Luddy School of Informatics, Computing and Engineering

APPLIED DISTRIBUTED SYSTEMS

AMPERSAND

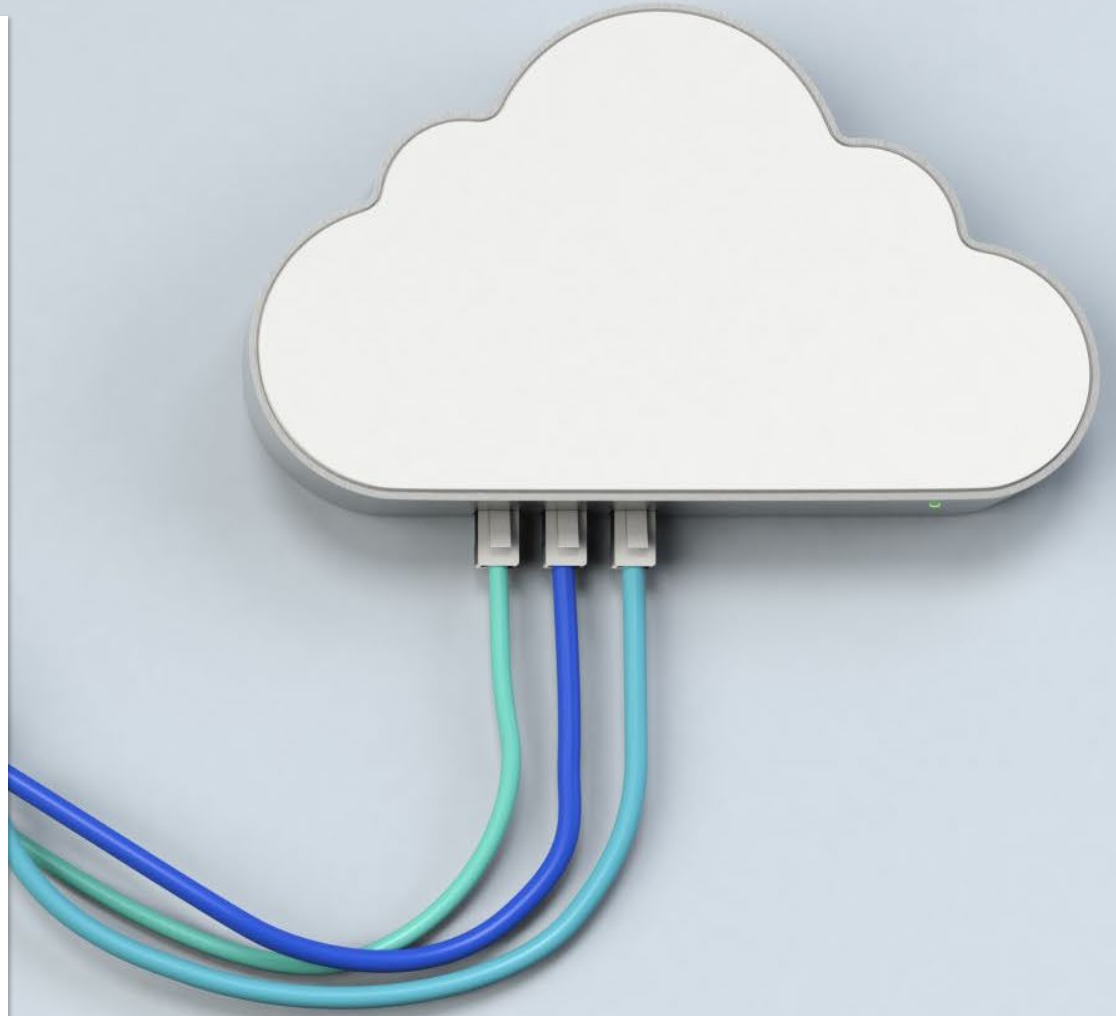
ADITI PEDNEKAR - DHRUTI PATEL – NIKHIL KAMBLE

SECTION 1

WHAT IS AMPERSAND?

AMPERSAND IS...

- ✓ a full-stack microservice-based cloud-native weather monitoring application.
- ✓ designed, developed and deployed as a distributed system in a significant span of 5 months.
- ✓ based on numerous technology stack and microservices.
- ✓ implementing Continuous Integration and Continuous Deployment of application
- ✓ using Virtualization, Containerization and Cloud Technologies to achieve reliable and efficient distributed system.



SECTION 2

MANAGING PROGRESS?

MANAGEMENT 1

GitHub Project Board

Created issues, pull requests for task allocation and tracking

Branching Strategy

1. Dev Branch
2. Feature Branch (Code / CI-CD / Custos)
3. Release Branch (for every project milestone)

MANAGEMENT 2

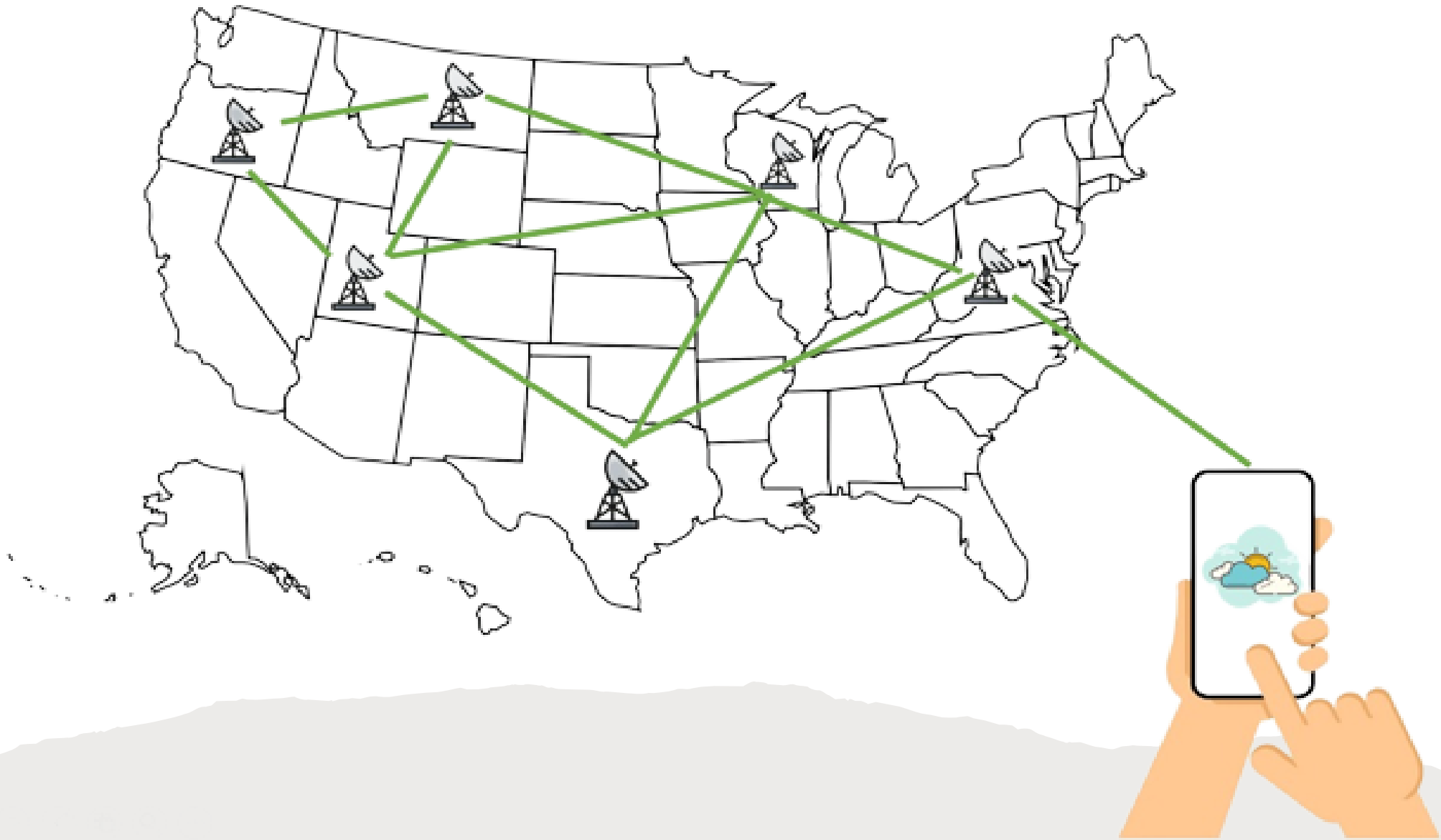
Project Documentation is maintained under Wiki

The team proactively used Labels and Release Tags

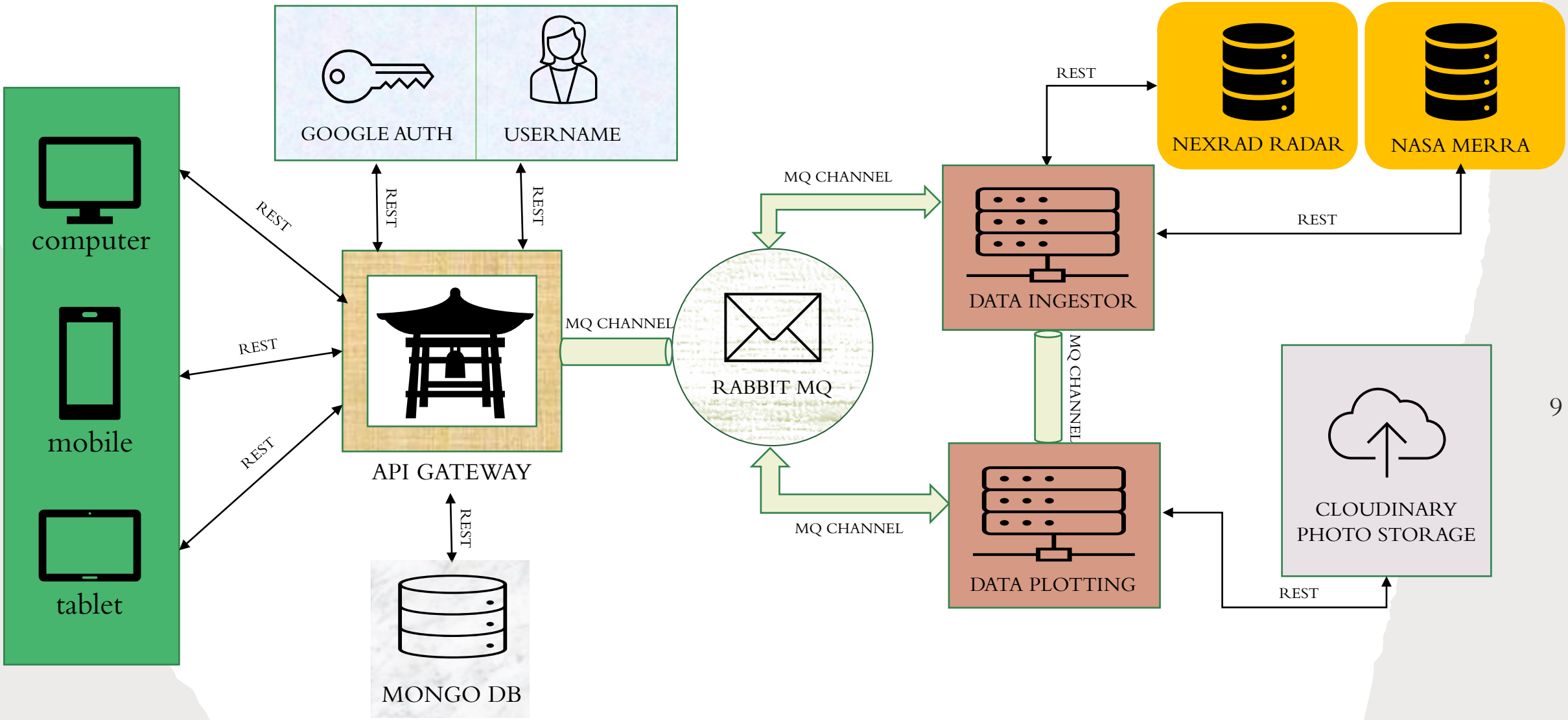
Bi-weekly meetings and merging all progress in one dev branch

SECTION 3

DESIGN AND ARCHITECTURE



WE WENT FROM THIS

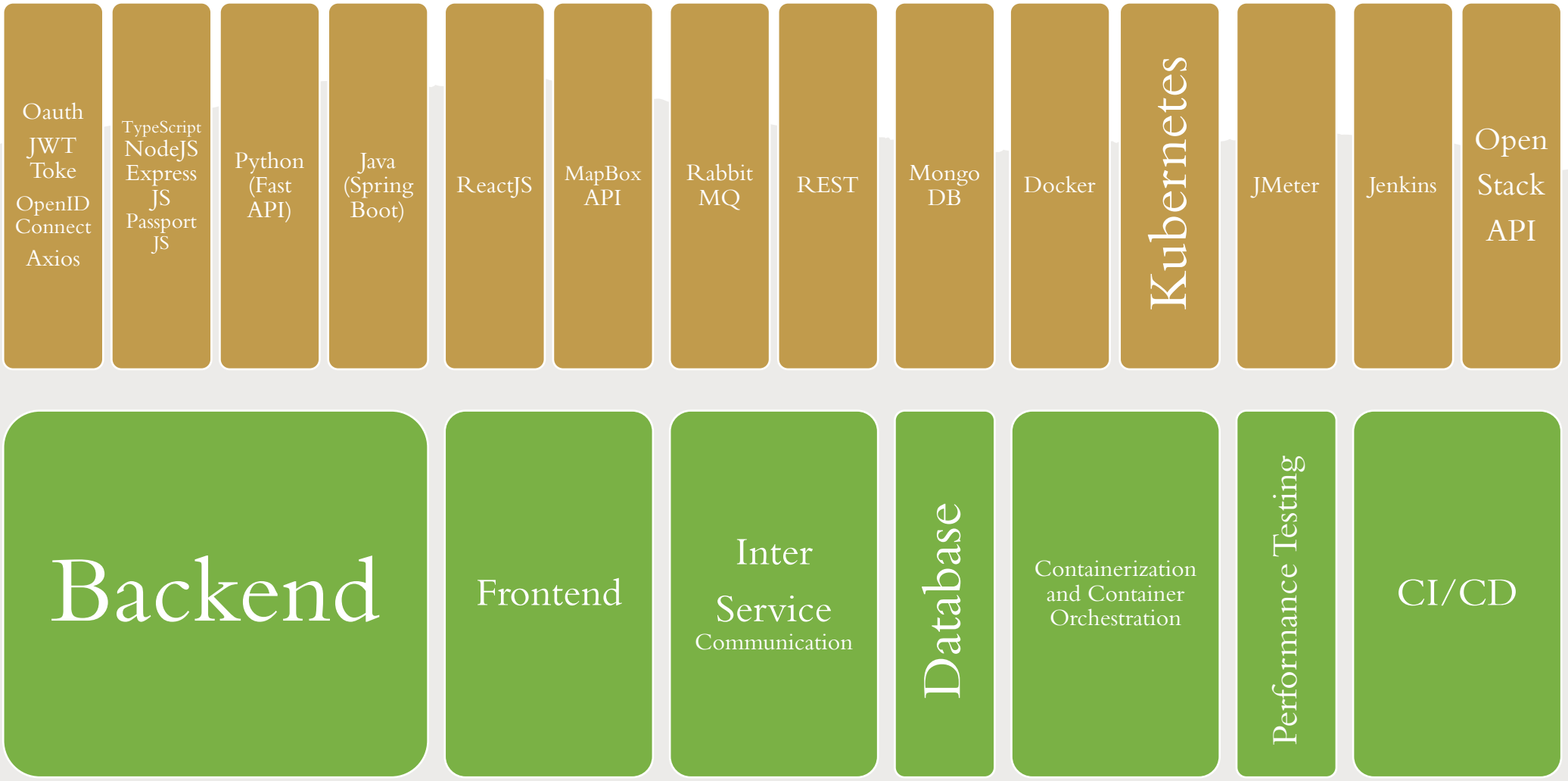

GOOGLE AUTH
USERNAME
computer
mobile
tablet
REST
REST
REST
REST
REST
REST
API GATEWAY
MONGO DB
MQ CHANNEL
RABBIT MQ
MQ CHANNEL
MQ CHANNEL
REST
NEXRAD RADAR
NASA MERRA
REST
DATA INGESTOR
MQ CHANNEL
DATA PLOTTING
CLOUDINARY
PHOTO STORAGE
REST

TO THIS

SECTION 4

PROGRAMMING LANGUAGES & MICROSERVICES

TECH STACK

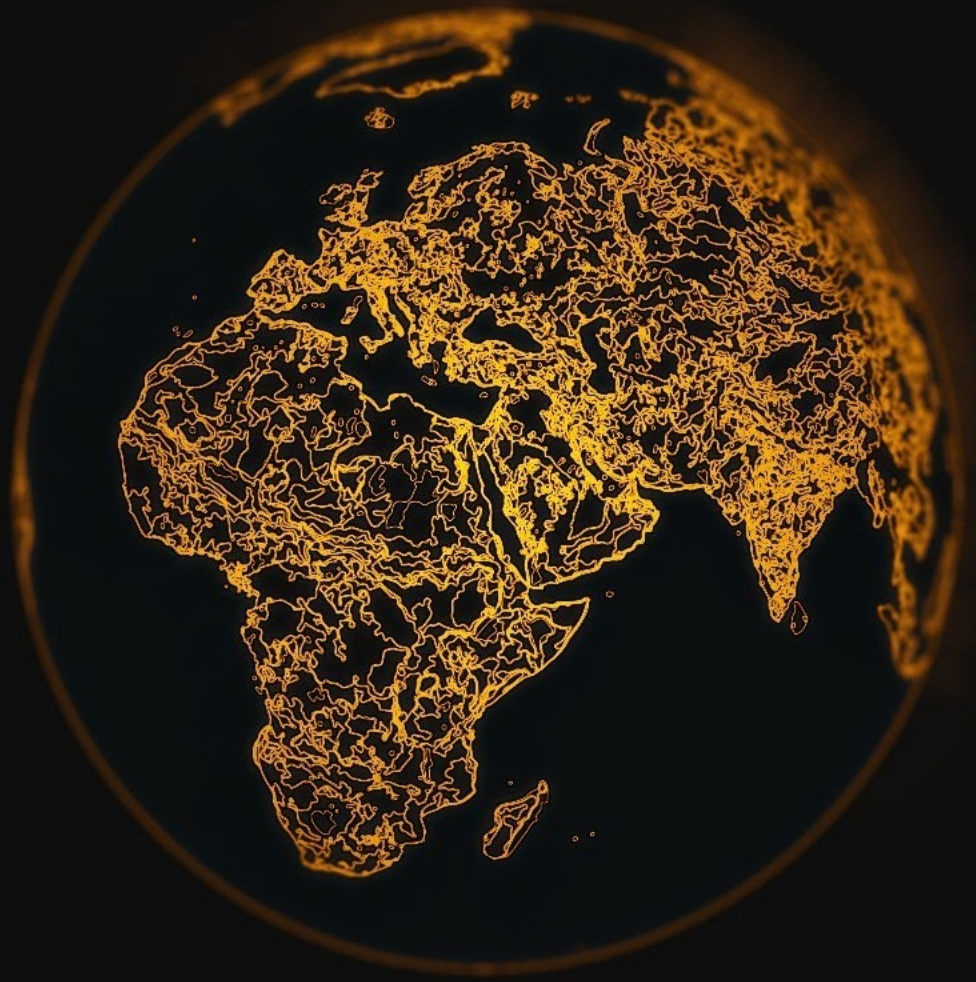

Oauth
JWT
Toke
OpenID
Connect
Axios
TypeScript
NodeJS
Express
JS
Passport
JS
Python
(Fast
API)
Java
(Spring
Boot)
ReactJS
MapBox
API
Rabbit
MQ
REST
Mongo
DB
Docker
Kubernetes
JMeter
Jenkins
Open
Stack
API
Backend
Frontend
Inter
Service
Communication
Database
Containerization
and Container
Orchestration
Performance Testing
CI/CD

SECTION 5

DATASETS

NEXRAD DATA

- The Next Generation Weather Radar (NEXRAD) system is a network of 160 high-resolution weather radar operated by the NWS, FAA and U.S. Air Force.
- If the user wants to access this data, they need to mention Radar Station, Date and Time.
- The Plot is generated based on reflectivity value stored by the Radar.



NASA MERRA DATA

- Modern-Era Retrospective analysis for Research and Application (MERRA-2) dataset is based on a version of the atmospheric data assimilation system.
- If the user wants to access this data, they need to mention the Longitude and Latitude of the location and desired Year.
- The Plot is generated based on reflectivity value stored by MERRA.

SECTION 6

INTER SERVICE COMMUNICATION

INTER SERVICE COMMUNICATION

Framework: RabbitMQ

Why Message Queues?

1. Light Weight
2. Efficient
3. Better Performance

RabbitMQ

INTER SERVICE COMMUNICATION

Framework: RabbitMQ

Challenges?

1. For every language, MQ implementation is different
2. Lack of testing tools

RabbitMQ

SECTION 7

GATEWAY

GATEWAY

Why REST API?

1. Entry point of architecture
2. Easy implementation
3. Universally accepted standard

SECTION 8

DATABASE

MONGO DB

- High Performance
- Scalable
- Change friendly Design

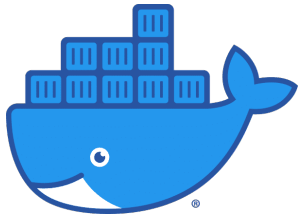
MongoDB®

SECTION 9

CONTAINERIZATION AND CONTAINER ORCHESTRATION

CONTAINERIZATION

- Reproducibility
- Isolation
- Platform Independent
- Environment Management



CONTAINER ORCHESTRATION

- Portability
- Reduced Cloud Complexity
- Reliability

kubernetes

SECTION 10

CONTINUOUS INTEGRATION & CONTINUOUS DEPLOYMENT

CONTINUOUS INTEGRATION

Why Jenkins?

Jenkins

1. Easily Configurable
2. Platform Independent
3. Most of the integration work is automated

CONTINUOUS DEPLOYMENT

Why Jetstream?

1. Easy management of VM's
2. Flexible Integration
3. Statistical Reports and Cloud Monitoring

IT'S SHOWTIME

SECTION 12

CHALLENGES

CHALLENGES

1. Significant investment required to decide on what technologies to use at different places in the software stack
2. Required learning programming languages and frameworks that we were not aware of
3. Focus on quality over quantity
4. Working remotely

SECTION 13

KEY TAKEAWAYS

KEY TAKEAWAYS

1. Perfect solution is a fairy tale. Everything is a trade-off to get better outcomes
2. Monolithic to Microservices
3. GitHub is a lifesaver
4. Know your limits
5. The world is filled with technologies

THANK YOU