# Feasibility Presentation Team ODB

Jacob Johnson
Emily Linderman
Tom Lobbestael
Michael Dykema

# Content

- Overview
- Toolset
- Repository Structure – GitHub
- System Organization
- Development
- Non-Trivial Requirements
- Work Policies
- Testing
- Deliverance and Milestones

# Overview

- Create a data lake to store ODB’s existing data and new data.
- Build a machine learning algorithm to generate content recommendations.
- Recommendations will be retrieved by an API built by the team
- Integration with ODB’s website as well as front end design will be taken care of by ODB’s web development team.
- Create a monitoring system for our API and data pipeline, as well as cost warnings for the various AWS services used.

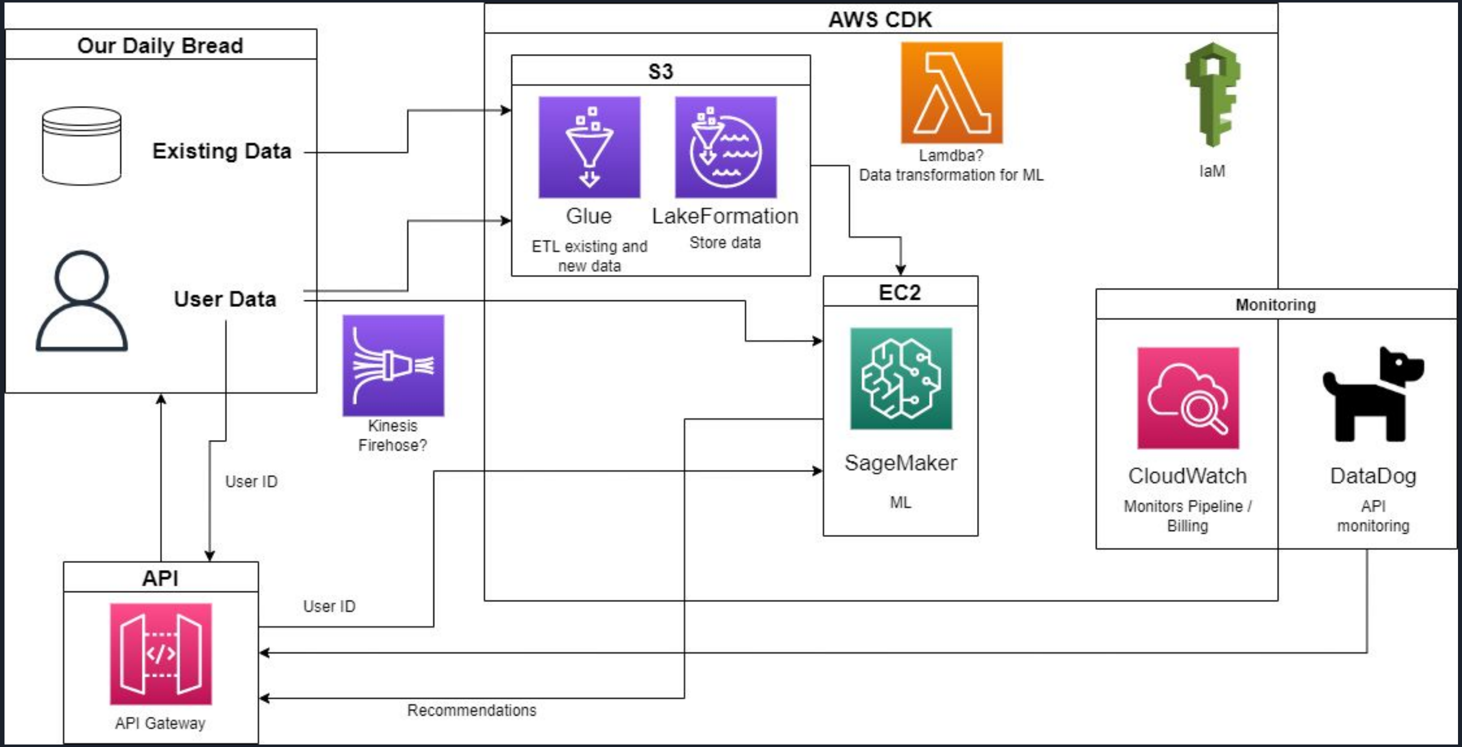
# Toolset

- **AWS LakeFormation** - Create data lake from pre-existing data stores as well as perform transformations on the data to prepare it for analysis.
- **AWS SageMaker**  - Provides common machine learning algorithms that are optimized to run efficiently against extremely large data sets. It also features “SageMaker Studio”, an integrated machine learning environment where you can build, train, deploy, and analyze your models all in the same application.
- **AWS CloudWatch** - Provides data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization. Using this service will allow an easy way to create a dashboard to monitor the health of the pipeline.

# Repository Structure – GitHub

- Our team will utilize Git Flow as our standard for branching and managing our repository.
- Two permanent branches, master and develop:
  - ***Master***: production ready code and will be the most stable version of our project.
  - ***Develop***: the most recent features and improvements that are awaiting release.
- Temporary branches:
  - ***Feature***: features that are actively being developed for future updates.
  - ***Release***: last minute fine tuning to the current develop branch before merging to master.
  - ***Hotfix***: fix bugs or other issues that are found within the master branch .

# System Organization



# Development

## Services

- AWS Cloud Development Kit
- AWS Command Line Interface
- AWS S3
- AWS SageMaker
- AWS Lake Formation
- AWS EC2
- AWS Lambda
- AWS CloudWatch
- DataDog
- AWS API Gateway
- AWS Kinesis Firehose
- AWS IAM
- AWS Glue

## IDEs

- VSCode

## Languages

- **Python**

# Non-Trivial Requirements

- Data Lake
  - AWS LakeFormation
  - AWS Glue
- Recommendation Engine
  - AWS Lambda
  - AWS SageMaker
- API
  - AWS API Gateway
- Monitoring
  - AWS CloudWatch
  - Datadog
- Security
  - AWS IAM

# Work Policies

- Due to linear structure all tasks will be collaborative with a lead developer
    - *Data ETL and Storage*: **Jacob**
    - *Recommendation engine*: **Tom**
    - *Security*: **Michael**
    - *API*: **Michael**
    - *Monitoring*: **Emily**

# Testing

- Infrastructure Testing
  - Infrastructure code will be tested throughout semester
  - Upon integration code will be reviewed and tested by employees of ODB
- Algorithm Comparison
  - Recommendation algorithms will be compared to determine best fit
  - A/B Testing

# Deliverance and Milestones

- Sprint 1
    - Create and populate data lake
    - Configure accounts and security access
    - Research and set up dashboard
- Sprint 2
    - Research
        - Recommendation Algorithms
    - Set up recommendation engine
    - Create API
- Sprint 3
    - Prep for website integration
    - Create alarms on dashboard
    - Set up monitoring software
- Sprint 4
    - Optimize recommendation engine
    - Debug
- Sprint 5
    - Continue optimization
    - Documentation
    - Refine code
- Sprint 6
    - Prepare final report for ODB
    - Submit project
    - Presentation

Thank you