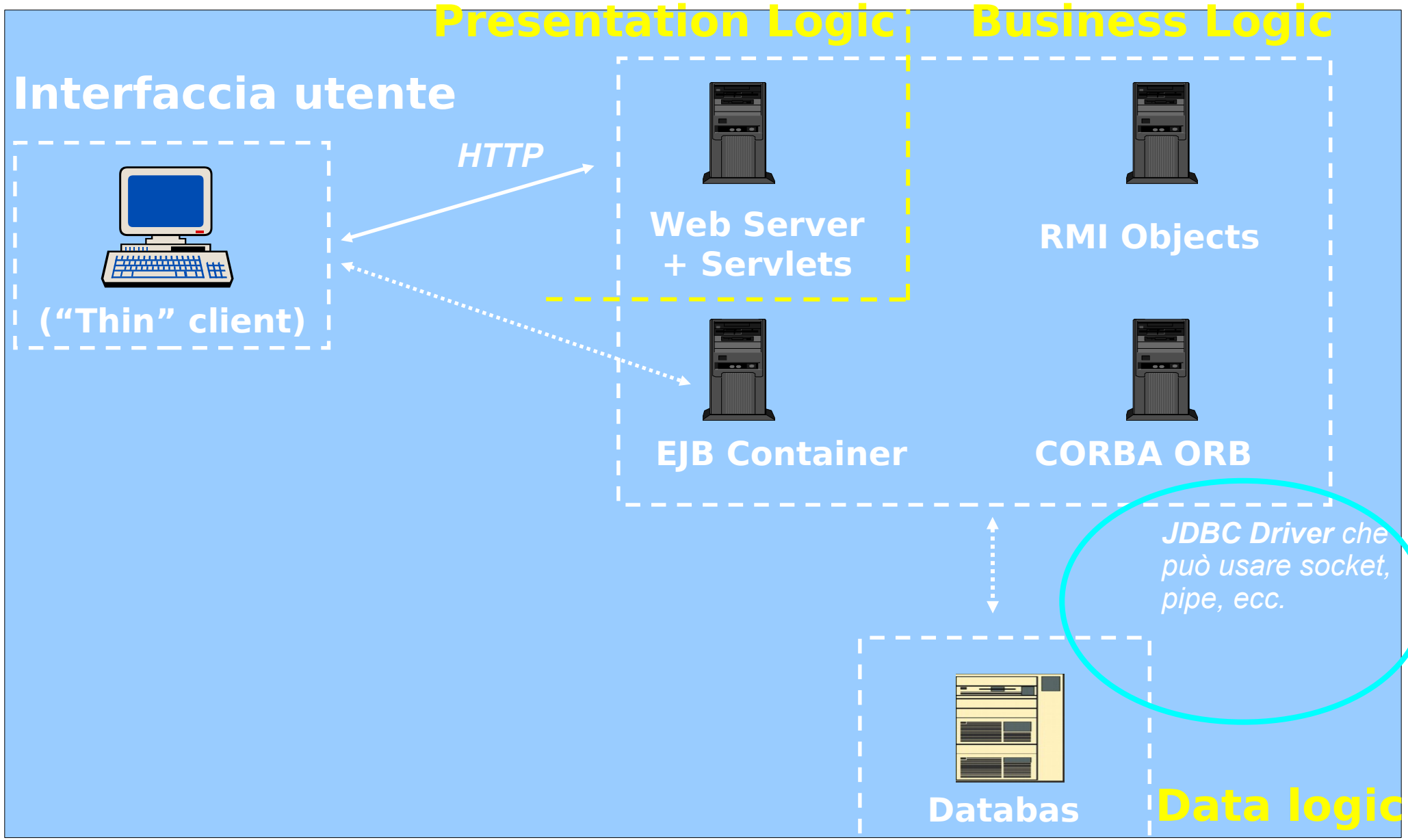
PARTE I
WEB APPLICATION SERVER TOMCAT

Modulo 1
Overview di Tomcat

Tomcat

- **Supporto per tecnologia Java e Web**
- **Sviluppato dalla Apache Software foundation**
- **Web server scritto interamente in Java**
- **Gestione di servlet e JSP**
- **Versione attualmente più diffusa: 7.0**
- **Versione più recente: 9.0**

Architettura n-tier basata su J2EE



Tomcat

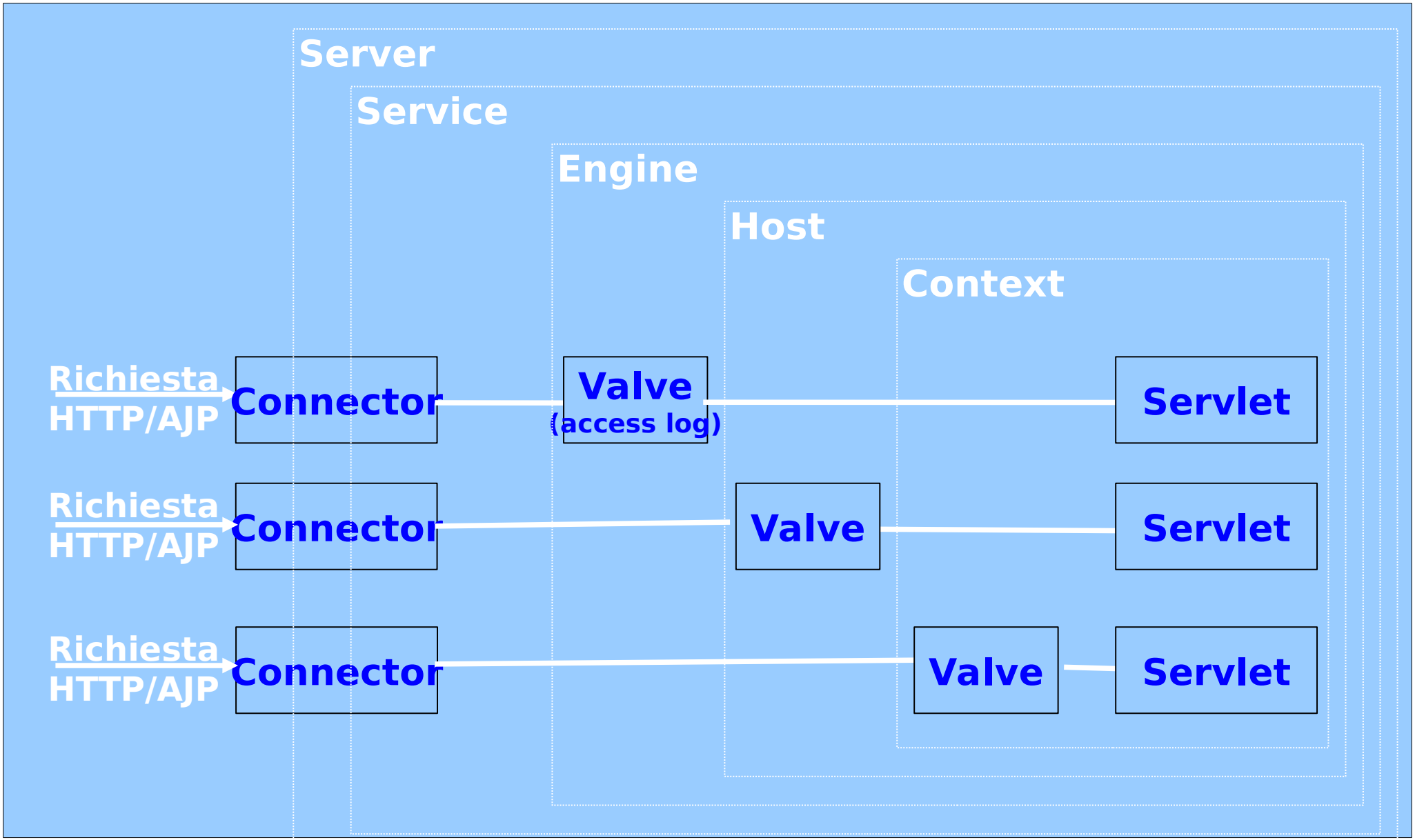
Limiti

- Prestazioni limitate rispetto ad Apache
- Fino alla versione 4 le prestazioni sono state un problema
- La versione 5 è stata ottimizzata rispetto alle versioni precedenti, ma presenta un alto consumo di memoria
- Nella versioni ≥ 6 il problema è ridotto

Soluzione per avere ottime prestazioni

- Apache o Nginx serve richieste statiche
- Uso di Apache o Nginx come load balancer
- Tomcat serve solo JSP
- Uso di un modulo di Apache come intermediario
- Soluzione simile per Nginx

Architettura di Tomcat



Terminologia

- **Server: un container**
- **Service: collega diversi connector ad un engine**
- **Engine: ciò che consente il servizio di una richiesta**
- **Host: Associa un nome simbolico ad un server Tomcat. Funzione analoga al virtual hosting**
- **Connector: un punto di comunicazione con l'esterno. Un connector supporta uno specifico protocollo, es. CoyoteConnector per HTTP, JK2 Connector per AJP**
- **Context: Una Web application**
- **Valve: Punto di passaggio dati con capacità di logging**

Context

- **Ogni insieme di URL appartiene ad una "zona" (in Tomcat, definita context)**
- **Ciascun context è controllato da un Web server**
- **Ogni context contiene oggetti validi sia per il loro ambito di azione sia per gli utenti che ad esso accedono**
- **I context non vengono generati sulla base di chiamate, ma pre-esistono dal momento in cui Tomcat viene attivato**
- **Come un server, i context attendono di rispondere alle richieste degli utenti senza mai arrestarsi**

Richiesta utente

- **Al momento della richiesta (URL tramite protocollo HTTP), viene creata l'istanza di un oggetto chiamato Request (o Req)**
- **Questo input porta il Servlet Engine a istanziare un oggetto chiamato Response (o Res) come reazione alla richiesta del client**
- **Req e Res sono dirottati verso il context contenente le risorse utili a concludere la transazione in atto**
- **Oltre a Req e Res, Tomcat istanzia un ulteriore oggetto riferito al Contenitore di Sessione, con il compito di identificare univocamente il richiedente e di controllare che la sua Sessione sia "aperta"**

Sessione utente

- **Ad ogni utente viene affidato dal Web server un container personale detto session**
- **Il container è caratterizzato dalle risorse a cui quel determinato utente ha la possibilità di accedere**
- **All'interno del container, e per tutta la durata del collegamento a Tomcat, l'utente può effettuare richieste sotto forma di input da soddisfare tramite risposte generate dinamicamente in output**
- **Le richieste possono essere considerate dei container particolari, in quanto esistono solo nel lasso di tempo che separa l'invio della domanda e la relativa risposta**

Gestione richiesta

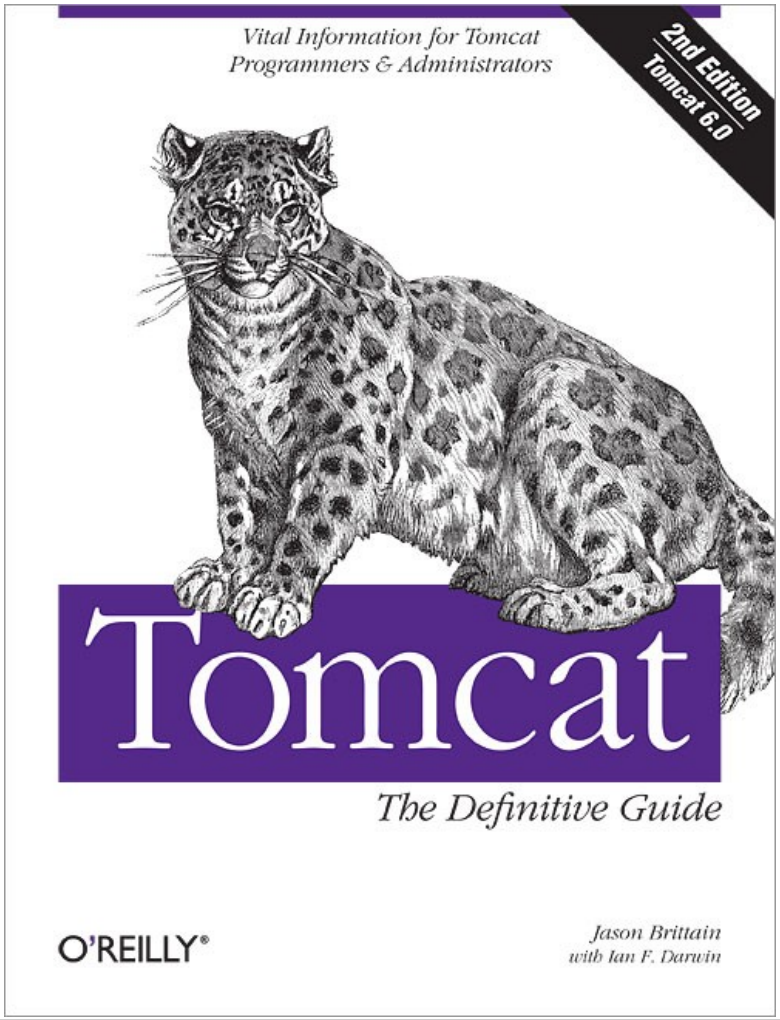
- **La vera gestione della richiesta è effettuata mediante le servlet**
- **Il context a cui appartengono le risorse necessarie per la soddisfazione della richiesta dovrà indirizzare queste ultime alla/e applicazione/i Java necessaria/e al completamento della transazione**

Gestione richiesta

- **Nel momento in cui è generato l'output vi sono tre esiti possibili:**
  - La Servlet esaudisce la richiesta e lascia a Tomcat il compito di consegnare l'output al client (per esempio, in forma di HTML)
  - La risorsa necessaria alla soddisfazione della richiesta non è presente nel context utilizzato, e dovrà quindi essere ricercata in un ulteriore context esterno. Questo compito è affidato a Tomcat che avvia una nuova procedura di input/output
  - La richiesta può essere soddisfatta all'interno del context, ma non tramite la risorsa richiesta. Dovrà quindi essere generato un Forward per il reindirizzamento verso la risorsa adeguata

Approfondimenti

Jason Brittain, Ian Darwin, “Tomcat: The Definitive Guide” (2nd edition), O’Reilly



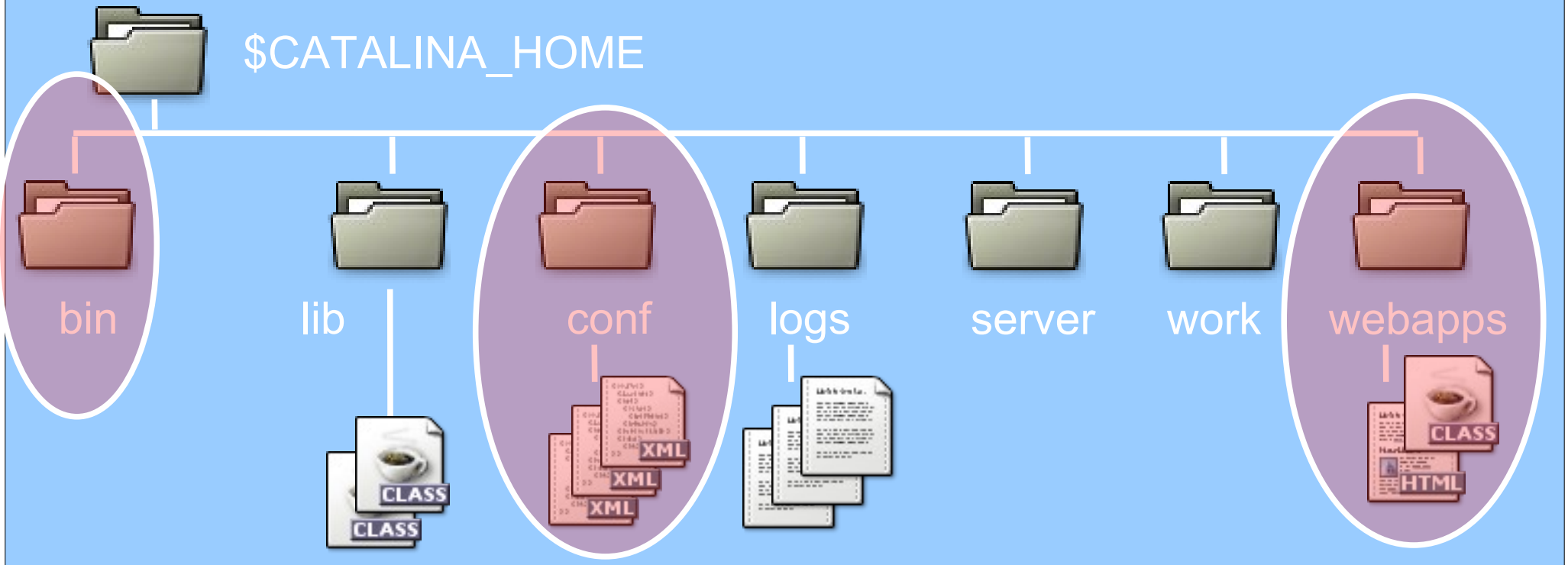
Modulo 2
Hands on Tomcat

Scaricare e installare Tomcat

- **Pacchetti software necessari:**
  - JDK
- **Tomcat**
  - http://tomcat.apache.org
- **Tomcat deve essere scompattato in una directory che sarà la base del server**
  - $CATALINA_HOME

Overview di Tomcat

- **Struttura delle directory più complessa rispetto ad Apache**



Overview di Tomcat

- **$CATALINA_HOME/bin: eseguibili per il funzionamento di tomcat**
  - startup.sh
  - shutdown.sh
- **$CATALINA_HOME/lib: classi per la JVM di jakarta/tomcat**
  - $CATALINA_HOME/lib/jasper.jar
  - $CATALINA_HOME/lib/servlet-api.jar
- **$CATALINA_HOME/conf: file di configuazione**
- **$CATALINA_HOME/webapps: pagine JSP e Servlet**

Configurazione di Tomcat

- **Molteplici files .xml**
  - server.xml (configurazione dei server, in particolare specifica e configura i connector)
  - context.xml (parametri per avere dati persistenti per le servlet)
  - tomcat-users.xml (informazioni per il controllo d'accesso)
  - web.xml (dati per la configurazione delle servlet)
- **Files .policy**
  - specificano configurazioni sulla sicurezza della JVM

Esempio di configurazione di connector

Configurazione nel file server.xml

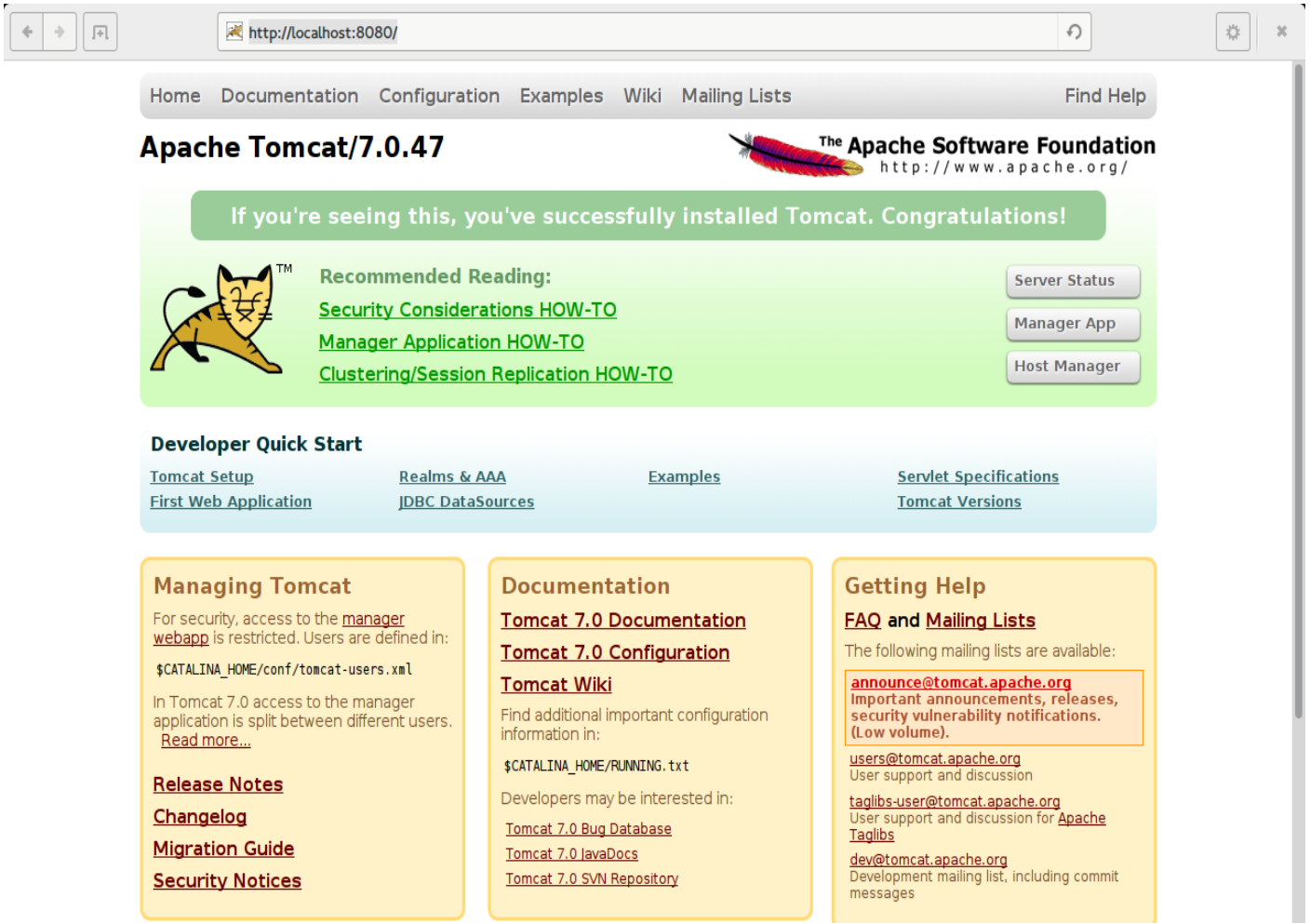
```
<Connector port="8080"
           protocol="HTTP/1.1"
               connectionTimeout="20000"
               RedirectPort="8443" />

<Connector port="8009"
               protocol="AJP/1.3"
               redirectPort="8443" />
```

Coyote connector

JK connector

Funzionamento

Di default Tomcat si pone in ascolto sulla porta 8080



Gestione utenti e ruoli in tomcat

- **Due concetti separati**
  - Utente → legato a credenziali di accesso
  - Ruolo → legato a cosa si può fare
- **Nota: il file con le credenziali contiene dati in chiaro**
  - Potenziale rischio di sicurezza
- **Mapping utenti/ruoli**
  - Un utente può avere uno o più ruoli
  - Un ruolo può essere assegnato a uno o più utenti

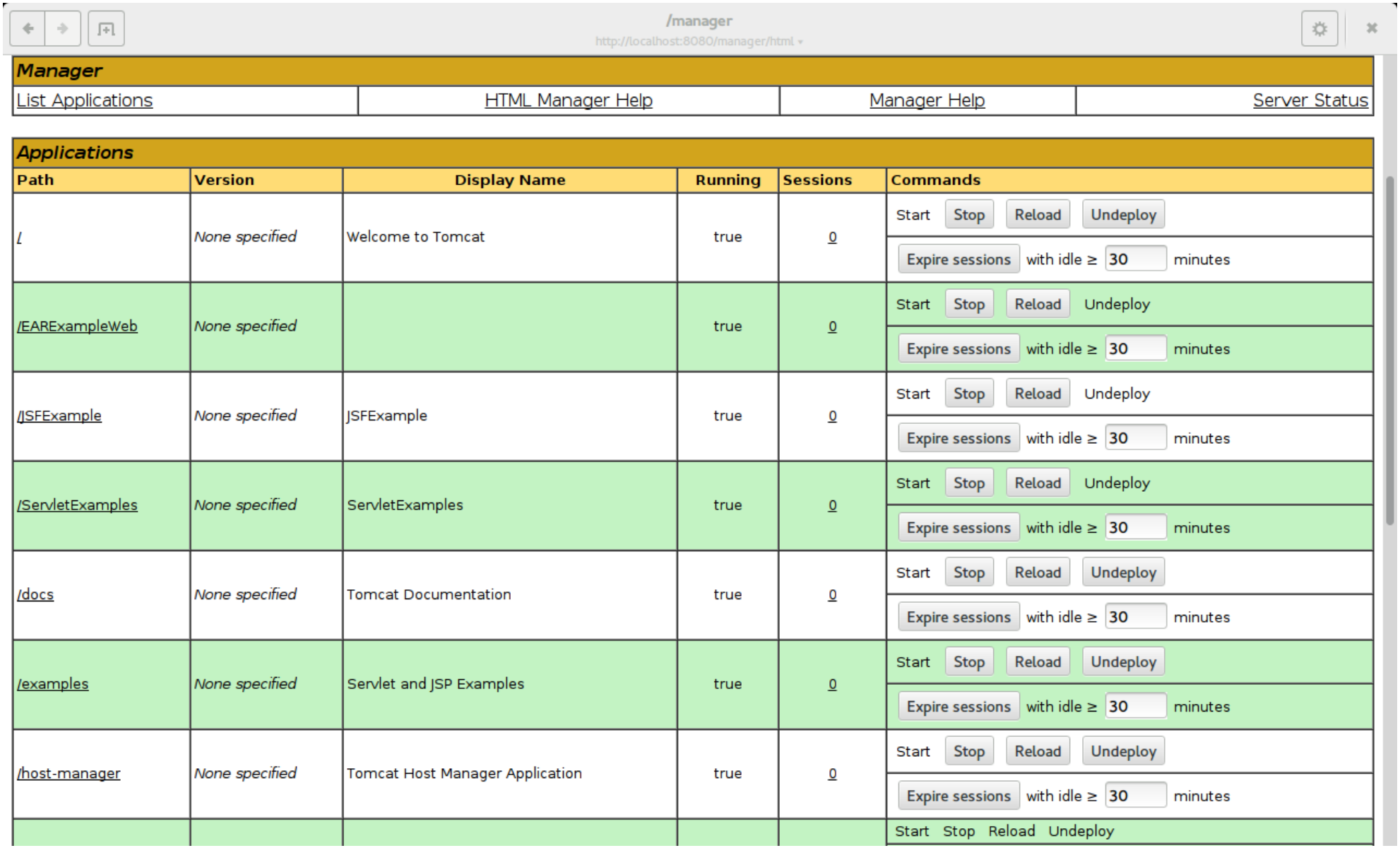
Gestione degli utenti e dei ruoli (v>6.0)

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="tomcat"/>
  <role rolename="manager-gui"/>
  <role rolename="manager-status"/>
  <role rolename="manager-script"/>
  <role rolename="manager-jmx"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="admin" password="tomcat" roles="tomcat,manager-
gui,manager-status,manager-script,manager-jmx"/>
</tomcat-users>
```

Applicazioni di default in Tomcat

- **Applicazioni che richiedono ruolo manager-gui**
- **Manager delle applicazioni**
  - Mostra stato Web application installate
  - Consente di installare/rimuovere applicazioni
  - Fornisce informazioni su sessioni utente
- **Stato del server**
  - JVM
  - Connector
  - ...

Manager applicazioni



Stato del server

/manager

http://localhost:8080/manager/status

Manager			
List Applications	HTML Manager Help	Manager Help	Complete Server Status

Server Information							
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.47	1.6.0_26-b03	Sun Microsystems Inc.	Linux	3.13-1-amd64	amd64	luna	127.0.0.1

JVM

Free memory: 110.10 MB Total memory: 143.31 MB Max memory: 880.00 MB

Memory Pool	Type	Initial	Total	Maximum	Used
PS Eden Space	Heap memory	15.43 MB	96.00 MB	316.12 MB	15.11 MB (4%)
PS Old Gen	Heap memory	41.25 MB	41.25 MB	660.00 MB	12.04 MB (1%)
PS Survivor Space	Heap memory	2.56 MB	6.06 MB	6.06 MB	6.03 MB (99%)
Code Cache	Non-heap memory	2.43 MB	2.43 MB	48.00 MB	1.48 MB (3%)
PS Perm Gen	Non-heap memory	20.75 MB	24.68 MB	82.00 MB	24.59 MB (29%)

"ajp-bio-8009"

Max threads: 200 Current thread count: 0 Current thread busy: 0
Max processing time: 0 ms Processing time: 0.0 s Request count: 0 Error count: 0 Bytes received: 0.00 MB Bytes sent: 0.00 MB

Stage	Time	B Sent	B Recv	Client	VHost	Request

P: Parse and prepare request S: Service F: Finishing R: Ready K: Keepalive

"http-bio-8080"

Max threads: 200 Current thread count: 10 Current thread busy: 1
Max processing time: 291 ms Processing time: 1.374 s Request count: 13 Error count: 1 Bytes received: 0.00 MB Bytes sent: 0.05 MB

Stage	Time	B Sent	B Recv	Client	VHost	Request
R	?	?	?	?	?	?
R	?	?	?	?	?	?

Due parole sulla memoria in Java

- **Gestione dimanica della memoria**
- **Uso di garbage collector**
- **Garbage collector → molto oneroso**
  - Major collection
  - Minor collection
- **Gestione della memoria basata su età degli oggetti**

Due parole sulla memoria in Java

- **Gestione della memoria basata su età degli oggetti**
- **→ Gestione generazionale della memoria [da documentazione Oracle]**
  - Eden Space (heap): The pool from which memory is initially allocated for most objects.
  - Survivor Space (heap): The pool containing objects that have survived the garbage collection of the Eden space.
  - Tenured Generation (heap): The pool containing objects that have existed for some time in the survivor space.

Due parole sulla memoria in Java

- **→ Gestione generazionale della memoria**
  - Permanent Generation (non-heap): The pool containing all the reflective data of the virtual machine itself, such as class and method objects. With Java VMs that use class data sharing, this generation is divided into read-only and read-write areas.
  - Code Cache (non-heap): The HotSpot Java VM also includes a code cache, containing memory that is used for compilation and storage of native code.
- **Un caos terribile da configurare**
- **Spesso dal punto di vista prestazionale è un elemento molto critico**

Modulo 3
Logging

Il logging

- **I logfile permettono di monitorare gli accessi ad un server Web**
  - Le informazioni che possono essere memorizzate nel logfile sono quelle che viaggiano all'interno dei messaggi di richiesta e risposta che il server scambia con il client usato dagli utenti
  - Generalmente i server Web permettono di definire quali campi dei messaggi devono essere memorizzati generando così dei logfile "custom" in modo da soddisfare al meglio le necessità dell'amministratore del sito Web

Logging in Tomcat

- **Elemento chiave del logging: Valve**
- **Consente di definire:**
  - Quale file di log vogliamo scrivere
  - Che informazioni conservare nel file
  - Quale classe si occupa del logging
- **Struttura del file di log:**
  - Una riga per ogni richiesta
  - Le righe sono composte da record

Esempio di configurazione di una Valve

```
<Valve

className="org.apache.catalina.valves.AccessLogValve"
    directory="${catalina.base}/logs"
    prefix="access_log"
    fileDateFormat="yyyy-MM-dd.HH"
    suffix=".log"
    pattern="%t %H cookie:%{SESSIONID}c
        request:%{SESSIONID}r  %m %U %s %q %r"
/>
```

Alcune funzioni supportate da Valve

- %a - Remote IP address
- %A - Local IP address
- %B - Bytes sent, excluding HTTP headers
- %h - Remote host name (or IP address if resolveHosts is false)
- %H - Request protocol
- %l - Remote logical username from identd (always returns '-')
- %m - Request method (GET, POST, etc.)
- %p - Local port on which this request was received
- %q - Query string (prepended with a '?' if it exists)
- %r - First line of the request (method and request URI)
- %s - HTTP status code of the response
- %S - User session ID
- %t - Date and time, in Common Log Format
- %u - Remote user that was authenticated (if any), else '-'
- %U - Requested URL path
- %D - Time taken to process the request, in millis

Dati estraibili da un log file

- **Orari di maggiore traffico**
- **Tipologia degli utenti (browser utilizzato, provenienza geografica)**
- **Pagine più popolari**
- **Quali siti fanno riferimento al proprio**

- **Attenzione: la presenza di proxy intermedi tra client e server Web può falsare i risultati**

Utilità dei log file

- **Monitorare lo stato del server**
- **Capacity planning**
- **Billing**
- **Attack detection** 

Esempio

127.0.0.1 - - [14/Oct/2002:18:00:16 +0200] "GET /icons/apache_pb.gif HTTP/1.1" 200 2326 "http://localhost/" "Mozilla/5.0 Galeon/1.2.6 (X11; Linux i686; U;) Gecko/20020913 Debian/1.2.6-2"

**211.97.159.184 - - [14/Oct/2002:16:06:44 +0200] "GET /default.ida?
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0" 400 341 "-" "-"**

Log analyzer

- **Esistono diversi strumenti che permettono di analizzare file di log. Alcuni sono generici, altri si focalizzano sui formati utilizzati dai più diffusi web server.**
- **Alcuni esempi sono:**
  - http://awstats.sourceforge.net/
  - http://www.mrunix.net/webalizer/
- **Per esigenze limitate è possibile anche importare i file di log in programmi per la gestione di fogli di calcolo (es. Calc di OpenOffice)**