

ORCHESTRA

Why an Open Service-Oriented Architecture ?

Reiner Güttler, Ralf Denzer
Environmental Informatics Group

guettler@enviromatics.net

© EIG, ORCHESTRA Consortium

orchestra

Open Service-Oriented Architecture



- Why a Service-Oriented Architecture SOA ?
- Why Open?



ORCHESTRA Architecture: High Level Decisions



first steps in Software Architecture: high level decisions

- architectural principles (derived from high level requirements)
- choice of an architectural approach (architectural model)

experience: two major problems

- terms not defined precisely

flexible?

open?

... ?

... ?

scalable?

secure?

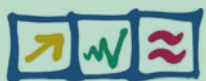
- no systematic justification

why flexible?

why ... ?

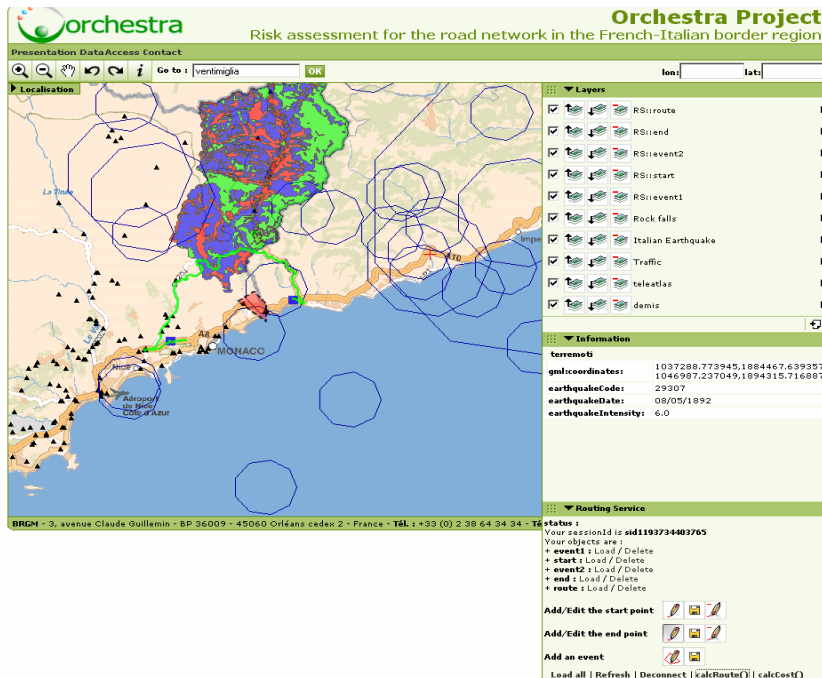
why open?

why ... ?



ORCHESTRA : from Use Cases to Architectural Principles

starting point: use cases




Task: (ex.)

Estimation of impact of rerouting traffic, caused by a blockage to the road network in the French-Italian border region

Concrete Problems:

- business processes need transborder, trans-organisational integration
- many heterogenous subsystems
- access control
- ...

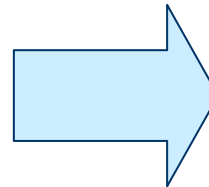
ORCHESTRA : from Use cases to Challenges



Task:
estimation of impact of rerouting traffic, caused by a blockage to the road network in the French-Italian border region

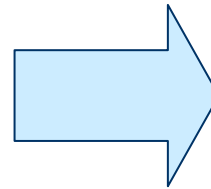
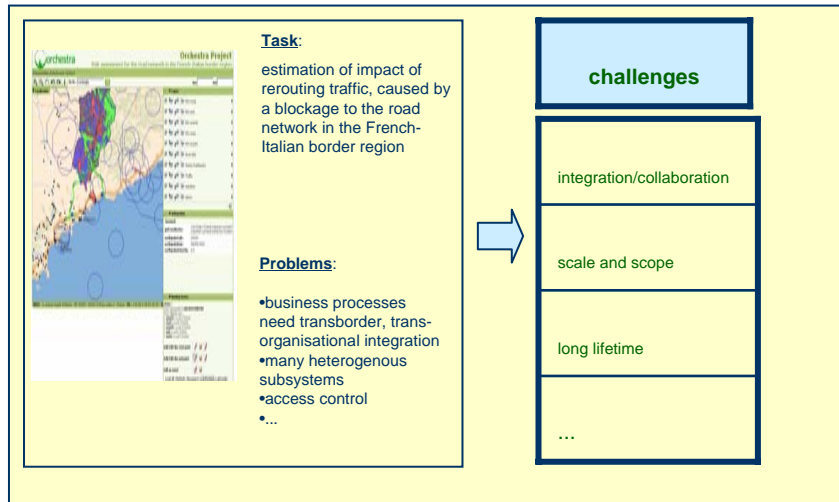
Problems:

- business processes need transborder, trans-organisational integration
- many heterogenous subsystems
- access control
- ...

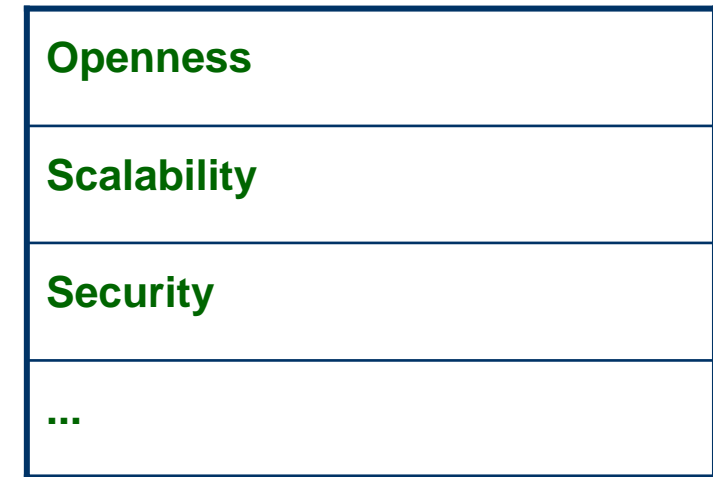


Challenges
Integration/Collaboration
Scale and Scope
Long Lifetime
...

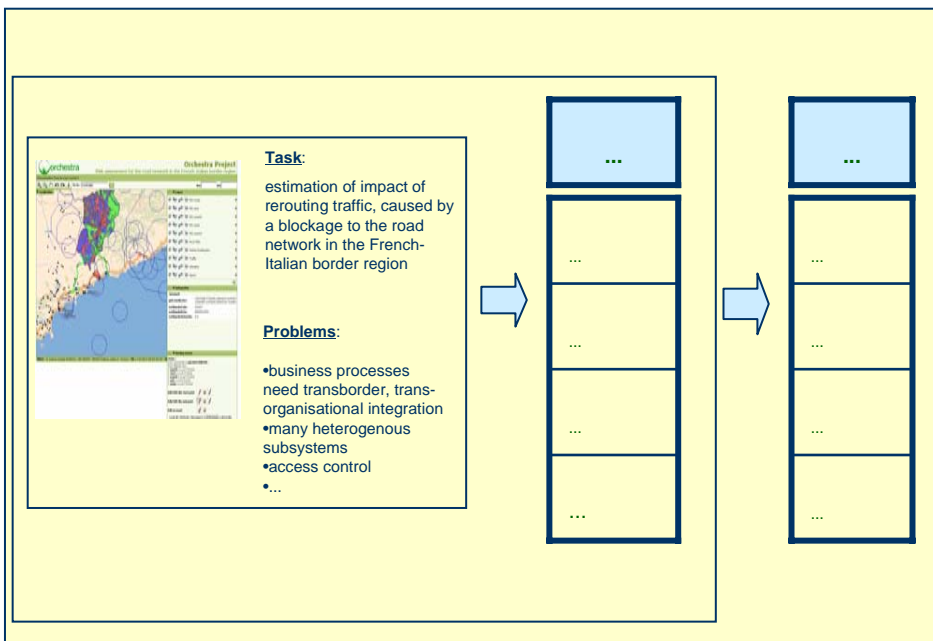
ORCHESTRA : from Challenges to High Level System Requirements



High Level System Requirements



ORCHESTRA : from High Level Requirements to Architectural Principles



Architectural Principles

Rigorous Use of Concepts and Standards

Loosely Coupled Components

Technology Independence

Evolutionary Development – Design for Change

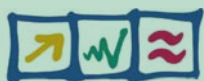
Component Architecture Independence

Generic Infrastructure

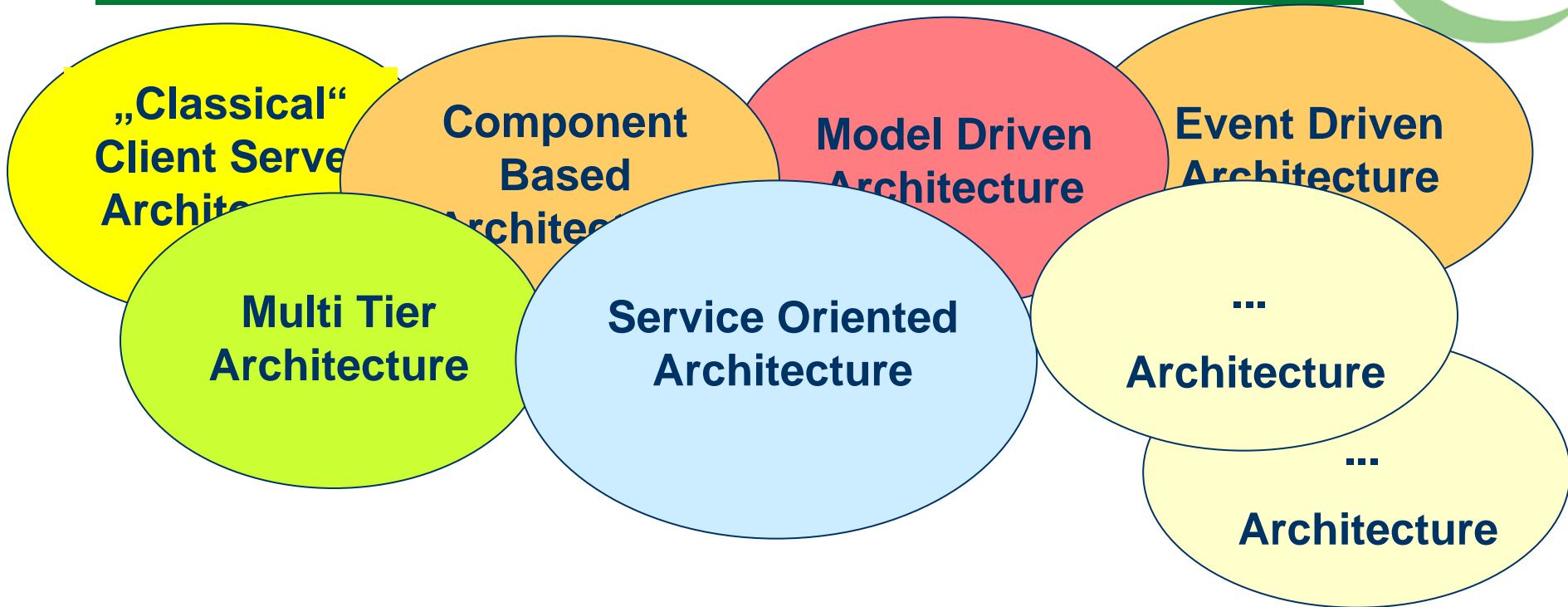
Self Describing Components

At each step:

- Definition of Terms
- Justification (how derived from previous step)



ORCHESTRA : Choice of an Architectural Approach (Architectural Model)



- many of them overlapping
- for some: clear common understanding
- for others: rather vague

ORCHESTRA: Choice of a Service oriented Architecture SOA, Arguments

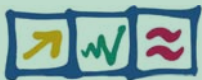
1. There is a well established common understanding of what an SOA is (e.g. OASIS, ...).
2. The agreed principles of an SOA fit well with the architectural principles of ORCHESTRA (high level requirements).



ORCHESTRA: Choice of a Service oriented Architecture SOA, Arguments



SOA Principles	ORCHESTRA Architectural Principles
Standardized Service Contract	Rigorous Definition and Use of Concepts and Standards - Self describing components
Service Loose Coupling	Loosely Coupled Components
Service Abstraction	Technology Independence
Service Reusability	Evolutionary Development - Design for Change, Generic Infrastructure
Service Autonomy	Component Architecture Independence
Service Discoverability	Self-describing Components



ORCHESTRA: Choice of a Service oriented Architecture SOA, Arguments



1. There is a well established common understanding of what an SOA is (e.g. OASIS, ...).
2. The agreed principles of an SOA fit well with the high level requirements of ORCHESTRA (architectural properties).
3. There exist a lot of software specifications and implementations in terms of services which should be usable (e.g. OGC Web Services)
4. There exists a Reference Model for Open Distributed Processing (ISO 10746) which can be interpreted to develop a geospatial SOA => a good starting point for the ORCHESTRA Architecture.



ORCHESTRA: Why an Open SOA?



What do we mean by „open“?

ORCHESTRA Glossary: „open architecture“

- Architecture which specifications are published and made freely available to interested vendors and users with a view of widespread adoption of the architecture. An open architecture makes use of existing standards where appropriate and possible and otherwise contributes to the evolution of relevant new standards.



ORCHESTRA: Why an Open SOA?



Impact from the perspective of this definition: open for what (concrete practical impact)

- contribution to standards & open for enhancement
- can serve as Reference Architecture
- allows vendor independent implementations and therefore
- enable & sustain multiplicity of applicable use cases
- ...



ORCHESTRA: An Open SOA!



Resume:

- An SOA is an appropriate software architecture approach for ORCHESTRA
- It should be “Open” in the sense of the ORCHESTRA definition

An Open Service Oriented Architecture is a good choice to meet the ORCHESTRA objectives



ORCHESTRA Architecture: High Level Decisions



IEEE Std 1471: Software Architecture: the fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.



ORCHESTRA Architecture: Architectural principles

Example of a Description



- Architectural independence describes the notion that existing information systems and information networks are independent of the new systems architecture in their architectural approach and vice versa.
- This means that
 - the new systems architecture does not impose any architectural patterns on existing information systems or information networks, for the purpose of them collaborating with the new system
 - no existing information system or information network can impose architectural patterns on the new systems architecture
 - the new system and existing information systems and information networks are architecturally decoupled.
- This will greatly improve the overall openness and acceptability of the new system, since participating organisations are not obliged to change their internal workflows, systems, etc. in order to collaborate with the new system.



ORCHESTRA Architecture: Architectural principles

Example of a Description



- The usage of self-describing components that provide context-sensitive formal and semantic descriptions of their interfaces can help to realise semantic interoperability. Components, such as data elements or models, should include descriptions of their critical characteristics and features, including sources, assumptions, etc. This information can be used to provide means to provide trace-, monitoring-, logging-facilities.



ORCHESTRA Architecture: SOA Principles

Example of a Description



- **Service Discoverability**

"Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted."

For services to be positioned as IT assets with repeatable ROI they need to be easily identified and understood when opportunities for reuse present themselves. The service design therefore needs to take the “communications quality” of the service and its individual capabilities into account, regardless of whether a discovery mechanism (such as a service registry) is an immediate part of the environment.

- From: Thomas ERL, „SOA – Principles of Service Design“, 2007

