Challenges in Virtualization and SLA Management

Experiences from CoreGRID and SLA@SOI

25 October 2008
Ramin Yahyapour
Technische Universität Dortmund

CoreGRID RMS Institute Objective

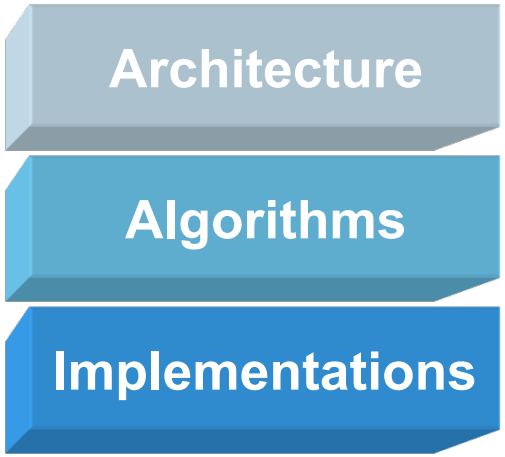
Objectives:

Development of a common and generic solution for Grid resource management/scheduling in Next Generation Grids.

Development of new algorithms for coordinated scheduling for all resource types, including data, network etc.

Support of Grid business models in the scheduling process

Goal: linking theoretical foundation and practical implementation on the different level of Resource Management



Key Question

"Which services/resources to use for an activity, when, where, how?"

Typically:

A particular user, or business application, or component application needs for an activity one or several services/resources under given constraints

- Trust & Security
- Timing & Economics
- Functionality & Service level
- Application-specifics & Inter-dependencies
- Scheduling and Access Policies

This question has to be answered in an automatic, efficient, and reliable way.

Part of the invisible and smart infrastructure!

Scheduling in Grids

Computational Grids are supposed to facilitate cooperation among users and increase system efficiency.

- Improvement of efficiency requires appropriate resource management
- Job scheduling is a key task of resource management

There is a development path from parallel and distributed systems to Grids.

- Solutions to Grid management problems may be derived by adapting approaches in parallel and distributed systems.
- Approaches used in parallel and distributed systems can be used as a reference to new approaches for Grids

Key questions of all kind of job scheduling problems in parallel systems:

- When and where to execute a job?

Differences between Grids and SOA environments

Grids have a notion of jobs:

- Submitted over time
- Mostly independent
- Batch queued
- Mostly space-shared execution
- Sequential or parallel application

SOA/SOI have different application types:

- Mostly transaction oriented, repeatable
- Relatively long running, hosting
- Support time-sharing execution
- Often synchronous workflow, related services
- Combined resource setting:
  - hosting environment, AppServer, DB

Using Virtualization…

Clear advantages

- Abstraction from hardware
- Dynamicity in associating virtual and physical resources
- Scaling on demand

Major impact in commercial DataCenters

- Consolidation of servers, storage

For Grids…

support different execution environments

- Different Grid infrastructures for the same physical resource
- Leverage individual execution environments per job
- Jobs become VM images

utilizing VM features in job/resource management

- Checkpoint/Restart
- Hot-Migration/Live-Migration
- Dynamic resource scaling

...into the Clouds

Virtualization laid the foundation for Cloud Computing

- Seamless execution environment
- Closed shop environment
- Simple business model
- 1:1 relationship between customer and provider
- Easy to use
- Simple security context

In comparison to Grids:

- Specific, complex execution environment
- Large-scale distributed with individual providers
- No well established business model
- N:m relationship between user and provider
- Complicated to use
- Complex security context

Scientific challenges

- **Whether Clouds or Grids…**
  …the dynamic management remains a crucial functionality:
  - In allocating services to a remote system,
  - and managing services on the provider side
- **Considering new different optimization goals:**
  - Performance
  - Cost
  - Reliability
  - Minimizing the resource footprint (!)
  - Power Consumption (!)
- **Coping with scalability:**
  - Managing x1000s of services remains difficult
  - Managing many business relationships is complex
- **Coordination challenge:**
  - Single services are typically not sufficient, but require the efficient orchestration in a business process/workflow

Practical challenges

- **Lack of a common management framework**
- **Different Cloud and Grid environments**
  - **Lack of standards**
    - OVF is not sufficient, but we need also higher level management features
    - Not in the view of vendors
- **Coping with complexity!**
- **Scalability, optimization, dynamicity, reliability**
  - **Basic research still needed for core problems**
    - How to manage large-scale SOA/SOI infrastructure
    - Considering CPU, network, and storage
    - How to manage relationships between business processes

Transition to SLA Management

There is a clear trend to use Service-Level Agreements (SLAs) as a means for resource management.

- Combines industrial requirements with scientific application scenarios.

Performance qualities are initially agreed on in an SLA (=contract)

- between user/application and provider/resource
- There can be several SLAs involved on different levels
  - High-level, human-readable SLAs
  - Technical SLAs within the middleware to guarantee service provisioning.

SLAs provide a uniform approach for the client to

- specify resource and QoS requirements, while
- hiding from the client details about the resources,
- such as queue names and current workload

Motivation for the SLA@SOI project

Vision

- A business-ready service-oriented infrastructure empowering the service economy in a flexible and dependable way.

Business-readiness requires

- predictability & dependability ➔ prerequisite for acceptance & uptake of (new) services
- holistic SLA management ➔ transparent IT management
- automated negotiation ➔ dynamic, scalable service consumption

Impact on the knowledge economy

- decreased time to market for new services
- increased productivity and competitiveness
- lower entry barriers, especially for SMEs

Technical Motivation

Service Consumer
- dynamic demand for complex business solutions at low costs

→ Flexible usage Business Services

Software Provider
- SOAs provide un-precedented flexibility

→ Engineering of predictable services

Service Provider
- service economy requires dependable services

→ Automated SLA negotiation and management

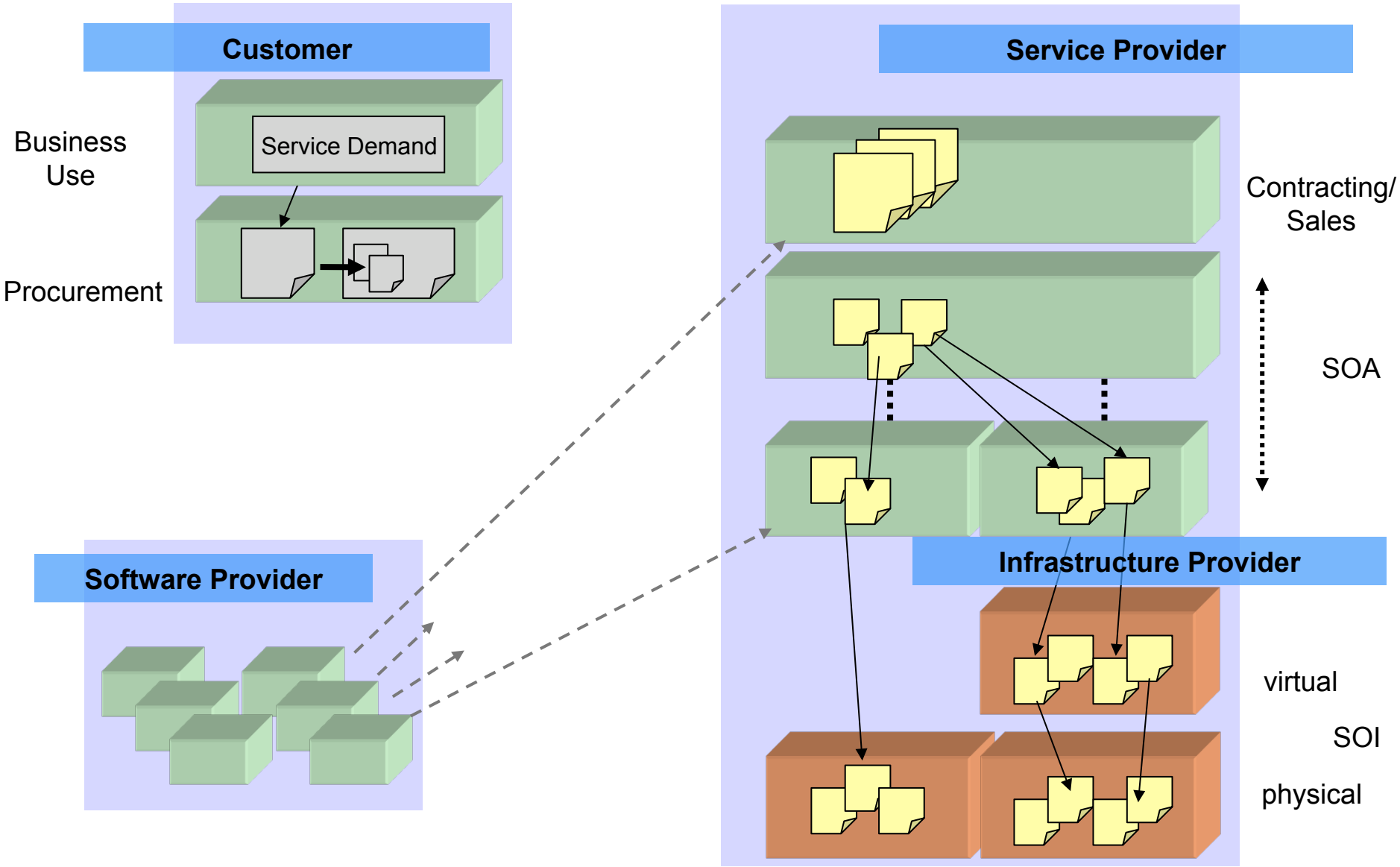
Infrastructure Provider
- virtualization technologies allow for adaptive SOIs

→ SLA enforcement via adaptive infrastructures

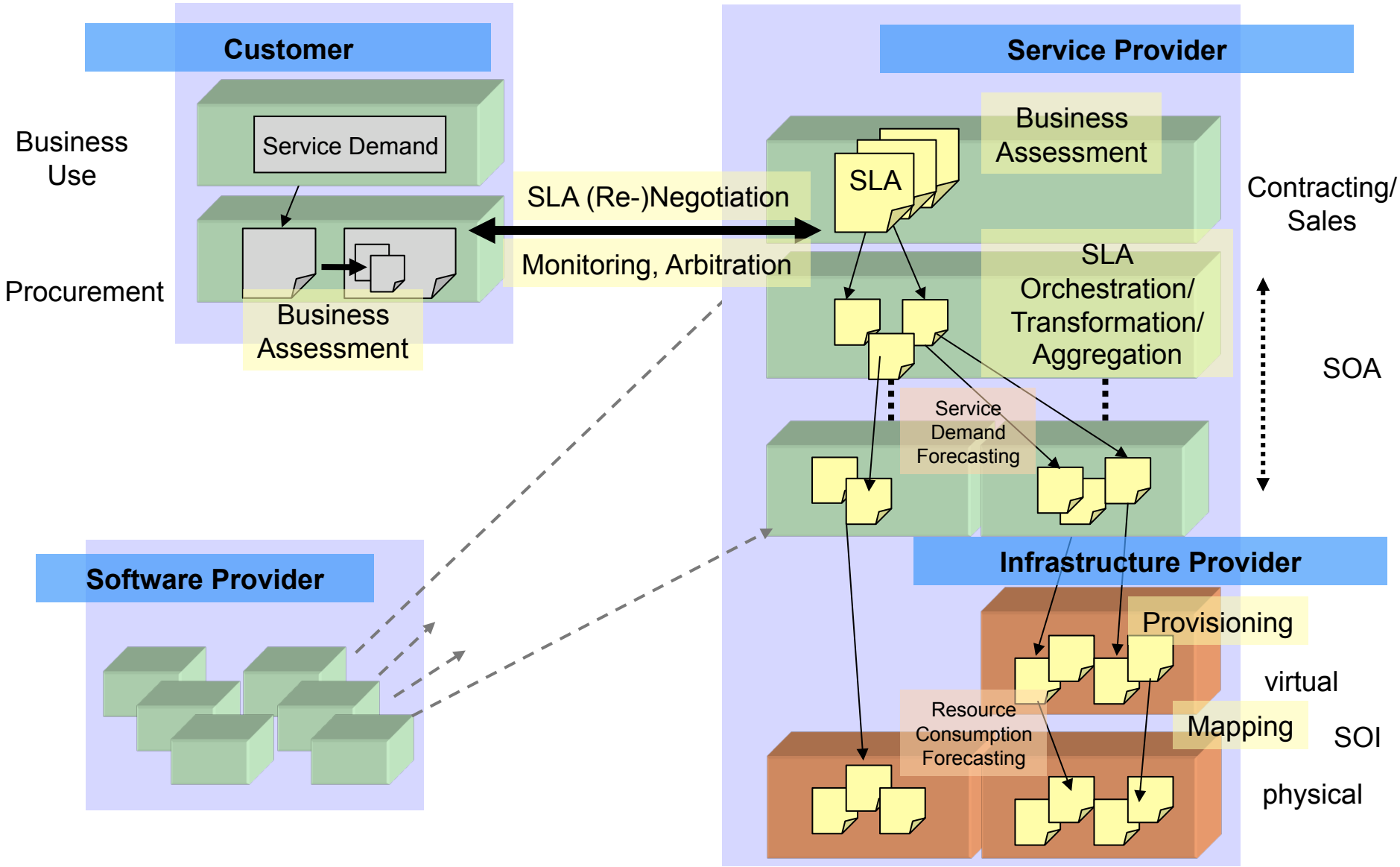
Vision of SLA@SOI

A business-ready service-oriented infrastructure empowering the service economy in a flexible and dependable way

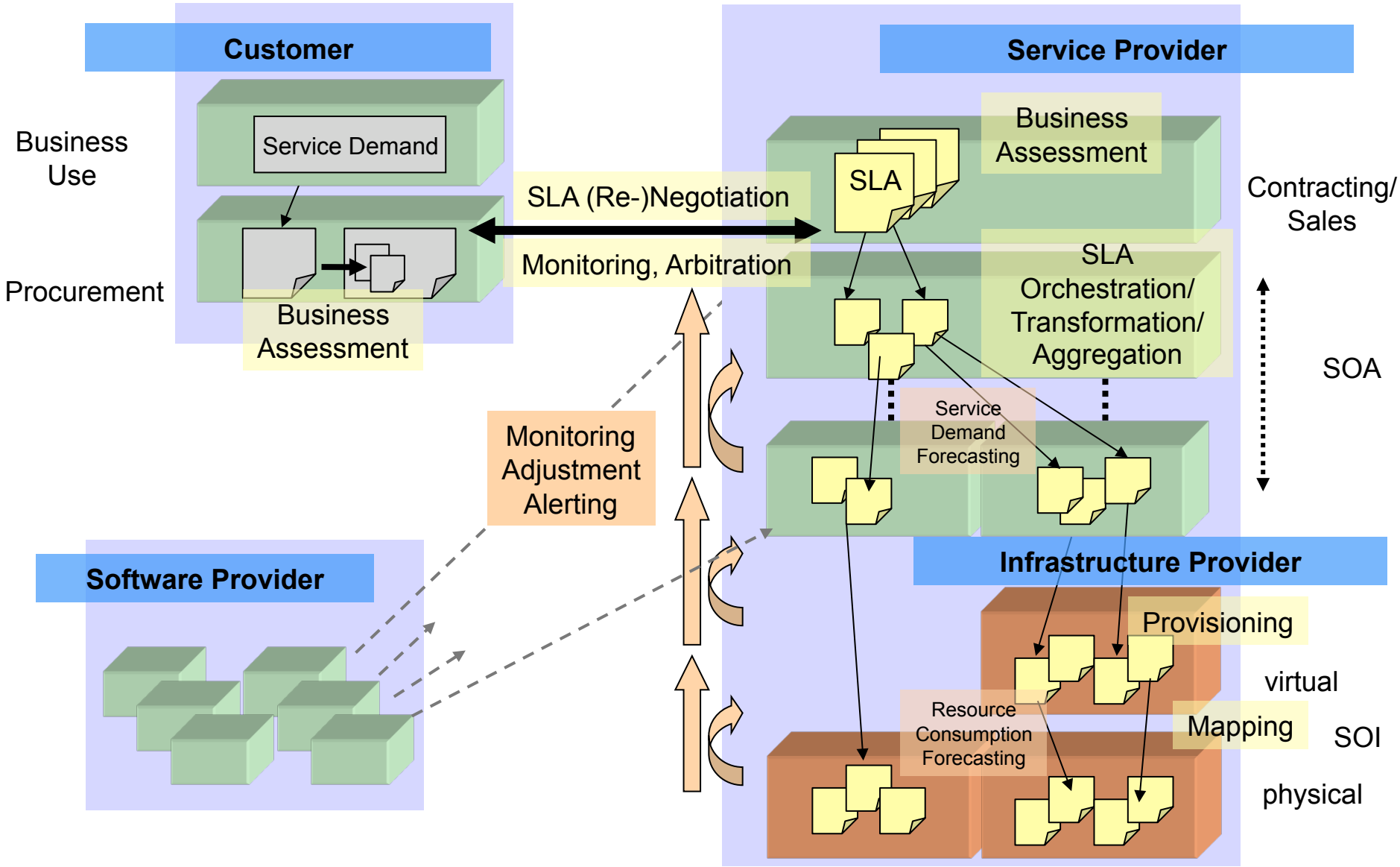
Envisioned Interaction



Envisioned Interaction



Envisioned Interaction



Project innovation

Main innovations

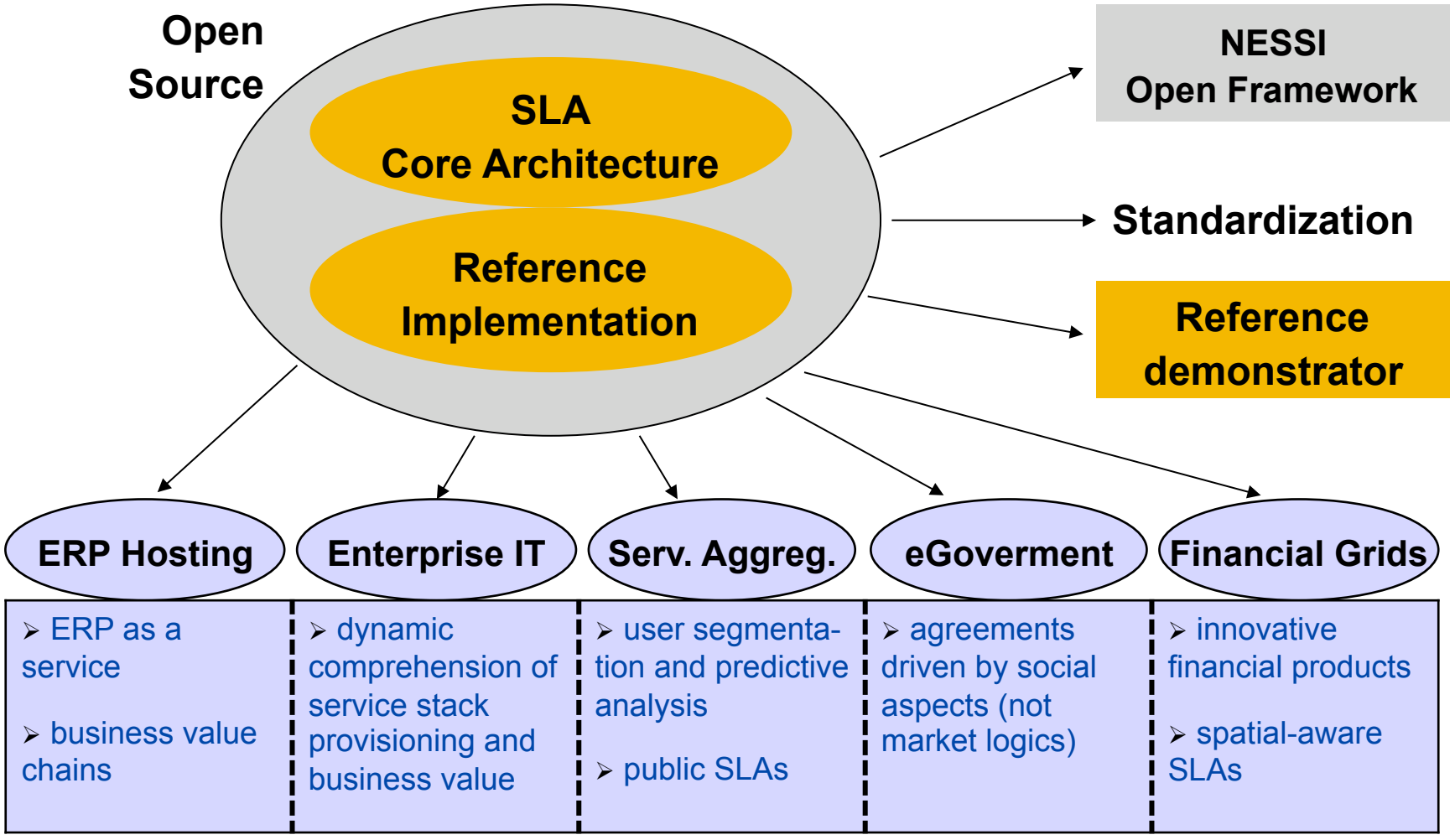
- **SLA management framework**
  - harmonizing perspectives of relevant stakeholders (software/service/infrastructure provider and customer)
  - standards for SLA specification and negotiation & systematic multi-layer SLA management (planning, optimization, and provisioning), monitoring and accounting
  - ➢ guaranteed QoS in a dynamic and end-to-end fashion via consistent SLA handling across IT stack

- **adaptive SLA-aware infrastructures**
  - standardized interfaces for adaptive infrastructures with harmonized access to different virtualization technologies.
  - advanced technologies for SLA enforcement on infrastructure level
  - ➢ efficient resource usage w/ reliable SLA enforcement at infrastructure level

- **engineering methods for predictable service-oriented systems**
  - modelling techniques and prediction tools for SOA and SOI components

- **business management suite for e-contracting**
  - covers complete business lifecycle of a service provisioning/delivery

Main project results



➔ **Industrial Evaluation Report: "How to run an SLA-driven business"**

Open Reference Case

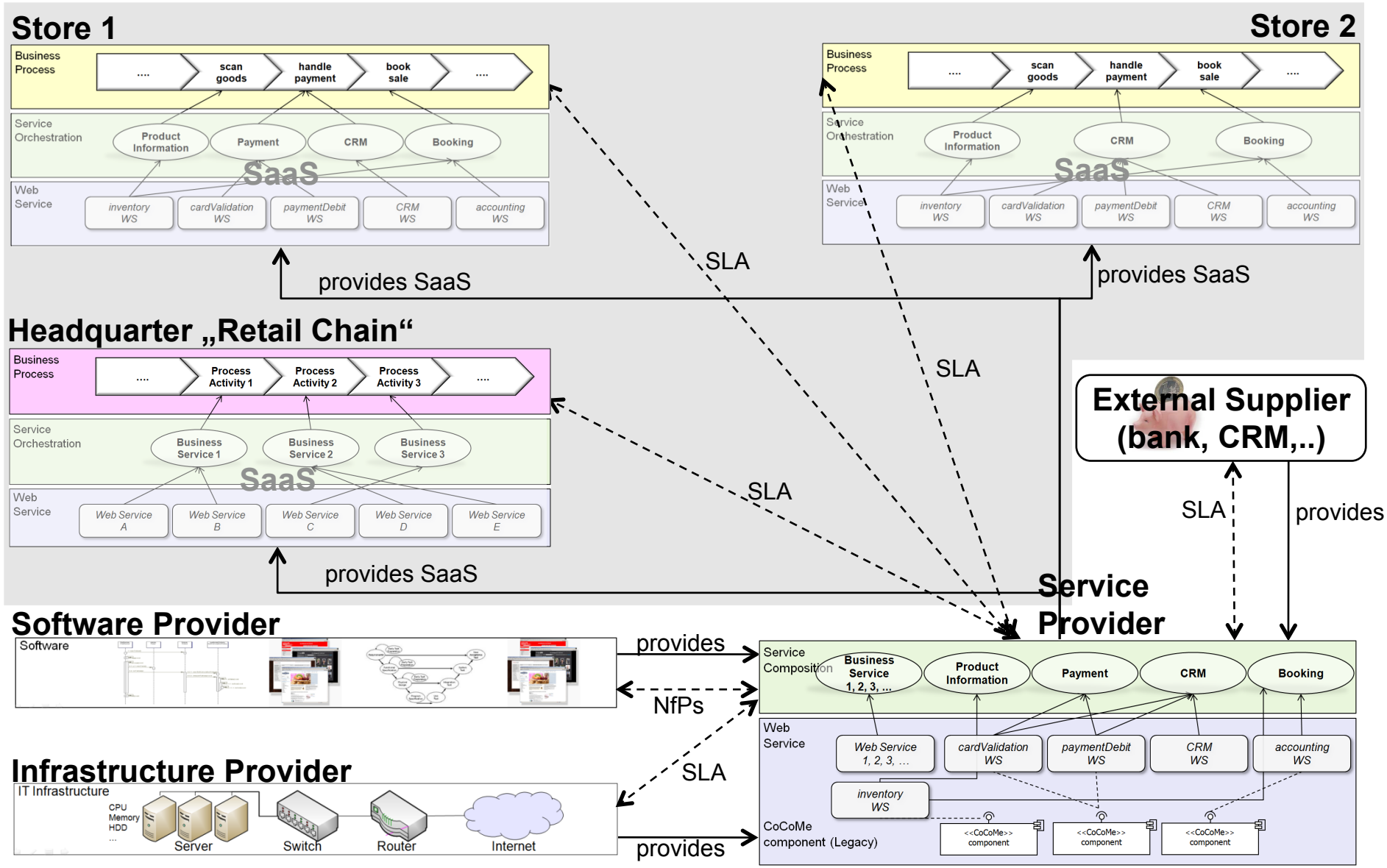
Idea

- a reference demo application (as open source)
- a reference SLA management demo scenario (as open source)
  - allows scientific community to compare results and to do subsequent research (replay and modification of SLA-experiments)
  - allows industrial stakeholders to get hands-on experience

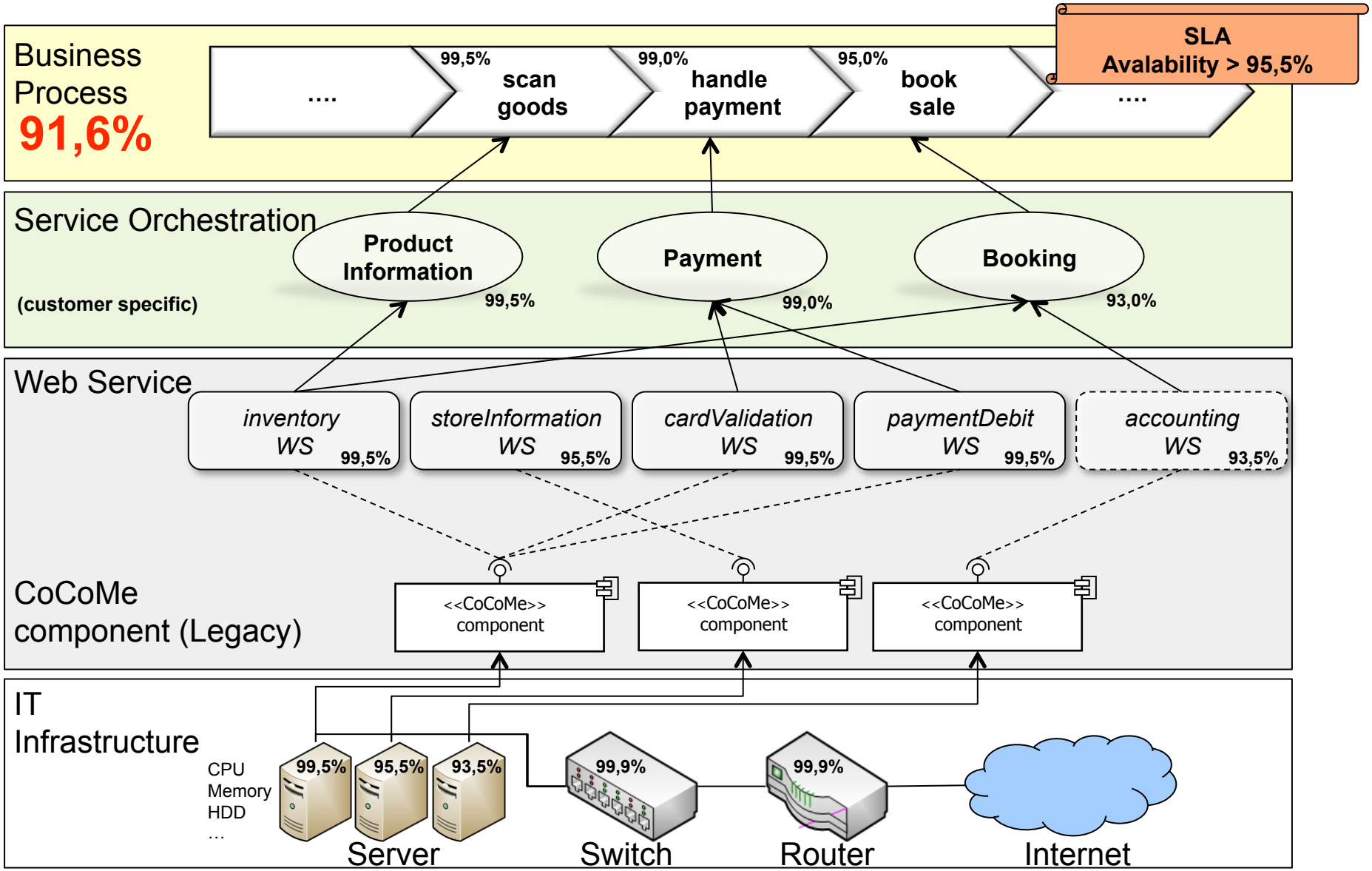
Features of the ORC

- an existing Java-based application
- precise specification & modelling available
- addresses management of retail chains (w/ core enterprise and several stores)
- different service selection and deployment options

Open Reference Case – Scenario



Open Reference Case – Scenario



System lifecycle

Conceptual system lifecycle

Engineering

- Engineering of system's building blocks (software & hardware)
- Modelling of system structure and (non-functional) behaviour
- Clean room experiments / measurements

Negotiation & Planning

- Translation of business SLA to technical level
  - concrete workload characterisation & non-functional requirements
- Translation & prediction & planning across whole business/IT stack
  - based on templates & historical traces & models

Provisioning

- Actual resource allocation

Operation

- Actual workloads
- SLA Monitoring & enforcement/adjustment
- Collection of historical data for improved model calibration / prediction

Analysis

- Detailed analysis of trace data
  - to update performance KPIs
  - to identify the necessity for manual efforts

Example Industrial Use Case: ERP Hosting

Partner & roles

- SAP: software & service provider
- Intel: infrastructure provider

Business context

- business applications (ERP, SCM, CRM, …)
- hosted in a Software-as-a-Service model

Technical service/SLA features

- Service types: A2A/B2B Web Services, UI services, business processes
- SLAs on performance, availability & security

Challenges

- complexity in terms of number of services, components, configurations & usage variants
- process flow partly implemented in a constraint model (no explicit flow information)
- underspecified environment: various parameters unknown at design time

Example Industrial Use Case: Enterprise IT

Partner & roles

- Intel: lead, use cases and infrastructure provider
- XLAB: specification, design & implementation contributions
- SAP: ERP process knowledge, monitoring capabilities
- UDO: design & implementation contributions

Business context

- SLA-aware dynamic provisioning of Enterprise IT Suite
- Supporting continuously evolving role and priorities of Enterprise IT
- CMF, ECF, Scalability + TCO

Technical service/SLA features

- Demonstrate tuned adaptation to provision competing demands appropriately
- SLAs potentially describing relative priority, response times & temporal variation
- Realistic infrastructure, services & workload simulation

Challenges

- Efficiently reconciling and dynamically provisioning a technology capability or investment relative to a specified process with business level metrics

Summary

- **Grids and Cloud Computing tackle the same problem space…**
- **…but from different directions!**
  - From the large-scale, distributed, being very complex…
  - From a simpler, limited use case, being a closed shop…
- **The future lies in between.**
- **Basic research work is still needed and needs major efforts.**
  - The core problems are complex.
  - Clouds will not sustain in the current niche if not moving up.
- **Current EC projects like RESERVOIR, SLA@SOI, NEXOF-RA show important activities to fill the landscape**
- **The scientific world needs to continue on the bleeding edge to pave the way for general adoption.**

Thank you!

Questions?

Contact:

Dr. Ramin Yahyapour
IT and Media Center
Technische Universität Dortmund

eMail: ramin.yahyapour@tu-dortmund.de