

Transactions

Spring 
OSCON 2005

Transactions Overview

- Transactions apply to databases and other resources like JMS etc.
- ACID properties:
 - ✓ Atomic
 - ✓ Consistent
 - ✓ Isolated
 - ✓ Durable
- Applied on the service layer - allows you to reuse access code in various transaction scenarios without modification

Spring's Transaction Support

- Programmatic transaction demarcation through direct use of a subclass of PlatformTransactionManager.
- A template style approach using a TransactionTemplate that will handle most of the transaction demarcation.
- Declarative transaction demarcation for POJOs using Spring's AOP features.

Transaction Managers

- Select a TransactionManager based on your environment:
 - ✓ JTA - JtaTransactionManager - used for JDBC, Hibernate, JDO etc.
 - ✓ Local Transactions - specific manager based on technology:
 - ▶ DataSourceTransactionManager (JDBC/iBATIS)
 - ▶ HibernateTransactionManager
 - ▶ JdoTransactionManager

Transaction Definition

Transaction Propagation

PROPAGATION_REQUIRED

Support a current transaction, create a new one if none exists. This is typically the default setting of a transaction definition.

PROPAGATION_SUPPORTS

Support a current transaction, execute non-transactionally if none exists.

PROPAGATION_MANDATORY

Support a current transaction, throw an exception if none exists.

PROPAGATION_REQUIRES_NEW

Create a new transaction, suspending the current transaction if one exists.

PROPAGATION_NOT_SUPPORTED

Execute non-transactionally, suspending the current transaction if one exists.

PROPAGATION_NEVER

Execute non-transactionally, throw an exception if a transaction exists.

PROPAGATION_NESTED

Execute within a nested transaction if a current transaction exists, or behave like PROPAGATION_REQUIRED else. There is no analogous feature in EJB.

Transaction Definition

Isolation Levels

ISOLATION_DEFAULT

Use the default value of the underlying data source.

ISOLATION_READ_UNCOMMITTED

Dirty reads, non-repeatable reads and phantom reads can occur.

ISOLATION_READ_COMMITTED

Dirty reads are prevented; non-repeatable reads and phantom reads can occur.

ISOLATION_REPEATABLE_READ

Dirty reads and non-repeatable reads are prevented; phantom reads can occur.

ISOLATION_SERIALIZABLE

Dirty reads, non-repeatable reads and phantom reads are prevented.

Other options: **Timeout, Read-only**

Declarative Transactions

- Declarative transaction demarcation for POJOs using AOP in the form of ProxyFactoryBean combined with a TransactionInterceptor.
- Simplified declarative transaction demarcation for POJOs with the one-stop-shop TransactionProxyFactoryBean.
- Automatic proxying of business interfaces using the BeanNameAutoProxyCreator.
- Demarcating transactions and specifying transaction attributes using source-level metadata with annotations.

Declarative Transaction Management

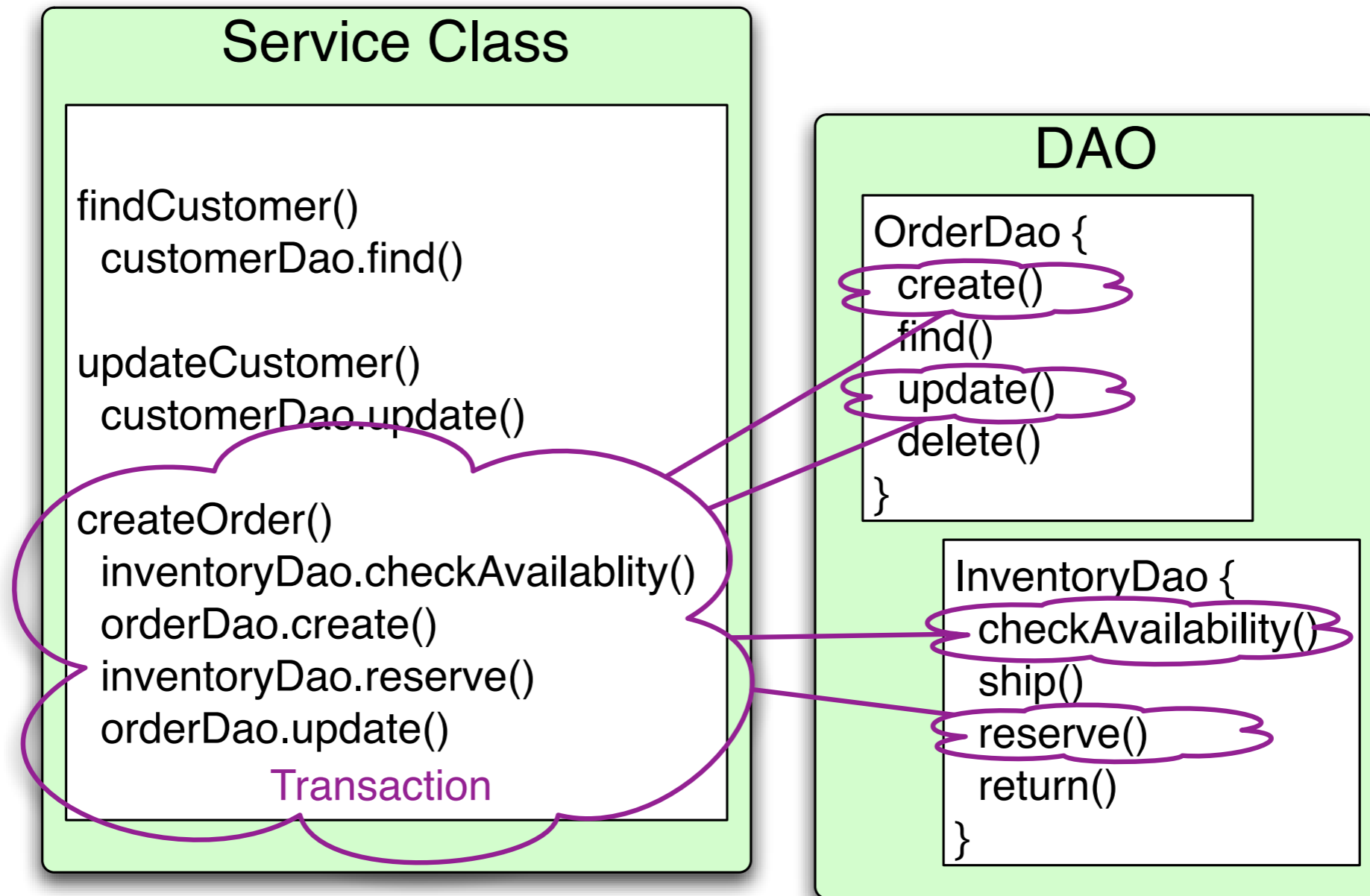
```
<bean id="dataSource"
...
</bean>

<bean id="transactionManager"
  class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource"><ref bean="dataSource"/></property>
</bean>

<bean id="beerDao" class="CheersDaoImpl">
  <property name="dataSource"><ref bean="dataSource"/></property>
</bean>

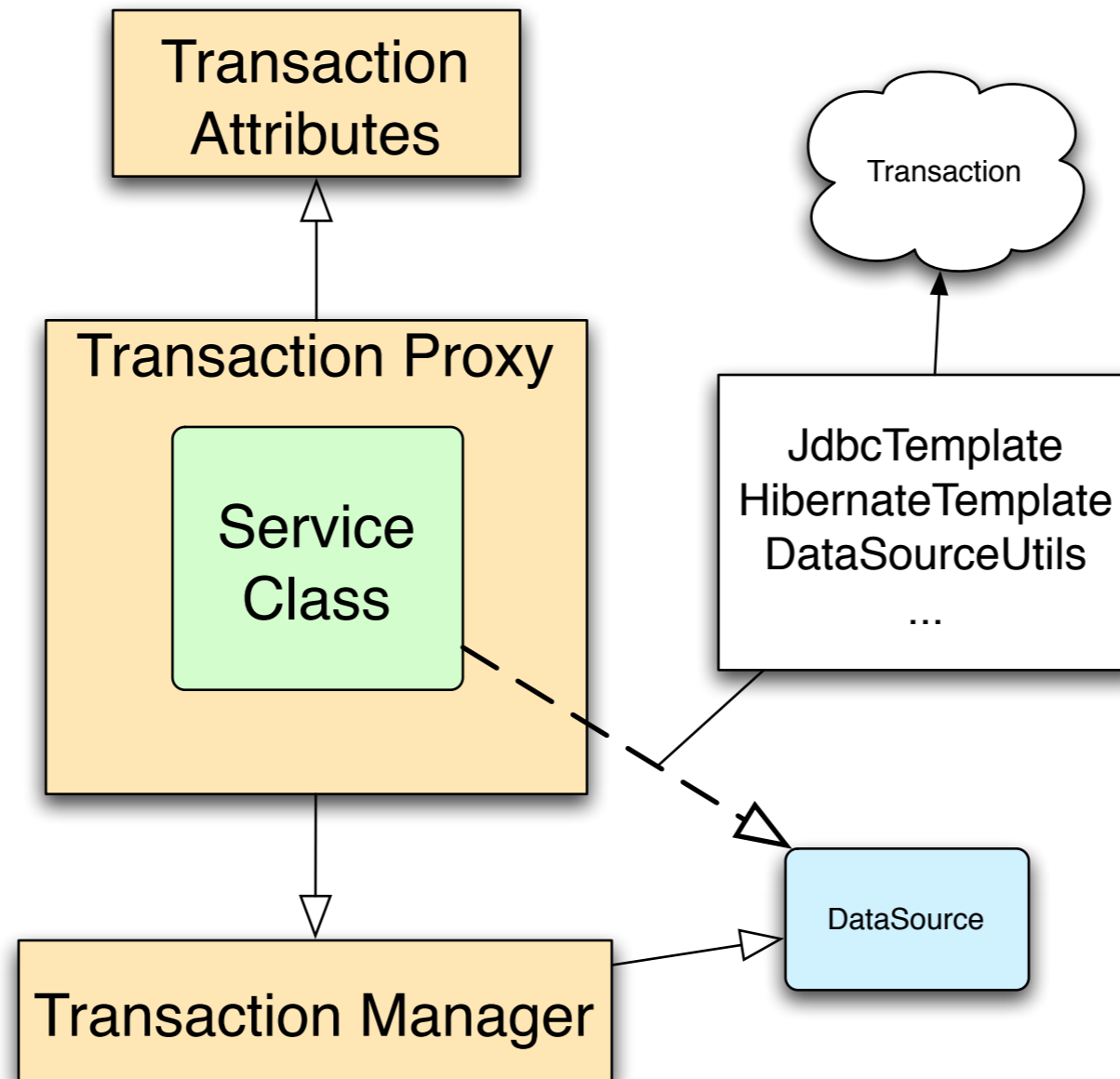
<bean id="service"
  class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
  <property name="transactionManager"><ref bean="transactionManager"/></property>
  <property name="target">
    <bean class="SpringCheersService">
      <property name="beerDao"><ref bean="beerDao"/></property>
    </bean>
  </property>
  <property name="transactionAttributes">
    <props>
      <prop key="*">PROPAGATION_REQUIRED</prop>
    </props>
  </property>
</bean>
```


Transaction Participants



Transaction

AOP Behind the Scenes



Transactions - other options

- **@Transactional** annotations will be more and more common for declarative transactions - use with a combination of:
 - ✓ DefaultAdvisorAutoProxyCreator
 - ✓ TransactionAttributeSourceAdvisor
 - ✓ AnnotationTransactionAttributeSource
- Use **TransactionProxyFactoryBean** in other pre Java 5 environments
- Use **BeanNameAutoProxyCreator** to advice many beans in an identical fashion

Transactions - Many Options

