

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. i di 21

PROGRAMMA OPERATIVO NAZIONALE 2000-2006 "RICERCA SCIENTIFICA, SVILUPPO TECNOLOGICO, ALTA FORMAZIONE" MISURA II.2 "SOCIETÀ DELL'INFORMAZIONE PER IL SISTEMA SCIENTIFICO MERIDIONALE" AZIONE A – SISTEMI DI CALCOLO E SIMULAZIONE AD ALTE PRESTAZIONI ENEA ENTE PER LE NUOVE TECNOLOGIE, L' ENERGIA E L' AMBIENTE CRESCO Centro computazionale di RicErca sui Sistemi Complessi
Integrare J2EE e AFS: l'esempio di un componente di SIGN-ON
C.R.I.A.I.

Codice Commessa:	ENE06008
Autore:	Team di Progetto (Anna Jannace)
Versione:	x.yy
Data:	30 Luglio 2007
Verificato da:	Verificatore Nome Cognome
Approvato da:	Approvatore Nome Cognome
Lista di distribuzione:	

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. ii di 21

Controllo della configurazione

Titolo: Integrare J2EE e AFS: l'esempio di un componente di SIGN-ON

Riferimento: SPI.2 - SPIII.5. Integrare J2EE e AFS: l'esempio di un componente di SIGN-ON.doc

Creato da: Anna Jannace

Storia del documento

Ver.	Stato	Autore	Data	Memorizzato in
0.01	Bozza	Jannace	17/07/2007	C:\...\Documenti\...\CRESCO\relazioni\CRESCO - SPI.2 - SPIII.5 - Struttura modulo autenticazione AFS.doc
0.02	Bozza	Jannace	24/07/2007	C:\...\Documenti\...\CRESCO\relazioni\CRESCO - SPI.2 - SPIII.5 - Struttura modulo autenticazione AFSv.2007_07_24.doc

Storia delle revisioni

Ver.	Data	Autore	Verif.	Approv.	Motivo Revisione
0.01	23/07/2007	Jannace			Correzione

Modifiche previste

Si prevede di apportare ulteriori integrazioni ed aggiunte alla bozza.

Stampato il: 30 Luglio 2007

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. iii di 21

PROGRAMMA OPERATIVO NAZIONALE 2000-2006
"RICERCA SCIENTIFICA, SVILUPPO TECNOLOGICO, ALTA FORMAZIONE"
MISURA II.2 "SOCIETÀ DELL'INFORMAZIONE PER IL SISTEMA SCIENTIFICO MERIDIONALE"
 AZIONE A – SISTEMI DI CALCOLO E SIMULAZIONE AD ALTE PRESTAZIONI

ENEA
 ENTE PER LE NUOVE TECNOLOGIE,
 L' ENERGIA E L' AMBIENTE

CRESCO
 Centro computazionale di RicErcA
 sui Sistemi Complessi

Integrare J2EE e AFS: l'esempio di un componente di SIGN-ON

C.R.I.A.I.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. iv di 21

Abstract

Il documento è stato prodotto nell'ambito delle attività SPI.2 - Sviluppo ed integrazione dell'architettura GRID e di grafica 3D e SPIII.5 - Interdipendenza tra Reti Complesse del Progetto PON CRESCO (Centro computazionale di RicErca sui Sistemi Complessi) ENEA.

Per tali sottoattività, alla data, sono in fase di realizzazione due sistemi software denominati rispettivamente **MAGO** (Monitoring All Grid Objects) per lo SPI.2 ed **ASSOCIA** (Application Server for Simulation Of Critical Infrastructures and data Adaption) per lo SPIII.5.

Una problematica tecnologica rilevante per i due sistemi software è stata la necessità di integrazione con il sistema Andrew File System (AFS) [4] di Enea.

In questo contesto si inserisce la progettazione di un modulo di accesso denominato SIGN-ON la cui funzionalità primaria è permettere ad un utente accreditato ad accedere alla rete ENEA, di utilizzare le funzionalità, i servizi e le risorse, disponibili sulla rete ENEA-GRID, esponendo una interfaccia applicativa verso MAGO ed ASSOCIA. Tale risultato potrà essere esteso ad ogni altro sistema software che intenda utilizzare tale interfaccia, interagendo in modo trasparente con i servizi offerti dalla infrastruttura di calcolo di ENEA.

In questo documento si intende dimostrare la fattibilità applicativa dell'integrazione tra un servizio web, ovvero il componente SIGN-ON, operante su un application server [1] e il sistema AFS di ENEA.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. v di 21

Indice dei contenuti

Abstract	iv
Indice dei contenuti	v
Indice delle tabelle	vi
Indice delle figure	vii
0. Introduzione	8
1. Scopo e Campo di applicazione	9
2. Funzionalità del sistema	10
3. Architettura del sistema	11
3.1 Application Server	11
3.1.1 <i>L'istanza tecnologica dell'Application Server: JBoss</i>	12
3.1.2 <i>Integrazione tra JBoss e AFS-Enea</i>	14
3.2 Web Application	17
4. Conclusioni finali	19
5. Riferimenti Bibliografici	20
6. Definizioni	21

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. vi di 21

Indice delle tabelle

Tabella n.1 -	<i>La struttura della directory top-level di JBoss.....</i>	<i>13</i>
Tabella n.2 -	<i>La struttura della directory di configurazione del server</i>	
<i>JBoss</i>	<i>14</i>	
Tabella n.3 -	<i>Modifiche effettuate in jboss-service.xml</i>	<i>15</i>

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. vii di 21

Indice delle figure

Figura n.1 -	<i>La struttura della directory di JBoss Application Server..</i>	<i>13</i>
Figura n.2 -	<i>Struttura della pubblicazione web in AFS-ENEA.....</i>	<i>16</i>
Figura n.3 -	<i>Caso d'uso Pubblicazione di Contenuti Web in AFS-ENEA</i>	<i>17</i>
Figura n.4 -	<i>Caso d'uso Navigazione nei contenuti web di ENEA-GRID</i>	<i>17</i>

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 8 di 21

0. Introduzione

In questo documento si descrive il componente software denominato SIGN-ON, allo scopo di evidenziarne le caratteristiche funzionali e tecnologiche.

Nel prossimo paragrafo è descritto lo scenario di riferimento.

Nella sezione “Funzionalità” si definisce la funzionalità del modulo SIGN-ON per evidenziarne le potenzialità derivabili dall'utilizzazione di questo approccio applicativo.

Nel paragrafo “Architettura del sistema” invece si descrive nel dettaglio la struttura tecnologica e si evidenziano le modalità sperimentate durante il processo di integrazione nella infrastruttura AFS-Enea.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 9 di 21

1. Scopo e Campo di applicazione

Il progetto PON Cresco ha come obiettivo la realizzazione, presso il Centro Ricerche ENEA di Portici (NA), di un *polo di Calcolo multidisciplinare per lo studio dei sistemi complessi di natura biologica e tecnologica*, capace di erogare facility di elaborazione, archiviazione e visualizzazione 3D di dati. Il Polo di Calcolo, a regime, sarà collegato con i Centri ENEA di Brindisi e Trisaia (MT).

L'implementazione del componente di SIGN-ON si integra completamente con i seguenti sistemi utilizzati nell'ambito ENEA-GRID:

- **AFS** per la distribuzione delle informazioni;
- **LSF** (Load Sharing Facility) per l'esecuzione di applicazioni.

La caratteristica di base del componente di SIGN-ON è la gestione dinamica e **personalizzabile** dell'input/output da e verso un qualunque browser-client operante su una postazione di un utente, da e verso le applicazioni disponibili sulla Rete ENEA.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 10 di 21

2. Funzionalità del sistema

In sintesi il componente di SIGN-ON consente l'accesso via web all'infrastruttura ENEA sfruttando **le credenziali già disponibili per AFS**.

E' bene precisare che tale funzionalità è attualmente già ottenuta grazie a specifici moduli del *web server* Apache.

Tuttavia, nell'ambito delle attività previste per SPI.2 e SPIII.5, la proposta mira ad offrire **le stesse funzionalità attualmente disponibili** in ENEA-GRID, **sfruttando però le potenzialità di tecnologie software adatte alla gestione dinamica di applicazioni server-side**, quali ad esempio i *servlet container* e gli *application server*.

La soluzione proposta è conforme alle specifiche J2EE (Java 2 Enterprise Edition) [3].

A titolo informativo, si ricorda che mentre la Java 2 Standard Edition (J2SE) [aggiungere il riferimento in bibliografia] comprende, oltre al compilatore ed alla Java Virtual Machine (JVM), tutti i package di base del linguaggio Java, utilizzati per realizzare applet, applicazioni stand-alone con interfacce grafiche e connessioni a database, la Java 2 Enterprise Editino (J2EE) [aggiungere il riferimento in bibliografia] contiene i package necessari per realizzare le servlet, i componenti distribuiti, come gli Enterprise Java Bean (EJB), il Java Message Service (JMS) e le transazioni.

L'adozione delle specifiche J2EE risulta essere di particolare interesse in quanto le caratteristiche di portabilità e distribuzione delle architetture softeare, sviluppate con questa tecnologia, rispondono alle specifiche applicative, in particolar modo per quanto riguardaper lo stack SPIII.5. Ciò è particolarmente adatto alla realizzazione di "un **Application Server**, che risiederà nel Centro di Simulazione, mediante il quale saranno forniti servizi agli utenti finali. Tali servizi verranno erogati sia sfruttando le capacità di calcolo proprie del Centro di Simulazione sia sfruttando, in maniera trasparente all'utente, le capacità di simulazione derivanti dall'utilizzo dei vari codici di simulazione".

Lo scenario architetturale che si prospetta è quello in cui è previsto l'uso di un server applicativo (Application Server). In tale contesto, risulta quindi necessario puntare a tecnologie sufficientemente innovative, che siano, allo stesso tempo validate e performanti.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 11 di 21

3. Architettura del sistema

Il sistema proposto si compone di un Application Server e di una web application.

3.1 Application Server

Un application server è un sistema software che fornisce l'infrastruttura e le funzionalità di supporto, sviluppo ed esecuzione di applicazioni e componenti *server side* in un contesto distribuito. Si tratta di un complesso di servizi orientati alla realizzazione di applicazioni per il web, multilivello ed enterprise, con alto grado di complessità.

L'adozione di un application server offre particolari benefici soprattutto nei settori dello sviluppo, dell'esecuzione e della gestione integrata dei sistemi.

I principali vantaggi possono essere così riassunti:

- Semplificazione delle attività di sviluppo: gli application server creano un ambiente nel quale si possono utilizzare gli strumenti di sviluppo più diffusi sul mercato, consentendo di produrre e distribuire rapidamente applicazioni transazionali altamente scalabili. In generale, questi ambienti comprendono modelli e strumenti di ausilio per sviluppare le applicazioni, riducendo i tempi di realizzazione e messa in esercizio dei programmi negli ambienti distribuiti.
- Supporto di vari linguaggi, strumenti e piattaforme: in funzione dell'application server utilizzato, le applicazioni possono essere scritte nel linguaggio preferito dal programmatore.
- Riusabilità del codice: la riusabilità del codice deriva sia dalla programmazione orientata agli oggetti, spesso utilizzata in questi casi, sia dall'utilizzo dell'approccio di sviluppo a componenti. Una volta sviluppata la logica applicativa, essa può essere condivisa e riutilizzata.
- Gestione delle transazioni: l'application server facilita la gestione delle operazioni basate su transazioni, assicurando l'integrità transazionale e la gestione affidabile dei back-end multipli per le risorse e i dati. Il sistema di gestione delle transazioni gestisce le interazioni con i database e le funzioni di: commit, rollback e recovery.
- Scalabilità: gli application server supportano il partizionamento delle applicazioni e la distribuzione in rete dei componenti. I sistemi multiprocessore e i cluster di application server assicurano la

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 12 di 21

scalabilità necessaria a gestire anche un gran numero di utenti concorrenti.

- Alte prestazioni: gli application server offrono caratteristiche architetturali che permettono di erogare elevate prestazioni. Tra esse: il multithreading, il bilanciamento dinamico dei carichi di lavoro (load balancing), il caching e il pooling degli oggetti e delle connessioni ai database.
- Estensibilità: l'architettura modulare degli application server e il supporto per i server e per i moduli applicativi, che possono essere caricati dinamicamente, consente alle aziende di estendere facilmente le funzionalità dei loro sistemi e delle relative applicazioni.
- Robustezza: l'architettura basata sui componenti degli application server e il bilanciamento dinamico dei carichi assicurano l'alta disponibilità dei sistemi. I componenti del server e la logica applicativa possono essere riconfigurati, aggiunti o rimossi senza interruzioni nell'erogazione dei servizi agli utenti. Queste caratteristiche sono particolarmente importanti per garantire l'alta disponibilità del sistema, requisito necessario per il buon esito delle operazioni mission-critical aziendali.
- Sicurezza: gli application server offrono funzioni specifiche di sicurezza end-to-end. Esse sono necessarie per l'esecuzione delle applicazioni aziendali che richiedono particolari misure di sicurezza e riservatezza dei dati. Per le comunicazioni tra client e server, vengono impiegati algoritmi standard, ampiamente testati e collaudati sul web, come quelli offerti dal protocollo SSL. Il logging e il tracking degli eventi forniscono protezione dagli accessi non autorizzati.

3.1.1 *L'istanza tecnologica dell'Application Server: JBoss*

JBoss [2] è un application server **open source**. E' conforme a tutte le specifiche J2EE imposte dalla Sun Microsystems e può essere installato su tutte le piattaforme (compreso i mainframe AS400 e OS390).

Si tratta di un motore applicativo in grado di contenere e gestire applicazioni Web ed Enterprise scritte in Java secondo le specifiche J2EE.

L'utilizzo di questa tecnologia porta con sé il vantaggio di essere basato su standard e di supportare le ultime novità tecnologiche nell'ambito dello sviluppo java quali il motore di persistenza Hibernate e le specifiche Enterprise Java Bean (EJB) 3.0.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 13 di 21

L'installazione di JBoss è molto semplice, ed i dettagli della stessa sono ampiamente documentati [inserire il riferimento ai manuali o alle procedure di installazione.

In sintesi l'installazione della distribuzione di JBoss per una architettura hardware (es. INTEL) crea una directory *jboss-*(es. 4.0.4)* che contiene gli script di start del server, JARs, i set di configurazione del server e le working directories.

La figura seguente illustra la directory di installazione del server JBoss.

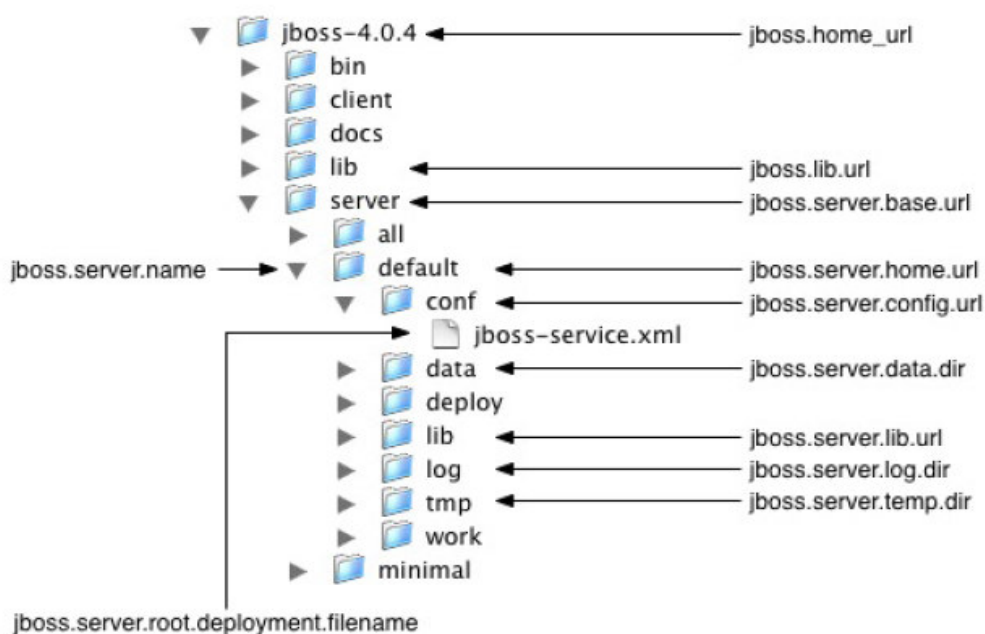


Figura n.1 - La struttura della directory di JBoss Application Server

La tabella seguente mostra le cartelle di top-level e le loro funzioni principali.

Directory	Description
bin	All the entry point JARs and start scripts included with the JBoss distribution are located in the bin directory.
client	The JARs that are required for clients that run outside of JBoss are located in the client directory.
server	The JBoss server configuration sets are located under the server directory. The default server configuration set is the server/default set. JBoss ships with minimal, default and all configuration sets. The subdirectories and key configuration files contained default configuration set are discussed in more detail after
lib	The lib directory contains startup JARs used by JBoss. Do not place your own libraries in this directory

Tabella n.1 - La struttura della directory top-level di JBoss

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 14 di 21

La tabella seguente mostra le cartelle interne a quella della configurazione del server (*jboss*-(es. 4.0.4)/server/(es. *default*))e le loro funzioni.

Directory	Description
conf	The <code>conf</code> directory contains the <code>jboss-service.xml</code> bootstrap descriptor file for a given server configuration. This defines the core services that are fixed for the lifetime of the server.
data	The <code>data</code> directory is available for use by services that want to store content in the file system.
deploy	The <code>deploy</code> directory is the default location the hot deployment service looks to for dynamic deployment content. This may be overridden through the <code>URLDeploymentScanner</code> URLs attribute.
lib	The <code>lib</code> directory is the default location for static Java libraries that should not be hot deployed. All JARs in this directory are loaded into the shared classpath at startup.
log	The <code>log</code> directory is the directory log files are written to. This may be overridden through the <code>conf/log4j.xml</code> configuration file.
tmp	The <code>tmp</code> directory is used by JBoss to store temporarily files such as unpacked deployments.

Tabella n.2 - La struttura della directory di configurazione del server JBoss

Nella cartella “deploy” si trovano le applicazioni pubblicate, che vengono eseguite attraverso un browser.

3.1.2 Integrazione tra JBoss e AFS-Enea

In questo paragrafo si descrivono i dettagli implementativi della sperimentazione effettuata dal CRIAI, al fine di evidenziare l'integrabilità con l'infrastruttura applicativa già esistente in AFS-Enea.

Innanzitutto è stato installato JBoss su un PC dotato di sistema operativo Scientific Linux 4 e di un client AFS.

Nel file *jboss-service.xml*, che è uno dei file di configurazione di JBoss, posizionato in *DIR_INSTALLAZIONE_JBOSS/server/default/conf* è stata aggiunta nella sezione “Deployment Scanning” all'interno del tag con attribute name “URLs”, l'indicazione *file:/afs/enea.it/.../public_html*.

```

<!-- ===== -->
<!-- Deployment Scanning -->
<!-- ===== -->
<!-- An mbean for hot deployment/undeployment of archives. -->
<mbean code="org.jboss.deployment.scanner.URLDeploymentScanner"

```

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 15 di 21

```

name="jboss.deployment:type=DeploymentScanner,flavor=URL">
<depends optional-attribute-name="Deployer">
    jboss.system:service=MainDeployer
</depends>
<attribute name="URLComparator">
    org.jboss.deployment.DeploymentSorter
</attribute>
<attribute name="FilterInstance"
    attributeClass="org.jboss.deployment.scanner.DeploymentFilter"
    serialDataType="javaBean">
<attribute name="ScanPeriod">5000</attribute>
<attribute name="ScanEnabled">true</attribute>

<attribute name="URLs">
    deploy/,file:/afs/enea.it/por/user/jannace/public_html
</attribute>
<attribute name="RecursiveSearch">True</attribute>
</mbean>

```

Tabella n.3 - Modifiche effettuate in jboss-service.xml

Ciò è stato fatto per consentire al server JBoss di estendere la pubblicazione delle proprie applicazioni anche al sistema AFS-Enea.

L'infrastruttura in questo modo è dotata di 2 server:

- il web server Apache all'indirizzo www.afs.enea.it
- l'application server JBoss ad un altro indirizzo IP (da definirsi)

In questo modo, richiamando da un browser, l'indirizzo:

http://indirizzo_IP_server_JBoss/Nome_Applicazione

sarà possibile visualizzare qualunque **applicazione web** che si trova nella directory [/afs/enea.it/.../public_html](http://afs/enea.it/.../public_html).

Si sottolinea che il web server Apache può continuare a funzionare secondo le attuali modalità di utilizzo, pubblicando le pagine HTML contenute in [/afs/enea.it/.../public_html](http://afs/enea.it/.../public_html).

Sia il web server che l'application server possono pubblicare contenuti della medesima directory di AFS-ENEA

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 16 di 21

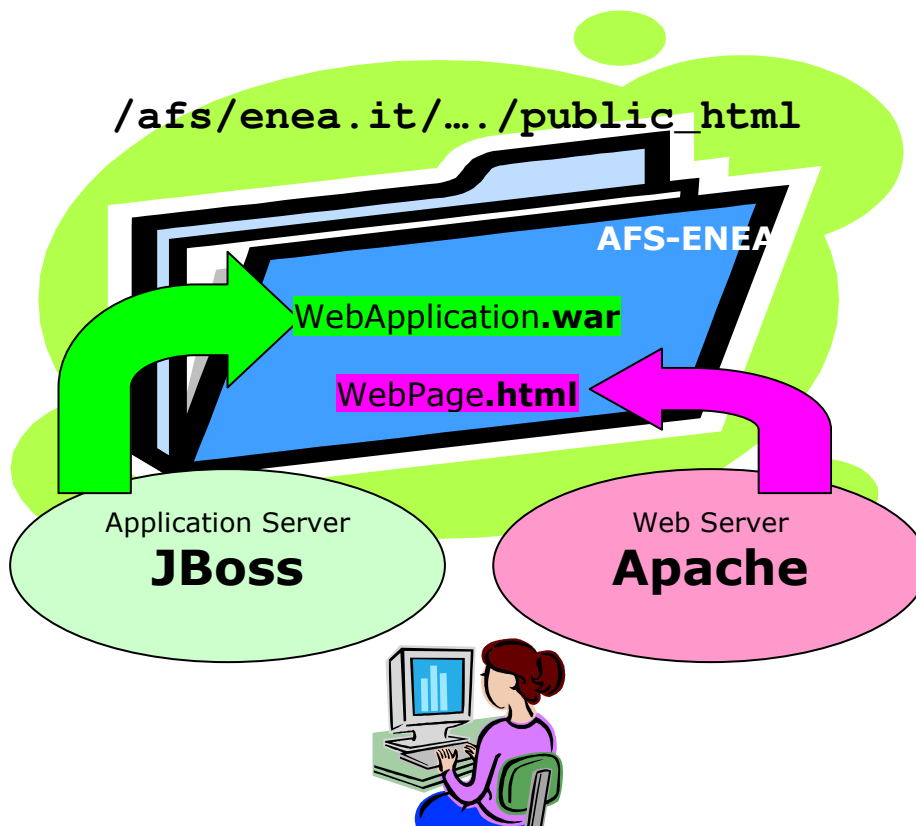


Figura n.2 - Struttura della pubblicazione web in AFS-ENEA

Ma essendo esso un web server, digitando nella barra degli indirizzi di un browser

[http:// www.afs.enea.it/user/Nome_Applicazione_Dinamica](http://www.afs.enea.it/user/Nome_Applicazione_Dinamica)

la web application non viene ovviamente elaborata e visualizzata come accade nel caso di un Application Server.

Se si richiama il nome completo del file che si trova in public_html, cioè

http://www.afs.enea.it/user/ Nome_Applicazione_Dinamica.war

allora il web server Apache ne consente il download al browser che lo sta richiedendo.

Seguono i casi d'uso dello scenario descritto.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 17 di 21

Un attore è l'utente dell'infrastruttura AFS-ENEA di tipo editore di contenuti web (Editore)

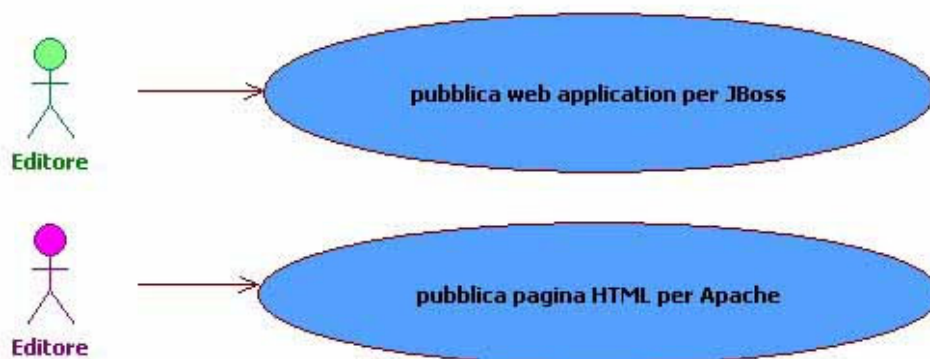


Figura n.3 - Caso d'uso Pubblicazione di Contenuti Web in AFS-ENEA

Un altro attore è l'utente generico dell'infrastruttura ENEA-GRID di tipo fruitore (Utente).

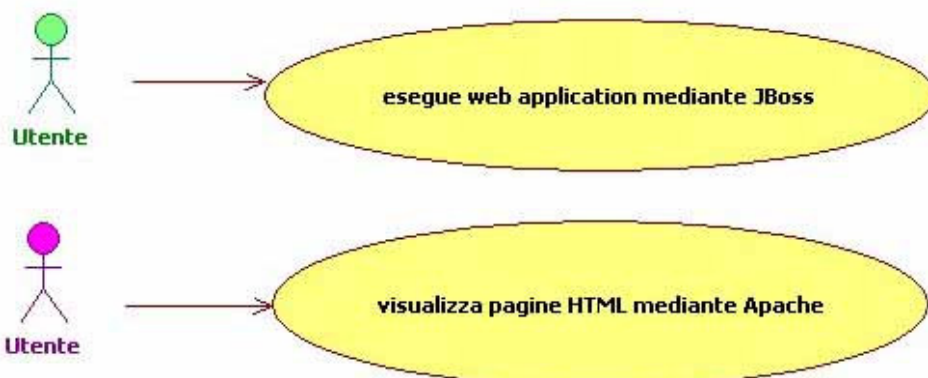


Figura n.4 - Caso d'uso Navigazione nei contenuti web di ENEA-GRID

3.2 Web Application

Una applicazione Web è una applicazione per Internet, a cui gli utenti accedono tramite un Web browser. Dal punto di vista del browser, l'interazione con una applicazione web è indistinguibile dall'accesso a un sito Web statico. Le pagine visualizzate dal browser, in questo caso, sono però generate dinamicamente dall'applicazione.

La web application in oggetto è costruita secondo il pattern Model View Controller (MVC) [5].

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 18 di 21

Essa è costituita da 3 livelli distinti:

- VIEW pagine JSP come web front end,
- CONTROLLER servlet,
- MODEL moduli applicativi per accedere ad AFS-Enea .

Nel contesto di questo documento è necessario evidenziare la modalità di pubblicazione della *web-application*.

Una web application può essere contenuta in un file con estensione “.war”.Una caratteristica importante del file in formato .war è quella di essere compresso.

In genere il file dell'applicazione è posizionato nella directory “/deploy” di JBoss. Esso viene elaborato dinamicamente dall'application server che ne produce di volta in volta le pagine HTML richieste.

Tale operazione è trasparente all'utente di un ambiente complesso e distribuito, come ad esempio quello di AFS-Enea. In ciò interviene la personalizzazione della configurazione di JBoss presentata in precedenza.

E' necessario posizionare il file *Nome_Applicazione_Dinamica.war* nella directory */afs/enea.it/.../public_html* per pubblicarla sul server JBoss, in maniera del tutto analoga a quanto viene fatto attualmente per la pubblicazione delle pagine HTML in AFS-Enea.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 19 di 21

4. Conclusioni finali

La soluzione descritta è stata sperimentata ed applicata dal CRIAI nell'ambito delle attività SPI.2 ed SPIII.5 del progetto PON CRESCO di Enea.

La soluzione applicativa, basata sull'uso di un Application Server, è modulare e scalabile.

Tale soluzione verrà predisposta per i sistemi MAGO ed ASSOCIA, previsti per la realizzazione del progetto CRESCO, che sono attualmente in fase di implementazione.

La soluzione potrà essere riutilizzata ed estesa nell'infrastruttura ENEA-Grid, per ogni ulteriore realizzazione che si vorrà effettuare.

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 20 di 21

5. Riferimenti Bibliografici

- [1] http://it.wikipedia.org/wiki/Application_server
- [2] <http://www.jboss.org/>
- [3] <http://java.sun.com/javaee/>
- [4] http://en.wikipedia.org/wiki/Andrew_file_system
- [5] <http://en.wikipedia.org/wiki/Model-view-controller>

C.R.I.A.I. Consorzio Campano di Ricerca per l'Informatica e l'Automazione Industriale	MODULISTICA QUALITA' Documentazione Tecnica	MOD-DT-73 Rev. A del 01.04.05
emesso da RAQ	Approvato da DG	pag. 21 di 21

6. Definizioni

CRESCO – Centro computazionale di RicErca sui Sistemi Complessi

MAGO – Monitoring All GRID Objects

ASSOCIA – Application Server for Simulation Of Critical
Infrastructures and data Adaption

AFS – Andrew File System

LSF – Load Sharing Facility

J2SE – Java 2 Standard Edition

J2EE – Java 2 Enterprise Edition

EJB – Enterprise Java Bean

JMS – Java Message Service