jAPS incontra Spring

Rinaldo Bonazzo
http://www.jroller.com/rbonazzo/

Chi sono

Responsabile informatico di Sardinia Point Srl (www.sardiniapoint.it) società che:

- sviluppa strumenti informatici destinati alle attività turistiche;
- produce contenuti finalizzati alla promozione turistica della Sardegna;
- gestisce il sito www.rent-sardinia.com oggetto della presentazione dell'anno scorso.

Cosa faccio

Sviluppo portali web

Ambiente di riferimento Java
Frameworks Spring, Struts, iBatis, hibernate, ...
Ajax (DWR, Echo2, ...)

Docente corsi :

- Java;
- Ajax;
- jAPS;
- ...

Cosa cercavo

Uno strumento che mi consentisse di ...

Cosa cercavo (1)

Cosa cercavo (2)

Ops...

Cercavo uno strumento per creare portali con uno o più di queste caratteristiche:

- Semplicità d'installazione;
- Supporto I18N (multilingua);
- Facile implementare servizi aggiuntivi;
- Una ampia comunity di sviluppatori;
- Italiano;
- ...

Cosa cercavo (3)

Dopo aver visto ed utilizzato una serie di sw per la creazione di portali Liferay, jetSpeed, easycms, ...

Ho scoperto che esisteva un progetto Open Source realizzato a Cagliari (jAPS www.japsportal.org).

Prime impressioni (1)

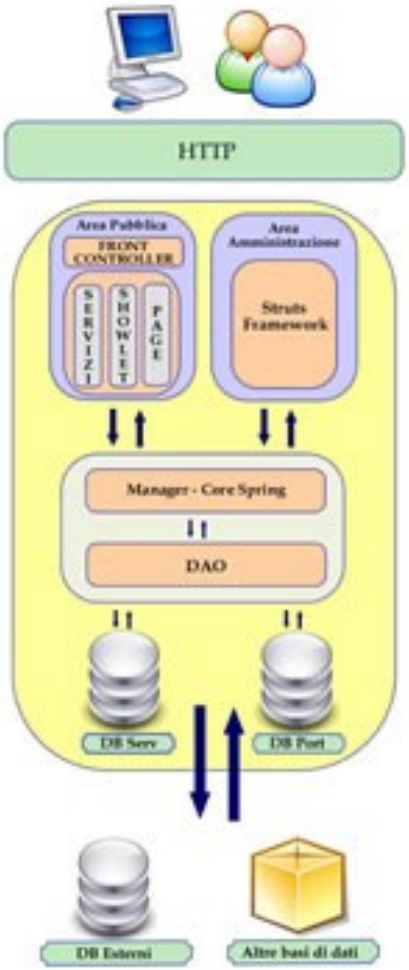
- Installazione prodotto, ok;
- I18N, ok;
- Comunity, ok;
- Italiano, ok (realizzato in Sardegna il massimo);
- Utilizzo, ok;
- Un plus, Rispetta legge Stanca;
- Creazione Servizi, da controllare;

Spring?

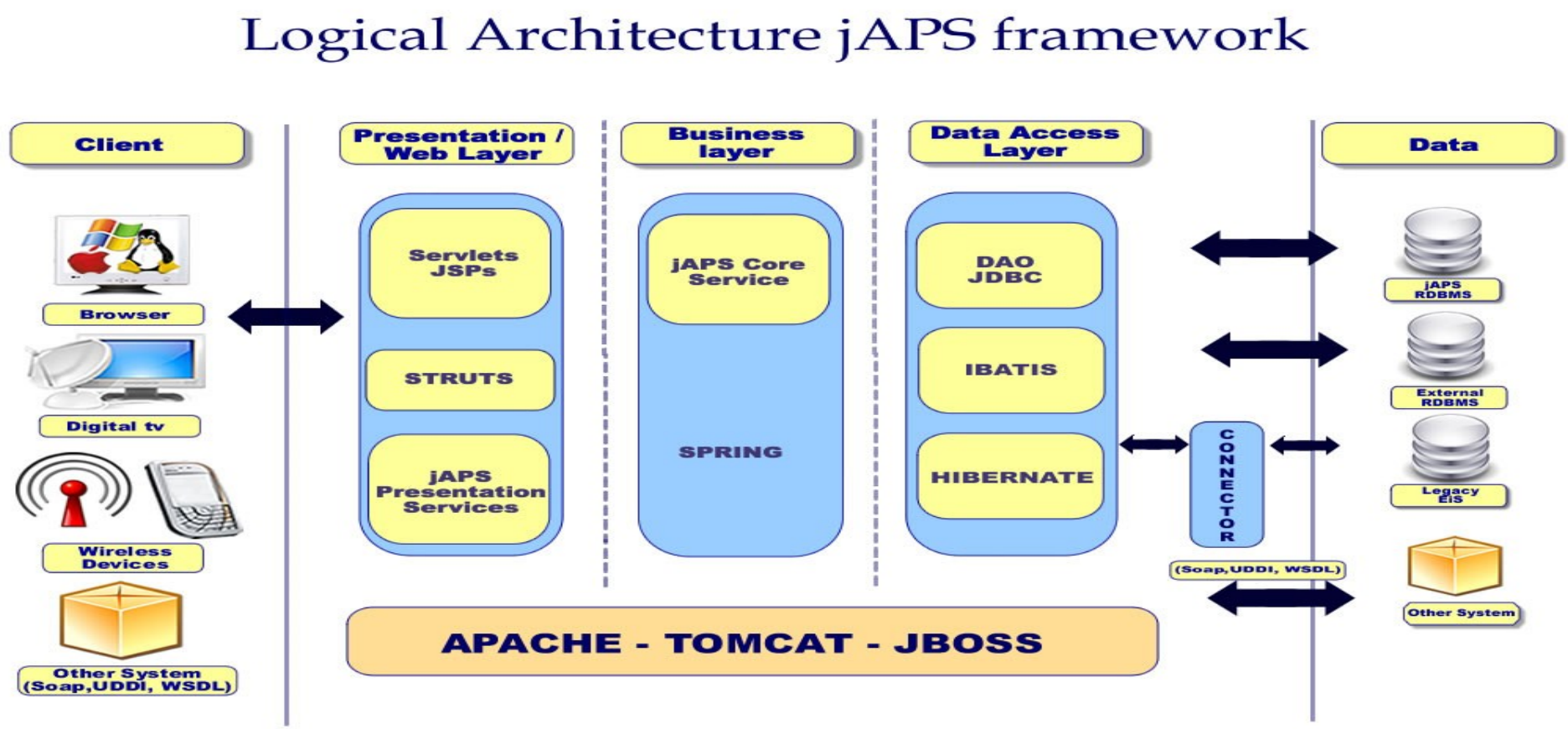
Parlando con il gruppo di sviluppo chiedo perché non usare Spring?

La risposta è positiva e nasce piano piano jAPS con Spring al suo interno.

jAPS e Spring Architettura



jAPS e Spring Architettura



jAPS e Spring - Gestione Db esterno(1)

Prima:

- Creare classe Listener

```
public class RsStartupListener extends StartupListener {
    /**
     * Recupera e restituisce le risorse e i parametri necessari alll'inizializzazione.
     * I parametri sono presi dagli InitParmeter del container, le risorse da JNDI
     */
    protected Map getConfig(ServletContext svCtx) throws ApsSystemException {
            Map props = super.getConfig(svCtx);
            DataSource cds = (DataSource) getJNDIResource("java:comp/env/jdbc/RsDatasource", svCtx);
      props.put(CaSystemConstants.INIT_RS_DATASOURCE, cds);
            return props;
    }
    protected SystemContextFactory getSystemContextFactory() {
            return new CaSystemContextFactory();
    }
    /**
     * Controlla la presenza dei parametri obbligatori
     */
    protected void checkConfig(Map props, ServletContext svCtx) throws ApsSystemException{
            super.checkConfig(props, svCtx);
            DataSource cds = (DataSource) props.get(CaSystemConstants.INIT_RS_DATASOURCE);
      checkResource(cds, "risorsa 'RS datasource' su JNDI (java:comp/env/jdbc/RSDatasource)",SERVER_XML);

    }
}
```

jAPS e Spring - Gestione Db esterno(2)

Prima:

- Creare systemContext

```
public class RsSystemContext extends SystemContext {
    /**
     * Restituisce una connessione SQL relativa al datasource di RS
     */
    public Connection getCanileConnection() throws ApsSystemException {
        Connection conn = null;
        try {
            conn = _rsDataSource.getConnection();
        } catch (SQLException e) {
            logThrowable(e, this, "getRSConnection");
            throw new ApsSystemException("Errore in creazione connessione da RS datasource", e);
        }
        return conn;
    }
    /**
     * Metodo riservato alla factory.
     * Imposta il datasource del rs.
    */
    protected void setRsDataSource(DataSource rsDataSource) {
        this._rsDataSource = rsDataSource;
    }
    /**
     * DataSource per il rs
     */
    private DataSource _rsDataSource;
•   }
```

jAPS e Spring - Gestione Db esterno(3)

Prima:

•Creare systemContextFactory

```
public class RsSystemContextFactory extends SystemContextFactory {

    protected SystemContext getNewContextInstance(Map properties) {
            CaSystemContext ctx = new CaSystemContext();
            DataSource cds = (DataSource) properties.get(CaSystemConstants.INIT_RS_DATASOURCE);
            ((CaSystemContext) ctx).setRsDataSource(cds);
            return ctx;
    }
}
```

• Specificare nuova risorsa JNDI nel contesto webapp;

• Nel file web.xml specificare il nuovo listener;

• Fare cast del contesto nel servizio che intende utilizzarlo.

jAPS e Spring - Gestione Db esterno (2)

Ora:

•Creare bean per il datasource in configurazione xml

```
<bean id="mioDataSource"      class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
      <property name="driverClassName"><value>${jdbc.servDb.driverClassName}</value></property>
      <property name="url"><value>${jdbc.mioDb.url}</value></property>
      <property name="username"><value>${jdbc.mioDb.username}</value></property>
      <property name="password"><value>${jdbc.mioDb.password}</value></property>
</bean>
```

• Ed infine utilizzare il bean appena creato.

```
<bean id="comuniManager"
      class="it.mioprogetto.aps.system.services.comuni.ComuniManager"
      parent="abstractService">
      <property name="comuniDao">
            <bean
                  class="it.mioprogetto.aps.system.services.comuni.ComuniDAO">
                  <property name="dataSource" ref="mioDataSource" />
            </bean>
      </property>
</bean>
```

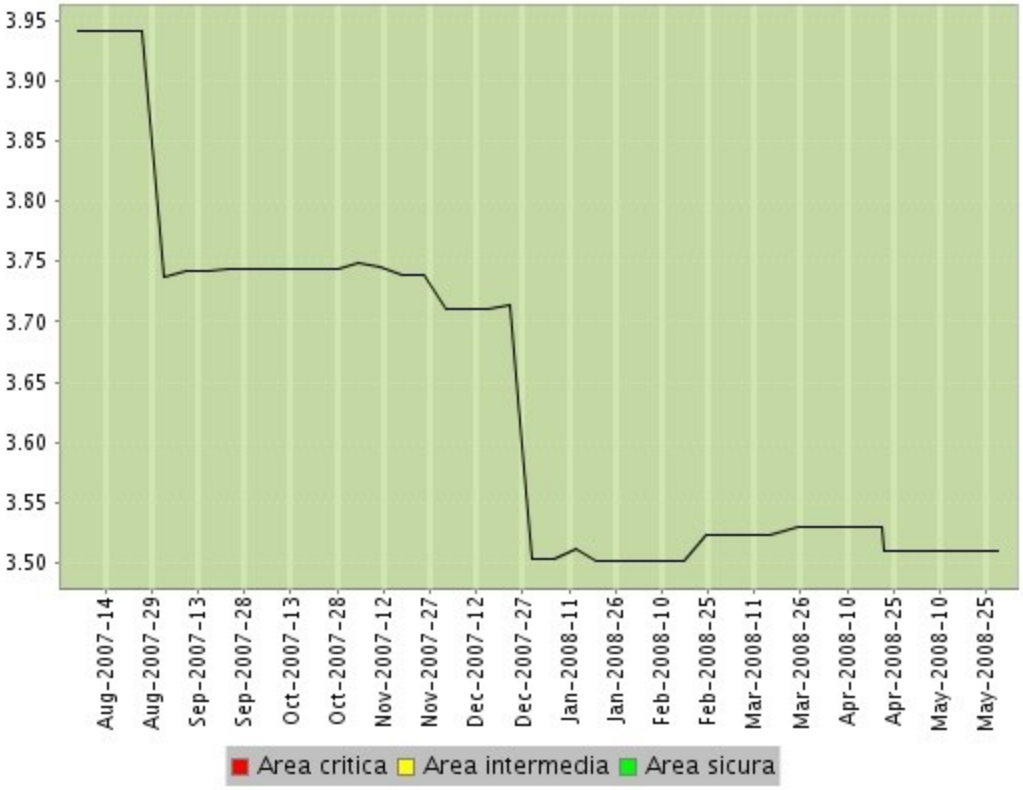
jAPS e Spring (Gestione utenti Prima e Dopo)

Ora vediamo un po' di codice per vedere la differenza tra il prima Spring e il dopo.

Switch su eclipse e jEdit

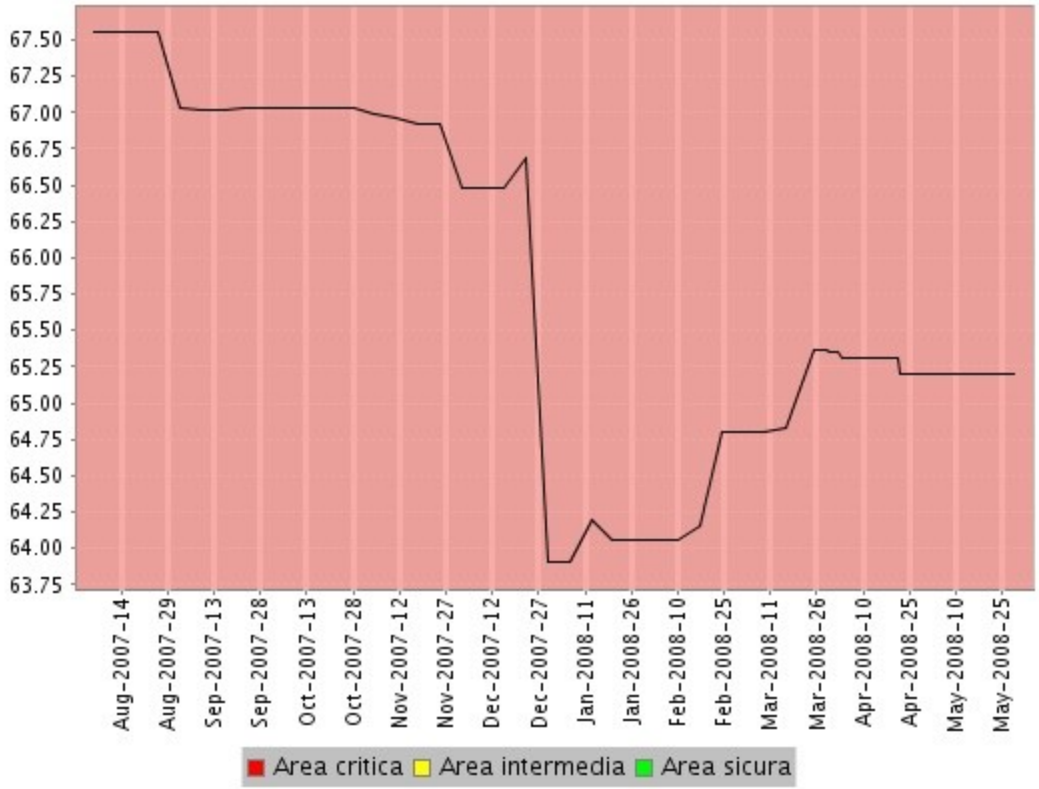
jAPS e Spring Un po' di statistiche (1)

CBO (accoppiamento di una classe con le altre classi di sistema)



jAPS e Spring Un po' di statistiche (1)

LOC (Linee di Codice)



Riferimenti

JAPS
www.japsportal.org
www.sourceforge.net/projects/japs

Spring Framework

www.springframework.org

Domande?

Grazie per l'attenzione.

rbonazzo@gmail.com