Application Integration with SOA and ROA

PV207 – Business Process Management

Spring 2023

Jiří Kolář, Lukáš Smiga, Lubomír Hruban

Last lecture recap

- Processes
    - **What is business process?**

Last lecture recap

- Processes
  - What is business process?
  - **What is BPM?**

Business Process Management

Is a **Management discipline**, focused on systematic **definition** and **execution measurement of processes** in organizations

- **An effort to describe processes** in organisation measure results and **manage process changes** towards higher efficiency

- **"Evolution not Revolution"**

Last lecture recap

- Processes
  - What is business process?
  - What is BPM?
  - **What is BPM adoption?**

Last lecture recap

- Processes
  - What is business process?
  - What is BPM?
  - What is BPM adoption?
  - Why BPM ?
  - Roles in BPM
  - Process life-cycle
  - Phases of process based development
- BPMS
  - BPMS components
  - Architecture
  - Human Tasks
  - Business Rules
  - BAM
  - Existing BPMS

Lecture summary

- Generations of EAI
- Motivation for SOA
- Role BPM in IT management
- Core BPM architecture
- BPM – SOA relationship
- Microservices
- Web Services
  - What are WS?
  - Artifacts WS
    - WSDL
    - SOAP
  - WS - standards
- WS in Java
- REST

3 meanings of the word "service"

- ....

3 meanings of the word "service"

- "Business" service
  - Google offers paid advertising to restaurants
  - Defined by contract / service offering

3 meanings of the word "service"

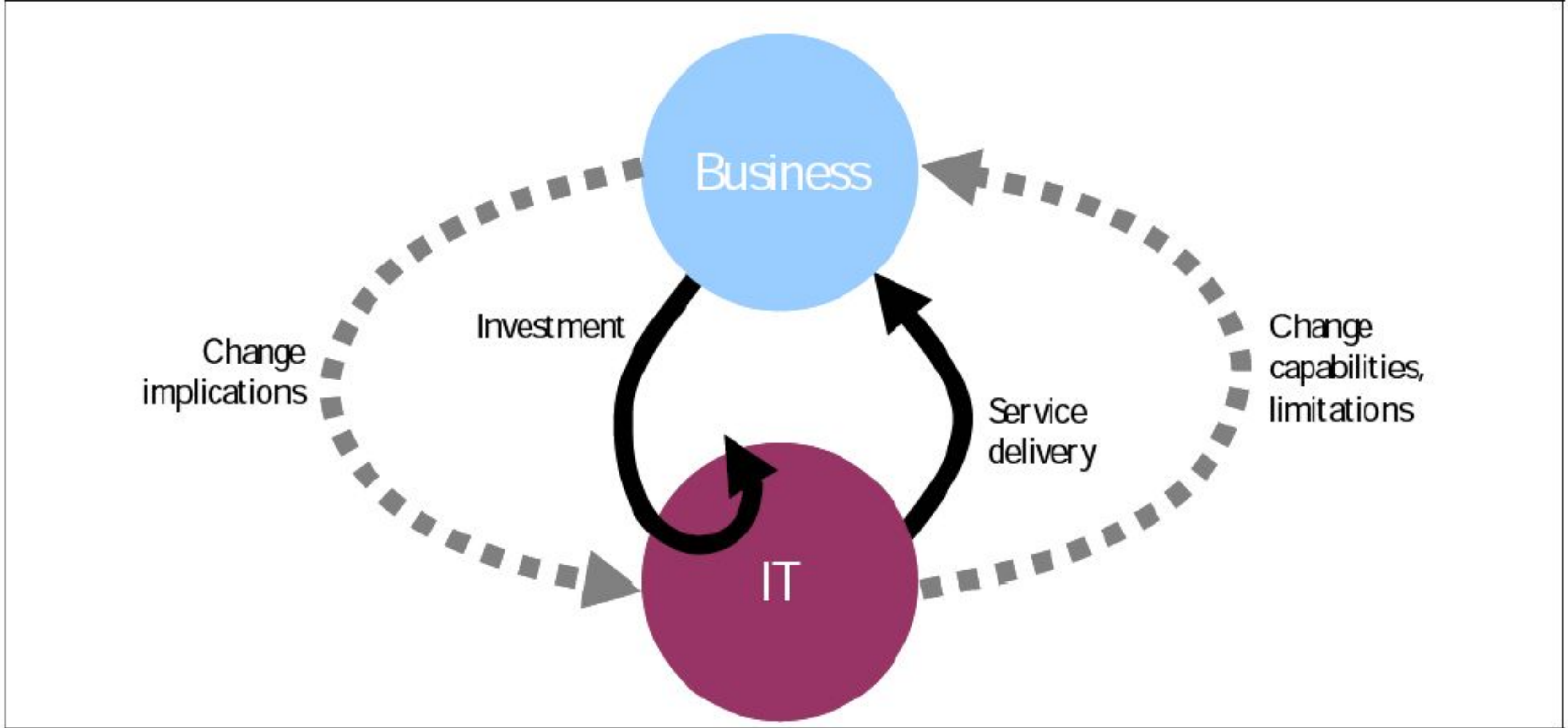
- "Business" service
  - Google offers paid advertising to restaurants
  - Defined by contract / service offering
- "Technical" service
  - Google provides a search for addresses of restaurants in neighbourhood
  - Defined by a User Interface / Programming interface

3 meanings of the word "service"

- "Business" service
    - Google offers paid advertising to restaurants
    - Defined by contract / service offering
- "Technical" service
    - Google provides a search for addresses of restaurants in neighbourhood
    - Defined by a User Interface / Programming interface
- Web Service
    - Google provides Web Service API for retrieving GPS coordinates of particular address
    - Defined by a WSDL/REST methods definition
    - Request - response model

Business & IT alignment

Figure 1: The elements of IT-business alignment



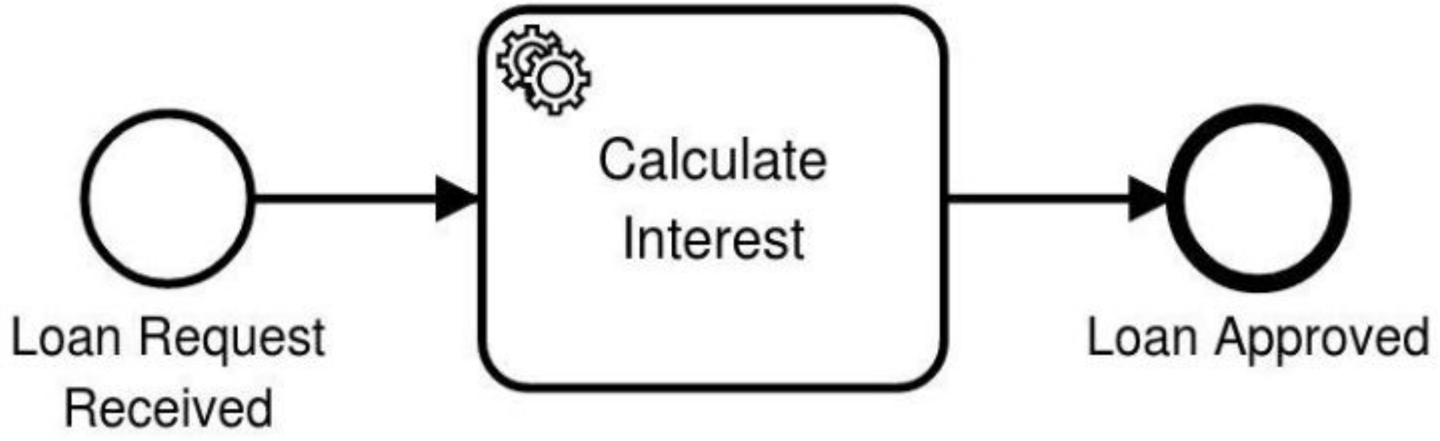
There are three important elements in IT-business alignment: investment, service delivery, and collaboration in change management.

Macehiter Ward-Dutton

Enterprise Application Integration

Why application integration?

- Allow different applications to share data and processes states.
- In BPMS systems we use the Service Task to directly invoke some functionality.



Services Examples (IT/Web)

- ….

Services Examples (IT/Web)

- createUserProfile
- setUserStatus
- searchFlights
- returnAccountBallance

- PUT /v2/user/{id}/profile
- GET /account/{id}/balance

EAI Generations – spaghetti

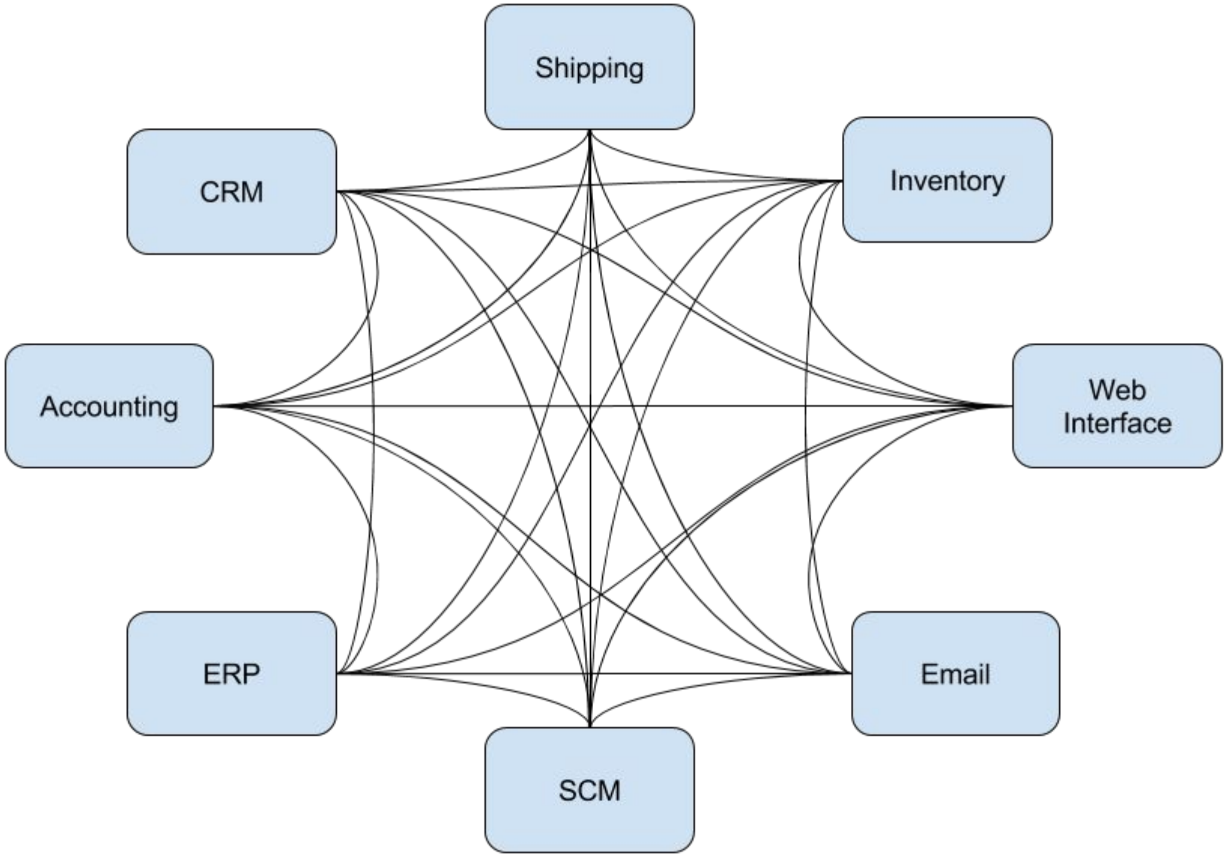


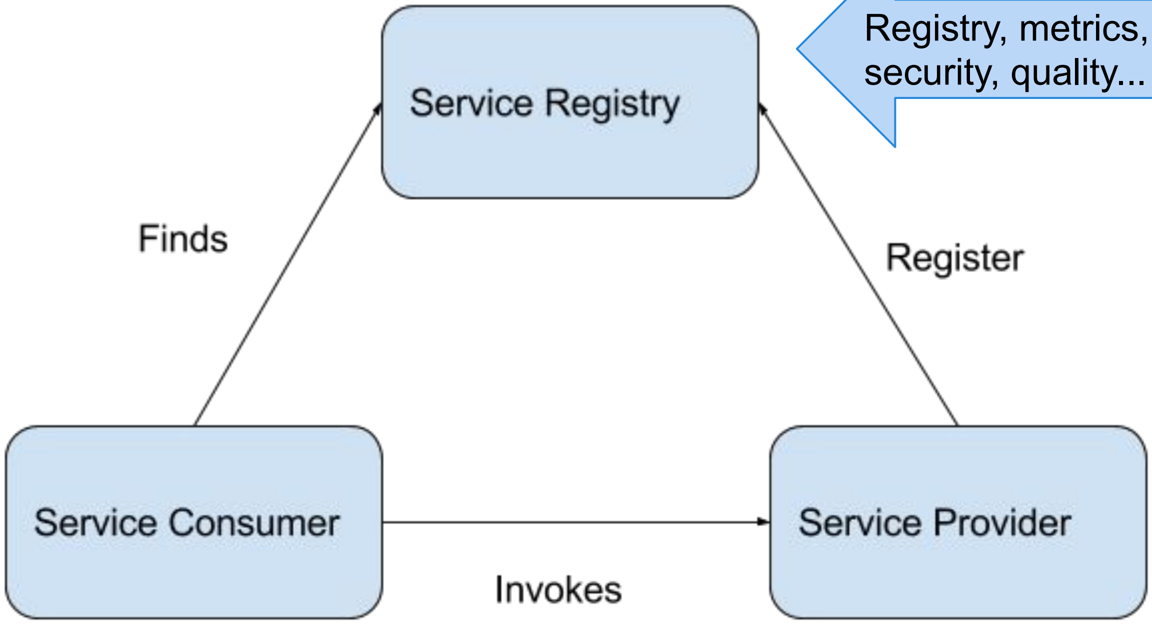
Figure 2: Spaghetti Integration

Different communication protocols and principles:

- File exchange
- DB access
- MQ messaging
- CORBA
- Web Services
- Proprietary connectors

https://dzone.com/articles/building-integration-solutions-a-rethink

EAI Generations – SOA



Registry, metrics, governance, security, quality...

Figure 1: Service Oriented Architecture

https://dzone.com/articles/building-integration-solutions-a-rethink

EAI Generations – SOA and ESB

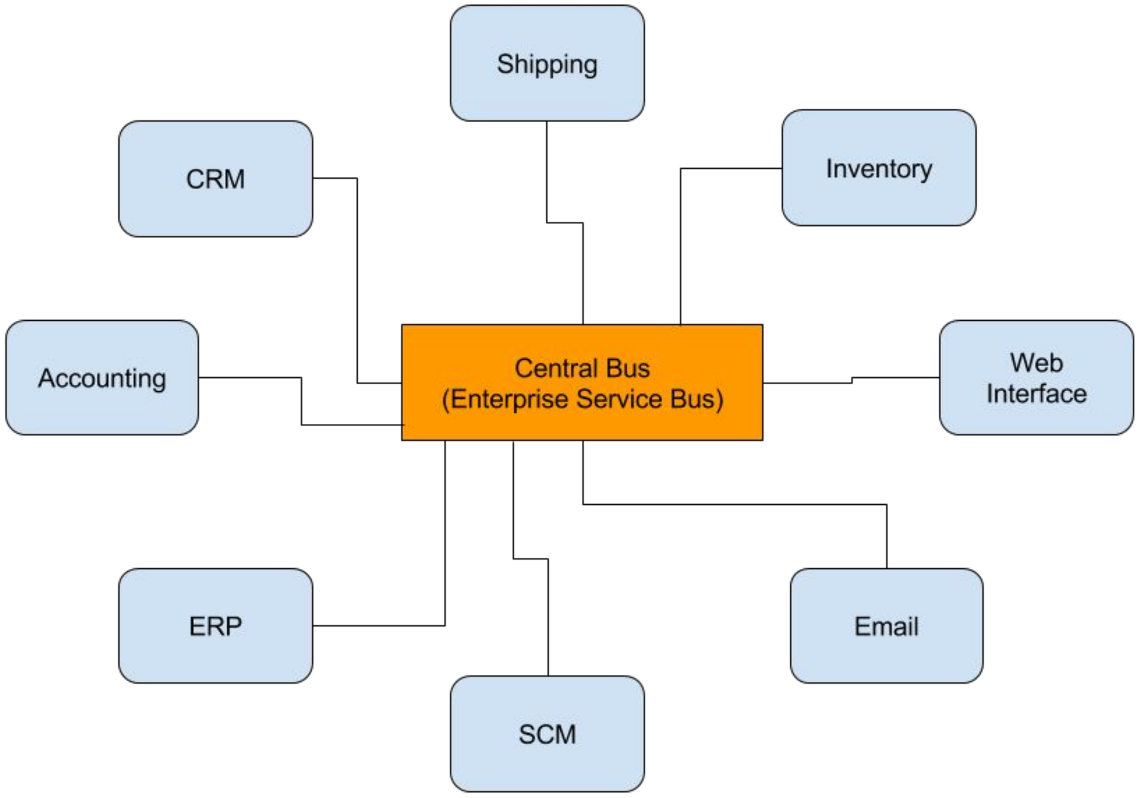
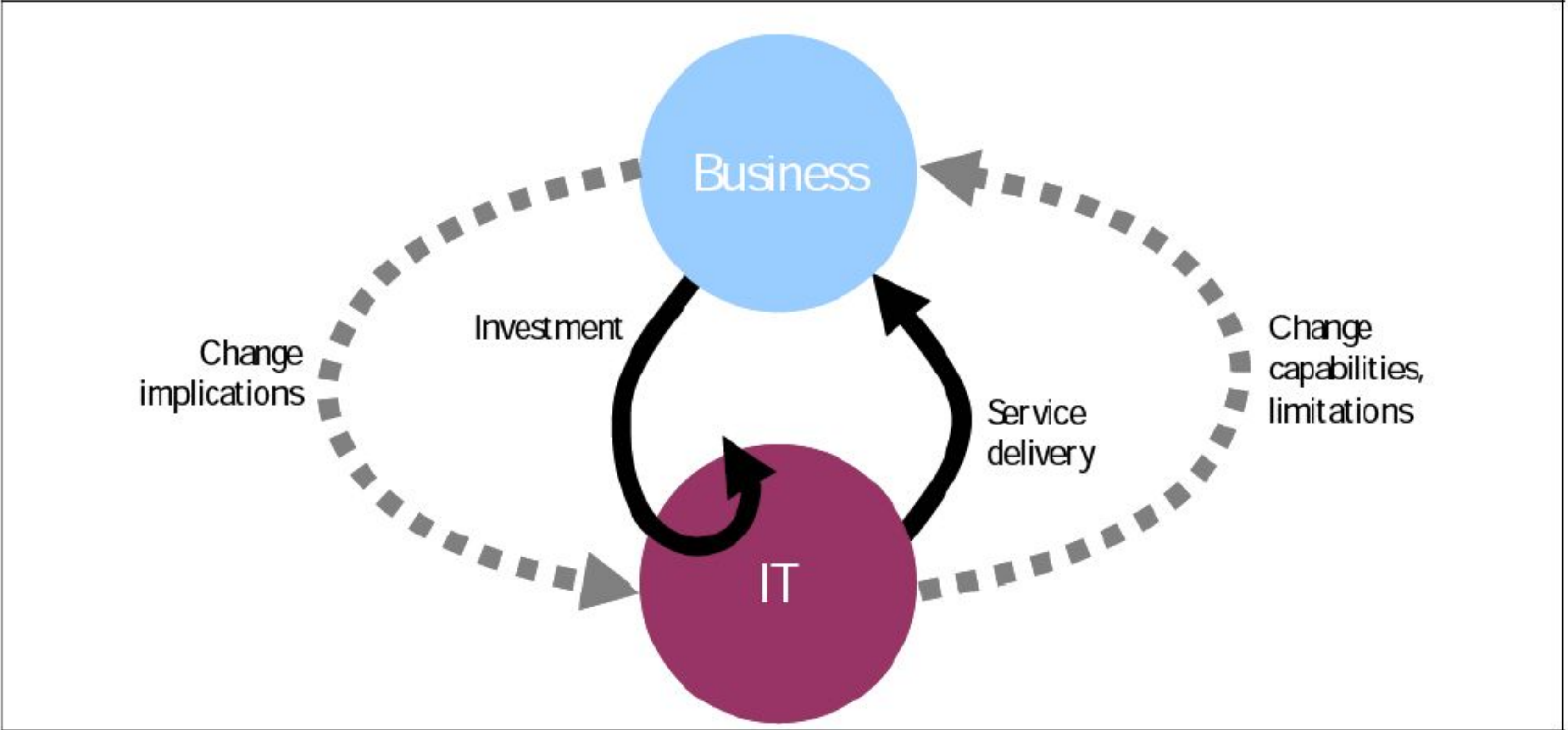


Figure 3: Bus Integration

https://dzone.com/articles/building-integration-solutions-a-rethink

Business & IT alignment

Figure 1: The elements of IT-business alignment



There are three important elements in IT-business alignment: investment, service delivery, and collaboration in change management.

Macehiter Ward-Dutton

SOA motivation

- **Reduction of costs** on development and integration
- Efficient **maintenance and integration** across various systems
- Component/service **reusability**
- Integration of **Legacy applications**
- Efficient **management and monitoring**
- **Just-in-time management** (real time business)

SOA definition

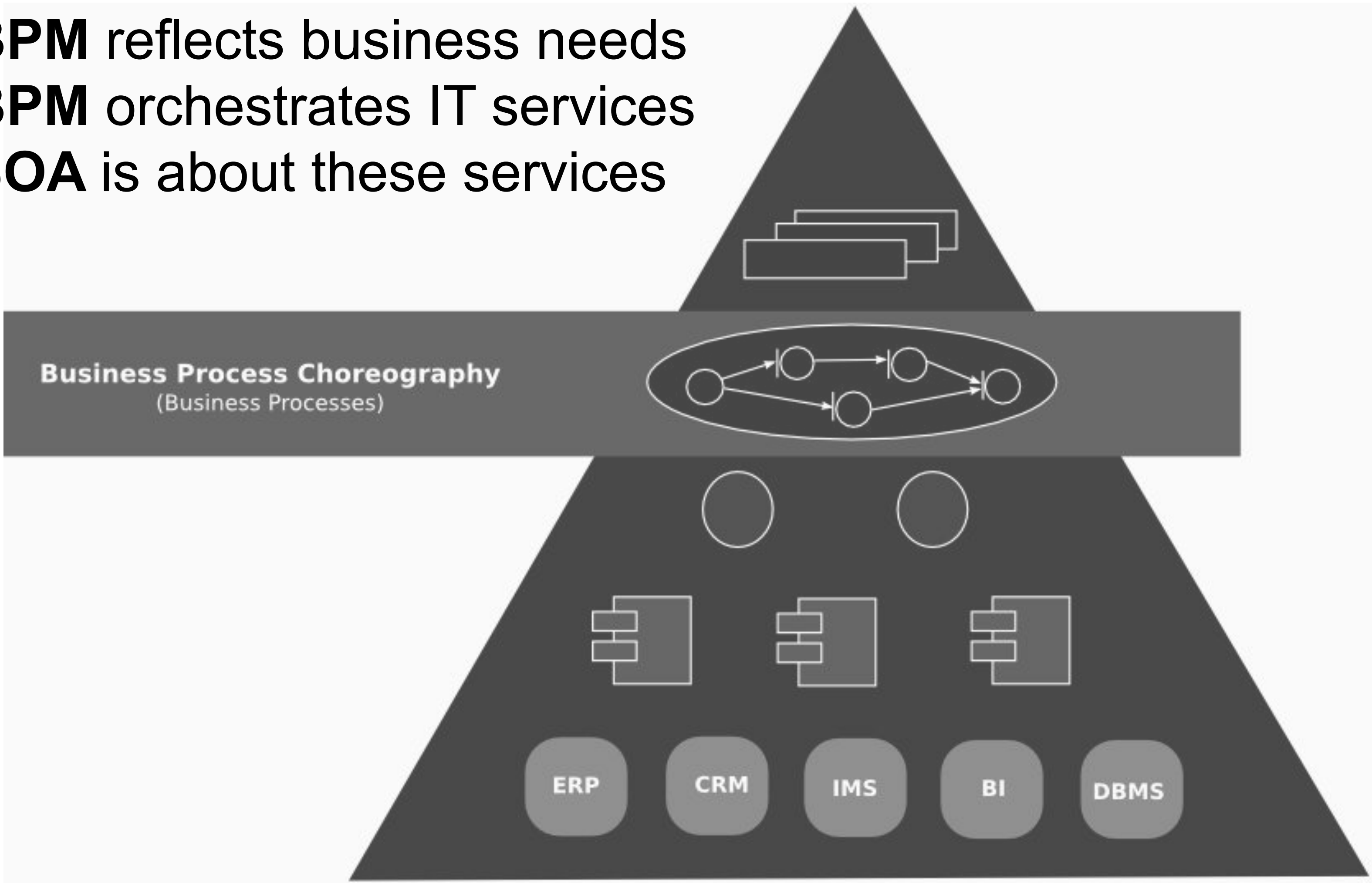
Service-Oriented Architecture *(SOA)* is an *architectural style* that supports *service-orientation*.

Service-orientation is a way of thinking in terms of services and service-based development and the outcomes of services.

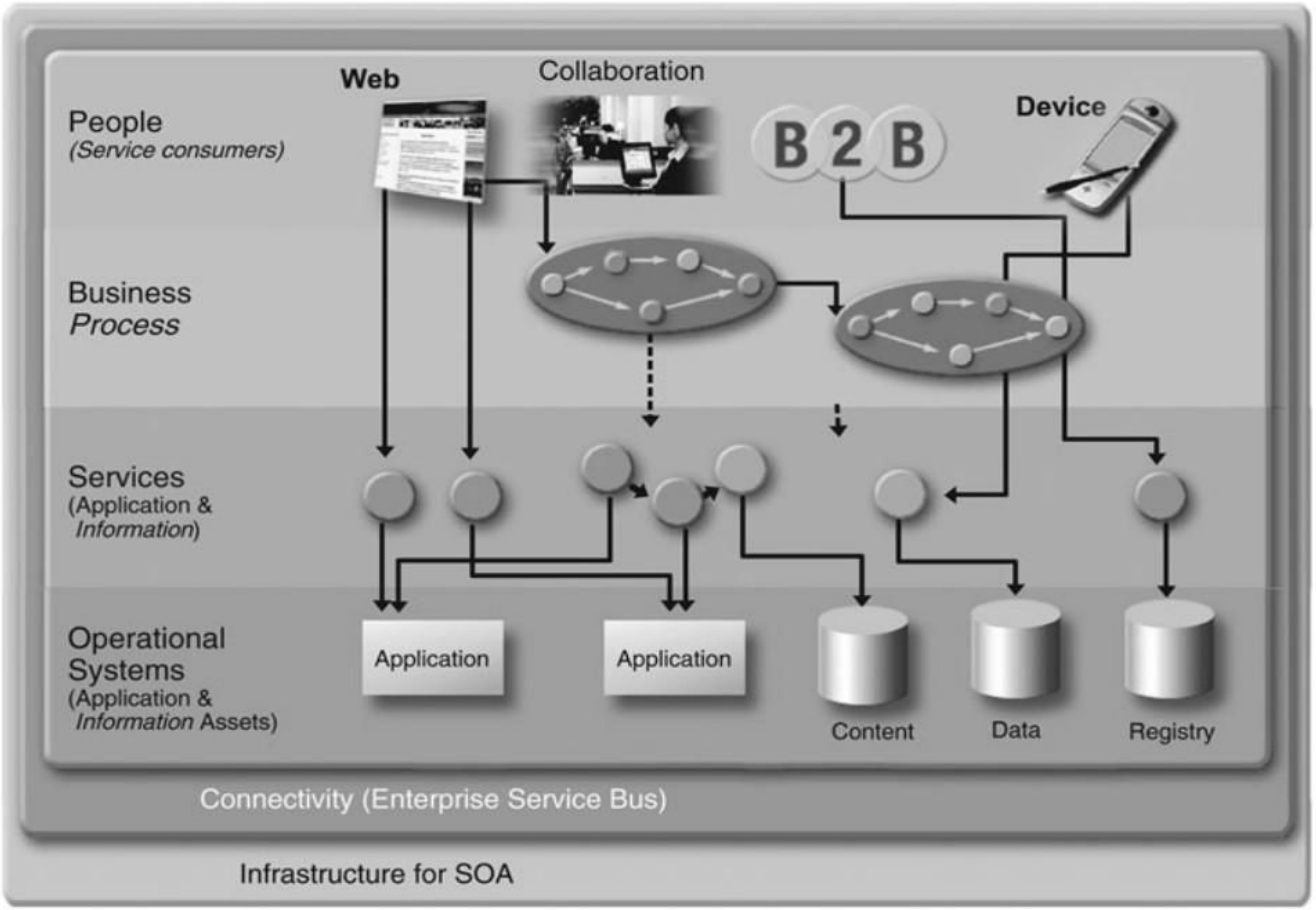
-- The Open Group

How is BPM and SOA related?

- BPM stands between IT and business
  - **BPM** reflects business needs
  - **BPM** orchestrates IT services
  - **SOA** is about these services



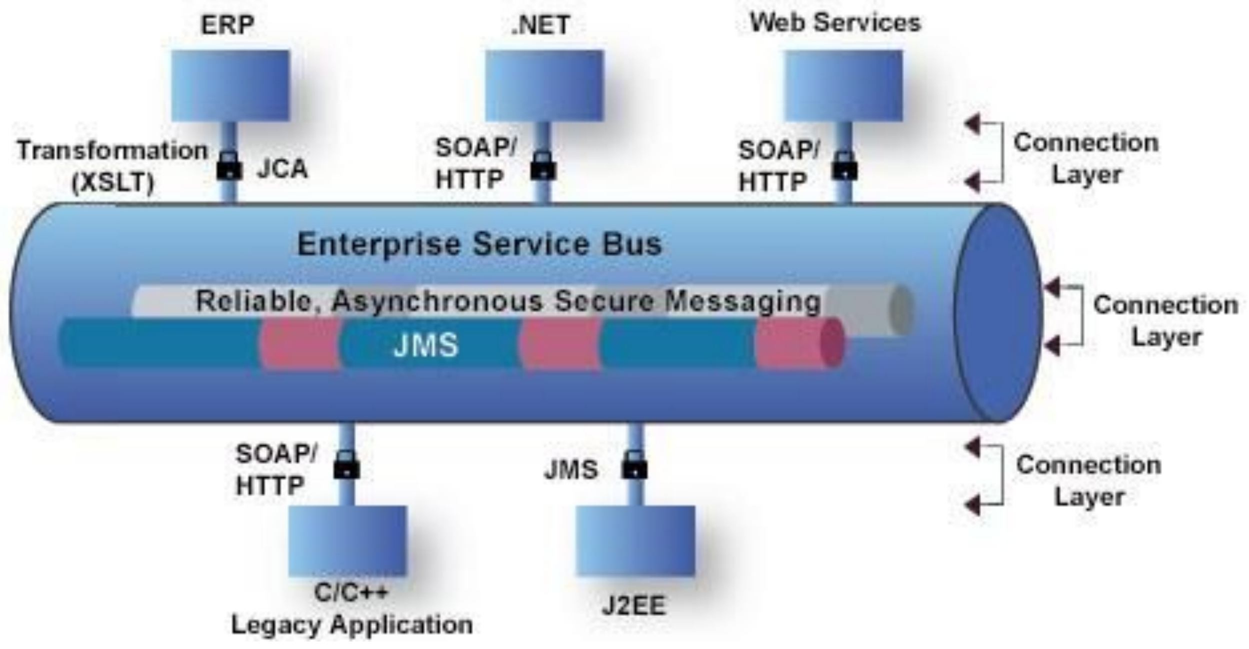
SOA Architecture



Source: SOA Community of Practice, SOA Solution Stack Project

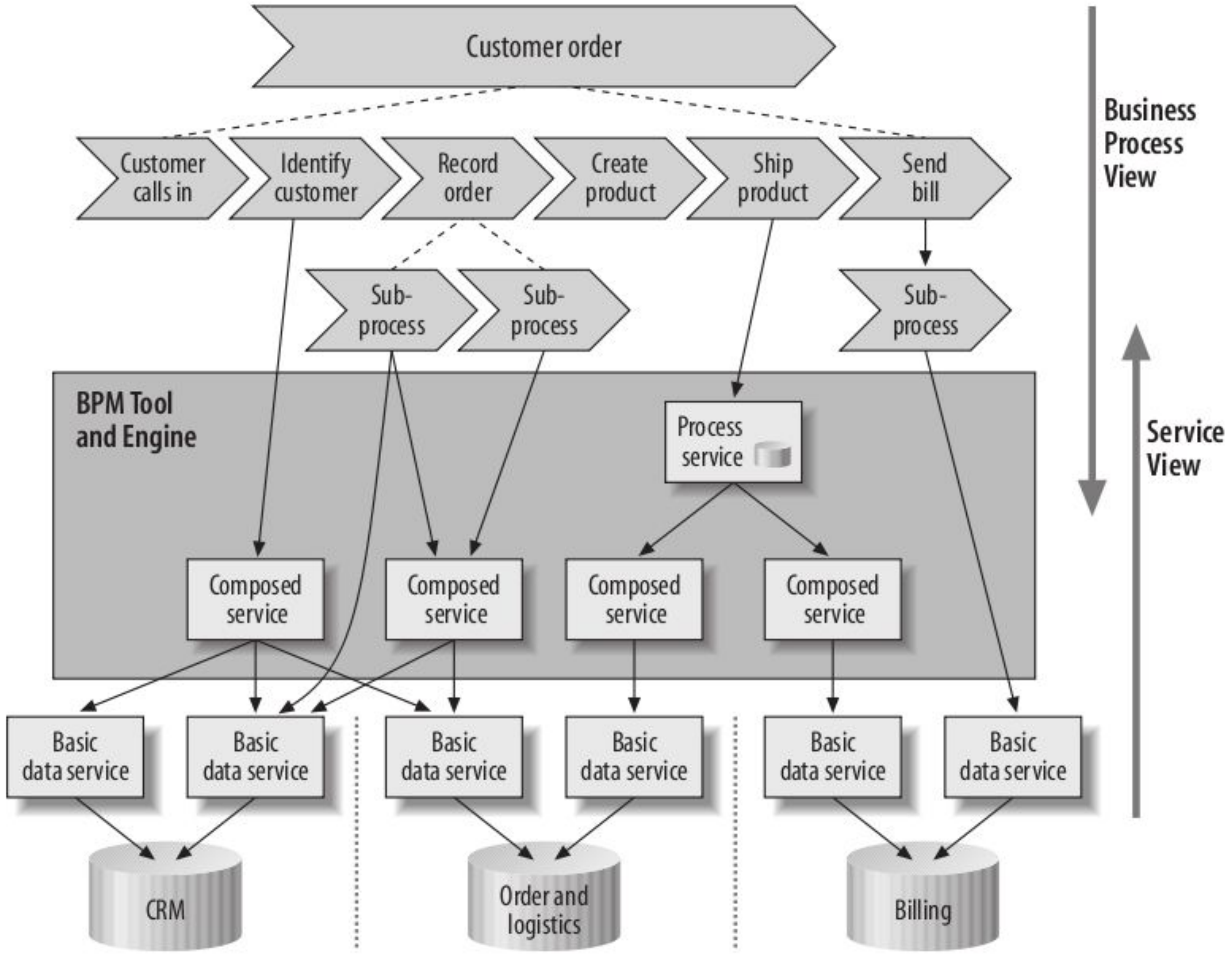
SOA in practice: ESB – Enterprise Service Bus

- Based on message queuing and routing, often combined w Enterprise Integration Patterns (see below)
- Protocol conversion
- Security, reliability, auditability…



http://wso2.com/products/enterprise-service-bus/

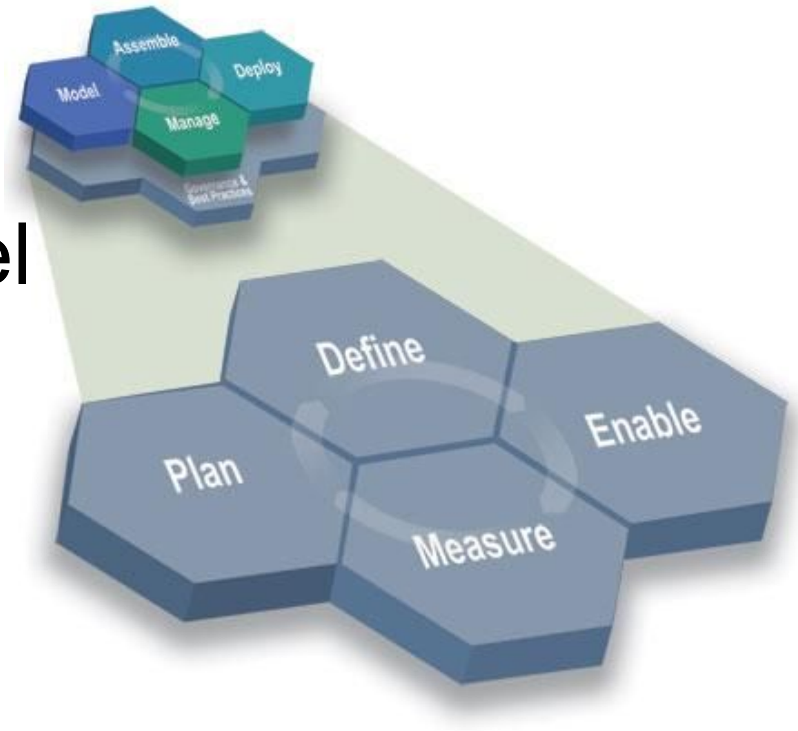
BPM and SOA Relationship



SOA in Practice, , Nicolai M. Josuttis

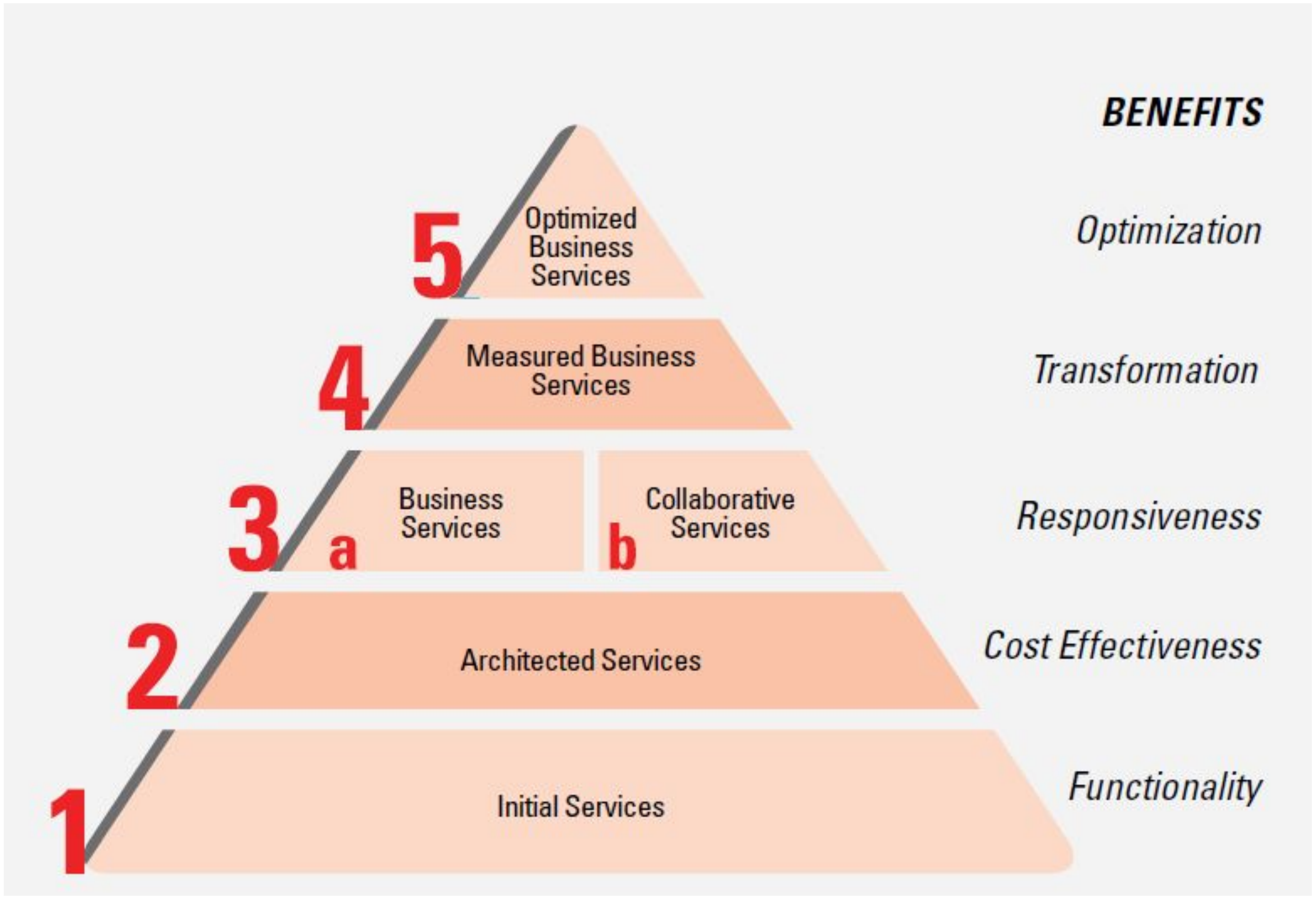
SOA Governance

- Service definition
- Service deployment life cycle
- Service versioning
- Service migration
- Service registries
- Service message model
- Service monitoring
- Service ownership
- Service testing
- Service security



IBM Software Group

SOA – Maturity Model



Sonic Software Corporation, AmberPoint Inc., 2005

SOA – Methodologies

- SOA methodologies
  - IBM SOAD (Proprietary)
  - IBM SOMA (Proprietary)
  - SOA RQ (Proprietary)
  - CBDI-SAE
  - SOAF
- SOMA
  - Service-oriented modeling and architecture

    --Ali Arsanjani, Chief Architect,
    SOA and Web services Center of Excellence,
    IBM, Software Group

EAI Generations – MSA / ROA

MSA – Microservices Architecture

Breaking *monolithic* application structure into set of *discrete* services (IT/web).

- https://martinfowler.com/articles/microservices.html

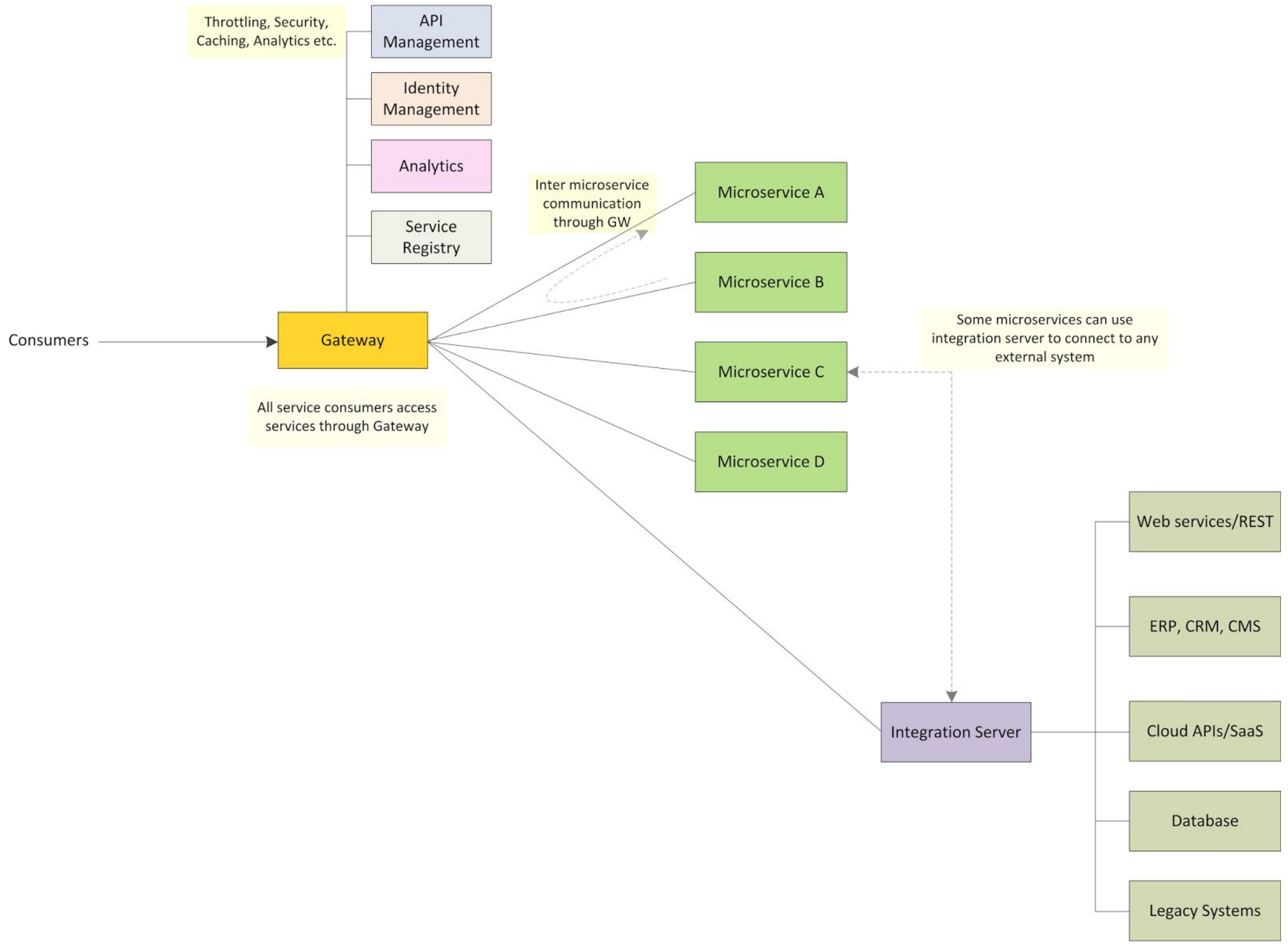
ROA – Resource oriented Architecture

(Inter-)networking application resources accessible through *RESTful* webservices.

- https://www.oreilly.com/library/view/restful-web-services/9780596529260/ch04.html

Driven by **DevOps** methodology and practice.

EAI Generations – MSA / ROA + SOA

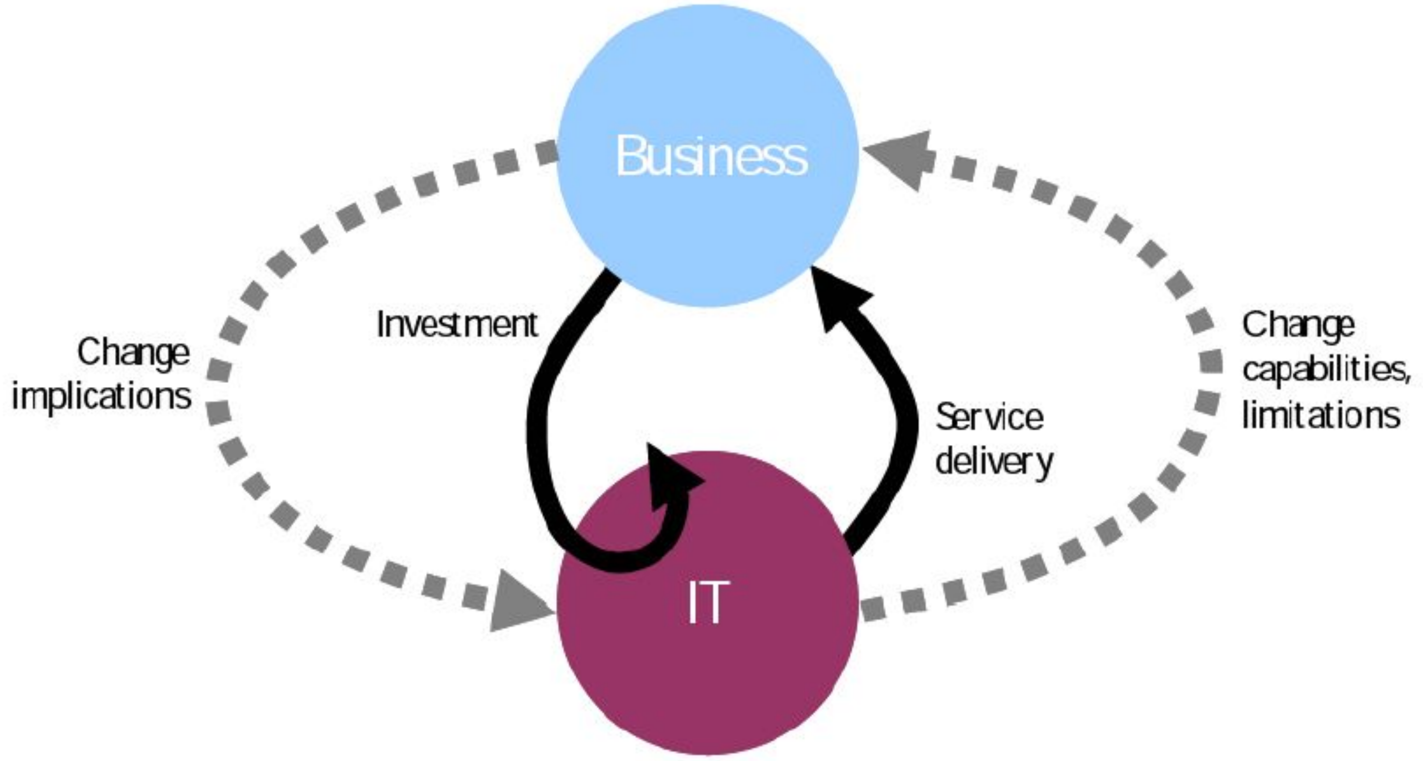


Throttling, Security, Caching, Analytics etc.
API Management
Identity Management
Analytics
Service Registry
Inter microservice communication through GW
Microservice A
Microservice B
Microservice C
Microservice D
Consumers
Gateway
All service consumers access services through Gateway
Some microservices can use integration server to connect to any external system
Web services/REST
ERP, CRM, CMS
Cloud APIs/SaaS
Database
Legacy Systems
Integration Server

https://dzone.com/articles/building-integration-solutions-a-rethink

Business & IT alignment

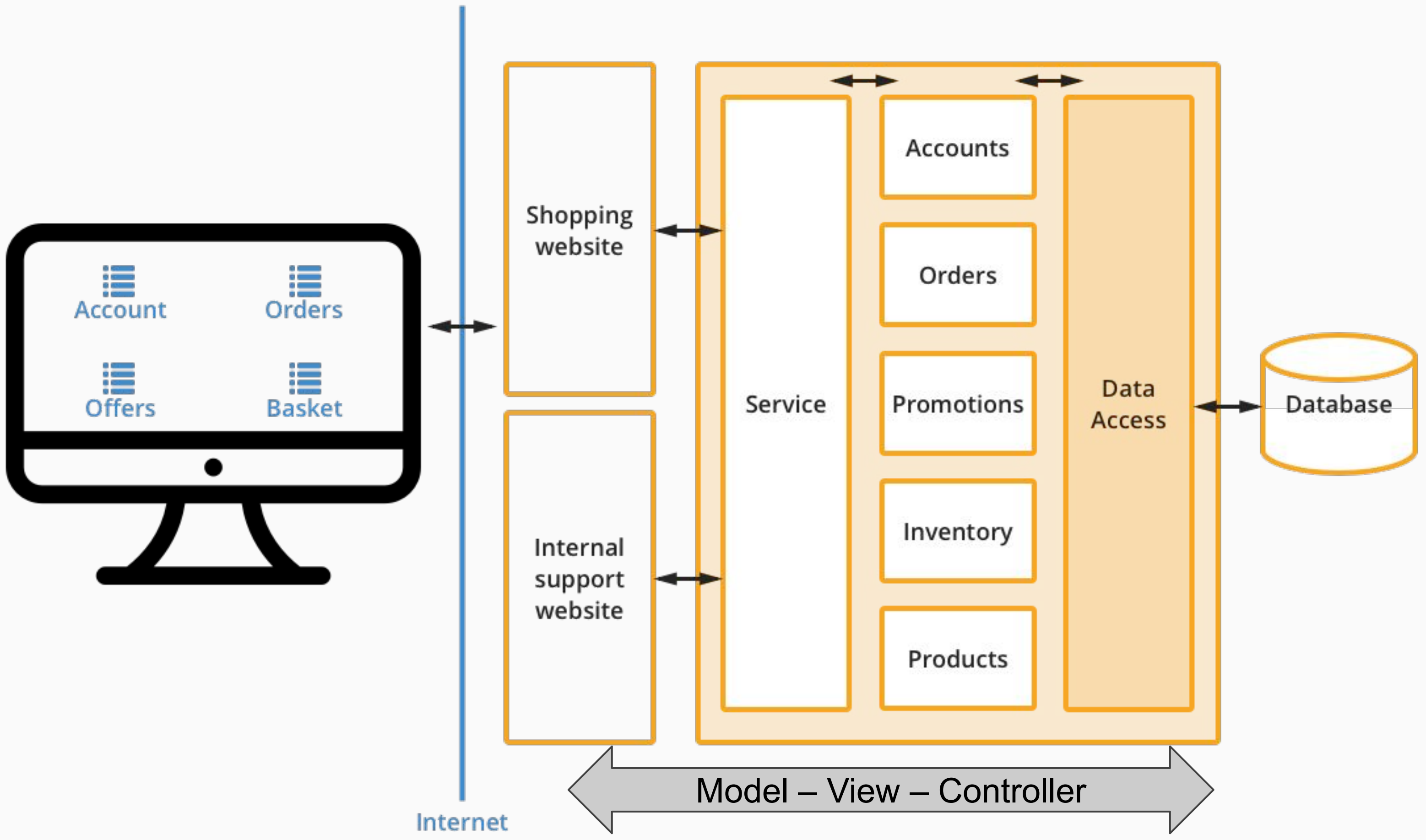
Figure 1: The elements of IT-business alignment



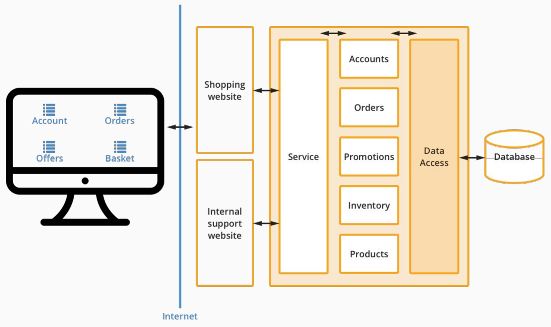
There are three important elements in IT-business alignment: investment, service delivery, and collaboration in change management.

Macehiter Ward-Dutton

Monolith


Account
Orders
Offers
Basket
Internet
Shopping website
Internal support website
Service
Accounts
Orders
Promotions
Inventory
Products
Data Access
Database
Model – View – Controller

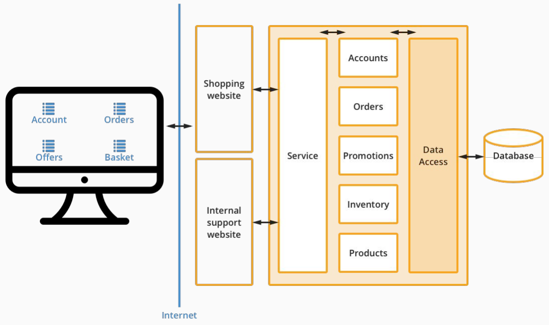
Pros of monolithic architecture



Pros

- natural evolution of system without restriction
- everything is accessible from one place
- does not push to automate infrastructure, deployment and testing

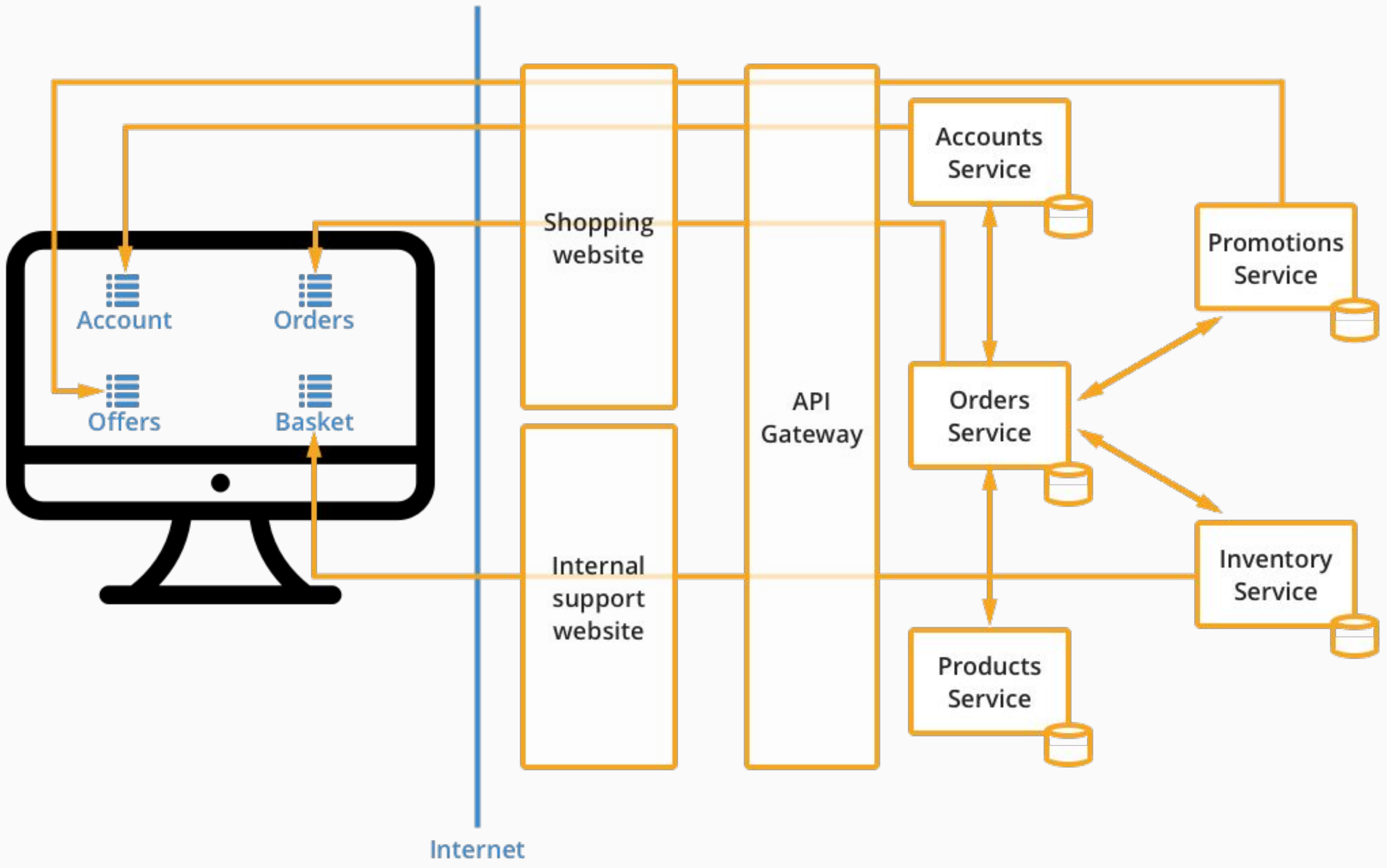
Cons of monolithic architecture



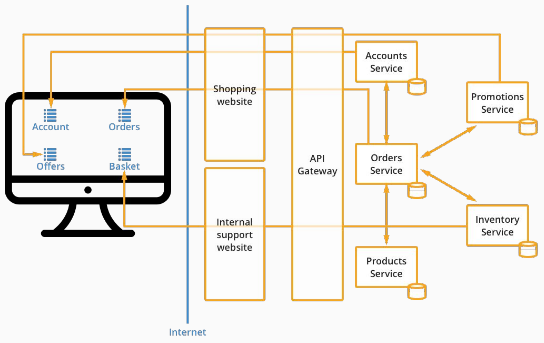
Cons

- large codebase
- deploy takes too long
- one fix means deploy the whole system and test everything
- hard to scale just single part of the system
- one failure usually equals downtime of the whole system
- team has to understand everything

Microservices



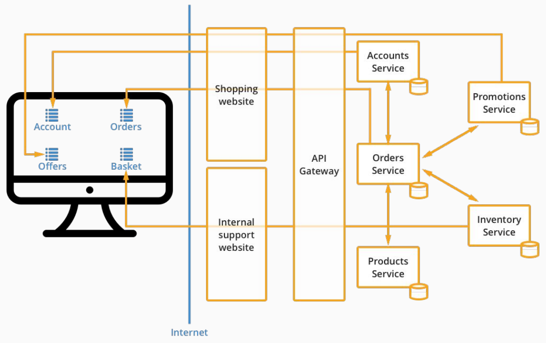
Pros of microservice architecture



Pros

- deploy of single service is easy
- scaling a service is possible
- one team is responsible for single service
- service can be created or changed in short amount of time
- slowdown or downtime of a service does not block the whole system
- services can be aligned to support new business needs in short amount of time
- APIs have to be defined

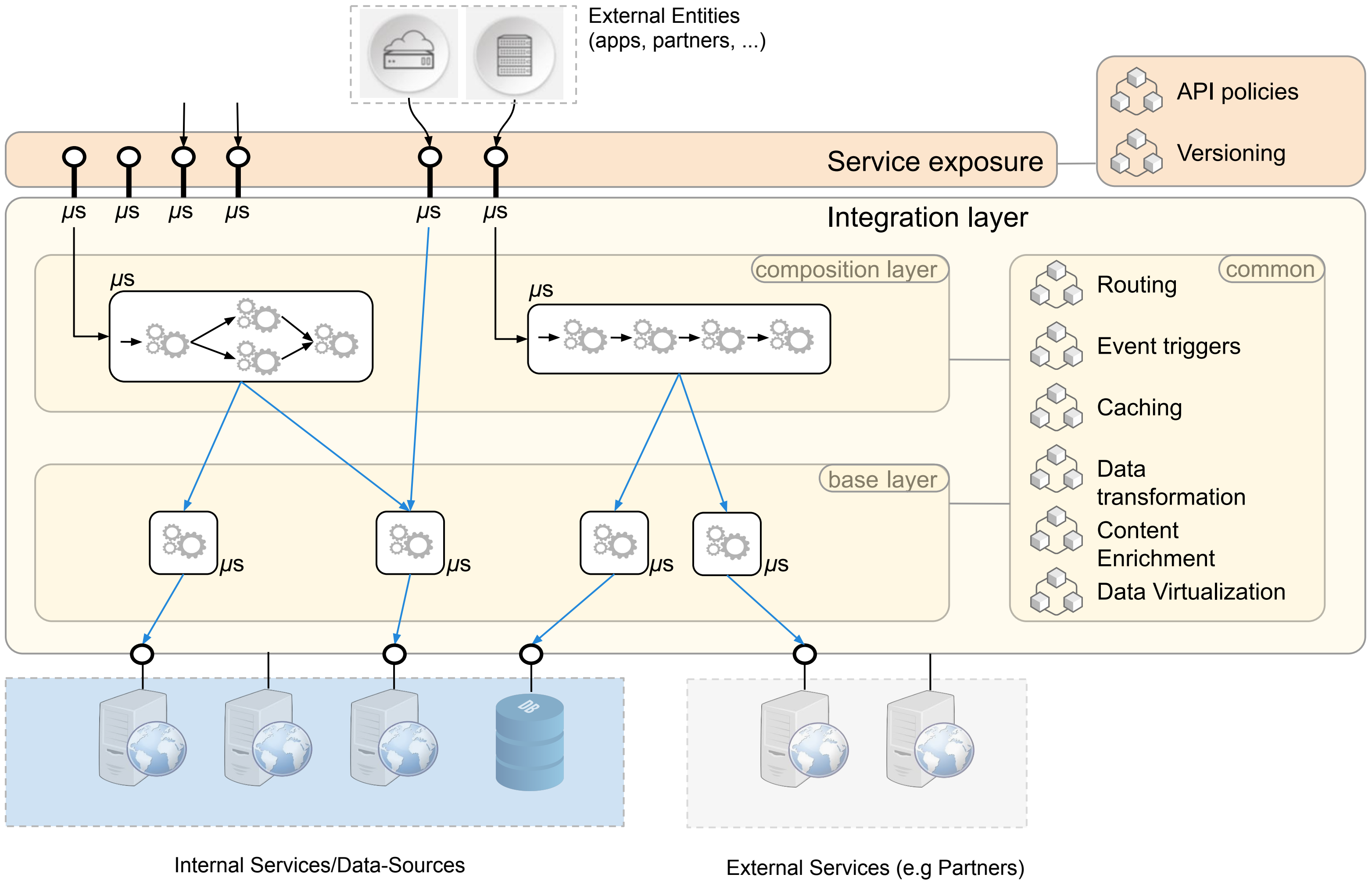
Cons of microservice architecture



Cons

- deployment and versioning is complex
- more automation and DevOps knowledge is needed
- Architecture, technology and performance overhead
- team does not have to know other parts of the whole system, only their services and related APIs
- no one know how the whole system works if business processes are not documented/automated
- tooling for API design and management are often necessary

Microservices in practice



Questions?

Break 10mins

3 meanings of word "service"

- "Business" service
  - Restaurant owner can register his restaurant to Google database and be shown in Google Maps
  - Defined by contract / service offering
- "Technical" service
  - Users can search for their favourite restaurant in Google Maps
  - User interface for "Human task"
- **Web Service**
  - Google provide Web Service API for retrieving location of certain address
  - WSDL interface definition
  - Request - response model

Web Service

- Service for message transport and remote procedure calls
- Messages are transported in XML/JSON
- Transport protocol is HTTP/HTTPS (mostly)
- Web service define:
  - Operations (method) a and their parameters
  - Return types
- SOAP and REST implementations

Modern alternatives: gRPC, GraphQL,..

WSDL

WSDL (Web Service Description Language)

- Describes basic interface of the service
- Methods
- Parameters and their types
- Return values
- Specify **where** is WS available
  - Protocol (HTTP/HTTPS/SMTP)
  - Port (:1666)
  - machine (kore.muni.cz)
  - URL (http://kore.muni.cz:1666/My Service)

WSDL example

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="PrvniSluzba"
 targetNamespace="urn:mojeURI"
 xmlns:tns="urn:mojeURI"

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/
"

xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/
"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:ns1="urn:mojeURI"
 xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/"
 xmlns="http://schemas.xmlsoap.org/wsdl/">

<!-- definice typů -->
<types>
 <schema targetNamespace="urn:mojeURI"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns="http://www.w3.org/2001/XMLSchema"
     elementFormDefault="unqualified"
     attributeFormDefault="unqualified">
  <element name="cislo" type="xsd:long"/>
  <element name="vysledek" type="xsd:boolean"/>
 </schema>
</types>

<!-- komunikační zprávy -->
<message name="jePrvocisloRequest">
 <part name="cislo" element="ns1:cislo"/>
</message>
<message name="jePrvocisloResponse">
 <part name="vysledek" element="ns1:vysledek"/>
</message>

<!-- dostupné operace -->
<portType name="Cisilka">
 <operation name="jePrvocislo">
   <documentation>Operace jePrvocislo()</documentation>
   <input message="tns:jePrvocisloRequest"/>
   <output message="tns:jePrvocisloResponse"/>
 </operation>
</portType>

<!-- volatelné přes HTTP -->
<binding name="PrvniSluzba" type="tns:Cisilka">
 <SOAP:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
 <operation name="jePrvocislo">
  <SOAP:operation style="rpc" soapAction=""/>
  <input>
   <SOAP:body use="literal" namespace="urn:mojeURI"/>
  </input>
  <output>
   <SOAP:body use="literal" namespace="urn:mojeURI"/>
  </output>
 </operation>
</binding>

<!-- adresy komunikačních bodů -->
<service name="PrvniSluzba">
 <documentation>Sluzba pocitajici
prvocisla</documentation>
   <port name="PrvniSluzba" binding="tns:PrvniSluzba">
    <SOAP:address location="http://localhost:10000"/>
   </port>
</service>
</definitions>
```

SOAP

- Protocol for **transfer of XML messages**
- Used for **communication between service and its consumer** (client)
- Common use of HTTP/HTTPS as a transport protocol
- Request – Response communication model

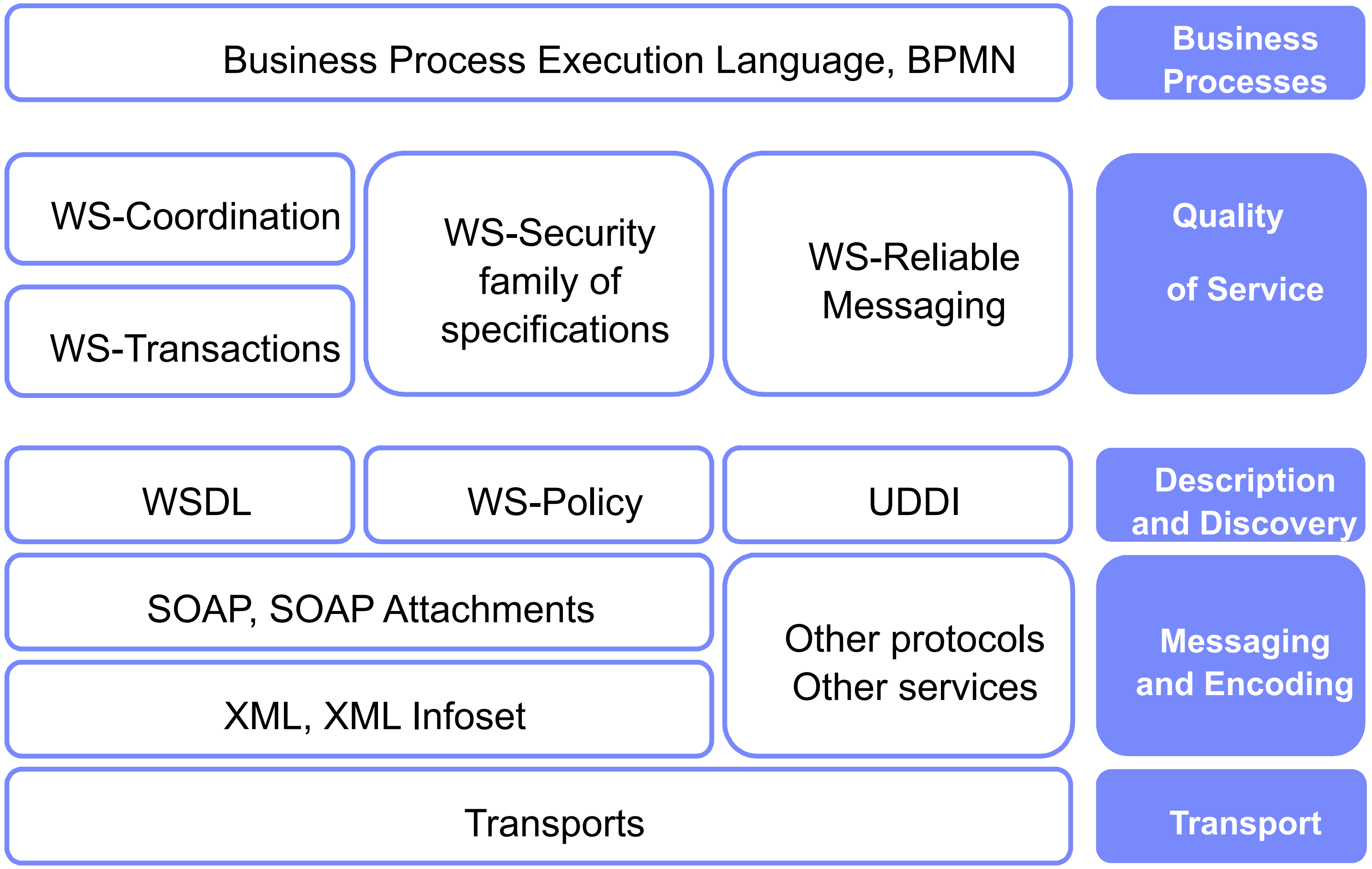
Example – Google Single Sign-on/Identity provider implementation of **SAML**: Security Assertion Markup Language

SOAP example

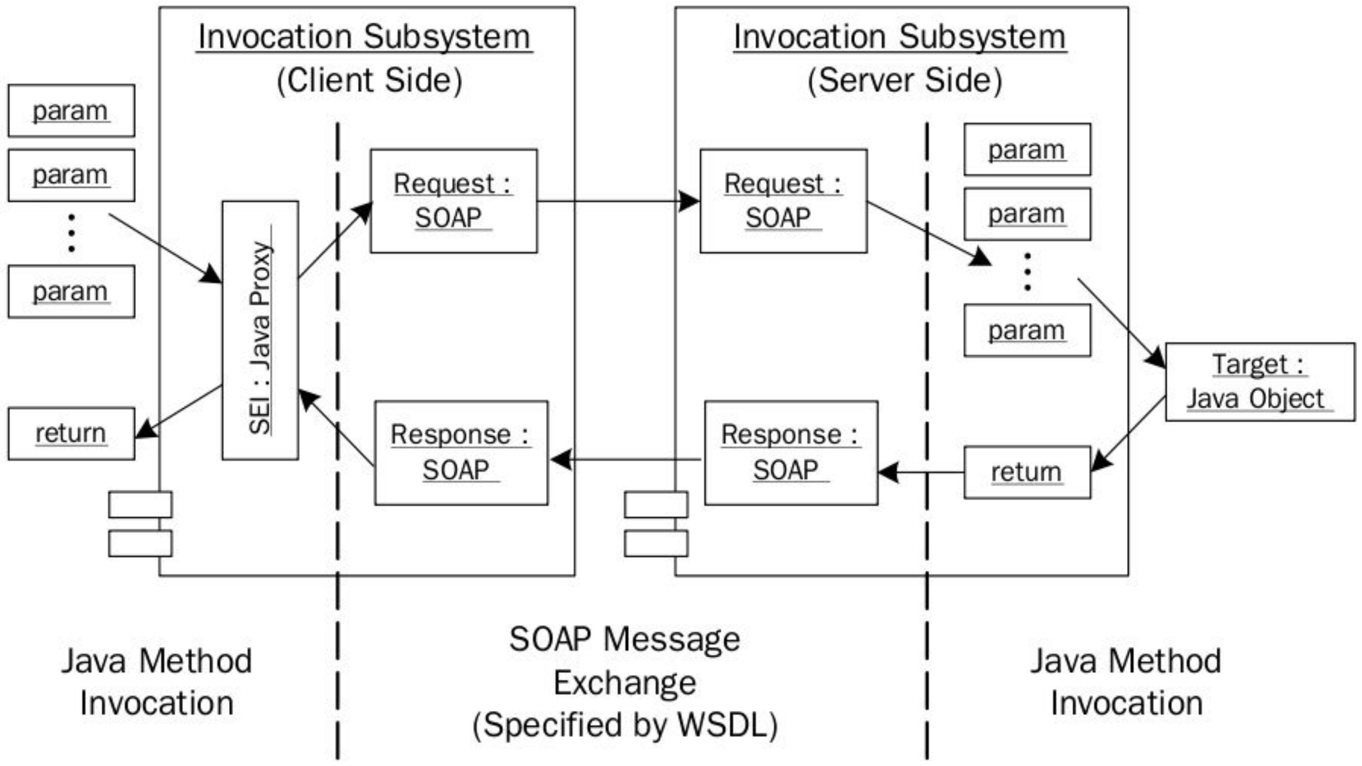
```
POST / HTTP/1.1
Content-Type: text/xml; charset=utf-8
Content-Length: 423
Connection: close
SOAPAction: ""

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope

xmlns:env="http://schemas.xmlsoap.org/soap/env
elope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi=""

 <env:Header/>
 <env:Body>
  <jePrvocislo xmlns="urn:mojeURI">
    <cislo xsi:type="xsd:long">1987</cislo>
  </jePrvocislo>
 </env:Body>
</env:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 468
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope

xmlns:env="http://schemas.xmlsoap.org/soap/envel
ope/"  xmlns:xsi=""
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <env:Body>
  <jePrvocisloResponse xmlns="urn:mojeURI">
   <vysledek
xsi:type="xsd:boolean">true</vysledek>
  </jePrvocisloResponse>
 </env:Body>
</env:Envelope>
```

WS - Standards



Web Services in Java



RESTful Web Service

Representational State Transfer

- Uniform resource interface (a set of constraints)
- Client-server separation
- Stateless
- Cacheable resources
- Layered system
- Code on demand (optional – JavaScript)

https://raygun.com/blog/soap-vs-rest-vs-json/

RESTful Web Service

RESTful Web Services characteristics:

- HTTP/HTTPS protocols:
  - Using URI for resource identification
  - Only POST, GET, PUT & DELETE (CRUD)verbs (or only others from HTTP specification)
  - HTTP Status Codes as flow control
  - XML, JSON, YAML text formats as resources representations
- OpenAPI (swagger), RAML, API Blueprint, WADL, HAL specification formats and tools

REST example

```
curl -i -X POST
"http://localhost:3000/api/personal/users/4c617f2b-2bad-498b-a9c6-4e9a8c303798/bookmarks"
-H "accept: */*" -H "Authorization: Bearer eyJhbGciOiJ...."
-H "Content-Type: application/json" -d
"{\"name\":......,\"public\":true,\"lastAccessedAt\":\"2020-03-06T20:14:28.101Z\",\"likeCount\":0}"


HTTP/1.1 201 Created
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Headers: Content-Type, Authorization, Location
Access-Control-Expose-Headers: Content-Type, Authorization, Location
Location:
http://localhost:3000/api/personal/users/4c617f2b-2bad-498b-a9c6-4e9a8c303798/bookmarks/5e62b18b59770b5
487a4c741
Content-Type: application/json; charset=utf-8
Content-Length: 79
ETag: W/"4f-26GcBfsvgN8d+T+zqql3Y5R+Rl8"
Date: Fri, 06 Mar 2020 20:24:44 GMT
Connection: keep-alive

{"response":"Bookmark created for userId 4c617f2b-2bad-498b-a9c6-4e9a8c303798"}
```

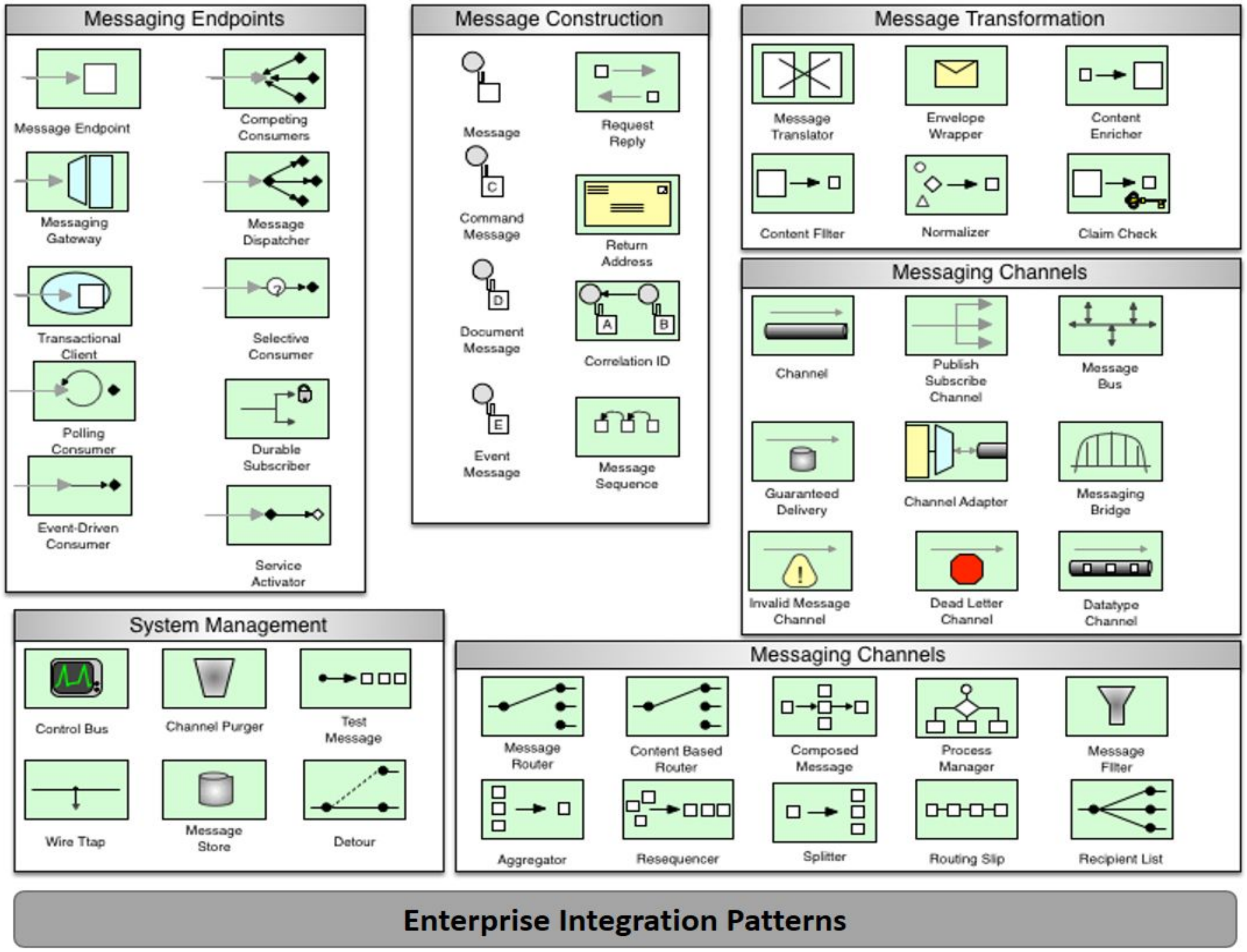
WS Standards

- JAX-WS (JSR-224)
- JAX-RS (JSR-311)

- Apache Axis, Axis2
- Apache CXF
- Jersey

- Spring Boot MSA framework
- Resteasy framework

Other options
EIP as microservices

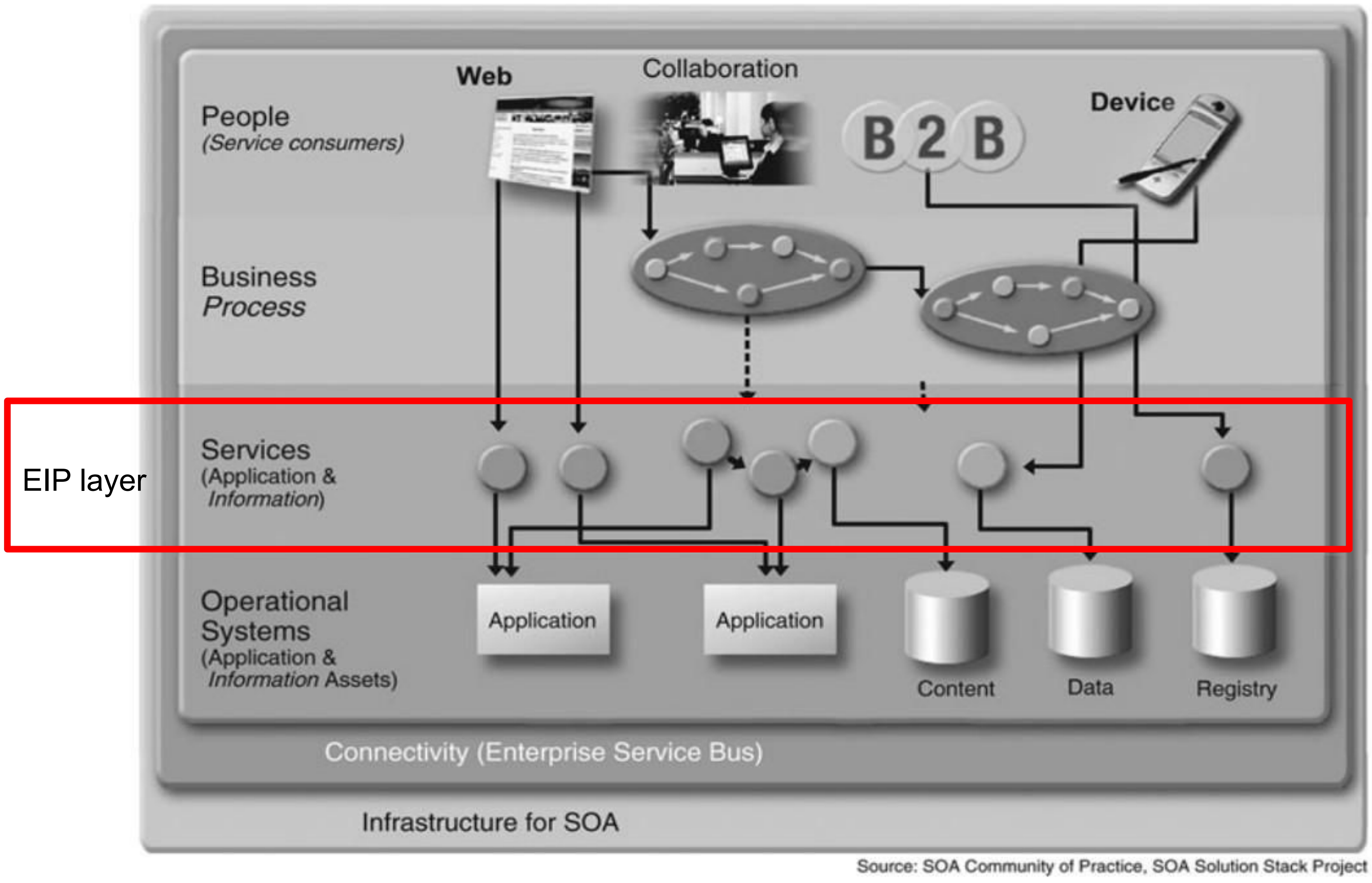


Enterprise Integration Patterns

- Based on traditional concept of asynchronous *Message Queuing and Routing*
- Application data is sent as messages that can be acted upon being sent through route, i.e. data can be aggregated, enriched, translated…
- Popular OSS software Apache Camel (Java)

- https://www.enterpriseintegrationpatterns.com/patterns/messaging/index.html

Enterprise Integration Patterns



Project service implementation tips

Mock your webservice

- Use existing public APIs:
  - SOAP https://documenter.getpostman.com/view/8854915/Szf26WHn?version=latest
  - REST https://documenter.getpostman.com/view/8854915/Szf7znEe

- Use mocking framework
  - I.e. Mockito for Java https://site.mockito.org
  - Online service i.e. https://www.mockable.io

- Use Apache Camel
  - Will be introduced and practiced in later lectures (Ivo Bek)

Implement your web service

Spring Boot REST Example

```
@RestController
public class HelloBootController {


   @GetMapping("/whereami")
   public String whereami(@Value("${message.prefix}") String prefix) {
       String resp = String.format("%s from %s", prefix,
         System.getenv().getOrDefault("HOSTNAME", "localhost"));
       return resp;
   }


}
```

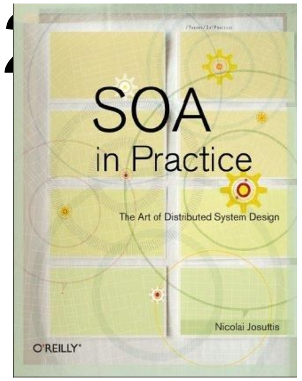
- There are many frameworks available for popular modern development platforms….
- If you implement your own microservices (not-trivial with some db backend), you can receive some bonus help for final evaluation :-)

Web Service tutorials

- Web Services
  - http://netbeans.org/kb/docs/websvc/jax-ws.html
- REST
  - http://netbeans.org/kb/docs/websvc/rest.html
- NetBeans Trail
  - http://netbeans.org/kb/trails/web.html

SOA - Information Resources

- SOA in Practice, Nicolai M. Josuttis,
ISBN-13: 978-0596529550



- IBM Systems Journal, Volume 47, Number 3, 2008

Thanks!
Questions?

PV207 – Business Process Management

Spring 2023