The Web, APIs, & Data

Prof. Craig Protzel
Interactive Media NYUAD
Fall 2017

Creating with WEB APIs

What is the **WEB**?

A DEFINITION

WIKIPEDIA
The Free Encyclopedia

World Wide Web

The World Wide Web (WWW) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. [1]. English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser computer program in 1990 while employed at CERN in Switzerland.[2][3]

The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet.[4][5][6]

A DEFINITION

WIKIPEDIA
The Free Encyclopedia

World Wide Web

The World Wide Web (WWW) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. [1], English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser computer program in 1990 while employed at CERN in Switzerland.[2][3]

The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet.[4][5][6]

A DEFINITION

WIKIPEDIA
The Free Encyclopedia

World Wide Web

The World Wide Web (WWW) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. [1]. English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser computer program in 1990 while employed at CERN in Switzerland.[2][3]

The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet.[4][5][6]

HTTP

A DEFINITION

WIKIPEDIA
The Free Encyclopedia

World Wide Web

The World Wide Web (WWW) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. [1]. English scientist Tim Berners-Lee invented the World Wide Web in 1989. He wrote the first web browser computer program in 1990 while employed at CERN in Switzerland.[2][3]

The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet.[4][5][6]

Hyper **T**ext **T**ransfer **P**rotocol

So how does it work?

WEB PROTOCOL

CLIENT

SERVER

WEB PROTOCOL

CLIENT

SERVER

Request

WEB PROTOCOL

CLIENT

SERVER

Request

Response

WEB PROTOCOL

CLIENT

SERVER

Request

Response

FRONTEND

HTTP

BACKEND

WEB PROTOCOL

CLIENT

SERVER

Request

Response

FRONTEND

HTTP

BACKEND

SOME FILES

GET/POST

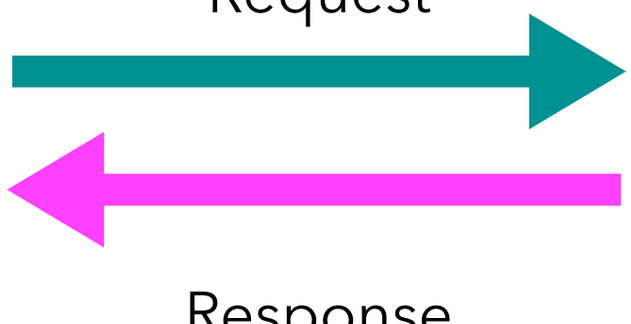
ALL THE FILES!!!

WEB PROTOCOL

CLIENT

SERVER

Request



Response

FRONTEND

HTTP

BACKEND

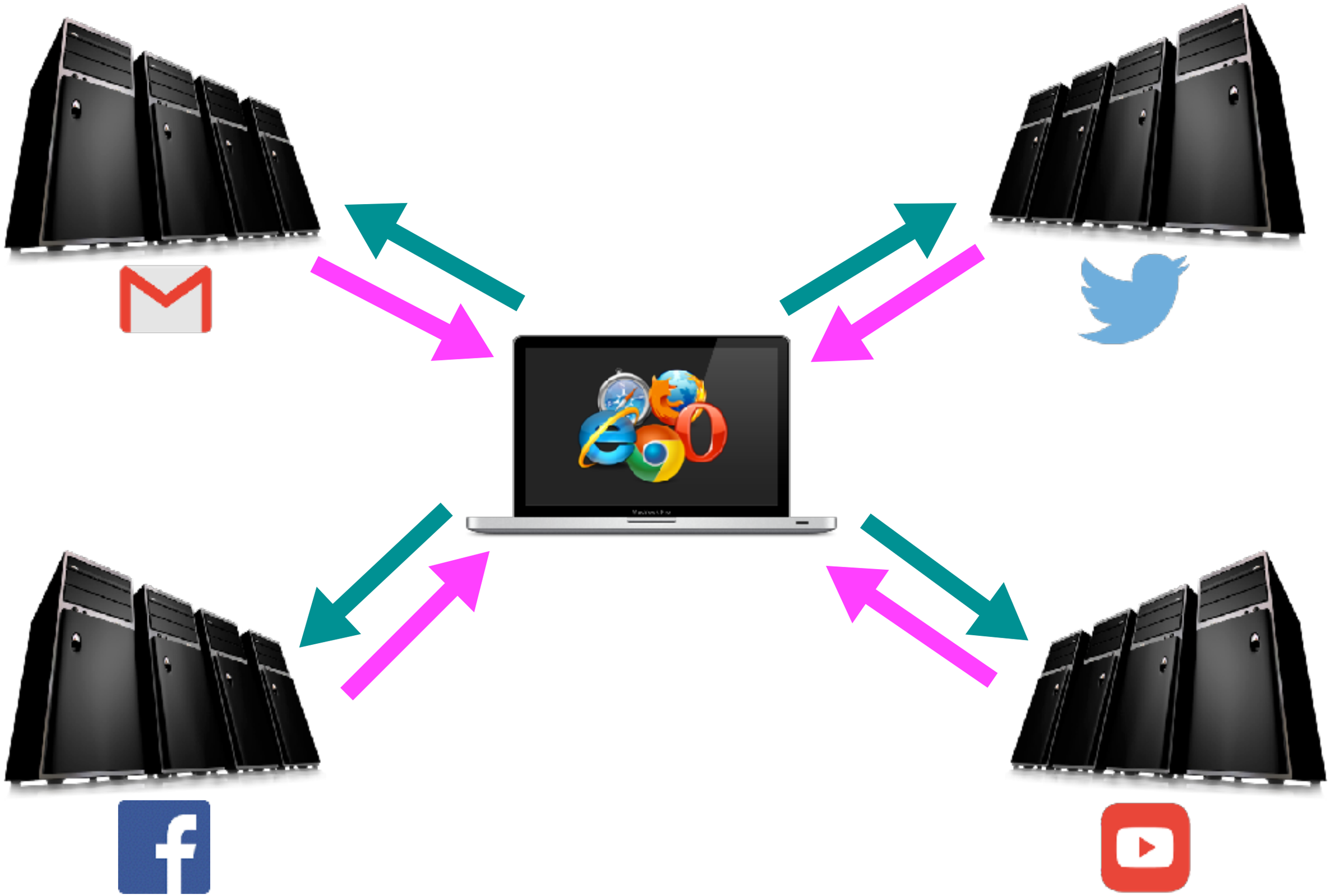
SOME FILES

GET/POST

ALL THE FILES!!!

CLIENT - SERVER

CLIENT - SERVER



HTTP REQUEST/RESPONSE

HTTP REQUEST

"I want my Facebook"

HTTP RESPONSE

"I want my Facebook"

"Here you go"

```
index.html
```

HTTP RESPONSE

"I want my Facebook"

"Here you go"

```
index.html

style.css
scripts.js
```

HTTP RESPONSE

"I want my Facebook"

"Here you go"

`index.html`

`style.css`
`scripts.js`

`objects.json`
`photo.img`
`media.mov`

THE FRONTEND

HTML

A Markup Language

CONTENT

What info is on the page

THE FRONTEND

HTML

A Markup Language

CONTENT

What info is on the page

CSS

A Markup Language

STYLE

How the info looks

THE FRONTEND

HTML

A Markup Language

CONTENT

What info is on the page

CSS

A Markup Language

STYLE

How the info looks

JAVASCRIPT

A Programming Language

INTERACTIVITY

How the info behaves

THE BACKEND

HOST

Computer + OS

NYU, Heroku,
DreamHost, Local

THE BACKEND

HOST
Computer + OS

NYU, Heroku,
DreamHost, Local

SERVER
Code + Application Files

Apache - PHP, Python - Flask,
Ruby - Sinatra, Node.js - Express
+ .html, .css, + .js files

THE BACKEND

HOST
Computer + OS

NYU, Heroku,
DreamHost, Local

SERVER
Code + Application Files

Apache - PHP, Python - Flask,
Ruby - Sinatra, Node.js - Express
+ .html, .css, + .js files

DATABASE
Data Files + Query Language

ORM, non-ORM
mySQL, mongo
SQL, mongoose
tables, XML, JSON

MASHUPS CLASS STACK

CLIENT SIDE

Content ____ HTML
Style ____ CSS
Interaction ____ Javascript
JS Libraries ____ jQuery*
Data Source ____ **Open API**
Data Service ____ **Open API**

* js libraries - d3, p5, three, & more

SERVER SIDE

Host ____ Heroku
Server ____ NodeJS
Application ____ Express
Database ____ CouchDB
DB Host ____ Cloudant
DB Queries ____ Request

MASHUPS CLASS STACK

CLIENT SIDE

Content ____ HTML
Style ____ CSS
Interaction ____ Javascript
JS Libraries ____ jQuery*
Data Source ____ **Open API**
Data Service ____ **Open API**

* js libraries - d3, p5, three, & more

SERVER SIDE

Host ____ Heroku
Server ____ NodeJS
Application ____ Express
Database ____ CouchDB
DB Host ____ Cloudant
DB Queries ____ Request

PARTS I & II

Weeks 1 - 8

MASHUPS CLASS STACK

CLIENT SIDE

Content ____ HTML
Style ____ CSS
Interaction ____ Javascript
JS Libraries ____ jQuery*
Data Source ____ **Open API**
Data Service ____ **Open API**

* js libraries - d3, p5, three, & more

SERVER SIDE

Host ____ Heroku
Server ____ NodeJS
Application ____ Express
Database ____ CouchDB
DB Host ____ Cloudant
DB Queries ____ Request

PARTS I & II

Weeks 1 - 8

PART III

Classes 9 - 14

How about some examples?

What is an **API**?

Application **P**rogramming **I**nterface

A set of requirements that govern how one application can talk to another

PROCESSING API

Cover

Download

Exhibition

Reference
Libraries
Tools
Environment

Tutorials
Examples
Books

Overview
People
Foundation

Shop

» Forum
» GitHub
» Issues
» Wiki
» FAQ
» Twitter
» Facebook

Reference. The Processing Language was designed to facilitate the creation of sophisticated visual structures.

Structure

() (parentheses)
, (comma)
. (dot)
/* */ (multiline comment)
/** */ (doc comment)
// (comment)
; (semicolon)
= (assign)
[] (array access)
{} (curly braces)
catch
class
draw()
exit()
extends
false
final
implements
import
loop()
new
noLoop()
null
popStyle()
private
public
pushStyle()
redraw()
return
setup()
static
super
this
true
try
void

Environment

Shape

createShape()
loadShape()
PShape

2D Primitives

arc()
ellipse()
line()
point()
quad()
rect()
triangle()

Curves

bezier()
bezierDetail()
bezierPoint()
bezierTangent()
curve()
curveDetail()
curvePoint()
curveTangent()
curveTightness()

3D Primitives

box()
sphere()
sphereDetail()

Attributes

ellipseMode()
noSmooth()
rectMode()
smooth()
strokeCap()
strokeJoin()
strokeWeight()

Color

Setting

background()
clear()
colorMode()
fill()
noFill()
noStroke()
stroke()

Creating & Reading

alpha()
blue()
brightness()
color()
green()
hue()
lerpColor()
red()
saturation()

Image

createImage()
PImage

Loading & Displaying

image()
imageMode()
loadImage()
noTint()
requestImage()
tint()

Textures

texture()
textureMode()
textureWrap()

MENU ANALOGY

API LANDSCAPE

DEVICE API - access accelerometer data on phone

OS API - cut and paste from Adobe Illustrator to MS Word

FRAMEWORK API - use Processing functions to execute Java

PLATFORM API - leverage the canvas API in the browser

DATA API - query a list of images from Flickr

RESOURCE API - embed a Google map on a web page

SERVICE API - send IBM Watson a data set to analyze

API FOR APIs - use the temboo SDK to access 100+ APIs

API LANDSCAPE

DEVICE API - access accelerometer data on phone

OS API - cut and paste from Adobe Illustrator to MS Word

FRAMEWORK API - use Processing functions to execute Java

PLATFORM API - leverage the canvas API in the browser

DATA API - **query** a list of images from Flickr

RESOURCE API - **embed** a Google map on a web page

SERVICE API - **send** IBM Watson a data set to analyze

API FOR APIs - use the temboo SDK to access 100+ APIs

“Public Web” APIs

PUBLIC WEB APIs

URLs that give access to data, resources, and services from a public web server

PUBLIC WEB APIs

URLs that give access to data, resources, and services from a public server

`http://api.nyu.edu/courses/mashups`

PUBLIC WEB APIs

URLs that give access to data, resources, and services from a public server

`http://api.nyu.edu/courses/mashups`

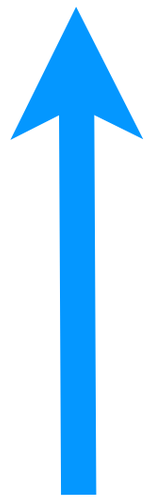
Network Protocol

PUBLIC WEB APIs

URLs that give access to data, resources, and services from a public server

`http://api.nyu.edu/courses/mashups`

Network Protocol



Host Name/Address

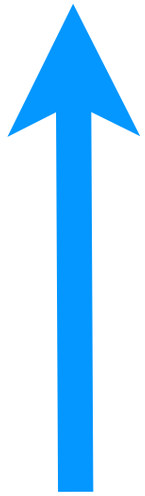
(IP - DNS)

PUBLIC WEB APIs

URLs that give access to data, resources, and services from a public server

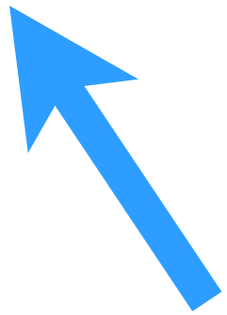
`http://api.nyu.edu/courses/mashups`

Network Protocol



Host Name/Address

(IP - DNS)



File/Resource Location

uri / route / path / endpoint

PUBLIC WEB APIs

URLs that give access to **data**, resources, and services from a public server

`http://api.nyu.edu/courses/mashups`

?

PUBLIC WEB APIs

URLs that give access to **data**, resources, and services from a public server

`http://api.nyu.edu/courses/mashups`

```
{
  “school” : “NYUAD”,
  “program” : “Interactive Media”,
  “level” : “undergraduate”,
  “instructor” : “Craig Protzel”,
  “units” : 4
}
```

JSON - JavaScript Object Notation

```
var mashups = {
  “school” : “NYUAD”,
  “program” : “Interactive Media”,
  “level” : “undergraduate”,
  “instructor” : “Craig Protzel”,
  “units” : 4
}
```

API DATA

```
http://api.nyu.edu/courses/mashups?year=2017&semester=fall
```

API DATA

`http://api.nyu.edu/courses/mashups?year=2017&semester=fall`

Query Parameters

API DATA

`http://api.nyu.edu/courses/mashups?year=2017&semester=fall`

API DATA

http://api.nyu.edu/courses/mashups?year=2017&semester=fall


```
{
  “school” : “NYUAD”,
  “program” : “Interactive Media”,
  “level” : “undergraduate”,
  “instructor” : “Craig Protzel”,
  “units” : 4,
  “location” : “C3 Room 029”,
  “schedule”: [
    {
      “day” : “Tuesday”,
      “time” : “1440”,
      “duration” : 75
    },
    {
      “day” : “Thursday”,
      “time” : “1440”,
      “duration” : 160
    }
  ]
}
```

openweathermap.org

ESSA BAGEL ANALOGY

ESSA BAGEL ANALOGY

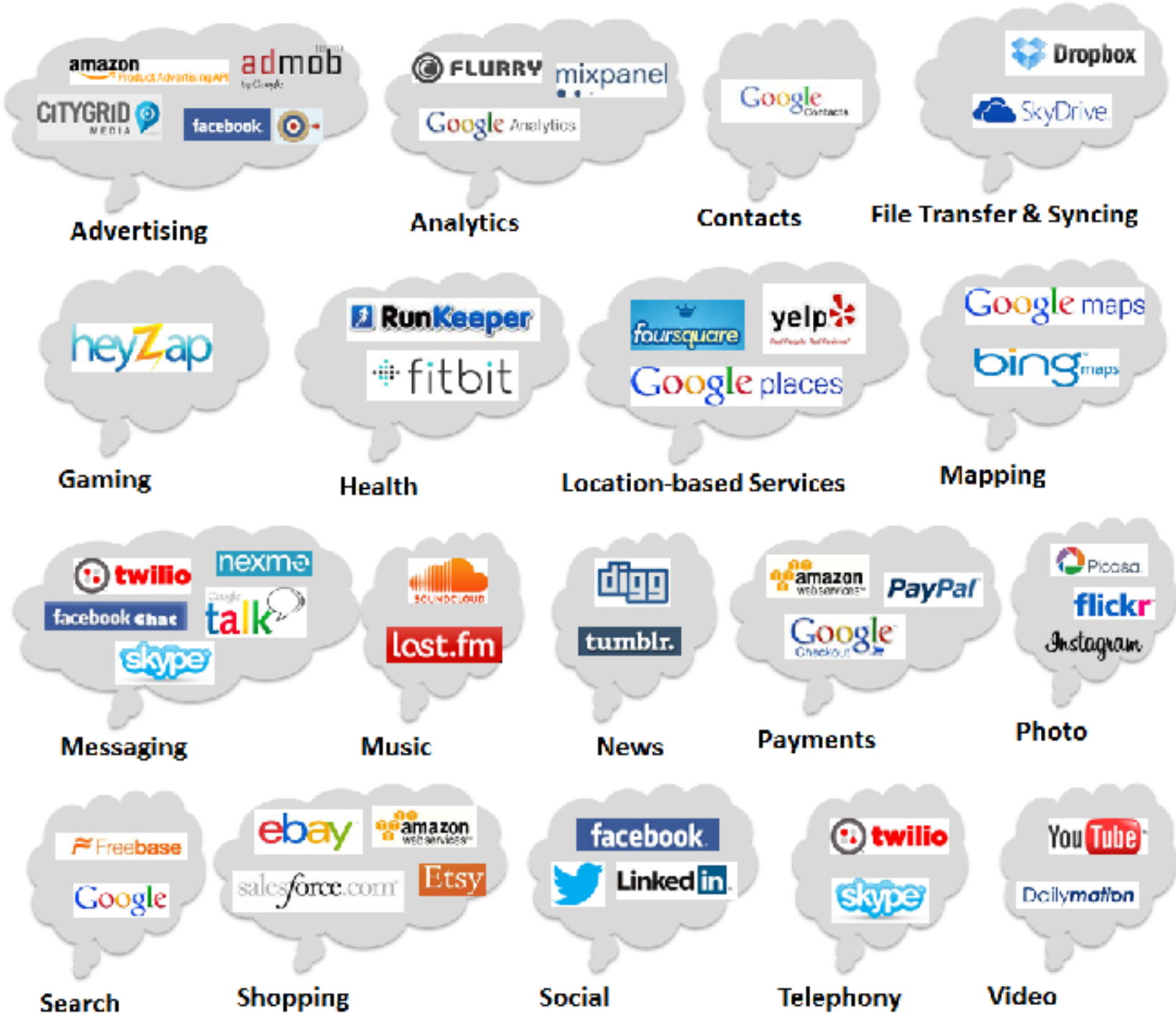
ESSA BAGEL "NORMAL" BIZ

ESSA BAGEL ANALOGY

ESSA BAGEL API

ESSA BAGEL "NORMAL" BIZ

MORE EXAMPLES



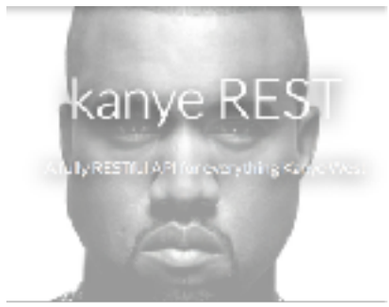
http://www.openi-ict.eu/openi-api-framework-part-i-studying-the-landscape-of-cloud-based-services/

Why?

OPEN API SPECTRUM

URL ONLY

HealthCare.gov



URL + KEY

OpenWeatherMap

The New York Times

URL + KEY *or*
URL + KEY + AUTHENTICATION

Instagram

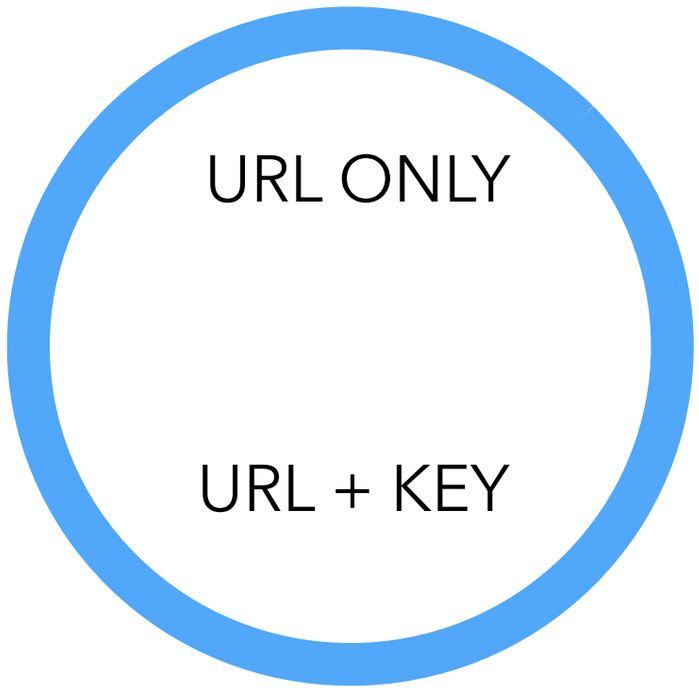
YouTube

URL + KEY + AUTHENTICATION

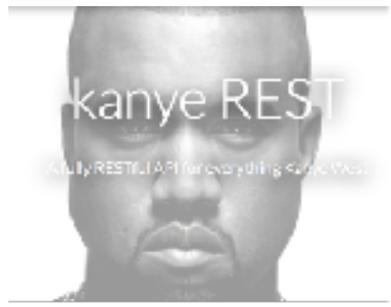
twitter

EVERNOTE

OPEN API SPECTRUM



HealthCare.gov



OpenWeatherMap

The New York Times

URL + KEY *or*
URL + KEY + AUTHENTICATION

Instagram

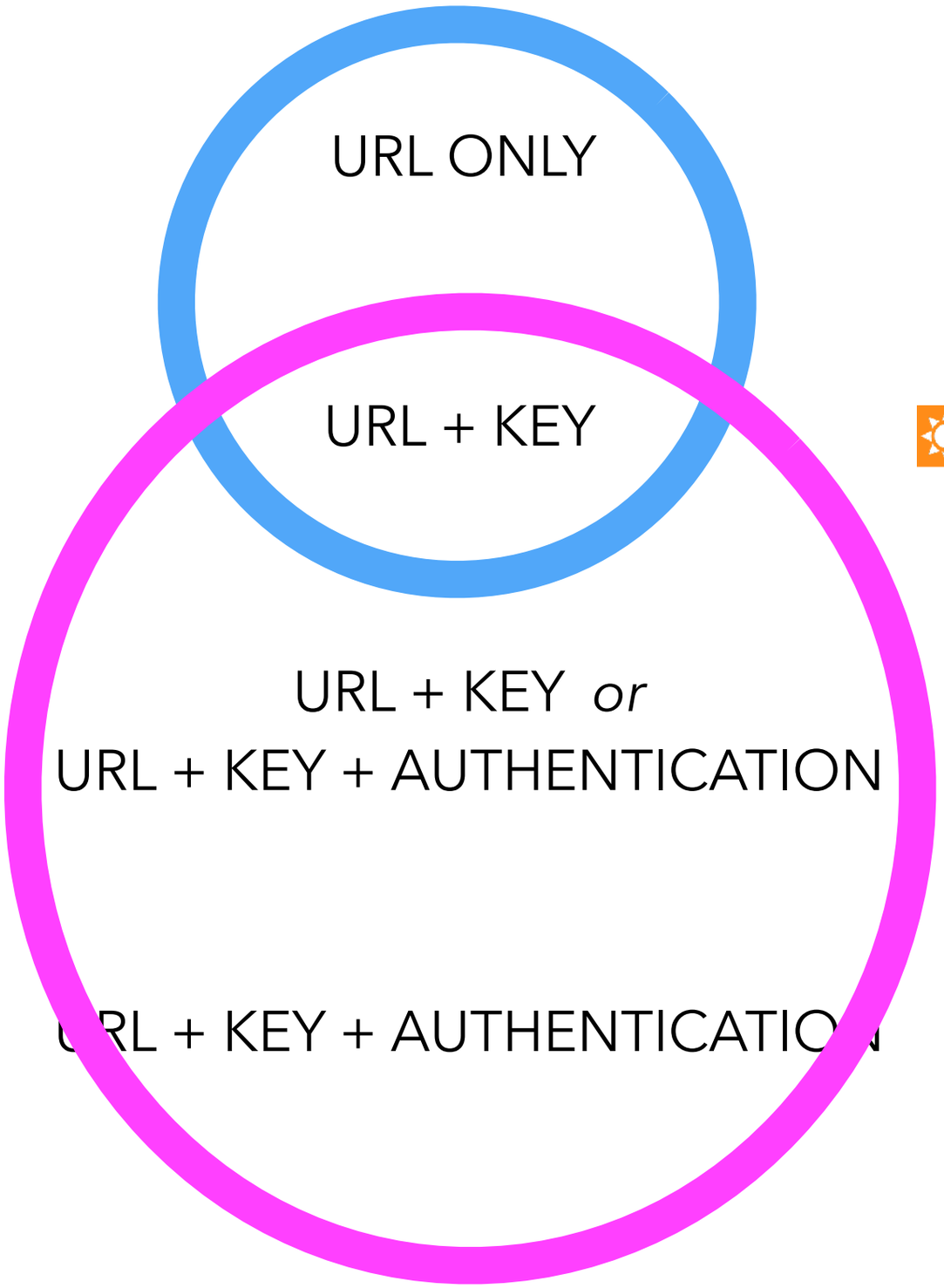
YouTube

URL + KEY + AUTHENTICATION

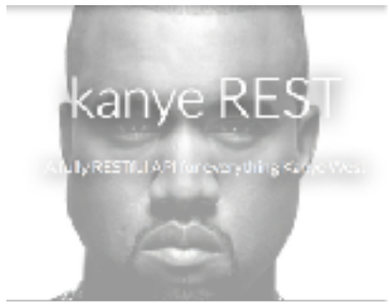
twitter

EVERNOTE

OPEN API SPECTRUM



HealthCare.gov



OpenWeatherMap

The New York Times

Instagram

YouTube

twitter

EVERNOTE

IN ACTION

HowManyPeopleAreInSpaceRightNow

MASHUPS FLOW

1)

Client

REQUEST - spacemashup.com

SpaceMashup Server
(local)

MASHUPS FLOW

1)

Client

REQUEST - spacemashup.com

SpaceMashup Server
(local)

2)

Client

RESPONSE - HTML, CSS, JS, & media

SpaceMashup Server
(local)

MASHUPS FLOW

1)

Client

REQUEST - spacemashup.com

SpaceMashup Server (local)

2)

Client

RESPONSE - HTML, CSS, JS, & media

SpaceMashup Server (local)

3)

Client

REQUEST- api.open-notify.org/astros.json

Open-Notify Server

MASHUPS FLOW

1)

Client

REQUEST - spacemashup.com

SpaceMashup Server (local)

2)

Client

RESPONSE - HTML, CSS, JS, & media

SpaceMashup Server (local)

3)

Client

REQUEST- api.open-notify.org/astros.json

Open-Notify Server

4)

Client

RESPONSE - json data

Open-Notify Server

HOMEWORK

github.com/craigprotzel/Mashups