

THE WONDERS OF REAL TIME SYSTEMS

THE FOSS FILES SEASON 4 | EPISODE 5

API ALCHEMY



🎙 Ameya Unchgaonkar

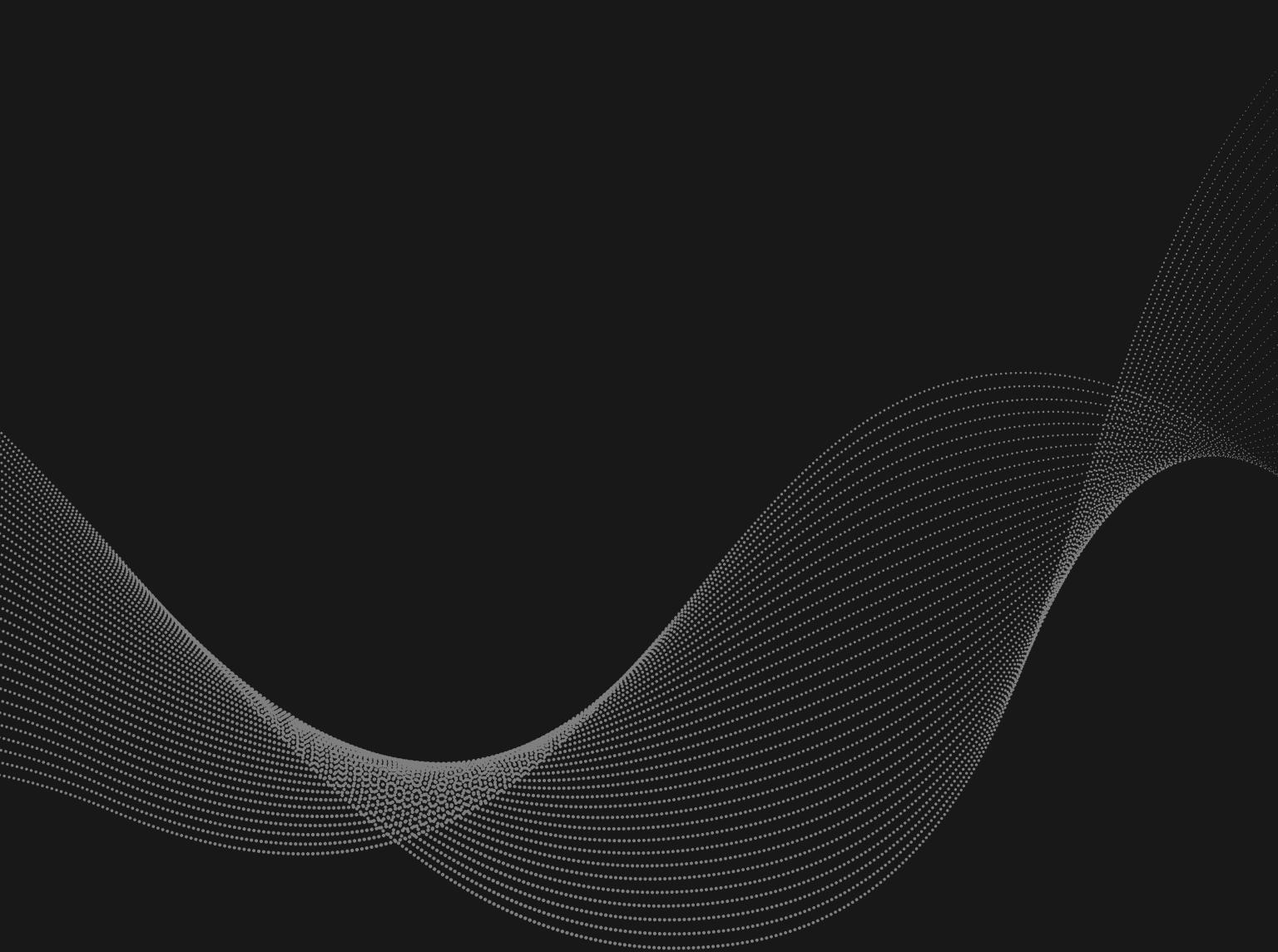
🎙 Vighnesh Potdar

🎙 Aditya Aparadh



13 July 2024

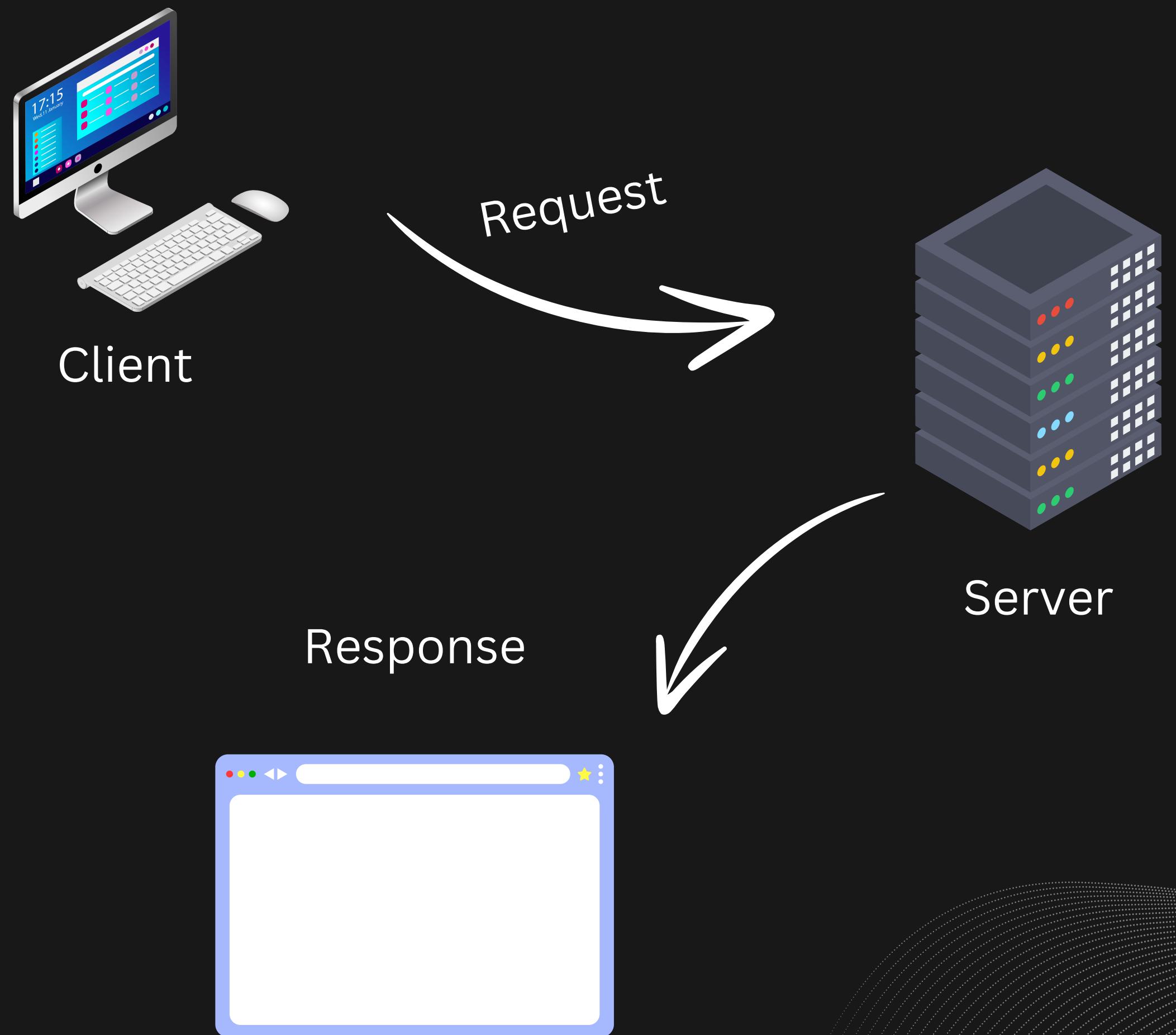
Table of Contents

- 
- 01 WebSockets
 - 02 Polling and other Architectures
 - 03 WebSocket Protocol
 - 04 WebRTC
 - 05 WebSockets and WebRTC
 - 06 Applications in Real-Time Systems

Introduction to WebSockets

- ❖ Web-sockets are bi-directional
- ❖ Lasts as long as any of the participant closes the connection.
- ❖ Makes use of HTTP to initiate connection
- ❖ Efficiency in Real time systems







WebSocket



**HTTP/2, REST, GraphQL, MQTT, AMQP,
SOAP, FTP**

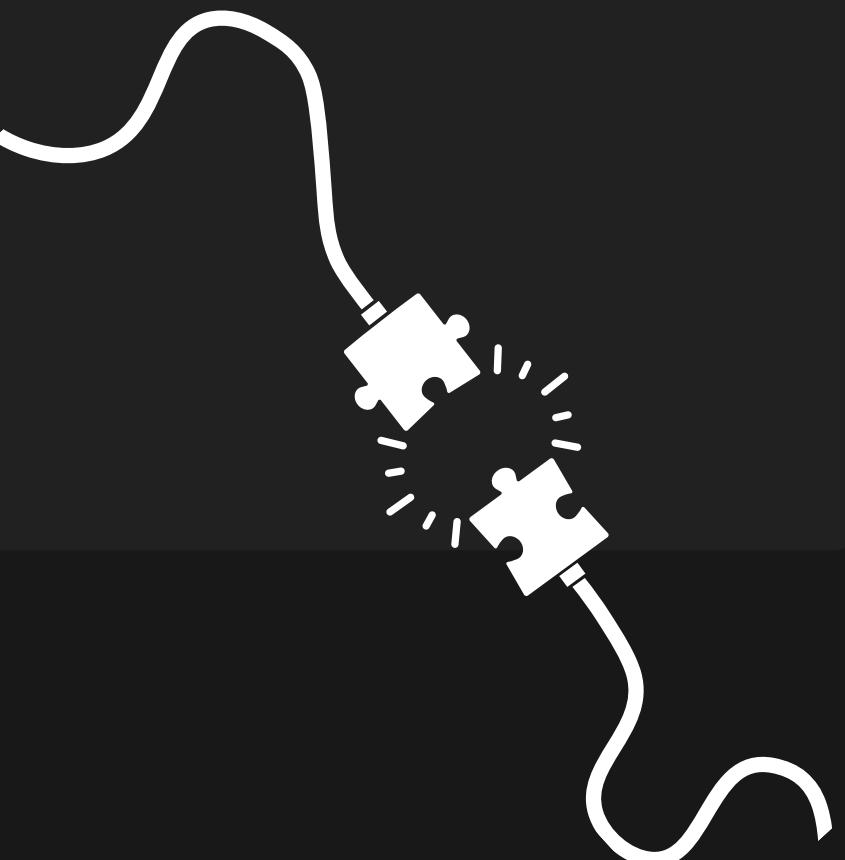
Where Websockets

- ✿ Stock exchange (for real time updates on ticker prices, order book, etc)
- ✿ Chat application (as Sending and receiving message should be quick)
- ✿ Web application where real time changes are required
- ✿ Gaming industry



How Web Sockets Work

- ✿ Connection Setup starts with a request initiated from the client for the handshake
- ✿ Upgrade Request in Header
- ✿ If Response is status **101** connection established



Request Header

GET /chat HTTP/1.1

Host:

server.example.com

Upgrade:

websocket

Connection:

Upgrade

Sec-WebSocket-Key:

dGhIHNhbXBsZSBub25jZQ==

Origin:

http://example.com

Sec-WebSocket-Protocol:

chat, superchat

Sec-WebSocket-Version:

13

Response Header

HTTP/1.1 101 Switching Protocols

Upgrade:

websocket

Connection:

Upgrade

Sec-WebSocket-Accept:

s3pPLMBiTxaQ9kYGzzhZRbK+xOo=

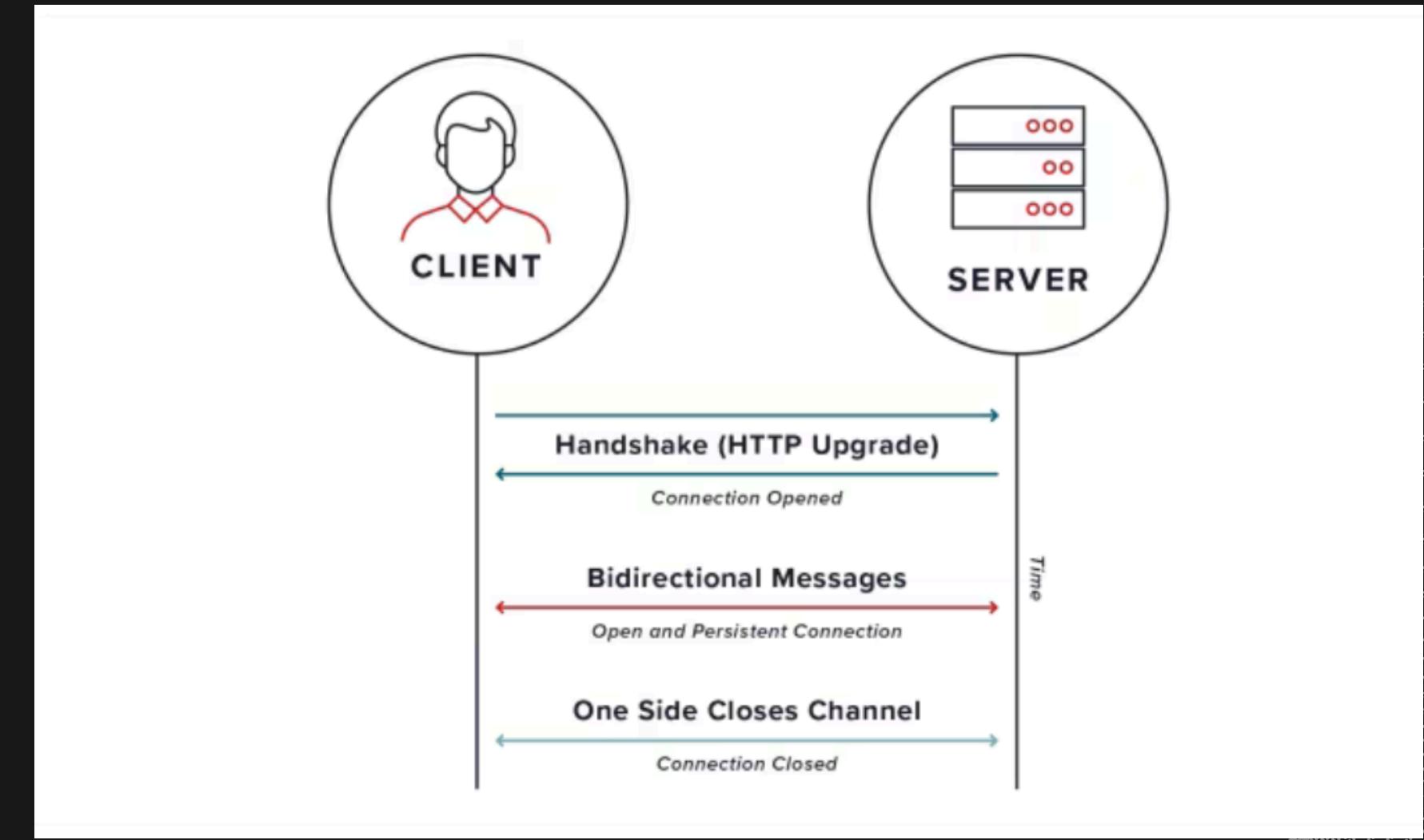
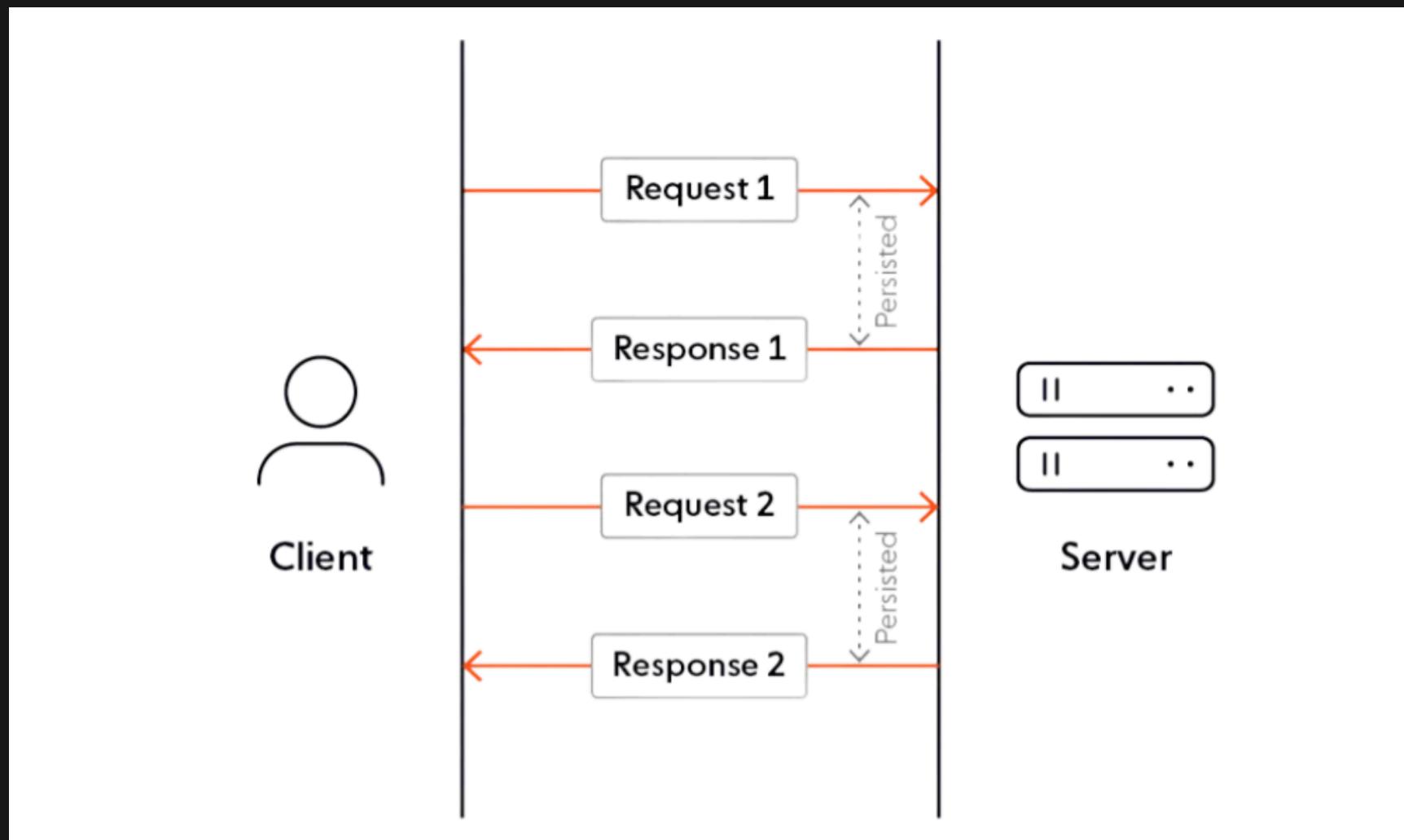
Sec-WebSocket-Protocol:

chat

Polling vs Web sockets

	Polling	WebSockets
Communication Method	Client repeatedly requests data at intervals.	Single, persistent connection for real-time communication.
Efficiency	High overhead due to repeated requests.	Low overhead with a single connection.
Scalability	More load on server, harder to scale.	Less load, easier to scale for many clients.

Polling vs Web sockets



Server During Polling



Socket io

- ❖ Real-Time Communication
- ❖ Event-Driven
- ❖ Cross-Browser Compatibility
- ❖ Automatic Reconnection



Code Example

```
io.of("/my-namespace").on("connection", (socket) => {
  socket
    .timeout(5000)
    .to("room1")
    .to(["room2", "room3"])
    .except("room4")
    .emit("hello", (err, responses) => {
      // ...
    });
});
```

WEBRTC

- ❖ What, Why and How
- ❖ 4 Pillars of WebRTC
- ❖ WebRTC vs WebSockets
- ❖ Applications of WebRTC



What is WebRTC

- ◆ WebRTC, short for Web Real-Time Communication, is both an API and a Protocol.
- ◆ It allows for peer-to-peer audio, video, and data sharing
- ◆ Widely used for building applications such as video conferencing, voice calling, live streaming, online gaming

Why use WebRTC

- ❖ Open Source
- ❖ Cross-Platform Compatibility
- ❖ No Plugins Needed
- ❖ Seamless Communication

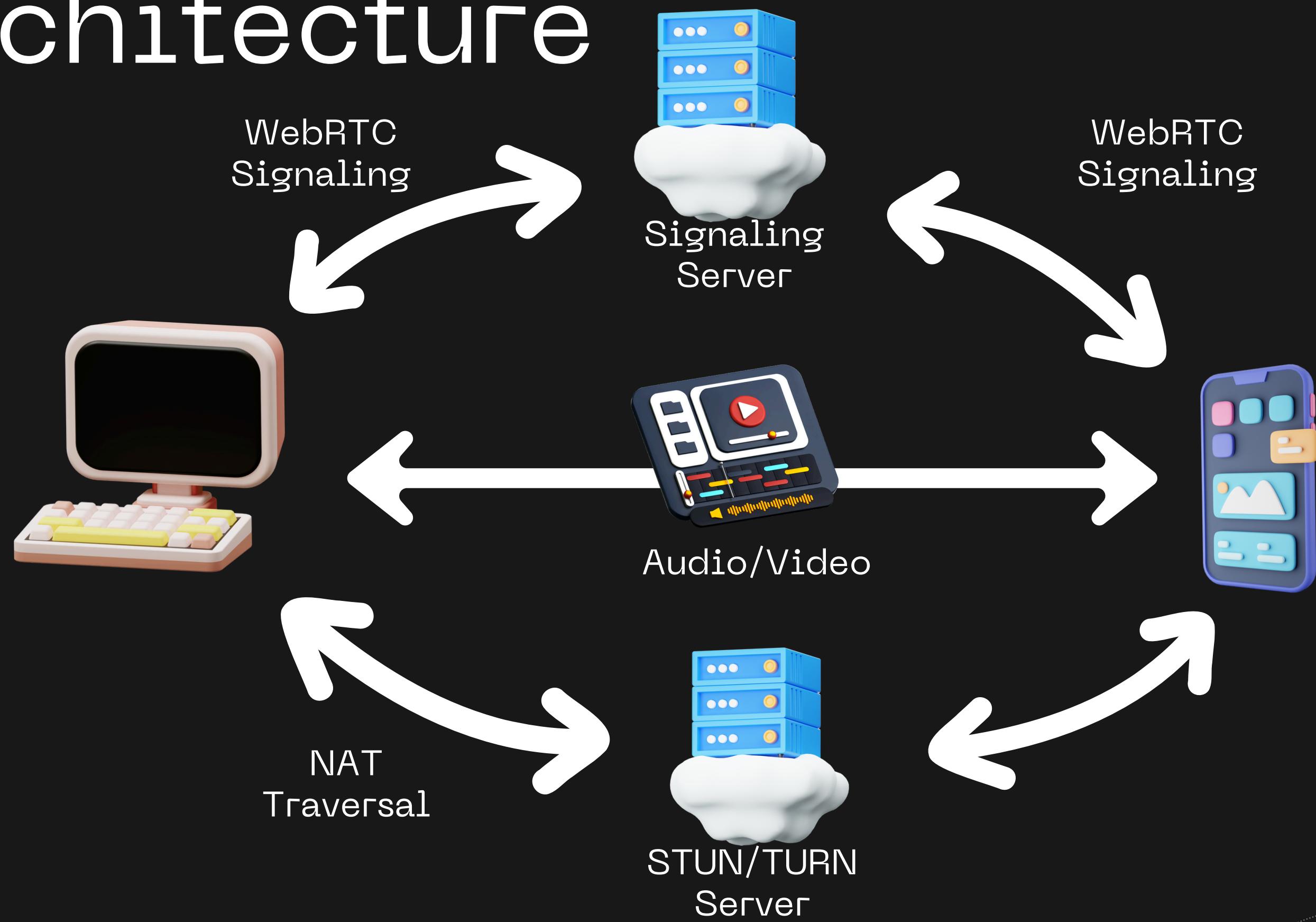


4 Pillars of WebRTC

- ❖ Signaling
- ❖ Connecting
- ❖ Securing
- ❖ Communicating



Architecture



WebSocket vs WebRTC

- ❖ Purpose and Use Cases
- ❖ Connection Model
- ❖ Security and Encryption
- ❖ Applications



Applications of WebRTC

- ◆ Cloud Gaming
- ◆ Video Conferencing
- ◆ Broadcasting
- ◆ Internet of Things



Practical Demo



Get Ready!



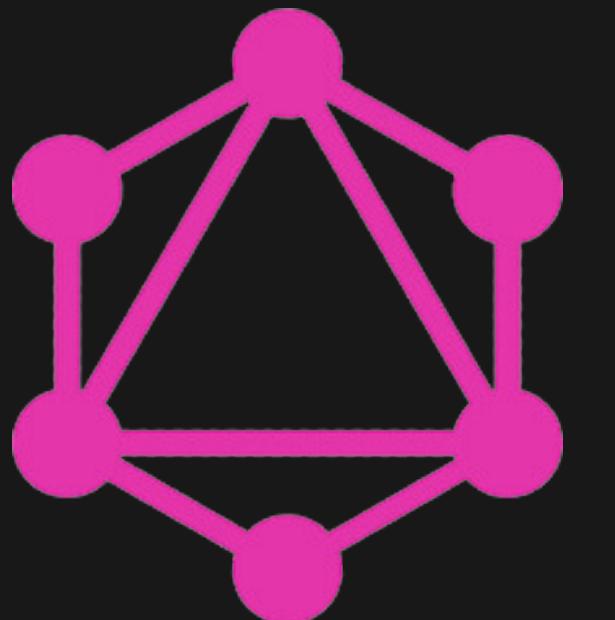


A large, stylized text graphic reading "QUIZ TIME". The word "QUIZ" is in orange with white outlines, and "TIME" is in blue with white outlines. Above the text are two speech bubbles: one white containing a red question mark and one dark grey containing a light blue exclamation mark.

quiz.wcewlug.org



CONCLUSION



THANK YOU

Community | Knowledge | Share

API ALCHEMY



🎙 Ameya Unchgaonkar

🎙 Aditya Aparadh

🎙 Vighnesh Potdar



13 July 2024