Web APIs

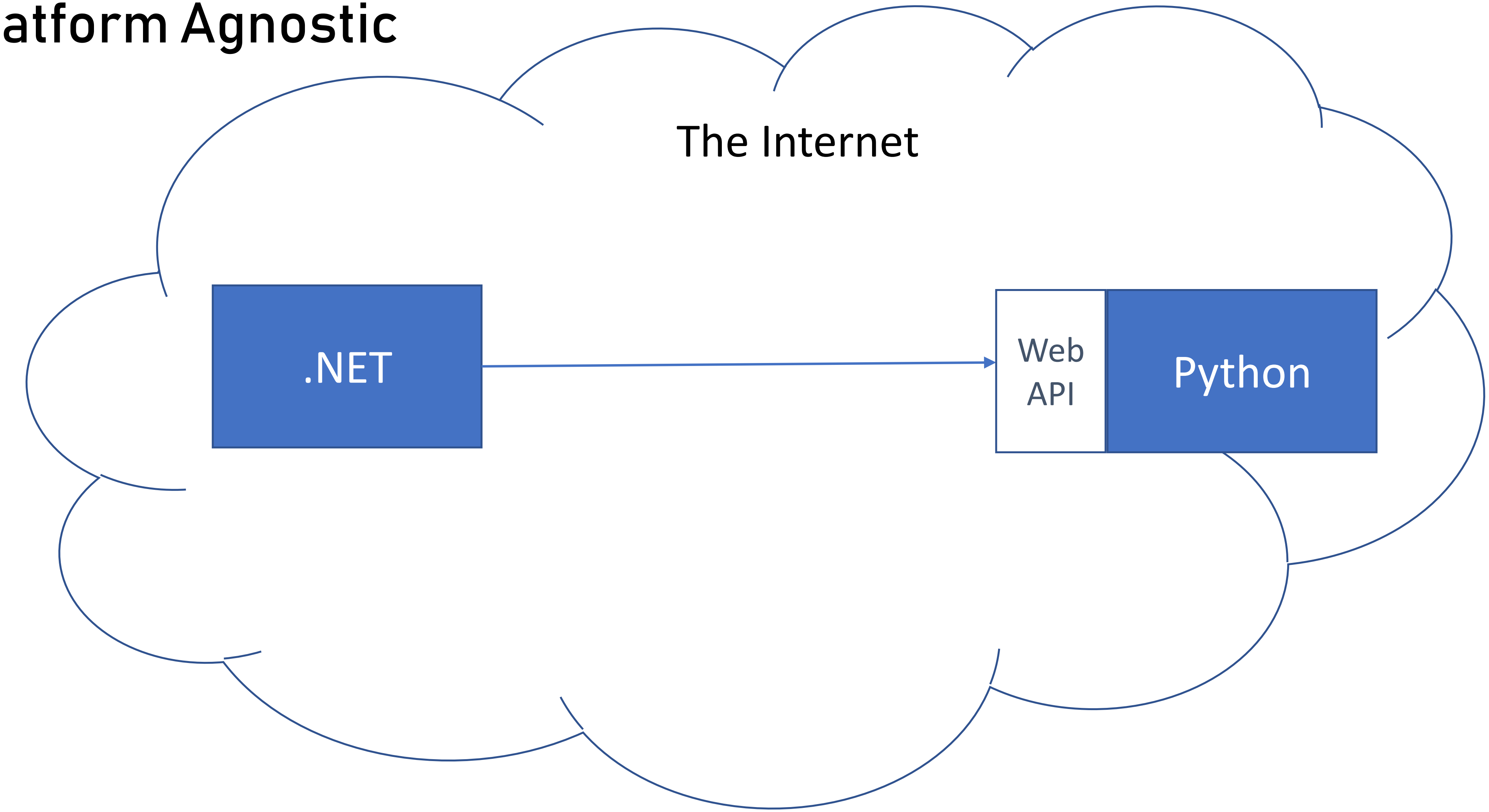
Memi Lavi
www.memilavi.com

Web API:

> *An API exposed by a web component,*
>
> *allowing other components to interact with it*

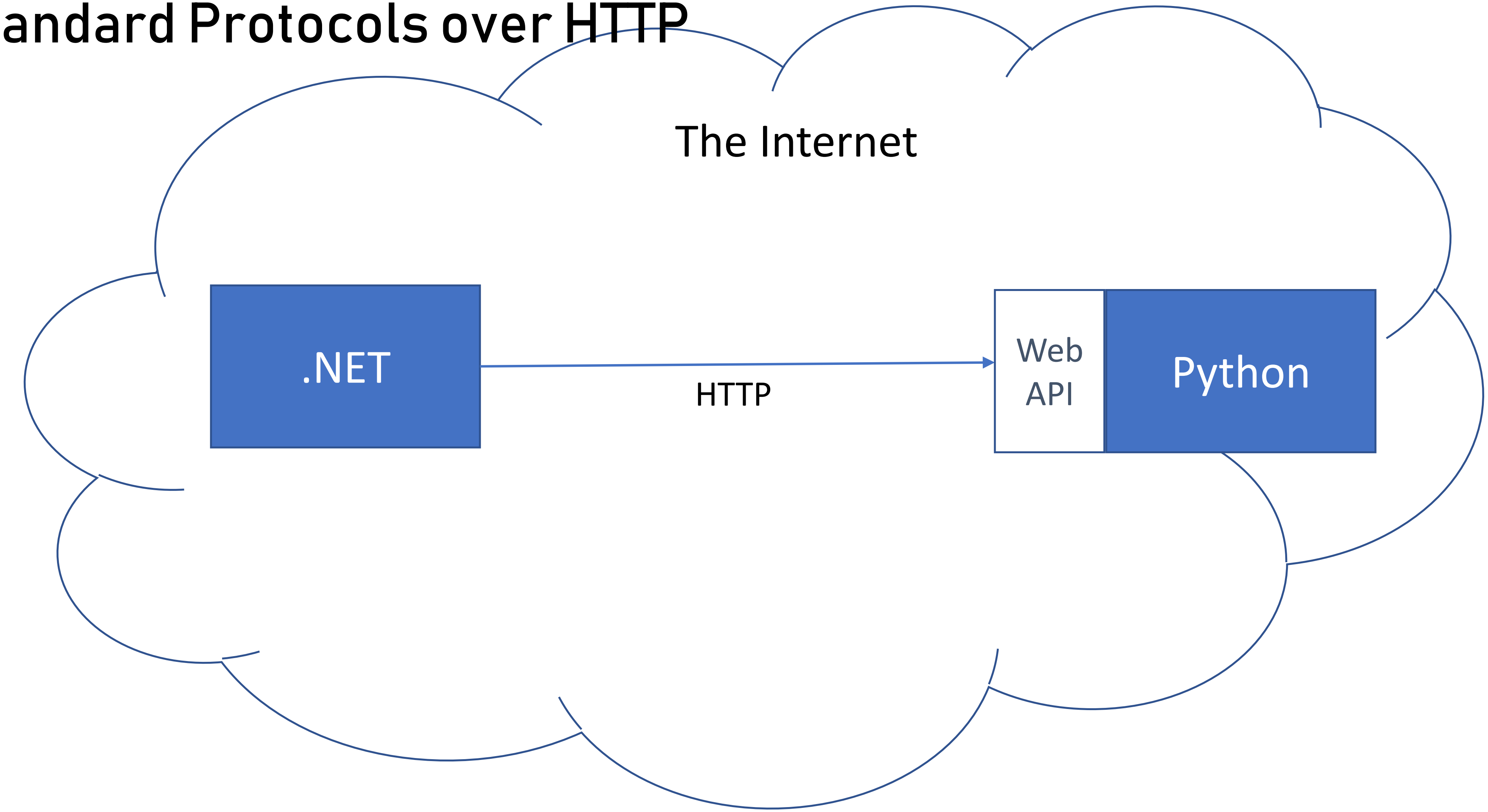
Common Characteristics of Web API

- Platform Agnostic



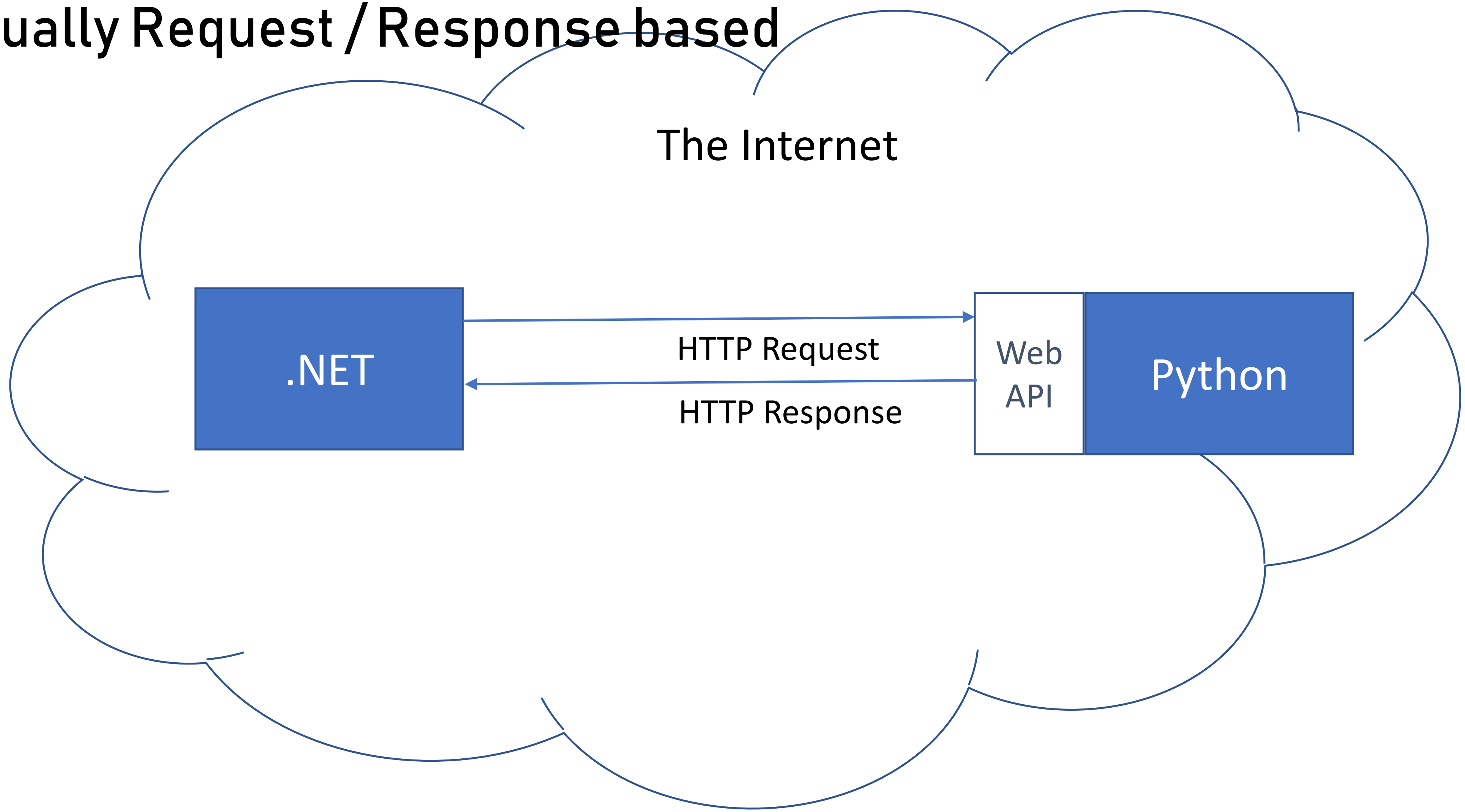
Common Characteristics of Web API

- Standard Protocols over HTTP



Common Characteristics of Web API

- Usually Request / Response based



Web API Types

- Differentiated mainly by:
  - Request Format
  - Request Contents
  - Response Format
  - Response Contents

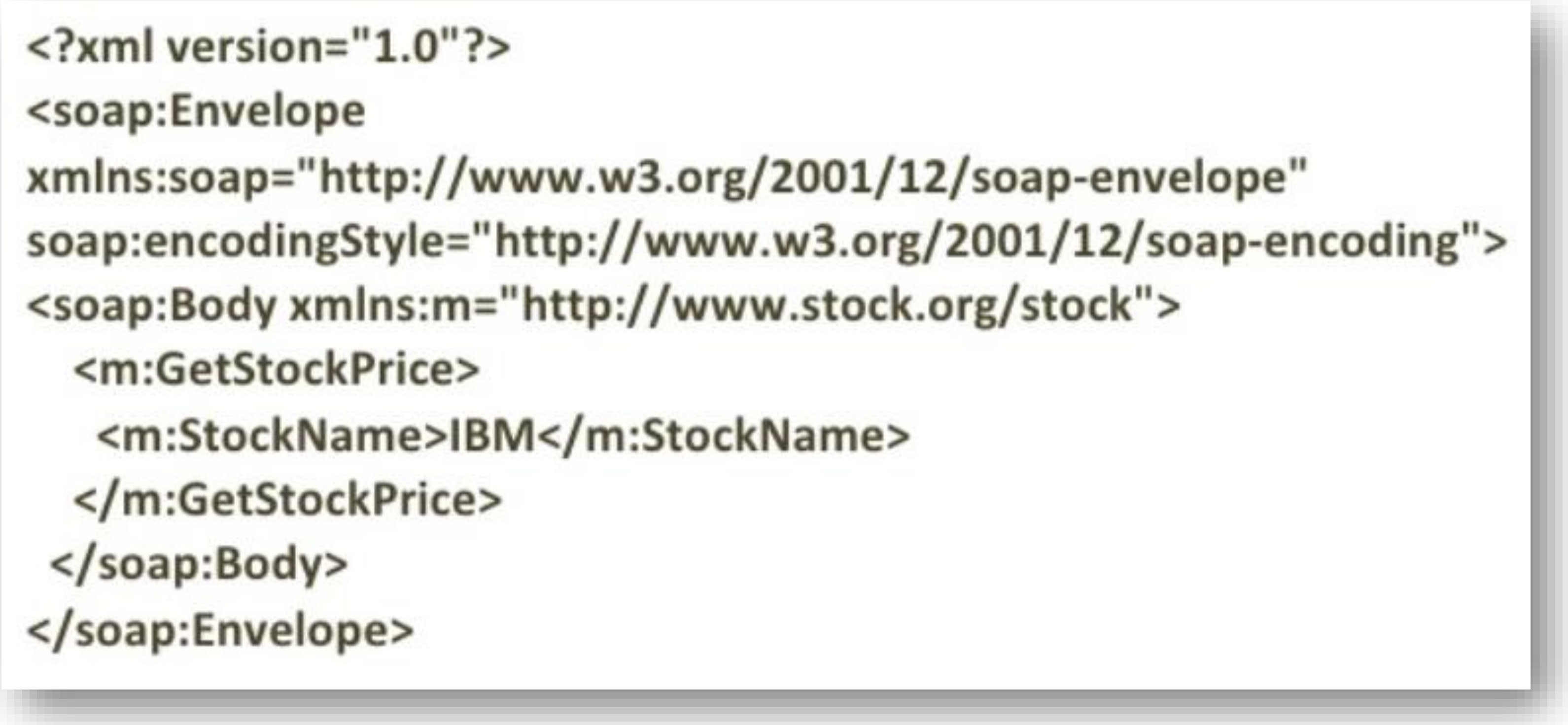
Web API Types

- SOAP
- REST
- GraphQL
- gRPC

SOAP

- Simple Object Access Protocol
- Designed in 1998 for Microsoft
- XML-Based
- RPC Style
- Extensible

SOAP



```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body xmlns:m="http://www.stock.org/stock">
  <m:GetStockPrice>
   <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
 </soap:Body>
</soap:Envelope>
```

SOAP

- Outdated
- Do not use, unless have to

REST

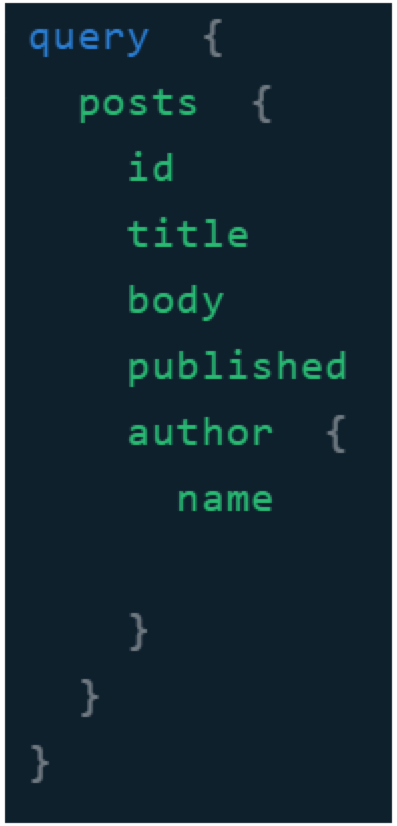
- Representational State Transfer
- Designed in 2000 by Roy Fielding
- URL + JSON Based
- Message Style
- De Facto Standard for APIs
- We’ll talk about it a lot later ☺

GraphQL

- Developed internally in 2012 by Facebook
- Released publicly in 2015
- Enables very flexible querying, updating and subscribing to data changes
- JSON in & out

GraphQL

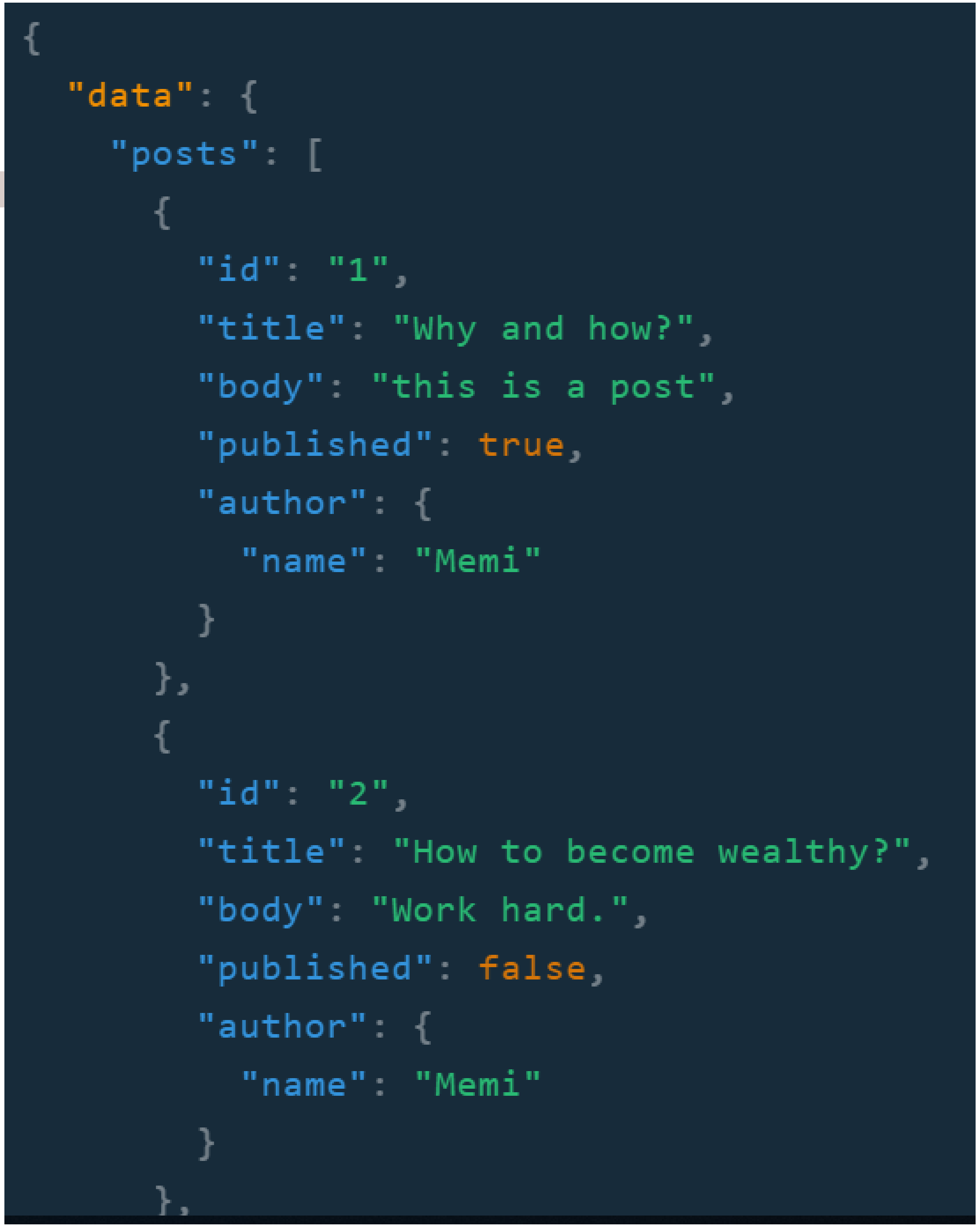
Request



```
query {
  posts {
    id
    title
    body
    published
    author {
      name

    }
  }
}
```

Response



```
{
  "data": {
    "posts": [
      {
        "id": "1",
        "title": "Why and how?",
        "body": "this is a post",
        "published": true,
        "author": {
          "name": "Memi"
        }
      },
      {
        "id": "2",
        "title": "How to become wealthy?",
        "body": "Work hard.",
        "published": false,
        "author": {
          "name": "Memi"
        }
      },
```

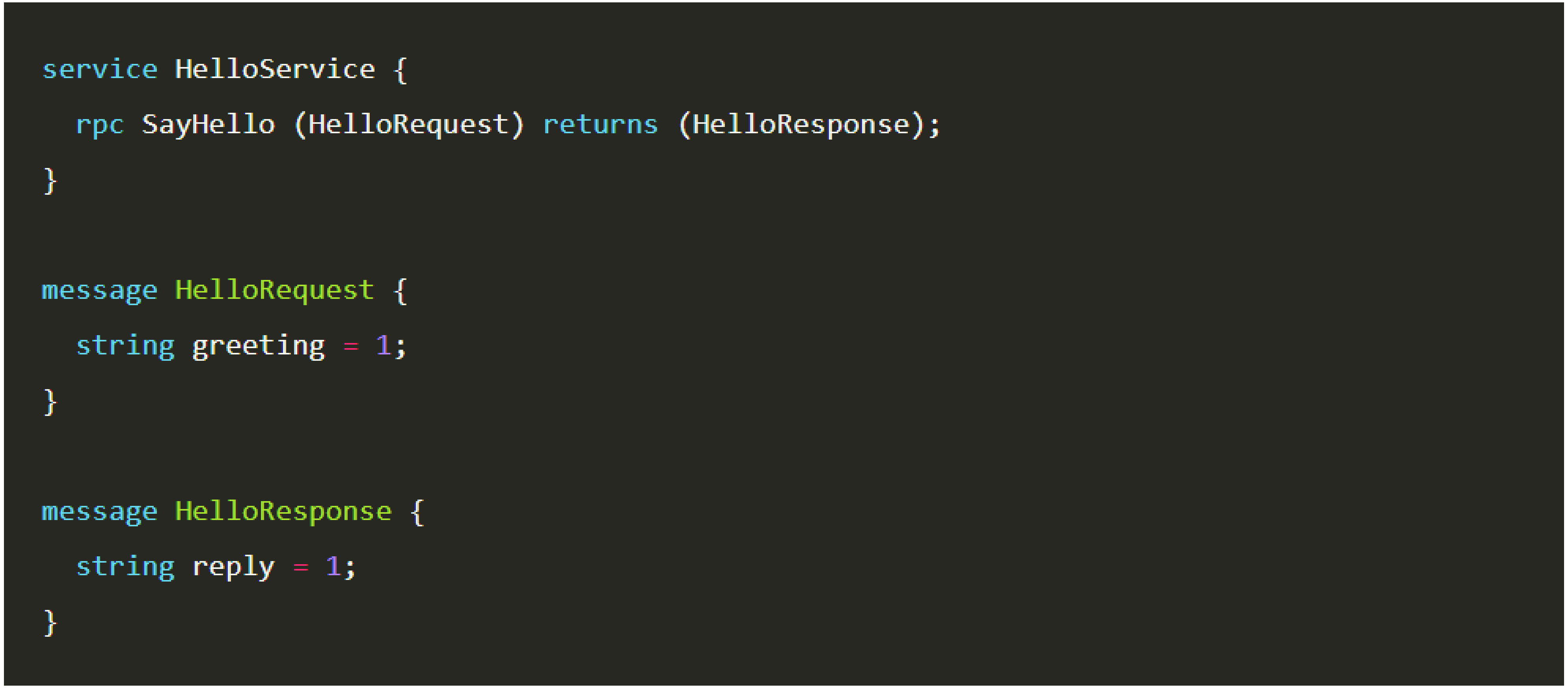
GraphQL

- VERY flexible
- Requires some upfront dev effort
- Requires performance optimization
- Gains ground (ie. Github)

gRPC

- Developed by Google in 2015
- Using HTTP/2
- Uses Protobuf
- Supports bi-directional & streaming messaging

gRPC



```
service HelloService {
  rpc SayHello (HelloRequest) returns (HelloResponse);
}

message HelloRequest {
  string greeting = 1;
}

message HelloResponse {
  string reply = 1;
}
```

gRPC

- Quite performant
- Not widely used yet
- Requires specialized libraries at both ends