



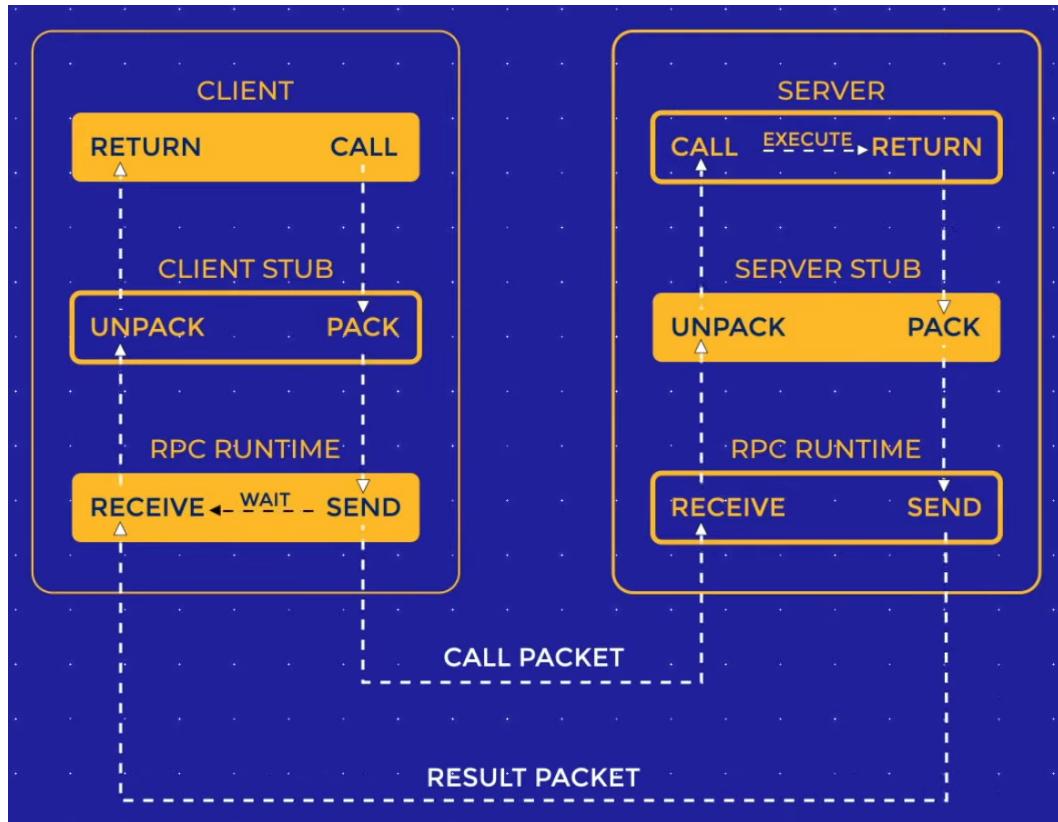
Types of APIs

- An application programming interface is a way for two or more computer programs to communicate with each other. It is a type of software interface, offering a service to other pieces of software.
- A document or standard that describes how to build or use such a connection or interface is called an API specification.

```
flowchart TB
main(API) --> rpc([RPC])
    rpc --> xml[XML-RPC]
    rpc --> jsonrpc[JSON-RPC]
    rpc --> grpc[gRPC]
main --> soap([SOAP])
main --> rest([REST])
main --> graphql([GraphQL])
```

RPC (Remote Procedure Call)

- A simple straightforward interaction between a local client and a server.
- Fast as it has short light-weight messages.
- Client sends a call to server which is unpacked using a server stub which is then processed and sent back to client where it is then unpacked using the client stub.



Flow of an RPC Call from a client to a server

XML-RPC

- Had a con: Couldn't distinguish between different datatypes (a feature absent in XML) so developers have to add an additional field or metadata to specify the data type.
- This problem was later solved by SOAP.

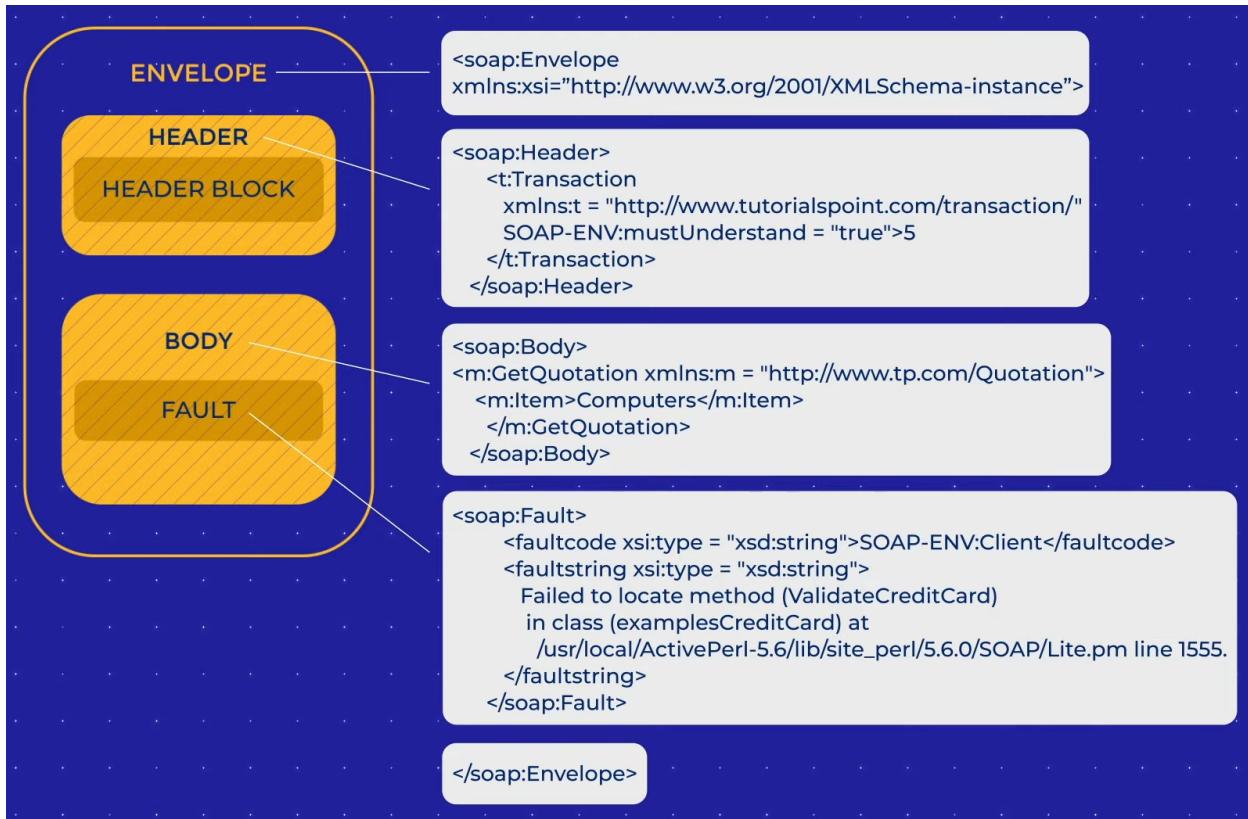
gRPC (Google RPC)

- Used by Apache Thrift and Twirp for internal microservice communication.
- High Performance API to optimize the network layer.
- Perfect for microservices as they need to make extensive amount of calls per day.

SOAP (Simple Object Access Protocol)

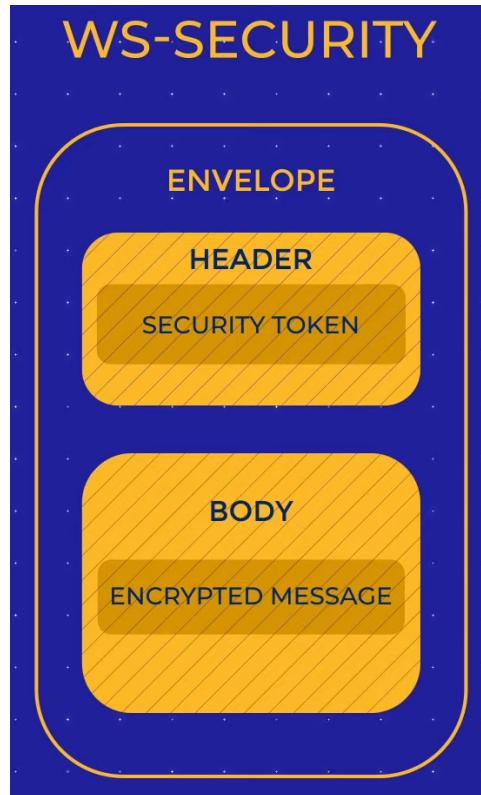
- Used in XML.

- Launched by Microsoft to curb the issues of XML-RPC.
- Mostly used by Financial Services and Corporate Systems like Salesforce
- Declared using an envelope tag, header tag is optional and give relevant data about the application, body tag contains either the request message or the response message and has an error scenario response built in, declared by the fault tag.



Format of an SOAP API Call

- Not Liked by Developers
- Highly Secure as Encrypts the message using a key which can only be decrypted by the recipient



SOAP Security using WS-Extension

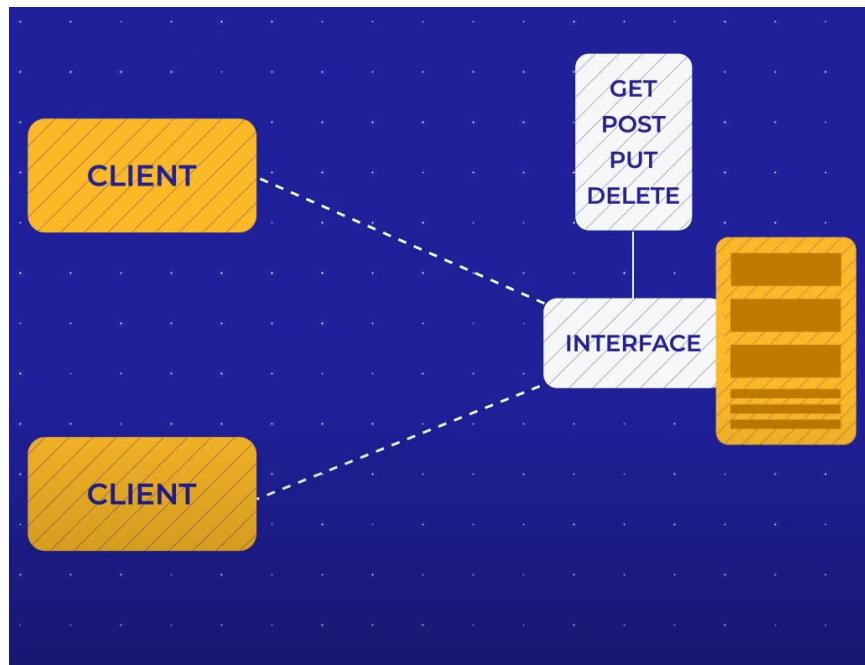
- Allows chaining and thus keeps the server aware of the previous requests
- Storing all the data of the calls puts a load on the server but can be essential for some services
- If you need to update the message properties you need to make changes to both server and client.
- Used long messages and was thus verbose

REST (Representational State Transfer)

- Uniform, Lightweight, Flexible, Fast and Efficient
- Replaced SOAP.

RPC/SOAP	REST
<ul style="list-style-type: none"> • RPC (POST, GET), SOAP (POST) • SendUserMessage GetUnits DeleteUser LocateVehicle addEntry ... 	<ul style="list-style-type: none"> • GET PUT POST DELETE

- Servers and Clients can have different internals and just need a common interface to make them work together.



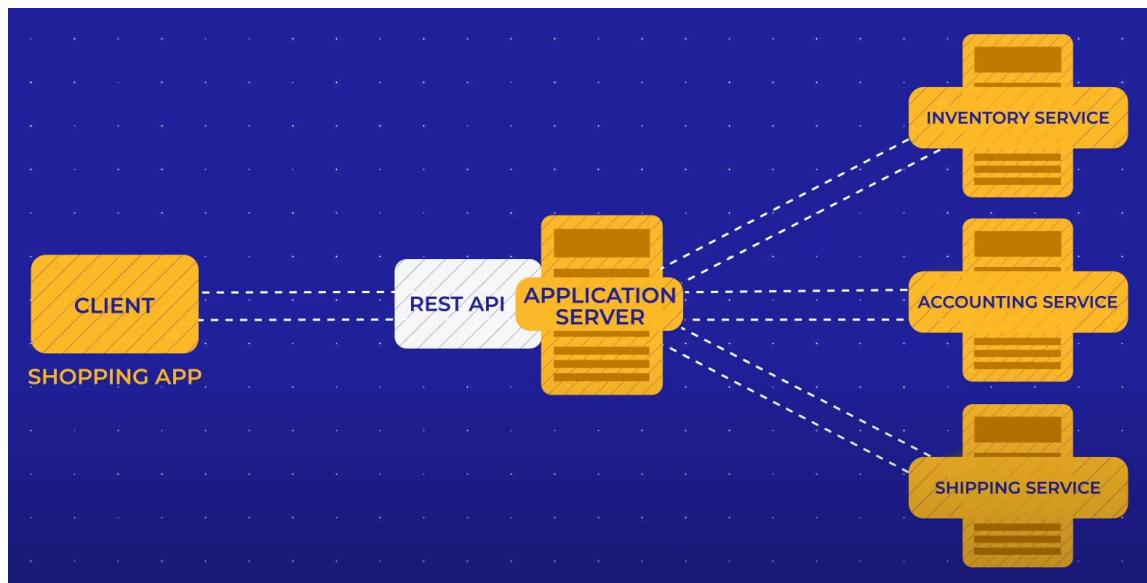
It has the following features:



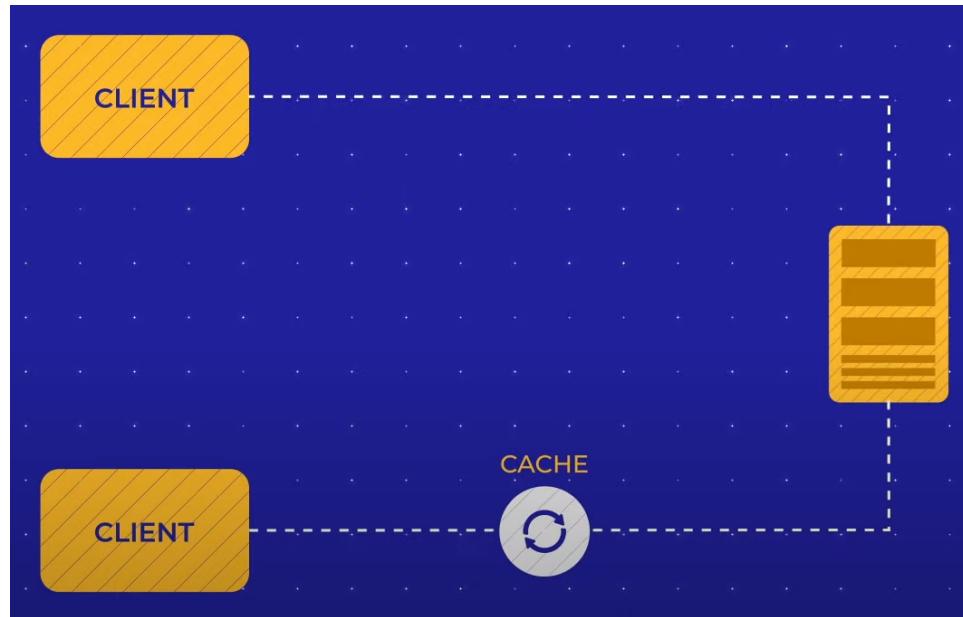
- Uniform Interface
 - Uses HTTP Methods:
 - POST: To Create the Data
 - GET: To Retrieve the Data
 - PUT: TO Update the Data
 - DELETE: To Erase the Data

METHOD	PATH	ACTION
GET	/photos	INDEX
GET	/photos/new	NEW
POST	/photos	CREATE
GET	/photos/:id	SHOW
GET	/photos/:id/edit	EDIT
PUT	/photos/:id	UPDATE
DELETE	/photos/:id	ERASE

- Layered System Architecture
 - The CLIENT doesn't know if the server receiving its request is the intermediary or the end server which allows the intermediary server to hide other servers behind it. This increases the scalability on the server side as the client doesn't need to know what's going in the background and only knows how to make a call.



- Stateless Interactions
 - The server treats each request as new and doesn't store the previous requests which is opposite to what happens in SOAP.
- Caching
 - REST uses HTTP Caching so the clients can retain content and reduce the load on the server

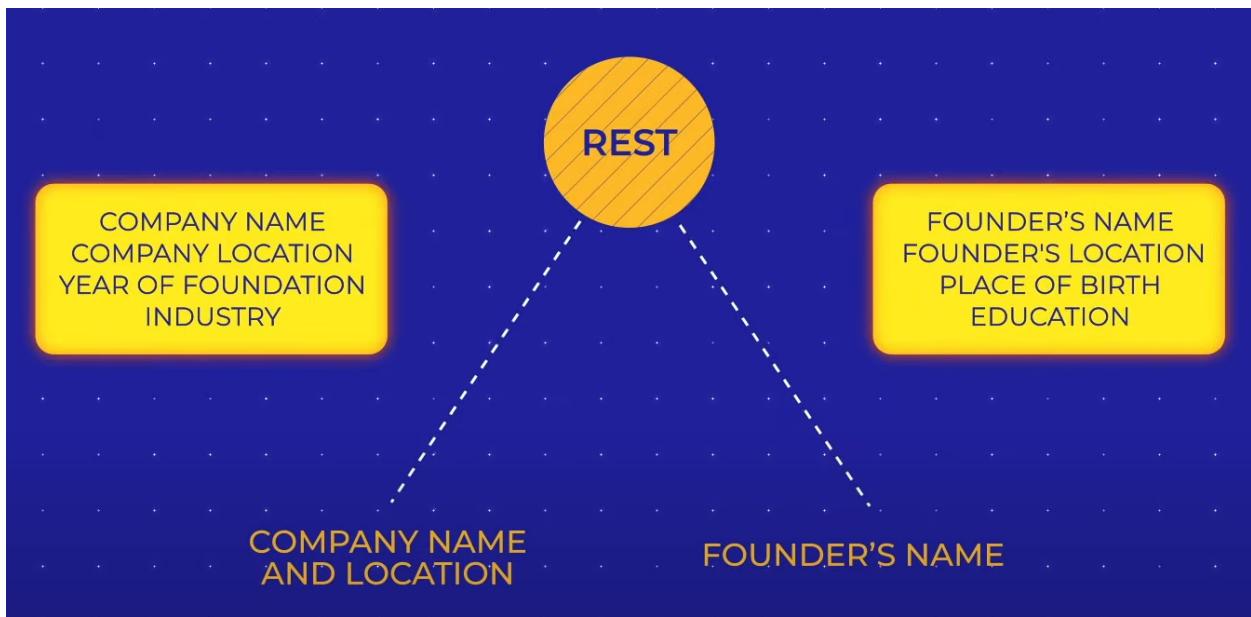


- Code on Demand
 - Supports feature such as this:

Q: What if I'm more paranoid than your regular user?

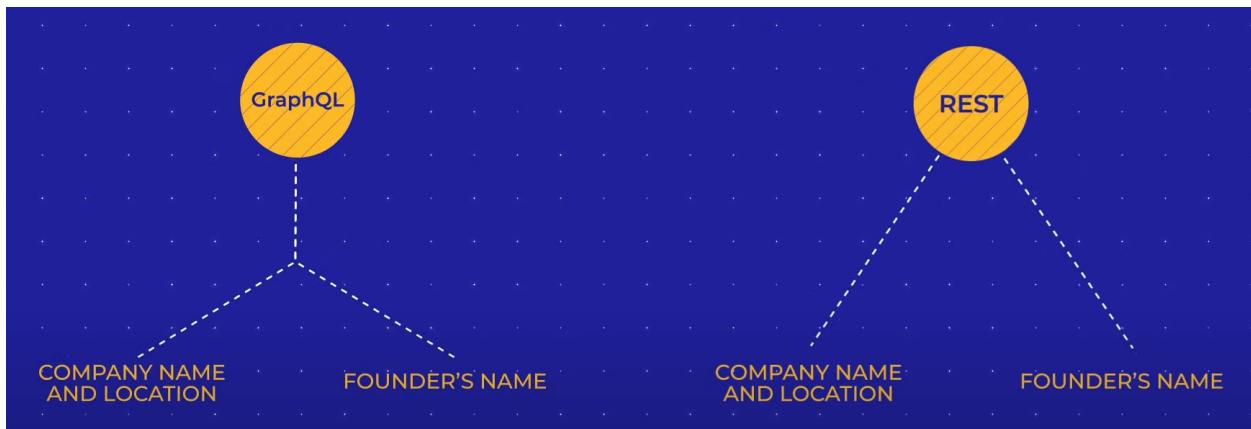
We've got you covered. Telegram's special [secret chats](#) use end-to-end encryption, leave no trace on our servers, support self-destructing messages and don't allow forwarding. On top of this, secret chats are not part of the Telegram cloud and can only be accessed on their devices of origin.

- The Client can make a call to the server to send a Encryption Key Generator Code to the REST API which the gets executed by REST on the client side, this way the server is not aware of the Key being generated, this is called Code on Demand.
- Con: REST APIs fetch all the data which might be unnecessary which can lead to use of extra time and resources

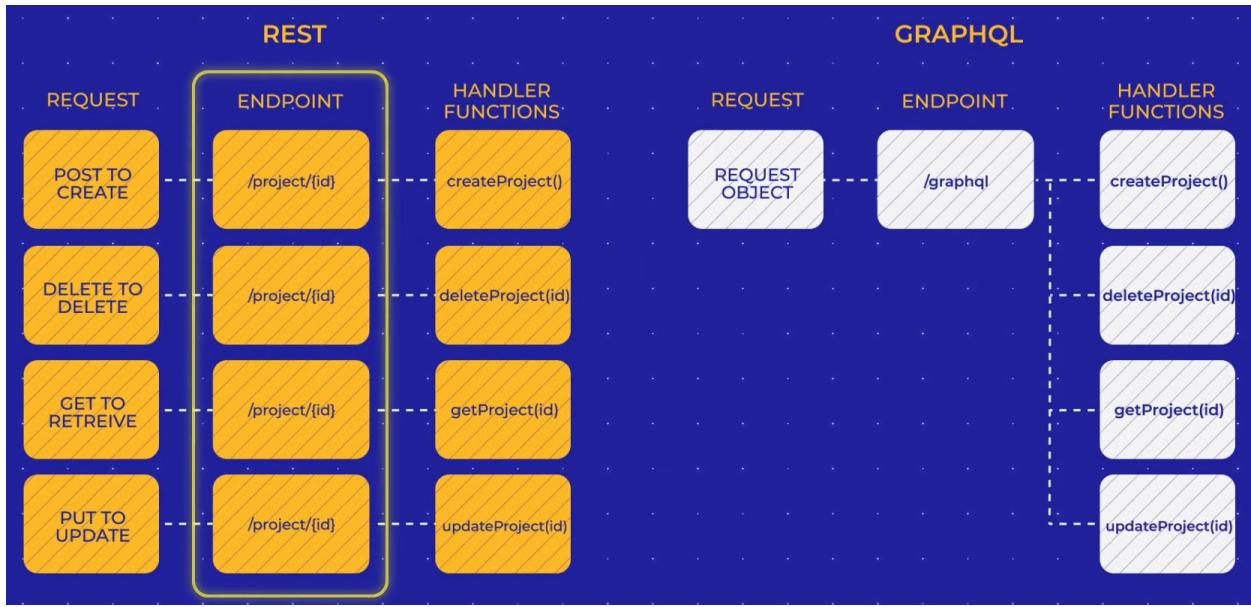


GraphQL (Graph Query Language)

- Created by Facebook in 2015 to prevent over-fetching done by REST APIs
- GraphQL makes a single precise request to fetch the exact data needed



- In REST there are multiple endpoints which fetch different data which then needs to be put together by the user but in GraphQL you only need to make request to one endpoint which fetches the compiled data, this is done by using a SCHEMA a feature absent from REST.



- SCHEMA provides the Client with a description of how the data is structured on the server, using the SCHEMA the Client can formulate the request to fetch the exact data it needs.
- This method greatly optimizes the payload, which is why it is the go-to solution for mobile APIs
- Learning Curve for GraphQL is Steeper than that of REST.

Event-Driven-Architecture (EDA)

- An EDA is a type of communication between a client and a server where instead of sending a request and fetching the data we instead instruct the server to send us a response whenever the state of a certain data changes.
- Eg: Request the server to send a response whenever the Courier Package reaches a Major Courier Hub.
- This way we don't have to constantly ping the server about the whereabouts of the package, putting unnecessary load on the server and can still get the requested data whenever the location of the package is updated.