



Python как первый язык

День 5

Инфраструктура

Что такое инфраструктура?



Что такое инфраструктура?

версионирование



Что такое инфраструктура?

версионирование



тестирование

Что такое инфраструктура?

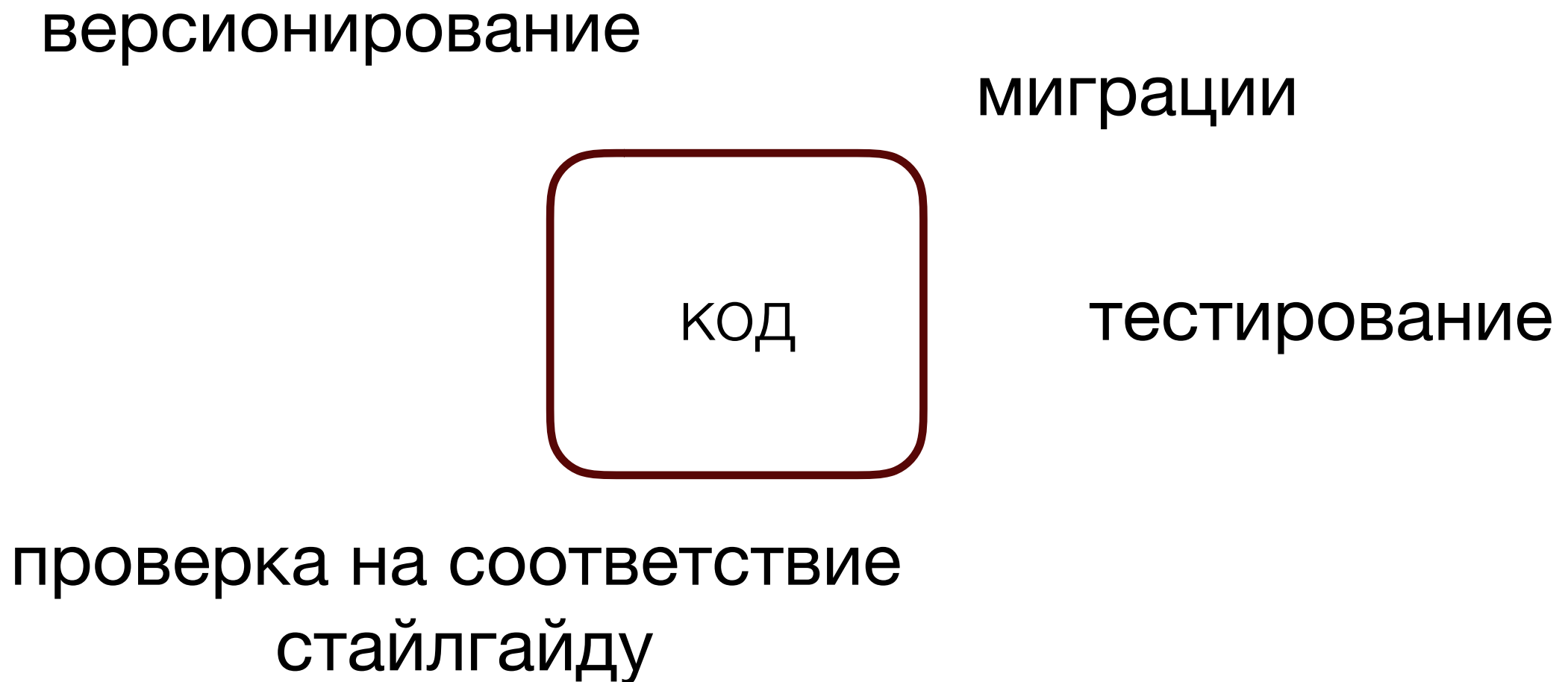
версионирование



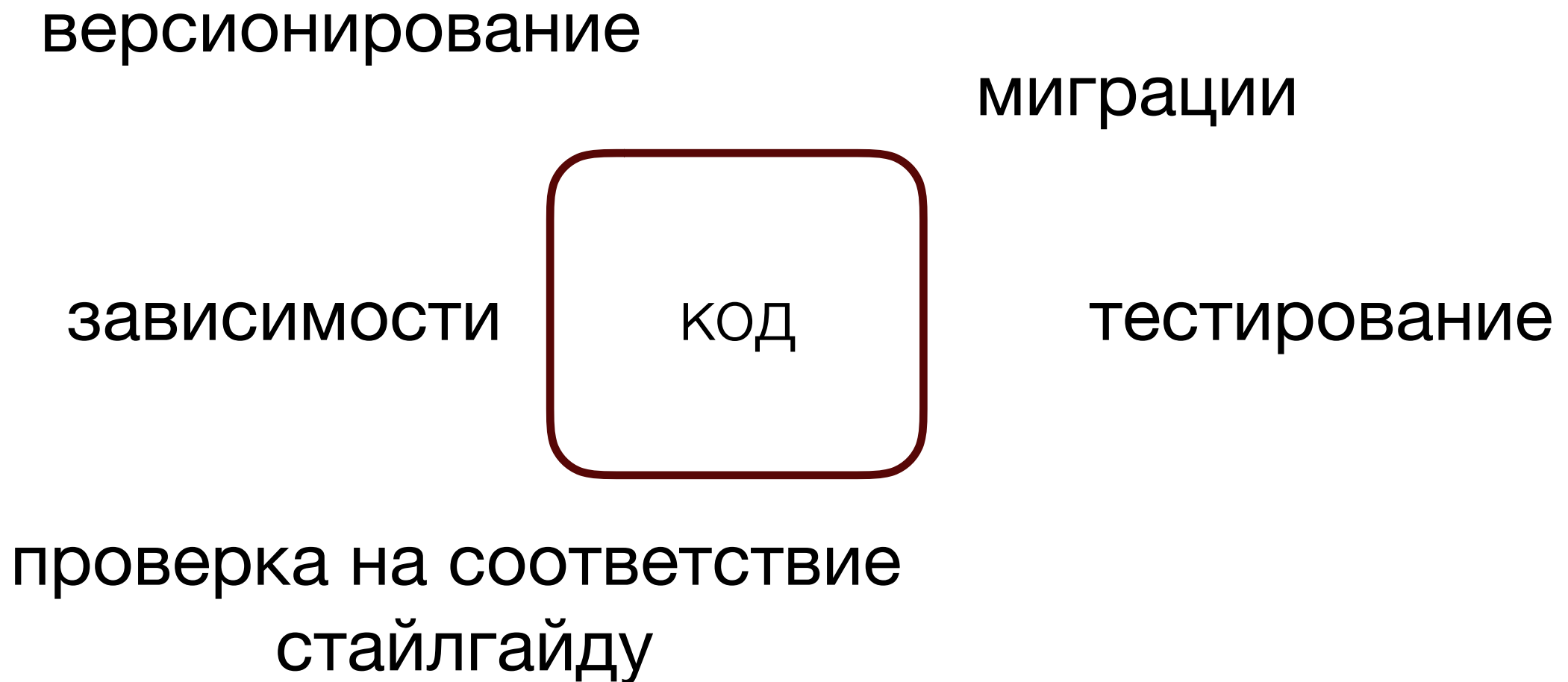
тестирование

проверка на соответствие
стайлгайду

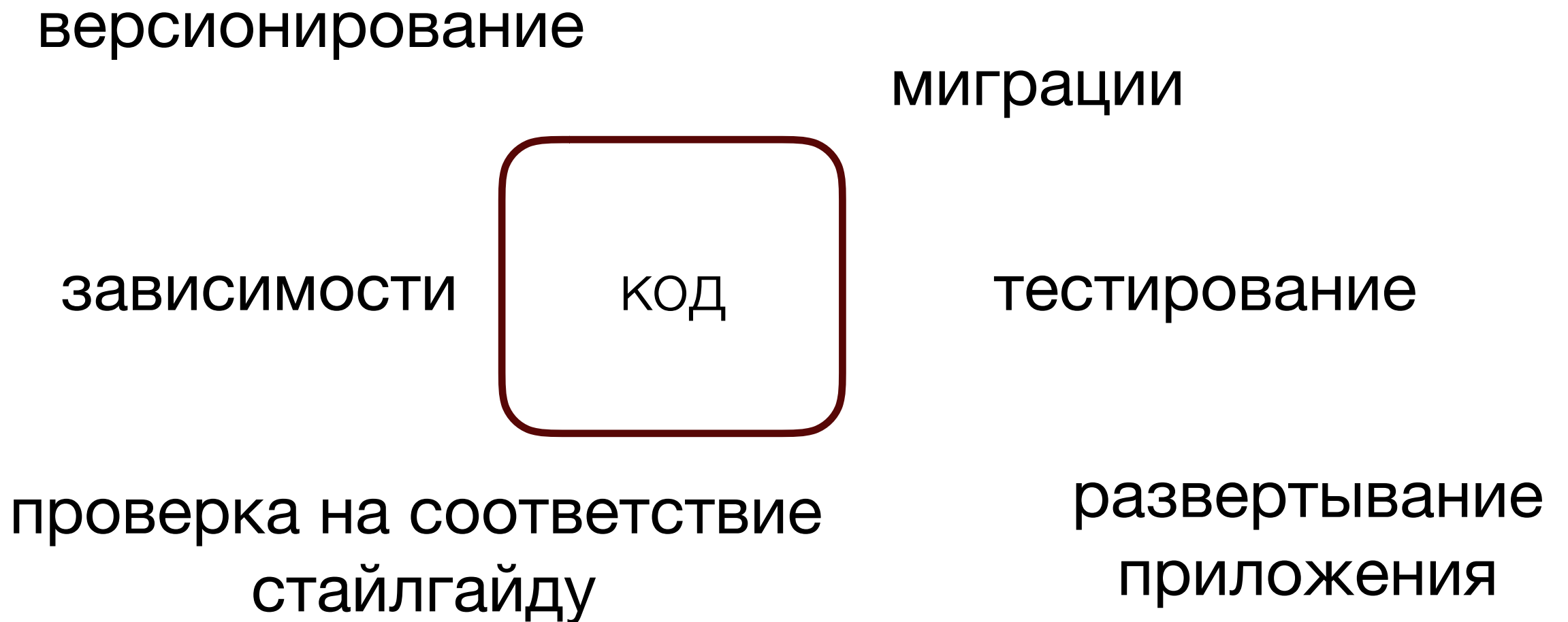
Что такое инфраструктура?



Что такое инфраструктура?



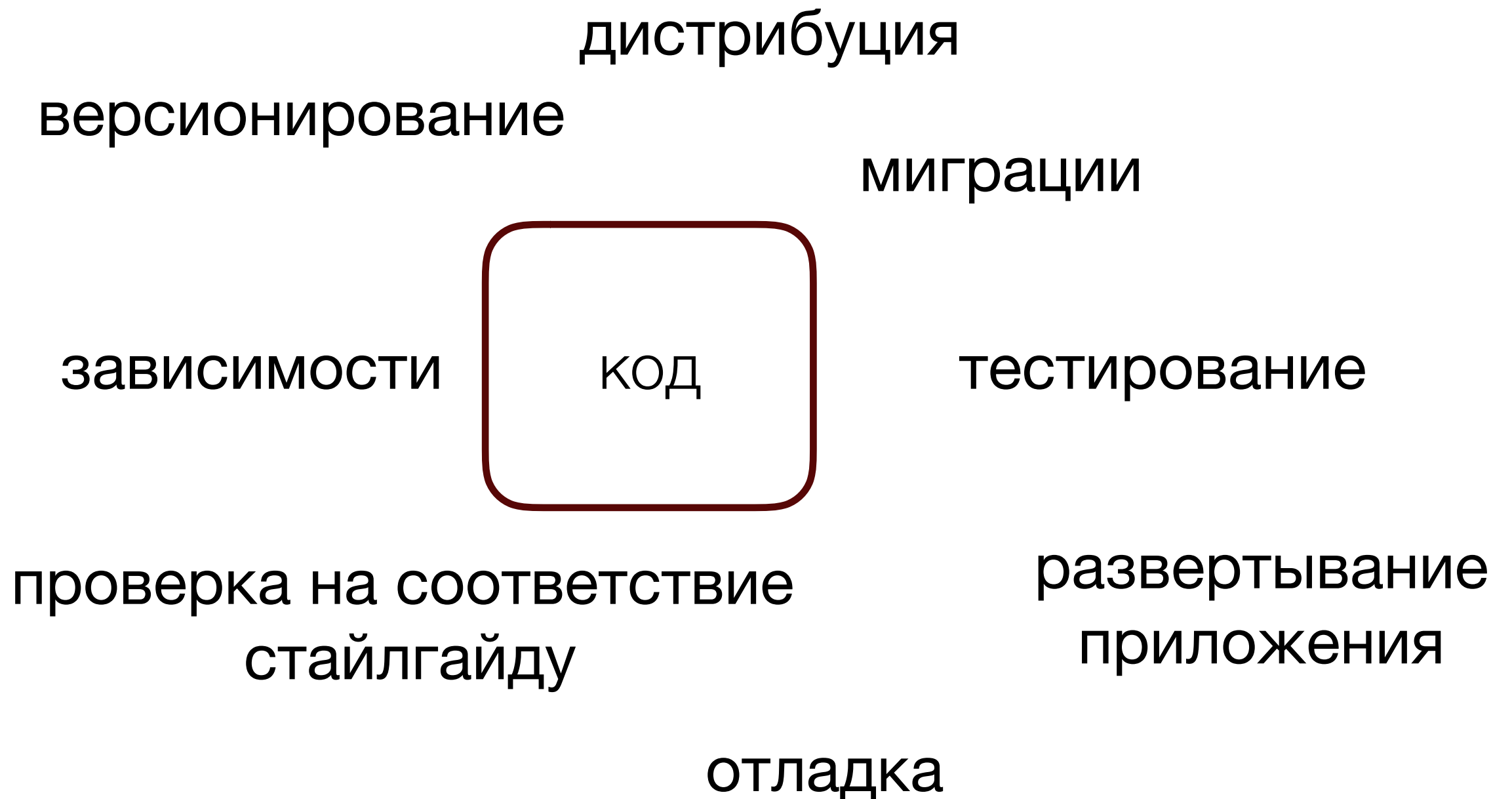
Что такое инфраструктура?



Что такое инфраструктура?



Что такое инфраструктура?



Что такое инфраструктура?



Что такое инфраструктура?



Что такое инфраструктура?

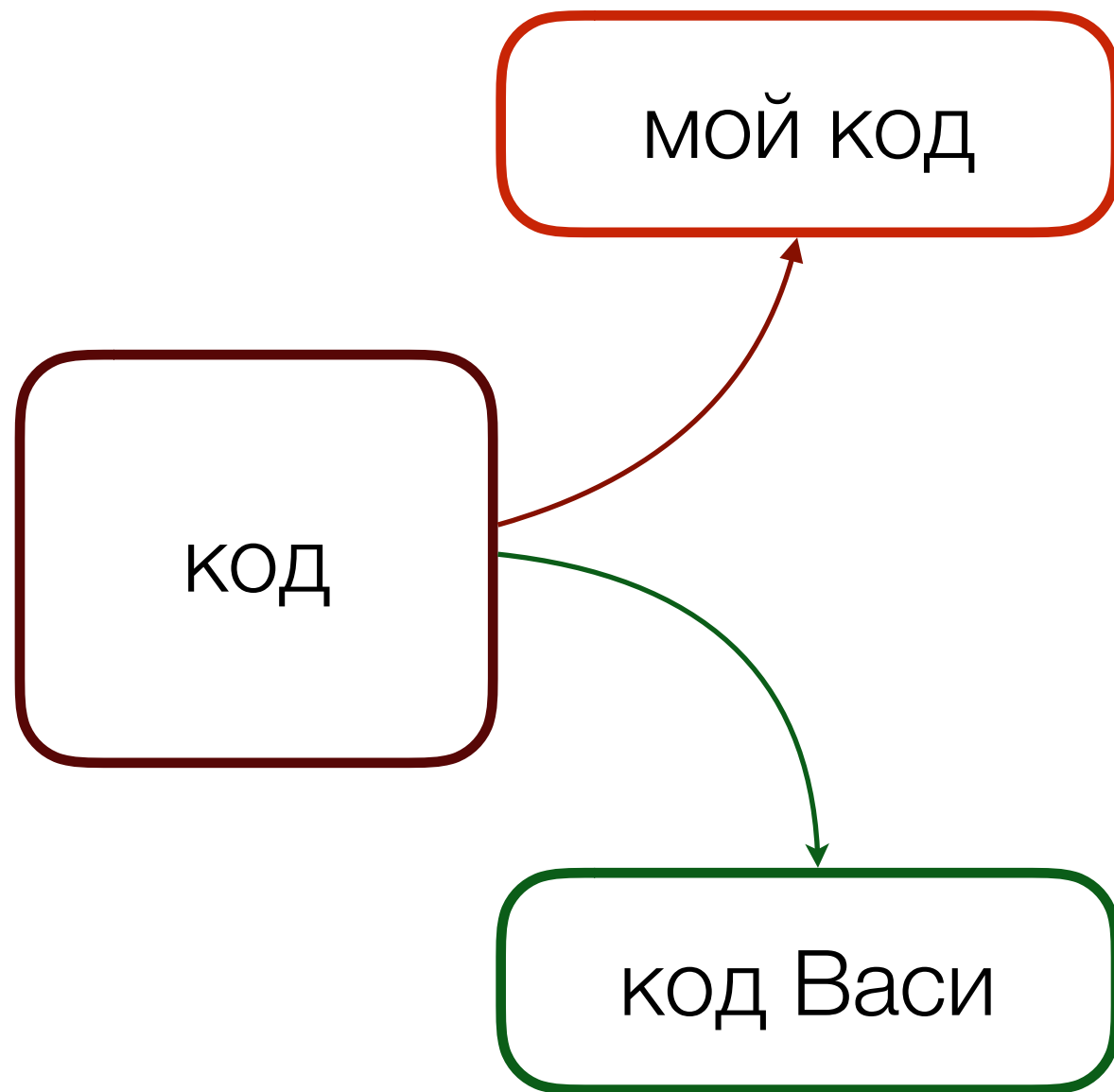


Версионирование

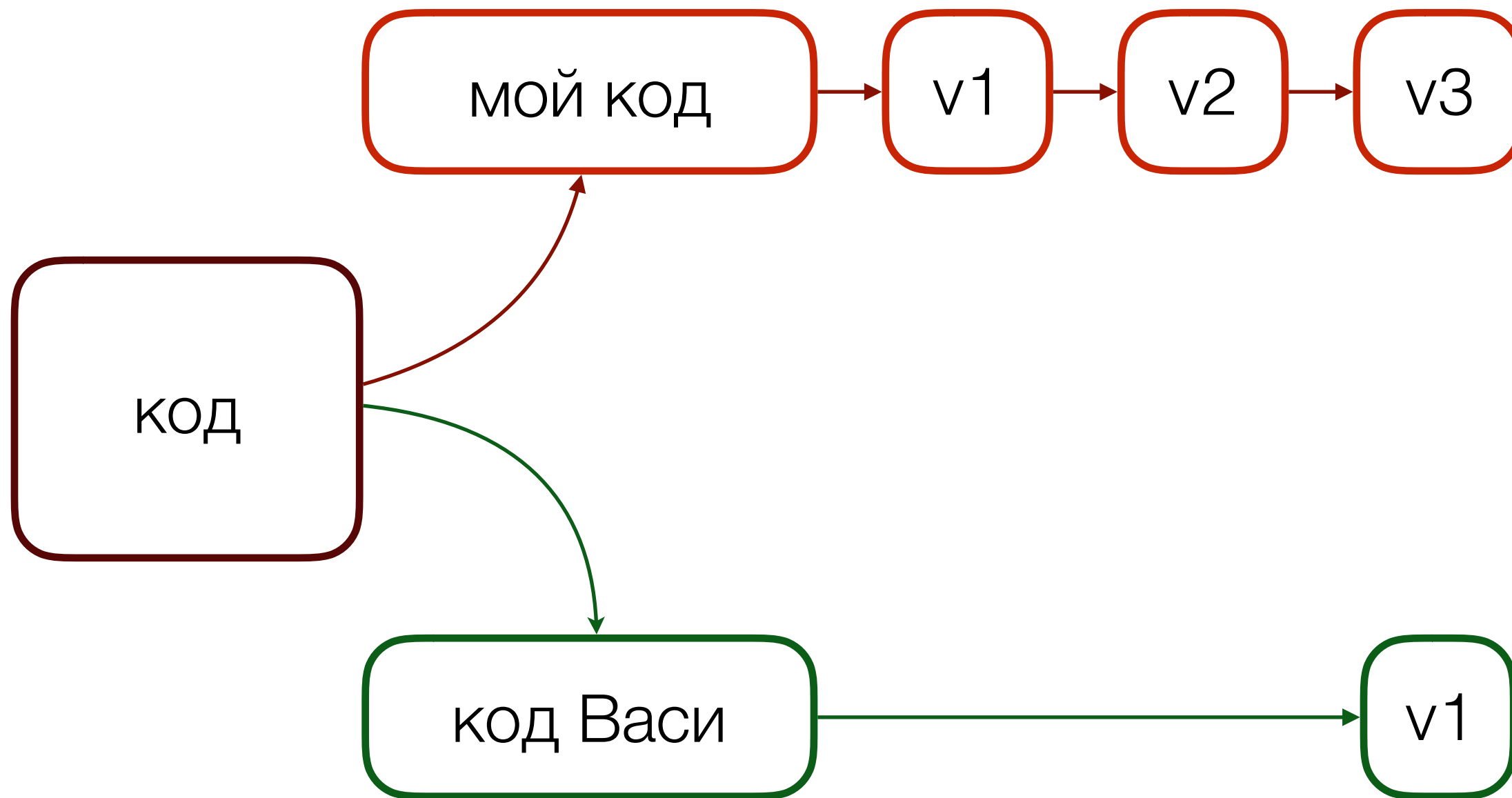


КОД

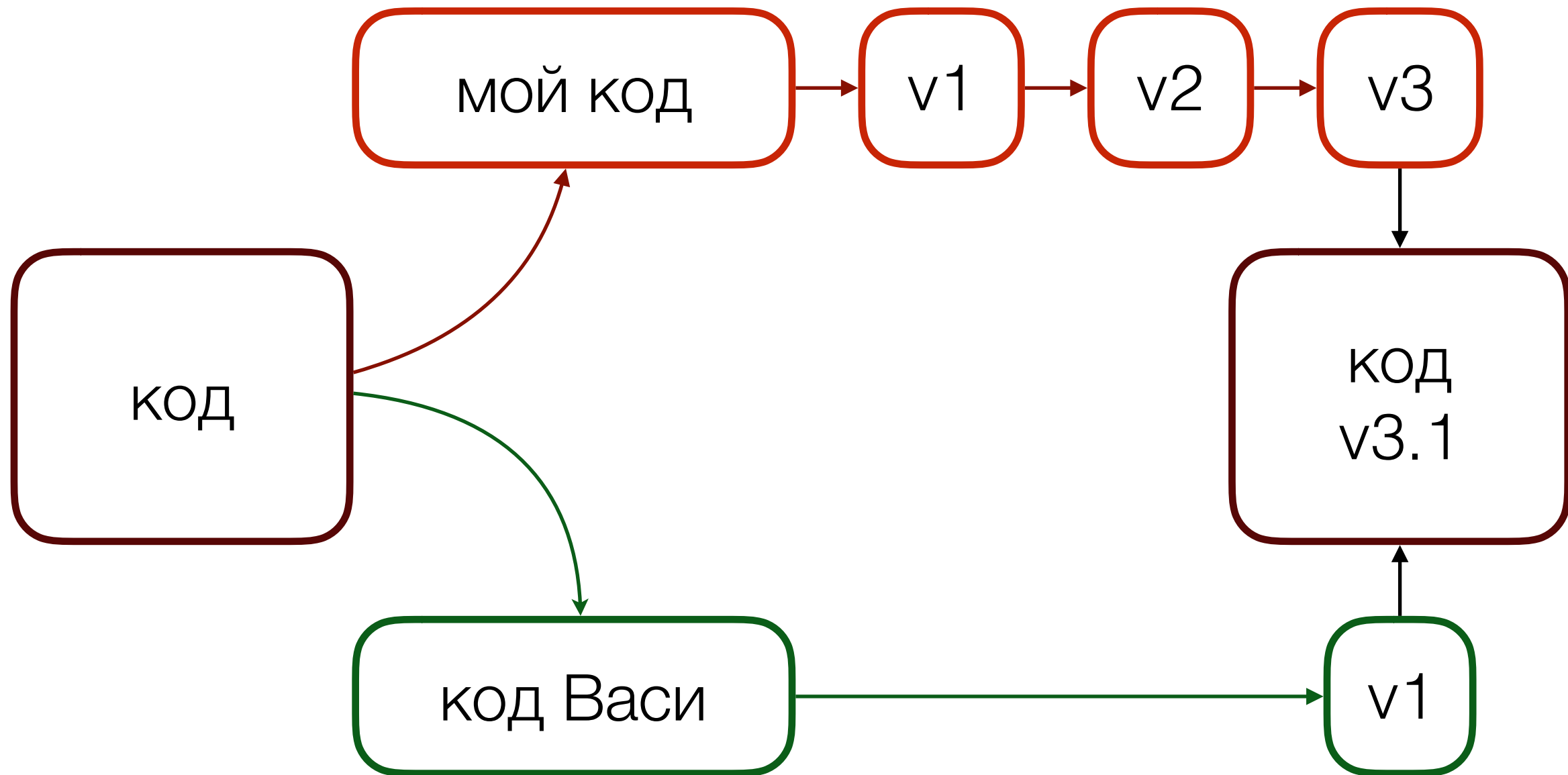
Версионирование



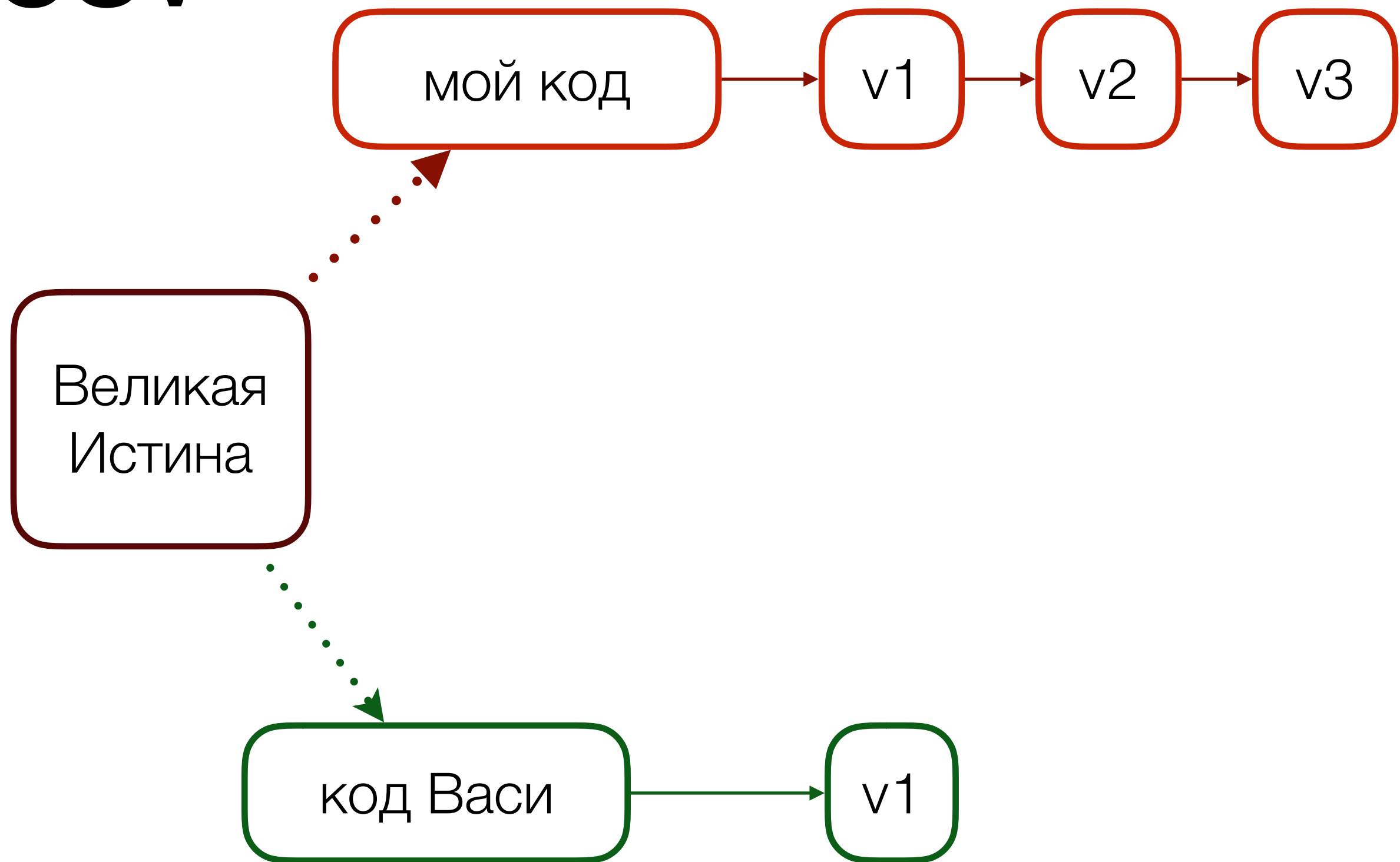
Версионирование



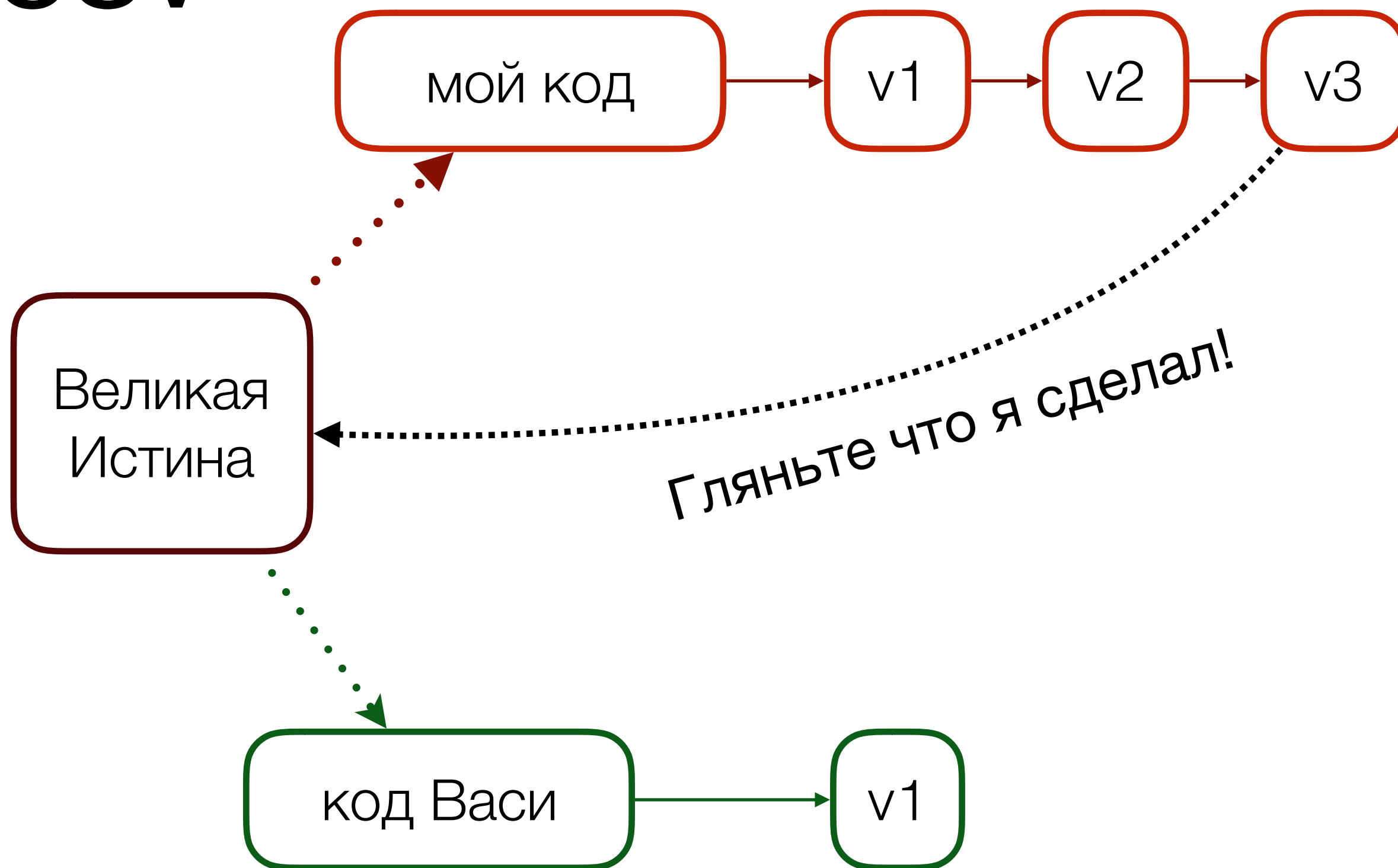
Версионирование



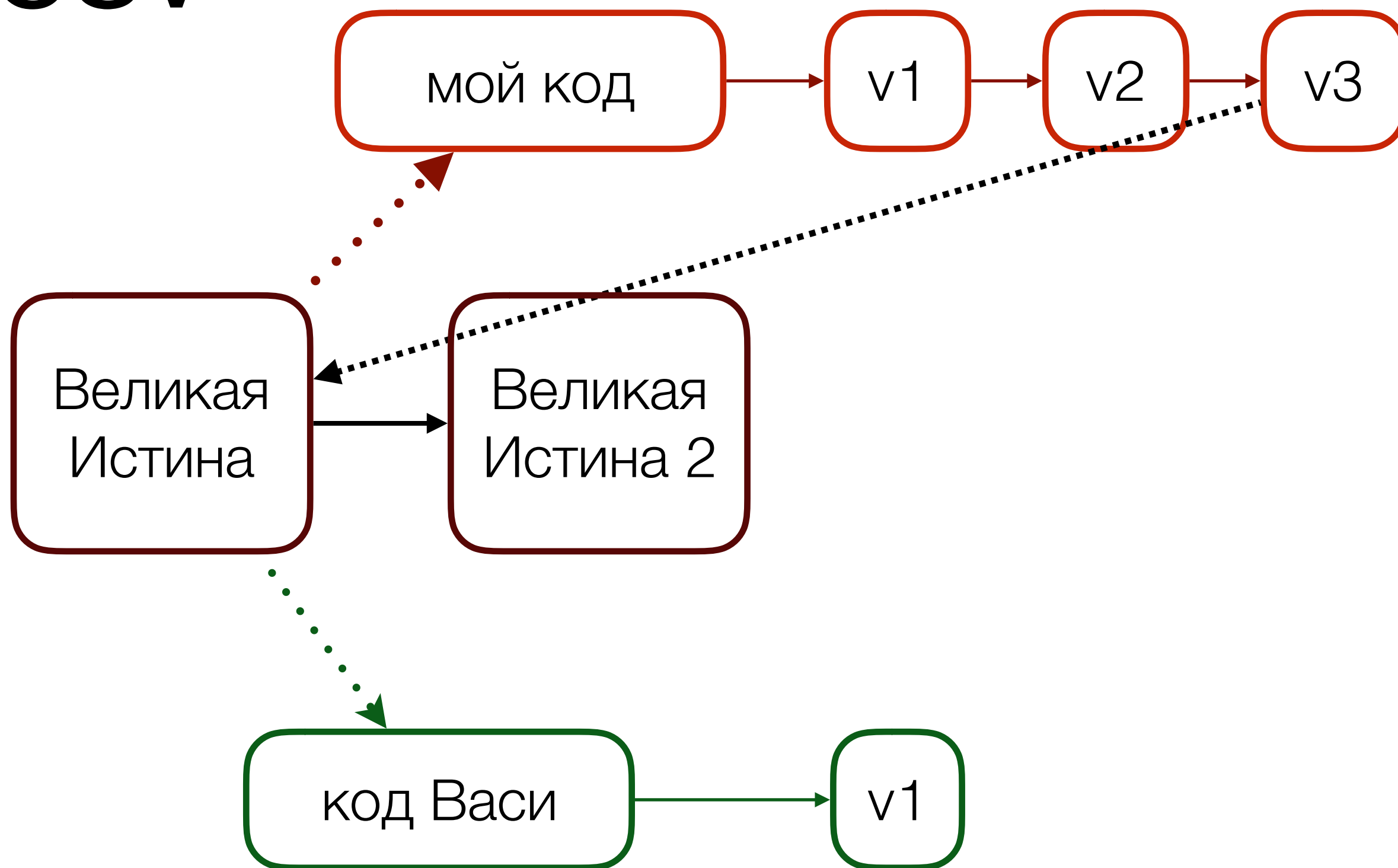
CSV



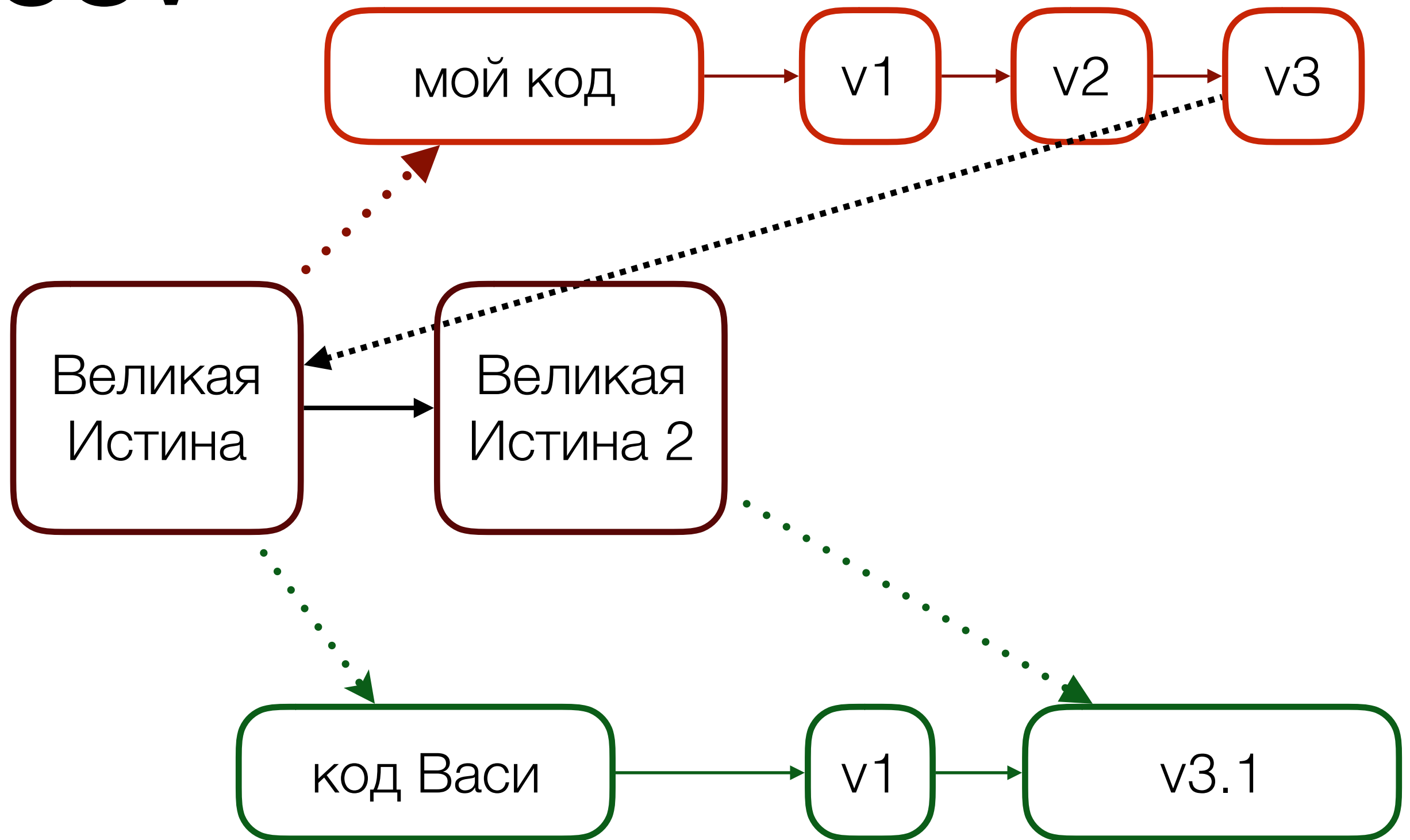
CSV



CSV



CSV



CSV Git



CSV Git

Инициализация репозитория

```
$ git init
```


CSV Git

Инициализация репозитория

```
$ git init
```

Добавление всех файлов в stage area

```
$ git add .
```

CSV Git

Инициализация репозитория

```
$ git init
```

Добавление всех файлов в stage area

```
$ git add .
```

Фиксирование ревизии (делаем коммит)

```
$ git commit -m "Первый коммит"
```

CSV Git

Инициализация репозитория

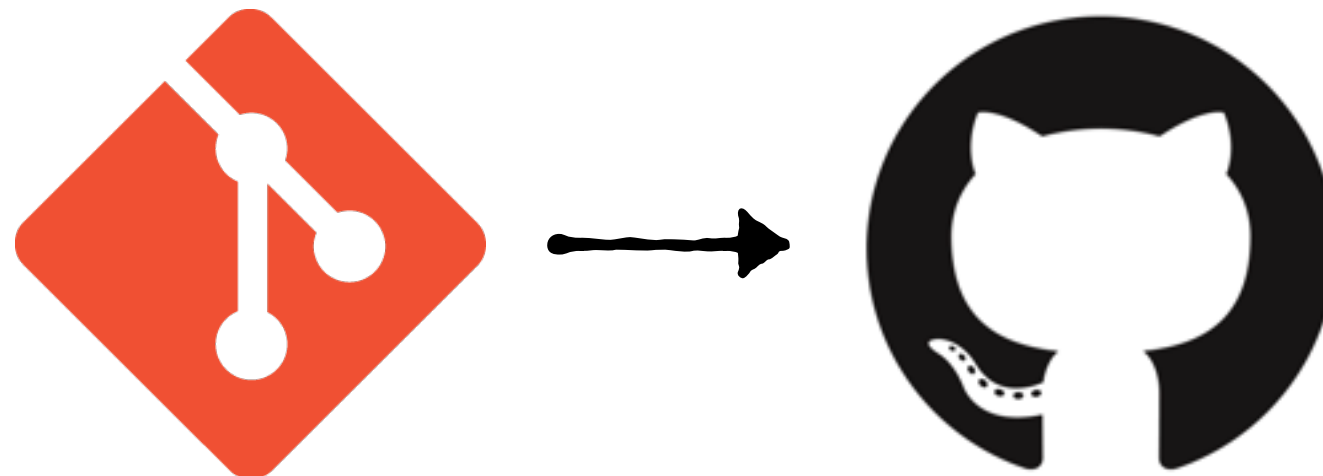
```
$ git init
```

Добавление всех файлов в stage area

```
$ git add .
```

Фиксирование ревизии (делаем коммит)

```
$ git commit -m "Первый коммит"
```



CSV Git

Добавление remote репозитория

```
$ git remote set-url origin <url>
```

CSV Git

Добавление remote репозитория

```
$ git remote set-url origin <url>
```

Push

```
$ git push origin master
```

CSV Git

Добавление remote репозитория

```
$ git remote set-url origin <url>
```

Push

```
$ git push origin master
```

Pull

```
$ git pull origin master
```



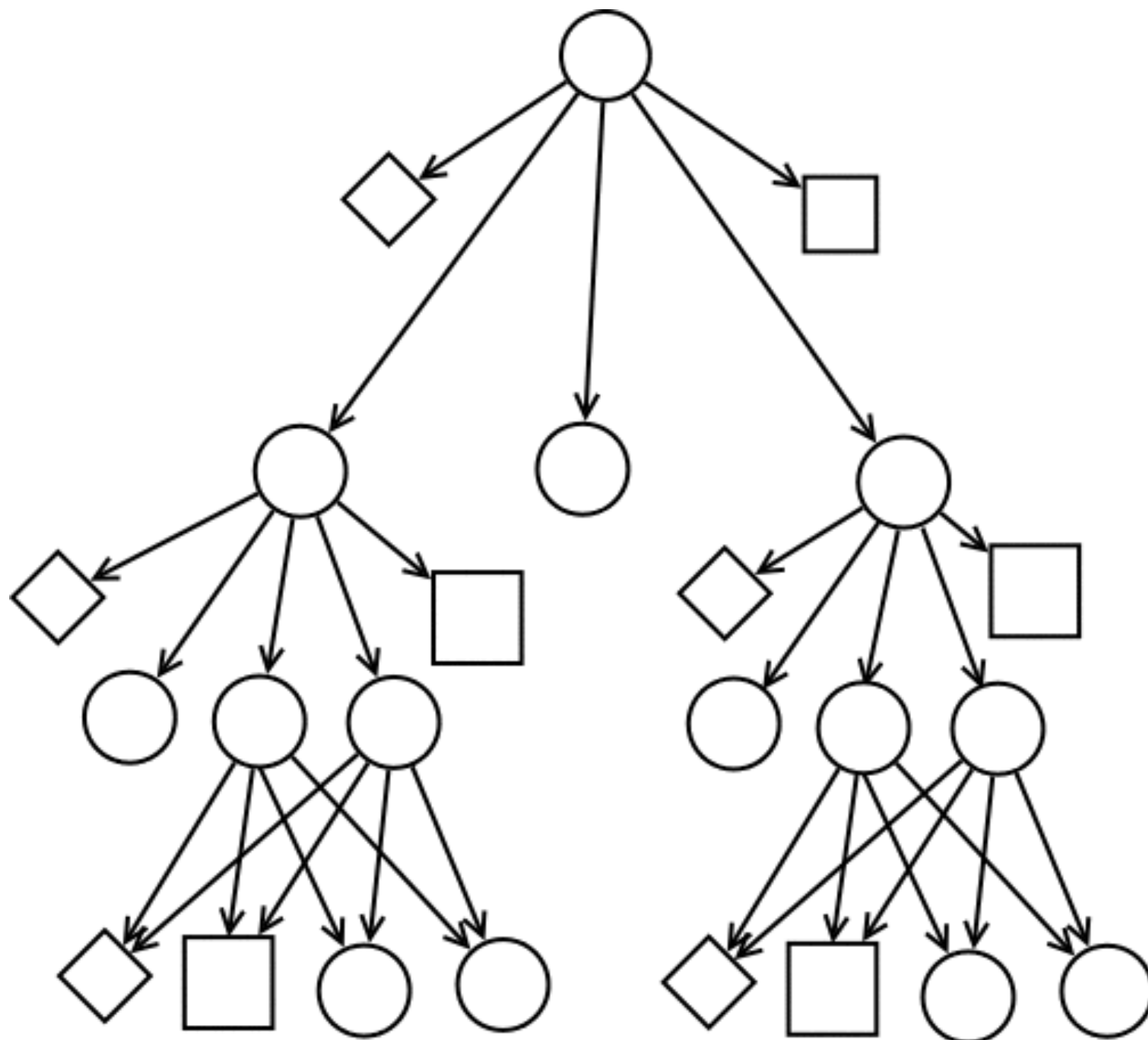
pip

pip

Установка пакета

```
$ pip install numpy
```


Зависимости



ЗАВИСИМОСТИ

Списывание зависимостей в requirements.txt

```
django==1.8  
requests==2.9.1
```

ЗАВИСИМОСТИ

Списывание зависимостей в requirements.txt

```
django==1.9.2  
requests==2.9.1
```

Установка зависимостей через pip

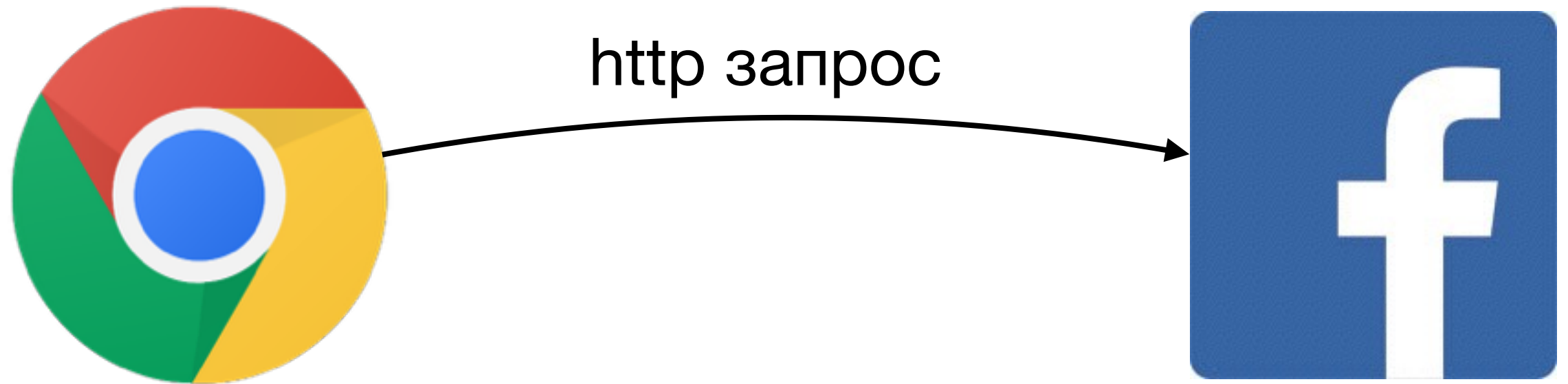
```
$ pip install -r requirements.txt
```

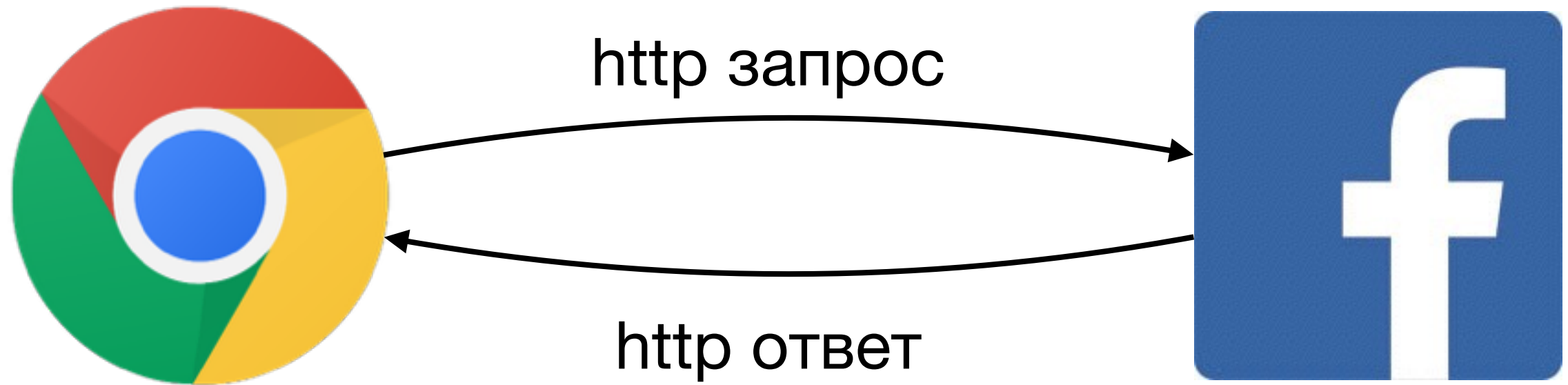
Web и Python

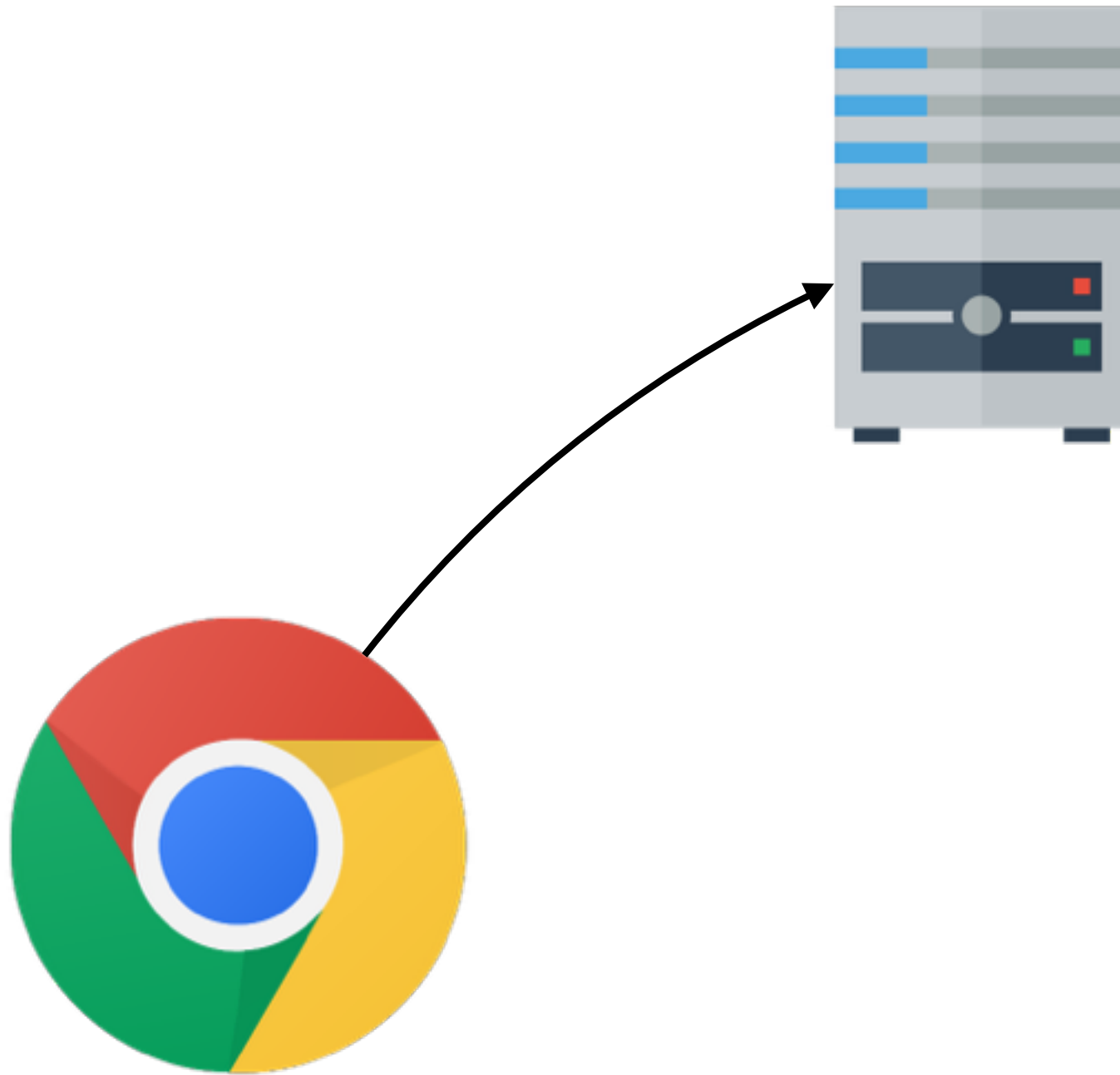


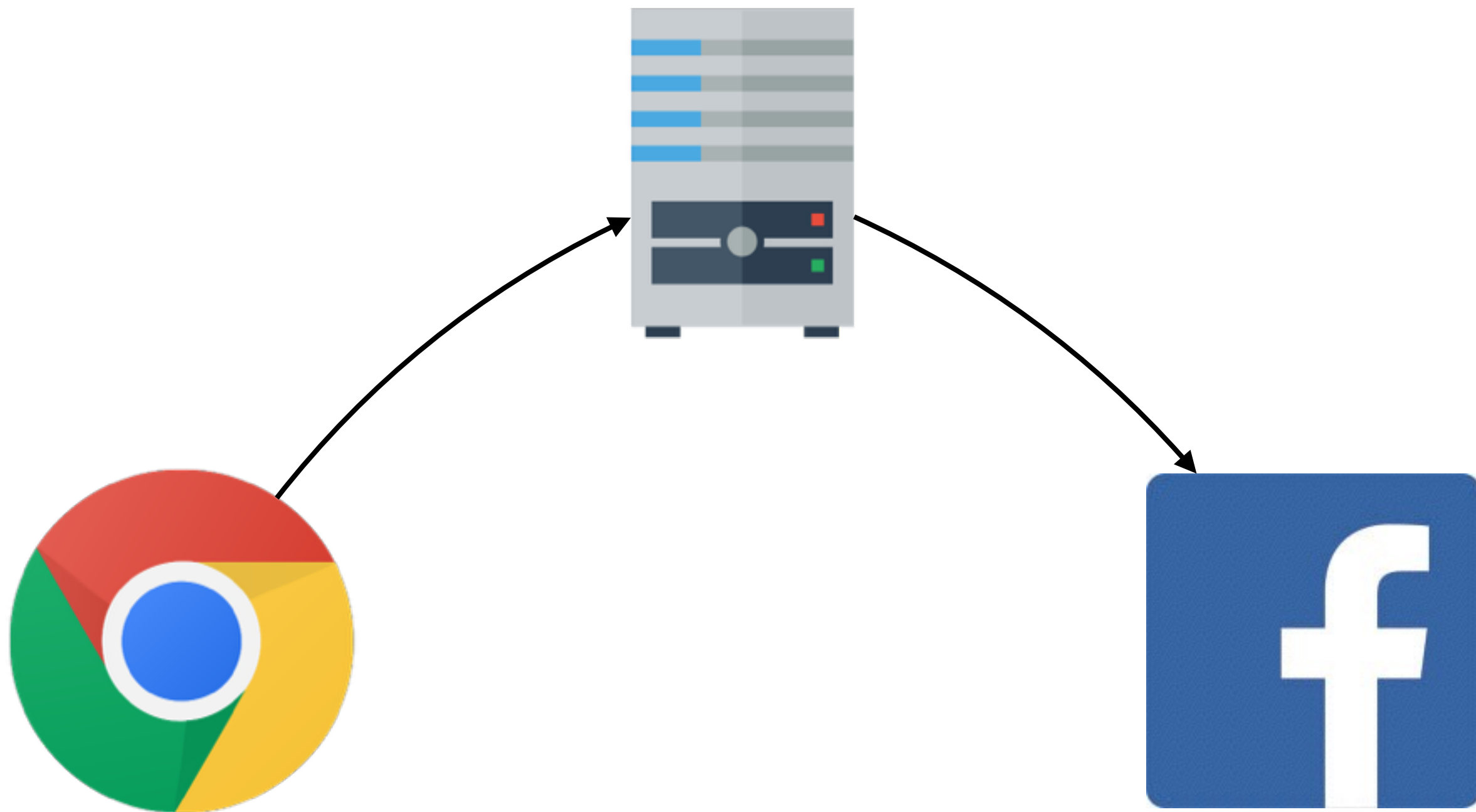
Как работает интернет

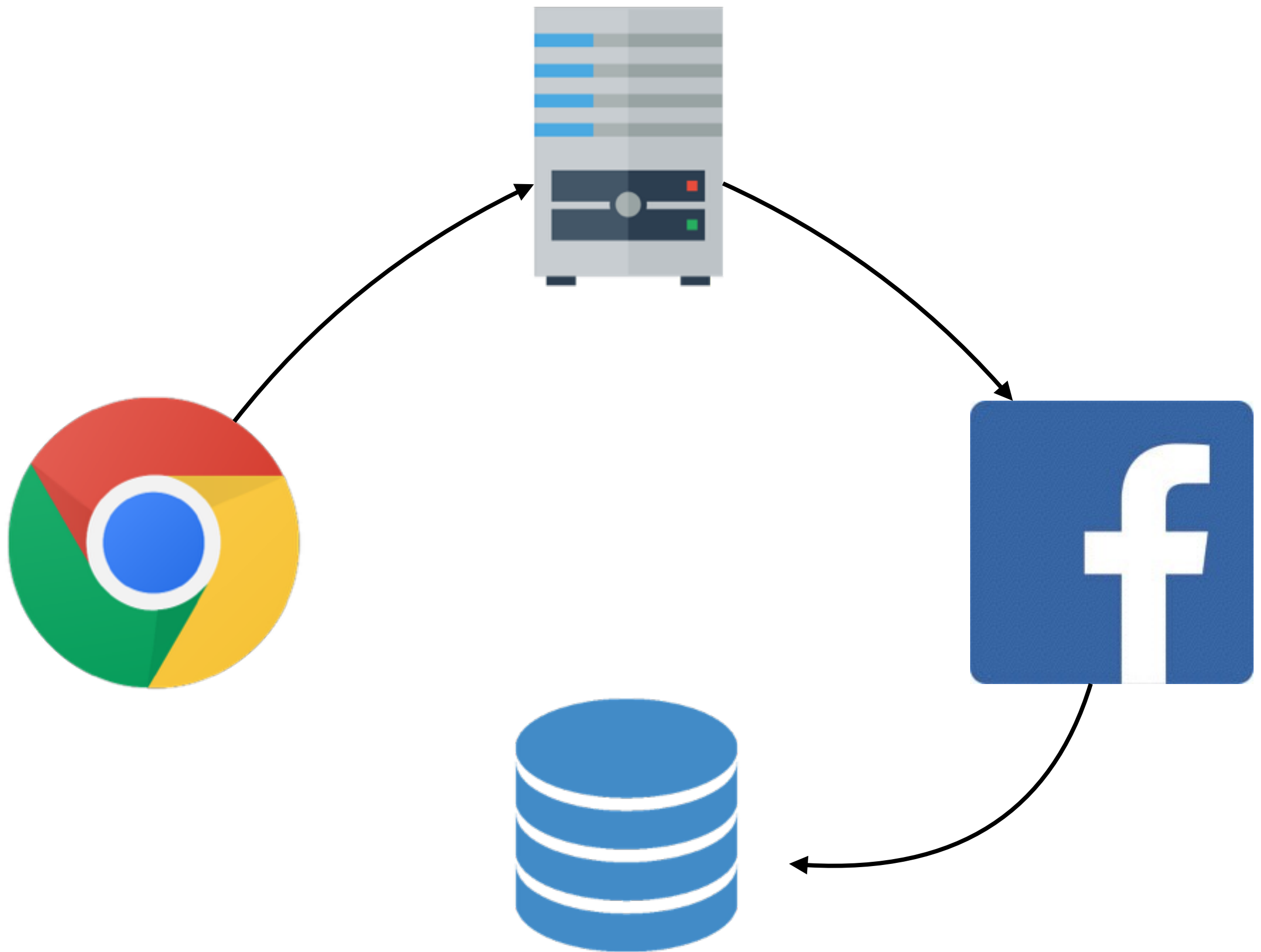


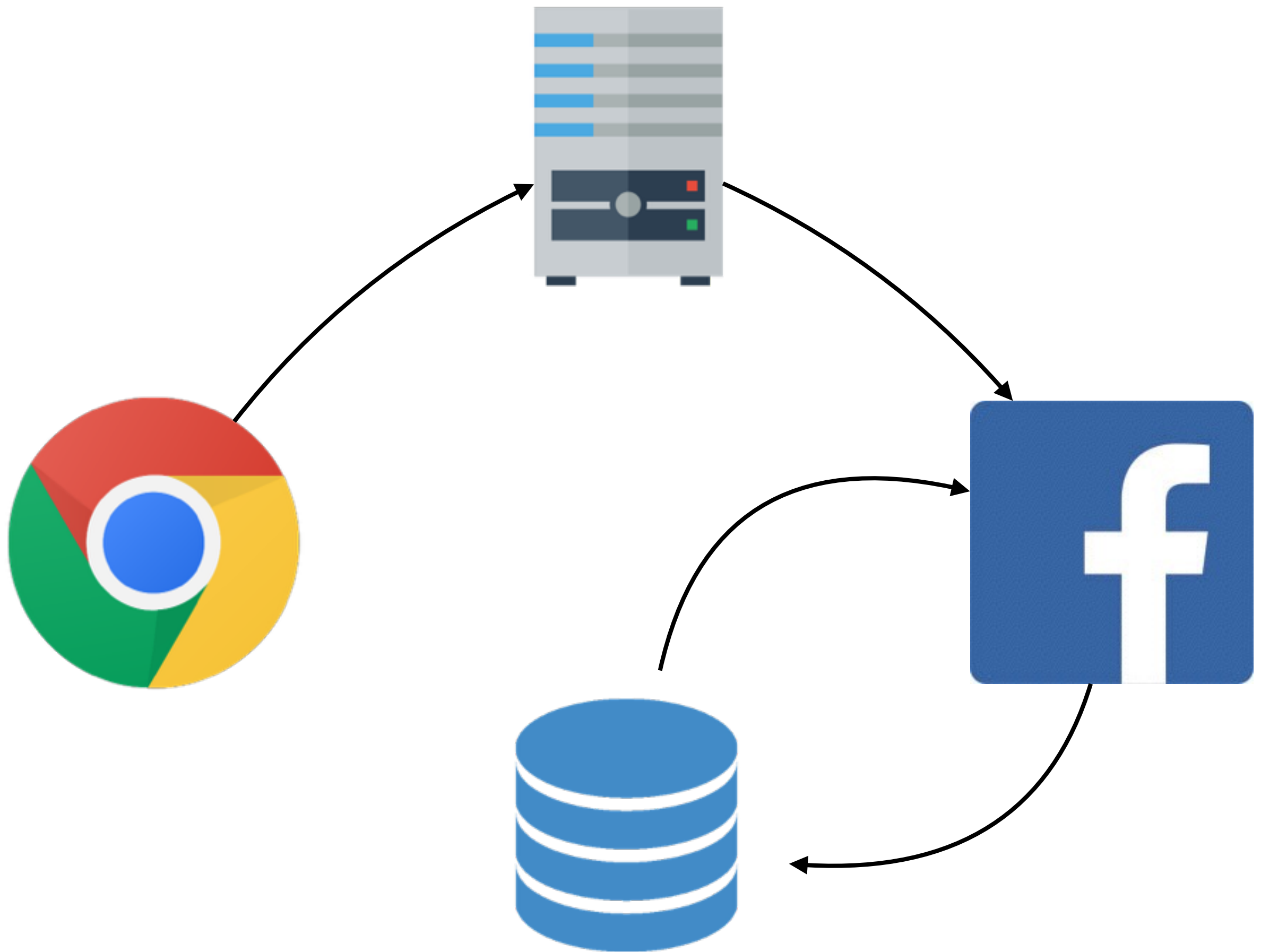












Веб-фреймворки

Веб-фреймворки

- Django
- Flask

Django

<https://www.djangoproject.com/>

Django

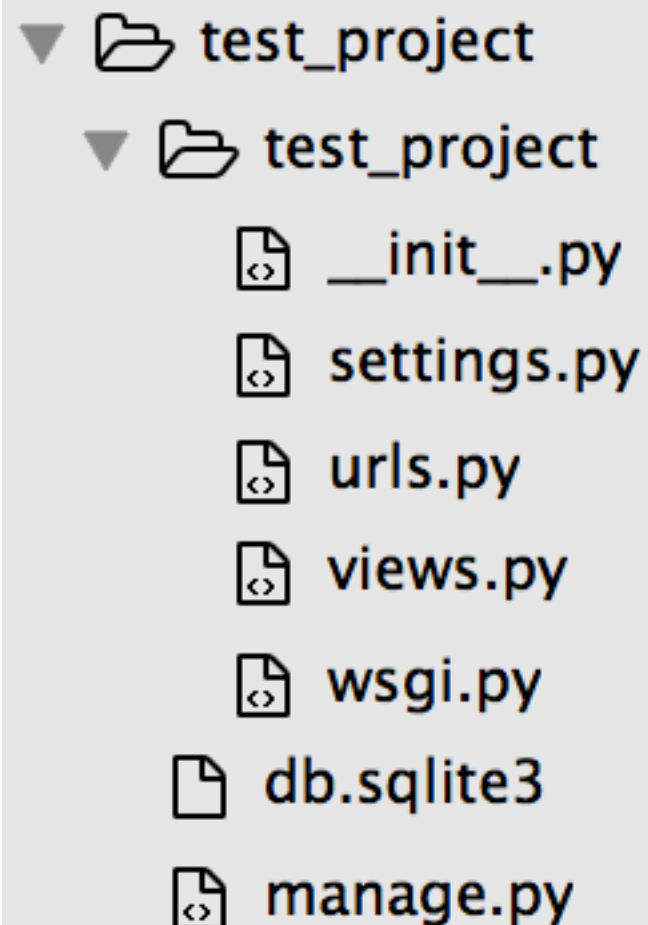
<https://www.djangoproject.com/>

```
$ django-admin startproject crm
```


Django

<https://www.djangoproject.com/>

```
$ django-admin startproject crm
```



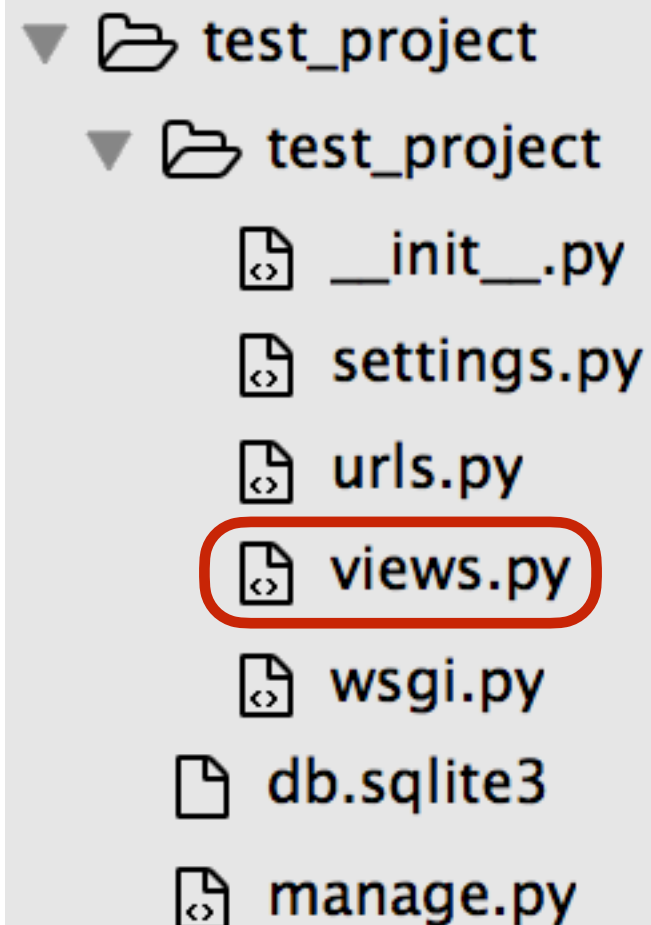
A file explorer view showing the structure of a Django project. The root directory is 'test_project', which is expanded to show its contents. Inside 'test_project', there is a subdirectory 'test_project' (also expanded) containing the following files: '__init__.py', 'settings.py', 'urls.py', 'views.py', and 'wsgi.py'. Additionally, there are two files at the root level: 'db.sqlite3' and 'manage.py'.

- ▼ test_project
 - ▼ test_project
 - __init__.py
 - settings.py
 - urls.py
 - views.py
 - wsgi.py
 - db.sqlite3
 - manage.py

Django

<https://www.djangoproject.com/>

```
$ django-admin startproject crm
```



A file explorer window showing the structure of a Django project named 'test_project'. The project folder is expanded, revealing a sub-folder 'test_project' which contains several Python files. The file 'views.py' is highlighted with a red circle.

- ▼ test_project
 - ▼ test_project
 - __init__.py
 - settings.py
 - urls.py
 - views.py
 - wsgi.py
 - db.sqlite3
 - manage.py

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home')  
03: ]
```

views.py

```
01: from django.http import HttpResponse
02:
03: def home(req):
04:     return HttpResponse('Hello')
```

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home')  
03: ]
```

URL - Uniform Resource Locator

`http://moscoding.ru:80/first?data=Today#sigh`

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home')  
03: ]
```

URL - Uniform Resource Locator

<http://moscoding.ru:80/first?data=Today#sigh>

протокол

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home')  
03: ]
```

URL - Uniform Resource Locator

http://moscoding.ru:80/first?data=Today#sigh

ДОМЕН

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home')  
03: ]
```

URL - Uniform Resource Locator

http://moscoding.ru:80/first?data=Today#sigh
port

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home')  
03: ]
```

URL - Uniform Resource Locator

`http://moscoding.ru:80/first?data=Today#sigh`

путь

path

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home')  
03: ]
```

URL - Uniform Resource Locator

http://moscoding.ru:80/first?data=Today#sigh
query string

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home')  
03: ]
```

URL - Uniform Resource Locator

<http://moscoding.ru:80/first?data=Today#sigh>

XЭШ

urls.py

```
01: urlpatterns = [  
02:     url(r'^$', '<project_name>.views.home'),  
03:     url(r'^/admin/(\w+)$',  
          '<project_name>.views.admin')  
04: ]
```

views.py

```
01: from django.http import HttpResponseRedirect
02:
03: def home(req):
04:     return HttpResponseRedirect('Hello')
05:
06: def admin(req, admin):
07:     return HttpResponseRedirect('Hello {}'.format(admin))
```

Запуск сервера Django

```
$ python manage.py runserver
```

Запуск сервера Django

```
$ python manage.py runserver
```

```
July 13, 2016 - 13:36:38
```

```
Django version 1.9.7, using settings '<project>.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CONTROL-C.
```



heroku.com

Настройка heroku

devcenter.heroku.com/articles/getting-started-with-python

```
$ heroku login
```

Логинимся в heroku

Настройка heroku

devcenter.heroku.com/articles/getting-started-with-python

```
$ heroku login  
$ heroku create
```

Создаем приложение из нашего репозитория

runtime.txt

Настройка python для heroku

```
python-3.5.1
```

Настройка Procfile

Настройка python для heroku

```
web: gunicorn <имя проекта>.wsgi --log-file -
```



Как работает API Telegram?

Как работает API Telegram?

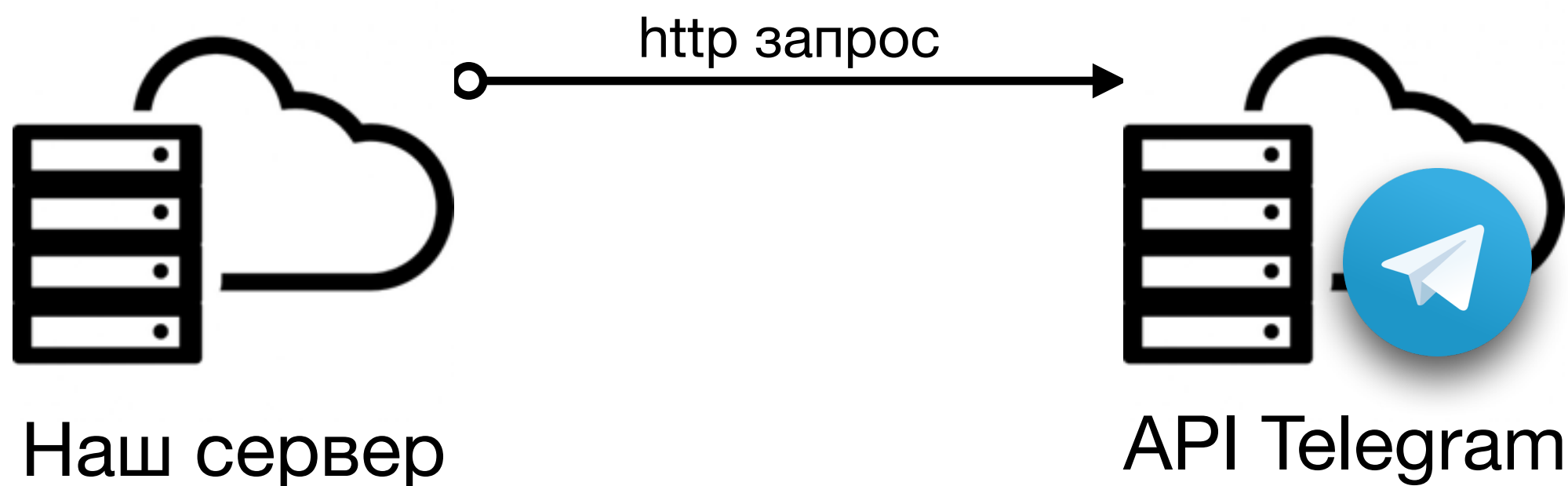


Наш сервер

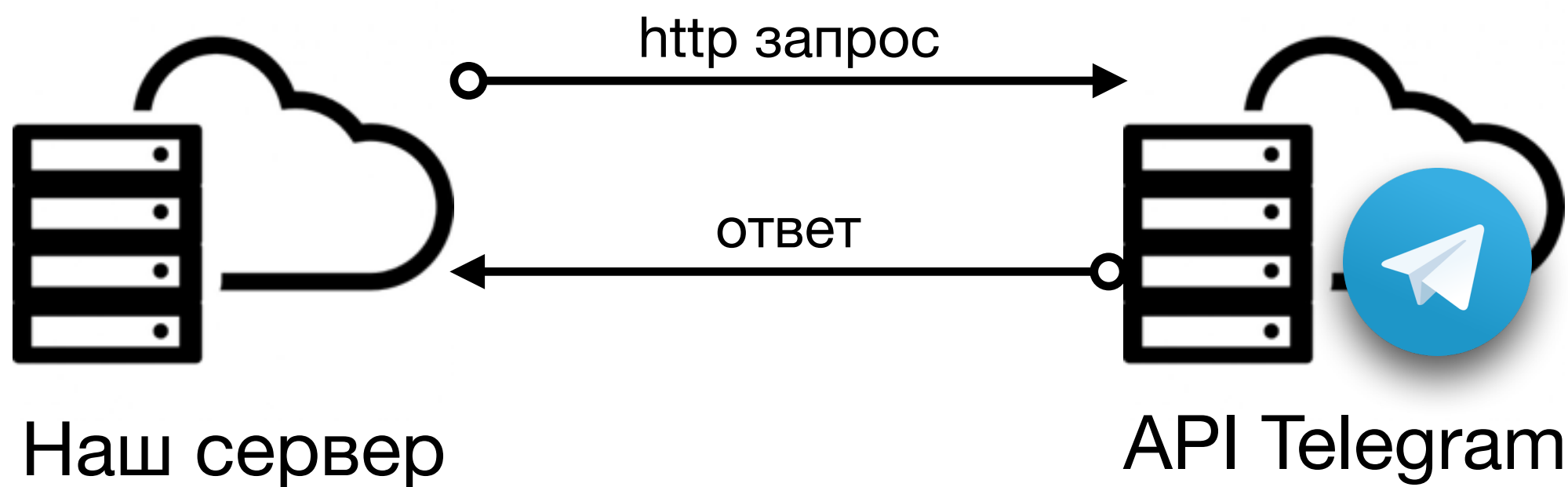


API Telegram

Как работает API Telegram?



Как работает API Telegram?



Как работает API Telegram?



Наш сервер



API Telegram

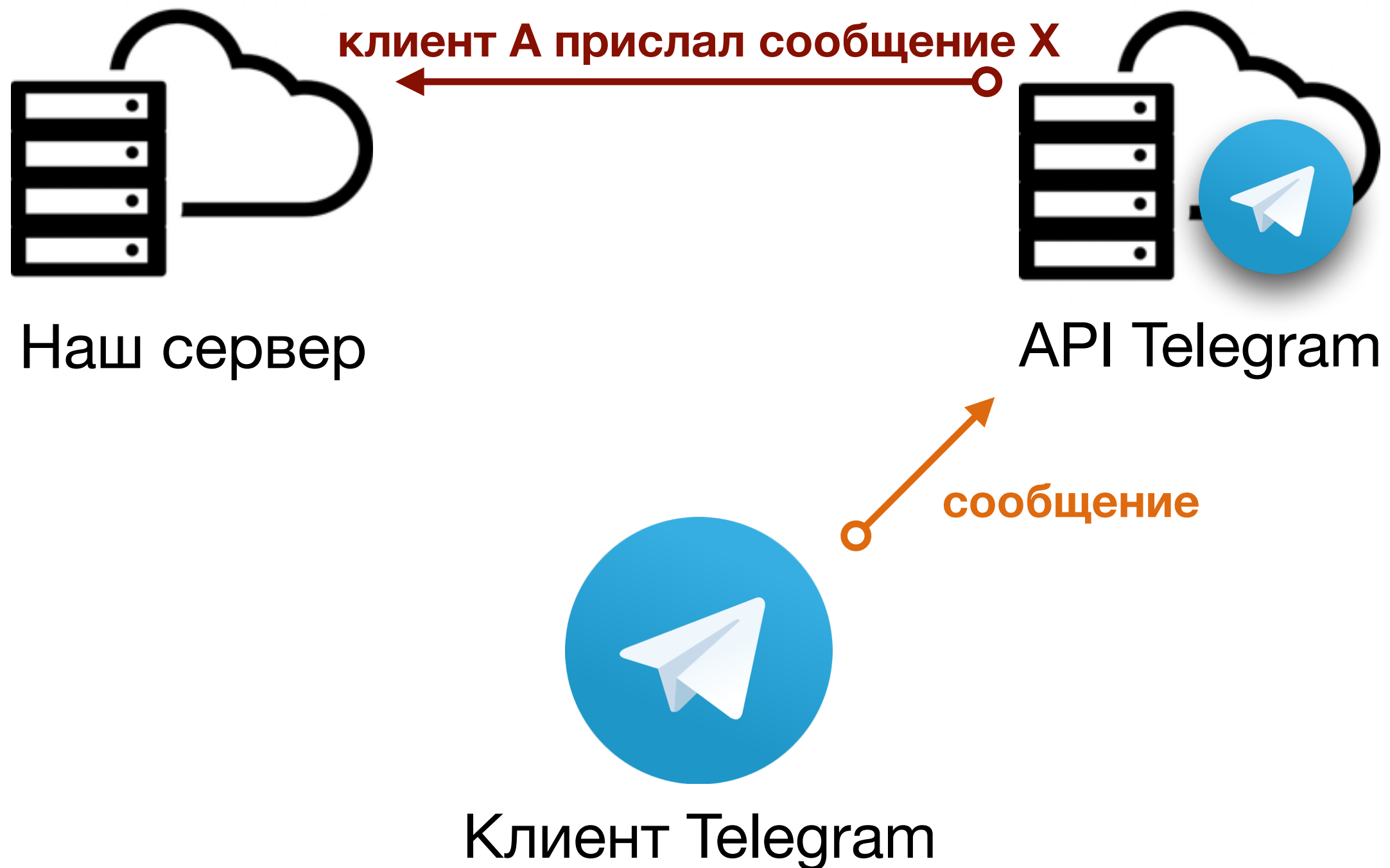


Клиент Telegram

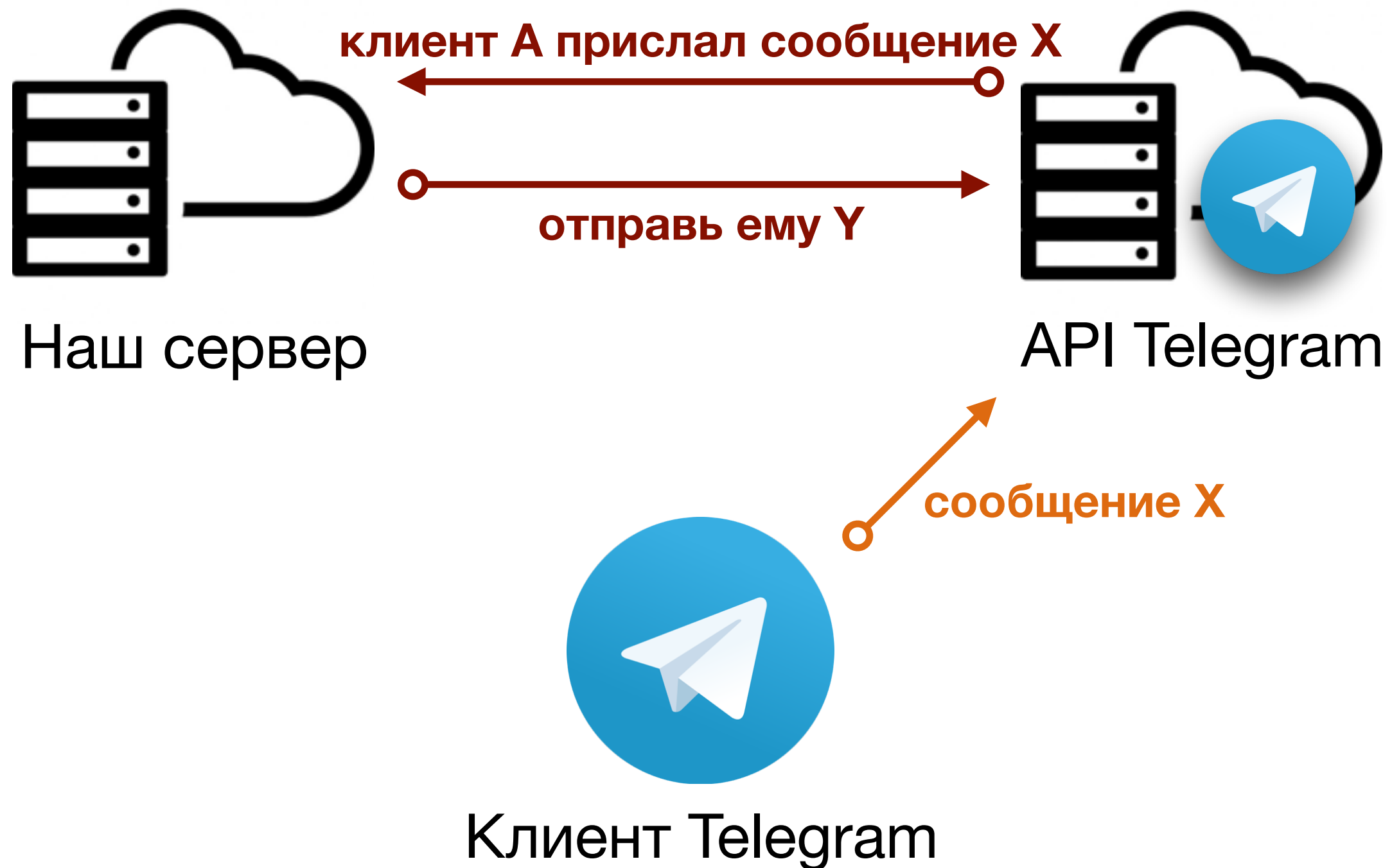


сообщение

Как работает API Telegram?



Как работает API Telegram?



Как работает API Telegram?

