

UNCOVERING RPCS

THE FOSS FILES SEASON 4 | EPISODE 4

API ALCHEMY

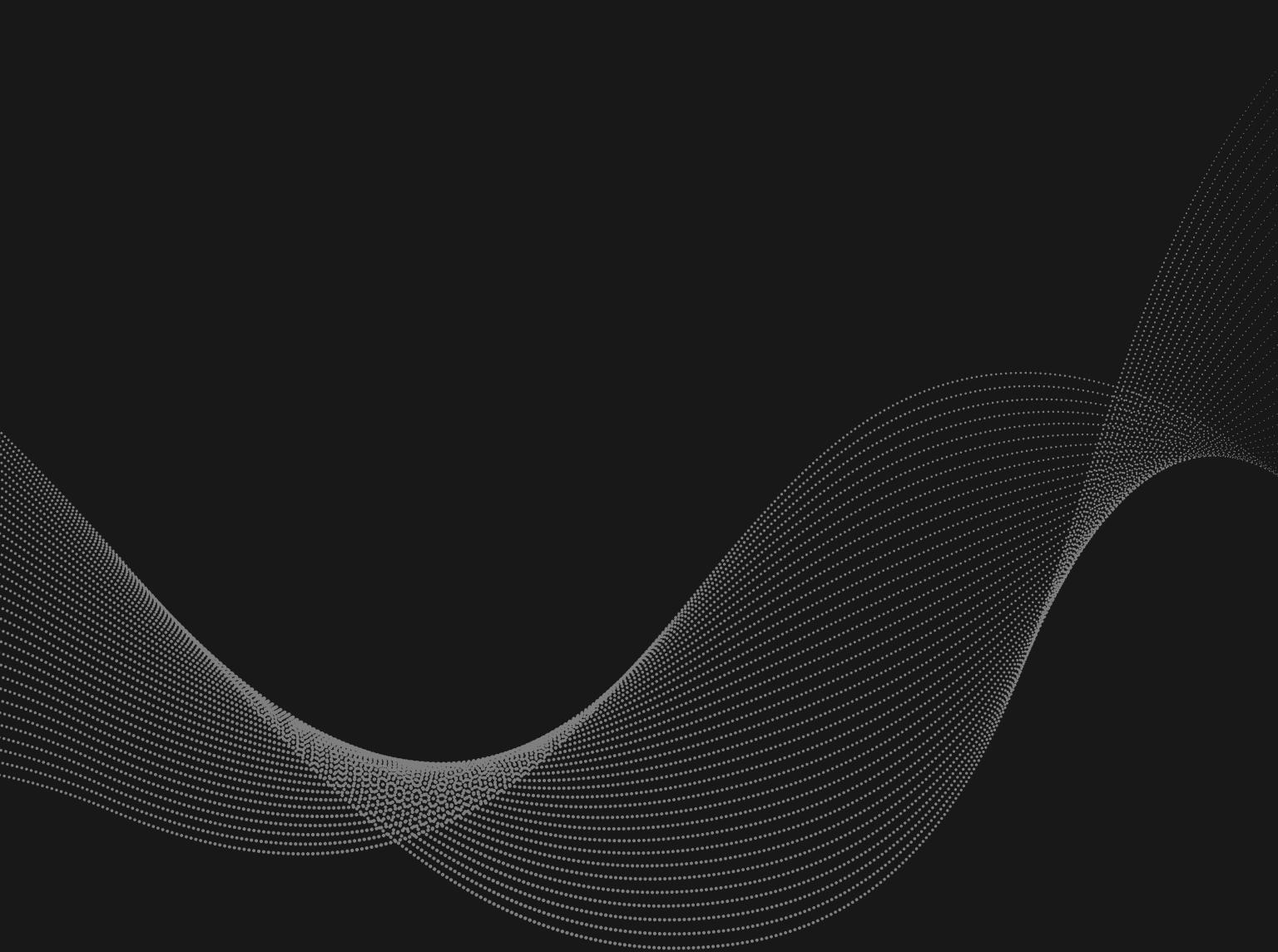


🎙 Pratiksha Hire

🎙 Aditya Aparadh

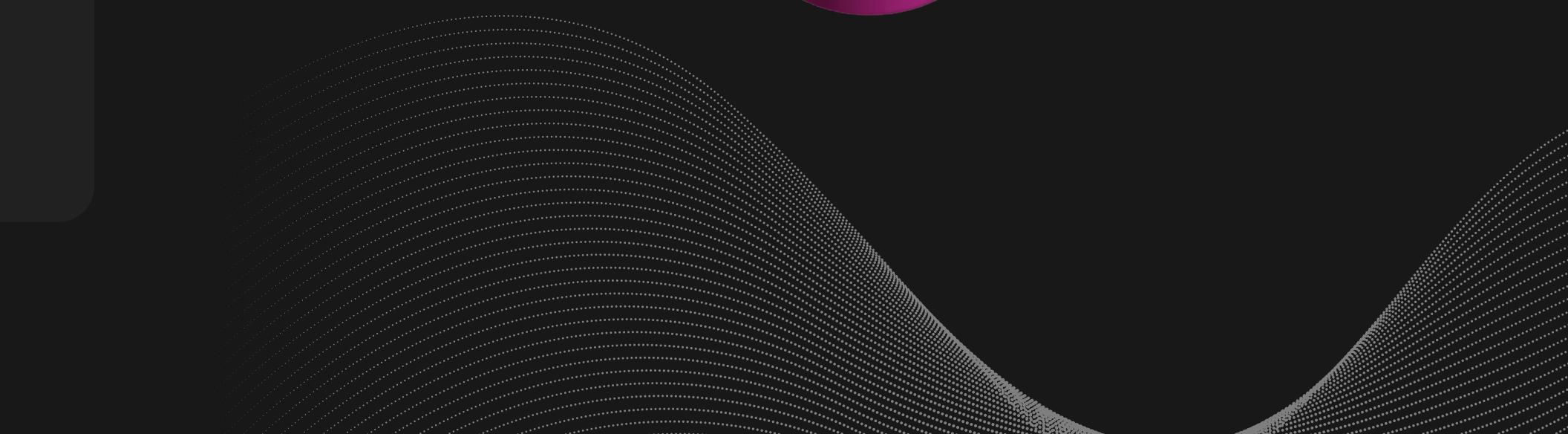
06 July 2024

Table of Contents

- 
- 01 Introduction to Remote Procedure Calls
 - 02 RPCs Vs REST
 - 03 HTTP - RPC
 - 04 gRPC and Protocol Buffers
 - 05 Comparison between RPC frameworks

Let's revise GraphQL

- ❖ What were drawbacks of REST, overcame by GraphQL?
- ❖ What is a Schema?
- ❖ What is used to modify data on the server?
- ❖ What is Apollo?



IPC

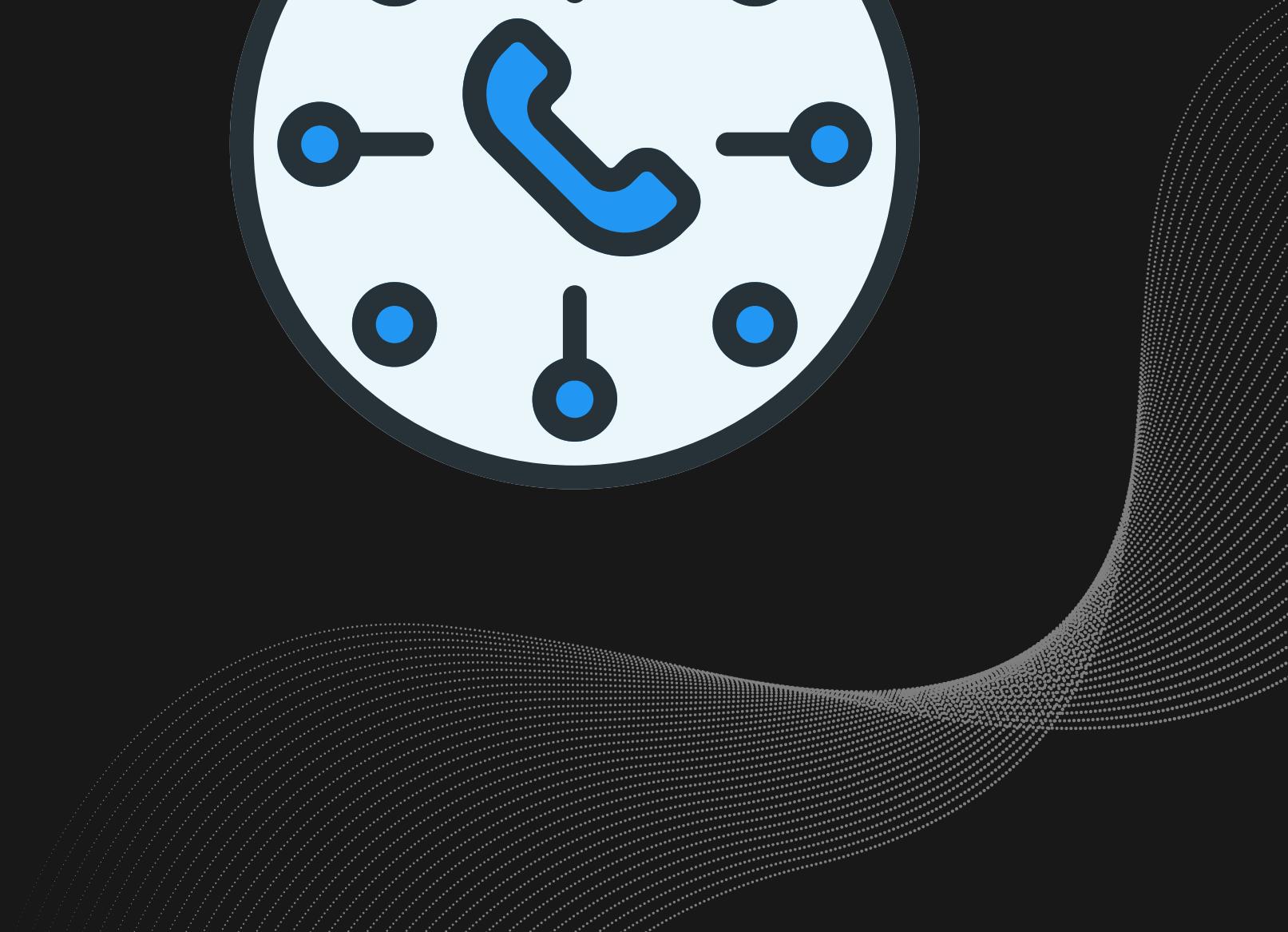


RPC



Remote Procedure Call:

- ✿ Communication protocol
- ✿ Allows program to execute a remote procedure as if it were local
- ✿ Hides the complexity of a network call
- ✿ Used in distributed systems



Authentication Service

```
func login(email, password) {  
    -----  
    -----  
    -----  
    notification_otp(email);  
}
```

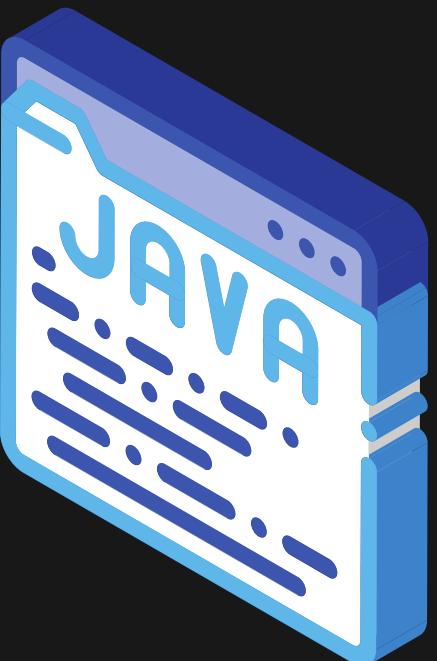


Notification Service

Stubs:

Authentication
Service
Client Stub

Notification
Service
Server Stub



RPCs VS REST

- ❖ Action-oriented
- ❖ Use HTTP, TCP, UDP
- ❖ Binary as well as textual data format
- ❖ Statefull as well as stateless interactions
- ❖ Resource-oriented
- ❖ Use HTTP / HTTPS
- ❖ Textual data format
- ❖ Stateless interactions

HTTP-RPCs



JSON-RPC

- ❖ Stateless, lightweight protocol
- ❖ Uses JSON as data format
- ❖ Request object:
 - jsonrpc
 - method
 - params
 - id



JSON-RPC

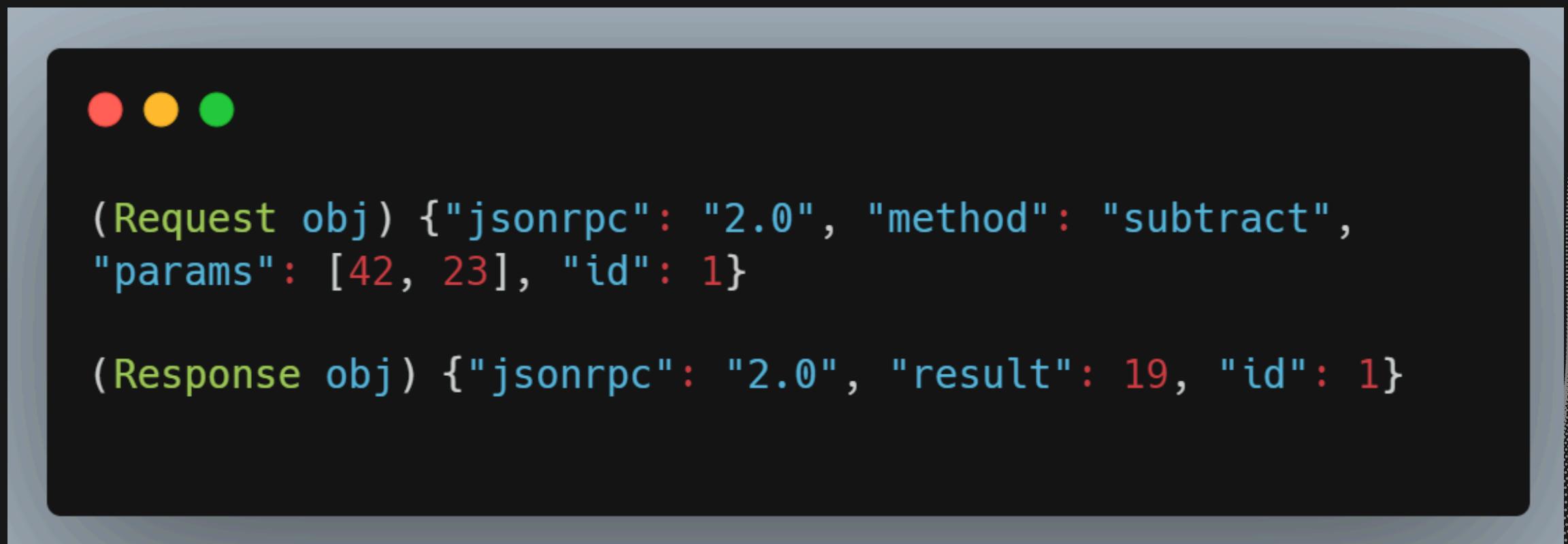
- ❖ Notification: Request object, needs no response
- ❖ Response object:
 - jsonrpc
 - result
 - error
 - id



JSON-RPC

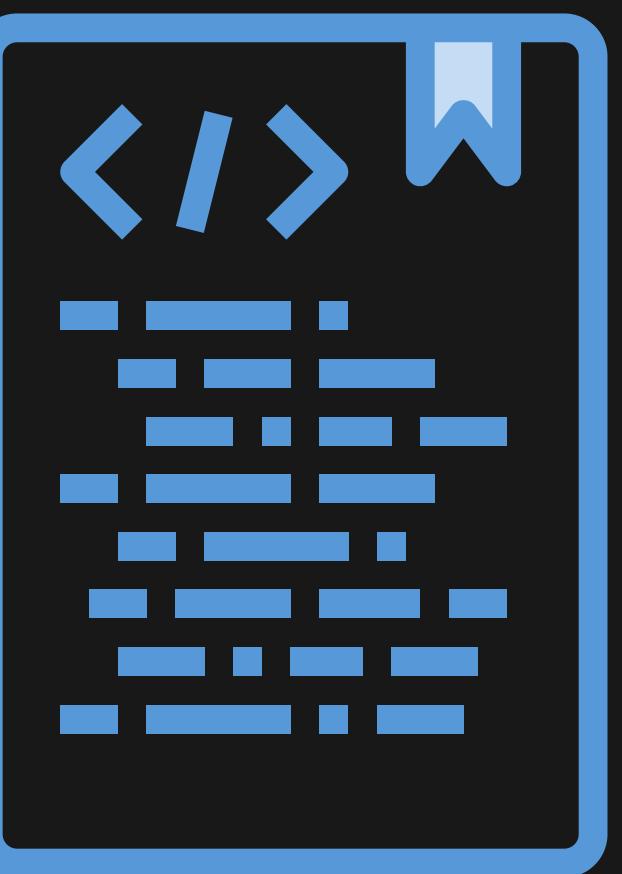
❖ Error object:

- code
- message
- data

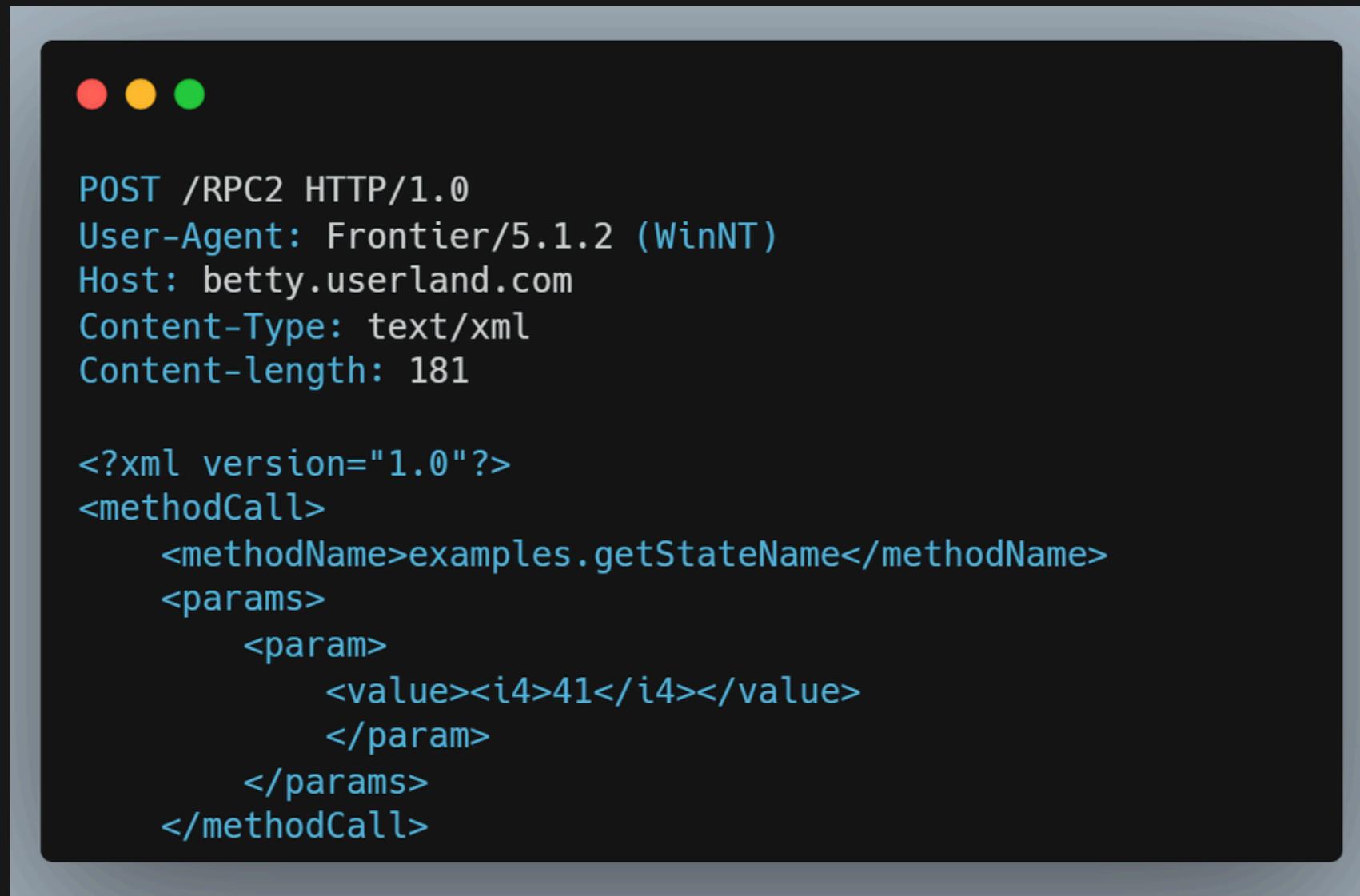


XML-RPC

- ❖ Data format is XML
- ❖ XML RPC message is an HTTP POST request



XML-RPC Request

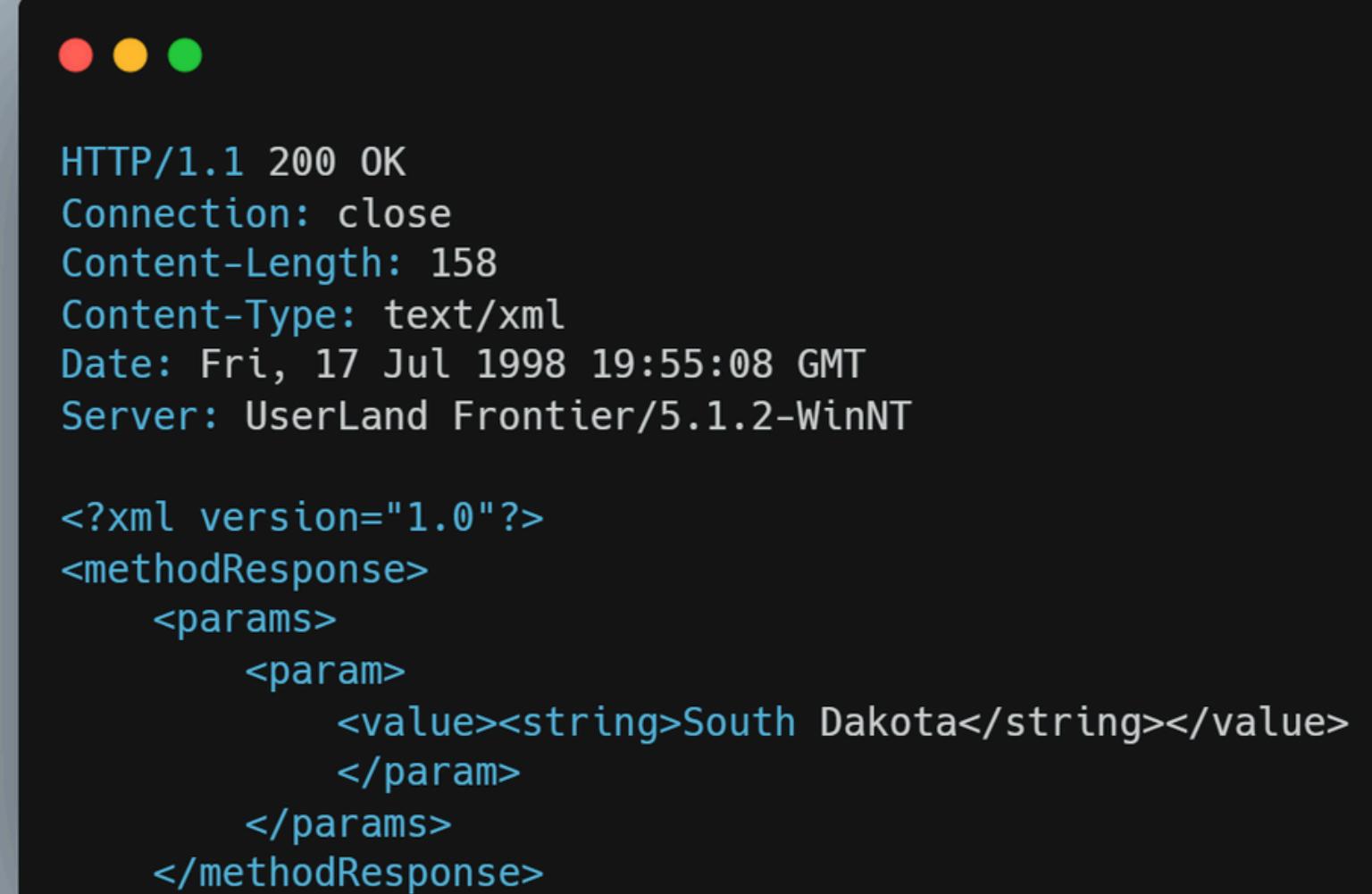


The image shows a terminal window with a dark background. In the top-left corner, there are three small colored circles: red, yellow, and green. The terminal output is as follows:

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: betty.userland.com
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value><i4>41</i4></value>
    </param>
  </params>
</methodCall>
```

XML-RPC Response



HTTP/1.1 200 OK
Connection: close
Content-Length: 158
Content-Type: text/xml
Date: Fri, 17 Jul 1998 19:55:08 GMT
Server: UserLand Frontier/5.1.2-WinNT

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>South Dakota</string></value>
    </param>
  </params>
</methodResponse>
```

gRPC

- ★ Introduced by Google in 2015
- ★ Uses Protocol Buffers as IDL and serialization format
- ★ Primarily operates over HTTP/2
- ★ Language agnostic



gRPC

The gRPC logo consists of the lowercase letters "gRPC" in a white, sans-serif font. A horizontal double-headed teal arrow is positioned above the letter "R".

Protocol Buffers

- ❖ Serialization alternative to JSON / XML
- ❖ Language and platform neutral
- ❖ Efficient binary serialization
- ❖ Use the **ProtoC** compiler to generate source code in language of choice



gRPC

The gRPC logo consists of the lowercase letter 'g' followed by the uppercase letters 'RPC'. A teal double-headed horizontal arrow is positioned above the 'g'.

ProtoBuf



```
message Person {  
    optional string name = 1;  
    optional int32 id = 2;  
    optional string email = 3;  
}
```

Sample Protocol Buffer file

ProtoBuf



```
Person john = Person.newBuilder()
    .setId(1234)
    .setName("John Doe")
    .setEmail("jdoe@example.com")
    .build();
output = new FileOutputStream(args[0]);
john.writeTo(output)
```

Java Server Code



```
Person john;
fstream input(argv[1], ios::in | ios::binary);
john.ParseFromIstream(&input);
int id = john.id();
std::string name = john.name();
std::string email = john.email();
```

C++ Client Code

gRPC



```
Person john = Person.newBuilder()
    .setId(1234)
    .setName("John Doe")
    .setEmail("jdoe@example.com")
    .build();
output = new FileOutputStream(args[0]);
john.writeTo(output)
```

Java Server Code



```
Person john;
fstream input(argv[1], ios::in | ios::binary);
john.ParseFromIstream(&input);
int id = john.id();
std::string name = john.name();
std::string email = john.email();
```

C++ Client Code

Comparing RPC frameworks

JSON RPC

- ❖ Uses JSON
- ❖ Less efficient
- ❖ Low Complexity

XML RPC

- ❖ Uses XML
- ❖ Less efficient
- ❖ Low Complexity

gRPC

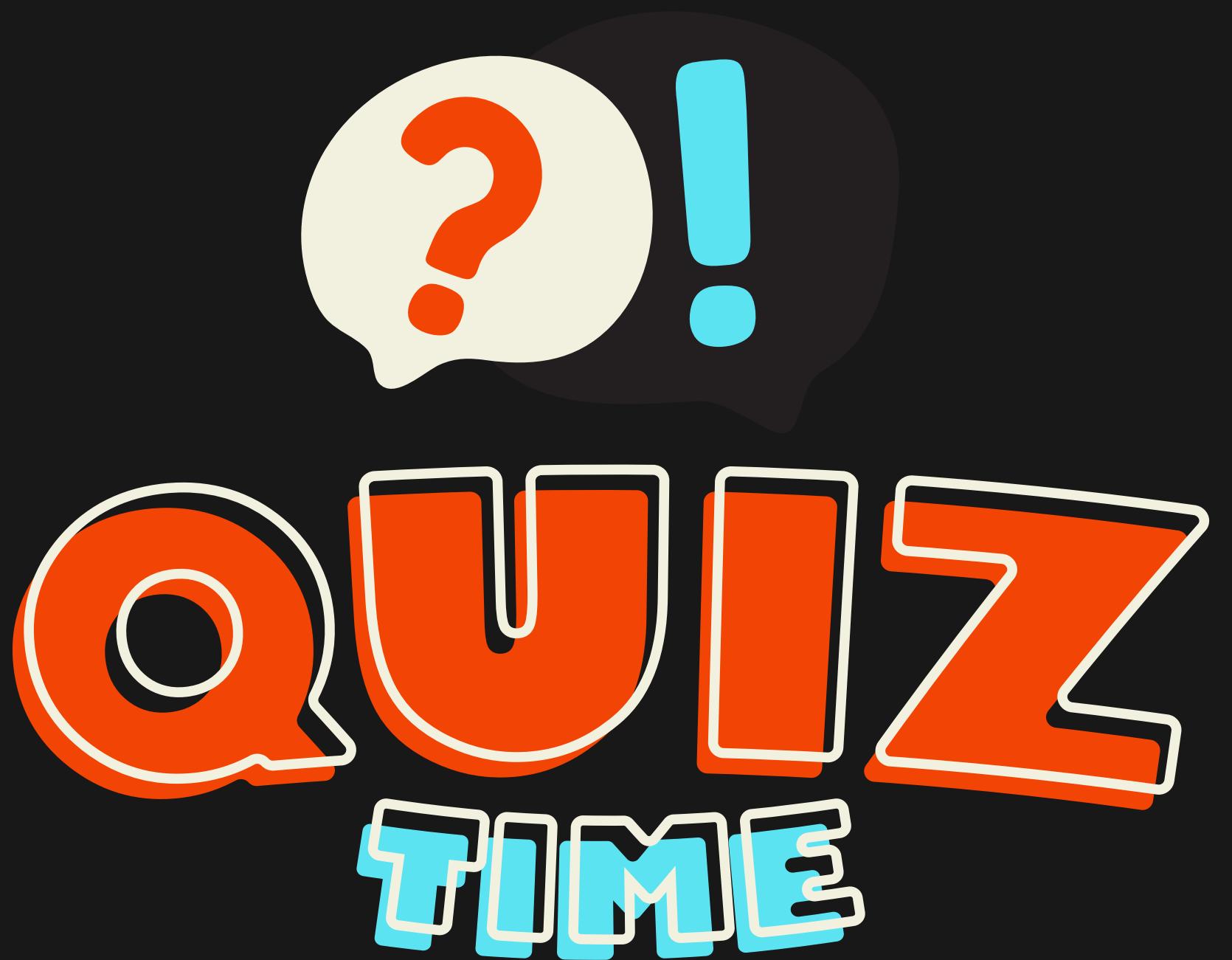
- ❖ Uses Protobufs
- ❖ More efficient
- ❖ High Complexity

gRPC Modes

- ❖ Unary RPC
- ❖ Server Streaming RPC
- ❖ Client Streaming RPC
- ❖ Bidirectional Streaming RPC

The gRPC logo consists of the lowercase letters "gRPC" in white. A teal double-headed horizontal arrow is positioned above the letter "R".

Get Ready!



THANK YOU

Community | Knowledge | Share

API ALCHEMY



🎙 Pratiksha Hire

🎙 Aditya Aparadh



06 July 2024