

it's about the people



inventi

# Integration platform development

David Bohmann

14.11.2018

# Customer's project assignment

Microsoft Dynamix AX (Axapta)

- Enterprise resource planning (ERP) tool

Etnetera a. s.

- New e-shop development

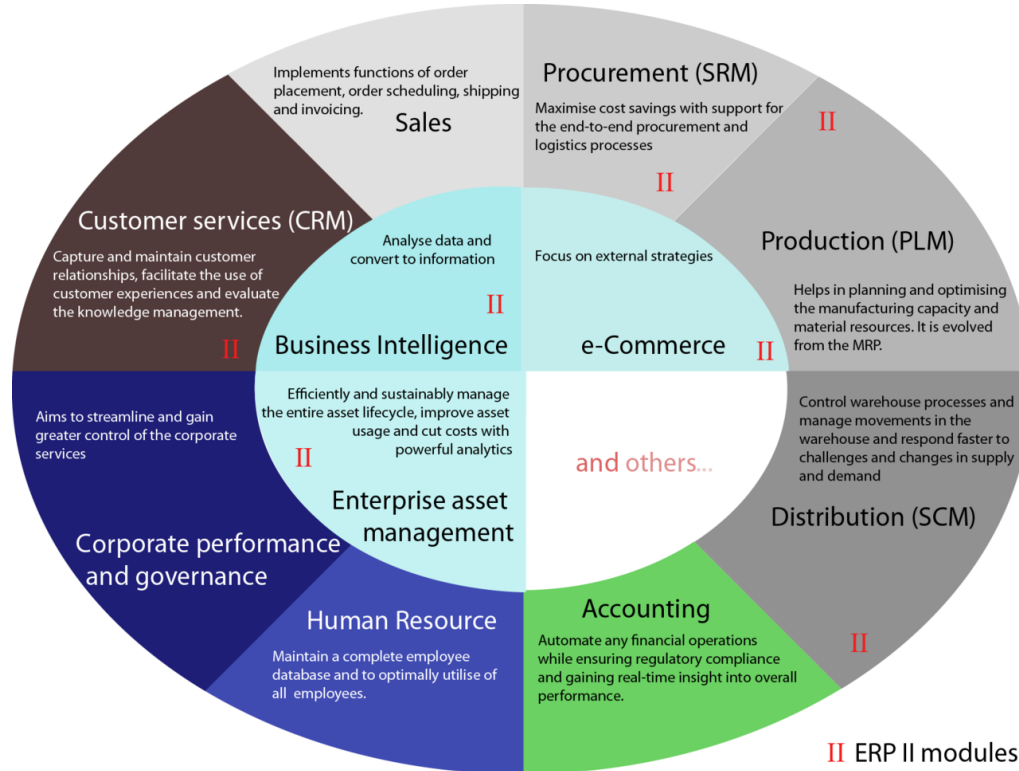
Existing middle-ware

- JBoss + custom implementation

Our tasks

- Create new middle-ware that takes into account changes in AX API and new e-shop

# ERP modules



# Our tasks

Why?

- Completely new e-shop
- Changes in AX API
- Changes in data flows

What?

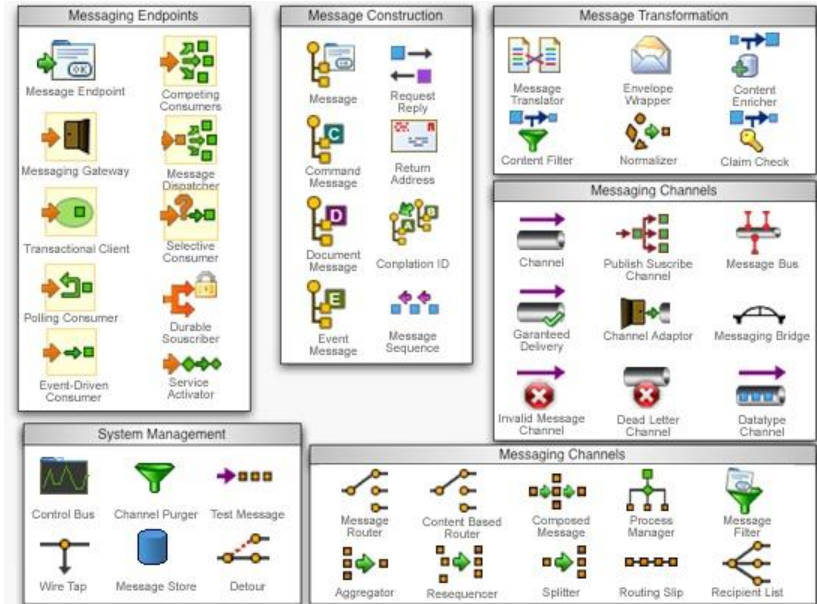
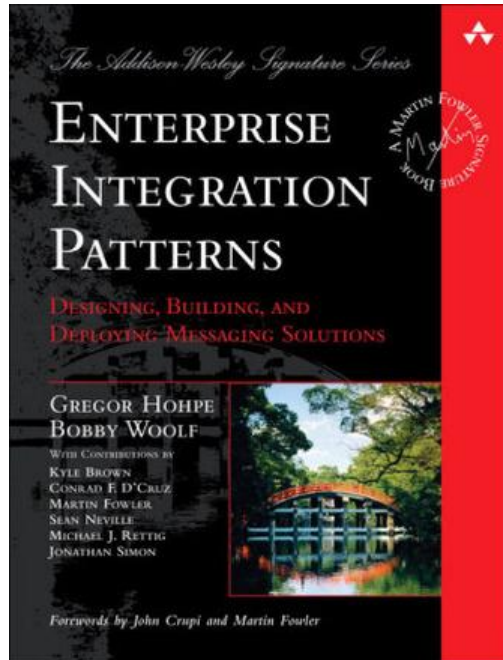
- Implementation of middle-ware between e-shop and AX
- Creation of monitoring console to monitor data flows
- **Our choice of technology**

# Middle-ware

## Apache Camel

- Java framework focused on easier integration
- Concrete implementation of used EIP (Enterprise Integration Patterns)
- Connects to wide spectrum of transports (facebook, Github, ...)
- Easy use of DSL (Domain Specific Language)
- Definition of data flows
- BeanBinding: defines which methods are invoked and how the message is converted into the parameters of the methods.
- WSDL (Web Services Description Language)
- Processor: Custom definition of conversion of the message

# Enterprise Integration Patterns



# Enterprise Integration Patterns

Message channel 

- Basic communication of two apps

Message Router 

- Sends message from app to chosen other app

Message Translator 

- Conversion from input message format to output message format

Dead Letter Channel 

- Treatment of undelivered message

Guaranteed Delivery 

- Confirmation about successful delivery of message

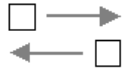
# Enterprise Integration Patterns

## Message Bus



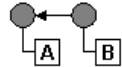
- Communication of more apps without effecting others

## Request Reply



- Specifies where the delivery subsystem responds

## Correlation Identifier



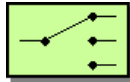
- How to assign response to concrete message

## Event Message



- Describes the sending of messages between subsystems

## Content Based Router

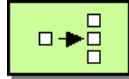


- Route message based on the content of the message



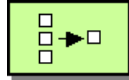
# Enterprise Integration Patterns

## Splitter



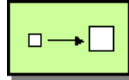
- Splits input message to parts

## Aggregator



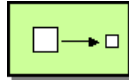
- Aggregates output message from parts to one

## Content Enricher



- Adding data to message, that it doesn't contain by default

## Content Filter



- Removing some unneeded data from message

## Log



- Logging information about processing the message

# Middle-ware

## ActiveMQ

- Saving and routing of messages among queues
- DeadQueue – if the sending of message fails, can be repeated

## Karaf

- Enables deploy of plugins or custom apps (monitoring console)

## JMX

- Enables access to MBeans
- Monitoring console receives data from server via Jolokia library

## Freemarker

- template language
- Parses XML and fills extracted values to predefined template

# Middle-ware – definition of data flow in Java

```
public class UpdatePricesRouteBuilder extends AbstractRouteBuilder {

    @Override
    public void configure() throws Exception {
        setupErrorHandler();

        from(getQueueUrl(FuseQueue.AXA_ESH_PRICES))
            .routeId(getRouteName())
            .setHeader("erpId", xpath("//price/erpId", String.class))
            .setHeader("recId", xpath("//price/recid", String.class))
            .choice()
                .when(xpath("//price/@deleted = 'true'"))
                    .setProperty("message", simple("Deleting price..."))
                    .bean("logDBBean", "logInfo")
                    .setProperty("message", simple("Setting headers..."))
                    .bean("logDBBean", "logDebug")
                    .removeHeaders("*, ENABLED_HEADERS")
                    .setHeader("CamelHttpMethod", constant("DELETE"))
                    .setProperty("message", simple("Sending request to eshop: "
                        + "{{endpoint.output.url.price}}erp-{{header.erpId}}"))
                    .bean("logDBBean", "logInfo")
                    .setHeader(Exchange.HTTP_URI, simple("{{endpoint.output.url.price}}erp-{{header.erpId}}"))
                    .to(getDummyHttpUrl())
                    .to(getQueueUrl(FuseQueue.CONFIRMATION)).endChoice()
                .otherwise()
                    .setProperty("message", simple("Updating price..."))
                    .bean("logDBBean", "logInfo")
                    .setHeader("CamelHttpMethod", constant("POST"))
                    .setProperty("message", simple("Converting message..."))
                    .bean("logDBBean", "logInfo")
                    .convertBodyTo(freemarker.ext.dom.NodeModel.class)
                    .to("freemarker:templates/UpdatePrices.ftl?contentCache=false")
                    .setProperty("message", simple("Setting headers..."))
                    .bean("logDBBean", "logDebug")
                    .removeHeaders("*, ENABLED_HEADERS")
                    .setHeader("Content-Type", constant("application/json;charset=utf-8"))
                    .setProperty("message", simple("Sending request to eshop: "
                        + "{{endpoint.output.url.price}}erp-{{header.erpId}};update=all"))
                    .bean("logDBBean", "logInfo")
                    .setHeader(Exchange.HTTP_URI, simple("{{endpoint.output.url.price}}erp-{{header.erpId}};update=all"))
                    .to(getDummyHttpUrl())
                    .to(getQueueUrl(FuseQueue.CONFIRMATION));
            }
    }
}
```

# Middle-ware – custom Processor

```
public class EndRouteBean {

    private LogDatabaseBean logBean;

    /**
     * Create response message.
     * @param e current exchange
     * @return resolved response
     */
    public UpdateResponseType process(Exchange e) {
        logBean.log("End route...", e, LogLevel.TRACE);

        String messageId = e.getIn().getMessageId();
        logBean.log("Route end: " + messageId, e, LogLevel.DEBUG);

        UpdateResponseType result = new UpdateResponseType();
        result.setResponse("OK");
        logBean.log("Setting response: " + result.getResponse(), e, LogLevel.DEBUG);
        return result;
    }

    public void setLogBean(LogDatabaseBean logBean) {
        this.logBean = logBean;
    }
}
```

# Middle-ware – Freemarker conversion

Conversion JSON / XML

From XML to JSON via  
predefined freemarker  
templates

From JSON to XML via  
XStream library

```
<#include "Common.ftl" />
<@compress single_line=true>
{
  <@string body.price, "productId" />
  <@number body.price, "price" />
  <@number body.price, "originalPrice" />
  <@number body.price, "vatRate" />
  <@string body.price, "validFrom" />
  <@string body.price, "validTo" />
  <@string body.price, "segment" />

  <#if body.price.properties?has_content>
    "properties" : {
      <#list body.price.properties.* as property>
        "${property?node_name}" : "${property?json_string}"
      <#sep>,</#sep>
      </#list>
    },
  </#if>

  <#if body.price.tags?has_content>
    "tags" : [
      <#list body.price.tags as tag>
        "${tag?json_string}"
      <#sep>,</#sep>
      </#list>
    ],
  </#if>

  <@string body.price, "erpId", false />
}
</@compress>
```

# Monitoring console – Jolokia access

## Logging to database

- In every data flow and processor
- JDBC connection to MySQL

## Monitoring console

- View of logs in database
- Filter and advanced search
- Access via Jolokia library

```
$scope.findLogBody = function () {  
  
    jolokia.request({  
        type: "exec",  
        operation: "getLogBody",  
        mbean: @REPLACE_THIS@.settings.mbean,  
        arguments: [$scope.myRowDetail.id]  
    }, {  
        method: "POST",  
        success: function (response) {  
            var beautyBody = "";  
            if (response.value) {  
                var trimmed = response.value.trim();  
                var firstChar = trimmed.charAt(0);  
                if (firstChar == "<") {  
                    beautyBody = vkbeautify(trimmed, "xml");  
                } else if (firstChar == '{') {  
                    beautyBody = vkbeautify(trimmed, "json");  
                } else {  
                    beautyBody = trimmed;  
                }  
            }  
            $scope.myRowDetail.body = beautyBody;  
  
            Core.$apply($scope);  
        },  
        error: function (response) {  
            $scope.response = response.error + " *** " + response.stacktrace;  
            Core.$apply($scope);  
        }  
    })  
};
```

# Monitoring console

**RED HAT JBOSSE FUSE** Management Console admin

ActiveMQ Camel Connect Dashboard **Datart CZ** Health Jetty JMX Logs OSGi Terminal Threads

Datart Monitoring CZ

Search:

Flow Name: Rec id: Erp id: Message ID: Time range: ALL TIME Log level: INFO

Filter Reset

#	Flow	RecId	ErpId	Message ID	Created	Level	Message
979	DEV-UpdateAttributesRouteBuilder	5640104335	20000021	3702-1504777096140-5-7	2017-09-07 13:35:52	INFO	Sending request to eshop: http://localhost:1080/intapi/catalog/attr/erp-20000021:update=all
980	DEV-ConfirmationRouteBuilder	5640104335	20000021	3702-1504777096140-5-8	2017-09-07 13:35:52	INFO	Preparing confirmation...
981	DEV-ConfirmationRouteBuilder	5640104335	20000021	3702-1504777096140-5-8	2017-09-07 13:35:52	INFO	Sending confirmation to Axapta...
982	DEV-InputRouteBuilder			3702-1504777096140-5-11	2017-09-07 13:35:54	INFO	New request detected...
983	DEV-DecisionRouteBuilder			3702-1504777096140-5-15	2017-09-07 13:35:54	INFO	Request type: update attributes
984	DEV-UpdateAttributesRouteBuilder	5639592907	X123456	3702-1504777096140-5-16	2017-09-07 13:35:54	INFO	Deleting attribute...
985	DEV-UpdateAttributesRouteBuilder	X123456		3702-1504777096140-5-16	2017-09-07 13:35:54	INFO	Sending request to eshop: http://localhost:1080/intapi/catalog/attr/erp-X123456
986	DEV-ConfirmationRouteBuilder	5639592907	X123456	3702-1504777096140-5-17	2017-09-07 13:35:54	INFO	Preparing confirmation...
987	DEV-ConfirmationRouteBuilder	5639592907	X123456	3702-1504777096140-5-17	2017-09-07 13:35:54	INFO	Sending confirmation to Axapta...
988	DEV-Input			3702-1504777096140-7-2	2017-09-07 13:37:26	INFO	New request detected...
989	DEV-Decision			3702-1504777096140-7-6	2017-09-07 13:37:26	INFO	Request type: update attributes
990	DEV-UpdateAttributes	5640104335	20000021	3702-1504777096140-7-7	2017-09-07 13:37:26	INFO	Updating attribute...
991	DEV-UpdateAttributes	5640104335	20000021	3702-1504777096140-7-7	2017-09-07 13:37:26	INFO	Converting message...
992	DEV-UpdateAttributes	5640104335	20000021	3702-1504777096140-7-7	2017-09-07 13:37:26	INFO	Sending request to eshop: http://localhost:1080/intapi/catalog/attr/erp-20000021:update=all
993	DEV-Confirmation	5640104335	20000021	3702-1504777096140-7-8	2017-09-07 13:37:26	INFO	Preparing confirmation...
994	DEV-Confirmation	5640104335	20000021	3702-1504777096140-7-8	2017-09-07 13:37:26	INFO	Sending confirmation to Axapta...
995	DEV-Input			3702-1504777096140-7-11	2017-09-07 13:37:28	INFO	New request detected...
996	DEV-Decision			3702-1504777096140-7-15	2017-09-07 13:37:28	INFO	Request type: update attributes
997	DEV-UpdateAttributes	5639592907	X123456	3702-1504777096140-7-16	2017-09-07 13:37:28	INFO	Deleting attribute...
998	DEV-UpdateAttributes	5639592907	X123456	3702-1504777096140-7-16	2017-09-07 13:37:28	INFO	Sending request to eshop: http://localhost:1080/intapi/catalog/attr/erp-X123456
999	DEV-Confirmation	5639592907	X123456	3702-1504777096140-7-17	2017-09-07 13:37:28	INFO	Preparing confirmation...
1000	DEV-Confirmation	5639592907	X123456	3702-1504777096140-7-17	2017-09-07 13:37:28	INFO	Sending confirmation to Axapta...

# Monitoring console

RED HAT JOSS FUSE Management Console

ActiveMQ Camel Connect Dashboard **Datart CZ** Health Jetty JMX Logs OSGi Terminal Threads

### Datart Monitoring CZ

Search:

Flow Name: Rec id: Erp id: Message ID: Time range: ALL TIME Log level: INFO

Filter Reset

#	Flow	RecId	ErpId	Message ID	Created	Level	Message
979	DEV-UpdateAttributesRouteBuilder	5640104335	20000021	3702-1504777096140-5-7	2017-09-07 13:35:52	INFO	Sending request to eshop: http://localhost:1080/intapi/catalog/attr/erp-20000021:update=all
980	DEV-ConfirmationRouteBuilder	5640104335	20000021	37			
981	DEV-ConfirmationRouteBuilder	5640104335	20000021	37			
982	DEV-InputRouteBuilder			37			
983	DEV-DecisionRouteBuilder			37			
984	DEV-UpdateAttributesRouteBuilder	5639592907	X123456	37			
985	DEV-UpdateAttributesRouteBuilder	5639592907	X123456	37			
986	DEV-ConfirmationRouteBuilder	5639592907	X123456	37			
987	DEV-ConfirmationRouteBuilder	5639592907	X123456	37			
988	DEV-Input			37			
989	DEV-Decision			37			
990	DEV-UpdateAttributes	5640104335	20000021	37			
991	DEV-UpdateAttributes	5640104335	20000021	37			
992	DEV-UpdateAttributes	5640104335	20000021	37			
993	DEV-Confirmation	5640104335	20000021	37			
994	DEV-Confirmation	5640104335	20000021	37			
995	DEV-Input			37			
996	DEV-Decision			37			
997	DEV-UpdateAttributes	5639592907	X123456	37			
998	DEV-UpdateAttributes	5639592907	X123456	37			
999	DEV-Confirmation	5639592907	X123456	37			
1000	DEV-Confirmation	5639592907	X123456	3702-1504777096140-7-17	2017-09-07 13:37:28	INFO	Sending confirmation to Axapta...

990 / 1000

Flow: DEV-UpdateAttributes  
Rec ID: 5640104335  
Erp ID: 20000021  
Message ID: 3702-1504777096140-7-7  
Created: 2017-09-07 13:37:26  
Level: INFO  
Message: Updating attribute...

Body:

```
<attribute deleted="false">
  <recid>5640104335</recid>
  <erpId>20000021</erpId>
  <title>Komunikační displej</title>
  <article>Komunikační displej</article>
  <valueType>ENUM</valueType>
  <urlId>komunikacni-displej</urlId>
  <unit>kg</unit>
</tag>
<tag>
  <tag>FILTER_HIDE</tag>
  <tag>TAG1</tag>
  <tag>TAG2</tag>
</tag>
```



# List of used technologies

XML, HTTP, WSDL, SOAP, REST, JSON

Camel, ActiveMQ, Karaf, JMX

MySQL

Javascript, Angular, HTML5, CCS3

Freemarker

Synchronous/asynchronous message delivery

- After e-shop request wait for AX, after AX response send confirmation and message to e-shop
- After AX request respond 200-OK immediately, after parsing of message send message to e-shop

# Testing

## Integration tests

- testing, if message and confirmation are sent and received correctly
- SoapUI + mockserver: imitation of e-shop and AX

## Performance tests

- JMeter for simpler tests without need of mockserver
- Creation of simple server (only responds 200-OK) and script that measures performance and queues via JMX

Thank you for your attention!

