Towards Blockchain Interoperability: Improving Video Games Data Exchange

Léo Besançon, Catarina Ferreira Da Silva, Parisa Ghodous
Univ Lyon, Université Claude Bernard Lyon 1
LIRIS, F-69100
Villeurbanne, France
{leo.besancon, catarina.ferreira-da-silva, parisa.ghodous}@liris.cnrs.fr

***Abstract*—Current solutions for designing and building decentralized blockchain applications lack interoperability. Consequently, blockchains and existing technologies do not integrate well in a unified framework. This integration is necessary to work around some of the blockchains constraints, such as scalability of transactions and ergonomics. Indeed, blockchains are not suitable for huge data storage, but there are distributed data storage solutions that can be used in a decentralized blockchain application. Regarding ergonomics, the use of blockchain technology should be in the background and transparent for users that may not know how to set up and secure a blockchain-based application.**

We propose an architecture aiming to easily link existing decentralized technologies and blockchains. We then discuss the impact of this architecture for the video game industry. As a result, we propose an original data representation of blockchain gaming assets in order to improve data exchanges in this industry.

***Index Terms*—Blockchain, interoperability, standards, video games**

I. Introduction

Blockchain (BC) is an innovative technology, which can have a high impact in numerous industries, such as healthcare [1], supply-chain [2], finance [3] and video games [4]. BC are append-only ledgers shared across a network of clients. Zheng et al. show in [5] some of the promises of this technology: decentralization, anonymity, persistency of information and auditability. However, they also highlight some of its current challenges: each node needs to keep the history of all the transactions made in the network, so the storage space keeps increasing, and the number of transactions that can be processed by the network is quite limited, around 7 transactions per second for Bitcoin. Deshpande et al. [6] also show the importance of resolving interoperability issues and developing standards in the BC field. This interoperability need can be found at multiple levels: a) between different BC, b) between different projects running on the same BC, and c) between BC and other technologies used to create decentralized applications.

BC are usually distributed, meaning the record of all transactions is replicated across multiple physical nodes. They can also be decentralized, meaning they are not controlled by a single entity (e.g. a government or company). In this case, control is determined by a consensus mechanism, which determines which blocks are considered valid for the network. In this paper, we mainly focus on decentralized blockchain applications (DBA).

978-1-7281-1328-9/19/$31.00 ©2019 IEEE

II. Related work

A. Interoperability between blockchains

Since the creation of Bitcoin, various new BC designs have tried to improve the technology. For example, EOS [7] uses Delegated Proof of Stake (DPoS) as a method for achieving consensus, which compromises decentralization in order to increase throughput. There is no unified standard between all BC designs, and this leads to the need for research regarding interoperability between BC [8]–[11]. Particularly, [12] proposes a layered architecture to improve communication between BC.

B. Interoperability in a particular field

Some research works also try to solve interoperability issues within a particular field. This is the case of [13], which analyzes how to leverage BC technology to improve data sharing between patients and healthcare institutions. Standardization efforts have come from the IEEE Blockchain Initiative [14] and the IEEE Standards Association [15]. For example, a framework focused on the Internet of Things is proposed in [16]. Concurrently, the Enterprise Ethereum Alliance (EEA) [17] designs specifications for BC clients, built for the Ethereum ecosystem, that could have enterprise usage. Unfortunately these proposals cannot be easily extended to other applications and applied to other BC. For example, the EEA aims to reach enterprises, so they don't take into account decentralization in their specifications [18]. The architecture proposed by IBM [19] has similar limitations: even though they include a public network for customers, the BC is managed by an administrator and its consensus is achieved by trusted participants.

Similarly, in the video game industry, Hoard [20] aims to better integrate BC in game engines for developers, as well as to abstract complexities of the BC for players. However they do not propose a generic framework for DBA. Approaching the problem from a semantic perspective, like [39] did for smart contract security, could improve interoperability.

C. Standardization of a particular blockchain

Protocols and commonly used interfaces in the BC space have mostly been standardized with a bottom-up approach. This is achieved mainly through Bitcoin Improvement Proposals (BIPs) and Ethereum Improvement Proposals (EIPs), as well as Ethereum Request for Comments (ERCs). The latter has seen several proposed standards for asset management, each built on the ERC-20 token standard [21].

In the video game industry, for non-fungible tokens, the most used token is ERC-721 [22] (e.g. collectible virtual objects such as CryptoKitties [23]). More recently, the ERC-1155 [24] proposes a unified interface able to manage both fungible and non-fungible assets.

Currently, these standards only cover the BC side of an asset, by specifying smart contract interfaces tools need to support in order to manage the assets. However, this approach is limited as it doesn't take into account the ecosystem as a whole. For example, collectible assets such as CryptoKitties are represented by images. These images are centralized and controlled by the servers of the project's company. This design choice could be challenged if any decentralized image storage standard was associated with the ERC-721 standard.

Indeed, most decentralized applications cannot use only BC technology, as it currently has several limitations. For example, the cost of permanently storing large amounts of data (e.g. images) on the Ethereum BC is prohibitive [25]. As a result, developers need to use BC only for the core processing of the application. Non-crucial processing, storage and other ancillary tasks have to be managed by other tools, such as distributed file storage solutions. Interoperability between a BC and these tools is a challenge, and it should be better taken into consideration when building standards.

D. Decentralized application example

The project Decentraland [26] uses a novel approach in order to create a virtual universe where users can purchase virtual land. Users can add 3D models, videos or sounds to their land, and script their content to interact with other users. Concretely, it is possible to design games that will run inside this virtual world. However, with the current specifications, developers need to design their games around the project's ecosystem. For example, the game's logic can only be programed using the project's language. Having a more generic design could help bring support for existing game engines more easily, and attract more content creators on the platform.

To summarize, to the best of our knowledge, no well-defined and complete architecture specifications for generic DBA have been proposed yet. As a result, integrating BC into video games is difficult to do with current technology and development tools. The National Institute of Standards and Technology confirms [27] that the current and future work regarding BC standardization concerns BC interoperability among others. This is why our work focuses on the proposition of a generic design for DBA that could be applied in most potential application of BC technology. We propose a design pattern to help developers better integrate BC into their applications.

III. Proposed architecture

The goal of the layered architecture shown in Fig. 1 is to provide the building blocks to support a DBA. It also avoids using a BC for non-suitable tasks. For sake of genericity, it is analogous to the OSI model, and each layer needs to communicate with its neighbors.

As seen previously in [25], BC technology isn't suitable for huge data storage. Fortunately, other decentralized tools can be used in addition to a BC to implement a complete application [28]. For example, static storage can be distributed on InterPlanetary File System [29] (IPFS) for free, as any node can choose the content they host. However, in practice, if a company does not want to lose the files needed for a product, they can host a node which acts as a gateway if no one else is incentivized to host the files. Other projects try instead to give economic incentive to store data. For example, FileCoin [30] is a BC layer built on top of IPFS, and Swarm [31] is an Ethereum Foundation project, aiming at bringing decentralized storage in the Ethereum ecosystem. One of the main challenges faced by the file storage layer is to correctly estimate the needs of an application, in terms of data availability, decentralization and data loss prevention.

For dynamic content and queries, we cannot directly use the decentralized storage tools mentioned above, as they only support static files. But several projects (OrbitDB [32], a layer on top of IPFS, and Gun [33]) make use of conflict-free replicated data types. These databases use data types which are suitable for a distributed environment, as it is always possible to resolve incoherence between peers, even when they go offline regularly. The Brewer's theorem states that a distributed database can have at most two of the following properties: consistency, availability and partition tolerance. Using this theorem, an application-specific choice has to be made in this layer in order to have the suitable trade-off for the considered use case.

The processing layer aims to validate data integrity and to manage crucial game mechanics (ensuring financial integrity,

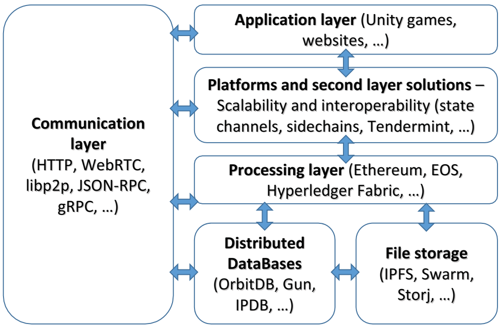
Application layer (Unity games, websites, …)
Platforms and second layer solutions – Scalability and interoperability (state channels, sidechains, Tendermint, …)
Communication layer (HTTP, WebRTC, libp2p, JSON-RPC, gRPC, …)
Processing layer (Ethereum, EOS, Hyperledger Fabric, …)
Distributed DataBases (OrbitDB, Gun, IPDB, …)
File storage (IPFS, Swarm, Storj, …)

Fig. 1. Decentralized blockchain application architecture

preventing cheating, etc.). This is done by any BC which supports smart contracts, e.g. Ethereum [34], EOS [7], Hyperledger Fabric [35]. The choice of the specific BC used in this layer has to be made by the developers depending on the use case. Indeed, it is sometimes preferable to prioritize throughput over decentralization or security. In these cases, it makes sense to use EOS or Hyperledger instead of Ethereum to process the application's smart contracts.

However the BC chosen in the processing layer may not have the exact properties needed. Platforms and second layer solutions can help to improve the interoperability between projects, as they can abstract the processing layer so that the application layer can interact with any of the possible BC. This abstraction is useful because it allows developers of projects built between different BC to use similar terminology, designs and mechanisms. Second layer solutions can also improve the scalability of the BC. Developers of DBA currently have two means to scale up the number of transactions. Both aim to avoid sending transaction on the public BC. There are sidechains [37] and state channels, popularized by the work made on the Lightning Network [36]. If a decentralized multiplayer game must have low-latency, the developers can implement a token ring network structure through state channels for player communications, instead of having all players transact on the BC.

The application layer is related to the interface the user connects to in order to use the application. It needs to interact with the BC but also to abstract complicated concepts regarding cryptography for the user. Indeed, developers cannot expect the users to know how the BC works and the consequences regarding security of their funds. This is why the ease of use and ergonomics of the technology is crucial. For example, gamifying wallet creation is an interesting way to make sure the user has stored securely the seed words for his wallet.

To enable communication between peers and these layers, a decentralized application should use peer-to-peer networking tools. Peer discovery may be difficult to achieve without a server, but it is possible to use Distributed Hash Tables (DHTs) as Guidi et al. did in [38].

Finally, we see that each layer of this architecture has its own challenges, both from research and engineering perspectives. Design and interface specifications would greatly help resolve these challenges.

IV. Application to the Video Game Industry

In the video game industry, BC can be used to improve trust between players and developers, as well as to reduce friction in the game implementation. For example:

- The founder of Ethereum, Vitalik Buterin realized the importance of decentralization when Blizzard unilaterally updated the rules regarding one of his World of Warcraft assets [40]. The player felt cheated by the developers because he understood he didn't truly own his assets.
- If players can interact with peer-to-peer technologies, game developers don't have to pay for expensive servers as all the processing can be done by the players' machines.
- BC are especially suitable for ownership management. Games like Lunar Mines [41] take advantage of BC by letting players easily craft and trade items with other players. This type of game mechanics could be done in a non-BC game, but developers would need to recreate the asset ownership database and trading features the BC provides, so it would be harder to implement.

The main drawbacks of games using BC technology compared to centralized games are the technological complexity of BC systems and the lack of control over certain aspects. For example, unwanted or illegal content could be harder to censor.

The video game industry entails various additional constraints. For example, most video games need real-time data exchange between multiple players. Moreover, graphical assets generally need a lot of storage space and bandwidth. In order to apply the proposed architecture to this industry, we show in Fig. 2 a possible life cycle of a BC game asset. Once it is created, we need to store it in the suitable format and storage solution. In order to validate the properties of the asset with the BC, hashes and the main properties should be stored in a smart contract that manages the asset. Depending on the application, the validation step can also aim to ensure the data inside the asset can be used by the application, and does not contain unwanted or illegal content. This step can be achieved by a centralized entity that stakes its reputation on the asset validity, by a community vote or by any other consensus method. Finally, whenever a player wants to use the asset (because they or another player own it in-game), the application layer should check its properties.

To correctly represent a video game asset on the BC, we've seen that ERC token standards currently do not interface well with the other layers. For example, only the ERC-721 interface allows for a reference to metadata, and it only consists of one URI that could potentially become obsolete. The approach described in [26] has a similar issue. The representation of a video game asset needs to be generic and be quickly implemented with existing technologies.

We define two types of parameters: asset handling prop-

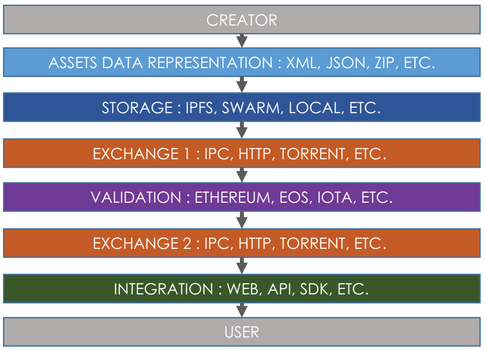
CREATOR
ASSETS DATA REPRESENTATION : XML, JSON, ZIP, ETC.
STORAGE : IPFS, SWARM, LOCAL, ETC.
EXCHANGE 1 : IPC, HTTP, TORRENT, ETC.
VALIDATION : ETHEREUM, EOS, IOTA, ETC.
EXCHANGE 2 : IPC, HTTP, TORRENT, ETC.
INTEGRATION : WEB, API, SDK, ETC.
USER

Fig. 2. Life cycle of a blockchain game asset

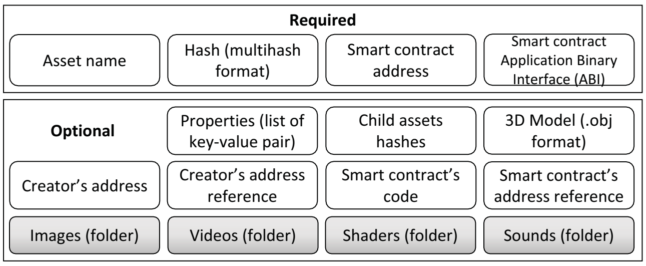

Fig. 3. Elements of the asset data representation. White boxes represent text files or strings, whereas gray boxes represent binary data.

erties, and asset specific properties. The first type describes required properties to identify the asset: for example its name, its hash, or its validation status. The second type contains anything else. In order to be able to represent generic assets, we focused on an archiving format similar to Java archives JAR. The content of this representation is described in Fig. 3. Most of the elements mentioned are self-explanatory. However, the following list adds precision to some of them:

- Hash (multihash [42] format) - can be used to quickly reference the asset. For example, if the hash is stored in a smart contract, one can retrieve the asset from the hash and then recompute it to ensure data integrity. The multihash format is self-describing, and we can implement it with any hashing function. This means that if an existing hashing function becomes obsolete because of hash collisions, we can change it with back-compatibility. However, hash collisions are less critical in our use case as we do not transfer value between users,
- Properties - related to the asset (e.g. its in-game effects),
- Smart contract's Application Binary Interface (ABI) - describes the prototypes of the functions of the contract,
- Smart contract's and creator's address reference (string) - a reference to the smart contract's address with a naming system such as the Ethereum Naming System (ENS),
- Child assets hashes - can be used for crafting different assets into one.

With this asset specification, game developers and BC engineers can use and agree on the same data representation. Also, it will be easier to develop tools to quickly import the assets and interact with the BC from game engines (Unity or Unreal Engine). To assess the feasibility of our proposal, we want to release a follow up research showing a prototype of a fully decentralized game implementing the architecture we propose and our proposed data representation for data transfers between players. This interoperability between game engines and the BC also allows for new game mechanisms, as shown in the next section.

V. User Generated Content

User Generated Content (UGC) gives players the opportunity to create assets and share them with anyone in the community. Our asset data representation and architecture can help game creators and users by providing a unified distributed game design framework, which supports interoperability. Indeed, anyone can create an asset following our representation. Then, the asset is verified on the BC by a smart contract and referenced by its hash, which ensures data integrity. We can ensure the asset follows community rules by Proof of Authority, as it is easy to implement, but it compromises decentralization. A more decentralized approach could use a community vote. In this case, to avoid Sybil attacks, only players above a certain level in-game (or having played a certain time) could vote. Another possibility is to automatically filter unwanted content using machine learning in a decentralized cloud computing framework (using, for example, the products proposed by iExec [43] or Golem [44]), but this approach brings a different set of constraints than reaching consensus within the community. For example, an artificial intelligence needs data for its training, and errors need to be handled. An advantage of using BC for UGC is that content creators can receive royalties automatically for the usage of their assets. An example of a business model would be to reserve part of the game's revenue for community content creators, based on how much content they provided and how much it is used by the community. This incentivizes content creation and involvement of players in the game.

VI. Conclusions and prospects

Better interoperability between BC and existing technologies is needed. This interoperability can be obtained by formalizing specifications for intercommunication between layers of the architecture of DBA. In this work, we presented an architecture applied to the video game industry. We saw that existing BC data representations could not easily be used throughout the whole architecture. That is why we described a new data representation for BC assets, that contain all the necessary information to be used in a BC environment. They also take into account the scalability issues of the BC, by allowing easier data sharing of BC assets. The next steps will be to generalize what we learned from the application of our proposal to the video game industry and refine our proposed design by considering applications in other industries, e.g. the Internet of Things, banking or supply chain. For example, we can use BC assets to create diploma certifications. Assets would contain the diploma and hashes would be referenced on the BC. The advantage of BC technology here would be to timestamp the certification, and allow for revocation.

The validation of our framework will be achieved by developing a proof-of-concept of a decentralized and real-time BC game using our architecture and asset data representation. Besides the decentralization, auditability and security benefits of BC, this allows for the game's community to be more involved in the governance and content creation of the game.

Acknowledgment

The PhD work of Léo Besançon is supported by B2Expand, 69100 Villeurbanne, France. We thank Éric Burgel, chairman of B2Expand, for his help and advice.

REFERENCES

[1] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “Medrec: Using blockchain for medical data access and permission management,” in *International Conference on Open and Big Data (OBD)*, 2016, pp. 25–30.

[2] S. A. Abeyratne, R. P. Monfared, “Blockchain ready manufacturing supply chain using distributed ledger,” *International Journal of Research in Engineering and Technology*, vol. 05, no. 09, pp. 1–10, Sep. 2016.

[3] Y. Guo and C. Liang, “Blockchain application and outlook in the banking industry,” *Financial Innovation*, vol. 2, no. 1, Dec. 2016.

[4] XAYA, “The ultimate blockchain gaming platform,” XAYA White paper, 2018. [Online]. Available: https://xaya.io/downloads/XAYA_White_Paper.pdf. [Accessed: 5-Dec-2018].

[5] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, “Blockchain challenges and opportunities: a survey,” *Int. J. Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[6] A. Deshpande, K. Stewart, L. Lepetit, and S. Gunashekar, “Distributed Ledger Technologies/Blockchain: Challenges, opportunities and the prospects for standards,” Prepared for the British Standards Institution (BSI), May 2017, May 2017.

[7] I. Grigg, “EOS - An Introduction,” 2017. [Online]. Available: https://eos.io/documents/EOS_An_Introduction.pdf. [Accessed: 12-Dec-2018].

[8] V. Buterin, “Chain Interoperability,” *R3 Research Paper*, 2016.

[9] J. Kwon and E. Buchman, “Cosmos: a network of distributed ledgers,” 2017. [Online]. Available: https://cosmos.network/resources/whitepaper. [Accessed: 12-Dec-2018].

[10] G. Wood, “Polkadot: Vision for a heterogeneous multi-chain framework,” White Paper, 2016. [Online]. Available: https://polkadot.network/PolkaDotPaper.pdf. [Accessed: 12-Dec-2018].

[11] T. Hardjono, A. Lipton, and A. Pentland, “Towards a Design Philosophy for Interoperable Blockchain Systems,” *arXiv*:1805.05934 [cs], May 2018.

[12] H. Jin, X. Dai, and J. Xiao, “Towards a Novel Architecture for Enabling Interoperability amongst Multiple Blockchains,” in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 1203–1211.

[13] W. J. Gordon and C. Catalini, “Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability,” *Computational and Structural Biotechnology Journal*, vol. 16, pp. 224–230, Jan. 2018.

[14] “Standards - IEEE Blockchain Initiative.” [Online]. Available: https://blockchain.ieee.org/standards. [Accessed: 11-Dec-2018].

[15] “IEEE-SA - The IEEE Standards Association - Home.” [Online]. Available: https://standards.ieee.org. [Accessed: 19-Dec-2018].

[16] Standard for the Framework of Blockchain Use in Internet of Things (IoT). P2418.1. 2017. [Online]. Available: https://standards.ieee.org/project/2418_1.html. [Accessed: 12-Dec-2018].

[17] “Enterprise Ethereum Alliance - Home.” [Online]. Available: https://entethalliance.org. [Accessed: 19-Dec-2018].

[18] Enterprise Ethereum Alliance, Enterprise Ethereum Client Specification V2. 2018. [Online]. Available: https://entethalliance.org/wp-content/uploads/2018/11/EEA_Enterprise_Ethereum_Client_Specification_V2.pdf. [Accessed: 12-Dec-2018].

[19] “Blockchain reference architecture - IBM Cloud Garage Method.” [Online]. Available: https://www.ibm.com/cloud/garage/architectures/blockchainArchitecture/reference-architecture/. [Accessed: 11-Dec-2018].

[20] Hoard — Facilitating True Ownership of Virtual Gaming Assets on the Ethereum Blockchain. Buy, Sell and Rent Downloadable Content on a Marketplace. Powered by Blockchain Technology. https://www.hoard.exchange/index.html

[21] F. Vogelsteller and V. Buterin. “ERC-20 Token Standard,” 2015. [Online]. Available at: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md [Accessed: 5-Dec-2018].

[22] W. Entriken, S. Dieter, E. Jacob and N. Sachs. “ERC-721 Non-Fungible Token Standard,” 2018. [Online] Available at: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-721.md [Accessed: 5-Dec-2018].

[23] CryptoKitties. ”CryptoKitties — Collect and breed digital cats!,” 2018. [Online] Available at: https://www.cryptokitties.co/ [Accessed: 6-Dec-2018].

[24] W. Radomski, A. Cooke, P. Castonguay, J. Therien and E. Binet. “Multi Token Standard,” 2018. [Online] Available at: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1155.md [Accessed: 5-Dec-2018].

[25] A. Palau, “Storing on Ethereum. Analyzing the costs,” Coinmonks, 2018. [Online]. Available: https://medium.com/coinmonks/storing-on-ethereum-analyzing-the-costs-922d41d6b316. [Accessed: 6-Dec-2018].

[26] E. Ordano, A. Meilich, Y. Jardi, and M. Araoz, “Decentraland, A blockchain-based virtual world,” 2017. [Online]. Available: https://decentraland.org/whitepaper.pdf. [Accessed: 6-Dec-2018].

[27] A. Regenscheid and D. Yaga, “Blockchain and Distributed Ledger Technologies: Opportunities, Challenges and Future Work,” 2017. [Online]. Available: https://csrc.nist.gov/CSRC/media/Presentations/NIST-Block-Chain-Research-Project/images-media/ar-dy-blockchain-combined.pdf. [Accessed: 6-Dec-2018].

[28] A. P. Kryukov and A. P. Demichev, “Decentralized Data Storages: Technologies of Construction,” Program Comput Soft, vol. 44, no. 5, pp. 303–315, Sep. 2018.

[29] J. Benet, “IPFS - Content Addressed, Versioned, P2P File System,” *arXiv*:1407.3561 [cs], Jul. 2014.

[30] Protocol Labs, “Filecoin: A Decentralized Storage Network,” 2018. [Online]. Available: https://filecoin.io/filecoin.pdf. [Accessed: 12-Dec-2018].

[31] V. Trón, A. Fischer, D. A. Nagy, Z. Felföldi, and N. Johnson, “Swap, swear and swindle: incentive system for swarm,” Technical Report, Ethersphere Orange Papers 1, 2016.

[32] G. Agrawal. ”OrbitDB: A peer-to-peer database for the decentralized web,” 2018. [Online] Available at: https://medium.com/coinmonks/orbitdb-a-peer-to-peer-database-for-the-decentralized-web-30bac1d056fe [Accessed: 6 Dec. 2018].

[33] Nadal, M. ”amark/gun: A realtime, decentralized, offline-first, graph database engine,” GitHub, 2018. [Online]. Available at: https://github.com/amark/gun [Accessed: 6 Dec. 2018].

[34] V. Buterin et al., “A next-generation smart contract and decentralized application platform,” 2014. [Online] Available at: https://github.com/ethereum/wiki/wiki/White-Paper [Accessed: 6 Dec. 2018].

[35] E. Androulaki et al., “Hyperledger fabric: a distributed operating system for permissioned blockchains,” in Proceedings of *the Thirteenth EuroSys Conference on - EuroSys ’18, Porto, Portugal*, 2018, pp. 1–15.

[36] J. Poon and T. Dryja, “The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments,” 2016. [Online]. Available: https://lightning.network/lightning-network-paper.pdf. [Accessed: 12-Dec-2018]

[37] A. Back et al., “Enabling Blockchain Innovations with Pegged Sidechains,” 2014. [Online]. Available: http://kevinriggen.com/files/sidechains.pdf. [Accessed: 12-Dec-2018]

[38] B. Guidi, M. Conti, A. Passarella, and L. Ricci, “Managing social contents in Decentralized Online Social Networks: A survey,” *Online Social Networks and Media*, vol. 7, pp. 12–29, Sep. 2018.

[39] I. Grishchenko, M. Maffei, and C. Schneidewind, “A Semantic Framework for the Security Analysis of Ethereum Smart Contracts,” in Principles of Security and Trust, 2018, pp. 243–269.

[40] V. Buterin, “Vitalik Buterin on about.me,” about.me. [Online]. Available: https://about.me/vitalik_buterin. [Accessed: 11-Mar-2019].

[41] Lunar Mines - Own your space. [Online]. Available: https://lunarmines.io. [Accessed: 11-Mar-2019].

[42] Multiformats/multihash. Self describing hashes - for future proofing. [Online]. Available: https://github.com/multiformats/multihash. [Accessed: 11-Mar-2019].

[43] G. Fedak, B. Wassim, and A. Eduardo, “iExec: Blockchain-Based Decentralized Cloud Computing,” Version 3.0, 2018. [Online]. Available: https://iex.ec/whitepaper/iExec-WPv3.0-English.pdf. [Accessed: 8-Dec-2018].

[44] Golem. ”The Golem Project. Whitepaper,” 2016. [Online]. Available: https://golem.network/crowdfunding/Golemwhitepaper.pdf. [Accessed: 8-Dec-2018].