Decentralized Radiology Application (RadApp)

Outline for Chainlink Hackathon 2021 Project
V0.2

clarksabenbus@gmail.com

October 2021

Abstract

This document contains the prospective outlook and framework for the RadApp based on the current team's vision

Contents

1 Welcome!

Welcome to the Radiology Decentralized Application Project

We are looking for Engineers, Programmers, and/or those with Medical domain knowledge to join our team for this Fall 2021 Chainlink Hackathon!

If you are interested in joining in and contributing after reading the White Paper Please reach out to me via discord(Clarke#0836) or email(clarksabenbus@gmail.com)

Thank you for your time.

1.1 RadApp Goals and Purpose

The Problem

The question in mind when discussing radiology in the current climate of machine learning is, "why haven't we solved this yet?". The answer: Data. Lack thereof, to be explicit.

How We Plan to Help Resolve It

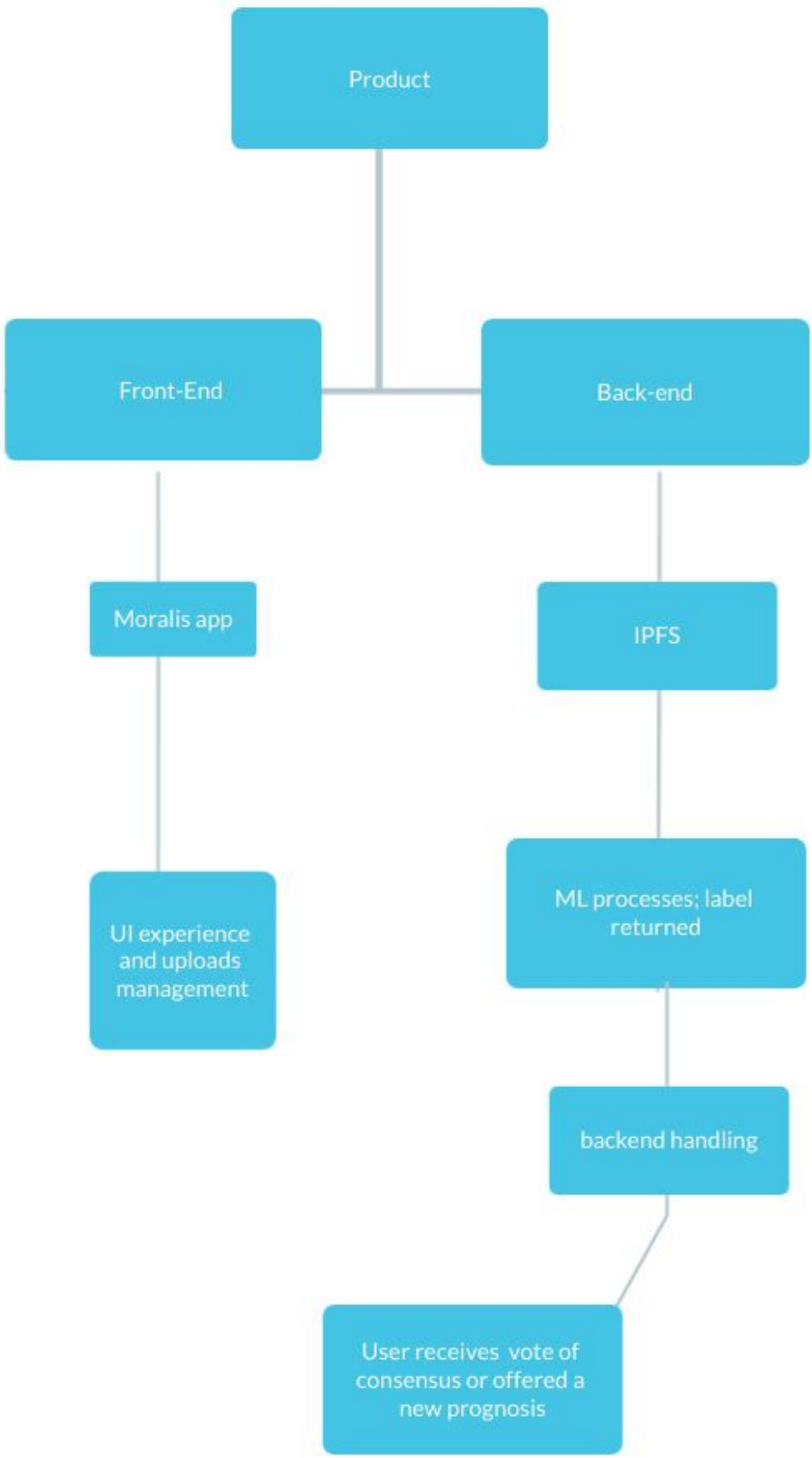
The multitude of imaging data that has been collected over the past century is often times either not labeled or contains personal information in the data file itself (this is standard practice in dicom files). The processing of labeling old data for increasing the prognosis quality has been mostly attended to but could be expedited by proactively adding new labeled data to the mix.

As such, the problem we set out to solve is to transform what would otherwise remain confidential data collecting dust on a hospital's database to a patient-protected dicom file(patient identification relevant data absent) to enable usage in model training . By providing a direct way to process dicom data for both corporate users (hospitals) and professional users (radiologists), the RadApp Project's focus is to allow for users to leverage the power of their patient's data without compromising the confidentiality of the patient data itself.

More Functionality

Furthermore, with the option to return tentative predictions to user from their uploaded data based on in-house machine learning models, we also aim to aid the work flow of radiologist via confirmation of scan interpretation.

1.2 RadApp High-Level Pipeline



1.3 Chainlink Hackathon Timeline

basis for framework timeline plans

- November 12: Deadline for Internal Team Recruitment
- November 14: Midway Check-in
  - team composition declaration and project idea submission
- November 28: Submissions Due
  - 5 minute video submission due to Chainlink organizers

2 Overview of the RadApp Processes

- Front end
  - serve users
  - data upload handling
- Back-end
  - main application logic
  - smart contract development and deployment
- Machine Learning
  - production of models
  - data preprocessing

2.1 Front-End

- Takes user uploaded data and labels
- Handles movement of data to IPFS; truncation of user personal metadata
- Returns prediction, suggested prognosis

2.2 Back-End

- Handles CID and IPFS directory
- Alchemy engineering
- smart contract calls

2.2.1 Machine Learning Development

- receives the uploaded dicom file from hashed location
- feature and model selector script based on user input
- preprocess the dicom to create relevant features for model inference
- prediction on preprocessed data

2.3 Goal Prize Technology Usage

subject to change based on expertise and interest of team

- Polygon
- Alchemy
- IFPS/Filecoin

3 Engineering Team Needs

(Recruitment Goals-> Please reach out if interested!!)

3.0 Brief Clarification

Developers can work in more than one of these subsections

3.1 Front-end Developer

Experienced with Moralis and Javascript (or interested in learning)

Experienced with web3

3.2 Back-end Developer

Experienced with IPFS and Alchemy or building out API's (or interested in learning)

Strong python programming background

3.3 Machine Learning Developer

Familiar with TensorFlow and or PyTorch

Strong background with algorithms (python)

3.4 Medical Field Specialist (Domain Expert)

Familiar with the field of Radiology

Familiar with HIPAA; willing to act as an pipeline auditor to insure protected PHI

3.4 The Current Two Team Member's Specialization

Yonathan

Systems analyst with a basic skill set in solidity. RadApp co-architect and developer

GitHub: https://github.com/YonathanTE

Clark

Python programmer with a specialization in machine learning development. Basic Solidity and JavaScript knowledge. RadApp co-architect and developer

Kaggle: https://www.kaggle.com/clarksaben

GitHub: https://github.com/csaben

4 Code Organization

4.1 GitHub Project Navigation Breakdown

(update post-recruiting)

invitation to GitHub Project will be sent if one becomes a team member; private for now

4.2 Brief Language & Technology Usage note

Use of programming language of choice, development environment of choice, and tools of choice totally dependent on developer so long as compatibility issues don't arise

4.0 Front-end To-dos w/ initial timeline checkpoints

Initial application with upload functionality TBD

4.0 Back-end To-dos w/ initial timeline checkpoints

Building out the internal IFPS file system [ongoing]

4.0 Machine Learning To-dos w/ initial timeline checkpoints

Preprocessing script development and collection [ongoing]

Model development and training [ongoing]

5 Conclusion

5.1 Thanks for reading, please reach out if you are interested!

Contacts:

- discord: Clarke#0836
- email: clarksabenbus@gmail.com

5.2 Recruiting Timeline

Official Team Declaration by Organizers: November 14th

Internal RadApp Project Recruiting Timeline: **November 12th**

*a full team assembled will also effectively end recruitment