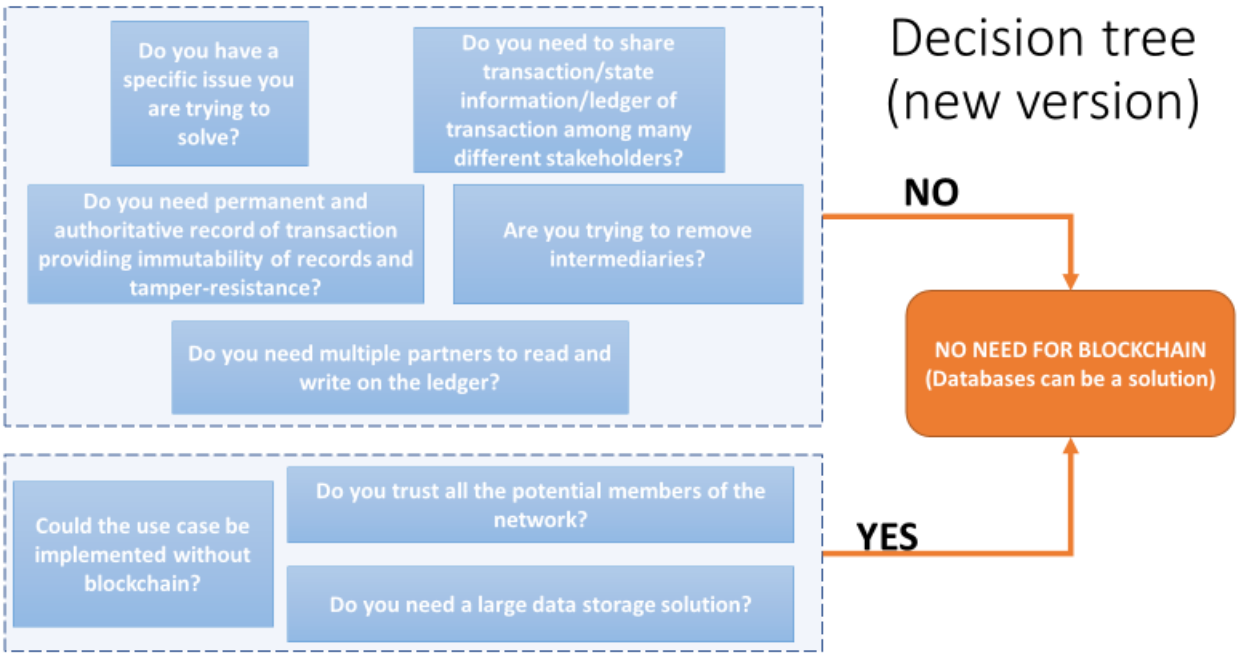
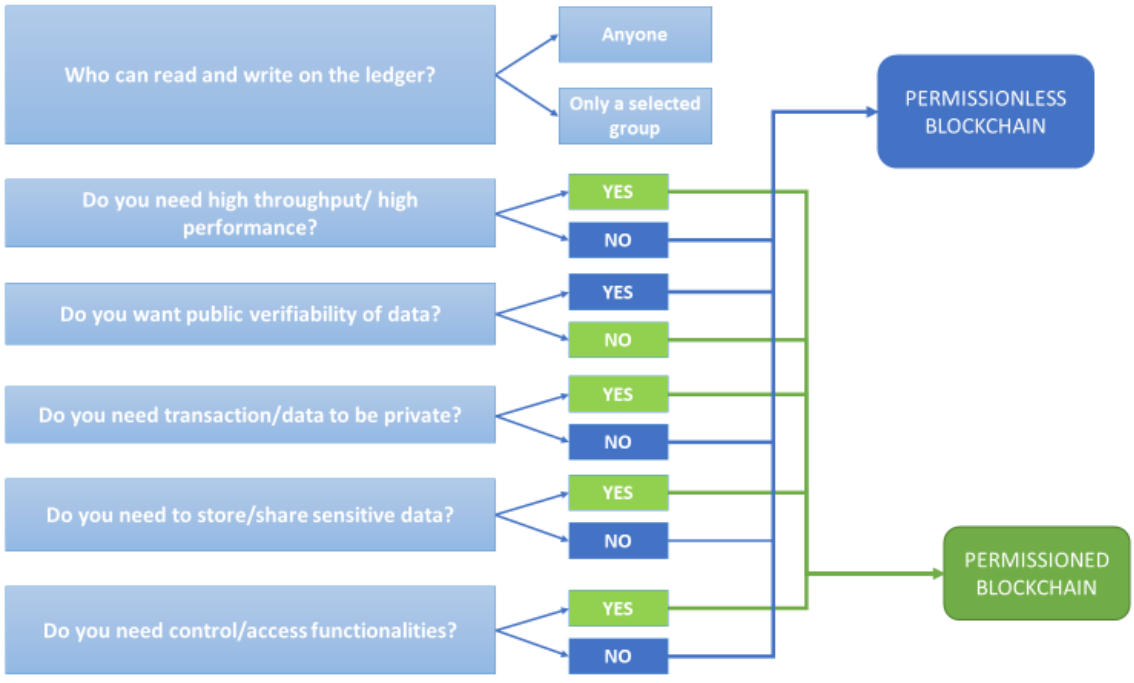
Blockchain adoption evaluation framework

The framework is the conclusive part of the Blockchain applications for health data management report, and is meant to serve as a guidance for establishing a) if a given use case can benefit from the adoption of blockchain technology; b) how to appropriately design a blockchain-based solution in non-technical fashion, in order to facilitate the discussion with blockchain-based solutions providers/developers.

The following pages will provide some key criteria and aspects to be taken into account when considering the adoption of blockchain-based solutions. The following step is to go through the whole framework for a number of specific use cases, in order to end up with a set of basic high-level designs for each of them, useful to discuss further the applicability of blockchain in specific real-world scenario and to transfer the technology into operational environments.

A general decision-making tree:





Basic framework for designing the solution

Once the first fundamental step is completed, we need to move forward in better defining the blockchain-based initiative, going through the following steps:

1) Make the case for blockchain innovation and clearly identify the use case

- Determining objectives, set expectations among different stakeholders
- Show how blockchain help achieving results not achievable through other means – also providing appropriate trade-off analysis of alternative solutions (including funding needs, capabilities, needs for external support)
- We can establish upfront the expected return of the initiative, both in economic terms (including efficiency gain) and in terms of improved technical capabilities (even without immediate business outcomes)
- **Identify the use case:** focusing on use case tied to the specific activities of your organisation and relevant business model and not doable without blockchain

As mentioned in the general decision tree model, the use case is particularly important, also to avoid the usual feeling (present effect among different stakeholders in different industries) of the blockchain a "solution in search for a problem", which – if not appropriately counterbalanced – can eventually lead to a lack of engagement of key partners and ultimately to the failure of the initiative.

Therefore, it is important to start the initiative by identifying the problem/opportunity we are seeking to address through the solution, also listening to different points of view from different stakeholders, seeking for a consensus on the selected use case in terms of expected outcomes, focusing – when possible – on tangible outcomes to be reached in the short/medium-term.

At this stage is also important to establish benchmarks for future references, by defining some key metrics to assess the results of the new blockchain-based system in comparison with existing systems/workflows. For this reason, is of paramount importance – once clearly identified the use case - to analyse in details what is the current workflow for that specific operations, and identify the relevant KPI, which need to be measured for the existing solution/workflow. This will serve as ground truth for assessing the performance of the blockchain-based system, providing evidence of its efficacy or further guidance for improvement or re-design.

Make key design decisions

Once the use case and relevant KPI are established, it will be possible, and needed, before even starting with any technical specification, to make some fundamental decision on the high-level design of the solutions. The following questions needs to be answered once the use case is understood, also on the basis of a comparative analysis with existing solutions:

- Nature of the process: are predictable/repeatable/automatable processes a key component of the use case?
- How the reconciliation process is currently performed within the network and by whom? Who plays the role of controlling data and authorising transactions among members of the network?

Understand what technology do you need (more in details):

- What are the needs in terms of speed, programmability and features of the systems?
- Do we need to process an elevate number of transactions?
- How often?
- Do we need to include specific business logic to be executed/automated within the process?
- Do we need to enforce specific rules/permission settings?
- Do you need to restrict participation to specific parties?

Understand the high-level design of the system:

- Do we need specific parties to perform specific functions?
- Who needs to be able to write transactions?
- Who can validate them?
- Who can read them?
- What are the technical requirements needed at each node (data storage/calculation)?

Understand usability and translation in the operational environments:

- What is the intended user experience and features available for the end-users?
- What is the current workflow and associated behaviours and relationship among stakeholders?
- How the new solution will impact the established behaviours?
- What are the requirements in terms of integration with legacy systems?

Answering these questions will help us in establishing key design elements such us:

- **Who will be involved;**
- **What will be shared within the network**;
- **What performances and scalability needs we are seeking for;**

Decide who will participate and how

Given the fact that a blockchain system is always based on collaboration among different stakeholders (see above) and it put in place a network for those parties to cooperate efficiently, it is important to make upfront the choice of who to involve and which data to share within this network. At the same time, this choice is fundamental as the correct implementation of the use case and relevant workflow depend on the involved stakeholders, and their respective responsibilities and functions, and how the network of stakeholders is organised (also in terms of rules, responsibilities and roles of each partner, and relationships between them).

This will allow to appropriately leverage the higher level of transparency provided by the system, as well as of the multi-party cooperation features that will be available. Such a decision will also lay the foundation on the choice of the preferred architecture for the solution, establishing whether we need a permission-less or permissioned blockchain, and how this should be put in place. It is important here to remember that permissioned blockchain allows: 1) identify and authorise participants upfront; 2) regulate participation to the network (also in force of existing legal agreement and relevant liabilities); 3) regulate the data flow and the way transactions are completed, also enforcing and automating specific business logic; 4) selectively decide with which partner share which information. On the contrary, permission-less architecture are more suitable when we need maximum level of transparency, publicity and auditability of the data, but we are not essentially seeking to regulate/operationalise specific contractual agreements that involve private information.

What will be shared - What data are we going to record on the ledger

For making an appropriate choice on that regard, it is important to be mindful of the following key elements: 1) as the blockchain is – by definition – a shared ledger, and that this entails that each participant to the network owns a copy of the full ledger, we need to be careful when sharing on such a system confidential or private information (as well as personal sensitive data), establishing clear rule for access and control, while keeping in mind that actual recording of sensitive private data should not happen at all; 2) the blockchain should not be considered as a substitute of normal database and is not well suited for storing large datasets (which could introduce latency or cause performance issues). On the contrary, blockchains are to be considered – once again - ledgers primarily intended as shared recordkeeping/event log system, suitable to host small and simple kind of datasets.

As a general rule, we should consider to be suitable for being recorder “on-chain” transactional data/metadata (as well as pointers and hashes), while “off-chain” should remain both large data and personal/sensitive data.

This means that – with the blockchain playing the role of permanent and shared log of transactions information (a sort of reference system or index) - it becomes also very important to both establish which data – essential to the workflow – we want to capture into the ledger and how – on the other side – we should establish appropriate communication between the index (data stored in the blockchain, which can include a set of metadata such as where the data is, who is the owner, when has been created, who can access it, what are the access permission settings, when it has been accessed, etc.) and the actual underlying data, stored in dedicated databases.

This is particularly relevant for medical records, which need to be appropriately stored in hospitals and other clinical centres, where they are also usually generated. In the blockchain, it will be possible to include a timestamp and the hash of the data, the pointer to it, as well as link to the location and associate to the data cryptographic keys so that access is only granted for those who are in possession of the appropriate authorisations (patients, clinicians…).

It is also important to establish appropriate workflow for data input on the blockchain, in order to avoid errors: in fact, keeping in mind that blockchain transactions are immutable and permanent, data cannot be deleted once created on the ledger. As a consequence, even though is possible to correct those mistakes by appending new (and correct) transactions to the ledger, is also important to minimise incorrect data input, for avoiding additional costs and time/efficiency loss.

Considerations on performance and scalability

A clear definition of the use case, the network and the kind of data will be shared through the network, will also allow to establish some key requirements in terms of technical performances.

In terms of performances, we want to consider number of transactions processed per second, which can depend on a variety of factor including the number of nodes as well as the amount of data and transactions to be processed. Permissioned blockchain have better performances than permission-less blockchain, and are usually both faster and less technically demanding. At the same time, this comes to a cost which consists on the need of trust specific nodes to perform transactions ordering, validation, and synchronisation. We can affect performance by design when deciding what data to be shared, the role of the participants, the kind of network we are going to establish, and so forth.

Similar considerations can be made In terms of scalability: the less demanding the kind of data and size of the data to be shared and the transactions to be recorded, the more scalable the solution will be, always keeping in mind the need of replicating data in each node and validate node at a fairly fast pace.

Once again, the initial design can play an important role in ensuring long-term sustainability, avoiding latencies which make the whole system less usable in operational environments.