ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

RICARDO QUEIRÓS

INDEX

- ME
- MOTIVATION
- WEB SERVICES
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

INDEX

- ME
- MOTIVATION
- WEB SERVICES
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ME

RICARDO QUEIRÓS

- 42 years old
- Born in PORTO
- PhD in Computer Science (FCUP)
- Professor (ESMAD/P.PORTO)
- Researcher (CRACS/INESC-TEC)
- Web & Mobile Developer

ME

RICARDO QUEIRÓS

- Main activities/hobbies:
  - Teach computer programming
  - Do research at CRACS/INESC-TEC
    - E-Learning applications, standards and Interoperability
    - Computer Science Education
    - Programming Languages processing
    - XML processing and Architectural Integration
    - Mobile computing
  - Code (Java, C#, JavaScript, C, Sass,…)
  - Play soccer and running

- More about me: http://www.dcc.fc.up.pt/~rqueiros

ME

RICARDO QUEIRÓS

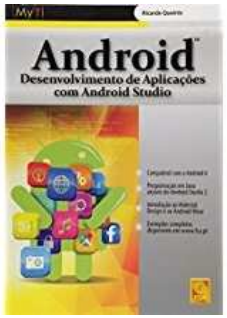
- Book writer

2008

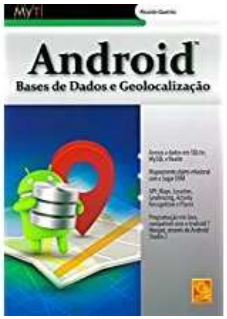
2013

2014

2015



2016

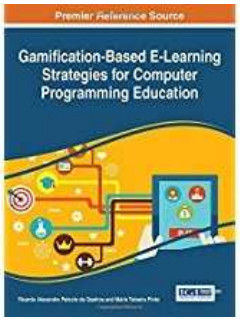


2016

2017

2012

2015



2016

ME

MY SCHOOL

- ESMAD (Media Arts and Design Superior School)
  - One of the schools of the POLYTECHNIC OF PORTO
  - In the middle of 2 cities: Vila do Conde & Póvoa de Varzim

P.PORTO

ESCOLA
SUPERIOR
DE MEDIA ARTES
E DESIGN

ME

MY SCHOOL

ME

MY SCHOOL

https://vimeo.com/212716938

ME

MY CITY

- PORTO
    - The second biggest city in Portugal
    - ESMAD is located 30km from Porto

ME

MY CITY

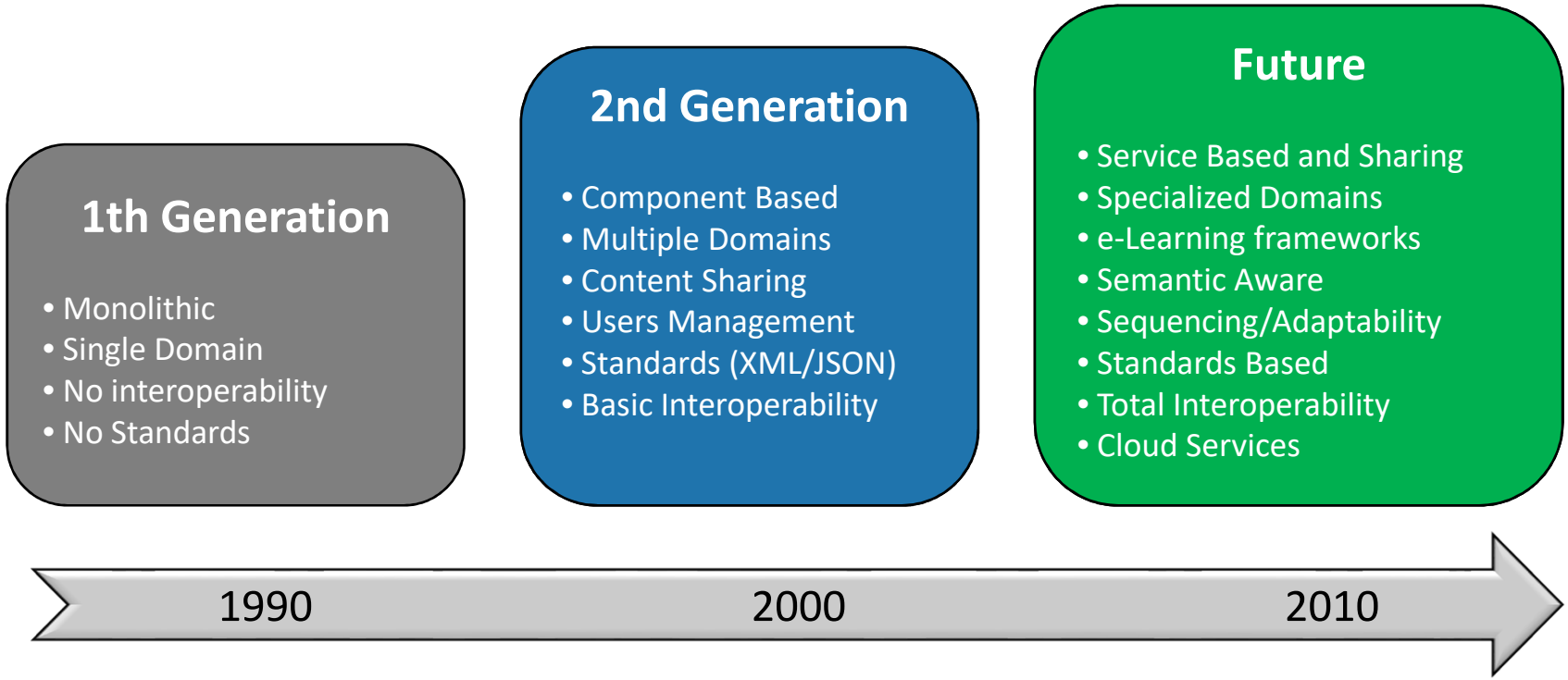
INDEX

- ME
- MOTIVATION
- WEB SERVICES
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

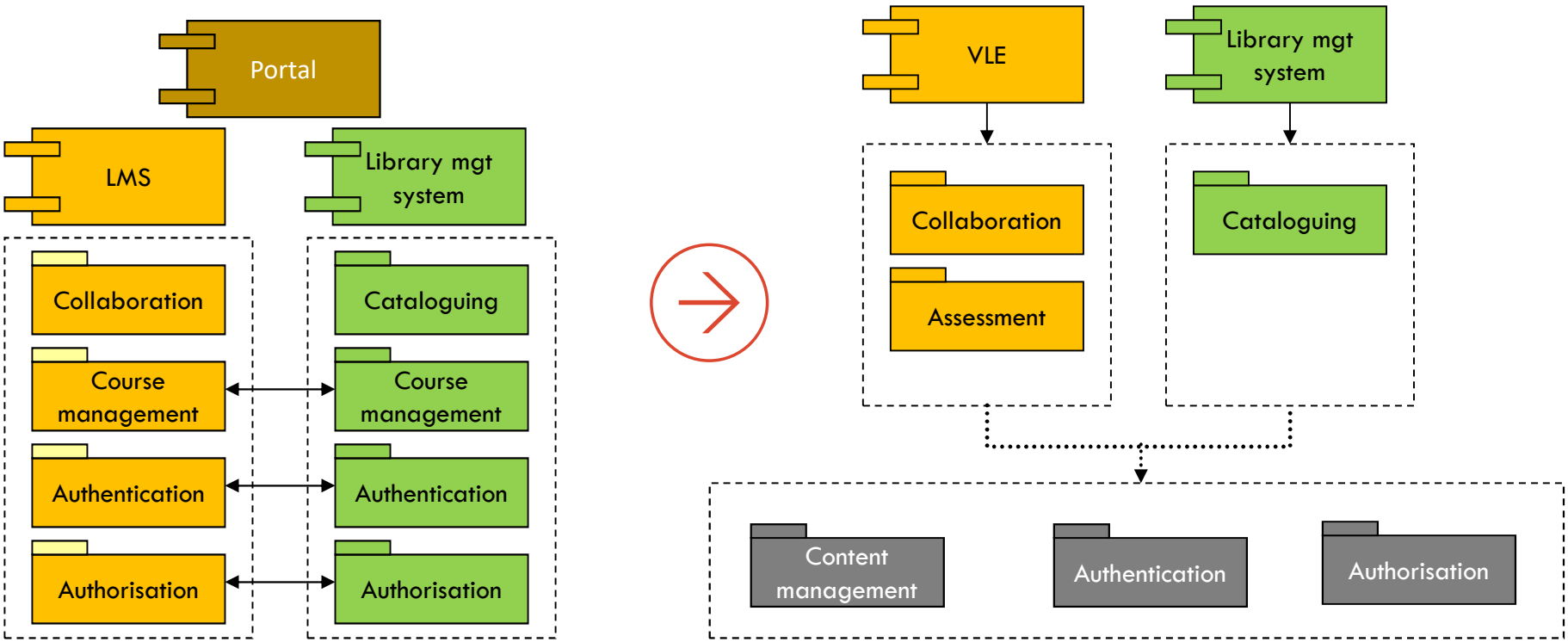
- From monolithic systems to service orientation



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

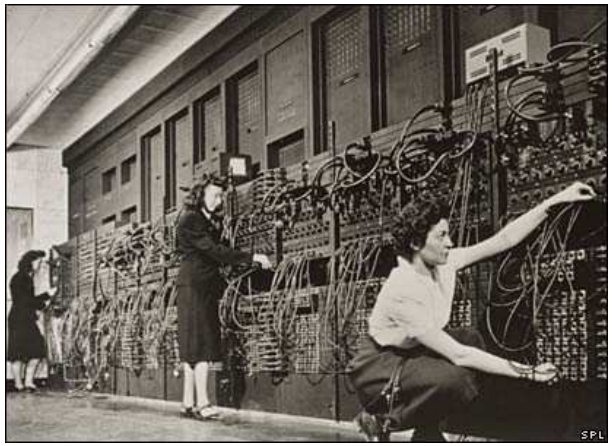
- From monolithic systems to service orientation



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

- From monolithic systems to service orientation
- Hardware/software evolution



ENIAC (1946)

Galaxy S7 (2016)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

- From monolithic systems to service orientation
- Hardware/software evolution

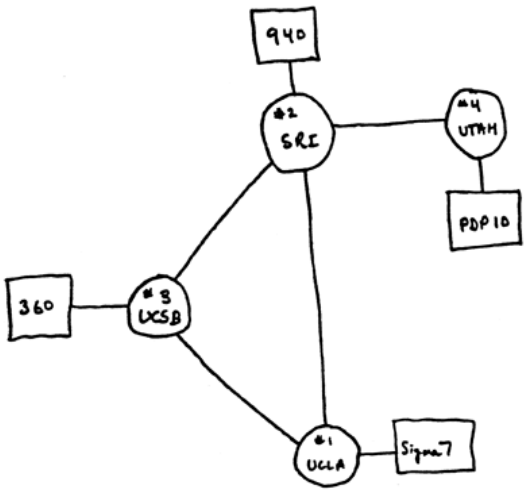
RAMAC HD 5MB (1956)

SSD 2.5’’ SATA III 2TB (2016)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

- From monolithic systems to service orientation
- Hardware/software evolution



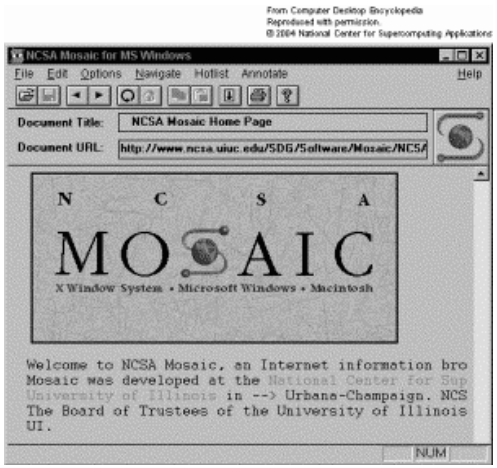
ARPANET (1968)

Web 2.0 (2004)

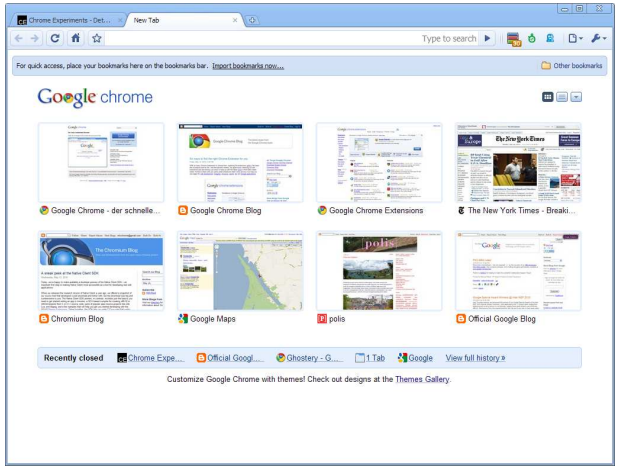
ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

- From monolithic systems to service orientation
- Hardware/software evolution



Mosaic (1993)

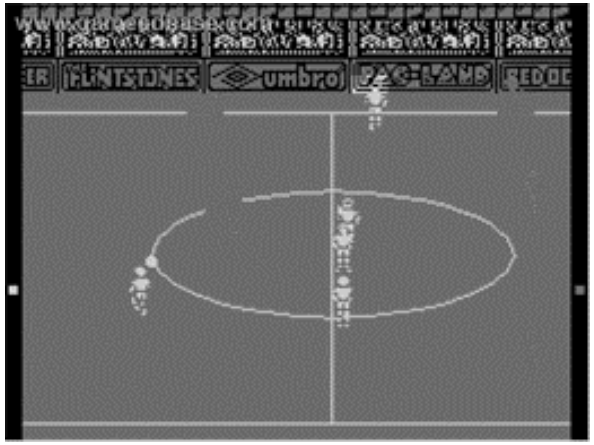


Google Chrome (2017)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

- From monolithic systems to service orientation
- Hardware/software evolution



Peter Beardsley's International Football (1988)

PES (2016)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

- From monolithic systems to service orientation
- Hardware/software evolution
- Heterogenous systems and languages

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

MOTIVATION

- From monolithic systems to service orientation
- Hardware/software evolution
- Heterogenous systems and languages

So, how to put these disparate systems to communicate?

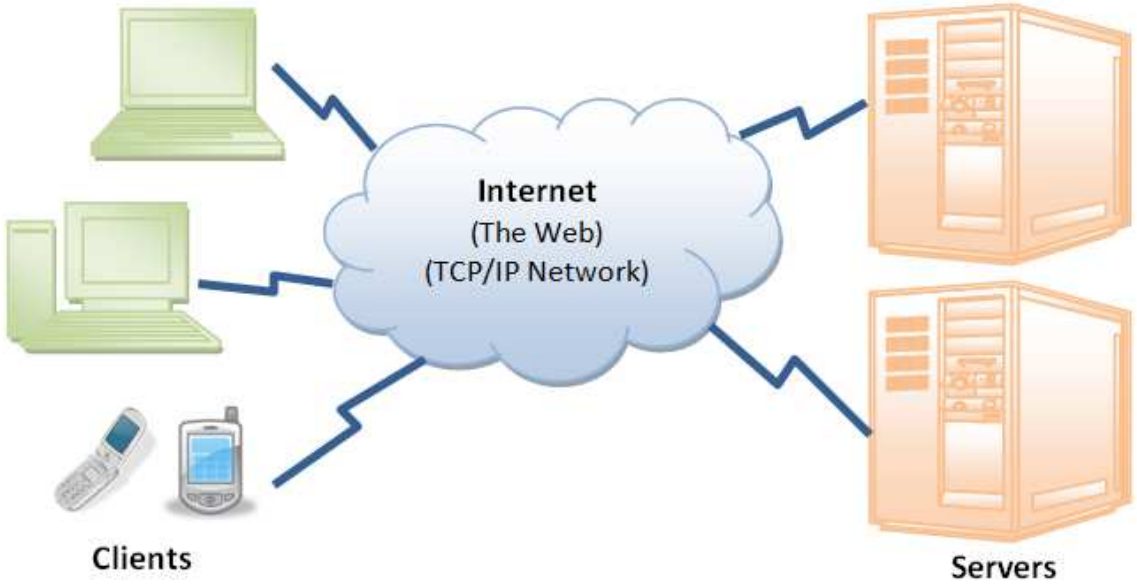
INDEX

- ME
- MOTIVATION
- WEB SERVICES
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- Internet (or WWW) is a massive distributed client/server information system



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST (REpresentational State Transfer)
  - is an architectural style for network-based software architectures
  - defined by Roy Fielding in his 2000 PhD dissertation *"Architectural Styles and the Design of Network-based Software Architectures"*

  - REST as a uniform interface:
    1. Identification of resources
    2. Manipulation of resources via representations
    3. Self-descriptive messages

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST (REpresentational State Transfer)
  - is an architectural style for network-based software architectures
  - defined by Roy Fielding in his 2000 PhD dissertation *"Architectural Styles and the Design of Network-based Software Architectures"*
  - REST as a uniform interface:
    1. Identification of resources
    2. Manipulation of resources via representations
    3. Self-descriptive messages

RESOURCES AS URIs
http://server/persons/12

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST (REpresentational State Transfer)
  - is an architectural style for network-based software architectures
  - defined by Roy Fielding in his 2000 PhD dissertation *"Architectural Styles and the Design of Network-based Software Architectures"*
  - REST as a uniform interface:
    1. Identification of resources
    2. Manipulation of resources via representations
    3. Self-descriptive messages

RESOURCES AS URIs
http://server/persons/12

JSON, XML, …

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST (REpresentational State Transfer)
  - is an architectural style for network-based software architectures
  - defined by Roy Fielding in his 2000 PhD dissertation *"Architectural Styles and the Design of Network-based Software Architectures"*
  - REST as a uniform interface:
    1. Identification of resources
    2. Manipulation of resources via representations
    3. Self-descriptive messages

RESOURCES AS URIs
http://server/persons/12

JSON, XML, …

HTTP
METHODS: GET, POST, PUT, DELETE

WEB SERVICES

- REST | Identification of resources
  - REST requires each resource to have at least one URI
  - URI (Uniform Resource Identifier) is used to uniquely identify a resource on the Web
  - Example of a resource:

```
http://server/persons/12
```

  - Prefer nouns to verbs
    - Nouns refer to resources
    - Resources are handled with HTTP verbs
  - Prefer plural forms: /persons/12 vs /person/12
  - Name convention?
    - UpperCamelCase or lowerCamelCase or snake_case or dashed-snaked-case? ☺
    - Prefer lowercase and snake_case
    - Underscores seem more common in APIs

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Representation of resources
  - Focus on REST services is on **resources** and how to provide access to them
  - A resource can easily be thought of as an object as in OOP
  - Tasks:
    - Identify the resources and determine how they are related to each other
    - Find a way to **represent** these resources in our system

XML

```
<Person>
  <ID>1</ID>
  <Name>Ricardo</Name>
  <Email>ricardo.queiros@gmail.com</Email>
  <Country>Portugal</Country>
</Person>
```

JSON

```
{
  "ID": "1",
  "Name": "Ricardo",
  "Email": "Ricardo.queiros@gmail.com",
  "Country": "Portugal"
}
```

  - Both client and server should be able to comprehend this format of representation

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Representation of resources
  - Both JSON and XML
    - be used to receive data from a web server
    - are "self describing" (human readable)
    - are hierarchical (values within values)
    - can be parsed and used by lots of programming languages
    - can be fetched with an XMLHttpRequest
  - Differences:
    - Length: JSON is shorter (usually)
    - Legibility: JSON is quicker to read and write
    - Parsing
      - XML has to be parsed with an XML parser and JSON can be parsed by a standard JavaScript function
      - XML is much more difficult to parse than JSON
      - JSON is parsed into a ready-to-use JavaScript object

<XML>
{JSON}

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

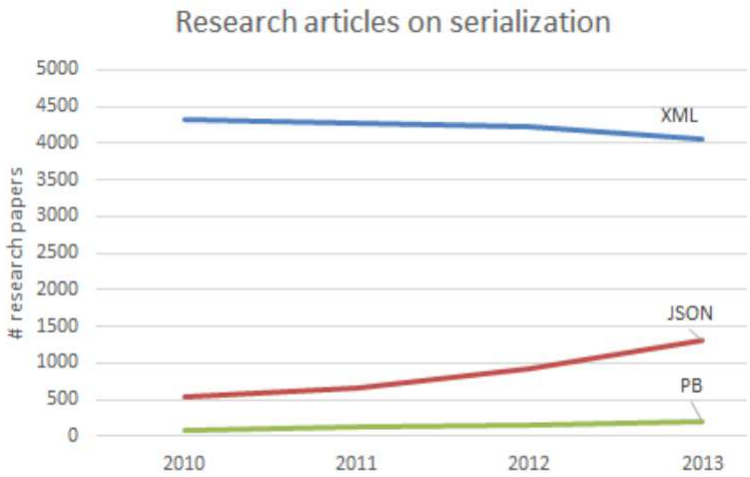
WEB SERVICES

- REST | Representation of resources
  - Serialization formats

Name	Date	Creator	Based on	Schema/IDL	Human-Readable
CSV	1967	Yakov Shafranovich	n/a	partial	yes
XML	1998	W3C	SGML	yes	yes
XML-RPC	1998	Dave Winer	XML/SOAP	no	yes
JSON	2001	Douglas Crockford	JavaScript	partial	yes
YAML	2001	Clark Evans	C/Perl/XML	partial	yes
Candle	2005	Henry Luo	XML/JSON	yes	yes
OpenDDL	2013	Eric Lengyel	C/PHP	no	yes

<XML>
{JSON}

  - Stats:



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

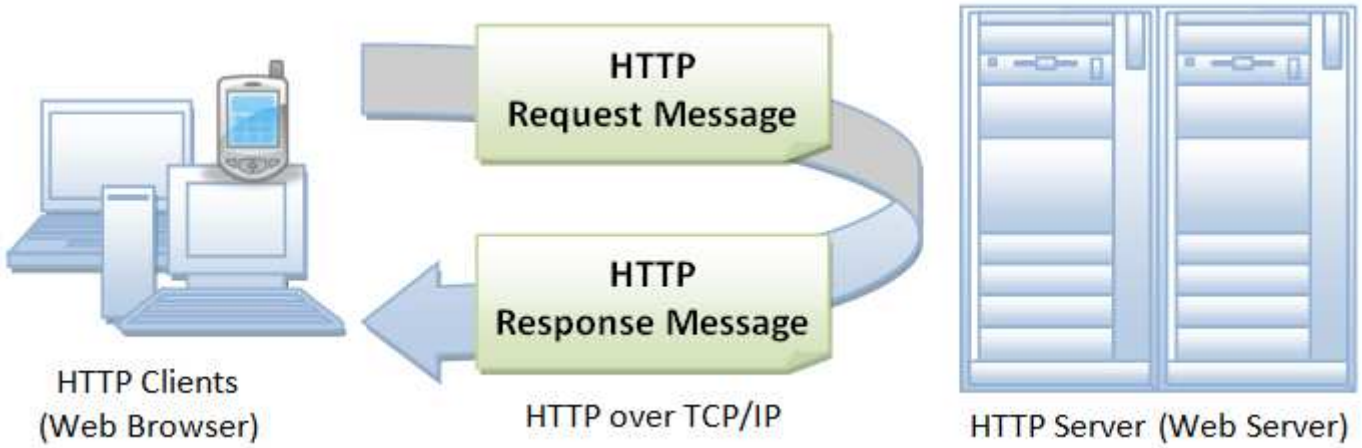
WEB SERVICES

- REST | Messaging
  - The client and service talk to each other via **messages**
  - Clients send a request to the server, and the server replies with a response
  - Apart from the actual data, these messages contain metadata about the message
  - It is important to have some background about the HTTP 1.1 request and response formats for designing REST Web services

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Messaging
  - HTTP (Hypertext Transfer Protocol) is perhaps the most popular application protocol used in the Internet (or The WEB)

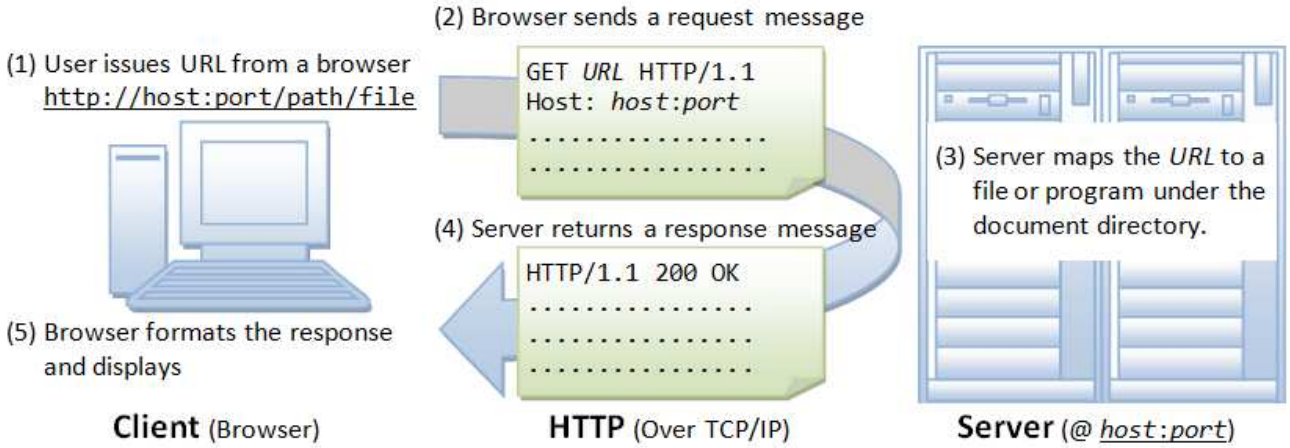


  - In order for proper communication to take place between the client and the server, these applications must agree on a specific application-level protocol such as HTTP, FTP, SMTP, POP, and etc.

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Messaging
  - HTTP in more detail



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Messaging
  - Example of a HTTP request:

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

Request Line
Request Headers
Request Message Header
A blank line separates header & body
Request Message Body

  - Example of a HTTP response:

```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

Status Line
Response Headers
Response Message Header
A blank line separates header & body
Response Message Body

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Messaging
  - HTTP 1.1 provides a set of request methods, called verbs:
    - GET: get a resource from the server
    - POST: Ask the server to store/update the data
    - PUT: Ask the server to store/update the data
    - DELETE: Ask the server to delete the data

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Messaging
  - HTTP 1.1 provides a set of request methods, called verbs:
    - GET: get a resource from the server
    - POST: Ask the server to store/update the data
    - PUT: Ask the server to store/update the data
    - DELETE: Ask the server to delete the data

	http://server/persons	http://server/persons/12
GET	List all the persons	Retrieve a single person
POST	Create a new person	Error
PUT	Replace the entire collection with a whole new list of persons	Replace or create a single person
DELETE	Delete all persons	Delete a single person

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Messaging
  - HTTP 1.1 provides a set of request methods, called verbs:
    - GET: get a resource from the server
    - POST: Ask the server to store/update the data
    - PUT: Ask the server to store/update the data
    - DELETE: Ask the server to delete the data

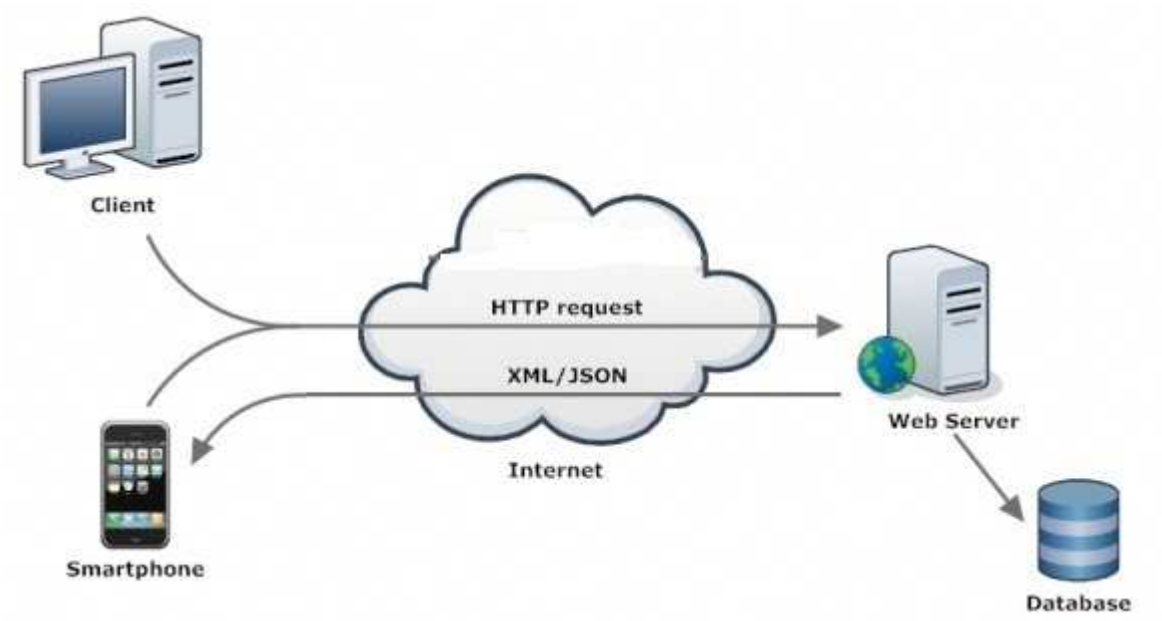
```
POST http://server/persons/
Host: server
Content-Type: text/xml; charset=utf-8
Content-Length: 123

<?xml version="1.0" encoding="utf-8"?>
<Person>
  <ID>1</ID>
  <Name>Ricardo</Name>
  <Email>ricardo.queiros@gmail.com</Email>
  <Country>Portugal</Country>
</Person>
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- A service designed to support machine-to-machine interaction over a network



- It uses HTTP for machine-to-machine communication transferring machine readable file formats such as XML and JSON

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

WEB SERVICES

- REST | Uniform interface
  - REST API example

teams : Teams

Show/Hide | List Operations | Expand Operations

GET	/teams/	List of teams
POST	/teams/	Create a new team
DELETE	/teams/{team_id}	Delete a team by ID
GET	/teams/{team_id}	Get team details by ID

INDEX

- ME
- MOTIVATION
- WEB SERVICES
  - HTTP
  - REST
  - XML/JSON
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

INDEX

- ME
- MOTIVATION
- WEB SERVICES
  - HTTP
  - REST
  - XML/JSON
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- ipify
  - Allows to get your public IP address programmatically
  - You can get your public IP directly in
    - Plain text
    - JSON format
    - JSONP format

API URI	Response Type	Sample Output (IPv4)	Sample Output (IPv6)
https://api.ipify.org	text	98.207.254.136	?
https://api.ipify.org?format=json	json	{"ip":"98.207.254.136"}	?
https://api.ipify.org?format=jsonp	jsonp	callback({"ip":"98.207.254.136"});	?
https://api.ipify.org?format=jsonp&callback=getip	jsonp	getip({"ip":"98.207.254.136"});	?

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- ipify

Bash

```bash
ip=$(curl -s https://api.ipify.org)
echo "My public IP address is: $ip"
```

PHP

```php
<?php
    $ip = file_get_contents('https://api.ipify.org');
    echo "My public IP address is: " . $ip;
?>
```

JavaScript

```javascript
<script type="application/javascript">
  function getIP(json) {
    document.write("My public IP address is: ", json.ip);
  }
</script>

<script type="application/javascript"
src="https://api.ipify.org?format=jsonp&callback=getIP"></script>
```

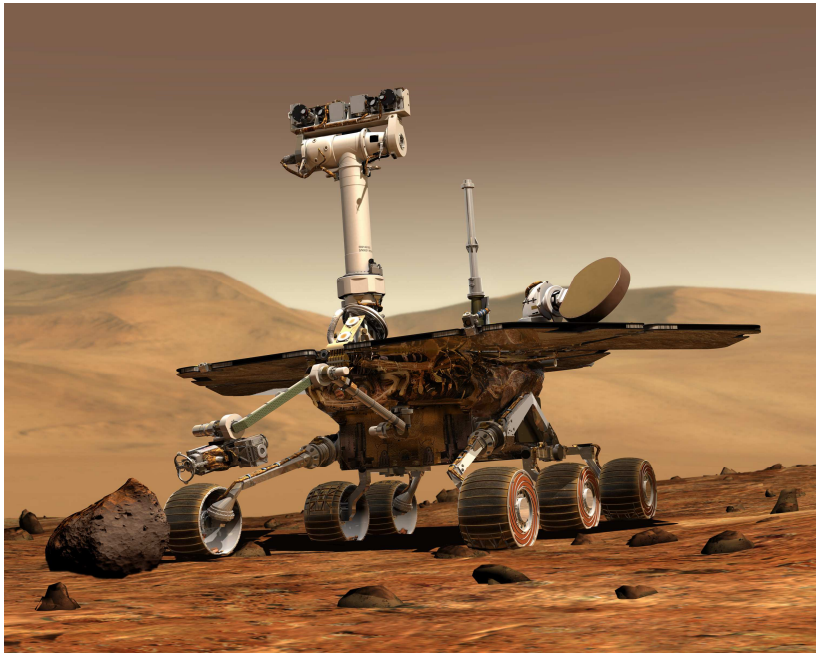
INDEX

- ME
- MOTIVATION
- WEB SERVICES
  - HTTP
  - REST
  - XML/JSON
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- NASA Open APIs
  - Goal: make NASA data, including imagery, accessible to application developers
  - Example: get a photo from Mars Exploration Rover at a specific date



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- NASA Open APIs
  - Goal: make NASA data, including imagery, accessible to application developers
  - API request

```
https://api.nasa.gov/mars-photos/api/v1/rovers/curiosity/photos?
earth_date=[DATE]&api_key=[API_KEY]
```

  - API response


```
{"photos":[{"id":102685,"sol":1004,"camera":{"id":20,"name":"FHAZ","rover_id":5,"full_name":"Front Hazard Avoidance Camera"},"img_src":"http://mars.jpl.nasa.gov/msl-raw-
images/proj/msl/redops/ods/surface/sol/01004/opgs/edr/fcam/FLB_486615455EDR_F0481570FHAZ00323M_.JPG","earth_date":"2015-06-03","rover":{"id":5,"name":"Curiosity","landing_date":"2012-08-
06","launch_date":"2011-11-26","status":"active","max_sol":1717,"max_date":"2017-06-05","total_photos":315921,"cameras":[{"name":"FHAZ","full_name":"Front Hazard Avoidance Camera"},
{"name":"NAVCAM","full_name":"Navigation Camera"},{"name":"MAST","full_name":"Mast Camera"},{"name":"CHEMCAM","full_name":"Chemistry and Camera Complex"},{"name":"MAHLI","full_name":"Mars Hand Lens
Imager"},{"name":"MARDI","full_name":"Mars Descent Imager"},{"name":"RHAZ","full_name":"Rear Hazard Avoidance Camera"}]}},{"id":102686,"sol":1004,"camera":
{"id":20,"name":"FHAZ","rover_id":5,"full_name":"Front Hazard Avoidance Camera"},"img_src":"http://mars.jpl.nasa.gov/msl-raw-
images/proj/msl/redops/ods/surface/sol/01004/opgs/edr/fcam/FRB_486615455EDR_F0481570FHAZ00323M_.JPG","earth_date":"2015-06-03","rover":{"id":5,"name":"Curiosity","landing_date":"2012-08-
06","launch_date":"2011-11-26","status":"active","max_sol":1717,"max_date":"2017-06-05","total_photos":315921,"cameras":[{"name":"FHAZ","full_name":"Front Hazard Avoidance Camera"},
{"name":"NAVCAM","full_name":"Navigation Camera"},{"name":"MAST","full_name":"Mast Camera"},{"name":"CHEMCAM","full_name":"Chemistry and Camera Complex"},{"name":"MAHLI","full_name":"Mars Hand Lens
Imager"},{"name":"MARDI","full_name":"Mars Descent Imager"},{"name":"RHAZ","full_name":"Rear Hazard Avoidance Camera"}]}},{"id":102842,"sol":1004,"camera":
{"id":21,"name":"RHAZ","rover_id":5,"full_name":"Rear Hazard Avoidance Camera"},"img_src":"http://mars.jpl.nasa.gov/msl-raw-
images/proj/msl/redops/ods/surface/sol/01004/opgs/edr/rcam/RLB_486615482EDR_F0481570RHAZ00323M_.JPG","earth_date":"2015-06-03","rover":{"id":5,"name":"Curiosity","landing_date":"2012-08-
06","launch_date":"2011-11-26","status":"active","max_sol":1717,"max_date":"2017-06-05","total_photos":315921,"cameras":[{"name":"FHAZ","full_name":"Front Hazard Avoidance Camera"},
{"name":"NAVCAM","full_name":"Navigation Camera"},{"name":"MAST","full_name":"Mast Camera"},{"name":"CHEMCAM","full_name":"Chemistry and Camera Complex"},{"name":"MAHLI","full_name":"Mars Hand Lens
Imager"},{"name":"MARDI","full_name":"Mars Descent Imager"},{"name":"RHAZ","full_name":"Rear Hazard Avoidance Camera"}]}},{"id":102843,"sol":1004,"camera":
{"id":21,"name":"RHAZ","rover_id":5,"full_name":"Rear Hazard Avoidance Camera"},"img_src":"http://mars.jpl.nasa.gov/msl-raw-
images/proj/msl/redops/ods/surface/sol/01004/opgs/edr/rcam/RRB_486615482EDR_F0481570RHAZ00323M_.JPG","earth_date":"2015-06-03","rover":{"id":5,"name":"Curiosity","landing_date":"2012-08-
06","launch_date":"2011-11-26","status":"active","max_sol":1717,"max_date":"2017-06-05","total_photos":315921,"cameras":[{"name":"FHAZ","full_name":"Front Hazard Avoidance Camera"},
{"name":"NAVCAM","full_name":"Navigation Camera"},{"name":"MAST","full_name":"Mast Camera"},{"name":"CHEMCAM","full_name":"Chemistry and Camera Complex"},{"name":"MAHLI","full_name":"Mars Hand Lens
Imager"},{"name":"MARDI","full_name":"Mars Descent Imager"},{"name":"RHAZ","full_name":"Rear Hazard Avoidance Camera"}]}}]}
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- NASA Open APIs
  - Student exercise

2015-06-03 Get Mars photo

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- NASA Open APIs
  - Student exercise

2015-06-03 Get Mars photo

```
$(function () {
 function getNasa() {
  var datePhoto = $("#txt_data").val();
  var stringRequest = "https://...?earth_date=" + datePhoto + "&api_key=…";
  var url;
  $.get(stringRequest, function (data, status) {
    if (status == "success") {
      url = data.photos[0].img_src;
      $("#img_mars").attr("src", url);
    } else {
     $("body").append("<p style='color:red'>Error: " + status.message + "</p>");
    }
  });
 }

 $("#btn_invocar").click(function () {
  getNasa();
 });
});
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- NASA Open APIs
  - Student exercise

INDEX

- ME
- MOTIVATION
- WEB SERVICES
  - HTTP
  - REST
  - XML/JSON
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

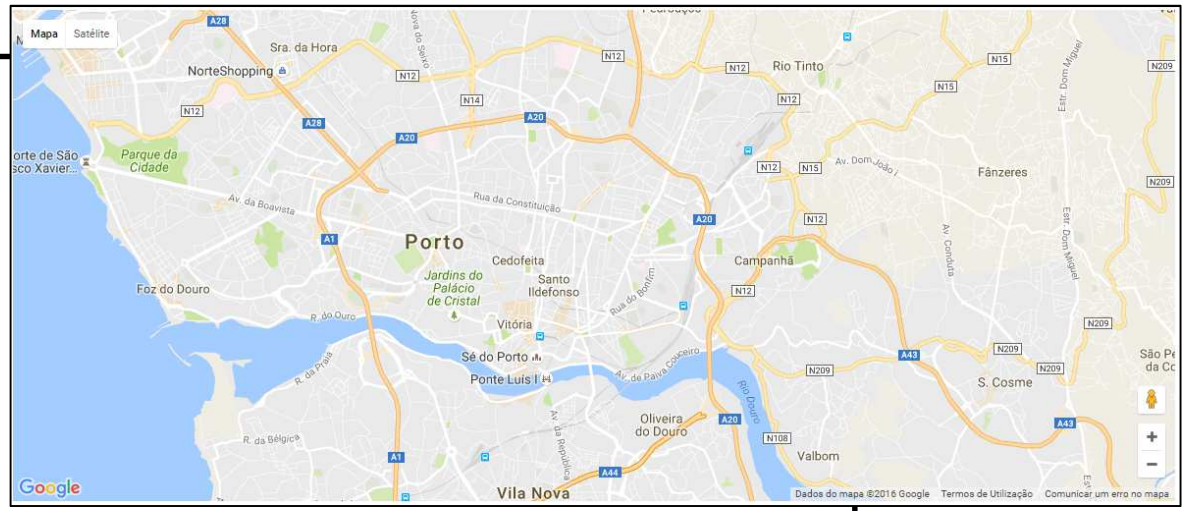
- Google Maps API
  - Created by Google in June 2005
  - Goal: allow developers integrate/manipulate Google maps on websites
  - Currently in version 3.26 (november, 2016)
  - Available for free
  - Free service limited (25 000 requests/day)
  - Avaliable for Android, iOS, web browsers and via Web services (HTTP)

  - More than 2 millions of websites use the Google Maps API

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Create a basic map



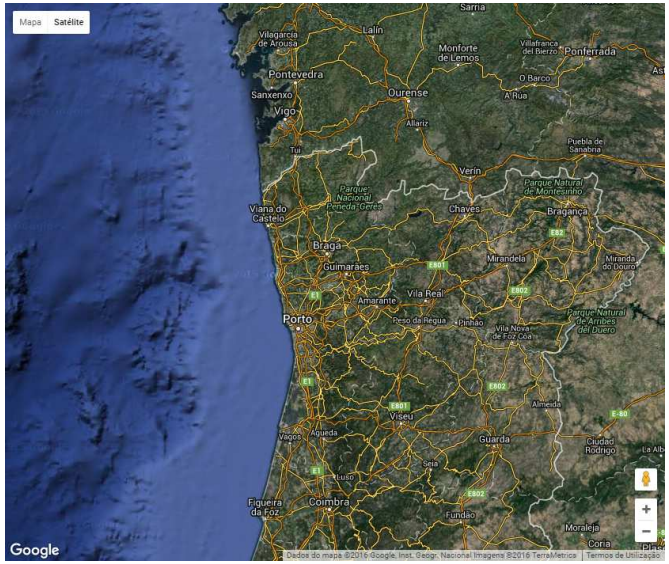

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #map { height: 500px; }
    </style>
  </head>
  <body>
    <div id="map"></div>
    <script>
      var map;
      function initMap() {
        map = new google.maps.Map(document.getElementById('map'), {
          center: {lat: 41.156111, lng: -8.601111}, zoom: 13
        });
      }
    </script>
    <script src=https://maps.googleapis.com/maps/api/js?key=XXX&callback=initMap async></script>
  </body>
</html>
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Change the map type

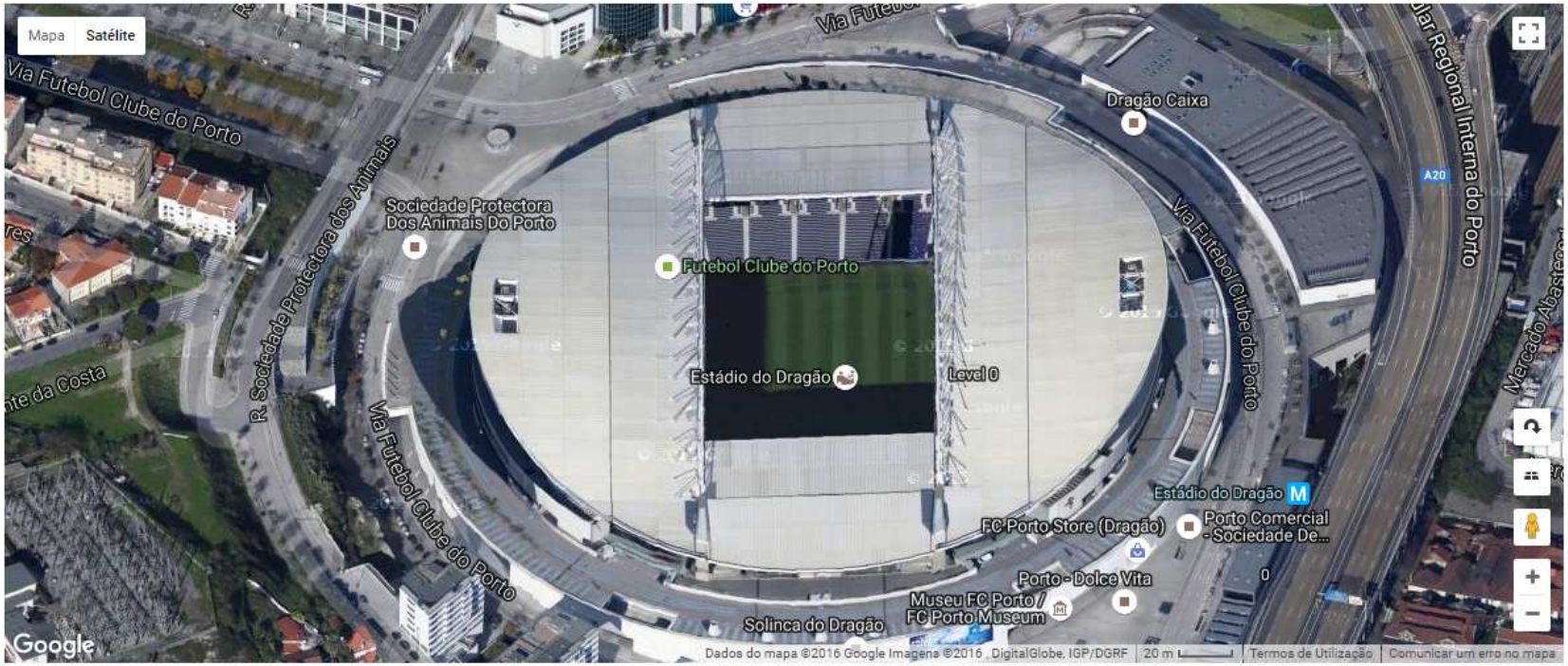
```
var myLatlng = new google.maps.LatLng(-34.397, 150.644);
var mapOptions = {
 zoom: 8,
 center: myLatlng,
 mapTypeId: google.maps.MapTypeId.HYBRID
};
var map = new google.maps.Map(document.getElementById("map"),
                    mapOptions);
```



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Controls and zoom



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

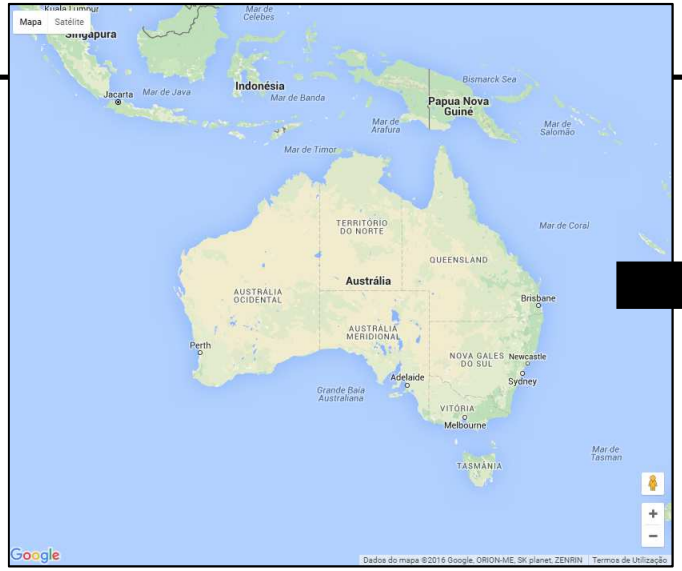
EXAMPLES

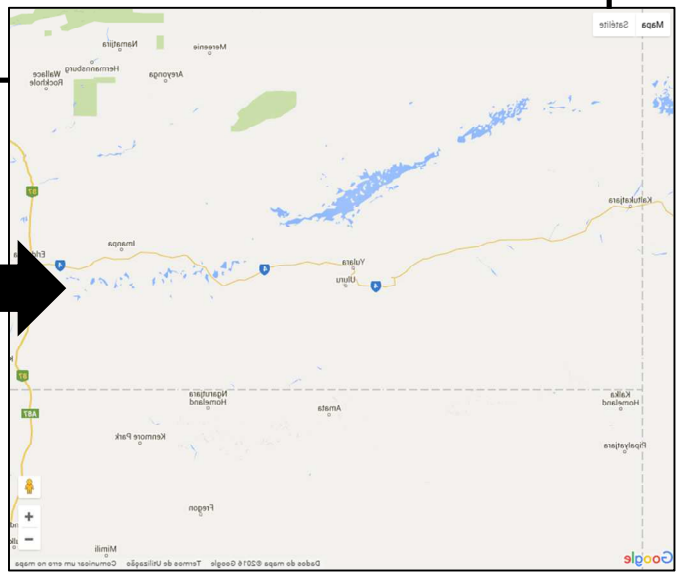
- Google Maps API
  - Events

```
function initMap() {
  var myLatlng = {lat: -25.363, lng: 131.044};

  var map = new google.maps.Map(document.getElementById('map'), { zoom: 4, center: myLatlng });

  map.addListener('click', function() {
    map.setZoom(8);
  });
}
```



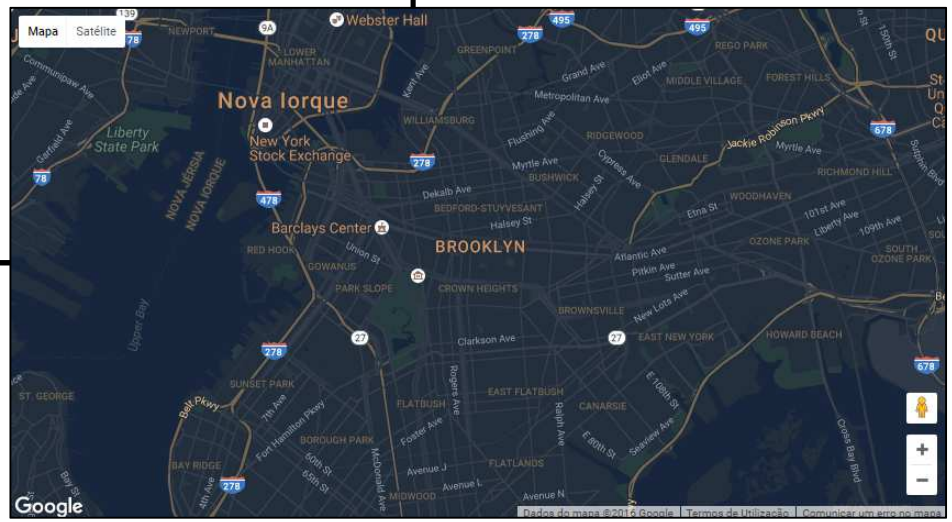


ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Styling
    - Style creator: https://mapstyle.withgoogle.com

```
var map = new google.maps.Map(document.getElementById('map'), {
  zoom: 4,
  center: myLatlng,
  styles: [
      {elementType: 'geometry', stylers: [{color: '#242f3e'}]},
      {elementType: 'labels.text.stroke', stylers: [{color: '#242f3e'}]},
      {
        featureType: 'road',
        elementType: 'geometry',
        stylers: [{color: '#38414e'}]
      }
  });
}
```

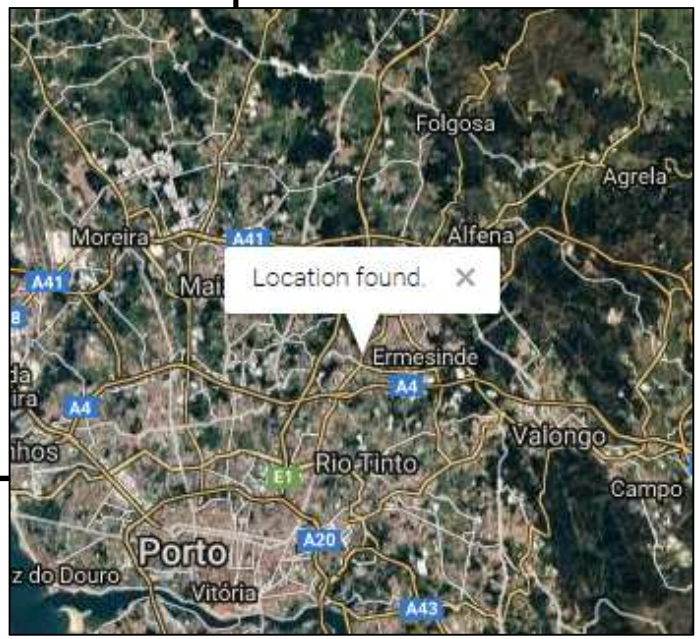


ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Location detection

```
// Geolocation
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(function(position) {
    var pos = { lat: position.coords.latitude, lng: position.coords.longitude };
    infoWindow.setPosition(pos);
    infoWindow.setContent('Location found.');
    map.setCenter(pos);
  }, function() { handleLocationError(true, infoWindow, map.getCenter()); });
} else {
  // Browser don't support Geolocation
  handleLocationError(false, infoWindow, map.getCenter());
}
}
```



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Markers

```
function initMap() {
 var map = new google.maps.Map(document.getElementById('map'), {
   zoom: 13,
   center: {lat: 59.325, lng: 18.070}
 });

 marker = new google.maps.Marker({
   map: map,
   animation: google.maps.Animation.BOUNCE,
   position: {lat: 59.327, lng: 18.067}
 });
}
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Info Windows

```
function initMap() {
 var esmad = {lat: 41.366949, lng: -8.738722};
 var map = new google.maps.Map(document.getElementById('map'), {
   zoom: 17,
   center: esmad
 });
 var contentString = '<div id="content"><div id="siteNotice"></div>'+
     '<h1 id="firstHeading" class="firstHeading">ESMAD</h1>'+
     '<div id="bodyContent">'<p>A <b>ESMAD</b> é uma das escolas do P.PORTO! </p>' +
     '<p><img src="https://www.esmad.ipp.pt/logo-ipp.png"/></p></div></div>';

var infowindow = new google.maps.InfoWindow({ content: contentString });
 var marker = new google.maps.Marker({
   position: esmad,
   map: map
 });
 marker.addListener('click', function() {
   infowindow.open(map, marker);
 });
}
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Info Windows

```
function initMap() {
 var esmad = {lat: 41.366949, lng: -8.738722};
 var map = new google.maps.Map(document.getElementById('map'), {
   zoom: 17,
   center: esmad
 });
 var contentString = '<div id="content"><div id="siteNotice"><
     '<h1 id="firstHeading" class="firstHeading">ESMAD</h1>
     '<div id="bodyContent">'<p>A <b>ESMAD</b> é uma das
     '<p><img src="https://www.esmad.ipp.pt/logo-ipp.png"/><

var infowindow = new google.maps.InfoWindow({ content: co
 var marker = new google.maps.Marker({
   position: esmad,
   map: map
 });
 marker.addListener('click', function() {
   infowindow.open(map, marker);
 });
}
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Shapes

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Layers

```
var mapOptions = {
 zoom: 13,
 center: new google.maps.LatLng(51.5,-0.11)
}

var map = new google.maps.Map(document.getElementById("map"),
                    mapOptions);

var transitLayer = new google.maps.TransitLayer();
transitLayer.setMap(map);
```

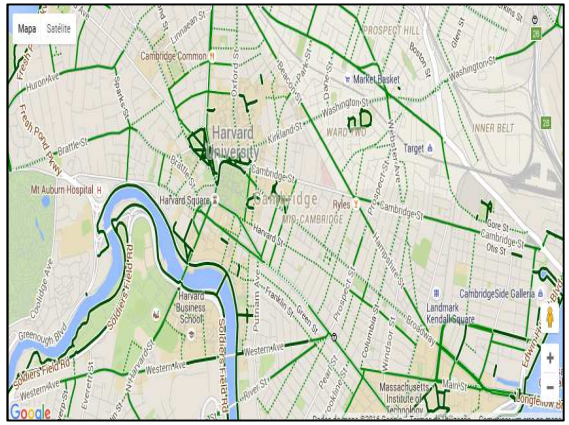
ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

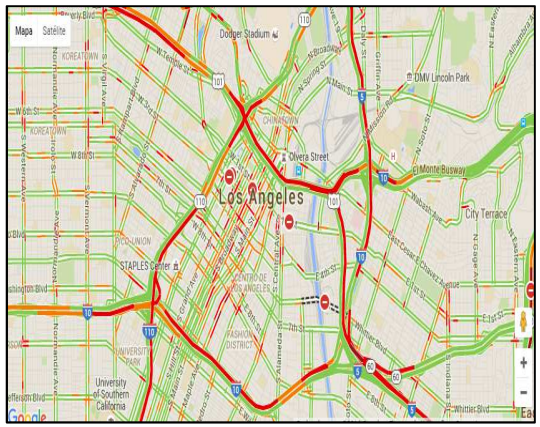
- Google Maps API
  - Layers

Transit layer

Bicycling Layer



Traffic layer



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Fusion Tables
    - Render data contained in Google Fusion Tables
    - Fusion Tables are relational databases
    - Link: https://research.google.com/tables?corpus=fusion
    - There are tables with relevant information:
      - Species of butterflies in the USA
      - Countries and their numbers

```
var layer = new google.maps.FusionTablesLayer({
  query: {
    select: ‘geometry’,
    from: '12e2VhiXyMzHWDl6aponObHH_gvlMDoac9RTrcJ0‘,
    where: “name = ‘Portugal’"
  }
});
layer.setMap(map);
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Services
    - Geocoding
    - Directions & Distances
    - Street View
    - Places

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Services
    - Geocoding
    - Directions & Distances
    - Street View
    - Places

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Services
    - Geocoding
    - Directions & Distances
    - Street View
    - Places

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Services
    - Geocoding
    - Directions & Distances
    - Street View

```
var panorama;
function initialize() {
  panorama = new google.maps.StreetViewPanorama(
      document.getElementById('map'),
        {
          position: {lat: 37.869260, lng: -122.254811},
          pov: {heading: 165, pitch: 0},
          zoom: 1
        });
   }
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Services
    - Geocoding
    - Directions & Distances
    - Street View

```
var panorama;
function initialize() {
  panorama = new google.maps.S
      document.getElementById('m
        {
          position: {lat: 37.869260, l
          pov: {heading: 165, pitch:
          zoom: 1
        });
  }
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Services
    - Geocoding
    - Directions & Distances
    - Street View
    - Places

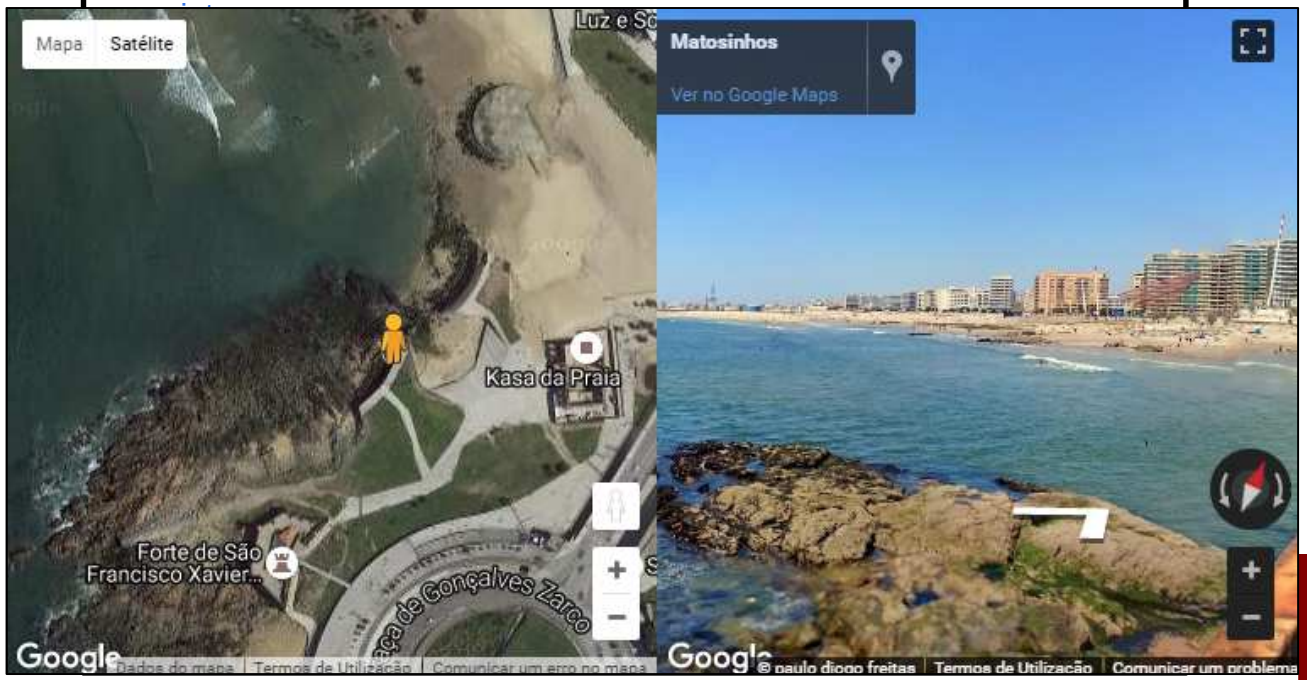
```
<div id="map"></div>
<div id="pano"></div>
<script>
 function initialize() {
   var matosinhos = {lat: 42.345573, lng: -71.098326};
   var map = new google.maps.Map(document.getElementById('map'), {
     center: matosinhos,
     zoom: 14
   });
   var panorama = new google.maps.StreetViewPanorama(
       document.getElementById('pano'), {
         position: matosinhos,
         pov: {
           heading: 34,
           pitch: 10
         }
       });
   map.setStreetView(panorama);
 }
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Services
    - Geocoding
    - Directions & Distances
    - Street View
    - Places

```
<div id="map"></div>
<div id="pano"></div>
```



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Google Maps API
  - Services
    - Geocoding
    - Directions & Distances
    - Street View
    - Places

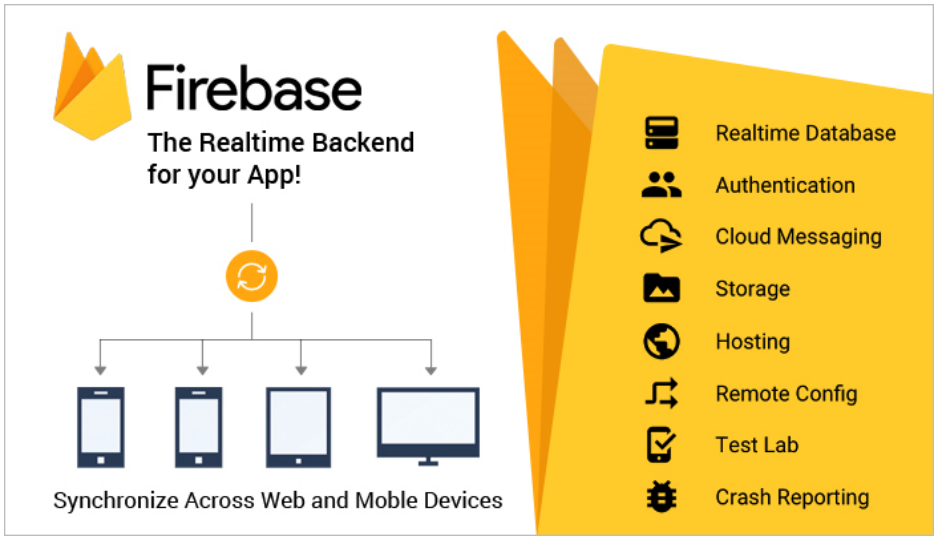
INDEX

- ME
- MOTIVATION
- WEB SERVICES
  - HTTP
  - REST
  - XML/JSON
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

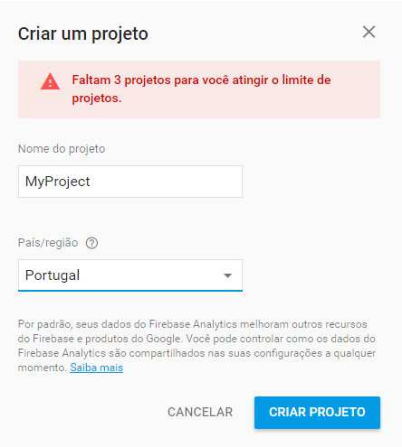
- Firebase
  - Mobile and web application platform with tools and infrastructure designed to help developers build high-quality apps
  - Company founded in 2011 by Andrew Lee and James Tamplin
  - Firebase's initial product was a realtime database
  - Acquired by Google in October 2014
  - URL: https://firebase.google.com

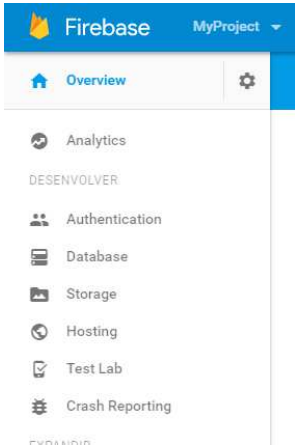


ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

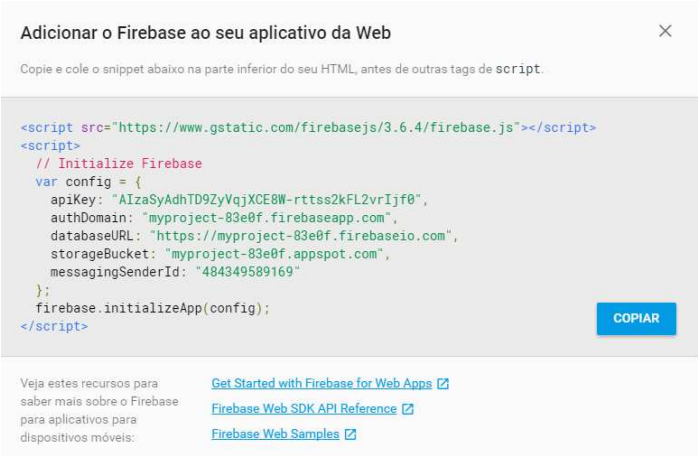
EXAMPLES

- Firebase
  1. Add and configure the Firebase JavaScript client SDK into your app
     - Create a Firebase project in the Firebase console: https://console.firebase.google.com/
     - Click Add Firebase to your web app
     - Copy the initialization code snippet generated





Adicionar o Firebase ao seu aplicativo da Web



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Firebase

  - Include the code snippet in your HTML page

```
<script src="https://www.gstatic.com/firebasejs/3.6.2/firebase.js"></script>
<script>
  // Initialize Firebase
  // TODO: Replace with your project's customized code snippet
  var config = {
    apiKey: "<API_KEY>",
    authDomain: "<PROJECT_ID>.firebaseapp.com",
    databaseURL: "https://<DATABASE_NAME>.firebaseio.com",
    storageBucket: "<BUCKET>.appspot.com",
    messagingSenderId: "<SENDER_ID>",
  };
  firebase.initializeApp(config);
</script>
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

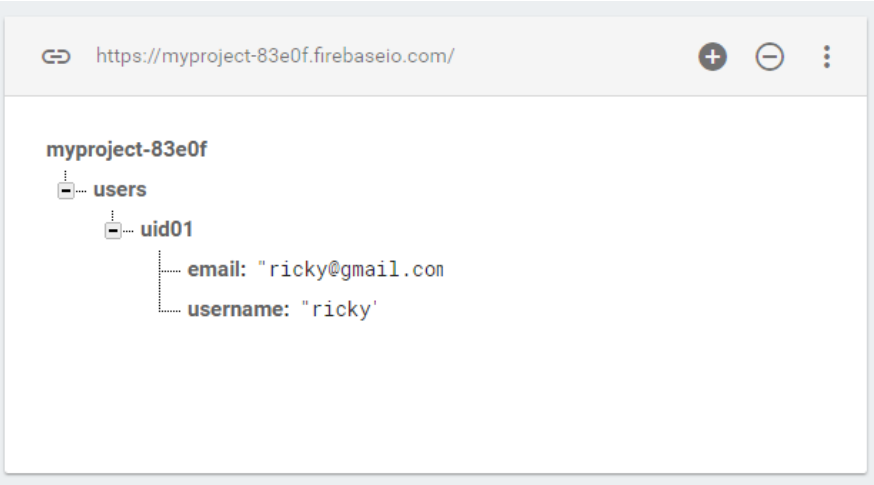
- Firebase

  - Write data

```
// Get a reference to the database service
var database = firebase.database();

// Basic write operation
 $('#postData').click(function() {
  var userId = $('#userId').val();
  var name = $('#name').val();
  var email = $('#email').val();

  database.ref('users/' + userId).set({
    username: name,
    email: email
 });
});
```



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

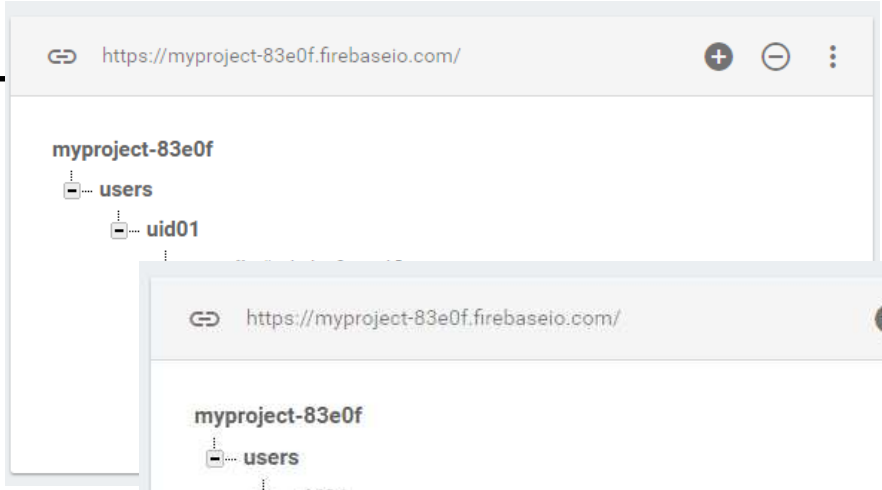
EXAMPLES

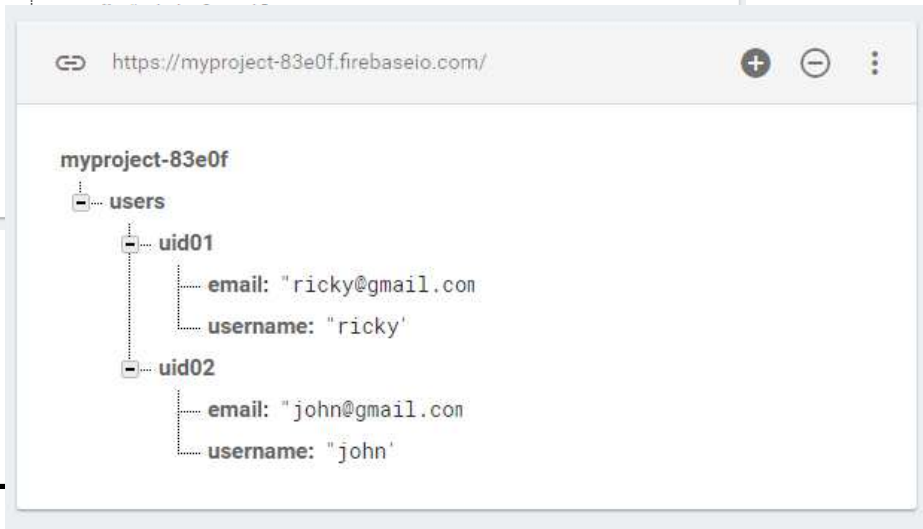
- Firebase
  - Write data

```
// Get a reference to the database service
var database = firebase.database();

// Basic write operation
 $('#postData').click(function() {
  var userId = $('#userId').val();
  var name = $('#name').val();
  var email = $('#email').val();

  database.ref('users/' + userId).set({
    username: name,
    email: email
 });
});
```





ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Firebase

  - Read data

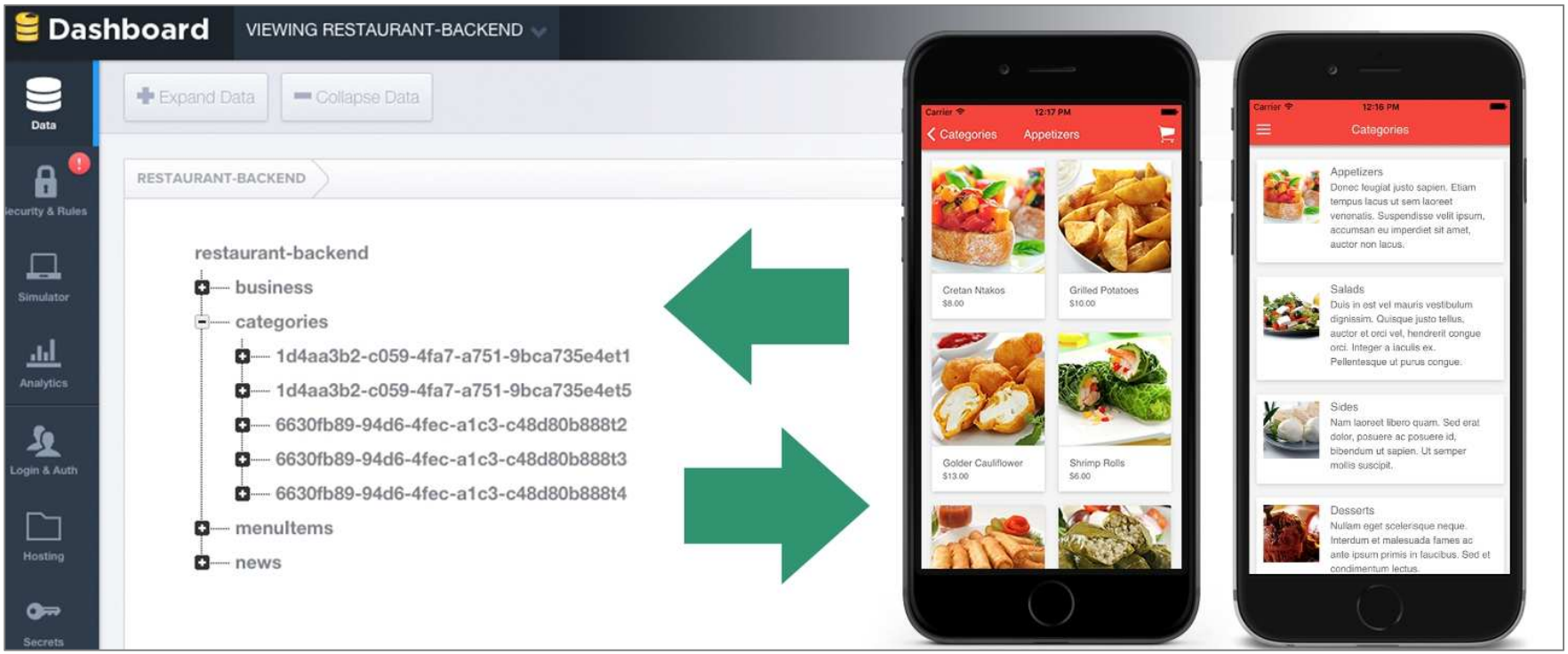
```
// Obtains a data snapshot without listening for changes
return database.ref('/users/' + userId).once('value').then(function(snapshot) {
  var username = snapshot.val().username;
  // ...
});

// Obtains a data snapshot and listens to child changes
return database.ref('/users/' + userId).on('value', function(snapshot) {
  var username = snapshot.val().username;
  // ...
 });
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Firebase



INDEX

- ME
- MOTIVATION
- WEB SERVICES
  - HTTP
  - REST
  - XML/JSON
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Reversi (Othello) is a strategy board game for two players, played on an 8×8 board
  - Othello + Android = Othelloid

  - Users played once but then did not want to play anymore
  - Something was missing to motivate them!

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Games industry is growing exponentially in the worldwide economy
  - Key factors: use of common game concepts such as leaderboards and achievements
  - The massive use of this approach and the impressive growth of players led to the concept of gamification as a service
  - Steps for GPGS integration:
    1. Configuration
    2. Implementation
    3. Test
    4. Publication

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Provides APIs that let you find, retain, and match players for your games on Android, C++, and the Web
  - Can be used by application developers to allow a more competitive and social experience through the use of

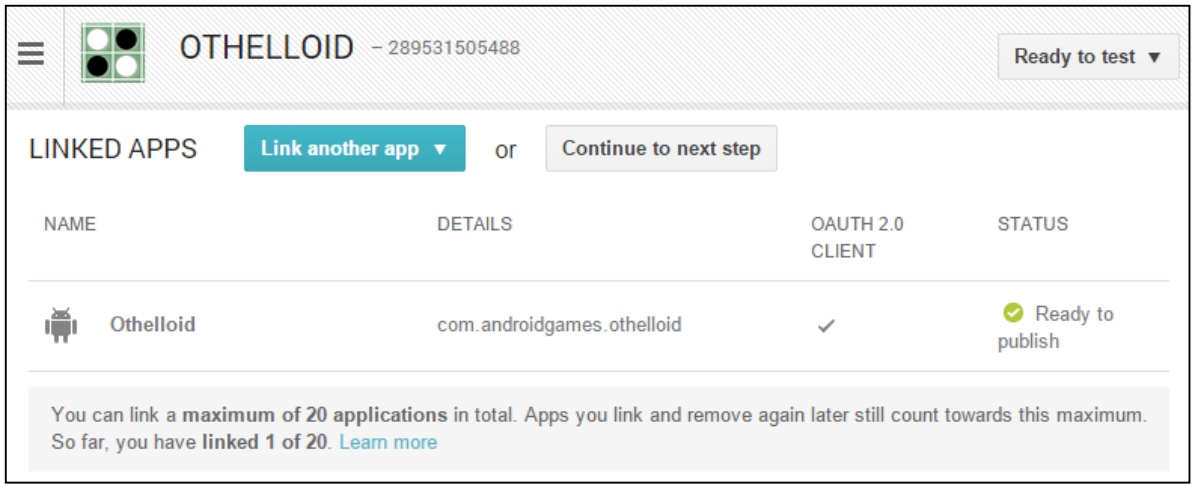
  - Link: https://developers.google.com/games/services/

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Steps for GPGS integration:

1. Configuration
2. Implementation
3. Test
4. Publication



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

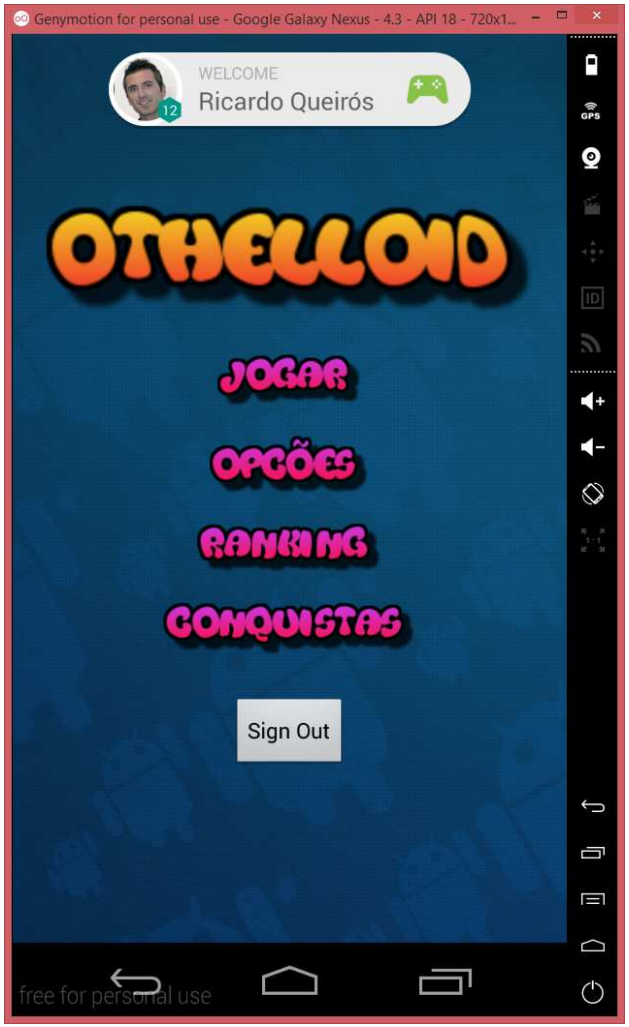
- Play Games Services API
  - Steps for GPGS integration:

1. Configuration
2. Implementation
3. Test
4. Publication

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

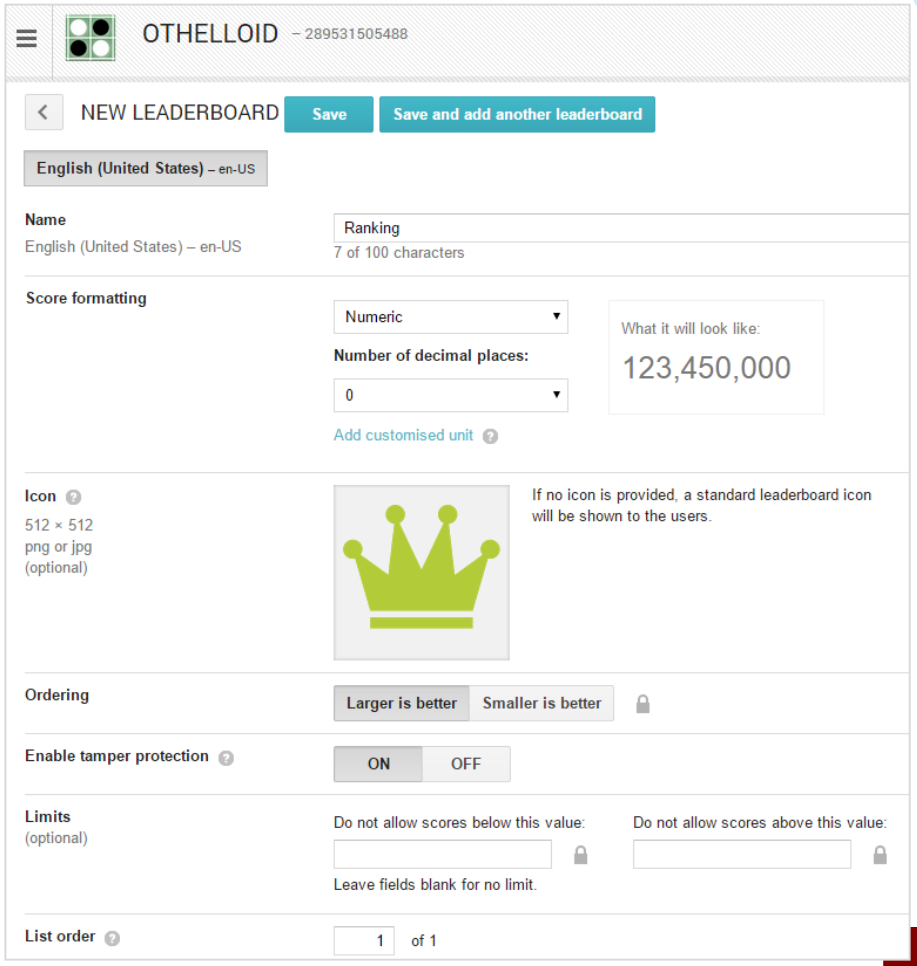
- Play Games Services API
  - Steps for GPGS integration:
  1. Configuration
  2. Implementation
  3. Test
  4. Publication



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

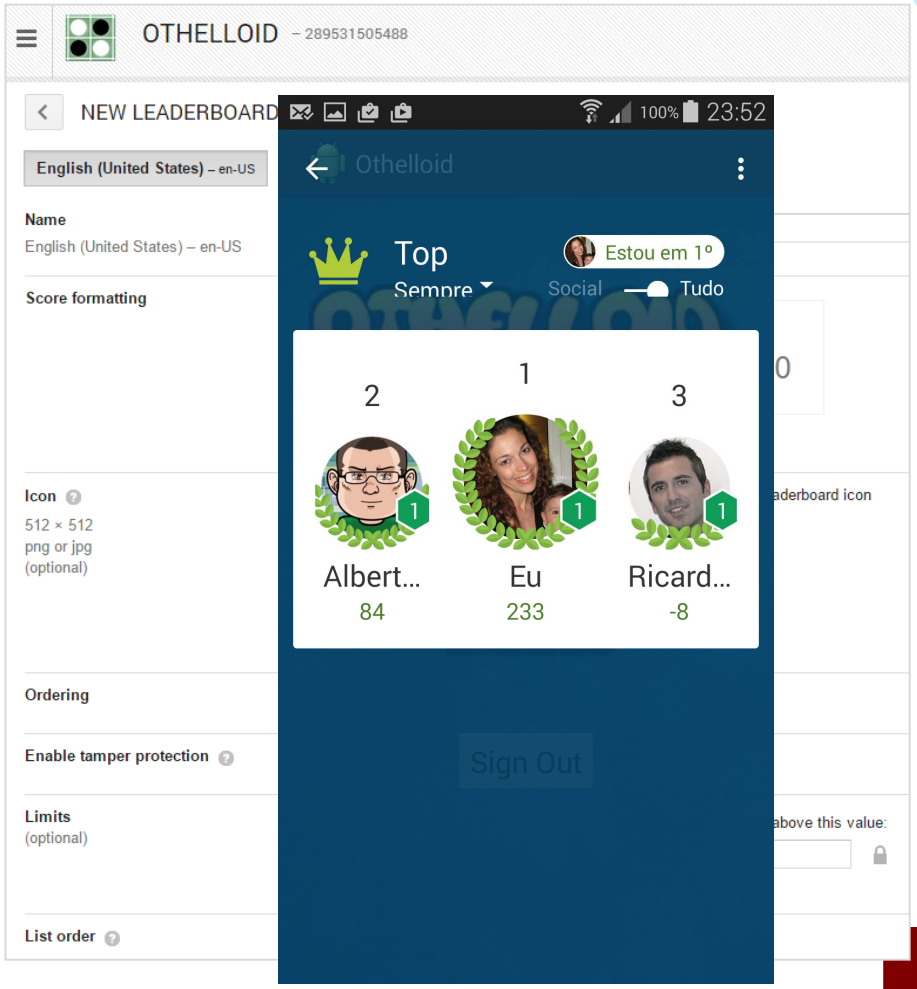
- Play Games Services API
  - Steps for GPGS integration:
  1. Configuration
  2. Implementation
  3. Test
  4. Publication



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

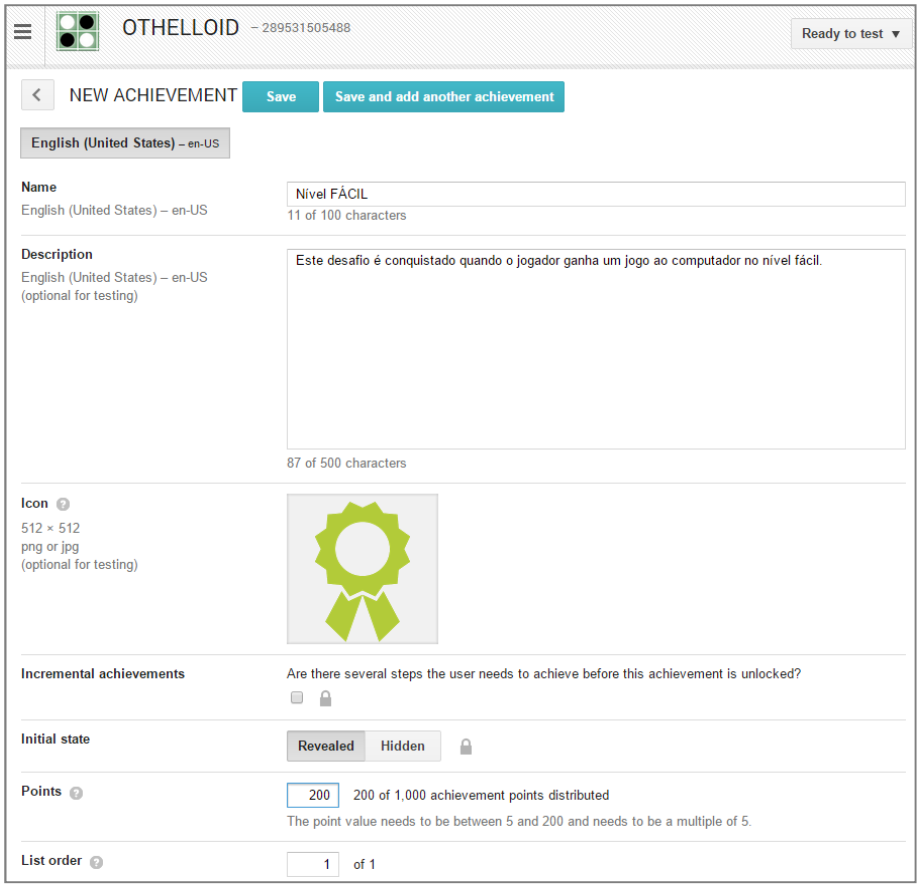
- Play Games Services API
  - Steps for GPGS integration:
    1. Configuration
    2. Implementation
    3. Test
    4. Publication



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

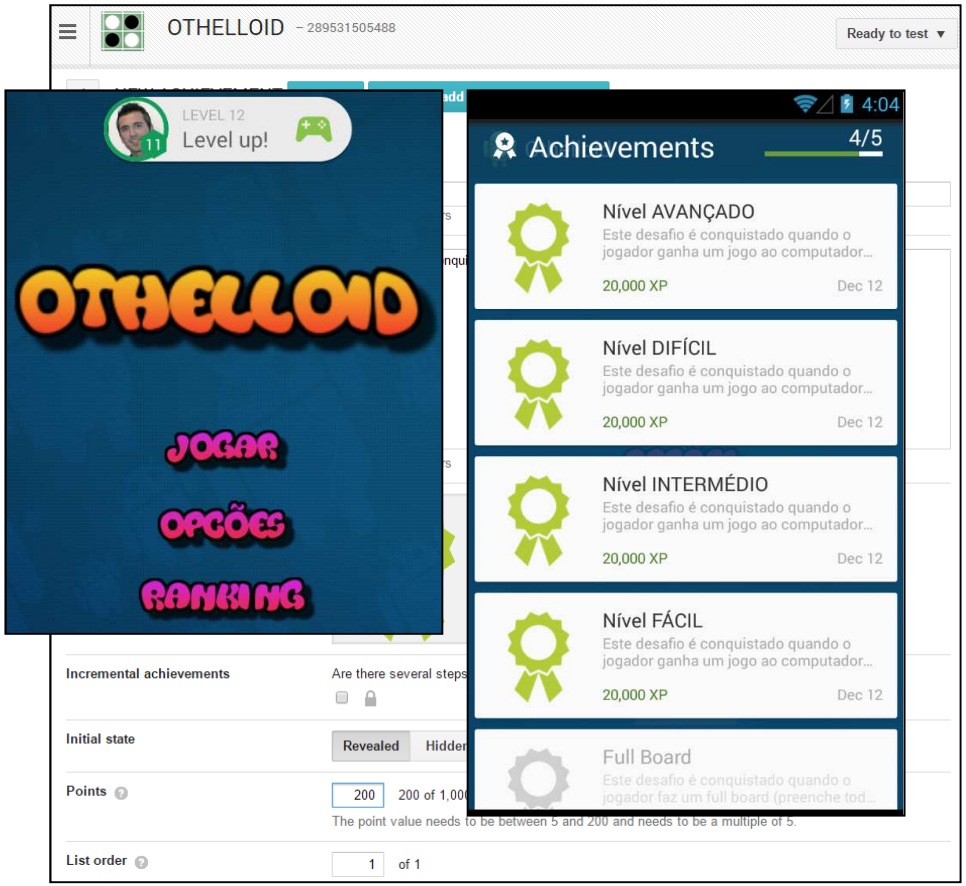
- Play Games Services API
  - Steps for GPGS integration:
  1. Configuration
  2. Implementation
  3. Test
  4. Publication



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

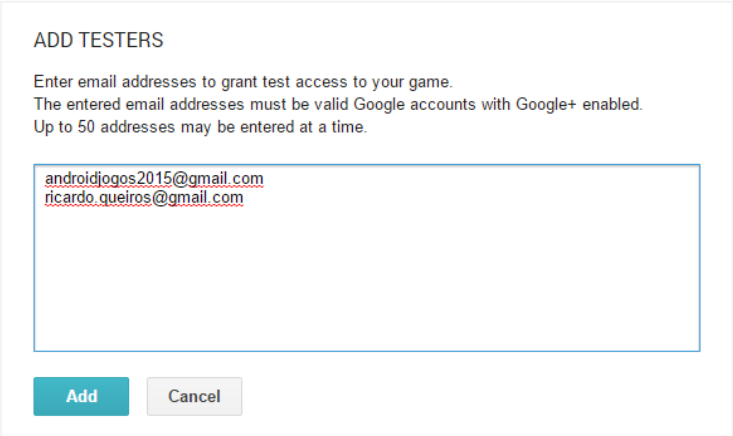
- Play Games Services API
  - Steps for GPGS integration:
  1. Configuration
  2. Implementation
  3. Test
  4. Publication

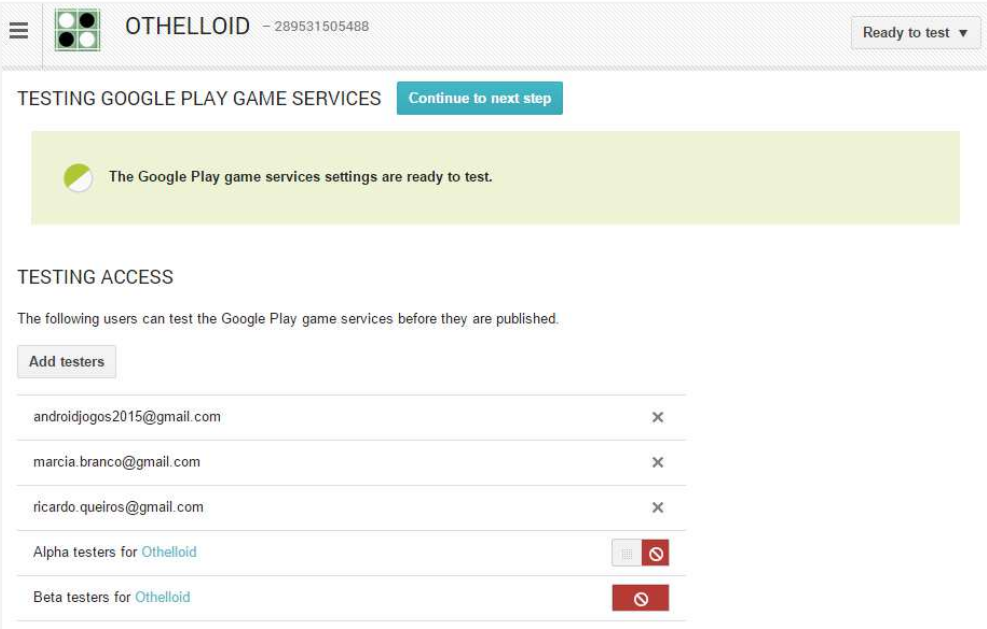


ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Steps for GPGS integration:
  1. Configuration
  2. Implementation
  3. Test
  4. Publication

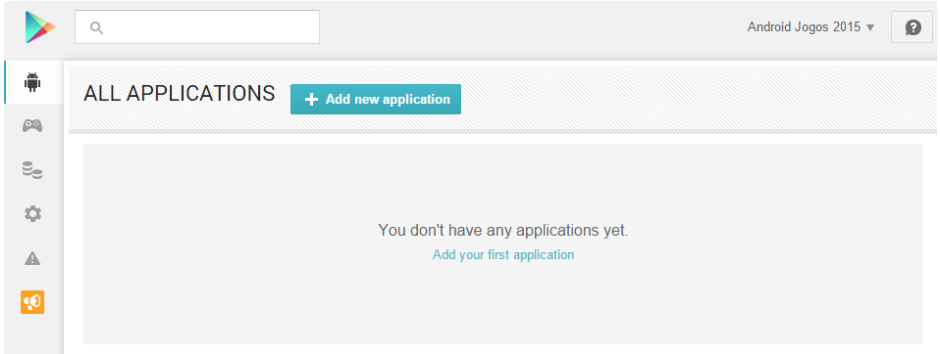




ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Steps for GPGS integration:
  1. Configuration
  2. Implementation
  3. Test
  4. Publication

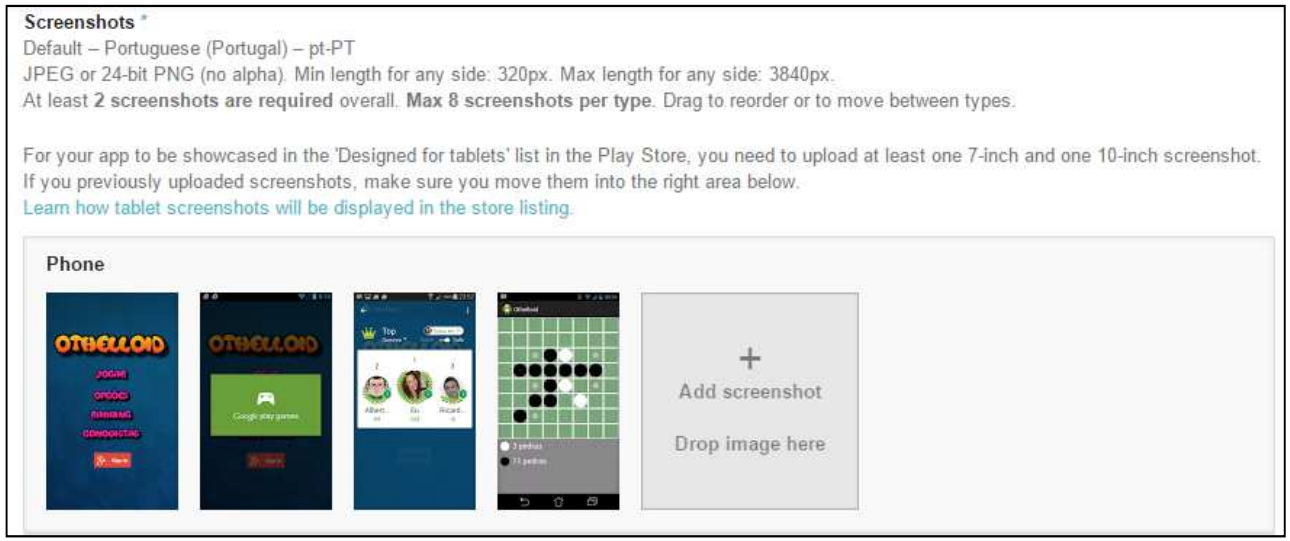


ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Steps for GPGS integration:

1. Configuration
2. Implementation
3. Test
4. **Publication**

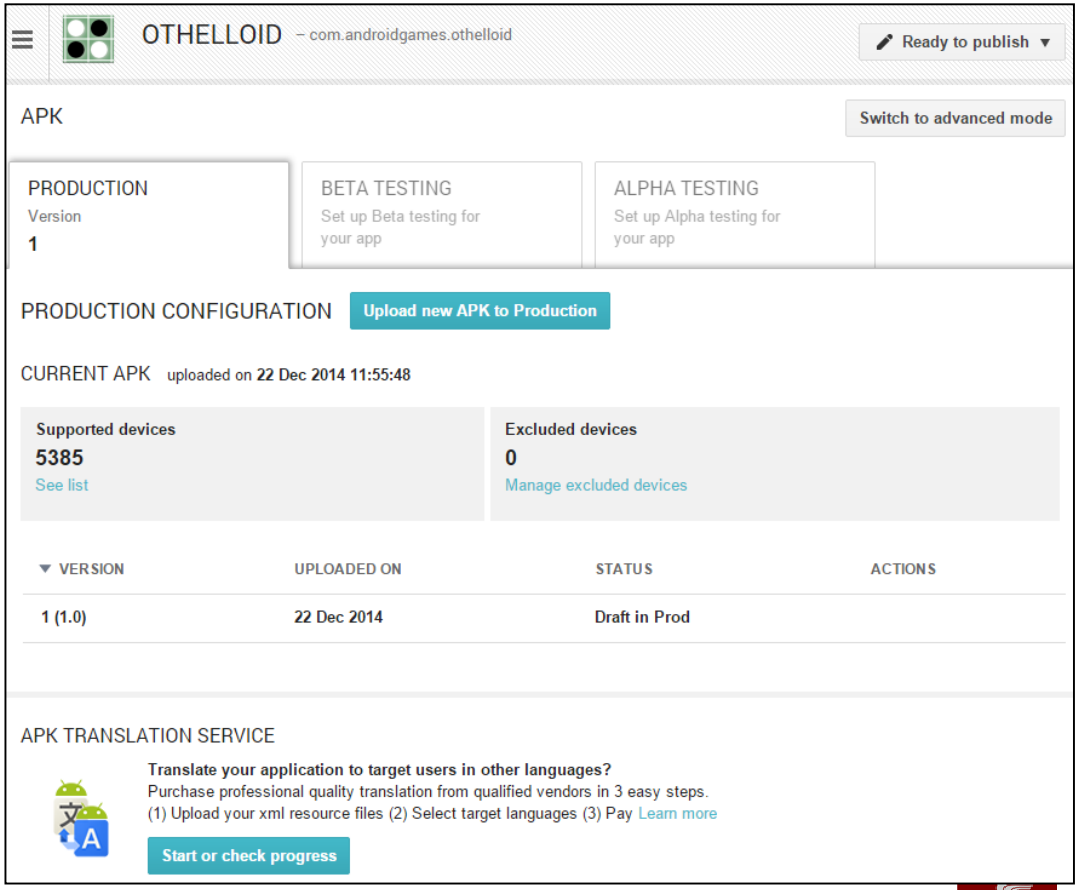


ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Steps for GPGS integration:

1. Configuration
2. Implementation
3. Test
4. **Publication**

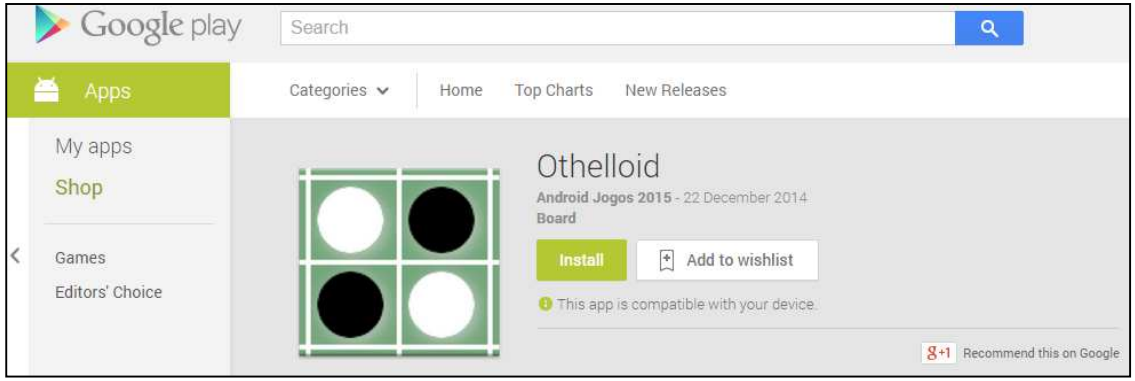


ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Play Games Services API
  - Steps for GPGS integration:

1. Configuration
2. Implementation
3. Test
4. **Publication**



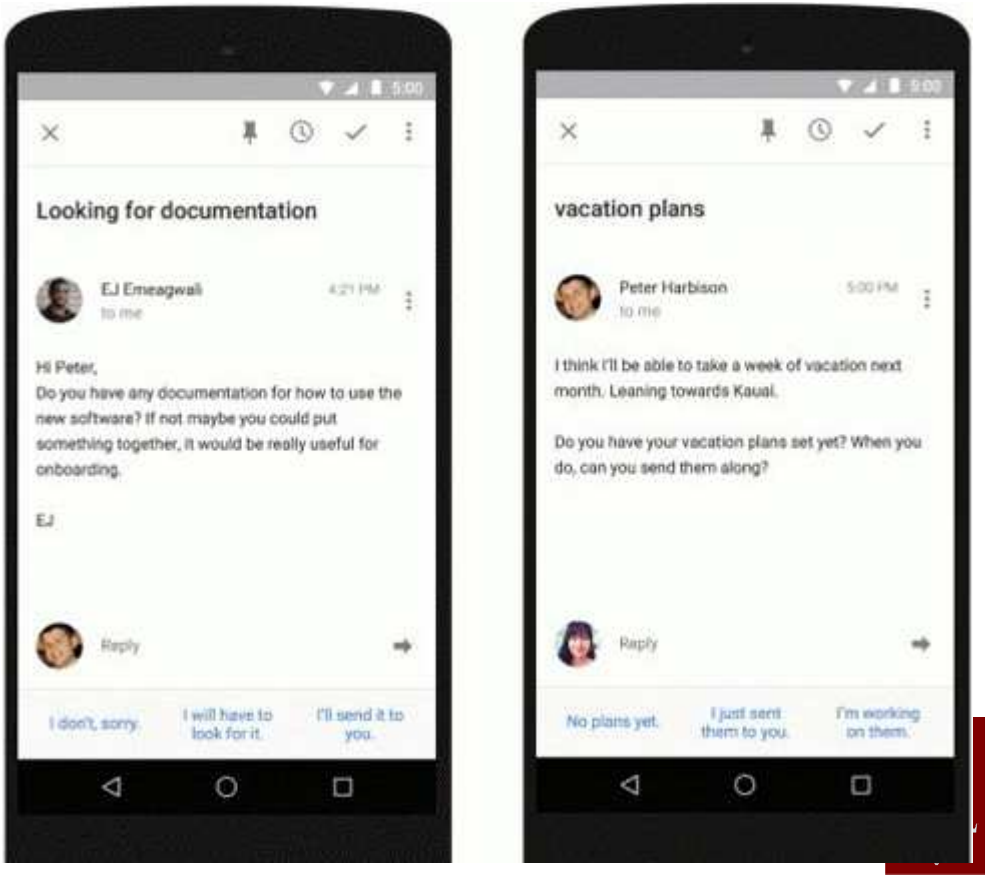
INDEX

- ME
- MOTIVATION
- WEB SERVICES
  - HTTP
  - REST
  - XML/JSON
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Cloud Machine Learning Services
  - Provides **modern machine learning services**
  - Examples:
    - Job Search and Discovery
    - Video Analysis
    - Image Analysis
    - Speech Recognition
    - Text Analysis
    - Dynamic Translation
  - Major Google applications
    - Photos (image search)
    - Google app (voice search)
    - Translate
    - Inbox (Smart Reply)



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Cloud Machine Learning Services
  - Student exercise
    - Take a photo with a smartphone
    - Get the name of the object photographed

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Cloud Machine Learning Services
  - Request code

```
String requestURL = "https://vision.googleapis.com/v1/images:annotate?key=myKey";

String myJson = '{"requests": [{
    "image": {
      "content": <Base64-encoded image data>
    },
    "features": [{
        "type": "LABEL_DETECTION"
    }]
}]}';
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Cloud Machine Learning Services
  - Request code

```
String requestURL = "https://vision.googleapis.com/v1/images:annotate?key=myKey";

String my
    "image
      "conte
    },
    "featur
        "type
    }]
}]}';
```

```
Fuel.post(requestURL)
  .header(
      new Pair<String, Object>("content-length", body.length()),
      new Pair<String, Object>("content-type", "application/json")
      )
  .body(myJson.getBytes())
  .responseString(new Handler<String>() {
      @Override
      public void success(@NotNull Request request,
                          @NotNull Response response,
                          String data) {
          // More code goes here
      }

      @Override
      public void failure(@NotNull Request request,
                          @NotNull Response response,
                          @NotNull FuelError fuelError) {}
  });
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Cloud Machine Learning Services
  - Response code


```
{"responses":[{
"labelAnnotations": [
  {
    "mid": "/m/id1",
    "description": "label1",
    "score": 0.91
  },
  {
    "mid": "/m/id2",
    "description": "label2",
    "score": 0.90
  },
 ...
}]}
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Cloud Machine Learning Services
  - Response code

```
{"responses":[{
"labelA
  {
    "mid
    "de
    "sco
  },
  {
    "mid
    "de
    "sco
  },
 ...
}]}
```

```
// Access the labelAnnotations arrays
JSONArray labels = new JSONObject(data)
                .getJSONArray("responses")
                .getJSONObject(0)
                .getJSONArray("labelAnnotations");

String results = "";

// Loop through the array and extract the
// description key for each item
for(int i=0;i<labels.length();i++) {
  results = results +
          labels.getJSONObject(i).getString("description") +
          "\n";
}

// Display the annotations inside the TextView
((TextView)findViewById(R.id.resultsText)).setText(results);
```

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- Cloud Machine Learning Services
  - Response code

```
{"responses":[{
"labelA
  {
    "mid
    "des
    "sco
  },
  {
    "mid
    "des
    "sco
  },
 ...
}]}
```

```
// Access the labelAnnotations arrays
JSONArray labels = new JSONObject(data)
                .getJSONArray("responses")
                .getJSONObject(0)
                .getJSONArray("labelAnnotations");

String results = "";

// Loop through the array and extract the
// description key for each item
for(int i=0;i<labels.length();i++) {
  results = results +
          labels.getJSONObject(i).getString("description") +
          "\n";
}

// Display the annotations inside the TextView
((TextView)findViewById(R.id.resultsText)).setText(results);
```

INDEX

- ME
- MOTIVATION
- WEB SERVICES
  - HTTP
  - REST
  - XML/JSON
- EXAMPLES
  - IPIFY
  - NASA
  - MAPS
  - FIREBASE
  - PLAY GAMES
  - CLOUD MACHINE LEARNING
  - THINGSPEAK (IoT)

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- ThingSpeak
  - Open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet
  - Originally launched by ioBridge in 2010 as a service in support of IoT applications
  - URL: https://thingspeak.com/
  - What we can do with Thingspeak?
    1. Send sensor data to the cloud
    2. Analyze and visualize the data with MATLAB
    3. Trigger a reaction
  - Works with?
    - Arduino
    - RaspberryPi
    - Mobile and Web apps
    - …

ThingSpeak

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

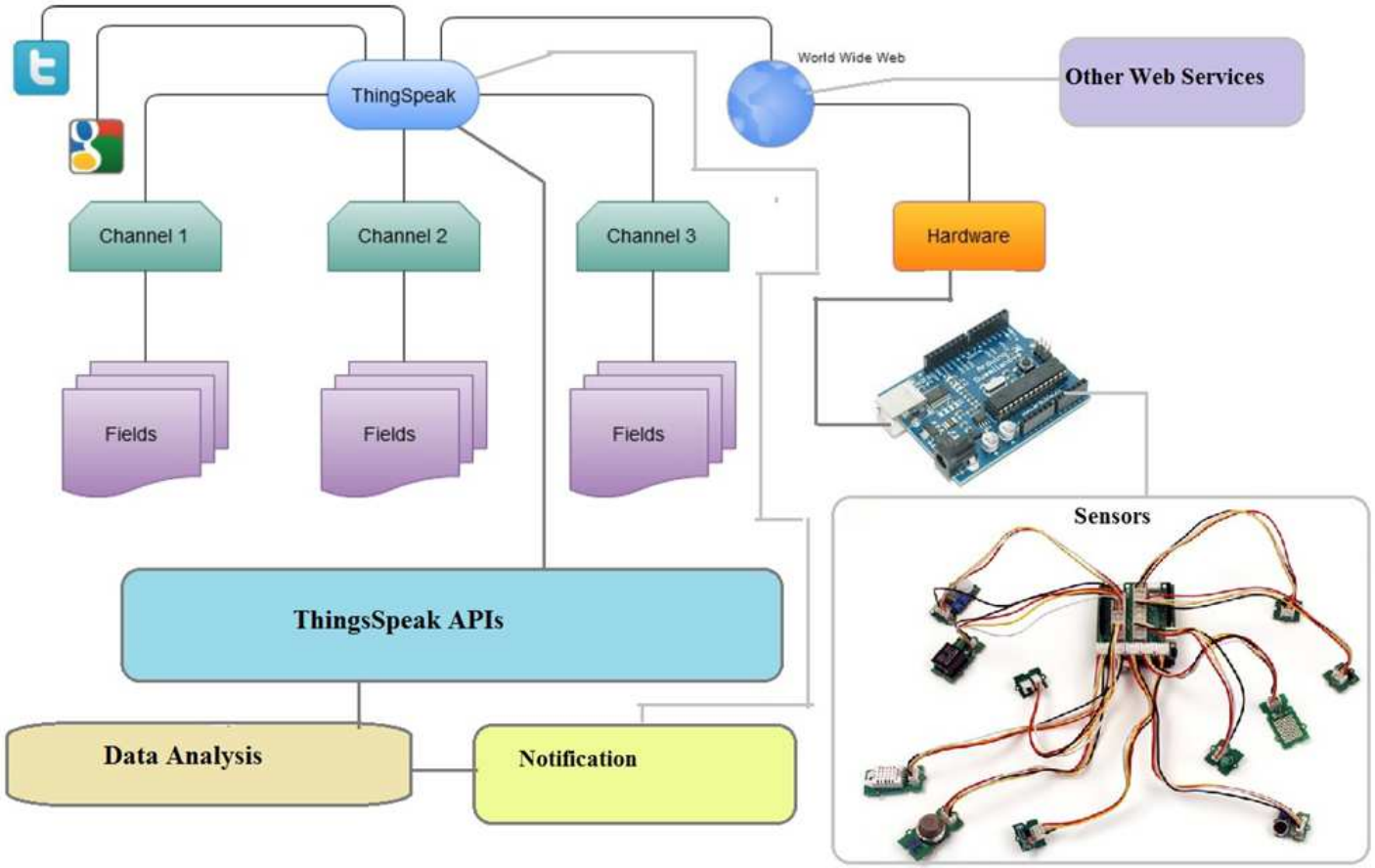
- ThingSpeak
  - ThingSpeak is an IoT platform that uses channels to store data sent from apps or devices
  - You create a channel, and then send and retrieve data to and from the channel
  - You can make your channels public to share data
  - Using the REST API calls such as GET, POST, PUT, and DELETE, you can:
    - create a channel
    - update its feed
    - update an existing channel
    - clear a channel feed
    - delete a channel

ThingSpeak

ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

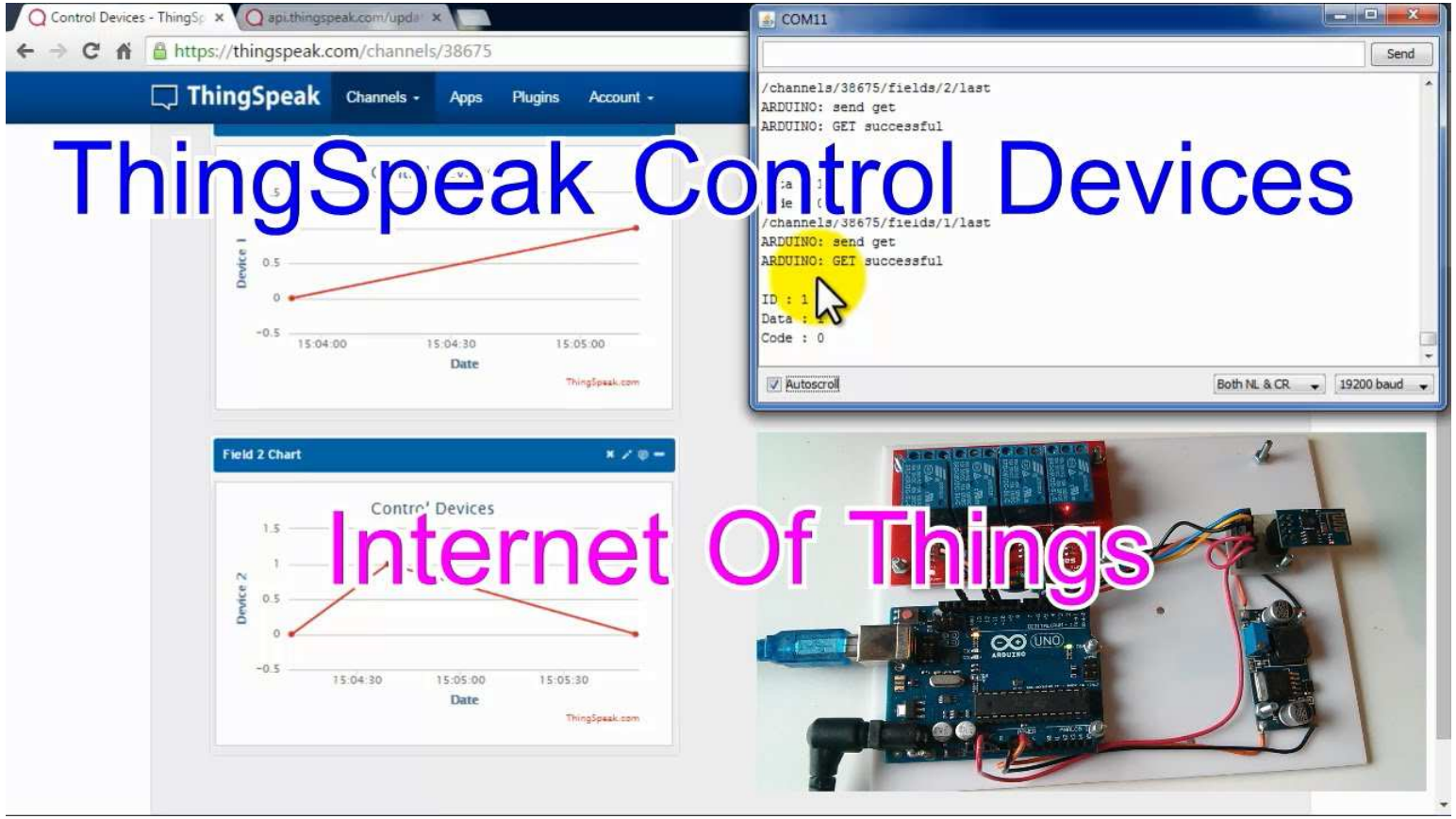
- ThingSpeak



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

EXAMPLES

- ThingSpeak
  - Example



ARCHITECTURAL INTEGRATION FROM MONOLITHIC TO CLOUD BASED SERVICES

Q/A

Ricardo Queirós
ricardoqueiros@esmad.ipp.pt

"I practice until I have my life in my fingers"
Pianists' expression