

GOTO **AMSTERDAM 2023**

#GOTOams



GOTO
Guide

LET US HELP YOU

Ask questions
through **the app**



also remember to rate session



Download on the
App Store



GET IT ON
Google Play

THANK YOU!

#GOTOams

GraphQL Mesh

Query Anything, Run Anywhere



Uri Goldshtein
[@urigoldshtein](#)

AGENDA

- Intro to GraphQL
- GraphQL and the existing API landscape
- Adopt GraphQL without migration
- Use “GraphQL Power-tools” in your existing APIs





THE-GUILD.DEV



Generate your code for
any technology to GraphQL



THE GUILD



Tooling for GraphQL to
Compare schemas & Validate docs.



THE GUILD



Enterprise Grade Tooling
For Your GraphQL Server



THE GUILD



A set of utilities for faster
development of GraphQL Schemas



THE GUILD



**THE
GUILD**®



INSPECTOR

GRAPHQL



GENERATOR

GRAPHQL.CODE



CLI

GRAPHQL



SCALARS

GRAPHQL



MESH

GRAPHQL



MODULES

GRAPHQL



TOOLKIT

GRAPHQL



SOFA

GRAPHQL



MERGE

GRRAPHQL



IMPORT

GRAPHQL



LOONA

GRAPHQL



CONFIG

GRAPHQL



ANGULAR

APOLLO GRAPHQL



CONNECTED

BUILD



WHATSAPP

CLONE TUTORIAL



STENCIL

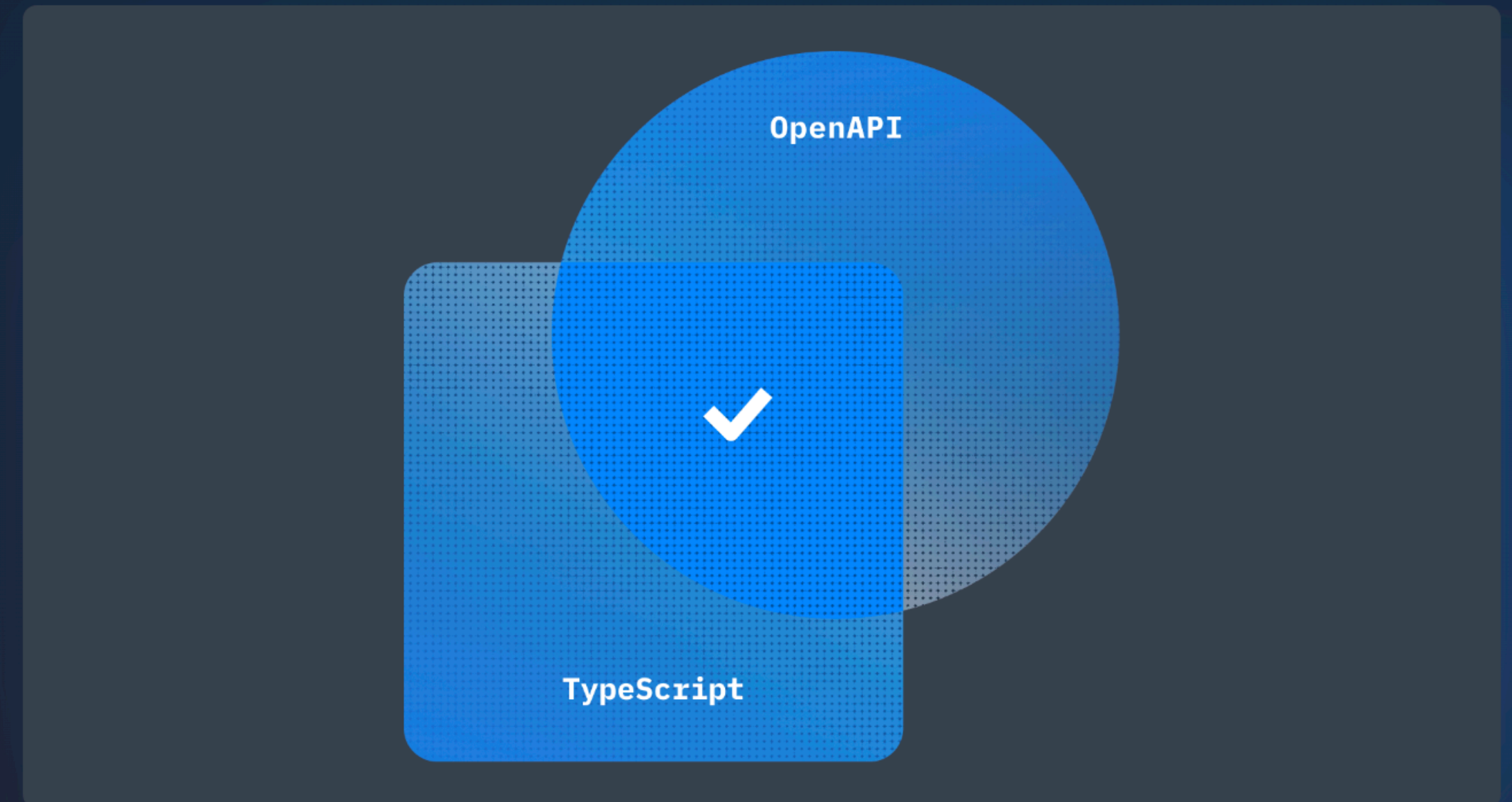
APOLLO



feTS

Fetch API ❤️ TypeScript

Build and consume **REST APIs** with ease. No more compromises on type safety in client-server communication. All thanks to **TypeScript** and **OpenAPI**.

[Client](#)[Server](#)[GitHub](#)

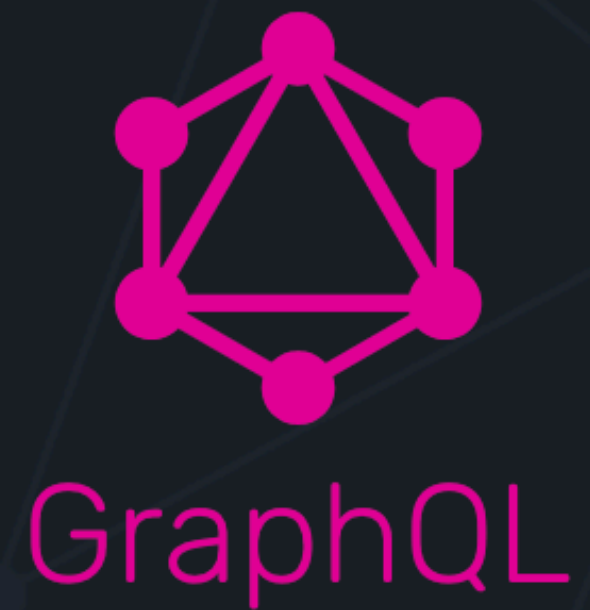
New Approach to REST APIs

GraphQL Meetup with Booking.com and Albert Heijn



Hosted By
Uri G. and GraphQL F.





Describe your data

```
type Project {  
  name: String  
  tagline: String  
  contributors: [User]  
}
```

Ask for what you want

```
{  
  project(name: "GraphQL") {  
    tagline  
  }  
}
```

Get predictable results

```
{  
  "project": {  
    "tagline": "A query language for APIs"  
  }  
}
```


Extra roundtrips

C L I E N T

S E R V E R

GET user/chats



GET user/chats/1/messages



GET user/chats/2/messages



GET user/chats/3/messages



```
{
  "id": 67563416,
  "name": "android-docs",
  "full_name": "apollostack/android-docs",
  "owner": {
    "login": "apollostack",
    "id": 17189275,
    "avatar_url": "https://avatars.githubusercontent.com/u/17189275?v=3",
    "gravatar_id": "",
    "url": "https://api.github.com/users/apollostack",
    "html_url": "https://github.com/apollostack",
    "followers_url": "https://api.github.com/users/apollostack/followers",
    "following_url": "https://api.github.com/users/apollostack/following{/other_user}",
    "gists_url": "https://api.github.com/users/apollostack/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/apollostack/starred{/owner}/{/repo}",
    "subscriptions_url": "https://api.github.com/users/apollostack/subscriptions",
    "organizations_url": "https://api.github.com/users/apollostack/orgs",
    "repos_url": "https://api.github.com/users/apollostack/repos",
    "events_url": "https://api.github.com/users/apollostack/events{/privacy}",
    "received_events_url": "https://api.github.com/users/apollostack/received_events",
    "type": "Organization",
    "site_admin": false
  },
  "private": false,
  "html_url": "https://github.com/apollostack/android-docs",
  "description": "[WIP] :book: Docs for using Apollo with Android",
  "fork": false,
  "url": "https://api.github.com/repos/apollostack/android-docs",
  "forks_url": "https://api.github.com/repos/apollostack/android-docs/forks",
  "keys_url": "https://api.github.com/repos/apollostack/android-docs/keys{/key_id}",
  "collaborators_url": "https://api.github.com/repos/apollostack/android-docs/collaborators{/collaborator}",
  "teams_url": "https://api.github.com/repos/apollostack/android-docs/teams",
  "hooks_url": "https://api.github.com/repos/apollostack/android-docs/hooks",
```

```
{{"login":"stubailo","id":448783,"avatar_url":"https://avatars.githubusercontent.com/u/448783?v=3","gravatar_id":"","url":"https://api.github.com/users/stubailo","html_url":"https://github.com/stubailo","followers_url":"https://api.github.com/users/stubailo/followers","following_url":"https://api.github.com/
```

```
{{"login":"stubailo","id":448783,"avatar_url":"https://avatars.githubusercontent.com/u/448783?v=3","gravatar_id":"","url":"https://api.github.com/users/stubailo","html_url":"https://github.com/stubailo","followers_url":"https://api.github.com/users/stubailo/followers","following_url":"https://api.github.com/
```

```
{{"login":"stubailo","id":448783,"avatar_url":"https://avatars.githubusercontent.com/u/448783?v=3","gravatar_id":"","url":"https://api.github.com/users/stubailo","html_url":"https://github.com/stubailo","followers_url":"https://api.github.com/users/stubailo/followers","following_url":"https://api.github.com/
```


Overfetching

C L I E N T

S E R V E R

GET user/chats



```
[
  {
    "id": 67563416,
    "name": "android-docs",
    "full_name": "apollostack/android-docs",
    "owner": {
      "login": "apollostack",
      "id": 17189275,
      "avatar_url": "https://avatars.githubusercontent.com/u/17189275?v=3",
      "gravatar_id": "",
      "url": "https://api.github.com/users/apollostack",
      "html_url": "https://github.com/apollostack",
      "followers_url": "https://api.github.com/users/apollostack/followers",
      "following_url": "https://api.github.com/users/apollostack/following{/other_user}",
      "gists_url": "https://api.github.com/users/apollostack/gists{/gist_id}",
      "starred_url": "https://api.github.com/users/apollostack/starred{/owner}/{/repo}",
      "subscriptions_url": "https://api.github.com/users/apollostack/subscriptions",
      "organizations_url": "https://api.github.com/users/apollostack/orgs",
      "repos_url": "https://api.github.com/users/apollostack/repos",
      "events_url": "https://api.github.com/users/apollostack/events{/privacy}",
      "received_events_url": "https://api.github.com/users/apollostack/received_events",
      "type": "Organization",
      "site_admin": false
    },
    "private": false,
    "html_url": "https://github.com/apollostack/android-docs",
```



```
{
  "id": 67563416,
  "name": "android-docs",
  "full_name": "apollostack/android-docs",
  "owner": {
    "login": "apollostack",
    "id": 17189275,
    "avatar_url": "https://avatars.githubusercontent.com/u/17189275?v=3",
    "gravatar_id": "",
    "url": "https://api.github.com/users/apollostack",
    "html_url": "https://github.com/apollostack",
    "followers_url": "https://api.github.com/users/apollostack/followers",
    "following_url": "https://api.github.com/users/apollostack/following{/other_user}",
    "gists_url": "https://api.github.com/users/apollostack/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/apollostack/starred{/owner}/{/repo}",
    "subscriptions_url": "https://api.github.com/users/apollostack/subscriptions",
    "organizations_url": "https://api.github.com/users/apollostack/orgs",
    "repos_url": "https://api.github.com/users/apollostack/repos",
    "events_url": "https://api.github.com/users/apollostack/events{/privacy}",
    "received_events_url": "https://api.github.com/users/apollostack/received_events",
    "type": "Organization",
    "site_admin": false
  },
  "private": false,
  "html_url": "https://github.com/apollostack/android-docs",
  "description": "[WIP] :book: Docs for using Apollo with Android",
  "fork": false,
  "url": "https://api.github.com/repos/apollostack/android-docs",
  "forks_url": "https://api.github.com/repos/apollostack/android-docs/forks",
```


G R A P H Q L

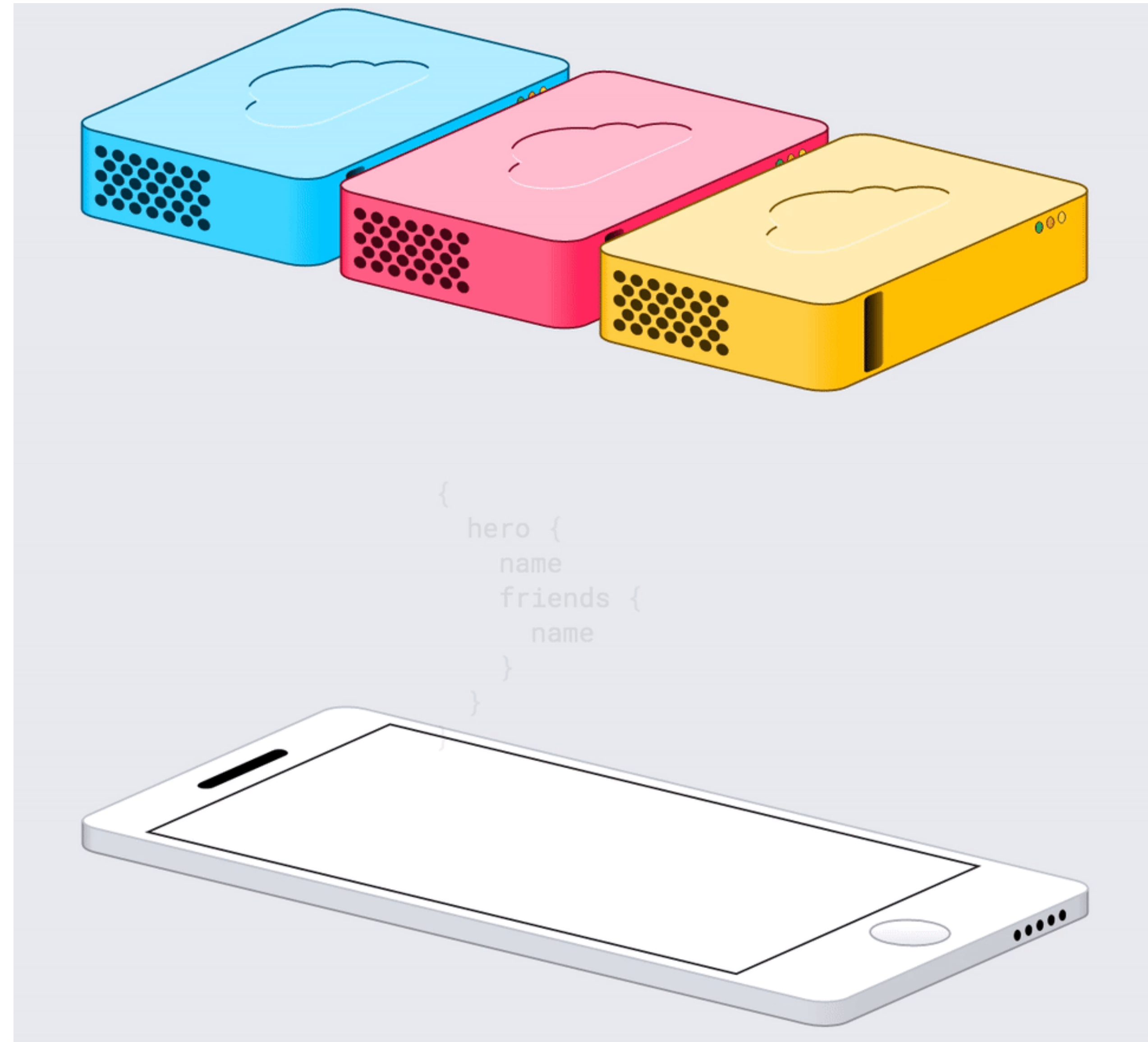
- Client **asks** for and **gets exactly** what it needs

```
{  
  hero {  
    name  
  }  
}
```

```
{  
  "hero": {  
    "name": "Luke Skywalker"  
  }  
}
```


G R A P H Q L

- **Multiple resources in single request**



GRAPHQL

DATA SOURCE

ASK FOR WHAT YOU WANT

CLIENT

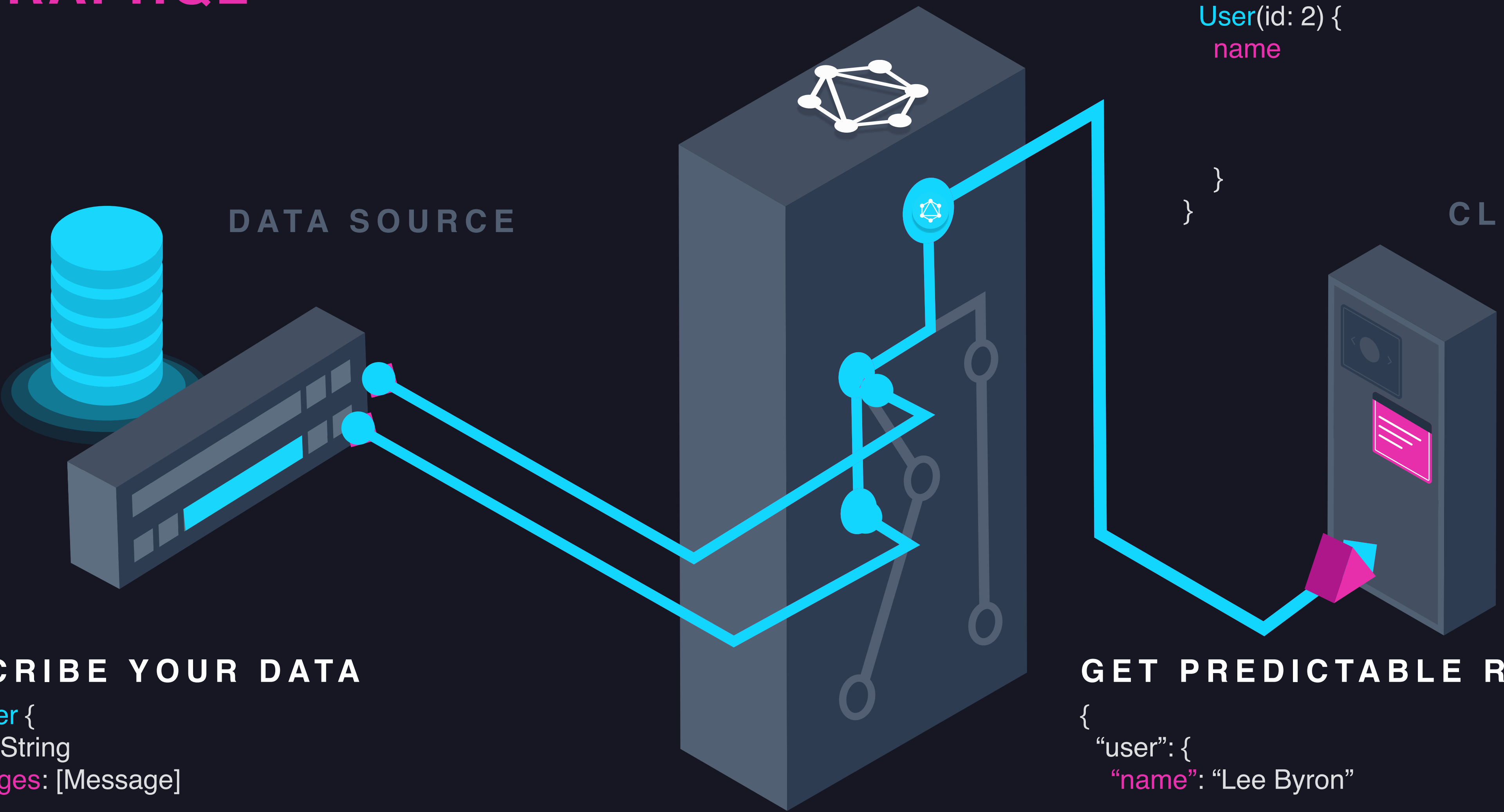
DESCRIBE YOUR DATA

```
type User {  
  name: String  
  messages: [Message]  
}  
type Message {  
  content: String  
  author: User  
}
```

GET PREDICTABLE RESULTS

```
{  
  "user": {  
    "name": "Lee Byron"  
  }  
}
```

```
{  
  User(id: 2) {  
    name  
  }  
}
```



GRAPHQL

SERVER

DESCRIBE YOUR DATA

```
type User {  
  name: String  
  messages: [Message]  
}  
  
type Message {  
  content: String  
  author: User  
}
```

3RD PARTY API

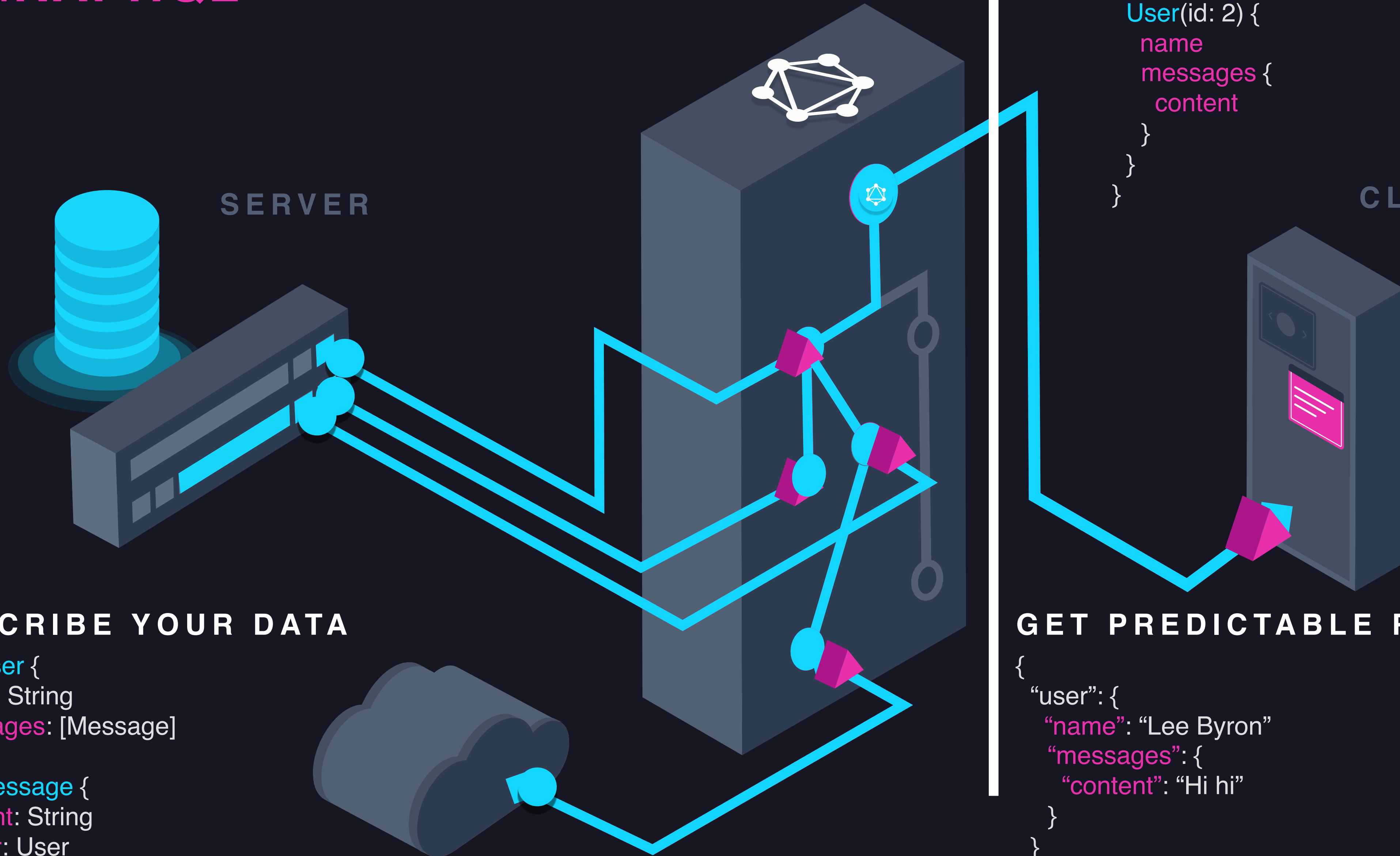
ASK FOR WHAT YOU WANT

```
{
  User(id: 2) {
    name
    messages {
      content
    }
  }
}
```

CLIENT

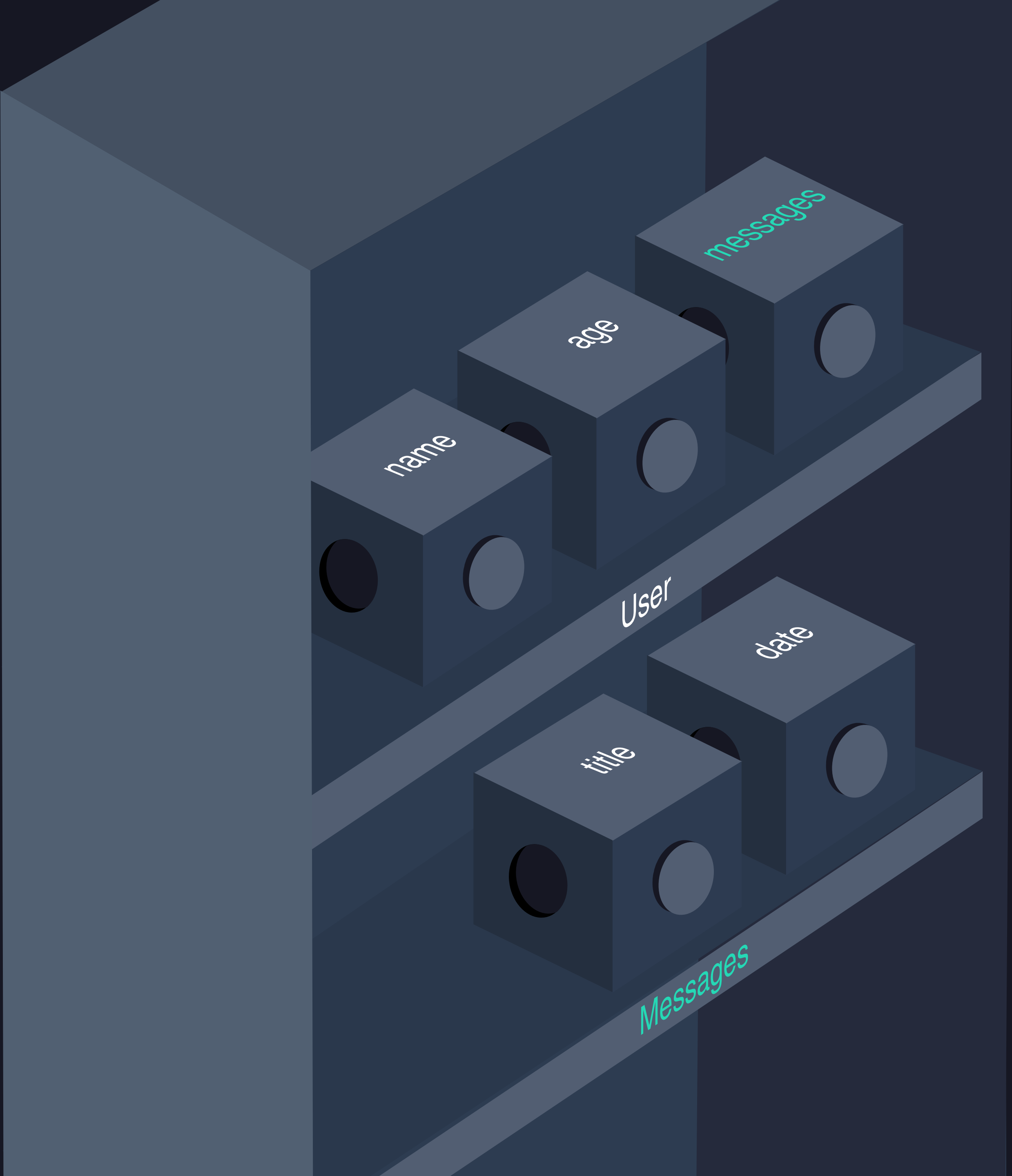
GET PREDICTABLE RESULTS

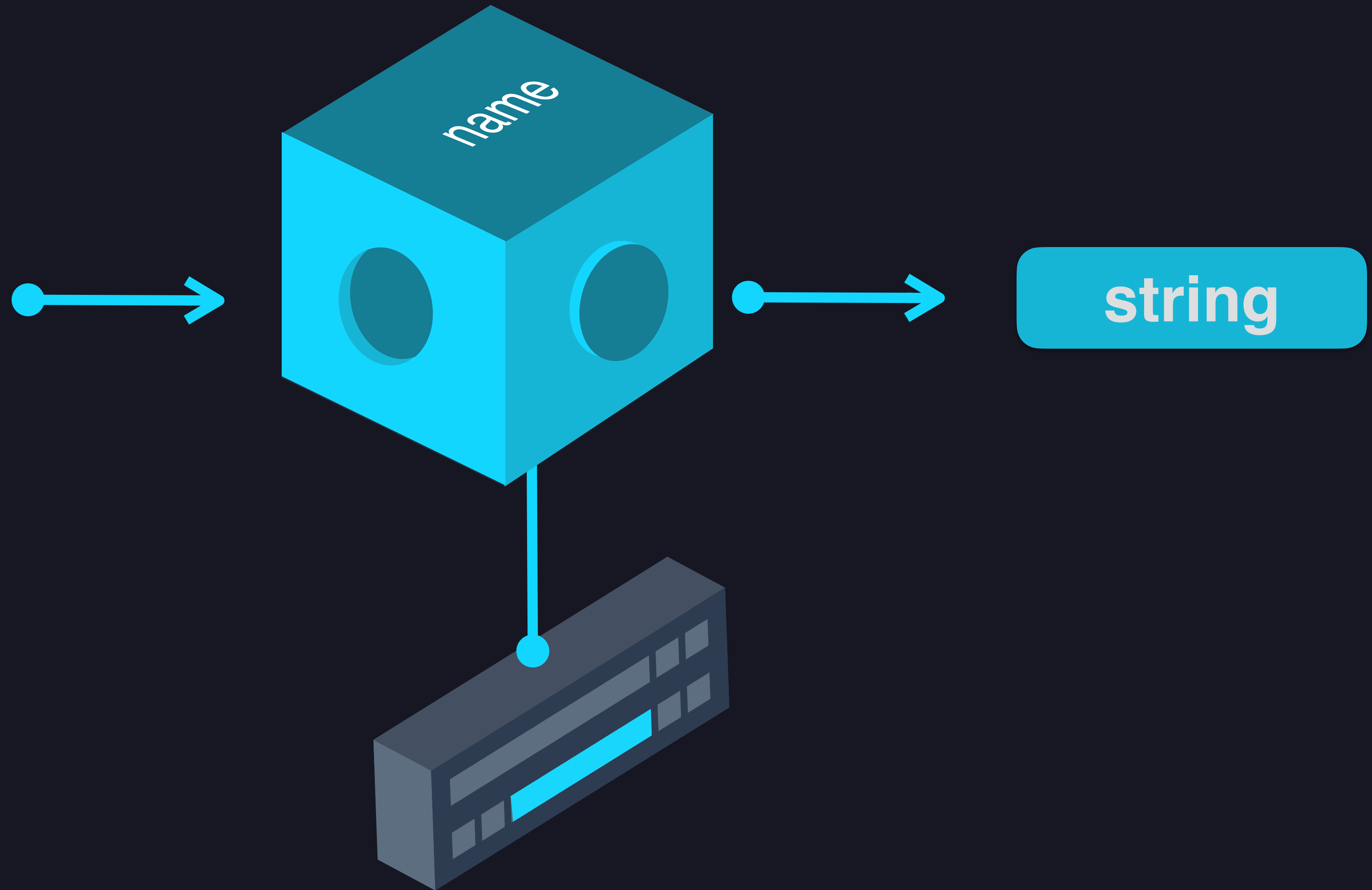
```
{
  "user": {
    "name": "Lee Byron"
    "messages": {
      "content": "Hi hi"
    }
  }
}
```



DESCRIBE YOUR DATA

```
type User {  
  name: String  
  age: Int  
  messages: [Message]  
}  
type Message {  
  title: String  
  date: Date  
}
```



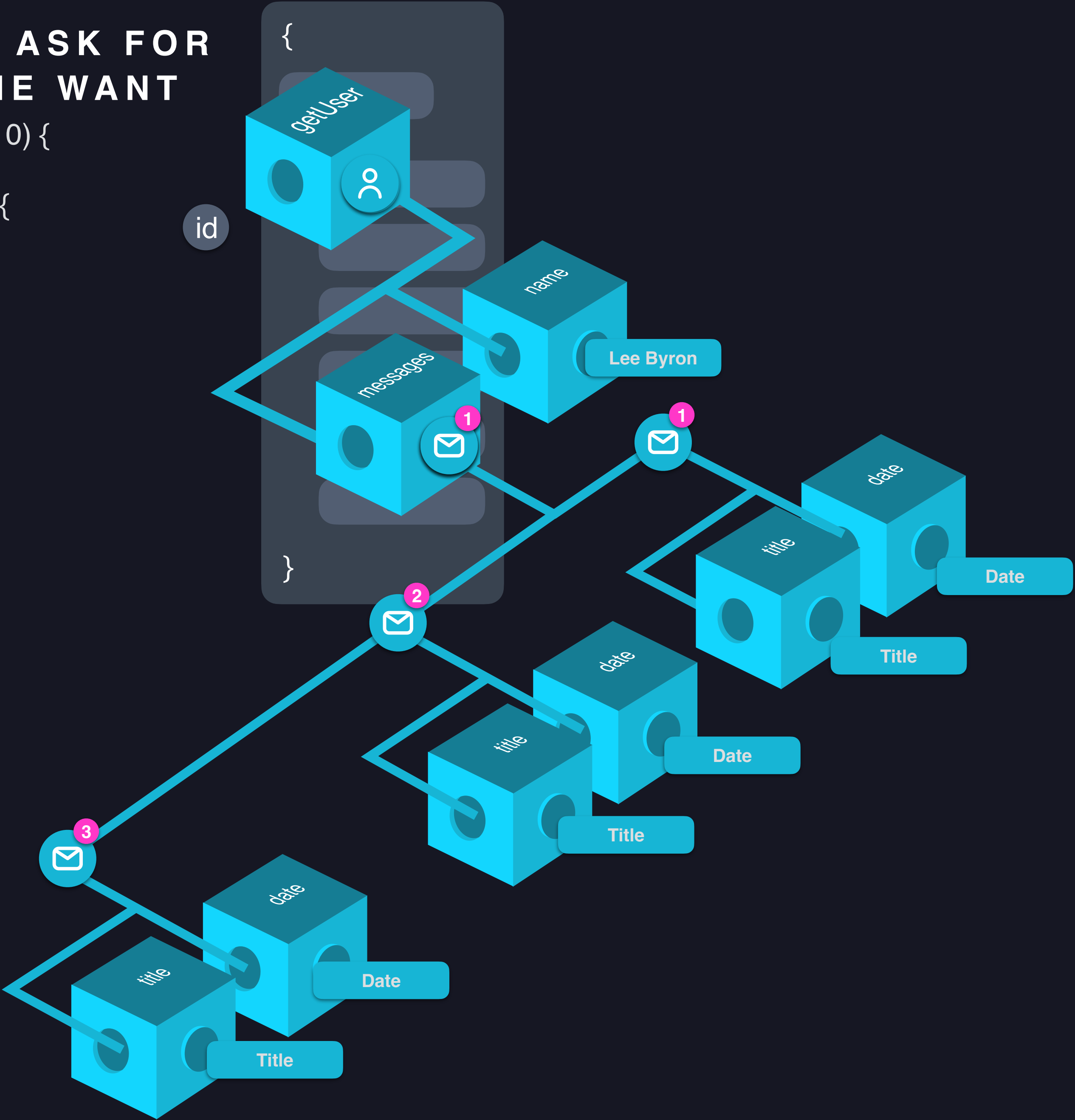
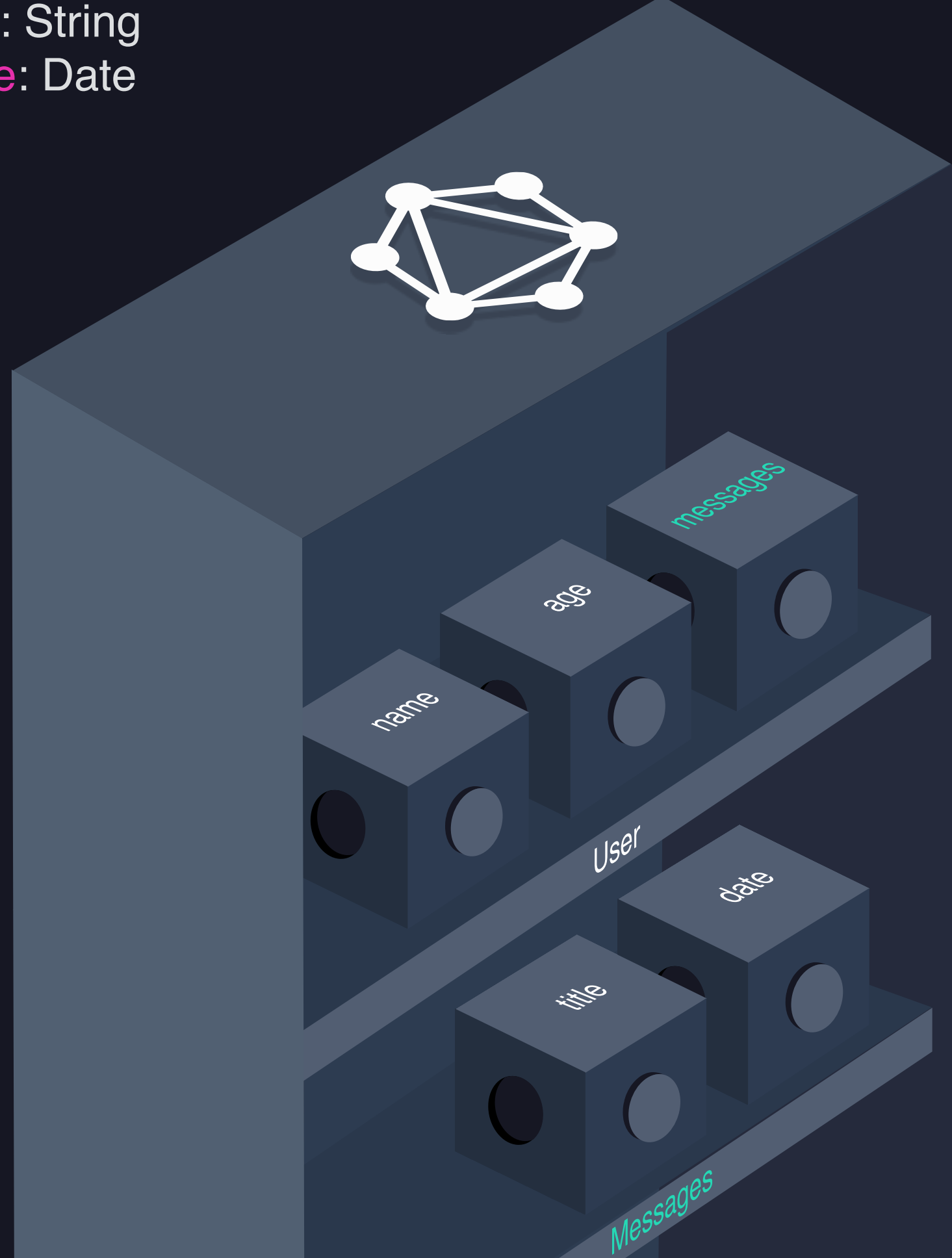


DESCRIBE YOUR DATA

```
type User {  
  name: String  
  age: Int  
  messages: [Message]  
}  
type Message {  
  title: String  
  date: Date  
}
```

CLIENT ASK FOR WHAT HE WANT

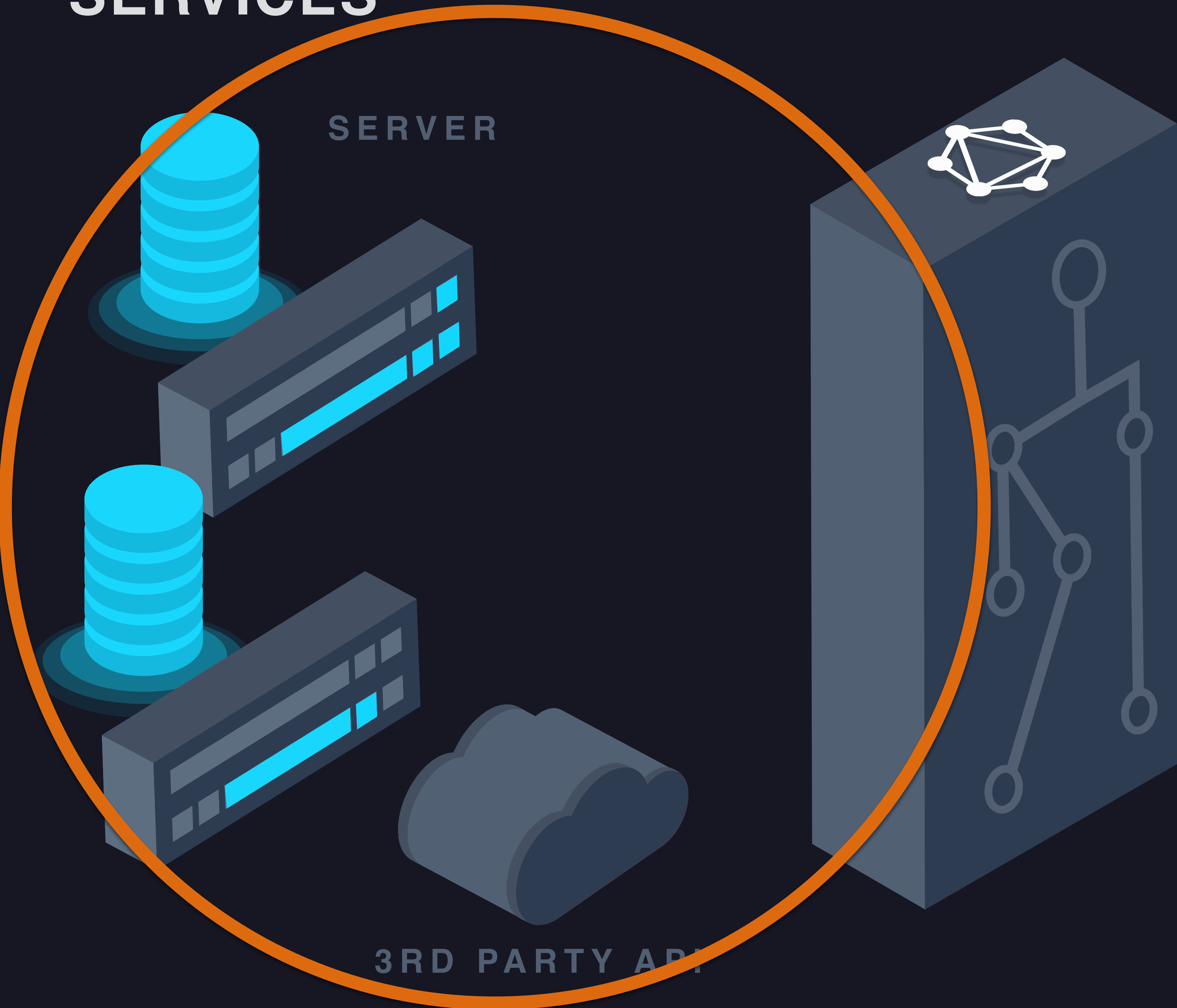
```
getUser (id:10) {  
  name  
  messages {  
    title  
    date  
  }  
}
```



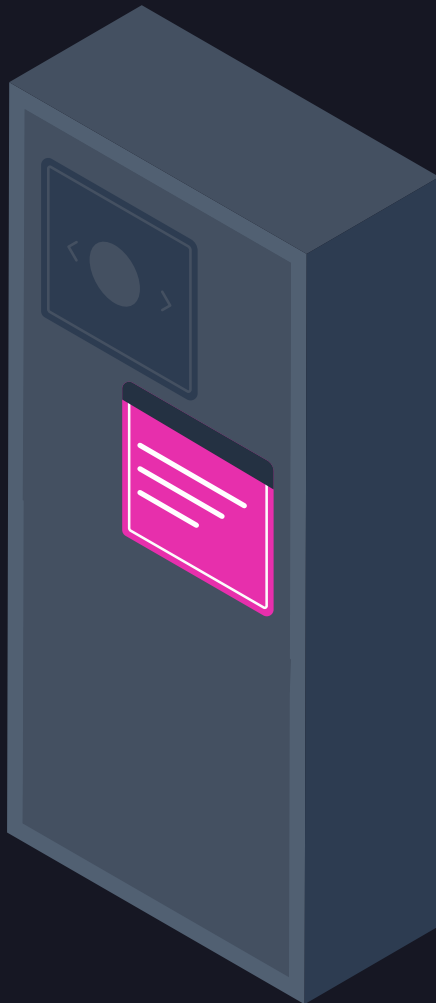
BENEFITS GRAPHQL

- Network Performance
- Schema of Data Graph
- Orchestration and Automation
- Explicit relations between Teams/Services

BACKEND
SERVICES

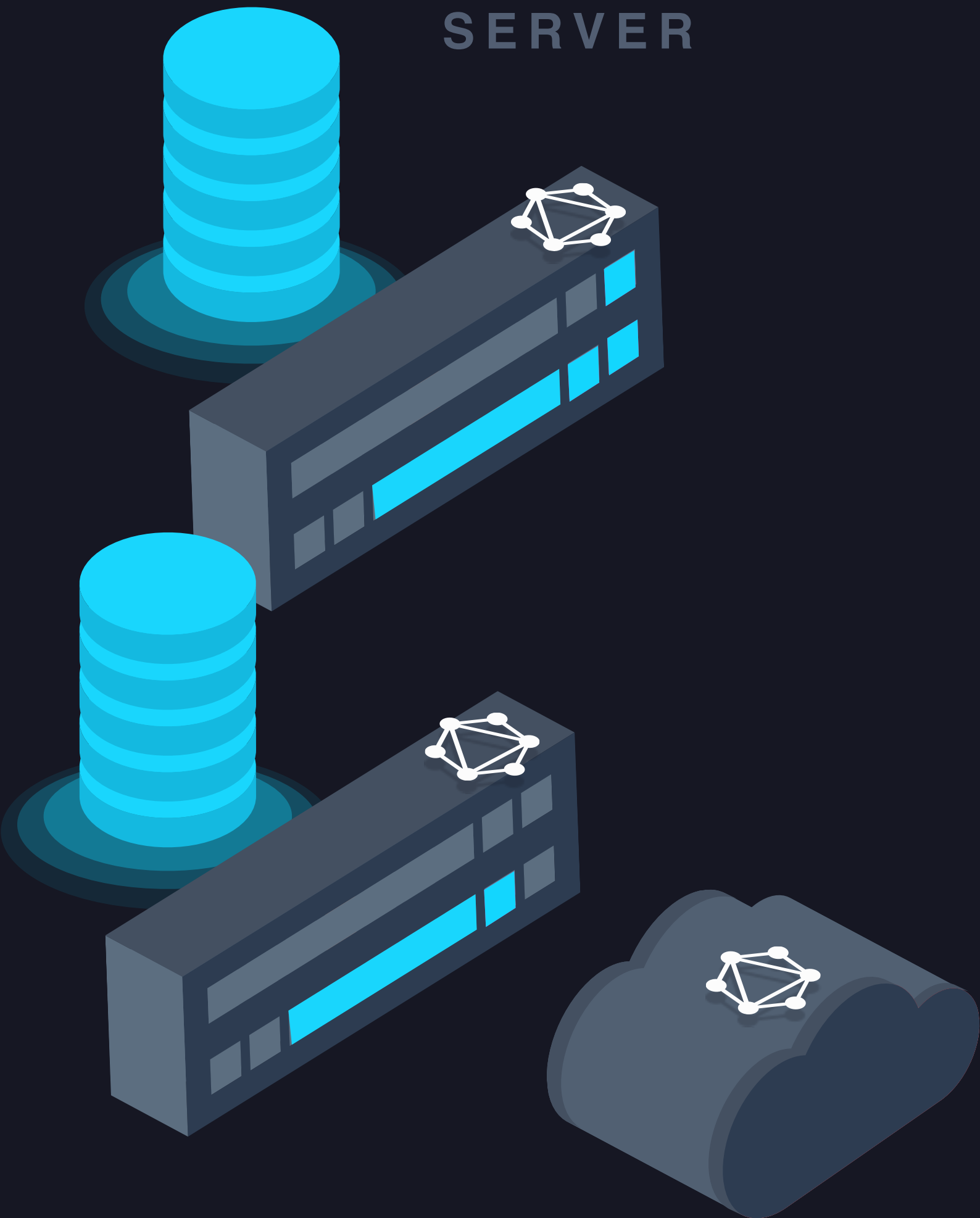


CLIENT

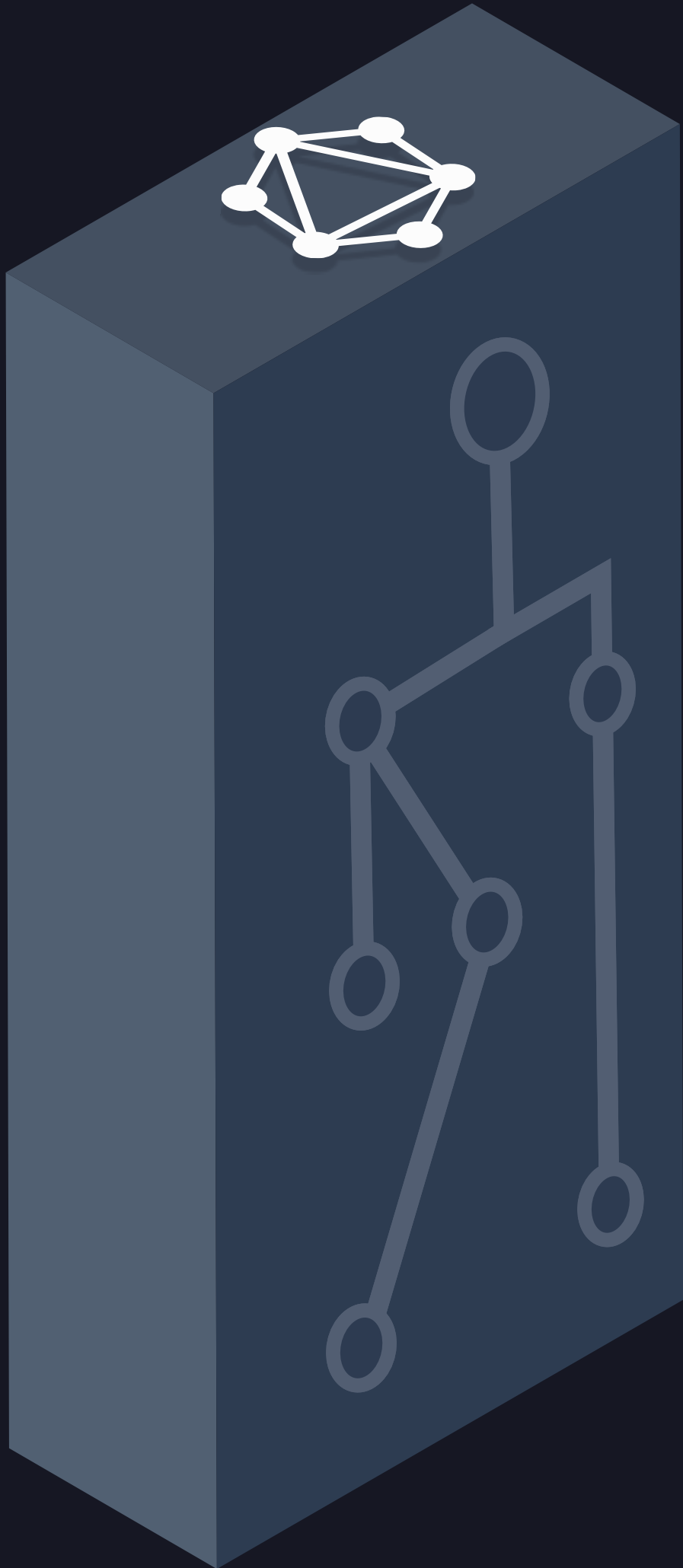


BACKEND
SERVICES

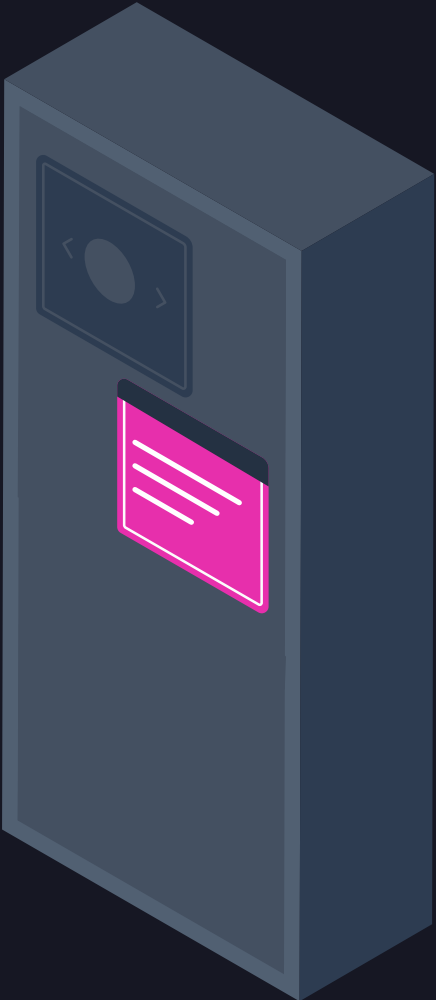
SERVER



3RD PARTY API

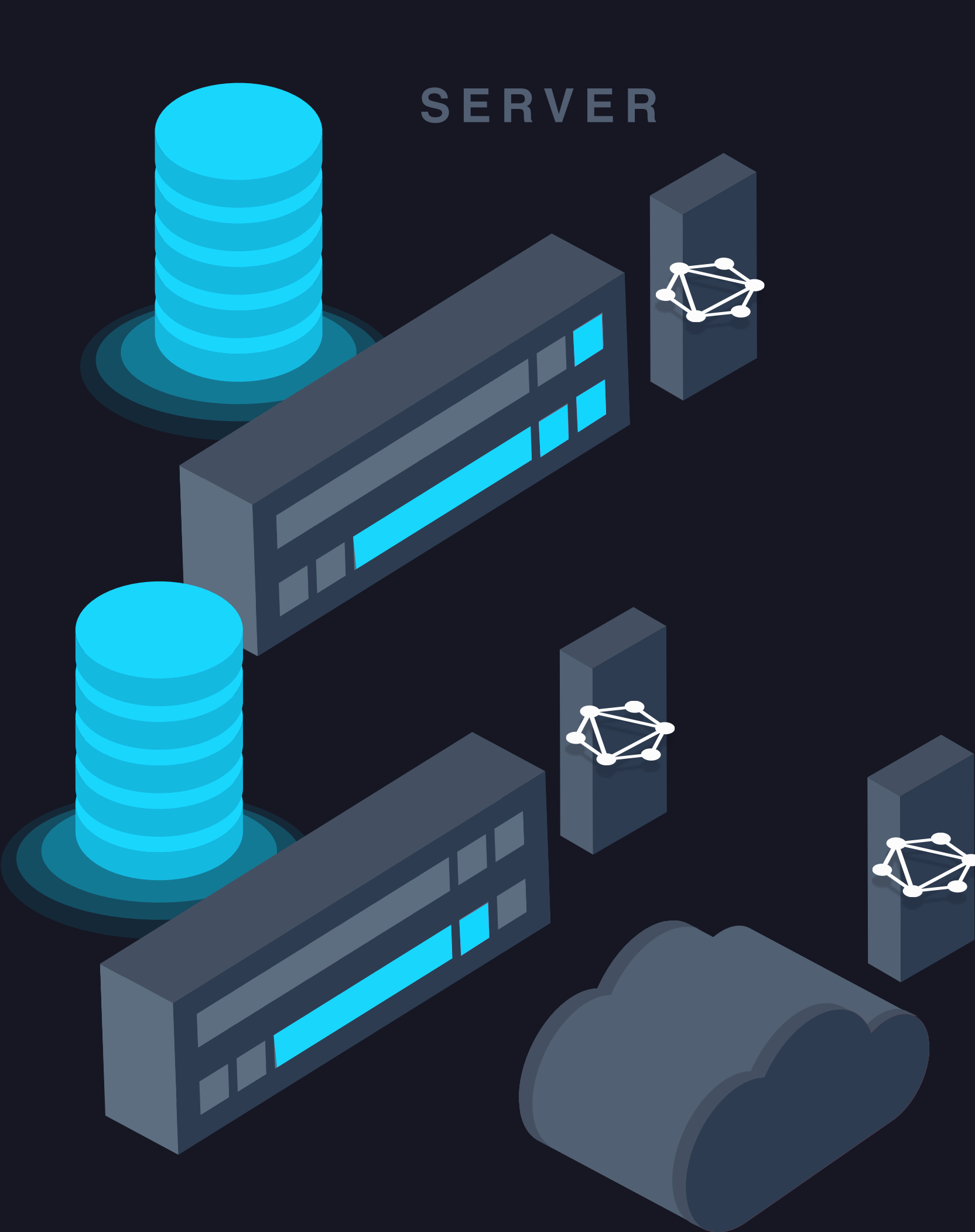


CLIENT

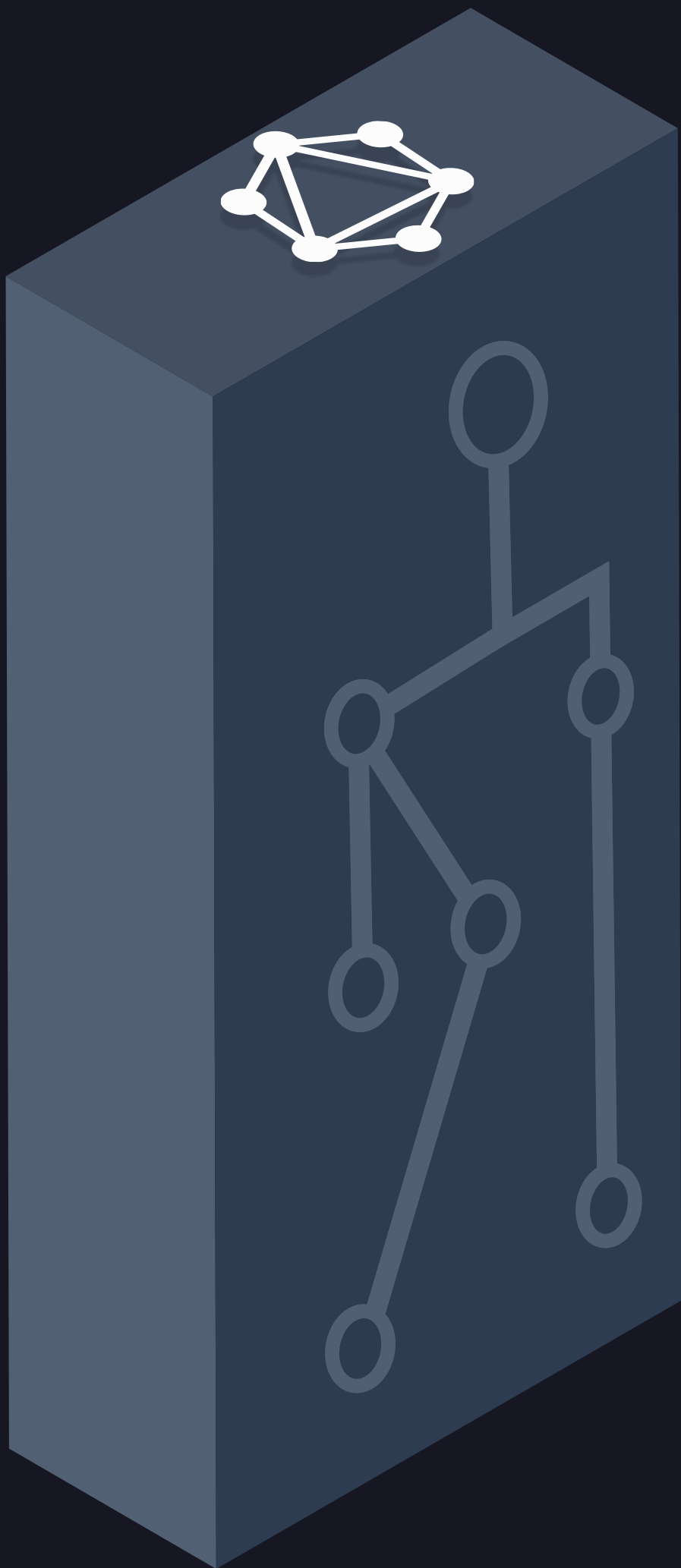


BACKEND SERVICES

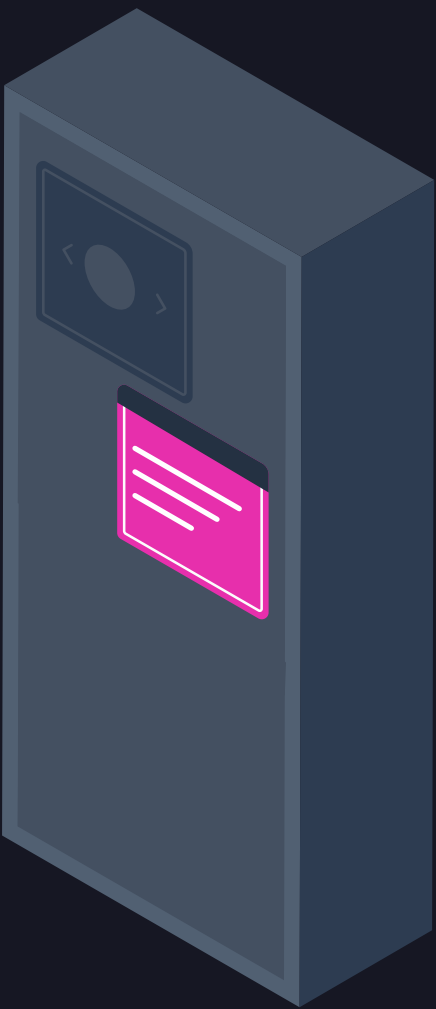
SERVER



3RD PARTY API



CLIENT

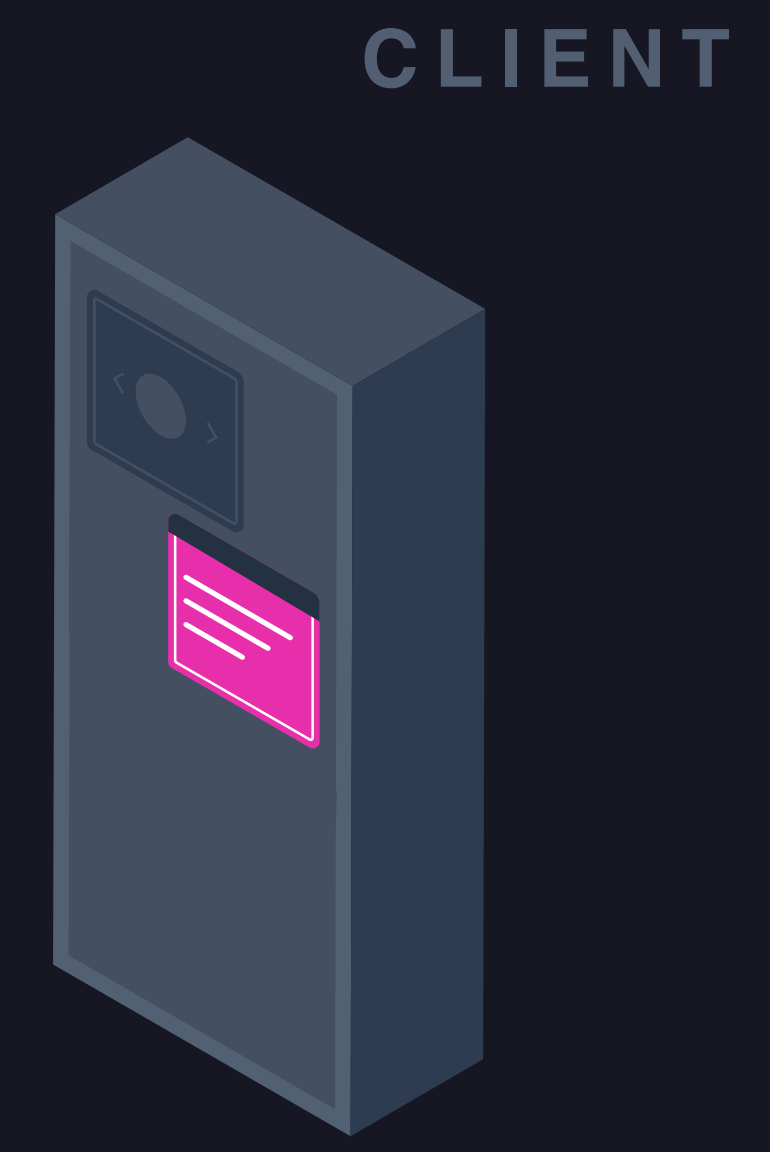
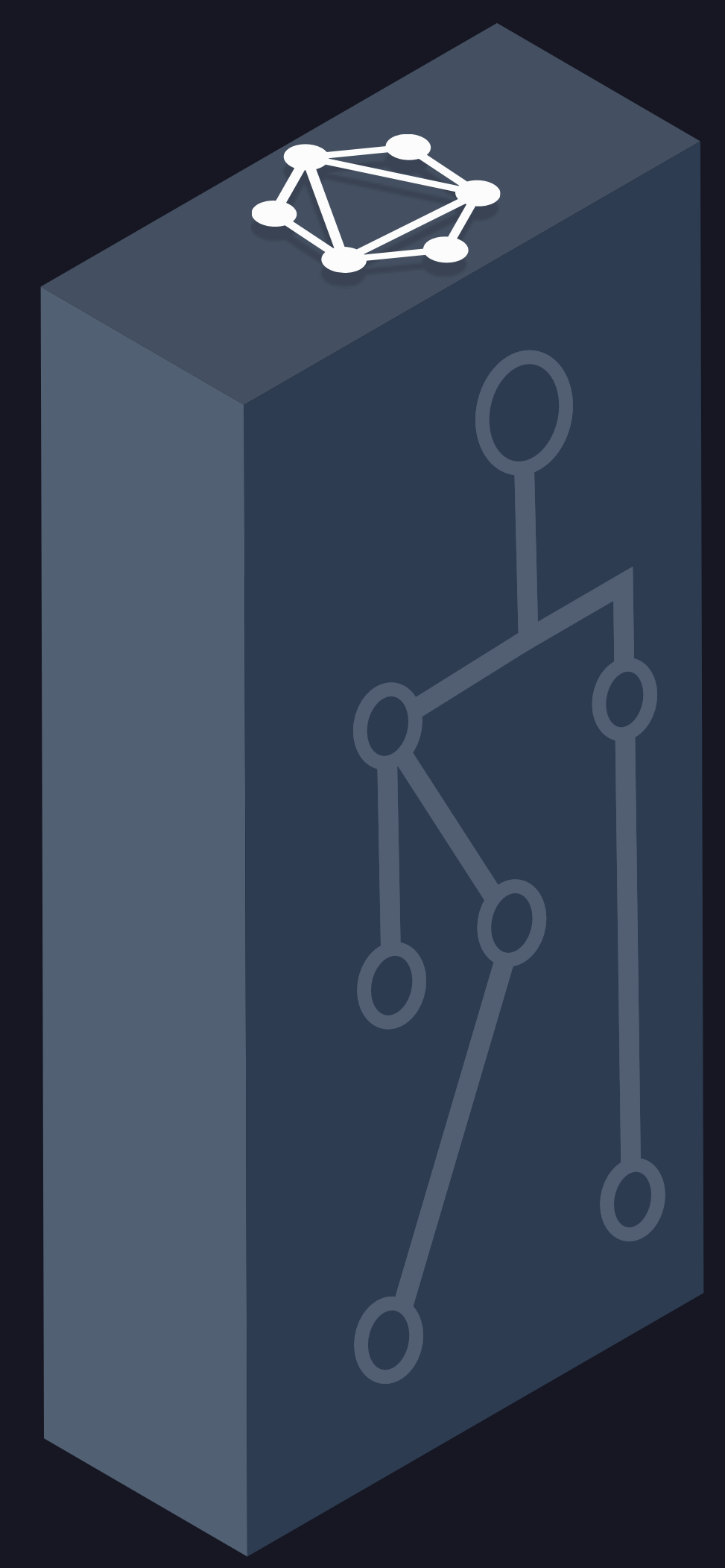
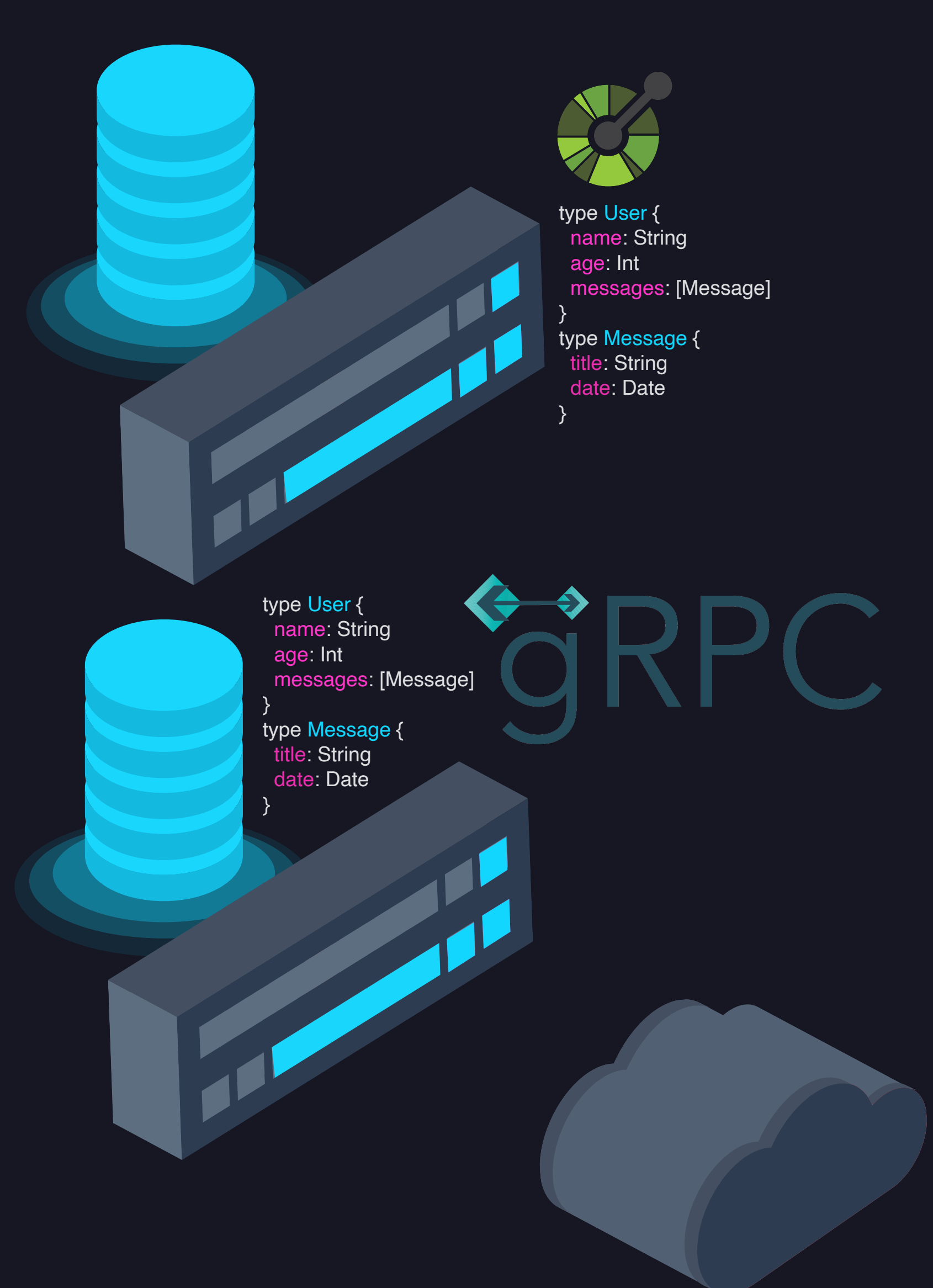


BACKEND APIS

WHAT WE LOVE ABOUT GRAPHQL?

- Schema
- Query language and engine

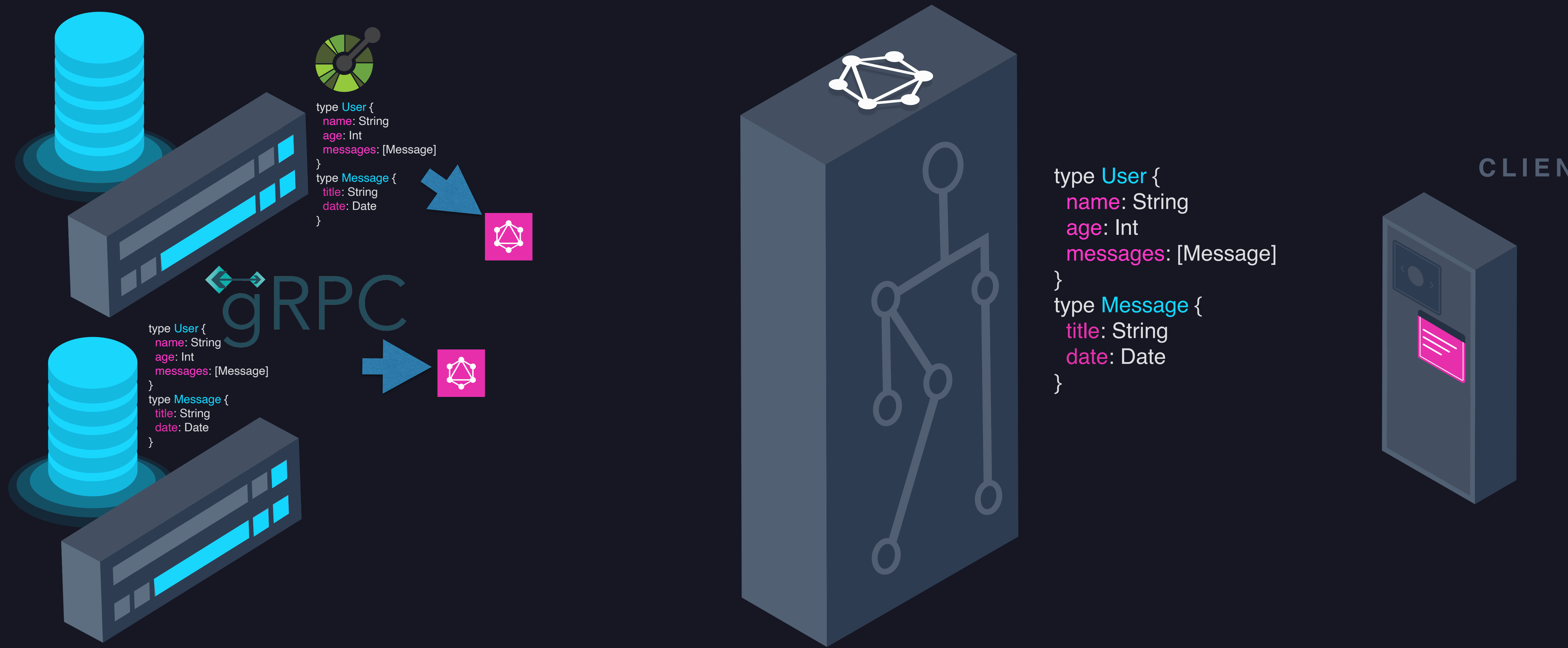
THE BACKEND COMPLEX



WHAT WE HAVE

- Schemas and protocols
- Docs
- Live data and logs

GRAPHQL MESH



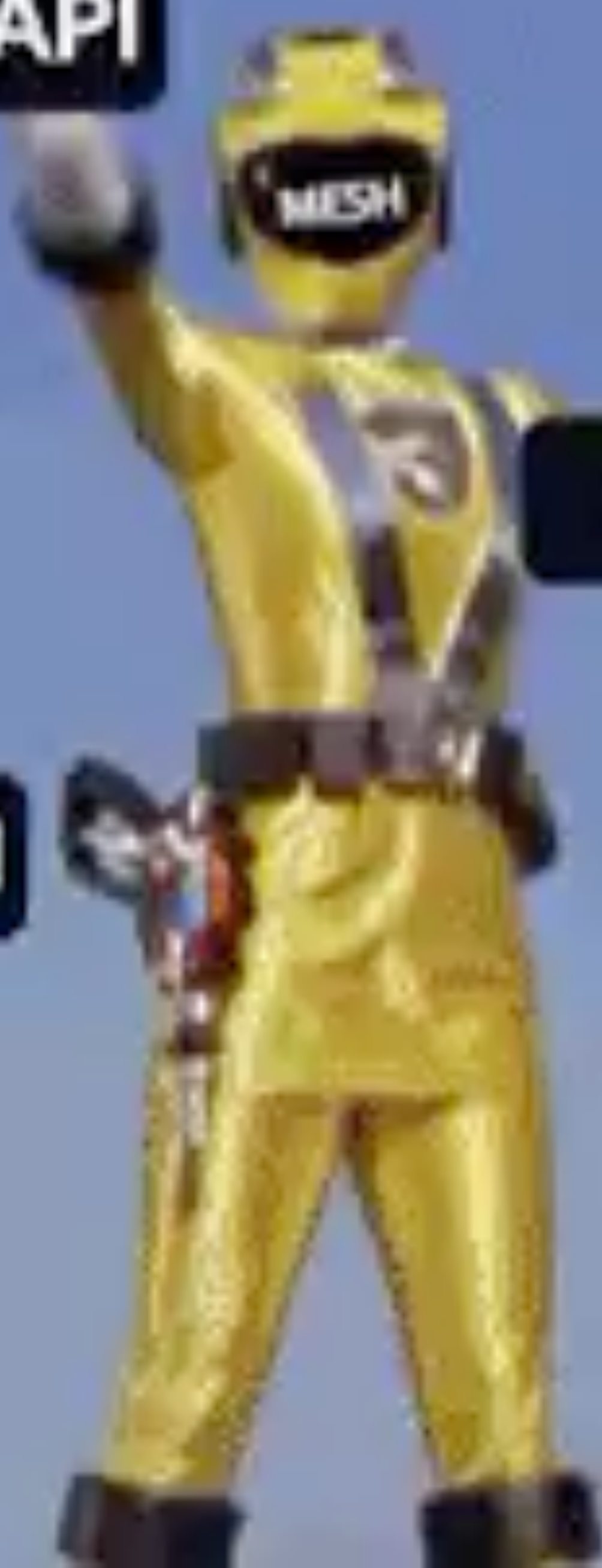
REST:API


MESH

OData

OPEN API

gRPC





Query
anything

Query anything

Using source handlers, you can consume any API in GraphQL Mesh

 OpenAPI/Swagger

 oData

 gRPC

 MongoDB

 Postgres

 SOAP

 MySQL

 JSON Schema

 Postgraphile

 Neo4j

 SQLite

 & More...

```

paths:
  /users/{id}:
    get:
      tags:
        - Users
      summary: Gets a user by ID.
      operationId: getUserById
      parameters:
        - name: id
          in: path
          description: User ID
          required: true
          schema:
            type: integer
      responses:
        '200':
          description: Successful operation
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/User'
        '404':
          description: User not found
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Error'
components:
  schemas:
    Error:
      type: object
      properties:
        code:
          type: integer
          format: int32
        message:
          type: string
    User:
      type: object
      properties:
        id:
          type: integer
        name:
          type: string
      required:
        - id
        - name

```



.meshrc.yaml

```

sources:
  - name: MyOpenapiApi
    handler:
      openapi:
        source: ./my-schema.json

```



```

type Query {
  ""
  Gets a user by ID.
  ""
  getUserById(id: Int!): GetUserByIdResponse!
}

union GetUserByIdResponse = User | Error

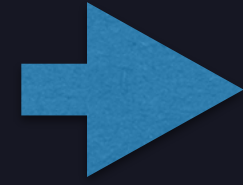
type User {
  id: Int!
  name: String!
}

type Error {
  code: Int!
  message: String!
}

```



```
{  
  "id": 1,  
  "name": "John Doe",  
  "email": "john@doe.com"  
}
```



```
sources:  
  - name: MyApi  
    handler:  
      jsonSchema:  
        endpoint: https://some-service-url/endpoint-path/  
        operations:  
          - type: Query  
            field: user  
            path: /user/{args.id}  
            method: GET  
            responseSample: ./user.sample.json  
            responseTypeName: User
```



```
type Query {  
  user(id: ID!): User  
}  
  
type User {  
  id: ID  
  name: String  
  email: EmailAddress  
}
```

.meshrc.yml

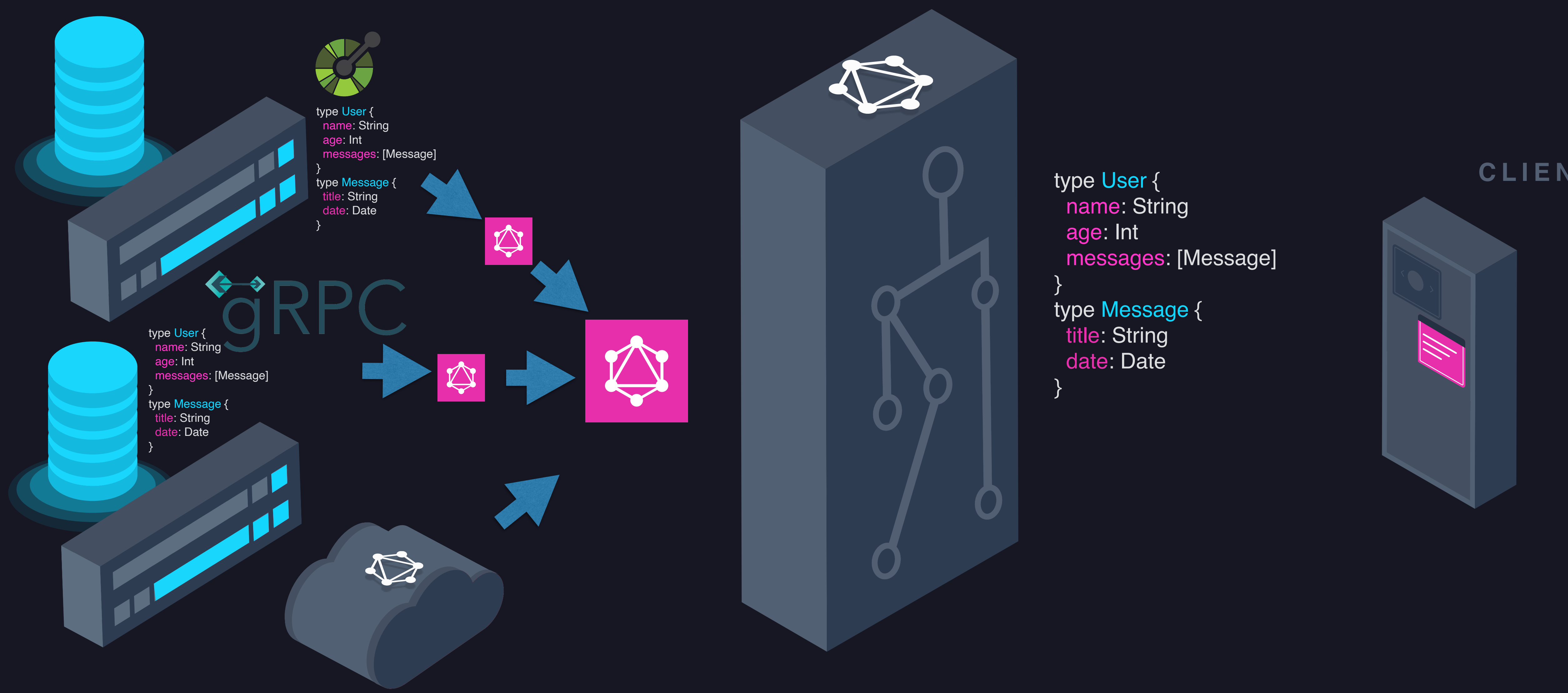
```
15 sources:
14   - name: Countries
13     handler:
12       graphql:
11         endpoint: https://countries.trevorblades.com
10   - name: CartQL
9     handler:
8       graphql:
7         endpoint: https://api.cartql.com
6   - name: Stripe
5     handler:
4       openapi:
3         source: https://raw.githubusercontent.com/stripe/openapi/master/openapi/spec3.json
2         baseUrl: https://api.stripe.com
1         operationHeaders:
16           Authorization: "Bearer
           sk_test_51IQEb1E7xu61WrBojW5M3B1xev2Td2SuMzy5UIc1enIZchumuSHmWvRbzdMSDFwcAA0suZFhCEf5nyiq0i1u7LxY00IGY29oNC"
1
```


Clients can work with services that
aren't GraphQL as if they were

GRAPHQL MESH

- Take the best out of GraphQL
- Using your existing services
- Use the original protocol with better dev experience

GRAPHQL MESH



GraphQL Mesh compose sources
as a single unified GraphQL Schema



Connect datasources

Work with services that aren't GraphQL as if they were.



Automatically create

type-safe GraphQL APIs from any datasource



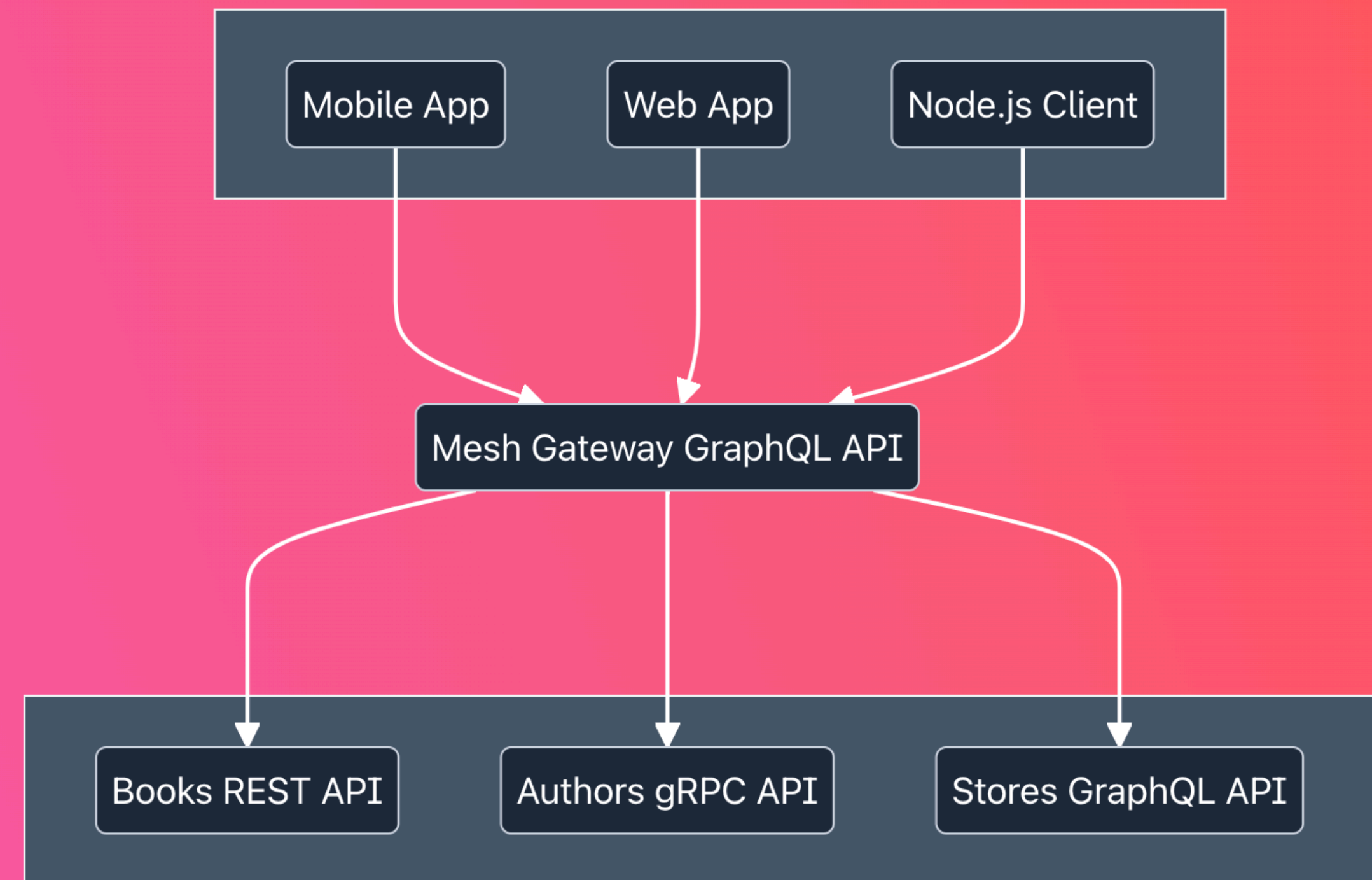
Extend datasource

with the data from another - fully type safe



Mock, cache and transform

your entire schema



Manipulate data

Easily transform your schema

Naming conventions

Rename fields, change casings, add pre/suffixes

Modify results

Filter, sort, replace and more

Alter resolvers

Add middleware to your existing resolvers

Schema structure

Type merging, encapsulate, extend sources



The Graph of Everything

Federated architecture for any API service



GRAPHQL MESH

Query anything, run
anywhere.

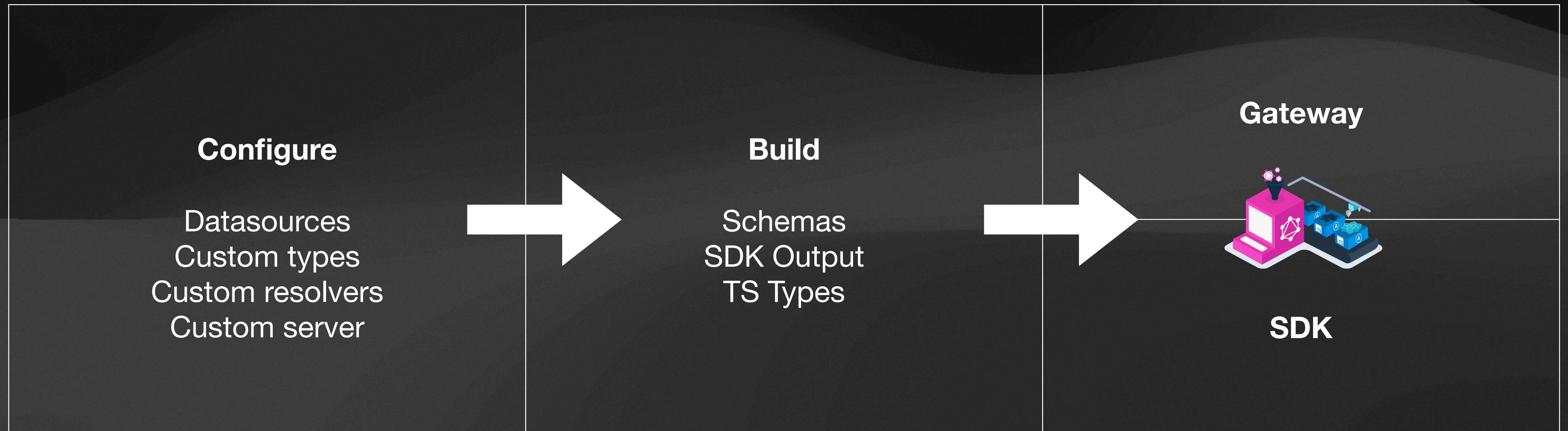


Mesh is a framework that helps shape
and build an executable GraphQL
Schema from multiple data sources

The background features a series of overlapping, stylized mountain ranges. The mountains are rendered in various shades of dark teal and navy blue, creating a sense of depth. The peaks are rounded and layered, with some appearing closer and more vibrant, while others recede into the dark background.

Run anywhere

How does GraphQL Mesh work?



Where can I use GraphQL Mesh?

Gateway

Run the gateway as a separate service, or inside your project as an additional route.

SDK

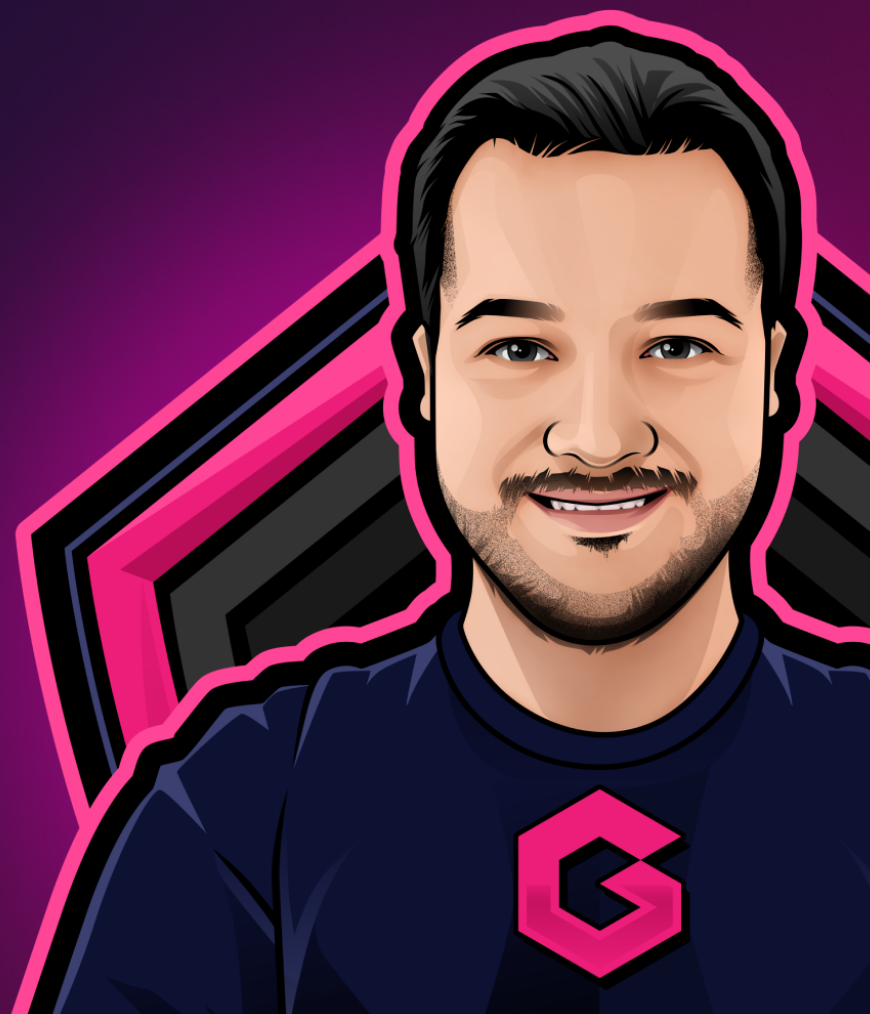
Use the SDK inside of your client project using the exported functions.





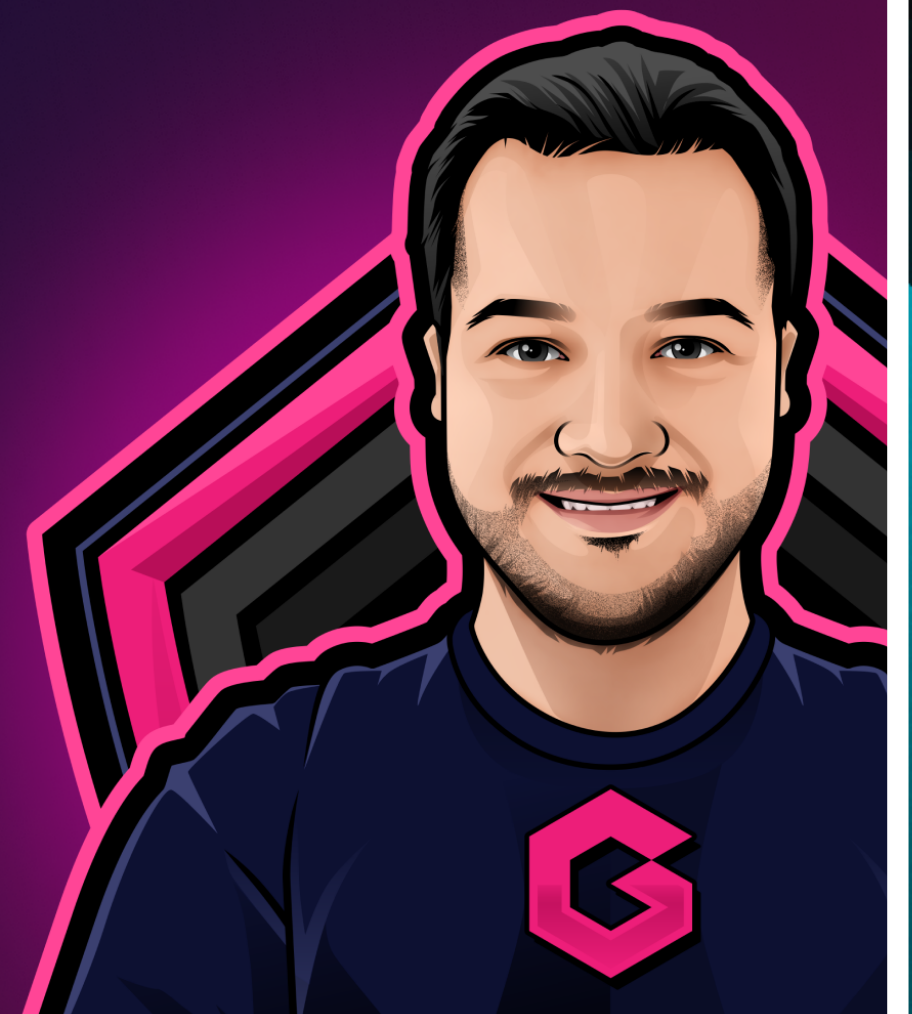
GRAPHQL MESH AS A GATEWAY

 graphql.wtf



GRAPHQL MESH (SDK) & NEXT.JS

 graphql.wtf



Deploy Anywhere

feTS Server provides a super fast HTTP server that can run anywhere with the power of [@whatwg-node/server](#)



AWS Lambda →



Azure Functions →



Bun →



Cloudflare Workers →



Deno →



Google Cloud Functions →



WebSockets →



Next.js →



Node.js →

Extend with plugins

Extend your gateway's capabilities with the plugins



Response Caching

Add caching to your GraphQL service easily



Monitoring & Tracing

Monitor your service with built-in support for Prometheus, NewRelic, Sentry, StatD and OpenTelemetry



Enhanced Security

Authentication (Basic/JWT/Auth0/...), authorization, rate-limit and more.



And much more!

Mocking, caching, live queries...

sources:

- name: OpenBrewery

handler:

jsonSchema:

baseUrl: <https://api.openbrewerydb.org>

operations:

- type: Query

field: breweries

path: breweries

method: GET

requestSample: [./json-samples/breweriesInput.json](#)

responseSample: [./json-samples/breweries.json](#)

plugins:

- responseCache:

ttlPerCoordinate:

- coordinate: Query.*

ttl: 3600

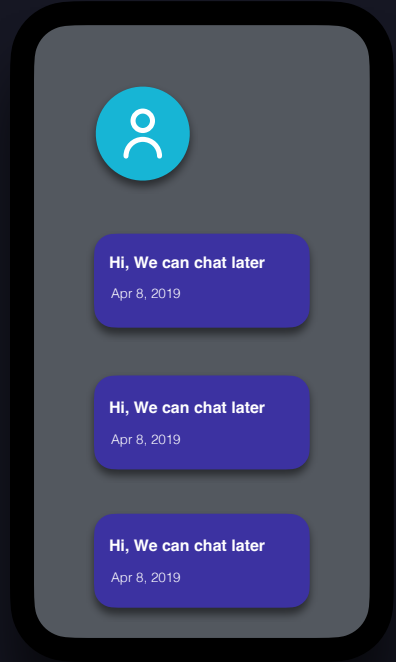
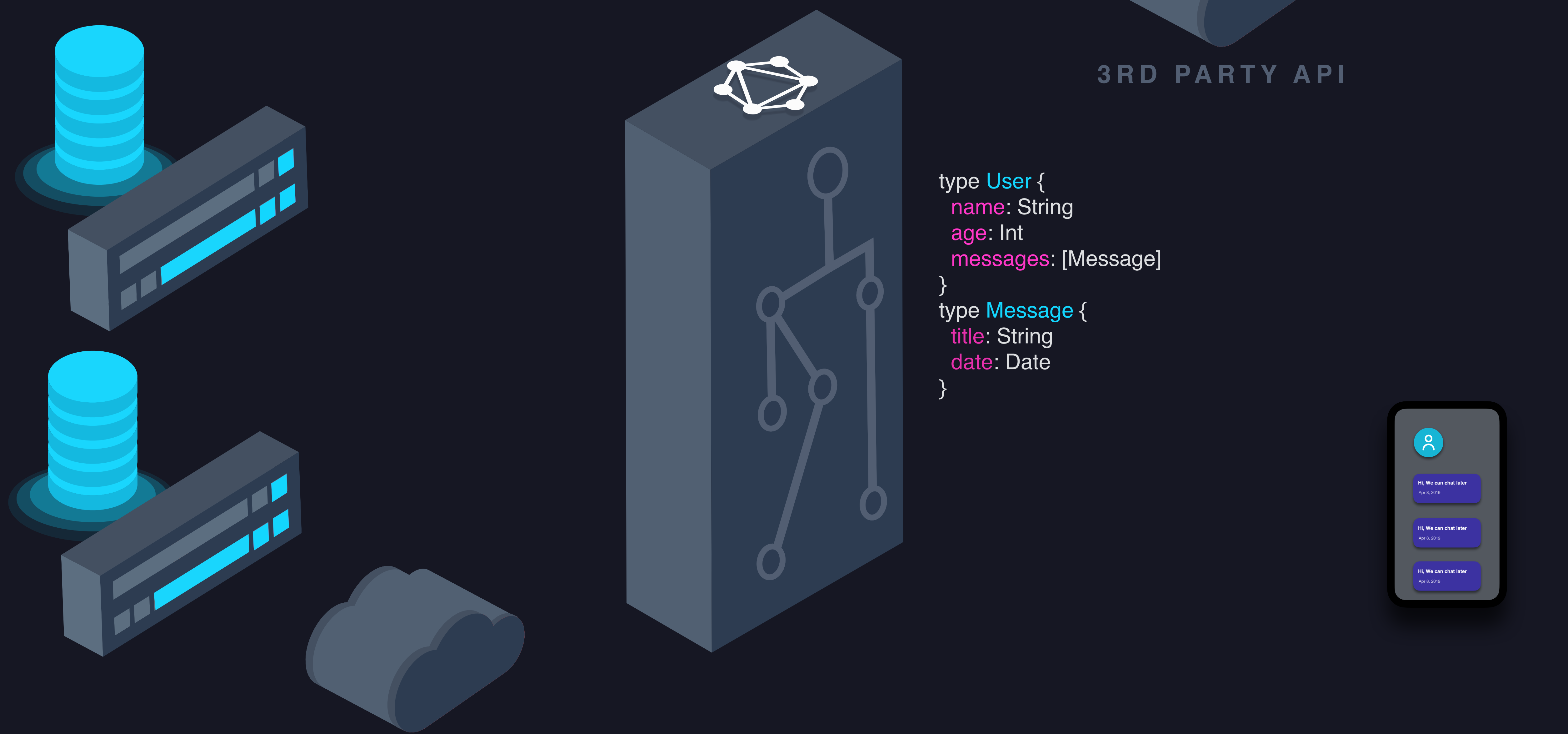
cache:

cfwKv:

namespace: MESH

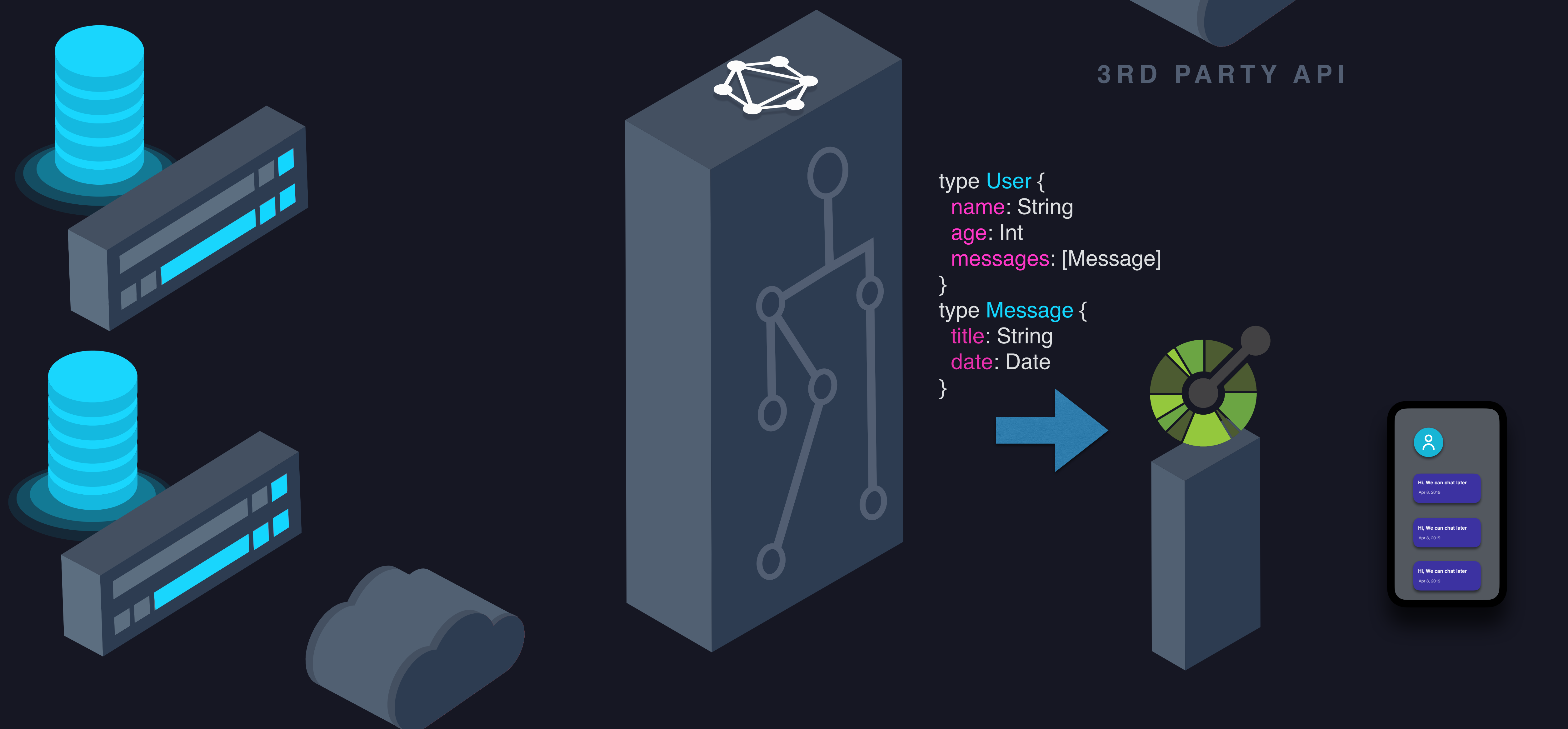
SOFA

GRAPHQL TO OPENAPI



SOFA

GRAPHQL TO OPENAPI





BENEFITS GRAPHQL

- Manage versions
- Prevent breaking changes
- Know your usage



Tooling for GraphQL to
Compare schemas & Validate docs.



THE GUILD

GRAPHQL INSPECTOR

Detected the following changes (6) between schemas:

- ✖ Field **posts** was removed from object type **Query**
- ✖ Field **modifiedAt** was removed from object type **Post**
- ✔ Field **Post.id** changed type from **ID** to **ID!**
- ✔ Deprecation reason on field **Post.title** has changed from **No more used** to **undefined**
- ✔ Field **Post.title** changed type from **String** to **String!**
- ✔ Field **Post.createdAt** changed type from **String** to **String!**

error Detected 2 breaking changes

GRAPHQL INSPECTOR

GraphQL Inspector

```
$ #!/bin/bash -eo pipefail
yarn test

yarn run v1.16.0
$ graphql-inspector diff 'git:origin/master:./schema.graphql' 'schema.graphql'

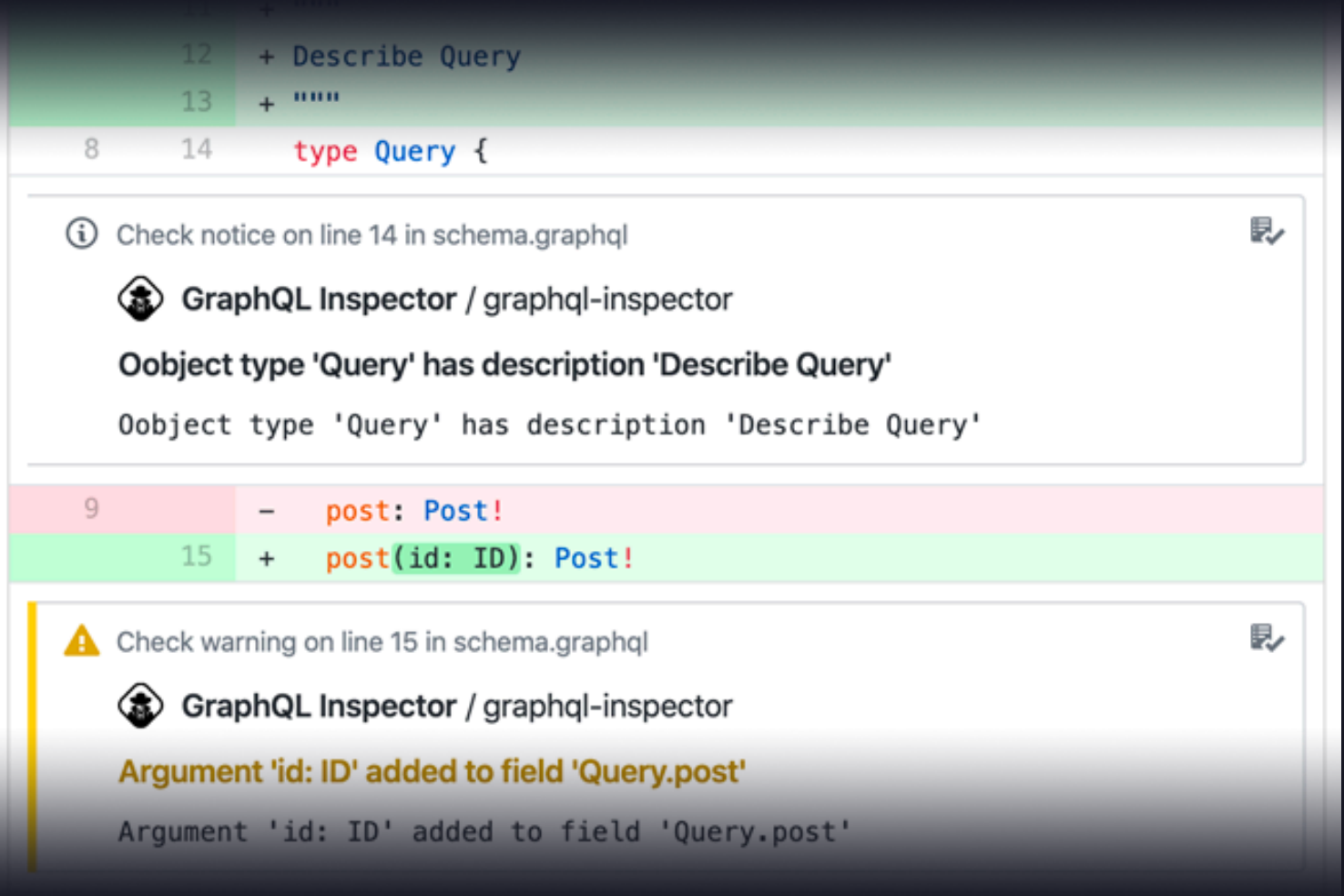
Detected the following changes (2) between schemas:

✖ Field user was removed from object type Query
✖ Field User.createdAt changed type from String to Int


error Detected 2 breaking changes

error Command failed with exit code 1.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
Exited with code 1
```


GRAPHQL INSPECTOR



GraphQL Inspector APP 00:56

 Hi, I found 3 changes in your schema:

Breaking changes


Field `bar` was removed from object type `Query`



Dangerous changes


Argument `limit: Int` added to field `Query.users`

Safe changes

Field `name` was added to object type `User`

 All checks have passed Hide all checks
3 successful checks

  graphql-inspector Successful in 5s — Everything looks good [Details](#)

 This branch has no conflicts with the base branch
Merging can be performed automatically.



GRAPHQL INSPECTOR

type **Post**
●●●●● (96%) EmailPost
●●●●● (78%) BlogPost
●●●●● (72%) MailPost

type **BlogPost**
●●●●● (94%) MailPost
●●●●● (78%) Post
●●●●● (72%) EmailPost

type **MailPost**
●●●●● (94%) BlogPost
●●●●● (76%) EmailPost
●●●●● (72%) Post

type **EmailPost**
●●●●● (96%) Post
●●●●● (76%) MailPost
●●●●● (72%) BlogPost

GRAPHQL INSPECTOR

success Schema coverage based on documents:

```
type Query {  
  post x 1  
  posts x 0  
}  
  
type Post {  
  id x 1  
  title x 1  
  createdAt x 0  
  modifiedAt x 0  
}
```

LIVE
DATA



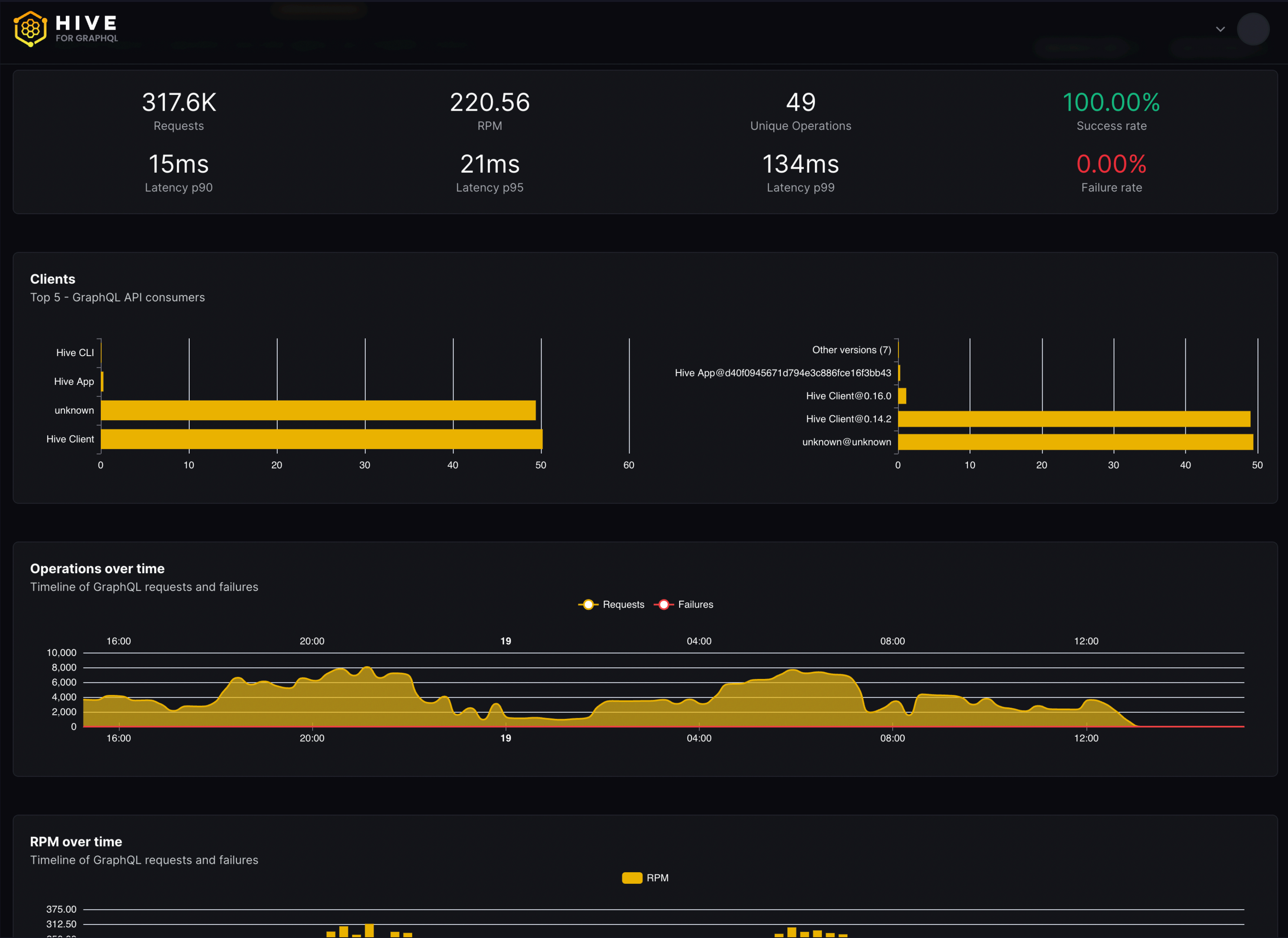
ANNOUNCING HIVE

GraphQL performance
monitoring, and schema registry

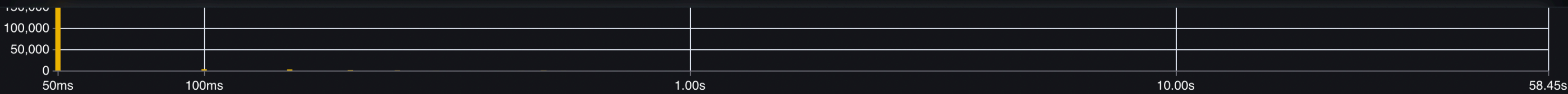


HIVE

- Monitoring
- Schema Checks
- Schema high availability storage
- History of changes
- Integrated into your workflow (CI/DC, Notifications)







Operations

List of all operations with their statistics


OPERATION	KIND	P90	P95	P99	FAILURE RATE	REQUESTS	TRAFFIC
a2ed_schemaPublish	mutation	9ms	12ms	151ms	0.00%	159.4K	50.19% ██████████
4960_myTokenInfo	query	16ms	25ms	50ms	0.00%	156.9K	49.41% ██████████
cc02_project	query	76ms	108ms	1.32s	0.01%	168	0.05% ██████████
44f9_targets	query	76ms	84ms	384ms	0.00%	160	0.05% ██████████
5a52_schemaCheck	mutation	892ms	999ms	1.24s	0.00%	135	0.04% ██████████
5f81_versions	query	61ms	78ms	447ms	0.00%	123	0.04% ██████████
e8dc_schemaPublish	mutation	920ms	1.58s	1.81s	0.00%	117	0.04% ██████████
1a21_organizationActivities	query	66ms	76ms	178ms	0.00%	85	0.03% ██████████
6153_projectActivities	query	59ms	64ms	75ms	0.00%	72	0.02% ██████████
349f_LatestSchema	query	54ms	68ms	79ms	0.00%	42	0.01% ██████████
a092_lab	query	117ms	200ms	290ms	0.00%	36	0.01% ██████████
3a56_durationHistogram	query	1.38s	31.81s	59.23s	0.00%	30	0.01% ██████████
2176_generalOperationsStats	query	1.39s	1.53s	1.63s	0.00%	30	0.01% ██████████
f17b_operationsStats	query	1.34s	9.03s	16.46s	0.00%	30	0.01% ██████████
a482_organizations	query	72ms	80ms	94ms	0.00%	27	0.01% ██████████
8c79_me	query	5ms	6ms	373ms	0.00%	27	0.01% ██████████
2c7b_organization	query	42ms	53ms	72ms	0.00%	24	0.01% ██████████
79f3_projectsWithTargets	query	96ms	130ms	469ms	0.00%	24	0.01% ██████████
08f3_SchemaView_SchemaExplorer	query	201ms	246ms	330ms	0.00%	17	0.01% ██████████
e652_SchemaExplorerFilter_AllTypes	query	82ms	87ms	90ms	0.00%	16	0.01% ██████████



Versions


9b6a6aed5e70ad85f506e4d6f54490a9f09b1b2e

Hive API

 Published 2d ago


6b18ec41f357ef3323b90442f2a08d3a573021bf

Hive API

 Published 5d ago


6540155fc76f487a1dd6db791d9f3800b6de216c

Hive API

 Published 13d ago


8cd7a4a885ad79a8f7968ab5b8609a9ee5abec90

Hive API

 Published 24d ago


43f6817e1fd8a0ddf5e5680512474be77ddf2dbe

Hive API

 Published 51d ago


9e90b1a53de59633884a4b792d8f7e0f06d26db4

Hive API

 Published 61d ago


1c2586f7a7f24ebb86ec2cf9c37dd21b836dbe0b

Hive API

 Published 69d ago

05890fa347bc80da6b8fd662975b3ab7314ecda8

Hive API

 Published 69d ago

Schema



Breaking Changes

- Field `resetInviteCode` was removed from object type `Mutation`
- Field `inviteCode` was removed from object type `Organization`

Safe Changes

- Type `DeleteOrganizationInvitationError` was added
- Type `DeleteOrganizationInvitationInput` was added
- Type `DeleteOrganizationInvitationResult` was added
- Type `InviteToOrganizationByEmailError` was added
- Type `InviteToOrganizationByEmailInput` was added
- Type `InviteToOrganizationByEmailInputErrors` was added
- Type `InviteToOrganizationByEmailResult` was added
- Type `OrganizationInvitation` was added
- Type `OrganizationInvitationConnection` was added
- Field `deleteOrganizationInvitation` was added to object type `Mutation`
- Field `inviteToOrganizationByEmail` was added to object type `Mutation`
- Field `invitations` was added to object type `Organization`

HIVE

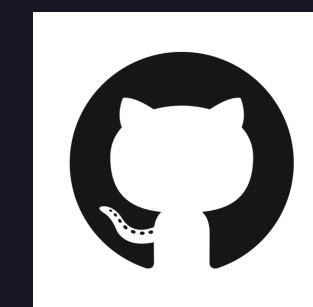
- Monitoring
- Schema Checks
- Schema high availability storage
- History of changes
- Integrated into your workflow (CI/DC, Notifications)





SUMMARY

- Don't think GraphQL, Think APIs
 - Leverage your existing services
 - Evolve gradually
 - Manage and track everything
 - Don't think GraphQL, Think APIs
-
- the-guild.dev



github.com/urigo



[@urigoldshtein](https://twitter.com/urigoldshtein)

GraphQL Meetup with Booking.com and Albert Heijn



Hosted By
Uri G. and GraphQL F.



GraphQL GATHERING EU

Hosted by the GraphQL Foundation

July 4th, Spielfeld, Berlin



GraphQLConf 2023

hosted by  GraphQL
Foundation

📅 September 19-21, 2023

🌐 San Francisco Bay Area, CA

[Join as a Sponsor](#)

[Join as a Partner](#)

Speakers



Lee Byron

Co-creator of GraphQL,
Director of the GraphQL
Foundation



Idit Levine

Founder of Solo.io



Matteo Collina

Creator of Fastify,
Platformatic CTO, Node TSC



Marc-André Giroux

Author of Production Ready
GraphQL, GraphQL TSC



Uri Goldshtein

Founder of The Guild,
GraphQL TSC



GOTO
Guide



Remember to
rate this session

THANK YOU!



Download on the
App Store



GET IT ON
Google Play

#GOTOams