

Курс «Администратор Linux»

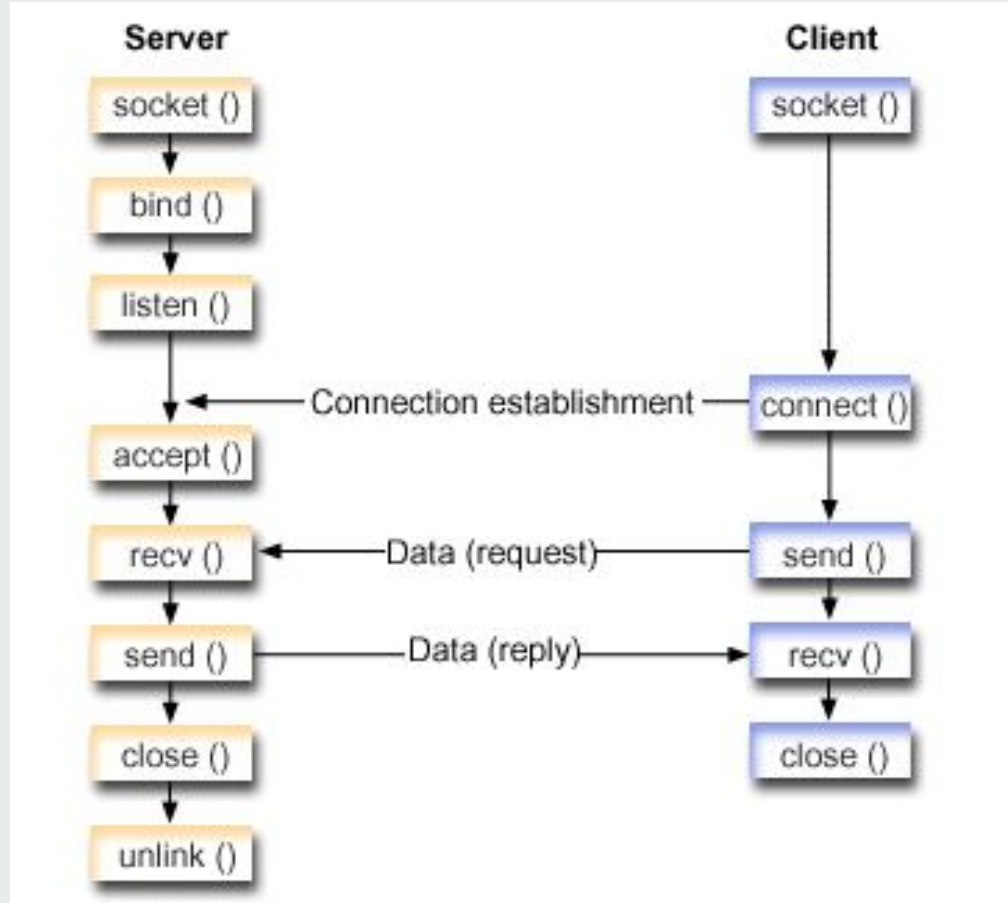
Динамический WEB

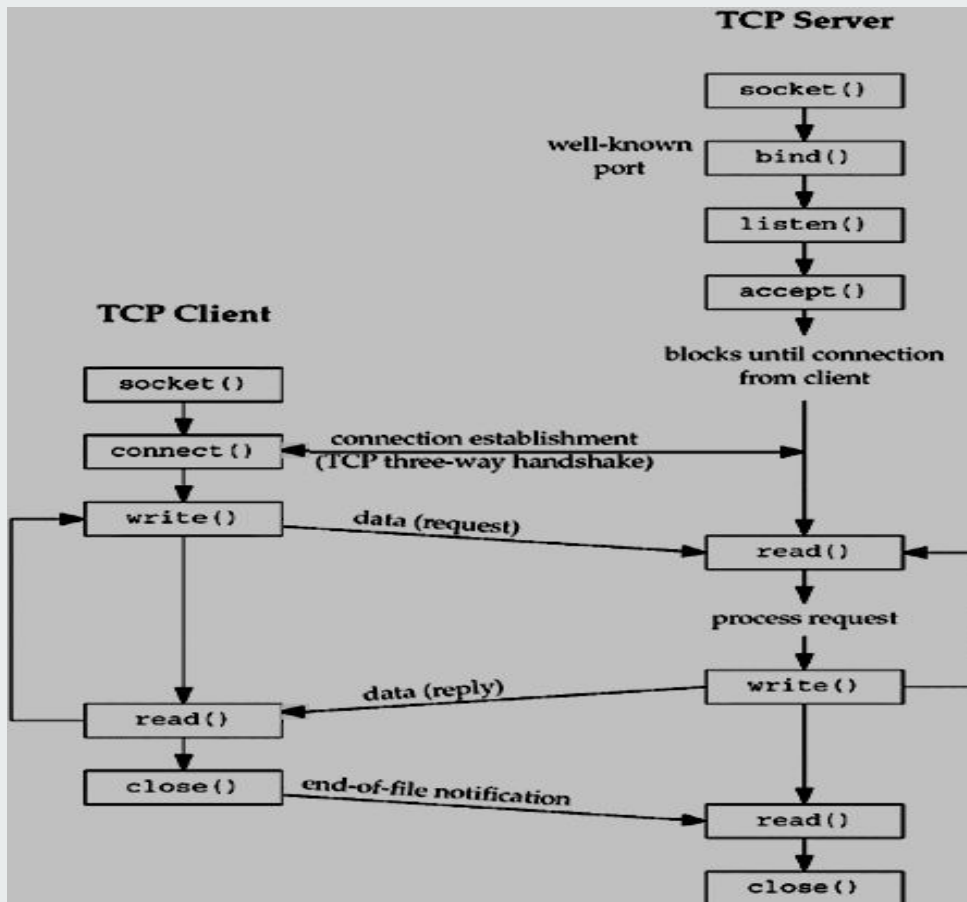
Занятие # 25

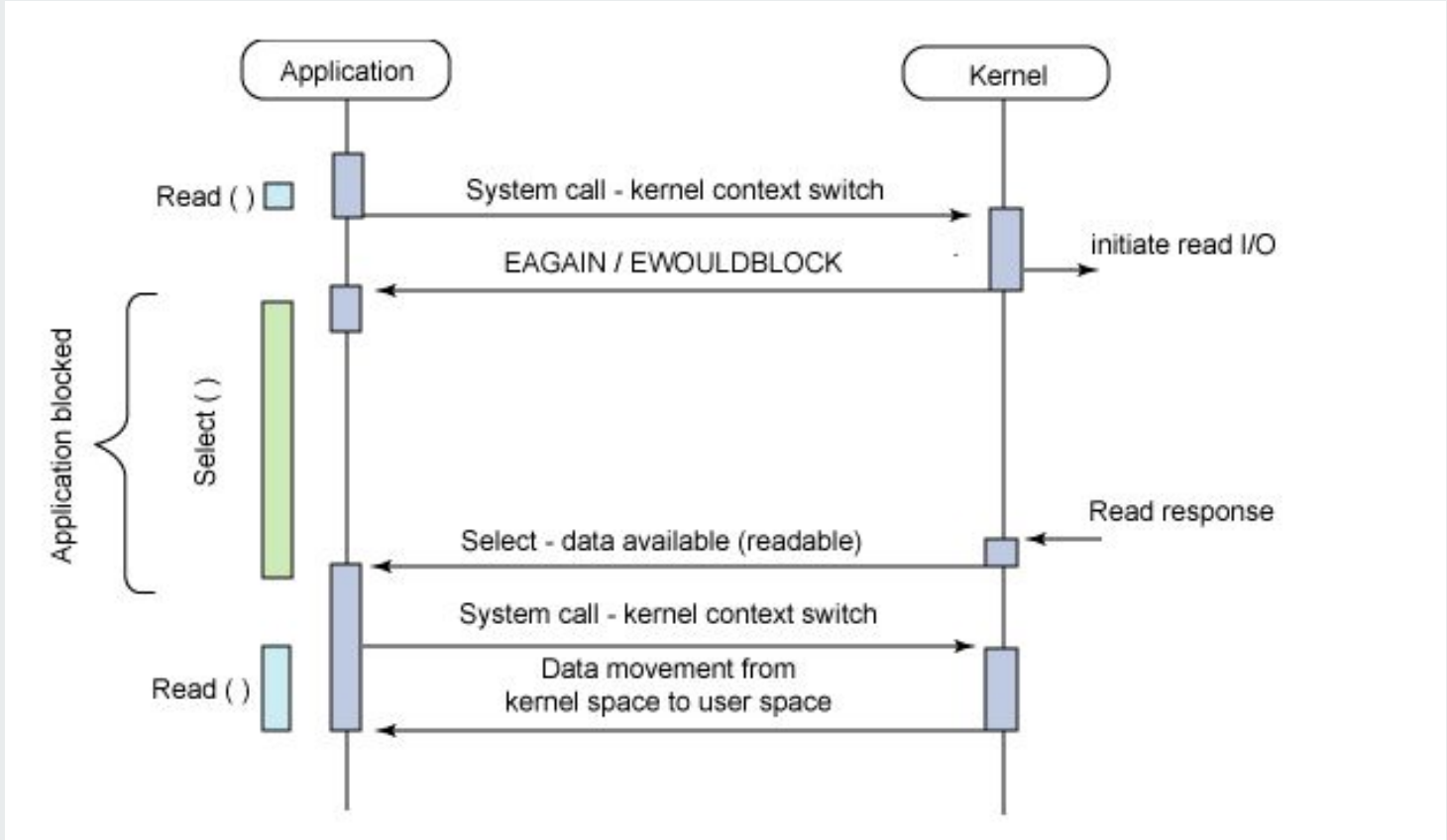
Алексей Цыкунов



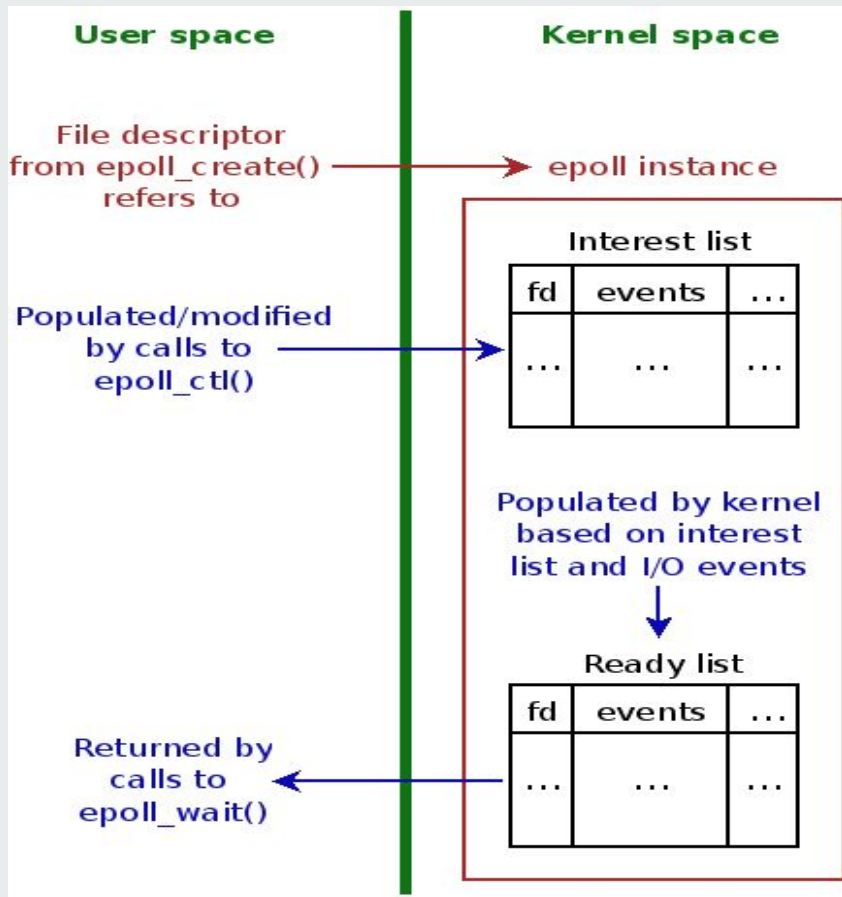
- сокетах
- Принципах работы веб порталов
- CGI
- mod_xxx
- FastCGI, PHP-fpm
- WSGI, uWSGI
- Ajax
- Websocket



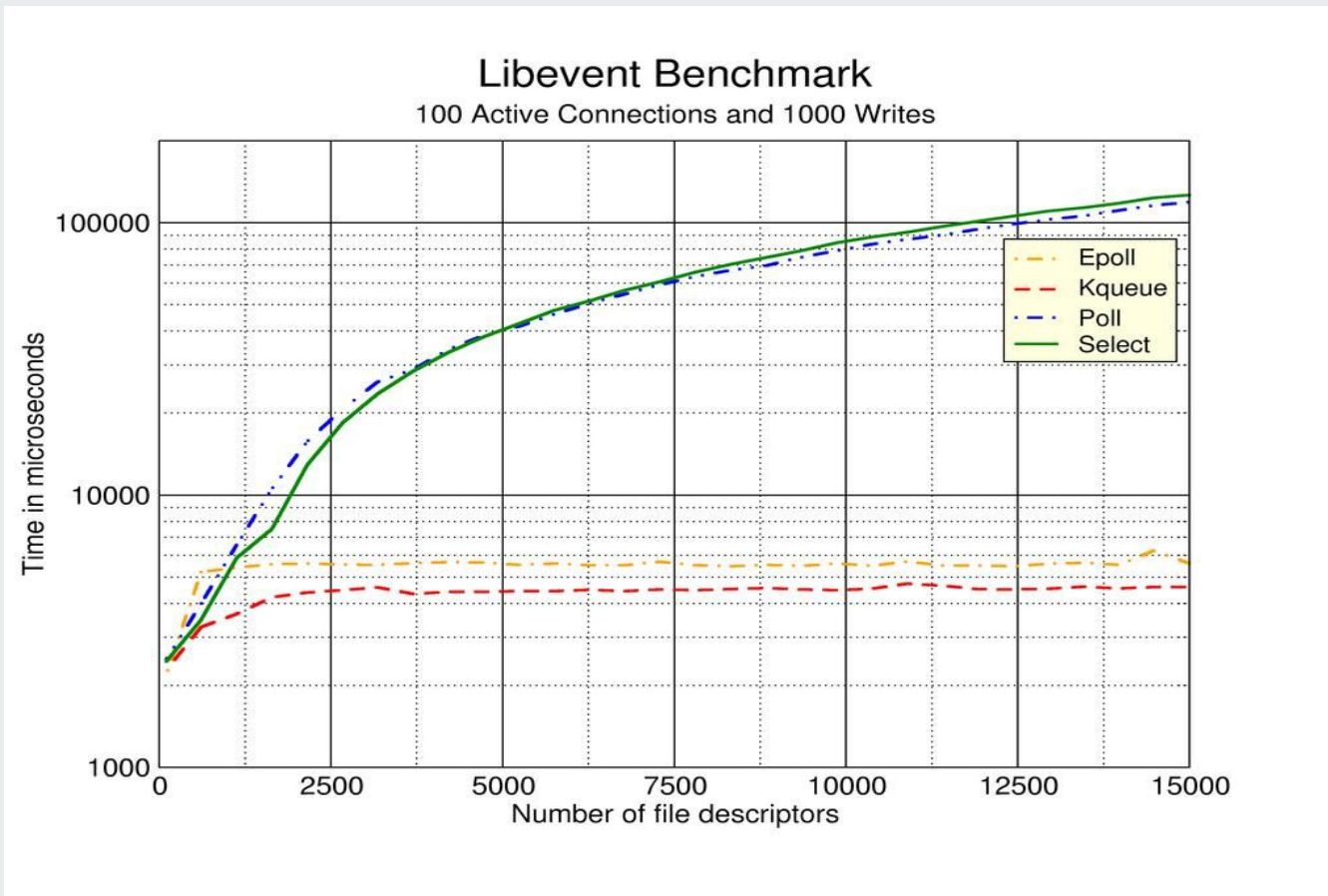




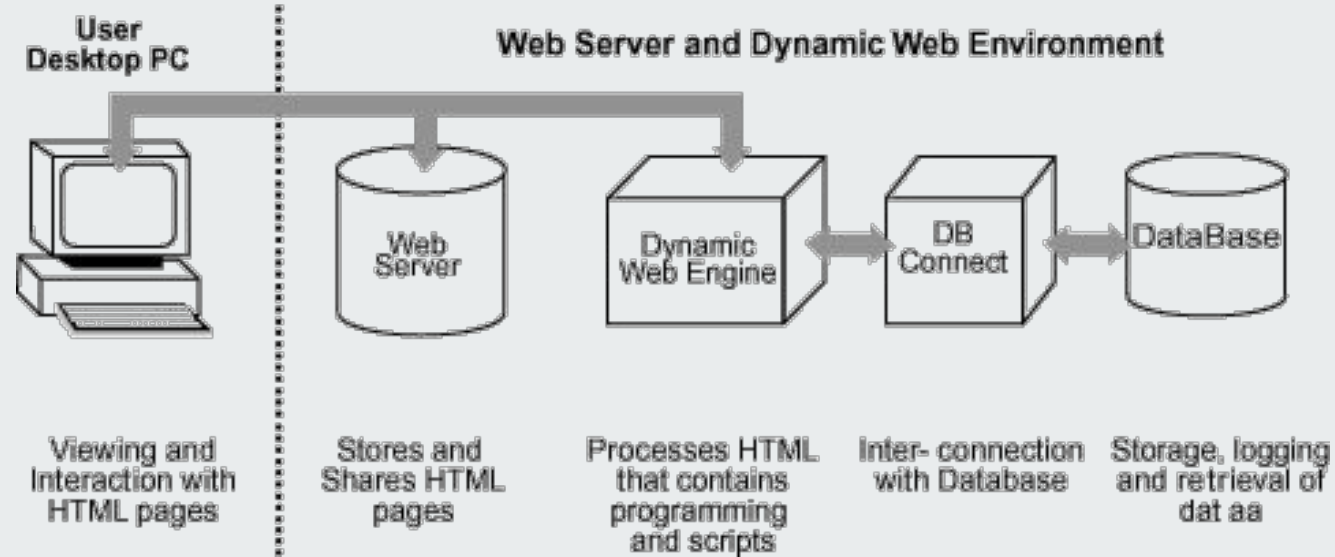
- На каждом вызове опрашиваются ВСЕ файловые дескрипторы на предмет изменения статуса
- Ограниченный набор дескрипторов (select())



- API мультиплексированного ввода-вывода, позволяет приложениям осуществлять мониторинг нескольких открытых файловых дескрипторов
- планировался как более эффективная замена вызовам `select()` и `poll()`, определёнными в POSIX.
- может предоставить более эффективный механизм для приложений, обрабатывающих большое количество одновременно открытых соединений — со сложностью $O(1)$ в отличие от стандартного механизма, обладающего сложностью $O(n)$.
- `/proc/sys/fs/epoll/max_user_watches` - максимальное кол-во наблюдаемых fd для каждого RealUID.

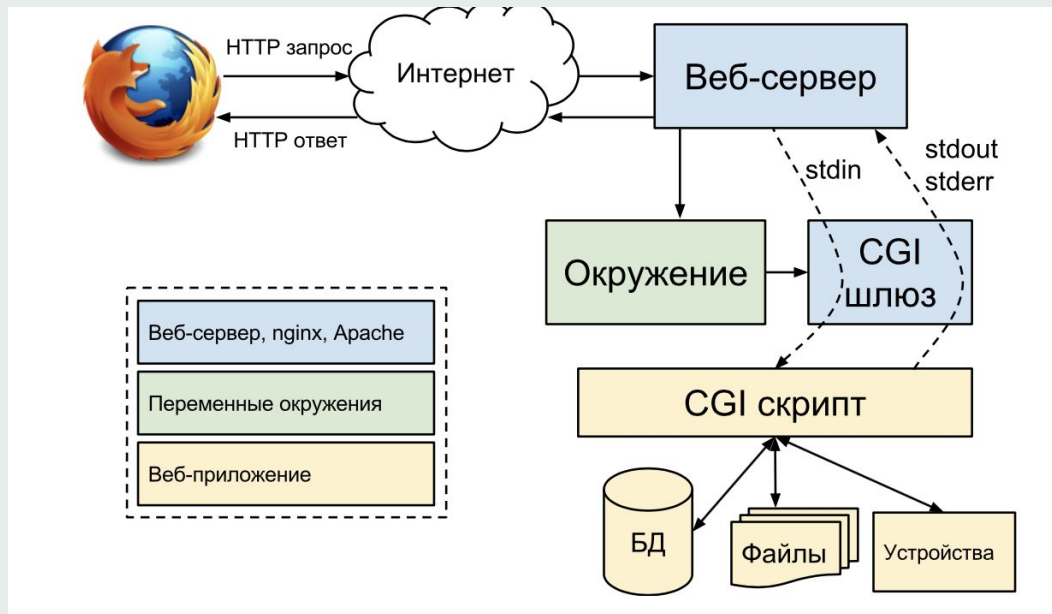


General Architecture for Dynamic Web Solutions



CGI (Common Gateway Interface) - стандарт интерфейса, используемого для связи внешней программы с веб-сервером.

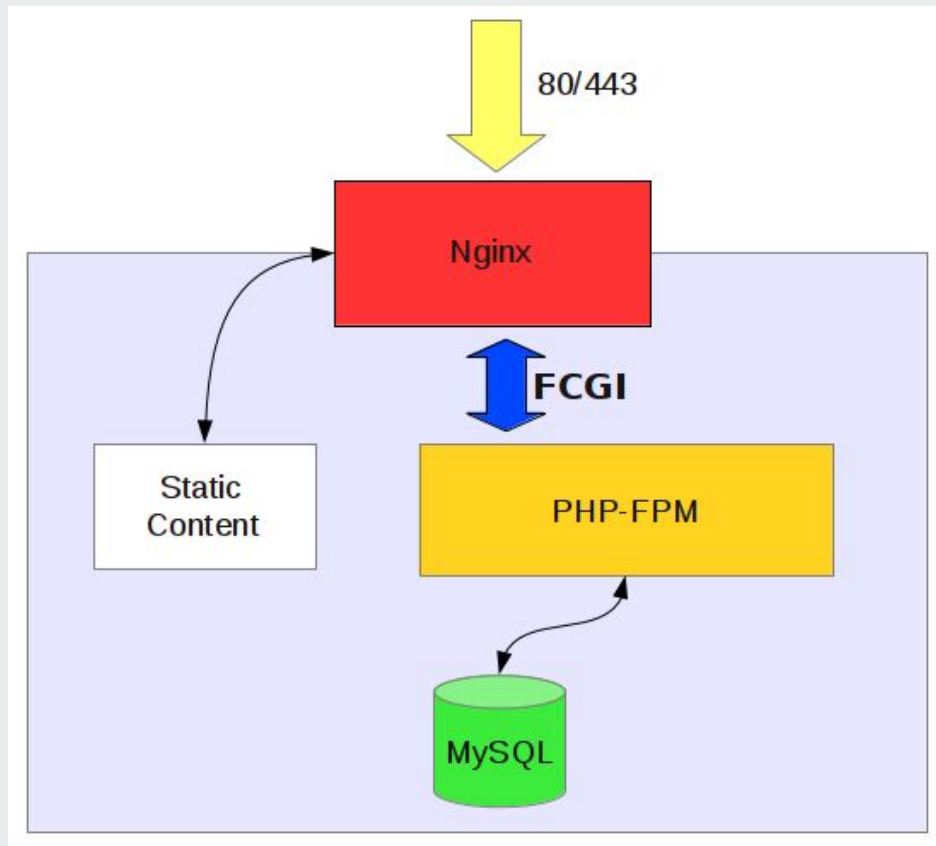
Веб сервер получая запрос вызывает внешний скрипт передавая в него параметры и переменные окружения и возвращает его вывод в качестве ответа



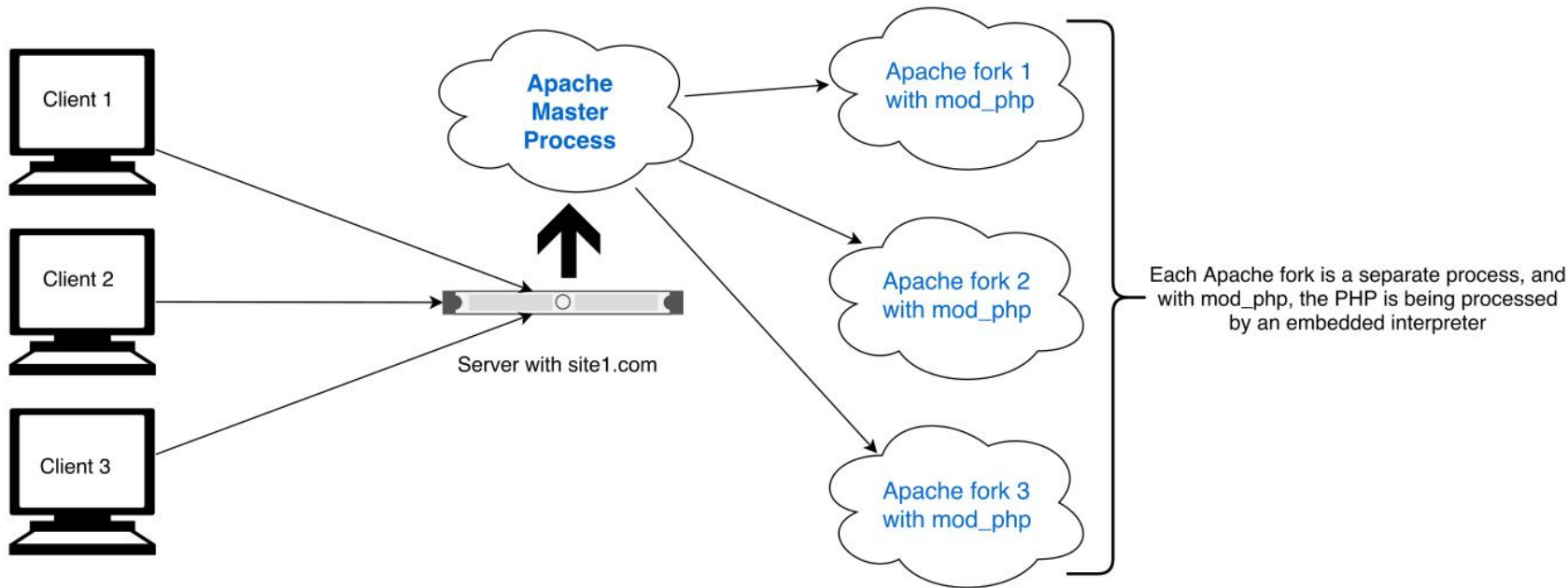
программа единожды загружается в память в качестве демона (независимо от HTTP-сервера), а затем входит в цикл обработки запросов от HTTP-сервера.

Один и тот же процесс обрабатывает несколько различных запросов один за другим, что отличается от работы в CGI-режиме

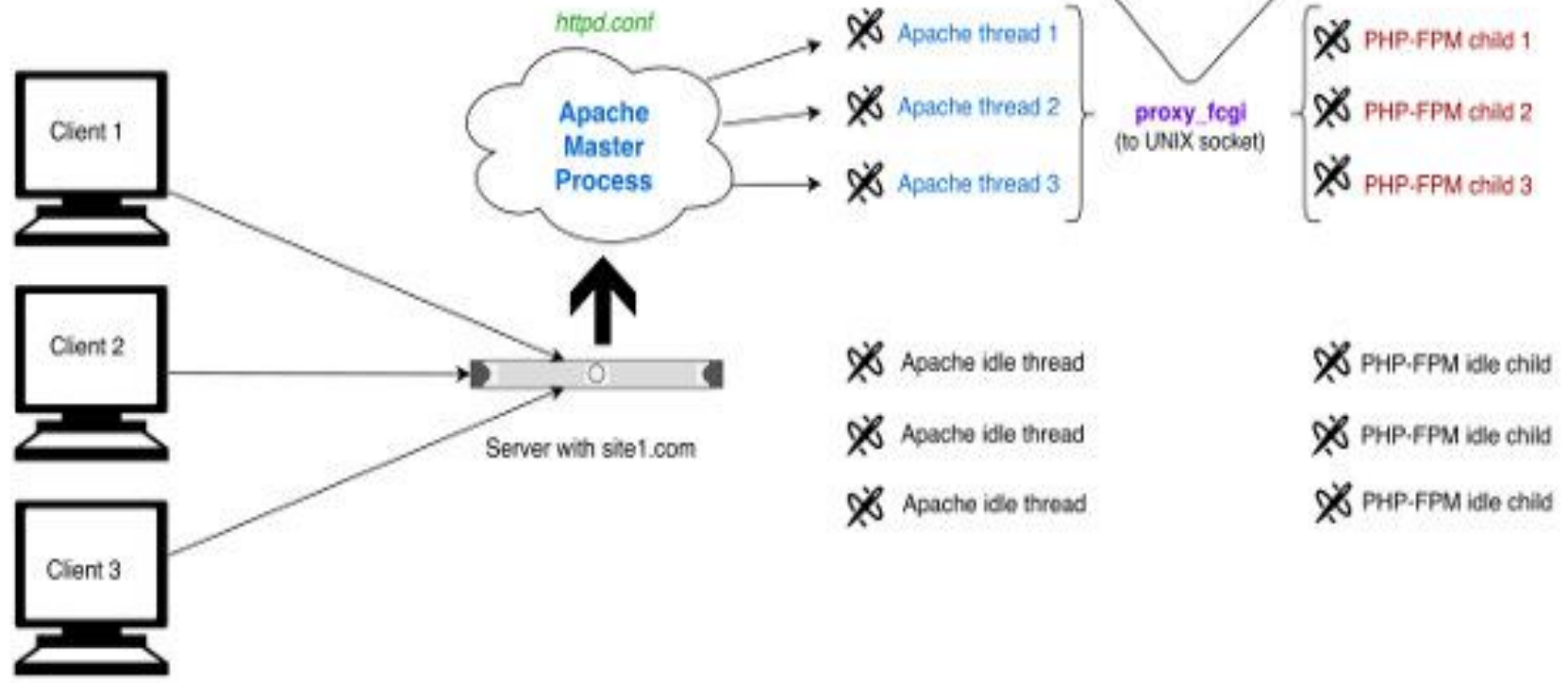
SCGI - аналог (mod_scgi)

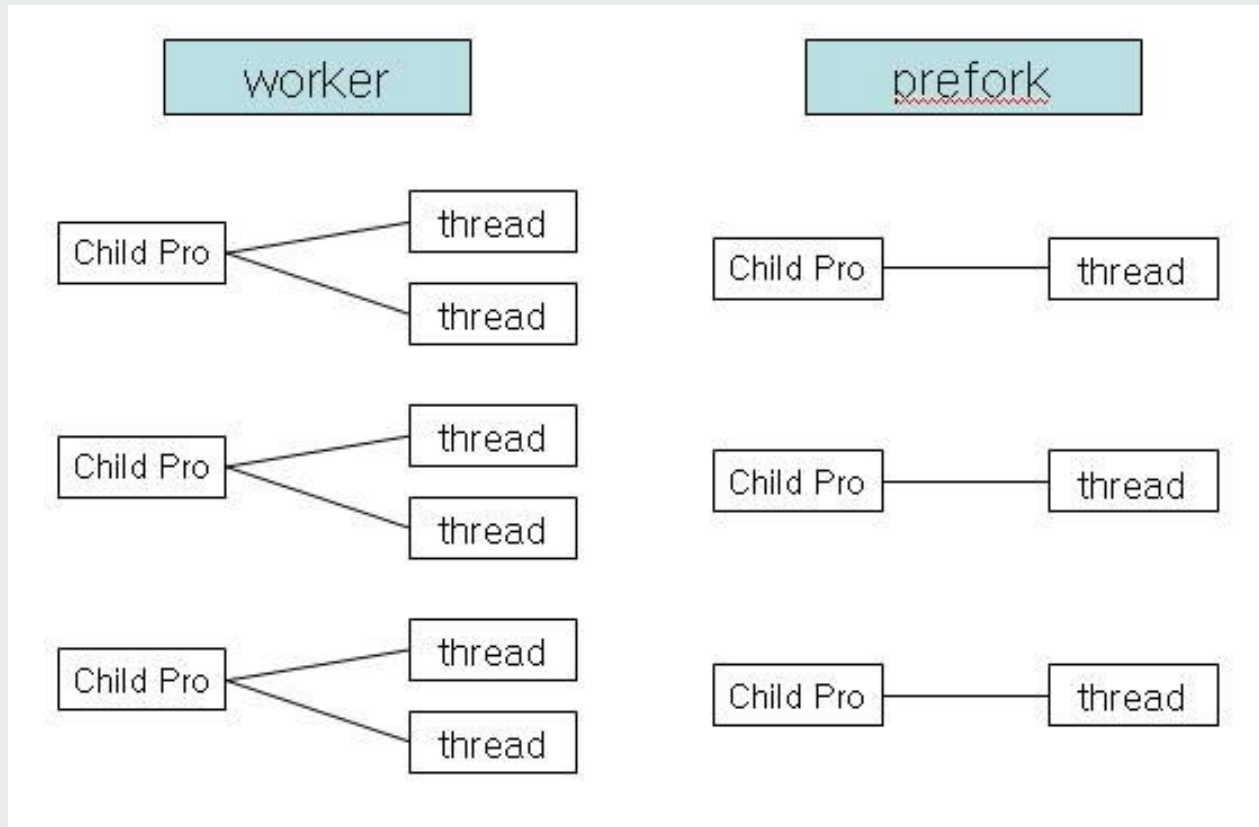


Apache with Prefork MPM and mod_php



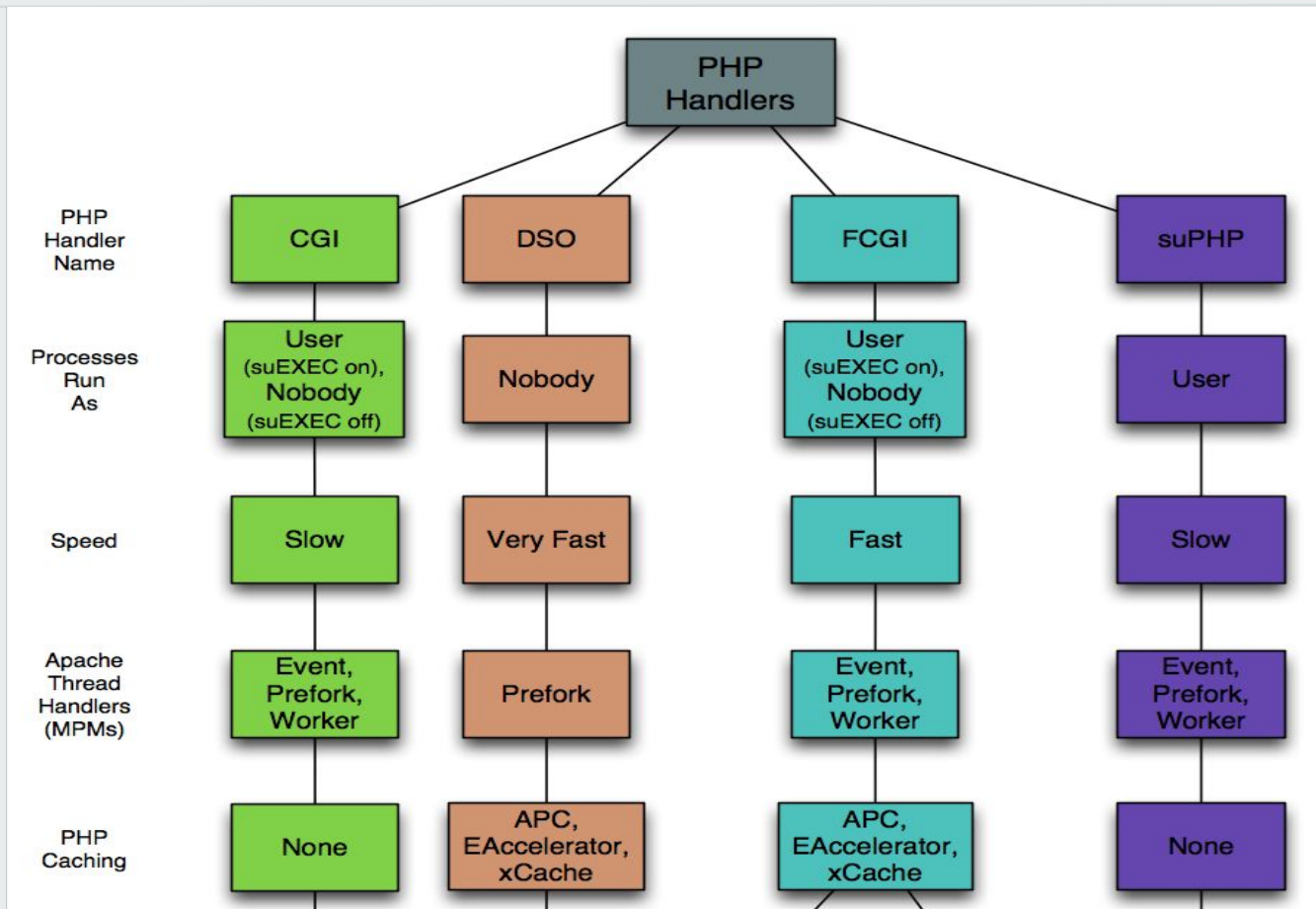
Apache with Event MPM and mod_proxy_fcgi to PHP-FPM

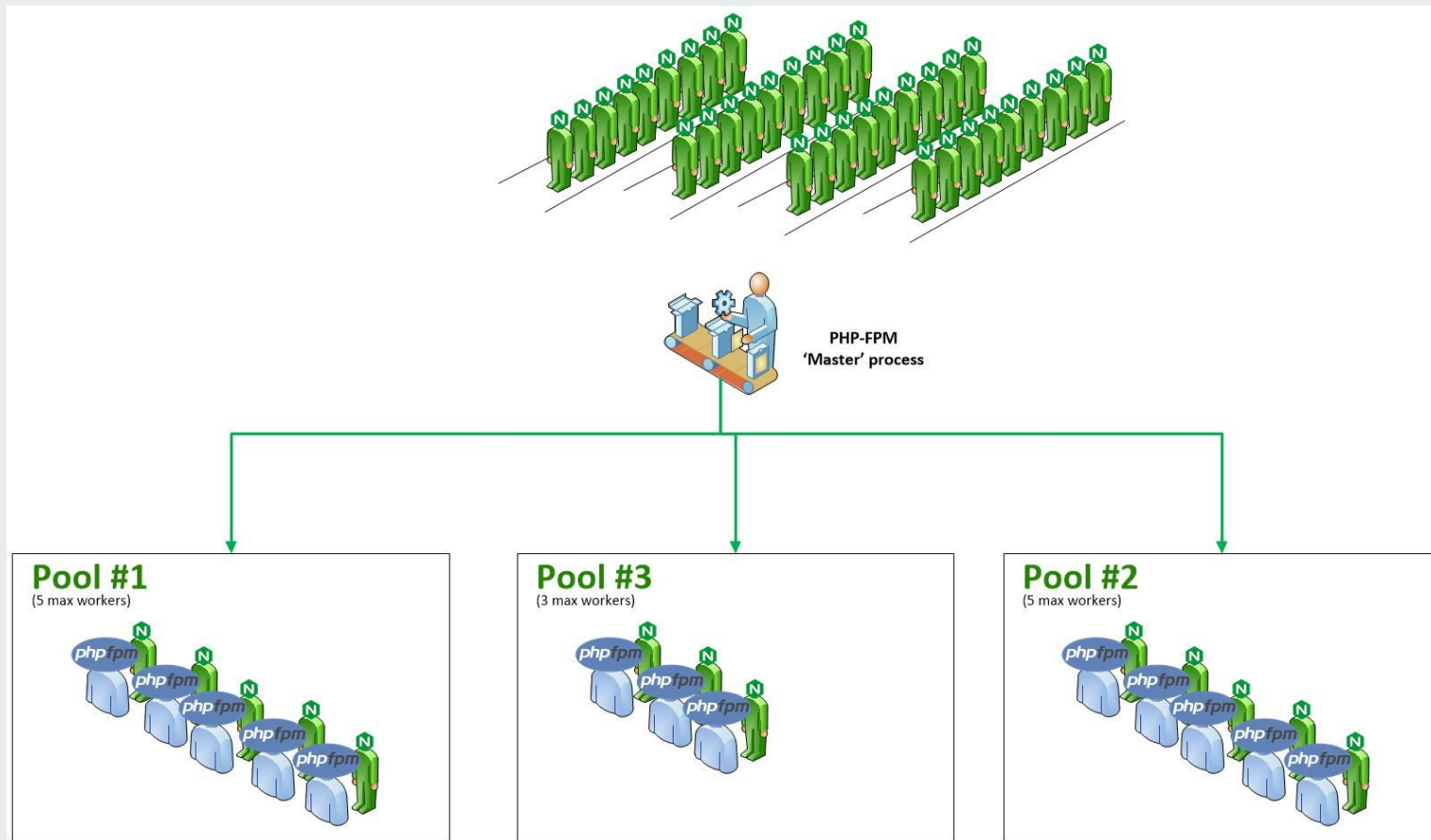




	Prefork	Event	Worker	ITK
threaded?	no	yes	yes	no
parent run as	nobody	nobody	nobody	root
children run as	nobody	nobody	nobody	user
can use DSO?	yes	no	no	yes
thread/child per?	connection	request	connection	connection
resource usage	high	lowest	low	high

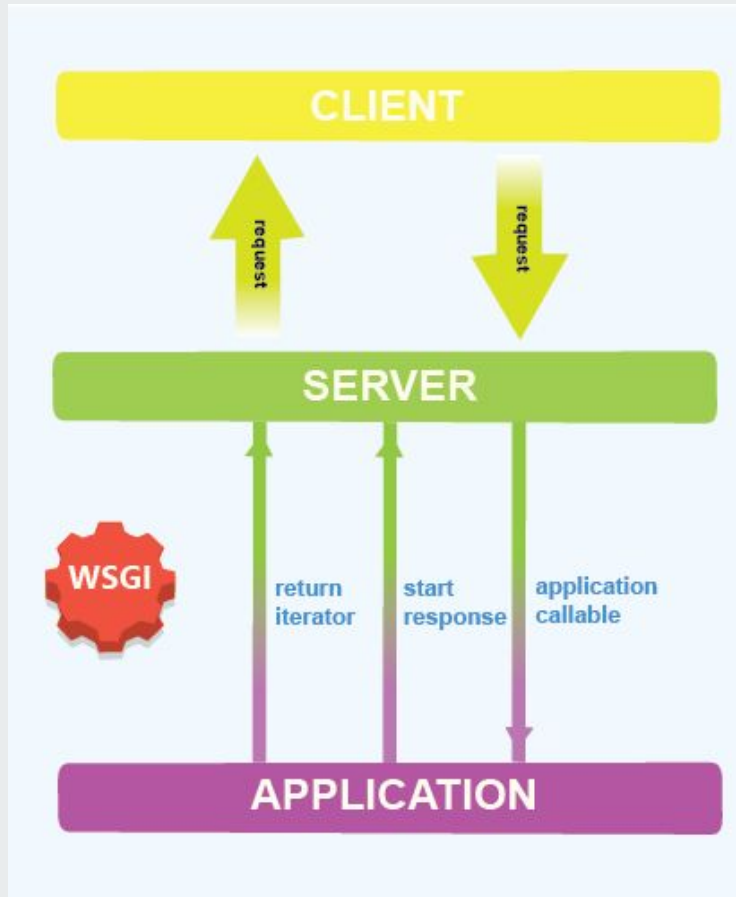
	mod_php	CGI	suPHP	FastCGI
Memory usage	Low	Low	Low	High
CPU Usage	Low	High	High	Low
Security	Low	Low	High	High
Run as file owner	No	No	Yes	Yes
Overall Performance	Fast	Slow	Slow	Fast



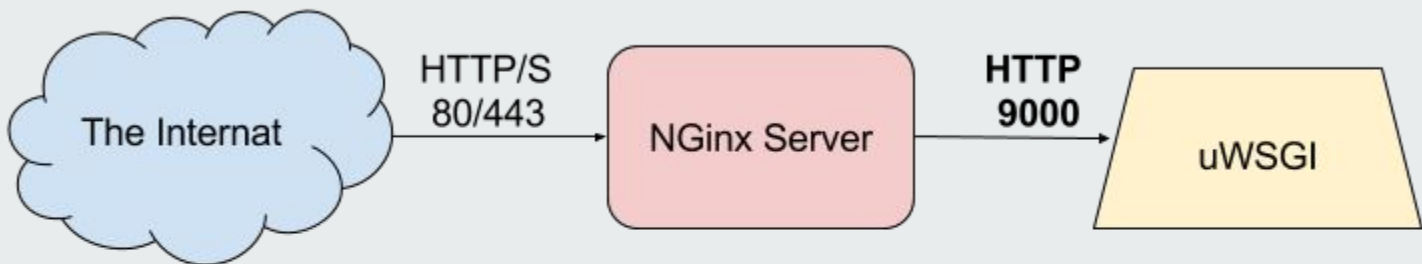


стандарт взаимодействия между Python-программой, выполняющейся на стороне сервера, и самим веб-сервером

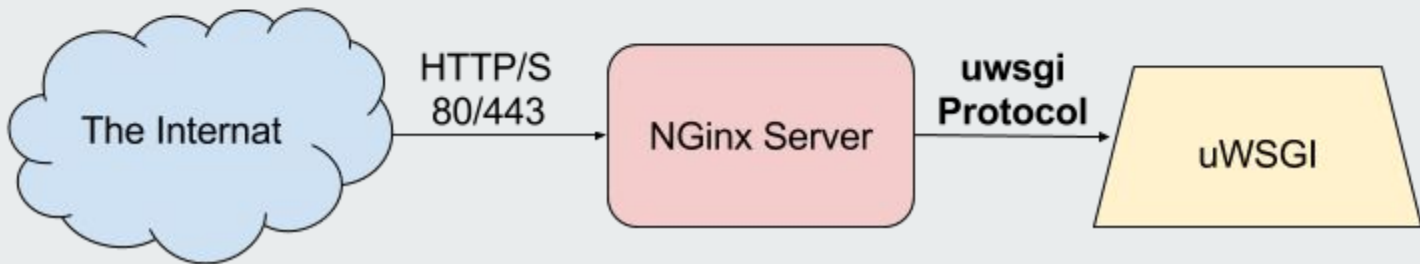
- должно быть вызываемым (**callable**) объектом (обычно это функция или метод)
- принимать два параметра:
 - словарь переменных окружения (**environ**)
 - обработчик запроса (**start_response**)
- вызывать обработчик запроса с кодом HTTP-ответа и HTTP-заголовками
- возвращать итерируемый объект с телом ответа

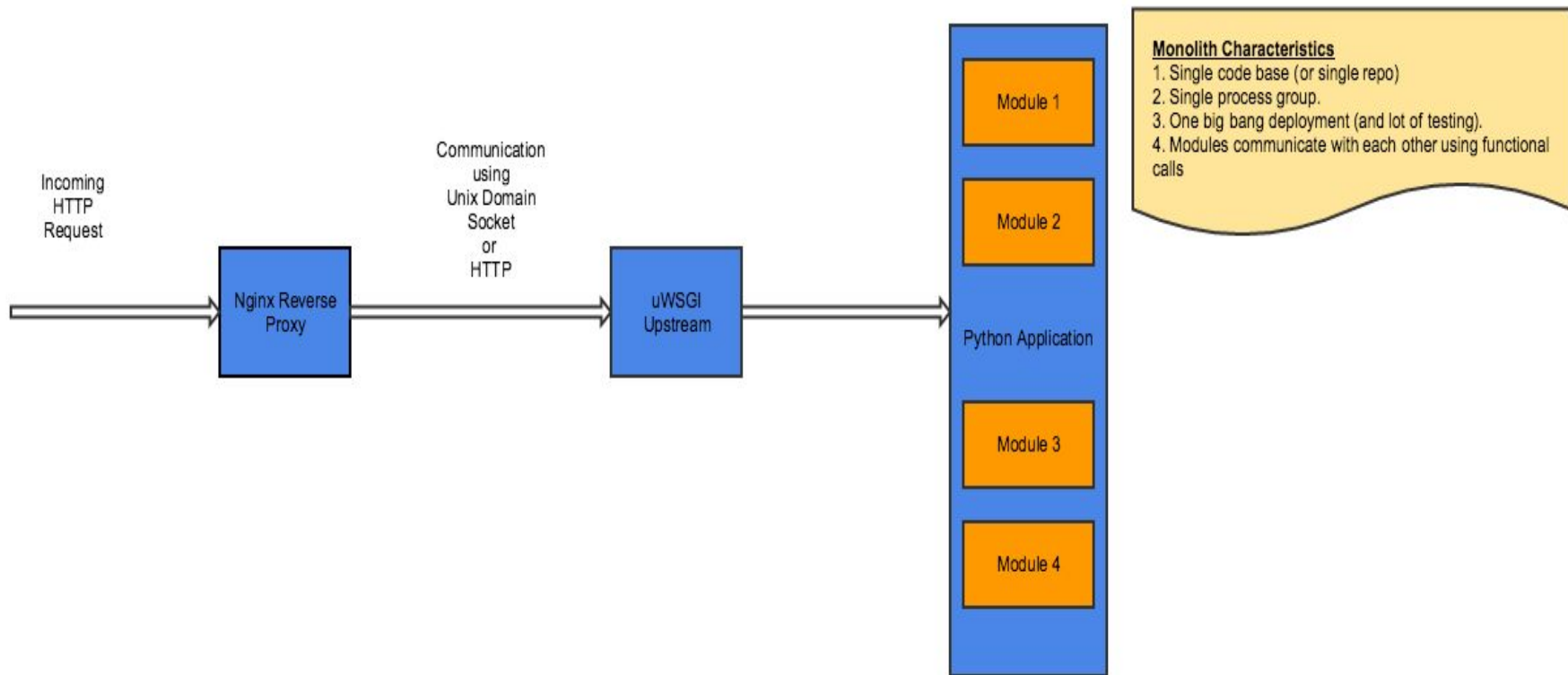


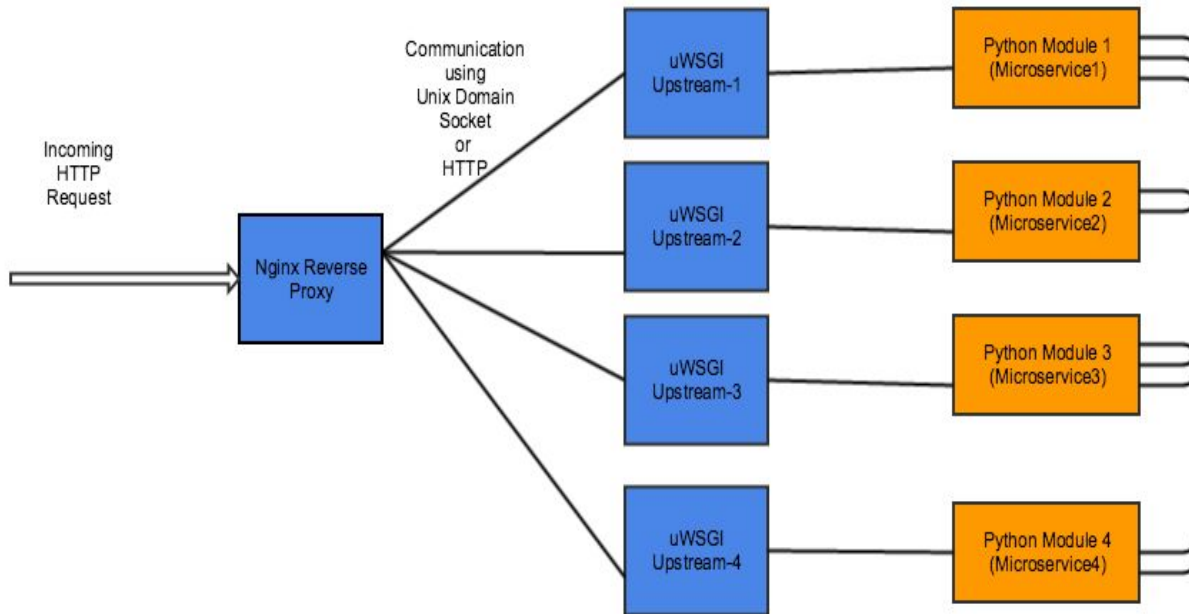
Option 1



Option 2



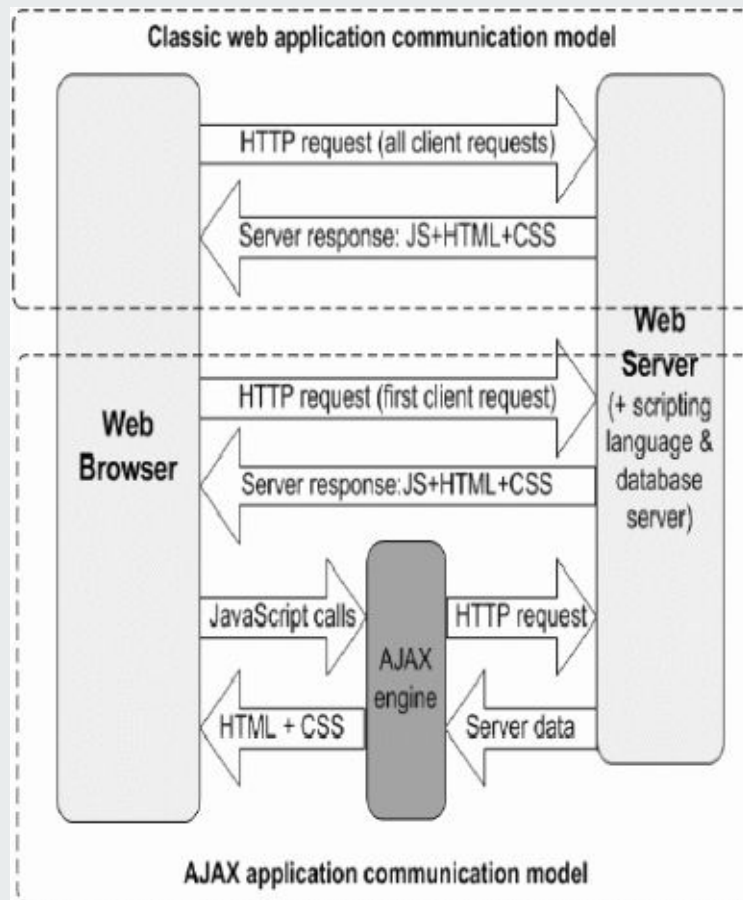


**Microservices Characteristics**

1. Multiple repos (or subrepos)
2. Multiple process group.
3. Only changed services need to be deployed (and tested)
4. Each service communicates to each other using http or other lightweight protocol.

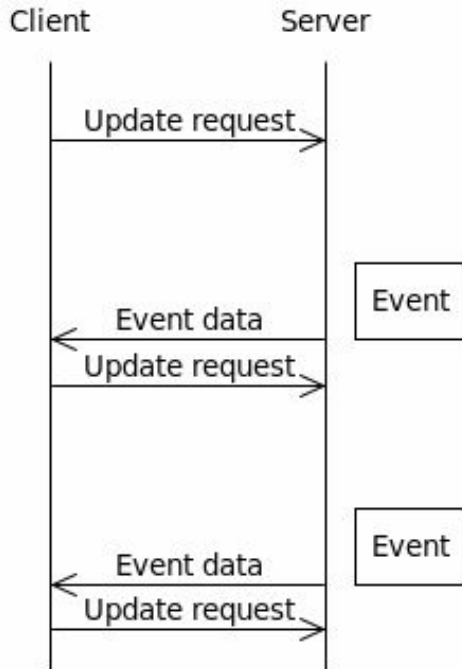
Inter micro-service
communication using http

1. Клиент посылает запрос на вебстраницу серверу, используя обычный **http**(regular http)
2. Запрошенная страница выполняет **JavaScript**, который запрашивает файл от сервера через какой-то интервал времени (например, 0.5 секунд)
3. Сервер формирует ответ для каждого запроса и отправляет его обратно



1. Клиент запрашивает страницу у сервера, используя обычный **http**
2. Запрошенная страница выполняет **JavaScript**, который запрашивает файл от сервера.
3. Сервер **НЕ** реагирует на запрошенную информацию и ждет, пока не появится новой информации
4. Когда появляется новая информация, сервер отправляет ее клиенту
5. Клиент получает новую информацию и **НЕМЕДЛЕННО** отправляет другой запрос серверу, запуская процесс ожидания на нем снова.

AJAX long polling

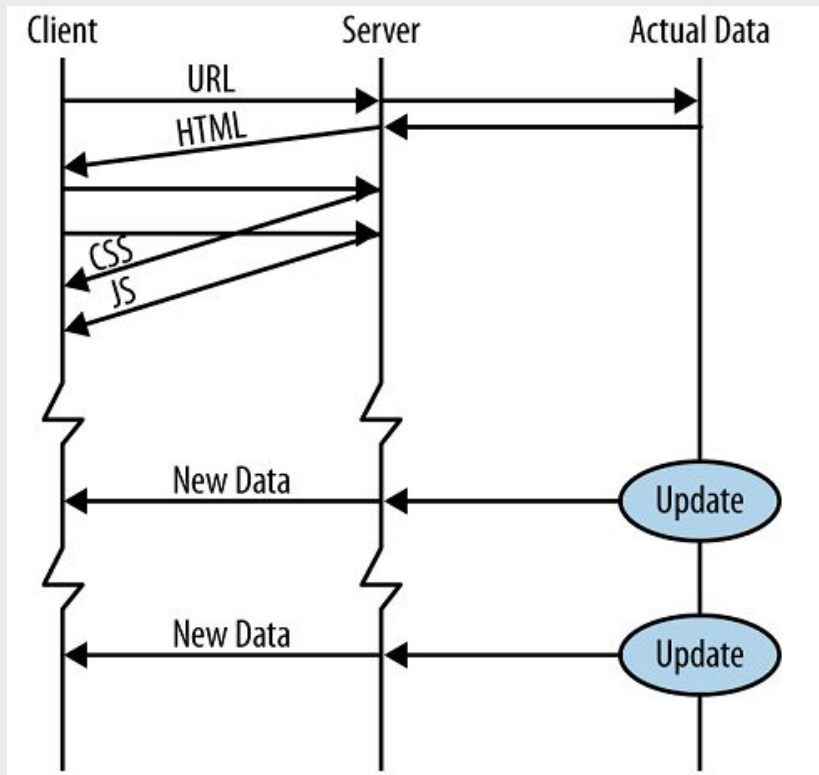


1. Клиент запрашивает страницу у сервера, используя обычный **http**

2. Запрошенная страница выполняет **JavaScript**, который открывает соединение с сервером

3. Сервер посылает событие клиенту, когда появляется новая информация

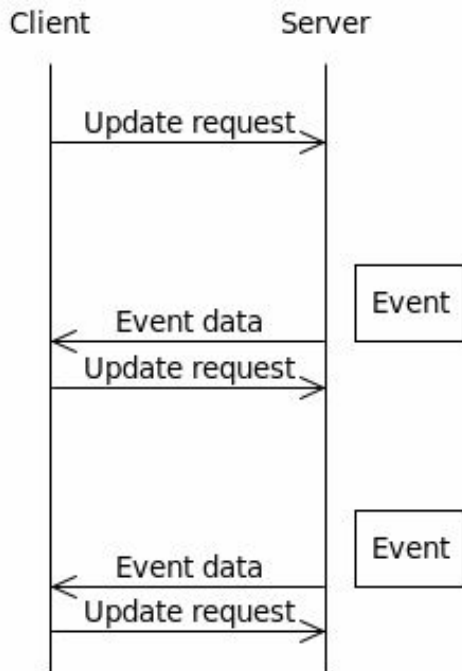
- Трафик в реальном времени от сервера клиенту, главным образом, то, что вам нужно
- Вы захотите использовать сервер, который имеет цикл событий
- Нет возможности соединиться с сервером с другого домена



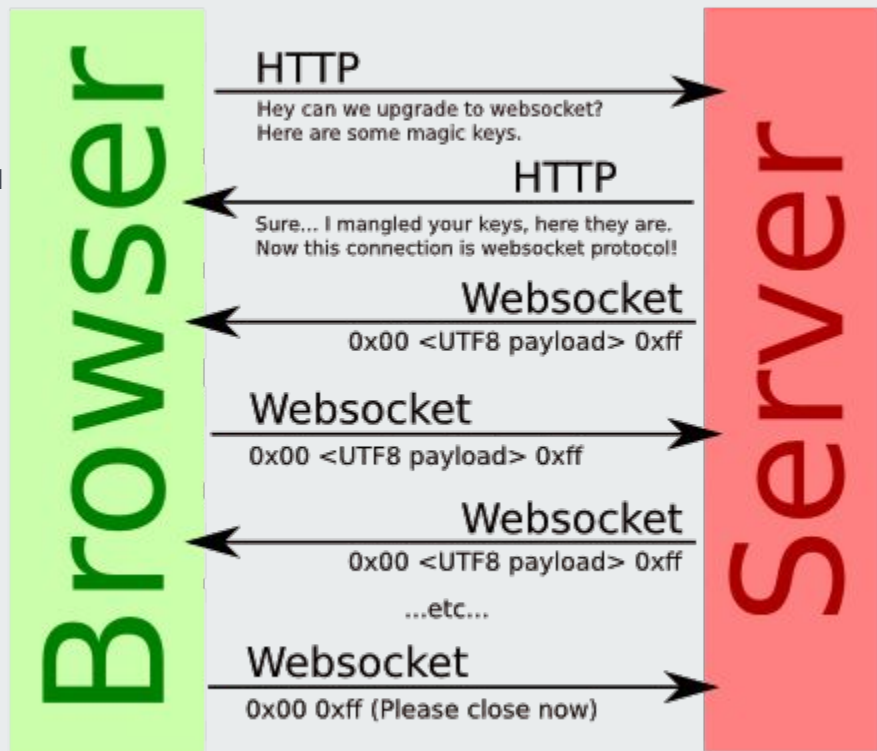
AJAX Long-Polling

1. Клиент запрашивает страницу у сервера, используя обычный **http**
2. Запрошенная страница выполняет **JavaScript**, который запрашивает файл от сервера.
3. Сервер **НЕ** реагирует на запрошенную информацию и ждет, пока не появится новая информация
4. Когда появляется новая информация, сервер отправляет ее клиенту
5. Клиент получает новую информацию и **НЕМЕДЛЕННО** отправляет другой запрос серверу, запуская процесс ожидания на нем снова.

AJAX long polling



1. HTTP handshake на подтверждение поддержки websocket
2. Работает на установленном TCP соединении
3. Фрейм - 0x00 <UTF8 payload> 0xff
4. Сообщения могут идти в обе стороны
5. ws://ws.my.site
6. wss://wss.my.site



Спасибо за внимание

Алексей Цыкунов

