

ROMULUS: Domain Driven Design and Mashup Oriented Development based on Open Source Java Metaframework for Pragmatic, Reliable and Secure Web Development



Boni García
Universidad Politécnica de Madrid

CSMR - 18th March, 2010



Table of Contents

















- Introduction
- Why Romulus
- Objectives
- Results
 - Development
 - Run-time
- Conclusions

Introduction



•7FP ICT-2007.1.2. Consortium:

	GESFOR	ES 	Coordinator, J2EE, OWASP
	ASSET	IT 	SME, Metaframework Product
	LIFERAY	DE 	SME, Portal Product
	UPM	ES 	R+D Soft. Engineering
	IMOLA	IT 	SME, J2EE, JBI
	DERI	IE 	R+D Semantic web
	ICI	RO 	R+D Soft. Engineering

Why Romulus



PROBLEM

Java web development is the leading trend in Europe with more than 1M qualified developers

- Java Enterprise Edition provides more than 900 **industry driven specifications**
- Robust, scalable and proved technology BUT...
- There are problems in its **productivity**
- Need of new methods such as Ruby on Rails

Not as productive
as desirable.
Solution:

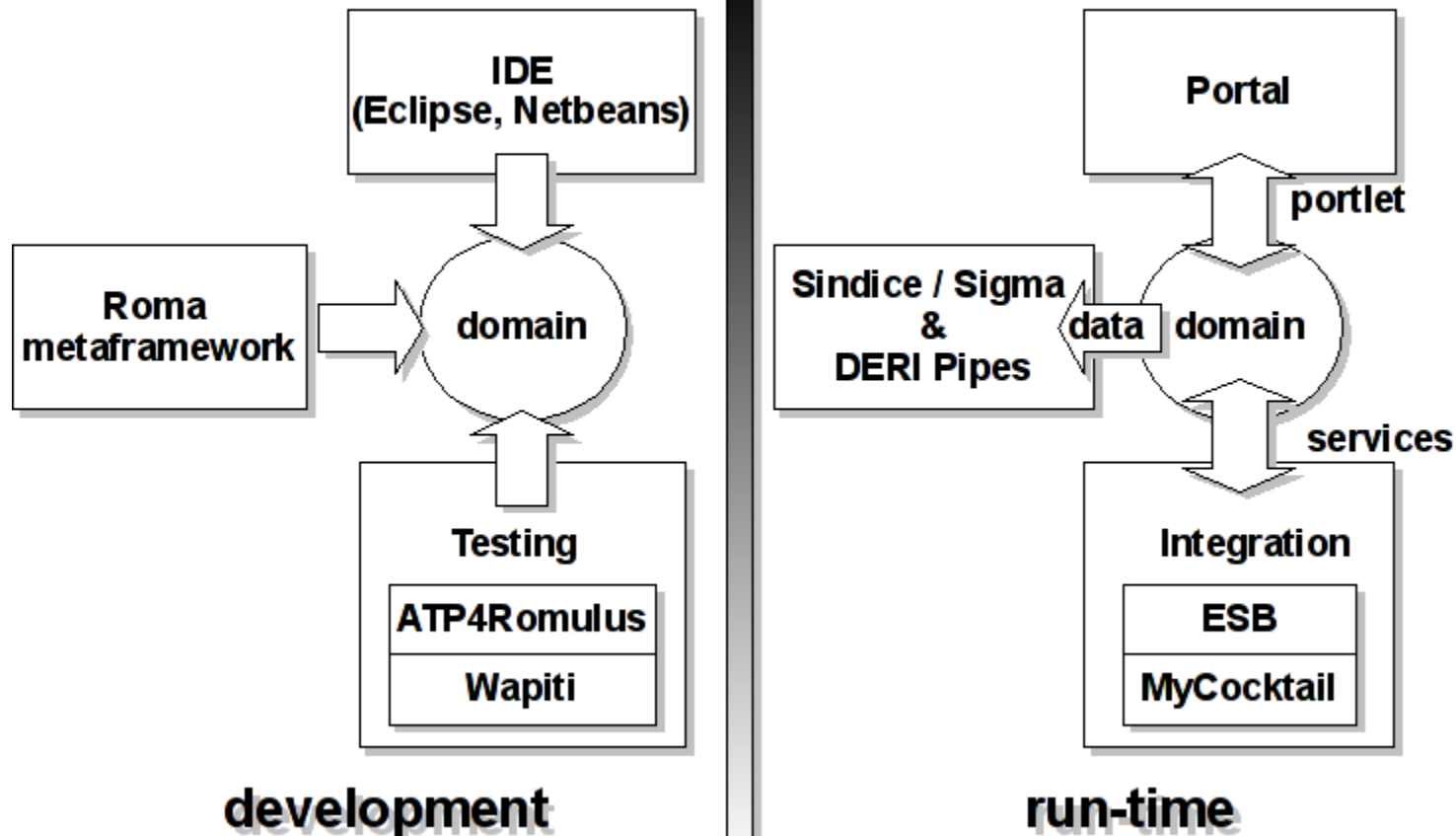


Romulus Objectives

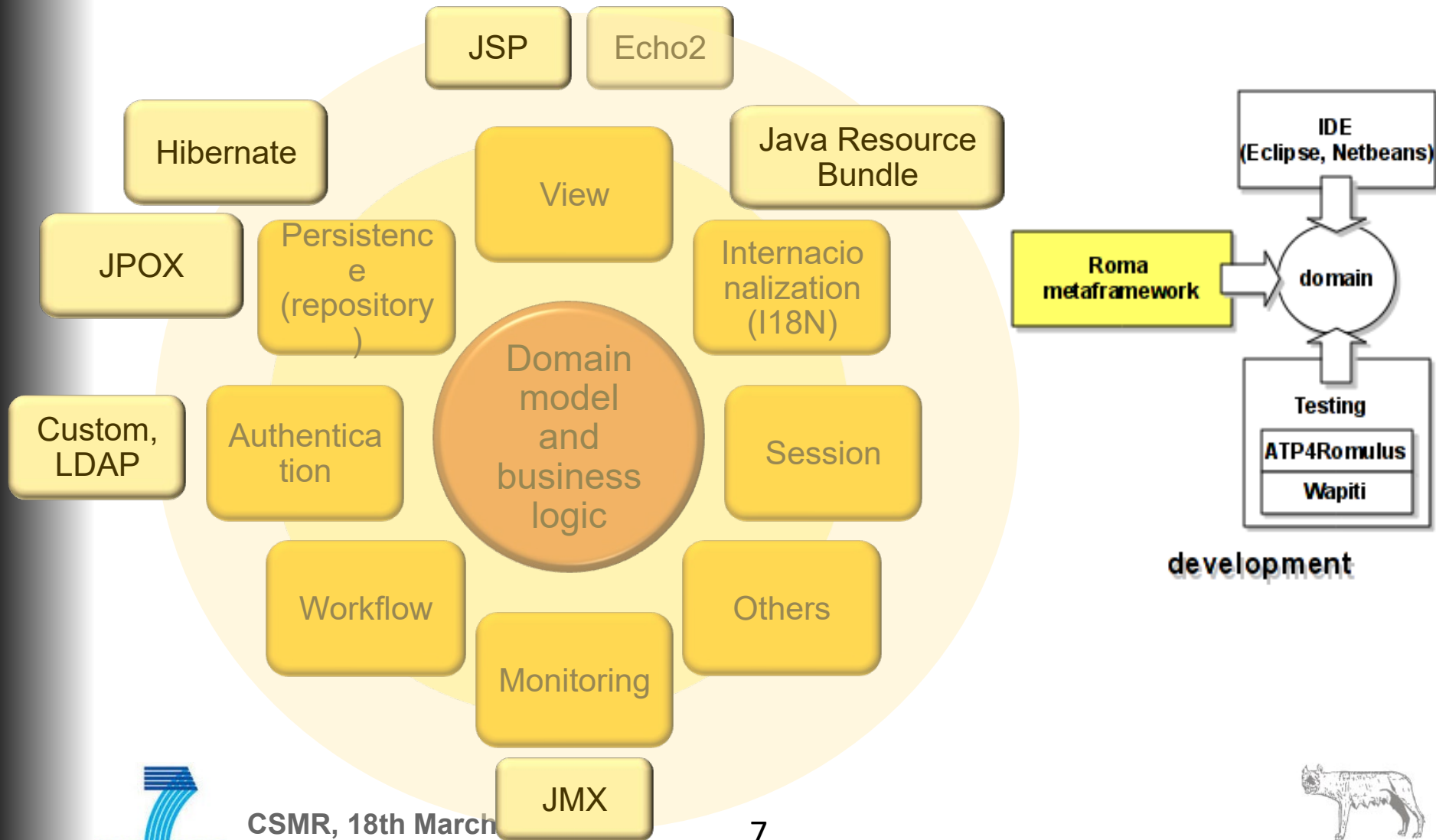


- Researching on novel methods for increasing **productivity** and **reliability** of **Java-based web development**
 - Provide methods and tools for **Domain Driven Design** based on a **Metaframework**
 - Extend domain description with **soft goals** and provide automatic generation of **functional**, **load** and **security tests**
 - Integrate **mashups** in the development cycle
 - Provide bias for seamless **balance** of server and client scripting technologies

Results: The Romulus Approach



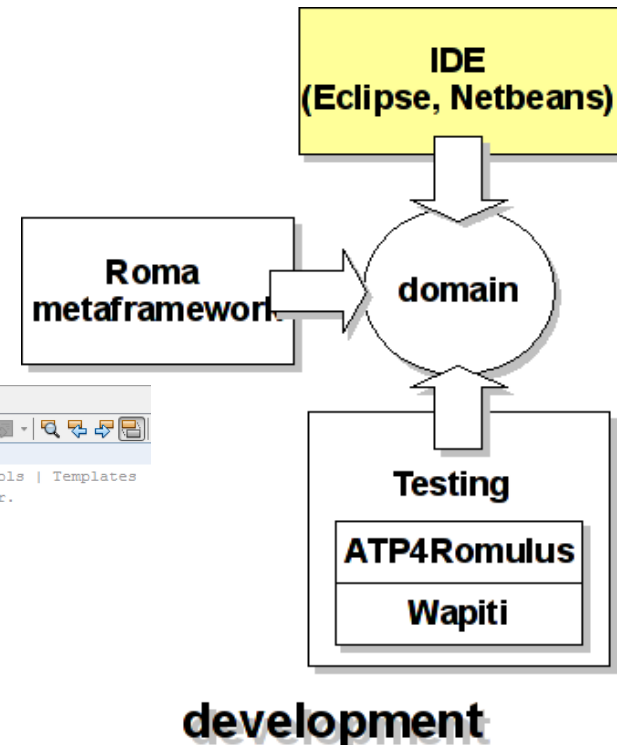
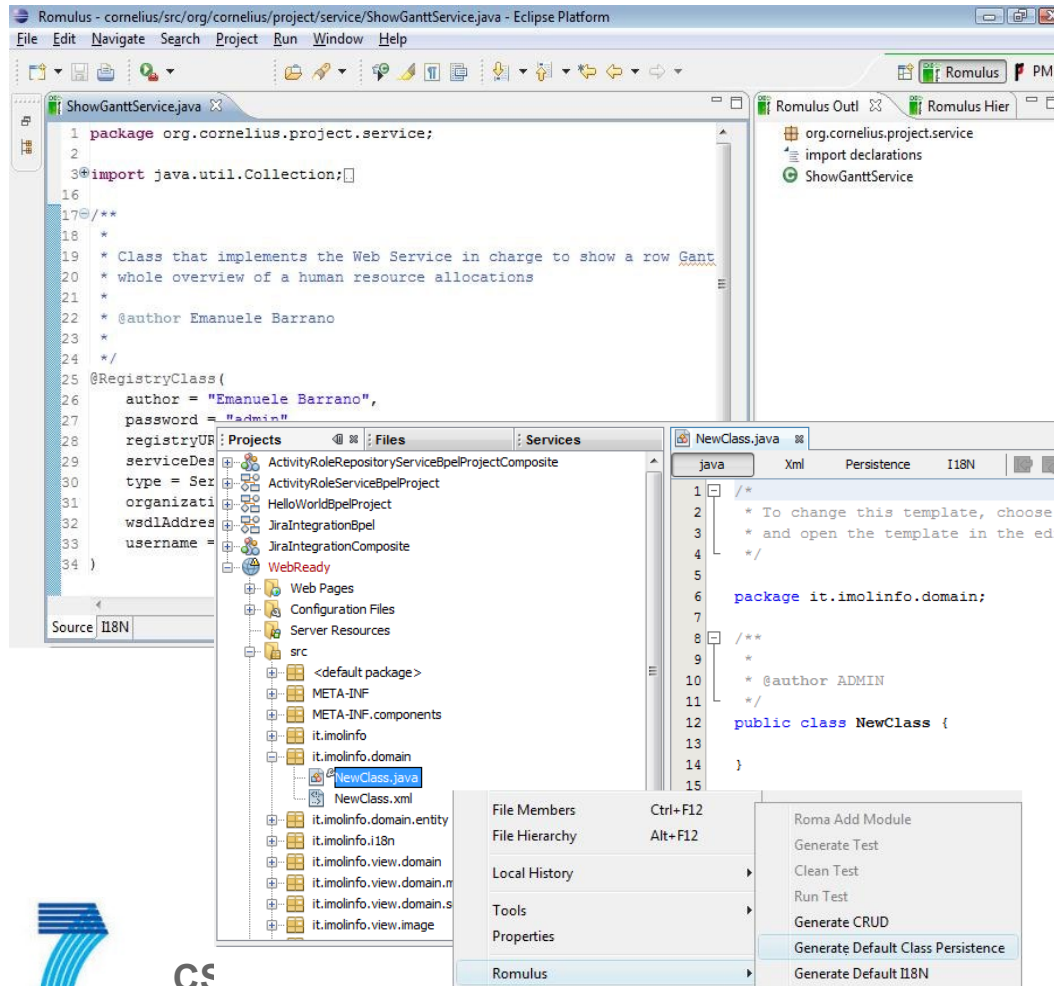
Roma Metaframework



IDE Tools: IDE4Romulus



• Integration with Eclipse and Netbeans



ROMULUS

Testing: ATP4Romulus



ATP4Romulus

- Functional/structural testing: JUnit/TestNG
- Web flow testing: HtmlUnit/Selenium
- Performance/load testing: JMeter/JUnitPerf
- Interaction with database: DBUnit

HtmlUnit

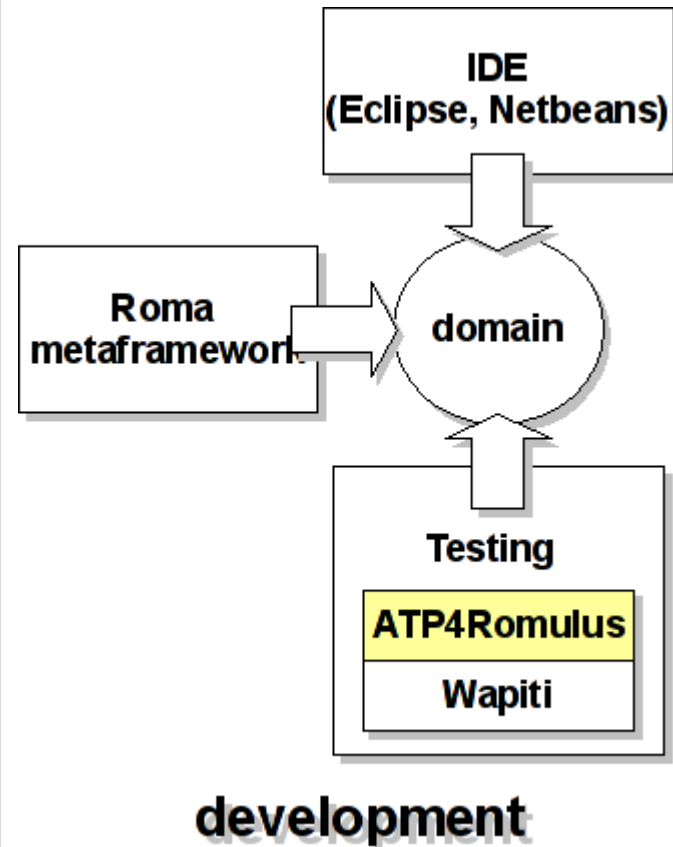
JUnit

dbUnit

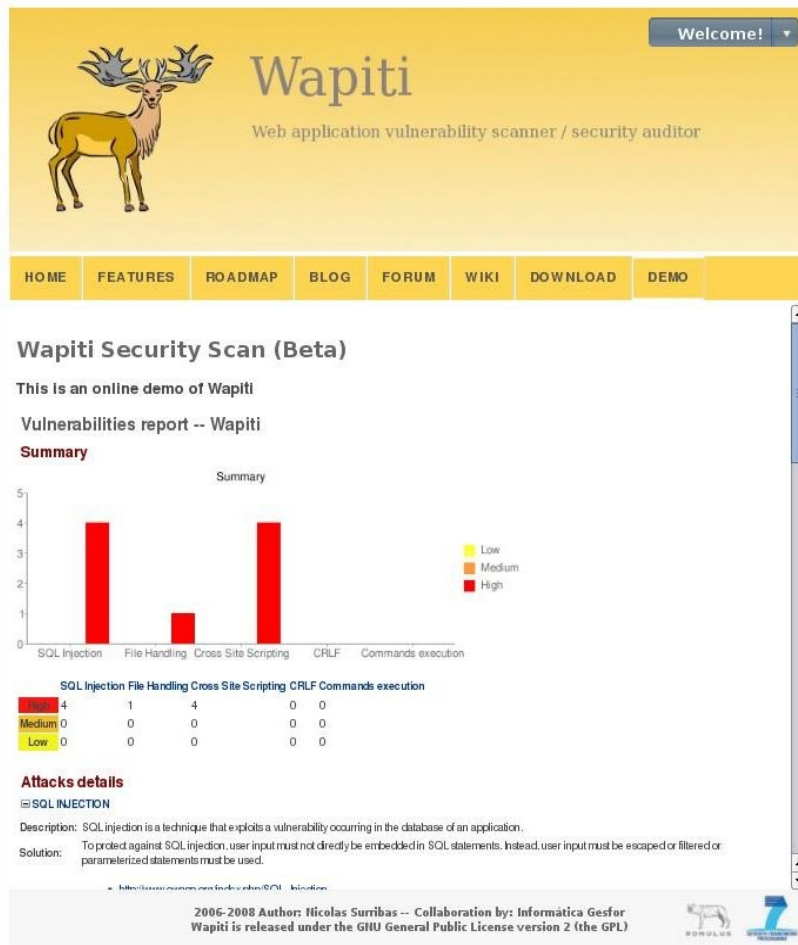


TestNG

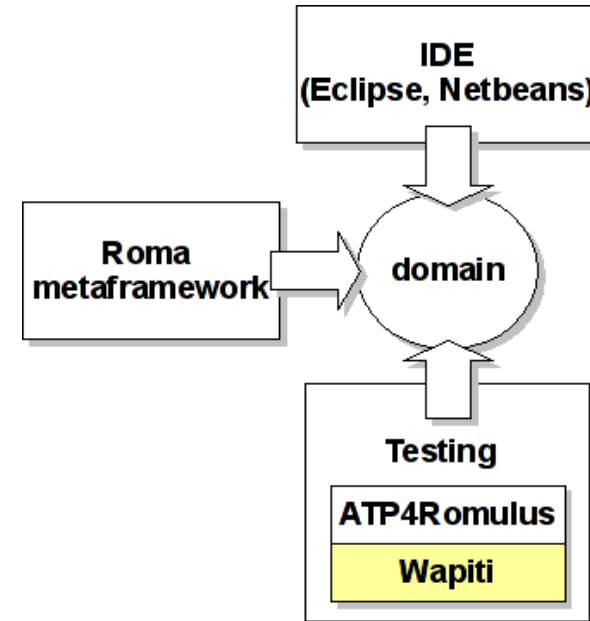
Apache JMeter



Testing: OWASP Wapiti



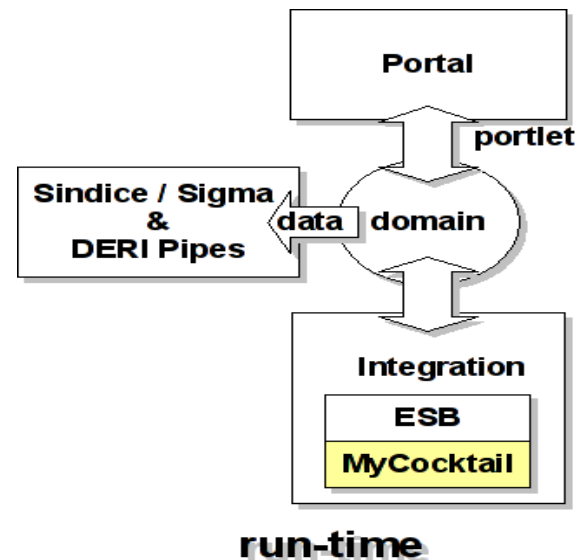
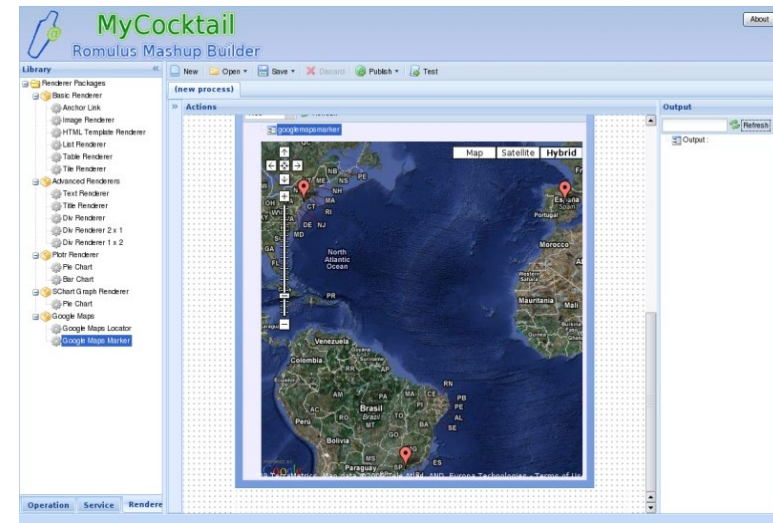
- Security auditor
- Black box testing
- Accepted and integrated in OWASP
- Extensible
- Integrated with Roma



MyCocktail: Mashup Builder

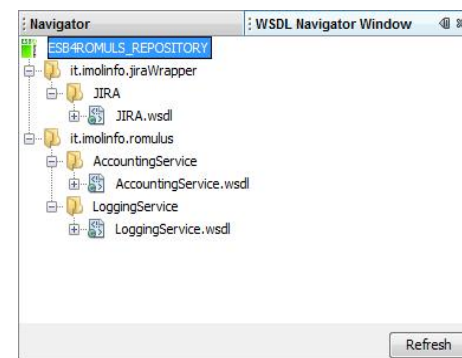
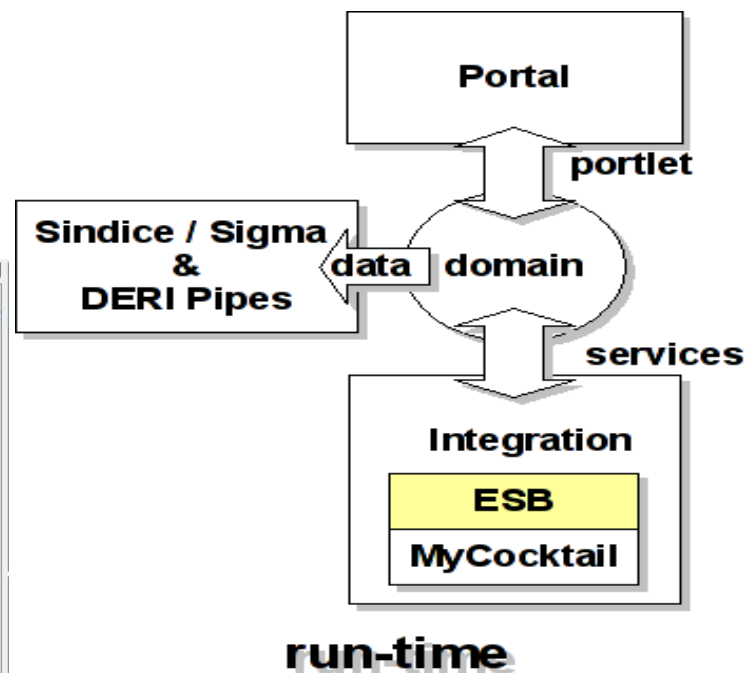
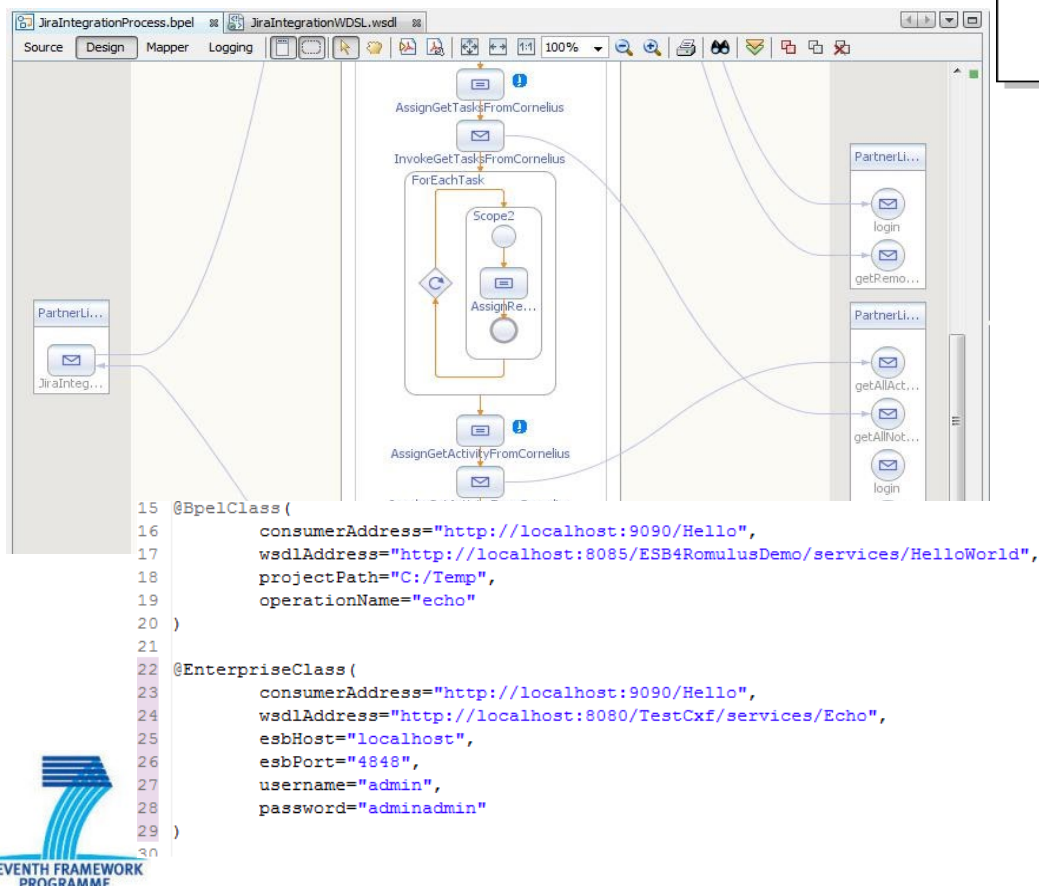


- Client-side Mashup Builder
- Integrates most popular mashups (Flickr, Google Maps, Delicious, ...)
- Exports mashups as JavaScript, HTML, Google Gadgets or Netvibes Gadgets
- Roma apps can be “mashed up” automatically

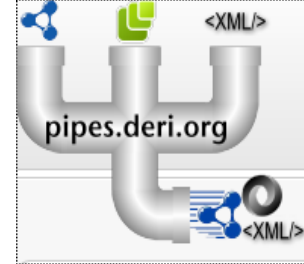


Enterprise Mashups

- JBI compliant
- Integrated with Apache CXF
- Integrated SOAP/Rest Registry



DERI Pipes: Semantic Mashups

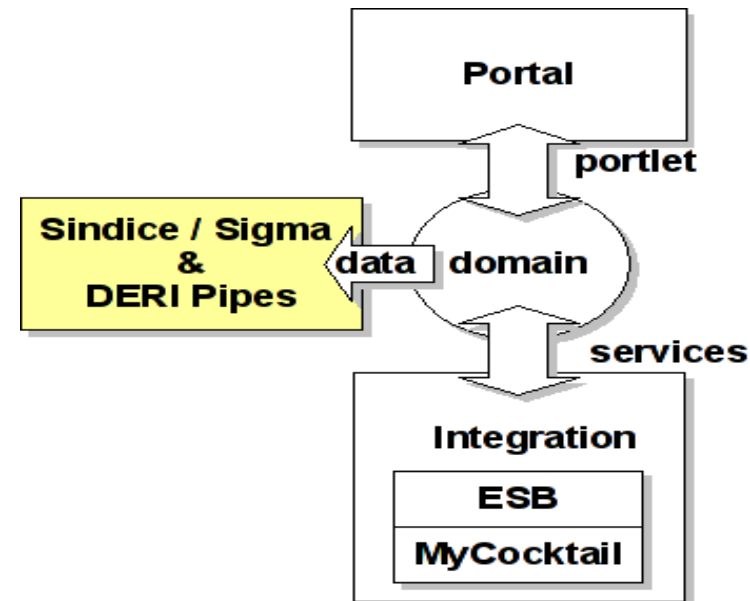


human interactions



software artefacts

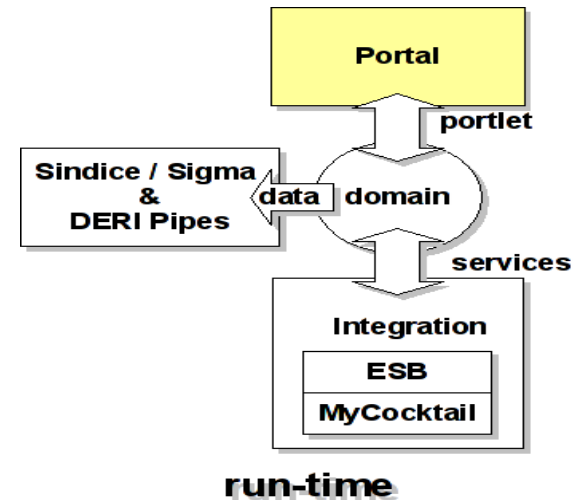
- Integrates the software assets from multiple projects
- Bug Tracker Systems
- Project Management tools
- Java Source code
- Wikis
- Discussion boards



run-time

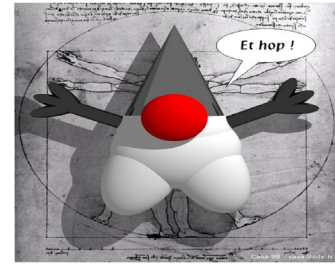
Portal

- Roma built-in JSR-168 portlet integration
- Portal services integration
- Ability to integrate widgets and export portlets as widgets
- Facility for composing application by 'copy&paste' interface with scripting technology



The screenshot shows the Liferay portal interface. The top navigation bar includes 'Welcome' and 'Cornelius'. The main content area is divided into two panes. The left pane shows the 'Cornelius Help System' with a search bar and a list of help topics. The right pane shows the 'Receiving Server' configuration page. This page includes a 'Logout' button, a 'Resources' tab, and a 'Back Office' tab. The 'Receiving Server' section contains a table with columns for Name, Receiving Types, Sending Types, Username, Password, Sending Mail, and Delete From Server. Below the table is a 'Change Password' section. Red arrows point from the 'Cornelius Help System' to the 'Receiving Server' configuration page.

Romulus Suite

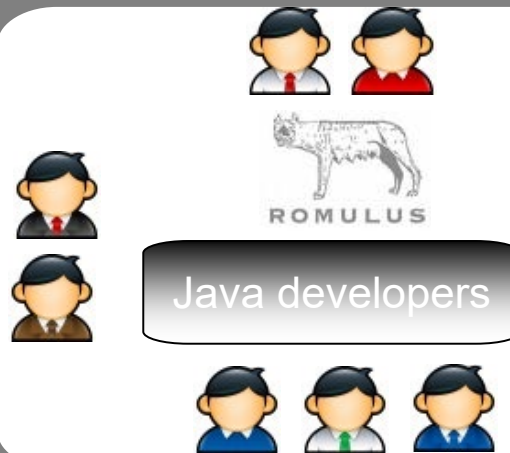


EndUser
Applications

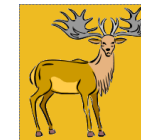
Community
needs



Productivity
needs



Security
needs

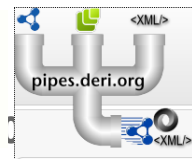


Reliability
needs



Mashup
needs

Enterprise
needs



CSMR, 18tl



Conclusions



- Within Romulus project time frame, most of the objectives of the project have been fulfilled
- Many open-source communities created or updated for Java web development
- Big effort in integration, due to the fact that different tools and methods has been developed by different partners
- Romulus approach provides an easy way to create and evolve web applications using new trends on software engineering such as DDD or mashup development

Thanks for your attention!

Boni García
Universidad Politécnica de Madrid

bgarcia@dit.upm.es

<http://www.ict-romulus.eu/>

