Programowanie portali biznesowych

WYKŁAD 2

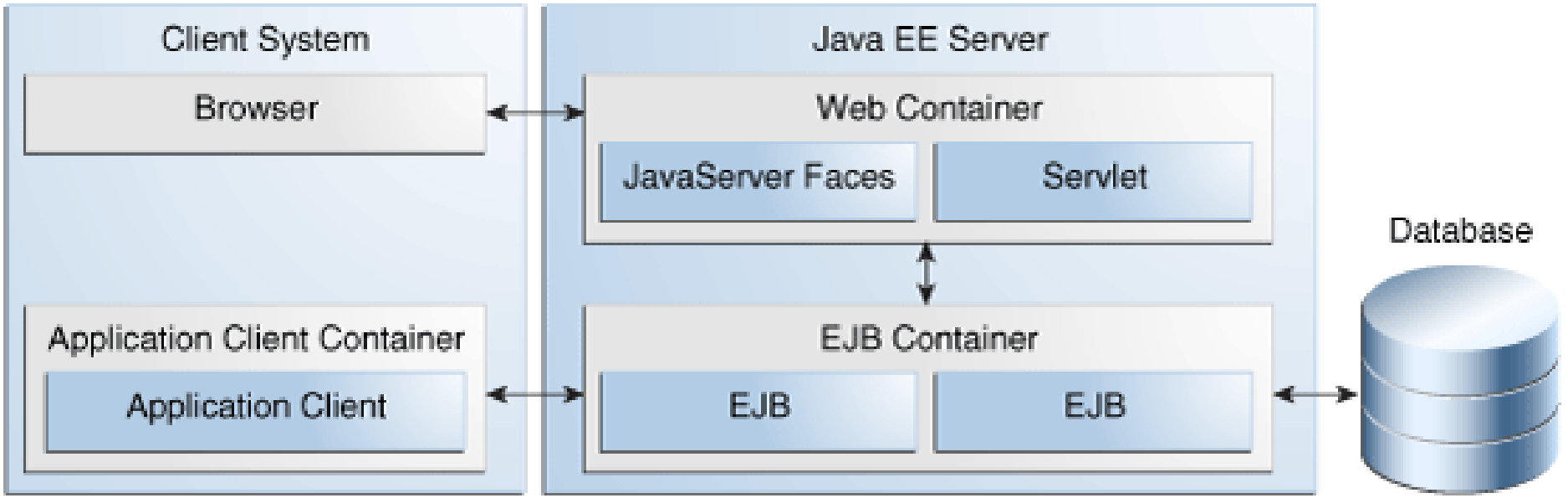
Java EE

- Java Enterprise Edition (EE) - platforma tworzenia aplikacji biznesowych w języku Java.
- Java EE jest rozwijana przy użyciu Java Community Process (JCP):

`www.jcp.org`

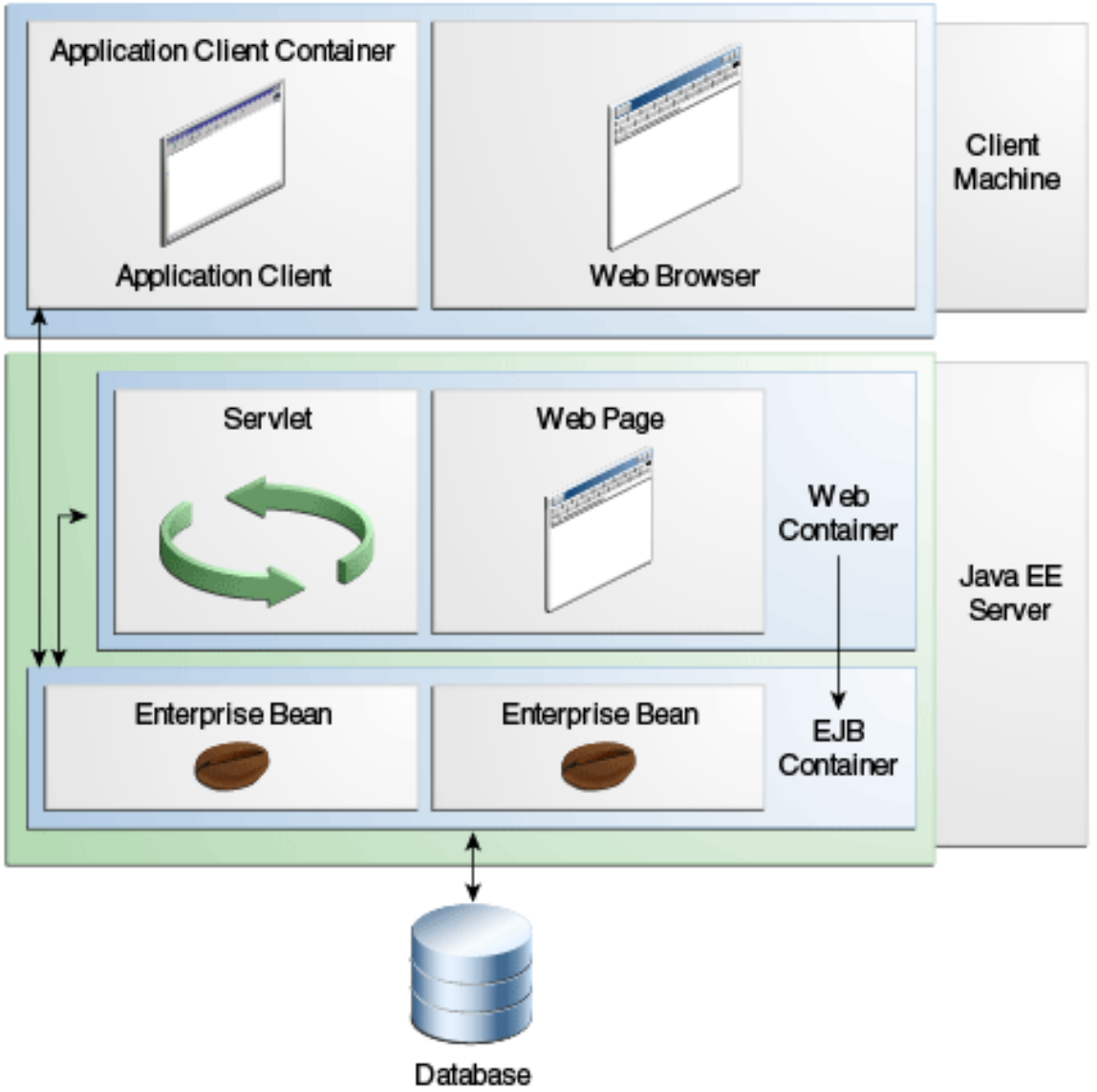
- JCP jest mechanizmem tworzenia standardowych specyfikacji technicznych technologii Java.
- Aktualna wersja: Java EE 8.

Java EE



Źródło: https://docs.oracle.com/javaee/7/tutorial/overview007.htm

Java EE



Źródło: https://docs.oracle.com/javaee/7/tutorial/overview004.htm

Java EE

- Kontener Web:
  - zarządza uruchamianiem stron WWW, serwletów oraz niektórych ziaren EJB,
  - uruchamiany jest na serwerze Java EE.
- Kontener EJB:
  - zarządza uruchamianiem ziaren EJB,
  - uruchamiany jest na serwerze Java EE.

Java EE 8

Specyfikacja JSR (Java Specification Request) 366 dla Java EE 8:

https://jcp.org/en/jsr/detail?id=366

Java EE 8

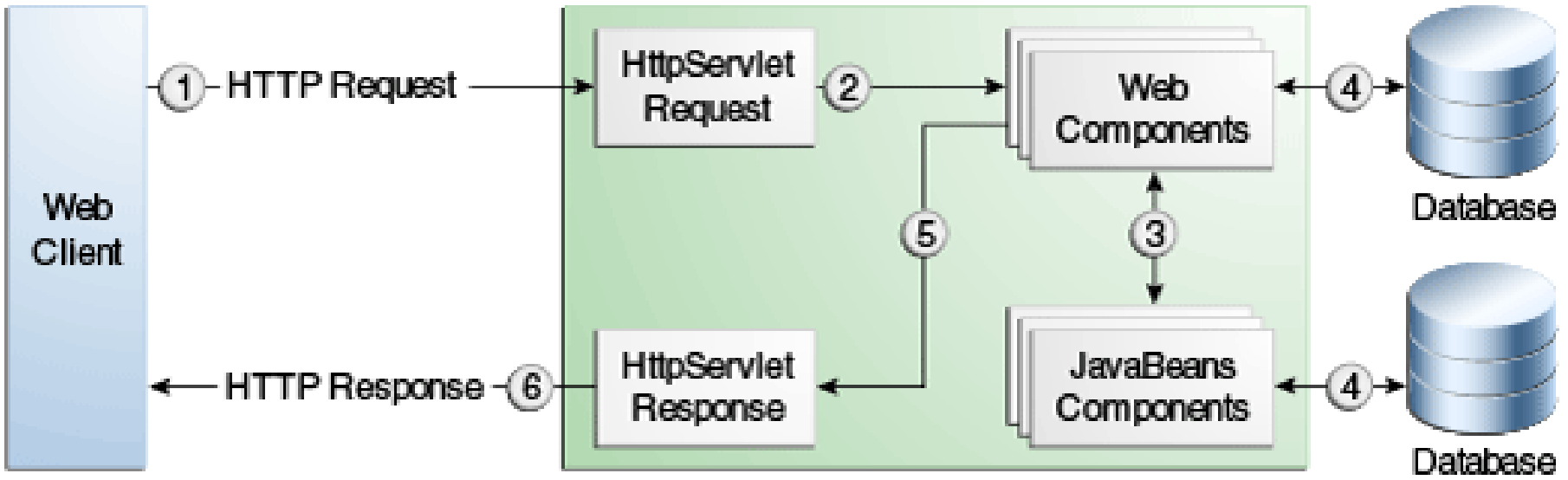
- Nowe specyfikacje w stosunku do Java EE 7:
  - JSR 366 – Java EE 8 Platform
  - JSR 365 – Contexts and Dependency Injection (CDI) 2.0
  - JSR 367 – The Java API for JSON Binding (JSON-B) 1.0
  - JSR 369 – Java Servlet 4.0
  - JSR 370 – Java API for RESTful Web Services (JAX-RS) 2.1
  - JSR 372 – JavaServer Faces (JSF) 2.3
  - JSR 374 – Java API for JSON Processing (JSON-P)1.1
  - JSR 375 – Java EE Security API 1.0
  - JSR 380 – Bean Validation 2.0
  - JSR 250 – Common Annotations 1.3
  - JSR 338 – Java Persistence 2.2
  - JSR 356 – Java API for WebSocket 1.1
  - JSR 919 – JavaMail 1.6

Adnotacje

- Platforma Java EE wykorzystuje uproszczony model programowania.
- Deskryptory XML (deployment descriptors - DDs) wykorzystania komponentów są opcjonalne. Informacja konfiguracyjna może być umieszczona bezpośrednio w kodzie Java jako tzw. **adnotacje:**

@adnotacja

Aplikacje webowe



Źródło: https://docs.oracle.com/javaee/7/tutorial/webapp001.htm

Aplikacje webowe

- Komponenty Web:
  - serwlety,
  - strony WWW tworzone przy użyciu technologii JavaServer Faces,
  - punkty końcowe usług sieciowych (Web Services),
  - strony JSP.

Aplikacje webowe

- Mapowanie URL na komponent Web:

```
http://host:port/context-root[/url-pattern]
```

- Przykład:

```
http://localhost:8080/Serwlet/test
```

Specyfikacja JSR (Java Specification Request) 369 dla Java Servlet 4.0:

https://jcp.org/en/jsr/detail?id=369

Serwlety

- Serwlet jest komponentem aplikacji webowej utrzymywanym przez kontener serwletów i generującym dynamiczną zawartość.
- Komunikacja z serwletem realizowana jest w modelu żądanie-odpowiedź.

Serwlety

- Kontener serwletów jest odpowiedzialny za:
  - zarządzanie cyklem życia serwletów,
  - odbieranie żądań,
  - wysyłanie odpowiedzi,
  - operacje kodowania i dekodowania danych.

Serwlety

- Definiowanie serwletu:
  - adnotacja **`@WebServlet`**
  - rozszerzenie klasy **`javax.servlet.http.HttpServlet`**

Interfejs Servlet

- Klasa HttpServlet definiuje po jednej metodzie dla każdej metody protokołu HTTP:

Metoda HTTP	Metoda serwletu
GET	doGet
POST	doPost
PUT	doPut
DELETE	doDelete
HEAD	doHead
OPTIONS	doOptions
TRACE	doTrace

Interfejsy HttpServletRequest i HttpServerResponse

- **HttpServletRequest** – interfejs dla obiektu reprezentującego żądanie klienta.
- **HttpServletResponse** – interfejs dla obiektu reprezentującego odpowiedź serwera.

Interfejsy HttpServletRequest i HttpServerResponse

- Obiekt HttpServletRequest zawiera m.in.:
  - parametry żądania,
  - nagłówek protokołu HTTP,
  - poszczególne ścieżki (*host*, *port*, *context*).

Metody cyklu życia serwletu

- Metody interfejsu Servlet cyklu życia serwletu wywoływane przez kontener serwletów:
  - `init`
  - `service`
  - `destroy`

Definiowanie wzorców URL

- Przykład:

```
@WebServlet(urlPatterns = {"/serwlet"})
```

- Serwlety mogą być uruchamiane pod różnymi adresami URL, np.:

```
@WebServlet(urlPatterns = {"/serwlet", "/przyklad"})
```

Pobieranie parametrów żądania

- Parametry żądania dla obu metod (GET i POST) pobierane są za pomocą metody getParameter.
- Przykład:

```
request.getParameter("liczba");
```

Generowanie odpowiedzi

- Przykład:

```
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Przykład</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Liczba: "+
     request.getParameter("liczba")+"</h1>");
out.println("</body>");
out.println("</html>");
```