KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

POLITECHNIKA BIAŁOSTOCKA

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Rozproszone systemy internetowe

Enterprise Java Beans: przykłady.

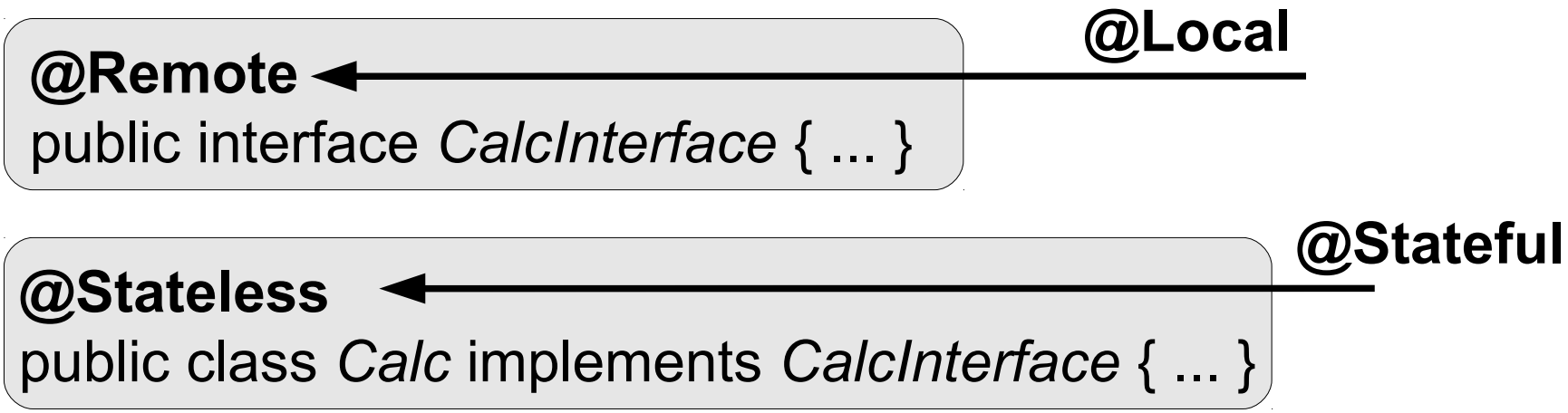
„Podniesienie potencjału uczelni wyższych jako czynnik rozwoju gospodarki opartej na wiedzy”

Nr projektu: UDA-POKL.04.01.01-00-143/09-00

Testy

- Różnica pomiędzy komponentem *Stateful* a *Stateless* (wartość pola obiektu)
- Protokoły sieciowe używane w trakcie wywołania (EJBD vs IIOP) – osadzenie komponentu na platformie Glassfish
- Obserwacja cyklu życia komponentu sesyjnego typu *Stateful* i *Stateless*
- Usunięcie komponentu typu *Stateful* - @Remove
- Interceptors – metoda wewnątrz obiektu oraz specjalizowane klasy
- Usługa „czasomierza”
- Komponenty reagujące na komunikaty - MDB

Komponent sesyjny (Open EJB)



Ustawienia dla kontekstu klienta

java.naming.factory.initial:
org.apache.openejb.client.RemoteInitialContextFactory
org.apache.openejb.client.RemoteInitialContextFactory
java.naming.provider.url: ejbd://localhost:4201

Glassfish

Ustawienia dla kontekstu klienta

java.naming.factory.initial:
com.sun.enterprise.naming.SerialInitContextFactory
org.omg.CORBA.ORBInitialHost: ...
org.omg.CORBA.ORBInitialPort: 3700

Nazwy obiektów:

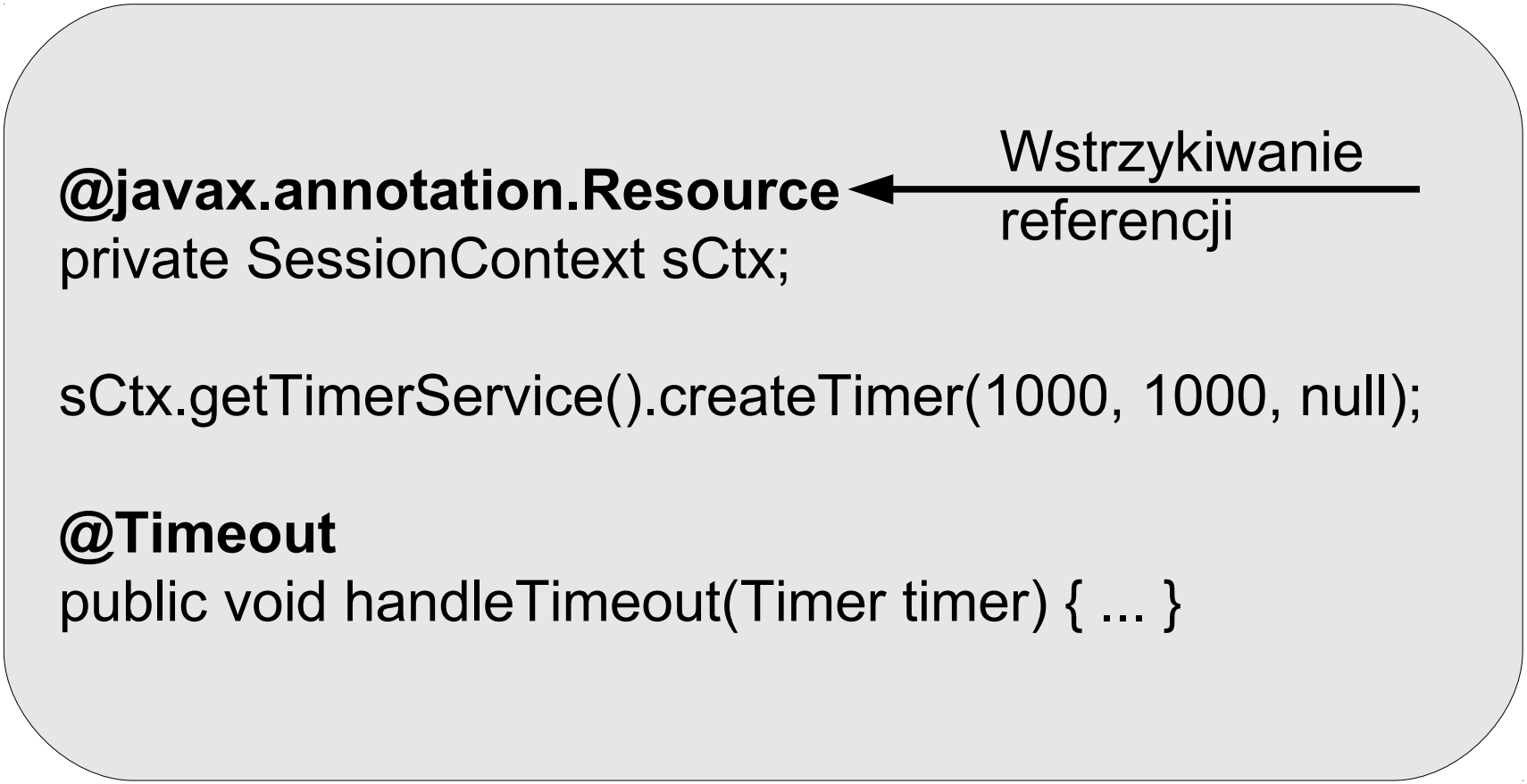
- V2 – mappedName lub pełna nazwa interfejsu zdalnego
- V3 – zgodnie z EJB 3.1
  - *java:global[/<app-name>]/<module-name>/<bean-name>*

Timer Service

```
public interface javax.ejb.TimerService {

public Timer createTimer(long duration,java.io.Serializable info);
public Timer createTimer(long initialDuration,
                         long intervalDuration,
                         java.io.Serializable info);
public Timer createTimer(java.util.Date expiration,
                         java.io.Serializable info);
public Timer createTimer(java.util.Date initialExpiration,
                         long intervalDuration,
                         java.io.Serializable info);
public Collection getTimers();
}
```

Wykorzystanie czasomierza Uwaga! Tylko typ *Stateless*




```
@javax.annotation.Resource
private SessionContext sCtx;

sCtx.getTimerService().createTimer(1000, 1000, null);

@Timeout
public void handleTimeout(Timer timer) { ... }
```

Message Driven Bean

Nazwa kolejki komunikatów w kontenerze

Interfejs specyficzny dla danego systemu komunikatów

```
@MessageDriven(mappedName = "Kolejka")
public class MDB implements javax.jms.MessageListener {

 @Resource
  private MessageDrivenContext mdc;

public void onMessage(Message inMessage) { ... }

}
```

Message Driven Bean - klient

```
Context ctx = new InitialContext(...);
ConnectionFactory connectionFactory =
        (ConnectionFactory)ctx.lookup("...");
Queue queue = (Queue)ctx.lookup("...");
javax.jms.Connection connection =
                connectionFactory.createConnection();
javax.jms.Session session =
connection.createSession(false,
                        Session.AUTO_ACKNOWLEDGE);
MessageProducer messageProducer =
                session.createProducer(queue);
TextMessage message = session.createTextMessage();
message.setText("...");
messageProducer.send(message);
connection.close();
```