

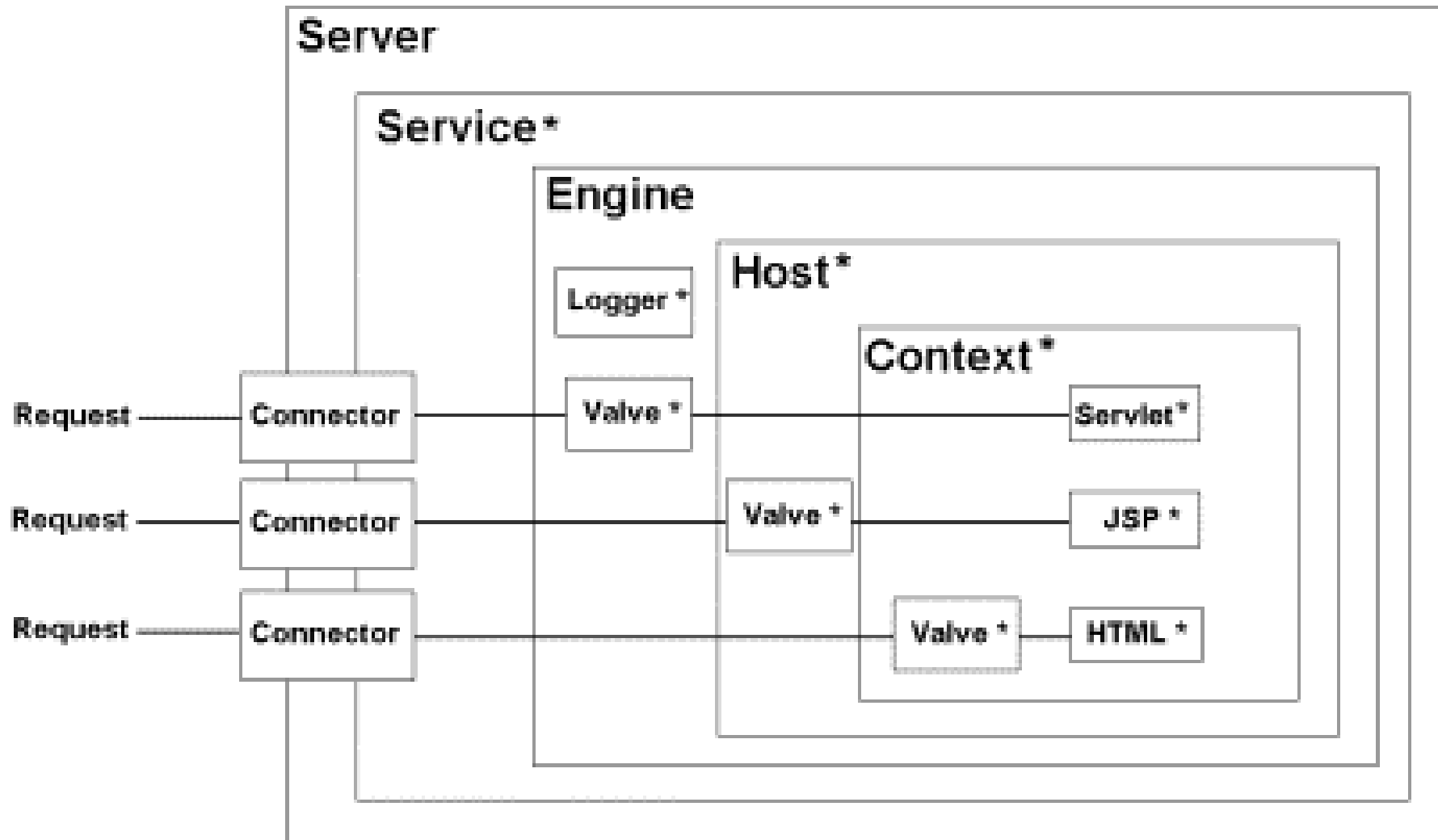


Jakarta Tomcat

Bartosz Walter

<Bartek.Walter@man.poznan.pl>

Architecture



Architecture

- **Top-level components**
 - exist at the top of the configuration hierarchy
- **Connectors**
 - connect the servlet container to the Web browser making requests
- **Container components**
 - contain a collection of other components
 - receive the requests from the top-level components, deal with the request process and return the response to the component that sent it to them.
- **Nested components**
 - can reside in containers, but cannot contain other components

Top-level components

- **Server**
 - single server resides within a single JVM
 - usually not implemented by users
- **Service**
 - groups an Engine component with its Connectors
- **Engine**
 - contained within a single Service
 - represents a request processing pipeline for a specific Service; examines HTTP headers and determines which host the request is designated for
 - may be used for container clustering

Connector components

- **Connector**
 - connects clients to webapps
 - each connector is assigned a unique TCP port
 - HTTP Connector and AJP Connector

Container components

- **Host**
 - represents a virtual host within an engine
- **Context**
 - the lowest-level container
 - represents a single web application

Nested components

- **Global resources**
 - can be only nested within a **Server** component
 - configures **JNDI** resources available to all components
- **Loader**
 - can be only nested within a **Context** component
 - specifies the webapp class loader
- **Logger**
 - reports on internal state of its parent container
 - containers inherit their **Loggers** unless overridden

Nested components

- **Realm**
 - manages A&A for an engine
 - typical implementations include database, LDAP, textfiles
- **Resources**
 - specifies static resources used by a webapp
- **Valve**
 - intercepts requests and processes them before they reach their destination

Context

- **configuration**
 - nested within Host in server.xml
 - individual [contextname].xml file in `$CATALINA_HOME/conf/[enginename]/[hostname]/`
 - META-INF/context.xml within WAR file
- **web application representation**
 - WAR file
 - corresponding directory containing unpacked content
- **servlet selection**
 - longest mapping in /WEB-INF/web.xml
- **unique context path**
- **default context**

Context example

```
<?xml version='1.0' encoding='utf-8'?>
```

```
<Context docBase="C:\myapp" path="/mycontext"  
  reloadable="true" >
```

```
<Logger className="org.apache.catalina.logger.FileLogger"  
  prefix="localhost_admin_log." suffix=".txt"  
  timestamp="true"/>
```

```
<Resource name="jdbc/mydb" type="javax.sql.DataSource"/>
```

```
<ResourceLink global="UserDatabase" name="users"  
  type="org.apache.catalina.UserDatabase"/>
```

```
</Context>
```

Web application layout

- ***.html, *.jsp, etc.** - The HTML and JSP pages, along with other files that must be visible to the client browser (such as JavaScript, stylesheet files, and images) for your application.
- **/WEB-INF/web.xml** - This is an XML file describing the servlets and other components that make up your application, along with any initialization parameters and container-managed security constraints that you want the server to enforce for you.
- **/WEB-INF/classes/** - This directory contains any Java class files (and associated resources) required for your application, including both servlet and non-servlet classes, that are not combined into JAR files. If your classes are organized into Java packages, you must reflect this in the directory hierarchy under `/WEB-INF/classes/`.
- **/WEB-INF/lib/** - This directory contains JAR files that contain Java class files (and associated resources) required for your application, such as third party class libraries or JDBC drivers.

Webapp deployment

- **local**

- [contextname].xml files and unpacked applications
- .war files

- **remote**

- manager app <http://localhost:8080/manager/html/>
- admin app <http://localhost:8080/admin>

- **Ant tasks**

Tomcat classloaders

- **common**
 - visible both to server and its webapps
 - `$CATALINA_HOME/common/{classes,endorsed,lib}`
- **server**
 - visible only to the server internal classes
 - `$CATALINA_HOME/server/{classes,lib}`
- **shared**
 - visible to all webapps
 - `$CATALINA_HOME/shared/{classes,lib}`
- **context**
 - visible to a single webapp only
 - `$CONTEXT_ROOT/WEB-INF/{classes,lib}`

Realm example

```
<Realm className="org.apache.catalina.realm.JDBCRealm"  
  debug="99"  
  driverName="oracle.jdbc.driver.OracleDriver"  
  connectionURL="jdbc:oracle:thin:@localhost:1521:orcl"  
  connectionName= "scott"  
  connectionPassword= "tiger"  
  userTable="users"  
  userNameCol="username"  
  userCredCol="password"  
  userRoleTable="user_roles"  
  roleNameCol="rolename" />
```

server.xml

Form-based authentication

- **j_username**
- **j_password**
- **j_security_check**

```
<login-config>  
  <auth-method>FORM</auth-method>  
  <form-login-config>  
    <form-login-page>/admin/main/login.html</form-login-page>  
    <form-error-page>/admin/main/login_failed.html</form-error-page>  
  </form-login-config>  
</login-config>
```

<http://www.onjava.com/pub/a/onjava/2002/06/12/form.html>

Connectors

- **Coyote**
 - default HTTP/1.1 connector
- **jk2**
 - handles AJP v.1.3 protocol
 - connects Tomcat to other HTTP servers (Apache, IIS, iPlanet, Domino)

Configuring DataSource in Tomcat

```
<Resource name="jdbc/TestDB" auth="Container"
  type="javax.sql.DataSource" />
<ResourceParams name="jdbc/TestDB">
  <parameter>
    <name>factory</name>
    <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
  </parameter>
  <parameter>
    <name>url</name>
    <value>jdbc:mysql://localhost:3306/javatest?autoReconnect=true</value>
  </parameter>
</ResourceParams>
```

server.xml

Configuring DataSource in Tomcat (cont.)

```
<resource-ref>  
  <description>DB Connection</description>  
  <res-ref-name>jdbc/TestDB</res-ref-name>  
  <res-type>javax.sql.DataSource</res-type>  
  <res-auth>Container</res-auth>  
</resource-ref>
```

web.xml

Configuring DataSource in Tomcat (cont.)

```
<resource-ref>  
  <description>DB Connection</description>  
  <res-ref-name>jdbc/TestDB</res-ref-name>  
  <res-type>javax.sql.DataSource</res-type>  
  <res-auth>Container</res-auth>  
</resource-ref>
```

web.xml

Configuring DataSource in Tomcat (cont.)

```
public void init() {
    try {
        Context ctx = new InitialContext();
        if (ctx == null)
            throw new Exception("Boom - No Context");
        DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/TestDB");
        if (ds != null) {
            Connection conn = ds.getConnection();
            if (conn != null) {
                foo = "Got Connection " + conn.toString();
            }
            conn.close();
        }
    } catch (Exception e) { e.printStackTrace(); }
}
```

MyServlet.java

Q & A

