TERMA

GSAW 2007

A Methodology for Mapping from a Service Oriented View of the Ground Segment to a Component Based View

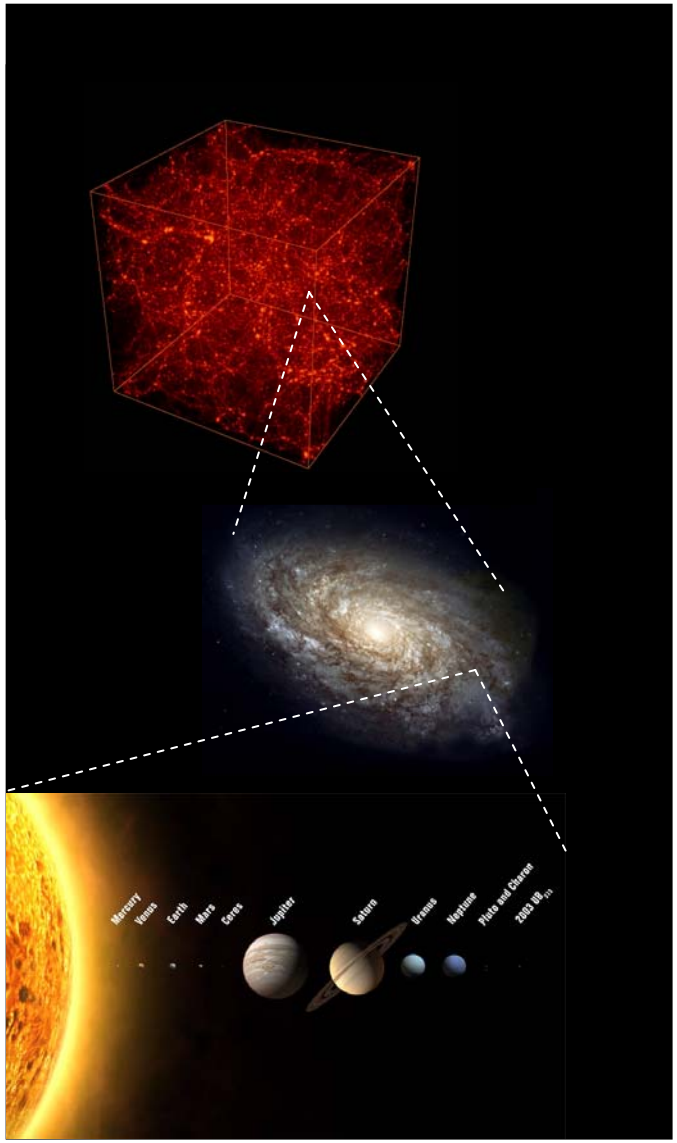
Project Manager/System Analyst, Gert Villemos (gev@terma.com), TERMA

Project Manager on the ESA contract to TERMA covering the development of the component based ESA/ESOC ground segment middleware, with the leadership of ESA/ESOC

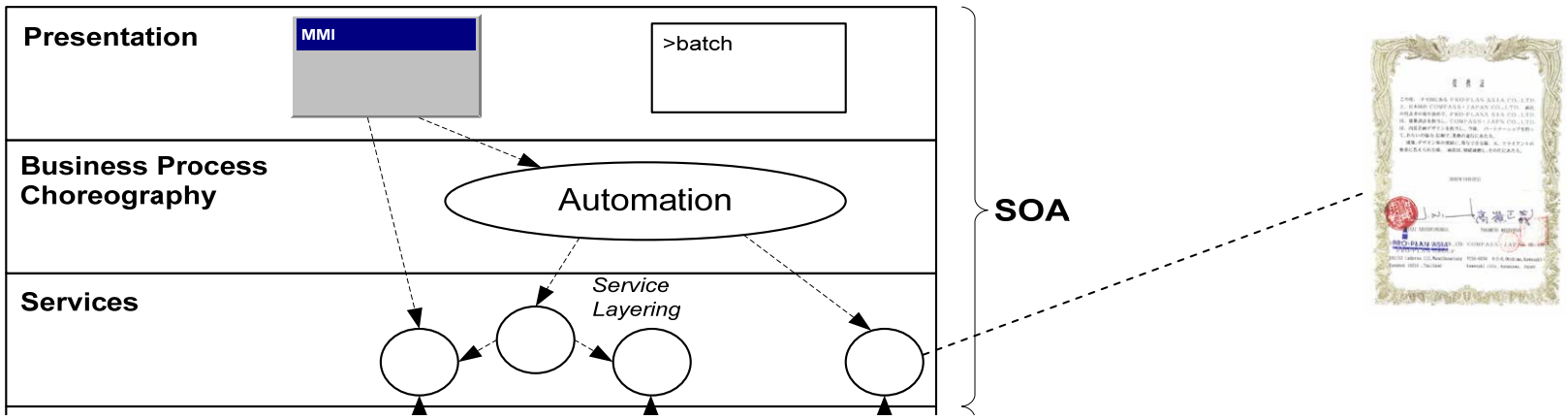
Roadmap

TERMA

- **Motivation**
  - **Service View.**
  - **Functional View.**
- Solution
  - Service Specification Standardization.
  - Automated Mapping.
- Added Benefits
  - Simplification.
  - Automated Interface Generation.
  - Automated Protocol Testing.
  - Generic Transfer Layer.
  - Generic Bridges.
- Conclusion.
- Questions.



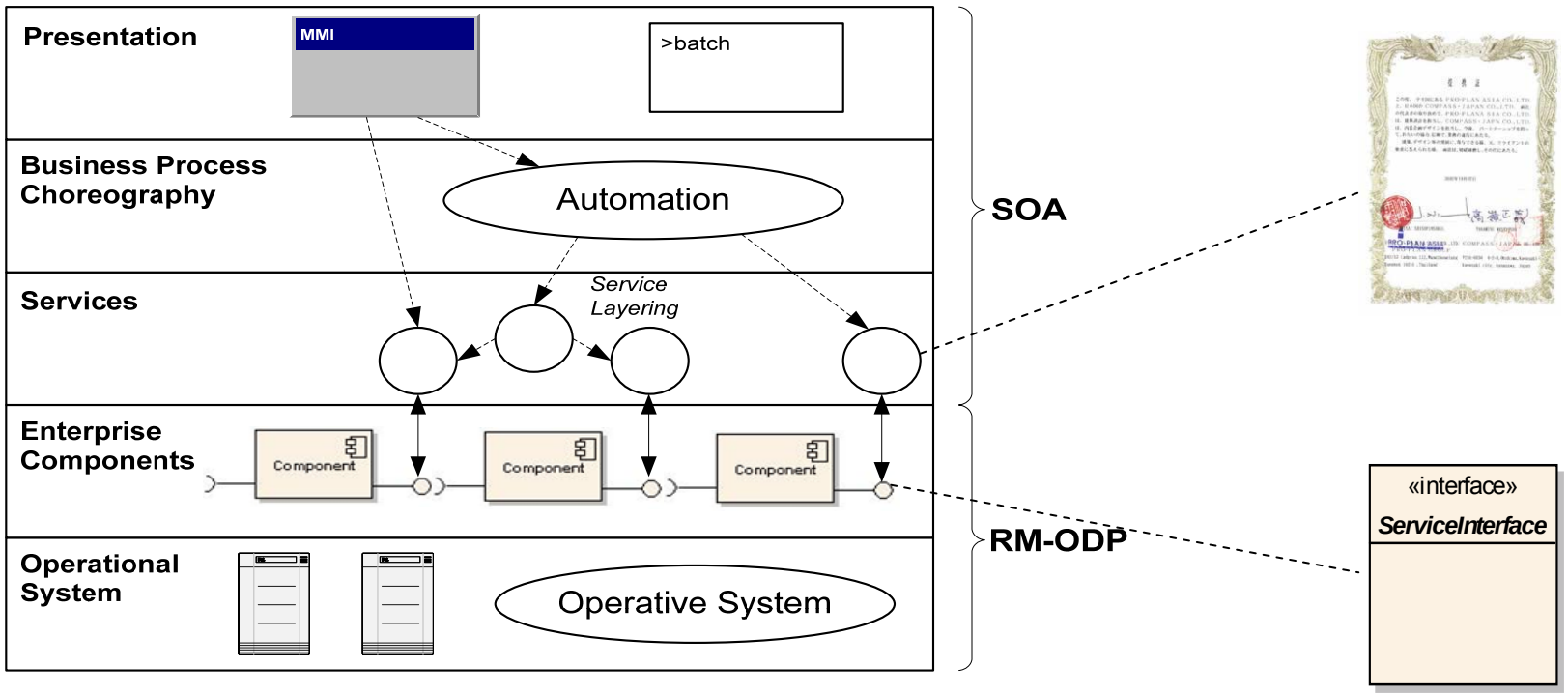
Service View



Services are typically standardized at highest level, such as CCSDS for the space domain. Services defines the message content, exchange protocol and the service state.

The service specification is a contract for the external behavior. The internal service implementation is not of importance, and can be changed, as long as the contract is maintained.

Functional View

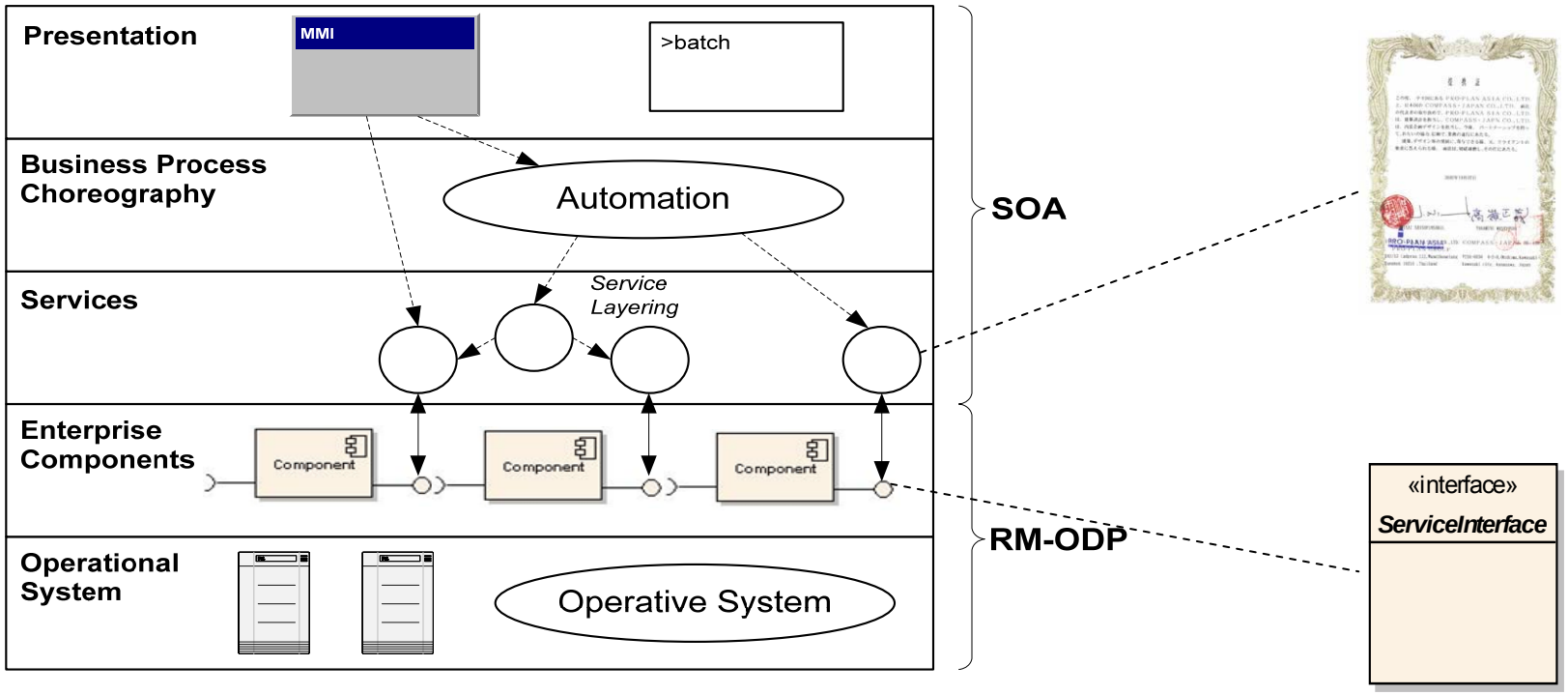


The services must be mapped to servants (endpoints) in the actual system. In a component based system, the services must be mapped to interfaces implemented by the components.

When using service busses, the mapping is to the interface of the service bus (e.g. WSDL). For end-to-end point communication it is to the actual interface of the component (e.g. IDL or ASN.1).

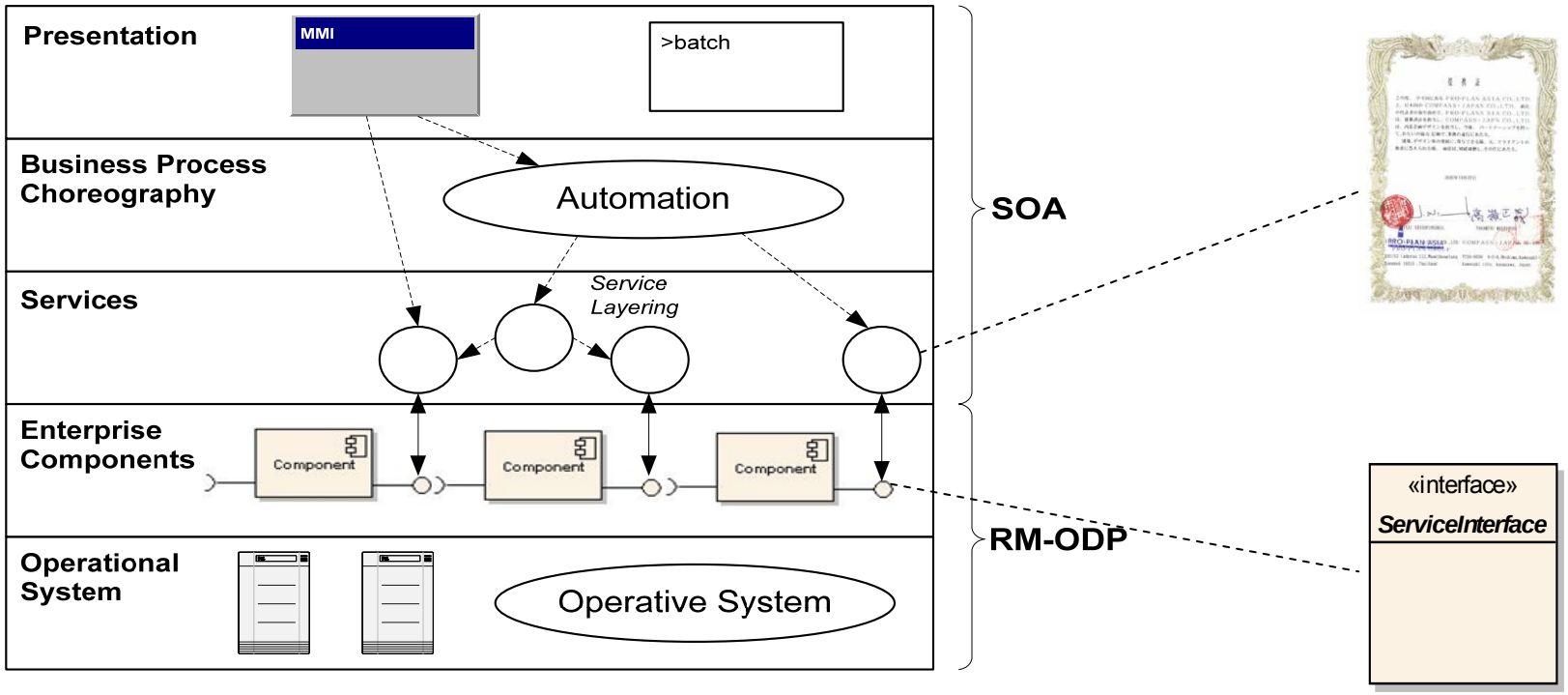
Code Base

TERMA



One of the benefits of SOA is that it allows the integration of components implemented in different ways. This sometimes lead the system architects to leave the development of the components completely to the contractors.

But reuse within the components limits the code base, optimizes development time and reduces costs. The next step after creating a SOA architecture, is therefore to create a component infrastructure.

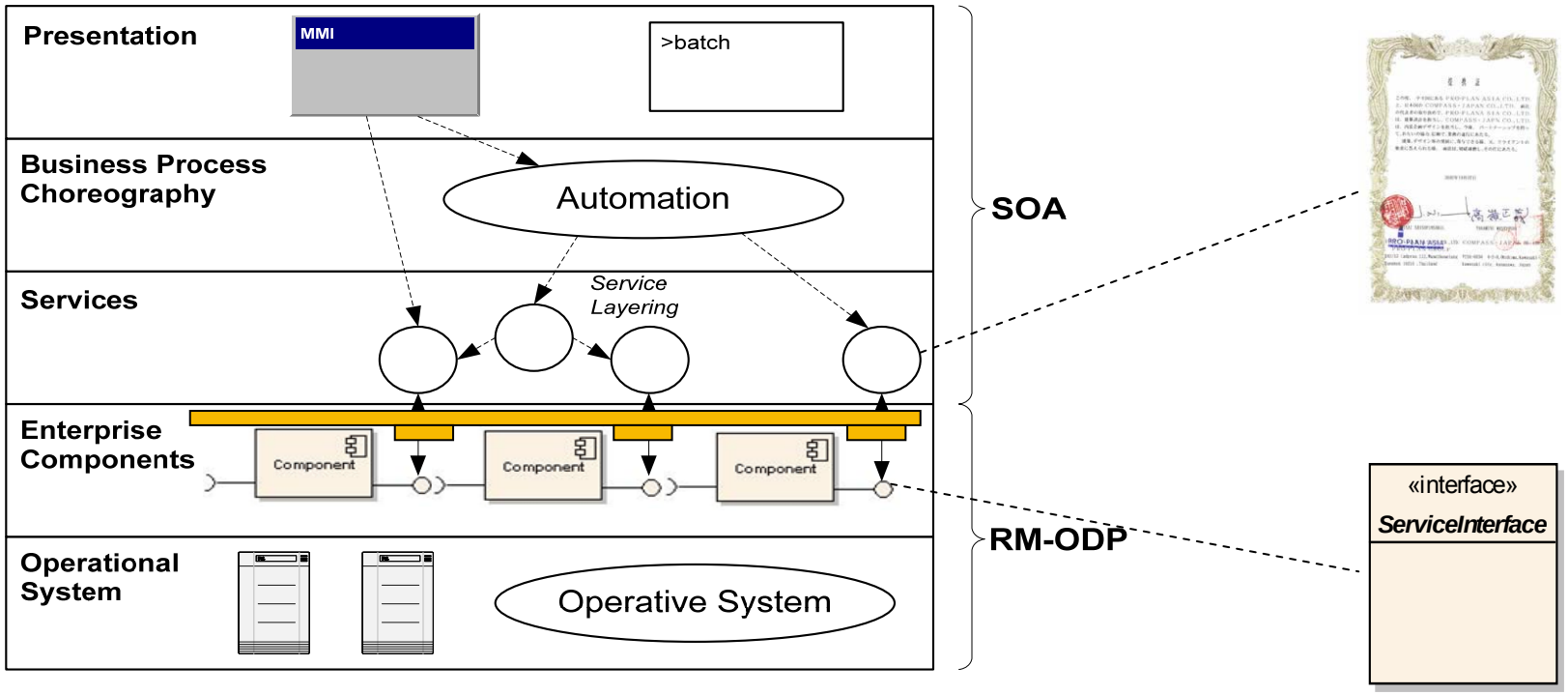


First step in creating the infrastructure, is identifying the generic elements.

Two problems makes this difficult

1. No harmonized way exist for specifying services within the space domain.
2. No standard way exist for creating the component interfaces from the service specification.

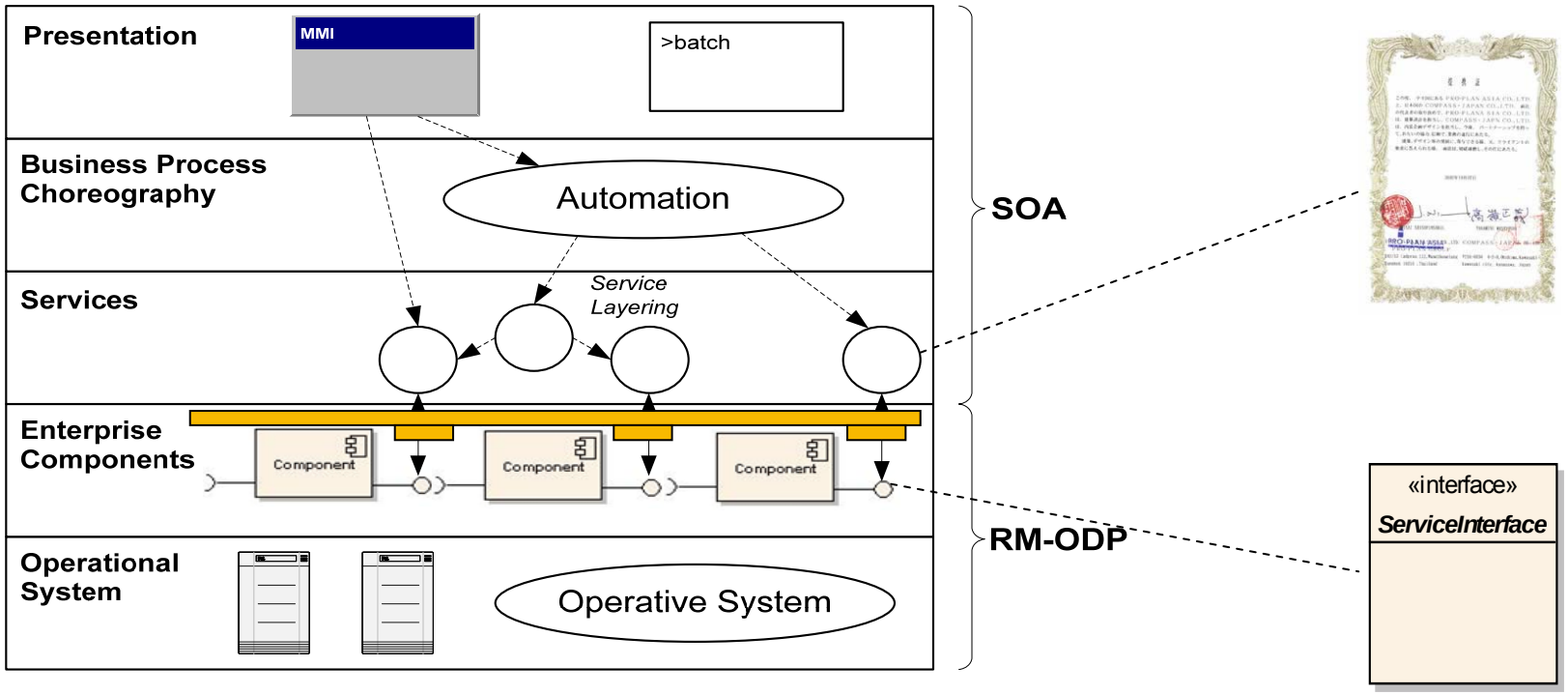
Service Mapping



Components are often integrated in the system through the introduction of adapters, allowing different systems to provide services in a (apparently) harmonized manner.

The functional interfaces of components are left to the developer. ‘You just have to make an adapter/driver’ syndrome…

Service Mapping



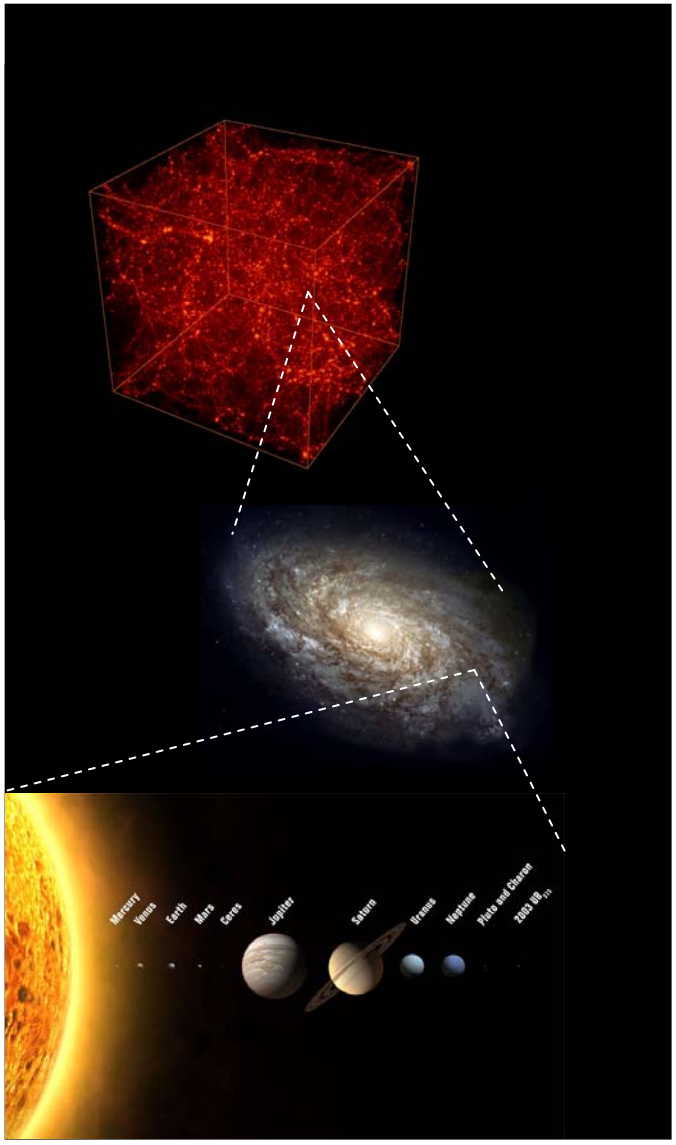
The adapter solves the problem for the architect, by moving it to the component implementer. Harmonization of functional interfaces, which would further reuse within the implementation, is not obtained.

In other words; the architecture becomes simpler, the implementation more complex and maintenance heavy.

Roadmap

TERMA

- Motivation
  - Service View.
  - Functional View.
- **Solution**
  - **Service Specification Standardization.**
  - **Automated Mapping.**
- Added Benefits
  - Simplification.
  - Automated Interface Generation.
  - Automated Protocol Testing.
  - Generic Transfer Layer.
  - Generic Bridges.
- Conclusion.
- Questions.



Solution

TERMA

ICDs are separated in two; the service specification, and the technology specific specification.

A standard way of specifying services was created, as well as a methodology for automated mapping to component interfaces.

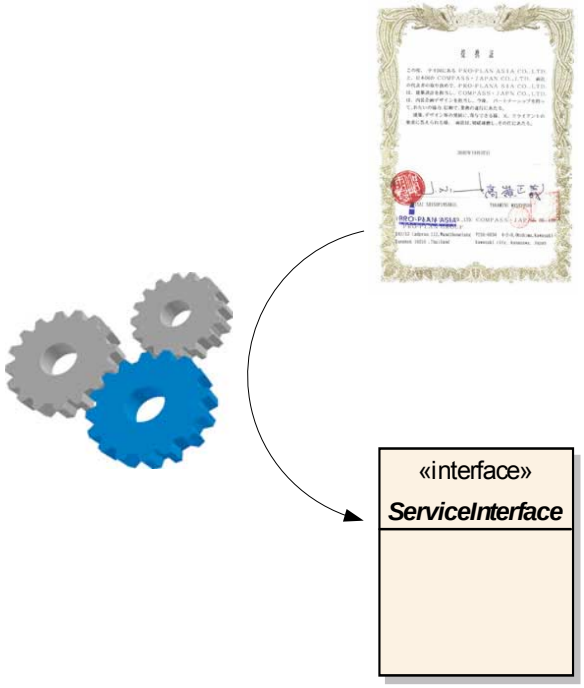
It solves the problems by

• Standardizing the creation of the functional interfaces, based on the service specification.

• Ensures the consistency and completeness of the functional interfaces (as long as the service specification is consistent and complete!).

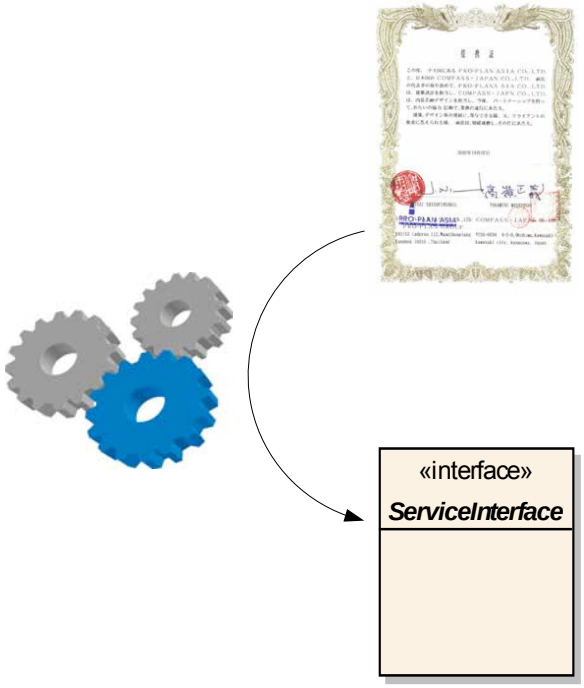
• Enables reuse within the components, as the same interface patterns can typically be implemented in the same manner.

• Make the functional interfaces predictable from the service specification, thereby enabling dynamic request invocation.



Prerequisites

To automate the mapping, three things were needed;



Prerequisites

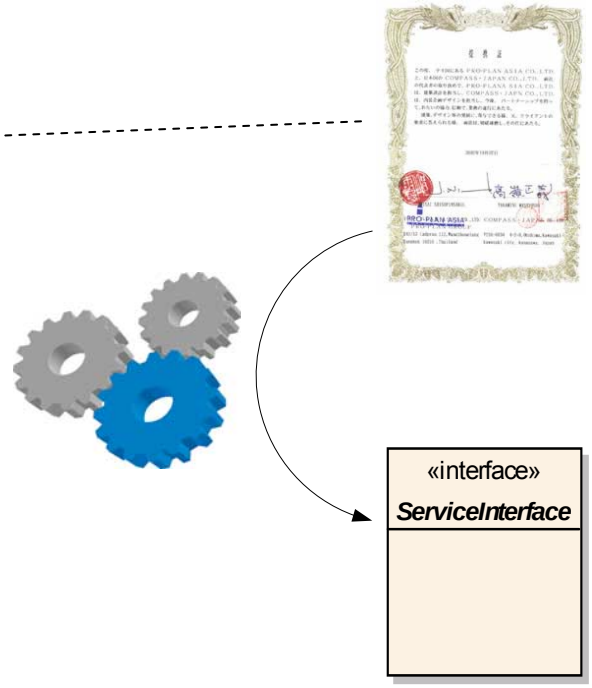
TERMA

To automate the mapping, three things were needed;

A standard way of specifying service.

Correspond to a Platform Independent Model (PIM) in MDA.

A UML 2.0 profile for a ‘Service Viewpoint‘ was defined.



Prerequisites

TERMA

To automate the mapping, three things were needed;

A standard way of specifying service.

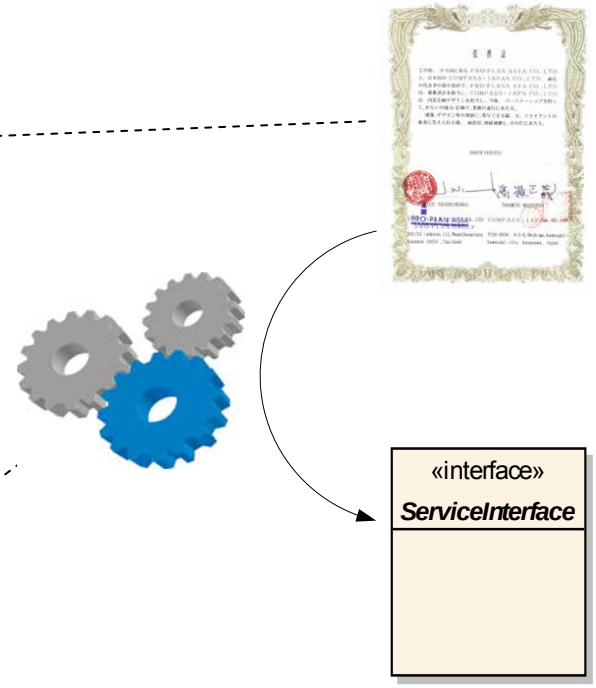
Correspond to a Platform Independent Model (PIM) in MDA.

A UML 2.0 profile for a ‘Service Viewpoint‘ was defined.

A set of mapping rules, transforming the services to functional interfaces.

Correspond to a Platform Model (PM) in MDA, transforming PIM to PSM.

A mapping to a UML 2.0 ‘Functional Viewpoint’ was defined.



Prerequisites

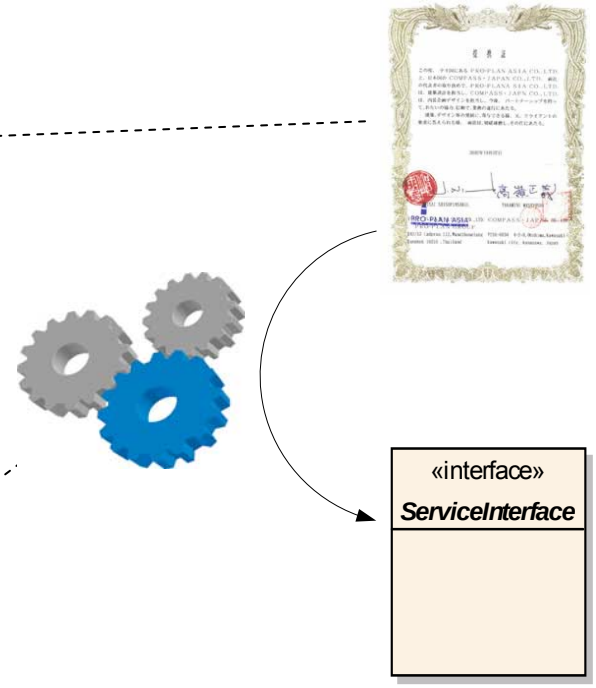
TERMA

To automate the mapping, three things were needed;

A standard way of specifying service.

Correspond to a Platform Independent Model (PIM) in MDA.

A UML 2.0 profile for a ‘Service Viewpoint‘ was defined.



A set of mapping rules, transforming the services to functional interfaces.

Correspond to a Platform Model (PM) in MDA, transforming PIM to PSM.

A mapping to a UML 2.0 ‘Functional Viewpoint’ was defined.

A common information model.

Specifying Services using the Service Viewpoint

TERMA

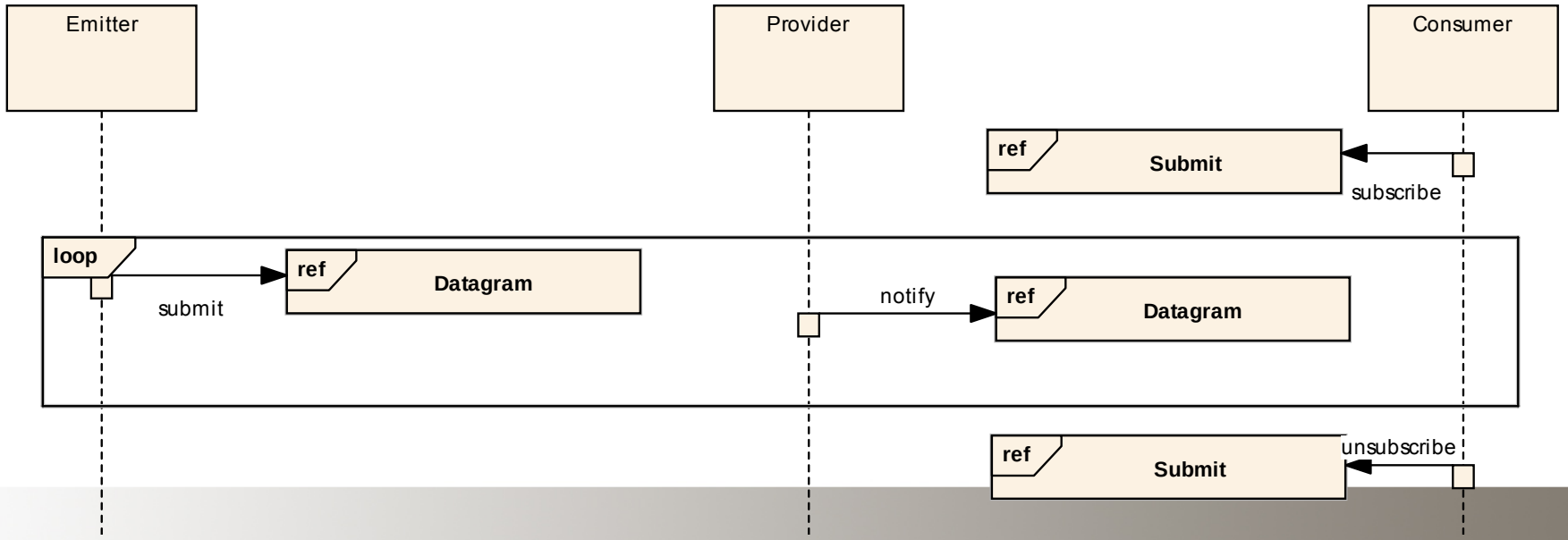
Within the service viewpoint, the service is specified using a UML 2.0 profile. The profile defines the types and stereotypes to be used within the model.

A minimum of assumptions have been made regarding the system reference architecture.

The service specification is based on the usage of

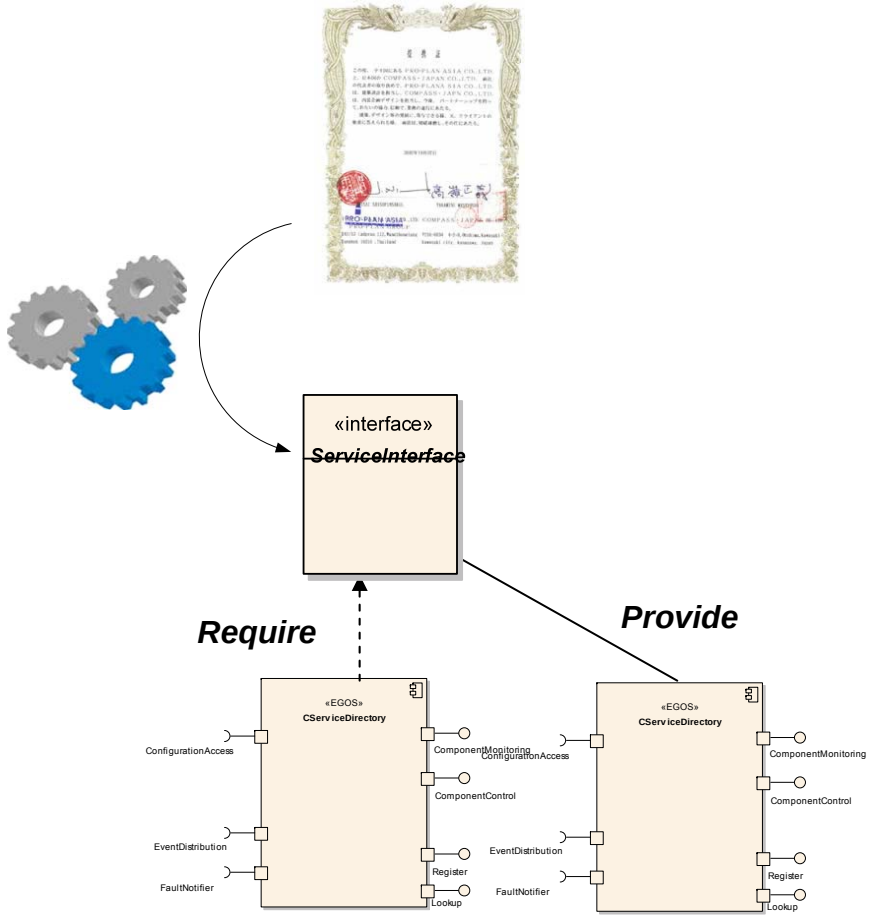
Message Exchange Patterns (MEP). Defines the sequence of message exchange. The content is irrelevant.

Interaction Patterns (INP). Defines typical sequence of usage of a service. The INP adds abstract content information, with parts intended specialized.



Mapping Rules

Ideally the mapping rules are the same across the complete ground segment, ensuring direct compatibility.

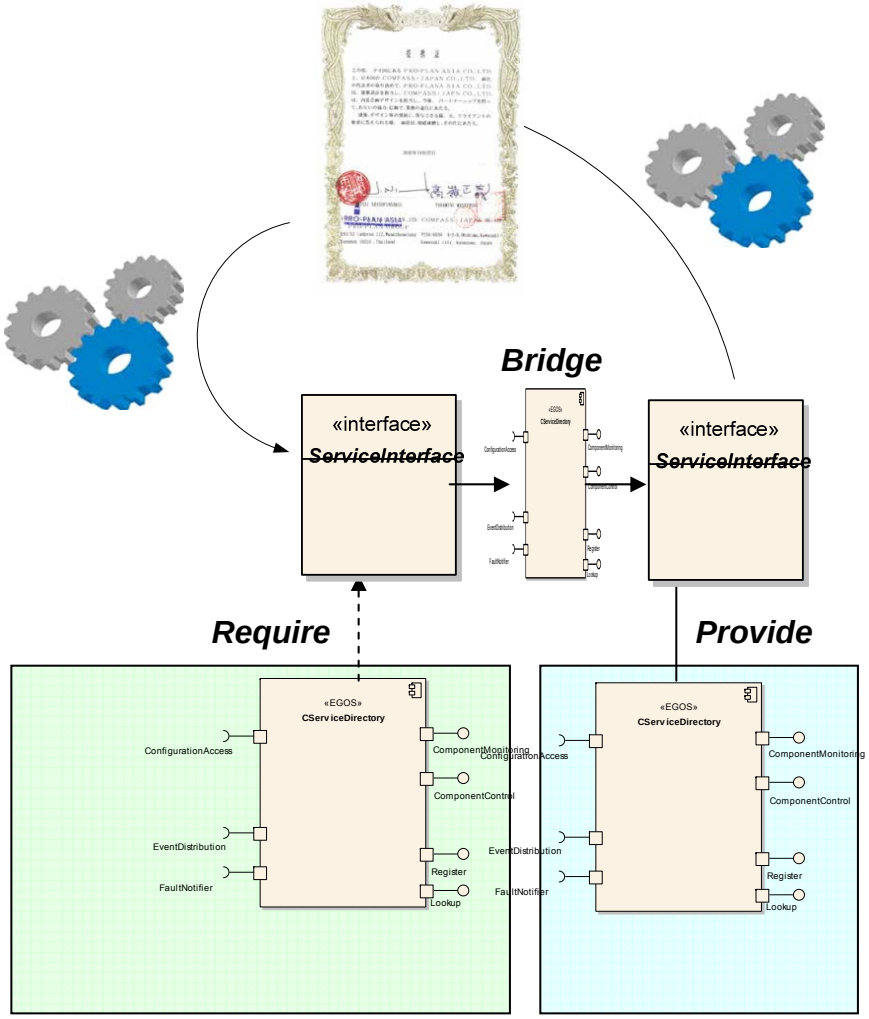


Mapping Rules

Ideally the mapping rules are the same across the complete ground segment, ensuring direct compatibility.

However in a federated system, different mappings may be applied by different organizations due to different concepts, naming techniques, technologies, etc, leading to different functional interfaces.

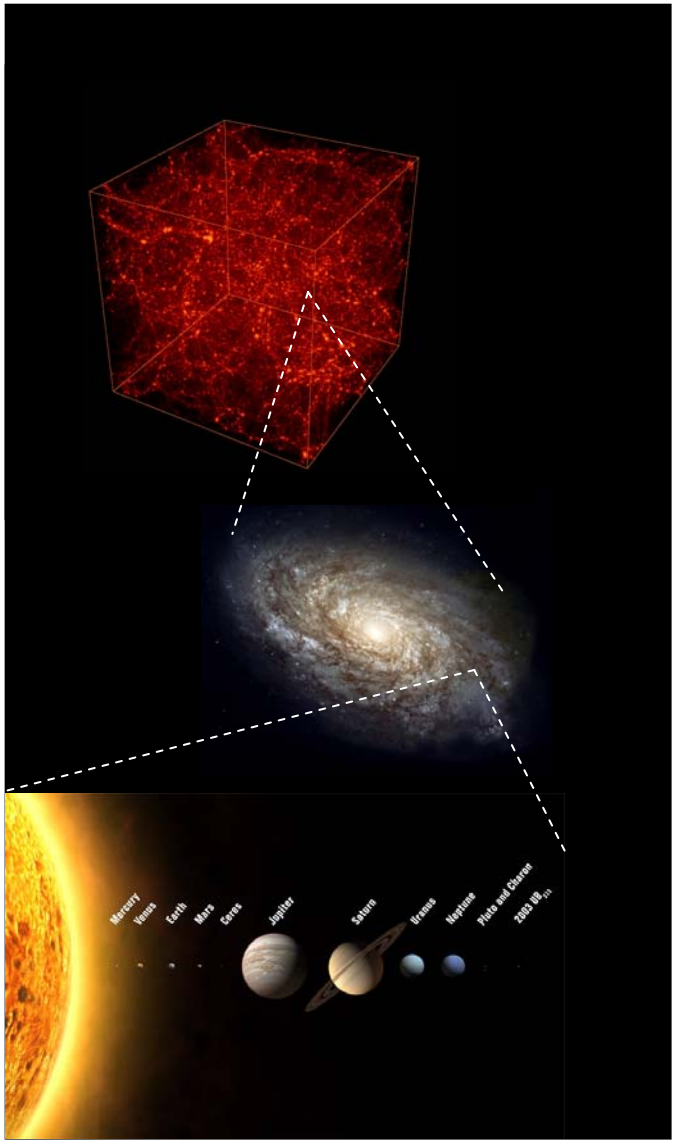
Bridging is needed…



Roadmap

TERMA

- Motivation
  - Service View.
  - Functional View.
- Solution
  - Service Specification Standardization.
  - Automated Mapping.
- **Benefits**
  - **Simplification of Service Specification.**
  - **Generic Transfer Layer.**
  - **Automated Protocol Testing.**
  - **Generic Bridges.**
- Conclusion.
- Questions.



Simplification of Service Specification

TERMA

Based on the service reference architecture, the specification of a service becomes the selection of generic operations and Interaction Patterns (INP), and the customization for the specific service. The customization is the definition of the service specific data.

Most of the protocol is defined ones within the generic parts. The size of a service specification has been reduced to less than 10 pages.

EventDistribution service;

Implements the PublisherSubscribe INP for the distribution of Events →

Provides the operations

*EventSubscribe [Subscribe] (in:DataFilter:**EventFilter**, in:Configuration, out:Result).*

*EventNotify [Notify] (in:Data:**Event**)*

EventUnsubscribe [Unsubscribe] (in: Void, out: Result)

Requires the operations

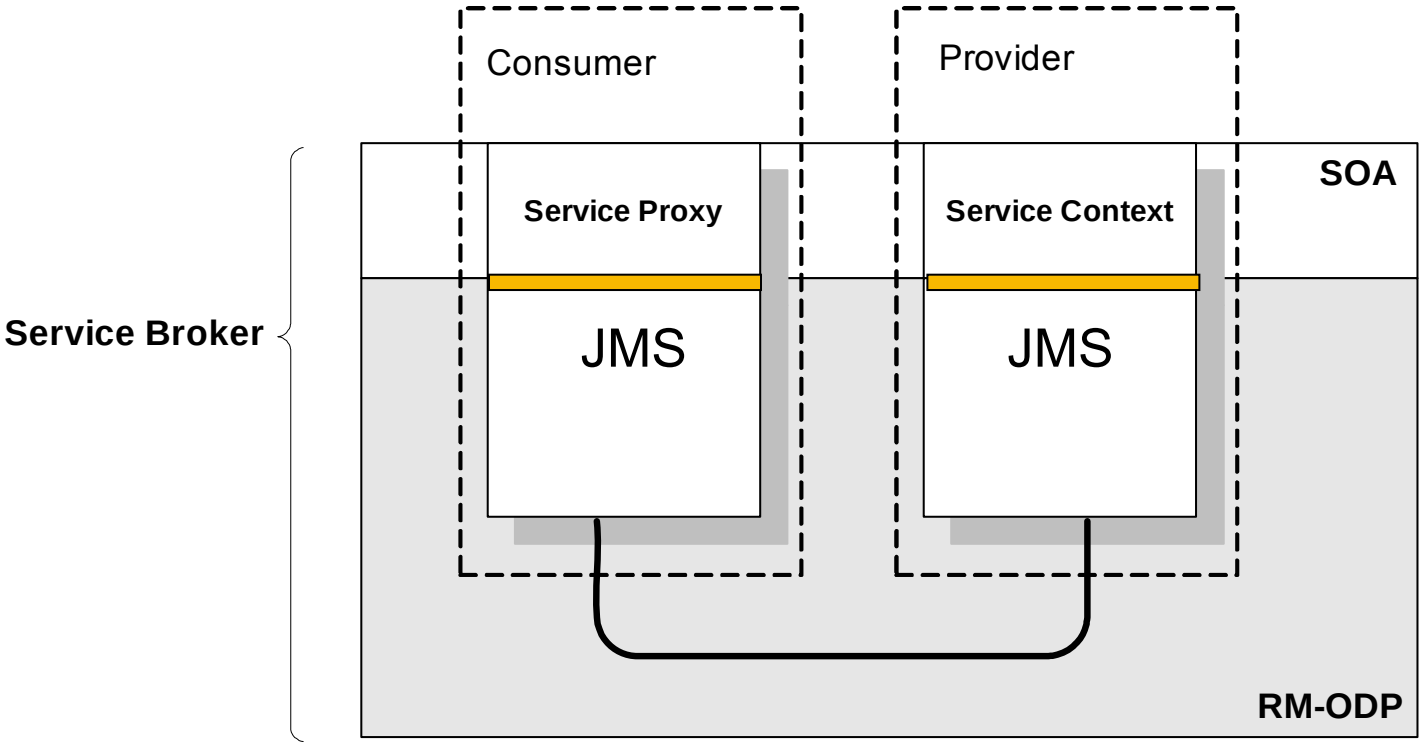
*EventNotify [Notify] (in:Data:**Event**)*

Generic Transfer Layer

TERMA

Based on the MEPs, the interface of a generic transfer layer has been defined. The layer can be implemented based on different technologies, including service message systems. The API to the consumer remains the same.

The service layer is build on top of the generic transfer layer, and uses the transfer interface.



Automated Protocol Testing

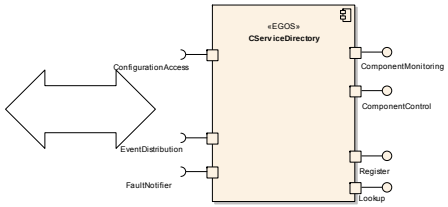
TERMA

The functional interfaces of the component is predictable, from the service specification.

A testbed has been defined, supporting the automated test of the service protocols. The testbed will dynamically create and invoke operations. The operations are created based on the service specification and the standard building blocks of the service reference architecture (Message Exchange Patterns, Interaction Patterns and Generic Operations).

The automated testbed operates at two levels

- Service protocol. The correctness of the returned data is not validate.
- Service. Service specific extensions of the testbed can test the specific service.



Generic Bridge

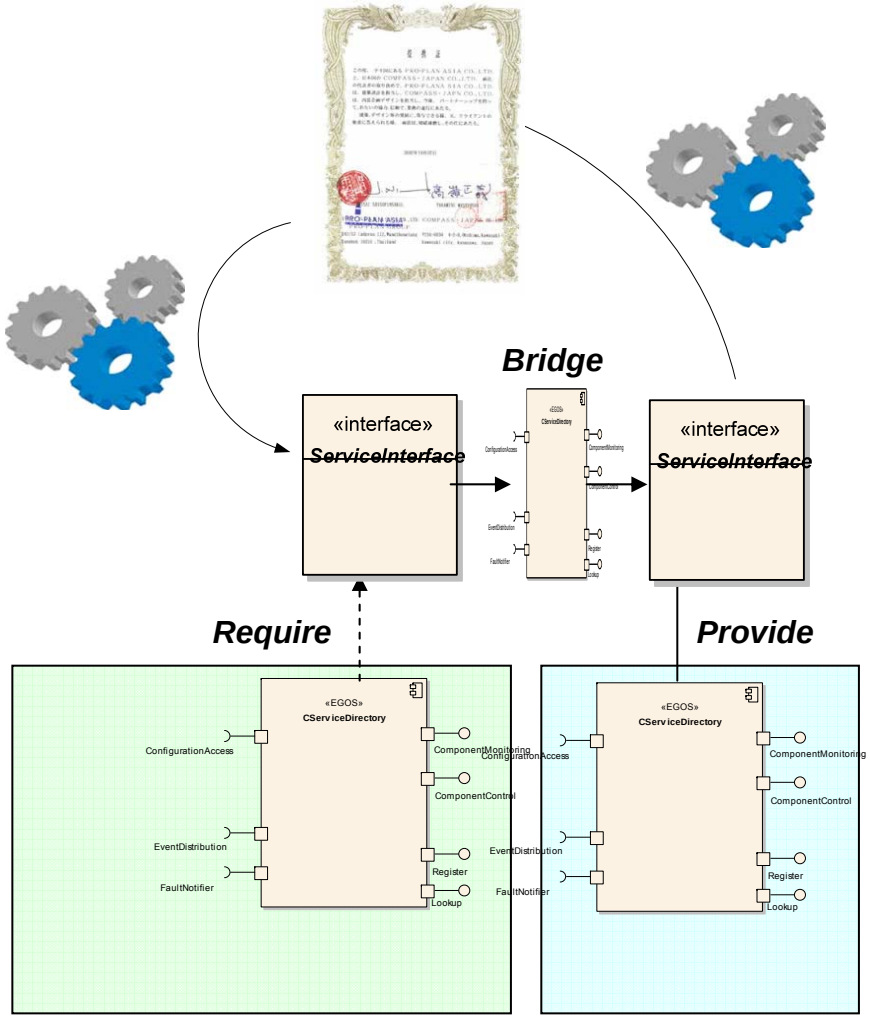
Ideally the same mapping from service specification to functional interfaces is used, in which case interoperability is immediate.

However organizational constraints and legacy systems typically imposes different technologies and/or conventions.

To allow interoperability, service bridges can be created, mapping from one interface to another, possibly using different technologies (CORBA to TCPIP).

The usage of a standard service reference architecture, ensures that such a bridging is possible.

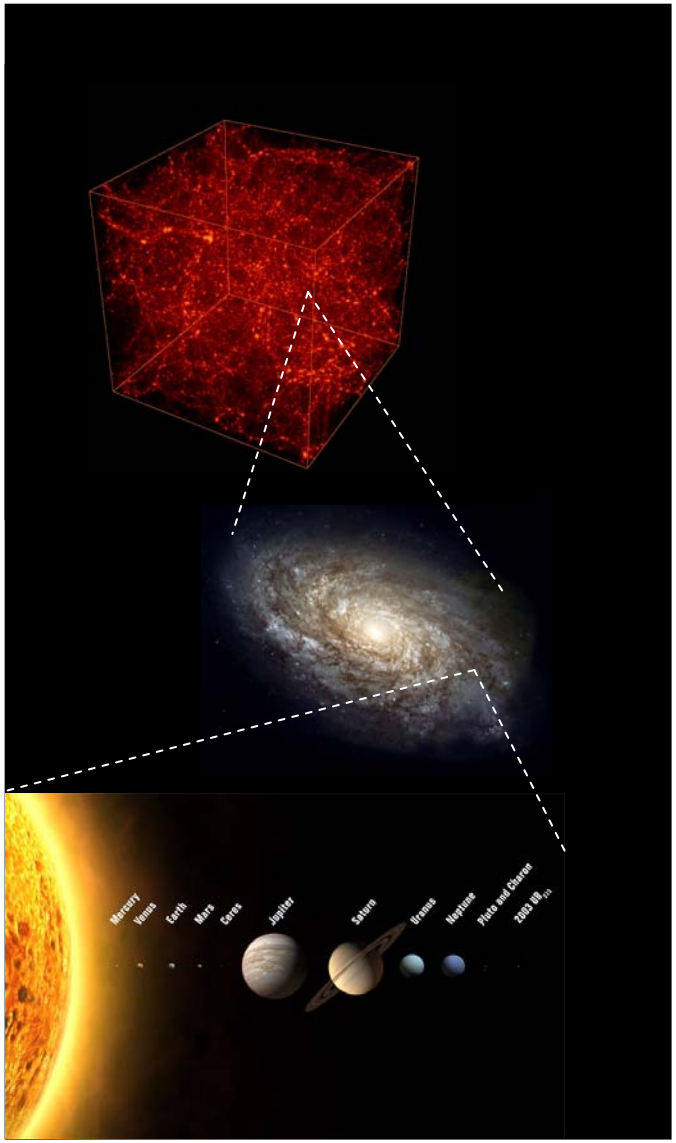
It also allows the creation of a generic bridging, based on differences in the mapping rules; ‘One-Bridge-Fits-All’.



Roadmap

TERMA

- Motivation
  - Service View.
  - Functional View.
- Solution
  - Service Specification Standardization.
  - Automated Mapping.
- Benefits
  - Simplification of Service Specification.
  - Generic Transfer Layer.
  - Automated Protocol Testing.
  - Generic Bridges.
- **Conclusion.**
- Questions.



Conclusion

TERMA

The methodology was developed to

Simplified Service Specification. Services are assembled based on standard, well proven building blocks. Specification effort is reduced to a minimum.

Automated Interface Generation. The component interfaces can be autogenerated.

But has proven to enable

Automated Component Testing. Based on the standard building blocks, a generic test tool can be build, capable of mimicking any service.

Generic Transfer Layer. A generic transfer layer can be build based on the MEP definitions.

Generic Bridging. Adaptation in a federated system through generic bridges.

The key is a standardized of a service reference architecture, including a data model.

Question?