Ethereum, IPFS and neural compression to decentralize and protect patient data in computational pathology

Manuel Cossio
Dept. of Mathematics and Computer Science
Universitat de Barcelona
Barcelona, Spain
Zürich, Switzerland
manuel.cossio@ub.edu

***Abstract*—The field of digital pathology produces a large number of images associated with patient metadata that are the raw material of computational pathology. The process of making images available with adequate privacy and data protection considerations takes a long time. Given that the Ethereum network associated with InterPlanetary File System (IPFS) promotes the exchange of information in a secure, private and decentralized manner, this association could be an important partner between digital and computational pathology. Therefore, here we propose and discuss a prototype with the aforementioned parts and the addition of neural compression, as an essential information preservation step. This prototype could constitute a link for the exchange of information in a secure way, providing transparency and reliability to the chain and empowering the field of manufacturing artificial vision solutions for the medical field.**

***Index Terms*—digital pathology, Ethereum, blockchain, IFPS, neural compression, computational pathology**

I. Introduction

Thanks to advances in artificial intelligence, especially computer vision, computational pathology applications have increased considerably in recent years.

We can define digital pathology as the process or set of processes by which slides with already processed samples are digitized. In this context, digitizing implies transferring the image that is observed in a light microscope to an image file [1]. These files are called WSI (Whole Slide Image) and have a characteristic that makes them essential for use in pathology: they store different levels of magnification in the same file with extremely high levels of compression. This type of configuration is referred to as pyramid and it allows to optimize the visualization and exploration of the image, by allowing macroscopic exploration at very low levels of magnification and then going to the detail of interest by increasing the magnification [2], [3]. The WSI can be stored on physical servers located in the same hospital or can be stored in cloud storage systems. Its storage has allowed the interconsultation of the same case between several pathologists from different parts of the world, considerably expanding the accuracy of the diagnosis in complex cases [4], [5].

It can be said that digital pathology ends with the digitization of the slide and its storage in the cloud or the physical server [1]. Part of the extension of the definition of digital pathology, recently includes the remote consultation of cases between several institutions from different physical locations in real time [5]. On the other hand, computational pathology includes all those projects and/or prototypes where WSI images are used as a dataset and machine learning or artificial vision algorithms are employed [6], [7]. Computational pathology does not include the use of scanners, or microscopes, or the implementation of molecular biology laboratory procedures. In this discipline, we already have the images stored, which have the authorization of the ethics committee of the donor institutions and, if the case warrants, the informed consent of the patients to whom they belong [8]. Computational pathology also includes digital modification of WSIs (cropping, alignment, contrast, brightness, color channels, color normalization, etc. [9]) and clinical data capture by aligning WSIs with electronic health records.

II. Data workflow

The data workflow in digital pathology can take many forms. However, the main skeleton is more or less homogeneous in different countries, health networks, hospitals and laboratories. We can begin to describe it by naming its first part: data acquisition. As we have already seen in other publications, data acquisition begins with the capture of the image by the scanner, once the sample is treated and prepared following the determined protocol (for example, HE). That captured image must be temporarily stored and then reviewed by a qualified professional to determine if the quality is correct for diagnosis (focus and tissue area). This verification is usually done with the help of the barcode (linear or 2D) that identifies the sample and attaches it to the patient's file. Once the image is verified as correct (without interpreting the diagnostic), the storage of the image is moved to a permanent physical or virtual disk. If the disk is physical, the storage server may be located in the same hospital or in the same laboratory. If the server is virtual, the host system can be in the same country or in a different country. Now, at the moment of establishing the diagnosis (interpreting the image), using it for a virtual consultation (let it be interpreted by another professional pathologist in another place) or when an algorithm analyzes it autonomously, the image must be made available.

III. Telepathology

A. Requirements

Telepathology is a term that was coined around 1980 and began referring to the real-time scanning of slides for multi-consultation in pathology athenaeum rooms [10]. The term today has evolved and refers mainly to the possibility of a pathologist analyzing a sample remotely, without even being in the same hospital, city or country where the sample resides. Now, as we mentioned before, the WSI must be available so that the pathologist who is going to carry out the teleconsultation can observe it. In addition, if there are notes on the image, these must also be accessible so that the professional can confirm or extend the diagnosis. So this is where there are some variables that are very important when it comes to providing an optimal service. These variables are:

1) Latency: which is the sum of all the delays that a data packet suffers when it crosses a network from one point to another (or also including the return to the starting point) [11]. These delays may be due to different causes such as the need to amplify the signal when the line is very long or in urban lines, the large number of small interconnections due to the higher population density (offices and buildings). The size of the data packets is also a factor that influences latency and the larger the size, the more latency associated with the transmission.

2) Bandwidth: This parameter can be defined as the maximum amount of data that can pass from one point to another in a network per unit of time. It is commonly measured in kylobytes, megabytes or gigabytes per second [12].

B. Requirements

To carry out a teleconsultation of a pathology image in real time and at the same time be able to listen to the feedback of the primary pathologist, the minimum requirements should be above a bandwidth of 10 MB/s and the lowest possible latency [10].

IV. Data protection

As we saw in the previous sections in digital pathology, telepathology and computational pathology there is a large amount of sensitive data transmission. Within the category of sensitive data, we can include all those that could identify a person, such as gender, age, clinical data, place of birth, treating physicians, place of residence, outpatient medicine center closest to home, habits food, social habits, family members, among others. That is why all kinds of measures and protocols must be implemented so that the protection of data is not violated, harming as little as possible the progress of scientific research. In the following sections we will discuss some protection concepts for data exchange and expand on the notion of decentralization of computing.

V. Ethereum

First of all, let's try to put together a definition of Ethereum. Ethereum is an open source platform. Ethereum technology allows you to create applications and organizations, carry out any type of transaction, store assets and communicate without the need for a centralizing authority that is the link for these activities.

1) Open source: When we say that a platform is open source, we are specifying that its code is public and that it can be modified and used freely. All products created with this platform can also be open source. This means that the products will have all the code open to the public, documents or any type of files can be created with the product and the content can be modified by other users.

2) Central authority: The world we live in today uses central authorities to run our digital activities. Examples of this are:

- Banks: all the payments that we want to make, deposits and investments are controlled by this entity.
- Communications: our social networks, emails, text messages are controlled by central authorities such as Facebook, Twitter, Gmail, Outlook or Whatsapp.
- File sharing: When we upload files and send a link for someone else to download, we again require a central authority: Dropbox, Google Drive or OneDrive.
- Video sharing: When we upload content in the form of videos, the central authority can be YouTube or Vimeo.
- Travel: when we decide to move from one place to another, the central authorities are applications such as Booking, Uber and all those applications of airlines and trains.

As we see in the previous examples, there is always a central authority (company) that coordinates the interactions of each user and that also stores all the private information of each person. Within the private information we can cite identification data, address, telephone, frequent trips, destinations and schedules, daily activities, personal content in the form of videos, people with whom we often interact, political and government positions, etc.

A. Ethereum Virtual Machine (EVM)

This term refers to the Ethereum space that is maintained by the client network. This network is the one that hosts all the accounts that are made in Ethereum and basically where the smart contracts subsist.

B. Nodes and clients

Within this network, a node is a computer that runs Ethereum software. On the contrary, a client is a node that fulfills the function of verifying the validity of the data of the network keeping it safe [13], [14].

Within Ethereum (more specifically, EVM) there are two types of clients:

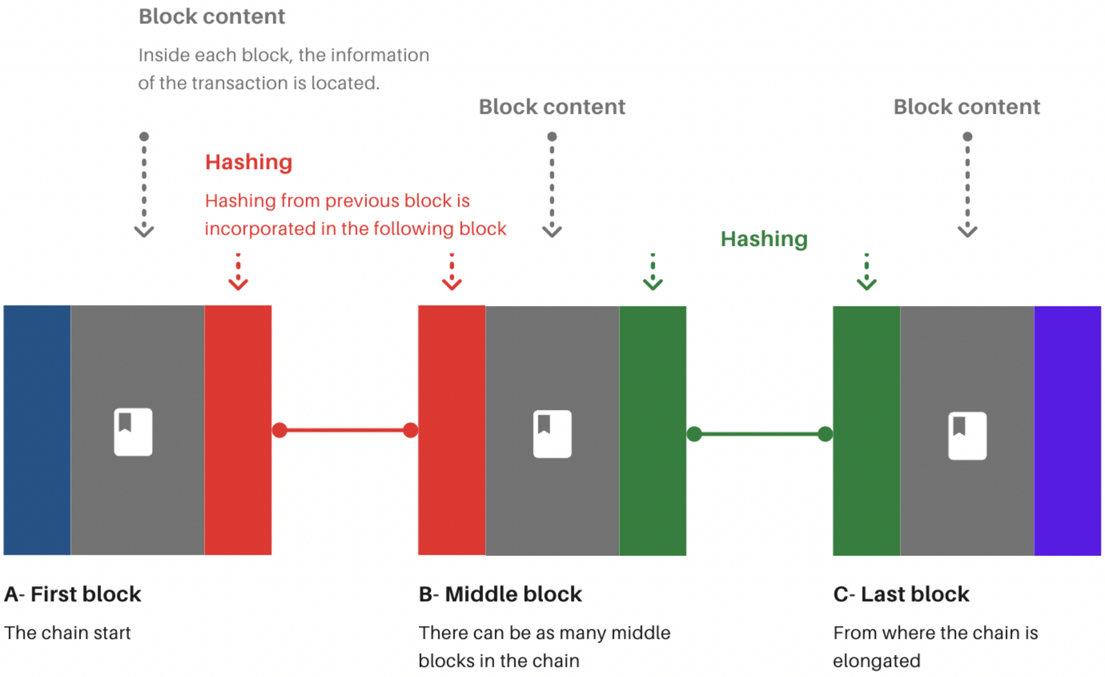
Block content
Inside each block, the information of the transaction is located.
Hashing
Hashing from previous block is incorporated in the following block
Block content
Hashing
Block content
A- First block
The chain start
B- Middle block
There can be as many middle blocks in the chain
C- Last block
From where the chain is elongated

Fig. 1. Representation of an Ethereum blockchain, with 3 blocks.

- Execution client: it is that client that tracks the transactions that are broadcasted on the network and executes them with the latest available statuses of information and updates.
- Consensus client: These are the clients that run proof of stake (see section) to gather consensus on validated data from the execution client.

C. Blockchain

Blockchain is basically a list of records or documents linked like a chain. Each document has a specific order in the chain and is called a block. Each block is also made up of a code called cryptographic hash that comes from the previous block (Fig. 1), a timestamp and specific data of the transaction that block represents [15].

D. Cryptographic hash

A cryptographic hash is the result of a mathematical function called a cryptographic hash function where information of any size (documents, photos, code) enters as input generating an output that is a code of a fixed size. This code is unique, which means that there can be no document other than the input that has that hash value. Moreover, a small alteration of the document (such as a single space in a 1000-page text document) when input to the cryptographic function, would generate a different hash. This is key to the validity and importance of this function, since it allows identifying files without revealing their content and with high reliability in the result [16].

A very important consideration about this function is that it is one way. We can generate the hashes with different inputs, but it is practically impossible to find out the input given a particular hash.

We can venture to list certain characteristics that make a good cryptographic hashing function [17]:

- It must be deterministic, that is, the same input must always generate the same hash.
- Must be fast to compute
- It should not allow the decryption of inputs from a hash
- It must not allow the existence of two different inputs with the same hash
- It must use the avalanche effect, where a small change in the input produces a huge change in the hash (see Fig.2). This prevents two hashes from being correlated and thus inferring correlation from two inputs [17].

E. Consensus

Another very important stepping stone of decentralized systems like Ethereum is consensus. In centralized systems, this problem does not exist since the central node (for example the bank) is the one that decides what to do with respect to a possible transaction. In decentralized systems, many nodes intervene in the decision of a possible future transaction and here some decision mechanism must be put into practice [13], [14].

1) Byzantine generals problem: It is a problem that emerges from game theory that describes the difficulty experienced by

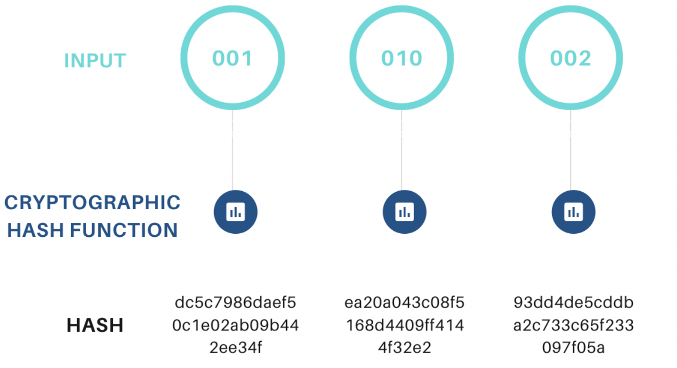
INPUT
001
010
002
CRYPTOGRAPHIC HASH FUNCTION
HASH
dc5c7986daef5 0c1e02ab09b44 2ee34f
ea20a043c08f5 168d4409ff414 4f32e2
93dd4de5cddb a2c733c65f233 097f05a

Fig. 2. Example of a cryptographic hashing function (MD5) obtained from a **generator**.

decentralized systems (several nodes that interact) when there is no verified and authenticated central node. In this problem, the nodes have to reach a consensus on what action to take in the future, and some nodes may have wrong or missing information about other nodes. This lack of information can cause a node to make an incorrect decision that affects the system. In the specific case of the example, the generals are the ones who decide whether or not to attack a castle. The generals all have the same rank so their influence on the decision weights the same. If all the four generals decide positively (and the information is true and correct) they all attack the castle in coordination and win the war (Fig. 3A.). Now, if two generals falsely express that they will attack the castle, the other generals take that statement as true and then only two of the four attack and they lose the war (Fig. 3B.) [18].

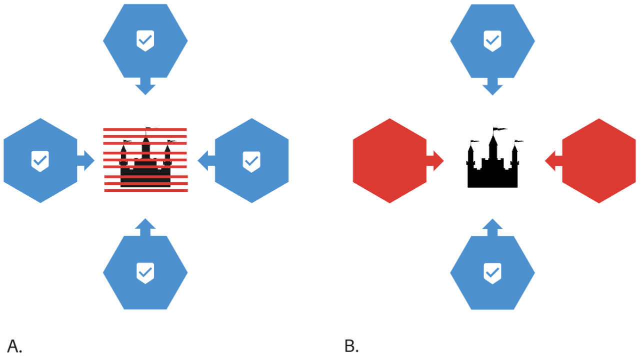

Fig. 3. Example of the Byzantine generals problem. A, they win the war and B, they loose the war.

As can be interpreted from the example, being able to count on some type of validation from each of the parties involved in a consensus is key. For this reason, various mechanisms have been proposed in decentralized systems to certify the trustworthiness of the nodes.

2) Proof of work: To explain this complex mechanism, we first have to recall the structure of a blockchain. As we mentioned before, it is a chain of blocks, where the block we analyze has a cryptographic hash of the previous block. In this same way, the next block has another hash of the block we looked at. This chain of hash-linked blocks is what somehow validates the list of transactions made on the network. The proof of work (PoW) mechanism basically consists of trying to guess a particular hash called the nonce of the last block in the chain. In order to find this hash, all the information found in the chain must be passed through an algorithm that, through trial and error, will return different nonce values [13], [14]. As the processing time passes, the algorithm will be closer to finding the correct nonce. When the algorithm hits the correct nonce, two events will occur [13], [14]:

- First, the entire chain will be validated and one more block will be added to it.
- Second, that node (miner) will get its proof of work and will receive a reward for the time invested. This reward is what encourages miners to validate blockchain and in this way, the longest chains are the most reliable (because they have been validated several times)

One criticism of this validation system is the large amount of power it consumes. Just to cite an example, to maintain the Ethereum network through PoW, 72 terawatts per hour are consumed annually. This amount of energy is consumed by a medium-sized country, such as Austria. That is why other validation measures are being implemented, such as proof of stake (PoS) [13].

3) Proof of stake: As we mentioned before, a big problem with PoW is the large amount of energy it consumes. For this reason, a new strategy has been proposed and validated and is currently available on Ethereum: proof of stake (PoS). In this new strategy, instead of executing complex computational tasks for time to verify a chain, verifiers are used. These verifiers, in order to be validated, must deliver a fixed amount of 32 ethereum coins (ETH) to the network. Once validated as verifiers, these agents must respond to calls from the network to verify blocks of a given chain. When they finish verifying it, they cast a vote (called an attestation, which can be positive or negative). The more ETH (greater than 32) a verifier gives to the network, the more weight their attestation will have. Considering the amount of ETHs in the network, the positive votes (with each weight) of all the verifiers of a particular block must be added. If the addition reaches 51%, the block will be validated.

Among the advantages of this new consensus system, we have the following:

- Less energy consumption, as we mentioned at the beginning.
- Less need for high power hardware. As we saw before, in PoW the faster the nonce was found, the verifier ’verified’ the chain and collected its reward. In this way, this consensus mechanism required more and more

computational power to be able to mine. In this new system, PoS, it is not necessary to have complex hardware since it depends on the amount of ETH that is invested.
- More decentralization. Since fewer hardware requirements are needed, many more agents can become validators and strengthen network security and reliability.

One last point worth mentioning about this consensus system is the penalty. Validators who do not respond to the network call on time or who act fraudulently could be penalized. The penalty may consist of the destruction of a percentage or all of their ETH assets. In more extreme cases, the network can vote and permanently ban that agent. These mechanisms reinforce the correct action that directly impacts on the security of the network.

4) Smart contracts: This section is one of the most important, for the purpose of this paper, which is to protect the sensitive information of patients who donate their data.

To start outlining it, a smart contract is a computer program. This program works in a similar way to a physical contract, that is, if certain conditions are met, an action will be carried out. In other words, this type of contract is an agreement where if the requirements detailed in the smart contract are met, it will be executed.

A very basic example of how a smart contract works is a parallel with a train ticket vending machine. Here we select the route and the number of passengers. The machine receives this request and analyzes if the route is available and if there are seats. If both previous options are 'True' then the machine will send us a message with the price. If we pay with our credit card and the amount is received by the system, the ticket will be printed. If the card does not work and the sum is not received, the ticket will not be printed.

Among the characteristics that make smart contracts exceptional tools for many tasks, we encounter the following:

- Automatic execution: when the conditions are met, the contract is executed automatically, without the need for human validation. This directly impacts positively on trust, since if one of the parties meets the conditions, they will receive what was agreed in the contract safely.
- Fixed and predictable outcomes: by eliminating the human factor, one and only one specific outcome will be the result of the execution of the contract. This eliminates the effect of physical contracts that two arbitrators (judges or mediators) decide differently on the same contract.
- Public record: smart contracts are public, so any member of the network can track their progress and completion.
- Privacy protection: since all the details of the smart contract are linked to a cryptographic address, there is no chance for users to discover the real identity of the person behind the virtual address.
- Terms and clauses visible to all: each part of the contract is fully visible to anyone. This allows you to interact with the smart contract before signing it to be sure of its content.

5) Non-fungible tokens (NFT): We will now explain what these types of tokens consist of, which fulfill a very important function to represent data that is unique and non-interchangeable. When we talk about patient information, this is of vital importance since, for example, a chest x-ray belongs to a single patient and cannot be exchanged for another.

To begin we will define the meaning of fungible. Fungibility is the ability of two objects to be interchangeable with each other. For example, two 1-euro bills can be exchanged with each other since they both fulfill the same function and are not unique. In the opposite case, a non-fungible object can be a work of art that obviously cannot be exchanged for another that represents the same thing.

In this way, NFTs represent assets or files that are unique. And they have the following properties:

- They cannot be divisible: that is, half an NFT cannot be delivered.
- They indicate ownership: the titular holder of the NFT owns what the NFT represents.
- They are transferable: they can change owners, for example through a smart contract that specifies it.

VI. Ethereum for patient data

As we have seen before, the Ethereum network provides endless possibilities to handle sensitive patient data safely. One of the advantages is decentralization, which would increase security against data loss compared to centralized storage. In the following sections, we will dig a little into some applications for sensitive data.

A. Smart contracts

One of the first applications that we should definitely mention for managing patient data is MEDREC. This application is one of the first created for the management of electronic health records in a decentralized way through the use of Ethereum and smart contracts. In this case the network is private, you must have permissions to perform data retrieval. One of the incentives for researchers and partner institutions to participate in block mining is the possibility of receiving fully secure research data [19].

Within the smart contract applications, a group from the Cooper Medical School of Rowan University created a protocol to securely store clinical information from patients with COVID-19. They used data such as patient ID, Chest CT grade (lung compromise), comorbidities and age and created a smart contract to store this data in Ethereum. They observed that as the number of records inserted in the database increased, the insertion (up to 674 milliseconds) and retrieval (up to 112 seconds) times increased significantly, but not the storage. However, taking into account the characteristics of the smart contract, they highlighted how automation in execution and security contributed to an operation focused on data protection. Given that only clients registered in the contract could make a query and that the entire database was encrypted at all times, they explained how this protocol could be useful in the future to make clinical data available without circumventing the privacy of the patients [20].

The same group also proposed a protocol similar to the previous one, but this time to save data from patients who have undergone neurology imaging procedures [21].

Another group from Khalifa University proposed a protocol also with smart contracts for the handling of information and the management of clinical trials. They established a structure where the regulatory body of health (Food and Drug Administration, FDA) and the sponsor are the ones that directly interact more closely with the contract. They can initiate a drug application and start the trial, if the principal investigator also submits the approval. The work highlights the advantage of decentralization to avoid system collapses and to prevent a single party from having complete information on all patients. They also emphasize the importance of encryption of all clinical variables, the null access that physicians have to the aggregated data (established by the contract) and the enforcement clause that the smart contract promotes to all parties. In this way the data is protected in a decentralized way and the trial protocol is carried out without modifications [22].

At the University of Brawijaya, a protocol was proposed to emulate an electronic health record system on Ethereum. To do this, they created a system with three side nodes, each connected to a hospital or institution with permission to upload data. They also included a central node that is the one that runs the smart contract, which in turn has a manager who is the only one with a license to deploy contracts. They carried out multiple tests where they managed to save and retrieve medical records from the different nodes, also testing the integrity of the contract, to ensure that there were no changes that could compromise the security of the network [23].

A group from Thakur College of Engineering and Technology developed an application with Ethereum and a smart contract to provide portability to patients' electronic health records. This application focuses data management on the patient, since it is the doctor who requests access to the records and it is the patient who authorizes access and the loading of new data. An important strategy is the use of the Inter Planetary File System (IPFS) to save large files (MRI or CT images, labs) without resorting directly to saving them in Ethereum given its high cost. In this way, information continues to be decentralized and protected in a hybrid way [24].

Authors at Yale University proposed a protocol to store gene variants and associated drugs in Ethereum through a smart contract. In this work, the query times were optimized and they emphasize the importance of being able to decentralize and protect genomic information, which constitutes sensitive patient information [25]. Additionally, the information on genomic variants in pathologies and drugs that act with these variants is public (e.g. GeneBank). However, when that information is associated with a patient, it ceases to belong to the public domain and becomes identifying information. This is why this preliminary work is important in order to later apply it to health records.

Researchers at Fordham University created a protocol to analyze data from mobile sensors in patients. This particular work uses the patient's own mobile phone with a data analyzer, so a large part of the data protection is preserved by not sending all the information to the network. Then the mobile sends the extracted data to a first general contract, which will act according to what is received. This contract, in turn, will send an extract to a second contract, which is the one that will generate the alert for the hospital (sending it to the mobile) and that will send the data to Ethereum. Once the mobile phone receives the alert for the hospital, it will send it directly to the institution, preventing the hospital from having access to the network [26].

In Indonesia, a group proposed a system with Ethereum smart contracts to carry out telemedicine consultations, where the information and network transactions are secure. Like the group mentioned above [24], they use IPFS to store files and thus reduce the cost and increase the speed of transactions, while maintaining the level of security. In this specific work, they also studied the speed of sending and retrieval of data to the network to improve optimization [27].

A group from The Center for Research Technology Hellas developed an application to share clinical information of patients using Ethereum and 3 different types of smart contracts. Of these contracts, we find one that manages the registration of new users, another that is responsible for managing the data and ensuring protection and a third one that coordinates access permissions. In this way, the patients are the ones who have their clinical data on a smartphone and who decide to donate the data. The donated data is transferred to a database that is managed by a smart contract that protects privacy. Privacy is protected by encapsulating the patient's ID in the contract and randomizing the data storage in the database (so as not to know the identity of the patient by order of registration). Partner institutions then, once they are granted access permission by contract, can perform fully anonymous data retrieval with clinical value [28].

A quite different application to the previous ones was carried out by a group from Nirma University with a solution for telesurgery. In this work, the authors created a prototype with two smart contracts in Ethereum, one for the patient and one for the surgeon. The patient handles all clinical information and records and the surgeon grants permission to access patient data and review the case. In order to access a surgeon contract, the person must meet requirements such as having a medical degree, residency and accredited practices. The patient will be able to choose, with which surgeon to carry out his intervention. When the intervention is finished he will be also able to leave a review of the practice. The reviews of the surgeons will only be visible by the patients and cannot be altered due to the control by the smart contract [29]. For the storage of files, IPFS will be used, as well as works already mentioned before [24], [27].

Finally, a group from Ajou University developed a protocol with Ethereum and IPFS (like previous works [24], [27], [29]) for sending and handling digital pathology images. In this prototype, users who want to access the images must receive a key to access the encrypted images. This protects privacy and eliminates unauthorized access to third parties. In this proposed model, the pathologist is the one who receives the images and uses them to train a neural network [30].

B. Considerations

After analyzing numerous examples of clinical data management applications with smart contracts on the Ethereum network, we can superficially summarize some important points to keep in mind if we want to start a project using these tools:

- As we have seen at the beginning and corroborated in the examples, the decentralization of information restores autonomy to the contributing parties (patients, institutions) and reinforces data protection.
- Applications can be created publicly, using the existing Ethereum network, or they can be created privately, emulating EVMs and clients on a local network.
- Networks built using the public platform, while highly secure, are slightly less secure than private networks. This consideration must be taken into account, according to the type of application that you want to develop. Private networks can be constituted in hospitals, hospital networks, cities, regions or countries.
- There are two fundamental variables that are the saving time and the retrieval time. Optimizing these parameters is essential to achieve a network that responds efficiently to information demands. For example, in the operation of remote agents (telesurgery) it is important to optimize these variables to the maximum to avoid delays in the execution of tasks.
- Different parties to a smart contract may have different levels of authority and specific tasks that they can perform. In this way, regulatory entities (FDA, EMA) may have permits to initiate processes and sponsor institutes permits to accept initiated processes.
- Different process professionals may have different levels of clearance in a smart contract, established by their login (principal researcher or assistant physician).
- When dealing with large health files, IPFS can be used to avoid increasing network transaction costs and delays. This technology continues to preserve security and anonymity on the network, enhancing data protection.
- In complex networks, more than one smart contract can be used to decentralize tasks. For example, one contract can centralize only the inclusion of patients, another the management of doctors, another that of the partner institutions and a more general one the interaction between all the parties.
- Multi-level contracts can also be used to manage data from mobile sensors. In this way, a first contract can store the ID of the patient's mobile and receive only the variables that the local mobile application considers. If the parameters are met, the first contract will just send a communication to the second. The second, again if the requirements are met, can just send specific filtered data to the network. In this way, there are 3 information protection filters before the network and in these filters the information is completely unlinked from the person who generated it.
- In the case of web/cloud services that store part of the patient records (in various institutions) outside the Ethereum network, we may encounter the honest-but-curious-partner problem. This is basically that if the storer processes the data, it can circumvent the protection and keep metrics or part of the data that it processed. There may also be the case of an unreliable partner (not honest) that does not comply with what was agreed by the contract in Ethereum, since it is outside the network.

VII. Ethereum prototype for digital pathology

As we saw in the previous sections, there are numerous projects that have proposed solutions to manage patient information securely through Ethereum. Moreover, one of the first applications, MEDREC [19], was published in 2016 and developments have continued since then. However, there is only one strict protocol for digital pathology using Ethereum. That is why we will try to propose an outline for a second protocol in order to further extend the field of possibilities and future applications [30].

A. Users

To prevent many people from having access to sensitive information, only two types of users can be created. There will be two types of partner institutions, a pathology laboratory and a computational pathology center. For each institution, there can be only one registered user, the pathologist and the data engineer. In order to create an account that grants permissions, the user must have an institutional email, must have authorization from the ethics committee and once that email has been used, another account from the same institution cannot be created. Each user will have different permissions:

- The pathologist user will be able to upload images and image metadata such as tissue masks, diagnosis, pathology stratification and accompanying clinical data.
- The data engineer user will only be able to read the aggregated data, which in order to be able to read them must be randomized in order (avoid any type of patient load order being related to the order of medical consultation in the institution).

B. Layers

To start with, we will define our prototype system with 3 layers. The first will host the web services and will contain the interface through which the physical users (pathologists, data engineers) will make the multiple requests.The second layer will group the middleware, which in our case will be the APIs for digital contracts, IPFS and web servers. The third layer will contain the Ethereum network with the different nodes and the smart contracts, following a similar structure as the one explained earlier [28] for electronic health records.

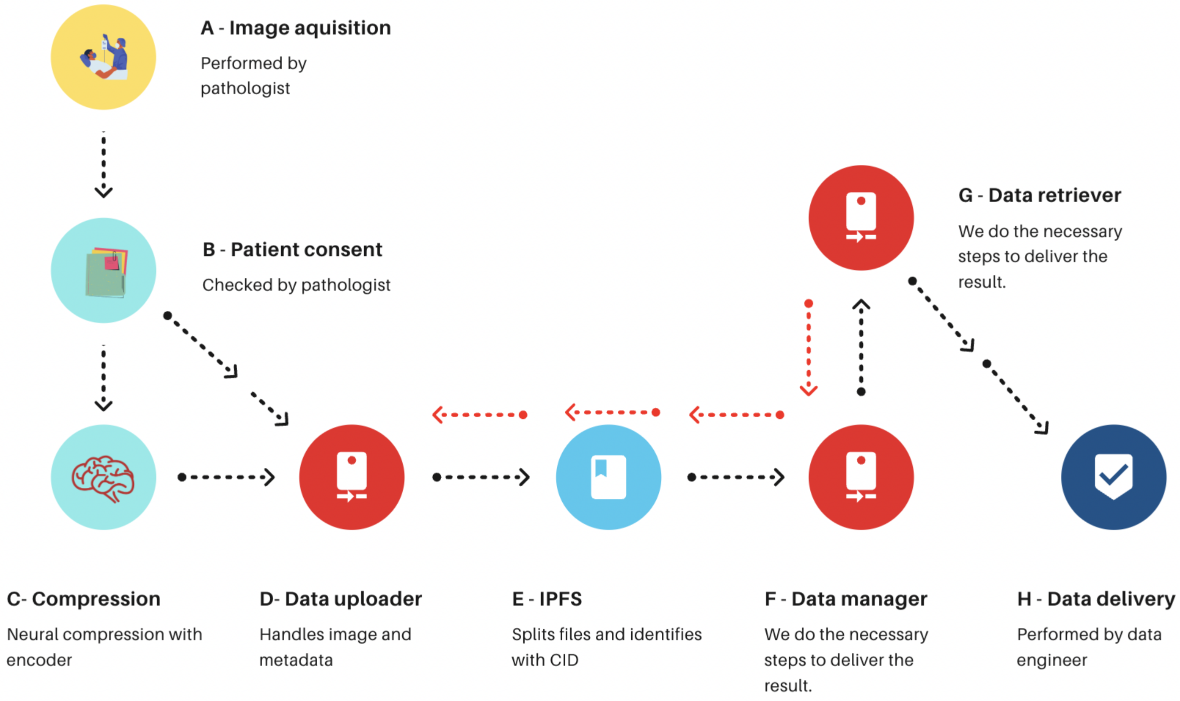
A - Image aquisition
Performed by pathologist
B - Patient consent
Checked by pathologist
C- Compression
Neural compression with encoder
D- Data uploader
Handles image and metadata
E - IPFS
Splits files and identifies with CID
F - Data manager
We do the necessary steps to deliver the result.
G - Data retriever
We do the necessary steps to deliver the result.
H - Data delivery
Performed by data engineer

Fig. 4. Representation of the network prototype. Black arrows indicate the forward flux of information. Red arrows indicate backward flux. Red icons (D, F, G) represent the smart contracts.

C. *Smart contracts*

Three different types of contracts will be created:

- The first of them will be the contract for uploading images. This contract will be in charge of authenticating the identity of the client that wants to upload the data, will verify the state of the informed consent, will request a verification of the integrity of the database, establish if the database can receive data and handle the reception of the complete data. Only the client who is a pathologist can upload data. The data package, in addition to the image, must contain the correct metadata. If there are missing fields in the metadata, the contract will reject the image upload. Once the image is sent, the client will not be able to undo the action. To remove an image from the database, you must send a request to the data manager to request removal.
- The second contract will be for data retrieval. Like the previous contract, this contract will validate the identity of the client (data engineer), request a verification of the integrity of the database (if there are images) and proceed to data retrieval. To avoid congestion and delays with other clients, once the request has been made, it will not be possible to make another request that contains the same data (images of the same pathology, with the same metadata). For exceptional cases, the data engineer client must communicate with the data manager who, if they approve the exception, can request the same data again.
- The third contract will be responsible for data management. This contract will communicate with the previous contracts, check the integrity of the database, receive and process requests for exception handling and process upload and retrieval requests. The integrity of the database will consist of checking the activity and availability of the IPFS and the availability of storage. This contract will receive the IDs and other sensitive information associated with the hashed images, to maximize protection. This will remove the link between IDs and real patients, who will be verified only by the image upload contract. When new data is received, all entries will be randomized to avoid linking images with upload orders.

D. *Informed consent*

To reinforce data protection, all images and metadata that are part of the network must have at least one instance in the process (diagnosis, medical consultation, treatment) where the patient has given their consent in writing for the use of the material. The consent must be aligned with the GDPR regulations [31] to maximize data protection and adhere to current regulations. The pathologist client must attach the informed consent to the image and this file will be checked by the data upload smart contract. Only if the consent is present, the smart contract will authorize the upload of the data.

E. *IPFS network*

This network will host each of the pathology image files that are accepted by the data management contract in a distributed manner. According to the operation of this network, the files

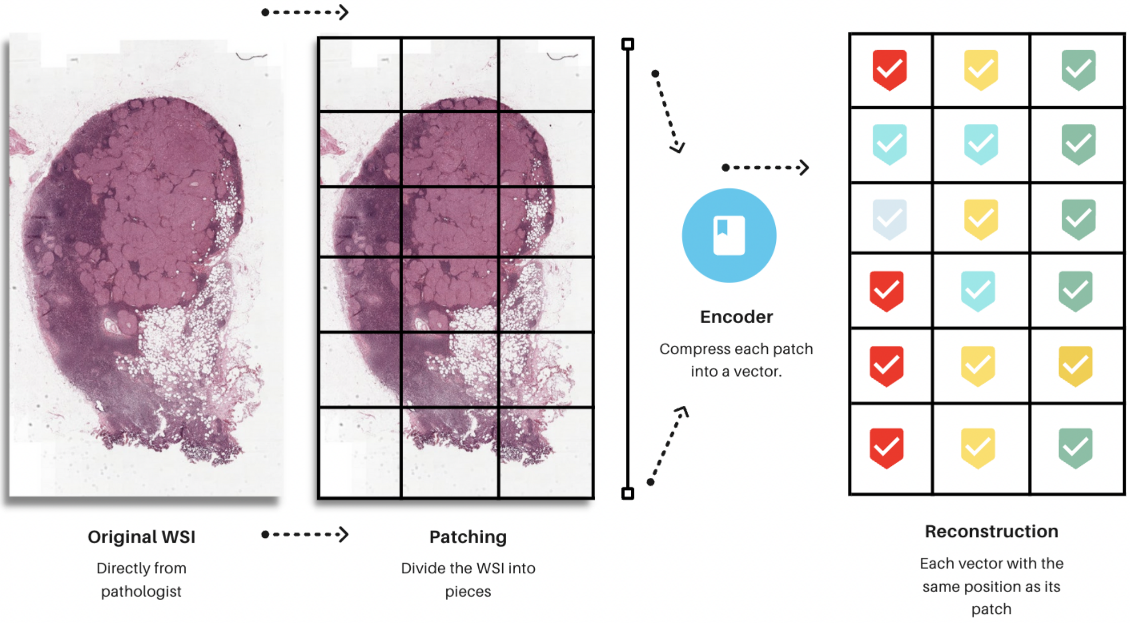
Original WSI
Directly from pathologist
Patching
Divide the WSI into pieces
Encoder
Compress each patch into a vector.
Reconstruction
Each vector with the same position as its patch

Fig. 5. Representation of the neural compression process for pathology WSI.

will be fragmented and hashed. Subsequently all fragments of the same file will be mapped with a content identifier (CID). The CIDs will be kept in the data management contract in order to perform the retrieval once the data engineer requests it. The status of this network, which will include node activity, free storage capacity and transfer availability, will be reported to the data manager each time a request for information is received. The data manager will have an identity verification system to prevent another unverified node from trying to make a data request.

F. Neural compression

Given that pathology images are very heavy and this makes storage and transfer very difficult, we will propose here the use of neural compression to speed up their handling and optimize the operation of the Ethereum network and the IPFS system. Neural compression is a procedure that uses an encoder to reduce the dimensions of an image and eliminate unnecessary information (noise) that does not benefit the task for which it is required [32], [33]. More specifically, the procedure consists of dividing the image into multiple patches, compressing those high-resolution patches into embedding vectors using an unsupervised neural network (encoder), and then reassembling a representation of the original image by placing the vectors in the same patch position from which they came (Fig. 5). If we have labels only at the patient level (one per image) then we will assign the label to all the vectors together rearmed. Now, if we have a mask, we can specify which of these vectors fall in a region of interest (for example tumor) and which of these fall in a region of normal cells [32], [34].

G. Prototype's fragility

Our prototype proposes the use of Ethereum blockchain, IPFS and neural compression to protect data privacy as much as possible, speeding up the availability of information. However, like any prototype, it could have several negative aspects. These negative aspects will be listed below, to facilitate a future discussion about their mitigating factors:

- As cited by several authors, smart contracts are subject to an inherent rigidity in their programming structure. This could be an inconvenience in the real world, when some conditions of the contract cannot be fulfilled 100% and this results in an impediment both for uploading images and for data retrieval. For example, if there is an error in the uploading process and when an exception is submitted for a new procedure, the data management contract does not accept the new procedure, that data package probably would not be uploaded.
- With the addition of hospitals to the network, new nodes must be created. This increases the complexity of the network and requires that the resources available for the maintenance of the servers grow. With that growth come associated costs that must be taken into account when projecting into the future.
- Although the requirement of informed consent increases data protection and ensures that patients are aware of the uses of their data, this could lead to delays in data loading. The data uploader contract will not proceed with the upload of an image and its metadata, if it does not receive the corresponding informed consent beforehand.
- Regarding neural compression, this technique is very useful to extract the essential information from images and thus streamline transactions in the blockchain. How-

ever, it is important to note that the efficiency of the representations (embedding vectors) depends on the type of encoder used. Similar results will not be generated by encoders coming from a network trained in general image recognition or pathology image recognition. In turn, the number of examples in which the encoder was trained before (when it was a convolutional neural network) also has an influence. In the latter case, the availability of large datasets of general images is greater than that of pathology images [32], [34]. Therefore, directly storing the representation could only be used for training, but it would not be optimal for other computational pathology tasks, such as model explainability.

- Single access per institution can be a bottleneck if we consider an institution with several pathology laboratories that produce many images per day. This could be solved by centralizing the loading of images in a single entity within the institution (1 pathologist) whose dedication is exclusive.
- The IPFS system is ideal for storing large data, promoting secure and decentralized storage. However, this system requires that if the required file is stored entirely on a terminal, that terminal be online. If, on the other hand, there are several copies in several terminals, then at least one of them must be online. This could be a drawback for the use of this prototype in countries where connections are fluctuating [35].

VIII. Conclusion

After having gone through the different sections of this article, we can stay with a brief idea of Ethereum blockchain and its possible application in the field of digital and computational pathology. As we are already aware, digital pathology produces a large number of image files associated with patient metadata and computational pathology uses them massively to train algorithms. Being able to introduce a partner like Ethereum with the support of IPFS brings a considerable degree of anonymization and decentralization to the process. Moreover, the use of multiple smart contracts where the identifying information of patients is kept completely isolated in one of them, reinforces privacy. Consequently, if we add a degree of randomization to the aggregate data, we are further promoting dis-identification by not even being able to correlate the order of data entry with real patients. However, it is important to continue working on the prototypes, carry out extensive tests to analyze the response of the oracles to 'humanize' the smart contracts and thus guarantee a proper functioning that does not endanger privacy or the progress of scientific research. .

Acknowledgment

The author would like to thank Dr. Ramiro Gilardino, Dr. Paula Marks and Dr. Inés Villalba for reviewing the manuscript.

References

[1] M. Cui and D. Y. Zhang, "Artificial intelligence and computational pathology," *Laboratory Investigation*, vol. 101, no. 4, pp. 412–422, 2021.

[2] G. Litjens, P. Bandi, B. Ehteshami Bejnordi, O. Geessink, M. Balkenhol, P. Bult, A. Halilovic, M. Hermsen, R. van de Loo, R. Vogels *et al.*, "1399 h&e-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset," *GigaScience*, vol. 7, no. 6, p. giy065, 2018.

[3] M. D. Zarella, D. Bowman, F. Aeffner, N. Farahani, A. Xthona, S. F. Absar, A. Parwani, M. Bui, and D. J. Hartman, "A practical guide to whole slide imaging: a white paper from the digital pathology association," *Archives of pathology & laboratory medicine*, vol. 143, no. 2, pp. 222–234, 2019.

[4] M. G. Hanna, O. Ardon, V. E. Reuter, S. J. Sirintrapun, C. England, D. S. Klimstra, and M. R. Hameed, "Integrating digital pathology into clinical practice," *Modern Pathology*, vol. 35, no. 2, pp. 152–164, 2022.

[5] Y. J. Kim, E. H. Roh, and S. Park, "A literature review of quality, costs, process-associated with digital pathology," *Journal of exercise rehabilitation*, vol. 17, no. 1, p. 11, 2021.

[6] E. Abels, L. Pantanowitz, F. Aeffner, M. D. Zarella, J. van der Laak, M. M. Bui, V. N. Vemuri, A. V. Parwani, J. Gibbs, E. Agosto-Arroyo *et al.*, "Computational pathology definitions, best practices, and recommendations for regulatory guidance: a white paper from the digital pathology association," *The Journal of pathology*, vol. 249, no. 3, pp. 286–294, 2019.

[7] A. B. Farris, J. Vizcarra, M. Amgad, L. A. Cooper, D. Gutman, and J. Hogan, "Artificial intelligence and algorithmic computational pathology: an introduction with renal allograft examples," *Histopathology*, vol. 78, no. 6, pp. 791–804, 2021.

[8] T. Sorell, N. Rajpoot, and C. Verrill, "Ethical issues in computational pathology," *Journal of Medical Ethics*, vol. 48, no. 4, pp. 278–284, 2022.

[9] D. Tellez, G. Litjens, P. Bándi, W. Bulten, J.-M. Bokhorst, F. Ciompi, and J. Van Der Laak, "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology," *Medical image analysis*, vol. 58, p. 101544, 2019.

[10] S. W. Jahn, M. Plass, and F. Moinfar, "Digital pathology: advantages, limitations and emerging perspectives," *Journal of Clinical Medicine*, vol. 9, no. 11, p. 3697, 2020.

[11] J. Jay, "Low signal latency in optical fiber networks," in *Corning Optical Fiber, Proc. of the 60th IWCS, Conference, Charlotte, NC, USA,(6-9 Nov. 2011). Citeseer*. Citeseer, 2011.

[12] C. Li, A. Burchard, and J. Liebeherr, "A network calculus with effective bandwidth," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1442–1453, 2007.

[13] "Ethereum development documentation." [Online]. Available: https://ethereum.org/en/developers/docs/

[14] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger."

[15] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[16] S. Halevi and H. Krawczyk, "Randomized hashing and digital signatures," 2006.

[17] S. Al-Kuwari, J. H. Davenport, and R. J. Bradford, "Cryptographic hash functions: Recent design trends and security notions," *Cryptology ePrint Archive*, 2011.

[18] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the works of leslie lamport*, 2019, pp. 203–226.

[19] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd international conference on open and big data (OBD)*. IEEE, 2016, pp. 25–30.

[20] S. Batchu, K. Patel, O. S. Henry, A. Mohamed, A. A. Agarwal, H. Hundal, A. Joshi, S. Thoota, and U. K. Patel, "Using ethereum smart contracts to store and share covid-19 patient data," *Cureus*, vol. 14, no. 1, 2022.

[21] S. Batchu, O. S. Henry, and A. A. Hakim, "A novel decentralized model for storing and sharing neuroimaging data using ethereum blockchain and the interplanetary file system," *International Journal of Information Technology*, vol. 13, no. 6, pp. 2145–2151, 2021.

[22] I. A. Omar, R. Jayaraman, K. Salah, and M. C. E. Simsekler, "Exploiting ethereum smart contracts for clinical trial management," in *2019*

IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA). IEEE, 2019, pp. 1–6.

[23] I. M. Akbar, A. Bhawiyuga, and R. Siregar, "An ethereum blockchain based electronic health record system for inter-hospital secure data sharing," in *6th International Conference on Sustainable Information Engineering and Technology 2021*, 2021, pp. 226–230.

[24] N. Rauta and K. Shah, "Implementation of ethereum blockchain in healthcare using ipfs," *Pulse*, vol. 2, no. 2, 2021.

[25] G. Gürsoy, C. M. Brannon, and M. Gerstein, "Using ethereum blockchain to store and query pharmacogenomics data via smart contracts," *BMC medical genomics*, vol. 13, no. 1, pp. 1–11, 2020.

[26] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, no. 7, pp. 1–7, 2018.

[27] D. Yonathan, D. Husna, F. A. Ekadiyanto, I. K. E. Purnama, A. N. Hidayati, M. H. Purnomo, S. M. S. Nugroho, R. F. Rachmadi, I. Nurtanio, and A. A. P. Ratna, "Design of decentralized application for telemedicine image record system with smart contract on ethereum," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, 2021.

[28] A. Theodouli, S. Arakliotis, K. Moschou, K. Votis, and D. Tzovaras, "On the design of a blockchain-based system to facilitate healthcare data sharing," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 1374–1379.

[29] R. Gupta, A. Shukla, and S. Tanwar, "Aayush: A smart contract-based telesurgery system for healthcare 4.0," in *2020 IEEE International conference on communications workshops (ICC Workshops)*. IEEE, 2020, pp. 1–6.

[30] S. QAMAR, T. NAEEM, and P. PARK, "Ethereum-blockchain based distribution and management system for ai classified biomedical images," *The Japanese Journal of Ergonomics*, vol. 57, no. Supplement-2, pp. K5–K5, 2021.

[31] Sep 2019. [Online]. Available: https://gdpr-info.eu/

[32] D. Tellez, G. Litjens, J. van der Laak, and F. Ciompi, "Neural image compression for gigapixel histopathology image analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp. 567–578, 2019.

[33] N. Pawlowski, S. Bhooshan, N. Ballas, F. Ciompi, B. Glocker, and M. Drozdzal, "Needles in haystacks: On classifying tiny objects in large images," *arXiv preprint arXiv:1908.06037*, 2019.

[34] D. Tellez, D. Höppener, C. Verhoef, D. Grünhagen, P. Nierop, M. Drozdzal, J. Laak, and F. Ciompi, "Extending unsupervised neural image compression with supervised multitask learning," in *Medical Imaging with Deep Learning*. PMLR, 2020, pp. 770–783.

[35] G. S. Reen, M. Mohandas, and S. Venkatesan, "Decentralized patient centric e-health record management system using blockchain and ipfs," in *2019 IEEE Conference on Information and Communication Technology*. IEEE, 2019, pp. 1–7.