Service-Oriented Architecture

Didier Donsez
Université Grenoble Alpes (UGA)
PolyTech Grenoble - LIG/ERODS
didier.donsez@univ-grenoble-alpes.fr
didier.donsez@ieee.org

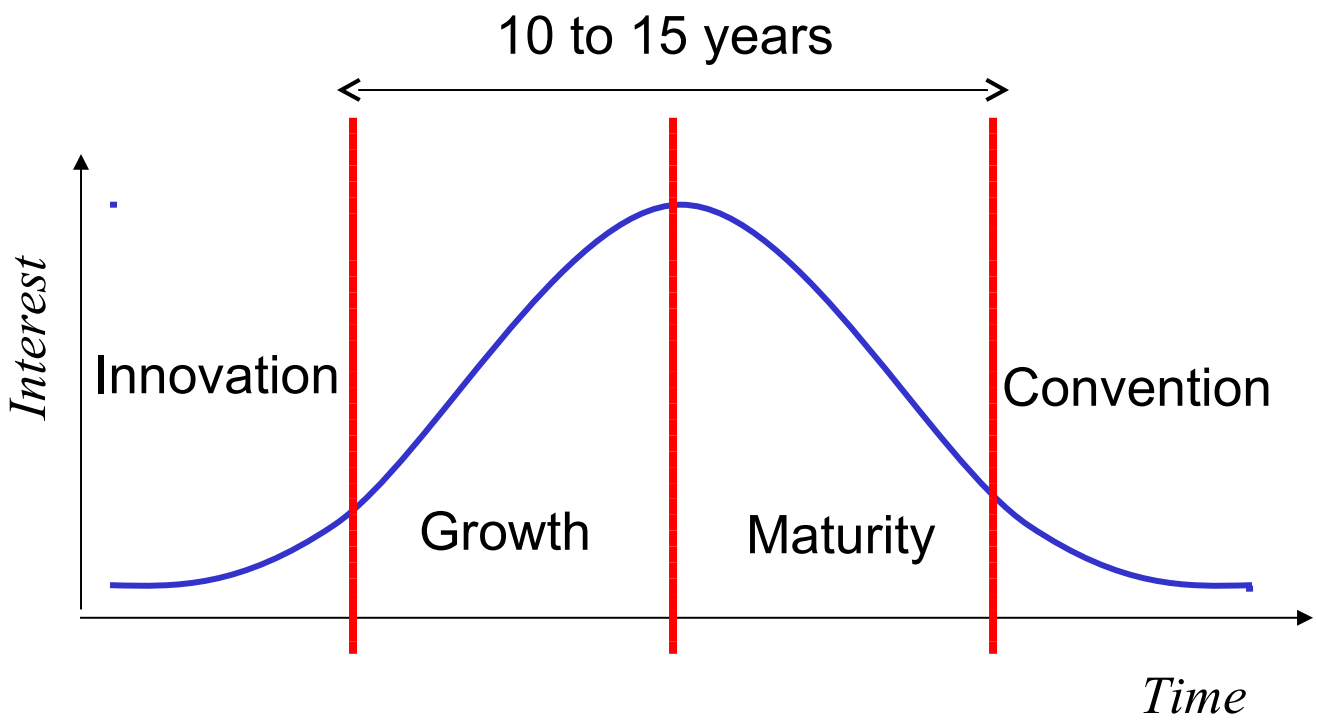
Outline

- Reminder
- Contracts

- SOA Entities
- Component versus Service
- SOA Administration Domains
- SOA Lifecycles
- SOA Frameworks

Gentle Reminder

- Racoon [1997]



Gentle Reminder

- Racoon [1997]

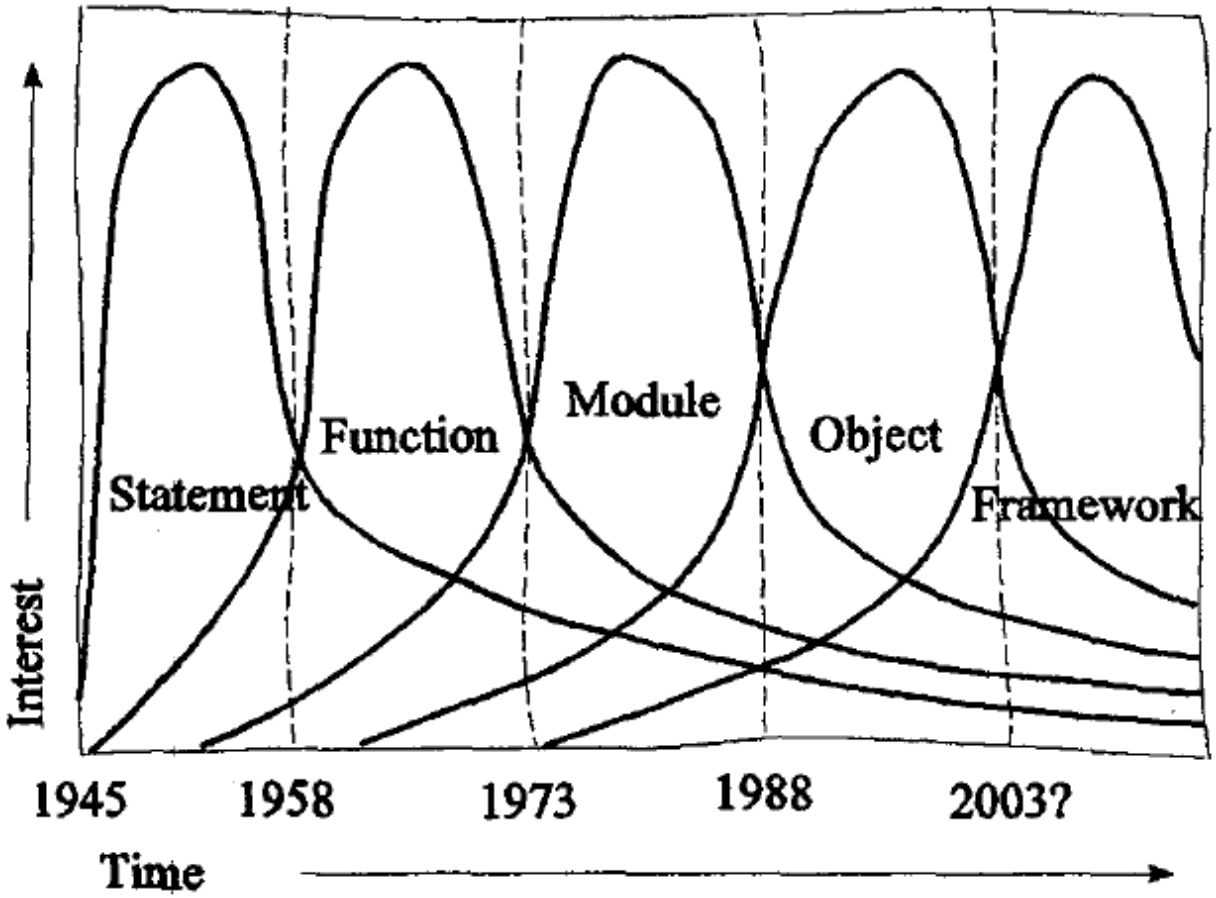
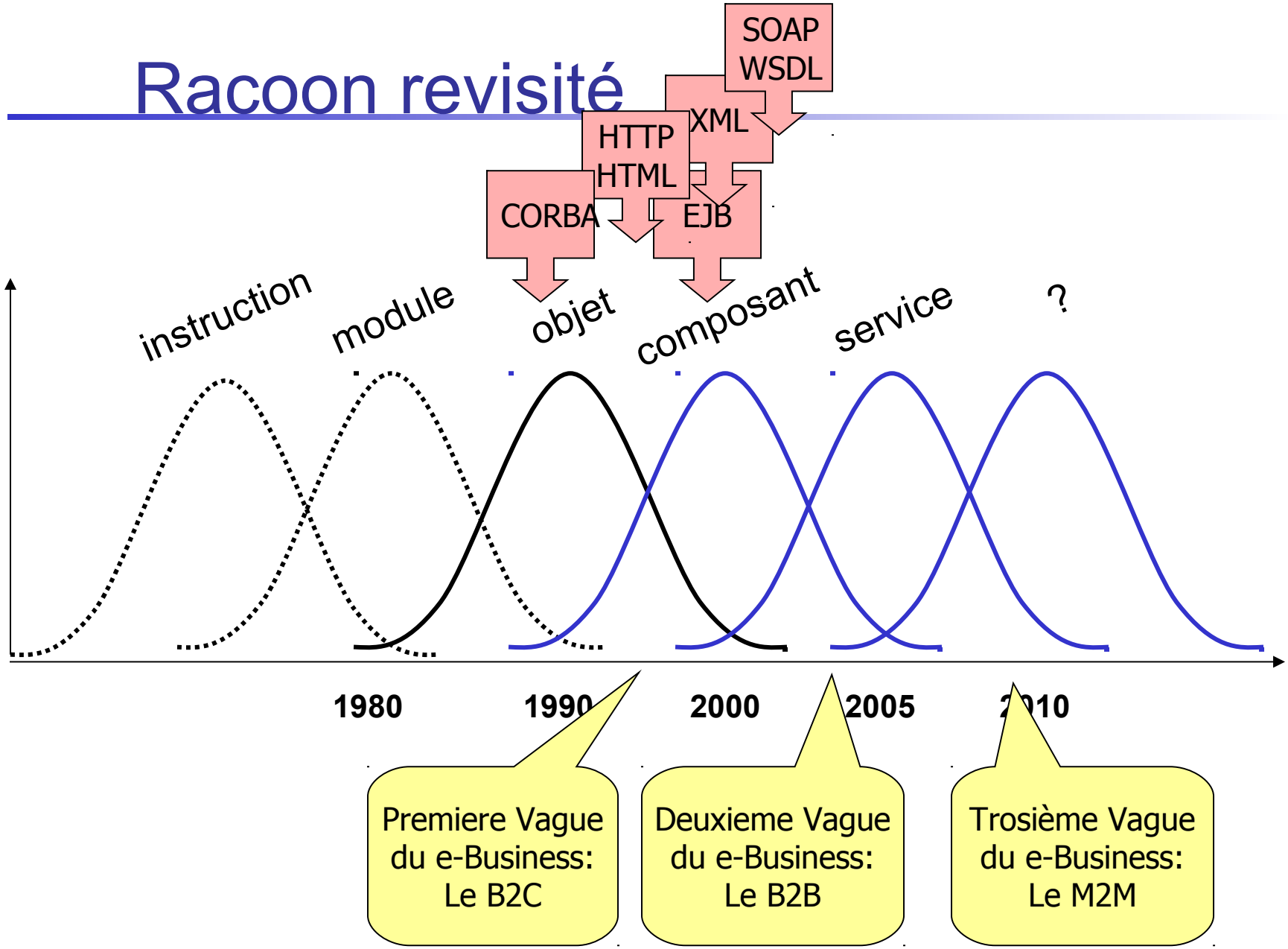
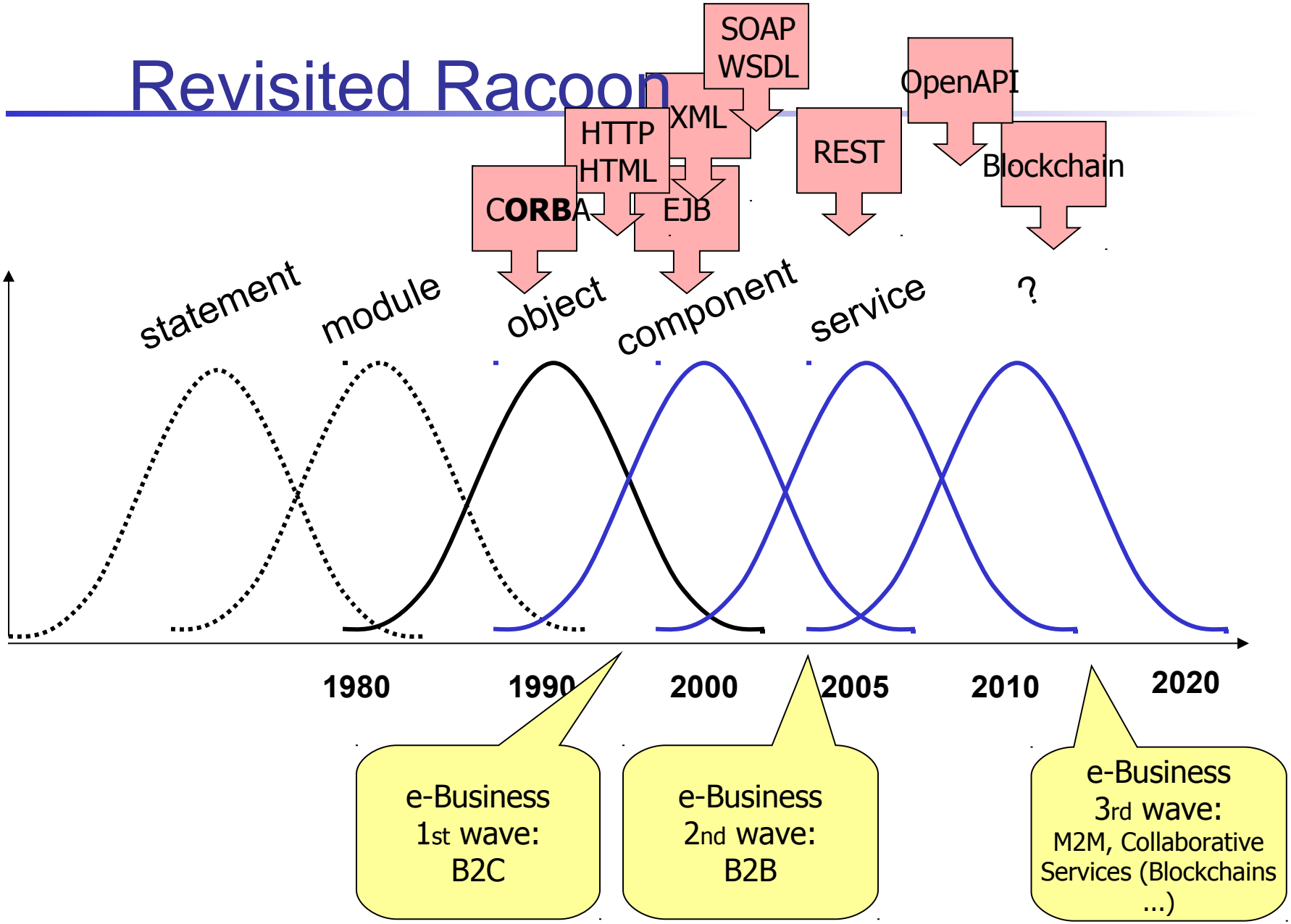


Figure 2: The Organization Stream.

Racoon revisité



Revisited Racoon

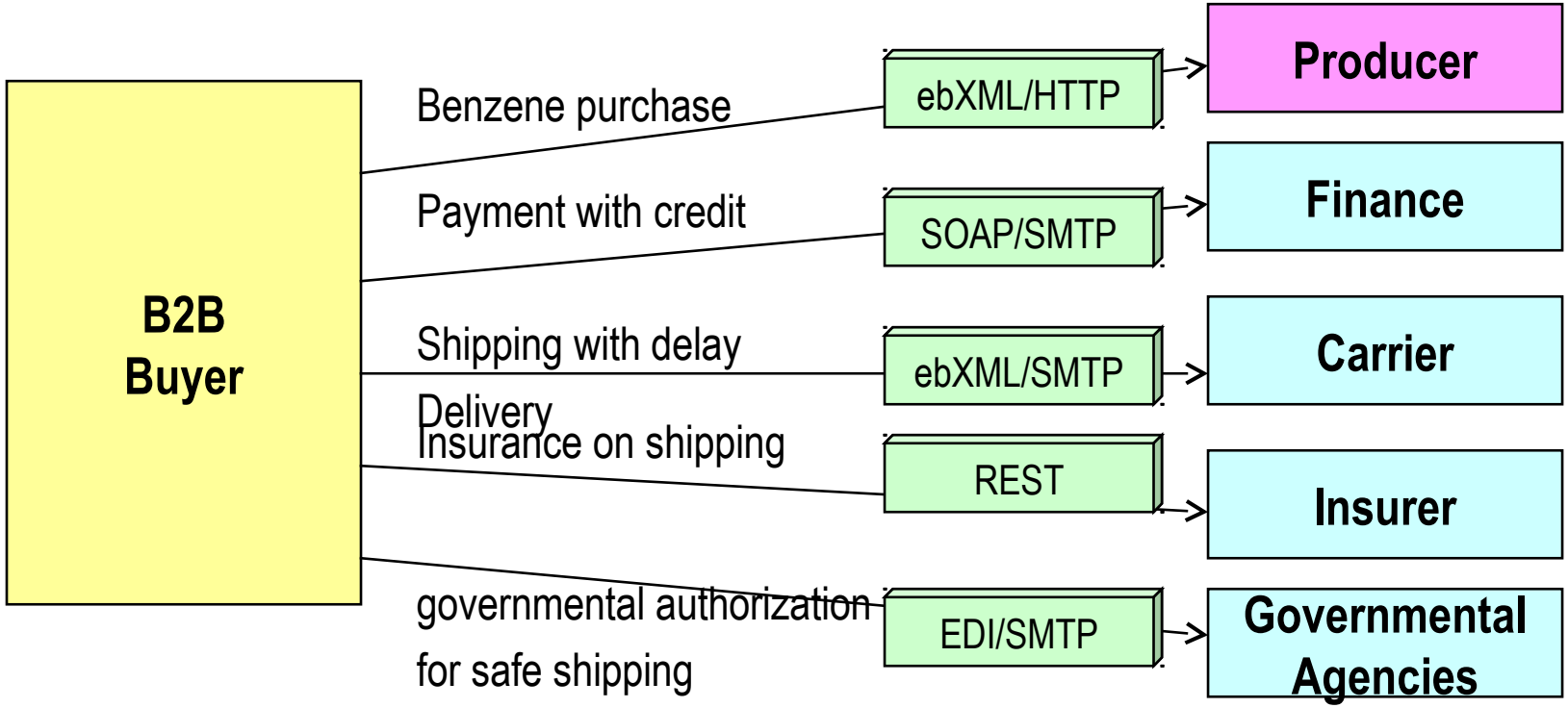


1st Wave of e-Business
B2C *Business to Consumer*

- Web 1.0
  - Google
  - YAHOO!
  - ebay
  - …
  - amazon.com
- Web 2.0
  - PayPal
  - iTune
  - Google/Yahoo Maps
  - Google AdSense
  - YouTube
  - Twitter
  - Facebook *(le 3ème pays du Monde)*
  - …
- Web 3.0
  - Virtual worlds SECOND LIFE WORLD OF WARCRAFT
  - Internet of (Every)Things
  - …

2nd Wave of e-Business
Example of B2B – B2Gov Usecase

- Benzene purchase by a producer on the Web
- + Requires additional services provided by third parties



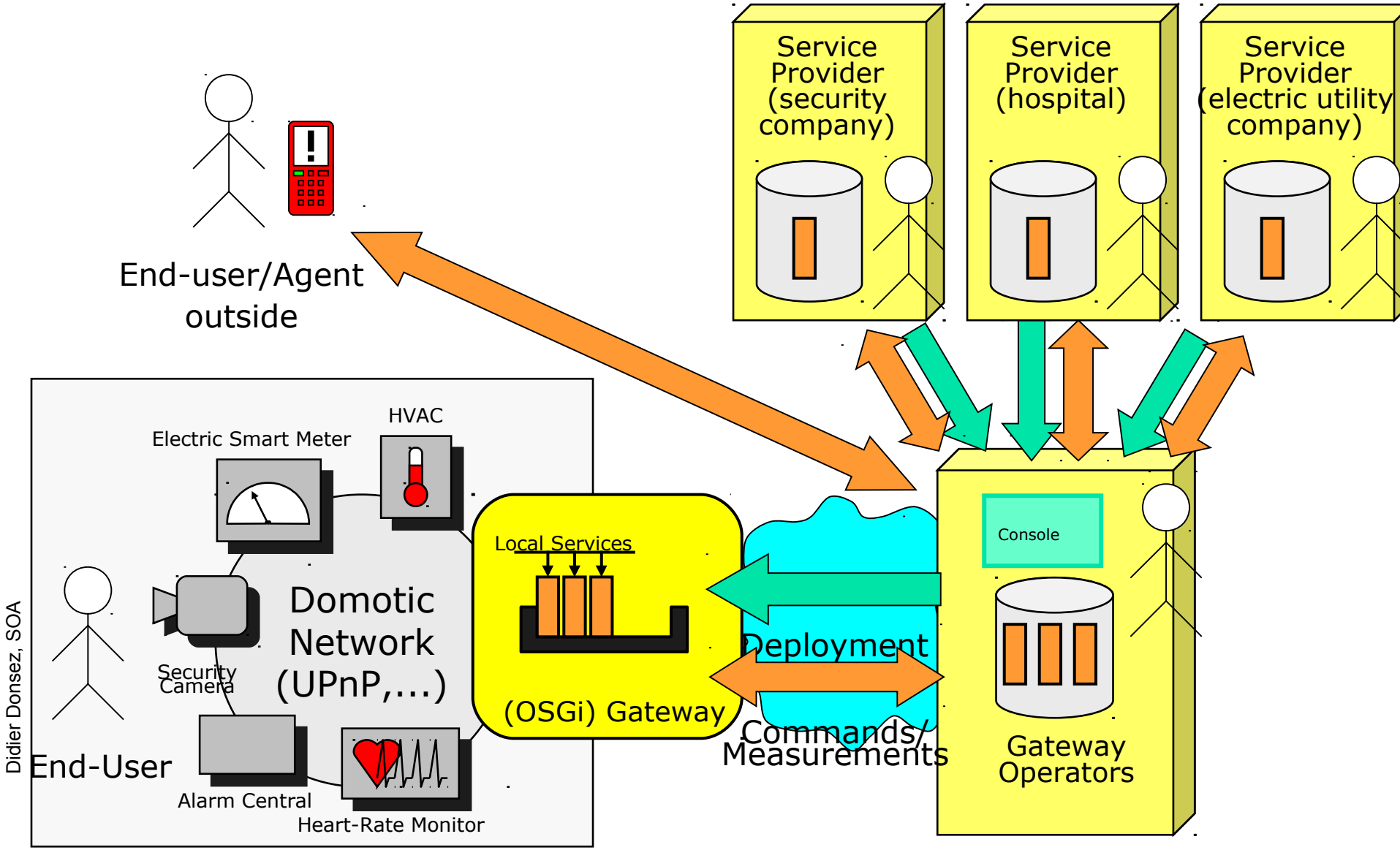
2.5 Wave of e-Business Cloud Computing

- On-Demand Computing
  - Synonyms : Edge computing, Utility computing, Elastic computing, …
- Trendy Acronyms XaaS
  - SaaS : **S**oftware **a**s **a**s **S**ervice
  - PaaS : **P**latform **a**s **a**s **S**ervice
  - IaaS : **I**nfrastructure **a**s **a**s **S**ervice
  - FaaS : Facilities **a**s **a**s **S**ervice
- World Players
  - Akamai, Amazon, Google, Azure, Alibaba, Saleforce …
- Requirements
  - Virtualization, Autonomic Computing, Green computing, …

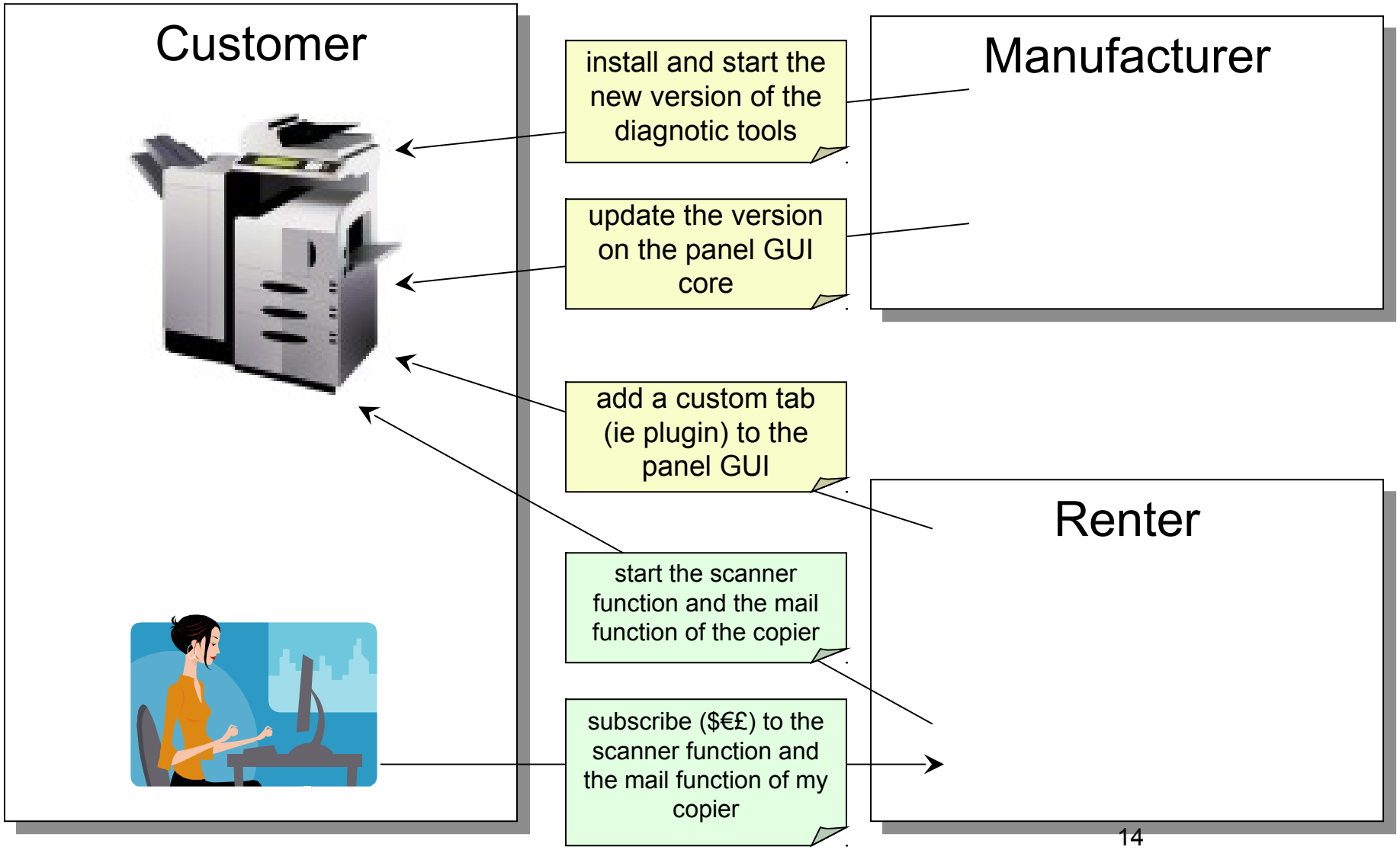
3rd Waves of e-Business
Internet(s) of Data, Things and Services

- Internet of (Chatty) Things
- (Internet of) Connected Objects
- Internet of Everything
- Internet of People
- Internet of My Things
- Industrial Internet of Things (IIoT) : *Industry 4.0*
- Fog Computing
- Cyber-Physical Systems
- ...

3rd Wave of e-Business
Internet of (Every)Things (M2M, IoT, IoE, ...)

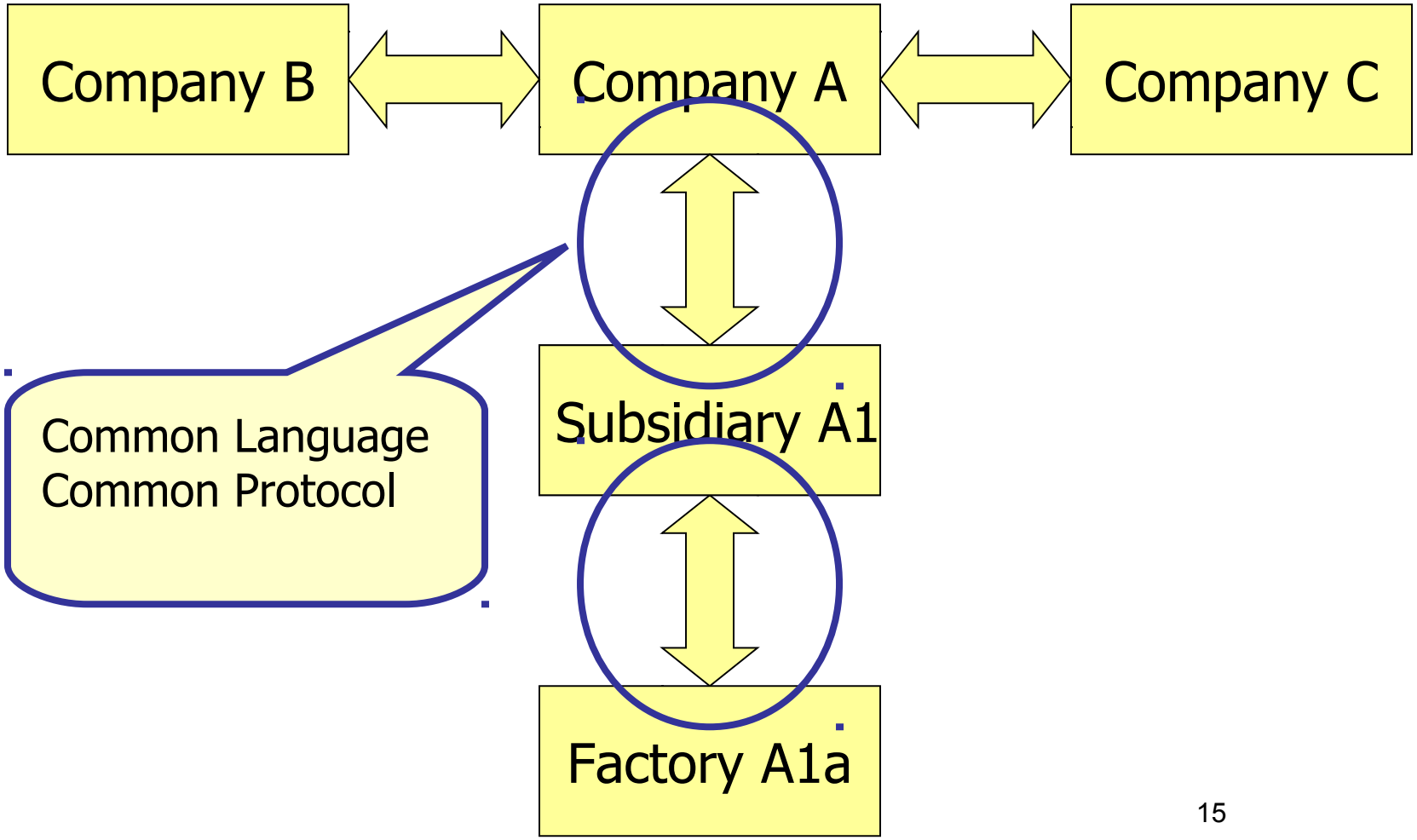


3rd Wave of e-Business IoT
The copier rental



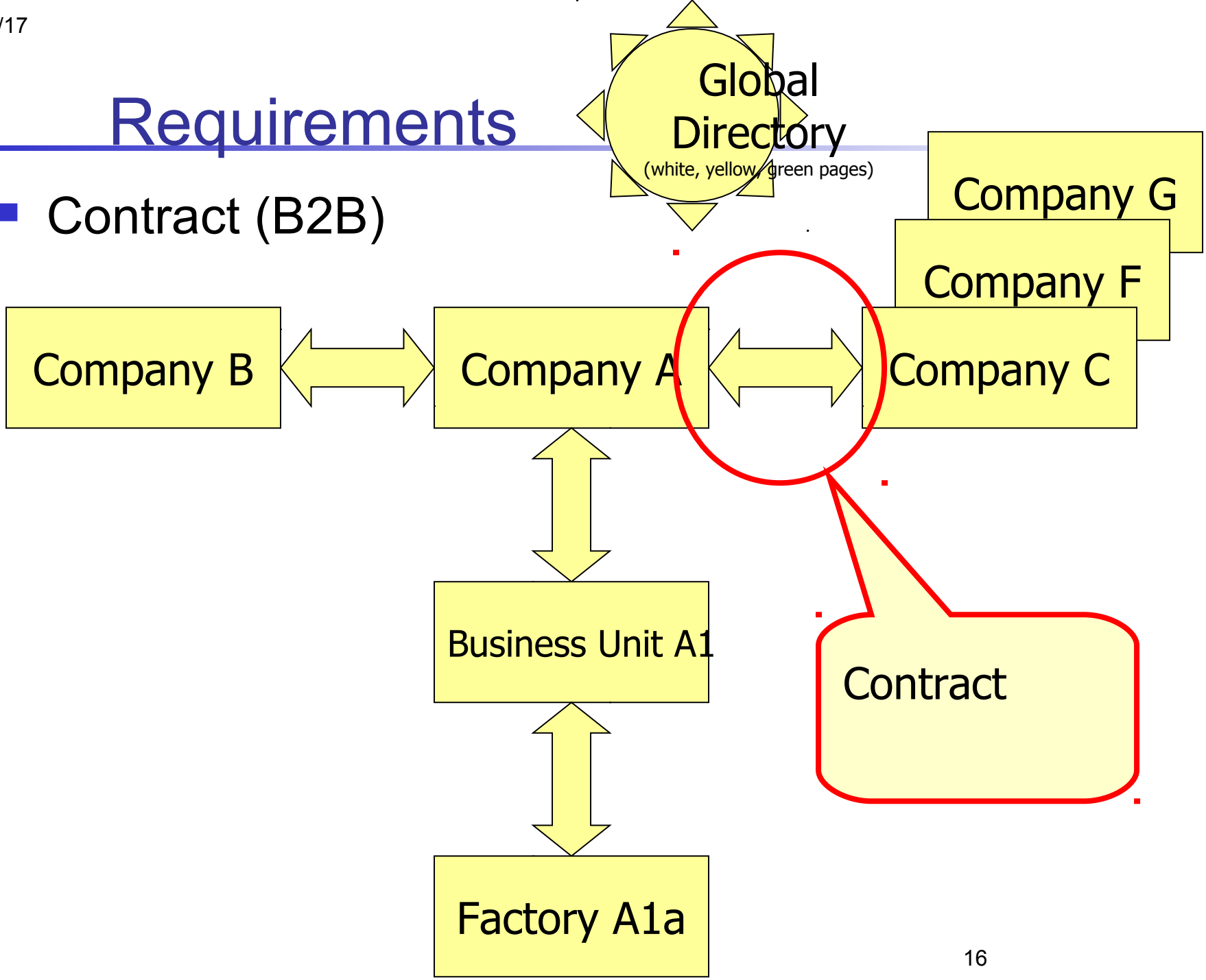
Requirements

- Entreprise Application Integration (EAI)



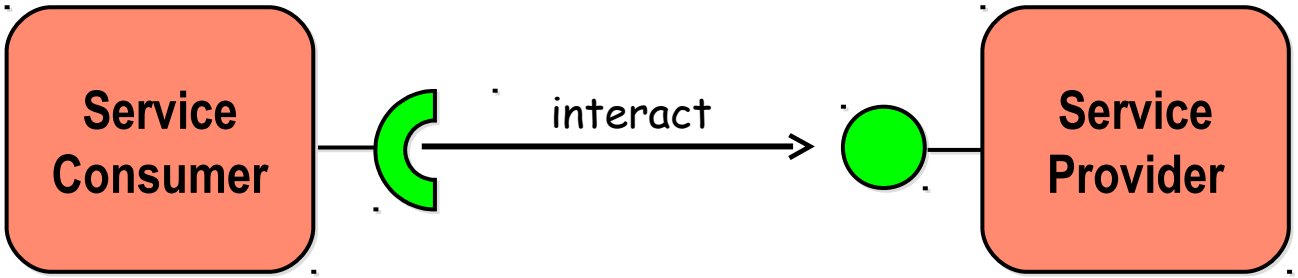
Requirements

- Contract (B2B)



Service Definition

- « a service is a contractually defined behavior that can be **implemented** and **provided** by **any** component for **use by any component**, based solely on the **contract**. »
[Bieber and Carpenter 2002].

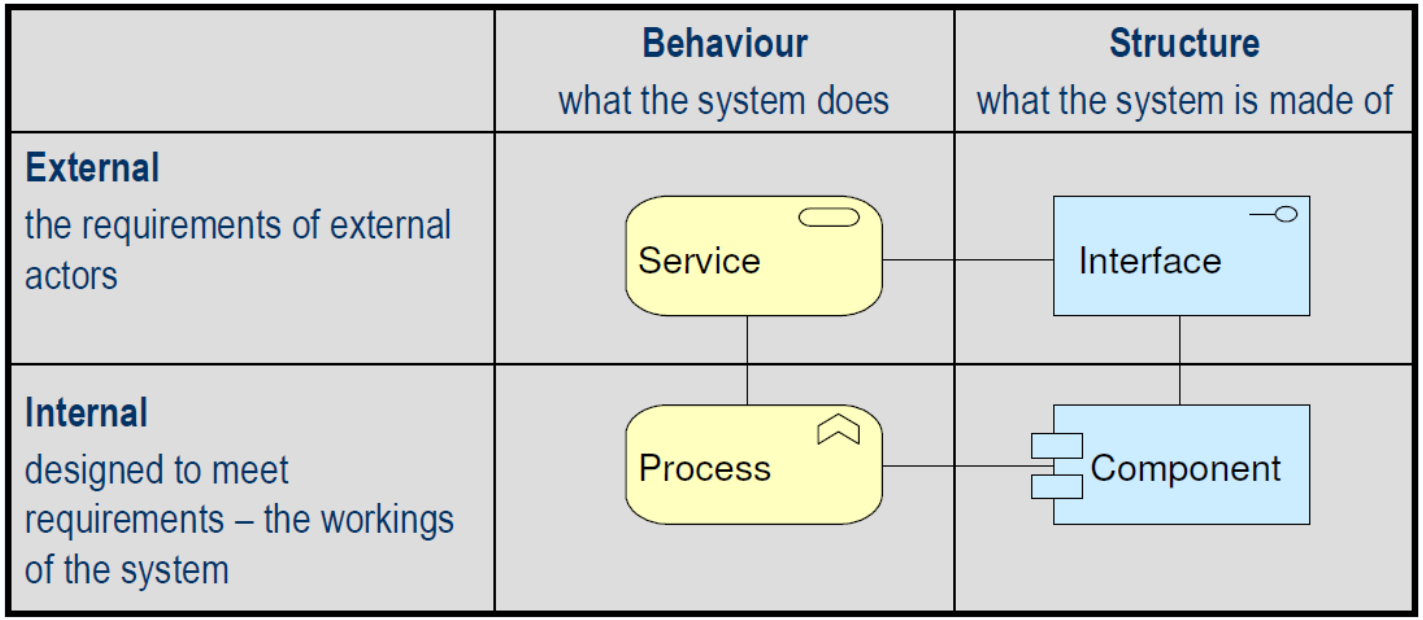


Service Definition

- No consensus
  - Tens definitions

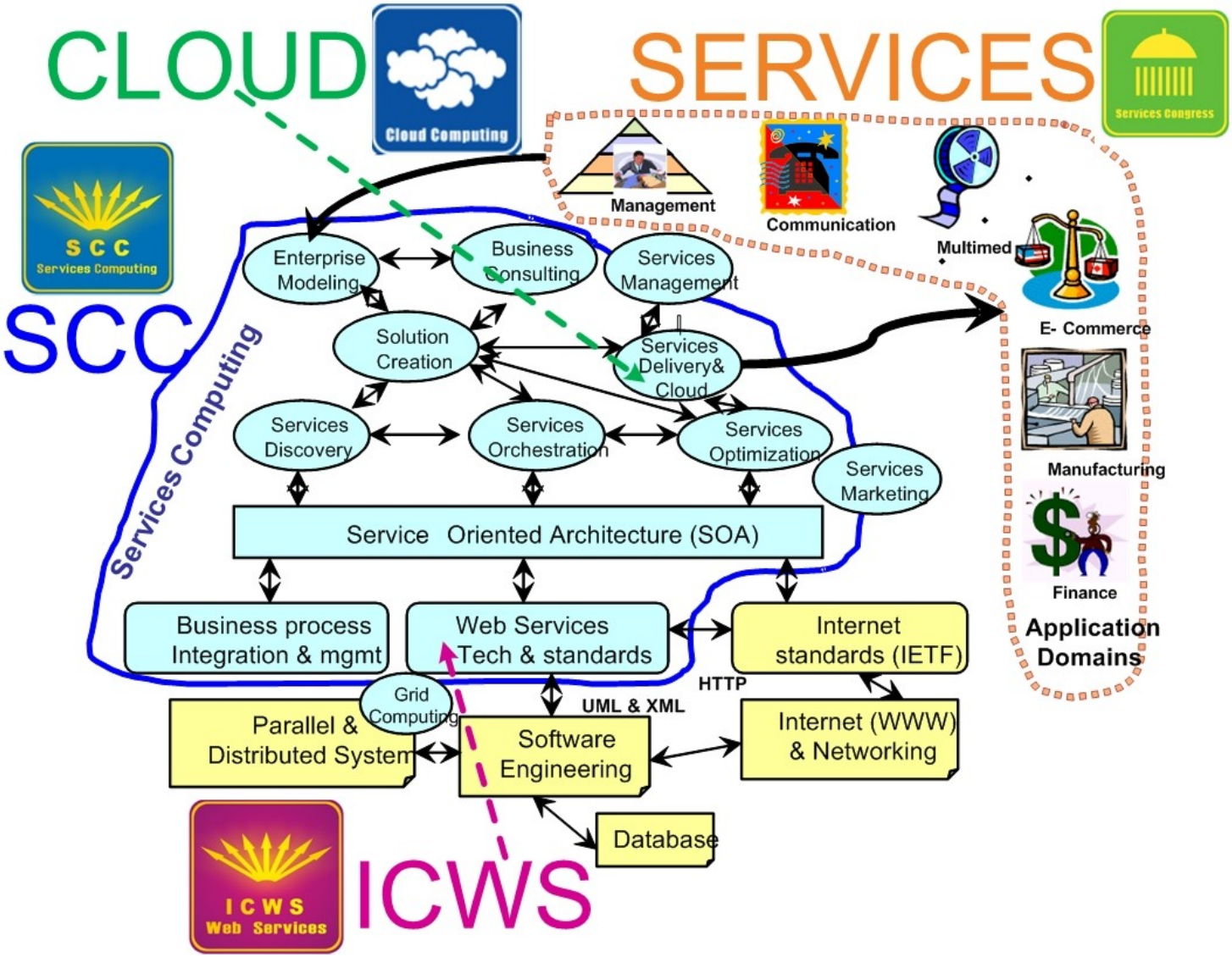
Another definiton of service

- All activity systems contain *components* and *processes*, which offer *services* via *interfaces*.

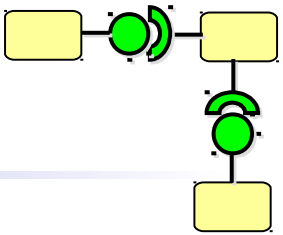


From Avancier Limited

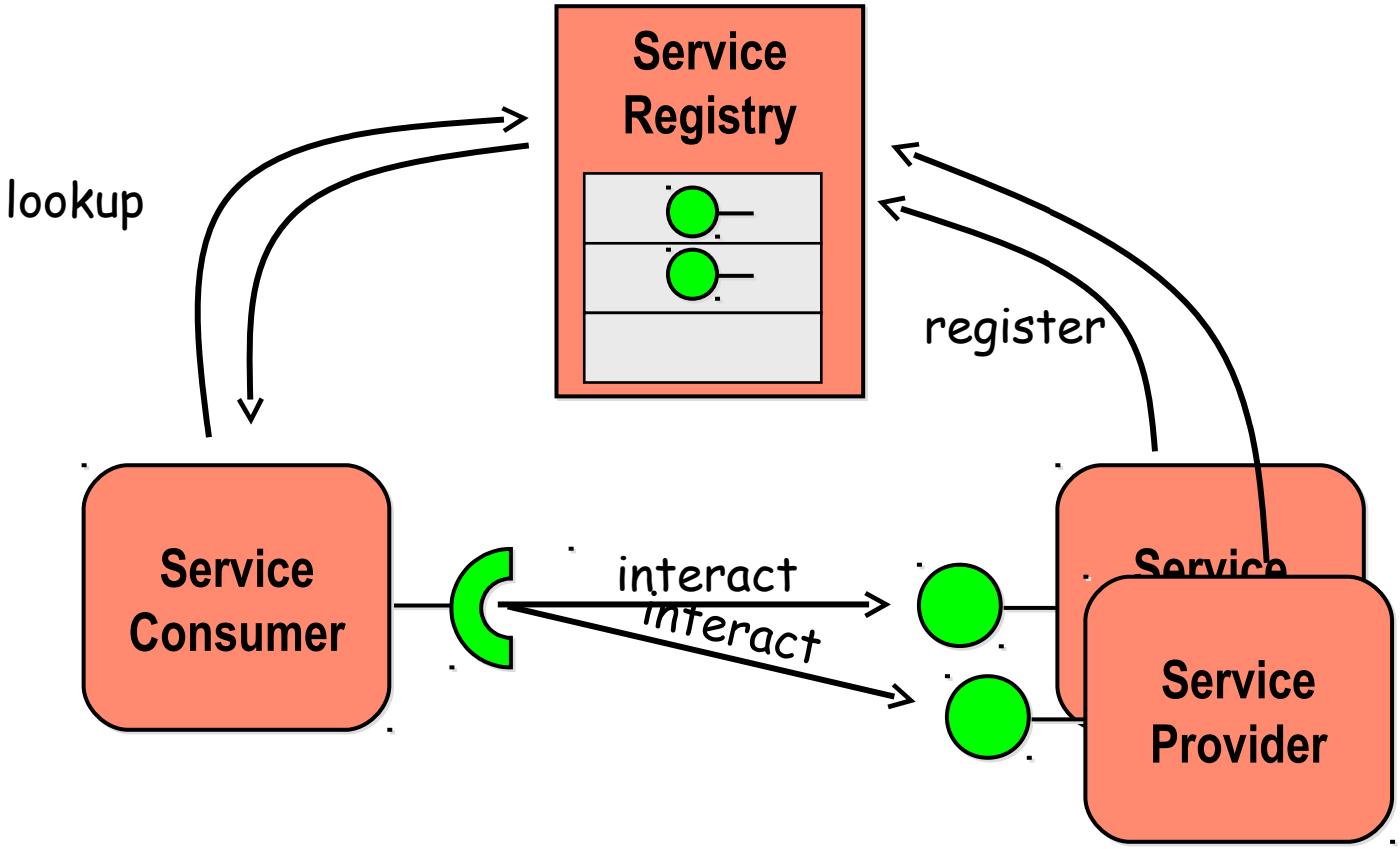
A Big Picture ;-(



http://conferences.computer.org/scc/2010/images/pic/SC-confs-Landscape.jpg



3 (Main) Entities of SOA

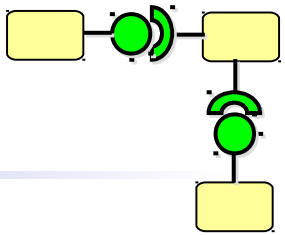


Service Characteristics

Bieber and Carpenter 2001, Stevens, Service-Oriented, 2002, Sun Microsystems, Jini Technology Architectural Overview 2001

- **Services are discoverable and dynamically bound. (Late Binding)**
- Services are self-contained and modular.
- Services stress interoperability.
- **Services are loosely coupled.**
- Services have a network-addressable interface.
- Services have coarse-grained interfaces.
- Services are location-transparent.
- Services are composable.
- Service-oriented architecture supports self-healing.
- Services are substituable

Trading and Late Binding



infrastructure
or adhoc

Service
Registry

search

register

Service
Consumer

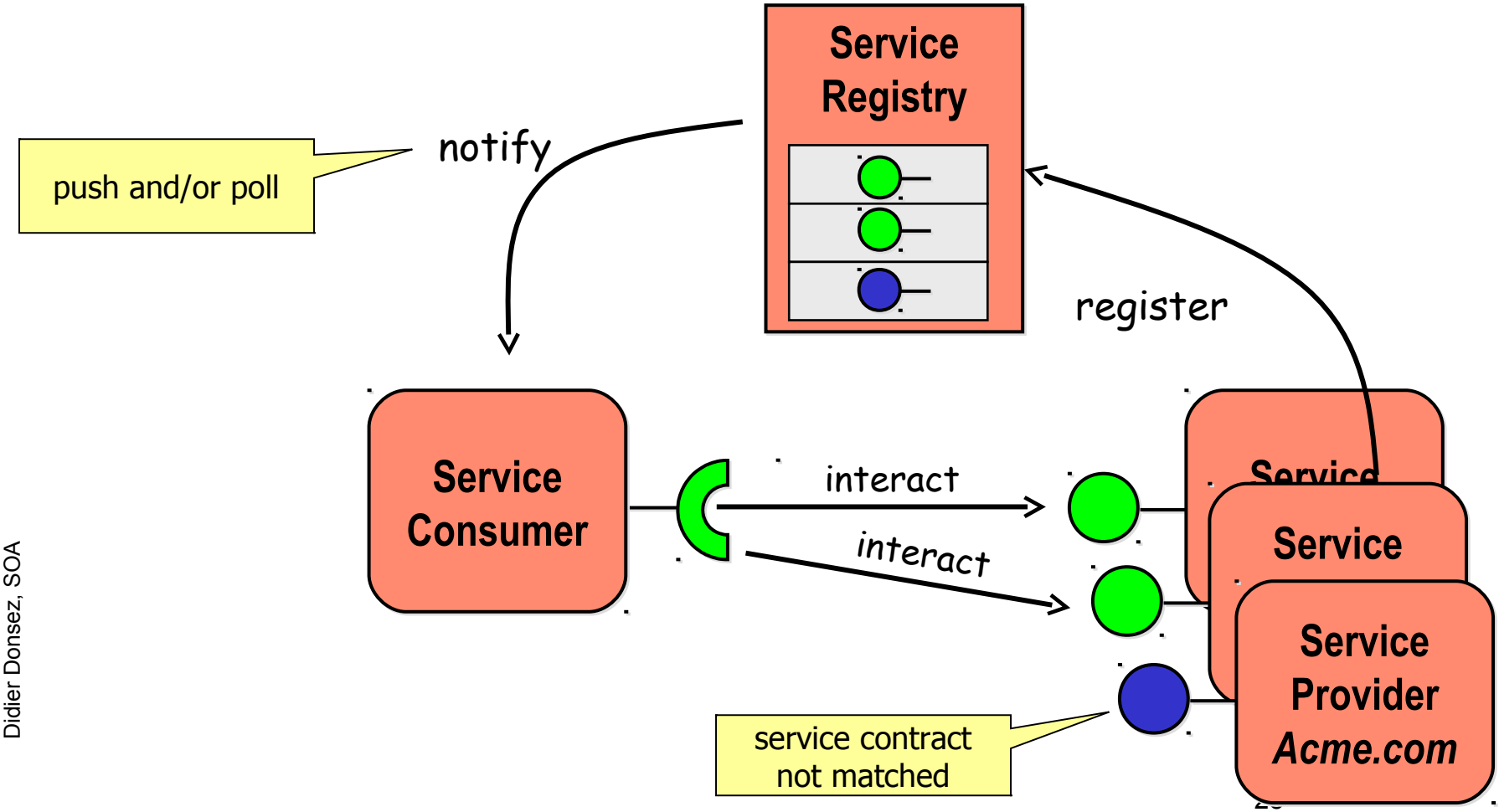
interact

Service
Provider

synchronous
or asynchronous

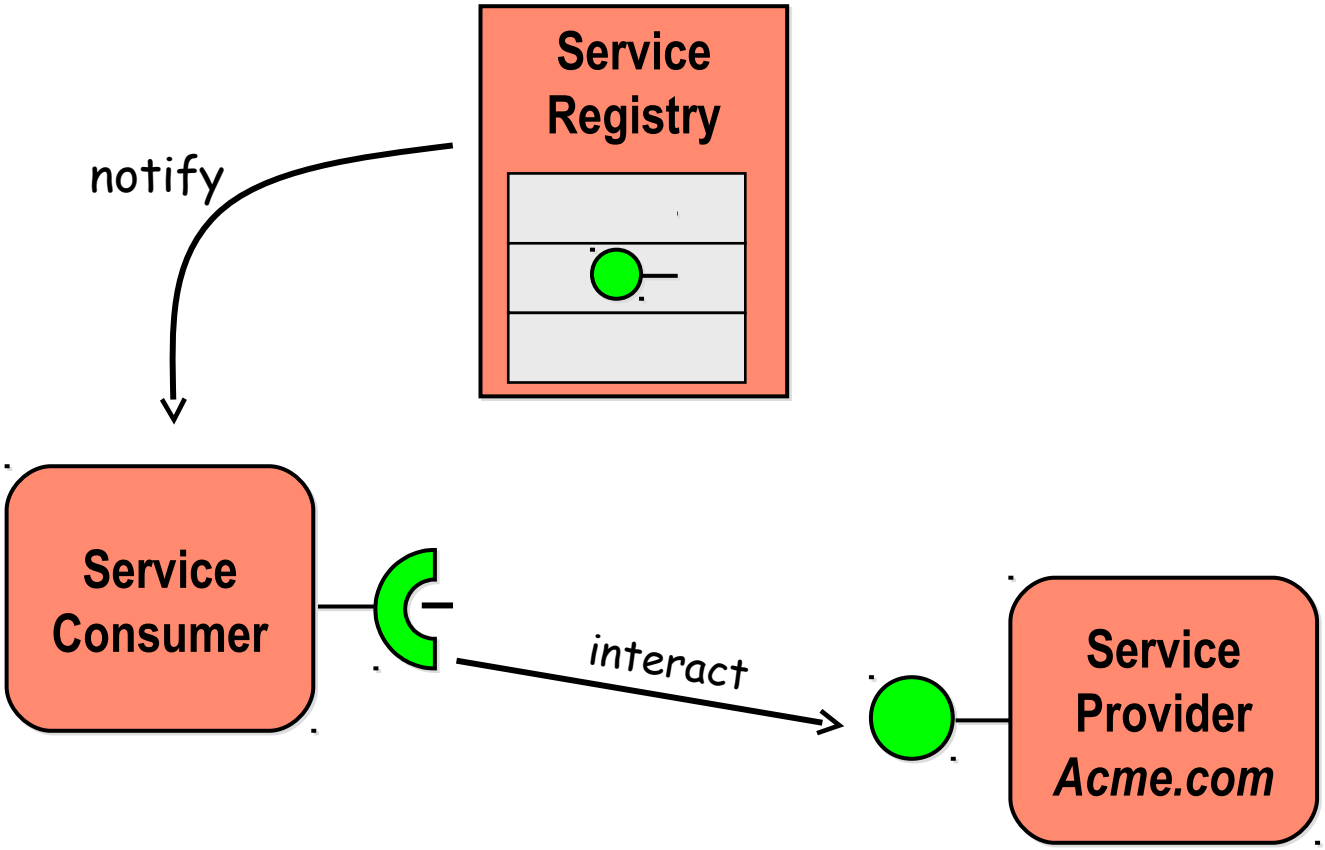
Reconfiguration at Runtime

- Dynamic arrival of new services



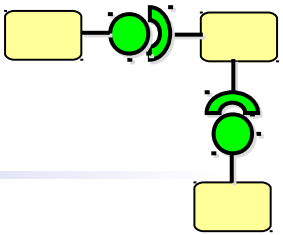
Reconfiguration at Runtime

- Dynamic removal of in-use services

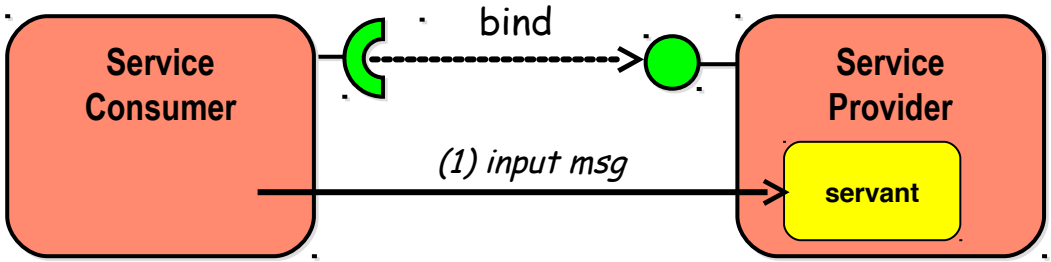


Interaction Types

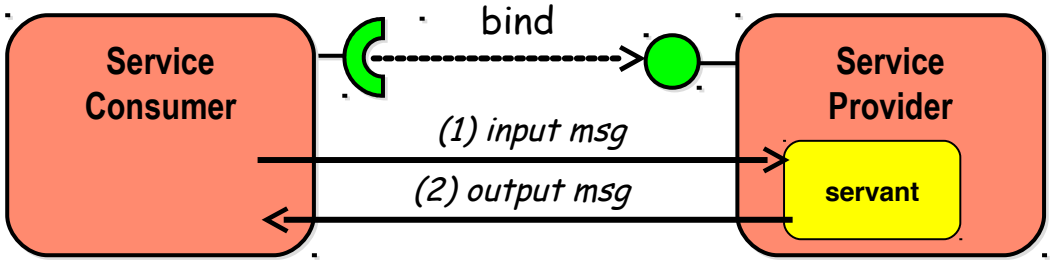
(Elément <operation> de WSDL)



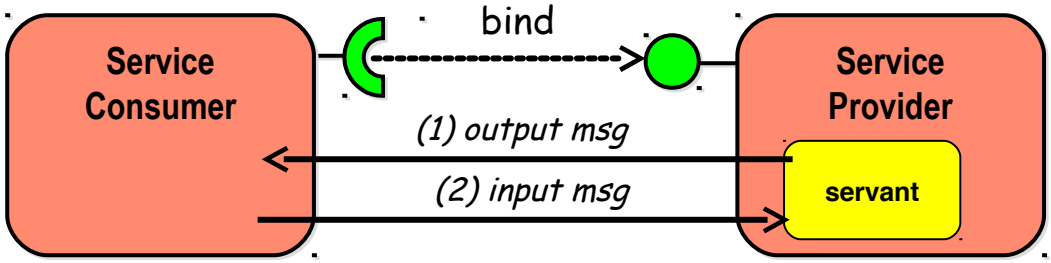
One-way



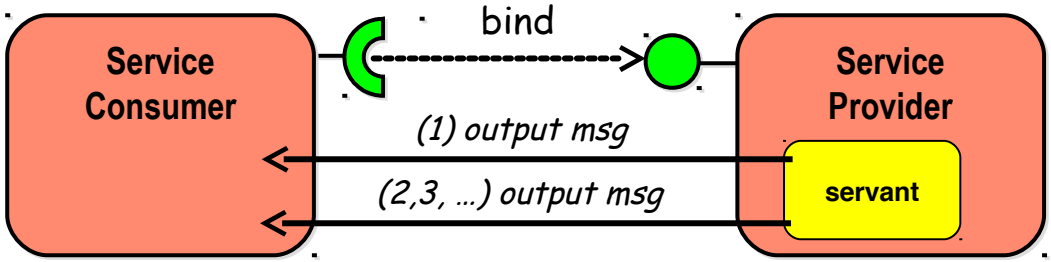
Request-Response



Solicit-Response



Notification



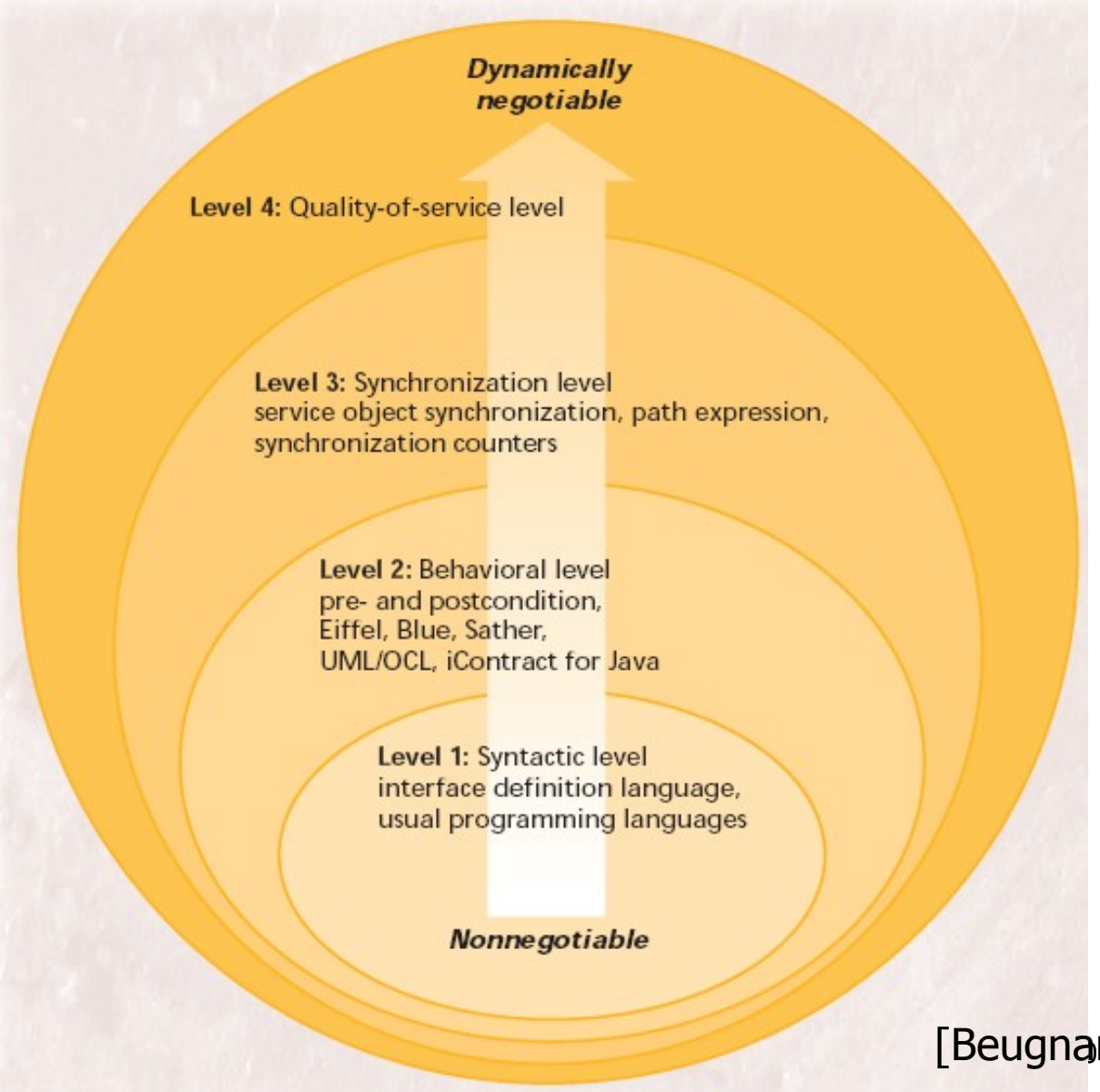
Interaction Types

- 1 request – 0..N responses (before timeout)
- PubSub
- Flow control (Wire Admin)

Service
Notion of Contract

- Forms
  - Various Representations : Java, WSDL, JSON, ...
  - Describe
    - Syntactic (L1),
    - Behavior (L2),
    - Synchronization (L3),
    - Quality of Service (L4)
- Conformity of a Service
  - To the Contract Terms
- Service Level Agreement (SLA)
  - Legal Contracts
    between the Service Consumer and the Service Provider
- Obligation
  - The contract commits applies in both directions :
    it commits the supplier as the applicant.

Service
Notion of Contract



[Beugnard et al 99]

Example of OSGi/Java Service

- A «published» Interface (L1)


```
package org.device.print;

public interface PrintService {

     public int print(OutputStream out,
                      String[] printparams)
             throws PrintException;

     public void kill(int jobnumber)
             throws PrintException;

     public Job[] list()
             throws PrintException;

     public Job status(int jobnumber)
             throws PrintException;

}
```

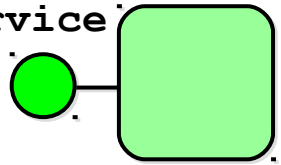
Example of OSGi/Java Service

- A Java «published» Interface (L1)
- Qualified by a set of properties (½ L4).
- Separation between Interface and Implementations

Interface

`org.device.print.PrintService`

*org.device.print.***dpimax=1200**
*org.device.print.***color=256**
*org.device.print.***type=laser**
*geo.***location=4th floor**



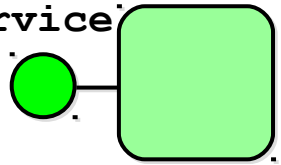
Implementation Class

`com.hp.printer.laserjet.impl.Driver`

Interface

`org.device.print.PrintService`

*org.device.print.***dpimax=2400**
*org.device.print.***color=2**
*org.device.print.***type=laser**
*geo.***location=1st floor**



Implementation Class

`com.lexmark.printer.laser.impl.Driver`

Service Trading in OSGi/Java (Query in LDAP Syntax)

- L1
  - All Print Services
    - (objectClass=org.device.print.PrintService)
    - (objectClass=org.device.print.*PrintService)
  - All Services with a contract of org.device
    - (objectClass=org.device.*)
- L1+L4 ½
  - Some Print Services
    - (& (objectClass=org.device.print.PrintService) (&(type=laser)(capability=double-sided)(!(dpi<=300))(location=*))
  - Fax and Print Services at the 4th Floor
    - (&(objectClass=org.device.print.PrintService)(objectClass=org.device.fax.FaxService) (location=4th floor))

Service Trading in OSGi/Java (Query in LDAP Syntax)

- L4
  - Color Print Services with discount / rush hour prices
  - Color Print Services with priority hour prices

Service Trading in OSGi/Java (Query in LDAP Syntax)

- L3 Synchronization
  - Concurrent
    - Serializibility (ACID)
  - Batch processing
    - First Arrived First Served
    - Shortest Job First

Example of WSDL

Web Services Description Language

- http://www.webservicex.net/New/
- http://www.webservicex.net/New/Home/ServiceDetail/56
- http://www.webservicex.net/globalweather.asmx/GetCitiesByCountry
- http://www.webservicex.net/globalweather.asmx?WSDL

Example with WSDL

Web Services Description Language

```
<definitions name="HelloService"
   targetNamespace="http://www.examples.com/wsdl/HelloService.wsdl"
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:tns="http://www.examples.com/wsdl/HelloService.wsdl"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <message name="SayHelloRequest">
      <part name="firstName" type="xsd:string"/>
   </message>

   <message name="SayHelloResponse">
      <part name="greeting" type="xsd:string"/>
   </message>

   <portType name="Hello_PortType">
      <operation name="sayHello">
         <input message="tns:SayHelloRequest"/>
         <output message="tns:SayHelloResponse"/>
      </operation>
   </portType>
```

Example with WSDL

Web Services Description Language

```
   <binding name="Hello_Binding" type="tns:Hello_PortType">
      <soap:binding style="rpc"
         transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation name="sayHello">
         <soap:operation soapAction="sayHello"/>
         <input>
            <soap:body
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
               namespace="urn:examples:helloservice"
               use="encoded"/>
         </input>

         <output>
            <soap:body
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
               namespace="urn:examples:helloservice"
               use="encoded"/>
         </output>
      </operation>
   </binding>

   <service name="Hello_Service">
      <documentation>WSDL File for HelloService</documentation>
      <port binding="tns:Hello_Binding" name="Hello_Port">
         <soap:address
            location="http://www.examples.com/SayHello/" />
      </port>
   </service>
</definitions>
```

Exercice

- Write the binding for SMTP/IMAP

Exercice

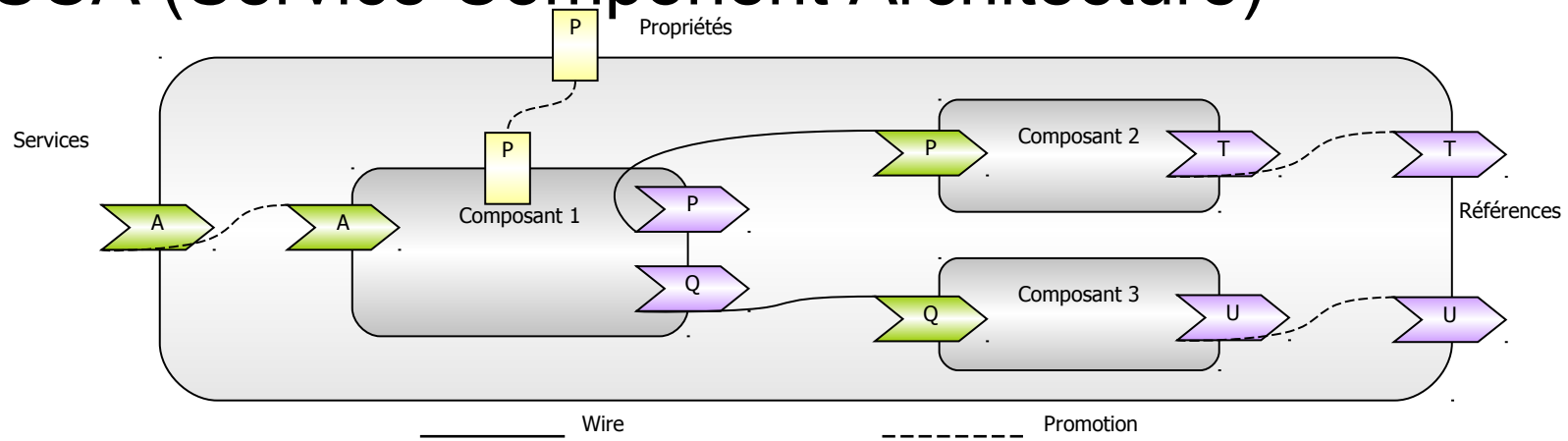
- Give the WSDL of 1 service provider in the Benzene usecase.

Service Composition

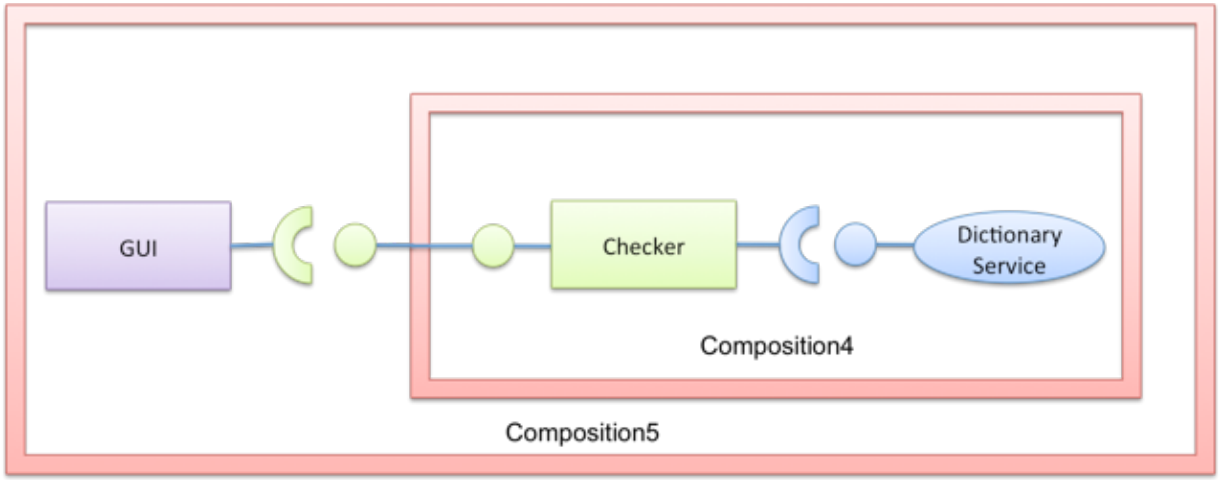
- Vertical Composition
  - Structural Composition
- Horizontal Composition
  - Composition by Process

Structural Composition

- SCA (Service Component Architecture)



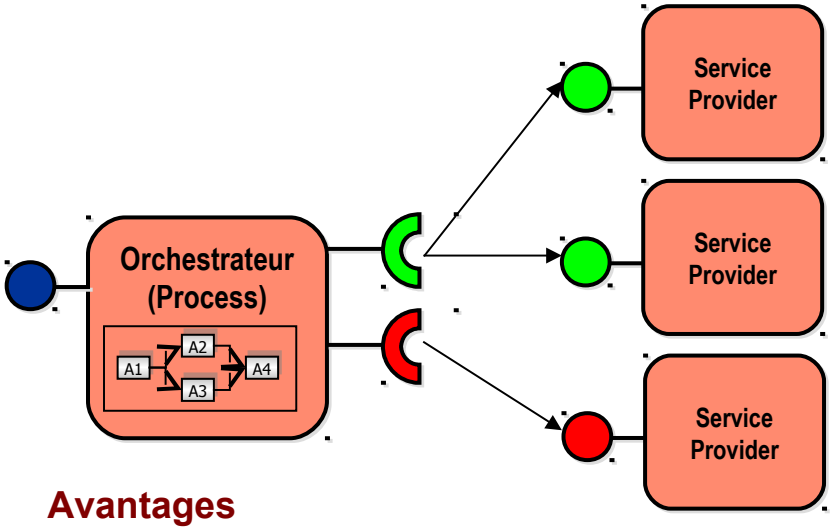
- iPOJO



Thanks to Gabriel Pedraza

Composition by Process

- Orchestration versus Choregraphy
  - BPM, BPEL, XPDL, FOCAS

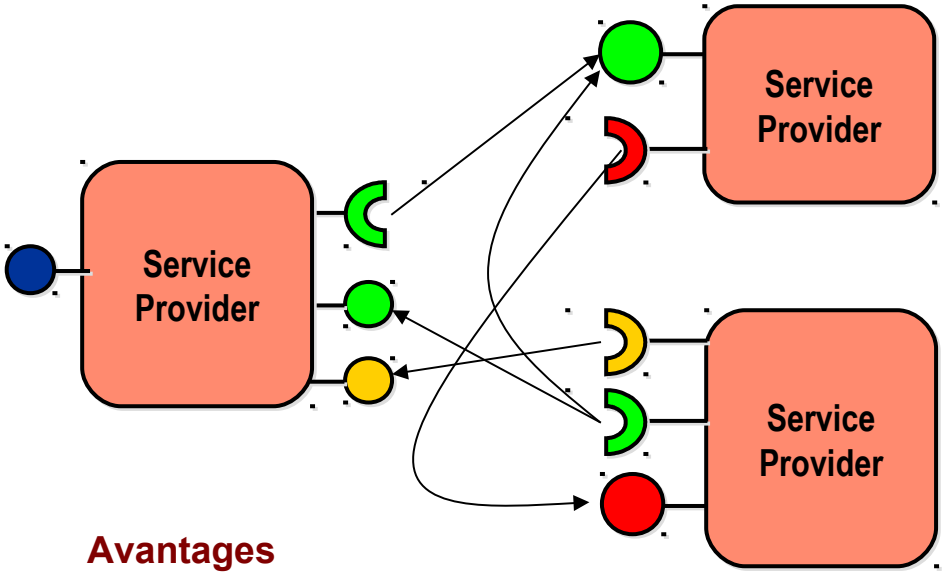


Avantages

- Easy modeling
- Monitoring, Management

Drawbacks

- Bottleneck
- Application Performance
- Scalability



Avantages

- Scalability
- Performance

Drawbacks

- Modelling
- Top Down Execution

Thanks to Gabriel Pedraza

Composition by Process Example

5.1 Shipment Process of a Hardware Retailer

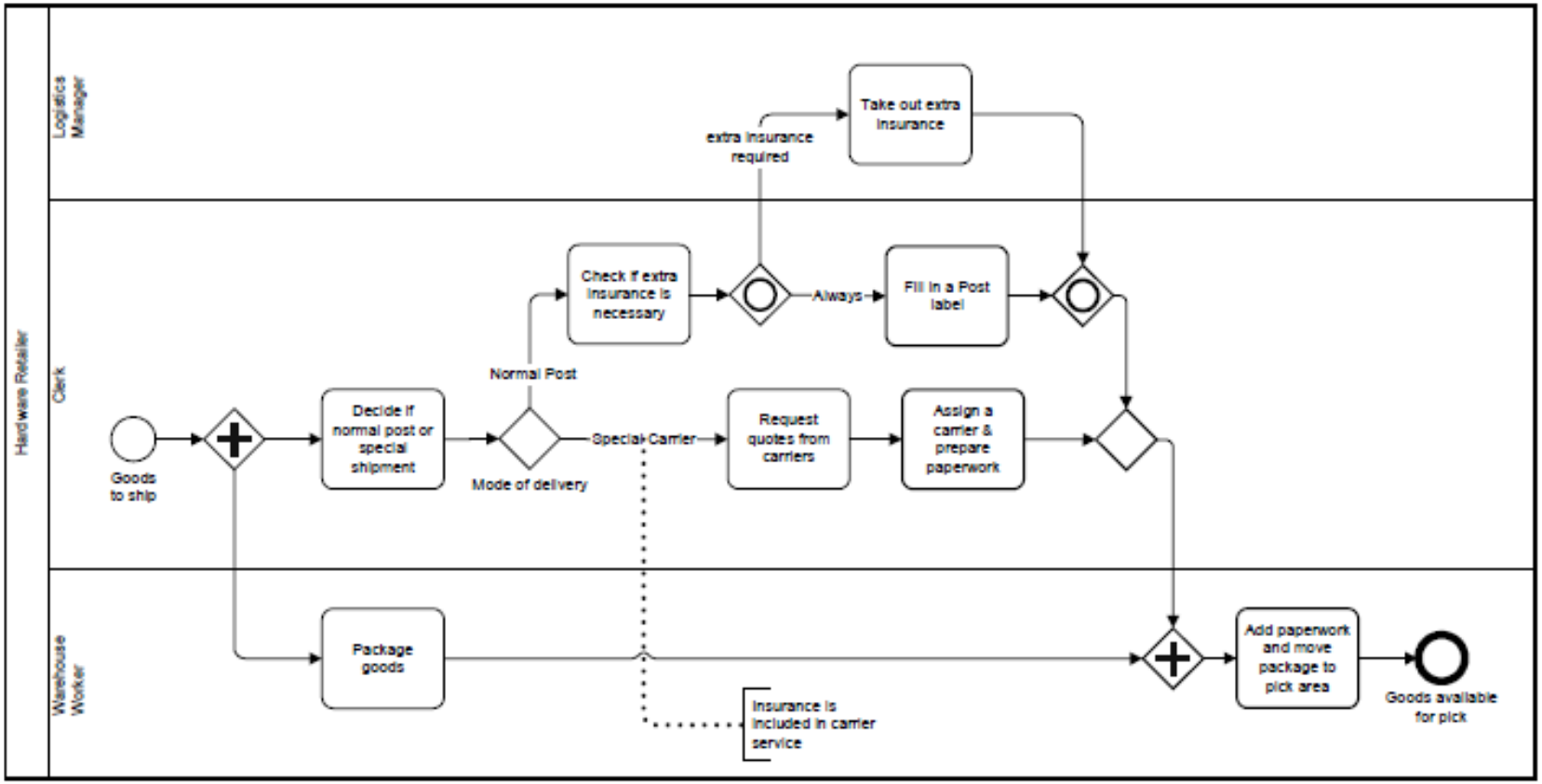
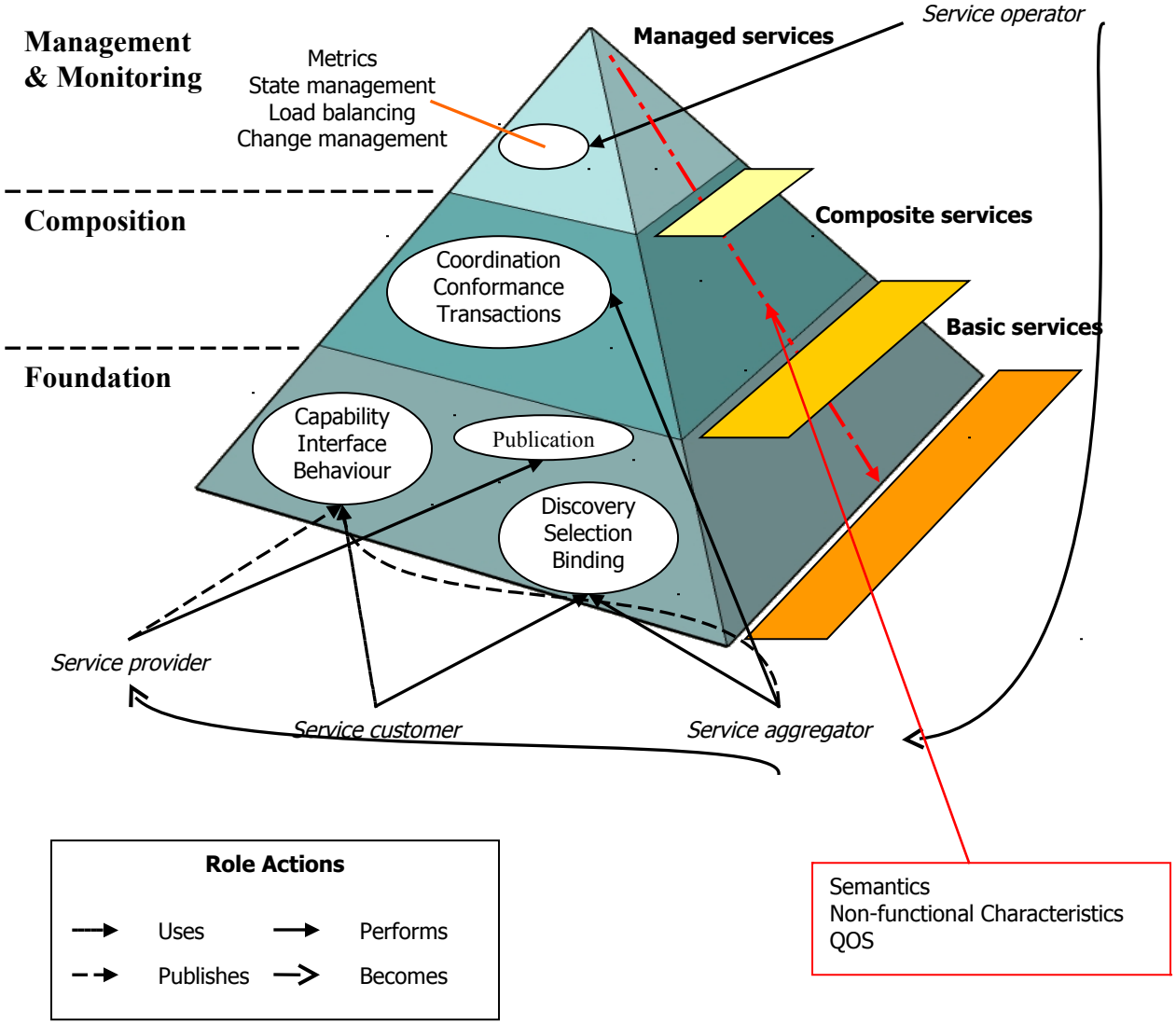


Figure 5.1: Shipment Process of a hardware retailer

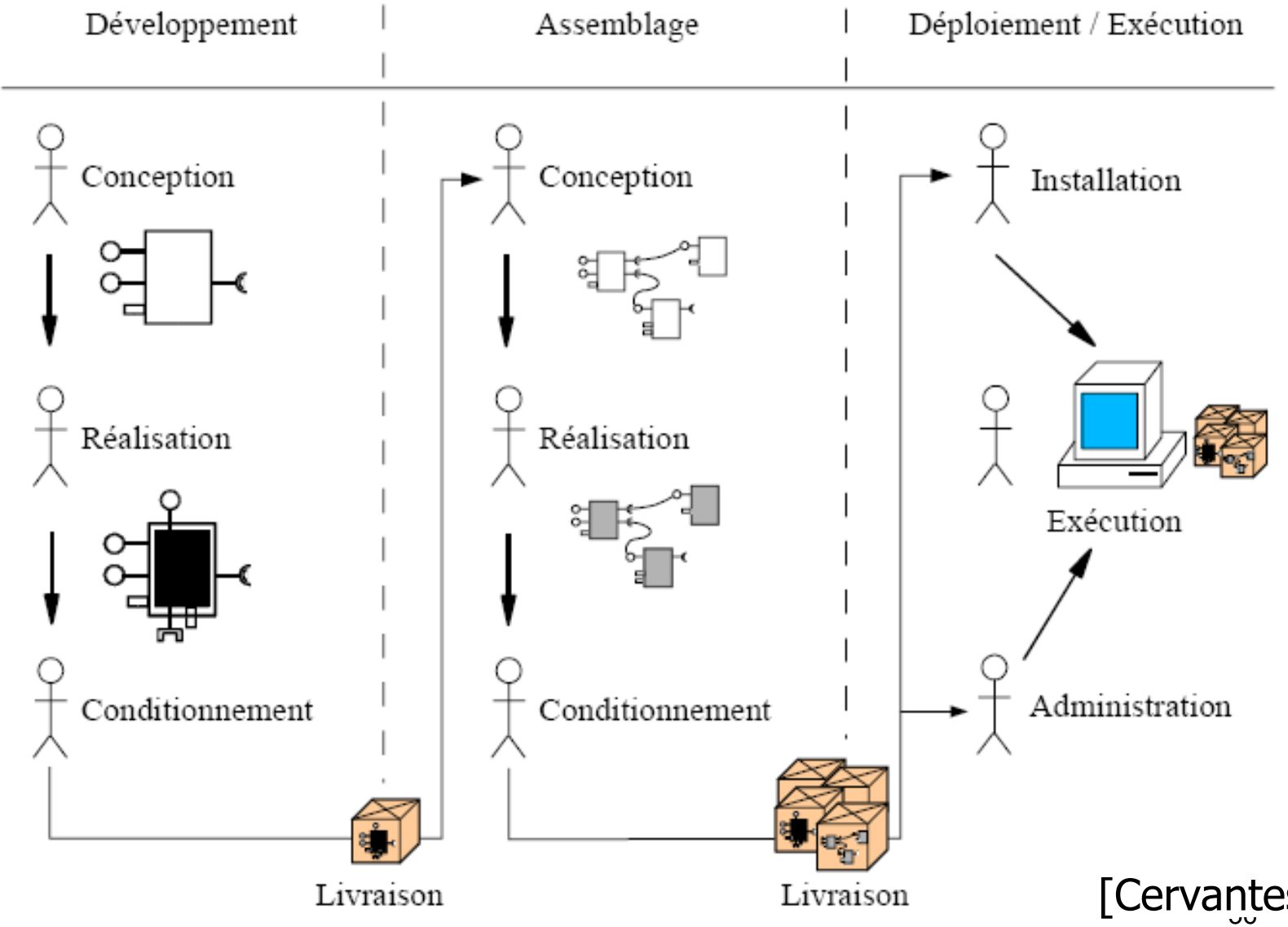
Exercice

- Modeling the Benzene use-case with
- 1 orchestration
- 1 choregraphy
- 1 structural composition

SOA Pyramid [Papazouglu]

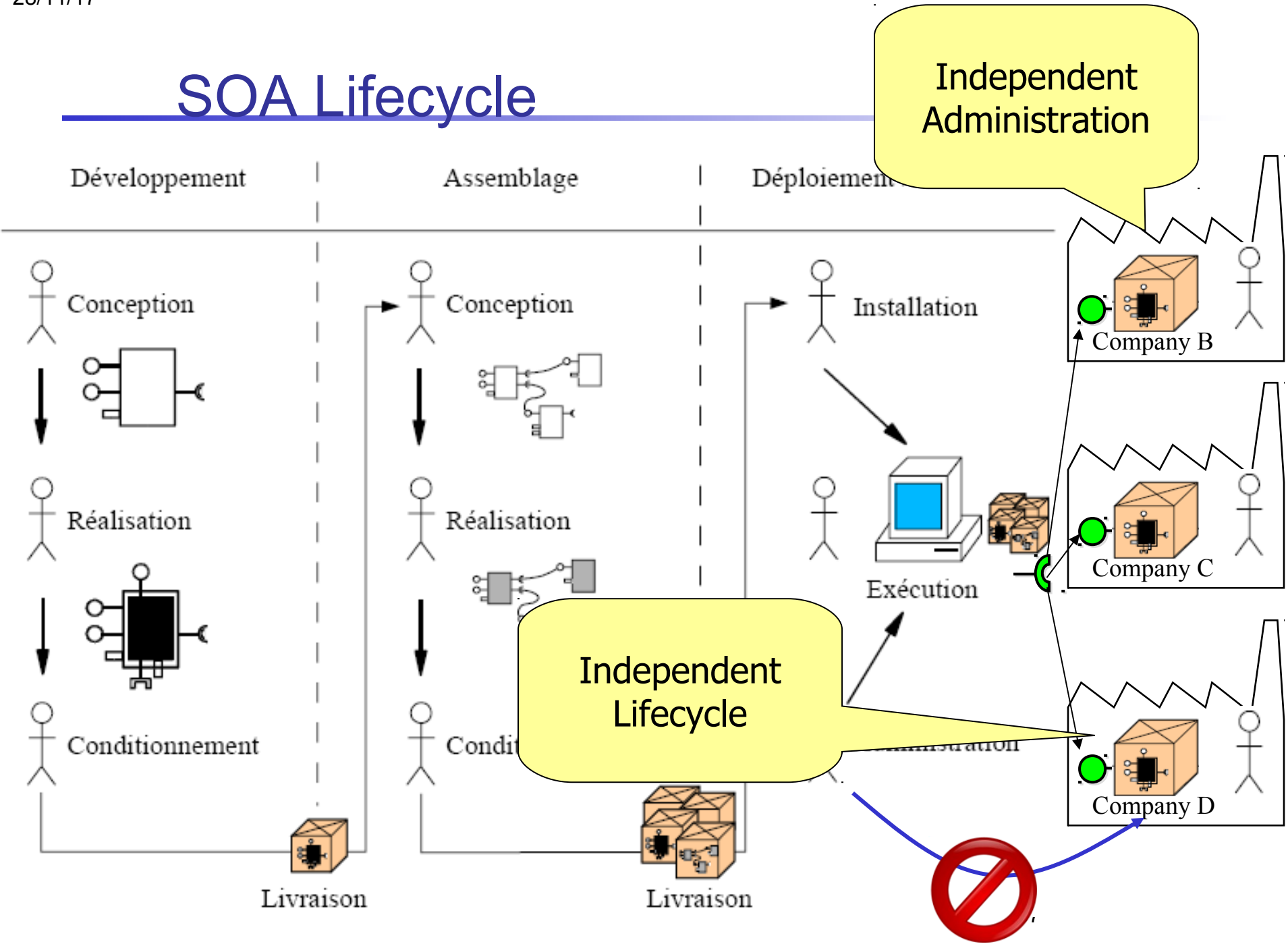


Reminder : Component Lifecycle



[Cervantes 04]

SOA Lifecycle



Servive-Level-Agreement

- Contract between service provider and service requester/consumer
  - Price of the Service
  - Quality of Service expected by the consumer

A brief history of SLA

Network QoS

- Autonomous system
- Telecom
- Throughput, response time, jitter, ...

Web hosting

- 99,999 % availability
  - 5 nines → 5.26 minutes/year of downtime
- 1 Gb storage space

Outsourcing

- Applications hosting

Service Level Agreement

This sample is a short form contract used to both document the SLA and report monthly on its status. One of these is produced for each service provided.

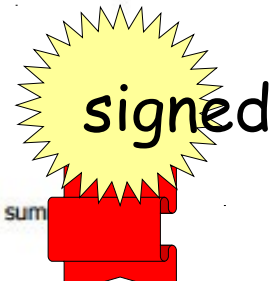
Between IT Department And ABC Department	Date: MM/YY
Contacts: **IT Department:** _____ **ABC Department:** _____	**Effective Dates: From MM/YY to MM/YY**
Approvals: **IT Department:** _____ **ABC Department:** _____	

Service Level Agreement

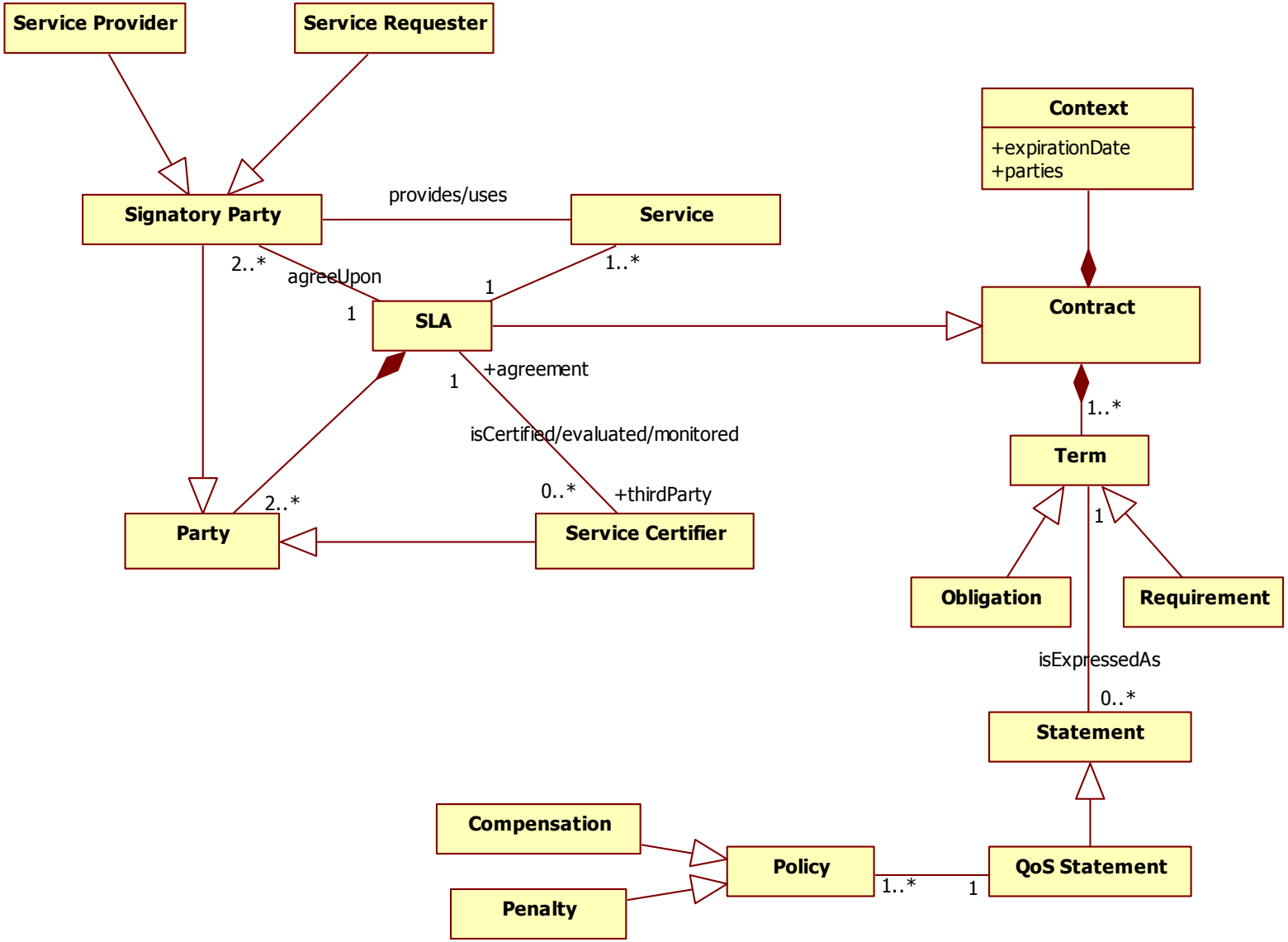
CICS Service				Actual	Difference
	Availat			100%	2%
	Response	% of response within 2 seconds (internal)	90%	95%	5%
		% of response within 2 seconds (internal)	95%	95%	0%
	Load	Transactions/min during peak (9 am – 11 am)	300	250	-50
		Daily CPU hours	3.5	3.0	-.5
	Accuracy	Errors due to DC Problems	0	0	0
		Errors due to Applications	0	0	0
	Batch Service	Class S: % turnaround in 30 minutes	95%	85%	-10%
		Class T: % turnaround in 15 minutes	98%	100%	2%

SLA Criteria:

1) Availability based on CICSPROD up and files open.
2) Penalties for missed services:
   a) 10% reduction in billing for 2% missed service unless miss caused by user.
3) Penalties for exceeded loads:
   a) 10% increase in billing and no penalty for missed service.
4) Reporting: Data Center will provide this report by 8 am each day. Weekly report will sum service for the week.



SLA concepts



Domain-specific SLAs

Network	Video	Web hosting	Ressources	Service disruptions
• Throughput • Response time • Latency • Jitter	• Frames / sec • Resolution • Codec • Error rate • Time-shifting	• Response time • Max clients • Hits / sec	• CPU • Espace mémoire • Mode alimentation • Autonomie • Empreinte physique	• Uptime • Downtime • Availability • Max interruption time / day • Min time between 2 interruptions • …

SLA phases

- SLA description
  - Formalisms, semantics, content
- SLA negotiation
- Service Level Management activities
  - Compliance monitoring
  - Policies enforcement, reactions
  - Penalties, compensations
  - Contract termination

Negotiation

1st level : selection

- Service provider publishes its *specification*
  - Extended service specification (functional & extra-functional)
  - QoS-level contract (Beugnard, 1999)
- A service is selected according to its *specification*
  - Scoring
  - Service ranking

Negotiation

2nd level : customization

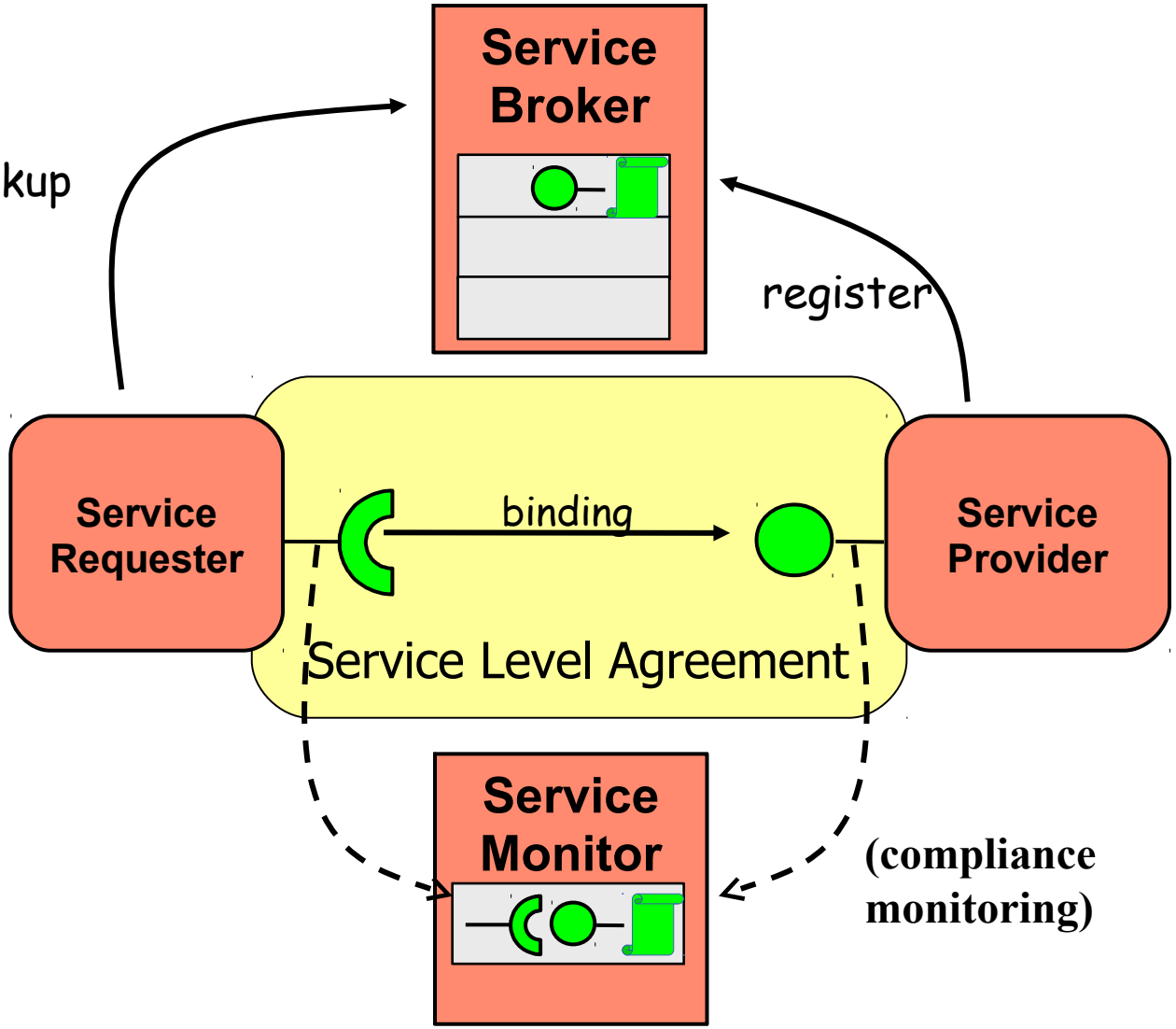
- Service provider publishes pre-defined offers
  - Fixed QoS levels
  - e.g. Premium, gold, silver, best-effort
- Or non-fixed contract terms
  - Negotiable
  - Range of QoS properties
    - can modify pricing for example

Negotiation

3rd level : discussion

- Actual negotiation
  - Parties go through a complete negotiation process
- A service provider may reevaluate its service specification in order to meet the requester's requirements
- A requester may lower its requirements in order to use a service

Service Level Management



Monitoring

Logging

Reporting

Billing

...

State of the Art

Formalisms

- SLAng
- Rule-based SLA
- WSLA (Web Service Level Agreement) specification

Frameworks

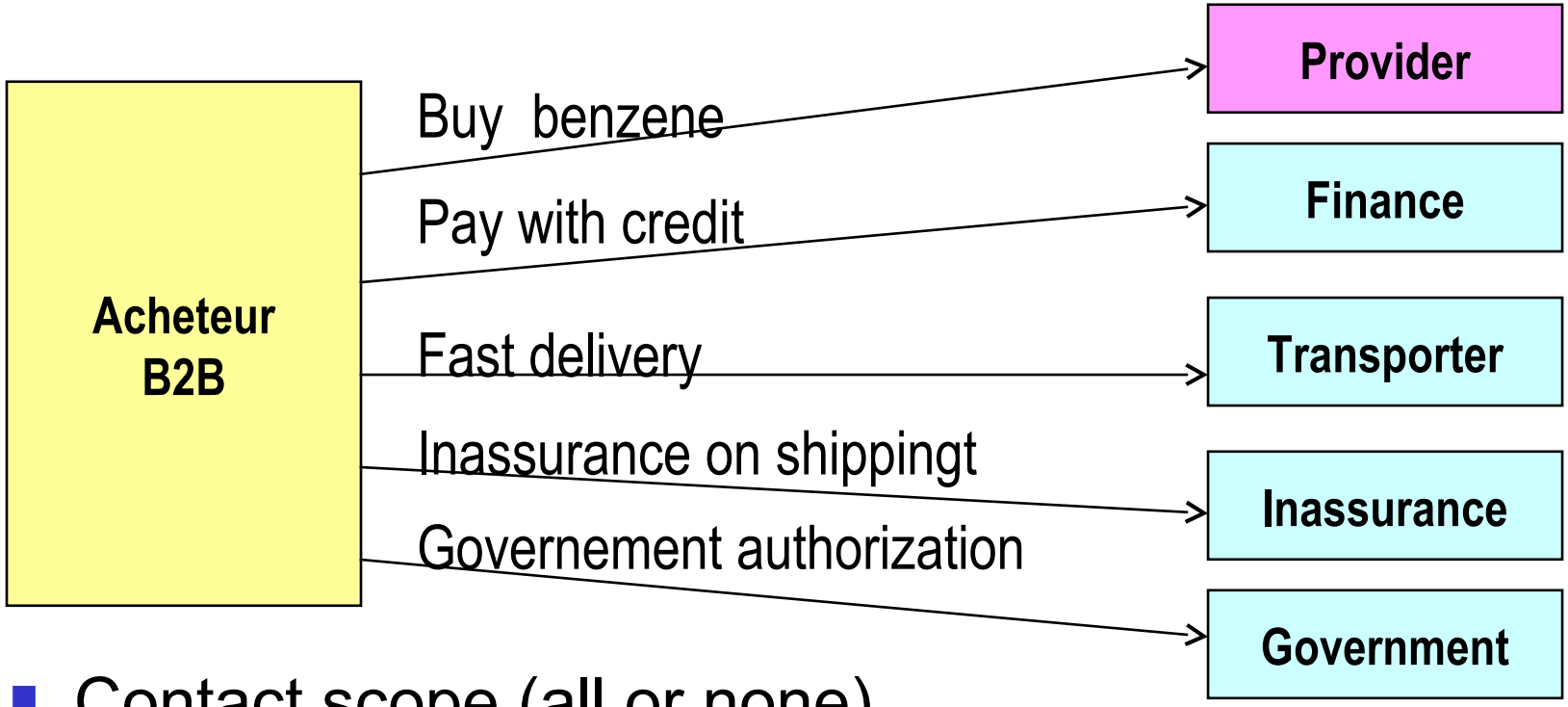
- ContractLog
- Compliance monitor (WSLA)

WS-Agreement specification

- Main standard
- Cremona framework (IBM)

Case of Multi-party Contract

- Example
  - Buy Benzene to a provider
  - require additionnal services provided by thrid-parties



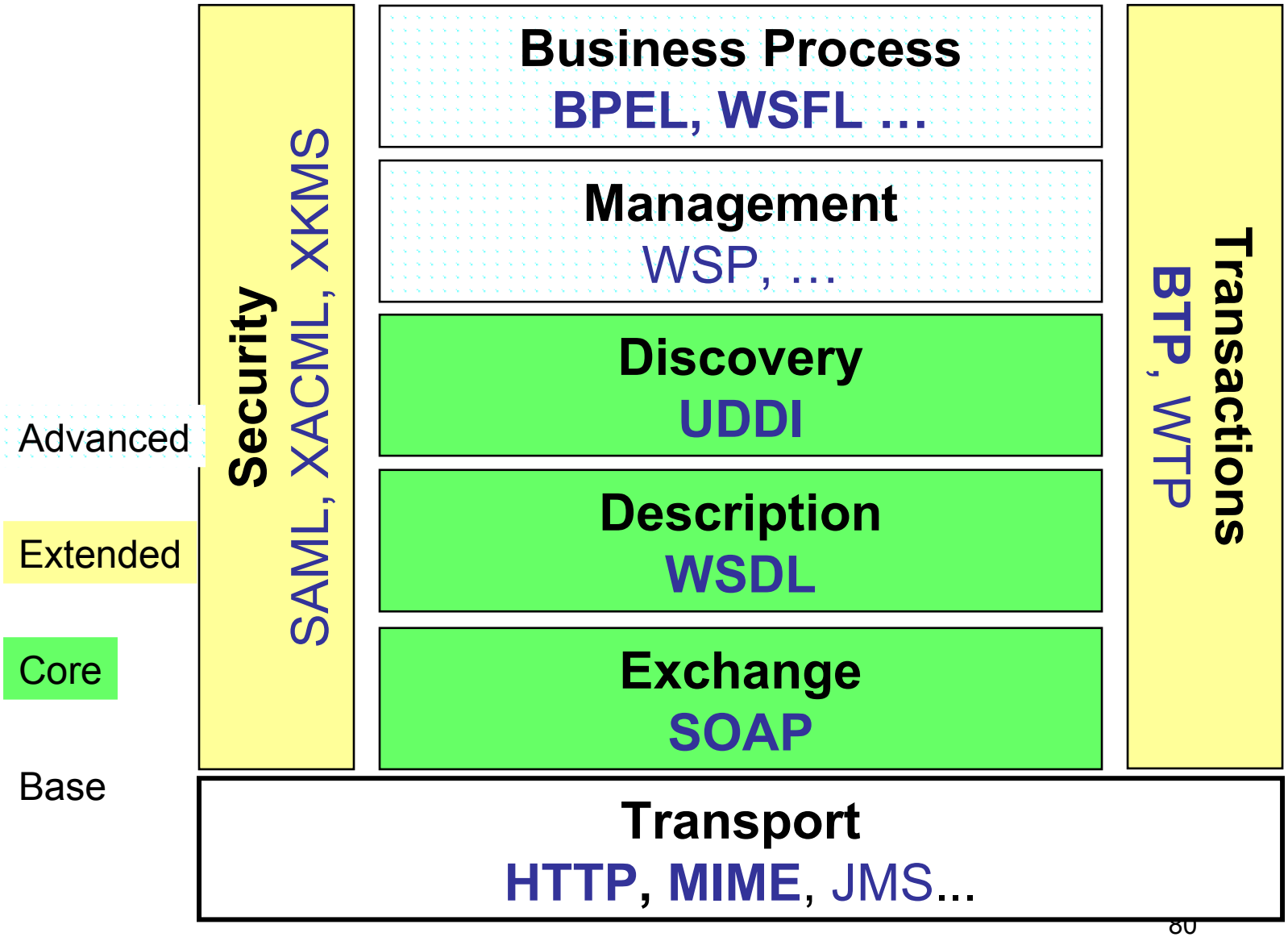
- Contact scope (all or none)
  - Multi-Party Agreement

Main Real Life Specifications and Frameworks

- Distributed
  - Web Services (W3C, OASIS, …)
  - UPnP (Universal Plug and Play), DPWS, DLNA
  - *DNS-SD*
  - *SLP*
  - *JINI*
- *Centralized*
  - OSGi (for Java)

- *Similar but …*
  - *REST*
  - *ESB (Event-Driven SOA)*

Web Services Global picture



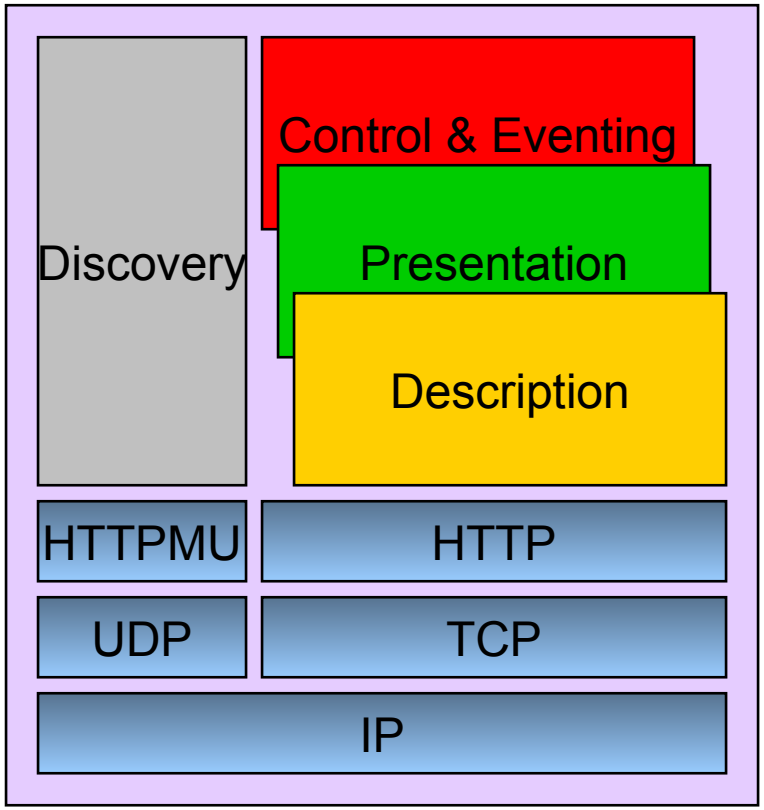
UPnP : Universal Plug and Play

- SOA for (IP-enabled) Consumer Electronic
  - Smart TV, ASDL Boxes, Media Servers, Security Cameras …
  - Basic specification for well-known DLNA

- Main entities
  - Devices
    - provide 1..N services
      - Standard devices and services
      - Extensible
  - Control Points
    - discover devices
    - consume services

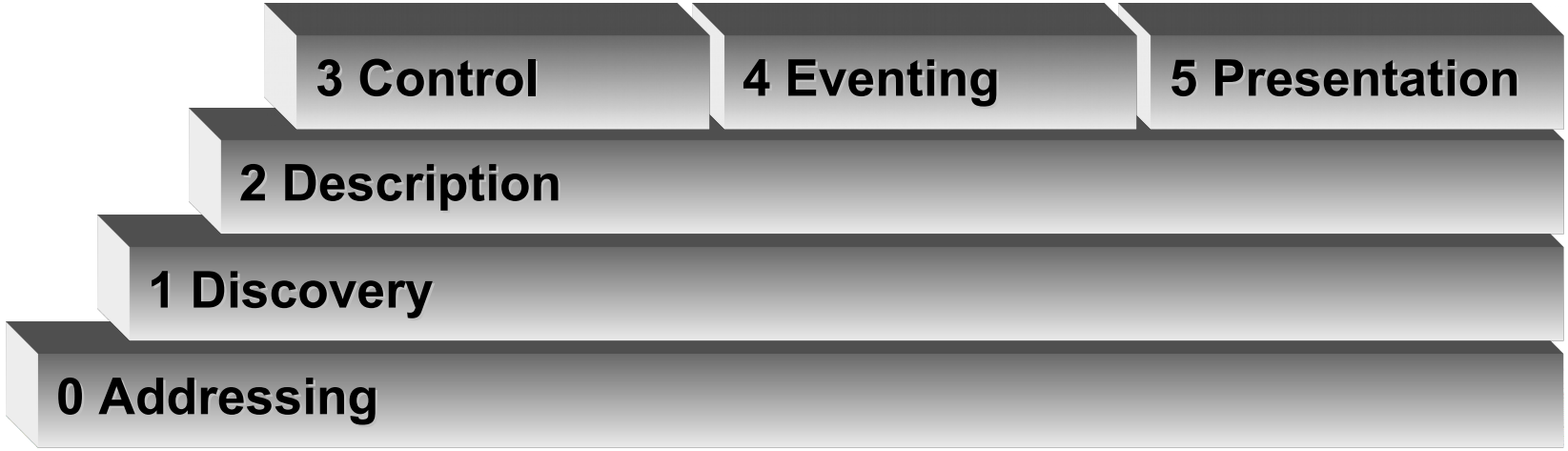
Anatomy of a UPnP Device

- Networking stack
- Discovery server
- Description server
- Presentation server
- Control & Eventing Services



D'après UPnP Forum

UPnP
How Does It Work ?



0 Devices and control points get a local address in the LAN

1 The control point discovers the available devices in the LAN

2 The control point inspects the device capabilities

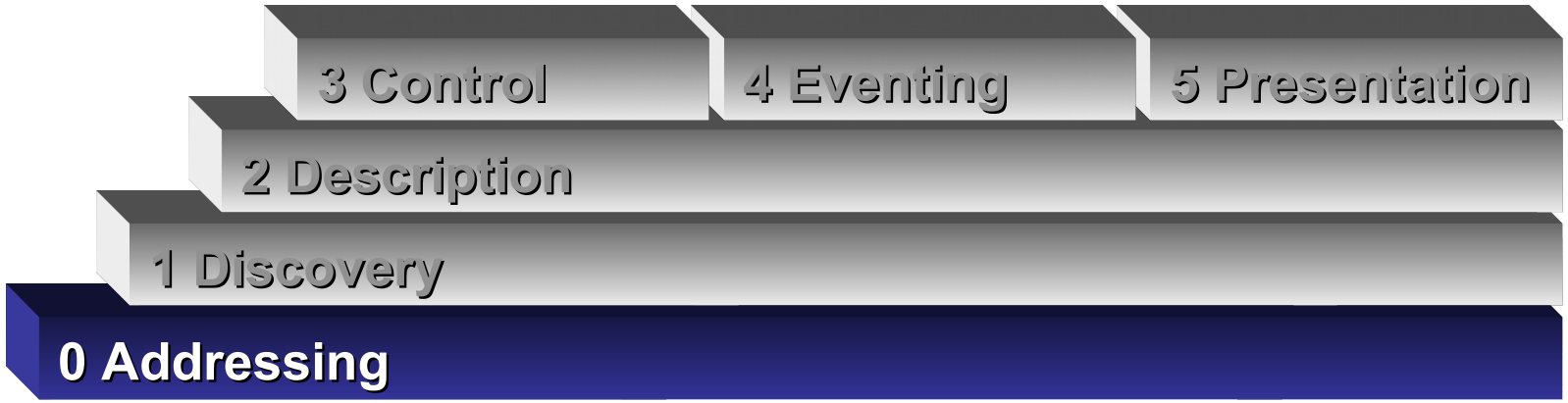
3 The control point invoke (remotely) an action of the device

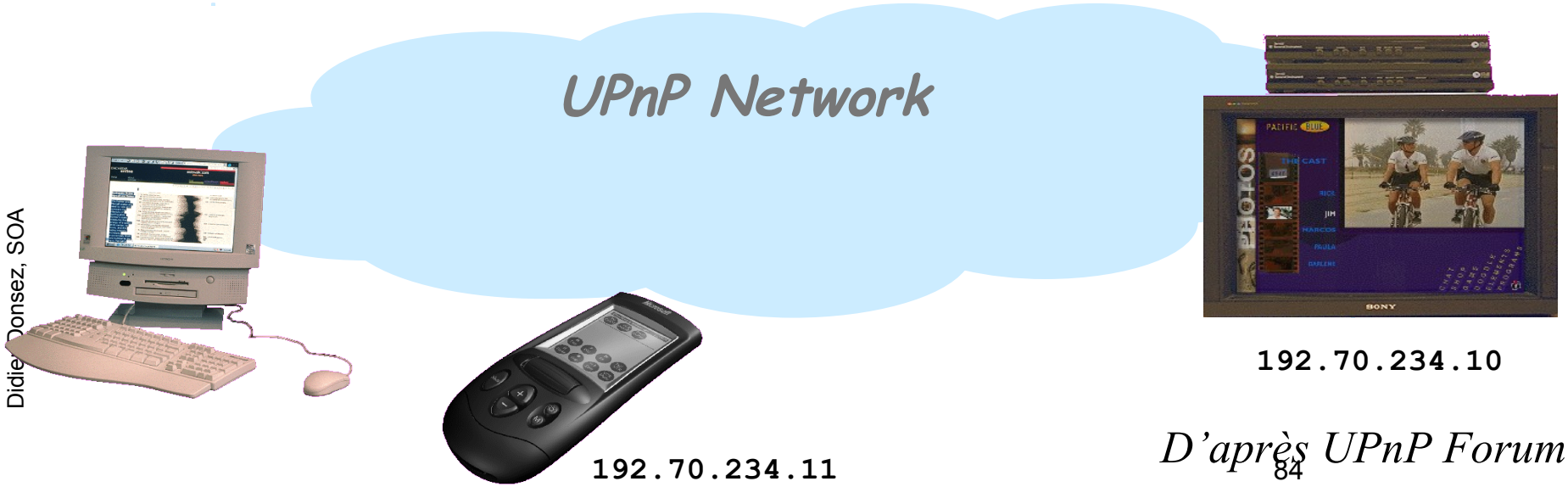
4 The control point is notified of value changes of the state variablesof the device

5 A web browser can display the webpage from the webserver embedded in the device

From UPnP Forum

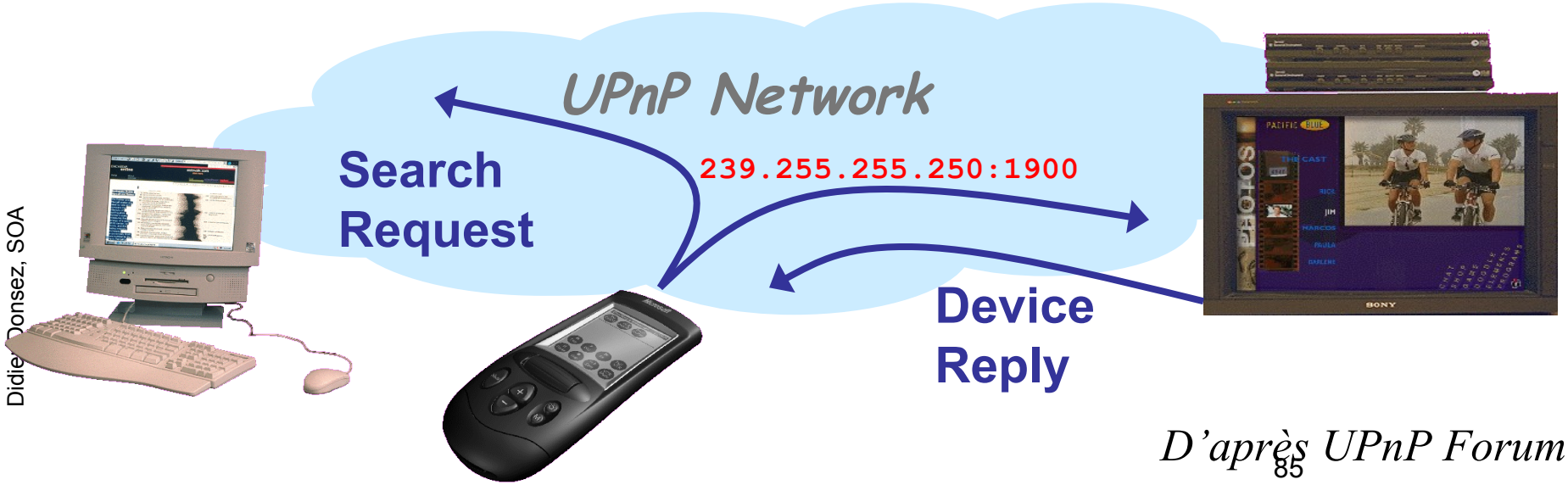
0 Addressing: DHCP or AutoIP (ARP)





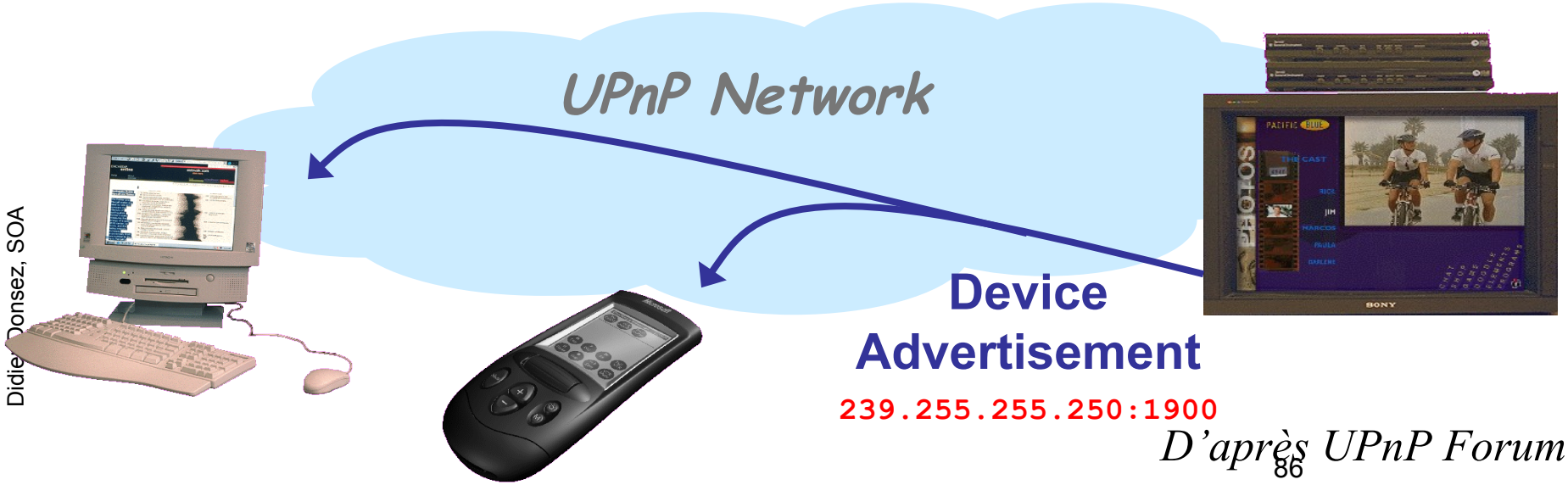
D'après UPnP Forum

1 Discovery: SSDP



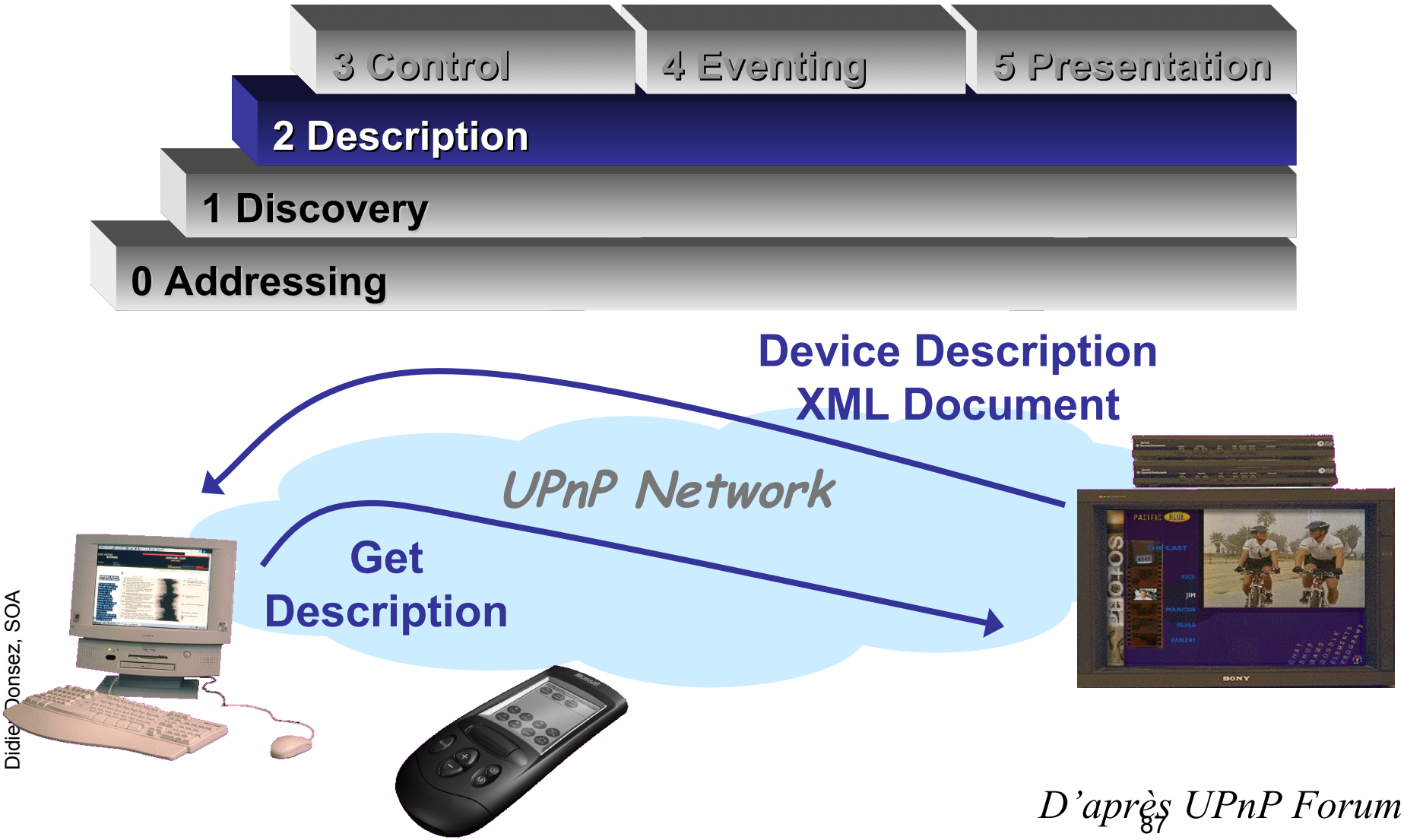
D'après UPnP Forum

1 Discovery: SSDP



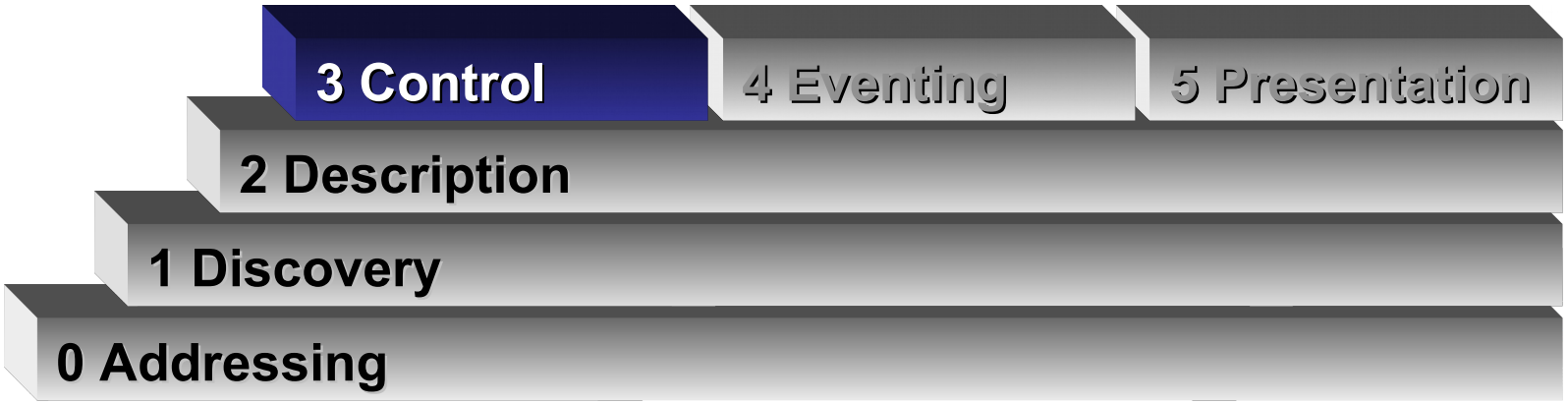
D'après UPnP Forum

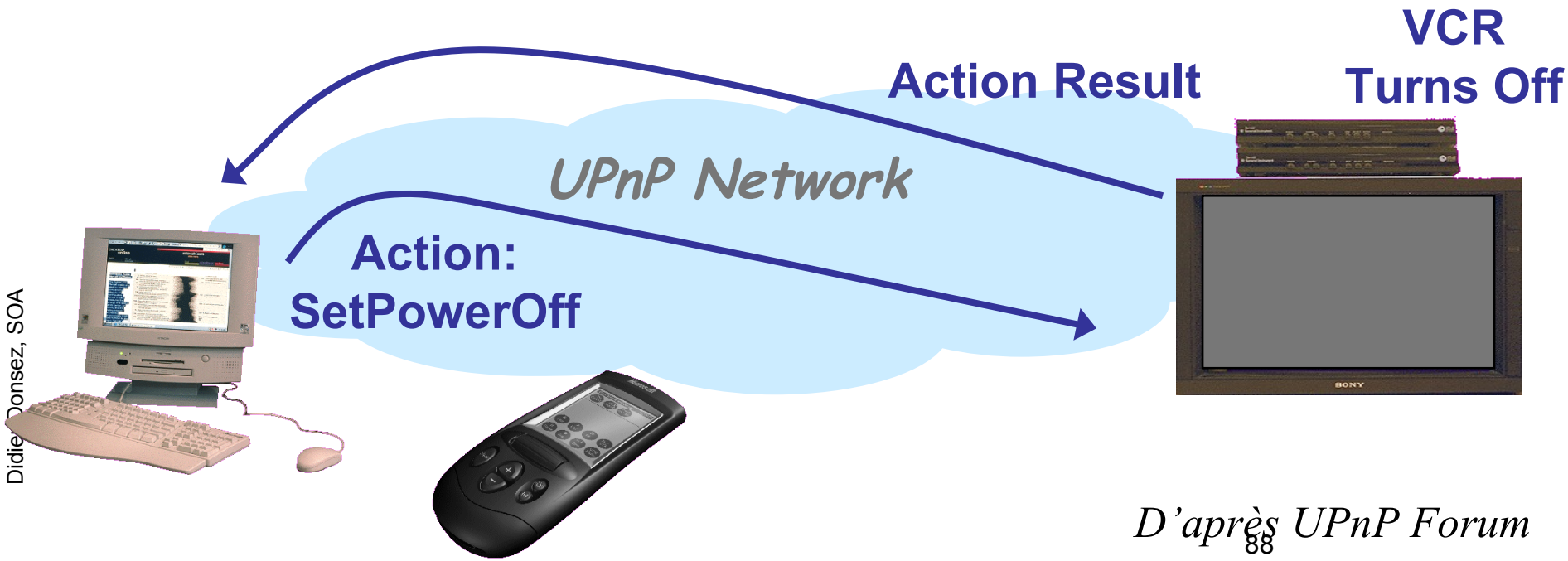
2 Description: XML



D'après UPnP Forum

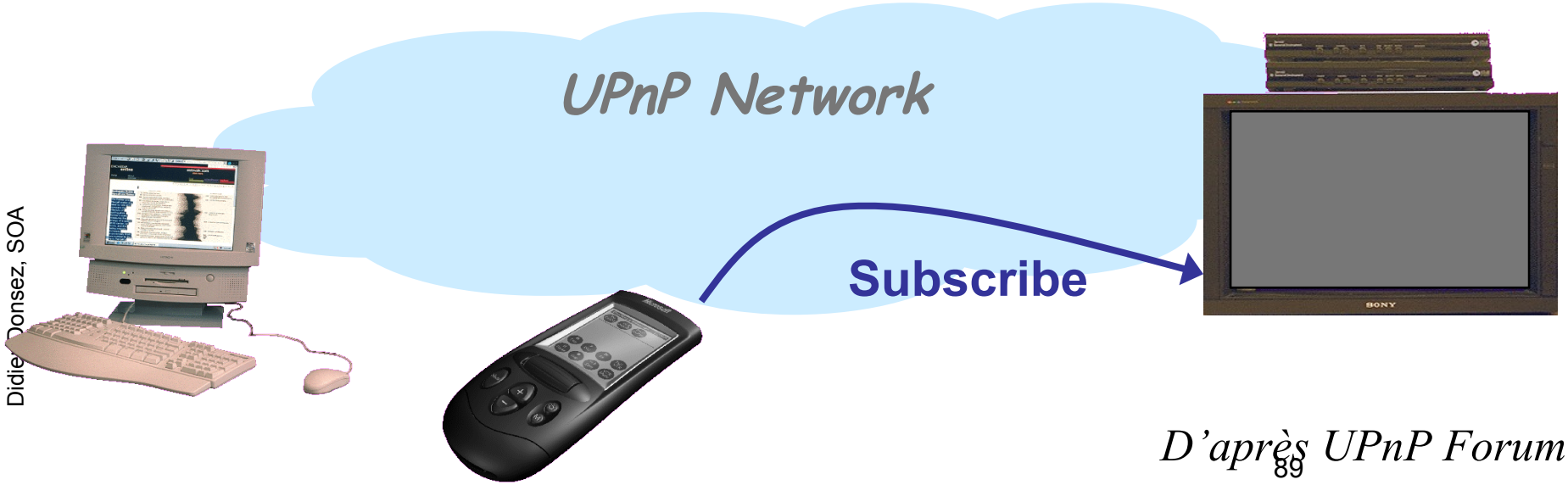
3 Control: SOAP





D'après UPnP Forum

4 Eventing: GENA



D'après UPnP Forum

4 Eventing: GENA

Action: SetPowerOn

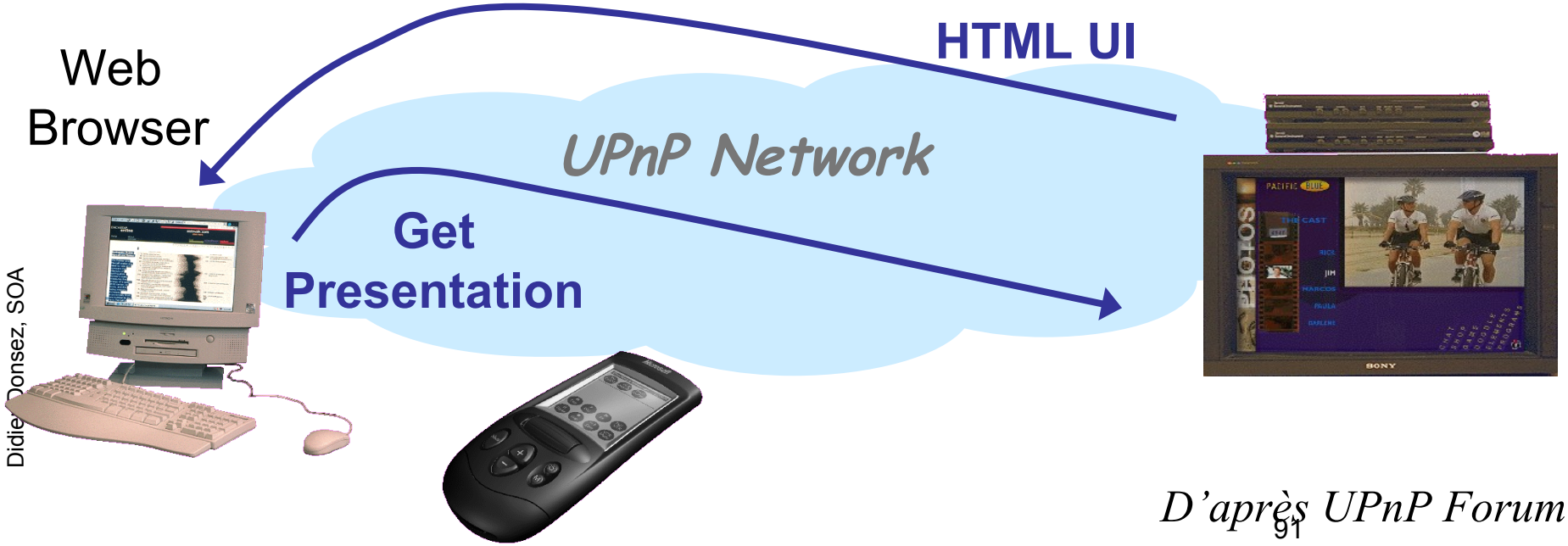
VCR Turns On

UPnP Network

Notify: PowerOn = True

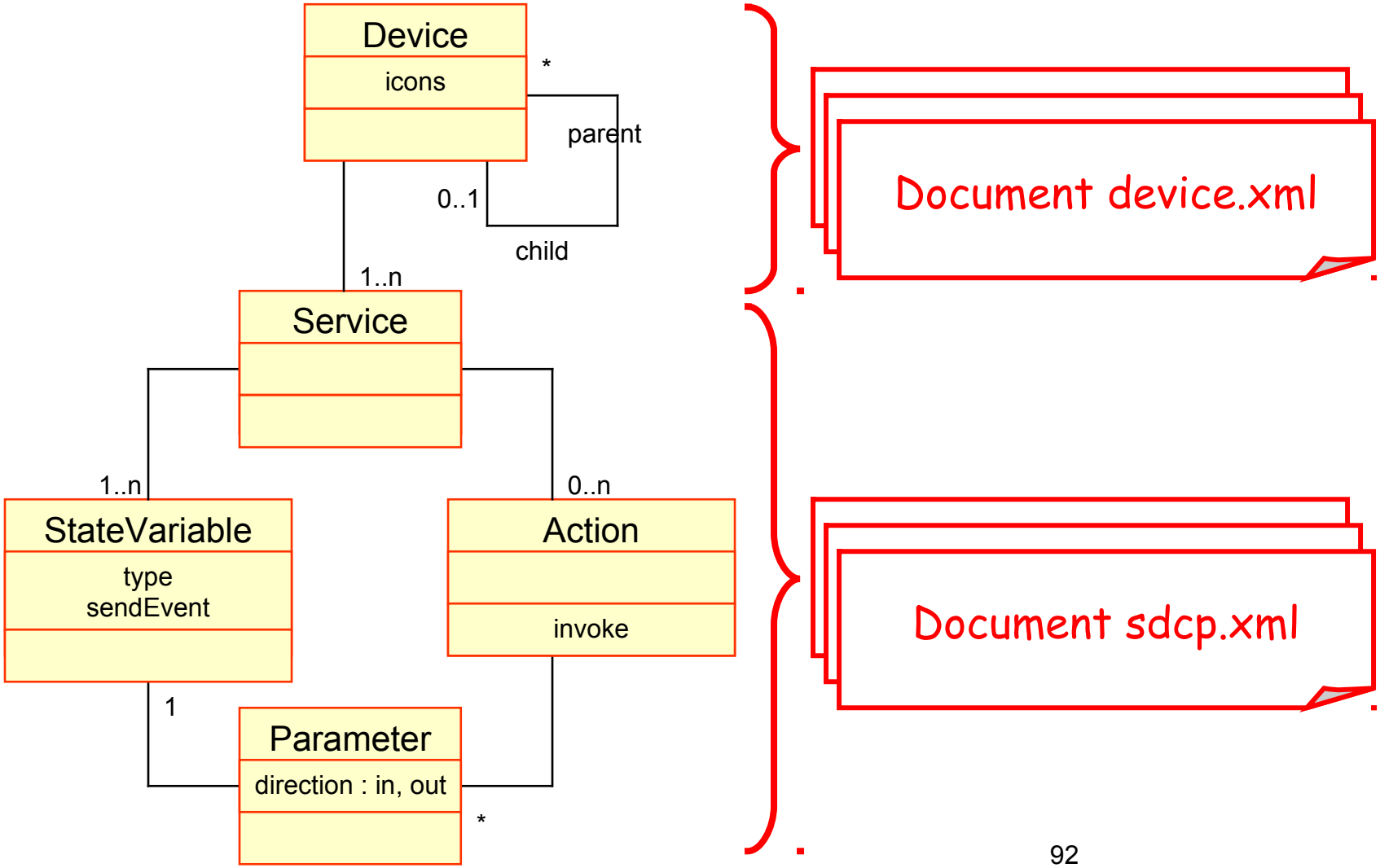
D'après UPnP Forum

5 Presentation: HTML

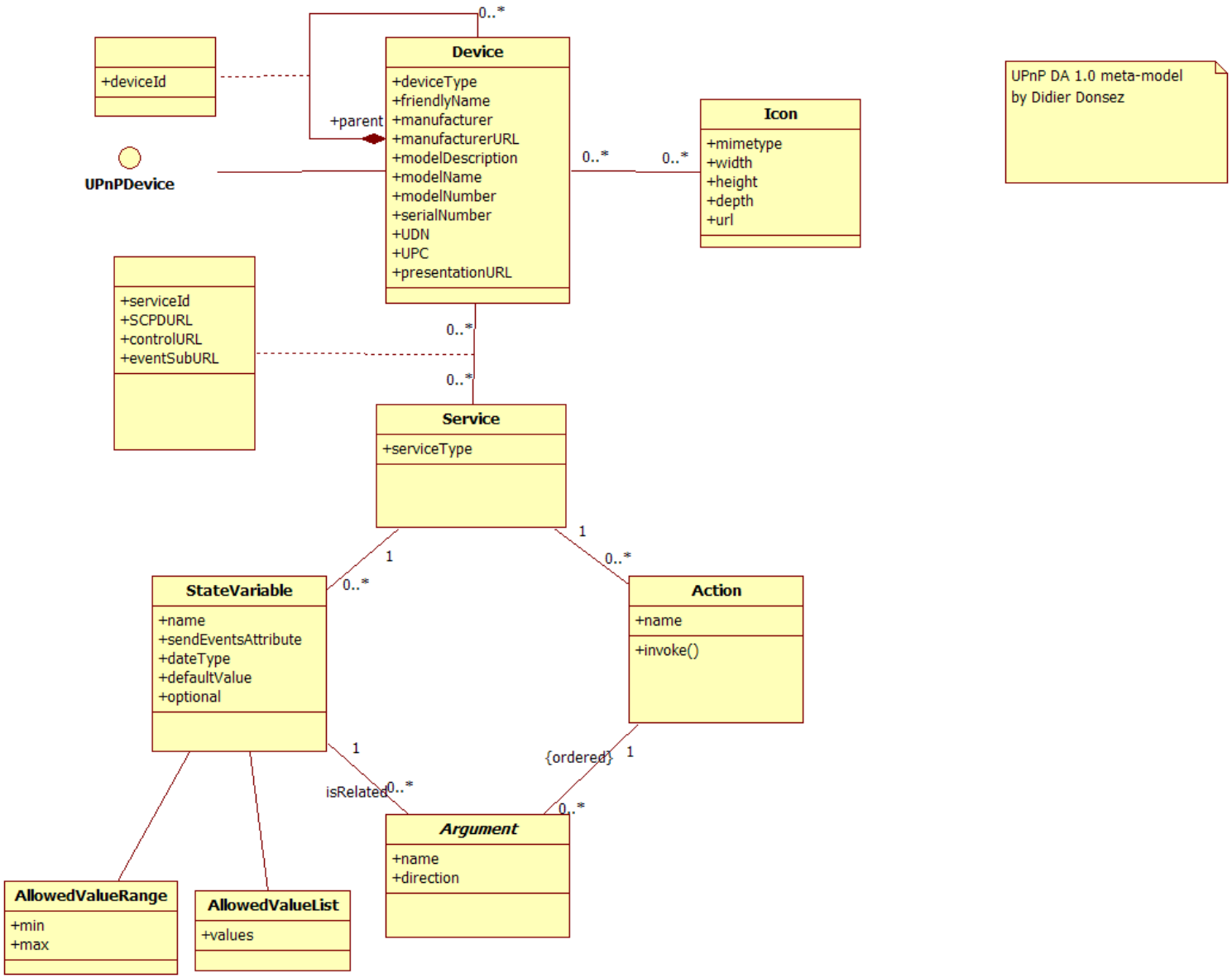


D'après UPnP Forum

Service Model (Simplified)



Service Model (Full)



Example
Lighting Control

- Device *BinaryLight*
  - 1 service
    - *SwitchPower*
- Device *DimmableLight*
  - 1 service
    - *SwitchPower*
    - *DimmingService*

Example
Lighting Control

- Homework
  - Design an additional device (and service) for RGB Leds

Explorateur Réseau Intel pour Technologie UPnP

Fichiers Visualization Aide

- UPnP Devices
  - Intel Media Renderer (PABLO)
  - Intel MicroLight (PABLO)
  - Intel's Media Server (PABLO)
  - Lumière (PABLO)
  - Lumière (RUBY)
    - um:schemas-upnp-org:service:DimmingServic
      - State variables
        - GetLoadLevelStatus(ui1 RetLoadLevelSt
        - GetMinLevel(ui1 MinLevel)
        - SetLoadLevelTarget(ui1 NewLoadLevelT
    - um:schemas-upnp-org:service:SwitchPower:1
      - State variables
        - Status
        - Target
      - GetStatus(boolean ResultStatus)
      - SetTarget(boolean newTargetValue)

Example
Lighting Control

- Service *SwitchPower*
  - 2 state variables
    - Target
    - Status (observable !)
  - 3 actions
    - SetTarget / GetTarget
    - GetStatus (no setter)

Explication:

La mise sous/hors tension d'un équipement électrique peut n'être effective que plusieurs secondes après l'invocation du *SetTarget*

Quand la mise sous/hors tension est effective, la variable Status notifie son chg d'état

- Service *DimmingService*
  - 3 state variables
    - LoadLevelStatus, LoadLevelTarget, MinLevel

Explorateur Réseau Intel pour Technologie UPnP

Fichiers Visualization Aide

- UPnP Devices
  - Intel Media Renderer (PABLO)
  - Intel MicroLight (PABLO)
  - Intel's Media Server (PABLO)
  - Lumière (PABLO)
  - Lumière (RUBY)
    - um:schemas-upnp-org:service:DimmingServic
      - State variables
      - GetLoadLevelStatus(ui1 RetLoadLevelSt:
      - GetMinLevel(ui1 MinLevel)
      - SetLoadLevelTarget(ui1 NewLoadLevelT
    - um:schemas-upnp-org:service:SwitchPower:1
      - State variables
        - Status
        - Target
      - GetStatus(boolean ResultStatus)
      - SetTarget(boolean newTargetValue)

Explorateur Réseau Intel pour Technologie UPnP

Fichiers Visualization Aide

- UPnP Devices
  - Intel Media Renderer (PABLO)
  - Intel MicroLight (PABLO)
  - Intel's Media Server (PABLO)
  - Lumière (PABLO)
  - Lumière (RUBY)
    - um:schemas-upnp-org:service:DimmingServic
      - State variables
        - GetLoadLevelStatus(ui1 RetLoadLevelSt
        - GetMinLevel(ui1 MinLevel)
        - SetLoadLevelTarget(ui1 NewLoadLevelT
    - um:schemas-upnp-org:service:SwitchPower:1
      - State variables
        - Status
        - Target
      - GetStatus(boolean ResultStatus)
      - SetTarget(boolean newTargetValue)

Standard name for standard services

Explorateur Réseau Intel pour Technologie UPnP

Fichiers Visualization Aide

- UPnP Devices
  - Intel Media Renderer (PABLO)
  - Intel MicroLight (PABLO)
  - Intel's Media Server (PABLO)
  - Lumière (PABLO)
  - Lumière (RUBY)
    - um:schemas-upnp-org:service:DimmingServic
      - State variables
        - GetLoadLevelStatus(ui1 RetLoadLevelSt
        - GetMinLevel(ui1 MinLevel)
        - SetLoadLevelTarget(ui1 NewLoadLevelT
    - um:schemas-upnp-org:service:SwitchPower:1
      - State variables
        - Status
        - Target
      - GetStatus(boolean ResultStatus)
      - SetTarget(boolean newTargetValue)

Only 24 primitif types :

boolean
ui1, ui2, i1, i2, i4, int, ui4
r4, float, r8, number,
fixed.14.4
char, string
uri, uuid
time, time.tz, date,
dateTime, dateTime.tz
bin.base64, bin.hex

No array
No struct

State variable notifiying its value changes

Explorateur Réseau Intel pour Technologie UPnP

Fichiers Visualization Aide

- UPnP Devices
  - Intel Media Renderer (PABLO)
  - Intel MicroLight (PABLO)
  - Intel's Media Server (PABLO)
  - Lumière (PABLO)
  - Lumière (RUBY)
    - um:schemas-upnp-org:service:DimmingServic
      - State variables
        - GetLoadLevelStatus(ui1 RetLoadLevelSt
        - GetMinLevel(ui1 MinLevel)
        - SetLoadLevelTarget(ui1 NewLoadLevelT
    - um:schemas-upnp-org:service:SwitchPower:1
      - State variables
        - Status
        - Target
      - GetStatus(boolean ResultStatus)
      - SetTarget(boolean newTargetValue)

UPnP Homework

- Install Kody Media Server (and activate DLNA/UPnP)
- Discover UPnP devices and associated services with Node UPnP Client
  - https://www.npmjs.com/package/node-upnp-client

- Desig,n a UPnP node for Node-RED platform
  - Choose the configurable properties
  - Choose the payload
- Implement it (optional)
- Try it (optional)
- Contribute it (optional)

OSGi : SOA for Java Programming

Why the Java/OSGi stack are promising ?

- Why Java ?

  It is object Oriented

  It has a huge standard API

  It is dynamic:

  New objects can be used at run-time

  New classes can be defined at run-time

  It is secured

  Types are known at compile time and at run-time

  A security Manager can control access to sensible methods

- Why OSGi ?

  It is component Based

  It has service-oriented programming features

  => It is simplifying Java dynamicity integration

  It defines standard services

OSGi Main Concepts

Framework:

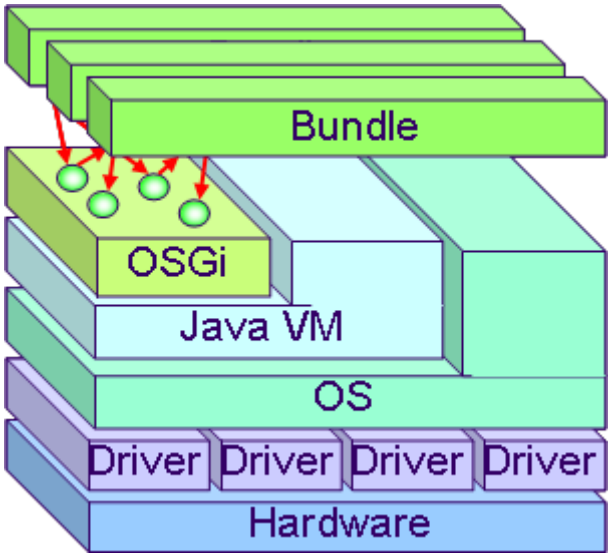
- Bundles execution environment
  - Felix (Apache), Equinox (Eclipse), ...
- Event notification

Bundles:

- Services diffusion and deployment unit

Services:

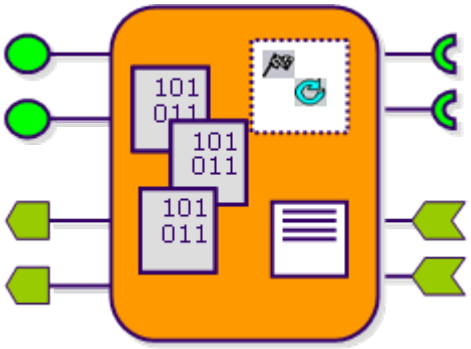
- Java Object implementing a well define contract



Middleware and Application Packaging

Modularize the middleware/application

- Distribute the different middleware services
- Better component visibility
- Need of a deployment container
- Partial update without restart all



Implementation

- Based on Jarfile and Manifest entries
- Explicit Package dependencies and Versioning (range)

De-facto standard for modular Java applications

- Eclipse platforms, JavaEE middlewares (Glassfish, Jboss, ...)
- Overtake JNLP(JSR-56), JavaEE EAR …
  and Java 1.9 Jigsaw Module System

Vices and vertues

Vertues

- Lightweight
- Clean Packaging
  - Avoid CLASSPATH Hell
- Dynamicity at run-time
- Non-stop VM (long-live Java applications)

Vices

- Programatic approach
  - Event programming is painfull
- No non-functionnal services
- Centralized approach

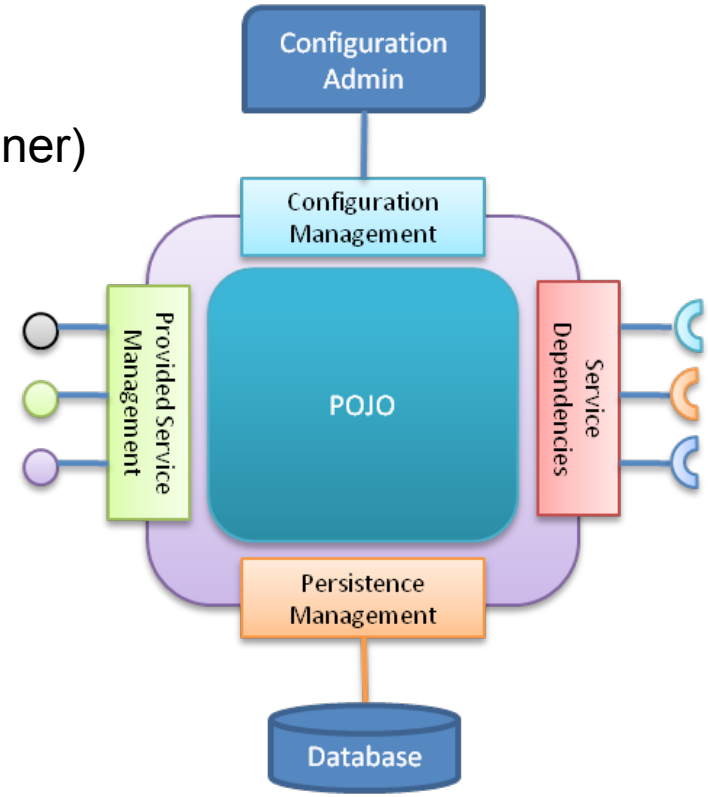
Service-Oriented Components for OSGi Development

- Service Binder
- Service Component Runtime (aka scr)
- Dependency Manager
- Blueprint
- iPOJO

- More CM
  - iPOPO (iPOJO like CM for Python programs), uBu (javascript, ...

Apache Felix iPOJO

- A service-oriented component model
  - Supporting structural compositions
    - Hierarchical
  - Built applications are natively dynamic
  - Extensible (implemented with an open container)
- Key concepts
  - Service implementations and instances
  - A service specification model
  - A service dependency model
  - Service context



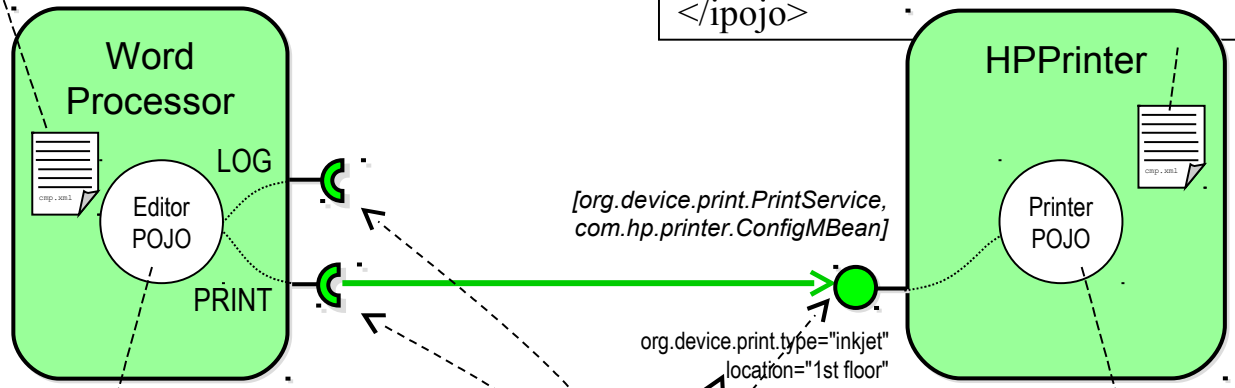
iPOJO example

/metadata.xml

```
<ipojo>
  <instance component="EditorType"
    name="Editor"/>
</ipojo>
```

/metadata.xml

```
<ipojo>
  <instance component="HPPrinterType"
    name="HPPrinter">
  <property name="location" value="1st floor"/>
  <property name="org.device.print.type" value="inkjet"/>
  <property name="jmx.objectName"
    value="printer.hp:name=DeskJet 930c 1st fl."/>
  </instance>
</ipojo>
```



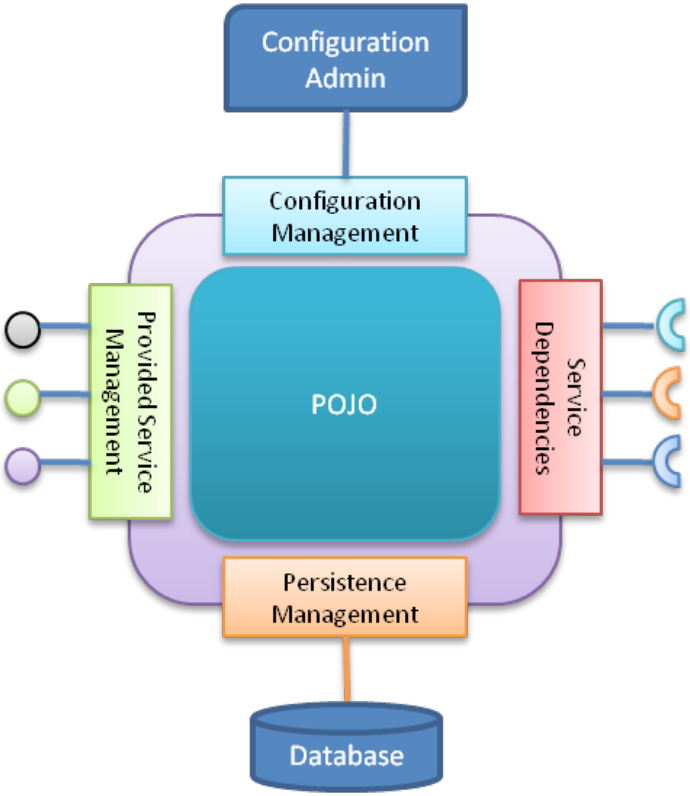

```
@Component(name="EditorType", immediate=true)
public class EditorComp {
   @Requires(filter="(&(location=*)(org.device.print.type=inkjet))")
    private List<PrintService> printServices;
   @Requires()
    private LogService logService;

// business methods
…
}
```

```
@Component(name="HPPrinterType", immediate=true)
@Provides()
public class PrinterComp
  implements PrintService, ConfigMBean {

   @ServiceProperty(name="location")
   private String position;

   // méthodes de PrintService
…
}
```

iPOJO Handlers

- Provide
- Service Requirement
  - Method injection and Field injection
  - Nullable object, Default implementation
- Temporal Service Requirement
- Lifecycle callback
- Config
- JMX
- Event
- Extender Pattern
- Whiteboard Pattern

- Custom ones …



iPOJO Architecture handler

- Disable architecture
  - <component classname="fr.imag.adele.escoffier.hello.impl.HelloServiceImpl" architecture="false">
- Command
  - ipojo:factories - Display iPOJO factories
  - ipojo:factory - Display the information about a specific factory
  - ipojo:handlers - Display iPOJO handlers
  - ipojo:instance - Display the architecture of a specific instance
  - ipojo:instances - Display iPOJO instances

iPOJO Configuration Handler

- Add a CM ManagedService

```
<iPOJO>
<Component className="system.impl.DiskServiceImpl">
  <Provides>
    <Property name="quota" field="m_quota"/>
  </Provides>
  <Properties propagation="true"
            updated="afterReconfiguration"
    />
    <Property name="quota" field="m_quota"/>
    <Property name="threshold" method="updateThresholdArray"/>
    <Property name="disk.name" type="java.lang.String"/>
  </Properties>
</Component>
<instance component="sysrtem.impl.DiskServiceImpl" name="DiskService">
  <property name="quota" value="100"/>
  <property name="threshold" value="\{10, 20, 30}"/>
  <property name="disk.name" value="D"/>
  <property name="managed.service.pid" value="com.acme.system.disk.D"/>
</instance>
</iPOJO>
```

Updates are propagated to the Service Registration

Invoke the method on update

Static property

Instance level

service.pid

```
class DiskServiceImpl implement DiskService {
  int m_quota;
  public void updateThresholdArray(int[] a) { … }
  public void afterReconfiguration(Dictionary config) { ... }
}
```

iPOJO JMX Handler

- XML

```
<ipojo xmlns:jmx="org.apache.felix.ipojo.handlers.jmx">
      ...
<jmx:config>
    <jmx:property name="quota" field="m_quota"
      rights="r" notification="true"/>
    <jmx:method name="setQuota"/>
  </jmx:config>
</ipojo>
```

- @nnotations (not JSR 255 JMX annotations)

```
@Component
@Config(domain="my-domain", usesMOSGi=false)
public class Disk {
    @Property(name="quota", notification=true, rights="r", description="Disk quota")
   private String m_quota;
    @Method(description="set the quota") // Method published in the MBean
    public void setQuota(int q) {
        if(q>0) m_quota = q;
    }
}
```

iPOJO Event Handler Example

- Publication

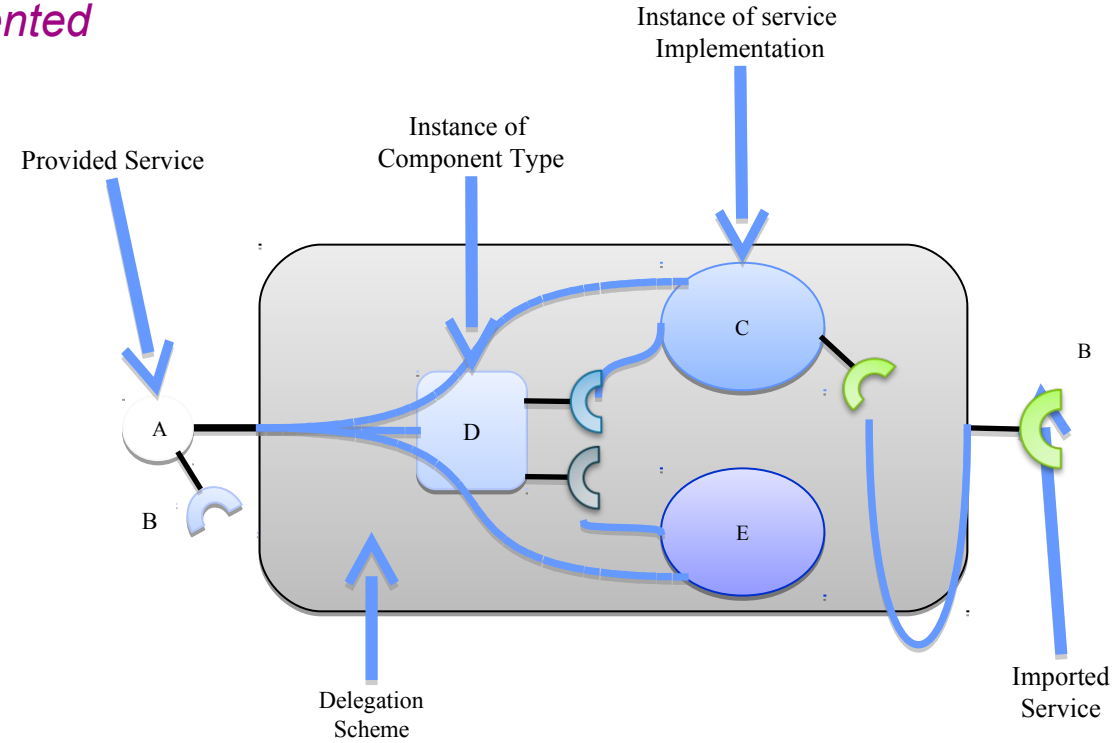
```
@org.apache.felix.ipojo.handlers.event.Publisher(name="p3", synchronous=true, topics="bar")
org.apache.felix.ipojo.handlers.event.publisher.Publisher publisher3;

public void doSomething() {
        Dictionary d = new Properties();
        // Fill out the even
            publisher3.send(d); // Send event
    }
```

- Subscription

```
@Subscriber(name="s1", data_key="data")
public void receive1(Object o) { // Nothing }
@Subscriber(name="s2", topics="foo,bar", filter="(data=DIDIER*)")
public void receive2(Event e) { // Nothing }
@Subscriber(name="s3", topics="foo", data_key="data", data_type="java.lang.String")
public void receive3(String s) { // Nothing }
```

iPOJO Composite

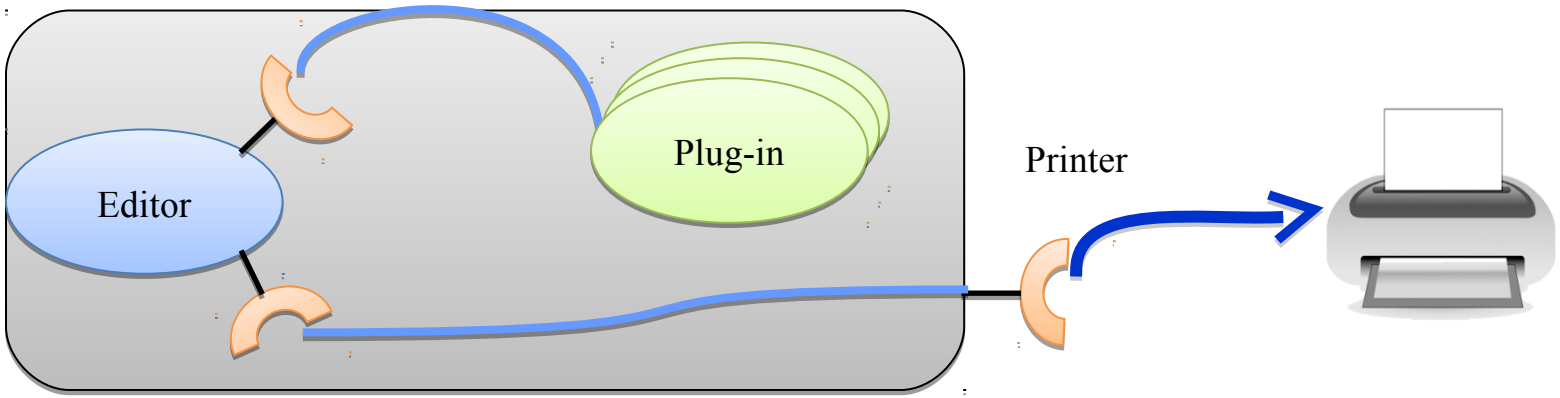
- Architecture Description Language for
  - Required Service Specifications
    - Instantiated and Imported
  - Provided Service Specifications
    - Exported and *Implemented*
  - Component Types



from Clément Escoffier' thesis defense

iPOJO Composite Example

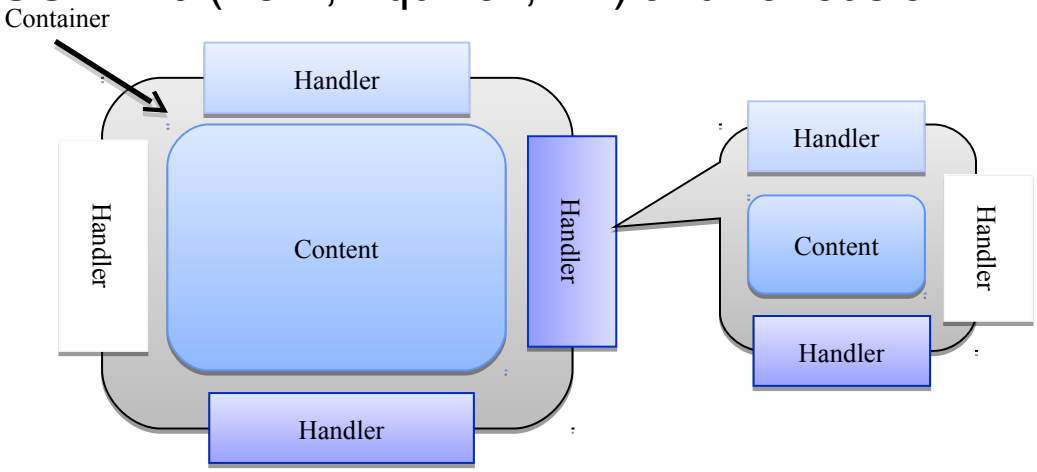
```
<composite name="Editor1">
<subservice action="instantiate"
        specification="…Editor"/>
<subservice action="instantiate"
        specification="… Plugin"  aggregate="true" />
<subservice action="import"
        specification="…Printer"  optional="true"/>
</composite>
```



from Clément Escoffier' thesis defense

iPOJO implementation

- Main features
  - *Bytecode* manipulation (ASM)
  - Extensible through *Handlers*
    - Handlers <u>are</u> iPOJO instances
    - Natively support dynamism
  - Heavy use of threads and synchronization constructions
  - on top of OSGi R4.0 (Felix, Equinox, KF) and various JVM 1.4, 1.5+



from Clément Escoffier' thesis defense

OSGi + UPnP Homework

- Launch Felix
- Install and start UPnP bundles
  - UPnP Base Driver, UPnP Extra, UPnP Tester
    - With OBR
    - From repo
- Build and install device samples
  - https://github.com/apache/felix/tree/trunk/upnp/samples

OSGi Homework

- Install and run Eclipse Smarthome demo
  - https://www.eclipse.org/smarthome
- Launch the OSGi console
  - Inspect bundles
  - Inspect services
  - Listen to events (Event Admin)

REST - Representational State Transfer

- Fielding, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.

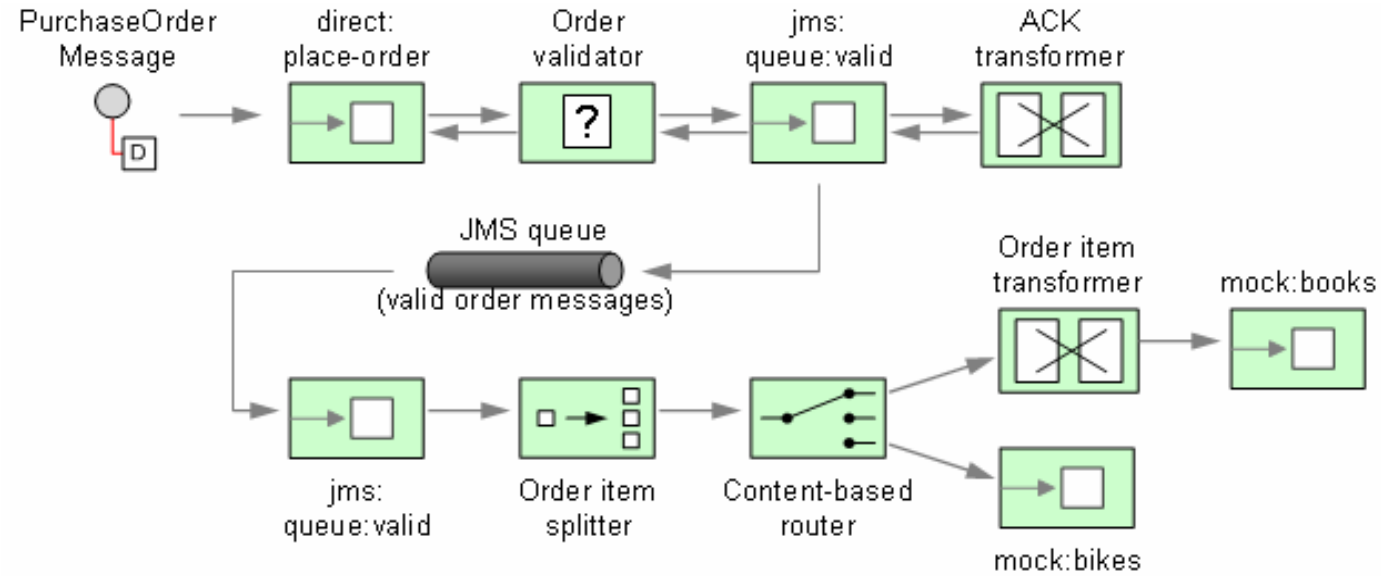
- KIS (Keep It Simple) for RIA, mashups, Web 2.0 …
- Vision CRUD de ce que doit être un service
  - Retour au client-serveur SQL (années 90)
- Richardson Maturity Model
- Web Technologies
  - Transport HTTP/HTTPS
  - Sans état (*stateless*)
    - encore moins transactionnel
  - Requêtes: POST/GET/PUT/DELETE for CRUD
  - Réponses: what you want (XML, HTML, JSON, …) + status code
  - REST conformance : swagger

Event-driven SOA

- Event-driven SOA
  - Loose-Coupling (pub/sub)
  - Event routing
  - Processes are triggered on event reception

- Enterprise Service Bus
  - API : JBI (Java Business Integration (JSR 208, …)
  - Products : Apache Camel, Mule, …

Event-Driven SOA Example with Apache Camel




```
// order placement route (main route)
val placeOrderRoute = validateOrder >=> oneway >=> to("jms:queue:valid") >=> {
  m: Message => m.setBody("order accepted")
}

// order placement route consuming from direct:place-order endpoint (incl. error har
from("direct:place-order") {
  attempt { placeOrderRoute } fallback {
    case e: ValidationException => { m: Message => m.setBody("order validation faile
    case e: Exception           => { m: Message => m.setBody("general processing er
  }
}

// order processing route
from("jms:queue:valid") {
  split { m: Message => for (item <- m.bodyAs[PurchaseOrder].items) yield m.setBody(
    case Message(PurchaseOrderItem(_, "books", _, _), _) => orderItemToTuple >=> to(
    case Message(PurchaseOrderItem(_, "bikes", _, _), _) => to("mock:bikes")
  } >=> { m: Message => println("received order item = %s" format m.body); m }
}
```

Bibliography

- Conferences & Journals
  - ICSOC, SCC, ICWS, IEEE Trans. on SOC