

Subversion и разработка web-приложений

- Что такое VCS
- Основные понятия VCS
- Работа с Subversion
- Организация репозитория Subversion
- Вопросы использования Subversion для веб-разработки

Системы контроля версий

Version Control Systems (VCS), Source Code Management Systems (SCM)

Позволяют:

- хранить последовательный набор версий проекта
- хранить комментарии к вносимым изменениям
- получать предыдущие версии файлов проекта
- отслеживать *когда, какие, кем и зачем* были внесены изменения
- вести параллельную разработку разных частей проекта, облегчая задачу синхронизации вносимых разными командами изменений
- получать изменения между двумя произвольными версиями
- облегчить задачу поиска мест возникновения логических ошибок

Виды VCS

По типу элементарных объектов

Отслеживание изменений
на уровне файлов

Отслеживание изменений
на уровне всего
дерева каталогов

По модели работы с репозиторием

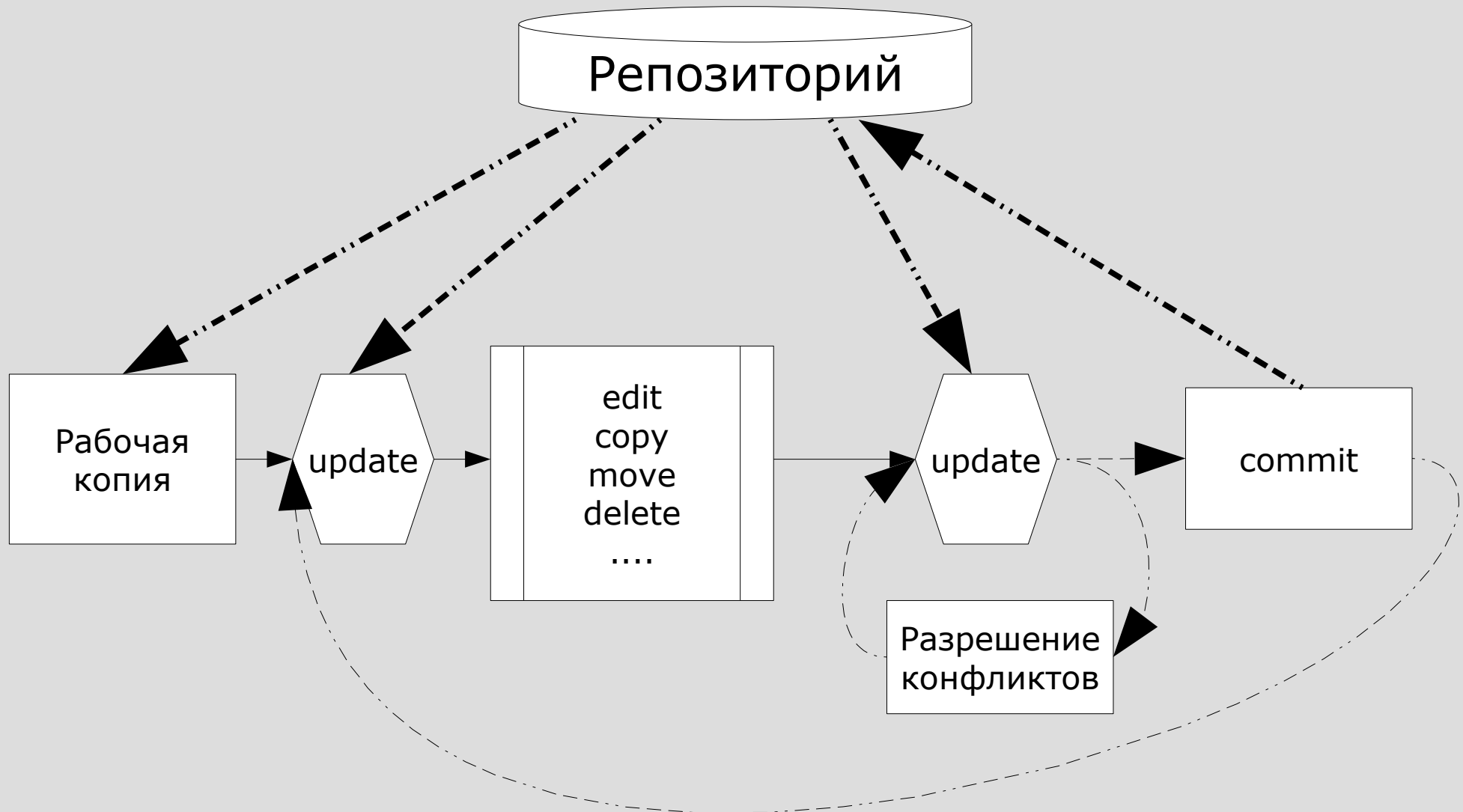
Централизованные
(Клиент-серверная модель,
1 репозиторий – N клиентов)

Децентрализованные
(1 репозиторий – 1 клиент,
все репозитории равноправны)

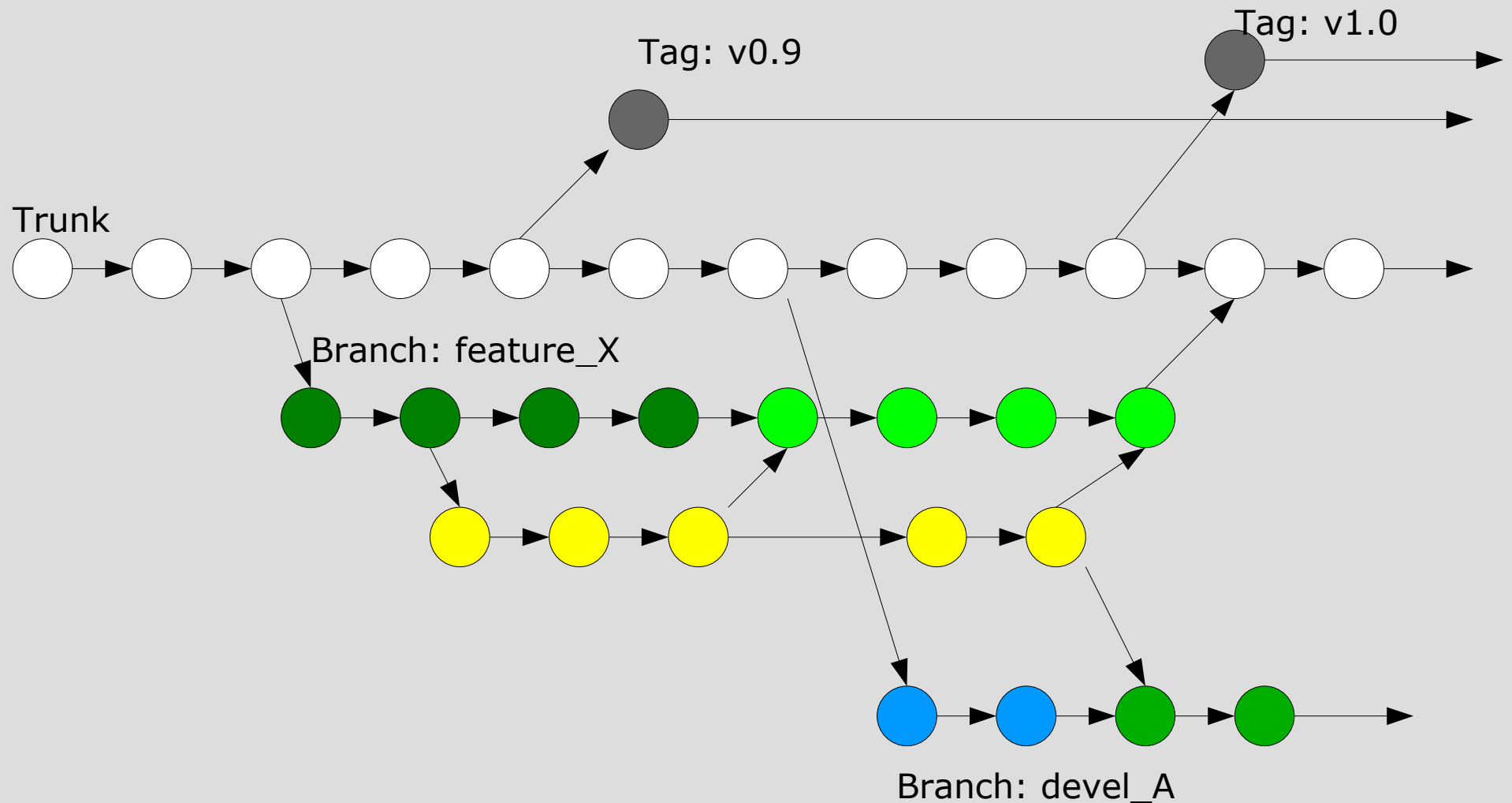
Основные понятия VCS

- **Репозиторий (repository)** – хранилище файлов, их версий, служебной информации и пр.
- **Версия (revision, version)** – версия проекта. Имеет уникальный идентификатор в пределах репозитория, позволяющий однозначно отличить одну версию от другой.
- **Ветвь разработки (branch)** – независимое направление разработки проекта.
- **Тэг (tag)** – метка конкретной версии, применяется для логического выделения этапов процесса разработки
- **Коммит (commit, submit, check-in)** – процесс (действие) создание новой версии.
- **Слияние (merge)** – объединение независимых изменений
- **Рабочая копия (working copy)** – полученное из репозитория дерево каталогов проекта некой версии (текущей или прошлой)
- **Check-out** – получение из репозитория рабочей копии.
- **Update** – обновление рабочей копии до текущего состояния репозитория
- **Конфликт (conflict)** – ситуация, когда VCS не может автоматически применить внесённые изменения (т.е. когда параллельно были исправлены одни и те же места в файлах)

Общий порядок работы с централизованными VCS



Общий порядок работы с централизованными VCS

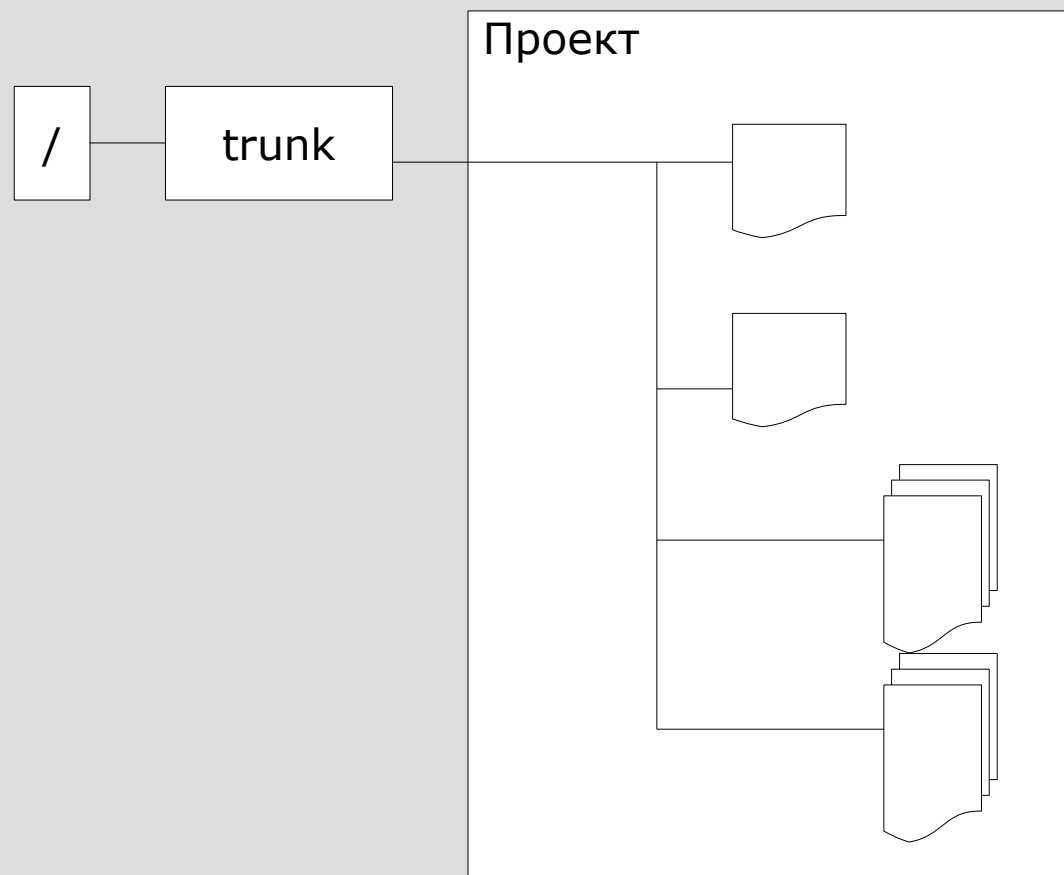


Особенности Subversion

- Атомарные коммиты
- Отслеживание состояния всего дерева каталогов
- Версия (=коммит) – мгновенный снимок целиком всего дерева каталогов
- Каждая версия имеет уникальный последовательный номер.
- Невозможность переписать историю
- Ведение истории изменений файлов и каталогов, включая их переименование/копирование. Для операций с файлами **обязательно** использование инструментов Subversion.
- Ветки и тэги реализованы через операции копирования поддеревя каталогов проекта.
- Возможность сравнения и слияния поддеревьев каталогов
- Возможность получения в виде рабочей копии поддеревя каталогов
- Система контроля доступа на уровне каталогов

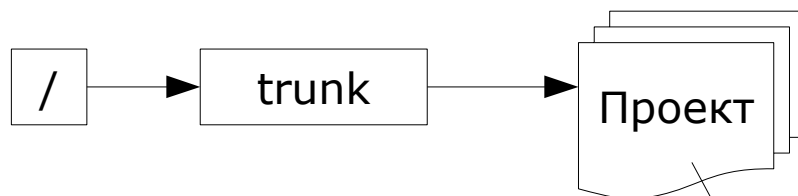
Особенности Subversion: организация репозитория

Текущая версия проекта (структура каталогов) размещается в /trunk/ :



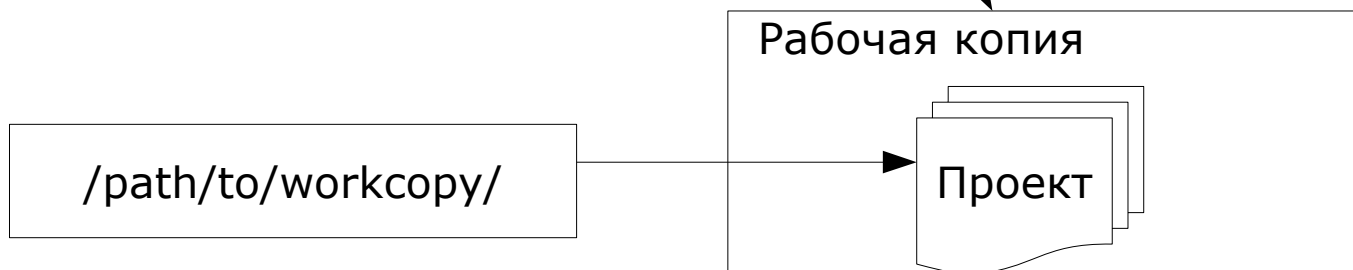
Особенности Subversion: рабочие копии

Репозиторий: \$URL/

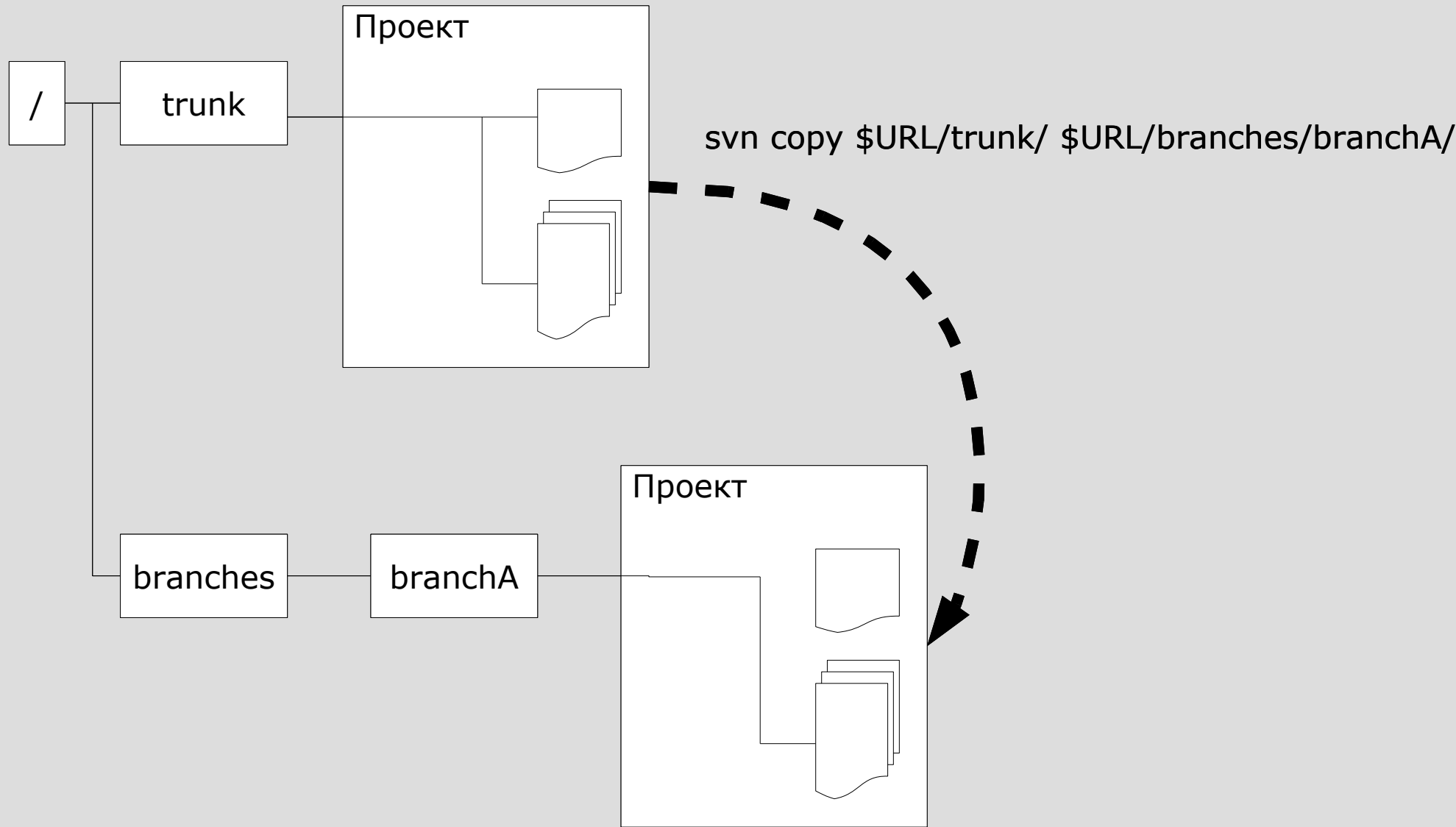


svn checkout \$URL/trunk/

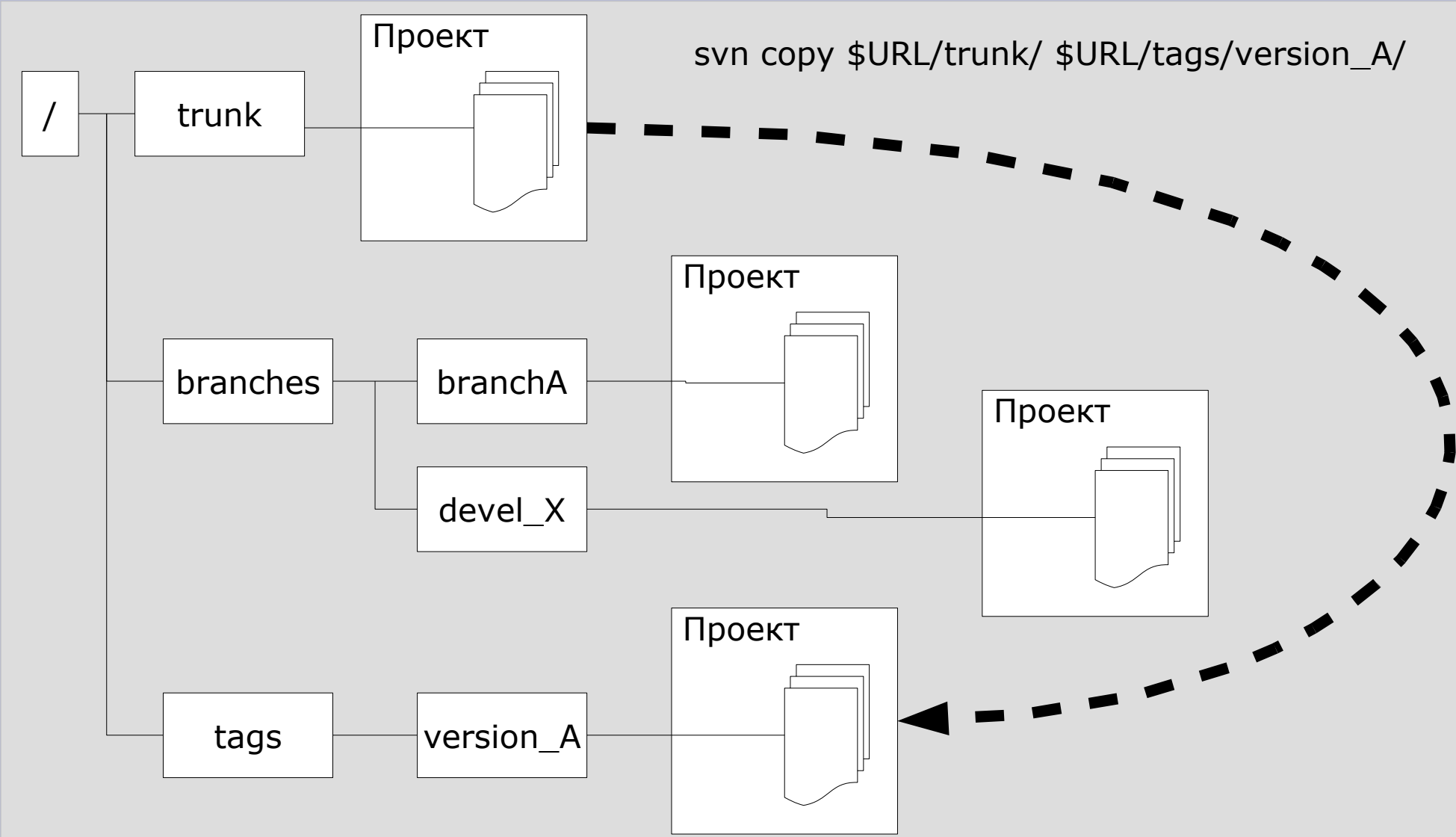
Компьютер разработчика



Особенности Subversion: ветви разработки

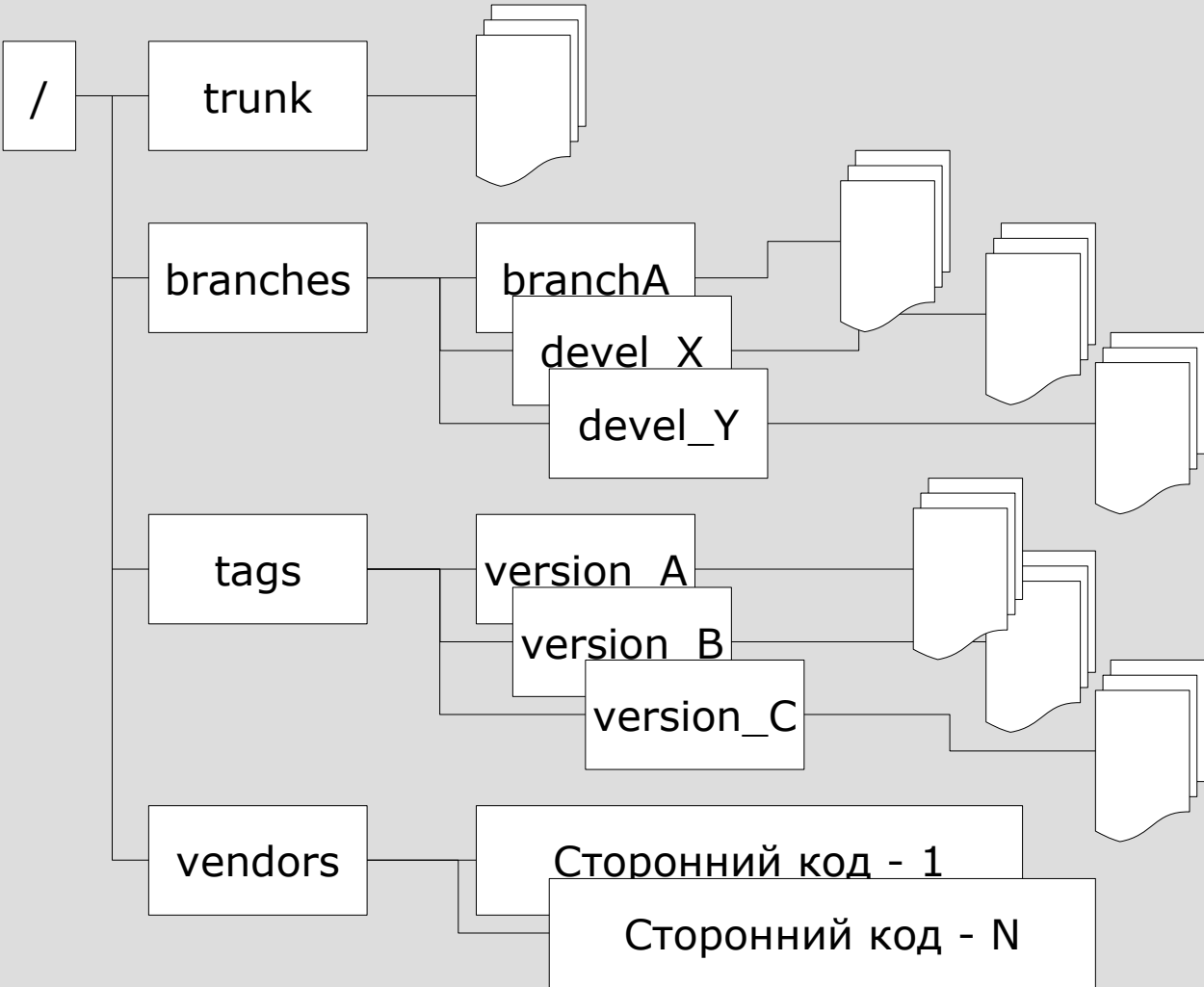


Особенности Subversion: ТЭГИ



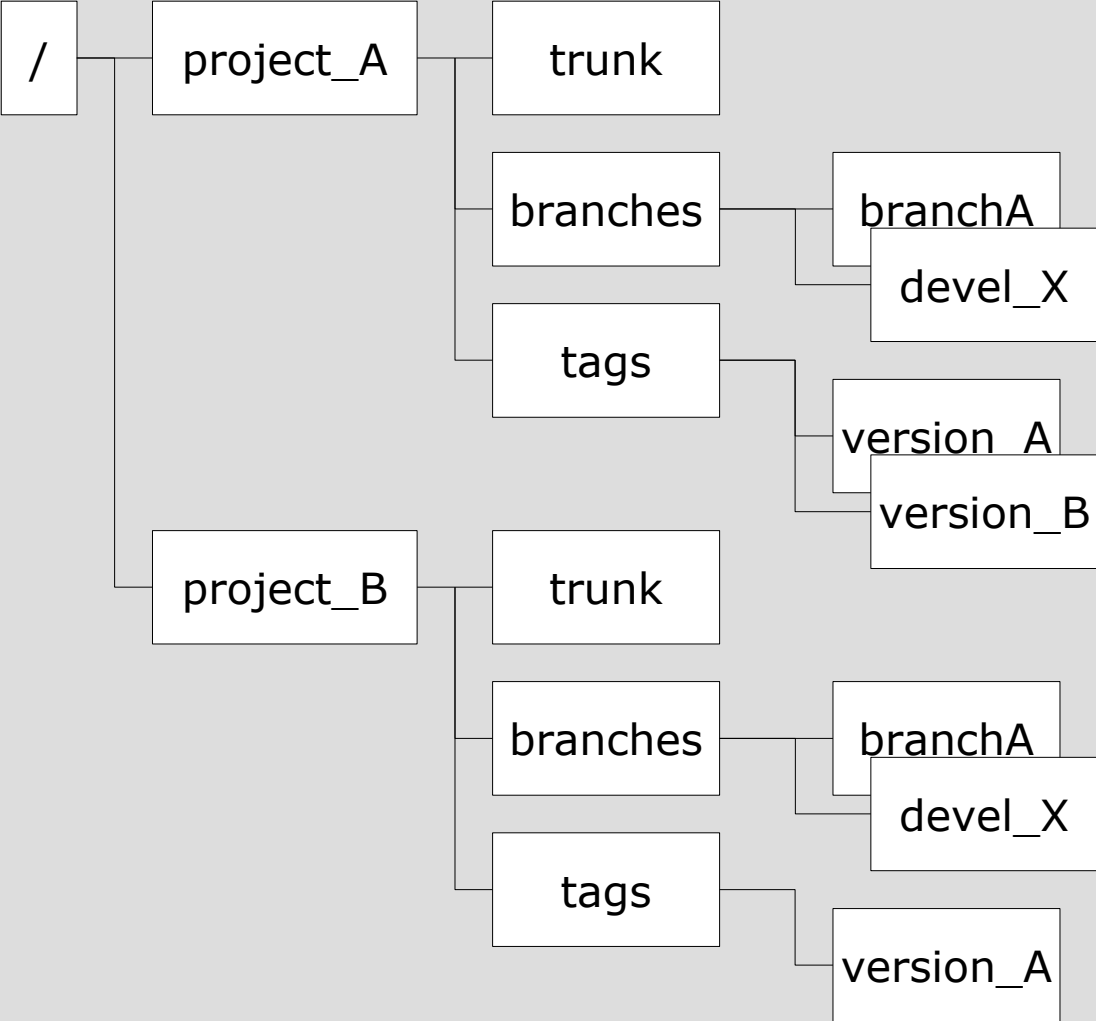
Особенности Subversion: структура репозитория

Для схемы "1 репозиторий – 1 проект"

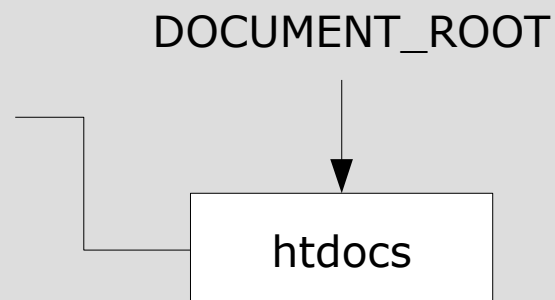


Особенности Subversion: структура репозитория

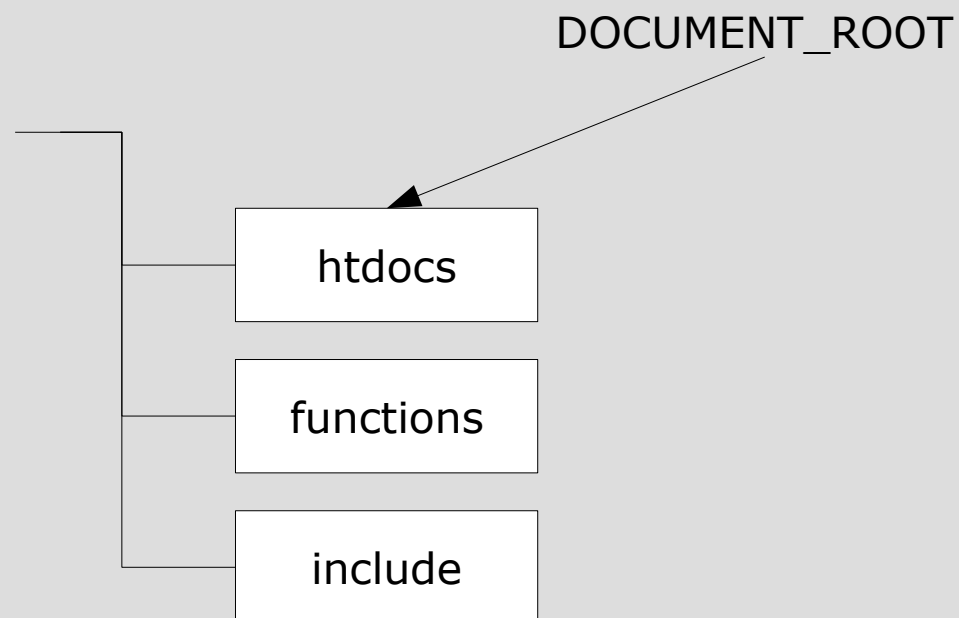
Для схемы "1 репозиторий – N проектов"



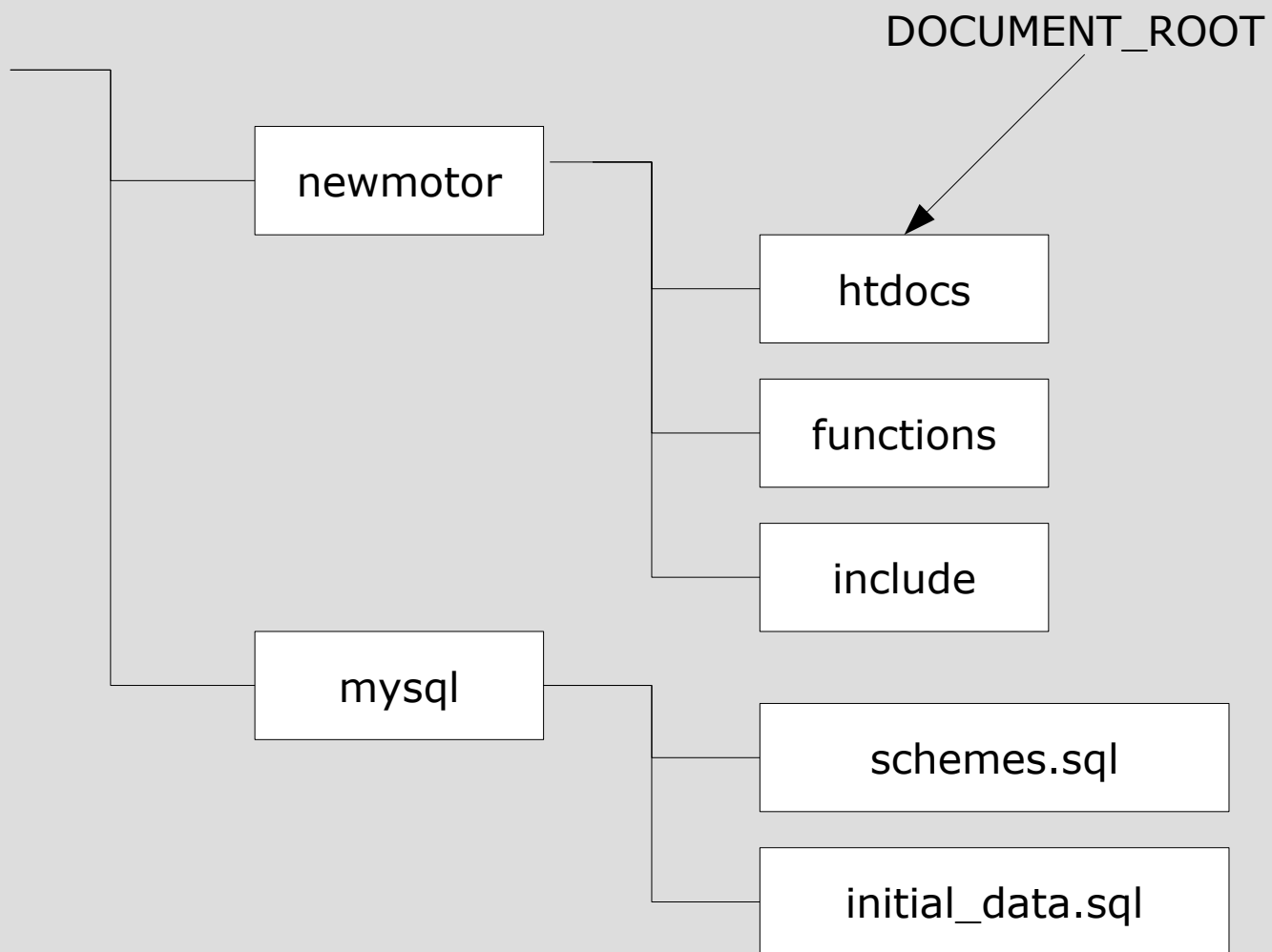
Структура каталогов проекта



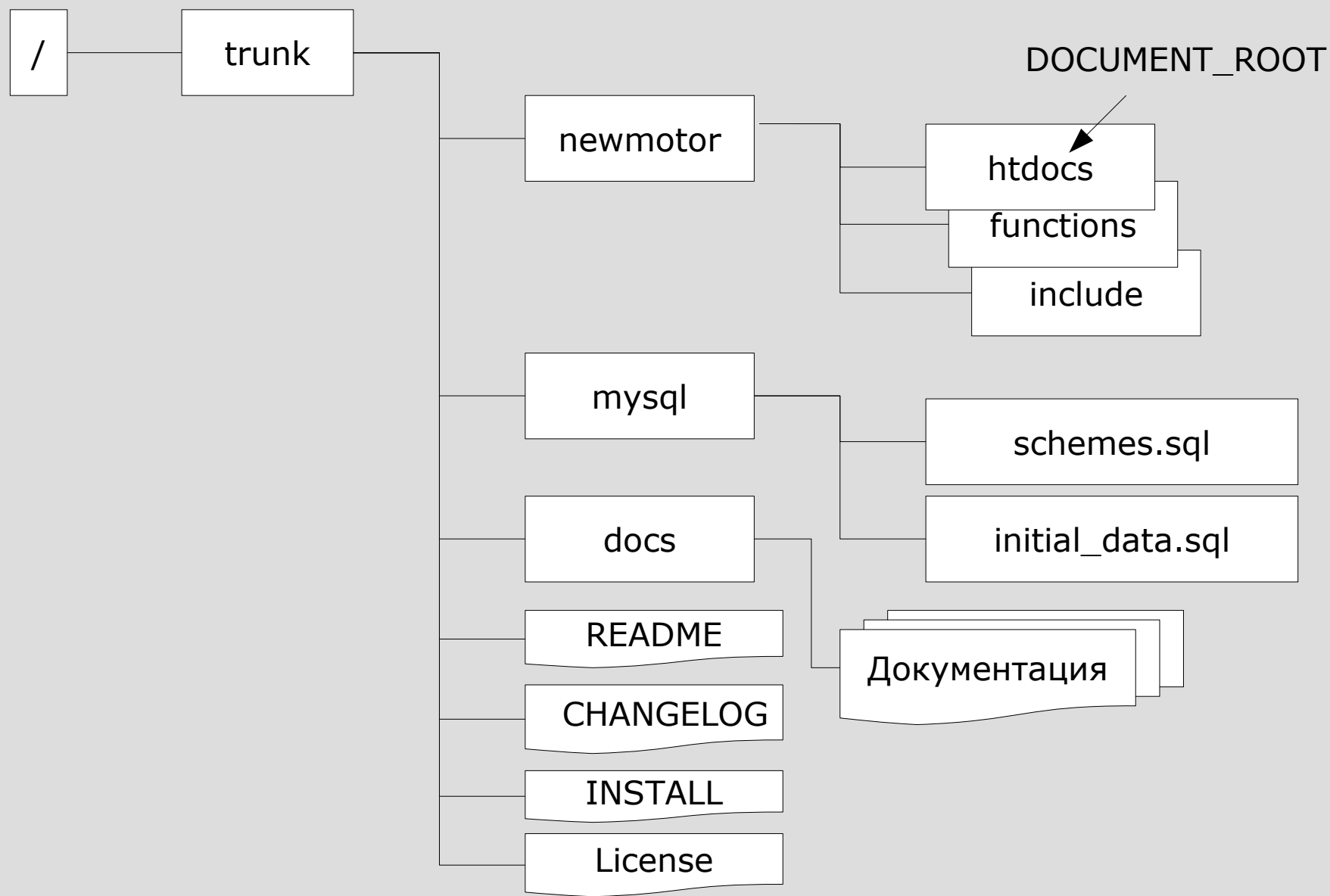
Структура каталогов проекта



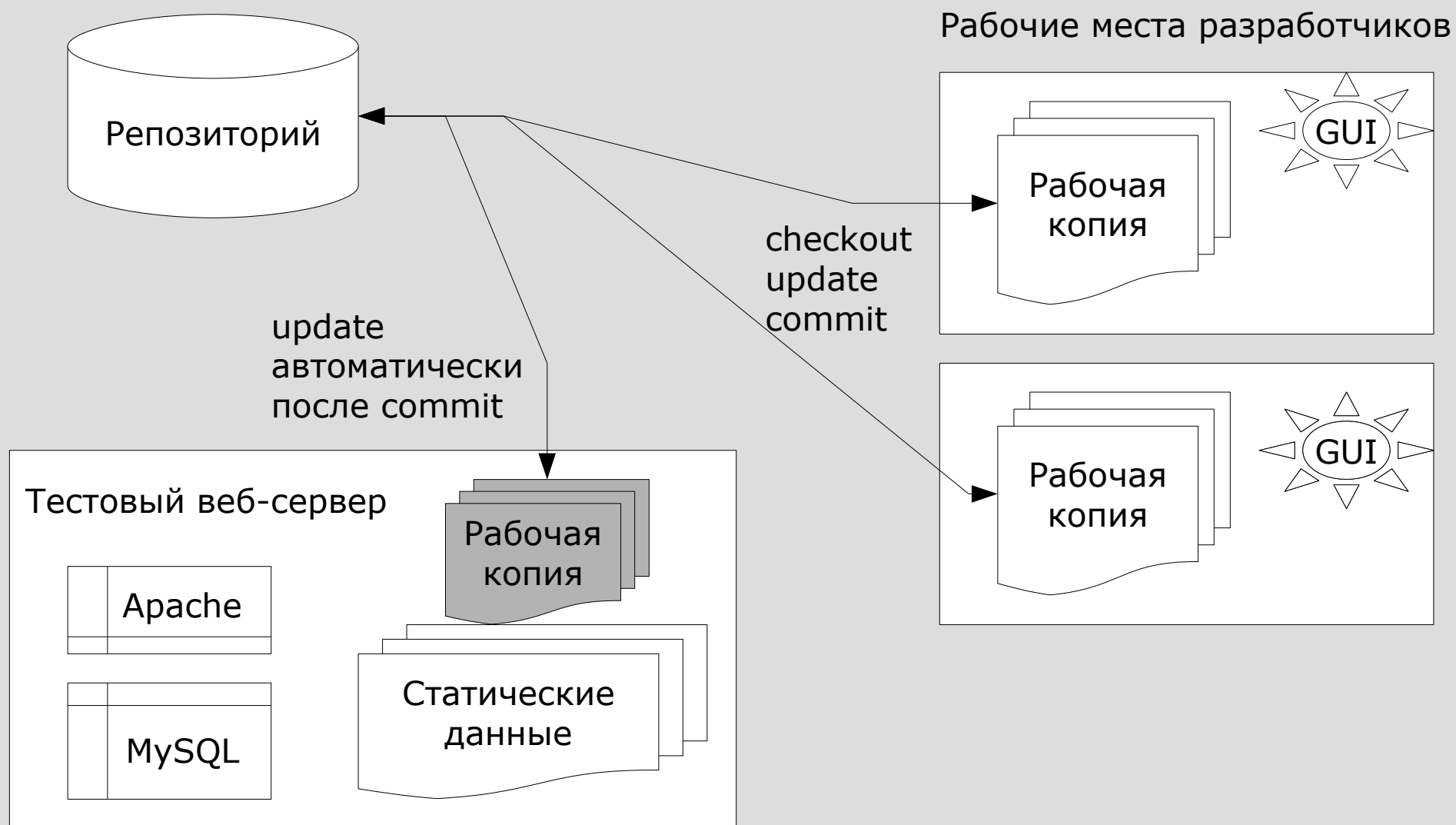
Структура каталогов проекта



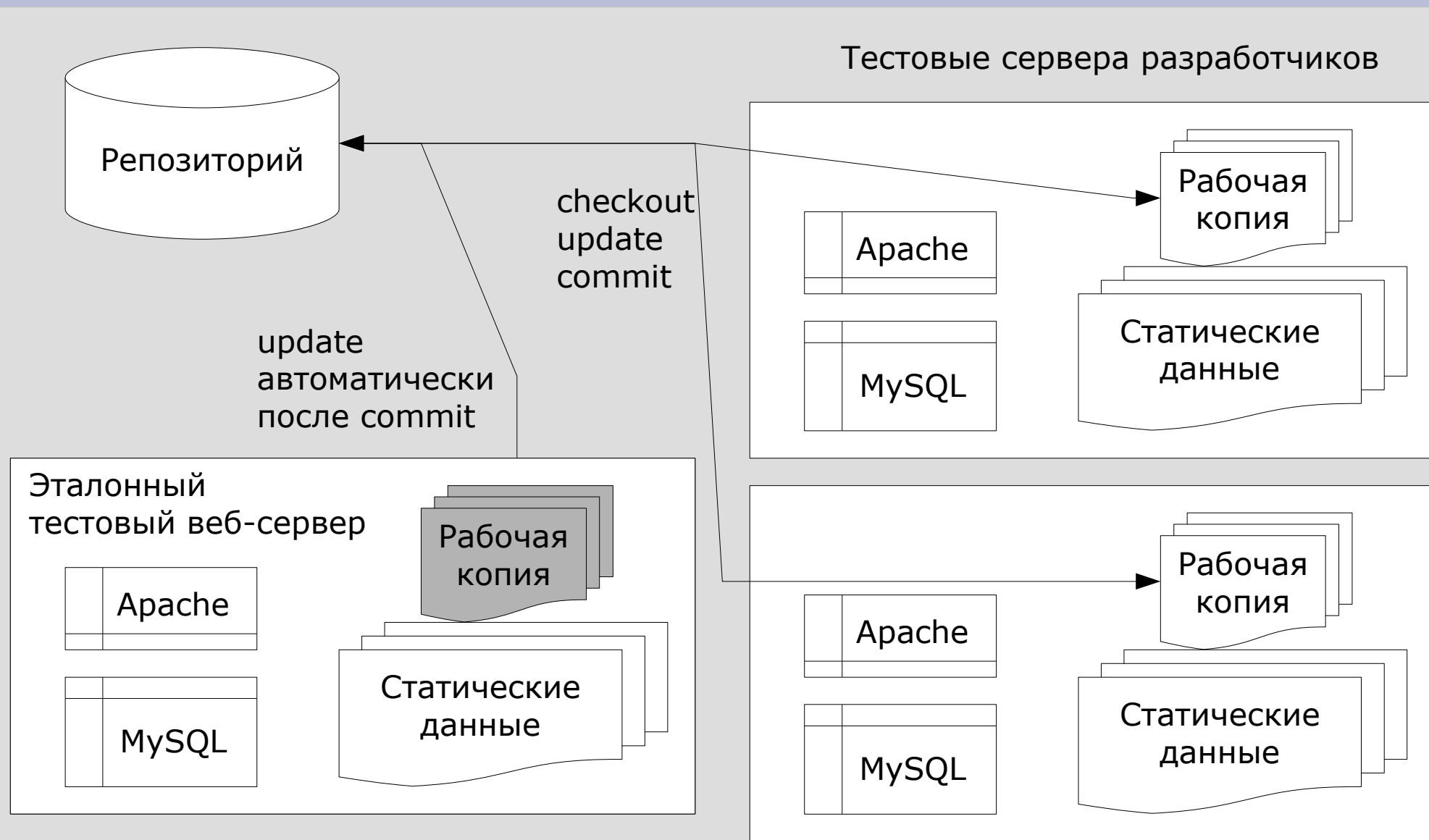
Структура каталогов проекта



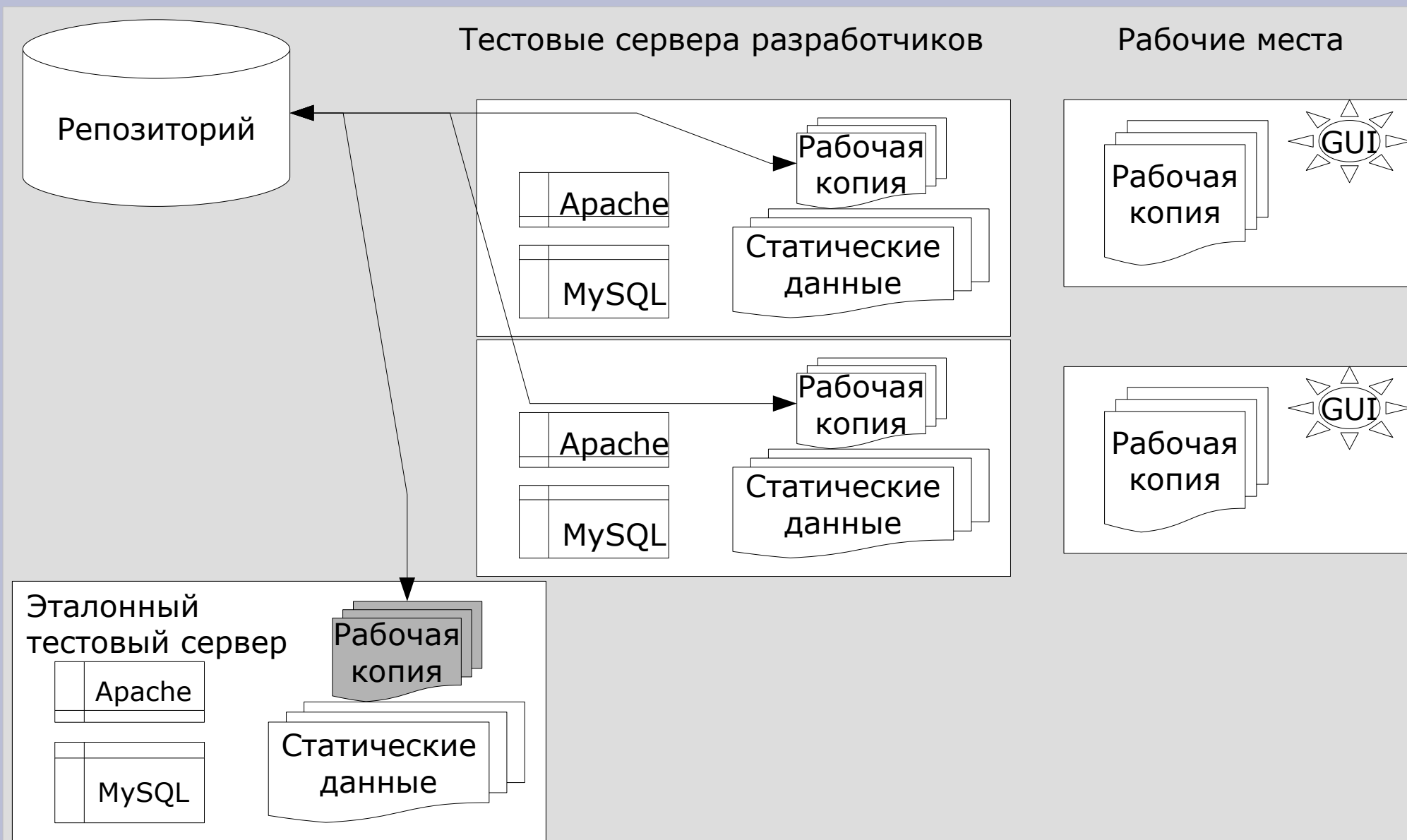
Использование Subversion при веб-разработке



Использование Subversion при веб-разработке



Использование Subversion при веб-разработке



Итоги

Внедрение Subversion:

- вредно для плохих программистов
- полезно для хороших программистов
- очень полезно для руководителей групп
- облегчает работу руководящего звена

При использовании Subversion:

- работать с файлами только средствами Subversion
- желательно регулярно выполнять update
- желательно регулярно выполнять commit
- при коммите описывать сделанные изменения

Надо:

- определиться с структурой тестовых серверов
- создать чистый репозиторий со структурой каталогов верхнего уровня
- создать начальную структуру проекта, со всеми каталогами и нужными файлами
- импортировать проект в /trunk