Homebase: Automating multi-party workflows via a Zero-Knowledge VM

Ethan Gordon
Falls Technology, LLC
ethan.gordon@fallstechnology.com

Many workflows of a business, government, non-profit, or any organization, require digital interaction or interoperability with other similarly situated organizations. For instance, a medical provider must digitally interoperate with their patients as well as the patients' insurers. Data must move from each party while being subject to privacy concerns and regulation, as well as financial concerns put forth in the respective parties' contractual agreements. Moreover, there are many different steps of computation that occur in order for these workflows to properly be executed. The inordinate amount of time and money spent on verifying, reconciling, interpreting, and finally executing these workflows can be saved by automation made possible from zero-knowledge proof technology. In this paper we put forth a system, Homebase, that can accomplish such automation. While our first use-case is medical billing, Homebase's architecture has been generalized to handle all digitized multi-party workflows.

1 Introduction

This author has recently been dealing with a seemingly intractable problem of interoperability between a large bank, and multiple DMV's. The problem is that the author paid off the lien for his vehicle , officially ending his obligation to the bank/lien-holder, and subsequently requested the title of his vehicle. Ideally, all that solving this problem should involve is the bank, who owned the car and thus the title, simply sending him the title. However, no party, including the DMV's where the vehicle has been previously registered, know where the title is. In short, all these parties have separate databases that cannot speak to each other–which is unfortunate because each party must interact in order to execute the workflow involved in obtaining a title. It has been 9 months since the vehicle was paid off–the title is still in limbo. This problem acts a very clear example of multi-party workflows and their present state– highly inefficient and, admittedly, highly frustrating.

Similarly, and more importantly, the healthcare industry is especially poor at interoperability. This is no secret. The US Dept. of Health and Human Services (HHS) has charged the Office of the National Coordinator for Health Information Technology (ONC) with developing best standards for interoperability within the industry. For the ONC, in order to solve for true interoperability, an application must provide a seamless process for exchange of health information between two or more systems, so that each system may *use* the information once it is received [18]. With the developments of HL7 and FHIR, international standards of health data formatting and transmission, have helped drive interoperability, but systems and processes simply remained siloed–causing inefficient and vulnerable workflows that are frustrating for patients and all users involved.

In order for workflows as vital as those in healthcare to be automated, each party must be able to trust not only every party involved, but also each computation executed within the workflow. A Zero-Knowledge Virtual Machine provides the environment needed for trustable execution of code and

Submitted to:

generation of zero knowledge proofs (zkps). For the unitiated, a much more detailed explanation of the tech stack will follow, but for now, we will use Wikipedia's definition for zkps: "method by which one party (the prover) can prove to another party (the verifier) that a given statement is true while the prover avoids conveying any additional information apart from the fact that the statement is indeed true" [7]. For instance, one could theoretically prove that their credit score is above or within a certain range without revealing what their credit score actually is [8].

In this paper we put forth a design of a system meant to automate multi-party workflows that currently suffer from inefficiencies and insecurities. We call this system Homebase; symbolizing a particular party's data rounding all bases securely and efficiently, undergoing the requisite computations in order to come home, visible in their own system of record and user interface of choice. We first describe the architecture of homebase, diving into the details of the tech stack. We then examine related applications and research concerning healthcare workflows and secure data exchange specifically, detailing where Homebase is different and, we think, better. Lastly, we run through a medical-billing example–explaining how Homebase would be used in such an environment.

The past several years has showcased a persistent increase in research and investment in applications that purport to provide a framework for secure healthcare data exchange via blockchain technology [13], [17], [12], [10], [16], [9], [11], [15], [3]. While each implementation pushes the research and technology forward, there has yet to be an application that provides a scalable, industry-ready solution to the healthcare interoperability problem. In order to solve for true interoperability, an application must provide a seamless process for exchange of health information between two or more systems, so that each system may *use* the information once it is received [18]. We argue that homebaseHEALTH (HBH) provides such a process.

As an implementation of the Baseline Protocol, HBH acts a secure exchange mechanism for patients, providers, and payers. Our first use case of concern is the medical billing process. Medical billing carries clear requirements for secure interoperability. Patients, providers, and payers must all coordinate to some extent–and they all rely on some trust mechanism. For instance, payers trust that providers will not submit multiple claims for the same service, or "upcoding"–billing for a pricier service than what was actually provided for the patient [2].While this first use-case will assume honest medical providers, we are primarily concerned with syncing the datasets of all three parties. We want to provide a process for patients that informs them of every step of the medical billing process that involves their personal medical data, as well as the fees they may be charged. Many patients experience "surprise billing" when asynchronous coordination issues arise between the three parties [19]. HBH solves this.

In this paper, we will first provide a high-level overview of homebaseHEALTH and its architecture. We then discuss related work and how HBH differs from previous blockchain implementations in the healthcare interoperability space. Next we will discuss the Baseline Protocol, why its important, and how HBH implements the standard. While blockchain is an important technology to the Baseline Protocol, the cryptographic technology of zero-knowledge proofs is perhaps equally vital. After we discuss our implementation of the Baseline Protocol and the specific technology stack utilized, we will describe how HBH solves medical billing.

2 High-Level Overview of homebaseHEALTH

3 Related Work: A Short Survey of Blockchain Applications for Healthcare Interoperability

Much of the research around applying blockchain to healthcare interoperability centers on Electronic Health Record (EHR) management [17], [13] [11]. Others focus on applying blockchain to healthcare applications and records in mobile devices [10] [16]. And another area of interest is integrating blockchain and edge computing to optimize the exchange of medical information [9]. The applications that we will focus most on are those that utilize zero knowledge proof (ZKP) technology [12], [3]. These ZKP applications will have many similarities with HBH and we borrow notation from [12]. In general, much of the HBH architecture is based on pieces of research from this fascinating literature.

3.0.1 Mobile Applications and Edge Computing

For instance, in [9], the authors propose a system utilizing edge computing and a blockchain management program that generates a different blockchain configuration based on the type of information obtained via their edge-computing platform. While interesting, this implementation is unlikely to scale as a different blockchain configuration for different channels of data would struggle to handle changing numbers and types of validators–as requests for data varies. Moreover, the system described does not utilize zero-knowledge proofs or any other proof-based technology that produces scalability. Another flaw would be releasing the medical data on-chain–leaving it vulnerable to blockchain's inherent transparency; thus, violating HIPAA.

Nguyen et al propose the use of blockchain for secure EHR sharing via mobile devices [10]. The authors put forth a thorough and interesting architecture that utilizes the InterPlanetary File System (IPFS) for the storage of medical records, noting how medical data cannot be stored on blockchain, a Distributed Hash Table (DHT) that stores hash values of the IPFS nodes, which then is sent to a transaction pool to be mined by miners of the ethereum blockchain. Access control is managed by a designed EHRs management and smart contract scheme. While the authors have developed something that works as a prototype, it would require industry-wide adoption and major alteration to hospitals' data storage methods. This isn't a realistic expectation.

3.0.2 EHR Management

Healthcare interoperability often is synonymous with the standards Fast Healthcare Interoperability Resources (FHIR) and Health Level 7 (HL7). Both FHIR and HL7 are standards for sharing clinical data between systems, where HL7 sets general standards and FHIR specifies standardized data formats and the shape of the data to be transmitted and received. Zhang et al designed FHIRChain, "a blockchain-based architecture for clinical data sharing" [13]. FHIRChain is the most similar design to homebaseHEALTH within this research space. The authors devised a way to share medical data without needing a new database storage system–hospitals can keep their system of record. Moreover, FHIRChain keeps medical data off-chain by only storing secure references to databases that can be accessed via smart contracts. The permissions to gain access derive from a token-based permission model. We argue that while FHIRChain is a great architecture, it could benefit from zero-knowledge technology and a non-token-based permission model. Moreover, the transaction fees will be exposed to end-users–something homebaseHEALTH avoids with roll-ups.

3.0.3 Utilization of Zero-Knowledge Proof Technology

Zheng et al develop a blockchain-based insurance claim system for the exact use-case homebaseHEALTH will be applied to in this paper [12]. Their paper is one that we will often revisit in this paper, as it is most similar to how HBH will be implemented. The authors define their system model as comprising patients, hospitals, insurance companies, private key generation (PKG), blockchain, and certificate authorities (CA)s. The system is comprised of three phases: a feeding authentic data (FAD) phase, a privacy-preserving transactions (PPT) phase, and an identity privacy-preserving (IPP) phase. Below is a quick breakdown of the process the authors propose. Technical details of the process will be expanded upon in sections 3.2 and 5. We will use the notation of [12] throughout the rest of the paper for continuity.

1. An insurance company B publishes requirements c_i for purchasing insurance contracts by the smart contract. The smart contract will be released in the blockchain.
2. A patient A will obtain medical data X from the hospital which is designated by B, think of an "in-network" vs "out-of-network" designation. The hospital will generate a unique ID for A
3. The hospital then builds an arithmetic circuit $\mathbf{C}$ that is constructed according to requirements c_i, which is then used to generate a ZKP, the price of the insurance contract M, a hash value h_x, based on X and ID. The hospital then encrypts M with a public key of the insurance company pk_B and a digital signature σ_{h_x} on the hash value h_x, resulting in h_M. The hospital submits the results to a smart contract, and the transaction will be recorded on-chain.
4. The encrypted value h_M will be decrypted as M with B's private key sk_B, and the smart contract executes an algorithm outputting whether the transaction is valid or not.
5. If the transaction is valid, the smart contract notifies A to transfer M to B. (This step involves the verification of whether the services provided by the hospital fall within the scope of the contract. If it does, then M represents a co-pay, for instance.)
6. The last step involves the authentication of A's identity so that B can ensure that they provide the rest of the insurance coverage (i.e., the bill minus A's co-pay) M_{claim}.
7. Lastly, B transfers M_{claim} to the hospital–ending the claim process.

The process described above is roughly what the authors propose. In this paper, we will dive deeper into some areas of the process by providing more explanation about Zheng et al's specific implementation, as well as imputing some of our own design decisions that we think necessarily fills gaps. Overall, the authors provide some great research and also cite a C++ implementation along with performance metrics.

The MediBloc project claims to provide a "patient-centered health data ecosystem" via its Panacea blockchain [3]. Panacea was built on the Cosmos SDK and Tendermint.[1] The technology used by Tendermint and why it's important to us will be addressed in later sections. For now, treat Tendermint as a black box. Sitting on top of Panacea's Tendermint implementation is a Decentralized Identifier (DID) component and a Data Exchange Coordination component. See the image below for reference. MediBloc's decentralized applications (dApps), Dr.palette, Medipass, etc., are interesting and claim to provide similar functionality that HBH will provide–for instance, Medipass "allows individual patients to conveniently utlize medical information stored and used by different hospitals and institutuions for

[1]The Cosmos SDK is a framework for building blockchains [1]. Tendermint Core is a "Byzantine Fault Tolerant (BFT) middleware that takes a state transition machine - written in any programming language - and securely replicates it on many machines" [4].

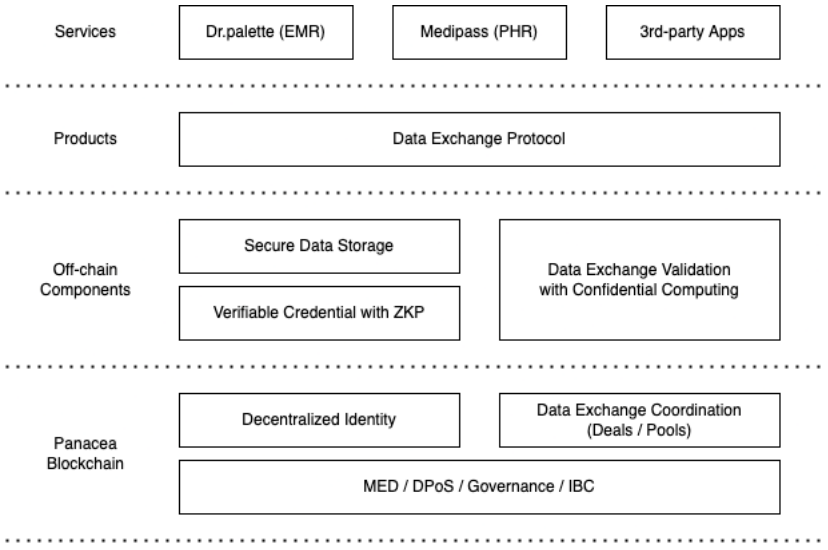

Figure 1: The MediBloc Tech Stack. See documentation here: https://medibloc.gitbook.io/panacea-core/overview/panacea-ecosystem

treatment and vaccination history management, insurance claims, etc." [3]. However, we are interested in *how* these apps are made possible by the technology underneath them, and, specifically how the technology is different from that of HBH's.

MediBloc tackles the problem of sharing medical data by classifying and converting healthcare data into verifiable credentials (VCs). VC technology is based on the World Wide Web Consortium's (W3C) published standard [5]. It is interesting to consider medical data as VCs because VCs are more often equated to a driver's license or a diploma–data that verifies one's identity, training, or education. For instance, a common example presented is usually that of a University Degree. W3C's VC trust model involves (i) an Issuer, the University, (ii) a Holder, the degree holder, and (iii) a Verifier, the hiring manager, for instance. MediBloc's VC component uses ZKP's to mask certain data fields that the Holder of the medical data does not want revealed, while still verifying data integrity. The specific technology used for this functionality are BBS+ signatures, which will be expanded upon in further sections.

Because medical data cannot be transacted with on-chain, MediBloc created an "off-chain decentralized oracle powered by confidential computing" [3].[2] Utilizing Intel's Software Guard Extensions, MediBloc set up a system where data can only be decrypted by oracle nodes that are running in a secure enclave. If the oracle node successfully verifies the data, it is then re-encrypted for the data consumer.

Lastly, MediBloc must present an option for where all the medical data will be stored. In the documents, the IPFS is considered but eventually passed on because of it essentially being a public network. It seems that MediBloc has not yet figured out the data storage problem, but hopefully HBH can.

[2]Oracle nodes are network nodes that connect blockchains to external systems [6]

4 Baseline Protocol

4.1 Ethereum as Mainnet

4.2 Zero-Knowledge Proofs

5 Medical Billing

6 homebaseHEALTH: solving medical billing

7 Bibliography

References

[1] *Cosmos SDK*. https://docs.cosmos.network/main/intro/overview. Accessed: 10-19-2022.

[2] *Healthcare Fraud*. https://www.fbi.gov/how-we-can-help-you/safety-resources/scams-and-safety/common-scams-and-crimes/health-care-fraud. Accessed: 9-29-2022.

[3] *MediBloc*. https://medibloc.gitbook.io/panacea-core/. Accessed: 10-9-2022.

[4] *Tendermint Core*. https://github.com/tendermint/tendermint. Accessed: 10-19-2022.

[5] *Verifiable Credentials Data Model v1.1*. https://www.w3.org/TR/vc-data-model/. Accessed: 10-19-2022.

[6] *What Is a Blockchain Oracle*. https://chain.link/education/blockchain-oracles. Accessed: 10-19-2022.

[7] *Zero-knowledge proof*. https://en.wikipedia.org/wiki/Zero-knowledge_proof. Accessed: 11-5-2022.

[8] *zkDocs: Zero-knowledge Information Sharing*. https://a16zcrypto.com/zkdocs-zero-knowledge-information-sharing/. Accessed: 11-5-2022.

[9] Abdellatif et al (2021): *MEdge-Chain: Leveraging Edge Computing and Blockchain for Efficient Medical Data Exchange*. IEEE Internet of Things Journal.

[10] Dinh C. Nguyen et al (2019): *Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems*. IEEE Access 7, pp. 66792–66806.

[11] Guo et al (2018): *Secure Attribute-Based Signature Scheme With Multiple Authorities for Blockchain in Electronic Health Records Systems*. IEEE Access.

[12] Houyu Zheng et al (2022): *A novel insurance claim blockchain scheme based on zero-knowledge proof technology*. Computer Communications, pp. 207–216.

[13] Peng Zhang et al (2018): *FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data*. Computational and Structural Biotechnology Journal 16(3), pp. 267–278.

[14] Sharma et al: *Blockchain-based Interoperable Healthcare Using Zero-knowledge Proofs and Proxy Re-Encryption*.

[15] Xia et al (2017): *MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain*. IEEE Access.

[16] Xueping Liang et al (2017): *Integrating Blockchain for Data Sharing and Collaboration in Mobile Healthcare Applications*. IEEE Xplore.

[17] Ariel C. Ekblaw (2017): *MedRec: Blockchain for Medical Data Access, Permission Management and Trend Analysis*.

[18] The Office of the National Coordinator for Health Information Technology (2013): *The Path to Interoperability*.

[19] Centers for Medicare & Medicaid Services: *No Surprises: Understand your rights against surprise medical bills*. `https://www.cms.gov/newsroom/fact-sheets/no-surprises-understand-your-rights-against-surprise-medical-bills`. Accessed: 10-15-2022.