



Social Media Apps Programming

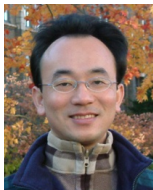
Google App Engine

1061SMAP11

TLMXM1A (8648) (M2143) (Fall 2017)

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

Fri 8,9 (15:10-17:00) B206



Min-Yuh Day, Ph.D.

Assistant Professor

Department of Information Management

Tamkang University

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

2017-12-15



Course Schedule (1/2)



Tamkang
University

Week	Date	Subject/Topics
1	2017/09/22	Course Orientation and Introduction to Social Media and Mobile Apps Programming
2	2017/09/29	Introduction to Android / iOS Apps Programming
3	2017/10/06	Developing Android Native Apps with Java (Android Studio)
4	2017/10/13	Developing iPhone / iPad Native Apps with Swift (XCode)
5	2017/10/20	Mobile Apps using HTML5/CSS3/JavaScript
6	2017/10/27	jQuery Mobile
7	2017/11/03	Create Hybrid Apps with Phonegap
8	2017/11/10	jQuery Mobile/Phonegap
9	2017/11/17	jQuery Mobile/Phonegap

Course Schedule (2/2)



**Tamkang
University**

Week	Date	Subject/Topics
10	2017/11/24	Midterm Project Report
11	2017/12/01	Case Study on Social Media Apps Programming and Marketing in Google Play and App Store
12	2017/12/08	Google Cloud Platform
13	2017/12/15	Google App Engine
14	2017/12/22	Google Map API
15	2017/12/29	Facebook API (Facebook JavaScript SDK) (Integrate Facebook with iOS/Android Apps)
16	2018/01/05	Twitter API
17	2018/01/12	Final Project Presentation
18	2018/01/19	Final Exam Week (Final Project Presentation)

Outline

- **Google App Engine**



- **Google Cloud Platform**



- **Google Cloud Datastore**



- **Google Firebase**

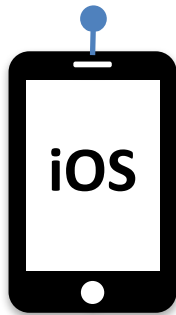


Firebase



**App
Backend**

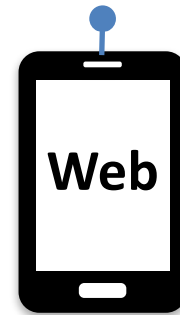
App Frontend



iOS



Android



Web

Mobile Apps Backend on Google Cloud

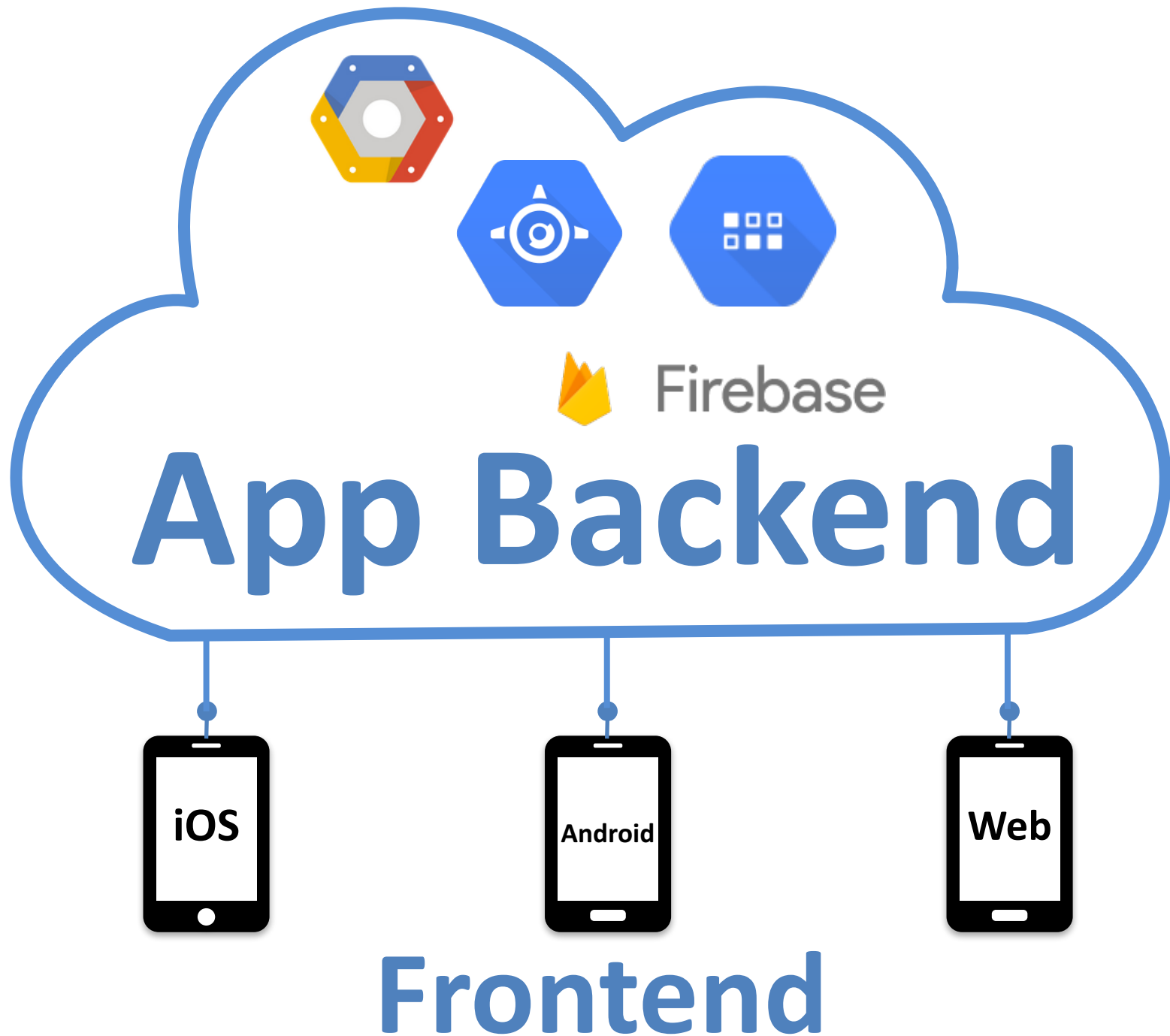


Firebase



App

Backend





Google Cloud Platform

**Hosting +
Compute**

Storage

Big Data

Services



Google Cloud Platform

Compute



App Engine



Compute Engine



Container Engine

Storage



Cloud Datastore



Cloud SQL



Cloud Storage

Big Data



Big Query

Services



Cloud Endpoints



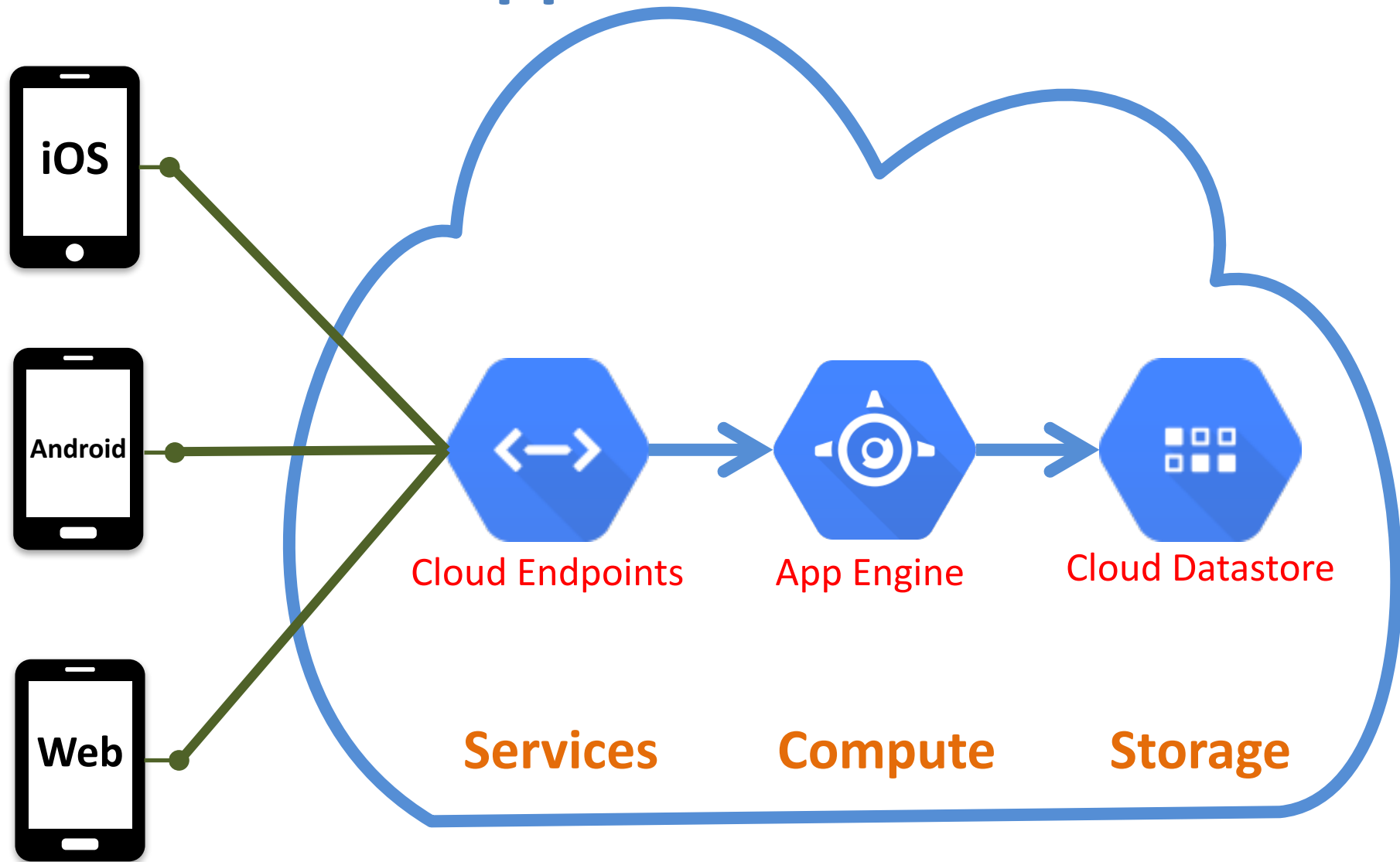
Translate API



Prediction API

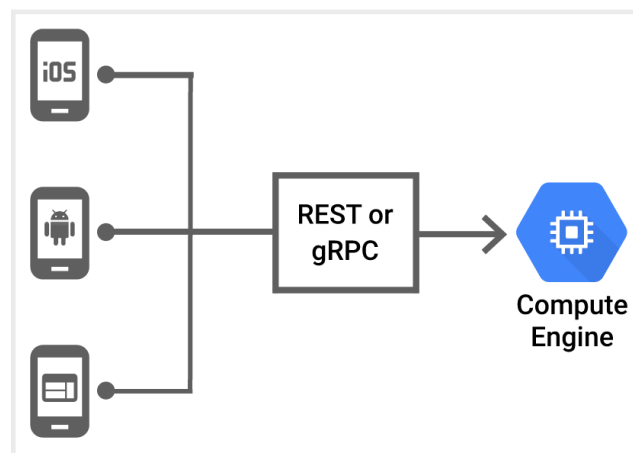
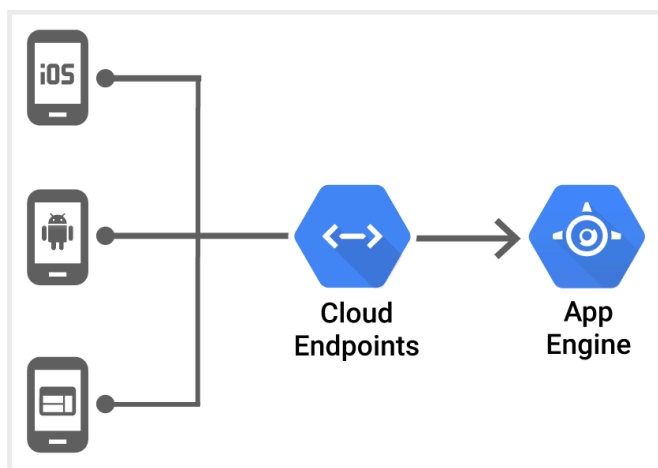
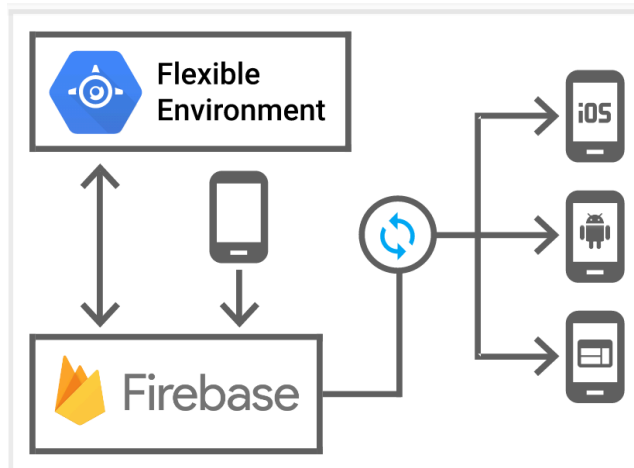
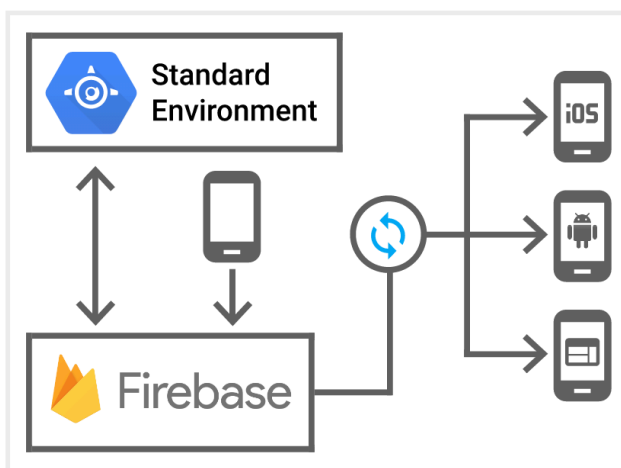
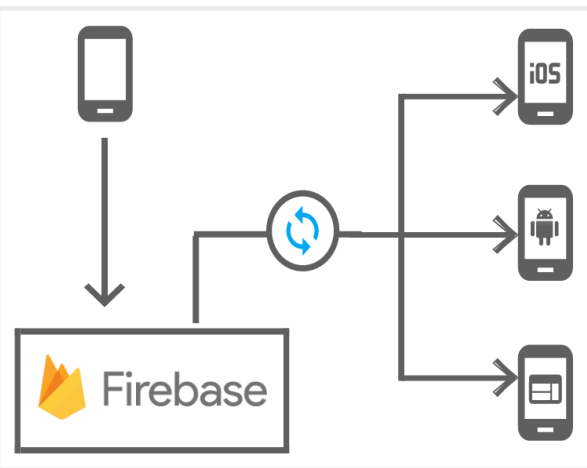


Mobile App Backend Services



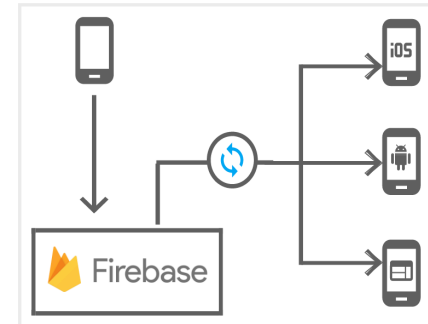
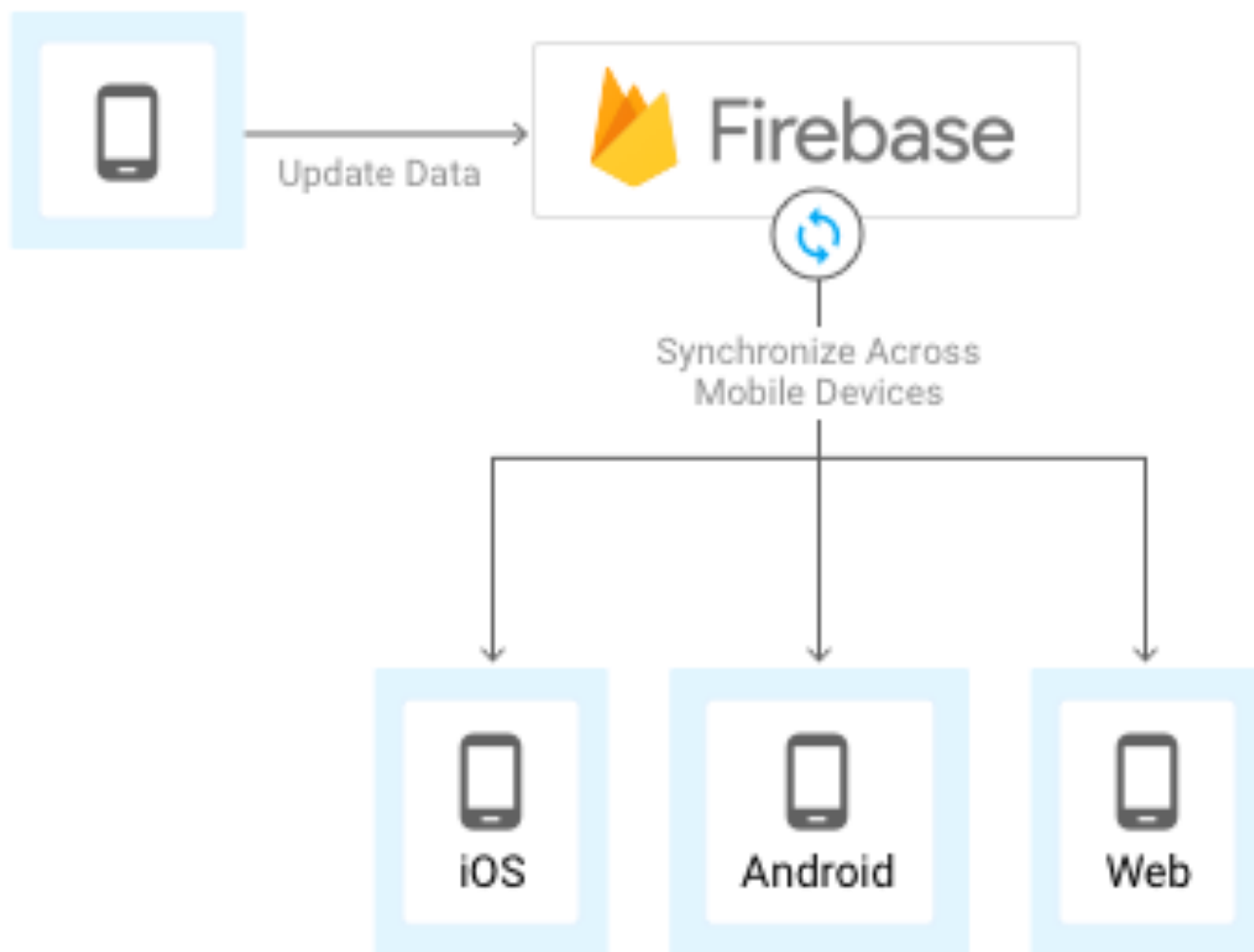


Mobile App Backend Services



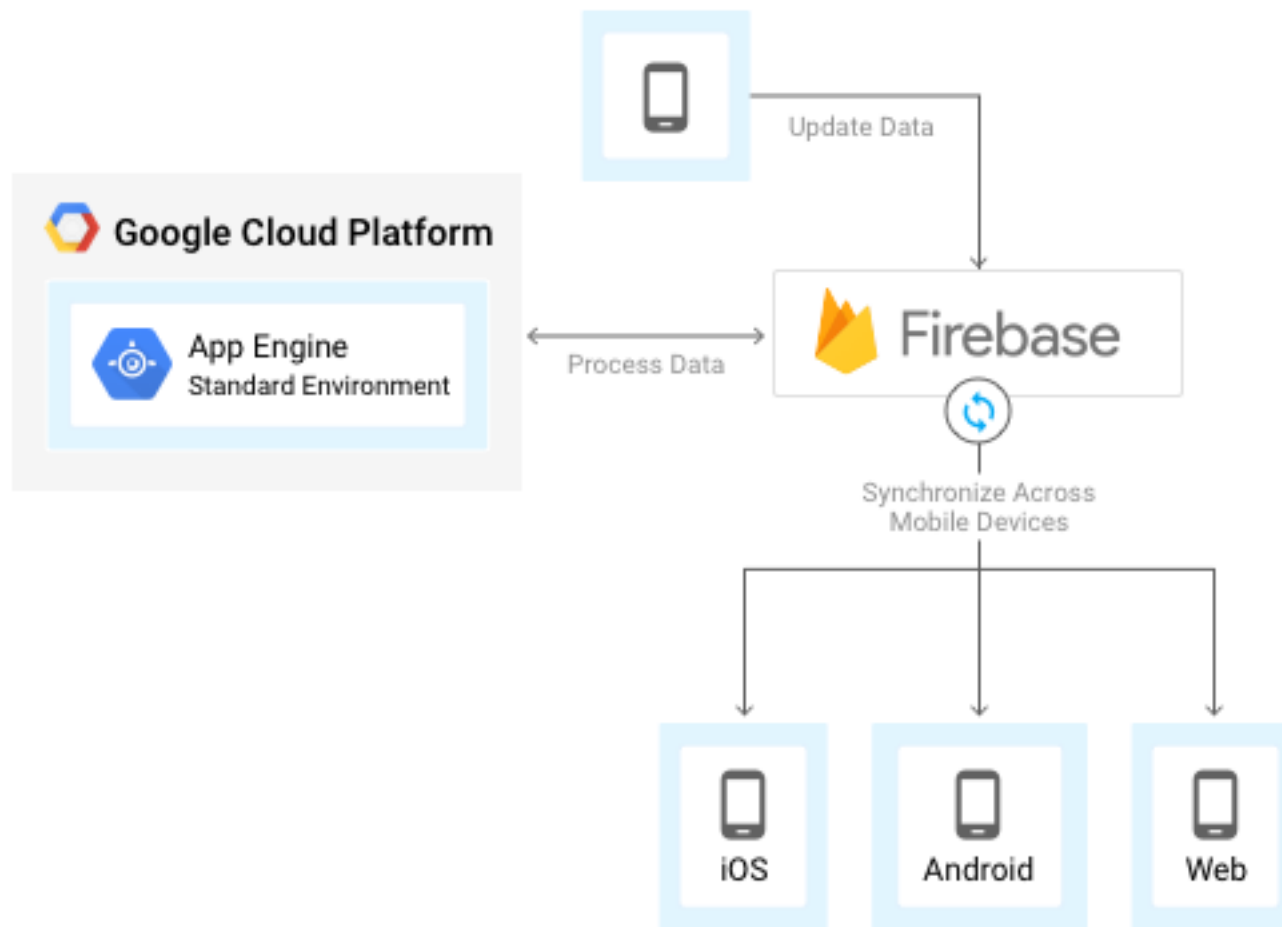
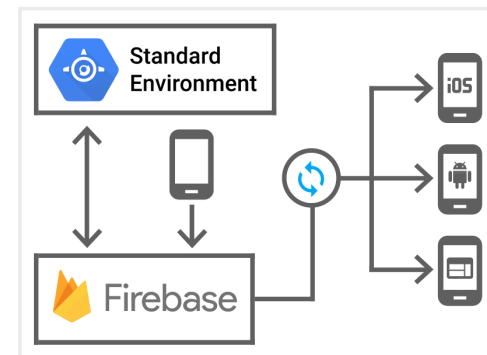


Firestore



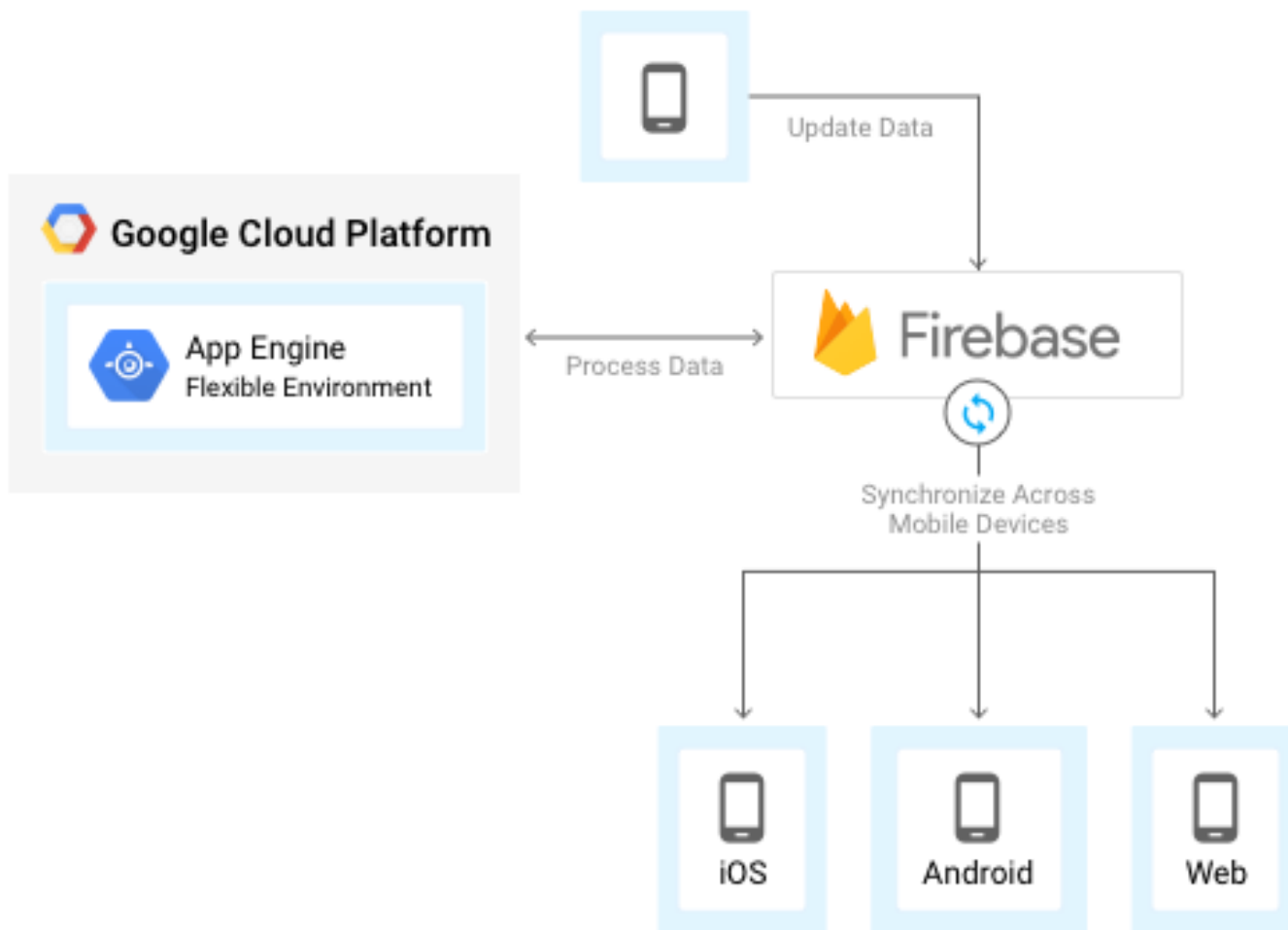
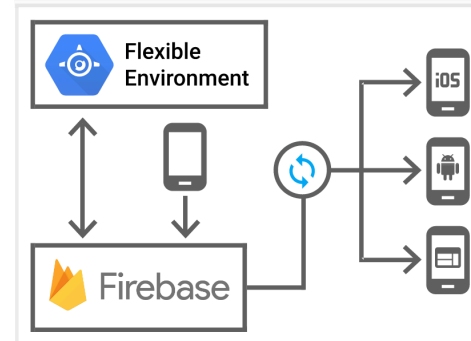


Firebase and Google App Engine standard environment



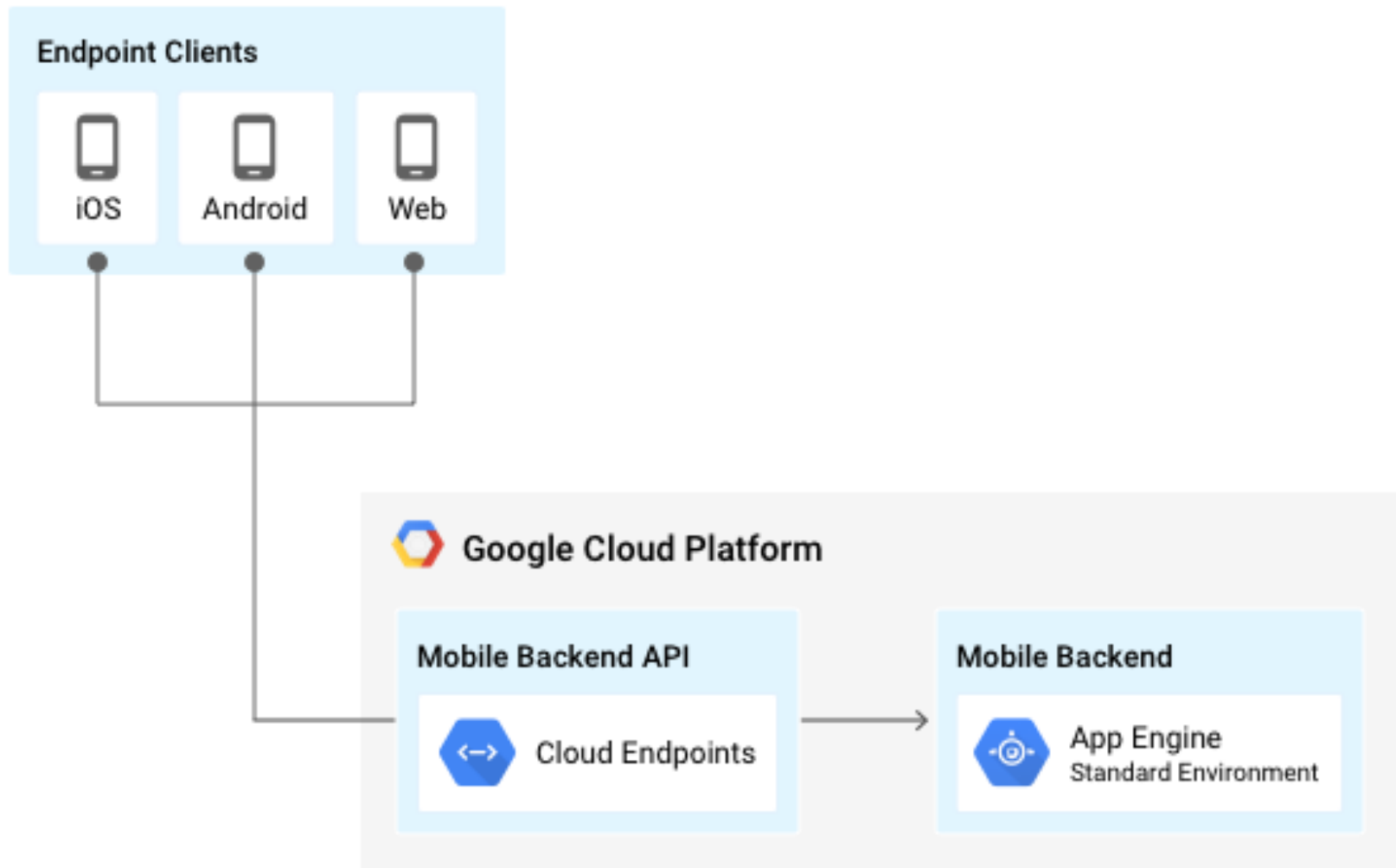
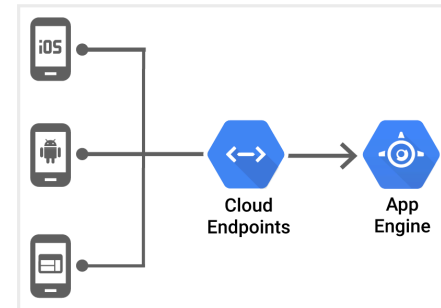


Firestore and App Engine flexible environment



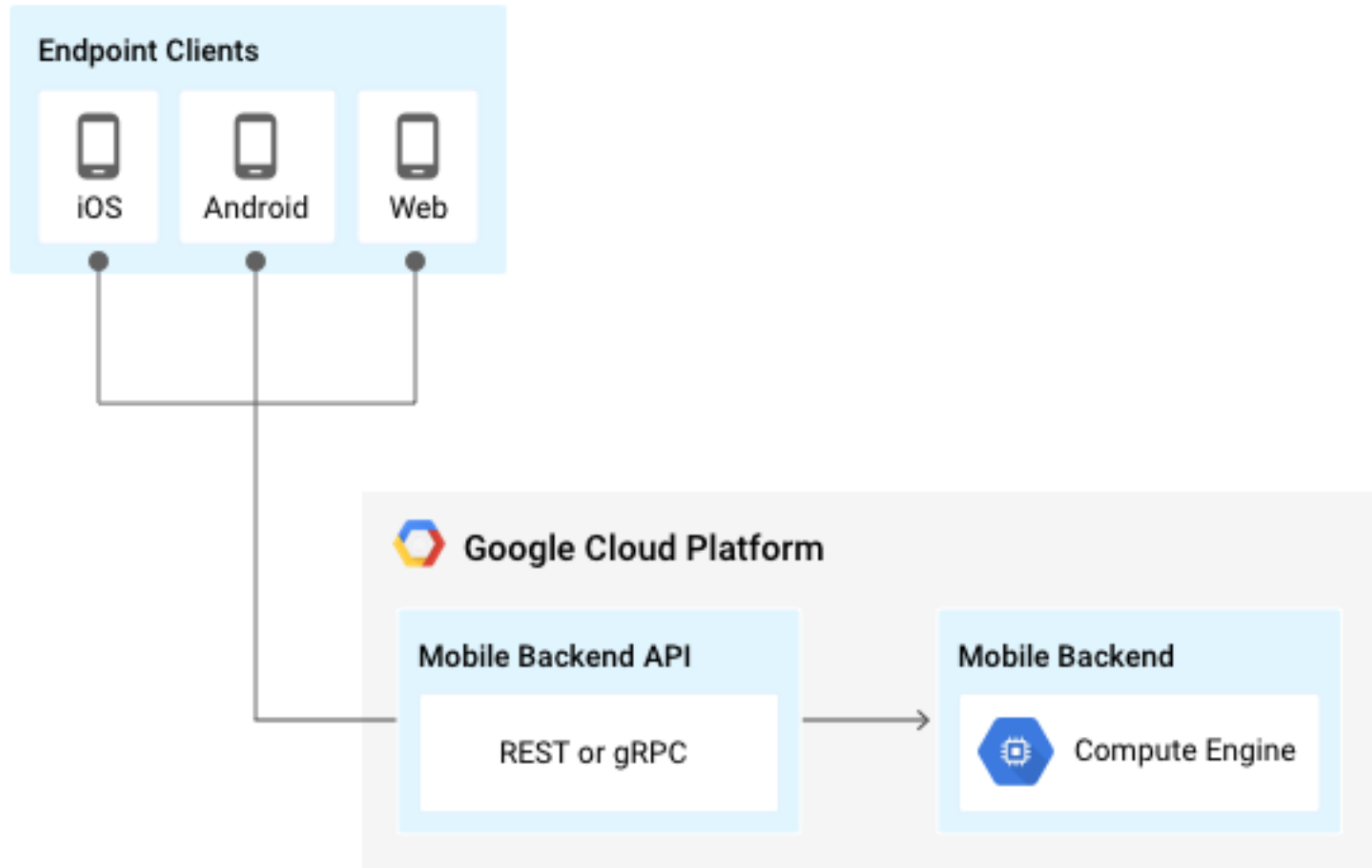
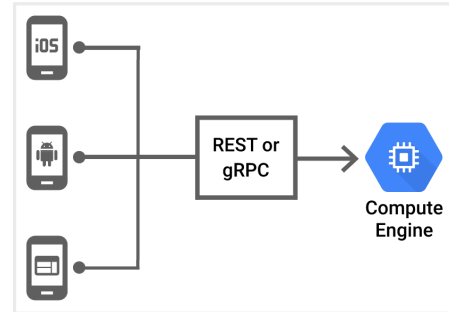


App Engine and Cloud Endpoints





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>
```

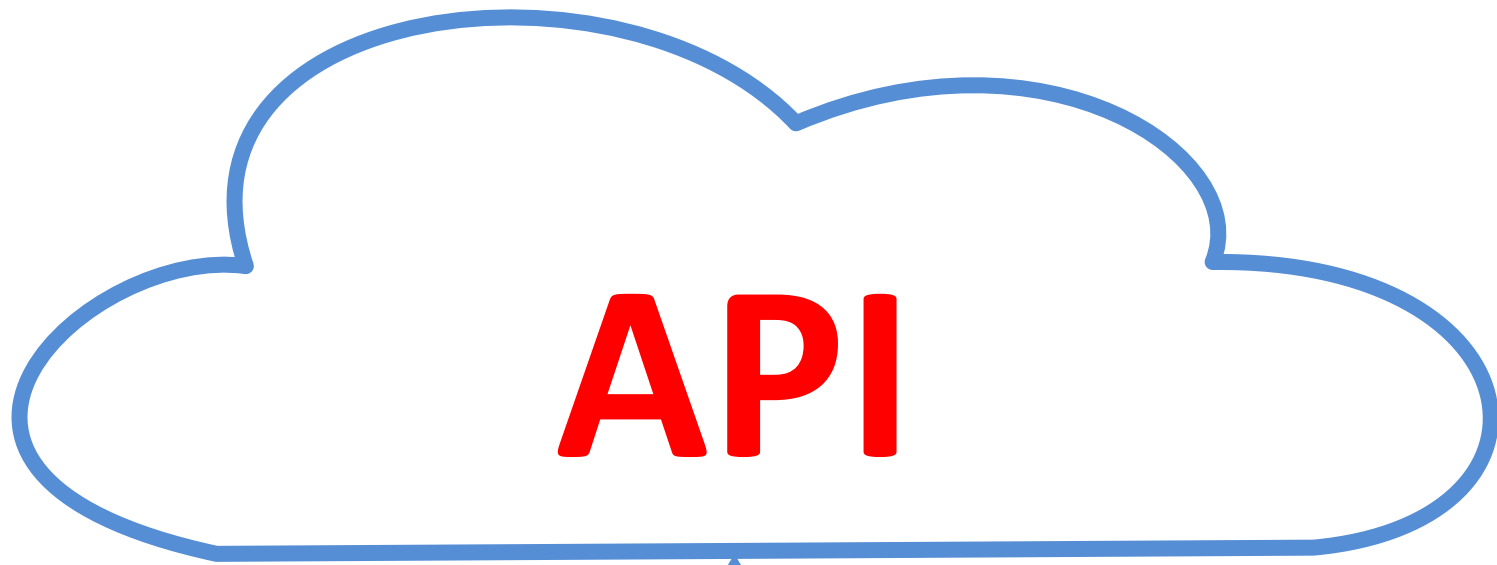
JSON vs. XML

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

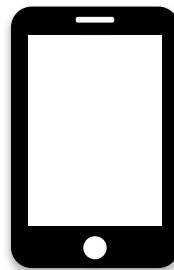
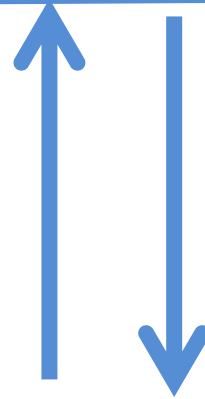
```
<employees>  
  <employee>  
    <firstName>John</firstName> <lastName>Doe</lastName>  
  </employee>  
  <employee>  
    <firstName>Anna</firstName> <lastName>Smith</lastName>  
  </employee>  
  <employee>  
    <firstName>Peter</firstName> <lastName>Jones</lastName>  
  </employee>  
</employees>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

API

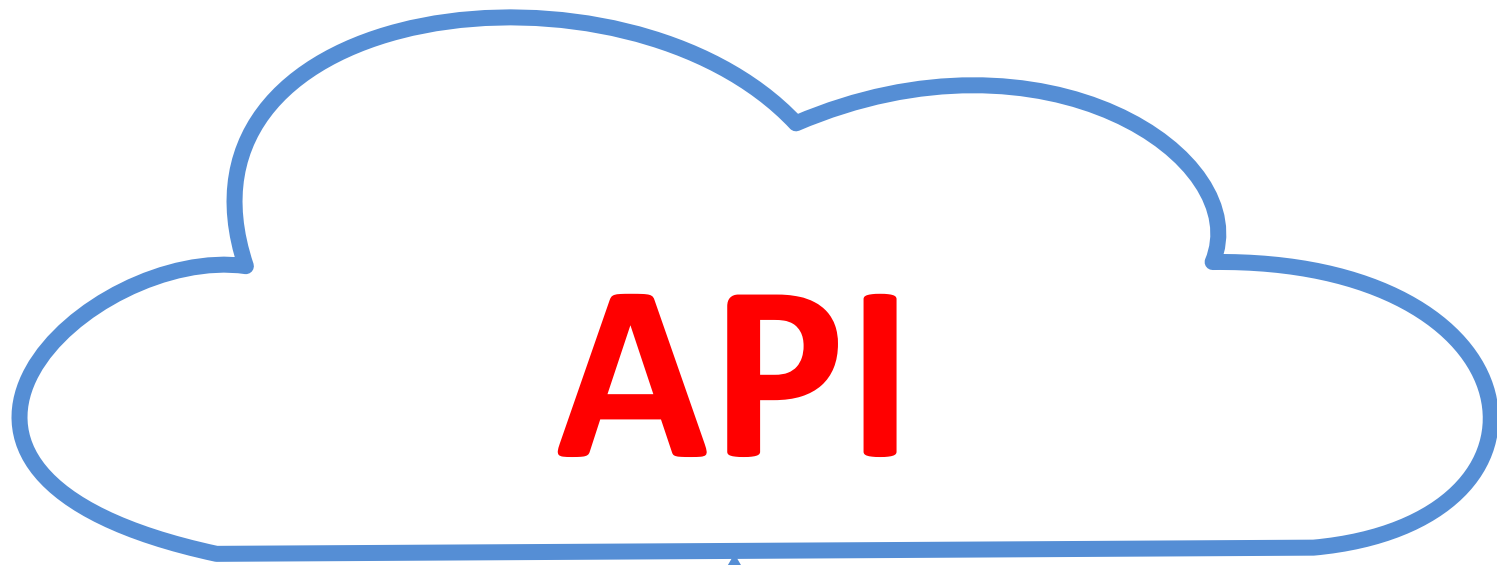
Application
Programming
Interface



API

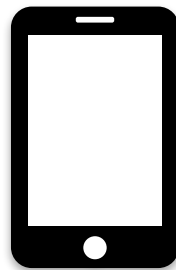


Your App

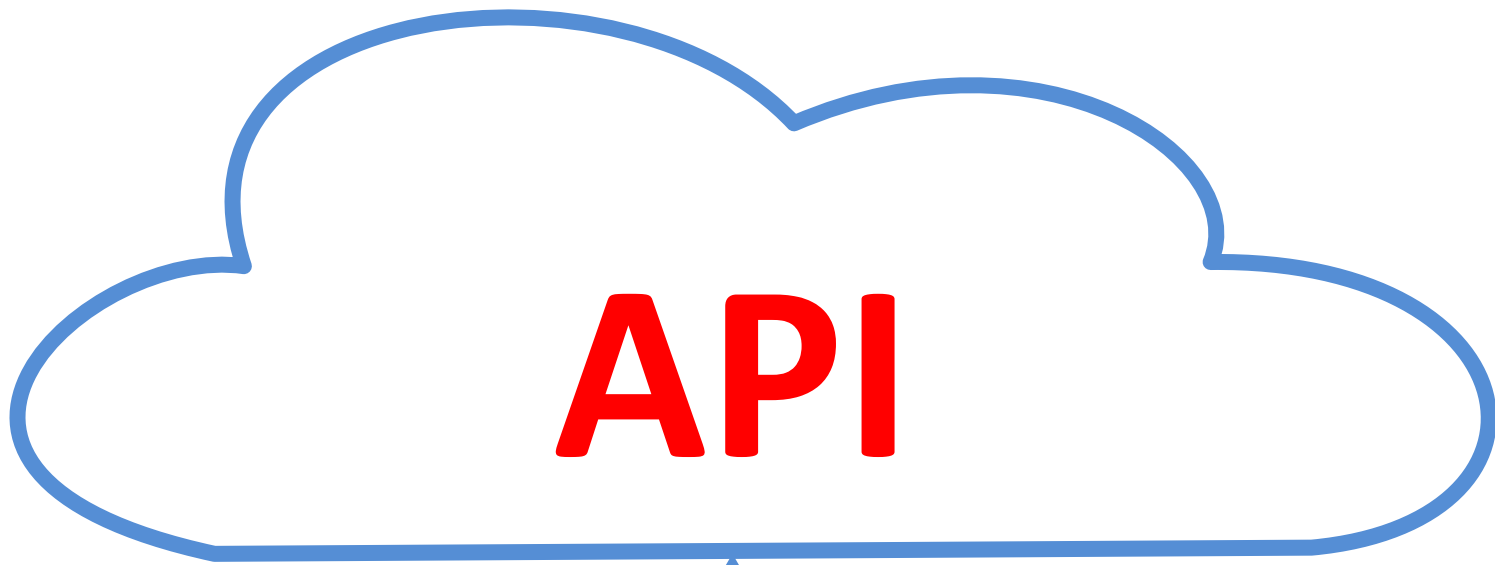


Request

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



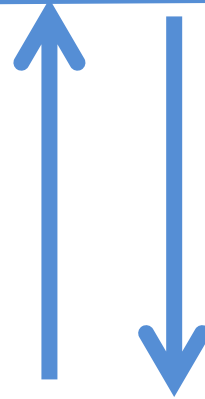
Your App



API

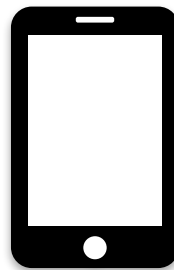
Request

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



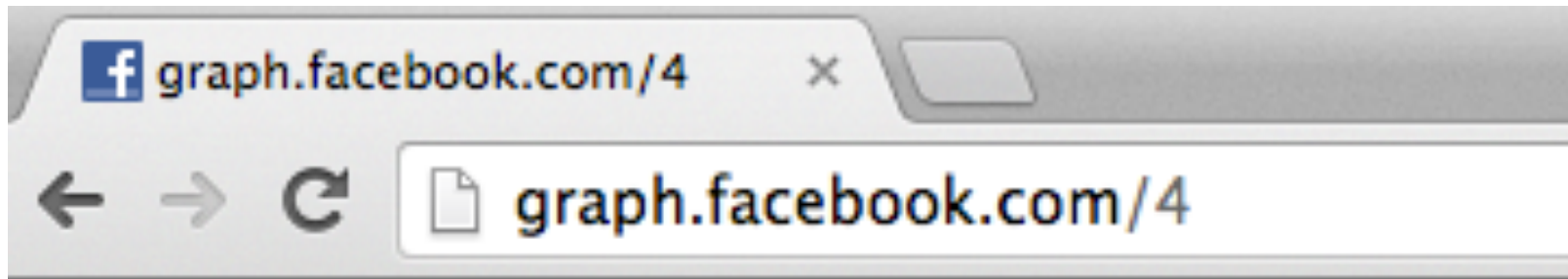
Response

```
{  
  "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>



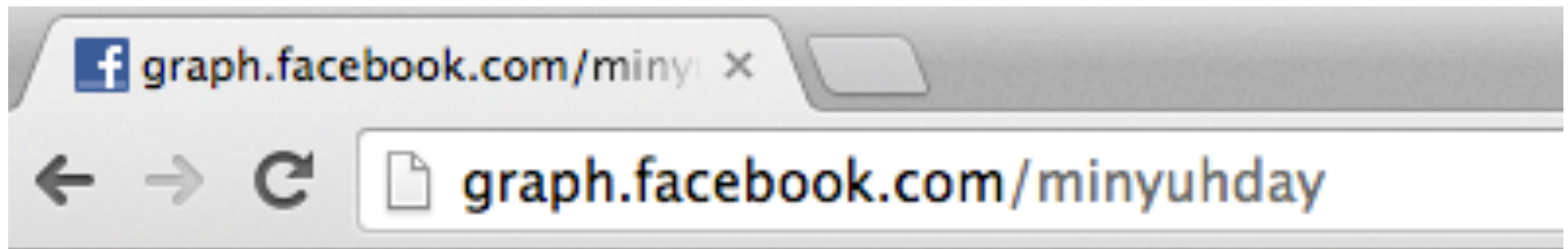
```
{
  "id": "4",
  "name": "Mark Zuckerberg",
  "first_name": "Mark",
  "last_name": "Zuckerberg",
  "link": "http://www.facebook.com/zuck",
  "username": "zuck",
  "gender": "male",
  "locale": "en_US"
}
```

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

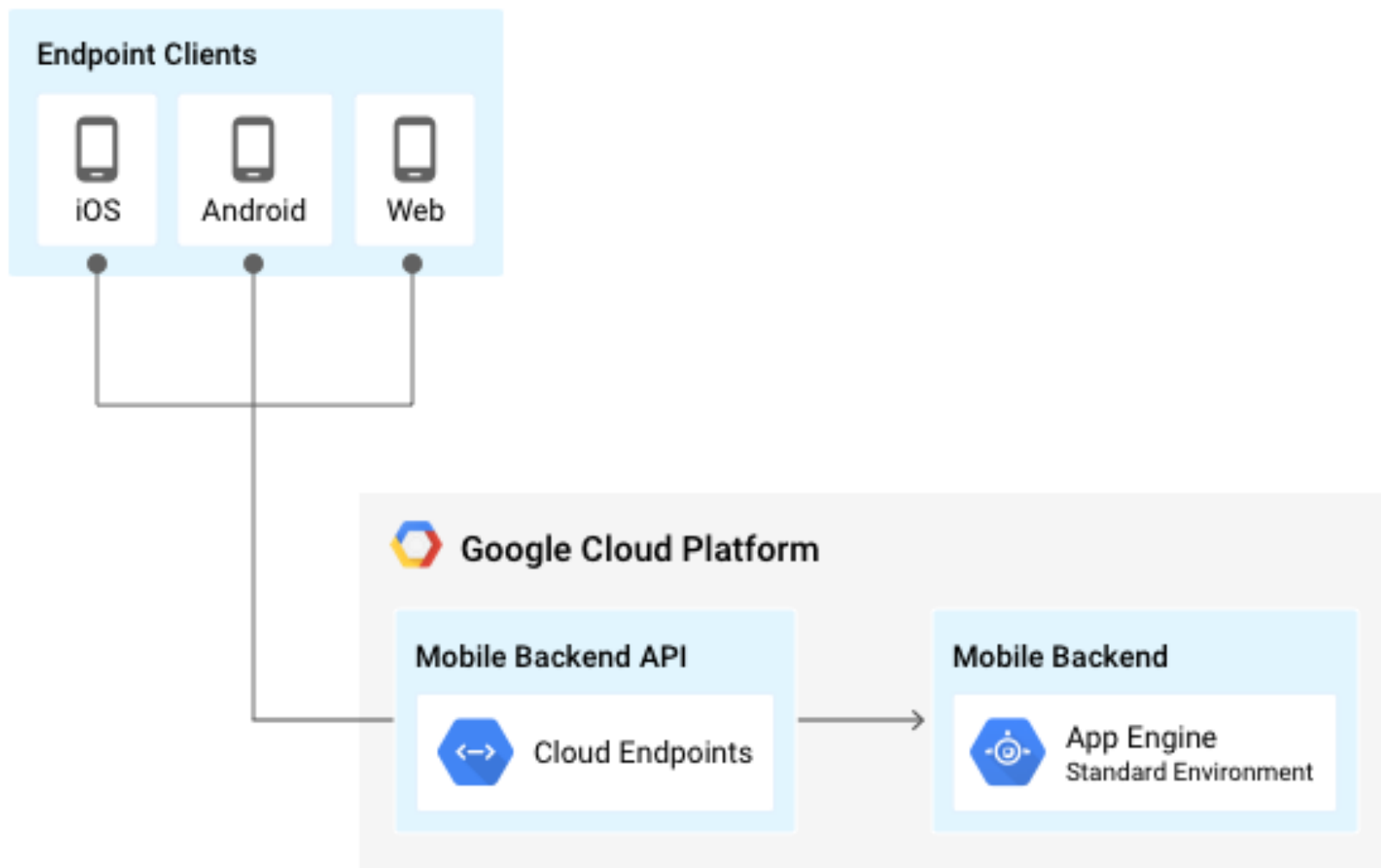
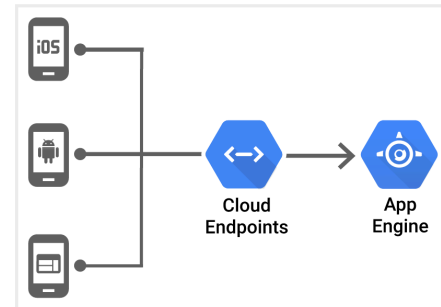
JSON

```
{  
  "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]
```

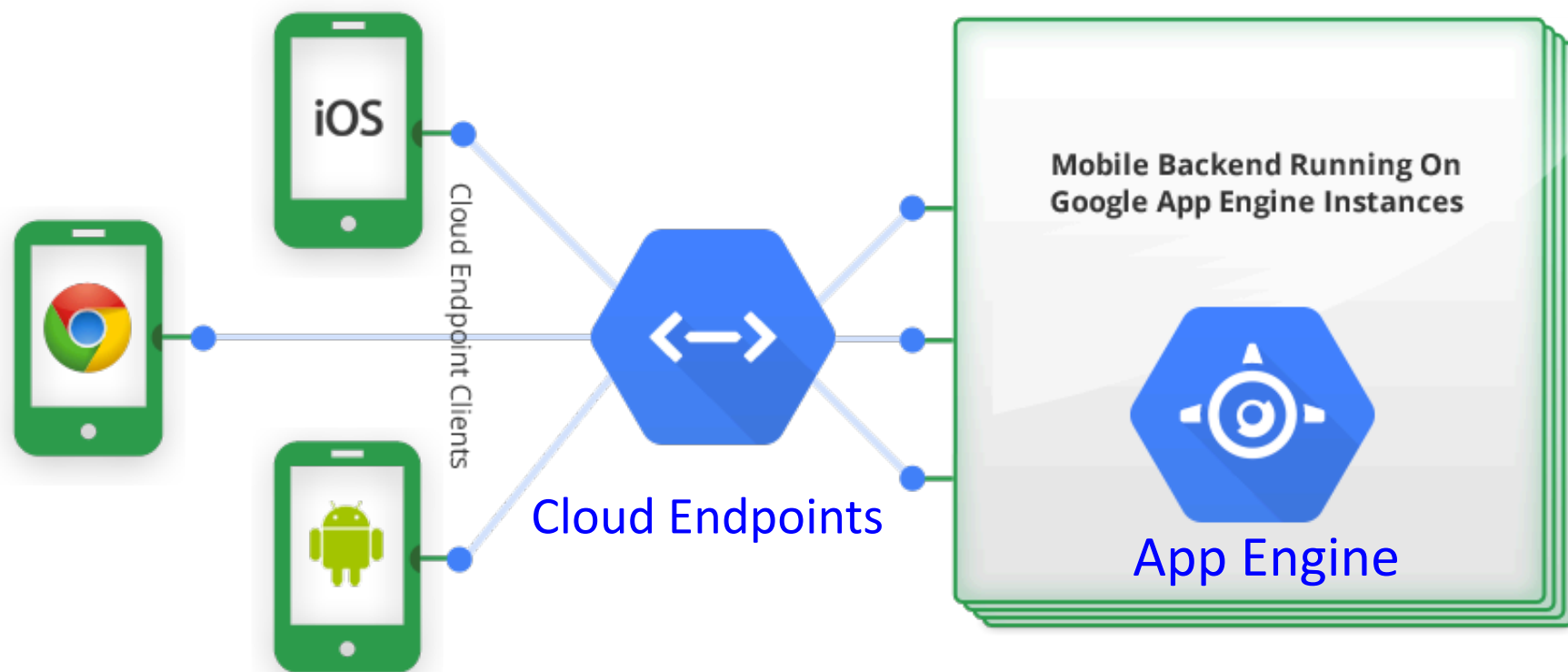


App Engine and Cloud Endpoints





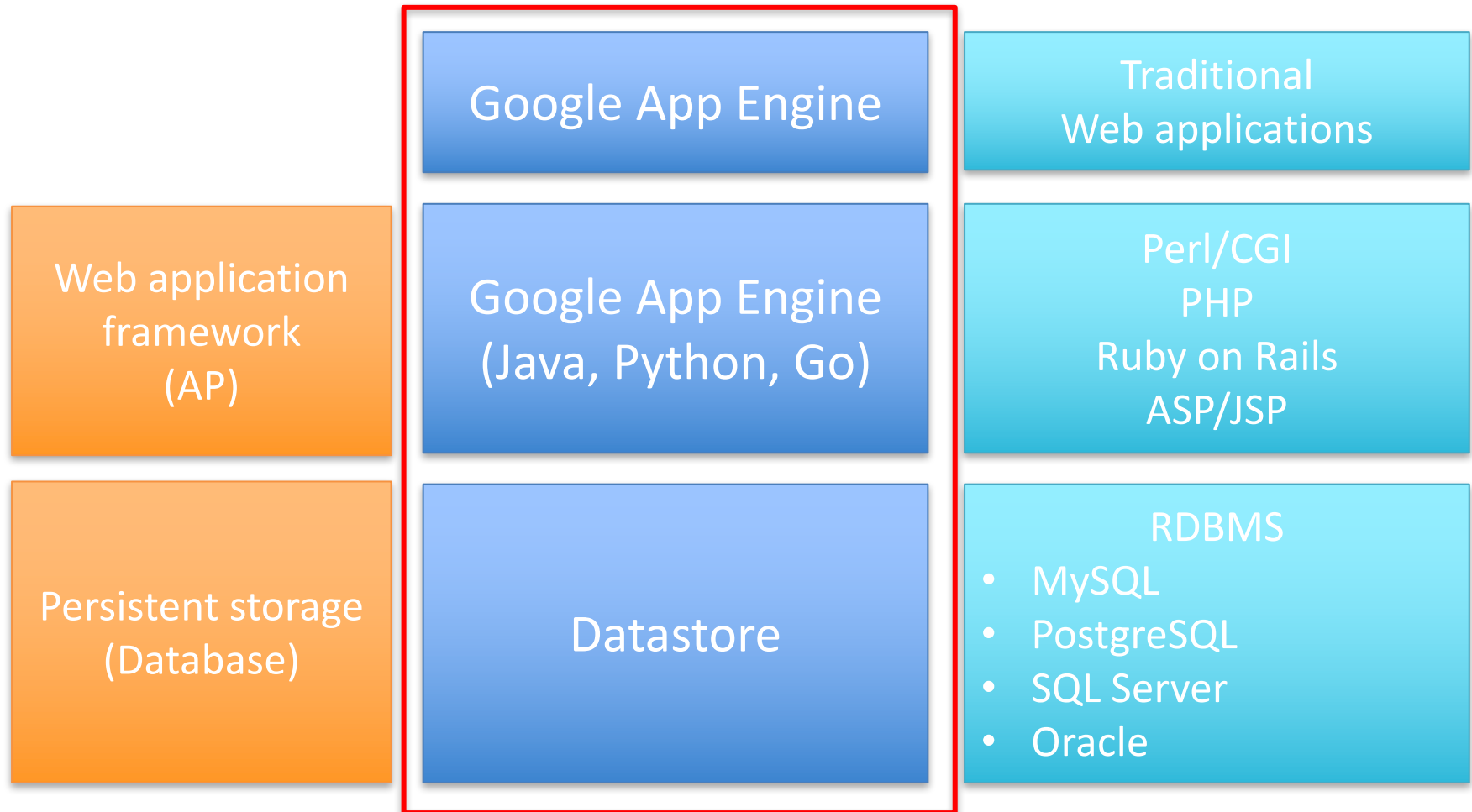
Mobile Apps Backend on Google App Engine




Google Cloud Endpoints Architecture

Google App Engine, Google Cloud Datastore

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

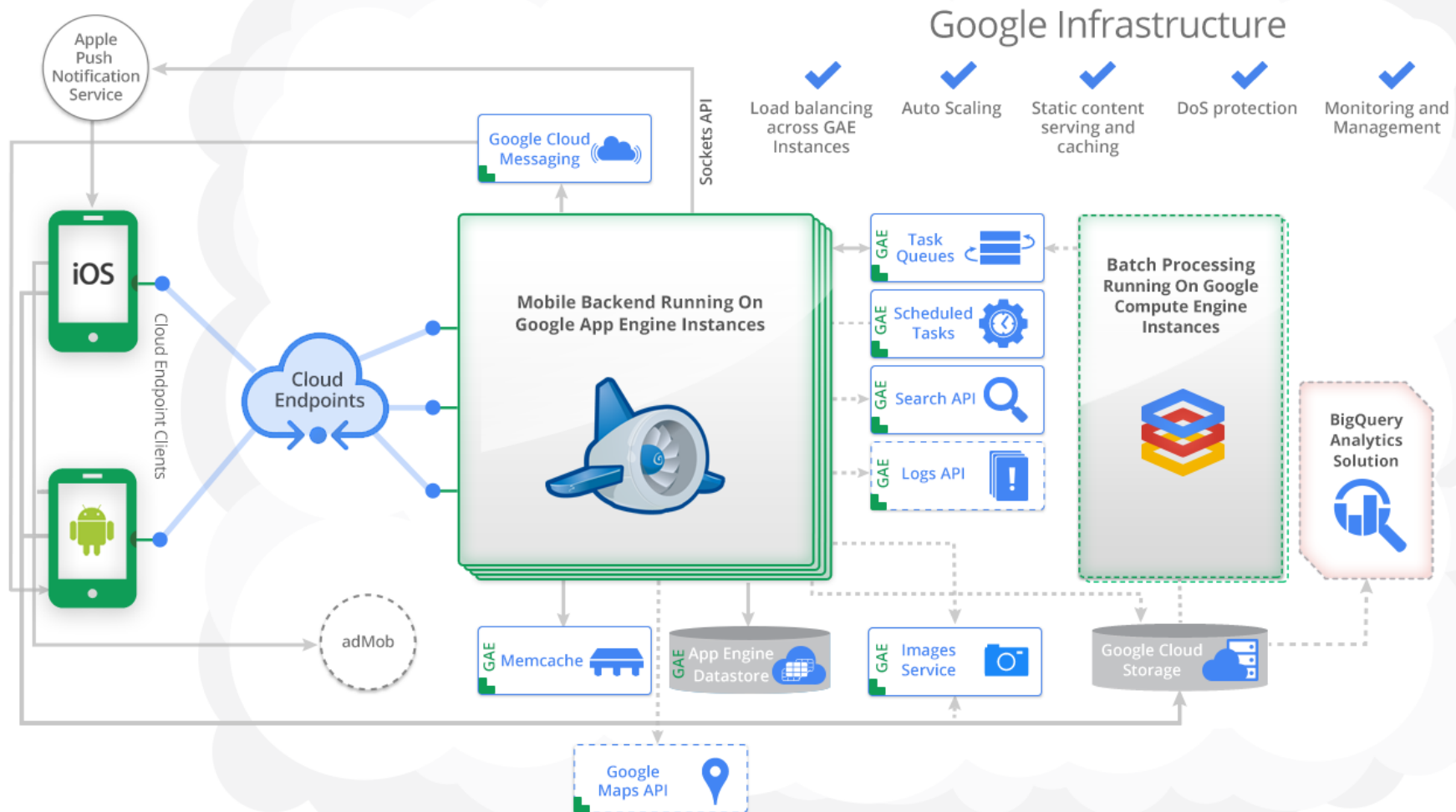


Mobile Solutions on the Google Cloud Platform

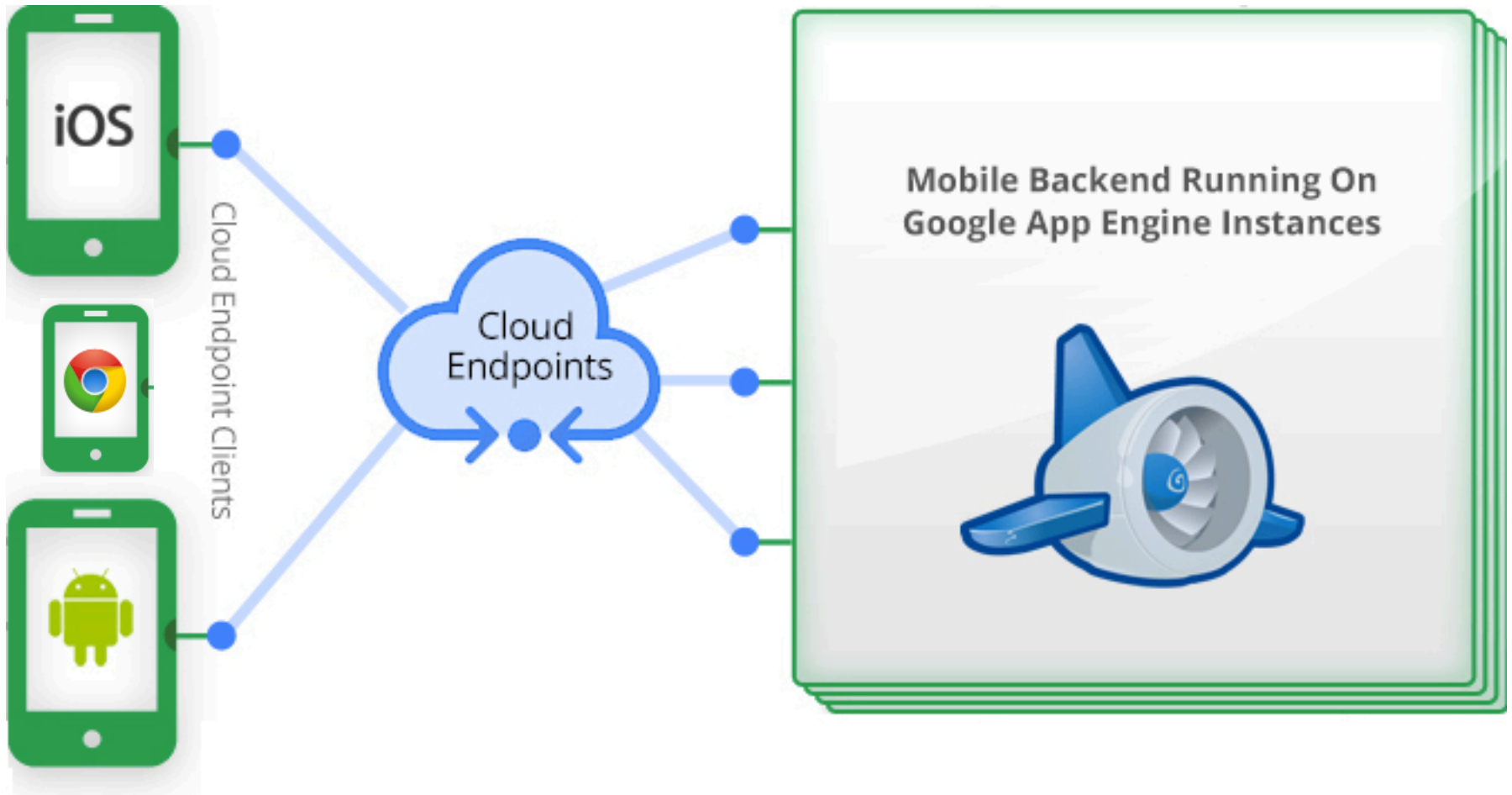
 Your Application Code running on Google App Engine (GAE), Google Compute Engine (GCE), and Client Devices

 Google Cloud Platform Services  Optional components

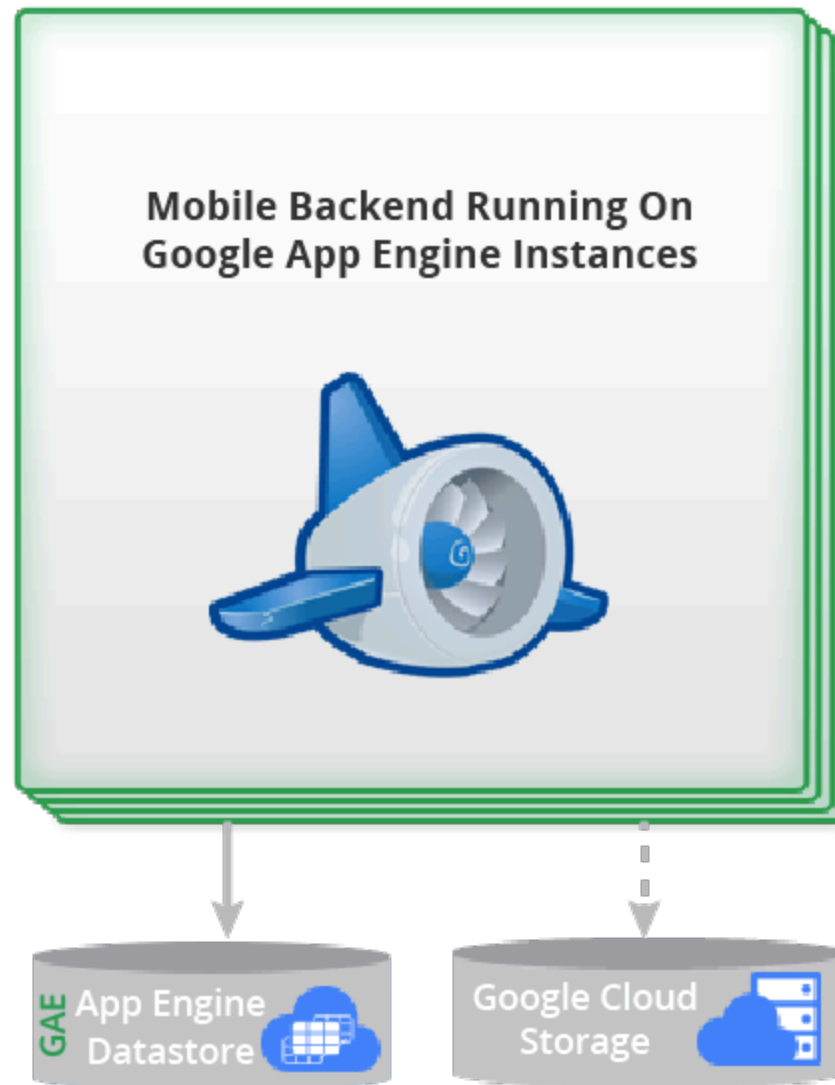
 Capabilities Included



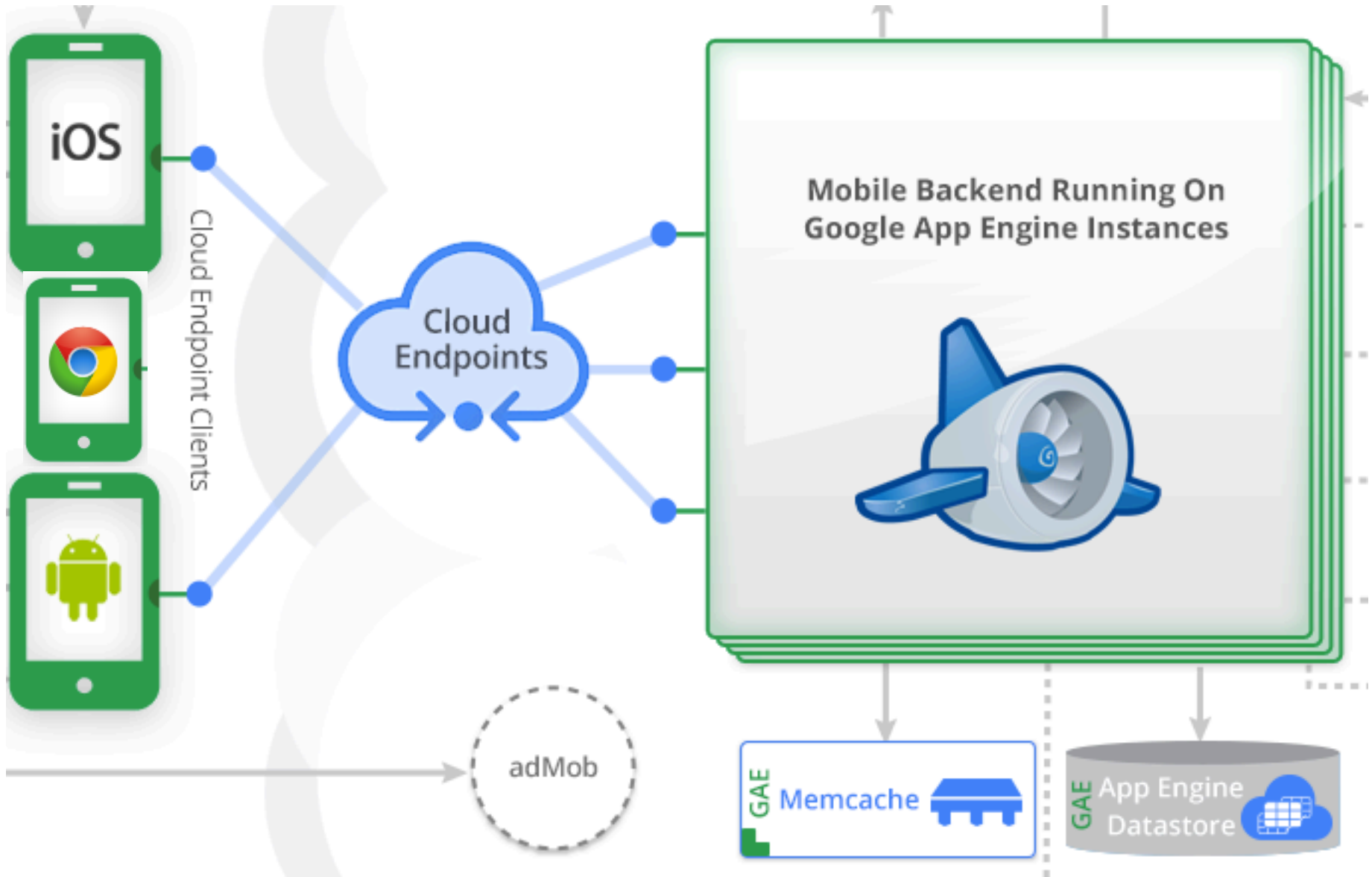
Mobile App Solutions Architecture



Storing data



Optimizing data access with Memcache





Google App Engine Platform as a Service (PaaS)

build and run applications on
Google's infrastructure



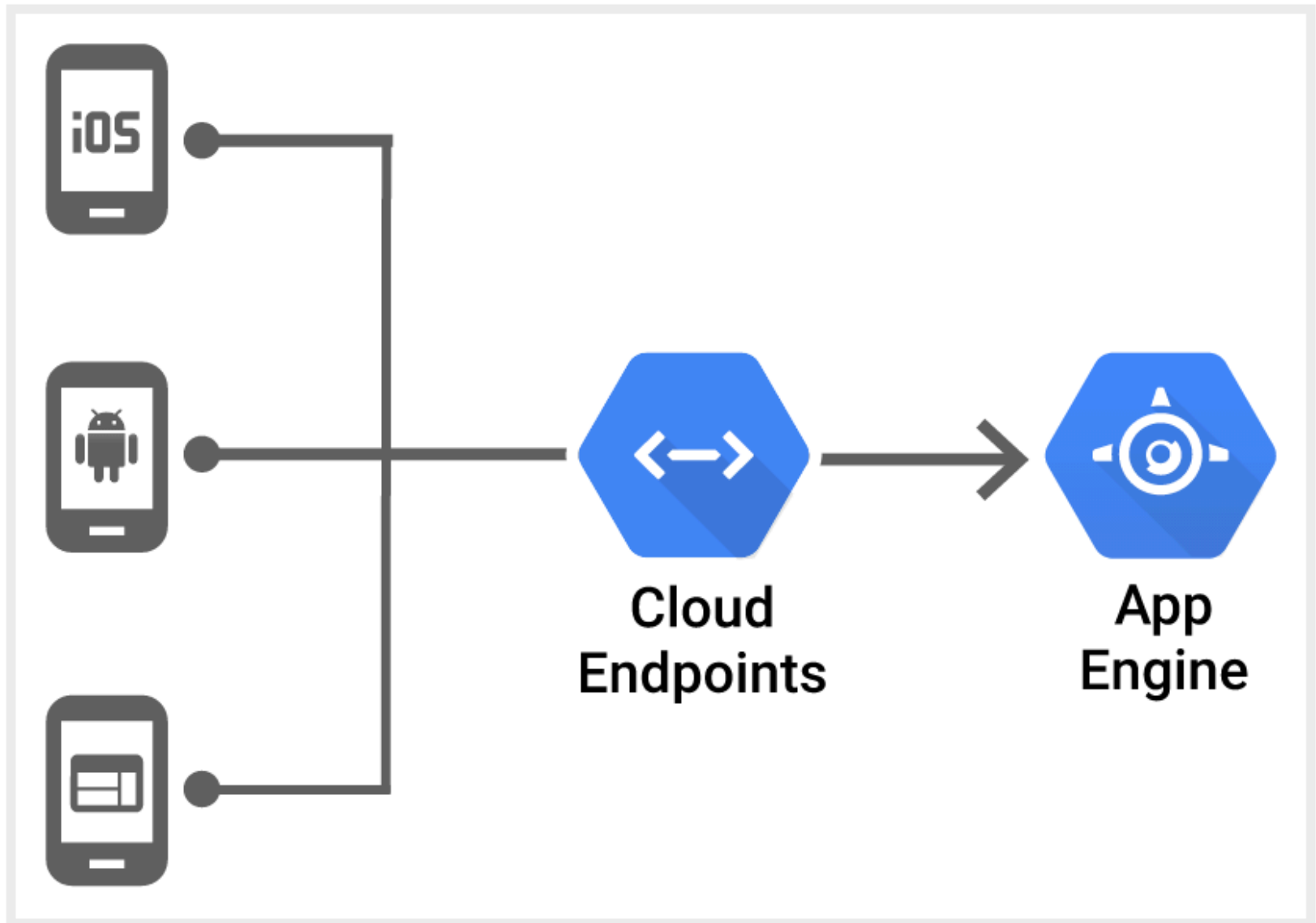
Google App Engine Platform as a Service (PaaS)



Google Compute Engine Infrastructure as a Service (IaaS)

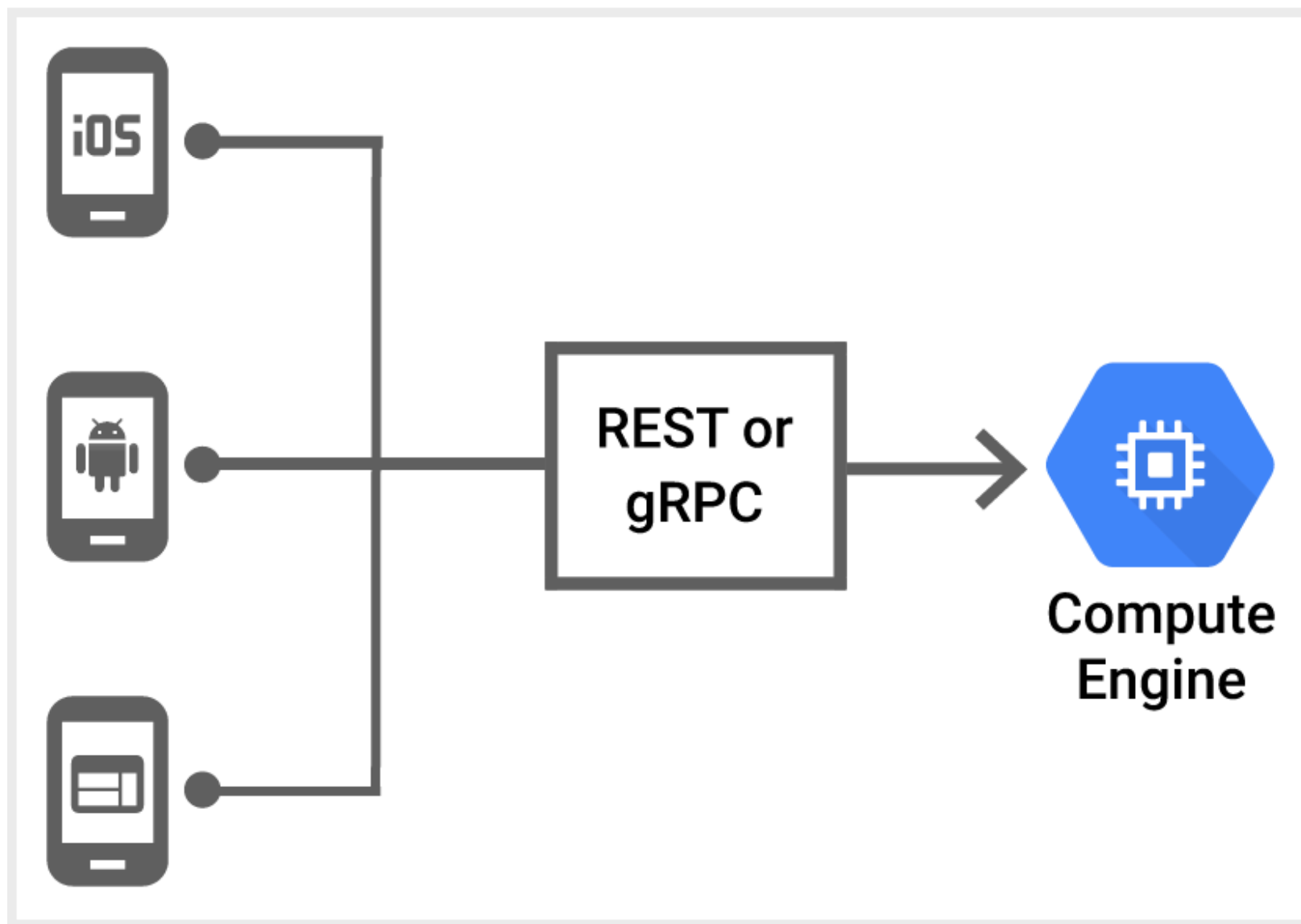


Mobile App Backend Services





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 homepage in a web browser. The browser's address bar displays the URL <https://cloud.google.com/appengine/>. The page header features the Google Cloud Platform logo, navigation links for 'Go to my console' and 'Sign out', a search bar, and buttons for 'Free Trial' and 'Contact Sales'. The main content area has a background image of server racks and includes the App Engine logo, a descriptive paragraph about PaaS, and a 'Start your free trial' button. A secondary navigation bar at the bottom of the main section lists '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

Cloud Datastore - NoSQL

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

Google Cloud Platform

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

[Why Google](#) [Products](#) [Solutions](#) [Pricing](#) [Customers](#) [Documentation](#) [Support](#) [Partners](#) [Free Trial](#) [Contact Sales](#)


Cloud Datastore

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.


[Start your free trial](#)

[Features](#) [Pricing Calculator](#) [Pricing](#) [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

The screenshot shows the Google Cloud Endpoints website. At the top, there's a navigation bar with the Google Cloud Platform logo, a search bar, and links like 'Go to my console | Sign out'. Below this is a secondary navigation bar with links for 'Why Google', 'Products', 'Solutions', 'Pricing', 'Customers', 'Documentation', 'Support', and 'Partners'. A 'Free Trial' button and a 'Contact Sales' button are also present. The main hero section features the Cloud Endpoints logo and a description: 'Create RESTful services and make them accessible to iOS, Android and Javascript clients. Automatically generate client libraries to make wiring up the frontend easy. Built-in features include denial-of-service protection, OAuth 2.0 support and client key management.' A 'Start your free trial' button is prominently displayed. Below the hero section, there's a row of links: 'Features', 'Pricing Calculator', 'Pricing', and 'Documentation'.

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



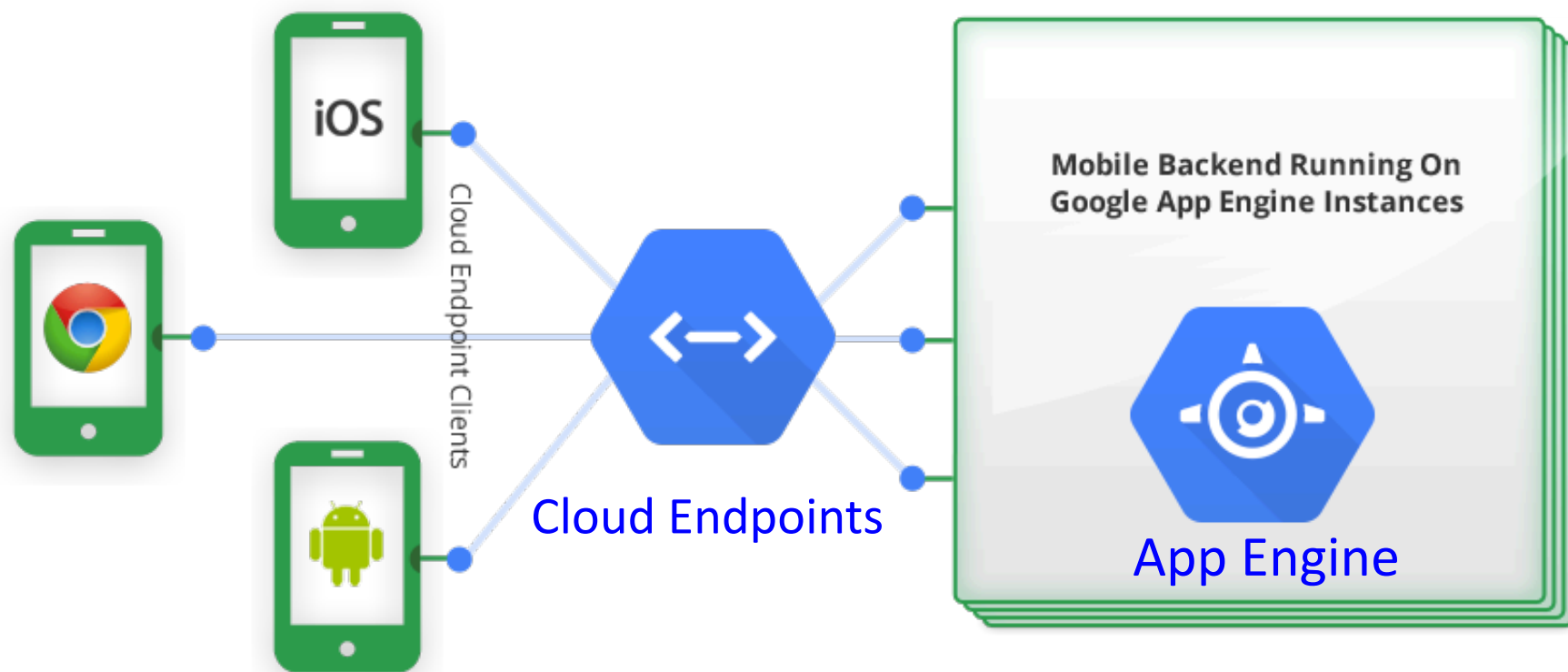
Low maintenance client-server

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/>



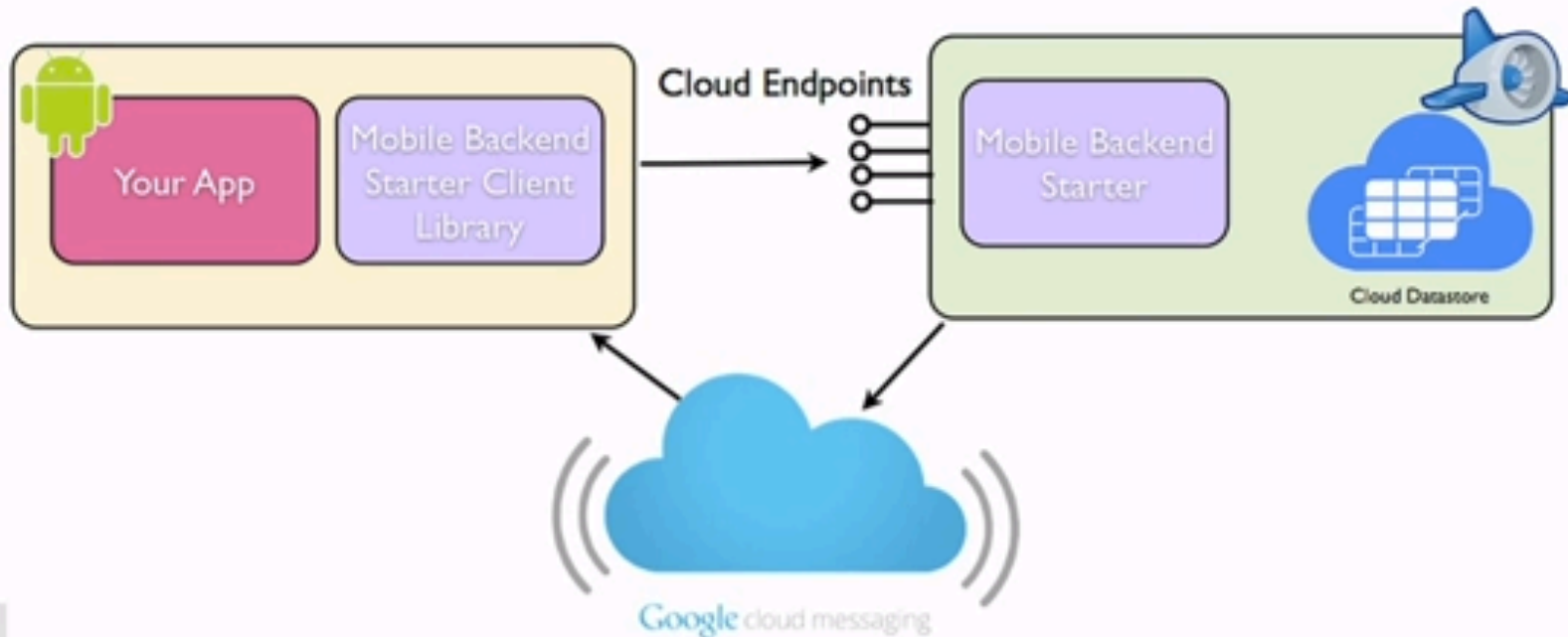
Mobile Apps Backend on Google App Engine



Google Cloud Endpoints Architecture

Mobile App, Google App Engine, Cloud Datastore

Mobile Backend Starter





Try Google Cloud Platform for free

Google Developers Console

https://console.developers.google.com/freetrial?hl=en&_ga=1.105945092.66560398.1402104546



Google Developers Console

+Min-Yuh



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

Taiwan

Account type

☒ Business

☐ Individual

Name and address ?

Business name

Name

Street address

士林區



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

Google Developers Console

士林區

台北市

Postal code

Primary contact

Name

Phone number

imyday@gmail.com

What you pay with

Credit or debit card

Card number

MM / YY

CVC

Cardholder name

☒ 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](#).

Accept and start free trial

Clear

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



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/>

Google Cloud Platform - Docs

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](#)

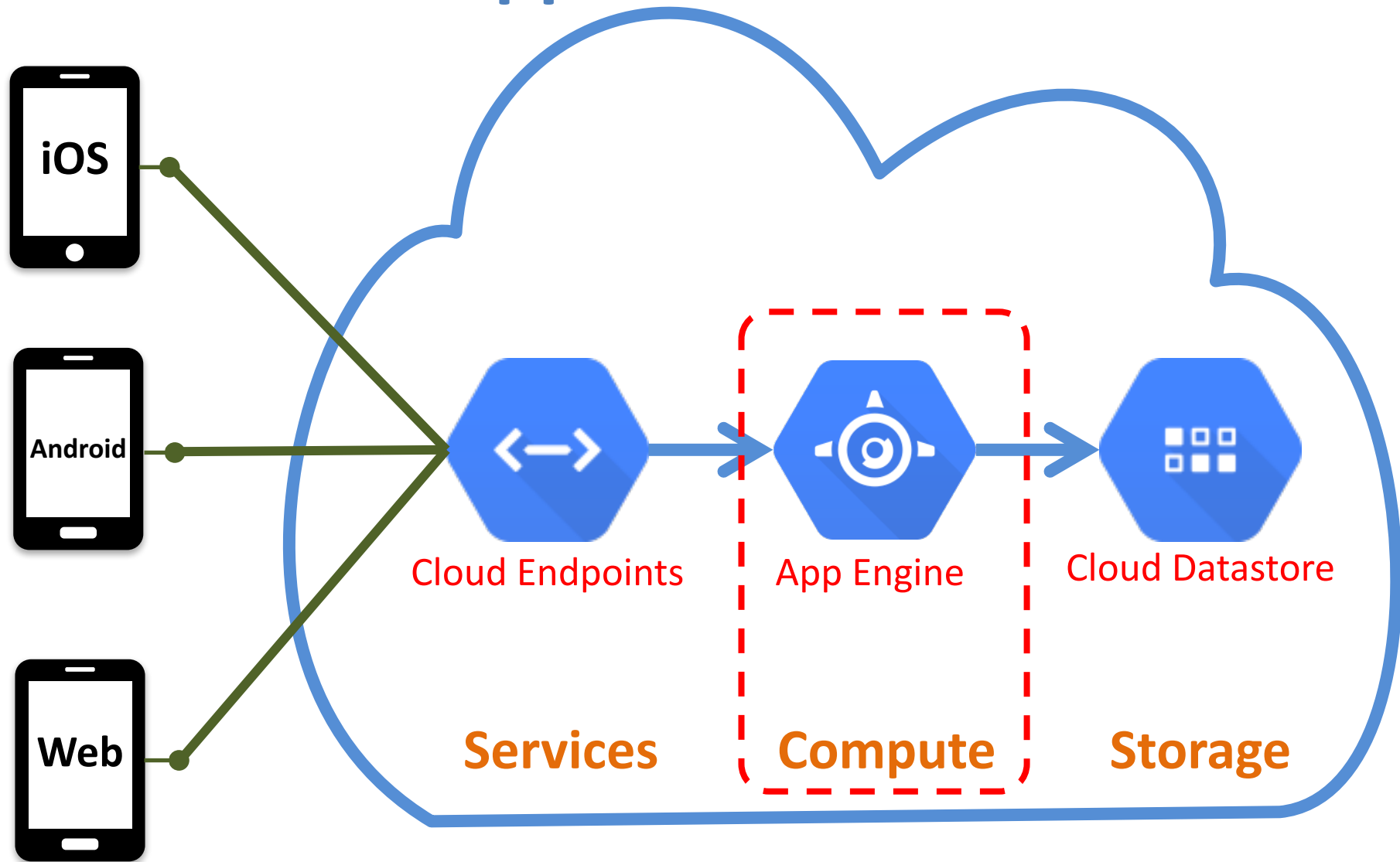
 **Cloud Datastore**

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

Google Developers Console

← → ↺ https://console.developers.google.com/start/appengine?_ga=1.26824890.66560398.1402104546 ☆ ☰

Google Developers Console

[Sign up for a free trial.](#)
[Go to my console](#)

+Min-Yuh

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.

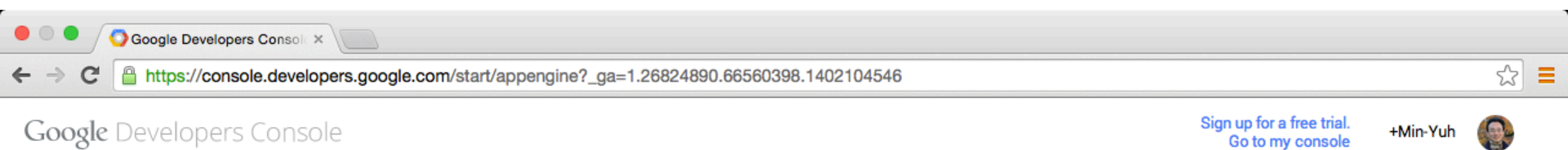
2 SELECT YOUR LANGUAGE

Python

Java



Google App Engine



2 SELECT YOUR LANGUAGE



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

Google Developers Console

Sign up for a free trial.
Go to my console

+Min-Yuh

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

pom.xml

src/main/java/myapp/DemoServlet.java

src/main/webapp/WEB-INF/web.xml

src/main/webapp/WEB-INF/classes

src/main/webapp/index.html

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

▶

Hello, World

60



Google App Engine

HELLO WORLD - JAVA

build.xml

pom.xml

src/main/java/myapp/D

src/main/webapp/WEB-INF

src/main/webapp/WEB-INF

src/main/webapp/WEB-INF

src/main/webapp/index.t

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

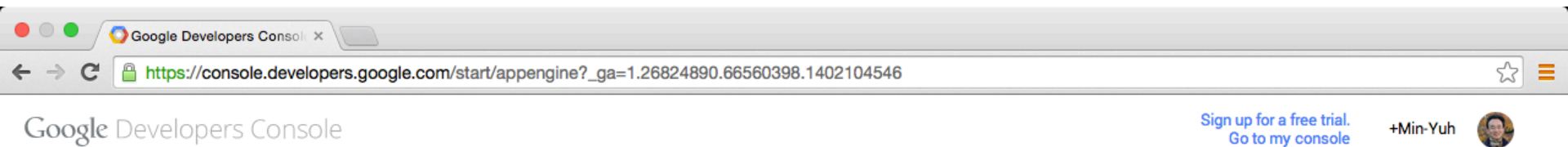


Hello, World

Download this code



Google App Engine



4 INSTALL GOOGLE CLOUD SDK

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

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

Google Developers Console

Sign up for a free trial.
Go to my console

+Min-Yuh

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

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.

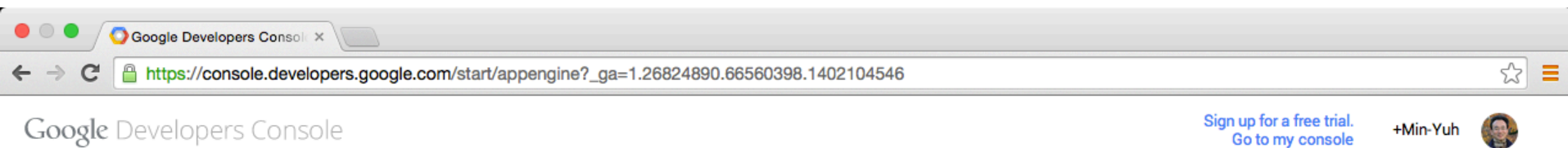
4. Visit the locally running app in your browser: <http://localhost:8080>

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.



Google App Engine

Google Developers Console

https://console.developers.google.com/start/appengine?_ga=1.26824890.66560398.1402104546

Google Developers Console

Sign up for a free trial.
Go to my console

+Min-Yuh

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

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)

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

Source: <https://cloud.google.com/appengine/docs/python/gettingstartedpython27/introduction>



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

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

[View on GitHub](#)

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.

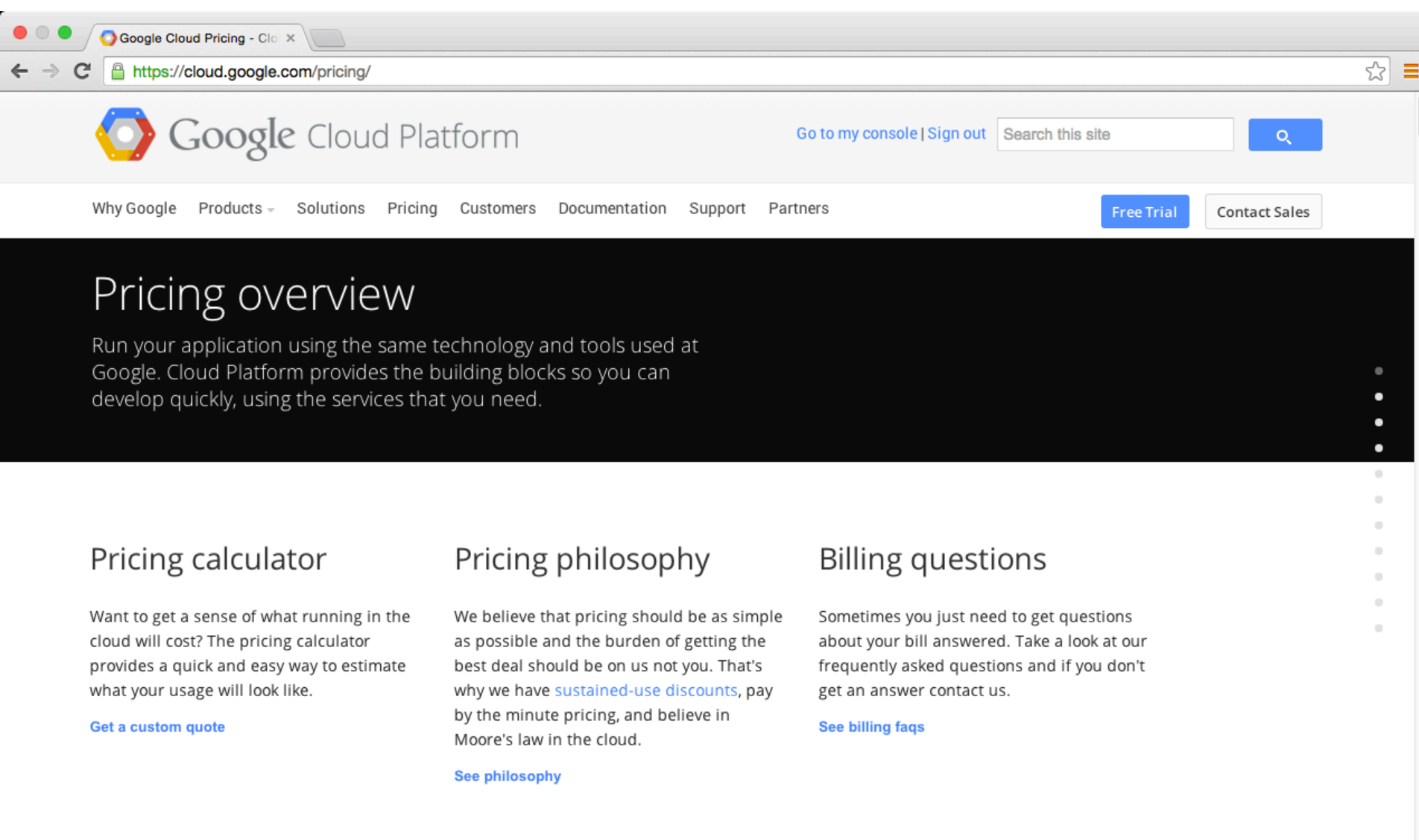
3. Note the application ID (project ID) you created above.
4. Upload your finished application to Google App Engine by invoking the following command. This opens a browser window for you to sign in using your Google account. You'll be providing the project ID as the argument for `-A`.

```
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



The screenshot shows a web browser window with the Google Cloud Platform Pricing page. The browser's address bar shows the URL <https://cloud.google.com/pricing/>. The page header includes the Google Cloud Platform logo, navigation links for 'Go to my console' and 'Sign out', a search bar, and a navigation menu with links for 'Why Google', 'Products', 'Solutions', 'Pricing', 'Customers', 'Documentation', 'Support', and 'Partners'. There are also buttons for 'Free Trial' and 'Contact Sales'. The main content area has a dark background with the title 'Pricing overview' and a paragraph: '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.' Below this, there are three columns: 'Pricing calculator' with a paragraph about estimating costs and a link 'Get a custom quote'; 'Pricing philosophy' with a paragraph about simple pricing and a link 'See philosophy'; and 'Billing questions' with a paragraph about getting answers and a link 'See billing FAQs'. A vertical list of dots on the right side of the page indicates the current position within the document.

Google Cloud Pricing - Clo x

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

Google Cloud Platform

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

[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/>

Google Cloud Platform - Docs

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

☆

Google Cloud Platform

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

Search this site

[Free Trial](#) [Contact Sales](#)

Products > Documentation

Documentation 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

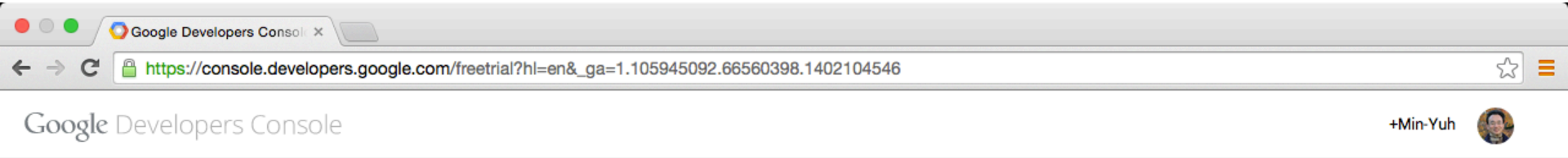
If you're ready to get started, try these solutions for getting software up and running on our platform!

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

Taiwan

Account type

☒ Business

☐ Individual

Name and address ?

Business name

Name

Street address

士林區



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

Google Developers Console

士林區

台北市

Postal code

Primary contact

Name

Phone number

imyday@gmail.com

What you pay with

Credit or debit card

Card number

VISA

AMEX

JCB

MM / YY

CVC

?

Cardholder name

☒ 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](#).

Accept and start free trial

Clear

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

+Min-Yuh



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

Google Developers Console

← → ↺ https://console.developers.google.com/start/appengine?_ga=1.26824890.66560398.1402104546 ☆ ☰

Google Developers Console

[Sign up for a free trial.](#)
[Go to my console](#)

+Min-Yuh

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.

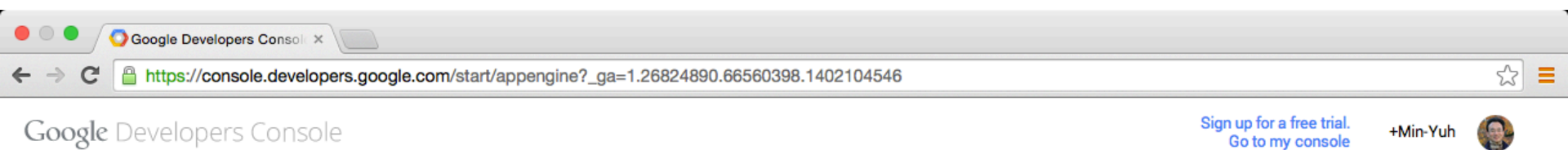
2 SELECT YOUR LANGUAGE

Python

Java



Google App Engine



2 SELECT YOUR LANGUAGE



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

Google Developers Console

Sign up for a free trial.
Go to my console

+Min-Yuh

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

pom.xml

src/main/java/myapp/DemoServlet.java

src/main/webapp/WEB-INF/web.xml

src/main/webapp/WEB-INF/classes

src/main/webapp/index.html

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

▶

Hello, World

85



Google App Engine

HELLO WORLD - JAVA

build.xml
pom.xml
[src/main/java/myapp/D](#)
src/main/webapp/WEB-INF
src/main/webapp/WEB-INF
src/main/webapp/WEB-INF
src/main/webapp/index.t

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

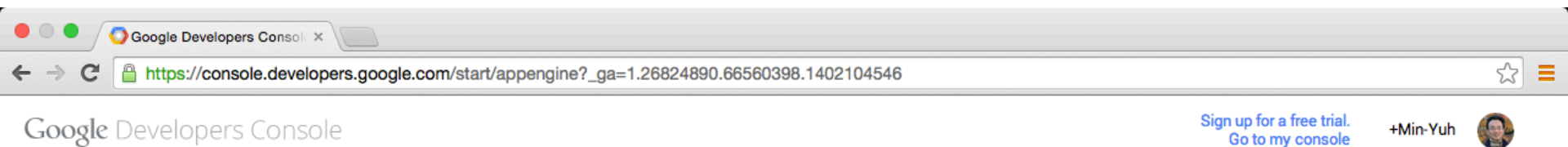


Hello, World

[Download this code](#)



Google App Engine



4 INSTALL GOOGLE CLOUD SDK

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

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
```

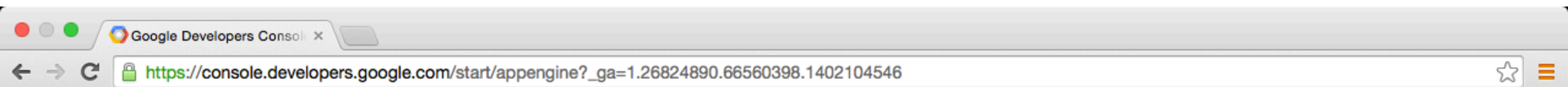
4. Install the App Engine package for Java using this command:

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

5 RUN YOUR APP LOCALLY



Google App Engine



Google Developers Console

Sign up for a free trial.
Go to my console

+Min-Yuh



5 RUN YOUR APP LOCALLY

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

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.

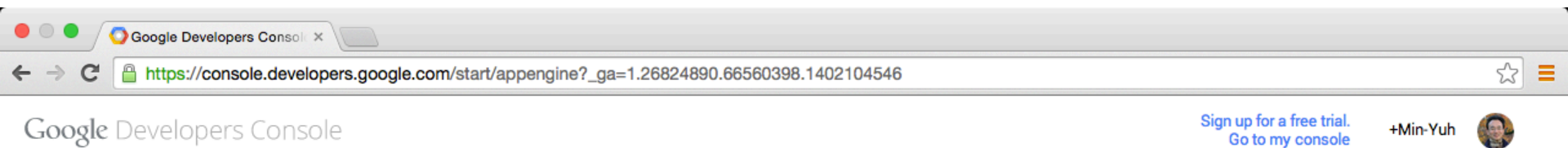
4. Visit the locally running app in your browser: <http://localhost:8080>

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.



Google App Engine

Google Developers Console

https://console.developers.google.com/start/appengine?_ga=1.26824890.66560398.1402104546

Google Developers Console

[Sign up for a free trial.](#)
[Go to my console](#)

+Min-Yuh

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

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)

Create Project: HelloWorldGoogleAppEngine

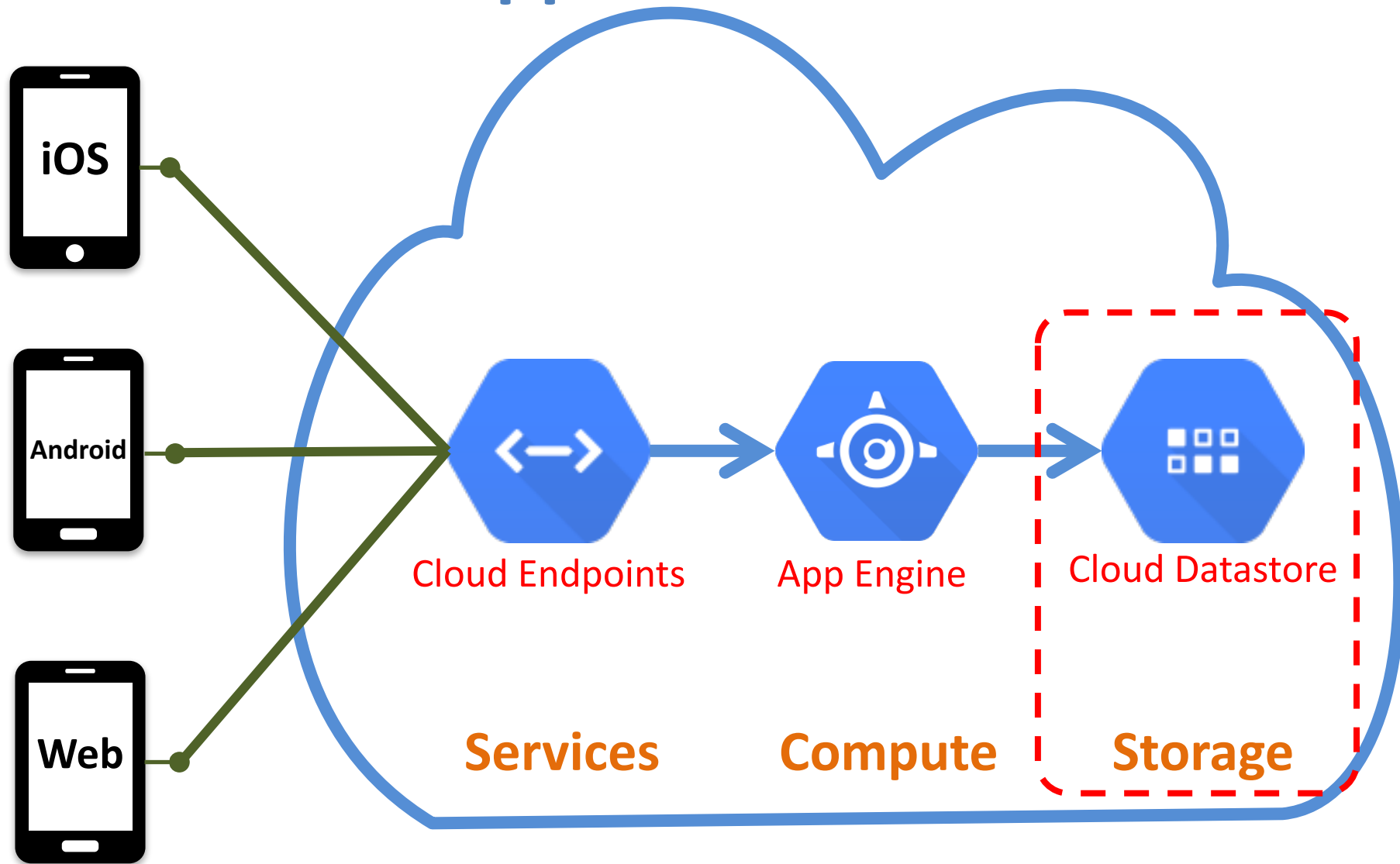
See all activity



Google Cloud Datastore

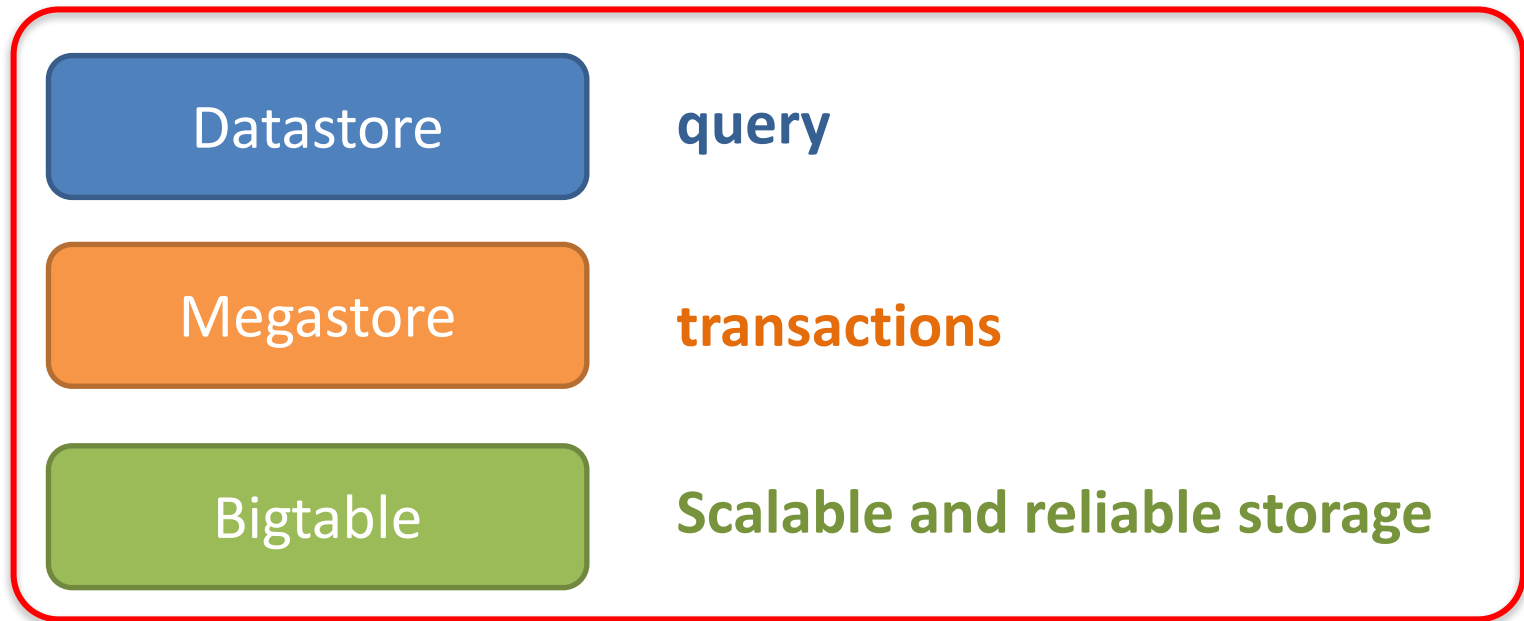


Mobile App Backend Services



Datastore Internals

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



What is Bigtable?

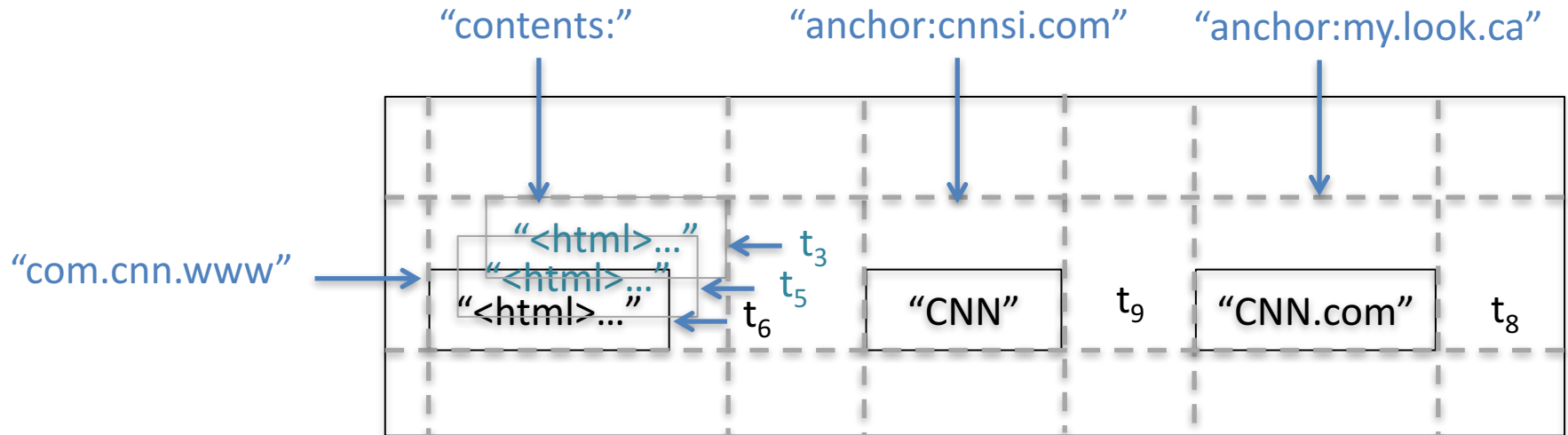
- Scalable, distributed, highly-available and structured storage
 - Bigtable is not database by itself (it doesn't support query)
- Google usage
 - In production since April 2005
 - Web Search, YouTube, Earth, Analytics

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

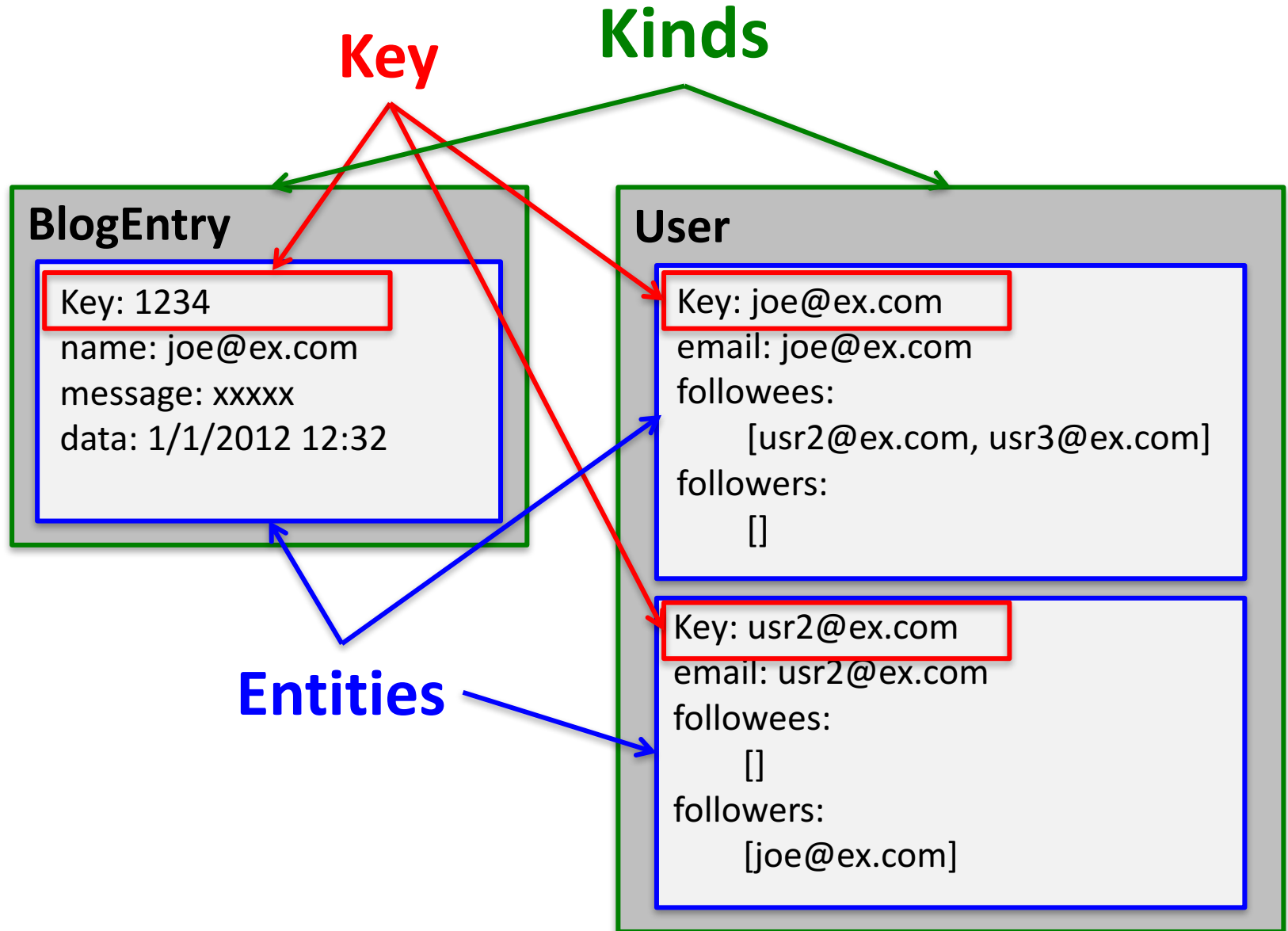


Google Datastore Basic Operation

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

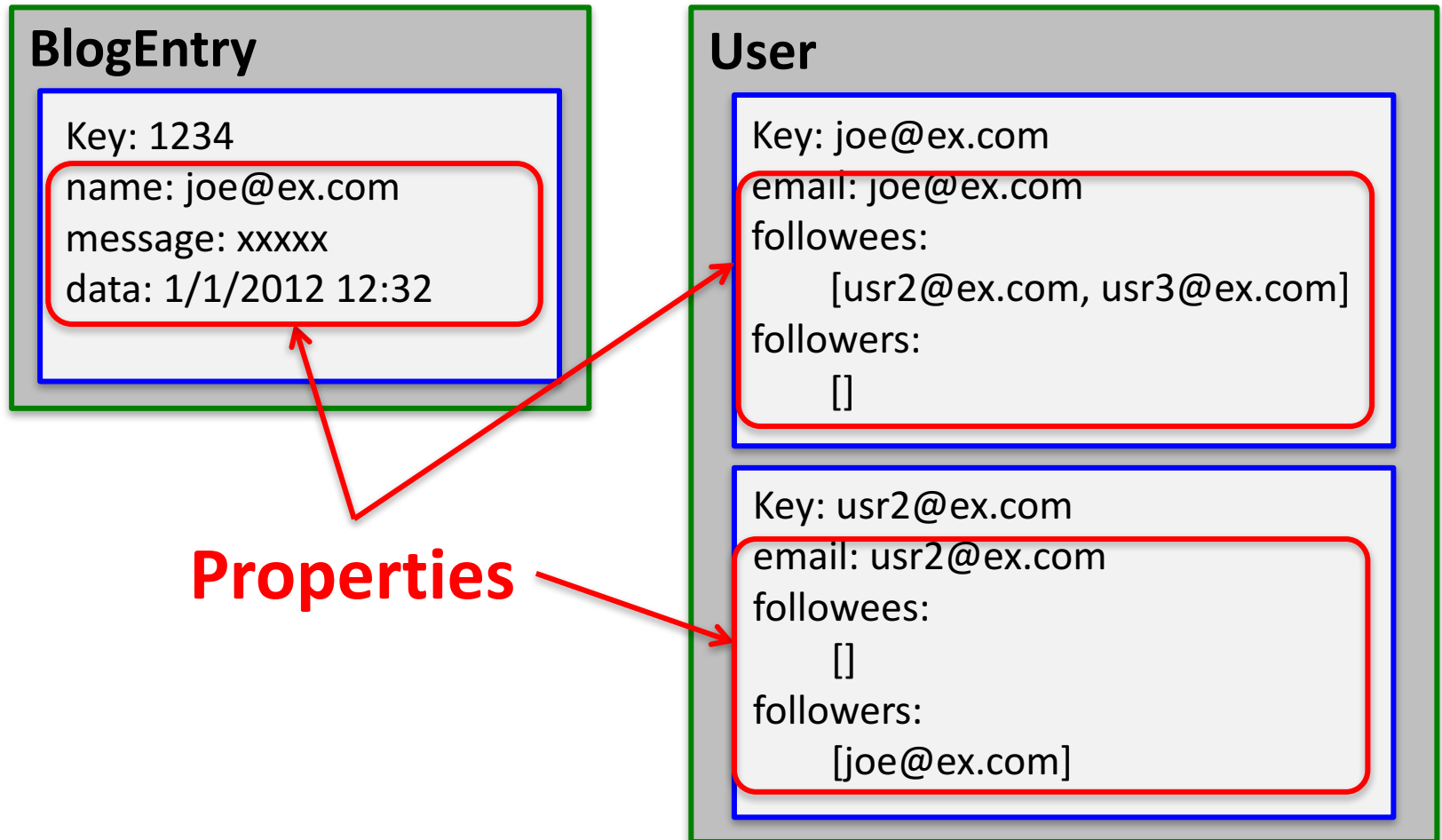
Kind, Entity and Key



Properties and Data Types

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

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

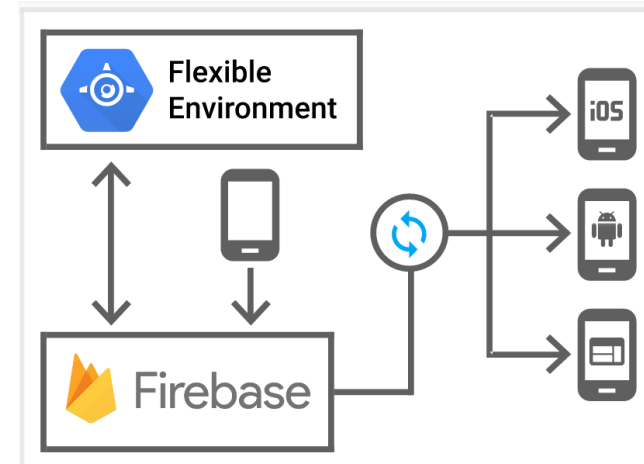
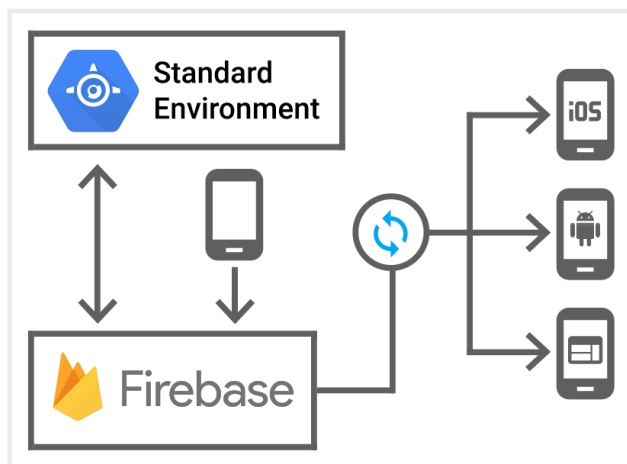
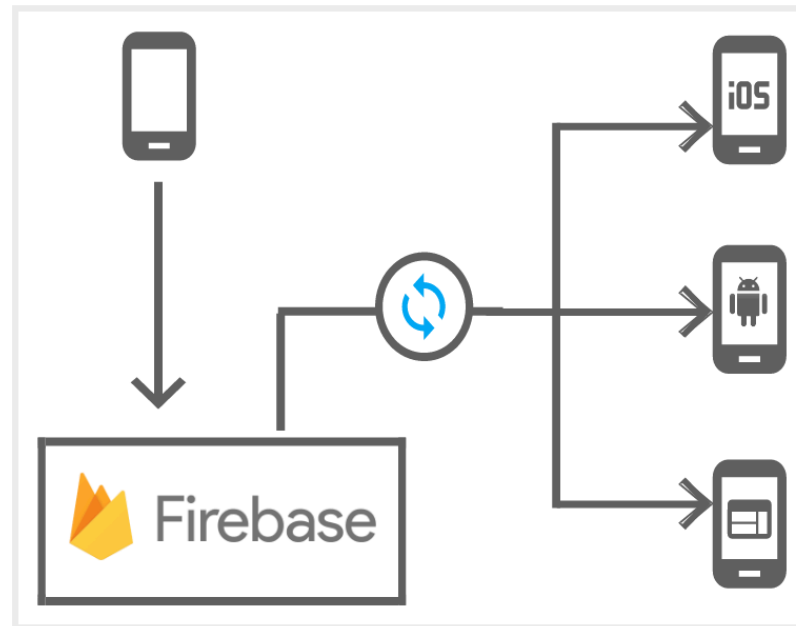


Creating an Entity with Java Low-level API

```
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

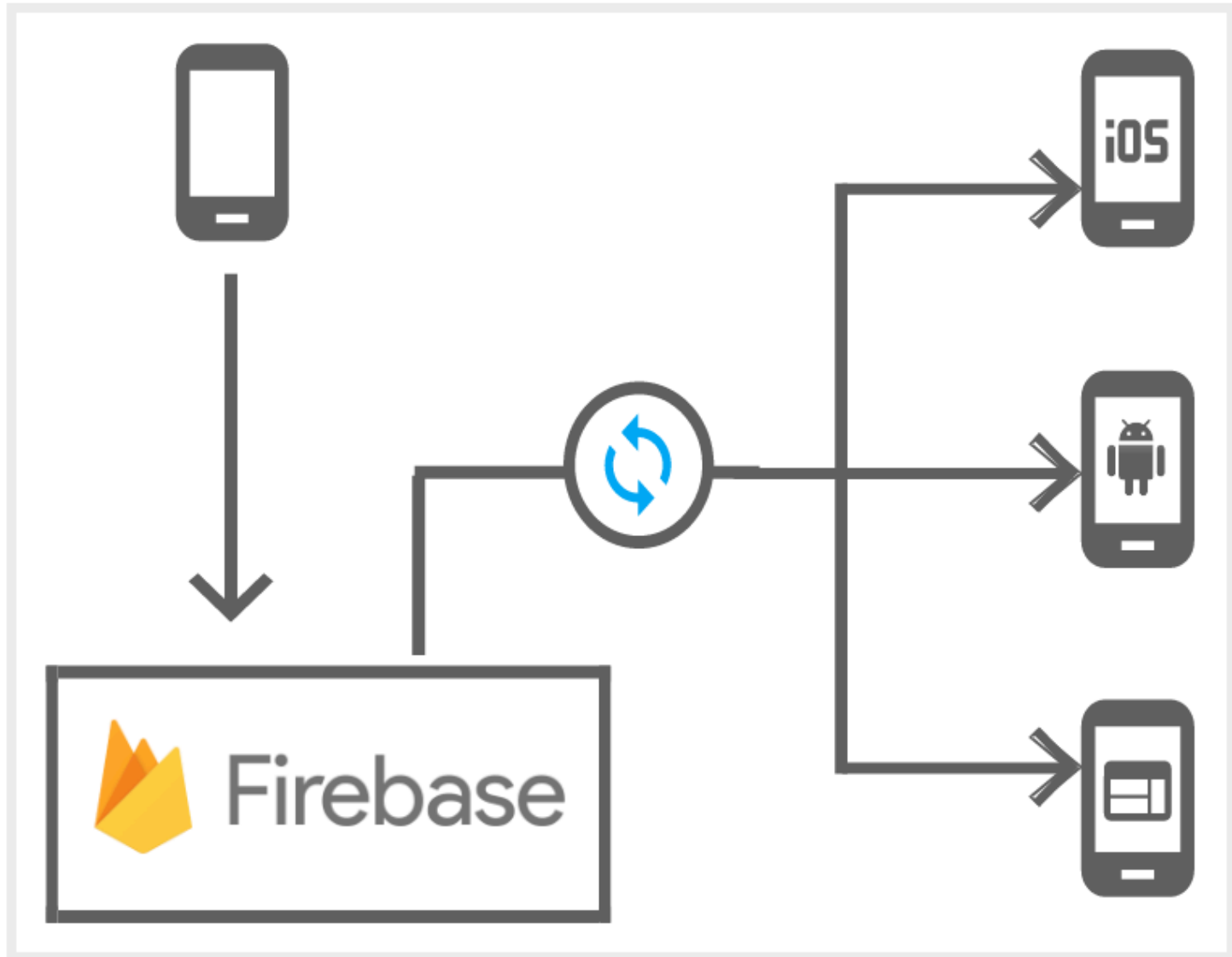


Firebase

Firebase: Backend as a Service (BaaS)

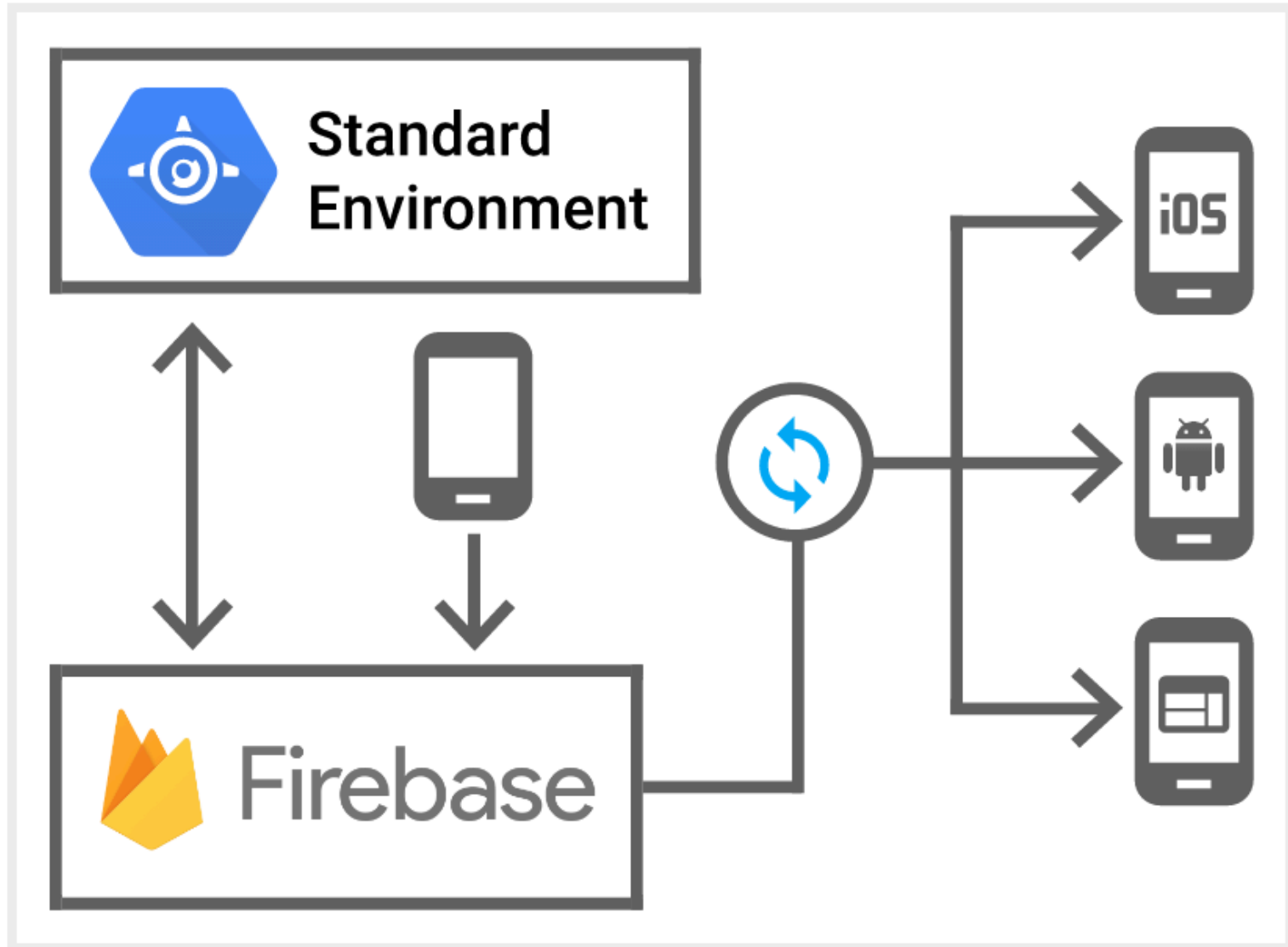


Mobile App Backend Services





Mobile App Backend Services





Firebase

Firebase

← → ↻ <https://firebase.google.com> ☆

Firebase Home Features Pricing Docs Customers Support 🔍 Search [Go to console](#)

App success made simple

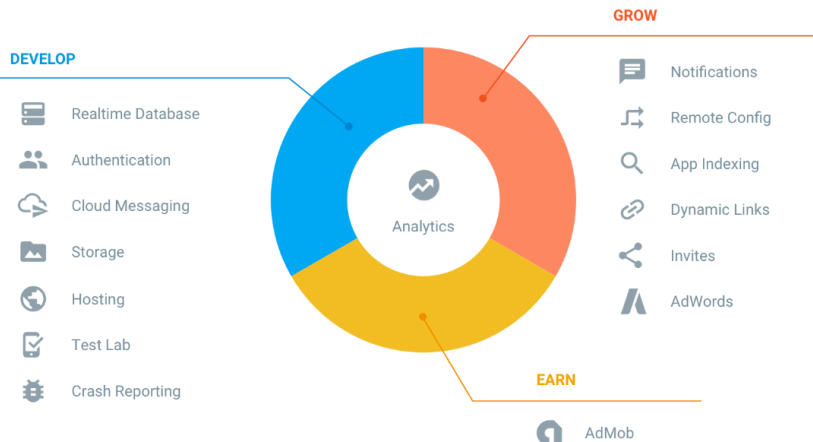
The tools and infrastructure you need to build better apps and grow successful businesses

[GET STARTED FOR FREE](#)

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.

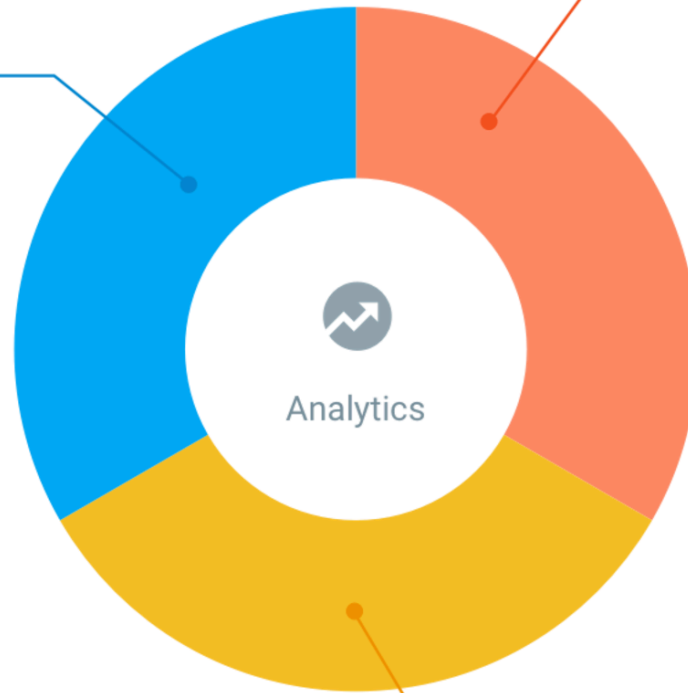
[ALL FEATURES](#)









Firebase

DEVELOP

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



GROW

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

EARN





Firebase

Firebase

← → ↻ <https://firebase.google.com> ☆



Home

Features

Pricing

Docs

Customers

Support



Search

[Go to console](#)



SWIFT

OBJECTIVE-C

JAVA

JAVASCRIPT

MORE ▾

```
FIRAuth.auth()?.addAuthStateChangeListener({ (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.





Firebase

Firebase

← → ↻ 🔒 <https://firebase.google.com> ☆

Firebase Home Features Pricing Docs Customers Support 🔍 Search [Go to console](#)

App success made simple

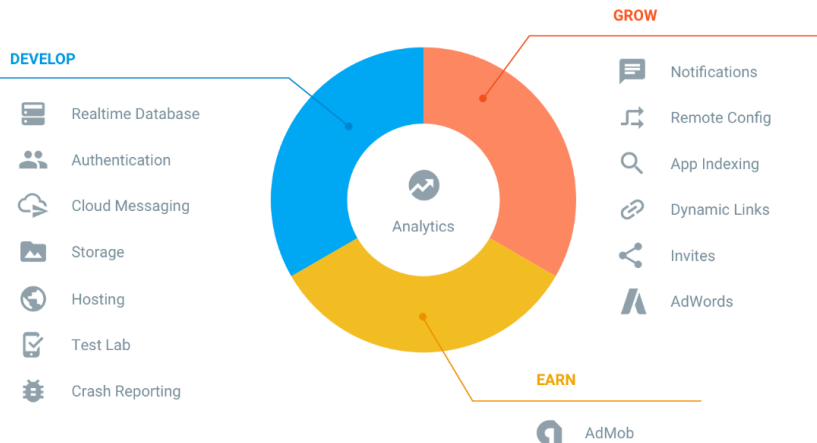
The tools and infrastructure you need to build better apps and grow successful businesses

[GET STARTED FOR FREE](#)

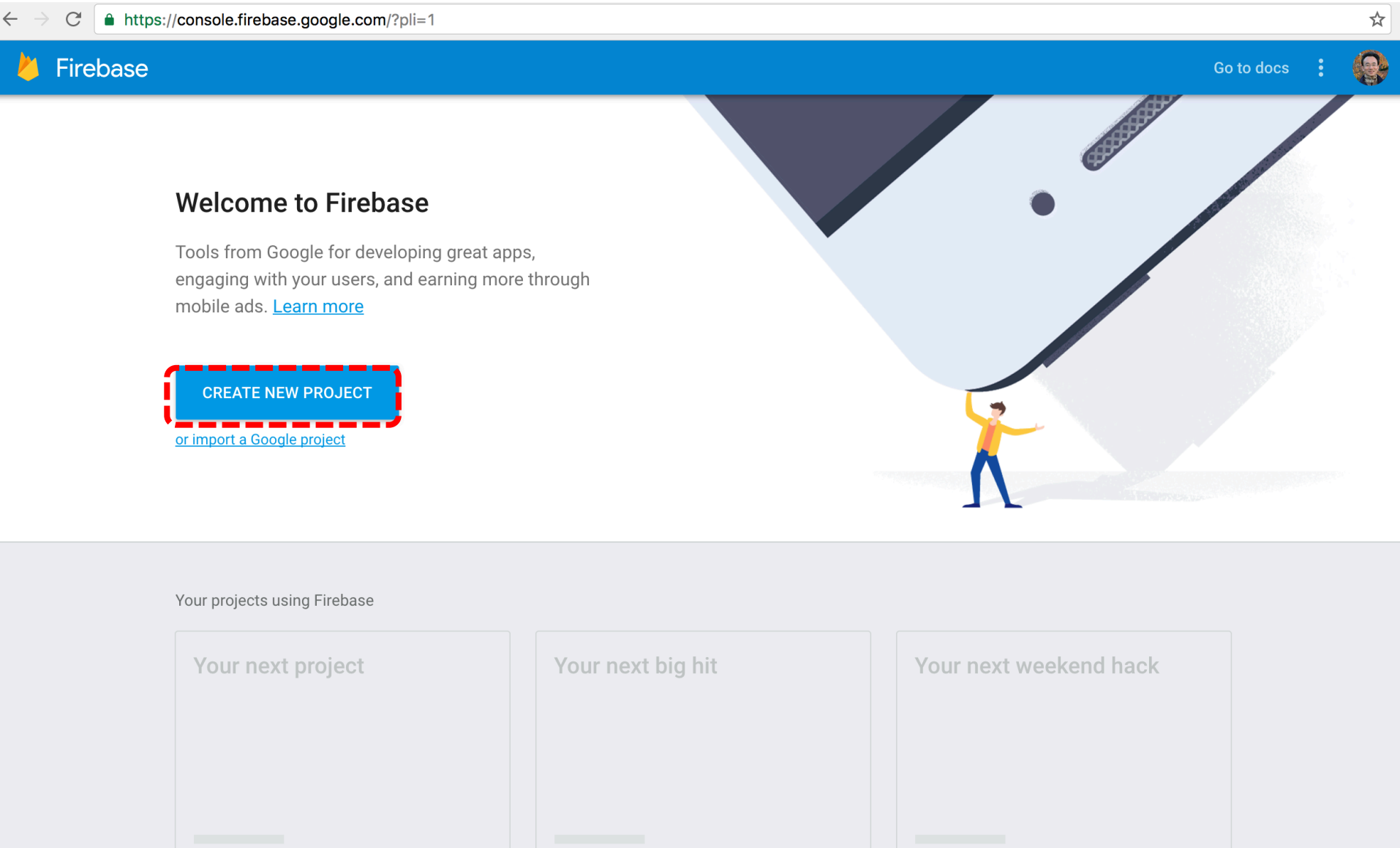
Move fast

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



The screenshot shows the Firebase console interface. At the top is a blue header with the Firebase logo on the left, a 'Go to docs' link, and a user profile icon on the right. Below the header, the main content area features a large illustration of a person in a yellow shirt and blue pants standing next to a giant smartphone. The phone's screen displays the text 'Welcome to Firebase' and 'Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads. [Learn more](#)'. Below this text is a prominent blue button with a red dashed border that says 'CREATE NEW PROJECT', followed by the text 'or import a Google project'. At the bottom of the page, there is a section titled 'Your projects using Firebase' which contains three placeholder boxes labeled 'Your next project', 'Your next big hit', and 'Your next weekend hack'.

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

 **Firebase** Go to docs 

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](#)

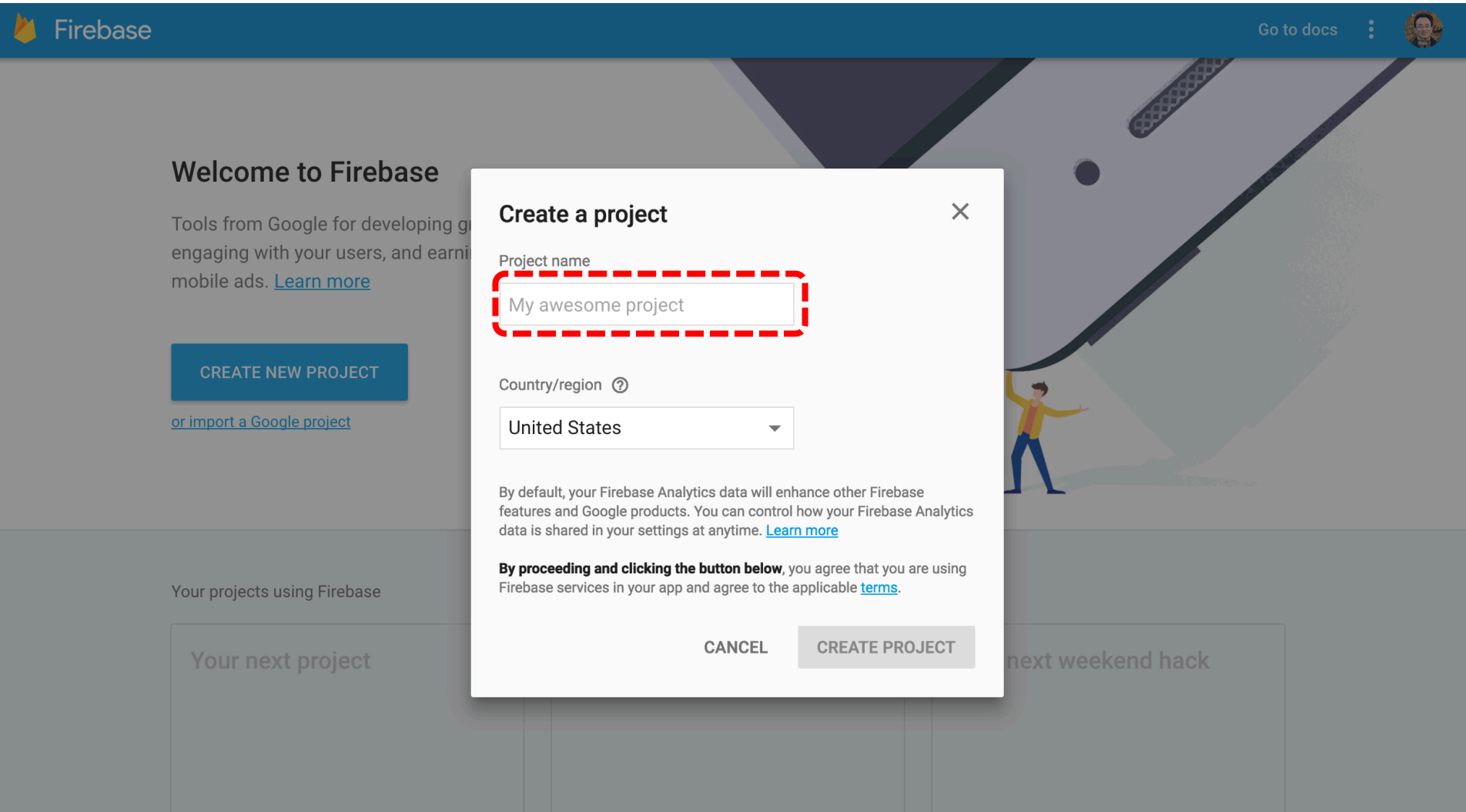
Your projects using Firebase

Your next project

Your next big hit

Your next weekend hack

Firebase Project



The screenshot shows the Firebase console interface. At the top, there's a dark blue header with the Firebase logo on the left and 'Go to docs' with a user profile icon on the right. The main content area has a light gray background with a large illustration of a person climbing a slide. On the left, there's a 'Welcome to Firebase' section with a 'CREATE NEW PROJECT' button and a link to 'Learn more'. Below this, there's a section for 'Your projects using Firebase' and 'Your next project'. A modal dialog box titled 'Create a project' is centered on the screen. It has a close button (X) in the top right corner. The dialog contains a 'Project name' field with the text 'My awesome project', which is highlighted by a red dashed border. Below this is a 'Country/region' dropdown menu with a question mark icon, currently set to 'United States'. At the bottom of the dialog, there's a paragraph of text about Firebase Analytics data sharing, followed by a bold statement: 'By proceeding and clicking the button below, you agree that you are using Firebase services in your app and agree to the applicable terms.' At the very bottom of the dialog are two buttons: 'CANCEL' and 'CREATE PROJECT'.

Firebase

Go to docs

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](#)

Create a project

Project name

My awesome project

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

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](#)

Your projects using Firestore

Your next 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](#)

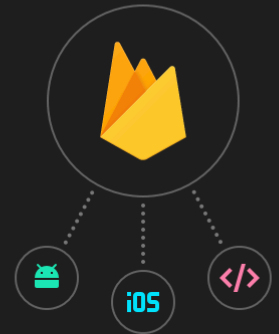
By proceeding and clicking the button below, you agree that you are using Firestore services in your app and agree to the applicable [terms](#).

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.



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

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

← → ↻ <https://console.firebase.google.com/project/helloworldfirebase-d6ef2/overview> ☆ ⋮


Firebase HelloWorldFirebase ▾ Go to docs ⋮ 

Overview Overview ?


Welcome to Firebase! Get started here.



Add Firebase to your iOS app




Add Firebase to your Android app




Add Firebase to your web app


Discover Firebase



Analytics
Get detailed analytics to measure and



Authentication
Authenticate and manage users from a



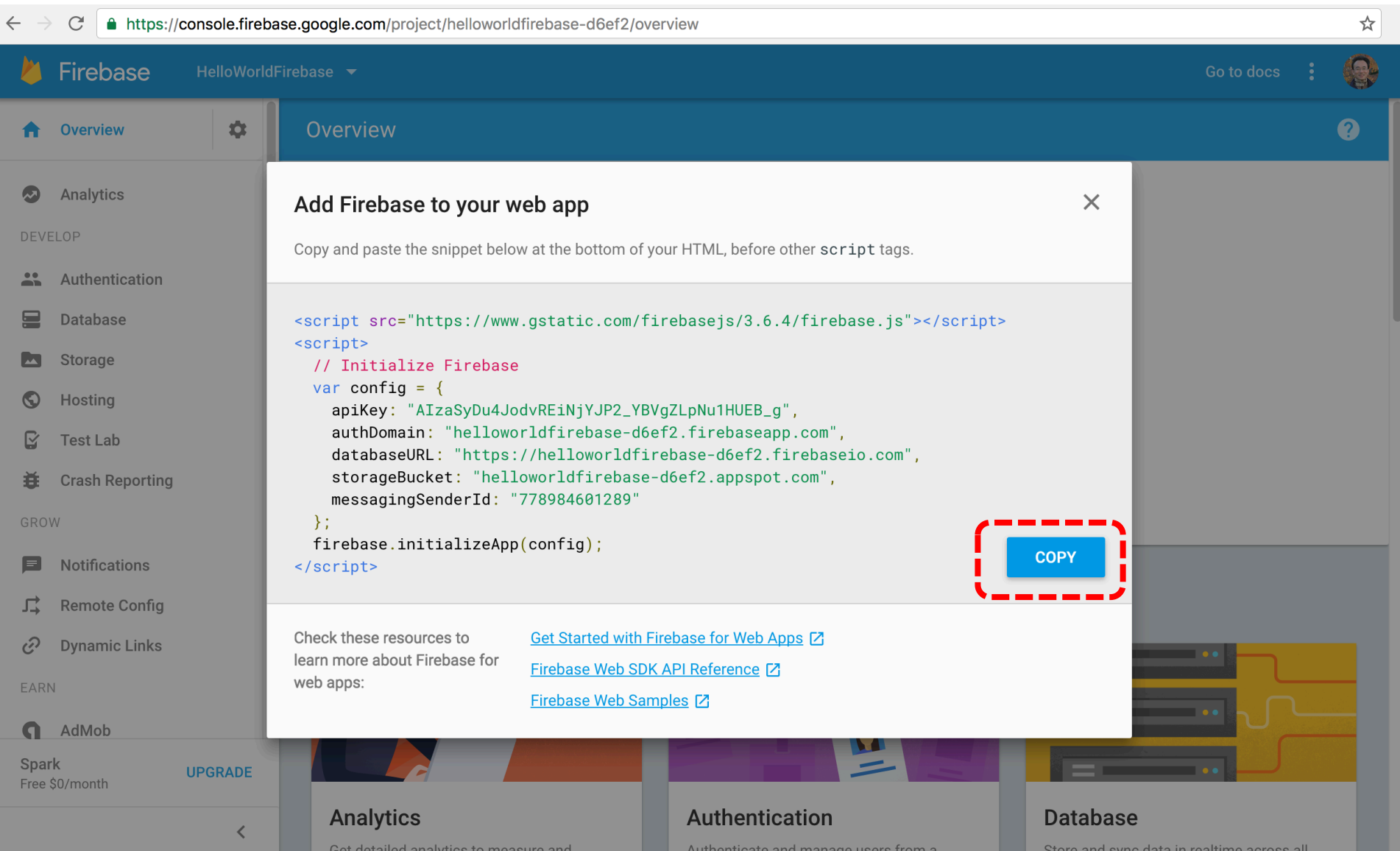
Database
Store and sync data in realtime across all

Analytics
Get detailed analytics to measure and

Authentication
Authenticate and manage users from a

Database
Store and sync data in realtime across all

Add Firebase to Your Web App



← → ↻ <https://console.firebase.google.com/project/helloworldfirebase-d6ef2/overview> ☆

Firebase HelloWorldFirebase ▾ Go to docs

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 Get detailed analytics to measure and

Authentication Authenticate and manage users from a

Database Store and sync data in realtime across all

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

Firebase 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();
```

Firebase 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": { ... }
}
```

Firebase 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  
  });  
}
```



Firebase 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;  
    // ...  
});
```


Firebase Database

← → ↻ 🔒 <https://console.firebase.google.com/project/helloworldfirebase-d6ef2/database/data> ☆ ⋮

 **Firebase** HelloWorldFirebase ▾ Go to docs ⋮ 

🏠 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/](https://helloworldfirebase-d6ef2.firebaseio.com/) + - ⋮

Firebase Database Rules

← → ↻ <https://console.firebase.google.com/project/helloworldfirebase-d6ef2/database/rules> ☆ ⋮

 **Firebase** HelloWorldFirebase ▾ Go to docs ⋮ 

🏠 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 }
```

Firebase Database

← → ↻ <https://console.firebase.google.com/project/helloworldfirebase-d6ef2/database/data> ☆

 **Firebase** HelloWorldFirebase ▾ Go to docs ⋮ 

🏠 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 + ✕

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

Firebase Database

The screenshot shows the Firebase Realtime Database console interface. The left sidebar contains navigation links for Overview, Analytics, and various development tools. The main content area is titled 'Realtime Database' and has tabs for DATA, RULES, USAGE, and BACKUPS. The 'DATA' tab is active, displaying a URL bar with the database's endpoint. A notification banner states that default security rules require authentication. Below this, a data entry form is shown for a node named 'helloworldfirebase-d6ef2'. The form includes a table with columns 'Name' and 'Value', where 'score' is entered as the name and '100' as the value. A red dashed box highlights the 'ADD' button. Below the table, there are fields for 'email' and 'username'.

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

Firebase HelloWorldFirebase

Go to docs

Realtime Database

DATA RULES USAGE BACKUPS

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

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

helloworldfirebase-d6ef2

Name	Value
score	100

CANCEL ADD

email: "imyday@gmail.com"

username: "imyday"

Spark Free \$0/month [UPGRADE](#)

Firebase Database

The screenshot shows the Firebase Realtime Database console 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), EARN (AdMob), and Spark (Free \$0/month) with an 'UPGRADE' button. The main content area is titled 'Realtime Database' and has tabs for DATA, RULES, USAGE, and BACKUPS. The 'DATA' tab is active, showing a URL 'https://helloworldfirebase-d6ef2.firebaseio.com/'. Below the URL is a notification: 'Default security rules require users to be authenticated' with 'LEARN MORE' and 'DISMISS' links. A JSON object is displayed, highlighted with a red dashed border:

```
helloworldfirebase-d6ef2
{
  email: "imyday@gmail.com"
  score: 100
  username: "imyday"
}
```

Summary

- **Google App Engine**

- **Google Cloud Platform**



- **Google Cloud Datastore**



- **Google Firebase**



Firebase



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/>