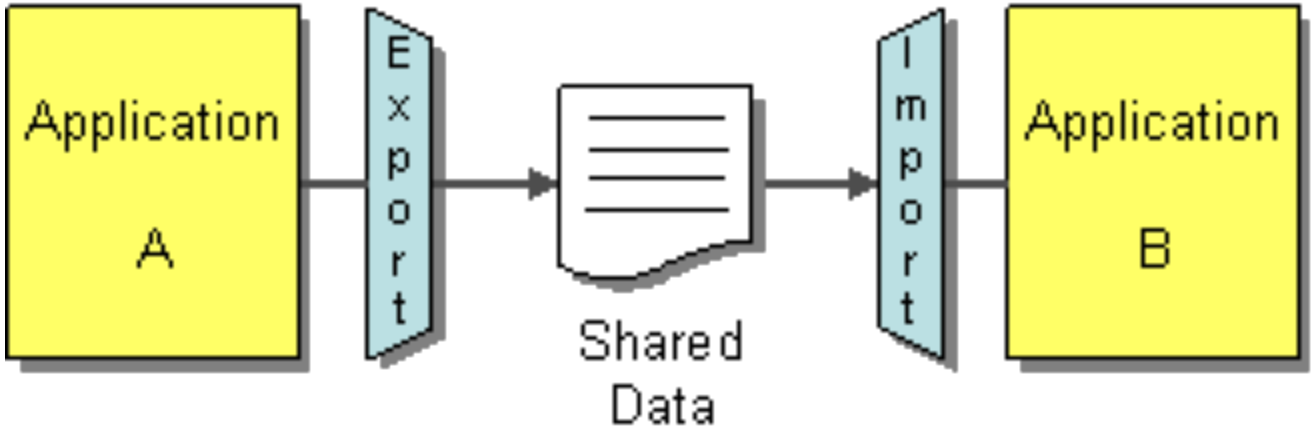


Integration Styles

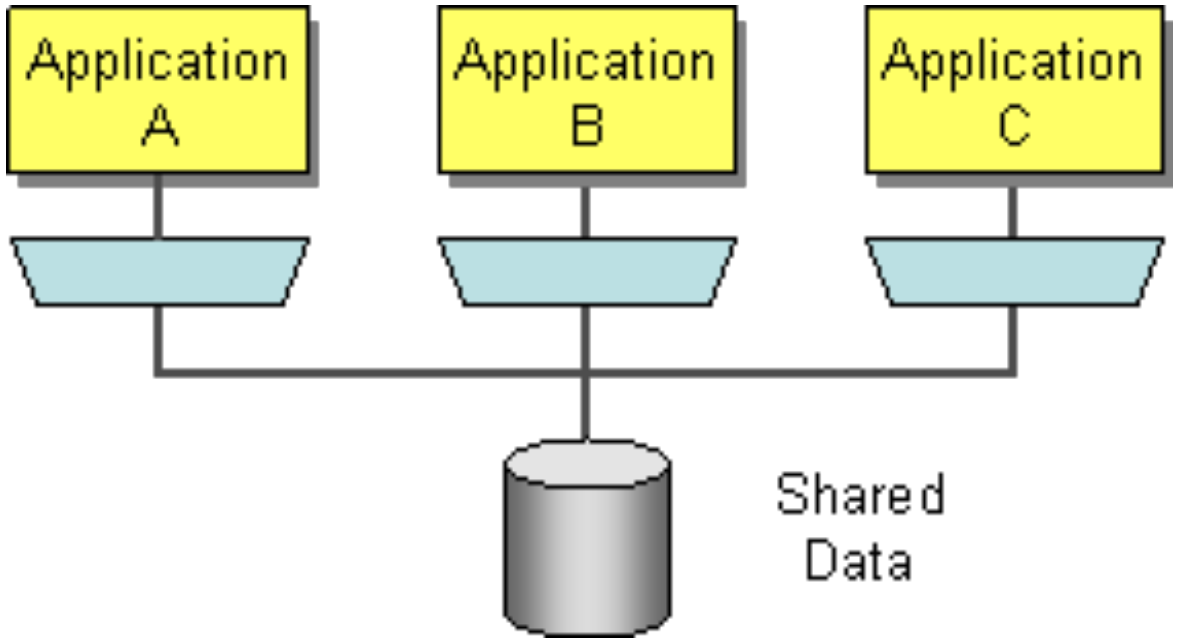
We take care of it - personally!

innoQ

File Transfer

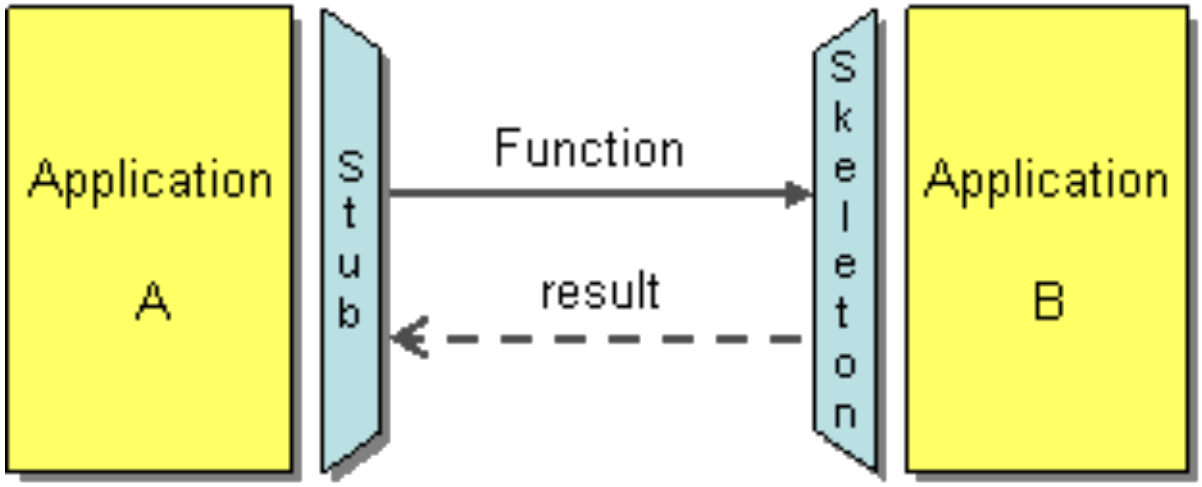


Shared Database

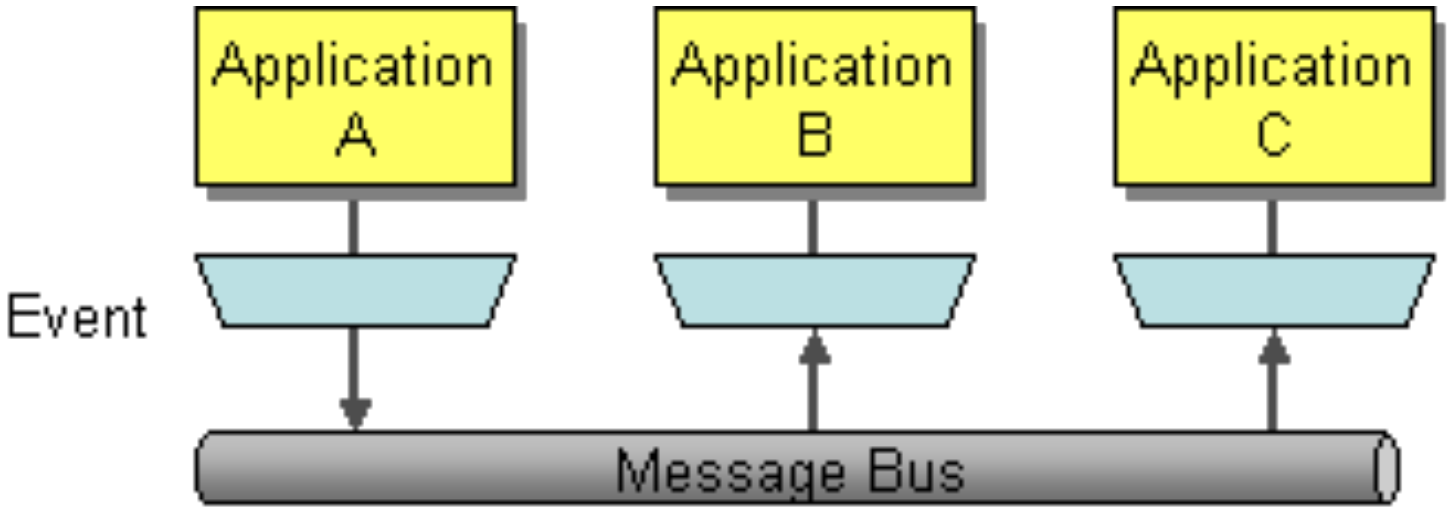


http://www.eaipatterns.com/SharedDataBaseIntegration.h

Remote Procedure Invocation

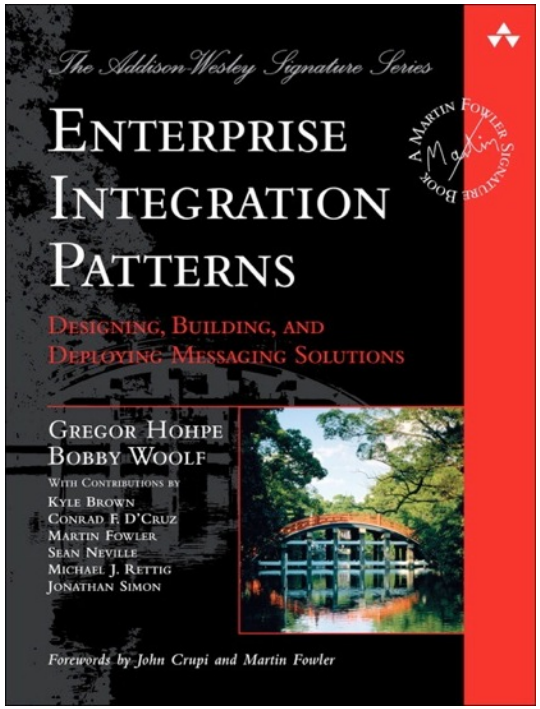


Messaging




http://bit.ly/PtrhWy
REICHSPOST
Deutsche Post
BRIEFKASTEN
Variationen
eines
Klassikers
DEUTSCHE
Messaging?

Searching for a book?



Enterprise Integration Patterns

(Hohpe & Woolf), 2003

www.eaipatterns.com

Swiss-army knife for asynchronous messaging

innoQ

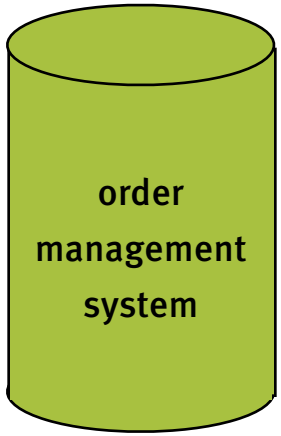
Theories in practice

We take care of it - personally!

innoQ

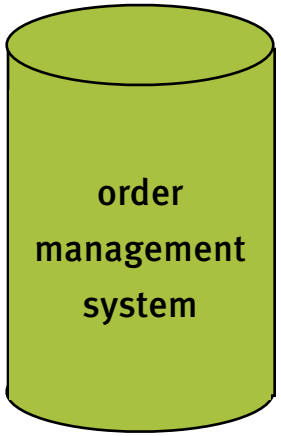
Real Life Scenario

Simple order management system (CRUD)



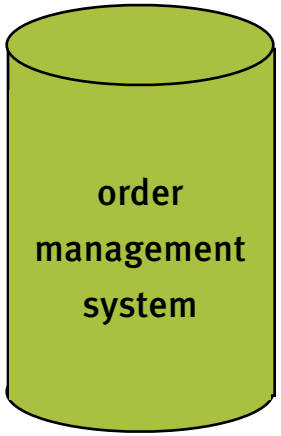
innoQ

Simple ChangeRequest



Simple ChangeRequest

We need an importer!



A Java EE servlet to the rescue

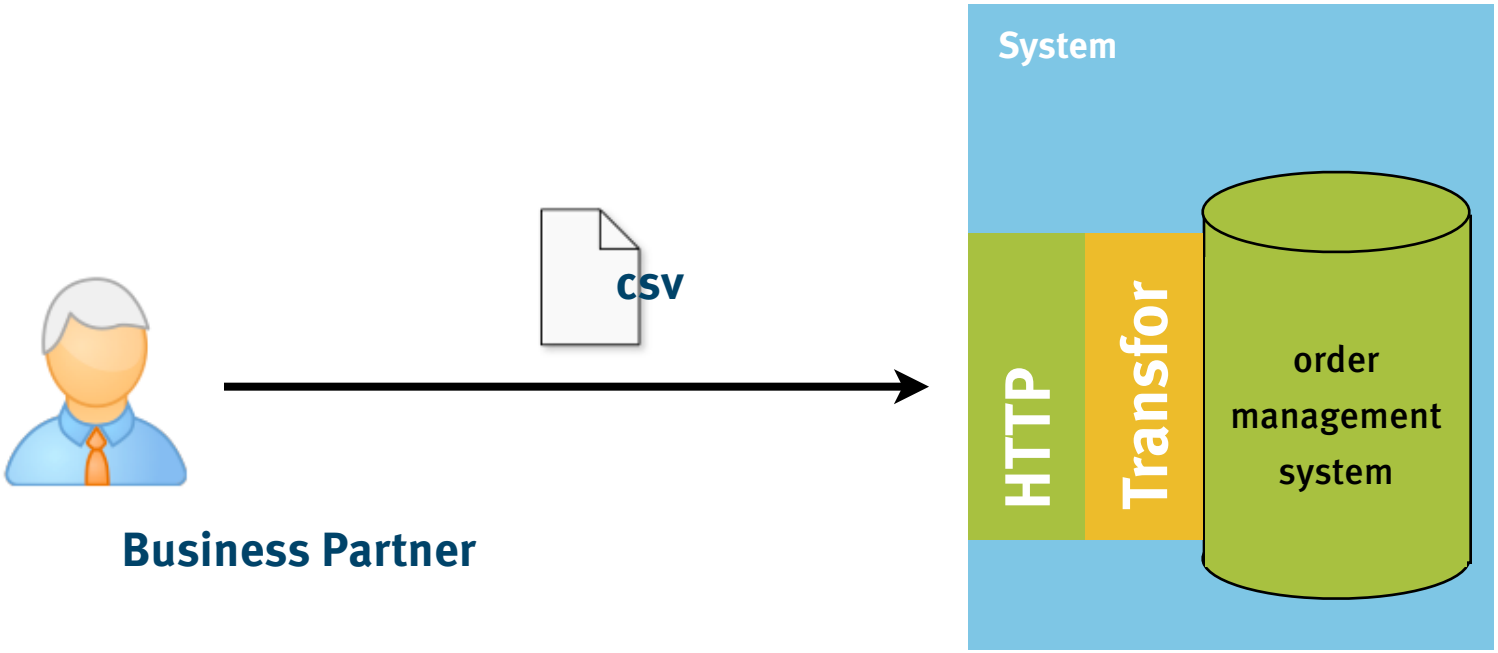
```
@WebServlet(urlPatterns = {"/order/import"})
public class ImporterServlet extends HttpServlet {

    @Inject OrderBean bean;
    @Inject CsvDataParser parser;

    @Override
    protected void doPost(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {
        String data = request.getParameter("csvdata");
        List<Order> orders = parser.transform(data);
        for (Order order : orders) {
            bean.persist(order);
        }
    }
}
```

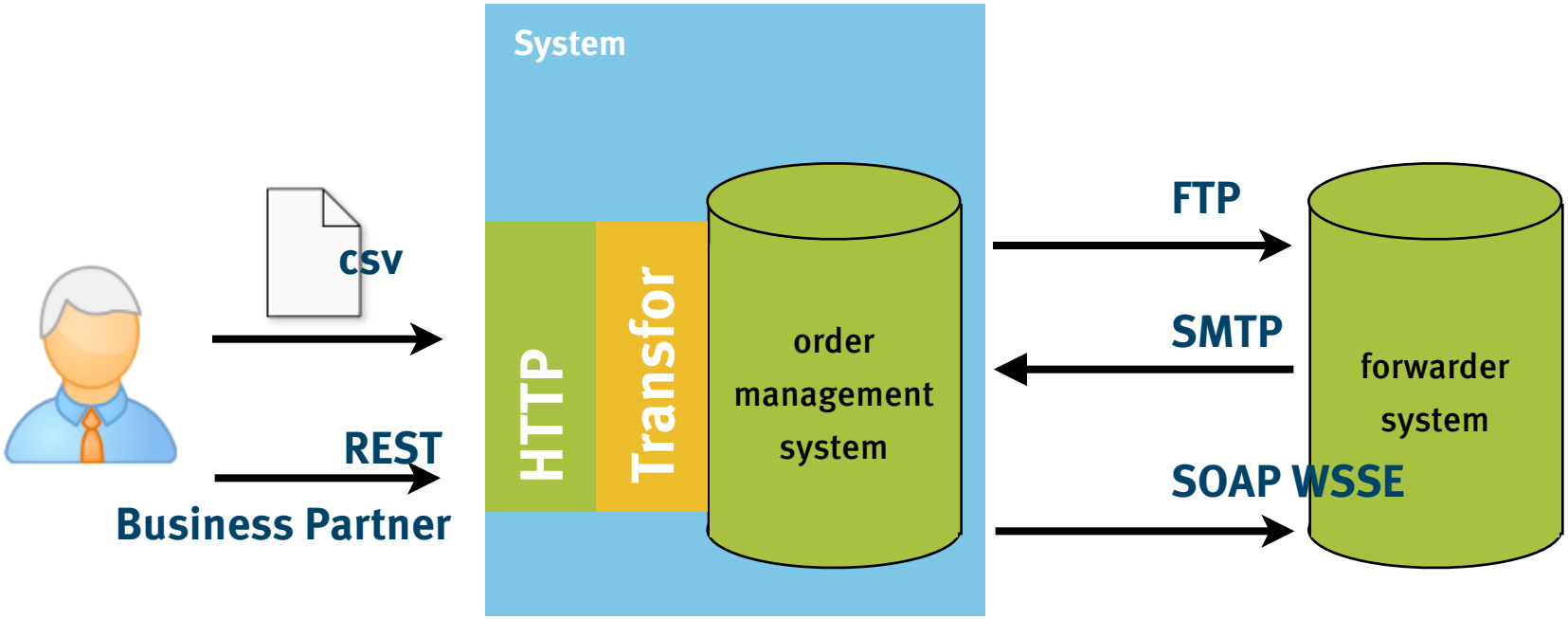
innoQ

Simple Change done



Change happens

Integrate the system of a forwarder



What are the lessons learned?

innoQ

Recognize an integration task

when it‘s staring in your face?


http://www.flickr.com/photos/wespionage/184793114
EASY!

Build an easy and maintainable solution

for multiple integration challenges?

innoQ


http://www.flickr.com/ph
NO PROBLEM!

Any help in the toolbox?

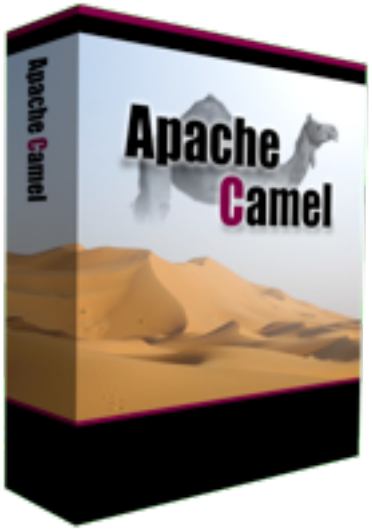
innoQ

Property of ChinBuye Limited. http://bit.ly/xBKwGw

ADAPTERS?

innoQ

EAI Frameworks



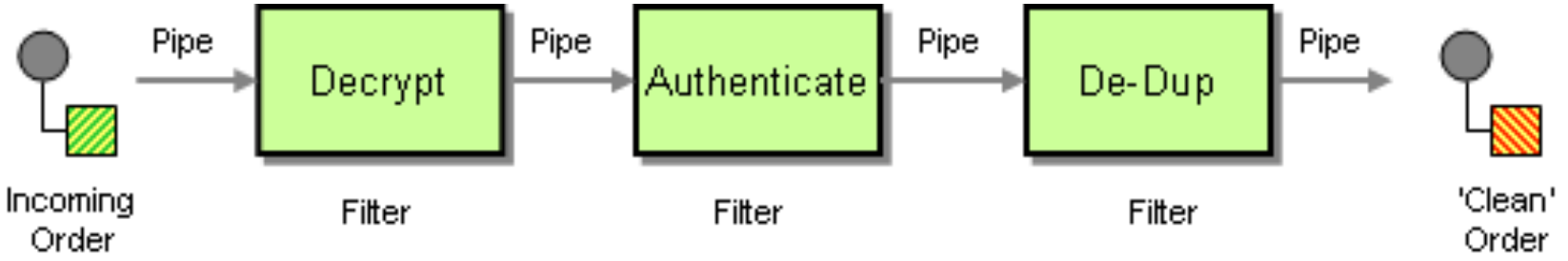
Apache Camel

Spring Integration

innoQ

Pipes and Filters

divide your task into small steps



Camel-XML with Spring

```
<beans>
…
   <CamelContext id="mycoolCamelContext">
      <route>
         <from uri="file://Users/martinh/temp/sample1"/>
         <bean ref="orderTransform" method="transformOrder"/>
         <bean ref="orderCheckBean" method="validateOrder"/>
         <to uri="jms:queue:order.process"/>
      </route>
    </CamelContext>

    <bean id="orderTransform" class="con.innoq.sample.OrderTransform">
        <property name="company" value="acme" />
    </bean>

    <bean id="orderCheckBean" class="con.innoq.sample.OrderCheck">
    </bean>
</beans>
```

Camel fluent API in Java

```
public class PipelineExample extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("file://Users/martinh/temp/sample1")
            .to("bean:orderTransform?method=transformOrder")
            .to("bean:orderCheckBean?method=validateOrder")
            .to("jms:queue:order.process");
    }
}
```

Pipes & Filters

```
public class PipesFiltersExample extends RouteBuilder {

    @Override
    public void configure() throws Exception {
        from("jms:queue:order.in?maxConcurrentConsumers=5")
            .to("bean:decryptBean")
            .to("seda:authenticate");

        from("seda:authenticate")
            .process(new AuthenticationProcessor())
            .to("seda:dedup");

        from("seda:dedup")
            .process(new DeDupProcessor()
            .to("jms:queue:order.processfurther");

    }
}
```

innoQ

Pipes & Filters

```
from("seda:authenticate")
    .process(new AuthenticationProcessor())
    .to("seda:dedup");
```

```
public class AuthenticationProcessor implements Processor {
    @Override
    public void process(Exchange exchange) throws Exception {
          MyOrder order = (MyOrder)exchange.getIn().getBody();
          String type = exchange.getIn().getHeader("orderType");
    }
}
```

innoQ

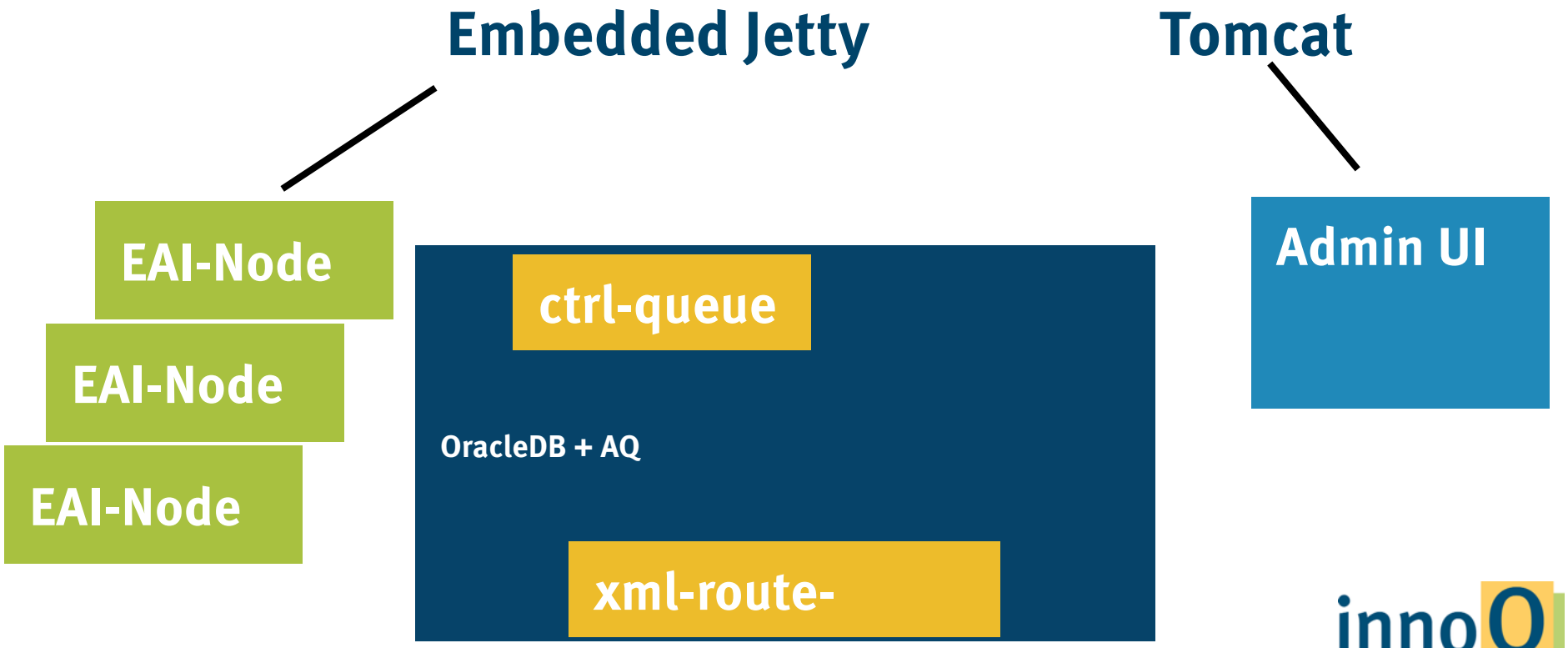
All you need are the right tools for the job?

innoQ

http://www.flickr.com/ph

ALMOST

Real-life scenario: Control Bus

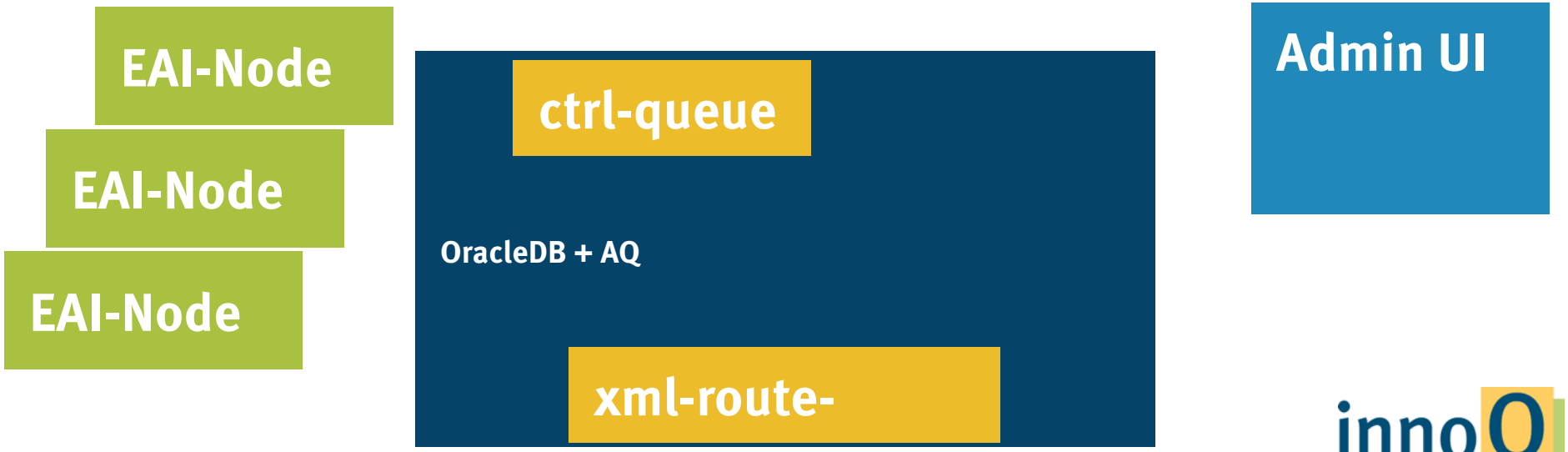


Real-life scenario: Control Bus

Initial Node configuration

database connection

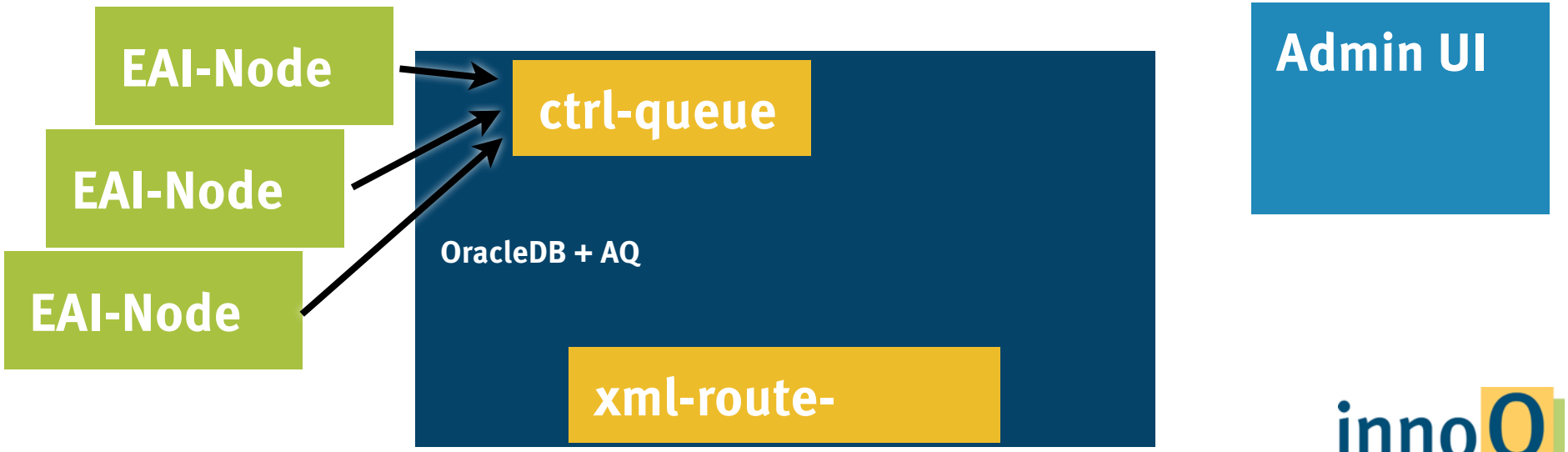
node type (command line arg)



Real-life scenario: Control Bus

1. Server startup

subscribe to control queue

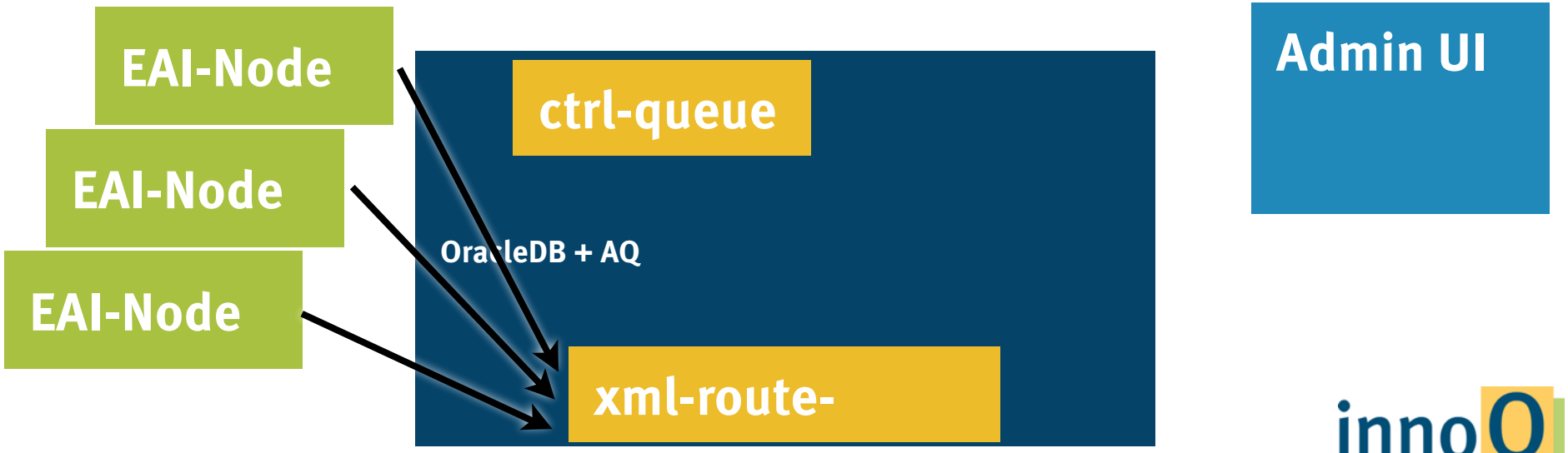


Real-life scenario: Control Bus

1. Server startup

subscribe to control queue

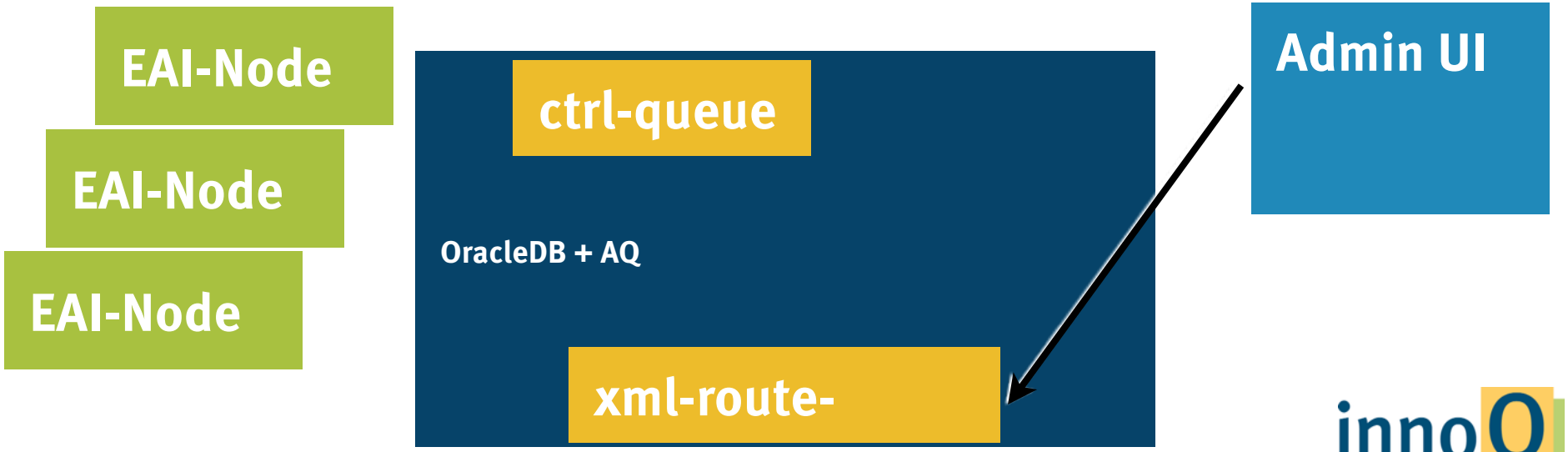
load *node-type specific* config



Config change @Runtime

2. during runtime

route config is changed

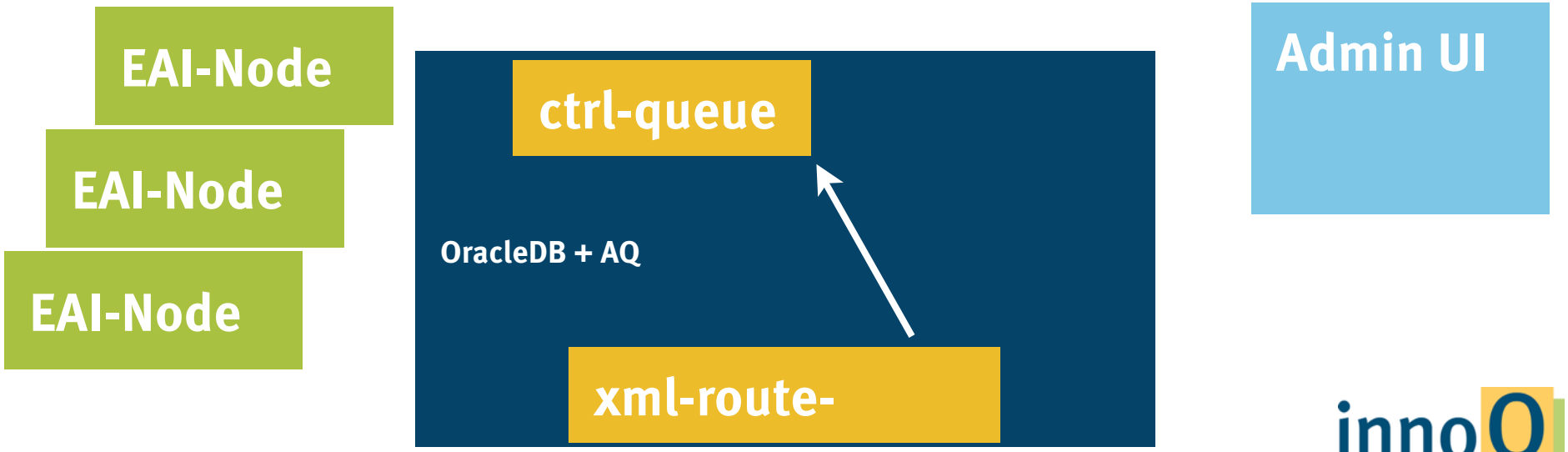


Config change @Runtime

2. during runtime

route config is changed

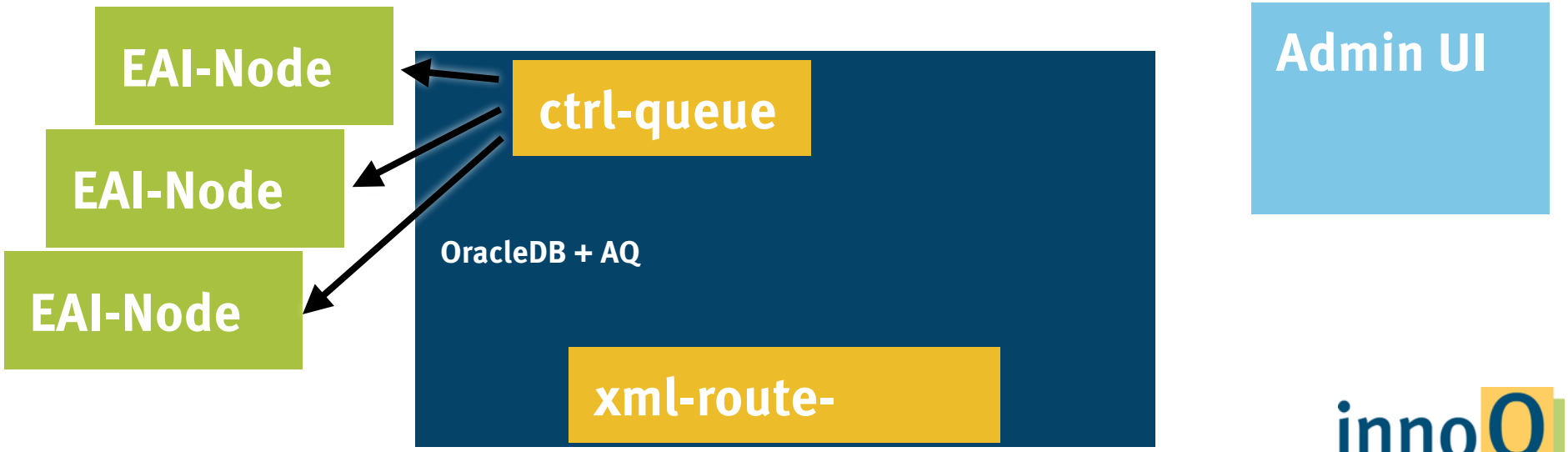
publish config update message



Config change @Runtime

3. upon control message arrival

interpret relevance of control message

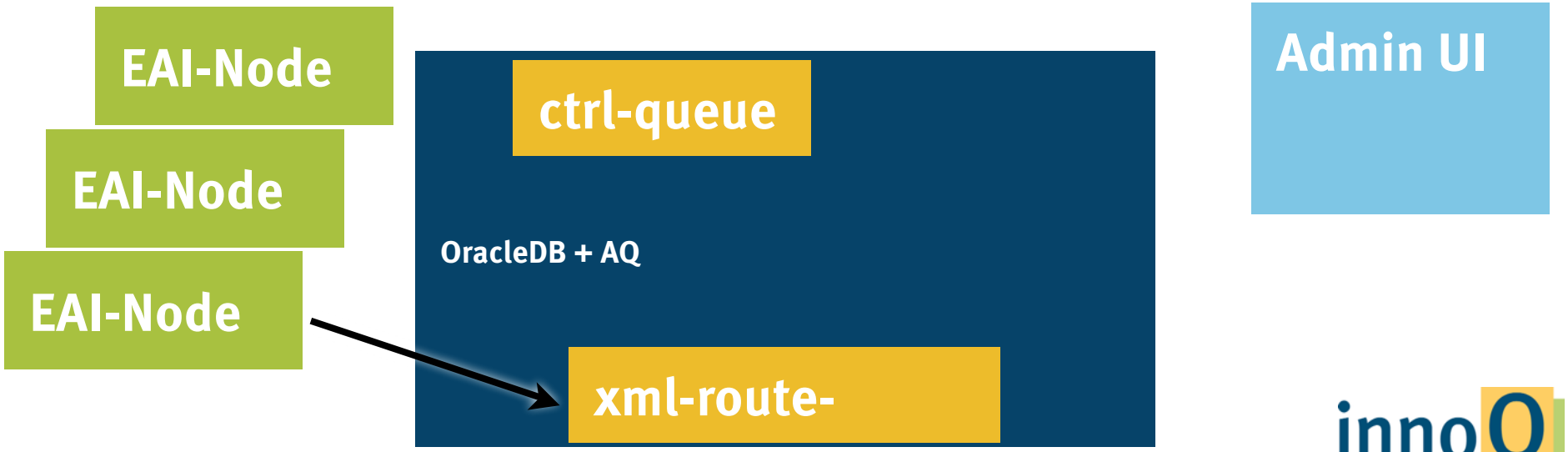


Config change @Runtime

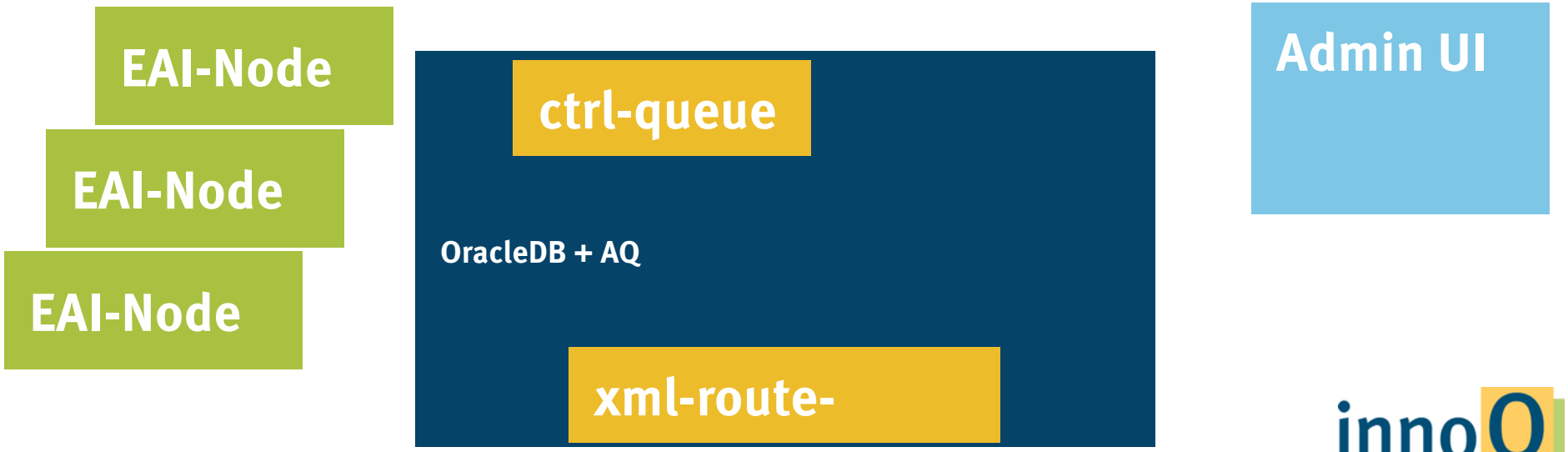
3. upon control message arrival

interpret relevance of control message

affected nodes reload configuration



Lessons Learned?



Testbarkeit vs. Konfigurierbarkeit?

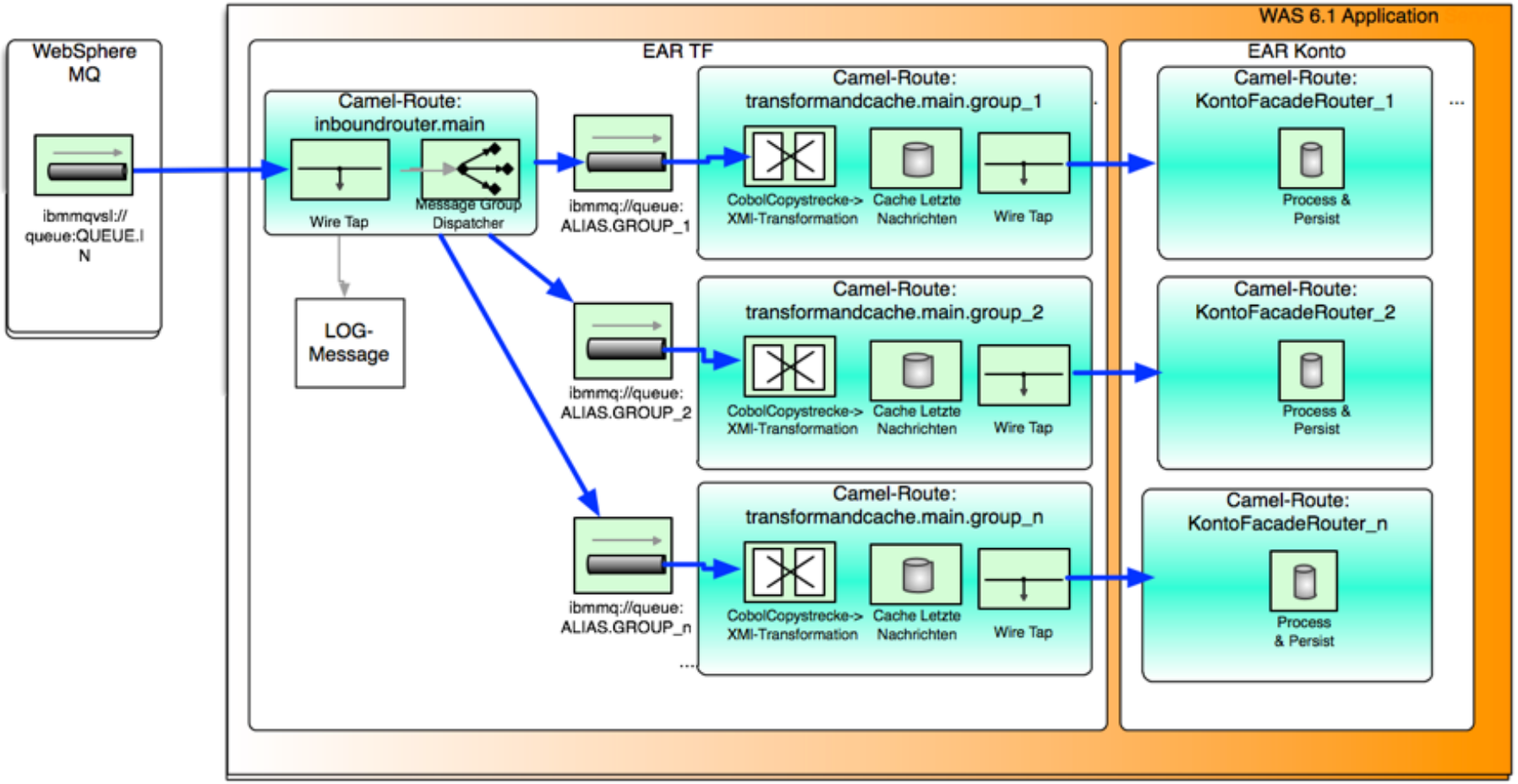
innoQ

Testbarkeit und Konfigurierbarkeit

innoQ

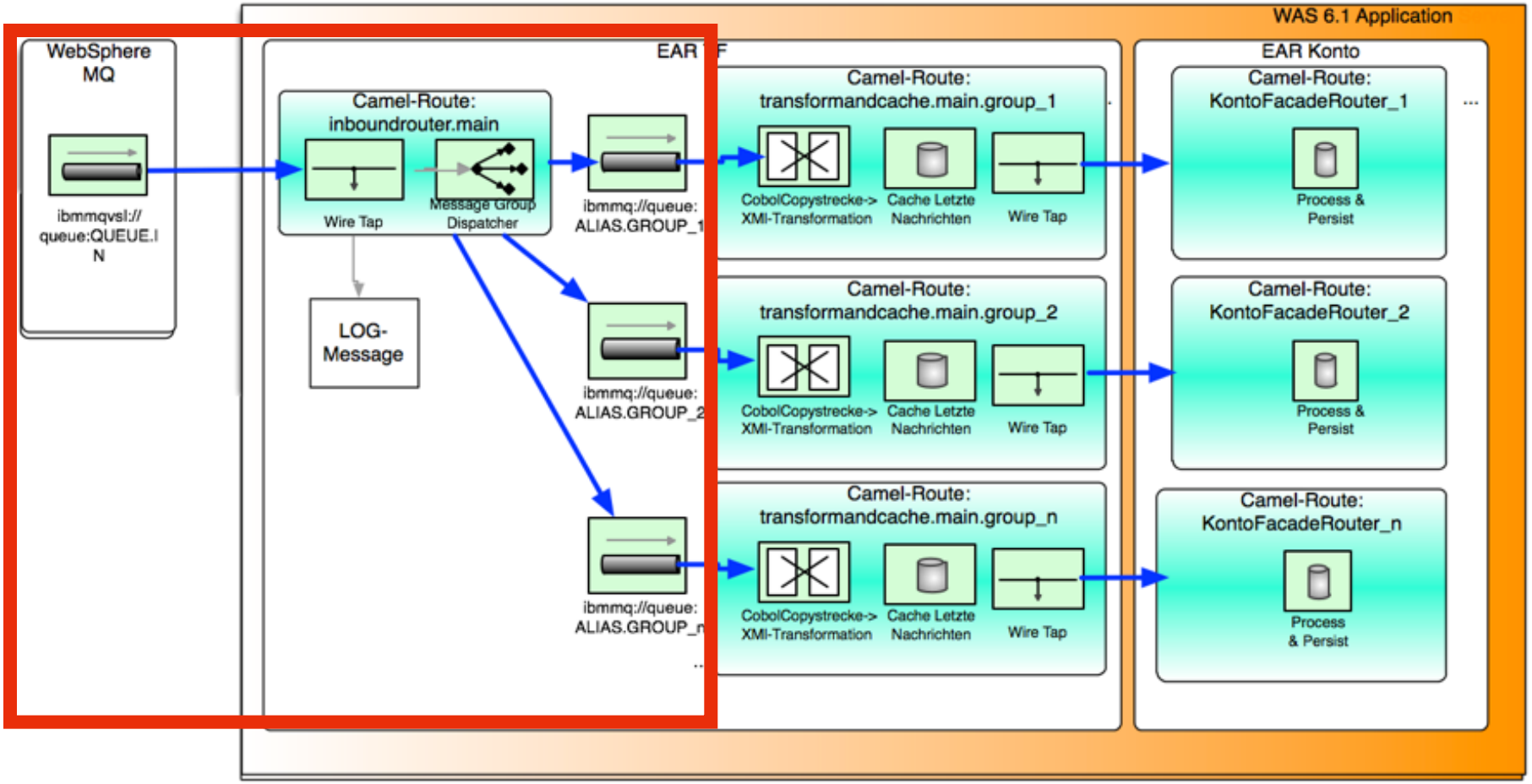
Message Ordering

Software-Architektur:
Skalierung & Pipes Filters bei Einhaltung der Nachrichten-Reihenfolge



Message Ordering

Software-Architektur:
Skalierung & Pipes Filters bei Einhaltung der Nachrichten-Reihenfolge



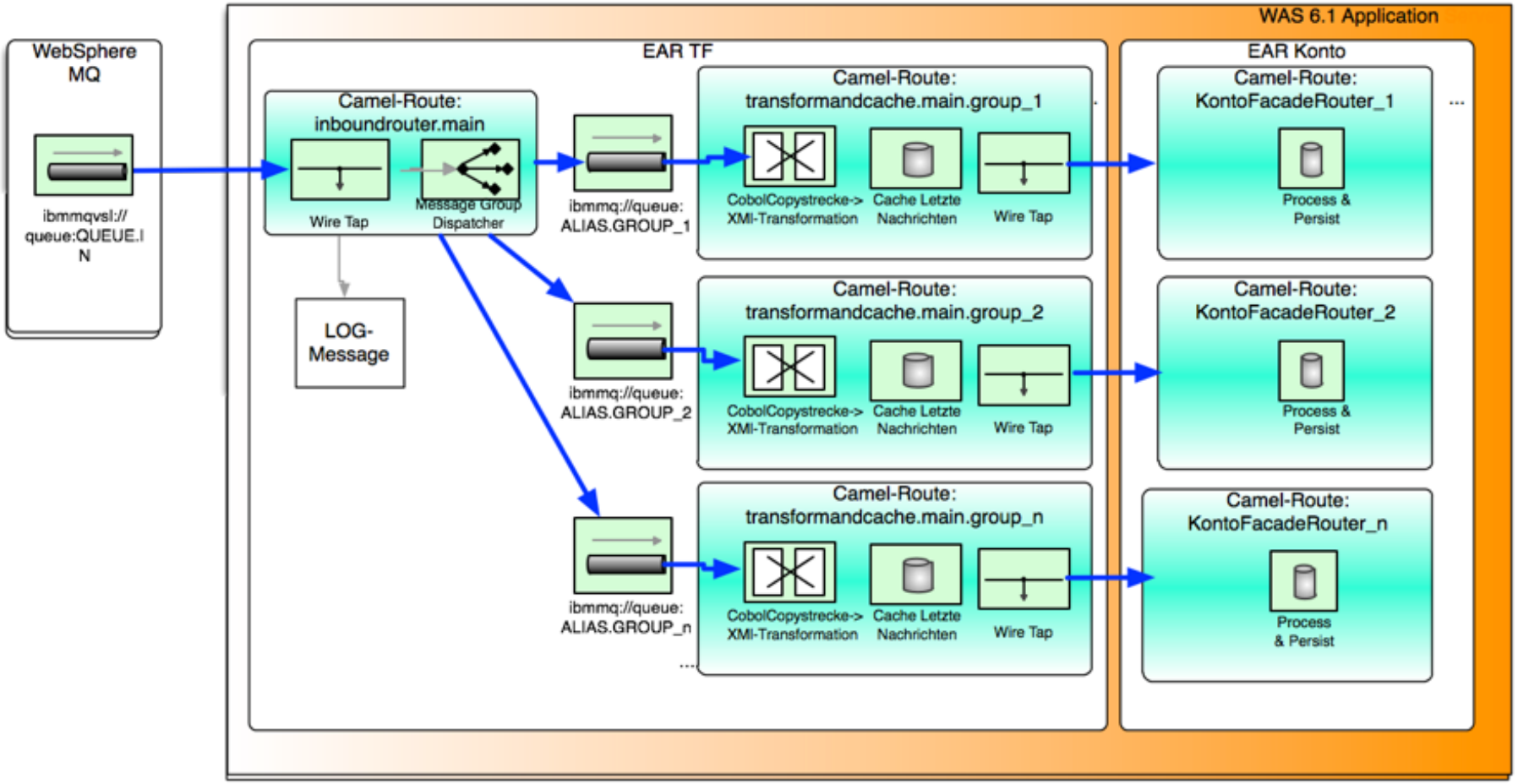
Message Ordering

```
from(“ibmmq:QUEUE.IN“).noAutoStartup()
            .transacted()
            .policy(required)
            .routeId(BasicRouteBuilder
                        .stdfromRouteId(this.componentBaseName))
            .process(vslHeadersProcessor)
            .setHeader(HDR_DUMMYCOMPLCHK_IBR, constant(PASSEDMSG))
            .bean(inboundTargetDispatcherBean)
            .wireTap("seda:inboundRouter.log");
```

innoQ

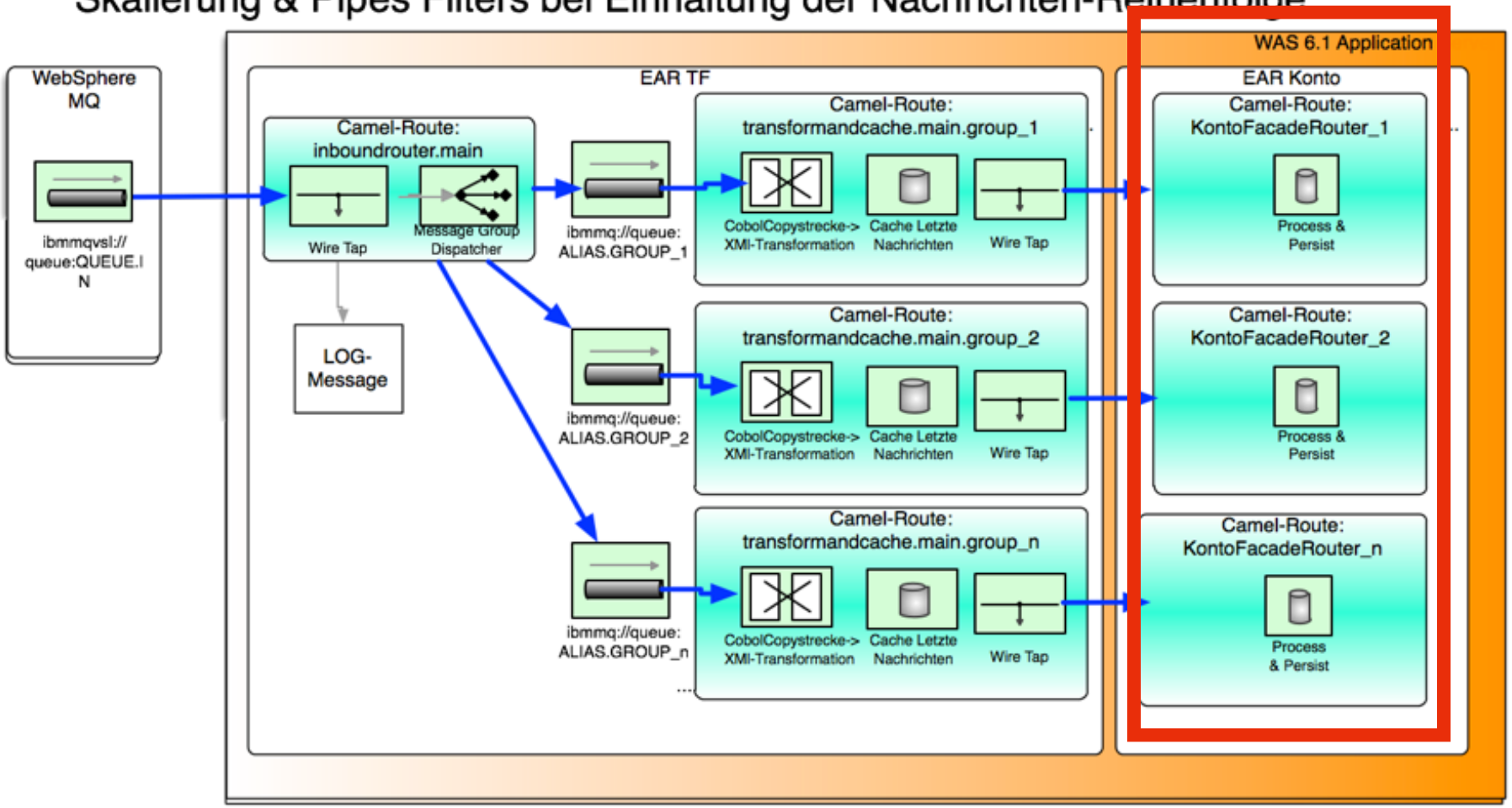
Message Ordering

Software-Architektur:
Skalierung & Pipes Filters bei Einhaltung der Nachrichten-Reihenfolge



Message Ordering

Software-Architektur:
Skalierung & Pipes Filters bei Einhaltung der Nachrichten-Reihenfolge



Message Ordering

```
SpringTransactionPolicy required = lookup("PROPAGATION_REQUIRED",
                                          SpringTransactionPolicy.class);
for (int i = 0; i < NROFROUTES; i++) {
    String iasS = String.valueOf(i);
    from(fromEndpoint.replaceAll("§", iasS)).noAutoStartup()
        .transacted()
        .policy(required)
        .routeId(KONTO_ZP_PREROUTEID + iasS)
        .process(new CheckRedeliveryProcessor())
        .process(new ZahlplanFacetConverter())
        .bean(zpBean, "process")
        .to("bean:kontostatctr?method=stopCounter")
        .wireTap(logEndpoint);

}
```

innoQ

Does it save money?

innoQ

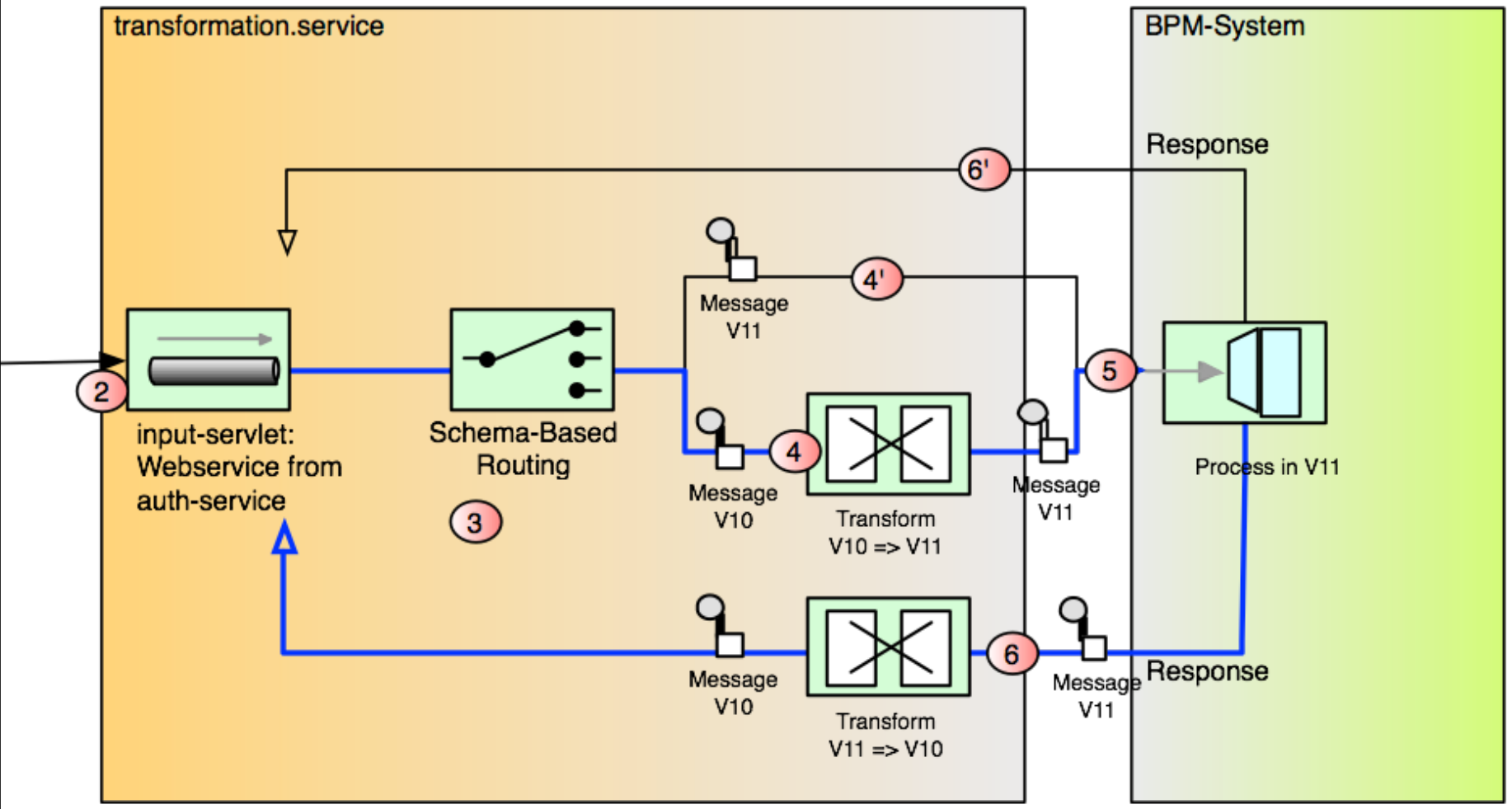
Does it save money?

innoQ

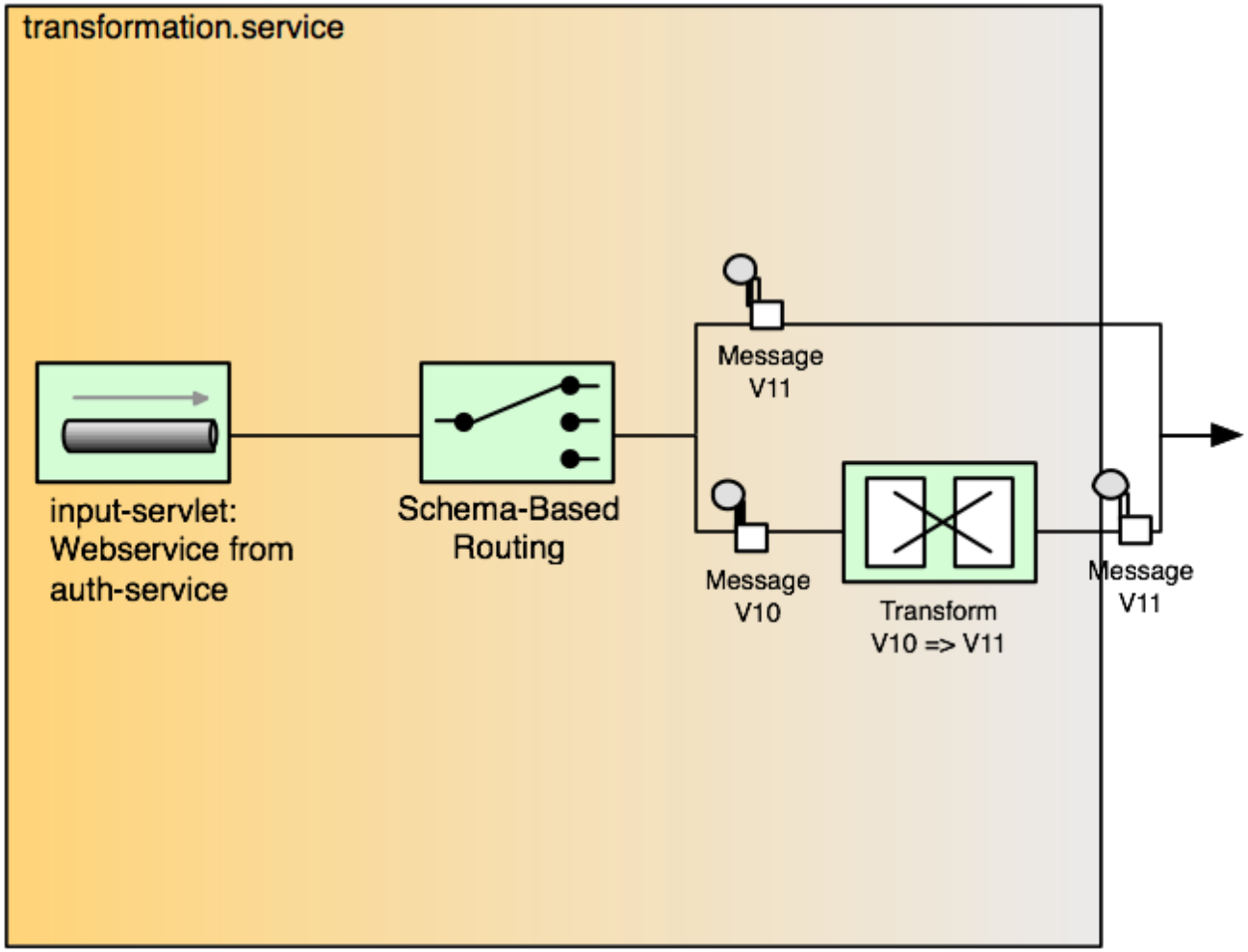
Backward compatibility via EAI

innoQ

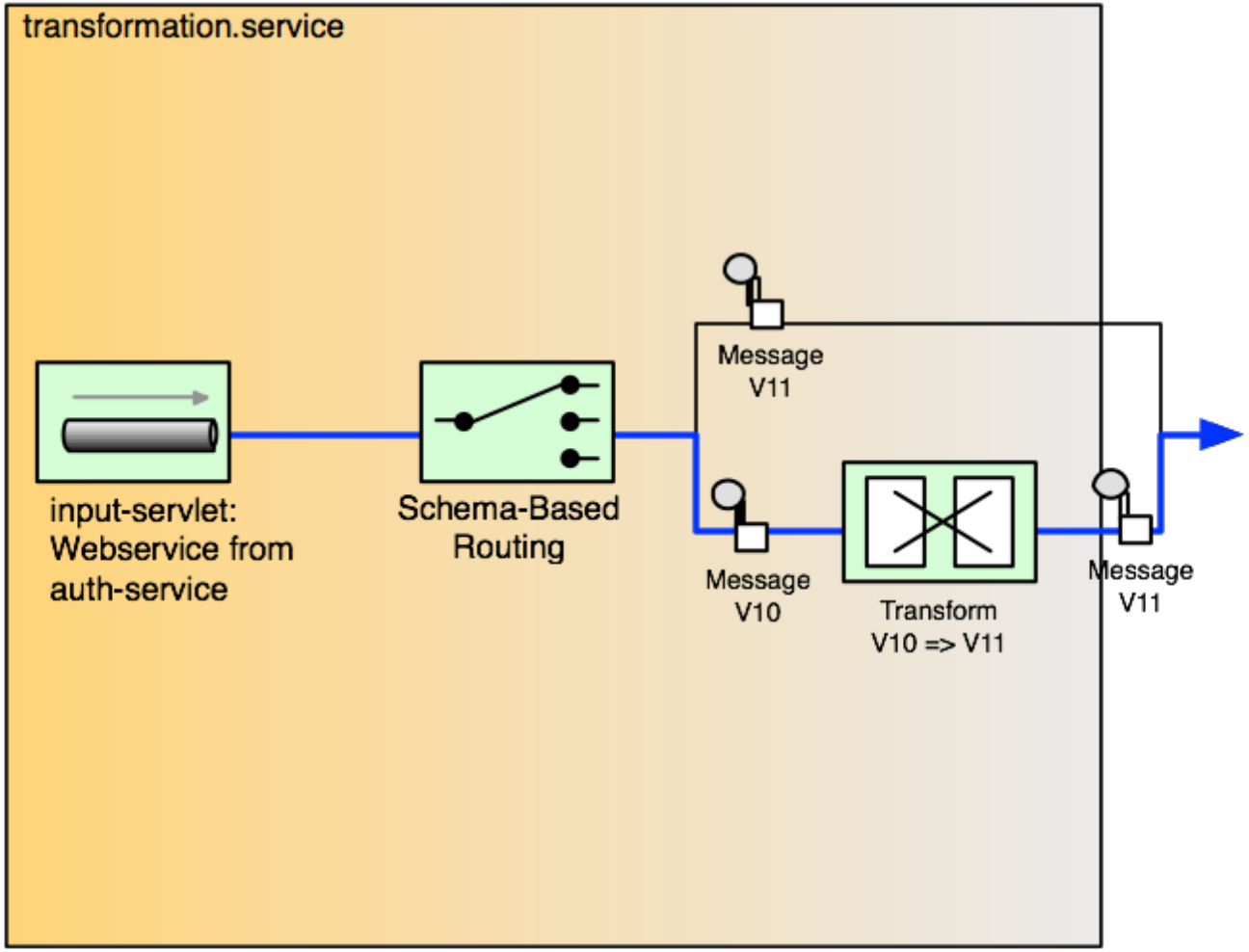
Transformation mit BPMN-System (1)



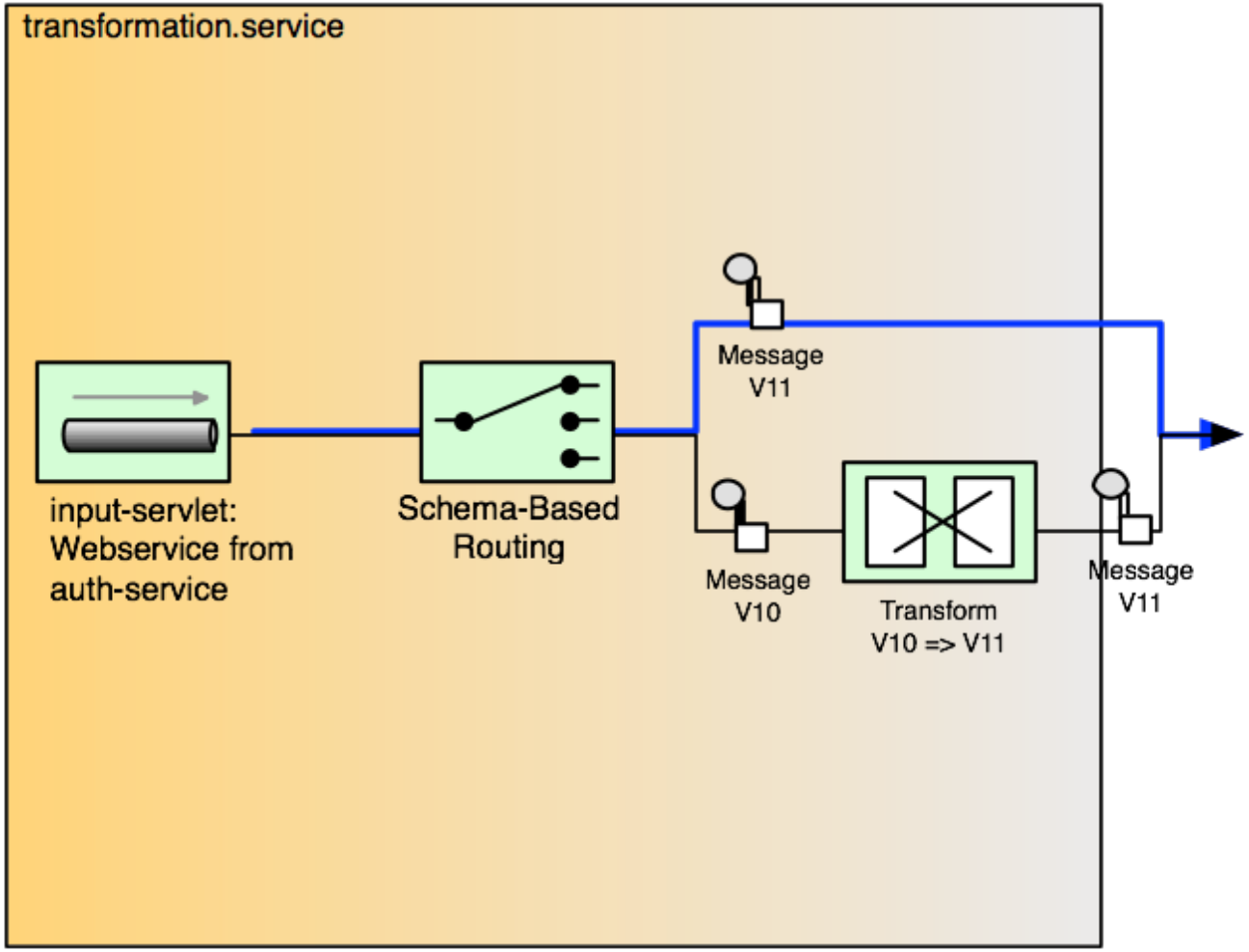
Transformation für BPMN-System (2)



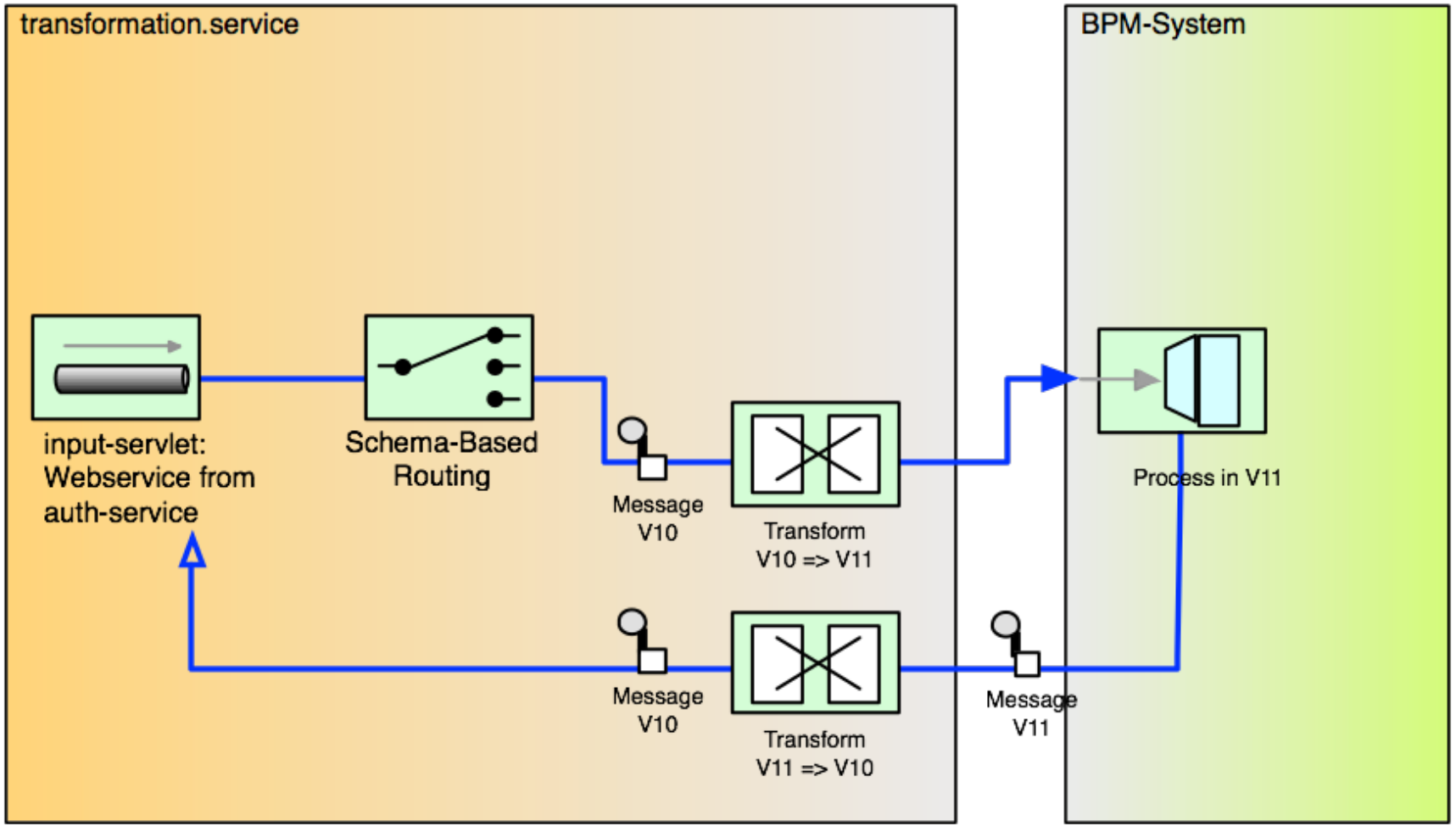
Transformation für BPMN-System (2)



Transformation für BPMN-System (2)



Transformation für BPMN-System (3)



Should I always use an EAI-Framework?

innoQ

Should I always use an EAI-Framework?

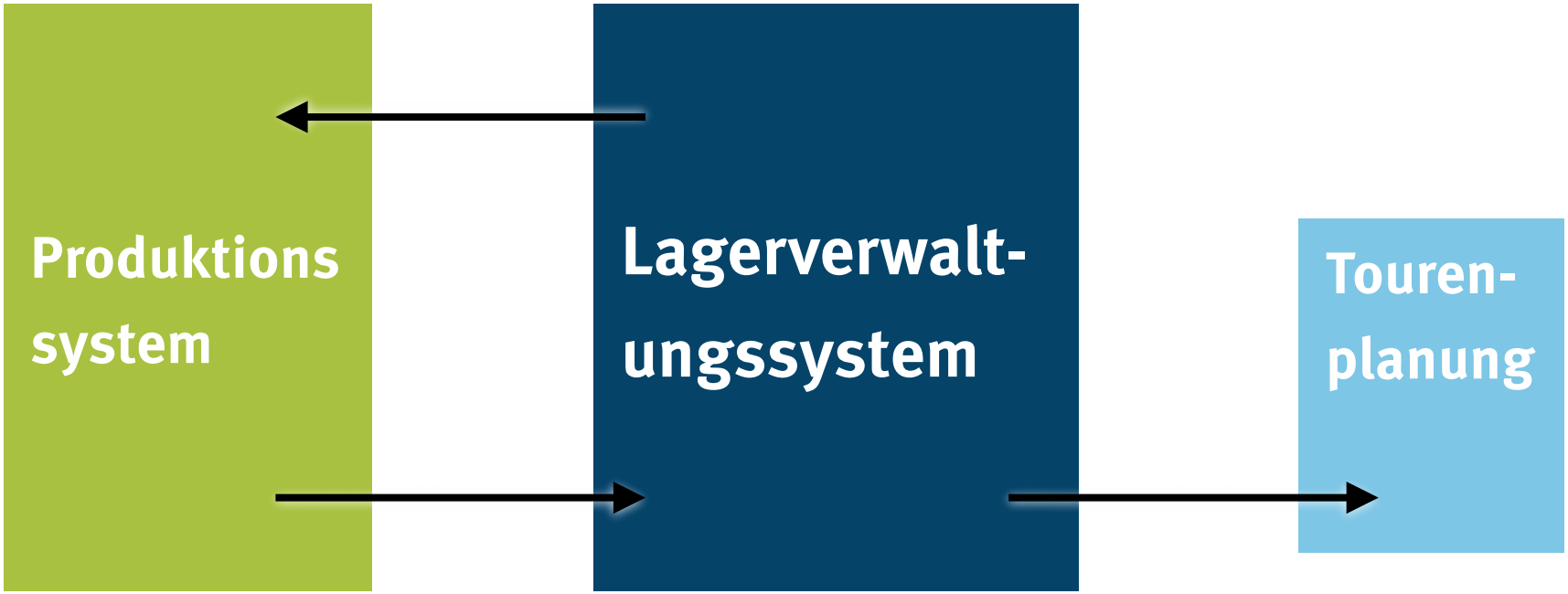
It depends!

innoQ

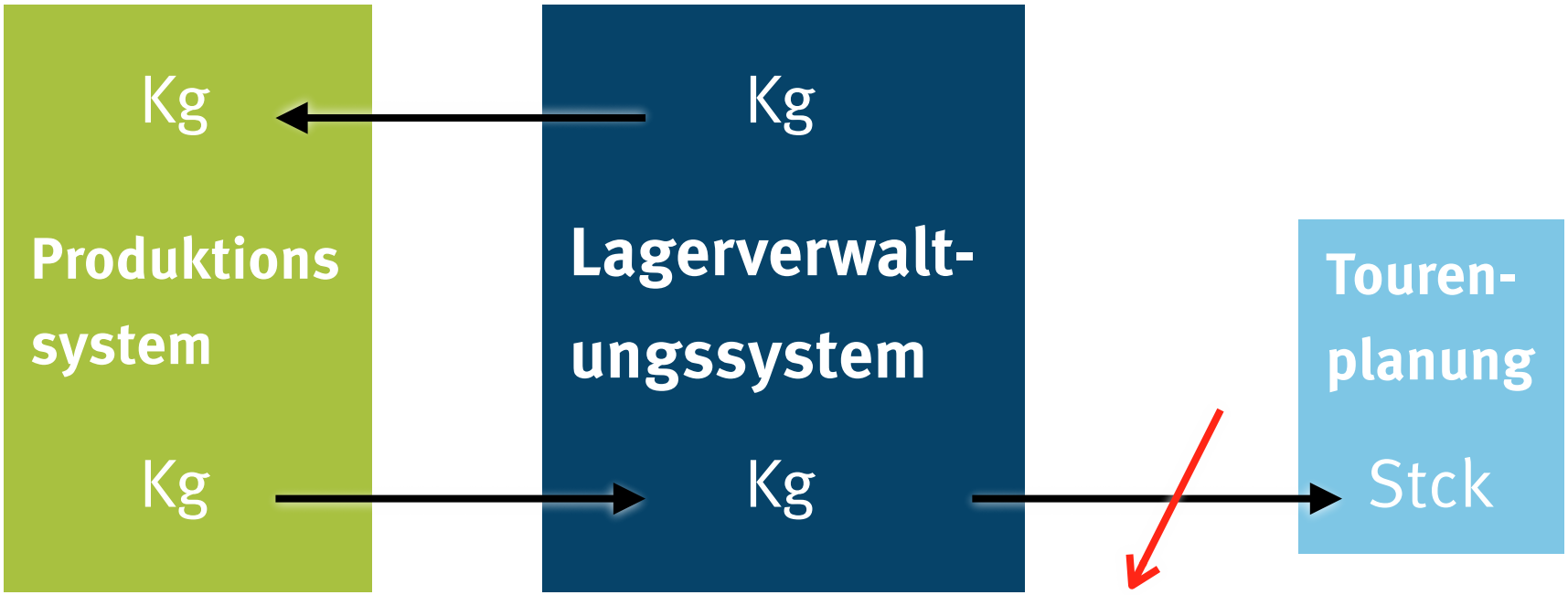
Scenario: Kilogramm vs. Stück

innoQ

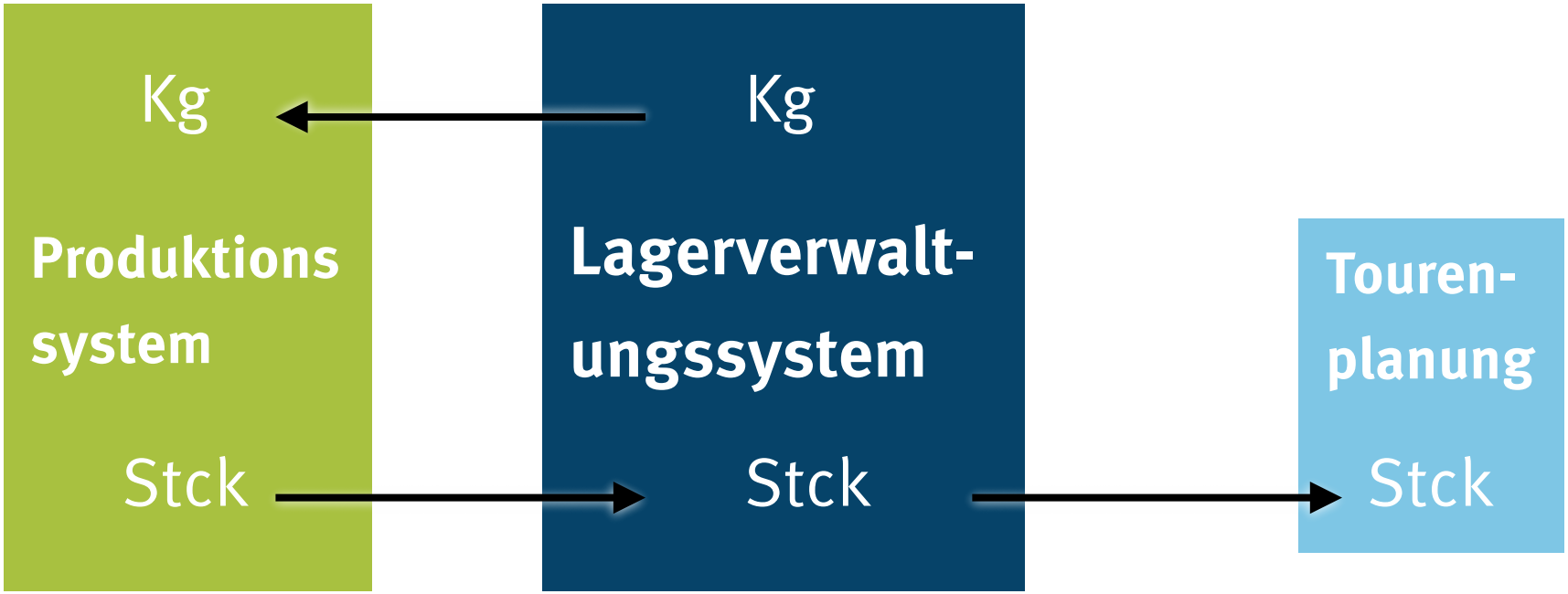
Ein Integrationsproblem?



Ein Integrationsproblem?



Kein Integrationsproblem



Further tips considering EAI

Further tips considering EAI

small building blocks

Further tips considering EAI

small building blocks

no shared mutable state

innoQ

Further tips considering EAI

small building blocks

no shared mutable state

use immutable DTOs

innoQ

Further tips considering EAI

small building blocks

no shared mutable state

use immutable DTOs

=› increased scalability

Further tips considering EAI

small building blocks

no shared mutable state

use immutable DTOs

=› increased scalability

advanced: see “SEDA” & “actor model”

Vielen Dank!

Alexander Heusingfeld, **@goldstift**

alexander.heusingfeld@innoq.com

Martin Huber, **@waterback**

martin.huber@innoq.com

http://www.innoq.com/de/talks

We take care of it - personally!

innoQ