Connecting Software
An overview about challenges, solutions and the need for software-based integration tools.

Henrik Gerdes

hegerdes@uos.de

January 28, 2020

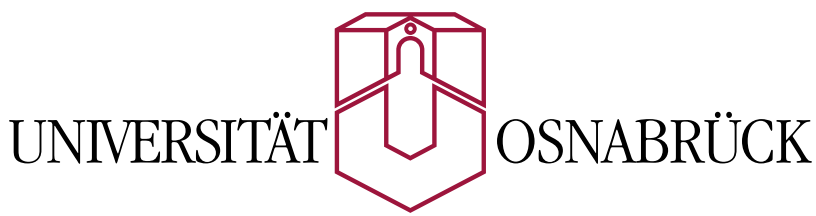


Table of Contents

Outline

Software Requirements

Source: edgeup.asus.com

Software should be:

- Functional
- Reliable
- Extendable
- Performant
- Predictable
- User friendly

Software grows

New Functions	
Support Social Media	Adding a In-App Store

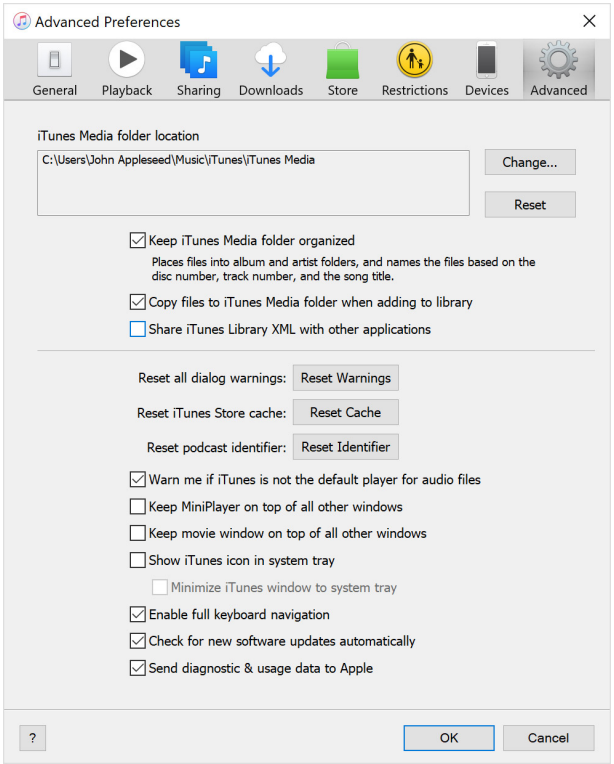
Adpot new Technologies	
HTML5	IPv6

Extended Plattform support	
Support for Android	Support for IOS

Bloated Software

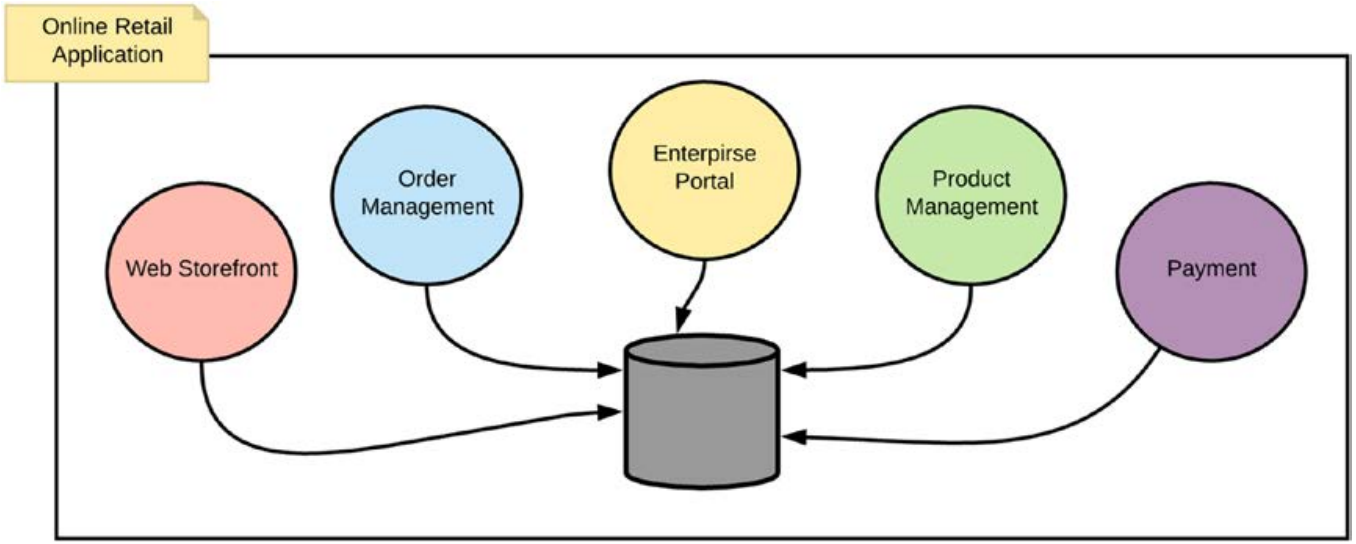
Remember the „all-rounder“ nero or discontinued ITunes?

Source: winfuture.de



Source: support.apple.com

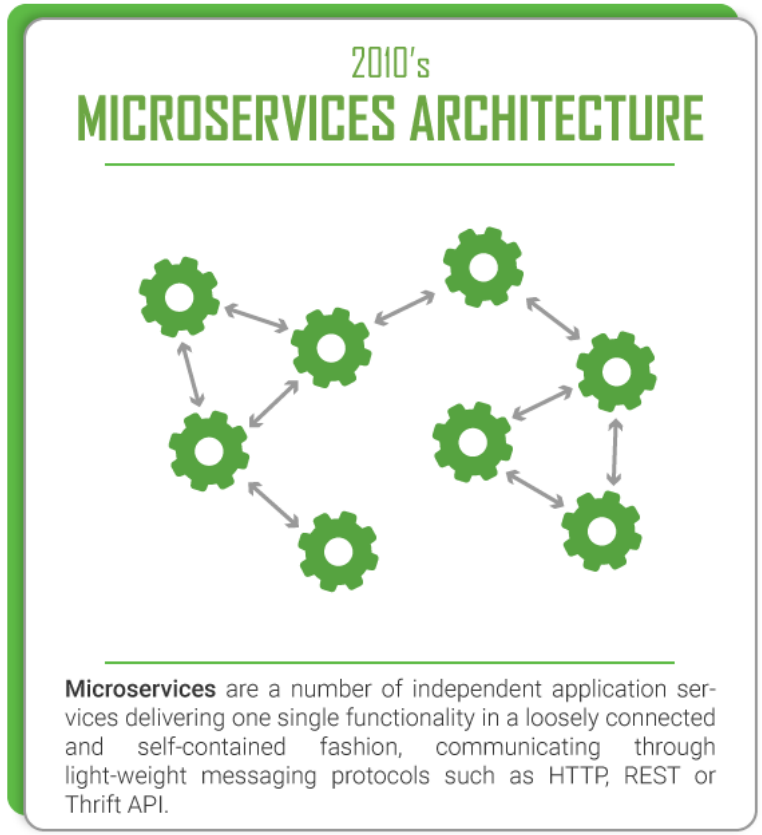
In enterprise



Source: [IS18]

⇒ Won't result in ONE good application

SOA and MSA



Source: dzone.com

Solutions

SOA

- Service $\equiv$ Component
- Self-contained Services
- One service per business logic
- Communication through an ESB

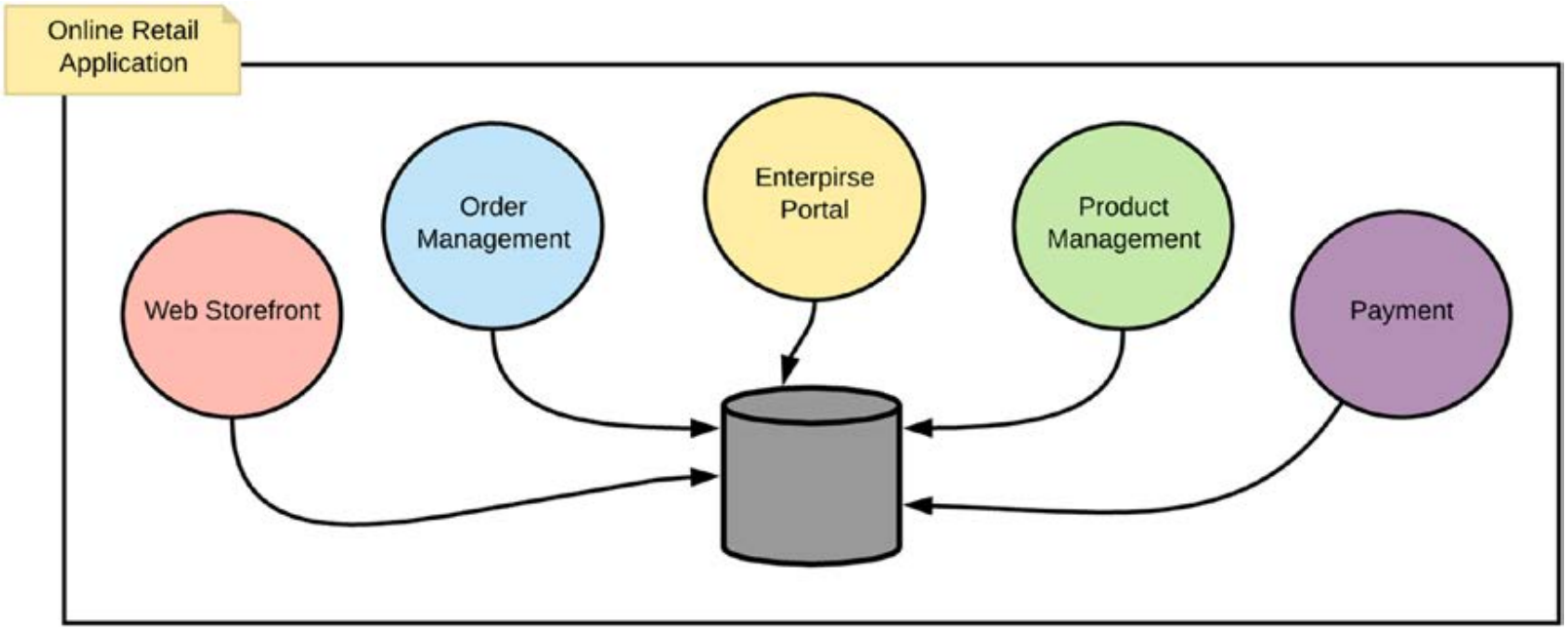
MSA

- Finer grained
- Independent Services
- Allows different technologies
- No central component
- Independent deployable

Outline

Integration Problems

Web-Frontend sends order to Backend.
What could go wrong?



Source: [IS18]

Integration Problems

A LOT

Direct integration

Advantages:

- Tightly coupled::
  - Same data format
  - Same platform
  - Direct API calls
- No extra application
- Homogeneously service environment
- Existing structure

Problems:

- Platform and framework restrictions
- Effort increases by growing components
- Single point of failure.
- Lack of scalability
- Hard to adopt new technologies

Network Problems

`Little Endian`

IBM 80x86

Port 4242

⇓

`1000010010010`

≠

`Big Endian`

Sun (Ultra) Sparc

Port 2337

⇓

`100100100001`

Network Problems

Networks are slow

Networks are unreliable

32 bit Int *vs* 64 bit Int

IPv4 *vs* IPv6

...

Many more

Outline

Enterprise Service Bus

How does an ESB solve this?

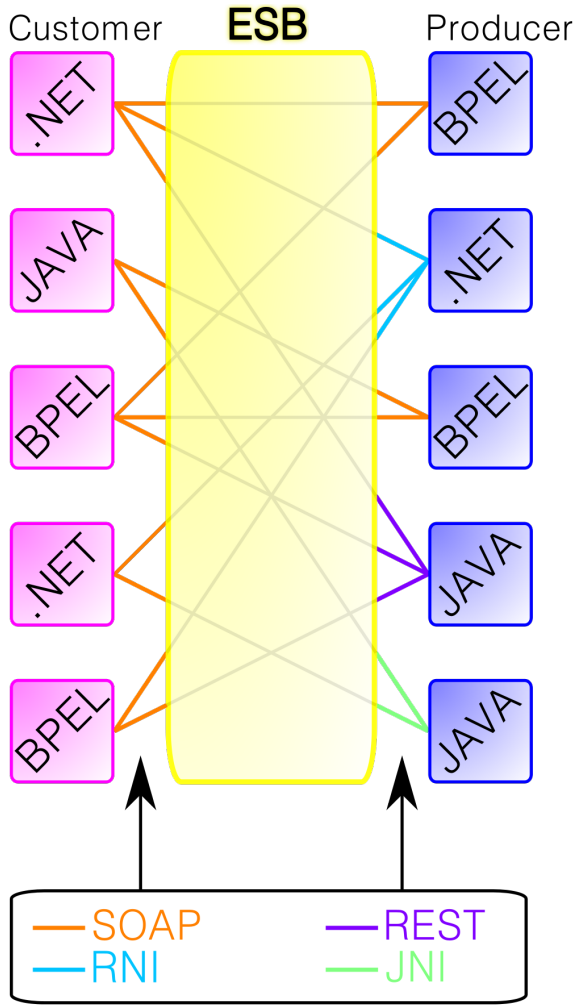
Enterprise Service Bus

Avoids Point to Point communication

An ESB adds abstraction layers

It solves most of the network problems

Uses self-describing data structures



Source: WikiMedia

ESB DEMO

List of Protocolls

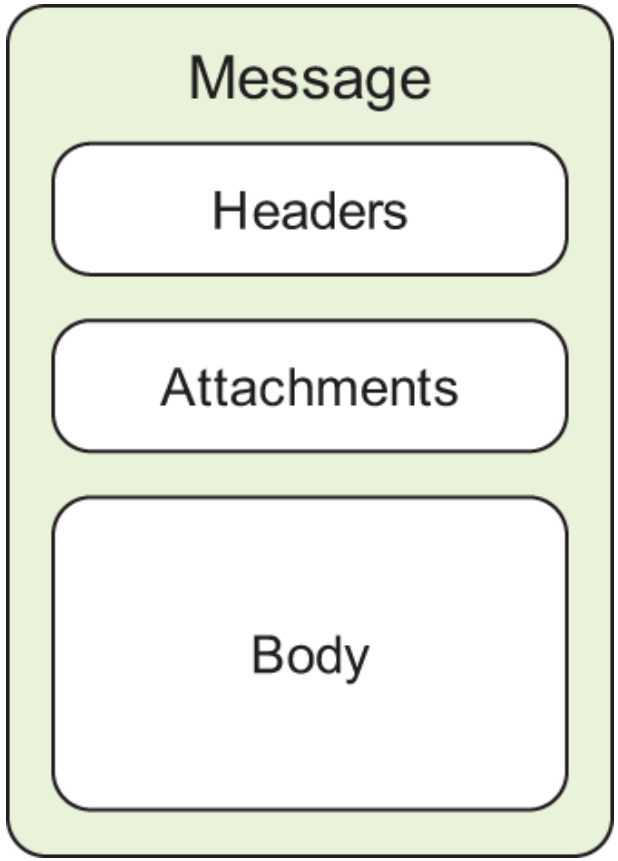
- Atmos
- AWS DynamoDB
- AWS S3 Storage
- Bean
- Box
- Cassandra CQL
- CouchDB
- DigitalOcean
- DNS
- Docker
- Dropbox
- Facebook
- File
- FTP
- Git
- Google Calendar
- Google Drive
- Google Mail
- Google Sheets
- GraphQL
- HTTP
- Jira
- JSON
- Kubernetes
- LDAP
- Mail
- MongoDB
- PDF
- PostgresSQL
- Printer
- REST
- RSS
- SAP
- Slack
- Spark
- Splunk
- Spring
- SQL
- SSH
- Telegramm
- Twitter
- WordPress

Enterprise Service Bus

How does it work internally?

Abstraction

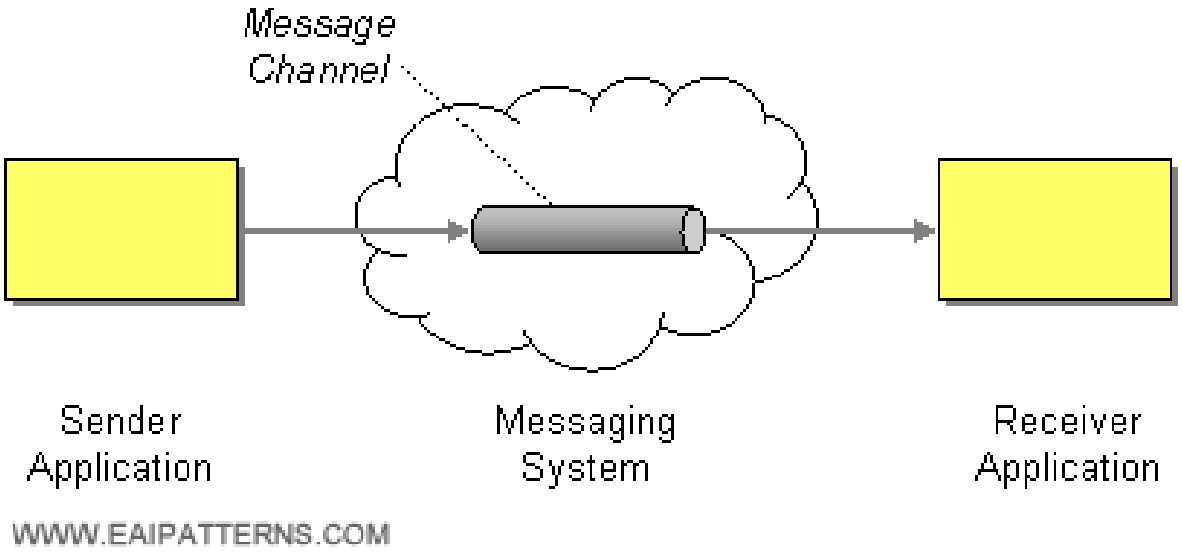
Data to share =



Source: [IA18]

Transportation

Transportation of a massage:

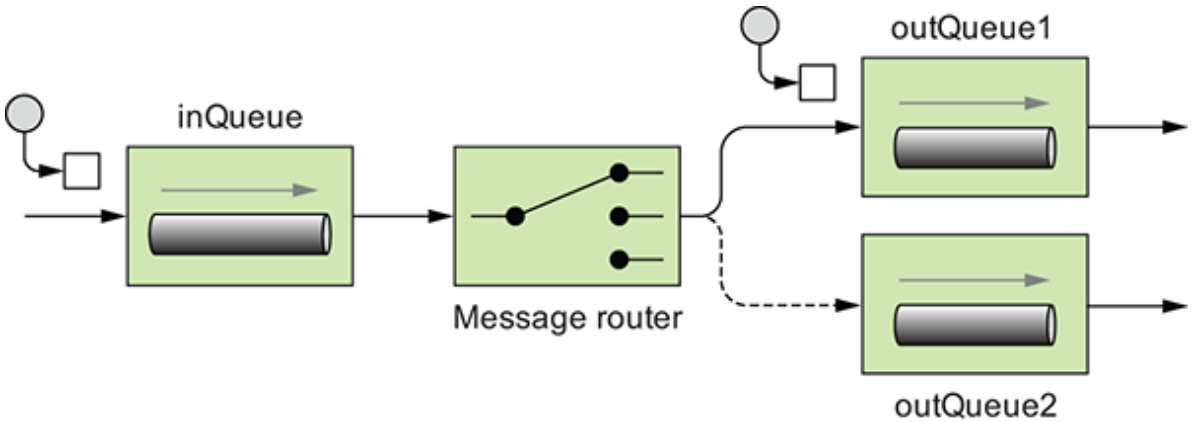


Source: [HW04]

Routing

Routing from one channel to others:

⇓



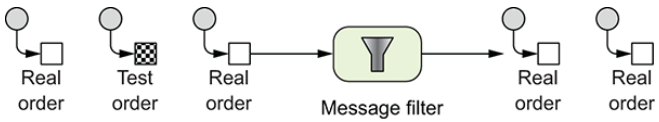
Source: [IA18]

Filter & Transform

Between channels and routing you can:

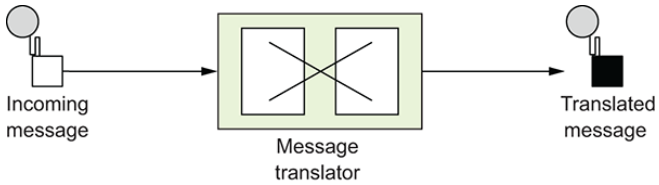
⇓

Filter



Source: [IA18]

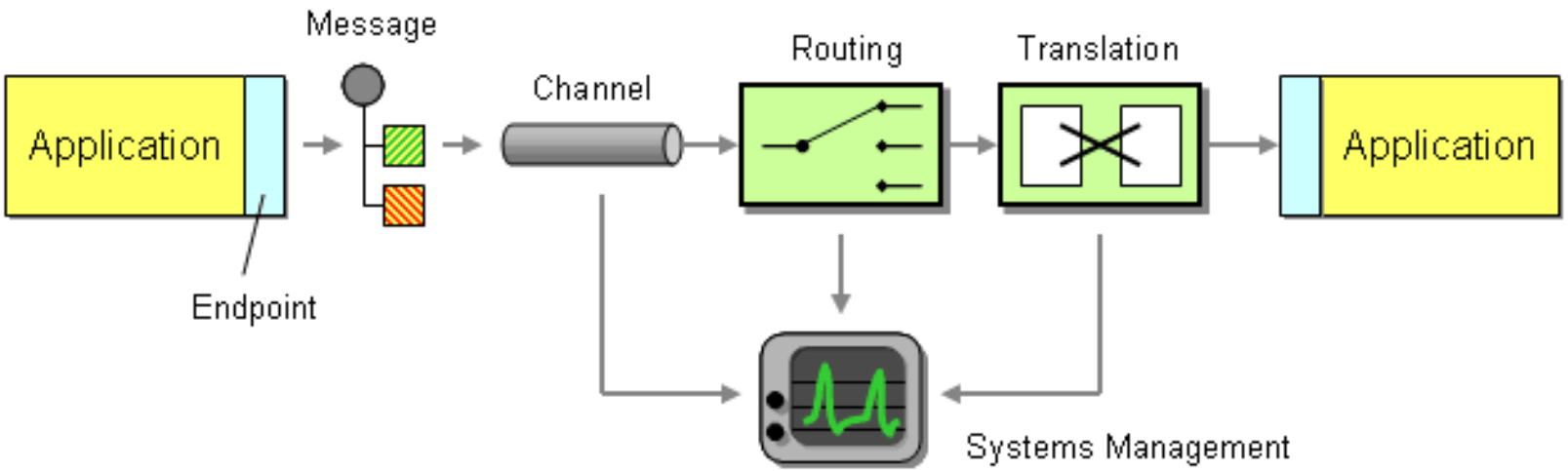
Transform



Source: [IA18]

Put everything together:

ESB-Model



Source: [HW04]

ESB in short

1. Connector use the native application API
2. Adds a communication layer between applications
3. Transportation by channels
4. Routing between channels
5. Messages can be queued, filtered and transformed

ESB DEMO

Routing

```
15 <camelContext xmlns="http://camel.apache.org/schema/spring">
16   <route>
17     <from uri="ftp://rider.com/orders?username=rider&amp;
           password=secret"/>
18     <to uri="jms:incomingOrders"/>
19   </route>
20 </camelContext>
```

Listing 1: Camel XML Route

Options

Use XML for routing

Use several ESB instances

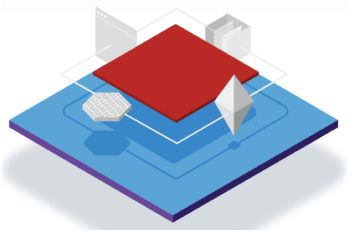
Use own components and Endpoints

Additional integration tools

Outline

Other ESBs

Red Hat Fuse



Source: redhat.com

spring Integration

Source: medium.com

MuleSoft

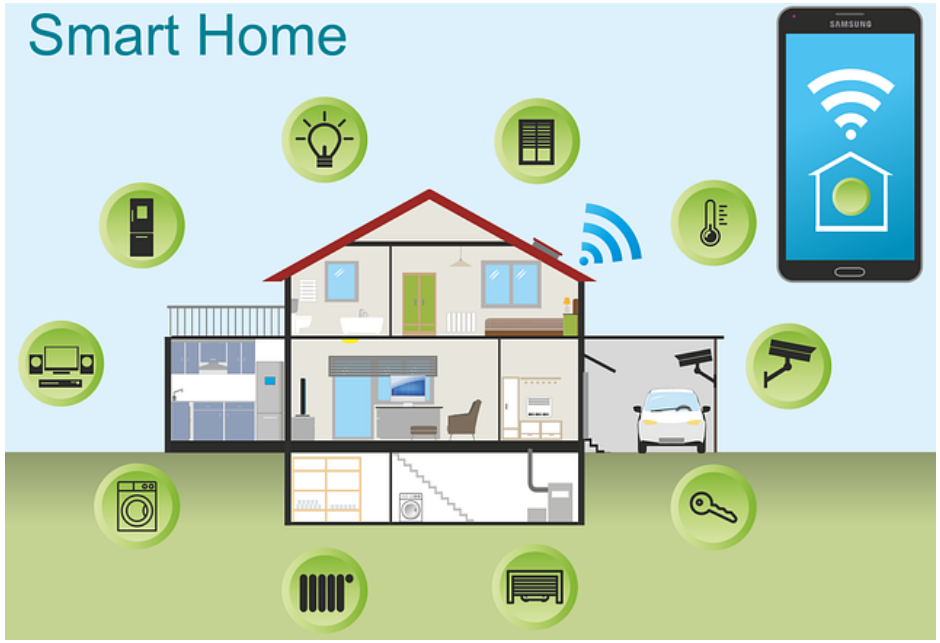
Source: mulesoft.com

1. Fuse
   - Integration framework by RedHat.
   - Based on Apache Camel.
   - Extends RedHat Services and focuses on cloud infrastructure.
2. Spring Integrating
   - Part of Spring Framework.
   - Aims for Plain Old Java Object.
   - Ships with commons endpoints.
3. Mule
   - Application specific ESB.
   - For enterprise software like SAP.
   - More service orientated.

Home Usage

Why should I?

- Extends functionality
- Variety manufacturers
- Avoids locked-in
- Automation
- Control
- Comfort

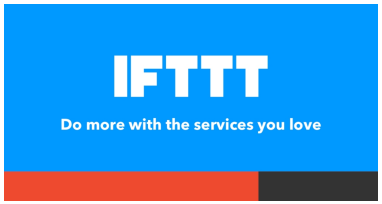


Source: it-wegweiser.de

Home Services

Microsoft Flow

Source: bluedock.dk



Source: a1blog.net

Source: zapier.com

1. Flow
   - Connects different webservices
   - Templates for common services.
   - Trigger - Event based.
2. IFTTT
   - More general alternative to Flow.
   - Automates recurring tasks.
   - Supports IoT and SmartHome devices.
3. Zapier
   - More business orientated.
   - Over 1500 Connector.
   - Greater customization.

FLOW DEMO

Questions?

Thank you for your attention!
Any questions?

References I

- Gregor Hohpe and Bobby Woolf.
  Enterprise integration patterns: Designing, building, and deploying messaging solutions.
  Addison-Wesley Professional, 2004.
- Claus Ibsen and Jonathan Anstey.
  Camel in action.
  Manning Publications Co., 2018.
- Kasun Indrasiri and Prabath Siriwardena.
  Microservices for the enterprise.
  Apress, Berkeley, 2018.

Routing

```
1  <beans xmlns="http://www.springframework.org/schema/beans"
2    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3    xsi:schemaLocation="http://www.springframework.org/schema/beans
4    http://www.springframework.org/schema/beans/spring-beans.xsd
5    http://camel.apache.org/schema/spring
6    http://camel.apache.org/schema/spring/camel-spring.xsd">
7
8    <bean id="jms" class="org.apache.camel.component.jms.JmsComponent">
9      <property name="connectionFactory">
10       <bean class="org.apache.activemq.ActiveMQConnectionFactory">
11         <property name="brokerURL" value="vm://localhost" />
12       </bean>
13     </property>
14   </bean>
15   <camelContext xmlns="http://camel.apache.org/schema/spring">
16     <route>
17       <from uri="ftp://rider.com/orders?username=rider&amp;password=secret"/>
18       <to uri="jms:incomingOrders"/>
19     </route>
20   </camelContext>
21 </beans>
```

Listing 2: Camel XML Route - Full