Reuse and Migration of Legacy Systems to Interoperable Cloud Services- The REMICS project

Contacts:
Alexis Henry (Netfective/Blu Age)
Parastoo Mohagheghi (SINTEF)

BLU AGE AGILE MODEL TRANSFORMATION

NETFECTIVE Creative Technology

Project facts

- REMICS is a research project (STREP) accepted in the Objective 1.2 of FP7 Call 5 (Internet of Services, Software and virtualization).
- Aims at migration of legacy systems into Cloud services based on service-oriented architectures
- REMICS runs from September 2010 for 3 years.
- The budget is 4.5 Meuro
- Partners are:
  - SINTEF (Coordinator) **(Norway)**,
  - Softeam, NetfectiveTechnology **(France)**
  - DI Systemer **(Norway)**
  - Fraunhofer **(Germany)**
  - ESI, DOME Consulting and Solutions **(Spain)**

BLU AGE AGILE MODEL TRANSFORMATION

NETFECTIVE Creative Technology

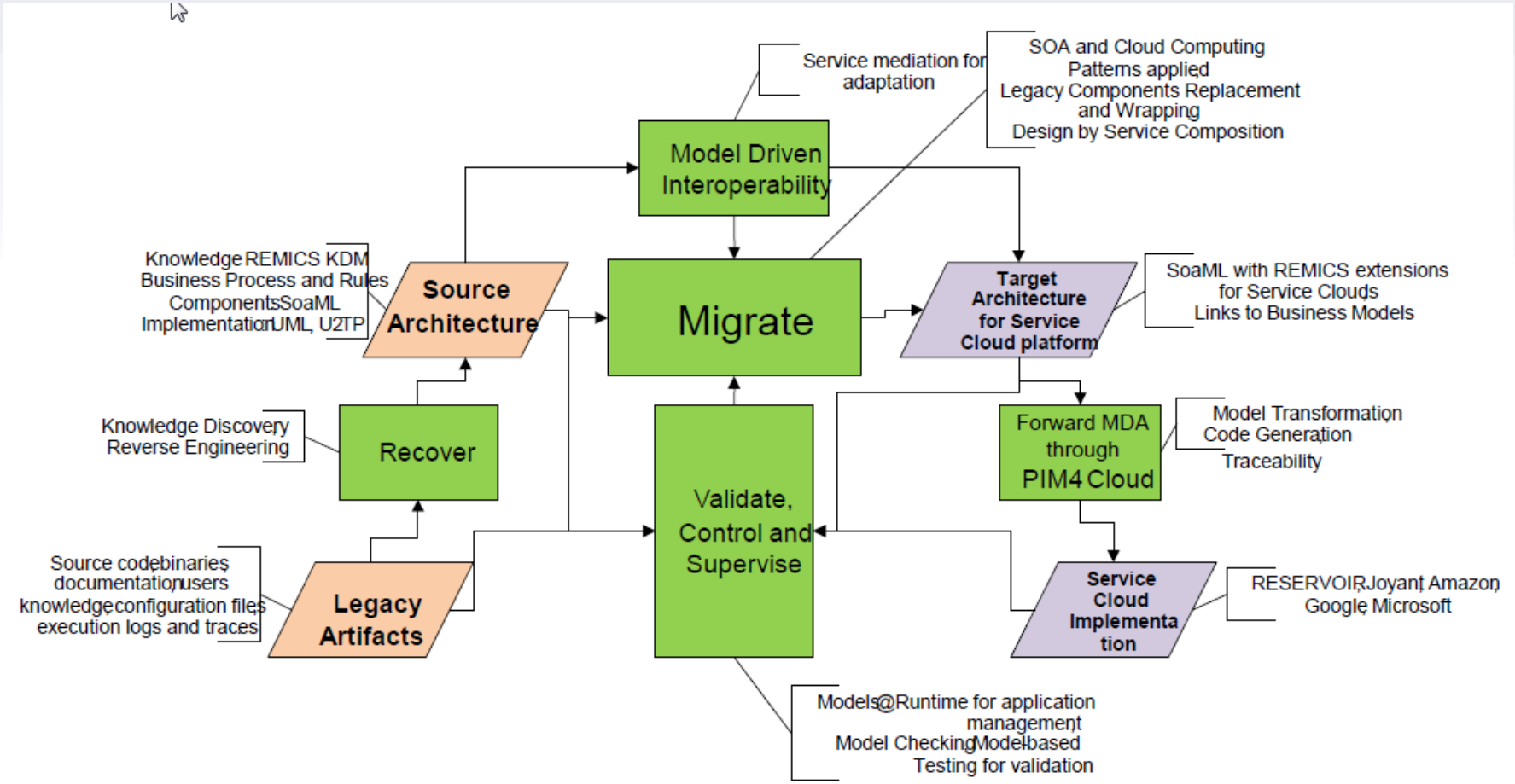
Problem to be adressed

- Legacy systems are sometimes (most of the times) of substantial value for companies:
  - They still function for the users' needs;
  - They capture important business logic;
- However:
  - Legacy systems are often difficult to reuse due to platform, documentation and architecture obsolescence.
  - *Legacy systems are facing critical issues:*
    - *Need for change with no capability to do so ($; retirement of resources, compliance, technical obsolescence)*
  - The risk of manual rewriting or replacement by ERP is high,
  - The cost of replacing them with systems designed from scratch is often too high.
  - New technologies arise such as Cloud Computing and Software as a Service that promise better performance or cost saving that motivate organizations to change there is and possibly modernize their applications:
    - Cloud is a solution to new applications !
    - What about my legacy application ?

Challenges

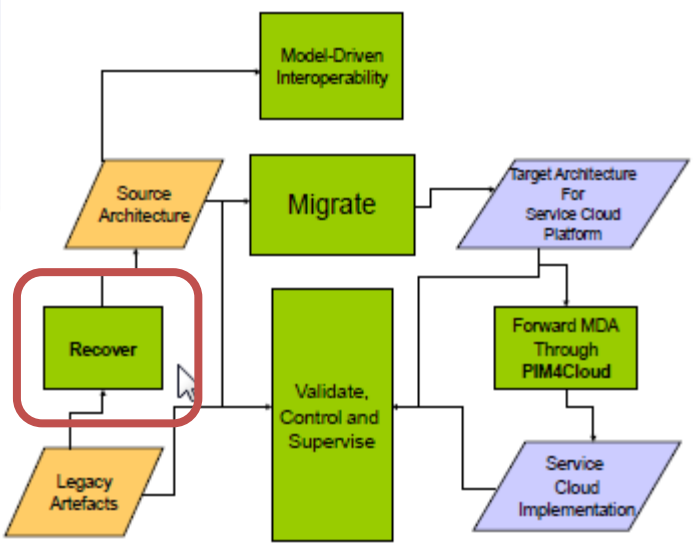
- The oldness degree of technologies to be reversed:
  - How to adapt them to the SaaS (SaaS, IaaS, PaaS) and cloud paradigms?
  - How to handle interoperability?
  - How to operate (control, supervise, bug identification and correction, ease of deployment)?
    - SaaS: no capability to modify the system !
    - Iaas, PaaS: what/how/at what speed can you deploy fix/evolution packages and control components ?
- The absence of knowledge:
  - How to extract business value information?
- QoS must be preserved:
  - Performance of heavy loaded and critical applications
    - X 000 users
    - X 000 screens, XX 000 batch
    - XX To of data
  - How to reuse legacy systems in automated testing of the new SaaS?
  - Size of systems : 1 to 30 Million LOC
- Cost of the migration process:
  - How to plan a progressive migration process?
    - Legacy system may have been built in 10 to 25 years
  - How to train people in new technologies
    - Change management for application users
    - Change management for developpers (MDE and cloud technologies)

Steps in Remics approach



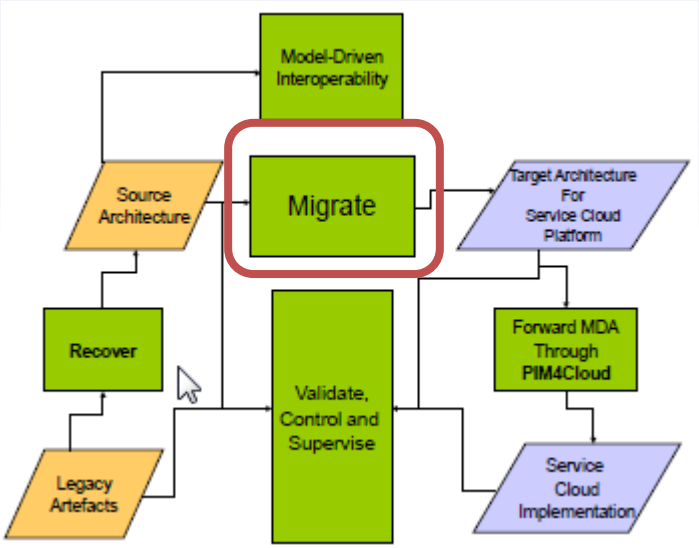
Recover

- Analyse feasibility of the modernization strategies and select one or multiple:
  - Automatic extraction
  - Computer assisted extraction
  - Annotation driven extaction
  - Refactoring at the PIM level
  - Paradigm change:
    - Cobol reports = batch and printer; OO report: BI tools for instance or RIA UI components
    - Usually code is migrated to XML type of data vs algorithmic
- We plan to use the OMG KDM standard and extend it when necessary.
  - Ex: "level" attributes for segment and rubric in Cobol
- Recover business value information:
  - requirements, processes, rules, non-functional properties etc.
  - Separation of concern
    - Business code vs technical code
    - UI/service/Batch/Report/data
- Use automated reverse engineering methods as much as possible;
- Develop models (business, components, test specifications etc.) that will be used further.



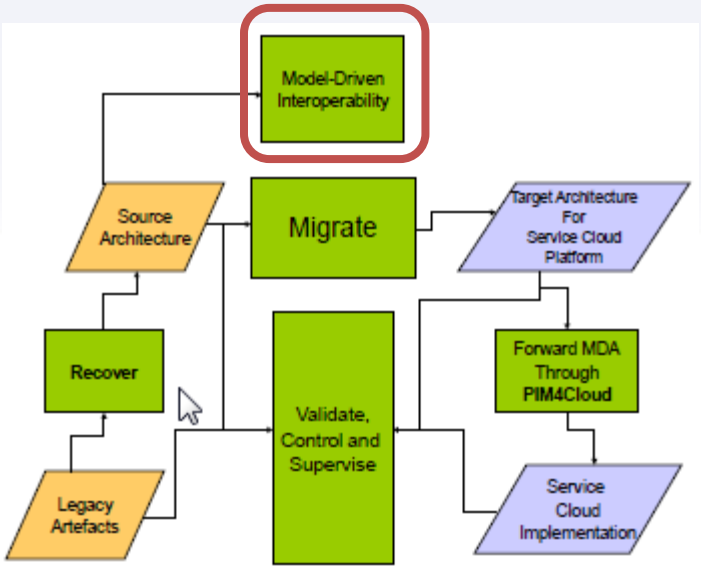
Migrate

- The purpose is to start from the legacy models and refactor them to build a new SOA by applying methods such as decomposition, component wrapping and replacements.
- Some components or services may be replaced by newly discovered ones.



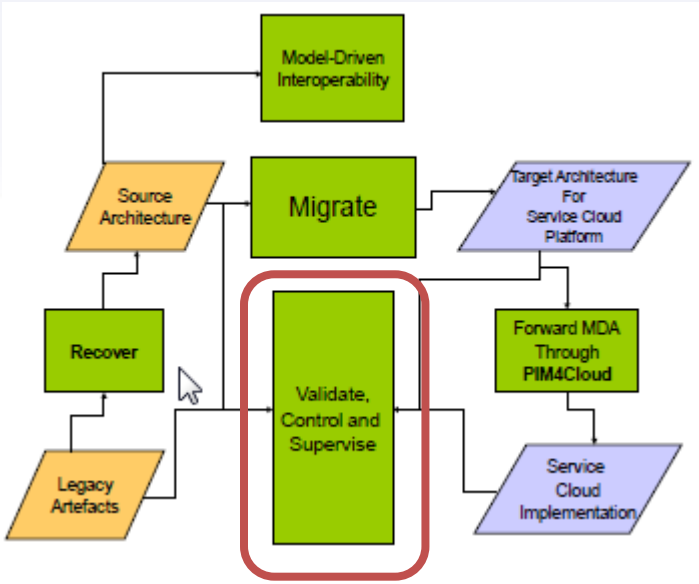
Compose and develop new services

- The legacy system may be enhanced by adding new services or services may be composed in new ways.
- Model-driven interoperability helps in adapting services using mediators. (Ref. Paper on Flora-2 interoperability mappings at MDI)
- Mediators or mediation services take input data in one format and provide it in another format.
- We plan to extend SoaML with data format models and behavioral model for mediation.



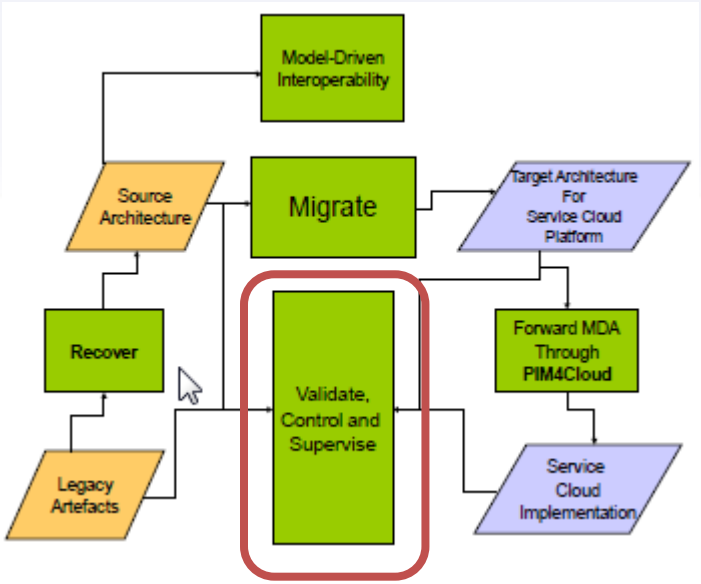
Validate

- The recovered architecture should
  - correspond to the legacy system
- and must provide the same or better
  - QoS, business goals, coverage, etc.
- Recovered models should be used in the validation process based on model-based testing techniques.
- The original system can act as a test oracle since requirements may not be well captured.



Control and supervise

- The goal is managing applications by observing them and performing corrective actions.
  - Legacy cobol are critical application with nearly 0 defects and optimized performance: ie a 25 years old application example
    - 10 years development/evolution
    - 5 years tuning and optimization
    - 10 years without any change
    - Resouces quit the system
    - Mainframe operation excellence
    - Cost for defects is extremy high (Millions USD)
- Models@runtime for self-manageability is one possible technique to use.
  - However: what is feasible within Cloud Architecture ? Does Cloud provider has APIs avalable to do so?
- OO IDE have extra capabilities vs legacy
  - Branch coverage at runtime
  - Code refactoring
  - UML impact analysis
- Cloud: elastic access to cpu and memory



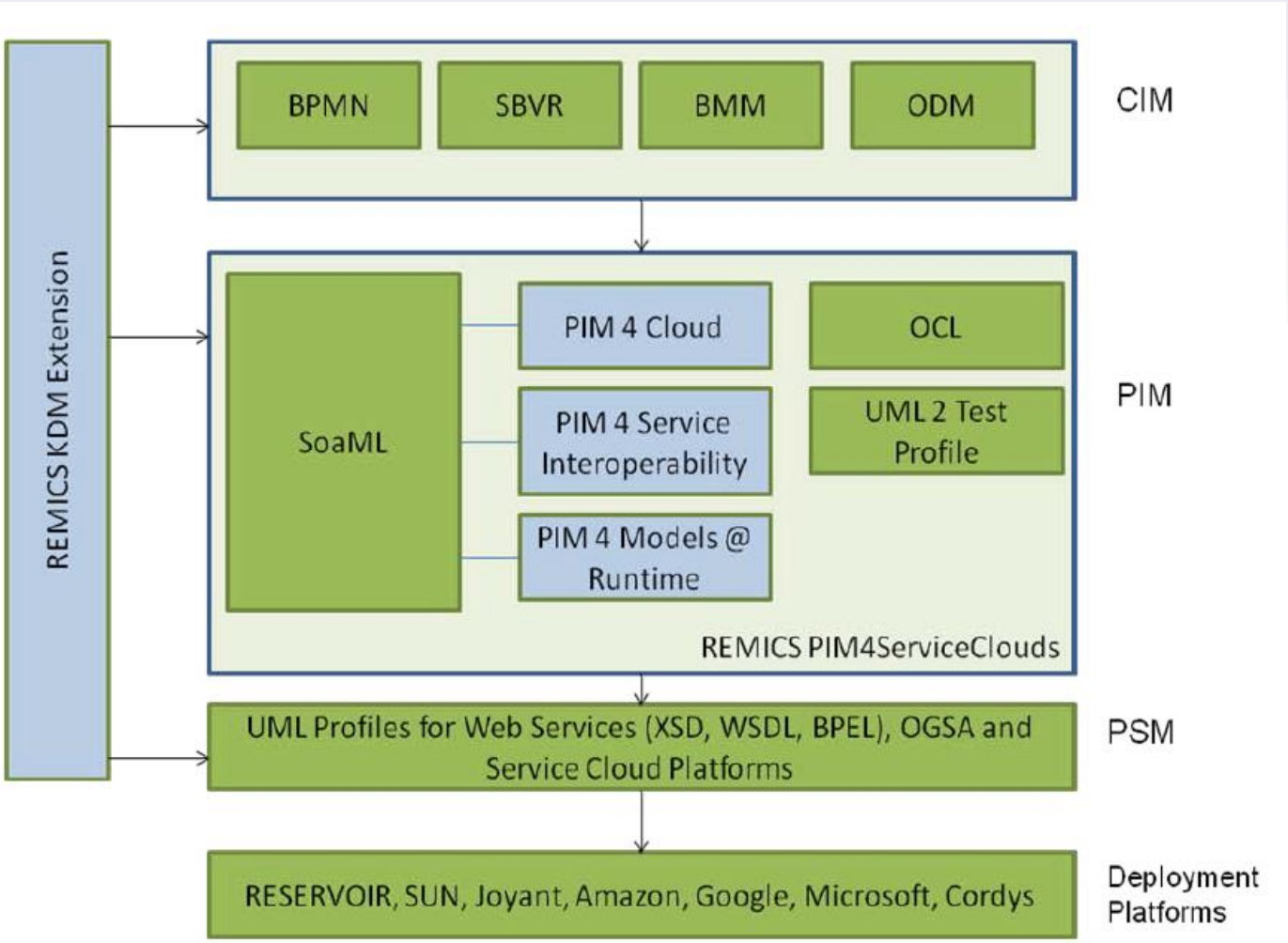
Technological approach: main points

- Model-driven techniques & Models everywhere:
  - A large set of metamodels and several dedicated extensions.
  - In particular, the PIM4 Cloud Computing, model-driven Service Interoperability, KDM extensions and Models@Runtime extensions are intended to support the REMICS methodology for service cloud architecture modelling.
- Open source Metamodels and Models with an emphasis on Open Models for standards.
- Two pilot cases:
  - DI systems from Norway with ERP/accounting
  - DOME consulting from Spain within the tourism section

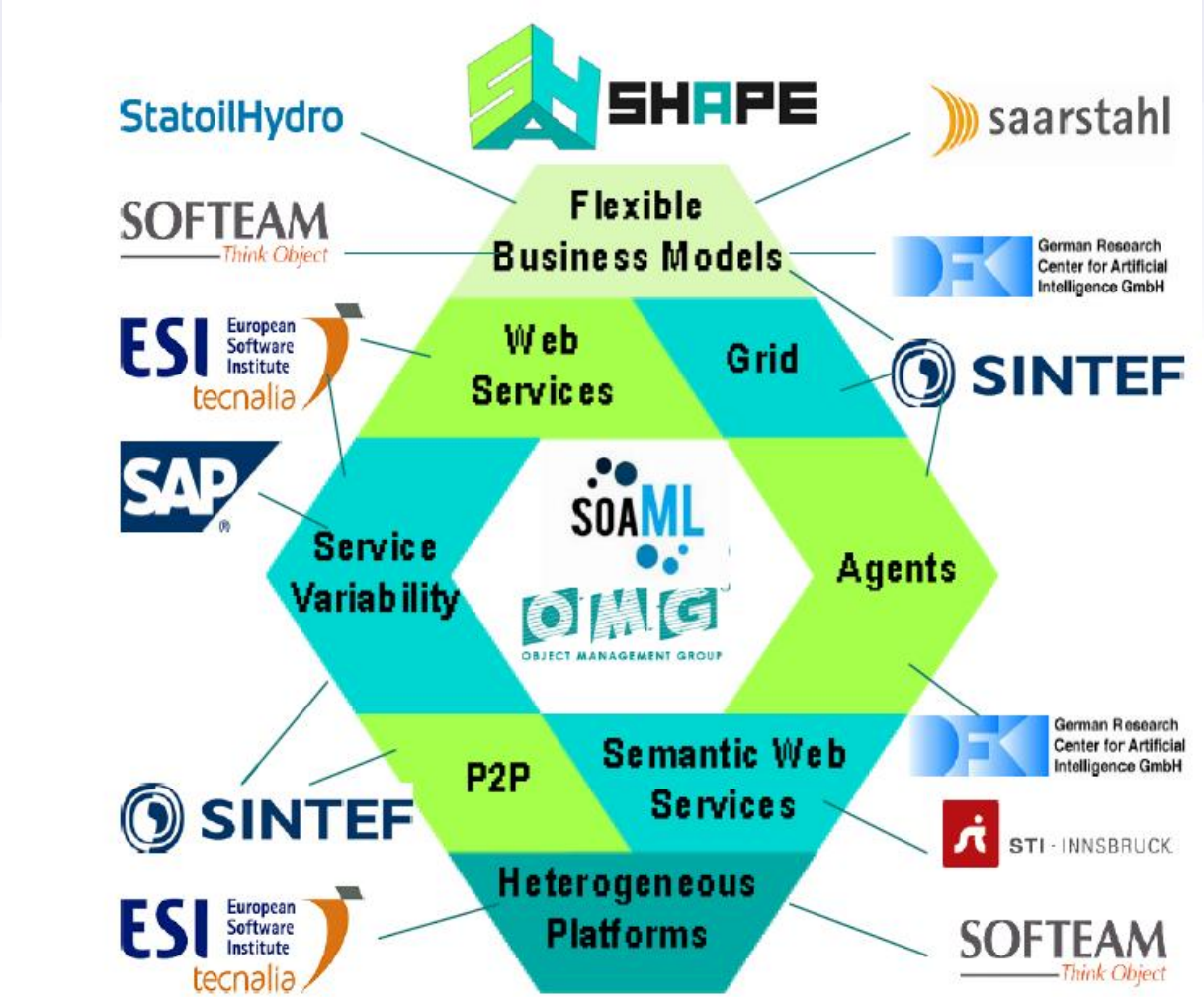
Expected impacts

- REMICS will preserve and capitalize on the business value engraved in legacy systems to gain:
  - Scalibility, flexibility brought by Service Clouds,
  - lower the cost of service provision,
  - shorten the time-to-market.
- REMICS research will provide innovations in advanced model driven methodologies, methods and tools in Software as a Service engineering.
- REMICS will provide standards-based foundation service engineering and will provide a suite of open ready-to-use metamodels that lowers barriers for service providers.

Remics and standards



Extending SOAML



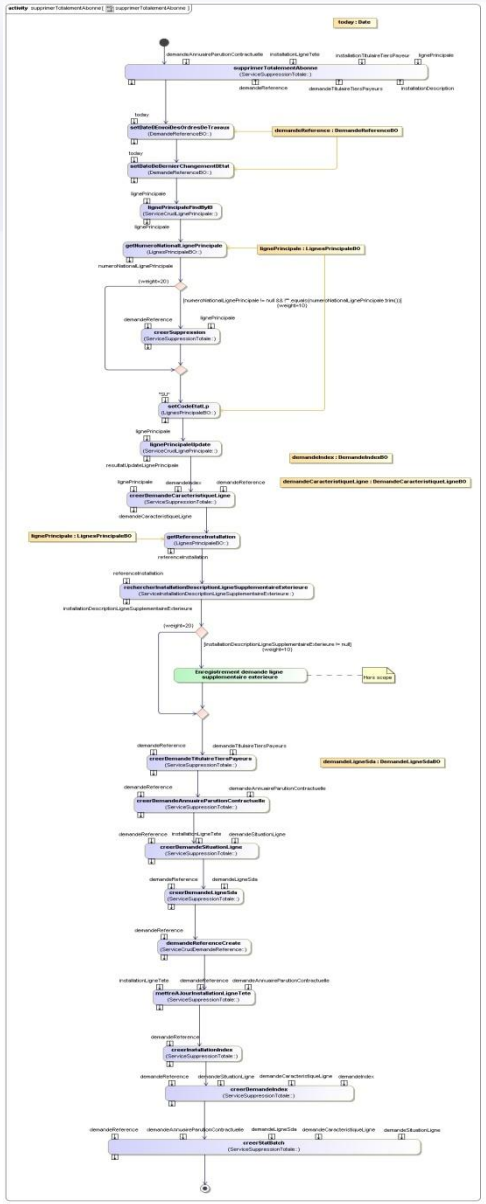
Demonstration

- Netfective/Blu Age example:
  - Cobol Example:

  - AGL Example:

Demonstration

- Netfective/Blu Age example:
  - Sample UML/SOA Organization:



Demonstration

- Netfective/Blu Age example:
  - Modernized code quality: Cobol To Java EE migration , first iteration (3 weeks prototype) of a 37 years old COBOL application (3 millions line of code)