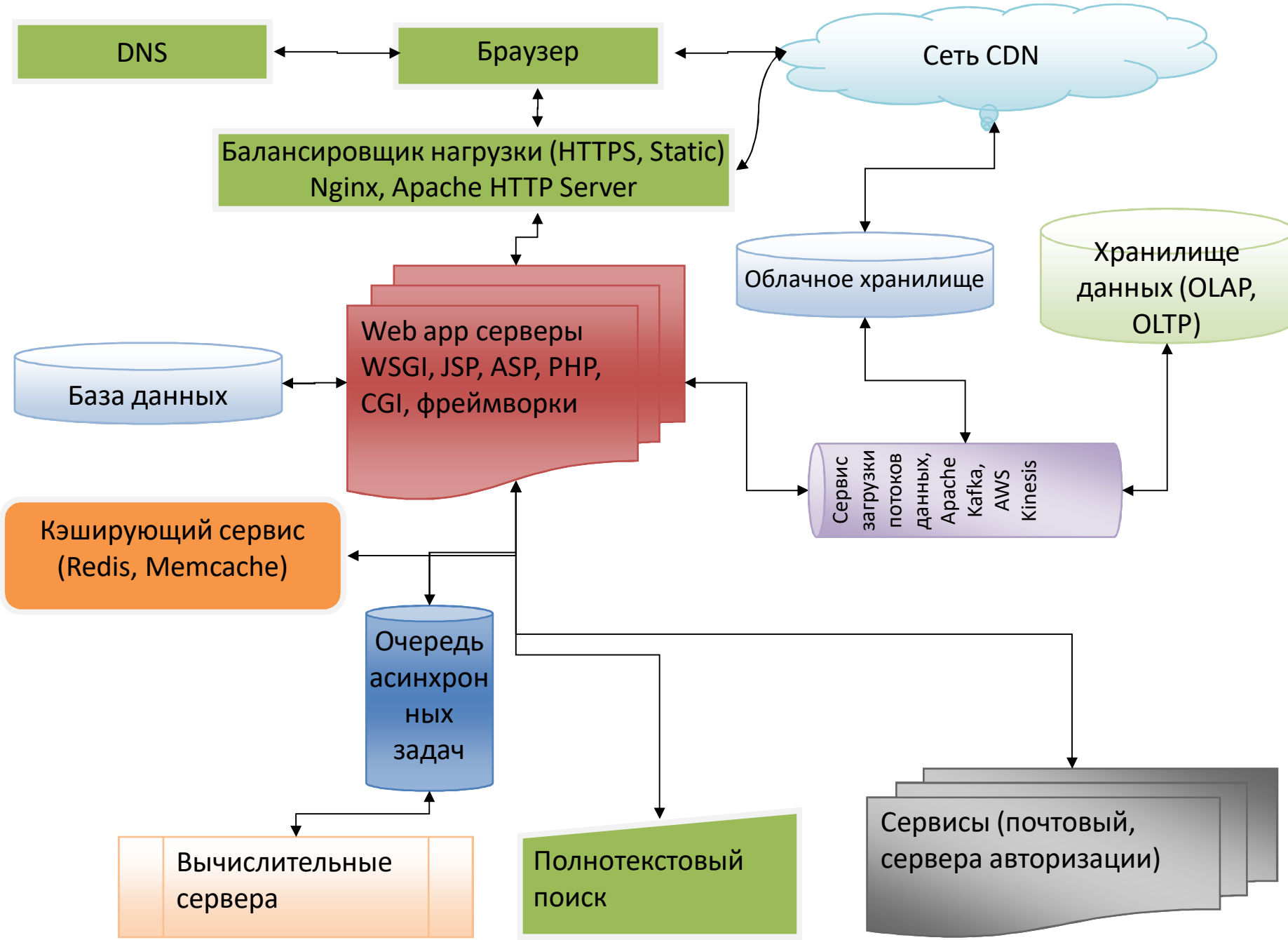
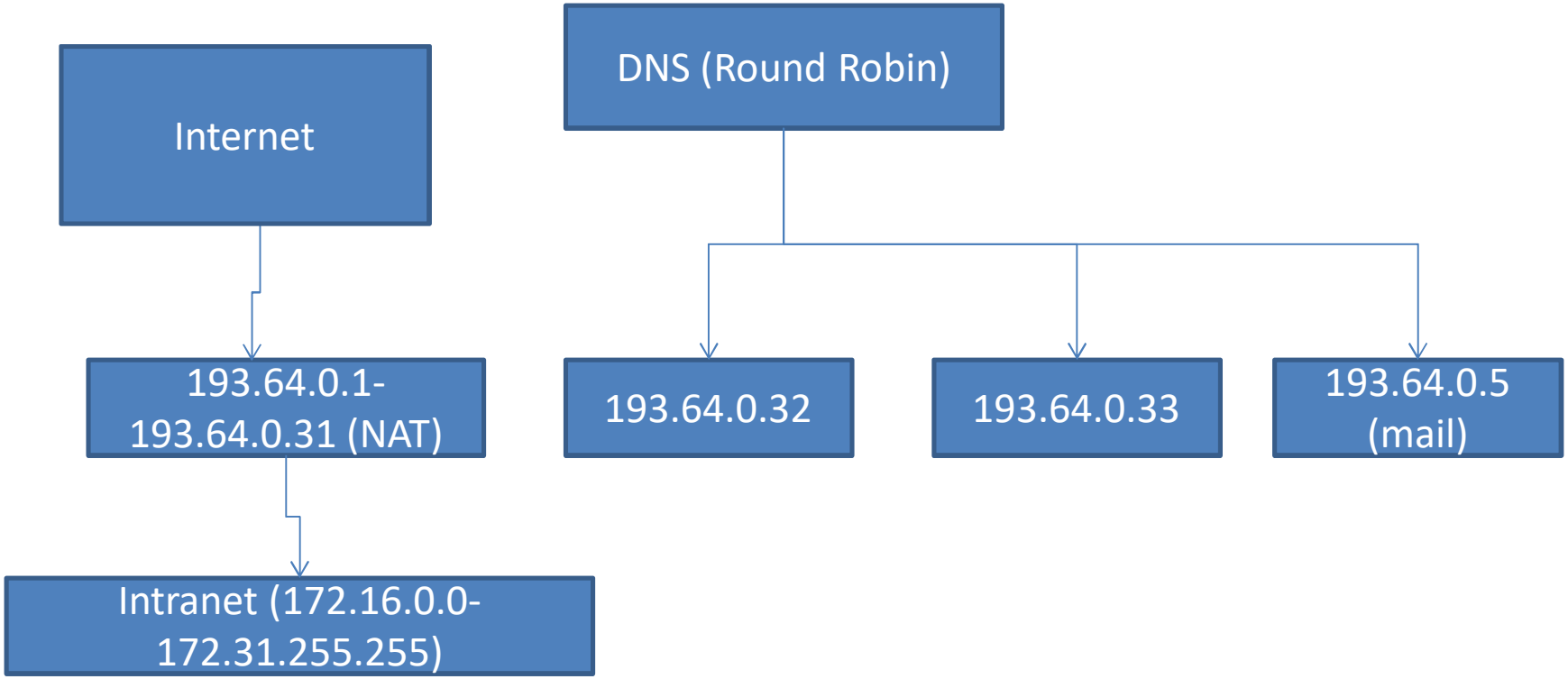
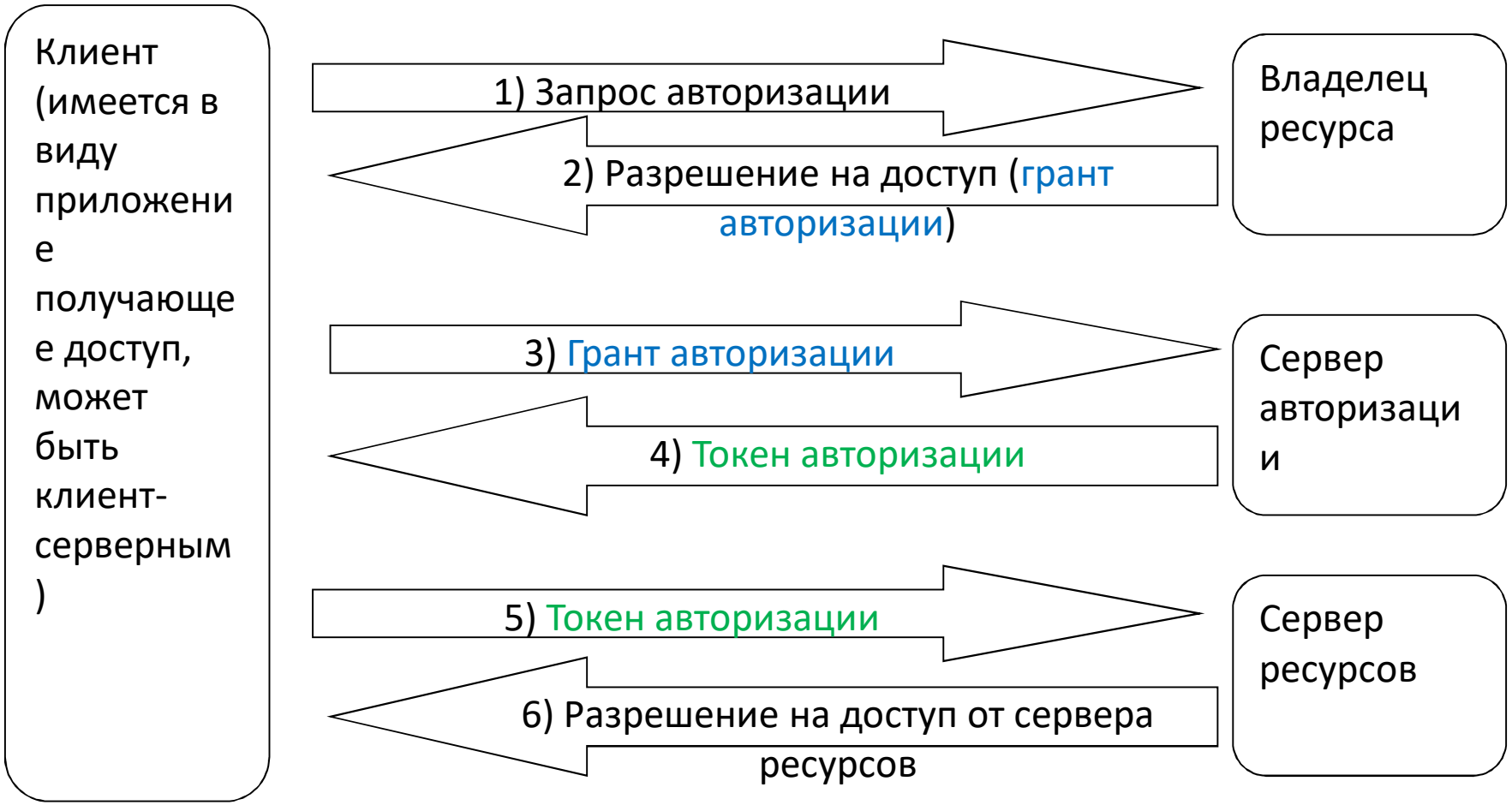
Архитектура веб-сервисов

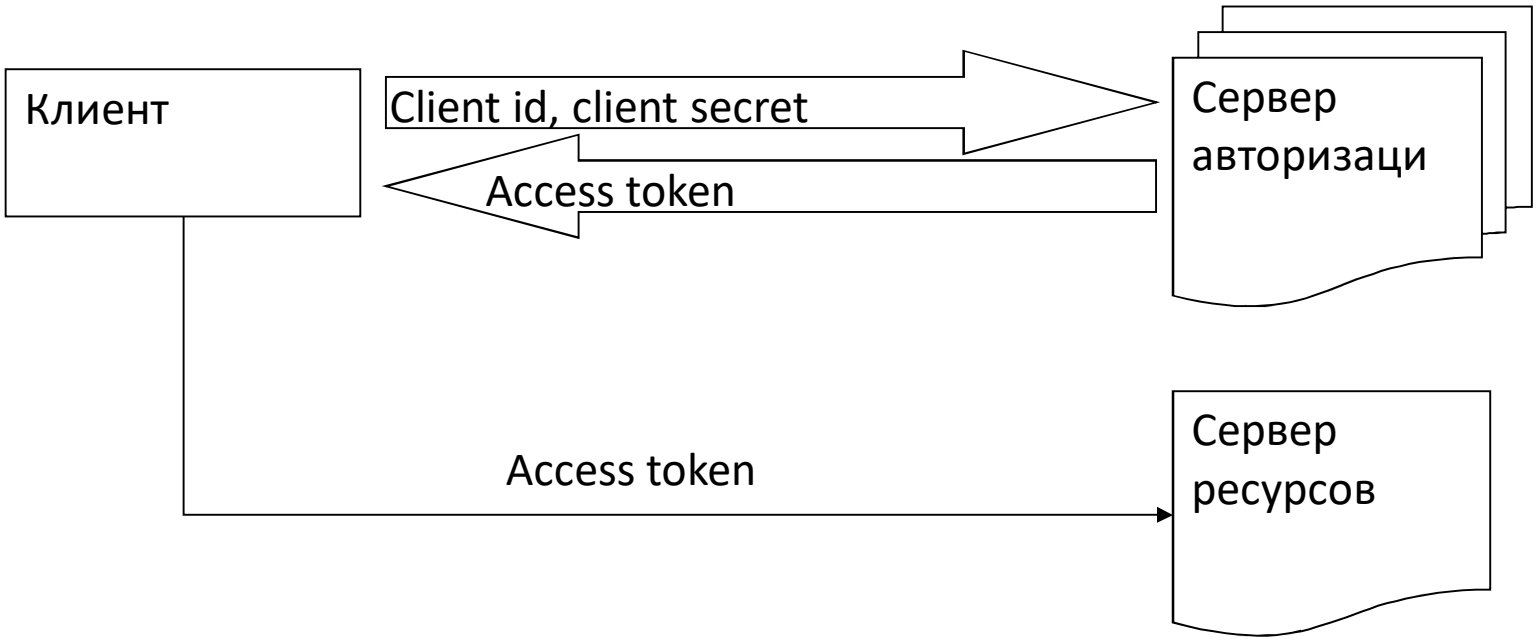

DNS
Браузер
Сеть CDN
Балансировщик нагрузки (HTTPS, Static)
Nginx, Apache HTTP Server
Облачное хранилище
Хранилище данных (OLAP, OLTP)
Web app серверы
WSGI, JSP, ASP, PHP, CGI, фреймворки
База данных
Сервис загрузки потоков данных, Apache Kafka, AWS Kinesis
Кэширующий сервис (Redis, Memcache)
Очередь асинхронных задач
Вычислительные сервера
Полнотекстовый поиск
Сервисы (почтовый, сервера авторизации)


Internet
DNS (Round Robin)
193.64.0.1-
193.64.0.31 (NAT)
193.64.0.32
193.64.0.33
193.64.0.5
(mail)
Intranet (172.16.0.0-
172.31.255.255)

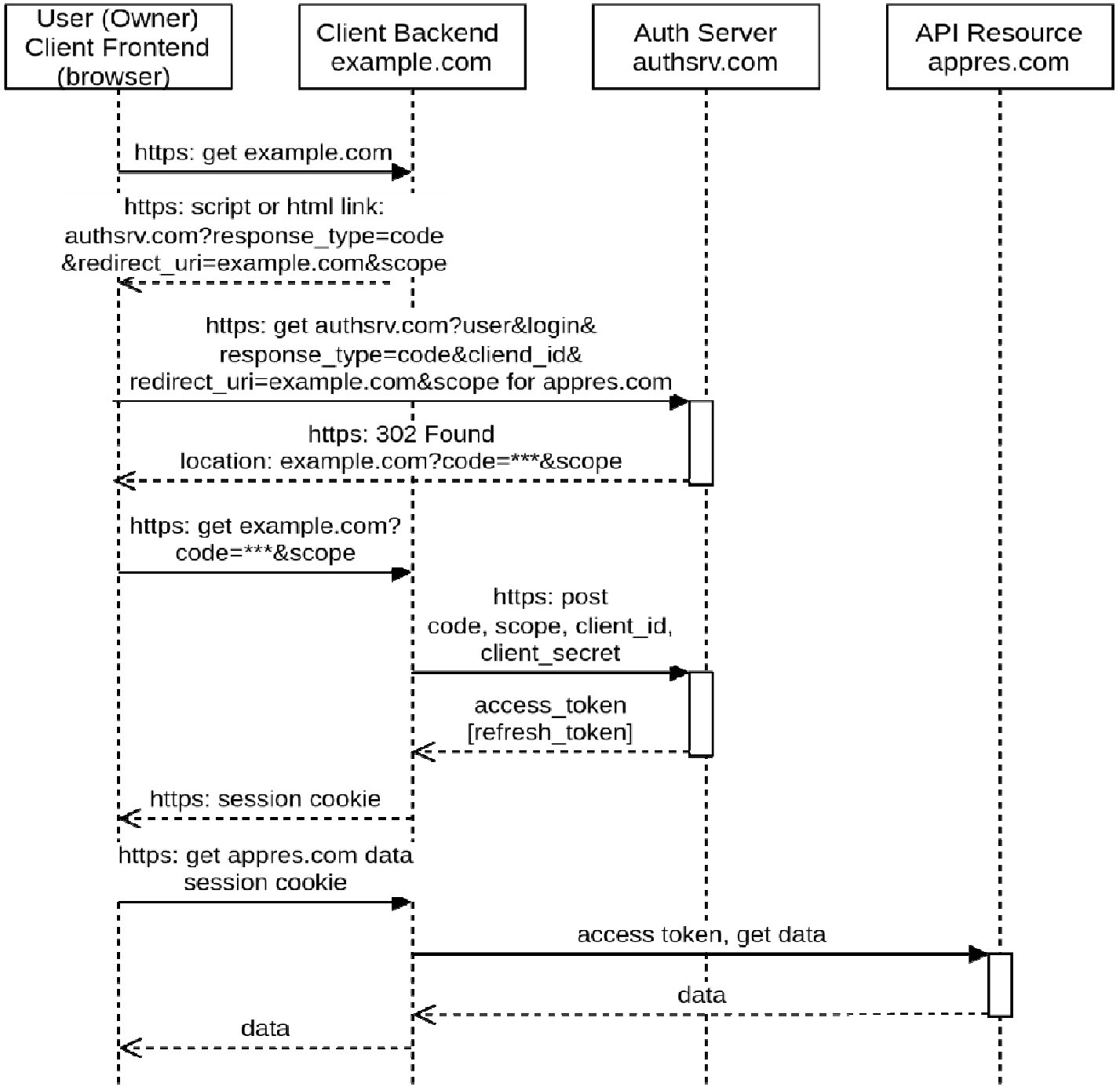
Auth 2.0


Клиент (имеется в виду приложение получающее доступ, может быть клиент-серверным)
1) Запрос авторизации
2) Разрешение на доступ (грант авторизации)
Владелец ресурса
3) Грант авторизации
4) Токен авторизации
Сервер авторизации
5) Токен авторизации
6) Разрешение на доступ от сервера ресурсов
Сервер ресурсов

Client credentials grant flow



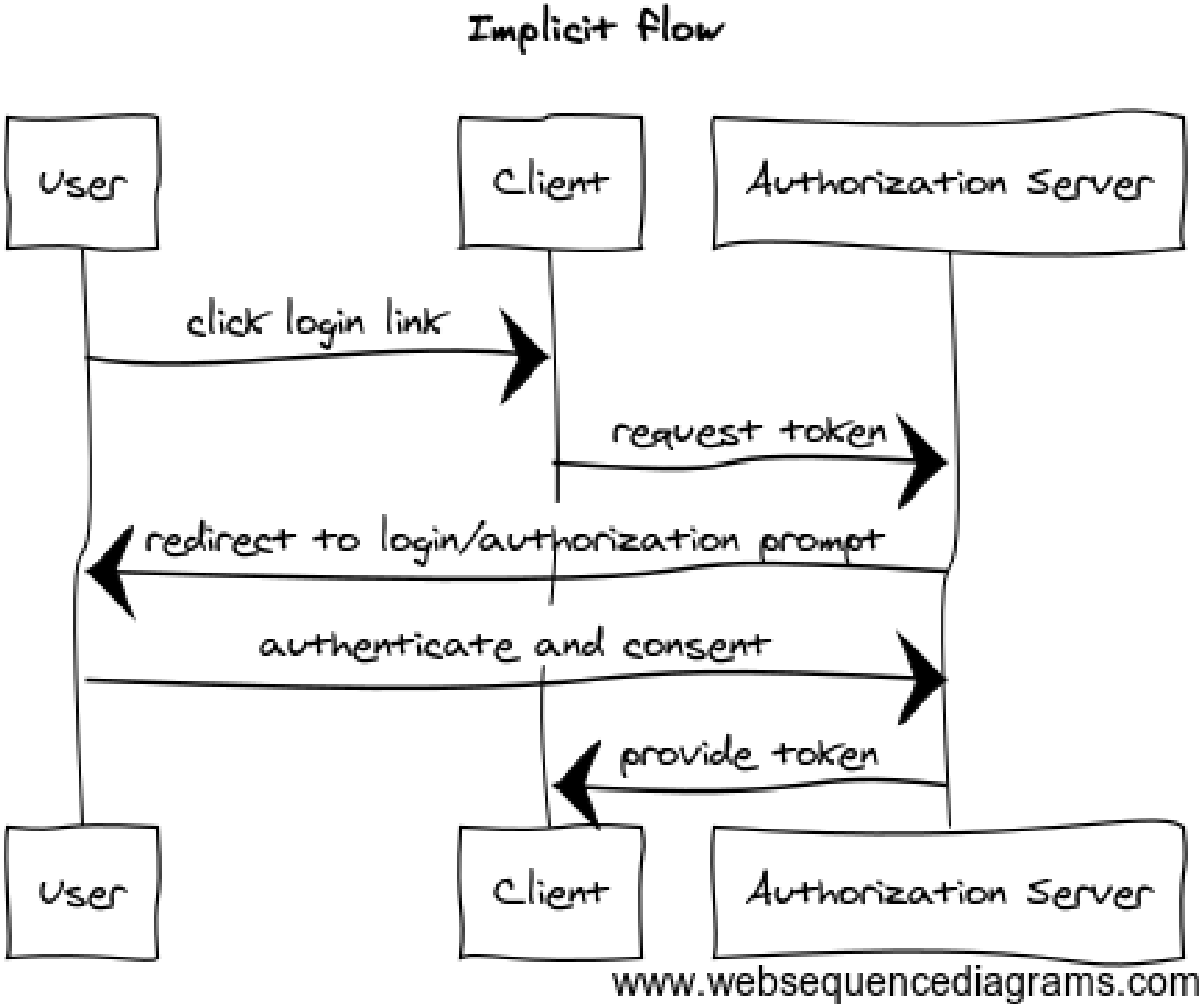
Authorization code flow

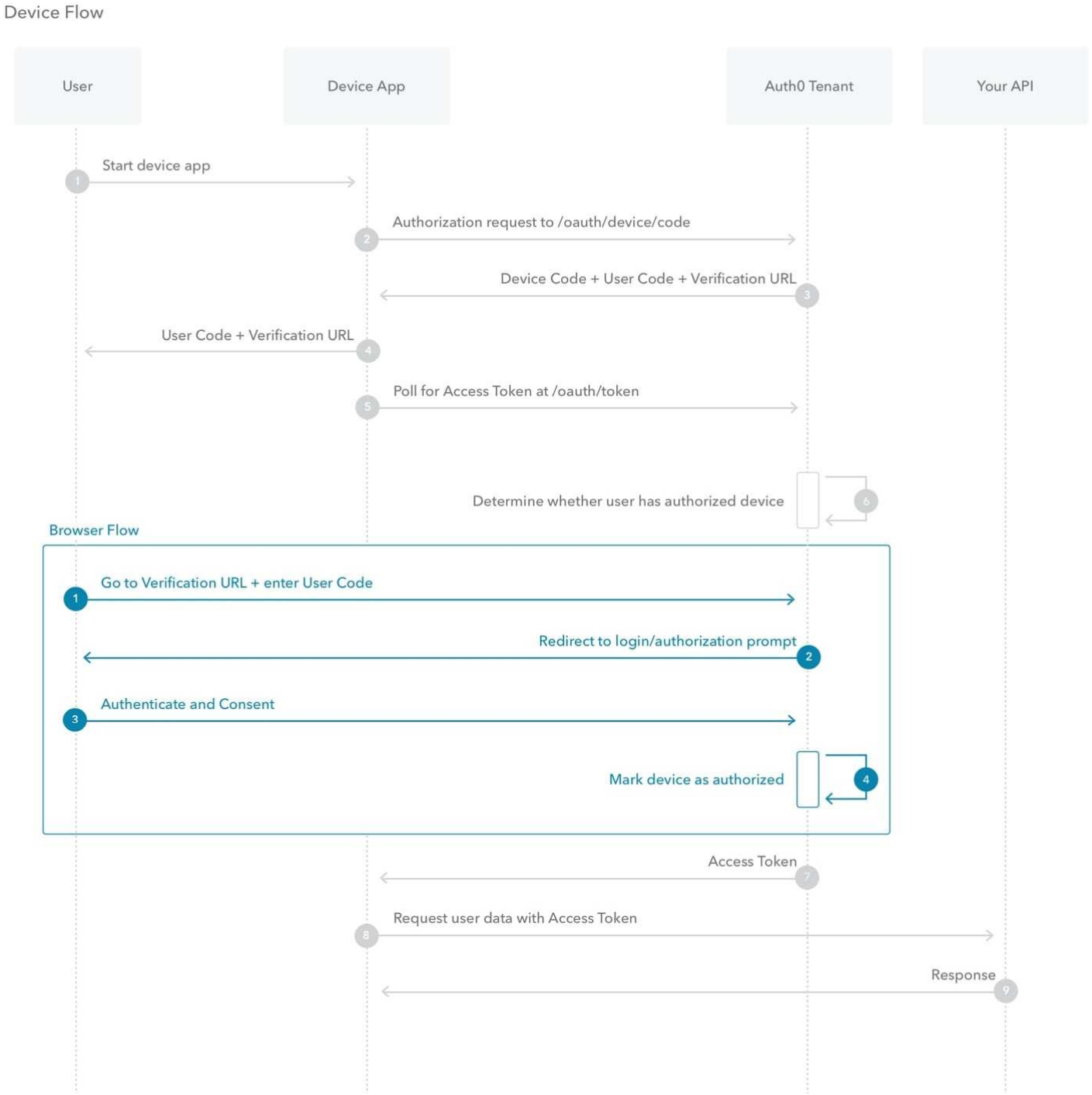


Authorization Code Flow with Proof Key for Code Exchange (PKCE).

- Полный процесс описанный в rfc7637:
- A. Клиент создает и записывает секрет с именем «code_verifier» и получает преобразованную версию "t(code_verifier)" (или «code_challenge»), которая отправляется по протоколу OAuth 2.0 в запросе авторизации вместе с методом преобразования "t_m".
- B. Конечная точка авторизации отвечает как обычно, отправляя код авторизации, но записывает "t(code_verifier)" и метод преобразования "t_m".
- C. Затем клиент отправляет код авторизации как обычно в точку токена авторизации, но включает сгенерированный секрет «code_verifier» из шага (A).
- D. Сервер авторизации преобразует «code_verifier» и сравнивает его в "t(code_verifier)" из (B). Доступ запрещен, если они не равны.
- Злоумышленник, перехватывающий код авторизации в точке (B), не может обменять его на токен доступа, так как не владеет секретом "code_verifier".

Implicit flow





Device code flow