

UNIVERSITA' DEGLI STUDI DI NAPOLI "FEDERICO II"



TESI DI LAUREA IN INFORMATICA

UNA WEB APPLICATION SU INFRASTRUTTURA GRID PER IL PROGETTO DAME

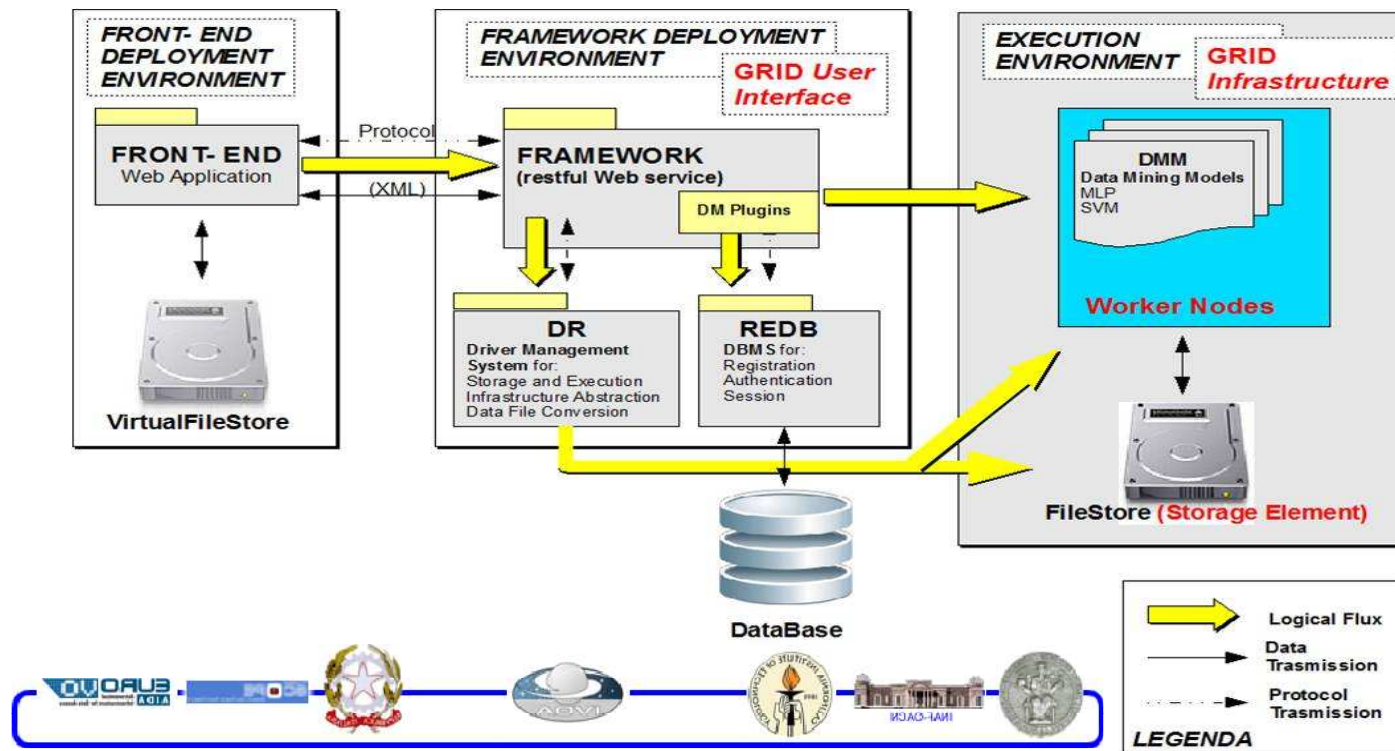
Tutor Accademico:
Dott.ssa Anna Corazza

Tutor Aziendale:
Dott. Massimo Brescia

Candidato:
Manna Francesco
matr. 566/1898

Anno Accademico 2008-2009

DAME (Data Mining and Exploration)



○ Scopo

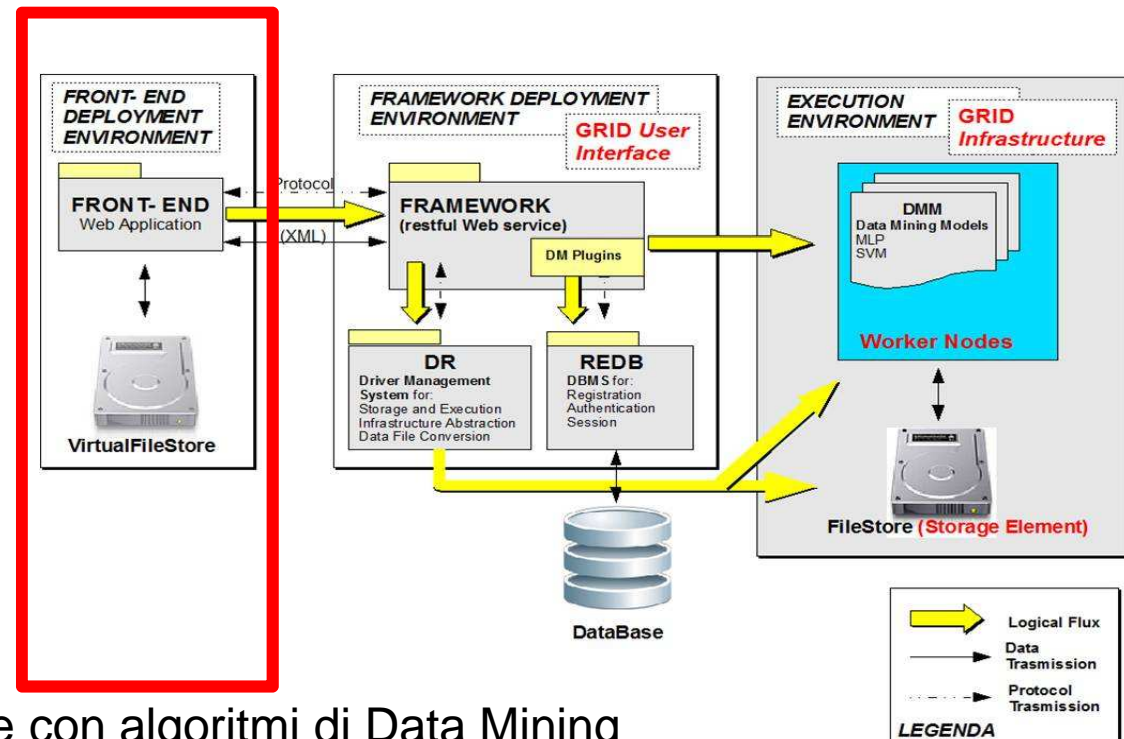
DAME si propone di realizzare una suite web-oriented per effettuare esperimenti di data mining su Massive Data Sets sfruttando le risorse del GRID computing (S.Co.P.E).

Componente Front End (FE)

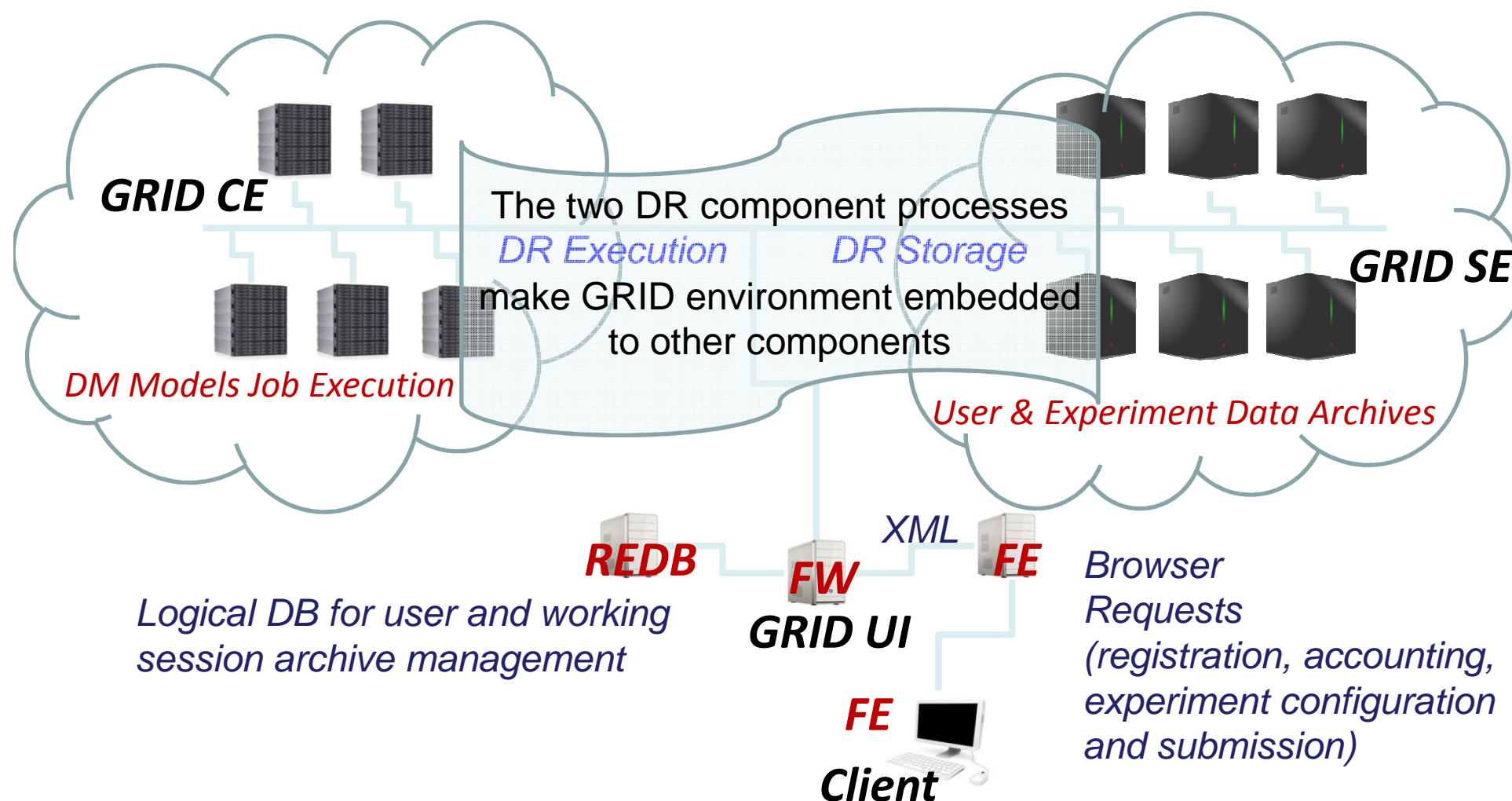
- Consiste in una *Rich Internet Application (RIA)*
- Si occupa dell'interazione con l'utente e con il componente Framework (FW)

Deve permettere all'utente di:

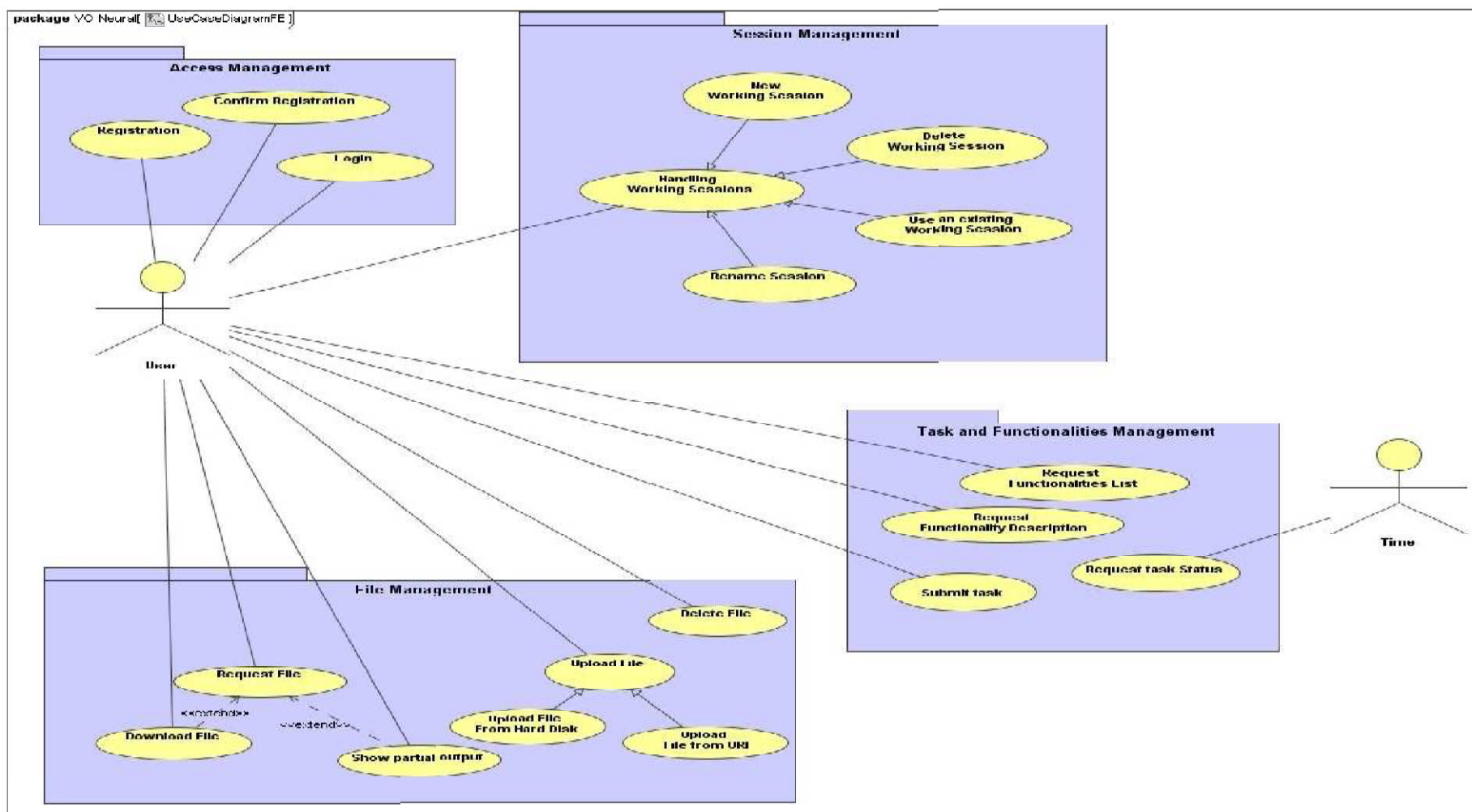
- Effettuare una registrazione
- Gestire il workspace
- Effettuare l'upload di dati
- Creare nuovi Data Set da esplorare con algoritmi di Data Mining
- Effettuare esperimenti (Configurazione ed esecuzione)
- Visualizzare i risultati (testuali/grafici) degli esperimenti



Workflow di DAME su GRID



Requisiti Funzionali

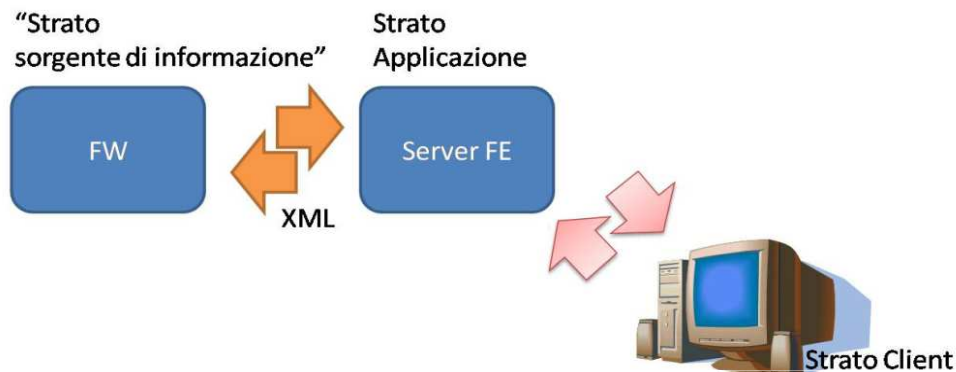


Specifiche di Accesso Utente

Ogni utente ha accesso a:

- **Virtual File Store**
collegamento logico (locale) al File Store reale
- **Session Manager**
gestore workspace (Crea, Rinomina, Elimina)
- **Functionalities Selector**
permette di selezionare e visualizzare le funzionalità di Data Mining disponibili.
- **DataSet Editor**
permette di costruire interattivamente nuovi dataset mediante manipolazione di meta-dati inerenti a file di input/output pre-caricati nel VFS dell'utente

Architettura Front End



La comunicazione tra FE e FW avviene mediante
5 tipi di file XML (VOTable Schema):

1. Error Report XML
2. FunctionalitiesList XML
3. Interactive Report XML
4. Session Manager XML
5. Functionality Description XML

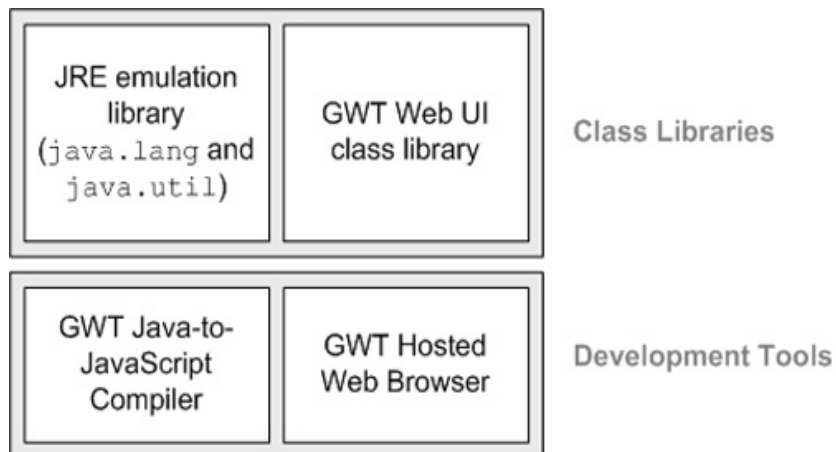
Use Case	Url
Crea Sessione	POST /sessions? ssid=x&userMail=y&name=z
Rinomina Sessione	PUT /sessions? ssid=x&userMail=y&sessionId=z
Cancella Sessione	DELETE /sessions? ssid=x&userMail=y&sessionId=z
Lista Funzionalità	GET /functionalities/list? ssid=x&userMail=y
Descrizione Funzionalità	GET /functionalities/description? ssid=x&userMail=y
Download File	GET /files? fileUri=x&ssid=y&userMail=z
Delete File	DELETE /files? fileUri=x&ssid=y&userMail=z
Upload da HD	POST /files/uploadFromHd? userMail=x&ssid=y&sessionId=z& fileUri=r&format=t
Upload da Uri	POST /files/uploadFromUri? userMail=x& ssid=y&sessionId=z& fileUri=r&format=t&length=u
Dataset Editor	POST /Dataset? userMail=x&ssid=y&length=z& sessionId=t
Crea Esperimento	POST /experiments/newExperiment ? ssid=x&userMail=y&length=z& sessionId=r&experimentName=t
Cancella Esperimento	POST /experiments/delete? ssid=x&userMail=y&experimentId=z
Controlla Stato Esperimento	GET /esperiments/status? ssid=x&userMail=y&experimentId=z
Registra Utente	PUT /users/newUser? name=u&surname=v&userMail=w &country=x&affiliation=y& password=z
Conferma Registrazione Utente	POST /users/newUser ?userMail=x
Login	GET /users/authentication? userMail=x&password=y
Logout	GET /Logout?ssid=x&userMail=y

Tecnologie Utilizzate

Google Web Toolkit (GWT)

GWT è uno tra i tanti toolkit/framework utili per scrivere applicazioni Ajax.

L'approccio di GWT è quello di permettere di scrivere le applicazioni utilizzando il linguaggio Java e solo alla fine convertire (tramite il GWT Java-to-Javascript Compiler) da classi Java a codice Javascript e HTML browser compliant (Internet Explorer, Firefox ect).

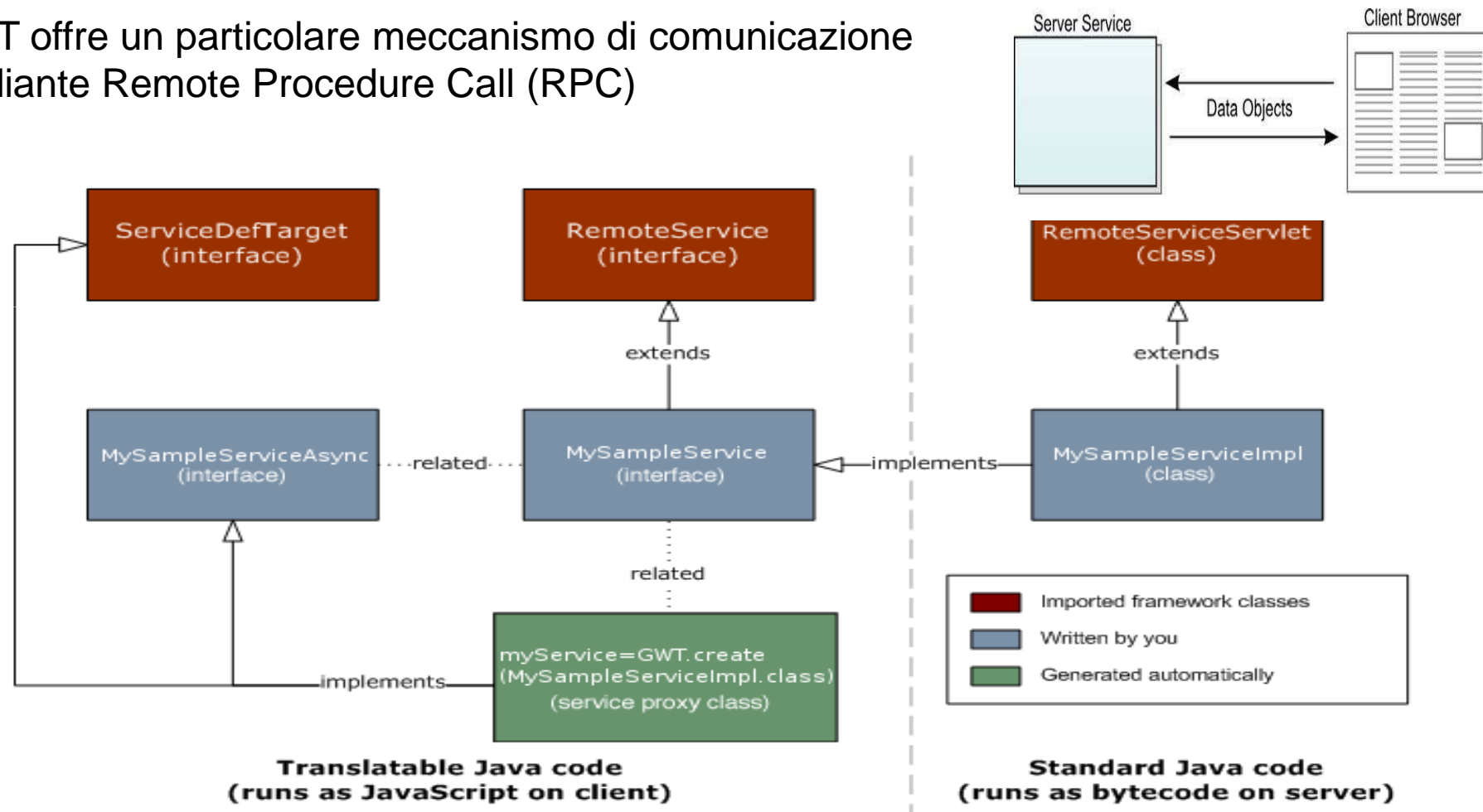


Perché GWT?

- ✓ Compatibilità con la maggior parte dei browser.
- ✓ Risultato di una webapp sviluppata con GWT simile ad una desktop application.
- ✓ Comunicazione client-server mediante *Remote Procedure Call*;
- ✓ Riutilizzabilità e per la sua manutenibilità;
- ✓ Rapida curva di apprendimento;

Comunicazione Client – Server con GWT

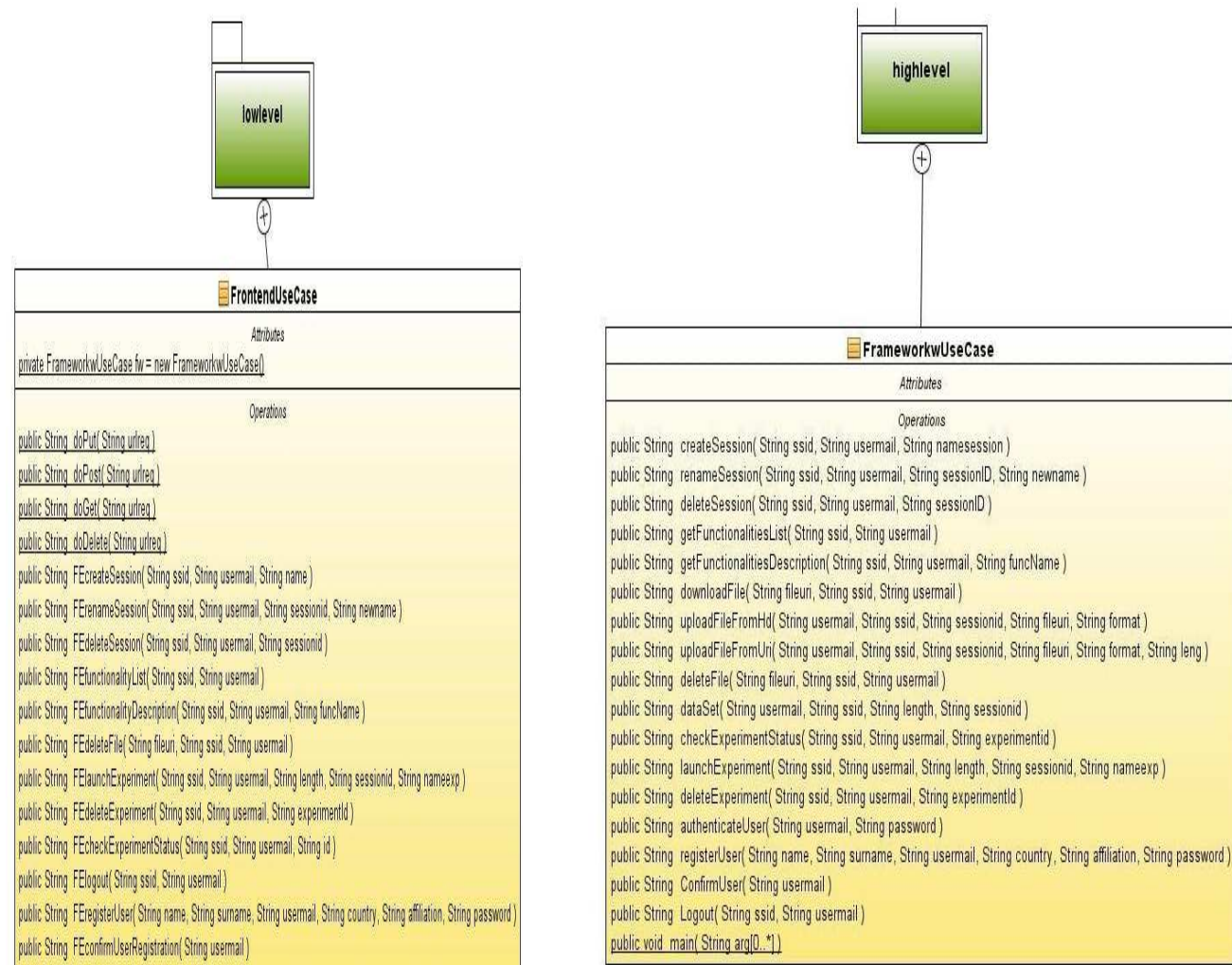
GWT offre un particolare meccanismo di comunicazione mediante Remote Procedure Call (RPC)



Server – Side: librerie High Level e Low Level

Librerie scritte in Java

Scopo: dare la possibilità
a chiunque in futuro
di realizzare la propria
implementazione del FE.




Server – Side : Implementazione dei Servizi e Parsing XML

 XmlParsingAndGenerator
<i>Attributes</i>
<code>private String temp_</code>
<i>Operations</i>
<code>public ErrorState parseErrorReportXml(String s)</code>
<code>public Functionality[0..*] _parseFunctionalityListXml(String s)</code>
<code>public FunctionalityDescription _parseFunctionalityDescriptionXml(String s)</code>
<code>public StatusReport _parseInteractiveReportXml(String result)</code>
<code>public Session[0..*] _parseSessionManagerXml(String s)</code>

 FileManagerServiceImpl
<i>Attributes</i>
<i>Operations</i>
<i>Operations Redefined From FileManagerService</i>
<code>public ErrorState deleteFile(String fileuri, String ssid, String usermail)</code>

 ExperimentServiceImpl
<i>Attributes</i>
<i>Operations</i>
<i>Operations Redefined From ExperimentService</i>
<code>public StatusReport checkStatus(String ssid, String usermail, String id)</code>
<code>public ErrorState deleteExperiment(String ssid, String usermail, String experimentid)</code>
<code>public ErrorState launchExperiment(String ssid, String usermail, String ExperimentName, String inputpar[0..*])</code>

 FunctionalityServiceImpl
<i>Attributes</i>
<i>Operations</i>
<i>Operations Redefined From FunctionalityService</i>
<code>public Functionality[0..*] getFunctionalityList(String ssid, String usermail)</code>
<code>public FunctionalityDescription getFunctionalityDescription(String ssid, String usermail, String funcName)</code>

 SessionServiceImpl
<i>Attributes</i>
<i>Operations</i>
<i>Operations Redefined From SessionService</i>
<code>public ErrorState createSession(String ssid, String usermail, String name)</code>
<code>public ErrorState renameSession(String ssid, String usermail, String sessionid, String newname)</code>
<code>public ErrorState deleteSession(String ssid, String usermail, String sessionid)</code>

 UserRLLServiceImpl
<i>Attributes</i>
<i>Operations</i>
<code>public void _sendMail(String dest, String mitt, String oggetto, String testoEmail)</code>
<i>Operations Redefined From UserRLLService</i>
<code>public String login(String s)</code>
<code>public ErrorState registration(User p)</code>
<code>public ErrorState logout(String ssid, String usermail)</code>

Gestione codici di Errore

Componente Funzionalità Modello Funzione Codice di errore

1	2	3	4	5	6	7	8	9	10	11	12	13
---	---	---	---	---	---	---	---	---	----	----	----	----

1. Component Type

2. Component Built

3-4. Built Version

5-6 Funzionalità (Classification, Regression, Dim Reduction)

7-8 Modello (MLP, SVM, PPS)

9-10 Funzione (Train, Test, Run, Full)

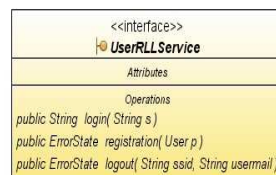
11-12-13 Codice di errore

- Standard tra tutti i componenti
- Composta da 13 caratteri
- Esempio di stringa di errore: aa01 01 02 01 002

Client - Side

La parte Client del Front End è divisa in 2 parti :

- classi interne che effettuano chiamate alle RPC
- GUI Graphical User Interface



Client – Side : Graphical User Interface

SmartGWT

è basato sulla libreria Javascript SmartClient e permette di realizzare il presentation layer delle webapp.

Come Funziona ?

Il codice Javascript, auto-generato dal compilatore GWT, effettua delle chiamate alla libreria SmartClient e il tutto viene visualizzato nel browser.

Perche' utilizzarlo?

- Continua Evoluzione.
- Permette funzionalità avanzate che GWT puro non supporta.
- Ottimo sistema di validazione dei dati con supporto di tooltip, animazioni e simili.
- Garantisce un alto grado di usabilità

Una Web Application su infrastruttura GRID
per il progetto DAME



SmartGWT

Dame - Data Mining and Exploration

Workspace

ID	Name
001234	Regression dataset prova
1023454	Classification Agosto

Workspace Manager

New Rename Delete WS Files

Work

Experiment
14501000
14174000
14175300
14173800
14176100
14173700
14501500
14176000

Registration Form

First Name:

Last Name:

Email: Invalid email address

Password:

Password Again:

Last Access	Status
12/9/2009	■
3/3/2009	■
2/12/2009	■
0/12/2009	■
8/10/2009	■
2/12/2009	■
0/10/2009	■
	■

View Final / Partial Output

Virtual File Store

Help

3. Select the View or Edit tab in the Item Details window as appropriate.
4. Edit information in the Edit pane of the Item Details window and click

Conclusioni – Sviluppi Futuri

- ✓ ulteriori funzionalità alla GUI che rendano ancora più semplice l'uso della suite.
- ✓ Irrobustimento delle politiche di sicurezza.
- ✓ Integrazione del Front End con una nuova versione del Framework sviluppata con KNIME (tool package di data analysis web-based).