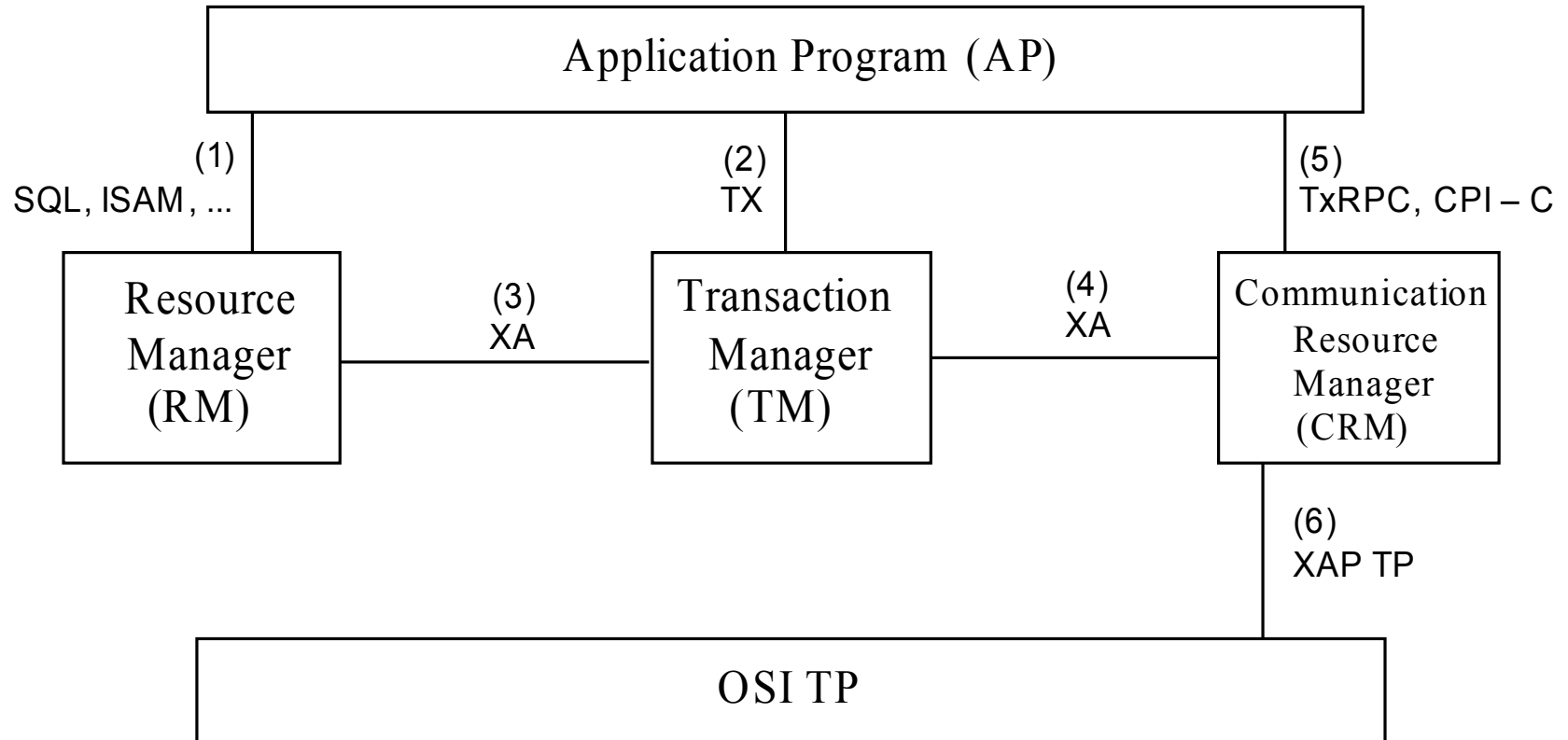


Модель обработки распределенных транзакций X/Open DTP

Модель X/Open DTP

- Используется как стандартная основа для любой реализации системы управления распределенными транзакциями
- Разработана европейским консорциумом X/Open
- Состоит из нескольких компонентов и определяет интерфейсы между ними

Модель X/Open DTP



Компоненты модели X/Open DTP

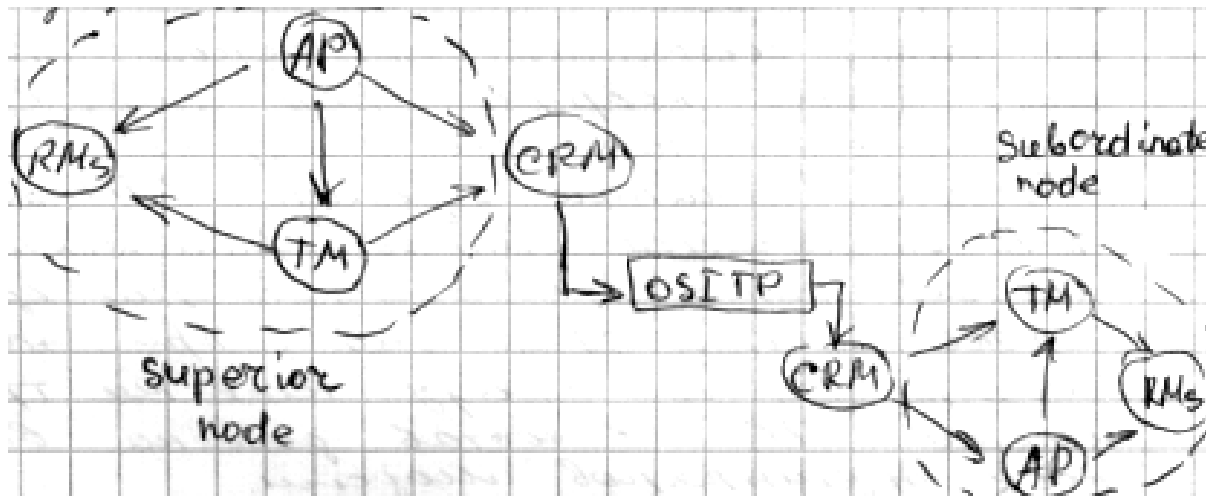
- AP – прикладная программа, которая участвует в распределенной транзакции
 - > определяет моменты начала и окончания транзакции
 - > взаимодействует с RM и TM, а также через компонент CRM с другими AP, вовлеченными в одну распределенную транзакцию
- TM – координатор распределенных транзакций
 - > обеспечивает атомарность распределенной транзакции
 - > синхронизирует моменты начала и окончания распределенной транзакции с моментами начала и окончания локальных транзакций в RM
 - > может взаимодействовать с удаленными координаторами через компонент CRM

Компоненты модели X/Open DTP

- RM – менеджер ресурсов (реляционная СУБД, система обмена сообщениями, файловая система и пр.)
 - > обеспечивает транзакционный доступ к информации
- CRM — менеджер коммуникационных ресурсов
 - > обеспечивает взаимодействие AP и TM с другими AP и TM по сети в рамках одной распределенной транзакции
- OSI TP – обеспечивает низкоуровневую коммуникационную среду
 - > сетевое взаимодействие между компонентами модели в рамках одной распределенной транзакции

Экземпляр модели и транзакционный домен

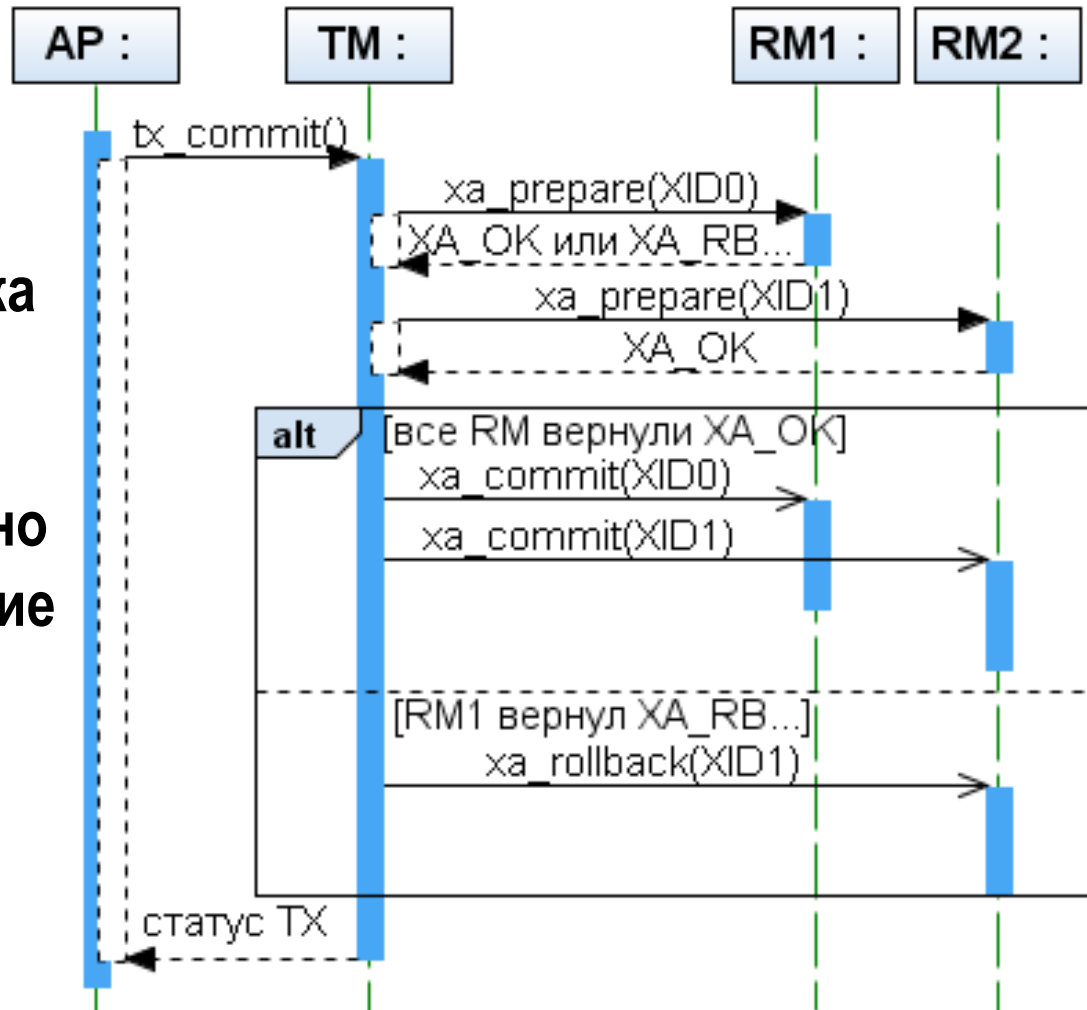
- Экземпляр = $1 \cdot AP + 1 \cdot TM + N \cdot RM$
- Транзакционный домен = $1 \cdot TM + N \cdot AP + N \cdot RM$
 - > Разные координаторы транзакций -> разные транзакционные домены



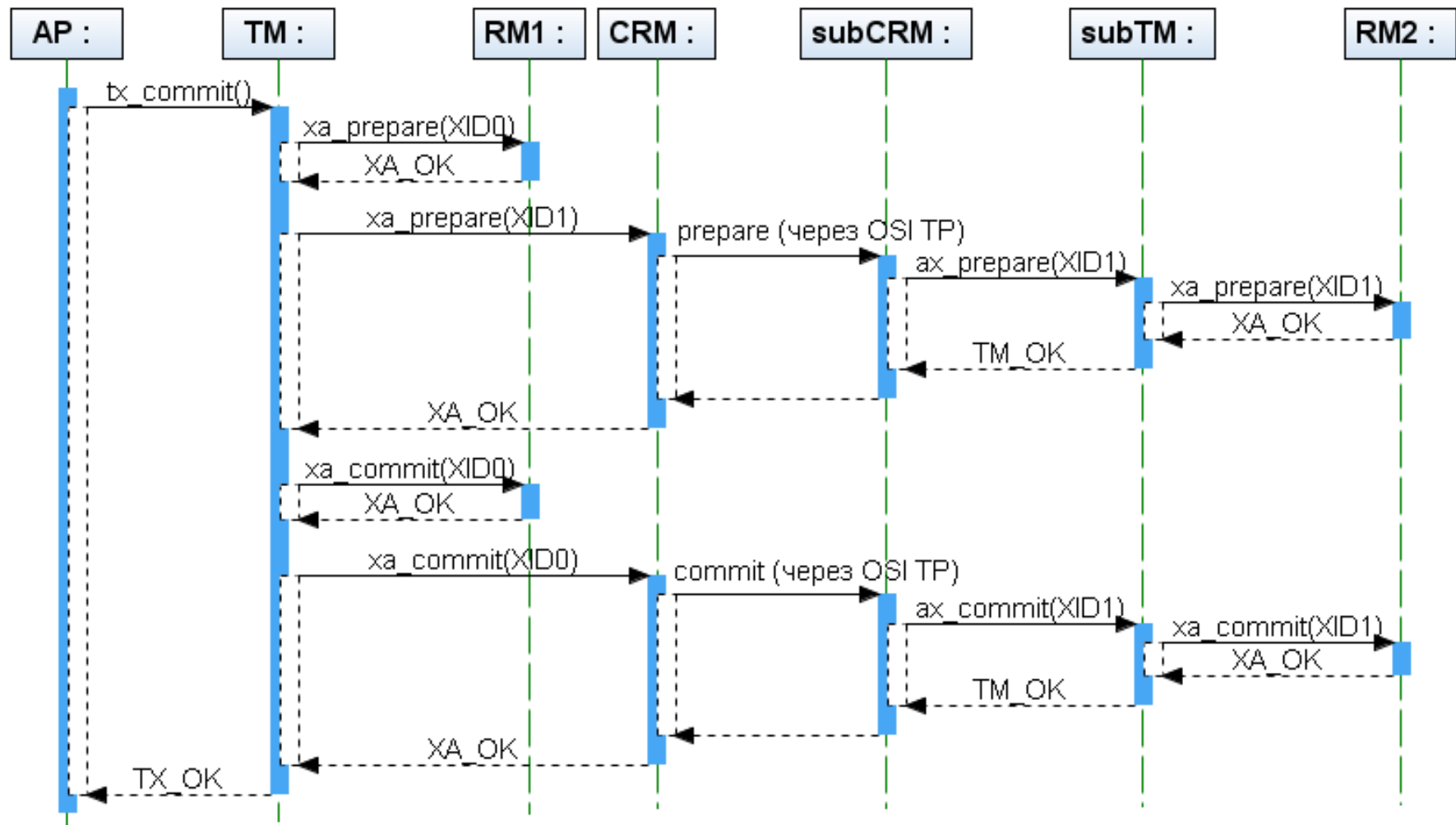
- Ветвь транзакции (XID) — часть глобальной транзакции, $1 \cdot TM + 1 \cdot RM$

Протокол двухфазной фиксации транзакций 2PC (Two-Phase Commit)

1. подготовка
2. собственно подтверждение



Протокол 2PC в случае нескольких транзакционных доменов



Оптимизация протокола 2PC

- если с конкретным источником данных в ходе глобальной транзакции выполнялись **ТОЛЬКО операции чтения** данных, то он может закончить свое участие в глобальной транзакции уже после первого этапа
- если в глобальной транзакции участвует **ТОЛЬКО ОДИН ИСТОЧНИК ДАННЫХ**, то вместо двухфазного протокола может использоваться обычный (однофазный) протокол подтверждения

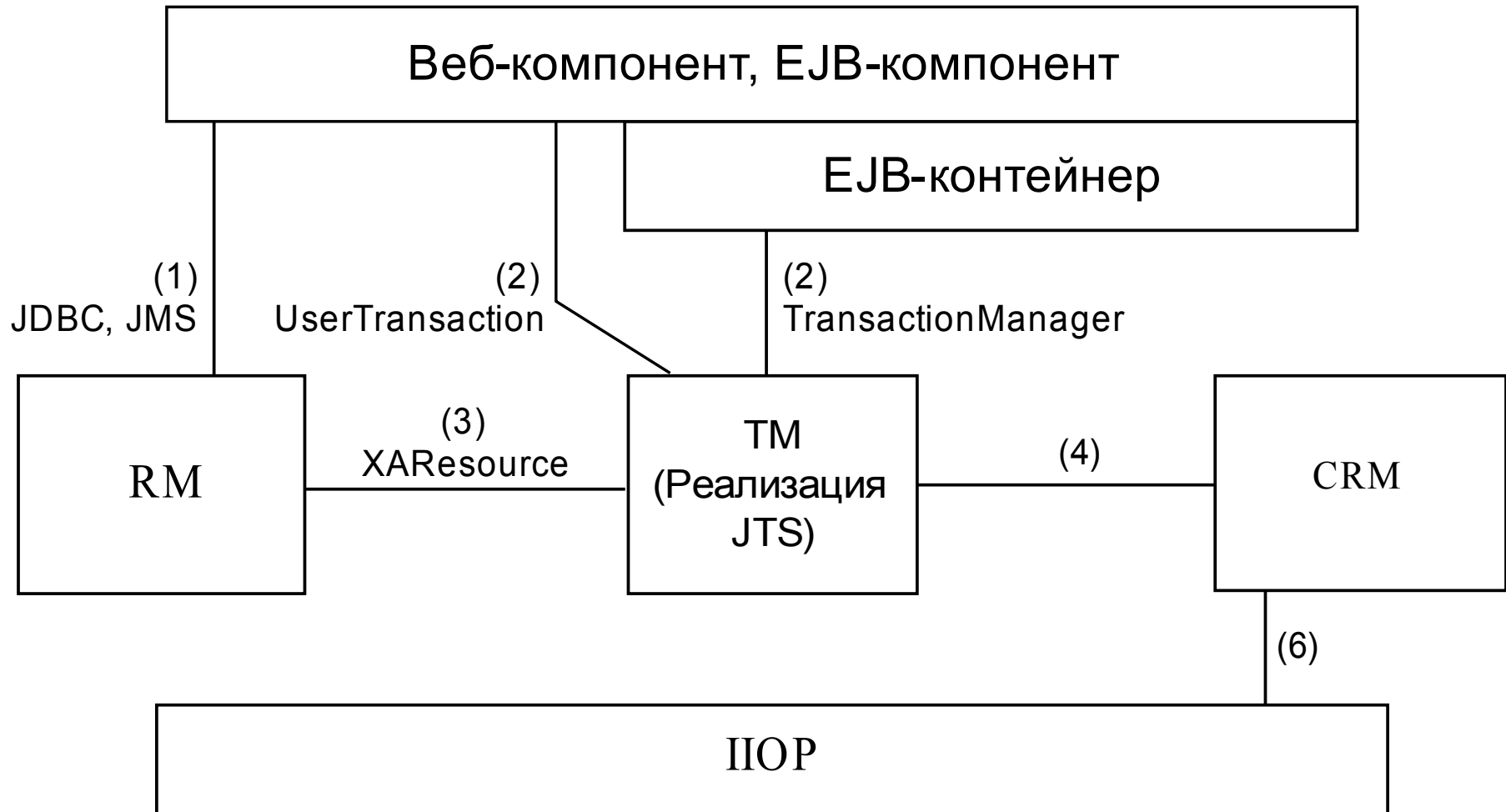
Откат глобальной транзакции

- **явный откат** (explicit rollback) – приложение явно инициирует откат
- **неявный откат** (implicit rollback) – приложение явно инициировало подтверждение транзакции, а в ходе подготовки был получен хотя бы один отрицательный ответ
- **подразумеваемый откат** (presumed rollback) – происходит при перезапуске источника данных после сбоя, произошедшего в нем в ходе глобальной транзакции

Обработка отказов в 2PC

- Единственный способ выявления сбоев – это ожидание сообщения в течение определенного промежутка времени
- Если в период подготовки сбой происходит и на координаторе, и на одном из участников, оставшиеся узлы не способны решить между собой судьбу транзакции и вынуждены оставаться в заблокированном состоянии
- Эвристическое завершение транзакции - локальная транзакция в источнике данных может завершиться автоматически подтверждением или откатом до того, как будет получен соответствующий сигнал от координатора
 - > Например, по таймауту

Реализация модели X/Open DTP на платформе Java EE



Реализация модели X/Open DTP на платформе Java EE

- Чтобы СУБД могла принимать участие в распределенной транзакции, для нее следует использовать особый тип источника данных **javax.sql.XADataSource**

JDBC Connection Pool Name:*

☐ Extract from Existing Connection:

☒ New Configuration using Database:

☒ XA (Global Transaction)

Datasource Classname:

Resource Type:

Description:

Properties:

| Name | Value |
|----------|---|
| URL | jdbc:mysql://<HOST>:<PORT>/<DB>?relaxAut... |
| User | |
| Password | |

- Код компонента **не зависит** от типа выполняемой транзакции (локальная или глобальная) – в сервере приложений

Оптимизация локальных транзакций на платформе Java EE

- В рамках распределенной транзакции может быть задействован **только один non-XA** data source
- Транзакция называется **локальной**, если в ней задействован только **non-XA** data source
- Транзакция называется **глобальной**, если в ней задействован хотя бы 1 **XA** data source
- Приложение с локальной транзакцией работает точно так же, как с глобальной
- Сервер приложений расходует гораздо меньше ресурсов, чем при обработке глобальной транзакции