Disease Progression Challenge

Design Example

Aaron Viviano, Software Architect and Engineer

Presentation Overview

- About Me
- Data Integration Strategy
- Data Model
- Data Pipelines
- Machine Learning Algorithms
- Dash Board System

About Me

- 16 years of professional Software Engineering experience
- 10 years of professional Software Architectural experience
- 7 years working on Medical Devices
- 3 years working on Machine Vision Systems
- I'm a team oriented self-motivated engineer who loves working with people and on challenging problems
- My software solutions have:
  - Produced and quality checked billions of products
  - Produced 2.2 million radioactive drug doses for patients
  - Operated in realtime, asynchronous high performance resource constrained environments
  - Successfully been submitted to the FDA on multiple occasions

Data Integration Strategy - Data Considerations

- 197K-234K new Lung Cancer cases in the US per year, taking 215K as average
- Lung Cancer scanned every 3, 6, and 12 months
- Theoretically 16 million COPD scans yearly
- Lung Cancer uses PET-CT[1]
- COPD uses CT[2]

Image Modality	Size	KB	Images in Study	Study Size (MB)
PET	128x128x1	16	100	1.5
MRI	256x256x1	128	200	25
CT	512x512x2	512	500	250

Disease	Cases	Scans Per Year	Total Study Data
Lung Cancer PET	215K	1x	322 GB
Lunch Cancer CT	215K	2x	107 TB
COPD CT	16M	1x	4 PB
Totals	16.5M	4x	4.107 PB

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3473490/ 2. https://www.copdfoundation.org/What-is-COPD/Understanding-COPD/How-is-COPD-Diagnosed.aspx

Data Integration Strategy - Further Data Considerations

- 4.1 PB of data seems off. Company has 500TB of lung data.
- “(CT) of the chest is not routinely recommended except for detection of bronchiectasis and COPD patients that meet the criteria for lung cancer risk assessment.”[1]
- These cases would be roughly 0.12% of the population
- 16M * 0.12% = 19,200 COPD CT Scans per year

Disease	Rates Per Year	Percent of Population
Bronchiectasis	200K	~0.06%
Lung Cancer	215K	~0.06%

Disease	Cases	Scans Per Year	Total Study Data
Lung Cancer PET	215K	1x	322 GB
Lunch Cancer CT	215K	2x	107 TB
COPD CT	19K	1x	4.8 TB
Totals	16.5M	4x	~122 TB

1. https://goldcopd.org/wp-content/uploads/2020/11/GOLD-REPORT-2021-v1.1-25Nov20_WMV.pdf

Data Integration Strategy - Data Sources

- EHR Files
- Claims Protocols
- HL7 v2 and v3 protocols
- DICOM: File and communication standard
- Other regulatory environment formats / APIs / protocols
- Proprietary formats / APIs / protocols
- Data per case is 1.5 to 250 MB in size
- AWS Lambda has a 6MB request limit

Data Model - Background

- Many Data Source formats
  - XML, Structured Binary Storage Formats, JSON, Image Data, etc.
- Goal: One format to hold this data
- Goal: Allow data to be accessible to Machine Learning Algorithms

Data Model - Widely Used Solutions

- **Consideration: Structured SQL**
  - **Advantages:** Strong Schema, Good Query Performance
  - **Disadvantages:** Inflexible, difficult for ML to access
- **Consideration: Unstructured JSON**
  - **Advantages:** Flexible, Easy to update
  - **Disadvantages:** No schema enforcement, slower query performance

Data Model - SQLite

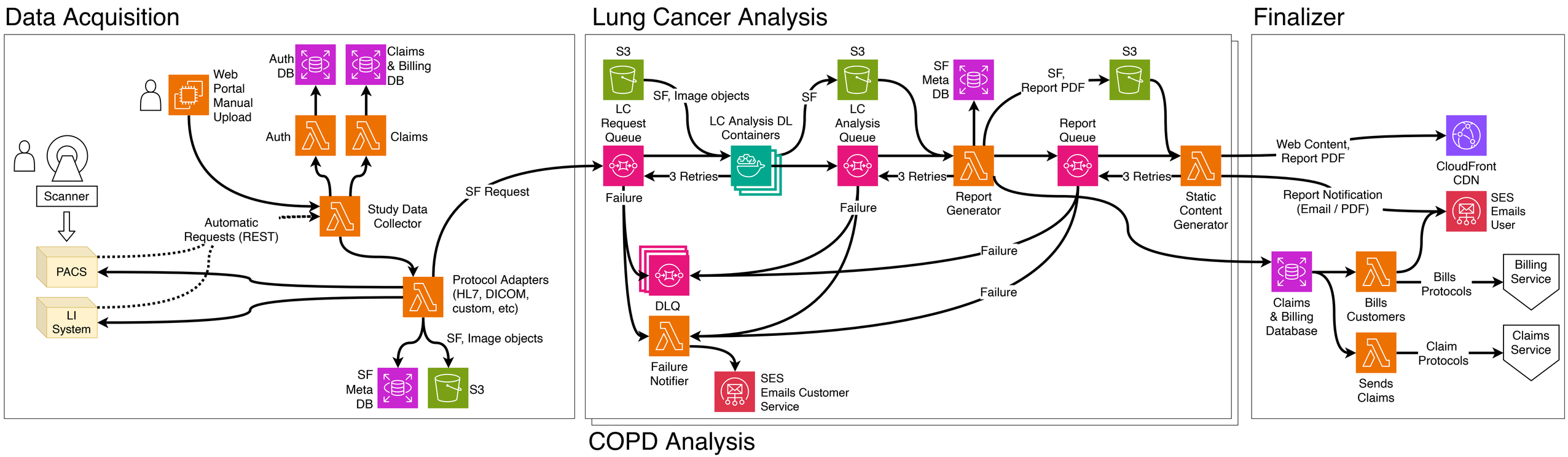
- SQLite is the most widely used database in the world.[1]
- SQLite is a file format, no db server required.
- SQLite can contain and store JSON, key-value pairs, and structured tables
- Each study is stored as a unique SQLite database file, called a Study File.
- The Study File contains three primary parts:
  - Company consistent data is stored in structured tables.
  - 3rd Party Unique data is stored as JSON.
  - The references to AWS S3 objects for study images.

1. https://www.sqlite.org/mostdeployed.html

Data Model - SQLite Study File

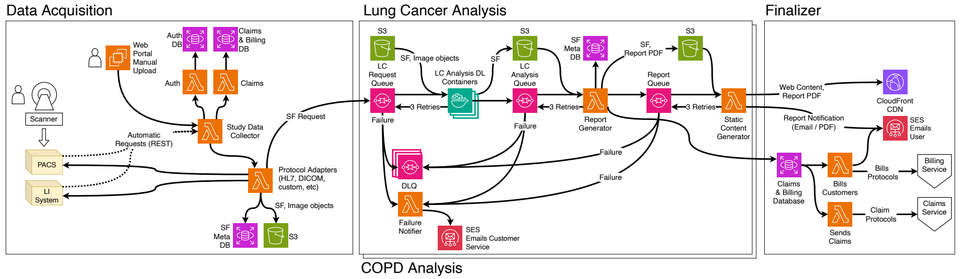
- **Advantages:**
  - Each study file can stay at its specific schema version
  - No need to update large numbers rows in a database on schema change
  - Asynchronously writing / reading across many files
  - Customer data specific recovery
  - Easily regional privacy guarantees
- **Disadvantages:**
  - Requires the system to be able to read old schemas, until schema is retired
  - Any schema updates requires updating existing files, which may be being processed
  - Multiple writers per file not allowed
  - Difficult to query across all data in all files
  - Requires another DB to track study file locations and status in S3
- **Mitigations**
  - Once a study is complete, consider copying data into a standard RDBMS for querying
  - Use JSON in the study file for company data until its known it will be used for a significant period of time

Lung Cancer and COPD Pipeline

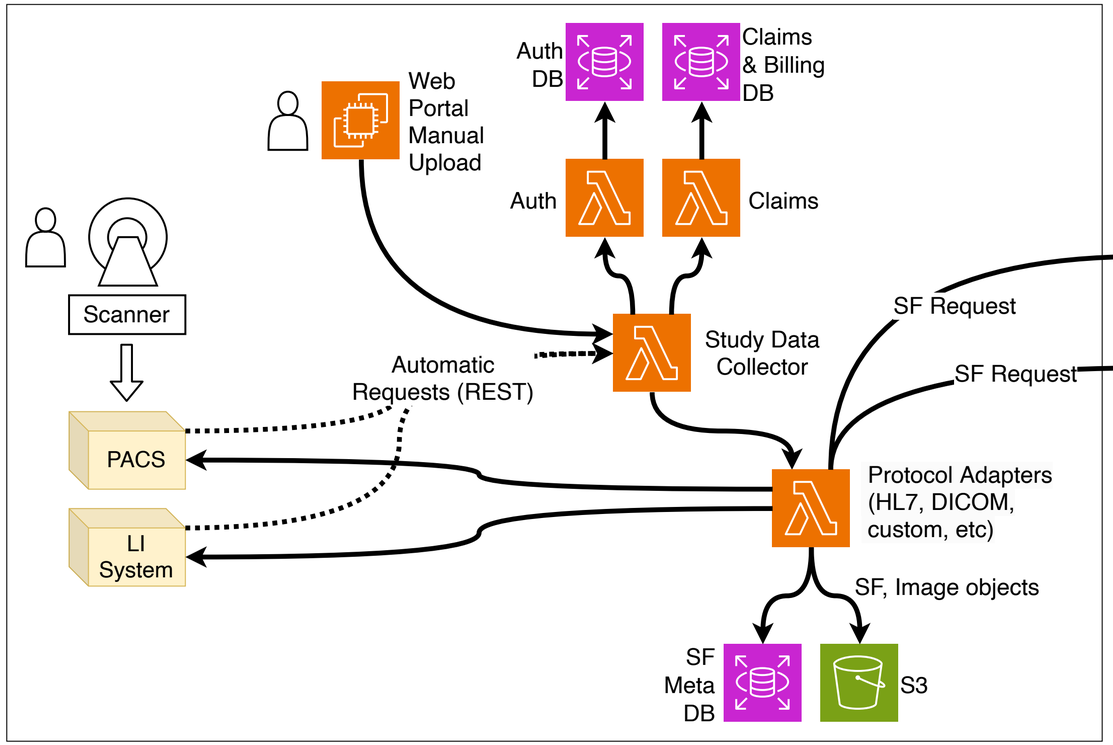


Pipeline - Data Acquisition

- Designed for automatic uploading
- Manual uploading possible
- Protocol Adapters can connect to various systems via HL7, DICOM, Proprietary, etc. to download data
- Adapters convert downloaded data to SQLite Study Files and Image S3 Objects
- Requests placed in relevant queue for analysis

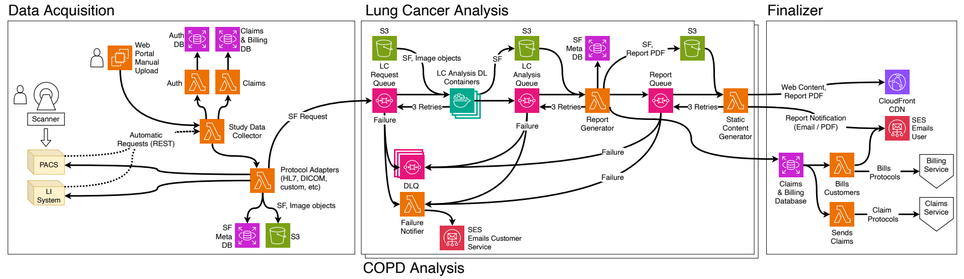


Data Acquisition

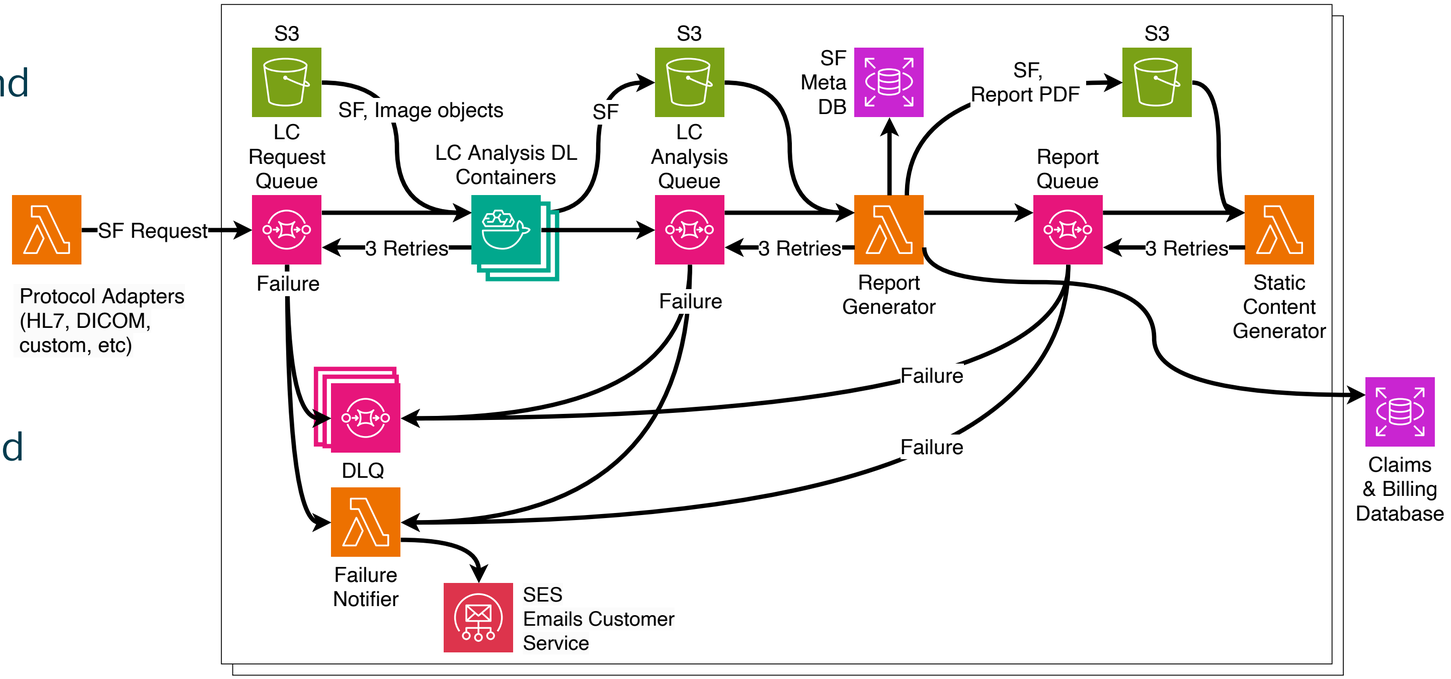


Pipeline - Analysis

- Requests are sent to AWS Deep Learning Containers to use GPUs
- Containers load Study Files and S3 Image objects to perform analysis
- Note: Containers can be replaced with serverless functions if ML Algorithms don't require GPUs
- Once analysis is complete, report data is generated and appended to the Study File and a PDF report created
- Static Content Generator pre-generates user facing content to minimize web server load
- Report data is also stored in the Claims and Billing Database to ensure customer is charged
- If any queue's work fails three times, the work is placed in a dead letter queue for inspection by customer service and, and an email is generated to notify customer service
- A similar set of analysis services also exist for COPD
- Data and AI models can be stored and processed on a per region basis to conform with specific regulatory environments



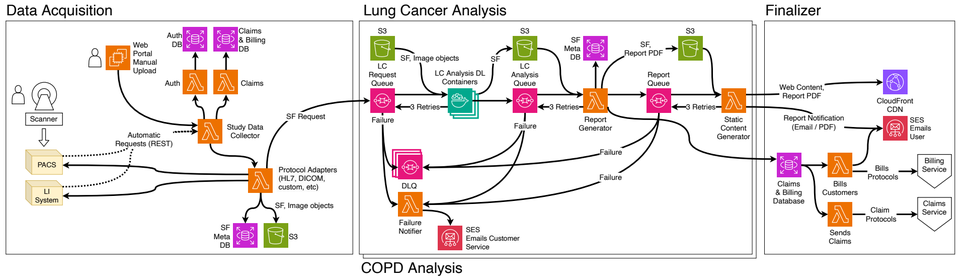
Lung Cancer Analysis



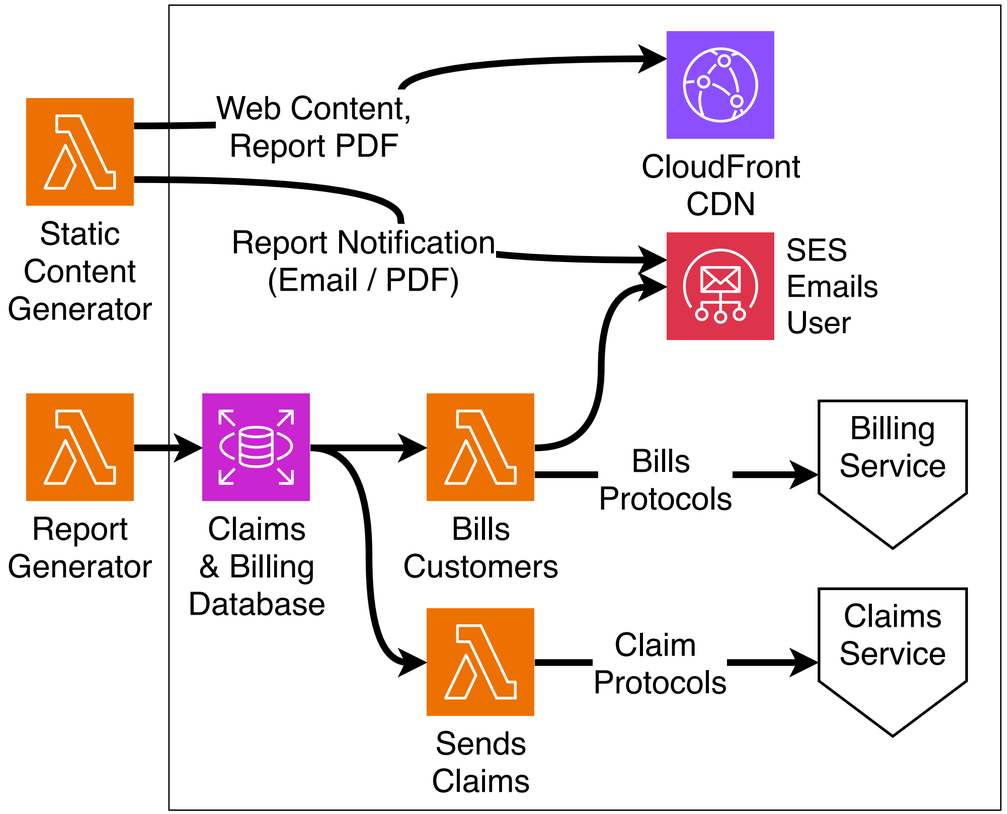
COPD Analysis

Pipeline - Finalizer

- Web Content and report PDFs are securely cached in CloudFront
- An email is sent to the user noting their analysis is complete (this can be done in batches to reduce emails sent per user)
- When the Claims and Billing database is updated it generates a billing email and calls the needed APIs to talk with the billing and claims services



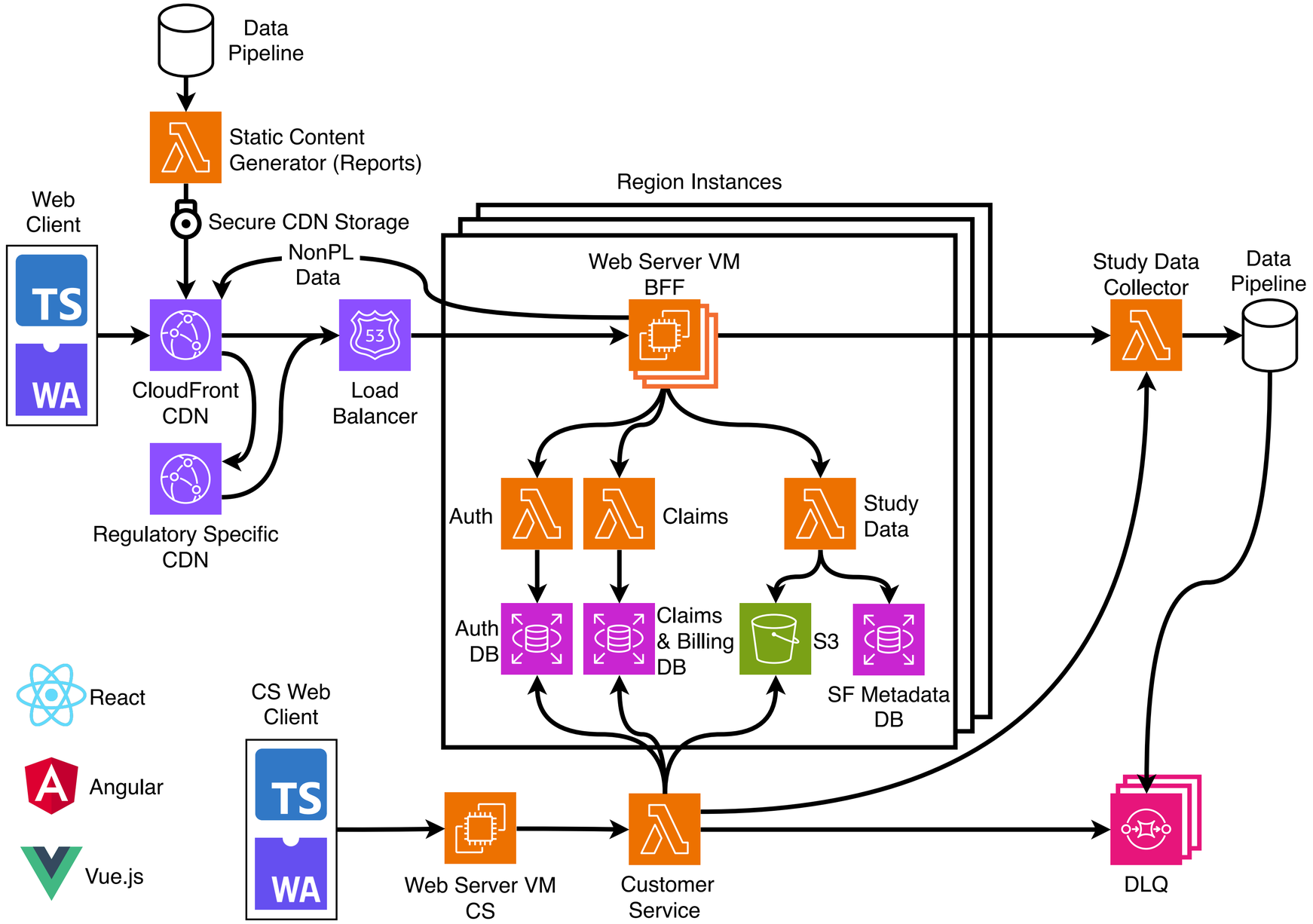
Finalizer



Machine Learning Algorithms

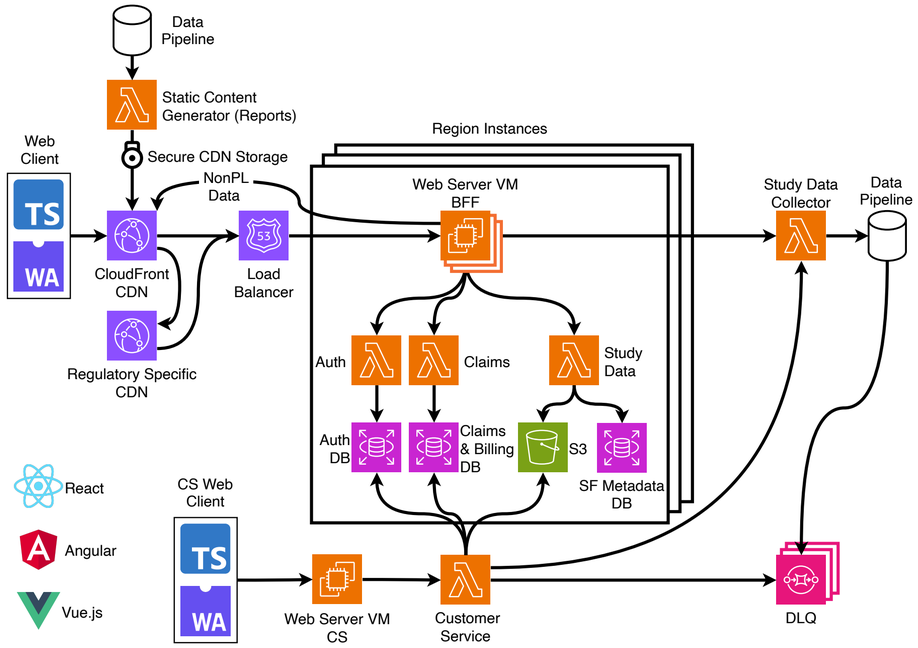
- I’m not a machine learning expert, I am certified for machine vision implementations
- I’d start with ensuring the availability of basic machine vision algorithms for the developers:
  - Filters, Edge Detection, Thresholds, Histograms, Dilation, Erosion, Image Measurements, etc.
  - Likely via OpenCV or another machine vision library
- I’d take an ML bootcamp to help get my knowledge off the ground
- I’d read various papers and books on machine learning in medical images
- Then I’d begin the process to hire an expert in the field to choose and or implement the appropriate algorithms. Using my acquired knowledge as a filter to find real experts

Dashboard - Web Architecture



Dashboard - Technical Information

- The front end is a thin client build with TypeScript for to ensure coding errors are caught at compile time
- WebAssembly is used to augment TypeScript for any performance based work (such as for 3D rendering study data via WebGL or WebGPU)
- Front end libraries (React, Angular, Vue.js) could be supported as needed by the project's requirements and developer preferences. Likely the Web App is MPA, but it may have some complex pages requiring SPA like approaches
- The web server acts as a thick server calling various APIs on behalf of the clients. This ensure the client's are simplified and helps to secure the serverless functions
- A secure CDN is used to contain pre-generated pages with results from the Data Pipeline
- A separate Customer Service website is used to allow for backend access when needed. This ensures that user and customer service functionality are never mixed



Thank you for your time, attention, and the opportunity.

Questions and comments?