:IRYO

Global participatory healthcare ecosystem.

Technical whitepaper v1.04 ©Iryo 2017

Iryo Network technical whitepaper

Iryo Network is a zero-knowledge health record storage platform, with an anonymous query interface. It uses blockchain permission controls for patient record access and tokens to incentivize end users consent enabling artificial intelligence (AI) research.

Table of contents

Abstract

Patient health records are considered one of the most sensitive kinds of personal information. Should personal health data leak, it could be used to defame an individual's reputation, jeopardize employment options, influence insurance premiums, or used as a marketing tool - all potentially exploiting an individual's health status.

However, there is little doubt that having medical data readily available to appropriate institutions globally can have major medical benefits. Access to this data can avoid unnecessary medical complications that arise due to incomplete patient medical history. Currently, many research artificial intelligence platforms are exploring ways to access patient data to improve the understanding of conditions, in order to formulate algorithms for early detection of diseases and develop new treatments.

Due to these perspectives being perceived as mutually exclusive, it is clear why users have erred on the side on caution resulting in medical data only being served within tightly controlled silos. By effectively limiting potential of health data, current health records don't empower their owners and frustrate artificial intelligence (AI) research.

> *Iryo is the first health record project which challenges this perception by tapping into the benefits of health data while ensuring data privacy.*

This solution consists of open access to the platform (API), open-source client code and, by using open standards ensure interoperability, transparency and security.

OpenEHR

According to a study published by Dell's, the healthcare industry is expected to generate more than 500 exabytes of data with an expected annual growth rate of 48%. This presents a looming challenge in data management. Although multiple standards try to address this issue, a lot of that data is still stored inside local silos in proprietary formats. As such reusability of the data and interoperability between different actors is often too expensive or even impossible.

To make our data as open and as meaningful as possible we decided to use openEHR's approach to data modeling and exchange. At the core of openEHR are simple and exchangeable archetypes that link values to their actual meaning (blood pressure as an example). Simple and widely used archetypes can then be linked together in more complex structures to support various types of procedures required by clinics.

Archetypes don't only solve data storage problems but are also used in openEHR's Archetype Querying Language (AQL) where archetypes can be reused in building and running extensive queries across the data.

The openEHR community (in collaboration with doctors and clinicians) have been preparing specifications and collecting archetypes for the last 15 years and have already been chosen as a level of standard in nation-wide data exchange programs in some European Union countries. Taking this into consideration, we deem it the best option to manage patient data with vendor independence by using openEHR.

http://www.openehr.org

Zero-knowledge storage

The Iryo Network is a global repository of openEHR data. Since few people are prepared to provide their medical data to a "GoogleEHR"-type of capture and shameless reaping of all the medical data for commercial purposes, Iryo has decided to give up it's access to plain data. Iryo perceives the medical data it holds as a "toxic asset", because we believe that holding too much data in one place presents too large a liability risk.

The solution to managing this risk is zero-knowledge data storage which is resistant to all attacks, including state-actors or "inside jobs". This works by way of users encrypting their data on their mobile device(s) with a public key.

A private decryption key remains on the patient's device. Whenever someone wants to access patient data (a doctor or researcher, for example) the patient has to approve their access. This will be done by the patient clicking "yes" in their IryoEHR app. This gives a re-encryption key to the doctor's public key. You can read more under the "Private key management section" to understand the details of this process and the application to the edge cases.

Copies of encrypted health records are stored on three geographically and managerially redundant storage nodes.

1 One encrypted backup copy stays on IRYO cloud node.

This is the default backup location that can be changed by a clinic or end-user to point to another storage API. The Iryo offering is centralized in the cloud with tight provisioning controls.

Pros: audited, maintained, secured and backed-up.

Cons: centralized.

2 A second encrypted copy stays in the home clinic storage node.

Pros: local copy, the clinic doesn't need to rely on an internet connection and it is fairly distributed.

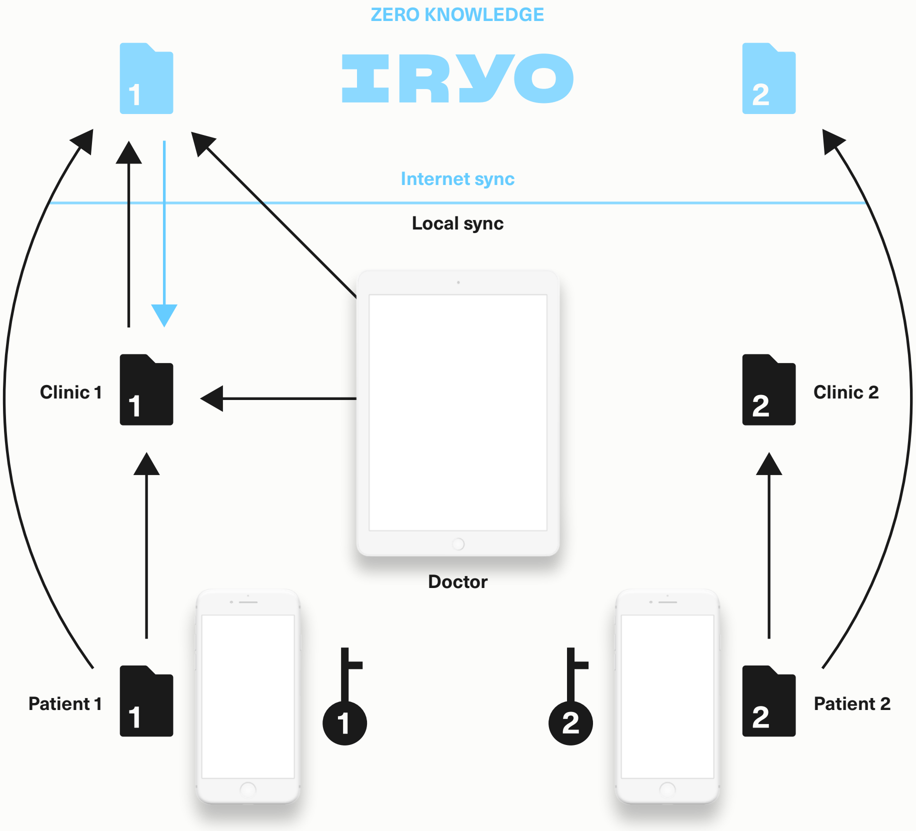
Cons: clinics' IT personnel lack specialization in secure deployment.

3 End-user devices (phones) distributed all over the world.

Pros: decentralized (not centrally controlled) and protected by thousands of people at the same time.

Cons: not enough space for all raw data, malware infected devices and old/lost or stolen devices.

Whenever data on end-user devices (POINT THREE ABOVE) is updated, the other devices would connect to the API of both redundant storage nodes (POINTS ONE AND TWO ABOVE) and sync/update the encrypted data to match the local copy. Both storage nodes would provide a "blockchain proof" (cryptographic receipt) of the location of the data saved with the same hash that clients requested. Clients would validate these by asking the independent node if the data was actually put in a chain.



If the device contains more current data (which could happen when a doctor syncs health record with a more recent version), then it would only connect to one endpoint API. This would be one that is reachable - preferably the local one (POINT TWO ABOVE) in the same clinic. In this manner, read access doesn't consume hospital internet connection.

The diversity of network topologies and endpoint reachability would allow clinics to operate even if their local network was down (as long as they can find emergency hotspot). This greatly reduces risks of access outages which could have fatal consequences.

While there could be a complete loss of data at each of these points independently, when working in unison they provide reliable and robust system redundancy.

Risks in distributed data storage systems like Filecoin, Sia, Storj, Maidsafe

The problem with distributed systems like Filecoin/Sia/Storj/MaidSafe is that they can't protect users from attackers storing and serving all data from the single server. Attackers can pretend to be geographically distributed and collect money for all 3-5 copies (Sybil attack). Trusting devices, hospitals, and the Iryo Network to keep at least one (encrypted) copy alive offers far greater guarantees against health data loss.

Who pays for the storage in Iryo network?

All text-based data will be funded by clinics who would stake IRYO tokens and the 1% yearly inflation would be partly used to cover the cost of storage on the Iryo platform.

In some cases, in order to secure the storage of additional gigabytes of raw data being generated and stored, clinics would have to stake additional tokens. If patient-users do not want to be dependent on the staking decisions of their clinics, they would have the option to stake coins themselves. This would unlock the storage for their use, and limit potential abusers.

The precise staking requirement would be updated based on the real data gathered when the Iryo Network goes live. We foresee that, at scale, this would be significantly cheaper than any current decentralized storage, especially in comparison to proprietary systems that cannot be easily upgraded.

The Iryo Network has distanced itself from the current fixation on "Big Data" and has chosen to rather focus on patient privacy. This alternative focus would allow the Iryo Network to scale globally, enabling it to attract more users. Because users, clinics and governments are assured data security and privacy, storing data on the Iryo platform means that the number of users willing to share their data with researchers could increase the rate of current EHR participation significantly. Open-source end-user apps would ensure that there are no secret back-doors circumventing the protection.

Anonymous query interface

enables AI learning over distributed & encrypted data

Since health data doesn't decrypt itself without patient consent, a new approach needs to be devised to allow for machine learning and AI capabilities. There are many complicated ways, from both a development and resource standpoint, to query encrypted data (e.g. multi-party computation and proxy re-encryption).

Fortunately, there is an actual 'trusted device' in the Iryo Network end-user device. This could be a phone, tablet or personal computer. Since these devices need to be able to read all health data in plain text, they would also be able to execute the queries across the same data.

A process to deliver queries to end-user without breaking its anonymity or given consent needs to be defined. Iryo has a solution to this process gap.

1. First Iryo would verify research institutions to make sure they are legitimate and not attempting to commercialize confidential information by re-selling the collected data.

2. Researchers would receive the Iryo Research Portal software which they can use to send queries to the Iryo Network, using the 'Archetype Query Language' (AQL), and openEHR specification.

3. When they do, Iryo would verify the query first. This is to check that the query is not too broad or asking for information repetitively which could indicate an attempt to reconstruct records (if done over an extended period of time).

4. The patient's own device verifies that the patient meets the query criteria. If the verification is successful, the query details with the name of the research institution and the amount of tokens to be received by a patient is shown on the device pending approval.

In actual implementation, the patient's device will receive a silent notification which will wake up a background process to query the requested criteria i.e. female, 30-35 years old with diabetes. If a patient does not fall within the defined parameters, the silent notification disappears. It will do so without providing a report to the requester thereby keeping patient-users anonymous. If the patient meets the criteria, a notification would be shown on the patient's device. The notification would include the name of the research institution, the justification for the query requested i.e. the aim(s) of the research, and the number of tokens available as an incentive to allow query results to be sent back.

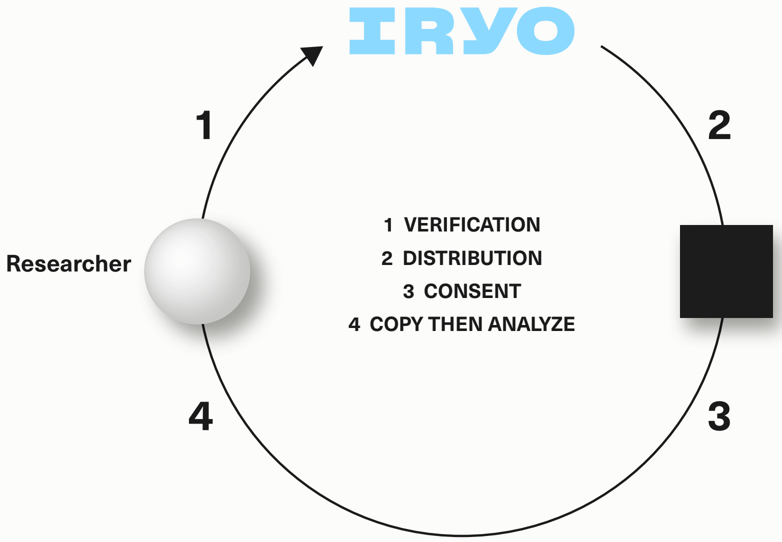
Iryo envisions three types of opt-out, anonymous requests that present various potential implications for privacy which would require distinct user consent. These types are identified as a pseudo anonymous query, an anonymous query used for AI validation across a dataset and an anonymous query to deliver patient value.

Copy then analyse

PseudoAnonymous query – used for AI training dataset

This is a request for medical data in plain form, without the directly identifiable personal information (pseudonymous). This bears high costs (in the region of $100 worth of IRYO tokens) since, even without personally identifiable information, this data can still be used to match against other databases and individuals could be identified if that data leaks from the researcher*.

The number of these requests should be kept low (up to 100 patients) to train and test machine learning algorithms freely. After results are determined and the algorithm needs to be validated (or invalidated) over much bigger population sample size, they proceed to the next type of query.



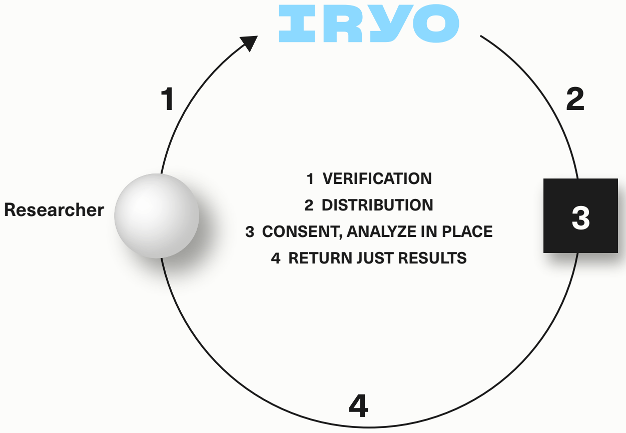
* Privacy for research participants

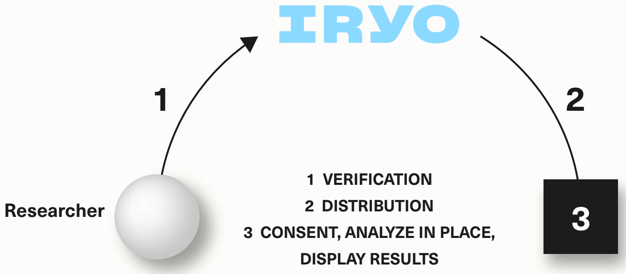
Analyse in place

Anonymous query – used for AI validation dataset

This is an anonymous query across medical data that bears a very low cost (estimated at $0.1 worth of IRYO tokens). This is because this request takes data, applies a formula to it, and then only sends back the result in numerical or binary form (true/false). There is no leakage of personal information and it cannot be compared to other databases.

These requests can be sent to all users in the Iryo Network and can be used to train and test machine learning algorithms, until a verified researcher finds and validates the missing link.





Anonymous query – used to provide patient value that saves lives

This is an anonymous query across medical data that does not return binary responses, but rather shows actionable results. Its philosophy is that researchers should/could share the actionable algorithms that they have validated to improve decision making, for example. Of course, to avoid health risks, algorithms need to be validated by a regulatory authority like the Food and Drug Administration (FDA) or European Medicines Agency (EMA). This must be done before the insights derived from this platform can be considered as accepted medical advice (i.e. clinically verified) and is offered to Iryo patients.

Benefits for researchers

Researchers will have access to larger research populations which would result in more robust research results. They will be able to use the Iryo Research Portal to enroll people with specific health conditions as determined by parameters selected by them. Direct access to EHRs of specifically defined patients can decrease the time required for patient recruitment, thereby potentially decreasing pre-recruitment process costs. Typically, recruitment agencies and services need to first attract potential patients and then check their eligibility. This can be a lot more time consuming than a query in the Iryo Network. The simplification of the research process is outlined in the steps below:

Obtain 100* individuals who respond to the query, and collect their health data.

1. Use machine learning (AI) to identify trends and test hypotheses.
2. Verify learned formulas on 10.000* more people with anonymous query, without ever exposing their information.

Since the Iryo Network scales better to more patients internationally (due to its inherent privacy), researchers would be able to approach more people and therefore enhance the robustness of their research findings.

Benefits for Patients - find actionable early indicators of health problems/diseases

Patients will be alerted to their identified modifiable risk factors of disease and indicators that may suggest the early onset of disease. Once correlations are clinically verified, users would get the anonymous queries that would seep through their data and present them with actionable advice. For example, they will receive information on which healthcare provider they need to consult and which tests would be beneficial for them to have. Those queries would not be reported back, and would remain on the patient's phone.

* fictitious values used for illustrative purposes

With Iryo's innovative design, patients now have the option to not share their health information but still receive research results – something that has not been achieved until now.

If the patient (older or non-technical) does not possess a smartphone or the IryoEHR app, they could give permission to their doctor to allow research to be conducted on their data after an approval processes at the clinic.

(public) Blockchain

Blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptographic algorithms. Each block typically contains a hash (a link to a previous block), a timestamp as well as transaction data. Full nodes validate all the transactions, but they can't settle the disagreements in which order they receive them. To prevent double-spending, the entire network needs to reach global consensus on the transaction order. It achieves this by using centralized parties or a decentralized proof of work or proof of stake algorithm (and its derivatives).

It's important to understand that your local blockchain node won't allow anything that is not "valid" (pre-defined consensus rules), even if the longest chain says otherwise.

Contrary to popular belief due to aggressive marketing, blockchains are not a good solution for storing data. Each piece of information that you store in the blockchain sits in hundreds or more nodes (more than 100 000 in case of the Bitcoin), making it very costly. This is why the Iryo Network doesn't store data on blockchain but uses blockchain to ensure the transparency of transactions.

Some projects pretend to be using blockchain by using 'private chains' which are usually just re-branded databases. Private chains use some elements of blockchain technology but miss key elements thereof like the oversight offered over the validity of the stored data.

Public blockchains are mainly useful for two things; value transfer (including initial creation and distribution) and trustless timestamping of the messages.

Blockchain permission controls

End-users can issue signed permission messages to the blockchain. This includes:

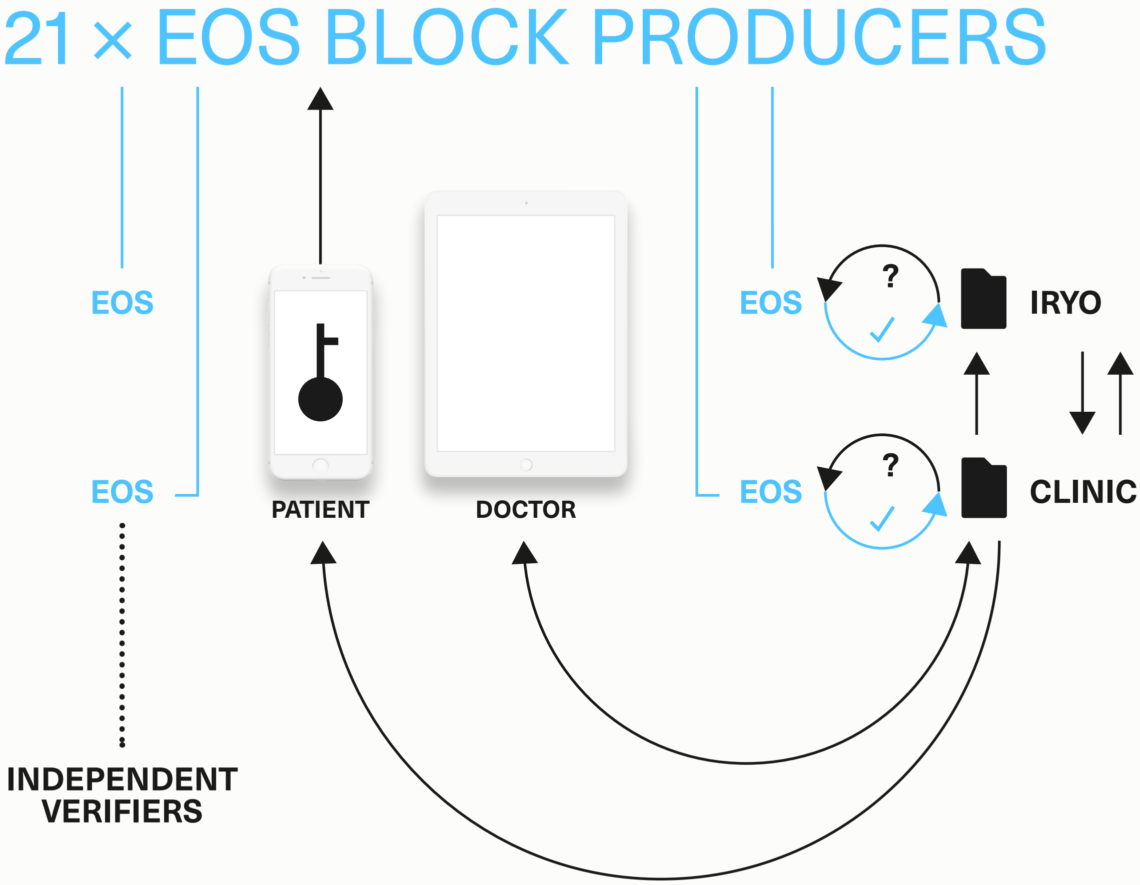
- sharing all electronic health records with their personal doctor (with or without time limits)
- the revocation of access to their EHR
- the sharing of limited relevant parts of health records with a specialist (with a time limit). Inpatient medical records will also be partitioned with a separate set of access rules
- issuing IoT device permissions to write but not to read data
- rotating the re-encryption key

The use of the smart contracts for permission controls is optional (there is no double spending problem). Iryo would use it, because that way the blockchain node can compute the state at the moment it received the ordered message, instead of waiting for the query to compute it i.e. where the state gets pre-computed.

Permission controls benefit from being in the blockchain in the following ways:

- *Trustless timestamping.* All Iryo full nodes, including consensus "block-producing" nodes, would validate all messages. Therefore there will not be any doubt with regards to what and when blockchain permission messages were issued.
- *Immutable log of all messages.* It will not be possible to deny that a valid access request was issued. This will reduce the possibility of litigation where dishonest parties claim that access permissions were never issued.
- Once your message gets included in the blockchain, *one part of the network cannot withhold it from another*. That way, users can always be sure that they have all the revocations, that is, assuming they are running the Iryo full node (which is still just a tiny subset of the whole EOS network).

All the storage endpoints (in the cloud and in the clinics) would run their independent Iryo ledger full node. A smart contract would be used to prepare the local state. Direct query on the full node would return either true or false for each access request the Iryo (storage) network needs to process.



Simplified overview

Immutability of medical data

The first line of defense is offered by zero-knowledge encryption itself. It is easier to defend data integrity when a potential attacker doesn't know what to change since everything is encrypted in the first place.

The second line is offered by redundant storage nodes and saved medical data checksums on patient devices support immutability. If anything is changed, a user will know.

The final line of defense is to find out which node was changed. All storage nodes would provide cryptographic proofs to patients, by writing hashes in the EOS blockchain. Patients would be able to independently verify that the provided proof was really there with a blockchain receipt. That way, if the checksum verification fails, the compromised storage node can be easily identified and replaced.

To reduce the number of hashes, aggregation into a Merkle tree will be used. Clients receive a blockchain receipt which they can use to independently verify the blockchain proof*†.

* www.chainpoint.org

† www.opentimestamps.org

Why EOS?

Choosing the right platform, EOS overview.

Generally speaking, there are two extremes in public blockchain technology:

1. using a proof of work (PoW) chain that offers immutability and censorship resistance, but you pay for that with fees for every transaction.
2. using federation/DPOS/masternodes that offers speed with low associated cost.

Most tokens on these platforms have a centralized team behind them, which means that if their token contract is compromised, they should be able to roll it back. Alternatively, this team could just issue a new contract taking into consideration factors like pre-hack balances.

Having a token that is solidified with PoW presents additional liability if something goes wrong. This is not a feature that a startup raising capital should go for. This option should only be considered once the code has been tested over time.

Ethereum sits in a thankless middle. It does not allow rolling back your app, since projects don't really control their smart contract. Furthermore, users still need to compete for limited block space and bid with gas or often just wait their turn (up to a day).
For many token issuers that is just not needed.

The token issuer requires just enough decentralization for the market to perceive it as decentralized and it's history to be independently verifiable (many/enough nodes that check it). Of course, they also need tokens to be easily listed on the exchanges - once the platform is added, the addition of further tokens should not require significant effort.

EOS will offer "shards" (they call it blockchain apps), which users would be able to utilise on their partitioned network - user nodes can discard all other messages, making them as light as the chain is designed to be. A lightly designed blockchain app would have the ability to spread more nodes since it's easy enough to verify, even if the entire EOS network gets "bloated".

Instead of transaction fees, EOS chooses a smarter way to charge its users. Every blockchain app (shard) must deposit coins for their users. Practically, 1% of locked coins means that users of this app can get 1% of all bandwidth of block producers... and until your app only uses 1% of the whole EOS ecosystem, all transactions are free for end-users.

That way new users don't need to pay for anything. However, the blockchain app developer still has to mind the resources, or else the requirement to buy more and more EOS tokens to stake would 'kill' his blockchain app'.

Block-producers dilute this stake with up to 5% per year inflation, to pay for investment into additional bandwidth, space or execution/verification CPU power. This presents one potential problem: whenever block-producers upgrade their equipment, the number of tokens that the blockchain app requires, falls. This is because those EOS tokens are no longer needed since people may sell them on an open market and crash the price.

Since token holders vote for block-producers, they might start voting for those who don't want to upgrade in order to keep the staking requirements high. It would be interesting to see how the tension that would arise due to, one the one hand, helping app developers with adoption and, on the other hand, the high token price would play out.

Application developers will need to pay for the cost of EOS user accounts. Let's hope the price for them won't skyrocket; allowing competition to spam apps with fake accounts, thereby draining the funding for the blockchain app and it's legitimate users.

Another strength of EOS is its use of WebAssembly instead of Solidity which ensures C++ toolchain compatibility.

In addition to that, Iryo would use the public EOS network to keep access control timestamped and synced on all nodes. Iryo deems the importance of access control messages and history logs to be propagated near real-time in emergency situations as essential. Right now, this cannot be achieved/guaranteed on a network like Ethereum or Bitcoin.

Token utility

Iryo would use EOS chain to intertwine its tokens in a number of ways:

1 **All institutions would have to provide a stake of $10 000 worth of IRYO tokens for their accounts (this value is adjustable).**
This would serve as spam protection - the app won't 'talk' to 1 000 fake institutions that do not have IRYO tokens who would most likely be attempting to spam users. It would also act as a transparent metric on the chain. More accounts with enough tokens should mean more institutions are using the system. Institutions would include organisations such as hospitals, clinics and research institutes.

2 **The clinic staking requirement would be used to cover the cost of storing EHR data for their patients.**
Should the data per patient exceed the threshold, clinics would have to stake more tokens to cover the cost of data storage. Patients would be able to cover their storage costs by staking coins themselves; that way they can become independent in storing as much data as they want to (as long as a sufficient number of coins are staked). If they exceed the limit, or the staking requirements suddenly change, they would not lose the data but, over time, their access would be increasingly limited until the stake is supplied. Actual hardware costs would be covered with 1% yearly inflation. Therefore, more data stored would mean more coins staked, which, in turn, provide price pressure to make that 1% yearly inflation worth enough to cover all the storage cost.

3 **With health record query tokens researchers would be able to incentivize end users to allow anonymized queries.**
Health data never leaves the patient's device (phone), or the device of the doctor whom the patient has assigned access privileges to. Researchers would have to buy the tokens from the market and distribute them to the users that allowed the queries on their health data to be executed. The amounts can be very small, and sent to thousands of people at the same time. High fees could kill this model.

4 **Cases of medical emergency**

When a patient can't give consent for access to his health data, the hospital can lock $1000 - worth of tokens (adjustable) in the smart contract which gives the patient permission to withdraw the tokens in one-month's time if he deemed access unjustified. In case of a legitimate emergency access, the patient (or their doctor with pre-approved access to the patient's medical records) would, when able, confirm the emergency access and the smart contract would return the funds to the hospital. If no action is taken within one month (or other specified timeframe), the funds are returned to the institution that staked them.

5 **Services in the clinics who have adopted Iryo could be paid with IRYO tokens instead of credit cards.**

Volatility, usability challenges with the security of tokens and the limited ability of the end user to purchase tokens on short notice would probably results in this option being used infrequently. With mature adoption, we could see tokens being used in situations where transaction costs using traditional payment methods are prohibitively high.

The aforementioned 1% yearly inflation would cover the storage costs and offer the development subsidy to ensure continuous Iryo Network development and provide a sustainable future for IRYO token holders, even when the initial funds run out. Until the platform is publicly available, the inflation tokens would be burned.

The 1% inflation recipient would be a multisignature account controlled by Iryo. Iryo would issue reports on how this money was spent. Should the token holders not be pleased with the development of the service, they would be able to fork the token contract to a new one, which doesn't have a development subsidy or has a different group who collects the inflation.

Cryptographic enforcement of permissions

over data using Proxy re-encryption (NuCypher)

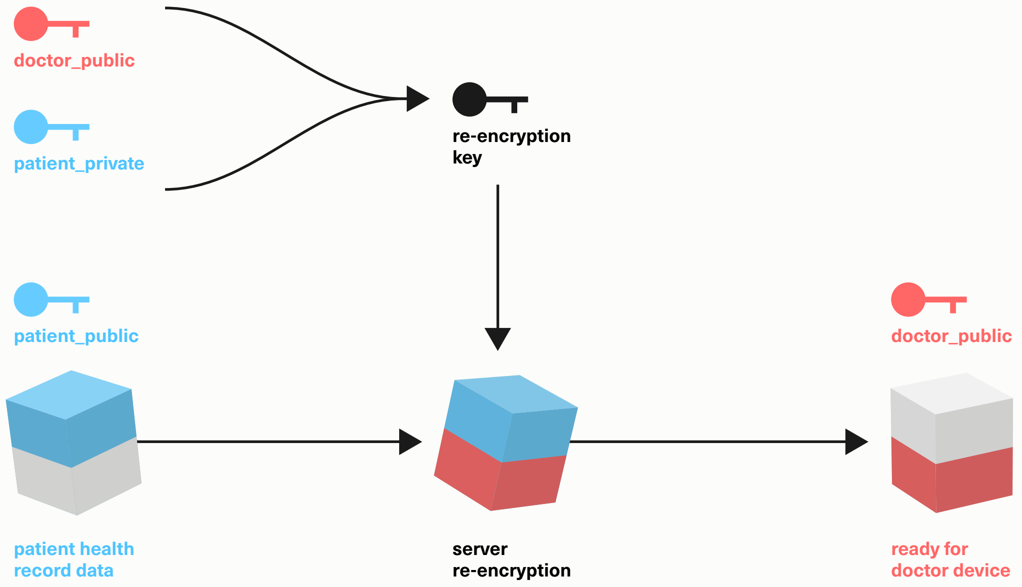
Public key cryptography (PKC) also known as asymmetric cryptography uses a key-pair of public-private keys to encrypt and decrypt data. This property gives the user the ability to share public keys outside of his control. PKC based on an elliptic curve is mostly used for digital data signing but it can be also used for data encryption.

ECIES encryption allows one to asymmetrically encrypt data with a public key and decrypt it using private key, instead of the typical two-way symmetric AES cipher which uses one key for encryption and decryption. This is done by creating an ephemeral random key based on the receiver's public key. Data is then symmetrically encrypted using this ephemeral key.

This key is then cryptographically encrypted so that only the owner of the private key that corresponds to the public key can decrypt it. The encrypted data, along with encrypted ephemeral key is sent to the receiver. This means that the patient can share his public key with anybody to give them write-only permission.

ECIES (patient_public, data)

To read the data, an Umbral algorithm is used, which allows patient to issue re-encryption keys. These keys can take the data and re-encrypt them to a doctor's public key on the fly. To create a re-encryption key, patient_private, doctor_public keys are used. A re-encryption key is then used to re-encrypt the data (encrypted ephemeral key) from the patient to doctor public key.



Iryo storage nodes can hold patient issued, re-encryption keys, and re-encrypt all patient data to doctor on the fly. Since the data gets transmuted on the server, Iryo only needs to keep one copy of the same data while keeping it zero-knowledge.

In reality, the ECIES encryption is very slow. This is why NuCypher uses hybrid encryption scheme. The data is first encrypted with temporary key using AES symmetric cipher and this key is then encrypted with sender's public key using ECIES. Re-encryption is then done only over an encrypted temp key rather than whole data. Their whitepaper provides a more detailed explanation.

NuCypher is building a decentralized service but, since Iryo has more trusted topology, it can achieve the same (or better) security properties when deploying the re-encryption software from the two storage nodes from which each user has their data served from.

Public key derivation (BIP32)

For the sake of simplicity of the paper, we only mention one encryption key-pair. In reality, the keys are deterministically derived from master private key. Separate child keys are used for different access permission levels. Each public child key would get separate re-encryption key. This way the patient would have mathematical control of the level of permissions each doctor/specialist would get. Youbase's whitepaper provides a good example of that model.

Key rotation

When access is revoked, a patient would issue new re-encryption keys for all new data. For old data, NuCypher proxy re-encryption relays on the servers to not serve the revoked clients (with data or re-encryption keys). In our case, the Iryo storage node and clinic storage node would simply throw the revoked re-encryption keys away and, by doing this, they protect the data even if the storage node gets hacked later, and all encrypted data leaked.

Due to BIP32 and proxy re-encryption, the key rotation process does not need any other device but the patient's to be online (it's non-interactive). This value is enhanced in the doctor's case, as he can re-issue all re-encryption keys to himself without 'bothering' any of their patients.

Private key management

Private keys are long and random and should never leave the end-user's device. Should encrypted data be leaked to the public, it would be useless without the private key. The value of private keys lie in the decentralized control of encrypted data and value.

However, if you lose your keys, you lose the data, credentials and value that it encrypted. It is for this reason that encrypted services provide a nerve-wrecking experience for most people. Some services are trying to solve this problem with recovery codes that should be printed out and put into a safe drawer. It is safe to say that there is no mobile printer on your phones, and there is no mobile safe drawer. Most people don't have "safe drawers", not even in their homes. In reality, this approach is not practical for the user and rather has a stronger role in providing the service provider with immunity against legal and reputational liability.

While many other projects re-introduce centralized solutions for key recovery, the Iryo Network takes a different, more distributed route.

In the Iryo Network private keys are everywhere. These keys can be grouped into patient keys, doctor keys, clinic keys, Iryo keys and token holder keys.

Patient keys:

PK-1: A master private key that derives two keys (using BIP32) which are used for:

- medical data encryption keys in the IryoEHR app
- EOS 'emergency access controls' and wallet private keys in the IryoEHR app

Doctor keys:

DK-1: A master private key that derives 2 keys (using BIP32), that are used for:

- medical data encryption key in the IryoEHR app
- EOS 'access controls' and wallet private keys in the IryoEHR app

Clinic keys:

CK-1: Half of the emergency medical data re-encryption key. One for each patient.

CK-2: An EOS 'emergency access controls' and wallet private key.

CK-3 (optional): A cloud backup AES encryption key that allows clinics to encrypt and then backup all their data to Iryo Network.

Iryo keys:

IK-1: An EOS attestation key.

IK-2: An Iryo EOS (account funding) wallet private key.

Token holder keys:

TK-1: An EOS wallet private key for EOS wallet app.

What happens if a user loses the device?

For *PK-1*, assuming that the patient doesn't have a second device with the same key, (patient medical data encryption key), the simplest answer is that they can visit their doctor, who can use his device to issue a re-encryption key back to patients' new device. Together with signed permission message, this would replace patient's wallet private key with a new one. If a patient doesn't want to visit his doctor every time their device gets destroyed, they can save (and move) the key to the ZeroPass app (explained later in this paper) using a one-click magnet link. When keys are protected with ZeroPass, a patient can revoke the ability of his doctor to re-assign his key, leaving the patient in full control.

If the patient's personal doctor performs an 'access recovery' instead of recovering a master key via ZeroPass, the patient's wallet would be emptied. This is because the doctor can't recover the actual keys but can only give access to patient's new key. Using ZeroPass solves this problem. Whenever a patient receives an IRYO token he will be asked to secure them within the ZeroPass web of trust.

For *PK-1* and *TK-1* the ZeroPass' distributed and trustless recovery service can be used. In practice, patients would simply click on the magnet link inside the IryoEHR app, that would save (and move) all his/her keys; *PK-1* and *DK-1* from IryoEHR to the ZeroPass app, automatically.

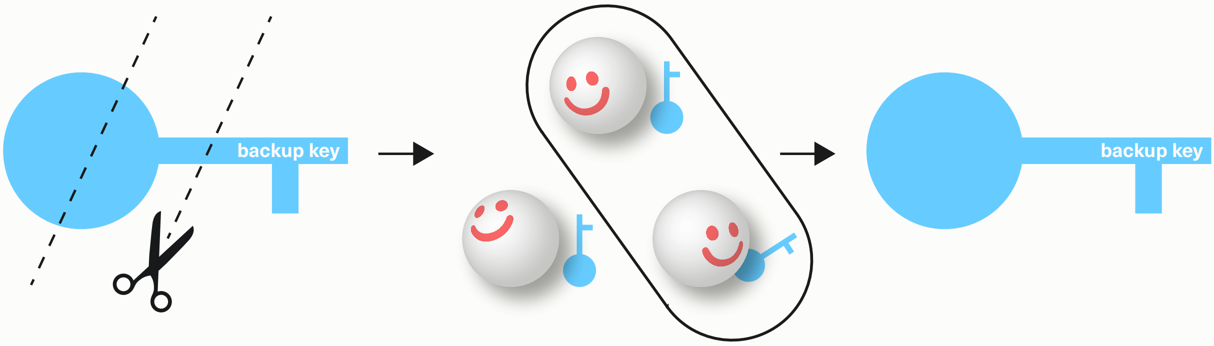
For clinic and Iryo keys (*CK-1,2,3* & *IK-1,2,3*) the ZeroPass 4Teams app would be used.

ZeroPass app

ZeroPass is a passwordless keychain and private key recovery manager. It can store keys in a zero-knowledge manner and provides trustless recovery that works even if ZeroPass servers are down. The development of the ZeroPass app is currently in the private beta stage. For an in-depth understanding of this app, you can read more on the ZeroPass website or fairly technical whitepaper.

To use a key from ZeroPass users don't need to remember anything; instead they need to sign every request for a requested key with two devices. These devices must have previously been paired with the ZeroPass app or, optionally, via YubiKey-style devices (FIDO compliant).

Later, ZeroPass would add multisignature transaction signing (thresholdECDSA), which will be used to provide an extremely secure signing process using a private key. *The transaction spending stays secure even if one of the user's devices is infected by malware.* This capability will be tightly integrated into IryoEHR app's token functionality.



If users lose their devices (referred to as ZeroPass "factors") and/or lock themselves out, two out of three trusted contacts which they had nominated can help them recover their key within the app or offline (where the service is unavailable online).

ZeroPass 4Teams app

ZeroPass 4Teams will provide clinics and Iryo multi-member teams environment with all of the core ZeroPass functionalities and benefits, while at the same time also taking into consideration their specific requirements:

- a global overview of all members and their actions,
- access to signing using a private key for multiple users (sharing functionality),
- an adjustable level of security (how many users and/or devices per user are required to sign a transaction or get access to the key).

User-owned devices will be encouraged as the primary and/or secondary factors. This is a ZeroPass answer to provide security in the rising ‘bring your own device trend’ (BYOD). Knowing that 50%+ of all employees use their own devices for work, the trend does not seem to be slowing down.

All described key (and subsequently health data) recovery procedures might be rendered useless in extreme situations, such as long internet outages which place lives in danger. Disaster access modes have been developed as a solution to avoid the potential negative consequences of these situations. Patients or clinics may opt-out of these modes. These are explained in the edge cases below.

Edge case: Emergency access for individual users

SCENARIO: Alice is an unconscious patient who is rushed to the hospital by an ambulance in the middle of the night.

The previous assumptions about a patient or their personal doctor being able to unlock the data can prove fatal in those cases. The solution is to escrow (split with Shamir Secret Sharing) the re-encryption key within the ZeroPass 4Teams app and lock it with the smart contract that is enforced by the ZeroPass server. The smart contract rules state that if the clinic wants to access the (missing half) of the key for patient Alice (identified with her public key), the clinic needs to put $1 000 worth of IRYO tokens (adjustable by Alice) to the 1-month contract with her EOS wallet public key.

Once this is done, Alice’s IryoEHR app will show her the notification. If her medical data access was unjustified she has 1 month to claim the $1 000-worth of tokens. If it was a legitimate life-threatening situation she can either approve the refund to the clinic or wait a month for the contract to expire - this would automatically return all the tokens to the clinic.

To stake the coins for emergency request, the clinic takes the tokens it already has from their wallet ($10.000-worth of IRYO tokens) and has 1 month to replenish the stake if it gets under 'staking requirement' for the clinic. This way they don't need to rush while trying to buy the token in an already stressful situation- it's just a matter of enough people in the clinic approving access. If multiple devices of a clinic were hacked (very low probability), the clinic's stake would be drained and the hack would stop after 10 unauthorized accesses (because the clinic would run out of tokens).

Instead of more than 1 000 stolen electronic health records only up to 10 records can be leaked in the catastrophic event of a data breach.

Potential abuse of this edge case functionality:

- If Alice (patient) abuses the stake; the clinic can invoice her and later file a lawsuit if the invoice was not paid for.
- If Alice's hospital issued fraudulent requests she gets automatically compensated and can sue based on the public blockchain proof of the breach. That holds true even if the clinic refuses to notify her of that catastrophic breach to multiple clinic personnel devices.

Edge case: Disaster 'Health Records' access mode for clinic

ZeroPass 4Teams recovery works the same way as the non-team version; two out of three trusted users can help recover stored keys in the team app. Those trusted users might not have the app themselves and would need other team members to prevent (offline) abuse.

This mode proves to be especially effective in cases where natural or other disasters occur and connection to the internet is cut off or clinics' devices are destroyed. Two out of three ZeroPass 4Teams trusted contacts can meet offline with team members that have the app and recover every single stored key together

These modes are not designed to be extra easy for the user experience because a higher level of coordination prevents hackers from abusing this disaster access mode. Hopefully, disaster access mode for clinics would never be used.

High net-value individuals, heads of the state and other individuals that might be targeted by criminal or nation-state hacking activities (for instance, during a war) can opt-out from the recovery access modes from the start. Individual patients can opt-out as well, as long as they provide consent indicating that they understand the risks and use the ZeroPass private key recovery app to its full extent (with added two out of three trusted contacts. Opting-out should be discouraged by the clinic on an individual level because these fail-safe modes can help save patient lives.

Conclusion

This comprehensive solution for key management is the first of its kind and it provides a strong foundation for the Iryo Network to stay usable while offering unprecedented privacy and security guarantees. These guarantees are not based on high-risk, centralised trust but instead the Iryo Network distributes access to the ‘edge’ of the platform, to people who care the most about the data.

In the spirit of the bitcoin, Iryo Network removes the need for (patient) identity, replaces it with mathematical control over his medical data. This allows Iryo Network to bypass traps associated with stolen identity, phishing attacks and avoids liability with verification of the identity (false positives, false negatives, missing the data on the patient).

Patient > Doctor > Clinic dynamics keep the system honest. Secure and private and allows the system to finally scale over international borders bringing an unprecedented network-effect to fruition. This historically fragmented industry can finally start building on top of open Iryo Network and start serving patients instead of going through yet another round of interoperability nightmares.

iryo.io