



NOVINKY V JEE – EJB 3.1

Zdeněk Troníček

Fakulta informačních technologií

ČVUT v Praze

PROGRAM

- Seznámení s Java Enterprise Edition (JEE)
- Enterprise Java Beans (EJB)
- Novinky v EJB 3.1

JAVA EDITIONS



J2ME

Java Micro Edition



JAVA™

Java Standard Edition



GlassFish

Java Enterprise Edition

MOTIVACE

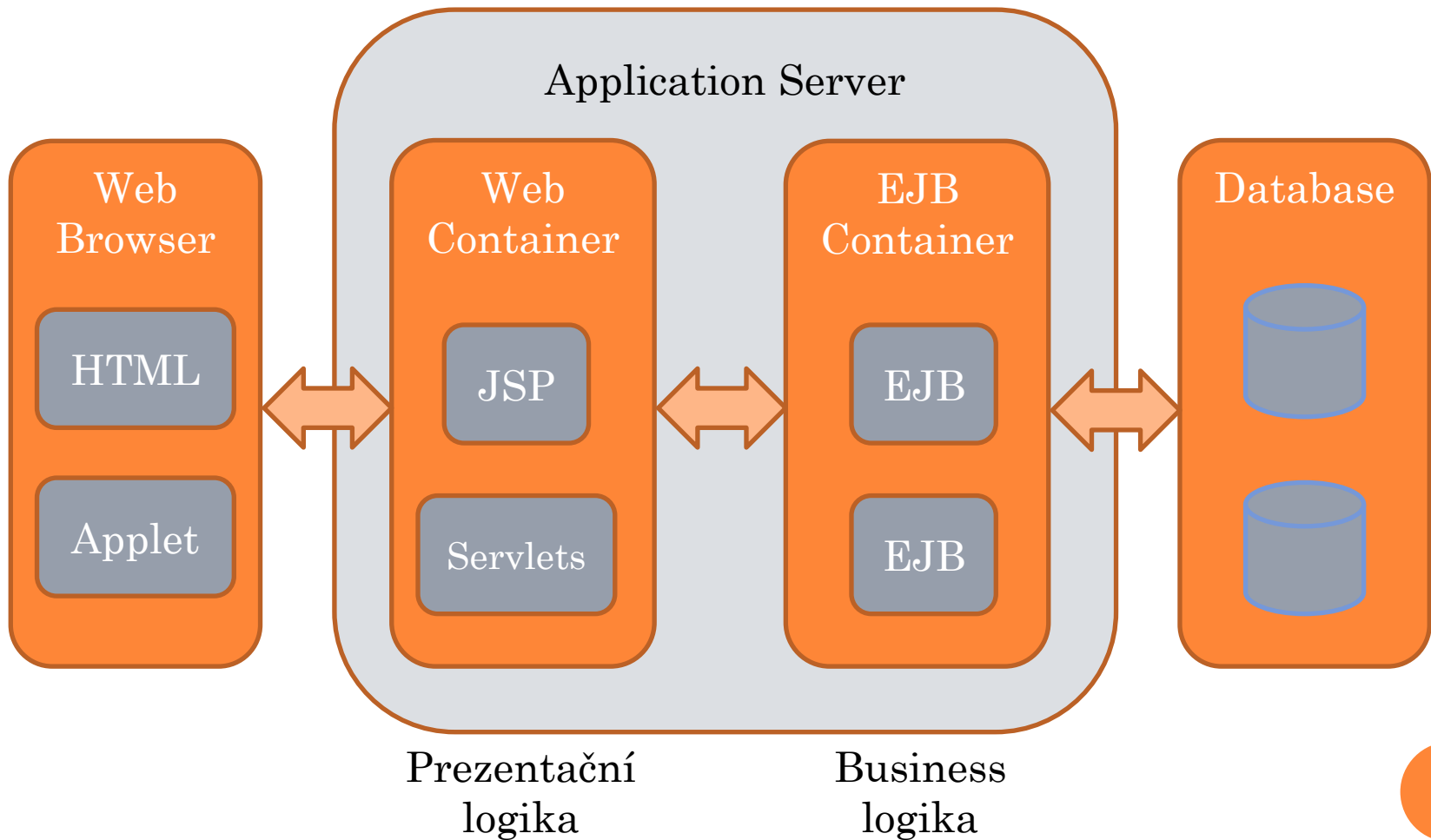
Co mají podnikové aplikace společného?

- User authentication
- Multi-user support
- Data persistence
- Data integrity (transactions)
- Client-tier communication
- Asynchronous communication
- Naming service
- Communication with legacy systems

JAVA ENTERPRISE EDITION 6

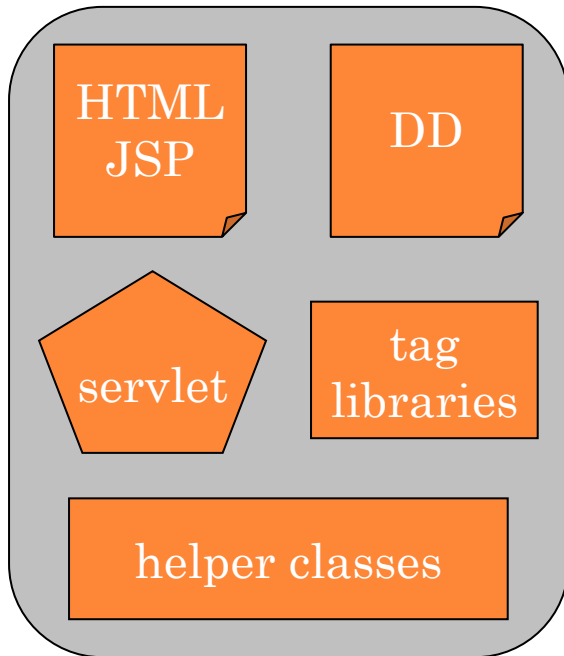
- Java Server Faces (JSF) 2.0
- Java Server Pages (JSP) 2.2
- Enterprise Java Beans (EJB) 3.1
- Java Persistence API (JPA) 2.0
- Java API for XML-Based Web Services (JAX-WS) 2.2
- Java API for RESTful Web Services (JAX-RS) 1.1
- Java Architecture for XML Binding (JAXB) 2.2
- Java Message Service API (JMS) 1.1
- Java Transaction API (JTA) 1.1
- ...

APLIKACE V JEE

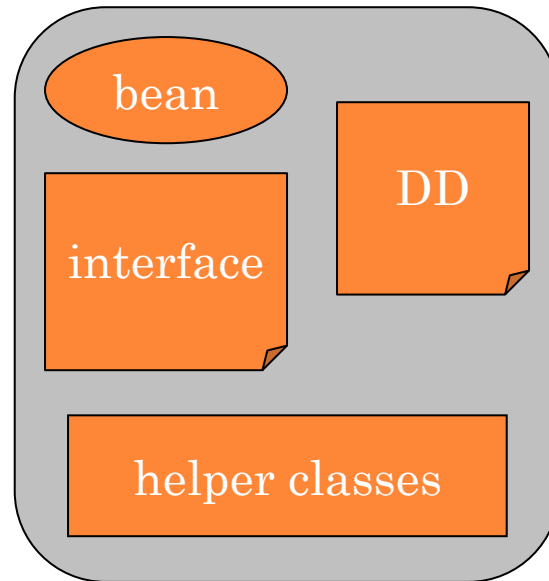


ARCHIVE FILES

Web Archive
(WAR)

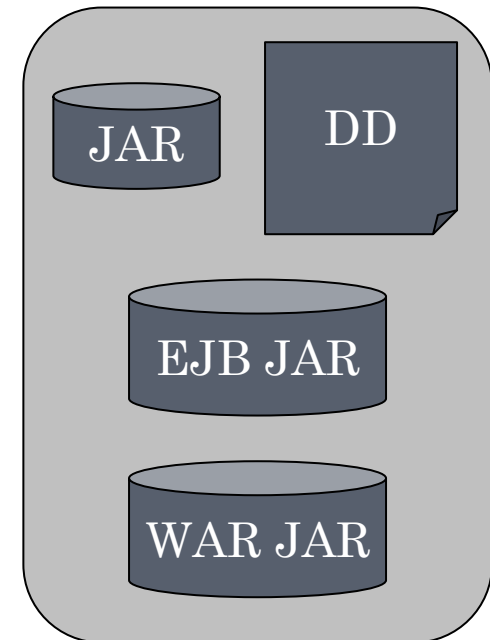


EJB Archive
(JAR)

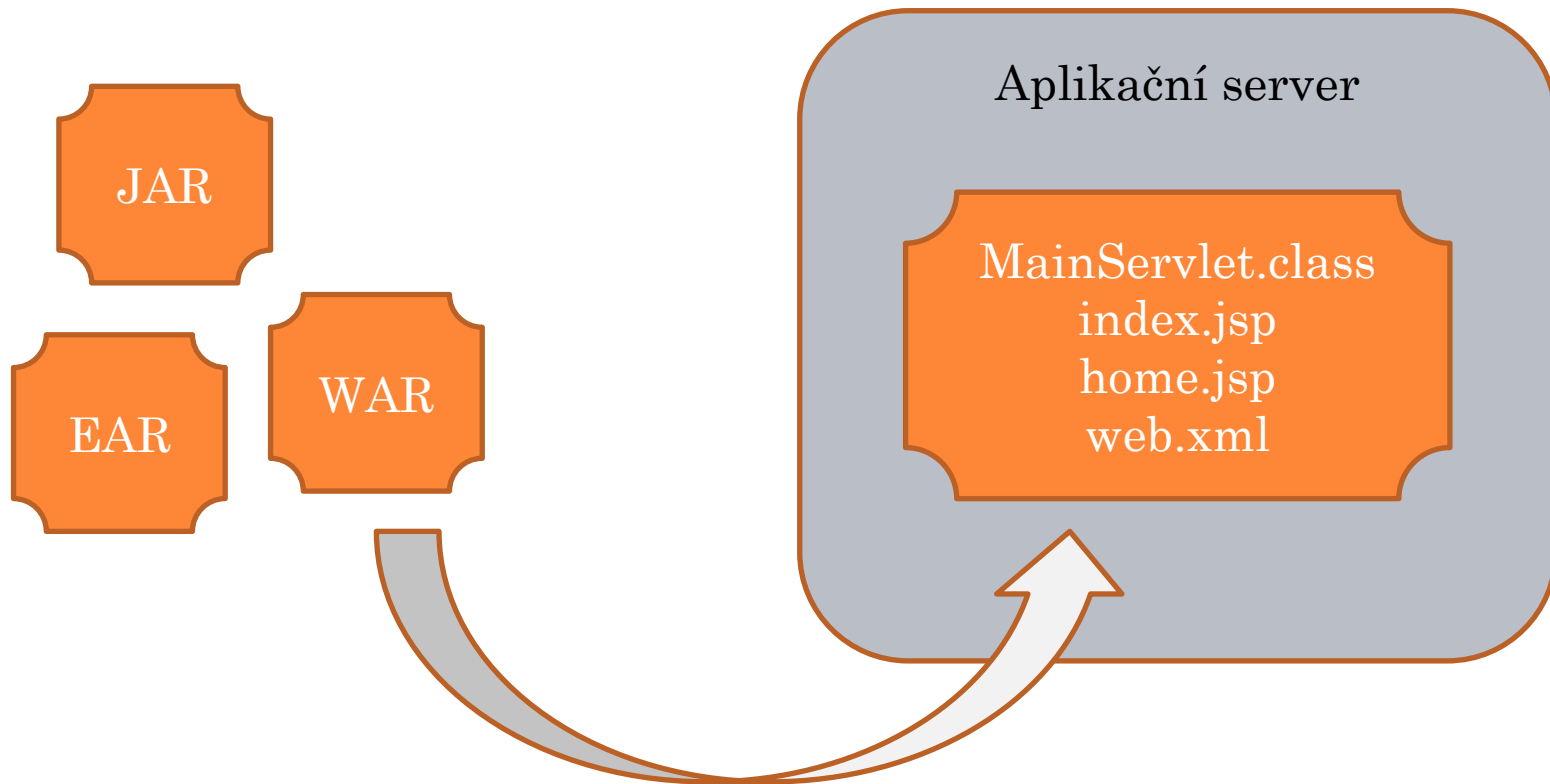


DD = Deployment
Descriptor

Enterprise
Archive
(EAR)



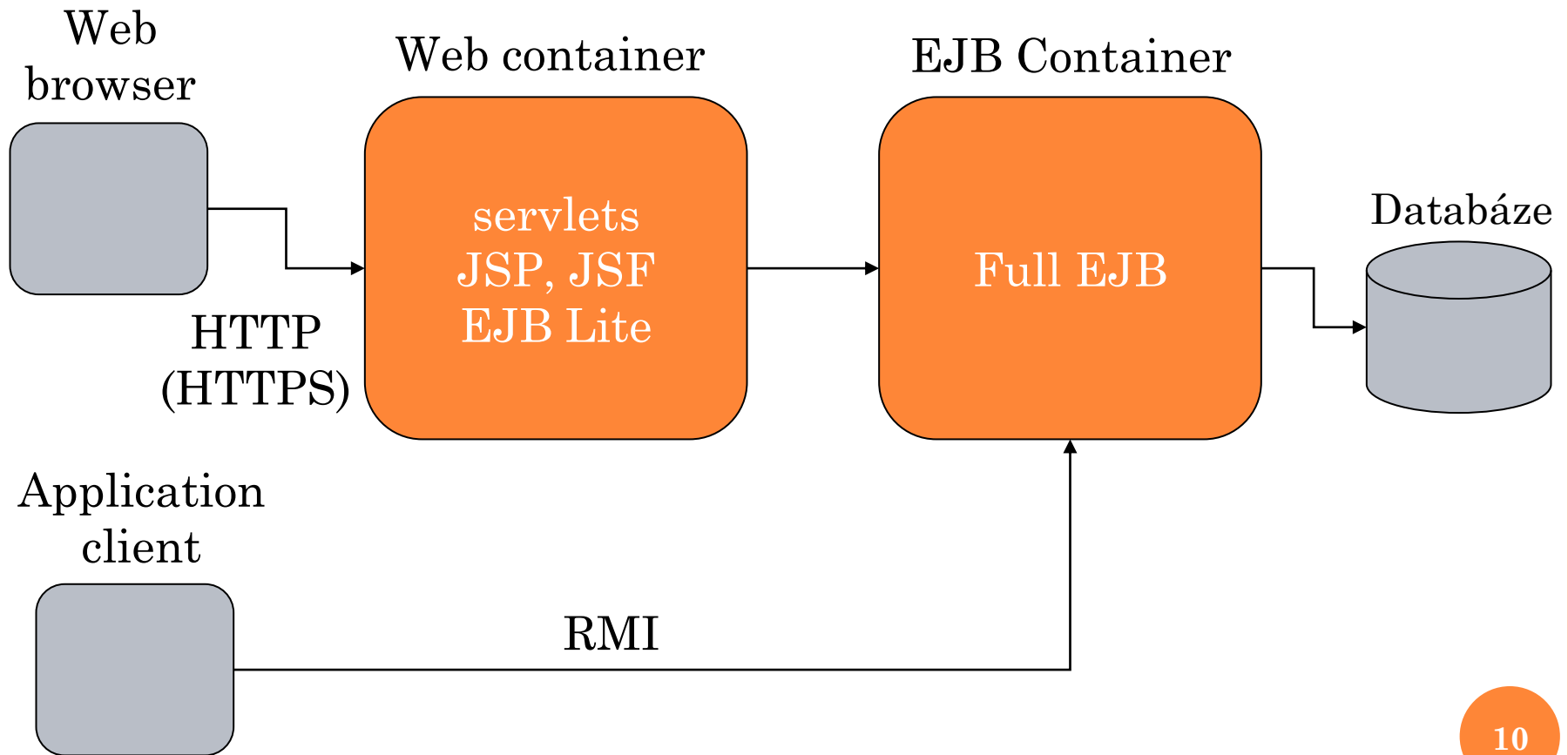
DEPLOYMENT



SLUŽBY POSKYTOVANÉ AS

- Komunikace s klientem
- Životní cyklus komponent (zahrnuje i řízení souběžného přístupu)
- Správa databázových spojení (Connection pool)
- Transakční zpracování
- Persistence objektů
- Asynchronní komunikace
- Přihlašování uživatelů a přidělování práv
- ...

CONTAINERS

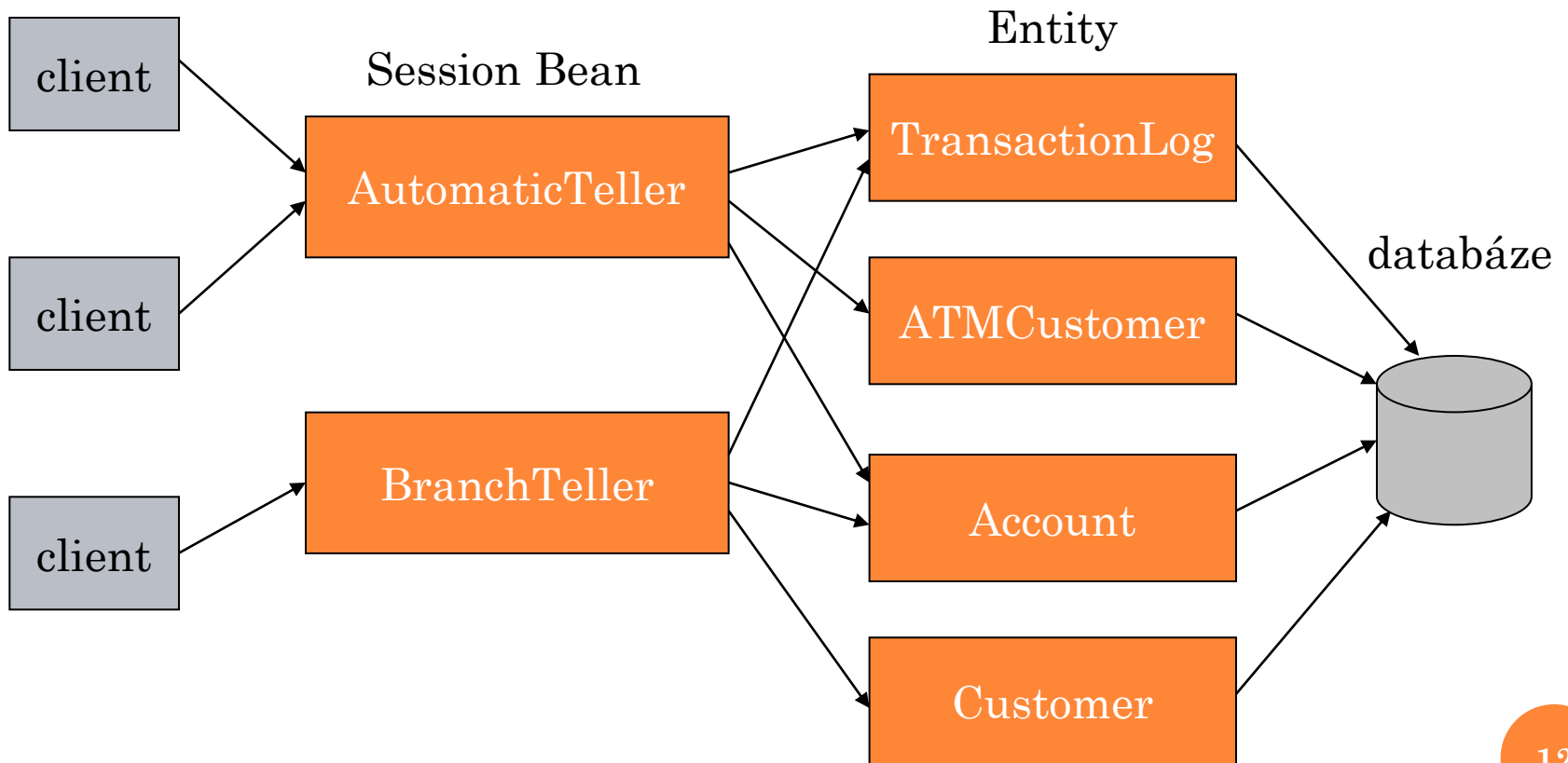


ENTERPRISE JAVA BEANS 3.1

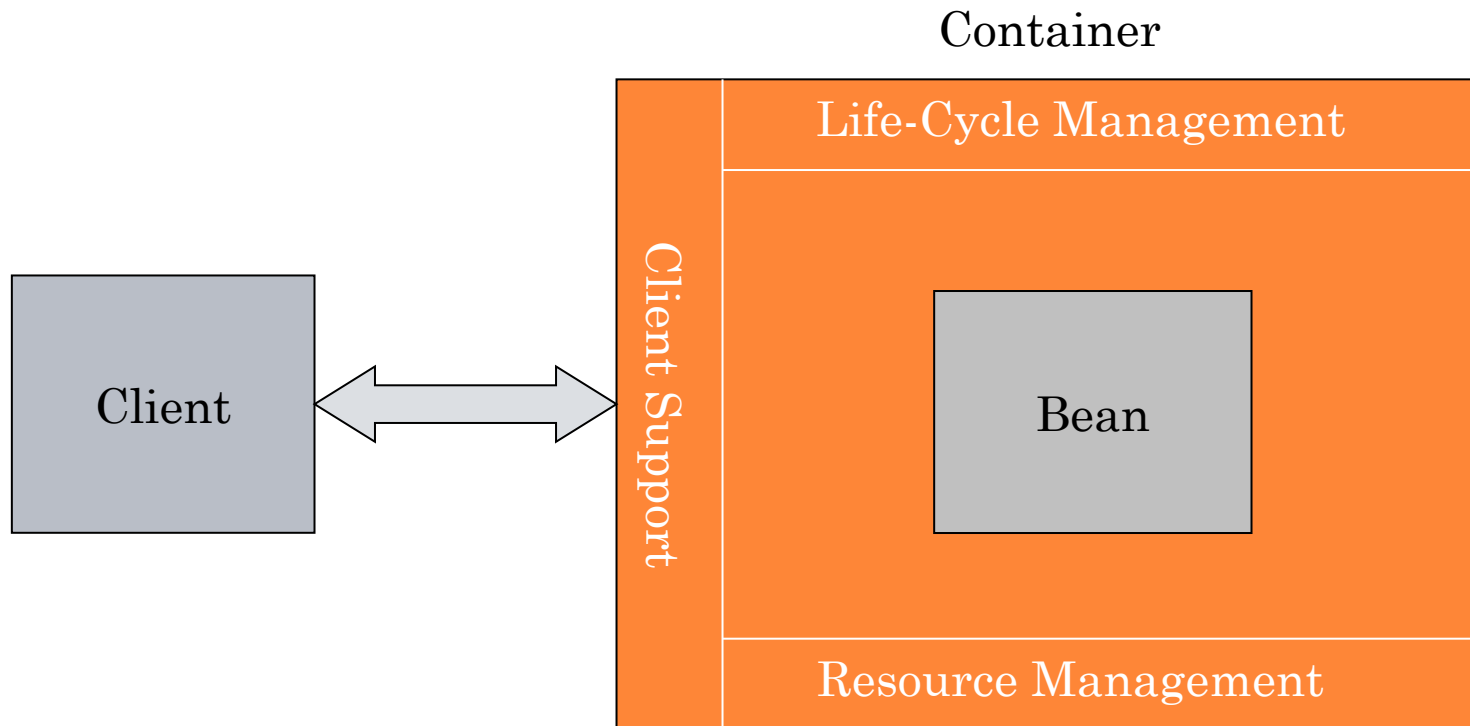
- Session Beans
- (Entity Beans 2.1)
- Message Driven Beans

Dependency Injection – kontejner se postará o nastavení odkazu např. na jinou beanu

EJB APPLICATION



EJB CONTAINER



SESSION EJB

- Plain Old Java Object (POJO)
- bezstavové (@Stateless)
- stavové (@Stateful)
- singleton (@Singleton) (EJB 3.1)
- remote a local business interface
- no interface (EJB 3.1)

IMPLEMENTACE

Business Interface

```
import javax.ejb.Local;  
  
@Local  
public interface HelloLocal {  
    String sayHello();  
}
```

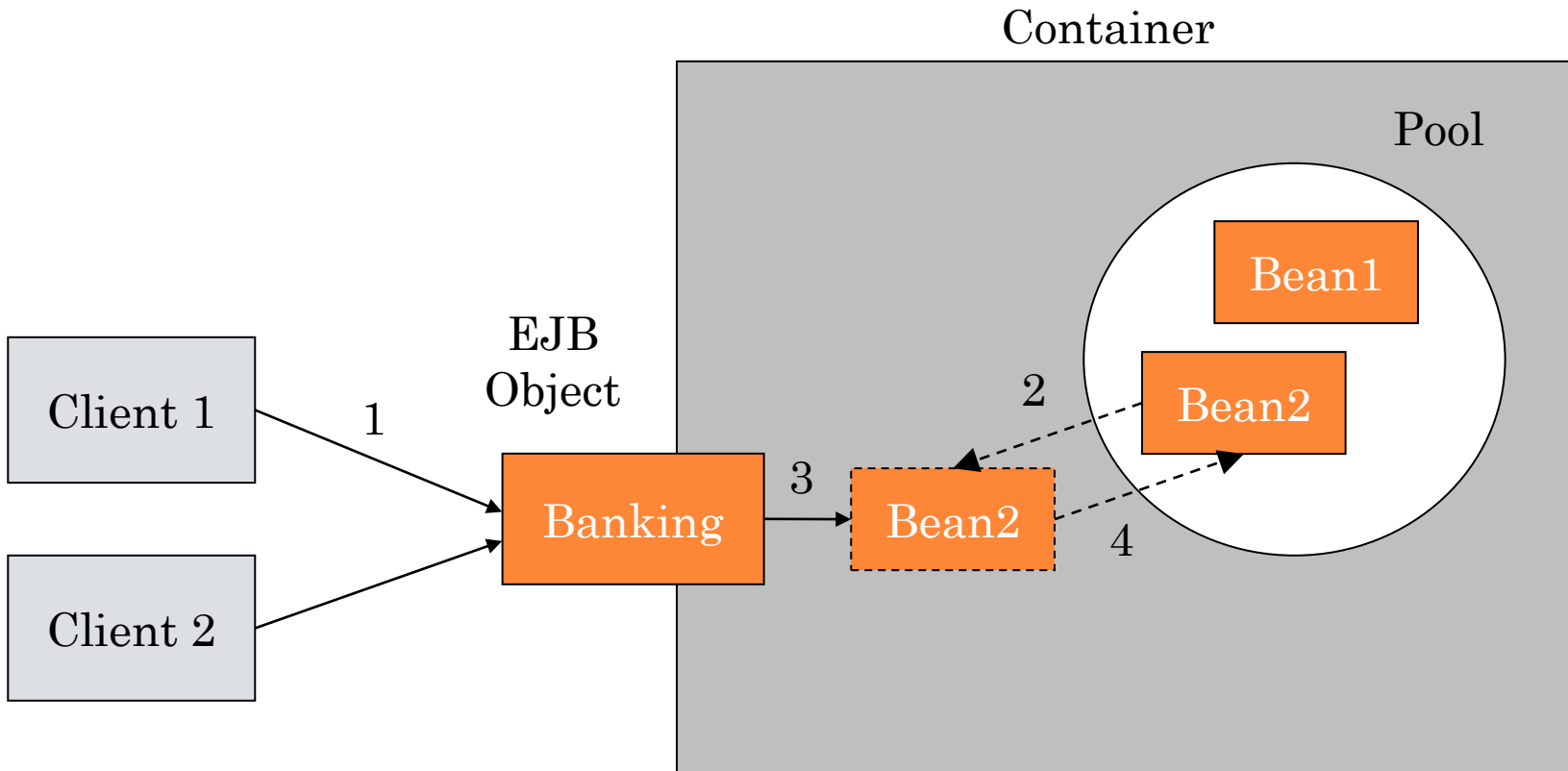
@Local
@Remote

Bean

```
import javax.ejb.Stateless;  
  
@Stateless  
public class HelloBean implements HelloLocal {  
    public String sayHello() {  
        return "Hello!";  
    }  
}
```

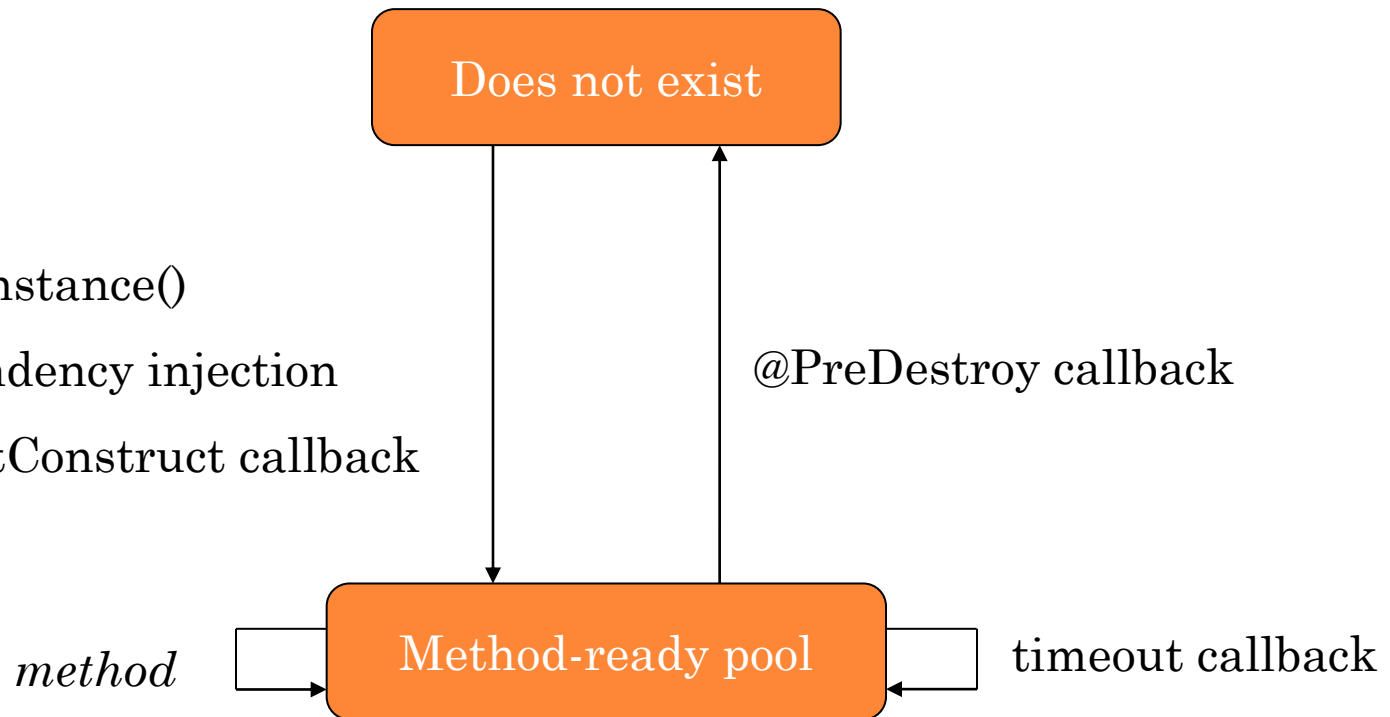
@Stateless
@Stateful

STATELESS SESSION EJB

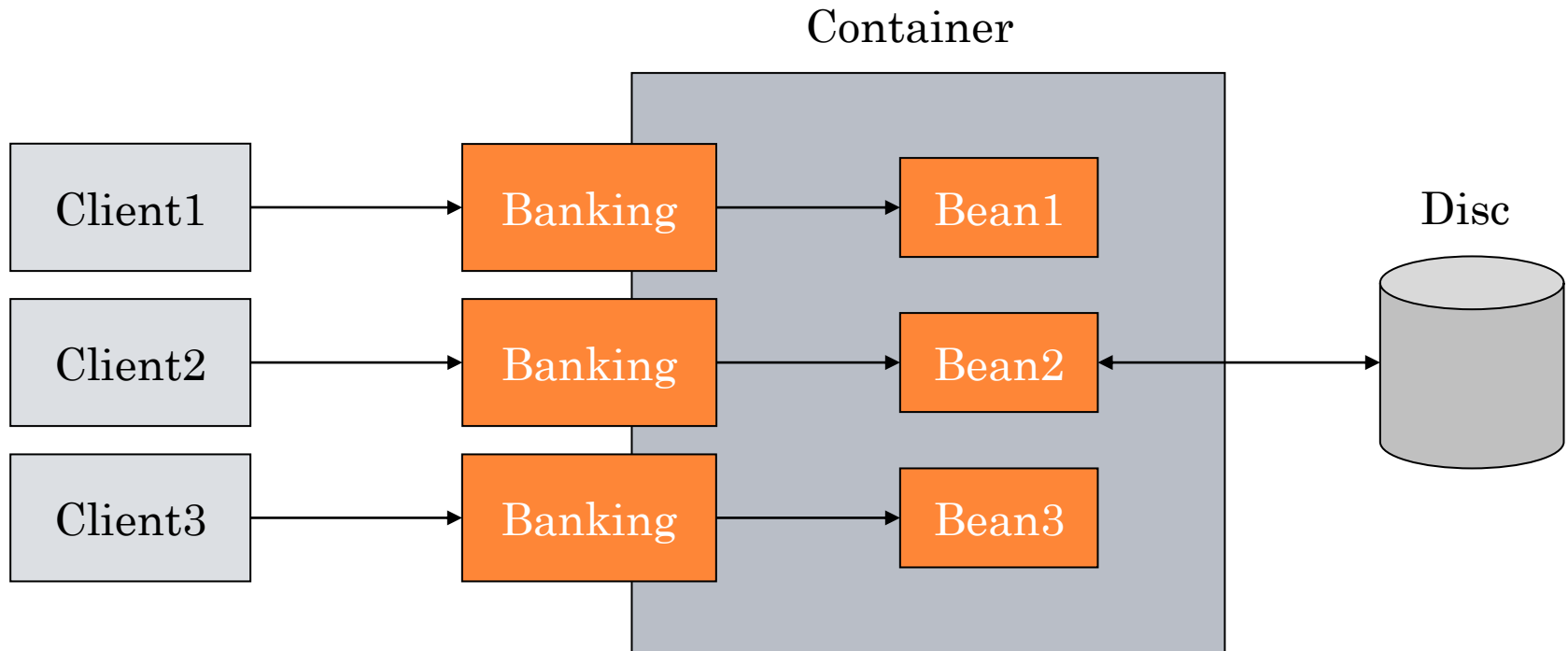


STATELESS SESSION EJB LIFE CYCLE

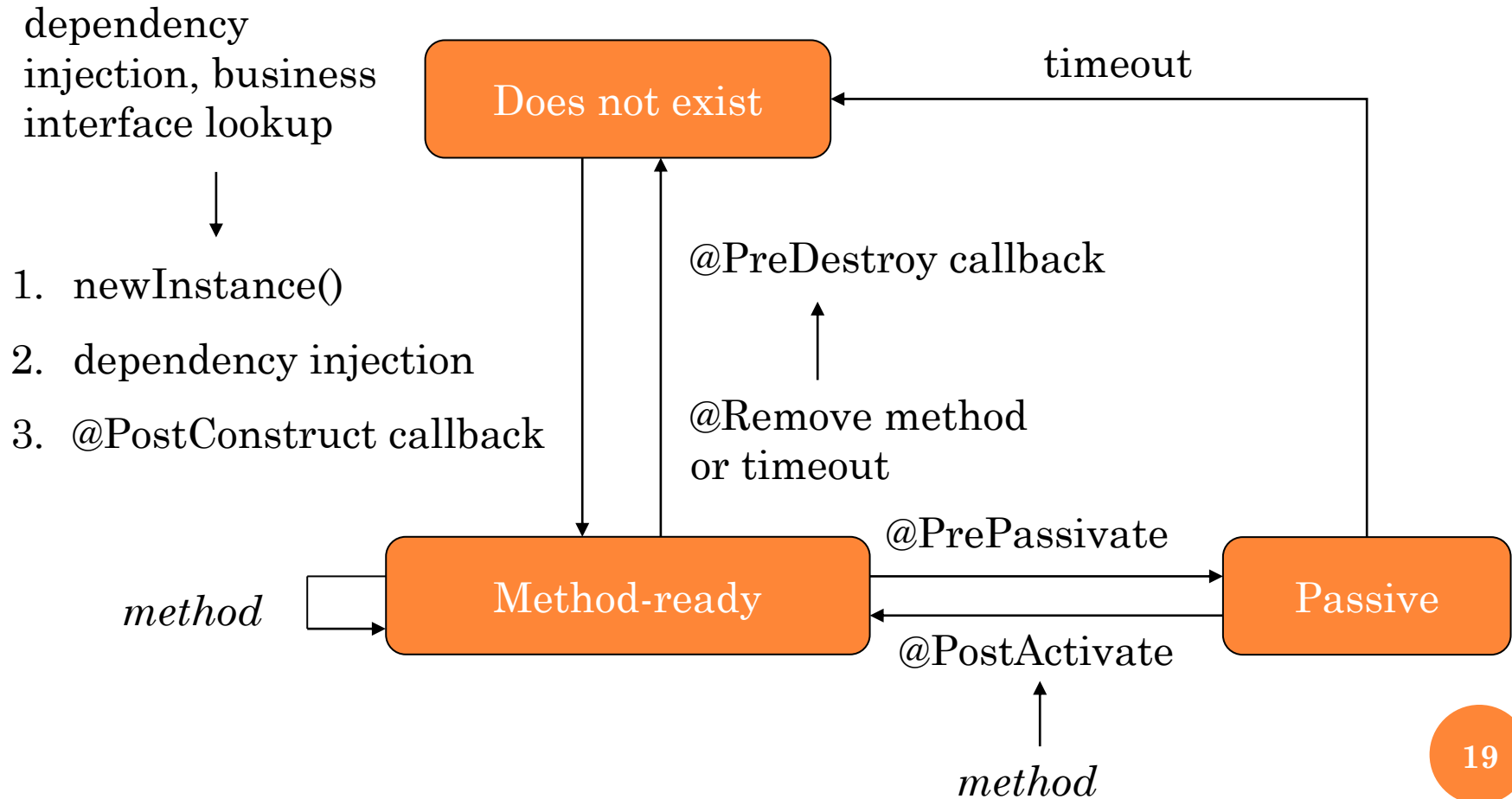
1. `newInstance()`
2. dependency injection
3. `@PostConstruct` callback



STATEFUL SESSION EJB



STATEFUL SESSION EJB LIFE CYCLE



NOVINKY V EJB 3.1

- No-interface view
- Global JNDI names
- Singleton session beans
- Calendar based timer service
- Asynchronous invocations
- EJB lite
- Simplified packaging
- Embeddable container

NO-INTERFACE VIEW

- EJB nemusí mít business interface
- Pokud EJB nemá business interface, klient vidí všechny public metody

```
@Stateless
public class HelloSessionBean {

    public String sayHello(String name) { ... }
}
```

```
@EJB
private HelloSessionBean helloBean;
```

GLOBAL JNDI NAMES

- `java:global[/<app-name>]/<module-name>/<bean-name>[!<fully-qualified-interface-name>]`
- `java:app/<module-name>/<bean-name>[!<fully-qualified-interface-name>]`
- `java:module/<bean-name>[!<fully-qualified-interface-name>]`

```
java:global/fooweb/FooBean
java:global/fooweb/FooBean!com.acme.Foo
java:app/fooweb/FooBean
java:app/fooweb/FooBean!com.acme.Foo
java:module/FooBean
java:module/FooBean!com.acme.Foo
```

SINGLETON SESSION BEANS

- Pouze jedna instance
- Lze stanovit pořadí inicializací (@DependsOn)
- Inicializace může být *eager* (@Startup)

```
@Startup
@Singleton( name = "A" )
@DependsOn( "B" )
public class ASessionBean { ... }
```

```
@Startup
@Singleton( name = "B" )
public class BSessionBean { ... }
```

CONCURRENCY MANAGEMENT (1)

@Singleton

```
//@ConcurrencyManagement(  
//    ConcurrencyManagementType.CONTAINER )  
public class SingletonSessionBean {
```

@Lock(LockType.READ)

```
public void doSomething() { ... }
```

@Lock(LockType.WRITE)

```
public void doSomethingElse() { ... }
```

```
}
```


CONCURRENCY MANAGEMENT (2)

@Singleton

**@ConcurrencyManagement(
 ConcurrencyManagementType.BEAN)**

```
public class SingletonSessionBean {  
  
    public void doSomething() {  
        ...  
        synchronized(this) { ... }  
    }  
    public synchronized void doSomethingElse() { ... }  
}
```

CALENDAR BASED TIMER SERVICE

Anotace @Schedule s atributy:

- year, month, dayOfMonth, dayOfWeek
- hour, minute, second
- timezone

```
@Singleton
public class ServiceBean {

    @Schedule( dayOfWeek = "Sun", hour = "2", minute = "30" )
    public void cleanDatabase() { ... }

}
```

ASYNCHRONOUS INVOCATIONS

- Metodu EJB lze volat asynchronně
- Návratovou hodnotou asynchronní metody je Future

```
@Stateless
public class MathSessionBean {
    @Asynchronous
    public Future<Integer> compute( Integer x, Integer y ) {
        Integer z = ...
        return new AsyncResult( z );
    }
}
```

```
Future<Integer> r = mathBean.compute( 20, 11 );
while ( !r.isDone() ) { ... }
Integer i = r.get();
```

EJB LITE

Podmnožina EJB Full:

- No Message Driven Beans
- No remote interfaces
- No EJB timers and scheduling
- No asynchronous invocation
- No web services

EJB Lite mohou být součástí souboru .war

EMBEDDABLE CONTAINER

- Umožňuje programově spustit kontejner
- Vhodné pro testování

```
@Test
public void hello() throws Exception {
    EJBContainer ec = EJBContainer.createEJBContainer();
    Context c = ec.getContext();
    HelloSessionBean hello = (HelloSessionBean)
        c.lookup( "java:global/classes/HelloSessionBean" );
    String s = hello.sayHello( "Eva" );
    assertEquals( "Hello, Eva", s );
}
```



OTÁZKY A ODPOVĚDI

Děkuji za pozornost