e-prescriptions using blockchain to detect dangerous drug interactions

Fall 2020, ISMT E-599, Capstone Seminar in Digital Enterprise

Team 4

Christopher Leuer, Jess Beard, Karthick Sundararajan, Per Kreutzfeldt-Jensen

Executive Summary

We are a consortium of Electronic Health Record (EHR) companies that will develop a unified e-prescription blockchain solution to detect dangerous drug interactions. Enterprise transformation is initiated by entities with large market share and impacts providers networks across the U.S.

Business Goals

1. Form a consortium to prevent medication fragmentation across EHR systems and providers. Only a third of hospital systems can exchange and integrate patient health data outside their organization electronically.[1]
2. Perform dangerous drug interaction detection at point of prescription. Adverse Drug Events (ADEs) result in 1.3 million emergency room visits every year.[2]
3. Allow multiple provider networks to participate, where no central authority owns the data, and prescription data is secure and HIPAA compliant.

Solution

A distributed blockchain allows prescriptions To-Be securely shared across entities. When a prescription is submitted to an EHR system, a smart contract event will fire the blockchain. Analytics software will subscribe to the event and perform real-time drug-to-drug analytics to generate an alert to warn the physician of a potential adverse drug interaction.

In addition to the implementation of a static regulatory model for drug interaction, the consortium will enable advanced computational pharmacology machine learning models to integrate with the API. Therefore, providers can select optimal models to expose clinical insights to the physician at the point of prescription. There are off-the-shelf, self-supervised learning models[3][4] available for more than 3,000 FDA approved drugs and more than 110,000 drug-drug combinations.

Technology

1. Develop a B2B solution on blockchain technology to persist, share and secure prescriptions. Implement a blockchain platform, such as HyperLedger, that supports a sufficiently large user-base, permissioned networking, event-subscription, as well as buy-in from a number of leading IT suppliers[5].
2. Develop event-based smart contract APIs to enable real-time predictive analytics.
3. Develop an EHR integrated interface to provide actionable clinical insights.

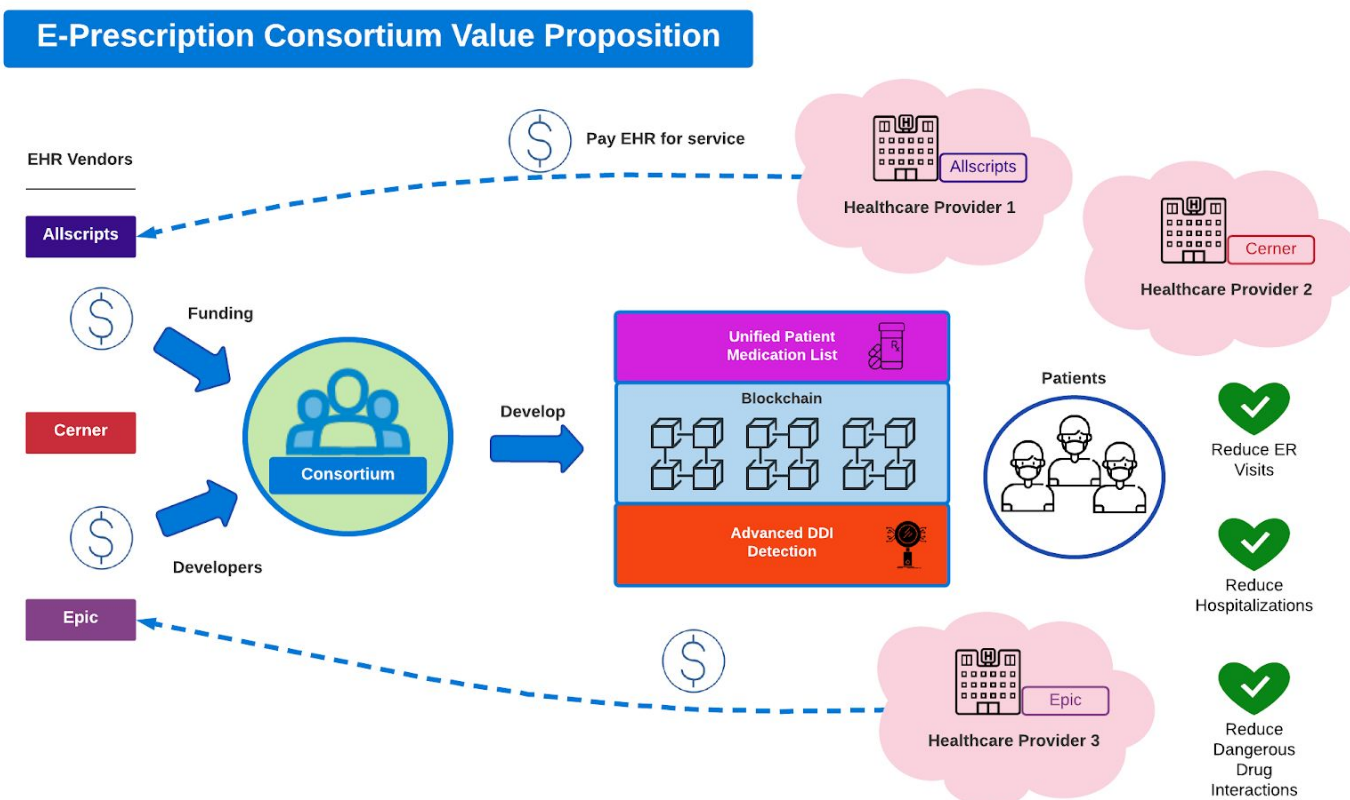
Part 1: Business Requirements

Electronic Health Care (EHR) vendors will be the key investors in and members of the consortium. EHR companies, Cerner, Epic, AllScripts etc. will be responsible for initial R&D. Today, Hospital systems and practices buy EHR software and services from the EHR software vendors. Tomorrow, these vendors will offer an API for drug interaction check as an enhancement of the existing products. So, when the EHR vendors band together to build the platform, they will count on the provider networks to pay for the new API service, ultimately being reimbursed by health insurance companies.

The ultimate winner will be the consumer/patient, receiving better care with fewer Adverse Drug Reactions (ADR) and, healthcare systems with less emergency room (ER) visits and hospitalizations. The advantage of e-prescription blockchain is mainly twofold:

(1) The Consortium will use blockchain technology to create an unified Patient Medication List spanning many Healthcare Providers and EHRs.

(2) The Consortium will develop an advanced deep learning algorithm (DNN) for DDI prediction based on the latest academic research. Optionally healthcare providers can build and integrate their own analytics.



Consortium Definition

Our consortium is funded and managed by participating EHR vendors. The consortium will establish a governing board and appoint and hire a Consortium Director. The consortium will hire dedicated resources (*refer to Part 3 Development and Implementation Roles*). Select healthcare providers will form a user group to provide feedback and help the board prioritize improvements. The consortium will develop the e-Prescription solution according to the consortium road map (*refer to Part 3 Solution Delivery Roadmap)*. In the initial planning phase, the Consortium will enter into an agreement with a single EHR vendor and healthcare provider to develop a Pilot. The Consortium funds and operates major parts of the Hyperledger Fabric infrastructure hosted in AWS Managed Blockchain (*refer to Part 3 Operationalization*). However, participation is truly distributed and healthcare providers can choose to build and operate infrastructure in their own cloud. The Consortium will gather and measure metrics such ER evaluate the success of the project (*refer to Success Measurements and Performance Indicators*).

Business Context

As-Is e-prescription Process Flow

The As-Is e-prescription process flow starts with the patient encounter. The patient meets with their primary physician, is asked to review their current list of medications and if needed, provides updates to current medications. The primary physician will then make manual medication updates to the patient's EHR. At the conclusion of the visit, new prescriptions or modifications to existing prescriptions are entered into the e-prescription software embedded in the EHR. The e-prescription software reconciles the patient's medication list, supports clinical decisions for dosage, routine, strength, and frequency and provides minimal notifications of drug interactions, duplicate prescriptions or patient allergies. Following prescription entry, the e-prescription software then sends the prescription request to the patients pharmacy.

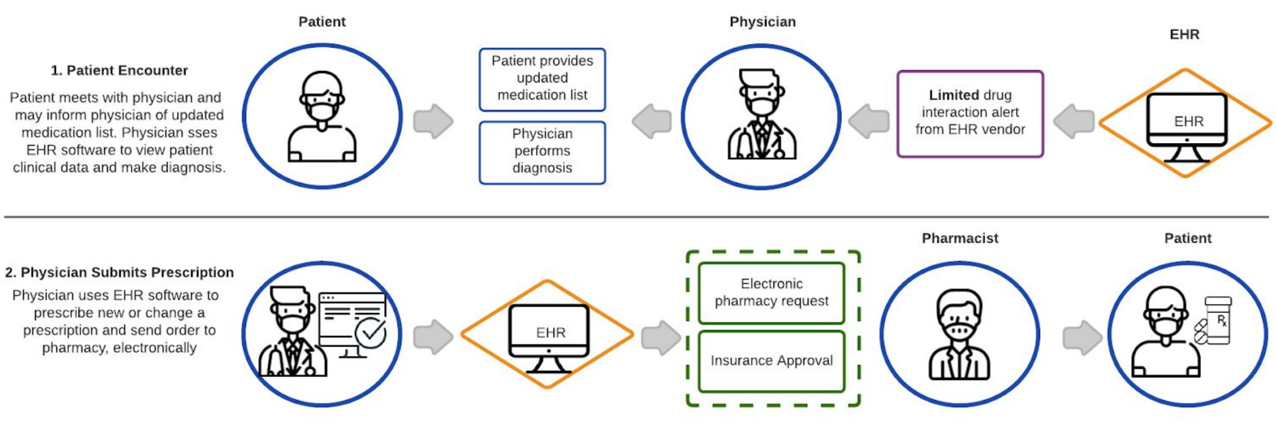


Figure 1.1: As-Is e-prescription Process Flow

To-Be e-prescription Process Flow

The To-Be e-prescription process flow mitigates the issue of disparate EHR system medication lists being available between EHR companies, significantly decreasing the potential for dangerous drug interactions. The To-Be e-prescription process flow starts with the patient encounter. Rather than ask the patient to provide a potentially erroneous update to their medication list, the EHR makes an API call to the blockchain using a smart contract query. The blockchain then returns the updated medication list. At the conclusion of the visit, the primary physician uses the EHR to add a new or modified prescription. The EHR then initiates a smart contract to add the prescription to the blockchain. Next, the analytics app, created to listen for prescription events, initiates drug interaction analysis in real time and then returns the results via a smart contract event. If a dangerous drug interaction is detected, the EHR app will receive an alert notification. Pharmacy and insurance approval workflows do not change.

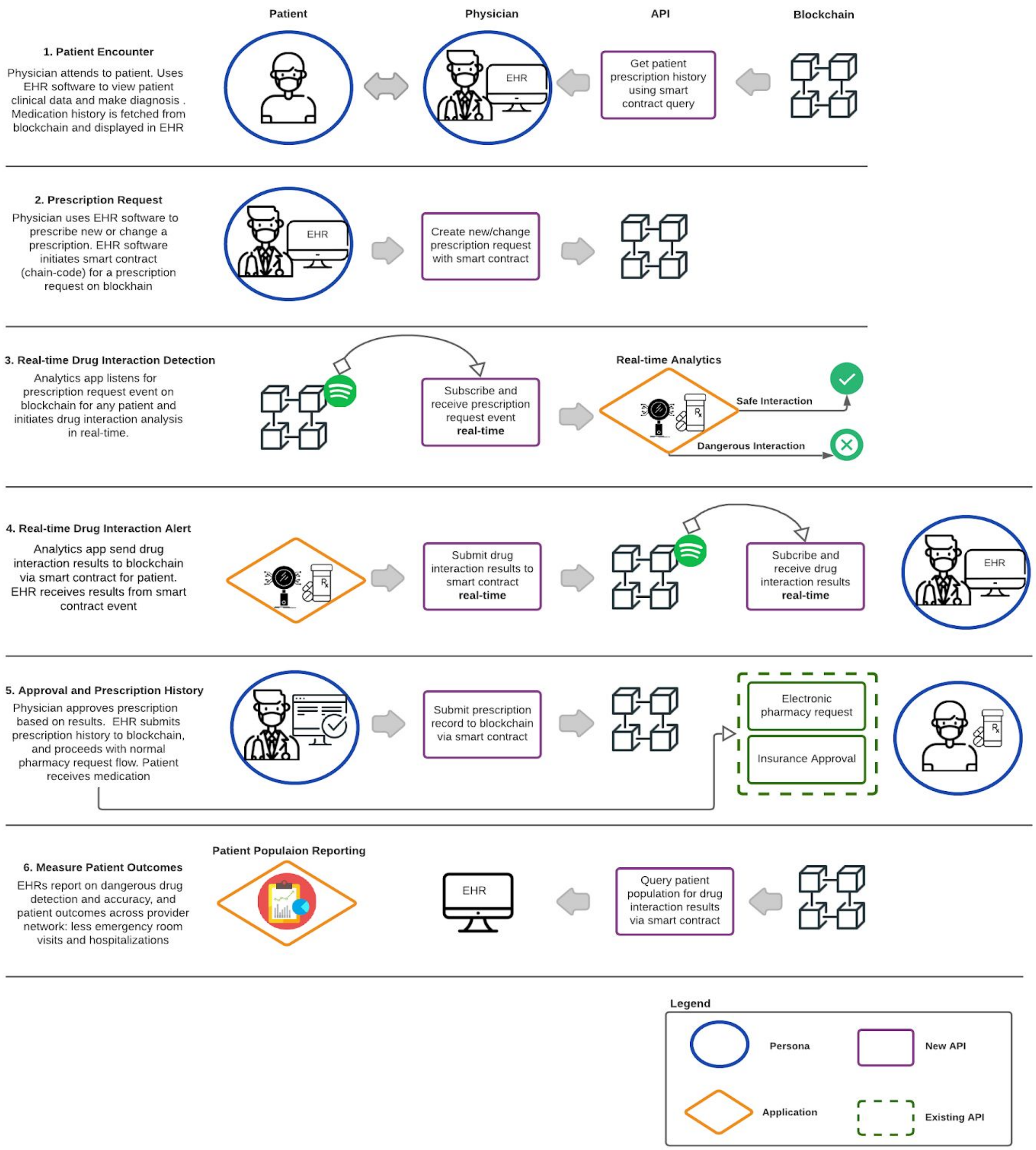


Figure 1.2: To-Be e-prescription Process Flow

Required Functionality

Use Cases

Use Case 1: **Prescribing Physician**

As I prescribe medication, I am very frequently neither able to identify previously administered medication nor place identified medication into a timeframe. An ideal tool would

- *Allow me to look up previously prescribed medication, with dates*
- *Allow me to combine my own medical and pharmaceutical knowledge with references on DDI research*
- *Alert me of dangerous Drug Interactions before a prescription is finalized. The preferred format is*

DDI probability	Drugs involved	Literature Reference
major	Drug pair1	Reference1
moderate	Drug pair2	Reference2
minor	Drug pair3	Reference3

Use Case 2: **Patient**
I'm a diabetes patient. Last year I fell ill while vacationing, and had a GP - who is unfamiliar with my medical history- prescribe a flu-drug to me. Within 10 hours I had a seizure and was admitted to hospital. Unfortunately, I was neither able to detail my previous medication nor when I took it. An ideal tool would

- *Allow me to look up my precise medication history*
- *Allow me to give an ER physician permission to consult my exact medication history*
- *Uphold the right to keep my data private, and share it only with organizations who are involved in my care*

Use Case 3: **ER Physician**
In the Emergency Room, many patients arrive unconscious. Others arrive in a state of confusion or otherwise unable to communicate coherently. A large portion of patients are experiencing drug interactions, for which we have ready-made cures. However, identifying the medication which led to the DDI is detective work. If no obvious vital signs indicate the type of DDI, we often end up resorting to gastric pumping, from which we can identify the drugs involved. Gastric pumping is both time consuming and labor intensive. A time-consuming diagnosis puts the patient at unnecessary risk. Lengthy diagnosis increases the risk of an inpatient visit. An ideal tool would

- *Allow me to consult the patient's medication history*
- *Allow me to consult DDI information instantaneously, in case I am unfamiliar with a medication, or want a second opinion*
- *Identify the patient if he/she carries a patient identifier*

Story 4: "**Allscripts" representative (EHR Vendor)**
Our company funded and backed the Blockchain E-prescription initiative. Want to ascertain that usage and efficacy of the system is as expected. Bigger is better: Not only will industry-wide adoption benefit patients, but via our strategic advantage, it will also increase our market-share and hence our company

bottom line. We believe making Blockchain E-prescription an industry-wide standard will be a financial win-win situation for all parties. An ideal tool would

- *Allow me to consult usage history of the E-prescription system*
- *Allow me to aggregate usage by medic type: ER, other hospital unit, GP, Clinic, etc.*
- *Allow me to estimate number of avoided interventions and their associated cost*
- *Allow me to compare before/after e-prescription outcomes between similar patients*
- *Allow me to determine if DDI patient hospitalization rates decrease after E-prescription system implementation*
- *Allow me to determine if DDI patients can be kept outpatients and/or inpatients without re-admission, thereby simplifying overall treatment*
- *Allow our IT-teams to incorporate state-of-the-art DDI lookups into existing systems*
- Ensure that any output returned use common-place drug-vocabularies (i.e. RxNorm)

Story 5: **Consortium Product Owner**
I was the main architect when setting up the E-subscription Blockchain consortium. As the present product owner, I have a commitment to ensuring smooth operation of the technical infrastructure. We also intend to expand the number of EHR partners over time. An ideal tool would

- Allow the continued evolution of mission and vision statements in collaboration with vendors
- Allow ambitious SLA performance goals (i.e. 3-5 seconds for query/response round-trip)
- Allow patient population overview for individual provider networks, justifying the continued participation in the consortium

Functional Requirements

Id	Name	Requirement	Description
FR1	Lookup	The platform shall allow lookup of a patient's previous medication and the respective prescription dates.	A correct patient identifier will give a table of drug-names and dates.
FR2	Entry	The platform shall allow entry of new prescriptions.	Easy entry of intended new medication.
FR3	DDI Probability	The platform shall be able to fetch and/or calculate the probability of DDIs between any 2 drugs.	If a medically verified DDI probability is available fetch this, otherwise an estimated probability with an appropriate disclaimer attached.
FR4	Literature	The platform shall provide reading resources for any DDI deemed major.	Enable a physician to check appropriate literature.
FR5	Vocabularies	The platform shall use common-place medical vocabularies.	RxNorm for drug-names.
FR6	Usage History	The platform shall allow the identification/affiliation of users and details of user requests to the	The consortium controls access to usage information, which may be handed over to consortium

		consortium only.	members. This will permit consortium members to perform various C/B calculations.
FR7	ID Collation	The platform shall allow identification of provider network clientele to the consortium only.	Only the consortium can collate patient identifiers in order to create population overviews for entire provider networks.

Non-Functional (Technical) Requirements

Id	Name	Category	Description	Example
T1	Modular API	Application	Frameworks allows different provider networks to select and integrate different drug interaction Analytics Apps with same API	Scripps Health organization chooses a static model for drug interaction while Partners HealthCare selects an Machine Learning model. Both providers integrate with same API
T2	Distributed Ownership	Data	No single Entity owns the data. Selected data-layer is fully distributed and shared such that removal of any data node results in no data loss	A patient receives prescriptions from Scripps Health and Partners Healthcare. If Scripps Health managed data-node is unavailable, the patient's complete prescription list unchanged and available.
T3	HIPAA Compliance	Security	Processes and technology must be compliant with HIPAA regulations	An unauthorized vendor cannot obtain access to view a patient's prescription list
T4	Identity authority	Security	An authority (or multiple authorities) are responsible for registering identities and certificates associated with persons and entities, and are trusted by the all participants	A user, owning registration identity private keys, signs a request to register a new restricted identity with attributes for organization, Scripps Health, and a specific role.
T5	Fine-grained read/write privileges	Security	Ability to assign read/write to specific data-sets to users within groups/roles	Only identities associated with physician role within Scripps Health network can read prescription data for patients who receive care from Scripps
T6	Secure patient data in transit	Security	Patient data is secure in transit as routed across Internet and insecure networks	Patient prescriptions as network traffic cannot be intercepted and deciphered as plain-text by an router in provider network
T7	Secure data at rest	Security	Prescription data in persistence layer is secure, and only authorized entities have access	A non-authorized network node cannot obtain an application connection to the data-layer (blockchain)

T8	Real-time event subscription within a network	Application	System must support real-time event notification and subscription to subsets of events. Client must receive notification with hundreds of millisecond of transaction completion for a subgroup of patients	Analytics app subscribes to prescription request events for Scripps Health in real-time and receives a notification in 3 seconds
T9	Near-time notification between different networks	Application	System must support near-time event notification and subscription to subsets of events. Client must receive notification within minutes for physically dnetworks belonging to different organizations	Patient receives prescriptions at Scripps Health in the morning.. Sharp Health nodes receive prescription events within minutes. When the patient arrives one hour later, at Sharp all prescriptions are available..
T10	Data Definitions	Data	Data layer and programming frameworks/languages support types and for prescription data elements, and object representations for prescription requests and medication lists	A prescription request is saved to a data-layer. It will include RxNorm identification of the medication and dosage amounts, and a patient identifier
T11	Large distributed networks	Scalability	Many thousands of application clients on different networks can participate via theInternet. Latency to perform operations remains near constant by horizontally scaling and adding more peer nodes or specialized nodes for synchronization	In maximum terms, all 6k hospitals in the U.S implement e prescriptions, and, for simplicity each hospital implements a single peer node to perform operations to support all EHR clients. Assume 0-20 transactions a second will be achieved
T12	Audit logs	Auditing	Logs associated with operations to create/view/update prescriptions are retrievable for privileged audit users	Scripps Health performs an audit for one its new locations to make sure only non-authorized users are not accessing patient prescriptions
T13	Restrict retrieval of data	Security	Prescription data can only be retrieved and read in plain text by an authorized identity via #4. It not enough security to allow any participant to get access to anonymous medication lists; consortium may need to include PHI in the future for better ML results	Physicians at Scripps Health are prevented from retrieving or saving patient prescriptions for Sharps Health.
T14	Patient identity	Identity	Patient Identifiers in provider EHRs need to link to	Patient x's E-prescription data is stored with unique address y in

	resolution		e-prescription identifiers	data-layer (blockchain) and links to Patient x's EHR identifier.

Business Benefit Justification

While not every Adverse Drug Reaction (ADR) is caused by Drug Interaction (DDI), it is generally acknowledged that preventable DDIs cause a sizable proportion of ADRs in polypharmacy[6].

Statistics

Here are some statistics on the Adverse Drug Reactions (ADR) in the US as per FDA.[6]

- Over 2 Million serious Adverse Drug Reactions documented yearly
- Over 100,000 deaths yearly
- ADRs are the 4th leading cause of death - ahead of pulmonary disease, diabetes, AIDS, pneumonia, accidents, and automobile deaths
- Ambulatory patients ADR rate is unknown
- Nursing Home patients ADR rate is about 350,000 yearly
- Current EHRs fail to detect up to 1 in 3 harmful drug interactions and other medical errors[7][8]

Furthermore, here are some further statistics showing the economic impact of Adverse Drug Reactions in the US as per FDA.[6]

- Estimate of $136 Billion yearly
- Greater than total cost of Cardiovascular and Diabetic care
- ADRs cause 1 out of 5 injuries or deaths per year to the hospitalized patients
- Length of hospital stay, cost and mortality rate for ADR patients are double than the control patients

It is very clear from the statistics above from the FDA publication[6] that Drug-Drug interactions and associated adverse drug reactions represent a significant public health problem that is, for the most part, preventable.

Moreover, a research paper[7] from 2020 shows that there are serious safety vulnerabilities in the current EHR systems. Scientists at the University of Utah Health, Harvard University and Brigham and Women's Hospital in Boston determined that the EHRs consistently failed to detect medication errors that could injure or kill patients more than 30% of the time.[8]

The goal of this paper is to propose a modern way using blockchain, statistical, and machine learning models for dealing with drug prescriptions and prevent Adverse Drug Reactions to a large extent, saving valuable human lives and billions of dollars that could be used for the welfare of the society.

Business Benefits

- Greatly reduced injuries or fatalities due to human errors in prescribing medication
- Improved Patient care
- Rights to Patient data privacy and greater access controls
- Reduced follow-up hospital visits

- Reduced length of hospital stays and thereby cost
- Reduced overall mortality rate due to Adverse Drug Reactions
- Savings of Millions of Dollars

Greatly reduced injuries or fatalities due to human errors in prescribing medication:
There is a 30% chance of EHRs not detecting prescription errors that could cause injury or kill patients. This will be reduced to a large extent. The proposed solution will use both static and advanced machine learning models to detect drug-drug interactions and alert the Physicians and ER staff real time when a prescription decision is made.

Improved Patient care:
Patients have the same access to the Physicians all the time and also benefit from the fact that there are fewer chances of human errors.

Rights to Patient data privacy and greater access controls:
Greater patient data privacy and security by having data in the blockchain and allowing access to only those who need it.

Reduced follow-up hospital visits:
Follow-up hospital visits are expensive for both the patients and the providers. The providers also get penalized for certain codes or cases coming back to the hospital as a follow-up. Reducing Adverse Drug Reactions due to drug interactions will largely reduce follow-up hospital visits by a large part of patients under polypharmacy medication.

Reduced length of hospital stays and thereby cost:
Greater understanding of the drug interaction will allow the physicians to prescribe right medication thereby cutting short the long hospital stays thereby reducing the overall cost and recovery time.

Reduced overall mortality rate due to Adverse Drug Reactions:
Adverse Drug Reactions are a leading cause of mortality and ADRs due to drug-drug interactions would be drastically reduced with the approach specified.

Savings of Millions of Dollars:
This approach saves millions of dollars in healthcare expenses for both patients and the providers by eliminating the preventable problems to a very large extent.

Part 2: Technical Specification and Prototype

The below section describes the technical architecture of the proposed system and the key integration points between the Provider network, EHR, Blockchain and the Analytics Engine.

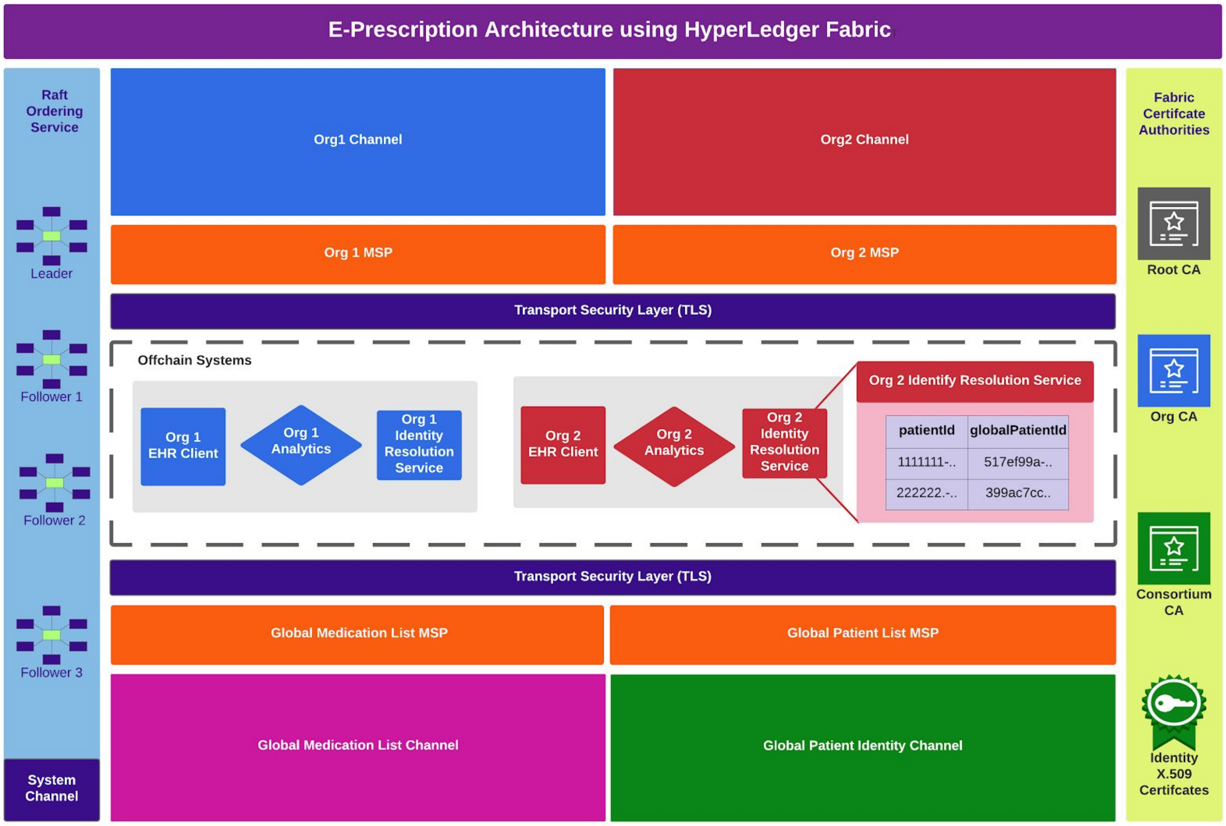
Architectural Approach

The System architecture is designed with blockchain technology to enable decentralized coordination between the various participants (Healthcare providers). It enables a global patient medication list to span multiple networks and should result in better patient outcomes from improved Dangerous Drug Interaction detection within the disparate EHR systems. The participating provider networks can be a national network or a subgroup within the same network. Various participating providers run the same code that updates the blockchain. The participating providers run their own Analytics engine and may choose to use the static, default models or may build and run their own machine learning models to detect dangerous drug-drug interactions.

System Architecture with HyperLedger Fabric

Why HyperLedger Fabric

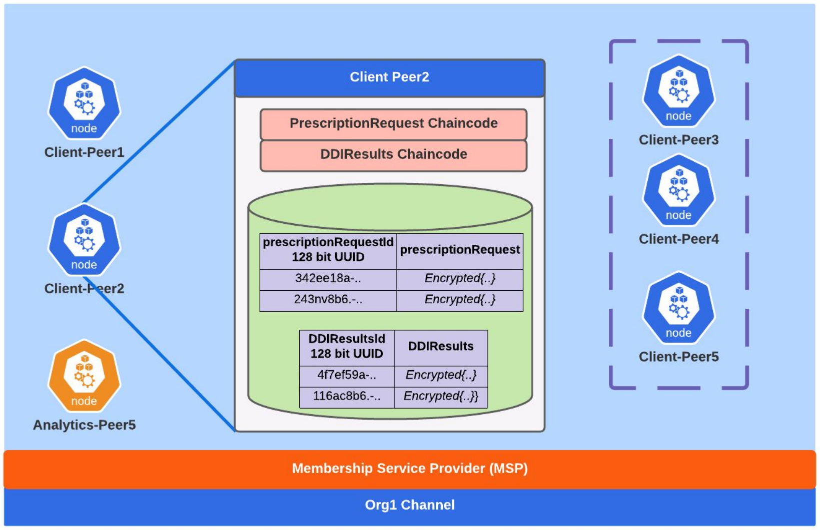
Fabric is an open-source Linux Foundation project to create permissioned blockchains and distributed applications on a modular architecture which supports general purpose programming languages. The Consortium selected HyperLedger Fabric as its platform because security is a core part of the architecture and it is one the most widely adopted enterprise-oriented blockchain projects. Fabric provides the features necessary to authorize participation in the network, and implement security policies for data access and confidentiality for an e-prescription solution.



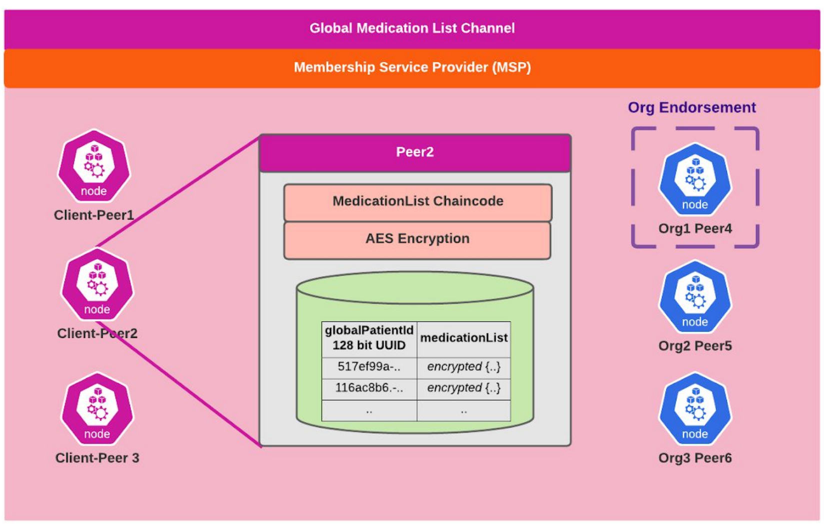
Fabric Channels

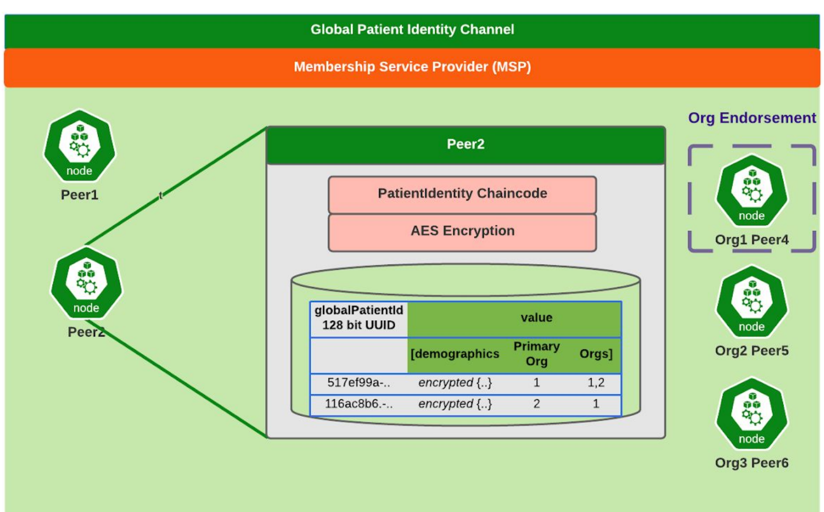
- Channels are created for each organization to isolate PrescriptionRequest and DDIResults from other organizations
- Two global channels are created for MedicationList and PatientIdentity so that patient records are globally shared among all organizations
- Channels are isolated ledgers with their own world state stored in database
- Each channel has a channel Member Service Provider (MSP) to restrict participation to organization identities for clients and peers
- Each channel peer will also have a Local MSP per Fabric requirements as a configured part of the node's physical file system
- Private channels for Orgs are preferred to a centralized database because:
  - Organizations themselves may be decentralized and sub-divided i.e. large medical groups with private practices and hospital systems, yet a patient receives care from all facilities (even though business model and IT structure is different)
  - If a single database system is used management of Identities and authorization to perform analytics is no longer controlled by Certificate Authority and Membership Service Provider

Organization Channel



Global Channels





Fabric Identities

- Consortium will introduce Root, intermediate X.509 certificates
- Each organization will create and manage certificates for channel identities using Fabric CA servers
- Identities will have standard roles to constrain behavior in network:
    - **client-peer**: Includes global or chaincode so that applications can interface with blockchain and propose transactions.
    - **peer**: peer endorses transactions within the channel for an organization. Includes a ledger and world state of channel
    - **admin**: Deploys and configures blockchain
    - **orderer**: Verifies and orders endorsed transactions, then distributes to transactions to peers
- Fabric CA admin will register and enroll identities to prevent sharing of private key
    - Admins register identities and generate enrollId and secret
    - Identities such client-peers, client analytics, client EHR services can use enrollId and secret to generate public/private keys using Fabric CA SDK
- Identity X.509 certificates are added to MSPs to permit participation in channels

Fabric Organizations

- An organization represents a provider network
- Each organization belongs to a channel for PrescriptionRequest and DDIResults
- Each organization deploys peers to the Global Medication List and Global Patient Identity channels for endorsement
- For larger provider networks a organizations can be subdivided into Fabric Organization Units i.e. org1.hospital1

Fabric Consensus

- Use majority peer endorsement policy for organization channels
    - Majority endorsement is the default. It make it unlikely a consensus can be corrupted because a majority of the nodes are participating
- Use custom peer endorsement policy for updating medication list
    - Members of same organization as creator must endorse transaction
    - Note: chaincode prevents non-authorized organization from updating list
    - Ensures unauthorized organization cannot update medication list for a patient
- Use the Crash Fault Tolerant Raft for Ordering
    - Infrastructure and cloud agnostic, and is scalable across multiple TCP/IP networks
    - Uses leader-follower model to prevent single node or network failure
    - No requirement to perform strict time-series transaction-creation ordering or custom ordering to generate ledger blocks. Default Raft consensus according to arrival of transactions is sufficient

Fabric Security and HIPAA Compliance

Our E-Prescription blockchain solution uses Fabric to (1) create trusted identities for Medical Group organizations, clients (analytics and EHR clients) and blockchain components such as ordering nodes (2) define roles and permissions associated with identities (3) divide transactions into separate ledgers using channels for each Organization (4) restrict access to discrete subsets of data, and (5) encrypt data following AES encryption scheme (described in software section).

In broad terms HIPAA compliance prohibits the release of personal health information (PHI) to unauthorized organizations and users.   In the scope of our solution, this means no application, logical organization or user that is public and does not have participation authority via the Fabric CA authority and Member Service Provider (MSP) on either organization channels and global channels can view or access PHI.   PHI data is also encrypted with strong AES symmetric encryption within each object row in the Fabric peer database (default LevelDB). Only authorized identities associated with a patient's organization(s) can get access to data in plain-text. Furthermore, there are no plans to anonymize PHI data for research or general public use.

Peers

- Client peers are introduced to each organization channel with chaincode to both:
    - (1) create and listen for PrescriptionRequest
    - (2) create and listen for DDIResults
- Client peers are introduced to global MedicationList channel with chaincode to get and persist MedicationList
- Client peers are introduced to global PatientIdentity channel with chaincode to match and persist Patient Identities
- Organization peers are introduced to global MedicationList channel to endorse transactions

Fabric Network Scalability

- The use of one channel per organization logically limits the number of nodes needed for majority endorsement of PrescriptionRequest and DDIResult transaction
- The use of one channel per organization logically limits the number of PrescriptionRequest events the Analytics client needs to listen to and process
    - Alternatively, without channels the network would become unsustainable with respect to majority endorsement and event processing
- We expect even the largest network provider will only need only a handful of nodes to run all blockchain
- Network performance of MedicationList and PatientIdentity chaincode should scale to patient size of hundreds of millions rows. LevelDb scales well as a key/object store. MediationLists and Patient
- Early Research on Fabric 1.x demonstrated slow execution times for when transactions reached ten thousand of transactions on more than 12 nodes[2]. Fabric 2x and RAFT consensus is designed to scale better but academic research is limited and we should perform our own performance tests to establish benchmarks for network size, transaction limits and latency.
- Scalability should not be a concern in the private organization channel because only a handful of nodes will be required to run all the client chaincode for a single organization

Software Solution

Build vs Buy

The Consortium will build and maintain the infrastructure and the product, create Global Medication List and Global Patient Identification Channels. Also, the consortium will create onboarding processes for any new provider network (organization) to join the consortium network. The Consortium will also create basic

Drug-Drug Interaction Machine Learning models that can be used by the participating Organizations by default if they choose not to have their own implementation. The Consortium will also create a default Analytics Engine and chaincode logic to Listen/Emit entries to the Blockchain which will be shared across the participating organizations for them to deploy and use in their private channels.
There are no out-of-the-box solutions in the market that can do all of these and so, building the software solution using existing and proven technologies is the most feasible way for this use case.

Tech stack

Blockchain: Hyperledger Fabric
Infrastructure: We will select AWS (Amazon managed Blockchain with Hyperledger Fabric) for initial consortium development. However, there is no cloud vendor "lock-in" as Fabric 2.x can be run with Raft ordering service across multiple clouds. We are designing a blockchain solution that can run in any cloud as Medical Network Providers run IT infrastructure on private clouds, on-premise data-centers and public clouds such as AWS. Medical groups are constrained by different budgetary, privacy-rule and regulatory requirements from adopting a single cloud provider. Our initial intention is to implement the two global channels in AWS but allow network TLS termination outside the AWS network to allow groups to deploy their nodes in approved infrastructure.
Virtualization and containerization: Docker
O/S: AWS Linux or CoreOS Linux to run Docker containers
Database (world state): LevelDB NoSQL Document-oriented database
Data exchange: JSON for format . RESTFul for API
Network Communication: TLS and use Fabric generated certificates
Wallet for Identities: Local filesystems for physical nodes and CouchDB Wallet store for logical entities such as organizations, applications and admins
Machine Learning:
- Data ETL: Python
- Create and Train a Sequential Neural Network Model: Python, TensorFlow keras
- Exposing the model as an API: Python/Django or Python/Flask

Analytics Engine:
- Programming (for Listener/Emitter): Node.js/Java

Individual Org Clients:
- Programming (for Listener/Emitter): Node.js/Java

Infrastructure/Deployment

In Hyperledger Fabric, smart contracts are deployed in packages referred to as chaincode. Organizations that want to validate transactions or query the ledger need to install a chaincode on their peers. After a chaincode has been installed on the peers joined to a channel, channel members can deploy the chaincode to the channel and use the smart contracts in the chaincode to create or update assets on the channel ledger.

A chaincode is deployed to a channel using a process known as the Fabric chaincode lifecycle. The Fabric chaincode lifecycle allows multiple organizations to agree how a chaincode will be operated before it can be used to create transactions. The chaincode can be deployed as containers that can be shared with other participant organizations on the same network.

Reliability

As a blockchain technology Hyperledger Fabric distributes state across peer nodes in key/value databases. Each node has a full copy of the world state and ledger of the channel. The Fabric ledger is a chain of all facts leading to current state and cryptographically guaranteed to be immutable and unaltered on every node. The loss of any node from a Fabric channel will not result in data loss or failure of system integrity.

The Raft ordering service is Non-Byzatine fault tolerant. It cannot protect against a majority attack by compromised nodes, but this is unnecessary because our Fabric is a permissioned network. Rather, Raft is crash fault tolerant and it's distributed leader-follower model protects against orderer node O/S crashes, network failures, and network delays[8]

High Availability

Fabric chaincode can run 24/7 in Docker containers distributed across the E-Prescription blockchain in many channels including the global Patient Identity and Medication List channels. To achieve redundancy in case of system failure, containers running nodes can be duplicated many times and run in different networks. Cloud containerized deployments, such as AWS promise high availability, replicating multiple copies of data within an AWS Availability Zone (AZ) as well as across 3 AZs in an AWS region, without any additional setup.

Security

As for the security of the participating organizations in the Consortium, the Consortium creates root and admin public/private keys using the Fabric Certification Authority. Channels are implemented for Prescription Requests and DDIResults for each organization such that no organization has access to analytics inputs/outputs of another organization. To prevent unauthorized access to Medication Lists or Patient demographics identities for EHR clients, analytics are defined and created in the Membership Service Providers (MSPs) of the global channels. Only these identities holding authorized X.509 digital certificates are permitted to interact and perform operations on the chaincode (smart contracts) for Medication Lists and patient demographics. Fabric includes a registration and enrollment API to allow identities to be registered and enrolled using a secret; the EHR client generates a private key to store on its filesystem but does not need to share with the admin.

Furthermore authorization to communicate with chaincode running on Node.js servers is restricted by OAUTH 2.0. While X.509 certificates permit a piece of software to request a medication list or submit an approved prescription, an OAUTH 2.0 token is required to allow an individual prescriber to execute the chaincode. It is unrealistic to generate X.509 digital certifications for every prescribing physician or nurse practitioner.

Encryption Scheme for Patient Data on Fabric

- Use symmetric AES 256 bit encryption to encrypt patient demographics and medication lists
    - Two organizations with authorized for same patient must be able to decrypt medication list
- To prevent leakage AES symmetric keys will be encrypted in offchain private collection of channel databases per organization
    - AES Keys used to encrypt medication list for patient are considered ultra-sensitive and must be protected
    - Every patient belong to primary organization and each Organization gets its own identity enrollment secret
- A CouchDB Wallet will be used to store the enrollment secret passwords

    - Secondary organizations are associated with each patient in the Global Patient Identity channel. Therefore, chaincode will lookup the primary organization and get primary enrollment secret (if not feasible we will store Org secrets in private collections and protect with chaincode that checks org privs)
- Chaincode permits a channel peer to retrieve cipher-text AES key from private collection by organization and passphrase.
    - F encrypt (org secret passphrase, plain-text AES key) → encrypted key by organization
    - F decrypt (org secret passphrase) -> plain-text AES key

Cloud Scalability and Performance

The E-Prescription Hyperledger Fabric blockchain network is designed to be distributed across any infrastructure including private clouds or public clouds, running as managed services such as AWS Managed Blockchain or with Kubernetes clusters running Microsoft Azure Kubernetes Service (AKS). For example, Amazon's managed blockchain is scalable to allow growth of applications, data and network usage over time. When a participating organization requires additional capacity for creating and validating transactions, they can efficiently add a new peer node using Managed Blockchain's APIs with a variety of options to configure around the CPU and memory of the nodes.

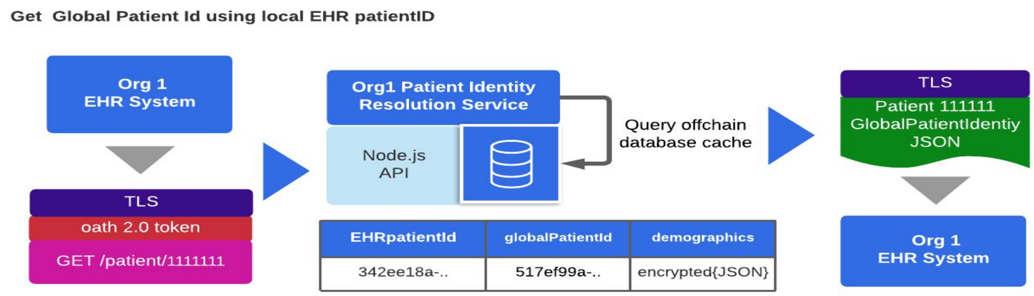
Integration with Other Applications and Dataflow

Patient Identity Resolution Service (Offchain)

- Enables EHR clients to get globalPatientId for use with MedicationList chaincode
- Acts as a cache of rows mapping EHR patient to globalPatientId
- Acts as proxy for EHR to get global patient identifier (globalPatientId) from blockchain
    - GET /patient/match/patientId
- EHR bulk loads demographics in batches and perform probabilistic matches against global patient identities
    - Matches are loaded into the patient identity service database using existing globalPatientId
    - New patients are loaded into with new globalPatientId using generated 128-bit UUID
    - Chaincode API is available on Global Patient Identity service
- Optionally, an Interface can be created to resolve partial probabilistic matches
- Is configured with relational database which acts as cache that can be bulk reloaded
- Service methods are stateless and, therefore, service instances can be replicated
- Should be secured via Fabric certificates for Service → Identity Chaincode
- Should be secured with TLS and app authentication for EHR clients → Service

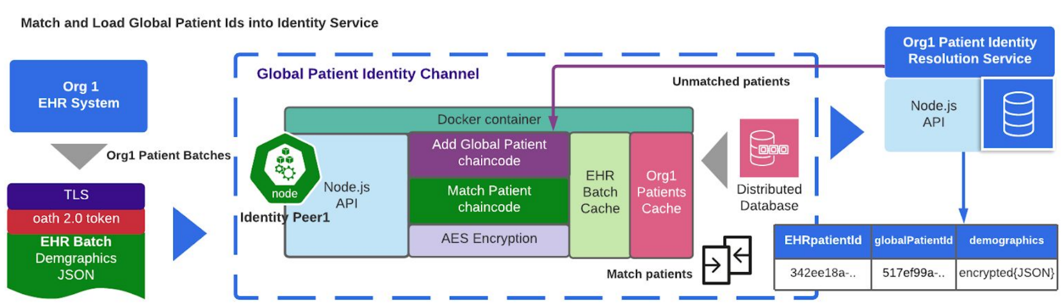
Get Global Patient Id

Org1 EHR System securely submits an patient identity request using the local EHR patientId to an off-chain Patient Identity RESTful Service. The service API verifies the security token associated with the EHR user. The Patient Identity service keeps a persistent cache of maps of local EHR patientIds to globalPatientIds. It performs a lookup and returns GlobalPatientIdentity JSON to the EHR System.



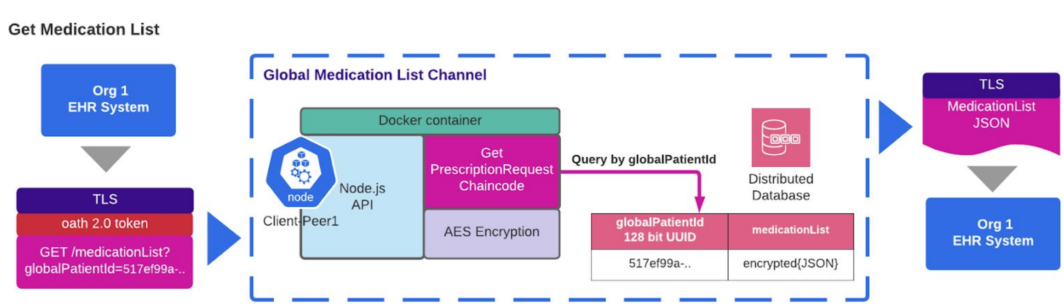
Match and Load Global Patient Ids into Identity Service

Org1 EHR System securely submits batches of patient identities as JSON to a client-peer Node.js API in the Global Patient Identity Channel. The Node.js API verifies the security token associated with the EHR user. The peer chaincode then performs probabilistic patient matches between the EHR and blockchain identities based on demographics. Matches are inserted into the Patient Identity database cache with the matching globalPatientId. 128-bit UUID globalPatientIds are generated for unmatched patients (blockchain data-flow to submit new identities not included) and persisted to the database cache. Once all batches are loaded EHR clients can query the Patient Identity service to obtain globalPatientIds.



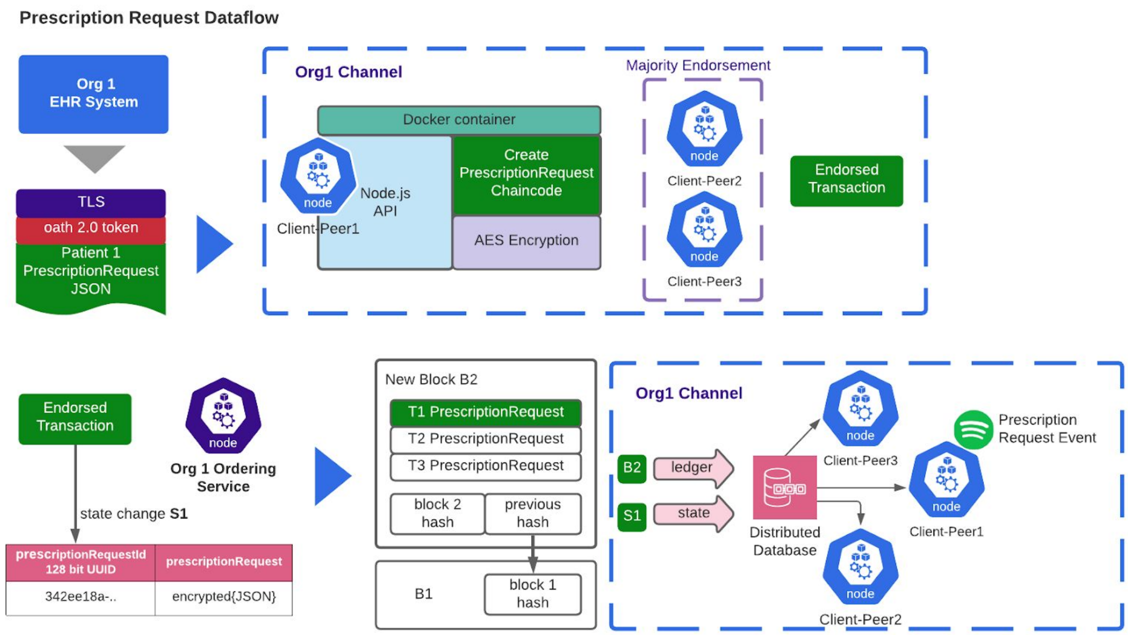
Get Medication List

Org1 EHR System securely submits an MedicationList request by globalPatientId to a client-peer Node.js API in the Global Medication List Channel. The Node.js API verifies the security token associated with the EHR user. As an Org 1 identity, the peer uses chaincode to query the MedicationList by globalPatientId and return the MedicationList JSON to the EHR System.



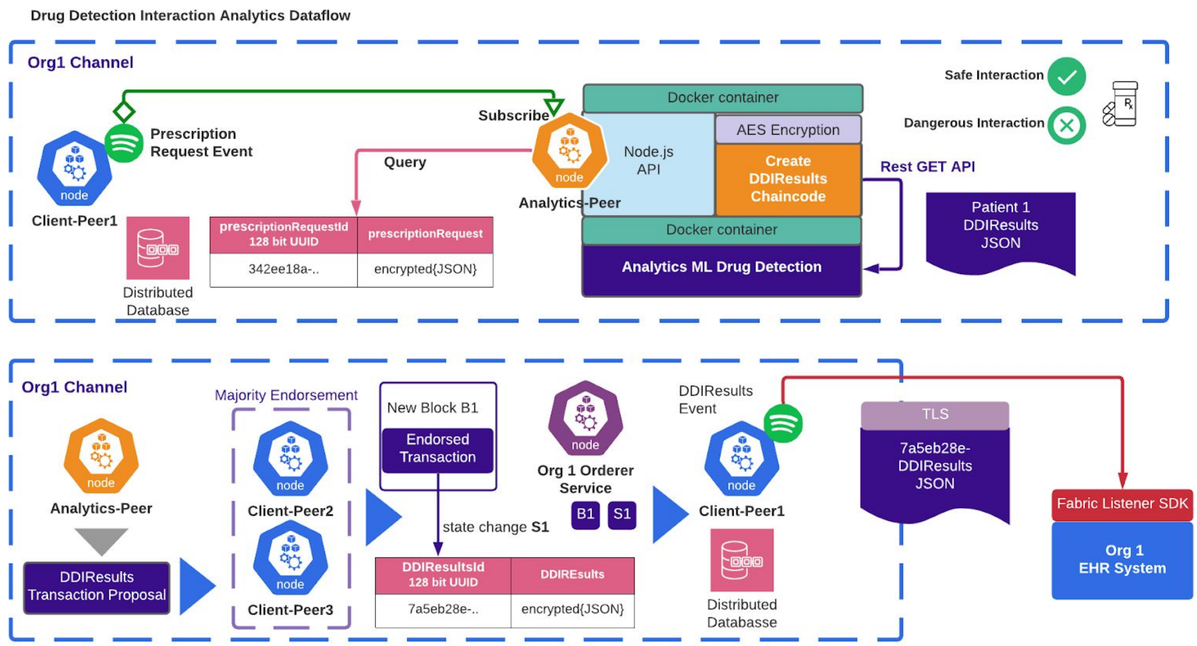
Prescription Request

Org1 EHR System securely submits a PrescriptionRequest as JSON to a client-peer Node.js API in the Org1 Channel. The Node.js API verifies the security token associated with the EHR user. In Org1 channel, Fabric Client peer chaincode creates a proposed transaction with new prescriptionId and it is endorsed by a majority of peers. The RAFT ordering service verifies the validity of the transaction and orders it in a block. All peers within the Org1 channel apply the new block to their ledger and change the world state for the PrescriptionRequest in their own database. A PrescriptionEvent fires on the Org1 channel.



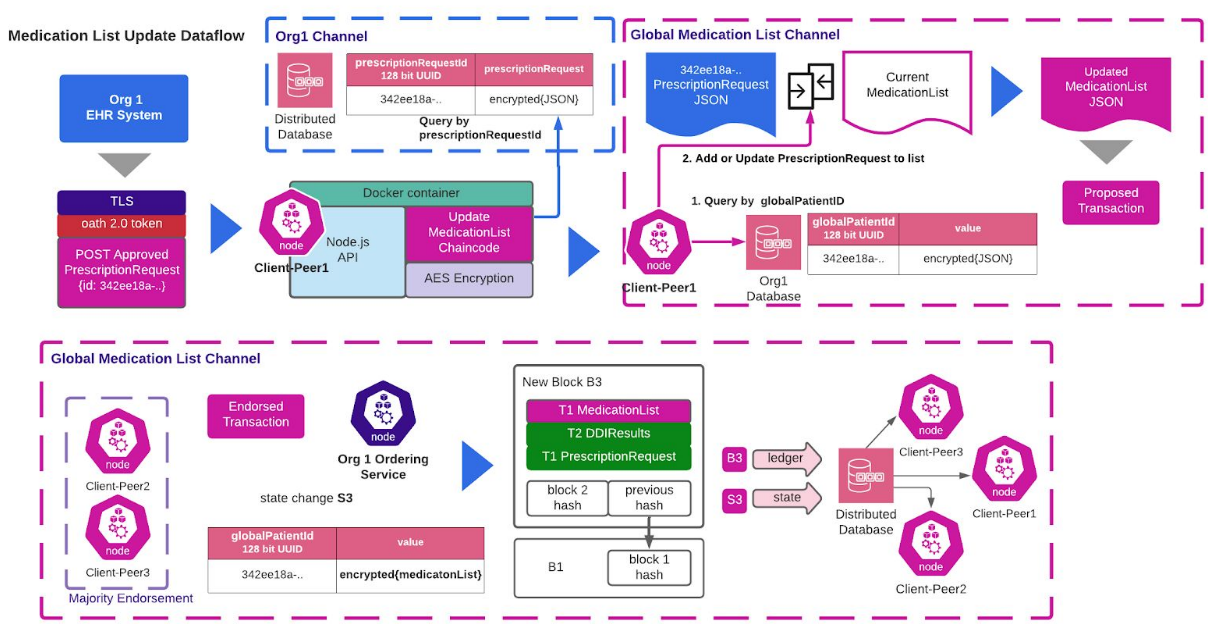
Real-time Dangerous Drug Interaction Analytics

In Org1 channel, a single node Analytics peer listens for the PrescriptionRequest event. In chaincode the peer submits an Dangerous Drug Interaction API request to another container running the Analytics model. The model evaluates safe or dangerous interactions between drugs and returns a DDIResults JSON payload to the peer. The Analytics peer then submits a DDIResults transaction and it is endorsed, ordered and distributed to all peers on the channel. A DDIResults fires on the Org1 channel, and the Org1 EHR uses a Fabric SDK to listen for the specific PrescriptionRequestId to obtain the DDIResults JSON and display alerts in the UI.



Approve Prescription and Update Medication List

An Org1 EHR System securely submits an approved PrescriptionRequestId to a client-peer Node.js API on the Global Medication List Channel. The Node.js API verifies the security token associated with the EHR user. As an Org 1 identity, the peer (joined to two channels) uses chaincode to query the PrescriptionRequest by Id in the Org 1 channel, and query the MedicationList by globalPatientId. The peer chaincode parses the MedicationList and determines if an update or insert of a new medication prescription will be performed. A new MedicationList JSON is formed into a proposed transaction. The peer then submits a MedicationList transaction and it is endorsed, ordered and distributed to all peers on the Global Medication List Channel.



Potential Analytics models

Many different machine learning models are available as methods for DDI prediction. As not all consortium members will want to use the same model, here we will describe 3 Machine Learning models along with their respective pros and cons. It is up to each consortium member to decide which method fits its requirements better.

The methods commonly used for DDI detection and risk assessment are:

1. Support Vector Machine (SVM) algorithm

SVM is a supervised classification algorithm[18]. In simple terms, classification is done using pre-annotated supervised data (aka “training data”) and placing each data-point in a coordinate system such that each pre-annotated category has the “maximum margin” to other categories. For our concrete Drug Interaction problem, that means receiving a number of patient records that were annotated by a physician as either “DDI” or “No DDI”. This training data is then run through the SVM algorithm, resulting in a predictor. The predictor can be applied to new, real-world patients, predicting whether these patients run the risk of experiencing a drug interaction. SVM has been used for a variety of tasks such as webpage-classification, image-classification, optical handwriting recognition, and classification of protein-molecules[18]. SVMs were used repeatedly in academic literature for DDI prediction in the mid-2000s[17]. There are a number of pros and cons associated with SVMs. Pro is that the method is mathematically well-known and a mainstay of machine learning. Cons are that training data is manually annotated and that prediction results vary greatly depending on the amount and quality of data the SVM is fed. In actual fact, most predictions are in the grey-zone; not yielding a clear yes/no answer to the DDI-problem at hand. As a result, SVMs remained in the academic domain, and have not yet given rise to a dominant industry-accepted solution.

2. Disproportionality methods

Somewhere on the continuum between standard statistical methods and machine learning lies disproportionality methods. These methods work by extracting key figures about a large body of data and classifying new data as being below or above the key baseline figures by a reasonable confidence interval. The typical PV data used is the FDA public domain adverse event database, AERS[12]. This database contains information of all ADEs deemed serious (i.e. life-threatening, hospitalization required, chronic condition) by the US FDA[14]. All pharmaceutical companies present on US soil are obliged to submit anonymized ADE information in order to maintain a commercial license in the country. Based on millions of cases of ADEs, a disproportionality method will make the bold assumption that non-DDI ADEs occur at a fixed rate per medication received. Since DDI ADEs occur multiplicatively, they are significantly above non-DDI rates. This simple rule tends to hold true for many real-life cases. Literature frequently shows predictions to be better than SVMs. However, there are noteworthy cons to using disproportionality methods. These methods do not take real-world medical assessment into consideration and are thus likely to be discarded by the medical community. Furthermore, prediction depends on figures extracted from the complete background database, meaning that frequent updates to the key baseline figures are necessary to keep classification scientifically valid.

3. **Deep Learning Algorithms** (Deep NNs aka DNNs)

A recent addition to the DDI prediction toolkit is Deep Learning[13]. Behind the catchy name lies an invention from the last century: Artificial Neural Networks. Modeled on the neural network (NN) architecture of the human brain, NNs are essentially multiple self-adapting pieces of calculus (layers) which combine to approximate an optimization problem. NNs are ubiquitous in today’s digital landscape.

They are used for purposes as diverse as x-ray interpretation, speech recognition, image motive recognition, recommendation systems, and credit fraud detection. NNs excel at suggesting valid predictions from very large datasets, predictions which experts regard as “intuitive” and difficult to formulate. In our concrete DDI example, the chemical structure of medication along with drug-name and a randomly selected training set (pre-annotated list of sampled accepted DDIs from the recognized DrugBank dataset(13)) were chosen as inputs(13). The resulting predictor was applied to the remainder of the (unseen) DrugBank dataset, yielding correct predictions of 92.4% of the interactions observed. A noteworthy pro of DNN algorithms is therefore the prediction accuracy. Another pro is the specific usage of low-level chemical data as the basis of prediction. The con of this type of algorithm is that it demands very large amounts of input data of which initial training data must be humanly annotated.

Analytics recommendation

The consortium recommends the usage of a deep learning algorithm (DNN) for DDI prediction whenever possible. It should be noted that DNNs require a very large amount of hand-annotated DDIs for initial algorithm training as well as frequent periodic re-fitting. Consortium members who are not able to provide these DNN inputs would likely do better with SVM or disproportionality methods.

Analytics Data sources

In the following we will specify the data which is input and output from the consortium’s recommended DNN analytics method.
Input data consists of 2 datasets: the Drugbank DDI dataset and the chemical structure dataset.
The DrugBank DDI dataset consists of 1.3 million DDIs covering all US FDA and Health Canada approved drugs[11]. These DDIs were gathered from official package inserts and academic literature. For every drug-pair, a severity category (minor, moderate, major) is given. The categorization can be defined as follows:

DDI Severity Categorization

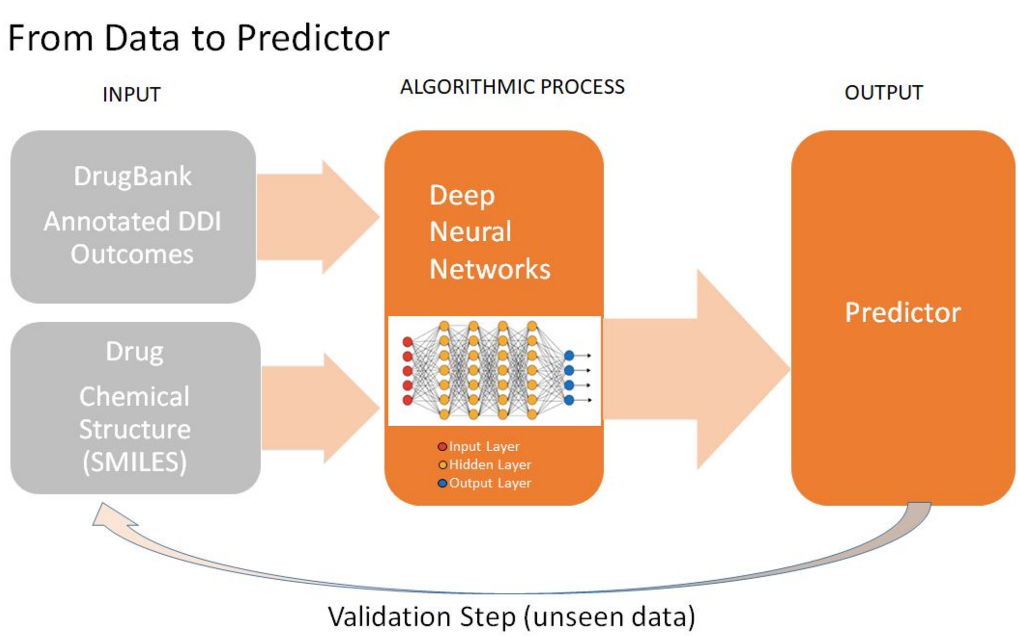
Severity	**Risk**	**Description**
minor	Observe and adjust	These medications may interact in a clinically significant manner, however, the benefits of concomitant use usually outweigh any risks. A monitoring plan should be put in place and dosage adjustments may be needed.
moderate	Adjustment should be considered	The benefits of continuing concomitant therapy should be evaluated on an individual basis. Actions such as aggressive observation, dosage changes, or alternative medications may need to be taken.
major	Combination should be avoided	Concomitant therapy with this combination may cause more harm than benefit and alternative medications or means of therapy should be employed.

In addition to the drug-pair categorization, a rule for body system pharmacokinetics was incorporated. Different drugs will have different rates of leaving an organism through excretion and other metabolic processes. Drugs which occur at chronologically very different periods of time should therefore not show up as alerts. Our model uses pharmacokinetic techniques albeit in a very simplified version: drugs prescribed longer than 12 months from today will be counted as minor risk DDIs as the likelihood that a person still taking a drug long after prescription is very low .

The second input dataset consists of drug chemical structures. These chemical structures are ideal for deep learning methods. The input format is known as SMILES, a 1-dimensional version of the 2d molecular diagrams (aka "structured formula diagrams") commonly used in chemistry. Transformed to SMILES format, the common painkiller Ibuprofen can be conveniently represented as CC(C)CC1=CC=C(C=C1)C(C)C(O)=O. Although SMILES-formatted molecules are nothing but a very long string of atoms and their connections, with no absolute demarcations, a deep learning algorithm will be able to chain together pieces of the sequence and point them out as important for the DDI prediction and hence categorization. Deep learning algorithms automate two parts needed for the final predictive abilities. Firstly DNNs point out which features (parts of the data) are predictive (leaving out features without predictive qualities). This process is known as feature extraction. Secondly, the DNN will be able to place input data (patient DDIs) into categories ["major","moderate","minor"]. This process is known as classification.

ML model information - from data to predictor

A complete DNN workflow can be depicted as follows:

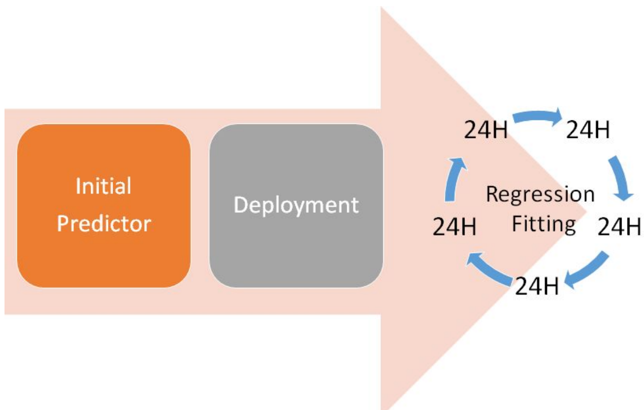


The process consists of a training part and a validation part. Training is done on a subset of the input data only, validation is done on the remainder of the data, often called "unseen data". This process is followed to avoid statistical overfitting.

Annotated DDI incomes and drug protein sequences are input. DNNs are allowed to perform iterative optimization. DNNs output a so-called “predictor”, a binary object which describes how to classify DDI combinations. This binary object will be deployed to a backend-database, as part of the consortium API which any consortium member can integrate into their EHR product.

Overall Analytics Workflow

The full workflow includes deployment and regression fitting.
Generating the initial predictor is not the end of the workflow. The predictor is deployed to a backend database from where the consortium API fetches data for the DDI alerts. Furthermore, the predictor will need to be updated for new data, aka “regression fitted”. Our data provider DrugBank adds new data every 24 hours, therefore a regression fitting and subsequent deployment will have to be updated in parallel.



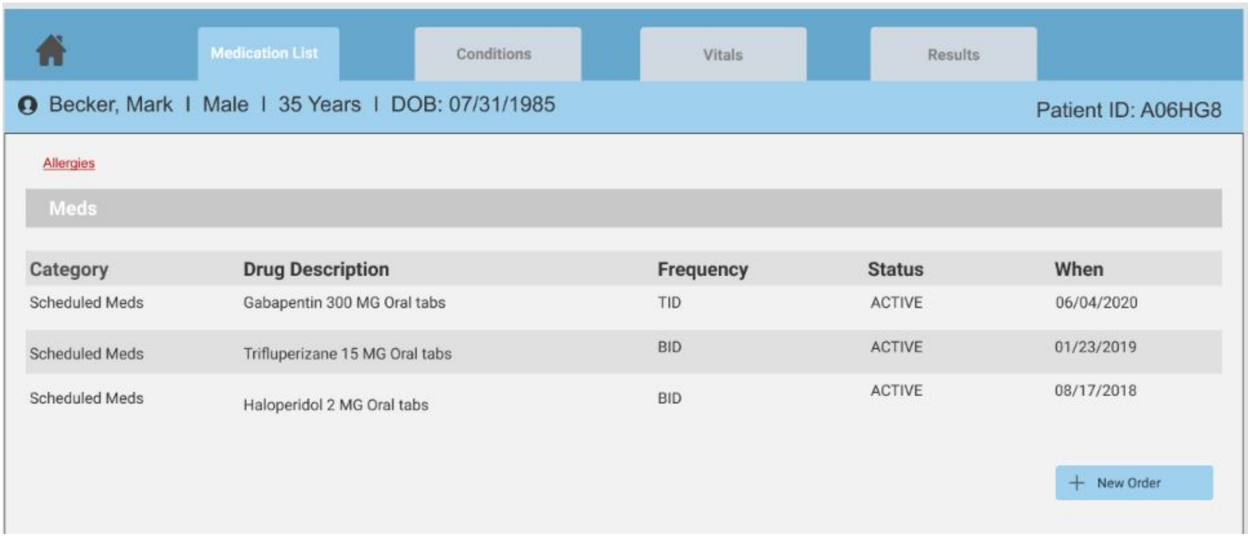
Analytics deployment

Deployment is done via JSON responses. A consortium member EHR suite will be able to have all relevant information returned via markup using just a global patient ID and a prescription request ID. With that markup, any GUI containing relevant key information as well as further input options can be created. Appendix 1, DDIResults section, details JSON structure needed for DDI Machine Learning results to be transported between system components.

User Interface Prototypes, GUI and walkthrough

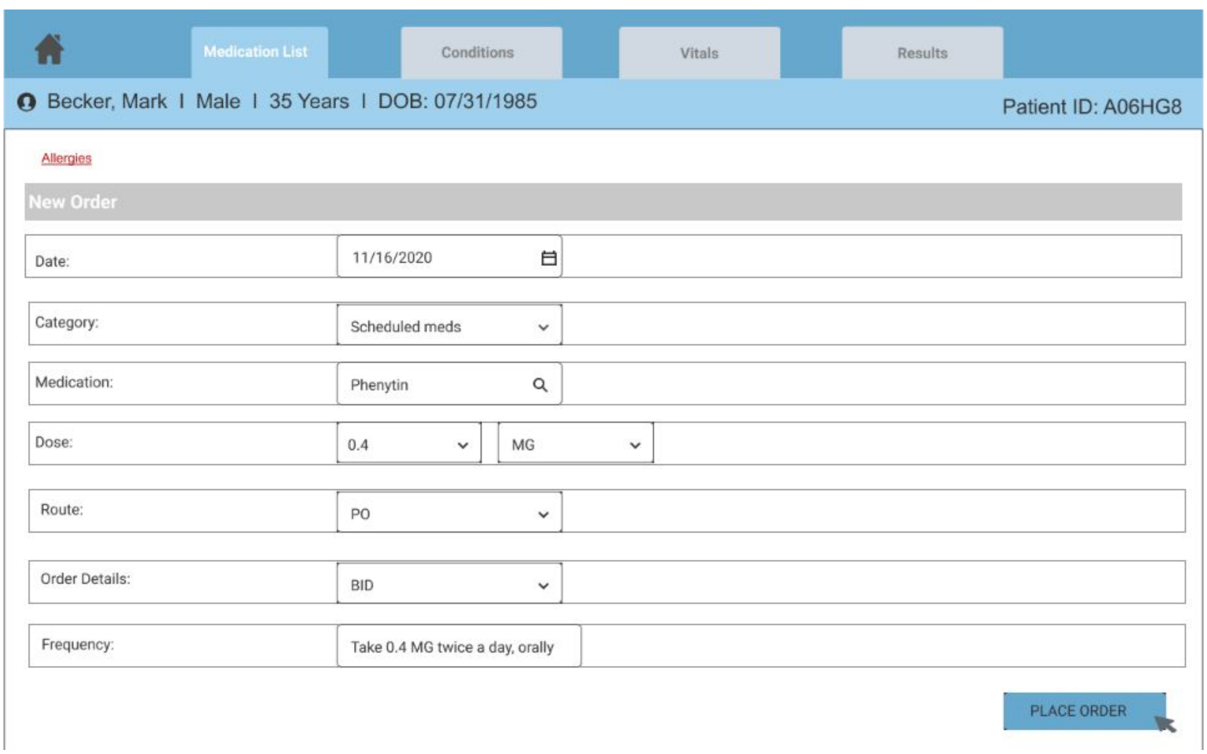
<u>Patient Medication List</u>

During the patient encounter, the physician is able to call the blockchain to receive an updated medication list. At the conclusion of the visit, the physician wants to order a new prescription.



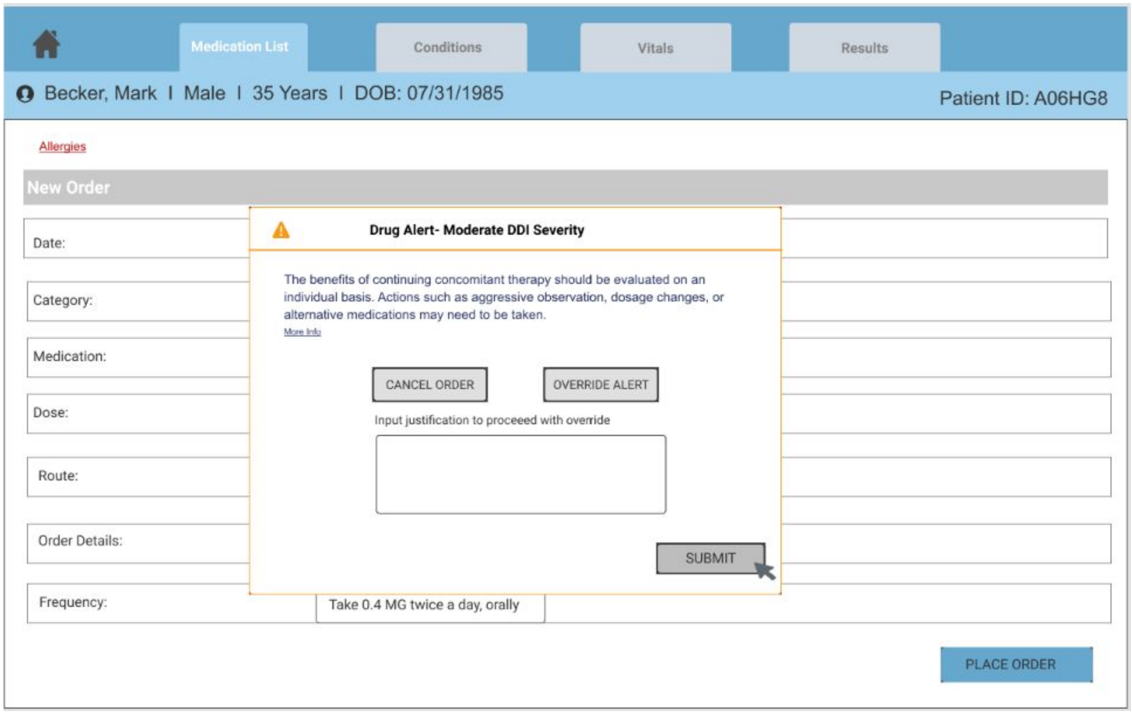
Input New Prescription Order

The physician inputs the new prescription, route and dosage details and then selects order. Upon selecting place order, the algorithm analyzes the potential prescription against the patient's existing prescriptions and a list of known drug-drug interactions (DDIs).



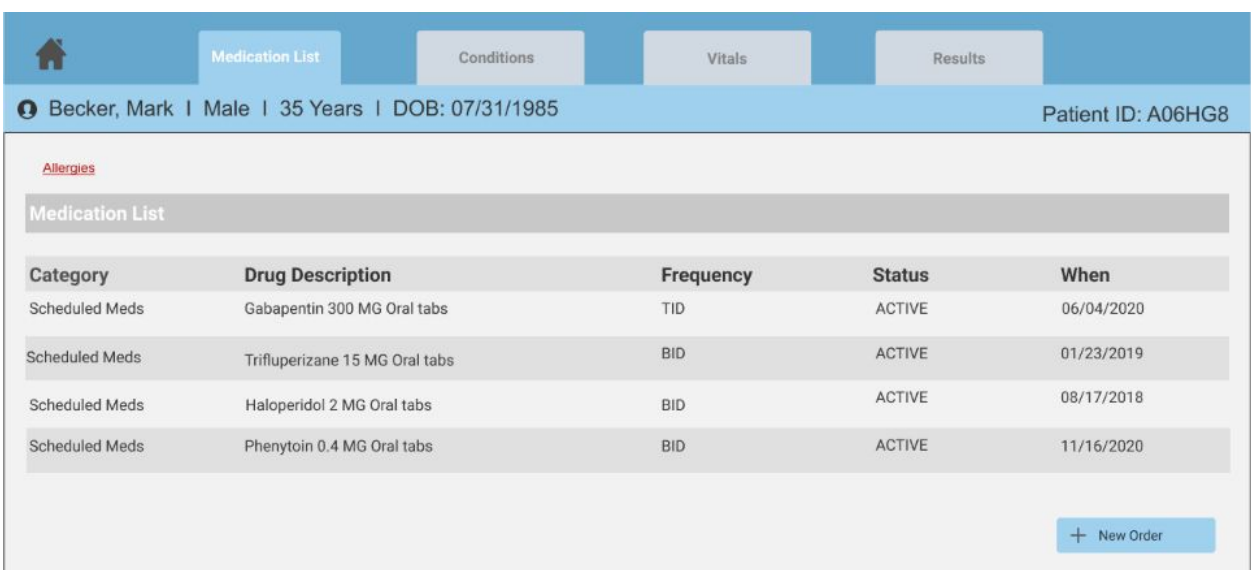
DDI Alert

The analytics model has detected a moderate potential DDI indicating that an adjustment should be considered. The physician receives an interruptive alert and is given the choices of cancelling the new prescription or overriding the alert, in which case the physician will need to provide justification for the override that is used for documentation purposes within the patient's chart.



New Prescription added to Blockchain

The physician understands the risks, adds a justification and submits the order which pushes the new prescription to the blockchain and is added to the patient's medication list.



Part 3: Implementation Plan

The below section describes the implementation plan of the proposed system.

Development Methodology

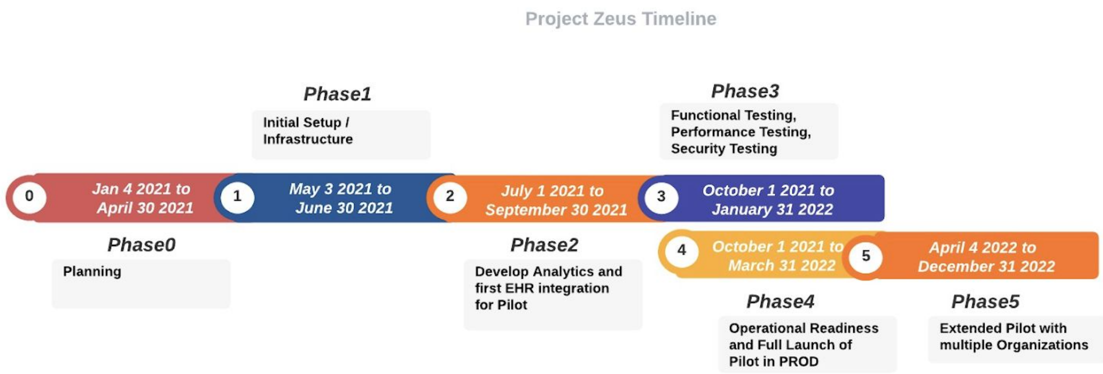
The consortium of EHR vendors will form and fund a development team to build the blockchain solution on the Hyperledger Fabric framework (see consortium member roles). The team will only use Agile methodology in phase 1 the build cycle. This will allow the team to give demonstrations to the consortium board on a two week cycle and get regular feedback. All other phases follow waterfall methodology.

Dimension	**Decision**
Development process phases 0,1,2,3,4,5	Hybrid - Waterfall with milestones (see schedule) Phase 1 with well defined scope will follow an agile delivery model for the initial launch.
Source code repository	Github project
Contribution form	Open-source with pull requests to maximize participation
Versioning	Gitflow
Bug fix tracking after initial build	Git issues reported on Github and tracked via JIRA Kanban. Enable high visibility
Consortium CI/CD	Jenkins Pipeline for two AWS managed global channels and Patient Identity Service. Will build and deploy peer nodes and chaincode (see Operationalization:upgrades) Organizations which use their own infrastructure must create their own pipeline to build chaincode and perform deployments for private channels.

Solution Delivery Roadmap

Schedule of Deliverables

Overall solution development and delivery will be done in multiple phases over 24 months. .
Each phase has a goal, fixed scope, start, end dates, stakeholders and deliverables.



Phase 0: Planning
Goal: Define the Scope and Work Breakdown, Sign up a pilot Healthcare Provider Network and their EHR vendor.
Start: January 4, 2021
End: April 30, 2021
Stakeholders: Consortium Board Members
Tasks:

1. Define the scope of work and break down the work into Phases and within individual phases, break them down into Epics and Stories
2. Establish clear development and delivery timelines for each phase
3. Establish stakeholders for each phase
4. Recruit Teams for each phase
5. Find a Healthcare Provider Network and their EHR vendor to enter into an agreement with the Consortium to work during the initial implementation as a Pilot vendor
6. Work on Regulatory needs and risk assessments

Deliverables:

1. Consortium Charter
2. Work Breakdown Structure
3. Stakeholder and Owner list for each phases
4. Healthcare Provider Network and EHR contract signed
5. Regulatory approvals
6. Risk mitigation plans

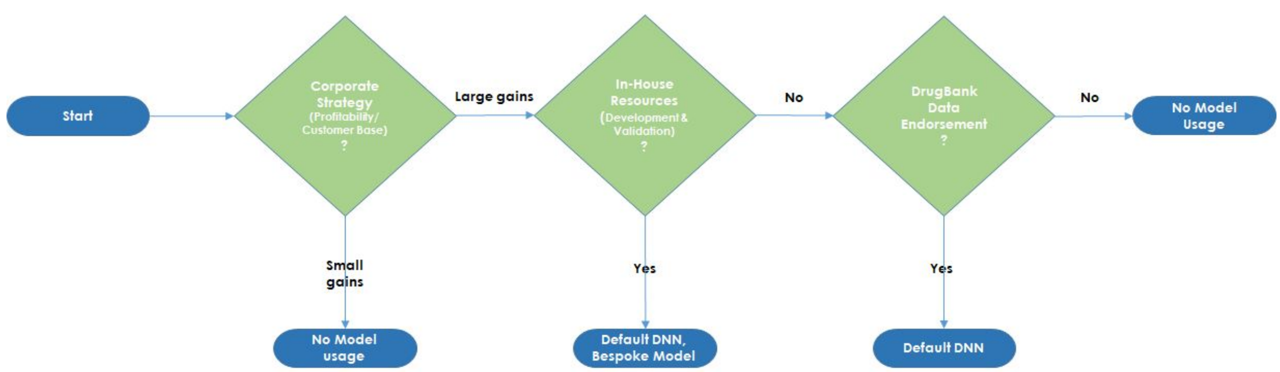
Phase 1: Initial Setup / Infrastructure / Development
Goal: Set up the initial TEST blockchain network, build the initial chaincode, build patient identification service.
Start: May 3, 2021
End: June 30, 2021
Stakeholders: Consortium Dev Team and DevOps Team members
Tasks:

1. Bootstrap consortium's Root Fabric Certificate Authority (CA) server and generate the self-signed Root certificate
2. Create a Managed Blockchain Infrastructure for the test network in AWS with two global channels, a test organization, MSPs and create genesis block for peers in each channel

3. Setup simplistic Analytics model to run as container in the test organization channel
4. Load test patient identities and medication list data into global patient and medication channels for test network
5. Develop and build initial chaincode for PrescriptionRequest, DDIResults, MedicationList, and PatientIdentity workflows
6. Create a lightweight SDK that'll enable the EHR Client-peer nodes to talk to the chaincode. This is useful when they are listening to a DDI response. EHR listens to the blockchain directly for DDI responses for better response times/performance.
7. Introduce OAUTH 2.0 to chaincode and test integration with multiple providers
8. Patient Identity Service - Consortium develops this (REST) service. Uses the SDK. Open source project. Integrates with PatientIdentity chaincode

Deliverables:
1. Functioning TEST Network ready for integration with the pilot EHR vendor
2. Basic Analytics models, tested and working
3. Global Patient and Medication TEST Channels
4. Chaincode implementation to make the prescription request and update blockchain back
5. Fully tested first version Patient Identity Service

Phase 2: Develop Analytics and first EHR integration for Pilot
Goal: To develop an advanced analytics model and collaborate with an EHR vendor (used by the pilot customer) to perform the first integration with the blockchain network. The pilot organization will join the TEST blockchain network.
Start: July 1, 2021
End: September 30, 2021
Stakeholders: Consortium Dev/DevOps teams, EHR Dev/DevOps teams, Healthcare Provider Network DevOps teams
Tasks:
1. Complete Premarket Submission[19] of DNN-model to FDA.
2. Deploy initial default model (DNN) and automated update-cycle.
3. Deploy API
4. Assist the new consortium member in selecting the analytic model. Selection will be based on the following decision process:



5. Create Fabric channel in AWS Managed Blockchain for Pilot organization to run analytics for its patients
6. Introduce containerized Fabric analytics-peer in Pilot organization channel using the default analytics model

7. Provide consortium API access information to consortium members using the default model, EHR integration falls outside consortium responsibility.
8. Ensure that a new consortium member understands the concept of FDA guidelines and “fit for purpose” argumentation, as majority of analytic models and infrastructure will be proprietary, and FDA inspects complete system fit.
9. EHRs/Provider networks/Medical groups will have the ability to download the prepackaged Client-peer nodes and build them as containers, configure them in their own organization channel.
10. Load real pilot patient identities and medication list data into global patient and medication channels for test network

Deliverables:

1. Fully onboarded / integrated EHR vendor and the Healthcare provider network into the TEST blockchain
2. EHR vendor and Healthcare provider network completing their infrastructure and deploying the Chaincode and Analytics engine as containers
3. Test data setup
4. Pre-market submission of DNN model to FDA
5. Submit the DNN predictor for internal approval (data science)
6. Access-log reporting for KPIs

Phase 3: Functional Testing, Performance Testing, Security Testing

Goal: Complete the end to end functional, performance and security testing of the system and infrastructure in the TEST network.

Start: October 1, 2021

End: January 31, 2022

Stakeholders: Consortium Board members, Dev/DevOps teams of the Consortium, Pilot EHR Vendor and the Pilot Healthcare Provider Network, Testing teams of all parties, Third-party Security testing vendor

Tasks:

1. Functionally test the Pilot integration
2. Write end to end automation tests that touches all the different components of the system like chaincode and the models in the TEST environment
3. Integration test with the pilot provider network and the pilot EHR and run the end to end automation tests submitting prescription requests from the Provider Network’s Physician offices but to the TEST network
4. Performance Test the TEST infrastructure with the Pilot EHR for varying loads and varying rates and capture the baseline performance of the network and system
5. Simulate multiple EHR vendors and performance test the network by scaling up and capture the baseline performance of the network and system under low/expected/high loads
6. Verify that the performance falls within the SLA. Make performance fixes/system tuning as needed
7. Perform a security/penetration testing on the TEST infrastructure and remediate the findings

Deliverables:

1. Functional Test Completion signoff
2. Documented system performance guidelines under varying loads
3. Security/Penetration test signoff and certification in TEST from industry standard vendors
4. Stable TEST infrastructure ready to scale up with multiple EHRs/provider networks

Phase 4: Operational Readiness and Full Launch of Pilot in PROD

Goal: Perform the first integration in the PROD blockchain network. The pilot organization will join the PROD blockchain network.
Prerequisites: FDA approval of the DNN-model (for PROD launch)
Start: October 1, 2021
End: March 31, 2022
Stakeholders: Consortium Board members, Dev/DevOps teams of the Consortium, Pilot EHR Vendor and the Pilot Healthcare Provider Network, Testing teams of all parties, Third-party Security testing vendor
Tasks:

1. Create an Operational Readiness and an onboarding manual and get buy in from the Pilot EHR/Pilot Healthcare Provider network
2. Automate the Infrastructure creation using code/market standard tools
3. Create the Production infrastructure using the automated process
4. Create the Global channels in Production and a Production Organization
5. Onboard the Pilot EHR Vendor and the Pilot Healthcare Provider Network in Production - Exchange certificates and run a few test transactions to verify the end to end connectivity
8. Pilot EHR to deploy their Machine Learning Models in their Production. Upon FDA's approval, they may choose to deploy and use the default model created by consortium in Production or use their own models
9. Perform a security/penetration testing on the PROD infrastructure and remediate the findings, if any. Get Third-party certification on the Production security
10. Launch the system to PROD use with the Pilot EHR/Pilot Healthcare Provider Network and start collecting metrics
11. Optimize the Production infrastructure and fix functional or integration issues as they come up and stabilize the system

Deliverables:

1. Operational Readiness and Onboarding manuals
2. Documented PROD system performance guidelines under varying loads
3. Security/Penetration test signoff and certification in PROD from industry standard vendors
4. Stable PROD infrastructure with expected volume of data performing under the agreed SLA for the Pilot EHR/Pilot Healthcare Provider network

Phase 5: Extended Pilot with multiple Organizations
Goal: Add more Pilot EHRs/Pilot Healthcare Provider Networks to the consortium.
Prerequisites: Stable Production environment with one Pilot EHR/Healthcare Provider Network
Start: April 4, 2022
End: December 30, 2022
Stakeholders: Consortium Board members, Dev/DevOps teams of the Consortium, Pilot EHR Vendors and the Pilot Healthcare Provider Networks, Testing teams of all parties, Third-party Security testing vendor
Tasks:

1. Consortium to actively market and bring multiple Healthcare Provider Networks (and associated EHRs) to participate and use the system.
2. Onboard the new Pilot EHRs to the TEST system and perform the End to End testing and sign off
3. Create Production Organizations for the new Pilot EHR Vendors and the Pilot Healthcare Provider Networks
4. Onboard the Pilot EHR Vendor and the Pilot Healthcare Provider Network in Production - Exchange certificates and run a few test transactions to verify the end to end connectivity

5. Pilot EHRs to deploy their Machine Learning Models in their Production. Upon FDA's approval, they may choose to deploy and use the default model created by consortium in Production or use their own models
6. Perform a security/penetration testing on the PROD infrastructure of the individual EHRs and remediate the findings, if any. Get Third-party certification on the Production security
7. Launch the system to PROD use with the Pilot EHR/Pilot Healthcare Provider Network and start collecting metrics

Deliverables:

1. Security/Penetration test signoff and certification in PROD from industry standard vendors
2. Stable PROD infrastructure with expected volume of data performing under the agreed SLA for all the Pilot EHRs/Pilot Healthcares Provider network
3. Production system stability metrics for an extended period of time

Operationalization

A new Organization (Healthcare Provider Network) must be approved by the consortium and given an E-prescription identity in the form of intermediate certification (X.509). The new Organization will need to make twol critical infrastructure decisions before joining the network.

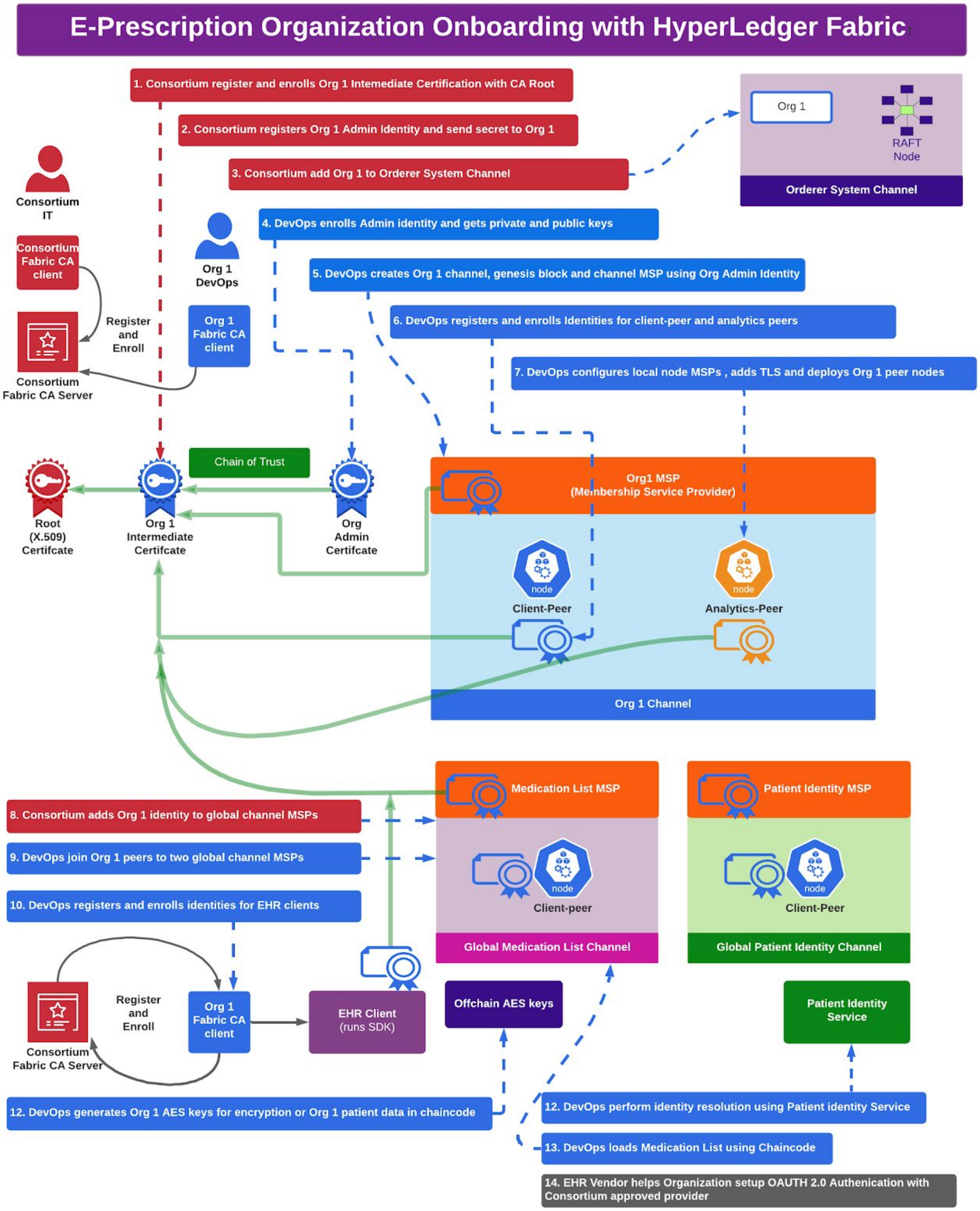
Organization Onboarding Decisions

An organization can choose to (1) allow the consortium to host the organization's Fabric infrastructure AWS Managed Blockchain network or (2) host their own infrastructure in a cloud-provider or on-premise.

(1) Cloud-hosted: An organization can contract with the consortium to acquire an isolated Fabric channel running the default analytics model and peer-client nodes, as well as deploy pre-configured peer nodes to run the two global channels. They will not need to deploy RAFT nodes or create peer-client nodes and configure MSPs on nodes. Instead, they will rely on the consortium to build and deploy all the blockchain peer nodes and perform all chaincode updates. The consortium will register identities for EHR clients, and allow the organization to enroll these identities using a Fabric CA client. However, the organization will not need to enroll admin identities or intermediate certificates. If they choose the managed network, they are limited to using a default Analytics channel and capabilities.

(2) A large sophisticated organization, such as a metropolitan hospital, may choose to implement and deploy its own infrastructure, peer nodes and analytics. The organization data scientist or clinical staff will evaluate cost-benefit and efficacy of potential models vs the default model. Since the Analytics model runs as a container with standardized REST endpoints in the same container as the analytics-peer, the organization will build any kind of containerized service they choose. The organization will be responsible for building and deploying peer nodes and peer-clients to the organization channel. They will be responsible using an admin certificate registering and enrolling all identities for the organization channel peers and EHR clients. They will be responsible for managing updates to chaincode in their own private channel. They are responsible for keeping their Fabric infrastructure (SDK, peer binaries, feature support) compatible with the supported versions of software. The consortium will enable the organization to participate in two global channels and perform requests to PatientIdentity and MedicationList chaincode. The consortium will allocate more resources (cpu/memory/# of peers) to global channel to scale for the new organization.

Join a new Organization to the Network



For an organization to join and deploy its own peers nodes to the blockchain network, it must coordinate with the consortium to create and enroll a new organization identity. A chain of trust is enforced by X.509 certificates (representing identities) generated by the consortium's Fabric certificate authority server. The organization admin identity has the privilege to create a channel for the organization and create the ledger genesis block. DevOps will create identities for peer nodes and add the organization's intermediate certificate to the Membership Provide Service (MSP) in each channel so that peers can participate in PrescriptionRequest and DDIResults transactions. The consortium will also add the organization's intermediate certificate to the global channels so that the organization can access Medication Lists and Patient Identities. Finally, DevOps will enable the EHR to integrate with the blockchain network by enrolling a client identity. For an organization which chooses to deploy its own infrastructure, it only needs to explicitly request an EHR client identity using a Fabric client. A Fabric client should always be used to prevent sharing of the private key of the identity. All other identities are created and managed by the consortium.

Availability

The operational availability of global channels should be characterized as 24/7 highly available. AWS Managed Blockchain abstracts away maintenance of individual EC2 instances and advertises as a reliable service. The peers for the two global channels run as managed instances in AWS and may run in multiple AWS availability zones. The distributed nature of Hyperledger Fabric blockchain means ledger and world state are replicated across many database instances running on each peer node. The RAFT order system is clustered and fault tolerant. Therefore, the failure of any node instance, single-node database corruption or network disruption will not affect integrity and availability to perform transactions and retrieve data by a peer-client or from the SDK.

An organization running their own infrastructure should select a containerized deployment such as Kubernetes which can be deployed and run in the cloud. For Organizations who choose to run their Kubernetes in the cloud, the consortium will provide guidance and instructions to deploy both the offchain components such as the Patient Identity Service and Fabric components such as peer nodes and RAFT orderers as Docker containers. Kubernetes provides the sophisticated capabilities to manage containers in declarative state and ensure availability with features such as automated restarts. To provide redundancy for the analytics-peer node, DevOps can deploy the analytics container as a Kubernetes replica-set. Only a well resourced organization with expertise in Kubernetes should pursue such an option. It is possible to run EC2 instances directly with peer chaincode and still integrate with a custom analytics service running on the same machine.

Recovery

The consortium will also perform frequent backups of global network components including, certificates, private/public keys, channel configurations, peer files (chaincode, ledgerData folders) and DB database files. In the event all the distributed blockchain is corrupted by incorrect transactions or in an irreconcilable ledger state, the consortium will perform point-in-time recovery, and rebuild a channel and all the peer nodes. Hyperledger Fabric does not include a utility. Rather the consortium will learn how to perform offline recoveries, and schedule backups daily. Recover is a last resort in blockchain. Newer blocks are permanently lost from ledger and cannot be recovered.

Monitoring and auditing

To monitor the health and performance Fabric components deployed in AWS Managed Blockchain such as peer nodes the consortium will use custom AWS Cloudwatch dashboards. TheConsortium DevOps will enable AWS Cloudwatch to capture chaincode logs for the global channels and managed organization channels. The consortium will also build custom Cloudwatch alerts to detect failing containers and chaincode error events. The consortium will sample logs for transaction latency and build custom Cloudwatch dashboards to display latency over time. It is important to measure slowness in getting results from Prescription Requests. Functional requirements state a prescriber should wait no longer than 3 seconds.

The consortium will backup logs offline for future auditing. In addition, the ledger which is persisted in the peer DB is a form of audit record. It tracks every transaction performed on Medication Lists, Prescription Requests, DDIResults and Patient Identity. The consortium will create query auditing reports which can be run on demand to retrieve operations performed at a patient level or for a subset of organizations.

Organizations can choose their own monitoring tools deploying their own infrastructure but the consortium will create instructions and demo projects using open-source Prometheus to capture Kubernetes container metrics and display resource usage in Grafana dashboards. To capture logs and provide an interface for searching logs, the consortium will provide instructions and demo projects using open-source ELK stack (Elastic Search, Logstash, Kibana).

Incident Management

The consortium provides first-line support to consortium members via a 3-step approach. First line support is a service desk organization which triages and prioritizes tickets. Once first line information gathering is completed, and user requests cannot be dealt with via training, tickets are distributed to second line IT technicians who ensure that infrastructure and other factors are not responsible for service disruptions. Should second line support not be able to solve the issue, third line support is responsible for problems related to the inner workings of the system, i.e. typically configuration items and codebase modifications. Long Term change requests received are studied, prioritized, assessed, and passed on to the change control board for further assessment.

Upgrades and Improvements

Stories are created in JIRA Kanban and prioritized by the consortium board. The consortium Director is responsible for organizing work and developing long term improvements, and as well as coordinating with the EHR vendor representatives for development resources. To introduce and complete a new feature a JIRA task with acceptance criteria is assigned to the development and team. The consortium uses Gitflow as the release model for versioning. The global CI/CD Jenkins pipeline can deploy changes to the network via the External Chaincode launcher available in Hyperledger Fabric 2.x. Initially, all the pipeline will deploy to TEST and go through a QA process before customer notification and deployment to PROD. Organizations with their infrastructure will need to build their pipeline or deploy chaincode manually. Since organizations run on their private channels they can run older versions of the chaincode, but the consortium will specify ranges of backwards compatibility. To upgrade the major Hyperledger Fabric version, the consortium will most likely need to take TEST and PROD network offline for several hours to upgrade binaries for node images in global channels. Organizations running their own infrastructure and private channels will also need to schedule downtime to upgrade major Fabric versions.

Bug-fixes

Issues will be reported on as Github issues and tracked in JIRA with Kanban board and assigned to consortium developers. Since this is open-source any developer can create pull-request to resolve and issue. A pull-request will be approved and merged by the Consortium team.

Offboarding

An Organization may choose to no longer participate in the blockchain network or an EHR vendor may choose to leave the consortium. In both cases, an organization or multiple organizations must be removed from the network; ledger and patient data will remain but the organization can no longer participate.   To remove organizations Hyperledger Fabric provides Certificate Revocation Lists (CRL). The consortium will add the organization identity to the global Membership Service Provider (MSP) as configuration. This will prevent any organization identity from accessing the blockchain to create transactions or read data.

Development and Implementation Roles

Consortium Member Roles

The EHR vendors will contribute members to the team or provide funding to directly hire specialists like blockchain developers or the consortium Director.

Role	Responsibilities
Blockchain Developer	1. Develops the fabric core 2. Develops chaincode 3. Develops APIs
EHR Developer	Responsible for Integration
DevOps Engineer	1. Creates CI/CD pipeline 2. Manages Infrastructure deployments and operations
Consortium Director	1. Reports to consortium board on budgetary concerns, project status and success metrics 2. Coordinates with Consortium to approve and onboard new organizationResponsible for

	regulatory compliance
Data Scientist	1. Implements Analytics model within organizations that choose to create their own
Consortium Project Manager/ Product Owner	1. Manages delivery timeline 2. Resource Allocation 3. Coordinates user training
Business Analyst	1. Creates Epics, Features and User stories in support of development 2. Testing 3. Creates user guides 4. Performs user training 5. Vendor Business Analyst system training
Vendor Business Analyst (System SME)	1. Trains new users 2. Coordinates training on enhanced functionality
Consortium EHR Board Representative	1. Represents interests of EHR like Cerner or AllScripts
Consortium Healthcare Provider Representative	1. Represents interests of a specific Healthcare provider Only some healthcare provider (who are interested and approved) will join the board. The first would be the Pilot customer.

Healthcare Provider Member Roles

Each Healthcare provider needs a team to implement the blockchain solution. The EHR vendor will already have developed the integration module for the provider's EHR software. Therefore, no new development is required by the provider.

Role	Function
Administrator	1. Administers roles 2. Facilitates vendor connectivity
DevOps Engineer	1. Responsible for onboarding to blockchain network and upgrades

	2. Integrate custom analytics 3. Monitor health and performance of local Fabric nodes and channel
Data scientist	1. Build analytics module Optional in case Healthcare provider decides to implement its own data analytics.

Consortium Member Roles

To ensure a successful implementation, the consortium's Onboarding team will work with EHR vendors business and technology units. The onboarding team will consist of an onboarding project manager; a solution expert; and two system trainers.

The onboarding team will be onsite with each vendor for five days (40-hours). Prior to Go-Live, the project manager and system trainers will contact the EHR vendor to establish the training schedule. The system trainers will work with the vendors System Administrator to setup test and production roles and register users. Training sessions will be tailored to the user's roles. For two weeks (80-hours) prior to onsite training, the vendor Analyst SME will work remotely with the onboarding team to gain an in-depth knowledge of the system to assist with user training and enablement both during and after the onboarding team has conducted user training.

User training will occur in the test environment to ensure a successful learning experience without the risk of impacting business operations. Each training course will consist of 4-hour sessions held twice daily in the morning and afternoon. Training will be recorded for users unable to attend training, and to assist users with an interactive form of a user guide for reference.

Together with vendor administration, the onboarding team will distribute emails to vendor users with descriptions of the course sessions, training times, and the URLs to register. Below are the training courses that will be offered based on the user's role.

User Role	Training Courses
Administrator	1. Register users 2. Remove users
Physicians, Physician Assistants and Registered Nurses	1. Navigate UI 2. Request patient medication list 3. Place prescription order 4. Understanding DDI results

Success Measurements and Performance Indicators

The consortium will generate reports from Fabric and EHR databases and logs to measure the following key performance indicators (KPIs). The list includes both metrics to be used from pilot onwards and metrics which would need minimum 6 months of production data to yield stable estimates.

KPI	**Execution timepoint**	**KPI Threshold**
DDI calculation count	Pilot and onwards	Minimum 1000 daily
DDI overrides frequency	Pilot and onwards	Maximum 10%
DDI literature usage frequency	Pilot and onwards	Minimum 20% or more
SLA adherence	Pilot and onwards	Minimum 99.5% of requests under 3 seconds roundtrip time (given minimum load)
DDI patient hospitalization frequency (before vs. after implementation)	Production	Minimum -20%
Before/After e-prescription outcomes (before vs. after implementation)	Production	Minimum -10% severe and sequelae
Avoided interventions frequency (before vs. after implementation)	Production 6months+	Minimum 20%
Avoided interventions cost / DDI patient	Production 6months+	Minimum USD 800
Type of medic user breakdown (ER,GP,Clinic,...)	Pilot and onwards	GP usage Minimum 90%
Reduced ER visit frequency in consortium member population	Production 6months+	Minimum 20% decrease
Consortium member savings versus consortium membership	Production 6months+	Minimum 3-fold profitability

Appendix 1: JSON

PrescriptionRequest

```
key = prescriptionRequestId UUID
value = {
    "type":  "PrescriptionRequest"
    "prescriber":
    "drugVocab": ""   //use RxNorm
    "drugName" :  ""
    "dosage" :  " take every 12 hours”
    “medicationList” : {
       "type":  "MedicationList",
       "orgs":  ["org1","org2"]
       "medications": [ {<PrescriptionRequest>}, {}, {} ]

     ]

}
```

DDIResults

```
key = prescriptionRequestId UUID
{
  type: "DDIResults",
    "GlobalPatient": "2FZbgi29cpjq2GjdwV8eyHuJJnkLtktZc6",
    "prescriptionRequestId": "123e4567-e89b-12d3-a456-426614174000",
    "Rxpair": [
      {
        "drug": [
          {
            "ingredientName": "gemfibrozil",
            "RxCui": "4719"
          },
          {
            "ingredientName": "cerivastatin",
            "RxCui": "596723"
          }
        ],
        "literature": "https://www.tandfonline.com/doi/pdf/10.1517/14740338.1.3.207",
        "DDIseverity": "0.99"
      },
      {
        "drug": [
          {
            "ingredientName": "haloperidol",
            "RxCui": "5093"
          },
          {
            "ingredientName": "trifluoperazine",
            "RxCui": "10800"
          }
```

```
        ],
        "literature":
"https://www.rxlist.com/drug-interactions/haloperidol-oral-and-trifluoperazine-oral-interactio
n.htm",
        "DDIseverity": "0.11"
      },
      {
        "drug": [
          {
            "ingredientName": "phenytoin",
            "RxCui": "8183"
          },
          {
            "ingredientName": "gabapentin",
            "RxCui": "25480"
          }
        ],
        "literature": "https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5308580/",
        "DDIseverity": "0.45"
      }
    ]
}
```

MedicationList

```
Key = globalPatientId 128-bit UUID
{
       "MedicationList": {
              "GlobalPatient": "2FZbgi29cpjq2GjdwV8eyHuJJnkLtktZc6",
              "organizations": ["Allscripts", "Athena", "GE"],
              "Rxdrug": [{
                            "ingredientName": "gemfibrozil",
                            "RxCui": "4719",
                            "prescriptionDate": "Tue Jul 12 18:35:37 EST 2016"
                     },
                     {
                            "ingredientName": "cerivastatin",
                            "RxCui": "596723",
                            "prescriptionDate": "Tue Jul 10 18:35:37 EST 2016"
                     },
                     {
                            "ingredientName": "haloperidol",
                            "RxCui": "5093",
                            "prescriptionDate": "Tue Jul 09 18:35:37 EST 2016"
                     },
                     {
                            "ingredientName": "trifluoperazine",
                            "RxCui": "10800",
                            "prescriptionDate": "Tue Jun 10 18:35:37 EST 2016"
                     },
                     {
                            "ingredientName": "phenytoin",
                            "RxCui": "8183",
                            "prescriptionDate": "Tue May 10 18:35:37 EST 2016"
```

```
                    },
                    {
                            "ingredientName": "gabapentin",
                            "RxCui": "25480",
                            "prescriptionDate": "Tue May 09 18:35:37 EST 2016"
                    }
            ]
    }
}
```

PatientIdentity

```
Key = globalPatientId 128-bit UUID
{
   "type":  "PatientIdentity",
   "globalPatientId": null,     //in this case it's not know yet so it is used in a request
   "orgs": ["org1", "org2"],
   "family": "Stardust",
    "given": [
        "Ziggy",
        "David"
      ],
    "dateOfBirth": "1947-01-08",
    "address": {
         "street": "10 Kings Crossing",
         "city":   "London",
         "country": "UK",
         "postalCode": "29293-22"
       }
}
```

PatientIdentityMatchResults

```
No key. Only transient
{
    "type": "PatientIdentityMatchResults",
    "input": PatientIdentity1  //an object of type PatientIdentity
    "output": [
        {
            "probability": ".40"        //40% probability based on demographics match
            "match": PatientIdentity2  //object of type PatientIdentity
        },
        {
            "probability": ".95"        //95% probability based on demographics match
            "match": PatientIdentity3  //object of type PatientIdentity
        },
     ]
}
```

Market research / Industry references:

[1] Holmgren, A., Patel, V., & Adler-Milstein, J. (2017). Progress in Interoperability: Measuring US Hospitals' Engagement in Sharing Patient Data. Health Affairs. 36, 1820-1827. Retrieved from https://www.healthaffairs.org/doi/full/10.1377/hlthaff.2017.0546

[2] Centers for Disease Control and Prevention (2020). Adverse Drug Events in Adults. Retrieved from https://www.cdc.gov/medicationsafety/adult_adversedrugevents.html

[3] Ning Liu, Cheng-Bang Chen, Soundar Kumara (2019). Semi-Supervised Learning Algorithm for Identifying High-Priority Drug–Drug Interactions Through Adverse Event Reports. Retrieved from https://ieeexplore.ieee.org/document/8786248

[4] Marinka Zitnik, Monica Agrawal, Jure Leskovec (2018). Modeling polypharmacy side effects with graph convolutional networks. Retrieved from https://academic.oup.com/bioinformatics/article/34/13/i457/5045770

[5] Tsung-Ting Kuo, Hugo Zavaleta Rojas, Lucila Ohno-Machado (2019). Comparison of blockchain platforms: a systematic review and healthcare examples. Retrieved from https://academic.oup.com/jamia/article/26/5/462/5419321

[6] Statistics on Adverse Drug Interactions Retrieved from https://www.fda.gov/drugs/drug-interactions-labeling/preventable-adverse-drug-reactions-focus-drug-interactions

[7] David C. Classen, MD, MS1; A. Jay Holmgren, MHI2; Zoe Co, BS3; et al. National Trends in the Safety Performance of Electronic Health Record Systems From 2009 to 2018. Retrieved from https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2766545

[8] Electronic Health Records Fail To Detect Up To 1 in 3 Harmful Drug Interactions And Other Medical Errors. Retrieved from https://www.forbes.com/sites/marlamilling/2020/05/29/electronic-health-records-fail-to-detect-up-to-1-in-3-harmful-drug-interactions-and-other-medical-errors/#41a49f8960a4

[9] Shyam Pratap Singh (May 19, 20190. Detail Analysis of Raft & its implementation in Hyperledger Fabric. Retrieved from https://spsingh559.medium.com/detail-analysis-of-raft-its-implementation-in-hyperledger-fabric-d269367a79c0

[10] Nasir, Qassim ; Qasse, Ilham A ; Abu Talib, Manar ; Nassif, Ali Bou (September 2018). Performance Analysis of Hyperledger Fabric Platforms. Retrieved from Harvard Hollis: http://downloads.hindawi.com/journals/scn/2018/3976093.pdf

[11] Wishart et al. (January 2018). Major update to Drugbank 5.0. Retrieved from https://academic.oup.com/nar/article/46/D1/D1074/4602867?searchresult=1

[12] US National Library of Medicine. SignalDetDDI. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6242685/

[13] PNAS. Deep learning improves prediction of drug-drug interactions. https://www.pnas.org/content/115/18/E4304

[14] Wikipedia. AERS. Retrieved from https://en.wikipedia.org/wiki/Adverse_Event_Reporting_System

[15] Noren. A statistical methodology for drug–drug interaction surveillance. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.3247

[16] Wikipedia. Adverse Event. Retrieved from https://en.wikipedia.org/wiki/Adverse_event

[17] Rubricki. Drug-Drug Interactions Discovery based on CRFs, SVMs and rule-based Methods. Retrieved from http://ceur-ws.org/Vol-761/paper8.pdf

[18] Wikipedia. Support Vector Machine. Retrieved from https://en.wikipedia.org/wiki/Support_vector_machine

[19] FDA (May 2018). Appropriate Use of Voluntary Consensus Standards in Premarket Submissions. Retrieved from https://www.fda.gov/media/71983/download