

Верхнеуровневая архитектура высоконагруженного сервиса

{«Андрюхин Илья Евгеньевич»: «git@github.com:lilchiken/test_task_vk.git», «date»: «2023-03-01»}

VK реклама

Python-разработчик

Ищем человека в проект автогенерации рекламного контента.
Стажёр будет участвовать в разработке и поддержке
микросервисов автоматической генерации контента.

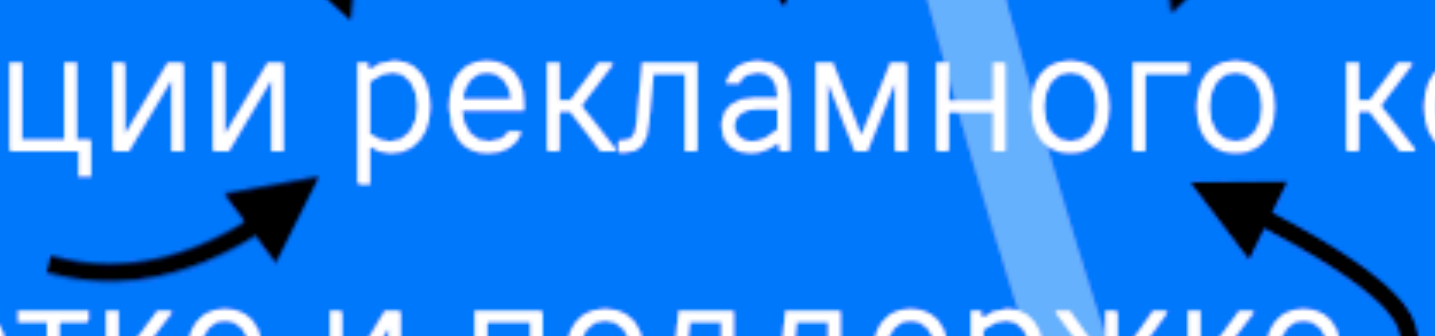
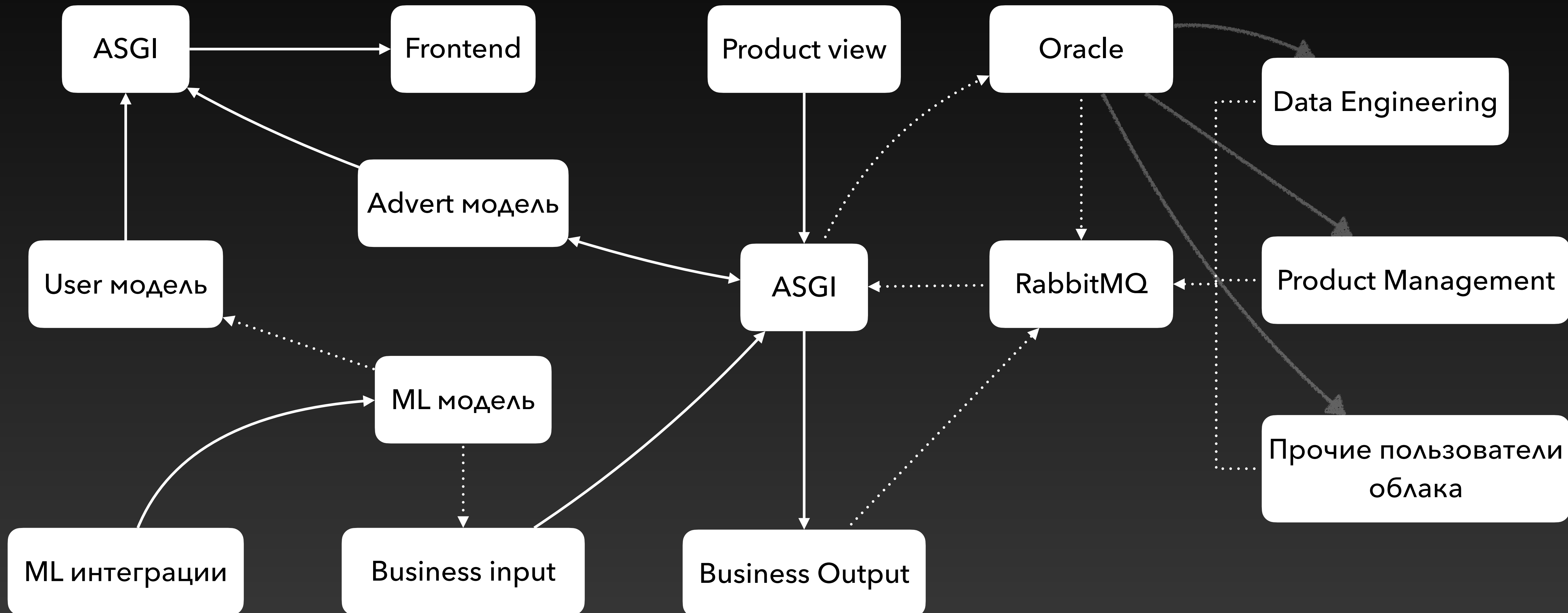


Схема в голове

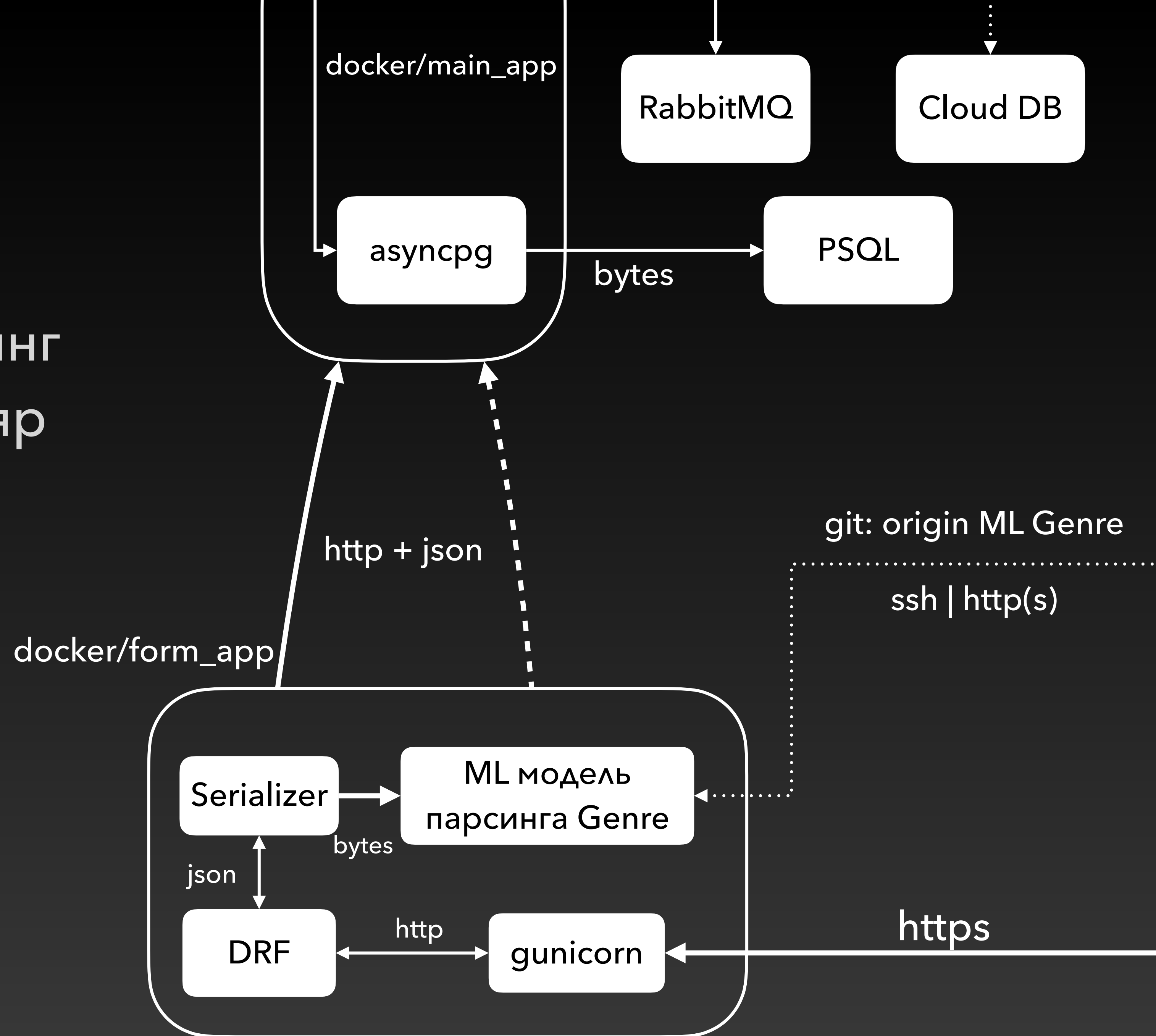


Business input

Первая задача - принятие формы от Customer, парсинг Genre, составить экземпляр модели Advert.

Deserializing/serializing.

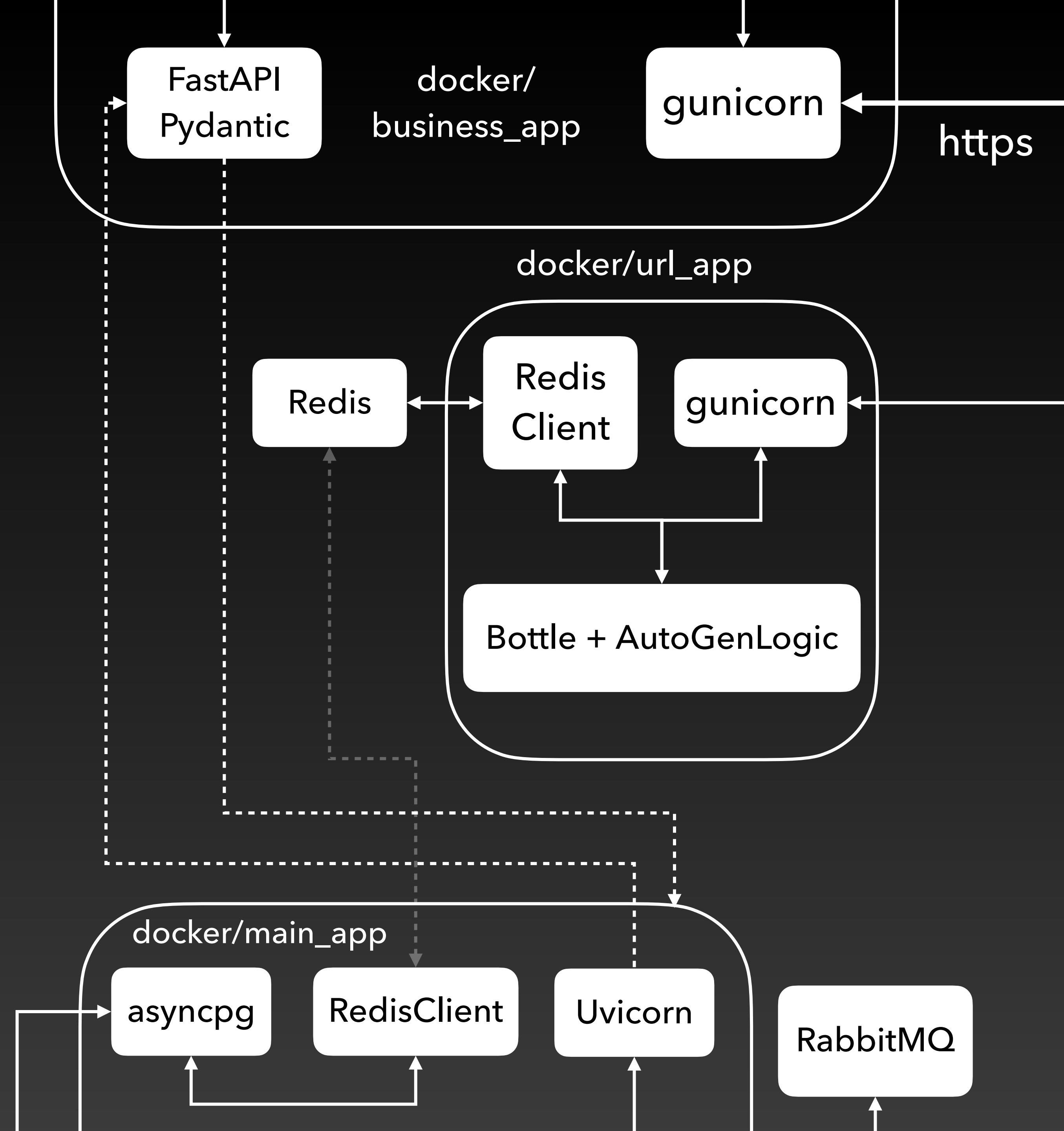
После отправить клон экземпляра в облако отложенным taskom.



Product view

Является следующей задачей после принятия формы. Здесь мы генерируем ссылку, которую будем отслеживать.

Чтобы уменьшить нагрузку на БД, будем кэшировать результаты в отдельном приложении и отправлять их только при достижении каких-либо условий. (запрос отчёта от Customer, прошло N временных единиц, было N количество редиректов)



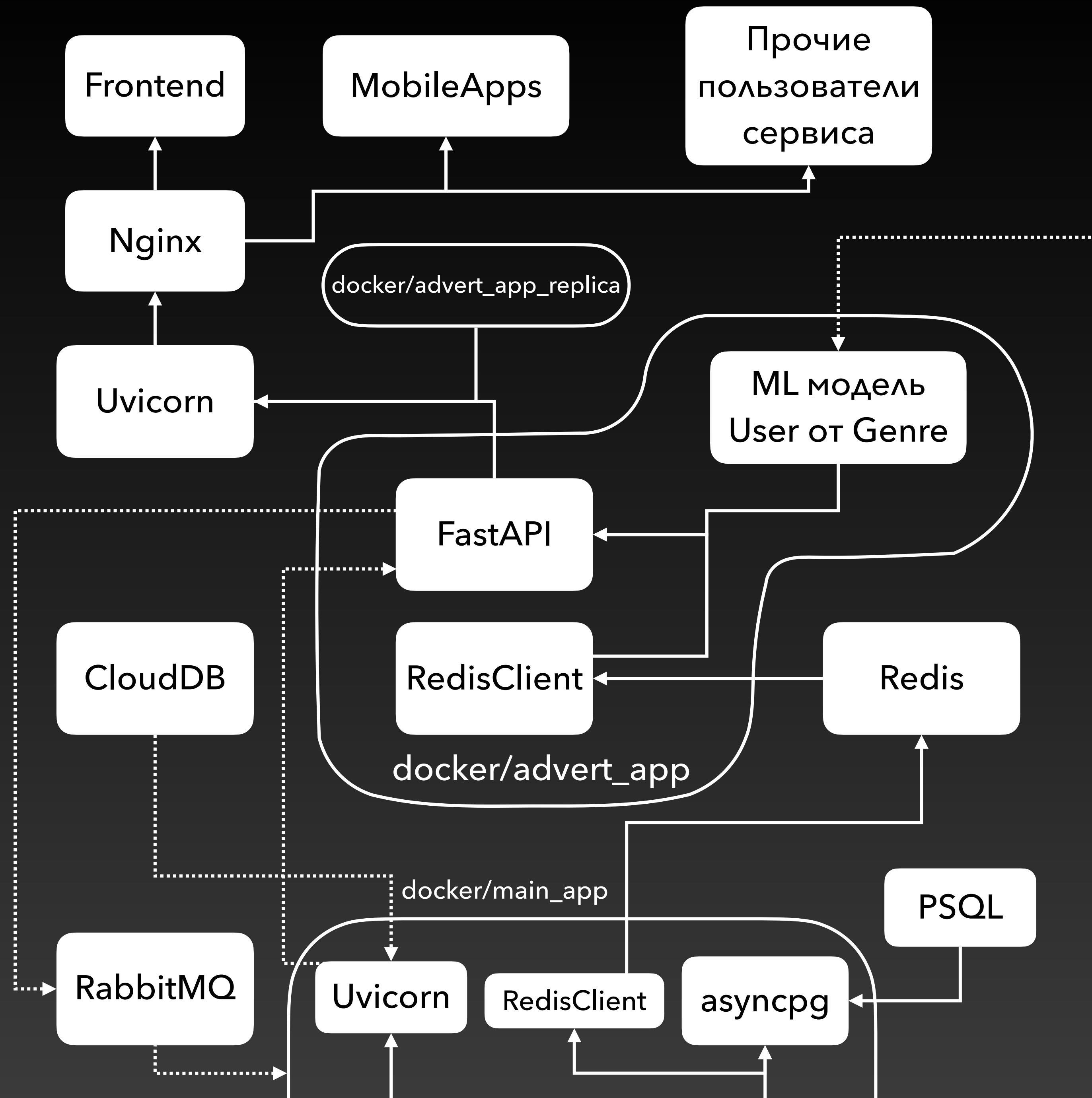
AutoGen Content

Самая важная задача, так как самая нагруженная. Здесь мы нужным User'ам даём нужный Advert.

Чтобы обеспечить быстрый ответ - будем отдавать только важную бизнес часть от Advert экземпляра модели (сжатая картинка, текст, тайтл, ссылка), хранить её в кэше.

Будем асинхронно выгружать из облака оставшуюся бизнес часть модели.

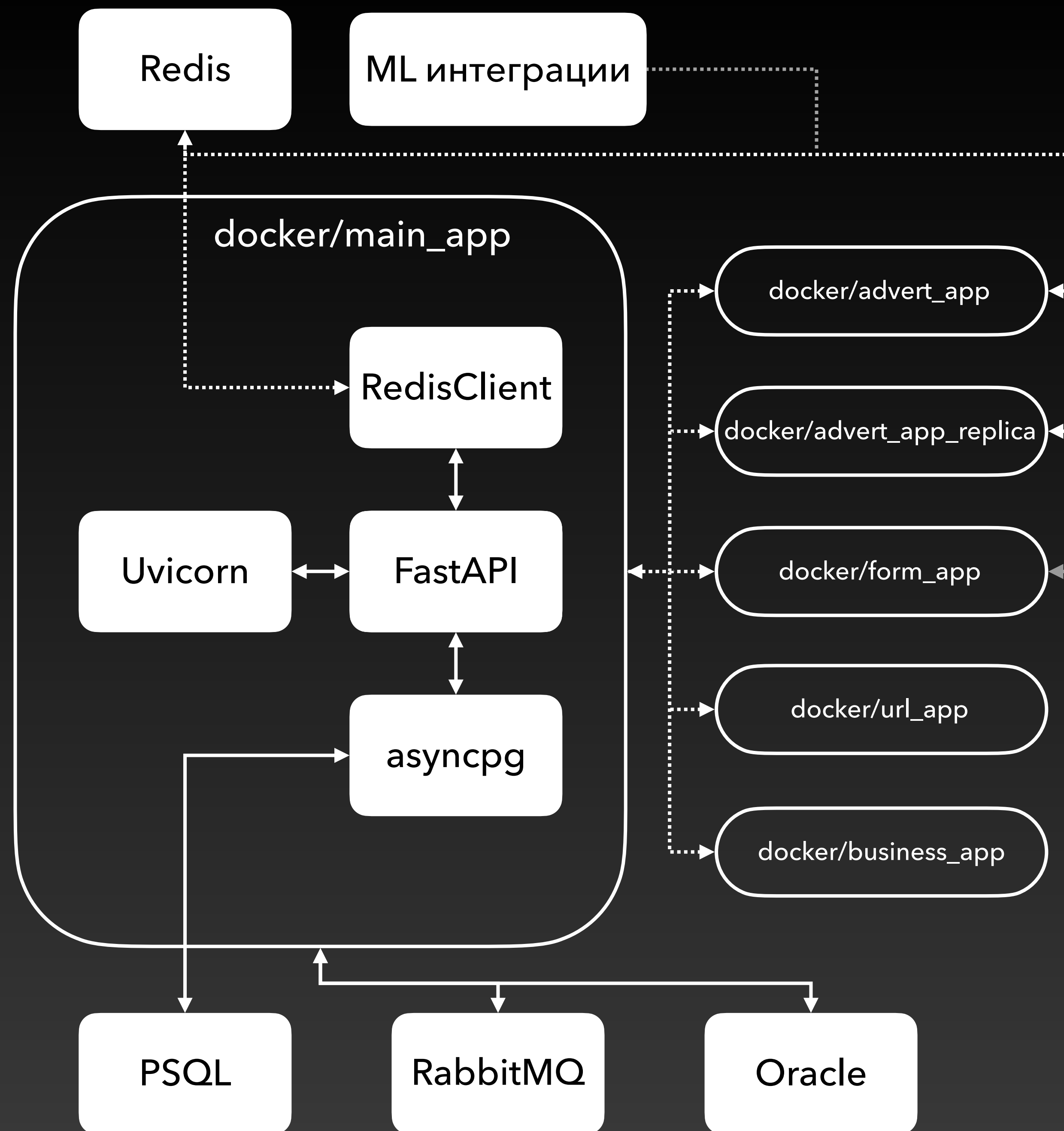
Балансировка и репликация - залог успеха высоконагруженного сервиса.



main_app

Некий монолит, от которого
зависят остальные
микросервисы.

С помощью него мы можем
изолировать все микросервисы
друг от друга.



Стек

- Python3
- FastAPI, Django Rest Framework, Bottle
- Redis, PostgreSQL, Oracle
- RabbitMQ
- Uvicorn, Gunicorn
- Docker, Kubernetes

Протоколы передачи данных

- Между контейнерами - http (cross origin resource sharing) + json
- ML интеграции - ssh
- Для отправки почты - imap, pop3
- Для Business Input/Output - https + ssl
- Для Product View (url_app) - http
- По поводу advert_app:
поскольку за генерацию html сервис не отвечает,
то будем использовать обычный http + json для передачи на фронт.

Системы хранения данных

Будем использовать сервер на Network Attached Storage по протоколу NFS. Является универсальным решением, потому что поддерживает доступ по TCP/IP. Можно использовать такое решение как и для СУБД PostgreSQL, так и для облачного хранилища, используемым Oracle'ом.

Нагрузка на кэш в таком проекте высока, и здесь, важно выбрать оптимальную систему для него. Под такие задачи подойдёт SSD - кэширование с структурой RAID 1+ ([*link*](#)). Такая система не безотказная, но, на мой взгляд, она извлекает почти все преимущества её аналогов и частично закрывает недостатки.

