Planning the Future with Planets
The Planets Interoperability Framework

Presented by Ross King
Austrian Research Centers GmbH – ARC
Outline

- Motivation
- Architecture
- Demonstration
Interoperability Framework: Motivation
Planets Software: Vision

- Planets tools available in a single downloadable software package
- This package will be simple to install, configure, administer
- When this is deployed, a Planets instance is created, in which
  - an administrator can
    - create user accounts
    - deploy and browse services
    - browse registries
  - a preservation expert can
    - define service workflows (Workflow Design Tool)
    - define and evaluate preservation plans (PLATO Application)
    - define and run experiments (Testbed Application)
  - a librarian or archivist can
    - evaluate and execute preservation plans (PLATO Application)
    - define service workflows from workflow templates and execute preservation processes on a repository (Online Design Tool)
Interoperability Framework: Motivation

- There are a number of functions that all (or nearly all) software applications commonly need. These include functions such as
  - Web application infrastructure
  - Data persistence
  - User management
  - Security, Authentication and Authorization
  - Monitoring, Logging, and Messaging

- In addition, there are some non-functional requirements on the infrastructure, which should be
  - Robust
  - Scalable
  - Distributed

- The **Interoperability Framework (IF)** software components will provide these commonly required functions and meet these non-functional requirements.
Interoperability Framework: Benefits

- **Efficiency**
  - If the above mentioned components are only developed once, rather than multiple times, then the Planets Sub-projects and their applications can concentrate on their specific process logic and will have more time and resources to do so.
  - Also, when packaging the Planets software, the number of components will be optimized; for example, because the IF provides a single database for all components, only one database need be installed.

- **Interoperability**
  - By providing common components, the IF can also help to assure that various applications remain interoperable.
  - By enforcing Web Service standards, the IF can support access to remote and distributed third-party characterization and migration services.
Interoperability Framework: Architecture
Interoperability Framework: Architecture

PLANETS Interoperability Framework

PLANETS Applications
- Workflow Designer
- Administration Tool
- Testbed
- Preservation Planner
- Characterization Tools

Core Elements
- Security/Authentication Authorization
- Monitoring/Logging/Auditing

Service Bus
- Workflow Execution Engine
- Transaction Manager
- Error/Exception Handling
- Database Layer

External Repositories
- External Registries
- External Services

Service Registry
- Planets Service
- Preservation
- Characterization Tool
Interoperability Framework: Application Server

PLANETS Applications
- Workflow Designer
- Administration Tool
- Testbed
- Preservation Planner
- Characterization Tools

PLANETS Interoperability Framework
- Core Elements
  - Security/Authentication Authorization
  - Monitoring/Logging/Auditing
- Workflow Execution Engine
- Transaction Manager
- Error/Exception Handling
- Database Layer

Service Bus
- External Repositories
- External Registries
- External Services
Interoperability Framework: Application Server

- An Application Server is a software engine that delivers applications to client computers or devices, and is necessary in order to host dynamic web applications.

- Advanced application servers provide a number of additional important features, such as Web Service support, thread pooling, and persistence management.

- We have selected the JBoss application server as the most robust, open-source java-based implementation, certified for the Java 2 Enterprise Edition (J2EE) 1.4 standard and supporting Enterprise Java Beans (EJB) 3.0.
Interoperability Framework: Service Registry

PLANETS Interoperability Framework

PLANETS Applications
- Workflow Designer
- Administration Tool
- Testbed
- Preservation Planner
- Characterization Tools
- ...

Service Bus
- External Repositories
- External Registries
- External Services

Core Elements
- Security/Authentication Authorization
- Monitoring/Logging/Auditing
- Workflow Execution Engine
- Transaction Manager
- Error/Exception Handling
- Database Layer

Work Space

Registry Services
Interoperability Framework: Service Registry

- The Planets Service Registry is developed on top of jUDDI, an open-source, Java-based implementation of the Universal Description, Discovery and Integration (UDDI) standard.

- On top of the UDDI standard, we intend to add semantic service descriptions. This will enhance the search for humans, but also support automatic service composition by machines.

- The Service Registry is the central discovery point for internal and external web services, for both users and applications.
Interoperability Framework: Data Registry

PLANETS Interoperability Framework

PLANETS Applications
- Workflow Designer
- Administration Tool
- Testbed
- Preservation Planner
- Characterisation Tools

Service Bus
- External Repositories
- External Registries

Core Elements
- Security/Authentication Authorization
- Monitoring/Logging/Auditing
- Workflow Execution Engine
- Transaction Manager
- Error/Exception Handling
- Work Space

Service Registry
- Planets Service Preservation
- Planets Service Characterisation

Database Layer
- Registry Services

External Registries
- Planets Service
- External

External Repositories
- Planets Service
- External

Characteris...
The Planets Data Registry implements the *Java Content Repository* (JCR) specification, and is built on top of the open-source implementation *Jackrabbit*.

- A JCR implementation has the following features:
  - a standardized approach to content repositories
  - automatic versioning support
  - dynamic definition of complex content models and metadata schema
  - queries using XPATH and XQUERY
  - queries using SQL

- The Data Registry is built on top of a relational database, in our case the open-source Apache *Derby* database management system.
Planets Application: Workflow Design

PLANETS Interoperability Framework

PLANETS Applications

- Workflow Designer
- Administration Tool
- Testbed
- Preservation Planner
- Characterization Tools

Service Bus

Core Elements

- Security/Authentication Authorization
- Monitoring/Logging/Auditing

- Workflow Execution Engine
- Transaction Manager
- Error/Exception Handling
- Database Layer

External Repositories

External Registries

External Services

Planets Application: Workflow Design
Interoperability Framework: Workflow Design

- **Expert Workflow Design Tool**
  - Planets Workflows are expressed using the *Business Process Execution Language*, or BPEL.
    - Why BPEL? The main argument is standardization
      - which means, Planets workflows can be created with any BPEL tool and run on any BPEL compliant engine - that is, they are not tied to the Interoperability Framework
  - The expert workflow design tool is a client application based on the Eclipse BPEL Plugin.

- **Online Workflow Design Tool**
  - A simplified web-based workflow tool requiring no knowledge of BPEL
  - Based on *workflow templates*
    - pre-defined workflow fragments that must be completed by the user by specifying Web service endpoints.
Interoperability Framework: Demonstration
PLANETS Application: Testbed

PLANETS Interoperability Framework

Core Elements
- Security/Authentication Authorization
- Monitoring/Logging/Auditing

- Workflow Execution Engine
- Transaction Manager
- Work Space
- Database Layer

Service Bus
- External Repositories
- External Registries
- Planets Service Preservation
- Planets Service Characterisation

PLANS Applications
- Workflow Designer
- Administration Tool
- Testbed
- Preservation Planner
- Characterisation Tools
Thank you for your attention!

Contact:
Ross King
Austrian Research Centers GmbH – ARC
ross.king@researchstudio.at