

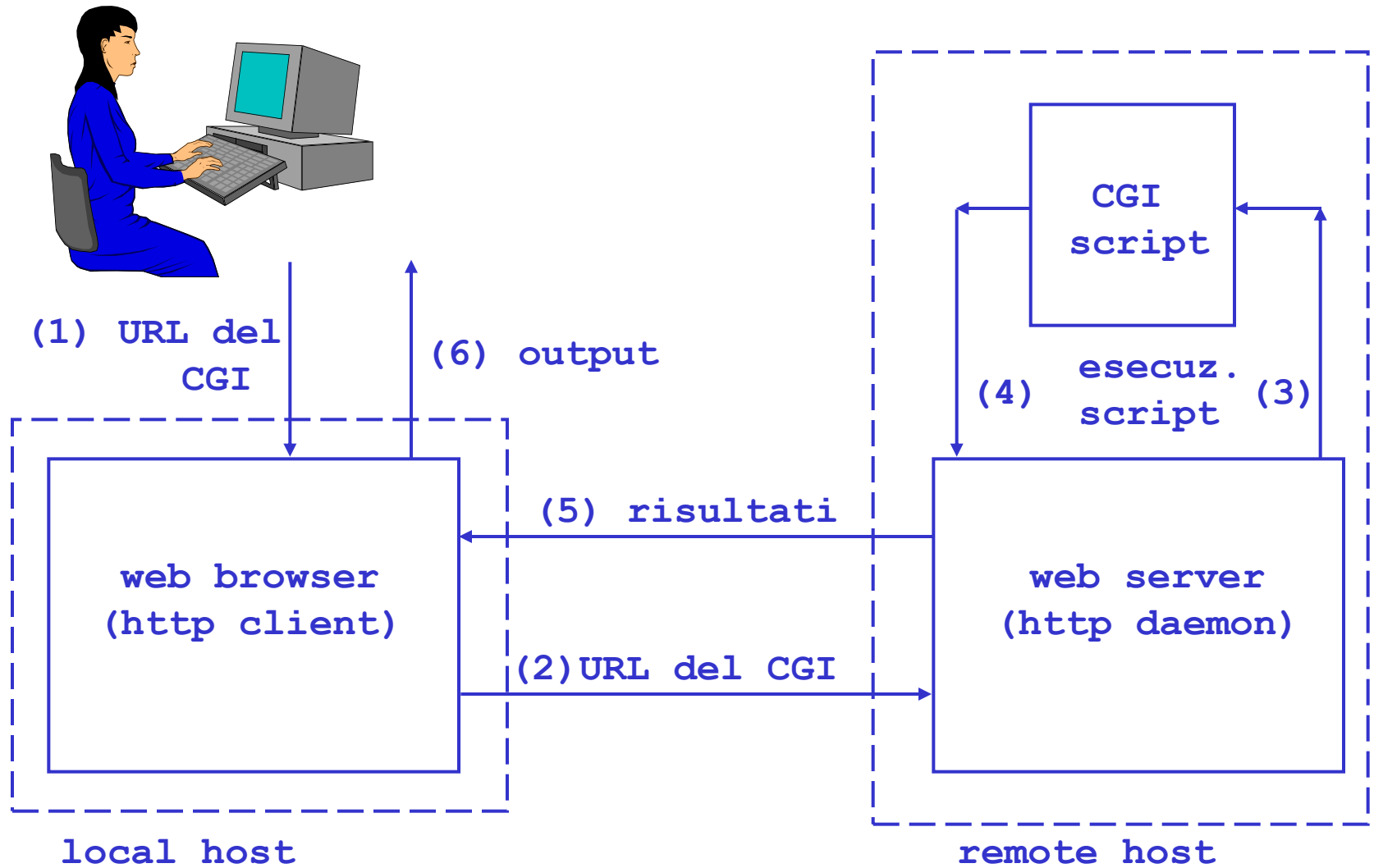
# Common Gateway Interface (CGI)

- ❑ E' uno standard per interfacciare applicazioni esterne con un web server
  - ❖ l'interfacciamento avviene tramite std I/O e variabili d'ambiente (fornite dalla maggior parte dei sistemi operativi)
- ❑ Esempi di applicazioni:
  - ❖ query di un database tramite Web
  - ❖ registrazione di moduli compilati tramite Web (form)
- ❑ Riferimenti:
  - ❖ Specifica CGI 1.1 (NCSA):
  - ❖ <http://www.w3.org/CGI>    <http://tools.ietf.org/html/draft-robinson-www-interface-00>

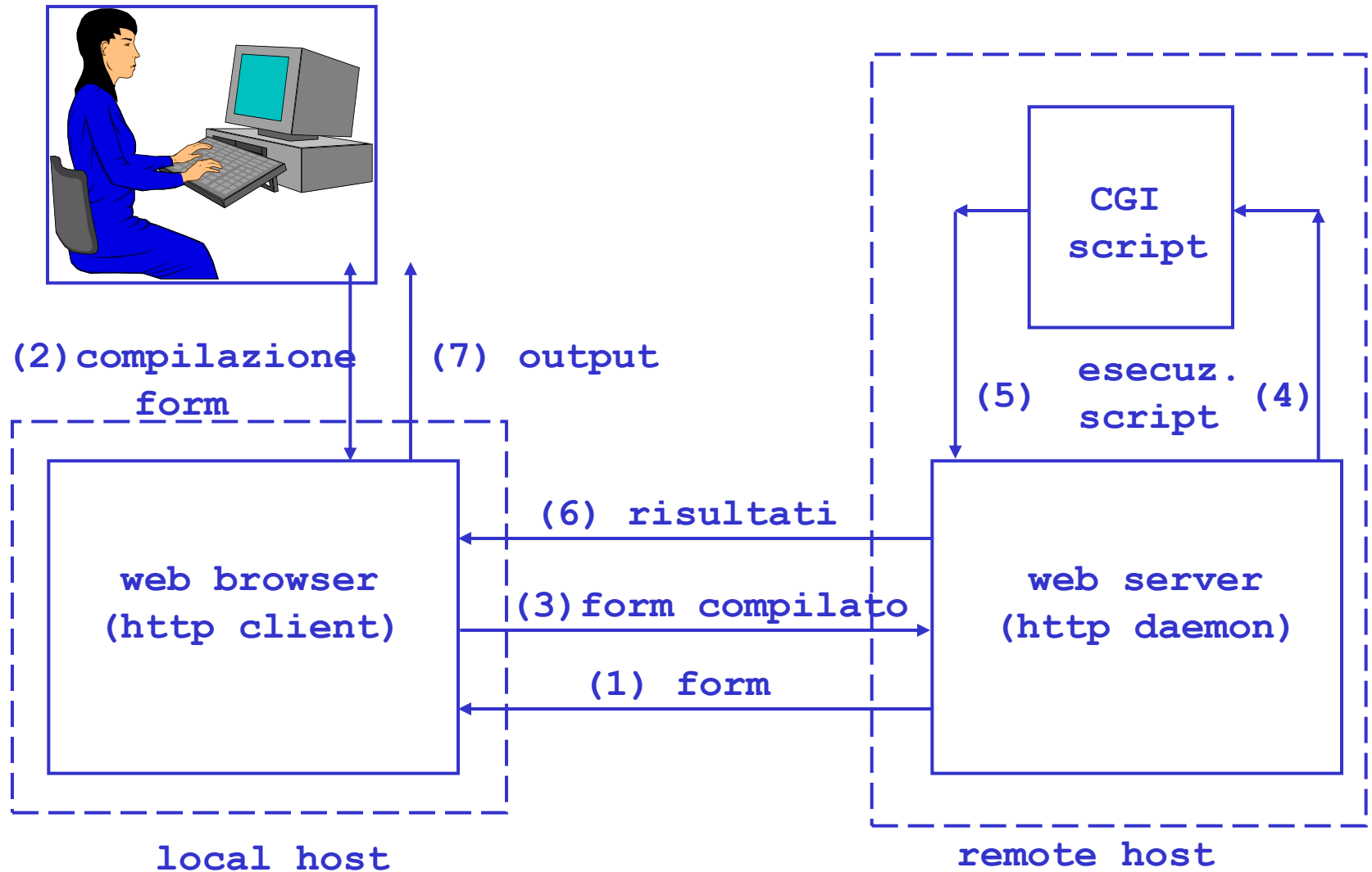
# Attivazione di un'Applicazione CGI

- ❑ L'applicazione può essere un qualunque programma, anche realizzato tramite script (script CGI). Tipico linguaggio usato: Perl
- ❑ L'esecuzione può essere attivata in due modi diversi:
  - ❖ inviando direttamente l'URL dell'applicazione CGI da attivare (attivazione diretta/metodo GET)
  - ❖ compilando e sottomettendo un form che fa riferimento all'applicazione CGI

# Esempio: attivazione diretta CGI



# Meccanismo CGI: Uso dei FORM



# Ricezione dei Dati di Input

## ❑ Metodo GET:

- ❖ Eventuali dati sono inclusi nell'URL del CGI (urlencoded)
- ❖ Lo script CGI riceve la stringa dei dati nella variabile di ambiente **QUERY\_STRING**

## ❑ Metodo POST:

- ❖ Eventuali dati vengono inviati dal client nella richiesta POST
- ❖ Lo script CGI riceve i dati dallo standard input e la loro lunghezza nella variabile di ambiente **CONTENT\_LENGTH**

- ❑ La variabile di ambiente **REQUEST\_METHOD** permette allo script CGI di determinare il metodo usato

# Invio del Risultato da parte del CGI

- ❑ Il risultato è codificato in una risposta HTTP (tipicamente un documento HTML)
- ❑ Un CGI scrive semplicemente sullo standard output
  - ❖ Quanto scritto viene inviato al client

# Accesso ai cookie da CGI

- ❑ I cookies spediti con la richiesta (nell'header `Cookie:` di HTTP ) sono accessibili nella variable d'ambiente `HTTP_COOKIE`

# Tipica Struttura di uno Script CGI

- ❖ Ricezione e Decodifica degli eventuali input (p. es. i campi del form)
- ❖ Elaborazione
- ❖ Generazione dei risultati su standard output in formato HTML
- ❑ Esistono librerie in diversi linguaggi per eseguire la parte di ricezione e decodifica
  - ❖ Esempio: ANSI C `cgic` library



# Esempio di CGI script in C

□ Esempio: hello.c

# Limiti del CGI

- ❑ Ogni richiesta si traduce in una nuova esecuzione del programma
  - ❖ Elevato overhead, elevati tempi di latenza
  - ❖ Assenza di un ambiente (e di uno stato) che sopravvive alla singola interazione
    - ⇒ difficoltà nella gestione di sessioni

# Alternative agli script CGI

## □ Java Servlets

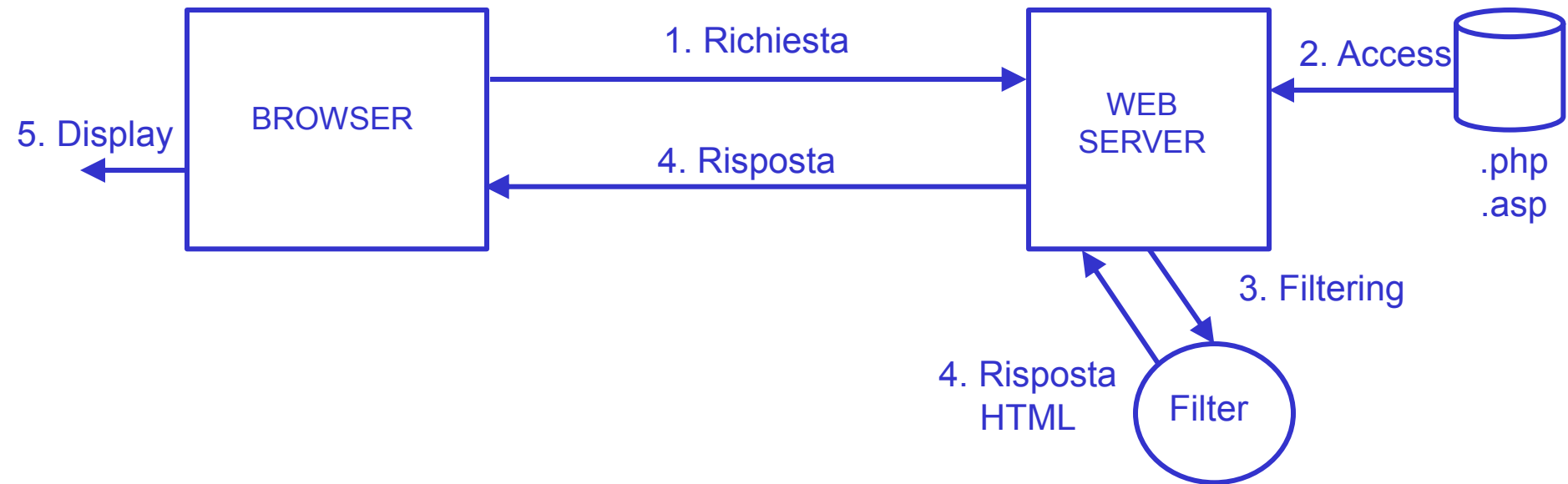
- ❖ **Componenti** Java che girano sul server in uno speciale «container» (sempre attivo)
- ❖ Ogni **servlet** è collegata ad un **URL**. Le richieste dirette a quell'URL sono automaticamente girate alla servlet tramite l'invocazione di un metodo della servlet

## □ Pagine Web dinamiche

- ❖ **Pagine HTML con un contenuto di markup particolare** (cioè istruzioni) che sono interpretate quando la pagina viene richiesta e generano il contenuto della pagina
- ❖ Rispetto ai CGI e alle servlet, che sono considerate difficili da programmare e nelle quali è facile sbagliarsi, le pagine web dinamiche sono facili, anche da parte di non programmatori

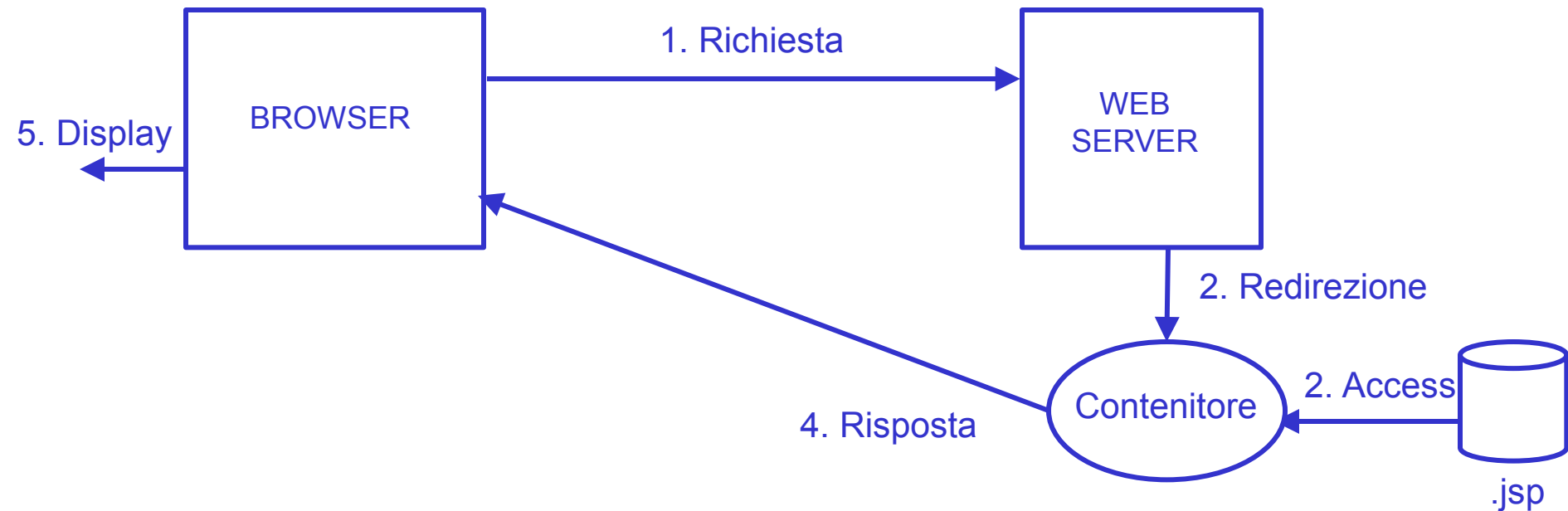
# Pagine Web Dinamiche

❑ Server-side processing. Esempio: PHP, ASP

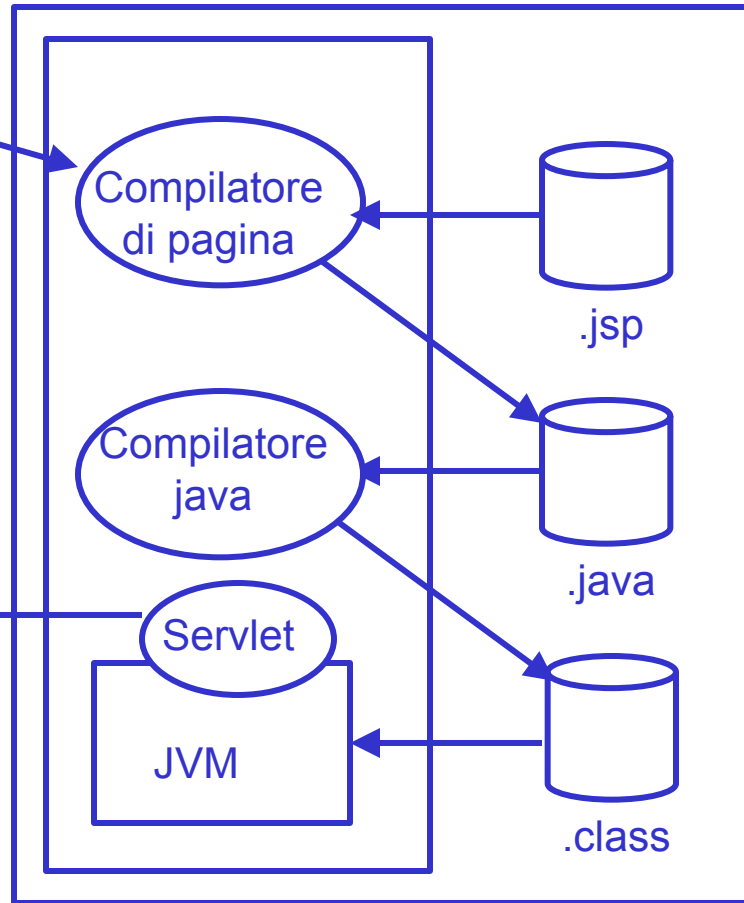


# Pagine Web Dinamiche

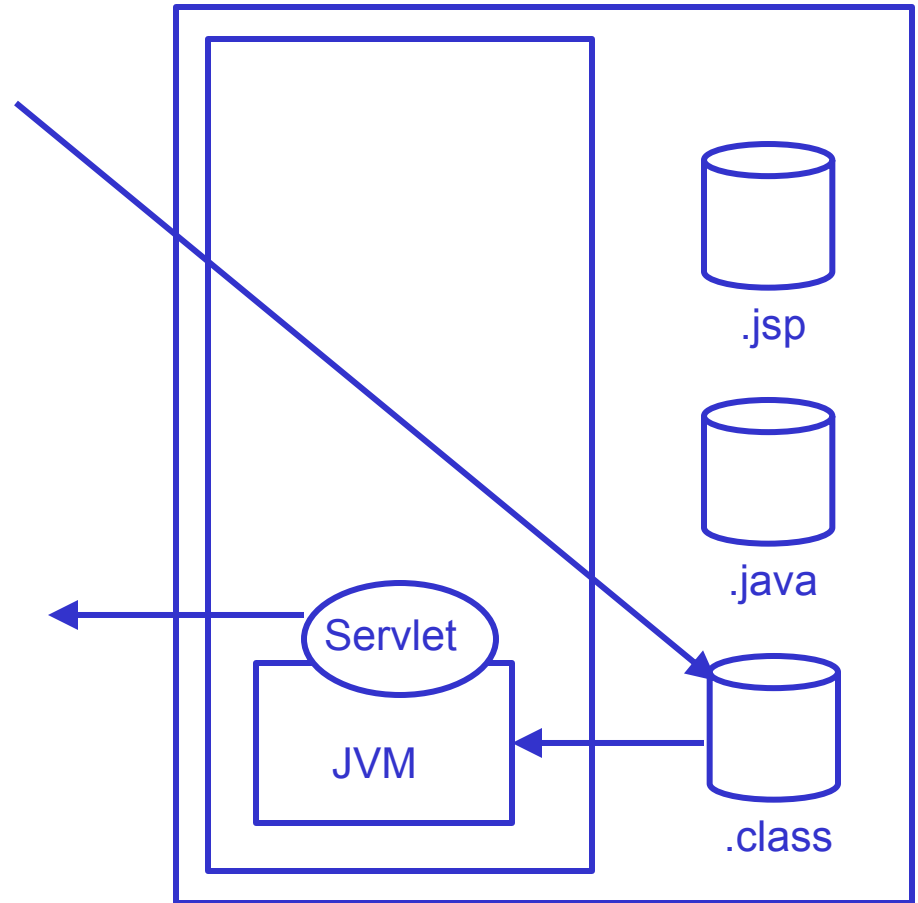
❑ Server-side processing. Esempio: JSP



# Contenitore JSP



La prima volta la pagina va compilata



Le volte successive si esegue solo