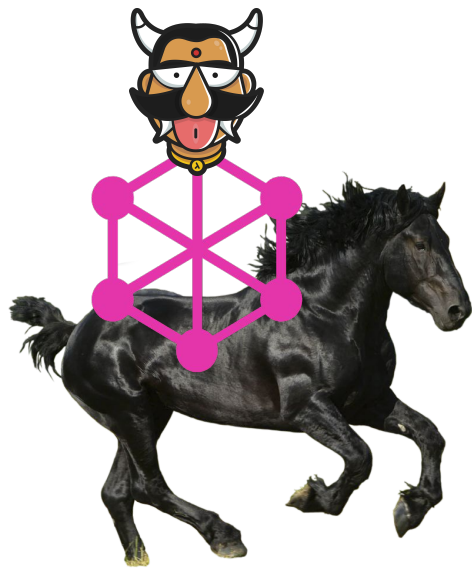


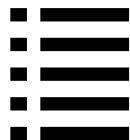
GraphQL

ВСАДНИК СИНГУЛЯРНОСТИ



Царегородцев Вадим, [Infobip](#)
Telegram: [@hadoocken](#)

Once upon a time...



$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



Once upon a time...



?



Singularity

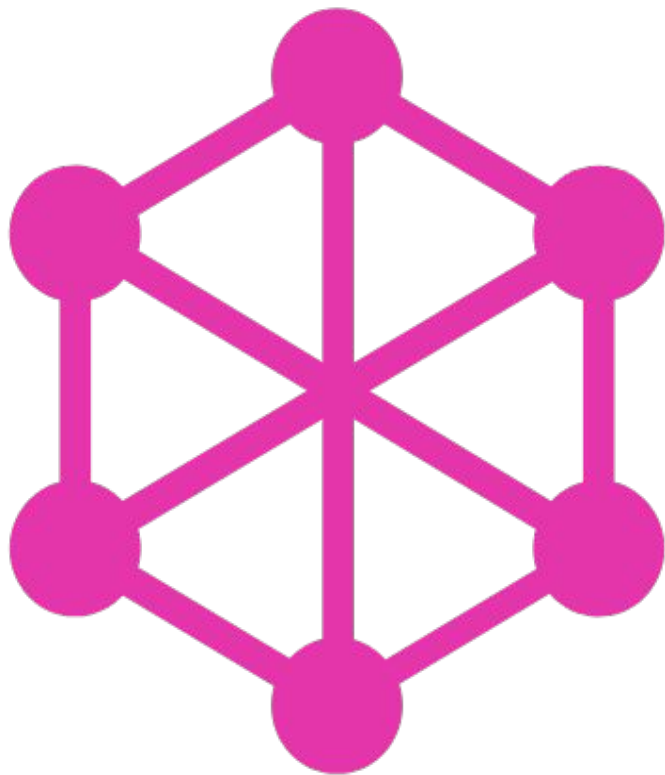
Писать машинный код

Писать код
на высокоуровневом языке

Разговаривать с колонкой



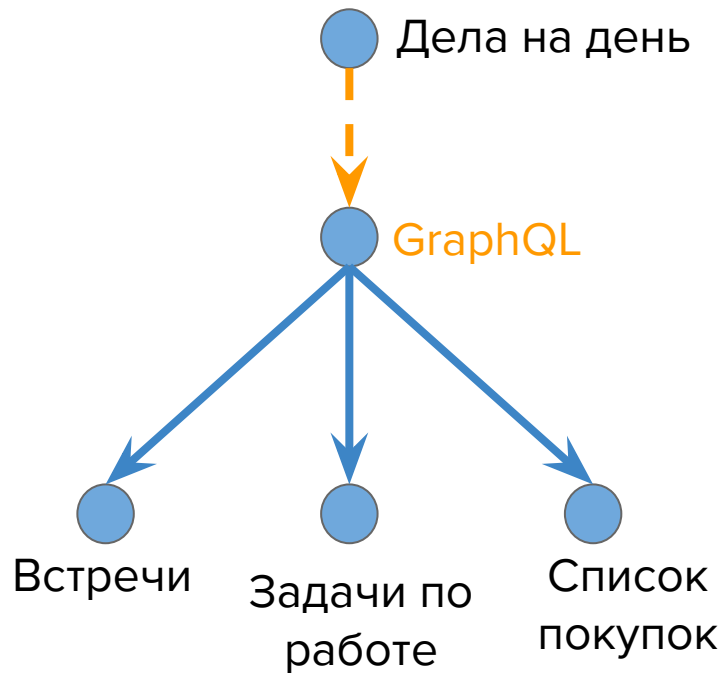
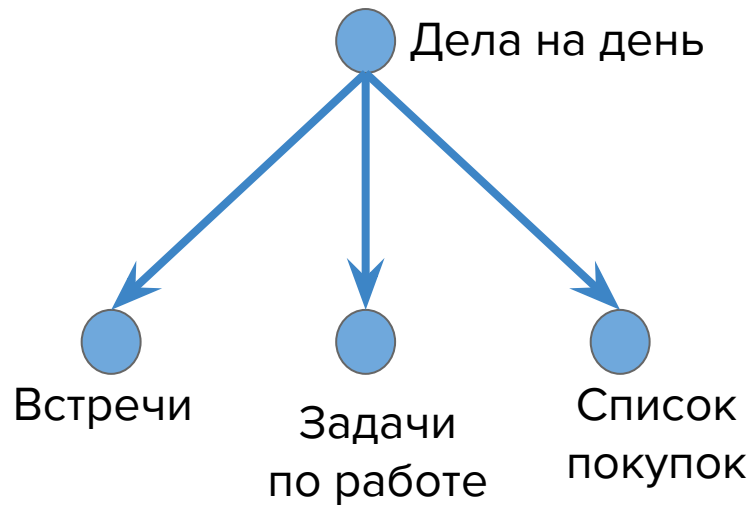
GraphQL

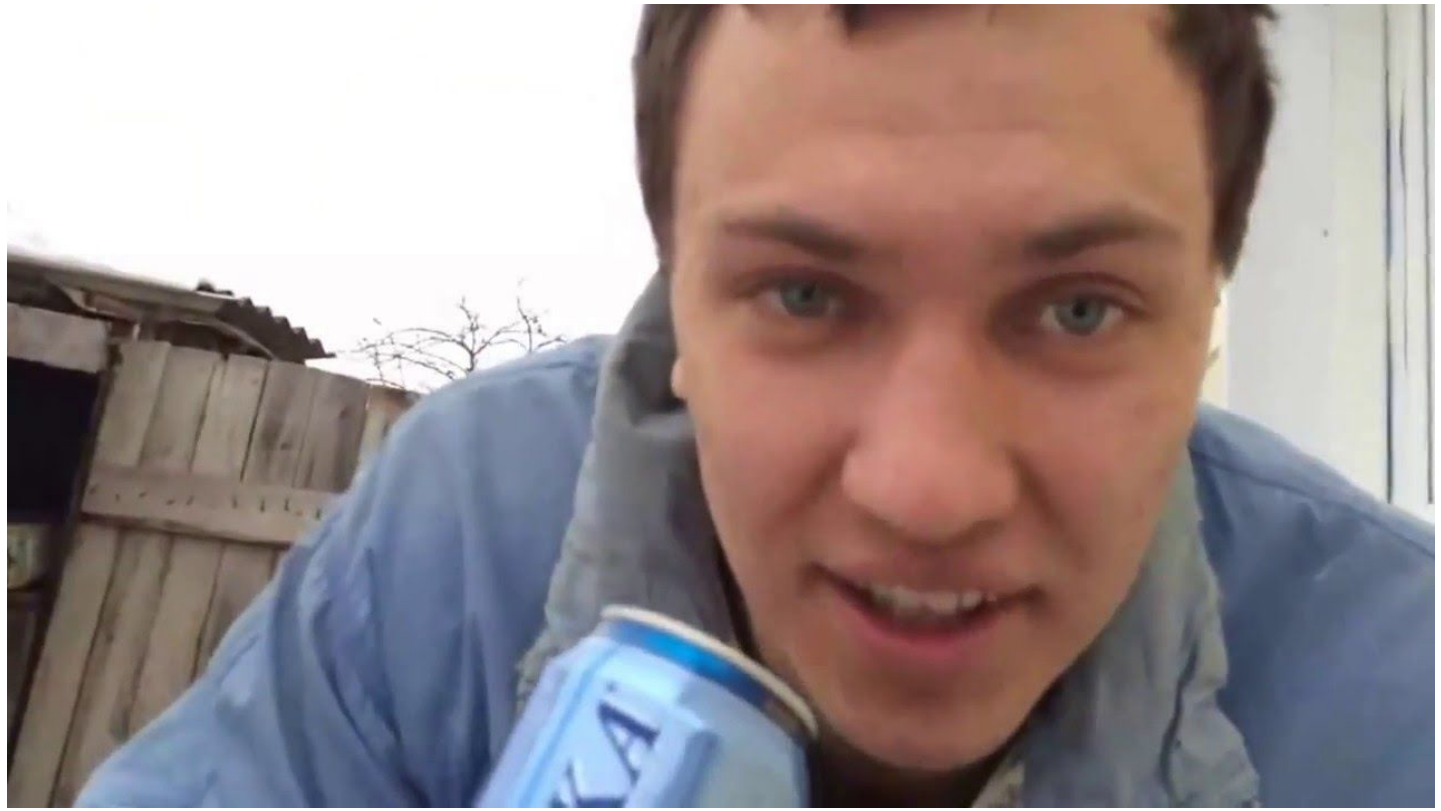


Чего не будет в докладе?

- Глубокого анализа GraphQL
- Мотивации создания и плюсы/минусы использования
- Сравнения Apollo и Relay

Пара слов про GraphQL





Пара слов про GraphQL

```
1 query {  
2   todos {  
3     meetings {  
4       date  
5     }  
6     products {  
7       caption  
8     }  
9     tasks {  
10      title  
11      estimation  
12    }  
13  }  
14 }  
15
```

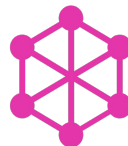
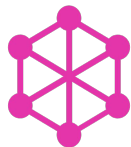


```
1 mutation IgnoreMeeting($mId: Int!) {  
2   ignore(meetingId: $mId) {  
3     todos {  
4       meetings  
5     }  
6   }  
7 }
```

FRONTEND

FRONTEND

OR?



BACKEND

BACKEND

Elementary, my dear Watson.

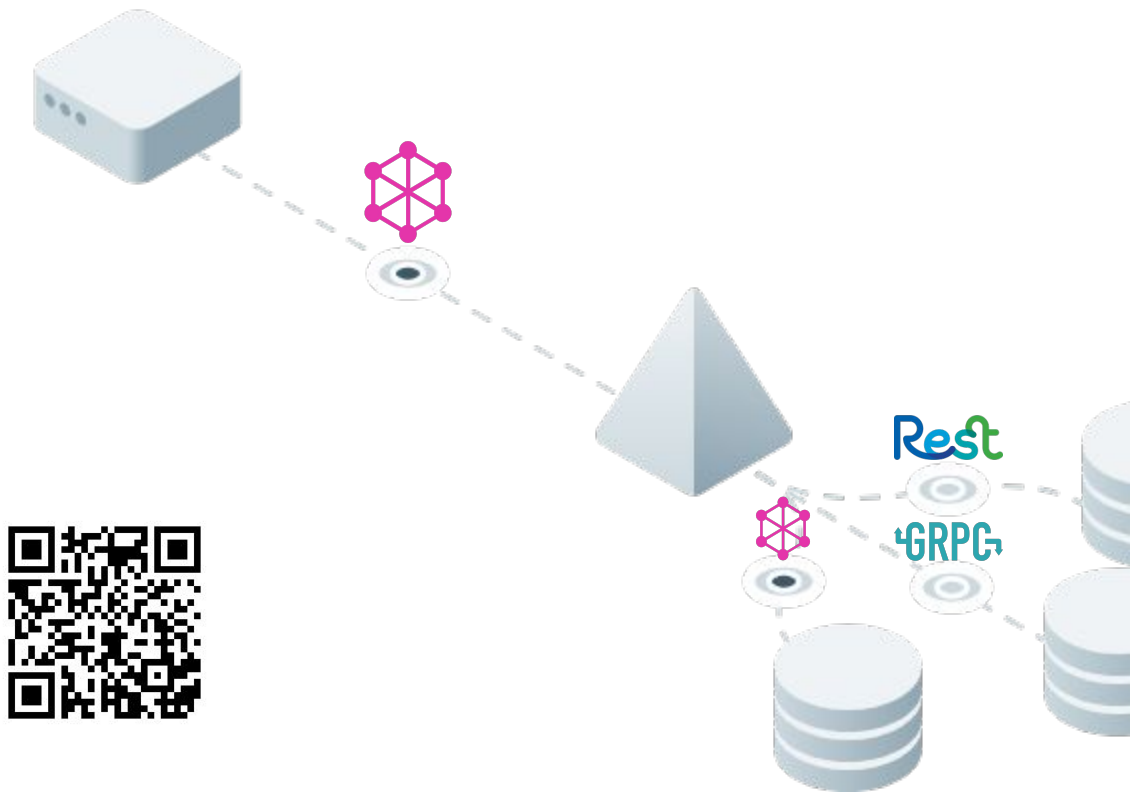


Сделать запрос на сервер, чтобы тот
взял данные из БД и отдал на фронт



Брать данные из БД напрямую

Prisma



Hasura

- **GraphQL Engine** в **Docker** контейнере поверх **postgres** (Работает везде, где работает **Docker**)
- Своя консоль разработчика из коробки
- Решения для **NoSql** баз данных (**Firebase**, **Json**)
- Настройка **postgres proxy** для других **Sql** баз данных

Hasura In Action

Quickstart

1. Deploy to Heroku

Deploy to Heroku and instantly get a realtime GraphQL API backed by Heroku Postgres:

 Deploy to Heroku



<https://github.com/hasura/graphql-engine-heroku>

Usage comparison - Firebase SDK vs GraphQL

A typical query to do a single read from the database using [Firebase SDK](#), (javascript) would look something like:

```
firebase.database().ref('/users/' + userId).once('value').then(function(snapshot) {  
  var username = (snapshot.val() && snapshot.val().username) || 'Anonymous';  
  // ...  
});
```

Equivalent GraphQL Query would look like:

```
query {  
  users(where: {uid: {_eq: userId}}) {  
    uid,  
    username  
  }  
}
```



Спасибо.

Telegram: [@hadoocken](https://t.me/hadoocken)