Blockchain Support in IoT Platforms

Pustišek, Matevž; Štefanič Južnič, Leon; and Kos, Andrej

Abstract: ***Blockchain (BC) technologies have a potential to blend with the existing Internet of things (IoT) platforms. BC enabled IoT platforms can be offered as a service (BIoTaaS) to provide scalable and trusted new approaches in e.g. IoT device authentication and management, trading with IoT data or in providing reliable and trusted interfaces between Web and smart contracts. At the same time this can lead to a gradual decentralization of highly centralized traditional cloud platforms – a needed change in IoT that can be anticipated from the fog computation and communication architectures, too. Currently the two viable BC candidates for BIoTaaS are the Ethereum (ETH) and the Hyperledger Fabric (HLF). Very diverse applications of BIoTaaS are possible, so it is unlikely that one platform approach or architecture will be meeting all these needs. Two differentiators have the key impact on selection of BC technology for particular BIoTaaS: existence of need for instant and independent on-chain payments and where the dominant focus is set – on the IoT devices or on the business-to-business (B2B) applications. If the devices are central and payments are required, then ETH BC is the favorite. In case of B2B HLF might be a preferable option due to security features beyond trust, derived from the permissioned network model. Beside the existence of BC, other requirements have to be met for efficient BIoTaaS. We defined a set of such common requirements, which include Web/HTTP/REST and other acknowledged application programming interfaces (API) for entire IoT and BC service access, on-chain smart contracts, low transaction confirmation delays for instant payments and near real-time operation, and smart oracles for interfacing the off-chain "real-world" objects and systems.***

Index Terms: ***API, blockchain, Ethereum, Hyperledger Fabric, Internet of things, platform***

1. INTRODUCTION

RECENT advancements in the Internet of things have brought it to the level of productivity. The Internet of things (IoT) solutions are being applied in industry, smart grids, health, mobility, wellbeing and other application domains. The IoT ecosystems are characterized by big number of heterogeneous and often constrained IoT devices, emerging user requirements and complex use-cases, and omnipresent demanding business and security requirements [1]. Traditional IoT architectures [2] are highly centralized, with cloud platforms providing services for collection, storage, analysis and use of vast quantities of data, provided by the IoT devices. However, with 5G network- and fog computation and communication architectures the traditional centralized IoT model has started evolving towards a more decentralized one. Decentralization is the key principle of the blockchain (BC) protocols and networks, too. BC enables trusted exchange of transactions in a system without trusted centralized authorities. Providing a native cryptocurrencies, autonomous machine-to-machine transactions including micropayments or distributed applications, BCs seem to be a valuable addition to IoT, too.

The key objective of this paper is to investigate the role of the BC in the IoT cloud platforms. In Section 2 we present cloud-based IoT platforms, which are along with communication gateways and IoT devices, the key building part of the IoT systems. Cloud APIs are presented as means for integration and use of cloud IoT services. Possible impact and decentralization of fog computation and communication architectures for IoT is discussed. In Section 3 a brief presentation of the blockchain and the distributed application concepts is given. Two key BC technologies for the IoT are exposed - the Ethereum and the Hyperledger Fabric. In Section 4 we summarize some of the initial blendings of established IoT platforms and blockchain technologies. Finally, based on our findings we present a set of general requirements for an IoT platform with blockchain support for IoT devices and for other cloud-based systems.

2. CLOUD BASED IOT PLATFORMS

IoT appears to be a playground for the most advanced new business and technological developments. 5G systems, machine-to-machine communications, fog computing and alike are constantly reshaping the established

Manuscript received June 10, 2016; revised June 27, 2017; accepted June 28, 2017. The work was supported in part by the Ministry of Education, Science and Sport of Slovenia. Authors are with the Faculty of Electrical Engineering, University of Ljubljana, Ljubljana SI-1000, Slovenia (matevz.pustisek@fe.uni-lj.si; leon.juznic@ltfe.org; andrej.kos@fe.uni-lj.si).

architectures of the IoT systems. But in spite of these new impacts, the IoT is reaching the productivity level and is being successfully applied in various application domains. All the key components to create IoT solutions are available as proven commercial (industry grade) products and services.

In a very simplified form, an IoT system—Figure 1—is comprised of devices, communication gateways and networks, and cloud-based backend systems. Devices are numerous and heterogeneous and have at least basic computation and communication capabilities. They incorporate sensors and/or actuators to face the real world environment.

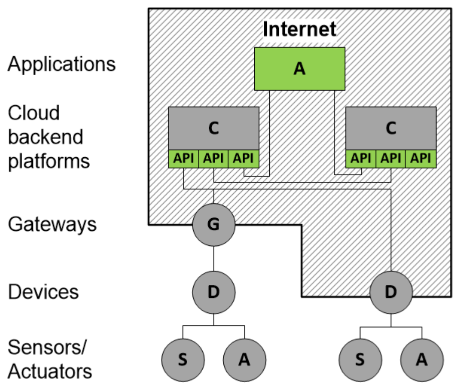

Figure 1: IoT architecture

Gateways add communication capabilities to connect devices to the Internet, where such a functionality is not already incorporated in the device, e.g. due to the lack of a complete network stack required for Internet connectivity, energy consumption constraints or limited computational capabilities. A gateway acts as a proxy, receiving data from devices and packaging it for transmission over Internet [3].

Backend systems/platforms collect, process, analyze and act on data generated by connected devices, enable long term storage and (big data) analysis, and facilitate easy development of the IoT applications that interact with the devices. The IoT platforms are frequently provided as a service (PaaS). Note: with term *IoT platform* in this paper we refer to backend cloud platforms and not to the IoT device platforms as for example single-board-computers (e.g. *RPi*) or microcontroller platforms (e.g. *Arduino*).

The leading IoT platforms are actually sets of products and services, which jointly provide the required functionalities. PaaS providers e.g. adapt their storage, big data or machine learning platforms, which are being used for non-IoT applications, as well. Apart from these more commonly or generally required functionalities, the IoT cloud platforms provide IoT specific services. These services shield applications from specific features of the IoT devices. They assure scalability (numerous IoT devices), security (access management, computation constraints) and management (registration, deployment, operation). IoT specific services also enable seamless interoperability among IoT specific parts and various other platforms that are combined in an IoT cloud backend.

There are numerous examples of IoT cloud platforms being in use. The largest share [4] of maker project is backed up with *MS Azure for IoT* [5] and *Amazon AWS for IoT* [6], closely followed by *Google Cloud IoT* [7]. These all are solutions from leading cloud service providers. They blend their existing cloud platforms with IoT specifics to provide a mash-up for IoT cloud services.

Google Cloud IoT is a rich set of various non-specific and IoT specific cloud platforms that are combined for IoT. Among the general services there are storage and databases, big data, machine learning and alike. The IoT specifics are arranged with *Cloud IoT Core* [8] which is a fully managed service that allows you to easily and securely connect, manage, and ingest data from numerous globally dispersed and heterogeneous devices.

Similar approach is taken in *Amazon AWS for IoT* [6]. IoT specific services are provided in the *AWS IoT Platform*, primarily focusing on secure and efficient communication between devices and other AWS. *AWS Greengrass* being more an IoT device- then a cloud-platform extends AWS to devices so they can act locally on the data they generate, while still using the cloud for management, analytics, and durable storage.

The *Microsoft Azure IoT Suite* [5] is an enterprise-grade cloud solution that enables you to get started quickly through a set of extensible preconfigured solutions. The services offer a broad range of capabilities. These enterprise grade services enable you to: collect data from devices, analyze data streams in-motion, store and query large data sets, visualize both real-time and historical data, integrate with back-office systems and manage your devices [9].

IBM Bluemix [10] is a cloud platform as a service (PaaS) developed by IBM. *Bluemix* is based on Cloud Foundry open technology and runs on SoftLayer infrastructure. It supports access to over 120 IBM cloud services (machine learning, storage, application services, blockchain, etc.). It also includes IBM's IoT Platform, which provides services for connecting and managing IoT devices, and analyzing the data they produce. It supports connecting to the cloud using open, lightweight MQTT messaging protocol or HTTP [11].

Some IoT cloud platforms operate at a smaller scale. They are not a part of an integral scope of cloud services (IoT and non-IoT), but have been developed specifically for the IoT. *Thingspeak* [12] for example supports collection, storage, analysis and visualization of IoT data. For the collection, dominant IoT device platforms (*RPi*, *Arduino*, *and BeagleBone*) are directly supported. Analysis and visualization are made with *Matlab*. Therefore *Thingspeak* gained a lot of interest in academic and research communities. The *Open Source Elastic Stack* [13] is specialized in real time data analysis and visualization. Their key products - *Elasticsearch* and *Kibana* - enable you to reliably and securely collect data from any source, in any format, and search, analyze, and visualize it in real time. These two systems are available as integral modules also in the *AWS for IoT*.

Opensensors [14] is oriented towards acquisition and interchange of open IoT data and provides interfaces to efficiently exchange data among various IoT backend platforms.

2.1. Server Application Programming Interfaces

IoT cloud solutions are distinguished by various APIs for interoperability of their building components, for communication with the devices and for the applications based on these cloud solutions. IoT cloud platform APIs reflect the specifics and constraints of IoT: numerous and heterogeneous devices, low power devices, limited communication and computation capabilities. The key is, of course:

- HTTP/Web and real time APIs: to reliably and securely interact with cloud applications and other devices.

Apart from support for data and message exchange, IoT APIs may include features for:

- Virtual representations of devices: device shadows – implement an always available REST API for offline operation. Even if the actual device is temporarily offline, applications retain possibility to communicate with the device.
- Device management: registration, provisioning, deployment, updates and operation of devices at scale.
- Security and access management: for authentication and authorization of devices and platform users in form of API keys, JSON Web Tokens (JWS).
- Rule engines: gather, process, analyze and act on data from the connected devices and route the messages to other PaaS or their components.

In terms of implementation cloud APIs may rely on WebSocets, HTTPS REST, general-purpose RPC (gRPC), server-sent events (SSE) and others. The variety of implementation options reflects different needs of application developers, as well as different characteristics of messages and data streams passed over APIs.

IoT cloud platform providers frequently publish client libraries for various IoT device platforms and programming languages. These libraries facilitate the use of their cloud APIs and make application development easier.

2.2. Fog Architecture

Lately another architecture related to the IoT has been widely discussed. This is the fog computation and communication [15]. It reflects changes which are anticipated in mobile edge networks as envisaged in future 5G and partially outlined in current Evolved packet system (EPS) with LTE-A. The fog doesn't exclude cloud services and systems. It merely redistributes the location of computation, storage and control to decentralized elements in the architecture. In a unified end-to-end fog-cloud platform, cloud services continue to have an indispensable role. But the IoT system architecture is no longer limited to a device (full of constraints), transparent (dumb) communication networks and the smart cloud. Integrated fog nodes combine computation and communication.

The reasons for decentralization towards the fog are multiple: security, having applications closer to the end user, agility in application development (changing client application without a need to have the change implemented in cloud backend first) and efficiency. But the primary benefit of the fog computing is its ability to reduce latency and delay. There are additional features required in the fog-cloud systems: new service discovery, request and delivery mechanisms; different data management, taking into account local processing and storage; and service orchestration. In fog not only vertical interactions between the users/devices, edge nodes and cloud are foreseen. There are interactions among instances at same level, too [16].

Decentralized and distributed architecture of the fog computing and networking has, therefore, several similarities with decentralized blockchains, discussed in Section 3.

3. BLOCKCHAIN PROTOCOLS AND NETWORKS

Blockchains and distributed ledgers are listed among top strategic technology trends in 2017 [17]. They provide a decentralized framework for trusted transactions. The blockchain technologies are well known fundament of cryptocurrencies, but offer many other possible applications areas, too. In terms of IoT, two not mutually exclusive roles of a blockchain can be pointed out:

- As a distributed, scalable and trusted database, where the act of inserting/reading a parameter value is called a transaction, which is verified by a distributed community. The blockchain technology does not (necessarily) provide privacy of this data.

- Decentralized application environment for distributed deployment of applications.

Various specifications and implementations of blockchain technologies are available, but in our opinion at the moment two have relevant prospects for IoT. These are the Ethereum (ETH) [18] and the Hyperledger Fabric (HLF) [19]. Although the Bitcoin [20] is probably the most prominent BC technology, which gained reputation mostly due to the popular cryptocurrency Bitcoin [21], its potential role in IoT is extremely limited and is not a viable candidate for an IoT BC solution. Bitcoin protocol is namely lacking the distributed on-chain smart applications. Its role is thus limited more or less just to supporting a cryptocurrency.

In terms of application development for BC two approaches can be combined—off-chain and on-chain—as seen in Figure 2:

- Off-chain applications are Web, mobile and other applications, which use BC via client APIs that are exposed by BC clients. The BC client is responsible for the entire communication with BC and the application part for business logic (relaying on BC operation).

- On-chain business logic refers to smart contracts (i.e. chaincode in HLF), which are programming code written in Solidity – ETH - or in Go (or Java) - HLF, compiled and deployed in the BC network. Executions of smart contract are validated in the BC. BC thus provides a decentralized and trusted virtual machine for smart contract executions.

In Table 1 we compare the three blockchain protocols from the IoT perspective. Although different in their approaches (programming languages, etc), ETH and HLF both enable on-chain applications. The two protocols differ importantly in their consensus algorithms, too. For ETH public network the proof of work (PoW) is used currently – as it is in Bitcoin. In HLF the Practical Byzantine fault tolerance requires permissioned validating nodes (See Section 3.1. for details).

3.1. Ethereum

The Ethereum protocol [18] and corresponding networks are the basis for trusted, decentralized applications – *Dapps*. Apart from enabling a relevant cryptocurrency – ether [22] – Ethereum protocol is distinguished by a highly generalized programming language. With it one can code a smart contract which is deployed to the ETH network. It forms a contract account which is controlled by its contract code. The code is executed every time such an account receives a message/transaction from another account. This is the fundament for various IoT related blockchain applications.

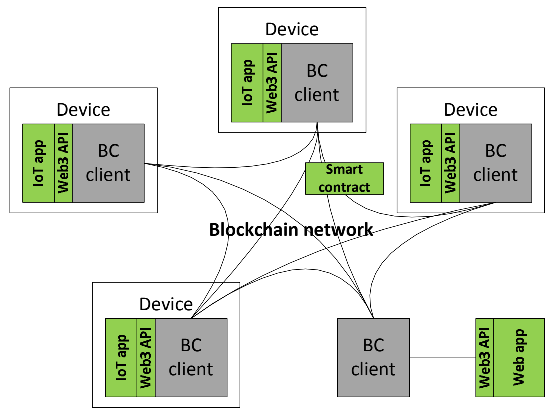

Figure 2: Ethereum BC architecture

3.1.1. Ethereum Client API

The entire functionality of ETH blockchain is available through the web3.js (and/or RPC/JSON) API [23] which geth – the ETH client - is exposing. Geth [24] is responsible for running ETH protocols and thus the entire communication with the blockchain. There are other ETH client implementations available, but geth usually serves as the reference, because it is being developed by Ethereum Foundation developers. Through this client API the entire functionality of ETH node can be exploited, including management of blocks and transactions, management of peers and network, monitoring of chain status, managing ETH accounts or mining ETH blocks. Application development in case of ETH and geth thus relies on using this client API, as depicted in Figure 2. Applications and geth can run on the same device or on two separate devices. In the latter case, HTTP protocol can be used in application to reach the distant geth node.

3.1.2. Smart Contract API

Smart contracts are on-chain business logic that is executed within the blockchain network. The execution can be verified by any network participant and thus trusted in the same way as any other transaction in BC network is.

A smart contract exposes functions, which are used by blockchain account. These functions represent a kind of an on-chain API for other BC accounts, and are accessible via blockchain.

Table 1: Comparison of Blockchain networks

	Bitcoin	Ethereum	Hyperledger Fabric
Native cryptocurrency	Yes	Yes	No
Distributed applications	No (very limited)	Yes – smart contracts	Yes - chaincode
Smart contracts	-	Solidity	Go, Java - (executed in containers)
Consensus algorithm	Proof of Work (PoW)	Proof of Work (PoW) Proof of stake (PoS) foreseen	PBFT - Practical Byzantine fault tolerance
Anonymous accounts	Yes	Yes	No (permissioned network)
Network	Public	Public or permissioned	Permissioned
Suitable for IoT	No	Yes	Yes
State channels	Yes (Lightning)	Yes (Raiden)	Not required

3.1.3. Decentralized Data Feeds with Smart Oracles

Smart contracts in ETH environment operate in a rather isolated space. The Solidity [25] smart contract programming language has e.g. no means to request data from URLs and thus to interface with "real" Internet world – e.g. Web sites or IoT devices - because these external information cannot be trustworthily verified by the contract. This shortcoming can be outdone by oracles [26]. These serve as intermediaries, providing data feeds along with an authenticity proof to the blockchain form/to external software (e.g. Web sites) or hardware entities.

3.1.4. State Channels

Another challenge in BC networks with PoW consensus is the transaction validation period. Due to the nature of the blockchain, this can take from several tens of seconds (in ETH about 20 s) up to several minutes (in Bitcoin 10 min) or more. Besides, distributed application developers do not have an influence on these times. In fact, the delays may become even longer due to higher transaction rates in the network or if several consecutive block verifications are required for security reasons. This limits, at least to some extent, the feasibility of scalable instant payments and is not suitable for near real-time IoT applications (e.g. door lock controlled by BC). A solution to this is being sought in state channels. This architecture combines off- and on-chain transactions to contribute to additional scalability, privacy and reduction of confirmation delays, compared to the current BC architecture. In ETH this approach is manifested in the Raiden [27] and in Bitcoin in the Lightning network [28].

3.2. Hyperledger Fabric

The Hyperledger Project [11] is a collaborative effort to create an enterprise-grade, open-source distributed ledger framework and code base. Established as a project of the Linux Foundation in early 2016, the Hyperledger Project currently has more than 130 members, including leaders in finance, banking, in the internet of things, supply chain, manufacturing and technology.

The Hyperledger Fabric [19], one of multiple projects currently in incubation under the Hyperledger Project, is a permissioned blockchain platform aimed at business use. It is open-source and based on standards, runs arbitrary smart contracts (called chaincode), supports strong security, identity features, basic REST APIs, CLIs and uses a modular architecture with pluggable consensus protocols (currently an implementation of Byzantine fault-tolerant consensus using the PBFT protocol [29] is supported).

The distributed ledger protocol of the fabric is run by peers. The fabric distinguishes between two kinds of peers: (i) *validating peer* is a node on the network responsible for running consensus, validating transactions, and maintaining the ledger and (ii) a *non-validating peer* which is a node that functions as a proxy to connect to validating peers [30].

4. IoT Platforms with BC Support

The IoT is facing several challenges that need to be addressed to continue with its successful practical deployments: centralized ecosystem, the cost of the connectivity, disrupted business models, security and trust and lack of functional value. A decentralized approach to IoT networking would solve many of the issues above. Blockchain technology is the missing link to cope with some of the future challenges in the IoT [31]. BC can:

- reduce costs - track billions of connected devices, enabling the processing of transactions and coordination between devices, managing updates [32],
- build trust - cryptographic algorithms used by blockchains would make consumer data more private, man-in-the-middle attacks

Table 2: Comparison of architectures: Cloud based Internet of Things PaaS vs. Blockchain

	Cloud based Internet of Things	Blockchain
Topology	• Centralized • (decentralization only being introduced in fog)	• Decentralized, fully distributed (P2P like)
APIs	• HTTP/Web and real time server APIs	• API at every particular BC client • Smart contract functions in form of backend API
Device libraries	• To use server APIs	• To use client API
Security focus	• Authentication and authorization of devices • Security and privacy of cloud services • Communication security • Availability	• Trust
Latency	• Low to moderate, near real time operation is possible • (fog architecture additionally reduces latency)	• High, due to the nature of transaction validation
(Micro)payments	• Not part of common IoT platforms	• Essential part of technology
Application logic in the platform	• In platform modules (big data, queries, etc). • Web applications accessing PaaS through APIs	• Smart contracts

cannot be staged, ledger cannot be manipulated,

- accelerate transactions - decentralized approach would eliminate single points of failure,
- keep an immutable record of the history of smart devices - no need for a centralized authority, and
- provide machine-to-machine transactions and micropayments.

Beside this, BC can easily facilitate:

- decentralized data feeds (Schelling coin [33]), where a vast amount of numerous concurrent (low fidelity) measurements from IoT devices is summarized into e.g. the most possible value of the temperature.

In spite of all its benefits, the blockchain model is not without its flaws and shortcomings. This is not surprising, because blockchain technologies are being relatively new and not as mature as e.g. IoT technologies.

4.1. Existing Examples

Big IT companies are already exploring the opportunities of blockchain in IoT. They are usually integrating blockchain as a service (BaaS) in their existing IoT platforms. So BaaS is provided along with the existing IoT PaaS, which were discussed in Section 1.

IBM is the leader in open-source blockchain solutions built for the enterprise. Their blockchain ecosystem brings together a range of people and organizations interested in building and leveraging blockchain solutions. IBM Watson IoT™ platform [34] enables IoT devices to send data to private blockchain ledgers for inclusion in shared transactions with tamper-resistant records. Its BaaS service is based on Hyperledger Fabric [19], which is one of the five frameworks hosted with Hyperledger. IBM contributed more than a half of the code used in HLF. This demonstrates IBM's strong commitment to provide open governance for the development of blockchain.

HLF in IBM Watson IoT is predominantly suitable for private blockchains in enterprise settings, because it is using a different consensus algorithm than e.g. ETH. It is distinguished by a well-documented HTTPS REST API [35] for all blockchain related functions. Web developers can thus benefit from BC features, but continue to utilize API technologies they are already familiar with. The Watson IoT API enables management of blocks and transactions as well as peers and networks, monitoring of chain status, and registrations and management of BC users.

Microsoft entered a partnership to create Ethereum blockchain as a service (EBaaS) on Microsoft Azure [36]. The service will allow users to efficiently create private, public and consortium based Blockchain environments using industry leading frameworks. Surrounding capabilities like Cortana Analytics (machine learning), Power BI, Azure Active Directory, can be integrated into apps launching a new generation of decentralized cross platform applications.

SAP Leonardo [37] is a digital innovation system, which integrates IoT, machine learning, analytics, big data and blockchain, and runs them seamlessly in the cloud. The blockchain element is based on the Hyperledger open source blockchain platform, using its standards and protocols. SAP joined Hyperledger as a premier member early in 2017.

4.2. Selection Criteria and Common Requirements for Iot Platform with BC Support

Numerous and very diverse applications which combine IoT and BC are possible, so it is unlikely that one blockchain IoT platform (BIoTaaS) or platform architecture will be meeting all these needs. Two differentiators have key impact on the selection of BC technology for particular BIoTaaS: (i) whether there is a need for instant and independent on-chain payments and (ii) where the dominant focus is set – on devices or on the business-to-business (B2B) applications. In developing applications based on BIoTaaS, BC can be integrated in the IoT devices as well as in the backend (cloud) part.

ETH and HLF are the two viable BC candidates for BIoTaaS. If the devices are central and payments are required, then ETH BC is the favorite. In case of ETH, nodes can benefit from reliable scalable public network, recognized cryptocurrency, and existing examples of client and application deployments in computers, mobile devices and embedded systems. In case of B2B, HLF might be a preferable option due to security features beyond trust, derived from the permissioned network model. Despite several different features, ETH and HLF share many common requirements. These requirements have to be met also in any alternative BC technologies considered for IoT:

- Web/HTTP/REST and other acknowledged APIs to access the entire set of IoT and BC services.
- On-chain smart contracts.
- Low transaction confirmation delays for instant payments and near real-time operation. This can be achieved either by the consensus algorithm or through the availability of state channels.
- Smart oracles for interfacing "real-world", which can be an integrated function of the BIoTaaS.

Beside the technical and functional features, other strategic decisions may determine selection of technologies for BIoTaaS. With BC being relatively new technology many parts remain in early development stages. So maturity of available solutions, size and support of the involved development community and successful use cases should be considered in selection, too.

5. CONCLUSIONS

Rapid progress in BC and lower maturity compared to the existing cloud services require a thoughtful positioning of BC in BIoTaaS. Many of current BC developments focus on creating yet another alternative coin along with corresponding smart contracts, to support vaguely defined potential use cases. Their motivation is in prospects of a successful initial coin offering (ICO). Not that many initiatives bring BC to the real world, including IoT.

We anticipate an important role of IoT platforms with highly integrated BC support in e.g. IoT device authentication and management, trading with IoT data or in providing a reliable and trusted interface between Web and smart contracts.

Our future research is oriented towards the architectures and position of BIoTaaS in smart city ecosystem.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of the research program "Algorithms and Optimization Procedures in Telecommunications", financed by the Slovenian Research Agency.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

[3] "Overview of Internet of Things," Google Cloud Platform, 19-Apr-2017. [Online]. Available: https://cloud.google.com/solutions/iot-overview. [Accessed: 05-Jun-2017].

[4] S. Palmer, "10 Best Internet of Things (IoT) Cloud Platforms – DevTeamSpace Blog," 01-Mar-2017. [Online]. Available: https://www.devteam.space/blog/10-best-internet-of-things-iot-cloud-platforms/. [Accessed: 05-Jun-2017].

[5] "Azure IoT Suite | Microsoft Azure." [Online]. Available: https://azure.microsoft.com/en-us/suites/iot-suite/. [Accessed: 19-Jun-2017].

[6] "AWS IoT," Amazon Web Services. [Online]. Available: //aws.amazon.com/iot/. [Accessed: 20-Jun-2017].

[7] "Google Cloud IoT - Fully managed IoT services from Google," Google Cloud Platform. [Online]. Available: https://cloud.google.com/solutions/iot/. [Accessed: 20-Jun-2017].

[8] "Cloud IoT Core." [Online]. Available: https://cloud.google.com/iot-core/. [Accessed: 20-Jun-2017].

[9] D. Betts, "Microsoft Azure IoT Suite overview." [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-suite/iot-suite-overview. [Accessed: 19-Jun-2017].

[10] "Cloud Infrastructure, Storage, Security &, More" IBM Bluemix. [Online]. Available: https://www.ibm.com/cloud-computing/bluemix/. [Accessed: 19-Jun-2017].

[11] "Hyperledger – Blockchain Technologies for Business." [Online]. Available: https://www.hyperledger.org/. [Accessed: 19-Jun-2017].
[12] "Learn More about ThingSpeak." [Online]. Available: https://thingspeak.com/pages/learn_more. [Accessed: 20-Jun-2017].
[13] "The Open Source Elastic Stack." [Online]. Available: https://www.elastic.co/products. [Accessed: 20-Jun-2017].
[14] "OpenSensors." [Online]. Available: https://www.opensensors.io/. [Accessed: 20-Jun-2017].
[15] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," in Big Data and Internet of Things: A Roadmap for Smart Environments, N. Bessis and C. Dobre, Eds. Springer International Publishing, 2014, pp. 169–186.
[16] M. Chiang, S. Ha, C. L. I, F. Risso, and T. Zhang, "Clarifying Fog Computing and Networking: 10 Questions and Answers," IEEE Communications Magazine, vol. 55, no. 4, pp. 18–20, Apr. 2017.
[17] Kasey Panetta, "Gartner's Top 10 Strategic Technology Trends for 2017 - Smarter with Gartner," 18-Oct-2016. [Online]. Available: http://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/. [Accessed: 05-May-2017].
[18] V. Trón, "Ethereum Specification," 23-Jul-2015. [Online]. Available: https://github.com/ethereum/go-ethereum/wiki/Ethereum-Specification. [Accessed: 05-May-2017].
[19] "IBM Blockchain - The Hyperledger Project," 16-Jun-2017. [Online]. Available: https://www.ibm.com/blockchain/hyperledger.html. [Accessed: 19-Jun-2017].
[20] "Protocol documentation - Bitcoin Wiki." [Online]. Available: https://en.bitcoin.it/wiki/Protocol_documentation. [Accessed: 02-Aug-2016].
[21] "Bitcoin - Open source P2P money." [Online]. Available: https://bitcoin.org/en/. [Accessed: 02-Aug-2016].
[22] "ETH/USDT Market - Poloniex Bitcoin/Cryptocurrency Exchange." [Online]. Available: https://poloniex.com/exchange#usdt_eth. [Accessed: 05-May-2017].
[23] "JavaScript API," ethereum/wiki Wiki · GitHub. [Online]. Available: https://github.com/ethereum/wiki/wiki/JavaScript-API. [Accessed: 05-May-2017].
[24] V. Trón, "Geth," ethereum/go-ethereum Wiki · GitHub. [Online]. Available: https://github.com/ethereum/go-ethereum/wiki/geth. [Accessed: 08-May-2017].
[25] "Solidity — Solidity 0.2.0 documentation." [Online]. Available: http://solidity.readthedocs.io/en/latest/index.html. [Accessed: 08-May-2017].
[26] "Oraclize Documentation," Overview. [Online]. Available: http://docs.oraclize.it/#overview. [Accessed: 05-May-2017].
[27] "Raiden Network," High speed asset transfers for Ethereum, 20-Dec-2016. [Online]. Available: http://raiden.network/. [Accessed: 05-May-2017].
[28] "Lightning Network," Scalable, Instant Bitcoin/Blockchain Transactions. [Online]. Available: http://lightning.network/. [Accessed: 05-May-2017].
[29] "Byzantine fault tolerance," Wikipedia. 10-Jun-2017.
[30] C. Cachin, "Architecture of the Hyperledger Blockchain Fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, Chicago, Illinois, USA, 2016.
[31] "IoT and Blockchain Convergence: Benefits and Challenges - IEEE Internet of Things." [Online]. Available: http://iot.ieee.org/newsletter/january-2017/iot-and-blockchain-convergence-benefits-and-challenges.html. [Accessed: 19-Jun-2017].
[32] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292–2303, 2016.
[33] V. Buterin, "SchellingCoin: A Minimal-Trust Universal Data Feed," Ethereum Blog, 28-Mar-2014. .
[34] "IBM Watson Internet of Things (IoT)," 09-Jun-2017. [Online]. Available: https://www.ibm.com/internet-of-things/. [Accessed: 19-Jun-2017].
[35] "Core API - Hyperledger Fabric." [Online]. Available: http://hyperledger-fabric.readthedocs.io/en/stable/API/CoreAPI/#rest-api. [Accessed: 20-Jun-2017].
[36] "Blockchain as a Service (BaaS)," Microsoft Azure. [Online]. Available: https://azure.microsoft.com/en-us/solutions/blockchain/. [Accessed: 19-Jun-2017].
[37] "SAP Leonardo | Digital Innovation System," SAP. [Online]. Available: https://www.sap.com/products/leonardo.html. [Accessed: 19-Jun-2017].