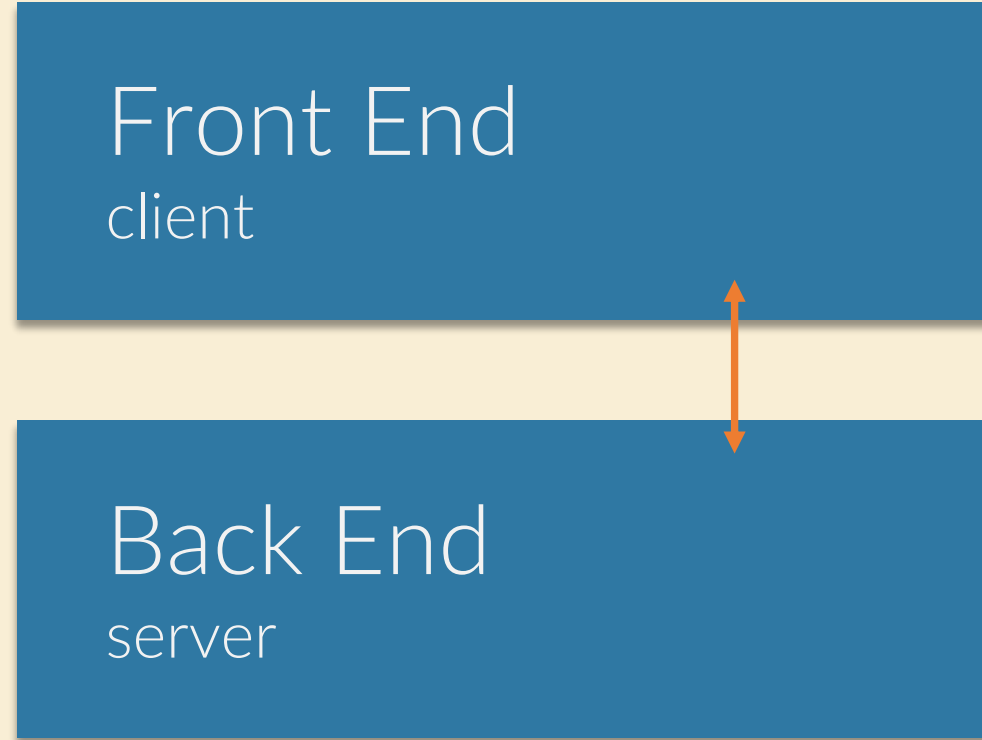


SOFTWARE ENGINEERING

*Chapter 3.5: System Design
Architectures Part II*

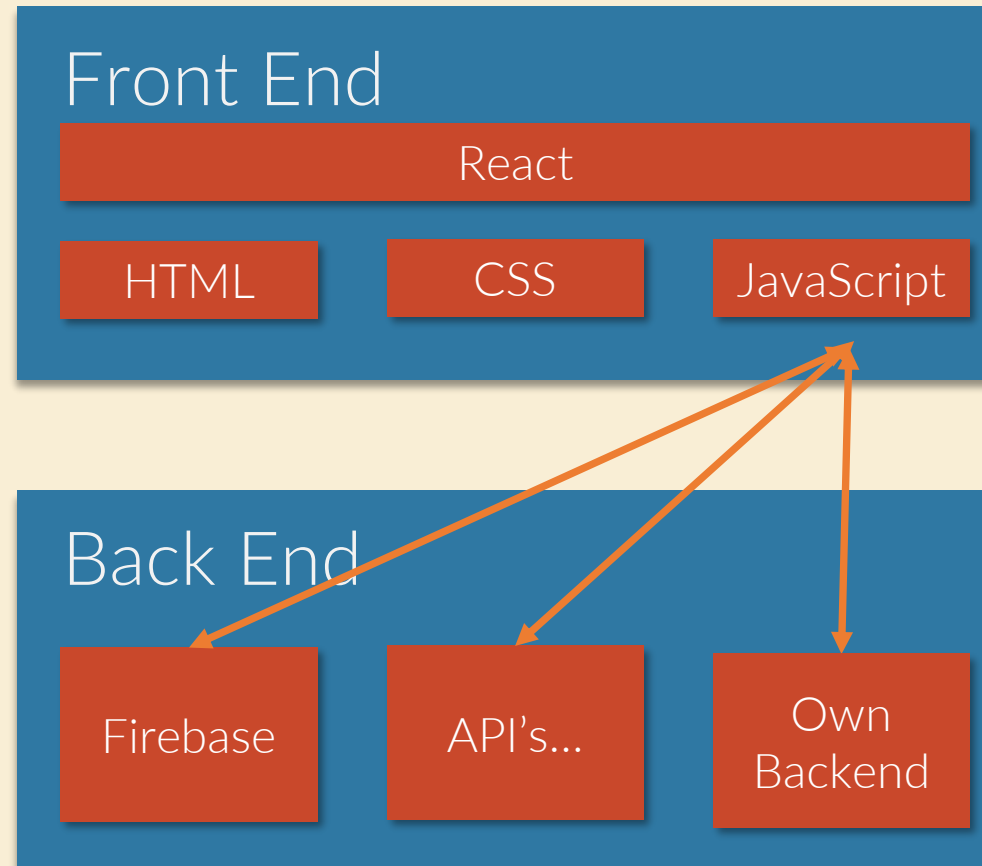
Simple Architecture pt I



Which kind of pattern is this one?, can you guess?



Simple Architecture

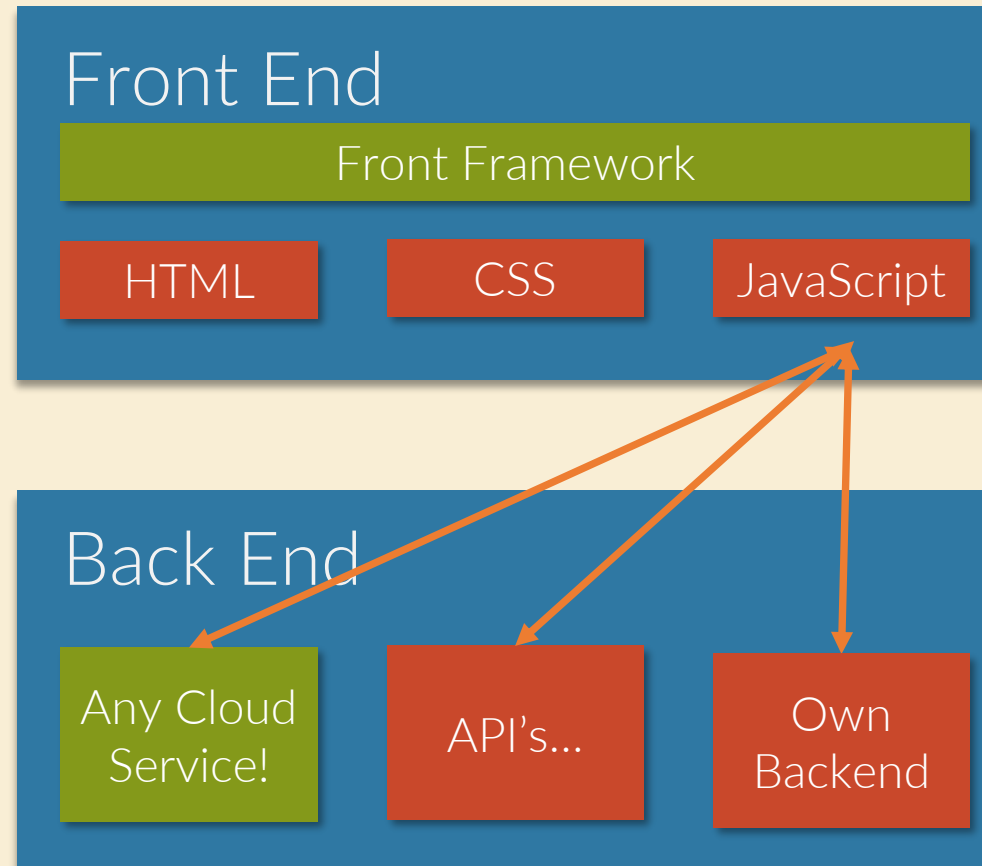


WAIT!

There are some technological insights I think that can be useful to understand better this architecture!

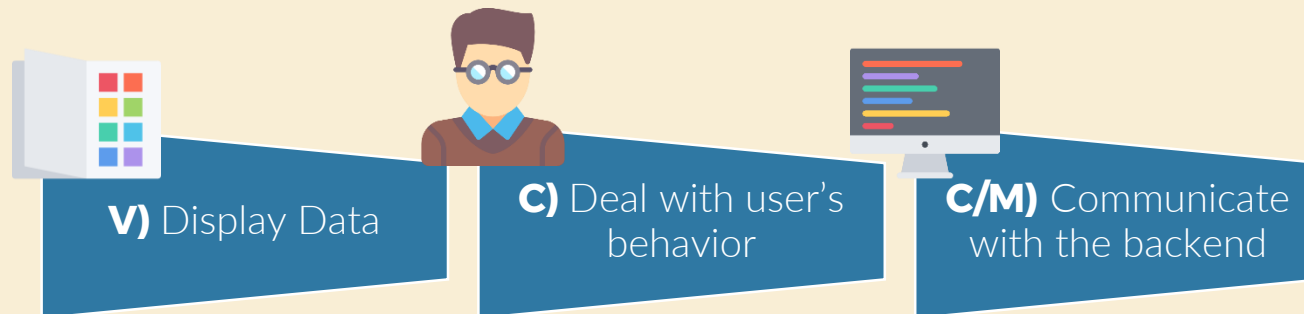


Simple Architecture



Front Framework

- Is used to build client-side applications!
- Normally this Front end frameworks rely on MVC architectures in order to:



Front Framework Comparison

Angular

- Full fledged MVC Javascript components
- Brings JS to HTML, Real DOM and Client Side rendering
- 2 Way Data Binding
- Javascript + HTML

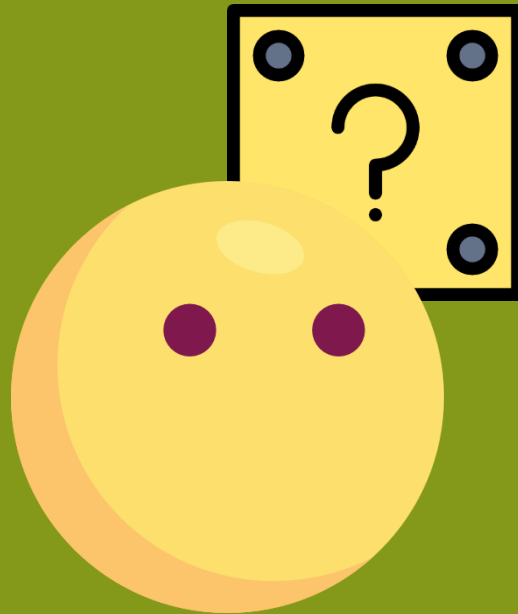
Does better if the SPA has a single view

React + Flux

- Javascript library (React), view in MVC and flux gets the rest of the architecture
- Virtual rendering, Virtual DOM and brings HTML to JS
- 1 Way Data Binding
- Javascript + JSX

Does better if the SPA has multiple views

What's that Data-Binding Thing?



Data Binding Example

```
public class User {  
    public String firstname;  
    public String lastname;  
    public int age;  
    public String gender;  
  
    public User(String firstname, String lastname, int age, String gender){  
        this.firstname = firstname;  
        this.lastname = lastname;  
        this.age = age;  
        this.gender = gender;  
    }  
}
```

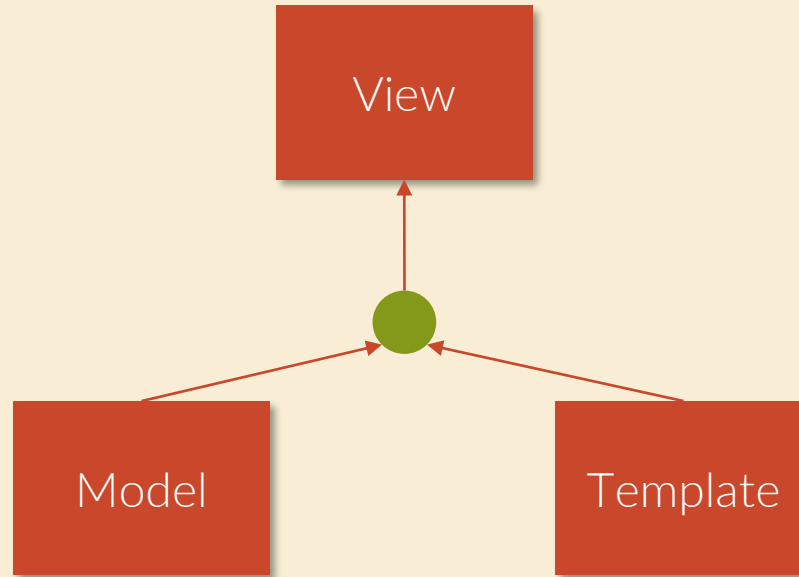
```
<RelativeLayout >  
    <TextView  
        android:id="@+id/firstnameLabel"  
        android:text="@string/firstname" />  
  
    <TextView  
        android:id="@+id/firstnameTextView"  
        android:text="@{user.firstname}" />  
  
    <TextView  
        android:id="@+id/lastnameLabel"  
        android:text="@string/lastname" />  
  
    <TextView  
        android:id="@+id/lastnameTextView"  
        android:text="@{user.lastname}" />  
</RelativeLayout>  
</layout>
```

Data Binding

- See how from the model (Business logic) you can update the view!
- This is so explicit for example in android!

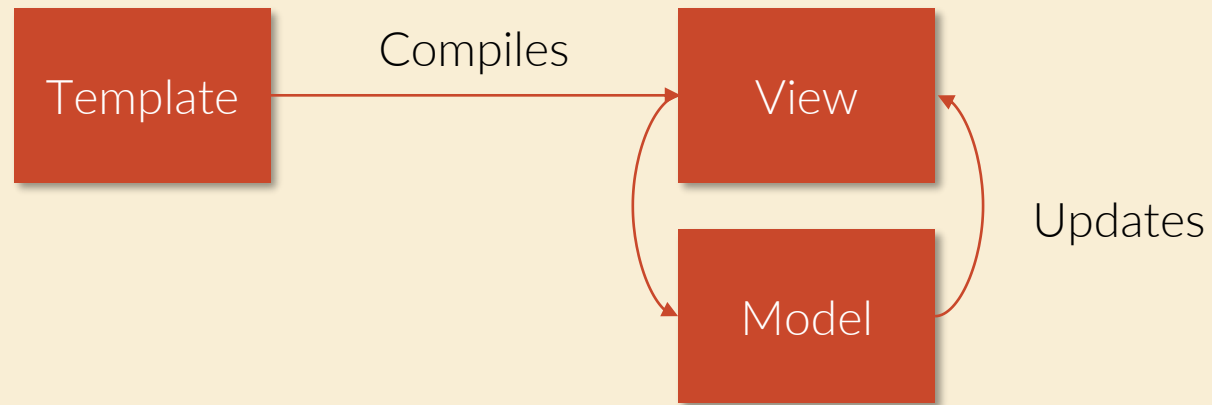


One way data binding



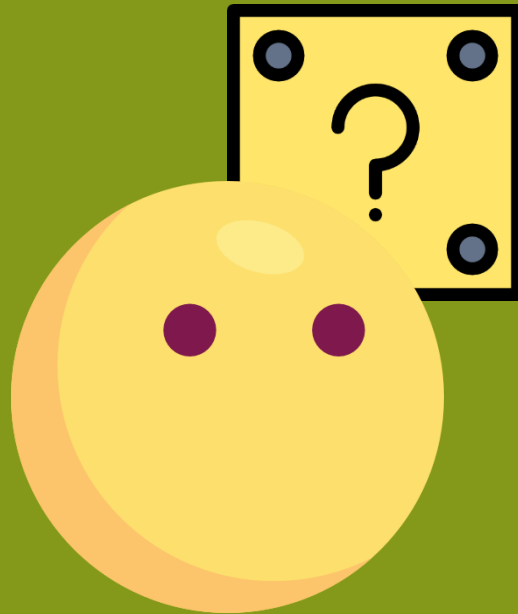
- The red dot is the only one merge time

Two ways data binding



- The view is constantly updated by the model, that increases performance!

What about those low level things?





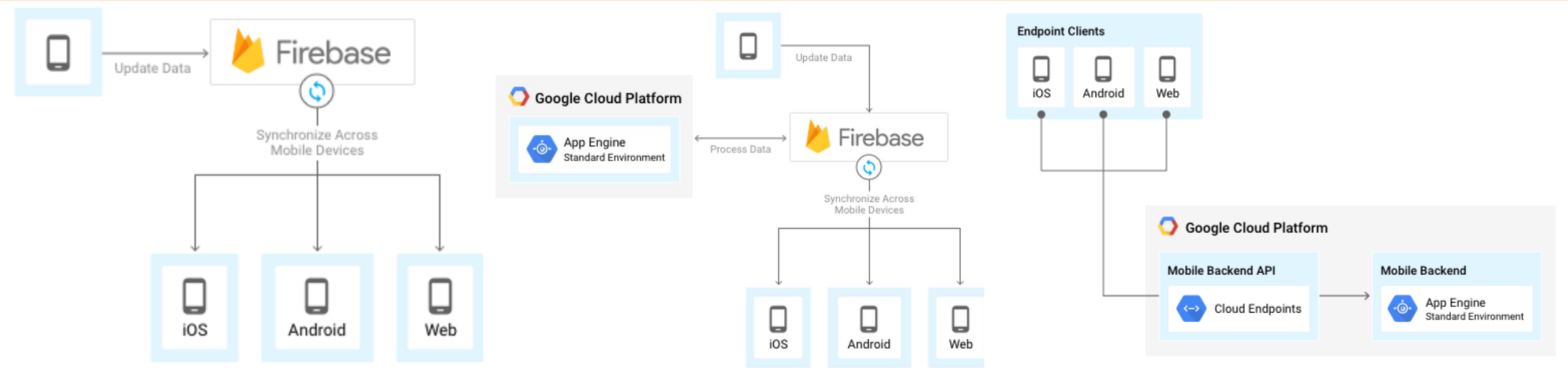
Cloud services

- See for example different patterns with Google App Engine and Firebase!

<https://cloud.google.com/solutions/mobile/mobile-app-backend-services#design-pattern>

Feature	Firebase	Firebase & App Engine standard environment	Firebase & App Engine flexible environment	App Engine standard environment & Endpoints	Compute Engine & REST/gRPC
Automatic capacity scaling	✓	✓	✓	✓	If you configure an autoscaler.
Automatic real-time data synchronization	✓	✓	✓		
Automatic server maintenance	✓	✓	✓	✓	
Backend logic		✓	✓	✓	✓
Call native binaries, write to the file system, or make other system calls.			✓		✓
Data storage	✓	✓	✓	If you add other Cloud Platform services	If you add other Cloud Platform services
File storage	✓	✓	✓	✓ with Cloud Storage	✓ with Cloud Storage
Easy user authentication	✓	✓	✓	OAuth 2.0	
Language support for backend service logic	N/A	Java, Python, Go, PHP	Any	Java, Python, Go (Cloud Endpoints for Go.)	Any
Messages and notifications, such as push notifications	✓	✓	✓	✓ with Cloud Messaging	✓ with Cloud Messaging
Platform support	iOS, Android, Web	iOS, Android, Web	iOS, Android, Web	iOS, Android, Web	iOS, Android, Web
Requires code to run within a sandbox .	N/A	✓		✓	
Requires SSL		✓		✓	

Pattern View



Cloud patterns using Firebase and / or GAE (Google App Engine)



HTTP:// REQUESTS

Are the way information is sent from the client to the server and is answered with files or data...

Client

HttpRequest

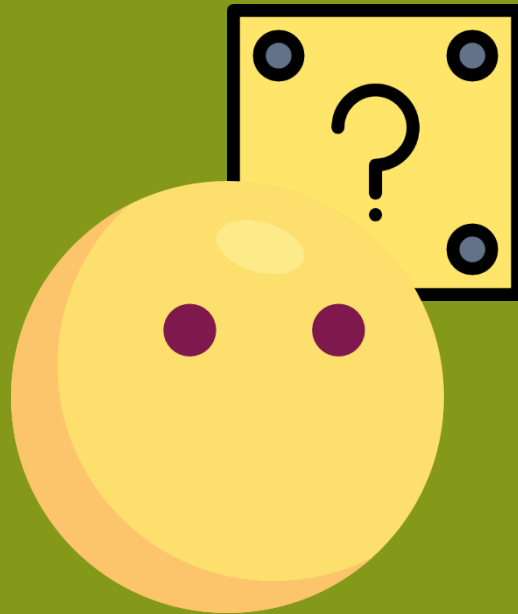
HTTP GET http://api.wunderground.com/api/
3bee87321900cf14/conditions/q/VA/Fairfax.json

HttpResponse

HTTP/1.1 200 OK Server: Apache/2.2.15
(CentOS) Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true X-
CreationTime: 0.134 Last-Modified: Mon, 19
Sep 2016 17:37:52 GMT Content-Type:
application/json; charset=UTF-8 Expires:
Mon, 19 Sep 2016 17:38:42 GMT Cache-
Control: max-age=0, no-cache Pragma: no-
cache Date: Mon, 19 Sep 2016 17:38:42 GMT
Content-Length: 2589 Connection: keep-
alive { "response": { "version":"0.1",
"termsOfService":"http://www.wunderground.
com/weather/api/d/terms.html",

Server

What about REST and SOAP?



REST

- Means representational state transfer
- It's an style of architecture describing the way networked resources should be adressed and accessed.

REST

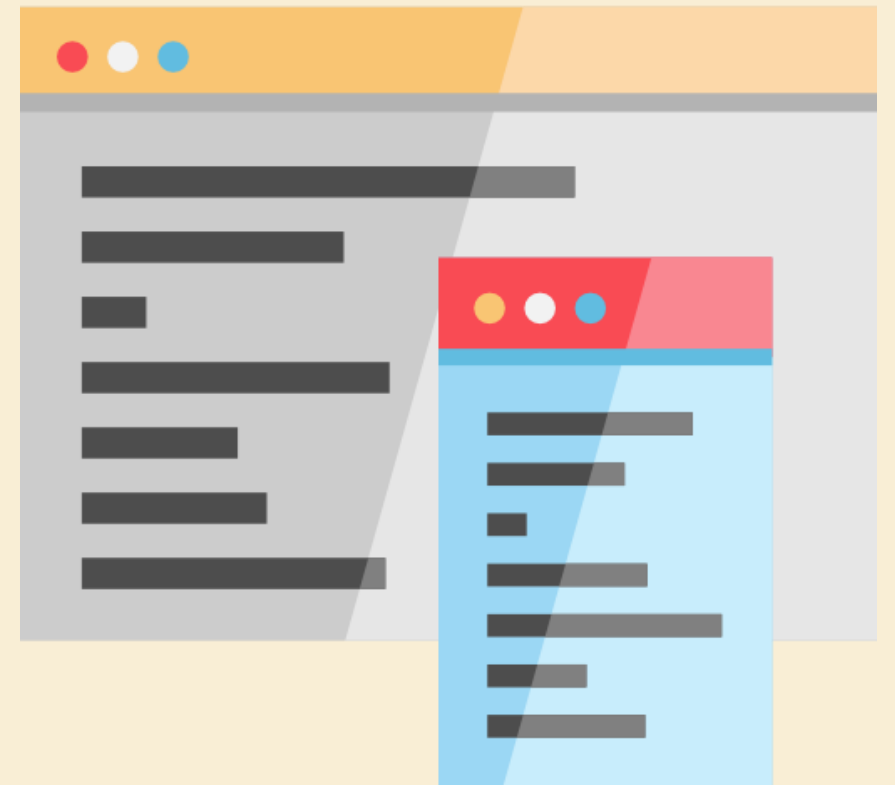
“Throughout the HTTP standardization process, I was called on to defend the design choices of the Web. That is an extremely difficult thing to do... I had comments from well over 500 developers, many of whom were distinguished engineers with decades of experience.

That process honed my model down to a core set of principles, properties, and constraints that are now called REST.”

REST

Maybe with all of this it can be familiar:

- URL/URI
- Hypertext
- Accept: text/html
- 400, 404, Bad Request
- GET, PUT, POST, DELETE

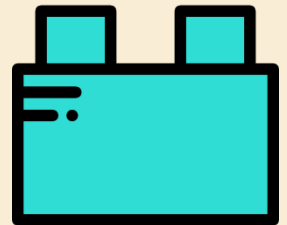


REST FEATURES



REST RESOURCE

- Is everything that can be accessed by an URI
- URI means universal resource identifier, they can be used to describe anything:
Persons, places, emails, tweets, posts...



REST RESOURCE

- An URL is a subset of an URI, it specifies where the resources live and how to retrieve it



REST REPRESENTATIONS

- RESTful systems ask for data in a way they can understand it

GET /pages/archive HTTP/1.1

Host: host.com

Accept: text/html

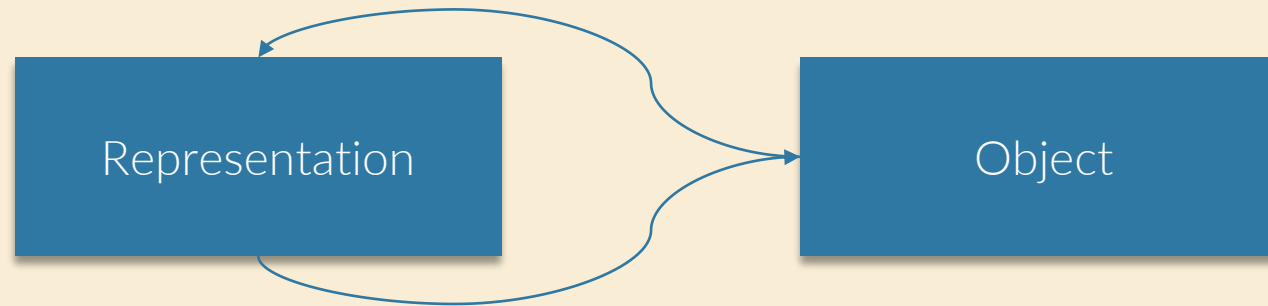
REPRESENTATIONS

- Each file can be represented by any of the 1000.... MIME types, not necessarily HTML.
- You can have different representations of the same resource (html, xls, JSON...)
- Things can be abstracted from the representation!

REPRESENTATIONS

Abstraction!

```
var yourJsonObject = JSON.parse(jsonString);
```



REST OPERATIONS

Operation	HTTP Request
Create	POST
Retrieve	GET
Update	PUT
Delete	DELETE

Operations Example

```
$.ajax({ url: "http://webservice.com/resource/940",  
        method: "PUT",  
        data: { "name": "My first resource!!" } });
```

REST EXAMPLE

The diagram illustrates an HTTP REST example, showing the request and response headers with callouts explaining their components.

Encabezados de la petición

Método: GET

URI de la solicitud: /sqlrest/INVOICE/9999/

Formatos aceptados para la representación: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Encabezados de la respuesta

Código de respuesta: HTTP/1.1 200 OK

Formato en que se envía la respuesta: Content-Type: application/xml

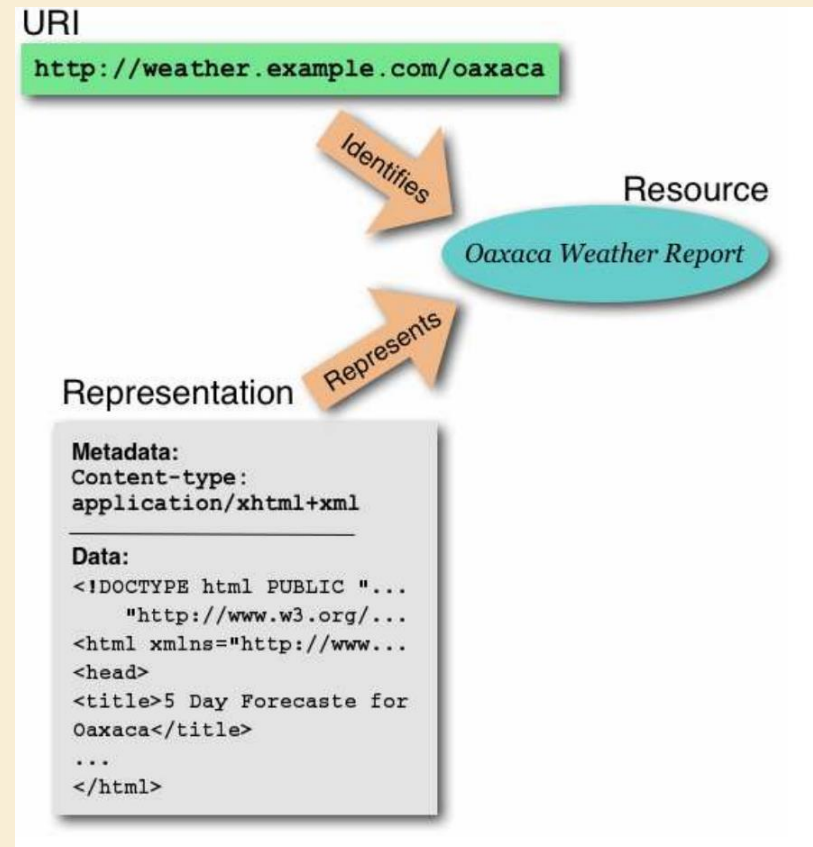
Other request headers: Host: www.thomas-bayer.com, User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; es-AR; rv:1.9.2.3), Accept-Language: es,en;q=0.8,es-ar;q=0.5,en-us;q=0.3, Accept-Encoding: gzip,deflate, Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7, Keep-Alive: 115, Connection: keep-alive

Other response headers: Server: Apache-Coyote/1.1, Transfer-Encoding: chunked, Date: Wed, 07 Apr 2010 01:30:00 GMT

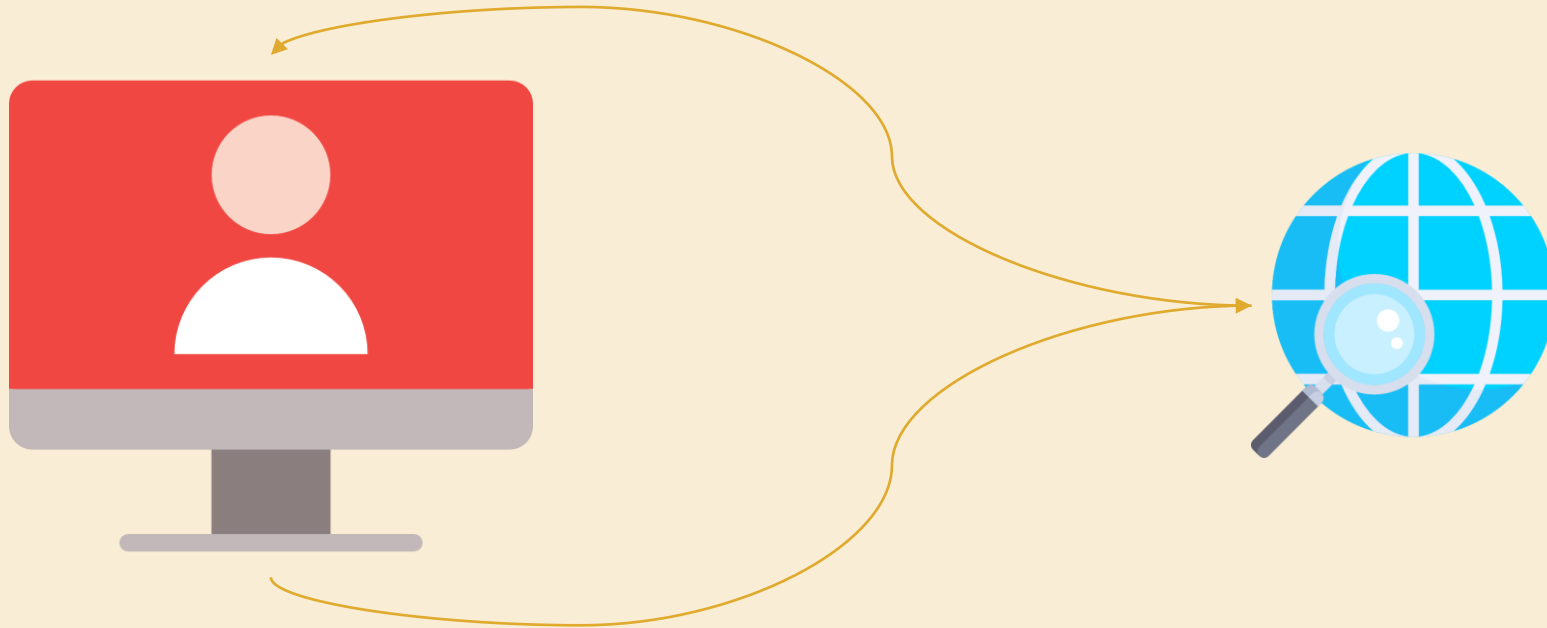
REST EXAMPLE II

Take into account those properties we've seen so far.

HTTP requests can be understood as RESTful processes



HYPertext!



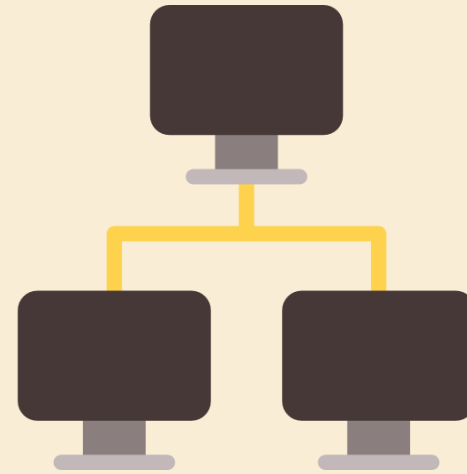
In REST, application state is transferred and discovered within hypertext responses.

Web servers publish hypertext, clients always know how to deal with it (change page, update...)

STATELESS



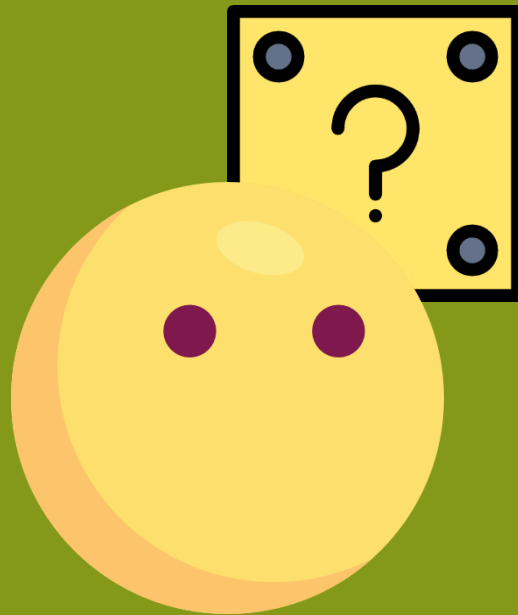
The client only deals with its states!



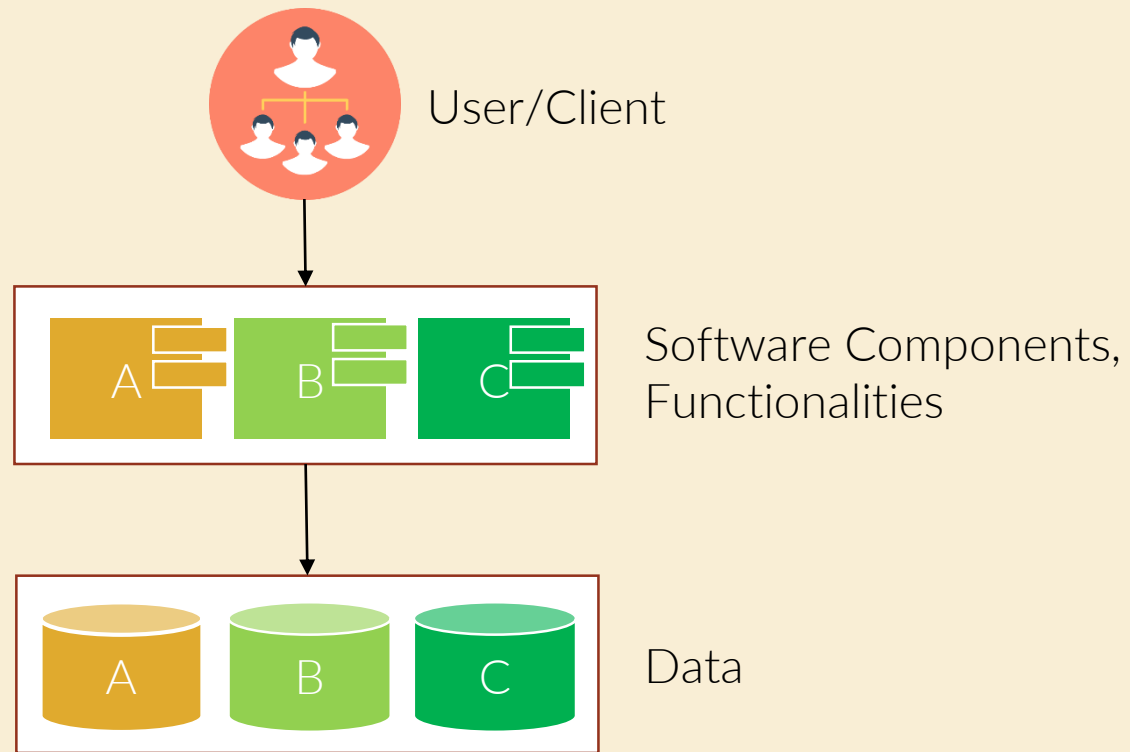
The server only deals with its states!

Decoupling Client / Server...!

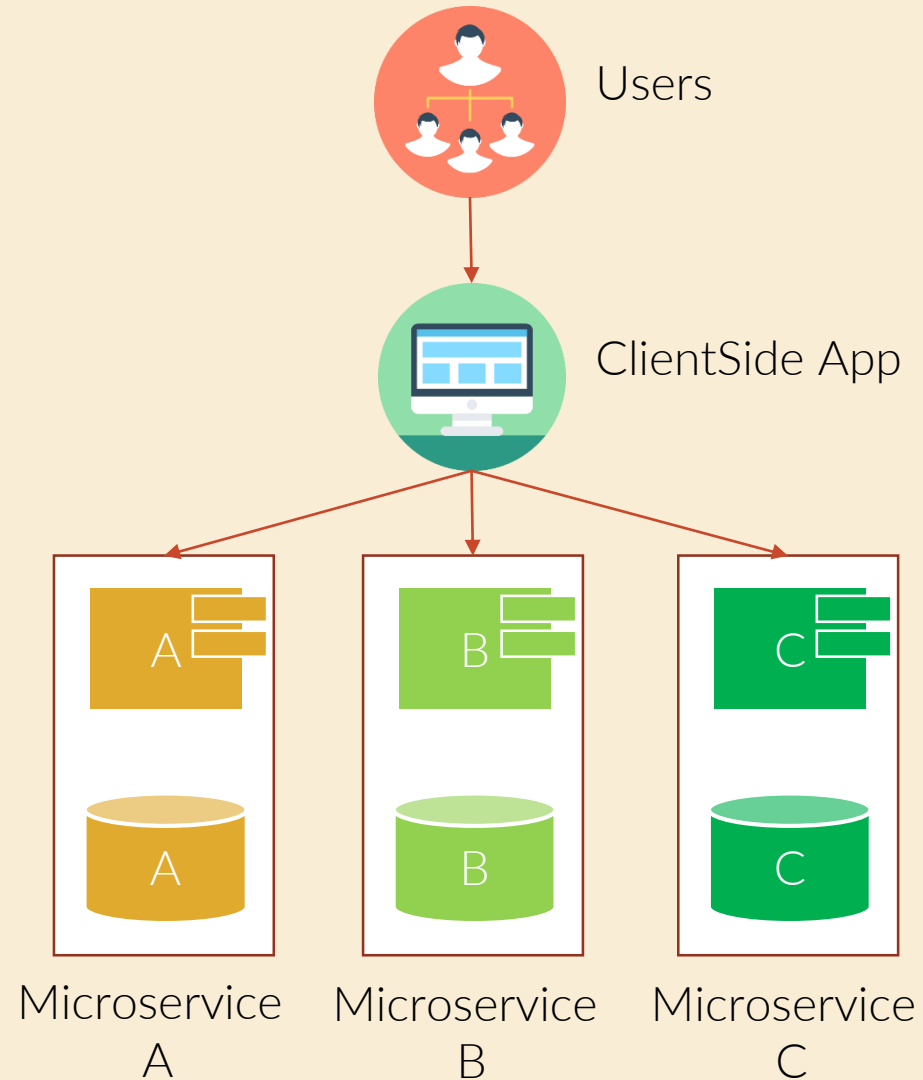
Why all of this?



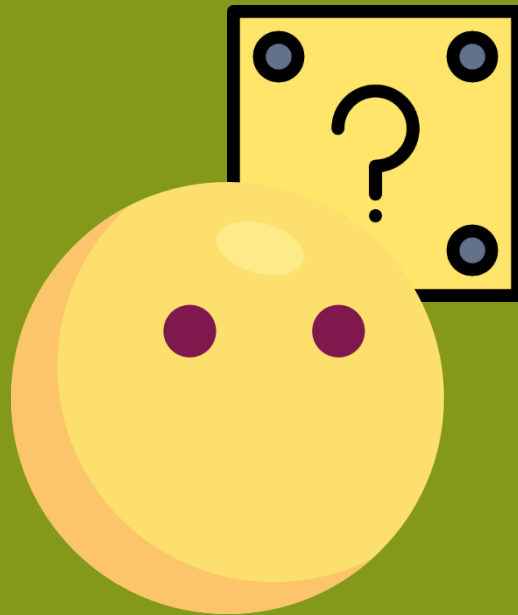
Monolithic Architectures



Microservices Architecture



What about SOAP?

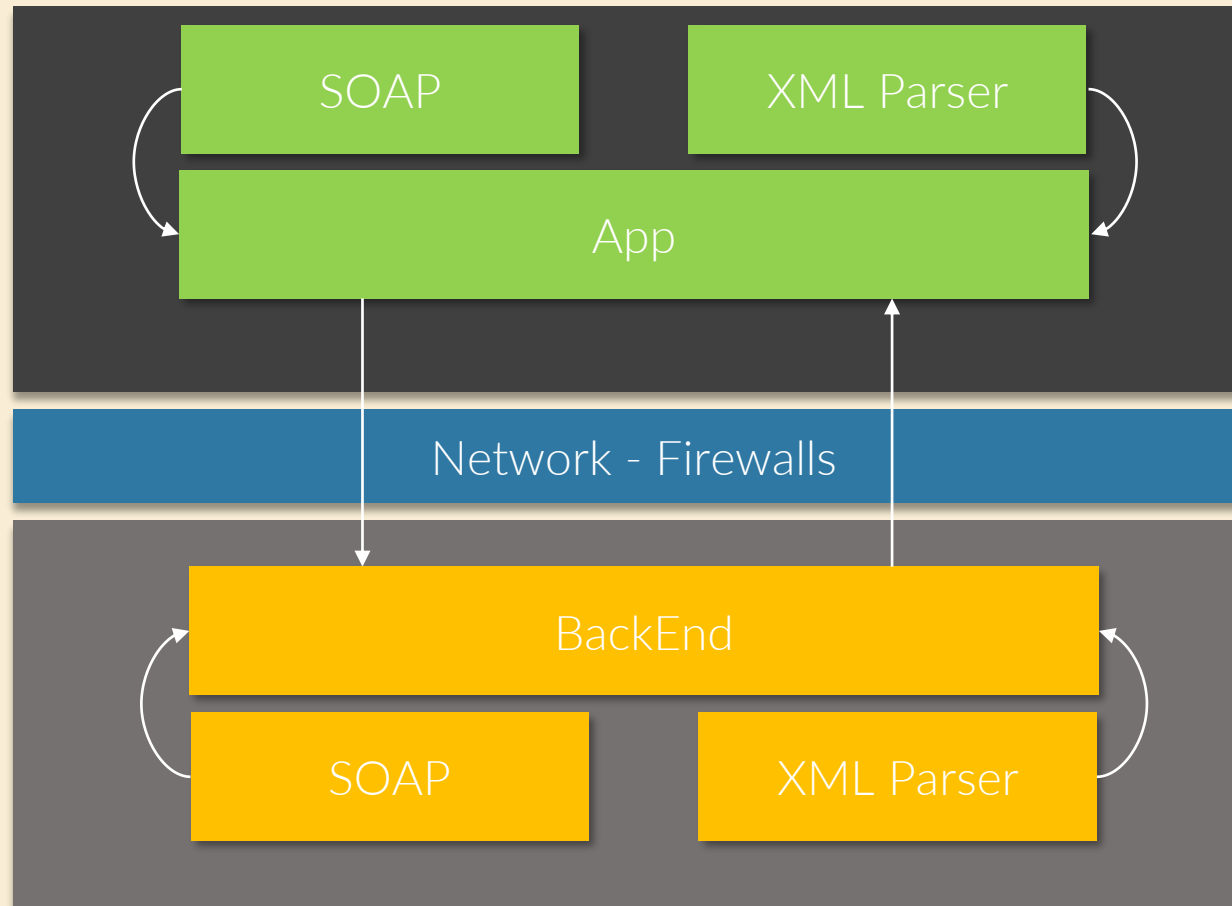


SOAP

Simple Object Access Protocol

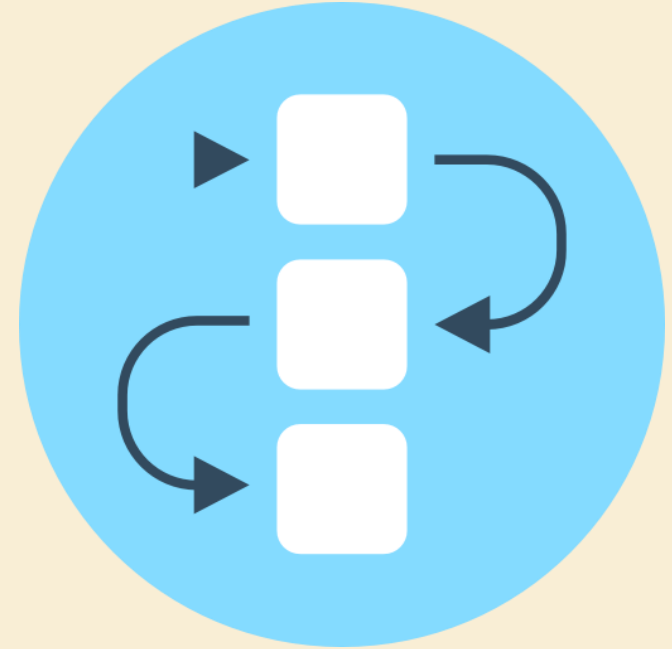
- Standard protocol used by web services
- It's main use is inter-application connectivity
- Encodes messages in xml and uses HTTP for transportation

SOAP Simple Use Case



SOAP Message Data

1. Envelope: Has the message and encoding format
2. Header: Additional parameters to add to the message
3. Body: Information that is going to be stored in the communication back and forth



SOAP Message

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Header>
    <t:Transaction xmlns:t="some-URI" SOAP-ENV:mustUnderstand="1">
      5
    </t:Transaction>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <Quote xmlns="http://namespaces.cafeconleche.org/xmljava/ch2/">
      <Price>4.12</Price>
    </Quote>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```


One Shot Review



Review!

- What is REST?
- Which patterns do you see in the web architecture?
- What are the 5 important features of REST?
- What do you think about REST?
- What is SOAP?



Class has died... for today!