Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.DOI

VigilRx: A Scalable and Interoperable Prescription Management System Using Blockchain

ALIXANDRA TAYLOR*[1], AUSTIN KUGLER*[2], PRANEETH BABU MARELLA[3], AND GABY G. DAGHER[4]

[1]North Idaho College, United States
[2]University of Idaho, United States
[3]George Mason University, United States
[4]Boise State University, United States

Corresponding author: Gaby G. Dagher (e-mail: gabydagher@boisestate.edu).

The authors acknowledge the generous support of funding by the National Science Foundation Computer and Information Science and Engineering (CISE) REU program, grant number 2051127.

*These authors contributed equally.

ABSTRACT Achieving interoperability between healthcare providers is a major challenge. Current systems for managing prescription records suffer from data siloing, unnecessary record duplication, and slow record transfers. In many systems, patients do not retain control over their prescription data. Instead, they must use an intermediary to access or transfer their records. Furthermore, record transfers suffer from differing standards between providers, outdated communication methods, and information blocking. Solving these problems necessitates the creation of an interoperable prescription management system. Realizing such a system requires considering security, efficiency, scalability, and other challenges. Recent regulatory actions attempt to address these challenges, but fundamental issues persist. This paper proposes a patient-centric and interoperable prescription system that ensures patient control, prevents information blocking, and improves transfer efficiency. We call our solution VigilRx—a system that uses blockchain and smart contracts to manage prescriptions. Stakeholders exist as one of three role-based smart contracts within the system: patient, prescriber, or pharmacy. These role contracts ensure the system is patient-centric by assigning ownership of prescription records directly to patients. Our smart contracts also ensure the system's interoperability, as we use a standardized prescription contract to ensure records can be easily managed. VigilRx's use of blockchain also promotes transparency by providing patients an explicit list of parties that hold permission to access their records. Use of existing software patterns allows the system to adapt as needed. We implement VigilRx and show that it is both scalable and efficient.

INDEX TERMS blockchain, electronic health records, Ethereum, healthcare, prescriptions, smart contracts

I. INTRODUCTION

MANAGING the transfer and control of healthcare records among prescribers, pharmacists, and patients presents ongoing concerns within the U.S. healthcare system. Healthcare records often become fractured between various care providers resulting in siloed, duplicated, or inconsistent information [10]. This partly results from the inefficient communication channels providers currently use [9]. Patients rarely have direct control over their records and must rely on care providers to transfer records on their behalf. Simply starting these transfers can be excessively slow with most facilities taking 5-10 days just to initiate a transfer [29]. The transfer process itself often requires a third-party health information service provider (HISP) who may participate in information blocking practices. Additionally, the sending and receiving parties may not share a common HISP or other compatible transfer system. In these cases, antiquated technologies such as fax machines or physical mail become necessary intermediaries that add to the delay. Even in cases where record transfer through a HISP is possible, differing storage standards and data formats may necessitate further translation before the data is useful within the receiving

party's system.

The realizing of an interoperable electronic health record (EHR) solution faces regulatory and security hurdles as well. The sensitive nature of patient EHRs makes them a valuable target for theft. Providers must facilitate the secure transfer and storage of healthcare records while also complying with shifting regulatory standards. The time sensitive nature of EHR transfers requires that system designers accommodate both efficiency and security. The need for efficient exchange is challenging for small providers who must pay for access to expensive HISP platforms [23]. Challenges regarding security and efficiency become especially relevant in the transfer of prescription records. Untimely or inaccurate transmission of sensitive data inhibits stakeholder access and obstructs informed decision making.

Current solutions meant to address these challenges rely on the exchange of prescription records between large HISP companies. These companies have a vested interest in protecting their market share which often leads them to engage in information blocking practices [8] [31]. Information blocking consolidates control for the blocking party and forces the use of antiquated technologies. Information blocking challenges the legally recognized idea that patients should be in control of their own health records (in the U.S., under the Health Insurance Portability and Accountability Act [26]). This paper proposes the use of blockchain technology as a method for ensuring patient control over their records.

The overhead costs of record transfers between healthcare providers is substantial. As of 2017, 61% of hospitals used more than one method for receiving electronic healthcare records with 33% using at least four [22]. The 21st Century Cures Act prompted the U.S. Department of Health and Human Services (HHS) to address these issues through regulatory reforms [24]. However, even under these reforms patients still do not have direct control over their records. These regulations only attempt to simplify record transfers and do not directly ensure patient control over their personal data. VigilRx's design addresses the concerns raised by existing HISP-centered infrastructure and restores record control to the patient using blockchain technology.

Other systems, such as MedRec [4], propose blockchain-based alternatives to the HISP-centered model of EHR management. Unlike our solution, MedRec focuses on an EHR solution at large while our system addresses the unique challenges of dealing with prescriptions specifically. The presence of a global registrar entity in the VigilRx system adapts ideas originally presented by MedRec. The Ancile [3] system proposes the use of encryption techniques to ensure secure data transfer and allow the storage of small health records on the blockchain, like VigilRx. A later system, called OpTrak [2], focuses on prescriptions but does not explore alternatives to HISP control of patient data. OpTrak uses blockchain as an access control mechanism to establish communication between siloed databases. Our system advocates for allowing patients to own and distribute access to their prescription records directly. More recently, SecureRx [1] proposes integration between a blockchain-based prescription solution and the existing federal opioid database RxCheck. This use of existing databases is unlike VigilRx. Our system does not rely on existing tracking systems but can still operate alongside them. In addition, a stakeholder within the SecureRx system interacts with a SecureRx server instead of the blockchain directly. VigilRx provides stakeholders with direct access to the blockchain and relevant data, decreasing the reliance on additional infrastructure.

VigilRx is a prescription management system that facilitates interoperability among stakeholders while remaining patient-centric. The system (1) provides patients direct authority over who can access their prescription data, (2) prevents information blocking by ensuring patients can perform actions related to their data without reliance on an intermediary, and (3) ensures prescription data can be efficiently transferred among relevant stakeholders.

To accomplish these three goals, VigilRx uses blockchain and smart contracts. Smart contracts define and enforce the actions stakeholders may take within the system. They allow prescribers to originate prescriptions but automatically transfer ownership to patients. Interactions are ultimately under patient control with patients deciding on a case-by-case basis how to distribute access to their data. Patient control becomes the default under our system, replacing abstraction through third party services.

The role of blockchain as a public and distributed ledger enables patient access to prescription records. We accomplish on-chain storage of prescription data using national drug codes (NDC). These codes are drug identification numbers already managed by the U.S. Food and Drug Administration [21]. This approach allows VigilRx to maximize accessibility and minimize transaction costs without compromising blockchain's value as a tool for access control. VigilRx also makes use of software patterns that allow for its implementation to evolve. Software patterns like abstract factory and flyweight provide a mechanism for future implementations of the system without compromising the utility of existing VigilRx data on the blockchain.

VigilRx proposes a patient-centric and interoperable prescription tracking solution. Our system combats information blocking by offering an alternative to traditional HISP-centered models of prescription data transfer. Originating prescriptions with the prescriber and automatically transferring ownership to the patient allows the flow of data to begin with the most affected stakeholders. Patients gain power over their data by consenting to its distribution on a case-by-case basis and have a clear record of permissioned parties. Our system may provide the most utility to small, rural, and critical access care providers with limited resources by removing the cost of unnecessary intermediaries that disproportionately affects these providers.

A. CONTRIBUTIONS

The contributions of this paper are as follows:

1) We introduce VigilRx, a blockchain-based prescription management system that aims to replace existing HISP architecture and prevent information blocking. VigilRx utilizes roles to determine which actions stakeholders may perform with a specific emphasis placed on patient control. Patients actively participate in the management of their prescription data. As a result, VigilRx achieves a patient-centric design that is interoperable, scalable, and efficient.
2) VigilRx achieves this patient-centric design by building a record of stakeholder access within the system and requiring active patient consent for actions related to their prescriptions. Patients are given constant access to a clear record of permissioned stakeholders with access to their data. Furthermore, patients must provide consent during relevant stages of their prescription management. Our system actively verifies the role of all parties in order to protect users and ensure that only the appropriate parties may access or alter records.
3) VigilRx achieves a patient-centric design by allowing patient consent to actions related to their prescriptions. Patients have constant access to a record of parties they have previously provided consent to. We verify the role of all parties during this consent process to ensure only those with patient permission may access records. A global registrar contract ensures the integrity of this verification by authenticating the role of each relevant stakeholder.
4) The free and efficient exchange of data between stakeholders is a vital part of our system. We achieve this interoperability by enabling the exchange of data without the requirement for translation between formats. Our system avoids data duplication between silos and accomplishes the exchange of prescription records without unnecessary overhead common in HISP-oriented systems. Therefore, patient data is current and readily accessible to stakeholders unlike in passive and siloed HISP systems.
5) To enable features not normally available in a blockchain-based system, we apply software patterns. Our system utilizes three software patterns to achieve adaptability (abstract factory pattern), efficiency (pub-sub pattern), and scalability (flyweight pattern) [7]. Abstract factory allows the creation of new role contract implementations in our system; pub-sub allows notification of changes in the status of patient prescriptions; flyweight enhances scalability by avoiding unnecessary data duplication.
6) We implemented VigilRx using Ethereum smart contracts. Our experimental testing results show that VigilRx is scalable with respect to the number of stakeholders. The results also show that VigilRx is efficient with most actions in the system being relatively low-cost once prescriptions have been initialized. As seen in our efficiency evaluation (see section VI-E), most actions remain under an average cost of 90,000 Ethereum gas.

II. RELATED WORK

A few research papers: Ekblaw et al. [4], Dagher et al. [3], Zhang et al. [2], and Alnafrani and Acharya [1] explore blockchain's use in healthcare, specifically regarding electronic health records (EHRs). Some of these articles focus on using blockchain as an access control method for external health record databases. Others utilize blockchain as a transparent logging mechanism to record actions related to personal health records.

Ekblaw et al. [4] created MedRec, a patient-centric modular architecture for managing EHRs. MedRec focused on providing faster access to health records, improving interoperability, increasing patient control, and providing data to researchers. VigilRx also focuses on increasing patient control and interoperability. However, we specifically address challenges unique to prescription records. MedRec used blockchain as an access control method to external databases without storing any medical records directly on the blockchain [4]. VigilRx differs in that it does not utilize off-chain data storage. MedRec is also notable for its use of anonymized healthcare data as a mining reward for its blockchain. In MedRec, a notification system informs patients on the usage of their data [4]. VigilRx ensures patient consent similarly to MedRec's notification system, although we add multi-signature confirmations to the process.

Dagher et al. [3] designed the Ancile system which used blockchain technology for access control. Ancile specifically focused on patient-centric control, security, and data sharing. In their system, on-chain data references are used to query established off-chain databases. Similarly, it is possible to implement VigilRx alongside existing databases. The Ancile system used Ethereum smart contracts and cryptographic methods to accomplish its goals. Notably, Ancile used proxy re-encryption to facilitate the secure transfer of records [3]. Ancile also proposed the storage of small, encrypted records directly on the blockchain to speed up record transfers [3]. In VigilRx, we use existing standards to allow the on-chain storage of small prescription records in a similar manner. Another unique aspect of Ancile is its use of a consensus algorithm based on JP Morgan's Quorum [3]. VigilRx differs in its goal of being blockchain agnostic, where it can exist on any smart-contract enabled blockchain without requiring a specific consensus algorithm.

Zhang et al. [2] proposed a system called OpTrak that focused heavily on using blockchain to address the U.S. opioid crisis. The system's goals were the prevention of data hoarding, doctor shopping, and over-prescription. OpTrak uses the Ethereum blockchain as an access control mechanism for referencing existing prescription databases. OpTrak aimed to quickly inform providers of important events, such as the filling of an opioid prescription [2]. VigilRx differs from OpTrak as we focus on building a prescription management system that allows for direct control of records instead of simply tracking stakeholder access. Furthermore,

VigilRx does not limit its focus to opioids, instead focusing on prescription records in general.

Alnafrani and Acharya [1] developed the SecureRx framework to facilitate interstate cooperation between existing Prescription Drug Monitoring Program (PDMP) databases. SecureRx particularly specified integration with the existing RxCheck PDMP database. Their system focused on prescription tracking and not management. In the system, SecureRx delivers patient prescription information to PDMPs for tracking purposes [1]. SecureRx's focus on prescriptions is like VigilRx but different in its intended use as a prescription tracker and reliance on RxCheck. SecureRx used Ethereum smart contracts and a web application, like our VigilRx system.

Other related works focused the use of smart contracts for prescription management, such as Garcia et al.'s [5] research on low-cost smart contract prescription systems. Thatcher and Acharya developed and tested a prescription system called RxBlock that featured user and prescription smart contracts [6]. Work by A. Gropper [14] researched the issue of identity management in the context of blockchain EHRs and proposed a method of associating patient identities with a blockchain identifier. Further work by Zhang et al. [16] applied the emerging HL7's Fast Healthcare Interoperability Resources (FHIR) standard in a blockchain-based EHR system called FHIRChain. Ruby Benita et al. [15] proposed a system that used Ethereum and a mobile app for prescription drug tracking and authentication. Zhang et al. [7] also developed a system called DASH that served to identify use cases for blockchain in healthcare and evaluated design considerations for its effective application.

III. BACKGROUND

A. HIPAA & HITECH

Systems developed for use within the U.S. healthcare industry must comply with relevant laws. These include the Health Insurance Portability and Accountability Act (HIPAA) of 1996 and the more recent Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009. Title II of HIPAA seeks to establish legal standards for the use of EHRs, create national provider identifiers (NPIs) [20], and promote widespread EHR use. Due to its strong focus on EHRs, Title II is most relevant to digital systems dealing with healthcare records.

Title II includes three distinct parts labeled 45 CFR Parts 160, 162, and 164. Parts 160 and 164 are especially relevant to blockchain-based prescription solutions. Part 160 defines relevant terms and establishes the obligations of involved parties. The section defines covered parties as organizations with a role in the transfer of electronic protected health information (e-PHI). Any organization considered a covered party must comply with the guidelines of HIPAA. VigilRx fits into this category due to its handling of e-PHI. Part 164 includes two sections vital to HIPAA compliance: the Privacy Rule [32] and the Security Rule [33].

The Privacy Rule is central to HIPAA, given its stated purpose to "define and limit the circumstances in which an individual's protected health information may be used or disclosed by covered entities." The Privacy Rule also defines protected health information: "all individually identifiable health information held or transmitted by a covered entity or business associate, in any form or media, whether electronic, paper, or oral" [32]. This definition includes any data pertaining to the:

- Individual's past, present, or future physical or mental health
- Provision of healthcare to the individual
- Past, present, or future payment for the provision of healthcare to the individual

The Privacy Rule goes on to enumerate the circumstances under which the usage or disclosure of protected health information is acceptable. Covered entities must disclose PHI when a patient requests it, or when the U.S. Department of Health and Human Services (HHS) is undertaking a compliance investigation, review, or enforcement action. Covered entities may use and disclose PHI without authorization in some cases. For example, covered entities can use PHI for their "own treatment, payment, and health care operations" [32]. VigilRx enhances transparency in regards to this provision because patients can view a list of entities that have permission to use their data.

Protecting patient prescription data is another major concern of HIPAA as defined within the Security Rule. The Security Rule can be understood in terms of its general rules [33], the first being to "ensure confidentiality, integrity, and availability of all electronic protected health information the covered entity or business associate creates, receives, maintains, or transmits." Additionally, confidentiality requires that e-PHI is not made available to unauthorized persons and its integrity is preserved. The preservation of integrity requires e-PHI "is not altered or destroyed in an unauthorized manner" [33]. VigilRx stores small prescription records directly on the blockchain which inherently protects its integrity. The blockchain itself is effectively immutable and therefore a record of data alterations is available.

Under the Security Rule, "availability" means all e-PHI must be accessible to any authorized person who demands it. The use of blockchain serves this requirement well by providing patients ready access to their data and taking away the cumbersome requirements of shuffling records between the different databases. Whether through a mobile app, home computer system, or other device, the patient simply needs a blockchain address in order to access their prescription information.

The provisions of HIPAA were further expanded in 2009 with the addition of the HITECH Act, focusing on making electronic copies of patient health records available to patients on request and tightening the maximum allowable time period for the delivery of such information. HITECH also strengthened requirements that providers keep disclosures of PHI "to the minimum necessary to accomplish the intended

purpose of such use, disclosure, or request..." [28]. This is another area in which blockchain naturally excels, allowing rapid record retrieval from the chain and providing patients with logs showing who can access their e-PHI.

B. INFORMATION BLOCKING

Healthcare information blocking occurs when persons or entities knowingly and unreasonably interfere with the exchange or use of electronic health information [31]. According to the Office of the National Coordinator for Health Information Technology (ONC), an information blocking incident must meet three criteria:

- **Interference.** Information blocking requires some act that interferes with the ability of authorized persons or entities to access, exchange, or use electronic health information. This can result from explicit policies prohibiting the sharing of information, or from more subtle business, technical, or organizational practices that add to the cost or difficulty.
- **Knowledge.** The decision to engage in information blocking must be knowingly made.
- **No reasonable justification.** Not all conduct that knowingly interferes with electronic health information exchange is information blocking. Concerns related to privacy, security, and patient safety can be interpreted as valid reasons for engaging in information blocking practices.

Examples of information blocking practices include contract terms that restrict access to EHR, charging prohibitive transfer fees, and implementing systems in ways that substantially increases the cost. As far back as 2015, ONC identified the prevalence of these practices and several of their side effects [31]. Information blocking slows EHR transfers, increases their cost, and fractures data between competing entities. As of 2021, information blocking practices continue to be an issue. Of surveyed health information exchange organizations, 55% experienced information blocking from EHR vendors and 30% experienced information blocking from healthcare providers [8].

The most commonly observed information blocking methods were HISPs charging unreasonably high transfer fees and large healthcare providers refusing to share patient records. These common information blocking methods suggest that EHR vendors and HISPs are using the premise of privacy, security, and safety to obscure a conflict of interest that benefits them financially while also preventing realization of the legal and ethical requirements of HIPAA.

C. BLOCKCHAIN

Satoshi Nakamoto originally proposed the concept of blockchain in the 2008 Bitcoin whitepaper [13]. Blockchain is a method of storing data in a distributed manner using a network of computers without reliance on a central authority. Groups of data exist as blocks that are "chained" together in chronological order from oldest to newest using a cryptographic method called hashing. Participants in a blockchain system store these blocks on their personal computers and communicate with each other to ensure they remain the same across the network. An individual's data is considered invalid if it is inconsistent with the majority of other participants.

One goal of blockchain is immutability, meaning data cannot change after it is stored. This property exists so long as most participants do not act maliciously. As a result, blockchain is extremely secure because any attacker must control the majority of participants before they can modify the blockchain-stored data. For the purposes of blockchain, the importance of participants is determined by the amount of computational power they can contribute to the network.

D. ETHEREUM & SMART CONTRACTS

Ethereum is a specific implementation of blockchain created by Vitalik Buterin in 2013 [12]. Like all blockchain implementations, it collects groups of transactions within the network into blocks and uses cryptographic hashes of previous blocks to secure the immutability of the chain's contents. Ethereum accomplishes block selection using a proof-of-work algorithm like that proposed in the Bitcoin whitepaper [13]. Ethereum is notable as it allows the inclusion of smart contracts, whose implementation can structure the flow, behavior, and accessibility of on-chain data.

Smart contracts are software programs stored on a blockchain. Programmers can use smart contracts to create decentralized applications (dapps) that expand the blockchain's functionality. For example, smart contracts can serve as an abstract representation for system users, providing them with relevant functionality. The use of smart contracts requires the paying of a fee, measured in an Ethereum-specific unit called gas. The most popular programming language for Ethereum smart contracts is called Solidity.

E. WEB3 INTERFACE

Web3 is a software library that allows programmers to connect traditional applications with smart contracts or other decentralized systems. Currently, the Web3 library is available for use in several popular programming languages, including JavaScript, Java, Python, and Go. The system proposed in VigilRx utilizes the Python distribution of this library, known as Web3.py.

F. GANACHE BLOCKCHAIN

Ganache [18] is a blockchain intended for developing and testing decentralized applications locally without having to interact with outside peers. Ganache allows developers to test smart contracts intended for deployment on the live Ethereum blockchain. Ganache's testing environment is configurable, allowing multiple tests to execute using the same settings.

G. SOFTWARE PATTERNS

Software patterns are general, reusable solutions to commonly occurring problems in software design. Software patterns are not typically represented by source code or libraries

but rather as a template that addresses a given challenge. Our system discusses the application of three such templates.

1) Abstract Factory

Abstract factory [7] is a pattern that builds on the concept of a "concrete factory" and increases its flexibility. A concrete factory is any software object that creates another object of a specific and static type. An abstract factory encapsulates a group of these individual factories who share a common theme. The goal of this pattern is to separate the details of an object's implementation from its general use as a member of a common class.

2) Flyweight

Flyweight [7] is a pattern dealing with data composed of simple, repeatable elements. The goal of flyweight implementation is to minimize memory usage by maintaining a consistent access structure between data objects. Flyweight works by storing commonly accessed data and providing references to its storage locations. This allows objects to access the data through references without storing it themselves. Thus, avoiding unnecessary data duplication.

3) Publisher-Subscriber

Publisher-Subscriber (pub-sub) [7] is a pattern that applies to the sending and receiving of messages. In the context of smart contracts, a pub-sub pattern is useful for monitoring changes in the status of an on-chain contract. For example, if a patient needs a fill or a refill of their prescription they can simply change a state variable within their smart contract (via a web app). Subscribed care providers, such as the prescription's issuer or a filling pharmacy, can actively monitor these state variables for changes. When such a change occurs the care provider can then delegate any necessary actions off-chain without accruing additional cost or processing strain on the blockchain itself.

IV. SOLUTION: VigilRx

A. OVERVIEW

VigilRx is an interoperable and patient-centric prescription management system where record management is accomplished through blockchain technology. Our system uses smart contracts to allow all actions necessary within a usable prescription system. Our system is role-based with a user's role defining the actions they may take within the system.

We represent these roles using the following smart contracts: the global registry contract for an administrator; prescription contract for medications; patient role contract for patients; prescriber role contract for prescribers; and pharmacy role contract for pharmacists. This model of contracts allows the creation of new role contract instances, prescribing, filling of prescriptions, requesting of refills, and permissioning required to initiate these processes. We make these actions accessible to non-technical stakeholders using a modern web application.

B. SOFTWARE COMPONENTS

The VigilRx system consists of two main software components: a web application and a decentralized application (dapp) running on smart contracts. A library known as Web3 serves as the bridge between these two components. We use an encryption scheme to ensure all HIPAA-protected information is secure.

The web application serves as a non-technical access point for the blockchain-based dapp. Every stakeholder creates an account via the web app through a process that requires specifying their respective role, providing relevant information on themselves for validation, and then associates the user with a smart contract on the blockchain. A public-private key pair assigns ownership of this role contract to the stakeholder. The public key identifies the stakeholder, and the private key allows the signing of contract actions. This scheme serves to verify the keyholder's identity through asymmetric cryptography. The structure of the role contracts further determines what actions are available through the web app.

The VigilRx dapp runs on smart contracts written in the Solidity programming language. New role contracts originate with a global registry contract (GRC) which uses the abstract factory pattern to create a role contract of the appropriate type for the given user.

The behavior of these role contracts follows their respective code implementation but also inherits from one of four standardized interfaces: patient, prescriber, pharmacy, or prescription. These interfaces allow for querying data within the system via the flyweight pattern and offer standard functions for all basic inter-contract actions.

The Web3 library serves as a bridge between these two components. Use of this library allows data to be fetched from the blockchain and displayed on the web app without requiring users to interact with the chain directly. Web3 also allows the web app back-end to take advantage of the pub-sub pattern and regularly check specific contracts for status changes. The front-end can then notify the user to act if needed. For example, the front-end will notify a prescriber of a refill request.

C. SYSTEM SMART CONTRACTS

VigilRx proposes six smart contract designs: the global registry contract, concrete factory contract, prescription contract, patient role contract, prescriber role contract, and pharmacy role contract. Figure 1 shows a UML diagram of the relationships between the principle smart contracts.

1) Global Registry Contract

The global registry contract (GRC) is responsible for user registration, validating user roles, and ensuring the creation of role contracts. User registration occurs through the transmitting of user data using the existing Transport Layer Security (TLS) protocol. System administrators validate this data and authorize the creation of the appropriate role contract. The GRC uses the abstract factory pattern and a mapping

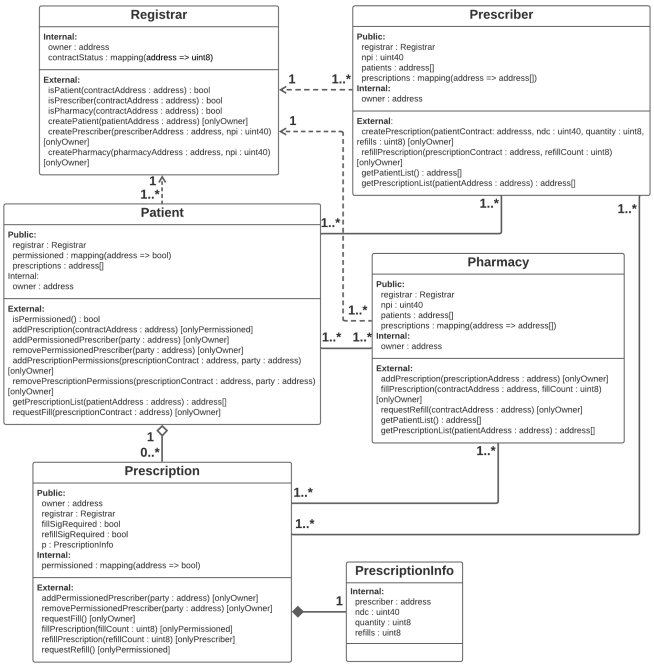


FIGURE 1. VigilRx role and prescription smart contracts.

of concrete factory contracts (CFC) to initiate this creation process and commits a new role contract to the blockchain. The user the receives the role contract address from the GRC.

The GRC is also the entry point for new types of concrete factory contracts. When the addition of a new CFC type is necessary, the owner of the GRC (effectively a system administrator) can verify the factory's address and add it to the GRC's mapping. This new address can override or augment existing CFC implementations for entirely new functionality.

Lastly, the GRC tracks the contract addresses of all role contracts and their classifications. This allows contracts within the system to verify the status of a contract address with the GRC before allowing certain actions. For example, if a patient is permissioning a new address, the patient's role contract can check the provided address against the GRC to ensure it belongs to a valid provider within the VigilRx system.

2) Concrete Factory Contract

There are multiple concrete factory contracts (CFC) in our system that create role contracts on behalf of the global registry (GRC). Each concrete factory creates a specific type of contract, inheriting functionality from one of the four standard interfaces in our system (prescription, patient, prescriber, or pharmacy). When coupled with the abstract factory pattern via the GRC, these new CFCs create alternate implementations of role contracts despite the immutable nature of blockchain smart contracts.

One alternate implementation might be a prescription for a medication with a consumption limit. Such a contract could prevent over-prescribing by tracking a prescription's quantity over a patient's lifetime. Another example is a special contract for tracking controlled substances. Smart contracts could convene with on-chain oracle services [19] or integrate with existing prescription drug monitoring programs. Solutions like these can exist on top of the VigilRx system without compromising the usefulness of existing contracts.

3) Prescription Contract

Prescription contracts represent patient prescriptions within the VigilRx system. All prescriptions include the contract address of the prescriber, the contract address of the owner (the patient), and a list of permissioned contracts (e.g. filling pharmacies). Also included is the national drug code of the medication prescribed, the quantity of the medication, and the number of refills remaining. Each prescription contract includes two special flags that allow the pub-sub pattern to indicate when a prescription is ready for filling or refilling.

Permissioning, filling, and refilling are handled inside the prescription contract. It provides functions that are callable by other role contracts within the system. Outside of the prescription contract, role contracts call a prescription contract function to check the sending address against the contract owner, prescriber, or permissions list as appropriate. The information in the contract is only altered if the request is made in the proper way by an authorized party. Flags indicating a fill or refill request can also be set at this point.

4) Patient Role Contract

Patient contracts represent unique patients in the VigilRx system. They include a list of the patient's prescription contract addresses and a mapping of prescriber contract addresses authorized to write prescriptions for the patient. Every patient contract has functionality for adding and removing permissioned prescribers, as well as adding and removing permissions from owned prescriptions. Additionally, the patient contract has a method for initiating a fill request by triggering a flag change on any owned prescriptions.

In our system, patient representation relies on a patient's role contract address without on-chain storage of a personally identifying information.

5) Prescriber Role Contract

Prescriber role contracts represent unique prescribers within VigilRx. Each prescriber contract has a list of patient contract addresses under their care, as well as a mapping of prescription contract addresses for each patient. Each prescriber contract includes a national provider identifier (NPI) from the National Plan and Provider Enumeration System (NPPES) database. Referencing this database allows the fetching of relevant data about the prescriber such as contact info, medical specialty, and licensing status [20].

Prescriber role contracts are the main vehicle by which new prescription contracts enter the VigilRx system. This process begins with a prescriber generating a new prescription contract using the necessary prescription data and the patient's role contract address. The patient's role contract then

receives the prescription contract address. If the prescriber has permission to write prescriptions for the patient, then the patient contract adds the new prescription address to an internal list. Otherwise, the process fails and any changes to the blockchain are reverted.

Prescribers have special privileges over any prescriptions they originate in this way. They may modify the number of fills at any time and make use of filling/refilling flags to facilitate patient care as needed.

6) Pharmacy Role Contract

Pharmacy role contracts represent pharmacies within the VigilRx system. Like the prescriber role contract, all pharmacy contracts have a list of patient contract addresses and a mapping of the prescription contracts for each patient. Each prescriber contract also includes an NPI.

Pharmacy role contracts are the primary means by which patients can get their prescriptions filled. To initiate that process, the patient must first give the pharmacy permission on one or more of their prescription contracts. The patient can then initiate a fill on one of their prescriptions, changing the fill flag. The change of this flag alerts the pharmacy to start filling the prescription. Once the patient receives the medication, the pharmacy interacts with the contract again to confirm the removal of one fill and reset the flag.

If a fill request occurs and the prescription is out of fills, then the pharmacy can set the prescription's refill flag. The change in this flag alerts the prescriber to review the prescription for refilling. If the prescriber approves the refill the pharmacy can then continue the rest of the filling process.

D. SYSTEM ACTIONS

The following sections demonstrate the architecture of VigilRx by showing how our system operates during the writing and filling of prescriptions. We provide diagrams for clarity, laying out the involvement of our smart contracts during the five primary actions within our system: *role contract creation*, *permissioning*, *prescribing*, *filling*, and *refilling*. We discuss these transactions in terms of the stakeholders for the sake of clarity but it should be understood that actions occur through role contracts and not the stakeholders directly. See Table 1 for a breakdown of system actions available to stakeholders.

TABLE 1. Actions available to each stakeholder.

	Prescription Actions						
Role	Create	View	Fill	Req. Fill	Refill	Req. Refill	Cancel
Patient	N	Y	N	Y	N	N	N
Prescriber	Y*	Y	N	N	Y	N	Y
Pharmacy	N	Y*	Y†	N	N	Y*	N
*Requires patient permission †Requires patient signature							

1) Role Contract Creation

In order to create a new role contract, a stakeholder must send a request and any necessary information to the administrator of the global registry contract (GRC). This information transfer uses the Transport Layer Security (TLS) protocol to ensure the security of any personally identifying data. Included with this information is the public key of the stakeholder and additional details such as the National Provider Identifier (NPI) of the stakeholder if they are a prescriber or pharmacist.

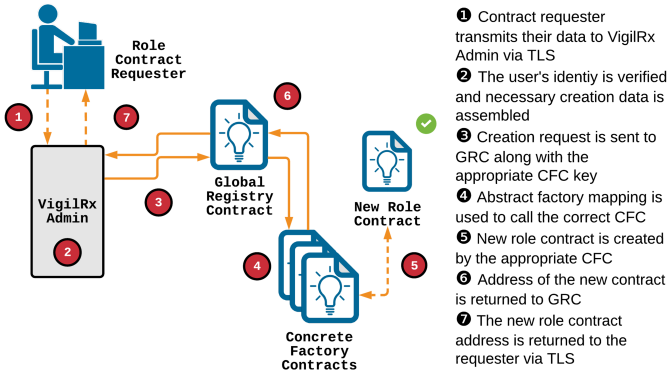


FIGURE 2. Role contract creation in VigilRx.

Contract creation takes place on-chain via the GRC through use of the abstract factory pattern. The administrator transmits the stakeholder's public key and any other necessary information to the GRC, along with a key referencing the GRC's mapping of concrete factory contract (CFC) addresses.

The GRC uses the key-specified CFC to create a new role contract on the blockchain and assigns contract ownership to the provided public key. The CFC returns the address of this newly created contract to the GRC, where it is registered according to its role. The original requester then receives the newly created role contract address. Any personally identifying information necessary to establish the stakeholder's identity at the administrative level is never placed on-chain.

2) Permissioning Action

VigilRx is patient-centric, meaning all access to prescription information originates with the patient. We achieve this through a permissioning process that is started on-chain using the patient's role and prescription contracts. The remainder of the process is completed off-chain using key encapsulation and exchange.

There are two cases in which a patient may need to permission a healthcare provider. The first case occurs when a patient wishes to allow a prescriber to write prescriptions for them. The second occurs when a patient wishes to give a pharmacist permission to fill their prescriptions.

In the first case the prescriber is permissioned via the patient's role contract. To begin this process, the patient

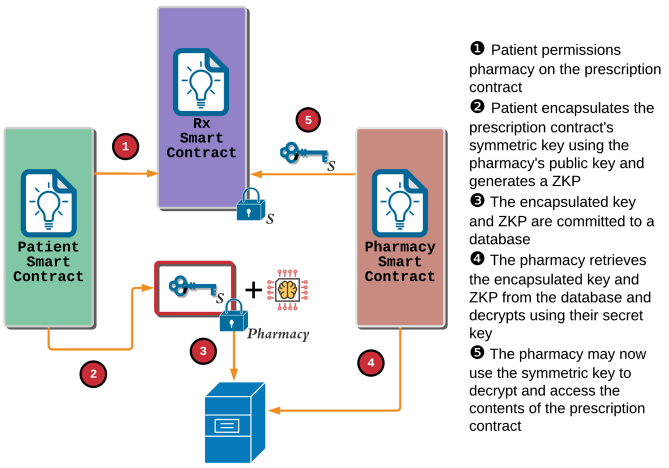


FIGURE 3. Permissioning in VigilRx.

provides the role contract address of the desired provider to their own role contract. The patient's role contract confers with the global registry contract and verifies whether the provided address belongs to a valid prescriber registered within the system. This step prevents the accidental or malicious entering of an invalid address. Once validated, this address is added to a permissioned parties mapping within the patient's role contract and the process is complete.

The process to permission a pharmacy begins similarly. The patient provides the address of a pharmacist role contract which is then validated through the GRC. However, the contents of the patient's prescription contracts are encrypted in order to ensure privacy (see section IV-D3). Decryption of the prescription contract's contents requires the distribution of a symmetric key associated with the prescription (S). Generation of this key occurs during the prescribing process (see IV-D3) and the patient stores it locally.

In order to remain selective, the distribution process must target the intended recipient. We achieve this targeting by encapsulating S using the target's public key, via asymmetric Elgamal encryption as follows:

Let $(\alpha_1, \beta_1) = (S.h_1^{r1} \ mod \ p, g_1^{r_1} \ mod \ p)$ be the Elgamal ciphertext encrypting secret key S using the public key $h_1 = g^{x_1}$ of provider P_1, where g_1 is the generator of group G whose order is a large p.

The patient performs the following:

1) Produces ciphertext $(\alpha_2, \beta_2) = (S.h_2^{r2} mod p, g_2^{r2} mod p)$ encrypting S with public key $h_2 = g_2^{x2}$ of provider P_2.
2) Uses Zero Knowledge Proof to prove that:
   a) The key S obtained by decrypting (α_1, β_1) with x_1 is the same key encrypted in (α_2, β_2) : $\alpha_2 = \alpha_1.\beta_1^{-x1}.h_2^{r2} \ mod \ p$, and
   b) The secret key corresponding to (h_1, g_1) is x_1, and
   c) The randomness used to generate (α_2, β_2) is r_2.

Once S encapsulation and the zero-knowledge proof (ZKP) generation occurs, a database receives and stores both of these. The intended party can then retrieve the encapsulated key, decrypt it using their own secret key, and access the contents of the patient's prescription contract.

3) Prescribing Action

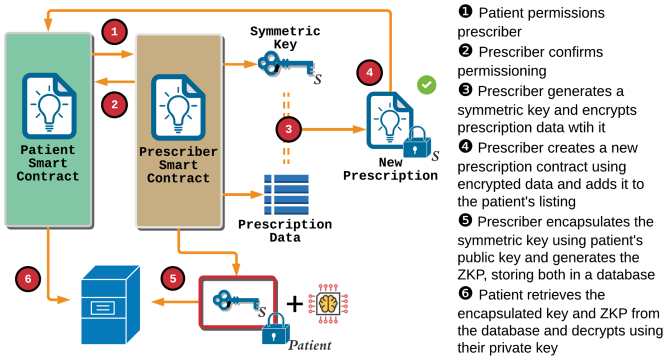


FIGURE 4. Prescribing action in VigilRx.

Writing a new prescription in the VigilRx system first requires the patient to permission the prescriber to issue prescriptions on their behalf (see section IV-D2). In order to confirm that this process is complete the prescriber checks the patient's role contract to validate their address, ensuring its presence within the patient's permission mapping. Once permissioning confirmation occurs, the prescriber generates a symmetric key (S) that encrypts sensitive prescription information, such as the national drug code (NDC) of the medication and quantity being prescribed. This encryption is necessary to ensure HIPAA compliance and ensure patient control of future data access via the selective distribution of this key.

Next, the prescriber encapsulates the symmetric key using the public address of the patient through Elgamal encryption and generates a zero knowledge proof (see section IV-D2). Both are then committed to a database for future retrieval by the patient.

This process could potentially occur on-chain rather than using a public database. For the purposes of our implementation, on-chain key exchange proved infeasible due to specific limitations with the Ethereum blockchain. Moving this process on-chain could be a fruitful area for future work.

After storing the encapsulated key, the prescriber assembles the prescription, encrypting the NDC and quantity using S. The prescriber then commits the encrypted data and the patient's role contract address to the prescriber role contract.

The role contract creates a new prescription contract on-chain using this encrypted data and assigns ownership of the prescription to the patient using their role contract address. This process calls an additional function on the patient's role contract directly, adding the address of the newly minted prescription contract to the patient's listing. This ensures the

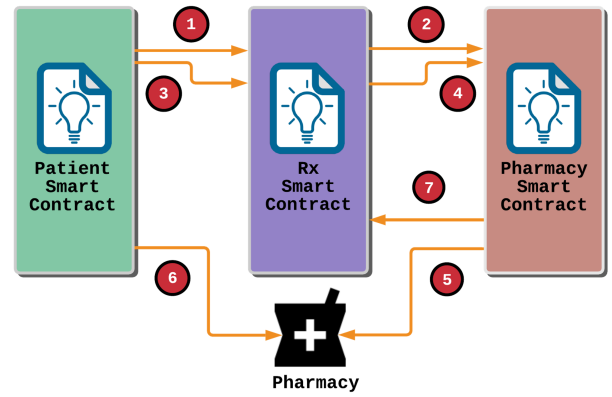
preservation of the prescription contract address throughout the creation process and prevents manipulation.

At this point, the patient can retrieve the encapsulated key and decrypt the prescription's contents. This gives them access to their data and allows them to distribute access to future care providers or pharmacies.

4) Filling Action

Filling begins with the patient permissioning the pharmacist on one of their prescription contracts (see section IV-D2). The pharmacy verifies this permissioning and adds the patient's prescription contract to the listing on their pharmacy role contract. The patient may then initiate a fill request by accessing the prescription contract. This sets the prescription's fill flag which notifies the pharmacy that the prescription requires filling. A listing of the patient's prescription contracts allows for the easy monitoring of these fill flag via the publisher-subscriber pattern.

After notification, the pharmacy can then physically prepare the prescription. When the patient arrives to pick up the prescription, the pharmacy turns over the medication and then completes the fill on the prescription contract. The fill flag resets, and the transaction is complete. Only the patient can initiate this two-stage process and only the pharmacy can complete it. This approach prevents either party from requesting more than the correct share of fills and other malicious behavior.



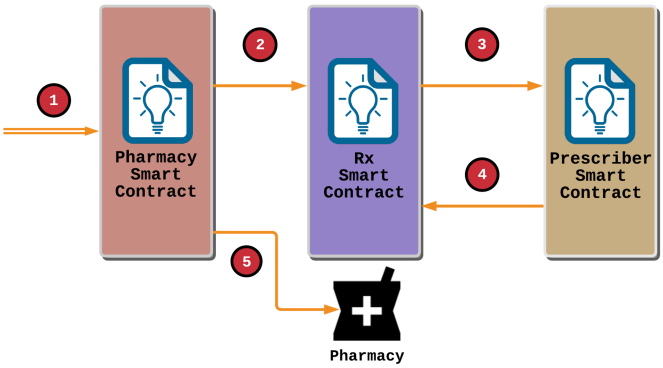
❶ The patient permissions the pharmacy on the smart contract
❷ The pharmacy adds the prescription and subscribes to the fill flag
❸ Patient sets the fill flag
❹ Pharmacy reads the fill flag change
❺ The pharmacy fills the prescription
❻ The patient picks up the filled perscription
❼ The pharmacy resets the fill flag, recording the transaction completion on-chain

FIGURE 5. Prescription fill action in VigilRx.

5) Refilling Action

Refilling adjusts the number of fills available on a prescription contract. The prescription prescriber can directly modify the number of refills at any time by calling the appropriate prescription contract function.

A different process occurs when a patient requests a fill with insufficient fills remaining. In this case, the pharmacist can request a refill from the prescriber. The prescription contract's refill flag is set, prompting the prescriber for approval. The prescriber then has the choice to authorize more fills or deny the request.



❶ Patient initiates a fill request on a prescription with no fills left
❷ The pharmacy sets the refill flag
❸ Prescriber reads the refill flag change
❹ Prescriber refills the prescription, adding more fills to the available total and resetting the refill flag
❺ The pharmacy continues with filling the prescription

FIGURE 6. Prescription refill action in VigilRx.

V. DISCUSSION

A. BLOCKCHAIN LIMITATIONS

VigilRx has certain limitations. Blockchain's status as an immutable ledger makes removing data in accordance with laws like the General Data Protection Regulation an inherent challenge which our system does not try to address. VigilRx also does not solve issues concerning identity verification, assuming an administrative compliance with current standards at the system's entry point. VigilRx does not directly attempt to address issues such as over-prescription, prescription errors, or filling errors that might originate from care providers, however it does have potential uses in tracking them.

B. ETHEREUM & ENCRYPTION

The encryption of on-chain personal health information is a necessary part of the VigilRx system. Our use of a public and permissionless blockchain makes the encryption of on-chain data vital in complying with HIPAA. One major limitation with Ethereum is its maximum data size of 256 bits. Given that the minimum key size for ElGamal encryption is 1024 bits, exchanging keys on-chain is not possible without high-overhead workarounds.

However, we designed our system with the intention of being feasible on any blockchain capable of running smart contracts. Using a different blockchain may make it possible

to accomplish key exchange entirely on-chain and enhance the decentralized nature of our system.

C. GDPR COMPLIANCE

The U.S. could potentially implement new legislation following the model of the European Union's General Data Protection Regulation (GDPR) [30]. GDPR includes the "right to be forgotten," meaning the right to have stored personally identifying information erased upon request.

While we see our system as taking the steps necessary to be compliant with HIPAA, the immutable nature of blockchain and our key-distribution scheme makes it difficult to comply with the right to be forgotten. Truong et al. [17] proposed a personal data management platform to address GDPR compliance. Integrating this platform into the VigilRx design may provide a sufficient solution to make our system compliant with GDPR.

VI. EXPERIMENTAL EVALUATION

A. IMPLEMENTATION

We implemented our proposed system for the purposes of testing scalability and efficiency. Most of our implementation focused on dapp development. However, we also built a Django-based implementation of the VigilRx web app to interface with our dapp as a proof-of-concept. Our basic web app implementation allows non-technical stakeholders to create role-based accounts for interacting with the VigilRx dapp.

Within the dapp, we built the global registry contract, prescription contract, and three role contracts proposed in our solution. We implemented these contracts in the Solidity programming language and included functionalities for all major prescription-related actions. We used the Python version of the Web3 library to build an object-oriented and comprehensive bridge between the smart contracts and our off-chain applications. We designed the bridge to interact with a local Ganache blockchain where deployment of our dapp occurred during each experiment. The bridge also connects our web app to the dapp.

Using the bridge, we created a testing suite that allows for simulation comparable to a real-world environment. In each simulation cycle, stakeholders perform their relevant prescription actions. Patients select and permission random providers (prescribers and pharmacies), prescribers create new prescriptions for their patients, and pharmacists fill prescriptions every cycle. Each stakeholder completes all their available actions in each cycle, although prescription creation must occur in the first cycle before some actions can be performed.

B. TESTING ENVIRONMENT

The system used for our experiments contains an Intel® Core™ i5-7400 processor, 16 gigabytes of DDR4 memory, and a Kingston A400 solid state drive, running the Ubuntu 20.04 operating system. During experimentation, we used Ganache version 6.12.2 on this machine.

Ganache allows for the specification of environment settings. We use the Muir Glacier fork of Ethereum, which is the newest version available through Ganache at the time of writing. We use Ganache's default gas limit of 6,721,975 units and no artificial delay in mining time. For any given experiment, we initiated Ganache with accounts equal to the number of stakeholders, plus one to represent the global registry contract. Accounts were each initialized with the Ganache default balance of 100 ether. We also set Ganache's *keepAliveTimeout* to 10,000 to decrease the likelihood of connection timeouts during testing.

C. SIMULATION SETUP

Our testing suite design allowed us to define the distribution of different stakeholder role within the simulation. We specified a ratio of 60% patients, 30% prescribers, and 10% pharmacies. This distribution approximates the real-life 3-to-1 ratio between doctors and licensed pharmacists in the United States, while still ensuring enough contracts for each role at low volumes [25] [27].

We used static values for all contract generation that occurred during our experiments. We populated NPI and NDC values with 5555555555. This number represents an average NPI or NDC value as both are 10-digit numbers. We populated prescription quantity, fill count, and refill count with the value 5. We chose 5 because it is a reasonable representation of single-digit numbers and ensures prescriptions require refilling in all our experiments.

Under these conditions, we evaluated the VigilRx system regarding scalability and efficiency. We allowed our simulation to run for 10 cycles in each experiment, increasing the number of stakeholders in each experiment. In each cycle, all stakeholders performed the actions relevant to their category at the same time. This number of cycles was chosen because it allowed us to conduct multiple tests within a reasonable amount of time, given our computing resources.

The distribution of actions performed by stakeholders in our experiments reflects our decision to use static values for the fill and refill variables. As shown in Table 2, the refilling action occurred the most as it is also the most common action in the real-world.

TABLE 2. Distribution of actions by category.

Action Category	Distribution
Fill	75.00%
Refill	8.33%
Prescribe	8.33%
Permission	8.33%

The full source code for our smart contract and simulator implementation is available here[1].

[1]https://github.com/vigilrxteam/VigilRx

D. SCALABILITY

We determined our system's scalability by measuring its runtime with an increasingly large number of stakeholders. The first test simulated 250 stakeholders and each subsequent test increased this number by 250, up to 2,500 stakeholders. The total runtimes and gas usage for each category of prescription actions were recorded.

Fig. 7 shows the runtime for each test, with the number of stakeholders on the x-axis and runtime in minutes on the y-axis. Additionally, each prescription action category appears individually in a stacked format. The total cost of running the system with increasingly more stakeholders is shown in Fig. 8. The x-axis contains the number of stakeholders during each test and the y-axis shows the Ethereum gas usage (in millions).

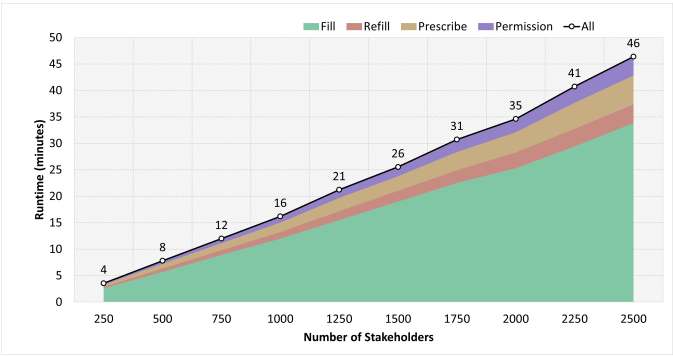


FIGURE 7. Scalability in relation to runtime.

We observed that VigilRx is scalable regarding runtime, given its linear increase with the number of stakeholders in the system. Fig. 7 shows this general linear trend, although minor deviations are present. It is likely these deviations occurred due to increased Ganache connection timeouts when it initialized with many accounts. Support of this theory lies in the higher deviations occurring at 2,250 and 2,500 accounts. At these levels, Ganache timeouts increased in frequency. It follows that the deviations reflect the abilities of the Ganache testing environment and not the VigilRx system itself.

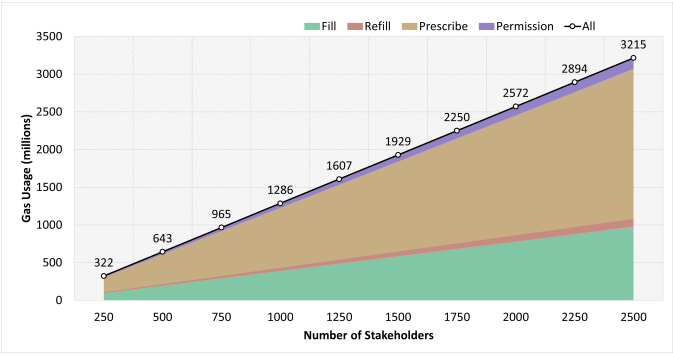


FIGURE 8. Scalability in relation to gas usage.

Testing shows VigilRx's near-perfect scalability regarding gas usage, as seen in Fig. 8. The linear increase of gas usage with number of stakeholders is obvious. The computation required for each prescription action does not change at scale and determines gas usage, thus the linear increase. It follows that gas usage is scalable in the same manner.

E. EFFICIENCY

We tested the efficiency of VigilRx as part of our experimental evaluation. Measurement of the average Ethereum gas use for each prescription action shows the system's per action cost. This data was produced in the same simulation used for determining scalability, allowing cross-reference between the system's efficiency and scalability. Efficiency tests focused on the average runtime and average gas usage for each type of prescription action.

Fig. 9 displays the average runtime for each category of prescription actions. The x-axis contains the number of stakeholders in each test; the y-axis is the average runtime in milliseconds. Fig. 10 shows the average quantity of gas used (in thousands) during each test. The x-axis is the same as in previous charts, with the y-axis representing Ethereum gas in the thousands of units.

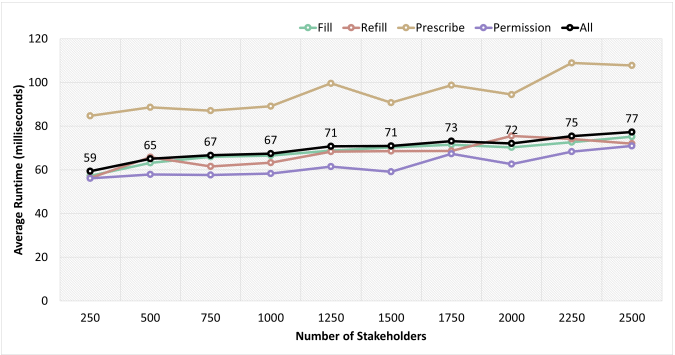


FIGURE 9. Efficiency in relation to runtime.

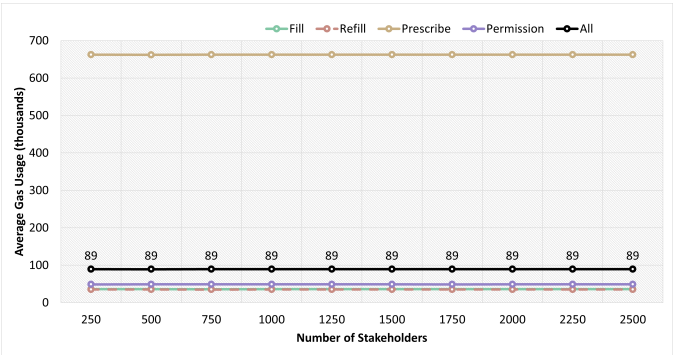


FIGURE 10. Efficiency in relation to gas usage.

Testing shows the efficiency of the VigilRx system in regard to runtime and gas usage. Most actions in the system require little overhead with average gas usage per action falling under 90,000. One category is notably more expensive than the others: prescribe actions. These actions include the

creation of new prescription contracts making the category's higher overhead expected (see Fig. 9). Prescription contract deployment has a high upfront gas cost, but subsequent prescription actions are cheap as they simply call functions on the newly deployed contract. This causes the overall average gas usage to remain relatively low. All action categories have near-constant average gas usage (shown in Fig. 10). Gas cost represents the computation required for an on-chain action and this computation does not change on a per-action basis under our design, regardless of the number of stakeholders.

Some abnormalities occurred in VigilRx's runtime during our experimentation. Fig. 9 shows a general upward trend for average runtime with variations especially prominent when the number of stakeholders reaches 1,750 and 2,000. The discussion on connection timeouts in section VI-D is one explanation for these deviations. General inconsistencies with computing hardware or software are other possible causes.

VII. CONCLUSION

In this paper, we sought to design a blockchain system for interoperable prescription management to give patients direct authority over their data and removing the need for intermediaries like HISPs. The implementation of the VigilRx system shows its ability to be deployed using smart contracts. Testing shows our system remains both scalable and efficient as the number of stakeholders increases. VigilRx promotes interoperability by removing existing differences in data formatting. The design removes the need for data duplication and prevents siloing. Our system also promotes a patient-centric design by requiring patient consent at every stage of the prescribing process. Use of a global registry further protects the patient by verifying the role of stakeholders throughout the permissioning process. Lastly, VigilRx has the ability to evolve. Use of software patterns allows future updates to the system and establishes a thorough record of patient care.

Technologies essential to our system are still under active research including smart contracts, identity management, and compliance with emerging regulation. Future research in blockchain-based prescription management systems could focus on topics like compliance with the General Data Protection Regulation (GDPR) [30]. Blockchain's immutability makes complying with GDPR's data removal provisions inherently difficult. Truong et al. [17] have already done significant research in this area but future regulation may require new solutions. Future work on the sharing of prescription data with researchers could also prove fruitful. Other systems, like MedRec [4], touch on this area of research but future work could develop new methods of data sharing. Additionally, future work could introduce drug compatibility checking to a VigilRx-like prescription management system. Such a system may benefit from use of the existing openFDA database to determine compatibility, perhaps informed by the patient's on-chain history of care.

REFERENCES

[1] M. Alnafrani and S. Acharya, "Securerx: A blockchain-based framework for an electronic prescription system with opioids tracking," *Health Policy and Technology*, vol. 10, no. 2, p. 100510, 2021.

[2] P. Zhang, B. Stodghill, C. Pitt, C. Briody, D. C. Schmidt, J. White, A. Pitt, and K. Aldrich, "OpTrak: Tracking opioid prescriptions via distributed ledger technology," *International Journal of Information Systems and Social Change*, vol. 10, no. 2, pp. 45–61, 2019.

[3] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable Cities and Society*, vol. 39, pp. 283–297, 2018.

[4] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," *2016 2nd International Conference on Open and Big Data (OBD)*, 2016.

[5] R. D. Garcia, G. A. Zutiao, G. Ramachandran, and J. Ueyama, "Towards a decentralized e-prescription system using smart contracts," *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*, 2021.

[6] C. Thatcher and S. Acharya, "Rxblock: Towards the design of a distributed immutable electronic prescription system," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 9, no. 1, 2020.

[7] P. Zhang, D. C. Schmidt, J. White, and G. Lenz, "Blockchain technology use cases in healthcare," *Advances in Computers*, pp. 1–41, 2018.

[8] J. Everson, V. Patel, and J. Adler-Milstein, "Information blocking remains prevalent at the start of 21st Century Cures act: Results from a survey of health information exchange organizations," *Journal of the American Medical Informatics Association*, vol. 28, no. 4, pp. 727–732, 2021.

[9] T. Shryock, "Sharing patient data: The challenges of healthcare interoperability," *Medical Economics Journal*, vol. 96, no. 5, pp. 14–19, 2019.

[10] B. A. Stewart, S. Fernandes, E. Rodriguez-Huertas, and M. Landzberg, "A preliminary look at duplicate testing associated with lack of electronic health record interoperability for transferred patients," *Journal of the American Medical Informatics Association*, vol. 17, no. 3, pp. 341–344, 2010.

[11] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *Advances in Cryptology*, pp. 10–18, 1984.

[12] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *EIP-150 REVISION*, 2014.

[13] S. Nakamoto, "Bitcoin: A Peer-to-Peer electronic cash system," 2008.

[14] A. Gropper, "Powering the physician-patient relationship with HIE of one blockchain health IT," *ONC/NIST use of Blockchain for healthcare and research workshop. Gaithersburg, Maryland, United States: ONC/NIST*, 2016. 18.

[15] R. Benita K, G. Kumar S, M. B, and M. A, "Authentic drug usage and tracking with blockchain using mobile apps," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 17, p. 20, 2020.

[16] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying blockchain to securely and Scalably share clinical data," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 267–278, 2018.

[17] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "GDPR-Compliant personal Data management: A blockchain-based solution," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1746–1761, 2020.

[18] Trufflesuite, "Trufflesuite/Ganache: A tool for creating a LOCAL blockchain for Fast Ethereum development.," *GitHub*, 2021. [Online]. Available: github.com/trufflesuite/ganache-cli.

[19] P. Grimaud, "Oracles," *ethereum.org*, 2021. [Online]. Available: ethereum.org/en/developers/docs/oracles/.

[20] "NPPES NPI registry," *U.S. Centers for Medicare & Medicaid Services*, 2021. [Online]. Available: npiregistry.cms.hhs.gov/.

[21] "National drug code directory," *accessdata.fda.gov*, 2021. [Online]. Available: www.accessdata.fda.gov/scripts/cder/ndc/index.cfm.

[22] "Methods used to enable interoperability among U.S. non-federal acute care hospitals in 2017," *The Ohio State University Wexner Medical Center*, 2017. [Online]. Available: wexnermedical.osu.edu/departments/catalyst-center/datacore/.

[23] "Hospitals' Use of Electronic Health Records Data, 2015-2017," *ONC for Health Information Technology*, 2019. [Online]. Available: www.healthit.gov/sites/default/files/page/2019-04/AHAEHRUseDataBrief.pdf.

[24] "The trusted exchange framework and common agreement draft 2," *ONC for Health Information Technology*, 2019. [Online]. Available: www.healthit.gov/topic/interoperability.

[25] "U.S. physicians - statistics & facts," *Statista*, 2020. [Online]. Available: www.statista.com/topics/1244/physicians/.
[26] "Health Insurance Portability and Accountability Act of 1996 (HIPAA)," *US Department of Health and Human Services*, 1996. [Online]. Available: www.cdc.gov/phlp/publications/topic/hipaa.html.
[27] "Number of pharmacists in the U.S. from 2001 to 2019," *Statista*, 2020. [Online]. Available: www.statista.com/statistics/185723/.
[28] "Minimum Necessary Requirement," *US Department of Health and Human Services*, 2013. [Online]. Available: www.hhs.gov/hipaa/for-professionals/privacy/guidance/.
[29] "How to request your medical records," *Drugwatch*. [Online]. Available: www.drugwatch.com/health/how-to-obtain-medical-records/.
[30] "General Data Protection Regulation (GDPR) compliance guidelines," *European Commission*, 2018. [Online]. Available: https://gdpr.eu/.
[31] "Report on health information blocking," *ONC for Health Information Technology*, 2015. [Online]. Available: www.healthit.gov/sites/default/files/reports/info_blocking_040915.pdf.
[32] "Summary of the HIPAA Privacy Rule," *US Department of Health and Human Services*, 2013. [Online]. Available: www.hhs.gov/hipaa/for-professionals/privacy/index.html.
[33] "Summary of the HIPAA Security Rule," *US Department of Health and Human Services*, 2013. [Online]. Available: www.hhs.gov/hipaa/for-professionals/security/index.html.

ALIXANDRA TAYLOR Alixandra Taylor is currently studying for a Computer Science degree at North Idaho College. Her research interests include decentralized systems, simulation design, gender studies, and applying computer science for social justice.

AUSTIN KUGLER Austin Kugler received his Associate of Science in Computer Science from North Idaho College in 2021. He is continuing his studies in Computer Science at the University of Idaho. His research interests include system software, FinTech, and EdTech.

PRANEETH BABU MARELLA Praneeth Babu Marella is currently pursuing a Master's in Information Security from George Mason University and works as a Senior Cyber Threat Analyst at Accenture Security. His work and research interests include threat intelligence/hunting, detection engineering, and cloud security.

GABY G. DAGHER Dr. Dagher is an assistant professor of Computer Science at Boise State University. He is the director of the Information Security, Privacy & Mining (ISPM) research lab, and the associate director of the Idaho Election Cybersecurity Center (INSURE). His work focuses on designing secure protocols and developing tools for solving real-life problems related to cybersecurity and applied cryptography, including blockchain and cryptocurrencies, cyber-forensics, and cloud computing.