Clinical Supply Blockchain Working Group

Transforming Pharmaceutical Clinical Supply Messaging with Blockchain

Chad Sklodosky (co-chair, Pfizer)
Imran Shakur (co-chair, Biogen)
Greg Plante (IQVIA)
Ben Taylor (LedgerDomain)

August 2019

Acknowledgements

The authors would like to thank their fellow Clinical Supply Blockchain Working Group members for their contributions and insight – Munther Baara, Keith A. Jackson, and Sachin Karnik (Pfizer); Jacob Bordens, Patrick Dougherty, and Jeffrey England (Merck); Jennifer Colon and Jess De Jesus (UCLA); Oliver Cunningham (Bracket); Mark Hanly and Mark McColgan (Almac Group); Jitendra Kumar (Thermo Fisher); Barry Moore (GlaxoSmithKline); and Joel Spangenberg (Marken).

To Pfizer, IEEE, and Almac, thanks for hosting our face-to-face meetings. To Dr. Victor Dods and Dr. Leonid Alekseyev (LedgerDomain), thanks for their roles in developing the blockchain-based platform and iPhone app that served as our demonstration project, termed “KitChain MVP.” Finally, thanks to GS1, especially co-chairman of the investigational products standard, Hans Von Steiger, for collaborating with us in the data science arena.

Executive Summary

Both clinical studies and their attendant supplies are exploding in number, complexity, and scope, with increasingly specialized storage conditions and shorter windows of expiration. Clinical sites are struggling to meet these demands, which have slowed enrollment and sometimes forced them to turn away new studies. With personalized medicine on the horizon, these trends are likely to become even more challenging. Over the course of nineteen months, the Clinical Supply Blockchain Working Group (hereafter “CSBWG”) defined the core message format, data flow, and sharing functionality requirements for a secure and streamlined solution to this challenge, with multiple rounds of development and testing on a prototype mobile application.

While sponsors and CROs have historically encouraged sites to use their proprietary systems, the sheer number of systems means that many sites have fallen back on paper documentation. The CSBWG was formed to explore blockchain as a way to address these growing demands, by providing a single access point to capture and share transactions with the parties that need them. As an immutable, time-stamped, near-real-time, auditable record of transactions, blockchain is well-suited to supply chains where numerous organizations handle sensitive information. After a vigorous technical review, the CSBWG selected an open-standard permissioned blockchain solution built on top of Hyperledger Fabric, an open source Linux Foundation project.

At the heart of this effort was KitChain, a mobile application designed to easily fit into the supply chain workflow. This iOS-based client was built to pass XML-based messages securely and seamlessly between shippers, recipients, and interested third parties. After a thorough investigation and design process, the application was successfully tested in May 2019 in an advance ship notification (ASN) and proof-of-delivery use case, with CSBWG members

assuming different roles within the supply chain – shipper, recipient, and interested third party. Overall, on a scale of 1 (missed requirements) to 3 (met requirements) to 5 (exceeded requirements), KitChain MVP scored 3.8, comfortably surpassing requirements.

A companion video outlining the challenges, vision, and workflow of the project is available at www.kitchain.org/video. The next stage of this project will involve expanding KitChain with role-based privileges, updated security, and GS1-standard unique identifiers, then evaluating it for business value and scalability.

Program Overview

The CSBWG started at a workshop November 1, 2017 in Cambridge, MA (USA) initiated by Munther Baara (chair) and the Pfizer Blockchain Center of Excellence. During the workshop, Chad Sklodosky of Pfizer and Imran Shakur of Biogen agreed to co-chair the working group. They established a charter and secured broad industry participation from multiple pharmaceutical and biotechnology companies, vendors to the pharmaceutical industry, and medical centers.

The initial output includes this white paper and a demonstration project, "KitChain," an iOS-based app which runs in Apple TestFlight on top of a live Hyperledger Fabric permissioned blockchain infrastructure. ***This paper is designed to stimulate discussion amongst clinical supply chain participants and particularly encourage the ongoing consolidation of the building blocks of data science harmonization and system interoperability.***

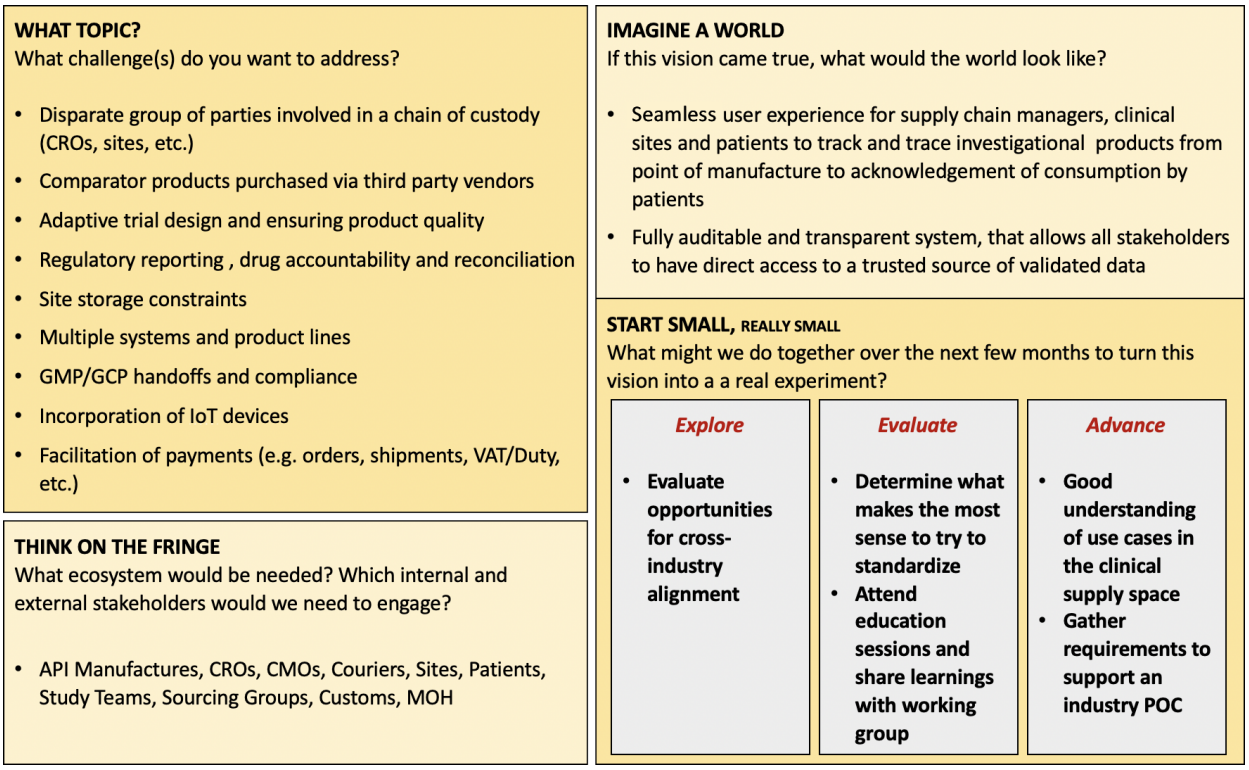
The medium-term goal of the CSBWG is to develop the fully interoperable, transparent and auditable platform that will enable investigational drug and comparators to be tracked from point of manufacture to the patient. Over the long term, the CSBWG envisions a messaging and notification platform that supports the performance requirements of the clinical supply chain, streamlining collaboration and cementing a single version of the truth.

While the pharmaceutical clinical supply chain is largely exempt from DSCSA[1] regulation, recent developments provide some waypoints and infrastructure to leverage. In particular, the GS1.org groups are seeking to harmonize the data science layers, although the unique requirements of the clinical supply supply chain – such as blinding – represent a challenge.

[1] The Drug Supply Chain Security Act (DSCSA) was enacted in 2013 as part of the Drug Quality and Security Act, which amended the Federal Food, Drug, and Cosmetic Act to grant the FDA more authority to regulate and monitor the manufacturing of compounded drugs. It requires an enhanced drug distribution security system for interoperable electronic tracing of product at the package level by 2023. See especially its electronic database requirements (127 STAT. 616). Additional resources: DSCSA Implementation Plan, FDA Overview of DSCSA, FDA Data Integrity and Compliance with Drug CGMP: Questions and Answers, FDA Standardization of Data and Documentation Practices for Product Tracing (draft), and Product Identifier Requirements Under the DSCSA – Compliance Policy.

CSBWG Charter

The CSBWG was created to address a range of challenges in the pharmaceutical clinical supply chain. Its ultimate vision is (1) a seamless user experience for supply chain managers, clinical sites, and patients to track and trace investigational products from point of manufacture to acknowledgement of consumption by patients, and (2) a fully auditable and transparent system that allows all stakeholders to have direct access to a trusted source of validated data.



The CSBWG Charter.

Why Blockchain?

While blockchain is often associated with cryptocurrency applications such as Bitcoin, its status as an *immutable, time-stamped, near-real-time, auditable record of transactions* makes it possible to enhance privacy and security across a range of collaborative applications.[2] The CSBWG saw blockchain as a potential route to allow hundreds of competing pharmaceutical and biotech companies and their vendors to work collaboratively and communicate with tens of thousands of clinical sites delivering care to clinical subjects.[3]

[2] See for instance NITAAC Solutions, Blockchain: Innovations in health data sharing with FDA and Booz Allen *(February 1, 2019).*

[3] For instance, at the time of writing, UCLA's team of 300 pharmacists supports over 670 active studies from nearly 150 sponsors.

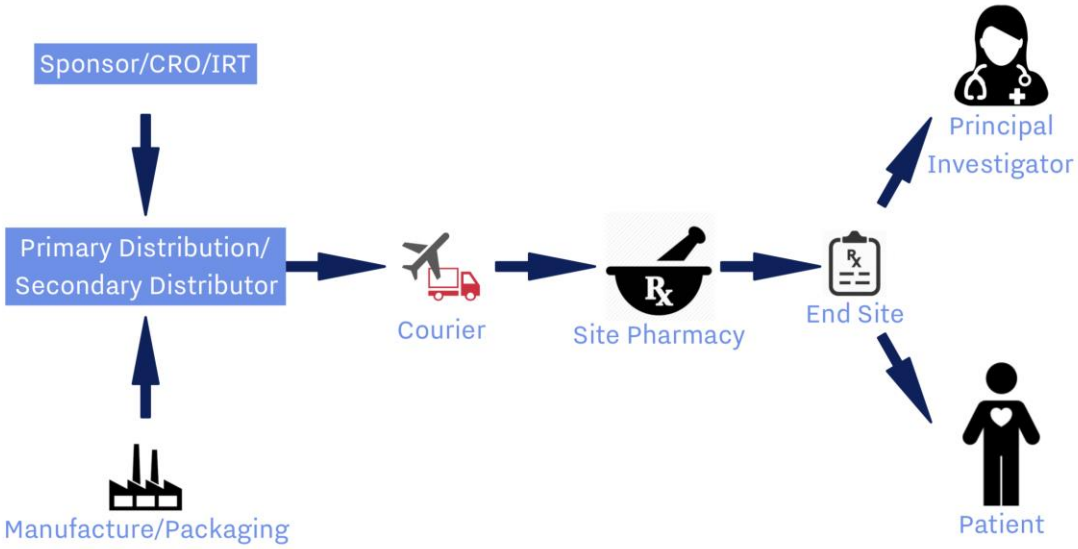
Blockchain is not a database, but rather sits atop a database. It is a time-stamped data organization, implemented with cell-level encryption. That encryption allows many participants to access their own cells – and not the cells of other users.

Each event within the blockchain occurs when parties agree to e-sign a transaction. The agreement, in turn, adheres to an associated "smart contract."[4] A contract makes each transaction both binding and irrevocable. If a transaction is struck between two parties, each will have keyed access, as might a regulator or auditor.

While we often think of blockchain in terms of use case, the key difference between blockchain systems lies in membership and data management. The data privacy and security requirements associated with healthcare demand a shared-permission blockchain-based system.[5] The CSBWG chose Hyperledger Fabric components as a scaffolding for the pilot, as well as a Fabric-based framework that allowed for off-chain private storage combined with blockchain-based authentication.

Clinical Supply Stakeholder Roles

The clinical supply chain is uniquely configured. Both the active and control medicines are provided free of charge, typically organized into kits, and are quite often blinded somewhere along the chain.[6] As such, role definition and separation is of paramount importance. Some of these roles are depicted below.

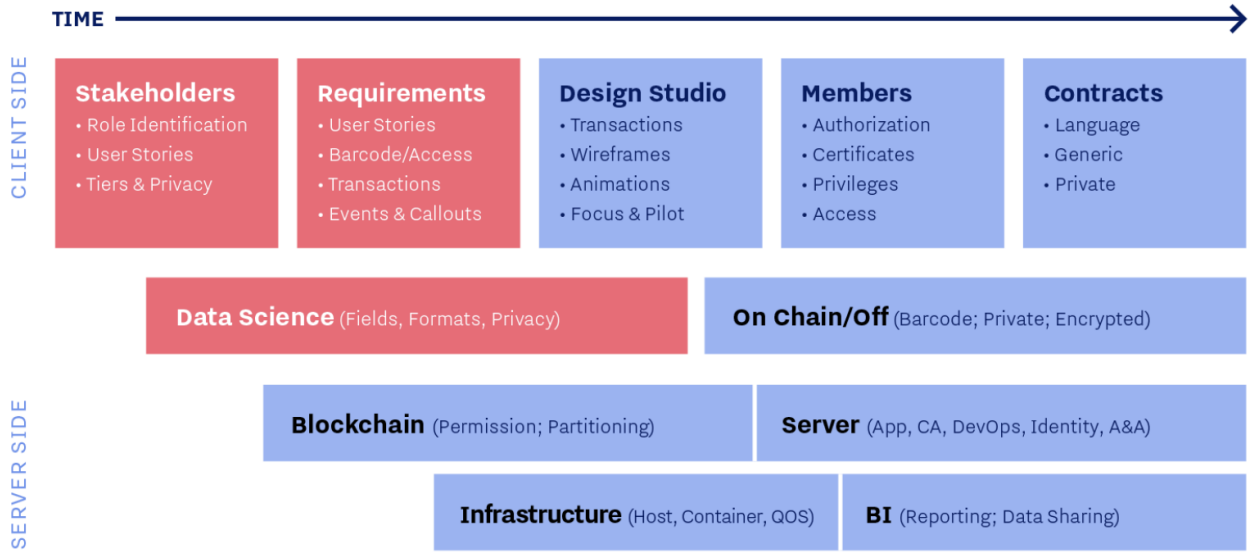


[4] See Appendix A.
[5] This model, as well as several other possible models among peer organizations and/or their members, may be found in Appendix B.
[6] In some respects, it is as if a consignment inventory system were managed by people wearing blindfolds!

As such, a key goal of the CSBWG is to provide a real-time system that allows all stakeholders to interact in a manner appropriate to their privileges. This collaborative system uses blockchain to expose only those transactions for which stakeholder should have visibility.

Blockchain Development Process

As mentioned earlier, the key differences between blockchain systems lies in membership and data management. Governance and workflows are also critical to defining how the system is set up, how it operates, and how users can interact with it. Much of the CSBWG's early work on KitChain involved defining these requirements.



KitChain Pilot

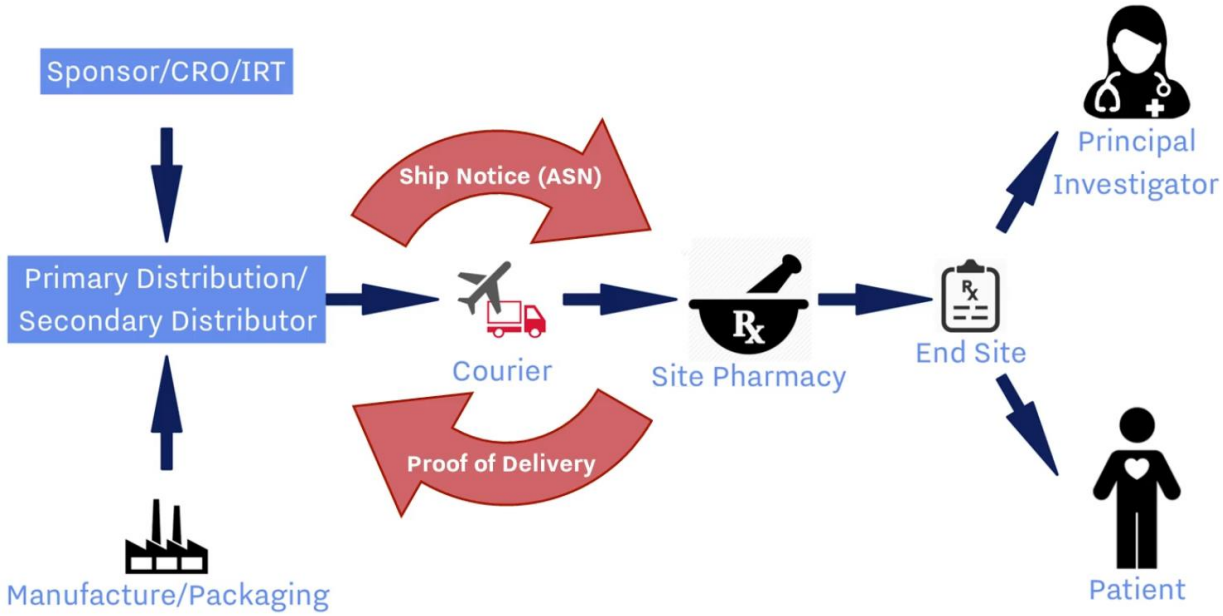
Goals

The CSBWG sponsored a pilot program in which multiple pharmaceutical companies, vendors to the pharmaceutical industry, and study sites explored sharing a common blockchain-based backend and a common iPhone client to track simulated shipments of packaged drug product – termed "kits" – being sent to clinical pharmacies at medical centers in the United States. This blockchain-based pilot application was named "KitChain MVP."

The aim of this pilot is to demonstrate a robust collaborative model for managing the pharmaceutical clinical supply chain, creating an immutable record for shipment and event tracking without the need to resort to paperwork and manual transcription from one system of

record to another. In addition, the secondary goal was to find ways to break the system and expose potential flaws.

To realize this vision, a prototype was created which meets the basic goal of proving out a consortium-sponsored, permissioned, partitioned blockchain-based system for sharing messages concerning clinical supplies.[7] To simplify, the client side was iOS-only (iPhone and iPad) and the data were shared as a single encrypted message off-chain.[8] In addition, smart contracts in the demonstration were skeletal to maximize participation: any CSBWG member could send a shipment message to any other member.



The sender uploads XML and forwards to the recipient for advance notice. Once the kit arrives, the recipient in turn confirms receipt.[9]

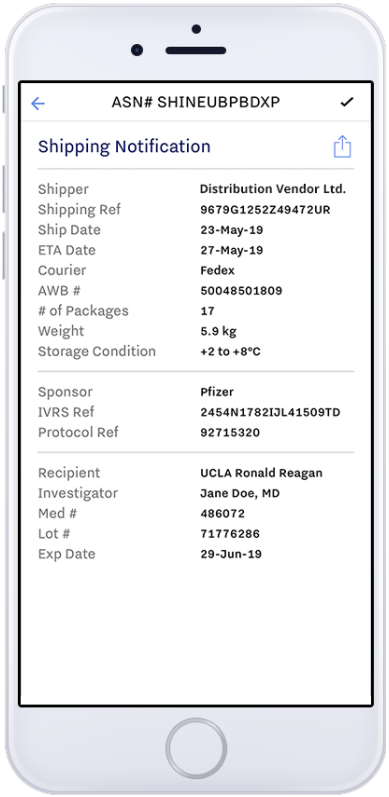
Data Science

GS1, which develops and maintains global standards for efficient business communication, has published a material identification standard for clinical supplies.[10] For the KitChain pilot, the CSBWG determined that an XML-based message would set the stage for machine-to-machine (M2M) messaging.

[7] See Appendix B.
[8] Off-chain: in a separate private database. See Appendix A.
[9] See Appendix E.
[10] GSI, Identification of Investigational Products in Clinical Trials Application Standard. March 2019.

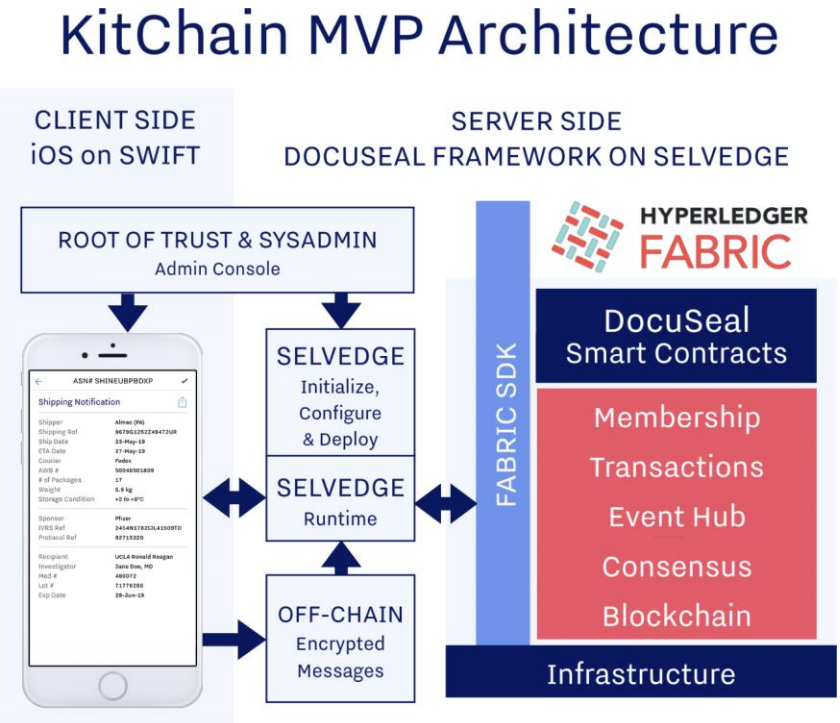
Message

The message was standardized with plausible, but simulated, data in standardized static fields, which may be a subset of future standardized fields that could be complex and dynamic. There was no barcode reading, storing or forwarding in the pilot as the members determined an XML message format suited their needs best at this stage.[11]



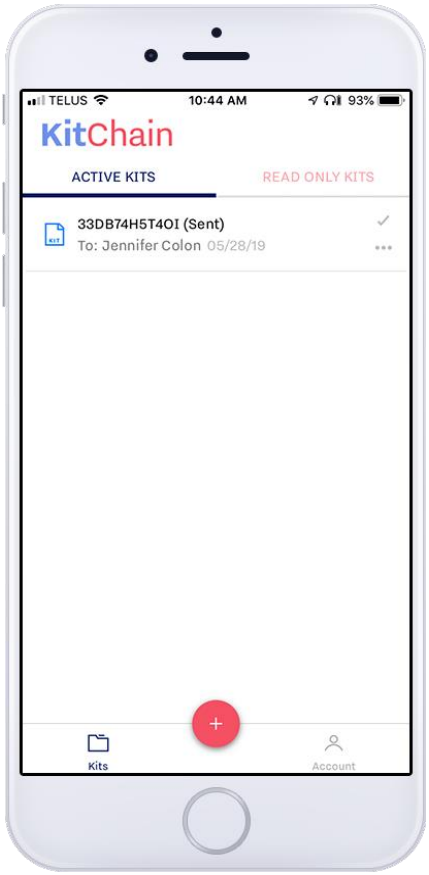
Technology

KitChain has two major components: a frontend mobile application and a backend blockchain server. The backend was implemented in Golang and used open-source Hyperledger Fabric blockchain, the LedgerDomain Selvedge blockchain app platform, and LedgerDomain's DocuSeal framework, encompassing smart contracts and application logic. As such, the pilot has a fully-functioning highly-secure blockchain backend.



[11] See Appendix C for a full breakdown of the message format.

Frontend



The first thing presented to the member is a modal login screen (member email and password). Once logged in, the full app UI is presented.[12]

- **Account actions.** Change first name, last name, and avatar; or logout. Admins have the ability to invite a new member (sending an email that grants the recipient access to the app).
- **Active Kits and Read Only Kits.** *Active Kits* displays shipments that you have sent, as well as kits that other users have shared with you for confirmation. *Read Only Kits* are shipments between two other users where you have been provided with visibility but cannot confirm receipt.
- **Upload.** A button reveals a file list of test kits. (Users can also add their own valid XML messages from iCloud.) Once a kit is selected, the user enters the email address of the recipient, and the shipment is uploaded.
- **View.** Users have the ability to view shipment notifications as well as save them locally. Recipients can confirm receipt of a shipment by tapping a checkmark.
- **More info.** Sender, recipient, date created, and any users with read-only access.

For secure member account registration in the pilot, an admin creates a new member account with email address. An email is sent to the member with a link to the app server and an embedded token. The application itself was distributed through Apple TestFlight, Apple's online service for mobile application installation and testing. (Unlike publishing on the App Store, TestFlight allows applications to be distributed confidentially.)

Backend

An overview of the backend architecture, including considerations related to membership & certificates, data flow, and notarization may be found in Appendix E.

[12] The full workflow is outlined in more detail in Appendix D.

Privacy and Data Retention

For the initial pilot, the KitChain app remained private amongst a few companies. Nonetheless the CSBWG endeavored to follow good practice with respect to privacy and data retention, looking ahead to scalable solutions that will need to have these principles built in. The application gathers personally identifiable information such as names, email addresses, and IP addresses. We performed three cycles of by-invitation only testing and discarded all personal and personally identifiable information after each test cycle.[13]

Critically, the messages themselves were uploaded to a secure message store, to be downloaded by other authorized users. Each message can be authenticated by a unique hash which is immutably stored in the blockchain.[14] However, it is virtually impossible to reconstruct a document based on a hash, meaning that data can be removed but never falsified.

Results of Test

In January 2019, the CSBWG convened to demo the KitChain pilot version 1.0(3) and defined some of the boundary conditions for future development. XML was selected as the target format and Advance Ship Notification (ASN) and Proof-of-Delivery were jointly chosen as the initial use case.

Over the next five months, the pilot was updated with the functionality outlined earlier, with a custom message format, data flow, and sharing functionality. Concurrently, LedgerDomain continued to test the KitChain framework and blockchain server. This allowed for a larger-scale deployment and stress-testing of the smart contracts and application logic, as well as the underlying Selvedge blockchain server.[15] In total, >180 users across 4 servers sent >3,000 messages driving >3,000 notifications during development.

In May 2019, KitChain 1.0(11) was released to the CSBWG as an early test of the new XML format. While there were no crashes, there were issues with the date format and several users had issues with profile pictures.

Later that month, KitChain 1.0(12) was released to the CSBWG for evaluation, and group members were surveyed anonymously. Eighty-three percent of respondents were able to log in

[13] See also KitChain Beta Terms and KitChain Privacy and Data Retention.
[14] Hashing is a method of cryptography that converts data into a string of text. Unlike encryption, which can be reversed with a specific key, hashes are nearly impossible to decrypt. Since a unique piece of data will always produce the same hash, it can be matched against a stored hash. This approach is most often used in saving passwords online.
[15] Critically, while KitChain is constrained to sharing XML messages (typically less than a kilobyte), the testing framework had no such restriction. Users frequently shared high-resolution images and other large files.

with no issues; 50% of respondents desired a clearer set of expectations on when to confirm proof-of-delivery and/or a clearer check for confirmation. ***Overall, on a scale of 1 (missed requirements) to 3 (met requirements) to 5 (exceeded requirements), the application scored 3.8, comfortably surpassing requirements.*** The XML message format and appearance also scored 3.8 on a scale of 1 to 5.

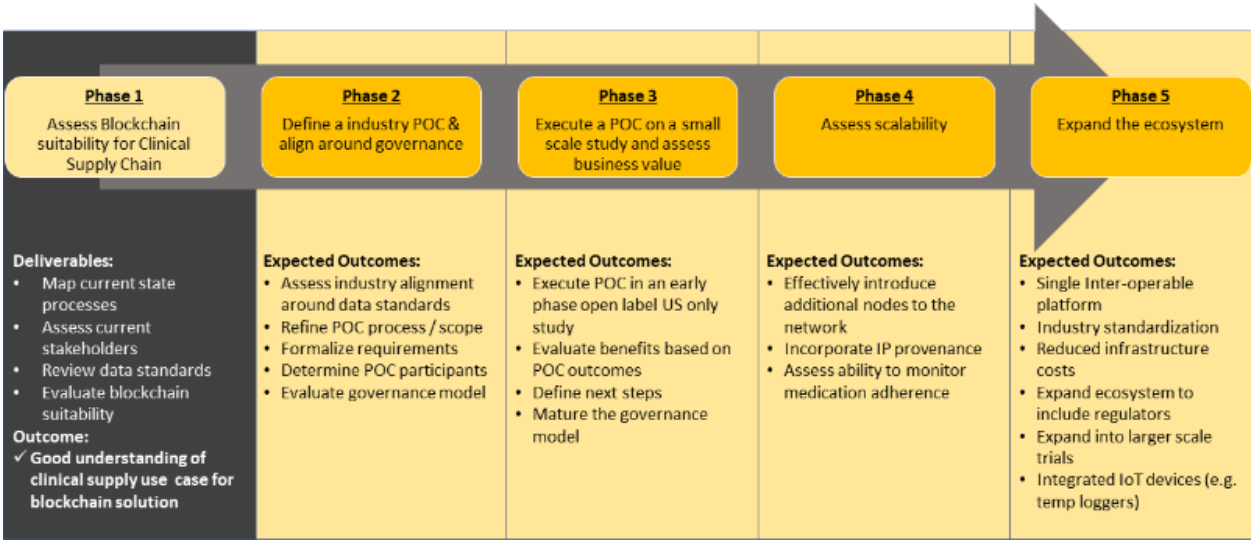
For KitChain versions 1.0(11) and (12), >100 users across 2 servers sent >1,000 messages driving >1,000 notifications during development.

Future Directions

This white paper describes the first two of five phases envisioned by the CSBWG. We formed the group, assessed the suitability of a blockchain-based solution, mapped out the data science, defined the pilot application, and aligned around governance.

The next generation of KitChain will leverage GS1's material identification standards and role-based privileges to source messages from multiple systems and securely route them to the people and systems that need them across the clinical supply ecosystem. A test with real data (run in parallel on a live study) will be the focal point for development in 2019. Based on CSBWG survey responses, we foresee an average of 21 users per organization, with each averaging 7,800 messages per month. Other decisions, such as barcode formats and support for Android and web clients, will be workshopped by the go-forward team.

We anticipate transitioning the CSBWG to a not-for-profit entity to manage the governance, as well as hold the trade name and any other copyrights and IP. It remains to be determined whether that entity will be a new fit-for-purpose entity or part of an existing industry body.



Appendix

A. Essential Terms

Smart Contract

Smart contracts are chunks of code dictating the contractual terms of a transaction upon the blockchain. As they have grown in syntactical power, smart contracts can now mediate a variety of transactional models, including “track and trace” in a supply chain and even personal medical information. Smart contracts model all those dealings; the blockchain both captures and stores them.

Personal medical information is highly regulated (HIPAA in the US, GDPR in the EU, etc.) Blockchain makes possible the sharing of anonymized compliant datasets. While the immutability of blockchain data is difficult to reconcile with the GDPR “right to be forgotten,” KitChain has been designed with these considerations in mind.

Unique Identifiers

Over time, most blockchain applications and systems will interface with real-world objects. For blockchain to perform properly, these objects must have unique identifiers: 2D barcodes, RFID tags, machine-readable labels, etc. Such systems are affordable; general-purpose input devices like iPhones can read them.

Off-Chain

Writing the full message to a blockchain can be computationally expensive (in the case of large datasets) and makes it impossible to delete data that is legally required to be deletable (e.g. in cases where GDPR applies) without deleting the entire blockchain. This can be resolved by off-chain storage, in which the key data fields for a transaction are stored in a further private database.

Under this model, the data is only referenced on the main blockchain through a pointer (e.g. through a cryptographically generated hash). The data cannot be altered without failing a hash check, but it can be deleted to comply with data privacy regulations. The hash itself does not contain any personally identifiable information and can be safely retained in the blockchain.

B. Types of Blockchains

The sponsors or system owners of blockchain-style communal applications might share the following models among peer organizations and/or their members:

1. a data model in which everyone agrees to use the same data fields and formats, yet has their own clients and blockchain backends;
2. client sharing in which all sites use the same client, while blockchain backends differ;
3. fully-shared open blockchain systems with encrypted data (e.g. Bitcoin or Ethereum);
4. fully-shared and permissioned blockchain systems with encrypted data; or
5. fully-shared and permissioned blockchain systems with encrypted data and partitioning.

This yields the following scenarios:

1. In the open blockchain's vanilla form, members can see/decrypt only their own data.
2. In the permissioned blockchain's vanilla form, only permissioned members can see/decrypt their own data.
3. In permissioned and partitioned blockchains, each member may use private channels to transact with trading partners. Likewise, others may possess their own channels.

While all five strategies have merit, the data privacy and security requirements associated with healthcare require a shared-permission partitioned blockchain-based system.

C. Message Format

KitChain messages consist of a 17-field XML format, along with three metadata elements that are appended to the message. Below is an example XML message:

```
<kitchain-message asn="DOLW7C2WRR7A" type="shipment-sent">
    <kitchain-uri>kitchain:DOLW7C2WRR7A/100</kitchain-uri>
    <Shipper>Distribution Vendor Ltd.</Shipper>
    <Shipping-Ref>9895U4673D53004FS</Shipping-Ref>
    <Ship-Date>25-May-19</Ship-Date>
    <ETA-Date>28-May-19</ETA-Date>
    <Courier>UPS</Courier>
    <AWB-Num>72010627456</AWB-Num>
    <Num-Packages>4</Num-Packages>
    <Weight>7.6 kg</Weight>
    <Storage-Cond>-40 to -60°C</Storage-Cond>
    <Sponsor>Pfizer</Sponsor>
    <IVRS-Ref>9035A5381DQP16868OA</IVRS-Ref>
    <Protocol-Ref>55979933</Protocol-Ref>
    <Recipient>UCLA Harbor</Recipient>
    <Investigator>Roger Comstock, MD</Investigator>
    <Med-Num>144476</Med-Num>
    <Lot-Num>31504017</Lot-Num>
    <Exp-Date>20-Jun-19</Exp-Date>
</kitchain-message>
```

Metadata

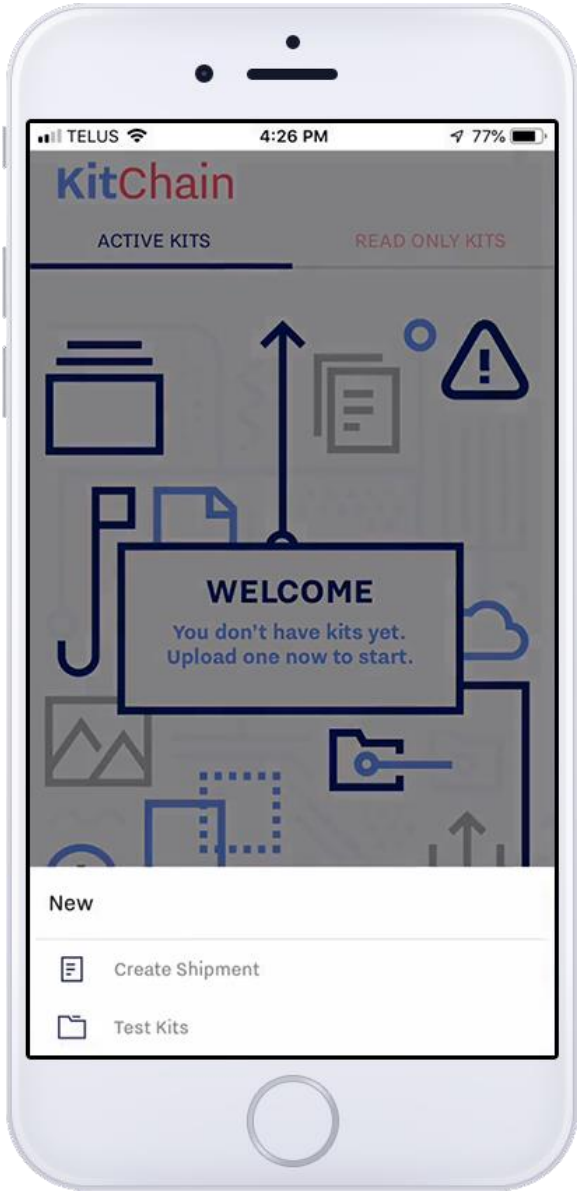
Property	Example	Notes
ASN / Unique ID	7n40075	The advance shipping notice ID.
Sender	Alice Jones	Distribution vendor/CMO contact.
System Timestamp	2019-07-19 03:14:07	

Data

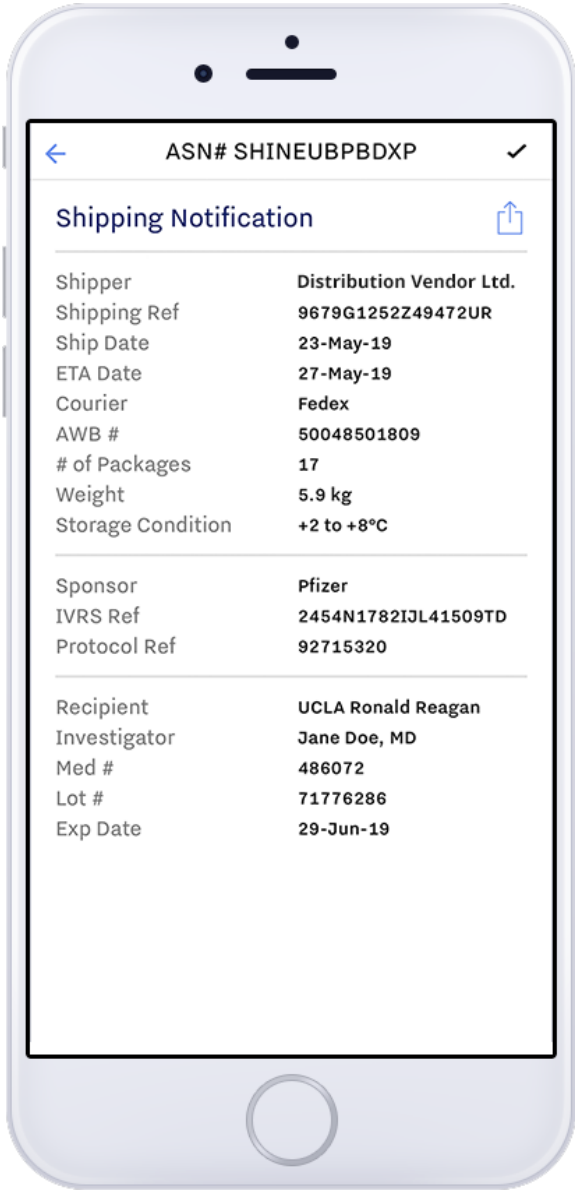
Property	Example	Notes
Shipper	Distribution Vendor Ltd.	Distribution vendor/CMO
Shipping Ref	1VH25TF334F8F46H95	The package reference number provided by the shipper. 18-digit alphanumeric.
Ship Date	07-FEB-19	
ETA Date	08-FEB-19	
Courier	UPS	
AWB #	814726660	Air waybill, 11-digit number.
# Packages	1	
Weight	8.1kg	Unit of measurement: kg
Storage Cond'n	+2 to +8 °C	Multiple temperature ranges: (a) -60 to -85 °C (b) -40 to -60 °C, (c) -15 to -40 °C, (d) +2 to +8 °C, (e) +8 to +15 °C, and (f) +15 to +25 °C, and (g) ambient.
Sponsor	Pfizer	
IVRS Ref	12A3LAD3GKG4155HF TKD	Interactive Voice Response System (IVRS); 20-digit alphanumeric
Protocol Ref	04339922	
Recipient	UCLA	
Investigator	John Taylor, MD	
Med #	107212	Can be multiples.
Lot #	10045968	Can be multiples.
Exp Date	18-MAY-19	Can be multiples.

D. Frontend

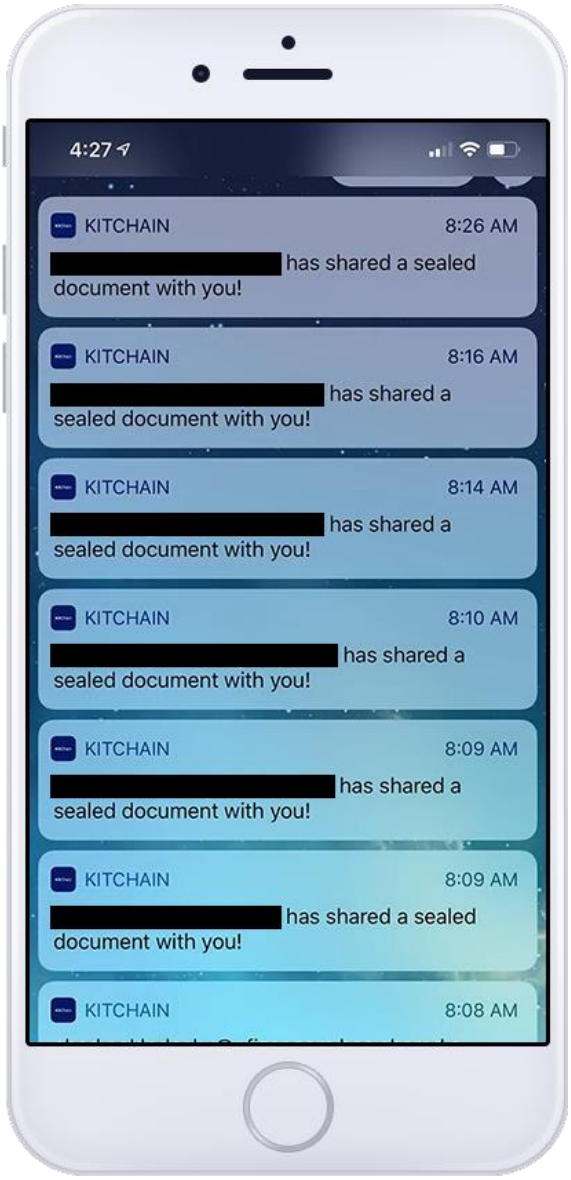
For the pilot program, 540 messages with simulated data were prepared for testers to send. In a real-world application, these XML messages will be created by the shippers, leveraging existing systems. The following exhibits from the KitChain MVP demonstrate the workflow as a shipping notification is uploaded, shared, and confirmed.



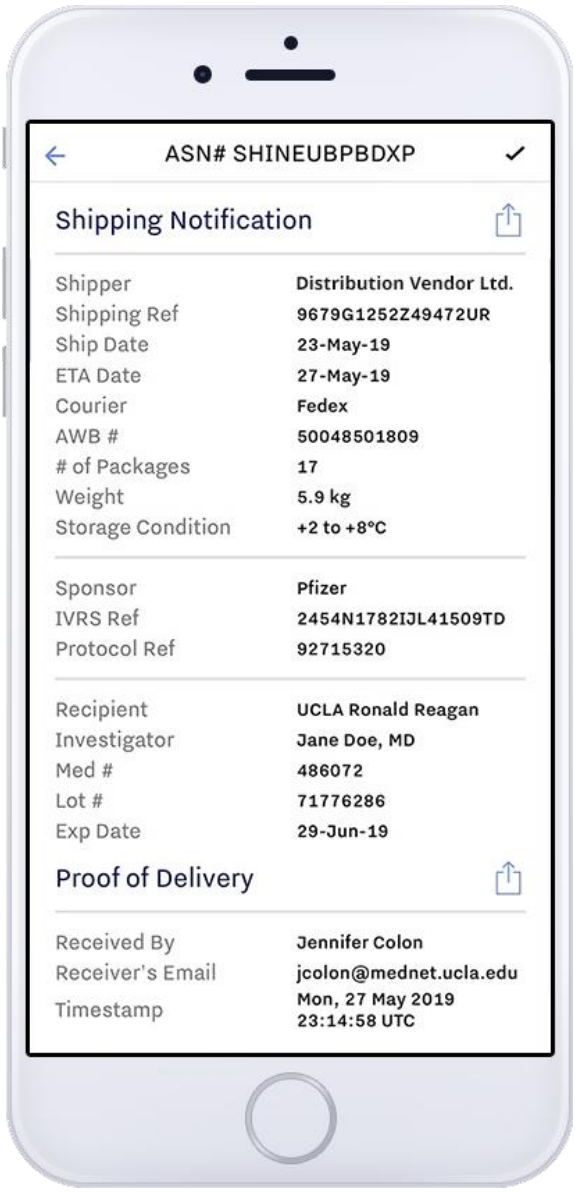
1. *Launching the app, the sender taps a button to create a shipment from one of the test kits. As the message is uploaded to the shipper's assigned lockbox on an encrypted server, a unique hash is generated and sealed into the blockchain.*



2. *The sender is able to download and view the XML message. Each time the message is downloaded, its authenticity is verified against the blockchain. This makes it impossible for the document to be secretly altered or falsified.*



3. *The receiver is notified on their Apple device that a shipment is on its way. They can open the message at any time.*



4. *Once the kit arrives, the recipient can confirm receipt by tapping the checkmark on the upper right. This is written to the blockchain and proof of delivery appears on both users' screens.*

E. Backend Architecture

As mentioned previously, KitChain's two major components are a frontend mobile application and a backend blockchain server. HTTP-based APIs were implemented to handle app members' login into the server on behalf of the app member. These also transact with the KitChain

blockchain network on behalf of the app members, and handles installation and instantiation of chaincode on the blockchain network.

The KitChain backend is built on three distinct layers, which are briefly described below.

Hyperledger Fabric. Fabric is an industry-leading blockchain project created by Hyperledger, an umbrella organization under the aegis of the Linux Foundation with currently over 270 members.. Hyperledger Fabric is the organization's most comprehensive and mature project. Fabric is open source, auditable, and componentized so that any of its eighteen or more elements may be substituted and replaced with other open-source or third-party equivalents. Hyperledger products are licensed under the Apache Software License Version 2.0.[16]

Selvedge. Built to configure and orchestrate Hyperledger Fabric applications, LedgerDomain Selvedge is a blockchain application platform for securely creating multi-party permissioned hybrid-cloud blockchain communities.

DocuSeal. As Hyperledger Fabric is an empty vessel, blockchain sponsors supply chaincode instructions, commonly termed “Smart Contracts,” as well as associated client software. DocuSeal is a LedgerDomain framework for managing documents in a blockchain community, encompassing an SDK, a run-time server and smart contracts.

Membership & Certificates

In blockchain deployments, the topic of membership may well be the most challenging. In an open blockchain such as bitcoin, every wallet is held by the wallet holder. The wallet holder is typically anonymous and the system maintains none of his/her attributes. If a wallet holder is presumed to owe taxes on a transaction, an honor system prevails; if another feels that they have been mistreated, remedies are pursued out of band.

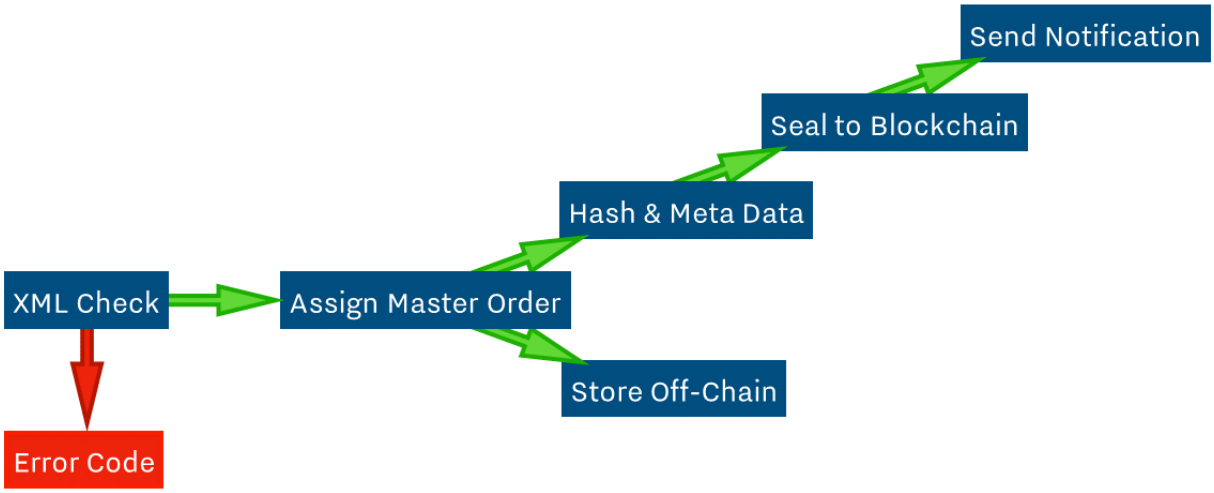
With a permissioned blockchain system such as Hyperledger Fabric, a sponsor could of course replicate the open blockchain experience. He/she could simply invite a large number of random participants, yet the presumption is that sponsors will invite responsible organizations. They will have vetted one another and, in turn, will manage their members’ participation.

As such, Hyperledger Fabric provides fine-grained controls emanating from a root server-- – through an organization to a defined and organizationally backed participant. Mapping and configuring these member relationships is far from trivial; care must be taken to set up members with appropriate privileges. Communities with many members joining and leaving need to resource their dev ops team appropriately.

[16] Elli Androulaki, Artem Barger, Vita Bortnikov, et al., Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains

In the KitChain pilots, CSBWG co-chairs Chad Sklodosky and Imran Shakur served as the administrative "Root of Trust", enrolling only affiliates of member organizations (and of course no one under 13 years of age). The administrator provisions both an iOS frontend and a Hyperledger backend wallet: the enrollee must first log into their iOS app and then "claim" their wallet.

Data Flow



Notarization

Many students of blockchain are aware that in open platforms (like Bitcoin and Ethereum) the so-called "mining" is a critical security layer. The miners' role is to safeguard against double-spending and other malfeasance.

In permissioned blockchain systems, the security envelope is a little different: the permissioning allows for access control, while the Membership Service Provider (MSP) coordinates the certificate cascade. The administrator then works with the sponsor/organization to grant member-appropriate privileges. Notarization is also an important consideration for the sponsor.

Hyperledger Fabric can be run with a variety of different notarization schemes, each with their own costs and benefits. *For security reasons, the CSBWG does not intend to publicly detail any encryption protocols, but new organizations may arrange for a briefing.*

F. Technical Specifications

While not exhaustive, the following captures the core components of the KitChain MVP on final test:

- KitChain Pilot App Version 1.0(12) (LedgerDomain, Swift 4.2)
- Instabug bug tracker
- Branch mobile link service
- OneSignal push notification service
- Mailgun email service
- DocuSeal Framework (LedgerDomain)
- Selvedge Application Server (LedgerDomain)
- Hyperledger Fabric 1.2 (Linux Foundation)
- Hyperledger Private Data Collections (LevelDB)
- Docker
- Amazon EC2
- Amazon Web Services