

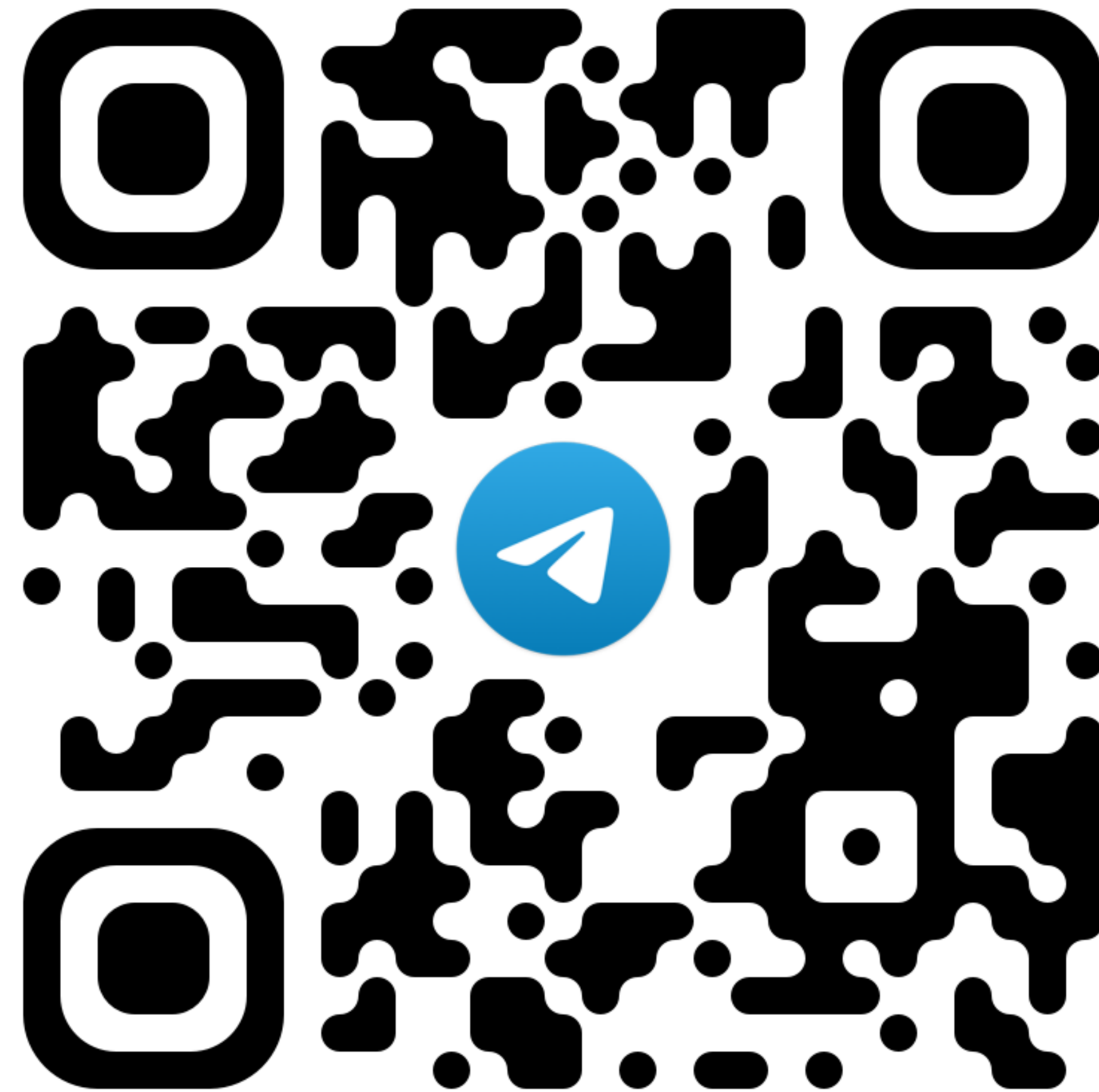


Введение в Git

Докладчик: Евграфов Михаил

Организационные вопросы

Организационные вопросы

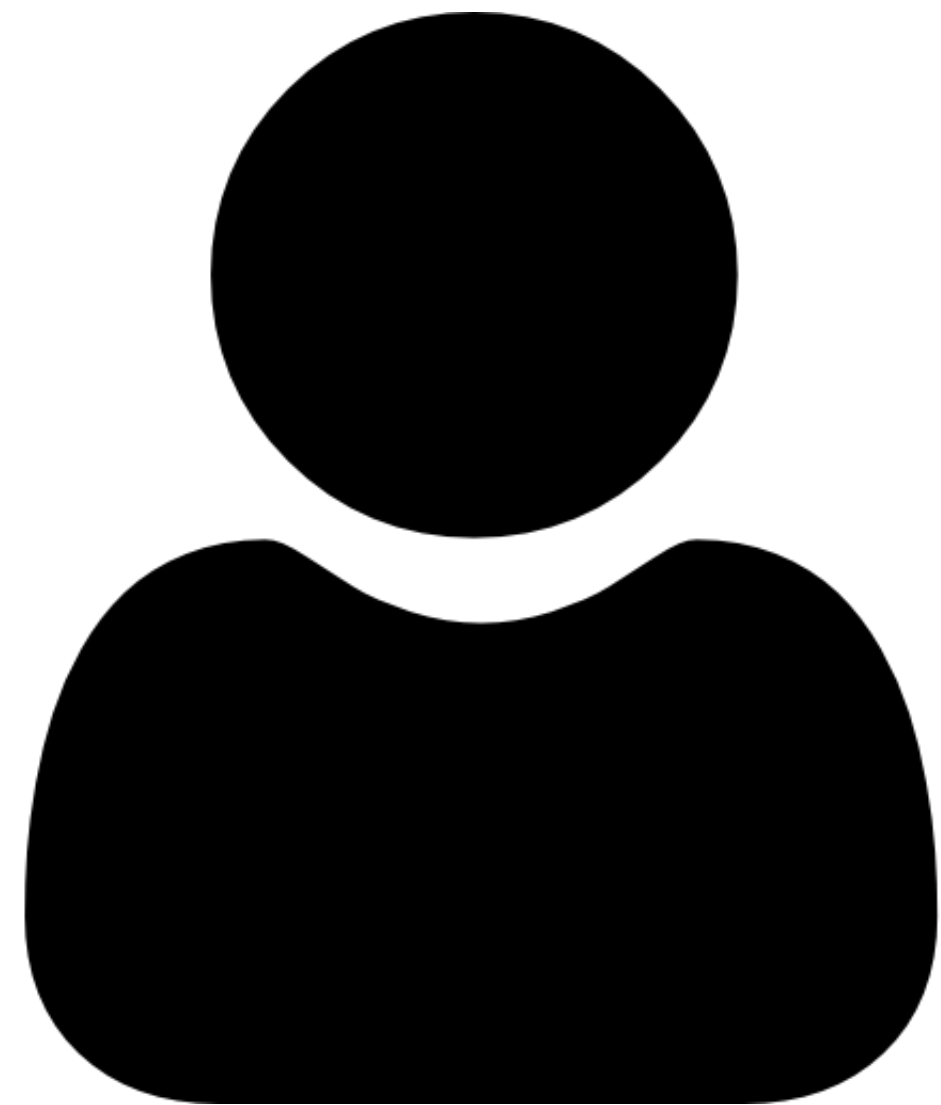


ТГ-канал курса

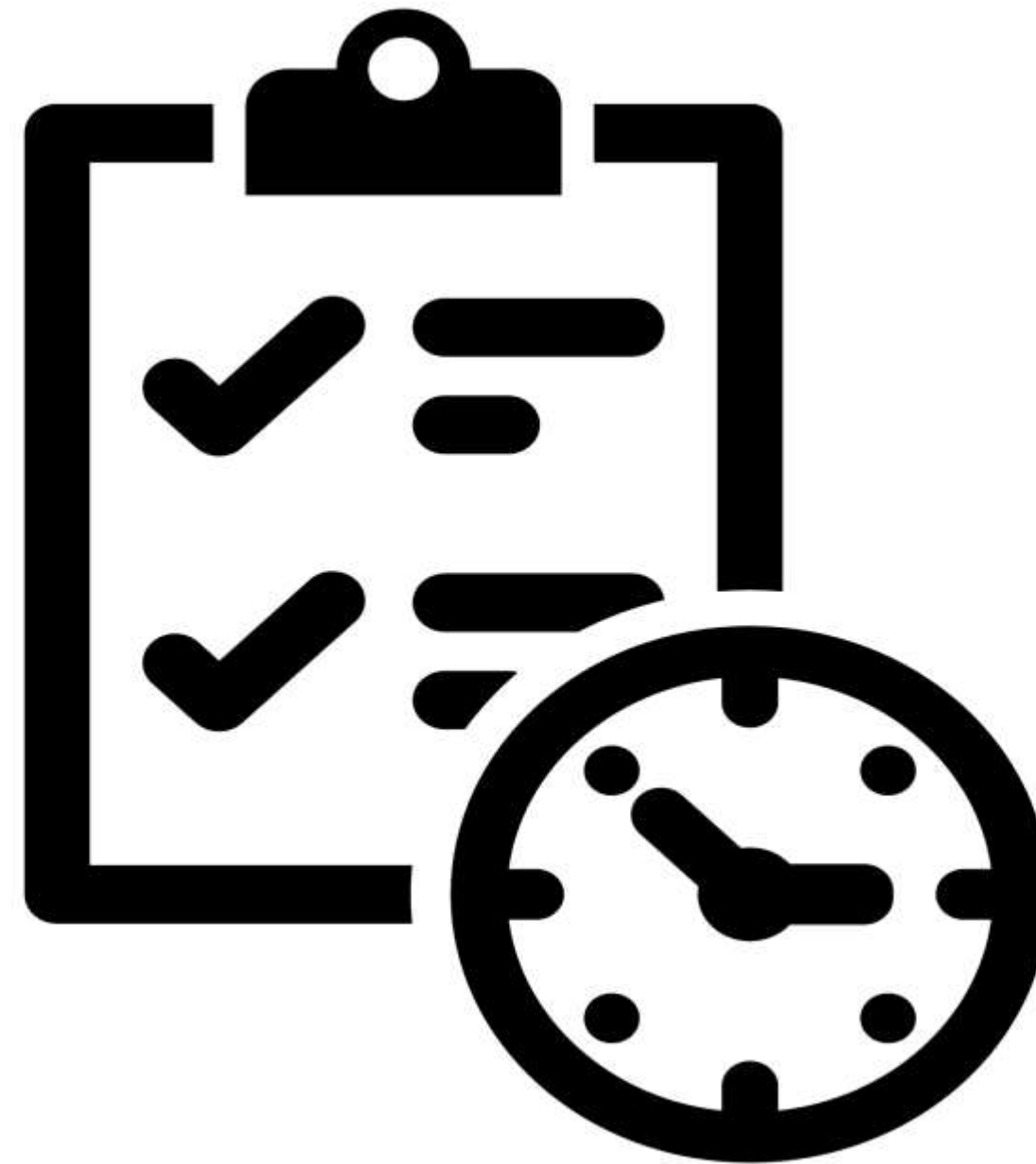
Основы Git

Проблема

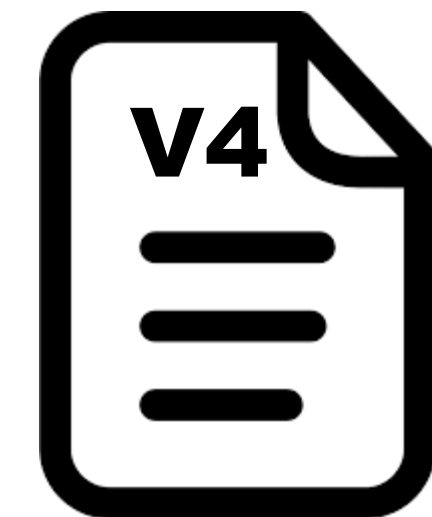
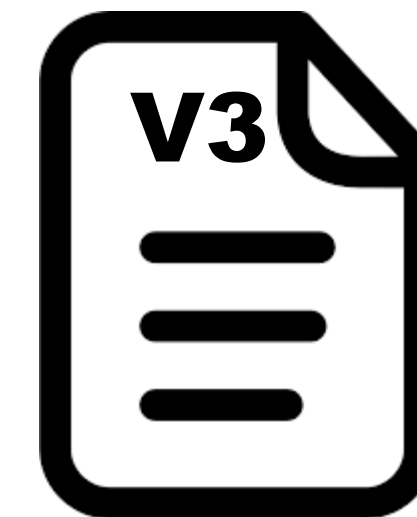
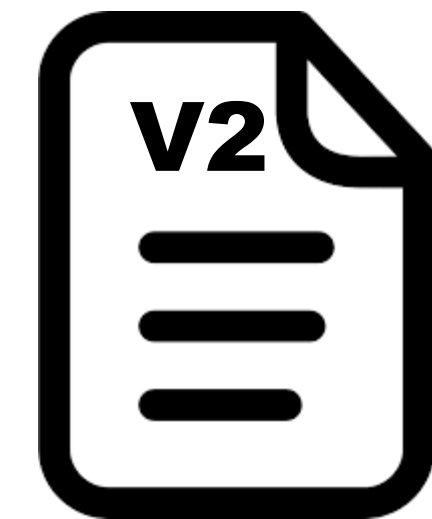
разработчик



проект



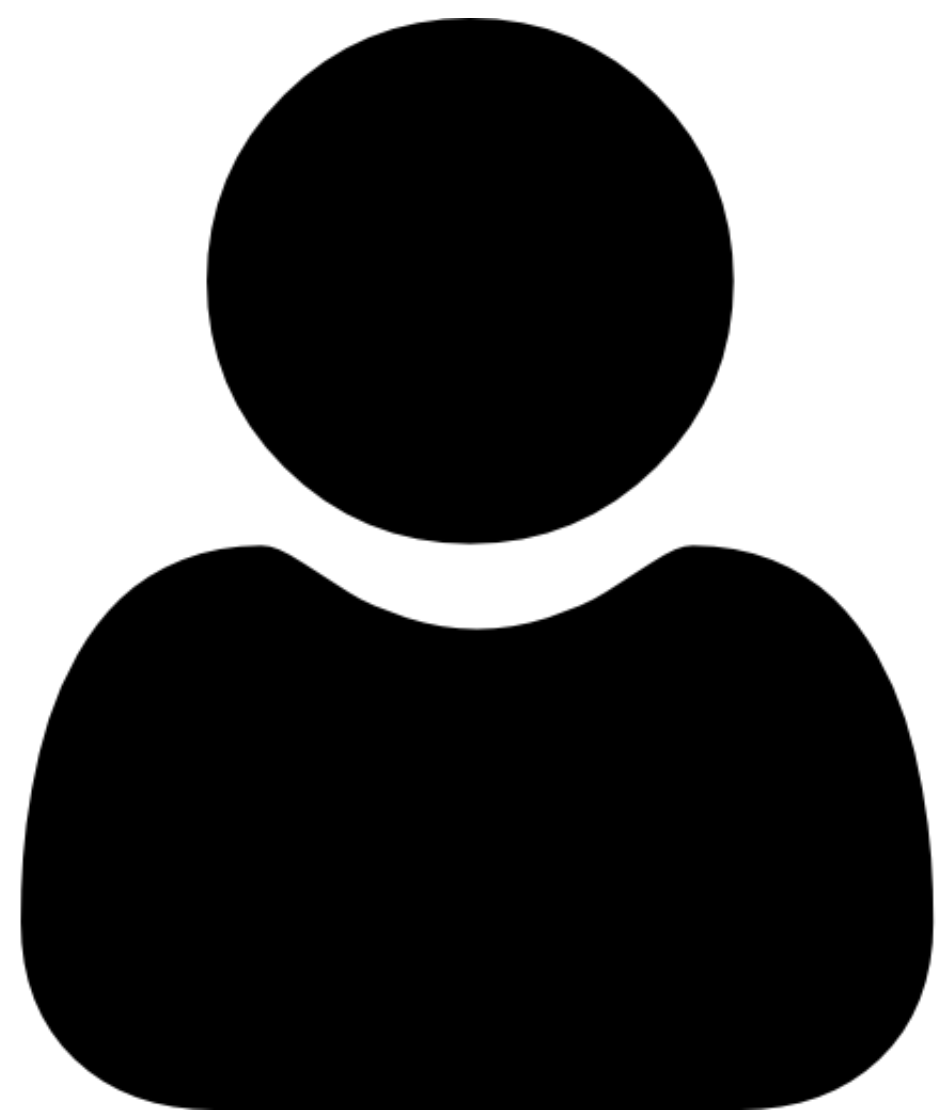
версии кода



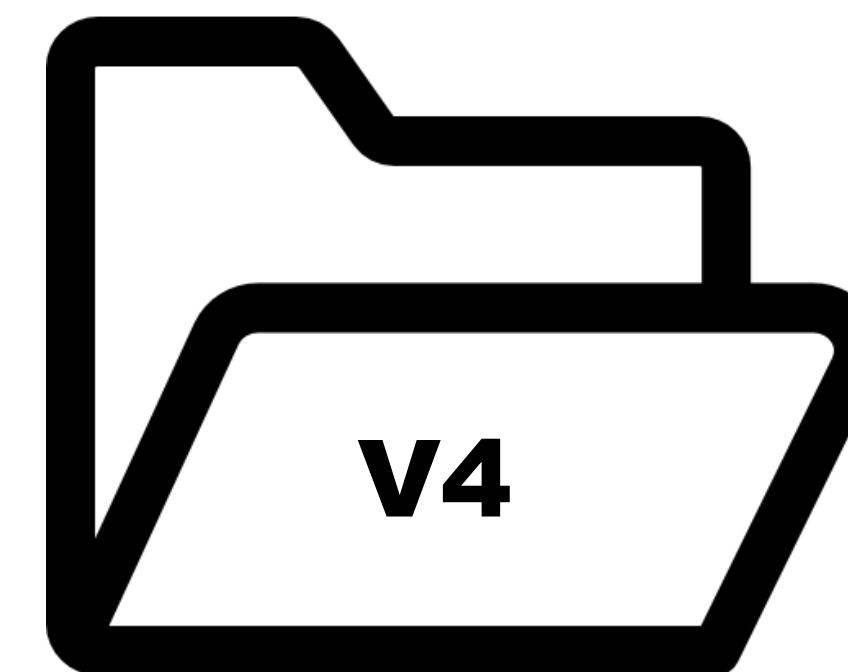
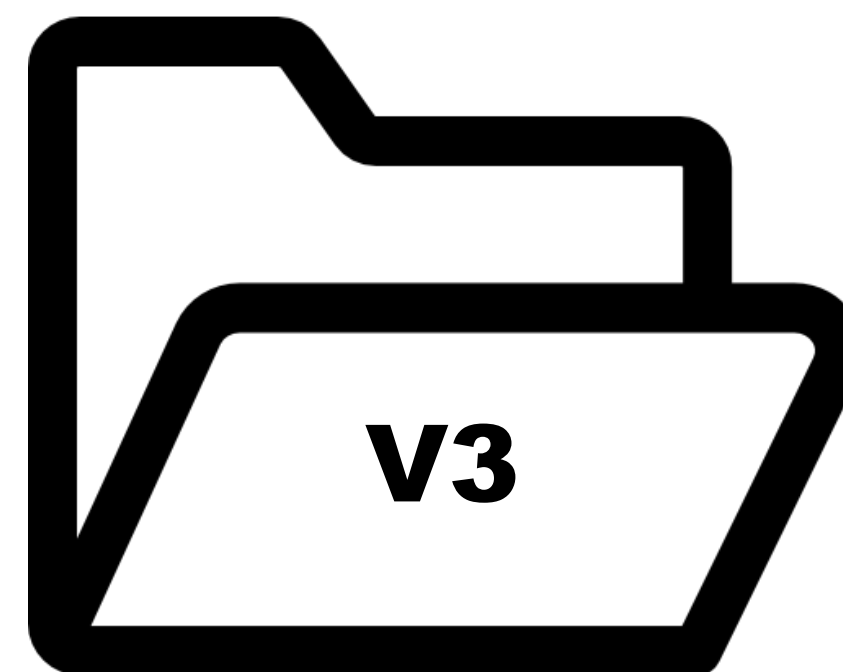
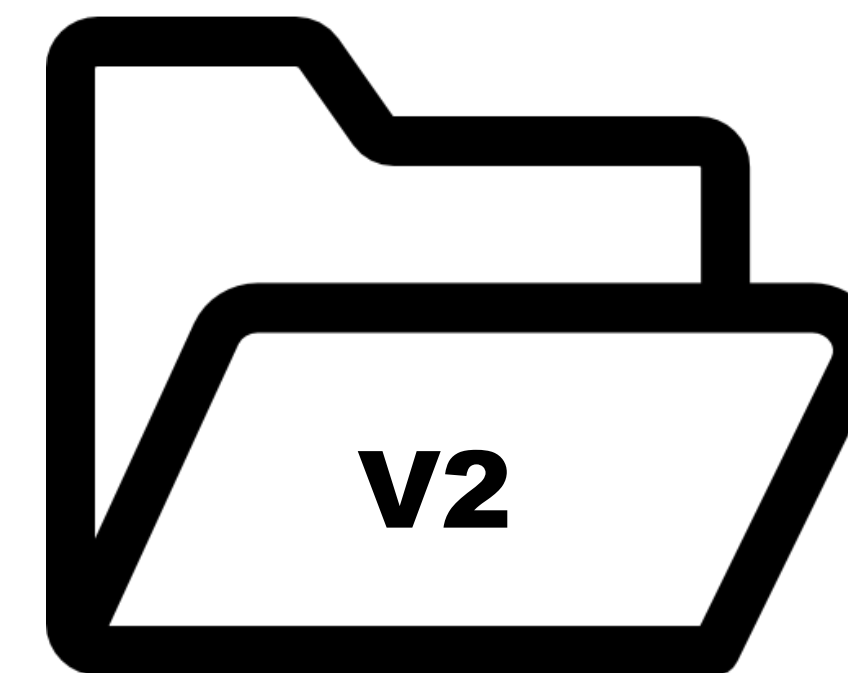
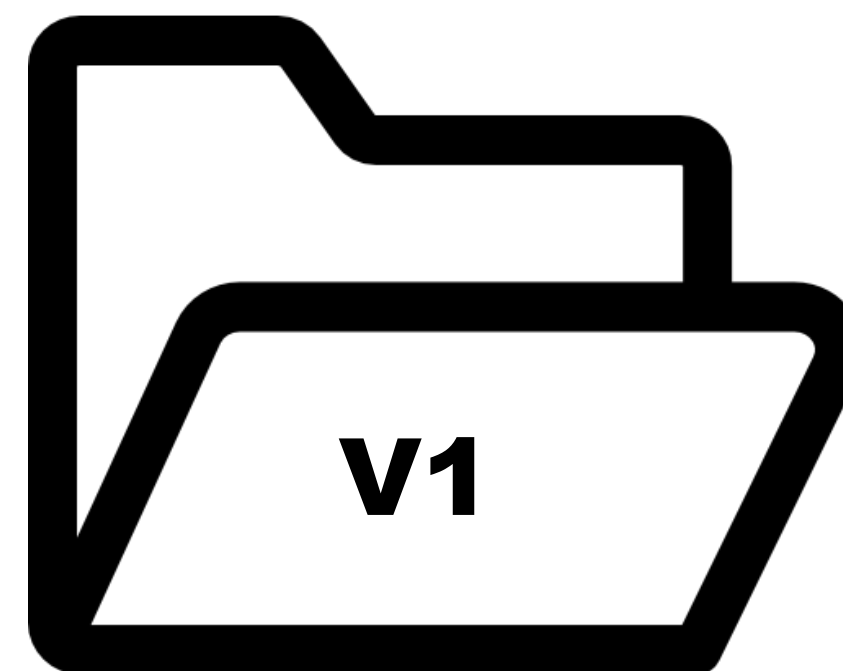
**Как быть с
версионированием?**

Локальное хранилище

разработчик

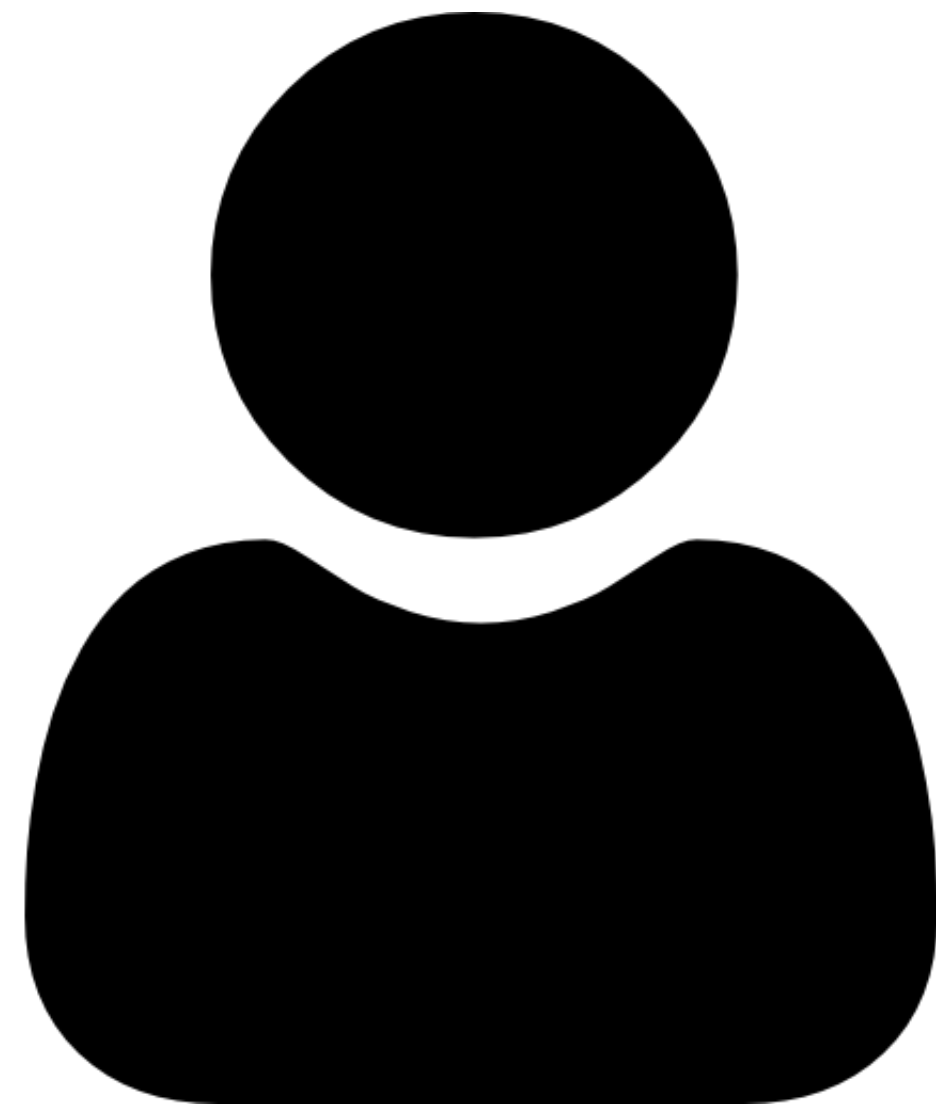


версии проекта



Локальная VCS

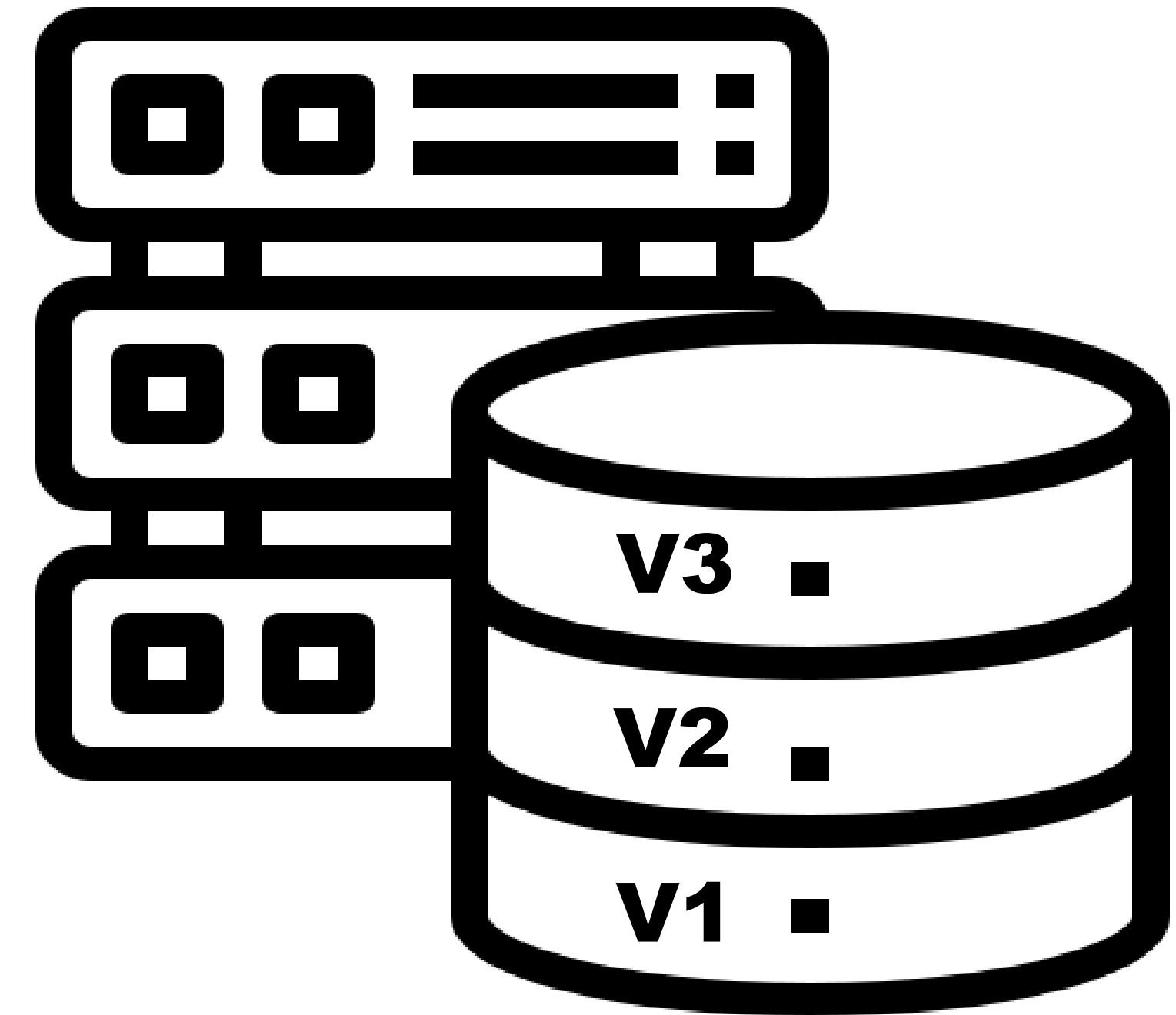
разработчик



код

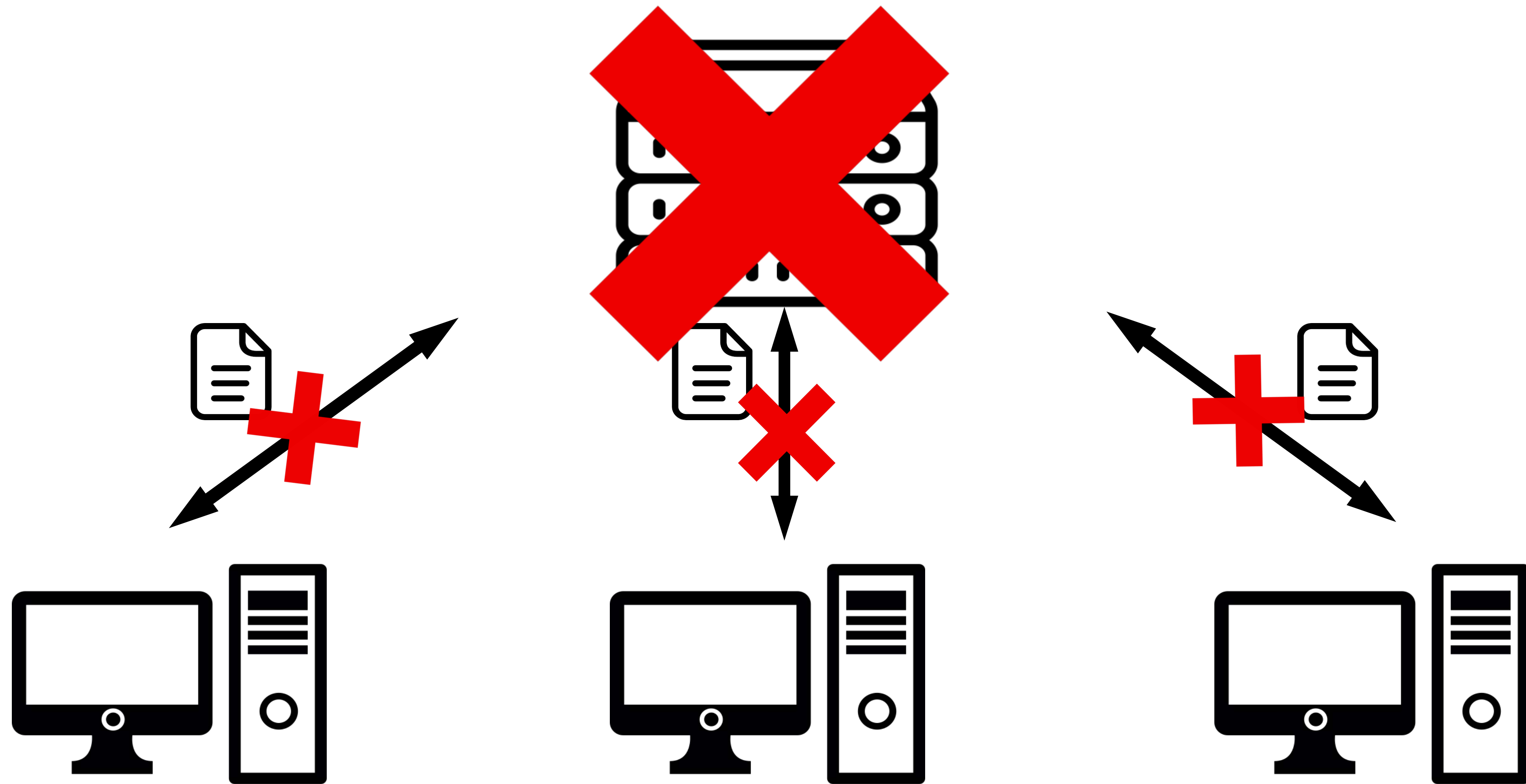


VCS

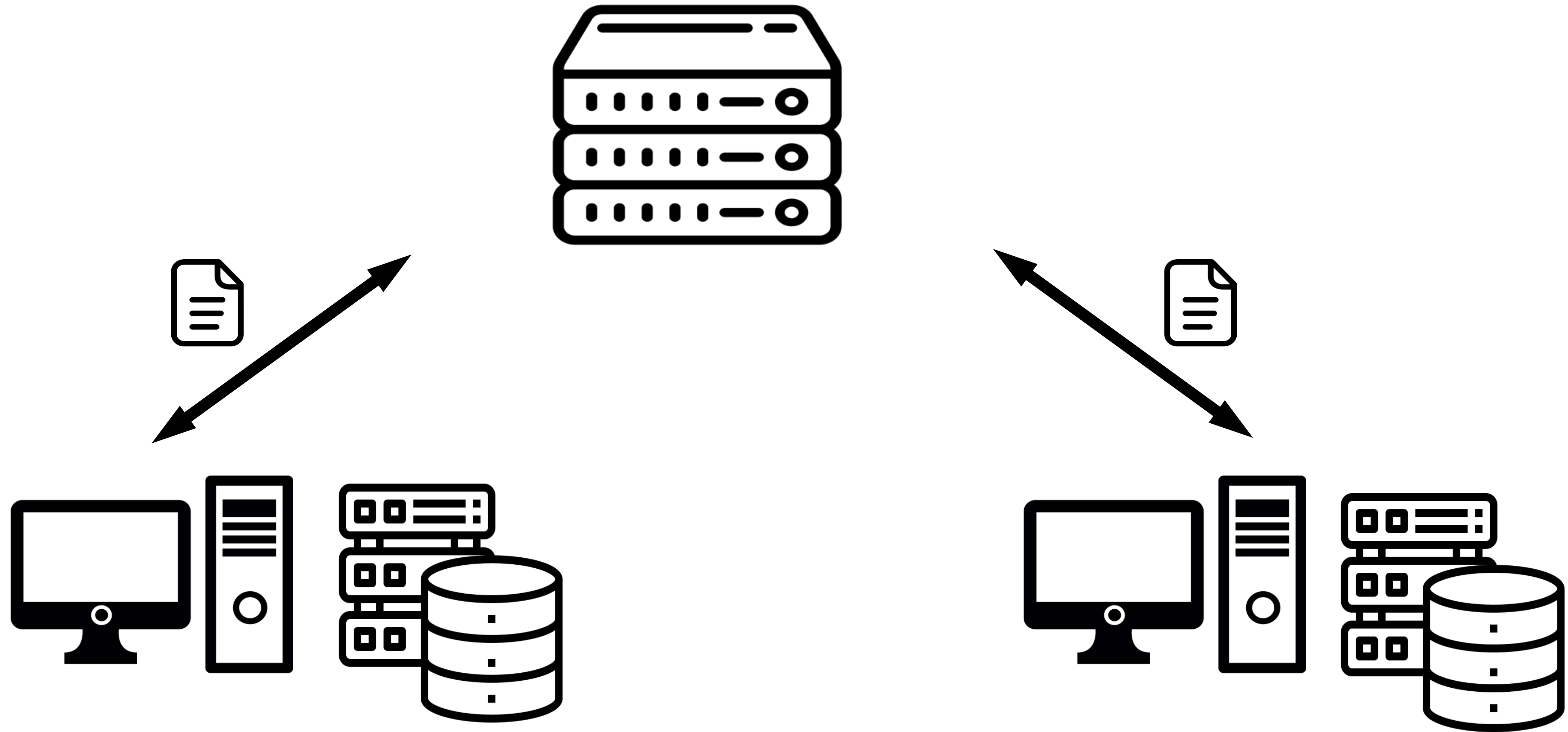


**А как быть если
разработчик не один?**

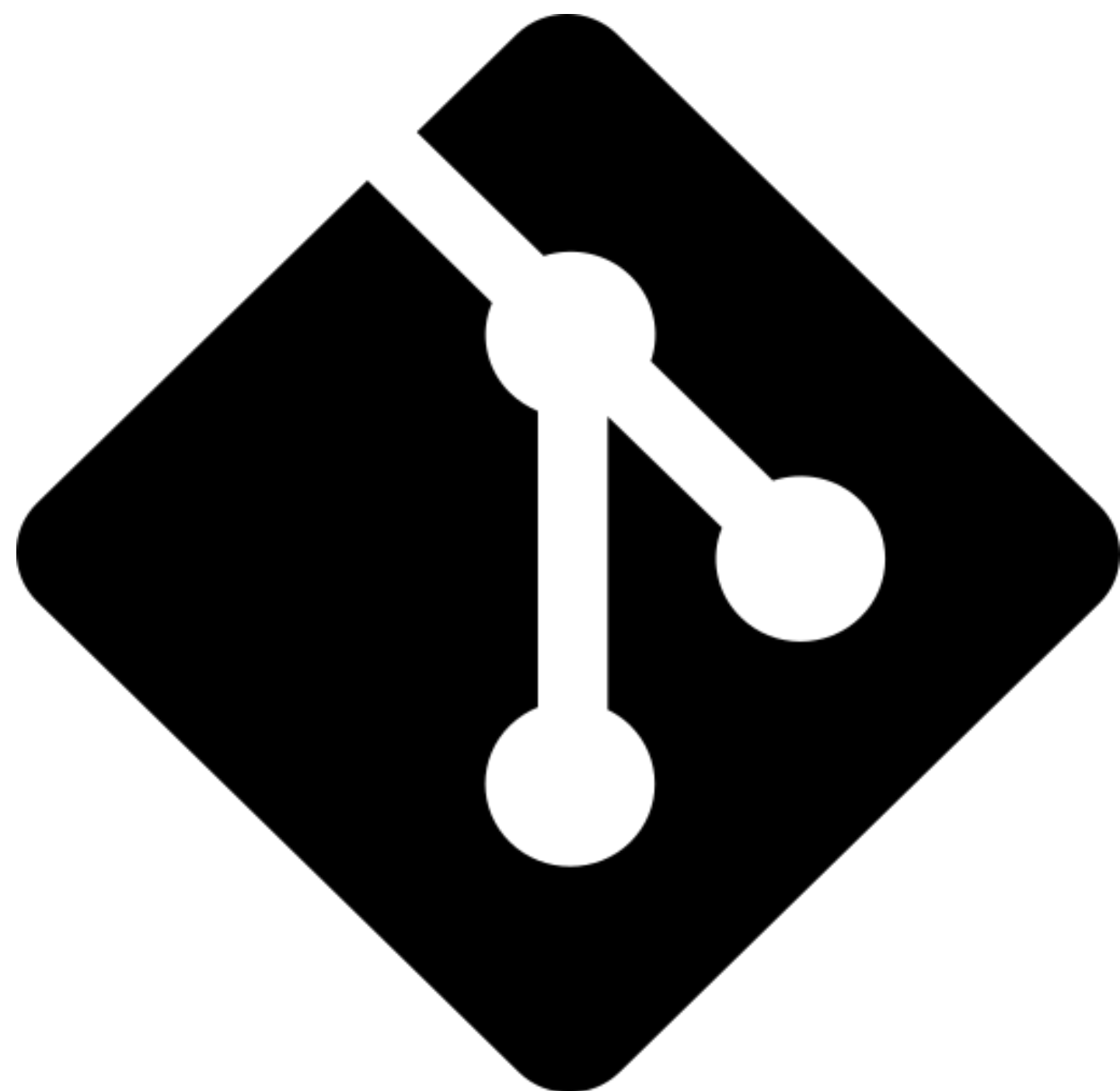
Централизованные VCS



Распределенные VCS



Git



git

Git - распределенная VCS

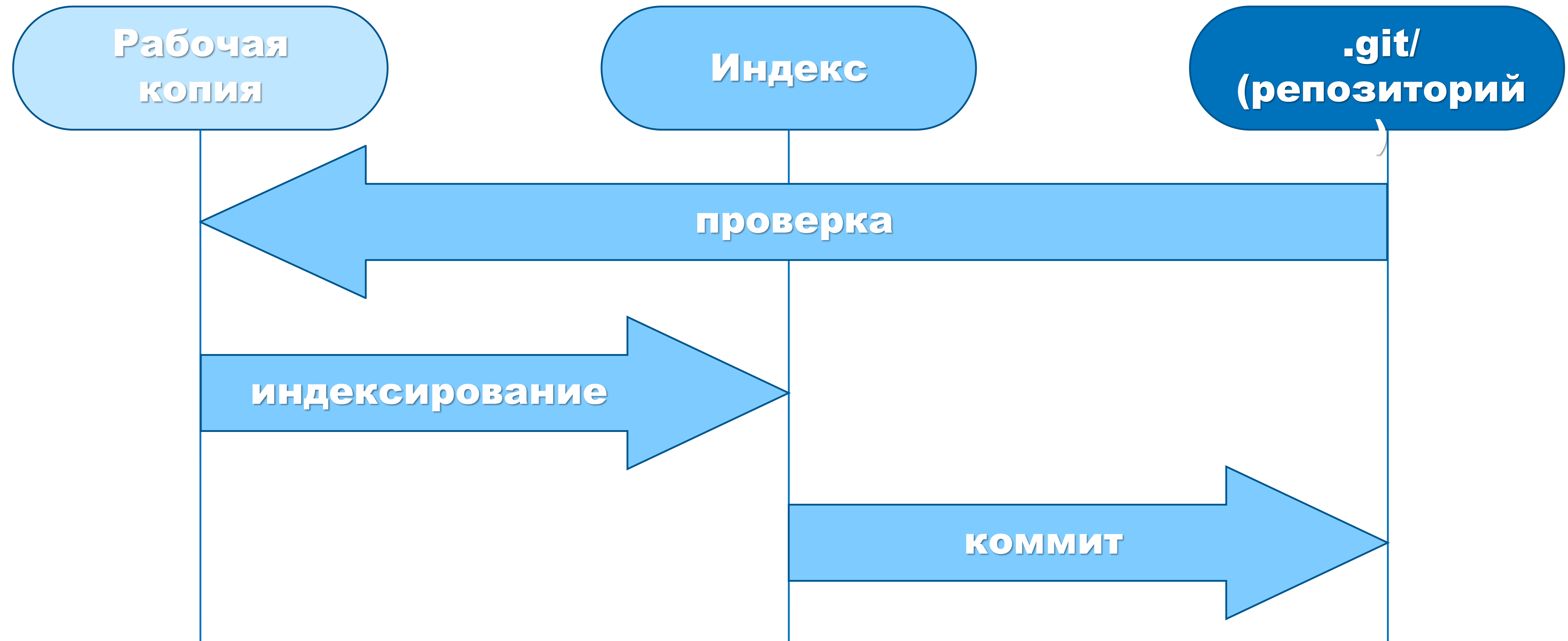
Репозиторий Git

Репозиторий Git - это хранилище изменений вашего проекта

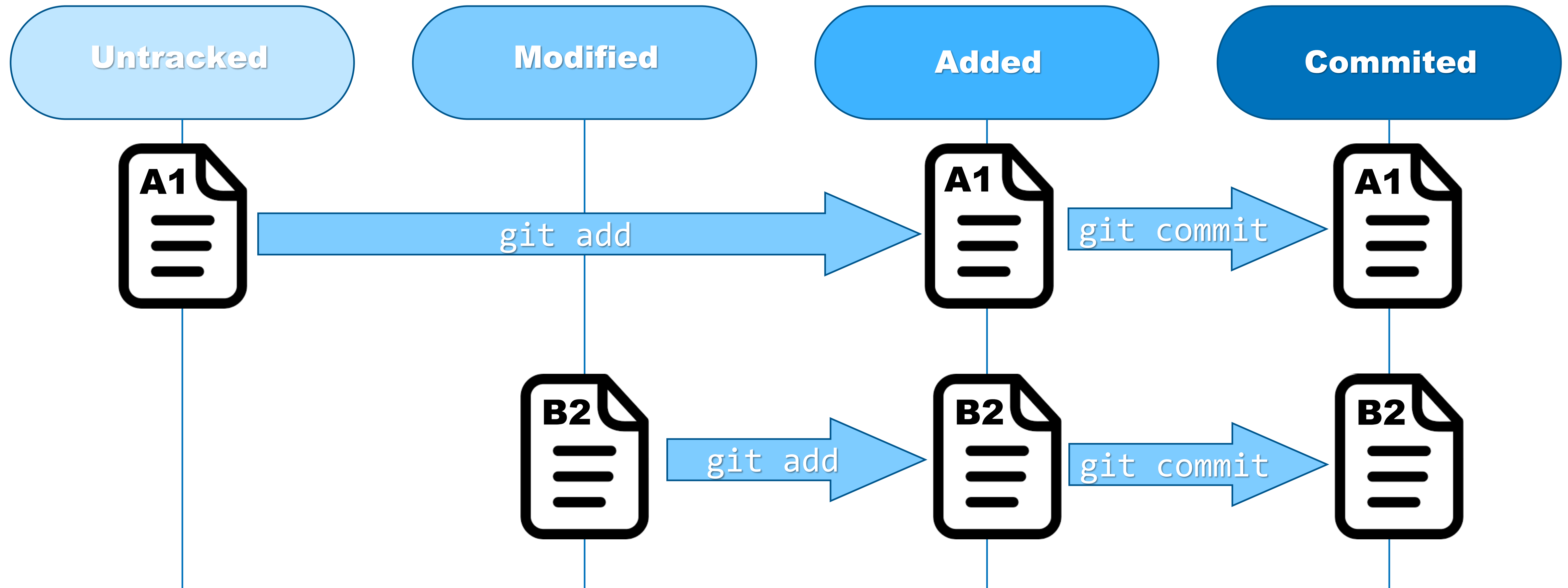
Git репозиторий может быть создан двумя способами:

- **`$ git init` - из текущего каталога;**
- **`$ git clone <url> [folder_name]` - клонирование существующего репозитория;**

Структура проекта Git



Основы работы с репозитием

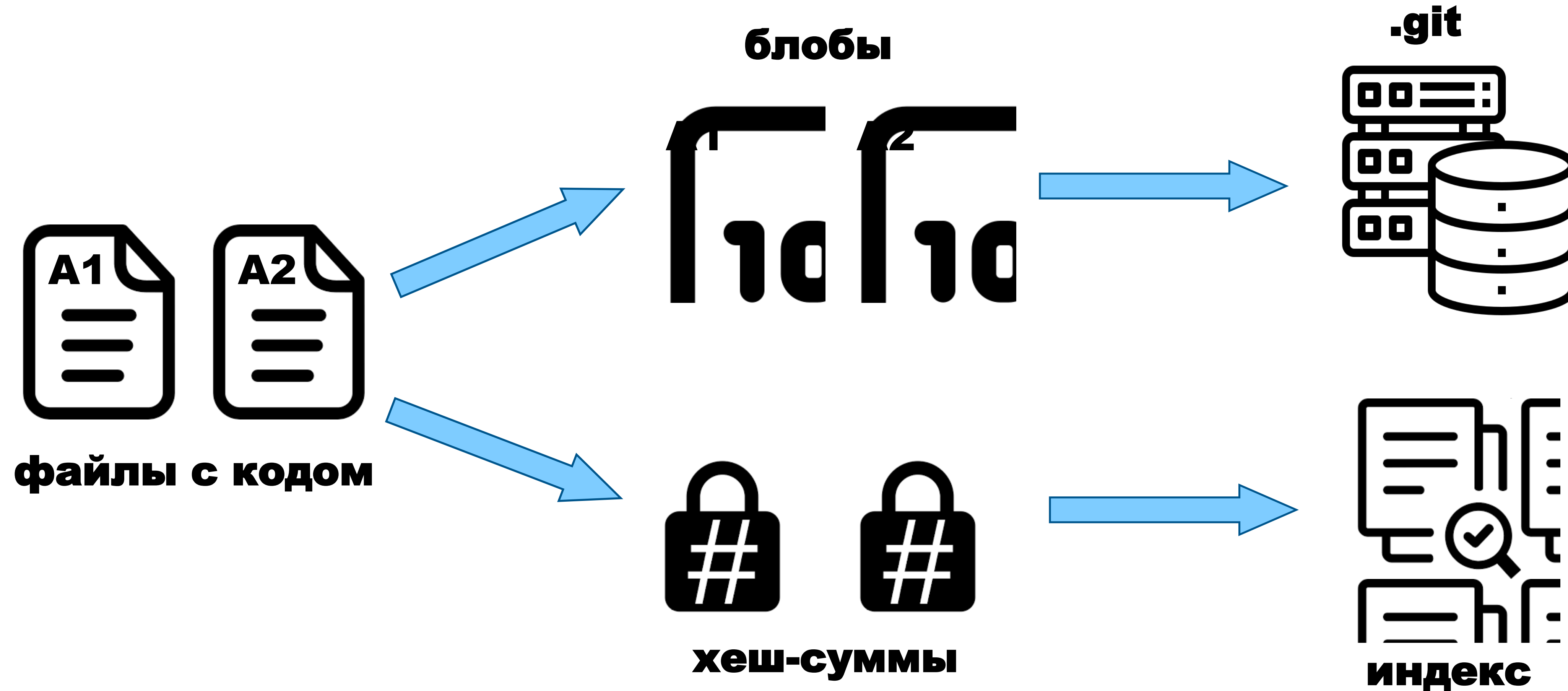


Запись изменений

- **Определение состояния файлов:**
 - `$ git status` - **подробный вывод;**
 - `$ git status -s` - **сокращенный вывод;**
- `$ git add <file>` - **индексация файлов ;**
- `$ git commit -m "useful comment"` - **КОММИТ ИЗМЕНЕНИЙ;**
- **Удаление файлов:**
 - `$ git rm <file>` - **перманентное;**
 - `$ git rm --cached <file>` - **из индекса;**

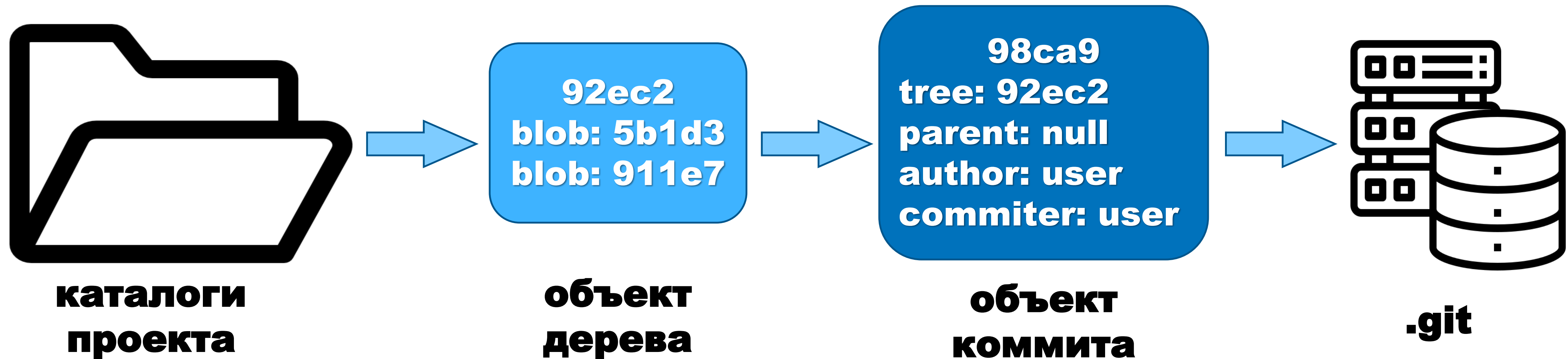
Как Git хранит данные?

```
git add A1 A2
```

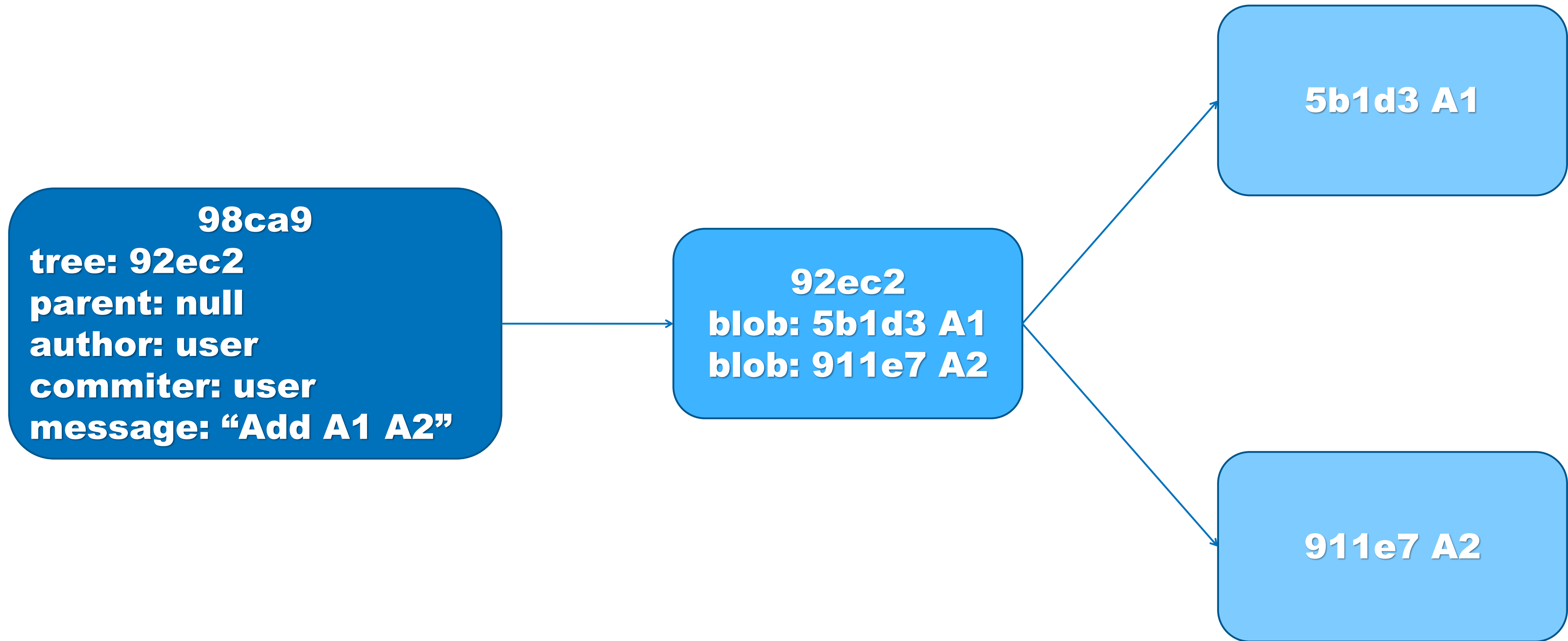


Как Git хранит данные?

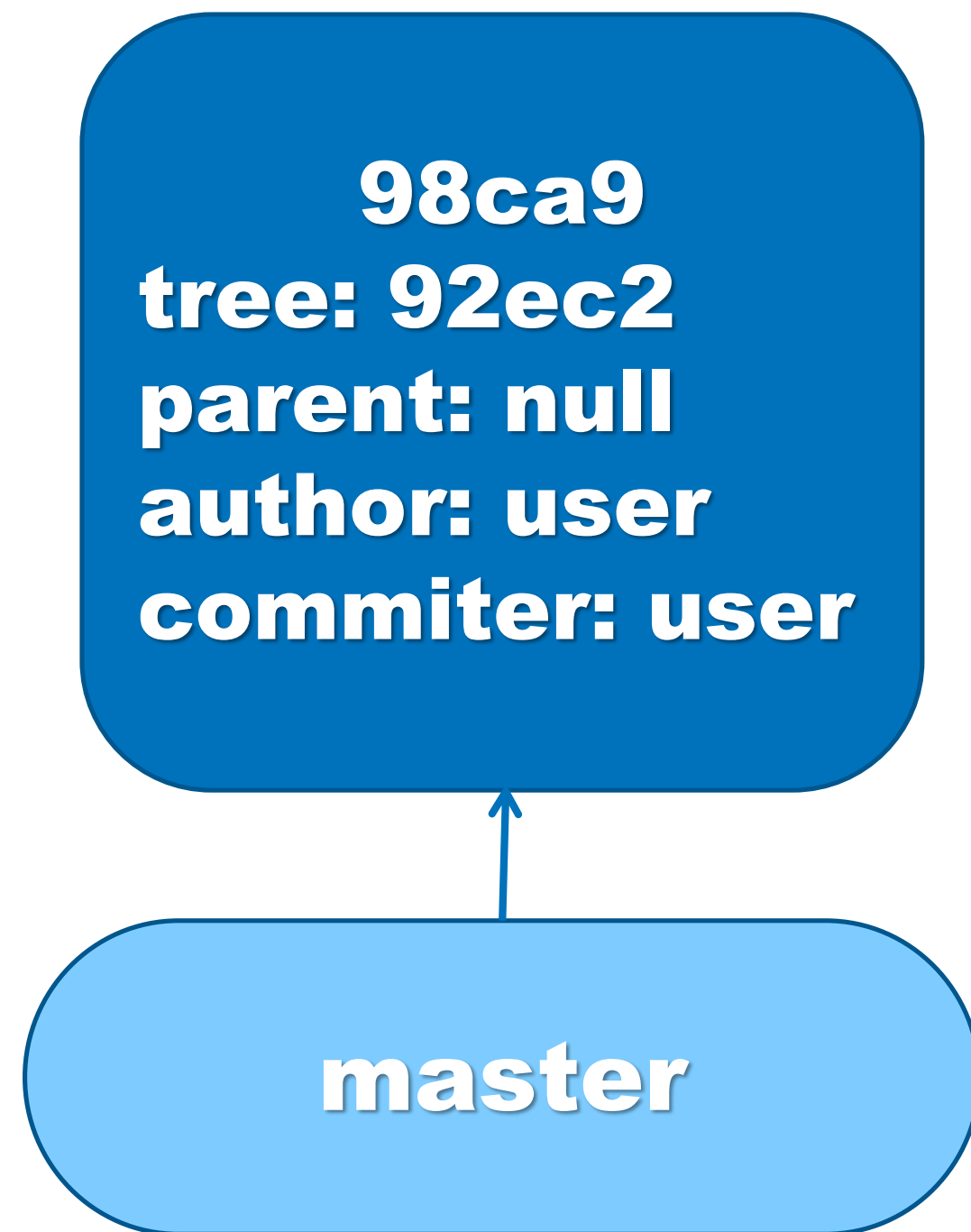
`git commit`



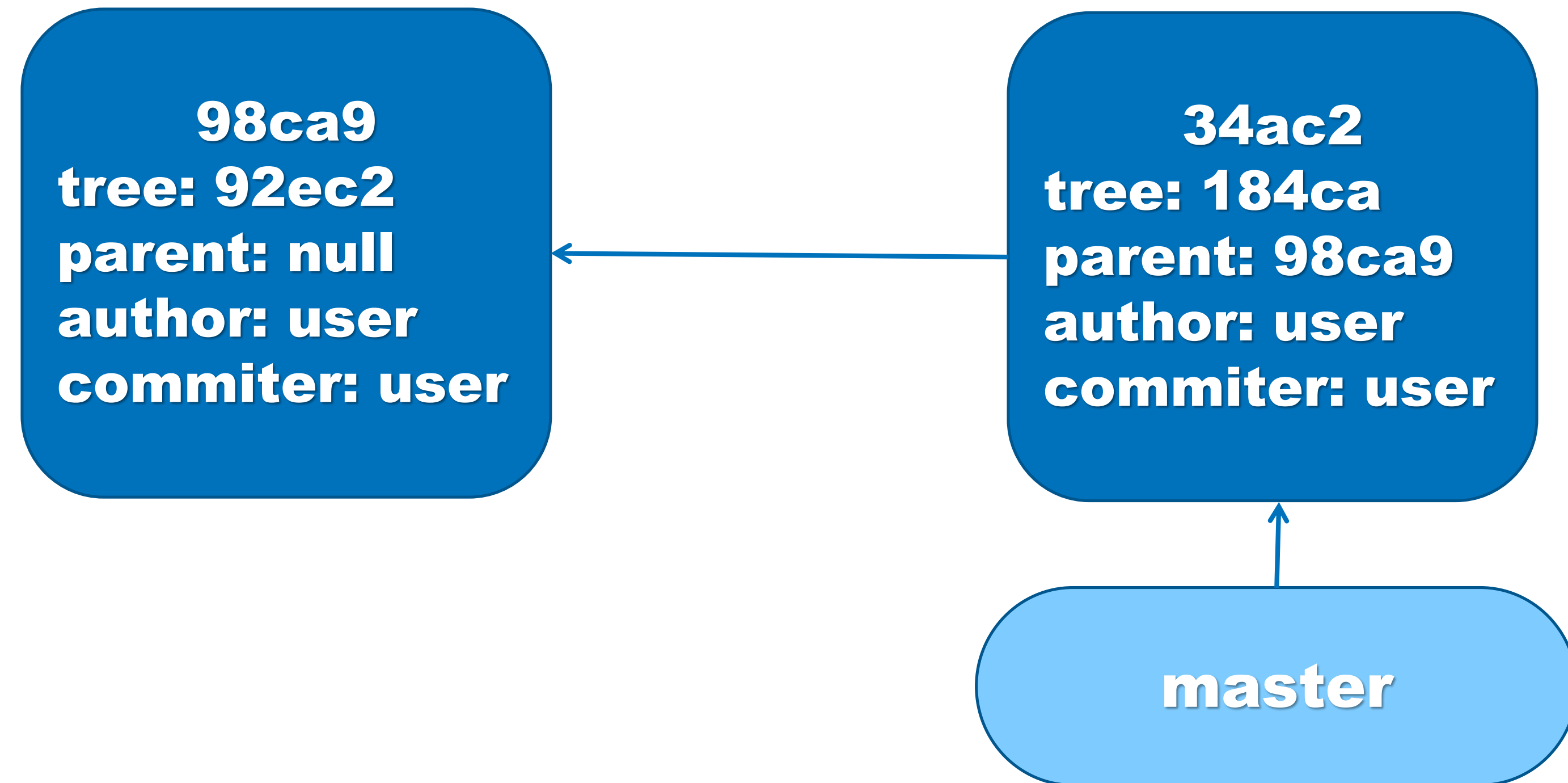
Структура обѣкта коммита



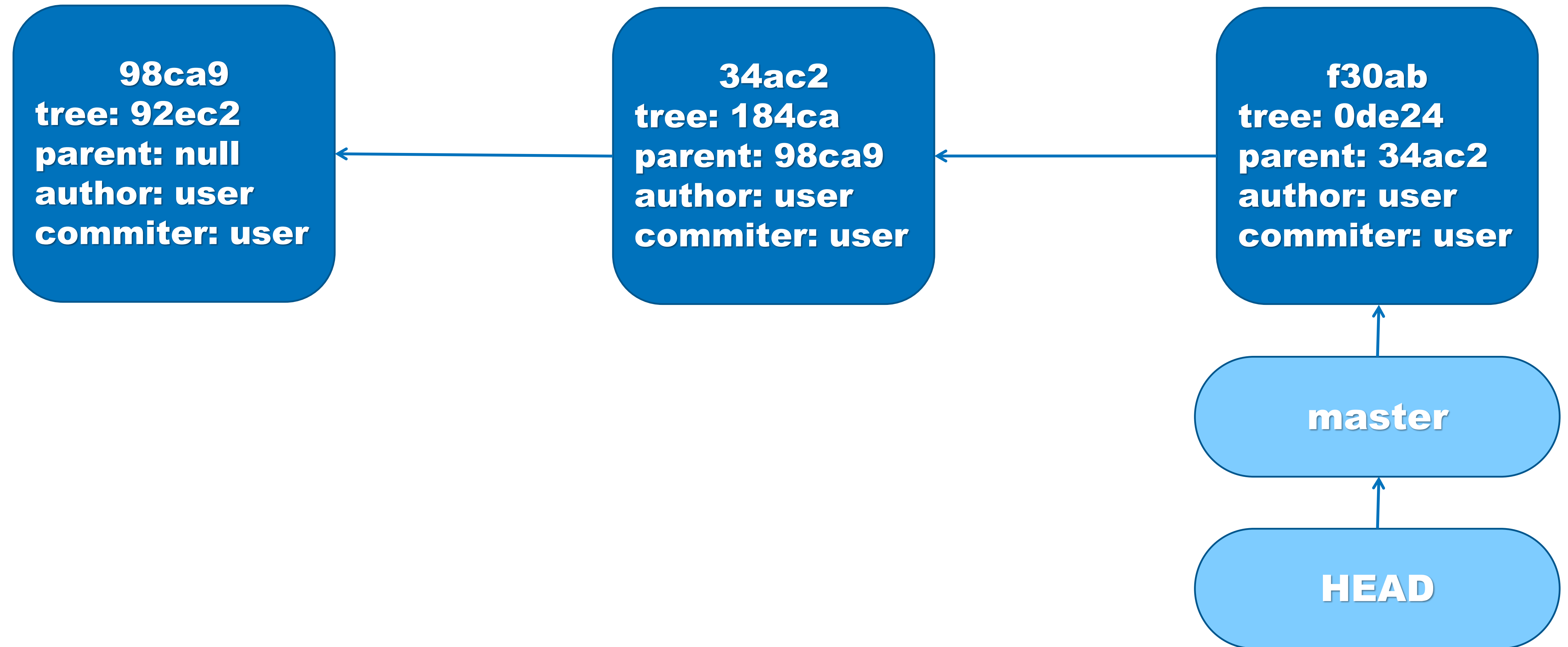
Что такое ветка?



Что такое ветка?



Что такое ветка?



Назад в прошлое

- **Просмотр истории изменений:**
 - `$ git log`
- **Переход к предыдущей версии:**
 - `$ git remote --soft <hash>` - откат до указанного коммита с сохранением добавленных изменений в индексе;
 - `$ git remote --mixed <hash>` - откат до указанного коммита, изменения останутся в файлах, но не будут внесены в индекс;
 - `$ git remote --hard <hash>` - откат до указанного коммита, все внесенные изменения будут утеряны;

Удаленные репозитории

- **Просмотр удаленных репозиториев:**
 - `$ git remote` - **без подробностей;**
 - `$ git remote -v` - **с подробностями;**
- `$ git remote add <repo_name> <url>` - **добавление репозиториев;**
- **Получение изменений:**
 - `$ git fetch <remote-name>` - **без слияния;**
 - `$ git pull` - **со слиянием;**
- `$ git push <remote-name> <branch-name>` - **отправка изменений;**
- `$ git remote remove <repo_name>` - **удаление:**

Практическая часть