Software Architecture Integrating Blockchain and Artificial Intelligence for Medical Data Aggregation

Jingpeng Tang¹, Qianwen Bi², Bradley Van Fleet¹, Jason Nelson¹, Carter Davis¹, and Joe Jacobson¹
¹Department of Computer Science, Utah Valley University, Orem, UT, USA
²School of Business, Utah Valley University, Orem, UT, USA

Abstract - Big data analytics are making revolutionary changes in the medical field, such as providing personalized medicine and prescriptive analytics, clinical risk intervention and predictive analytics, automated external and internal reporting of patient data, standardized medical terms, and patient registries. The key question is how to use this data intelligently and securely. Artificial Intelligence (AI) and blockchain are two cutting-edge technologies to support this data revolution. Our primary concern is how these technologies will impact medical big data analysis and corresponding software architecture design in this proposed research. Motivated by Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), we propose AI as a Service (AaaS) and Blockchain as a Service (BaaS) in this paper.

Keywords: Artificial Intelligence, Blockchain, Data Aggregation, Medical Data, Software Architecture

1 Introduction

Artificial Intelligence (AI) and blockchain are two cutting-edge technologies in the 21st century. The combination of the two will bring dramatic change and impact on traditional software architecture design. We are concerned primarily with how these technologies will impact medical data. As healthcare providers continue to expand their technology footprint, an increasing concern is storage and access to Protected Health Information (PHI). Health Insurance Portability and Accountability Act (HIPAA) regulations mandate how PHI is stored, transmitted, and accessed. According to the U.S. Department of Health & Human Services, HIPAA “[establishes] important protections for individually identifiable health information… including limitations on uses and disclosures of such information, safeguards against inappropriate uses and disclosures, and individuals’ rights with respect to their health information” [1].

However, these regulations are not all-encompassing. As a result, some providers have made it difficult for patients to access their PHI by introducing costs that can add up to high totals to print or copy data [2].

In this work, the integration of blockchain and artificial intelligence in a medical patient data aggregation platform for optimizing healthcare communication and delivery is explored. By leveraging the immutable, append-only architecture associated with distributed ledgers like Bitcoin and Ethereum, we hope to reduce data transmission overhead as patients interact with multiple facilities. Also integrating artificial intelligence, specifically machine learning, we seek for a way to provide diagnosing assistance to providers.

The Medical Assistant platform integrates these technologies into an Angular web application that a patient can interact with to access their data. This platform utilizes Representational State Transfer (REST) APIs to interact with the Ethereum blockchain and Inter-Planetary File System (IPFS) for file storage.

To facilitate the goal of diagnosing assistance, a neural network is implemented and integrated to analyze MRI brain scan images to determine the presence of tumors. Once complete, a binary value is assigned indicating the presence of a tumor.

2 Existing Research and Platforms

Warren Sarle’s address on AI shares one of the most commonly used artificial neural networks called multilayer perceptrons. The definition of a simple perceptron and its use of activation functions are explained. Lastly, the benefits of training models with multiple hidden layers, or deep models, are outlined [3]. The simple linear and multilayer perceptrons models discussed in Sarle’s work were used during this research.

Bengio and Glorot’s paper on understanding the difficulty of training AI provided additional information about deep neural networks [4]. It outlines the reasons why deep multilayer neural networks were unsuccessful in the past and ways to improve them. Experiments with different activation functions and the results showing how some are not suitable for deep layer models are explained. The paper also discusses the effect of the cost function on the network.
3 Software Architecture Design

As AI and blockchain are the two main concerns, a software prototype with the following architecture was designed and built:

3.1 SETH

It was identified early on that security is critical in handling medical data and should be designed as a service. Both blockchain and IPFS use decentralization, cryptographic hash, and Merkle tree to achieve data security [5]. In this research, we propose BaaS as a central security and protection design for medical data. Early iterations of the Medical Assistant platform implemented the Hyperledger Fabric blockchain. The Hyperledger Fabric architecture allows for hosting private, permissioned nodes that could be interacted with via the Hyperledger Composer REST API [6].

After further difficulty with configuration and integration, the Ethereum blockchain platform was used for the duration of the research. By using MetaMask, a blockchain approval service, multi-factor approval could be provided for transactions by utilizing their “secure identity vault” [7]. This allows for greater security and data reliability. Integration was achieved through a JavaScript distributed application (dApp).

A major concern that was discussed was the security of patient information. Ethereum is a public blockchain, meaning that any data stored, even if encrypted, is available for public download. To alleviate this concern, the Inter-Planetary File System was used to store files containing the data. Utilizing IPFS, hidden nodes could be created to store the data, while the blockchain stored file path hashes.

The added benefit of integrating IPFS was reduced data storage in Ethereum, which may also reduce the cost per transaction significantly.

Figure 1 is the resultant platform containing both the Ethereum and IPFS integrations packaged into a NodeJS solution – named SETH – and placed on a server. For Medical Assistant to interact with SETH, a REST API has been implemented for front-end development.

3.2 Artificial Intelligence

A second major part of the Medical Assistant platform is diagnosing assistance. Initial investigations were performed using Google’s TensorFlow AI. Choosing this platform facilitated the creation of a basic demo to understand AI development, and practice setting up neural networks.

![Figure 2. CNN model with the highest percentage of accuracy](image)

Experiments were done using different numbers of layers and perceptrons in order to create the most accurate model. In Figure 2 neurons are represented by nodes. Each column of nodes is called a layer. The first layer is the input layer, with 12288 nodes to match the 12288 pixels in the test images. The last layer is the output layer, with only 1 neuron representing the prediction value. The layers in between represent the hidden layers. After several tests with a variable number of hidden layers, this model gave the highest percentage of accuracy.

Once the initial design and sample demo were established, work was put forward to create a training set. The initial use case was to perform image classification using an AI with a Convolutional Neural Network to identify brain scans. This later developed into identifying potential masses in an MRI scan.

![Figure 3](image)

Sample images were taken from publicly available MRI scan images. To increase the sample size, images were duplicated and reflected about the vertical axis. Figures 3 to 5 show sample images used as part of the training and test design. Pseudo-tumors were drawn onto arbitrarily selected images.
3.3 Medical Assistant Web Application

In order to facilitate the two major parts of this application (AI and blockchain), a web application was developed to provide a user-friendly interface. This user interface (UI) was designed to be patient facing, allowing end-user access to the blockchain data.

An initial concern that was brought up was patient data access and security. As a result, two key mechanisms were put in place:

3.3.1 Multi-Factor Transaction Approval

For a user to create a transaction, such as sending medical history to a new provider, they must: 1. Select the information to send; 2. Select the provider to grant access; 3. Start the transaction, and 4. Approve the transaction in MetaMask. The addition of approving the transaction in MetaMask is to alert users to unauthorized activity and prevent potentially fraudulent transactions. The private key used to decrypt files is separated from the user credentials used to access the application allowed device management.

3.3.2 Front-End Design

Once a set of features were designed, and initial wireframes were approved, architectural design and development were approached.

Figure 6, Medical Assistant patient overview wireframe

The application design approach was to reduce the workflow for users to obtain information from all medical providers. Creating an overview (see Figure 6) that could be synchronized potentially facilitates the aggregation of medical information, regardless of the provider or facility.

A focus of the overview was to create a page that was concise and displayed important information up front.

One key area of focus was patient interaction with medical providers by updating personal information. The possibility of a workflow that would allow a patient to update their information (e.g. residential address) was discussed. These updates would then be immediately available for providers and facilities to access.

Providing an interface that was simple to use was a key element of this feature.

Part of the design took into consideration the structure of the data being stored in SETH, and the REST methods used for communication. The final structure of data was a JSON object with the following definition:

```json
{
    "PatientId": "",
    "Attention": "",
    "Line1": "",
    "City": "",
    "State": "",
    "PostalCode": "",
    "Country": ""
}
```

The last feature identified for this research was creating an interface for a user to manage all their providers across different facilities. This interface, shown in Figure 7, should expose actions for adding and removing providers, as well as managing provider access.

Figure 7, Medical Assistant provider management wireframe
The tools used to implement this application included Angular 6 and the Health Catalyst Cashmere module. The use of Angular allows a full JavaScript web stack as well as object-oriented design.

There were several design patterns used with a layered architectural pattern to implement the proof of concept, including reactive programming, unidirectional data flow, and centralized state management. Some of these patterns, while not strictly enforced, were built into the Angular 6 framework [8].

During the implementation, the final design (see Figure 8) added an extra integration with the SETH application for users to upload files, such as images. This added extra control for patients to add information to their profile, without requiring intervention from the provider or facility.

![Figure 8, final Medical Assistant application design](image)

4 Discussion

Throughout the research, there were major concerns with patient data security. However, further concerns with diagnosing prediction accuracy were brought up in a meeting with potential investors. Regarding the diagnosing accuracy, it was determined to be better to alert the provider at an arbitrary threshold (e.g. 5% identification) than to throw a possible false-negative.

Additionally, concerns with data storage efficiency were discussed, primarily around potential data duplication. Because a record in IPFS is given a hash address based on its content, concern was raised that two records could match and only one stored. This was addressed by including the patient identifier in the records being uploaded.

By adding this unique identifier, two similar documents would be stored separately. This would also maintain the deduplication capability of the IPFS network, maintaining data storage efficiency.

5 Future Work

5.1 SETH

The next priority for the SETH dApp is improving the development pipeline to package the modules and scripts into a single module. This allows faster deployment, by providing the capability to drop the files in place and run without having to download or configure the application.

Future development may also be focused on providing functionality for storing and retrieving a standard health data format, HL7. This feature, discussed by potential investors, could allow the platform to integrate with existing electronic health record platforms.

5.2 Artificial Intelligence

Further research of other deep learning models and how they can improve the accuracy of this project are going to be explored. CNN and RNN models are very popular models and this project could benefit extremely from them. Additional research into an AI library known as OpenCV could prove useful in being able to detect bad data before the training takes place.

The current architecture has several limitations which would prevent further adoption. Some of these limitations include lacking configuration options, limited visualizations, and poor scalability.

During testing, it was discovered that multiple training processes required intense system resources. This critical limitation may be addressed by implementing a scheduling and management module, like an operating systems’ management of processes.

A design limitation of this architecture is the need to redesign, retrain, and redeploy for each user story. Introducing an AI as a Service platform may help mitigate this issue.

5.2.1 AaaS Frontend

The AaaS platform would work on a request-report architecture, and support REST API calls.

A configuration manager may also be implemented with the AaaS platform to allow the creation of different AI models dynamically. This allows a technical user to configure an AI without having to modify the existing codebase, further improving system maintainability.

5.3 Medical Assistant Web Application

Future iterations of the Medical Assistant application will include complete integration with the SETH and AI architectures. Including Ethereum interactions with encrypting and decrypting data.

After a stable application is developed, a mobile application framework may be implemented to provide the same functionality on Andriod and iOS platforms.
A provider or facility targeted application may be designed and developed to allow access to the patient data, without having to export the data.

This application would provide more controls for updating a patient’s profile based on the care given. Further integration with the AI architecture would also be implemented to provide diagnosing assistance for the providers.

6 References


