A Requirements-Driven Methodology for the Proper Selection and Configuration of Blockchains

Mirko Staderini, Enrico Schiavone✉, Andrea Bondavalli
Department of Mathematics and Informatics, University of Florence, Florence, Italy
mirko.staderini@stud.unifi.it,{enrico.schiavone, bondavalli}@unifi.it

Abstract—**In recent years, the interest in blockchain has grown exponentially, and nowadays it is foreseen as a technology with the potential to revolutionize the way data is maintained and transferred around the globe. The reason of this excitement is ascribable to the ability of enabling new forms of transactions and interactions between mistrusting and decentralized entities. Indeed, it has attracted interests and huge investments from enterprises, and it is predictable that in a near future many industries will adopt it. However, it is not a panacea and in some cases may even become useless or not convenient. Moreover, even when it can really constitute an added value, selecting the proper blockchain and configuring it may not be trivial. Trying to go beyond the hype and to address this problem, this paper proposes a methodology addressing: i) whether, given a specific problem requirements, the blockchain is a proper solution for it ii) in such a case which is the blockchain category more suitable, and finally iii) guiding the designer throughout its configuration.**

Keywords—Blockchain; Configuration; Requirements-Driven; Flow Diagram.

I. Introduction

While it has been known primarily for being the underlying technology of Bitcoin [1], and is often associated with finance and cryptocurrencies, blockchain is rapidly becoming one of the most exciting areas of interest for researchers, investors, and companies also in many other domains: scientific, social, humanitarian, political and more [2].

The reason of this huge excitement is ascribable to its properties of enabling mistrusting entities, applications and systems to interact in a fully decentralized fashion without the need for any Trusted Third Party (TTP). Blockchains are indeed trustless and decentralized databases of records, distributed applications or smart contracts that can be shared and executed among peers. Important properties are provided by such technology, including but not limited to integrity, non-repudiability, pseudo-anonymity, fault tolerance as a consequence of redundancy, and transparency. Examples of blockchain-based applications are: financial services, ownership or provenance tracking of physical and digital assets, voting, etc.

Motivation. Without any doubt, blockchain has been a hot issue in recent years, and in the very near future, a wide range of businesses and industries, which for many reasons need to keep a register, will probably adopt it into their existing platforms. Blockchain-as-a-Service (BaaS) will facilitate this adoption, delegating to the service provider the setup and the complex technical decisions. But this approach can be seen as reintroducing centralization in a decentralized system. To avoid this problem, a system designer in charge of selecting and configuring the proper blockchain for its company or project, has to deal with a huge number of issues, as types of blockchains are not a Boolean choice. In fact, features of *public*, *private*, and *permissioned* ledgers can be combined in various ways, giving birth to a huge variety of custom blockchains [3], [4]. For this reason, depending on the given requirements, a specific blockchain can be designed to best satisfy the needs

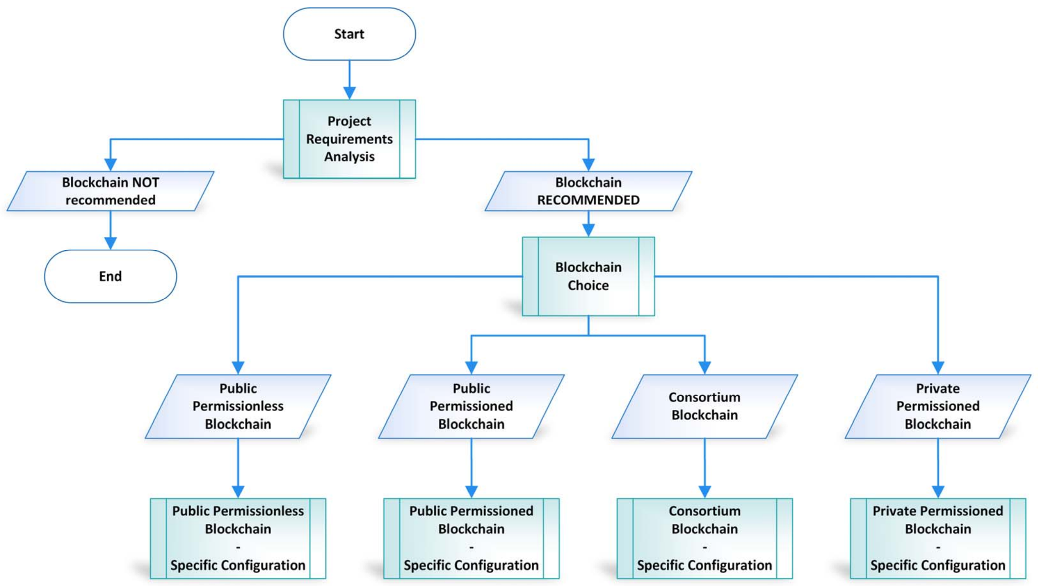
Our Contribution. This paper proposes a methodology to assist a designer in determining whether a blockchain is or not the most appropriate solution to address a specific problem, given its requirements. Then, another challenge we address is the selection of the proper category of blockchain (i.e., public or private, and permissionless or permissioned), as many alternatives do exist and this decision making process may not be trivial. Finally, the methodology we propose guides also the configuration of the selected blockchain, considering criteria related to consensus algorithm, smart contracts support, security measures, privacy, anonymity, data computation, and storage.

The remainder of the article is organized as follows. Section II describes the phases of our new requirements-driven methodology, addressing: i) the suitability of a blockchain, ii) the choice of the most appropriate blockchain category, and iii) its proper configuration. Related work is in Section III while Section IV concludes the paper and indicates future directions.

II. The Proposed Methodology

We have to acknowledge that nowadays there is a tendency towards considering the blockchain a solution for many issues, mainly because of the wide range of crucial properties it can provide, or just because of its popularity: it is one of the emerging technologies at the peak of inflated expectations as well as artificial intelligence or augmented reality [9]. However, a system designer, having in mind the system requirements, should consider whether this technology is actually applicable to the specific problem and convenient for its solution.

The approach we propose to support the designer in this task is described with the help of flow diagrams; in part it can be considered an extension of [4], with the introduction of more criteria in the requirements analysis step and, especially, because it addresses not only the general choice of the blockchain category, but also its configuration. An overview of the whole methodology is shown in Figure 1. The flow begins at the *Start* block and goes through the following sub processes:



Start
Project Requirements Analysis
Blockchain NOT recommended
Blockchain RECOMMENDED
End
Blockchain Choice
Public Permissionless Blockchain
Public Permissioned Blockchain
Consortium Blockchain
Private Permissioned Blockchain
Public Permissionless Blockchain - Specific Configuration
Public Permissioned Blockchain - Specific Configuration
Consortium Blockchain - Specific Configuration
Private Permissioned Blockchain - Specific Configuration

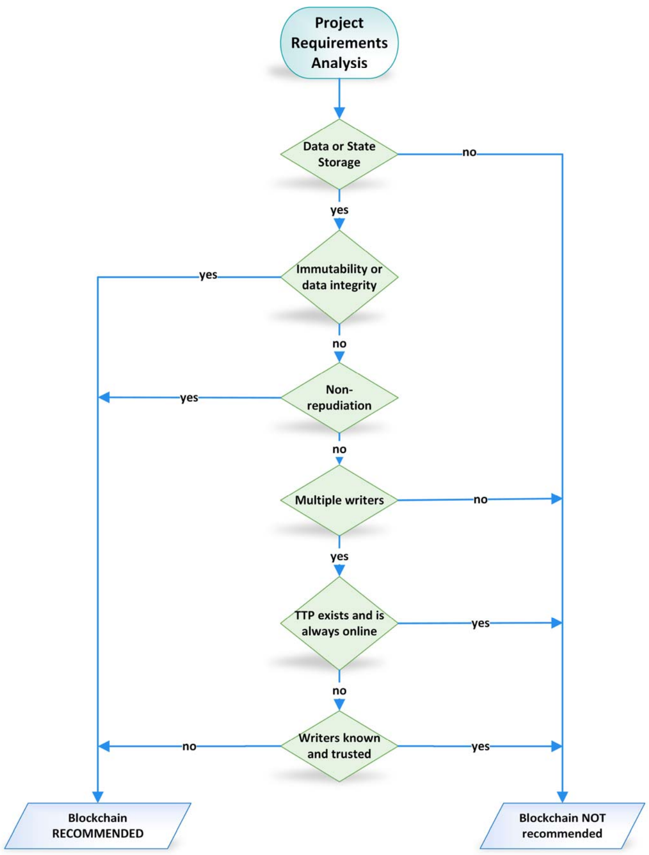
Figure 1 Overview of the Proposed Methodology

- *Project Requirements Analysis*, for the dichotomous choice between blockchain recommended or not recommended.
- *Blockchain Choice,* the step in which, based on blockchain-specific criteria, the designer is guided on the choice of the most suitable blockchain category [3].
- *Specific Configuration,* which assists the designer throughout the decision making process for the configuration of the blockchain compliantly with the chosen category and the given project requirements.

A. Project Requirements Analysis. Blockchain: Yes or No?

This Section describes the first step of proposed methodology, which consists in determining whether a blockchain is or not the most appropriate solution to address a specific problem, given its requirements. In other words, it consists in answering the question: "Blockchain: Yes or No?" It is called Project Requirements Analysis, and the related flow diagram is shown in Figure 1. It extends [4], introducing immutability, integrity and non-repudiability as additional criteria. It has to be noticed that some of the criteria directly drive to the result of recommending blockchain or advice against its usage.

- *Data or State Storage.* The first criterion considered is the requirement of designing a system capable of storing data or system state. If in the project no information needs to be stored, no blockchain is needed at all, as well as any traditional database. Thus, if this is not a requirement, the diagram brings to the answer "Blockchain NOT recommended".
- *Immutability* and *Data Integrity*. If immutability is a requirement, then we can recommend using a blockhain, as this is probably the most distinctive property of any blockchain. At the same decision block of the diagram we analyze also Integrity, as these two requirements are both closely related to cryptography. If a project requires data protection from unauthorized modifications, than this requirement can be met with a blockchain.
- *Non-repudiation.* Non-repudiation [8] is another fundamental property which can be satisfied using blockchains, thus if it is considered a requirement, the diagram directly produces the output of the first phase of the methodology: "Blockchain RECOMMENDED".
- *Multiple writers.* This block of the diagram is related to the decision regarding *data or state storage*, and in particular it considers the multiplicity of entities in charge of writing [4]. If only one entity is a writer, thus a common database is probably most appropriate than a blockchain, i.e., in terms of throughput and latency.



Project Requirements Analysis
Data or State Storage
no
yes
Immutability or data integrity
yes
no
Non-repudiation
yes
no
Multiple writers
no
yes
TTP exists and is always online
yes
no
Writers known and trusted
no
yes
Blockchain RECOMMENDED
Blockchain NOT recommended

Figure 2 Project Requirements Analysis: Blockchain YES or NO?

- *Trusted Third Party always online.* A Trusted Third Party (TTP) is an entity which facilitates interactions between mutually mistrusting entities. If in the system a TTP is required and it is planned to be always online: entities can delegate to it write operations as transactions or state changes. Therefore, the TTP plays the role of a trusted deliverer and verifier. In this case, a blockchain, known for being a trustless technology, becomes useless, and the methodology brings to the related output. Otherwise, it can happen that the involvement of a TTP is planned but not for being always online: in this case it could play the role of an authority giving authorizations for permissioned blockchains. Alternatively, a TTP may not exist at all. In the latter both situations we cannot exclude the recommendation of using a blockchain, and the analysis of the following decision block becomes necessary.
- *Writers are known and trusted.* If all the entities interested in writing, know and mutually trust each other, a blockchain is superfluous, then not recommended. Our advice, in this case, is to consider a traditional database shared among the writers.

B. *Choosing the Blockchain Category*

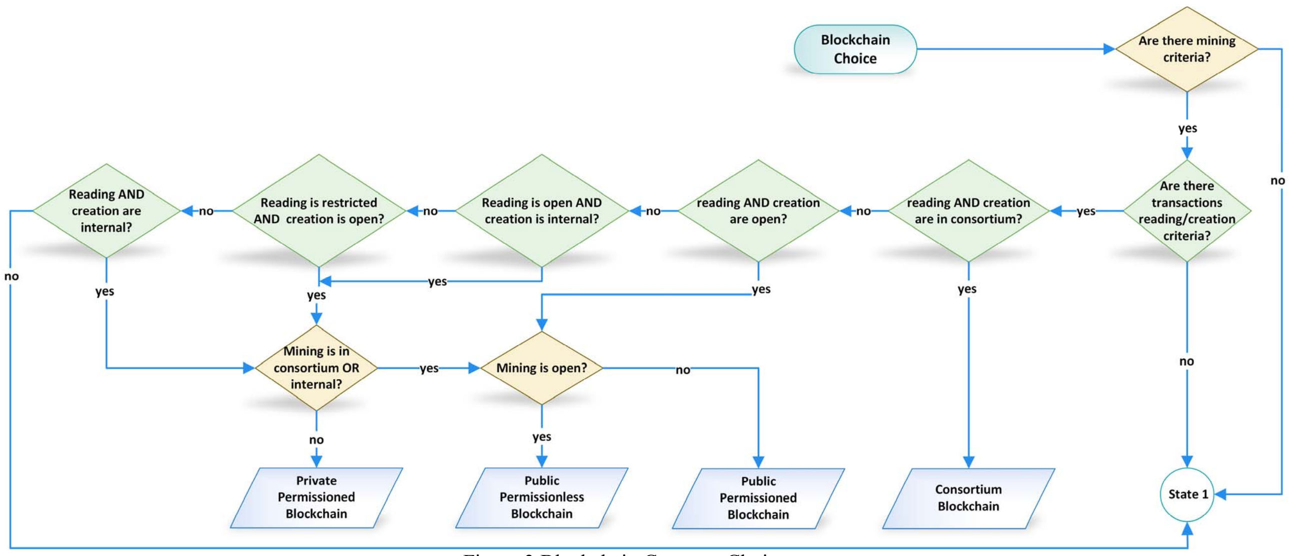
Supposing that the outcome of the Project Requirements Analysis is that adopting a distributed ledger is recommended, the subsequent step guides the designer in the choice of the most suitable blockchain category between the possible alternatives.

The *Blockchain Choice* step first considers reading and writing operations as main criteria for the choice. As described in [4], these terms indicate multiple operations. In the flow diagram of Figure 3, we split reading in two criteria, separately considering: i) consulting the state of the ledger (referred as *reading* in the diagram), and ii) creating transactions (referred as *creation*). The possible configurations are: *open*, *restricted*, and *internal*, meaning respectively that the operation is permitted to any node, permitted to some privileged nodes only, and permitted to some privileged nodes belonging to a unique organization only.

We define as *mining criterion* a project requirement which establishes the entities allowed to perform writing operations [4]; the possibilities we consider, as shown in the diagram, are: *open* mining, meaning that every peer node can be a miner, *consortium* mining, where only some nodes are allowed to write, but they can be entities belonging to different organizations, or *internal* mining, where the miners are just a subset of the nodes of the same organization. However, it can happen that for a given project, the mining criterion is not known a priori, thus in order to conclude the choice step, we have to consider more criteria. Where this is the case, the flow brings to a sub-diagram, indicated by *State 1*; the same situation occurs if reading or creation criteria are missing.

As an example, let us consider what happens when State 1 is reached: Figure 4 shows the sub-process of Blockchain Choice step in which scalability, interoperability and flexibility criteria are considered. *Scalability* is a non-functional property of blockchains which includes transaction processing rate, number of nodes participating, and latency of data transmission. For this criterion, public permissionless blockchains have been often criticized. Instead, studies exist that have shown how, i.e. with Hyperledger Fabric [10], sacrificing decentralization greatly improves scalability: this is why in our methodology permissioned and especially consortium/private blockchains are considered better suited in case of medium and high scalability requirements.

Interoperability is intended as the need of cooperation between different organizations sharing a unique ledger; it is typical of consortium blockchains, and, therefore, it can serve as a discriminating criterion for the blockchain choice. If interoperability is missing as a requirement, the sub-process of blockchain choice step considers *flexibility*: we define a blockchain as flexible if it allows the designer to perform changes in its configuration, i.e. in terms of permissions attribution. The highest flexibility is attributable to private blockchains, which are managed and controlled by a single organization: on the contrary, changes in permission attribution in a consortium blockchain must be agreed between all the partners. If State 2 is reached, meaning that scalability is not expressed in the requirements list, once again the blockchain choice step has to be addressed through more criteria.


Blockchain Choice
Are there mining criteria?
yes
no
Are there transactions reading/creation criteria?
yes
reading AND creation are in consortium?
no
reading AND creation are open?
no
Reading is open AND creation is internal?
no
Reading is restricted AND creation is open?
no
Reading AND creation are internal?
no
yes
yes
yes
yes
yes
Mining is in consortium OR internal?
yes
Mining is open?
no
no
yes
Private Permissioned Blockchain
Public Permissionless Blockchain
Public Permissioned Blockchain
Consortium Blockchain
no
State 1
Figure 3 Blockchain Category Choice

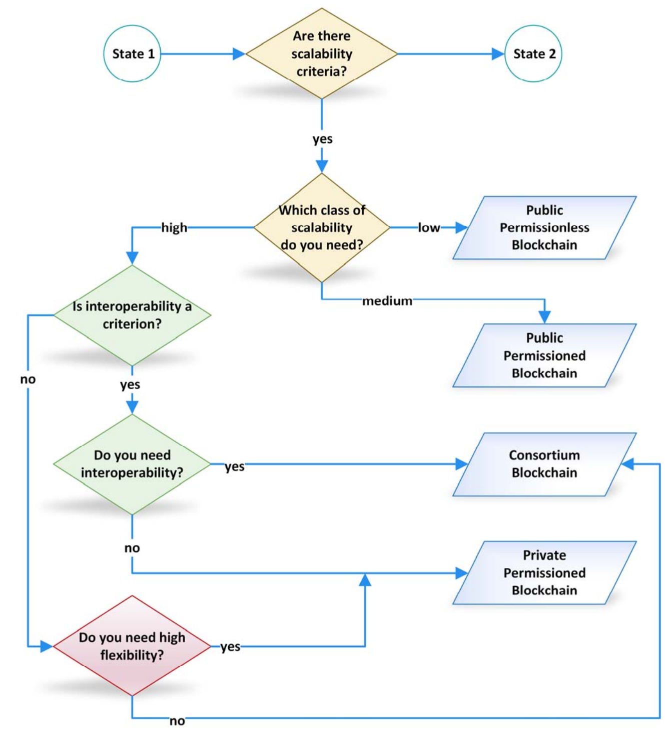

State 1
Are there scalability criteria?
State 2
yes
Which class of scalability do you need?
high
low
medium
Public Permissionless Blockchain
Public Permissioned Blockchain
Is interoperability a criterion?
no
yes
Do you need interoperability?
yes
Consortium Blockchain
no
Private Permissioned Blockchain
Do you need high flexibility?
yes
no

Figure 4 One of the Sub-processes of Blockchain Category Choice

Cost is the subsequent criterion addressed. It is necessary to specify that it can be divided into different components, as transactions cost (in cryptocurrency units or similarly, e.g., gas in Ethereum), and operative cost (computational, storage). At this phase of the methodology, the flow diagram addresses the transactions cost only; then, operative cost regarding data computation and storage is addressed during the configuration phase of the methodology (Section II.C). At the current phase, private and consortium blockchains are suggested in case of a cost-effective optimization is a mandatory requirement [16]. Instead, public blockchains, in particular permissionless ones, are often considered more expensive from this point of view.

The set of other criteria considered from State 2 on, is the following: *performance, decentralization, availability, anonymity, privacy, confidentiality, transparency,* and *trustless*. This generates a huge number of sub-diagrams. Due to space constraints, diagrams not in the paper are publicly accessible in [12]. However, we summarize in Table I the outcome of the methodology for the criteria not described so far, where the highest number of * symbols identifies the most appropriate blockchain category for every respective criterion.

TABLE I. THE OUTCOME OF BLOCKCHAIN CATEGORY CHOICE PHASE FOR THE REMAINING CRITERIA (NOT ADDRESSED IN FIGURE 4)

Criterion	Public perm.less	Public perm.ned	Consortium	Private
performance	*	*	**	***
decentralization	****	***	**	*
availability	****	***	**	*
anonymity	***	**	*	*
privacy	*	**	***	****
confidentiality	*	**	***	****
transparency	****	***	**	*
trustless	****	***	**	*

C. *Specific Configuration of Blockchains*

The last step of the proposed methodology consists in the blockchain configuration. It is composed of five stages*: i) Consensus Algorithm*, *ii) Smart Contracts*, *iii) Security Measures*, *iv) Privacy and Anonymity*, and *v) Data Computation and Storage.*

The specific configuration varies depending on the output of the choice phase. In the following of the paper we choose as exemplary output the private permissioned blockchain, and we describe the configuration step by step. The diagram in Figure 5, for the consensus algorithm configuration related to that output, is the only diagram of configuration step that we show in the paper. The charts of all the configurations steps and for all the categories of blockchains are available in [12].

- *Consensus Algorithm.* The algorithms considered here constitutes a selection of the most widely adopted consensus algorithms, and for which sufficient data regarding their properties and performance exist. However, it would not be difficult to extend the proposed methodology including other alternatives.
  As shown in Figure 5, the first criterion analyzed is *transactions finality:* depending on the algorithm, a transaction that is included in a block can be whether immediately or probabilistically considered final. Supposing that this parameter is present into the system requirements list, the designer can make use of the methodology and obtain a set of suggested algorithms. If the choice is probabilistic finality, the possible outcome is one algorithm between PoS, PoET [6] and Randomized

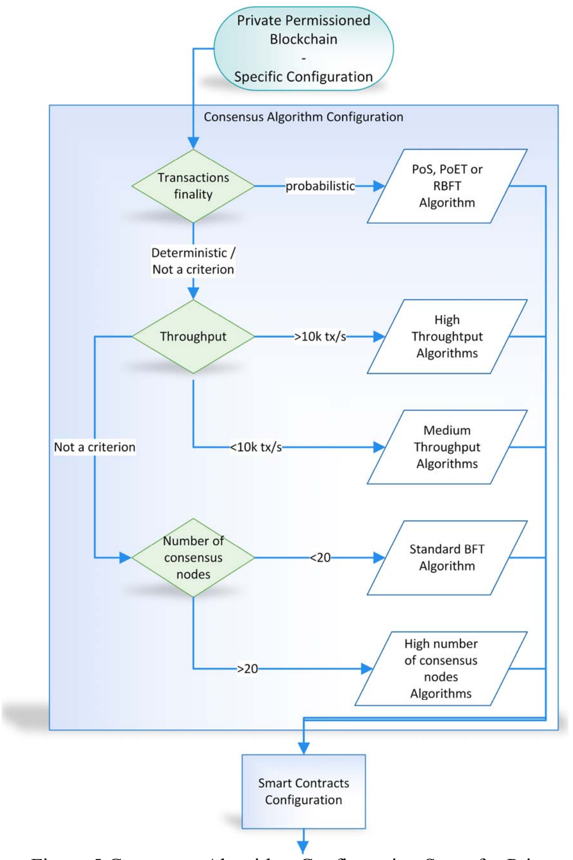

Private Permissioned Blockchain
-
Specific Configuration
Consensus Algorithm Configuration
Transactions finality
probabilistic
PoS, PoET or RBFT Algorithm
Deterministic / Not a criterion
Throughput
>10k tx/s
High Throughtput Algorithms
Not a criterion
<10k tx/s
Medium Throughput Algorithms
Number of consensus nodes
<20
Standard BFT Algorithm
>20
High number of consensus nodes Algorithms
Smart Contracts Configuration

Figure 5 Consensus Algorithm Configuration Stage for Private Permissioned Blockchains

BFT [11]. It has to be noticed that also PoW is based on a probabilistic transaction finality model, but it is excluded because the choice is limited to algorithms suitable for permissioned blockchains, in accordance with the output of blockchain choice step. In this way, the range of possible outcomes is reduced to three consensus algorithm. Then, the flow continues considering the next stage, regarding smart contracts configuration.

Otherwise, if the desired transactions finality is deterministic, as well as if the criterion is not addressed at all in the requirements list, we follow the alternative arrow coming out from the transactions finality decision block, which brings to the subsequent criterion to be analyzed: the *throughput*. Based on the throughput, we distinguish two classes of algorithms suitable for private permissioned blockchains. Algorithms capable of processing between 100 and 10^4 transactions per second, that we call *Medium Throughput Algorithms*, are: Randomized BFT, Hybrid BFT, Scalable-BFT, Kafka, and Tendermint. Algorithms capable of processing more than 10^4 transactions per second, that in the diagram correspond to *High Throughput Algorithms*, are: Optimistic BFT, BFT, PBFT, XFT, BFT-SMaRT, PoA.

If instead also the throughput is not considered, the *number of consensus nodes* can serve as a criterion guiding the consensus algorithm choice. This number should not be confused with the total number of nodes in the blockchain. The most suitable choice, if this number is not greater than 20, is BFT algorithm (in fact, the first algorithm of BFT family, referred as *Standard BFT* in the diagram of Figure 5, has limited scalability, and has been tested with a maximum of 20 nodes [11]). With more than 20 nodes involved in the consensus mechanism, the preferable choice is one algorithm between: XFT, Optimistic BFT, PBFT, Hybrid BFT, Randomized BFT, and Scalable-BFT.

- *Smart Contracts.* As shown in Figure 5, after the consensus algorithm configuration stage, the flow goes into the smart contracts configuration stage. If smart contracts support is not required by the project, this stage is completely skipped. Otherwise, the configuration continues considering the *smart contracts supported* criterion, checking whether the chosen algorithm supports smart contracts or not. If the consensus algorithm is not suitable for this purpose, the designer is pointed in the direction of a compatible solution, if exists. Based on our findings, PBFT, SIEVE and XFT are examples of available solutions for permissioned blockchains [6].

  The next criterion is *external interactions* [12] and the methodology permits to choose the type of Oracle between the following possibilities: Software Oracle, Hardware Oracle, Inbound Oracle, Outbound Oracle, and Consensus Based Oracle. Again, the decision must be compatible with the blockchain category and the consensus algorithm.
- *Security Measures.* This stage of the configuration considers the risks related to security issues, excluding the risks not applicable for private permissioned blockchains, and guides the designer towards the proper countermeasures. To support the configuration we performed a risk assessment [23]. The desired level of security is classified as: *medium,* and *high*, and for each level, countermeasures related to the specific criterion analyzed are proposed. The medium level addresses only the issues marked as M and H during the assessment, thus corresponds to a subset of the countermeasures proposed with high level, which addresses also less risky issues (L).

  In the context of private blockchains, nodes have to be enrolled and authenticated in order to take part in the network. The access control mechanism eliminates some security issues, e.g., the risk of 51% attack because the validators are known, but may reinstate some other, for instance related to the partial reintroduction of an intermediary.

  The *client-side security* criterion is the first to be addressed, and the methodology lists a collection of attacks and related specific countermeasures, in this case applicable also for private blockchains [12]. For example, an attack existing in both *medium* and *high level security* classes is as wallet theft, where adversaries may steal or destroy the private key of a user. Related possible countermeasures which are suggested by the methodology are: threshold signature based two-factor security [18], and Password-Protected Secret Sharing (PPSS) [19].

  Then, the flow continues with the analysis of *smart contract security* criterion. In fact, based on the requirement of smart contract support, specific countermeasures exist. If this is the case, the *bug analysis* decision block proposes Oyente or similar approaches; while a solution like Town Crier is suggested if there is a need of *external communication* [21].

  Other criteria addressed are [12]: *cryptocurrency*, as many of the security issues analyzed in literature are related to its existence in the blockchain, *network attacks,* such as DDoS, and *mining attacks* (not applicable in the case of private blockchains as specific for PoW consensus algorithm).
- *Privacy and Anonymity.* At this stage the approach is equivalent to the one at security measures stage, with the difference that now, obviously, the considered issues are related to privacy or anonymity. If read permissions are restricted, as in the case of private permissioned blockchains, the level of privacy provided is higher by-design than for the other categories of blockchains.

  The methodology first considers the *privacy* criterion, and for public blockchains proposes techniques for enhancing it [12], e.g., the usage of peer-to-peer mixing protocols like CoinJoin. Then, the *unlinkability* criterion is considered, and the designer is pointed towards techniques specifically designed for this purpose, i.e., ValueShaffle based on confidential transactions mixing approach.
- *Data Computation and Storage.* This is the last stage of the configuration phase and its role is to guide the designer in addressing the trade-off between on-chain and off-chain computation and storage. This is relevant because the space available on the chain for computation and storage is limited. Moreover, the cost of using public blockchains can be really considerable. Based on the blockchain category chosen in the previous phase, the methodology proposes different solutions. The criteria are *smart contract support, Turing-completeness* and *cost-efficiency*.

  First, the *smart contract support* is checked with the corresponding decision block. If this is the case, different solutions are suggested, as it creates the possibility to store data directly in smart contract variables.

Then, if *Turing-completeness* is available for the selected blockchain category and expressed as a requirement for the project, data is suggested to be computed within the smart contract.

Finally, the *cost-efficiency* criterion is addressed and the designer has to specify whether this is a key requirement or not. As an example, giving priority to cost-efficiency, the advice is to store data off-chain: which can be a private cloud on proprietary infrastructure, in case of private blockchain, while for public blockchain categories a third party storage as IPFS or Storj, is the recommended solution.

At this point, the methodology terminates and the flow diagram reaches its final state. As a result, the designer should have been assisted throughout the decision making process, being able to make choices regarding the configuration of the blockchain compliantly with its project requirements. If not assisted, those choices may affect the provision of fundamental properties typically offered by blockchains, as well as raise issues related to non-functional properties like security, privacy, or scalability.

III. Related Works

In [4], the authors analyze the properties of blockchain categories in comparison with traditional databases. They introduce a flow chart to determine if a blockchain is needed and constitutes the appropriate solution to a problem; it clearly inspired our work, especially the first phase of our methodology. Our approach introduces more criteria in the requirements analysis step and, moreover, does not address only the general choice of the blockchain category, but goes through it configuration.

Many other works which analyze blockchains fundamentals already exist. Between them, we especially reviewed articles focused on differences between blockchain categories [3], consensus algorithms [5], [6], [7], [13], [14] and security and privacy issues with respective countermeasures [15], [21], [22]. In particular, [3] proposes a conceptual model for assisting the decision making process during the design of a system potentially based on blockchains. While their taxonomy captures major architectural characteristics of blockchains, their conceptual model, as explained by the authors themselves, is only indicative. Instead, our methodology and the diagrams constituting it, address the design process thoroughly and in a more comprehensive way.

IV. Conclusions and Future Works

Blockchain is considered a disruptive techonology and a possible revolution for businesses. However, many blockchain skeptics do exist..Our opinion is that it is actually a promising technology, but not a panacea, and if used when not necessary or not properly configured becomes costly, unconvenient or even useless. Selecting and configuring the most suitable blockchain autonomously is not easy, and delegating these tasks to a service provider (with BaaS), reintroduces centralization. In this paper we proposed a requirements-driven methodology to first determine whether a blockchain is or not the most appropriate solution to address a specific problem. Then, it aims to facilitate the selection of the proper category of blockchain. Finally, it guides the configuration of the selected ledger, considering various criteria related to it. As a future direction, we plan to investigate the usage of an ontology based language to better describe relations among requirements and blockchain categories, together with information about their functional and non-functional properties. Finally, we think that the proposed methodology has to be automatized in order to really constitute a valid and useful instrument.

Acknowledgments

This work has been partially supported by the projects TOSCA-FI (FAR-FAS 2014), and SISTER (POR FESR Toscana 2014-2020) both funded by the Tuscany Region.

References

[1] S. Nakamoto. Bitcoin: A Peer-to-Peer electronic cash system, 2008.

[2] M. Swan, *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.

[3] X. Xu, et al. A taxonomy of blockchain-based systems for architecture design. In *Software Architecture (ICSA), 2017 IEEE Int. Conf. on* (pp. 243-252). IEEE.

[4] K. Wüst, and A. Gervais. Do you need a Blockchain?. *IACR Cryptology ePrint Archive*, *2017*, 375.

[5] Z. Zheng et al. Blockchain challenges and opportunities: A survey. *Work Pap.–2016*.

[6] A. Baliga. *Understanding blockchain consensus models*. Tech. rep. 2017

[7] Z. Zheng, et al.. An overview of blockchain technology: Architecture, consensus, and future trends. In *Big Data (BigData Congress), 2017 IEEE Int. Congress on*

[8] E. Schiavone, et al. Continuous Biometric Verification for Non-Repudiation of Remote Services. In *Proc.of the 12th Int. Conf. on Availability Reliability and Security, ARES 2017* (p. 4). ACM.

[9] C. K. Panetta. Top trends in the Gartner Hype Cycle for emerging technologies. 2017.

[10] M. Scherer. Performance and Scalability of Blockchain Networks and Smart Contracts. 2017.

[11] M. Vukolić,. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *Int. Workshop iNetSec*. Springer. 2015.

[12] http://rcl.dimai.unifi.it/~enrico/blockchoice/

[13] C. Cachin and M. Vukolic. *Blockchain Consensus Protocol in the Wild* – IBM Research Zurich, 2017.

[14] S. Bano et al., *SoK: Consensus in the Age of Blockchains,* The Alan Turing Institute, ArXiv:1711.03936v2, 2017.

[15] M. Conti et al., *A Survey on Security and Privacy Issues of Bitcoin,* arXiv:1706009116v3, 2017.

[16] J. J. Jiang. How much does trust cost? Analysis of the consensus mechanism of distributed ledger technology and use-cases in securitization – M.Sc., in Management of Technology at the MIT, 2017 .

[17] G. O. Karame et al., "Double-spending fast payments in bitcoin," in Proc. of the 2012 ACM Conf. on CCS,'12. ACM, 2012, pp. 906–917.

[18] R. Gennaro et al., "Threshold-optimal dsa/ecdsa signatures and an application to bitcoin wallet security," in Int. Conf., ACNS 2016..

[19] S. Jarecki, et al., "Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online)," in 2016 IEEE EuroS P, pp. 276–291.

[20] NIST. Guide for Conducting Risk Assessments. NIST SP-800-30, Rev.1, 2012.

[21] X. Li et al.. A survey on the security of blockchain systems. *Future Generation Computer Systems*. 2017

[22] I. C Lin, and T. C. Liao. A Survey of Blockchain Security Issues and Challenges. *IJ Network Security*, *19*(5), 653-659. 2017.

[23] G. Morganti. Risk Assessment of Blockchain Technology. Bachelor's Thesis, Computer Science, University of Florence, 2017.