



Overview of Technology Candidates for the European Ground Systems – Common Core (EGS-CC)

Pierre-Yves Schmerber, Thales Alenia Space

SESP 2012

27th September 2012



EGS-CC Background

- The EGS-CC initiative aims at developing a common infrastructure to support space systems monitoring and control in pre- and post-launch phases
 - To improve interoperability of facilities between project phases and between programmes
 - To reduce costs and risks by sharing development, validation, sustaining and maintenance of the common infrastructure



Lower System costs



EGS-CC Background

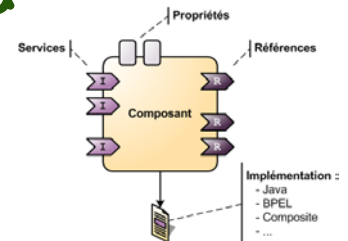
- General EGS-CC design constraints
 - Released under ESA Community Software License with copyleft.
 - Follows a service oriented and component based architecture
 - Linux Operating system based, with Windows compatibility for at least the front-ends (User Interface).
 - Java language used for EGS-CC kernel components, but keep capability to integrate components developed in different languages.



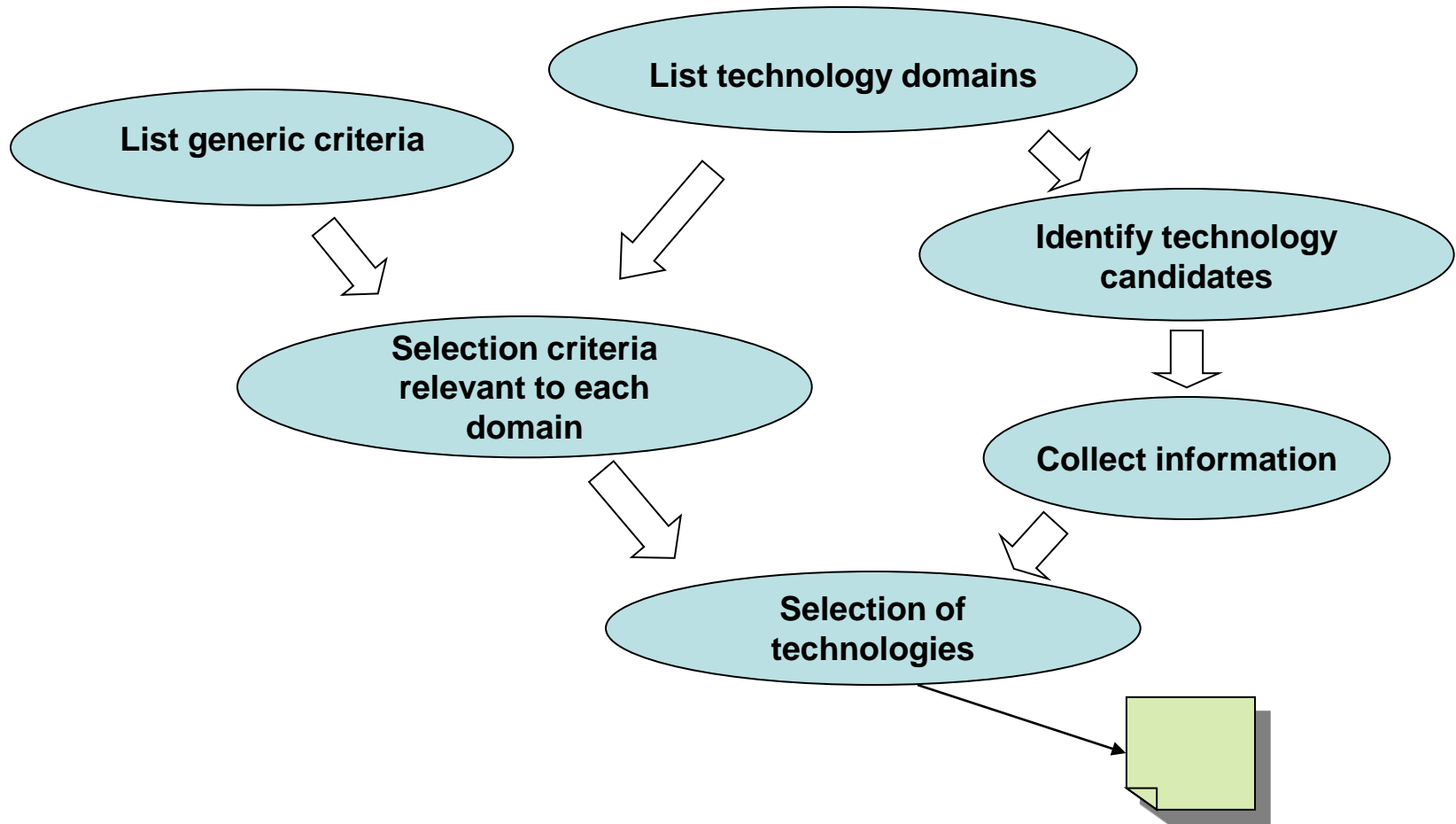
Open Source



Service Based



EGS-CC Technology Selection Process



EGS-CC Technology Selection Criteria

- Selection criteria model based on an extended FURPS+ model
 - The FURPS model
 - Functionality
 - Usability
 - Reliability
 - Performance
 - Supportability
 - The "+" further adds:
 - Design requirements
 - Implementation requirements
 - Interface requirements
 - Physical requirements
 - The EGS-CC specific extension
 - Marketability
 - Security
 - Costs

EGS-CC FURPS Criteria

- **Functionality**
Features set, Capabilities, Generality
- **Usability**
Human Factors, Aesthetics, Consistency, Documentation
- **Reliability**
Bugs & Stability, Recoverability, Predictability, Accuracy, Redundancy
- **Performance**
Speed, Efficiency, Resource consumption, Throughput, Response time
- **Supportability**
Testability, Extensibility, Portability, Adaptability, Maintainability, Compatibility, Configurability, Installability, Scalability, Community

- Design requirement:

Specifies or constrains the options for designing a system. For example, a relational database may be explicitly required.
- Implementation requirement:

Specifies or constrains the coding or construction of a system. Examples include required standards, implementation languages, operating systems and resource limits.
- Interface requirement:

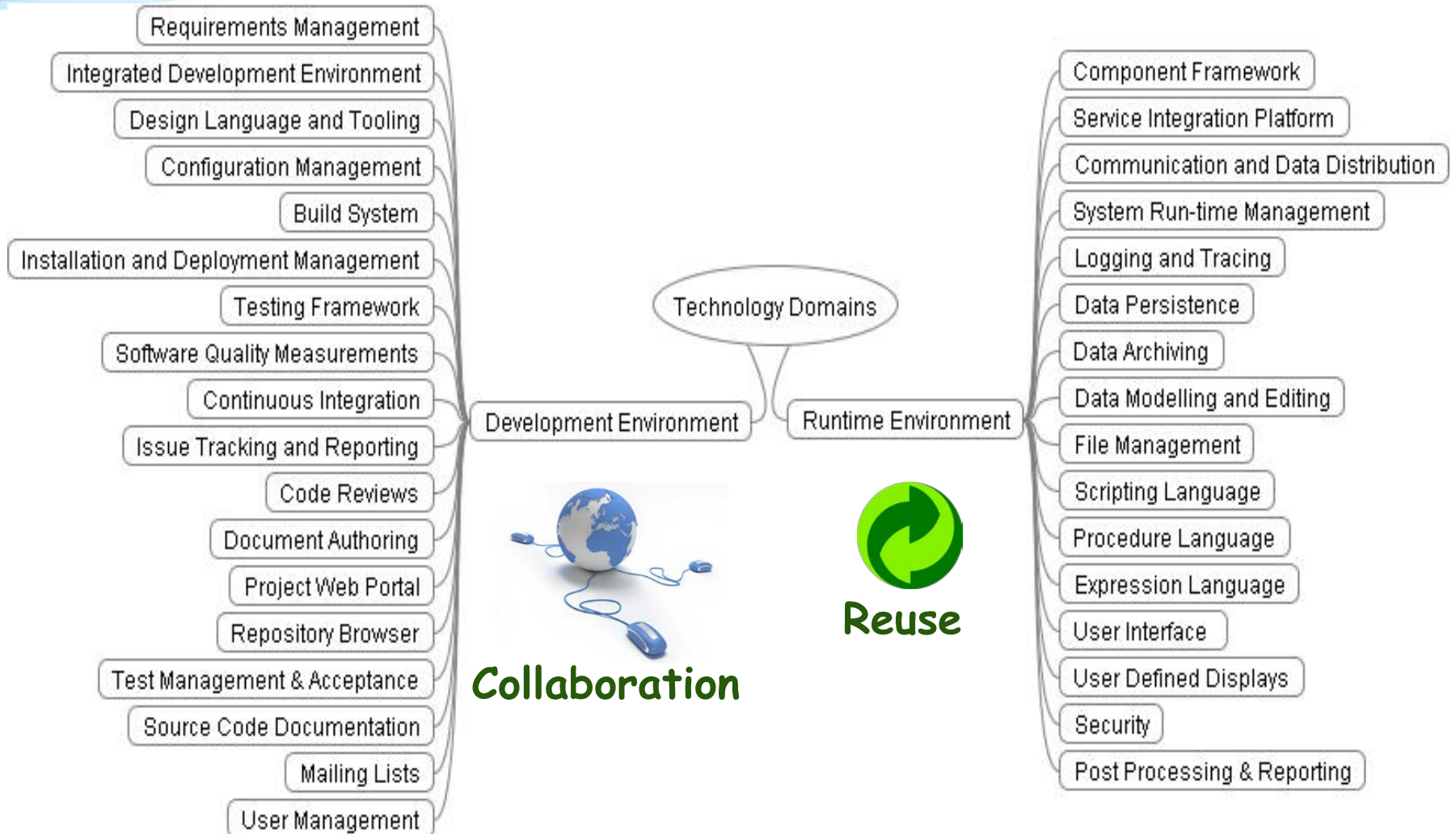
Specifies an external item with which a system must interact, or constraints on formats or other factors used within such an interaction.
- Physical requirement:

Specifies a physical constraint imposed on the hardware used to house the system — for example: shape, size, or weight,

EGS-CC Extension Criteria










- **Costs**
Implementation costs, Possession costs, Configuration costs, Adaptations costs.
- **Marketability**
Licensing, Property rights, Export rights
- **Security**
Authentication, authorization, encryption

EGS-CC Technology Domains













Best overall integration of the technologies is a key driver










Development Environment Technology Candidates 1

Technology Domain	Candidates
Requirements Management	DOORS 
Integrated Development Environment (IDE)	Eclipse 
Design Language and Tooling	UML Enterprise Architect 
Software Configuration Management	Subversion 
Build System	Maven & Nexus 
Installation & Deployment	Maven & RPM 
Unit Testing Framework	Junit / TestNG  
Software Quality Measurements	Sonar / Checkstyle / FindBugs
Continuous Integration	Jenkins 

Development Environment Technology Candidates 2












Technology Domain	Candidates
Issue Tracking and Reporting	JIRA 
Code Reviews	Crucible 
Documentat Authoring	Microsoft Office / (Docbook) 
Project Web Portal	Confluence 
Repository Browser	FishEye 
Test Management & Acceptance	SpiraTest & Jubula  
Source Code Documentation	Javadoc 
Mailing Lists	Mailman 
User Management	Crowd 

Run-Time Environment Technology Candidates 1

Technology Domain	Candidates
Component Framework	OSGI (Equinox, Felix), Spring  
Service Integration Platform	ESB (ServiceMix, JBoss), Camel 
Communication & Data Distribution	JMS (ActiveMQ), ZeroMQ 
System Run-time Management	JMX 
Logging and Tracing	Log4j / Logback 
Data Persistence	JPA, JDO, SDO, EMF 
Data Archiving	RDBS (MySQL), NoSQL (Kyoto Cabinet, HDF5) 
Data Modelling and Editing	EMF 

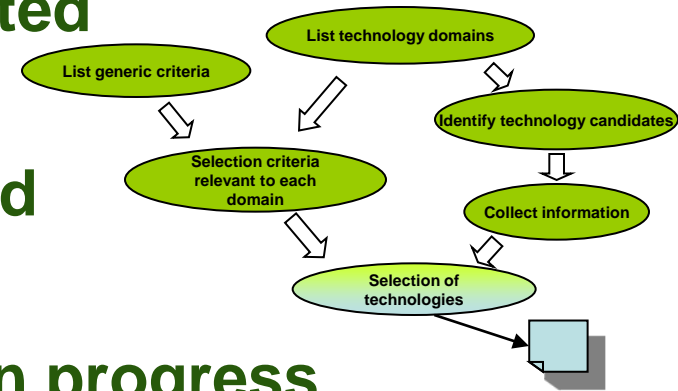


Run-Time Environment Technology Candidates 2

Technology Domain	Candidates
File Management	SVN, GIT, JCR (Jackrabbit)  
Scripting Language	JSR223 API (Rhino, Jython, Groovy) 
Procedure Language	ECSS E70-32, Java, Script Language 
Expression Language	Reuse Script or Procedure Expression Language
User Interface	Eclipse RCP + Thin client (GWT, Flex,...)   
User Defined Displays	Charts (JFreeChart) + Mimics (SVG) + 3D (X3D)  
Security	JAAS, LDAP, Shiro, OpenEM, Progress Actional, CodeLogin, Whitelisting 
Post Processing & Reporting	BIRT 



- Selection Criteria Definition **completed**
- Technology Domains List **completed**
- Candidate Technologies Selection **in progress**



- Development Environment Technologies are selected. A collaborative environment based on these technologies is currently tested in another project.
- Some Run-time Environment technology choices are delayed to the design phase
- Evaluation of critical Run-time Environment technology candidates will soon be contracted to European Industry.

European Ground System – Common Core

Thank You

