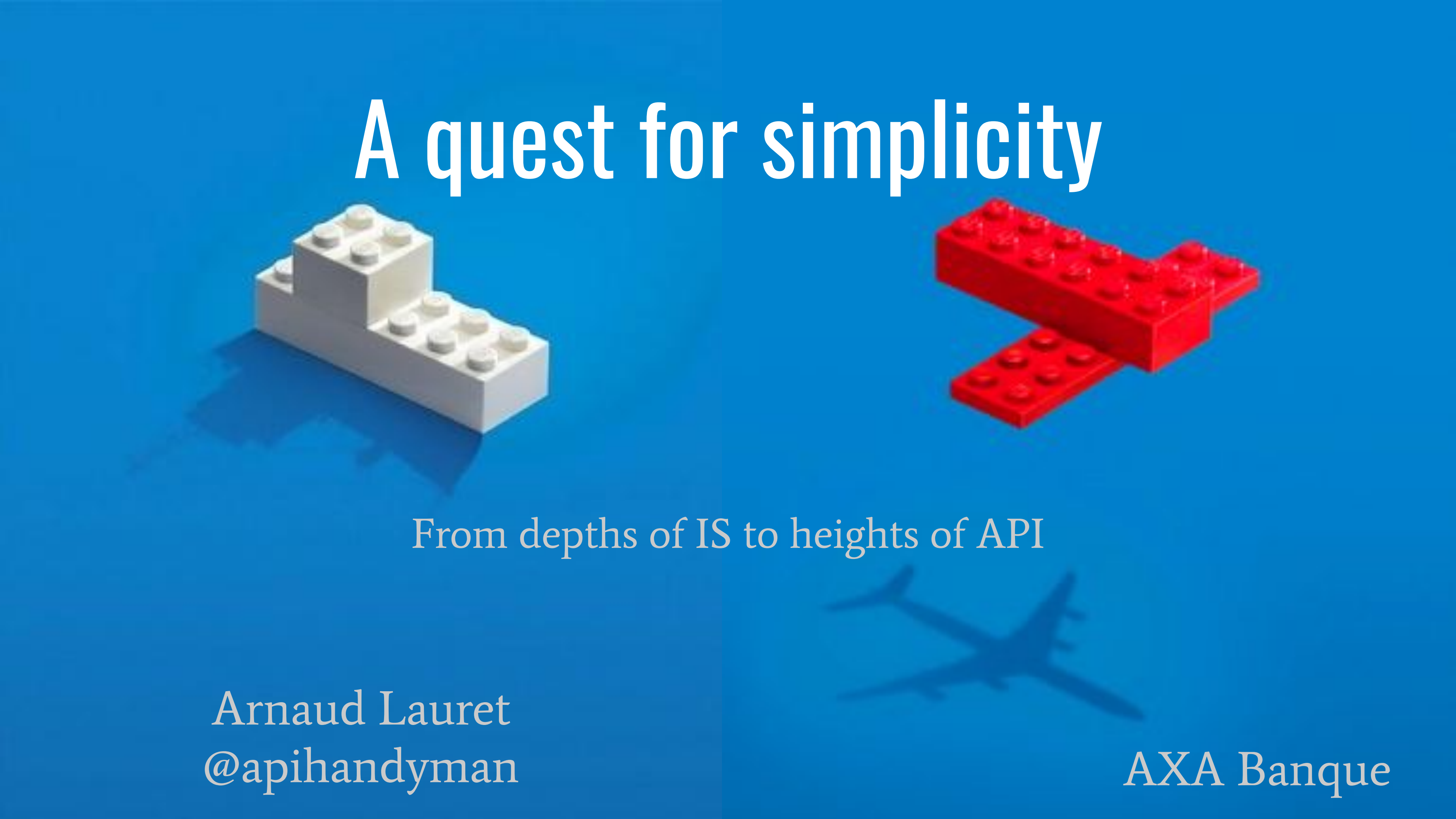
A quest for simplicity



From depths of IS to heights of API

Arnaud Lauret
@apihandyman

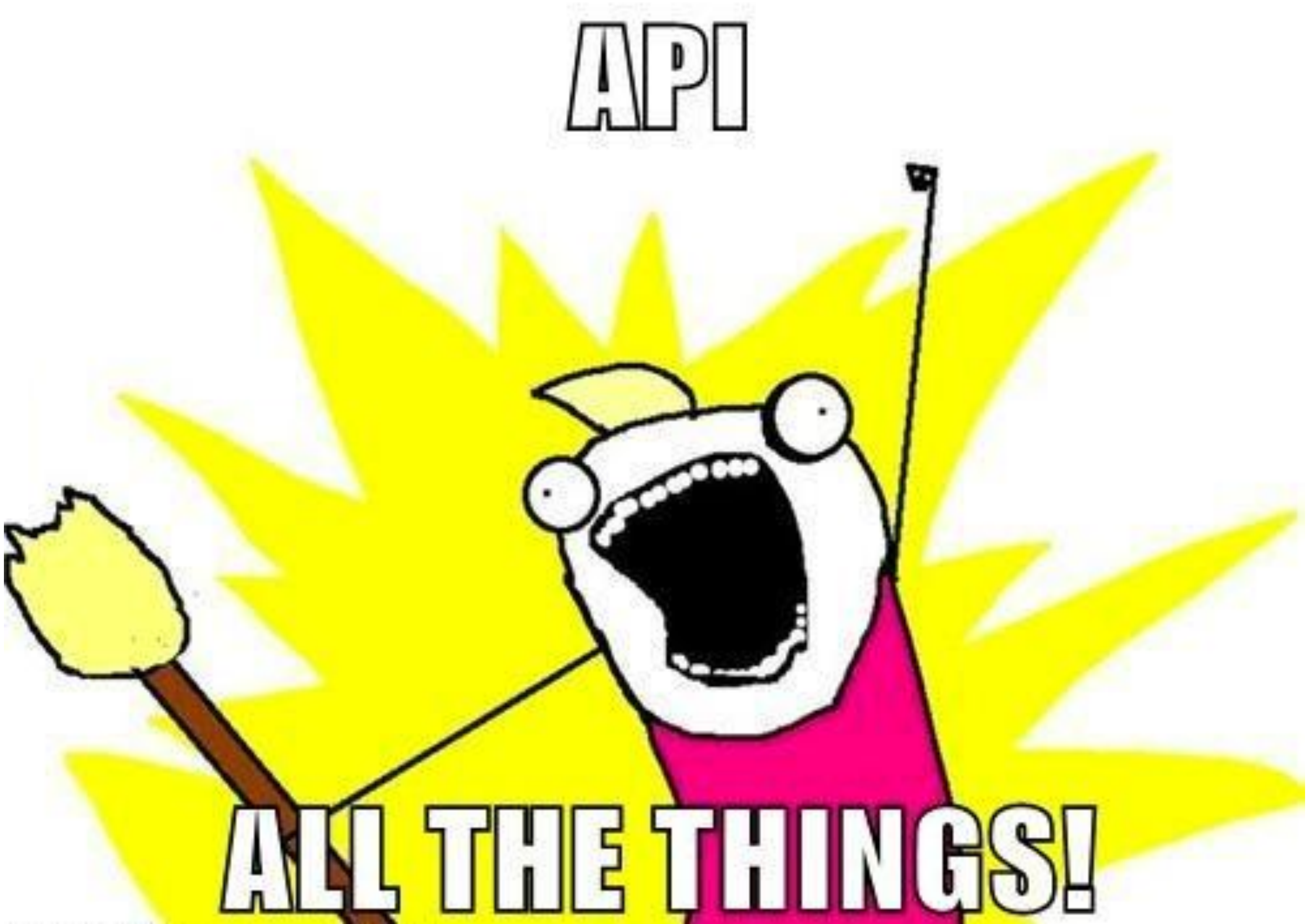
AXA Banque

One does not simply start a quest without a goal

•••

The events depicted in this talk are fictitious. Any similarity to any information system living or dead is merely coincidental.

What is your quest?


API
ALL THE THINGS!

What is your favorite word in API?

Application?

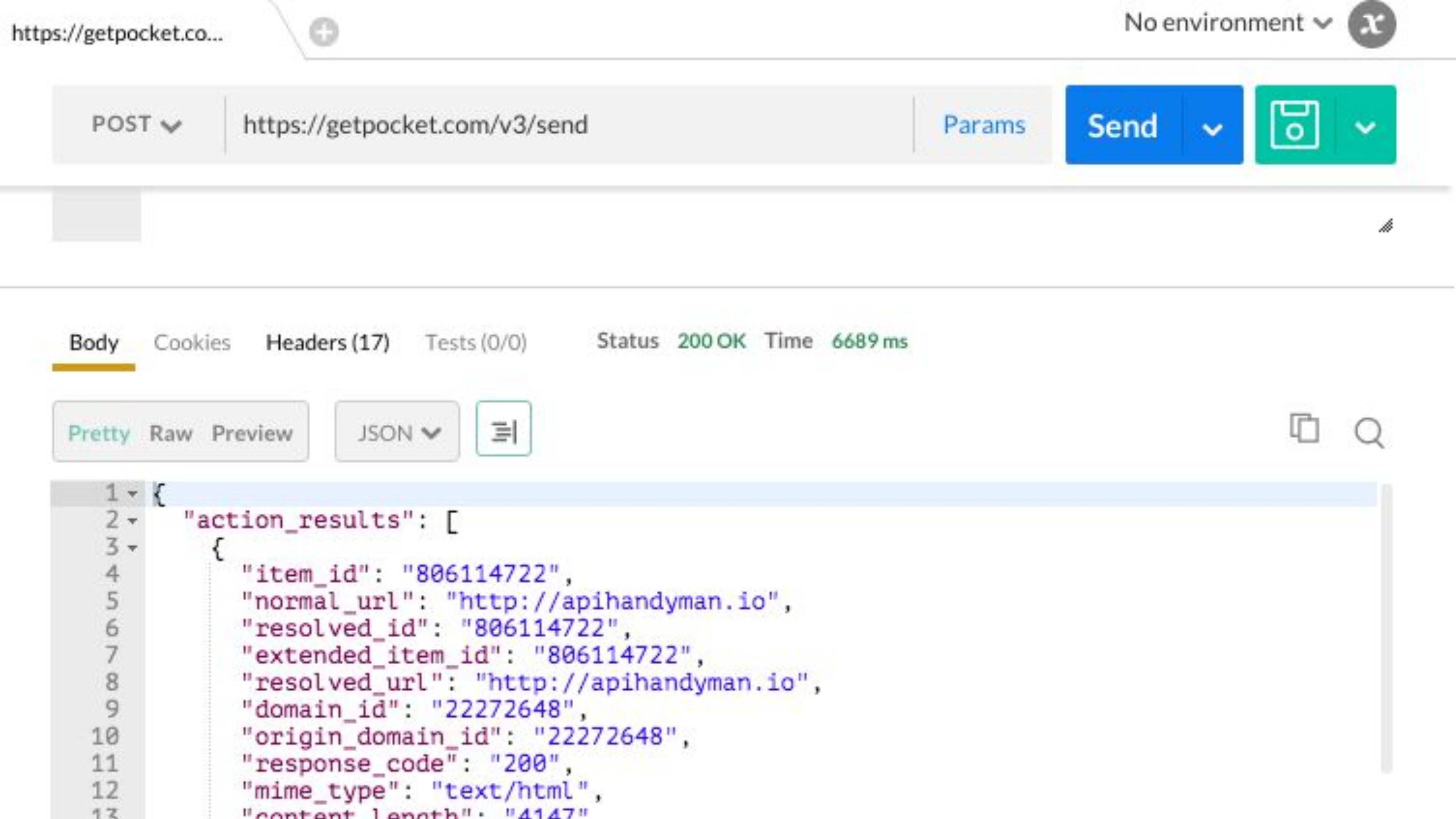
Programming?

Interface?

Interface!

The place at which **independent** and often **unrelated systems** meet and **interact** with each other

SOURCE
TTX/MIX
PRE-CH
MUTE
CH LIST
P
MENU
SMART HUB
GUIDE
TOOLS
INFO
RETURN
EXIT
A
B
C
D
E-MANUAL
AD/SUBT.
AA59-00786A
Nintendo
B
A
SELECT
START
17 16
300
450
600
700
800W
100
Pulse
KitchenAid
17:17
0
fn
ctrl
alt
cmd



https://getpocket.com/v3/send

Status 200 OK Time 6689 ms

```
{
  "action_results": [
    {
      "item_id": "806114722",
      "normal_url": "http://apihandyman.io",
      "resolved_id": "806114722",
      "extended_item_id": "806114722",
      "resolved_url": "http://apihandyman.io",
      "domain_id": "22272648",
      "origin_domain_id": "22272648",
      "response_code": "200",
      "mime_type": "text/html",
```

What have they in common?

xignite
Fueling Fintech Innovation
akana
Powering the API Economy
akana
Powering the API Economy
API Management
apigee
apigee

Us

People

(Even APIs)

API = UI

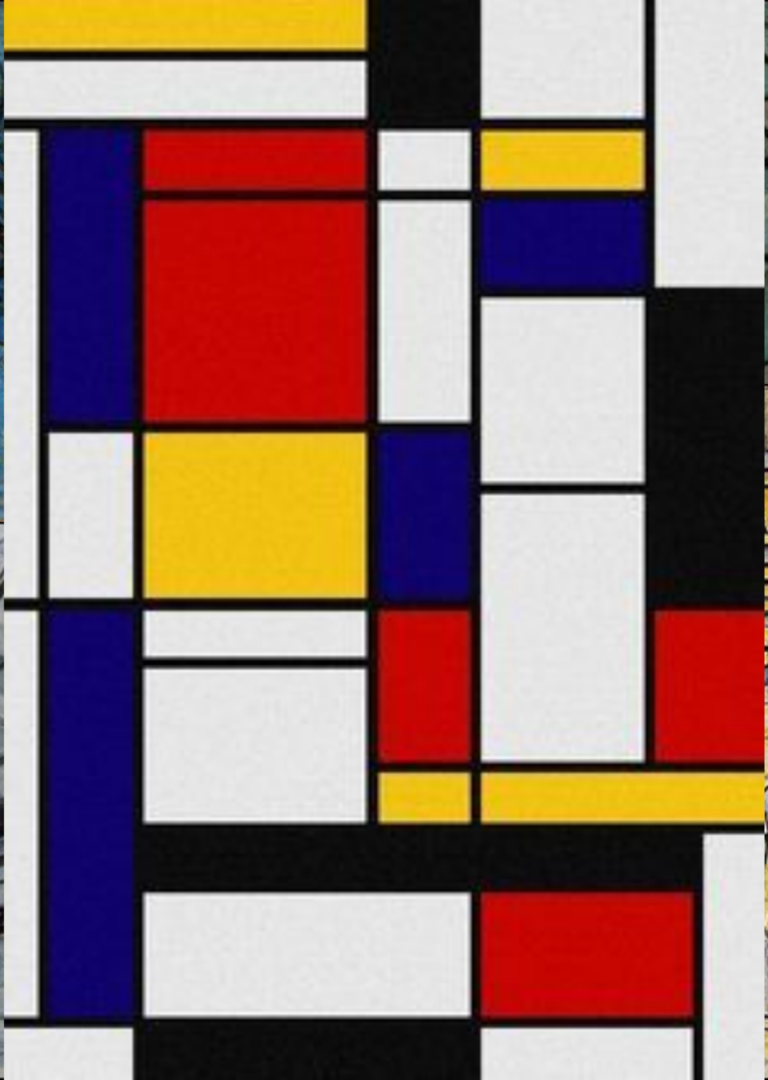
for people building programs

What's happening behind an interface?

The microwave oven parable

Abstraction

A picture is worth a thousand words



What do we seek when using an interface (especially an API) ?

Simplicity

Easy to use

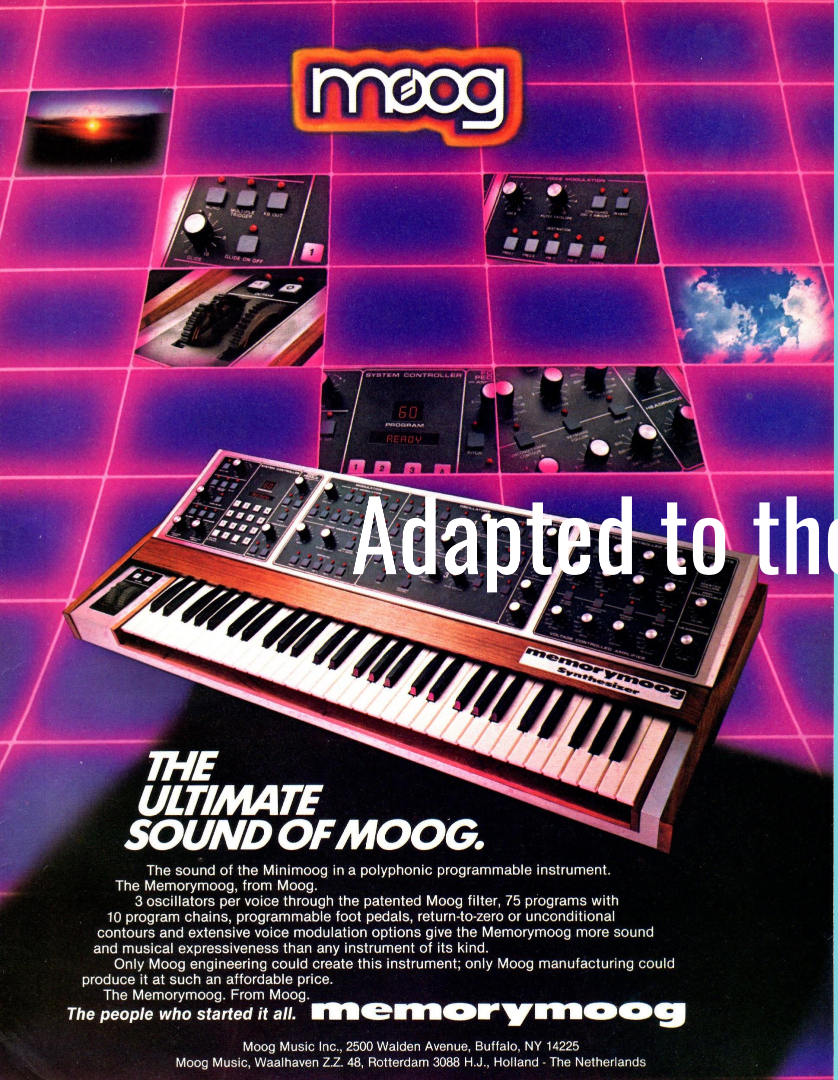
```
>AJDSH@TREGJHSBUUACLYHHAEXRSD
I DON'T KNOW THAT WORD
>MY NAMEIS COLPANTS
I DON'T KNOW THAT WORD.
>FGDHUDAURULHWFA
I DON'T KNOW THAT WORD.
>CHRISQ
I DON'T KNOW THAT WORD.
>
>
>
```

Easy to understand

Error 1543

Missing email

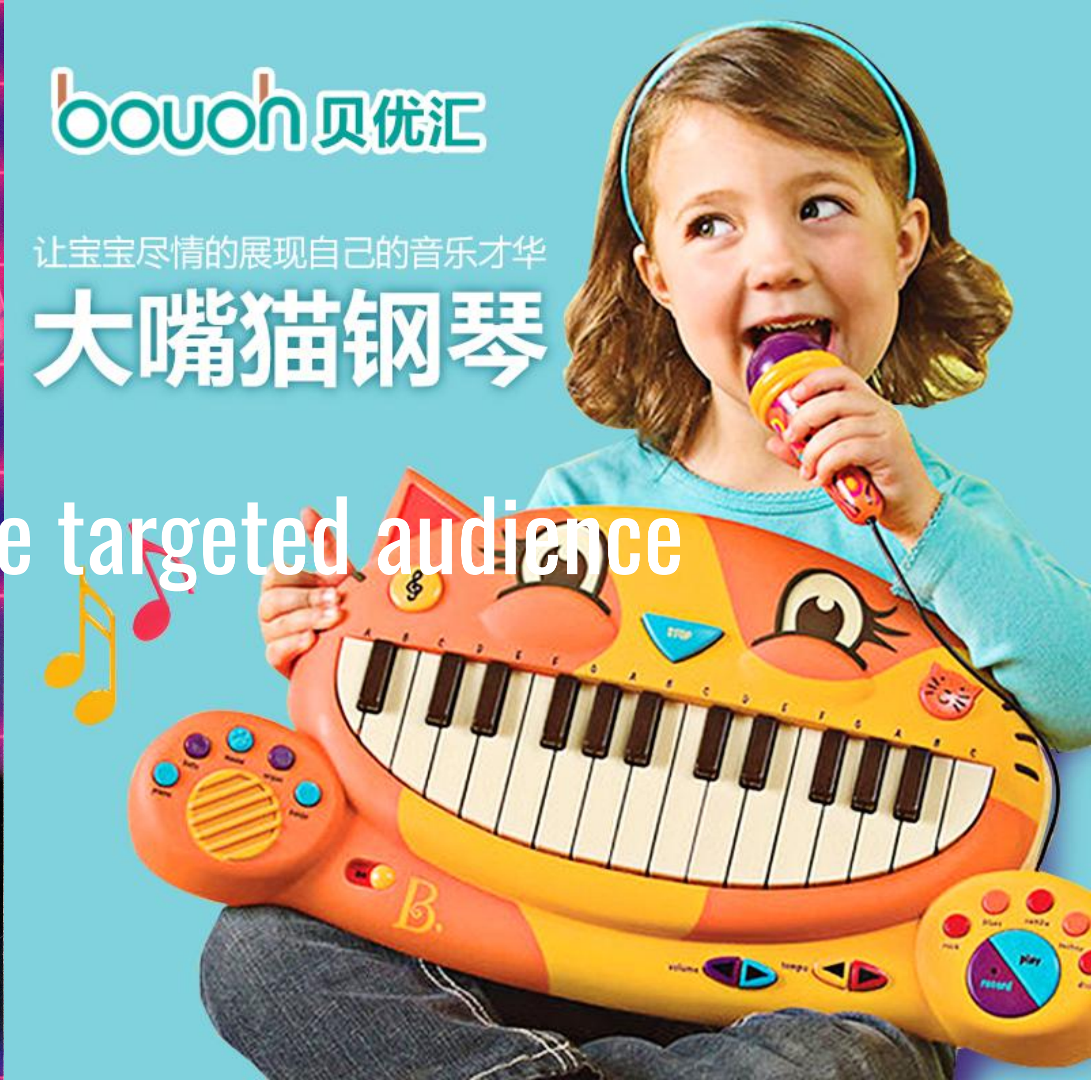
Adapted to the targeted audience



THE ULTIMATE SOUND OF MOOG.

The sound of the Minimoog in a polyphonic programmable instrument.
The Memorymoog, from Moog.
3 oscillators per voice through the patented Moog filter, 75 programs with 10 program chains, programmable foot pedals, return-to-zero or unconditional contours and extensive voice modulation options give the Memorymoog more sound and musical expressiveness than any instrument of its kind.
Only Moog engineering could create this instrument; only Moog manufacturing could produce it at such an affordable price.
The Memorymoog. From Moog.
The people who started it all. **memorymoog**

Moog Music Inc., 2500 Walden Avenue, Buffalo, NY 14225
Moog Music, Waalhaven Z.Z. 48, Rotterdam 3088 H.J., Holland - The Netherlands



So, what is our quest?

~~API~~ Simplify
ALL THE THINGS!

Core Banking System

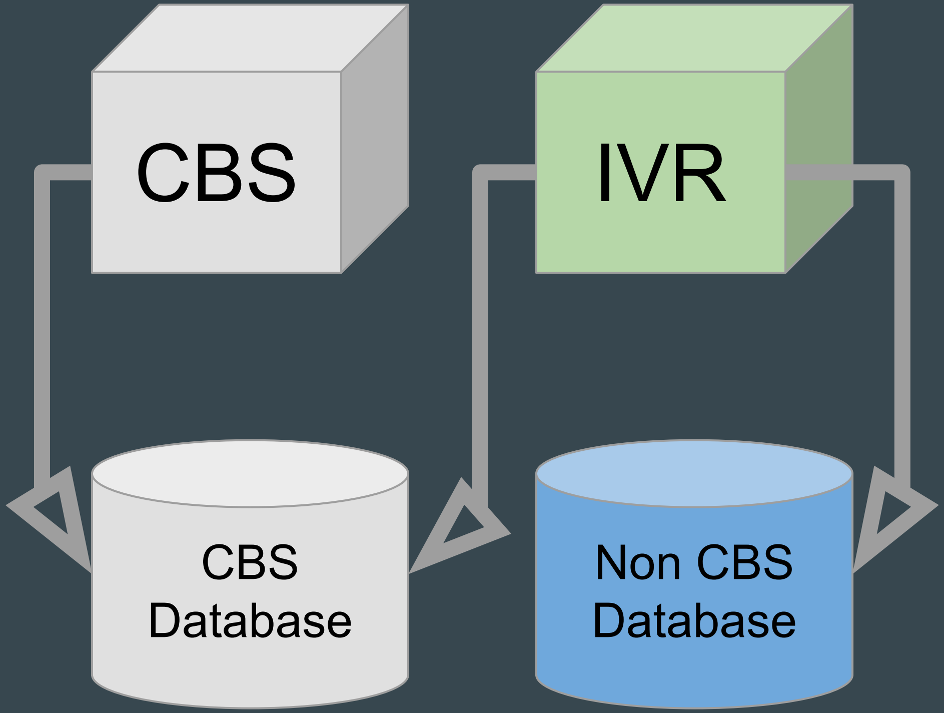
A long time ago…

This is an interface

~~API~~ Access
ALL THE THINGS!

IVR

Interactive Voice Response

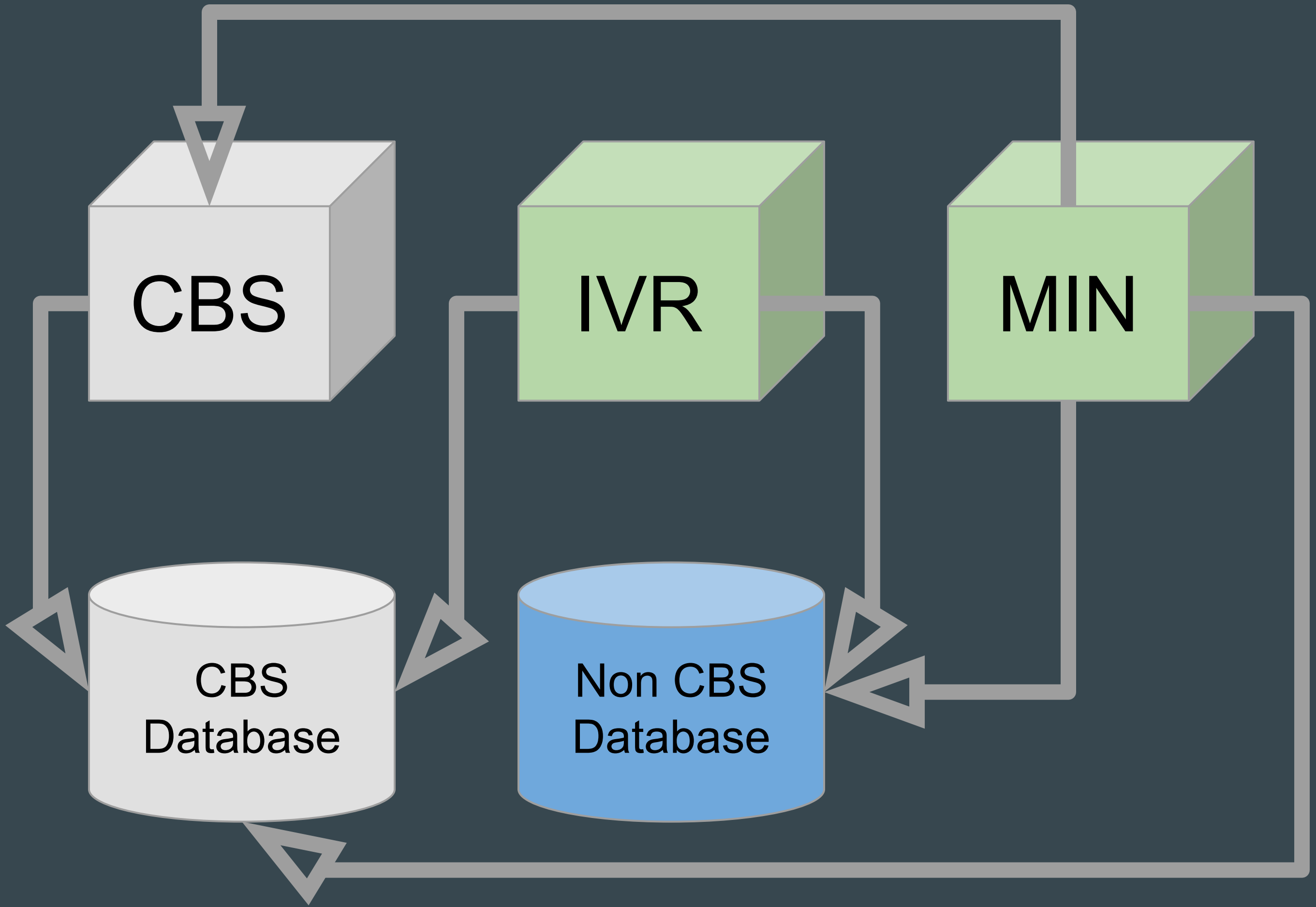


ZBAL0
ZBALV
ZBALA
ZBALY

FWW

Minitel

MQ Message:
F12021677541519125401237777463779FRF0300020160917



Is interfacing with CBS simple?

- Is it easy to use?
- It it easy to understand?
- Is the abstraction adapted to the audience?

So, it’s not simple

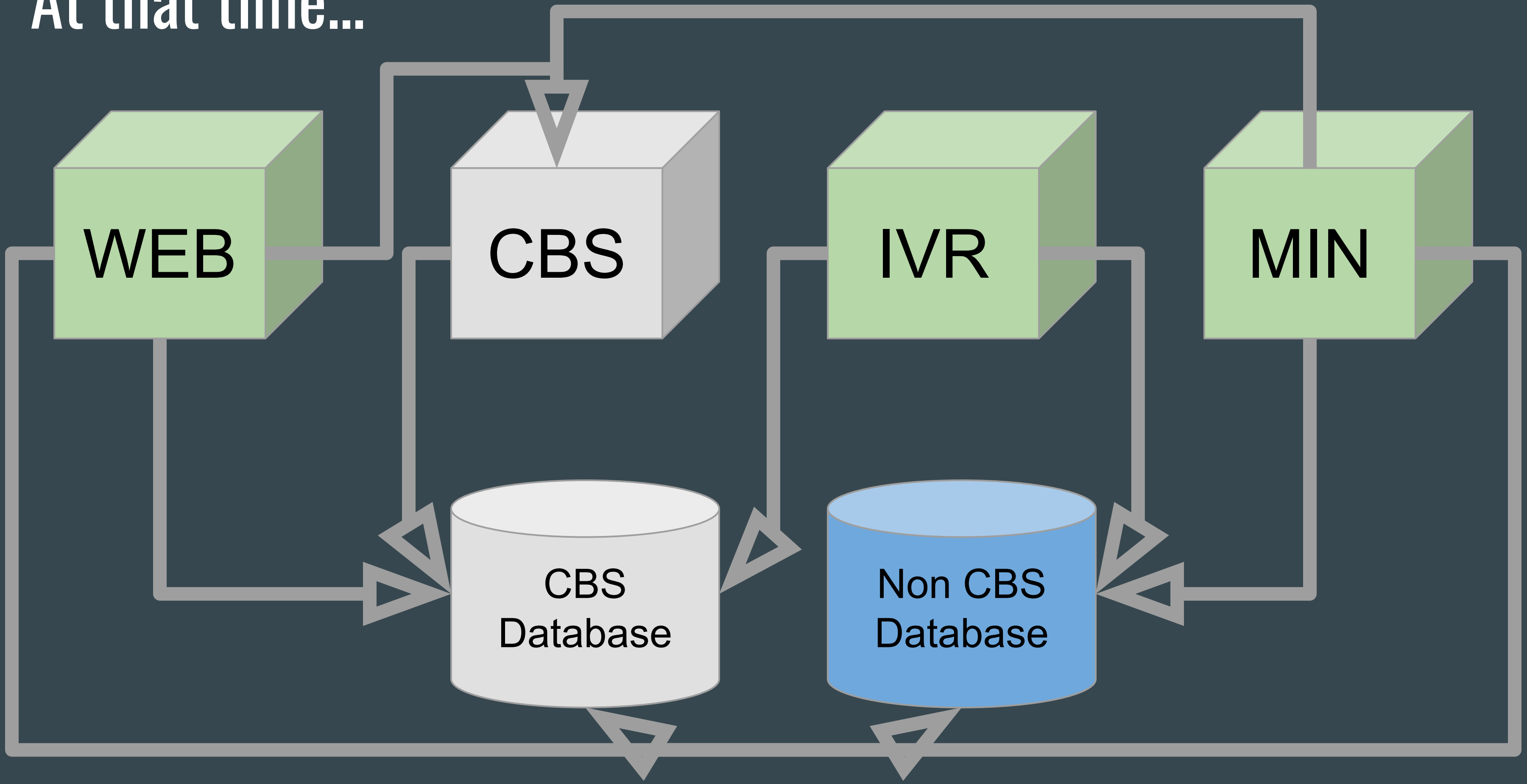
Web services

•••

A few years laters...

This is a dinosaur

At that time...



SOA(P)

SOA Principles

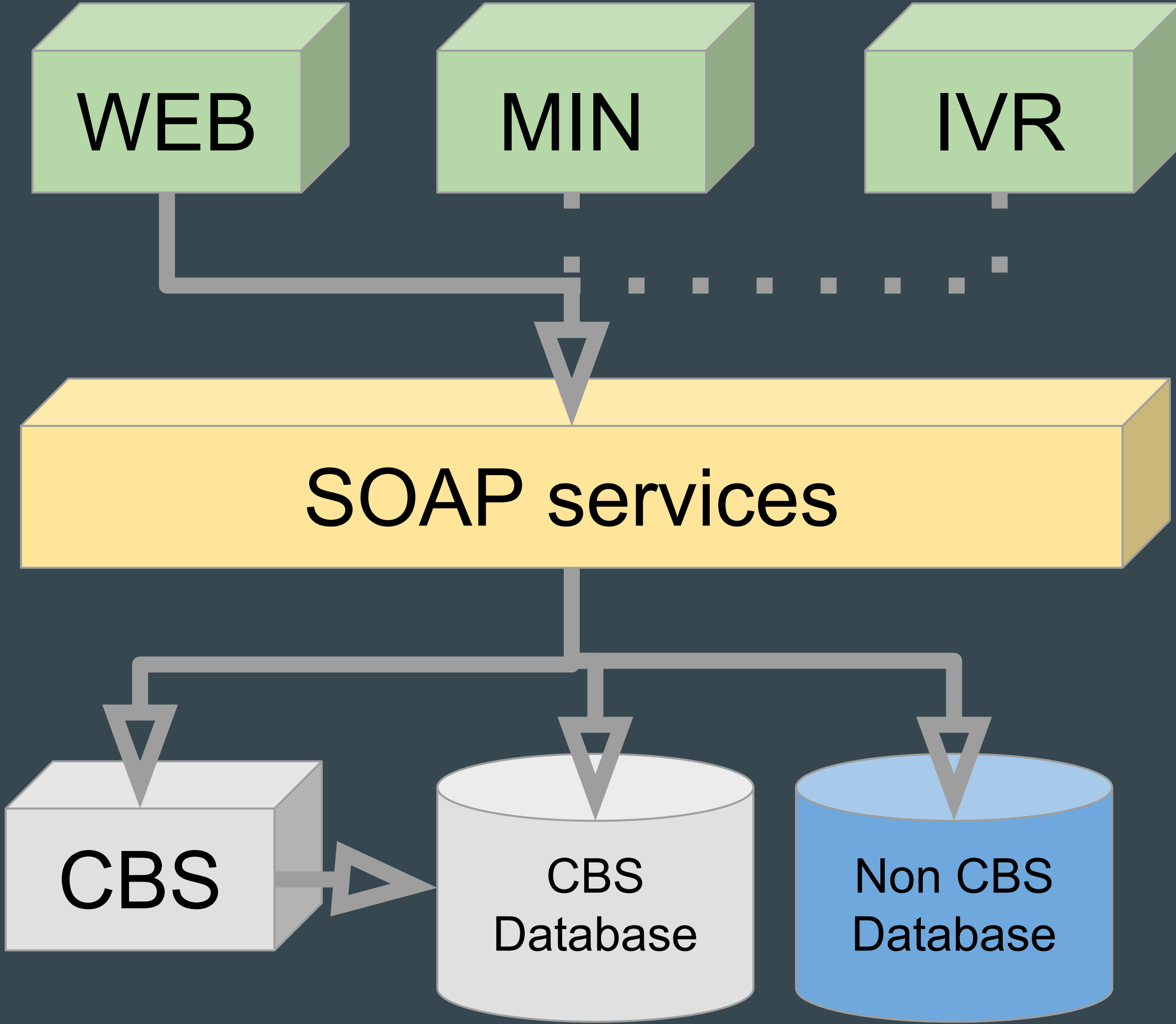
- Service
- Loose coupling
- Reusability

SOAP Protocol

- Use HTTP as a transport protocol
- XML based
- Input message contains the action to trigger and the data

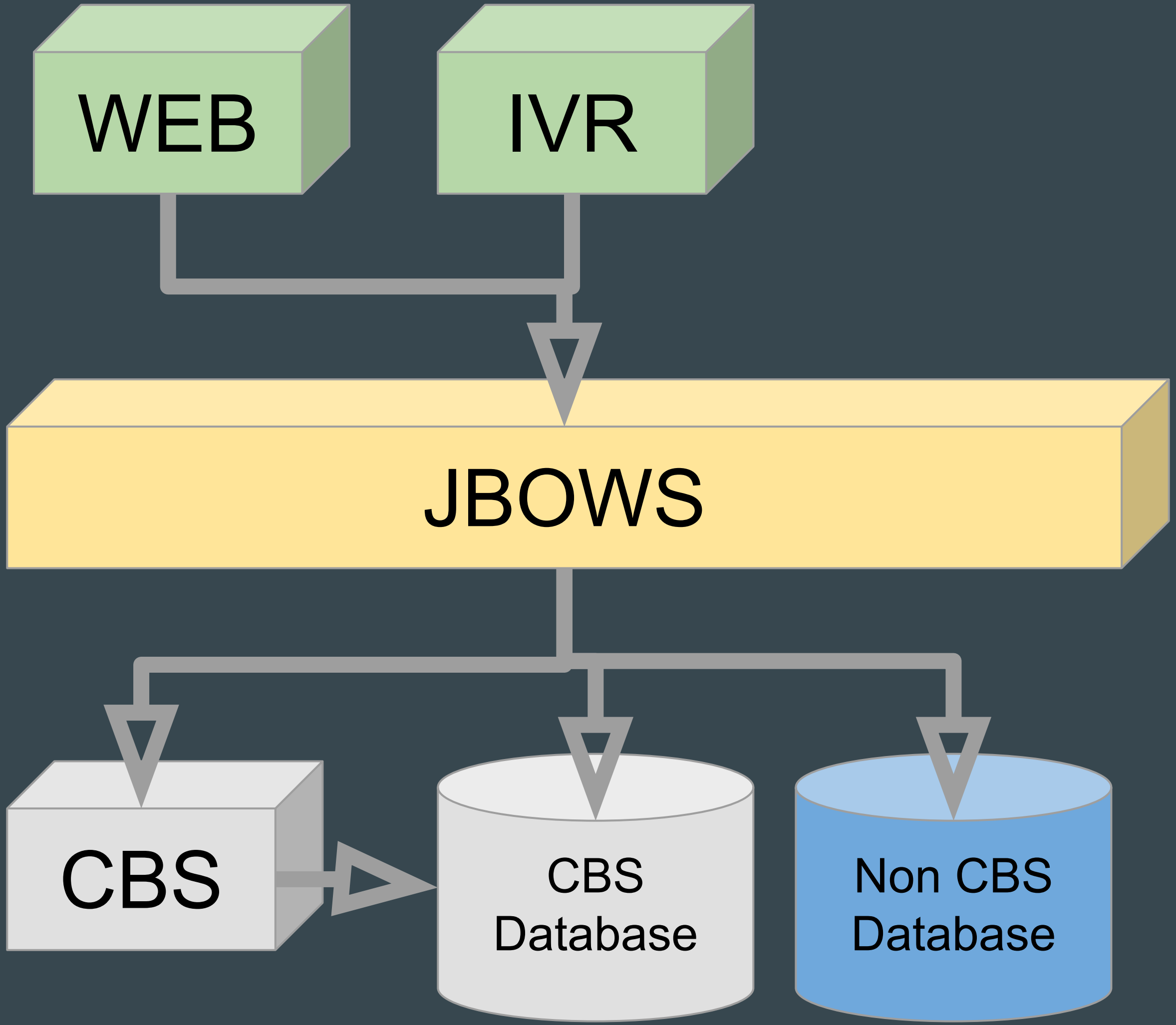
~~API~~ Service
ALL THE THINGS!
or you're fired!

SOA(P)



Several years later

SOA(P)



Is interfacing with SOA simple?

- Is it easy to use?
- It it easy to understand?
- Is the abstraction adapted to the audience?

```
>AJDSHQTREGJHSBUUACLYHHAEKRGD
I DON'T KNOW THAT WORD
>MY NAMEIS COLPANTS
I DON'T KNOW THAT WORD.
>FGDHUDAURULHWFA
I DON'T KNOW THAT WORD.
>CHRISQ
I DON'T KNOW THAT WORD.
>
>
```

Better but not awesome...

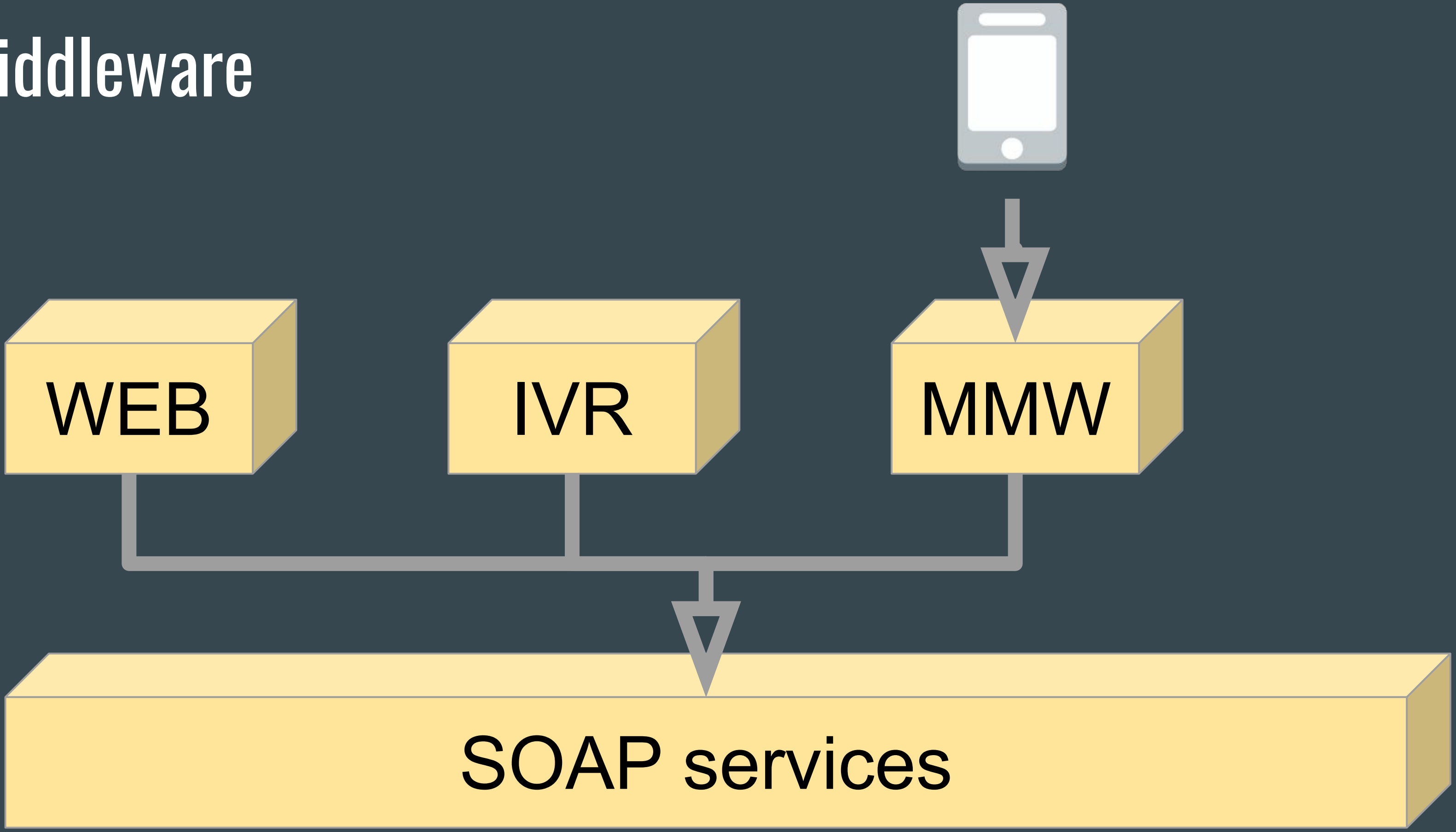
APIs

•••

A few years ago …

~~API~~ Mobile
ALL THE THINGS!

Middleware



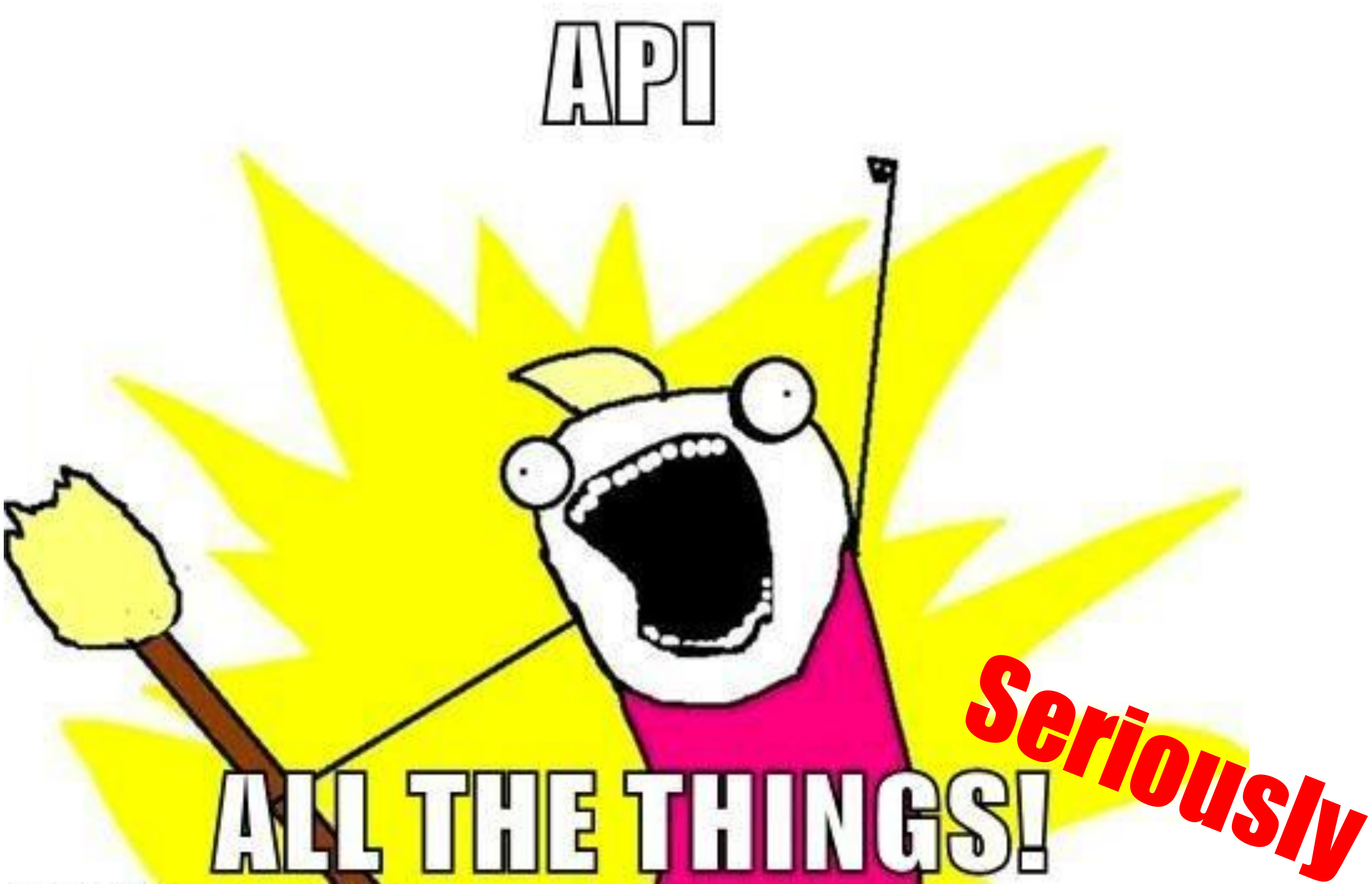
ROAST

ROAST API recipe

1. Take a SOAP/XML web service name add a / before it
2. Choose randomly an HTTP method between GET, PUT, POST, PATCH or DELETE, put it before the /
3. Transform input/output data from XML to JSON
4. If the method is GET or DELETE, put all parameters in query variables
5. And be sure to always return HTTP status 200

The mobile team discovering
GET /cancelTrfr?ztr1={id}

Several years later


API
ALL THE THINGS!
Seriously

A REST API

Design First

Use resource instead of actions

A list of wire transfers

/transfers

A wire transfers

/transfers/{transferId}

Use relevant HTTP method

Create a transfer

POST /transfers

Delete a pending wire transfer

DELETE /transfers/{transferId}

Update a customer email

PATCH /customers/me

Update a customer phone number

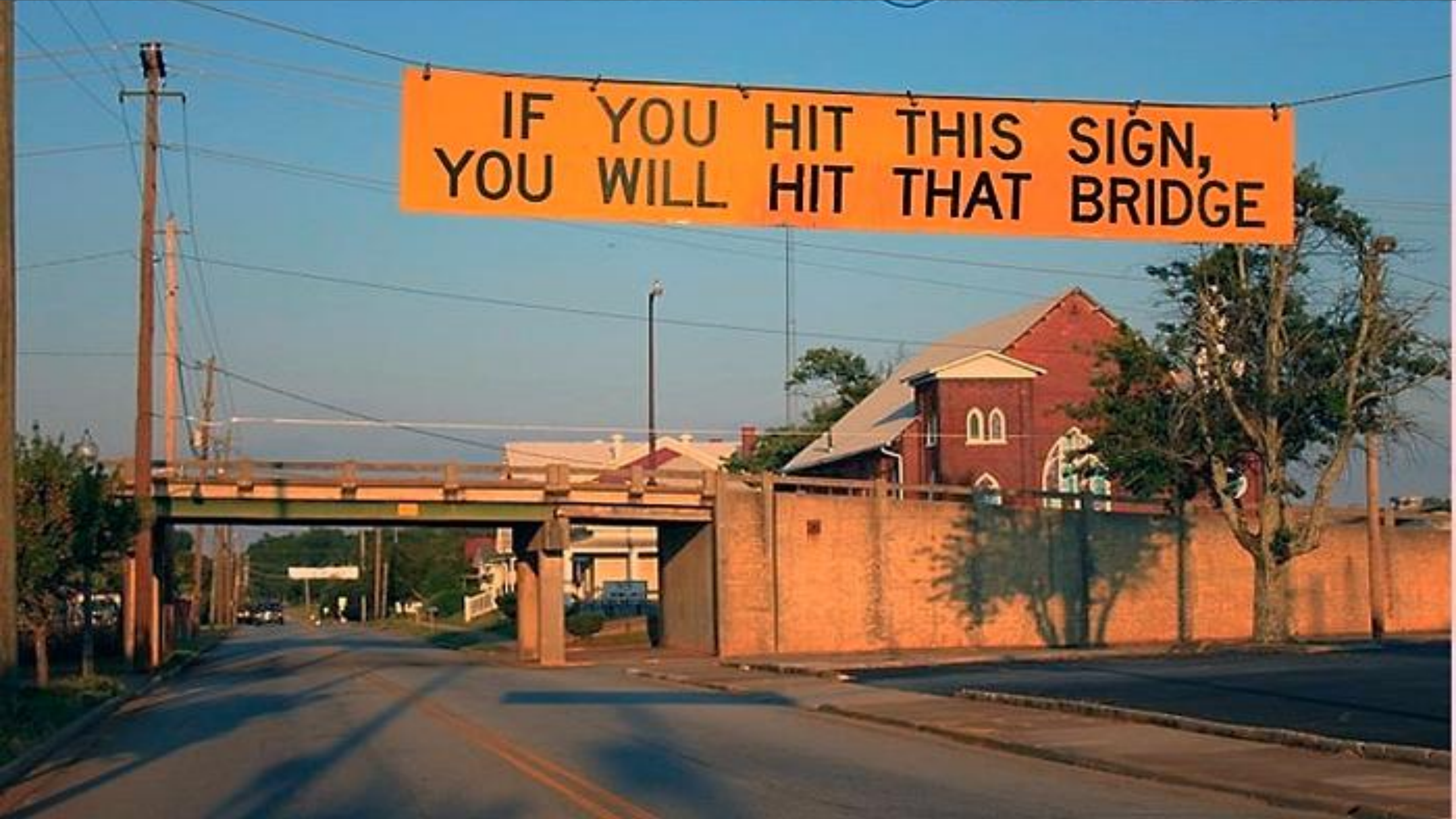
PATCH /customers/me

Use relevant HTTP status

403 Not enough money

503 No transfer between 1 am and 2 am

Provide hypermedia controls


IF YOU HIT THIS SIGN,
YOU WILL HIT THAT BRIDGE

GET /accounts/C1

```
{
    “id”: “C1”,
    “balance”: <how much money I have>,
    “actions”: <hypermedia controls>
    [
        { “name”: “transfer”,
          “method”: “POST”,
          “href”: “https://bank.com/transfers”}
    ]
}
```

Different ways to fill the set of actions

#1
Takeshi’s Castle
Knock Knock

403 Forbidden

```
{
    “code”: 1012,
    “message”: “Insufficient
    balance.”
}
```

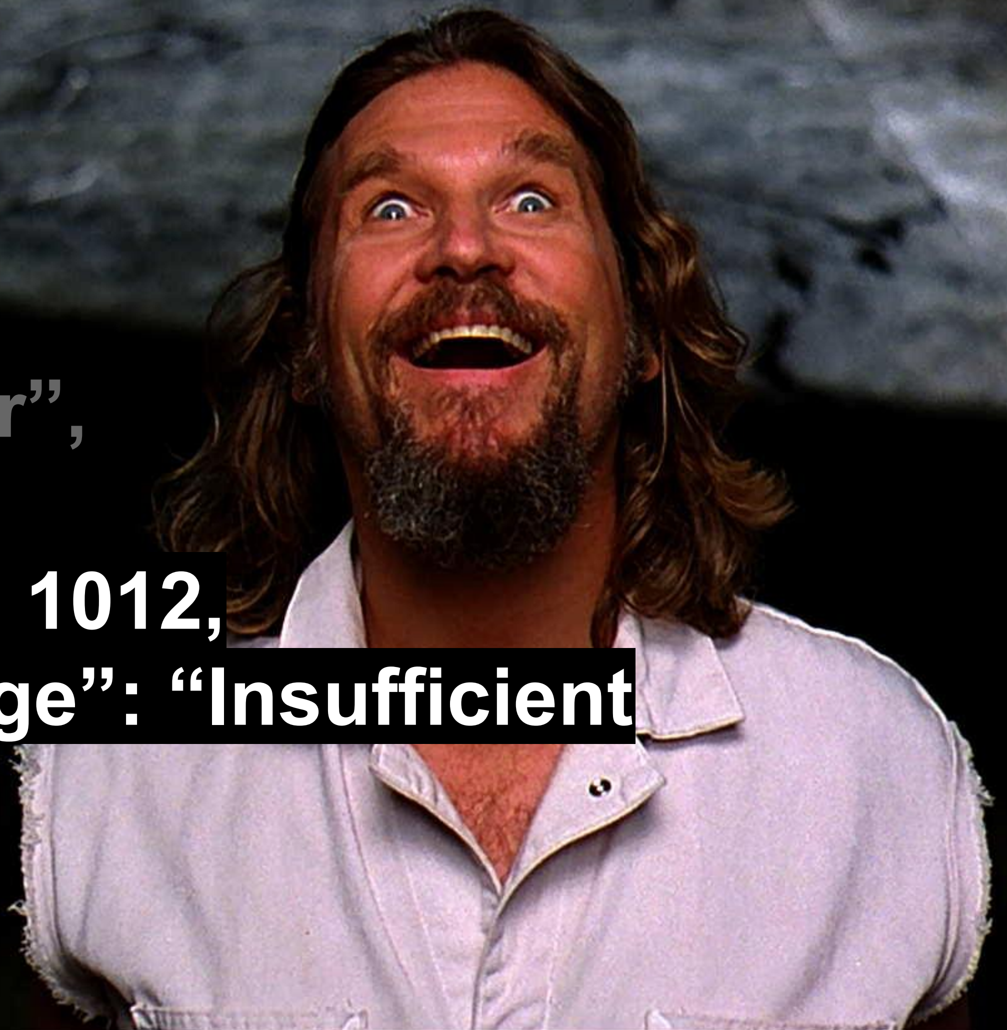
```
503 Service unavailable
{
    “code”: 1214,
    “message”: “No
    transfer between 1am
    and 2am.”
}
```

#2
This is bowling.
There are rules.

```
GET /accounts/C1
{
    “id”: “C1”,
    “balance”: -200,
    “actions”: [ ]
}
```

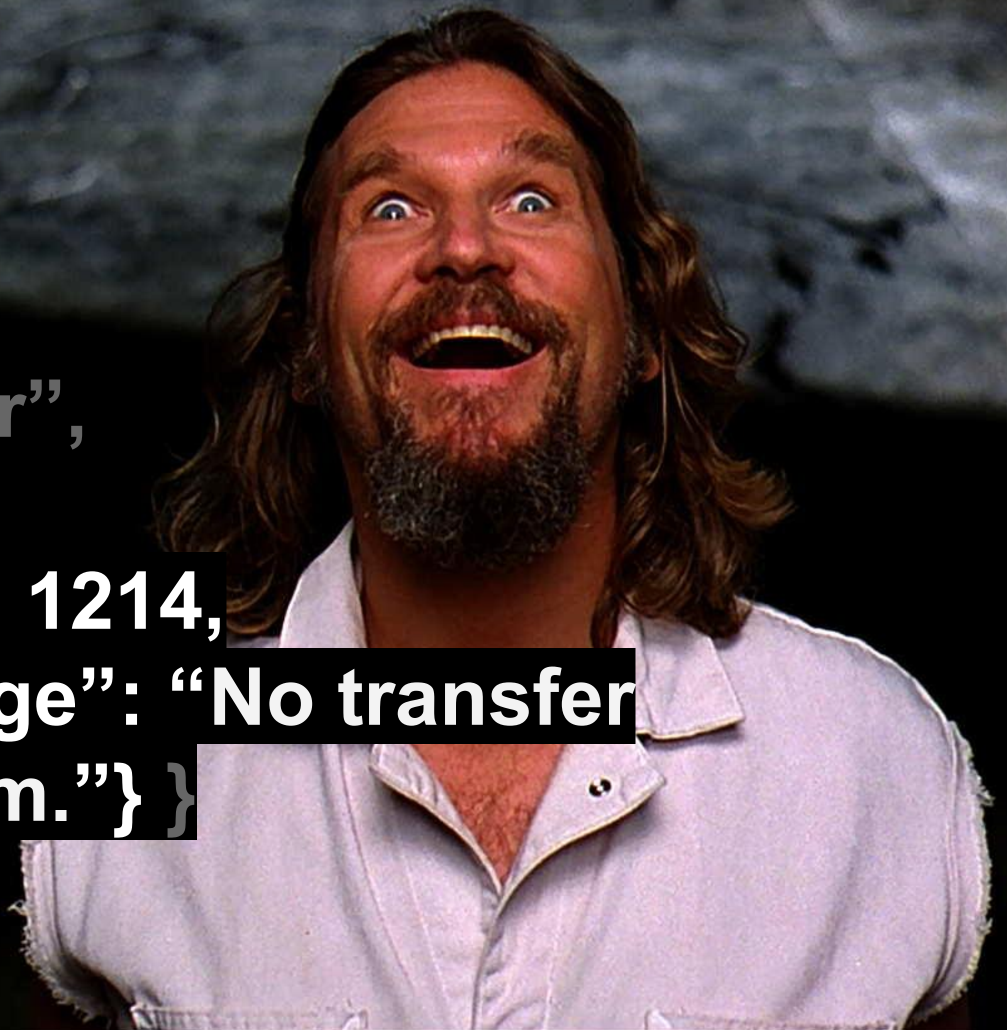
```
GET /accounts/C1
{
    "id": "C1",
    "balance": 20000,
    "actions": [ ]
}
```

#3
The Dude abides.



```
GET /accounts/C1
{   "id": "C1",
    "balance": -200,
    "actions":
    [
        { "name": "transfer",
          "status": 403,
          "error": { "code": 1012,
                     "message": "Insufficient
    balance."} }
    ]
}
```

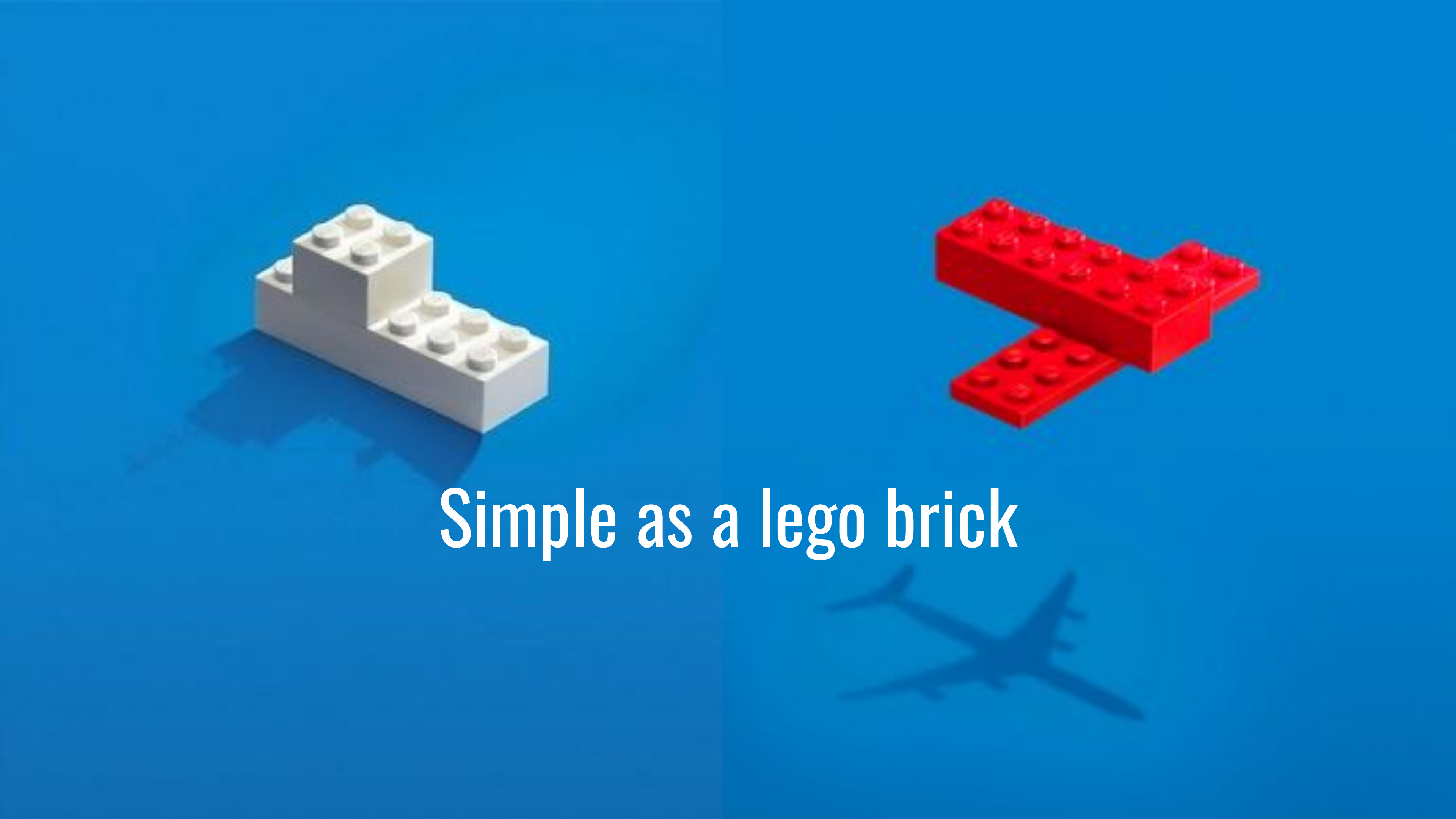
```
GET /accounts/C1
{   "id": "C1",
    "balance": 20000,
    "actions":
    [
        { "name": "transfer",
          "status": 503,
          "error": { "code": 1214,
                     "message": "No transfer
    between 1am and 2am."} }
    ]
}
```



It it really so simple to design an API?

Is interfacing with a RESTful API simple?

- Is it easy to use?
- It it easy to understand?
- Is the abstraction adapted to the audience?



Simple as a lego brick

The end?

•••

To the heights of API and beyond…