ГИД

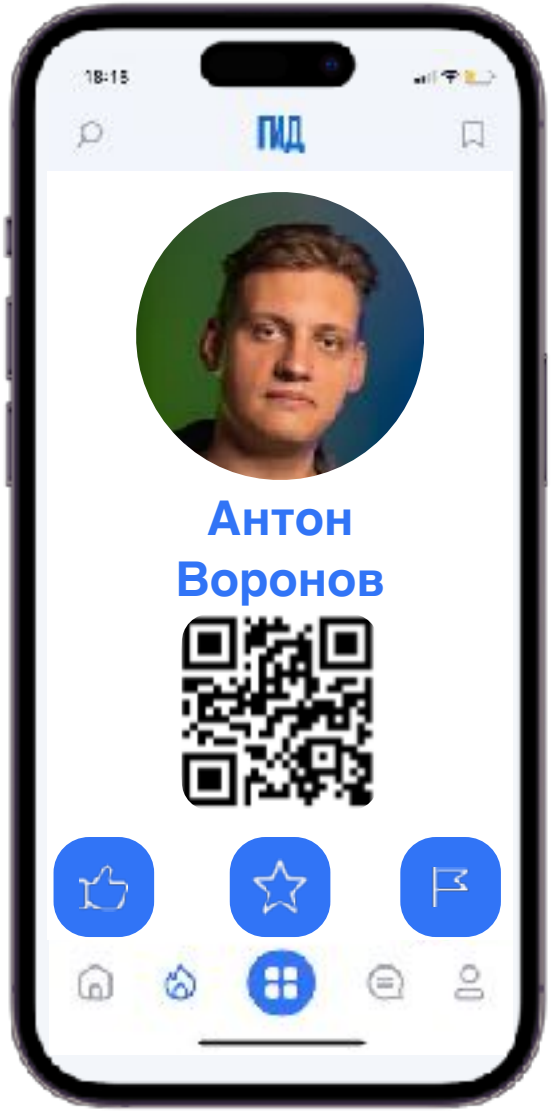
Тот самый ANN

Выбираем самую быструю и оптимальную базу для векторного поиска

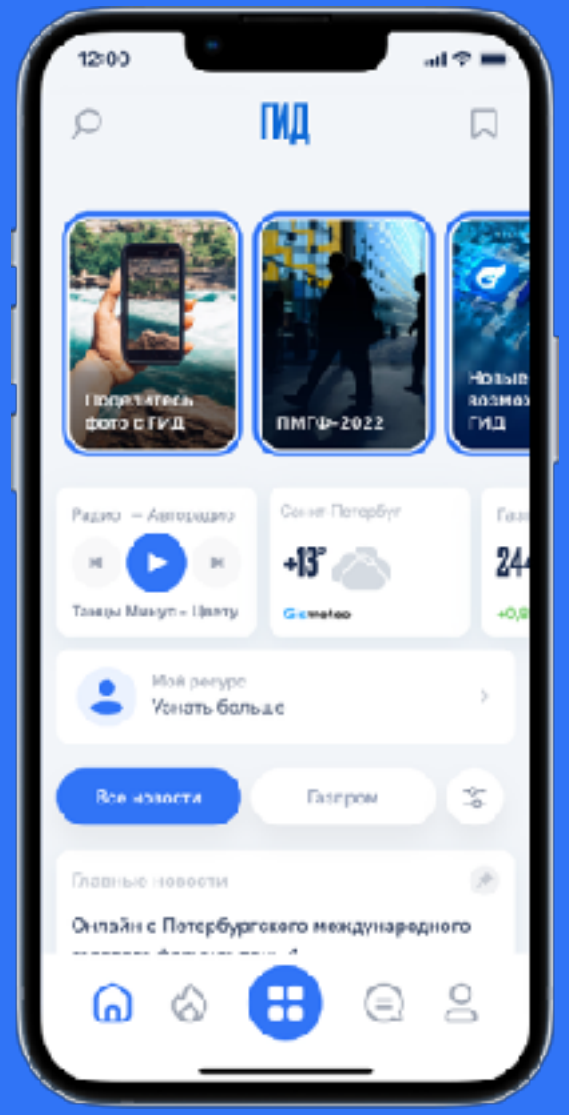
Спикер: Воронов Антон

Давайте знакомиться!

- ML Platform Lead @ Gazprom ID 🔥
- Do Data Science > 10 years 👨🏼‍💻
- 100k code lines/year 😱
- ODS.ai Track Lead 😎
- Live with 3 Maine Coon 🐈🐱🐈‍⬛ and still alive!
- NLP 1️⃣ ❤️‍🔥



Кто мы?



Мы — молодая компания ГИД и создаем продукты для внутреннего и внешних рынков

ГИД — корпоративная экосистема, формирующая цифровое пространство Группы Газпром

Газпром ID - универсальный идентификатор, позволяющий авторизовываться в различных сервисах и площадках

RecSys

Search&Suggest

О чем доклад

Что такое ANN?

Зачем ANN в поиске и рекомендациях?

Какой поиск нас устроит?

Что можно взять из готового?

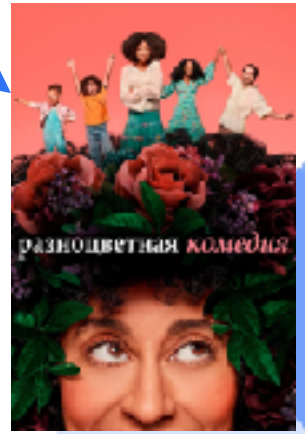
Как построили эксперимент?

Как все прошло?

Какие результаты?

Как обычно мы ищем?

Новые комедии



+TF-IDF

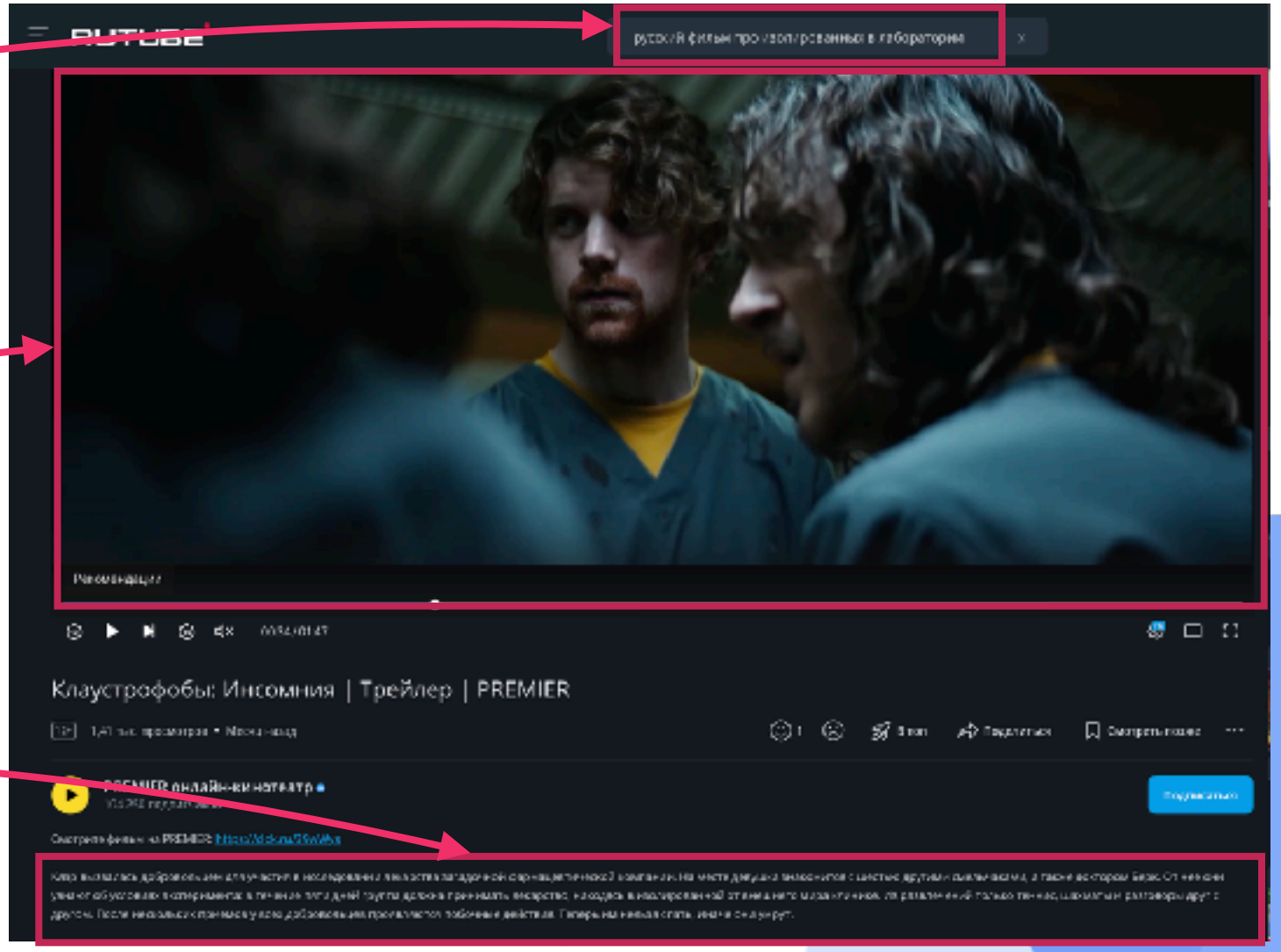
+ Rank

А что можно использовать в хорошем поиске?

Контекст запроса

Контент

Описание



Но что-то не так….

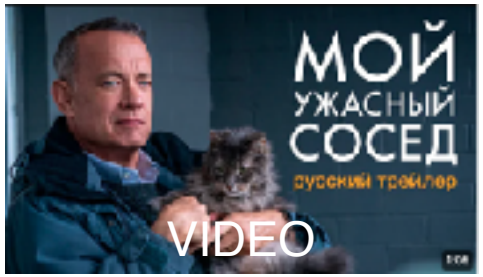
Содержание?

Контекст?

Описание?

Добавим немного магии…

Запрос

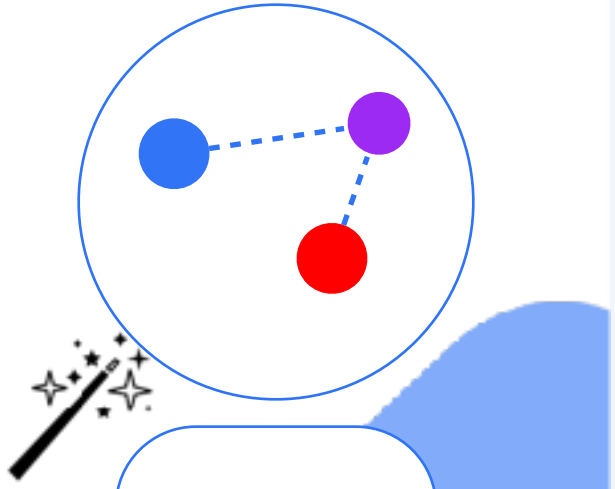


[0.123, 0.238…..0.234]

[0.123, 0.238…..0.234]

[0.123, 0.238…..0.234]

VECTOR

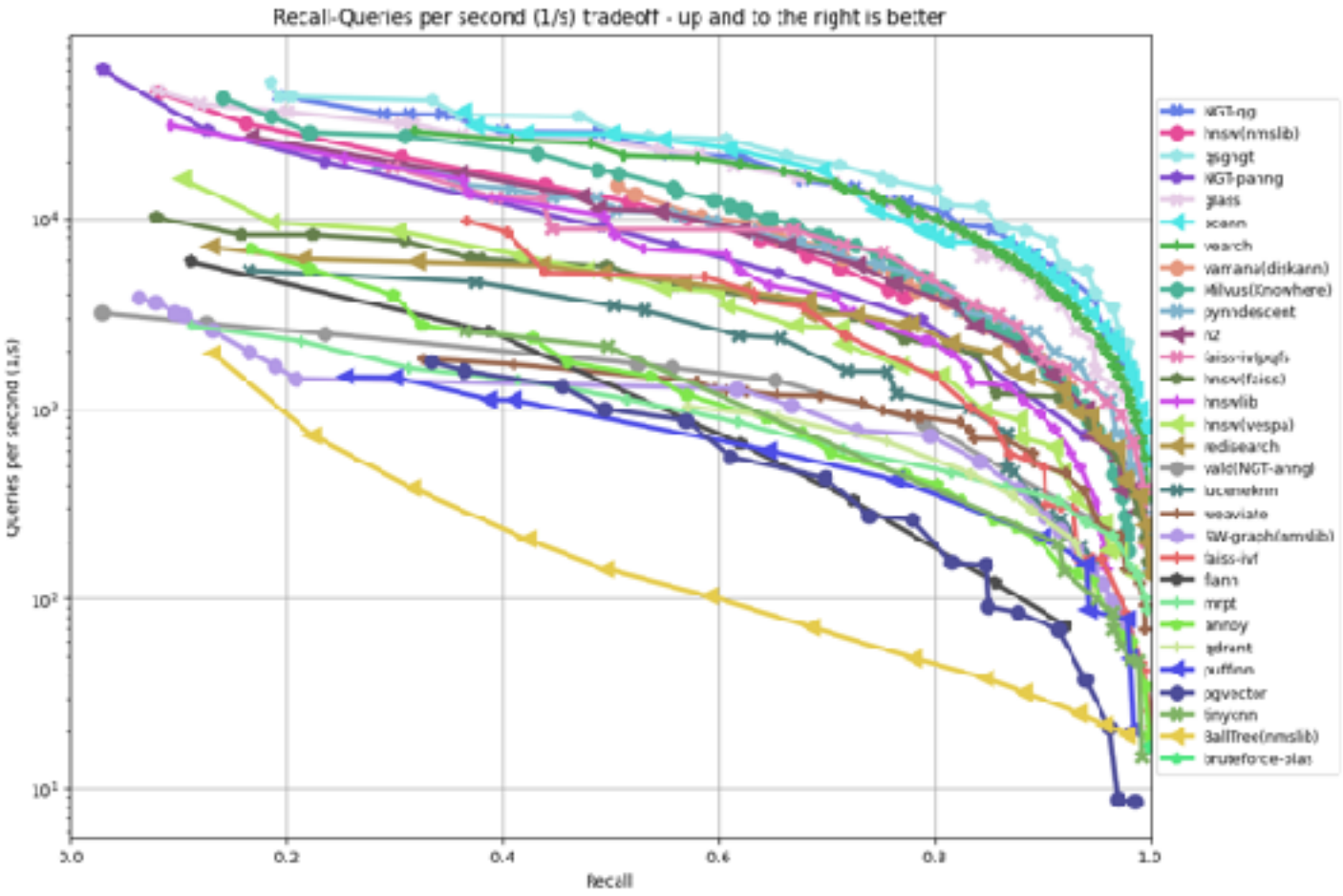


SPACE

Что это дает?

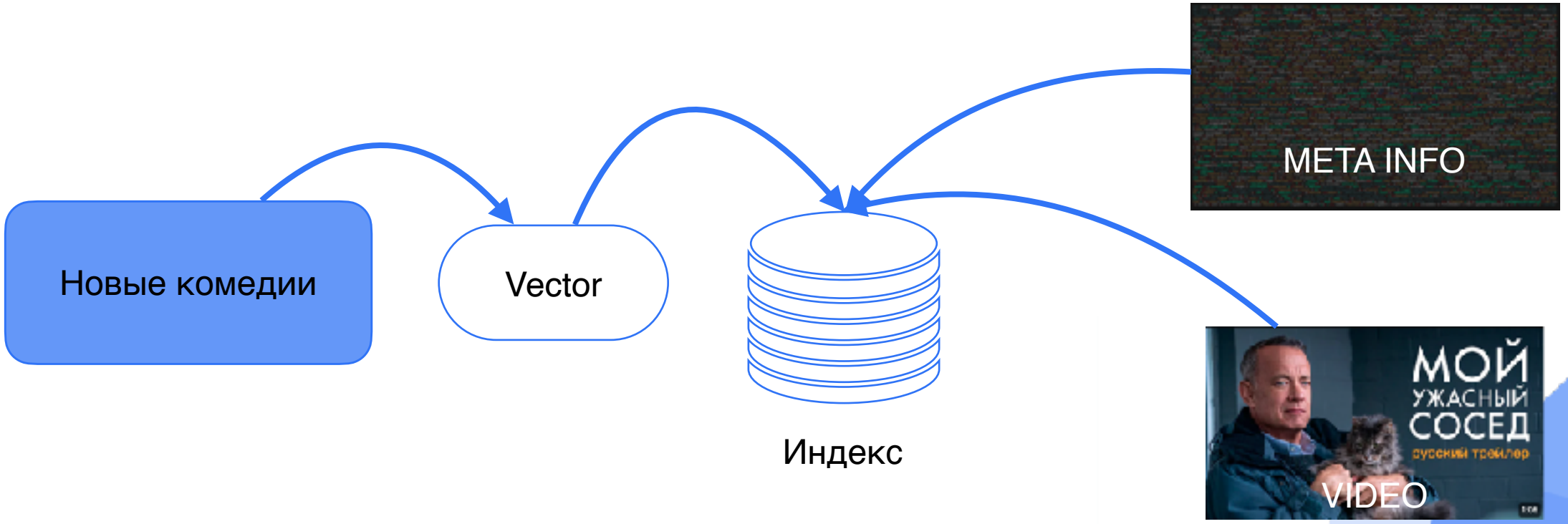
- Используем контекст запроса
- Более глубокое понимание описания контента
- Анализ самого контента
- Универсальный механизм поиска
- Быстро!

Действительно быстро….



Надо брать!

Как будем использовать в поиске?

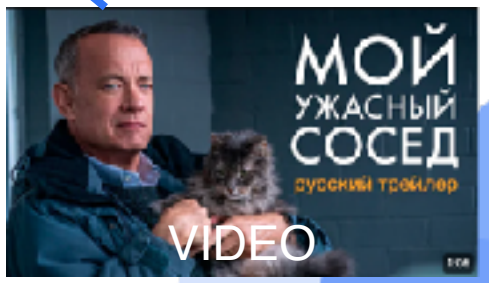


Может порекомендуем похожее видео?

Vector

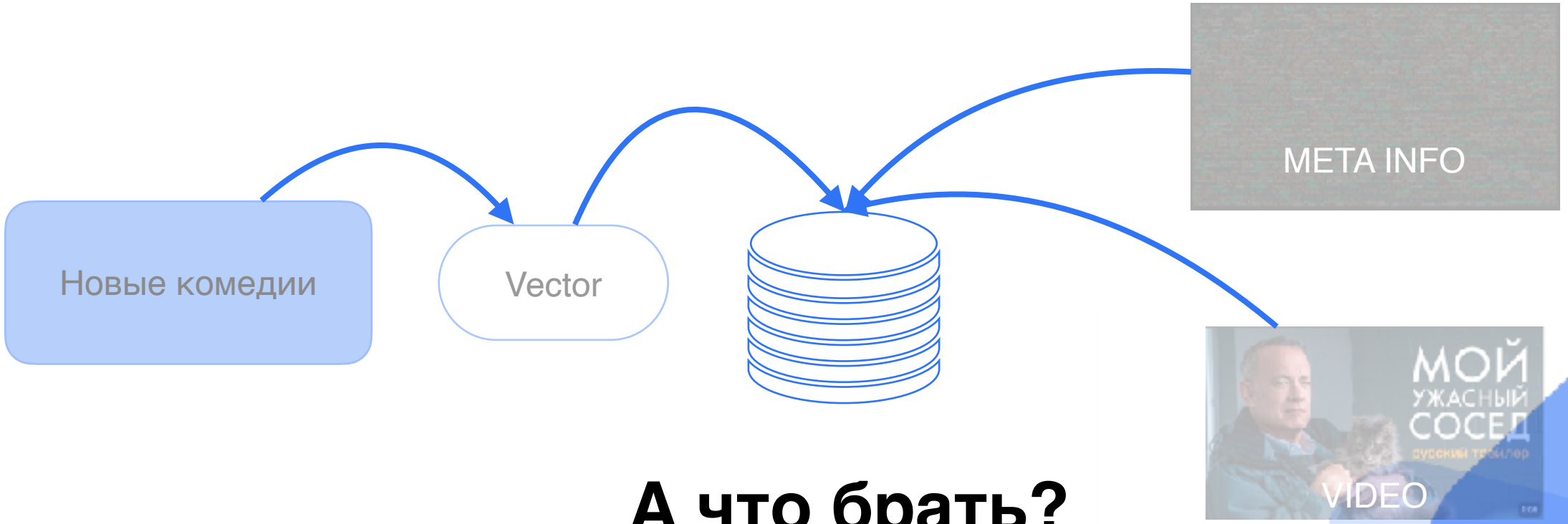
Индекс

META INFO



Надо точно брать!

Как будем использовать в поиске?

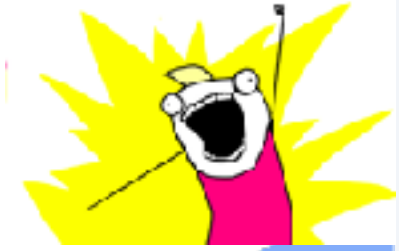


А что брать?

Есть план!

Какой план?

Базовый стек + Немного ”Стильно, модно, молодежно!” + Тест кейсы =



PostgreSQL

- Классическая реляционная база
- Есть плагин pgvector
- Поиск и фильтрация простым SQL запросом
- Есть L2, **Cosine**, dotProduct
- Движки **HNSW**, IVF-FLAT
- Сложно совместить TF-IDF и ANN

ClickHouse

- OLAP база данных с поддержкой SQL
- Векторный поиск через создание индекса
- Можно так же использовать быстрое агрегирование
- Есть L2, **Cosine**
- Есть движок **HNSW**
- Не создан для Real-Time трафика

OpenSearch

- Поисковый индекс
- Векторный поиск по настройке поля
- Можно совмещать TF-IDF, фильтрацию и ANN
- Есть L1, L2, **Cosine,** dotProduct
- Есть движок **HNSW**, FAISS, lucene и FLAT
- Можно запускать модели прямо в базе
- Движки не однообразны

Redis

- In-Memory Key-Value база данных
- Векторный поиск через создание индекса
- Есть L2, **Cosine,** dotProduct
- Есть движок **HNSW** и FLAT
- Очень не просто работать с кластерной версией

MongoDB

- Документо-ориентированная база данных
- Векторный поиск в версии Atlas
- Можно совмещать TF-IDF, фильтрацию и ANN
- Есть L1, L2, **Cosine,** dotProduct
- Есть движок **HNSW**

qDrant

- Векторная база данных
- Прямая интеграция к ML Inference
- Есть L1, L2, **Cosine,** dotProduct
- Есть движок **HNSW**
- Сложен в эксплуатации и тюнинге

Надо действовать!

Мантра выбора решения…

Просто обслуживать

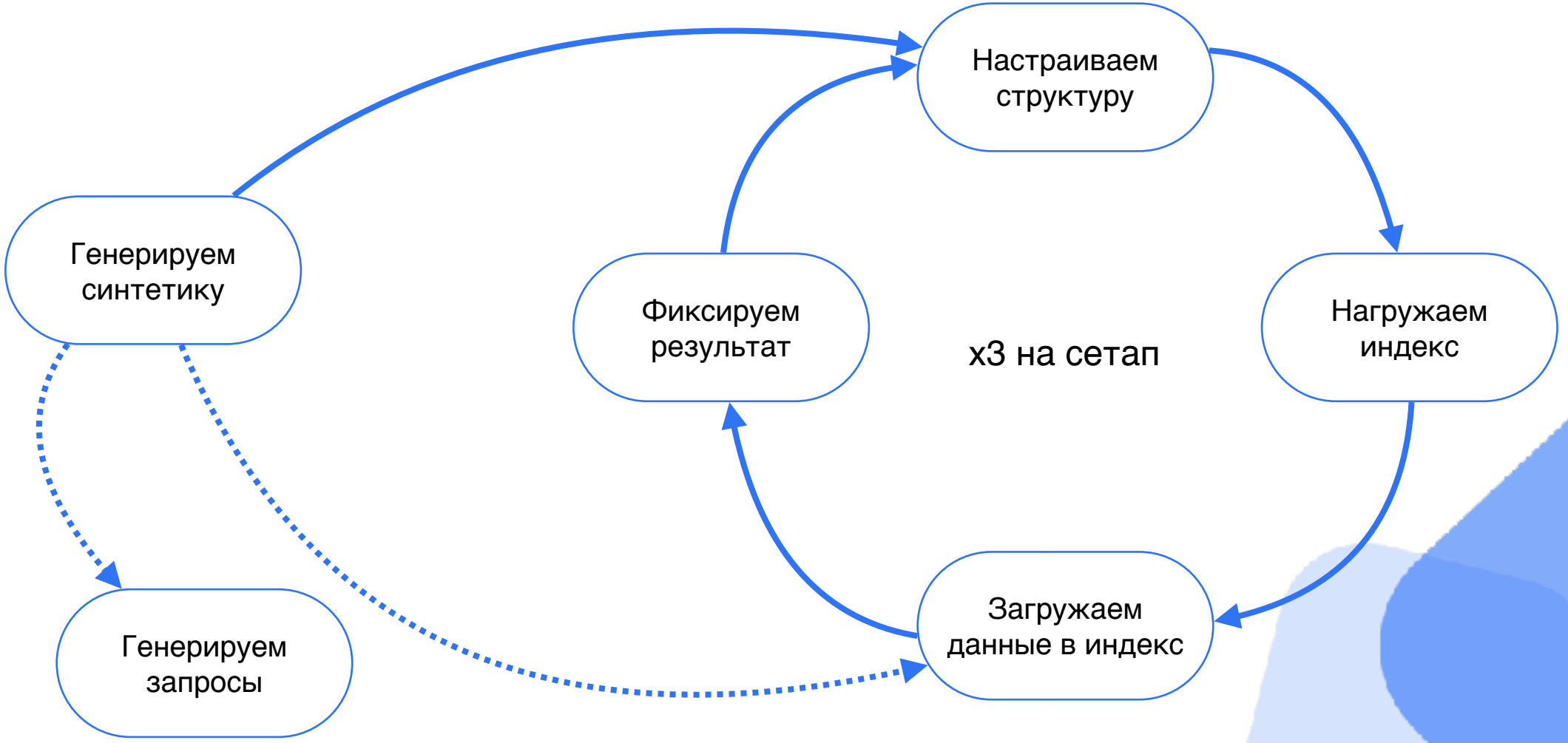
Удобно разрабатывать

Быстро искать

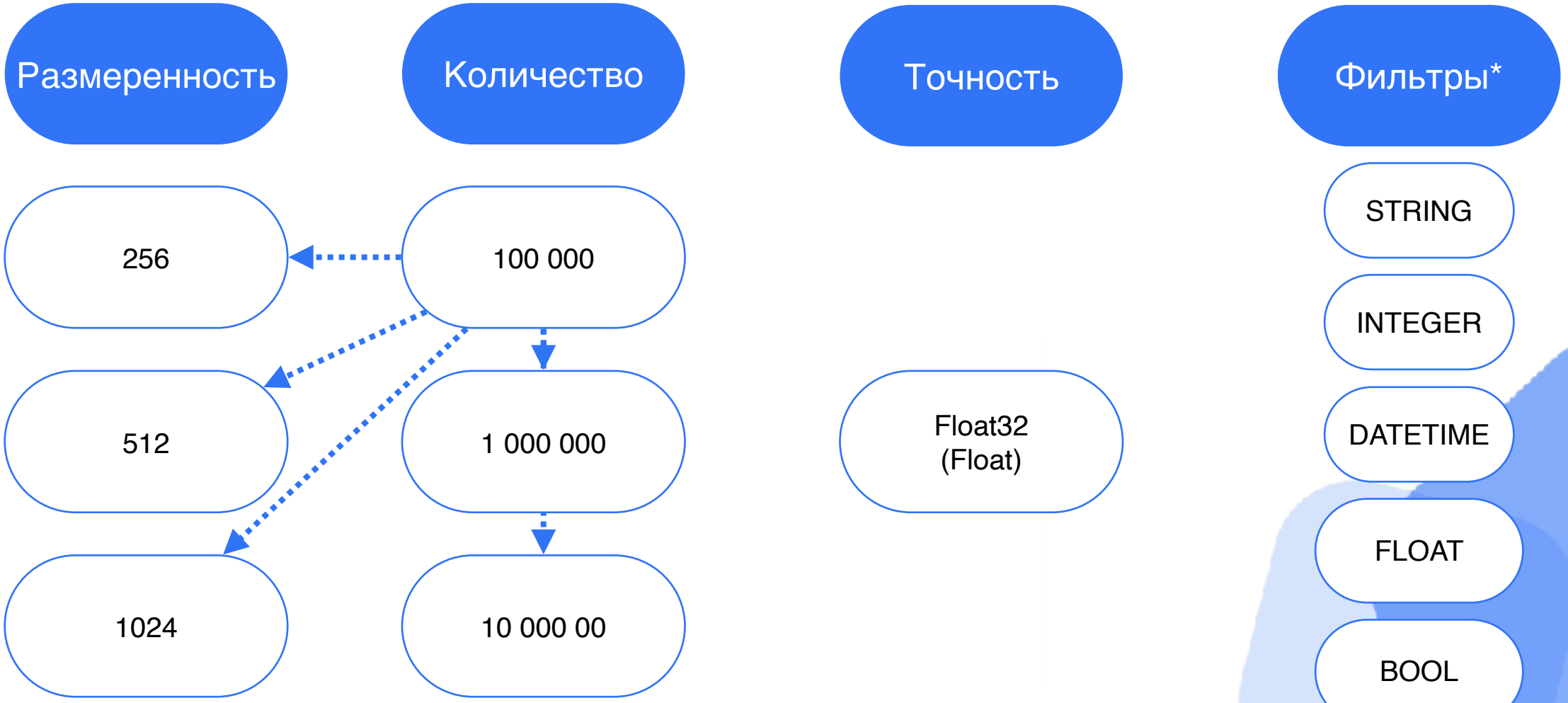
От мантры к критериям

	Просто обслуживать	Удобно разрабатывать	Быстро искать
PostgreSQL	Развертка	Документация	Размерность
ClickHouse			
OpenSearch	Масштабирование		
Redis			
MongoDB	Устойчивость	Совместимость	Количество
Qdrant			

Процесс тестирования



Что в индексе?

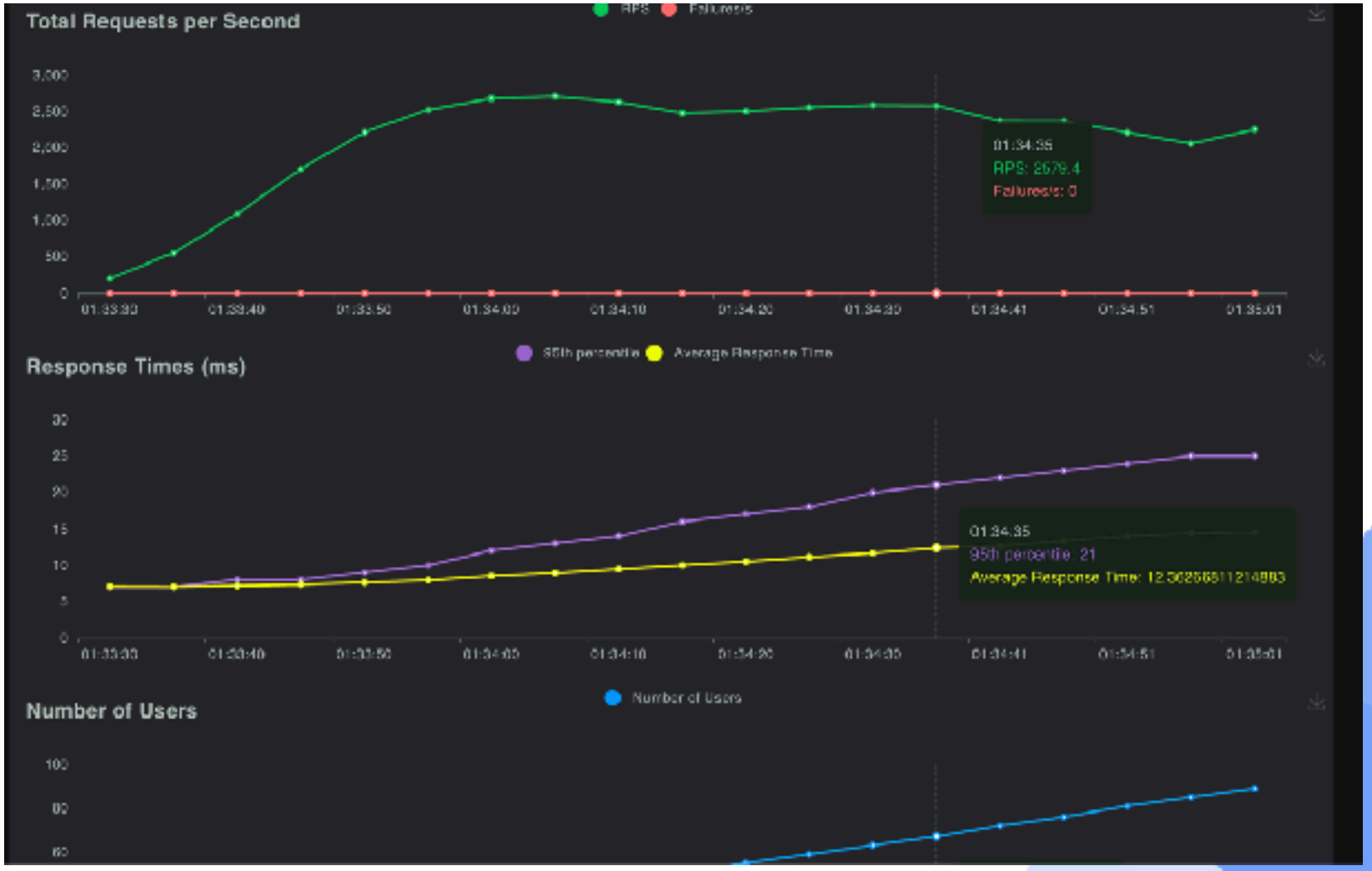


*Фильтры применяются на каждый запрос

Инструментарий

python™ + LOCUST +

Как выглядят репорты



NOT BAD

Пошли тестировать...

Начнем с документации

Документация | Совместимость

	Просто обслуживать	Удобно разрабатывать	Быстро искать
PostgreSQL	Развертка		Размерность
ClickHouse			
OpenSearch	Масштабирование		Количество
Redis			
MongoDB	Устойчивость		Точность
Qdrant			

Что с докой?

Leader Board

Документация

Совместимость

	Просто обслуживать	Удобно разрабатывать	Быстро искать
PostgreSQL		2	
ClickHouse		7	
OpenSearch		1	
Redis		5	
MongoDB		3	
Qdrant		4	

Размерность

Количество

Точность

Leader Board

Документация

Совместимость

	Просто обслуживать	Удобно разрабатывать		Быстро искать
PostgreSQL	Развертка	3	3	Размерность
ClickHouse		6	1	
OpenSearch	Масштабирование	1	4	Количество
Redis		5	6	
MongoDB	Устойчивость	4	5	Точность
Qdrant		2	2	

Python setup

- clickhouse_connect
- psycopg2, pgvector, sqlalchemy
- pymongo
- opensearchpy
- redis, redisvl
- qdrant-client

А что с инфрой?

Leader Board

Развертка | Масштабирование | Устойчивость | Документация | Совместимость

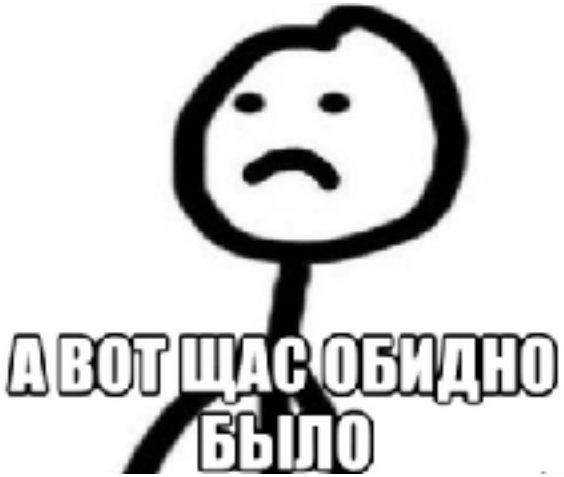
	Просто обслуживать			Удобно разрабатывать		Быстро искать
PostgreSQL				3	3	Размерность
ClickHouse				6	1	
OpenSearch				1	4	Количество
Redis				5	6	
MongoDB				4	5	Точность
Qdrant				2	2	

DevOps через неделю рисерча…

Выбывает из гонки

- Векторный поиск в версии Atlas и никак иначе

Mongo DB



Cluster setup

- 3 ноды для боевых запросов на разных хостах
- Все кандидаты развернуты через Helm Charts в Kubernetes
- Host с инструментом нагрузки на отдельной Node
- Каждой Node выделено 4 CPU

kubernetes

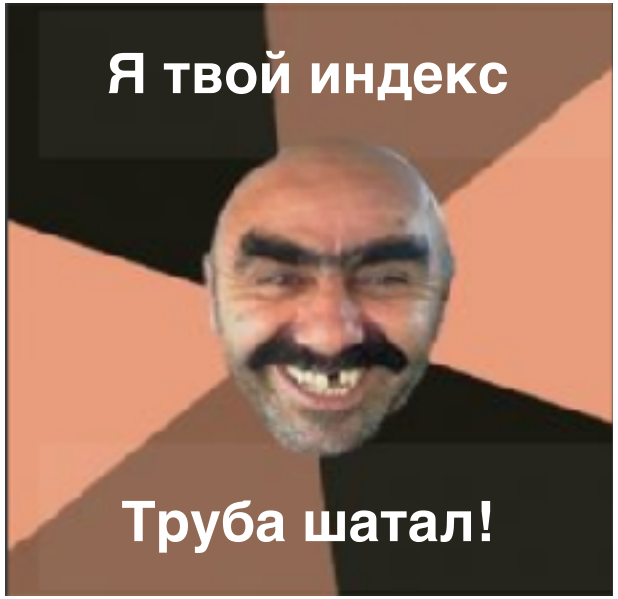
Leader Board

	Развертка	Масштабирование	Устойчивость	Документация	Совместимость	Количество	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3			3	3		
ClickHouse	4			6	1		
OpenSearch	2			1	4		
Redis	5			5	6		
~~MongoDB~~	~~6~~			~~4~~	~~5~~		
Qdrant	1			2	2		

На заметку

- В **Redis** и **qDrant** надо быть очень осторожными с настройками сетей, при локальном запуске
- **Redis Vector Search** доступен в кластере только в Redis Stack, альтернатива - Redis Replicas, что тоже добротно
- **Helm Charts** настраивать довольно быстро, по сравнению с **Compose**
- Лучше настроить все локально 1 раз, чтобы понять возможные проблемы при обслуживании

Давайте нагрузим!



Хотелки

< 200ms

95pp

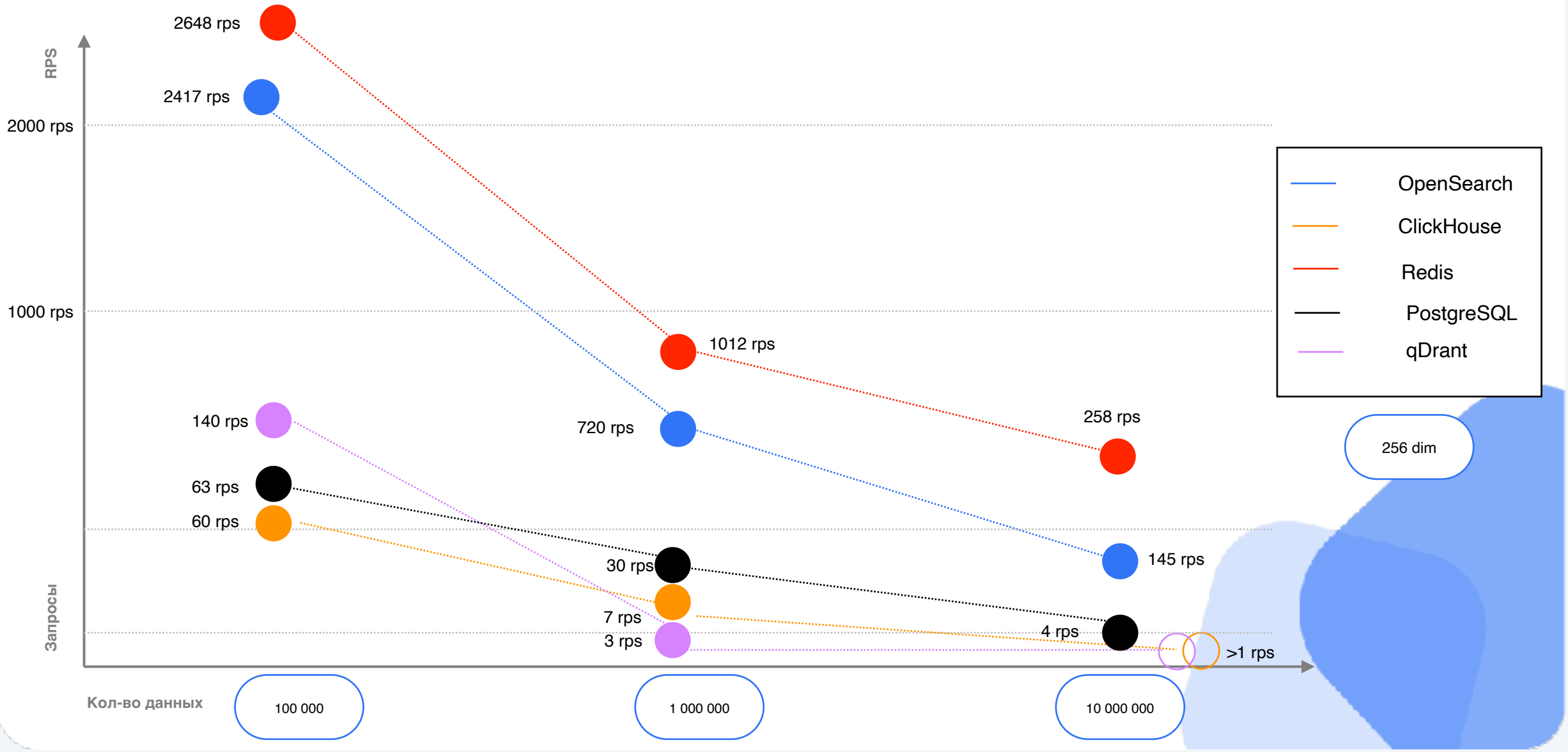
Post фильтрация

Leader Board

	Развертка	Масштабирование	Устойчивость	Документация	Совместимость	**Количество**	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3			3	3		
ClickHouse	4			6	1		
OpenSearch	2			1	4		
Redis	5			5	6		
~~MongoDB~~	~~6~~			~~4~~	~~5~~		
Qdrant	1			2	2		

Тестируем

Больше данных



Leader Board

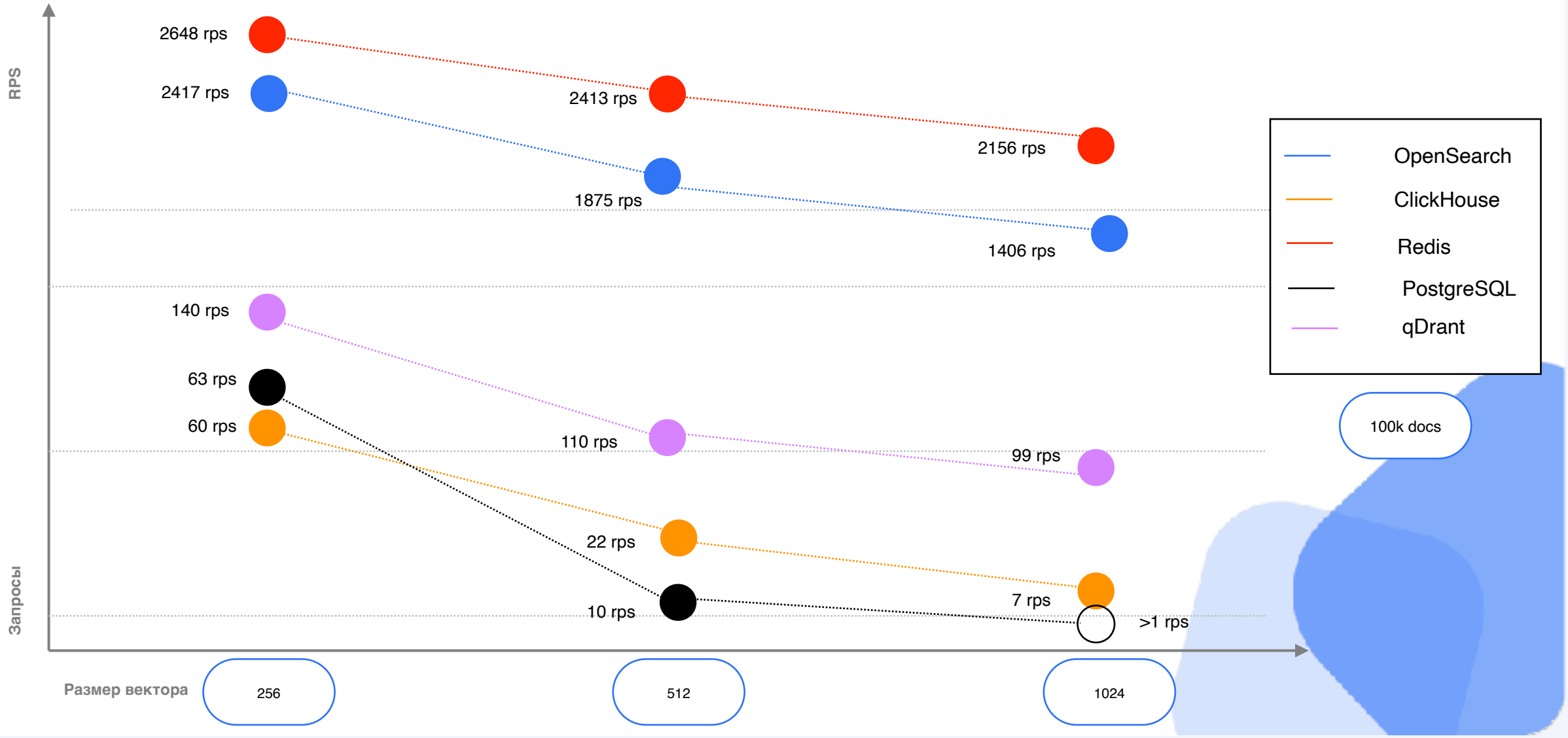
	Развертка	Масштабирование	Устойчивость	Документация	Совместимость	**Количество**	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3					3	
ClickHouse	4					5	
OpenSearch	2					2	
Redis	5					1	
~~MongoDB~~	~~6~~						
Qdrant	1					4	

Изменим размерность

Leader Board

	Развертка	Масштабирование	Устойчивость	Документация	Совместимость	Количество	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3			3	3	3	
ClickHouse	4			6	1	5	
OpenSearch	2			1	4	2	
Redis	5			5	6	1	
~~MongoDB~~	~~6~~			~~4~~	~~5~~		
Qdrant	1			2	2	4	

Больше, лучше, медленнее…



Leader Board

	Развертка	Масштабирование	Устойчивость	Документация	Совместимость	Количество	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3			3	3	3	5
ClickHouse	4			6	1	5	4
OpenSearch	2			1	4	2	2
Redis	5			5	6	1	1
~~MongoDB~~	~~6~~			~~4~~	~~5~~		
Qdrant	1			2	2	4	3

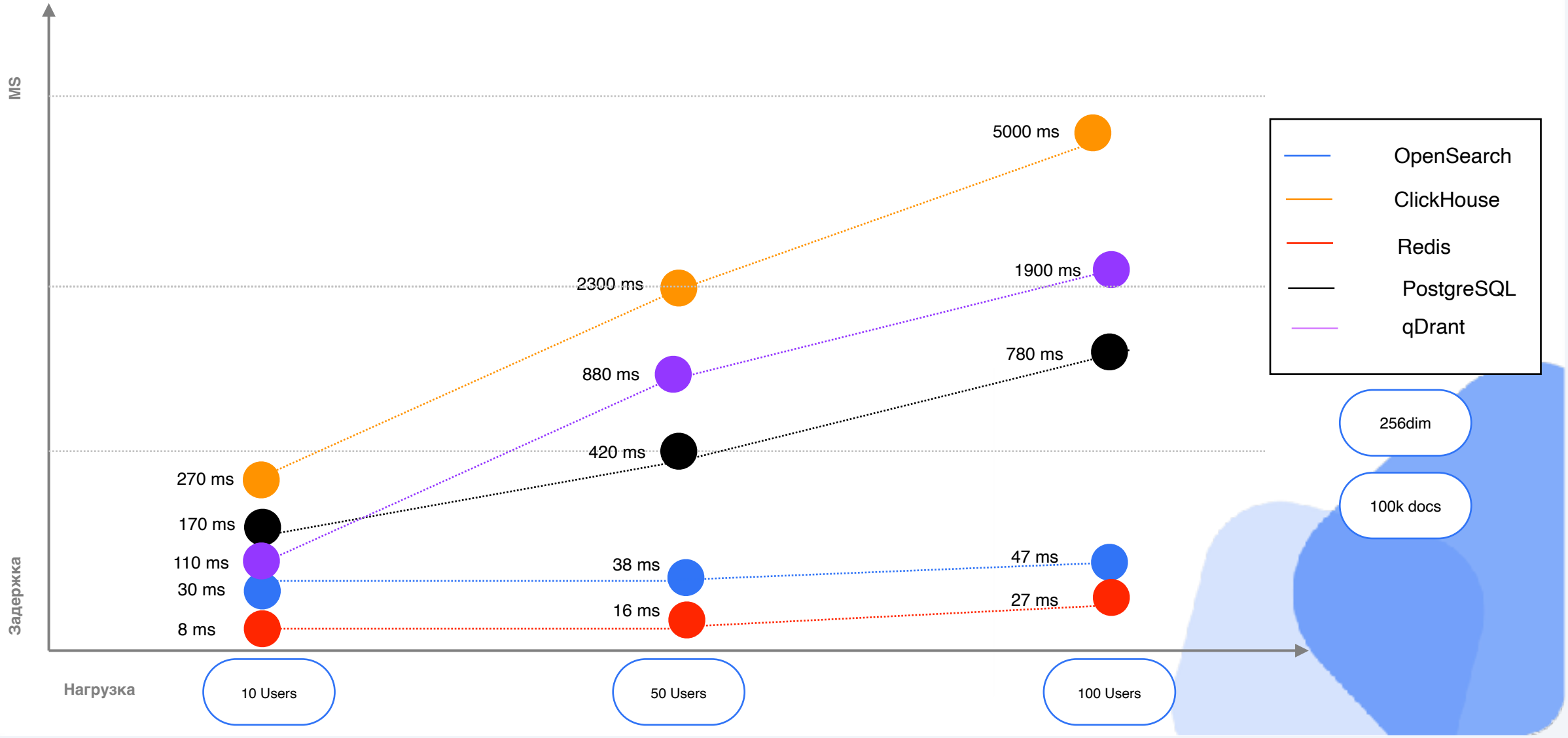
А если overload?

Leader Board

	Развертка	Масштабирование	**Устойчивость**	Документация	Совместимость	Количество	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3			3	3	3	5
ClickHouse	4			6	1	5	4
OpenSearch	2			1	4	2	2
Redis	5			5	6	1	1
~~MongoDB~~	~~6~~			~~4~~	~~5~~		
Qdrant	1			2	2	4	3

Тестируем

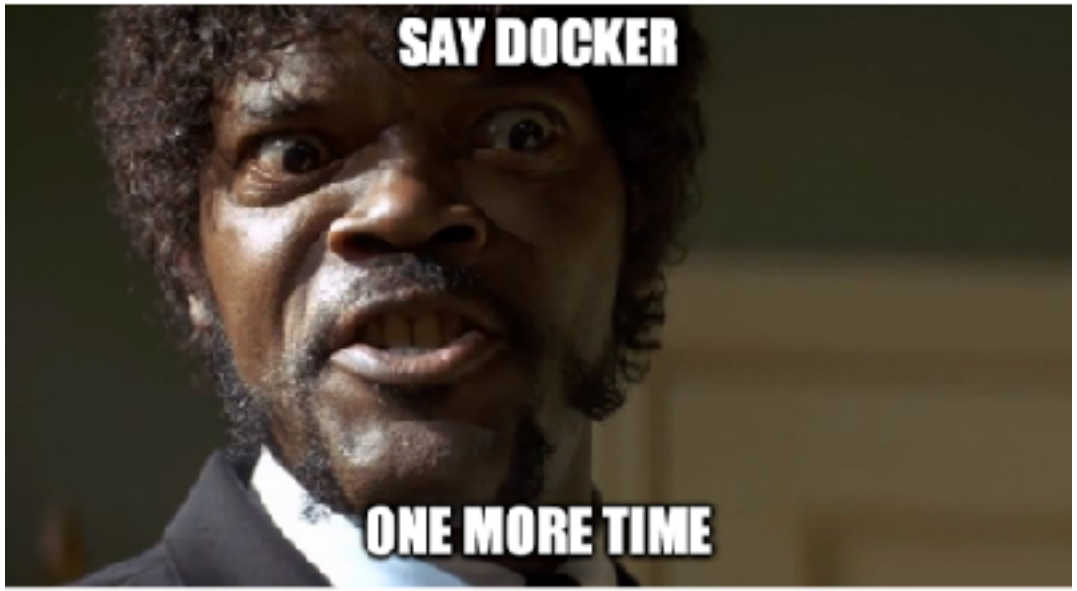
Нагружаем в бесконечность



Leader Board

	Развертка	Масштабирование	**Устойчивость**	Документация	Совместимость	Количество	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3		3	3	3	3	5
ClickHouse	4		5	6	1	5	4
OpenSearch	2		2	1	4	2	2
Redis	5		1	5	6	1	1
~~**MongoDB**~~	~~6~~			~~4~~	~~5~~		
Qdrant	1		4	2	2	4	3

А как с масштабированием?



Leader Board

	Развертка	Масштабирование	Устойчивость	Документация	Совместимость	Количество	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3		3	3	3	3	5
ClickHouse	4		5	6	1	5	4
OpenSearch	2		2	1	4	2	2
Redis	5		1	5	6	1	1
MongoDB	6			4	5		
Qdrant	1		4	2	2	4	3

На заметку

- Для **ClickHouse** нужно пересоздать таблицы + перезалить данные
- Для данных >100k лучше наливать батчами
- **OpenSearch** - достаточно добавить реплик в настройке индекса
- **qDrand**- нужно перезалить данные или перераспределить *shards* руками
- Для **PostgreSQL** - можно просто подключить реплику
- В **Redis** - достаточно добавить реплик

Leader Board

	Развертка	Масштабирование	Устойчивость	Документация	Совместимость	Количество	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3	3	3	3	3	3	5
ClickHouse	4	5	5	6	1	5	4
OpenSearch	2	1	2	1	4	2	2
Redis	5	2	1	5	6	1	1
~~MongoDB~~	~~6~~			~~4~~	~~5~~		
Milvus	1	4	4	2	2	4	3

Подведем итоги

Подведем результаты

	Развертка	Масштабирование	Устойчивость	Документация	Совместимость	Количество	Размерность
	Просто обслуживать			Удобно разрабатывать		Быстро искать	
PostgreSQL	3	3	3	3	3	3	5
ClickHouse	4	5	5	6	1	5	4
OpenSearch	2	1	2	1	4	2	2
Redis	5	2	1	5	6	1	1
~~MongoDB~~	~~6~~			~~4~~	~~5~~		
Qdrant	1	4	4	2	2	4	3
	OpenSearch, redis			OpenSearch, Qdrant		Qdrant, redis, OpenSearch	

Что выбрали для себя?

Online

Search

RecSys

OpenSearch

Text

Non-Text

- Эффективно совмещать поиски
- Быстро масштабировать

Near-Real Time

RecSys

OpenSearch

- Выть уверенным в случае highload
- Работать с большим кол-вом данных

Что дальше?

P.S.

Что еще будет в тестах

- Еще базы для тестов
- ML Inference on DB
- Тесты скорости индексации
- Данные из Benchmarks
- Публикация репозитория benchmark

Weaviate

Dragonfly

Milvus

cassandra

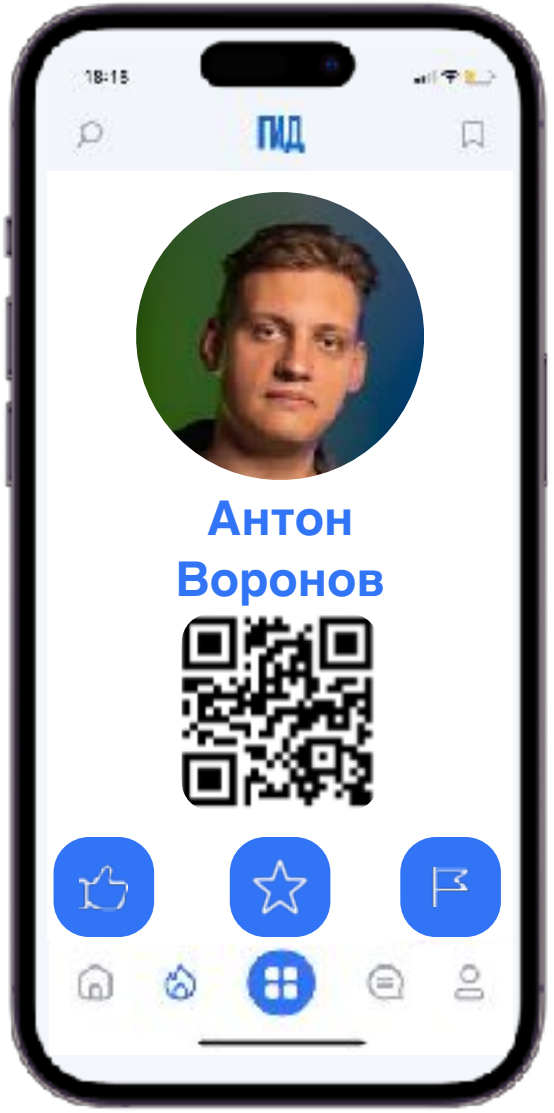
Огромное спасибо героям!

Кожарин Алексей
Hard-Core Developer

Виктор Казаченко
Hard-Core Developer

Спасибо!

- TG: voronov_ad
- Mail: advoronov@gid.ru
- Если интересны примеры кода - пишите, расскажем)



ГИД

Ваши вопросы приветствуются!