Transaction Management

Spring Transactions

- Provides declarative transaction management.
- Seamless support for both native transactions(local) and J2EE server transactions (JTA, global)
- Flexible replacement for EJB CMT

Spring Framework transaction abstraction

- The spring interface that provides the transaction abstraction is ***org.springframework.transaction.PlatformTransactionManager***
- This is primarily a service provider interface (SPI), although it can be used programmatically from your application code

Steps to Configure Spring Transaction

- Define the transaction manager
- Define the transaction boundaries

Transaction Managers

- Native transaction managers
  - JDBC: *DataSourceTransactionManager*
  - Hibernate: *HibernateTransactionManager*
  - TopLink: *TopLinkTransactionManager*
  - JDO: *JdoTransactionManager*
- Native strategies work in any environment
  - but only against a single database!
    - no distributed transaction coordination
  - leverage full power of underlying resource
    - isolation levels, savepoints, etc

Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:jee="http://www.springframework.org/schema/jee"
      xsi:schemaLocation="
      http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
      http://www.springframework.org/schema/jee
      http://www.springframework.org/schema/jee/spring-jee-3.0.xsd">

      <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-
          method="close">
           <property name="driverClassName" value="${jdbc.driverClassName}" />
           <property name="url" value="${jdbc.url}" />
           <property name="username" value="${jdbc.username}" />
           <property name="password" value="${jdbc.password}" />
      </bean>


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


<!-- other <bean/> definitions here -->
</beans>
```

Transaction Managers

- JTA-based transaction coordinator
  - Spring's JtaTransactionManager adapter
- Java Transaction API to talk to an external transaction coordinator
    - typically using the standard XA protocol to coordinate heterogeneous transactional resources

Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:jee="http://www.springframework.org/schema/jee"
    xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
    http://www.springframework.org/schema/jee
    http://www.springframework.org/schema/jee/spring-jee-3.0.xsd">

    <jee:jndi-lookup id="dataSource" jndi-name="jdbc/jpetstore"/>

    <bean id="txManager"
        class="org.springframework.transaction.jta.JtaTransactionManager" >
            <property name="dataSource" ref="dataSource"/>
    </bean>

<!-- other <bean/> definitions here -->
</beans>
```

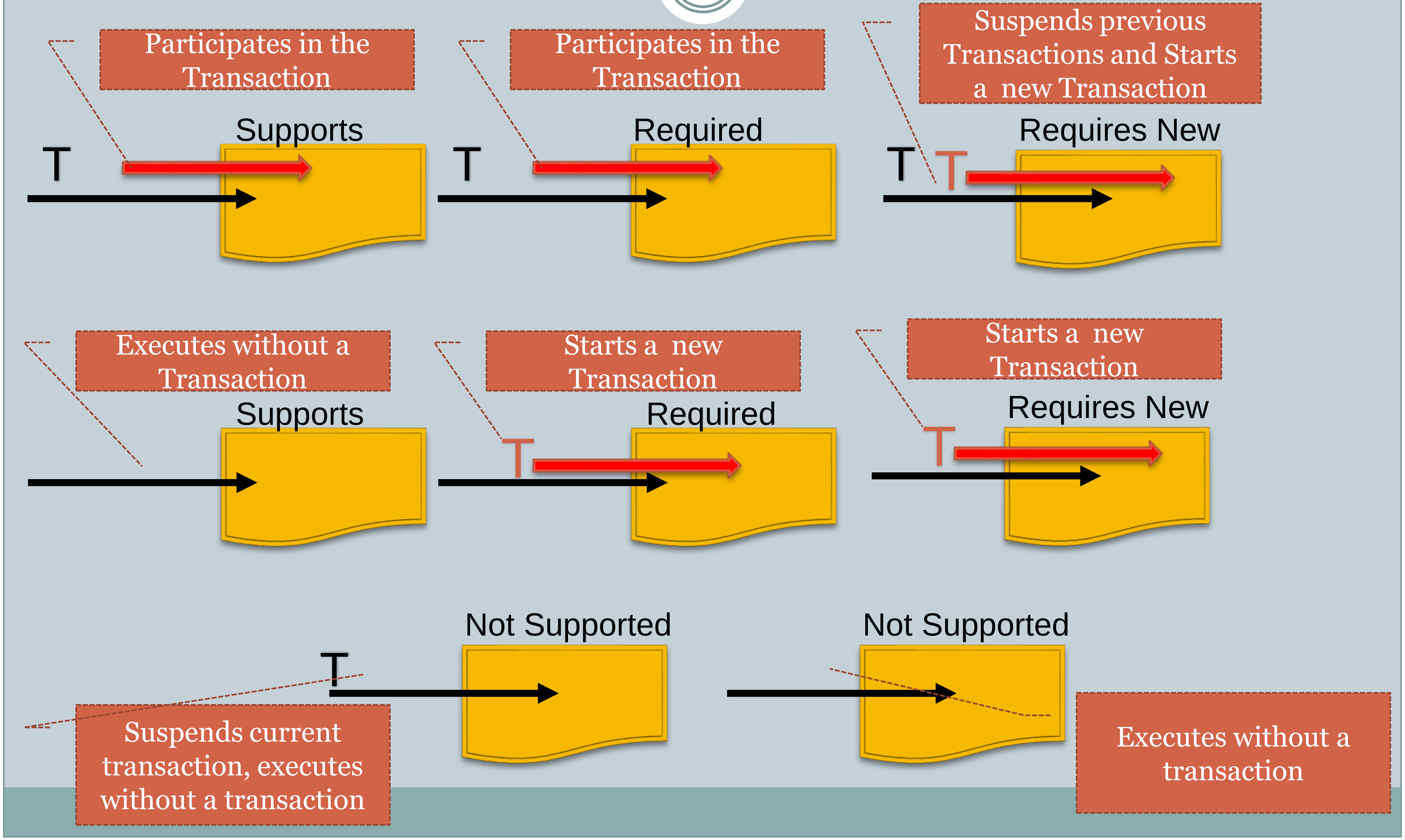
Transaction Demarcation

- Transaction definition
  - propagation behavior
    - defaults to PROPAGATION_REQUIRED
  - isolation level
    - defaults to ISOLATION_DEFAULT
  - read-only flag
    - defaults to false
  - transaction timeout
    - defaults to none

Transaction Demarcation

- Propagation behaviors
  - supporting all EJB CMT propagation codes
    - SUPPORTS
    - REQUIRED
    - REQUIRES_NEW
    - NOT_SUPPORTED

Propagation Behaviors



Participates in the Transaction

Participates in the Transaction

Suspends previous Transactions and Starts a new Transaction

Executes without a Transaction

Starts a new Transaction

Starts a new Transaction

Suspends current transaction, executes without a transaction

Executes without a transaction