



MPCS 51033 • SPRING 2017 • SESSION 1

BACKENDS FOR MOBILE APPLICATIONS

WELCOME TO BACKENDS
FOR MOBILE APPLICATIONS

COURSE LOGISTICS

COURSE LOGISTICS

- Lectures
 - Wednesday 5:30-8:30pm
 - Young, 302
- Attendance is highly recommended



COURSE LOGISTICS

- Office Hours
 - TBD
- TA Office Hours
 - Susan



COURSE LOGISTICS

- Prerequisites
 - iOS
 - Android



COURSE LOGISTICS

- Instructors
 - Andrew Binkowski
 - Susan Stevens



COURSE RESOURCES

COURSE RESOURCES

- No required text books
- Documentation from services
- Blogs
- Online tutorials

COURSE RESOURCES

- Additional Resources:

- Apple Dev Forums
- Stack Overflow
- Google Groups (iPhone SDK, etc.)
- Blogs
- You Tube
- ...

WARNING: CHECK THE DATES

Welcome to Q&A for professional and enthusiast programmers — check out the [FAQ!](#)

log in | careers | chat | meta | about | faq | search

StackExchange v

stackoverflow Questions Tags Users Badges Unanswered Ask Question

Top Questions interesting 249 featured hot week month

Open Source Equivalent of TFS? 0 votes 0 answers 1 view .net version-control tfs continuous-integration work-item-tracking 50s ago Lawrence Wagerfield 333

android spinner and button problem 0 votes 0 answers 1 view android 57s ago hardy.exe 15

R combining split and cumsum 0 votes 0 answers 2 views r sports 1m ago pssguy 169

How to create resources, when using capybara+selenium? 0 votes 0 answers 2 views ruby-on-rails-3 integration-testing 1m ago Uilitiy 84

English equivalents to Vim's shortcuts 0 votes 0 answers 5 views vim 1m ago Rook 1,458

Extracting specific data from a string with regex and Powershell 0 votes 0 answers 3 views regex powershell 1m ago user353401 3

Adding a target to an Xcode project 0 votes 0 answers 2 views iphone objective-c ios xcode iphone-sdk-4.0 2m ago imperiousdev 101

Prevent WCF *.datasource files from being checked-in 0 votes 0 answers 2 views iffs2010 2m ago kroonwijk 134

jQuery getJSON data order varies across browsers 1 vote 0 answers 15 views jquery getjson 3m ago Joseph 6,957

How to create or write STDF files? 0 votes 0 answers 5 views c# c++ c 3m ago rnbadawi23 51

Why does a thread, on ubuntu 2.6.38-generic or 3.0.0-lowlatency kernel, block another thread from taking mmap_sem? 0 votes 0 answers 7 views c linux ubuntu kernel real-time 3m ago Dan Pisarski 1

Should reversing an NSMutableArray be avoided when possible? 0 votes 0 answers 4 views objective-c cocoa-touch nsmutablearray enumeration 3m ago maxedison 1,052

Javamelody and Play Framework 0 votes 0 answers 2 views playframework monitoring 3m ago peperg 842

PHP - JS and CSS minifier 1 vote 1 answer 17 views php javascript css minify 3m ago genesis φ 16.9k

Hello World! This is a collaboratively edited question and answer site for professional and enthusiast programmers. It's 100% free, no registration required. [about »](#) [faq »](#)

CAREERS 2.0

- Android Developer GPSshopper Chicago, IL; New York, NY
- Internet Web Developer Monotype Imaging Elk Grove Village, IL
- Front-End Web Developer Straight North Oak Brook, IL
- Senior Software Engineer - Server Products (C++) Trading Technologies Chicago, IL
- Software Engineer (Python) Leapfrog Online Evanston, IL
- Quality Assurance Analyst FactSet Research Systems Chicago, IL; Norwalk, CT

Recent Tags

- c# × 76
- java × 66
- php × 51
- jquery × 47
- javascript × 43
- c++ × 29
- mysql × 29
- android × 28
- sql × 24
- python × 24
- css × 22
- asp.net × 21
- .net × 21
- sql-server × 21
- html × 20
- iphone × 20

COURSE RESOURCES

- Course website
 - <http://uchicago.cloud>
 - Lecture slides
 - Assignments
 - Resources and links
- Class communication
 - Course website for information, articles, homework, etc.



MPCS
51033

backends for mobile
applications

[Home](#)
[About This Course](#)
[Sessions Notes](#)
[Course Information](#)
[Forum](#)

anytime has become th
and development. New
to connect applications
considerations such as
balanced to mee the de

© 2017 T.A. Binkowski · All rights reserved · The

COURSE RESOURCES

- Course forum located on GitHub Issues
 - [uchicago-cloud/2017-Spring-Forum](#)
- Post questions, interesting links, etc.
 - Anyone can post
 - Anyone can answer

ALL CLASS COMMUNICATION WILL BE DONE THROUGH THIS CLASS

The screenshot shows a GitHub repository page for [uchicago-cloud / mpcs51033-2017-spring-forum](#). The 'Issues' tab is selected, showing 6 open issues. A yellow arrow points from the text 'ALL CLASS COMMUNICATION WILL BE DONE THROUGH THIS CLASS' to the first issue listed.

Issue #	Title	Opened By	Created At
#6	Assignment Clarification	tabinks	3 days ago
#5	Submitting to Assignment 1 Repositories	susanstevens	3 days ago
#4	Office Hours	susanstevens	6 days ago
#3	gcloud command not found	tabinks	6 days ago
#2	Assignment 1 Repository	tabinks	6 days ago
#1	Welcome to Backends for Mobile Applications!	tabinks	6 days ago

Filters: is:issue is:open

Author: Author ▾ **Labels:** Labels ▾ **Milestones:** Milestones ▾

ProTip! Updated in the last three days: [updated:>2017-04-02](#).

Inc. Terms Privacy Security Status Help

COURSE DESIGN

COURSE DESIGN

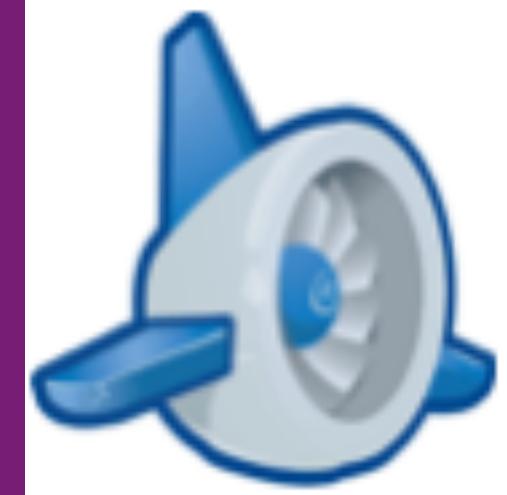
- Today an app is not enough
 - Complexity
 - Interconnected
 - User expectations
 - Insights/analytics
- Provide exposure to technologies to complete mobile applications

COURSE DESIGN



COURSE DESIGN

- Current state of the field
 - Technologies come and go
 - Consolidation/duplication of services
 - DIY mentality
 - Industry proving ground doesn't fit with the narrative



**Google App
Engine**



COURSE DESIGN

- Polled former students and colleagues about the topics that are most beneficial to industry

The screenshot shows the 'Jobs at Apple' website interface. At the top, there are navigation links for Mac, iPad, iPhone, Watch, TV, Music, and Support. Below that, there are links for Corporate, Retail, and Students. The main title is 'Jobs at Apple' and it displays '591 job(s) found'. On the left, there is a 'Filter by' sidebar with options like Keywords (selected), Location, Language Skills, Business Line, Job Function, and My Filter Mixes. Below the filter sidebar, there are buttons for 'Clear all filters' and 'Hide filters'. The search bar contains the text 'ios'. A note below the search bar states: 'Certain filters have already been applied. To change the results, change the filters. Corporate jobs require English-language skills.' The main content area shows a table of 20 job listings, each with columns for Job Title, Job Function, and Location. All listed jobs are for Software Engineering in Santa Clara Valley.

Job Title	Job Function	Location
iOS QA Engineer	Software Engineering	Santa Clara Valley
Siri iOS Engineer	Software Engineering	Santa Clara Valley
Siri iOS Engineer	Software Engineering	Santa Clara Valley
Siri iOS Engineer	Software Engineering	Santa Clara Valley
Siri iOS Engineer	Software Engineering	Santa Clara Valley
iOS App Reviewer	Marketing	Santa Clara Valley
iOS Power QA Engineer	Software Engineering	Santa Clara Valley
iOS Home App Engineer	Software Engineering	Santa Clara Valley

COURSE DESIGN

- Challenges for this course...which ones to explore?
 - Most likely to be around next year
 - Easiest
 - Most powerful
 - Cost effective
 - Use in industry

COURSE DESIGN

- This class is meant to be a discussion and exploration
 - Please ask questions frequently



COURSE DESIGN

- Expectations
 - You are highly motivated
 - You will do a lot of self-learning to address questions that effect your work
 - You will not wait until the last minute to
 - You will show up and participate in class

COURSE WORK

SYLLABUS

- Class Sessions:

- Lectures
- Discussion
- Demos
- Lab (last 30 minutes)

SYLLABUS

- Lab
 - "White board" ideas for upcoming assignment
 - Address pain points on current assignment
 - Review code
 - Share solutions with peers

SYLLABUS

- Topics to be covered
 - Google Cloud Platform (AppEngine)
 - Firebase
 - CloudKit
 - Cloud Functions/Lambda
 - Swift on the Server
 - ...

COURSE DESIGN

- Assignments
 - Explore/create different backends and strategies
 - Majority of the coursework
- Case Study
 - Teach us something that we haven't seen before related to mobile backends
- Create "something" for a final project

COURSE DESIGN

- Week 1 - Homework 1 Assigned
- Week 2 - Homework 1 Due, Homework 2 Assigned
- Week 3 -
- Week 4 - Homework 2 Due, Homework 3 Assigned
- Week 5 -
- Week 6 - Homework 3 Due, Homework 4 Assigned
- Week 7 -
- Week 8 - Homework 4 Due, Homework 5 Assigned
- Week 9 - Homework 5 Due
- Week 10 - Case Study Showcase
- Final Exam Week - Final Projects Due

COURSE DESIGN

- Assignment 1 (5 points) 5%
- Assignment 2-4 (4*15) 60%
- Assignment 5 (5 points) 5%
- Case Study (15 points) 15%
- Final Project (15 points) 15%

COURSE WORK

- Honor Code
 - All the assignments should be your own work
 - Department policies are strictly enforced
- Citing Resources
 - Cite any resources you use on homework
 - Includes online resources (StackOverflow, BioStars, blogs, GitHub, etc.)
- No late work (except for extraordinary circumstances)

ASSIGNMENTS

ASSIGNMENTS

- Assignments will have 2 parts
 - Tutorial based; add/improve features to existing code
 - New task designed and developed by you

[Why Google](#)[Products](#)[Solutions](#)[Launcher](#)[Pricing](#)[Cu](#)

- ▶ NDB Client Library Reference
- Built-in Third-Party Libraries
- Glossary
- ▶ Command-Line Reference

Concepts

- All Concepts
- An Overview of App Engine
- The Python Runtime Environment
- ▶ Microservices on App Engine
- How Requests are Handled
- How Requests are Routed
- Outbound Requests
- How Instances are Managed
- Access Control
- Application Security
- Cloud Storage Features
- ▶ NDB Client Library Features

Tutorials

[All Tutorials](#)

App Engine > Documentation > Python

Tutorials

Contents

- Firebase Authentication tutorial
- Guestbook tutorial
- Hello World tutorial

Firebase Authentication

- Authenticating users with Firebase
- Verify and store user credentials with s

Guestbook tutorial

ASSIGNMENTS

- There will be varying degrees of integration required with an actual mobile application
- In these cases, the mobile application can be a minimal function product (MFP?) 🤔
- The backend should be the main artifact of the assignment

▶ NDB Client Library Reference

Built-in Third-Party Libraries

Glossary

▶ Command-Line Reference

Concepts

All Concepts

An Overview of App Engine

The Python Runtime Environment

▶ Microservices on App Engine

How Requests are Handled

How Requests are Routed

Outbound Requests

How Instances are Managed

Access Control

Application Security

Cloud Storage Features

▶ NDB Client Library Features

Tutorials

[All Tutorials](#)

[App Engine](#) > [Documentation](#) > [Python](#)

Tutorials

Contents

[Firebase Authentication tutorial](#)

[Guestbook tutorial](#)

[Hello World tutorial](#)

Firebase Authentication

Authenticating users with Firebase

Verify and store user credentials with s

Guestbook tutorial

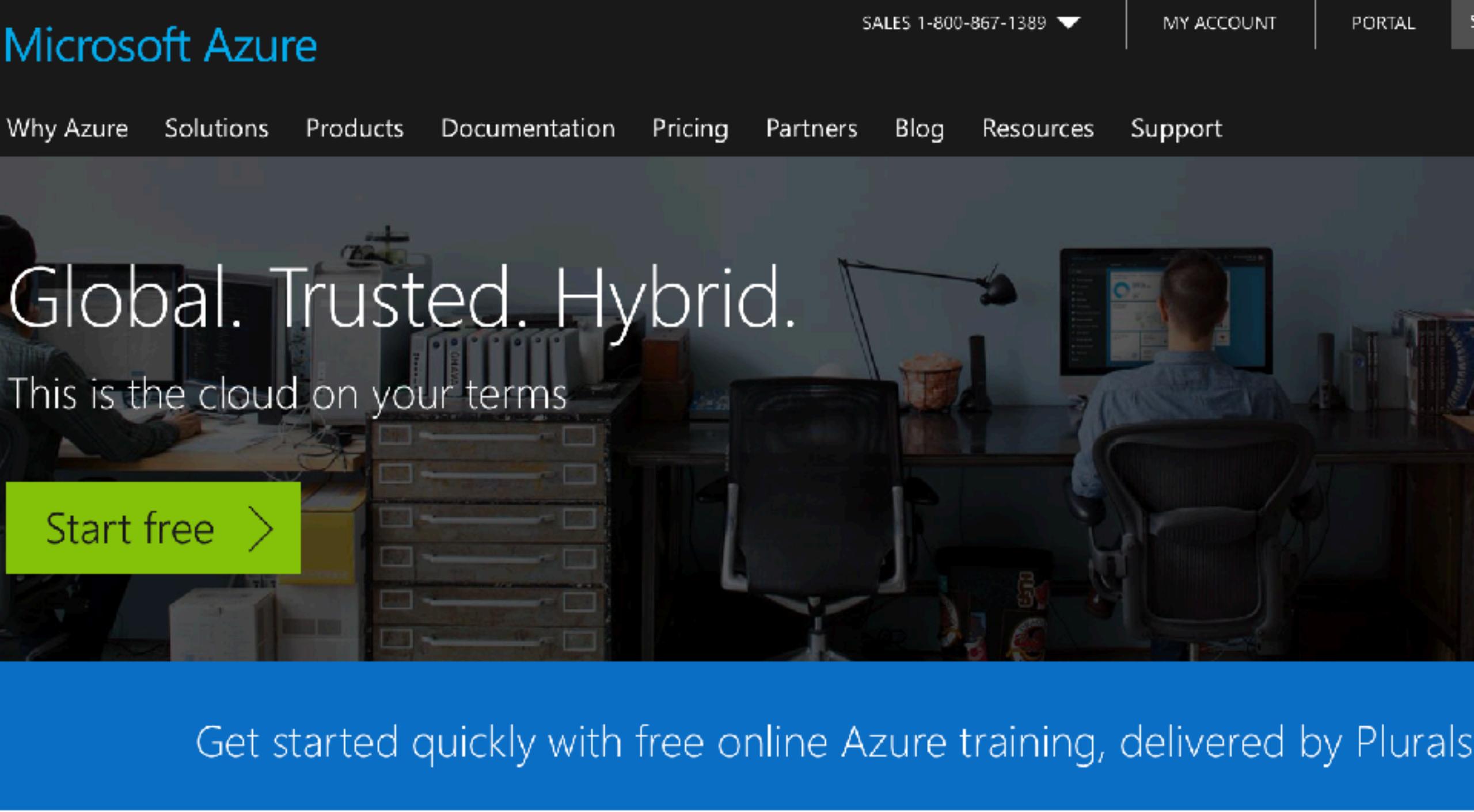
ASSIGNMENTS

- Homework grading
 - Specific requirements to be met
 - Functioning solution on a server
 - Style Points
 - Application design, Coding factoring/style, Readability, Best practices
- Some assignments are over the course of two weeks
 - I will recommend what to cover each week

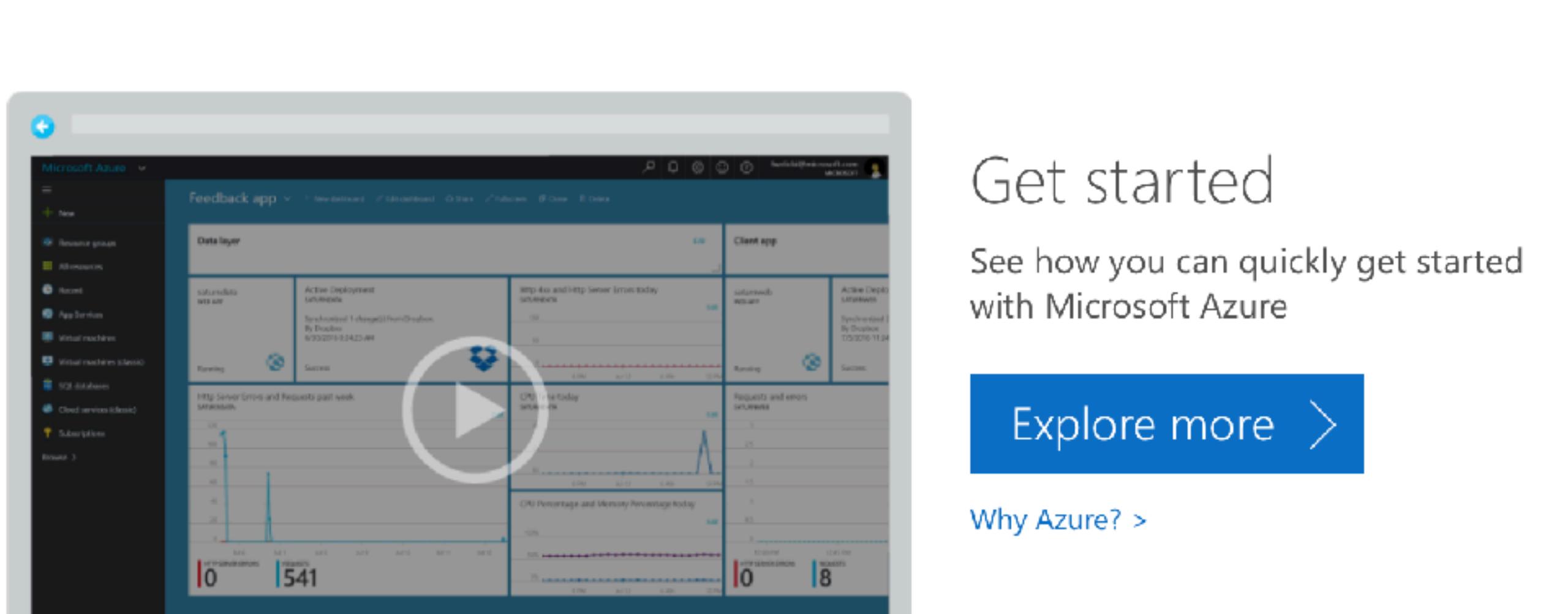
CASE STUDIES

CASE STUDIES

- Teach us all something we did not discuss
- Everyone will present Week 10 in class



The Microsoft Azure homepage features a large banner with the text "Global. Trusted. Hybrid." and "This is the cloud on your terms". Below the banner is a green button labeled "Start free >". The background of the banner shows a person working at a desk with multiple monitors.



A screenshot of the Microsoft Azure portal dashboard. The left sidebar shows navigation options like "New", "Resource groups", "All resources", "Azure", "App Services", "Virtual machines", "Virtual machines (classic)", "SQL databases", "Cloud services (classic)", and "Subscriptions". The main area displays a "Feedback app" overview with sections for "Data layer" and "Client app". It includes metrics such as "HTTP Requests and Response (past week)" (541), "CPU Usage and Memory Percentage (today)" (0%), and "Requests and Errors (past week)" (0). A central video player icon with a play button is overlaid on the dashboard.

Get started
See how you can quickly get started with Microsoft Azure

[Explore more >](#)

[Why Azure? >](#)

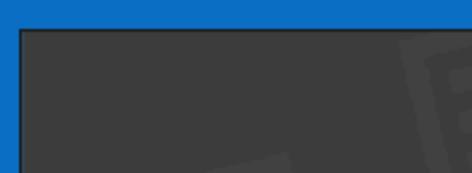
Webinars + Talks



Azure Fridays



Blog



CONTROL STUDIES

POTENTIAL TOPICS

- Up-and-computing technology
- Strategy/technique to deal with problem
- Parse on App Engine
- Microsoft Azure
- Compare costs of different services
- Industry survey
- Case study on company (SnapChat, Spotify, Evernote)

Using Microservices to Encode and Publish Videos at The New York Times

By FLAVIO RIBEIRO , FRANCISCO SOUZA , MAXWELL DA SILVA and THOMPSON MARZAGÃO
NOVEMBER 1, 2016 10:30 AM [Comment](#)

 Email

 Share

 Tweet

 Save

 More

Video publishing at The Times is growing

For the past 10 years, the video publishing lifecycle at The New York Times has relied on vendors and in-house hardware solutions. With our growing investment in video journalism over the past couple of years, we've found ourselves producing more video content every month, along with supporting new initiatives such as 360-degree video and Virtual Reality. This growth has created the need to migrate to a video publishing platform that could adapt to, and keep up with, the fast pace that our newsroom demands and the continued evolution of our production process. Along with this, we needed a system that could continuously scale in both capacity and features while not compromising on either quality or reliability.

A solution

At the beginning of this year, we created a group inside our video engineering team to implement a new solution for the ingest, encoding, publishing and the syndication of our growing library of video content. The main goal of the team was to implement a job processing pipeline that was vendor agnostic and cloud-based, along with being highly efficient, elastic, and, of course, reliable. Another goal was to make the system as easy to use as possible, removing any hurdles that might get in the way of our video producers publishing their work and distributing it to our platforms and third-party partners. To do that, we decided to leverage the

FINAL PROJECTS

FINAL PROJECTS

- Opportunity to use what you've learned in a project you care about
 - Adapt your iOS project
 - MVP for new project
 - ...

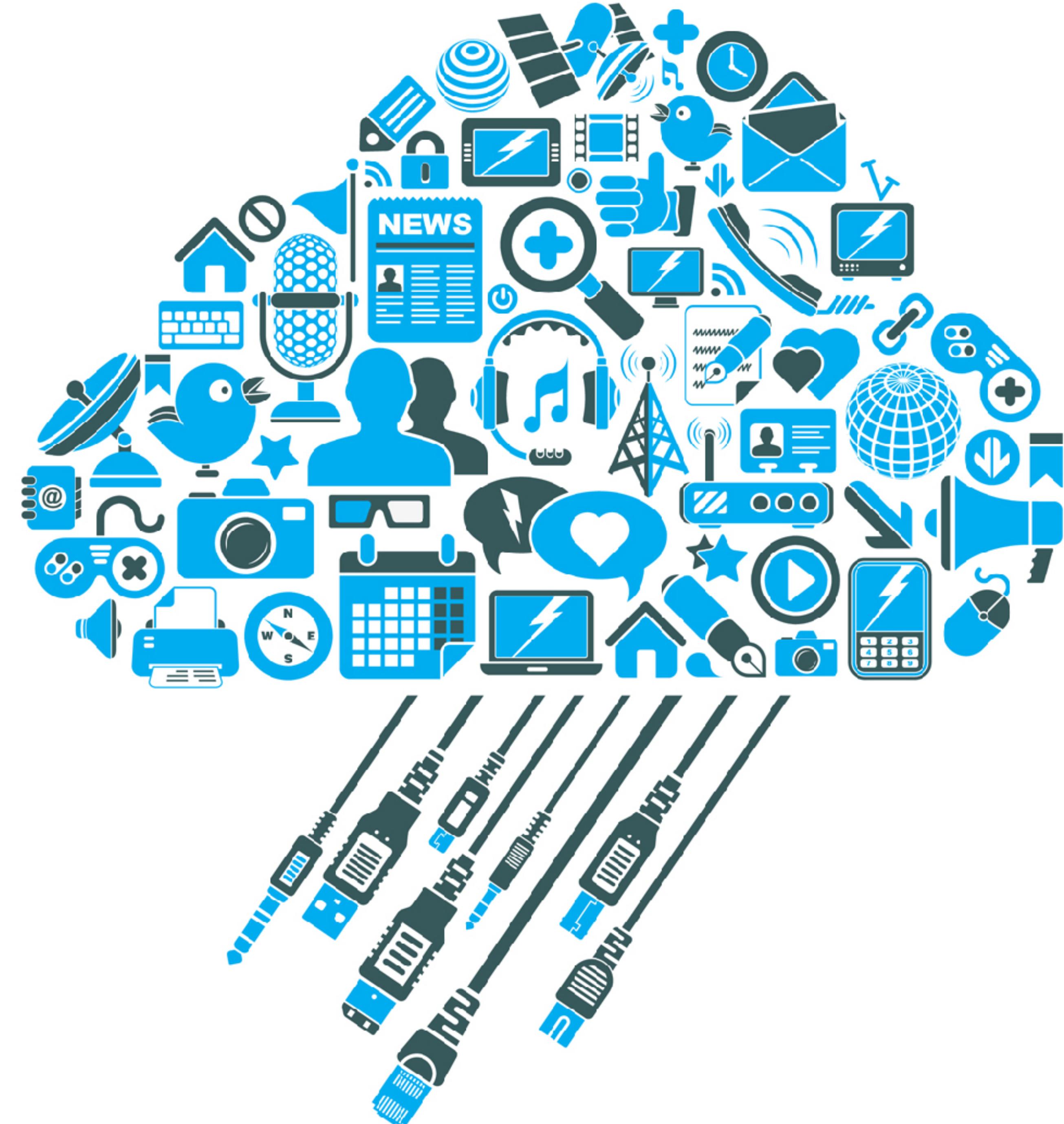
Sound On Sound
by Diomedé, Inc.

be creative again

BACKENDS FOR MOBILE APPLICATION

BACKENDS FOR MOBILE APPLICATION

- Backends is a specific utilization of cloud computing



BACKENDS FOR MOBILE APPLICATION

- Cloud Computing
 - General term used to describe a new class of network based computing over the Internet
 - Collection/group of integrated and networked hardware, software and Internet infrastructure (platform)
 - Use the Internet for communication and transport and provides hardware, software and networking services to clients
 - These platforms hide the complexity and details of the underlying infrastructure from users and applications through API (Applications Programming Interface)

BACKENDS FOR MOBILE APPLICATION

- Platforms
 - Provides on demand services
 - Pay for use and as needed, elastic
 - Scale up and down in capacity and functionalities
 - The hardware and software services are available to general public, enterprises, corporations and businesses markets



BACKENDS FOR MOBILE APPLICATION

- A number of characteristics define cloud data, applications services and infrastructure
 - Remotely hosted - services or data are hosted on remote infrastructure
 - Ubiquitous - services or data are available from anywhere
 - Commodified - a utility



BACKENDS FOR MOBILE APPLICATION

Common Characteristics:

Massive Scale

Resilient Computing

Homogeneity

Geographic Distribution

Virtualization

Service Orientation

Low Cost Software

Advanced Security

Essential Characteristics:

On Demand Self-Service

Broad Network Access

Rapid Elasticity

Resource Pooling

Measured Service

BACKENDS FOR MOBILE APPLICATION

- Cloud service models
 - Software as a service (SaaS)
 - Adobe Creative Cloud
 - Platform as a service (PaaS)
 - S3
 - Infrastructure as a service (IaaS)
 - Compute Engine

BACKENDS FOR MOBILE APPLICATION

- Different Cloud Computing Layers

Application Service (SaaS)	MS Live/ExchangeLabs, IBM, Google Apps; Salesforce.com Quicken Online, Zoho , Cisco
Application Platform	Google App Engine, Mosso, Force.com, Engine Yard, Facebook, Heroku , AWS
Server Platform	3Tera, EC2, SliceHost, GoGrid, RightScale, Linode
Storage Platform	Amazon S3, Dell, Apple, ...

BACKENDS FOR MOBILE APPLICATION

- Different Cloud Computing service layers

Services	Description
Services	Services - Complete business services such as PayPal, OpenID, OAuth, Google Maps, Alexa
Application	Application - Cloud based software that eliminates the need for local installation such as Google Apps, Microsoft Online
Development	Development - Software development platforms used to build custom cloud based applications (PAAS & SAAS) such as SalesForce
Platform	Platform - Cloud based platforms, typically provided using virtualization, such as Amazon ECC, Sun Grid
Storage	Storage - Data storage or cloud based NAS such as CTERA, iDisk, CloudNAS
Hosting	Hosting - Physical data centers such as those run by IBM, HP, NaviSite , etc.

Application Focused

Infrastructure Focused

BACKENDS FOR MOBILE APPLICATION

- The “no-need-to-know” in terms of the underlying details of infrastructure, applications interface with the infrastructure via the APIs.
- The “flexibility and elasticity” allows these systems to scale up and down at will utilizing the resources of all kinds
 - CPU, storage, server capacity, load balancing, and databases
 - The “pay as much as used and needed” type of utility computing and the “always on!, anywhere and any place” type of network-based computing

BACKENDS FOR MOBILE APPLICATION

- Cloud are transparent to users and applications
- Built on clusters of PC servers and off-the-shelf components plus Open Source software combined with in-house applications and/or system software



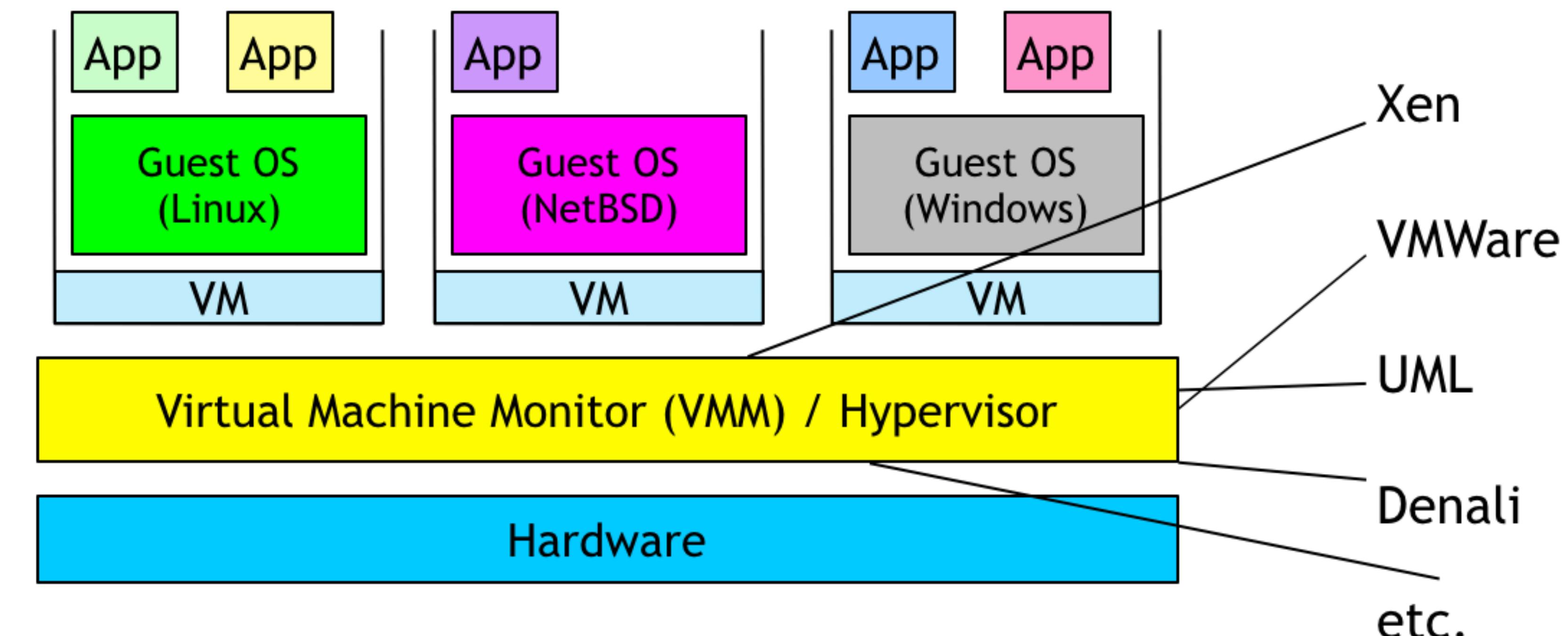
BACKENDS FOR MOBILE APPLICATION

- Cloud are transparent to users and applications
- Built on clusters of PC servers and off-the-shelf components plus Open Source software combined with in-house applications and/or system software

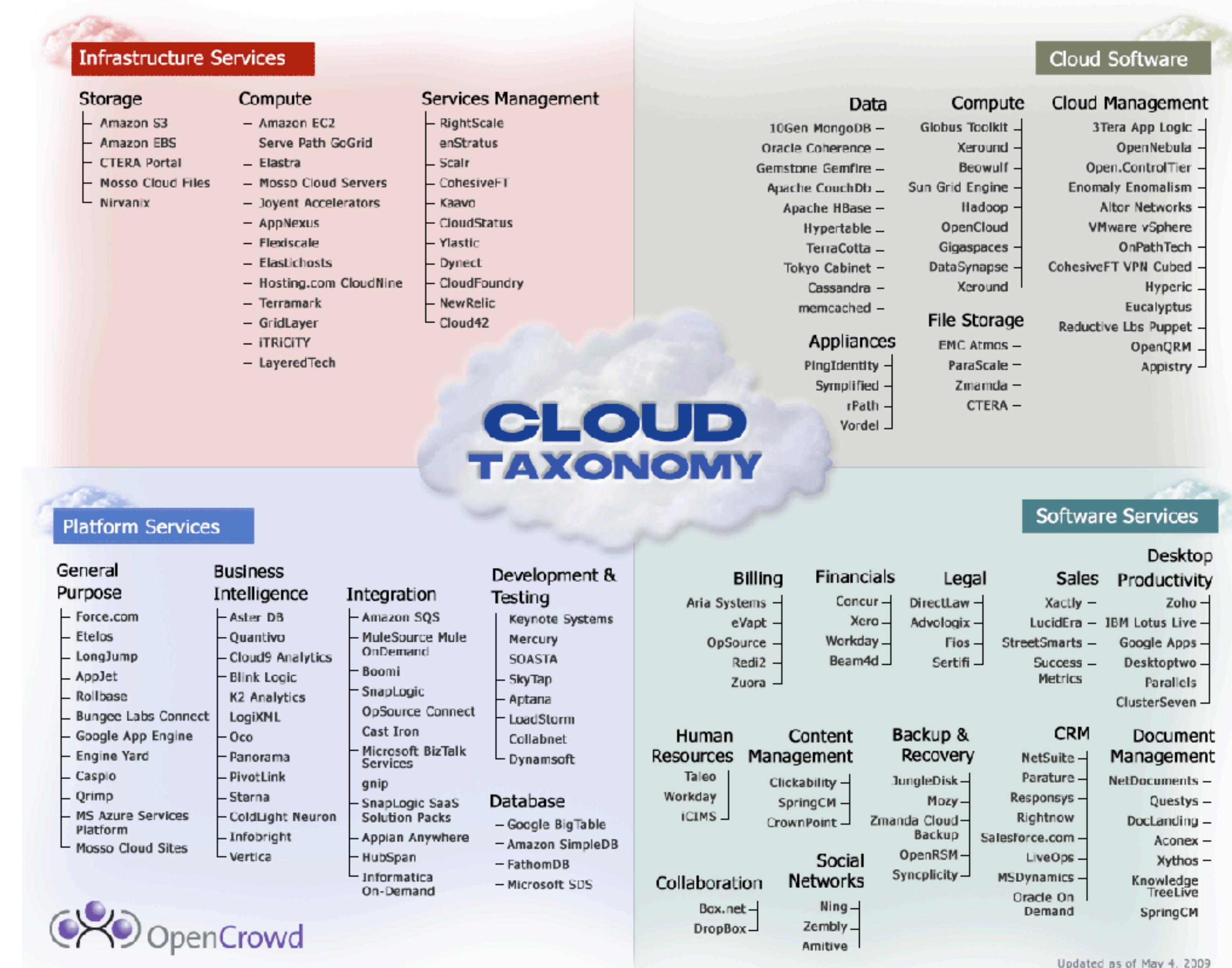


BACKENDS FOR MOBILE APPLICATION

- VM (virtual machine) technology allows multiple virtual machines to run on a single physical machine
- Abstraction of a physical machine



BACKENDS FOR MOBILE APPLICATION



BACKENDS FOR MOBILE APPLICATION

- Advantages
 - Performance, cost, access, collaboration, reliability, bought expertise, security
- Disadvantages
 - Need a network, give up control, monopolization of industry, security



BREAK TIME



GOOGLE CLOUD PLATFORM FOR MOBILE APPLICATION DEVELOPERS

GOOGLE CLOUD PLATFORM

- Google Cloud Platform provides offerings of every type to run code in the cloud
 - Control
 - Ease of use
 - Cost
 - Product

 Google Cloud Platform

Why Google Products Solutions Launcher Pricing Customers Doc >

Build What's Next Better software. Faster.

- ✓ Use Google's core infrastructure, data analytics and machine learning.
- ✓ Secure and fully featured for all enterprises.
- ✓ Committed to open source and industry leading price-performance.

[GO TO CONSOLE](#) [CONTACT SALES](#)

Latest Product Updates
Read all the launch announcements made at Next '17, Google Cloud's annual event.
[VISIT THE BLOG →](#)

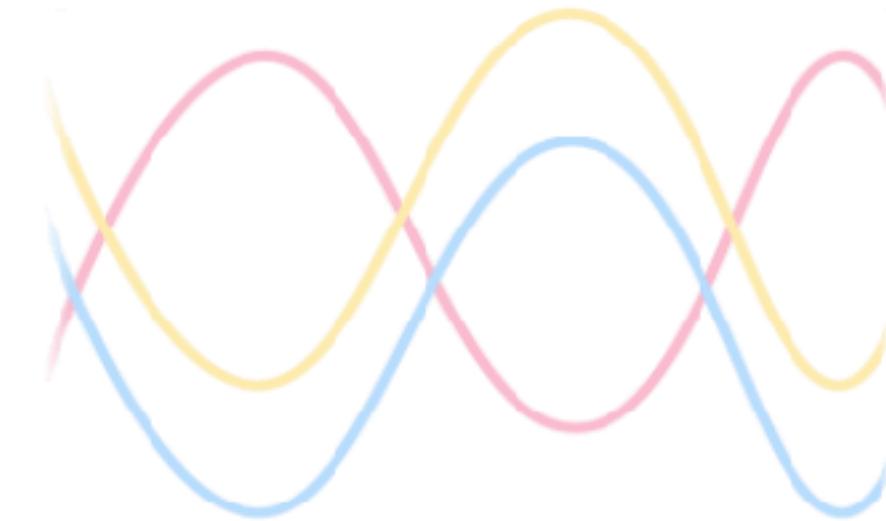
NEXT '17 Keynotes
Learn about our vision for Google Cloud and how customers are embracing the cloud in new and innovative ways.
[WATCH THE KEYNOTES →](#)

Google Cloud I
Learn, engage, and n
immersive, cloud eve
[APPLY TO ATTEND →](#)

Why Google Cloud Platform?

GOOGLE CLOUD PLATFORM

Why Google Cloud Platform?



Future-Proof Infrastructure

Secure, global, high-performance, cost-effective and constantly improving. We've built our cloud for the long haul.

Seriously Powerful Data & Analytics

Tap into big data to find answers faster and build better products.

Serverless, Just Code

Grow from prototype to production to planet-scale, without having to think about capacity, reliability or performance.

GOOGLE CLOUD PLATFORM

GOOGLE HAS A LOT OF PRODUCTS

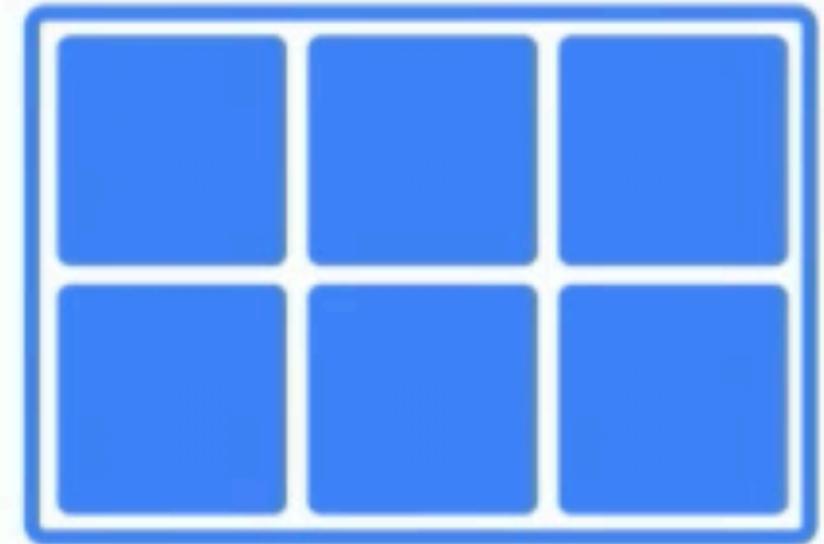


GOOGLE CLOUD PLATFORM

Virtual machines



Containers



Serverless



- Computing options on Google Cloud Platform

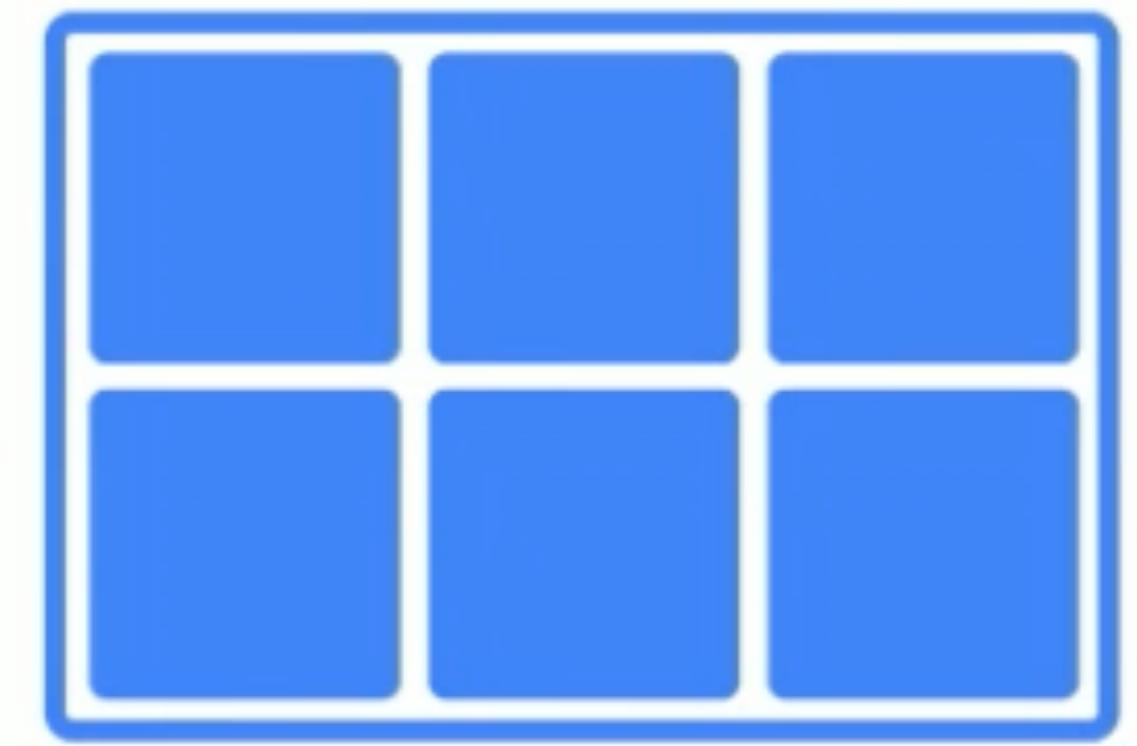
GOOGLE CLOUD PLATFORM

- Virtual Machines
 - 1- 64 processors
 - .6 - 416 GB
 - 65 TB Persistent data
 - 3TB Ram
- Cloud Launcher
 - Prebuilt VM Configurations



GOOGLE CLOUD PLATFORM

- Containers
 - Alternative to VM
 - Container operates on a part of a VM but thinks its a VM
 - Many containers can be used for a workflow
 - Each container would have a specific job



GOOGLE CLOUD PLATFORM



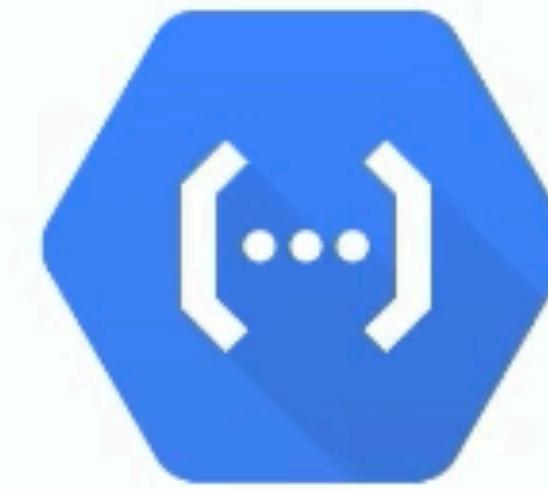
App Engine flexible runtime

- Languages
 - Java 8 / Servlet 3.1 / Jetty 9
 - Python 2.7 & 3.5
 - Node.js,
 - Ruby
 - PHP
 - C



App Engine standard runtime

- Languages
 - Python 2.7
 - Java 7
 - PHP 5.5
 - Go
- Constrained capabilities
- Incredibly rapid scale



Cloud functions

- Javascript / Node.js
- Respond to events

- Serverless (platform as a service)

GOOGLE APP ENGINE

APP ENGINE

- Google App Engine (GAE) is a Platform as a Service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers

GOOGLE APP ENGINE

Build scalable web and mobile backends in any language on Google's infrastructure

 TRY IT FREE

App Engine for All

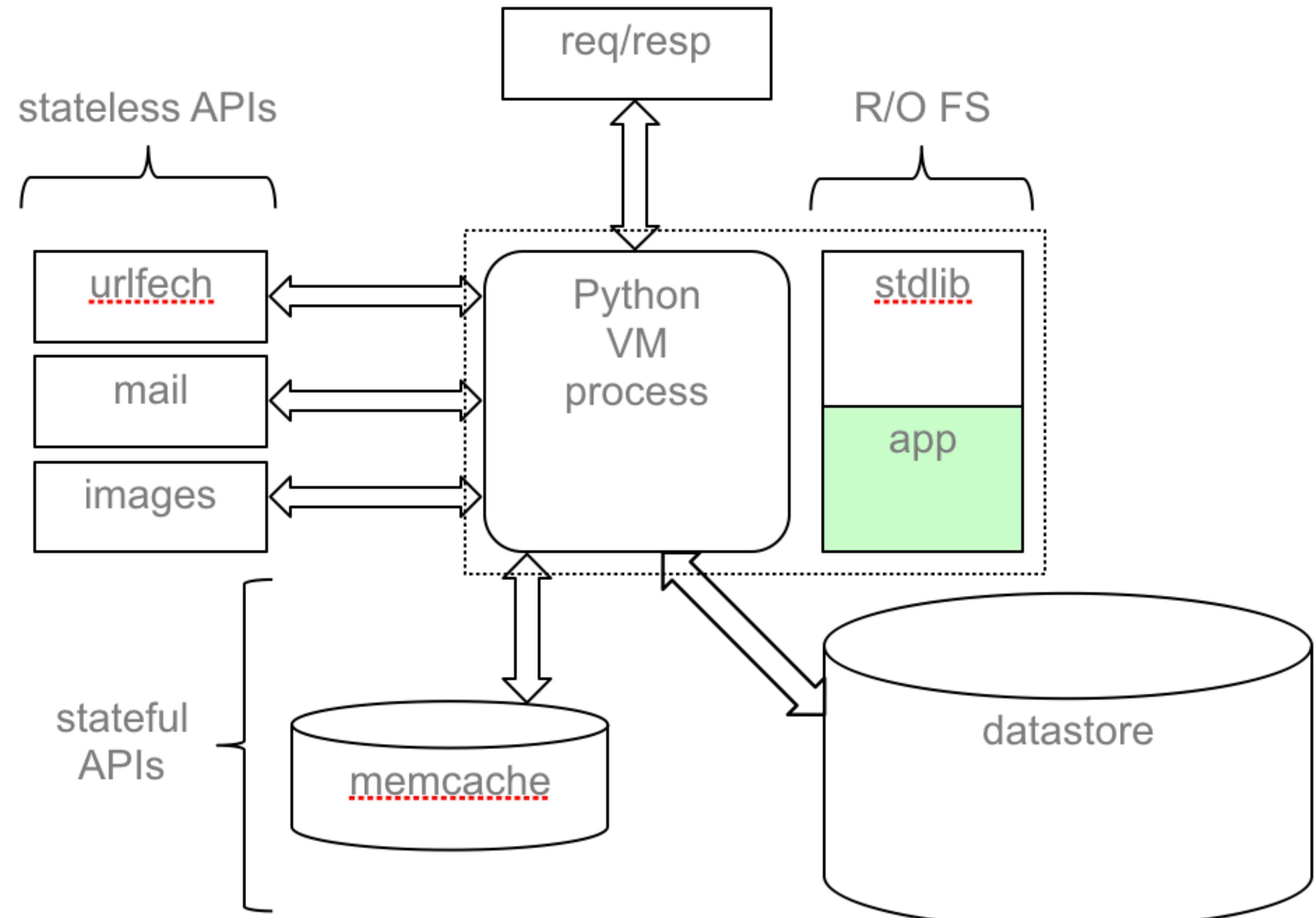
Build modern web and mobile applications on an open cloud platform: bring your own language runtimes, frameworks, and third party libraries. Google App Engine is a fully managed platform that completely abstracts away infrastructure so you focus only on code. Go from zero to planet-scale and see why some of today's most successful companies power their applications on App Engine.

For All Language Communities

Out of the box, App Engine supports Node.js, Java, Ruby, C#, Go, Python, and PHP. Developers from these language communities can be productive immediately in a familiar environment. Add code.

APP ENGINE

- App Engine architecture



APP ENGINE

- App Engine is tuned to do handles HTTP(S) requests
- App configuration is preconfigured
 - No performance tuning needed
 - Everything is built to scale
 - “infinite” number of apps, requests/sec, storage capacity

APP ENGINE

SECURITY

- Constrain direct OS functionality
 - no processes, threads, dynamic library loading
 - no sockets (use urlfetch API)
 - can't write files (use datastore)
 - disallow unsafe Python extensions (e.g. ctypes)
- Limit resource usage
 - Limit 1000 files per app, each at most 1MB
 - Hard time limit of 10 seconds per request
 - Most requests must use less than 300 msec CPU time
 - Hard limit of 1MB on request/response size, API call size, etc.
 - Quota system for number of requests, API calls, emails sent, etc

APP ENGINE

SERVICES

- URLFetch – fetch web resources/services
- Images – manipulate images: resize, rotate, flip, crop
- Google Accounts
- Mail
- XMPP – instant messages
- Task Queue – message queue; allow integration with non-GAPPS
- Datastore – managing data objects
- Blobstore – large files, much larger than objects in datastore, use <key, object> to access

APP ENGINE

DATASTORE

- Entities have a Kind, a Key, and Properties
 - Entity ~~ Record ~~ Python dict ~~ Python class instance
 - Key ~~ structured foreign key; includes Kind
 - Kind ~~ Table ~~ Python class
 - Property ~~ Column or Field; has a type
- Dynamically typed: Property types are recorded per Entity
- Key has either id or name
 - the id is auto-assigned; alternatively, the name is set by app
 - A key can be a path including the parent key, and so on
- Paths define entity groups which limit transactions
 - A transaction locks the root entity (parentless ancestor key)

APP ENGINE

DATASTORE

- Properties are automatically indexed by type+value
 - There is an index for each Kind / property name combo
 - Whenever an entity is written all relevant indexes are updated
 - However Blob and Text properties are never indexed
- This supports basic queries: AND on property equality
- For more advanced query needs, create composite indexes
 - SDK auto-updates index.yaml based on queries executed
 - These support inequalities (<, <=, >, >=) and result ordering
 - Index building has to scan all entities due to parent keys

APP ENGINE

Google Cloud Platform

Try Cloud Platform for free

Country

United States

Acceptances

Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers.

Yes No

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

Yes No

Agree and continue

Google

Access to all Cloud Platform Products
Get everything you need to build and run your apps, websites, and services, including Firebase and the Google Maps API.

\$300 credit for free
Sign up and get \$300 to spend on Google Cloud Platform over the next 12 months.

No autocharge after free trial ends
We ask you for your credit card to make sure you are not a robot. You won't be charged during or after your free trial ends.

Privacy policy

APP ENGINE

- We can get additional credit for the course (if needed), however, we will be consciously trying to architect to the free tier
- Be mindful about costs and old projects



Google Cloud Platform

Try Cloud Platform for free

Google

Country

United States

Acceptances

Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers.

Yes

No

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

Required to continue

Yes

No

Agree and continue

Privacy policy

PYTHON STANDARD ENVIRONMENT

PYTHON STANDARD ENVIRONMENT

- Standard Environment

- Constrained, cheaper, scales better

- Flexible environment

- Full control and access
- Potentially more expensive
- Scaling requires more involvement

Flexible environment

- Full [open source](#) language runtimes.
- Use any framework, library or binary of your choice.
- Code is portable to anywhere that supports Docker containers.
- Most cost-effective for applications that serve traffic continuously.
- [Python 2.7, 3.5](#)

[VIEW DOCS](#)

Standard environment

- Lightweight Python 2.7 runtime is optimized to scale nearly instantaneously to handle huge traffic spikes.
- Does not allow native code, filesystem access or arbitrary network connections. [Learn more](#)
- Uses proprietary APIs to simplify common tasks like database access, queuing and in-memory caching.
- Most cost-effective for applications that have significant periods where they are not serving traffic.
- [Python 2.7](#)

[VIEW DOCS](#)

PYTHON STANDARD ENVIRONMENT

Feature	Flexible environment	Standard environment
Instance startup time	Minutes	Milliseconds
Maximum request timeout	60 minutes	60 seconds
Background threads	Yes	Yes, with restrictions
Background processes	Yes	No
SSH debugging	Yes	No
Scaling	Manual, Automatic	Manual, Basic, Automatic
Writing to local disk	Yes, ephemeral (disk initialized on each VM startup)	No
Modifying the runtime	Yes (through Dockerfile)	No
Automatic in-place security patches	Yes (excludes container image runtime)	Yes
Network access	Yes	Only via App Engine services (includes outbound sockets)
Supports installing third-party binaries	Yes	No
Location	North America, Asia Pacific, or Europe	North America, Asia Pacific, or Europe
Pricing	Based on usage of vCPU , memory , and persistent disks	Based on instance hours

PYTHON STANDARD ENVIRONMENT

- Limited number of third party packages and modules in standard environment
- There is no package manager for Python on App Engine
 - You have to include additional libraries as source code in your standard environment apps
 - Pure Python only (no C)

Name	Default version	Supported versions	Description
django	(None)	"1.2" "1.3" "1.4" "1.5" "1.9"	A full-featured web application framework for Python.
enum	(None)	"0.9.23"	A backport of the enum module introduced in python 3.4
endpoints	(None)	"1.0"	Libraries for building APIs in an App Engine application.
jinja2	(None)	"2.6"	A modern and designer friendly templating language for Python.
lxml	(None)	"2.3" "2.3.5" (alpha)	A Pythonic binding for the C libraries libxml2 and libxslt.
markupsafe	(None)	"0.15" "0.23"	A XML/HTML/XHTML markup safe string for Python.
matplotlib	(None)	"1.2.0"	A 2D plotting library which produces publication-quality figures.
MySQLdb	(None)	"1.2.4b4" (alpha) "1.2.4" (alpha) "1.2.5" (alpha)	A Python DB API v2.0 compatible interface to MySQL.
numpy	(None)	"1.6.1"	A general purpose library for array processing.
PIL	(None)	"1.1.7"	A library for creating and transforming images.
protorpc	"1.0"	"1.0"	A framework for implementing HTTP-based remote procedure call (RPC) services.
pytz	"2016.4"	"2016.4"	A library for cross-platform timezone calculations
crcmod	(None)	"1.7"	A library for generating Cyclic Redundancy Checks (CRC).
PyAMF	(None)	"0.6.1" "0.7.2" (alpha)	A library that provides (AMF) Action Message Format functionality.
pycrypto	(None)	"2.3" "2.6" "2.6.1"	A library of cryptography functions such as random number generation.
setuptools	(None)	"0.6c11"	A library that provides package and module discovery capabilities.
six	(None)	"1.9.0"	Abstract differences between py2.x and py3
ssl	(None)	"2.7" "2.7.11"	The SSL socket wrapper built-in module.
webapp2	"2.3"	"2.3" (deprecated) "2.5.1" "2.5.2"	A lightweight Python web framework.
webob	"1.1.1"	"1.1.1" "1.2.3"	A library that provides wrappers around the WSGI request environment.
werkzeug	"0.11.10"	"0.11.10"	A WSGI utility library.
yaml	"3.10"	"3.10"	A library for YAML serialization and deserialization.

INSTALLING APP ENGINE

INSTALLING APP ENGINE

Download and Install the SDK for App Engine

[SEND FEEDBACK](#)

The SDK for App Engine includes a local development server as well as the tooling for deploying and managing your applications in App Engine.

Go

Java

PHP

Python

Was this page helpful? Let us know how we did:



[SEND FEEDBACK](#)

INSTALLING APP ENGINE

The preferred tooling for managing your App Engine applications in [Python](#) is now the Google Cloud SDK. The Cloud SDK includes a local development server as well as the tooling for deploying and managing your applications.

By downloading, you agree to be bound by the [Terms](#) that govern use of the SDK for App Engine.

To install the SDK for Python:

1. Install [Python version 2.7](#):

[DOWNLOAD AND INSTALL](#)

2. Install and initialize the Google Cloud SDK:

[DOWNLOAD AND INSTALL](#)

3. Run the following command to install the [gcloud component](#) that includes the App Engine extension for Python:

```
gcloud components install app-engine-python
```

4. Optional: Run the following command to install the [Extra Libraries](#) component for Python, which includes the [graphy](#)  and [Django](#) libraries:

```
gcloud components install app-engine-python-extras
```

**USE TO BE A MAC GUY "APP
ENGINE LAUNCHER"**

INSTALLING APP ENGINE

```
    • ○ ● ? gae_practice — more ~/bash_profile — 204×35

# added by Anaconda2 4.3.0 installer
export PATH="/Applications/anaconda/bin:$PATH"

# The next line updates PATH for the Google Cloud SDK.
if [ -f '/Applications/google-cloud-sdk/path.bash.inc' ]; then source '/Applications/google-cloud-sdk/path.bash.inc'; fi

# The next line enables shell command completion for gcloud.
if [ -f '/Applications/google-cloud-sdk/completion.bash.inc' ]; then source '/Applications/google-cloud-sdk/completion.bash.inc'

#
export TERM=xterm-color
export GREP_OPTIONS='--color=auto' GREP_COLOR='1;32'
export CLICOLOR=1

# Aliases
alias clean='rm -- *.o core .*- #*'
alias h="history"
alias rm="rm -i"
alias mv="mv -i"
alias cp="cp -i"
alias m="more"

BLACK="\[\033[0m\]";
DARK_GRAY="\[\033[0;30m\]";
```

APP WALK THROUGH: PHOTO TIMELINE

PHOTO TIMELINE

- The staches and glasses backend from Assignment 6
- App behaviors
 - Post pictures with a comment
 - Retrieve pictures

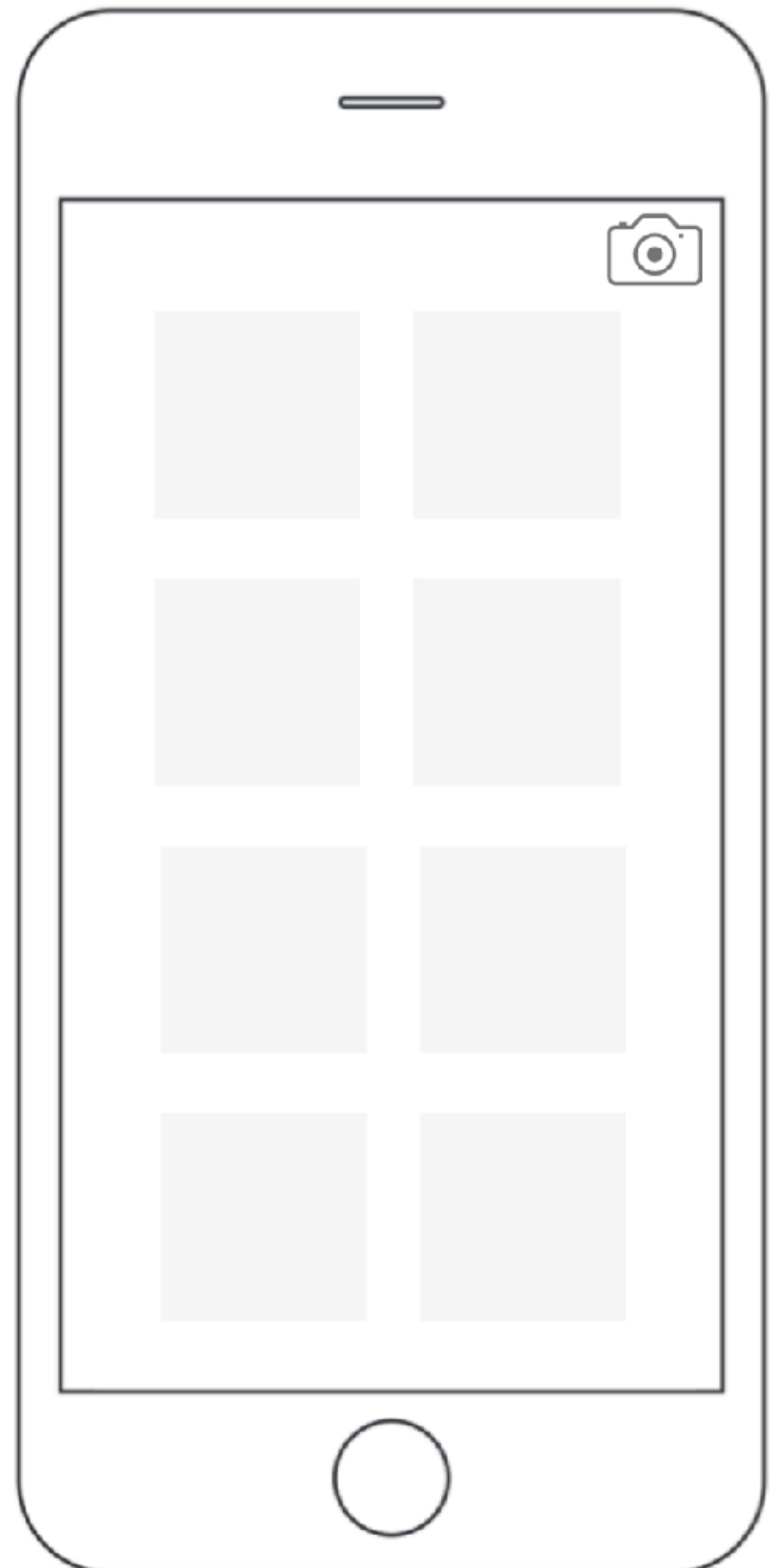
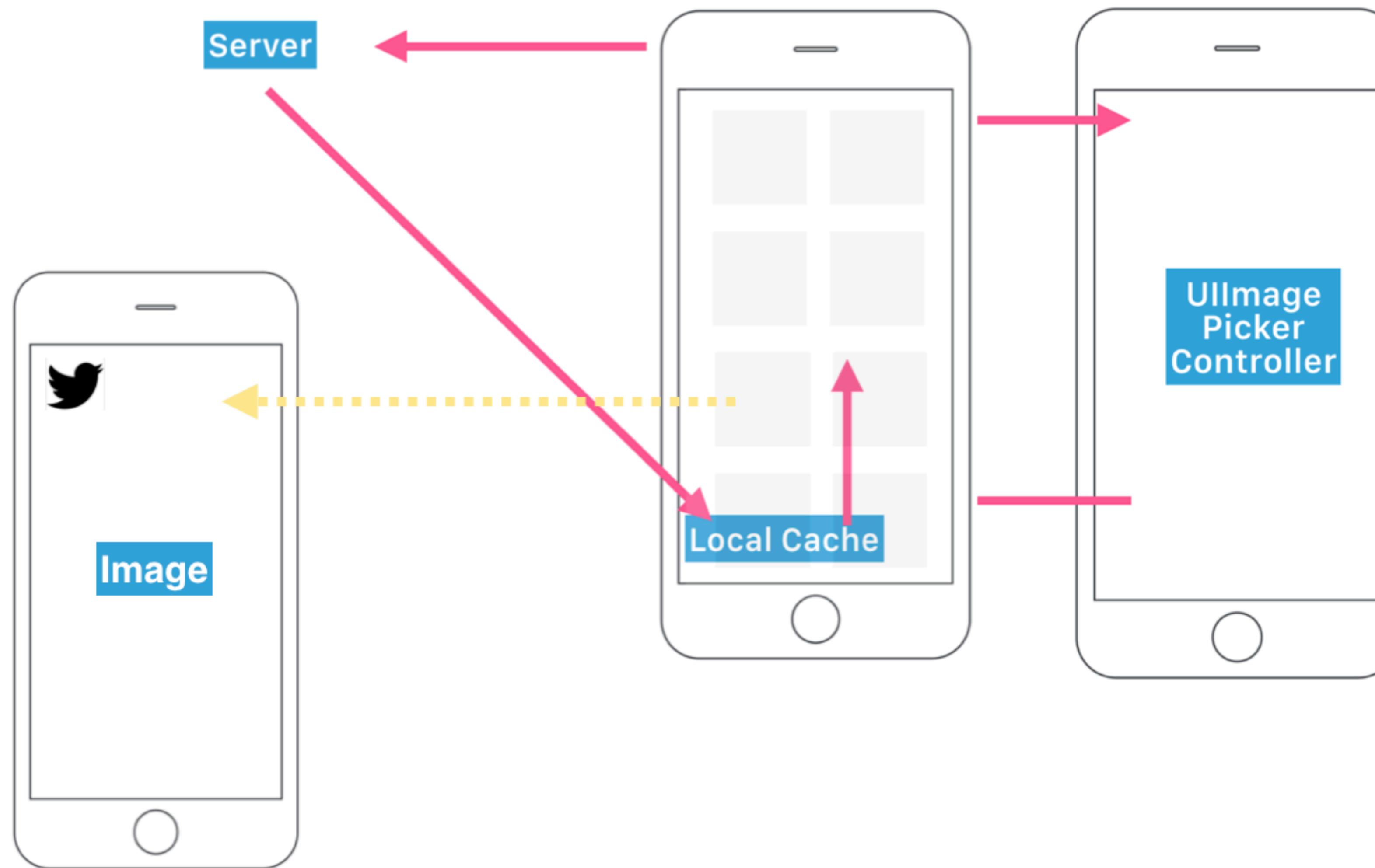


PHOTO TIMELINE



API

API

Get a json list of most recent submitted pictures

```
http://stachesandglasses.appspot.com/user/<USERNAME>/json/
```

See a list of the most recent on a web page (useful for debugging)

```
http://stachesandglasses.appspot.com/user/<USERNAME>/web/
```

Endpoint for posting images to server.

You will need to post using a multipart form (see the example in the project). There is an optional "caption" parameter that you can use.

```
http://stachesandglasses.appspot.com/post/<USERNAME>/
```

PHOTO TIMELINE

API

Get a json list of most recent submitted pictures
<http://--.appspot.com/user/<USERNAME>/json/>

See a list of the most recent on a web page (useful for
debugging
<http://--.appspot.com/user/<USERNAME>/web/>

Endpoint for posting images to server. There is an
optional "caption" parameter that you can use.
[http://--.appspot.com/post/<USERNAME>/](http://--.appspot.com/post/<USERNAME>)

API



Caption: bye

User: tabinks

Date:2016-08-01 14:13:30.511470

<http://stachesandglasses.appspot.com/user/tabinks/web/>



Caption: bye

User: tabinks

Date:2016-08-01 14:06:58.791530

API

```
"user": "default"}, {"date": "2015-04-15 08:55:14.177850", "caption": "Test2", "image_url":  
"image/ahNzfnN0YWNoZXNhbmRnbGFzc2Vzch4LEgRVc2VyIgdkZWZhdWx0DAsSBVB  
"user": "default"}, {"date": "2014-04-29 17:20:39.047800", "caption": "", "image_url":  
"image/ahNzfnN0YWNoZXNhbmRnbGFzc2Vzch0LEgRVc2VyIgdkZWZhdWx0DAsSBVB  
"user": "default"}, {"date": "2014-02-06 02:33:25.947030", "caption": "", "image_url":  
"image/ahNzfnN0YWNoZXNhbmRnbGFzc2Vzch0LEgRVc2VyIgdkZWZhdWx0DAsSBVB  
"user": "default"}, {"date": "2013-05-08 18:11:30.375380", "caption": "test", "image_url":  
"image/ahNzfnN0YWNoZXNhbmRnbGFzc2VzchwLEgRVc2VyIgdkZWZhdWx0DAsSBVB  
"user": "default"}]]}
```

<http://stachesandglasses.appspot.com/user/tabinks/json/>

API



[http://stachesandglasses.appspot.com/image/
ahNzfnN0YWNoZXNhbmRnbGFzc2Vzch4LEgRVc2Vylg
d0YWJpbmtzDAsSBVBob3RvGNOGAww/](http://stachesandglasses.appspot.com/image/ahNzfnN0YWNoZXNhbmRnbGFzc2Vzch4LEgRVc2Vylgd0YWJpbmtzDAsSBVBob3RvGNOGAww/)

API

```
func uploadImage(image: UIImage, caption: String) {
    let boundary = generateBoundaryString()
    let scaledImage = resize(image)
    let imageJPEGData = UIImageJPEGRepresentation(scaledImage, 0.1)

    guard let imageData = imageData else { return }

    // Create the URL, the user should be unique
    let url = NSURL(string: "http://stachesandglasses.appspot.com/post/\(user)/")

    // Create the request
    let request = NSMutableURLRequest(URL: url!)
    request.HTTPMethod = "POST"
    request.setValue("multipart/form-data; boundary=\(boundary)", forHTTPHeaderField: "")

    // Set the type of the data being sent
    let mimetype = "image/jpeg"
    // This is not necessary
    let fileName = "test.png"

    // Create data for the body
    let body = NSMutableData()
    body.appendData("\r\n--\(boundary)\r\n".dataUsingEncoding(NSUTFStringEncoding)!)

    // Caption data
    body.appendData("Content-Disposition:form-data; name=\"caption\"\r\n\r\n".dataUsingEncoding(NSUTFStringEncoding)!)
    body.appendData("CaptionText\r\n".dataUsingEncoding(NSUTFStringEncoding)!)

    // Image data
    body.appendData("--\(boundary)\r\n".dataUsingEncoding(NSUTFStringEncoding)!)
    body.appendData("Content-Disposition:form-data; name=\"image\"; filename=\"\(\(fname))\"\r\n".dataUsingEncoding(NSUTFStringEncoding)!)
    body.appendData("Content-Type: \(mimetype)\r\n\r\n".dataUsingEncoding(NSUTFStringEncoding)!)
    body.appendData(imageData)
    body.appendData("\r\n".dataUsingEncoding(NSUTFStringEncoding)!)

    // Trailing boundary
    body.appendData("\r\n--\(boundary)--\r\n".dataUsingEncoding(NSUTFStringEncoding)!)
}
```

THERE IS A PLAYGROUND FOR WORKING WITH THE API (AND YOUR ASSIGNMENT)

FILE STRUCTURE

PHOTO TIMELINE

- Application files
 - app.yaml - App Engine application settings
 - main.py - Main application
 - models.py - Datastore models
 - index.yaml - Generated files for indexing

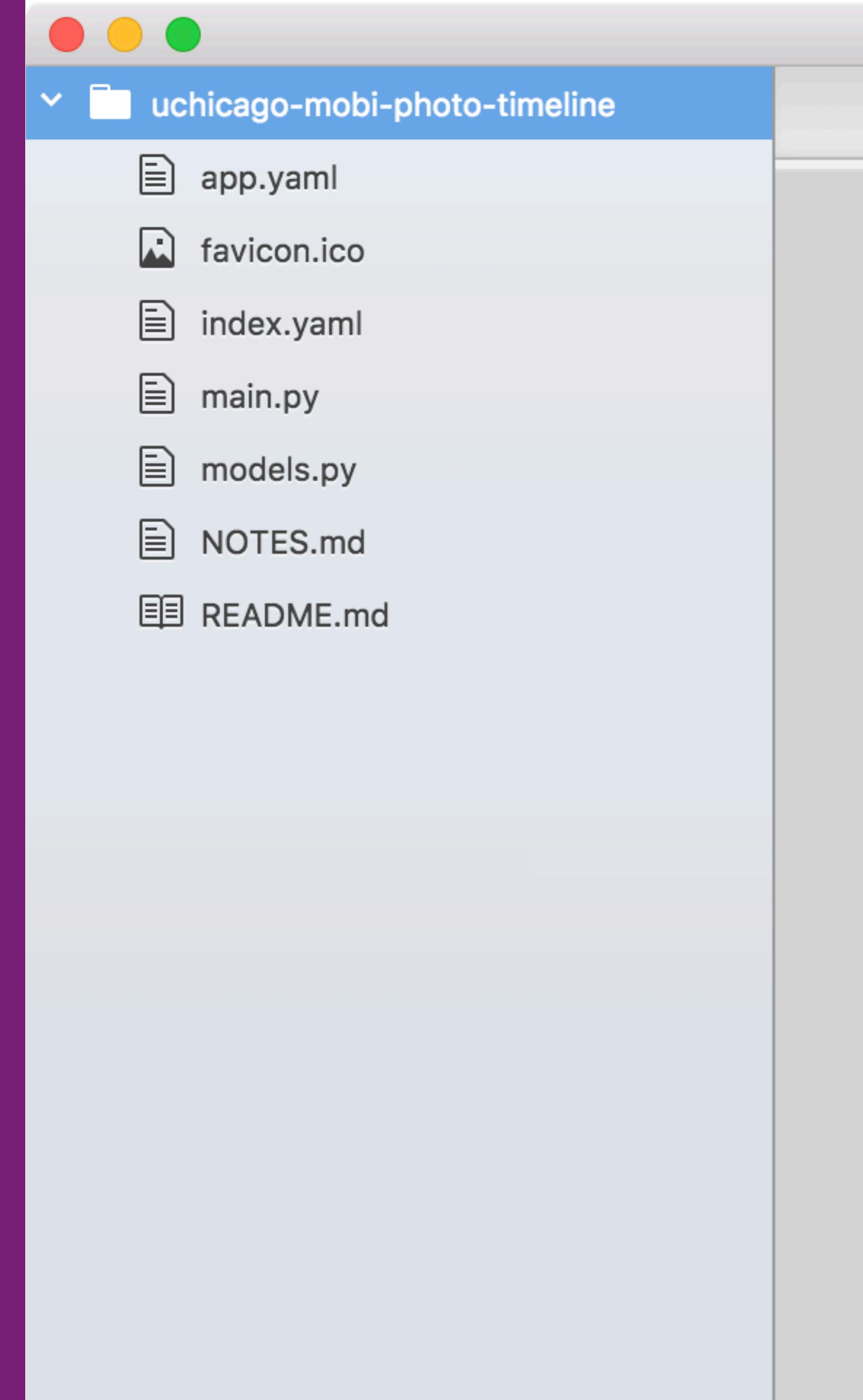


PHOTO TIMELINE

- app.yaml file
- More options
 - Specify machines
 - Global variables
 - ...

HOW SHOULD THE APPLICATION ROUTE REQUESTS

```
application: uchicago-mobi-photo-timeline
runtime: python27
api_version: 1
threadsafe: true
```

ENVIRONMENT VARIABLES

```
# [START handlers]
handlers:
- url: /favicon\.ico
  static_files: favicon.ico
  upload: favicon\.ico

- url:.*/
  script: main.app
# [END handlers]
```

LIBRARIES TO IMPORT

```
# [START libraries]
libraries:
- name: webapp2
  version: latest
- name: jinja2
  version: latest
# [END libraries]
```

FILES TO SKIP WHEN UPLOADING

```
skip_files:
- ^(.*/)?#.*#$
- ^(.*/)?.*~$
- ^(.*/)?.*\.\py[co]$
```

- ^(.*/)?\..*\$

- ^(.*/)?\.txt\$

PHOTO TIMELINE

- models.py
 - Datastore classes
 - Common organization

MODEL CLASS ENTITIES

```
from google.appengine.ext import ndb

class Photo(ndb.Model):
    """Models a user uploaded photo entry"""

    user = ndb.StringProperty()
    image = ndb.BlobProperty()
    caption = ndb.StringProperty()
    date = ndb.DateTimeProperty(auto_now_add=True)

    @classmethod
    def query_user(cls, ancestor_key):
        """Return all photos for a given user"""
        return cls.query(ancestor=ancestor_key).order(-cls.date)

    @classmethod
    def query_user_alternate(cls, ancestor_key):
        """Return all photos for a given user using the gql syntax.
        It returns the same as the above method.
        """

        return ndb.gql('SELECT * '
                      'FROM Photo '
                      'WHERE ANCESTOR IS :1 '
                      'ORDER BY date DESC LIMIT 10',
                      ancestor_key)
```

CLASS METHODS TO QUERY ENTITIES

PHOTO TIMELINE

MODELS.PY

```
@classmethod  
def query_user(cls, ancestor_key):  
    """Return all photos for a given user"""  
    return cls.query(ancestor=ancestor_key).order(-cls.date)
```

```
@classmethod  
def query_user_alternate(cls, ancestor_key):  
    """Return all photos for a given user using the gql syntax.  
    It returns the same as the above method.  
    """  
  
    return ndb.gql('SELECT *'  
                  'FROM Photo '  
                  'WHERE ANCESTOR IS :1 '  
                  'ORDER BY date DESC LIMIT 10',  
                  ancestor_key)
```

SAME THING
DIFFERENT
WAY

PHOTO TIMELINE

MAIN.PY

```
app = webapp2.WSGIApplication([
    ('/', HomeHandler),
    webapp2.Route('/logging/', handler=LoggingHandler),
    webapp2.Route('/post/', , handler=PostHandler),
    webapp2.Route('/user/<user>/<type>', handler=UserHandler)
],
debug=True)
```

PHOTO TIMELINE

MAIN.PY

HTTP://LOCALHOST:8080/LOGGING/

```
app = webapp2.WSGIApplication([
    ('/', HomeHandler),
    webapp2.Route('/logging/', handler=LoggingHandler),
    webapp2.Route('/image/<key>/', handler=ImageHandler),
    webapp2.Route('/post/<user>/', handler=PostHandler),
    webapp2.Route('/user/<user>/<type>/', handler=UserHandler)
],
debug=True)
```

PHOTO TIMELINE

MAIN.PY

```
app = webapp2.WSGIApplication([
    ('/', HTTP://LOCALHOST:8080/IMAGE/ABCDEFG12345/),
    webapp2.Route('/logging/', handler=LoggingHandler),
    webapp2.Route('/image/<key>/', handler=ImageHandler),
    webapp2.Route('/post/<user>/', handler=PostHandler),
    webapp2.Route('/user/<user>/<type>/', handler=UserHandler)
],  
debug=True)
```

HTTP://LOCALHOST:8080/USER/DEFAULT/JSON/

PHOTO TIMELINE

MAIN.PY

```
app = webapp2.WSGIApplication([
    ('/', HomeHandler),
    webapp2.Route('/logging/', handler=LoggingHandler),
```

"""The home page of the app"""

```
class HomeHandler(webapp2.RequestHandler):
```

"""Show the webform when the user is on the home page"""

```
def get(self):
```

PHOTO TIMELINE

```
webapp2.Route('/logging', handler=LoggingHandler),  
webapp2.Route('/image/<key>', handler=ImageHandler),  
webapp2.Route('/post/<user>', handler=PostHandler),
```

"""Handle requests for an image ebased on its key"""

```
class ImageHandler(webapp2.RequestHandler):
```

```
    def get(self, key):
```

"""Write a response of an image (or 'no image') based on a key

```
        photo = ndb.Key(urlsafe=key).get()
```

```
        if photo.image:
```

PHOTO TIMELINE

```
webapp2.Route('/post/<user>/', handler=PostHandler),  
webapp2.Route('/user/<user>/<type>', handler=UserHandler)  
],
```



```
"""Handle activities associated with a given user"""
```

```
class UserHandler(webapp2.RequestHandler):
```



```
"""Print json or html version of the users photos"""
```

```
def get(self, user, type):  
    ancestor_key = ndb.Key("User", user)
```

PHOTO TIMELINE

- The URI (uniform resource identifiers) is a matter of style

```
app = webapp2.WSGIApplication([
    ('/', HomeHandler),
    webapp2.Route('/logging/', handler=LoggingHandler),
    webapp2.Route('/image/<key>/', handler=ImageHandler),
    webapp2.Route('/post/<user>/', handler=PostHandler),
    webapp2.Route('/user/<user>/<type>/',handler=UserHandler)
],  
    debug=True)
```

/users/<user>/json

?user=<user>&style=json

MEMCACHE

MEMCACHE

- App Engine memcache service provides a fast in-memory cache of data
 - Speed up common datastore queries (if the same)
 - Store temporary values

Memcache Overview

Contents ▾

[When to use a memory cache](#)

[Service levels](#)

[Limits](#)

[How cached data expires](#)

...

[Python](#) | [Java](#) | [PHP](#) | [Go](#)

This page provides an overview of the App Engine memcache service. High performance scalable web applications often use a distributed in-memory data cache in front of or in place of robust persistent storage for some tasks. App Engine includes a memory cache service for this purpose. To learn how to configure, monitor, and use the memcache service, read [Using Memcache](#).



Note: The cache is global and is shared across the application's frontend, backend, and all of its services and versions.

MEMCACHE

- Memcache contains key/value pairs
 - The pairs in memory at any time change as items are written and retrieved from the cache
 - Value evicted when memory runs low
 - Expire policy

Memcache Overview

Contents ▾

[When to use a memory cache](#)

[Service levels](#)

[Limits](#)

[How cached data expires](#)

...

[Python](#) | [Java](#) | [PHP](#) | [Go](#)

This page provides an overview of the App Engine memcache service. High performance scalable web applications often use a distributed in-memory data cache in front of or in place of robust persistent storage for some tasks. App Engine includes a memory cache service for this purpose. To learn how to configure, monitor, and use the memcache service, read [Using Memcache](#).



Note: The cache is global and is shared across the application's frontend, backend, and all of its services and versions.

MEMCACHE

- The cache is global and is shared across the application's frontend, backend, and all of its services and versions.

Memcache Overview

Contents ▾

[When to use a memory cache](#)

[Service levels](#)

[Limits](#)

[How cached data expires](#)

...

[Python](#) | [Java](#) | [PHP](#) | [Go](#)

This page provides an overview of the App Engine memcache service. High performance scalable web applications often use a distributed in-memory data cache in front of or in place of robust persistent storage for some tasks. App Engine includes a memory cache service for this purpose. To learn how to configure, monitor, and use the memcache service, read [Using Memcache](#).



Note: The cache is global and is shared across the application's frontend, backend, and all of its services and versions.

MEMCACHE

- Memcache pattern is similar to other cache patterns
- Fallback to generating the data

```
def get_data():
    data = memcache.get('key')
    if data is not None:
        return data
    else:
        data = query_for_data()
        memcache.add('key', data, 60)
    return data
```

MEMCACHE

```
# Add a value if it doesn't exist in the cache  
# with a cache expiration of 1 hour.  
memcache.add(key="weather_USA_98105", value="raining", time=3600)  
  
# Set several values, overwriting any existing values for these keys.  
memcache.set_multi(  
    {"USA_98115": "cloudy", "USA_94105": "foggy", "USA_94043": "sunny"},  
    key_prefix="weather_",  
    time=3600  
)  
  
# Atomically increment an integer value.  
memcache.set(key="counter", value=0)  
memcache.incr("counter")  
memcache.incr("counter")  
memcache.incr("counter")
```

MEMCACHE

- Best practices
 - Handle memcache API failures gracefully
 - Use the batching capability of the API
 - Distribute load across your memcache keyspace.
 - Having a single or small set of memcache items represent a disproportionate amount of traffic will hinder your app from scaling

MEMCACHE

```
def get_data(user):
    """Get data from the datastore only if we don't have it cached"""
    key = user + "_photos"
    data = memcache.get(key)
    if data is not None:
        logging.info("Found in cache")
        return data
    else:
        logging.info("Cache miss")
        ancestor_key = ndb.Key("User", user)
        data = Photo.query_user(ancestor_key).fetch(100)
        if not memcache.add(key, data, 3600):
            logging.info("Memcache failed")
    return data
```

```
# Clear the cache (the cached version is going to be outdated)
key = user + "_photos"
memcache.delete(key)
```

LOCAL DEVELOPMENT AND LOGGING

DEVELOPMENT AND LOGGING

- Different levels of logging
- Used to filter logs

```
FROM GOOGLE.APPENGINE.API IMPORT MEMCACHE

class LoggingHandler(webapp2.RequestHandler):
    """Demonstrate the different levels of logging"""

    def get(self):
        logging.debug('This is a debug message')
        logging.info('This is an info message')
        logging.warning('This is a warning message')
        logging.error('This is an error message')
        logging.critical('This is a critical message')

    try:
        raise ValueError('This is a sample value error.')
    except ValueError:
        logging.exception('A example exception log.')

    self.response.out.write('Logging example.')
```

DEVELOPMENT AND LOGGING

```
# Download the logs from a running application  
% gcloud app logs read -s default
```

PHOTO TIMELINE

```
# Test your application from the command line using `curl`
```

```
curl -X GET http://localhost:8080/
```

```
curl -X GET http://localhost:8080/user/default/json/
```

```
curl -X POST -H "Content-Type: multipart/form-data" -F  
caption='curl' -F "image=@kitten.jpg" http://localhost:  
8080/post/lolakitty/
```

POTENTIAL PROBLEMS WITH THE ARCHITECTURE

PHOTO TIMELINE

FORESEEABLE PROBLEMS

- This solution will not scale very well once we have more users
- All the data is in a single entity, Photos, has to be searched to retrieve the photos for a given user name
 - This could be very costly

PHOTO TIMELINE

FORESEEABLE PROBLEMS

- No security
 - Anyone can post or retrieve with just a username
 - No way to uniquely identify a user
- No way to delete embarrassing photos

ASSIGNMENT 2

ASSIGNMENT 2

- Part 1
 - Address the problems we discussed in class
- Part 2
 - Develop a mobile analytics backend

PART 1

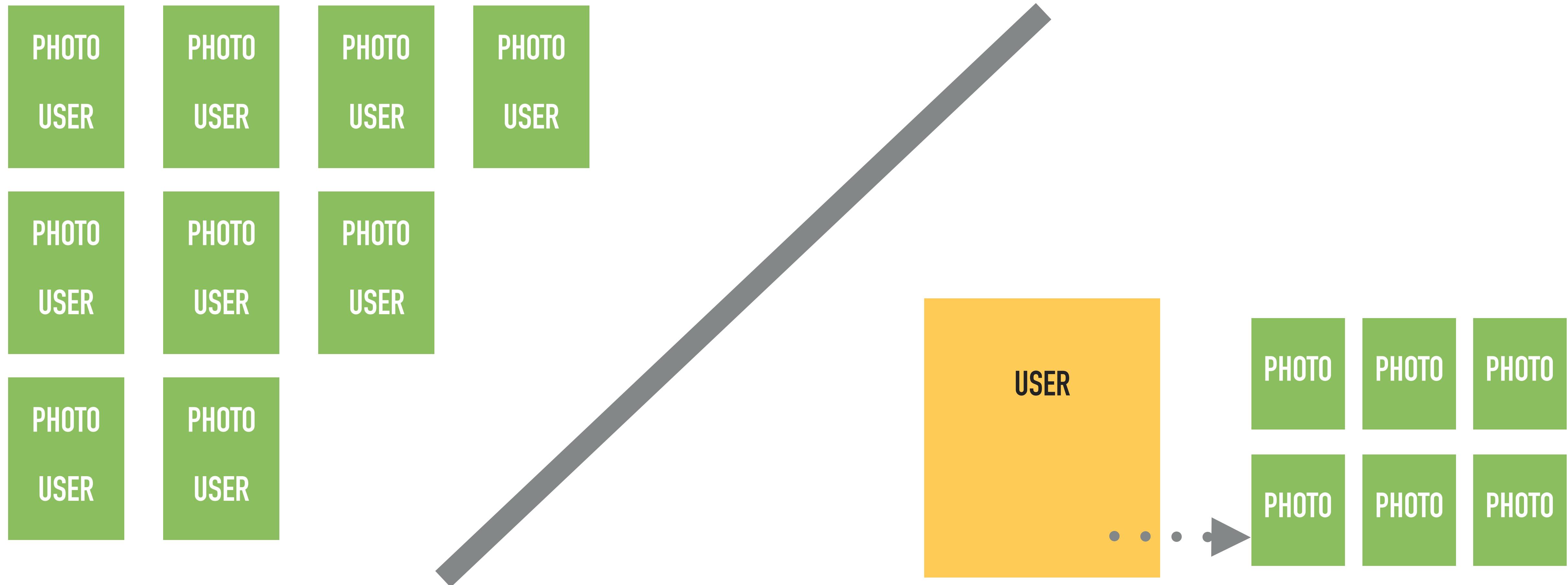
ASSIGNMENT 2 - PART 1

ADDRESSING THE ISSUES

- Address some of the concerns we have with our existing solution
 - Changes to model
 - Increased security
 - Improved functionality

ASSIGNMENT 2 - PART 1

DATA MODEL



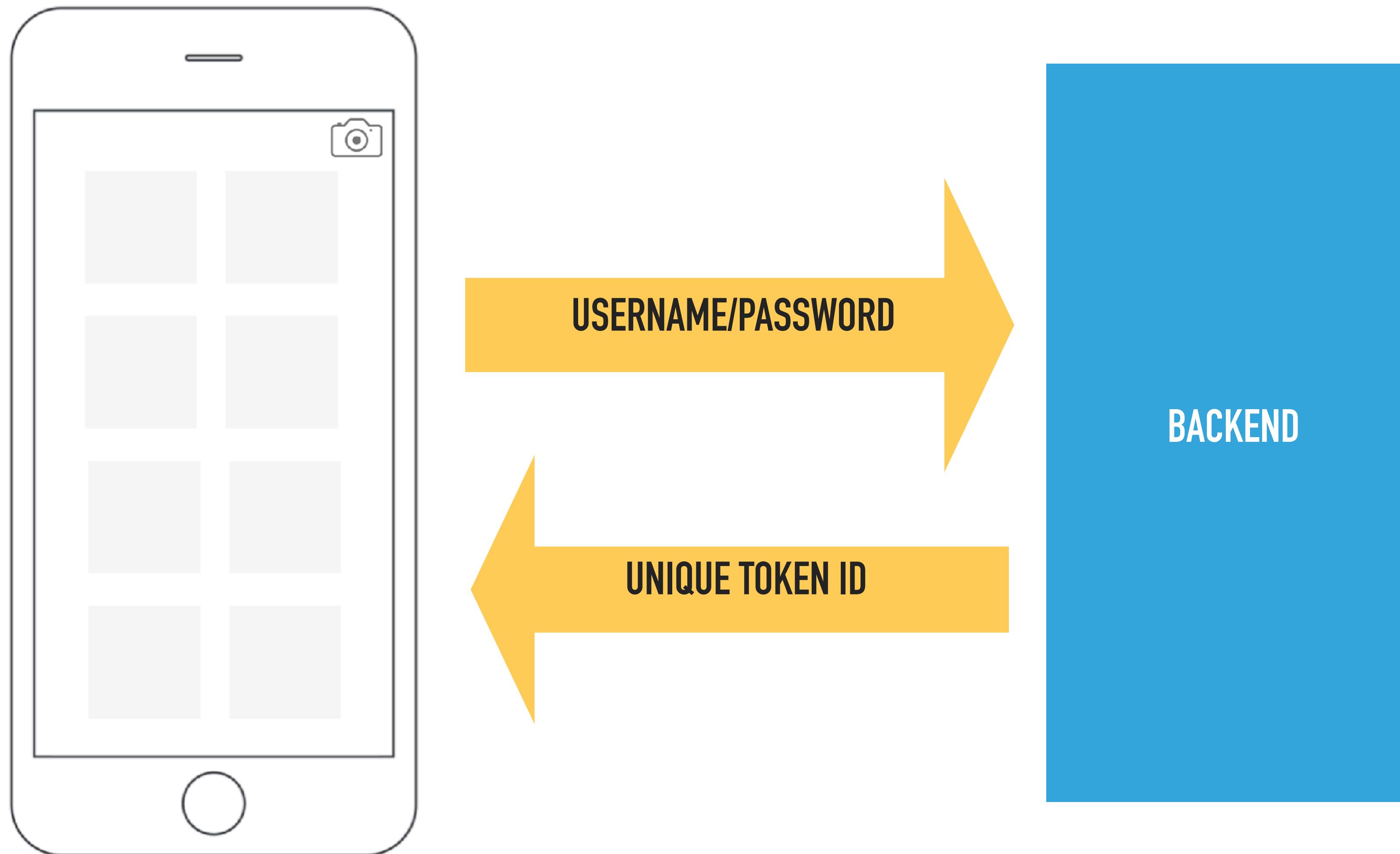
ASSIGNMENT 2 - PART 1

DATA MODEL

- Add a "User" model to the application
 - name (ex. johnny)
 - email (ex. johnny@xx.com)
 - unique_id (ex. can be user entity key)
 - photos (all the photos uploaded to this account)
- Update all the queries to take advantage of the new User model
- Update the Photo model to reflect the new User model

ASSIGNMENT 2 - PART 1

SECURITY



ASSIGNMENT 2 - PART 1

SECURITY

```
# Get a json list of most recent submitted pictures  
http://--.appspot.com/user/authenticate/?  
username=XXX&password=XXXX  
  
# This should return a unique id token that would be stored  
# on the device
```

ASSIGNMENT 2 - PART 1

SECURITY

```
# Get a json list of most recent submitted pictures  
http://--.appspot.com/user/<USERNAME>/json/?id_token=XXXX
```



```
# See a list of the most recent on a web page (useful for  
debugging  
http://--.appspot.com/user/<USERNAME>/web/?id_token=XXXX
```



```
# Endpoint for posting images to server. There is an  
optional "caption" parameter that you can use.  
http://--.appspot.com/post/<USERNAME>/?id_token=XXXX
```

ASSIGNMENT 2 - PART 1

FUNCTIONALITY

```
# Add ability to delete a photo  
http://--.appspot.com/image/<key>/delete/?id_token=XXX
```

PART 2

ASSIGNMENT 2 - PART 2

MOBILE ANALYTICS PLATFORM

- Create a mobile analytics platform
 - Track unique users - visits, time duration
 - Track events - clicks, navigation
 - Touch heat map - all touches (for a reason)
 - Daily summary email

ASSIGNMENT 2 - PART 2

MOBILE ANALYTICS PLATFORM

- You will design all the aspects of the analytics platform using Google App Engine
- Simple companion iOS app (reuse old app)

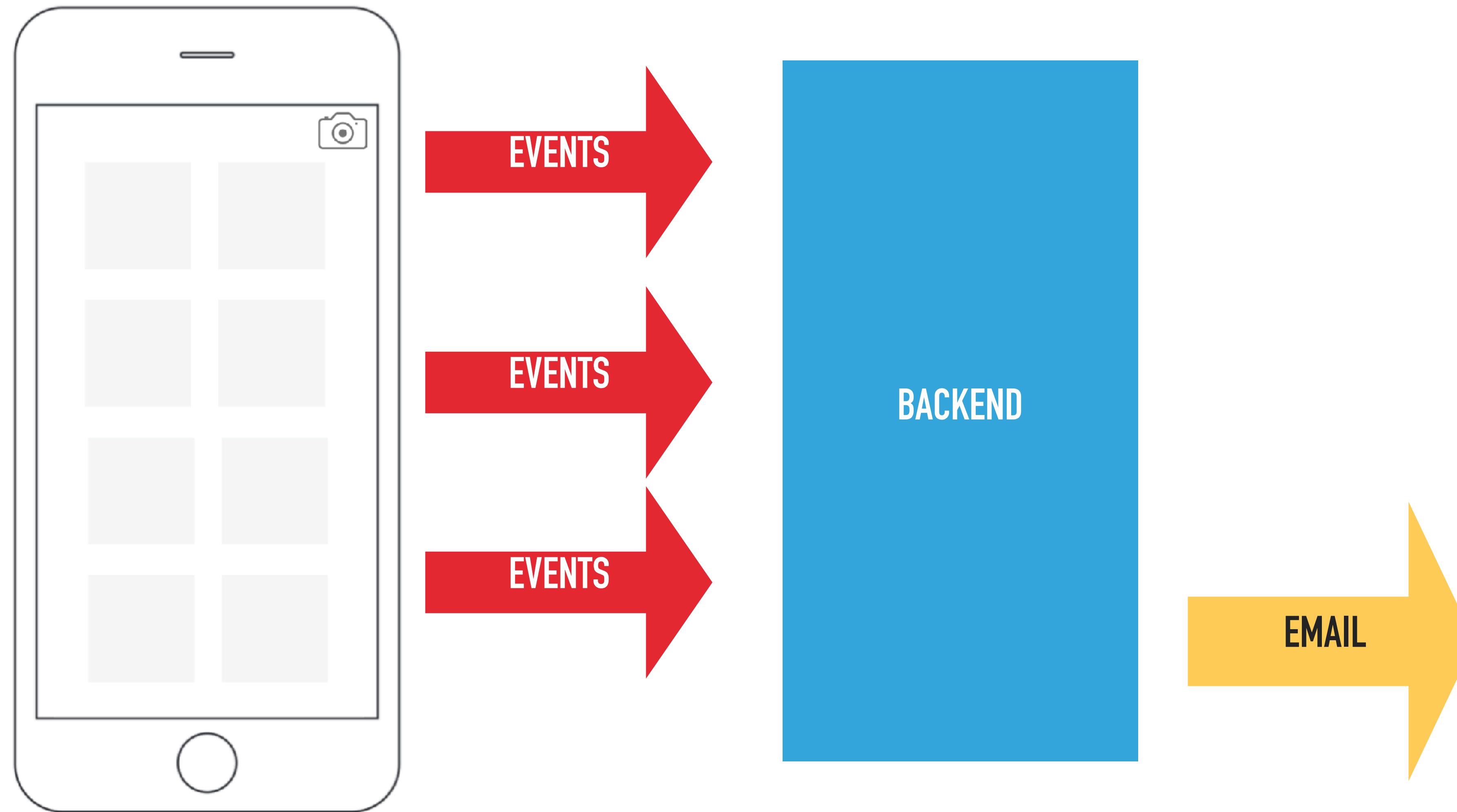
ASSIGNMENT 2 - PART 2

MOBILE ANALYTICS PLATFORM

- Next week
 - Sharding
 - Cron
 - Task queue
 - Microservices
 - Email

ASSIGNMENT 2 - PART 1

SECURITY





MPCS 51033 • SPRING 2017 • SESSION 1

BACKENDS FOR MOBILE APPLICATIONS