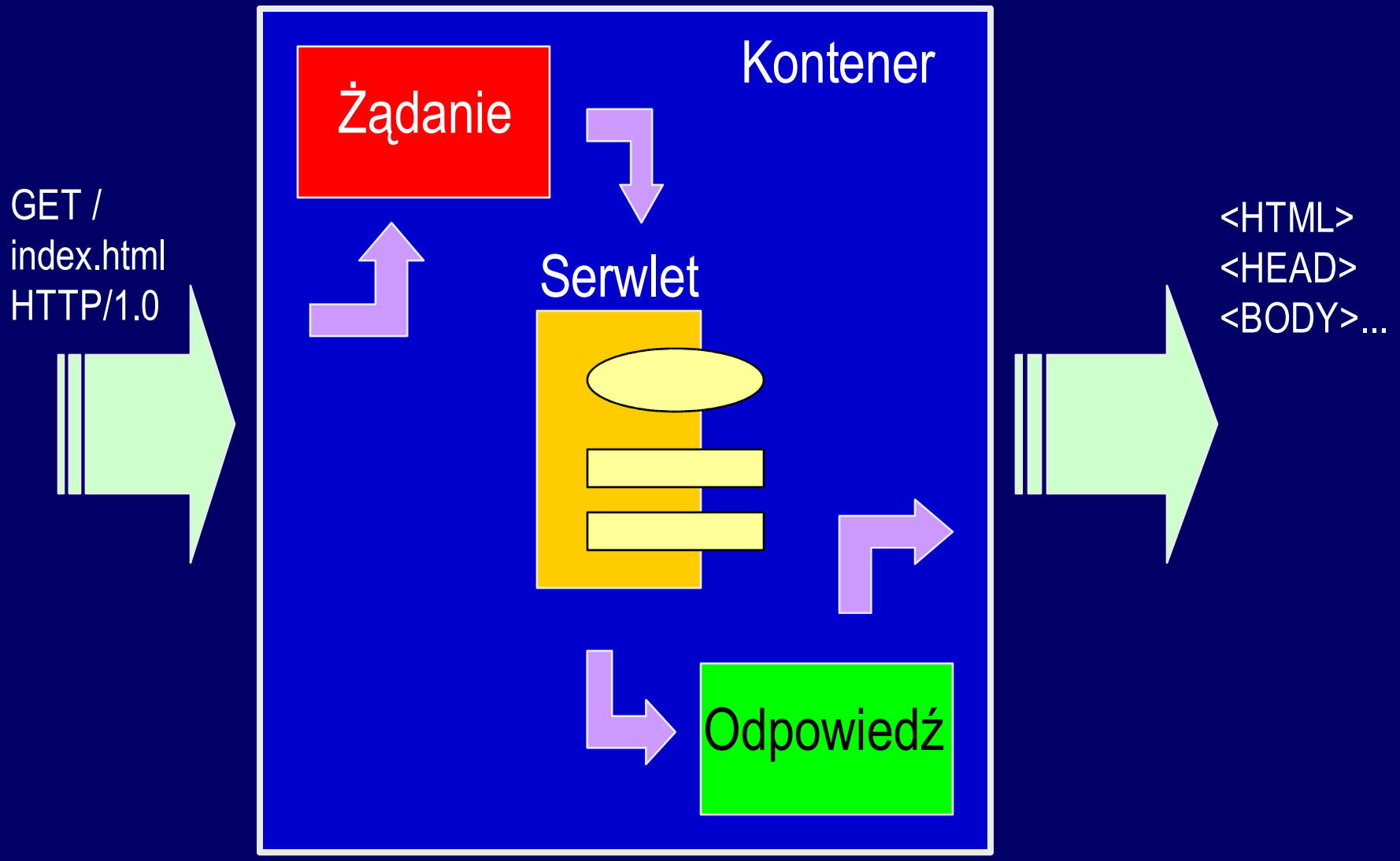
Komunikacja między serwletami

Bartosz Walter
Instytut Informatyki Politechniki Poznańskiej

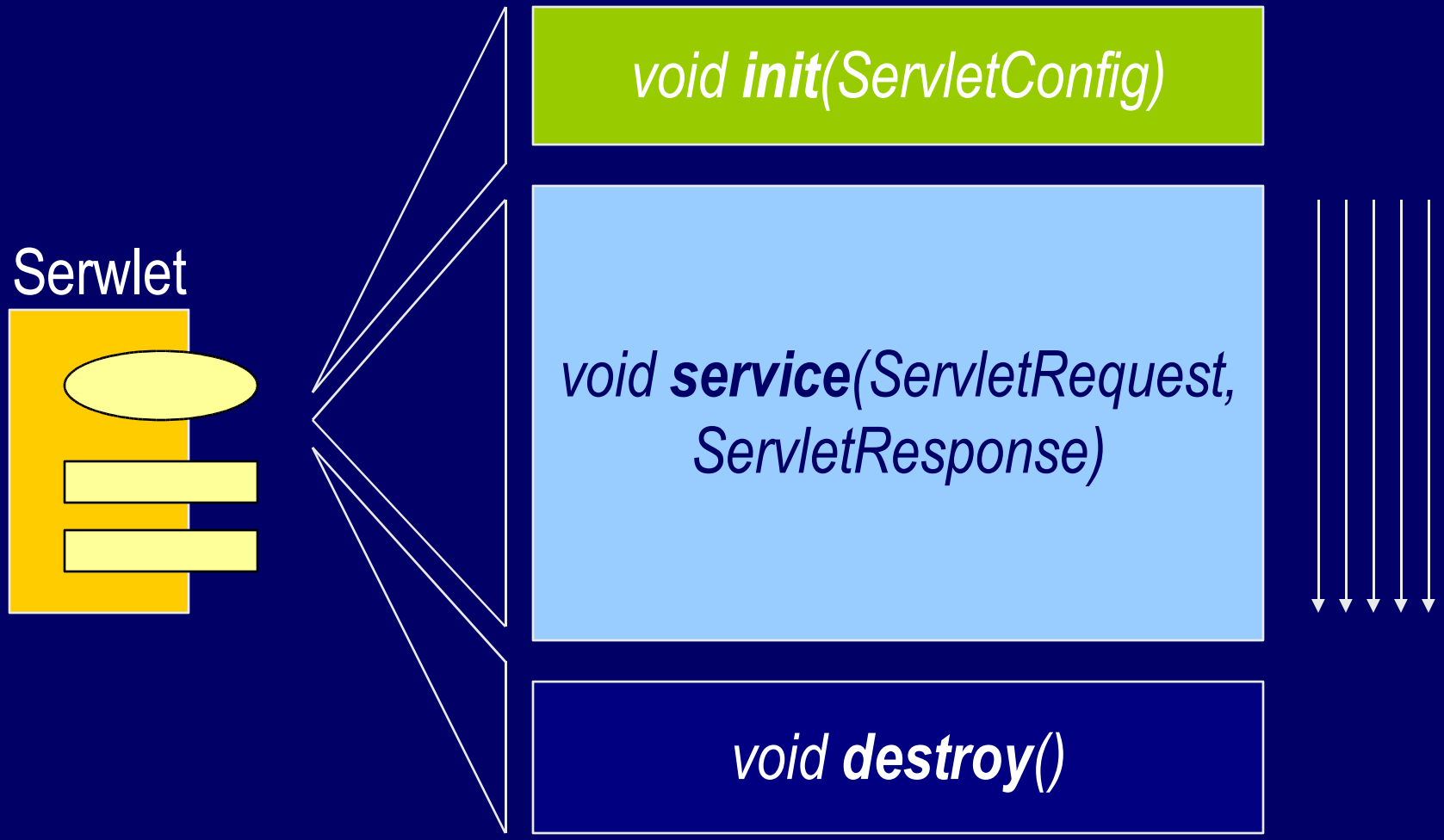
Sceny z życia serwletów

Obsługa żądań



Sceny z życia serwletów

Narodziny, życie i śmierć



`javax.servlet.Servlet`

Sceny z życia serwletów

Środowisko i konfiguracja

Definicja nazwy serwleta

Parametry serwleta

Przypisanie serwleta do URLa

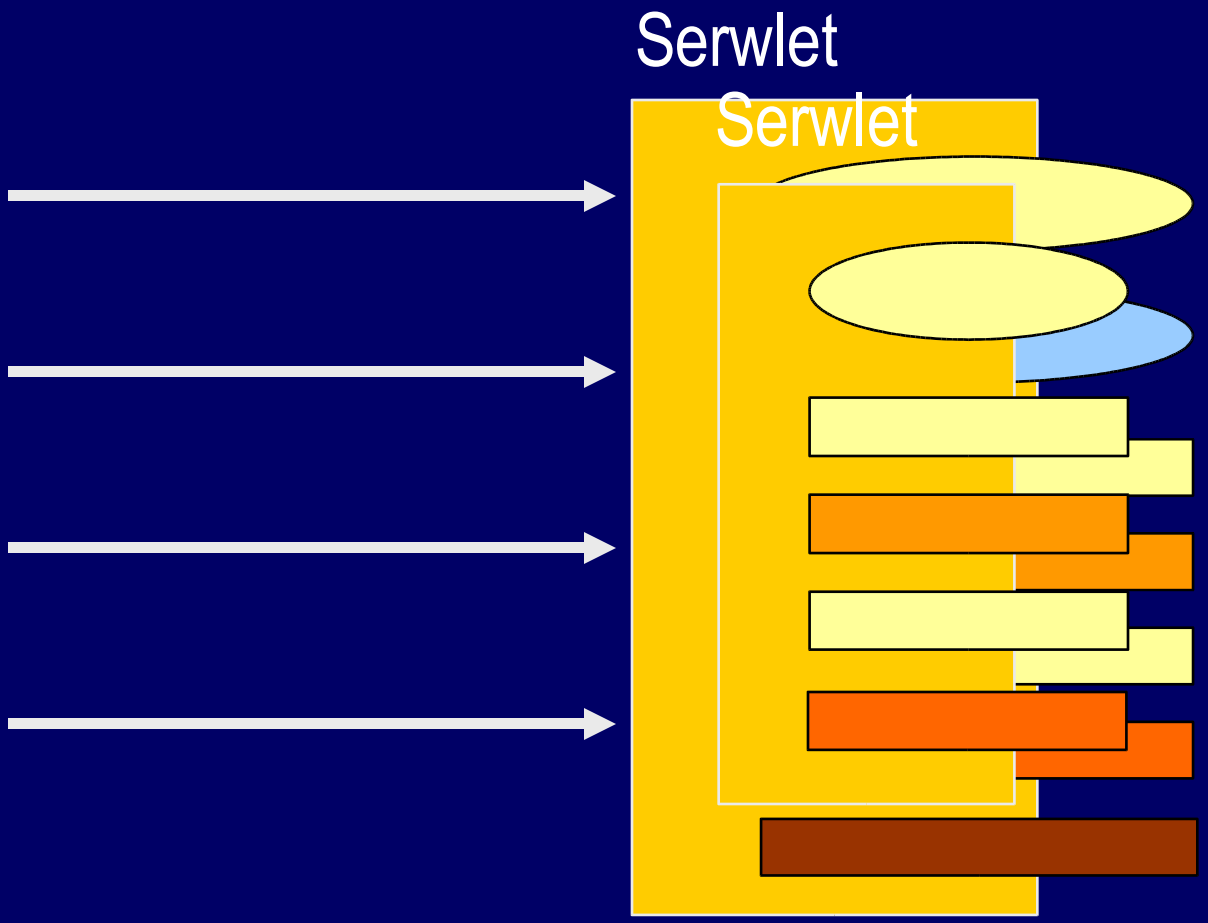
web.xml

```
<servlet>
  <servlet-name>serwlet_A</servlet-name>
  <servlet-class>
    moj.pakiet.ServletA
  </servlet-class>
  <init-param>
    <param-name>parametr</param-name>
    <param-value>wartość</param-value>
  </init-param>
</servlet>

<servlet-mapping>
  <servlet-name>serwlet_A</servlet-name>
  <url-pattern>/A/*</url-pattern>
</servlet-mapping>
```

Aplikacje się rozrastają...

Zwiększamy funkcjonalność

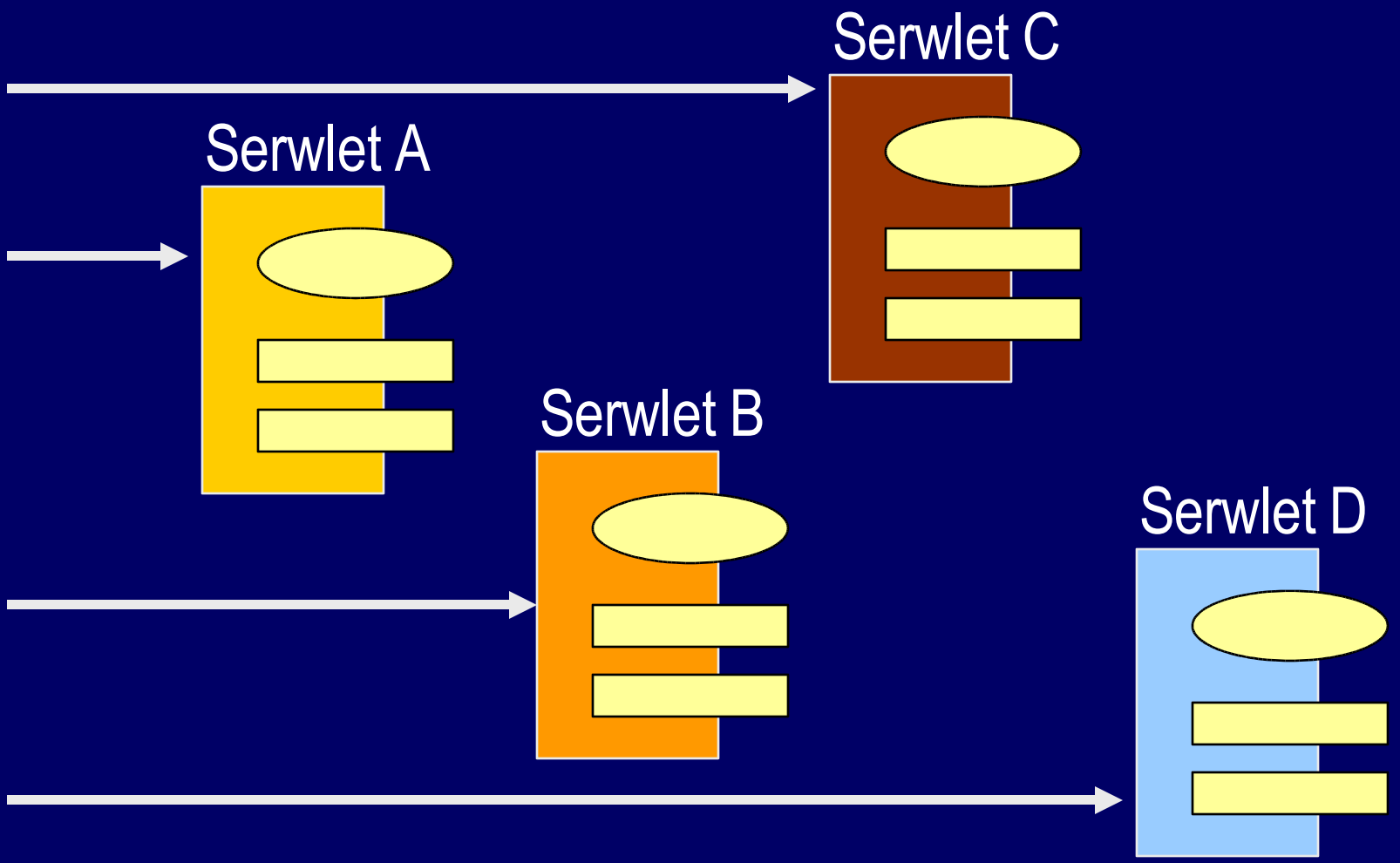
Pomysł 1: puchnący serwlet



Aplikacje się rozrastają...

Zwiększamy funkcjonalność

Pomysł 2: specjalizowane serwlety



Aplikacje się rozrastają...

Zwiększamy funkcjonalność

A czego potrzebujemy?

- ✓ **komunikacji między serwletami**
- ✓ **specjalizacji funkcjonalnej**
- ✓ **modularyzacji**
- ✓ **konfiguracji przez administratora**

Rys historyczny cz. I

Java Servlets™ 2.0

dawno temu, w odległej galaktyce...

- ✓serwlet jako jednostka przetwarzająca żądania
- ✓brak mechanizmów komunikacyjnych
- ✓podstawowe mechanizmy obsługi żądań
- ✓Apache JServ

Rys historyczny cz. II

Java Servlets™ 2.1

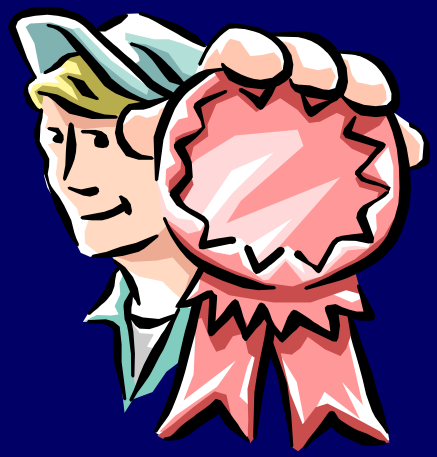
listopad 1998

- ✓podstawy komunikacji między serwletami
- ✓wprowadzenie pojęcia *kontenera serwletów*
- ✓Apache Tomcat 3.1

Rys historyczny cz. III

Java Servlets™ 2.2

grudzień 1999



- ✓wprowadzenie pojęcia *aplikacji webowej*
- ✓rozszerzenie mechanizmu komunikacji między serwletami
- ✓obserwatory zdarzeń
- ✓Apache Tomcat 3.2

Rys historyczny cz. IV

Java Servlets™ 2.3

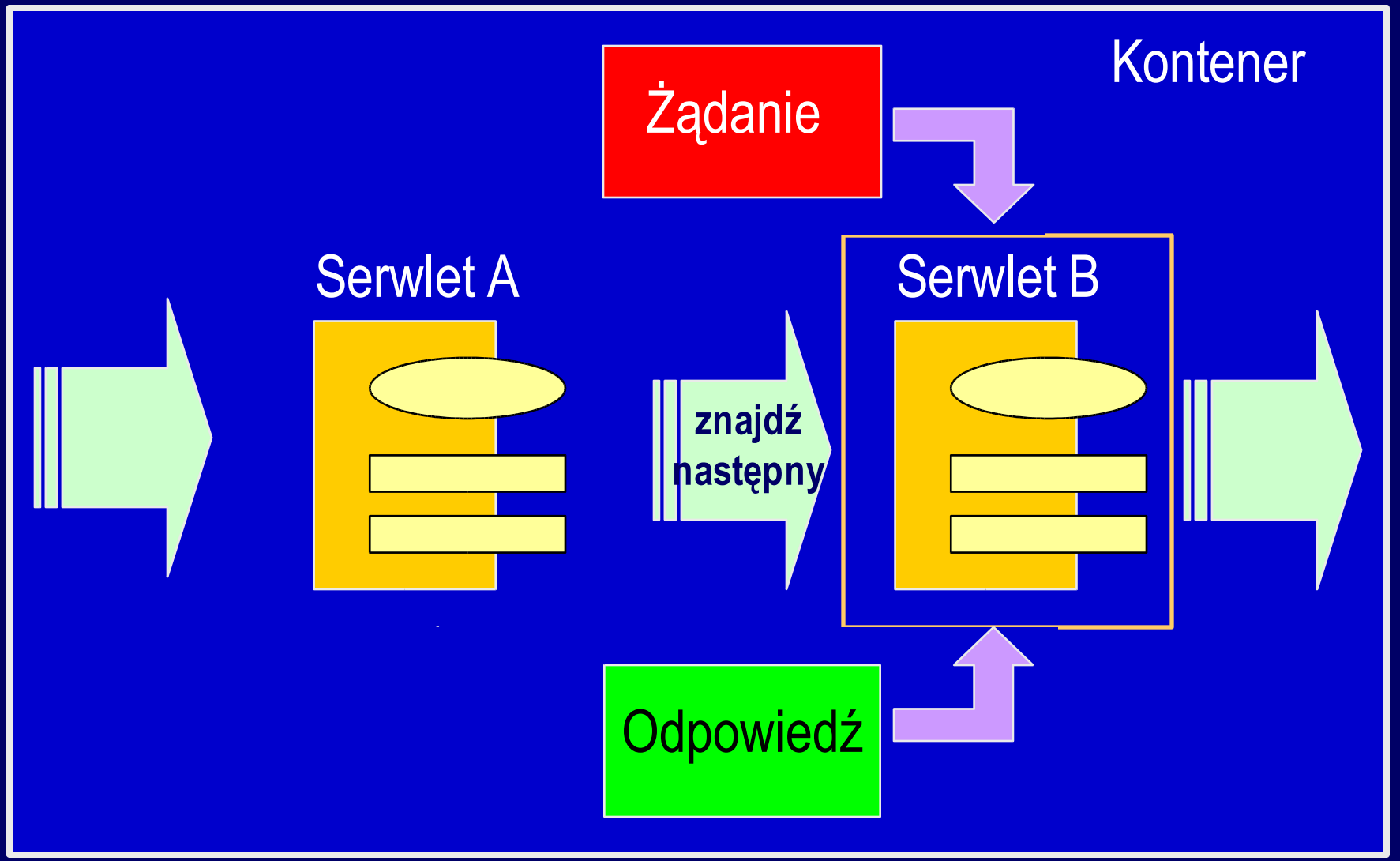
sierpień 2001

- ✓mechanizm filtrowania
- ✓zdarzenia w cyklu życia serwleta
- ✓integracja z J2EE
- ✓Apache Tomcat 4.x i Borland Enterprise Server

O czym chcemy mówić?

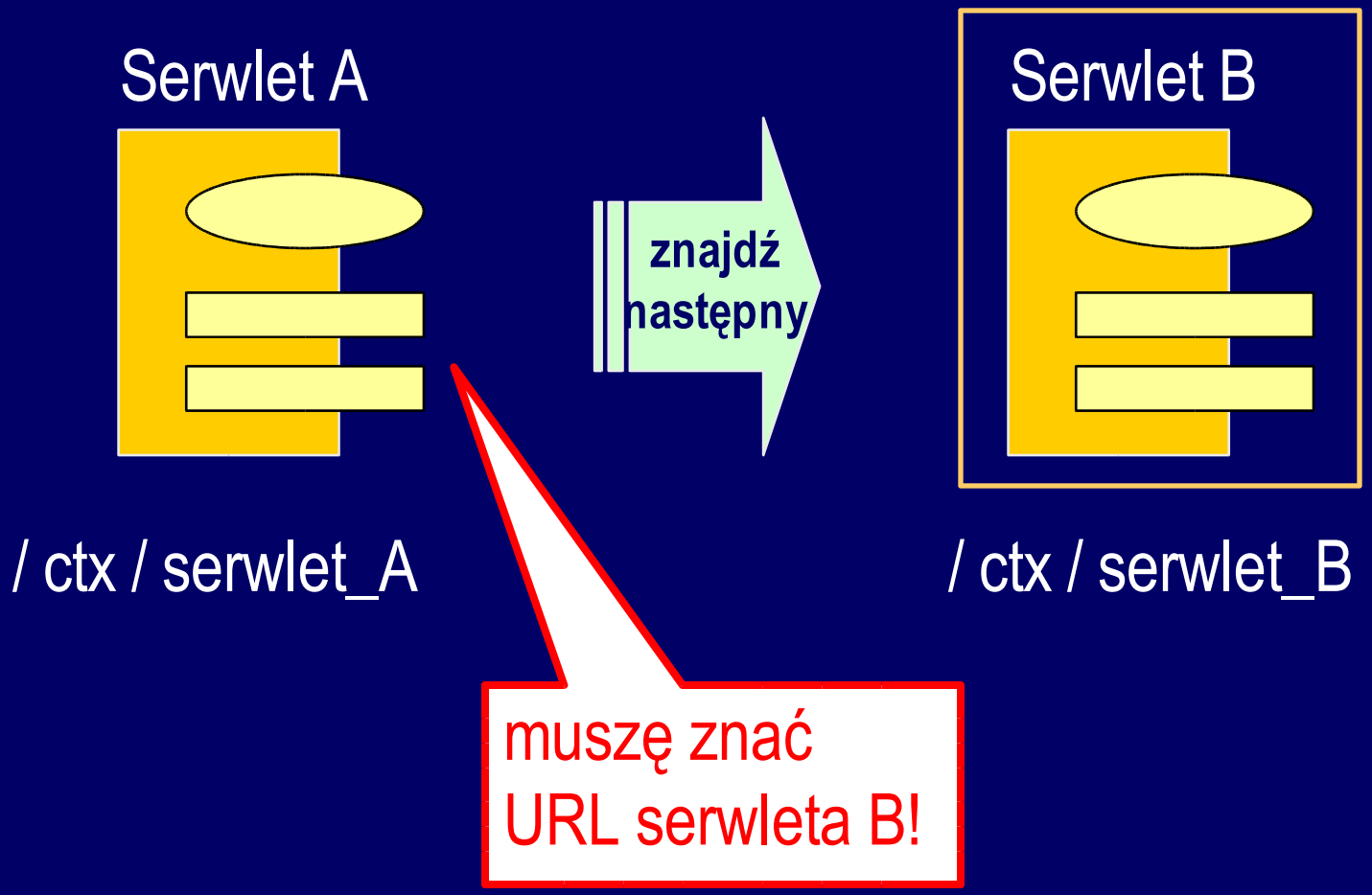
- ✓ **Komunikacja między serwletami**
  - ✓ jak to działa?
  - ✓ modele przetwarzania
- ✓ **Filtrowanie**
  - ✓ model przetwarzania
  - ✓ filtrowanie żądań i odpowiedzi
  - ✓ kilka sztuczek

Komunikacja między serwletami



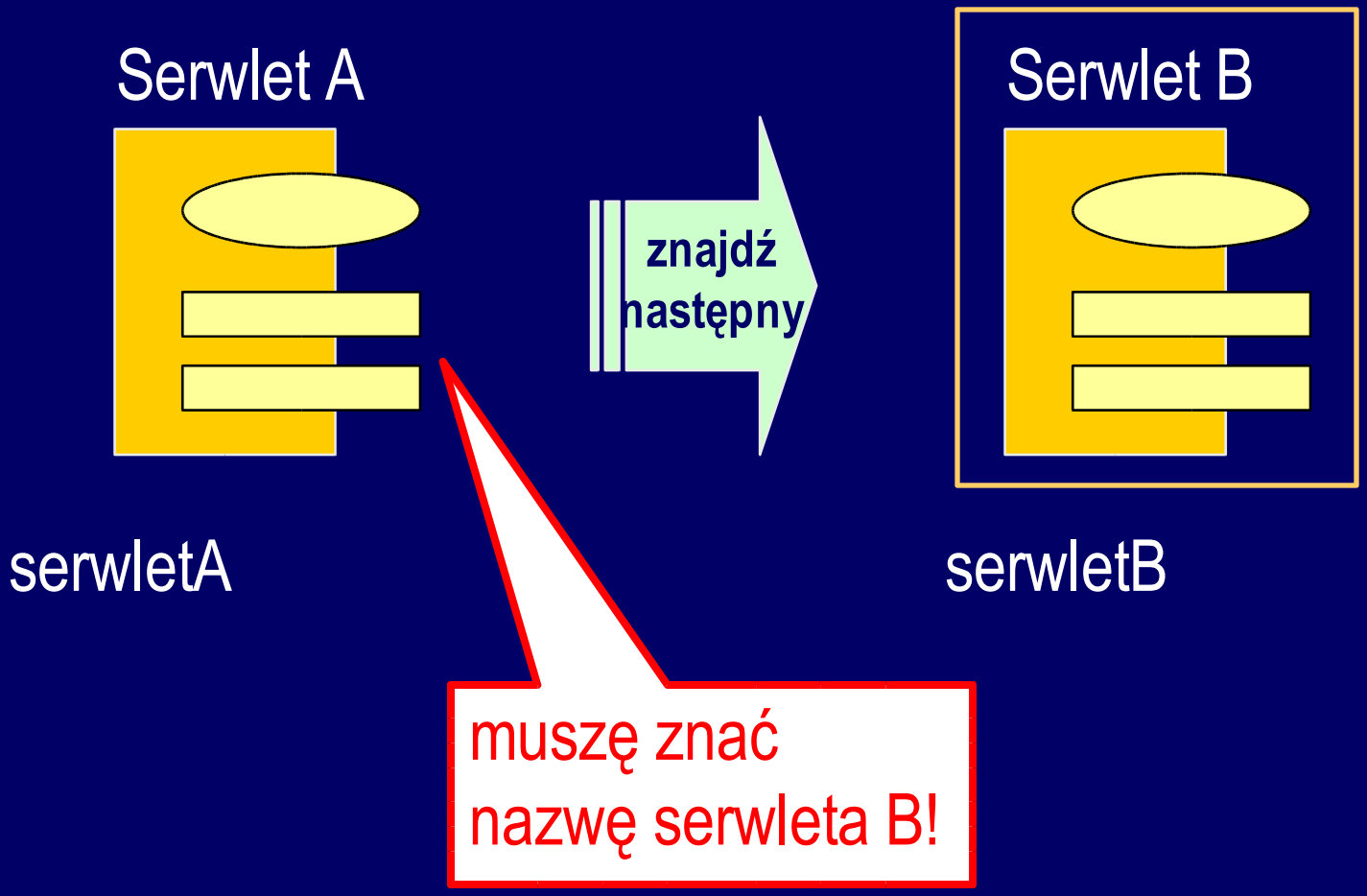
Znajdowanie serwletów cz. I

Sposób 1: podanie URLa

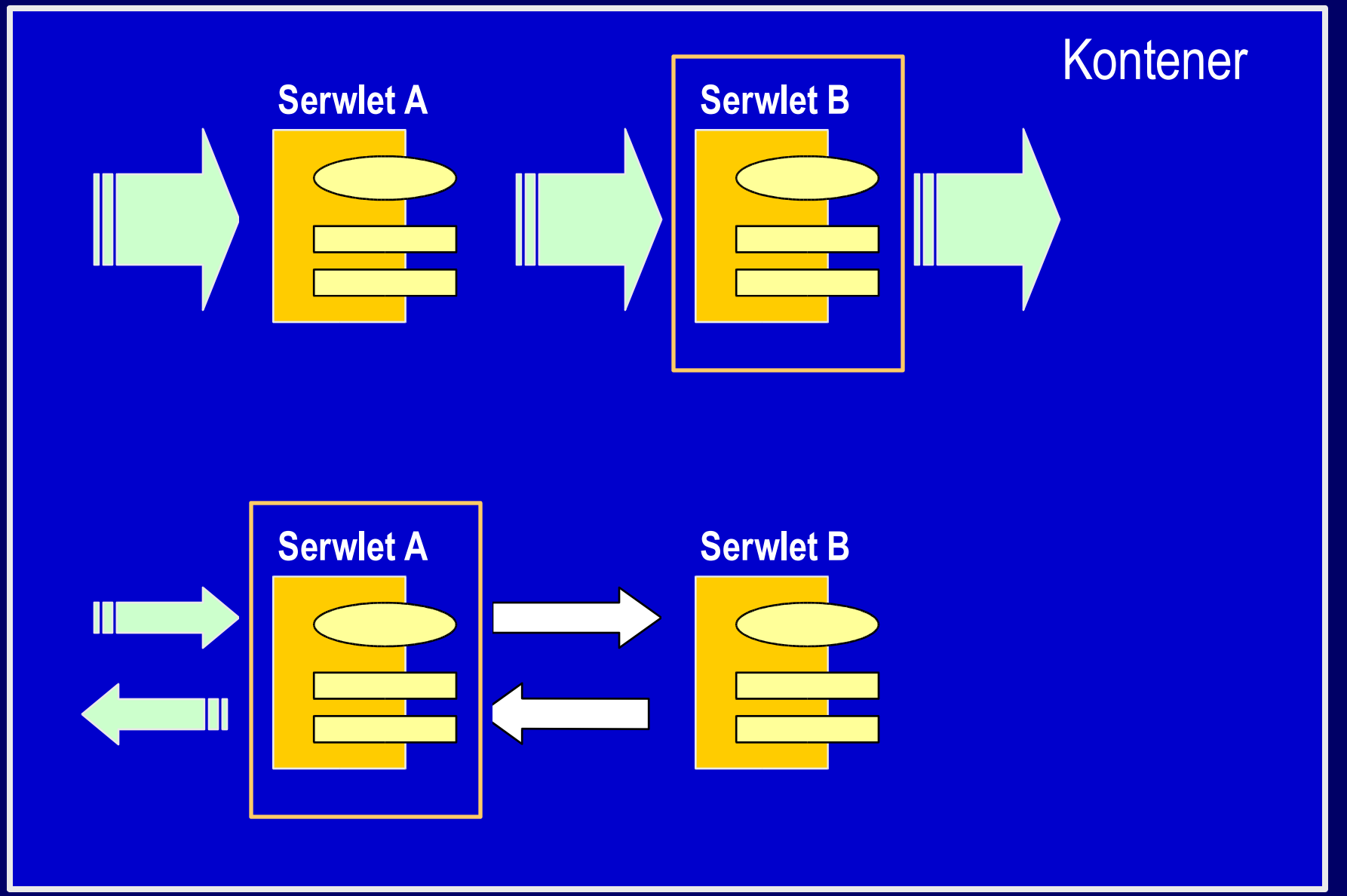


Znajdowanie serwletów cz. II

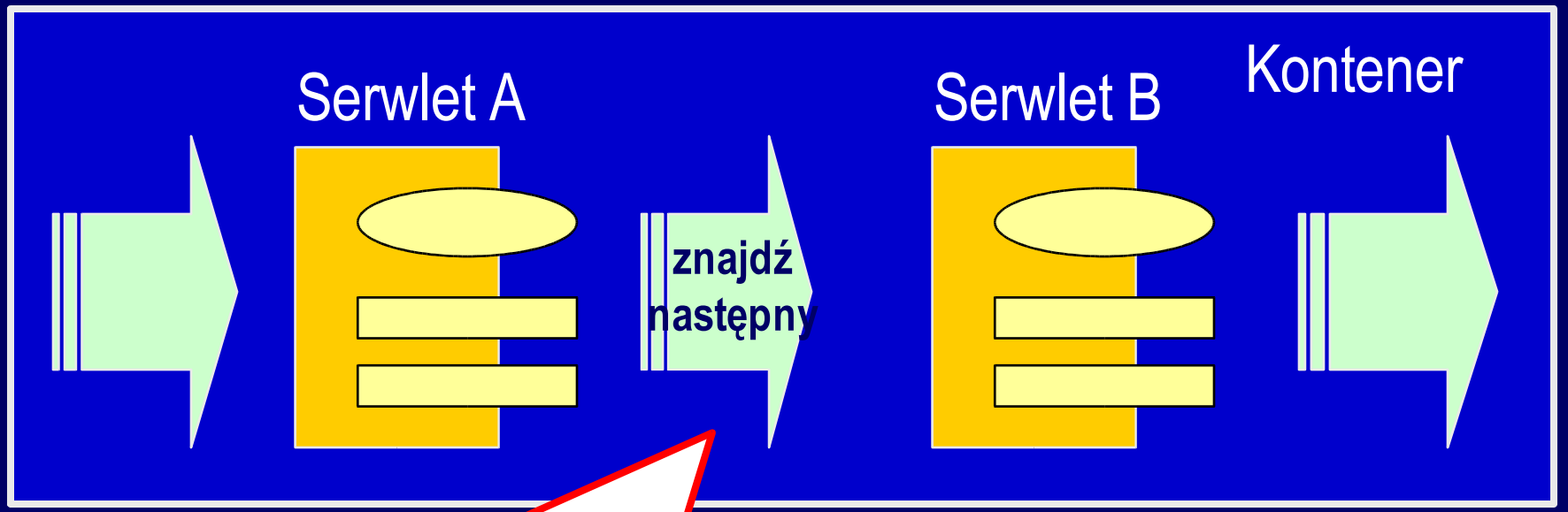
Sposób 2: podanie nazwy serwleta



Komunikacja między serwletami

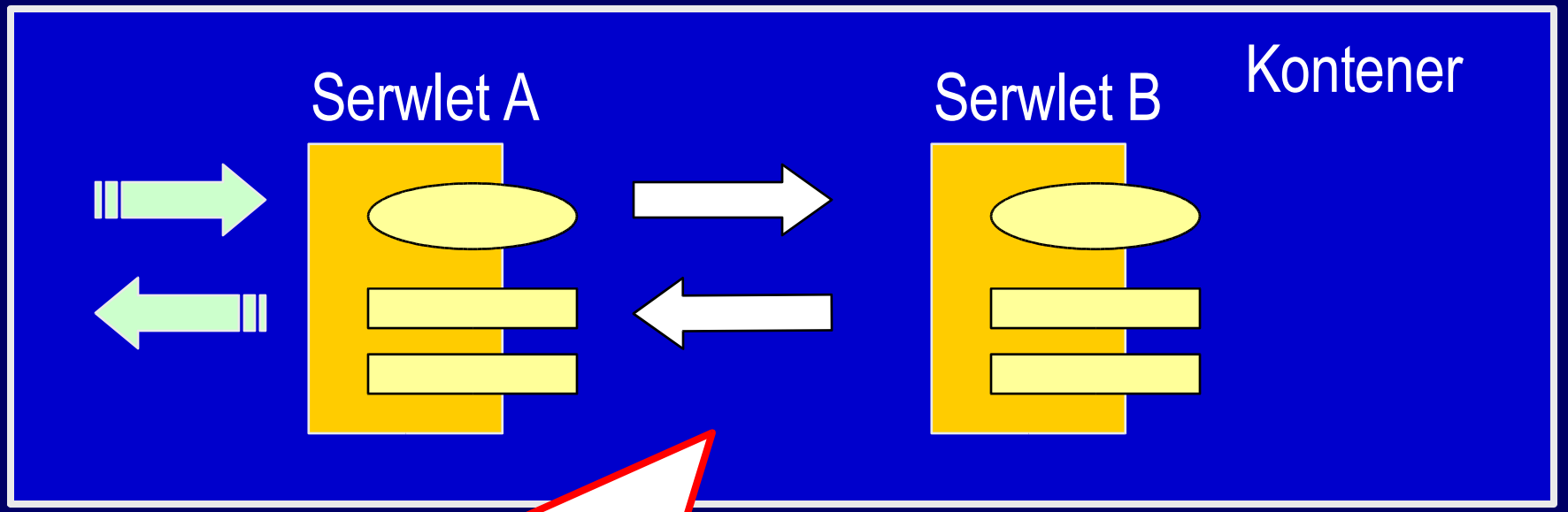


Metody komunikacji: *forwarding*




```
RequestDispatcher rd =  ctx.getRequestDispatcher
("/serwlet_B");

rd.forward(request, response);

// nie ma powrotu do serwleta A
```

Metody komunikacji: *including*

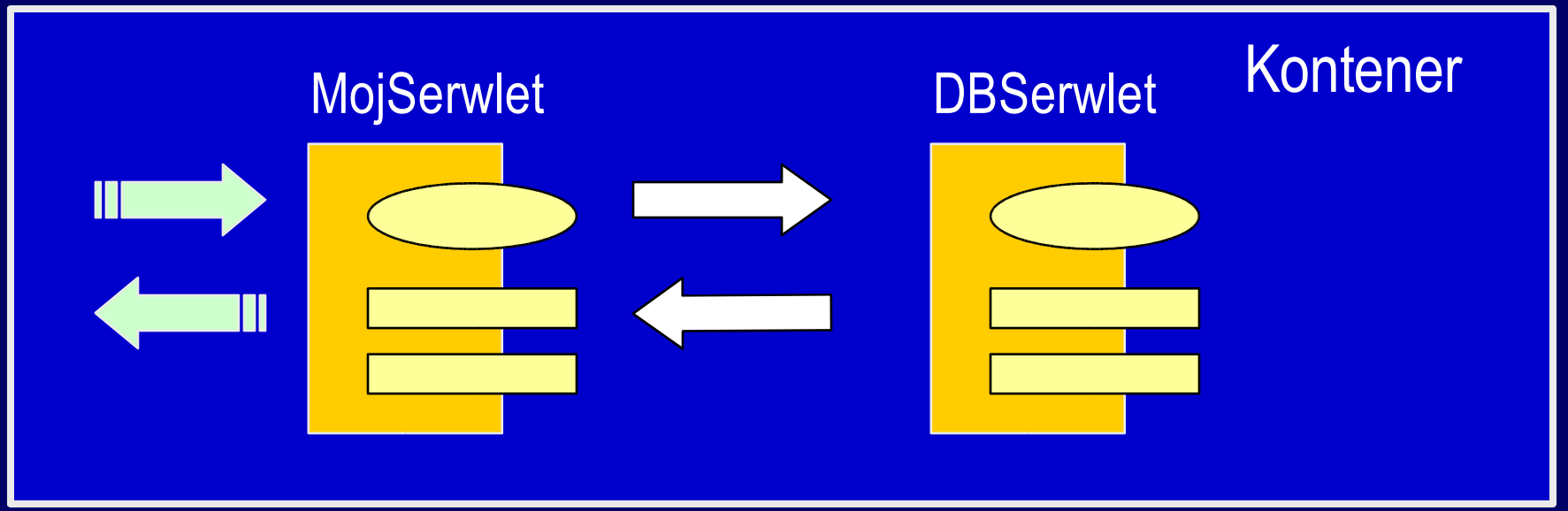



```
RequestDispatcher rd =  ctx.getRequestDispatcher
("/serwlet_B");

rd.include(request, response);

// powrót sterowania do serwleta A
```

Przykład

MojSerwlet – wyświetla stronę, zarządza prezentacją

DBSerwlet – odczytuje z bazy danych imię i nazwisko użytkownika



Wady i zalety

Forwarding

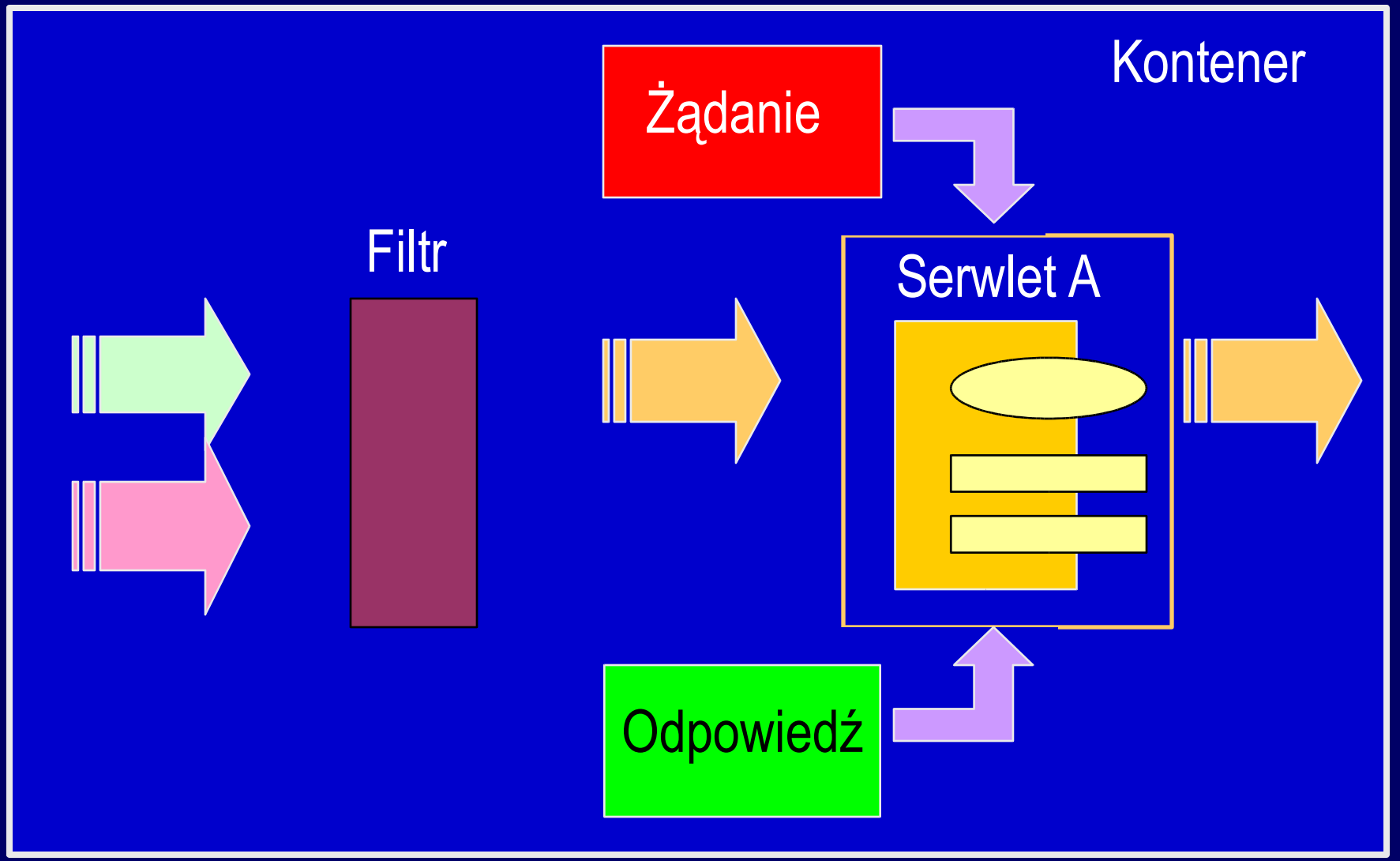
- ✓prosty, potokowy model
- ✓quasi-filtrowanie
- ✓problem z zatwierdzaniem odpowiedzi

Including

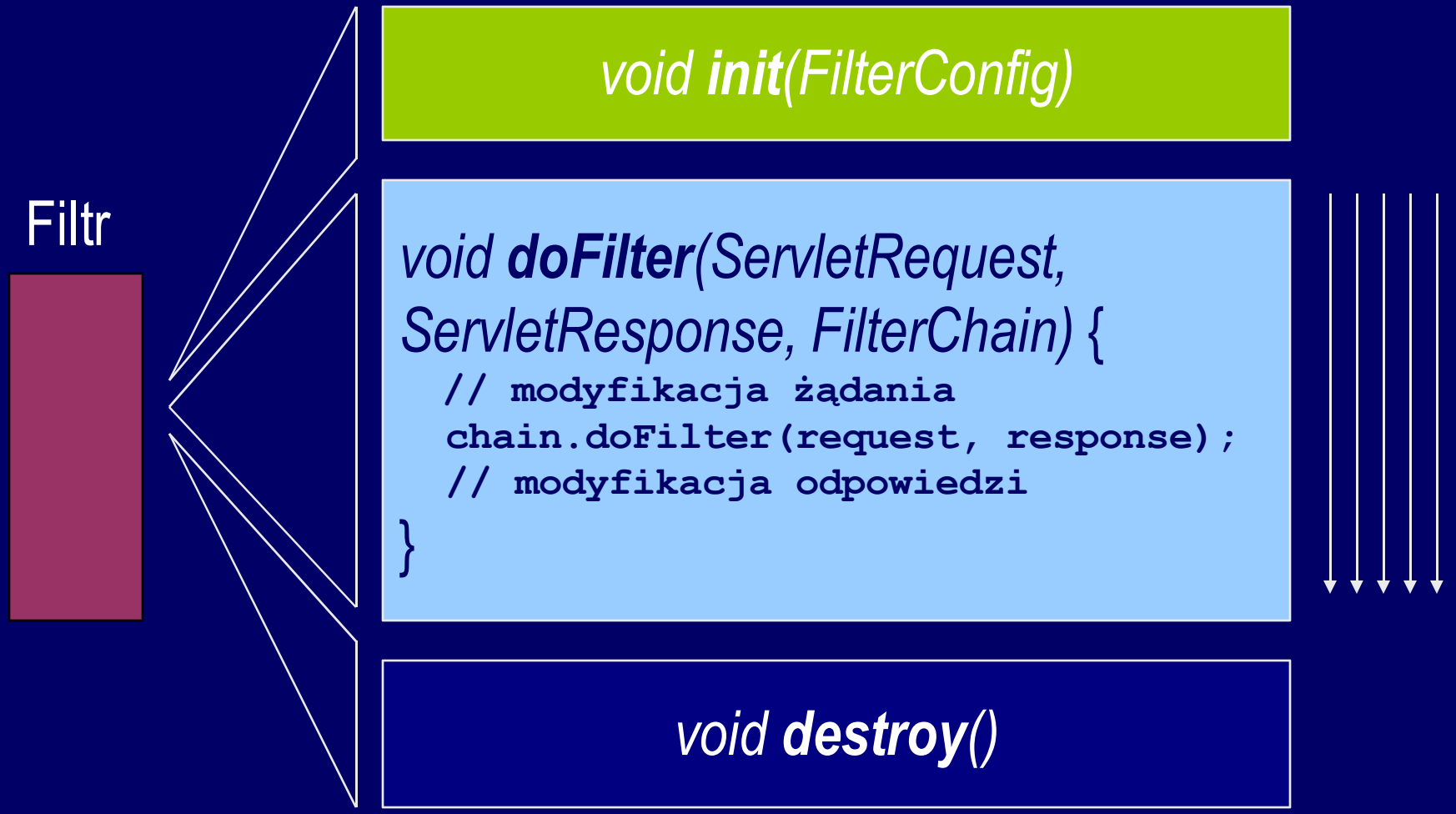
- ✓delegacyjny model przetwarzania
- ✓problem z modularnością

- ✓słaba konfigurowalność
- ✓zmiana w topologii wymaga rekompilacji

Ogólna zasada filtrowania



Budowa filtra



`javax.servlet.Filter`

Konfiguracja filtra

Definicja nazwy filtra

Parametry filtra

Przypisanie do URLa lub serwleta

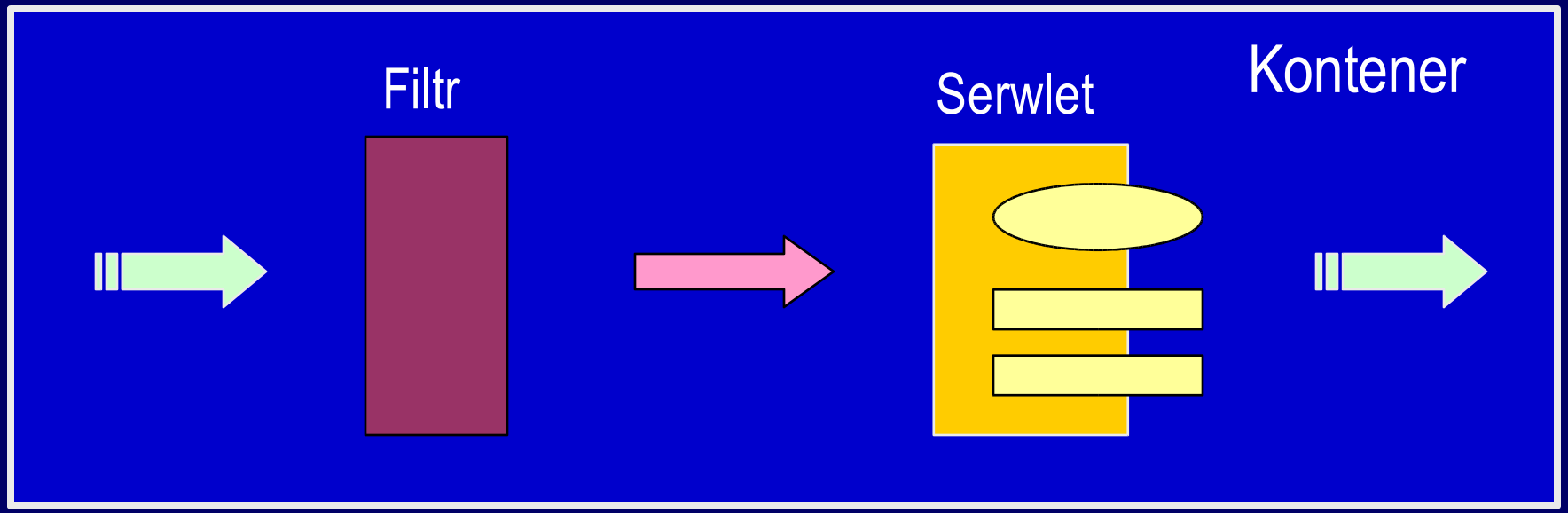
web.xml

```
<filter>
  <filter-name>filter_1</filter-name>
  <filter-class>
    moj.pakiet.Filter1
  </filter-class>
  <init-param>
    <param-name>parametr</param-name>
    <param-value>wartość</param-value>
  </init-param>
</servlet>

<filter-mapping>
  <filter-name>filter_1</filter-name>
  <url-pattern>/A/*</url-pattern>
</servlet-mapping>
```

Przykład

Filtr – zmienia kodowanie żądania HTTP

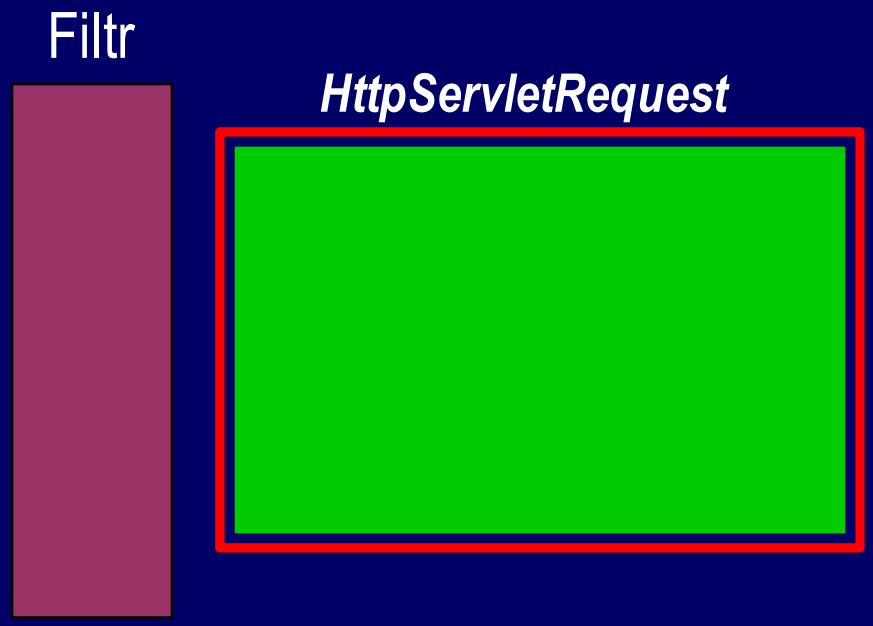
Serwlet – przetwarza żądanie



Filtry i opakowania

Żądanie i odpowiedź są niezmienne!

Zatwierdzonej odpowiedzi nie można modyfikować!



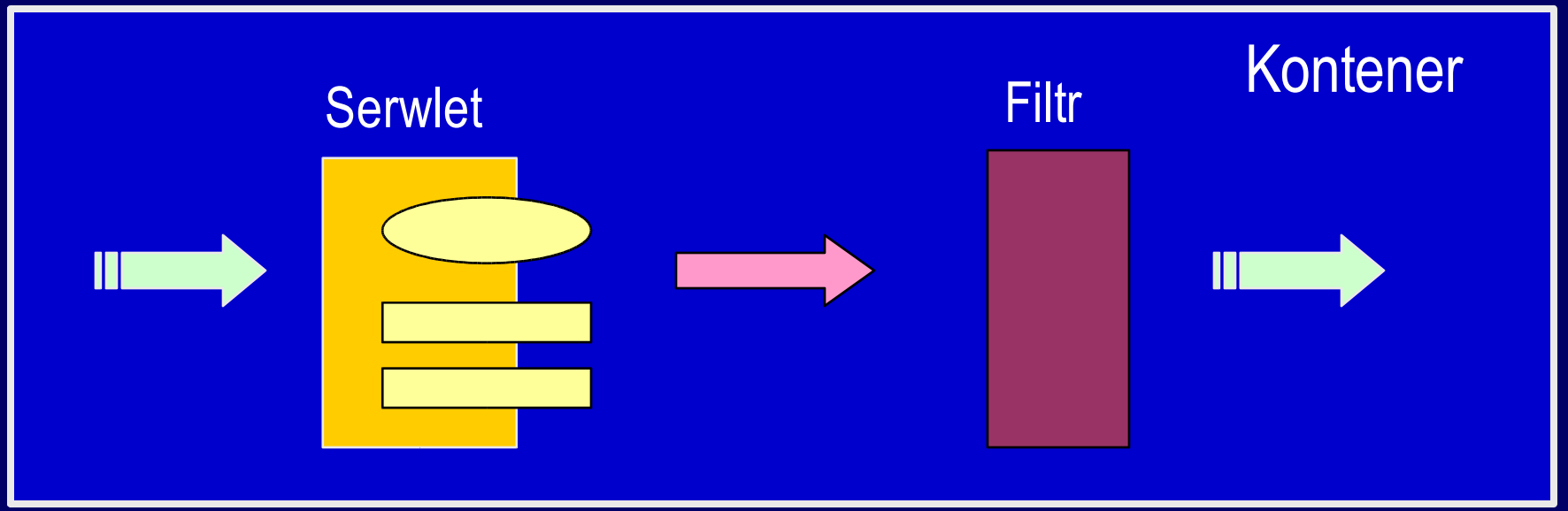
`javax.servlet.HttpServletRequestWrapper`

`javax.servlet.HttpServletResponseWrapper`

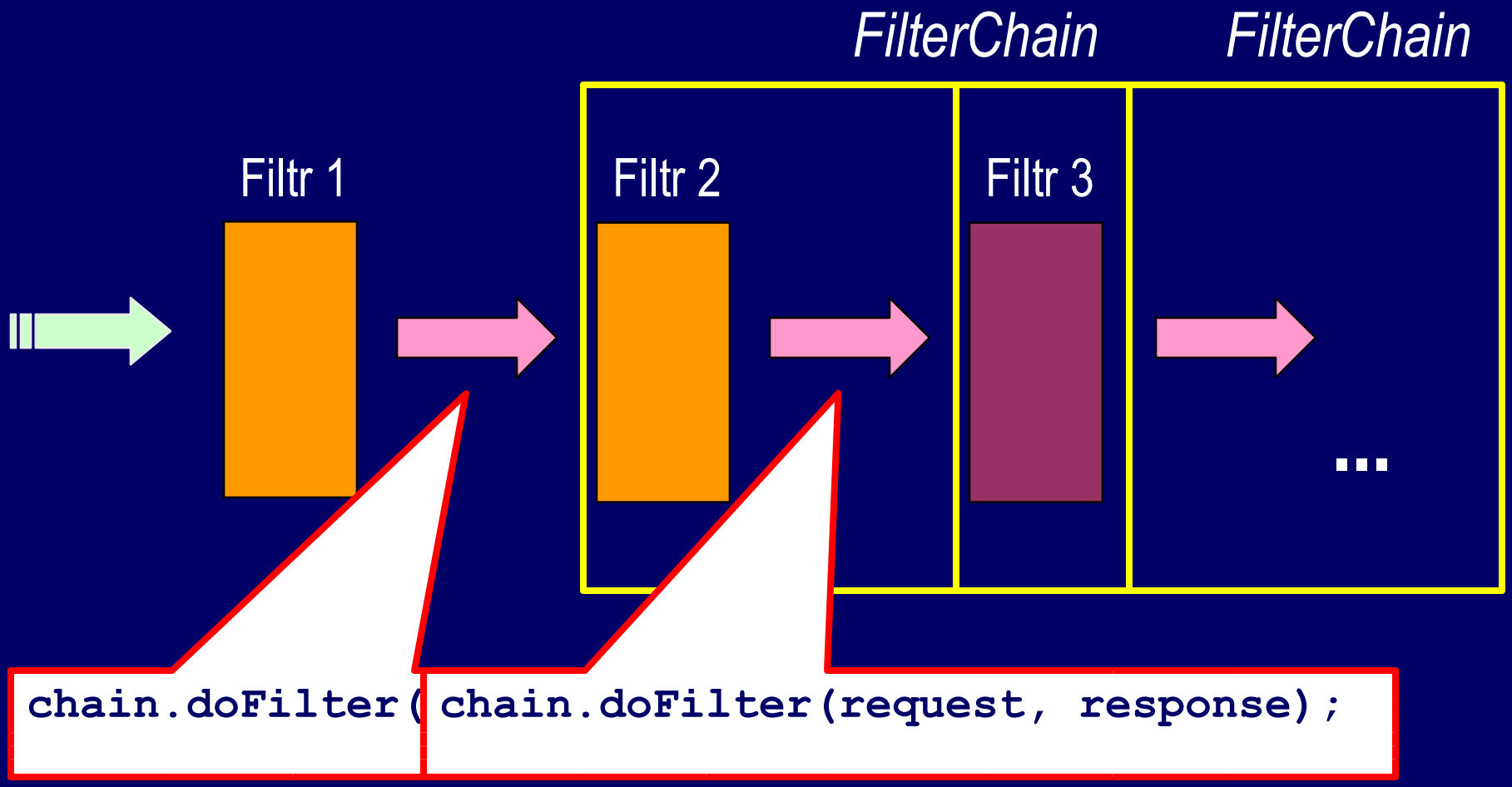
Przykład

Filtr – modyfikuje odpowiedź HTTP

Serwlet – przetwarza żądanie



Łańcuchy filtrów



Konfiguracja łańcuchów

Definicja nazw filtrów 1 i 2

Filtr 1 najpierw...

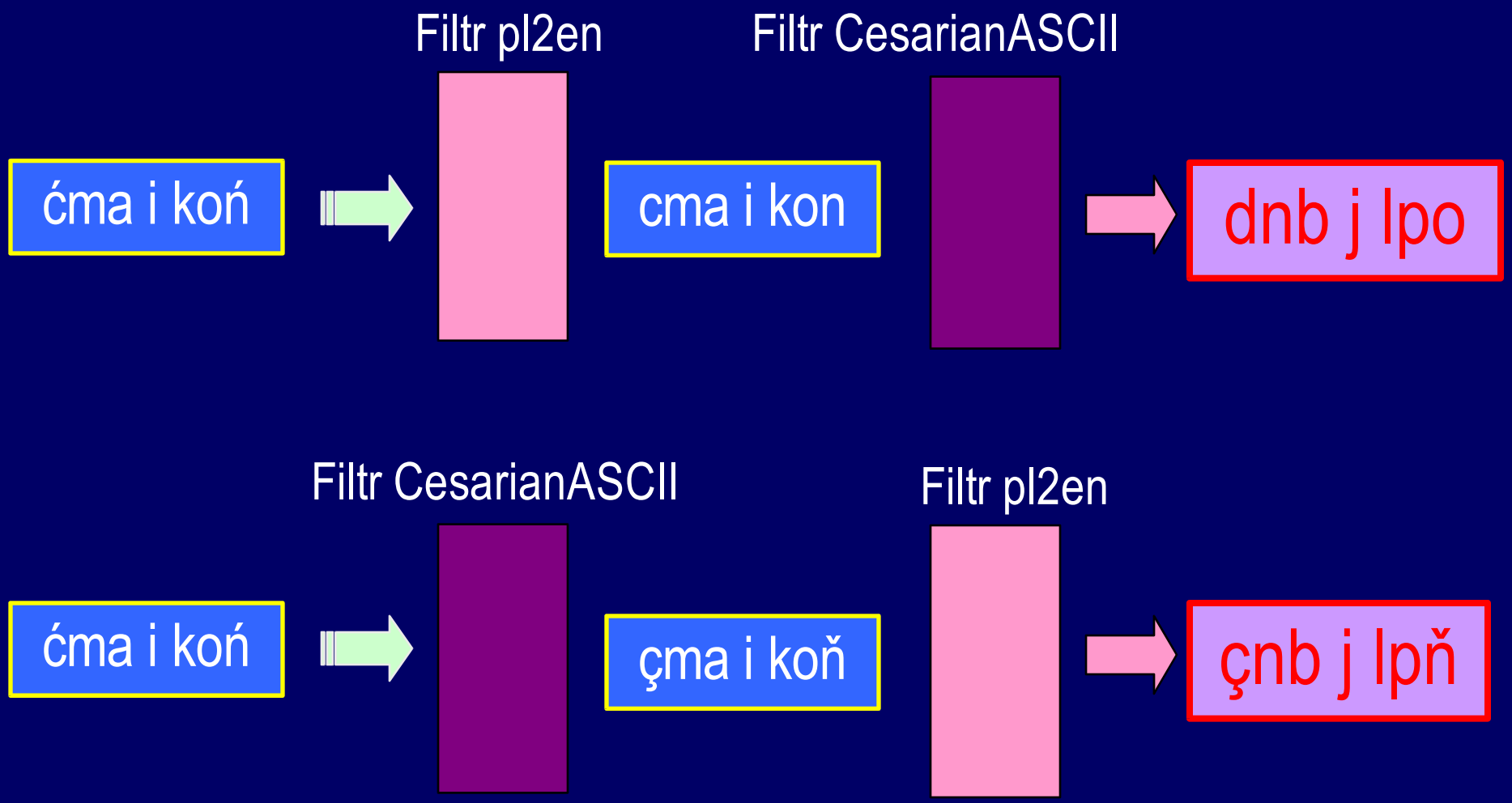
...a potem filtr 2

web.xml

```
<filter>
  <filter-name>filter_1</filter-name>
  ...
</servlet>
<filter>
  <filter-name>filter_2</filter-name>
  ...
</servlet>

<filter-mapping>
  <filter-name>filter_1</filter-name>
  <filter-name>filter_2</filter-name>
  <url-pattern>/A/*</url-pattern>
</servlet-mapping>
```

Kolejność filtrów

A co jeśli zmienimy kolejność filtrów?



Podsumowanie

- Droga serwletów do aplikacji klasy *enterprise*
- Większa elastyczność dzięki filtrom
- Konfiguracja przez administratora bez rekompilacji
- Pełna modularyzacja
- Przetwarzanie potokowe i zagnieżdżone

Pytania?