



# Social Media Apps Programming

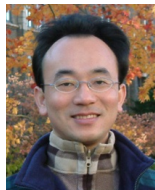
## Google App Engine and Google Map API

1051SMAP11

TLMXM1A (8648) (M2143) (Fall 2016)

(MIS MBA) (2 Credits, Elective) [Full English Course]

Wed 8,9 (15:10-17:00) B310



Min-Yuh Day, Ph.D.

Assistant Professor

Department of Information Management

Tamkang University

<http://mail.tku.edu.tw/myday>

2016-12-14



# Course Schedule (1/3)

Week	Date	Subject/Topics
1	2016/09/14	Course Orientation and Introduction to Social Media and Mobile Apps Programming
2	2016/09/21	Introduction to Android / iOS Apps Programming
3	2016/09/28	Developing Android Native Apps with Java (Eclipse) (MIT App Inventor)
4	2016/10/05	Developing iPhone / iPad Native Apps with Swift (XCode)
5	2016/10/12	Mobile Apps using HTML5/CSS3/JavaScript
6	2016/10/19	jQuery Mobile

# Course Schedule (2/3)

Week	Date	Subject/Topics
7	2016/10/26	Create Hybrid Apps with PhoneGap
8	2016/11/02	jQuery Mobile/PhoneGap
9	2016/11/09	jQuery Mobile/PhoneGap
10	2016/11/16	Midterm Exam Week (Midterm Project Report)
11	2016/11/23	Case Study on Social Media Apps Programming and Marketing in Google Play and App Store
12	2016/11/30	Invited Talk [B302b]: Challenges in Natural Language Processing: Question Answering and Dialog System Invited Speaker: Prof. Yoshinobu Kano, Associate Professor, Faculty of Informatics, Shizuoka University, Japan

# Course Schedule (3/3)

Week	Date	Subject/Topics
13	2016/12/07	Google Cloud Platform
14	2016/12/14	Google App Engine and Google Map API
15	2016/12/21	Facebook API (Facebook JavaScript SDK) (Integrate Facebook with iOS/Android Apps)
16	2016/12/28	Twitter API
17	2017/01/04	Final Project Presentation
18	2017/01/11	Final Exam Week (Final Project Presentation)



# Outline

- **Google App Engine**



- **Google Cloud Platform**



- **Google Cloud Datastore**



- **Google Firebase**

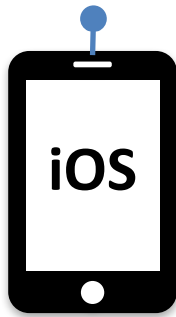


Firestore

- **Google Maps API**



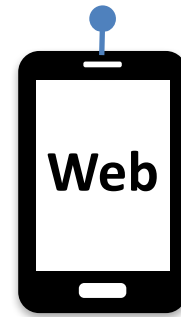
# App Frontend



**iOS**



**Android**



**Web**

# Mobile Apps Backend on Google Cloud

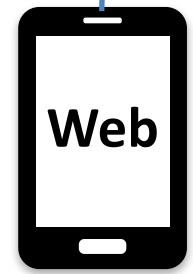
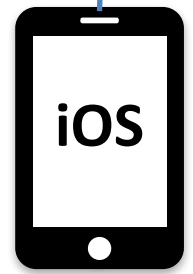


Firebase



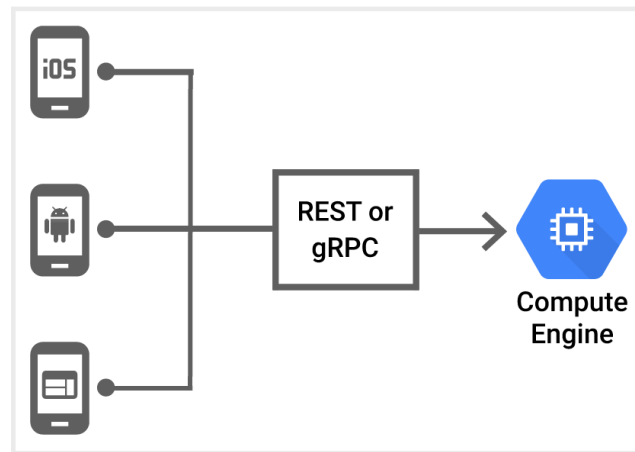
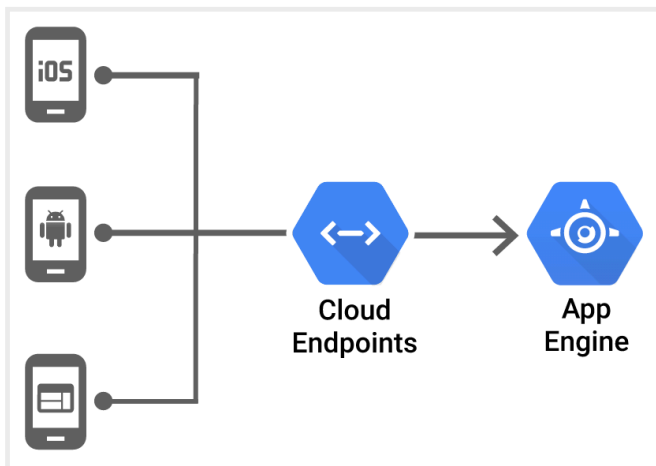
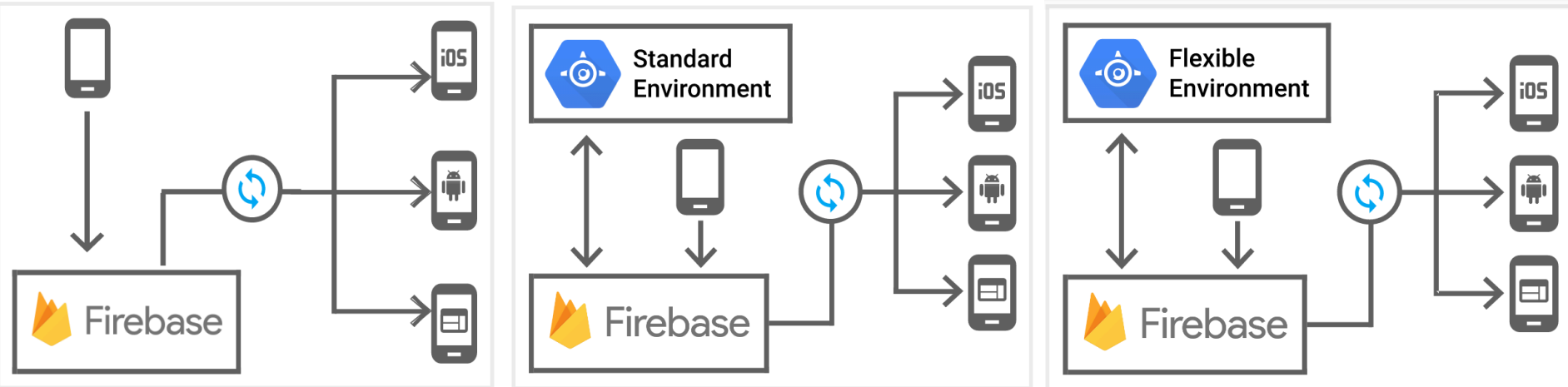
App

Backend

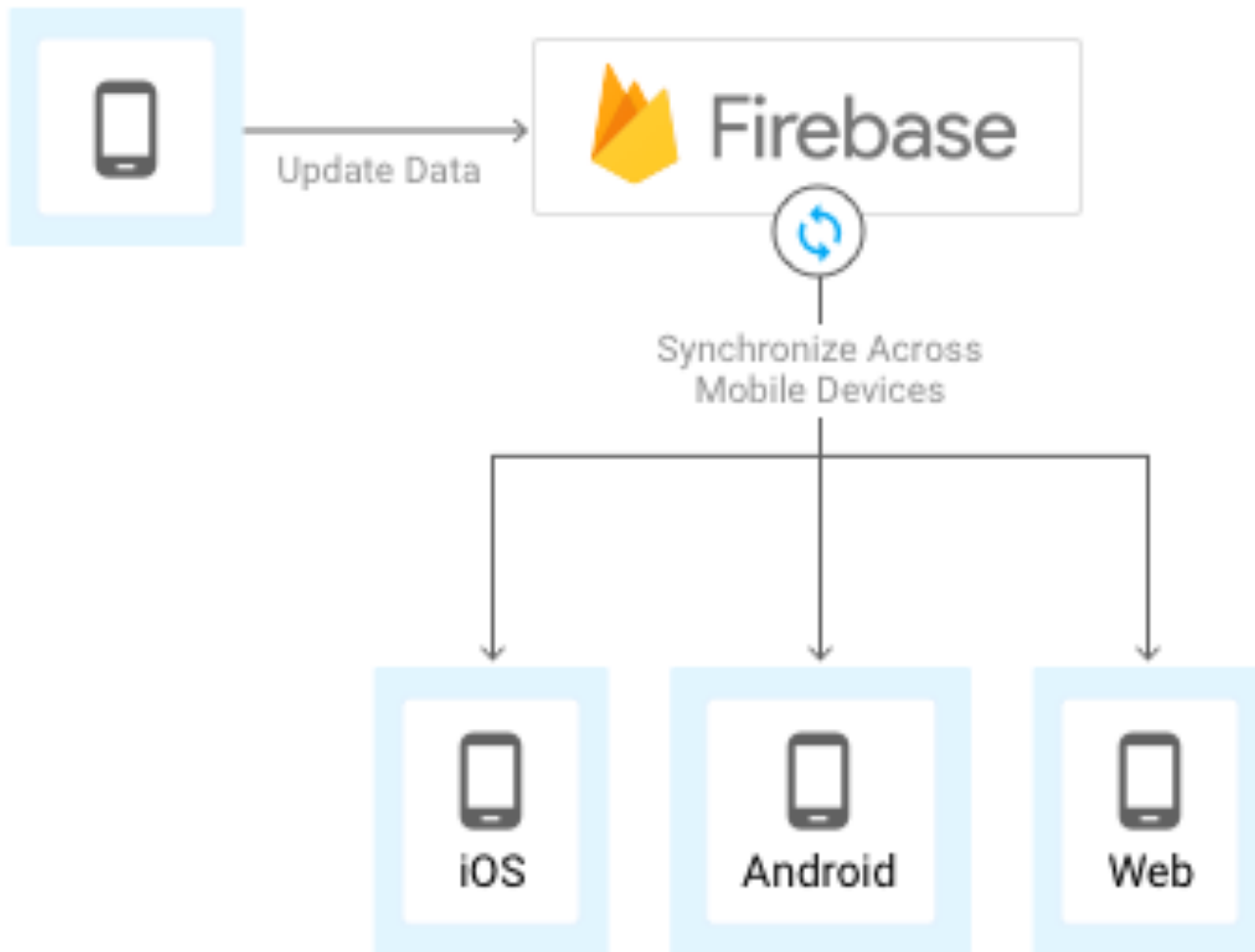
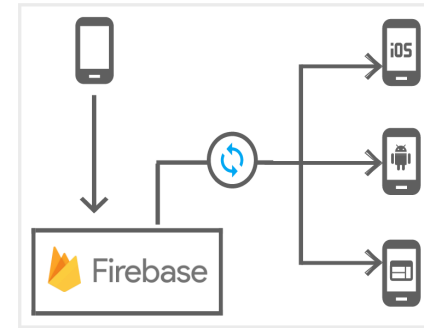


# Frontend

# Mobile App Backend Services

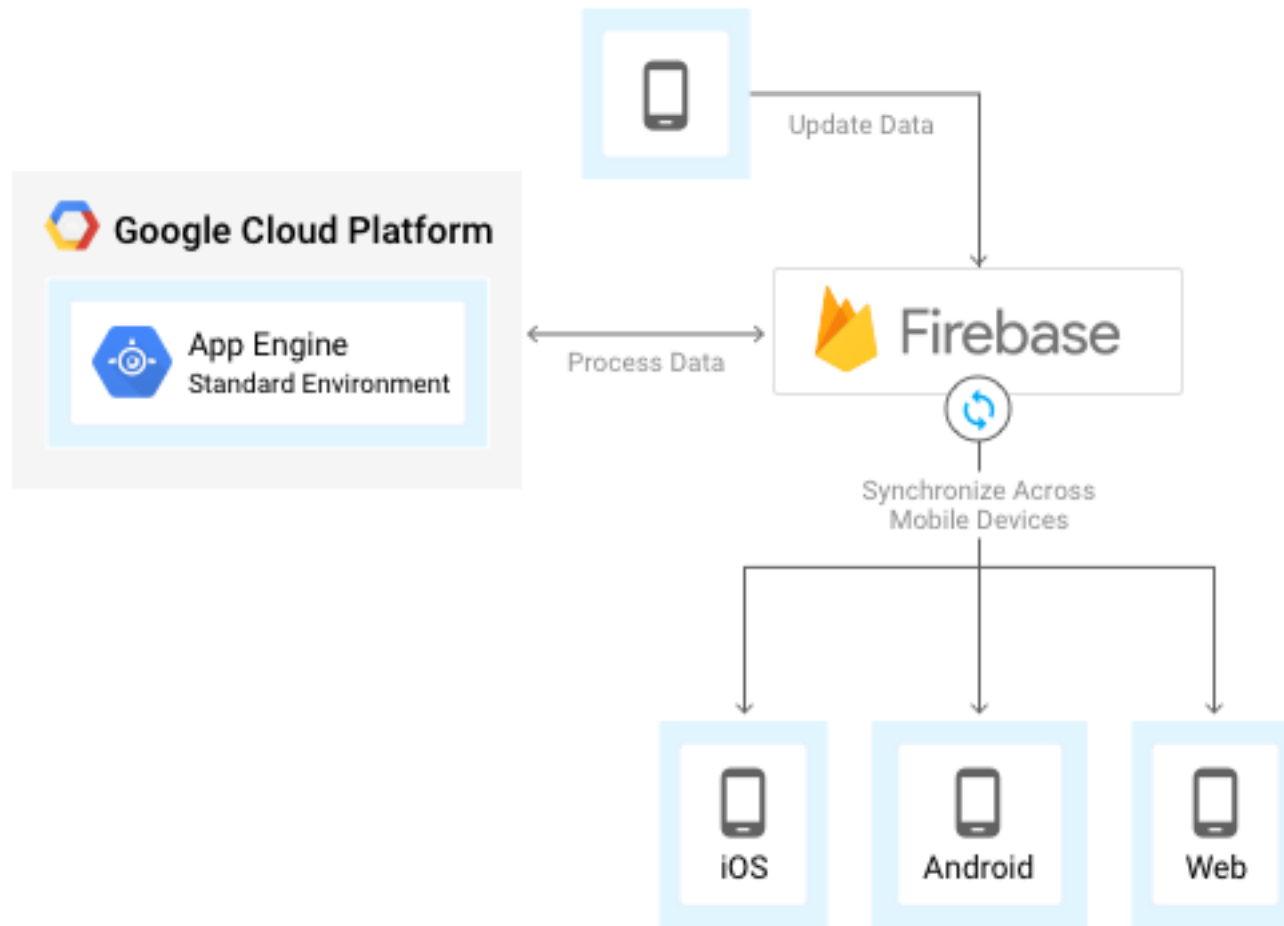
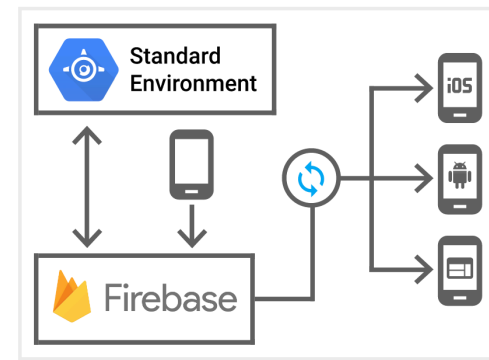


# Firestore



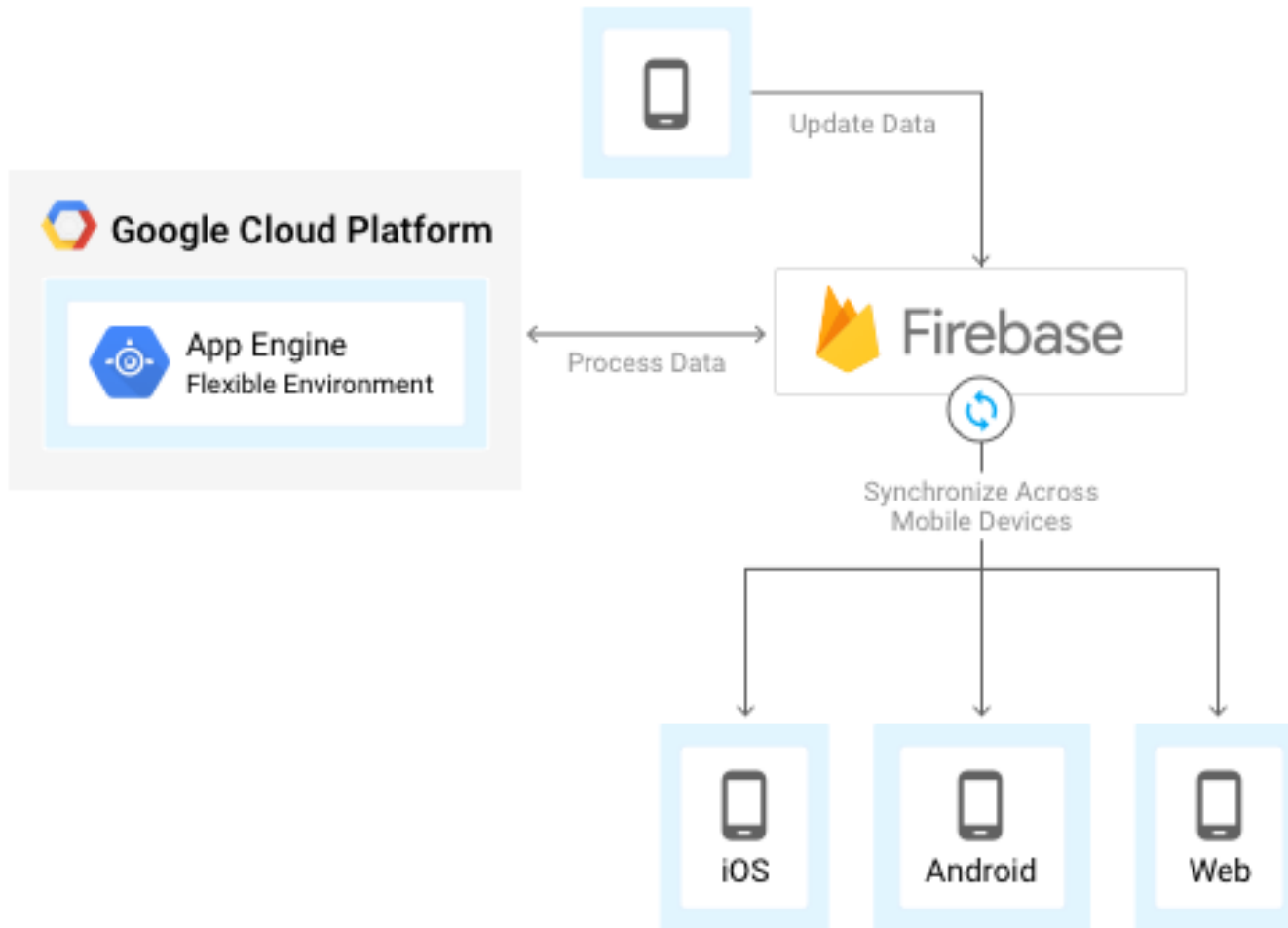
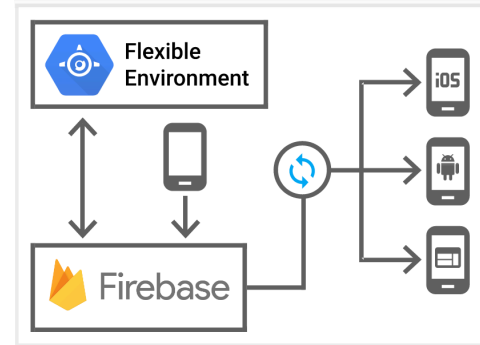


# Firestore and Google App Engine standard environment



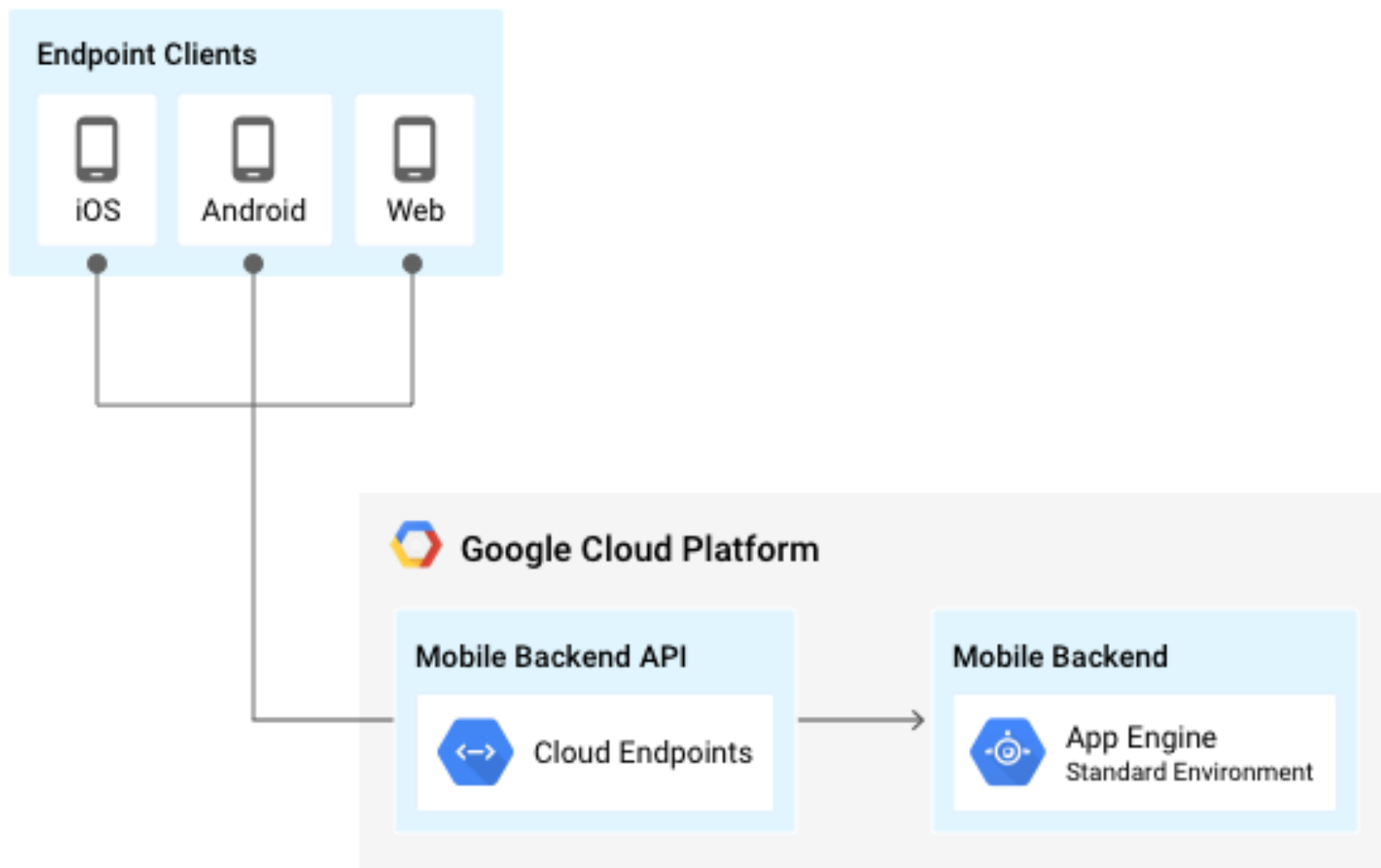
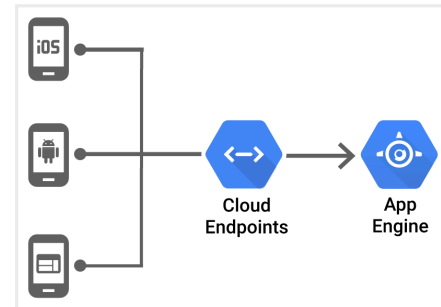


# Firestore and App Engine flexible environment



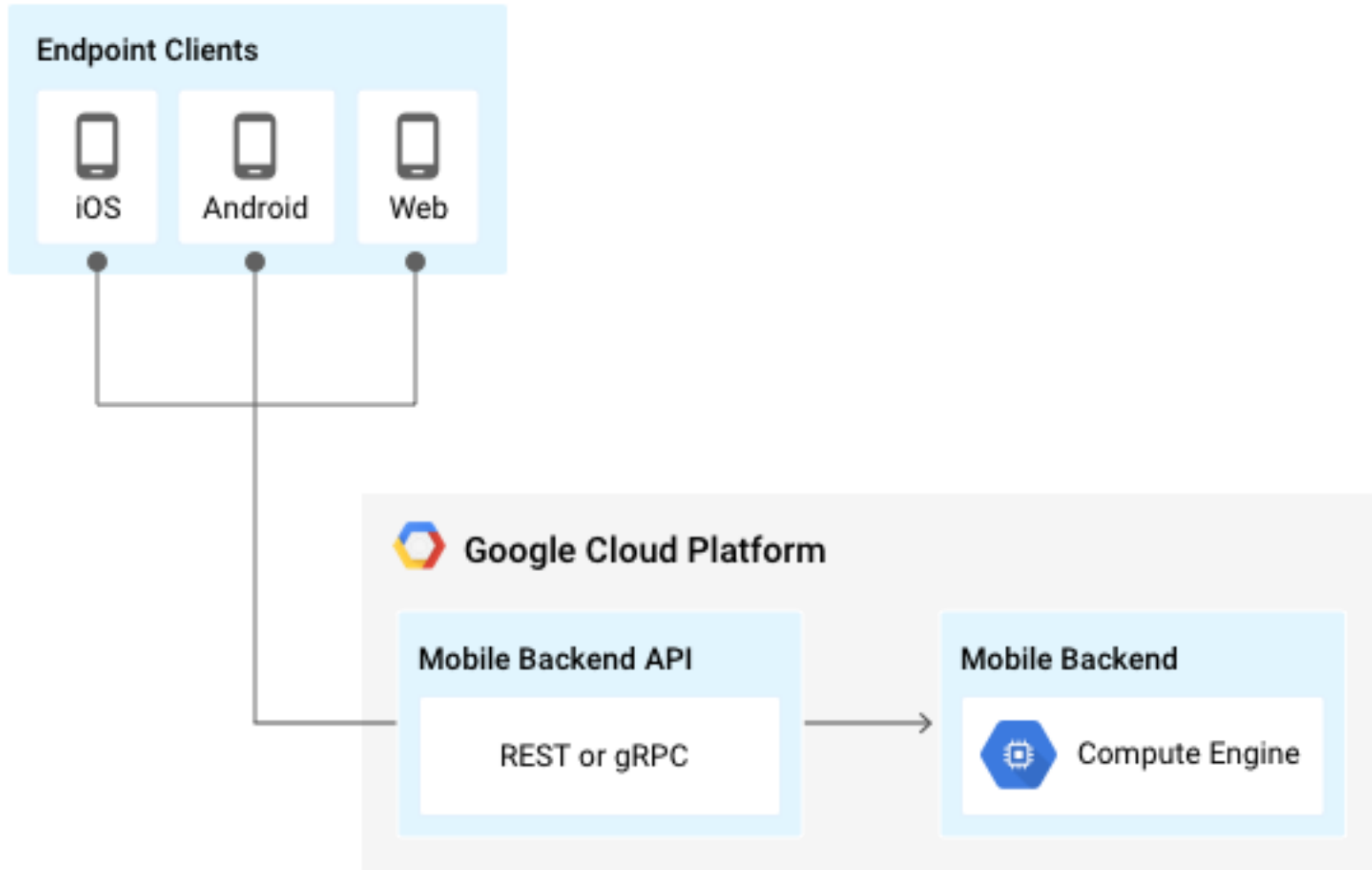
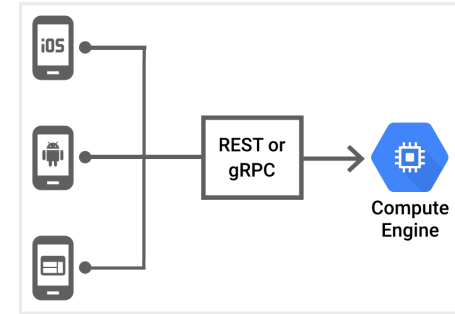


# App Engine and Cloud Endpoints





# Compute Engine and REST or gRPC



# Storing data and Exchanging data

# JSON

- JSON
  - JavaScript Object Notation.
- JSON is a syntax for storing and exchanging data.
- JSON is an easier-to-use alternative to XML.

# JSON

```
{"employees":  
  {"firstName":"John", "lastName":"Doe"},  
  {"firstName":"Anna", "lastName":"Smith"},  
  {"firstName":"Peter", "lastName":"Jones"}  
}]
```

# XML

```
<employees>  
  <employee>  
    <firstName>John</firstName> <lastName>Doe</lastName>  
  </employee>  
  <employee>  
    <firstName>Anna</firstName> <lastName>Smith</lastName>  
  </employee>  
  <employee>  
    <firstName>Peter</firstName> <lastName>Jones</lastName>  
  </employee>  
</employees>
```

# JSON vs. XML

```
{"employees":[
  {"firstName":"John", "lastName":"Doe"},
  {"firstName":"Anna", "lastName":"Smith"},
  {"firstName":"Peter", "lastName":"Jones"}
]}
```

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

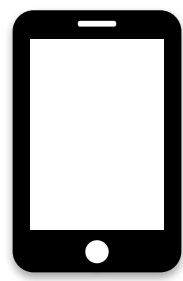
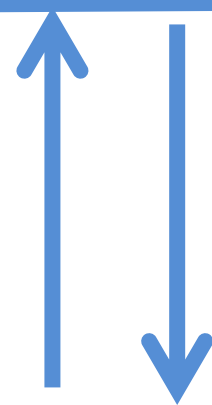
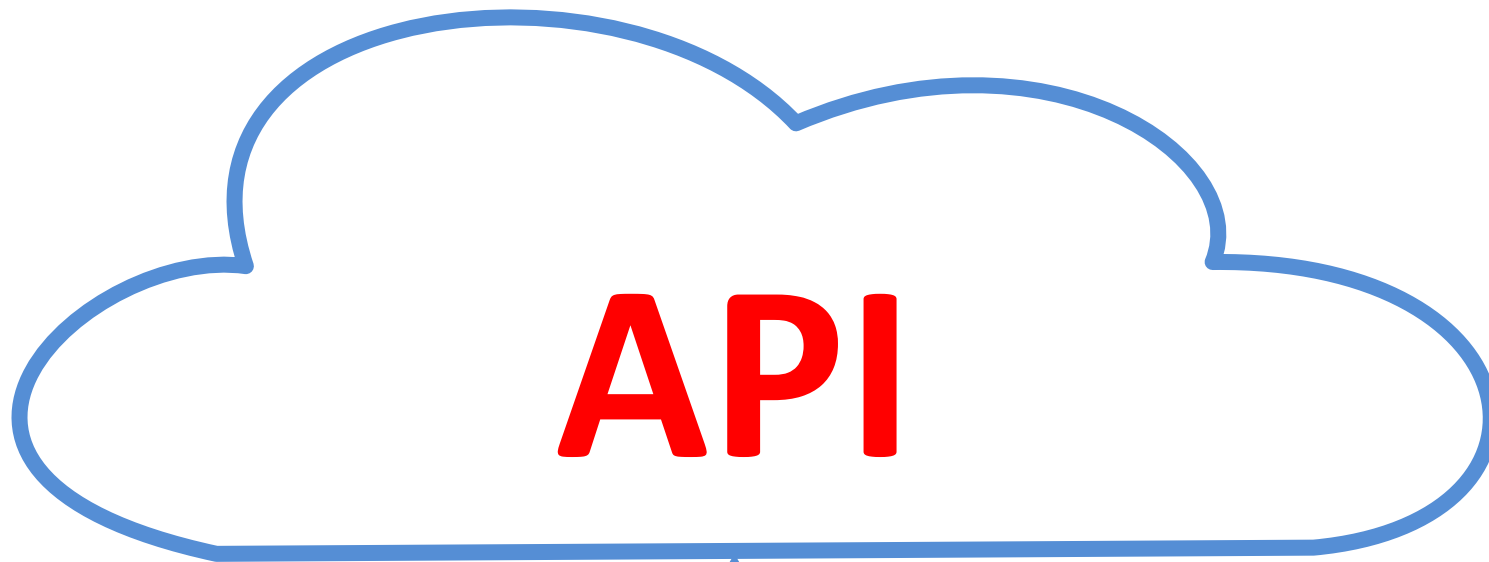


# API

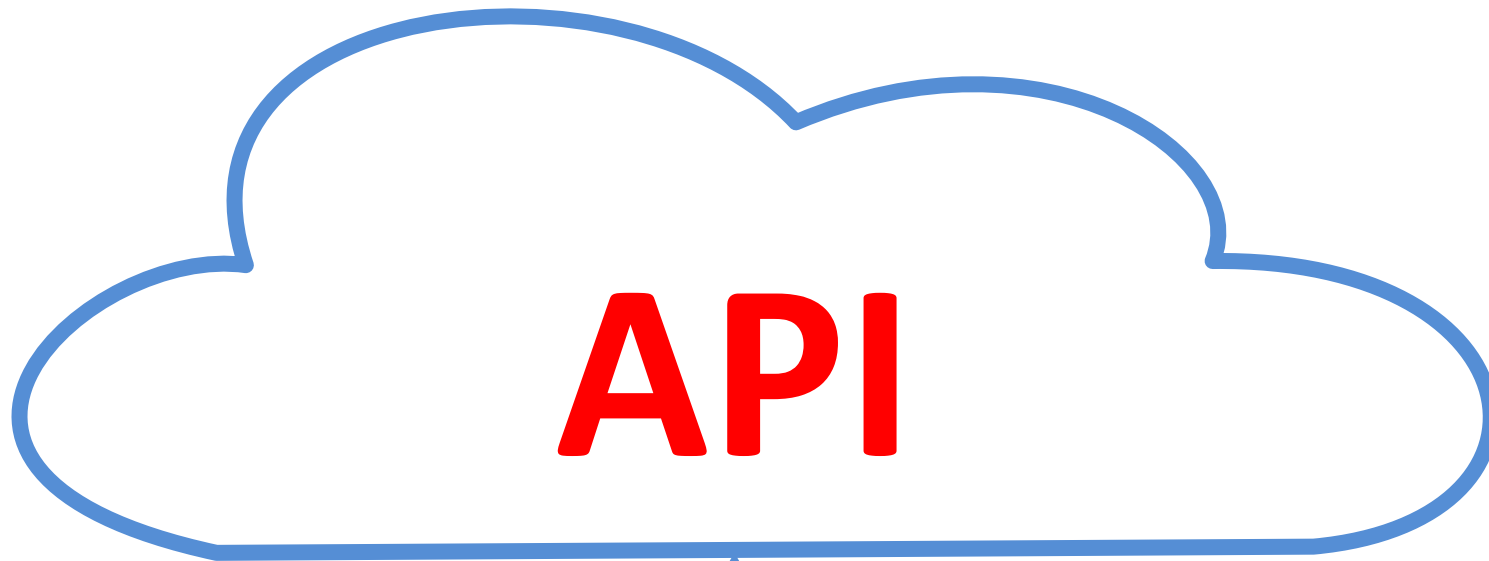
**A**pplication

**P**rogramming

**I**nterface



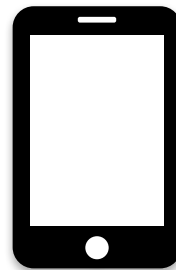
**Your App**



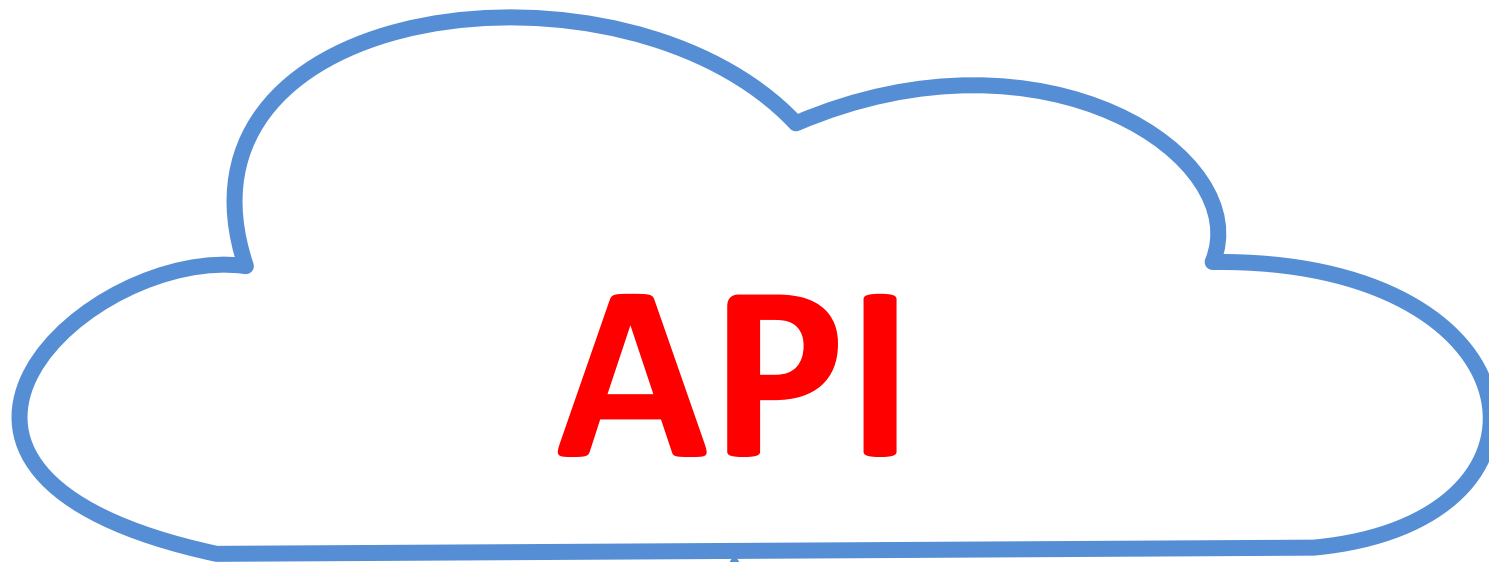
**API**

**Request**

<http://graph.facebook.com/4>



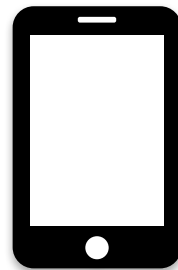
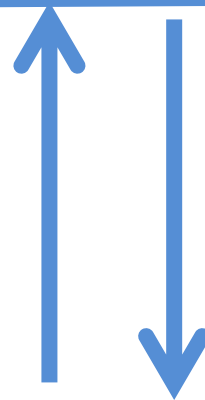
**Your App**



**API**

**Request**

<http://graph.facebook.com/4>

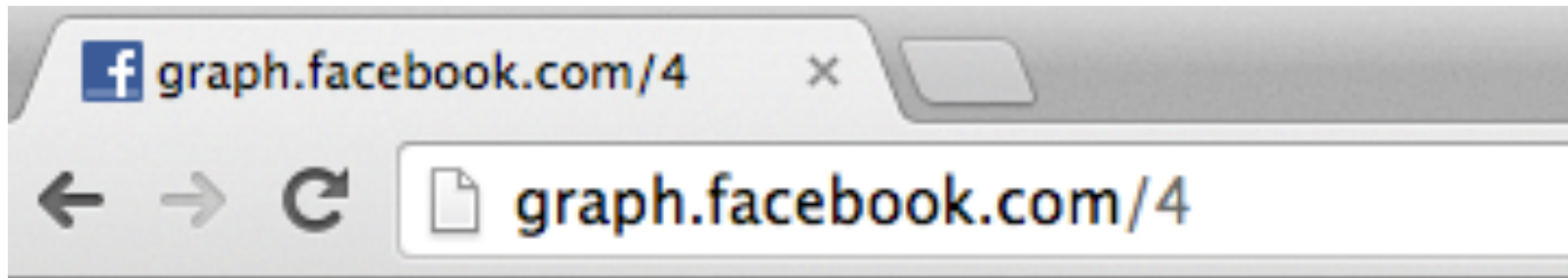


**Response**

```
{  
  "id": "4",  
  "first_name": "Mark",  
  "gender": "male",  
  "last_name": "Zuckerberg",  
  "link": "https://www.facebook.com/zuck",  
  "locale": "en_US",  
  "name": "Mark Zuckerberg",  
  "username": "zuck"  
}
```

**Your App**

# <http://graph.facebook.com/4>



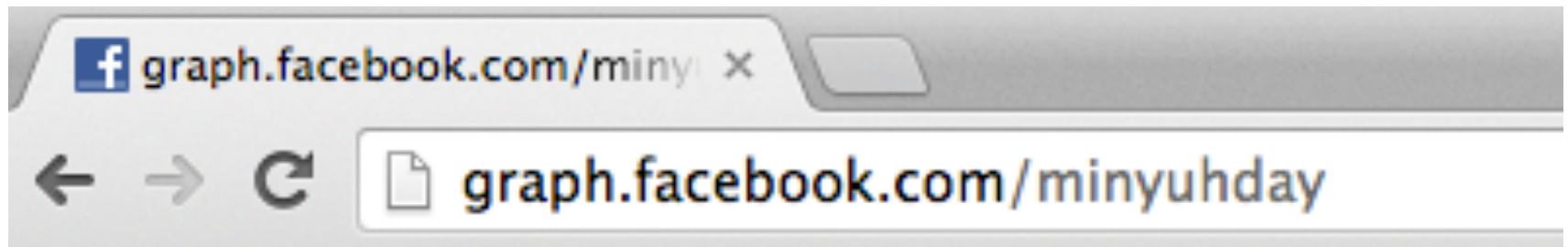
```
{
  "id": "4",
  "name": "Mark Zuckerberg",
  "first_name": "Mark",
  "last_name": "Zuckerberg",
  "link": "http://www.facebook.com/zuck",
  "username": "zuck",
  "gender": "male",
  "locale": "en_US"
}
```

# Facebook API (JSON)

<http://graph.facebook.com/4>

```
{  
  "id": "4",  
  "first_name": "Mark",  
  "gender": "male",  
  "last_name": "Zuckerberg",  
  "link": "https://www.facebook.com/zuck",  
  "locale": "en_US",  
  "name": "Mark Zuckerberg",  
  "username": "zuck"  
}
```

# <http://graph.facebook.com/minyuhday>



```
{  
  "id": "684393172",  
  "name": "Min-Yuh Day",  
  "first_name": "Min-Yuh",  
  "last_name": "Day",  
  "link": "http://www.facebook.com/minyuhday",  
  "username": "minyuhday",  
  "gender": "male",  
  "locale": "en_US"  
}
```

# JavaScript vs. JSON

- JSON

- JavaScript Object Notation

- Format for sharing data
    - Derived from JavaScript
    - Language independent
    - An alternative to XML



# JSON

- Advantages

- Easy to read
- Easy to write
- Easy to Parse

```
Var info = JSON.parse(data);  
info.name  
info.position  
info.courses[i]
```

- Learner than XML
- Growing support in APIs (i.e., Facebook, Twitter)
- Natural format for JavaScript
- Implementation in many languages

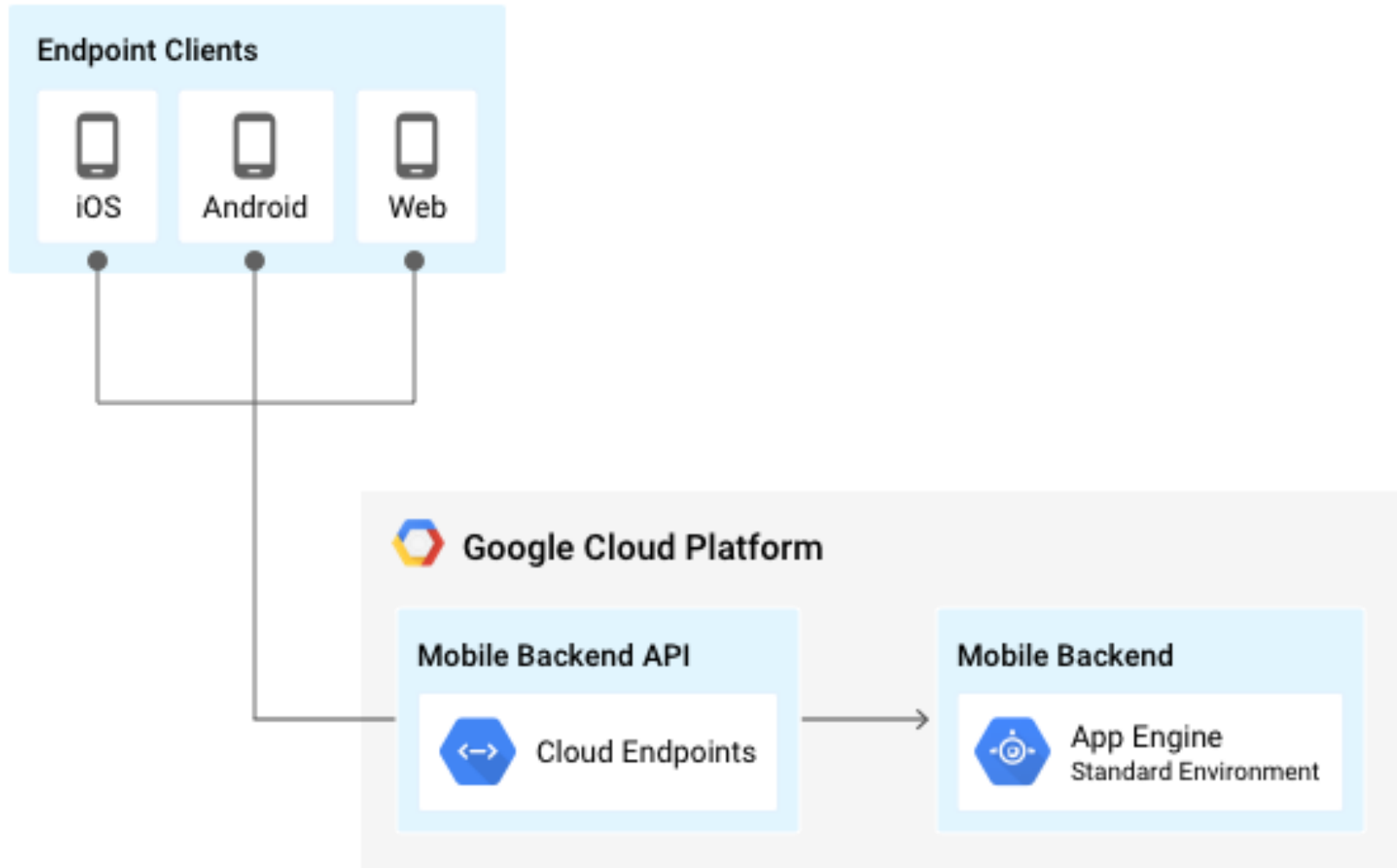
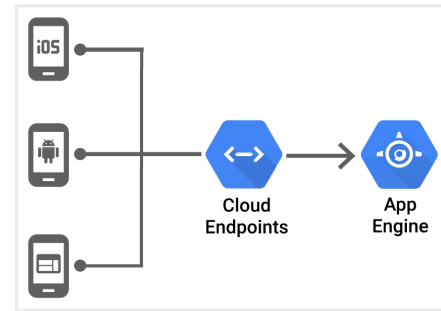
# JSON

```
{  
  "name" : "Min-Yuh Day"  
  "position" : "Assistant Professor"  
  "courses" : [  
    "Social Media Apps Programming"  
    "Social Media Marketing"  
    "Data Mining"  
  ]  
}
```

```
var info = JSON.parse(data);  
info.name  
info.position  
info.courses[i]
```



# App Engine and Cloud Endpoints





# Google Cloud Platform

## Compute



App Engine



Compute Engine



Container Engine

## Storage



Cloud Datastore



Cloud SQL



Cloud Storage

## Big Data



Big Query

## Services



Cloud Endpoints



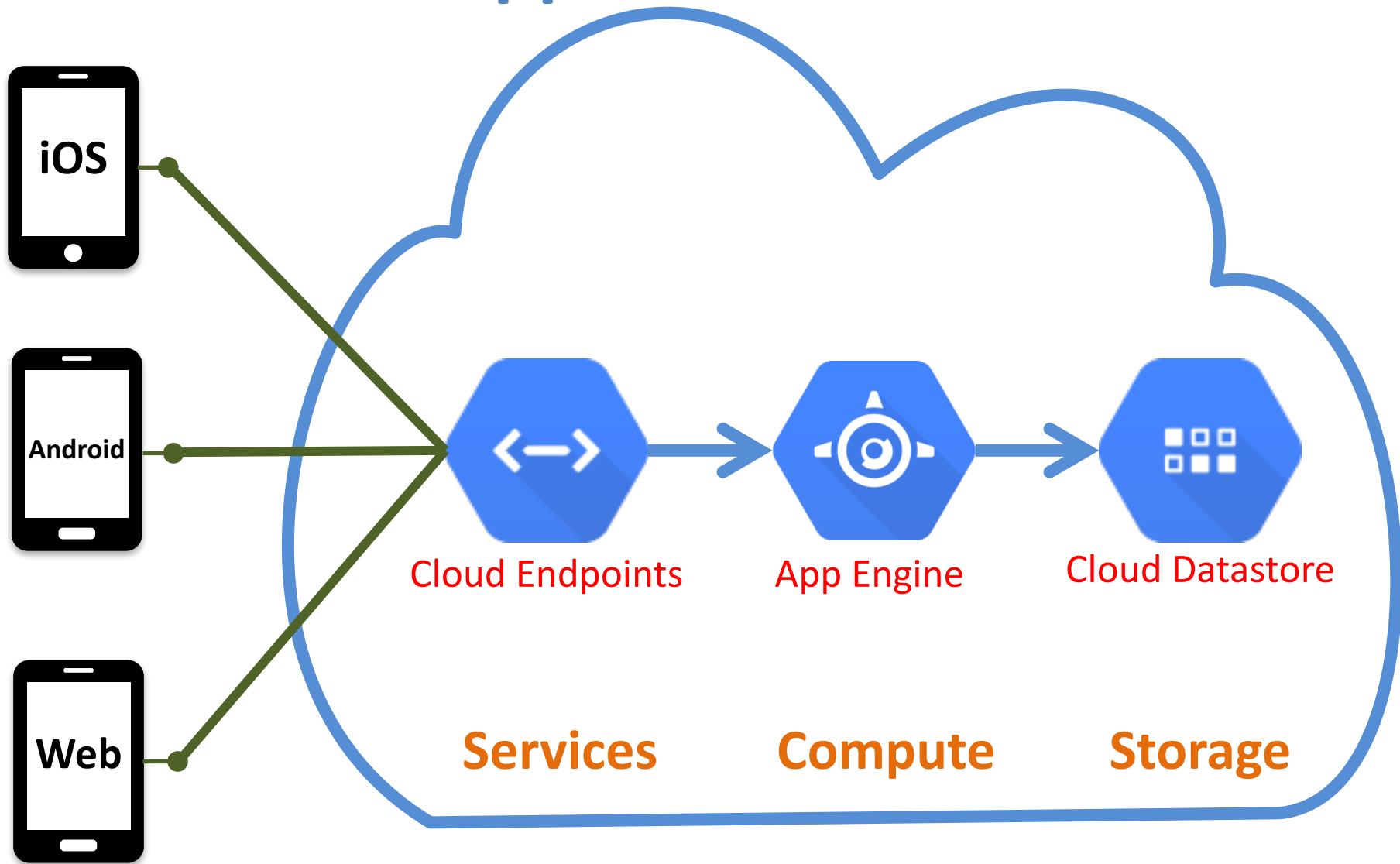
Translate API



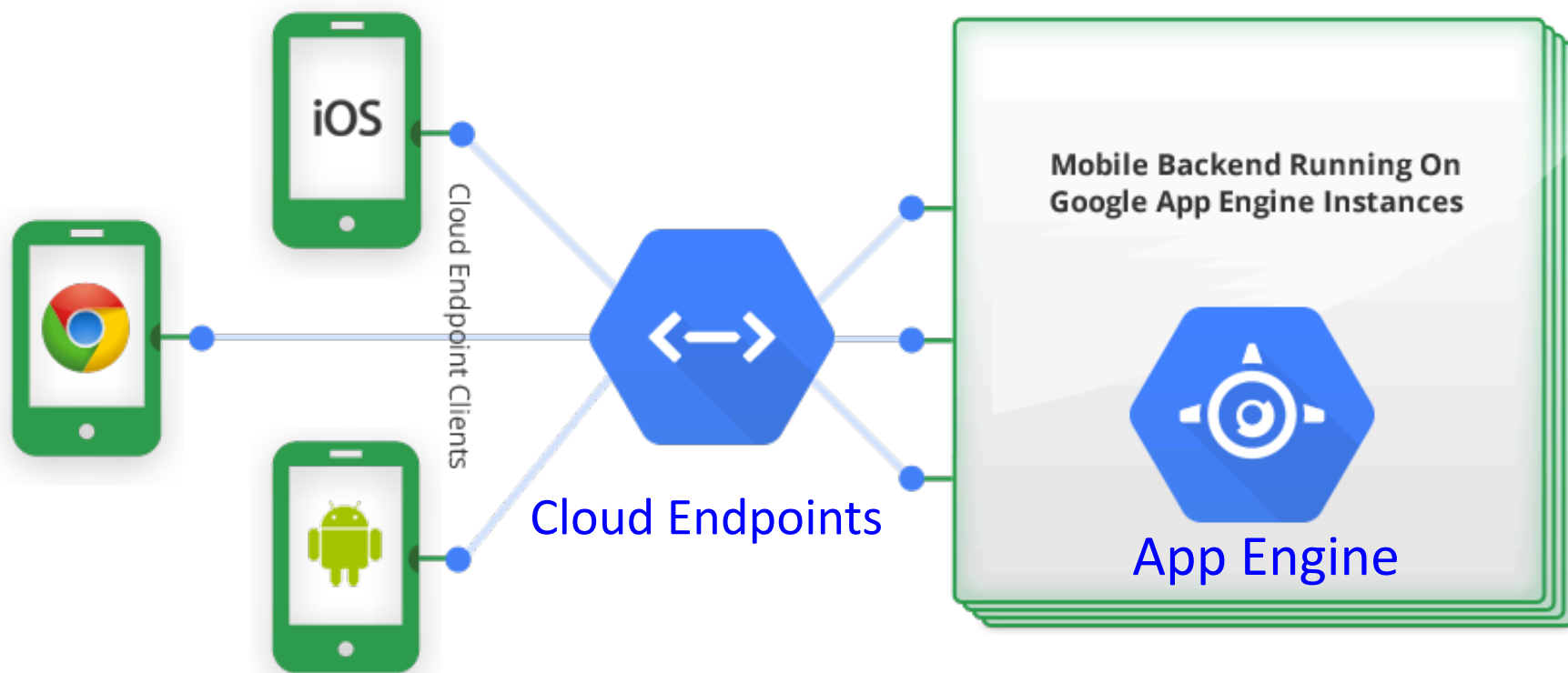
Prediction API



# Mobile App Backend Services



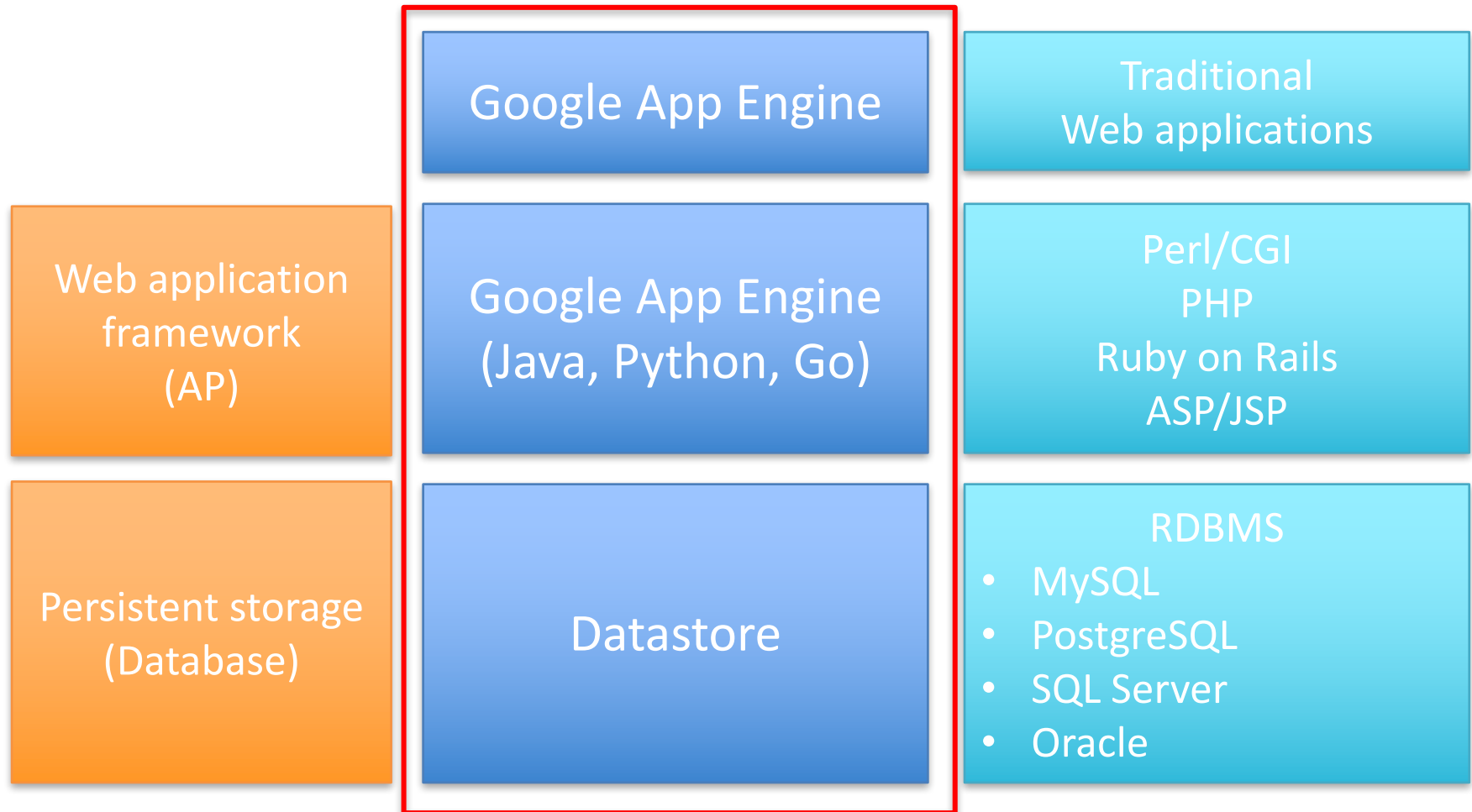
# Mobile Apps Backend on Google App Engine




## Google Cloud Endpoints Architecture

# Google App Engine, Google Cloud Datastore


**Datastore** is a database (persistent storage) for **App Engine**

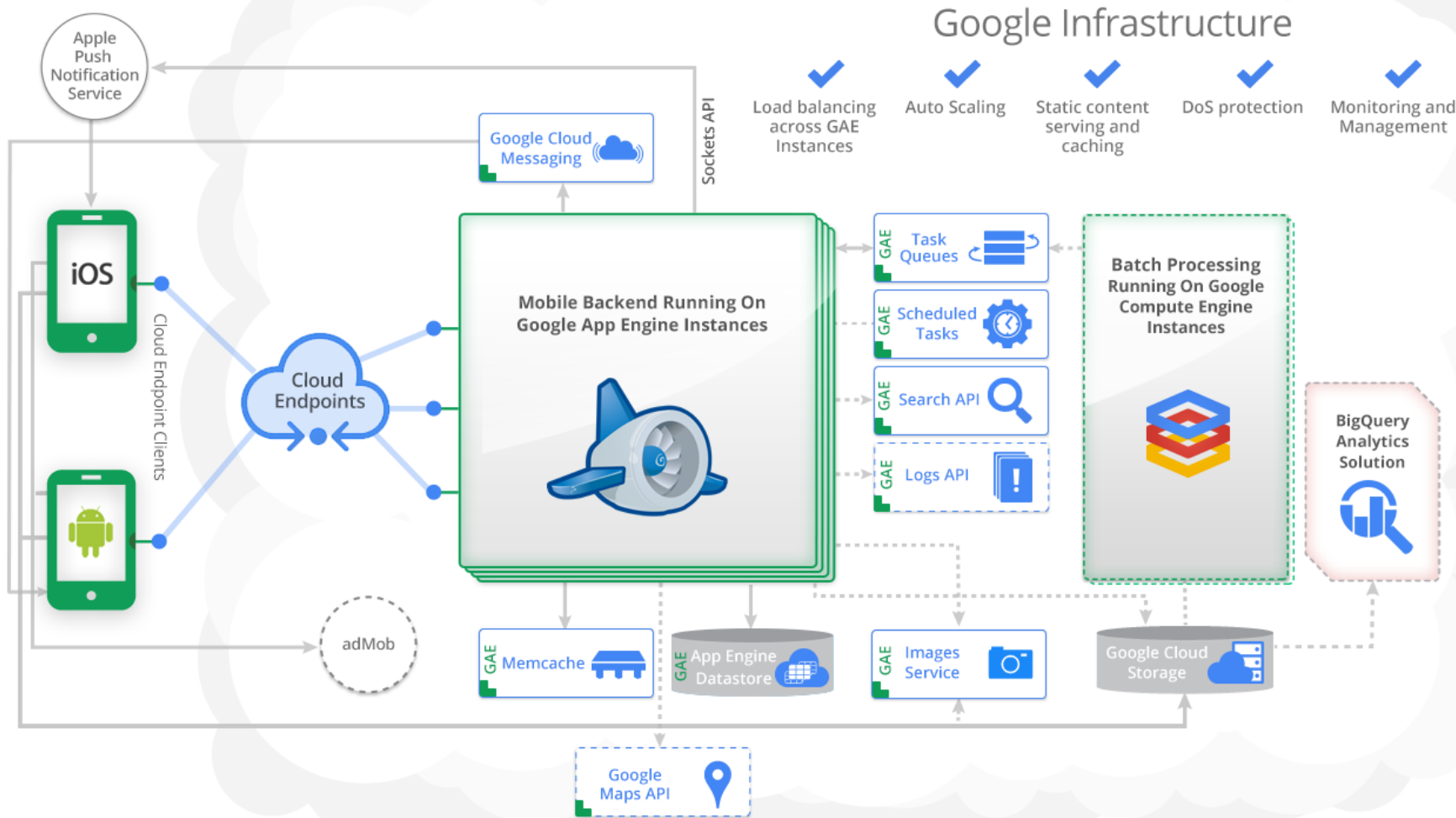


# Mobile Solutions on the Google Cloud Platform

 Your Application Code running on Google App Engine (GAE), Google Compute Engine (GCE), and Client Devices

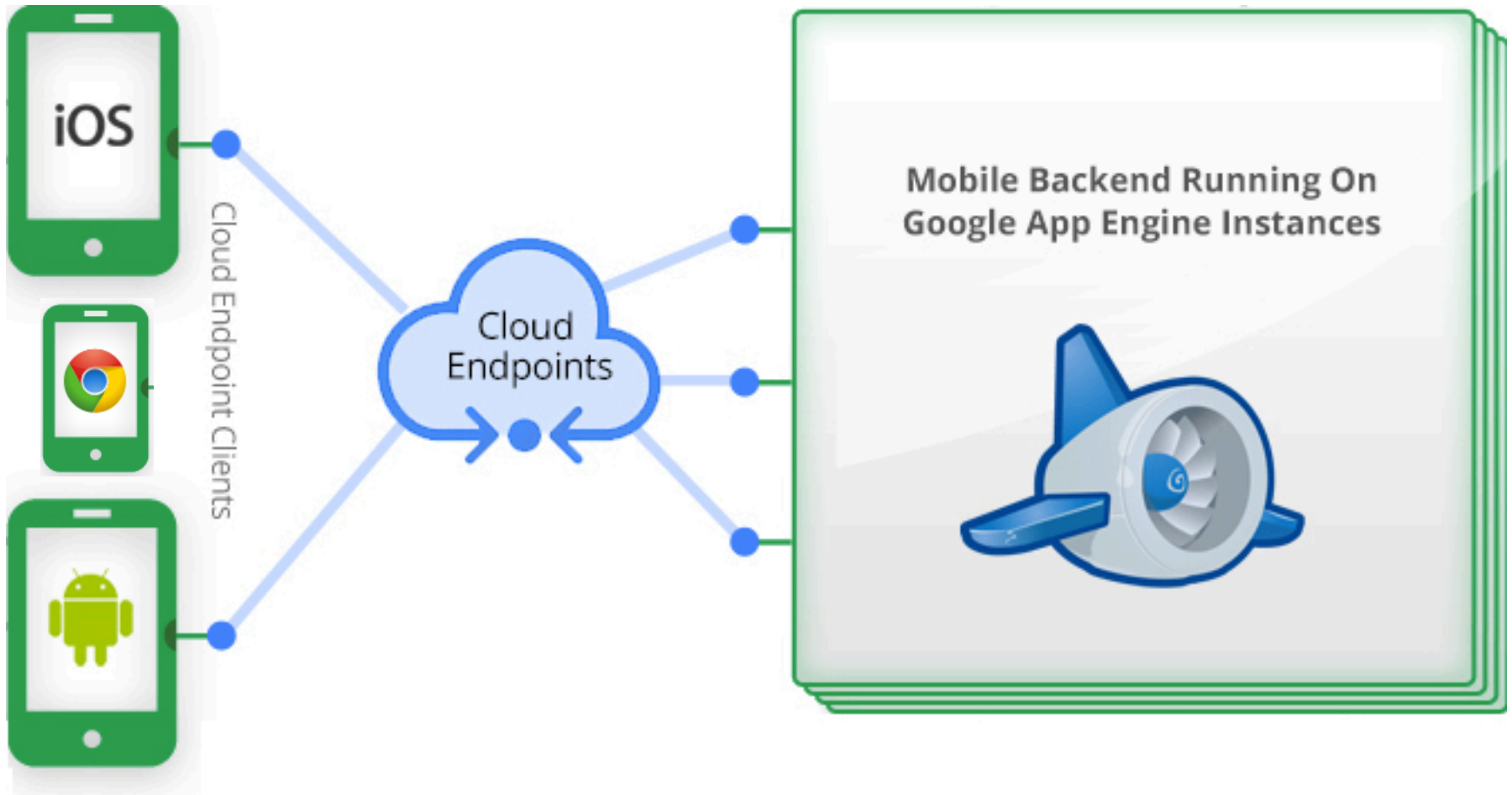
 Google Cloud Platform Services  Optional components

 Capabilities Included

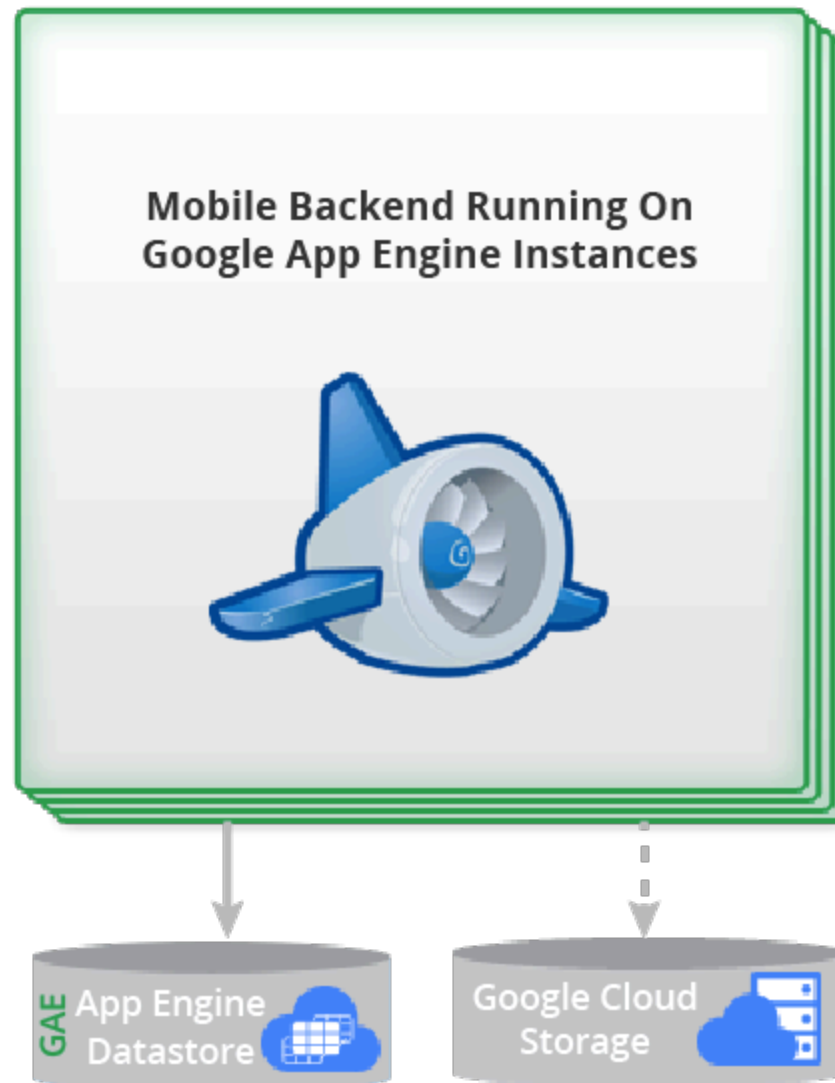




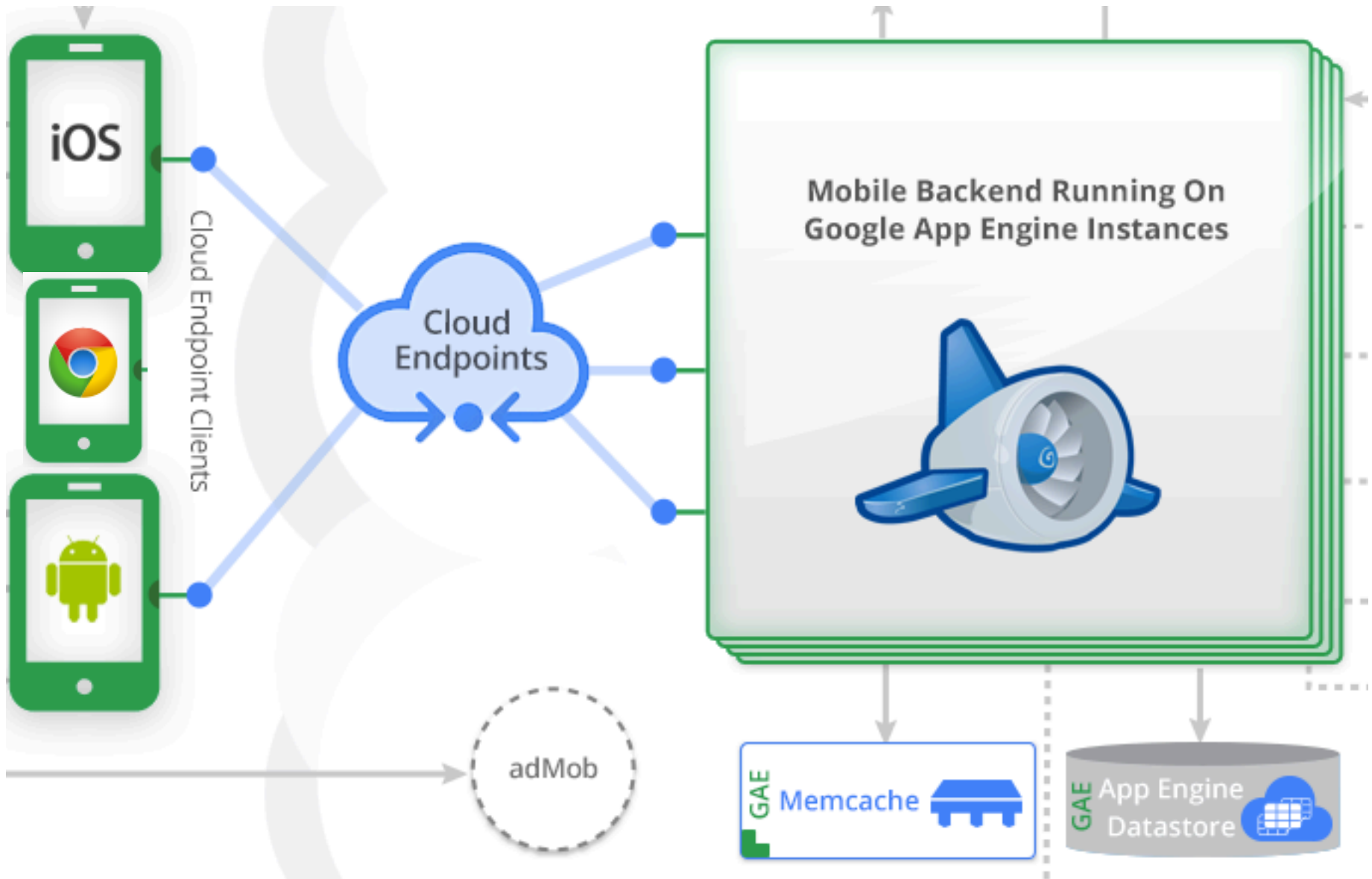
# Mobile App Solutions Architecture



# Storing data



# Optimizing data access with Memcache





# Google App Engine Platform as a Service (PaaS)

build and run applications on  
Google's infrastructure



# Google App Engine

## Platform as a Service (PaaS)

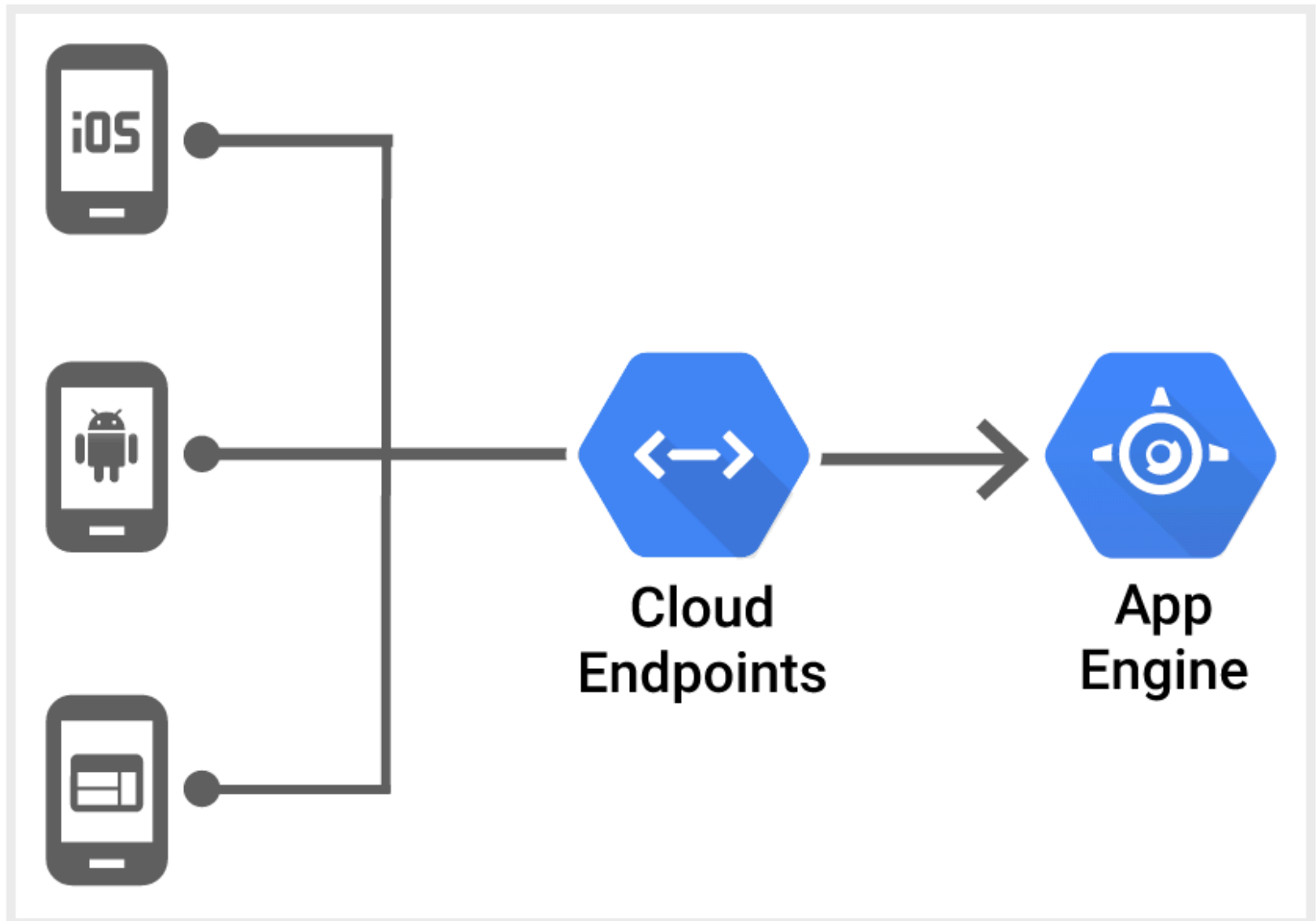


# Google Compute Engine

## Infrastructure as a Service (IaaS)

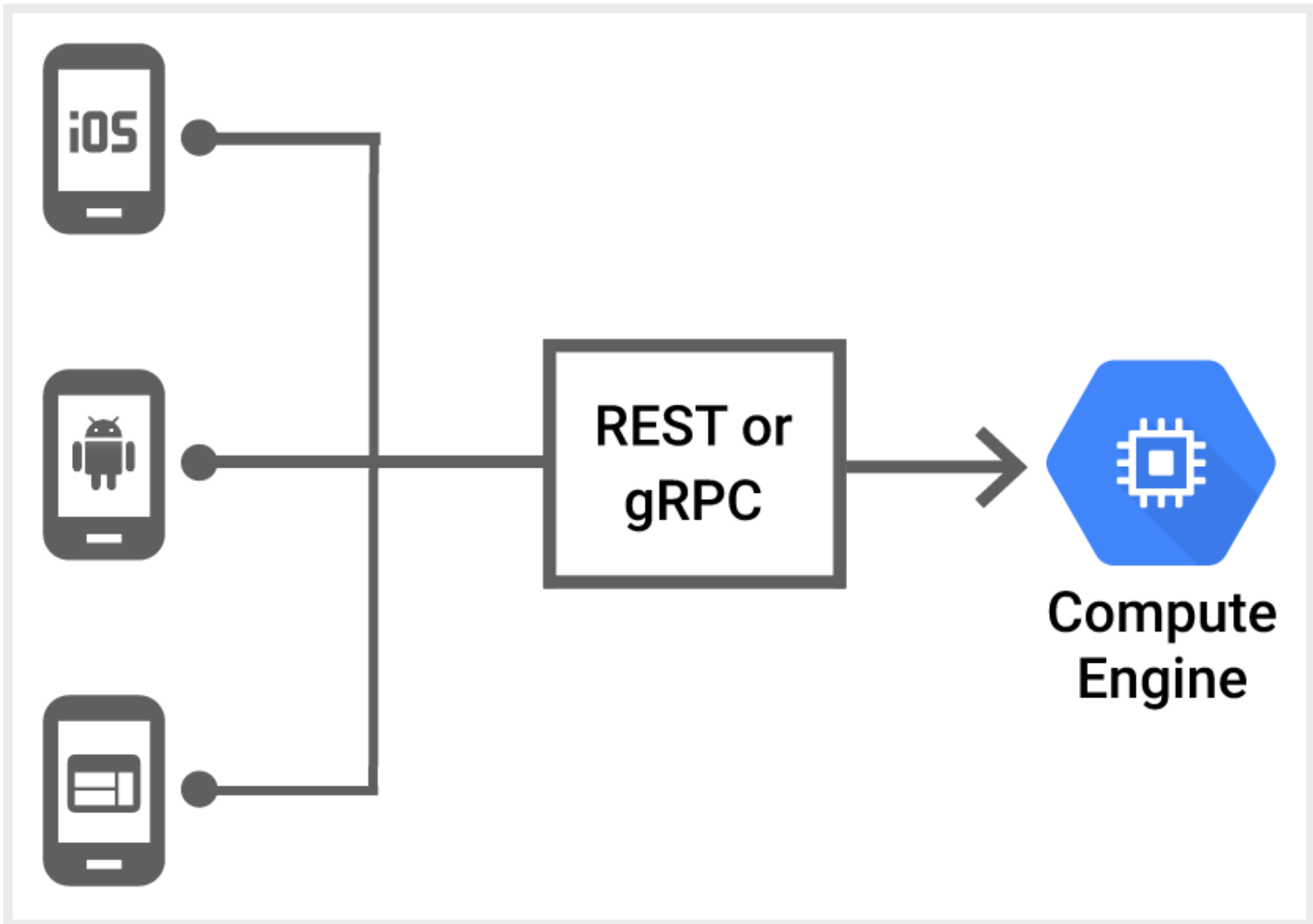


# Mobile App Backend Services





# Mobile App Backend Services



# Google App Engine

- **1 GB** of data storage and traffic for **free**
  - can be increased by enabling paid applications



# Google App Engine

Google App Engine supports apps written in a variety of programming languages: Python, Java, PHP, Go



Python



Java



PHP



Go



# Google App Engine

The screenshot shows the Google App Engine website homepage. At the top, there's a navigation bar with the Google Cloud Platform logo, a search bar, and links for 'Go to my console', 'Sign out', 'Free Trial', and 'Contact Sales'. Below the navigation bar, the main heading 'App Engine' is displayed with a sub-headline: 'Run your applications on a fully-managed Platform-as-a-Service (PaaS) using built-in services that make you more productive. Just download the SDK and start building immediately.' A prominent blue button labeled 'Start your free trial' is positioned below the text. At the bottom of the main content area, there are links for 'Features', 'Case Studies', 'Pricing Calculator', 'Pricing', and 'Documentation'.

## Features

### Popular languages and frameworks

Write applications in some of the most popular programming languages: Python, Java, PHP and Go. Use existing frameworks such as Django, Flask, Spring and webapp2. Develop locally with

### Focus on your code

Let Google worry about database administration, server configuration, sharding and load balancing. With Traffic Splitting, you can A/B test different live versions of your app. Multitenancy support lets you compartmentalize your application data.

### Multiple storage options

Choose the storage option you need: a traditional MySQL database using Cloud SQL, a schemaless NoSQL datastore, or object storage using Cloud Storage.

<https://cloud.google.com/appengine/>



# Google Cloud Datastore

The screenshot shows the Google Cloud Datastore website. At the top, there's a navigation bar with the Google Cloud Platform logo, a search bar, and links for 'Go to my console' and 'Sign out'. Below this is a secondary navigation bar with links for 'Why Google', 'Products', 'Solutions', 'Pricing', 'Customers', 'Documentation', 'Support', and 'Partners', along with 'Free Trial' and 'Contact Sales' buttons. The main content area features a large background image of server racks. On the left, there's a blue hexagonal icon with a grid pattern. To its right, the text reads 'Cloud Datastore' followed by a description: 'Use a managed, NoSQL, schemaless database for storing non-relational data. Cloud Datastore automatically scales as you need it and supports transactions as well as robust, SQL-like queries.' Below this is a prominent blue button that says 'Start your free trial'. At the bottom of the main content area, there's a horizontal menu with links for 'Features', 'Pricing Calculator', 'Pricing', and 'Documentation'.

## Features

### Schemaless access, with SQL-like querying

No need to worry about data models and migration. Cloud Datastore is a schemaless storage service that allows you to be agile by removing the need to think about the underlying structure of the

### Managed database

Cloud Datastore is fully managed. Google automatically handles sharding and replication in order to provide you with a highly available and consistent database.

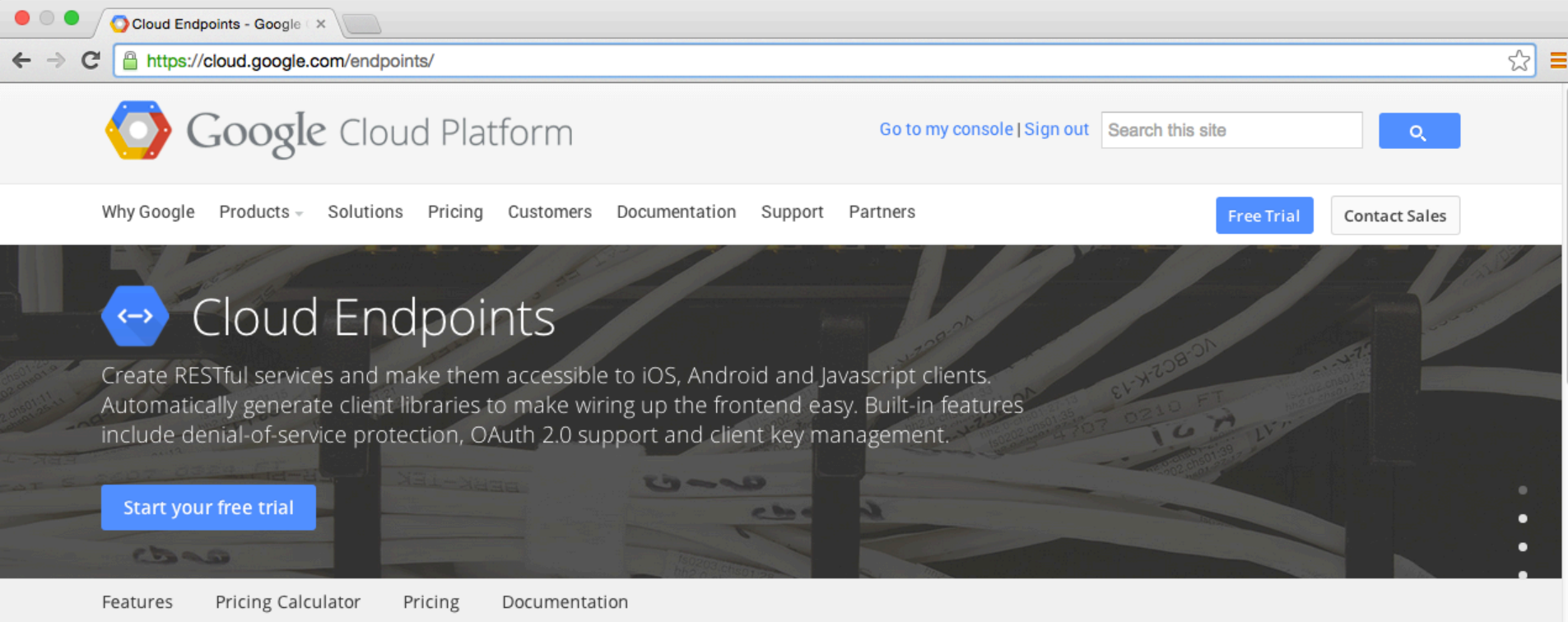
### Autoscale with your users

Cloud Datastore automatically scales depending on your needs. This allows you to focus on building your application and not on worrying about provisioning and load anticipation.

<https://cloud.google.com/datastore/>



# Google Cloud Endpoints



## Features



### One tool, multiple clients

Build client libraries for Android, iOS and web-based clients from one source. Cloud Endpoints wraps your code to build an [API server](#) in just a few steps. Cloud Endpoint API libraries are available in Java, Python, Go and PHP. [Learn more](#)



### Extending App Engine infrastructure

All of the tools and libraries made available in App Engine are now available to your mobile devices. Access Datastore, Cloud Storage and Task Queues using your App Engine backend with no extra

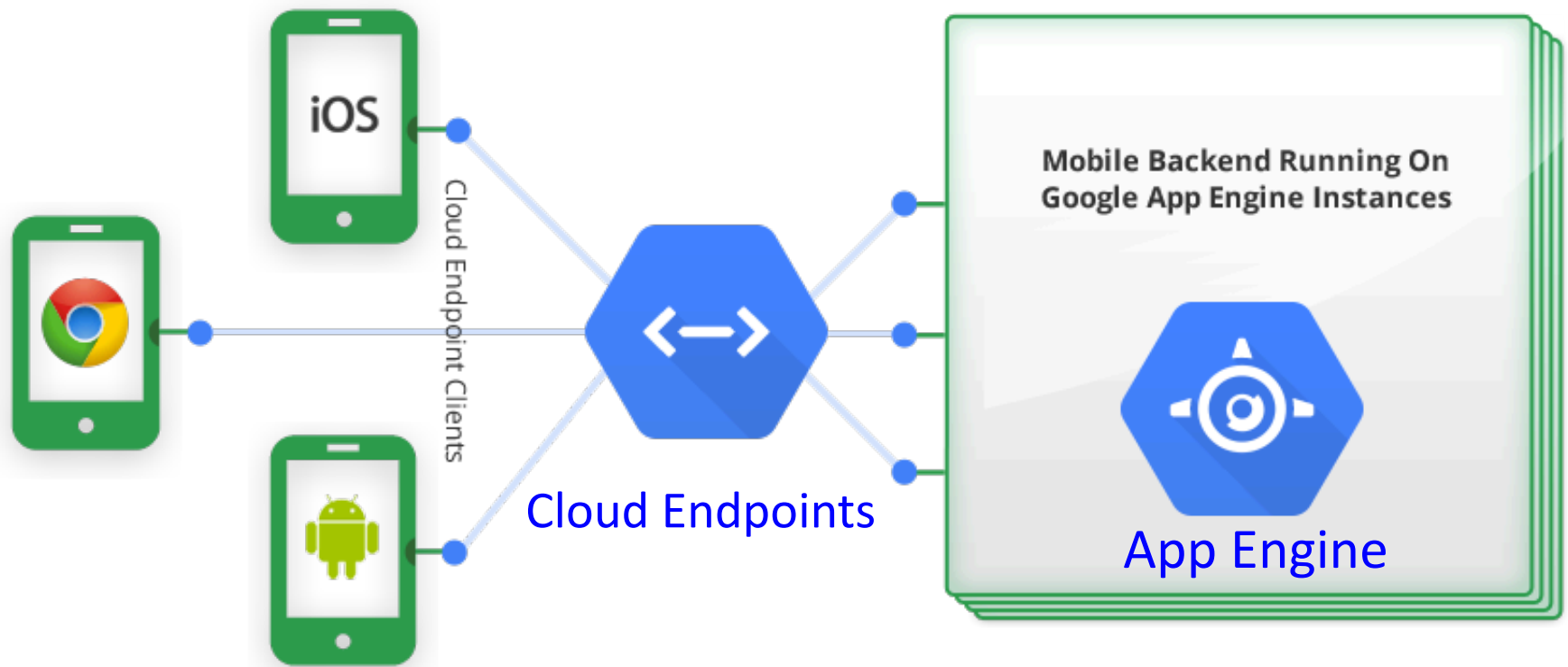


### Low maintenance client-server

Because Cloud Endpoints is backed by App Engine, you have no servers to maintain, no load balancing to worry about and the same quick and painless scaling. Like App Engine and our other Cloud services, you only pay for what you use.

Source: <https://cloud.google.com/endpoints/>

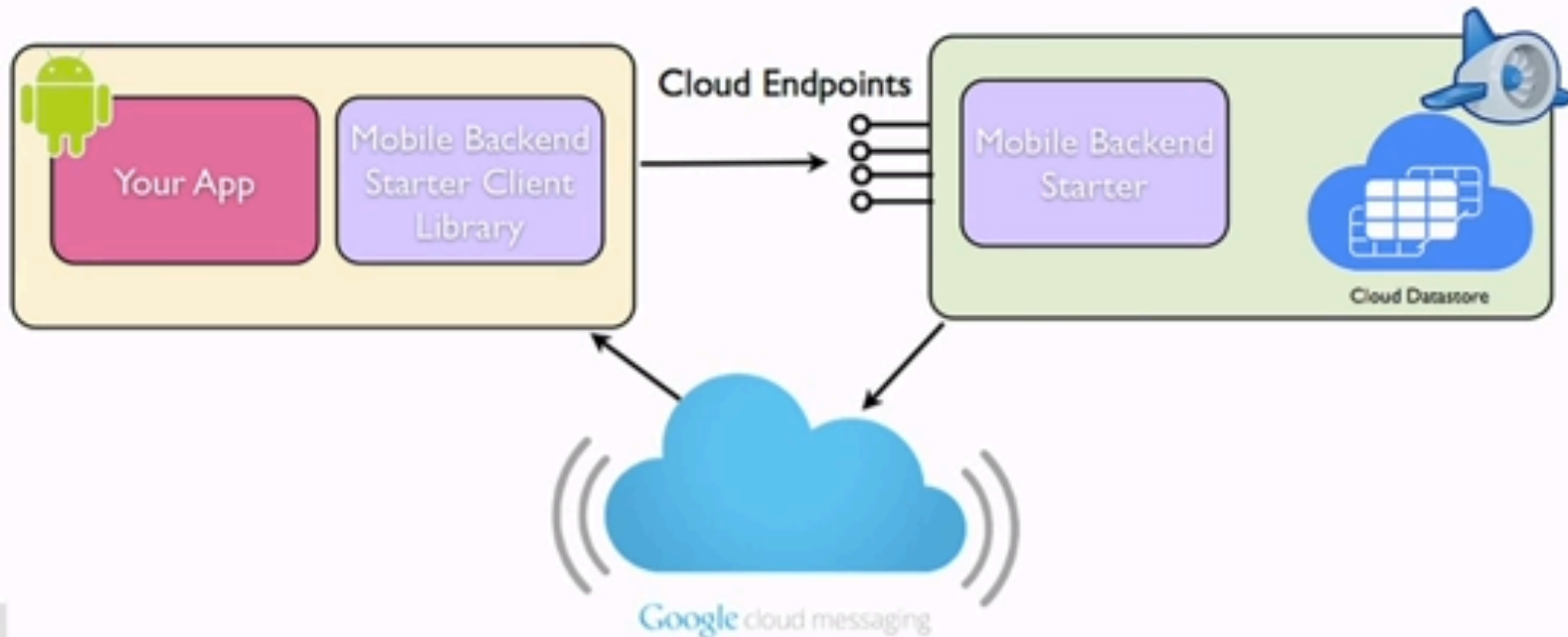
# Mobile Apps Backend on Google App Engine



## Google Cloud Endpoints Architecture

# Mobile App, Google App Engine, Cloud Datastore

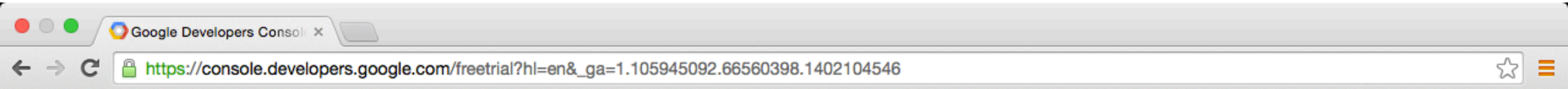
## Mobile Backend Starter







# Try Google Cloud Platform for free



## Try Google Cloud Platform for free

Build on top of the infrastructure that powers Google.  
Sign up for free and get \$300 to spend over 60 days on all Google Cloud Platform services.

Certain terms and conditions apply. [Learn more](#)


**Country**


**Account type**


Business


Individual

**Name and address** ?

 **Get \$300 to kick start your app.**  
Sign up for free and get \$300 to spend on Google Cloud Platform over the next 60 days.

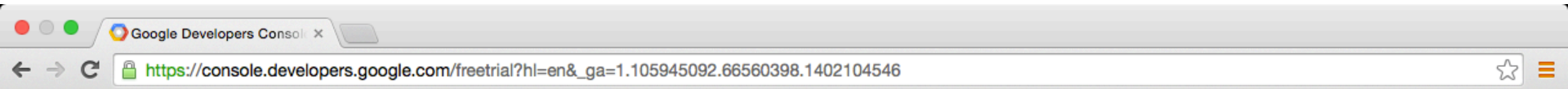
 **Why do you need my billing information?**  
We use your billing information to verify that you're a real person. Don't worry, you will not be billed for the free trial.

 **Do I have to pay when my free trial ends?**  
No. You're under no obligation to buy anything when the free trial ends. If you want to continue to use Google Cloud Platform, just upgrade before your trial runs out.

 **Questions?**  
Check out the [FAQ](#) or [leave us a message](#).




# Try Google Cloud Platform for free



士林區



台北市


 Check out the [FAQ](#) or [leave us a message](#).

Primary contact


**What you pay with**

Credit or debit card

VISA  AMEX 

/   

Credit or debit card address is same as above

**Billing communication language** 

English (United States)

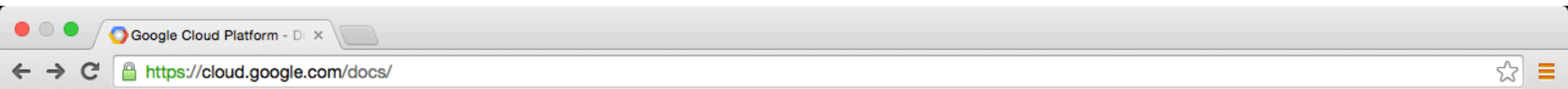
I have read and agree to the Google Cloud Platform Free Trial [Terms of Service](#).





# Google Cloud Platform

<https://cloud.google.com/docs/>



- Balancing Beta
- Network Load
- Balancing
- ▼ **Big Data**
- BigQuery
- Hadoop on Google Cloud Platform
- ▼ **Services**
- Cloud Endpoints
- Cloud Pub/Sub
- Cloud Monitoring
- Prediction API
- Translate API
- ▼ **Management**
- Deployment Manager
- ▼ **Tools**
- Overview
- Cloud SDK
- ▶ Android Studio
- Eclipse Plugin
- Cloud Repositories
- Source Code Tools
- Release Pipelines
- Cloud Debugger
- Cloud Trace
- Cloud Playground
- ▶ Cloud Logging
- ▼ **Click-to-Deploy**

## Get started quickly

If you're ready to get started, try these solutions for getting software up and running on our platform!



### LAMP development stack

LAMP (an acronym for Linux, Apache, MySQL, and PHP) is the archetypal open-source web development stack for many developers, and it runs great on Compute Engine!



### Ruby development stack

Ruby on Rails is one of the most popular frameworks for developing web applications, powering sites like Github, Basecamp and Shopify. Rapidly create new features, easily maintain code, and take advantage of the many open source contributions to Ruby on Rails, running on Compute Engine.



### Quickstart for WordPress

Set up a project, download a zip, change your config file, and deploy -and you'll have a working WordPress project running on Google Cloud Platform, with App Engine as your hosting environment.



### App Engine "Hello World" starter

Start editing a working "Hello World" app right now, in the browser. This gives you a good starting point and a feel for what it's like editing a working App Engine application.

## Documentation

Use the following section or the left-hand navigation to access the various sets of documentation that cover Google Cloud Platform. Choose from [computing and hosting](#), [storage](#), [big data](#), [management](#), [services](#), and [developer tools](#).

## Computing and hosting



App Engine

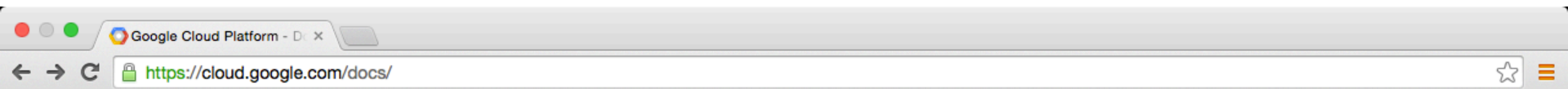


Compute Engine



# Google Cloud Platform

<https://cloud.google.com/docs/>



- Cloud Trace
- Cloud Playground
- ▶ Cloud Logging
- ▼ Click-to-Deploy
  - Cassandra
  - GitLab
  - LAMP Stack
  - MEAN Stack
  - MongoDB
  - RabbitMQ
  - Redis
  - Ruby on Rails
- ▶ Architecture Diagrams
- ▼ Solutions
  - Mobile
  - Hadoop
  - Gaming
  - MongoDB
  - RabbitMQ
  - Redis
  - Cassandra
- ▼ Sample Code and Videos
  - Videos
  - Sample Applications
- Support Center
- Terms of Service

## Computing and hosting



### App Engine

App Engine is Google's Platform-as-a-Service (PaaS). Develop your application easily using built-in services that make you more productive. Deploy to a fully-managed platform and let Google carry the pager. Just download the SDK and start building immediately for free with no credit card required.

[Overview](#)

[Get Started](#)

Tutorials: [Java](#), [PHP](#), [Python](#), [Go](#)

[Documentation](#)



### Compute Engine

Compute Engine is Google's Infrastructure-as-a-Service (IaaS). Run large-scale workloads on virtual machines hosted on Google's infrastructure. Choose a VM that fits your needs and gain the performance and consistency of Google's worldwide fiber network. With per-minute billing, you pay only for what you use.

[Overview](#)

[Get Started](#)

[Tutorial](#)

[Documentation](#)

## Storage



### Cloud SQL

Store and manage data using a relational MySQL database. Google handles replication, patch management and database management to ensure availability and performance, and you can even have your instance automatically co-locate with your deployed applications.

[Overview](#)

[Tutorial](#)



### Cloud Storage

Use a durable and highly available object storage service. With global edge-caching, your users have fast access to your app's data from any location. Google manages versioning, guarantees a strong SLA and provides a simple API that allows you to manage your data programmatically.

[Overview](#)

[Tutorial](#)



### Cloud Datastore

Cloud Datastore provides a managed, NoSQL, schemaless database for storing non-relational data. Cloud Datastore automatically scales as you need it and supports transactions as well as robust, SQL-like queries.

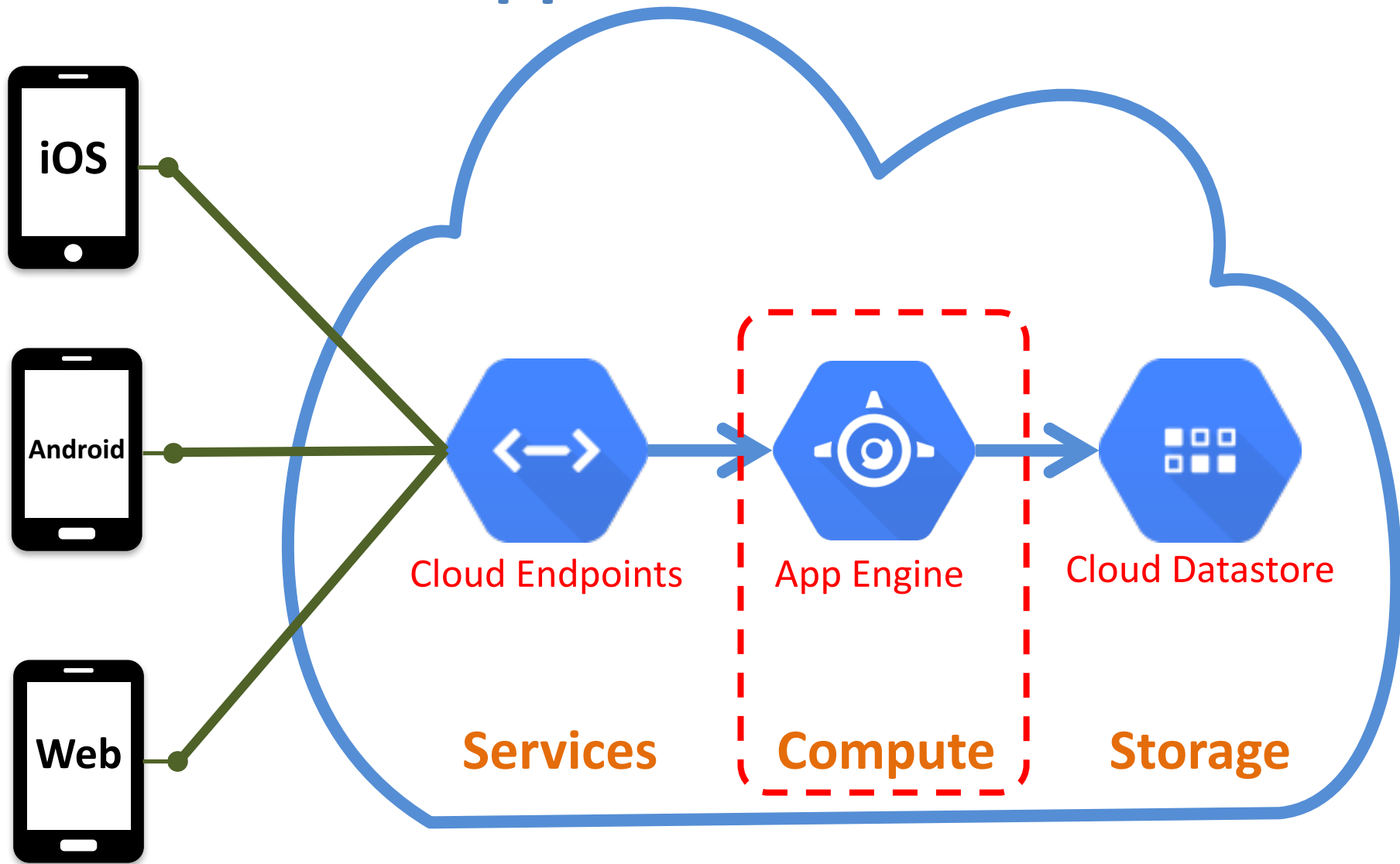
[Overview](#)

[Tutorial](#)

[Documentation](#)



# Mobile App Backend Services





# Google App Engine

## App Engine "Hello World" starter

<https://console.developers.google.com/start/appengine>

### **Deploy your first app in five minutes**

- Start editing a working "Hello World" app right now, in the browser.
- This gives you a good starting point and a feel for what it's like editing a working App Engine application.

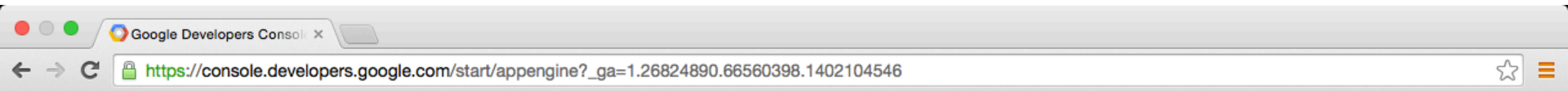


# Try Google App Engine Now

1. NAME YOUR PROJECT
2. SELECT YOUR LANGUAGE
3. EXPLORE THE STARTER CODE
4. INSTALL GOOGLE CLOUD SDK
5. RUN YOUR APP LOCALLY
6. CREATE YOUR PROJECT AND DEPLOY



# Google App Engine



- 1. NAME YOUR PROJECT
- 2. SELECT YOUR LANGUAGE
- 3. EXPLORE THE STARTER CODE
- 4. INSTALL GOOGLE CLOUD SDK
- 5. RUN YOUR APP LOCALLY
- 6. CREATE YOUR PROJECT AND DEPLOY

## Try Google App Engine Now

Creating an App Engine app is easy, and it's free to start. Upload your app and share it with users right away, at no charge and with no commitment required.

### 1 NAME YOUR PROJECT

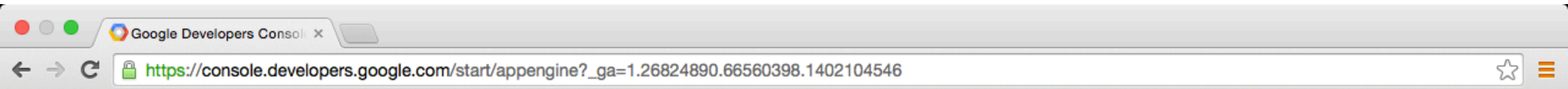
You use your project to manage all of the Google Cloud Platform resources for your app, including deployment, access control, billing, and services. You can change your project name later.

### 2 SELECT YOUR LANGUAGE

**Python** **Java**



# Google App Engine



## 2 SELECT YOUR LANGUAGE

- 1. NAME YOUR PROJECT
- 2. SELECT YOUR LANGUAGE
- 3. EXPLORE THE STARTER CODE
- 4. INSTALL GOOGLE CLOUD SDK
- 5. RUN YOUR APP LOCALLY
- 6. CREATE YOUR PROJECT AND DEPLOY



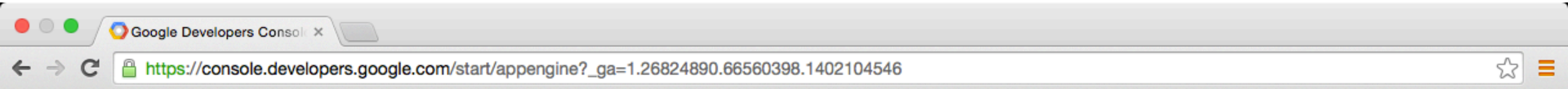
## 3 EXPLORE THE STARTER CODE

Browse the starter code and see the app running below.

```
HELLO WORLD - JAVA
build.xml      1 package myapp;
pom.xml       2
src/main/java/myapp/D 3 import java.io.IOException;
src/main/webapp/WEB-INF 4 import javax.servlet.http.*;
src/main/webapp/WEB-INF 5
src/main/webapp/WEB-INF 6 public class DemoServlet extends HttpServlet {
src/main/webapp/WEB-INF 7     @Override
src/main/webapp/WEB-INF 8     public void doGet(HttpServletRequest req, HttpServletResponse resp)
src/main/webapp/WEB-INF 9         throws IOException {
src/main/webapp/WEB-INF 10        resp.setContentType("text/plain");
```



# Google App Engine



## 3 EXPLORE THE STARTER CODE

Browse the starter code and see the app running below.

- 1. NAME YOUR PROJECT
- 2. SELECT YOUR LANGUAGE
- 3. EXPLORE THE STARTER CODE
- 4. INSTALL GOOGLE CLOUD SDK
- 5. RUN YOUR APP LOCALLY
- 6. CREATE YOUR PROJECT AND DEPLOY

### HELLO WORLD - JAVA

```
build.xml 1 package myapp;
pom.xml   2
src/main/java/myapp/D 3 import java.io.IOException;
4 import javax.servlet.http.*;
5
src/main/webapp/WEB-INF 6 public class DemoServlet extends HttpServlet {
7     @Override
src/main/webapp/WEB-INF 8     public void doGet(HttpServletRequest req, HttpServletResponse resp)
9         throws IOException {
src/main/webapp/WEB-INF 10         resp.setContentType("text/plain");
11         resp.getWriter().println("{ \"name\": \"World\" }");
12     }
13 }
14
```



Hello, World





# Google App Engine

## HELLO WORLD - JAVA

```
build.xml      1 package myapp;
pom.xml       2
src/main/java/myapp/D 3 import java.io.IOException;
src/main/webapp/WEB-INF 4 import javax.servlet.http.*;
src/main/webapp/WEB-INF 5
src/main/webapp/WEB-INF 6 public class DemoServlet extends HttpServlet {
src/main/webapp/WEB-INF 7     @Override
src/main/webapp/WEB-INF 8     public void doGet(HttpServletRequest req, HttpServletResponse resp)
src/main/webapp/WEB-INF 9         throws IOException {
src/main/webapp/index.t 10         resp.setContentType("text/plain");
11         resp.getWriter().println("{ \"name\": \"World\" }");
12     }
13 }
14
```

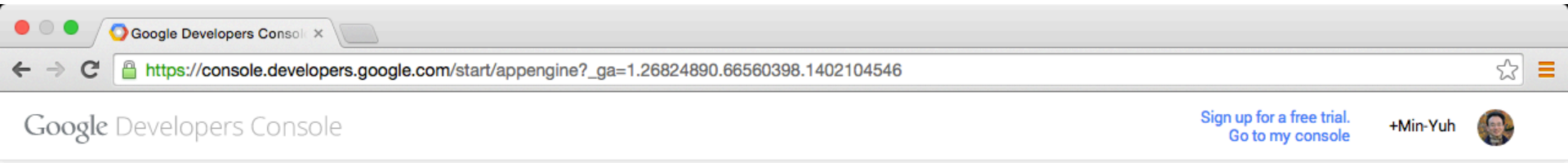


Hello, World

[Download this code](#)



# Google App Engine



## 4 INSTALL GOOGLE CLOUD SDK

Mac OS X/Linux

1. Download and install Google Cloud SDK by running the following command in your shell or Terminal:

```
curl https://sdk.cloud.google.com/ | bash
```

Follow the prompts to install the **Java** App Engine package.

2. Restart your shell or Terminal to pick up environment changes.
3. Sign in to Google Cloud Platform using this command:

```
gcloud auth login
```

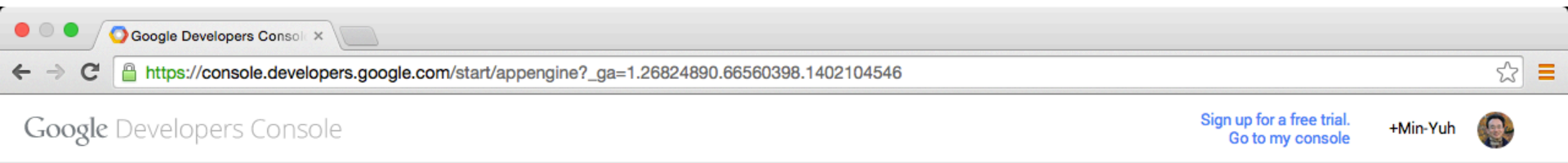
4. Install the App Engine package for Java using this command:

```
gcloud components update gae-java
```

## 5 RUN YOUR APP LOCALLY



# Google App Engine



## 5 RUN YOUR APP LOCALLY

**Note:** App Engine supports Java 7. Make sure you have the [Java 7 JDK](#) installed.

1. Download [appengine-try-java.zip](#) and unpack it. This creates your project directory, including `src/` and `war/` subdirectories.
2. Download and install [Apache Maven](#) version 3.0 or later. The starter app includes an Apache Maven build file.
3. Build and run the sample locally using the following commands:

```
cd appengine-try-java
mvn package
dev_appserver.sh target/appengine-try-java-1.0
```

You can stop the server by pressing Ctrl-C in the command window.

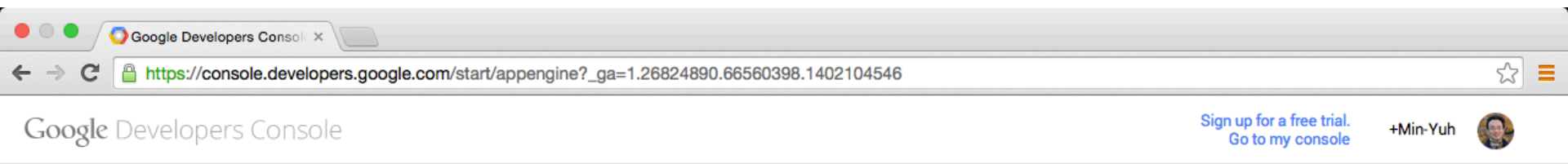
4. Visit the locally running app in your browser: <http://localhost:8080>

## 6 CREATE YOUR PROJECT AND DEPLOY

Now that you've seen your app running on your local machine, you're ready to create and deploy your project, HelloWorldGoogleAppEngine.



# Google App Engine



## 5 RUN YOUR APP LOCALLY

**Note:** App Engine supports Java 7. Make sure you have the [Java 7 JDK](#) installed.

1. Download [appengine-try-java.zip](#) and unpack it. This creates your project directory, including `src/` and `war/` subdirectories.
2. Download and install [Apache Maven](#) version 3.0 or later. The starter app includes an Apache Maven build file.
3. Build and run the sample locally using the following commands:

```
cd appengine-try-java
mvn package
dev_appserver.sh target/appengine-try-java-1.0
```

You can stop the server by pressing Ctrl-C in the command window.

4. Visit the locally running app in your browser: <http://localhost:8080>

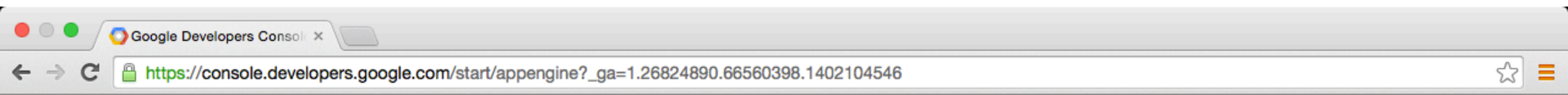
## 6 CREATE YOUR PROJECT AND DEPLOY

Now that you've seen your app running on your local machine, you're ready to create and deploy your project, HelloWorldGoogleAppEngine.

1. NAME YOUR PROJECT
2. SELECT YOUR LANGUAGE
3. EXPLORE THE STARTER CODE
4. INSTALL GOOGLE CLOUD SDK
5. RUN YOUR APP LOCALLY
6. CREATE YOUR PROJECT AND DEPLOY



# Google App Engine



- 1. NAME YOUR PROJECT
- 2. SELECT YOUR LANGUAGE
- 3. EXPLORE THE STARTER CODE
- 4. INSTALL GOOGLE CLOUD SDK
- 5. RUN YOUR APP LOCALLY
- 6. CREATE YOUR PROJECT AND DEPLOY

```
mvn package
dev_appserver.sh target/appengine-try-java-1.0
```

You can stop the server by pressing Ctrl-C in the command window.

- 4. Visit the locally running app in your browser: <http://localhost:8080>

## 6 CREATE YOUR PROJECT AND DEPLOY

Congratulations! Your project is ready. Your unique project ID is **psychic-outcome-783**.

Deploy your app from your local dev environment using this command:

```
appcfg.sh -A psychic-outcome-783 update target/appengine-try-java-1.0
```

After deploying your app, you can visit it with your browser at this URL:

[psychic-outcome-783.appspot.com](http://psychic-outcome-783.appspot.com)

That's it! You're running on Google App Engine. Go to your project dashboard to see how your app is performing.

[View your project dashboard](#)

**Activities (Idle)** G \_ X

Create Project: HelloWorldGoogleAppEngine ✔

[See all activity](#)



# Google App Engine



## Build an App Engine Application using Python

### Creating a Guestbook

1. Download the App Engine SDK
2. Explaining the webapp2 Framework
3. Using the Users Service
4. Handling Forms with webapp2
5. Using the Datastore
6. Using Templates
7. Using Static Files
8. Uploading Your Application



## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

##### 1. Download the App Engine SDK

2. Explaining the webapp2 Framework

3. Using the Users Service

4. Handling Forms with webapp2

5. Using the Datastore

6. Using Templates

7. Using Static Files

8. Uploading Your Application

Sample Applications

## Introduction

Welcome to Google App Engine! Creating an App Engine application is easy, only takes a few minutes, and it's free to start.

Google App Engine applications can be written in the Python 2.7, Java, Go or PHP programming languages. This tutorial covers **Python 2.7**. If you would prefer to use Java, Go or PHP to build your applications, see the [Java](#), [Go](#) or [PHP](#) guides.

In this tutorial, you will learn how to:

- build an App Engine application using Python
- use the [webapp2](#) web application framework
- use the App Engine datastore with the Python modeling API
- integrate an App Engine application with Google Accounts for user authentication
- use [Jinja2](#) templates with your app
- upload your app to App Engine

By the end of the tutorial, you will have implemented a working application, a simple guest book that lets users post messages to a public message board.

## Get set up

Before we continue, you will need to download the [App Engine Python SDK](#), which includes a web server application that simulates the



## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

##### 1. Download the App Engine SDK

2. Explaining the webapp2 Framework

3. Using the Users Service

4. Handling Forms with webapp2

5. Using the Datastore

6. Using Templates

7. Using Static Files

8. Uploading Your Application

Sample Applications

### Overviews

How App Engine Works

Developing Python Apps on App Engine

Managing and Configuring Your App

## Introduction

Welcome to Google App Engine! Creating an App Engine application is easy, only takes a few minutes, and it's free to start.

Google App Engine applications can be written in the Python 2.7, Java, Go or PHP programming languages. This tutorial covers **Python 2.7**. If you would prefer to use Java, Go or PHP to build your applications, see the [Java](#), [Go](#) or [PHP](#) guides.

In this tutorial, you will learn how to:

- build an App Engine application using Python
- use the [webapp2](#) web application framework
- use the App Engine datastore with the Python modeling API
- integrate an App Engine application with Google Accounts for user authentication
- use [Jinja2](#) templates with your app
- upload your app to App Engine

By the end of the tutorial, you will have implemented a working application, a simple guest book that lets users post messages to a public message board.

## Get set up

Before we continue, you will need to download the [App Engine Python SDK](#), which includes a web server application that simulates the App Engine environment, and tools to deploy your application to the App Engine production environment. Follow the directions for your operating system, then come back here so we can get going!

[Explaining the webapp2 Framework! >>](#)





## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

1. Download the App Engine SDK

2. **Explaining the webapp2 Framework**

3. Using the Users Service

4. Handling Forms with webapp2

5. Using the Datastore

6. Using Templates

7. Using Static Files

8. Uploading Your Application

### Sample Applications

#### Overviews

How App Engine Works

Developing Python Apps on App Engine

Managing and Configuring Your App

## Explaining the webapp2 Framework

The Web Server Gateway Interface (WSGI) standard is simple, but it would be cumbersome to write all of the code that uses it by hand. Web application frameworks handle these details for you, so you can focus your development efforts on your application's features. Google App Engine supports any framework written in pure Python that speaks WSGI, including [Django](#), [CherryPy](#), [Pylons](#), [web.py](#), and [web2py](#). You can bundle a framework of your choosing with your application code by copying its code into your application directory.

App Engine includes a simple web application framework, called `webapp2`. The `webapp2` framework is already installed in the App Engine environment and in the [App Engine Python SDK](#), so you do not need to bundle it with your application code to use it. We will use `webapp2` for the rest of this tutorial.

## Hello, webapp2!

A `webapp2` application has two parts:

- one or more `RequestHandler` classes that process requests and build responses
- a `WSGIApplication` instance that routes incoming requests to handlers based on the URL

Let's take a look at [Hello World!](#):

helloworld.py

[View on GitHub](#)

```
import webapp2

class MainPage(webapp2.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.write('Hello, World!')

app = webapp2.WSGIApplication([
```



## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

1. Download the App Engine SDK
2. Explaining the webapp2 Framework
- 3. Using the Users Service**
4. Handling Forms with webapp2
5. Using the Datastore
6. Using Templates
7. Using Static Files
8. Uploading Your Application

Sample Applications

### Overviews

- How App Engine Works
- Developing Python Apps on App Engine
- Managing and Configuring Your App

## Using the Users Service

Google App Engine provides several useful services based on Google infrastructure, accessible by applications using libraries included with the [App Engine Python SDK](#). One such service is the Users service, which lets your application integrate with Google user accounts. With the Users service, your users can use the Google accounts they already have to sign in to your application.

Let's use the Users service to personalize this application's greeting..

## Using Users

We're going to build on the [Hello, World! application](#). You can clone the code for this step from [this branch on GitHub](#), or if you already have the original Hello, World! app, replace its contents with the following:

helloworld.py

[View on GitHub](#)

```
from google.appengine.api import users

import webapp2

class MainPage(webapp2.RequestHandler):

    def get(self):
        # Checks for active Google account session
        user = users.get_current_user()

        if user:
            self.response.headers['Content-Type'] = 'text/html; charset=utf-8'
            self.response.write('Hello, ' + user.nickname())
```



## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

1. Download the App Engine SDK
2. Explaining the webapp2 Framework
3. Using the Users Service
- 4. Handling Forms with webapp2**
5. Using the Datastore
6. Using Templates
7. Using Static Files
8. Uploading Your Application

Sample Applications

### Overviews

- How App Engine Works
- Developing Python Apps on App Engine
- Managing and Configuring Your App

## Handling Forms with webapp2

If we want users to be able to post their own greetings, we need a way to process information submitted by the user with a web form. The `webapp2` framework makes processing form data easy.

### From Hello World to Guestbook

In order to prepare the Hello World app we've created thus far, please make the following changes:

- Rename the top level `helloworld` directory to `guestbook`
- Rename `helloworld.py` to `guestbook.py`
- Replace the `handlers` section of `app.yaml` with:

```
app.yaml View on GitHub
handlers:
- url: /*
  script: guestbook.app
```

Restart the development server using the new `guestbook` directory.

### Handling Web Forms With webapp2

Declare that you are using `webapp2` by adding this `libraries` section to your `app.yaml`:

```
app.yaml View on GitHub
```



## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

1. Download the App Engine SDK
2. Explaining the webapp2 Framework
3. Using the Users Service
4. Handling Forms with webapp2
- 5. Using the Datastore**
6. Using Templates
7. Using Static Files
8. Uploading Your Application

Sample Applications

### Overviews

- How App Engine Works
- Developing Python Apps on App Engine
- Managing and Configuring Your App
- Storing Data

## Using the Datastore

Storing data in a scalable web application can be tricky. A user could be interacting with any of dozens of web servers at a given time, and the user's next request could go to a different web server than the previous request. All web servers need to be interacting with data that is also spread out across dozens of machines, possibly in different locations around the world.

With Google App Engine, you don't have to worry about any of that. App Engine's infrastructure takes care of all of the distribution, replication, and load balancing of data behind a simple API—and you get a powerful query engine and transactions as well.

App Engine's data repository, the *High Replication Datastore (HRD)*, uses the [Paxos algorithm](#) to replicate data across multiple datacenters. Data is written to the Datastore in objects known as *entities*. Each entity has a *key* that uniquely identifies it. An entity can optionally designate another entity as its *parent*; the first entity is a *child* of the parent entity. The entities in the Datastore thus form a hierarchically-structured space similar to the directory structure of a file system. An entity's parent, parent's parent, and so on recursively, are its *ancestors*; its children, children's children, and so on, are its *descendants*. An entity without a parent is a *root entity*.

The Datastore is extremely resilient in the face of catastrophic failure, but its consistency guarantees may differ from what you're familiar with. Entities descended from a common ancestor are said to belong to the same *entity group*; the common ancestor's key is the group's *parent key*, which serves to identify the entire group. Queries over a single entity group, called *ancestor queries*, refer to the parent key instead of a specific entity's key. Entity groups are a unit of both consistency and transactionality: whereas queries over multiple entity groups may return stale, *eventually consistent* results, those limited to a single entity group always return up-to-date, *strongly consistent* results.

The sample application in this guide organizes related entities into entity groups, and uses ancestor queries on those entity groups to return strongly consistent results. In the example code comments, we highlight some ways this approach might affect the design of your application. For more detailed information, see [Structuring Data for Strong Consistency](#).

## A Complete Example Using the Datastore

Here is a new version of `guestbook/guestbook.py` that creates a page footer that stores greetings in the Datastore. The rest of this page discusses excerpts from this larger example, organized under the topics of storing the greetings and retrieving them.



## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

1. Download the App Engine SDK
2. Explaining the webapp2 Framework
3. Using the Users Service
4. Handling Forms with webapp2
5. Using the Datastore
- 6. Using Templates**
7. Using Static Files
8. Uploading Your Application

Sample Applications

### Overviews

- How App Engine Works
- Developing Python Apps on App Engine
- Managing and Configuring Your App

## Using Templates

HTML embedded in code is messy and difficult to maintain. It's better to use a templating system, where the HTML is kept in a separate file with special syntax to indicate where the data from the application appears. There are many templating systems for Python: [EZT](#), [Cheetah](#), [ClearSilver](#), [Quixote](#), [Django](#), and [Jinja2](#) are just a few. You can use your template engine of choice by bundling it with your application code.

For your convenience, App Engine includes the Django and Jinja2 templating engines.

## Using Jinja2 Templates

First modify the `libraries` section at the bottom of `guestbook/app.yaml`:

```
app.yaml
libraries:
- name: webapp2
  version: latest
- name: jinja2
  version: latest
```

[View on GitHub](#)

This configuration makes the newest supported version of Jinja2 available to your application. To avoid possible compatibility issues, serious applications should use an [actual version number](#) rather than `latest`.

Now modify the statements at the top of `guestbook/guestbook.py`:

```
guestbook.py
import os
```

[View on GitHub](#)



## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

1. Download the App Engine SDK
2. Explaining the webapp2 Framework
3. Using the Users Service
4. Handling Forms with webapp2
5. Using the Datastore
6. Using Templates
- 7. Using Static Files**
8. Uploading Your Application

Sample Applications

### Overviews

- How App Engine Works
- Developing Python Apps on App Engine
- Managing and Configuring Your App

## Using Static Files

Unlike a traditional web hosting environment, Google App Engine does not serve files directly out of your application's source directory unless configured to do so. We named our template file `index.html`, but this does not automatically make the file available at the URL `/index.html`.

But there are many cases where you want to serve static files directly to the web browser. Images, CSS stylesheets, JavaScript code, movies and Flash animations are all typically stored with a web application and served directly to the browser. App Engine can serve specific files directly without you having to code your own handler.

## Using Static Files

Edit `guestbook/app.yaml` and replace its contents with the following:

```
app.yaml View on GitHub
version: 1
runtime: python27
api_version: 1
threadsafe: true

# Handlers match in order, put above the default handler.
handlers:
- url: /stylesheets
  static_dir: stylesheets

- url: /.*
  script: guestbook.app
```





## Python

App Engine Home

### Training and Sample Applications

Hello, World! in 5 minutes

#### Further Training

##### Creating a Guestbook

1. Download the App Engine SDK
2. Explaining the webapp2 Framework
3. Using the Users Service
4. Handling Forms with webapp2
5. Using the Datastore
6. Using Templates
7. Using Static Files

##### 8. Uploading Your Application

Sample Applications

### Overviews

- How App Engine Works
- Developing Python Apps on App Engine
- Managing and Configuring Your App

## Uploading Your Application

To upload your application:

1. Sign in to App Engine using your Google account. If you do not have a Google account, you can [create a Google account](#) with an email address and password.
2. If you haven't already done so, create a project for your App Engine app as follows:
  - a. Visit the [Google Cloud Platform Console](#) and click **Create Project**.
  - b. Supply the desired project name in the New Project form. It doesn't have to match your app name, but using the same name as your app might make administration easier.
  - c. Accept the generated project ID or supply your own ID. *This project ID is used as the App Engine application ID.* Note that this ID can only be used once: if you subsequently delete your project, you won't be able to re-use the ID in a new project.

**Note:** You can specify that your new application should reside in the European Union, rather than the United States. Hosting applications in the European Union is especially useful if your application's users are closer to Europe than to the United States. There is less network latency and the End User Content will be stored at rest in the European Union. You must specify this location when you register the application; you cannot change it later. Click **Show Advanced Options** in the Create Project section to select a location option, either United States or European Union.

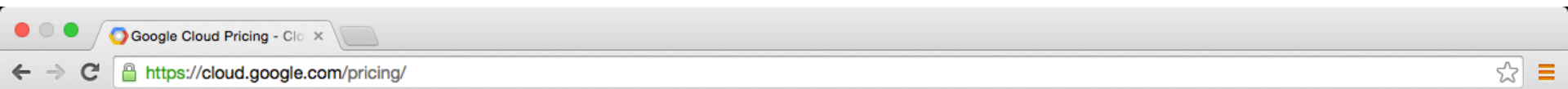
3. Note the application ID (project ID) you created above.
4. Upload your finished application to Google App Engine by invoking the following command. This opens a browser window for you to sign in using your Google account. You'll be providing the project ID as the argument for **-A**.

```
appcfg.py -A <YOUR_PROJECT_ID> update guestbook/
```

5. The [Datastore Indexes](#) may take some time to generate before your application is available. You will receive a `NeedIndexError` when accessing your app if the indexes are still in the process of being generated. This is a transient error for the example, so try a little later if at first you receive this exception.
6. Your app is now deployed and ready for users!



# Google Cloud Platform



Google Cloud Platform

[Go to my console](#) | [Sign out](#)

Search this site



[Why Google](#) [Products](#) [Solutions](#) [Pricing](#) [Customers](#) [Documentation](#) [Support](#) [Partners](#)

[Free Trial](#)

[Contact Sales](#)

## Pricing overview

Run your application using the same technology and tools used at Google. Cloud Platform provides the building blocks so you can develop quickly, using the services that you need.

### Pricing calculator

Want to get a sense of what running in the cloud will cost? The pricing calculator provides a quick and easy way to estimate what your usage will look like.

[Get a custom quote](#)

### Pricing philosophy

We believe that pricing should be as simple as possible and the burden of getting the best deal should be on us not you. That's why we have [sustained-use discounts](#), pay by the minute pricing, and believe in Moore's law in the cloud.

[See philosophy](#)

### Billing questions

Sometimes you just need to get questions about your bill answered. Take a look at our frequently asked questions and if you don't get an answer contact us.

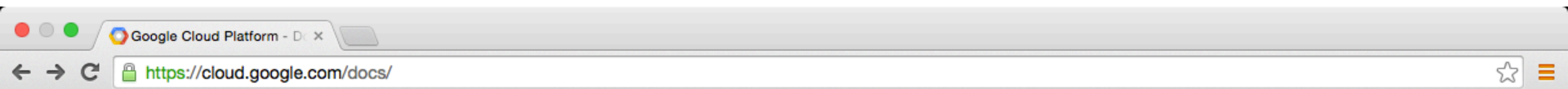
[See billing faqs](#)





# Google Cloud Platform

<https://cloud.google.com/docs/>



[Go to my console](#) | [Sign out](#)

Search this site



- Why Google
- Products ▾
- Solutions
- Pricing
- Customers
- Documentation
- Support
- Partners

Free Trial

Contact Sales

Products > Documentation

## Documentation g+1 878

[Report documentation issue](#)

### ▼ Compute

- App Engine
- Compute Engine
- Container Engine

### ▼ Storage

- Cloud SQL
- Cloud Storage
- Cloud Datastore

### ▼ Networking

- Cloud DNS
- Carrier Interconnect
- Direct Peering
- HTTP Load Balancing Beta
- Network Load Balancing

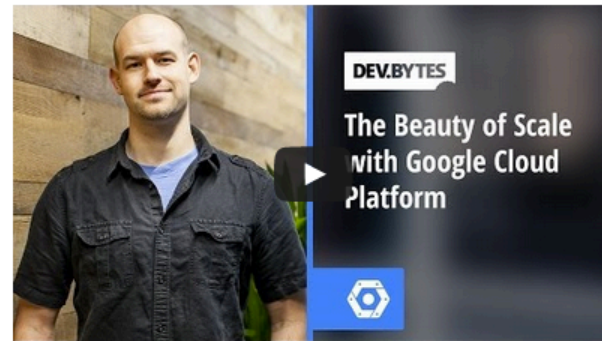
### ▼ Big Data

- BigQuery
- Hadoop on Google

## Google Cloud Platform

With Google Cloud Platform, developers can build, test and deploy applications on Google's highly-scalable and reliable infrastructure for your web, mobile and backend solutions.

Focus on writing code, not on infrastructure, and use the same infrastructure that Google uses for your application, computing and big data needs.



## Get started quickly

If you're ready to get started, try these solutions for getting software up and running on our platform!



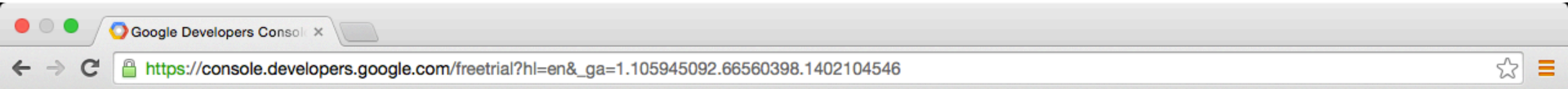
LAMP development stack



Ruby development stack



# Try Google Cloud Platform for free



## Try Google Cloud Platform for free

Build on top of the infrastructure that powers Google.  
Sign up for free and get \$300 to spend over 60 days on all Google Cloud Platform services.

Certain terms and conditions apply. [Learn more](#)

**Country**

**Account type**

Business

Individual

**Name and address** ?



### Get \$300 to kick start your app.

Sign up for free and get \$300 to spend on Google Cloud Platform over the next 60 days.



### Why do you need my billing information?

We use your billing information to verify that you're a real person. Don't worry, you will not be billed for the free trial.



### Do I have to pay when my free trial ends?

No. You're under no obligation to buy anything when the free trial ends. If you want to continue to use Google Cloud Platform, just upgrade before your trial runs out.

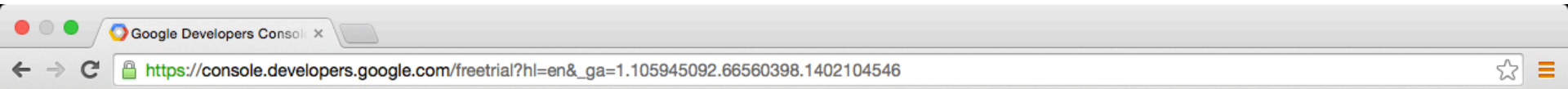


### Questions?

Check out the [FAQ](#) or [leave us a message](#).




# Try Google Cloud Platform for free



士林區



台北市


 Check out the [FAQ](#) or [leave us a message](#).

Primary contact


**What you pay with**

Credit or debit card

VISA  AMEX 

/   

Credit or debit card address is same as above

**Billing communication language** 

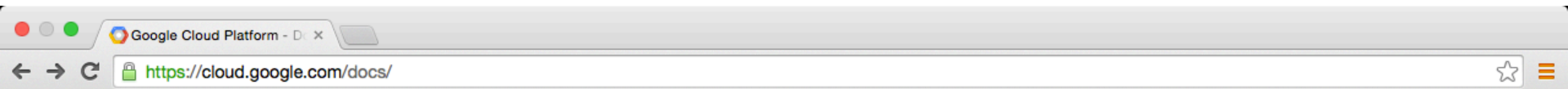
English (United States)

I have read and agree to the Google Cloud Platform Free Trial [Terms of Service](#).



# Google Cloud Platform

<https://cloud.google.com/docs/>



- Balancing Beta
- Network Load
- Balancing
- ▼ **Big Data**
- BigQuery
- Hadoop on Google Cloud Platform
- ▼ **Services**
- Cloud Endpoints
- Cloud Pub/Sub
- Cloud Monitoring
- Prediction API
- Translate API
- ▼ **Management**
- Deployment Manager
- ▼ **Tools**
- Overview
- Cloud SDK
- ▶ Android Studio
- Eclipse Plugin
- Cloud Repositories
- Source Code Tools
- Release Pipelines
- Cloud Debugger
- Cloud Trace
- Cloud Playground
- ▶ Cloud Logging
- ▼ **Click-to-Deploy**

## Get started quickly

If you're ready to get started, try these solutions for getting software up and running on our platform!



### LAMP development stack

LAMP (an acronym for Linux, Apache, MySQL, and PHP) is the archetypal open-source web development stack for many developers, and it runs great on Compute Engine!



### Ruby development stack

Ruby on Rails is one of the most popular frameworks for developing web applications, powering sites like Github, Basecamp and Shopify. Rapidly create new features, easily maintain code, and take advantage of the many open source contributions to Ruby on Rails, running on Compute Engine.



### Quickstart for WordPress

Set up a project, download a zip, change your config file, and deploy -and you'll have a working WordPress project running on Google Cloud Platform, with App Engine as your hosting environment.



### App Engine "Hello World" starter

Start editing a working "Hello World" app right now, in the browser. This gives you a good starting point and a feel for what it's like editing a working App Engine application.

## Documentation

Use the following section or the left-hand navigation to access the various sets of documentation that cover Google Cloud Platform. Choose from [computing and hosting](#), [storage](#), [big data](#), [management](#), [services](#), and [developer tools](#).

## Computing and hosting



App Engine



Compute Engine



# Google App Engine

## App Engine "Hello World" starter

<https://console.developers.google.com/start/appengine>

### **Deploy your first app in five minutes**

- Start editing a working "Hello World" app right now, in the browser.
- This gives you a good starting point and a feel for what it's like editing a working App Engine application.

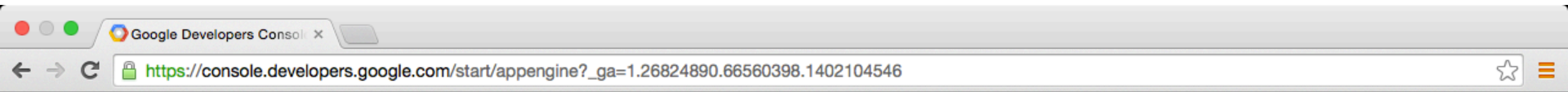


# Try Google App Engine Now

1. NAME YOUR PROJECT
2. SELECT YOUR LANGUAGE
3. EXPLORE THE STARTER CODE
4. INSTALL GOOGLE CLOUD SDK
5. RUN YOUR APP LOCALLY
6. CREATE YOUR PROJECT AND DEPLOY



# Google App Engine



- 1. NAME YOUR PROJECT
- 2. SELECT YOUR LANGUAGE
- 3. EXPLORE THE STARTER CODE
- 4. INSTALL GOOGLE CLOUD SDK
- 5. RUN YOUR APP LOCALLY
- 6. CREATE YOUR PROJECT AND DEPLOY

## Try Google App Engine Now

Creating an App Engine app is easy, and it's free to start. Upload your app and share it with users right away, at no charge and with no commitment required.

### 1 NAME YOUR PROJECT

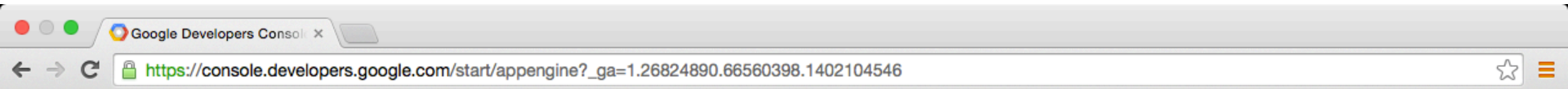
You use your project to manage all of the Google Cloud Platform resources for your app, including deployment, access control, billing, and services. You can change your project name later.

### 2 SELECT YOUR LANGUAGE

**Python** **Java**



# Google App Engine



Google Developers Console

Sign up for a free trial.  
Go to my console

+Min-Yuh



## 2 SELECT YOUR LANGUAGE



1. NAME YOUR PROJECT
2. SELECT YOUR LANGUAGE
3. EXPLORE THE STARTER CODE
4. INSTALL GOOGLE CLOUD SDK
5. RUN YOUR APP LOCALLY
6. CREATE YOUR PROJECT AND DEPLOY

## 3 EXPLORE THE STARTER CODE

Browse the starter code and see the app running below.

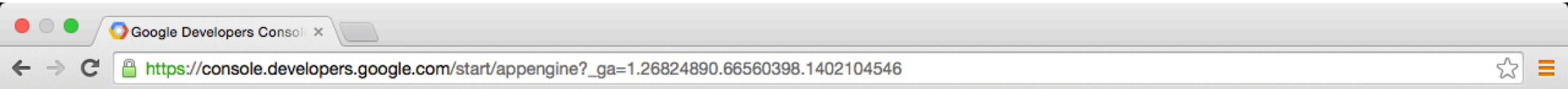
### HELLO WORLD - JAVA

```
build.xml 1 package myapp;
pom.xml    2
src/main/java/myapp/D 3 import java.io.IOException;
4 import javax.servlet.http.*;
5
src/main/webapp/WEB-INF 6 public class DemoServlet extends HttpServlet {
7     @Override
src/main/webapp/WEB-INF 8     public void doGet(HttpServletRequest req, HttpServletResponse resp)
9         throws IOException {
src/main/webapp/WEB-INF 10        resp.setContentType("text/plain");
```





# Google App Engine



## 3 EXPLORE THE STARTER CODE

Browse the starter code and see the app running below.

- 1. NAME YOUR PROJECT
- 2. SELECT YOUR LANGUAGE
- 3. EXPLORE THE STARTER CODE
- 4. INSTALL GOOGLE CLOUD SDK
- 5. RUN YOUR APP LOCALLY
- 6. CREATE YOUR PROJECT AND DEPLOY

### HELLO WORLD - JAVA

```
build.xml 1 package myapp;
pom.xml   2
src/main/java/myapp/D 3 import java.io.IOException;
4 import javax.servlet.http.*;
5
src/main/webapp/WEB-INF 6 public class DemoServlet extends HttpServlet {
7     @Override
src/main/webapp/WEB-INF 8     public void doGet(HttpServletRequest req, HttpServletResponse resp)
9         throws IOException {
src/main/webapp/WEB-INF 10         resp.setContentType("text/plain");
11         resp.getWriter().println("{ \"name\": \"World\" }");
12     }
13 }
14
```



Hello, World



# Google App Engine

## HELLO WORLD - JAVA

```
build.xml      1 package myapp;
pom.xml       2
src/main/java/myapp/D 3 import java.io.IOException;
src/main/webapp/WEB-INF 4 import javax.servlet.http.*;
src/main/webapp/WEB-INF 5
src/main/webapp/WEB-INF 6 public class DemoServlet extends HttpServlet {
src/main/webapp/WEB-INF 7     @Override
src/main/webapp/WEB-INF 8     public void doGet(HttpServletRequest req, HttpServletResponse resp)
src/main/webapp/WEB-INF 9         throws IOException {
src/main/webapp/index.t 10         resp.setContentType("text/plain");
11         resp.getWriter().println("{ \"name\": \"World\" }");
12     }
13 }
14
```

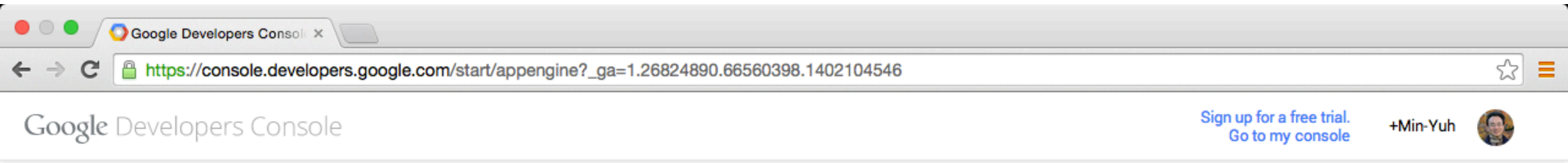


Hello, World

[Download this code](#)



# Google App Engine



## 4 INSTALL GOOGLE CLOUD SDK

Mac OS X/Linux

1. Download and install Google Cloud SDK by running the following command in your shell or Terminal:

```
curl https://sdk.cloud.google.com/ | bash
```

Follow the prompts to install the **Java** App Engine package.

2. Restart your shell or Terminal to pick up environment changes.
3. Sign in to Google Cloud Platform using this command:

```
gcloud auth login
```

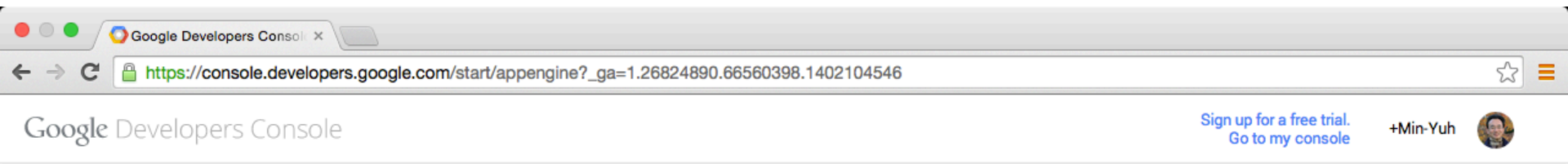
4. Install the App Engine package for Java using this command:

```
gcloud components update gae-java
```

## 5 RUN YOUR APP LOCALLY



# Google App Engine



## 5 RUN YOUR APP LOCALLY

**Note:** App Engine supports Java 7. Make sure you have the [Java 7 JDK](#) installed.

1. Download [appengine-try-java.zip](#) and unpack it. This creates your project directory, including `src/` and `war/` subdirectories.
2. Download and install [Apache Maven](#) version 3.0 or later. The starter app includes an Apache Maven build file.
3. Build and run the sample locally using the following commands:

```
cd appengine-try-java
mvn package
dev_appserver.sh target/appengine-try-java-1.0
```

You can stop the server by pressing Ctrl-C in the command window.

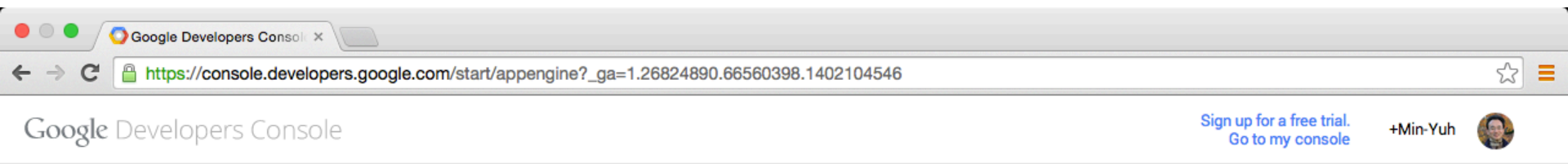
4. Visit the locally running app in your browser: <http://localhost:8080>

## 6 CREATE YOUR PROJECT AND DEPLOY

Now that you've seen your app running on your local machine, you're ready to create and deploy your project, HelloWorldGoogleAppEngine.



# Google App Engine



## 5 RUN YOUR APP LOCALLY

**Note:** App Engine supports Java 7. Make sure you have the [Java 7 JDK](#) installed.

1. Download [appengine-try-java.zip](#) and unpack it. This creates your project directory, including `src/` and `war/` subdirectories.
2. Download and install [Apache Maven](#) version 3.0 or later. The starter app includes an Apache Maven build file.
3. Build and run the sample locally using the following commands:

```
cd appengine-try-java
mvn package
dev_appserver.sh target/appengine-try-java-1.0
```

You can stop the server by pressing Ctrl-C in the command window.

4. Visit the locally running app in your browser: <http://localhost:8080>

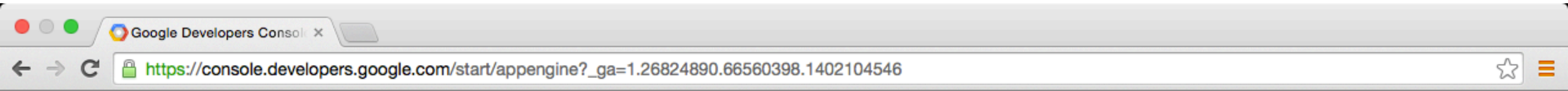
## 6 CREATE YOUR PROJECT AND DEPLOY

Now that you've seen your app running on your local machine, you're ready to create and deploy your project, HelloWorldGoogleAppEngine.

1. NAME YOUR PROJECT
2. SELECT YOUR LANGUAGE
3. EXPLORE THE STARTER CODE
4. INSTALL GOOGLE CLOUD SDK
5. RUN YOUR APP LOCALLY
6. CREATE YOUR PROJECT AND DEPLOY



# Google App Engine



- 1. NAME YOUR PROJECT
- 2. SELECT YOUR LANGUAGE
- 3. EXPLORE THE STARTER CODE
- 4. INSTALL GOOGLE CLOUD SDK
- 5. RUN YOUR APP LOCALLY
- 6. CREATE YOUR PROJECT AND DEPLOY

```
mvn package
dev_appserver.sh target/appengine-try-java-1.0
```

You can stop the server by pressing Ctrl-C in the command window.

- 4. Visit the locally running app in your browser: <http://localhost:8080>

## 6 CREATE YOUR PROJECT AND DEPLOY

Congratulations! Your project is ready. Your unique project ID is **psychic-outcome-783**.

Deploy your app from your local dev environment using this command:

```
appcfg.sh -A psychic-outcome-783 update target/appengine-try-java-1.0
```

After deploying your app, you can visit it with your browser at this URL:

[psychic-outcome-783.appspot.com](http://psychic-outcome-783.appspot.com)

That's it! You're running on Google App Engine. Go to your project dashboard to see how your app is performing.

[View your project dashboard](#)

**Activities (Idle)** G \_ x

Create Project: HelloWorldGoogleAppEngine ✔

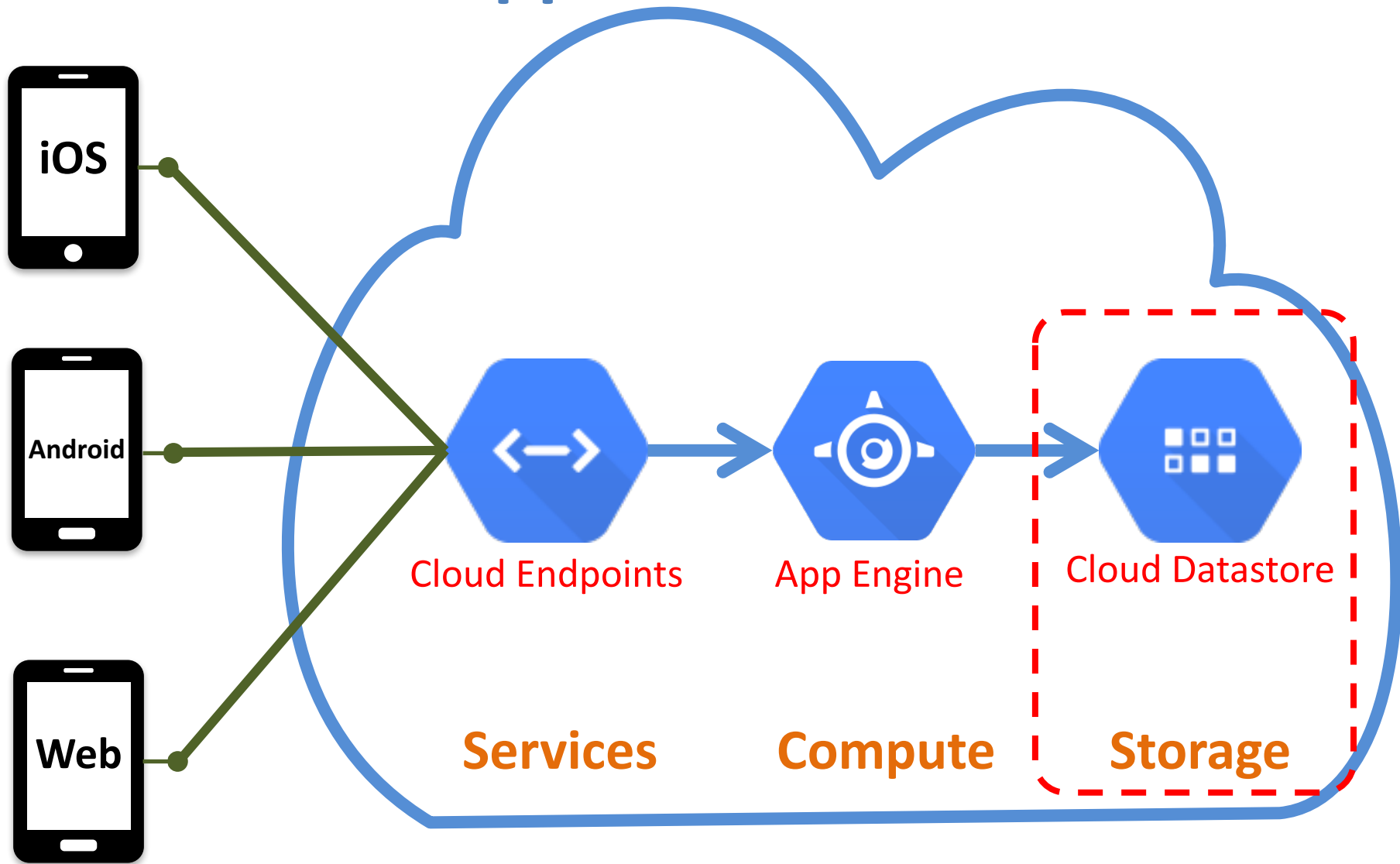
[See all activity](#)



# Google Cloud Datastore



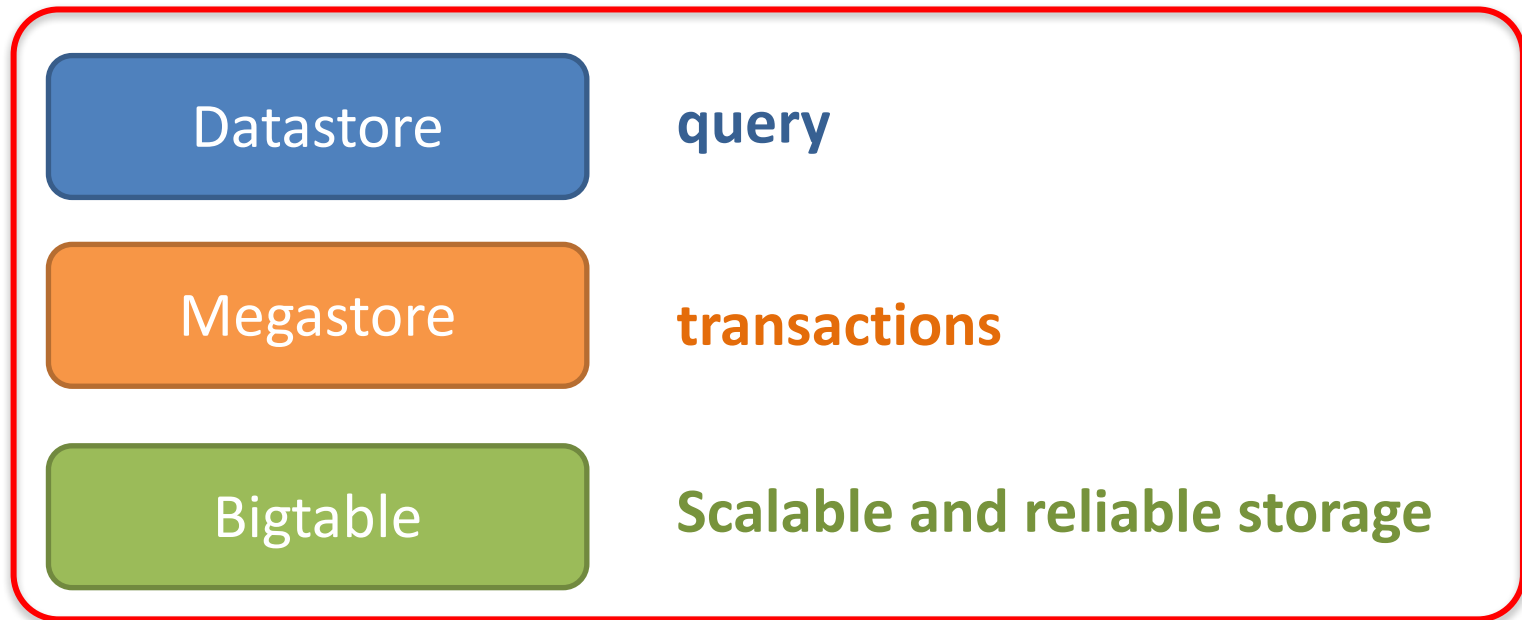
# Mobile App Backend Services





# Datastore Internals

- Based on Bigtable
  - high scalability
  - High availability
    - synchronous writes on multiple datacenters



# What is Bigtable?

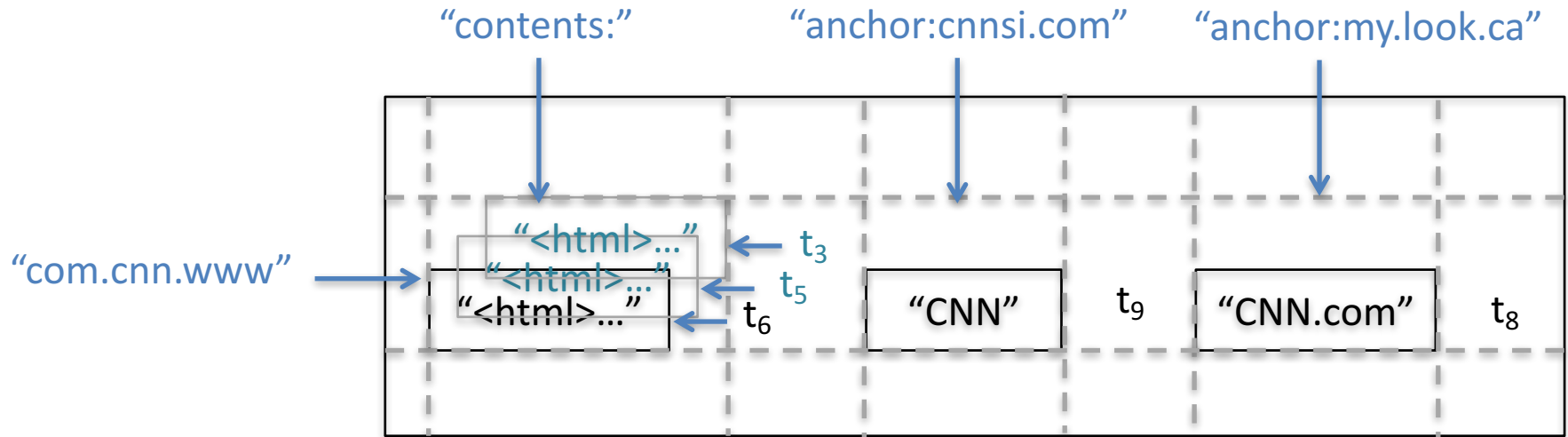
- Scalable, distributed, highly-available and structured storage
  - Bigtable is not database by itself (it doesn't support query)
- Google usage
  - In production since April 2005
  - Web Search, YouTube, Earth, Analytics

Bigtable

Scalable and reliable storage

# Bigtable Data Model

- A row has a Key and Columns
- Sorted by Key
  - In lexical order
  - Enables range query by application

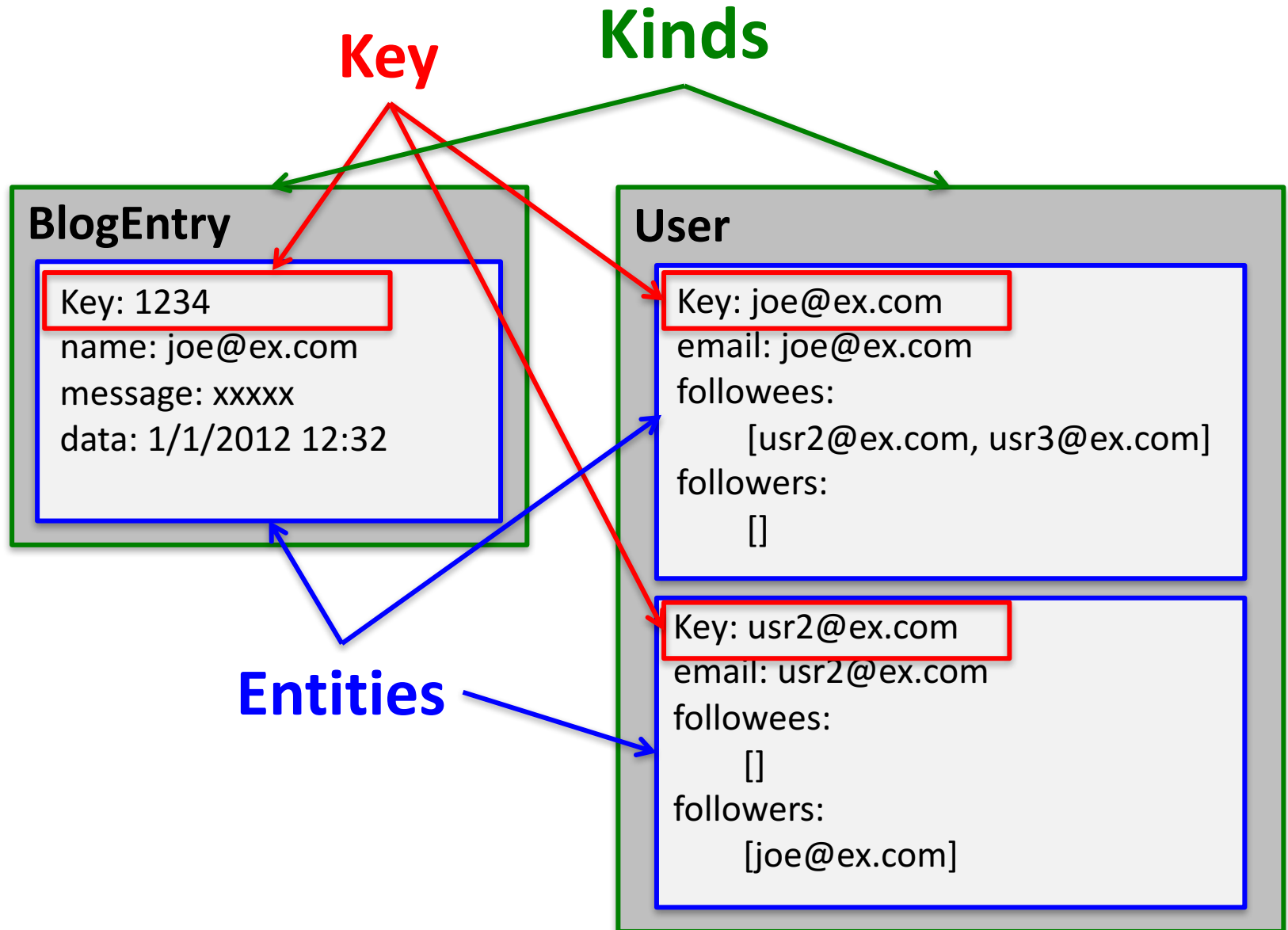


# Google Datastore Basic Operation

Different terms for corresponding concepts

	Google Datastore	Relational Database Management System (RDBMS)
Category of object	Kind	Table
One entry/object	Entity	Row
Unique identifier of data entry	Key	Primary Key (PK)
Individual data	Property	Field

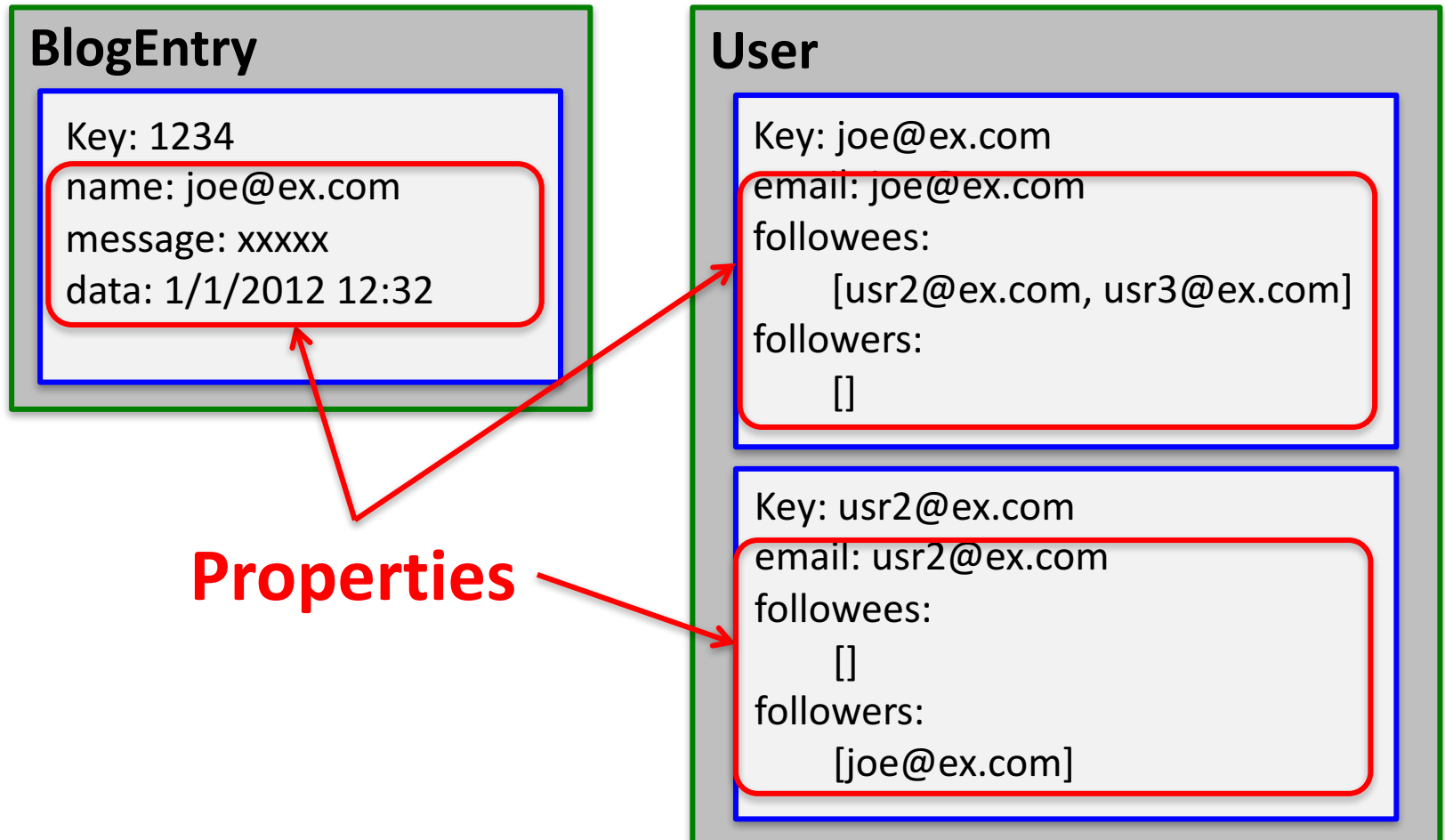
# Kind, Entity and Key



# Properties and Data Types

Each entity has one or more **named properties**

- Variety of datatypes (int, float, boolean, String, Date,...)
- Can be multi-valued

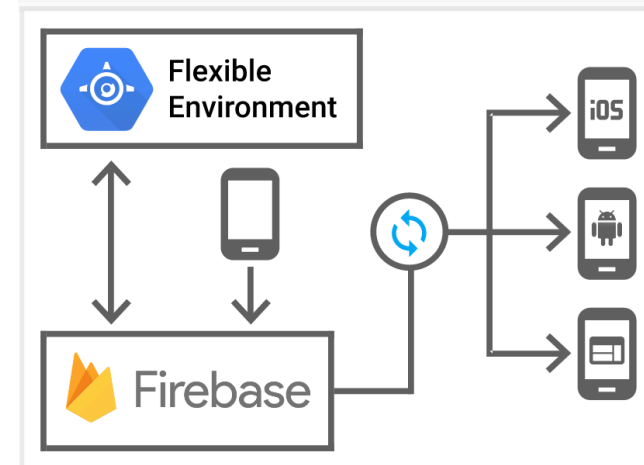
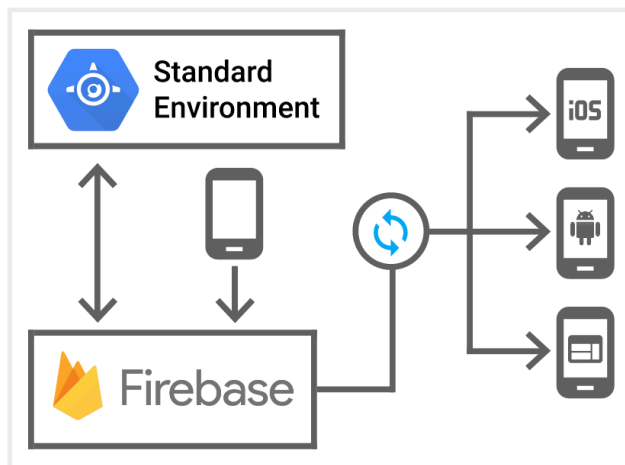
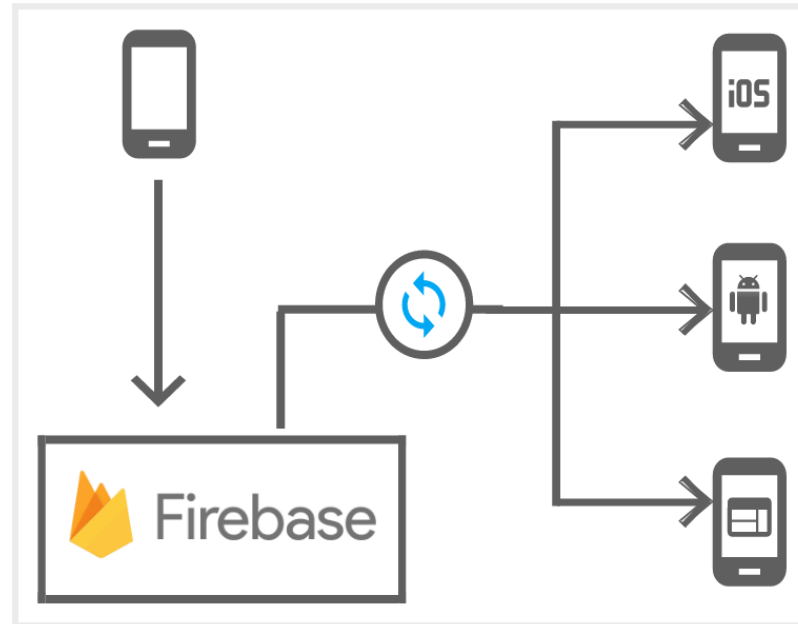


# Creating an Entity with Java Low-level API

```
DatastoreService datastore =  
DatastoreServiceFactory.getDatastoreService();  
  
Entity employee = new Entity("Employee");  
employee.setProperty("name", "Antonio Saliery");  
employee.setProperty("hireDate", new Date());  
employee.setProperty("attendedHrTraining", true);  
datastore.put(employee);
```



# Mobile App Backend Services







# Firebase

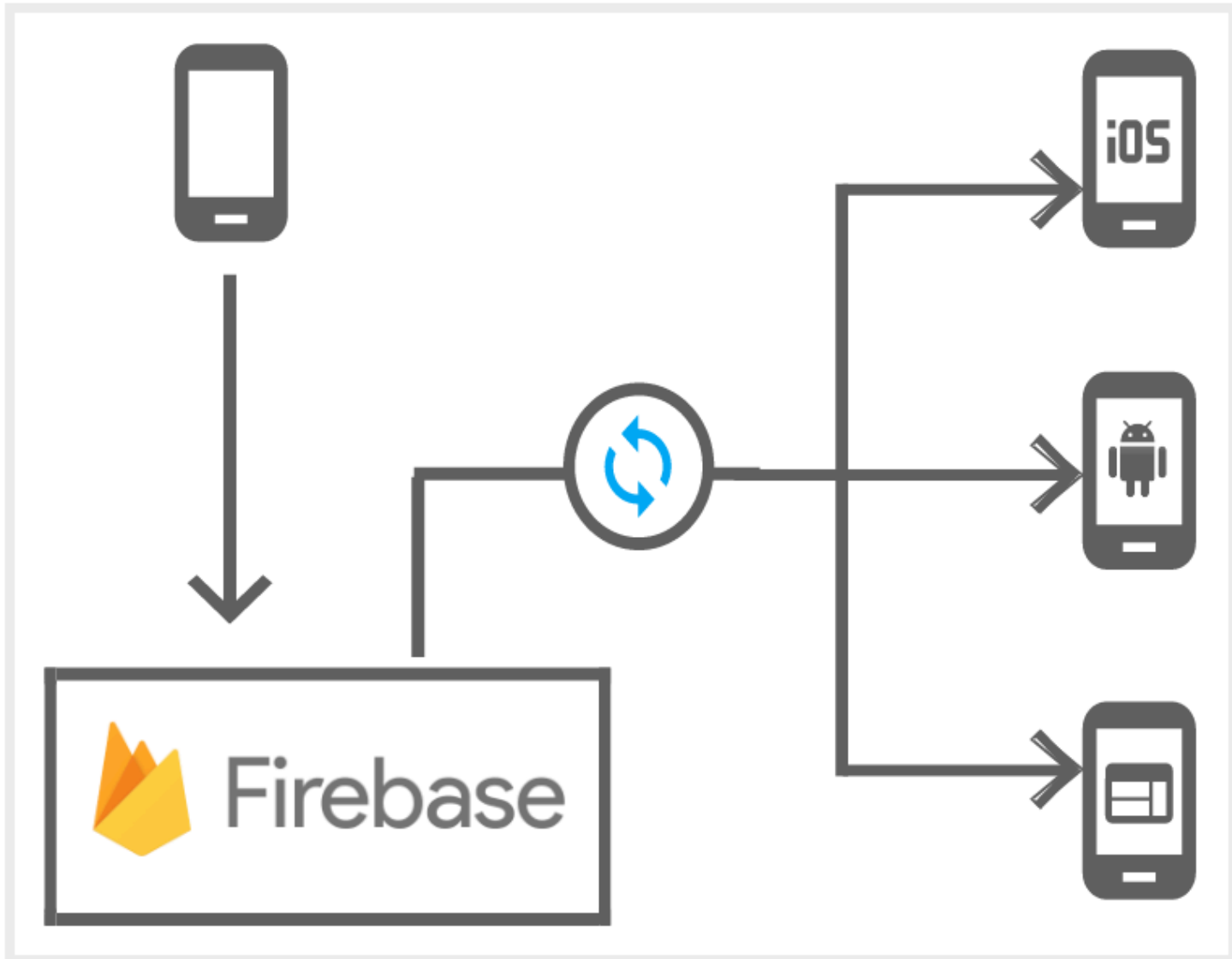


# Firebase

## Firebase: Backend as a Service (BaaS)

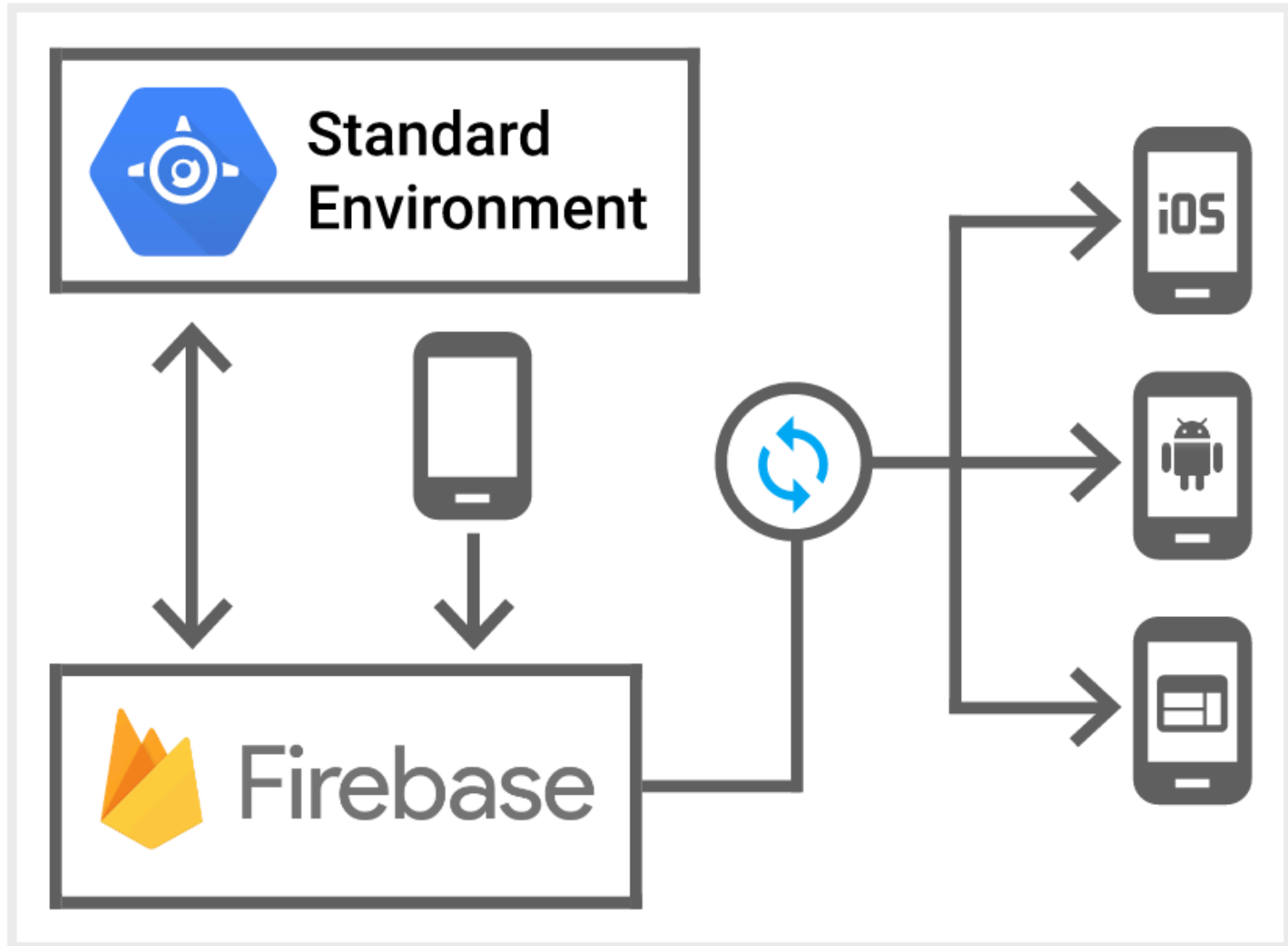


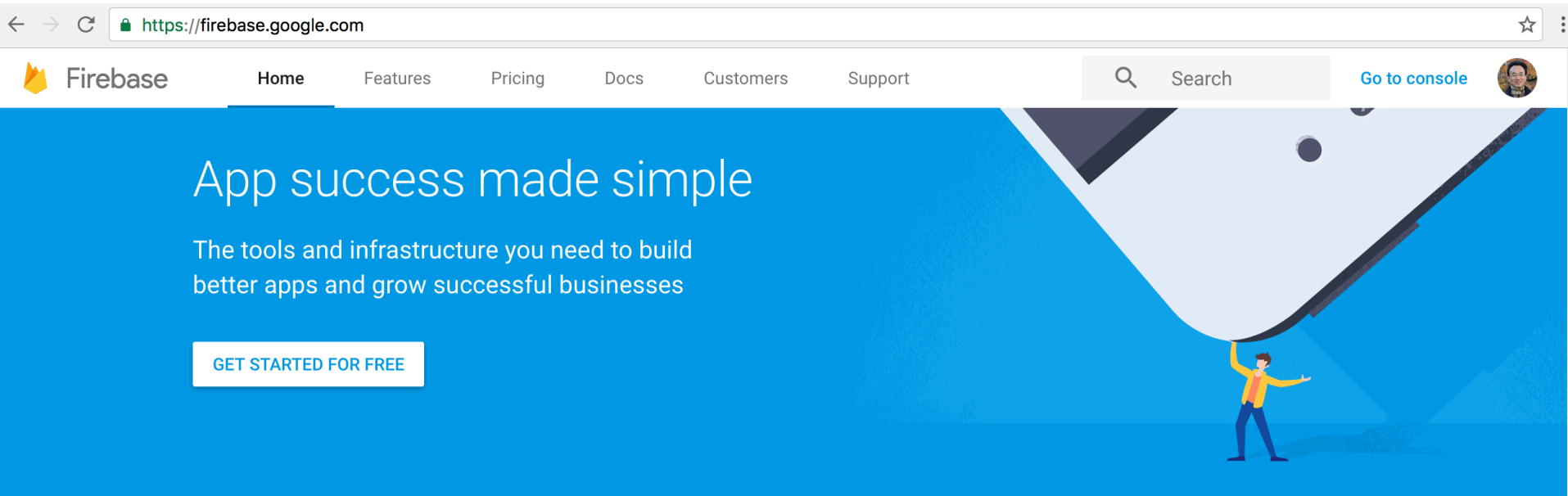
# Mobile App Backend Services





# Mobile App Backend Services

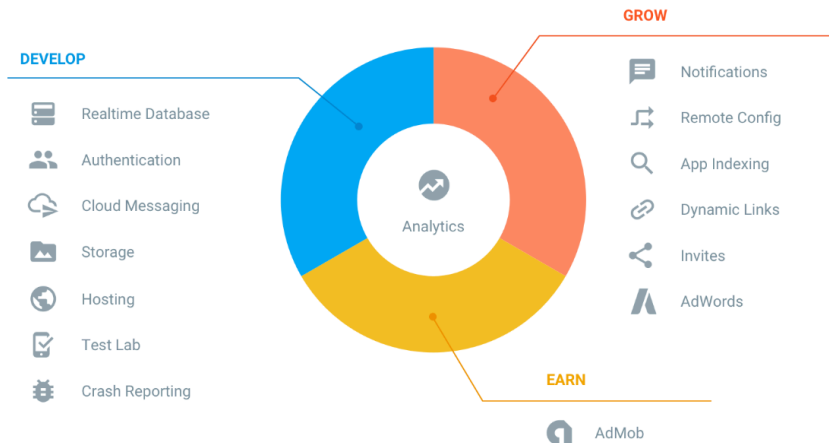




## Move fast








Firebase is a mobile platform that helps you quickly **develop** high-quality apps, **grow** your user base, and **earn** more money. Firebase is made up of complementary features that you can mix-and-match to fit your needs.

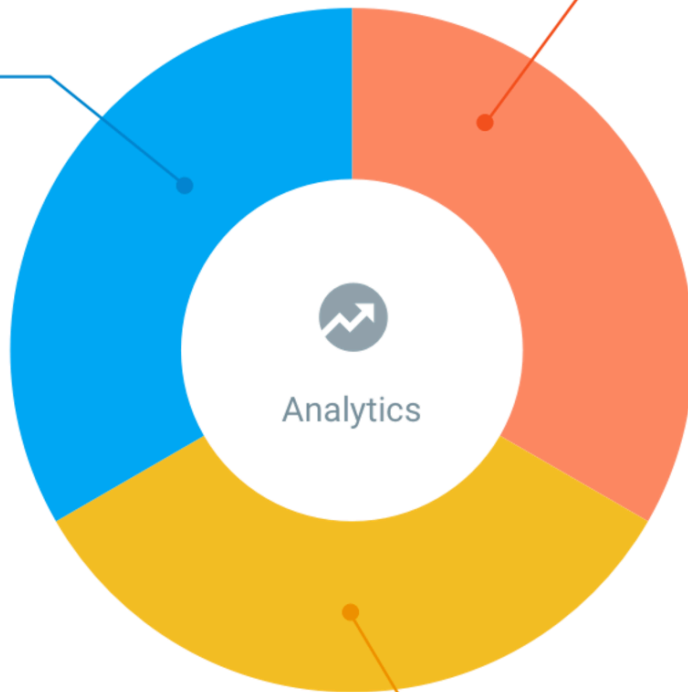
[ALL FEATURES](#)









# Firebase

## DEVELOP

-  Realtime Database
-  Authentication
-  Cloud Messaging
-  Storage
-  Hosting
-  Test Lab
-  Crash Reporting

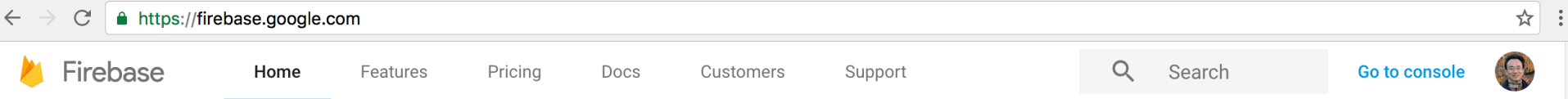


## GROW

-  Notifications
-  Remote Config
-  App Indexing
-  Dynamic Links
-  Invites
-  AdWords

## EARN




-  AdMob



```
SWIFT OBJECTIVE-C JAVA JAVASCRIPT MORE ▾  
  
FIRAuth.auth()?.addAuthStateDidChangeListener({ (auth, user) in  
  if (user) {  
    var email = user.email  
    // ...  
  }  
})  
FIRAuth.auth()?.signInWithEmail(email,  
                                password: password,  
                                completion: { (_, error) in  
  
  if (error) {  
    // ...  
  }  
})
```

## Work across platforms

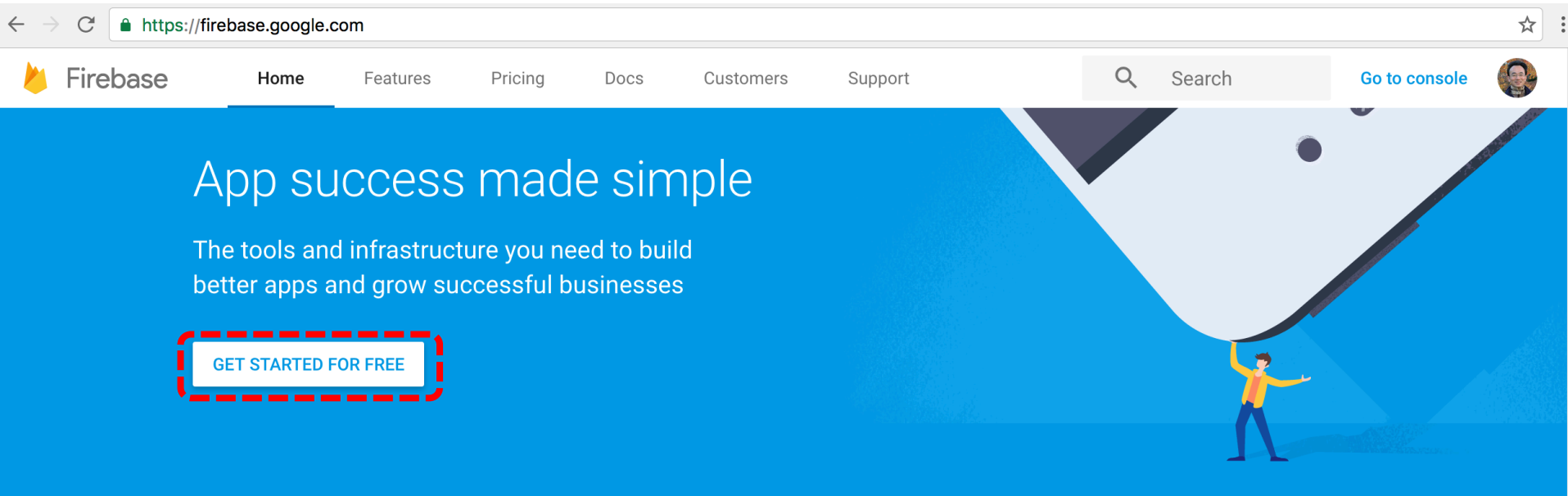
Deliver cross-platform apps with APIs packaged into single SDKs for iOS, Android, JavaScript and C++. Expand to a different platform without modifying your infrastructure.

-  [Try iOS tutorial](#)
-  [Try Android tutorial](#)
-  [Try Web tutorial](#)

## Free to start, scale with ease

Most Firebase features are free forever, for any scale. Our four paid features have a generous free tier and two paid plans once you begin to grow.

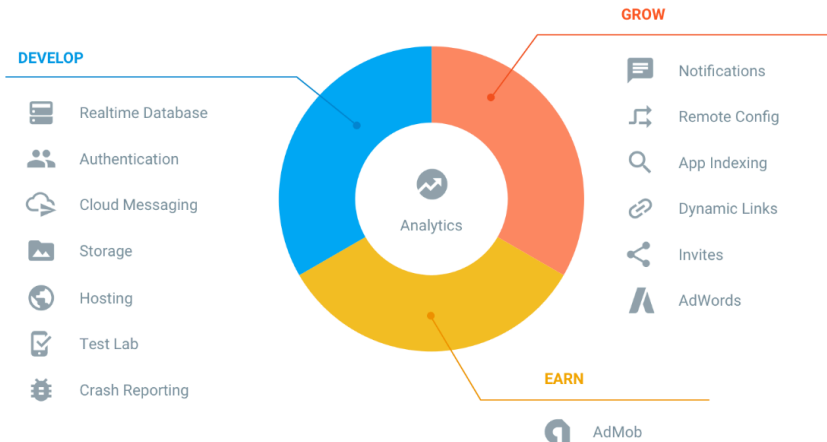




## Move fast

Firebase is a mobile platform that helps you quickly **develop** high-quality apps, **grow** your user base, and **earn** more money. Firebase is made up of complementary features that you can mix-and-match to fit your needs.

[ALL FEATURES](#)





# Firebase Project

← → ↻ <https://console.firebase.google.com/?pli=1> ☆

## Welcome to Firebase

Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads. [Learn more](#)

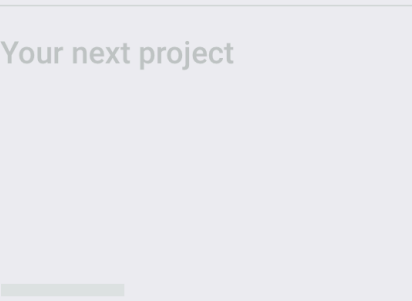
**CREATE NEW PROJECT**

[or import a Google project](#)

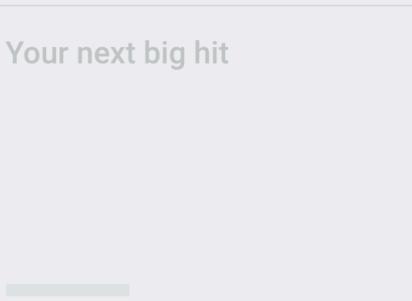


Your projects using Firebase

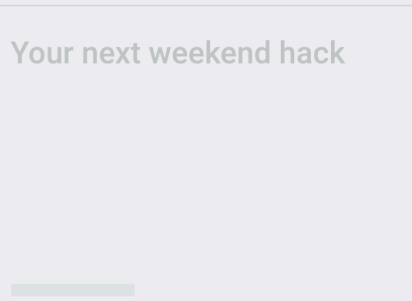
Your next project



Your next big hit



Your next weekend hack



# Firebase Project



## Welcome to Firebase

Tools from Google for developing g  
engaging with your users, and earni  
mobile ads. [Learn more](#)

CREATE NEW PROJECT

[or import a Google project](#)

Your projects using Firebase

Your next project



### Create a project



Project name

Country/region [?](#)

By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. [Learn more](#)

**By proceeding and clicking the button below**, you agree that you are using Firebase services in your app and agree to the applicable [terms](#).

CANCEL

CREATE PROJECT

next weekend hack

# Firestore Project

## Welcome to Firestore

Tools from Google for developing g...  
engaging with your users, and earni...  
mobile ads. [Learn more](#)

CREATE NEW PROJECT

[or import a Google project](#)

### Create a project

Project name

My awesome project

Country/region ?

United States

By default, your Firestore Analytics data will enhance other Firestore features and Google products. You can control how your Firestore Analytics data is shared in your settings at anytime. [Learn more](#)

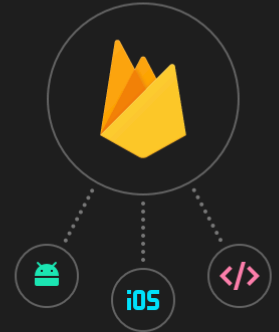
**By proceeding and clicking the button below**, you agree that you are using Firestore services in your app and agree to the applicable [terms](#).

CANCEL

CREATE PROJECT

### Projects span apps across platforms.

A project is a container for your apps across platforms: Android, iOS and web. It supports sharing features such as Database, User Management and Remote Config across your apps.



Your next project

next weekend hack

# Firebase Project

## Welcome to Firebase

Tools from Google for developing great apps, growing your business, and engaging with your users, and earning money from mobile ads. [Learn more](#)

CREATE NEW PROJECT

[or import a Google project](#)

Your projects using Firebase

Your next project

## Create a project ×

Project name

HelloWorldFirebase

Country/region ?

United States ▼

By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. [Learn more](#)

**By proceeding and clicking the button below**, you agree that you are using Firebase services in your app and agree to the applicable [terms](#).

CANCEL

CREATE PROJECT

# Add Firebase to Your Web App

The screenshot shows the Firebase console interface. At the top, the browser address bar displays the URL `https://console.firebase.google.com/project/helloworldfirebase-d6ef2/overview`. The page header includes the Firebase logo, the project name "HelloWorldFirebase", and a "Go to docs" link. A left-hand navigation menu lists various services: Overview, Analytics, Authentication, Database, Storage, Hosting, Test Lab, Crash Reporting, Notifications, Remote Config, Dynamic Links, and AdMob. The main content area features a "Welcome to Firebase! Get started here." message with three prominent options: "Add Firebase to your iOS app", "Add Firebase to your Android app", and "Add Firebase to your web app". The "Add Firebase to your web app" option is highlighted with a red dashed border. Below this, a "Discover Firebase" section introduces "Analytics", "Authentication", and "Database" with brief descriptions and illustrative images.

# Add Firebase to Your Web App

https://console.firebase.google.com/project/helloworldfirebase-d6ef2/overview

Overview

## Add Firebase to your web app

Copy and paste the snippet below at the bottom of your HTML, before other `script` tags.

```
<script src="https://www.gstatic.com/firebasejs/3.6.4/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyDu4JodvREiNjYJP2_YBVgZLpNu1HUEB_g",
    authDomain: "helloworldfirebase-d6ef2.firebaseio.com",
    databaseURL: "https://helloworldfirebase-d6ef2.firebaseio.com",
    storageBucket: "helloworldfirebase-d6ef2.appspot.com",
    messagingSenderId: "778984601289"
  };
  firebase.initializeApp(config);
</script>
```

**COPY**

Check these resources to learn more about Firebase for web apps:

- [Get Started with Firebase for Web Apps](#)
- [Firebase Web SDK API Reference](#)
- [Firebase Web Samples](#)

Analytics  
Authentication  
Database  
Storage  
Hosting  
Test Lab  
Crash Reporting  
Notifications  
Remote Config  
Dynamic Links  
AdMob

Analytics  
Authentication  
Database

# Add Firebase to Your Web App

```
<script src="https://www.gstatic.com/firebasejs/3.6.4/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyDu4JodvREiNjYJP2_YBVgZLpNu1HUEB_g",
    authDomain: "helloworldfirebase-d6ef2.firebaseio.com",
    databaseURL: "https://helloworldfirebase-d6ef2.firebaseio.com",
    storageBucket: "helloworldfirebase-d6ef2.appspot.com",
    messagingSenderId: "778984601289"
  };
  firebase.initializeApp(config);
</script>
```

# Add Firebase to Your Web App

```
<script src="https://www.gstatic.com/firebasejs/3.6.1/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>
```

**firebase-app** - The core firebase client (required).

**firebase-auth** - Firebase Authentication (optional).

**firebase-database** - The Firebase Realtime Database (optional).



# Add Firebase to Your Web App

```
<script src="https://www.gstatic.com/firebasejs/3.6.1/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/3.6.1/firebase-auth.js"></script>
<script src="https://www.gstatic.com/firebasejs/3.6.1/firebase-database.js"></script>
<script src="https://www.gstatic.com/firebasejs/3.6.1/firebase-messaging.js"></script>

<!-- Leave out Storage -->
<!-- <script src="https://www.gstatic.com/firebasejs/3.6.1/firebase-storage.js"></script> -->

<script>
  var config = {
    // ...
  };
  firebase.initializeApp(config);
</script>
```

# Add Firebase to Your Web App

```
var firebase = require("firebase/app");
require("firebase/auth");
require("firebase/database");

// Leave out Storage
//require("firebase/storage");

var config = {
  // ...
};
firebase.initializeApp(config);
```

# Add Firebase to Your Web App

- `firebase.auth()` - Authentication
- `firebase.storage()` - Storage
- `firebase.database()` - Realtime Database

# Firestore Database Web Start

```
// Set the configuration for your app
// TODO: Replace with your project's config object
var config = {
  apiKey: "apiKey",
  authDomain: "projectId.firebaseapp.com",
  databaseURL: "https://databaseName.firebaseio.com",
  storageBucket: "bucket.appspot.com"
};
firebase.initializeApp(config);

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

# Firestore Database Structure Data

```
{
  // Chats contains only meta info about each conversation
  // stored under the chats's unique ID
  "chats": {
    "one": {
      "title": "Historical Tech Pioneers",
      "lastMessage": "ghopper: Relay malfunction found. Cause: moth.",
      "timestamp": 1459361875666
    },
    "two": { ... },
    "three": { ... }
  },

  // Conversation members are easily accessible
  // and stored by chat conversation ID
  "members": {
    // we'll talk about indices like this below
    "one": {
      "ghopper": true,
      "alovelace": true,
      "eclarke": true
    },
    "two": { ... },
    "three": { ... }
  },
}
```

# Firestore Database Structure Data

```
// Messages are separate from data we may want to iterate quickly
// but still easily paginated and queried, and organized by chat
// conversation ID
"messages": {
  "one": {
    "m1": {
      "name": "eclarke",
      "message": "The relay seems to be malfunctioning.",
      "timestamp": 1459361875337
    },
    "m2": { ... },
    "m3": { ... }
  },
  "two": { ... },
  "three": { ... }
}
```

# Firestore Database Write Data

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

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

# Firestore Database Read Data

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

```
var userId = firebase.auth().currentUser.uid;
return firebase.database().ref('/users/' + userId).once('value').then(function(snapshot) {
  var username = snapshot.val().username;
  // ...
});
```



# Firestore Database

The screenshot displays the Firebase console interface for a project named 'HelloWorldFirebase'. The left sidebar contains a navigation menu with categories: Overview, Analytics, DEVELOP (Authentication, Database, Storage, Hosting, Test Lab, Crash Reporting), GROW (Notifications, Remote Config, Dynamic Links), and EARN (AdMob). The 'Database' item is highlighted with a red dashed box. The main content area is titled 'Realtime Database' and has tabs for DATA, RULES, USAGE, and BACKUPS. A URL bar at the top of the main area shows 'https://helloworldfirebase-d6ef2.firebaseio.com/'.

# Firestore Database Rules

https://console.firebase.google.com/project/helloworldfirebase-d6ef2/database/rules

Overview Analytics DEVELOP Authentication Database Storage Hosting Test Lab Crash Reporting GROW Notifications Remote Config Dynamic Links EARN AdMob Spark Free \$0/month UPGRADE

Realtime Database DATA **RULES** USAGE BACKUPS

SIMULATOR

★ Default security rules require users to be authenticated LEARN MORE DISMISS

```
1 {
2   "rules": {
3     ".read": "auth != null",
4     ".write": "auth != null"
5   }
6 }
```

# Firestore Database

https://console.firebase.google.com/project/helloworldfirebase-d6ef2/database/data

Overview Analytics DEVELOP Authentication Database Storage Hosting Test Lab Crash Reporting GROW Notifications Remote Config Dynamic Links EARN AdMob Spark Free \$0/month [UPGRADE](#)

## Realtime Database

DATA RULES USAGE BACKUPS

https://helloworldfirebase-d6ef2.firebaseio.com/

★ Default security rules require users to be authenticated [LEARN MORE](#) [DISMISS](#)

helloworldfirebase-d6ef2: null + x

Store and sync data in realtime across all connected clients [Learn more](#)

# Firestore Database

The screenshot shows the Firebase Realtime Database console interface. The browser address bar displays the URL: `https://console.firebase.google.com/project/helloworldfirebase-d6ef2/database/data`. The page title is "Realtime Database" and the project name is "HelloWorldFirebase". The left sidebar contains navigation options: Overview, Analytics, Authentication, Database (highlighted), Storage, Hosting, Test Lab, Crash Reporting, Notifications, Remote Config, Dynamic Links, and AdMob. The main content area has tabs for DATA, RULES, USAGE, and BACKUPS. A notification banner states: "Default security rules require users to be authenticated". Below this, a data entry form is visible for the database instance "helloworldfirebase-d6ef2". The form includes a "Name" field with the value "score" and a "Value" field with the value "100". A red dashed box highlights the "ADD" button. Below the form, the JSON representation of the data is shown: `{ "email": "imyday@gmail.com", "username": "imyday" }`.

# Firestore Database

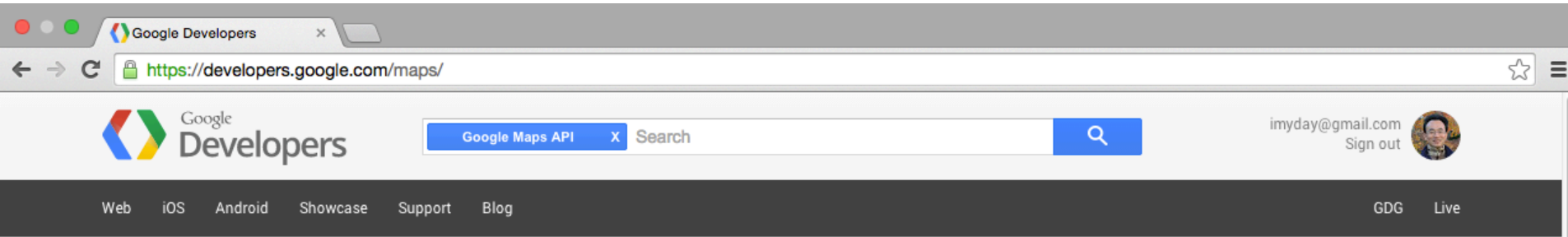
The screenshot shows the Firebase Realtime Database console. The browser address bar displays `https://console.firebase.google.com/project/helloworldfirebase-d6ef2/database/data`. The page title is "Realtime Database" with sub-tabs for "DATA", "RULES", "USAGE", and "BACKUPS". A left sidebar lists various Firebase services, with "Database" highlighted. The main content area shows a JSON object for the path `helloworldfirebase-d6ef2`, with fields `email: "imyday@gmail.com"`, `score: 100`, and `username: "imyday"`. A red dashed box highlights this JSON object. A notification banner at the top of the content area states "Default security rules require users to be authenticated" with "LEARN MORE" and "DISMISS" links.

```
helloworldfirebase-d6ef2
├── email: "imyday@gmail.com"
├── score: 100
└── username: "imyday"
```

# Google Maps API

<https://developers.google.com/maps/>

# Google Maps API

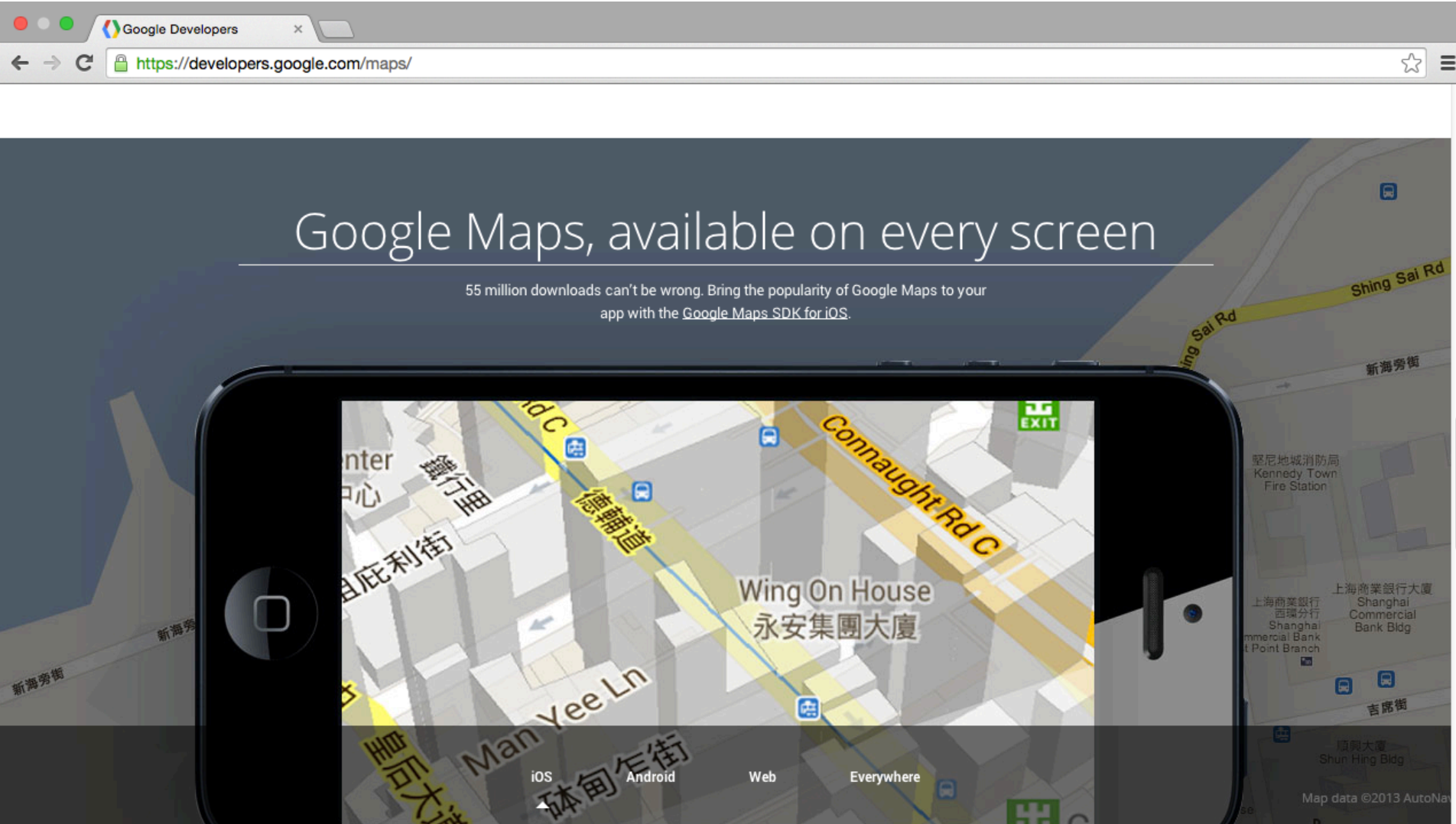


## Hello Map



<https://developers.google.com/maps/>

# Google Maps API



The image shows a browser window with the URL <https://developers.google.com/maps/>. The main heading reads "Google Maps, available on every screen". Below this, a sub-headline states: "55 million downloads can't be wrong. Bring the popularity of Google Maps to your app with the Google Maps SDK for iOS." The central focus is a smartphone displaying a 3D map of a city street. The map shows buildings, streets, and a highlighted path. Labels on the map include "Wing On House 永安集團大廈", "Connaught Rd C", "Man Yee Ln", and "德輔道". At the bottom of the smartphone screen, there are four platform icons: "iOS", "Android", "Web", and "Everywhere".

Google Developers

<https://developers.google.com/maps/>

## Google Maps, available on every screen

55 million downloads can't be wrong. Bring the popularity of Google Maps to your app with the Google Maps SDK for iOS.

Wing On House  
永安集團大廈

Connaught Rd C

Man Yee Ln

iOS Android Web Everywhere

<https://developers.google.com/maps/>



# Google Maps API Demos

Google Maps API

HOME DEMOS DEVELOPER STORIES LEARN MORE

## More than a Map

The Google Maps API is more than a map for your apps. Explore demos of unique features that you can use in your apps today!

[Go to the demos](#)

### Learn More

Build your own apps with the Google Maps API

[Read more](#)

### Developer Stories

Visit Google Maps developers from around the world

[Read more](#)

[#morethanamap on Google+](#)

[Map of the Week blog](#)

# Google Maps API Demos

Google Maps API

HOME DEMOS DEVELOPER STORIES LEARN MORE

## Demos

Base Maps Satellite Street View

Places Routing Data Visualization

www.morethanamap.com/demos

Map data ©2014 Google, SK planet Terms of Use

<http://www.morethanamap.com/demos>



# Google Maps API Demos

Google Maps API

HOME DEMOS DEVELOPER STORIES LEARN MORE

## Base Maps

For the last decade, we've obsessed over building great maps—maps that are comprehensive, accurate, and easy to use.

Base Maps **Styled Maps**

GO FURTHER

New York

Hong Kong

Search

www.morethanamap.com/demos/basemaps

Map data ©2014 Google Terms of Use Report a map error

<http://www.morethanamap.com/demos/basemaps/new-york>



# Google Maps API Demos

Google Maps API

HOME DEMOS DEVELOPER STORIES LEARN MORE

### Routing

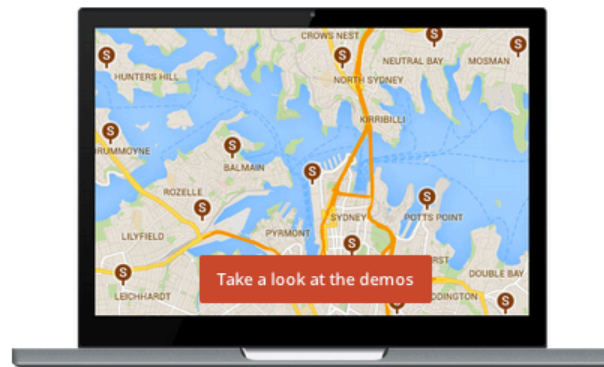
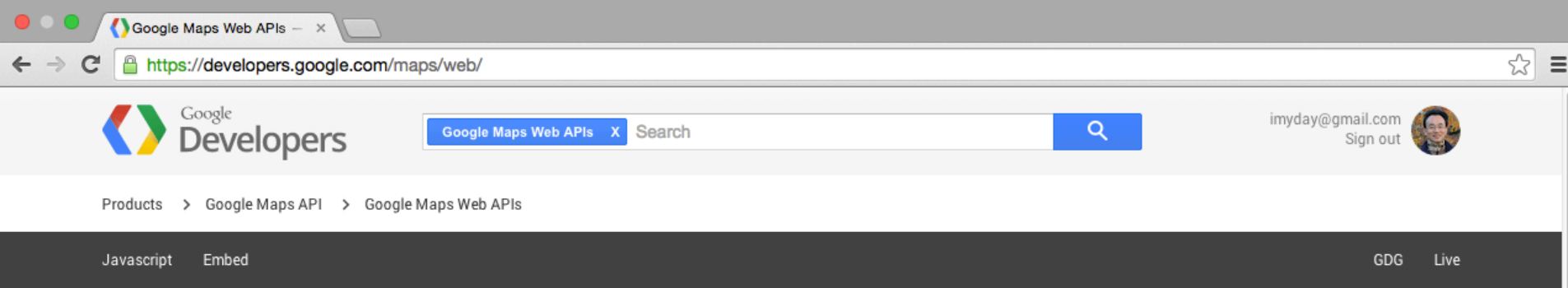
In a car, on a bike, on foot, or riding a subway, help users find the best route. Use elevation profiles and live traffic to guide the way.

SHOW ME:

- Cycling through Stockholm
- Public Transit in Sydney
- Elevation in San Francisco

Map data ©2014 Google Terms of Use Report a map error

# Google Maps JavaScript API v3



## Google Maps JavaScript API v3

### Build highly customisable maps with your own content and imagery

Create rich applications and stunning visualisations of your data, leveraging the comprehensiveness, accuracy, and usability of Google Maps and a modern web platform that scales as you grow.

In only a few lines of JavaScript code, build and style a map to call your own. With plenty of Google libraries and services at your disposal (including Geocoding, Directions, Street View and more) your imagination is truly the limit.

[Get Started with the JavaScript API v3](#)

## Google Maps Embed API

Make places easily discoverable with interactive maps built for your users

<https://developers.google.com/maps/web/>

# Google Maps JavaScript API

The screenshot shows a web browser window with the URL <https://developers.google.com/maps/documentation/javascript/tutorial>. The page header includes the Google Developers logo, a search bar with the text "Google Maps JavaScri...", and a user profile for "imyday@gmail.com" with a "Sign out" link. The breadcrumb trail is "Products > Google Maps API > Google Maps JavaScript API v3". The navigation menu includes "Get Started", "Documentation", "Reference", "Showcase", "Support", and "Blog". The main content area is titled "Google Maps JavaScript API v3" with a "G+1" button and a count of "507". A "Report documentation issue" link is also present. The left sidebar shows a "Developer's Guide" with a "Getting Started" section highlighted in red. The main content area has a "Getting Started" section with a list of links: "Audience", "Obtaining an API Key", "Hello, World", "Declaring Your Application as HTML5", "Loading the Google Maps API", "Map DOM Elements", "Map Options", "The Map Object", "Loading the Map", and "Troubleshooting". Below this is an "Audience" section with a horizontal line. The text under "Audience" states: "This documentation is designed for people familiar with JavaScript programming and object-oriented programming concepts. You should also be familiar with Google Maps from a user's point of view. There are many JavaScript tutorials available on the Web." Below this is another section of text: "This conceptual documentation is designed to let you quickly start exploring and developing applications with the Google Maps API. We also".

<https://developers.google.com/maps/documentation/javascript/tutorial>

# Obtaining an Google Maps API Key

Getting Started - Google M x

← → ↻ <https://developers.google.com/maps/documentation/javascript/tutorial> ☆ ☰

More Resources

- Blog
- Support
- FAQ
- API Picker

-----

- Google Maps API for Work
- Maps API Web Services
- Google Places API
- Static Maps API
- Street View Image API
- Earth API Deprecated

## Obtaining an API Key

All Maps API applications\* should load the Maps API using an API key. Using an API key enables you to monitor your application's Maps API usage, and ensures that Google can contact you about your application if necessary. If your application's Maps API usage exceeds the [Usage Limits](#), you must load the Maps API using an API key in order to purchase additional quota.

\* **Google Maps API for Work** developers must *not* include a key in their requests. Please refer to [Loading the Google Maps JavaScript API](#) for instructions.

To create your API key:

1. Visit the APIs Console at <https://code.google.com/apis/console> and log in with your Google Account.
2. Click the **Services** link from the left-hand menu.
3. Activate the **Google Maps JavaScript API v3** service.
4. Click the **API Access** link from the left-hand menu. Your API key is available from the **API Access** page, in the **Simple API Access** section.

Maps API applications use the **Key for browser apps**.

Google apis

API Project

- Overview
- Services
- Team
- API Access
- Billing
- Reports
- Quotas

### API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

#### Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. [Learn more](#)

Create an OAuth 2.0 client ID...

#### Simple API Access

Use API keys to identify your project when you do not need to access user data. [Learn more](#)

**Key for browser apps (with referers)**

API key: R9Waa2yA1c42oB4D\_hhT12e1lc4ZRwofCcyeeRNU

Referers: Any Referer allowed

Activated on: Nov 3, 2010 12:57 PM

Activated by: @gmail.com - you

Create new Server key... Create new Browser key...

**Demo:**  
**Integrate**  
**Google Maps JavaScript API**  
**with**  
**jQuery Mobile**




# Start using the Google APIs console



The screenshot shows a web browser window with the Google APIs Console interface. The browser's address bar displays the URL <https://code.google.com/apis/console/?noredirect&pli=1>. The page features the Google APIs logo on the left. In the center, a white box contains the heading "Start using the Google APIs console" and the subtext "to manage your API usage". Below this is a blue cube icon with green arrows. A list of benefits for creating an APIs project is provided, followed by a prominent blue button labeled "Create project..." which is highlighted with a red rectangular border. At the bottom of the page, there are links for "Code Home" and "Privacy Policy".

Google APIs

**Start using the Google APIs console**  
to manage your API usage



Creating an **APIs project** will let you:

- Use Google APIs **beyond anonymous limits**.
- **Monitor** API usage and **control** API access.
- **Share** API management with a team.

**Create project...**

[Code Home](#) - [Privacy Policy](#)

<https://code.google.com/apis/console>

<https://code.google.com/apis/console/?noredirect&pli=1>

Send Feedback

# Google APIs Console

Google APIs Console

https://code.google.com/apis/console/?noredirect&pli=1#project:841318404499:services

Search Images Maps Play YouTube News Gmail Drive More

imyday@gmail.com | Settings | Help | Sign out

Google apis

We are improving the experience. Try the new [Cloud Console](#).

API Project

All (71) Active (0) Inactive (71) Google Cloud Platform

**All services**  
Select services for the project.





































Service	Status	Notes
Ad Exchange Buyer API	<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
Ad Exchange Seller API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
Admin SDK	<input type="checkbox"/> OFF	
AdSense Host API	<a href="#">Request access...</a>	Courtesy limit: 100,000 requests/day
AdSense Management API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
Analytics API	<input type="checkbox"/> OFF	Courtesy limit: 50,000 requests/day
Audit API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
BigQuery API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day • <a href="#">Pricing</a>
Blogger API v3	<a href="#">Request access...</a>	Courtesy limit: 10,000 requests/day
Books API	<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
CalDAV API	<input type="checkbox"/> OFF	Courtesy limit: 1,000,000 requests/day
Calendar API	<input type="checkbox"/> OFF	Courtesy limit: 100,000 requests/day
Chrome Web Store API	<input type="checkbox"/> OFF	

Send Feedback

# Google APIs Console

Google APIs Console

https://code.google.com/apis/console/?noredirect&pli=1#project:841318404499:services

 Google Apps Reseller API		<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 Google Civic Information API		<input type="checkbox"/> OFF	Courtesy limit: 25,000 requests/day
 Google Cloud Datastore API		<input type="checkbox"/> OFF	Courtesy limit: 10,000,000 requests/day
 Google Cloud Messaging for Android		<input type="checkbox"/> OFF	
 Google Cloud Messaging for Chrome		<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 Google Cloud SQL		<input type="checkbox"/> OFF	<a href="#">Pricing</a>
 Google Cloud SQL API		<input type="checkbox"/> OFF	
 Google Cloud Storage		<input type="checkbox"/> OFF	<a href="#">Pricing</a>
 Google Cloud Storage JSON API		<input type="checkbox"/> OFF	
 Google Compute Engine		<input type="checkbox"/> OFF	<a href="#">Pricing</a>
 Google Contacts CardDAV API		<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 Google Maps Android API v2		<input type="checkbox"/> OFF	
 Google Maps API v3		<input type="checkbox"/> OFF	Courtesy limit: 25,000 requests/day • <a href="#">Pricing</a>
 Google Maps Coordinate API		<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
 Google Maps Engine API		<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 Google Maps Geolocation API		<input type="checkbox"/> OFF	Courtesy limit: 0 requests/day • <a href="#">Pricing</a>
 Google Maps SDK for iOS		<input type="checkbox"/> OFF	
 Google Maps Tracks API		<input type="checkbox"/> OFF	

Send Feedback

# Google APIs Console

Google APIs Console

https://code.google.com/apis/console/?noredirect&pli=1#project:841318404499:services

Search Images Maps Play YouTube News Gmail Drive More

imyday@gmail.com | Settings | Help | Sign out

Google apis

We are improving the experience. Try the new [Cloud Console](#).

Review terms of service

Print

## Google Maps/Google Earth APIs Terms of Service

Last Updated: May 10, 2013

### 1. Your relationship with Google.

1.1 Use of the Service is Subject to these Terms. Your use of any of the Google Maps/Google Earth APIs (referred to in this document as the "**Maps API(s)**" or the "**Service**") is subject to the terms of a legal agreement between you and Google (the "**Terms**"). "Google" means either (a) Google Ireland Limited, with offices at Gordon House, Barrow Street, Dublin 4, Ireland, if Customer's billing address is in any country within Europe, the Middle East, or Africa ("**EMEA**"); (b) Google Asia Pacific Pte. Ltd., with offices at 8 Marina View Asia Square 1 #30-01 Singapore 018960, if Customer's billing address is in any country within the Asia Pacific region ("**APAC**"); or (c) Google Inc., with offices at 1600 Amphitheatre Parkway, Mountain View, California 94043, USA, if Customer's billing address is in any country in the world other than those in EMEA and APAC.

1.2 The Terms include Google's Legal Notices and Privacy Policy.

(a) Unless otherwise agreed in writing with Google, the Terms will include the following:

- (i) the terms and conditions set forth in this document (the "**Maps APIs Terms**");
- (ii) the [Legal Notices](#); and
- (iii) the [Privacy Policy](#).

I agree to these terms.

0 of 1 terms of service accepted.  
Google Maps/Earth APIs

[Code Home](#) - [Privacy Policy](#)

Send Feedback

# Google APIs Console

Google APIs Console

https://code.google.com/apis/console/?noredirect&pli=1#project:841318404499:services

Search Images Maps Play YouTube News Gmail Drive More

imyday@gmail.com | Settings | Help | Sign out

**Google apis**

We are improving the experience. Try the new [Cloud Console](#).



API Project

All (71) Active (1) Inactive (70) Google Cloud Platform

Overview  
Services  
Team  
API Access  
Billing  
Reports  
Quotas

### Active services

Select services for the project.

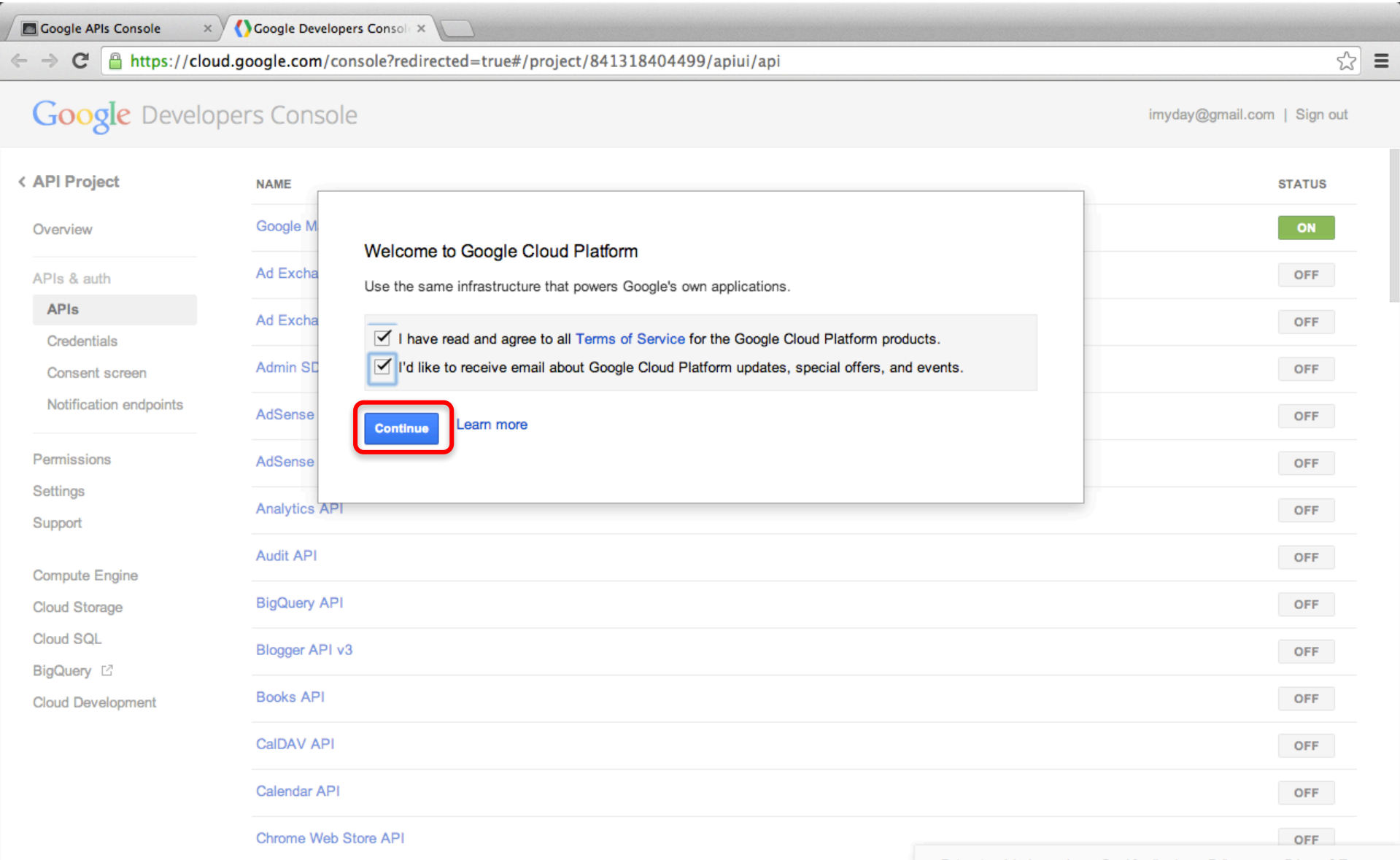
Service	Status	Notes
 Google Maps API v3 	<input checked="" type="checkbox"/>	Courtesy limit: 25,000 requests/day • <a href="#">Pricing</a>

[Code Home](#) - [Privacy Policy](#)

Send Feedback

# Google Developers Console

## Google Cloud Platform



The screenshot shows the Google Developers Console interface. A modal dialog box is centered on the screen, titled "Welcome to Google Cloud Platform". The dialog contains the following text and elements:

- Header: "Welcome to Google Cloud Platform"
- Text: "Use the same infrastructure that powers Google's own applications."
- Form fields:
  - I have read and agree to all [Terms of Service](#) for the Google Cloud Platform products.
  - I'd like to receive email about Google Cloud Platform updates, special offers, and events.
- Buttons: A blue "Continue" button (highlighted with a red box) and a "Learn more" link.

The background shows a table of APIs with their status:

NAME	STATUS
Google M	ON
Ad Excha	OFF
Ad Excha	OFF
Admin SD	OFF
AdSense	OFF
AdSense	OFF
Analytics API	OFF
Audit API	OFF
BigQuery API	OFF
Blogger API v3	OFF
Books API	OFF
CalDAV API	OFF
Calendar API	OFF
Chrome Web Store API	OFF

# Google Maps API v3

The screenshot shows the Google Developers Console interface. At the top, there are two browser tabs: "Google APIs Console" and "Google Developers Console". The address bar shows the URL: [https://cloud.google.com/console?redirected=true#/project/841318404499/apiui/api/maps\\_backend](https://cloud.google.com/console?redirected=true#/project/841318404499/apiui/api/maps_backend). The page header includes the "Google Developers Console" logo and the user email "imyday@gmail.com | Sign out".

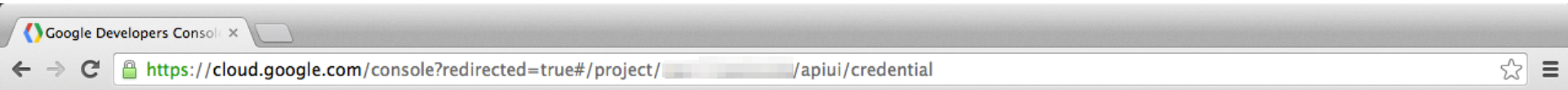
The main content area is titled "API Project" and features a sidebar on the left with navigation links: Overview, APIs & auth (with "APIs" selected), Credentials, Consent screen, Notification endpoints, Permissions, Settings, Support, Compute Engine, Cloud Storage, Cloud SQL, BigQuery, and Cloud Development.

The main content area displays "Google Maps API v3" with a green "ON" toggle switch, and links for "Quota" and "Reports". Below this, a description states: "The Google Maps API lets you embed Google Maps in your own web pages with Javascript. [Learn more](#)". A red rounded rectangle highlights the "ON" toggle and the description text.

At the bottom right, there are links for "Return to original console", "Send feedback", "Follow us", and "Privacy & Terms".

# Credentials: Public API access

## Get Google Maps API Key



### < API Project

Overview

APIs & auth

APIs

**Credentials**

Consent screen

Notification endpoints

Permissions

Settings

Support

Compute Engine

Cloud Storage

Cloud SQL

BigQuery [↗](#)

Cloud Development

### OAuth

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

[Learn more](#)

[CREATE NEW CLIENT ID](#)

# API key

### Public API access

Use of this key does not require any user action or consent, does not grant access to any account information, and is not used for authorization.

[Learn more](#)

[CREATE NEW KEY](#)

### Key for browser applications

API key	<b>AIzaSyBk9zQmepC</b>
Referers	Any referer allowed
Activation date	
Activated by	@gmail.com (you)

[Edit allowed referers](#)

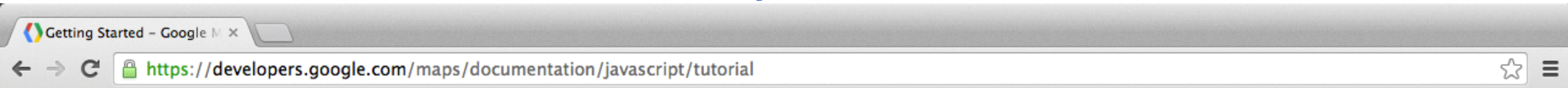
[Regenerate key](#)

[Delete](#)



# Google Map JavaScript API

## Hello, World



### Hello, World

The easiest way to start learning about the Google Maps API is to see a simple example. The following web page displays a map centered on Sydney, New South Wales, Australia:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map-canvas { height: 100% }
    </style>
    <script type="text/javascript"
      src="https://maps.googleapis.com/maps/api/js?key={API_KEY}&sensor=SET_TO_TRUE_OR_FALSE">
    </script>
    <script type="text/javascript">
      function initialize() {
        var mapOptions = {
          center: new google.maps.LatLng(-34.397, 150.644),
          zoom: 8
        };
        var map = new google.maps.Map(document.getElementById("map-canvas"),
          mapOptions);
      }
      google.maps.event.addDomListener(window, 'load', initialize);
    </script>
  </head>
  <body>
    <div id="map-canvas"/>
  </body>
</html>
```

[View example \(map-simple.html\)](#)

Even for this simple example, there are a few things to note:  
<https://developers.google.com/maps/documentation/javascript/tutorial>

# Google Map JavaScript API

## Hello, World

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map-canvas { height: 100% }
    </style>
    <script type="text/javascript"
      src="https://maps.googleapis.com/maps/api/js?key=API_KEY&sensor=SET_TO_TRUE_OR_FALSE">
    </script>
    <script type="text/javascript">
      function initialize() {
        var mapOptions = {
          center: new google.maps.LatLng(-34.397, 150.644),
          zoom: 8
        };
        var map = new google.maps.Map(document.getElementById("map-canvas"),
          mapOptions);
      }
      google.maps.event.addDomListener(window, 'load', initialize);
    </script>
  </head>
  <body>
    <div id="map-canvas"/>
  </body>
</html>
```

**API key**

API\_KEY

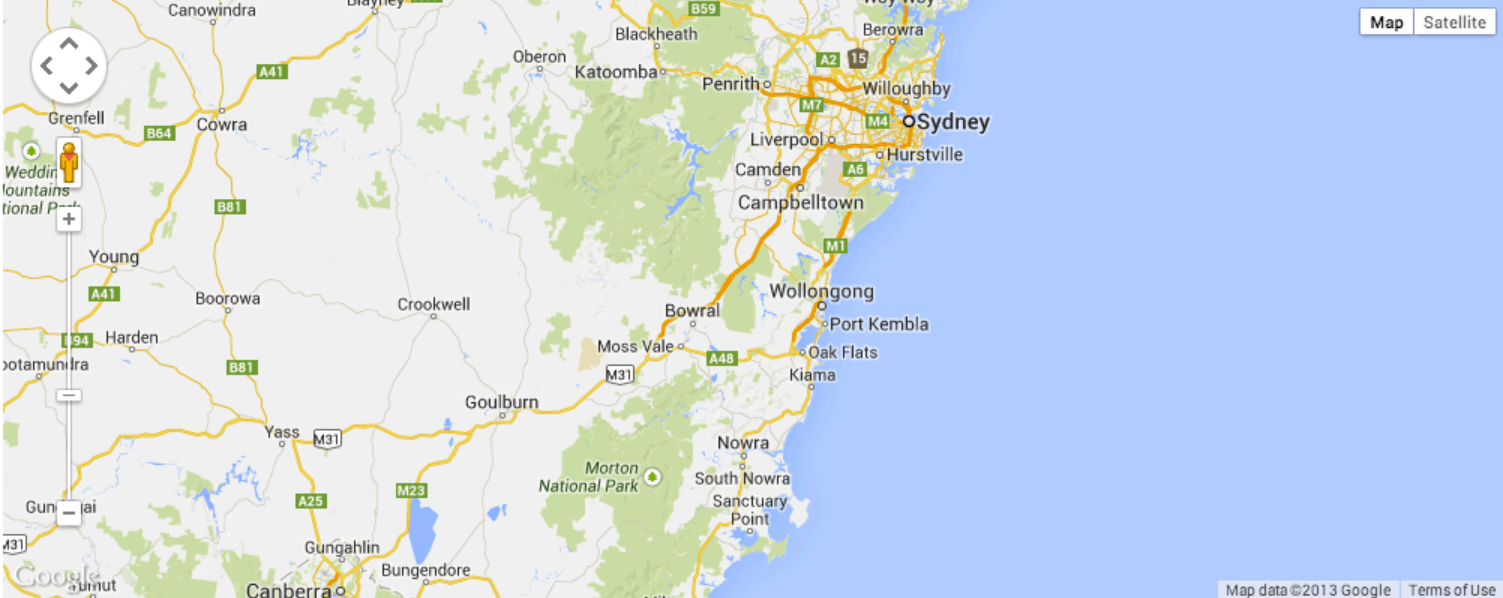
# Google Maps JavaScript API: Simple Map

Simple Map - Google Maps

https://developers.google.com/maps/documentation/javascript/examples/map-simple

- Developer's Guide
  - API Reference
  - Code Samples
    - Basics
      - Simple map**
      - Visual Refresh
      - Showing pixel and tile coordinates
      - Geolocation
      - Localizing the map
      - Right-to-left languages
      - Asynchronous loading
      - Custom map projections
    - Events
    - Controls
    - Styles
    - Drawing on the Map
    - Layers
    - MapTypes
    - Services
    - Libraries
    - Demo Gallery
    - More Resources

## Simple Map



View this example [full screen](#).

JavaScript JavaScript + HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Map</title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <style>
      html, body, #map-canvas {
        height: 100%;
```

## Google Maps JavaScript API Example JavaScript + HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Map</title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    html, body, #map-canvas {
      height: 100%;
      margin: 0px;
      padding: 0px
    }
  </style>
  <script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></script>
  <script>
var map;
function initialize() {
  var mapOptions = {
    zoom: 8,
    center: new google.maps.LatLng(-34.397, 150.644)
  };
  map = new google.maps.Map(document.getElementById('map-canvas'),
    mapOptions);
}

google.maps.event.addDomListener(window, 'load', initialize);

  </script>
</head>
<body>
  <div id="map-canvas"></div>
</body>
</html>
```

```
<div style="position:absolute; height:100%; width:100%;">  
  <div id="map-canvas"></div>  
</div>
```



<http://mail.tku.edu.tw/myday/app/map.html>

# Google Maps JavaScript API

```
<style>
```

```
  #map-canvas {  
    height: 100%;  
    margin: 0px;  
    padding: 0px  
  }
```

```
</style>
```

```
<script>
```

```
  function initialize() {  
    var mapOptions = {  
      zoom: 15,  
      center: new google.maps.LatLng(25.174738, 121.450381)  
    };
```

```
    var map = new google.maps.Map(document.getElementById('map-canvas'),  
      mapOptions);  
  }
```

```
  function loadScript() {  
    var script = document.createElement('script');  
    script.type = 'text/javascript';  
    script.src = 'https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true&callback=initialize';  
    document.body.appendChild(script);  
  }
```

```
  window.onload = loadScript;
```

```
</script>
```

```

<!DOCTYPE html>
<html>
<head>
<title>Google Maps</title>
<meta charset=utf-8 />
<meta name="viewport" content="width=device-width, initial-scale=1" />

<script src="js/jquery.js"></script>
<link type="text/css" href="css/jquery.mobile-1.3.2.min.css" rel="stylesheet" />
<script type="text/javascript" src="js/jquery.mobile-1.3.2.min.js"></script>

<!--PhoneGap-->
<link rel="stylesheet" type="text/css" href="css/index.css" />
<!--
<script type="text/javascript" src="phonegap.js"></script>
-->
<script type="text/javascript" src="js/index.js"></script>
<!--/PhoneGap-->
<script type="text/javascript">
    app.initialize();
</script>

<style>
#map-canvas {
    height: 100%;
    margin: 0px;
    padding: 0px
}
</style>

<script>
function initialize() {
    var mapOptions = {
        zoom: 15,
        center: new google.maps.LatLng(25.174738, 121.450381)
    };

    var map = new google.maps.Map(document.getElementById('map-canvas'),
        mapOptions);
}

function loadScript() {
    var script = document.createElement('script');
    script.type = 'text/javascript';
    script.src = 'https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true&callback=initialize';
    document.body.appendChild(script);
}

window.onload = loadScript;
</script>

</head>

```

```

<body>

<div data-role="page" id="googlemaps" data-theme="b">
<div data-role="header" data-position="inline" data-theme="b">
    <a data-icon="back" data-rel="back" back-btn="true">Back</a>
    <h1>Google Maps</h1>
    <a href="index.html#MyHome" rel="external" data-icon="home">Home</a>
</div><!-- /header -->

<div style="position:absolute; height:100%; width:100%;">
<div id="map-canvas"></div>
</div>

<div data-role="footer" data-position="fixed" data-theme="b">
<div data-role="navbar">
    <ul>
        <li><a href="index.html#MyHome" rel="external" class="ui-btn-active ui-state-persist"
data-transition="fade" data-icon="home">Home</a></li>
        <li><a href="index.html#Research" rel="external" class="ui-btn-active ui-state-persist"
data-transition="fade" data-icon="star">Research</a></li>
        <li><a href="index.html#Teaching" rel="external" class="ui-btn-active ui-state-persist"
data-transition="fade" data-icon="check">Teaching</a></li>
        <li><a href="index.html#More" rel="external" class="ui-btn-active ui-state-persist"
data-transition="fade" data-icon="bars">More</a></li>
        <li><a href="index.html#About" rel="external" class="ui-btn-active ui-state-persist"
data-transition="fade" data-icon="grid">About</a></li>
    </ul>
</div>
</div><!-- /footer -->
</div><!-- /page Google Map-->
</body>
</html>

```

# map.html

# Google Maps JavaScript API + jQuery Mobile



<http://mail.tku.edu.tw/myday/app/map.html>



# Summary

- **Google App Engine**



- **Google Cloud Platform**



- **Google Cloud Datastore**



- **Google Firebase**



Firestore

- **Google Maps API**

# References

- Google Cloud Platform,  
<https://cloud.google.com/>
- Google App Engine,  
<https://cloud.google.com/appengine/>
- Google Cloud Datastore,  
<https://cloud.google.com/datastore/>
- Google Cloud Endpoints,  
<https://cloud.google.com/endpoints/>
- Google Firebase  
<https://firebase.google.com/>
- Google Maps API,  
<https://developers.google.com/maps/>
- Google Maps JavaScript API v3 Tutorial,  
<https://developers.google.com/maps/documentation/javascript/tutorial>
- Google Maps API Tutorial,  
<http://www.w3schools.com/googleAPI/>