



# Planning for the Future Internet of Services

Paolo Traverso  
Centre for Information Technology - IRST  
Bruno Kessler Foundation (FBK)  
Trento, Italy  
[traverso@fbk.eu](mailto:traverso@fbk.eu)

## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - Modeling, Composition, Monitoring, Adaptation

## PART II: Future Internet

- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, Composition, Monitoring, Adaptation

## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - Modeling, Composition, Monitoring, Adaptation

## PART II: Future Internet

- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, Composition, Monitoring, Adaptation

# Services ...

- Services

- business model from **products** to **services** ..
- services are used, they are not owned





# Software Services ...



*The key to a new generation of software systems*

# Software Services ...

---

- Services
  - business model from products to services ...
  - services are used, they are not owned
- Software services
  - software components that can be used ...
  - ... but are not owned
- Service-oriented applications
  - constructed by **composing and configuring** software services...
  - ... most often provided by "third parties"
  - ... software that is **not under control**

## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - **Modeling**, Composition, Monitoring, Adaptation

## PART II: Future Internet

- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, Composition, Monitoring, Adaptation

# Software Services ...

---

- Services
  - business model from products to services ...
  - services are used, they are not owned
- Software services
  - software components that can be used ...
  - ... but are not owned
- Service-oriented applications
  - constructed by **composing and configuring** software services...
  - ... most often provided by "third parties"
  - ... software that is **not under control**



# Modeling: Service Level Agreements

---

- Service Level Agreement (SLA)
  - formal negotiated agreement between two parties. ...
  - contract between customers and their service provider, or between service providers
  - it records the common understanding about services, priorities, responsibilities, guarantee, etc.
  - For example, it may specify
    - the levels of availability, and performance ...
    - ... and even penalties in the case of violation of the SLA.

# Modeling: SLAs

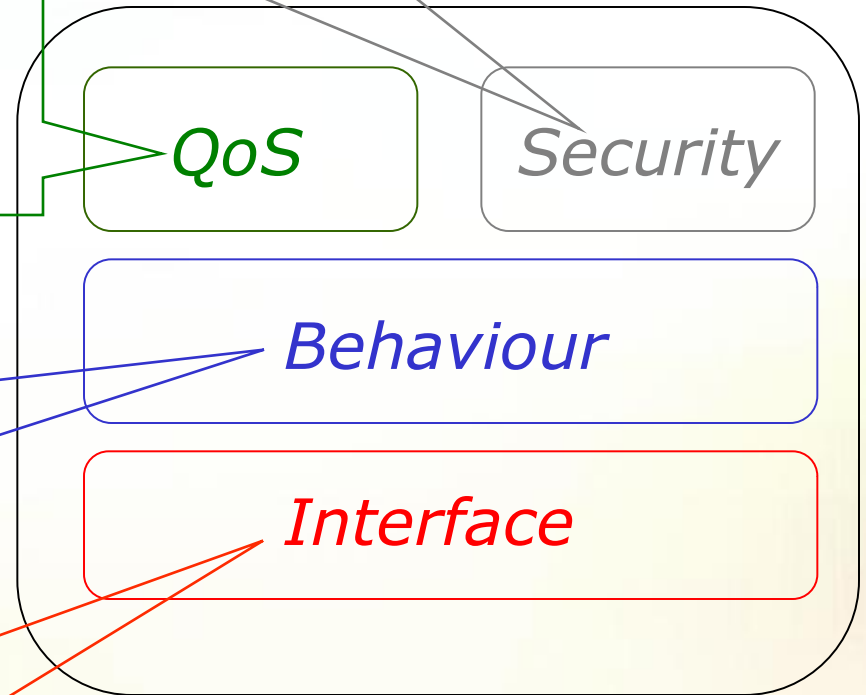
a description of the access/authorization permissions (e.g., in *SAML*)

a description of the non-functional properties of the service (e.g., with *WS-Agreement*)

a description of the interaction flow that is required to interact with the service (e.g., in *BPEL*)

a description of messages and data that are exchanged (e.g., in *WSDL*)

**SLA**



# Example: Interface in WSDL

```
<wsdl:operation name="makeCall">
  <wsdl:input
    message= "parlayx_third_party_call:
              ThirdPartyCall_makeCallRequest" />
  <wsdl:output
    message="parlayx_third_party_call:
             ThirdPartyCall_makeCallResponse" />
  <wsdl:fault name="ServiceException"
    message="parlayx_common_faults:
             ServiceException" />
  <wsdl:fault name="PolicyException"
    message="parlayx_common_faults:
             PolicyException" />
</wsdl:operation>
```

# Example: Interface in WSDL

```
<wsdl:operation name="makeCall">
  <wsdl:input
    message="parlayx_thirdPartyRequest"
    type="tns:ThirdPartyRequest"
  >
  <wsdl:output
    message="parlayx_thirdPartyResponse"
    type="tns:ThirdPartyResponse"
  >
  <wsdl:fault
    name="parlayx_fault"
    type="tns:Fault"
  >
  <wsdl:fault
    name="parlayx_fault"
    type="tns:Fault"
  >
  </wsdl:operation>
```

**Intrinsic uncertainty,  
nondeterminism**

# Service Level Agreements (SLA)

a description of the access/authorization permissions (e.g., in *SAML*)

**SLA**

a description of the non-functional properties of the service (e.g., with *WS-Agreement*)

*QoS*

*Security*

a description of the interaction flow that is required to interact with the service (e.g., in *BPEL*)

*Behaviour*

*Interface*

a description of messages and data that are exchanged (e.g., in *WSDL*)

# Example: Behaviour SLA

*The interaction flow for a phone conference service (e.g. in BPEL)*



*request* →  
*enter* ←  
*check call status* ←  
*on going* →  
*terminated* →  
*send sms* ←

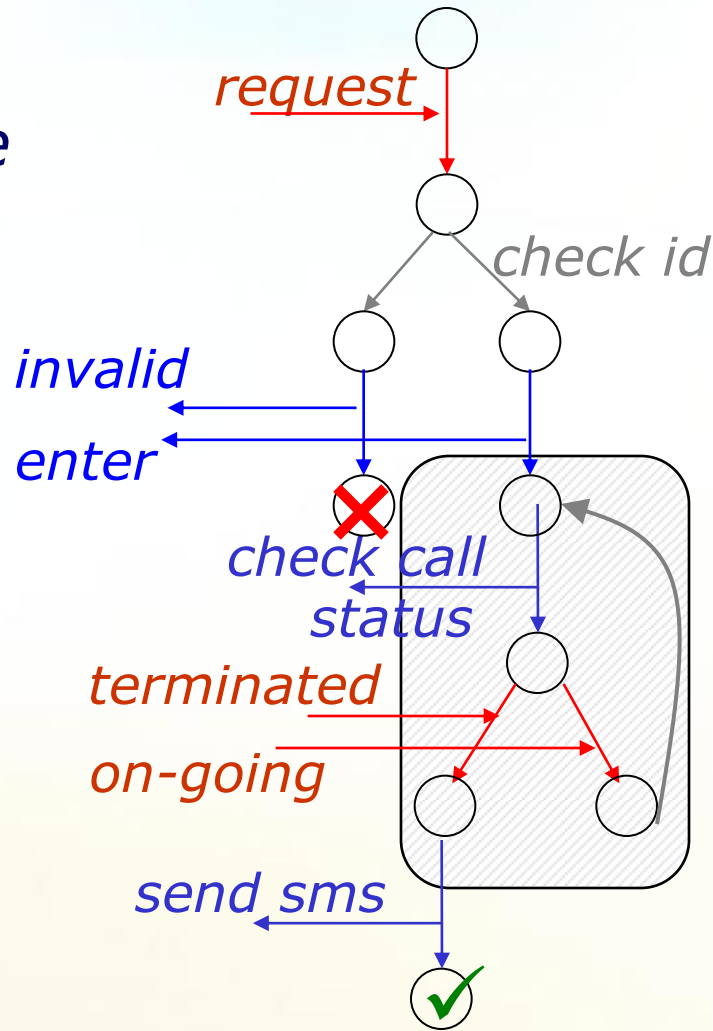
PHONE  
CONFERENCE  
SERVICE





# Example: Behaviour SLA

*The interaction flow for a phone conference service (e.g. in BPEL)*

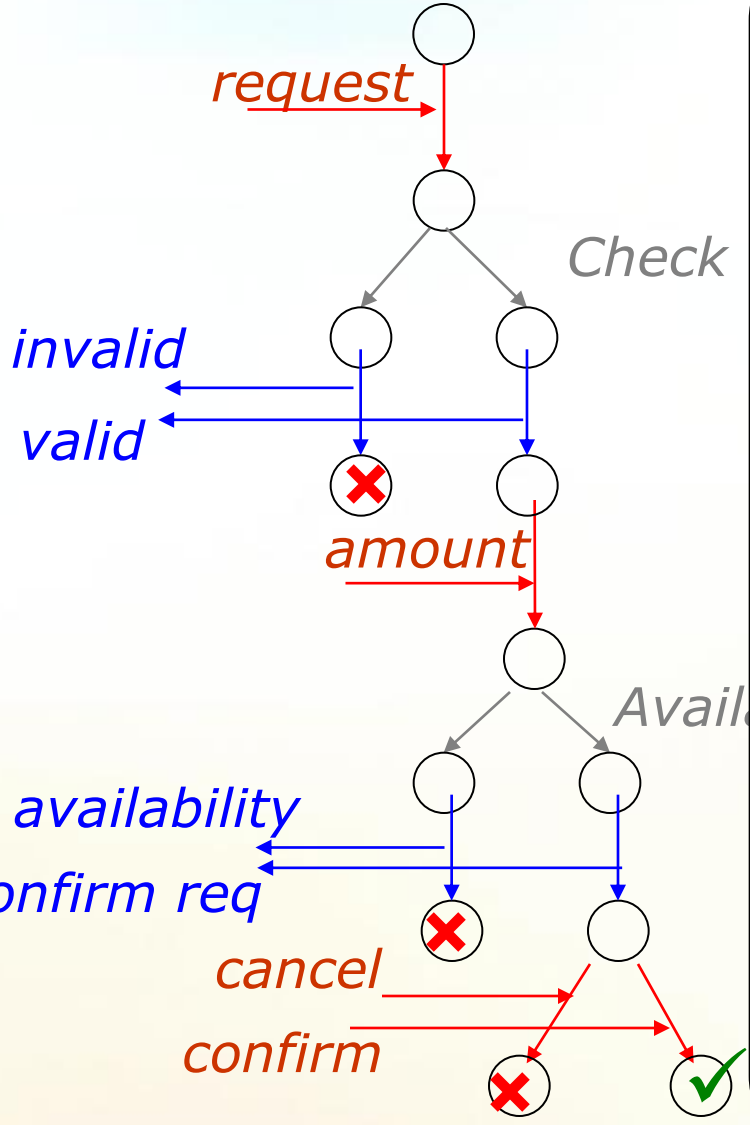


**PHONE  
CONFERENCE  
SERVICE**



# Example: Behaviour SLA

The interaction flow for an on-line bank payment (e.g. in BPEL)



**BANK ON-LINE SERVICE**

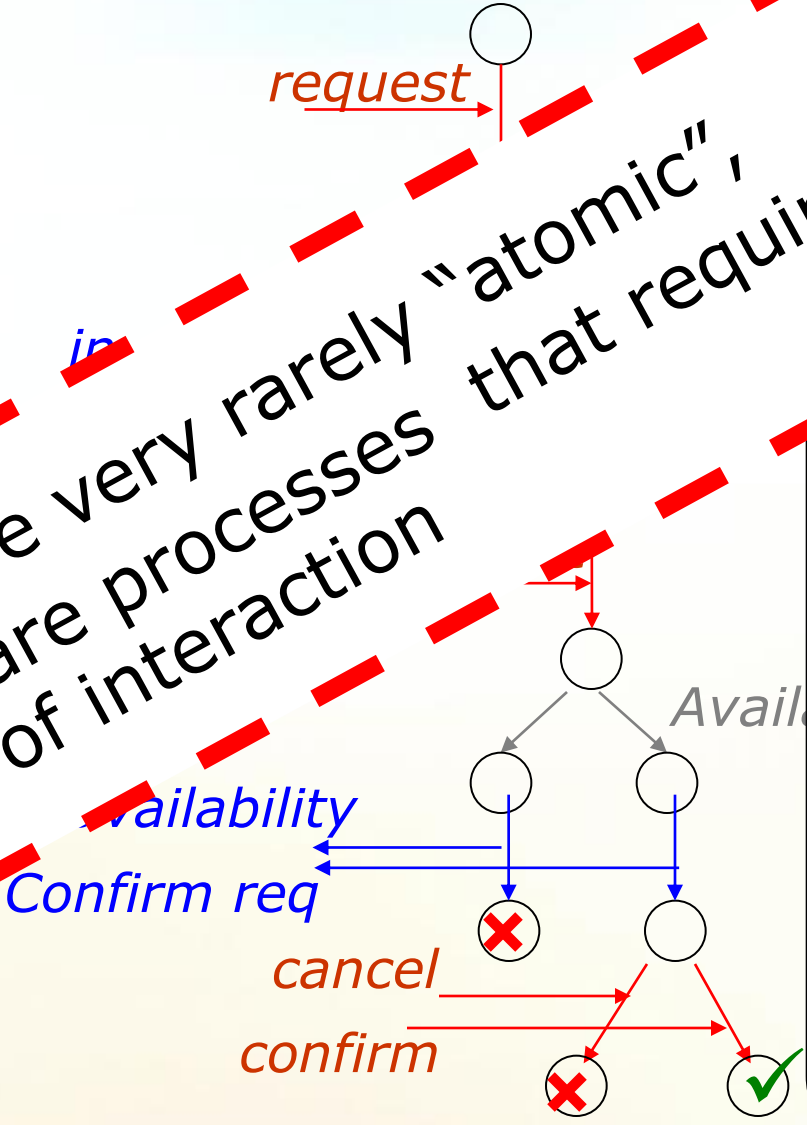


# Example: Behaviour SLA

The interaction flow for an on-line bank payment (e.g. in BPEL)

services are very rarely "atomic", services are processes a flow of interaction

BANK ON-LINE SERVICE



## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - Modeling, **Composition**, Monitoring, Adaptation

## PART II: Future Internet

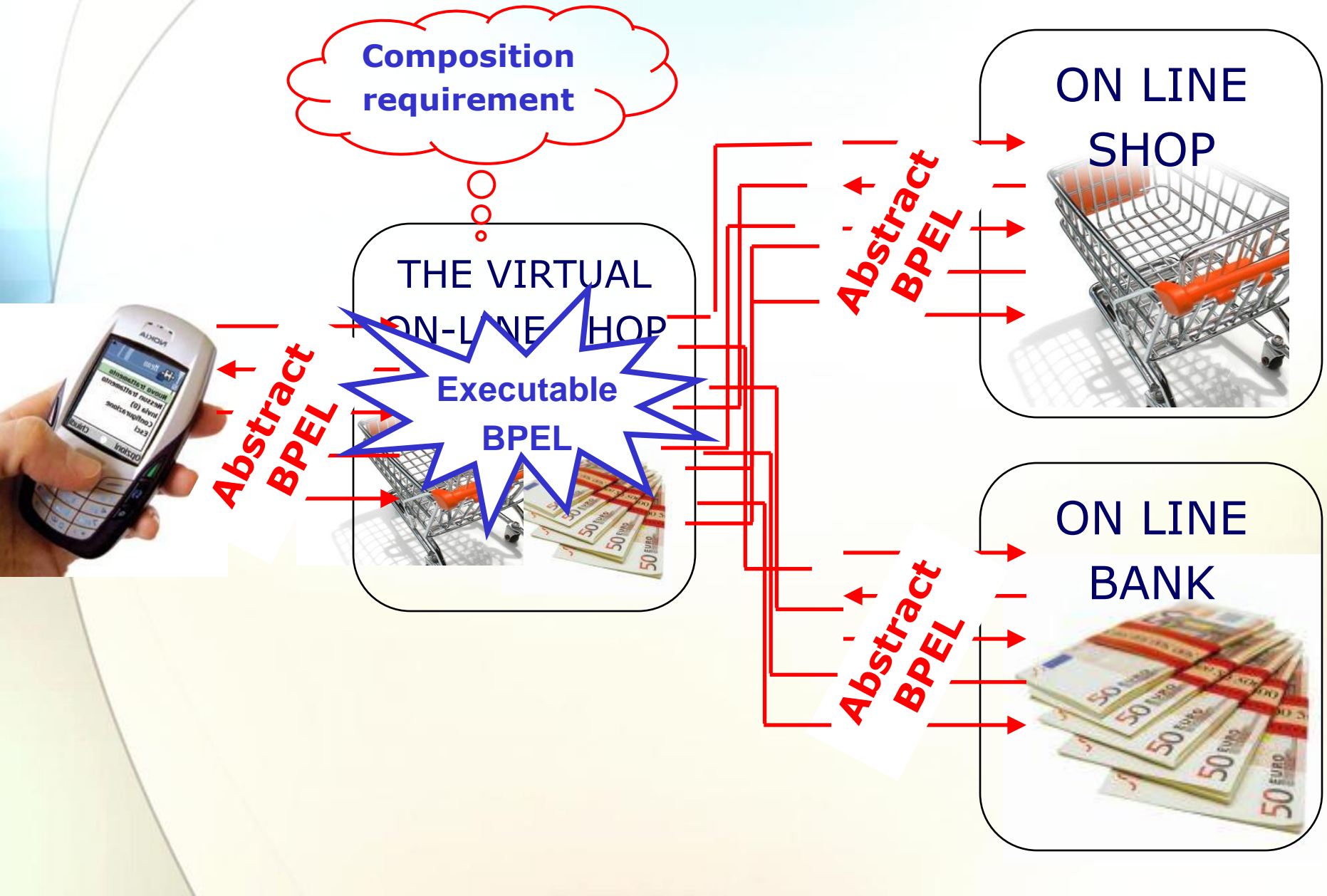
- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, Composition, Monitoring, Adaptation

# The Automated Composition Problem



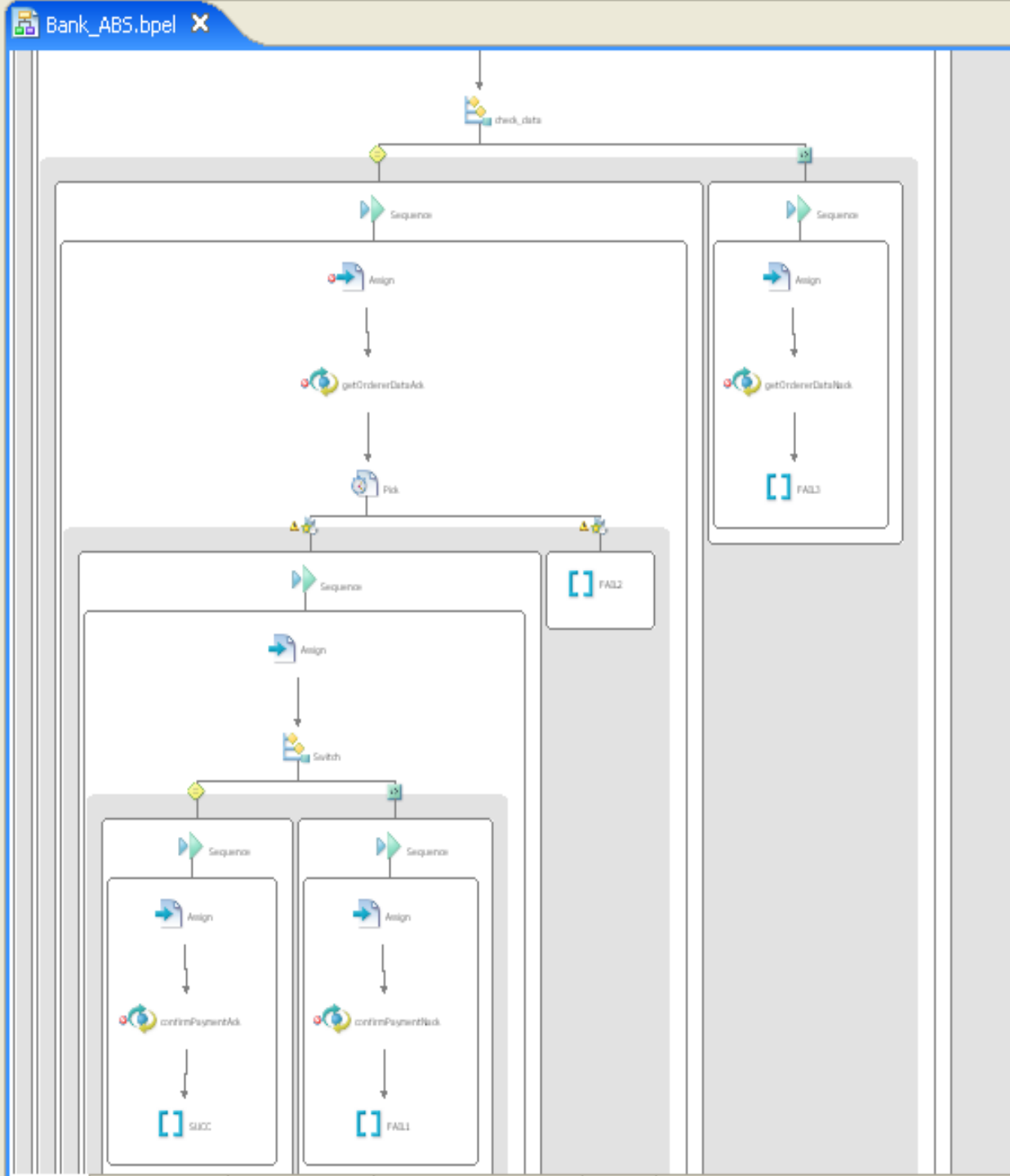


# The Automated Composition Problem





# Abstract BPEL



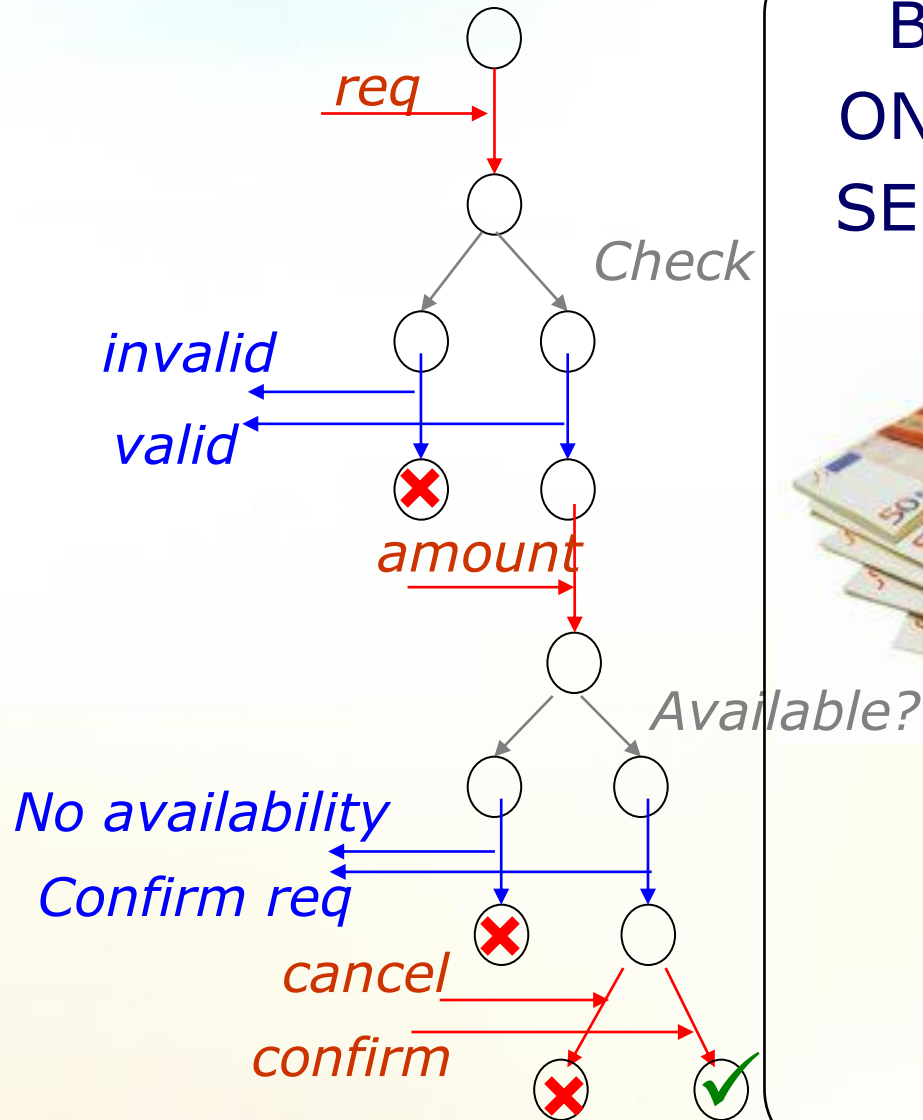
# From Abstract BPEL to State Transition Systems

Abstract BPEL ->  
State Transition  
Systems

*Input actions I*  
(reception of  
messages)

*Output actions O*  
(message sent)

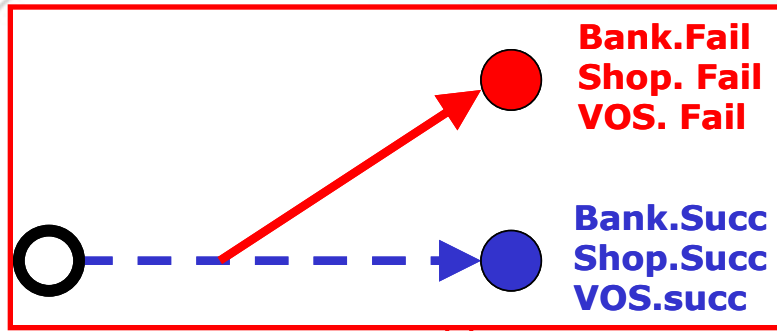
*Internal action  $\tau$*   
(internal  
evolutions that  
are not visible to  
external  
services)



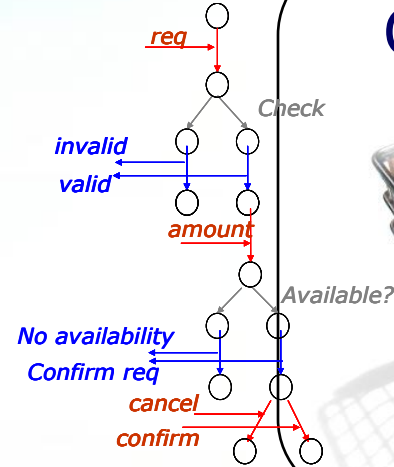
**BANK  
ON-LINE  
SERVICE**



# The Automated Composition Problem



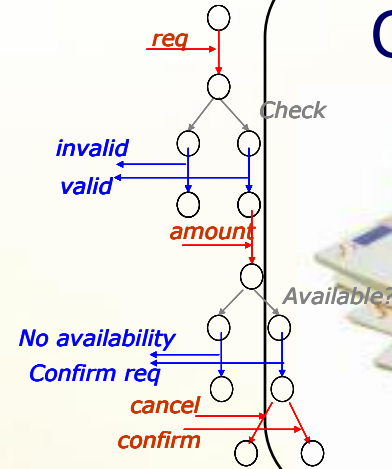
ON LINE SHOP



THE VIRTUAL ON-LINE SHOP COMPOSED SERVICE



ON LINE BANK



# The Composition Algorithm: Intuitions

- The **Parallel Product** of the State Transitions Systems (STSs) of Available Interaction Flows (Components + Composed)
- Search the Product STS to satisfy the Composition Requirement
- Find a **subgraph of the Product** STS which satisfies the following conditions (example with **reachability conditions**):

1. All terminal states satisfy the condition
2. If a state belongs to the subgraph, then

**a. one outgoing input**

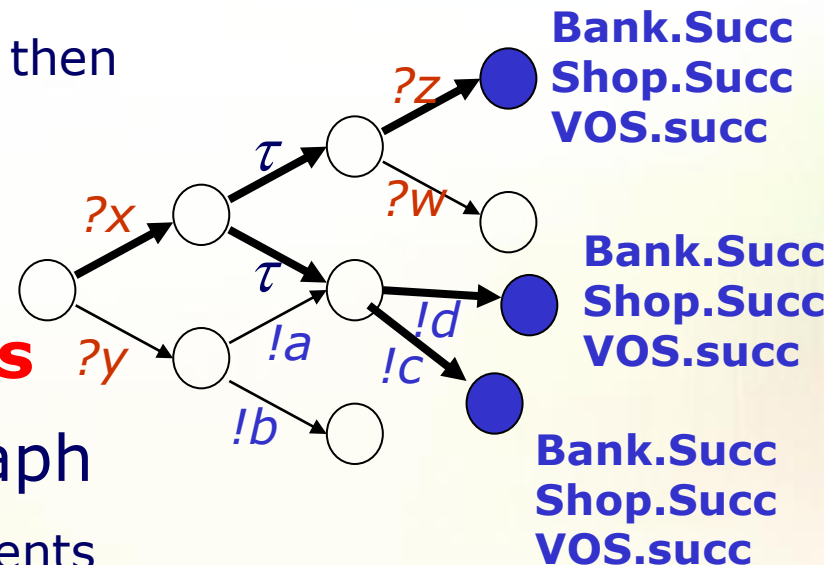
**b. all outgoing taus**

**c. all outgoing outputs**

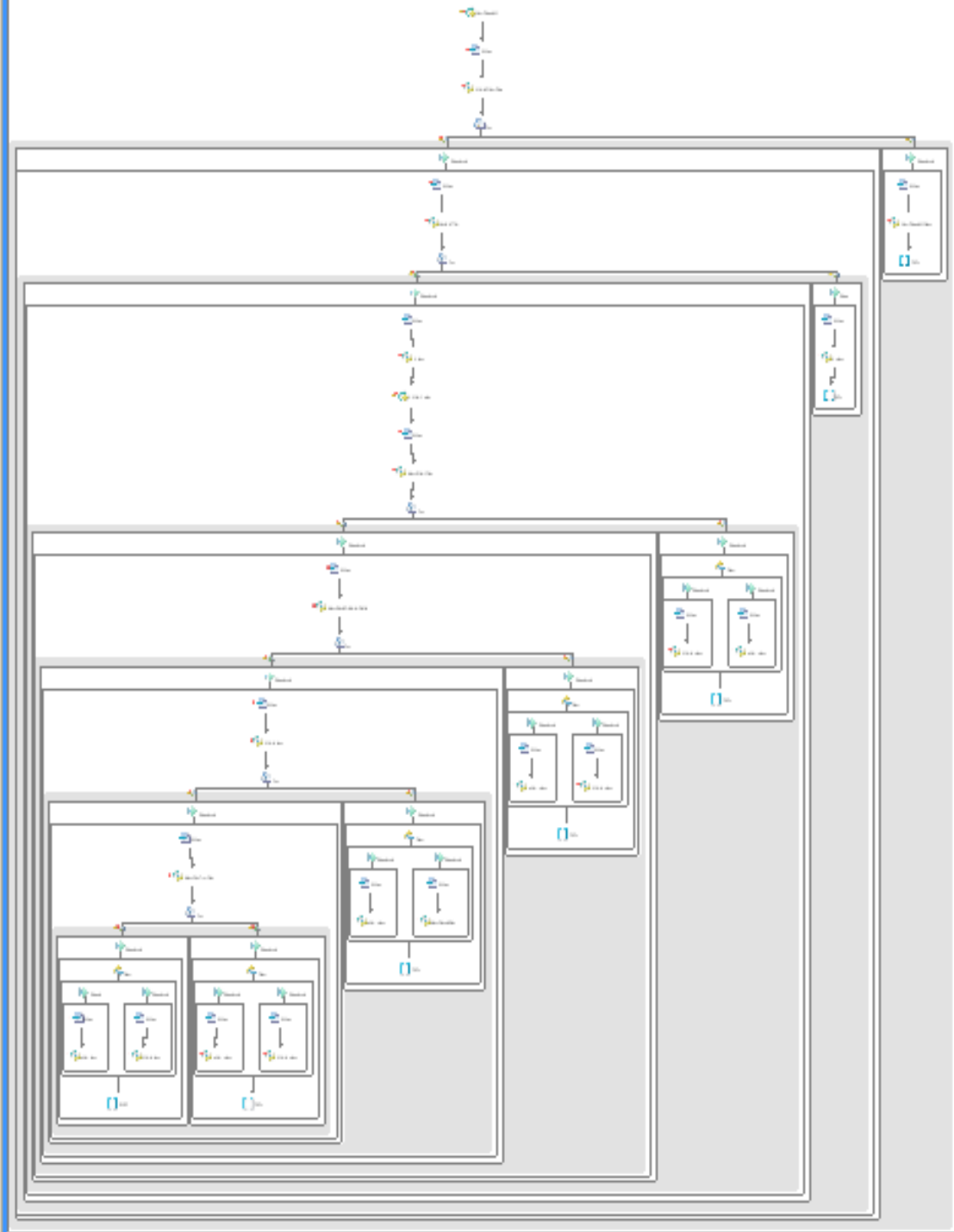
belong to the subgraph

3. remove non deadlock-free components

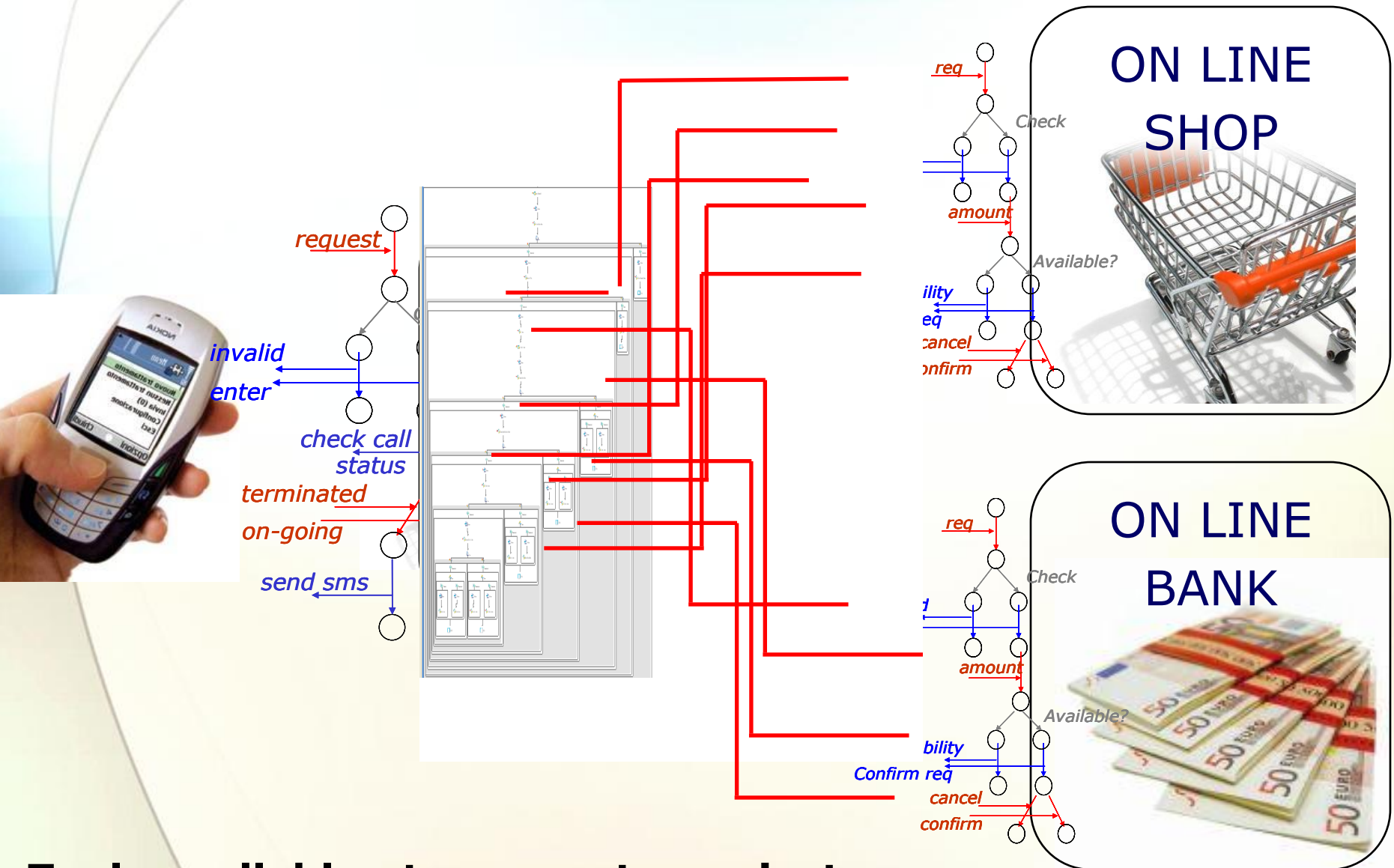
- Product STSs can be extremely large: we use **BDD-based exploration primitives** from the “**Planning as Model Checking**” framework



# The Automatically C



# Deployment of Executable BPEL



Tools available at [www.astroproject.org](http://www.astroproject.org)



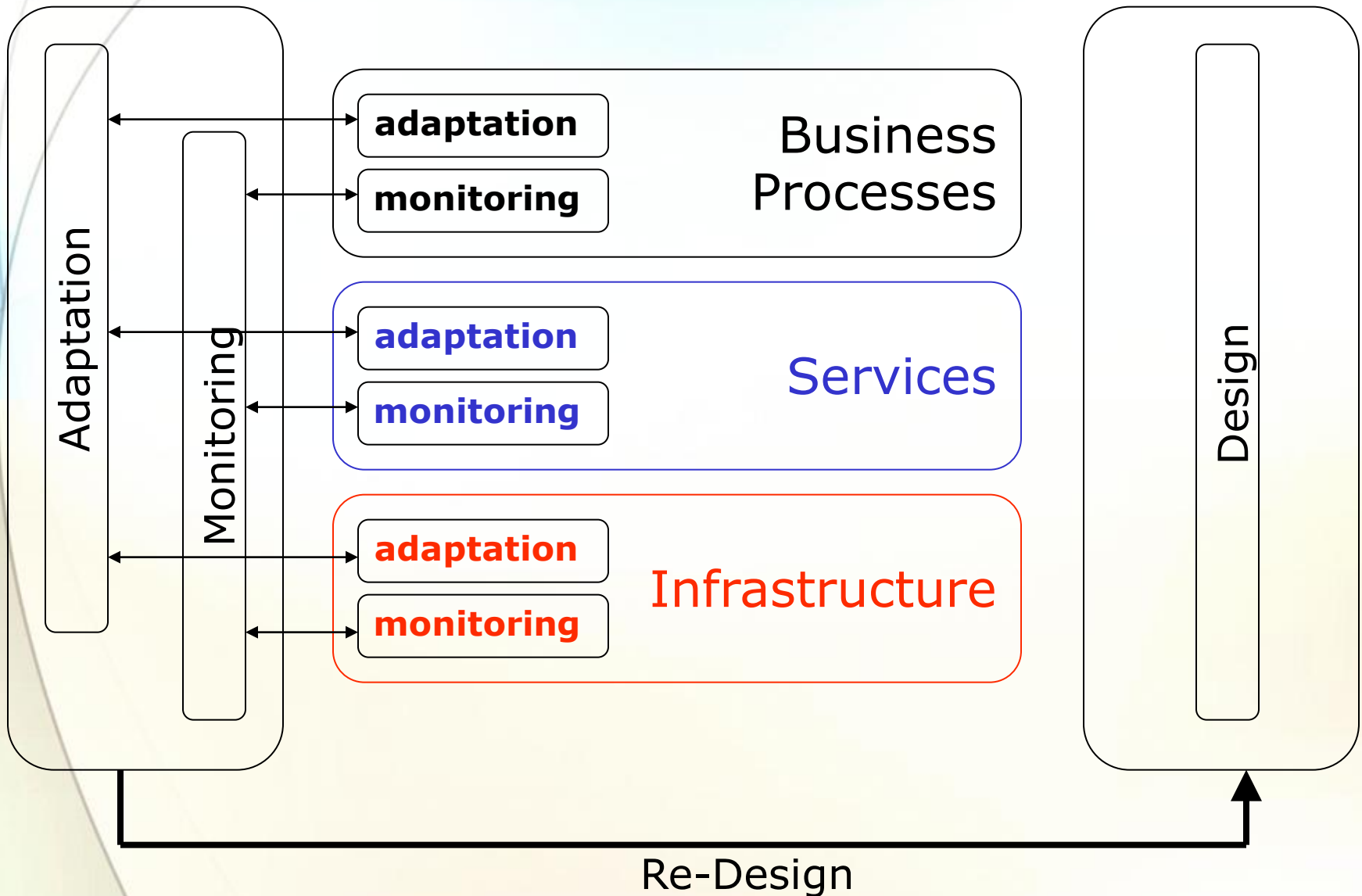
## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - Modeling, Composition, **Monitoring, Adaptation**

## PART II: Future Internet

- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, Composition, Monitoring, Adaptation

# A Framework for Monitoring & Adaptation



# Monitoring of Infrastructure



- Performances
- Available bandwidth
- Resource allocations

# Monitoring of Services



- service functions and behaviors
- service availability
- service access

# Monitoring of Business Processes



- costs & revenues
- effectiveness of services, e.g., social service
- clients satisfaction

# "Cross-cutting monitoring": example



SLA

PHONE  
CONFERENCE  
SERVICE



- Business Process:
  - Monitor: decreasing revenue
  - Action: decrease price
- Service:
  - Monitor: high frequency of cancellations
  - Action: change service
- Infrastructure:
  - Monitor: low performances
  - Action: buy more bandwidth

# "Cross-cutting monitoring"



PHONE  
CONFERENCE  
SERVICE



- Monitoring must understand at which level the problem is and at which level to react
  - Representation of co-relation among different levels
  - Learning co-relations ("proactive monitoring")



# Some monitoring techniques

---

## Assertion-based monitoring:

- Monitors specified as assertions that annotate the BPEL code.
- Annotated BPEL processes are then automatically translated to “monitored processes”
- The approach allows for monitoring of time-outs, runtime errors, behavioral and functional properties.
- “Dynamic monitoring” monitoring rules selected at run-time

***Baresi, Ghezzi, Guinea. ICSOC 2004, 2005***

# Some monitoring techniques (cont.)

---

## Event-based monitoring:

- Monitoring requirements expressed in event-calculus.
- The specified events are observed at run-time and stored in a database.
- Algorithm based on integrity constraint checking analyzes the database

***Mahbub, Spanoudakis, ICSOC 2004, IEEE ICWS 2005***

# Some monitoring techniques (cont.)

---

## Assumption-based monitoring:

- Architecture that separates the business logic from monitoring.
- *"Instance Monitors"* & *"Class Monitors"*
- Formal language for the specification of instance and class monitors
- Automatic translation of monitor specs to Java programs.

***Barbon, Pistore, Trainotti, Traverso. ICSSOC 2005, ICWS 2005***

# Run-Time Monitoring (cont.)

---

## Inputs:

- Abstract BPEL descriptions of component services
- A SLA specification

## Output:

- Notification of violation of SLA ...
- Notification of situations of interest ...
- Aggregated/statistical information

# Run-Time Monitoring: an example



# Run-Time Monitoring: Examples

Check  
Violations  
of SLA



ON LINE  
SHOP



SLA

ON LINE  
BANK



*The Bank does not refuse  
the credentials of the Shop*





# Run-Time Monitoring: Examples

Properties related to **classes** of processes

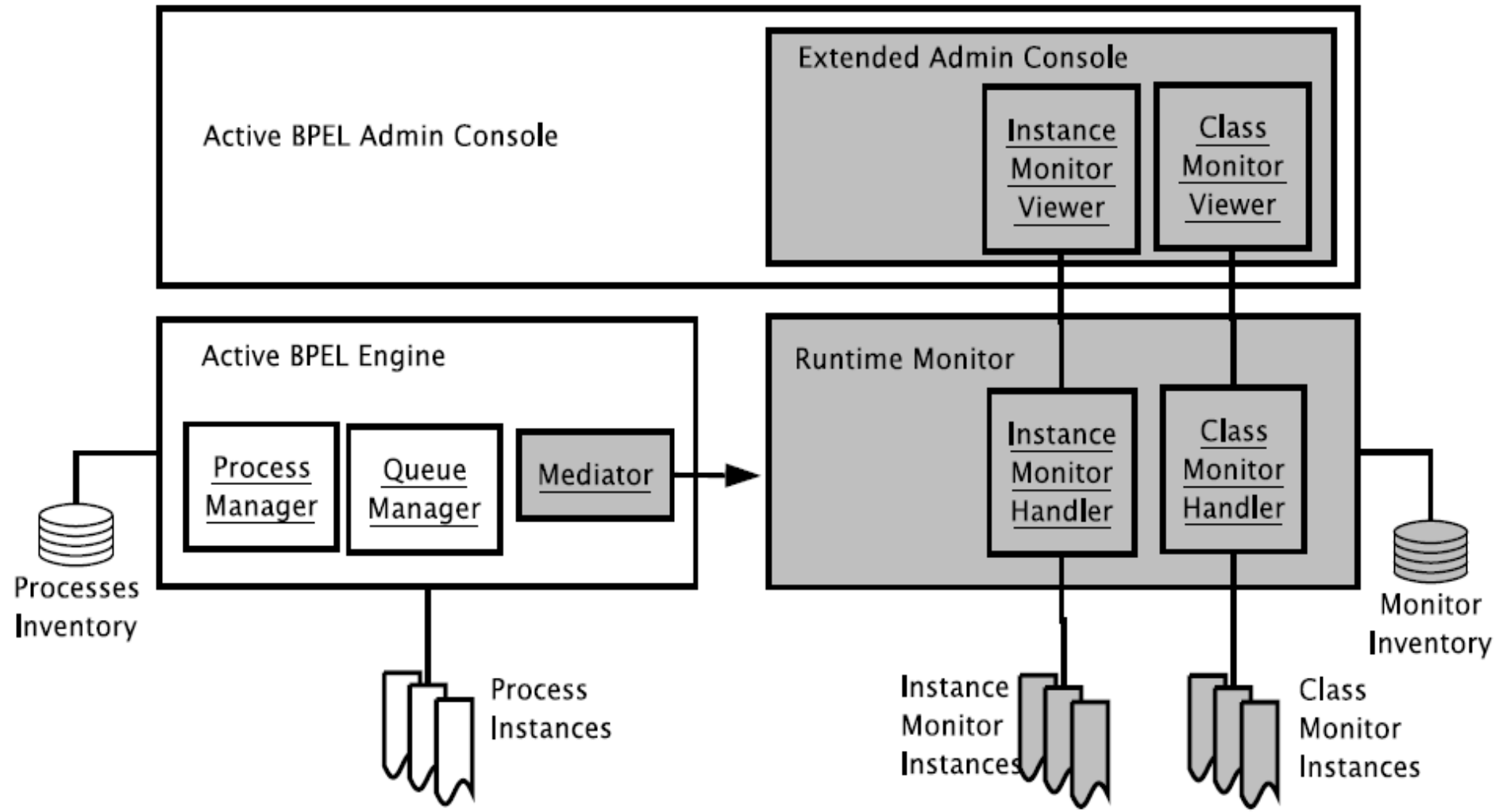


SLA

*The Bank NEVER refuses the credentials of the Store*

*AVERAGE duration for the payment procedure*

# Architecture for Run-time Monitoring




# Process Monitoring Console

ActiveBPEL(TM) Administration - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:50000/BpelAdminExt/active\_process\_detail.jsp?pid=1

CentOS Support ActiveBPEL(TM) Admi...



## Process Detail

<b>Home</b>
<b>Engine</b>
<a href="#">Configuration</a>
<a href="#">Storage</a>
<a href="#">Version Detail</a>
<b>Deployment Status</b>
<a href="#">Deployment Log</a>
<a href="#">Deployed Processes</a>
<a href="#">Partner Definitions</a>
<a href="#">WSDL Catalog</a>
<b>Process Status</b>
<a href="#">Active Processes</a>
<a href="#">Active Class Monitors</a>
<a href="#">Alarm Queue</a>
<a href="#">Receive Queue</a>
<b>Process ID</b>
<input type="text"/> <input type="button" value="Go"/>
<a href="#">Help</a>

<b>ID:</b>	1
<b>Name:</b>	VOS
<b>Namespace:</b>	http://astroproject.org/BusinessProcesses/VOS
<b>Started:</b>	2006/09/19 05:13:14
<b>Ended:</b>	2006/09/19 05:13:29
<b>State:</b>	Completed

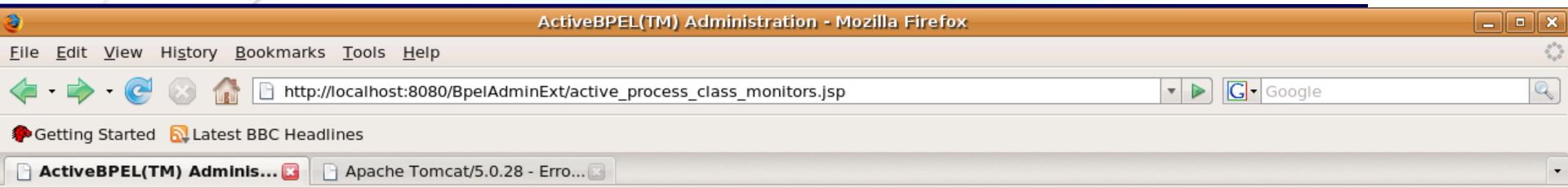
Monitor	Description	Property
VOS_PaymentTime	Payment time	4901
VOS_RetryOnSucc	number of retries for a successful buy	0
VOS_OfferBeforeBank	Bank interaction don't start before user accepts offer	Valid
VOS_Protocol	Communication protocol compliance	Valid
VOS_NotAvailCount	Not availability count	0
VOS_StoreRefuseCc	Bank cannot refuse Shop's payment information	Error: Property StoreRefuseCc violated.


## Log

```
[1][2006-09-19 05:13:14.479] : Executing [/process]
[1][2006-09-19 05:13:14.479] : Executing [/process/sequence]
[1][2006-09-19 05:13:14.480] : Executing [/process/sequence/receive]
[1][2006-09-19 05:13:14.512] : Completed normally [/process/sequence/r
[1][2006-09-19 05:13:14.513] : Executing [/process/sequence/assign]
[1][2006-09-19 05:13:14.514] : Completed normally [/process/sequence/a
[1][2006-09-19 05:13:14.514] : Executing [/process/sequence/assign[2]]
[1][2006-09-19 05:13:14.526] : Completed normally [/process/sequence/a
[1][2006-09-19 05:13:14.526] : Executing [/process/sequence/assign[3]]
```

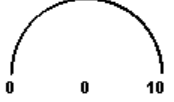
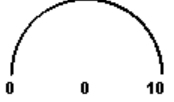
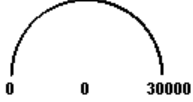

Done

# Process Monitoring Console

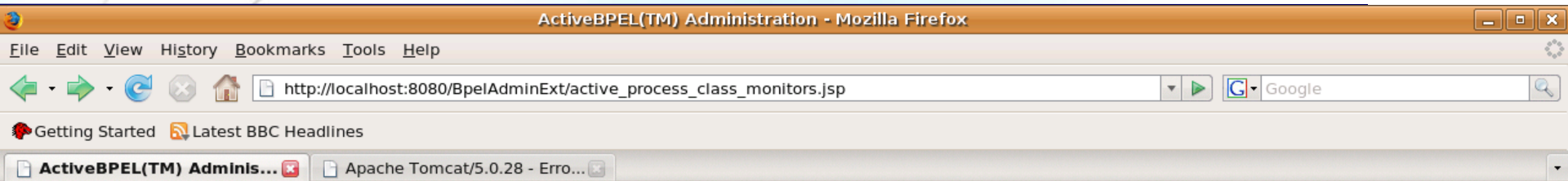



<a href="#">Home</a>
<b>Engine</b>
<a href="#">Configuration</a>
<a href="#">Storage</a>
<a href="#">Version Detail</a>
<b>Deployment Status</b>
<a href="#">Deployment Log</a>
<a href="#">Deployed Processes</a>
<a href="#">Partner Definitions</a>
<a href="#">WSDL Catalog</a>
<b>Process Status</b>
<a href="#">Active Processes</a>
<a href="#">Active Class Monitors</a>
<a href="#">Alarm Queue</a>
<a href="#">Receive Queue</a>
<b>Process ID</b>
<input type="text"/>
<input type="button" value="Go"/>
<a href="#">Help</a>

## Process Class Monitor

Monitor	Description	Dynamic
VOS_CountStoreRefuseCcstore	Number of times that credentials are refused by bank	
VOS_AvgUserRetries	Average of how many times user gets offers	
VOS_AvgPaymentTime	Average payment time	
VOS_GlobStoreRefuseCc	Store credentials are never refused by bank	

# Process Monitoring Console

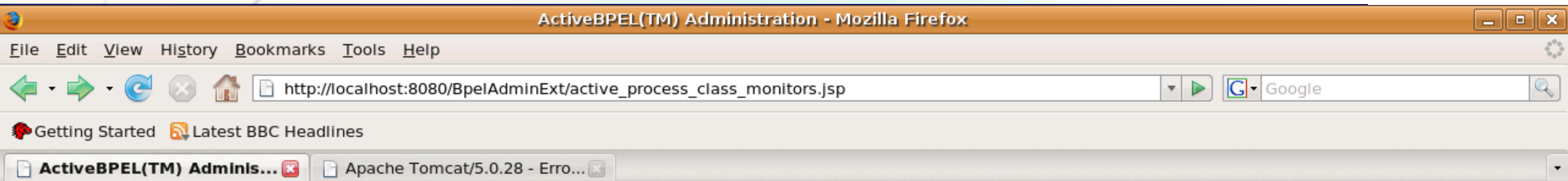



<a href="#">Home</a>
<b>Engine</b>
<a href="#">Configuration</a>
<a href="#">Storage</a>
<a href="#">Version Detail</a>
<b>Deployment Status</b>
<a href="#">Deployment Log</a>
<a href="#">Deployed Processes</a>
<a href="#">Partner Definitions</a>
<a href="#">WSDL Catalog</a>
<b>Process Status</b>
<a href="#">Active Processes</a>
<a href="#">Active Class Monitors</a>
<a href="#">Alarm Queue</a>
<a href="#">Receive Queue</a>
<b>Process ID</b>
<input type="text"/>
<input type="button" value="Go"/>
<a href="#">Help</a>

## Process Class Monitor

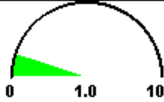
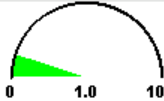
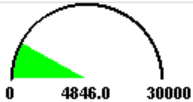

Monitor	Description	Dynamic
VOS_CountStoreRefuseCcstore	Number of times that credentials are refused by bank	
VOS_AvgUserRetries	Average of how many times user gets offers	
VOS_AvgPaymentTime	Average payment time	
VOS_GlobStoreRefuseCc	Store credentials are never refused by bank	

# Process Monitoring Console




<a href="#">Home</a>
<b>Engine</b>
<a href="#">Configuration</a>
<a href="#">Storage</a>
<a href="#">Version Detail</a>
<b>Deployment Status</b>
<a href="#">Deployment Log</a>
<a href="#">Deployed Processes</a>
<a href="#">Partner Definitions</a>
<a href="#">WSDL Catalog</a>
<b>Process Status</b>
<a href="#">Active Processes</a>
<a href="#">Active Class Monitors</a>
<a href="#">Alarm Queue</a>
<a href="#">Receive Queue</a>
<b>Process ID</b>
<input type="text"/>
<input type="button" value="Go"/>
<a href="#">Help</a>

## Process Class Monitor

Monitor	Description	Dynamic
VOS_CountStoreRefuseCcstore	Number of times that credentials are refused by bank	
VOS_AvgUserRetries	Average of how many times user gets offers	
VOS_AvgPaymentTime	Average payment time	
VOS_GlobStoreRefuseCc	Store credentials are never refused by bank	

## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - Modeling, Composition, Monitoring, **Adaptation**

## PART II: Future Internet

- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, Composition, Monitoring, Adaptation



# Run-Time Adaptation (Autonomic)

---

- **Infrastructural Level:**
  - modify bandwidth, change resource allocation (e.g., Grids)
- **Service Level:**
  - dynamic binding
  - reconfiguration of composition
  - **Adaptation to behavioral changes of component services ...?**
  - **... by automated composition!**

# Run-Time Adaptation by composition: example

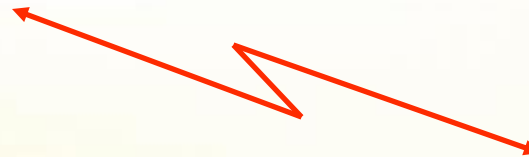
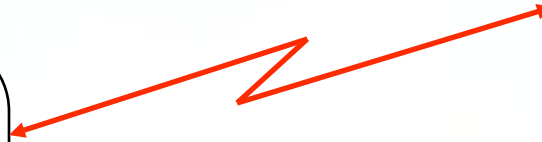
CONF/SMS  
SERVICE



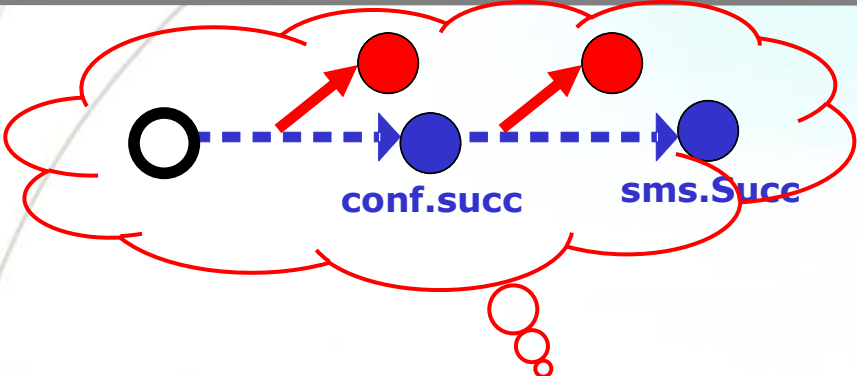
PHONE  
CONFERENCE  
SERVICE



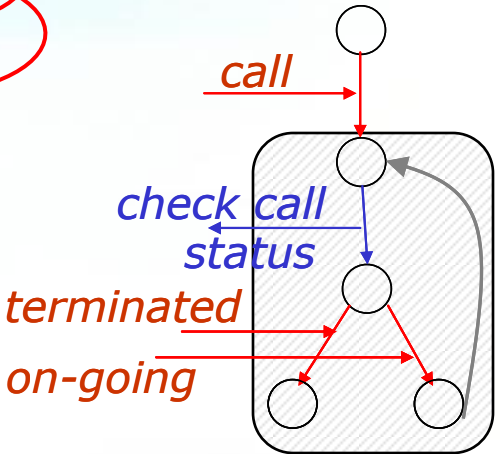
SMS SERVICE  
CENTER



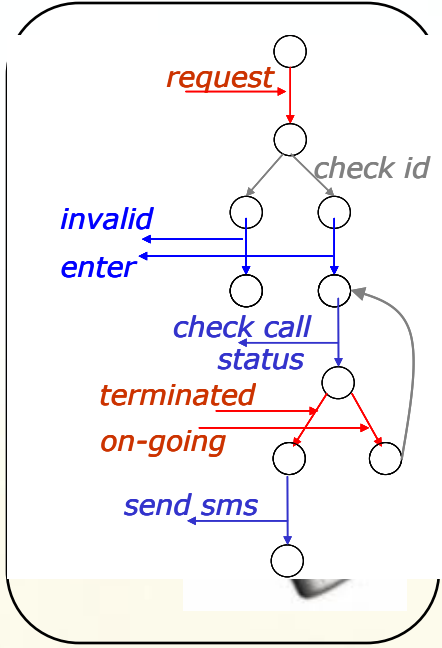
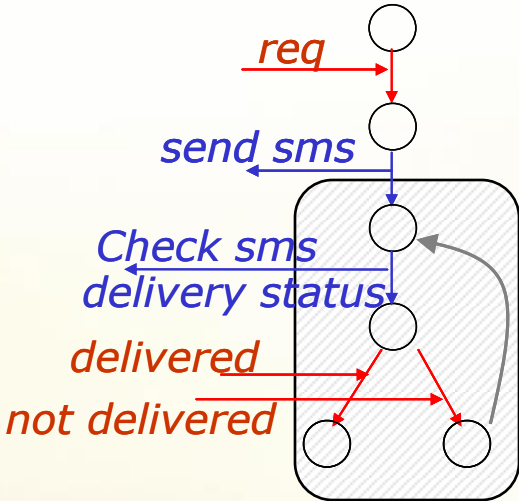
# Run-Time Adaptation by composition: example



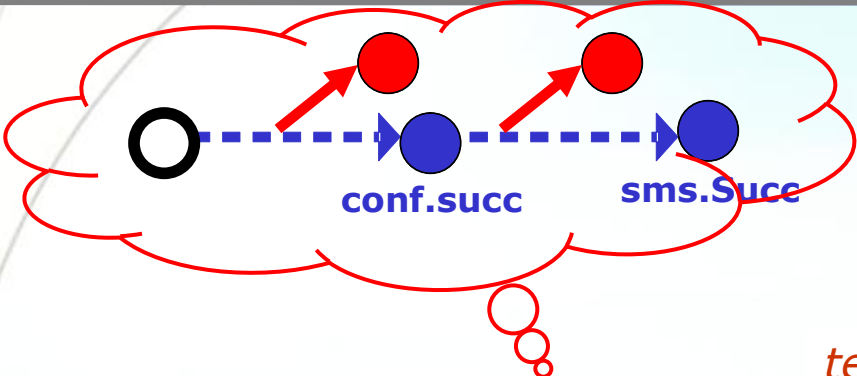
**PHONE CONFERENCE SERVICE**



**SMS SERVICE CENTER**

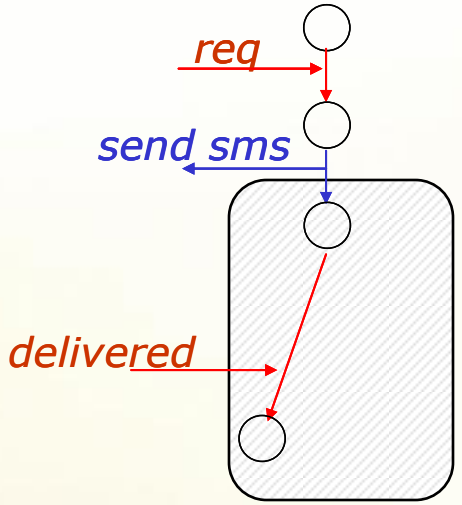
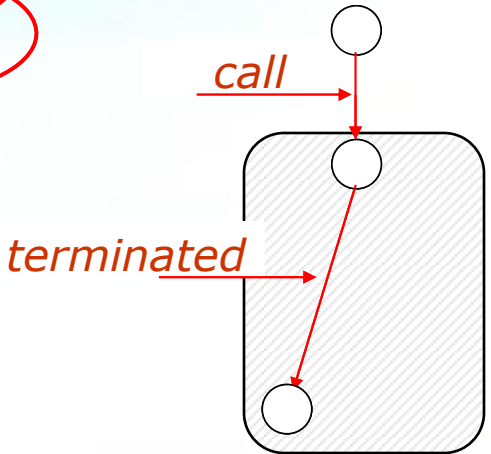
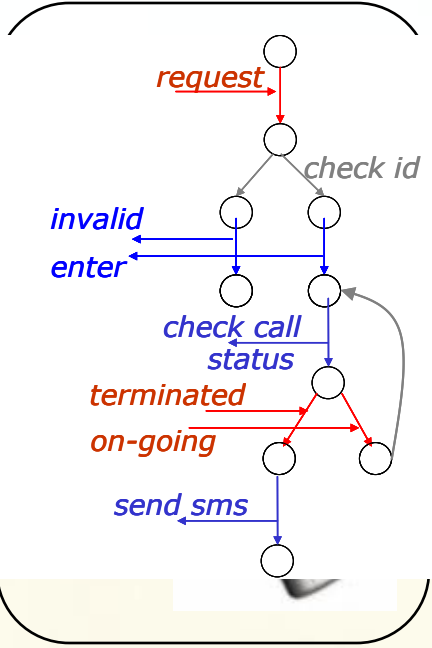


# Run-Time Adaptation by composition: example



**PHONE CONFERENCE SERVICE**

**SMS SERVICE CENTER**



# Run-Time Adaptation (Autonomic)

---

- Infrastructural Level:
  - modify bandwidth, change resource allocation (e.g., Grids)
- Service Level:
  - dynamic binding
  - reconfiguration of composition
  - Adaptation to behavioral changes of component services ...
  - ... can be autonomic ...
  - ... by re-running the automated composition procedure



## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - Modeling, Composition, Monitoring, Adaptation

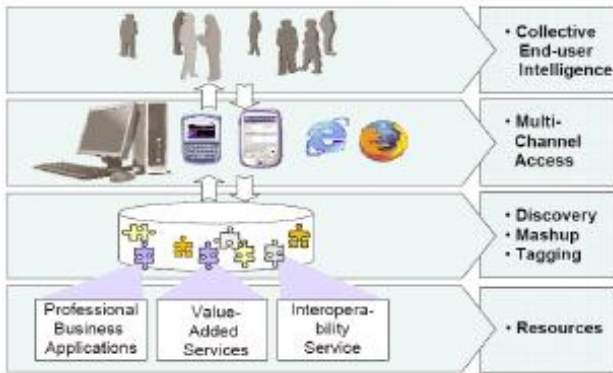
## PART II: Future Internet

- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, Composition, Monitoring, Adaptation



# The Vision of "Future Internet"

## Internet of Services, Service Web



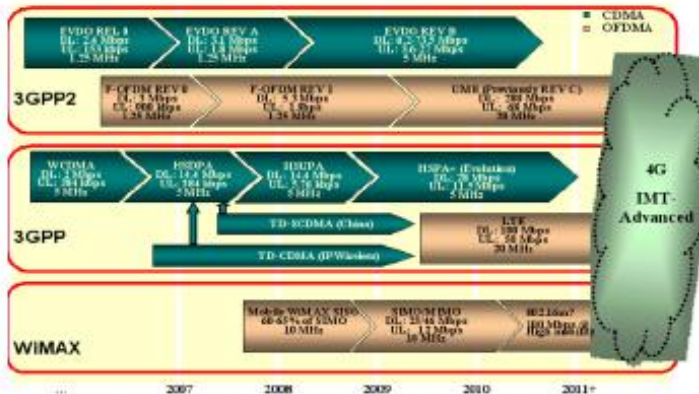
## 3D Internet



Trust



Security



## Networks of the Future

Sources: 3GPP, 3GPP2, Qualcomm, WiMAX Forum  
<http://www.alexandria.unisg.ch/EXPORT/DL/38496.pdf>  
<http://www.itu.int/osg/spu/publications/internetofthings/>  
 Second Life

## Internet of Things

## The Internet of Services ...



**... is extending today's Internet  
to become service-enabled**



**... will enable agile enterprises  
to reach out to a global market and  
focus on core competencies  
but also create global competition**

[Prof.Dr. Lutz Heuser, Vice President SAP Research]



**... will allow the permanent, transparent, seamless, and  
trustworthy provision of services covering all the  
aspects of user's life and business.**

[FIA WG on Internet of Services]

# The Claim

---

- A **key aspect has been underestimated** so far in the research in "Software and Services"
- Software services are software components that provide electronic access to **"real services"**
- **Characteristics** of real services are **very different** from those of the corresponding software services
- The **"Future Internet of Services" should focus on real services,** rather than software services.
- This requires a **research paradigm shift**

# Example: Flight Service

Italy Italiano | Other countries | Help & Contact |  Search

There's no better way to fly.  
 **Lufthansa**

Booking Top Offers Information & Service Miles & More My Account → Login

Booking Check-in

✈ Flight 🏠 Hotel 🚗 Car

Round-trip  One-way

From → Airports Depart on  
 11.08.2008

To → Airports Return on  
 11.16.2008

Search by price +/- 3 days  
 Search for specified travel dates

Adults (>12 years) Children (2-11 years) Infants (up to 2 years)  
1 ▾ 0 ▾ 0 ▾

Class Economy ▾

→ Search flights

→ Search with more options  
→ Multi-segment journey  
→ Award Booking



lufthansa.com lufthansa.com

USA from 459€ Middle East from 399€

\* Return including taxes, fees and charges for flights until March 09.

## Discover America

Tickets available until 11.11.08.

- Buenos Aires from 649 €\*
- Caracas from 499 €\*
- Mexico City from 599 €\*
- Sao Paulo from 649 €\*

\* Return incl.all taxes and service fee

## Online boarding pass ↗

Now you can print out your online boarding pass at home for all flights from Italy to all Lufthansa destinations worldwide

## Current travel information ↗

We keep you updated with the newest flight information and schedule

## Flights from Milan to Europe

Choose your next destination: book your flight for Europe with special price.

- Barcelona from 99 €\*
- Brussels from 99 €\*
- Bucharest from 99 €\*
- Budapest from 99 €\*
- Madrid from 99 €\*
- Paris from 99 €\*

\* return incl.all taxes and service fee

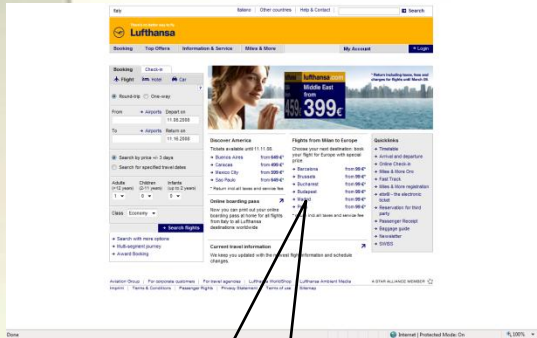
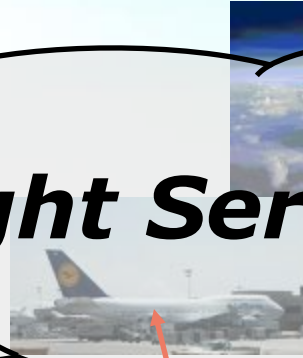
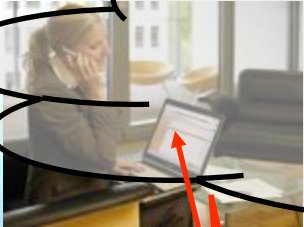
## Quicklinks

- Timetable
- Arrival and departure
- Online Check-in
- Miles & More Oro
- Fast Track
- Miles & More registration
- etix® - the electronic ticket
- Reservation for third party
- Passenger Receipt
- Baggage guide
- Newsletter
- SWISS



# Example: Flight Service

## Flight Service



*Book  
Flight*

*Flight  
delay*

*Your bag is on!*

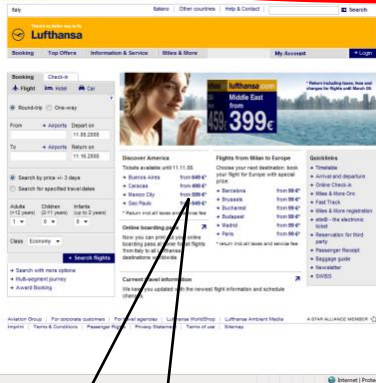
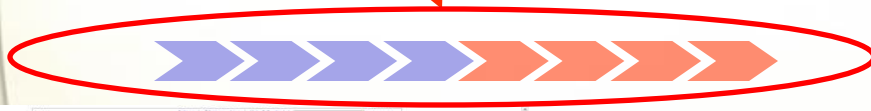
*Connecting  
flights*

# A simple example

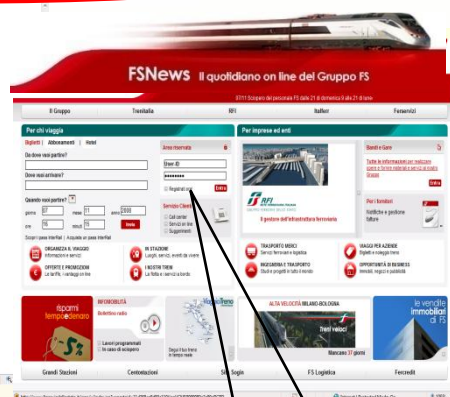


Travel to London

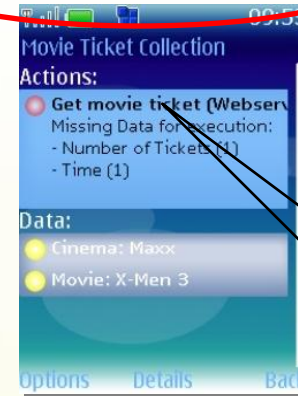
10:00 - 13:00  
Business meeting



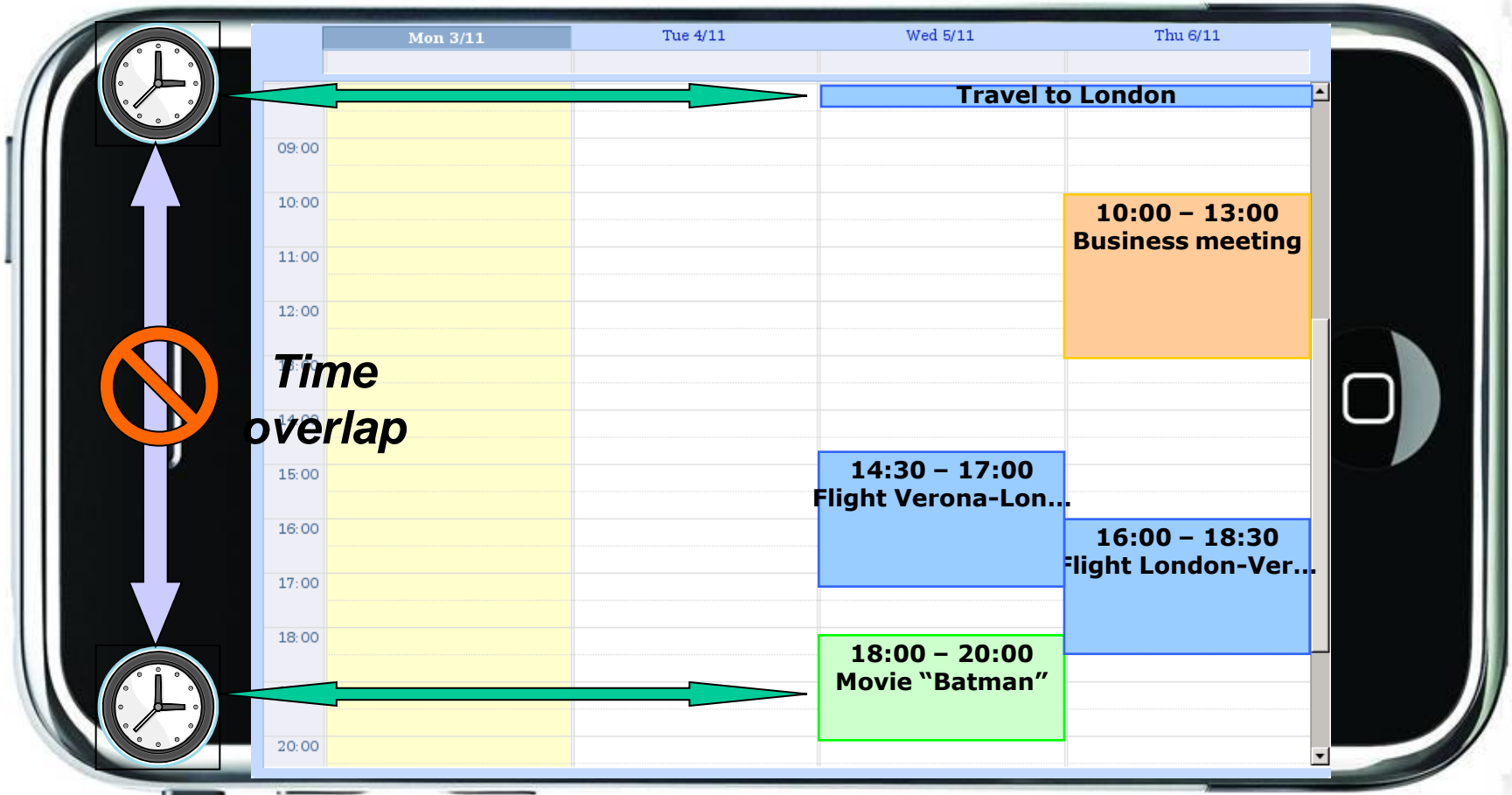
Book  
Flight



Book  
Train



Buy  
Movie  
Ticket



**Time  
overlap**

Mon 3/11

Tue 4/11

Wed 5/11

Thu 6/11

**Travel to London**

09:00  
10:00  
11:00  
12:00  
13:00  
14:00  
15:00  
16:00  
17:00  
18:00  
20:00

**10:00 - 13:00  
Business meeting**

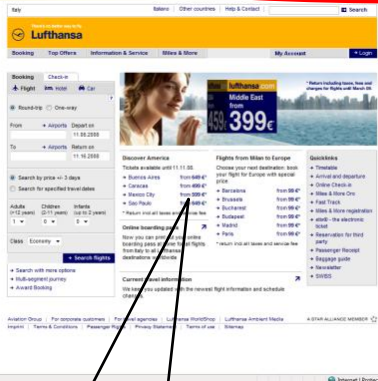
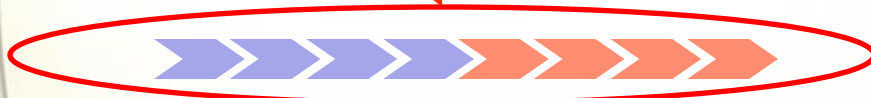
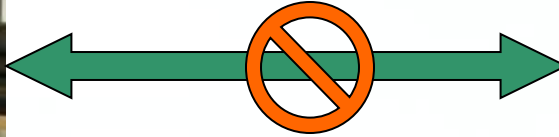
**14:30 - 17:00  
Flight Verona-Lon...**

**16:00 - 18:30  
Flight London-Ver...**

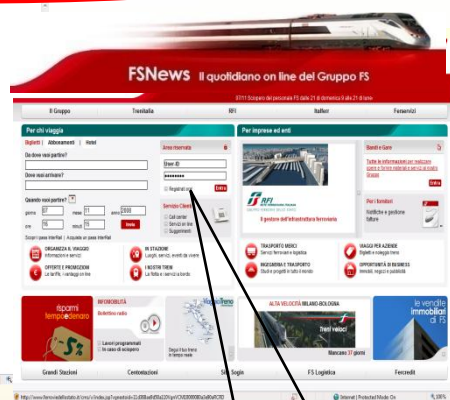
**18:00 - 20:00  
Movie "Batman"**



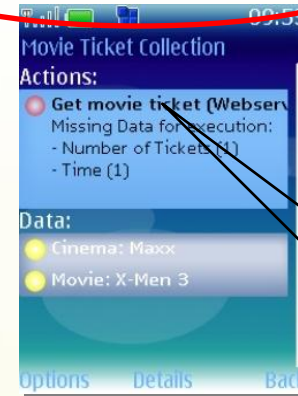
# A simple example



Book  
Flight



Book  
Train



Buy  
Movie  
Ticket

# The Vision (one step forward ...)

**Real Services are very different from Software Services,  
for instance with respect to ...**

→ **... duration:** the time for booking a travel is limited with respect to the duration of the actual travel.

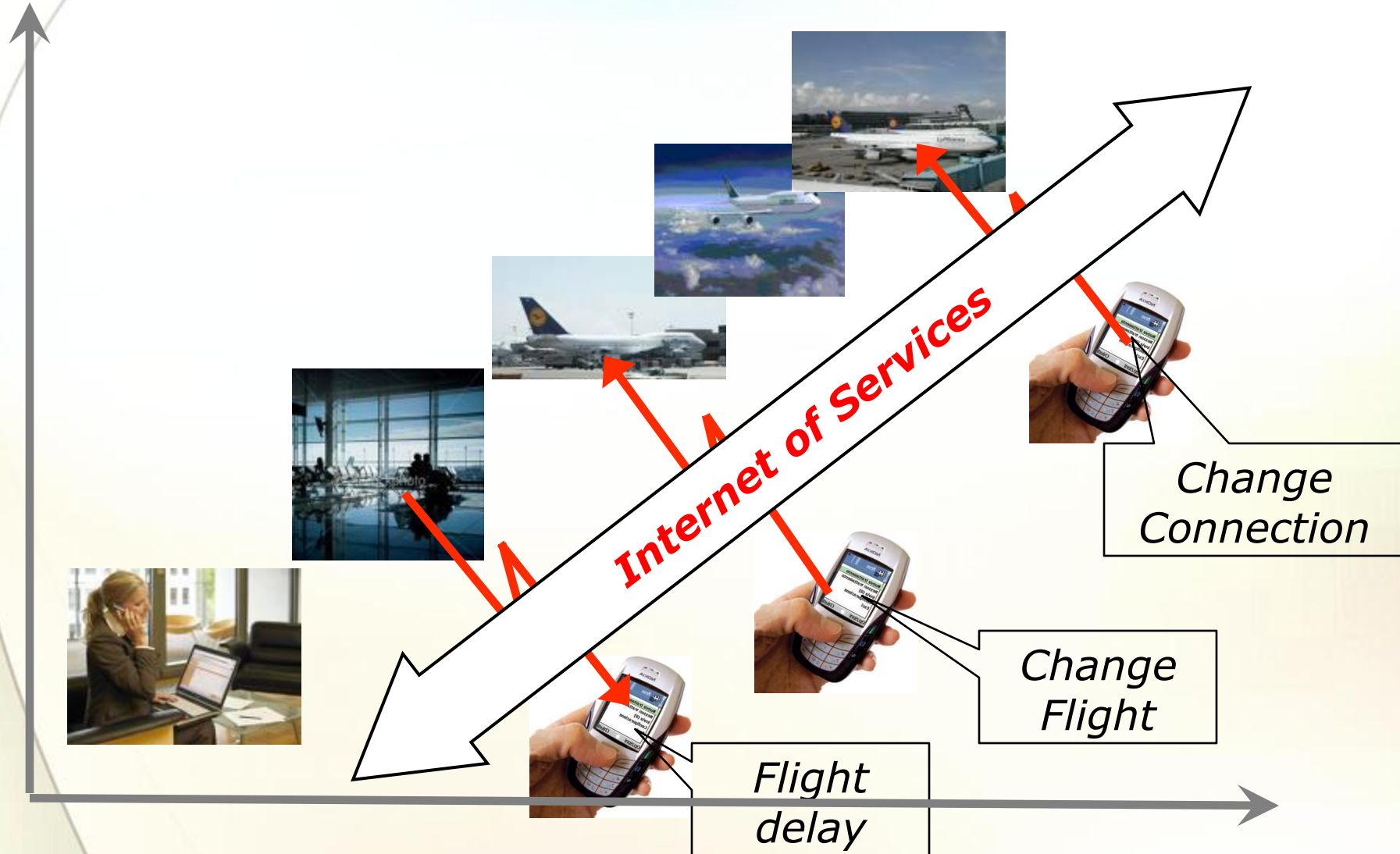
→ **... dynamic:** software services are static and accessible anywhere and anytime; the real services are dynamic and context dependent.

→ **... coupling:** software services are independent and loosely coupled; the real services we use are strongly related.

## The SOC concepts and approaches have to be re-thought:

- ... **monitoring:** from monitoring the execution of software to monitoring the environment where the service operates
- ... **adaptation:** from adaptation among software services to adaptation to service and environment changes and to user's reactions
- ... **composition:** from task/goal driven composition of software to a composition based on how a service relates to core assets for the users

# And ... what about internet?





## **While internet has a minor role for software services ...**

it provides a convenient infrastructure for publishing, discovering, and executing software components.

## **Internet is instead a key enabler for “real” services ...**

it offers a unique capability to ...

→ communicate to the user changes in services and in context,

→ allow the user to react immediately to this dynamicity

**Changes in the services are immediately visible to everybody, similarly to what happens in the Internet of Web pages**

**Internet as the service delivery platform of the future**



## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - Modeling, Composition, Monitoring, Adaptation

## PART II: Future Internet

- Introduction: The Vision
- Planning for the Internet of Services
  - **Modeling**, Composition, Monitoring, Adaptation



# And what about ... Service Modeling?

---

## State of the art:

- A technical description of the functional and non-functional aspects of services
- Interfaces, behavior, quality, security, ....
- A description representing of the technical aspects of software services

# And what about ... Semantics?

---

- “Classical” Semantic Web Services (SWS) allow for:
  - Representing technical aspects of software services (input/output, preconditions, effects)
  - Defining a taxonomy of services.

Example:

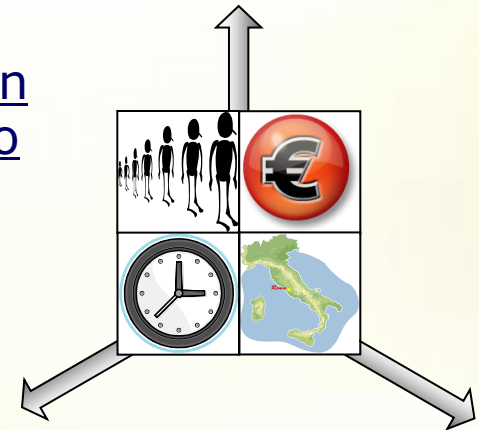
*Food*  
↑  
*Restaurant*  
↑  
*Pizzeria*  
↑  
*PizzaHut*

*Travel*  
↑  
*Flight*  
↑  
*Lufthansa Flight*  
↑  
*LH 3275*

*Entertainment*  
↑  
*Movie*  
↑  
*Batman movie*

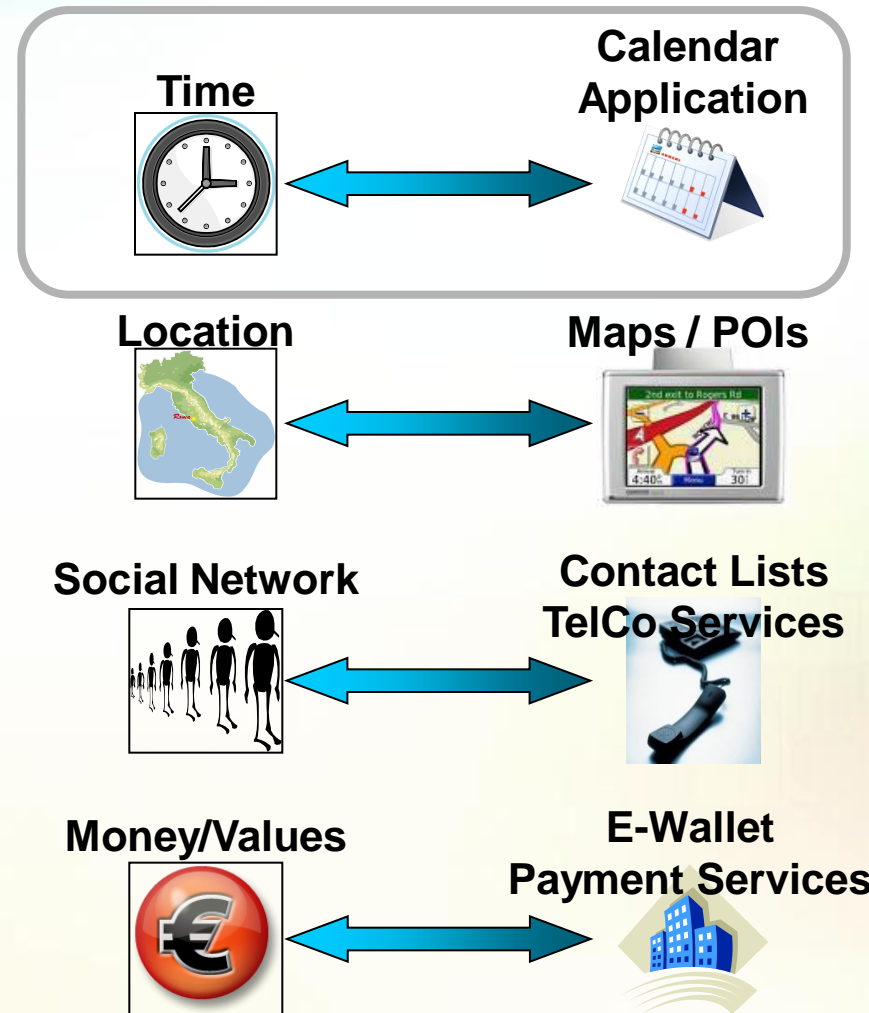
# We need something different!

- A novel way to model services, based on **key assets for the user**. E.g., in the example they might be key parameters such as time, location, money, social network.
  - In the case of *PizzaHut*, I want to know where it is, when it is open, how much does it cost...
  - In the case of *my flight*, I want to know when (schedule), where (airports), who travels...
  - In the case of *Cinema*, I want to know when (movie time), where (cinema location), who comes with me ...

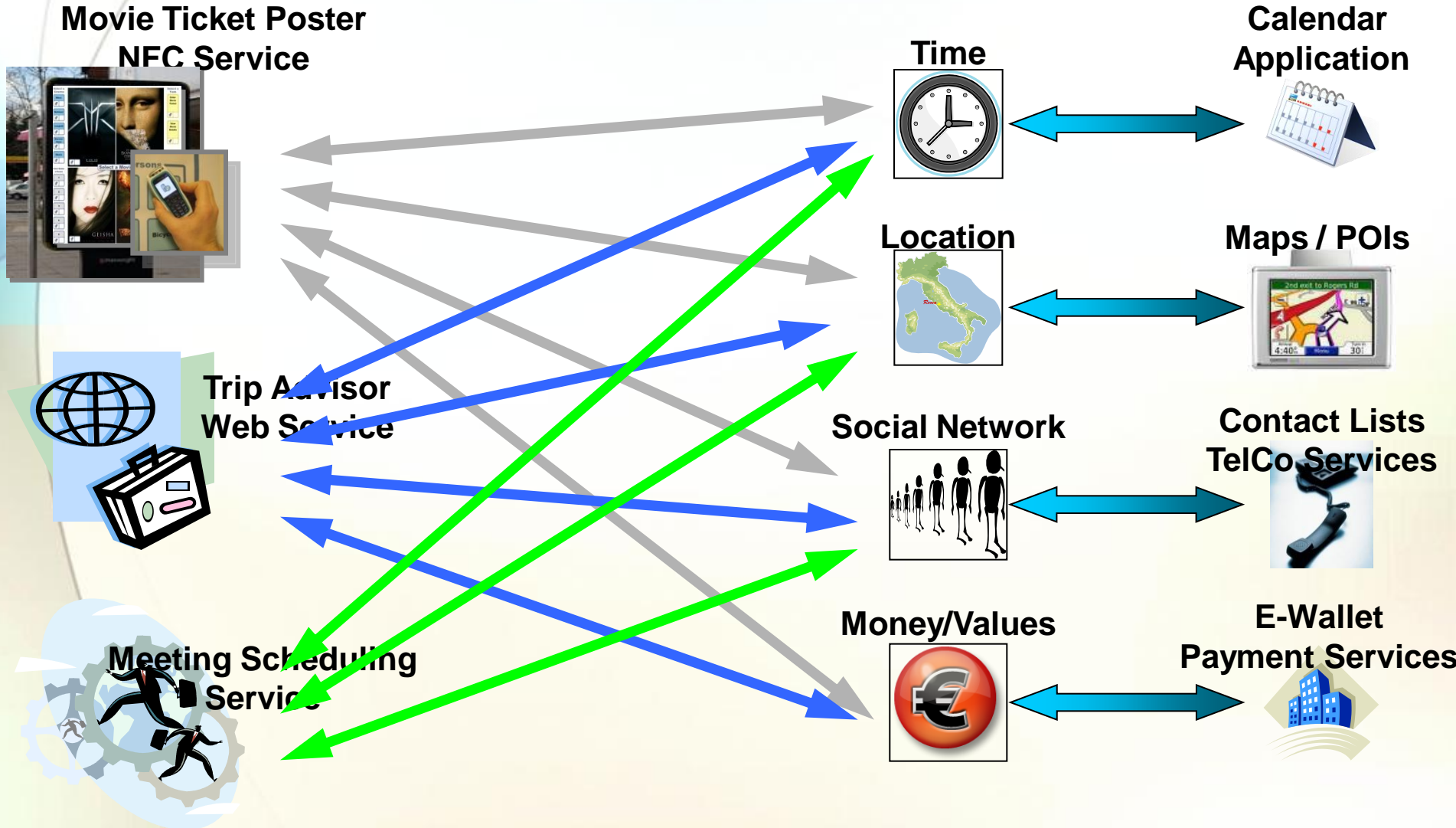


# Example: Personal Mobile Services

- Services are organized along **four core assets** of the user
  - Time** representing the temporal relation of user activities
  - Location** representing the spatial relation
  - Social** representing other parties involved in those activities
  - Money/Values** representing costs and assets of the user's activities
- Well know applications** such as calendars, maps, contact lists, e-wallets can be used to expose services to the user



# Example: Personal Mobile Services



## PART I: Software Services

- Introduction: The Idea
- Planning for Software Services
  - Modeling, Composition, Monitoring, Adaptation

## PART II: Future Internet

- Introduction: The Vision
- Planning for the Internet of Services
  - Modeling, **Composition, Monitoring, Adaptation**

# System (and project) architecture

**Coordinate the execution of external services (composition)**

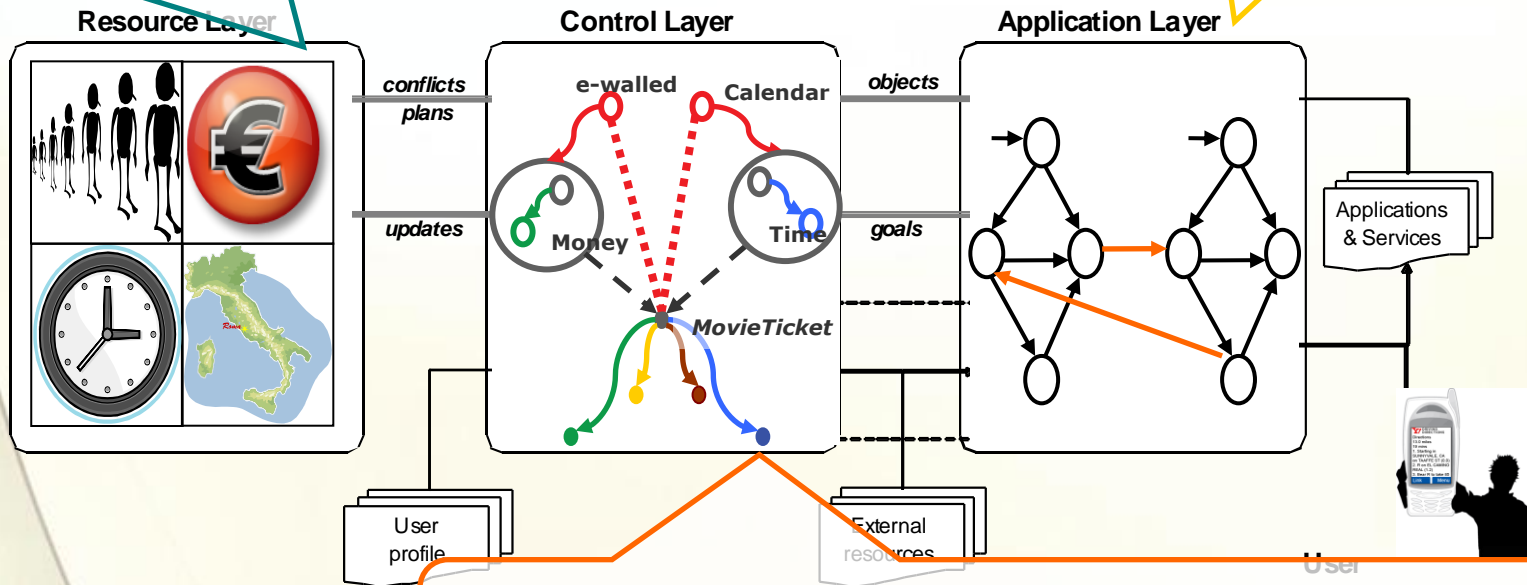
**Monitor and filter external events**

**Manage interactions with the user**

**Keep track of the use of the resources**

**Identify and analyze potential inconsistencies**

**Find ways of resolving these problems**

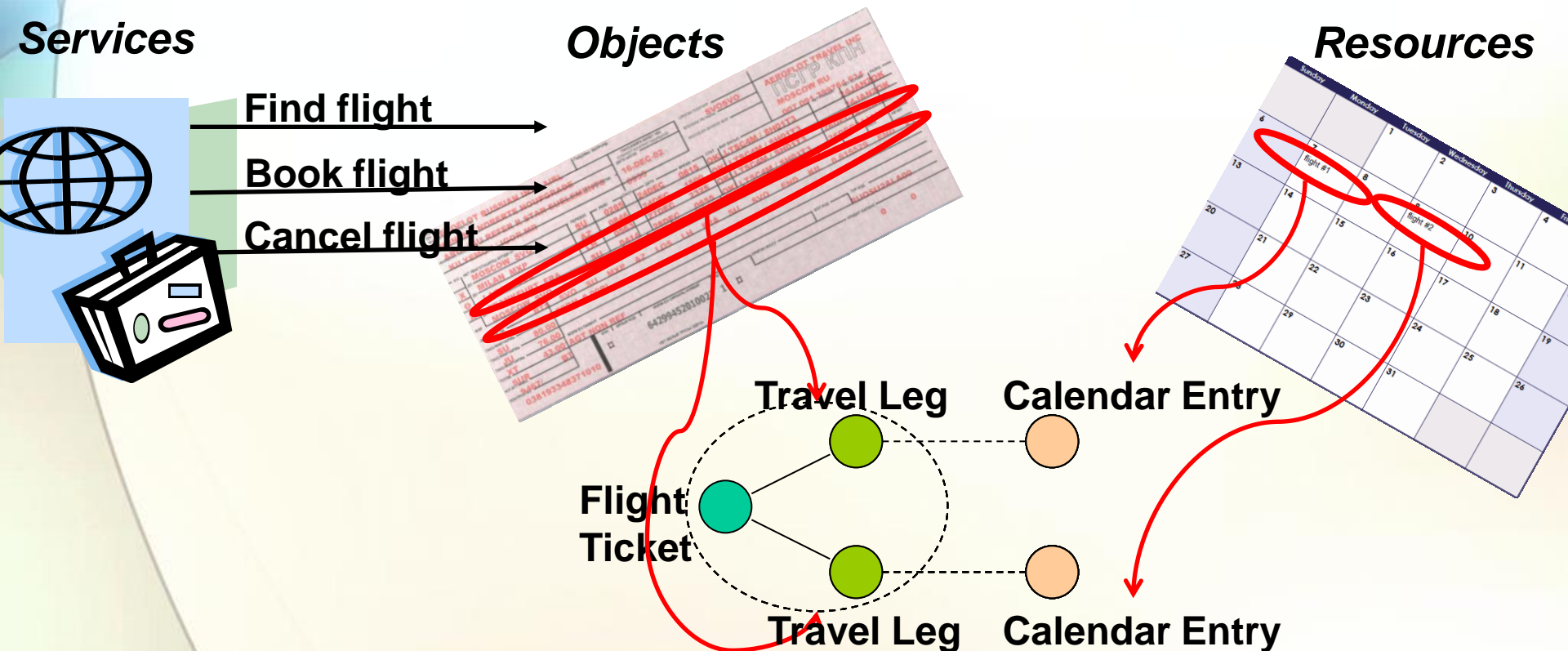


**Coordinate the activities on the customer services**  
**Find relations among services (semantics)**

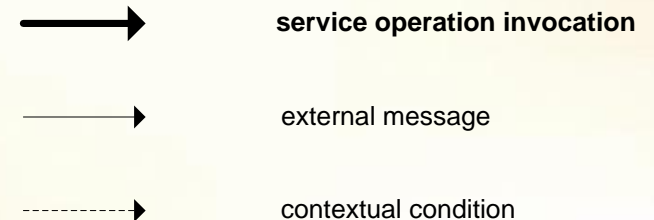
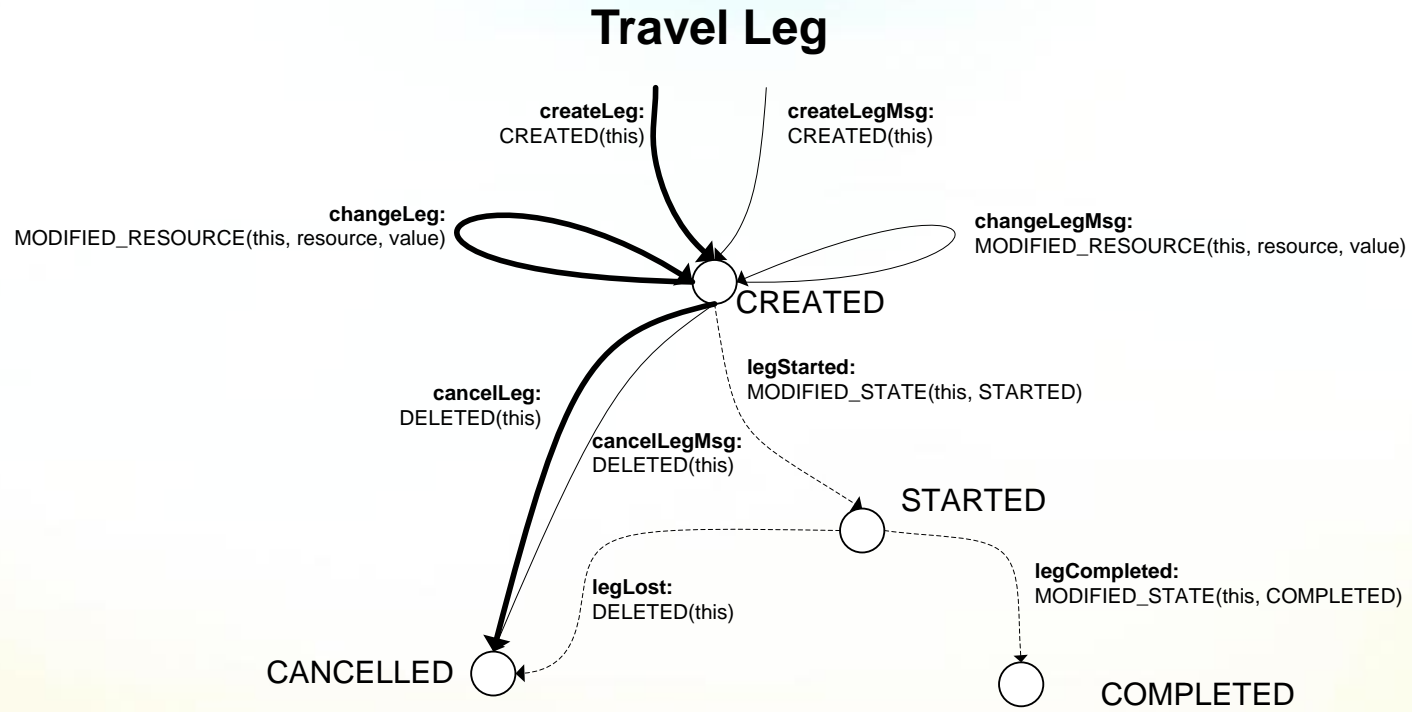


# Application layer

- Problem: how to keep **services and resources aligned**
- Solution: **objects** are used to model service behaviors and resource allocation

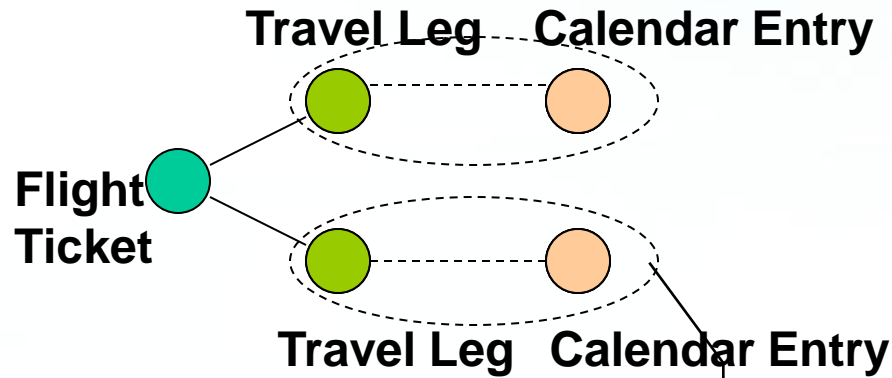


# Application layer: Modeling object/service evolution



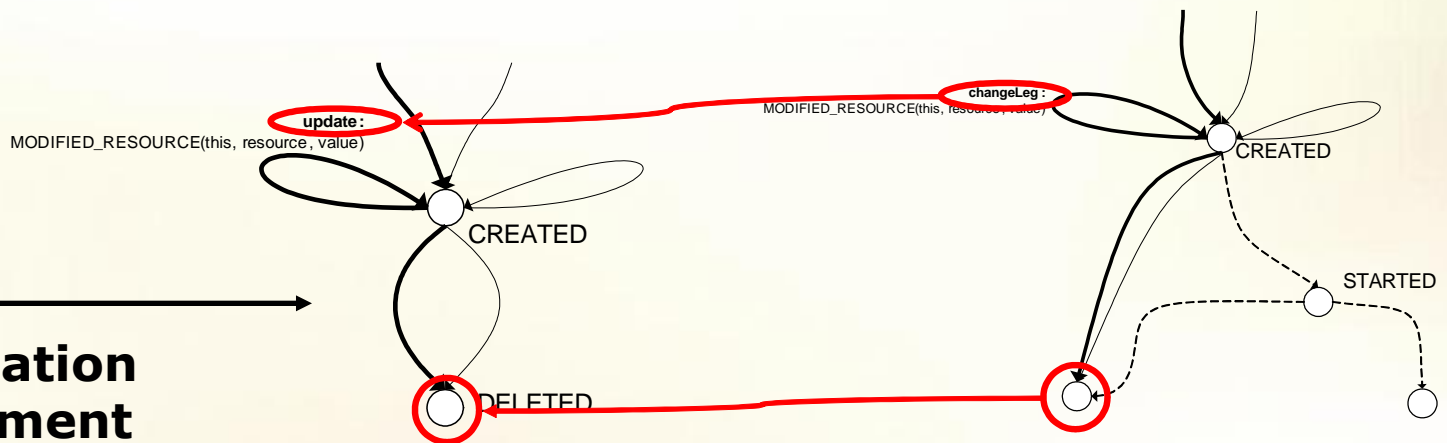
# Application layer

## Object and service coordination



**Coordination group**

**Coordination requirement**



# The Vision: Personal Mobile Services



State of the art:

**“Many Services – Many Applications”**

**Vision**

**“Many Services – One Application”**

## Life Centered Domains ...



- user specific needs



- user assets and resources

## Business Centered Domains ...



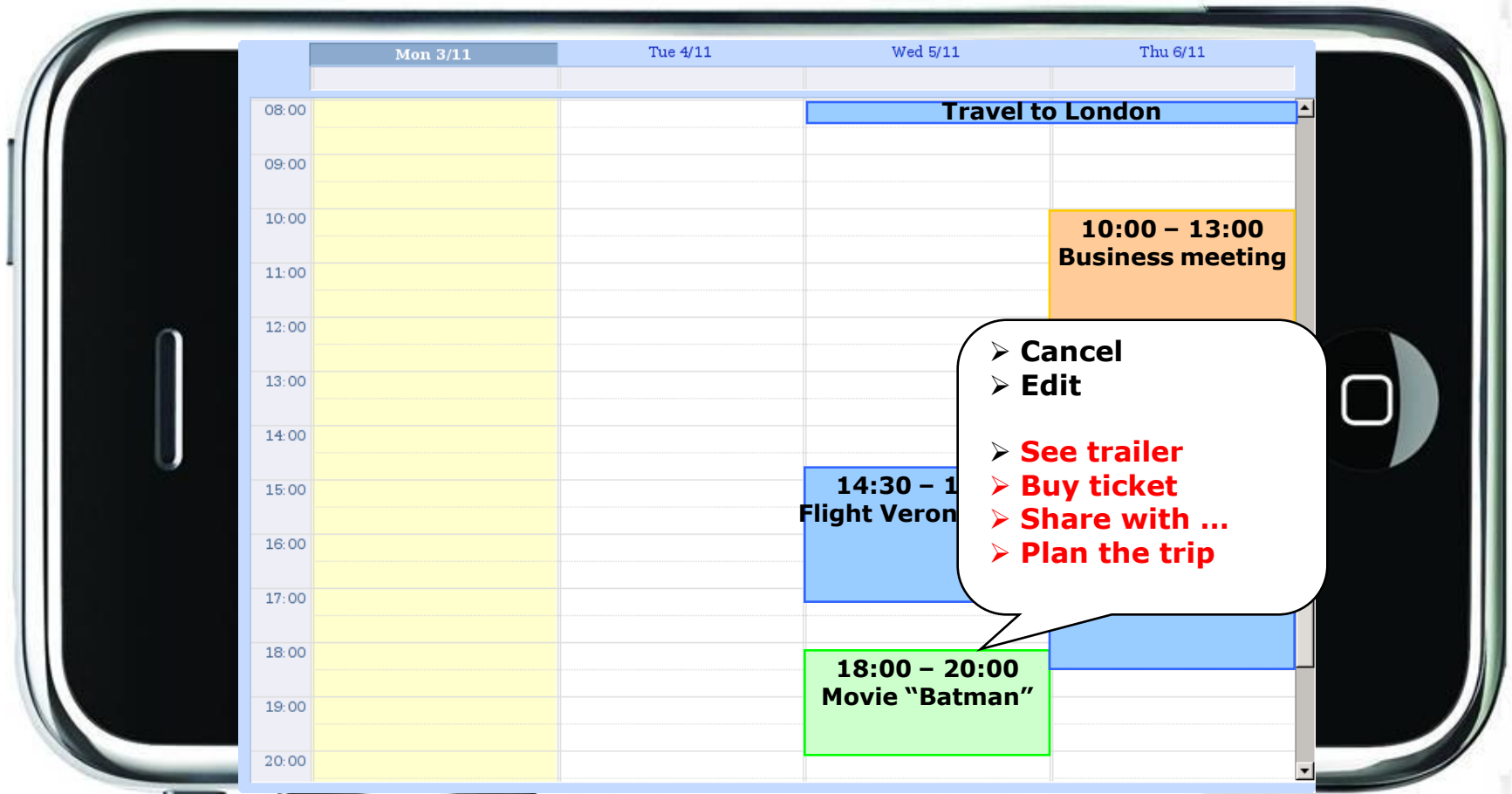
- business specific needs



- company's key assets

# Life-centered domains

- The Agenda as the “service composition work-bench” for mobile personal services!





# Business-centered domains

- Which is the “service composition workbench” for em





## **Each business specific domain requires a specific workbench ...**



**... emergency management:  
map for planning & running emergency services**



**... e-health and social services:  
citizen personal record**



**... small medium banks:  
bank account plan**

## The Future Internet of Services requires a Research Paradigm Shift in:

### Modeling:

technical description of  
the functional & non-  
functional aspects



description of how the  
use of real services  
affects consumers

### Monitoring:

properties related to  
the execution of  
software components



properties of the  
physical environment  
where the real services  
operate

## The Future Internet of Services requires a Research Paradigm Shift in:

### Adaptation:

reaction to changes in software services



reaction to changes in the physical environment where services operate, and to users' behaviors

### Composition:

a software engineering task: "task/goal driven" composition of software components



composition based on emergent needs, constraint/opportunities of the consumers

**Thank you for your attention!**

