REMICS Project

Talk in Timisoara

Parastoo.Mohagheghi@sintef.no
Andrey.Sadovykh@softeam.fr

14/11/2011

Outlines

1. Introduction
2. Underlying concepts
3. REMICS Project
4. Work in Progress
5. Demonstrations
6. Conclusions

1. INTRODUCTION

Motivation – Why to migrate?

- Long life-cycle of IT systems and Business Processes
  - Aerospace&Defense, Energy, Telecoms
  - Banking, Insurances, Retail
- Huge investments in legacy
  - Billions of lines of code
- Platform obsoleteness
  - Software: COBOL, Pascal, FORTRAN, Libraries
  - Hardware: IBM mainframes, HPs, Sun Stations etc.
- Skills disappearance
  - Personnel retirement
  - Education program centered on the cutting edge technologies





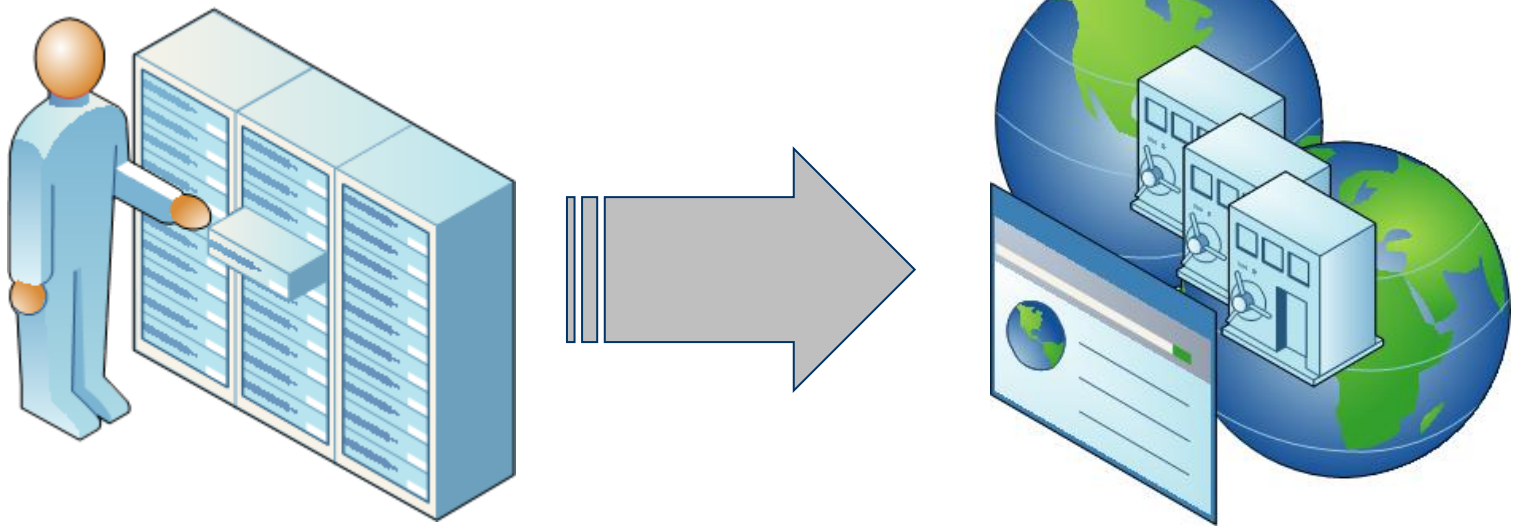
Is COBOL dying?

Example – COBOL code, Gartner reports

- 310 B LOC, 80% of actively used code, 75% business processes
- 80% of point-of-sales transactions
- 72 K shipping containers, 60 M patients, 500 M mobile phone users
- 5 B LOC written every year

- Less and less COBOL programmers trained
- Java programmer needs 20% of time to complete the same task than a COBOL programmer

REMICS objective



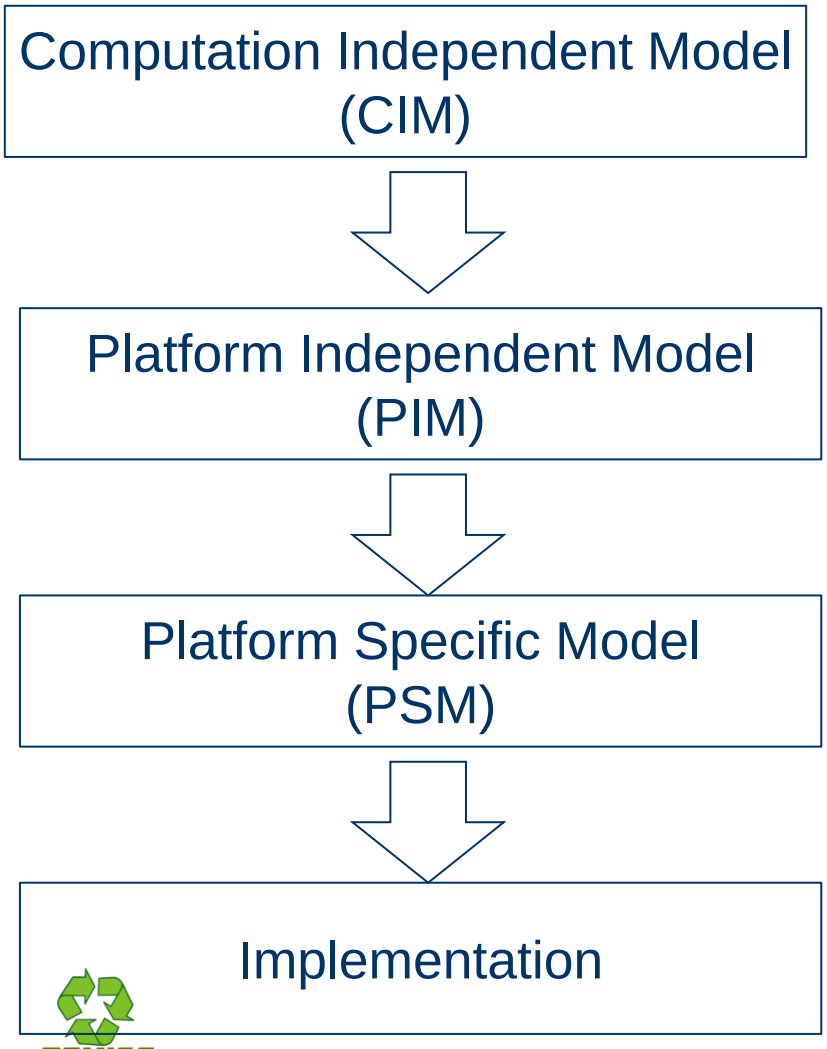
Legacy System:
e.g. Cobol, PL SQL

Modern Application

2. UNDERLYING CONCEPTS

1. Model-driven architecture
2. Architecture-driven modernization
3. Cloud computing

OMG Model-driven architecture (MDA)



- Business Models
- Semantics

- Business Logic of IT
- Data Structures
- Services, Components

- Technicalities and details
- Ex. data persistency modeling
- Ex. communication modeling

- Code
- Ex. Java, C++, C# code

Does MDA really work?

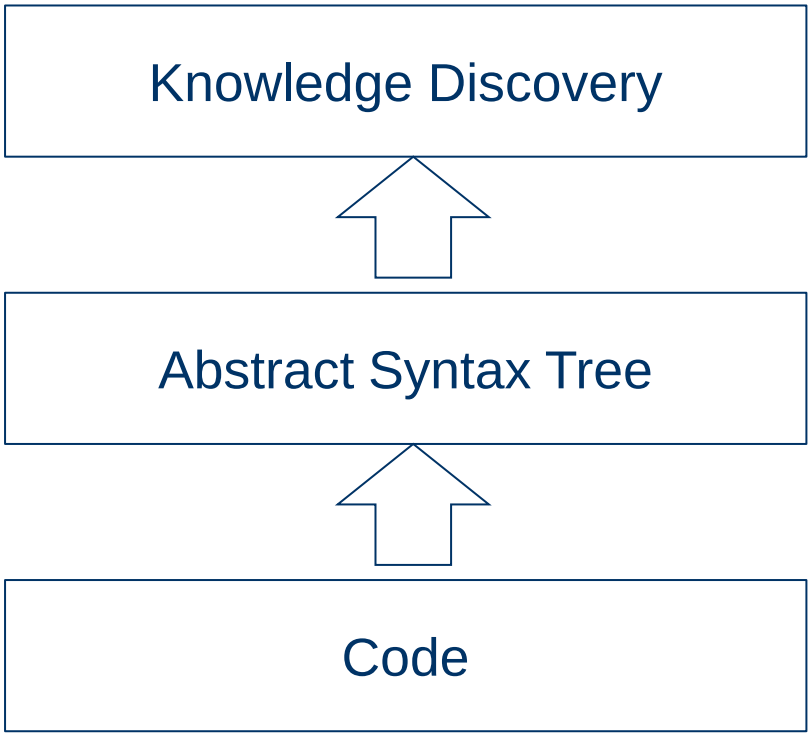
Few MDA examples

- Ericsson, Sweden
  - Typical base station life-cycle 30 years. Million LOCs
  - Keynote at ECMFA 2010
- DCNS Group, France
  - 10M LOC systems, 800 communicating components.
  - MDDays 2010 presentation
- Thalès Group, France
  - Embedded systems methodologies based on MDA

- Tool providers: Netfective, Mendix, Web Ratio, ModelioSoft, Obeo, Mentor Graphics, …

Architecture-Driven Modernization

Architecture-driven modernization (ADM)



- Recovery of models from legacy artifacts

Cloud Computing

What is cloud computing?

- “on-demand network access to a shared pool of configurable computing resources”*

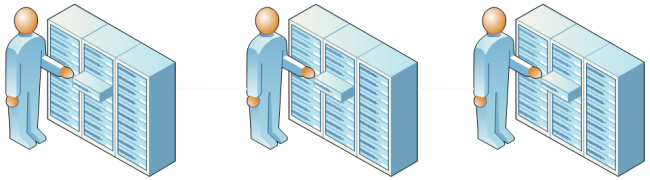
Cloud Services

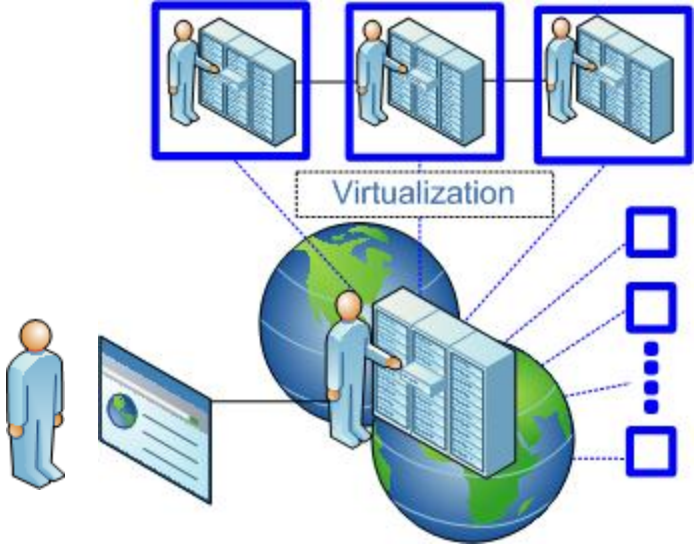
Infrastructure Providers

Virtualization Technologies for Cloud

*National Institute of Standards and Technology (NIST), http://csrc.nist.gov

What is the difference?





- <30% server utilization
- Duplicated systems
- Long procurement
- High administration costs

- >60% server utilization
- Scalability and elasticity
- Agility
  - Service composition
  - Pay per use
- Outsourced administration

Tremendous cost savings – 40%-80 %

Cloud computing, is it for real?

Cloud computing in numbers

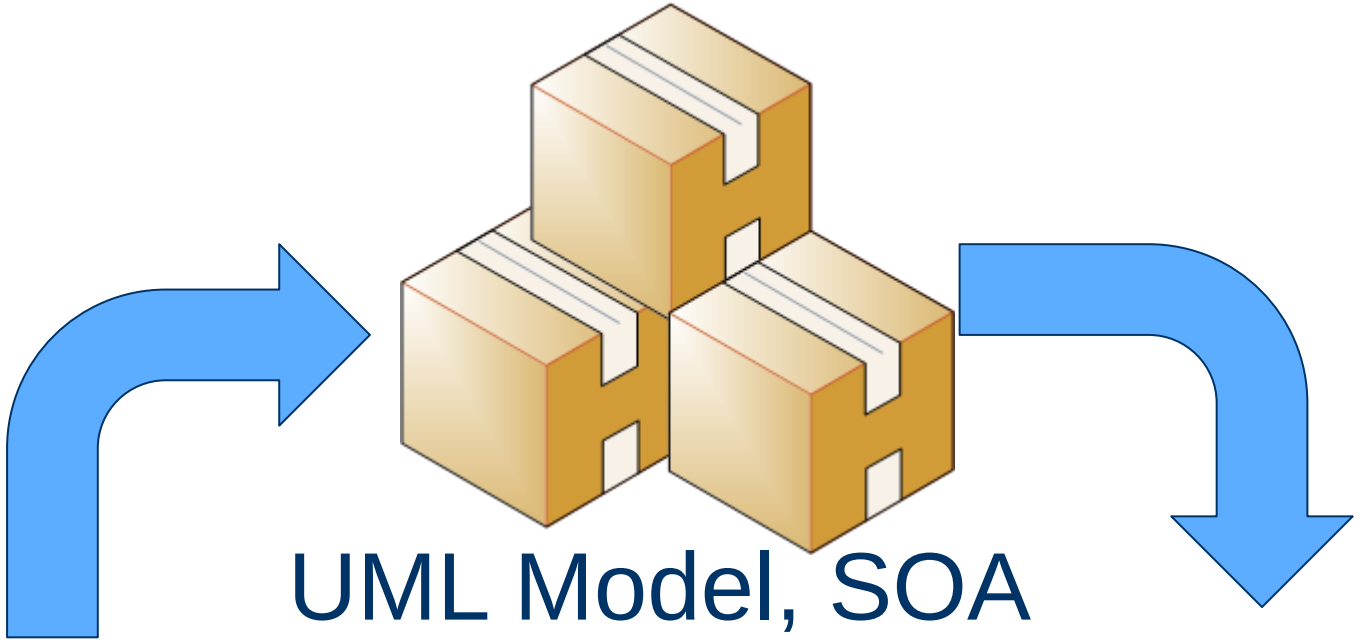
- IDC, “SMBs spent $2.4 billion on cloud computing in 2010”

- Gartner, “SaaS to reach $10.7 billion in 2011”.
- IDC, “SaaS to reach $72.9 billion by 2015”.
- Forester, “CC business will be worth of $241 billion by 2020”.

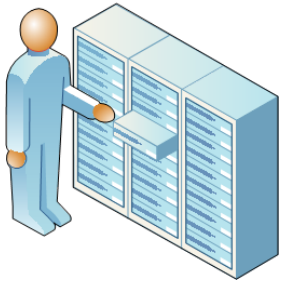
- Apple, AT&T and CA to invest $1 billion each in CC
- CIO.gov, “US government will spend $20 billion on cloud computing by 2015”

3. REMICS PROJECT

REMICS = REuse and Migration of legacy applications to Interoperable Cloud Services



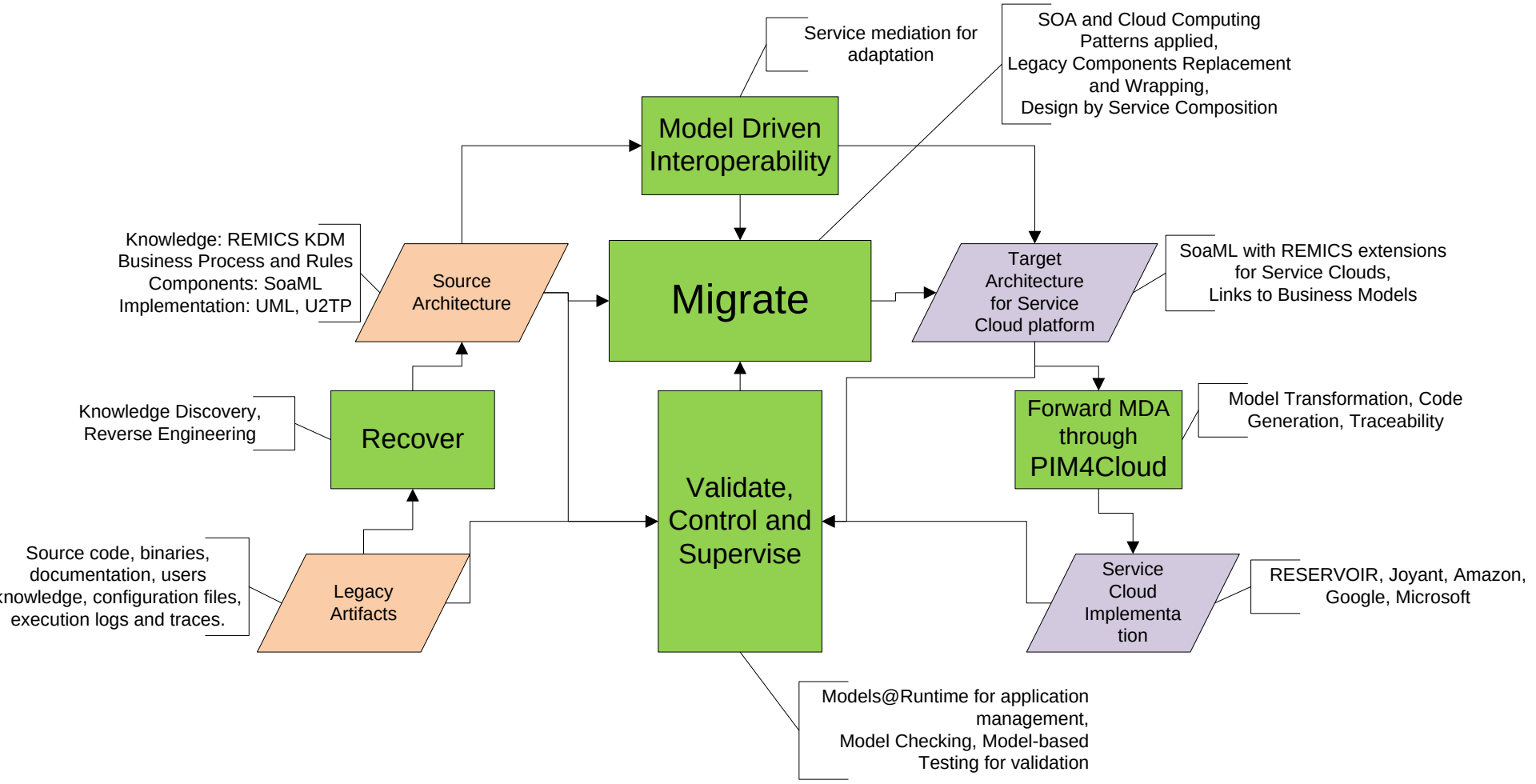
UML Model, SOA



Legacy System:
e.g. Cobol

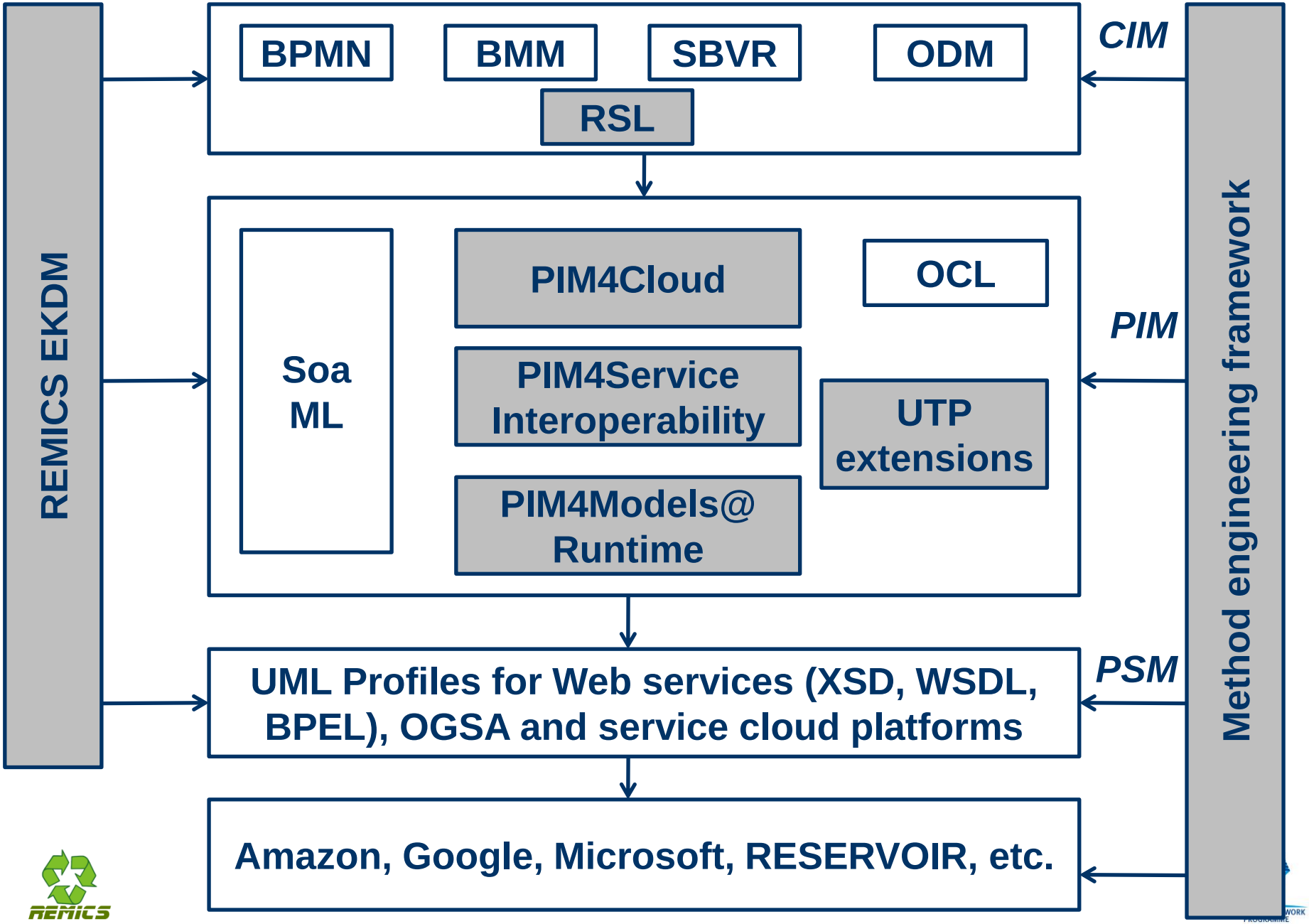
Modern Cloud Application:
e.g. Amazon EC2 based

REMICS Process overview



REMICS facts

- Started in September 2010, Duration – 3 years
- Partners:
  - SINTEF, Norway – Coordinator, MDD, Model-driven interoperability, methodology
  - SOFTEAM, France – MDA, Modelio UML/BPMN tool
  - Netfective, France - MDA, ADM, Models@Runtime, BluAge tool
  - Fraunhofer FOKUS, Germany – Model-based testing, model analysis
  - Tecnalia, Spain – MDA, Softeware methodology engineering
  - DI Systemer, Norway – Case Study, Enterprise Software in Cobol
  - DOME, Spain – Case Study, Travel Agency Software in Delphi and PL/SQL
- Extended partners:
  - Warsaw University, Poland – Requirements Engineering
  - Tartu University, Estonia – Grids, Scientific Calculations
  - Institute ICT, Bulgaria – Agile Methodologies


REMICS EKDM
BPMN
BMM
SBVR
ODM
RSL
CIM
Soa ML
PIM4Cloud
OCL
PIM4Service Interoperability
UTP extensions
PIM4Models@ Runtime
PIM
UML Profiles for Web services (XSD, WSDL, BPEL), OGSA and service cloud platforms
PSM
Amazon, Google, Microsoft, RESERVOIR, etc.
Method engineering framework
deployment platforms

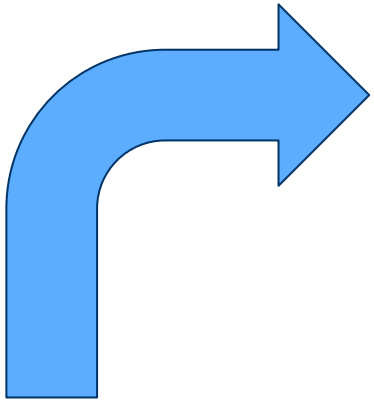
Key achievements in the first year

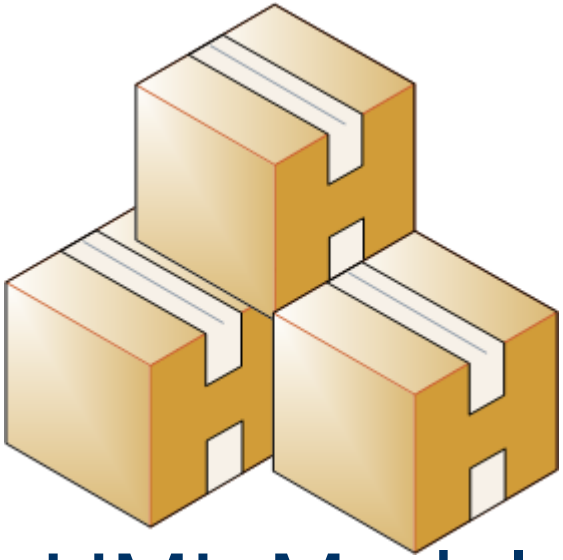
- Methodology
  - State-of-the-art study
  - Preliminary methodology
- Recovery phase
  - Extended Knowledge Discovery
  - Recover of DOME case study – PL/SQL
  - Preliminary recovery of DI case study – COBOL
- Migration phase
  - PIM4Cloud language – preliminary specification , tool implementation
  - Experiments with SOA models extraction
- Testing, Metrication, Interoperability, Models@Runtime
  - Reports, prototypes, demonstration.

Model Recovery

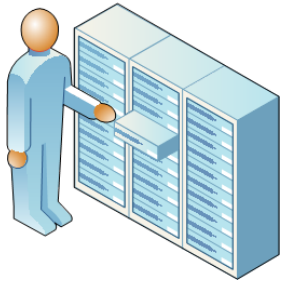
Netfective

Recovery





UML Model



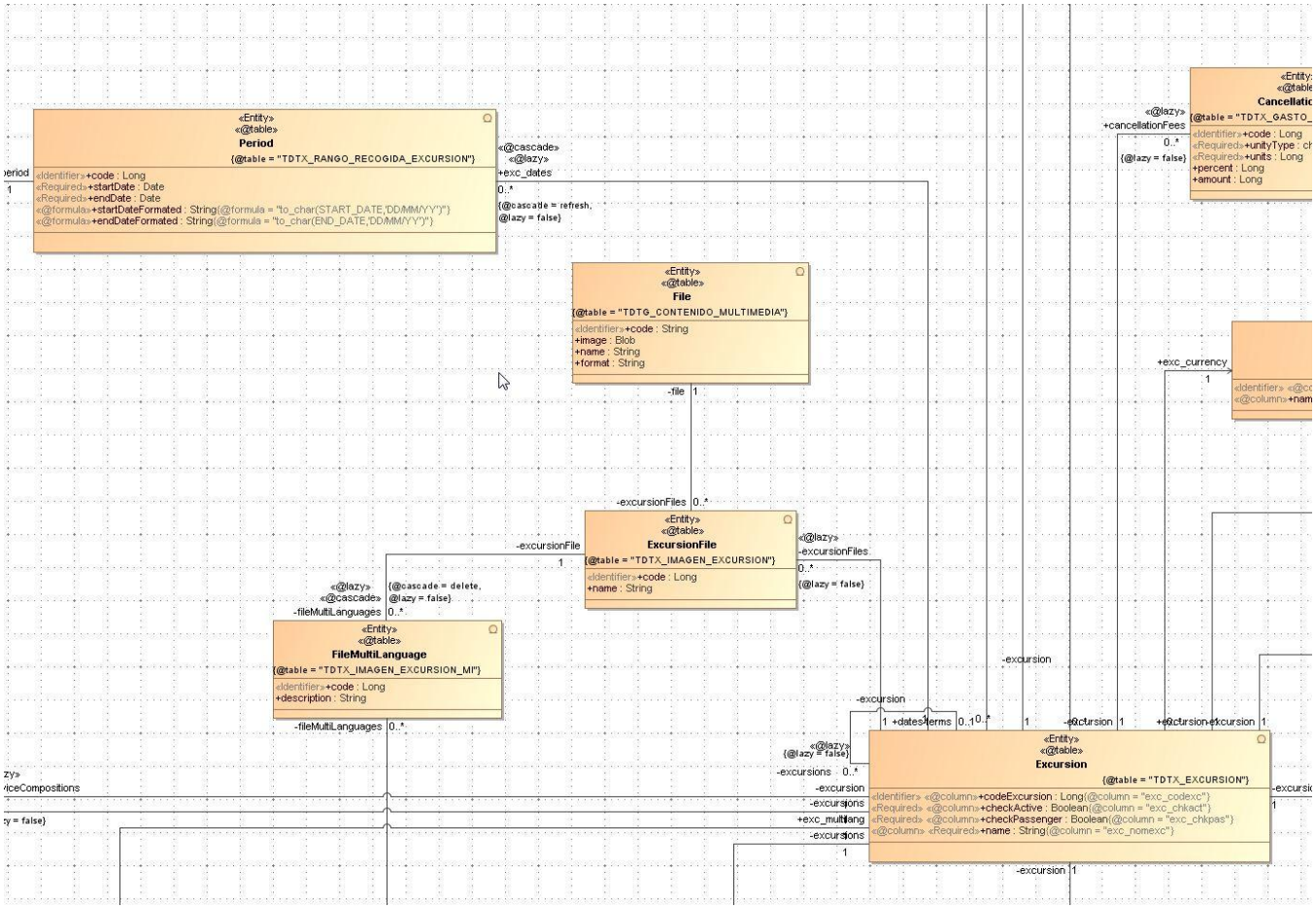
Legacy System:
e.g. Cobol

Modern Cloud Application:
e.g. Amazon EC2 based

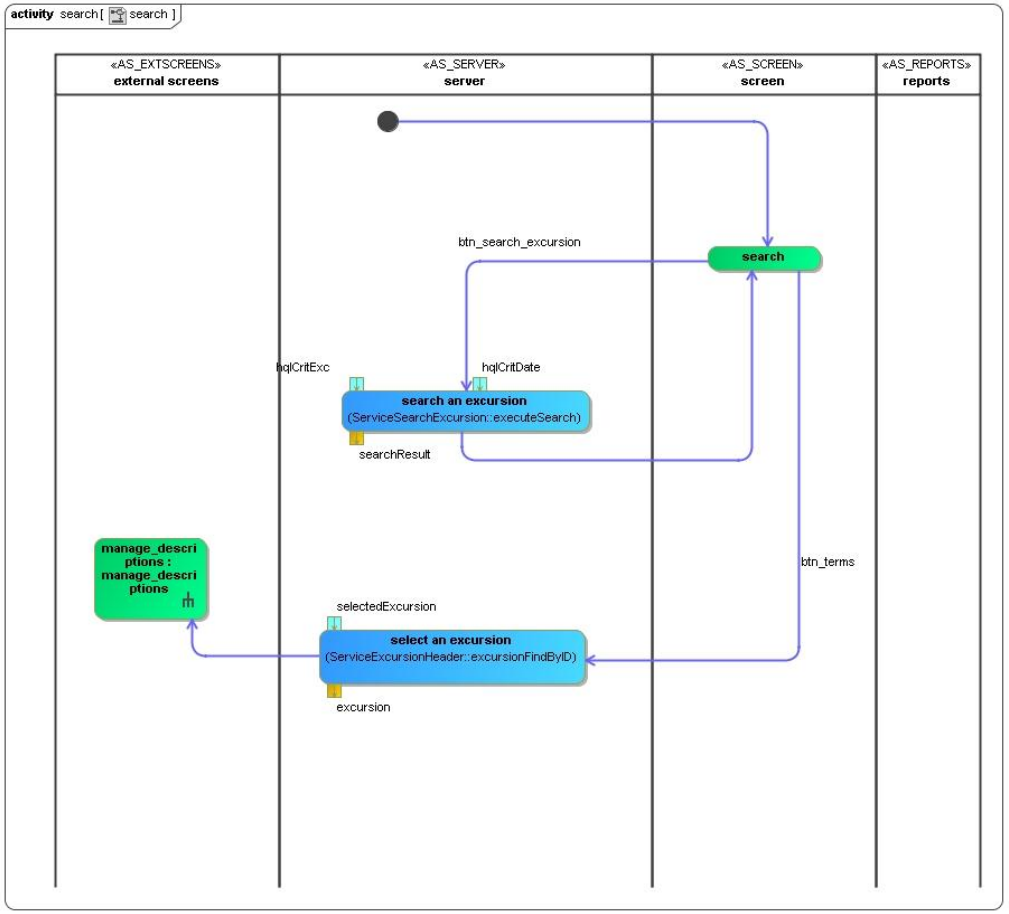
Key achievements

1. Two significant experimentations of the *Recover* technology on DOME and DISYS case studies
2. Ability to straightforward *Migration* through UML
3. Full openness of metamodels and thus business models within the Ecore/EMF worldwide standard
4. Definition of EKDM, a bridge between KDM and ASTM

BLU AGE® UML models: samples (DOME “entities”)



BLU AGE® UML models: samples (DOME “activities”)



Example steps

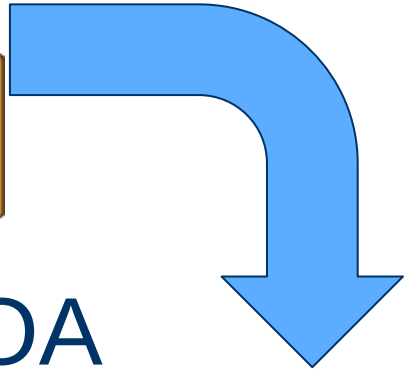
1. Synchronization and navigation between DISYS Cobol code and model-based representation in Ecore/EMF
2. Initial discovery and analysis may be refined by users
3. Annotation mechanism enables to guiding the *Recover* phase along with capitalization on transformation rules
4. UML Platform-Independent Models are progressively accessible for the *Migration* phase

PS: this demo is based on the DISYS case study that is under treatment at Netfective (data definitions were not available in source code at demo. design time)

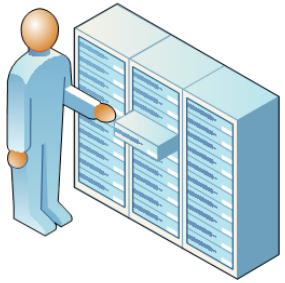
Migration

Softeam

Migration and PIM4Cloud



UML Model, SOA



Legacy System:
e.g. Cobol

Modern Cloud Application:
e.g. Amazon EC2 based

Key achievements

1. Netfective experiments for forward MDA
   - DOME application generation
2. BluAge to SoaML
   - Proof-of-concept transformation
   - Basis for further research on components recovery
3. PIM4Cloud UML Profile
   - Modeling cloud application deployment
   - Linking between SOA and Cloud
   - Basis for further experiments, research and standardization

Migrate

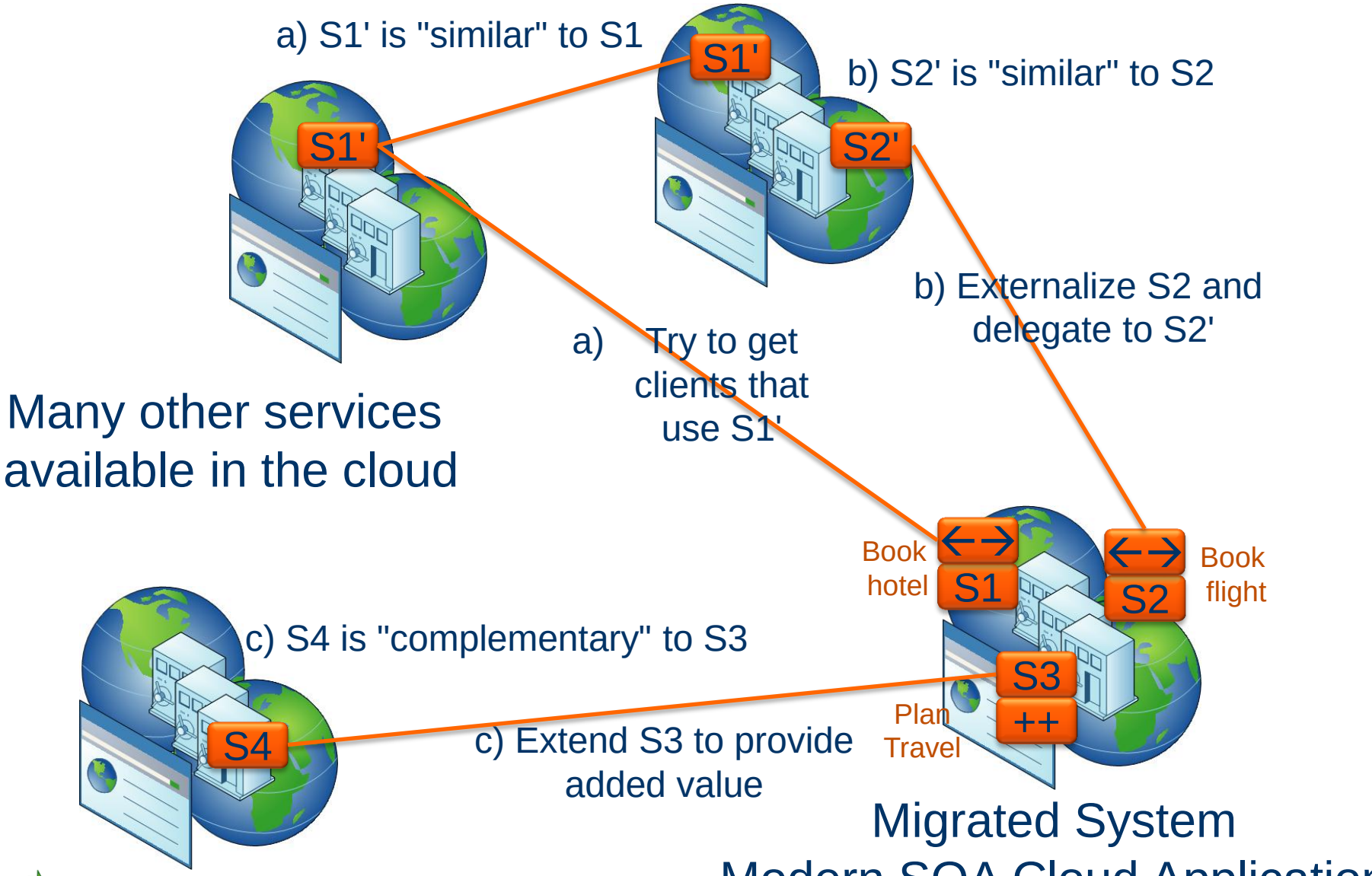
- Forward engineering for adding new functionality;
- Deployment in Cloud:
  - **PIM4Cloud** planned as an abstraction of cloud computing platforms and a language for modeling deployment in cloud.
  - Initial focus is on IaaS aspects, but we are also interested in PaaS.
- Model transformation from these models to cloud platforms.
- Standardising **RSL** within the OMG as a language for semantically precise requirements.

What is SoaML?

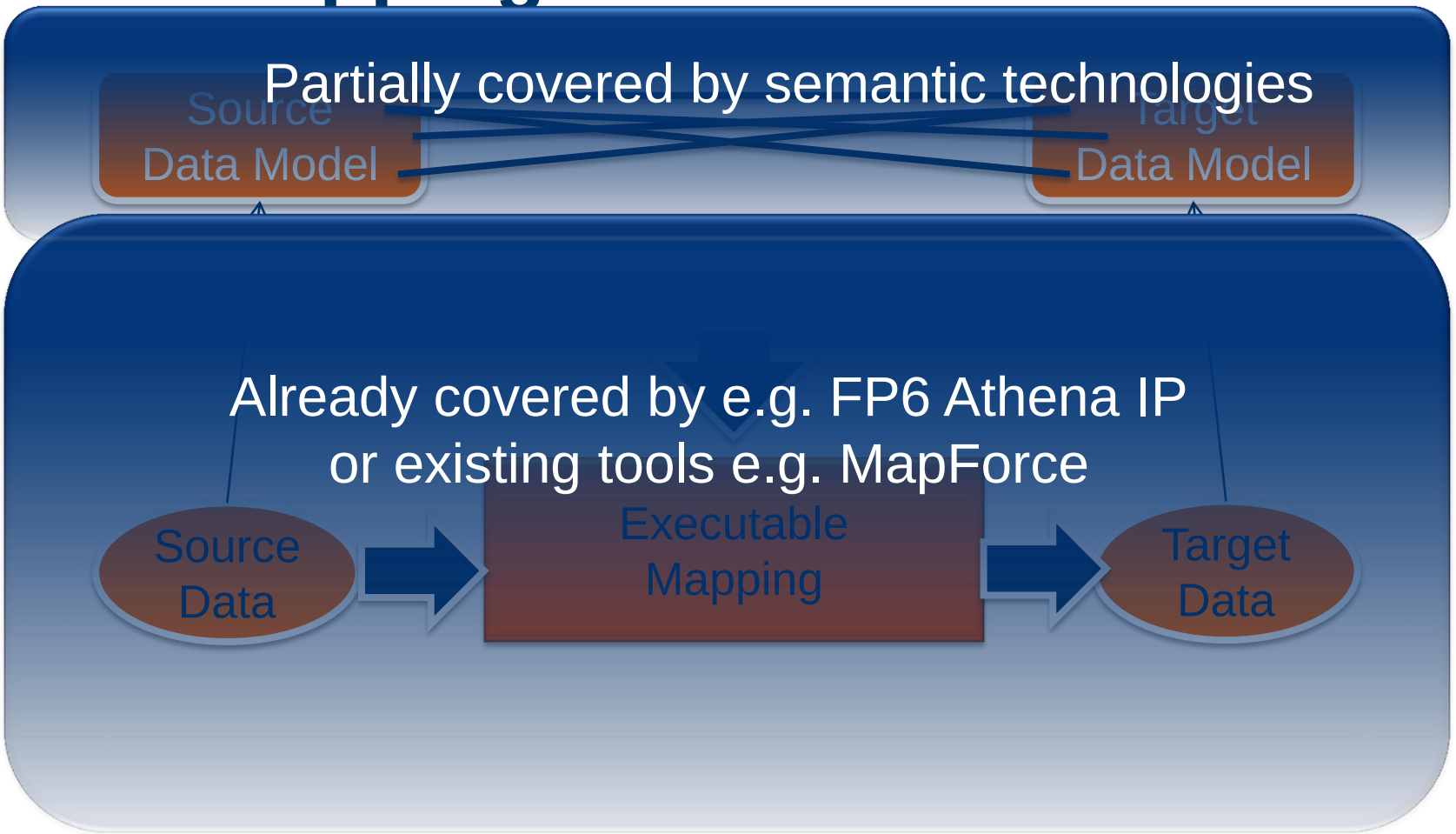
- Service oriented architecture Modeling Language (SoaML)
- Defines language constructs and extensions to UML2 to support service concepts (metamodel and UML profile)
- Focuses on basic service modelling concepts and structure.
- A foundation for further extensions and integration with BPMN, BMM and other metamodels.

- Key language constructs
  - Consumer
  - MessageType
  - Participant
  - Provider
  - ServiceContract
  - ServiceInterface
  - ServicesArchitecture

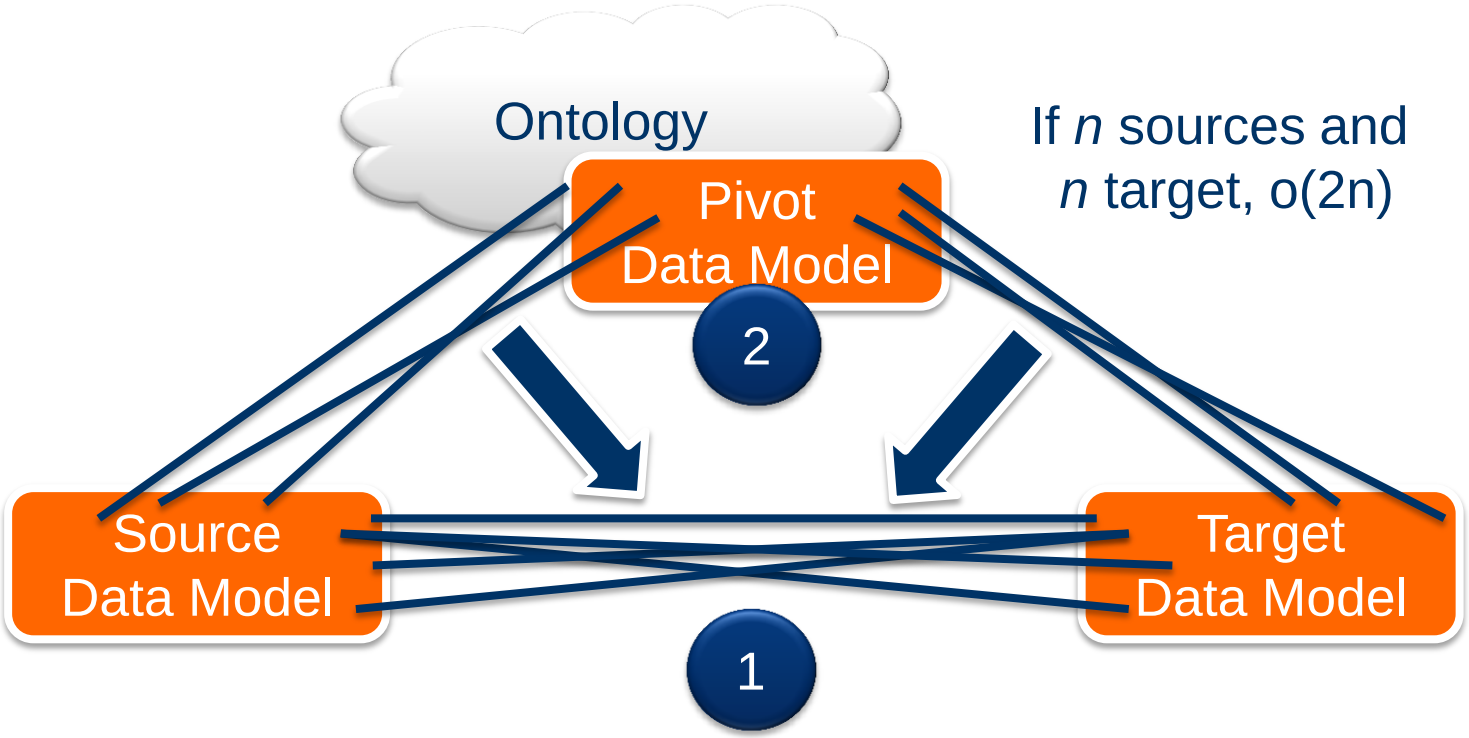
Interoperability in REMICS



Data Mapping



Data Mapping: Still intensively manual



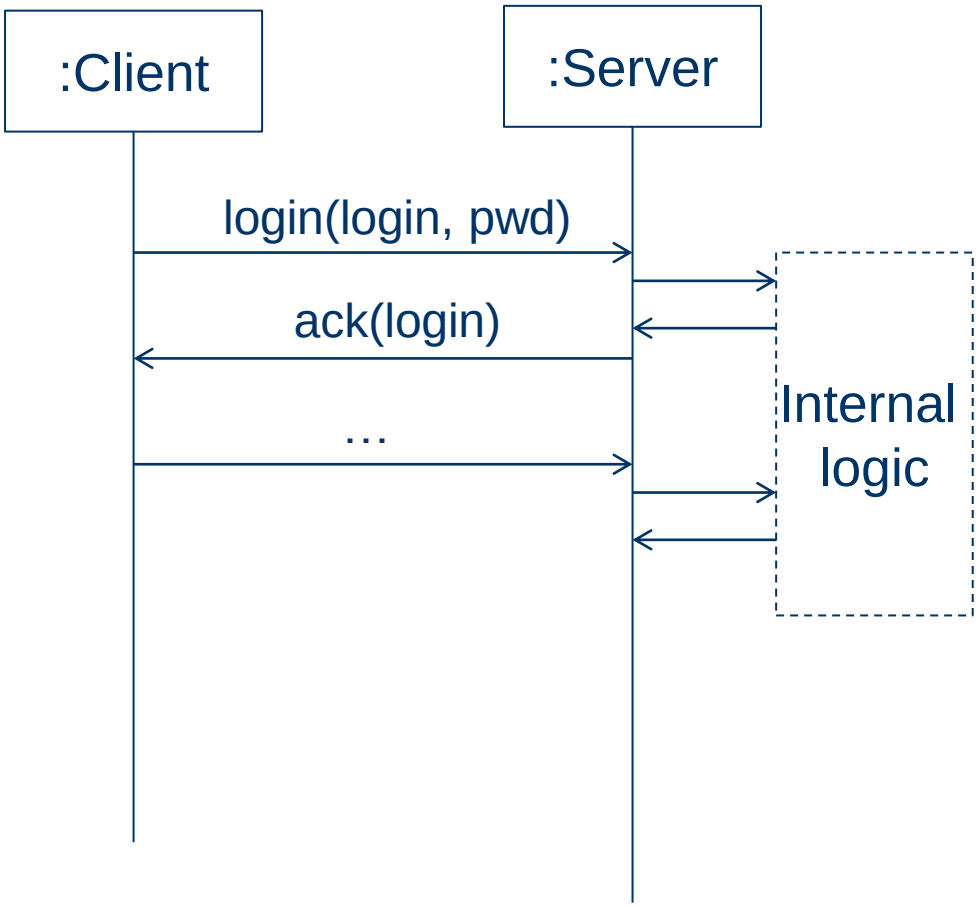
If *n* sources and
n target, o(2n)

If *n* sources and *n* target, o(n^2)

Data Mapping: REMICS contribution

- Provide mapping recommendations to designers
  - Go beyond syntactical matching, even when no ontology exists
  - Leverage/learn from user knowledge (basic learning)
  - Reasonably good accuracy/performance
- Scope of the tool
  - Compute mapping recommendations between data models, which
    - Can be abstracted as class diagrams
    - Are expressed in "standard" english
  - Simple yet usable GUI: visualization issues are out of the scope

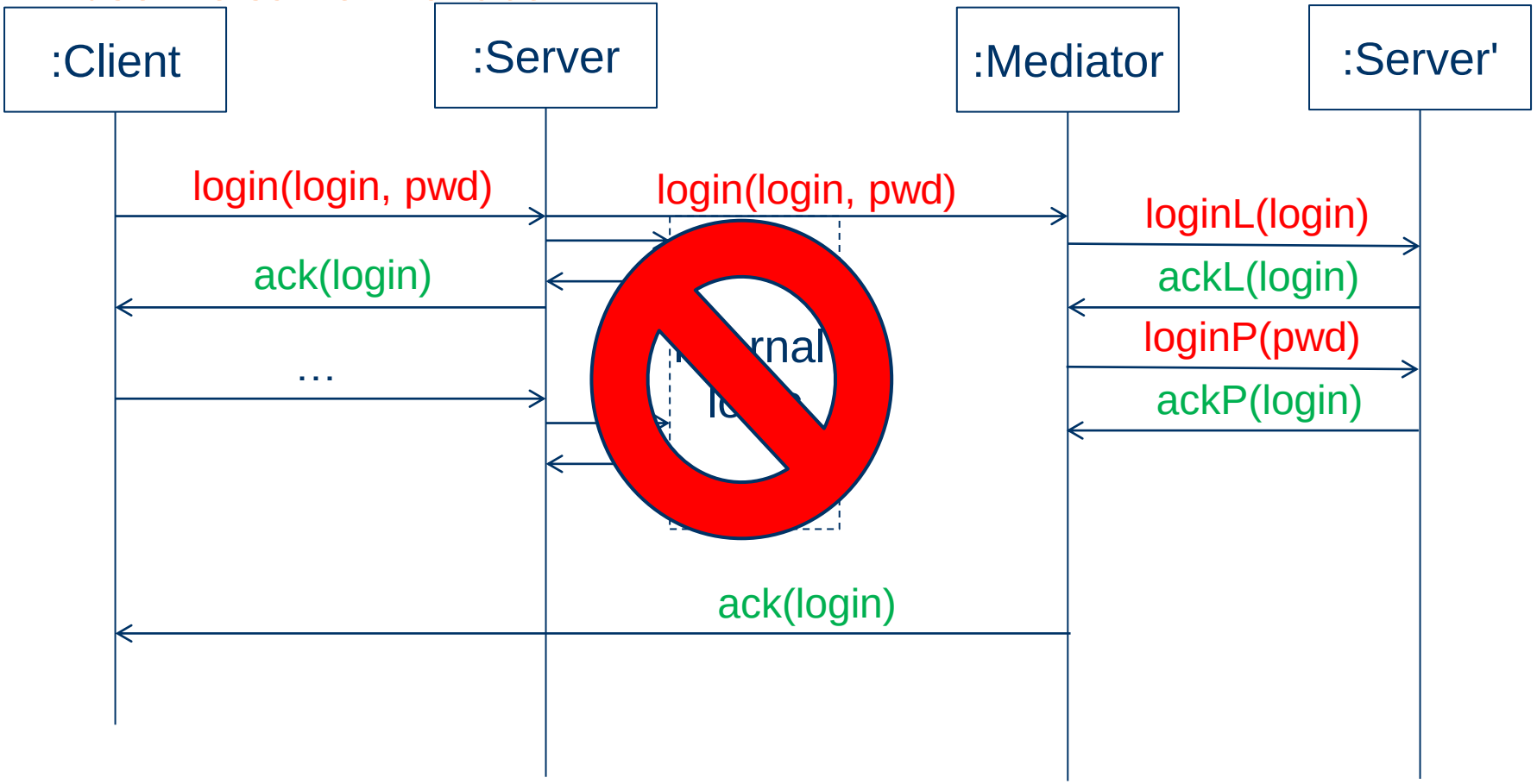
Service Mediation: Migrated service



Service Mediation: Externalization

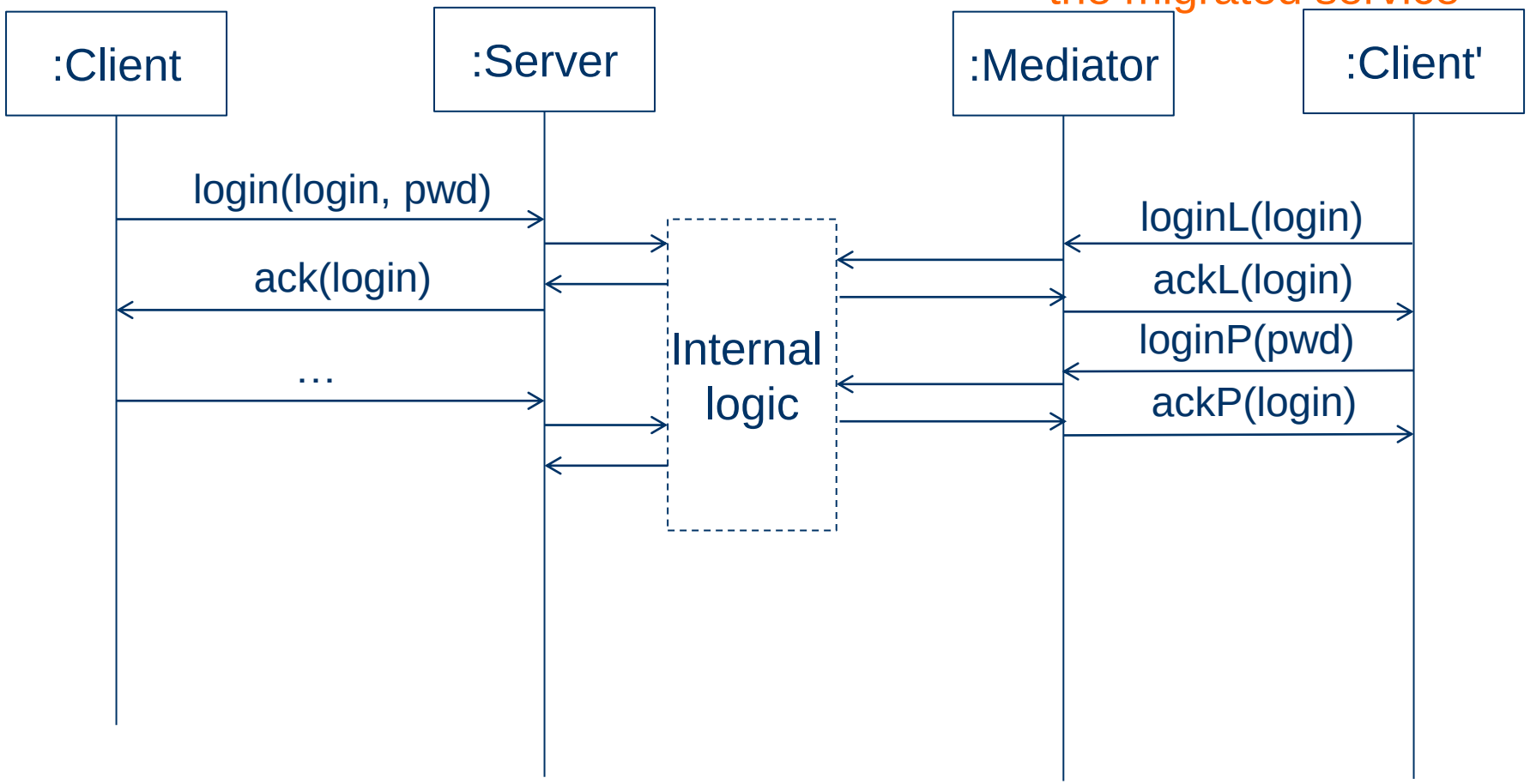
Former clients can still use the same interface…

…but the actual logic is delegated to an external service



Service Mediation: Market penetration

New clients seamlessly use the migrated service



Service Mediation: On-going work

- Generation of "rich" service mock-ups
  - Makes it possible to test mediators in a controlled environment (no charge for pay-per-use service, etc)
- Design of mediators as state machines
- Code generation to make mediators operational (in a controlled environment)
- WP6 is performing some experiments to deploy mediators on models@runtime engine
- Future work
  - Generate mediators (at least rich skeleton)
  - Code generation to WP6 models@runtime engine

6. CONCLUSIONS

Software engineering challenges during migration

- Feasibility stage
  - Set of questions to answer
  - Identifying SOA and cloud benefits for the context
- Modernizing the architecture
  - Separating business logic from APIs;
  - Separating business logic from data;
  - Changing the synchronous behavior of legacy systems to the asynchronous behavior of services;
  - Componentization of architecture to improve scalability
- Defining quality characteristics in the cloud: max load, accessibility, etc.

Conclusions

- Migration is an important topic in the modern life
- MDA and ADM are OMG concepts for modernization with models
- REMICS = REuse and Migration of legacy applications to Interoperable Cloud Services
- Work in progress: Recovery of COBOL code, PIM4Cloud modeling language

- More results soon at http://www.remics.eu