Deploying Metadata on Blockchain Technologies

Elena García-Barriocanal, Salvador Sánchez-Alonso,
and Miguel-Angel Sicilia(✉)

Computer Science Department, University of Alcalá, Polytechnic building,
Ctra. Barcelona km. 33.6, 28871 Alcalá de Henares, Madrid, Spain
{elena.garciab,salvador.sanchez,msicilia}@uah.es

Abstract. Metadata repositories and services support the key functions required by the curation of digital resources, including description, management and provenance. They typically use conventional databases owned and managed by different kinds of organizations that are trusted by their users. Blockchains have emerged as a means to deploy decentralized databases secured from tampering and revision, opening the doors for a new way of deploying that kind of digital archival systems. In this paper we review and evaluate the functions of metadata in that new light and propose an approach in which a blockchain combined with other related technologies can be arranged in a particular way to obtain a decentralized solution for metadata supporting key functions. We discuss how the approach overcomes some weaknesses of current digital archives, along with its important implications for the management and sustainability of digital archives.

Keywords: Blockchain · Metadata · Decentralization · Provenance · Trust · Ethereum · IPFS · BlockchainDB

1 Introduction

Blockchain technologies have emerged as a means to deploy applications that provide decentralized database functions secured from tampering and revision and that are able to operate on open networks in a completely decentralized way. Complementary technologies as decentralized file systems (as the *Interplanetary File System*, IPFS [1]) and blockchain layers for databases has also been developed to fill particular gaps that are currently not provided currently by public blockchain system implementations in an efficient way. The first successful application of blockchain technology has been the Bitcoin crypto-currency, but the applicability of the underlying distributed ledger technology spans many other areas beyond [15], that are currently being explored.

Digital repositories and systems serve different functions, notably the permanent archival of digital resources and the provision of descriptions that ease their retrieval, encoding provenance and descriptive information as metadata. There exist a number of software solutions supporting various forms of management of metadata and digital resource preservation. Concretely, open source ones as

E. Garoufallou et al. (Eds.): MTSR 2017, CCIS 755, pp. 38–49, 2017.
https://doi.org/10.1007/978-3-319-70863-8_4

DSpace, EPrints or Fedora [10] are widespread, and they fundamentally serve the role of institutional repositories. The actual usage of these systems nowadays assume trust in the repository owner, and rely on the institution or network of institutions for its sustainability along time. This is not without problems, as some of them might fail in funding the costs of their running activity [2,12] or they may be exposed to other risks concerning the concern for sustainability [3].

Blockchain and associated technologies provide a new type of platform to overcome some of the problems of current repository technology and are thus promising candidates to build a decentralized approach to the archival of digital resources. However, reformulating the current system of digital archives, aggregators and services requires a careful consideration of the functions of metadata, its desirable properties, and the extent to which different technologies are able to support them. Further, blockchain-based decentralized systems are not without risks, especially as they are built around systems of incentives for the participants in the network, so sustainability should be incorporated in their design.

In this paper, we report on the analysis and early proof-of-concept implementation of a decentralized metadata system that considers the different functions of metadata as point of departure. We start from an account of these functions and then assess the fit of different kind of technologies. Then, the fundamental data models, interfaces and processes are described and contrasted with current practice. We only address the base mechanisms for deploying metadata on blockchains and not the required systems of incentives and end-user systems needed for a complete solution, as those should be subject to separate inquiry and research.

The rest of this paper is structured as follows. Section 2 provides the rationale and requirements for the approach presented. Then, the proposed approach is described in Sect. 3. Section 4 describes a proof-of-concept implementation using concrete technologies. Finally, conclusions and outlook are provided in Sect. 5.

2 Background and Rationale

2.1 Blockchains as Shared, Immutable Repositories

The term “blockchain” refers to the core data structure of a category of decentralized database architectures that rely on cryptographic techniques and distributed consensus to provide tamper-proof distributed ledgers. The first widespread blockchain application was that of Bitcoin [15], that used that technology to implement the possibility of interchanging a token (a digital currency) among non-trusted parties without the need of a central authority and preventing double spending.

The application of blockchains have since Bitcoin inception be extended to the notion of “smart contracts” (first proposed by Szabo [14]) in which non-trusted parties can interact with the blockchain for different kinds of transactions including some logic that can be implemented in Turing-complete programming languages. This supports the implementation of token interchange systems as Bitcoin, but also many other applications as voting, registries or future markets to name a few. The main current exponent of such technology is Ethereum [19], that we consider here as the foundation for the analysis.

Here we are interested fundamentally in blockchains as immutable archives that do not rely on a trusted party, in contrast with current repository systems. While this is discussed here essentially from its practical, technological implications, it has the potential to impact the economics of current archival institutions, their funding models, and eventually, their archival cycles and responsibilities. There are a few recent proposals for digital repository architectures based on blockchains [9,18], but the core underlying representational commitments of metadata have not been discussed to date.

2.2 Metadata Functions and Requirements

Metadata schemas are the specifications of how metadata descriptions should be interpreted, syntactically and semantically. While different schemas developed and evolved by different communities may differ in their objectives (e.g. an schema for museum collections differs in its aims from one devised for environmental data), their resulting metadata serves a number of functions in all cases. Here we use the classification in [5] that in turn consolidates classifications from other authors. The departure point for the approach presented here is thus mapping those functions to required core architectural properties, which are summarized in Table 1.

Table 1. Types, functions and architectural requirements for metadata

Type	Function facilitated	Architectural req.
Identification/Description	Resource discovery Information retrieval	Decentralized identification Dereferencing Indexing
Administrative	Resource management	Pricing, interchange
Terms and conditions	Resource usage	Rights management
Content rating	Resource use by appropriate audiences	Labelling
Provenance	Authentication and related	Proof of statement
Linkage/relationship	Linking with related resources	Referencing
Structural	Software and hardware needs	Media typing

Among the requirements in Table 1, a first analysis results in three categories[1] depending on how they are influenced by blockchain and related decentralized technology:

- Requirements not directly supported by the blockchain network. This includes *media typing*, as these is related to the long-term capability to understand,

[1] It should be noted that we are considered concretely Ethereum as the reference model for public blockchain technology combined with a decentralized P2P storage model as IPFS. The analysis reported here may be different considering other emerging blockchain technologies.

process and render digital objects. Decentralized storage can only refer in the metadata to the software of specifications needed, but this remains a matter of preserving the capability of processing that has been addressed before typically via emulation or migration [16]. Also *labeling* is out of the core blockchain implementation, as this entails interpreting the needs of different audiences, which is an interpretive assessment by nature.

- Requirements that are impacted by decentralized solutions. *Indexing* is not a capability directly supported by current public blockchains and thus requires some additional infrastructure, but is dependent on referencing the blockchain for tamper-proof provenance. The same occurs with referencing (linking) that takes a different form if referencing resources in decentralized systems [13], but it is not fully resolved with it and requires additional conventions. *Pricing*, *rights management* and *interchange* are not granted as a direct consequence of using a blockchain, but the blockchain enables the creation of new mechanisms for such applications. The music industry is the focus of inquiry on the application of rights management, but there are still not clear, comprehensive and widespread blockchain solutions for it but only some experimental work [4].
- Requirements that are directly supported by the blockchain network. These include *identification*, *dereferencing* and *proof of statement* as functions that are directly supported by a combination of a blockchain and a decentralized file system as described below.

One additional key issue in metadata is that of interoperability. While interoperability at a syntactic or data transfer level is tackled by mappings or transformations, semantic interoperability poses different challenges [6] requiring model mapping at different levels. Decentralization entails the exposure of a heterogeneity of autonomous, incompatible media repositories and it is unlikely that there will ever exist a single agreed-upon metadata schema (if it ever exists it should be based on a system of incentives that is still to be conceived). This entails that the interpretation of the metadata also requires that the schemas and ontologies or terminologies used by them are also deployed in immutable decentralized systems, but this has been focus of previous research [13] and is not considered in further detail here.

3 Proposed Approach

3.1 Resources and Identifiers

At a very basic level, metadata can be considered an statement about a *resource*. Digital resources[2] are usually identified by different means, as URIs or DOIs. However, these have two problems associated: (a) they rely on some trusted party

[2] Here we limit our discussion to digital resources, but they could be surrogates of physical entities, provided that these surrogates can be unambiguously linked to its corresponding physical counterpart.

or authority, as in the case of the DOI and (b) some of them (as URIs) cannot be guaranteed (and are not intended to) retrieve the same resource over time.

In a context of untrusted parties, digital resources should be identified via mechanisms that uniquely identify them from their content. As we are dealing with resources that are coded and represented by different media types and conventions, using hashes of their byte content appears as a universally applicable approach. An example of this is the addressing used in IPFS, that relies on a P2P architecture to store (almost) permanently digital resources in decentralized storage. Existing or future identification systems may eventually link other identifiers to the actual hashes, serving thus as effective aliases. This brings the decentralized file system as the first component of the approach.

It should be noted that metadata records are themselves digital resources, so that they can also be identified by content.

An additional desirable requirement is that identifiers are dereferenceable, i.e. that they can be used to resolve the actual resource. This is a basic principle in the Web of Linked Data, for example. In consequence, a basic model for identifiers is that of introducing a notion of decentralized *handlers*, defined as follows (using Solidity, from here on we use Ethereum as the smart contract technology):

```
struct Handler {
   bytes  hash;     // identifier
   string service;  // location
   string meta;     // processing info
}
```

The field `meta` is a placeholder for additional information required for dereferencing the item from the given `service`. The requirement on the service is that it provides decentralized, tamper resistant permanent storage, thus avoiding the problems of availability that hamper the usefulness of approaches as that of Linked Data [11]. An example service that could be used is IPFS. The `hash` will be an IPFS hash, service will be some conventional reference to IPFS and no additional information would be required in this case. Note that `meta` could be used for supporting *media typing* functions (requiring some new or existing conventions), a simple case is providing the MIME type, or some reference to the metadata schema if the resource is actually a piece of metadata. For example, a simple handler representation may be:

```
Handler h = Handler({hash:"QmYwAPJzv5CZsnA625s3Xf2nemtYgPpHd...",
                     service:"ipfs-base58", meta:"audio/mpeg3"});
```

Note that this does not resolve the problem of the different representations of intellectual works which remains a matter of modeling and ontology, as addressed for example in the FRBR model [17]. This simple representation also allows both for public schemes or private ones, as the underlying representation is just a stream of bytes which may be encrypted itself or require some special non-public software for processing. However, we limit ourselves here on public descriptions and open access resources.

3.2 Statements and Provenance

Once we have an account by which resources and their metadata can be identified, the next important feature is that of provenance. This has two levels: (a) allowing for metadata authors (individuals but also institutions as libraries or archives) to proof that a given metadata is provided by them, and (b) describing the provenance that the metadata author claims to be associated to the object. The former is in some schemas as IEEE LOM known as "meta-metadata". Once (a) can be established, then (b) becomes a matter of trust in the issuer of the metadata.

We are here concerned with (a), which is the basic problem of authenticity of metadata. A blockchain can be the appropriate solution here as it allows for metadata authors to add transactions to the blockchain in which they can cryptographically proof their provenance, and that cannot be removed, so services relying on them are not dependant on their availability. It should be noted that blockchains can also be used to claim authorship priority just by depositing hashes to files in transactions, but this is not our focus here.

The relation of a metadata record to a resource then becomes a claim with a core data model sketched as follows.

```
enum Verb {Add, Retract, Replace}
struct Claim {
  Handler resource;
  Handler metadata;
  Verb OP;
  uint256 timestamp;
}
```

Then, different operations can be implemented using a smart contract in a straightforward way:

```
contract MetadataRepository {
  // ...
  mapping(address => mapping(bytes => Claim[])) statements;

  function claim(bytes rhash, string rservice, string rmeta,
                 bytes mhash, string mservice, string mmeta){
      var resource = Handler(rhash, rservice, rmeta);
      var meta = Handler(mhash, mservice, mmeta);
      oraclize(resource); oraclize(meta);
      var claim = Claim(resource, meta, Verb.Add, block.timestamp);
      statements[msg.sender][resource.hash].push(claim);
  }
function retract(...){
      // ...
  }
function check(address curator, bytes rhash)
                 constant public returns (bool){
      //...
}
```

The key here is that the transactions need to check that the resource is really available using an external oracle. A possible solution is for example that of using *Oraclize*[3], that provides proof for different external sources using in turn *TLSNotary*. This provides the basic functionality of registering claims (that may be a retraction of a previous one) of dereferenceable resources and metadata records. However, given the characteristics of public blockchains as Ethereum this cannot be used to implement indexing functions.

Metadata then becomes a set of series of immutable claims, that reflect the current description of the digital object by a network participant, where claims may be revised, so that there is some non-monotonicity entailed that require some handling of fact retraction (as in e.g. [8]) if metadata is used in reasoning systems. Each series comes from an address, which is in the blockchain a pseudonym. However, institutions or individuals may disclose their physical world identity via means as digital certificates.

3.3 Search and Discovery

Resource search and discovery is currently done in archives by using search technology on top of conventional databases, and using typically harvesting protocols as OAI-PMH for mirroring, sub-setting or aggregating archives. Real-time queries using rich syntaxes require an additional layer on top of the blockchain and decentralized storage of resources. Using conventional indexing and retrieval engines as Apache Lucene is an option, but it requires a copy of the resources to be indexed, thus becoming a trusted party. An option that brings some of the benefits of blockchains is using blockchain databases, that add functionalities to conventional scalable database systems. BigchainDB adds a layer on top of RethinkDB or MongoDB featuring a number of query facilities.

Deploying databases as BigchainDB as front-ends should attempt to bring decentralization and tamper resistance to the query layer, which in that case is facilitated by *node diversity*.

BigchainDB is optimized to the transfer of assets in high loading cases, so it would become an ideal candidate to resource usage and management functions of metadata. While this can also be done using platforms as Ethereum that support arbitrarily complex contracts, it brings scalability.

While BigchainDB is not currently featuring a full-fledged search language, its asset consensus model can be used to implement mirrors. There are two use cases here:

(a) The creators of metadata records submit CREATE transactions for each of their records in the blockchain.
(b) The owner of the aggregator or surrogate submit the CREATE transactions and include a back-reference to the transaction and block in the original blockchain.

[3] http://www.oraclize.it/.

Case (a) has the interesting feature of supporting cryptographic transfer of the metadata (asset in BigchainDB jargon). Case (b) has the benefit of allowing third parties build their systems independently while still providing a way by which the records can be trusted by inspecting the original blockchain transaction. In this second case, the consensus protocol of the blockchain database could be augmented to reject transactions not consistent with the original blockchain, which can be verified by using the `check` operation described above.

In both cases, as BigchainDB is a front-end for a NoSQL engine, the query facilities of that engine can be used for trusted records.

3.4 Semantics

Semantics is introduced by the use of vocabularies, terminologies or ontologies (we can collectively refer to all of them as Knowledge Organization Systems, KOS) in particular metadata elements. The implications is that terminologies can be considered another resource that must be subject to decentralized storage and attestation of authenticity via blockchain transactions. In [13] the distributed storage part is discussed, and the proof of authenticity could be achieved by similar means to that of the metadata, but in this case, just registering the different versions of the KOS.

The problem is that current metadata is now using typically URIs or other codes for the referencing. This has many problems, as in many cases these codes do not refer to a concrete version, or in some cases are not dereferenceable or are at risk of becoming unavailable. However, converting all those references is not trivial and changes the contents (and thus the handlers) of metadata records, which makes this seldom viable in the short-term.

3.5 Other Functions

Media typing in the approach presented is maintained as an informative function of the metadata elements, but it is not dealt with specifically. The field `meta` in handlers may be used for that purpose, but a more complex approach would be that of using a similar architecture for storing media type declarations associated to media type descriptions, converters or even software artifacts that can be used to process them, supporting different preservation strategies [7].

Content rating allows for the real-time selective dissemination of resources depending on the user or audience. We have not found any blockchain-specific advantage for this kind of functionality, as it is typically deployed by players. Those players may use the information on the blockchain to retrieve trusted information on ratings, but this is not different from the retrieval of any other kind of description.

Linkage is also not dealt here. The common use of linkage in metadata uses URIs or other conventional identifiers along with predicate vocabularies. Making these links first-class in a blockchain solution would require the use of handlers inside the metadata or as separate link metadata records (to avoid the problem of circular references). This, as mentioned in the section about semantics, has been discussed elsewhere in the context of linked data over IPFS [13].

4 Example Implementation

A proof of concept design prototype was built using a combination of the Ethereum blockchain, the IPFS decentralized file system, and the BigchainDB database.

4.1 Metadata Repositories on Ethereum and IPFS

The registration of metadata claims and its eventual update is realized using the `MetadataRepository` contract discussed before, devised and deployed over a Ethereum test net. One or several instances of the contract may be deployed in the network, and as they become autonomous from the original creator, their number is not a problem.

Preservation of metadata requires the availability of the schemas that are used for the expression of metadata. This is usually achieved in metadata registries. A metadata registry could be implemented in Ethereum with a similar `MetadataRegistry` contract in which schema curators or communities could deploy the different versions of their schemas. Then, a higher level of preservation integrity could be achieved by extending the `Claim` model with a mandatory additional handler to the metadata schema used. This would enable checking the validity of metadata records before they are registered as claims, and guarantee that the specifications are also preserved. However, this may be considered unpractical for the registering of legacy metadata in the blockchain architecture, which is often not free of errors.

This entails that the preservation of the claims is supported by the sustainability of the blockchain, so it rests on a global incentive system. The cost for curators is then that of registering the claims, which have a cost (used by the Ethereum's *gas* required to execute the transactions). This is in a sense inverting the cost, from maintaining the database to registering or updating new elements which was before with no inherent cost.

4.2 Indexing and Search on BigchainDB

The model for registering metadata claims as a collection of series supports metadata curation, but contracts are not currently devised to perform iterations over mappings or return large collections. One option may be that of using external blockchain *explorers*, that read the database directly and scan it. However, explorers are not adequate as query systems, and this is where other solutions may have a place, concretely databases that use an additional consensus layer. Essentially, these provide a middle ground so that it is possible to copy fragments of the claims of the blockchain to a database dedicated for query, while retaining a degree of security and providing a way to check provenance referring to the original blockchain record.

BigchainDB[4] stores digital *assets*, which are essentially JSON documents with some optional metadata. In our case, we would want to include metadata records as assets. The basic functionality is then that of storing metadata records using `CREATE` transactions, and eventually using `TRANSFER` transactions to change their ownership. As transactions are digitally signed by its owner, this guarantees provenance and enables a degree of administration. The `metadata` part of the asset refers to the handler and the location of the claim (the transaction hash of the claim in this case):

```
{"hash": "QmYwAPJzv5CZsnA625s...",
"service": "ipfs",
"meta":"",
"claim":"0xbd53b39f64ce9a96d..."}
```

It should be noted that the transaction hash provides sufficient reference, but the rest of the information is used for convenience and ease of retrieval in queries.

Then, the `data` element of the asset would be a representation of the metadata part of the handler. This entails the need for transforming records in other metadata language bindings as XML or RDF into JSON. However, this is a limitation of the database backend of the solution, and not an inherent constraint of the overall architecture.

The adaptation required a modification in the consensus rules of BigchainBD by writing and including a simple plugin. This was limited to a change in the `validate_transaction()` method on a subclass of `BaseConsensusRules`. The method then could do the checking that the document to be included is in the Ethereum component, by using `check`[5]. As the registration of the metadata in that part used the `oraclize()` calls in the Ethereum contracts, then we guarantee that the metadata can be retrieved via its handler and it is a legitimate claim.

Once a metadata record is in the database, we are able to know: (a) its owner, i.e. the signer of the transaction or a subsequent recipient, (b) that the metadata record is legitimate and current up to the given timestamp, and (c) that we can at any moment query in the Ethereum component if the record is updated or retracted.

Then, it is possible to use the underlying query mechanisms of the storage component (in our case, a *MongoDB* backend) to query or build search systems on top of it.

[4] https://www.bigchaindb.com/.

[5] Or alternatively exploring the blockchain, but that would be inefficient.

5 Conclusions and Outlook

Blockchain technologies considered as immutable decentralized databases have the potential to change the practices and systems used for archival functions, both for the storage of the digital resources and of the metadata describing them.

In this paper, we have sketched a possible architecture that fulfills with three different components several of the key functions of metadata: decentralized identification, deferencing, proof of statement and (separately) indexing. The discussion has stayed at a generic level, but it could be extended to cover domain-specific cases that may require additional processing before the claims are included in the blockchain or that may include additional information.

It is still too early to value if a blockchain architecture as the one presented here is acceptable as an alternative to current centralized systems. But in any case, it represents an option to achieve higher levels of availability, transparency and tamper resistance, which would solve some of the problems of current metadata systems built on conventional databases.

References

1. Benet, J.: IPFS-Content Addressed, Versioned, P2P File System. arXiv preprint arXiv:1407.3561 (2014)
2. Burns, C.S., Lana, A., Budd, J.M.: Institutional repositories: exploration of costs and value. D-Lib Mag. **19**(1/2), 1–17 (2013)
3. Eschenfelder, K.R., Shankar, K., Williams, R., Lanham, A., Salo, D., Zhang, M.: What are we talking about when we talk about sustainability of digital archives, repositories and libraries? Proc. Assoc. Inf. Sci. Technol. **53**(1), 1–6 (2016)
4. Fujimura, S., Watanabe, H., Nakadaira, A., Yamada, T., Akutsu, A., Kishigami, J.J.: BRIGHT: a concept for a decentralized rights management system based on blockchain. In: Proceedings of the IEEE 5th International Conference on Consumer Electronics, pp. 345–346 (2015)
5. Greenberg, J.: Understanding metadata and metadata schemes. Cat. Classif. Q. **40**(3–4), 17–36 (2005)
6. Haslhofer, B., Klas, W.: A survey of techniques for achieving metadata interoperability. ACM Comput. Surv. (CSUR) **42**(2), 7 (2010)
7. Hunter, J., Choudhury, S.: Implementing preservation strategies for complex multimedia objects. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 473–486. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45175-4_43
8. Lam, E.S.L., Cervesato, I.: Modeling datalog fact assertion and retraction in linear logic. In: Proceedings of the 14th Symposium on Principles and Practice of Declarative Programming, pp. 67–78. ACM (2012)
9. Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., Njilla, L.: Provchain: a blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 468–477. IEEE Press (2017)
10. Pyrounakis, G., Nikolaidou, M., Hatzopoulos, M.: Building digital collections using open source digital repository software: a comparative study. Int. J. Digit. Libr. Syst. (IJDLS) **4**(1), 10–24 (2014)

11. Rajabi, E., Sanchez-Alonso, S., Sicilia, M.A.: Analyzing broken links on the web of data: an experiment with DBpedia. J. Assoc. Inf. Sci. Technol. **65**(8), 1721–1727 (2014)
12. Rinehart, K., Prud'homme, P.A., Reid Huot, A.: Overwhelmed to action: digital preservation challenges at the under-resourced institution. OCLC Syst. Serv. **30**(1), 28–42 (2014)
13. Sicilia, M.-A., Sánchez-Alonso, S., García-Barriocanal, E.: Sharing linked open data over peer-to-peer distributed file systems: the case of IPFS. In: Garoufallou, E., Subirats Coll, I., Stellato, A., Greenberg, J. (eds.) MTSR 2016. CCIS, vol. 672, pp. 3–14. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49157-8_1
14. Szabo, N.: Formalizing and securing relationships on public networks. First Monday **2**(9) (1997)
15. Tapscott, D., Tapscott, A.: Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World. Penguin, New York (2016)
16. Thibodeau, K.: Overview of technological approaches to digital preservation and challenges in coming years. CLIR Reports 107, The state of digital preservation: an international perspective, pp. 4–31 (2002). https://www.clir.org/pubs/reports/pub107
17. Tillett, B.: What is FRBR? A conceptual model for the bibliographic universe. Aust. Libr. J. **54**(1), 24–30 (2005)
18. Tran, A.B., Weber, X.X., Staples, M., Rimba, P.: Regerator: a registry generator for blockchain. In: Proceedings of the CAiSE-Forum-DC 2017, pp. 81–88 (2017)
19. Wood, G.: Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 151 (2014). http://www.cryptopapers.net/papers/ethereum-yellowpaper.pdf