Service-oriented Architecture – Experiences, Ideas, Innovations

Prof. Dr. Gregor Engels
engels@upb.de

Szklarska Poręba, Poland
19 September 2011

That‘s my life

...

1991 -1997	**Professor Software Engineering and Information Systems, University of Leiden (NL)**
since 1997	**Professor Information Systems, University of Paderborn (D)** - Research Topics: MDA, UML, DSL, SOA, SPL, SQA, MBT, … - currently 19 PhD students, > 200 scientific publications
since 2005	**Chairman of the Board of Directors, s-lab (Software Quality Lab), PPP-institute, University of Paderborn**
since 2005	**Scientific Director Capgemini, CSD Research, Munich**
since 2011	**Collaborative Research Center „On-the-Fly Computing“, funded by Deutsche Forschungsgemeinschaft**

s-lab – Software Quality Lab

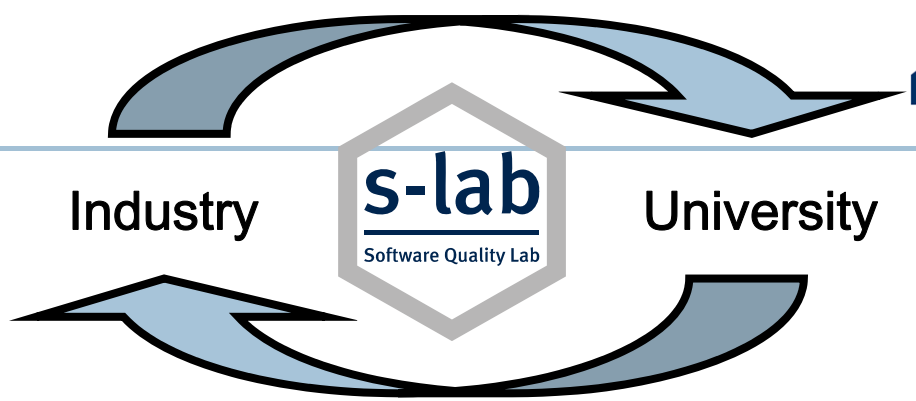
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

- **Software Engineering competence & technology transfer**
- **Objective: improved quality of industrial software development**

- **Impact on academic research and university education**

- **Prerequisite: scientific relevance**

s-lab Partners



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

dSPACE

HELLA Ideen für das Auto der Zukunft

HJP CONSULTING.

Orga Systems.

SAFRAN Morpho

S&η

WINCOR NIXDORF

UNITY

Prof. Dr. Gregor Engels
Information Systems

Prof. Dr. Uwe Kastens
Programming Languages

Prof. Dr. Hans Kleine Büning
Knowledge-based Systems

Prof. Dr. Franz J. Rammig
Embedded Systems

Prof. Dr. Wilhelm Schäfer
Software Engineeering

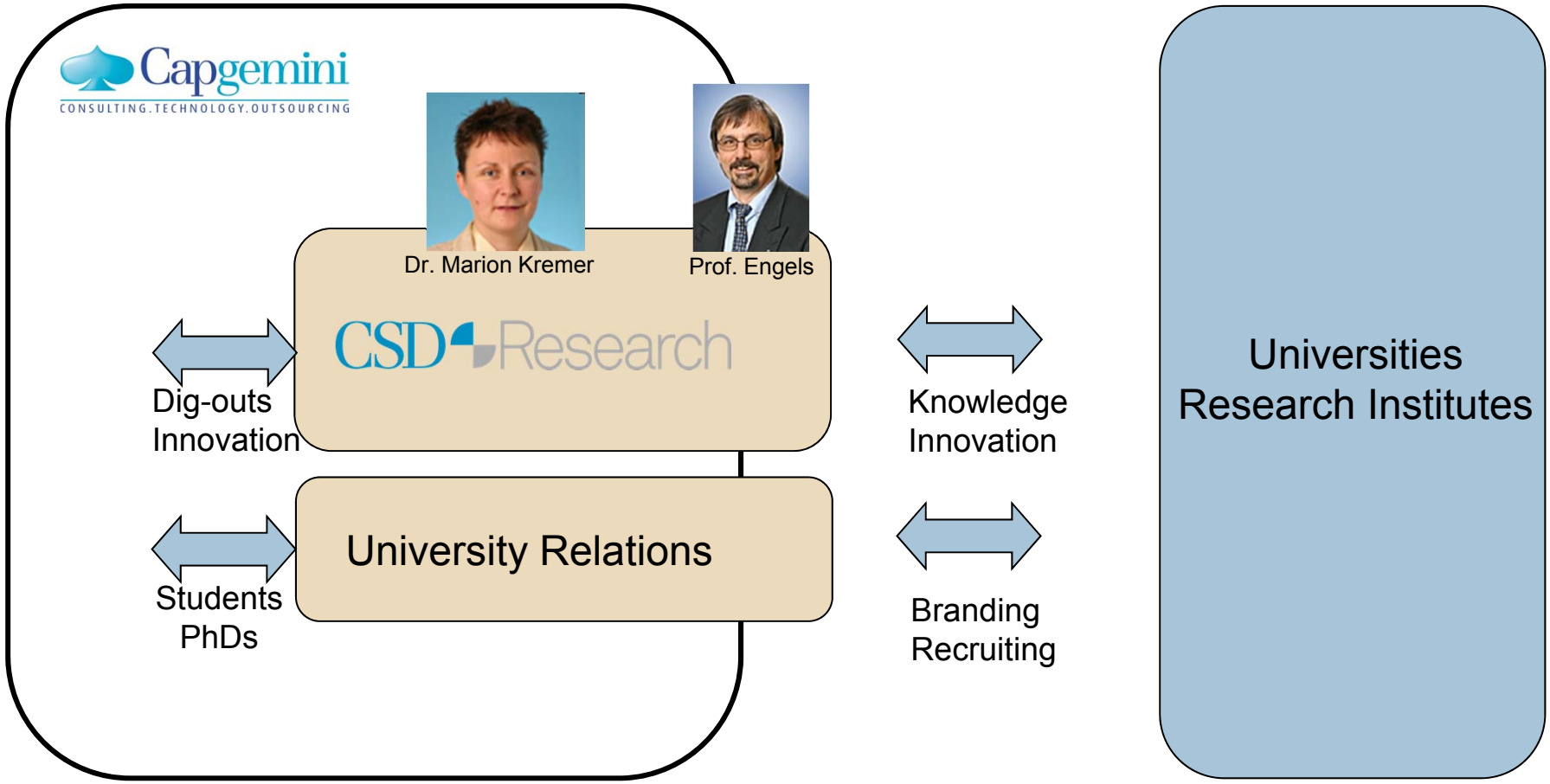
- 20 PhD students
- Funding: more than 5 Mio Euro (since 2005)

That‘s my life

...

1991 -1997	**Professor Software Engineering and Information Systems, University of Leiden (NL)**
since 1997	**Professor Information Systems, University of Paderborn (D)** - Research Topics: MDA, UML, DSL, SOA, SPL, SQA, MBT, … - currently 19 PhD students, > 200 scientific publications
since 2005	**Chairman of the Board of Directors, s-lab (Software Quality Lab), PPP-institute, University of Paderborn**
since 2005	**Scientific Director Capgemini, CSD Research, Munich**
since 2011	**Collaborative Research Center „On-the-Fly Computing“, funded by Deutsche Forschungsgemeinschaft**

Capgemini - CSD Research / University Relations



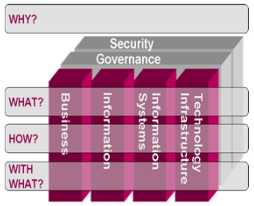
Our architecture methods provide comprehensive, proven solution templates at all levels of a project

Architecture framework

Corporate level Business Architecture

IAF – Integrated Architecture Framework

- Comprehensive architecture framework
- 100% adaptable to specific client situations; from single projects to corporate transformation
- Recognised by the Open Group IT Architecture Certification Program
- Viewed as market leader by analysts (Forrester, Gartner, IDC) and key players (Microsoft, HP, Cisco, SAP, …)



Corporate level IT Architecture

Quasar Enterprise

- Service-oriented design of application landscapes
- Method module to supplement IAF
- Identification and design of domains and services
- Reference architectures and samples for integration and evolution

Application level IT Architecture

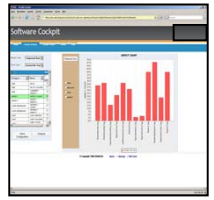
Quasar – Quality Software Architecture

- Overall methodology for system design (specification, construction)
- Software development environment “out-of-the-box”
- Components on Open Source basis
- Model-driven development (models and tools)

Proactive controlling of software quality

Quasar Analytics

- Quality model, figures, software cockpit
- Quality gates in the development process
- Intelligent test procedures and test automation



Motivation

Role of Capgemini CSD Research

Projects

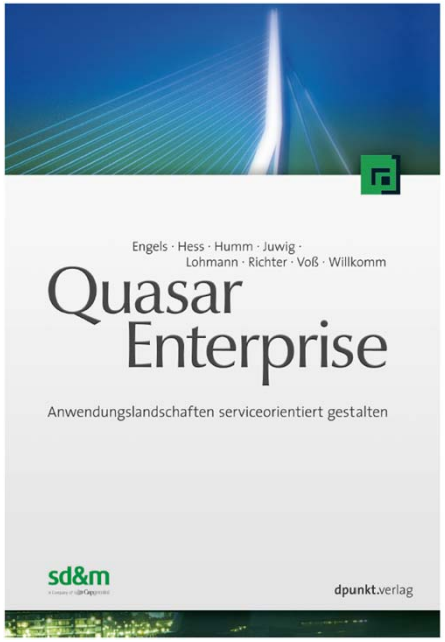
- Development of tailored software solutions
- Implementation and roll-out of standard software
- System integration

dig-out best practices and generalize

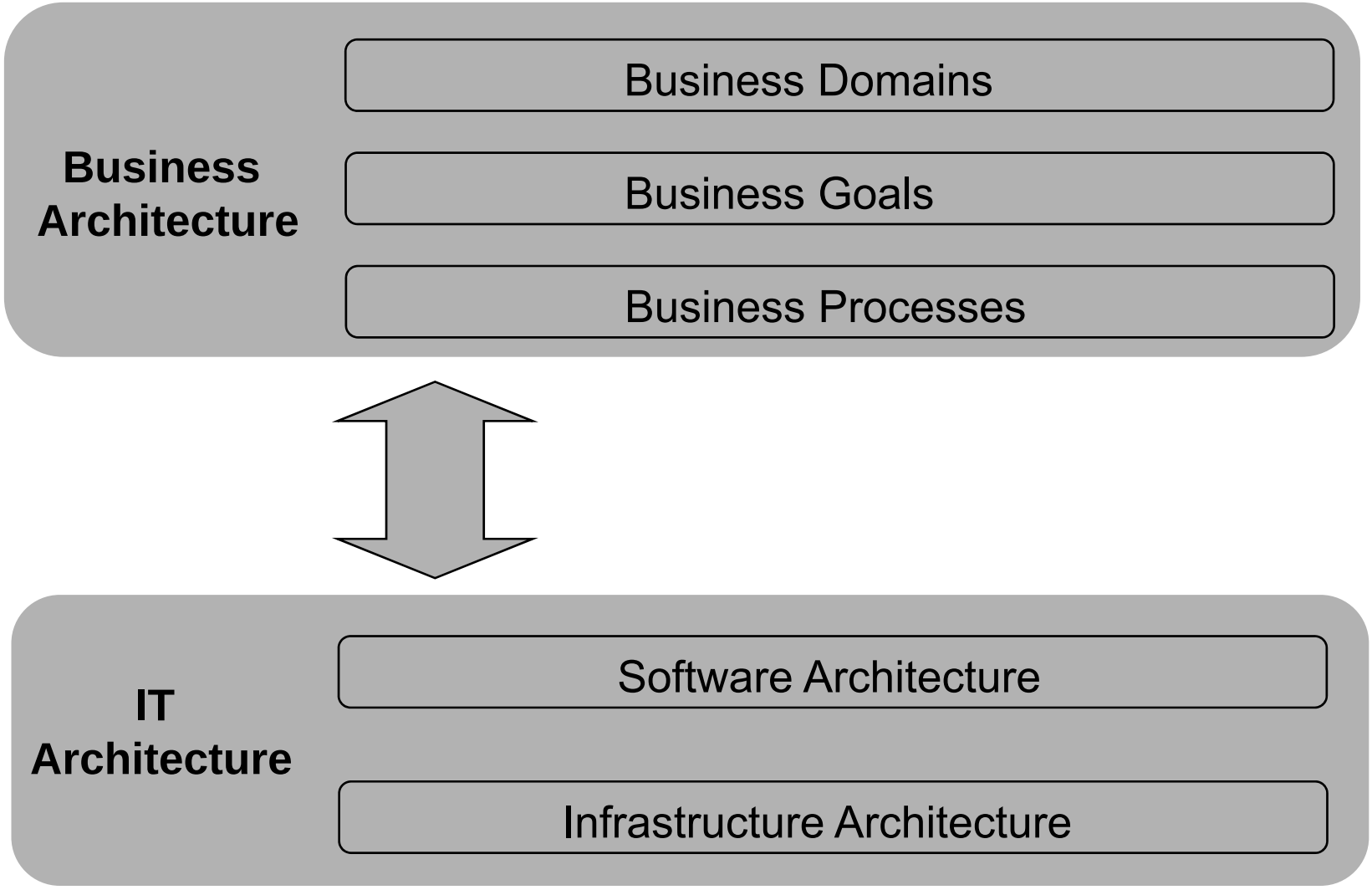
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
CSD Research

roll-out and teach

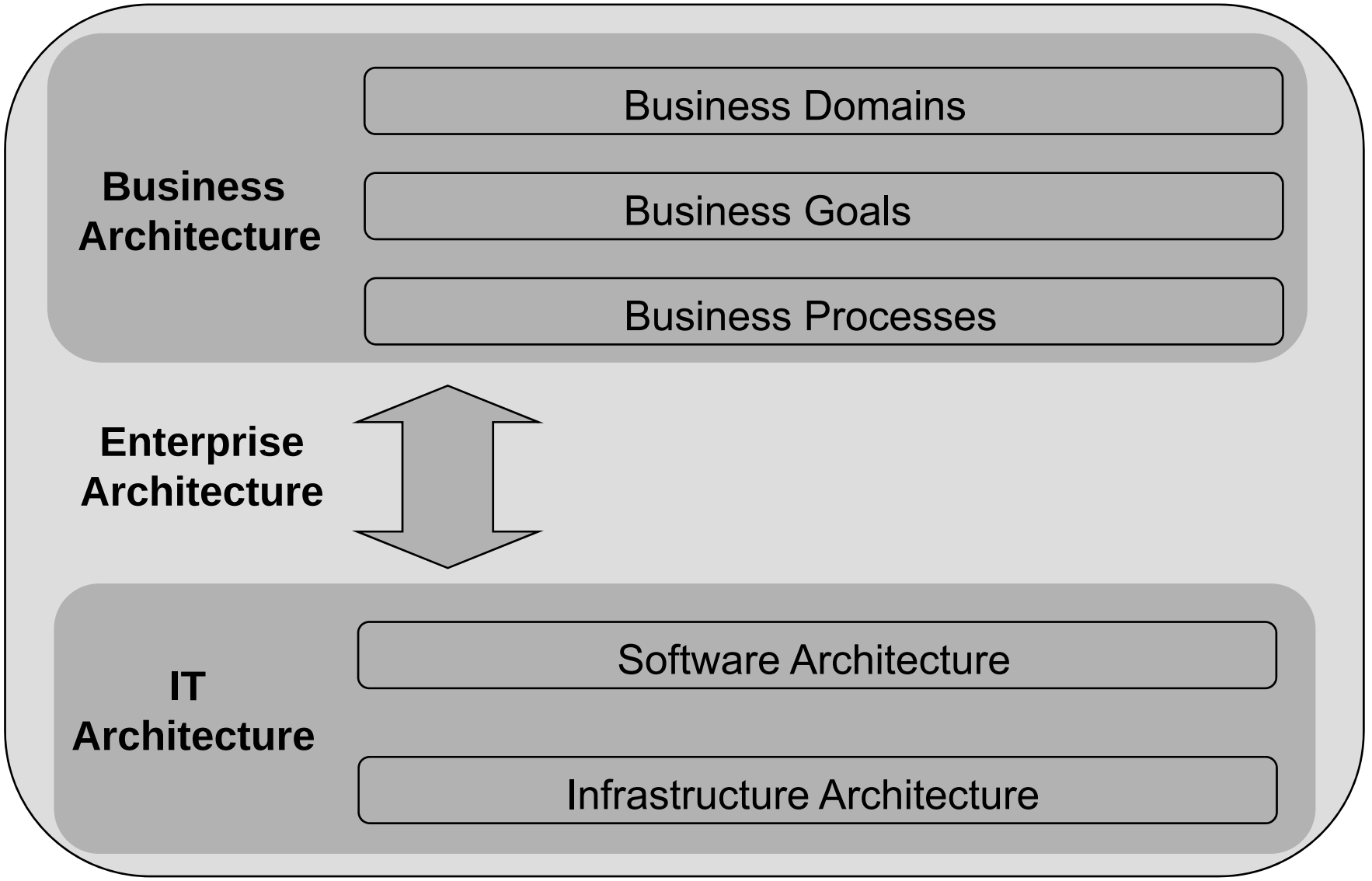
Software Engineering Support



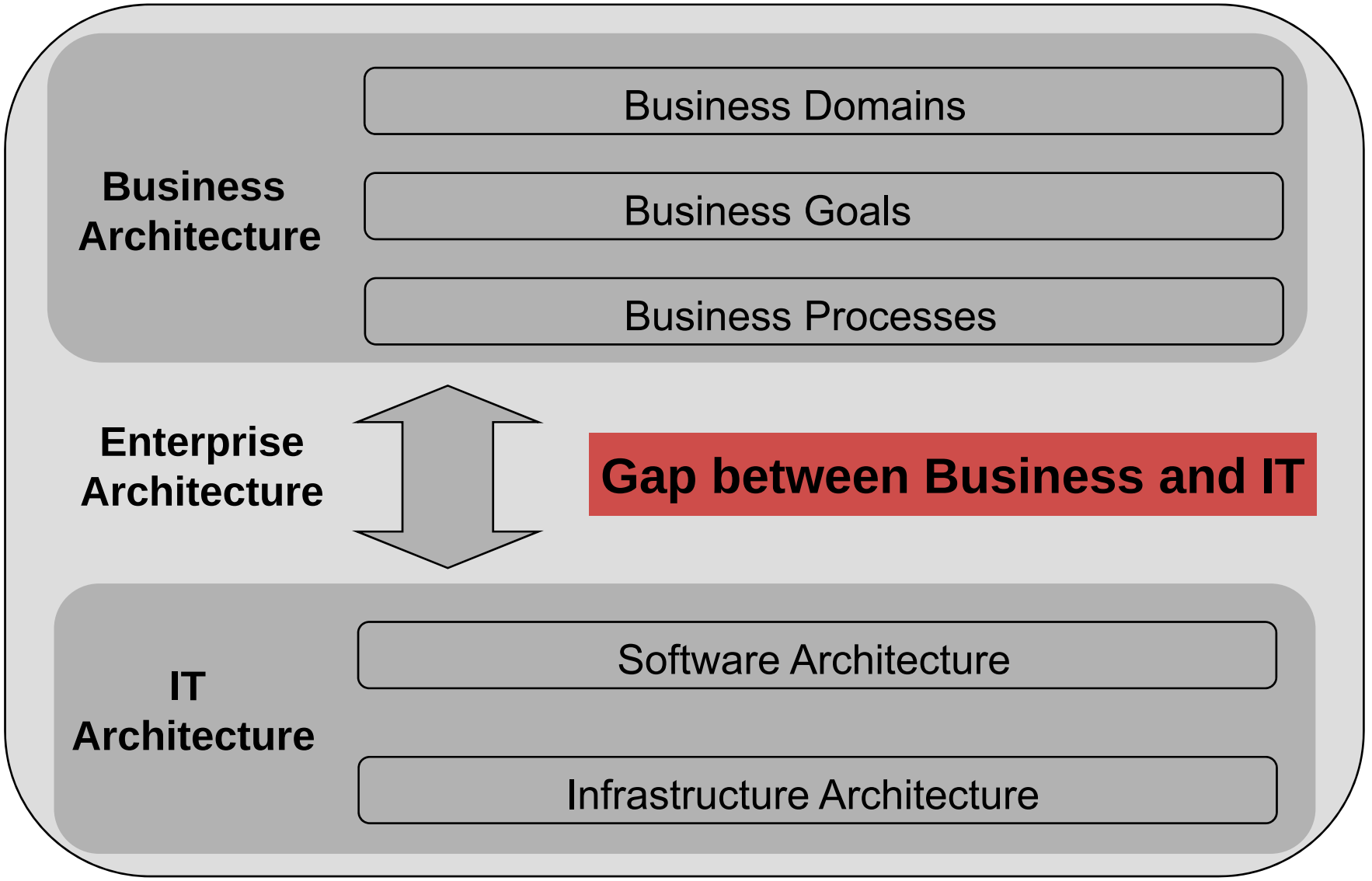
Business meets IT



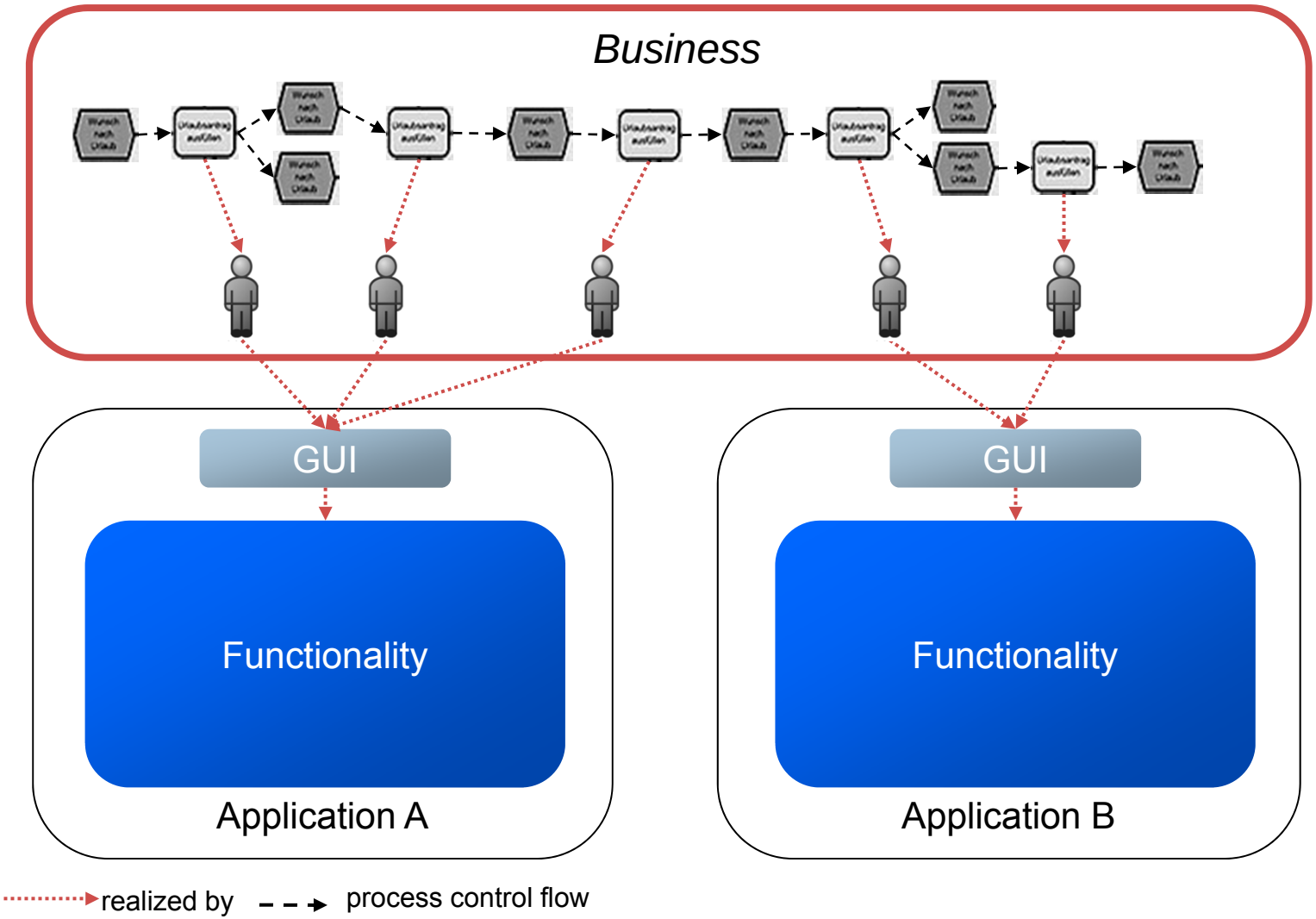
Business meets IT



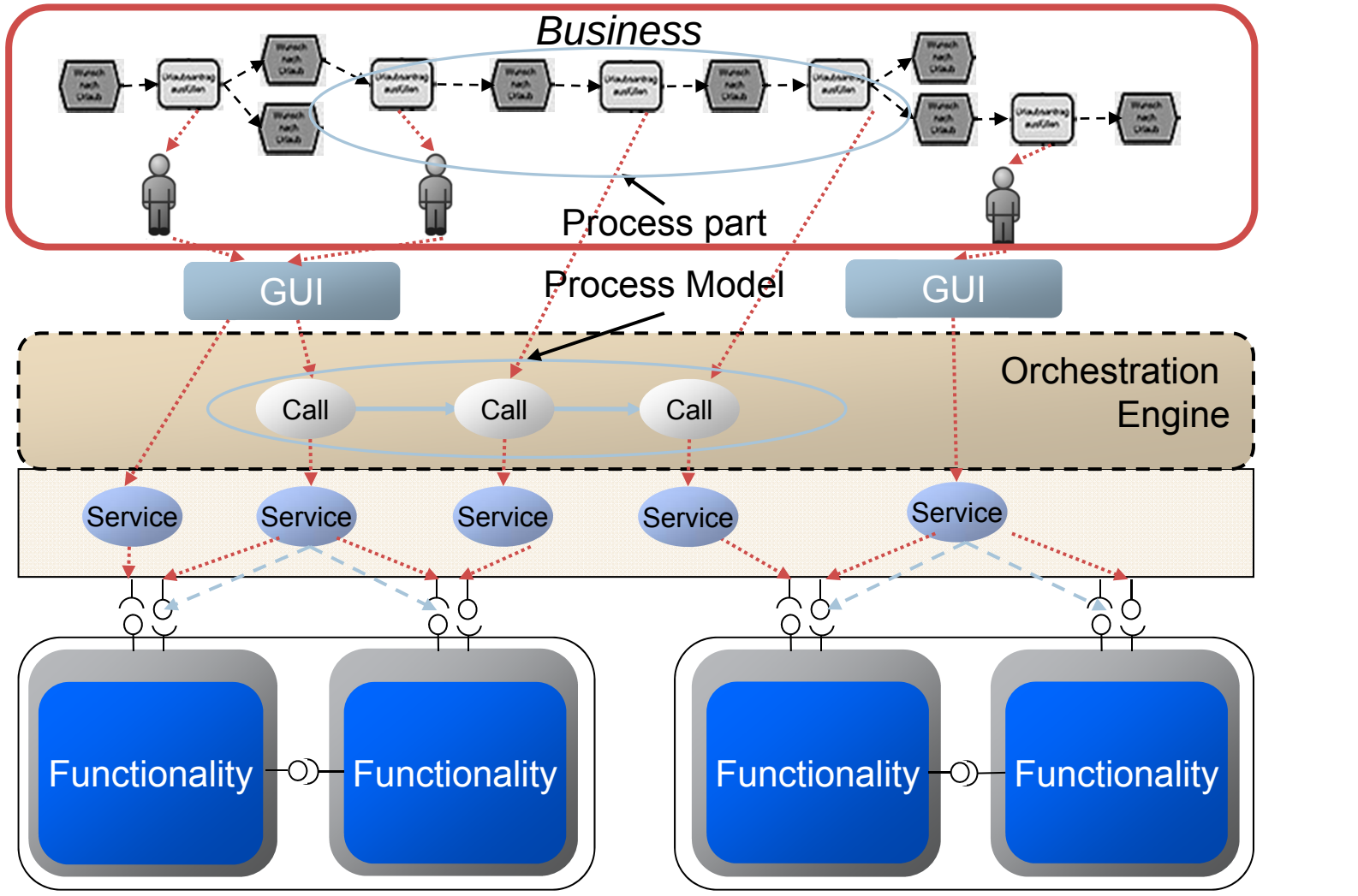
Business meets IT



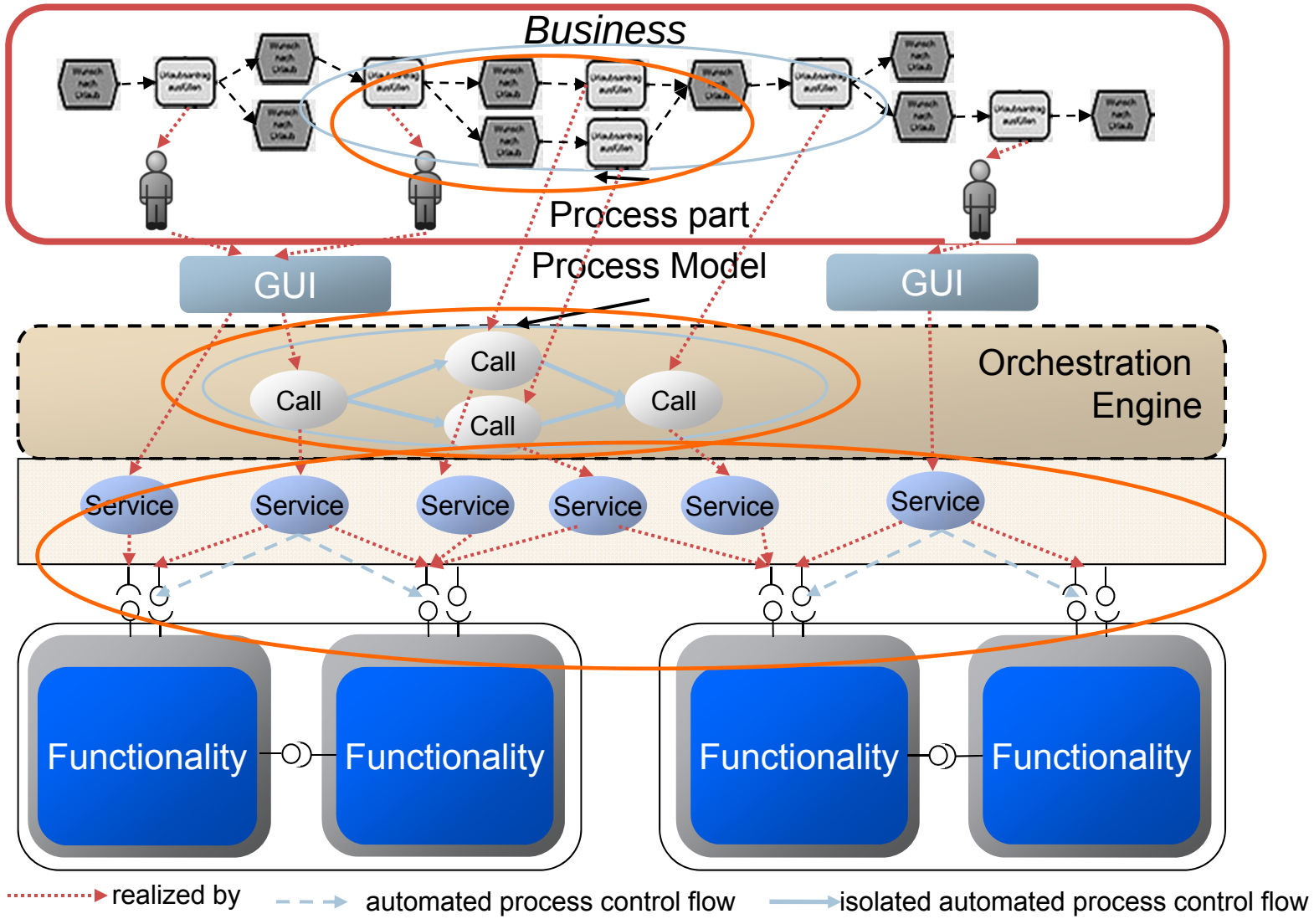
Monolithic Applications



Service Orchestration by Interpretation of Process Models



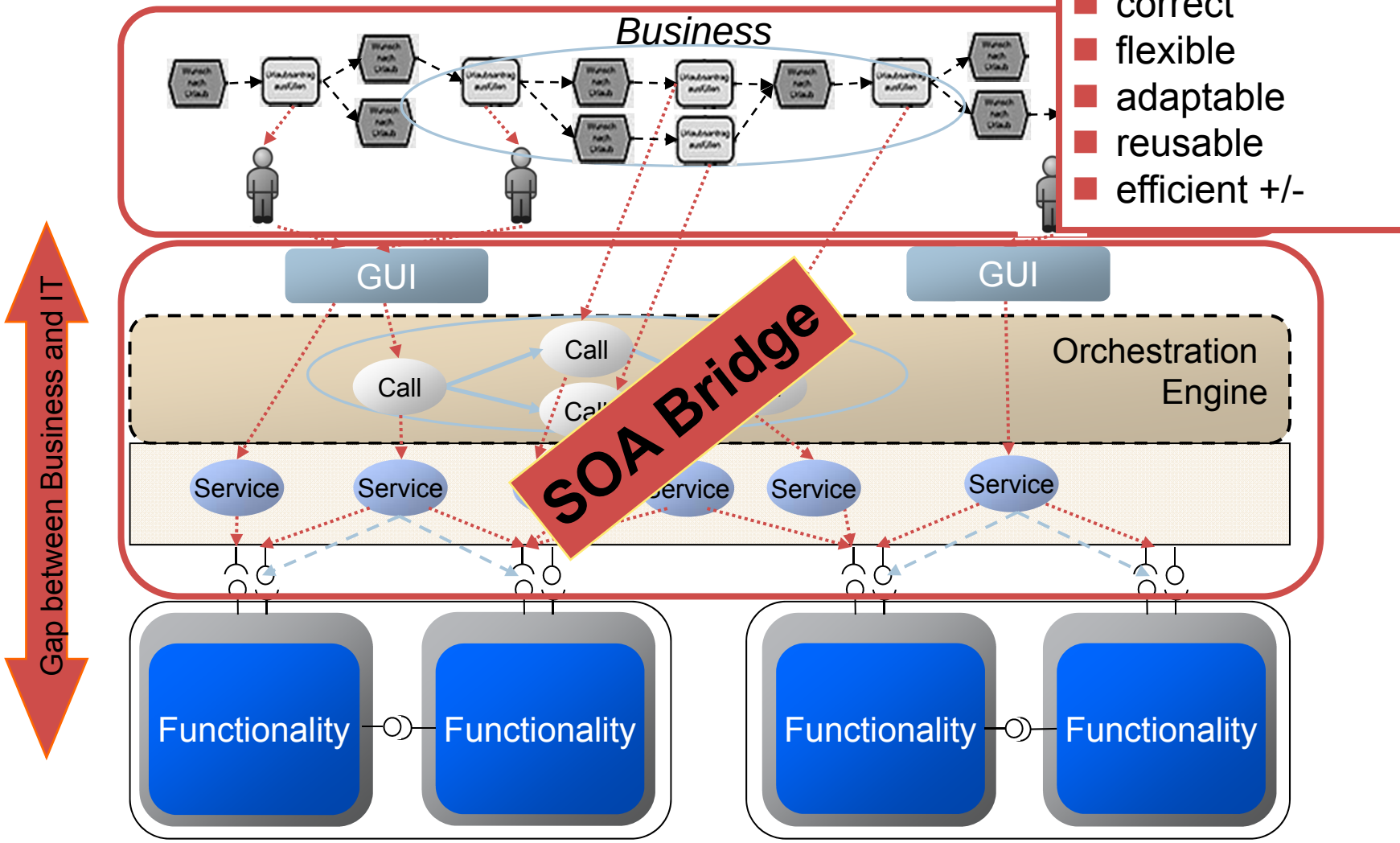
Business/IT-Alignment by Adaptation of Process Models



Service-oriented Architecture (SOA)

Quality Objectives:

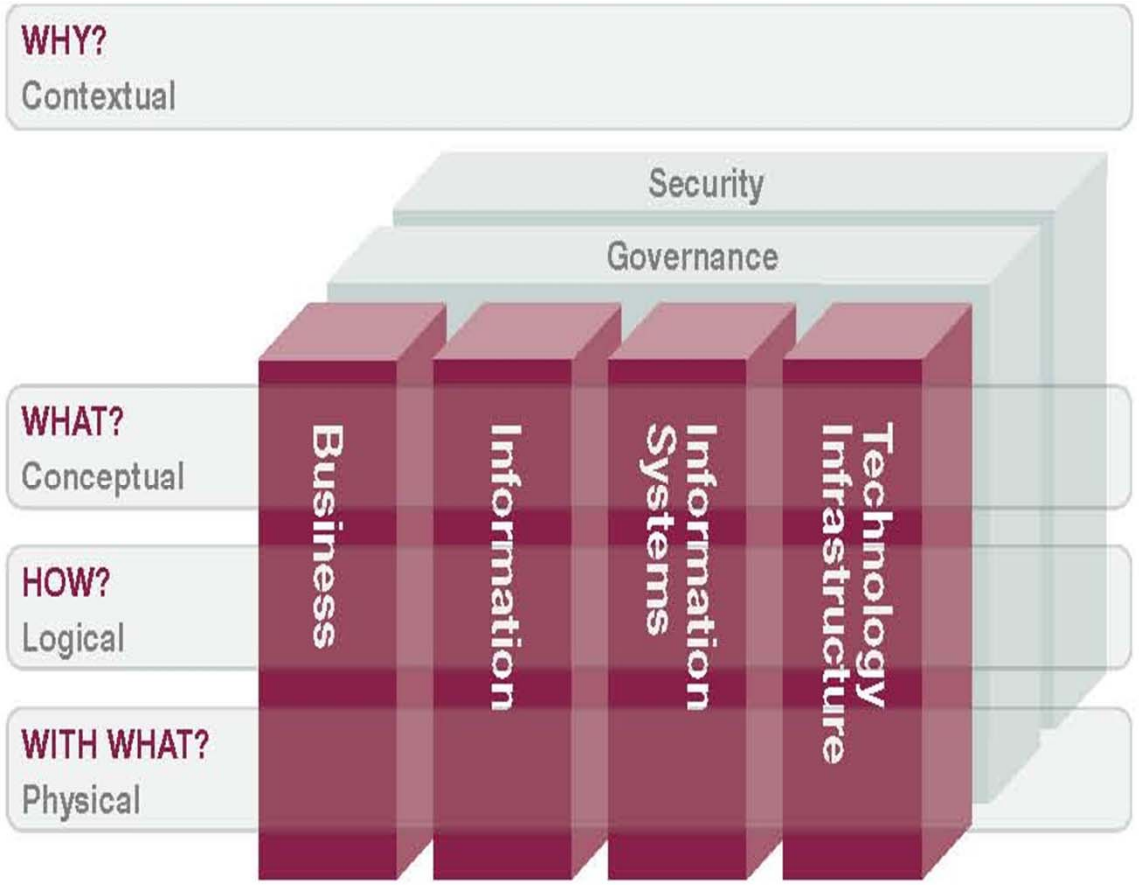
- correct
- flexible
- adaptable
- reusable
- efficient +/-



Research Challenges

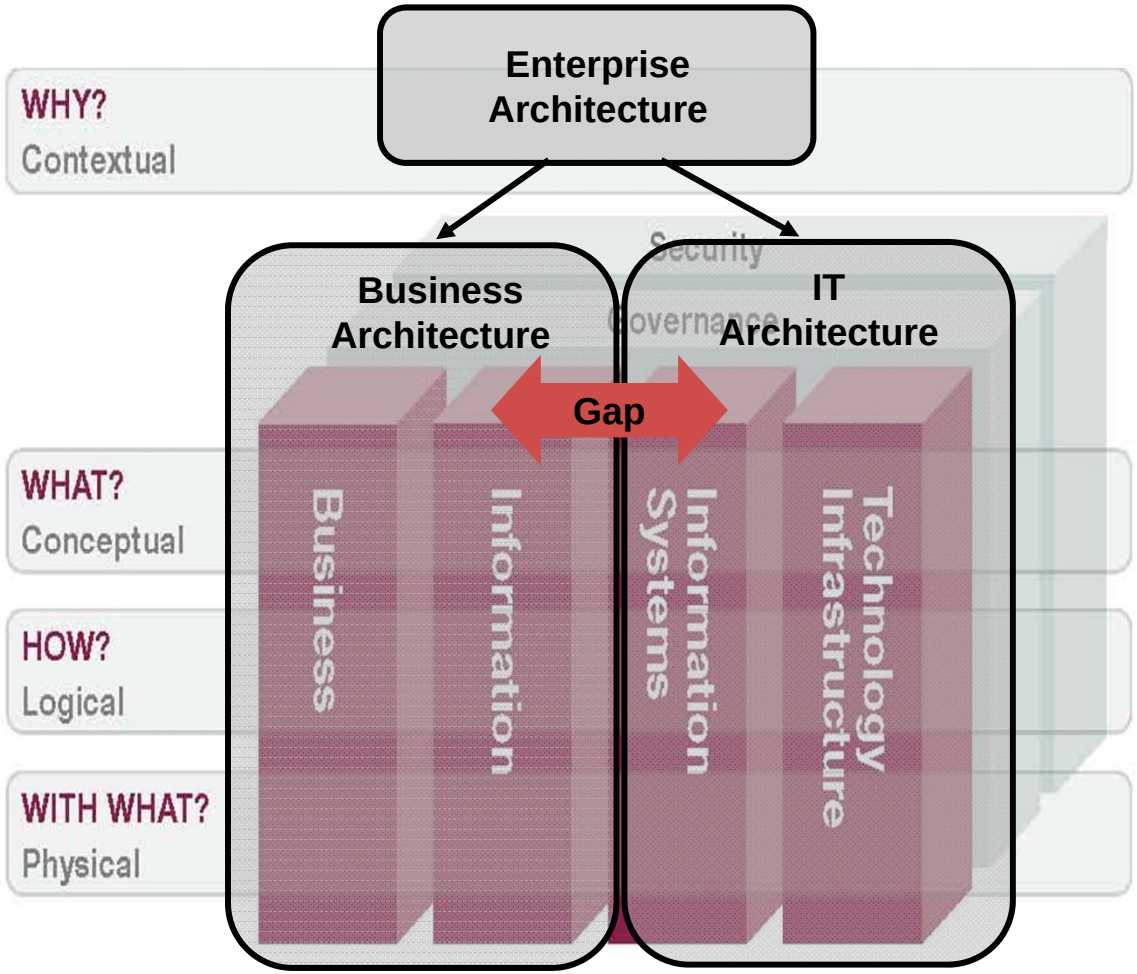
- **Service-orientation** looks like a promising approach to bridge the gap between Business and IT („Business/IT-Alignment“)
- How to find the **right services**?
- We need a **method** for developing and maintaining service-oriented enterprise architectures!

Integrated Architecture Framework (IAF)



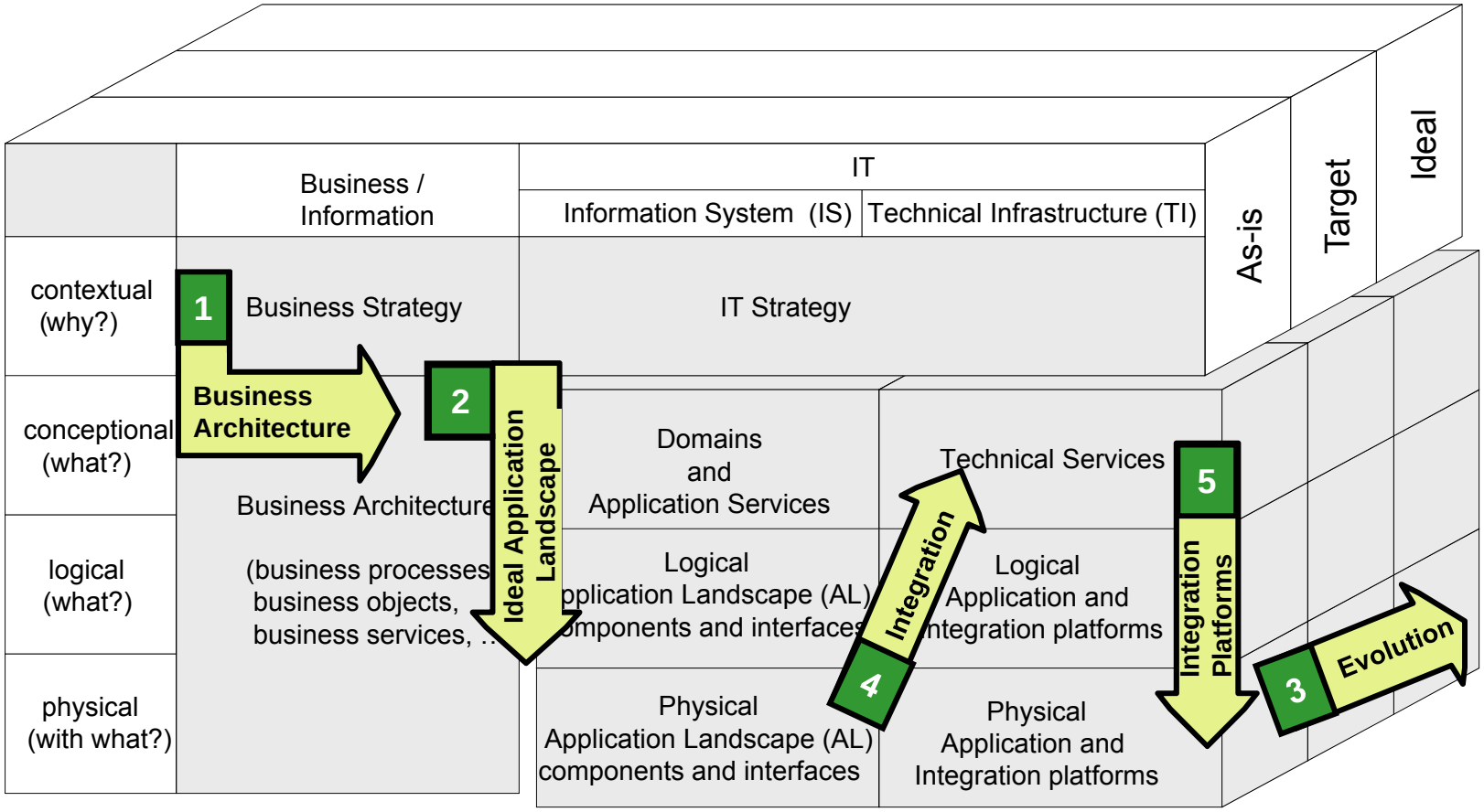
Integrated Architecture Framework (IAF), Capgemini

Integrated Architecture Framework (IAF)



Integrated Architecture Framework (IAF), Capgemini

Quasar Enterprise: Roadmap within Refined IAF Structure

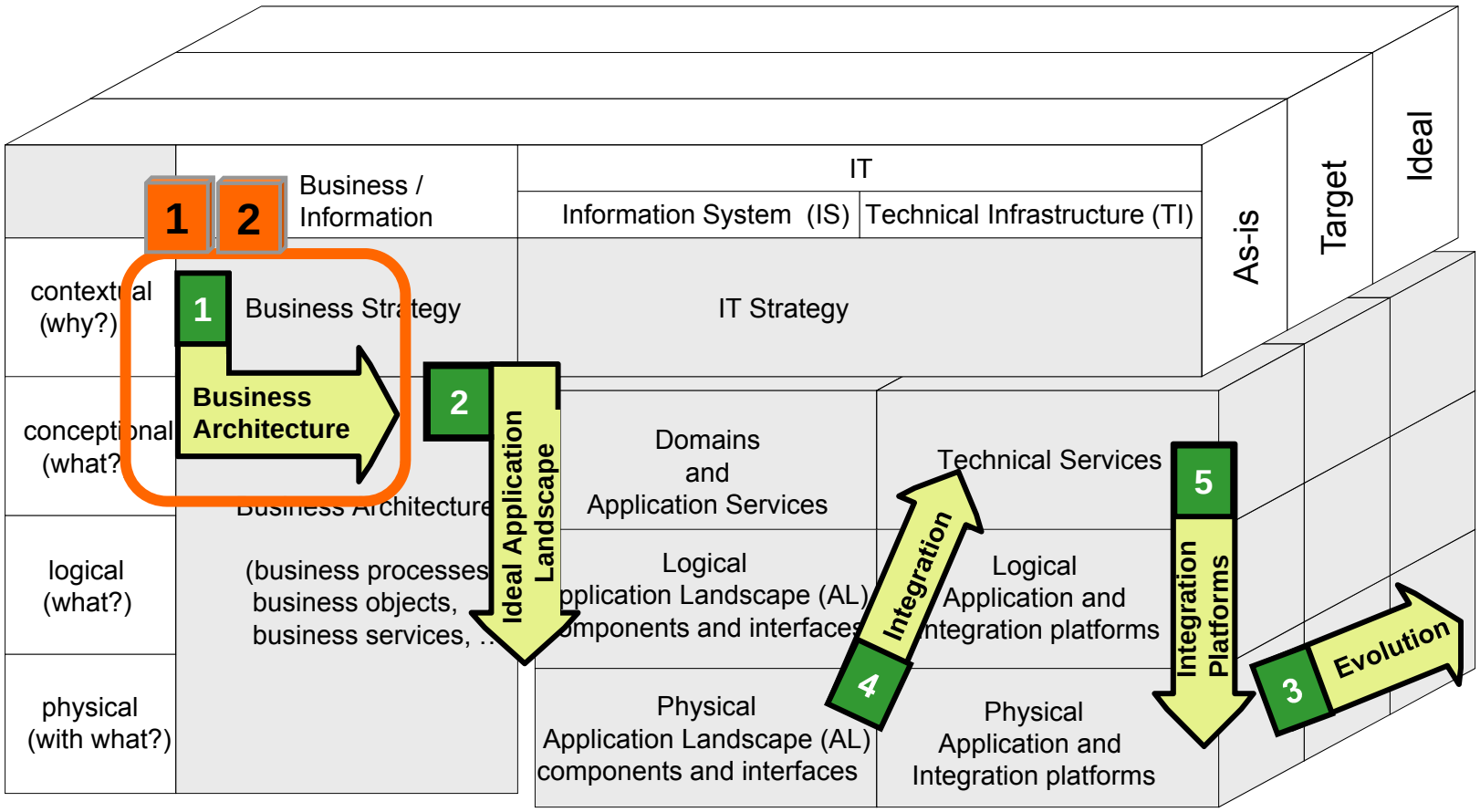


The running example

Christopher Columbus Travel (CCT)

- fictitious travel agency
- sells package travels and (individual) custom travels

Quasar Enterprise: Roadmap within Refined IAF Structure



Quasar Enterprise Method:
Main Step 1 : Analysis of Business Architecture

1 **Derivation of Architectural Guidelines**

2 **Identification and Refinement of Business Services**

Def.

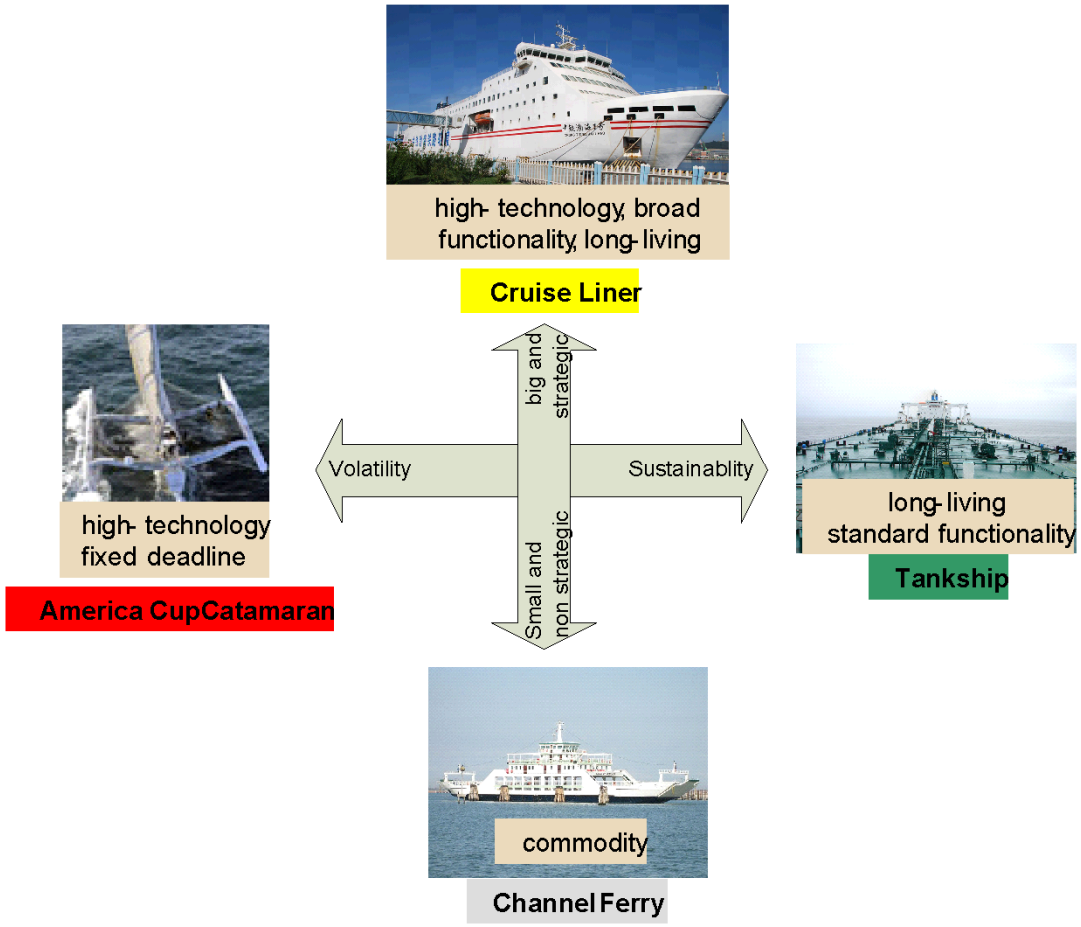
Business Service

- behavioural element to fulfill a business need
- delivered by a service provider for a service requestor
- described by a contract (in-/output, observable business service actions)

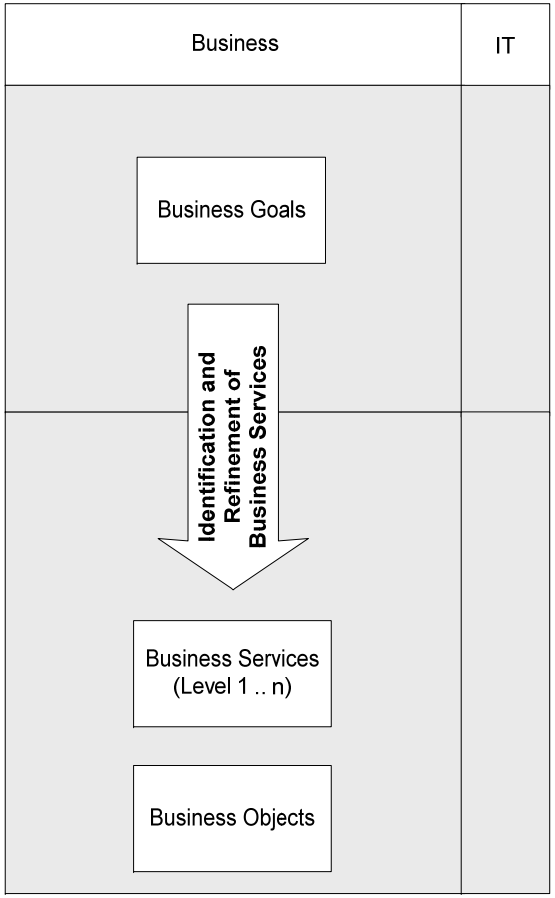
Quasar Enterprise Method:
Main Step 1 : Analysis of Business Architecture

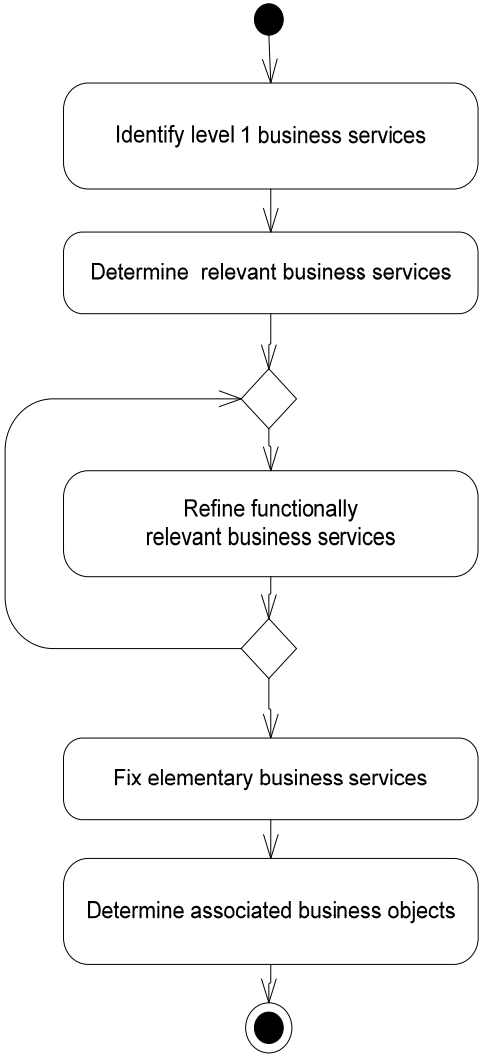
1 Derivation of Architectural Guidelines

eee@Quasar
Effectiveness, Efficiency and Economics of Price



Main Step 1: Analysis of Business Architecture
2 Identification and Refinement of Business Services





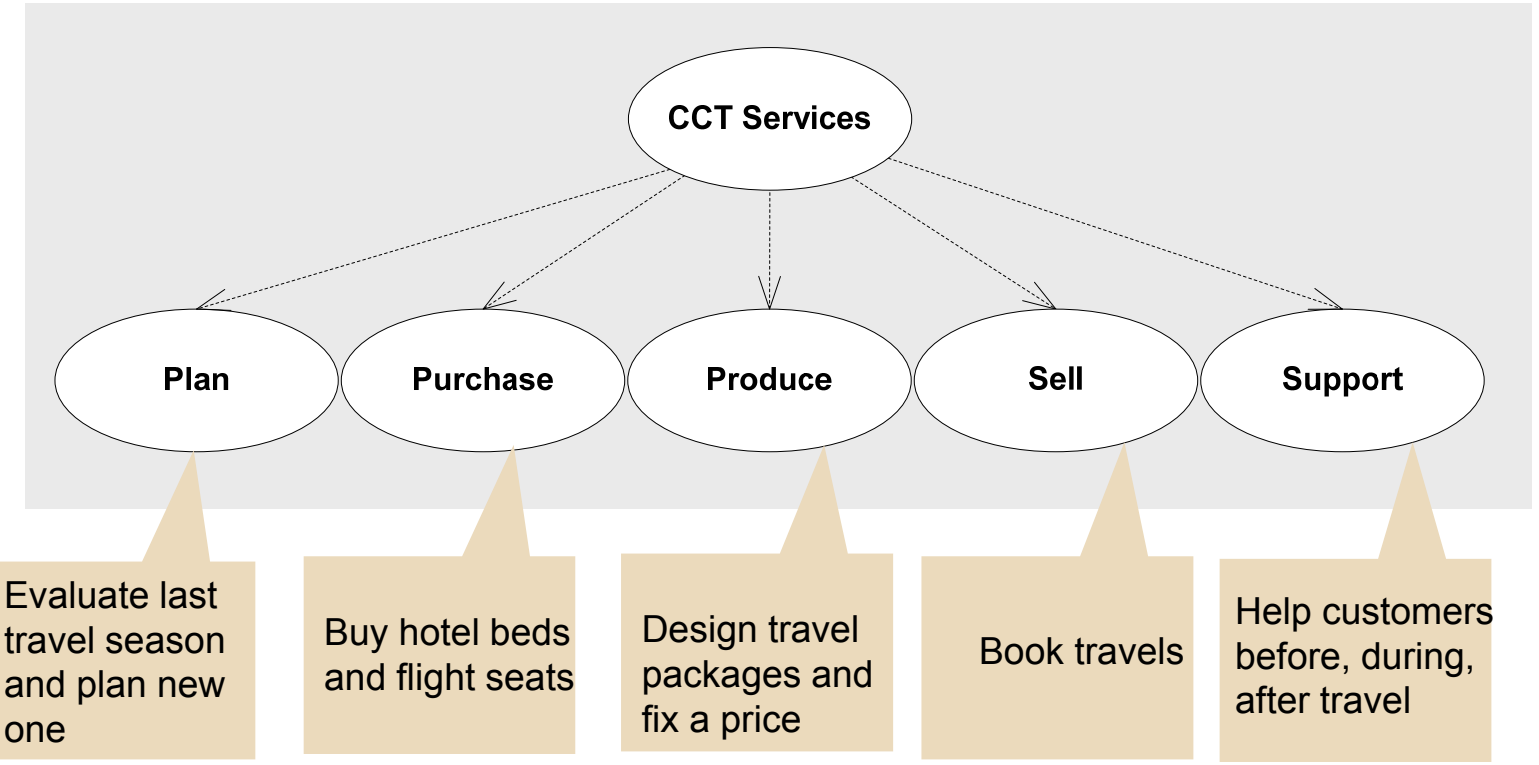
2 : Example

Identify level 1 business services

Def.

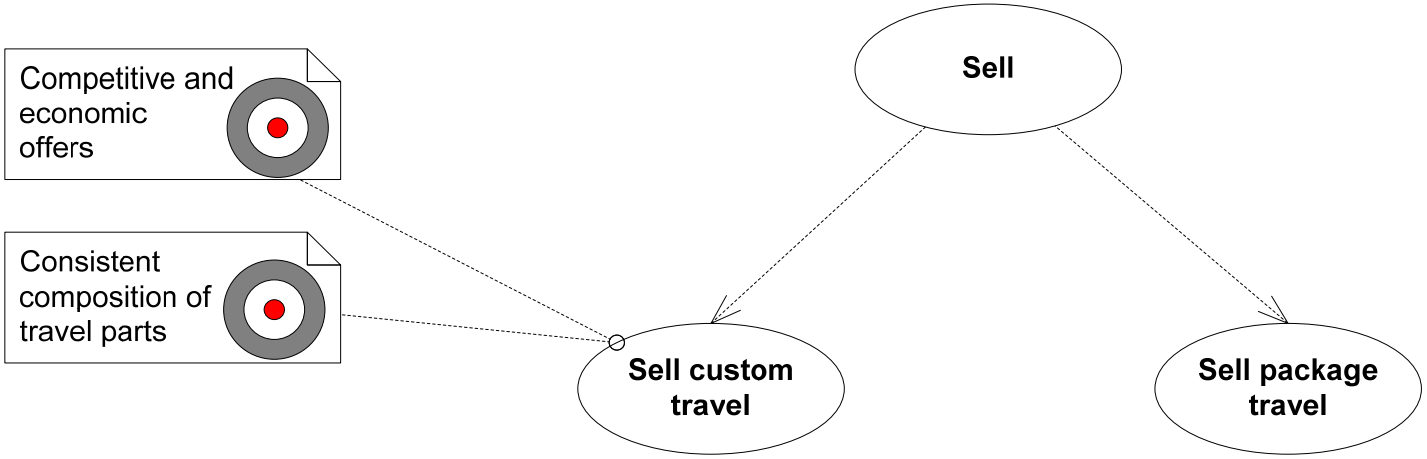
Level 1 business services

- Internally and externally offered core services of an enterprise to fulfill its business objectives.



2 : Example
Refine business services

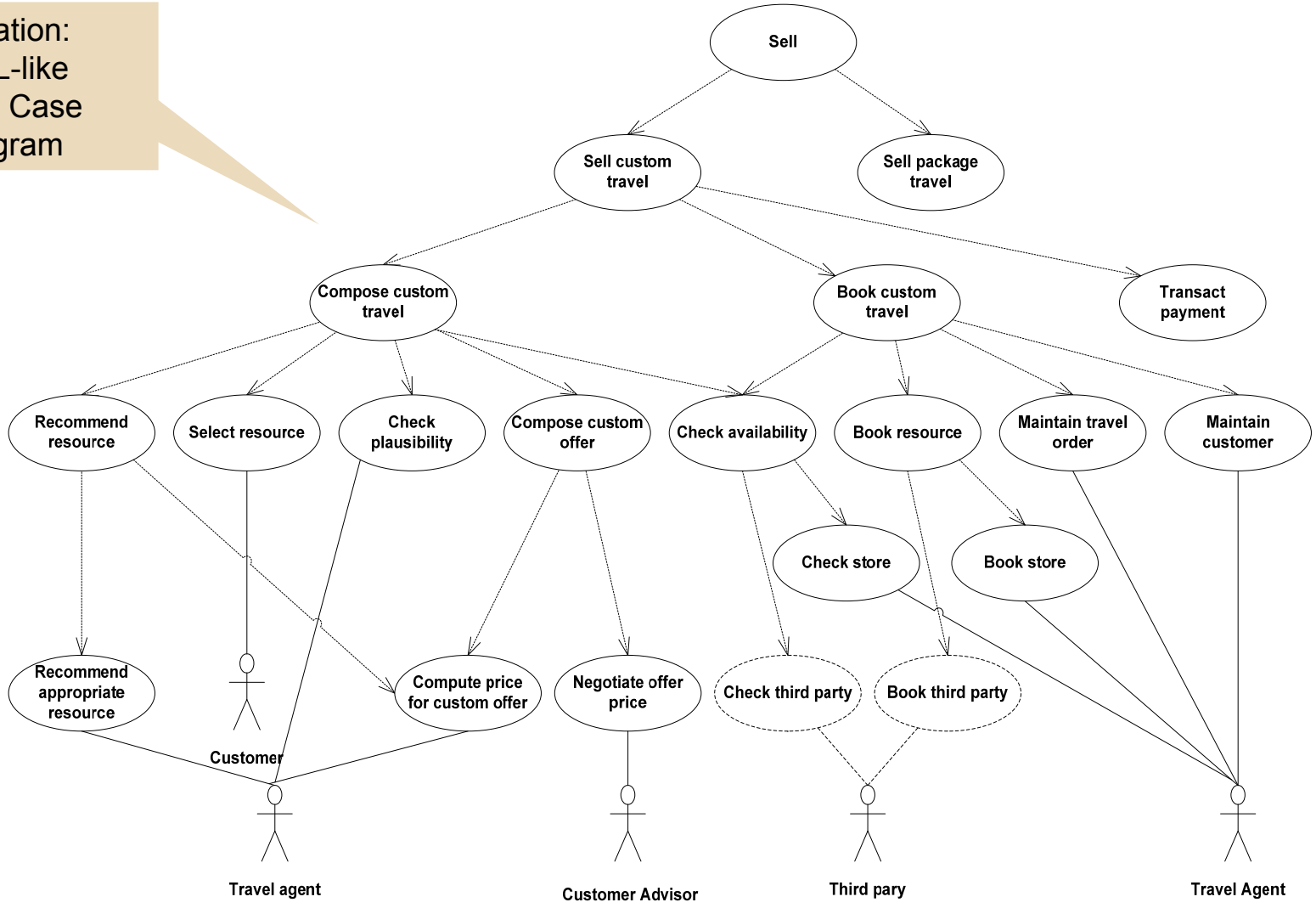
- Functional decomposition in case of
  - multiple actors for a service exist
  - multiple, diverse business goals are supported
  - different product types have to be supported
  - …



2 : Example

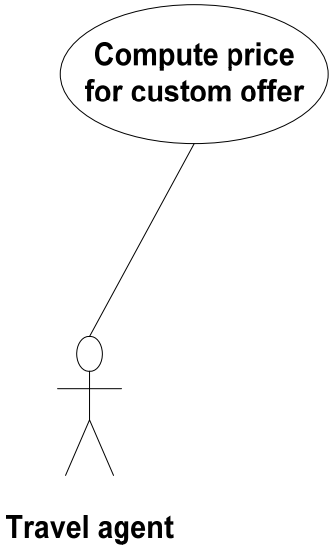
Fix elementary business services

Notation:
UML-like
Use Case
Diagram



2 : Example

Contract description of elementary business service

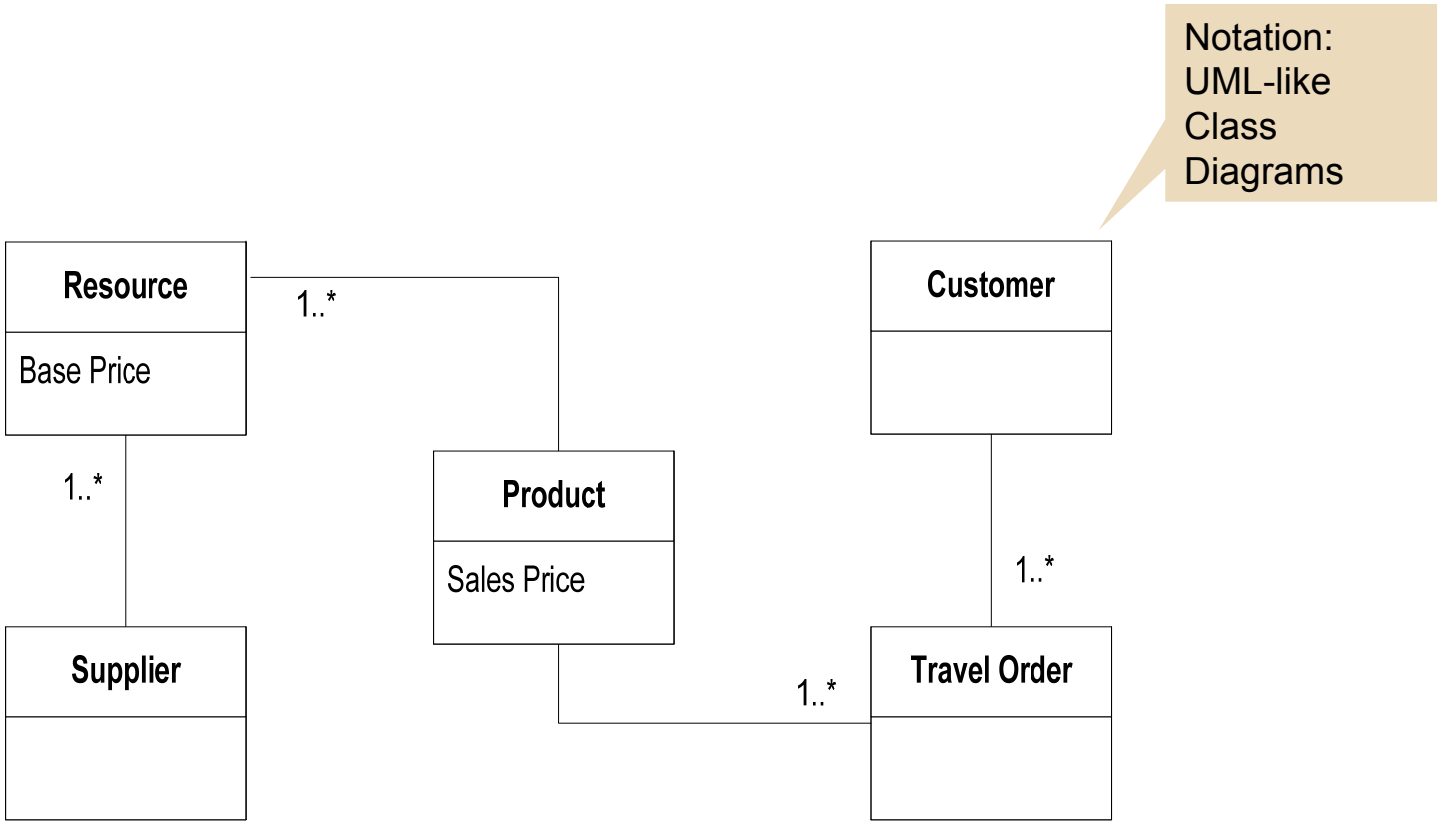


Name	**Compute price for custom offer**
Service user	***Travel agent***
Trigger / Preconditions	***Price request by travel agent. Custom travel as product already composed, its plausibility has been checked.***
Actions and Service Protocol	***no protocol, as service consists of exactly one action***
Result / Postconditions	***Lump sum price for complete custom travel is computed (in Euro). All possible reductions are taken into account.***
Non-functional requirements	***Response time < 1 s***

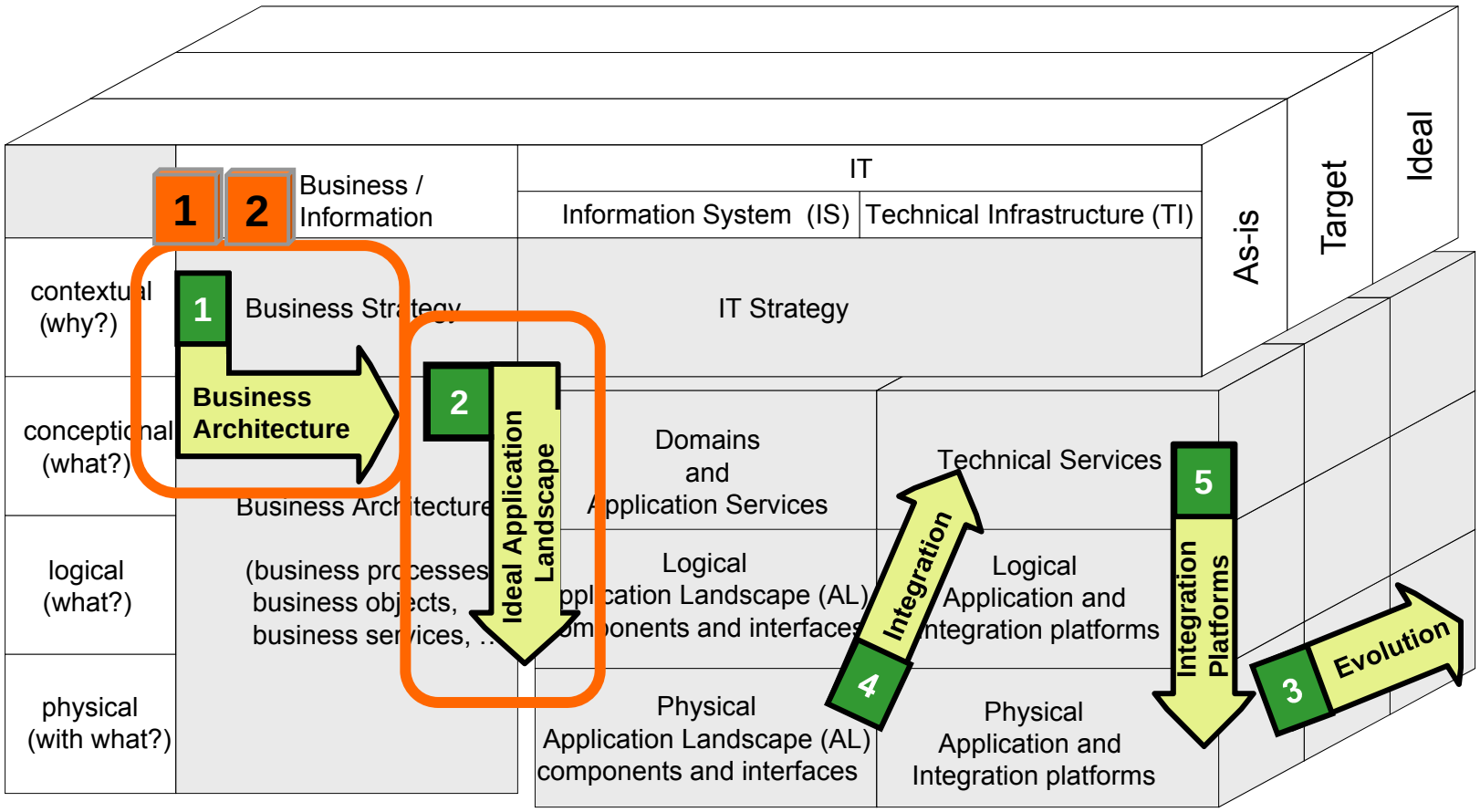
2 : Example

Determine associated business objects

- Derived from business services:
  required and provided business objects as in-/outputs of services



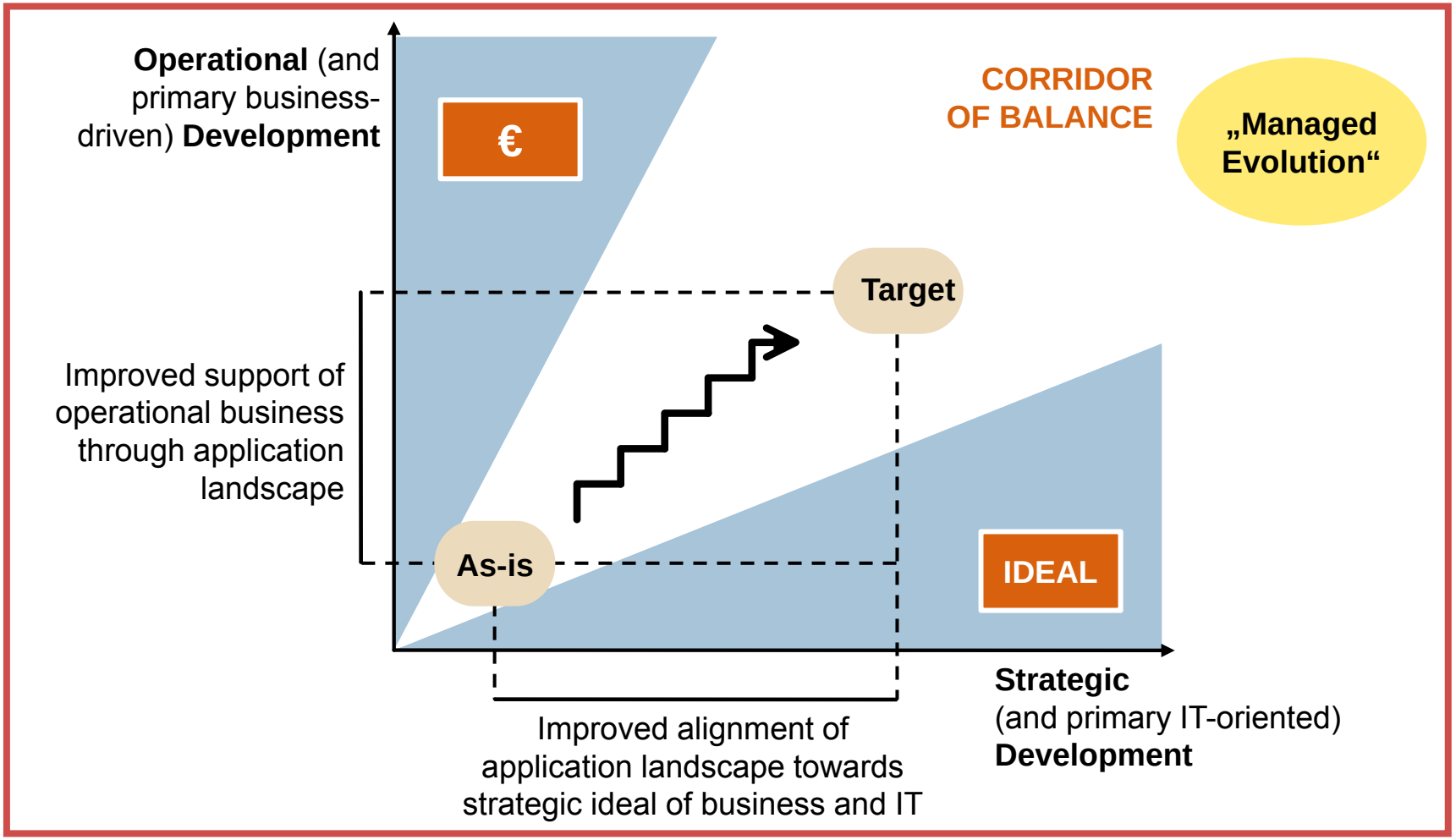
Quasar Enterprise: Roadmap within Refined IAF Structure



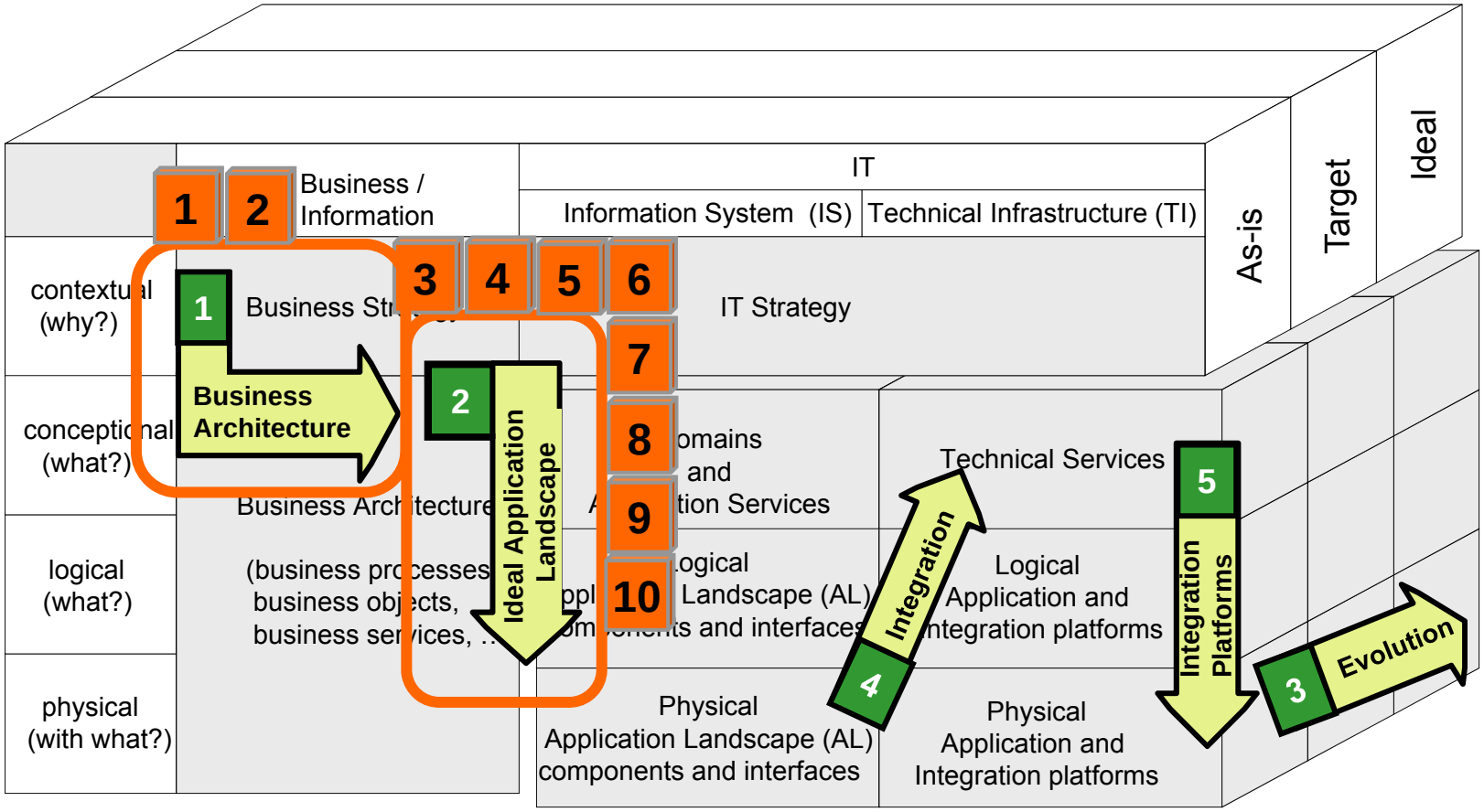
Quasar Enterprise Method:

Why ideal?

Main Step 2 : Definition of an Ideal Application Landscape



Quasar Enterprise: Roadmap within Refined IAF Structure



Quasar Enterprise Method:
Main Step 2: Definition of an Ideal Application Landscape

3 **Designing Domains**

4 **Identification of Application Services**

5 **Designing Components**

6 **Reference Architecture Categorized Application Landscape**

7 **Design Rules for Components**

8 **Designing Interfaces and Operations**

9 **Design Rules for Interfaces and Operations**

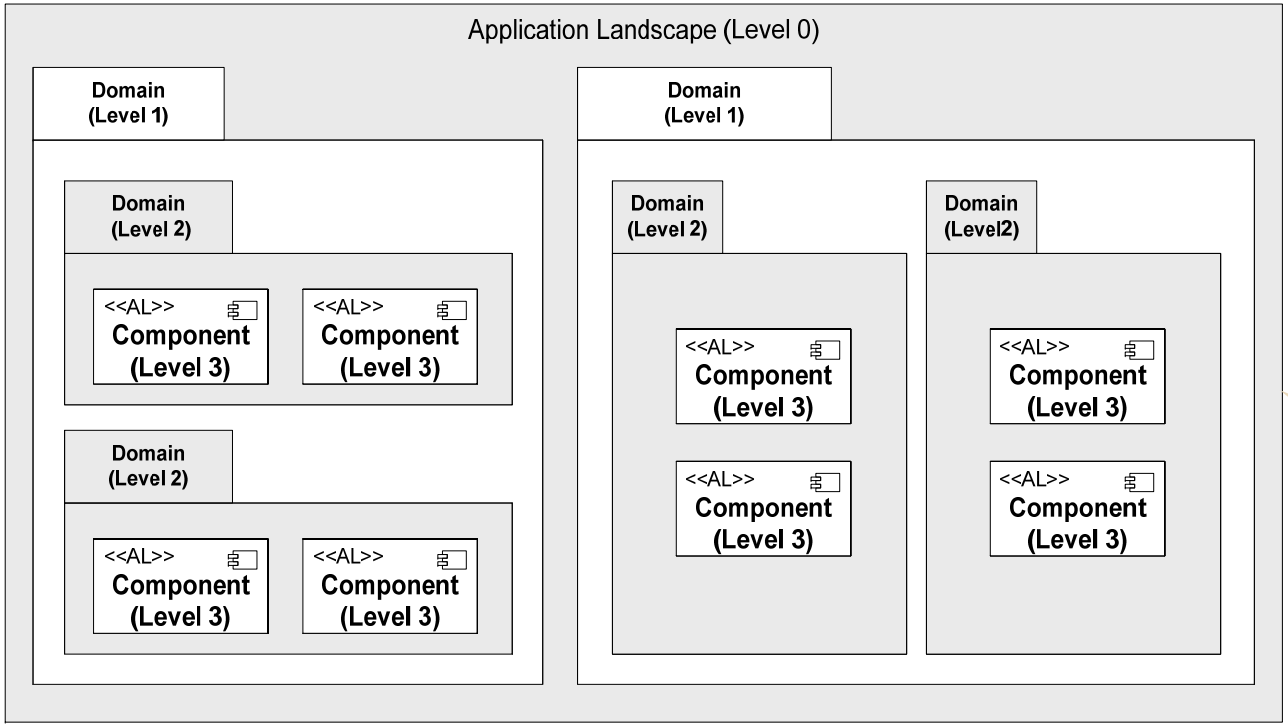
10 **Design Rules for Coupling Architecture**

Main Step 2: Definition of an Ideal Application Landscape

3 Designing Domains

Domain

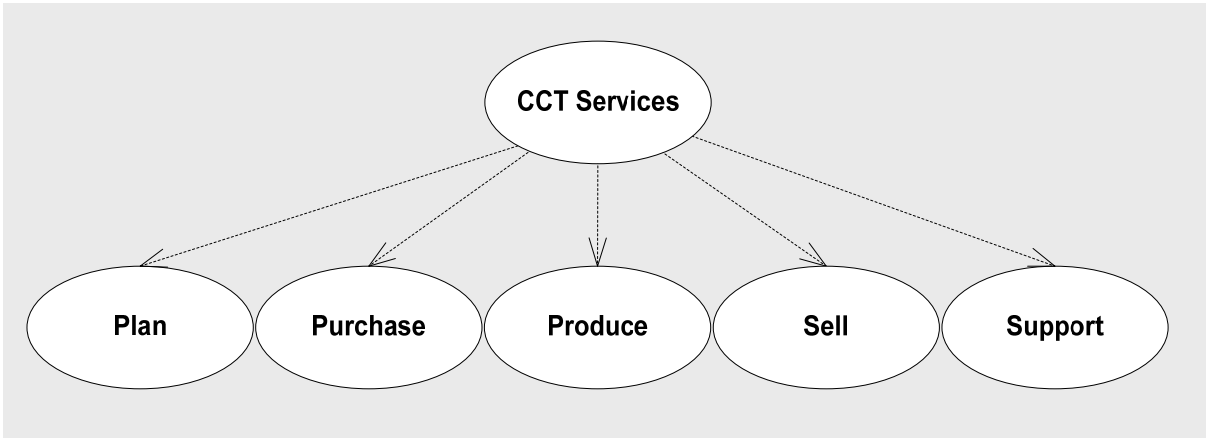
- business-oriented clustering of components of an application landscape
- may be hierachically structured
- components are associated to leaf domains

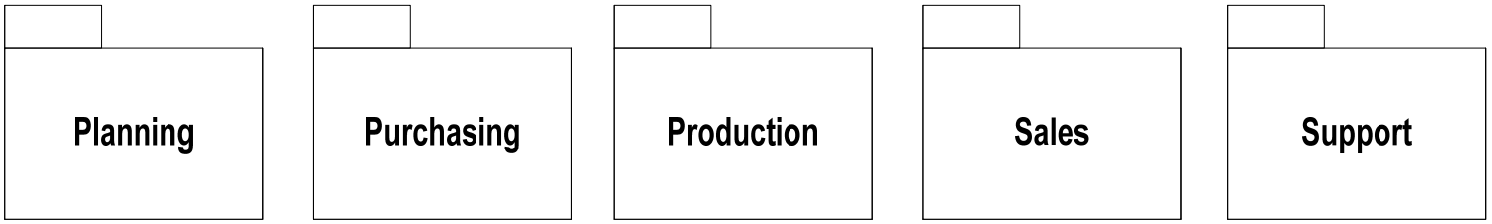


Notation: UML-like Package and Component Diagrams

3 : Example

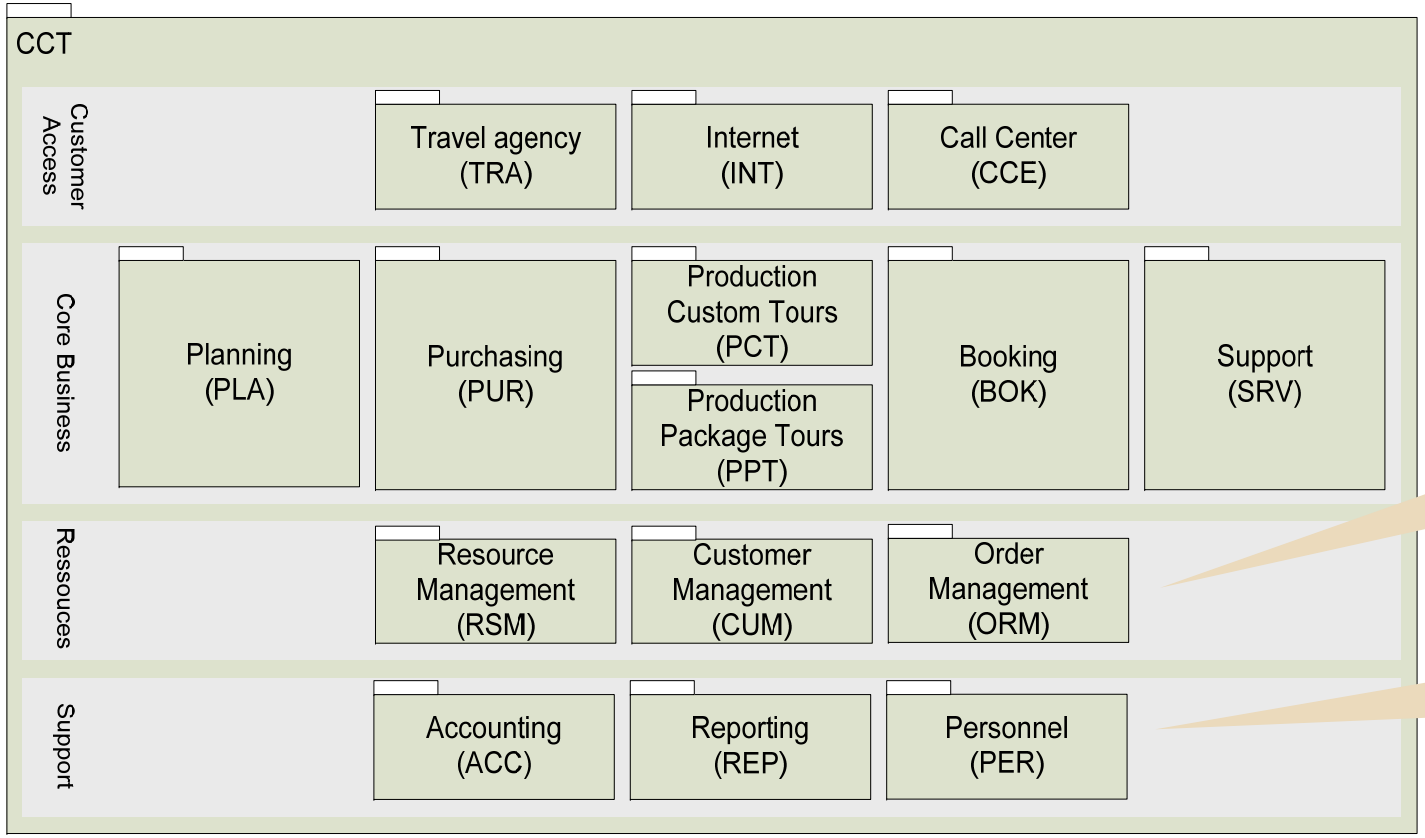
Core business services are candidate domains



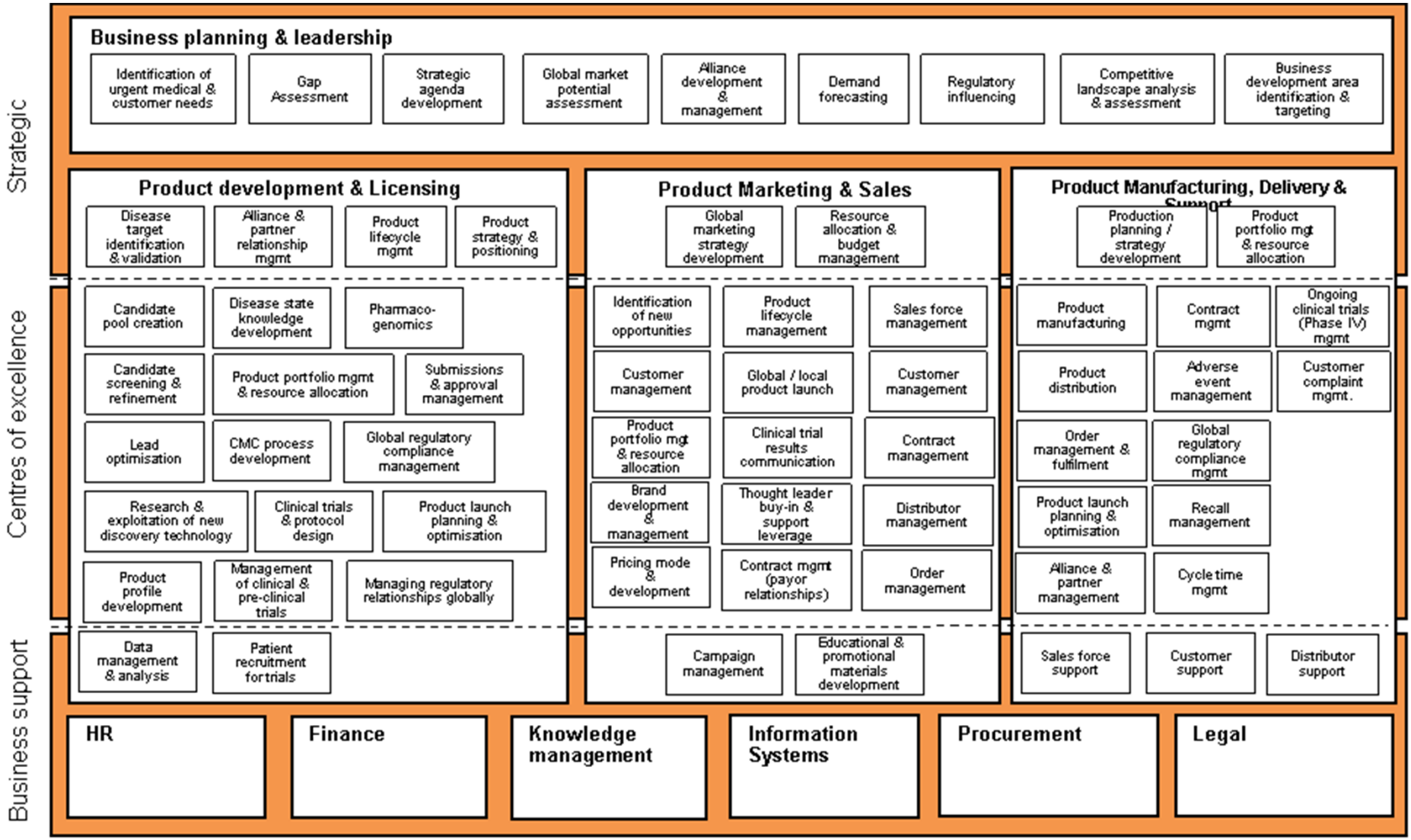


3 : Example

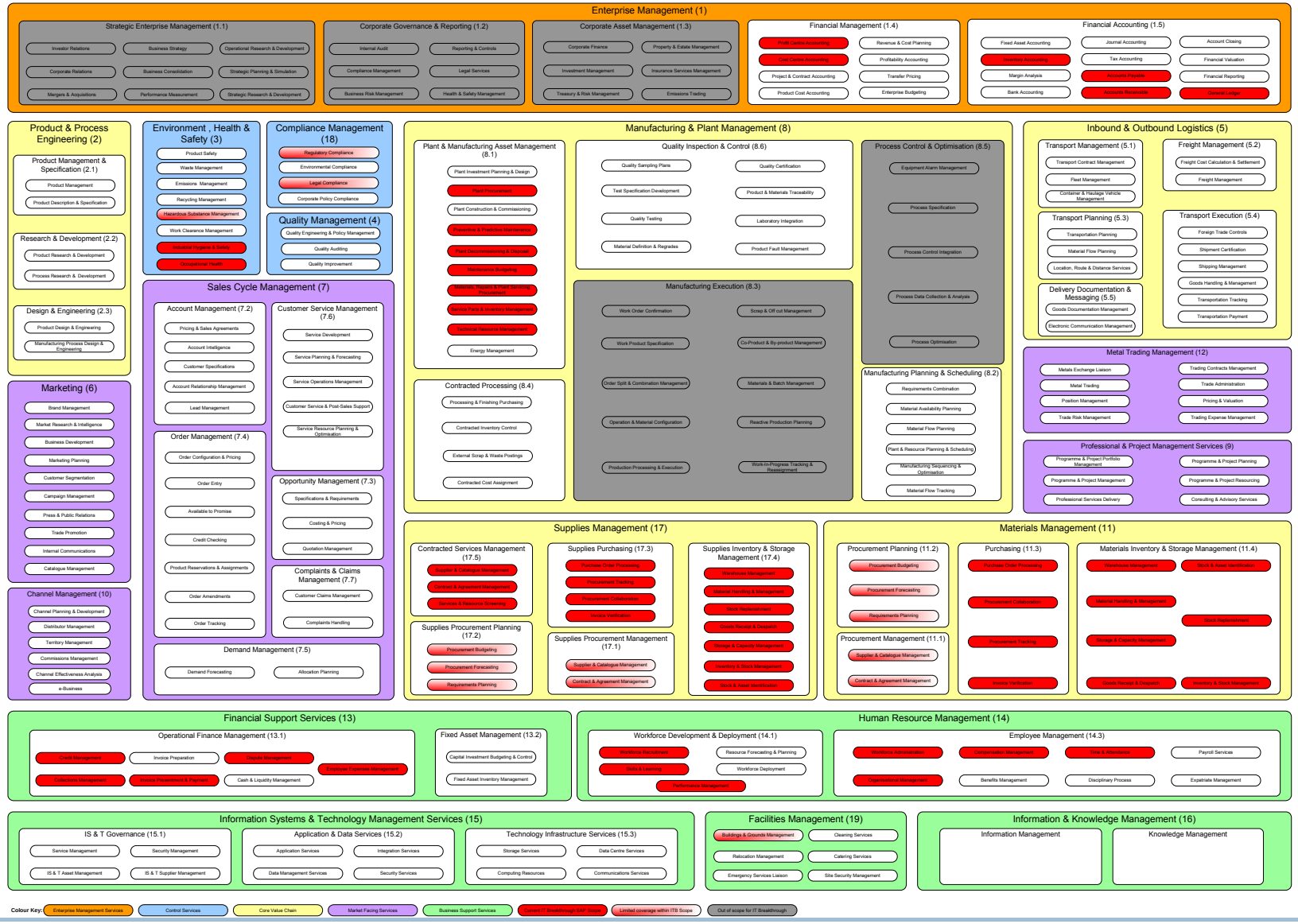
Final Domains of Christoper Columbus Travel



Real Example: Life Sciences

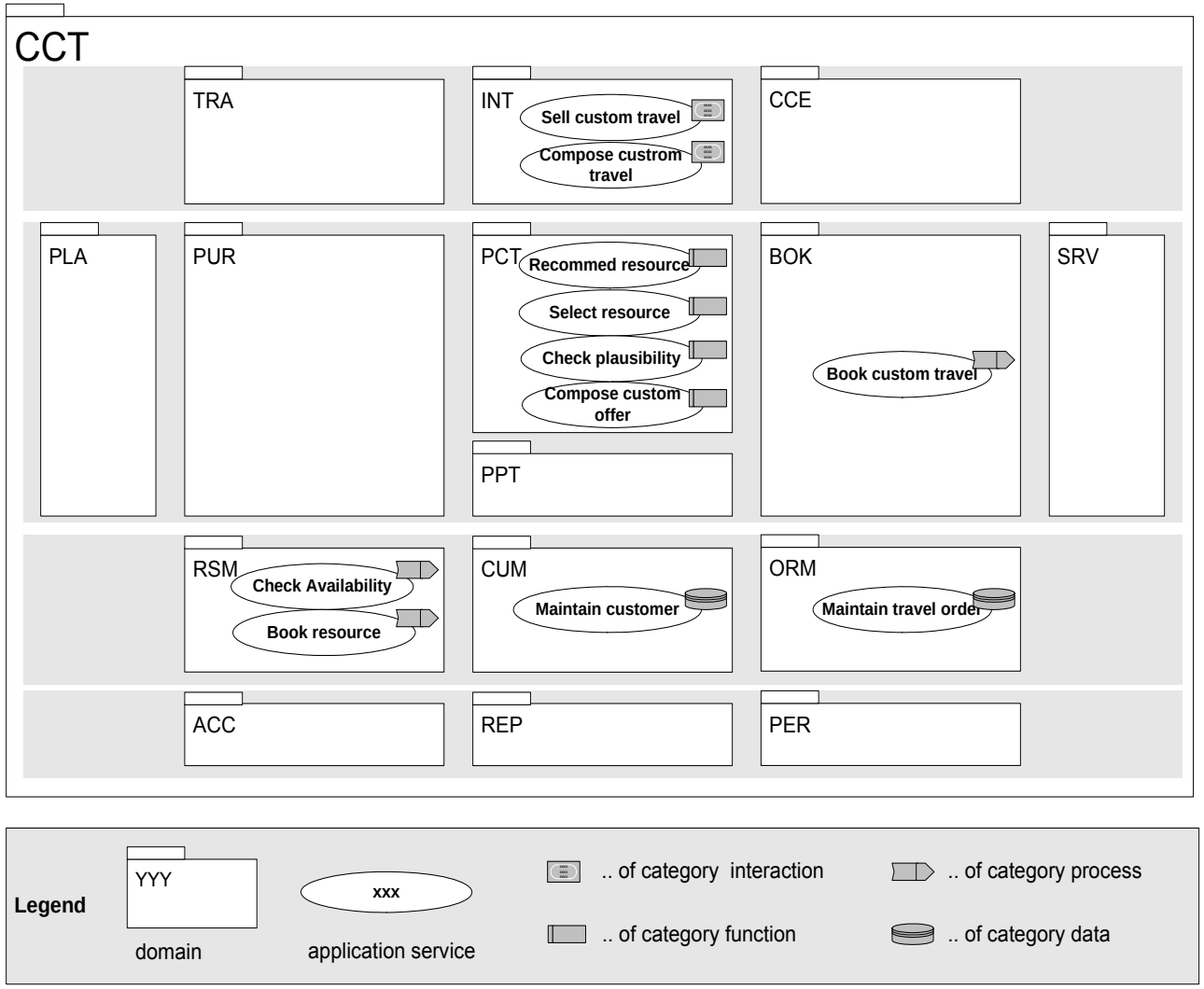


Real Example: Manufactoring



4 : Example

Domains with associated categorized application services



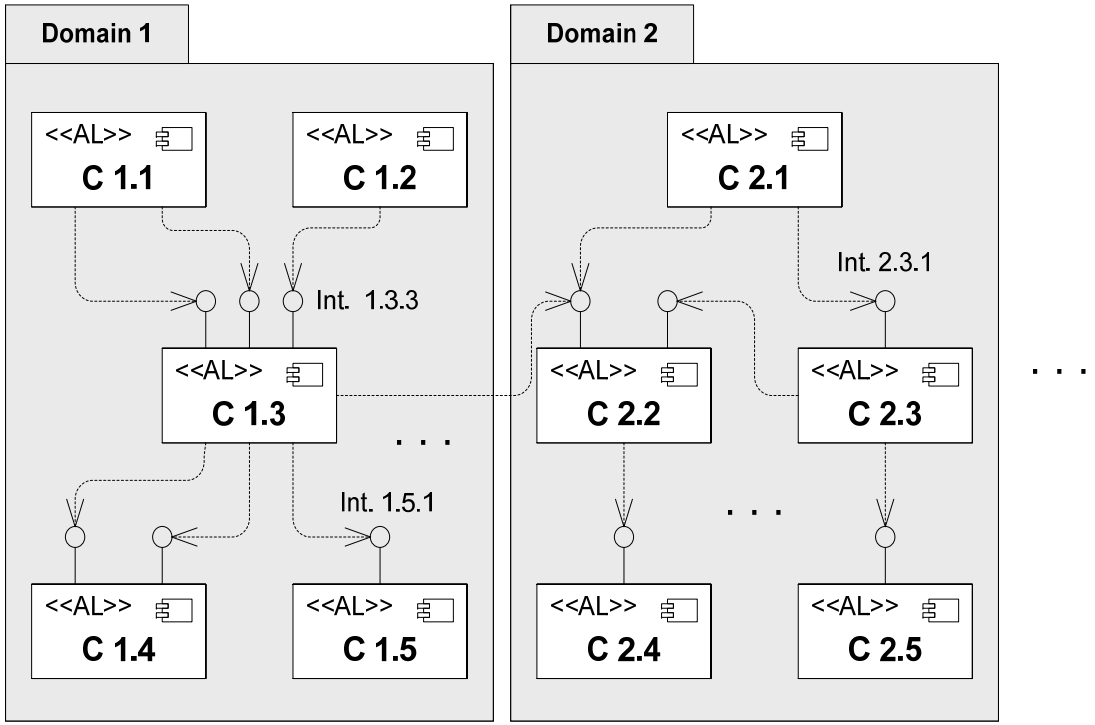
Main Step 2: Definition of an Ideal Application Landscape

5 Designing Components

Def.

Application Landscape (AL) component

- realizes application services
- has explicit interfaces for provided and requested operations
- is coupled with other AL components and uses their provided operations



5, 7 : Example

Determine candidate components and refine

Candidate component

- All application services of the same domain and category become a candidate component

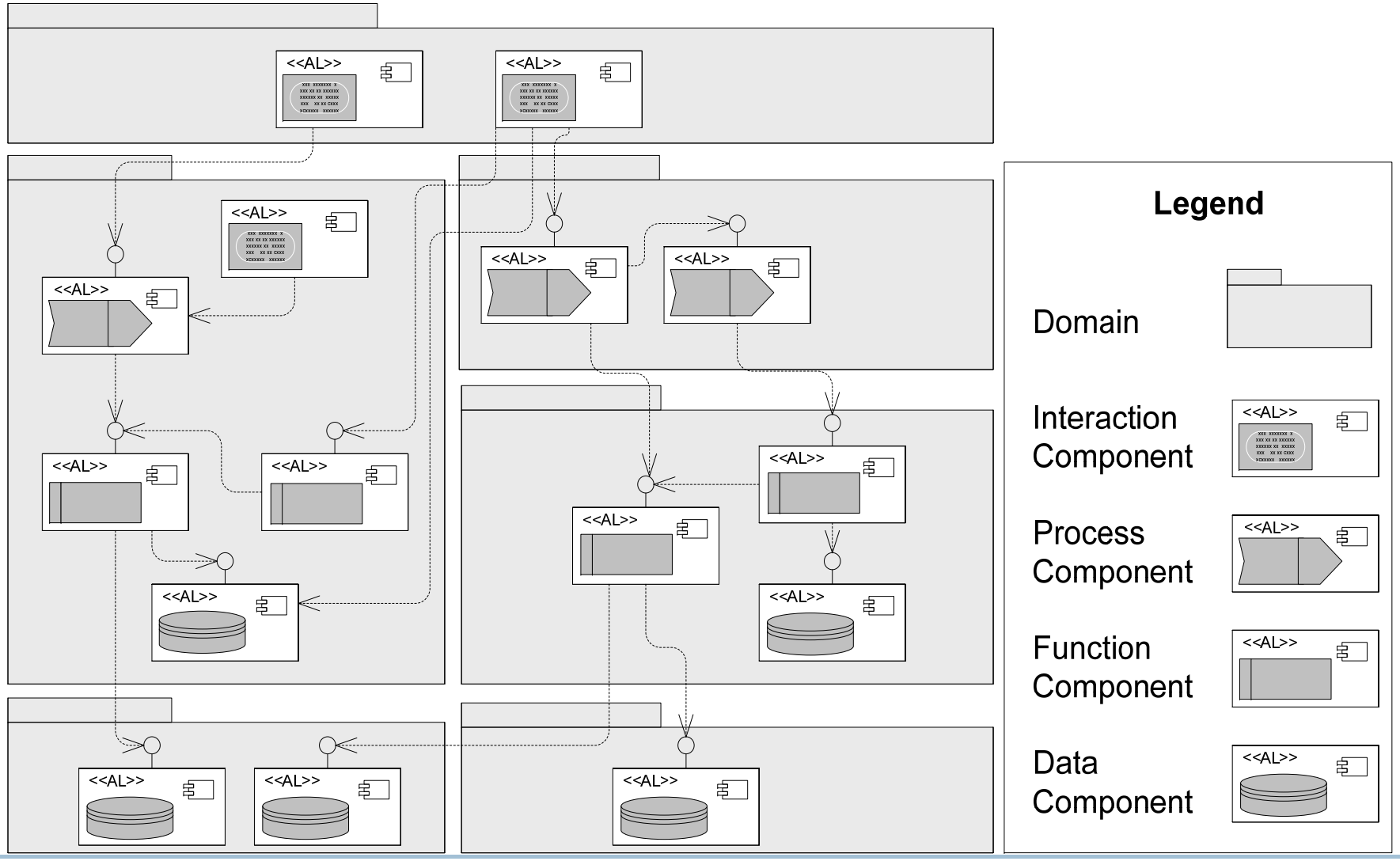
7

Design Rules for refining candidate components

- An AL component belongs to exactly one domain.
- All operations of an AL component shall be of the same category.
- Business logic that changes at a different pace shall be separated.
- AL components of category data have responsibility of business objects.
- AL components shall not have cyclic dependencies.
- AL components of different categories shall have layered dependencies according to interaction → process → function → data.
- AL components shall have low coupling and high cohesion.
- …

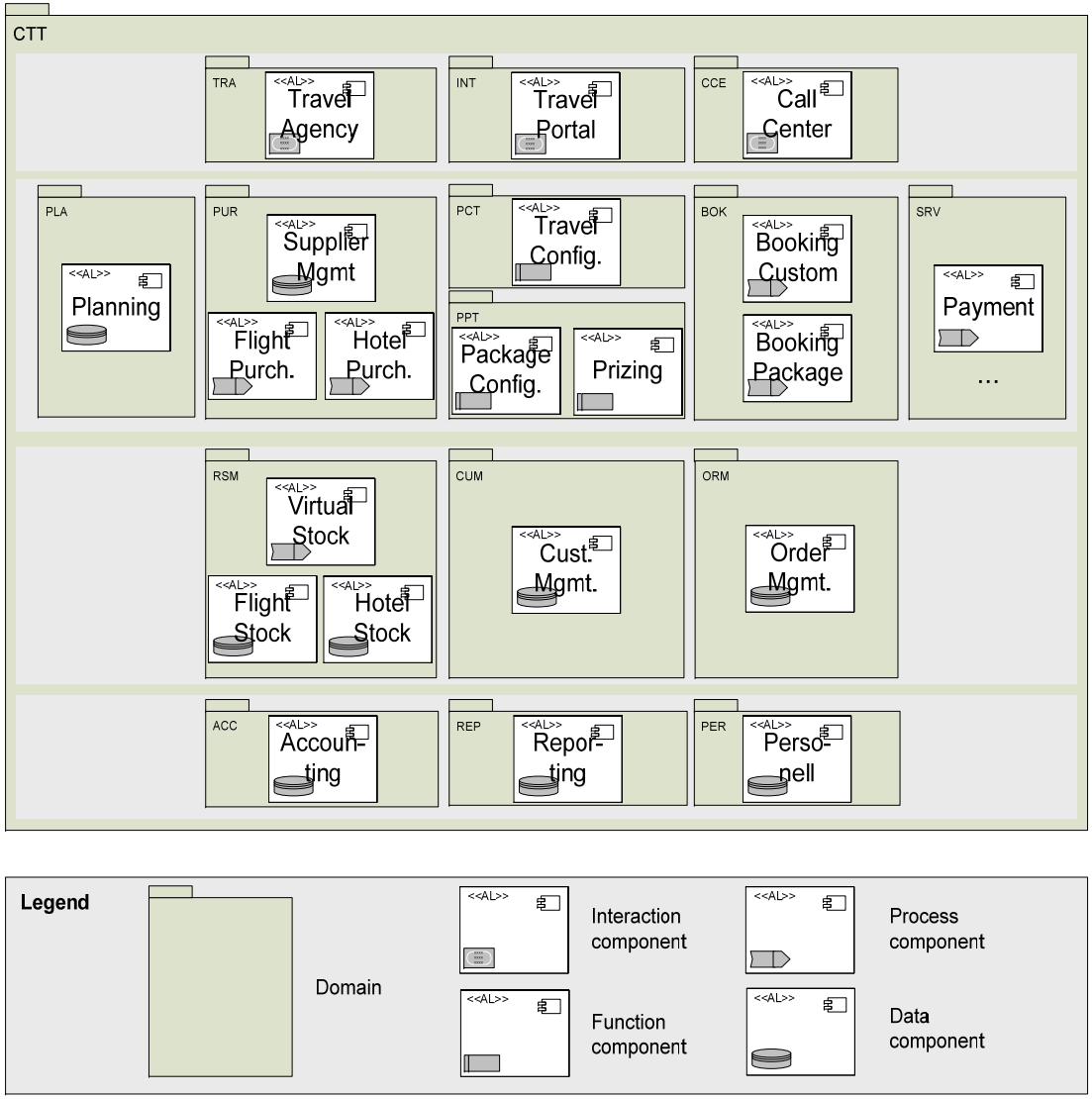
Main Step 2: Definition of an Ideal Application Landscape

6 Reference Architecture Categorized Application Landscape



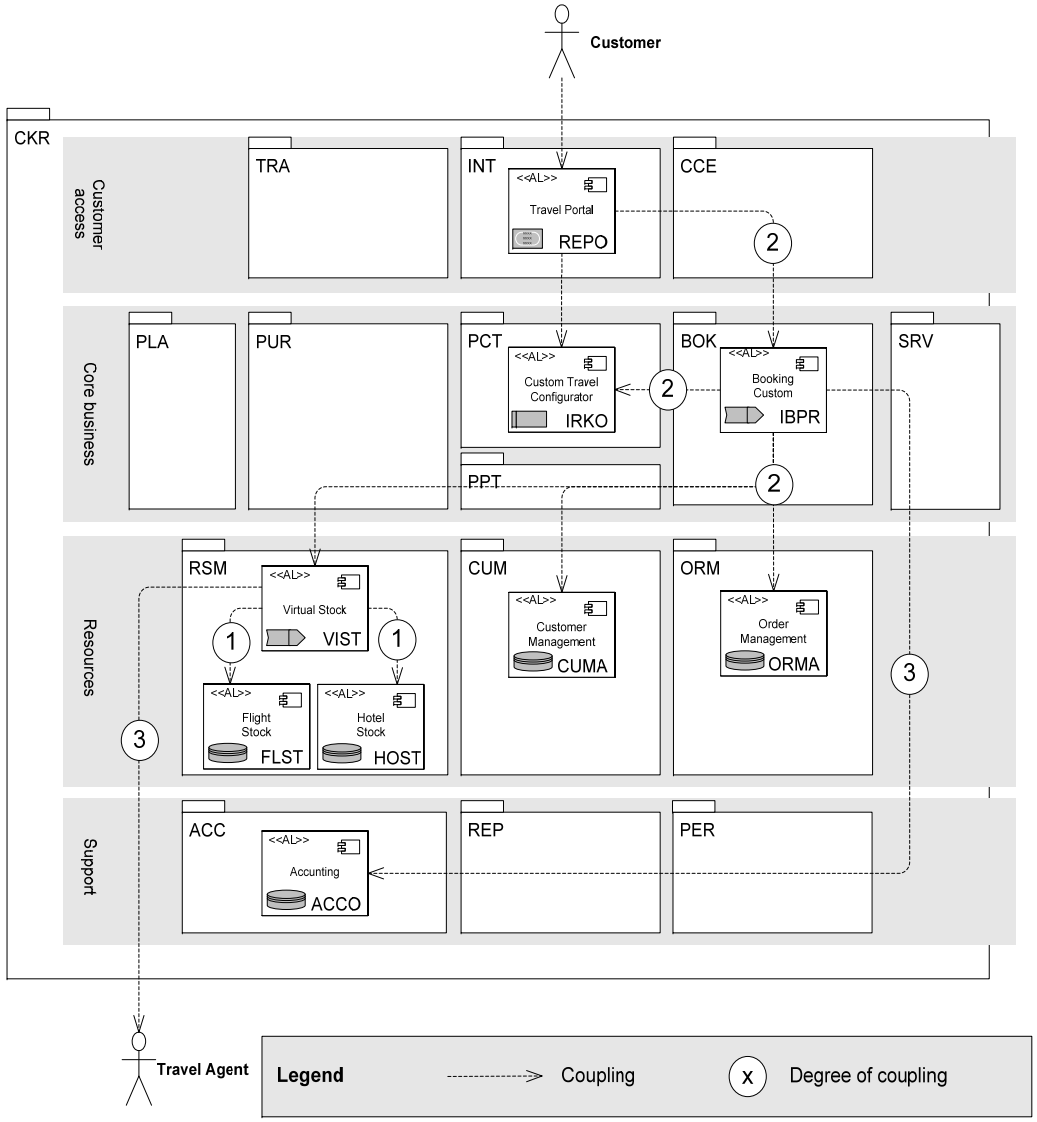
5, 6, 7 : Example

Final AL components of CCT



Main Step 2: Definition of an Ideal Application Landscape

10 Define Coupling



Degree of Coupling

1 Tight coupling

- synchronous
- trust (no validation)
- common data

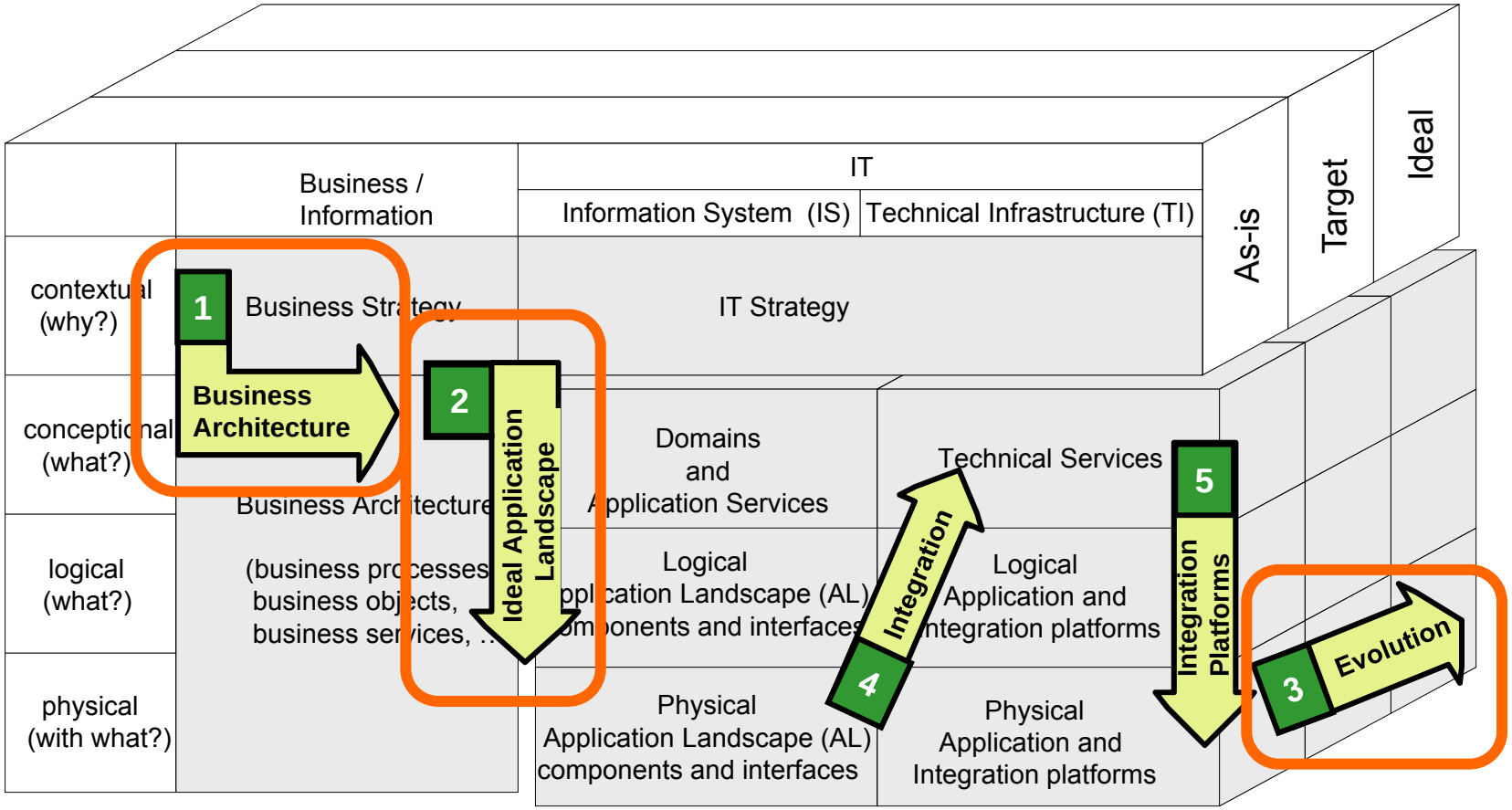
2 Medium coupling

- synchronous
- no trust (i.e. validation)
- no common data

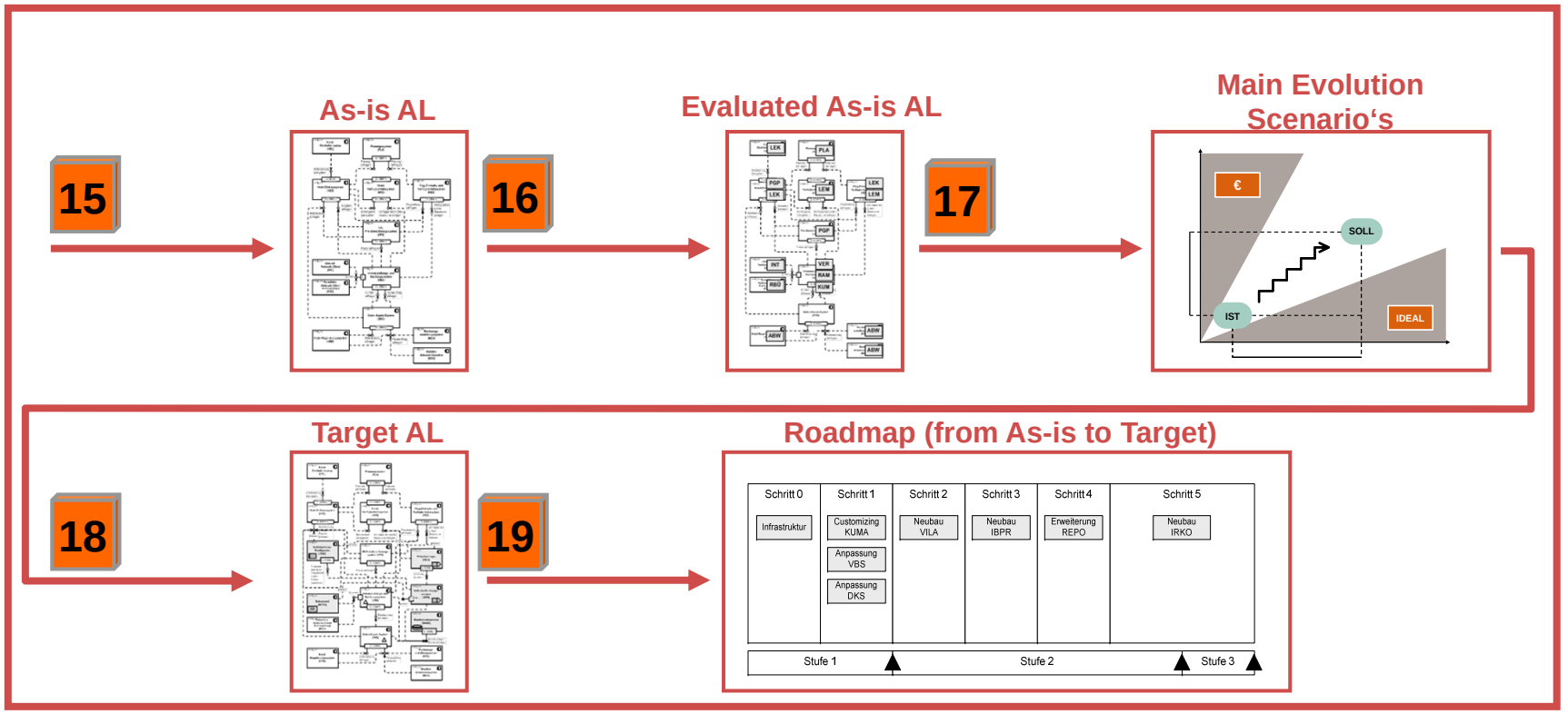
3 Loose coupling

- asynchronous
- no trust (i.e. validation)
- no common data

Quasar Enterprise: Roadmap within Refined IAF Structure



Quasar Enterprise Method:
Main Step 3: Evolution Planning



15 Identify As-is State of Application Landscape

16 Evaluate As-Is State of Application Landscape

17 Determine Main Evolution Scenario‘s

18 Design Target Application Landscape

19 Determine Evolution Roadmap

20 Reference Evolution Scenario‘s

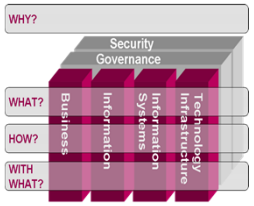
Quasar Enterprise – the SOA method of Capgemini CSD

Architecture framework

Corporate level Business Architecture

IAF – Integrated Architecture Framework

- Comprehensive architecture framework
- 100% adaptable to specific client situations; from single projects to corporate transformation
- Recognised by the Open Group IT Architecture Certification Program
- Viewed as market leader by analysts (Forrester, Gartner, IDC) and key players (Microsoft, HP, Cisco, SAP, …)



Corporate level IT Architecture

Quasar Enterprise

- Service-oriented design of application landscapes
- Method module to supplement IAF
- Identification and design of domains and services
- Reference architectures and samples for integration and evolution

Application level IT Architecture

Quasar – Quality Software Architecture

- Overall methodology for system design (specification, construction)
- Software development environment “out-of-the-box”
- Components on Open Source basis
- Model-driven development (models and tools)

Proactive controlling of software quality

Quasar Analytics

- Quality model, figures, software cockpit
- Quality gates in the development process
- Intelligent test procedures and test automation

References Quasar Enterprise

Gregor Engels, Markus Voß: Quasar Enterprise. In Informatik-Spektrum, vol. 31, no. 6, pp. 548-555. Springer (Berlin/Heidelberg) (2008)

Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Quasar Enterprise: Anwendungslandschaften serviceorientiert gestalten. dpunkt-Verlag (München) (2008)

Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: Anwendungslandschaften serviceorientiert gestalten. In R. Reussner, W. Hasselbring (eds.): Handbuch der Softwarearchitektur, pp. 151-178. dpunkt-Verlag (2008)

Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: A Method for Engineering a true Service-Oriented Architecture. In J. Cordeiro, J. Filipe (eds.): Proceedings of the Tenth International Conference on Enterprise Information Systems (ICEIS 2008), Barcelona (Spain). Springer (Berlin/Heidelberg), vol. ISAS-2, pp. 272-281 (2008)

Andreas Hess, Bernhard Humm, Markus Voß, Gregor Engels: Structuring Software Cities - A Multidimensional Approach. In Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007). IEEE Computer Society (Washington, DC, USA), pp. 122-129 (2007)

Gregor Engels, Markus Voß: Quasar Enterprise - Anwendungslandschaften serviceorientiert gestalten. In K. Herrmann, B. Bruegge (eds.): Software Engineering 2008. Fachtagung des GI-Fachbereichs Softwaretechnik. GI, LNI, vol. 121, pp. 24-27 (2008)

Andreas Hess, Bernhard Humm, Markus Voß, Gregor Engels: Structuring Software Cities - A Multidimensional Approach. In Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007). IEEE Computer Society (Washington, DC, USA), pp. 122-129 (2007)

Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, Johannes Willkomm: A Method for Engineering a true Service-Oriented Architecture. In J. Cordeiro, J. Filipe (eds.): Proceedings of the Tenth International Conference on Enterprise Information Systems (ICEIS 2008), Barcelona (Spain). Springer (Berlin/Heidelberg), vol. ISAS-2, pp. 272-281 (2008)

That‘s my life

...

1991 -1997 **Professor Software Engineering and Information Systems, University of Leiden (NL)**

since 1997 **Professor Information Systems, University of Paderborn (D)**
- Research Topics: MDA, UML, DSL, SOA, SPL, SQA, MBT, …
- currently 19 PhD students, > 200 scientific publications

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

since 2005 **Chairman of the Board of Directors, s-lab (Software Quality Lab), PPP-institute, University of Paderborn**

s-lab
Software Quality Lab

since 2005 **Scientific Director Capgemini, CSD Research, Munich**

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
CSD Research

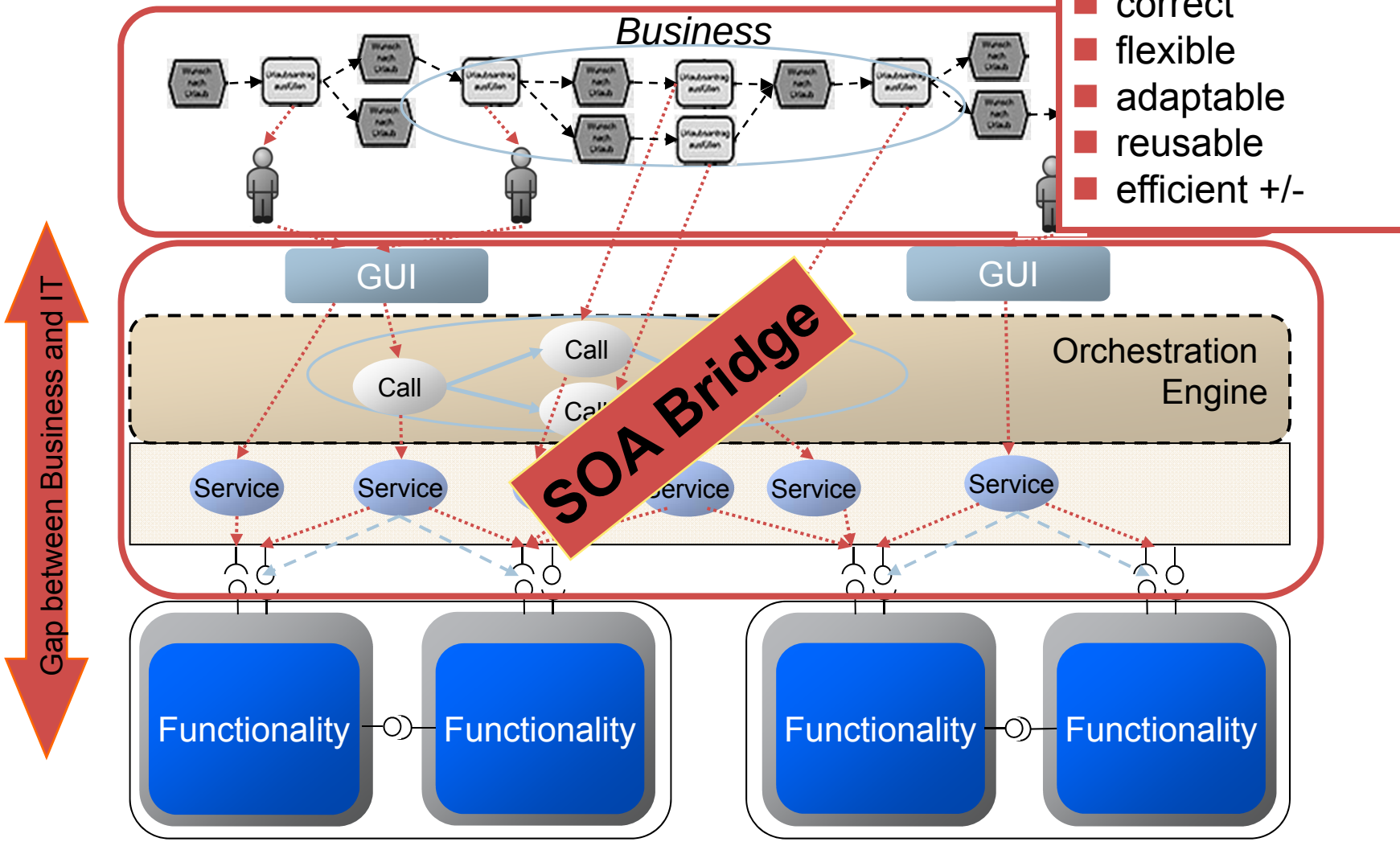
since 2011 **Collaborative Research Center „On-the-Fly Computing“, funded by Deutsche Forschungsgemeinschaft**

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Quality of Business Process Models

Quality Objectives:

- correct
- flexible
- adaptable
- reusable
- efficient +/-

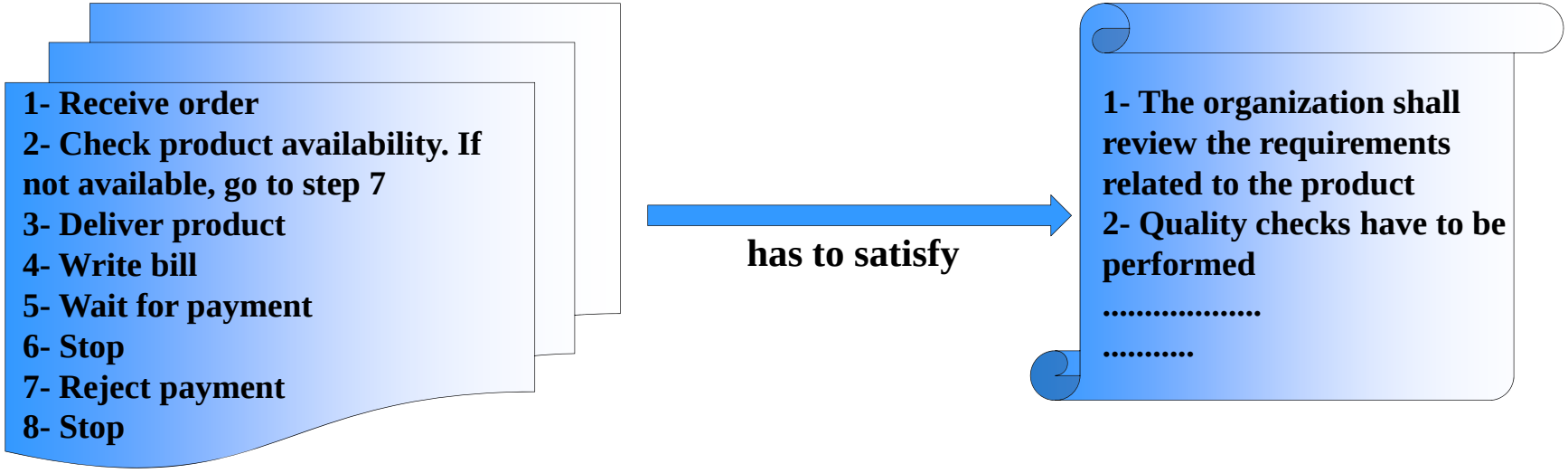


Quality of Business Process Models

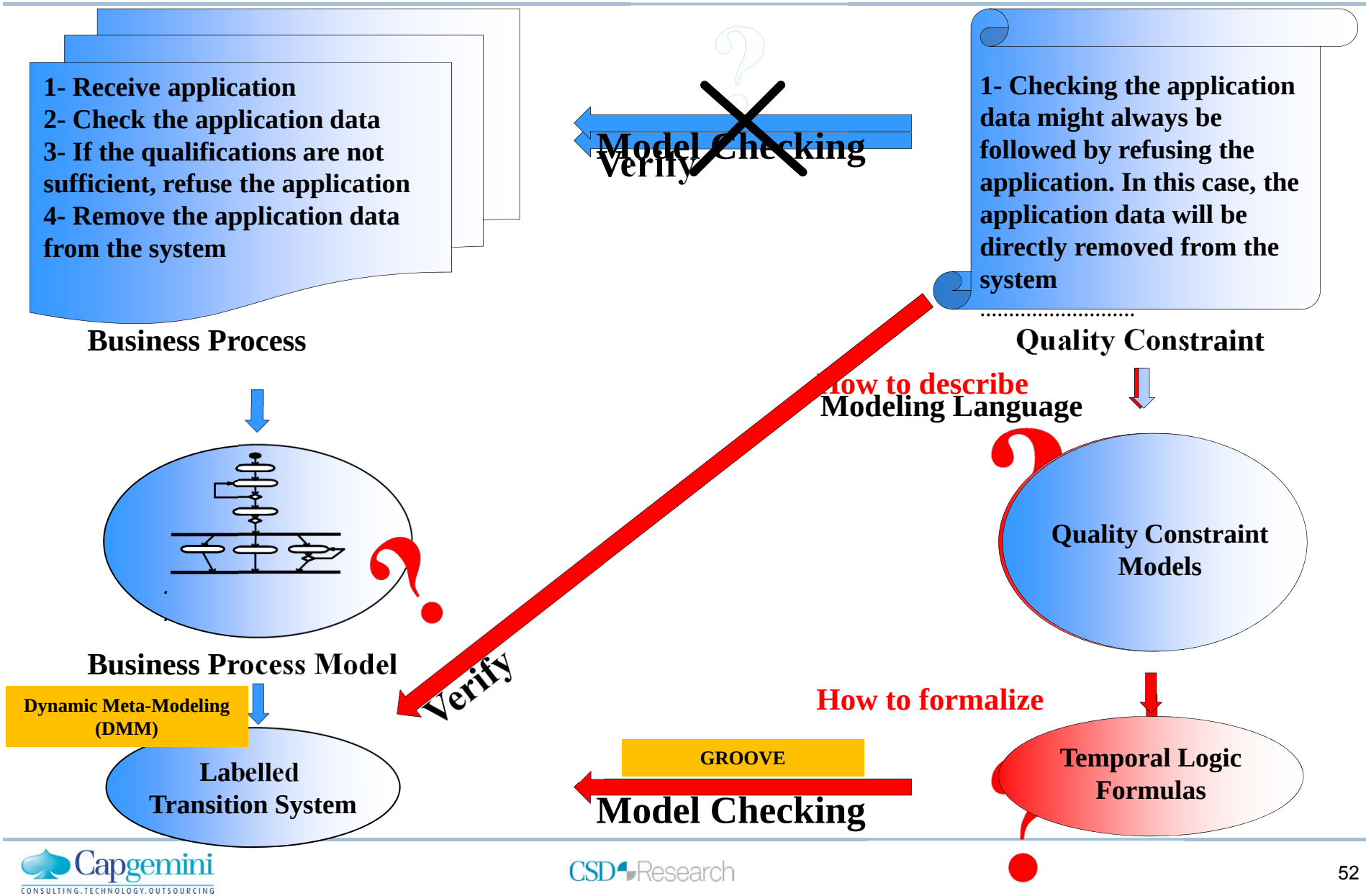
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Alexander Förster

- **Quality Constraints as Business Rules**
  - **standards-driven (e.g., ISO 9001)**
  - **customer-defined**



Motivation


1- Receive application
2- Check the application data
3- If the qualifications are not sufficient, refuse the application
4- Remove the application data from the system
Business Process
Model Checking
Verify
1- Checking the application data might always be followed by refusing the application. In this case, the application data will be directly removed from the system
Quality Constraint
How to describe
Modeling Language
Quality Constraint Models
Business Process Model
Verify
Dynamic Meta-Modeling (DMM)
Labelled Transition System
How to formalize
GROOVE
Model Checking
Temporal Logic Formulas

Modeling Quality Constraints

Extended Process Pattern Specification Language (EPPSL)

Example1

- It is always the case that checking the applicatio followed by refusing the application. In this case data will be directly removed from the system

It is always the case that checking the application data might be followed by refusing the application. In this case, the application data will be directly removed from the system

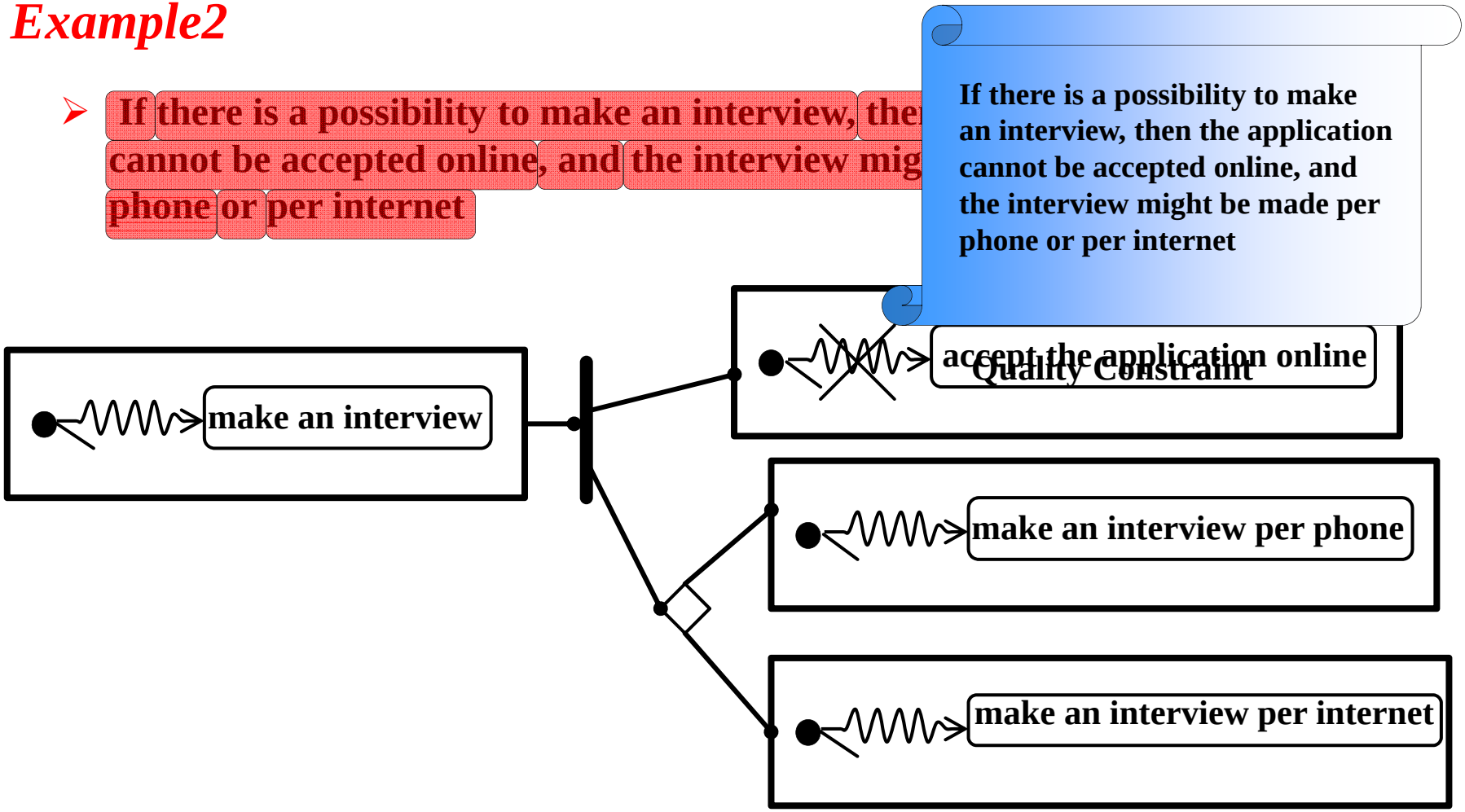
Quality Constraint

Modeling Quality Constraints with EPPSL

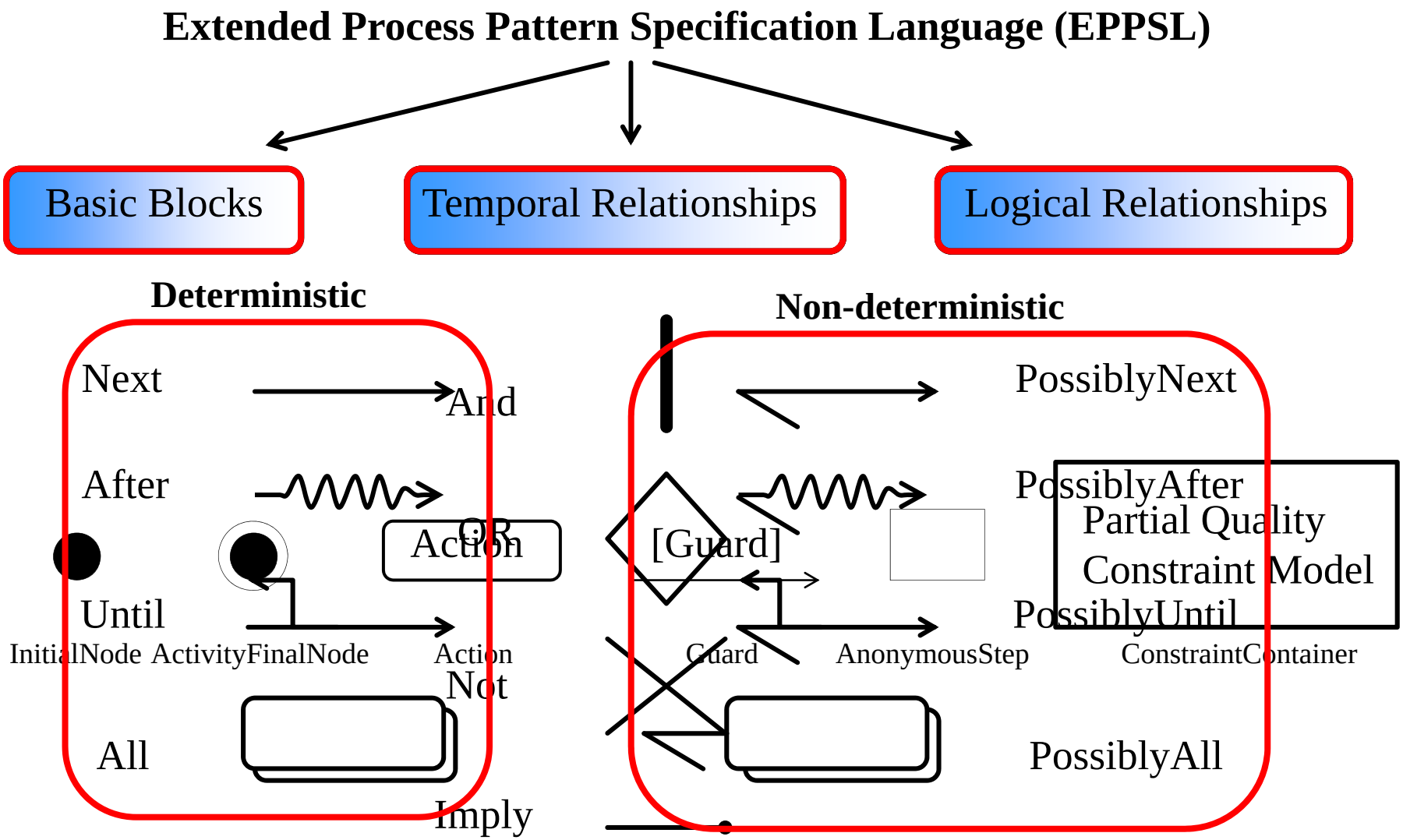
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Example2

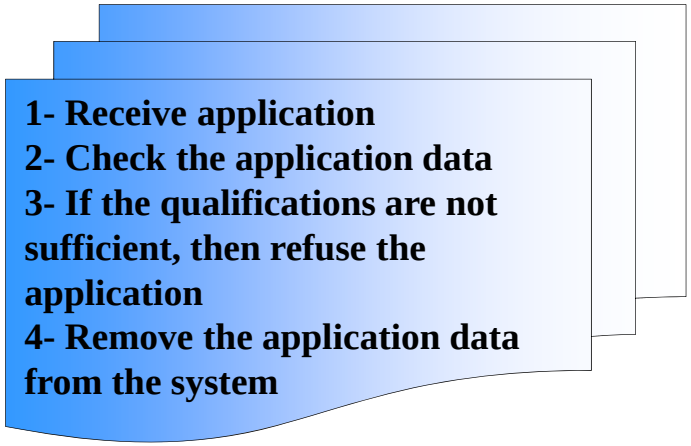
- If there is a possibility to make an interview, ther
  cannot be accepted online, and the interview mig
  phone or per internet



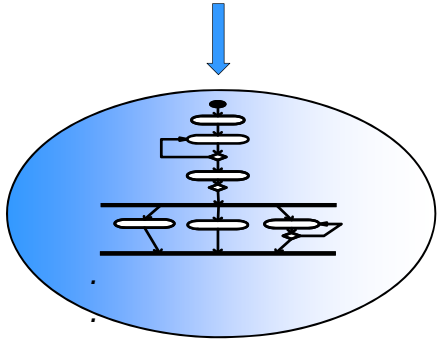
Modeling Language



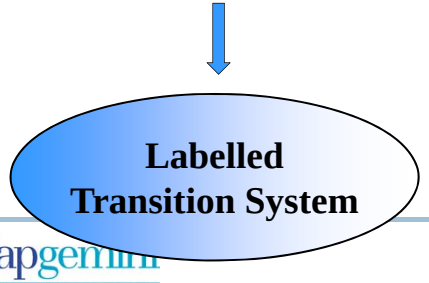
Formalizing Quality Constraints

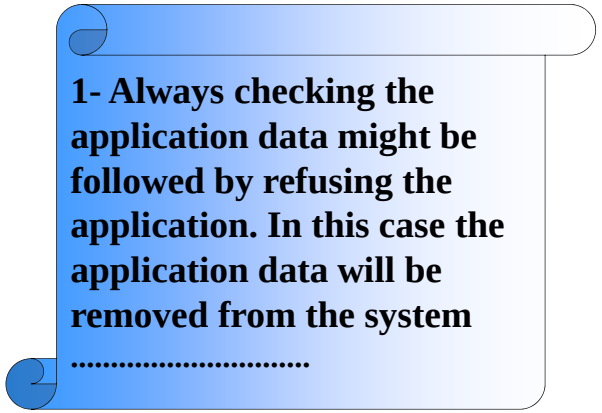


Business Process



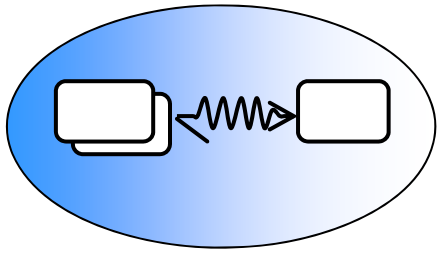
Business Process Model





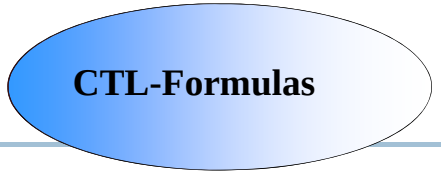
Quality Constraint

EPPSL



EPPSL Quality Constraint Models

Translation



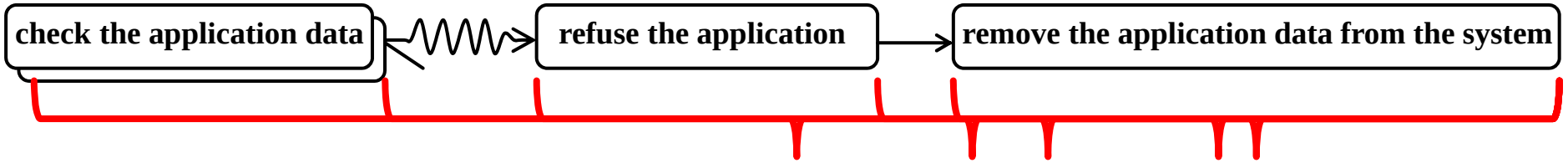
Model Checking

Pattern-Based Translation into CTL

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Example1

- **It is always the case that checking the application data might be followed by refusing the application. In this case, the application data will be directly removed from the system**



AG(check the application data → AX(refuse the application → AX(remove the application data from the system)))

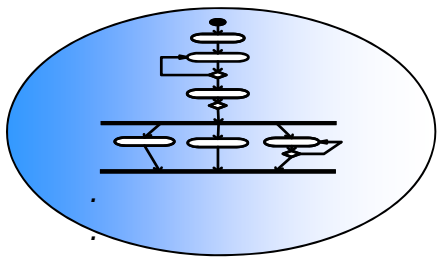
EPPSL Pattern	CTL Formula
Action	Action
→ S	AX(S*)
Action S	Action → S*
S	EF(S*)
Action S	AG(Action → S*)

Quality of Business Process Models

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

1- Receive application
2- Check the application data
3- If the qualifications are not sufficient, then refuse the application
4- Remove the application data from the system

Business Process



Business Process Model

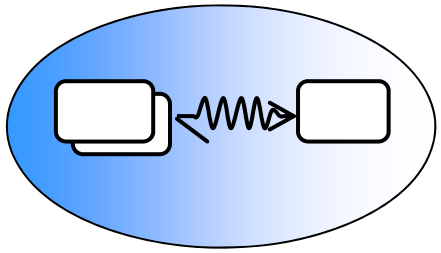
Dynamic Meta-Modeling (DMM)

Labelled Transition System

1- Always checking the application data might be followed by refusing the application. In this case the application data will be removed from the system
.....

Quality Constraints

EPPSL



EPPSL Quality Constraint Models

Translation

CTL-Formulas

GROOVE

Model Checking

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

CSD Research

Literature on Business Process Patterns

A. Förster, G. Engels, T. Schattkowsky, R. Van Der Straeten: Verification of Business Process Quality Constraints Based on Visual Process Patterns. In Proc. 1st IEEE Int. Symposium on Theoretical Aspects of Stoftware Engineering (TASE) 2007, Shanghai, China, pp. 197-208, IEEE Press, 2007

A. Förster, T. Schattkowsky, G. Engels, R. Van Der Straeten: A Pattern-driven Development Process for Quality Standard-conforming Business Process Models: In Proc. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Brighton 2006, pp. 135 - 142. IEEE Computer Society, 2006.

A. Förster, G. Engels, T. Schattkowsky: Activity Diagram Patterns for Modeling Quality Constraints in Business Processes. In Proc. ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2005), Jamaica, October 2005, pp. 2-16. LNCS 3713, Springer-Verlag, 2005.

Lial Khaluf, Christian Gerth, Gregor Engels: Pattern-Based Modeling and Formalizing of Business Process Quality Constraints. In H. Mouratidis and C. Rolland (eds.): Proceedings of the 23rd International Conference on Advanced Information System Engineering (CAiSE'11). Springer (Berlin/Heidelberg), LNCS, vol. 6741, pp. 521-535 (2011)

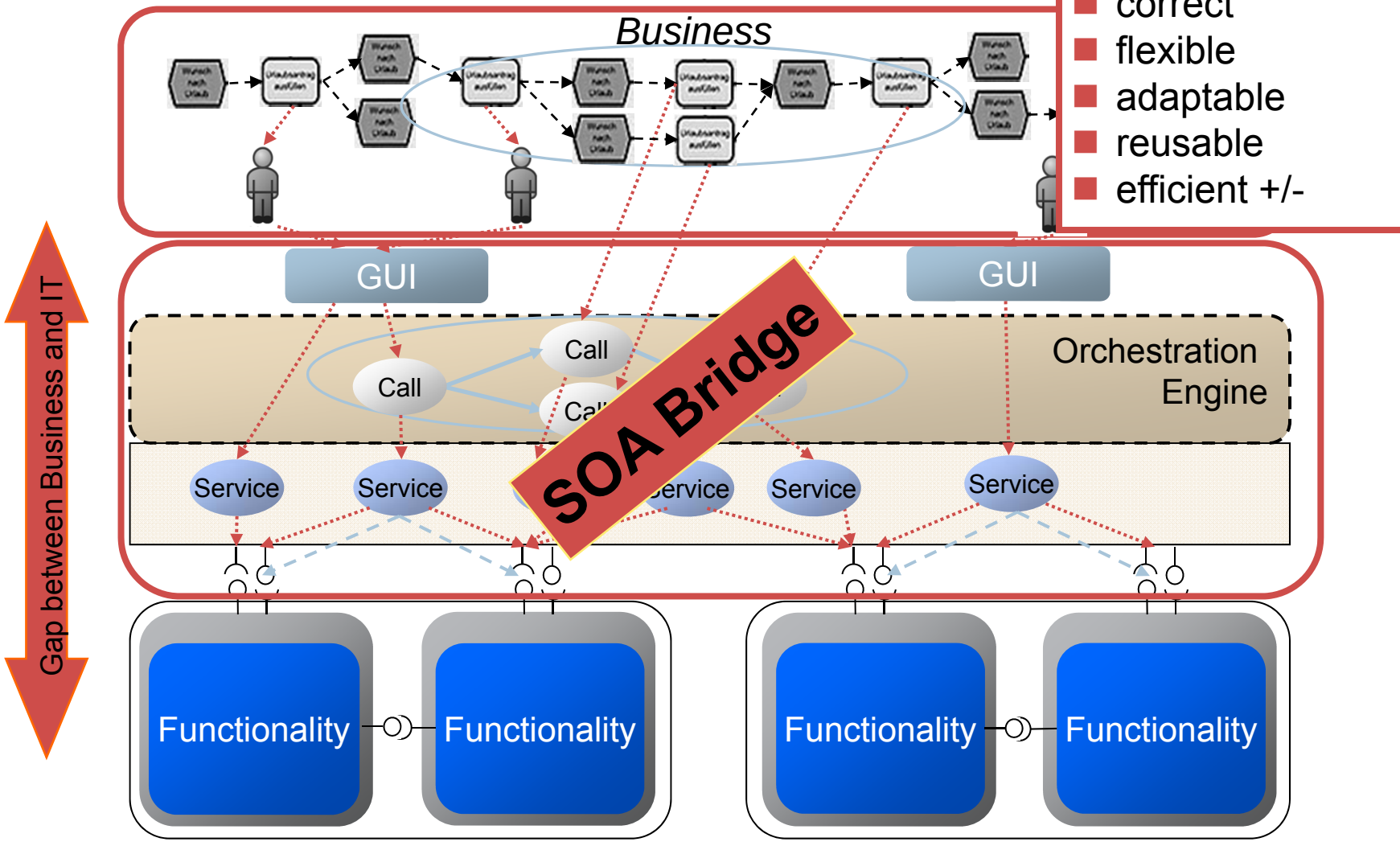
Gregor Engels, **Jan Hendrik Hausmann**, Reiko Heckel, Stefan Sauer: Dynamic Meta-Modeling: A Graphical Approach to the Operational Semantics of Behavioral Diagrams in UML. In A. Evans, S. Kent, B. Selic (eds.): Proceedings of the 3rd international conference on the Unified Modeling Language (UML 2000), York (UK). Springer (Berlin/Heidelberg), LNCS, vol. 1939, pp. 323-337 (2000)

Christian Soltenborn, Gregor Engels: Analysis of UML Activities with Dynamic Meta Modeling Techniques. In T. Kühne (eds.): Symposium "A Formal Semantics for UML" (satellite event of the MoDELS conference 2006), Genova (Italy). Springer (Berlin/Heidelberg), LNCS, vol. 4364, pp. 329-330 (2007)

Quality of Business Process Models

Quality Objectives:

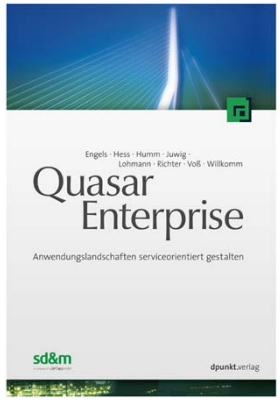
- correct
- flexible
- adaptable
- reusable
- efficient +/-



Topics of the Talk

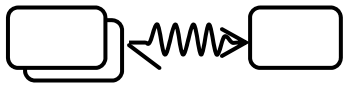
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Service-oriented Architecture –
Experiences, Ideas, Innovations



Quality Constraints

EPPSL



?

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

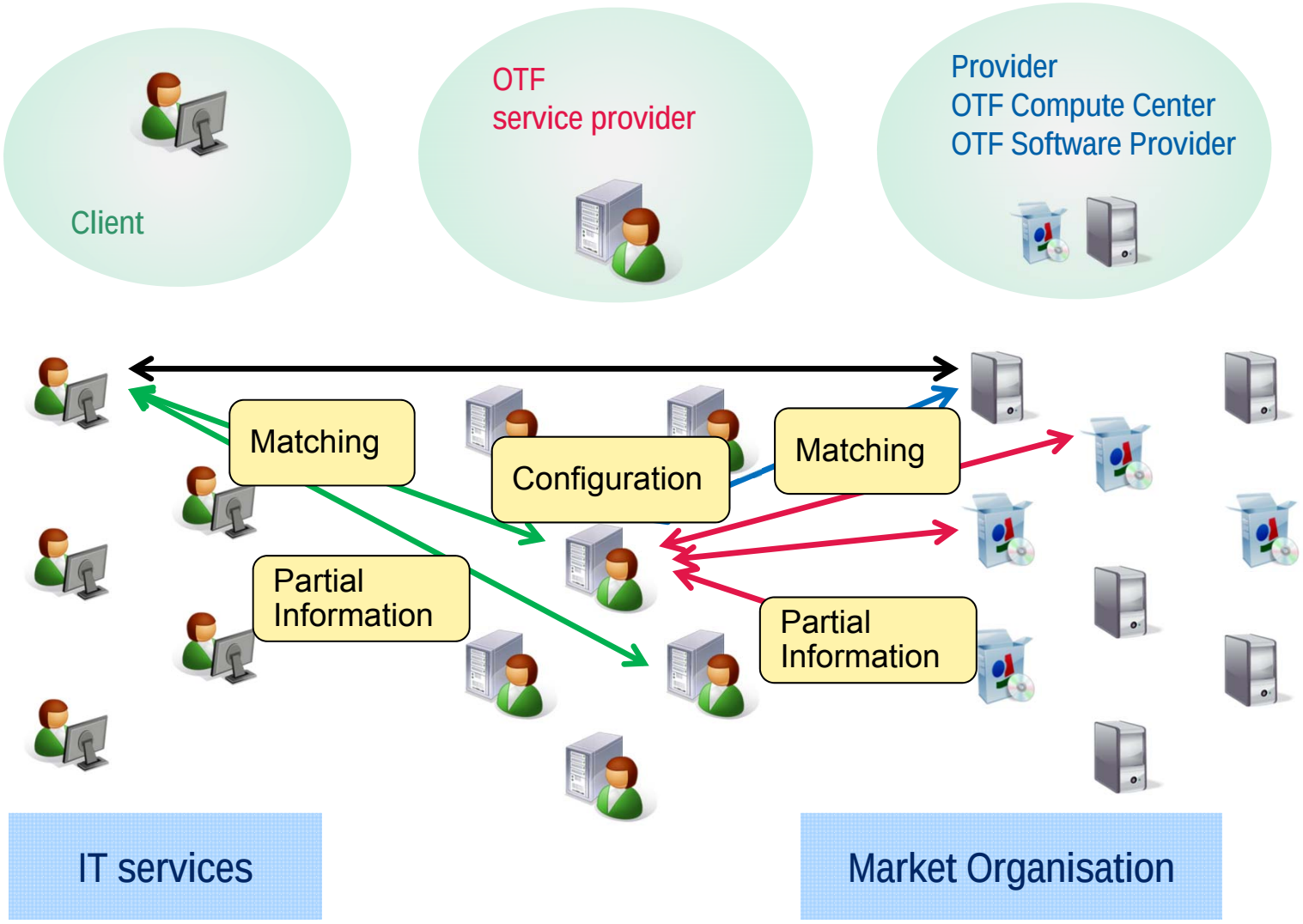
DFG

Collaborative Research Center 901

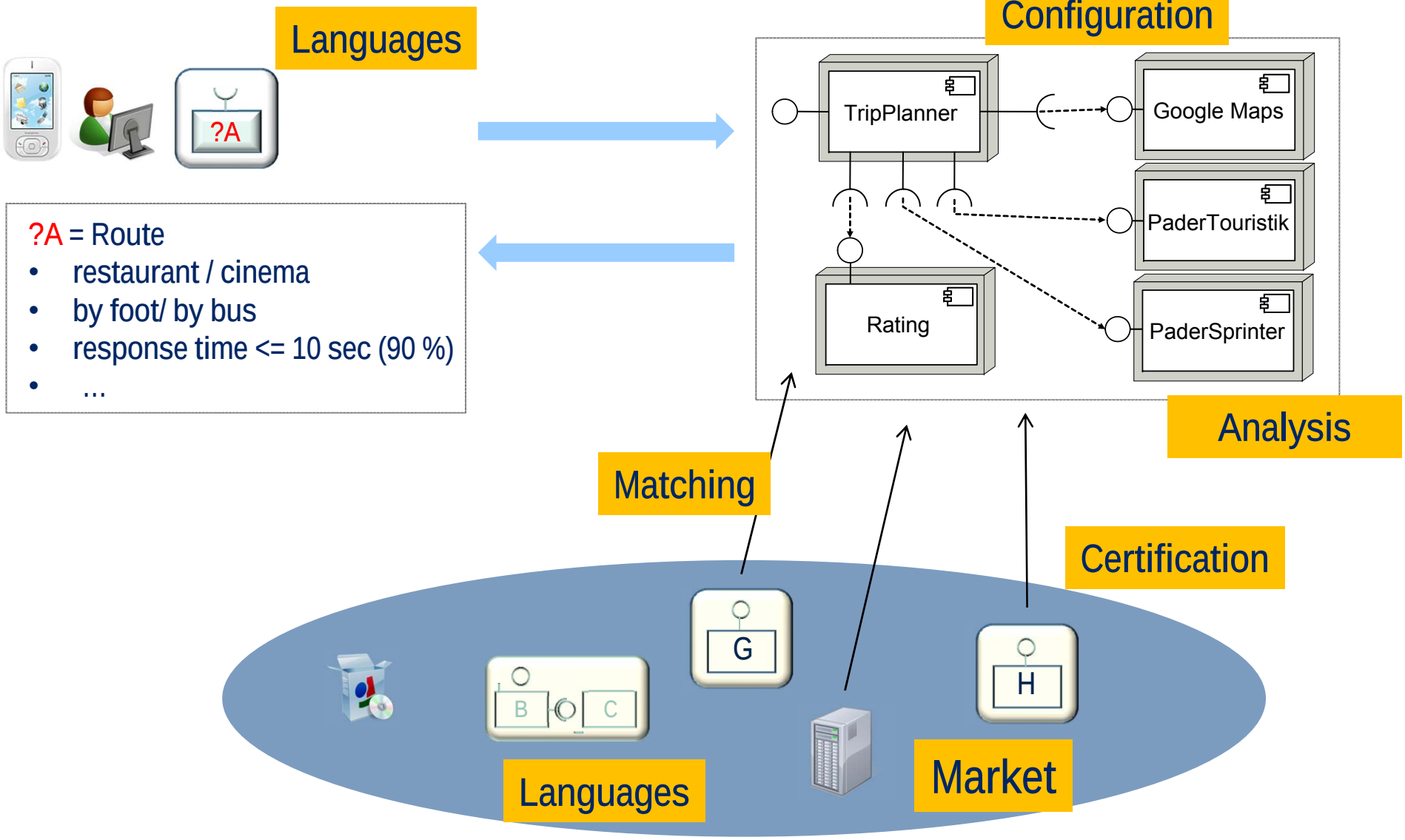
On-The-Fly Computing

- **Automated configuration and execution of IT Services**
- **Organisation of world-wide distributed service markets**

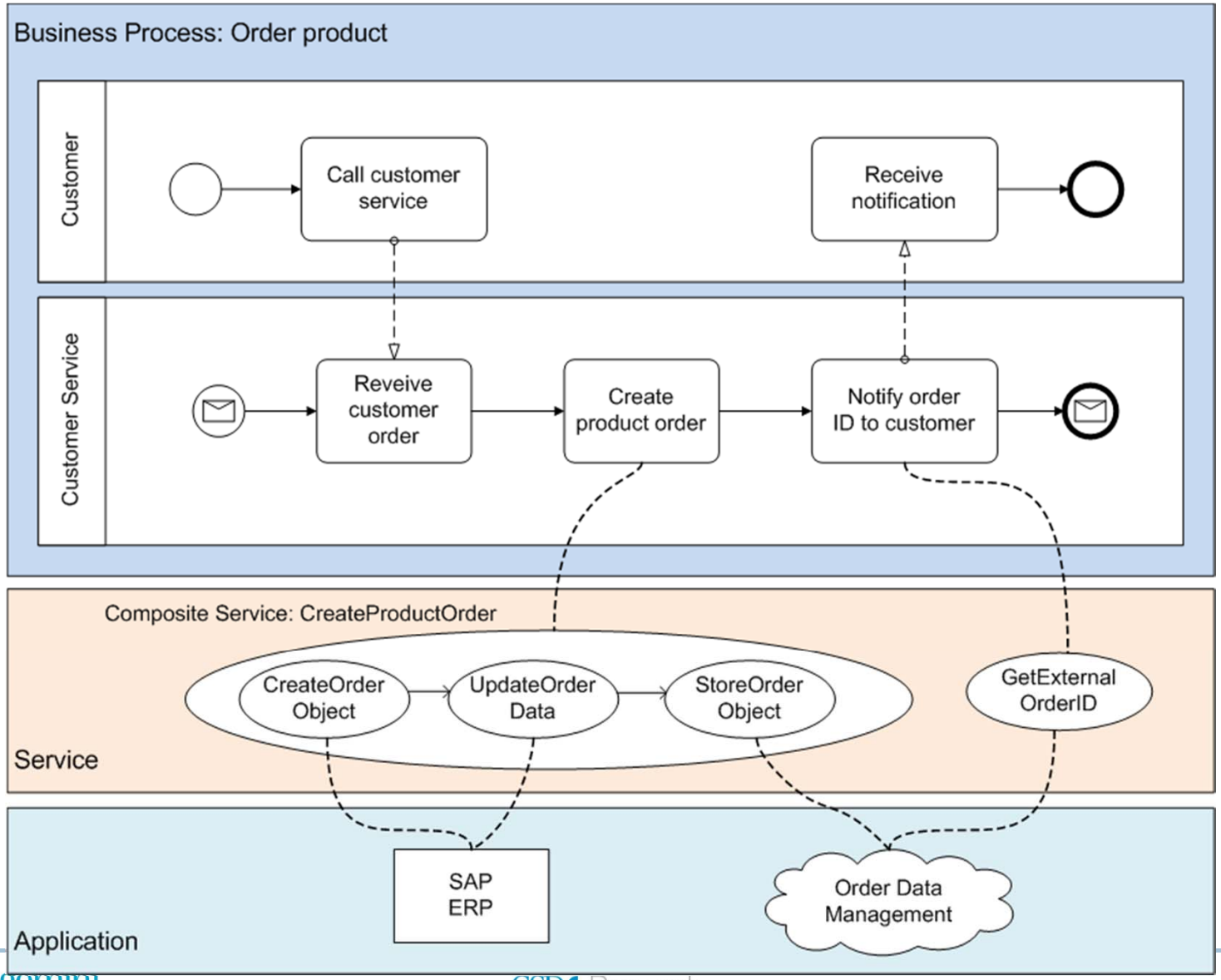
Actors in On-The-Fly Computing Markets



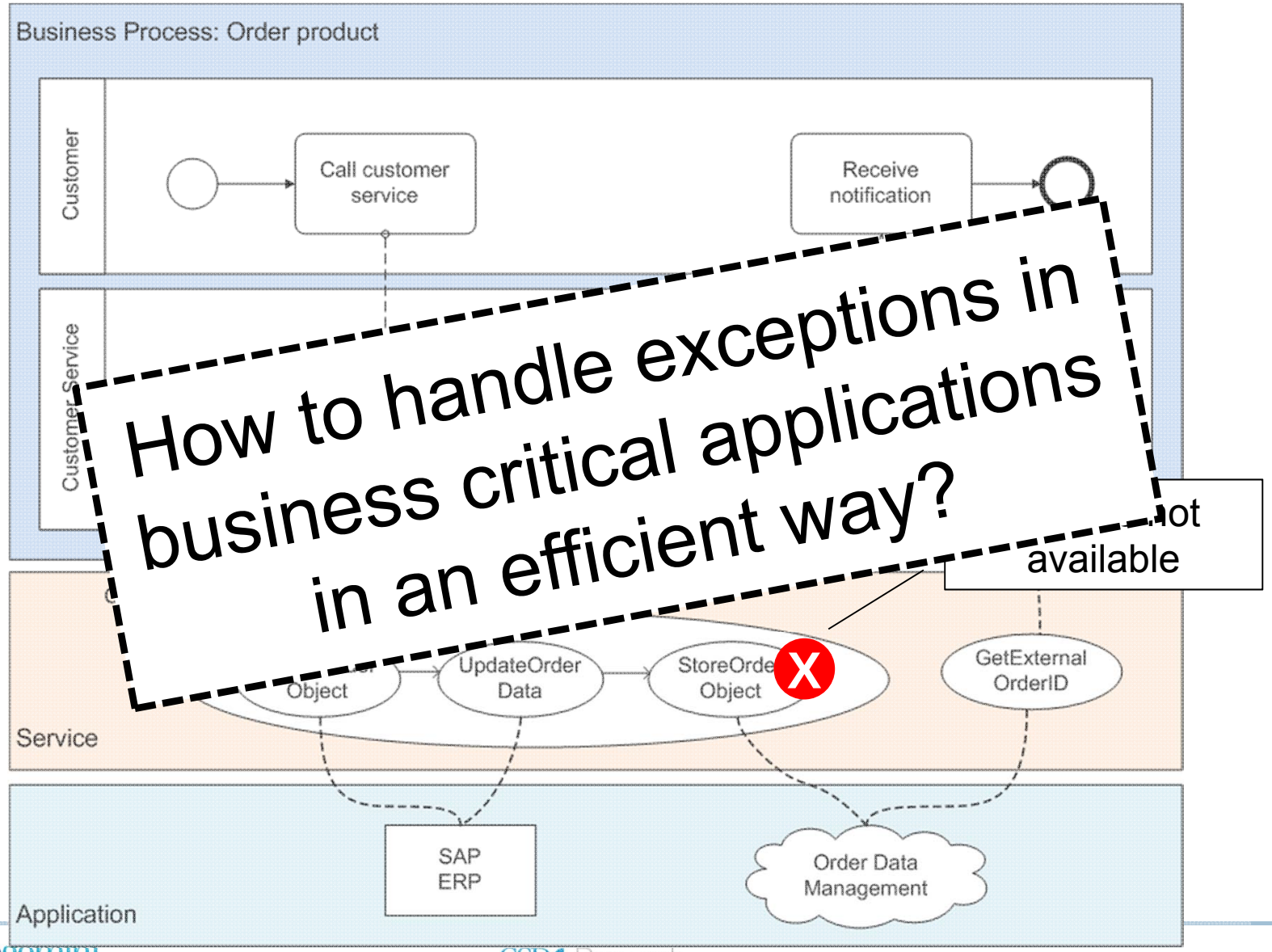
Sample Scenario



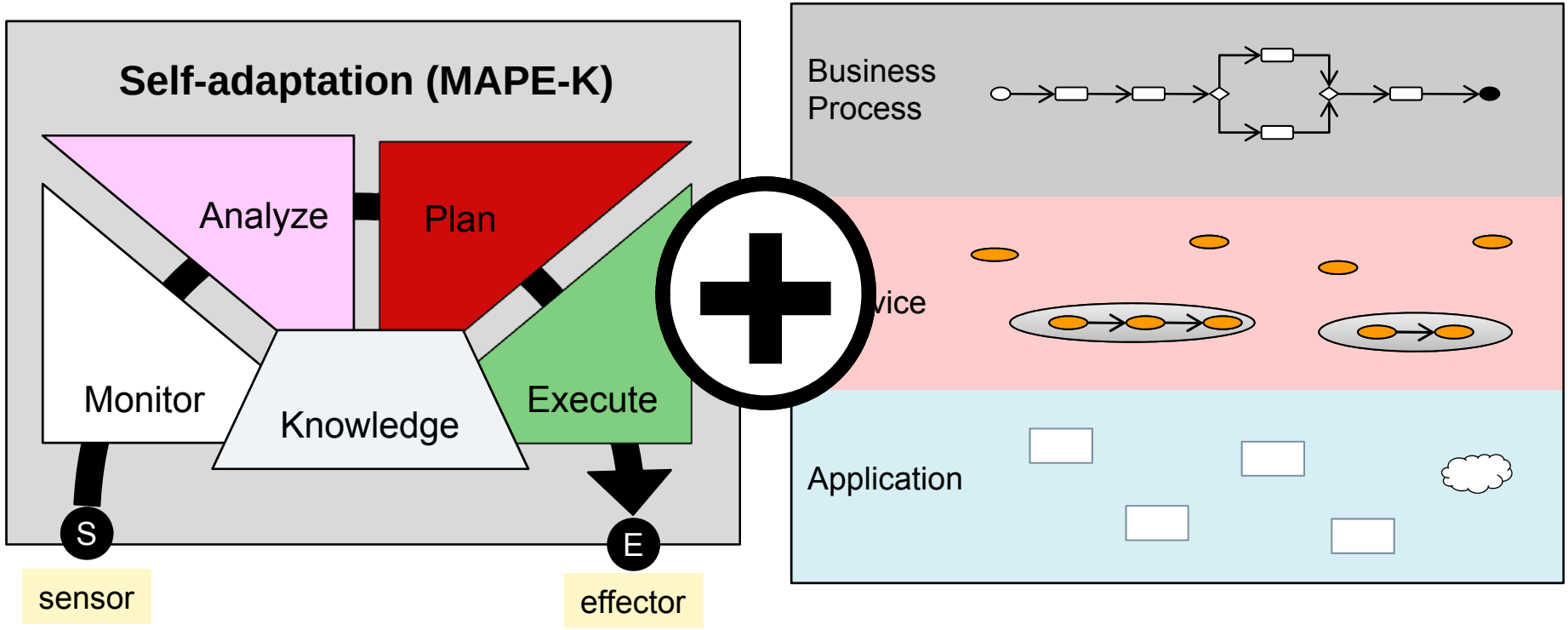
Example


Business Process: Order product
Customer
Call customer service
Receive notification
Customer Service
Reveive customer order
Create product order
Notify order ID to customer
Composite Service: CreateProductOrder
CreateOrder Object
UpdateOrder Data
StoreOrder Object
GetExternal OrderID
Service
SAP ERP
Order Data Management
Application

Life is not that easy ...

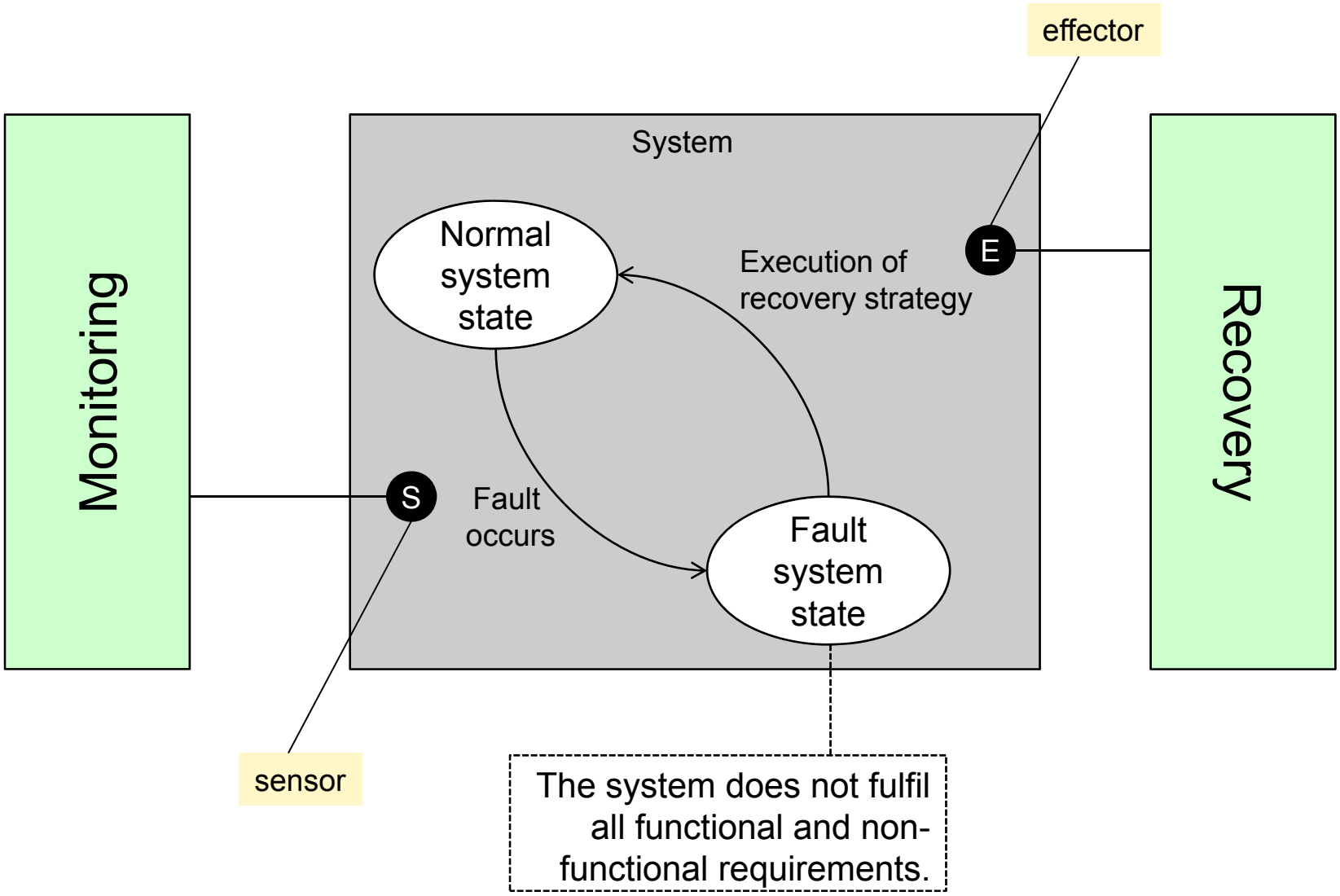


Self-adaptation meets SOA

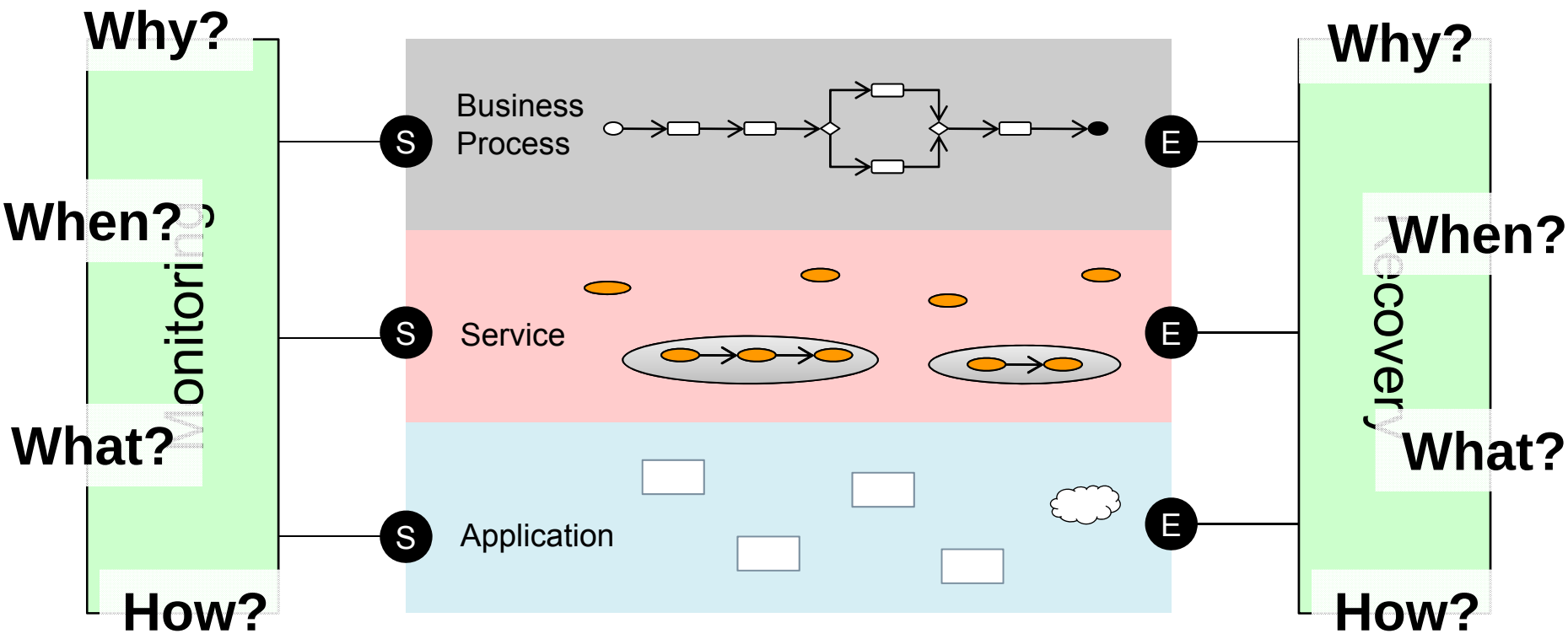


→Self-recovering from faults occurring at runtime without human intervention

Abstract view on a self-recovering system



Self-recovering Business Applications



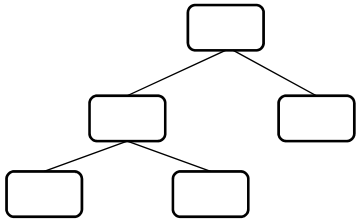
Metzger, A.; Pohl, K.
Towards the Next Generation of Service-Based Systems: The S-Cube Research Framework
In *Advanced Information Systems Engineering*, 2009.

How to answer these questions?

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Benjamin Nagel

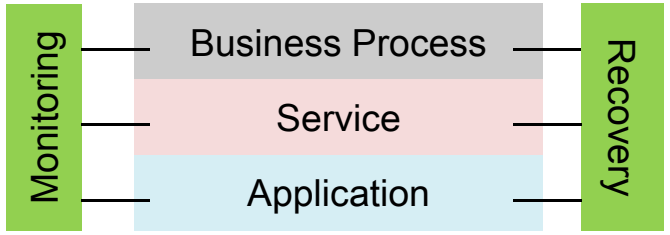
Goal-oriented Requirements Engineering



Process-Design

Adaptation-Design

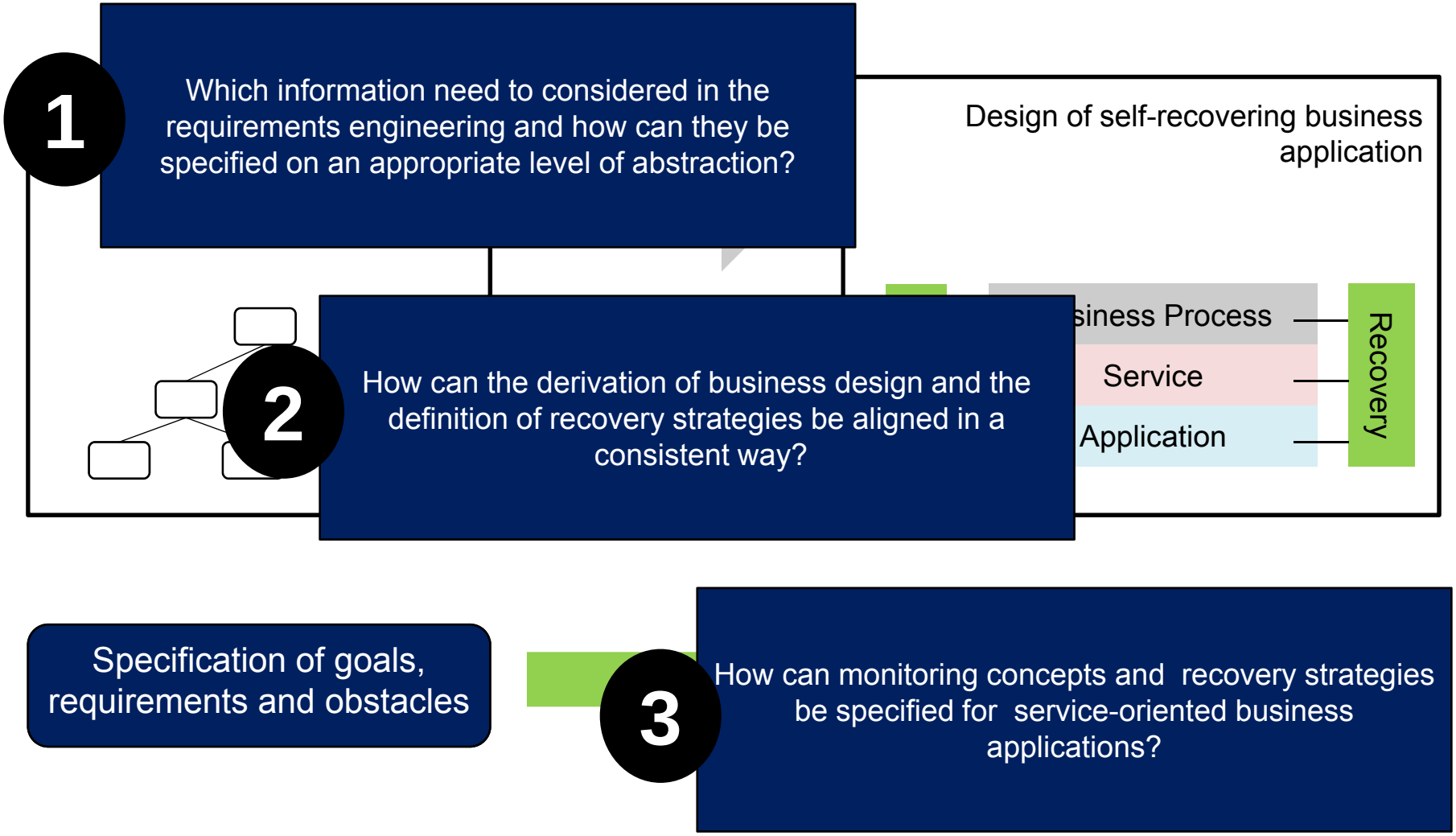
Design of self-recovering business application



Specification of goals, requirements and obstacles

Specification of monitoring concepts and recovery strategies

How to answer these questions?

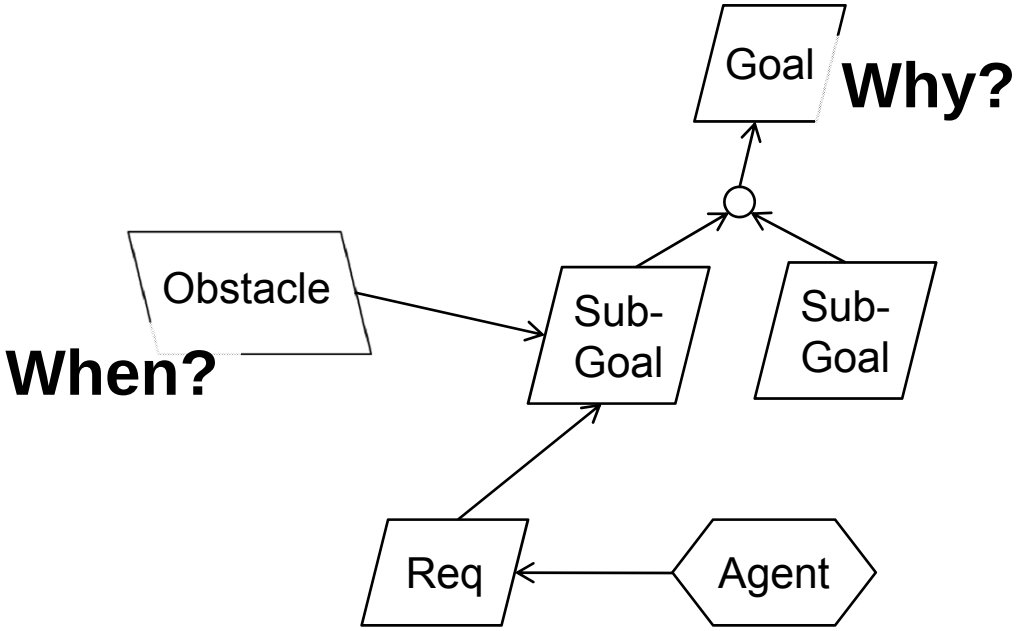


UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Extended requirements specification

Existing work of goal-oriented RE

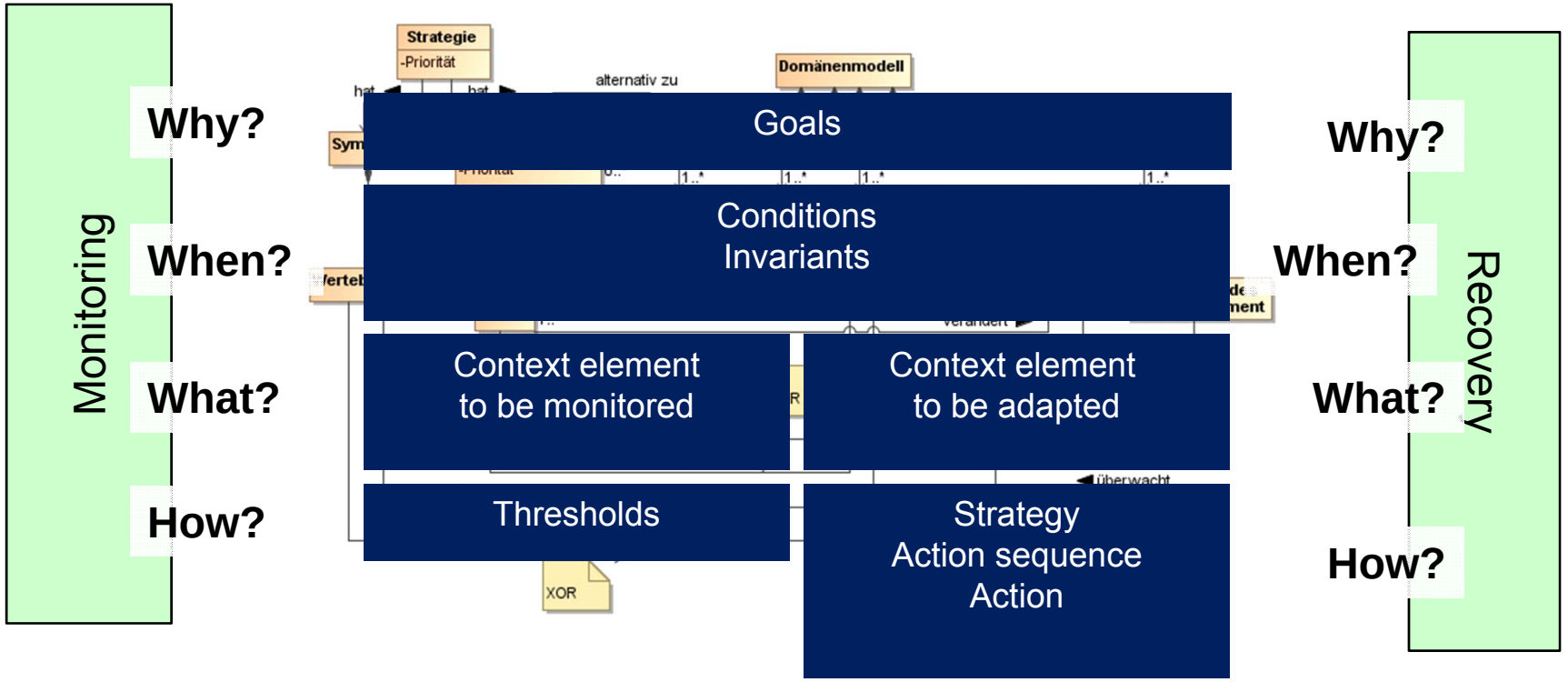
- KAOS [Dardenne1993]
- i* [Yu1997], Tropos [Besciani2004]
- Combined approach of KAOS & RELAX [Cheng2009]



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Extending KAOS with adaptive aspects

Extended requirements specification



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Derivation methodology

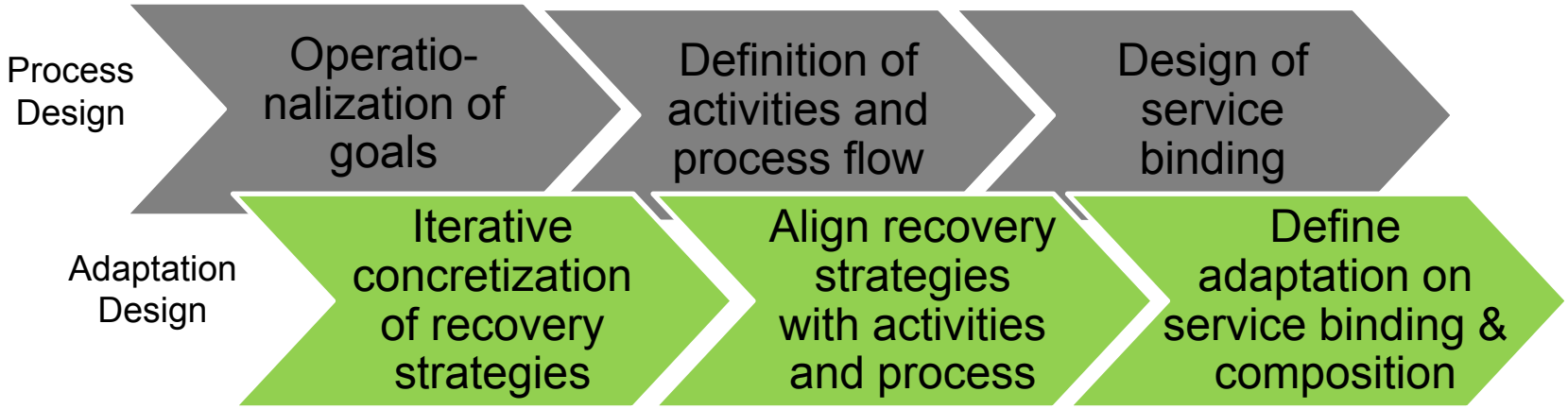
Existing work

From goals to processes [Pasquale2009]

From goals to software design [Lamsweerde2003], [Yu2008]

Agent-oriented approach [Morandini2008]

Design pattern [Ramirez2010]

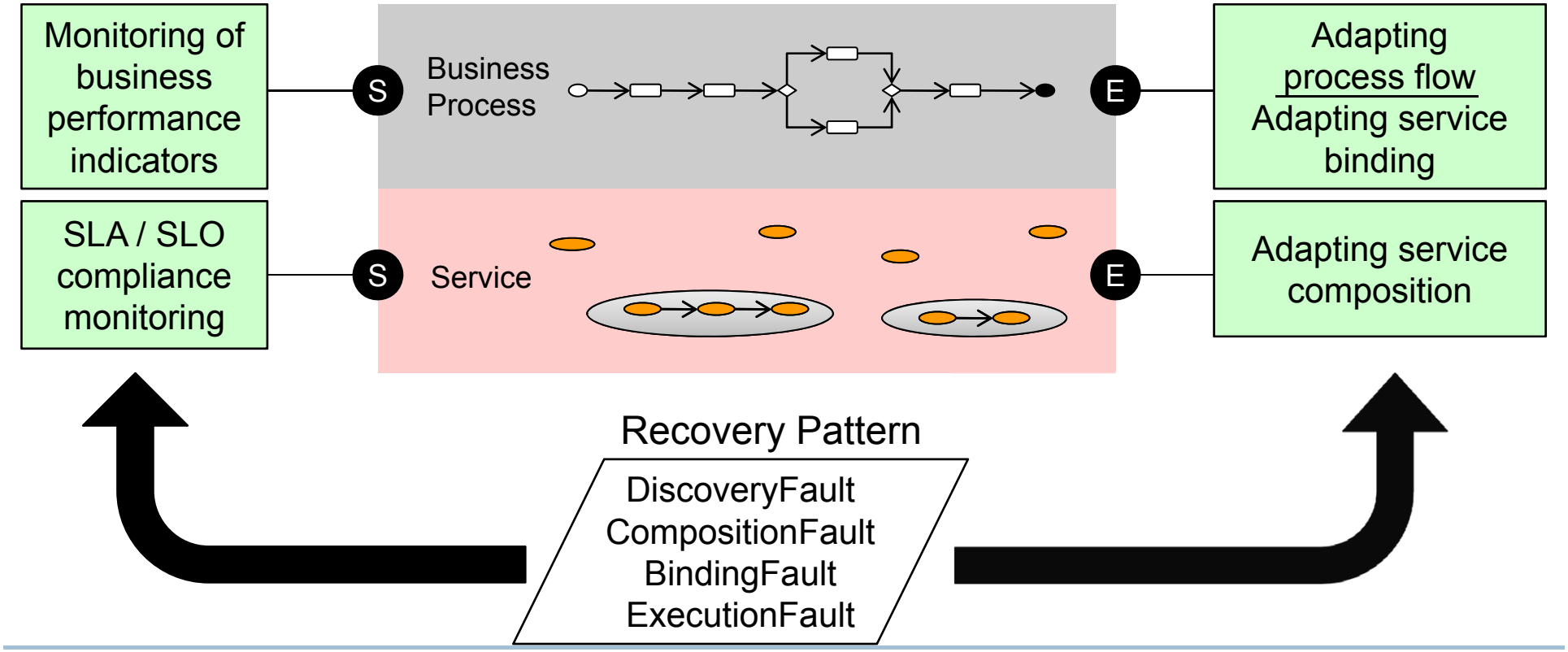


Recovery Strategies

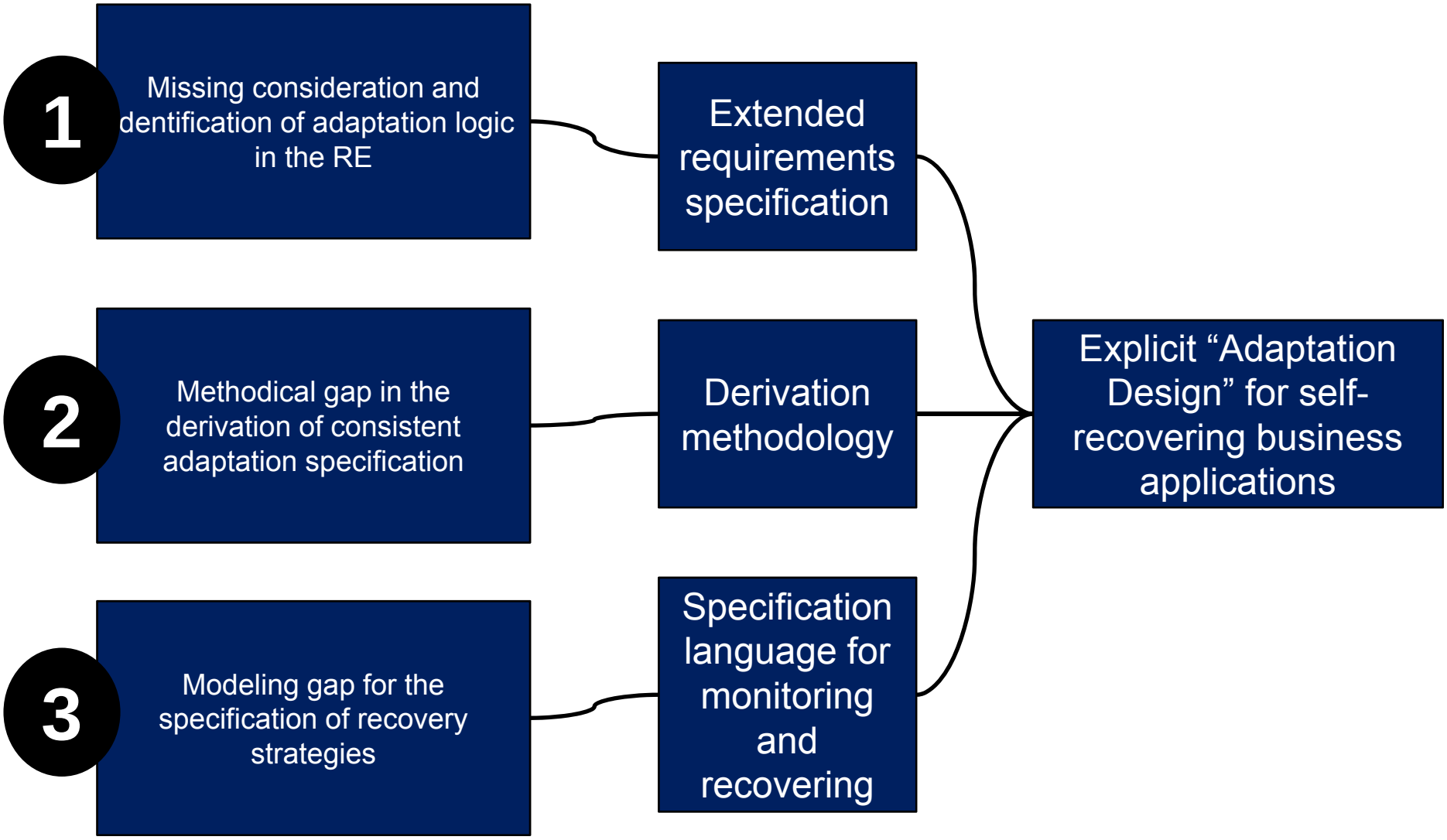
UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

- **Formal specification of monitoring concepts and recovery strategies**
- **Inter- and intra-layer adaptation**

Specification of recovery strategies

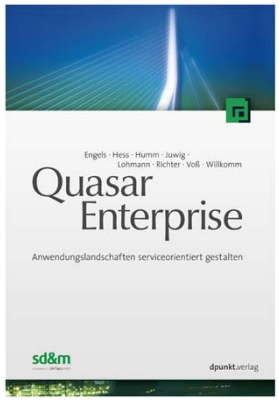


Problem – Solution Mapping



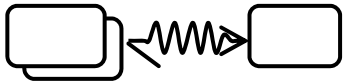
Conclusions

Service-oriented Architecture –
Experiences, Ideas, Innovations



Quality Constraints

EPPSL



On-The-Fly Computing

SOA Method

Quality of SOA Systems

Self-adaptive SOA Systems

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

CSD Research

UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

s-lab
Software Quality Lab

Thank you very much for your attention!

Questions?
engels@upb.de

www.de.capgemini.com