

# Blockchain and Cryptocurrency: A comparative framework of the main Architectural Drivers

Martin Garriga  
 CONICET, National Council for  
 Scientific and Technical Research  
 Buenos Aires 1400  
 Neuquen 8300, Argentina  
 martin.garriga@fi.uncoma.edu.ar

Maxmiliano Arias  
 Fidtech  
 Independencia 596  
 Neuquen 8300, Argentina  
 maximiliano.arias@fidtech.net

Alan De Renzis  
 Fidtech  
 Independencia 596  
 Neuquen 8300, Argentina  
 alan.derenzis@fidtech.net

## ABSTRACT

Blockchain is a decentralized transaction and data management solution, the technological weapon-of-choice behind the success of Bitcoin and other cryptocurrencies. As the number and variety of existing blockchain implementations continues to increase, adopters should focus on selecting the best one to support their decentralized applications (dApps), rather than developing new ones from scratch. In this paper we present a framework to aid software architects, developers, tool selectors and decision makers to adopt the right blockchain technology for their problem at hand. The framework exposes the correlation between technological decisions and architectural features, capturing the knowledge from existing industrial products, technical forums/blogs, experts' feedback and academic literature; plus our own experience using and developing blockchain-based applications. We validate our framework by applying it to dissect the most outstanding blockchain platforms, i.e., the ones behind the top 10 cryptocurrencies apart from Bitcoin. Then, we show how we applied it to a real-world case study in the insurtech domain.

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

## KEYWORDS

Blockchain, Cryptocurrencies, Software Architectures

### ACM Reference format:

Martin Garriga, Maxmiliano Arias, and Alan De Renzis. 2018. Blockchain and Cryptocurrency: A comparative framework of the main Architectural Drivers. In *Proceedings of Sample Conference*, November, 2018 (*CONF*), 8 pages. DOI: xx.xxx/xxx\_x

## 1 INTRODUCTION

Blockchain is a decentralized transaction and data management solution, well-known for being the technology behind

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CONF,

© 2018 Copyright held by the owner/author(s). xxx-x-xxxx-xxxx-x/xx/xx. . . \$15.00

DOI: xx.xxx/xxx\_x

the success of Bitcoin cryptocurrency [13]. Its main goal is to create a decentralized environment where no third party is in control of the transactions and data [26]. This technology is now mainstream because it solves problems in a way people could not before, generating a business value-add that will reach about \$176 billion by 2025 and \$3.1 trillion by 2030<sup>1</sup>. The prime reason behind this expansion is the already widespread adoption of blockchain in financial transactions and cross-border payments [6].

Even though the most popular blockchain implementation is Bitcoin, a myriad others are currently running or still in development. Different implementations vary in many ways such as their purpose, governance and efficiency, among others.

However, no single blockchain by itself can meet the requirements for all usage scenarios, e.g., those that require real-time processing. When building blockchain-based applications, we need to systematically consider the key technological features and configurations, and assess their impact on quality attributes for the overall systems [25]. Moreover, to determine which blockchain implementation should be leveraged (or even if a new one is needed) for a given application, it is crucial to be familiar with the differences among them [7].

In this paper we propose a framework to help software architects, developers, tool selectors and decision makers to adopt the right blockchain<sup>2</sup> technology for their problem at hand. Even though new blockchains increased exponentially up to 2017, nowadays it makes no sense to “reinvent the wheel” by building a custom blockchain from scratch every time; but rather to leverage, and probably combine existing, battle-hardened solutions to support new applications. For crafting our framework, we surveyed the knowledge from existing industrial products, technical forums/blogs, academic literature and our own experience using and developing blockchain-based applications.

Afterwards, we applied the framework to analyze and score the most outstanding solutions in the real-world market — i.e., the technology behind the top 11 cryptocurrencies<sup>3</sup>, giving an overview of the current ecosystem of mainstream blockchain solutions. We then show how such an assessment framework can be applied through a real-world case study in the insurtech domain.

<sup>1</sup><https://www.gartner.com/doc/3627117/forecast-blockchain-business-value-worldwide>

<sup>2</sup>For the sake of simplicity we will use the word blockchain to refer to any distributed ledger implementation, except when explicitly clarified.

<sup>3</sup>According to their market cap. See: <https://coinmarketcap.com>

The rest of the paper is organized as follows. Section 2 provides an overview of key concepts in blockchain and cryptocurrencies, as well as related work. Section 4 details our framework for assessment of blockchain technologies. Section 5 assesses the top blockchain solutions by means of our framework and then applies it to a real-world case study. Finally, Section 6 concludes the paper.

## 2 BACKGROUND

A *blockchain* is a distributed ledger, in the form of a totally ordered, back-linked list of blocks [13]. Each block contains transactions that are hashed into a binary hash tree (also called a *merkle tree*), with the top (root) of the tree stored alongside the transactions. Each block also contains the previous block's hash, thus guaranteeing integrity and determinism — i.e., any node replaying all blocks starting from the first one (genesis block) should end up with the same state as every other node [3]. This forbids to call external APIs whose responses may change over time<sup>4</sup>. Blockchain distribution is coupled with trust creation and a consensus mechanism for determining agreement on the next block to add. *Cryptocurrencies* have emerged as the first generation of blockchain-based applications, as digital currencies that are based on cryptography techniques and peer-to-peer networks. The first and most popular example is Bitcoin [1, 13].

One of the fundamental disruptions that blockchain technology is causing is the redefinition of digital *trust*, which manifests itself in a fully distributed way without anyone having to trust any single member of the network. The only trust required is that, on average, the participants of the network are not colluding against the others in a coordinated manner [25].

*Transactions* are data packages that store information — e.g., monetary value for cryptocurrencies, or results of function calls for other decentralized applications (dApps). The integrity of a transaction is checked by algorithmic rules and cryptographic techniques. A transaction, signed by its initiator, is sent to a node connected to the blockchain network, which validates the transaction and propagates it to other nodes in the network. These also validate and propagate the transaction to their peers, until it reaches all nodes in the network [25]. Transaction processing involves a *transaction fee*, given the cost imposed to the network, and as an incentives for nodes to stay honest [13].

Transactions are grouped in blocks that are appended to the existing chain, a process known as *mining*. The network aims to reach a *consensus* about the next block to be included into the blockchain by means of a *Consensus Protocol*. Their features include assuring decentralized governance, quorum structure, authentication, integrity, non-repudiation, byzantine fault tolerance and performance [11]. The de-facto consensus protocol is Proof-of-Work [2] (PoW, the one behind Bitcoin and Ethereum), which imposes miners to compute a hash function that should be efficiently verifiable,

but parameterisably expensive to compute. Given the ludicrous energy consumption of PoW<sup>5</sup>, other blockchains opted for greener but rather centralized options such as Proof-of-Stake [8] (PoS), where miner are limited to a percentage of transactions that is reflective of his or her ownership stake of the token, and lately Delegated Proof of Stake (DPoS). Finally, several blockchains provided their own hybrid or ad-hoc protocols [16].

The first generation of blockchains (e.g., Bitcoin) provided very limited capability to support programmable transactions, apart from value transfer from one account to another. The second generation (e.g., Ethereum [3]) aims to provide a general-purpose programmable infrastructure, whose programs are known as *smart contracts* [21, 25]. Originally, smart contracts were defined as the digital equivalent of a paper contract: an agreement between parties with a set of promises that are legally enforceable [21]. Nowadays, any general purpose computation that takes place on a blockchain or distributed ledger is considered a smart contract.

## 3 RELATED WORK

Yli-huumo et al. [26] identified that a majority of the blockchain-related papers focused on certain technical challenges: throughput, latency, size and bandwidth, security, wasted resources, usability, versioning, hard forks, and multiple chains [20]. In addition, they identified privacy, smart contracts, new cryptocurrencies, botnets, consensus protocols and trustworthiness. As future research directions, authors highlight scalability issues; other uses beyond cryptocurrency systems (i.e., dApps) and effectiveness of the proposed solutions — the latter being one of the contributions of our paper. In a similar direction, Alharby and Van Morsel [1] present a systematic mapping study of smart contracts. They identified four groups according to the challenges tackled: codifying, security, privacy and performance.

Scriber [18] performed a literature review and evaluated 23 blockchain implementation projects. This evaluation revealed 10 architectural or blockchain characteristics that can help determine whether blockchain is appropriate for a given problem, namely immutability, transparency, trust, identity, distribution, transactions, historical record, ecosystem and inefficiency. However, from the 23 analyzed projects, only four reached an advanced stage while the others failed or were abandoned. The paper does not provide insights regarding which of the most popular blockchains would be suitable for a given problem.

Xu et al. [25], present a taxonomy of blockchain systems and their architectural characteristics, to assist with the design and assessment of their impact on software architectures. First, authors identify fundamental properties of blockchain networks, namely: immutable data, non-repudiation, data integrity, transparency, equal rights (of participants), and trust mechanisms. Other properties include data privacy, scalability, cost and performance. Afterwards, they distilled architectural design issues, which impact on such properties: decentralization, support for client storage and computation,

<sup>4</sup><http://www.truthcoin.info/blog/contracts-oracles-sidechains/>

<sup>5</sup><https://digiconomist.net/bitcoin-energy-consumption>

scope (public/consortium-community/private), data structure, Consensus protocol and new side-chains. Then, they propose a series of decisions while designing a blockchain-based systems, that may affect those drivers. However, the decisions are intended for the development of a new blockchain rather than evaluating the adoption of an existing one.

## 4 A COMPARATIVE FRAMEWORK FOR BLOCKCHAIN IMPLEMENTATIONS

First consideration for a blockchain implementation is its *purpose*. This seems obvious, but is truly overlooked by many architects and developers. Most existing blockchains are specialized for cryptocurrencies, and might not be a good fit for other applications whose intent is different [7]. Purposes<sup>6</sup> are categorized as:

- **Currencies** used for transactions or as a store of value, e.g., Bitcoin, Litecoin<sup>7</sup>, Tether [22].
- **Exchanges and Interoperability** designed to enable communication among different blockchains, e.g., Binance Coin<sup>8</sup>, 0x<sup>9</sup>.
- **Data and Cloud Services** used to interact with data management or cloud service platforms, e.g., Golem<sup>10</sup>.
- **dApps Platforms**: used as part of a smart contract network or dApps platform, e.g., Ethereum, Cardano [8], EOS [10].
- **Gaming, Media, and Social** used for gaming, online content and social media, e.g., Steem<sup>11</sup>, Tron [19].
- **Privacy**, with built-in features to facilitate anonymous or untraceable transactions online, e.g., Monero<sup>12</sup>, Zcash<sup>13</sup>.
- **FinTech** for financial services and technologies, e.g., Ripple [17], Stellar [12].
- **Business/Enterprise** helps businesses improve efficiency, transparency, and security, e.g., Waltonchain<sup>14</sup>.
- **Others**, for example those related to prediction markets, oracles, IoT or AI projects, e.g., IOTA [15].

In the following sections, we define the architectural features to guide this analysis. Alongside, we define technological decisions and the possible values that they may assume, which finally impact on the features (see Table 1 for a summary).

### 4.1 Cost

Although adopting a blockchain is theoretically free, at least three aspects impose a cost for using the network: a variable cost for running transactions, composed by the *transaction*

*fee* [3] and *incentives* for processing transactions; and a minimal fixed cost to deploy applications (in the form of smart contracts).

The default approach is to have purely voluntary fees with dynamic minimums [13]. However, this approach can become prohibitively expensive when the network is congested: for example, transaction fees in Bitcoin have raised up to 40 USD during peaks of workload.

On the other end, implementations such as Ripple and Tron foster minimal transaction fees to prevent malicious users to perform DDoS attacks for free. Sitting in the middle, a widely used approach is to define a cost per instruction and then calculate the overall cost of the transaction (as in Ethereum), with a maximum limit in order to avoid infinite loops. Yet another approach is to impose a non-monetary fee for running transactions. For example, in IOTA, every node sending a transaction is required to validate two other transactions, which assures enough processing power.

Possible values for the *transaction fee* are: Minimal (only to avoid DDoS attacks), per transaction and per instruction. Possible values for the *incentive* are: Big (bitcoin-alike) and small (equivalent to a fee).

### 4.2 Consistency

Different strategies have been used to confirm that a transaction is securely appended to the blockchain – that is, to ensure strong *consistency*: wait for a certain number of blocks (e.g., 6 for Bitcoin, 12 for Ethereum) to have been generated after the transaction is strong consistent into the blockchain[25]; add a checkpoint to the blockchain, so that all the participants will accept the transactions up to the checkpoint as valid and irreversible. Other implementations of the distributed ledger (e.g., a DAG) can drastically reduce the time to confirmation as they do not rely on blocks with multiple transactions, but in transactions that are propagated independently in a matter of seconds.

Thus, consistency is a function of the *time to confirmation* (i.e., the number of blocks after which one can consider a transaction securely appended to the blockchain), which in turn depends on the *block production rate* (BPR, the amount of time required to mine a block), configured for each implementation at design time.

Possible values for *time to confirmation* are: seconds, minutes and hours. Possible values for *block production rate* are: 10 minutes or more, 1 to 10 minutes and seconds.

### 4.3 Functionality and Extensibility

Bitcoin’s main intent was to become a decentralized cryptocurrency [13]. Rapidly, the idea of applying it to other concepts and decentralized application (dApps) emerged, e.g., for name registration and tokens for corporate use [3], being more flexible and extensible through smart contracts. Some implementations support turing-complete smart contracts using ad-hoc languages (Solidity in Ethereum, Plutus in Cardano), while others support traditional languages (such as

<sup>6</sup>See <https://goo.gl/rDAfkK>

<sup>7</sup><https://litecoin.org/>

<sup>8</sup><https://www.binance.com/en>

<sup>9</sup><https://0xproject.com/>

<sup>10</sup><https://golem.network/>

<sup>11</sup><https://steem.io/>

<sup>12</sup><https://getmonero.org/>

<sup>13</sup><https://z.cash/>

<sup>14</sup><https://www.waltonchain.org/>

C++ in EOS) that are then compiled/transpiled to a byte-code. However, the latter are not fully supported yet in any of the existing blockchain implementation.

A latter point is *interchain communication*, allowing multiple parallel blockchains to interoperate retaining their security properties [9]. Some blockchain implementations provide native support for interchain communication (e.g., EOS), while certain frameworks allow it on top of existing implementations (e.g., Cosmos [9]).

Possible values for *smart contracts* are: Yes (specifying the language(s)), No, and Very Limited. Possible values for *Interchain communication* are: Yes, No.

#### 4.4 Performance and Scalability

Decoupling *performance* (latency and throughput) and *scalability* (with the number of nodes and clients in the system) is not entirely possible [23], thus we will group them together for analysis. Those became the bottleneck for the most popular blockchain implementations, such as Bitcoin (consensus latency of about an hour) and Ethereum. Additionally, PoW-based networks use a lot of power, equivalent to a small country such as Austria [23]. Thus performance and scalability of permissionless generic blockchains is limited by their design decisions [24], namely sequential execution of transactions and hard-coded consensus protocol.

Scalability, in turn, refers to the ability to maintain performance indicators when serving more users and transactions, limited by: (i) the size of the data on blockchain, (ii) the transaction processing rate, and (iii) the latency of data transmission. Roughly speaking, PoW offers good scalability with poor performance, whereas other protocols offer good performance for small numbers of replicas, with limited scalability. Given seemingly inherent tradeoffs between the number of nodes and performance, it is not clear today what the optimal blockchain solution is, for the majority of use cases in which the number of nodes ranges from a few tens to a few thousands.

Conclusively, performance and scalability are affected by the *transactions per second* (TPS), *block production rate*, *consensus protocol* and certain *technological choices*.

Possible values for *transactions per second* are: less than 100, between 100 and 1000; and 1000 or more. Possible values for the *Consensus Protocol* are: Proof-of-Work (PoW), Proof-of-Stake (PoS), Distributed Proof-of-Stake (DPoS), and other (specify). Possible values for technological choices are: Merkle/Patricia trees, Segregated Witness (segwit), data sharding, parallel execution of transactions, GHOST (Greedy Heaviest Observed Sub-Tree), Lightning Network, and other (specify).

#### 4.5 Security

One of the main features of blockchain is that its public ledger cannot be modified or deleted after the data has been approved by all nodes, providing data integrity and security characteristics [26]. Security issues mean bugs or vulnerabilities that an adversary might utilize to launch an

attack. Currently, the most secure implementations are PoW-based. Even though, they have a possibility of a 51% attack, where a single entity would have full control of the majority of the network's mining hash-rate and would be able to manipulate it.

Alternative consensus protocols such as PoS and DPoS may provide better performance and/or scalability, but they imply a tradeoff w.r.t. security: most tolerate up to 1/3 (33%) of malicious nodes. Other algorithms implementing Byzantine Fault Tolerance (BFT, e.g., the Ripple Consensus Protocol) may improve security up to 2/3 malicious nodes, but they impose additional restrictions such as requiring nodes to know each other.

Security is thus affected by the following technological decisions: *Fault tolerance* (possible values are 2/3 attack, 1/2 attack, 1/3 attack, and other); *ledger implementation* (possible values are Blockchain, DAG, and other); and *consensus protocol*.

#### 4.6 Decentralization

Theoretically, blockchain does not rely on any centralized node or authority, allowing data to be recorded, stored and updated in a distributed fashion. However, some blockchains introduce certain degree of centralization. In case of public, permissionless blockchain, no centralized authority or party has more power than the rest (Bitcoin), and everyone has the right to validate a transaction [23].

In the case of consortium, permissioned blockchain, only few nodes are given certain privileges over validation (PoS and DPoS-based ones such as EOS). A fully private blockchain has a centralized structure with the power to take decisions and control the validation process (e.g., Ripple and the Ripple Consensus Protocol). Permissioned blockchains are faster, more energy efficient and easily implementable compared to permissionless blockchains, but introduce certain degree of centralization. Thus, the decentralization degree is constrained by the *consensus protocol*, and the *ledger implementation*.

### 5 ASSESSMENT OF BLOCKCHAIN SOLUTIONS

The initial list of features and decisions extracted from the literature was a subset of the ones in Table 1. Those were delivered to a group of three experts, comprising researchers and industry practitioners. They evaluated the list and suggested to add, remove, group, or decompose concepts, and pointed out correlations, based on their own expertise.

Then, we proceeded to extract the technological features lying under the most popular blockchain implementations, according to their market cap<sup>15</sup>. Even though we acknowledge other possible ways to measure popularity — e.g., number of wallets, number of exchanges, transaction volume, etc. — they usually converge to a similar ranking at a given point in time [6]. Popularity is not affected by the technical decisions behind the implementation, but may impact certain

<sup>15</sup>source: <http://coinmarketcap.com>, July 2018

**Table 1: Correlation between Architectural Features and Technological Decisions**

Technological Decision	Architectural Feature					
	Cost	Consistency	Functionality	Performance	Security	Decentralization
Fees	x					
Incentive	x					
Confirmation Time		x				
Block Production Rate		x		x		
Smart Contracts			x			
Interchain			x			
Consensus Technology				x		x
Fault Tolerance					x	
Ledger					x	x
TPS				x		

features of the blockchain such as cost, time to confirmation and security. Additionally, the popularity of the underlying blockchain may be the key enabler for the success of a dApp, as demonstrated by the myriad of dApps in Ethereum<sup>16</sup> and the limited offer in others. All in all, the technological analysis of the different blockchains is summarized in Table 2.

Based on the identified architectural and technological aspects, we conducted a second round of feedback through structured interviews with the experts, in order to come up with a quantitative assessment of the top blockchain solutions. They completed a questionnaire<sup>17</sup>, assigning scores for each architectural feature to the different blockchain implementations (from *very low* to *very high*) which were then fuzzified into numerical scale from 1 to 5. For example, if an expert considers that *Bitcoin* has a very high *Cost*, then in the questionnaire she marks the corresponding cell as “very high”.

Experts fulfilled the scorecards as described above, also declaring their confidence (fuzzified from “low” to “very high”). Both the confidence values and the scores were then defuzzified using a triangular membership function [14] and combined on a weighted average scheme. The triangular function allows one to map and normalize the linguistic scale to a given scale, in the range [1, 5] for the scores and [0, 1] for the confidence values.

The result of the process is a normalized weight matrix, which numerically represent the scores for each feature on each blockchain implementation as values in the interval [1, 5], as shown in Table 3. The information contained on the scorecard allows software engineers, architects and decision makers to assess the different blockchain implementations,

being able to select the most suitable one for their problem at hand, as illustrated in the following section.

## 5.1 Open Challenges

From the analysis of literature, top blockchain implementations, and feedback from experts, we were able to identify some open challenges and concerns regarding the future of the field. Although all blockchain implementations promise to be secure and efficient, most of them fall short in some of these aspects. Particularly, Proof-of-Stake and Distributed Proof-of-Stake blockchains are risky since critical decisions fall on a small group of people or company. Even though, in traditional implementations based on Proof-of-Work, grouping of miners into mining pools are effectively centralizing these networks [5].

In this direction, platforms with the potential to be scalable and energy-efficient will be the weapon-of-choice for dApps development. Other “legacy” blockchains such as Bitcoin will remain for big, sporadic transactions and as long-term investment.

Blockchain is still an emerging technology, thus not a lot of developers are concerned with the principles of Software Engineering applied to blockchain-based systems. Moreover, the lack of guidelines and standards on how to design software architectures that include smart contracts as part of the system calls for further attention [4, 25].

Finally, only a handful of experiences in real-world dApps exist<sup>18</sup>, and still a lot of controversy on whether an application requires the use of blockchain. The emergence of frameworks like the one presented in this paper may help to overcome such difficulties.

As threats to validity for our approach, we can highlight the following. First, the short number of experts that participated in the analysis. This might result on a possible bias, but avoids the answers to be meaningless because of the lack of experience of surveyed experts. One should also note that the number of experts in the blockchain world is not that big. Another concern is the number of analyzed cryptocurrencies, since among the top ones there is Bitcoin itself and two Bitcoin forks (Litecoin, Bitcoin Cash). Some revolutionary approaches may have not gained momentum yet, reason why we are planning to extend our analysis, covering more implementations.

## 5.2 Case study: A trusted images application for the insurtech domain

In this section we illustrate the value of our framework to assess blockchain alternatives for a real-world application. Photofied<sup>19</sup> is a mobile application developed by Fidtech, as a solution for worldwide insurance activity. It certifies digital images in the blockchain for fraud prevention, granting reliability of the status of an insurable risk, both at policy emission and execution stages. All images taken with Photofied are certified by means of an ad-hoc protocol, namely Three Way Certification (3WC). 3WC features blockchain, a P2P

<sup>16</sup><https://github.com/avadhootkulkarni/UltimateICOCalendar>

<sup>17</sup><https://goo.gl/forms/8mE1mdJ55VJRJw2E3>

<sup>18</sup><https://dappradar.com/>

<sup>19</sup><https://photofied.tech>

Table 2: Qualitative assessment of the most popular blockchain alternatives according to the technological characteristics

	Bitcoin (BTC)	Ethereum (ETH)	Ripple (XRP)	Bitcoin Cash (BCH)	EOS (EOS)	Litecoin (LTC)	Stellar (XLM)	Cardano (ADA)	Iota (MIOTA)	Tether (USDT)	Tron (TRX)
<b>Purpose</b>	Currency	Platform	Fintech	Currency	Platform	Currency	Fintech	Platform	Other (IoT)	Currency	Media/Social
<b>Fees</b>	Per transaction	Per Instruction (configurable)	Minimal	Per Transaction	Per instruction	Per transaction	Per operation	Per transaction size	Approve 2 transactions	20 USD	Minimal
<b>Incentive</b>	12.5 BTC/hash	Configurable	N/A	12.5 BCH/hash	Configurable (by BPs)	6.5 LTC	N/A	Availability based	N/A	No	Configurable (by SRs)
<b>Block Production rate</b>	10 min	10-19 sec	Not specified	10 min	0.5 sec	2.5 min	5 seconds	20 sec	N/A	10 min	15 sec
<b>Confirmation time</b>	60 min (6 blocks)	1 min	seconds	60 min	1 second	20 min (6 blocks)	30 seconds	1 min	2 min (scalable)	60 min (6 blocks)	1 min
<b>TPS</b>	7	15	1500	23	1000+	50	1000+	250	1000+	7	1000+
<b>Smart contracts</b>	Very Limited	Yes	Very Limited	Very Limited	Yes	Very Limited	Very Limited	Yes	No	No	Yes
<b>dApps</b>	No	Yes	No	No	Yes	No	No	Yes	Yes, IoT level	Very Limited	Yes
<b>Languages</b>	C++, Script	Solidity	Any	Script	Any	Script	Any	Functional	Any	Ruby	Java
<b>Interchain</b>	No	No	No	No	Yes	No	No	Yes	Yes	No	Yes
<b>Consensus Algorithm</b>	PoW (sha-256)	PoW (ethash)	PoC (Ripple CP)	PoW (sha-256)	Delegated PoS	PoW (scrypt)	Federated BFT Stellar CP	Pos (Ourboros)	Markov Chain (MCMC)	PoR (Omni)	Pos (Tendermint)
<b>Enhanced Technology</b>	Merkle Trees, Segwit	Patricia Trees, Sharding	-	Merkle Trees	Merkle Proofs, Segwit	Merkle Trees, Segwit	-	Sharding	Tangle for IoT Scale	Merkle trees, Segwit	Graph database
<b>Fault Tolerance</b>	51% CPU attack	51% ether attack	20% malicious nodes	51% CPU attack	66% Block Producers	51% CPU Attack	Asymptotic Security	Asymptotic Security	51% CPU attack	51% CPU Attack	33% Byzantine Failure
<b>Ledger</b>	Blockchain	Blockchain	Interledger Protocol	Blockchain	Blockchain	Blockchain	Blockchain	Blockchain	Directed Acyclic Graph	Blockchain	Blockchain

**Table 3: Quantitative assessment of blockchain solutions according to experts’ feedback regarding architectural decisions.**

	BTC	ETH	XRP	BCH	EOS	XLM	LTC	ADA	USDT	MIOTA	TRX
<i>Popularity</i>	1	2	3	4	5	6	7	8	9	10	11
Cost	1.33	2	4.66	1.66	5	4.66	2.66	4.33	5	5	5
Consistency	1.33	2.33	4.33	1.33	5	4	2	3.66	1	4.66	4
Functionality	2	5	1.33	2	5	1.33	2	4.33	2	3.66	5
Performance	1.33	1.66	4.33	2	4.66	4	2.33	3	1	5	4.66
Security	4	4	2.33	4	3.33	4	4	4	3.33	3.66	3.33
Decentralization	5	3.33	1	4.33	2.66	2.33	3.66	3.33	1.33	2.33	3.33
<b>Total</b>	14.99	18.32	17.98	15.32	25.65	20.32	16.65	22.65	13.66	24.31	25.32

distributed file system and digital signature to grant the immutability, perdurability and verifiability of the images.

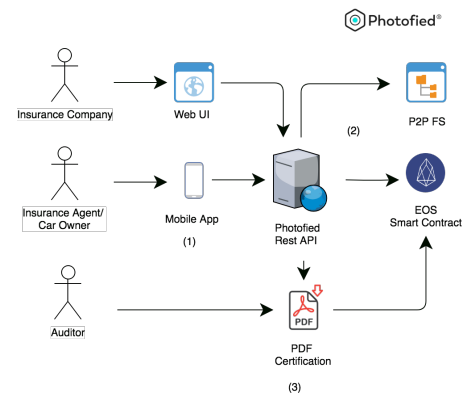
Figure 1 depicts the architecture of the application, where insurance agents or car owners use the mobile app (1) to send packages (containing images and metadata to certify) to the Rest API. The server forwards the package to the EOS smart contract and the p2p FS (2). After that, each certification is printed to PDF, allowing offline audit (3). At any time, insurance companies can access the certifications using a Web interface.

As a first stage, Photofied is used during the policy emission process, certifying the images taken by the insurance agent. The images are later audited only if needed, thus there is no need for fast transaction confirmation.

End users are neither supposed to know about blockchain or cryptocurrencies and/or own accounts; nor responsible for paying for the service — thus, Cost should be low to attract insurance companies as potential customers.

Also, as images can be captured either by insurance agents or car owners, the application needs to identify who took the images, when, and where, providing functionality and flexibility. Each image should be independently certified, which implies a high number of transactions, calling for performance. Additionally, security and consistency are concerns to maximize, granting the trustability of certified images. Each certification, containing images and metadata (username, GPS coordinates and mobile device’s information), has to be auditable by third parties without using Photofied services/servers (i.e., by querying directly the underlying blockchain).

At application’s design time, the developers were not aware of all the advantages and drawbacks of each blockchain platform. A first selection, purely based on popularity, led first to a Bitcoin-based implementation and then an Ethereum-based one. The former used a naïve (*Data hash* → *Timestamp*) structure, given the lack of proper smart contracts support in the Bitcoin blockchain. The latest used a smart contract that stored a mapping from each uploaded piece of information to the account from which it was uploaded along with some metadata.

**Figure 1: Photofied architecture overview**

However, both implementations suffered the drawbacks of the underlying blockchains (See Table 2 and Table 3), requiring an expensive transaction fee and relying on congested networks, with a prohibitively low number of transactions per second. In parallel, the number of novel blockchain platforms increased exponentially, paving the way for the adoption of a most suitable one. After crafting the comparison framework and fine-tuning the importance for each feature, the most suitable options were EOS and TRON, due to the nonexistent fee, high transactions per second and high reliability. To untie, EOS was selected based on its popularity, as it implies more active developers, nodes, available dApps and supporting community. Also, by that time, the TRON mainnet was not yet online and had no near release date.

All in all, the current version of Photofied is running using EOS in a collaborate effort with EOS Argentina<sup>20</sup>, one of the top block producers on the EOS blockchain. The smart contract that handles all the needed data is managed by a custom account in charge of time-stamping transactions — combining block timestamp and server timestamp. This way, end users don’t need an EOS account, the application can

<sup>20</sup><https://www.eosargentina.io/>

run on mobile devices as it uses a central server that manages the transactions (ensuring high transactions per second and reliability). Finally, as EOS is a public, decentralized blockchain, each piece of certified data can be audited from any EOS node.

## 6 CONCLUSIONS AND FUTURE WORK

Nowadays, the number and variety of existing blockchain implementations continues to increase. Adopters should focus on selecting the best one — rather than developing yet another one from scratch — to support their decentralized applications (dApps). In this paper we presented a framework to aid software architects, developers, tool selectors and decision makers to adopt the right blockchain technology for their problem at hand. The framework exposes the correlation between technological decisions (such as consensus protocols and support for smart contracts) and architectural decisions (such as cost and decentralization). For crafting our framework, we surveyed the knowledge from existing industrial products, technical forums/blogs, experts' feedback and academic literature; plus our own experience using and developing blockchain-based applications.

We have shown the suitability of our framework in two ways. First, we applied it to analyze the most popular blockchain implementations in the real world, according to their market cap. This shed light regarding the current ecosystem of mainstream blockchain solutions. Second, we shown how the framework can be applied by dApps developers through a real-world case study: a trusted images application for the insurtech domain. Developers were able to successfully select a new blockchain and migrate their application based on the insights obtained from the framework.

Our future work comprises fine-tuning the framework by engaging yet more experts from the blockchain world. Afterwards we plan to assess the top 50 implementations to have a complete panorama of the existing solutions, beyond the Bitcoin and Ethereum hype. Finally, we are currently developing a series of questions in the form of a wizard, to guide practitioners in the use of our framework for selecting the most suitable blockchain.

## ACKNOWLEDGMENTS

The authors would like to thank to Nicolas Arias, Diego Anabalon, Nahuel Vazquez and Claudio Arce from Fidtech, for their helpful insights. This work is partially supported by ANPCyT project PICT-1725-2017 and CONICET.

## REFERENCES

- [1] Maher Alharby and Aad van Moorsel. 2017. Blockchain-based smart contracts: A systematic mapping study. *arXiv preprint arXiv:1710.06372* (2017).
- [2] Adam Back et al. 2002. Hashcash-a denial of service countermeasure. <http://www.hashcash.org/papers/hashcash.pdf>.
- [3] Vitalik Buterin et al. 2014. A next-generation smart contract and decentralized application platform. *white paper* (2014).
- [4] Giuseppe Destefanis, Michele Marchesi, Marco Ortu, Roberto Tonelli, Andrea Bracciali, and Robert Hierons. 2018. Smart contracts vulnerabilities: a call for blockchain software engineering?. In *Blockchain Oriented Software Engineering (IWBOSE), 2018 International Workshop on*. IEEE, 19–25.
- [5] Arthur Gervais, Ghassan Karame, Srdjan Capkun, and Vedran Capkun. 2014. Is bitcoin a decentralized currency? *IEEE security & privacy* 12, 3 (2014), 54–60.
- [6] Garrick Hileman and Michel Rauchs. 2017. Global cryptocurrency benchmarking study. *Cambridge Centre for Alternative Finance* (2017).
- [7] Zane Hintzman. 2017. *Comparing Blockchain Implementation*. Technical Report. SCTE/ISBE Society of Cable Telecommunication Engineers and International Society of Broadband Experts, Exton, PA, USA.
- [8] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*. Springer, 357–388.
- [9] Jae Kwon and Ethan Buchman. 2016. Cosmos: A network of distributed ledgers. *URL https://cosmos.network/whitepaper* (2016).
- [10] Daniel Larimer et al. 2018. *EOS Whitepaper*. Technical Report. block.one. <https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md>.
- [11] Juri Mattila et al. 2016. *The blockchain phenomenon—the disruptive potential of distributed consensus architectures*. Technical Report. The Research Institute of the Finnish Economy.
- [12] David Mazieres. 2015. The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation* (2015).
- [13] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).
- [14] Witold Pedrycz. 1994. Why triangular membership functions? *Fuzzy Sets and Systems* 64, 1 (1994), 21 – 30. [https://doi.org/10.1016/0165-0114\(94\)90003-5](https://doi.org/10.1016/0165-0114(94)90003-5)
- [15] Sergei Popov. 2017. *IOTA Whitepaper*. Technical Report. IOTA Foundation, Berlin, Germany. <http://iotatoken.com/IOTA.Whitepaper.pdf>.
- [16] Lakshmi Siva Sankar, M Sindhu, and M Sethumadhavan. 2017. Survey of consensus protocols on blockchain applications. In *Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on*. IEEE, 1–5.
- [17] David Schwartz, Noah Youngs, Arthur Britto, et al. 2014. The Ripple protocol consensus algorithm. *Ripple Labs Inc White Paper* 5 (2014).
- [18] Brian A Scriber. 2018. A Framework for Determining Blockchain Applicability. *IEEE Software* 35, 4 (2018), 70–77.
- [19] Justin Sun et al. 2017. *Tron Whitepaper*. Technical Report. tron network, Beijing, China. <https://tron.network/>.
- [20] Melanie Swan. 2015. *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc."
- [21] Nick Szabo. 1997. Formalizing and securing relationships on public networks. *First Monday* 2, 9 (1997).
- [22] Tether International. 2018. *Tether Whitepaper*. Technical Report. Tether International Limited, Hong Kong. <https://tether.to/wp-content/uploads/2016/06/TetherWhitePaper.pdf>.
- [23] Marko Vukolić. 2015. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *International Workshop on Open Problems in Network Security*. Springer, 112–125.
- [24] Marko Vukolić. 2017. Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*. ACM, 3–7.
- [25] Xiwei Xu, Ingo Weber, Mark Staples, Liming Zhu, Jan Bosch, Len Bass, Cesare Pautasso, and Paul Rimba. 2017. A taxonomy of blockchain-based systems for architecture design. In *Software Architecture (ICSA), 2017 IEEE International Conference on*. IEEE, 243–252.
- [26] Jesse Yli-Huomo, Deokyoong Ko, Sujin Choi, Sooyong Park, and Kari Smolander. 2016. Where is current research on blockchain technology? A systematic review. *PloS one* 11, 10 (2016), e0163477.