

RECAP JDBC:

PER IMPLEMENTARE QUALSIASI OPERAZIONE DI CRUD OCCORRE PRIMA APRIRE UNA CONNESSIONE AL DATABASE
APERTURA DELLA CONNESSIONE:

```
Connection connection = DriverManager.getConnection(url,username,password)
```

INSERIMENTO

PreparedStatement ps = **connection.prepareStatement** (stringa che contiene uno script sql di inserimento con i parametri anonimi)

SOSTITUZIONE DEI PARAMETRI ANONIMI TRAMITE I METODI DI SET INDICANDO CORRETTAMENTE GLI INDICI DEI PUNTI INTERROGATIVI

```
ps.executeUpdate();--→>>ESECUZIONE DELLO SCRIPT SQL COMPLETO
```

AGGIORNAMENTO

PreparedStatement ps = **connection.prepareStatement** (stringa che contiene uno script sql di aggiornamento con i parametri anonimi) SOSTITUZIONE DEI PARAMETRI ANONIMI TRAMITE I METODI DI SET INDICANDO CORRETTAMENTE GLI INDICI DEI PUNTI INTERROGATIVI (PLACEHOLDER)

```
ps.executeUpdate();--→>>ESECUZIONE DELLO SCRIPT SQL COMPLETO
```

CANCELLAZIONE

PreparedStatement ps = connection.prepareStatement(stringa che contiene uno script sql di cancellazione con i parametri anonimi)

SOSTITUZIONE DEI PARAMETRI ANONIMI TRAMITE I METODI DI SET INDICANDO CORRETTAMENTE GLI INDICI DEI PUNTI INTERROGATIVI (PLACEHOLDER)

```
ps.executeUpdate();--→>>ESECUZIONE DELLO SCRIPT SQL COMPLETO
```

LETTURA NON PARAMETRICA SENZA WHERE

Statement stmt = connection.createStatement();

ResultSet rs = stmt.executeQuery(script sql di lettura senza where)

ITERAZIONE DEL RUSELTSET

LETTURA PARAMETRICA (CON WHERE)

PreparedStatement ps = connection.prepareStatement(script sql di lettura con parametri anonimi)

SOSTITUZIONE DEI PARAMETRI ANONIMI TRAMITE I METODI DI SET INDICANDO CORRETTAMENTE GLI INDICI DEI PUNTI INTERROGATIVI

ResultSet rs = ps.executeQuery();

CHIUSURA DELLA CONNESSIONE

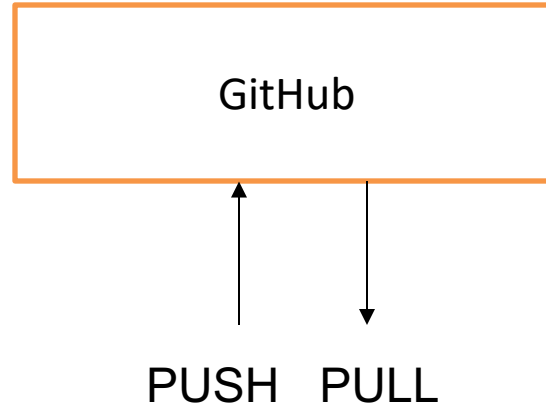
connection.close();

SVANTAGGI JDBC

- BOILERPLATE CODE
- GRANDE QUANTITA' DI CODICE
- NECESSITA' DELL'USO DEL LINGUAGGIO SQL IL CHE COMPORTA il MIX DI DUE LINGUAGGI IN UNA STESSA APPLICAZIONE (Java + SQL)->>>2 LIVELLI DI BUG FIXING
- IMPEDENCE MISMATCHING PROBLEM->>> PROBLEMA DI ADERIRE AD UNA INFRASTRUTTURA ESTERNA

GitHub è una soluzione in CLOUD che consente di archiviare il codice progettuale in maniera tale che sia condiviso/aggiornabile da tutti i componenti di un Team di sviluppo.

Per caricare/aggiornare il codice all'interno di un Repository GitHub occorre utilizzare il software GIT.



STEP PER PUSARE UN NUOVO BRANCH SU UN REPOSITORY GitHub REMOTO:

INIZIALIZZAZIONE DI UN REPOSITORY LOCALE

git init

SINCRONIZZAZIONE TRA REPOSITORY LOCALE E REPOSITORY REMOTO

git add .

CREAZIONE DI UN NUOVO BRANCH LOCALE

git checkout -b [branch name]

SINCRONIZZAZIONE DEL BRANCH LOCALE SUL REPOSITORY REMOTO

git branch

FASE PRECEDENTE AL COMMIT

git remote add origin [link github repository]

COMMIT (FASE PRECEDENTE AL PUSH)

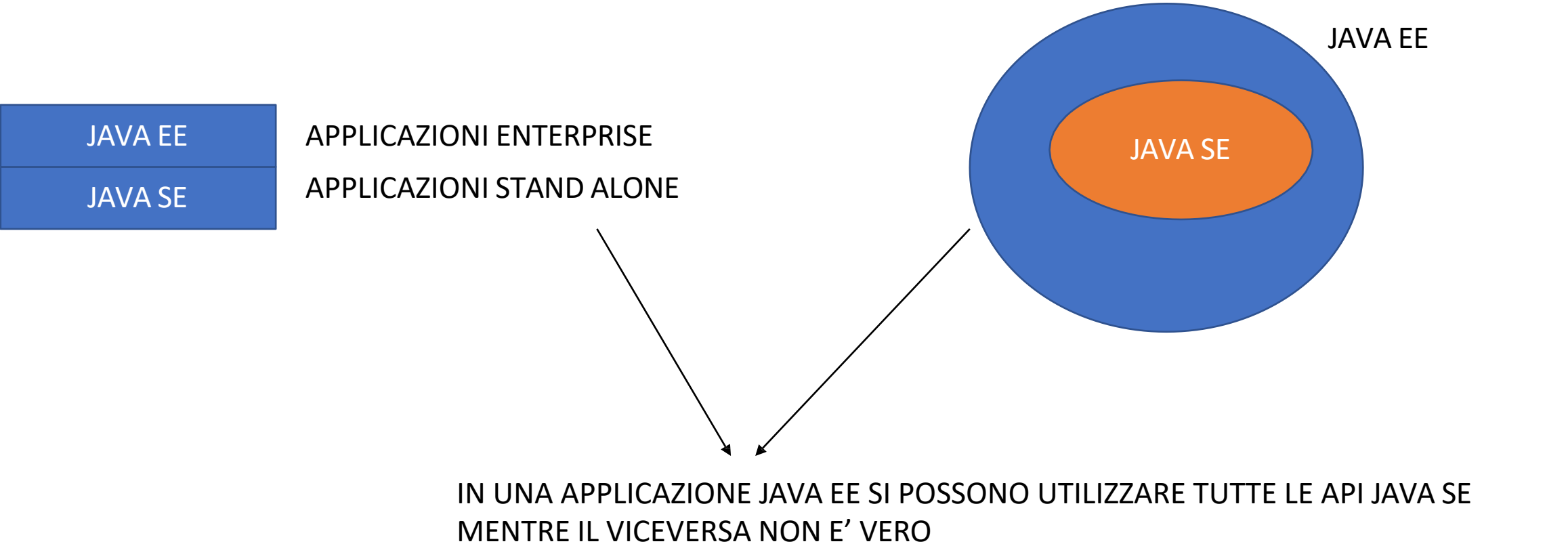
git commit -m "new branch first example"

PUSH

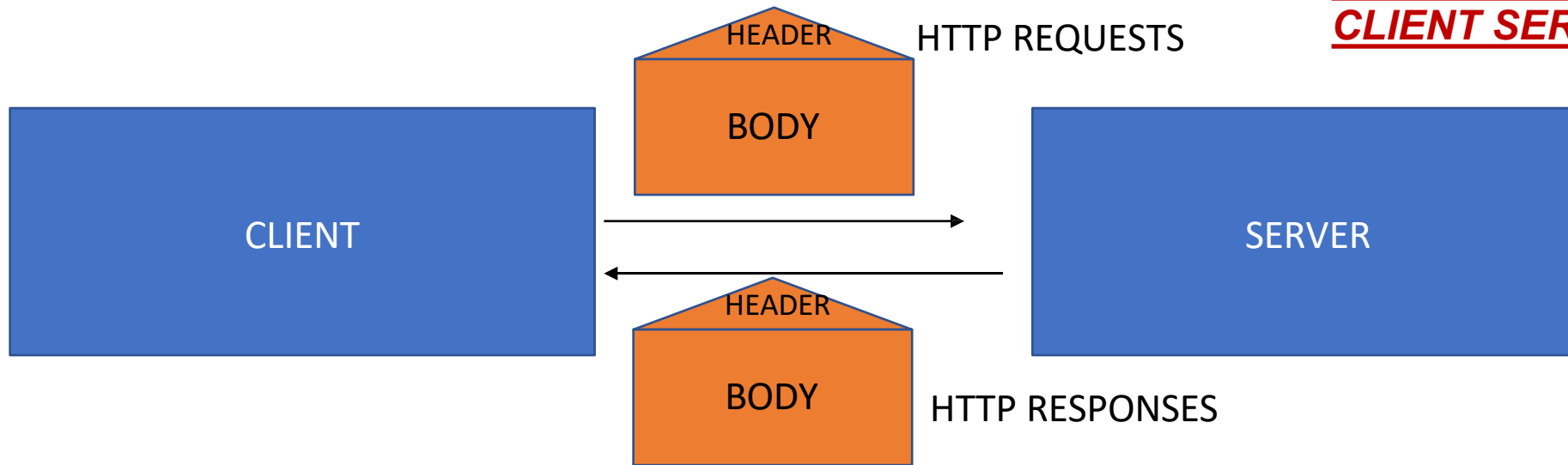
git push origin [branch name]

JAVA EE = INSIEME DI API CHE CONSENTONO L'IMPLEMENTAZIONE DI APPLICAZIONI ENTERPRISE, OVVERO APPLICAZIONI CHE POSSONO ESSERE ESEGUITE TRAMITE PROTOCOLLO HTTP ADERENDO AL MODELLO INFRASTRUTTURALE CLIENT-SERVER

LO STACK TECNOLOGICO JAVA EE DEVE ESSERE CONSIDERATO COME UN SUPERSET (PLUG-IN) DELLO STACK TECNOLOGICO JAVA SE



MODELLO INFRASTRUTTURALE CLIENT SERVER



UNA HTTP REQUEST DEVE AVERE UNA SPECIFICA FORMA (ANATOMIA)

BODY (CONTENUTO INFORMATIVO DELLA RICHIESTA)->>> VUOTO O PIENO

HEADER DELLA RICHIESTA HTTP (INFORMAZIONI UTILI AL DESTINATARIO OVVERO AL SERVER PER CAPIRE L'ORIGINE DELLA RICHIESTA)

- **BROWSER E RELATIVA VERSIONE UTILIZZATI DAL CLIENT**
- **IP DELLA MACCHINA CLIENT**
- **MEDIA TYPE UTILIZZATO PER LA RICHIESTA (FORMATO JSON)**
- **TIPOLOGIA DI RICHIESTA EFFETTUATA/VERBO HTTP**
- **URL PATTERN DELLA RICHIESTA = IDENTIFICATIVO UNIVOCO PER OGNI SINGOLA AZIONE LATO CLIENT ATTRAVERSO IL QUALE IL SERVER RIESCE AD IDENTIFICARE LA RICHIESTA E A DIFFERENZiarLE DALLE ALTRE**

PRINCIPALI VERBI HTTP: **GET,PUT,PATCH,POST,DELETE**

GET

- CONSENTE DI INSERIRE NEL BODY FINO AD UN MASSIMO DI 256 caratteri
- IL CONTENUTO DEL BODY VIAGGIA IN CHIARO VERSO IL SERVER OVVERO E' VISUALIZZABILE SULLA URL DEL BROWSER

POST

- CONSENTE DI INSERIRE NEL BODY UN NUMERO ILLIMITATO DI CARATTERI
- IL CONTENUTO DEL BODY VIAGGIA NASCOSTO VERSO IL SERVER OVVERO NON E' VISUALIZZABILE SULLA URL DEL BROWSER

ANTI PATTERN

USARE LA GET PER RICHIESTE HTTP CHE NECESSITINO DELL'INVIO DI INFORMAZIONI VERSO IL SERVER

BEST PRACTICE = USARE LA GET PER RICHIESTE HTTP CHE NON PRESUPPONGONO L'INVIO DI INFORMAZIONI DEL SERVER

PROCEDURA STANDARD

USARE LA POST PER RICHIESTE HTTP CHE NECESSITINO DELL'INVIO DI INFORMAZIONI VERSO IL SERVER

PRINCIPALI API/SPECIFICHE JAVA EE

- **SERVLET ---→>>> LA PRIMA COMPONENTE APPLICATIVA LATO SERVER IN GRADO DI GESTIRE LE RICHIESTE HTTP PROVENIENTI DA UN CLIENT E DI RESTITUIRE LE RISPOSTE HTTP INVIATE AL CLIENT, RILASCIATA SOTTO FORMA DI SLO (SPECIALIZED OBJECT)**
- **EJB (ENTERPRISE JAVA BEAN)-→>>IMPLEMENTAZIONE DELLA BUSINESS LOGIC ALL'INTERNO DI APPLICAZIONI WEB**
- **JPA (Java Persistence API)-→>>IMPLEMENTAZIONE DI OPERAZIONI DI CRUD SU DATABASE RELAZIONALI**
- **JSP (Java Servlet Page)-→>>> PAGINE WEB DINAMICHE UTILIZZANDO CODICE JAVA MISTO A CODICE HTML**

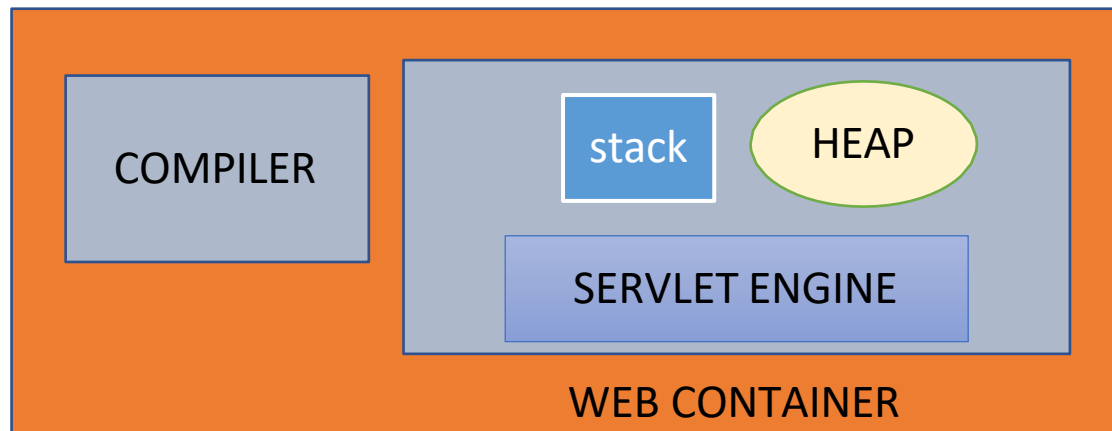
SLO = CLASSE CONCRETA NON ISTANZIABILE APPLICATIVAMENTE DALLO SVILUPPATORE TRAMITE L'UTILIZZO DELLA PAROLA CHIAVE NEW E RELATIVO COSTRUTTORE, MA CHE VIENE ISTANZIATA DA UNA COMPONENTE INFRASTRUTTURALE

OGNI SERVLET VIENE ISTANZIATA DA UNA COMPONENTE INFRASTRUTTURALE CHIAMATA WEB CONTAINER

LA SERVLET E' PERTANTO DA CONSIDERARSI UNA SLO IL CUI CICLO DI VITA E' GESTITO DAL WEB CONTAINER

ESISTONO DUE PIATTAFORME DI RIFERIMENTO PER LA COMPILAZIONE ED ESECUZIONE DI UNA APPLICAZIONE JAVA EE (LA SCELTA SE UTILIZZARE L'UNA O L'ALTRA DIPENDE DALLE SPECIFICHE JAVA EE USATE)

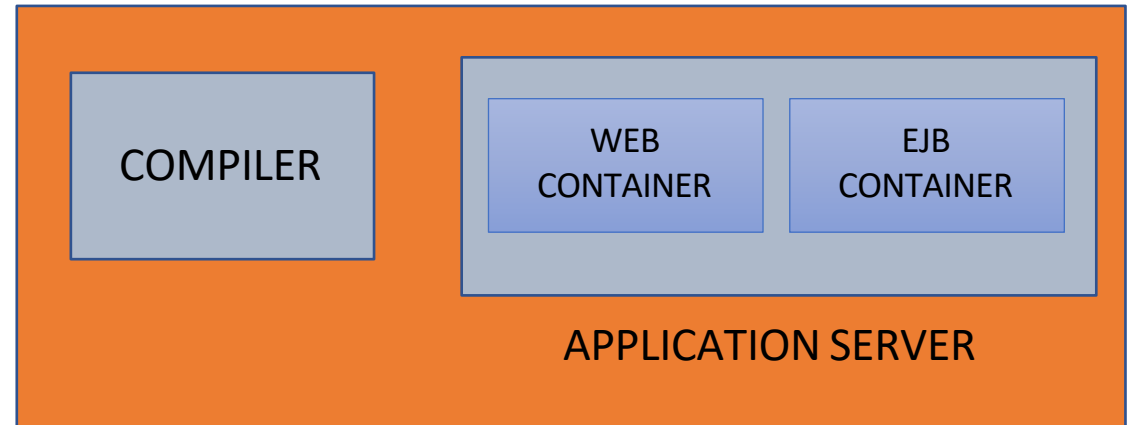
PIATTAFORMA JAVA EE



UN WEB CONTAINER E' UNA JVM «COMPLESSA» CHE CONTIENE AL SUO INTERNO STACK, HEAP E SERVLET ENGINE. IL SERVLET ENGINE SERVE PER GESTIRE IL CICLO DI VITA DELLE SERVLET E DELLE JSP. VA BENE QUINDI UTILIZZARE QUESTA PIATTAFORMA NEL CASO IN CUI LE API JAVA EE UTILIZZATE NEL PROGETTO SIANO JSP E SERVLET.

IL WEB CONTAINER PIU' UTILIZZATO E' TOMCAT

PIATTAFORMA JAVA EE



UN APPLICATION SERVER E' UNA JVM «COMPLESSA» CHE CONTIENE AL SUO INTERNO WEB CONTAINER E EJB CONTAINER. L'EJB CONTAINER SERVE PER GESTIRE IL CICLO DI VITA DELLE EJB. VA BENE QUINDI UTILIZZARE QUESTA PIATTAFORMA NEL CASO IN CUI LE API JAVA EE UTILIZZATE NEL PROGETTO SIANO JSP, SERVLET ED EJB.

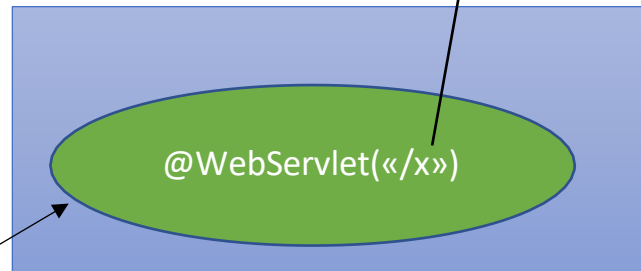
ESEMPI DI APPLICATION SERVER : GLASSFISH, JBOSS, ORACLE WEBLOGIC, IBM WEBSHERE APPLICATION SERVER, TOMCAT EE

LA PIATTAFORMA JAVA EE, INDIPENDENTEMENTE DA QUALE SI SCELGA DEVE ESSERE INSTALLATA SU DI UN SERVER

SERVER

URI

AD OGNI SERVLET PUO' ESSERE ASSOCIATA UNA URI
(UNIFORM RESOURCE IDENTIFIER) CHE DEVE MATCHARE
L'URL PATTERN DI UNA RICHIESTA HTTP, ATTRAVERSO UNA
ANNOTATION @WebServlet



SERVLET ENGINE

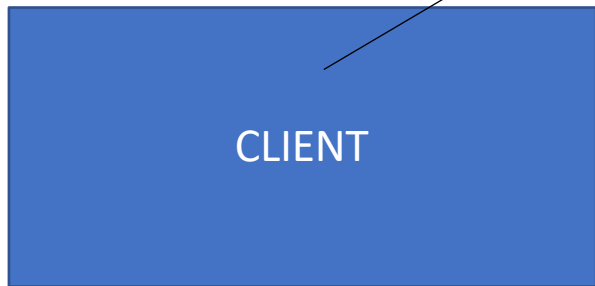
OGNI VOLTA CHE ARRIVA UNA RICHIESTA DAL CLIENT IL WEB CONTAINER
ESAMINA L'URL PATTERN CONTENUTO NELL'HEADER DELLA RICHIESTA E
VERIFICA SE ESISTE UNA SERVLET CHE, ALL'INTERNO DELL'ANNOTATION
@WebServlet, HA UNA URI CORRISPONDENTE ALL'URL PATTERN, E IN CASO
AFFERMATIVO ISTANZIA LA SERVLET E NE ESEGUI IL CODICE, ALTRIMENTI
RESTITUISCE UN ERRORE

url pattern = x

url pattern

HTTP REQUEST

CLIENT



RECAP TIPOLOGIA COMPONENTI APPLICATIVE (JAVA SE/JAVA EE)

IN JAVA ESISTONO 3 TIPOLOGIE DI COMPONENTI APPLICATIVE

- CLASSE CONCRETA (OOP)

UNA CLASSE CONCRETA PUO' ESSERE DI DUE SOTTOTIPOLOGIE:

- POJO (PLAIN OLD JAVA OBJECT) →>>UNA CLASSE CONCRETA CHE PUO' ESSERE ISTANZIATA APPLICATIVAMENTE DALLO SVILUPPATORE E IL CICLO DI VITA DEGLI OGGETTI VIENE GESTITO DALL'HEAP
- SLO (SPECIALIZED OBJECT)--→>> UNA CLASSE CONCRETA CHE NON PUO' ESSERE ISTANZIATA APPLICATIVAMENTE DALLO SVILUPPATORE E IL CICLO DI VITA DEGLI OGGETTI VIENE GESTITO DA UN CONTAINER DIVERSO DALL'HEAP

LE SERVLET SONO SLO IL CUI CICLO DI VITA VIENE GESTITO DAL WEB CONTAINER

LE EJB SONO SLO IL CUI CILO DI VITA VIENE GESTITO DALL'EJB CONTAINER

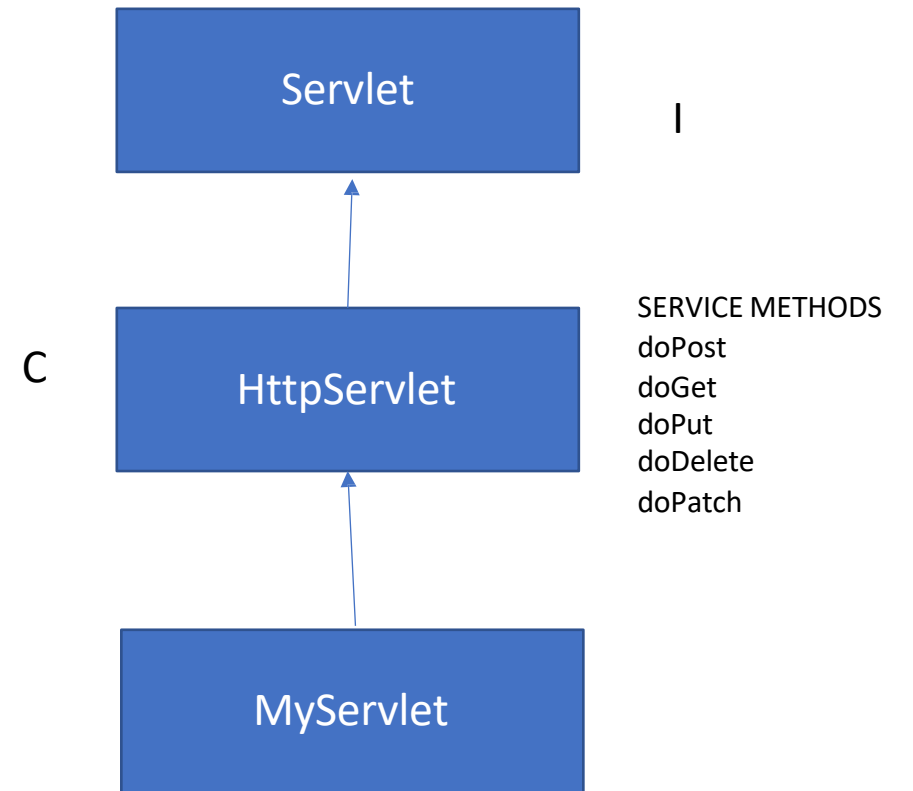
PER JAVA SE NON ESISTONO SLO MA SOLO POJO

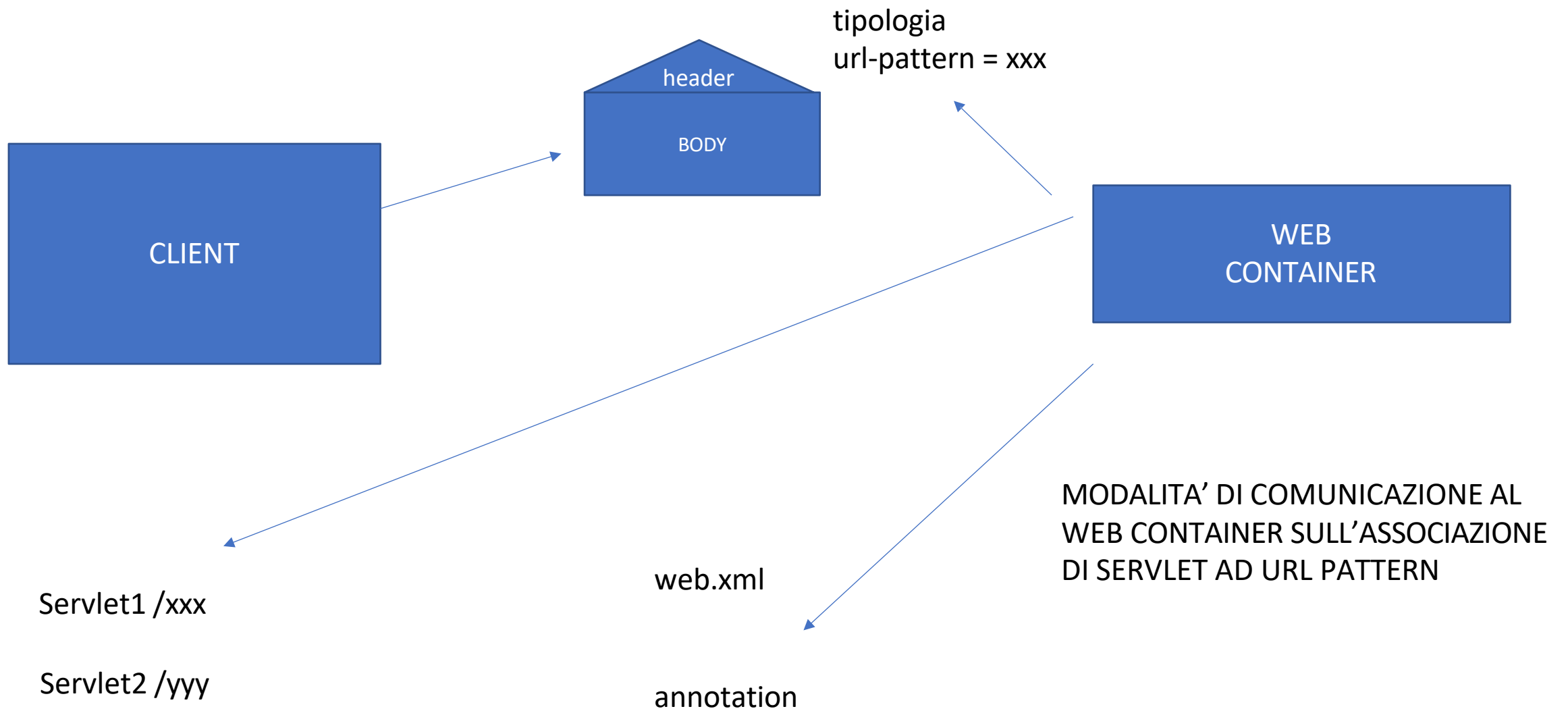
PER JAVA EE ESISTONO SIA POJO SIA SLO (SERVLET/EJB)

- CLASSE ASTRATTA (OOP)
- INTERFACCIA (OOP/FUNZIONALE)

Affinchè una classe possa essere considerata una Servlet occorre che la classe estenda l'API HttpServlet

```
public class MyServlet extends HttpServlet {  
.....  
.....  
}
```

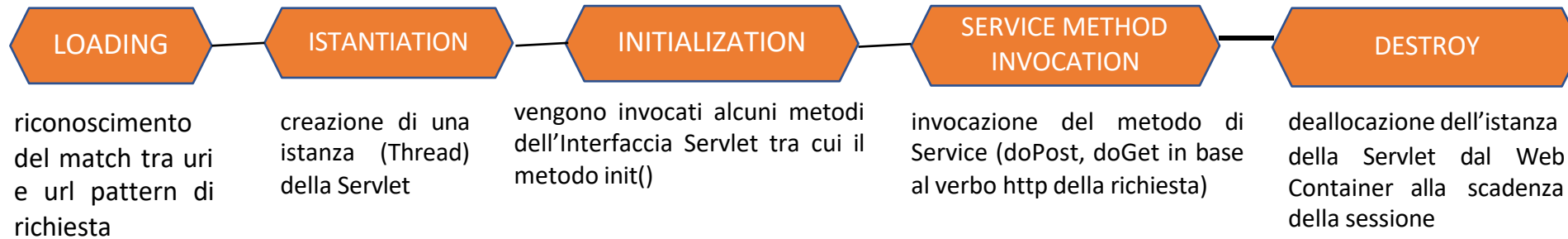




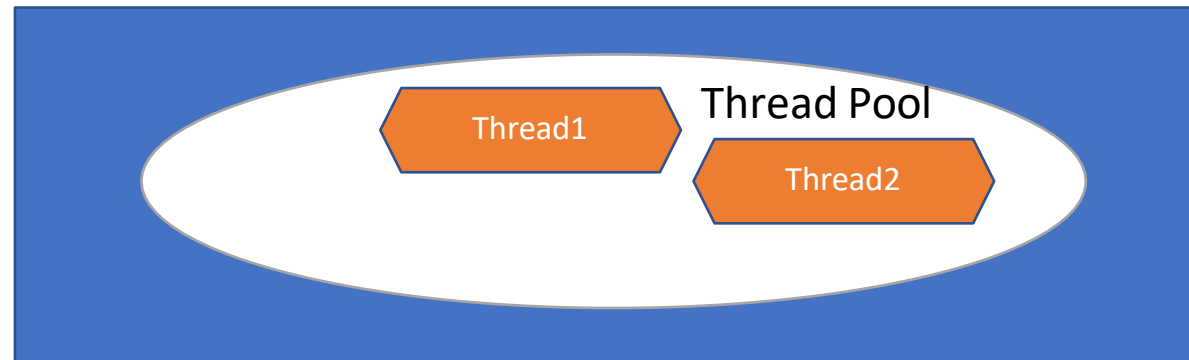
MULTITHREADING

HttpServlet è una API Multithreading di default, per cui per ogni Client che effettua una richiesta viene istanziato dal Web Container un Thread ad esso dedicato. In tal modo viene gestita nativamente la concorrenza dei Client, senza che lo sviluppatore debba preoccuparsene.

In particolare, per ogni richiesta del Client, il Web Container gestisce 5 fasi del ciclo di vita di un Thread della Servlet



WEB CONTAINER



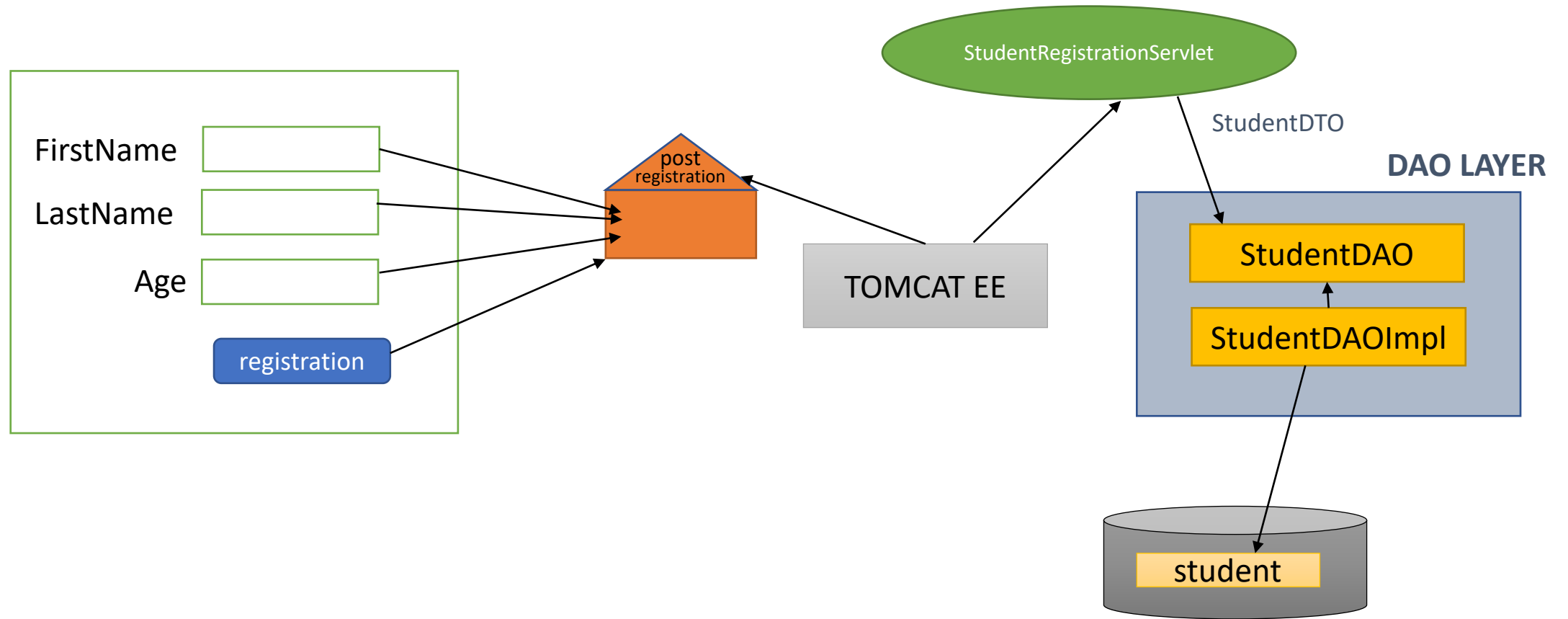
Durante la fase di Istantiation ogni Thread viene allocato all'interno del Thread Pool durante la fase di Destroy

PROCEDURA L'INSTALLAZIONE DI TOMCAT EE:

- Installazione Plugin Eclipse
Help – Install New Software ---->>> <https://download.eclipse.org/releases/latest>
- Download Apache TomEE Plum 8.0.15 al seguente link: <https://tomee.apache.org/download.html>

Name	Version	Date	Size	Signatures & Hashes
📄 TomEE Microprofile TAR.GZ	8.0.15	8 May 2023	62 MB	📄 PGP 📄 SHA512
📄 TomEE Microprofile ZIP	8.0.15	8 May 2023	62 MB	📄 PGP 📄 SHA512
📄 TomEE Plume TAR.GZ	8.0.15	8 May 2023	75 MB	📄 PGP 📄 SHA512
📄 TomEE Plume ZIP	8.0.15	8 May 2023	75 MB	📄 PGP 📄 SHA512
📄 TomEE Plus TAR.GZ	8.0.15	8 May 2023	67 MB	📄 PGP 📄 SHA512
📄 TomEE Plus ZIP	8.0.15	8 May 2023	67 MB	📄 PGP 📄 SHA512
📄 TomEE Webprofile TAR.GZ	8.0.15	8 May 2023	46 MB	📄 PGP 📄 SHA512
📄 TomEE Webprofile ZIP	8.0.15	8 May 2023	46 MB	📄 PGP 📄 SHA512
📄 TomEE Project Source Release ZIP	8.0.15	8 May 2023	17 MB	📄 PGP 📄 SHA512

ARCHITETTURA APPLICAZIONE CRUD_SERVLET



PATTERN MVC = MODEL VIEW CONTROLLER =

OGNI APPLICAZIONE ENTERPRISE (WEB) DOVREBBE ESSERE SUDDIVISA (SCOMPOSTA) IN TRE

STRATI APPLICATIVI (APPLICATION LAYER) AD OGNUNO DEI QUALI VENGONO ASSEGNATI DEI NOMI E DEI RUOLI:

- MODEL (GESTIONE DEI DATI APPLICATIVI RELATIVI AL DATABASE)
- VIEW (PRESENTAZIONE DEI DATI)
- CONTROLLER (GESTIONE DELLE RICHIESTE HTTP PROVENIENTI DAL CLIENT, DELLE RISPOSTE HTTP DA DARE AL CLIENT E DELLA BUSINESS LOGIC)

IL PATTERN MVC VIENE DENOMINATO ANCHE PATTERN THREE TIER

