The Blockchain and the Universally Global Electronic Health Record

> **Page Under Construction**
>
> This page is still being completed and revised.

This page contains a summary of the technology that is shaping the future of medical health records: blockchain, asymmetric cryptography, hashing algorithms and peer-to-peer networks. I cover the fundamentals helpful for anyone in the medical or dental field.

I Once Had a Dream...

When i was a kid and first heard about computers, and was exposed to them with dad's first experiences with a Commodore C-64 and an IBM PC Convertible 5140, immediately the question was "Dad, what is this? What is it for?". And an all familiar answer came, one we haven't had to explain in a few decades now:

Dad: "it's a computer. It can help write papers, kind of like a typewriter, except i can make corrections before i print the paper, as many as i wish! And it can also calculations for me: like i can keep track of my finances."

Son: "What else can i do with it?"

Dad: "One day, computers will be everywhere and we will store all sort of information in them. We will be able to store our medical history in them. Medical charts. Imagine: a person will be able to go in a hospital, any hospital in the world, they'll just walk in, show their ID, and the staff will be able to access their entire medical history, since they were born. No more ambiguous notes or prescription, it will all be just one".

Son: "Wow. Why? How is it now?"...

Anyways, we've had this dream for a long time now. Computers have been able to make things come true, things we never even dreamed of before. Well, how about things we did dream of before? Some if it became reality, but what happened with this specific medical informatics dream? 30 years have past and it never turned into reality! How come? Certainly many people must have worked on it.

Who Owns a Patient's Clinical Data?

I've personally struggled with this concept for a long time. When i thought about it, the options i would come up seemed to be only 3:

1. Vendors: those who develop the systems that providers use to collect and store clinical data retrieved from patients.
2. Providers: the doctors who provide medical care also by collecting and storing patient's data in systems developed by the vendors.
3. Patients.

At the end, i came to the realization that, even though each one of these parties can have influence on the data itself, none of them are the actual owners of the data: clinical data is a human heritage and should be accessible to everyone.

And here's why none of the above options really convinced me.

Vendors

Providers

Patients

I have heard some stories where a product vendor holds the doctor hostage of clinical data: the doctor might not pay for the vendor's services any longer and the vendor, instead of simply interrupting its actual services, it also decides not to allow access to the clinical data to the doctor any more. My initial reaction to that has always been "hey, that's not ethical: the vendor's don't own that data, it's the practice's (or doctor's) data!". Yes, i feel strongly against this behavior.

Maybe doctors own the data they collect. After all, they have put in the investment of purchasing the hardware and finding the patients and actually collecting the data. But even this never felt quite complete, because what happens when the provider retires and doesn't really do anything with the data they collected? Should all that data get lost with him or her? Decades of hard work gone to waste? That doesn't seem quite right either.

So if the patient owns the data, then they would have to be the ones ultimately responsible for it's integrity and safety. Certainly they could allow others to take care of it for them, however it could be hard to make sure that the entire medical record remains "in one piece". What if the patient accidentally or purposely deletes parts of the medical record?

So after a lot of struggle with this, i decided to tackle the problem starting from first principles. And i came up with this list:

- Everyone wants to suffer less.
- Illness, pain and death are most commonly associated with suffering.
- Humanity is striving to reduce its diseases, physical pains and to elongate its lifetime.
- Medicine has provided ways to fight diseases, reduce pains and increase life expectancy.
- Improvements in the medical field require lots of scientific research.
- Scientific research requires lots of clinical data.

If the above points hold true, then it follows that there is a direct correlation between human suffering and clinical data. Which means that

> **Lots of Clinical Data Help Humanity Suffer Less**
>
> So at this point it became pretty apparent to me that clinical data is a human's heritage and should be therefore available to anyone for research or patient's care purposes, provided the datum consumer is not ill-intended.

Certainly patient and provider should have a say as to who should have access to which part's of the data, and if the data should be able to identify the patient or not. In some cases, it is best if the patient is not aware of some opinions, thoughts or discoveries of the medical provider yet. Similarly, a patient might not want to fully trust a specific medical provider or for some other reason want to just give access to a partial subset of his or her clinical data to a provider. However these configurable permissions should not stop the data from being anonymously, globally and forever available. Besides, isn't this what humanity has always wanted? The ability to just walk into any medical provider's office without anything more than an ID, and allow them to access our medical records in order to provide medical care for us?

What remains now is figuring out a technology that can allow for all this to work. Up until 2008, the software we at our disposal was not able to provide a functional, reliable, secure, scalable solution to the problem. However, the invention of the Bitcoin has introduced the new concept of the blockchain, a technology which theoretically has all the physical capabilities to make the above concept become real. Now, just the fact that the technology is there, doesn't mean the problem has been solved: after all, we also currently have the technology to colonize Mars. That doesn't mean we can colonize Mars without putting in some serious work. And serious work is required in the medical field as well, before we can manage to make this technology become widespread.

Turning the Dream into Reality

If you can agree with me that the medical data is in fact a heritage of humanity, then you might also conclude that it be preserved and available for humanity in the most reliable way possible. It needs to be available for research, it needs to be protected from hackers, it needs to really represent humanity as it is and not be manipulated. If we can imagine an ideal global and universal healthcare network infrastructure, it will fulfill these requirements as well as those of anonymity and privacy required.

So, as it turns out, this dream is a tricky one. In theory, it shouldn't be that hard: we just need to have someone write some good database system which is scalable, very scalable and then have everyone use it.

Yep. Everyone. Otherwise it won't work. The dream of being able to walk into *any* medical institution *anyw here* and them having access to my medical records (if i allowed them), requires *every single medical institution* to use this very scalable database. Well, with the traditional client-server based network schema this doesn't seem to be possible. Let's take a closer look...

Requirements

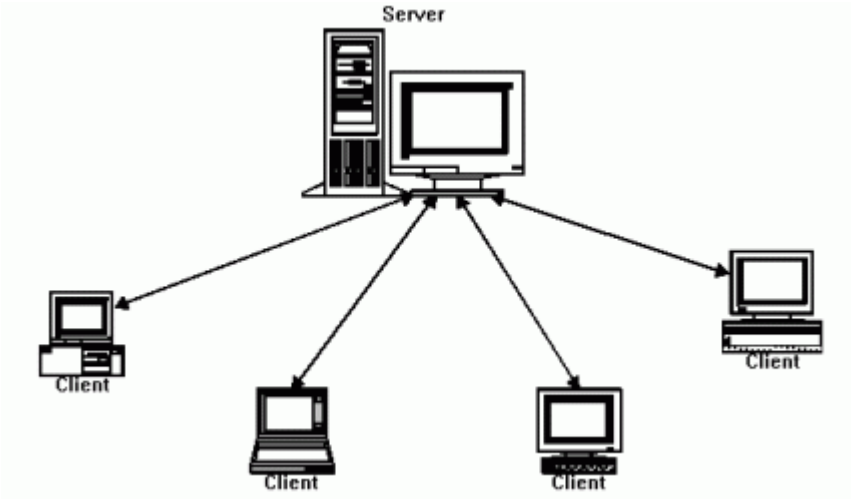
In order to host medical records on a global computer network, we need to make sure the network can accomplish the following requirements:

1. data lineage and data integrity: audit trail
2. data security: storage and transmission encryptions
3. data authenticity
4. the ability to lockout users
5. backups and snapshots of the data
6. cloud-based
7. on-premise/hosted
8. data restoration in the event of a system wide crash
9. scalability
10. high availability
11. difficult to hack
12. agreeable: open and transparent such that every nation and institution can easily agree to join

In order for the "dream" to become true, a system needs to provide 100% of the above points in a strong and stable matter. This is no joke, and it's why for the past 30-40 years humanity failed. Let's start by looking at what we had available back in the 80's and 90's.

Client-Server Network Scheme

This is the scheme that has been around the longest. It's very simple: there's a central server which provides information or services to it's clients, which are the consumers of it's services. One can't really do much with the server alone, unless one has a client as well.

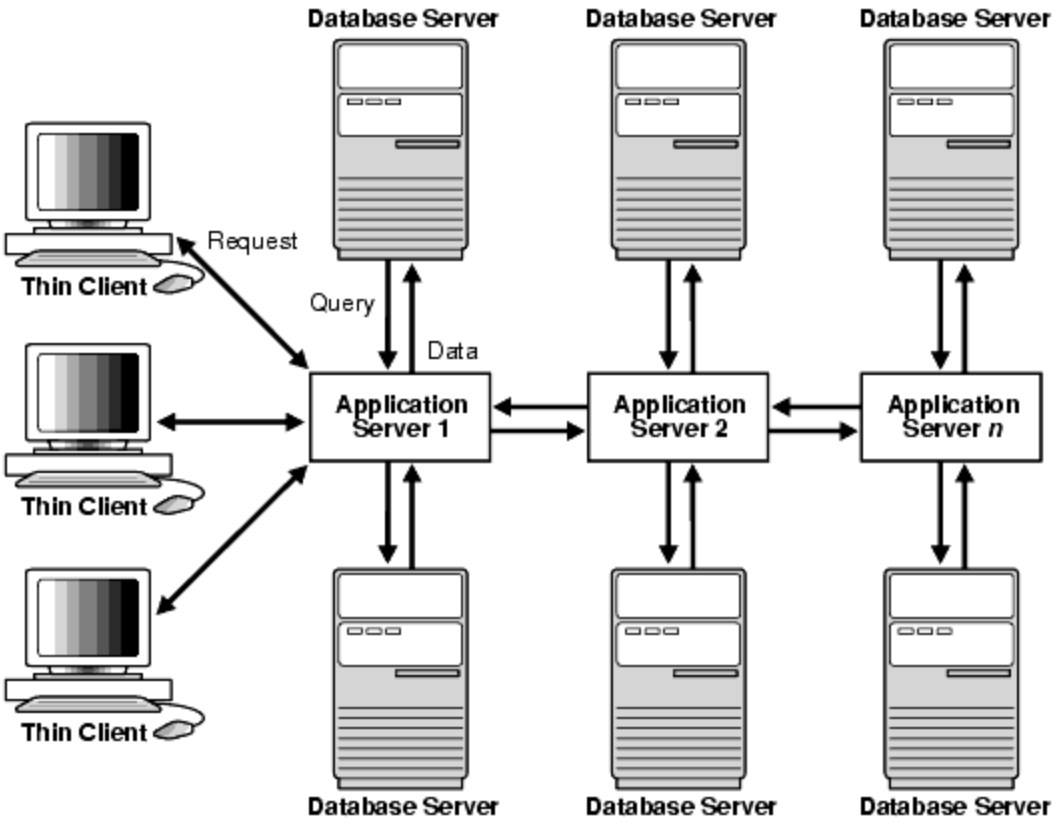


The problems associated with this scheme are

- scalability: if there's one server, then each client has to go to that server in order to get what they want. So the server's network traffic will be used quite a bit. And if it's the entire world, well, that's a lot of traffic. We can fix this today by distributing the main server across a bunch of servers, which are physically located in different places and have a way to sync between each other. This is a concept known as a specific kind of distributed computing called content distribution network (CDN).

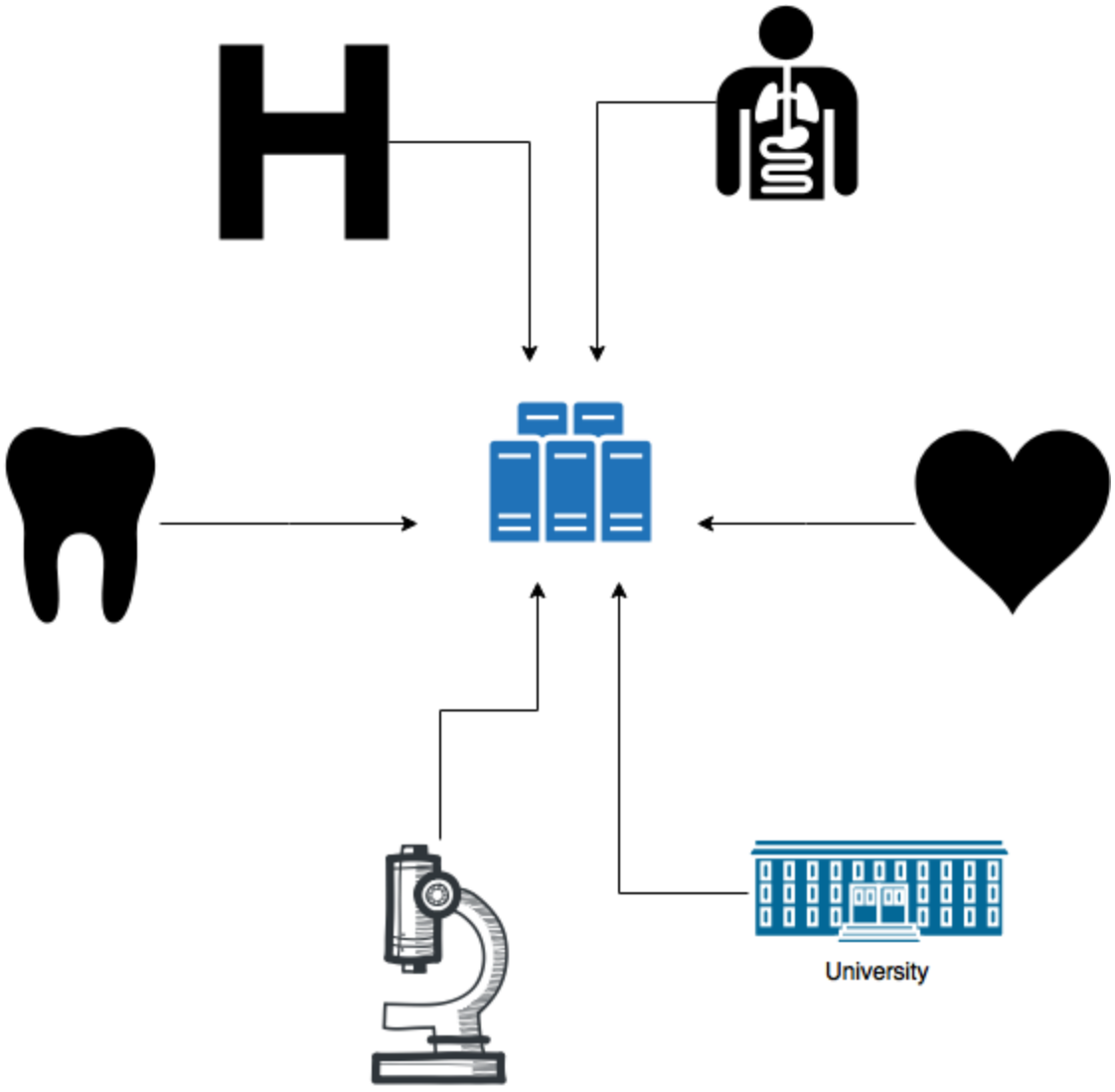
Simple un-encrypted client-server model chart

NO	data lineage and data integrity: audit trail
NO	data security: storage and transmission encryptions
NO	data authenticity
YES	the ability to lockout users
EXPENSIVE	backups and snapshots of the data
YES	cloud-based
YES	on-premise /hosted
SLOW	data restoration in the event of a system wide crash
EXPENSIVE	scalability
EXPENSIVE	high availability
NO	difficult to hack
NO	agreeable: open and transparent such that every nation and institution can easily agree to join



Here, from the clients perspective, they are still connecting to a single server on the network, however, behind the scenes, the server talk to each other to distribute the data, so that a lot of clients can connect at once. This how Google, Amazon, Apple and Microsoft deal with their respective "clouds".

- agreeability: if there's only one entry point, then there's one organization that is controlling it. So we need to have all medical institutions agree to use this single entry point.



Hmm, that doesn't sound too easy. OK, maybe we can get the government involved in this. Yeah, what if we somehow manage to make it mandatory for every medical institution to connect to this network? OK, i guess that could work. Well, for a single country. How about the rest of the globe?

And then there are more problems and concerns which actually make this task quite hard, as addressed in Vest and Gamm's paper titled "Health information exchange: persistent challenges and new strategies", including:

- healthcare providers' hesitation to share what they perceive to be proprietary data
- patient concerns about security and privacy
- lack of strong political will from regulators
- historically costly technological solutions, whose costs often fall to healthcare providers but whose benefits often accrue to patients, payers (e.g. insurance companies), and the healthcare system as a whole

We need to look at another way. And here's another way:

Asymmetric Cryptography

Back in the 80's, we already had invented asymmetric cryptography: it is a kind of encryption scheme which makes use of a pair of keys, as opposed to Public Shared Key cryptography, which makes use of a single key. However, the fact that there were not so many users on the networks and that computers were pretty slow made it impractical to implement. At any rate, theoretically some visionary could have implemented this, and added to the client-server architecture above. By the late 90's we definitely could have had a server-client and encrypted healthcare network.

In public shared key (PSK), two parties that want to communicate with each other in public without having others understand, need to privately exchange a key first. This is what has been done, for example, during the second world war, when spies used to have these code books. The trick here is to make sure each character is encrypted with a different key, and never to repeat the same key. That's why they needed code "books".

Now, one can understand how inconvenient this would be for the internet. Using conventional PSKs, one cannot establish a secure communication, one could not open a website with the secure https channel, without first physically going to the website's headquarters and exchanging a key. Or having the key arrive in a sealed envelope.

With asymmetric encryption, one key is used to encrypt, while the other is used to decrypt. It is not possible to use the same key to do both. So here's how ti goes:

1. Alice logs onto the Bank's website.
2. Bank replies to Alice with it's public key. Who cares if somebody else intercepts the key. It will only mean that they will be able to encrypt things that only Bank can decrypt.
3. Alice generates a random PSK, and encrypts it with Bank's public key, which it just got over the internet.
4. Alice sends over the open internet the encrypted PSK. Who cares, cuz only Bank can decrypt the PSK with it's private key.
5. Bank decrypts the PSK with it's private key.

Client-server model with asymmetric encryption chart

NO	data lineage and data integrity: audit trail
YES	data security: storage and transmission encryptions
YES	data authenticity
YES	the ability to lockout users
EXPENSIVE	backups and snapshots of the data
YES	cloud-based
YES	on-premise /hosted

6. Now Alice and Bank both have a PSK which they can use to communicate securely and keep changing it as necessary.

Pretty smart, huh? Now, why not continue to communicate with the public/private key scheme? Well, cuz it requires a lot more computational power. In other words PSK encryption is simply way more light weight. Using asymmetric encryption would unnecessarily slow down communications.

Now, because of the nature of this dual key mechanism, we can also use it digitally "sign" a file. See, Alice can sign a document using her private key and save it on the internet somewhere, or send it to Bank over regular email. At a later time, Bank can verify Alice's signature, by using her public key, which is openly available on some kind of identity validating network, like a BMV or city hall or passport office. Some office that already has infrastructure to authenticate someone identity. They are the ones that can store public keys. Say Alice wants to publish her public key on the BMV's database of public keys. Next time her driver's license expires, she could, at her home, or with her mobile device, generate a public /private key pair, and keep the private key with her. Then at the BMV, when they verify her identity, she can give them her public key and show them she has the corresponding private key.

OK, so if we add asymmetric encryption to the client-server schema above, we will have achieved:

- identity verification through digital signatures
- theft protection through encryption

However, we will still be left with data integrity. The above methods will make it extremely hard to counterfeit medical records, (if we make sure they are signed and/or encrypted) however they don't alone prevent an ill-intentioned individual from deleting records. In addition, backups, restores and scalability remain still a huge expensive issue. Not to mention agreeability.

SLOW	data restoration in the event of a system wide crash
EXPENSIVE	scalability
EXPENSIVE	high availability
PARTIAL	difficult to hack
NO	agreeable: open and transparent such that every nation and institution can easily agree to join

Peer-To-Peer Networks

This is a network topology that is fundamentally different than the client server based model. Here's a table that summarizes the basic differences.

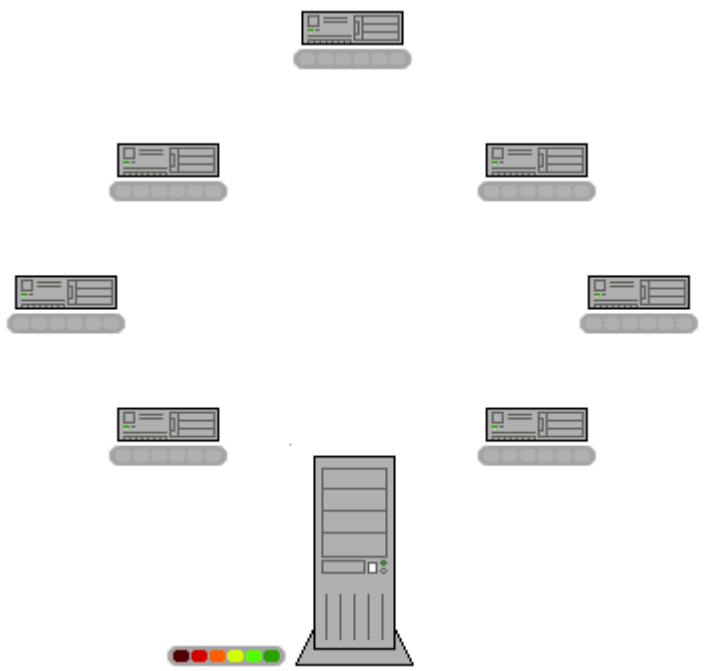
Client-Server	Peer-To-Peer
Server is the only provider of resources, clients are the only consumers of resources.	Each node is both consumer and provider of resources.
Server decides who has access to what: it runs different software than the clients and is the most powerful node of the network.	Each node is equipotent. Each node runs exactly the same software.
To distribute information, a client has to upload it to the server, who is the only one responsible to distribute information to the clients. Clients don't communicate with each other.	To distribute information, a node only needs to upload it to the network only once. Then the rest of the nodes distribute it amongst each other.
Performance decreases as clients increase, because the server gets overwhelmed.	Performance increases as nodes increase, because the number of resource providers increase as well (each node is a resource provider, see above).
Hackers need to take one server down to kill the entire network.	Hackers need to take over (in most cases) at least 51% of the network nodes in order to compromise data on the network.

In a p2p network (peer-to-peer) there is no central servers. This means that there is no central authority which needs to setup the network and pay for infrastructure costs. Every peer on the network agrees, by joining, to share it's resources. so the network sets its self up. Or, each peer on the network is a contributor for setting up the network.

Peer-To-Peer network with asymmetric encryption chart

NO	data lineage and data integrity: audit trail
YES	data security: storage and transmission encryptions
YES	data authenticity
POSSIBLY	the ability to lockout users
YES	backups and snapshots of the data
YES	cloud-based
YES	on-premise /hosted
N/A	data restoration in the event of a system wide crash
YES	scalability
YES	high availability
POSSIBLY	difficult to hack

YES	agreeable: open and transparent such that every nation and institution can easily agree to join



There are plenty of resources on the internet that explain how p2p networks work. You can start with the wikipedia page. If we look at our requirements table, we can see a significant improvement: this technology is really powerful. However, an encrypted p2p network alone will not help us make our dream come true, because it will not address one of the fundamental requirements of counterfeit protection: audit trail and data lineage. In other words, just by slapping our medical records on an encrypted p2p network will still allow anyone to delete anybody else's records. Not good.

Bitcoin and the Blockchain

In 2008 a white paper has been distributed in mailing lists by an unidentified individual called "Satoshi Nakamoto" describing a novel approach of using the hashing algorithm to create a financial ledger which is unmodifiable once approved, without making use of a central authorizing institution. An open source implementation soon followed in 2009.

There are many resources available on how the Bitcoin technology works: a peer-to-peer distributed public ledger, which groups transactions into blocks. The ledger is made out of blocks, one chained together to the other, and to calculate these blocks, a proof-of-work needs to be accomplished. In other words, a block cannot be simply posted onto the ledger, unless it has taken a lot of work to calculate it.

Summarizing, this is more or less how Bitcoin works:

1. Alice wants to send 10 BTC to Bob. She already has a Bitcoin "account" with 10BTC in it.
2. Bob generates a new Bitcoin account (which is nothing else than a public/private key pair, as described above) on the fly, and sends Alice its account address (i.e. the public key).
3. Alice uses a Bitcoin client (called *Wallet*) to create the transaction. Just like any peer-to-peer network, in order to access the network, you need an application. Skype, bittorrent, TorBrowser all work the same way.
4. Alice's Bitcoin client (*Wallet*) creates the transaction signing it digitally, and uploads it on the Bitcoin network.
5. Special bitcoin nodes (called miners) will
    a. verify that Alice has enough funds;
    b. package the transaction along with many others into a block of transactions;
    c. add a hash (unique digital signature) of the last approved block on the blockchain to this new block
    d. guess an extra random variable (called "Nonce"), such that the resulting hash of this block will start with a predetermined number of zeros. This is the key part of mining, takes up a lot of work and is what makes the blockchain very hard to modify
6. Once the block is ready, the miner announces the new block to the Bitcoin peer-to-peer network.
7. Each node on the network can easily verify that the new block is, in fact, valid. The hard procedure performed by the miner in step 5d is very easy to verify due to the one-way nature of the hashing algorithm.
8. If the block is valid (transaction are valid and hashes are valid) the bitcoin nodes accept that as the latest approved blockchain and move on.
9. It is now official! Alice has sent 10BTC to Bob and everyone agrees.

This is the big picture. You can take a look at some diagrams on the web which try to depict this, if it helps. I personally find them confusing. However, here are some references:

Blockchain: Peer-To-Peer network with asymmetric encryption on signed public ledge chart

YES	data lineage and data integrity: audit trail
YES	data security: storage and transmission encryptions
YES	data authenticity
NO	the ability to lockout users
YES	backups and snapshots of the data
YES	cloud-based
YES	on-premise /hosted
N/A	data restoration in the event of a system wide crash

- https://vulcanpost.com/235071/tiasg2015-day-2-startups-bitcoin-trend/
- http://bitcoin.stackexchange.com/questions/4838/what-does-a-bitcoin-transaction-consist-of
- https://pbs.twimg.com/media/BXYPgF5IIAE5xdU.jpg
- http://www.nanozine.org/bitcoin-mining-diagram.html

Now let's tackle the key points. Before we can understand anything about Bitcoin (and the Blockchain) we must understand that the entire concept is based on asymmetric cryptography and hashing functions. We already covered asymmetric cryptography above.

Hashing Algorithms

Don't get scared. A hashing algorithm is nothing else but a tiny little computer program that assigns a unique identifier to anything we feed into it. That's all.

So if i have a word file, i can feed it into this tiny little program, which will read every bit of the word file, then spit out a unique ID for that file. It's unique in a sense that for any exact copy of the file, the program will generate the same ID. But any kind of variation to the file, however tiny it is, will produce a new ID.

Another peculiarity of the generated ID, is that it's of a fixed length. So, no matter what i feed into it, whether it's the number "1" or the entire internet zipped up into a single compressed file, the length of the produced ID will always be the same.

To give you a feel of what hashing alghoritm is, let's consider a very simple hashing algorithm that works like this:

1. first, let's assign a number to every letter of the alphabet. So a=1, b=2, c=3, etc;
2. then, we'll sum all the numerical values of the letters;
3. the final number would then be used to find the character that represents the 'digest' (lingo for *ou tput of the hashing algorithm*). The algorithm to find the final character in this case is to simply use the letter assignment we did in step 1, but backwards. If the algorithm returns a value greater than 26 (=z), then wrap around the alphabet, and start from 'a' again.

message	algorithm	digest
'hello'	8+5+12+12+16=53	'a'
'cello'	3+5+12+12+16=48	'v'

As you can see, this algorithm is not reversible. In other words, you can not compute the message 'hello' given the digest 'a'. You could only brute force it, by producing a long list of combinations which create the digest 'a'. However, every time we compute the hash of 'hello', we will only get 'a'.

Another important characteristic of hashing algorithms, is that they are completely unpredictable. So it's not possible to predict a hash, from another one. So, even a small change in the message, should produce a seemingly completely random change in the digest.

A real hashing algorithm looks more like this:

message	hash
"1"	`b026324c6904b2a9cb4b88d6d61c81d1`
"2"	`26ab0db90d72e28ad0ba1e22ee510510`
"Medical Blockchain"	`22050fc796acc79785a2ebe3144a1d02`
"Medical Blockchains"	`adcdd542ac78a1882e6c5389815f0865`

How does one create a Bitcoin account, if there is no central organization involved?

asdf

POSSIBLY	scalability
YES	high availability
POSSIBLY	difficult to hack
YES	agreeable: open and transparent such that every nation and institution can easily agree to join

Informatics Standards: a Necessary Step

As the global and universal network slowly takes shape, the question of which format and standard to use for the information to be shared will come up. For now, the big talk is about this new technology, and how it can work. Once this ground will be covered, all medical institutions will have to be able to communicate and exchange data on this medical network. Current software will probably need some sort of import/export conversion program, unless the software already supports the standard.

Current medical informatics standards being used today are HL7 for all kinds of medical messaging and DICOM for all kinds of medical images and videos. There are other ANSI approved SDOs (Standard Development Organizations) like the American Dental Association Standards Committee for Dental Informatics (ADA SCDI) which specialize in specific medical fields. While it is not guaranteed that these will become the official communications standards for this network, it sure would be a shame if they would not (or wouldn't at least lie the foundations). These SDOs have been around for decades and have well established standard development committees and have been implemented widely.

How It Would All Look Like

1. The patient arrives, to see a doctor for the first time (ER, dentist, ...).
2. The medical provider's front desk would ask for their digital ID.
3. Kind of like a credit card today, or Apple Pay or other such mechanism, the patient interacts with the card reader, and enters his/her credentials (with OTP, fingerprint, or whatever).
4. The system is now able to retrieve the patient's ID and digital key by combining their card with their authentication.
5. With the digital key of the patient and the provider's own digital key, the system can:
    a. have access to the lookup table of "the network" with the provider's key. this gives access to de-identified records.
    b. search for and download the correct record using the patient's digital ID.
    c. decrypt identity information, using the patient's digital ID, and by so doing, obtaining the full record of the patient.

Interesting quotes

However, for the technology to be adopted in the mainstream, all players – financial services companies, regulators, governments – need to agree on certain standards.

The blockchain is the most significant new technology of the decade.

(https://techfinancials.co.za/2017/02/28/blockchain-potentially-transformational-financial-services/)

Interesting and Related Reads

- https://etheal.com From DokList.com, some kind of travel-agency for connecting doctors to patients. Hungarian. Reasons listed in white paper for Blockchain usage and advantages don't seem to make sense. I.e. it seems like all they claim the blockchain can solve, can be solved w /o blockchain as well. White paper talks mostly on profits and gains.
- https://healthbase.digital (working in dental clinics). Want to expand an already existing software to use blockchain, in order to make clinical data globally available also to facilitate research. Goal seems to be offering a quality product for medical providers, thus enhancing patient care.
- http://markets.businessinsider.com/news/stocks/Taipei-Medical-University-Hospital-and-Digital-Treasury-Corporation-Jointly-Release-phrOS-The-First-Healthcare-Blockchain-Platform-Worldwide-1007998017
- https://phros.io Want to create a global medical records platforms. Research is clearly stated in their vision.
- The Potential for Blockchain Technology in Health IT
- http://blockgeeks.com/could-blockchain-be-the-answer-to-healthcare/
- The permanent web for healthcare with IPFS and blockchain
- https://www.healthit.gov/newsroom/blockchain-challenge ad the http://wayback.archive-it.org/3926/20170127190114/https://www.hhs.gov/about/news/2016/08/29/onc-announces-blockchain-challenge-winners.html
- A great example of how the future could look like if we applied blockchains to medicine
- IBM Watson Health (NYSE: IBM) has signed a research initiative with the U.S. Food and Drug Administration (FDA) aimed at defining a secure, efficient and scalable exchange of health data using blockchain technology.
- Blockchains and electronic health records (Ben Yuan, Wendy Lin, and Colin McDonnell)
- A Case Study for Blockchain in Healthcare: “MedRec” prototype for electronic health records and medical research data
- HIMSS organized an entire day dedicated on BLOCKCHAIN IN HEALTHCARE: A ROCK STARS OF TECHNOLOGY EVENT
- Blockchain Code-A-Thon backed up by the ONC

Footnotes

Unknown macro: 'display-footnotes'