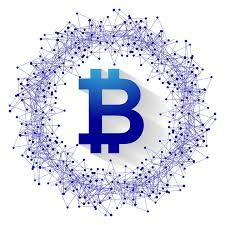
Data Warehouse Project Architecture

Team BLOCKCHAIN



# HealthCare Blockchain System

**Members**

Aarthi Madhavan

Akhila Jana

Priyanka Sharma

Rubini Balasubramanian

Kavya Mandli

Table of Contents

[HealthCare Blockchain System 1](#_Toc510647236)

[1. Introduction 4](#_Toc510647237)

[1.1 Purpose 5](#_Toc510647238)

[1.2 Scope 5](#_Toc510647239)

[1.3 References 5](#_Toc510647240)

[2.  Architectural Representation and Technology Used 5](#_Toc510647241)

[2.1 Basic Workflow of how Blockchain Works 5](#_Toc510647242)

[2.2 Web and mobile interface: 7](#_Toc510647243)

[2.3 Configuration 7](#_Toc510647244)

[2.4 PERSONAS 8](#_Toc510647245)

[2.5 Ledger 8](#_Toc510647246)

[3.  Architectural Goals and Constraints 9](#_Toc510647247)

[4.  Use-Case View 9](#_Toc510647248)

[4.1  Architecturally-Significant Use Cases 10](#_Toc510647249)

[*4.1.1 - Register* 10](#_Toc510647250)

[*4.1.2 Insurance Plan approval* 10](#_Toc510647251)

[*4.1.3 Medical Condition Report* 11](#_Toc510647252)

[*4.1.4 Medical Counsellor Generation* 11](#_Toc510647253)

[*4.1.5 Bill and claim Generation* 11](#_Toc510647254)

[*4.1.6 Claim Settlement* 11](#_Toc510647255)

[*4.1.7 View Medical Card* 11](#_Toc510647256)

[*4.1.8 Close Registration* 11](#_Toc510647257)

[5.  Logical View 11](#_Toc510647258)

[5.1 State 12](#_Toc510647259)

[5.2 Ledger 12](#_Toc510647260)

[5.3. Nodes 12](#_Toc510647261)

[5.4  Architecture Overview – Package and Subsystem Layering 13](#_Toc510647262)

[6.  Process View 13](#_Toc510647263)

[6.1 Processes 14](#_Toc510647264)

[6.1.1 The client creates a transaction and sends it to endorsing peers of its choice 15](#_Toc510647265)

[6.1.2 The endorsing peer simulates a transaction and produces an endorsement signature 15](#_Toc510647266)

[6.1.3 The submitting client collects an endorsement for a transaction and broadcasts it through ordering service 15](#_Toc510647267)

[6.1.4. The ordering service delivers a transactions to the peers 16](#_Toc510647268)

[6.2 Process to be delivered in current model. 17](#_Toc510647269)

[6.3 Rest Proxy to CouchDB 19](#_Toc510647270)

[8.  Size and Performance 19](#_Toc510647271)

[9.  Quality 19](#_Toc510647272)

# Introduction

The blockchain is an undeniably ingenious invention – the brainchild of a person or group of people known by the pseudonym, Satoshi Nakamoto. But since then, it has evolved into something greater, and the main question every single person is asking is

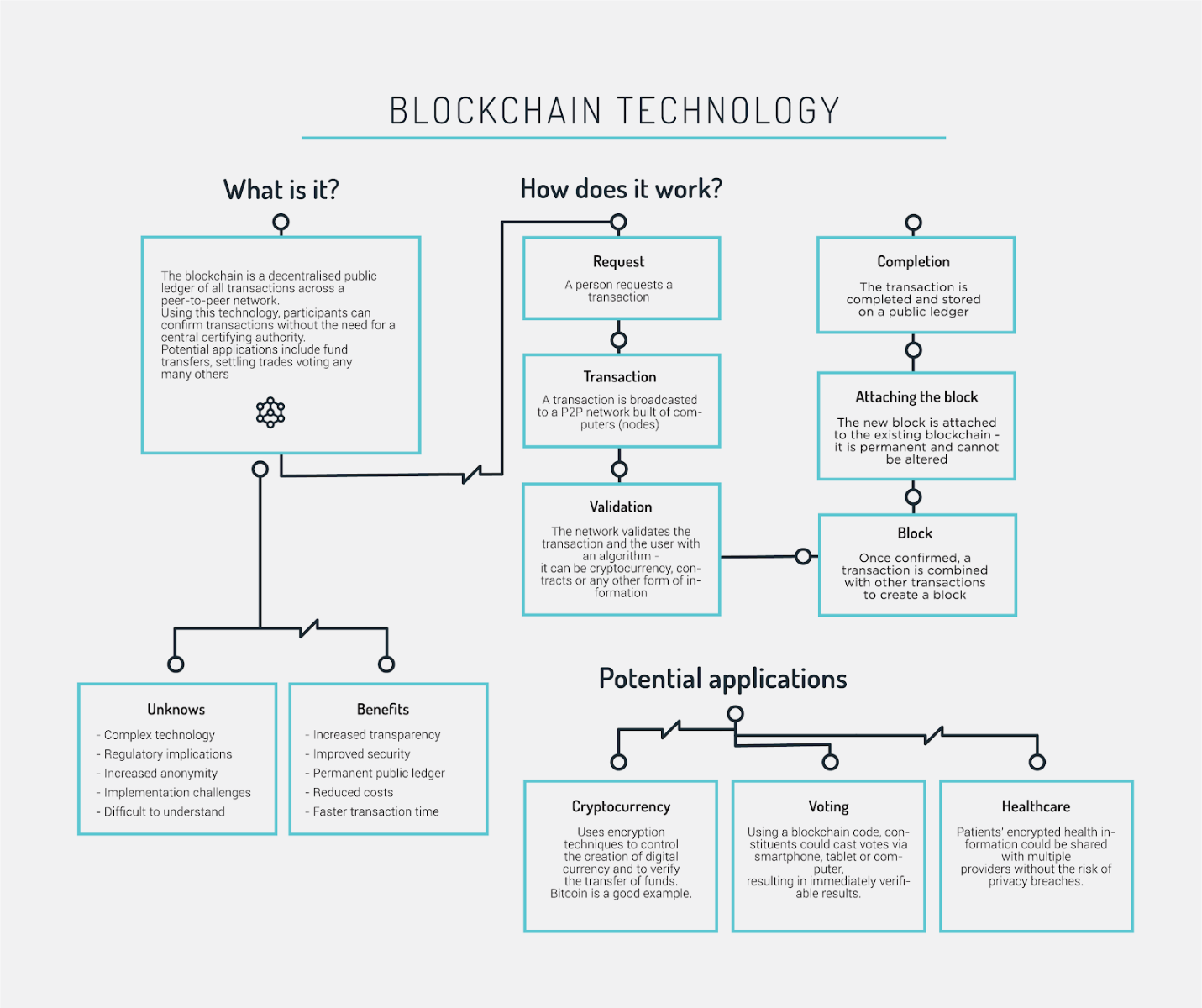
What is Blockchain?

By allowing digital information to be distributed but not copied, blockchain technology created the backbone of a new type of internet. Originally devised for the digital currency, Bitcoin,  (Buy Bitcoin) the tech community is now finding other potential uses for the technology.

“The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value.”

Blockchain technology is like the internet in that it has a built-in robustness. By storing blocks of information that are identical across its network, the blockchain cannot:

1. Be controlled by any single entity.  
2. Has no single point of failure.



1.1 Purpose

The goal of the blockchain in healthcare domain:  
1. Improve the patient experience of care (including quality and satisfaction).  
2. Improve the health of populations.  
3. Reduce the per capita cost of healthcare.

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

This Software Architecture Document provides an architectural overview of the HealthCare Blockchain System. We define how we establish the connection, communication between actors through user interface and blockchain network.

1.3 References

Applicable references are:

1. http://hyperledger-fabric.readthedocs.io/en/release-1.1/prereqs.html
2. http://avexiahealth.com/
3. https://hitconsultant.net/2018/01/29/blockchain-technology-in-healthcare-benefits/

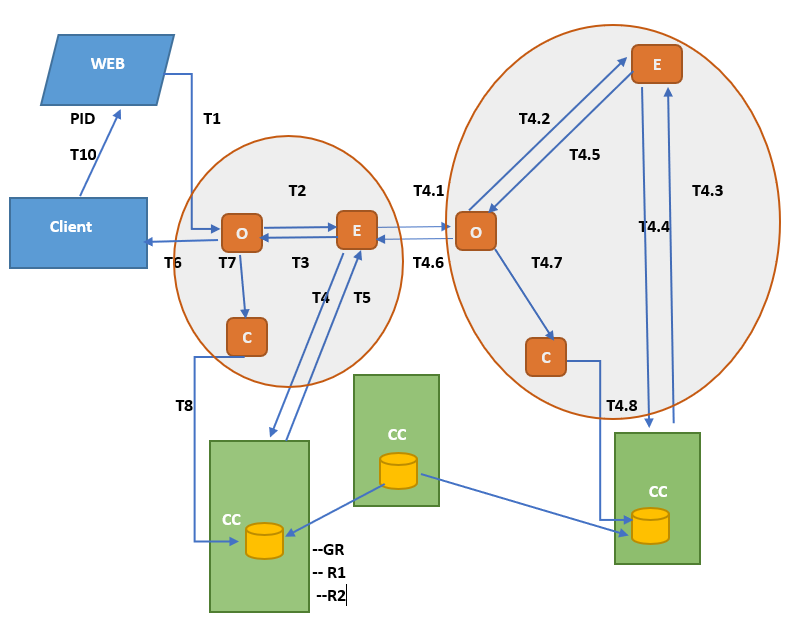
# 2.  Architectural Representation and Technology Used

This document presents the architecture as a series of views; use case view, logical view and process view.

## 2.1 Basic Workflow of how Blockchain Works

Hyperledger Fabric is a platform for distributed ledger solutions, underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility and scalability. It is designed to support pluggable implementations of different components, and accommodate the complexity and intricacies that exist across the economic ecosystem.

Hyperledger Fabric delivers a uniquely elastic and extensible architecture, distinguishing it from alternative blockchain solutions.

  
  
***Blockchain in Healthcare Intelligence Network***

Ti<-Transaction flow

O<-Orderer

C<-Committer

E<-Endorser

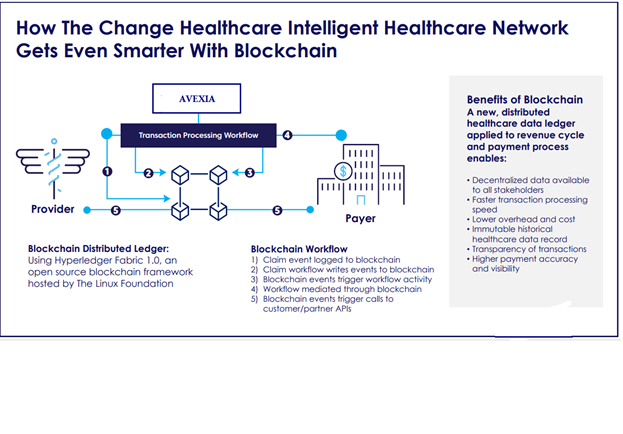
CC<-ChainCode

GR<-Genesis Node entered into CouchDB warehouse

Ri<-Subsequent nodes stored in CouchDB warehouse

Blockchain system facilitates our current model as:-

* **Chaincode trust flexibility.** The architecture separates trust assumptions for chaincodes (blockchain applications) from trust assumptions for ordering. In other words, the ordering service may be provided by one set of nodes (orderers) and tolerate some of them to fail or misbehave, and the endorsers may be different for each chaincode.
* **Scalability**. As the endorser nodes responsible for particular chaincode are orthogonal to the orderers, the system may scale better than if these functions were done by the same nodes. In particular, this results when different chaincodes specify disjoint endorsers, which introduces a partitioning of chaincodes between endorsers and allows parallel chaincode execution (endorsement). Besides, chaincode execution, which can potentially be costly, is removed from the critical path of the ordering service.
* **Confidentiality.** The architecture facilitates deployment of chaincodes that have confidentiality requirements with respect to the content and state updates of its transactions.
* **Consensus modularity.** The architecture is modular and allows pluggable consensus (i.e., ordering service) implementations.



## 2.2 Web and mobile interface:

To register and update medical records a web-interface and mobile application will be developed. All actors patient, hospitals, doctors, counsellor, insurance company use this interface to communicate fields in an interactive way abstract to the underlying blockchain architecture complexity.

## 2.3 Configuration

* Install cURL
* Docker and Docker Compose
* Go programming language 1.9.x for many of its components.

## 2.4 PERSONAS

* Patient
* Counsellor
* Insurance Company
* Doctor and Hospital

## 2.5 Ledger

* Ledger is constructed by the ordering service as a totally ordered hashchain of blocks of (valid or invalid) transactions. Following information will be stored in the Ledger.

|  |  |
| --- | --- |
| PatientInfo | * PatientID * Name * Location * Salary * EmergencyContact |
| MedicalInsurancePlan | * PolicyNo * CoveragePlan * InsurerId * Coverage * Claim |
| CurrentCondition | * MedicalIssue |
| Credits | * AvailableCredits * CreditsCovered * AmountUncovered |
| Counselors | * ID * Name * Location * Type * Bill |

# 3.  Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

* 1. Information related to insurance policy and rules are difficult to retrieve and implement with the current time frame so we would be using few randomly generated test sample values for our framework.
  2. The existing legacy Billing System at Hospital/Clinics must be interfaced with to support billing of patient.
  3. All participants have internet dial up connections.
  4. The must ensure complete protection of data from unauthorized access.

# 4.  Use-Case View

A description of the use-case view of the software architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

The Insurance-Blockchain use cases are:

- Register

- Insurance Plan approval

- Medical Condition Report

- Medical Counsellor Generation

- Bill and claim Generation

- Claim Settlement

- View Medical Card

- Close Registration.

These use cases are initiated by the patient, insurance company, councellor, clinic or hospital. In addition, interaction with external actors; Billing System occur.

## 4.1  Architecturally-Significant Use Cases

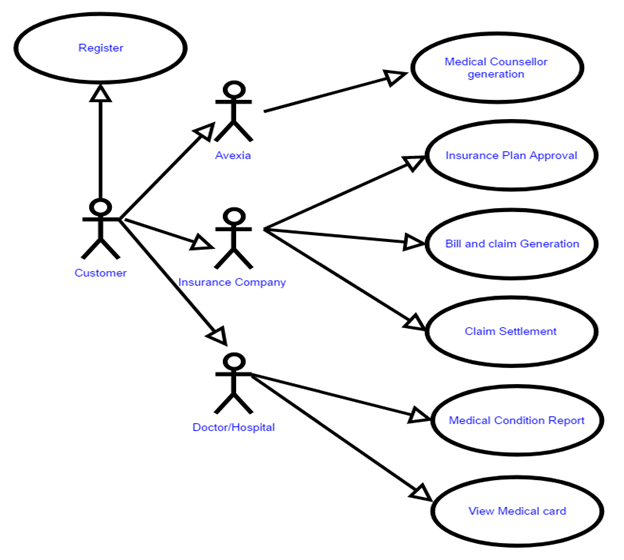


Diagram Name: Architecturally Significant Use-Cases

### *4.1.1 - Register*

Brief Description: This use case allows a registration process.

### *4.1.2 Insurance Plan approval*

Brief Description: This use case allows verification on an onboard customer through the insurance company under policyID.

### *4.1.3 Medical Condition Report*

Brief Description: This use case allows the patient to enter in current medical condition and update on progress.

### *4.1.4 Medical Counsellor Generation*

Brief Description: This use case allows Councellor to generate medical aid to the patient current medical condition, it generates a list of counsellors also reports close peer group of the medical condition after first treatment done.

### *4.1.5 Bill and claim Generation*

Brief Description: This use case allows the billing system to put patient bill into system and current medical claim in generated for insurance company consideration.

### *4.1.6 Claim Settlement*

Brief Description: This use case allows a insurance company to examine bill and compute amount covered and un-covered.

### *4.1.7 View Medical Card*

Brief Description: This use case allows clinic/hospital to examine patient medical history based on previous states of the block chain network.

### *4.1.8 Close Registration*

Brief Description: This use case allows both insurance company and patient to end their services on network.

# 5.  Logical View

A description of the logical view of the architecture. Describes the most important components, their organization in service packages and subsystems, and the organization of these subsystems into layers.

The logical view of the course registration system is comprised of the 3 main packages: User Interface, Business Services, and Business Objects.

The User Interface Package contains web and mobile interface for each of the forms that the actors use to communicate with the System. Boundary cases exist to support login, register, maintaining of schedules, maintaining of patient info, selecting counselor, providing counsellor, updating patient condition, submitting bills, maintaining insurance money disbursement, closing registration, and viewing patients medical card.

The Business Services Package contains chaincodes for interfacing between with the ordered, endorser and committer of the participating nodes on the blockchain network.

The Business Objects Package includes state of ledger, which information is visible through read/write access and which participants have access to the channels.

## 5.1 State

The latest state of the blockchain (or, simply, *state*) is modeled as a versioned key-value store (KVS), where keys are names and values are arbitrary blobs. These entries are manipulated by the chaincodes (applications) running on the blockchain through put and get KVS-operations. The state is stored persistently and updates to the state are logged.

## 5.2 Ledger

Ledger provides a verifiable history of all successful state changes (we talk about *valid*transactions) and unsuccessful attempts to change state (we talk about *invalid* transactions), occurring during the operation of the system.

Ledger is constructed by the ordering service as a totally ordered hashchain of *blocks* of (valid or invalid) transactions. The hashchain imposes the total order of blocks in a ledger and each block contains an array of totally ordered transactions. This imposes total order across all transactions.

Ledger is kept at all peers and, optionally, at a subset of orderers

## 5.3. Nodes

Nodes are the communication entities of the blockchain. A “node” is only a logical function in the sense that multiple nodes of different types can run on the same physical server. What counts is how nodes are grouped in “trust domains” and associated to logical entities that control them.

There are three types of nodes:

* **Client** or **submitting-client**: a client that submits an actual transaction-invocation to the endorsers, and broadcasts transaction-proposals to the ordering service.
* **Peer**: a node that commits transactions and maintains the state and a copy of the ledger. Besides, peers can have a special **endorser** role.
* **Ordering-service-node** or **orderer**: a node running the communication service that implements a delivery guarantee, such as atomic or total order broadcast.

## 5.4  Architecture Overview – Package and Subsystem Layering

UI

UI

HealthCare Blockchain Network

UI

UI

In our current model we propose to store following:-

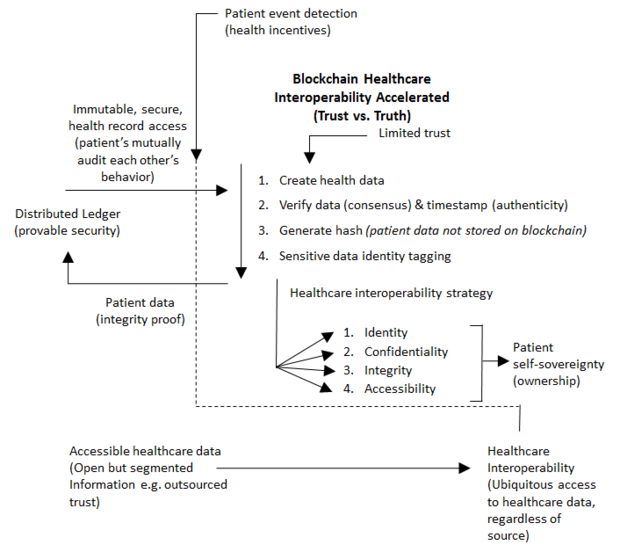
* A List of proposed counselors(doctors/hospitals).
* Bill details based on Counselor opted by patient.
* Based on bill detail the Insurance company is informed of the claim.
* Insurance company updates credits covered and AvailableCredits.
* Patient needs to pay the AmountUncovered thus computed.
* Once consensus on Claim,Bill,AvailableCredits,AmountUncovered is made, new state is committed in the network.

# 6.  Process View

A description of the process view of the architecture. Describes the tasks (processes and threads) involved in the system's execution, their interactions and configurations.

To protect information access to different participants on the network, we have created two channels:-

1. Channel Confidential: -
   1. Insurance Company
   2. Doctor
   3. Hospital
2. Channels Protected: -
   1. Councellor
   2. Insurance Company

* 

## 6.1 Processes

In the following we outline the high-level request flow for a transaction.

### 6.1.1 The client creates a transaction and sends it to endorsing peers of its choice

To invoke a transaction, the client sends a PROPOSE message to a set of endorsing peers of its choice. The set of endorsing peers for a given chaincodeID is made available to client via peer, which in turn knows the set of endorsing peers from endorsement policy. The transaction could be sent to *all* endorsers of a given chaincodeID. That said, some endorsers could be offline, others may object and choose not to endorse the transaction. The submitting client tries to satisfy the policy expression with the endorsers available.

### 6.1.2 The endorsing peer simulates a transaction and produces an endorsement signature

On reception of a <PROPOSE,tx,[anchor]> message from a client, the endorsing peer epID first verifies the client’s signature clientSig and then simulates a transaction. If the client specifiesanchor then endorsing peer simulates the transactions only upon read version numbers of corresponding keys in its local KVS match those version numbers specified by anchor.

Simulating a transaction involves endorsing peer tentatively *executing* a transaction (txPayload), by invoking the chaincode to which the transaction refers (chaincodeID) and the copy of the state that the endorsing peer locally holds.

As a result of the execution, the endorsing peer computes *read version dependencies* (readset) and *state updates* (writeset), also called *MVCC+postimage info* in DB language.

### 6.1.3 The submitting client collects an endorsement for a transaction and broadcasts it through ordering service

The submitting client waits until it receives “enough” messages and signatures on (TRANSACTION-ENDORSED, tid, \*, \*) statements to conclude that the transaction proposal is endorsed.

The exact number of “enough” depend on the chaincode endorsement policy. If the endorsement policy is satisfied, the transaction has been *endorsed*; note that it is not yet committed. The collection of signed TRANSACTION-ENDORSED messages from endorsing peers which establish that a transaction is endorsed is called an *endorsement* and denoted by endorsement.

If the submitting client does not manage to collect an endorsement for a transaction proposal, it abandons this transaction with an option to retry later.

For transaction with a valid endorsement, we now start using the ordering service. The submitting client invokes ordering service using the broadcast(blob), where blob=endorsement. If the client does not have capability of invoking ordering service directly, it may proxy its broadcast through some peer of its choice. Such a peer must be trusted by the client not to remove any message from the endorsement or otherwise the transaction may be deemed invalid. Notice that, however, a proxy peer may not fabricate a valid endorsement.

### 6.1.4. The ordering service delivers a transactions to the peers

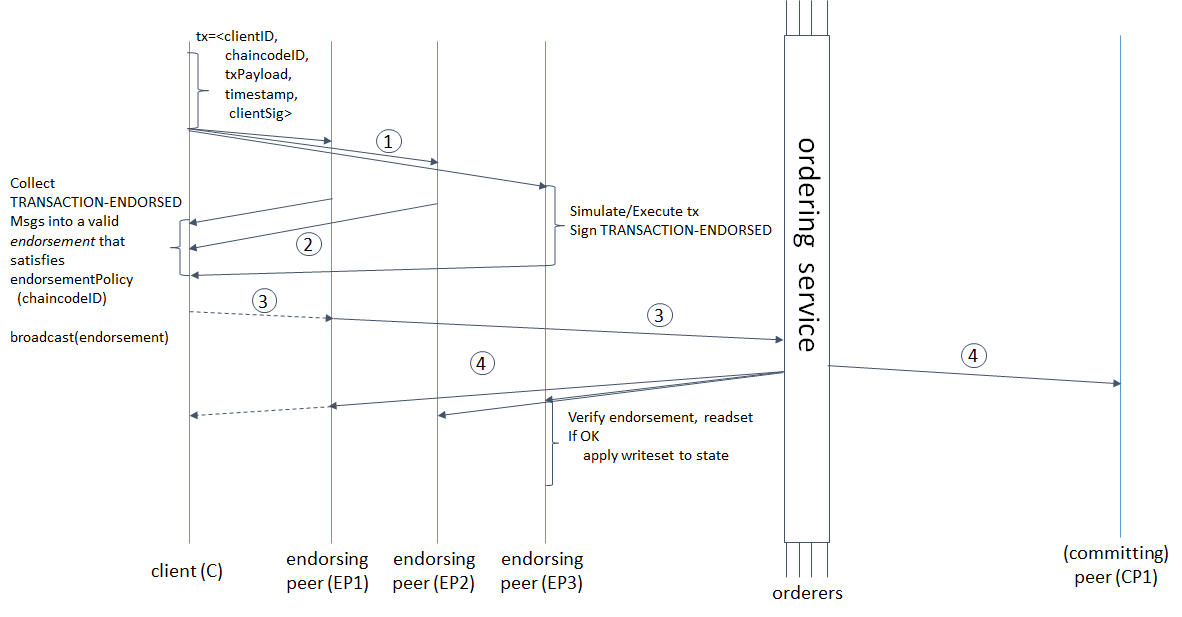
When an event deliver(seqno, prevhash, blob) occurs and a peer has applied all state updates for blobs with sequence number lower than seqno, a peer does the following:

1. It checks that the blob.endorsement is valid according to the policy of the chaincode (blob.tran-proposal.chaincodeID) to which it refers.
2. In a typical case, it also verifies that the dependencies (blob.endorsement.tran-proposal.readset) have not been violated meanwhile. In more complex use cases, tran-proposal fields in endorsement may differ and in this case endorsement policy specifies how the state evolves.

Verification of dependencies can be implemented in different ways, according to a consistency property or “isolation guarantee” that is chosen for the state updates. **Serializability** is a default isolation guarantee, unless chaincode endorsement policy specifies a different one. Serializability can be provided by requiring the version associated with *every* key in the readsetto be equal to that key’s version in the state, and rejecting transactions that do not satisfy this requirement.

* If all these checks pass, the transaction is deemed *valid* or *committed*. In this case, the peer marks the transaction with 1 in the bitmask of the PeerLedger, appliesblob.endorsement.tran-proposal.writeset to blockchain state (if tran-proposals are the same, otherwise endorsement policy logic defines the function that takes blob.endorsement).
* If the endorsement policy verification of blob.endorsement fails, the transaction is invalid and the peer marks the transaction with 0 in the bitmask of the PeerLedger. It is important to note that invalid transactions do not change the state.

Note that this is sufficient to have all (correct) peers have the same state after processing a deliver event (block) with a given sequence number. Namely, by the guarantees of the ordering service, all correct peers will receive an identical sequence of deliver(seqno, prevhash, blob)events. As the evaluation of the endorsement policy and evaluation of version dependencies in readset are deterministic, all correct peers will also come to the same conclusion whether a transaction contained in a blob is valid. Hence, all peers commit and apply the same sequence of transactions and update their state in the same way.



## 6.2 Process to be delivered in current model.

**Customer**

1. Orderer
   1. Once information is put into the network from customer app, the ledger is updates with asset information. For founder being us, it will be genesis node.
   2. Once current condition is updated by a patient from app, a block is created.
   3. Payment for amount in t3 is made.
2. Endorser
   1. The endorser checks with Insurance company if the policy details match and initiates this transaction t1.
   2. The endorser checks with Councellor with current condition, for possible counsellors and initiate transaction t2.
   3. Claim settlement transaction t4 for amount paid is initiated.
3. Committer
   1. Once consensus is achieved from Insurance Company for valid policy and customer details the Ledger is updated for t1.
   2. Once counsellors are received based on decided counsellor Bill is committed on Ledger with 0.
   3. Payment is confirmed after consensus from Doctor/Hospital and Patient.
4. CC

Will include chain code for above transaction flows and asset Initialization.

**Councellor**

1. Ordered
   1. Patient’s medical condition is obtained through t2.
2. Endorser
   1. Check for counselor based on patient’s policy.
3. Committer
   1. Commit list of Counsellors on Ledger for t2.
4. CC
5. Will include chain code for prescriptive algorithm to generate list on Counsellors based on condition and insurance plan.
6. Chain code for update into data warehouse for patient.

**Insurance Company**

1. Orderer
   1. Information of customer and policy are received through t1.
   2. Information of bill is received by insurance company for claim made t3.
2. Endorser
   1. The insurance company validates and send rejection/acceptance or patient through t1.
   2. The Insurance company based on policies decide final insured amount on t3.
3. Committer
   1. Insurance details are committed based on acceptance/rejection on t1.
   2. Available credits, CreditsCovered,AmountUncovered are updated for t3.
4. CC
5. Chain code for the transactions and computation of bill.
6. We can use nltk for policy guidelines and compute fields covered and non covered for insurance on bill by creating labels for bill category and compute percentage covered.

**Doctor and Hospital**

1. Orderer
   1. Once patient decided to opt for a counselor based on advised counsellor.
   2. Payment settlement is received for t4.
2. Endorser
   1. Patients final bill is entered and a new block for transaction t3 is initiated.
   2. Rest service to show complete transaction on existing infrastructure based on amount received through t4.
3. Committer
   1. Based on Amount uncovered final bill is generated for t3.
   2. Patient Bill Report is closed and committed as hash in ledger for t4..
4. CC
5. Chain code for the transactions and computation of bill.

## 6.3 Rest Proxy to CouchDB

As the data warehouse is implemented in CouchDB the, the transaction on Ledger will be time-stamped into the warehouse through Rest services.

No separate field for user medical history needs to be maintained as it will be a fetch to previous Node on the network.

8.  Size and Performance

The blockchain network lives in a state of consensus, one that automatically checks in with itself every ten minutes.  A kind of self-auditing ecosystem of a digital value, the network reconciles every transaction that happens in ten-minute intervals. Each group of these transactions is referred to as a “block”. Two important properties result from this:

1. Transparency data is embedded within the network as a whole, by definition it is public.
2. It cannot be corrupted altering any unit of information on the blockchain would mean using a huge amount of computing power to override the entire network.

Node:(computer connected to the blockchain network using a client that performs the task of validating and relaying transactions) gets a copy of the blockchain, which gets downloaded automatically upon joining the blockchain network.

9.  Quality

Here are five ways blockchain can benefit healthcare:

**1.) Single, longitudinal patient records**

Longitudinal patient records- compiling episodes, disease registries, lab results, treatments- can be achieved through blockchain, including inpatient, ambulatory and wearable data- assisting providers in coming up with better ways of delivering care.

**2.) Master patient indices**

Often when dealing with healthcare data, records get mismatched or duplicated. Also, different EHRs have a different schema for every single field- coming up with different ways of entering and manipulating the simplest of data sets. With blockchain, the entire data set is hashed to a ledger, and not just the primary key. The user would look for the address- there can be multiple addresses and multiple keys, but they will all yield to a single patient identification.

**3.) Claims adjudication**

Since blockchain works on a validation-based exchange, the claims can be automatically verified where the network agrees upon the way a contract is executed. Also, since there is no central authority, there would be fewer errors or frauds.

**4.) Supply chain management**

[Blockchain](https://hitconsultant.net/tag/healthcare-blockchain/)-based contracts can assist healthcare organizations in monitoring supply-demand cycles through its entire lifecycle- how is the transaction taking place, whether the contract is successful, or if there are any delays.

**5.) Interoperability**

Interoperability, the very promise of blockchain, can be realized by the use of sophisticated APIs to make [EHR](https://hitconsultant.net/category/emr-ehr/)interoperability and data storage a reliable process. With blockchain network being shared with authorized providers in a secure and standardized way, that would eliminate the cost and burden associated with data reconciliation.

Other than these, blockchain can transform revenue cycle management, drug supply management, clinical trials and prevent frauds.