Аутентификация
Авторизация

Как удостовериться в том, кто пришел
и что у него есть права?

Терминология

- Аутентификация – проверка подлинности
- Авторизация – проверка наличия прав

История

- HTTP Basic
- HTTP Digest
- Формы
- Токены
- oAuth, OpenID & etc…

https://habr.com/en/company/dataart/blog/311376/

Stateful vs. Stateless

- State - состояние, возможность помнить клиента между запросами
- Почему это проблема? HTTP по своей природе stateless, т.е. не запоминает “клиента”.
- А так хочется stateful…

Сессии

Клиент и сервер должны обменяться “секретиком” и узнавать друг друга по нему. Какие бывают механизмы обмена?

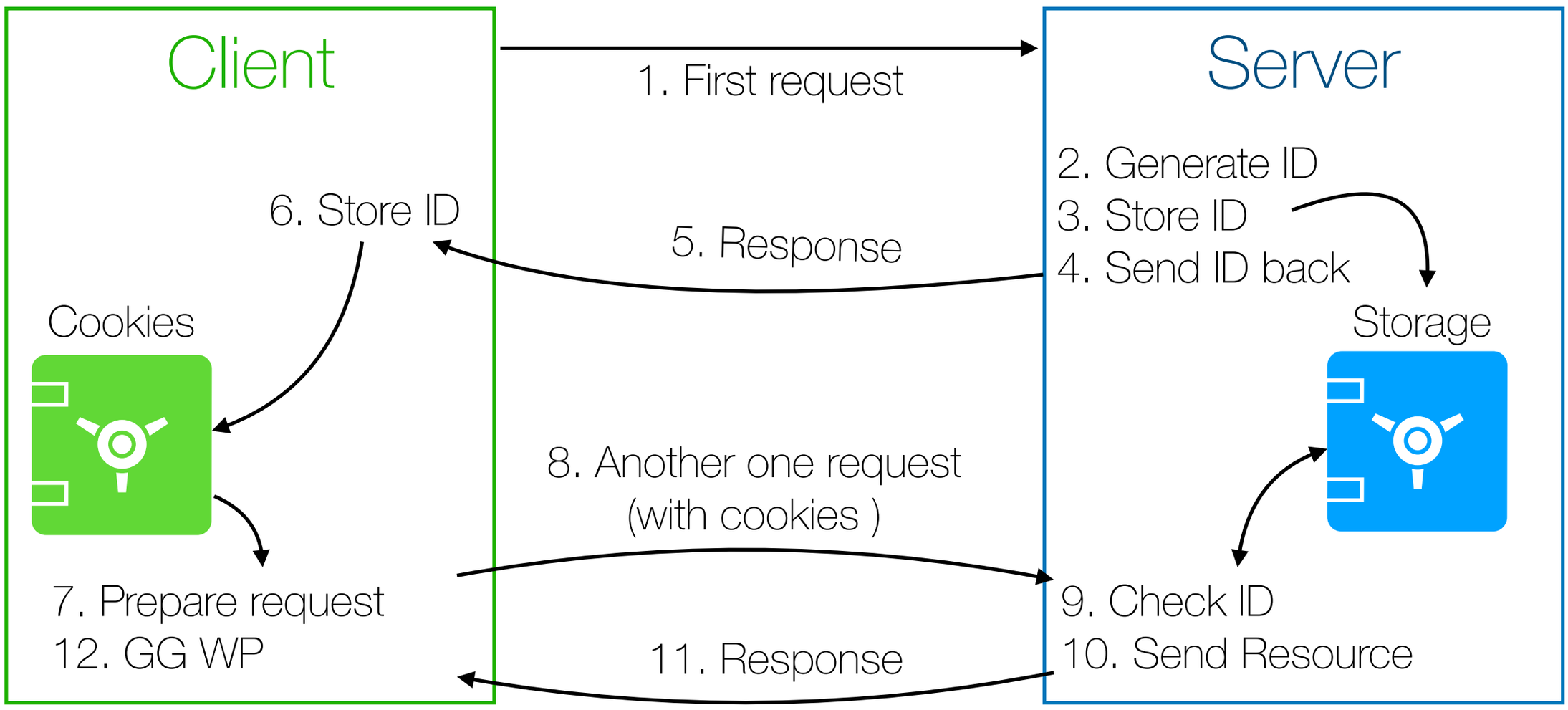
1. Заголовок HTTP-запроса (HTTP Basic, Token, Digest)

2. Тело запроса (зависит от предпочтений… Token, JWT)

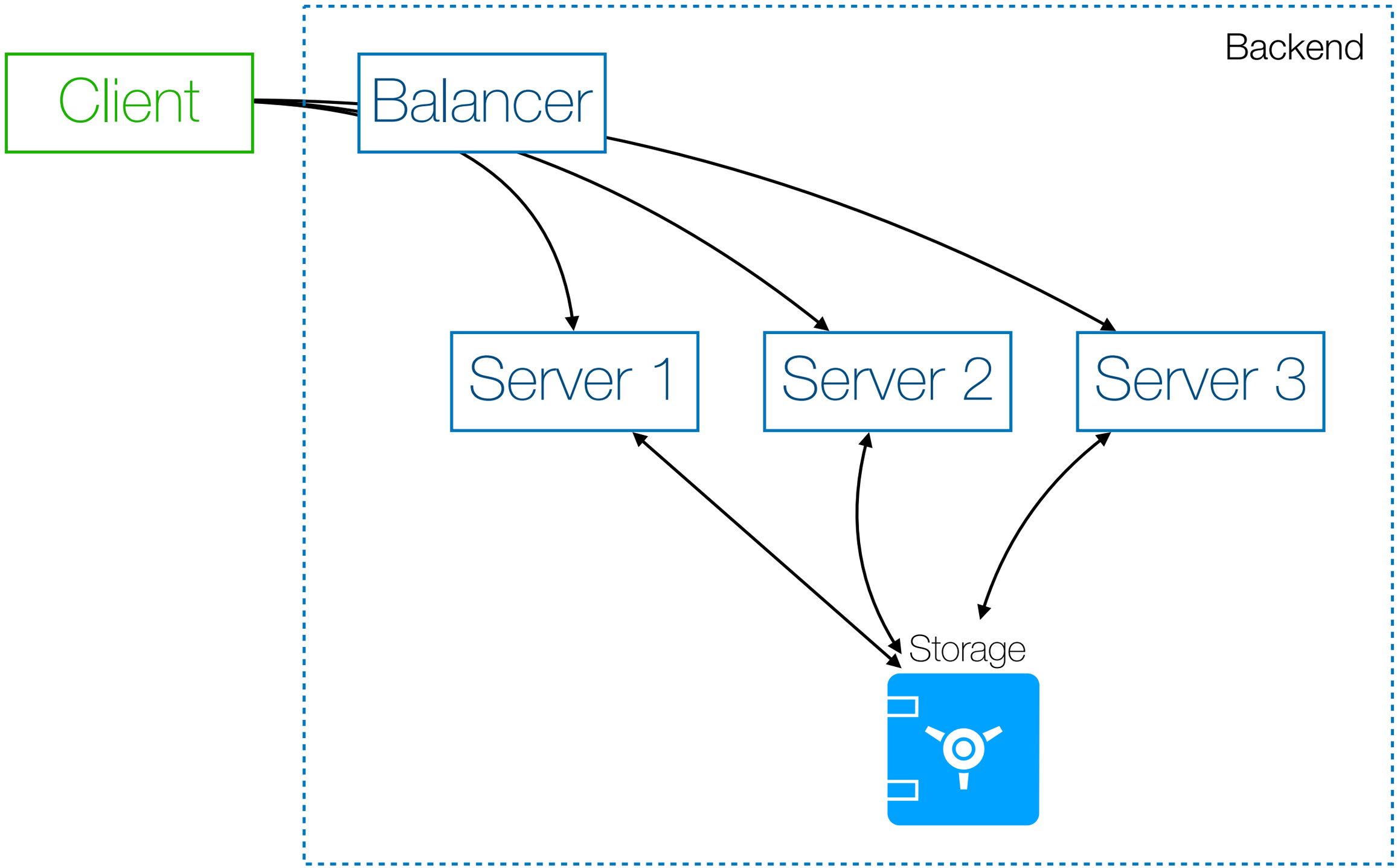
3. Cookies (самообман… это же заголовок =)

Как запомнить клиента?

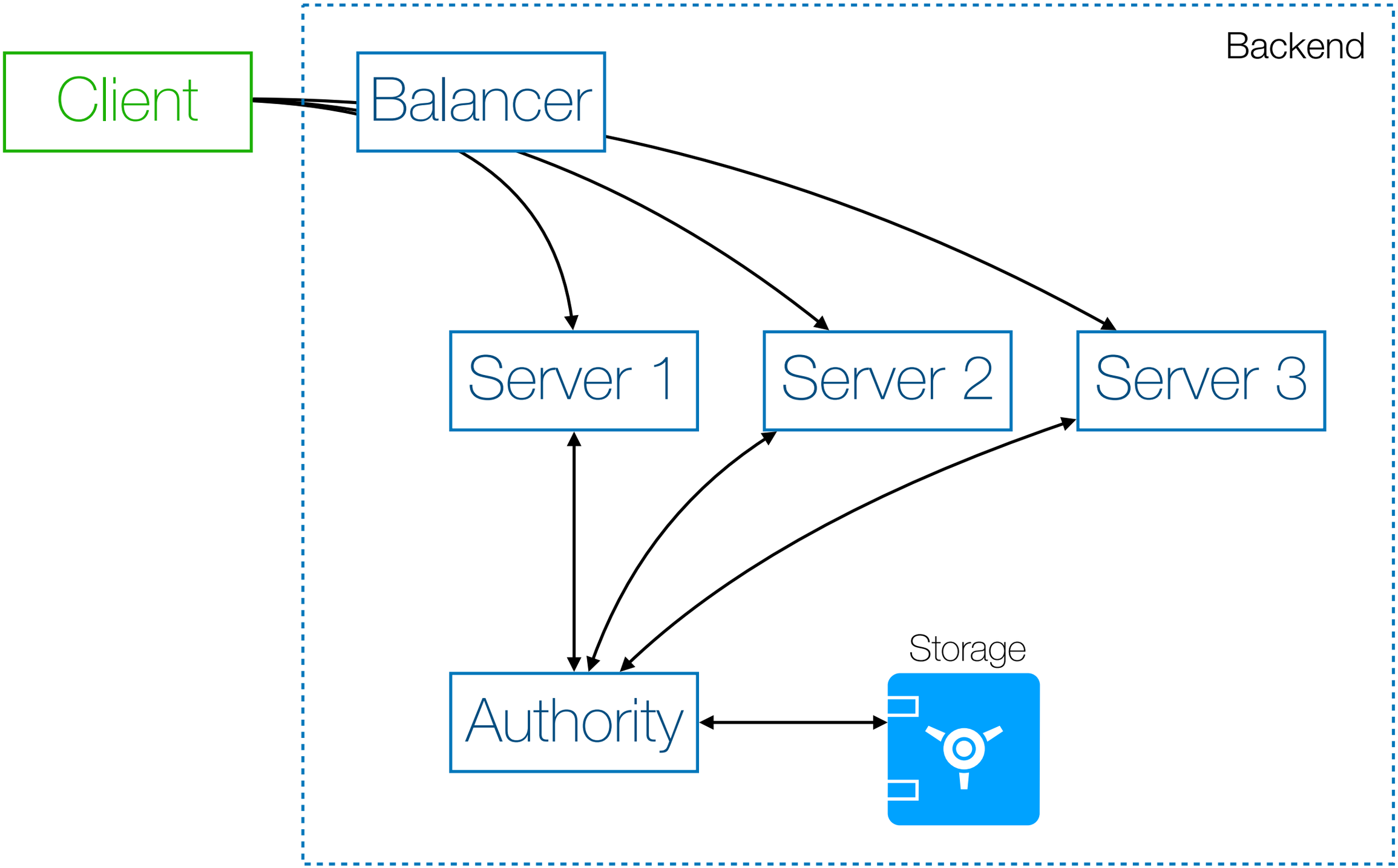
Выдать идентификатор, сохранить его на сервере и заставлять клиента его предъявлять с каждым запросом.



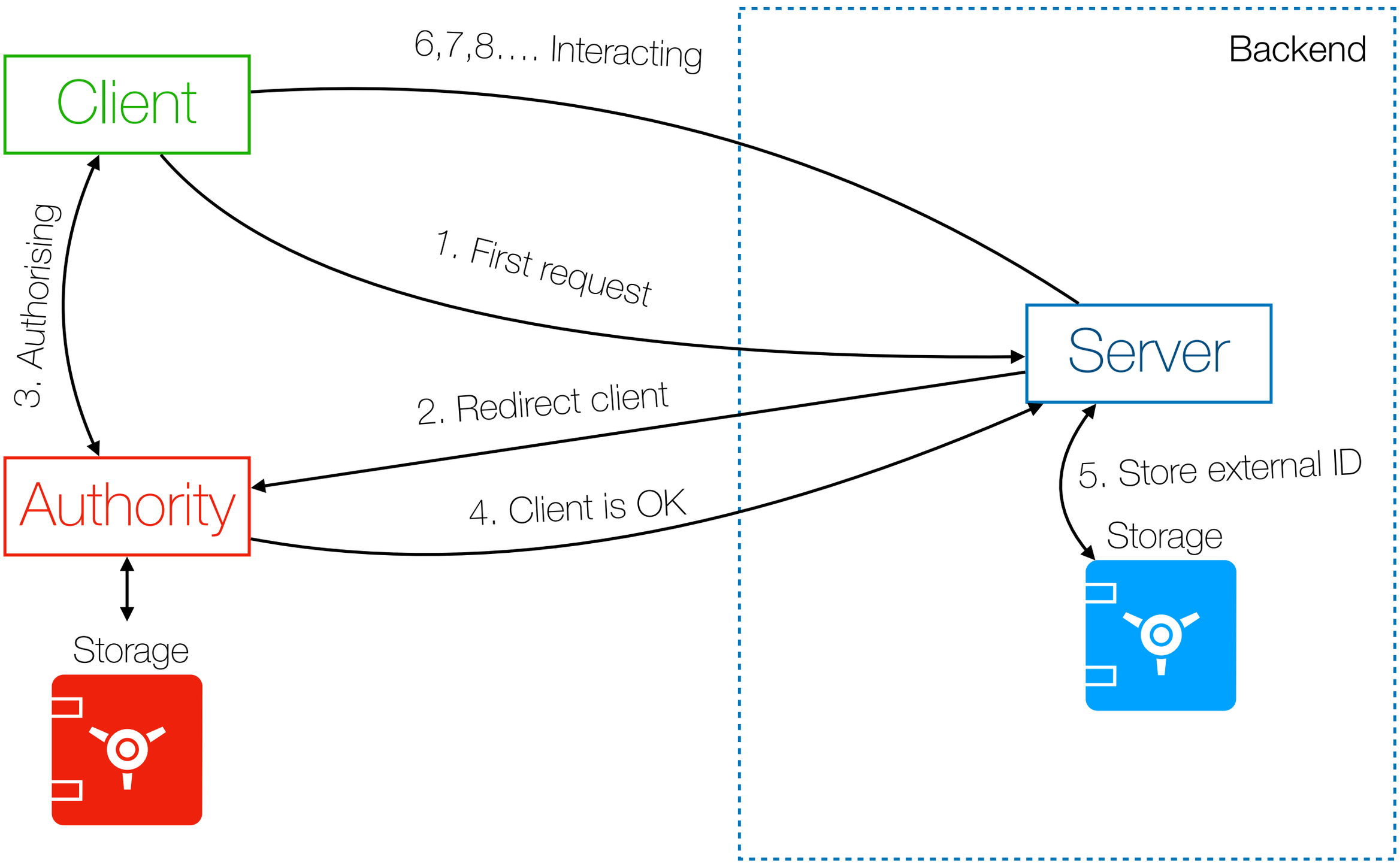
А если серверов много?



Разделение ролей



Проблема ji32k7au4a83



Сложный пароль

ji3 -> 我 -> M

2K7 -> 的 -> Y

au4 -> 密 -> PASS

a83 -> 碼 -> WORD

Ckj;ysqGfhjkm! -> СложныйПароль!

Apps with oAuth 2.0

- Offline mode (PHP, Node, Python & etc…)
  Server-side Apps

- Online mode (React, Angular… + Mobile)
  Client-side Apps

Single Sing-On.

Одна учетная запись для множества приложений.
Один протокол, множество провайдеров.

Этапы

1. Регистрация приложения у провайдера oAuth Google, Twitter, VK, Facebook, Yandex, Mail.ru, Github, something else…
2. Создание в приложении routes
    - public page (кнопка войти через X)
    - protected page (или api?)
    - callback
3. Использование API провайдера