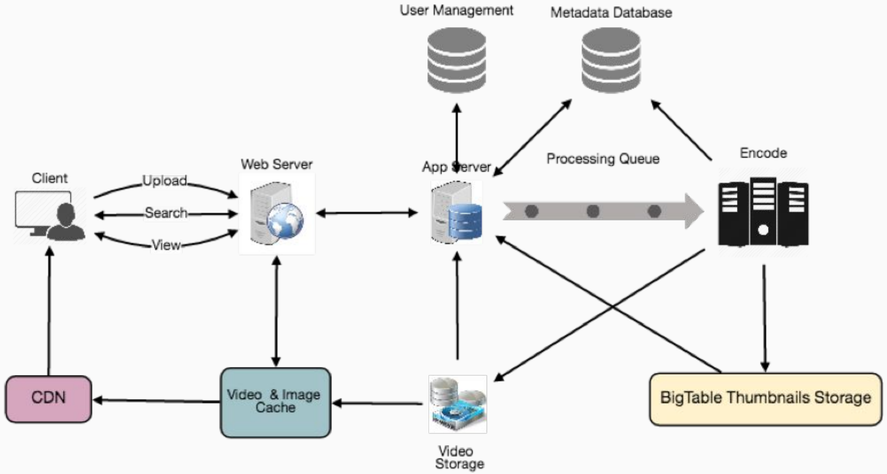
Distributed Systems Design

System Design

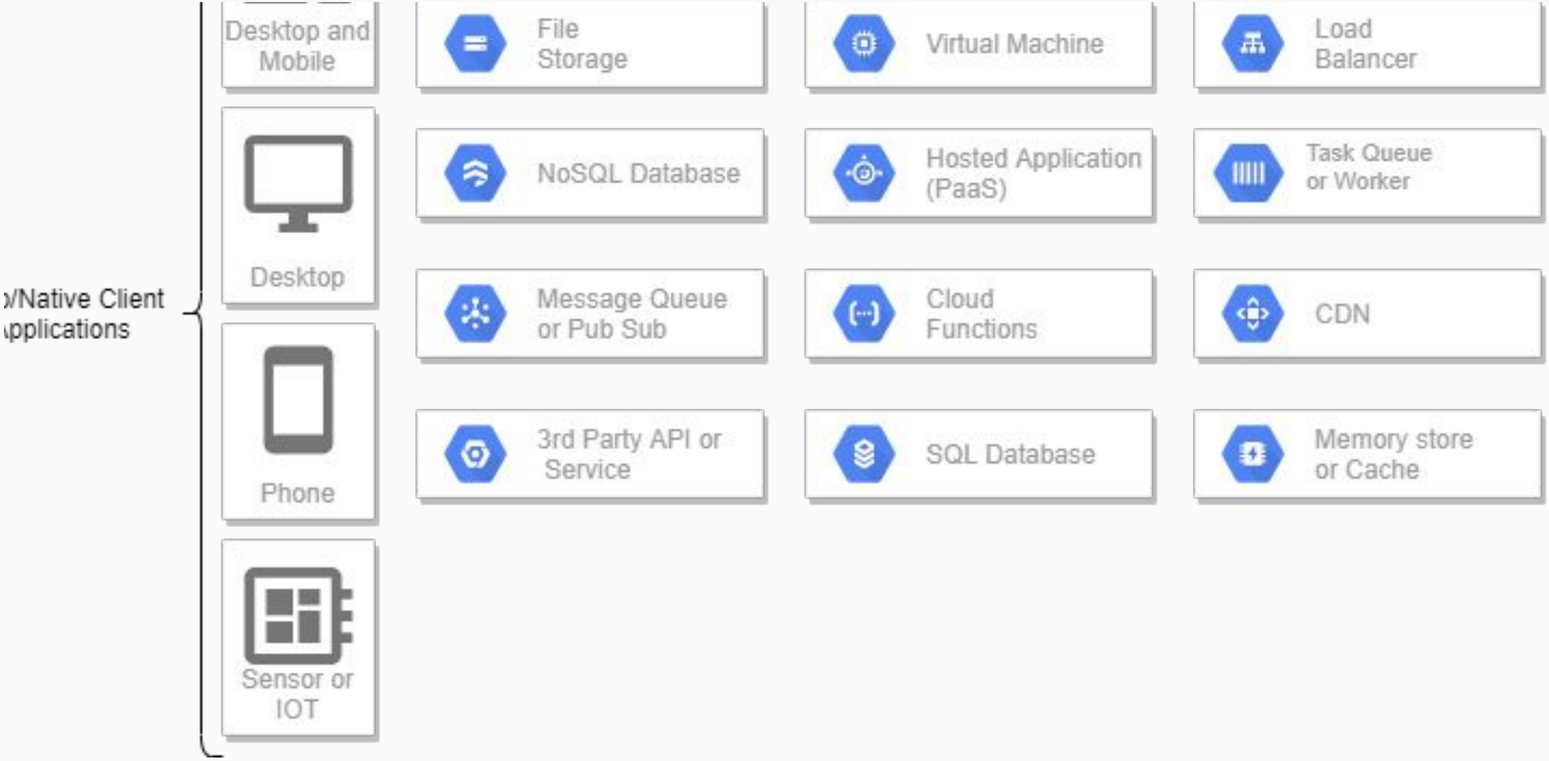
- Process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.
- Applies:
  - Cloud Design Patterns
  - Distributed System Architectural Patterns
- Concepts:
  - Database replication
  - Database partitioning
  - Map Reduce
  - Microservices
  - Message Brokers
  - Queuing
  - Caching
  - Event Stream
  - Load Balancing
- Case Study: Tinder



System Components

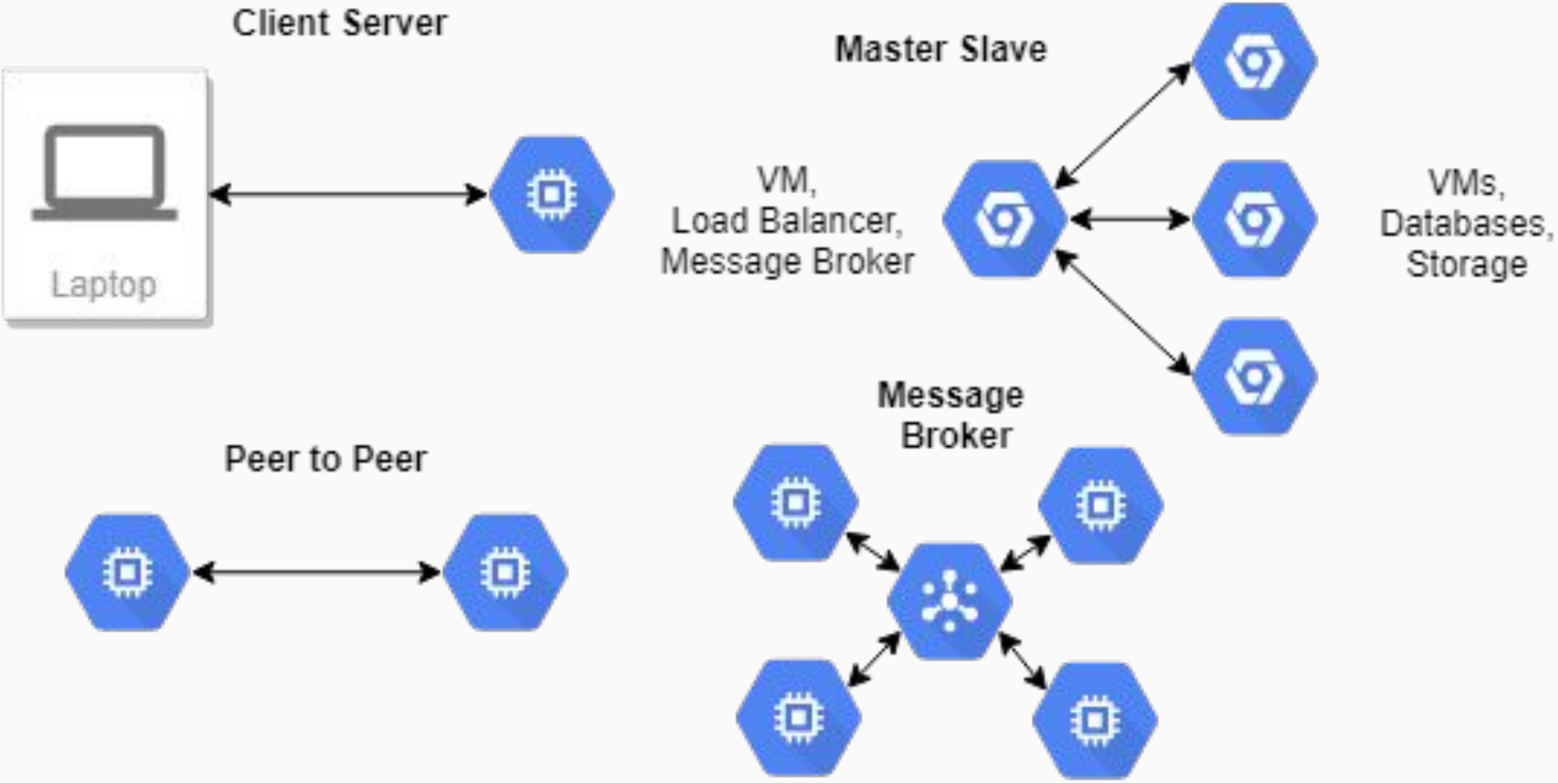
1. **File Storage:** Services for hosting, optimizing and distributing media files
2. **Static Hosting:** Github pages, firebase hosting, netlify
3. **Compute:**
    a. Cloud Virtual Machine: Eg Digital Ocean Droplets, Google Compute Engine, AWS EC2
    b. Cloud Function: Firebase Function, AWS Lambda, Netlify Functions
    c. App/Container Hosting or Platform as a Service (PaaS): Heroku, KintoHub,
    d. VPS/Shared/Managed/Dedicated Hosting
4. **External Systems, 3rd Party APIs/Service:** Eg. Google Maps, Facebook API, YouTube API
5. **Databases**
    a. No/SQL: MongoDB, Firebase Firestore/Real Time Database, Harper DB
    b. Relational: PostGres, MySQL, SQL Server, MariaDB
    c. Graph: TigerDB, Neo4j, DGraph
6. **Message Queue/Pub Sub:** Services for receiving, queueing and delivering messages to multiple components
7. **Memory Store /Caching:** Memcached, Redis
8. **Load Balancer:** A component that receives requests and distributes them evenly among other components
9. **Task Queue/Worker:** Systems for managing scheduled tasks or background workers Eg: Google Cloud Task, Celery, Redis Queue
10. **Content Delivery Network:** Fast high availability hosting of assets and static web sites

Conventions



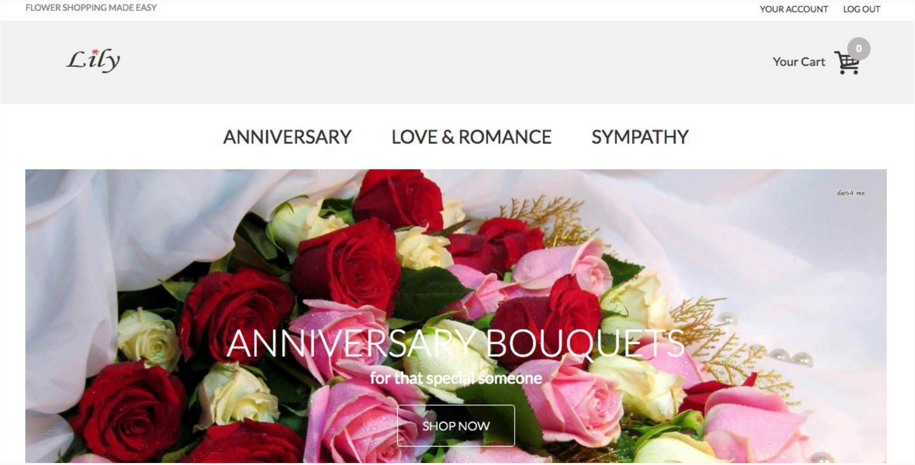
Draw.io Template

Illustrating Distributed Architectural Patterns



Case Study: lily’s

- Lily’s Flower E-commerce Application
- Requirements:
  - Web and Mobile clients
  - Product browsing
  - Order fulfillment
  - Product analytics dashboard
  - Location Based coupons (mobile only)
- Non functional requirements:
  - High volume real time coupon delivery
  - Fast read writes of coupon delivery
  - Coupon delivery service should be independent of web app
  - Prototype deployment < 1000 users
  - Testing load < 100 requests per min

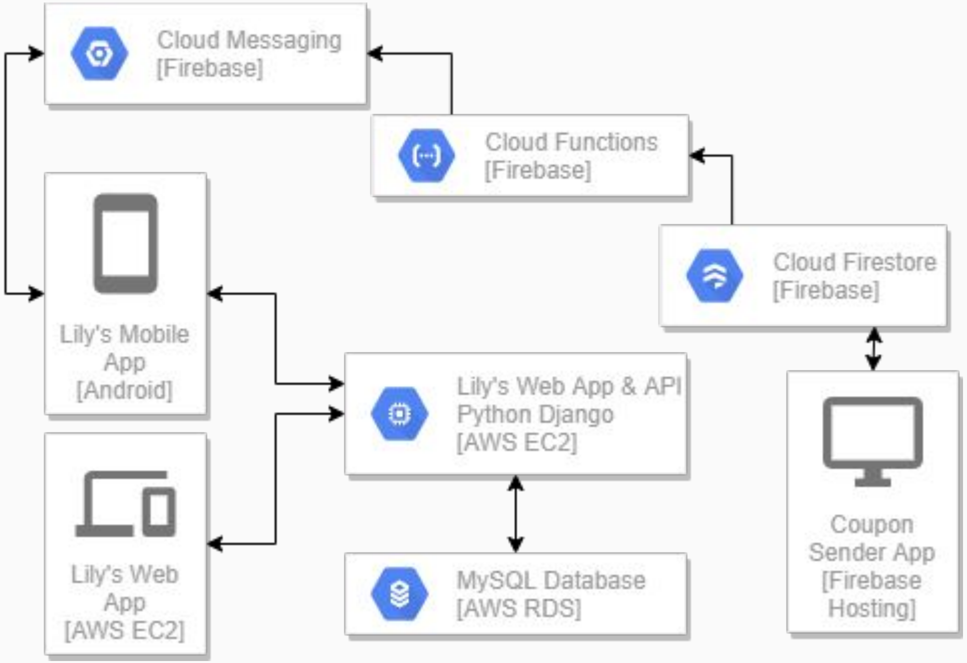


Animi voluptates dolorem id quae cumque vero. Officiis iure aliquid laborum eum at. Sunt nihil error nobis aliquam nobis ratione. Explicabo id iste beatae error rerum. Dicta sunt beatae quasi voluptas dolorum quibusdam iure quas.

Dicta labore laudantium laborum beatae voluptates qui maxime. Temporibus quibusdam eaque animi laboriosam delectus. Incidunt accusamus nisi tenetur molestias

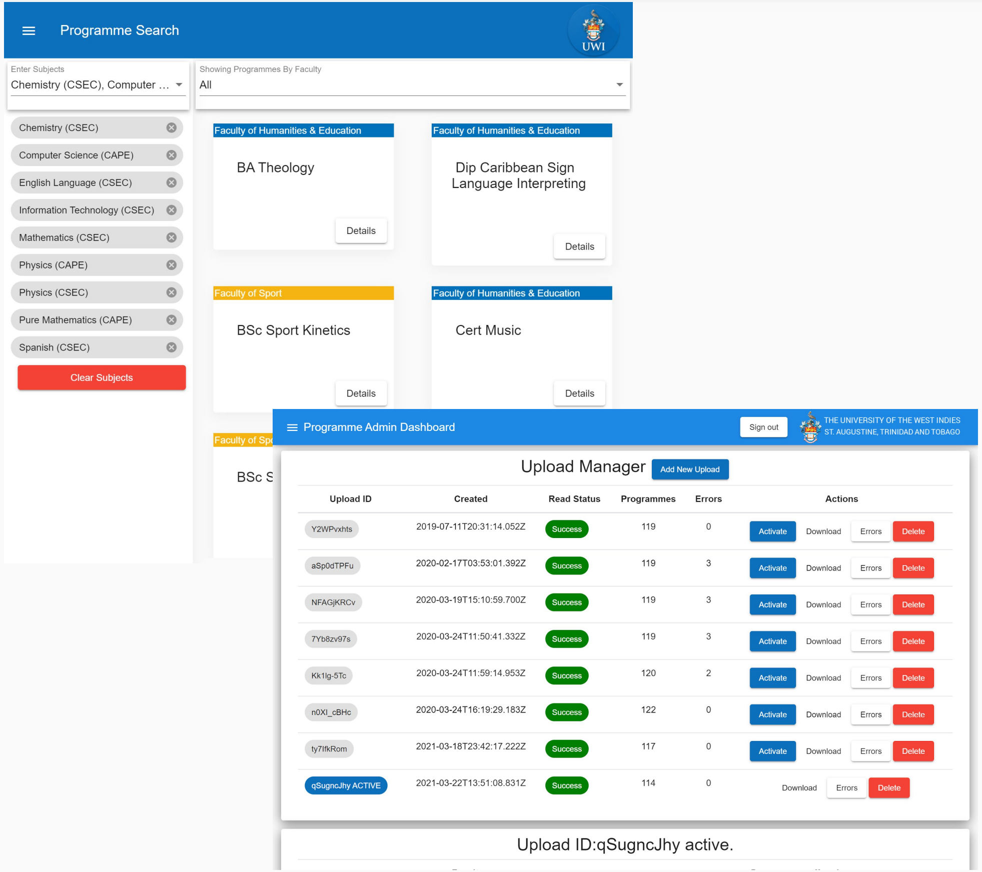
Lily's System Design

- Single VM for website and API
- Database hosted on separate machine
- 3rd Party service for storing coupons and sending notifications
- 3 Party service for hosting coupon delivery app
- Note:
    - Arrows indicate flow of communication, unidirectional/bidirectional
    - Platform/Provider indicated with [brackets]



Case Study: BeUWI

- Requirements:
  - Programme Search Client
    - Browse Programmes
    - Search programmes by requirements
  - Programme Admin Client
    - Upload Programmes via csv files
    - Manage and rollback uploads
- Non functional requirements:
  - High traffic on Search Client
  - Requirements Query Time Should be <500ms
  - Backend Should scale but Database is fixed read only



Enter Subjects: Chemistry (CSEC), Computer …

Showing Programmes By Faculty: All

Faculty of Humanities & Education
BA Theology

Faculty of Humanities & Education
Dip Caribbean Sign Language Interpreting

Faculty of Sport
BSc Sport Kinetics

Faculty of Humanities & Education
Cert Music

Faculty of Spo
BSc S

Programme Admin Dashboard
Sign out
THE UNIVERSITY OF THE WEST INDIES
ST. AUGUSTINE, TRINIDAD AND TOBAGO

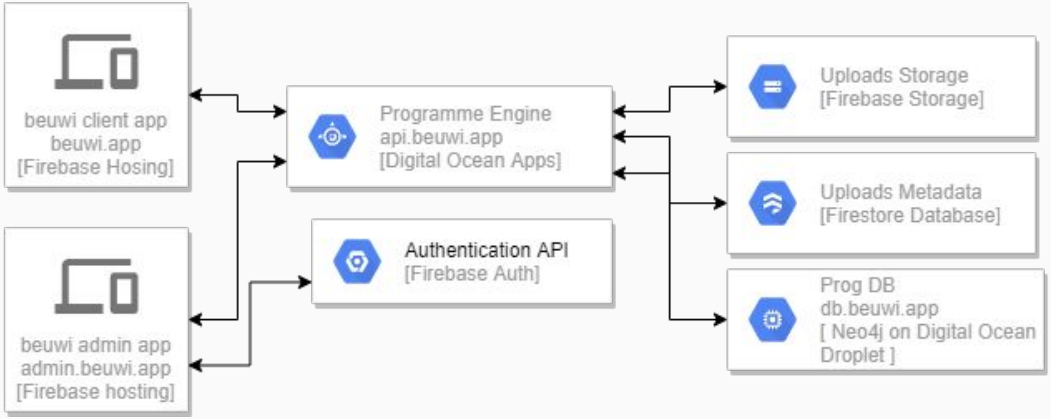
Upload Manager

Upload ID	Created	Read Status	Programmes	Errors	Actions
Y2WPvxhts	2019-07-11T20:31:14.052Z	Success	119	0	Activate Download Errors Delete
aSp0dTPFu	2020-02-17T03:53:01.392Z	Success	119	3	Activate Download Errors Delete
NFAGjKRCv	2020-03-19T15:10:59.700Z	Success	119	3	Activate Download Errors Delete
7Yb8zv97s	2020-03-24T11:50:41.332Z	Success	119	3	Activate Download Errors Delete
Kk1lg-5Tc	2020-03-24T11:59:14.953Z	Success	120	2	Activate Download Errors Delete
n0XI_cBHc	2020-03-24T16:19:29.183Z	Success	122	0	Activate Download Errors Delete
ty7IfkRom	2021-03-18T23:42:17.222Z	Success	117	0	Activate Download Errors Delete
qSugncJhy ACTIVE	2021-03-22T13:51:08.831Z	Success	114	0	Download Errors Delete

Upload ID:qSugncJhy active.

BeUWI System Design

- Graph Database used, deployed on dedicated Digital Ocean Droplet (Virtual Machine)
- Programme engine deploy on Digitalocean Apps platform for auto scaling and DDoS protection
- Stores uploads separately from backend in firebase storage
- Upload metadata stored in NoSQL database, firebase firestore
- Admin and Client apps on CDN via firebase hosting
- Used firebase auth on admin app


beuwi client app
beuwi.app
[Firebase Hosing]
beuwi admin app
admin.beuwi.app
[Firebase hosting]
Programme Engine
api.beuwi.app
[Digital Ocean Apps]
Authentication API
[Firebase Auth]
Uploads Storage
[Firebase Storage]
Uploads Metadata
[Firestore Database]
Prog DB
db.beuwi.app
[Neo4j on Digital Ocean Droplet]