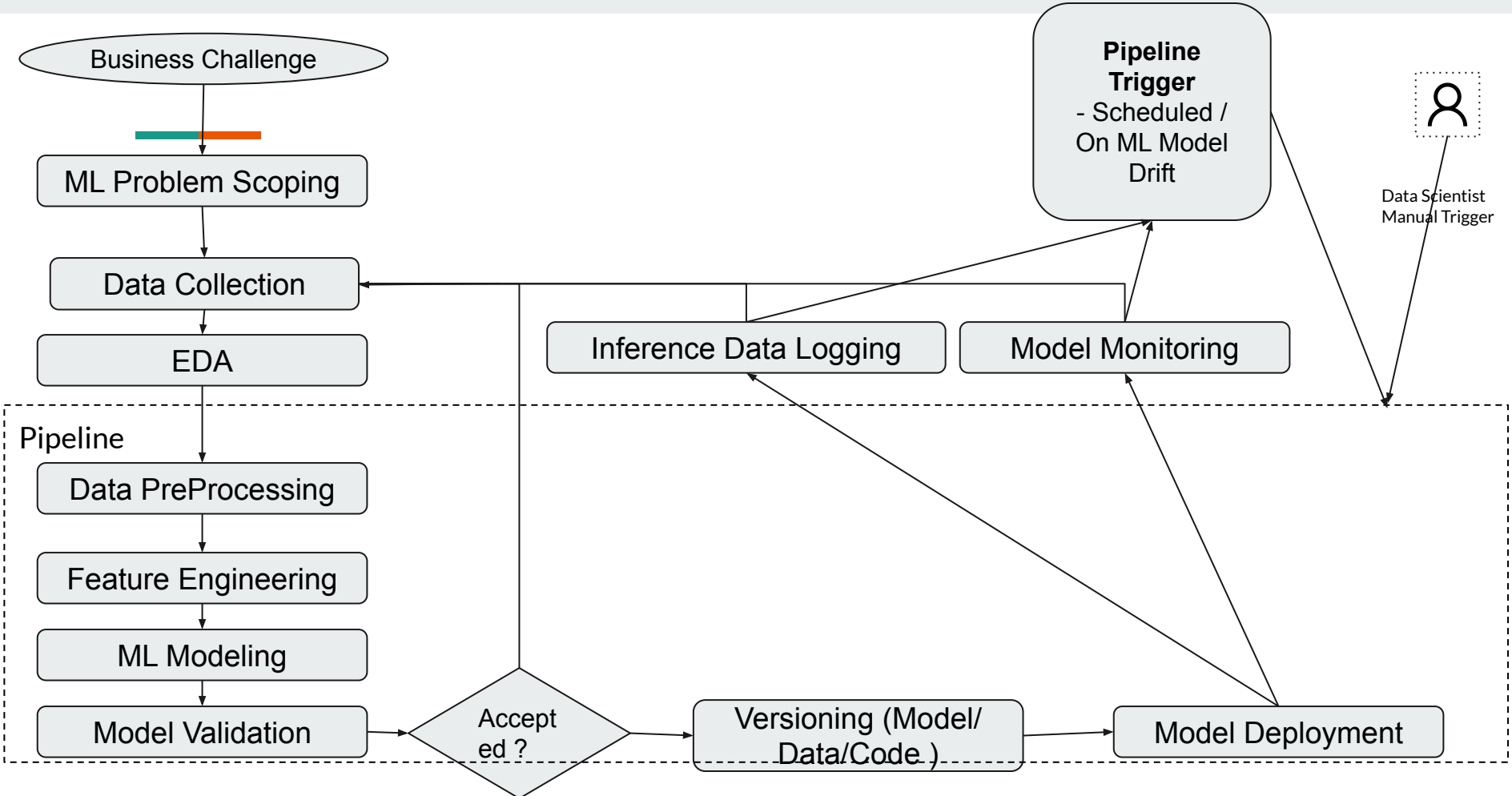
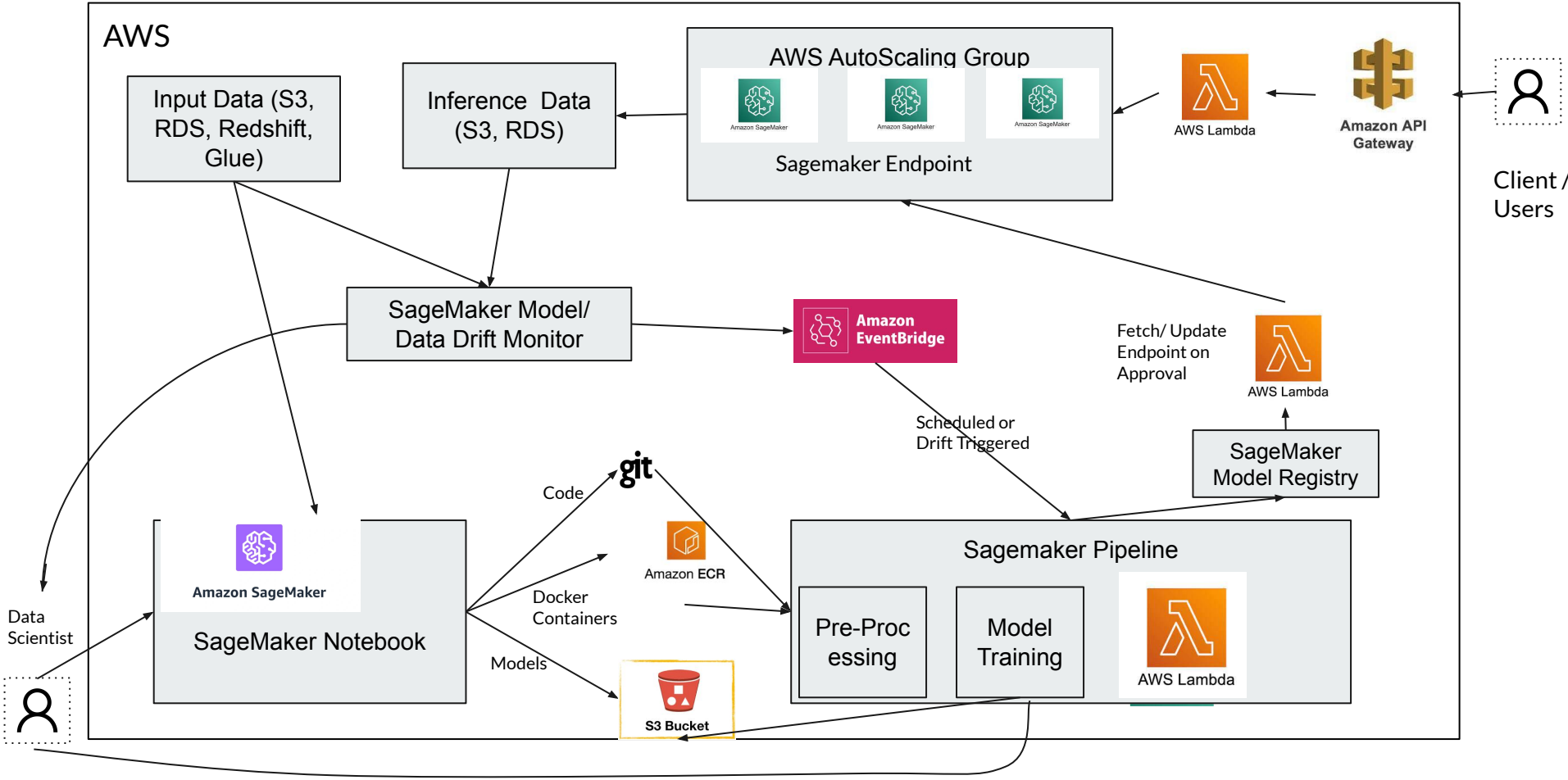
ML Process


Business Challenge
ML Problem Scoping
Data Collection
EDA
Pipeline
Data PreProcessing
Feature Engineering
ML Modeling
Model Validation
Accept ed ?
Versioning (Model/ Data/Code)
Model Deployment
Inference Data Logging
Model Monitoring
Pipeline Trigger
- Scheduled / On ML Model Drift
Data Scientist Manual Trigger

ML Architecture - AWS

(Excluding Staging / Dev / Prod Separate Deployment Separate Setup Arch)



ML Deployment Arch (Rough Outline)

Fully Self-Managed ML Environment

1. Model Serialisation : (Pickle)

2. Web Server : (Flask)
 - To provide web access to model.
 - Model Deserialisation/ Pipeline/ Inference code integration to Flask

3. Containerisation : (docker/ packer)
 - Package libraries, dependencies here
 - Self–Contained deployable images created
 - Fetch latest codes from git
 - Auto-refresh models from s3
 - Connect to DB(Amazon RDS- for data storage)

4. Deployment : (BitBucket Pipeline / TerraForm)
- Deploy to AWS EC2/ GCP instances/ any cloud
- Multiple prod instances (scalability)

5. Load Balancer :
 - AWS load balancer to balance load between prod servers

SageMaker

0. Model Trained & Deployed to Sagemaker
 - Direct deployment from notebook available

1. API Rest EndPoint
(Prediction Request (JSON) received here)

2. AWS lambda
 - 2 way communication with Sagemaker Inference API Endpoint
 - Pre-Inference :
  - Pre inference Custom Logic not in ML serialised object can be added here
  - Data can be logged to DB here
 - Post_inference :
  - All steps post receiving here
  - Prediction received post-process here, if needed

3. Amazon Sagemaker Inference EndPoint

PS: Due to time constraint, I have provided rough arch outline. I may be able to make detailed & arch diagram and upload it to git later.