SOFIE - Secure Open Federation for Internet Everywhere

779984

DELIVERABLE D2.1

State of the Art Report

Project title	SOFIE – Secure Open Federation for Internet Everywhere
Contract Number	H2020-IOT-2017-3 – 779984
Duration	1.1.2018 – 31.12.2020
Date of preparation	29.06.2018
Authors (alphabetically)	Priit Anton (Guardtime AS), Tommi Elo (AALTO), Nikos Fotiou (AUEB-RC), Mikael Jaatinen (LMF), Yki Kortesniemi (AALTO), Petri Laari (LMF), Dmitrij Lagutin (AALTO), Nelly Lelligou (SYN), Pekka Nikander (AALTO), Yannis Oikonomidis (SYN), Santeri Paavolainen (AALTO), George C. Polyzos (AUEB-RC), Vasilios A. Siris (AUEB-RC), Spyros Voulgaris (AUEB-RC), George Xylomenos (AUEB-RC)
Responsible person	Spyros Voulgaris (AUEB-RC), voulgaris@aueb.gr
Target Dissemination Level	Public
Status of the Document	Completed
Version	1.00
Project web-site	https://www.sofie-iot.eu/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 779984.

Table of Contents

1. Introduction

This document provides an overview of the State of the Art in blockchains and Distributed Ledger Technologies (DLTs) more generally, with the intent of employing them to support trust and realize automatic operations in the Internet of Things (IoT) through the use of smart contracts (realized over blockchains). For this reason, IoT characteristics and architectures are also briefly reviewed and some IoT systems that have embraced DLTs are discussed, together with a presentation of security and privacy aspects.

Distributed Ledger Technologies (DLTs) provide a tamper-proof database where trust is supported in a distributed manner across a set of computers, rather than centrally through a single or group of institutions. Blockchains are one type of distributed ledgers, where data records are grouped into blocks that are linked through cryptographic hashes, thus forming a chain of blocks. The main elements of a blockchain are the consensus mechanism and the programming language; the latter can range from simple scripting to more powerful smart contract programming languages that support Turing-complete computation, can maintain a state, and support the interaction with other contracts. Blockchain systems can belong to two broad categories that differ in the policy that defines which nodes can participate in the blockchain's distributed network and the roles that they can perform. The consensus mechanism is central in providing the guarantees and properties of the DLT and is still a topic of active research. Noteworthy blockchain systems discussed in Section 2 include Bitcoin, Ethereum, Cardano, IOTA, which are public (open) blockchains, and Hyperledger Fabric, MultiChain, and Corda, which are permissioned (private) blockchains.

Interledger approaches and mechanisms, which are discussed in Section 2, have been developed to connect different DLTs and payment networks because no one technology seems to prevail and new ones continue being successfully introduced. They differ in whether they provide support for trading value between two blockchains, in which case the total amount of value in each blockchain remains the same, or whether they support the transfer of value among blockchains. Moreover, different interledger approaches have varying support for transferring information across different blockchains. Atomic cross-chain transactions form an important subclass of interledger approaches, and are based on more basic mechanisms, namely hashlocks and timelocks. Atomic cross-chain transactions allow trading of value across blockchains between two parties that do not trust each other, without requiring the presence of a trusted third party. Sidechains enable the transfer of value from one blockchain to another. The main motivation for implementing sidechains is to achieve higher performance, such as lower block confirmation times, lower transaction costs, or to support for more flexible smart contracts, which are not supported by the main chain. An important advantage of sidechains is that they can offload the main chain from handling all transactions, hence can enhance the scalability of blockchains. One approach for implementing sidechains is based on the federated peg, which relies on semi-trusted functionaries that achieve agreement through Byzantine consensus. Another approach for implementing sidechains, on which Etherium's Plasma proposal is based, is to create a hierarchical tree of sidechains. Each sidechain can be governed by its own set of rules and constraints, while full security is provided only by the root chain. Bridging approaches typically support the transfer of data, in addition to the transfer of value between blockchains. Unlike bridging, ledger-of-ledger approaches rely on a super ledger for interconnecting sidechains or other ledgers. Proposals such as the Lightning Network and Raiden seek to increase the scalability of blockchains by performing specific transactions among two parties off-chain; moreover, they support transferring payments across a network of payment channels between entity pairs.

Payments across a network of DLT systems is also the focus of the Inter-ledger Protocol (ILP), which aims to provide a coherent, standards-based payment infrastructure. Finally, blockchains do not allow contracts, including smart contracts, to directly interface with the outside world. This interfacing of blockchains with the outside world is provided by oracles, that can obtain data from external sources or call external APIs, which we also discuss in Section 2.

Currently, there is a large number of IoT platforms that are proprietary and either fully or partially closed. There have been various attempts for defining or describing IoT architectures and systems using a layered and modular approach for facilitating interoperability. We discuss these efforts in Section 3. There are also proposals that aim to achieve interoperability at a higher semantic layer, compared to technical and syntactic interoperability, in order to enable different entities to exchange information, data, and knowledge in a meaningful way. Some IoT systems have already adopted blockchain technology. We discuss such systems in Section 4, focusing in particular on how they support IoT data and transactions. These systems include both industrial solutions, but also blockchain-community originating proposals that specifically target IoT data handling.

In Section 5 we discuss privacy related issues, especially those that arise when a large amount of data needs to be handled from different sources. Decentralized identity solutions are necessary for large distributed systems with key evaluation criteria including the trust model, degree of interoperability, and cost. We conclude Section 5 with a discussion of vulnerabilities that target specifically the consensus mechanism of blockchains.

Concluding, in Section 6, we provide a summary of this document on DLTs and the IoT, with conclusions and an outlook.

Finally, as this is a very new and rapidly-evolving area, a large fraction of information comes from projects' web sites, blogs, and various forms of online material. In order to include links to this material, yet keep an explicit distinction to refereed publications and thoroughly-written whitepapers, we include the former as footnotes, while we reserve classic bibliography-style references for the latter.

2. Distributed Ledger Technology

Distributed Ledger Technologies (DLT) are a relatively new set of technologies, with no more than a decade of history. Despite their recent conception, DLTs have attracted tremendous attention in a number of diverse fields, most notably in computer science, cryptography, finance, economy, civil law, healthcare, rights management, real estate, auctions, gambling, and generally all industries for which *provable trust*, *accountability*, and/or *transparency* play key business roles. Some people claim that DLTs will bring to asset management as big of a revolution as the Internet brought to communication.

In the following sections we give a conceptual overview of DLTs, identify and explain their fundamental building blocks, and elaborate on the most notable implementations.

2.1 Conceptual Overview

In a nutshell, a DLT is a giant *append-only* log file replicated across a set of participating nodes. When a new log entry is to be appended, participating nodes *vote* on whether it complies with the DLT's rules, and come to an agreement regarding the *admission* and the *order* of new log entries. This agreement is known as *consensus*, and the protocol ensuring it is called the *consensus protocol* (of the particular DLT).

What makes DLTs a disruptive technology is that they offer, for the first time ever, a tamper-proof database where trust emerges through the collaboration of a set of computers, rather than through an institution or organisation that imposes trust from the external world onto the system.

Blockchains, on the other hand, are just one type of distributed ledgers. They have become so hyped that many people perceive the two terms as synonyms. Although blockchains *are* distributed ledgers, not all distributed ledgers are based on blockchain technology.

In the case of blockchains, data records (log entries) are grouped into blocks. The very first block, known as the *genesis block*, is a special block known to everyone. Each other block links to its previous block by incorporating a cryptographic hash of the previous block's contents, creating a *chain* of blocks. In case a block gets tampered with, the hash stored in its successor will not match its (updated) content anymore, effectively breaking the link. That is, tampering with a block deprives it of its successors, rendering the altered block as the last block of this (altered) blockchain. Given the blockchain policy that given two blockchains that have the same genesis block and have different lengths, the longest one is considered valid and any shorter ones as orphans, the tampered with shorter blockchain will be instantly ignored by all correctly behaving nodes, rendering tampering with the block meaningless. As creating new blocks is made deliberately hard (see below), an attacker wanting to tamper with a block will have a very hard time in trying to produce a longer chain than the so-far longest chain.

2.2 Blockchain Elements

Each blockchain is designed as a combination of certain elements. Identifying and understanding these elements is essential to the comprehension of blockchains and their innerworkings. The main elements determining the operation of a blockchain are the following:

- Consensus Protocol
- Smart Contracts (if any)
- Permission Model

2.2.1 Consensus Protocols

DLTs' most innovative breakthrough is the creation of trust based on a large number of generally untrusted nodes. This is achieved through sophisticated consensus mechanisms, which, as outlined above, are central to the operation of DLTs. A number of DLT consensus mechanisms have been devised, having significant differences, yet a common goal: Enabling the entire network to decide unanimously and inadvertently on which records to include next, and in which order, into the DLT. The protocol constitutes, essentially, a *voting mechanism* used for filtering and ordering the records that are stored into the DLT.

In the following subsections we review the most common consensus mechanisms.

2.2.1.1 Proof-of-Work (PoW)

Proof-of-Work (PoW), or Nakamoto consensus, is the first and most popular consensus mechanism for DLTs to date, primarily known through its use in Bitcoin. It is based on the combination of (i) the "longest chain wins" principle, (ii) a computationally hard problem to build (or *mine*) the next block, and (iii) a reward for mining a new block into the chain. The most often used computationally hard problem is *reverse hashing*, having an inherently only brute force (like) solution. Unfortunately, reverse hashing has no real-world scientific or societal causes or benefits, beyond just achieving the consensus per se. Its sole purpose is to incentivize participating nodes (known as *miners*) to devote their resources to the chain that has grown to be the longest, i.e., the winning chain, thereby effectively achieving a universal, decentralized consensus on which specific chain version is valid.

Unless a single entity controls more than 50% of the world's mining capacity, it is in each miner's interest to abide by the rules. Unfortunately, with the increasing popularity of massive mining pools, the scenario of aggregating more than half of the world's hash power under a single entity is no longer unlikely. Thus, the danger of what is known as the *51% attack* cannot be entirely ruled out.

PoW's main drawback is, however, the exorbitant amount of energy it requires. For example, Bitcoin's mining power is expected to surpass the entire power consumption of countries like Ireland or Denmark by 2020[1]. This puts the long term viability of PoW-based consensus at stake, notably in terms of their carbon footprint and their effect on global ecology. It has, thus, fueled strong research efforts in devising alternative consensus mechanisms, some of which are described below.

[1] *Bitcoin Could Consume as Much Electricity as Denmark by 2020*. URL: https://motherboard.vice.com/en_us/article/aek3za/bitcoin-could-consume-as-much-electricity-as-denmark-by-2020

2.2.1.2 Proof-of-Stake (PoS)

Proof-of-Stake (PoS) is an alternative consensus mechanism, addressing the two main deficiencies of PoW, namely the huge waste of energy and the 51% attack. To get the terminology straight, in PoS terms miners are called *validators* and the act of mining a block is called *minting* or *forging* a block. Contrary to PoW miners, PoS validators are *not* in direct competition against each other in terms of computational power for minting the next block. Instead, the network elects which validator should mint the next block, thus preventing the wasteful process of PoW mining.

The selection of the next block's validator is not random. To become a validator, a node has to deposit something[2] as a security deposit, known as its *stake*. This stake remains reserved and is returned to its owner, along with all the transaction fees of the minted block, only after some time has passed. In case the validator has approved a fraudulent transaction, it loses all transaction fees *and* its originally pledged stake. This creates a profit-based motivation to follow the protocol.

The rationale behind electing a validator proportionally to its stake is straightforward: The more you risk losing, the less likely you are to cheat. This, however, encourages a rich-gets-richer pattern, where the richest nodes are repetitively appointed validators, collecting all transaction fees, and getting even richer. To alleviate this problem, the use of *coin age* has been devised: A coin's power in increasing a node's chances to be appointed a validator is dependent on the time since the coin was last used as stake. That is, coins that were recently used as stake are of no use in being used again, until a certain time period has passed.

Although PoS significantly lowers its ecological impact as a consensus algorithm compared to PoW, it is not without its drawbacks. The operational cost of a PoS miner is negligible, thus enabling a single miner to participate in multiple PoS blockchains using the same machine. As a result, a new attack vector called *Nothing at Stake* is feasible.

In PoW systems, when a blockchain fork occurs, miners are indirectly forced to choose a fork they will continue supporting as they will not be able to participate in multiple PoW blockchains due to the computational capacity required. In PoS systems, this protection is nullified and, as such, nothing prevents a validator from supporting any number or even all the forks of a PoS blockchain simultaneously, as his profits will increase regardless of which fork becomes dominant.

Additionally, a *tragedy of commons* also facilitates the above modus operandi. If the stake of a miner who wishes to initiate a fork in a PoS system is sufficiently large, other miners will join his fork in fear of losing their stake and being unable to overcome and prevent the fork.

As this problem exists inherently in the absence of a restrictive resource for operating nodes in a PoS scheme, its solution remains the subject of academic research initiatives. One such initiative is called *Ouroboros*, led by the Universities of Edinburgh, Connecticut, and Aarhus, and a private organisation named IOHK. They explore a solution based on the forking mechanism of a blockchain.

There are a number of systems using various flavours of PoS, including Cardano, Lisk, BlackCoin, Peercoin, and Nxt. Ethereum has announced the intention to switch to PoS.

[2] Essentially a pledge or collateral of something considered valuable by the community.

2.2.1.3 Proof-of-Authority (PoA)

Proof-of-Authority (PoA) delegates transaction validation and block creation to a certain set of *authorized nodes* who are acting as the administrators of the system. This paradigm best fits private blockchains, as it exhibits a centralized operation: denoting who are eligible as administrators (and who are not). Transaction throughput can be high, and blockchain parameters can be fine-tuned to the specific needs of the private networks they serve. However, trust does not emerge from the inherent dynamics of stakeholders, but is rather "outsourced" to the sysadmins guaranteeing the secure and flawless operation of a big enough fraction of the authorized nodes.

Most of the PoA-based consensus mechanisms are based on the so-called Byzantine consensus protocols. These protocols stem from the seminal work of Lamport, Shostak, and Pease in the early 1980s [PSL80], [LSP82] and later works of Castro and Liskov [C+99]. While many Byzantine protocols are being employed elsewhere, e.g., in many cloud databases, their use in DLTs is mainly differentiated through the DLTs being append-only databases.

2.2.1.4 Proof-of-Elapsed Time (PoET)

Proof or Elapsed Time (PoET) is a consensus mechanism introduced by Intel, making use of their CPUs' Software Guard Extensions (SGX) feature, enabling processors to run trusted code that cannot be tampered with. The logic behind PoET is simple. Each participating node waits for a *random period*, and the first node to finish waiting becomes the validator for the next block. Essentially this is a basic leader election protocol, which uses negligible CPU power to execute. Clearly, its correctness depends on nodes being honest with respect to waiting for a *really randomly chosen* time period. However, this is guaranteed through code that is trusted based on Intel's SGX feature.

One example of use of PoET is in Hyperledger Sawtooth[3], developed primarily by Intel.

2.2.1.5 Hybrid approaches

In addition to mechanisms that are pure PoW, PoS, PoA, or PoET, we are starting to see more and more mechanisms that mix some features of these together, e.g., into systems that require both staking a pledge and performing some work. However, as these mechanisms are still being developed and, as far as we can see, none of them are considered well established by the larger community, they mostly fall beyond the scope of this document. However, some examples of such hybrid mechanisms can be found in, e.g., some of the so-called ledger-of-ledger approaches, which often seem to combine PoW and PoS mechanisms.

2.2.2 Smart Contracts

Smart Contracts bring another groundbreaking innovation to the DLT world. Rather than using DLTs' decentralized trust model for offering just an immutable decentralized append-only data store, they exploit the mechanisms to provide a tamper-proof decentralized "world computer".

[3] *Hyperledger Sawtooth*. URL: https://www.hyperledger.org/projects/sawtooth

Through smart contracts, DLTs are promoted from special-purpose tools serving a single application (e.g., Bitcoin) to general-purpose platforms, allowing developers to deploy and execute custom code that may implement arbitrary application logic.

A smart contract is essentially a program running on a ledger, maintaining some internal state, and updating this state through transactions. To allow transactions to be made, a smart contract exports an API consisting of one or more functions. Read-only functions, i.e., functions that only read the smart contract's program state without changing it, are executed locally on the node that calls them. Calling such functions does not create new transactions to the underlying ledger. Read-write functions, i.e., functions that update the smart contract's state, are executed on all nodes running the ledger, and require new transactions to be entered into the ledger. More specifically, all nodes that validate a block now, or any time into the future, have to execute all smart contract function calls included in that block, to check whether the resulting overall state matches with the state recorded into the block's hash.

Even more specifically, smart contracts gain their power through the following fundamental features:

- **State storage:** It is possible for a smart contract to store its own state into a ledger. This makes it possible for a smart contract to store arbitrary data, not just data specific to Bitcoin or some another application.
- **Turing-complete computation:** Many smart contracts enjoy a Turing-complete computation environment. This allows them to implement arbitrary logic for any type of custom ledger applications.
- **Interaction with other contracts:** A smart contract can call functions of other contracts to interact with them.
- **Input from the external world:** A smart contract can get input from external sources, such as what is the current cryptocurrency value in fiat money (euro, dollar, etc.). This can be done indirectly through interaction with other contracts, which, in turn, get updated by (trusted) external organizations pushing frequent updates to them. These so-called oracles are discussed further in Section 2.5.
- **Input from the ledger itself:** A smart contract is generally allowed to read and use any value in the ledger, such as last block's hash value. For example, the block hash value can serve as a deterministic pseudo-random number generator.

A platform supporting smart contracts essentially allows arbitrary applications to be developed, and to benefit from the same level of trust and transaction irreversibility enforced by a larger community of participating nodes.

The possibilities opening up with smart contracts are unlimited. Some examples can help to better understand possible scenarios:

- **Multisignature wallets:** A group of two or more people may set up a shared wallet, configuring it in such a way that only the *combination* of all signatures allows the withdrawal of arbitrary amount, but a single signature *alone* may spend up to 1% of the stored amount, on a daily bases.

- **Insurance:** An insurance company could issue insurances through smart contracts, e.g., letting a farmer insure his crop against bad weather conditions. The smart contract would require the farmer to pay his insurance fees up to a given deadline, and would cover that farmer's potential loss based on input from a trusted weather-logging smart contract, which, in turn, would get its input from an externally trusted IoT system (an "oracle").
- **Auctions:** People could bid on an asset in a system without a middle man, in such a way that for every bid it is verified that the owner indeed has that amount, and by reserving that amount until either a higher bid is made or a timeout elapses and the purchase is completed.
- **Prediction markets or betting:** Bets could be set on predicting future events that can be verified through trusted feeds (e.g., which team will win a soccer tournament).

2.2.3 Permission model

So far we have covered two aspects of distributed ledgers: the consensus model and smart contracts. The third aspect, the ledgers' policy on which nodes are allowed to act in which roles, places the ledgers into two broad categories: *permissionless* and *permissioned* ones.

In permissionless or open ledgers any computer that has network access may join the ledger, basically taking up any role. That is, it may opt to participate as a validator (miner) to contribute in building consensus, as a verifier (full node) to read and locally verify blocks, or simply as a user and issue new transactions. This model is the most well known one, used in a number of blockchains, including Bitcoin, Ethereum, Litecoin, Monero, Dash, and Dodgecoin.

In permissioned ledgers, a node needs to be authenticated and authorized to take up *certain* roles. For instance, a ledger could restrict the validators to a predefined set of authorized nodes but let any node to locally verify the correctness of the ledger. Other ledgers could also require authentication and authorization to read the ledger, in which case they essentially become private ledgers.

There is a considerable variation in the permission model among ledgers. Furthermore, there is an interplay between the permission model and the consensus protocols: the PoA consensus model is only possible in permissioned ledgers.

2.3 Noteworthy Blockchain Implementations

The number of reported cryptocurrencies has exceeded the whopping number of 1600, as of June 2018. Clearly, a large fraction of them are identical to each other from a technical point of view. Hence, there would be no point in even listing them all. Instead of that, we collected a number of well known (but different) ledgers, i.e., ones that may be considered noteworthy, also for their technical implementation.

In this section, we present a number of the most notable blockchain implementations, and elaborate on their operation.

2.3.1 Public or open DLTs

2.3.1.1 Bitcoin

Bitcoin, introduced by pseudonymous Satoshi Nakamoto in [Nak01], undoubtedly earns the first place in this list, being the seminal blockchain design and implementation, as well as the first succeeding fully decentralized digital currency.

Bitcoin's blockchain is used exclusively for transactions in its own cryptocurrency (called *bitcoin*). That is, there is no support for smart contracts or arbitrary applications. Bitcoin does support some elementary scripting; however, the sole purpose of the scripting is limited to providing flexibility in setting the conditions for spending assets, e.g., requiring either a single user's key, or n users' keys, or k-out-of-n users' keys, etc., to dispose of assets.

Bitcoin is an open-source protocol running on its permissionless blockchain. It employs a Proof-of-Work consensus mechanism, based on double SHA-256 reverse hashing. As an explicit design decision, Bitcoin generates one block per 10 minutes, on average. This is achieved through a *difficulty* parameter for the Proof-of-Work algorithm, which is automatically adjusted every 2600 blocks to counterbalance global hashrate increase (or decrease), by considering how much faster (or slower) than 10 minutes the average generation of the last 2600 blocks (about 18 days) was. The incentivization mechanism to attract miners consists in allowing them to pay themselves a predefined amount of newly generated bitcoins for each block they mine. This is also the only way bitcoins are being generated.

There are significant scalability concerns regarding Bitcoin, as its transaction processing rate of circa 5–7 transactions per second is deemed too low for serving trade at a global scale.

2.3.1.2 Ethereum

Ethereum[4] was the first — and is to date the largest — deployed blockchain to support smart contracts. It was proposed by Vitalik Buterin in 2013 in [But13] (white paper), detailed by Galvin Wood in 2014 in [Woo14] (yellow paper), and deployed in July 2015. It introduced the notion of smart contracts, explained in Section 2.2.2 above.

Ethereum incorporates its own currency, called *ether*. Ether is fundamental for Ethereum's operation in two ways. First, it constitutes the incentive for *validators* (Ethereum term for *miners*) to contribute their resources to the Proof-of-Work algorithm. Second, it is used to regulate the use of the blockchain's resources by charging for its use. More specifically, ether is needed to pay for *gas*, a unit used to measure the computation, storage, and bandwidth cost an operation imposes on the blockchain. To invoke a smart contract function, the caller has to specify how much ether he is paying per gas spent, as well as an upper limit on the gas that can be spent. This way, Ethereum can support a Turing-complete virtual machine, called the *Ethereum Virtual Machine* (*EVM*), without fearing a denial-of-service abuse of the system: any long- or eternal-running function costs money, and will eventually be killed for having run out of gas.

In Ethereum, it has been a design decision to produce a new block every 14–15 seconds on average, with clear user experience benefits stemming from much faster transactions than in Bitcoin. At this block generation rate, there is a non-negligible probability for more than one

[4] *Ethereum*. URL: https://ethereum.org

blocks to be mined in parallel, effectively creating a *fork* until subsequent blocks determine the *winning block* and the *orphan block(s)* among the competing blocks. Additionally, as 14–15 seconds are comparable to the time it takes to disseminate a block to the entire network, receiving a new block a few seconds earlier gives a significant advantage, which would normally favor large pools of nodes that could give higher priority to disseminating new blocks among themselves before spreading them to the rest of the community. To weaken this effect, Ethereum does not only reward blocks that do get accepted into the main chain, but also (valid) orphan/stale blocks that were on the abandoned path of a fork. For this reward to be handed out, a newer block (called *nephew*) should link to a past stale block (called *uncle*) in addition to its direct parent. Then, the uncle block receives 87.5% of a new block reward, and the nephew receives the remaining 12.5% of that reward, as an incentive to include it.

2.3.1.3 Cardano

Cardano[5] is being developed in two layers that separate the ledger of account values from the reason why values are moved from one account to the other. The Cardano Settlement Layer (CSL) acts as the *balance ledger*. It uses a Proof-of-Stake consensus algorithm to generate new blocks and confirm transactions. Moreover, CSL supports sidechains for moving assets from the CSL to the Cardano Computation Layer (CCL) and any other blockchains that support the Cardano KMZ protocol, which is used for efficient Simplified Payment Verification (SPV) proofs [KMZ17]. CCL (Cardano Computation Layer) is the second layer of the Cardano platform. It contains the information on why transactions occur. Because the computation layer is detached from the CSL, different users of the CCL can create different rules when evaluating transactions. The Cardano team is creating a new programming language to use to develop smart contracts on the CCL, which is called Plutus. The CCL also supports Solidity — used in Ethereum smart contracts — for low-assurance applications.

The Cardano ledger will also implement its own proprietary virtual machine, called IELE[6]. The VM is based on the LLVM[7], along with its own unique low-level language. The core difference between IELE and the Ethereum VM is that IELE is a register-based machine with an unbounded number of registers, while EVM is a stack-based machine with a stack limit of 1024. As direct result of IELE's design paradigm, it supports unbounded integers, enabling easier development of secure smart contracts. The rationale behind the development of IELE is the creation of a uniform low-level language that elegantly translates to and from high-level languages, due to its register-based nature.

2.3.1.4 IOTA

IOTA is a rather different type of a distributed ledger, in the sense that it is *not* based on a blockchain structure. Instead of letting miners confirm transactions in blocks, IOTA decentralizes this process even further, requiring users themselves to confirm each other's transactions. More specifically, each new IOTA transaction has to include *links* to at least two past transactions, which it *approves directly*. By linking to them, it also *approves indirectly* all past transactions

[5] *Cardano*. URL: https://whycardano.com
[6] Named after a mythical creature
[7] *Low-Level Virtual Machine (LLVM): a compiler technologies umbrella project*. URL: https://llvm.org

approved (directly or indirectly) by them, all the way to IOTA's genesis transaction. This creates a directed acyclic graph (DAG) linking newer transactions to older ones. Contrary to other distributed ledgers, this process does not involve any transaction fees, as the approval of the transactions is made by the users themselves.

Attempting to approve a transaction that is either invalid in itself, or which has approved (directly or indirectly) some invalid transaction, will result in getting the transaction eventually being neglected, and therefore dropped. This constitutes a strong motivation for performing a thorough check when approving past transactions, essentially delegating the task of banning invalid transactions and maintaining consensus to the users themselves. The more new transactions approve, directly or indirectly, a given transaction T, the higher is the confidence on T being valid. There is no fixed confidence level for considering a transaction definite; it is subject to the risk level each user deems acceptable.

Currently the users are not allowed to pick arbitrary transactions to approve. Instead, a centralized third party, known as the "coordinator", allocates approval tasks to users submitting new transactions. This is considered a temporary measure that is planned to be dropped later on.

A core advantage of IOTA concerns its scalability. As registering a transaction requires checking and approving two other transactions, the IOTA system enjoys an inherent elasticity: The higher the transaction rate, the higher the collective capacity of "approvers". An important issue remains to find the right motivation for ensuring that all pending transactions will be picked for approval, without any of them being left waiting indefinitely. Another unique feature of IOTA is the use of "Winternitz One-Time Signatures", which are safe even for post-quantum usage. On the other hand, the use of this type of signature results in big transactions, in the order of 10KB, far larger than Bitcoin's transactions that have the average size 600 bytes.

IOTA is still in a beta phase and has received some negative criticism; for example, MIT's Media Lab criticizes it for implementing its custom cryptographic solutions[8,9].

2.3.2 Private/Permissioned DLTs

2.3.2.1 Hyperledger Fabric

Hyperledger Fabric [And+18] is one of the first completely permissioned distributed ledger implementations that are designed for enterprise usage. Following the norm of the latest released ledgers, it is a smart contract capable ledger in which a contract is referred to as a *chaincode*. The core feature of Hyperledger Fabric is the modularity its internal operational structure provides. The various services that contribute to its operations are split into secluded containers that are independent of each other, enabling users to hot-swap integral components of the system, such as the consensus mechanism or the virtual environment enabling the execution of chaincode, with new and completely different ones.

This novel modularity enables Fabric to facilitate the operation of chaincode within conventional operating systems. This, in turn, allows developers to create chaincode in general-purpose programming languages, such as Python and Java, with Go being the language of choice for

[8] *Cryptographic Vulnerabilities in IOTA*. URL: https://medium.com/@neha/cryptographic-vulnerabilities-in-iota-9a6a9ddc4367

[9] *IOTA Vulnerability Report: Cryptanalysis of the Curl Hash Function Enabling Practical Signature Forgery Attacks on the IOTA Cryptocurrency*. URL: https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md

Fabric. The only prerequisite of a fully functional chaincode is the deterministic execution of its functions, as well as the conformity of its specification to the Chaincode Interface, as defined by Hyperledger Fabric.

Being a permissioned ledger, Fabric employs a layer of membership delegation through the usage of a *Public Key Infrastructure*. Each entity within the Hyperledger Fabric network is uniquely identified by its cryptographic key-pair, for which a valid certificate is assigned by the *Certificate Authority* (*CA*) operating in the network. Although the CAs are part of the Fabric network, the ledger operators are split into two subcategories, the *Orderers* and the *Peers*. The Orderers are in charge of, as their name implies, ordering the transactions within the blockchain network, resulting in an atomic broadcast of new blocks. Orderers implement the consensus mechanism of the ledger and ensure that the ledger is kept in synchrony among all the Peers. The Peers, on the other hand, simply act as hosts of the ledger and are in charge of maintaining the ledger network by keeping a copy of the ledger and allowing clients to interact with it.

The Hyperledger Fabric network does not contain any form of a cryptocurrency. Instead, all transactions are treated as chaincode invocations within the network. Whether a chaincode invocation is valid and should be included in a new block is defined by the chaincode itself once it is deployed on the network. Specifically, a chaincode may specify a set of signatures that need to be acquired by Peers operating within the network in order for a new transaction to be submitted on the blockchain that will alter the chaincode's state. The Peers that are able to endorse chaincode transactions are called *Endorser Peers* and they provide their endorsement upon successful execution of the chaincode within their respective environment.

The definition of a network in Hyperledger Fabric differs from many other ledgers in that it enables multiple ledgers to operate within the same network, via the segmentation of the network into different *channels*. Each channel retains a separate distributed ledger, which can communicate with the ledgers running on other channels within the same network. This feature enables organisations that operate on the same network to keep private information within their own Peers, by maintaining their own distributed ledger.

2.3.2.2 MultiChain

Multichain[10] is a platform for creation and deployment of private blockchains. It supports both intra- and inter-organizational deployment and it offers managed permissions, rapid deployment, unlimited assets, data streams, customizations, and bitcoin compatibility. It is derived from the Bitcoin Core software but it is not limited to the Bitcoin blockchain. MultiChain aims to provide privacy and control in an easy-to-use package so that the deployment of blockchain technology will no longer be a key obstacle. At the same time, advantages of it being a private blockchain are also eminent; no scalability issues, blockchain only contains transactions relevant to the participants.

Multichain applies integrated management of user permissions to solve mining, privacy and openness problems. By using private/public key cryptography, which enables any message to be signed by a user and not just transactions. The platform is able to restrict blockchain access to a list of permitted users, embedding a permission protocol in the handshaking process that occurs during the connection of two blockchain nodes. The same principle can also be extended to other operations, such as the right to send and/or receive transactions to a set of addresses.

[10] *MultiChain*. URL: https://www.multichain.com/

In that way, the MultiChain platform manages to:

- ensure that the blockchain's activity is only visible to selected participants
- introduce controls and permissions on transactions
- enable mining to take place free of Proof-of-Work and the associated costs in a secure way

The risk of a single participant of a private blockchain to monopolize the mining process is mitigated by placing a constraint on the number of blocks which may be created by the same miner within a given window. In addition, as mentioned above, MultiChain is not limited to a specific blockchain, as it is easy to configure (configuration during uptime is reported as future work) and deploy, not just by specialized developers but also by system administrators. Multichain also supports multicurrency, utilizing the same techniques used in CoinSpark[11], and moves even further by improving on these schemes by integrating support for third party assets directly into the chain's rules.

A main focus of the MultiChain platform is to provide a way for smooth transitions between private blockchains and the bitcoin blockchain in either direction. This is achieved by the following:

- MultiChain is based on a fork of Bitcoin Core, its interface is fully compatible with it, and it can act as a node on the regular bitcoin network.
- It utilizes Bitcoin's protocol, transaction and blockchain architecture, enhancing just the handshaking process.
- It adopts multicurrency and messaging features offered also by the CoinSpark protocol, to enhance bitcoin transactions and to allow applications which use asset tokenization and messaging to move between bitcoin and private blockchains with minimal code changes.

While there does not appear to be any published criticism specifically on Multichain, to us there appears to be at least two problematic areas:

- Any blockchains that use Bitcoin's protocol and block structure are open to 51% attacks by existing Bitcoin miners [Ali+16].
- Using PoW in a private blockchain is not very energy efficient, since in most private blockchains also PoA or PoS can be used.

2.3.2.3 Corda

It served the purpose of exchanging information between nodes and was fast and easy to adapt for building a customer specific PoC.

Corda is a blockchain-inspired distributed ledger by R3, an enterprise software firm owned by banks and other financial institutions. It is meant to serve the purpose of exchanging information

[11] *CoinSpark*. URL: http://coinspark.org

between nodes belonging to different companies. It is designed to be fast and easy to adapt for building a customer specific solutions.

Considering production-ready solution, there appears to be several technological capability gaps in the current open source version:

1) To the current stage there is no backwards compatible software available.
2) Testing the scalability and performance with 500 devices showed that the overhead to the current storage and network was too high in order to be able to receive millions of devices from different users (geographically distributed). It needs more investigation how architecture and data storage upgrade could handle such volumes together with CORDA Notar.
3) There was lack of APIs when using Corda technology.

2.4 Interledger approaches

The term *interledger* denotes a number of different approaches that attempt to establish interoperability among a number of distributed ledgers. The currently proposed interledger approaches vary widely in their purpose and structure, from the W3C Interledger protocol (ILP), which is a TCP/IP-inspired architecture and protocol specification for transferring "money" through Polkadot and Cøsmos, effectively building another (Byzantine PoS) ledger to bridge different ledgers, to atomic cross-chain transactions, which are timelock- and hashlock-based mechanisms (explained below) to either perform *all* or *none* of a coordinated set of transactions in separate ledgers.

A major shared motivation appears to be moving away from the "one chain rules them all" model and increasing flexibility and innovation. Beyond that, there appears to be several quite different motivations for the various interledger approaches:

- **Transferring and/or trading value** between chains. ILP focuses on this, and at least Cøsmos and Polkadot seem to support such value transfer or trade. The essential difference between transfer and trade is in symmetry vs asymmetry, where in the latter value is mobile (moves from one ledger to another): in trade people exchange values at several ledgers simultaneously; in transfer the "original" value token in the first ledger is frozen or locked (or destroyed) and the "new" value token in the other ledger is unfrozen or unlocked (or created).
- **Transferring information** or generic messages between chains; a goal in Polkadot.
- **Increasing** the overall **security** of the blockchain ecosystem.

A major difference between the various approaches in how the overall immutable state is assumed to be formed. In some approaches, such as ILP[12], the immutable state is stored only in

[12] *Interledger Protocol (ILP)*. URL: https://interledger.org/rfcs/0003-interledger-protocol

the chains being interconnected. *Atomic cross-chain transactions*[13] form a subclass of these approaches, combining hashlocks and timelocks. In other approaches, such as Cøsmos and Polkadot, there is an attempt to establish a "superchain" that allows one to (double) verify the consensus on various "sub-chains" . The sub-chains are called *zones* in Cøsmos and *parachains* in Polkadot. In still other approaches, the cross-chain immutability assumptions are relaxed, essentially creating two-phase transactions that either get confirmed or expire.

In this section we have divided the various interledger approaches into the following sub-categories:

1) Atomic cross-chain transactions
2) Sidechains
3) Bridging approaches
4) Ledger of ledgers approaches
5) Off-chain transactions and transactions across a network: Lightning and Raiden
6) Layered value transfer protocols (W3C ILP)

We cover each of these separately in the following subsections. Since sidechains are in many cases — but not always — based on atomic cross-chain transactions, and the various bridging approaches are mostly based on generalised sidechains, we start with atomic cross-chain transactions, followed by sidechains and bridging approaches. The ledger of ledger and W3C ILP are explained towards the end of this section.

At the time of this writing there are very few academic peer-reviewed works in the inter-blockchain and inter-ledger areas. Adam Back et al. [Bac+14], in their — not-formally peer-reviewed yet widely available and cited — working paper, frozen in October 2014, explain the background and the basic ideas of sidechains. Chen et al. [Che+] give a half-baked idea for a BFP-based ledger-or-ledgers approach, apparently not being aware of the other approaches in the field, e.g., ILP and Polkadot. Croman et al. [Cro+16] discuss blockchain scalability in general terms, and mention sidechains and off-chain transactions as two possible scalability approaches. Dillon et al. in another non-published paper [Dil+16], propose *strong federations,* a byzantine layer on top of multiple blockchains. English, Orlandi, and Aueris, in yet another preprint [EOA16], describe the Überledger framework; see Section 2.4.5. Herlihy, in a paper to appear in July 2018 and available as a preprint [Her18], appears to present the first comprehensive work analysing and explaining the basic atomic cross-chain swap. Sun et al. [Sun+17] describe Multi-Blockchain Digital Currency (MBDC), a permissioned blockchain technology making use of a multi-blockchain architecture, targeting central banks. It falls clearly beyond the scope of our IoT applications, therefore we do not consider it any further. Tate, Johnstone, and Fielt [TJF17] briefly mention inter-blockchain communication using Cøsmos as an example, and mention that inter-blockchain communication may allow users to transfer their reputation between blockchains.

2.4.1 Atomic cross-chain transactions

This section discusses cross-chain transactions. We consider the basic problem of trading assets on two unrelated blockchains. Specifically, two parties, U1 and U2, wish to trade digital

[13] *Atomic cross-chain trading*. URL: https://en.bitcoin.it/wiki/Atomic_cross-chain_trading

assets on two blockchains, A and B: U1 wants to give user U2 some amount of assets on chain A in exchange for some amount of assets on chain B, owned by U2. Note that such an exchange involves *trading* of digital assets rather than actually *moving* digital assets from one blockchain to the other. Such a trade entails risks, since the user who first gives the other user the agreed amount of assets is faced with the risk of the other user not obeying the agreement and keeping the amount of assets he received from the first user, as well as the assets he promised to give. Indeed, the immutable nature of blockchains aggravates this issue, since transactions recorded on a blockchain cannot be revoked.

In the financial sector, this risk is handled by the so-called Delivery-versus-Payment (or Payment-versus-Payment if the trade involves currency assets) procedure, which requires the presence of a trusted third party that ensures that either both transfers occur or neither does. Hence, a first approach for performing transactions across chains involves a trusted third party that ensures the trade is atomic. Indeed, instead of a single party, the assurance of atomicity can be provided by a multisig notary scheme or a group of parties (federation).

A second approach for performing transactions across two distinct blockchains is to use *relays*. Relays are nodes participating in one of the two blockchains that can additionally read and validate the state of the other blockchain [But16]. Relays can be implemented as smart contracts on one chain B, which use the verification procedure for the other chain A's consensus algorithm. In this way, relays on chain B can verify that a particular event took place or an object is at a particular state on chain A. The above approach is also known as *sidechains* or *pegged sidechains* [Bac+14]. Sidechains can be used to move digital assets from one blockchain to another, rather than simply trading between digital assets; see Section 2.4.2 below.

A third approach for trading digital assets across chains is *atomic swaps* [But16], also known as *atomic cross-chain trading*[14,15]. This approach involves publishing transactions on the two blockchains in such a way that atomicity is ensured: either both transactions take place or neither takes place. The above atomicity is achieved without requiring a trusted third party, which can be some escrow service or exchange, and while the users involved in the trade do not trust each other. Atomic swaps are based on Hash Time-Lock Contracts (HTLC)[16], which utilize the following mechanisms:

- Multi-signature: transactions can be signed by two (or more) parties, thus making the parties accountable for the multi-signature transaction.
- Hashlocks[17]: a cryptographic lock that can be unlocked by revealing a secret s whose hash H(s) equals some value h.
- Timelocks: these lock transactions so that their assets cannot be obtained until a specific time interval has elapsed. This time can be relative to the time the transaction is published on the blockchain or absolute[18].
- Basic scripting: this is required to indicate that a transaction can be unlocked only if both a) the secret to unlock a hashlock is revealed and b) a particular signature is provided.

[14] *Alt chains and atomic transfers*. URL: https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949
[15] *Atomic cross-chain trading*. URL: https://en.bitcoin.it/wiki/Atomic_cross-chain_trading
[16] *Hash Time-Lock Contracts (HTLC)*. URL: https://en.bitcoin.it/wiki/Hashed_Timelock_Contracts
[17] *Hashlock*. URL: https://en.m.bitcoinwiki.org/wiki/Hashlock
[18] *Timelock*. URL: https://en.bitcoinwiki.org/wiki/Timelock

Hashlocks on two blockchains are used to link transactions on the two chains: opening one hashlock requires that the secret is revealed. Once revealed, the secret can be used to open a hashlock on the other chain. For this to work, the hash functions on both chains must be the same. Of course, hashlocks can also be used to link two transactions on the same blockchain.

A requirement for the atomic swap protocol is that the two blockchains must support the same hash function, such as SHA-256. Moreover, it is interesting to note that atomic swaps require basic scripting capabilities, such as those available in Bitcoin, hence do not require the more advanced smart contract programming capabilities in blockchains such as Ethereum.

Although blockchains provide an immutable recording of transactions based on distributed trust, cross-chain trading is centralized, performed by third parties. Atomic cross-chain protocols enable peer-to-peer trading, hence support decentralization and decentralized exchanges. Moreover, atomic swaps are typically cross-chain. However, atomic swaps over the same chain can be used for obfuscating the transaction graph, hence increasing privacy.

Atomic swaps have been performed between different cryptocurrencies, such as Decred and Litecoin in September 2017, between Ethereum and Bitcoin in October 2017, and between many other cryptocurrency pairs.

Hash Time-Locked Contracts[19] are also used in constructs such as the Lightning Network, for improving the scalability of Bitcoin, by enabling off-chain transactions between untrusted parties [PD16].

Atomic swaps are well-known in the blockchain community, but there is no systematic analysis of their properties. Herlihy [Her18] provides an analysis of atomic swaps when multiple parties exchange assets, showing its time and communication complexity. Also, atomic cross-chain protocols have been considered only for cryptocurrency cross-chain transactions. However, they have not been investigated for blockchains used for recording IoT events and actions.

2.4.2 Sidechains

The basic idea of a sidechain is to move some assets from one blockchain, often called the main or parent chain, to another one (or more), to conduct some transactions in the other chain(s), and then at some point move the assets back to the original chain. The most common motivation for using sidechains is that the transaction confirmation time in the sidechain is typically shorter than in the main chain. The main chain is most often Bitcoin, with its 10 minutes basic confirmation time and the longer times needed for higher security. Another typical reason is that the sidechain has some functionality which that main chain does not have, e.g., support for smart contracts.

It is important to understand the underlying market forces here. While developing new blockchains is technically easy, creating a market for them is hard, and a market is required for creating an incentive for mining. However, if the assets in a new blockchain can be securely bound to existing assets in a major blockchain, such as Bitcoin, many of these problems may be alleviated or circumvented. For example, it may be possible to issue transaction fees in a sidechain in such a manner that the sidechain miners can exchange them into main chain assets without any interaction from the other parties in the sidechain or the main chain.

The most typical sidechain logic involves the following steps:

[19] *Lightning Networks Part II: Hashed Timelock Contracts (HTLCs).* URL: https://rusty.ozlabs.org/?p=462

1) Freezing some assets in the main chain, in such a way that they can be unfrozen later.
2) Creating (or unfreezing) corresponding assets in the sidechain(s).
3) Performing transactions in the sidechain(s), perhaps moving assets further between two or more sidechains.
4) Deleting (or freezing) some or all of the original assets in the sidechain(s). This forms an agreement on how the assets are going to be further distributed in the main chain.
5) Unfreezing the assets in the main chain, moving them forward to one or more parties based on the agreement in the main chain.

While most of this can be implemented technically in a relatively straightforward manner, the tricky part is freezing the assets in the main chain so that they can be later unfrozen securely and based on the agreement(s) in the sidechain(s). The sidechain approaches differ from each other in who is trusted to unfreeze the assets in the main chain, how much they must trust each other, and what happens if some of them come to a disagreement. For example, if the validators (miners) of the main chain are completely unaware of the sidechain, the freezing of the assets in the main chain should be done in such a manner that the activity in the sidechain creates evidence that, when presented at the main chain, is considered as valid evidence of possession, allowing moving (some of) the frozen assets forward in the main chain. Furthermore, when more transactions are created in the sidechain, some of the evidence generated at the sidechain should become invalid, as the assets move forward in the sidechain.

Back et al. [Bac+14] define the following requirements for sidechains:

1) Assets should be able to be moved back to the main chain by whoever their current holder in the sidechain is, and nobody else (including previous holders).
2) There should be no ability for a dishonest party to prevent the transfer from occurring.
3) Transfers should be atomic, i.e., happen entirely or not at all.
4) Sidechains should be firewalled: a bug in a sidechain enabling creation (or theft) of assets in that chain should not result in creation or theft of assets on any other chain.
5) Blockchain reorganisations should be handled cleanly, even during transfers.
6) Users should not be required to track sidechains that they are not actively using.

While Back et al. [Bac+14] describe only a few *at that time future possibilities* for creating sidechains as they envision them, the paper has inspired a number of commercial attempts to create and utilise them, including e.g., Rootstack, Blockstream sidechains, and Lightning Network.

2.4.2.1 Federated pegs, Blockstream's Elements and Liquid

The idea of federated pegs was originally described by Back et al. in Appendix A of [Bac+14], and probably elsewhere before that, as part of the Bitcoin community folklore. The basic idea of

a federated peg is to define a fixed set of known and semi-trusted *functionaries* that jointly agree to form a Byzantine consensus on some outcomes and indicate their agreement by signing a k-of-n *multiparty signature* (a *multisignature script* in Bitcoin parlance). In the Bitcoin context, the functionaries may simply observe the Bitcoin chain and whenever they recognize there is an extension they know about, they enter their signature.

For sidechains, the functionaries would observe both the main chain, verifying that the initial transaction freezing the assets is still valid and not spent, and the sidechain, looking for a valid transaction that freezes (or destroys) some sidechain assets while requesting them to be unfrozen in the main chain. Once they see both transactions being valid, for a sufficiently long time, they give their signature in the main chain to unfreeze the assets there.

In the Bitcoin context, the federated pegs were first successfully implemented by the *Elements Project*[20] in late 2016. The Elements Project appears to be a loose collection of various experimental activities on bitcoin, utilising sidechains. The project hosts a number of repositories on Github, including the Elements library, the official version of the C language Lightning implementation, and a number of projects building on it.

The *Elements*[21] library itself is a collection of experimental features and extensions to the Bitcoin protocol, some of which have been integrated into Bitcoin. It constitutes the core software platform of Blockstream[22], a commercial startup company that has received 70M of conventional funding in two rounds (in 2014 and 2016). Their vision is to create an ecosystem of financial networks, largely based on sidechains and cross-chain compatibility.

Liquid[23] by Blockstream is an implementation of a sidechain based on strong federation [Dil+16], providing a private blockchain with different features, capabilities, and benefits than the main Bitcoin blockchain.

2.4.2.2 Merged mining, further alternatives, and Rootstock

Sergio Lerner of Rootstock[24] has attempted to analyse sidechains and related phenomena in a working paper [Ler16], mostly from an economic incentives point of view. Rootstock's current plan is to launch a system using federated pegs, however their longer term plans are unclear. They require a relatively large change to Bitcoin and merged mining, enabling what they call drivechains. In drivechains locking and unlocking of assets is controlled by the (merged) miners, while in federated pegs it is done by the functionaries outside of the main chain.

2.4.2.3 Ethereum's Plasma

Plasma [PB17] is a proposal for creating hierarchical trees of sidechains (or child blockchains) using a series of smart contracts. The Ethereum blockchain (root chain) needs to process only a small amount of commitments from sidechains, which however can perform a large amount of computations. Each sidechain is implemented through a smart contract, which can be governed by its own set of rules and constraints. The Plasma sidechains use Proof-of-Stake consensus.

[20] *The Elements Project*. URL: https://elementsproject.org
[21] *Elements Repository*. URL: https://github.com/ElementsProject/elements
[22] URL: https://blockstream.com
[23] URL: https://blockstream.com/liquid
[24] *Rootstock*. URL: https://www.rsk.co/

Mining is done with full security only on the root chain. Unlike the Lightning and Raiden Network which work strictly for payments, Plasma extends the idea to Ethereum smart contracts.

Validators report the activity taking place on a child chain to the root chain, in the form of block-header hashes rather than a full list of transactions performed on the child chain. Data is propagated only to parties that wish to validate the state of particular sidechains they are interested in. The parties monitoring a particular sidechain are responsible for penalizing fraud. States within this child blockchain are enforced via fraud proofs (smart-contract logic) that ensure that all state transitions are validated. Fraud proofs can also enforce an interactive protocol for fund withdrawals, similar to how *HTLC*s are used in the Lightning Network; these fund withdrawals need to be published on the root chain, hence require the corresponding time for performing transactions on the Ethereum blockchain.

Plasma is being actively developed and used by projects such as OmiseGo[25], whose goal is to build a peer-to-peer cryptocurrency exchange platform, and Loom[26], which provides an SDK environment for building distributed applications on their own sidechain, focusing on large-scale online games and social network applications.

2.4.3 Bridging approaches

Bridging refers to approaches that aim to provide one or two-way transfer of information between blockchains, which can support the transfer of both data and value between blockchains. Bridging approaches are simpler than ledger-of-ledger approaches, such as Cosmos and Polkadot. Bridging approaches typically involve modules or smart contracts running in nodes of the two, or in some cases in only one, interconnected chains (sidechain and mainchain) that are used for exchanging information between the two chains.

Blocknet[27] aims to be “The Internet of Blockchains” [CM]. It is founded on a protocol called XBridge, a peer-to-peer protocol that aims to enable communication between nodes on different blockchains. They were launched in 2014. In early 2017, they appeared not to be respected very high in the blockchain and crypto community. However, they are actively trying to improve their impression, e.g. by publishing the 1.0 version of their whitepaper in April 2018. Their market capitalisation has improved quite a lot since March 2017.

Their main product is a decentralized exchange that allows any Bitcoin-like cryptocurrencies to be exchanged without a centralised party, as long as the currencies involved support BIP65 (*CHECKLOCKTIMEVERIFY*), which has been in Bitcoin since late 2015. In that sense their immediate aims are similar to those of ILP, though only for cryptocurrencies, while ILP aims to address also other types of exchanges. Blocknet uses a Proof-of-Stake consensus algorithm, with three types of nodes to maintain the network: service nodes, staking nodes, and trading nodes.

ARK[28] is another system that markets itself as a bridge [Ark]. In that sense it is somewhat similar to Blocknet. That is, ARK’s so called Smart Bridges are similar to Blocknet’s XBridge in that they connect distinct blockchains and facilitate communication between them. However, ARK becomes the intermediary between different chains, using a Delegated Proof-of-Stake (DPoS)

[25] *OmiseGo*. URL: https://omisego.network
[26] *Loom*. URL: https://loomx.io
[27] *Blocknet Decentralized Application Platform*. URL: http://blocknet.co
[28] *Ark*. URL: https://ark.io

consensus algorithm. In that sense it resembles more the ledger of ledger approaches, covered below. ARK allows existing and new blockchains to communicate with one another and trigger events across chains. Additionally, ARK aims to make it easy to build new blockchains just by forking the source code from ARK.

BTC Relay[29] which was initiated by the Ethereum Foundation is a smart contract on Ethereum that can read the Bitcoin chain, thus enabling verification of Bitcoin transactions in a decentralized and trustless manner. This allows Bitcoin payments for using Ethereum smart contracts. BTC Relay was released in early 2016, being the first to provide cross-chain communication. BTC Relay uses Bitcoin block headers to build a mini-version of the Bitcoin blockchain, a method used by Bitcoin SPV light wallets. When an application processes a Bitcoin payment, it uses a header to verify that the payment is legitimate. Relayers are those who submit block headers to BTC Relay. When any transaction is verified in the block, or the header is retrieved, relayers are rewarded with a fee. Technically, BTC Relay is closer to providing an atomic swap between Ethereum and Bitcoin. Also, note that the interoperability supported by BTC Relay is one-way: Bitcoin cannot read the Ethereum chain, because its scripting language is not sophisticated enough.

The PoA (Proof-of-Authority) Network[30] is another attempt for developing a cross-chain bridge solution for connecting Ethereum-compatible blockchains. The PoA Network is based on the Parity bridge open source project[31]. Transfers are performed by depositing an amount of currency to a smart contract on the Ethereum network, thereby locking this amount. The “foreign” chain uses Proof-of-Authority consensus, whereby transactions between the main Ethereum and the foreign chains happen in a byzantine fault tolerant way using the authorities of the foreign chain.

2.4.4 Off-chain transactions and transactions across a network: Lightning and Raiden

The *Lightning Network* [PD16] is an evolving protocol and compliant implementations for two party asset locking in bitcoin, with later reallocation and commitment of the assets. That is, any two parties, say Alice and Bob, may agree to *both store* some bitcoins to a Lightning channel by publishing a funding transaction in Bitcoin. While moving funds in Lightning, both Alice and Bob always also have a latest commitment transaction, signed by both parties, which they may opt to unilaterally publish in Bitcoin. In practice, the commitment transaction is two distinct transactions: One where Alice releases Bob’s share immediately but where her share is time locked and revocable by Bob, if Bob knows the revocation key, and vice versa. When performing a transaction within Lightning, and thereby generate a new pair of commitment transactions, the revocation keys of the previous commitment transaction are revealed.

Lightning depends on the Segwit Bitcoin extension. The extension has been activated in the Bitcoin network in August 2017, at block 477,120, and has gained reasonable usage since then. Hence, at this writing Lightning appears to be a fully functional part of the Bitcoin ecosystem, undergoing active development in the community. The Lightning Network core is a separate small group of people, coordinating the maintenance of the Lightning daemon (written in Go) and protocol specification at Github.

[29] *BTCRelay*. URL: http://btc-relay.readthedocs.io
[30] URL: https://poa.network
[31] URL: https://github.com/paritytech/parity-bridge

Similar to the Lightning Network, the *Raiden Network*[32] supports off-chain transactions and transfer of value across a network of interconnected payment channels. At the end of 2017, a simplified version of Raiden called μRaiden was activated on the Ethereum mainnet. *μRaiden*[33] supports off-chain token transfers to predetermined receivers. Specifically, μRaiden uses the Raiden Network's most basic building block, namely payment channels, to allow the transfer of tokens between any two parties, with zero fees. μRaiden does not support multihop token transfers.

2.4.5 Ledger-of-Ledgers Approaches

While one of the overall goals of the interledger approaches is to move away from the "one chain rules them all" mentality, and while the Polkadot whitepaper [Woo16] explicitly mentions this as one of the motivating factors, with a closer look both Polkadot and Cøsmos appear to belong to the ledger-of-ledgers approaches. In a ledger of ledgers approach, a new "super ledger" is introduced, with the goal of having multiple sidechain-like or other subledgers.

Polkadot[34] is a proposal introduced by Gavin Wood, the author of the Ethereum yellow paper. Polkadot is open source, but most of the work appears to take place by Parity Technologies. While Polkadot is primarily described as a scalable heterogeneous "multi-chain," in reality it attempts to introduce a new, overarching relay-chain, upon which a large number of parachains can be build. Typically the parachains would be new types blockchain using the Polkadot specific *Byzantine Fault Tolerance* (*BFT*) consensus algorithm, inspired by Tangaora, Tendermint and HoneyBadgerBFT. The BFT is further turned into a PoS-like system with periodically electing the set of validators randomly from a set of bonded potential validators, using the size of their bonds as a measure of their stake. This, of course, depends on the bonds and thereby the underlying Polkadot tokens having real world value through an exchange or other mechanism. Polkadot itself provides no inherent application functionality, other than allowing the parachains (including Ethereum and Bitcoin) to relay data and eventually value between the parachains.

Cøsmos Network[35] is a project by the Interchain Foundation, a Swiss foundation registered in early 2017 by people from All In Bits, Inc. It is very similar in structure to Polkadot. The Cøsmos background is in Tendermint, a byzantine fault tolerant blockchain technology. The goal is to establish a heterogenous network of Proof-of-Stake blockchains that can interoperate with one-another. They explicitly aim to preserve the sovereignty of the side blockchains.

In addition to Polkadot and Cøsmos, English et al. [EOA16] described in 2016 the Überledger framework, a hierarchical meta-blockchain layer and an open source initiative that aimed to preserve all information related to past transactions, across block chains, thereby allowing the behaviour of the parties to accumulate, forming a basis for reputation and trust. However, the efforts seem to have cased after the initial activity in late 2016.

In the end of the day, these projects attempt to build yet another DLT to allow a set of underlying blockchains (parachains, sidechains, XXX) to pass information and value among each other. Both Polkadot and Cøsmos approach this by relying on a Byzantine consensus, but changing that into a bonded one, thereby making it a PoS system. Hence, their security at the interledger

[32] *Raiden Network*. URL: https://raiden.network/faq.html
[33] *Micro-Raiden Network*. URL: https://microraiden.readthedocs.io/en/latest/
[34] *Polkadot*. URL: https://polkadot.network
[35] *Cøsmos, Internet of Blockchains*. URL: https://cosmos.network/intro/hub

level depends by and large on creating yet another ecosystem and token, with the tokens apparently having value due to them functioning as bonds or stakes in the ecosystem and thereby giving power to the ecosystem. Whether such a practice has sustainable social value should be evaluated very carefully.

2.4.6 W3C Interledger Protocol (ILP)

The Inter-ledger protocol (ILP) project focuses on payments across payment DLT systems. In the proposed approach, transfers are escrowed in series from the sender to the recipient. The approach is heavily inspired by the Internet architecture, including layering and the ILP protocol itself drawing inspiration from the Internet Protocol (IP) [TSb].

It is possible to have "multiple Interledger protocol suites" working simultaneously. "The Interledger" is then assumed to be one public instance of such a global network, with the explicit aim of providing *a coherent global payment infrastructure*[36].

2.4.6.1 Interledger Architecture

Interledger[37] provides secure payments of any asset via different ledgers (or, classically, domains of payment). The acronym ILP can refer to two distinct entities: first, ILP is a core protocol of the interledger architecture. Second, ILP is sometimes used as a reference to a whole interledger protocol stack, which includes multiple protocols on a layered Interledger architecture. The key design philosophy is inspired by the Internet Protocol (IP): ILP aims to set as weak requirements as possible for the underlying ledgers, thus, opening up as wide of a domain of application as possible.

The interledger approach does not mandate that the parties be connected to any particular network, making it very generally applicable.

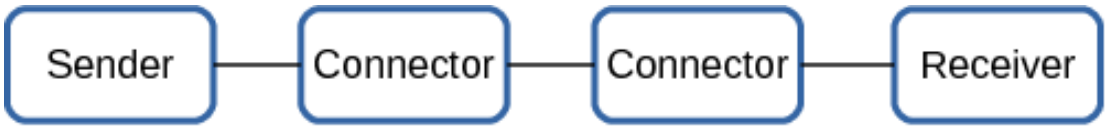

Figure 1: A Simple Interledger model

Figure 1 depicts a simple interledger model. A *sender* is a system wishing to send assets to a receiver, or to exchange an asset type for another one, by sending interledger packets. A *receiver* is a system that accepts incoming interledger packets and reacts to receiving assets by providing outbound cryptographic proofs of having received the asset.

Connectors are key systems in the whole interledger approach. They provide a routing service to interledger packets by forwarding the packets towards the destination provided in each packet, and forwarding replies towards the sender along the reverse path. Packets are forwarded via connections called *accounts*. Typically a connector would have at least two connections (or

[36] This may sound weird, but such a coherent abstraction has never been provided by the financial industry before Ripple's original contribution of the interledger protocol suite. This may be an instance of incentive incompatibilities inside the financial system.

[37] *Interledger Architecture*. URL: https://interledger.org/rfcs/0001-interledger-architecture

accounts) in different *ledgers*. Connectors may charge for their services by retaining a small fee from the forwarded amount. Along with the opportunity for revenue, connectors also have to assume a business risk, effectively having to balance the two through their own business model.

While connectors endure a risk of losing funds, senders and receivers have no such risk in the Interledger approach. Here it is assumed ledgers are shared, undeniable, and accessible between parties, for example, between the receiver and the connector nearest to it. This may or may not be true in practice with regards to a traditional banking institution.Usually it is assumed sufficient legal services are available that account holder's funds cannot just, for example, disappear in a bank, but said account holder has sufficient means and grounds to force a payment of funds, which he or she is entitled to.

2.4.6.2 The nature of ledgers in the interledger approach

In the terminology of Interledger, *a ledger* can be any payment system or domain of payment. For example, blockchains, banks and peer-to-peer payment systems are thus all ledgers. The only restricting definition is that a ledger account must consist of a single asset. Thus, a typical bank could be modelled as multiple different ledgers, each corresponding to a separate asset class with its own unit of account (or measurement). Even central banks can be considered as just one ledger each in the interledger approach.

2.4.6.3 Interledger security

The interledger protocol guarantees end-to-end security for all asset transfers: either the receiver receives the amount agreed on by the settling parties (sender and receiver), or the candidate amount is returned to the sender. Each party only needs to trust their direct peers no matter how many connectors are along the path of payment. This means a malicious connector cannot fraudulently keep sender's funds. All the risk is with the connectors: connectors can loose money because of fraudulent actions of other connectors. However, this can be managed as a normal business risk (of a financial institution), and connectors have this risk of only of their immidiate peers.

The ILP protocol essentially performs a distributed two-phase commit: first a preparation phase, and then a commit phase.

2.4.6.4 General Interledger protocol flow in the Universal mode

[TSa] describes both Universal mode and Atomic mode.

In the Universal mode, first, a prepare-packet is sent along a candidate route (see Figure 2). Each node may respond with either reject or forward the prepare packet to a next node. If the prepare packet is received by the receiver, it responds with either fulfill or reject. Fulfill-packet starts the second phase of the distributed two-phase commit. Reject, at any time from any peer on the path, starts the rollback process.

A prepare packet consists of both a cryptographic condition and a timeout. If timeout is reached, a rollback ensues. But if a fulfilment for the cryptographic condition is received before the timeout, an obligation to pay is considered generated and the receiving connector must forward

the fulfilment further. In ILP, the cryptoconditions can be anything. Usually, they are SHA256 hashes, where the fulfill packet contains a valid pre-image of the hash result that was on the corresponding prepare-packet.

The timeout values are decreased at each hop when the prepare is forwarded, to give each party a reasonable time frame to forward or to cancel.

Settlement may be based either on ledger functionality or it may be based on trust and managed as a business risk.

As the prepare packet propagates towards the receiver, funds are reserved for the case of a successful fulfilment happening later. Since no payment obligation is created yet, no funds can be lost. Once the last connector in the chain receives a successful fulfilment cryptocondition, an obligation to pay is considered generated, and the last connector has started to carry the business risk. The last connector then sends backwards the exact same fulfilment (an answer to a joint cryptopuzzle), which starts to propagate on the reverse path of connectors towards the sender. Thus, also a business risk begins shifting towards the sender.

It may happen that some connectors fail to deliver the incoming fulfilment packet before a time-out. This represents a connector risk. It is upto the transport layer protocol to decide how many retries will be made, if any.

2.4.6.5 Protocol flow in the Atomic mode

Atomic mode can be used if the underlying payment ledger support it. The high level protocol flow is exactly the same as in the Universal case. Only the security basis is different: essentially, the difference is in the trust vs. trustless paradigm of the underlying ledgers. Atomic mode is no longer currently considered part of the ILP standard.

2.4.6.6 Protocols: ILQP

Since the very early days of Interledger protocols suite, two protocols have been considered mandatory: ILP and ILQP. ILQP stands for InterLedger Quoting Protocol and it allows for a sender and a receiver to find out the costs before sending a payment. In september 2017[38], ILQP was removed from the mandatory standard.

The logic behind the removal was the following: the standard needs to be as simple as possible. All interledger payments will be small and fast. Therefore, the functionality of ILQP can be best realized by sending an even smaller test payment and observe the results. The sender and receiver can then exchange this information directly (end-to-end quoting) without requiring any special functionality in each of the connectors, thus simplifying the connector implementations significantly.

[38] https://medium.com/interledger-blog/simplifying-interledger-the-graveyard-of-possible-protocol-features-b35bf67439be

2.4.6.7 Interledger Core Protocols: SPSP and PSK2

SPSP, short for Simple Payment Setup Protocol[39], is a protocol which can be used by applications to send and receive interledger payments. SPSP uses HTTPS to exchange payment details needed by the transport layer protocols PSK2.

PSK2 is short for Pre-Shared Key V2 Transport Protocol. It uses ILP from the lower layer of the stack. It can be used for individual payment and message packets to provide condition generation, authentication and data encryption. PSK2 can be used to provide multiple unfulfillable test payments, which can be used as end-to-end-quotes. It acts as a building block for higher level protocols, such as SPSP or STREAM.

In PSK2, a sender first generates a *PSK Request*, with a source amount and a minimum destination amount, possibly by using the information from any quotes which may have been previously generated. Next, the sender generates an encryption key and a fulfillable condition. Then it encrypts the PSK request. The sender then sends an *ILP Prepare* packet, using the source amount, the fulfillable condition, and the encrypted PSK Request as the data. When the receiver gets the prepare packet, it regenerates the encryption/decryption key and decrypts the data. The amount in the ILP packet is compared with the Request Amount in the PSK packet; it should be greater than or equal. The receiver creates, encrypts and sends a *PSK Response* with the same ID towards the sender. The Receiver also generates a fulfilment, including the original PSK request, and sends it. Finally, the sender gets the fulfilment and can use it to determine the exchange rates of the payment path.

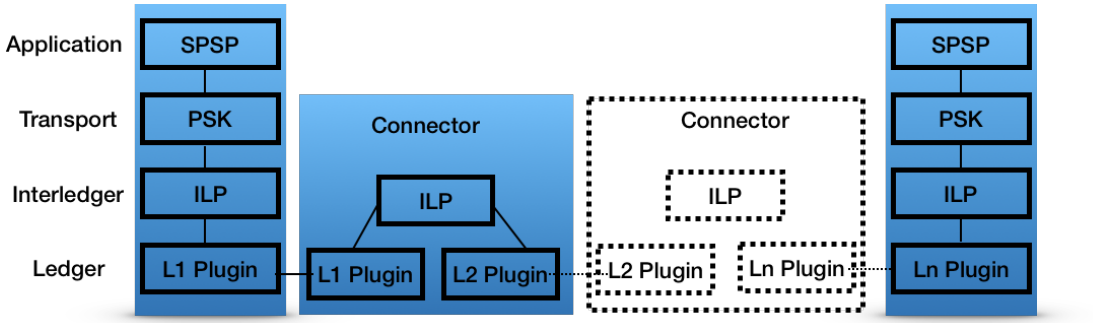

Figure 2: Full protocol stack of Interledger sender, receiver, and connectors

Figure 2 depicts how the aforementioned protocols interact in the protocol stack of ILP devices, namely, senders, receivers, and connectors.

2.4.6.8 Interledger protocol: STREAM

STREAM[40] (a successor of PSKv2) is short for STREAMing Transport for the Real-time Exchange of Assets and Messages. It is a first multiplexed Interledger Transport Protocol that provides for sending multiple “streams” of money and data between two parties using ILP. STREAM is bidirectional, that is, data can flow simultaneously in both directions. By default, it supports 10 streams in each direction.

[39] URL: https://github.com/interledger/rfcs/blob/master/0009-simple-payment-setup-protocol/0009-simple-payment-setup-protocol.md
[40] URL: https://interledger.org/rfcs/0029-stream

2.4.6.9 Interledger protocol: IPR

IPR stands for Interledger Payment Request. With this protocol a receiver initiates payment by requesting a sender to settle a particular debt. Unlike PSK and PSK2, IPR is based on signed statements and thus parties can prove to third parties that a payment has taken place. This is a very important property in a modern society, with regards to, for example, the needs of taxation or other collective societal functions.

2.4.6.10 Interledger standardization effort

W3C Interledger Payments Community Group aims to connecting the many payment networks (ledgers) around the world via the Web. The group is chaired by Adrian Hope-Bailie of Ripple and Dimitri De Jonghe. The group's vision is an open, universal payment scheme built on Web standards that allows any payer to pay any payee, regardless of the payer's choice of payment instrument or the payee's account [IPCG2017]. Their recent workshop in February 2017 focused largely around the ILP protocol proposal, and most of the traffic at their mailing list centres around ILP. The approach has also been presented at the IETF and has an active IETF mailing list.

2.5 Interfacing ledgers with external systems

In general, distributed ledgers as such do not allow contracts, including smart contracts, to access information outside the ledger. An *oracle* is a software module that obtains data from a source outside of the ledger and allows triggering of the execution of smart contracts or call APIs by external events. Hence, they can be seen as bridges interconnecting the ledger with the outside world. Oracles are required because the ledger itself is a deterministic system: the outcome of a series of smart contract computations is always the same, if the computations are performed by any ledger node or at any point of the recorded history. Determinism is necessary for the ledger consensus mechanisms to work. Unlike the determinism of a ledger, the real world is (apparently) non-deterministic. Hence, directly considering real world variable values in a smart contract would break the blockchain's consensus process, as distinct nodes in the ledger could perceive differing values. Oracles enable two-way interaction between a ledger and the outside world, resolving real world details that cannot be known at the time a smart contract is written or published on the ledger[41]. In the rest of this section, we consider some of the better known oracles.

TownCrier [Zha+16] acts as a bridge between smart contracts and existing websites, which are already commonly trusted for non-blockchain applications. It combines a blockchain front end with a trusted hardware back end to scrape HTTPS enabled websites and serves source-authenticated data to relying smart contracts. TownCrier also supports confidentiality. It enables private data requests with encrypted parameters. Additionally, in a generalization that executes smart-contract logic within TC, the system permits secure use of user credentials to scrape access-controlled online data sources.

[41] *Blockchain Oracles Will Make Smart Contracts Fly*. URL: https://hackernoon.com/oracles-help-smart-contracts-resolve-subjective-events-d81639d8291c

Oraclize[42] allows smart contract developers to fetch the results of arbitrary computations performed on systems, such as on an AWS virtual machine. Specifically, developers can upload a zip archive containing a Dockerfile, along with any necessary data, to IPFS (Inter-Planetary File System). Then, developers can include within their smart contacts queries to an Oraclize smart contract that contains the IPFS hash of the archive. An Oraclize module monitors the ledger, and once it identifies that the a Oraclize smart contract is queried, it triggers an Oraclize AWS instance to initialize and execute the Docker application identified by the IPFS hash provided by the developer's smart contract. Once the computation is completed, the Oraclize module performs a smart contract call that sends the result of the computation back to the original smart contract. An interesting extension supported by Oraclize is that it can use a (TLS) notary proof, which can be used to check the integrity of the response provided by Oraclize.

ChainLink[43] provides a decentralized oracle network, which includes on-chain components for smart contracts to request external connectivity and an off-chain consensus mechanism, reputation and bidding system to ensure that the external data is trustworthy, even though it is provided by independent oracle service providers. With ChainLink, smart contract developers do not contact oracles directly, but rather submit service requests, which include requirements in the form of SLAs, to an order-matching contract. Oracle providers can monitor and filter requests based on their capabilities and service objectives. Once a SLA request has received a sufficient number of qualifying bids, the set of oracles is selected from the pool of bids. Similar to the other approaches, ChainLink tokens provide the economic incentives for oracle providers to be truthful. While ChainLink initially focuses on the Ethereum blockchain, Hivemind[44] is another decentralized oracle proposal that is developed as a Bitcoin sidechain. Similar to other proposals, it uses economic incentives and reputation to ensure trustworthy reporting of external data.

The above solutions are data oracles[45]: they read data from external data sources; hence, these data sources need to be trusted. Computation oracles go one step further and have the oracle nodes actually perform the relevant computations. One proposal of a computation oracle was the SchellingCoin protocol proposal. It proposed a decentralized network of oracles that would perform computation, providing incentives by rewarding participants who submit results that are closest to the median of all submitted results. The proposal included a verification model that was based on m-out-of-n oracles performing computation and voting on the correct result; each node had the ability to challenge results by submitting a security deposit.

Another proposal for supporting verifiable computation is TrueBit, which introduces a system of solvers and verifiers[46]. Solvers are compensated for performing computation and verifiers are compensated for detecting errors in solutions submitted by solvers. In the event of a challenged solution, solvers and verifiers play an interactive verification game such that only a small portion of the computation is performed on-chain after a number of rounds during which the challenger disputes increasingly smaller subsets of the original computation.

Microsoft's *Coco Framework* is a project under development that targets to leverage existing blockchain technologies to support business processes [Mic17]. In particular, Coco's goals are to deliver better performance, business logic confidentiality models, distributed governance, and reduced energy consumption to existing blockchain protocols, including Ethereum, Quorum,

[42] *Oraclize*. URL: https://oraclize.it
[43] URL: https://www.smartcontract.com/link
[44] *Hivemind*. URL: http://bitcoinhivemind.com
[45] *Off-Chain Computation Solutions for Ethereum Developers*. URL: https://medium.com/@YondonFu/off-chain-computation-solutions-for-ethereum-developers-507b23355b17
[46] *TrueBit*. URL: https://truebit.io

Corda, and Hyperledger Sawtooth. Coco promises to fulfil these goals through the use of trusted execution environments, such as Intel's SGX and Windows Virtual Secure Mode, enabling the creation of a trusted network of physical nodes on which to run a distributed ledger.

3. IoT Systems and Semantic Interoperability

3.1 IoT platforms

Currently, there is a large number of competing IoT platforms [Min+15], [Min+16]. According to [Gmb16], there are some 360 distinct IoT platforms in various categories. Many of the platforms are proprietary and either fully or partially closed source, e.g., Axeda, Carriots, and Niagara [Min+15].

Figure 3, below, illustrates an eight-layer IoT architecture, adopted from the UNIFY-IoT project report. At the bottom, we have the usual physical and network layers, followed by processing, storage, data abstraction and service layers, roughly corresponding to the layers typical in the OSI 7-layer model and many other architectures. On the top there are application and collaboration layers.

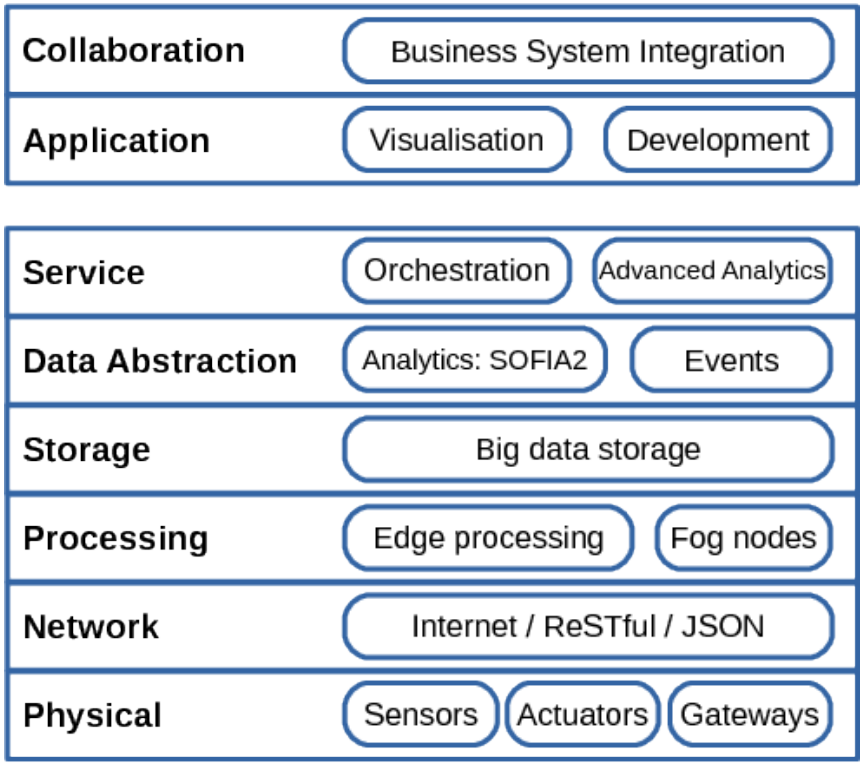

Figure 3: IoT eight-layer architecture, adopted from UNIFY-IoT project report [H2016]

In many typical cases, there are several potential federation or integration points. Firstly, at the lowest network layer it may be possible to access the devices directly with CoAP or HTTP based ReSTful APIs. The semantic internetworking, processing and storage layers may provide another set of APIs, such as the NGSI 9/10 APIs provided by the FIWARE Orion router or the FIWARE Cygnus storage. Similar APIs may also be found at even higher layers.

A common aspect of these APIs is that there does not seem to be any unified way of addressing security. For most deployment, security is dealt with separately by running the APIs within TLS or DTLS “tunnels” . However, this does not provide any end-to-end security.

The W3C Web of Things (WoT) Interest Group (IG) and Working Group (WG) have taken a

different, perhaps more practical approach. Their main concept, the WoT servient, is illustrated in Figure 4. From the UNIFY-IoT eight layer architecture point of view, the Proprietary APIs may be considered as a part of the physical layer, the Protocol Bindings as a part of the Network layer, and the Scripting API belonging to the Processing layer as a part of edge processing. The Thing Descriptions probably contribute to the Data Abstraction layer, though the UNIFY-IoT data abstraction seems to mostly concerned about higher layer abstractions.

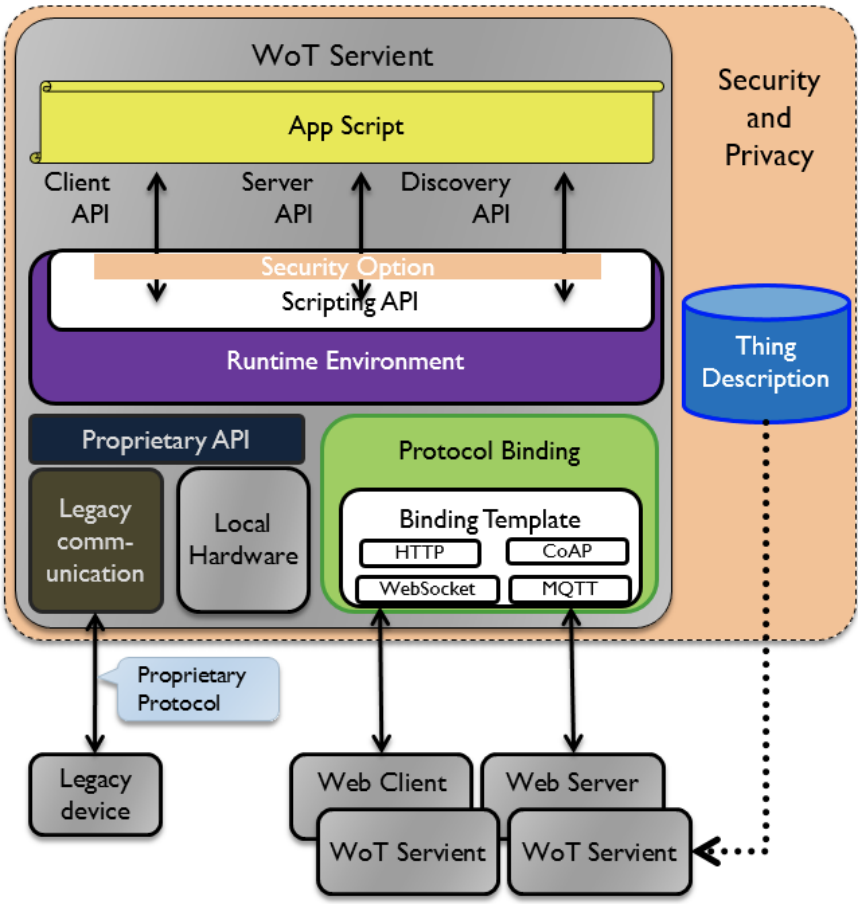

Figure 4: Functional Architecture of W3C WoT Servient [KKD17]

Alternatively, one may claim that the WoT architecture is sufficient to fulfil all the functions that belong to the Network to Service layers in the UNIFY-IoT architecture by allowing WoT servients to be layered on the top of each other, with the potential exception of Storage that should most probably be handled separately.

In addition to the generic IoT service architectures, as those briefly described above, there are several tens of architectures that are meant for specific application areas. There the focus is often more on application specific data and control abstractions. However, those fall beyond the scope of this document.

While UNITY-IoT architecture gives one conceptual tool for modelling IoT architectures and W3C WoT is mostly concerned about lower-layer practical interoperability between conforming IoT systems, there are a few efforts that aim to standardise and provide interoperability throughout the stack, from devices to semantically rich applications.

The oneM2M standards[47] focus in providing semantic interoperability between different IoT platforms. They introduce a common services layer, consisting of Common Services Entities (CSE), between applications (where data processing occurs) and networks (where communications capabilities reside). The service layer concept facilitates interoperability with existing technology modules through plug-in capabilities, including security-management modules. For example, FIWARE includes a oneM2M IoT agent[48]. Interoperability between FIWARE and oneM2M has been demonstrated several times, e.g., at the ETSI 2015 workshop [Vel15] and is also available as an open source component.

While the oneM2M standards cover a wide range of services, the standards are mostly descriptive and typically allow for many different options. Furthermore, most of the focus is on high level concepts, making it hard to pinpoint exactly how practical interoperability is meant to be implemented.

Among the industry and open-source efforts to standardise interoperability, the following appear noteworthy, especially from the device and networking interoperability point of view. IoTivity[49] is an open-source software framework that aims at facilitating device-to-device connectivity. IoTivity is sponsored by the Open Connectivity Foundation (OCF) and targets to an open source reference implementation of the OCF standard specifications. In a similar spirit, OMA's LightweightM2M (LWM2M)[50] defines an ontology model that can be used for managing things.

At this point, we would like to present two **well-established, open-source** IoT platforms, that are being used both commercially and research-wise by companies and other EU projects; *Kaa* and *DeviceHive*:

The **Kaa IoT platform**[51] provides an open-source, Cloud-based, scalable IoT framework which is feature-rich and flexible in terms of data management, device integration and protection. Kaa enables data management of connected things and back-end infrastructure by providing the Kaa server (or Kaa node) and the endpoint Service Development Kit (SDK) components[52]. The Kaa server provides all the back-end functionality needed to operate the IoT solution and offers administrative capabilities. It handles all the communication across connected devices, including data consistency and security, device interoperability, and failure-proof connectivity and features well-established interfaces for integration with internal and external data management and analytics systems. In Kaa terminology, an endpoint SDK is a library which provides communication, data marshalling, persistence, and other functions available in Kaa for specific type of an endpoint (e.g. Java-based, C++-based, C-based, Objective-C-based). The SDKs get embedded into the connected devices and implement real-time bi-directional data exchange with the server. These SDKs can be used to create Kaa clients, which are any pieces of software that utilize Kaa functionality and are installed on the connected devices. It is the responsibility of the Kaa client to process structured data provided by the Kaa server side (configuration, notifications, etc.) and to supply data to the return path interfaces (profiles, logs, etc.).

A Kaa endpoint is a particular application which uses the Kaa client SDK and resides on a particular connected device. The Kaa endpoint SDK provides functionality for communicating with

[47] *oneM2M: Standards for M2M and the Internet of Things*. URL: http://www.onem2m.org
[48] *FIWARE oneM2M adapter by Telefonica*. URL: https://github.com/telefonicaid/iotagent-onem2m
[49] *IoTivity Project*. URL: https://www.iotivity.org
[50] *OMA LightweightM2M v1.0 Overview*. URL: http://www.openmobilealliance.org/wp/Overviews/lightweightm2m_overview.html
[51] *Kaa IoT platform*. URL: https://www.kaaproject.org
[52] *Kaa IoT platform, High-level architecture*. URL: https://docs.kaaproject.org/display/KAA/Design+reference$%5C#$%20Designreference-High-levelarchitecture

the Kaa cluster, managing data locally in the client application, as well as provides integration APIs. The client SDK abstracts the communication protocol, data persistence, and other implementation details that may be specific for any concrete solution based on Kaa. A Kaa cluster represents a number of interconnected Kaa servers delivering three types of services, i.e. control, operations, and bootstrap services, that can be enabled or disabled by the Kaa administrator in an individual manner. Specifically,

- A Kaa control service is responsible for managing overall system data, processing API calls from the Web UI and external integrated systems, and delivering notifications to operations servers.
- A Kaa operations service is responsible for concurrently handling multiple requests such as endpoint registration, processing endpoint profile updates, configuration updates distribution, and notifications delivery. In the case of multiple nodes (and thus operations services), Kaa cluster provides tools for workload re-balancing at run time to effectively routing endpoints to the less loaded nodes.
- A Kaa bootstrap service is responsible for directing endpoints to operations services. Kaa endpoints have a built-in list of set bootstrap services to query and retrieve a list of currently available operations services from them, as well as security credentials.

Apart from Kaa endpoints and cluster, the Kaa instance also makes use of the following:

- an Apache ZooKeeper for services coordination, continuous monitoring and information sharing about servers' connection parameters and workload status,
- an SQL database instance to store metadata about endpoints, applications, endpoint groups, etc.,
- a NoSQL database instance (Apache Cassandra or MongoDB up to Kaa 0.8.x), with volume scaling linearly with the number of endpoints, to store measurements of devices and also information about endpoint profiles, notifications, configurations, etc.

The **DeviceHive**[53] IoT platform is a customizable, feature-rich and open-source platform with a programming framework. Further, DeviceHive is a well-maintained, up-to-date solution that integrates several cutting-edge technologies in IoT communication and offers a series of powerful services and plugins supporting the development of more efficient, secure and highly performing smart applications. As shown in Figure 5, DeviceHive exposes an API as the central part of its framework allowing the different components to interact with each other and exchange messages in real time. The platform has its own GUI - admin console which allows end user to create and register devices, connect them to networks, and manage users' authentication and authorization services.

One of the advantages of DeviceHive is that it empowers online processing and machine learning over the collected data (which are stored in a time-series database) with Apache Spark to enable business intelligence and analytics. In addition to Apache Spark, the DeviceHive platform is integrated with other state-of-the-art technologies and frameworks in big-data stacks, such as ElasticSearch, Cassandra, Kafka, etc. An additional attractive feature is the integration of the

[53] *DeviceHive IoT platform*. URL: https://devicehive.com

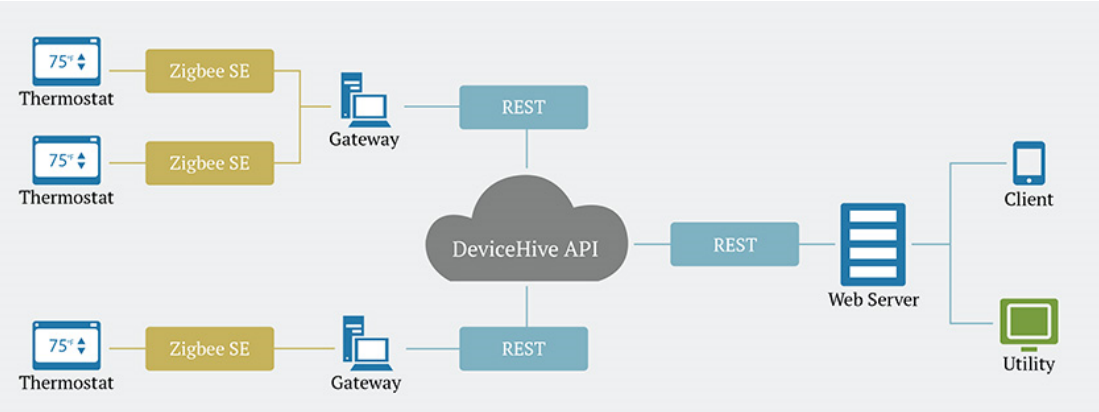

Figure 5: High-level architecture of DeviceHive IoT platform

platform with an out-of-box gateway that supports the LoRa[54] protocol for data transmission over long distances with low power consumption.

EU funded projects

WISE IoT is a collaboration project between Europe and Korea funded under the H2020 framework program at the EU side. It aims at deepening the interoperability and interworking of IoT existing systems. The Wise-IoT[55] vision is creating a reference architecture for semantic interoperability of worldwide IoT systems. Wise-IoT is built upon existing systems, standards, and reference implementations, as far as they are publicly available, e.g. as open source.

IoT-EPI is a European initiative for IoT platform development aiming at creating opportunities for platform development, interoperability and information sharing. Currently, IoT-EPI acts as an umbrella of seven projects, namely symbIoTe, bIoTope, BIG-IoT, Agile, Vicinity, TagItSmart!, and INTER-IoT. The **symbIoTe**[56], **INTER-IoT**[57], and **bIoTope**[58] projects aim at providing interconnection and interoperability of heterogeneous IoT platforms through a "super platform" that will enable a unified view of the available resources, IoT platform federation, and it will facilitate discovery, sharing of resources, and new applications.

The **BIG-IoT** project[59] targets to define a unified Web API for IoT platforms, aligned with the standards currently developed by the W3C Web of Things (WoT) group.

The goal of the **Agile** project[60] is to build a modular and adaptive gateway for the IoT; this gateway will support multiple hardware components and various protocols, providing at the same time a unified API to applications.

The **Vicinity** project[61] will build decentralized "social networks" for smart objects, called virtual neighborhoods, that will enable secure sharing of resources and data, whereas the **TagItSmart!**

[54] *LoRa Alliance*. URL: https://lora-alliance.org/
[55] *Wise IoT*. URL: http://wise-iot.eu/en/home/
[56] *symbiosis of smart objects across IoT environments (symbIoTe)*. URL: https://www.symbiote-h2020.eu
[57] *Interoperability of heterogeneous IoT platforms (INTER-IoT)*. URL: http://www.inter-iot-project.eu
[58] *The bIoTope Project*. URL: http://www.biotope-project.eu
[59] *Bridging the Interoperability Gap of the Internet of Things (BIG-IoT)*. URL: http://big-iot.eu
[60] *Adaptive Gateways for dIverse muLtiple Environments (Agile)*. URL: http://agile-iot.eu
[61] *Open virtual neighbourhood network to connect IoT infrastructures and smart objects (Vicinity)*. URL: https://www.vicinity2020.eu

project[62] aims at leveraging printable QR codes, printed using context-sensitive ink (e.g., sensitive to humidity) in order to build novel services.

3.1.1 FIWARE

FIWARE is an EU initiative supported by European Union's Seventh Programme for research, technological development and demonstration with the aim "to build an open sustainable ecosystem around public, royalty-free, and implementation-driven software platform standards that will ease the development of new Smart Applications in multiple sectors"[63]. Based on this definition of the FIWARE mission, the FIWARE community is an open community whose members assists towards the materialization of this mission/vision. In general, the FIWARE ecosystem is composed of:

- A concrete set of software components exposing their functionalities that are meant to be generic-enough to be usable in a variety of cases spanning inter-disciplinary areas related to practical computer science applications. These software components are usually referred to as Generic Enablers (GEs). The GEs are generally split into seven distinct categories (chapters in FIWARE terminology) addressing multiple contexts and application domains including data/context management, Internet of Things (IoT) services enablement, advanced Web-based user interfaces, security, interfaces to networks and devices, services ecosystem and delivery frameworks and, last, Cloud hosting services[64]. These APIs are supported by a complete set of reference GE implementations (GEri) implementing them. The GEris are open source, licensed with generally exploitation-friendly licenses.
- *FIWARE Lab*[65], a federation of FIWARE-enabled Data Centres (DCs), offering Cloud-hosting services to FIWARE-oriented projects, activities, and initiatives, actively delivering a "sandboxed open innovation framework". See Figure 6.
- An acceleration programme, entitled FIWARE Accelerator[66], promoting the FIWARE initiative through managed incubation processes offered by sixteen (16) different incubator projects.
- A business incubation programme, named FIWARE iHUBs[67], aiming at creating emerging local digital hubs and communities to foster FIWARE-enabled "Internet-based business creation at local level".
- A globalization/expansion programme, named FIWARE Mundus**fiwre_mundus** targeting at expanding FIWARE presence (including iHUBs, supporting FIWARE Lab nodes, accelerators etc.) outside the EU.

Apart from the FIWARE-supported infrastructures and software (in the form of the GEs), the FIWARE initiative has successfully hosted several domain-specific research projects that (i) have validated the operational efficiency of the FIWARE Lab and GEs, and (ii) have developed

[62] *TagItSmart!* URL: http://www.tagitsmart.eu
[63] *FIWARE Community*. URL: https://www.fiware.org/about-us
[64] *FIWARE Cloud Hosting Services*. URL: https://catalogue.fiware.org/
[65] *FIWARE Lab*. URL: https://www.fiware.org/lab
[66] *FIWARE Accelerator Programme*. URL: https://www.fiware.org/community/fiware-accelerator-programme/
[67] *FIWARE iHUBs*. URL: https://www.fiware.org/ihubs/

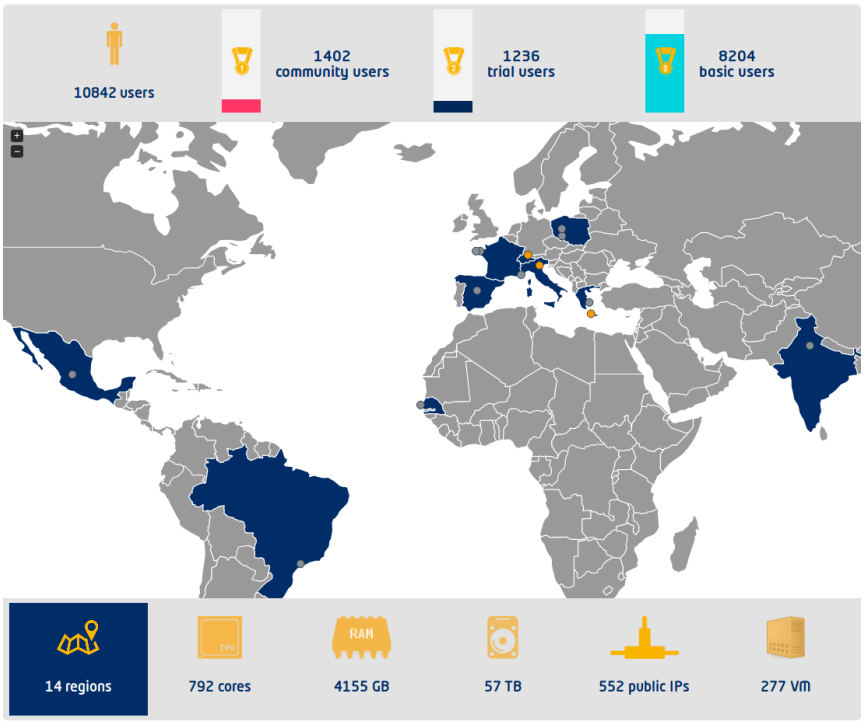

Figure 6: FIWARE Lab capacity details (http://infographic.lab.fiware.org/, accessed 2018-06-11)

Domain-Specific Enablers (DSEs) that facilitate the integration of domain-specific services into the FIWARE ecosystem. In the context of SOFIE, the potential exploitation of the FIWARE platform, and especially of the IoT functionality provided by specific FIWARE GEs, will be considered and evaluated.

Based on a preliminary internal study, especially the following FIWARE components look promising:

- FIWARE Orion context broker
- FIWARE LWM2M IoT agent
- FIWARE Cygnus persistence connector

3.2 Semantic interoperability

Both the number and the amount of data produced by IoT devices can be huge. Moreover, the characteristics of IoT devices, as well as the format, type, and domain constraints of the heterogeneous data they produce, can differ significantly. To enable wider use of IoT devices and the data they produce across different IoT platforms, a common format for describing them is necessary. Semantic interoperability is defined as "enabling different agents, services, and applications to exchange information, data and knowledge in a meaningful way, on and off the Web"[68].

[68] *Semantic Integration & Interoperability Using RDF and OWL*. URL: http://www.w3.org/2001/sw/BestPractices/OEP/SemInt

A common representation framework enables the identification of the capabilities of sensors in different IoT platforms and the interpretation of the data they produce in a unified and unambiguous way, by both humans and machines. Semantic interoperability is different (and at a higher level) compared to technical and syntactic interoperability. Technical Interoperability is usually associated with hardware/software components, systems and platforms that enable machine-to-machine communication to take place, and focuses on the communication protocols. Syntactical Interoperability is usually associated with data formats and considers high-level transfer syntaxes such as HTML, XML, ASN.1, or JSON.

3.2.1 Metadata

Metadata is about describing the contents and context of data to facilitate discovery, understanding and (re)usability of that data [Bas+16]. The actual meaning of data can only be discovered by examining the software that generated and processed the data, i.e., the context in which the data was produced. Hence, metadata should provide this context and descriptions that enable the correct interpretation and understanding of the data values. Only then can interoperability and reusability be achieved. Devices may expose their metadata directly, or it could be held separately. This allows the Web of Things to support existing IoT devices, including those without the necessary resources to handle the metadata themselves. Also, it is not the goal of the Web of Things working group to standardize domain-specific metadata vocabularies, since this is the role of industry specific organizations [RAC16].

3.2.1.1 W3C Web of Things

The Web of Things working group of W3C seeks to extend existing Web standards and develop standard metadata and APIs to allow the interoperability of IoT platforms. Hence, they seek to extend the Web from a Web of Pages to a Web of Things[69]. A Thing can be a physical or virtual entity. Although the Thing doesn't have to be online, their capabilities or data are made available to applications through APIs according to the Thing's properties, supported actions, and events. The three building blocks for achieving the Web of Things are the following [RAC16]:

- URIs (Uniform Resource Identifiers) for identifying Things and their descriptions
- A variety of protocols for accessing Things, that can be used for different contexts
- Metadata for describing Things as a basis for interoperability and discovery

A URI (or Internationalized Resource Identifiers – IRI, which support all Unicode characters) provides two functions: (1) assigns uniquely identifiable names to things (resources) and (2) specifies the location of the resource [SW16]. Note that a URL provides only the second functionality. URIs can be used to access machine-interpretable descriptions of Things that enable the automatic generation of scriptable objects, whose interaction capabilities correspond to those of the Thing the object represents [RAC16]. The application logic using the software objects can be hosted on the same device as the Thing, or on other devices such as a local hub or a Cloud platform. The interactions between Things and applications are performed through

[69] *3C Web of Things*. URL: http://www.w3.org/WoT/

defined Application Programming Interfaces (APIs). The protocols used for accessing Things include HTTP, CoAP, MQTT, XMPP, WebSockets, WebRTC, and others, which can support various communication patterns such as pull, push, pub-sub, and peer-to-peer, as well as support for multiplexing and buffering.

A central building block for the Web of Things is the W3C Thing Description (TD) [KK18], which can be considered as the entry point of a Thing, often referred to as the `index.html` of the Thing. The TD consists of semantic metadata for the Thing itself, a narrow-waist interaction model with WoT's Properties, Actions, and Events, which is machine-understandable and features web-linking to express relations among Things.

3.2.1.2 Open Group IoT WG

Two Open Group Internet of Things (IoT) Standards related to the description of Things and the transportation of IoT data are the Open Data Format (O-DF) and the Open Messaging Interface (O-MI), respectively. The Open Data Format (O-DF)[70] is a generic content description model for Things in the IoT, fulfilling the same role as HTML does for the Internet. However, while HTML-coded information is mainly intended for human users, O-DF represents IoT information mainly for processing by information systems. O-DF is structured as a hierarchy with an object element at the top level and sub-elements contained in any number of levels. Object elements are identified by at least one id and can optionally have a description. Object elements usually have properties, which in turn can have metadata such as value type, units, and other similar information.

The Open Messaging Interface (O-MI)[71] is used for transporting IoT data between nodes in a peer-to-peer fashion. O-MI messages can be transported using most "lower-level" protocols, such as HTTP, SOAP, SMTP, etc. Other properties and requirements for O-MI include the following: read, write, and cancel operations are supported; subscriptions can be made for deferred retrieval of data; all requests and responses can specify a Time-to-Live (TTL); synchronous communication is supported by allowing a response message to include a new request.

3.2.2 Ontologies

Ontologies build on metadata to provide a representation of knowledge about a given domain and to provide a core resource for reasoning about a domain and a context [Bas+16]. An ontology is a vocabulary with a structure that 1) captures a shared understanding of a domain of interest and 2) provides a formal and machine interpretable model of the domain [one18].

3.2.2.1 W3C Ontology

RDFS (Resource Description Framework Schema) and OWL (Web Ontology Language) are ontology languages standardized by W3C. RDFS is based on RDF (Resource Description Framework) that provides a mechanism to describe Web resource uses triples, which consist of sub-

[70] *Open Data Format (O-DF), an Open Group Internet of Things (IoT) Standard*. URL: http://www.opengroup.org/iot/odf/

[71] *Open Messaging Interface (O-MI), an Open Group Internet of Things (IoT) Standard*. URL: http://www.opengroup.org/iot/omi/

ject, property and object; all three can be a URI, while the object can also be a literal. Linking heterogeneous formats has been identified as a promising way of integrating data from different sources and interlinking semantic descriptions by the European Research Cluster on the Internet of Things [Ser+15]. RDFS (RDF Schema) provides a Vocabulary Description Language for RDF, defining class and attribute hierarchies. OWL further extends RDFS to model knowledge bases, with support for logic inference that can be used to acquire implicit knowledge. In the Semantic Web, vocabularies define the concepts and relationships (also referred to as "terms") used to describe and represent an area of concern[72]. There is no clear division between what is referred to as "vocabularies" and "ontologies". The trend is to use the word "ontology" for more complex, and possibly quite formal collection of terms, whereas "vocabulary" is used when such strict formalism is not necessarily used or only in a very loose sense.

Ontologies are often modular, which is driven by use-cases where only parts of an existing ontology are needed, or where constrained devices are unable to process the full ontology. Moreover, modularization can be horizontal or vertical. Horizontal modules are subsets of an ontology that have very loose or no coupling. Examples of such modules include the geolocation and time ontologies. On the other hand, vertical modules form a hierarchy where the modules on the top are more general ontologies than those on the bottom. Example ontologies include the Semantic Sensor Network (SSN) ontology [Atk+17] and the oneM2M Base Ontology [one18].

The W3C Semantic Sensor Network (SSN) ontology [Atk+17] is an ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties, as well as actuators. The SSN ontology was developed jointly by the W3C and OGC (Open Geospatial Consortium) Spatial Data on the Web Working Group. The SSN ontology addressed limitations of the old SSN ontology developed by the W3C Semantic Sensor Network Incubator Group, which was perceived as too heavyweight for many use cases. The SSN ontology consists of a lightweight but self-contained core ontology called SOSA (Sensor, Observation, Sample, and Actuator) and the more expressive SSN (Semantic Sensor Network) ontology that follows a horizontal and vertical modularization architecture. The W3C SSN ontology has been used and extended by a number of IoT efforts and projects. This includes the OpenIoT platform which has extended SSN with Cloud-computing concepts to support on-demand Cloud-based access to the IoT resources [Sol+15]. Another example is the IoT-Lite ontology, developed by the EC funded FIWARE and FIESTA-IoT projects, which is an instantiation of the SSN to provide a lightweight ontology to represent Internet of Things (IoT) resources, entities and services [Ber+17]. IoT-Lite can be combined with modules for representing IoT data streams and supports dynamic semantics for inferring missing sensor values.

3.2.2.2 OneM2M Ontology

Another ontology is the oneM2M Base Ontology [one18], which is used by the popular OneM2M IoT platform. oneM2M's Base Ontology constitutes a basis framework for specifying the semantics of data that are handled in oneM2M. Sub-classes of some of its concepts are expected to be defined by other bodies in order to enable semantic interworking.

[72] *W3C Semantic Web Ontologies*. URL: http://www.w3.org/standards/semanticweb/ontology

3.2.2.3 Other Ontologies

An important conclusion from recent research is that, although there exist general ontologies for sensors and sensing, such as the W3C SSN, and M2M device level ontologies, such as the oneM2M Base Ontology, the level of maturity and openness of application specific ontologies differs [Gan+17]: In the health domain there exist mature ontologies, mainly in the form of taxonomies, that have been systematically developed, but the extraction of a formal ontology for interoperability may be necessary. On the other hand, in the transport and logistics sector, local/proprietary formats are mainly used and in most cases the semantics are not formalized.

The work in [Aga+16] defines an ontology that leverages a number of core concepts from various mainstream ontologies and taxonomies, which include SSN, IoT-lite, Time and others. The work in [Gan+18] investigates different approaches for the interoperability of ontologies, which include a one-to-one translation and a translation of ontologies to a central ontology; the latter approach can further use modularization, where a central or core module is common to all stakeholder participating in an ecosystem and can be extended using domain and context specific modules. The specific approach is applied to the transport and logistics application domain.

In order to drive IoT metadata sharing and interoperability, similar to the Linked Open Vocabularies (LOV) community initiative to drive the creation and use of shared vocabularies, there is an IoT-specific initiative[73] for collecting vocabularies in the IoT domain. Similar to linking of data formats, alignment between different vocabularies is important for achieving semantic interoperability [Ser+15]. Another effort to promote common ontologies for semantics interoperability through a community effort is iot.schema.org, which focuses on simple semantic models to abstract the functionality of Things.

During the past few years, some surveys have been done on applying the Semantic Web technologies to the IoT. [Bar+12] is an early survey on the application of Semantic Web technologies and standards to IoT, and in particular, information modeling, ontology design, and processing of semantic data. [Jar+14] is another dated survey that focuses on Semantic Web standards for heterogeneous device integration, device abstraction, and different semantic descriptions in the IoT. Both surveys pre-dated more recent developments such as the W3C's lightweight core ontology SOSA (Sensor, Observation, Sample, and Actuator). [Sey+17] presents a classification of research contributions related to the application of Semantic Web technologies to IoT using a categorization of IoT nodes based on their capabilities and functions. [Baj+17] surveys semantic ontologies, including both generic and domain specific ontologies, highlighting their ability to address fundamental ontological concepts, such as sensor-capabilities and context-awareness.

[73] *Linked Open Vocabularies for Internet of Things*. URL: http://lov4iot.appspot.com

4. DLT meets IoT

This section visits the most prominent existing IoT systems that have already adopted distributed ledger technologies into their platforms. At a high level, these can be split into three broad categories, namely industry-led efforts (such as the Trusted IoT Alliance, Hyundai's HDAC and IBM's Watson IoT platform), blockchain-community efforts (such as Streamr and Flowchain), and marketplace-oriented efforts (such as the IOTA Marketplace and Datum).

Currently, there exists a variety of IoT solutions that take advantage of blockchain technologies. Tileplay[74] offers a blockchain-based marketplace, where users can sell and buy data generated by Things they own. Catenis[75] provides Bitcoin-blockchain based secure messaging for the IoT. *Riddle&Code*[76], as well as *Chronicled*[77], provide a service that allows users to register real-world object identities in a blockchain (e.g., for claiming ownership or for tracking fraud). Slock.it[78] implements a blockchain-based market place where users can pay for services provided by real-world Things (e.g., a user can pay in virtual coins in order to open the door of a rented apartment). Transactivegrid[79], Gridgularity[80], and SolarCoin[81] combine smart grid with blockchains and they allow devices to record energy consumption, as well as to buy energy, all with the help of a blockchain. Farmshare[82] leverages blockchain to facilitate food donations by local farmers. Provenance[83] uses a blockchain to record events related to product supply chains, focusing on agricultural products.

4.1 Hyundai Digital Asset Currency

Hyundai Digital Asset Currency (*HDAC*) is a platform that targets to combine IoT technologies with a blockchain-based solution in order to provide "Authentication", "Mapping", and "Machine-to-Machine" communication [Hyu17]. The HDAC platform implements a private blockchain, code-named HdacT, that provides a simple transaction environment targeting M2M communications. HDAC plans to provide a hardware-based wallet that will enable users to interact with HdacT, as well as a public blockchain that will be used for inter-connecting private ones.

HDAC is based on Multichain. It targets 160 transactions/sec in the public blockchain and 1000 transactions/sec in private blockchains. It implements a consensus algorithm based on Proof-of-Work (PoW), code-named ePoW, targeting energy efficiency. In short, ePoW limits the number of miners participating in a particular mining task by specifying a "block window", a period of time during which miners are not allowed to mine blocks.

The main component of the HDAC platform is the "IoT contract", a smart contract intended to be installed on IoT devices to interact with the blockchain. Using IoT contracts, Thing owners can specify operations for the devices to perform, as well as security and access control policies.

[74] *Tileplay*. URL: http://www.tilepay.org
[75] *Catenis*. URL: http://blockchainofthings.com
[76] *The blockchain interface company*. URL: https://www.riddleandcode.com/
[77] *Chronicled*. URL: http://www.chronicled.com/
[78] *Slock.it*. URL: https://slock.it
[79] *LO3 Energy*. URL: https://lo3energy.com/
[80] *Gridgularity*. URL: http://gridsingularity.com
[81] *SolarCoin*. URL: https://solarcoin.org
[82] *Farmshare*. URL: http://farmshare.org
[83] *Provenance*. URL: https://www.provenance.org/

4.2 IBM Watson with Hyperledger integration

Of the big companies, IBM appears to be the market leader with its Watson IoT with Blockchain, which enables IoT devices to send data to private blockchain ledgers for inclusion in shared transactions with tamper-resistant records. IBM Blockchain on Bluemix (IBM's Cloud platform) is apparently the commercial version of the Hyperledger Fabric, of which IBM is a premier member.

IBM aims at promoting its blockchain platform for use in a number of industries, including banking and financial markets, insurance, retail and consumer goods, government, healthcare, automotive, travel and transportation, and media and entertainment. It remains somewhat unclear how IBM plans to apply their blockchain architecture in the IoT domain.

4.3 Streamr

Streamr is a platform that utilizes blockchain technology to allow the creation of real-time data decentralized applications (Dapps) for IoT. Comparing to IOTA, which was designed specifically for IoT, it leverages existing blockchain technology and builds upon that. It is, in essence, a decentralized peer-to-peer (P2P) network that (claims it) offers scalability, low-latency, untamperable data delivery, and persistence. The solution that the Streamr platform offers is decentralized messaging and event processing, while it aims to replace platforms such as Azure EventHub[84] and Azure StreamAnalytics[85].

Streamr platform stack consists of 5 layers for which a brief description follows:

1) **Streamr Editor:** A graphical user interface offering non blockchain technology experts a toolbox for the development of Dapps.
2) **Streamr Engine:** An event processing and analytics decentralized engine that can run within a decentralized computing provider.
3) **Streamr Data Market:** A universe of data streams which can be published or subscribed to. It allows the exchange and monetization of data. The data that traverse throughout the network are events (timestamped data) batched into streams.
4) **Streamr Network:** The data transport backbone and core of the platform. A P2P network whose basic software building block is the Broker Node.
5) **Streamr Smart Contracts:** This is how the platform utilizes the blockchain technology. Smart contracts are used for incentivization, network coordination, permissioning, and integrity checking.

We will not discuss any of the details of the two first layers despite them being very interesting conceptually, we will not focus on the Data Market layer which, simply put, is the place where users can "buy" and "sell" their data using DATAcoin and a publish/subscribe mechanism, but we will rather provide some more details about the last two layers, the Network layer and the Smart Contract layer.

[84] *Azure Eventhub*. URL: https://docs.microsoft.com/en-us/azure/event-hubs/
[85] *Azure StreamAnalytics*. URL: https://azure.microsoft.com/en-us/services/stream-analytics/

The Streamr Network combines the characteristics of systems such as Kafka, ZeroMQ, etc., which are scalable Cloud-based real-time data transport systems and decentralized P2P systems such as Whisper, Bitmessage, etc.. In that way, it can offer high throughput for real time data applications, that the former type of systems achieve, while at the same time being able to effectively route data, discover peers, etc, that the latter systems support. Through the Smart Contracts layer the Network uses an underlying Ethereum (not limited to this specific technology) stack which is not used for data storage but rather for:

- **stream registry,** storing administrative information about data streams,
- **network partitioning and coordination,** i.e. reaching consensus on splitting the network into partitions and assigning them to specific broker nodes,
- **incentivization,** using DATAcoin as the network's usage token to subscribe to data streams and to reward broker nodes for carrying out their tasks; data integrity service via checksums and data delivery service,
- **event persistence,** i.e storing the series of events offering robustness, fault tolerance,
- **data provenance and data licensing,** cryptographically sign data with a private key and grant access only to authorized users for a specific time period.

4.4 IOTA Marketplace

IOTA Marketplace[86] is a pilot application of the IOTA technology. IOTA marketplace is in essence a data marketplace where users can purchase access to a sensor data stream. Currently in the platform there are tens of sensors available, mainly provided by the IOTA consortium members.Payments are made using the IOTA currency.

Sensor measurements are propagated in real time over the IOTA network using the "Masked Authenticated Messaging" (MAM). MAM is "data communication protocol which adds functionality to emit and access an encrypted data stream, like RSS, over the Tangle [Pop17] (IOTA's distributed ledger) regardless of the size or cost of device. IOTA's consensus protocol adds integrity to these message streams"[87].

4.5 Flowchain

Flowchain [Che] is a proposal for supporting peer-to-peer IoT networks and real-time transactions over a blockchain system. In Flowchain, each node can mine blocks, referred to as *virtual blocks*, in its own branch. Some of these blocks are valid, while others are invalid. One approach is to characterize the most recently used block of a branch as the only valid block of the specific branch. Only valid blocks contain IoT transaction data. Valid blocks can be merged using a branch merge algorithm to form a single blockchain. Such an approach for mining is used to support real-time data transactions, by avoiding the mining delay that exists in Proof-of-Work blockchain systems, such as Bitcoin. Characterization of valid blocks considers a Proof-of-Stake

[86] *IOTA Marketplace*. URL: https://data.iota.org/

[87] *Masked Authenticated Messaging*. URL: https://blog.iota.org/introducing-masked-authenticated-messaging-e55c1822d50e

mechanism. Chunks of IoT data, along with their hashes, are exchanged over the peer-to-peer network and are stored in the distributed data store between IoT devices. The forwarding of data chunks is based on the Chord protocol.

Flowchain uses W3C's WoT ontology as a standards-based model to represent a physical device as an application server, which can run on a high-performance device or a microcontroller. Moreover, a JavaScript runtime environment is used for high-performance devices, while a more lightweight environment is used for resource-constrained-devices. Finally, Flowchain supports a gateway mechanism for interoperability between different ledgers within the same IoT peer-to-peer network.

4.6 Trusted IoT Alliance

Cisco, Bosch, and a number of other major IoT vendors launched the Trusted IoT Alliance in September 2017 to catalyze the development of blockchain-enabled, trusted IoT. The mission of the alliance is to bring companies together to develop and set the standard for an open source, interoperable blockchain protocol to support different IoT ecosystems on a global scale.

Trust for data produced by such IoT systems is provided in a distributed ledger/blockchain agnostic fashion, thereby enabling a decentralized trust model for interoperable digitized identities of physical goods, documents, immobilized assets, sensors, and machines. The ambition is to achieve performance and resiliency that can scale to support billions of connected devices.

The alliance invites developers and enterprises to join and co-create PoCs and testbeds jointly with the partners onboard.

4.7 Samsung Nexledger

Samsung Nexledger is a permissioned blockchain platform targeting enterprises in different industry domains. Nexledger claims to already support the following use cases.

- Digital Identity: Solution that utilizes blockchain technology to create digital identities for customers
- Digital Payment: Blockchain-integrated solution that can supplement credit and debit cards for customer payments
- Digital Stamping: Blockchain-based solution that can be used for creating secure digitally stamped signatures without the need for a third-party authenticator

Support for new use cases can be added on as applications on the platform.

Nexledger claims support for the following features in their solution:

- Bolstered contingent measures with management monitoring of block information
- Reduced lead time with improvements in transaction verification and processing algorithm
- Reduced resource consumption with improvements in confirmation racing algorithm for Proof-of-Work

- Optimized management for the distributed ledger with multi-chains and partitioned chains
- Enhanced security with FIDO certifications and multiple biometric modalities

4.8 Datum

Datum is a decentralized and distributed marketplace for data monetization that leverages on an Ethereum-based smart contract platform. The platform allows anyone to securely and anonymously store and trade structured data collected from social networks, wearables, smart homes, and other IoT devices.

The Datum network claims to consist out of 3 billion DAT tokens, with 1.53 billion available in a public token crowdsale. Up to 40% of raised funds is hedged in USD/EUR/BTC. The Datum network aims to disrupt current data broker models by eliminating disintermediation in trading of social and IoT data.

Datum encompasses the following key elements:

- A decentralized data store allowing users to store structured data securely running on a smart contract blockchain.
- The DAT token enabling this data storage and sharing.
- A data marketplace, enabling individuals to monetize their data on their terms
- A mobile client application, available on Android and iOS.
- Datum leverages Ethereum, BigchainDB and IPFS to provide a scalable, decentralized data storage backend.
- Data storage and data sharing is paid for by the DAT token. Data can be purchased as one-off or on an ongoing subscription basis. Interestingly, trading of DAT Tokens is prohibited for U.S. Citizens and residents in China and South Korea.

4.9 KSI® Blockchain

KSI® Blockchain is a globally distributed infrastructure for the issuance and verification of KSI signatures. These signatures serve as a solid proof of *when* and *by whom* some data item of arbitrary size and format was generated. A user interacts with the KSI system by submitting the hash-value of the data to be signed. Then the KSI system returns to the user a signature providing cryptographic proof of the *time of signature*, the *integrity of the signed data*, and the *data origin*, i.e., which entity generated the signature.

Unlike traditional digital signature approaches, such as the Public Key Infrastructure (PKI), which depend on asymmetric key cryptography, KSI uses only hash-function cryptography. Thus, the generation and verification of KSI signatures is based solely on the security of hash-functions and the availability of a public ledger commonly referred to as a blockchain. KSI core technology is based on Buldas' and Saarepera's work [Bul+14].

The main benefits of the KSI system can be summarized as follows:

- **Unlimited Signing Rate:** KSI signatures can be generated at a practically unlimited rate. Even if an exabyte (10^{18} bytes or 1,000,000 terabytes) of data is generated around the planet per second, and data records have an average size of 1MB, KSI is able to provide signatures for all one trillion (10^{12}) data records per second with negligible computational, storage, and network overhead.
- **Portability:** The properties of the signed data can be verified even after that data has crossed geographic or organizational boundaries and service providers.
- **Data Privacy:** KSI does not ingest any customer data; data never leaves the customer premises. Instead the system is based on one-way cryptographic hash functions that result in hash values uniquely representing the data but are irreversible, such that one cannot start with the hash value and reconstruct the data. Data privacy is guaranteed at all times.
- **Quantum Immunity:** The cryptography behind the KSI signatures ensures that they never expire and remain quantum-immune, i.e., secure even after the realization of quantum computation.
- **Independent Verification:** The properties of the signed data can be verified without reliance on or need of a trusted authority.

4.9.1 KSI® Blockchain in Action

System logs are an essential component of modern information systems that are used for monitoring, debugging and for forensics to analyze incidents. We have integrated the rsyslog logging server with the KSI Blockchain via a plugin[88]. The logs are aggregated *before* signing, thus the KSI Blockchain round time does not affect performance. This plugin is used by several companies (including banks) for the integrity and authenticity of their logs.

Although we do sign healthcare data directly on the database side to provide integrity and authenticity for each individual record (i.e., that a record has not been tampered with or replaced by fake data), we have expanded the log signing use case even further by combining it with the database audit log. As the audit log is performed at the lowest logical level, monitoring operations on the data does not rely on the logging of complex systems built on top of the database and does not require any changes to these systems. As the database audit log provides information on data access too, this integration also serves as an ad-hoc asset for GDPR compliance.

4.10 AWS Blockchain Partners Portal

Amazon Web Service Blockchain Partners portal provides a wide range of capabilities and the largest global infrastructure for building end-to-end blockchain platforms cost efficiently and at large scale. APN Technology and Consulting partners offer a rapidly growing selection of blockchain and distributed-ledger solutions with support for multiple protocols.

Based on the experience of running different customer proof of concepts on the AWS infrastructure there are several key attributes to bring out, that the platform offers:

[88] *KSI*. URL: https://www.rsyslog.com/tag/ksi/

1) Easy to deploy development environment that most of the community working with Cloud infrastructure is familiar with. Sort of a standard that reduces the time of agreeing the basics when there are multiple partners developing the distributed ledger and blockchain side.

2) Scalability and vast amount of resources (pay as you grow) that can help a lot when testing out new limits in sharing information between nodes.

3) The open source code and support available through the AWS blockchain partners. The reuse of code can speed up the basic PoC concept building and enables more diversity for testing different customer use-cases.

Based on the AWS infrastructure the are several key partners that have developed their own blockchain technology solutions and distribute ledger capabilities that most effectively use AWS as IaaS. There is time needed that these AWS partners come out with industry production grade solutions and SaaS an PaaS the most developed partners are Kaleido[89], Corda R3[90] and PokitDok[91].

In order to give an overview how AWS environment with blockchain partner assisted code repository can be used there is one example in SOFIE Estonian energy pilot. The Corda R3 was used to build up distribute ledger and Corda Notar for data storage, sharing data between participating nodes and securing the access rights and history of the transactions made between participants. The scenario was to store and exchange 500 smart meters data stream (every 3-6 seconds) between two nodes and give access to the smart meter data to each other. The goal was to test how to handle smart meters that are deployed on a large scale with and take measures to avoid potential bottlenecks for information exchange and seamless integration with Transmission system operators and Distribution system operators specific databases

4.11 Cyber-Physical Chain (CPChain)

CPChain is a data platform for the IoT systems supporting data acquisition, storage, sharing and applications [LZS18]. It separates the data, contract, and application layers from the control, which supervises data interaction through a blockchain. Raw data is encrypted on the user side and stored in a distributed hash table (DHT). Only hash values, as unique identification of data, and credentials for integrity and correctness are published on the blockchain. Parallelization is supported by separating data storage from control. Re-encryption and homomorphic encryption technologies are combined to support one-to-many authorization.

CPChain plans to develop a hybrid consensus model that combines dynamic committee election with PoW consensus. The election process is performed in rounds. A node is eligible for election only if its credibility is above a threshold. However, how this node credibility is determined and updated is not specified.

To address the different requirements of IoT applications in terms of delay and security, sidechains implementing lightweight consensus protocols are proposed. For industrial scenarios, altruistic cooperative models are proposed.

[89] *Kaleido*. URL: https://kaleido.io/
[90] *Corda R3*. URL: https://www.corda.net/
[91] *PokitDok*. URL: https://pokitdok.com/

5. Privacy, Decentralised Identities, and Vulnerabilities

IoT systems federated by SOFIE will handle large amounts of personal data. In addition, a large number of identities will be used for both users and devices. These issues raise privacy concerns. Data management should be designed in a way that promotes the system's scalability while ensuring users' privacy.

This section covers privacy issues of modern decentralized systems, concentrating on identity management solutions. Furthermore, blockchain-related security attacks and blockchain simulators are discussed.

5.1 Privacy Issues

This section first describes the EU's General Data Protection Regulation (GDPR), which forms the basis of handling personal data. Afterwards, the MyData approach for personal data management and most promising decentralized identity technologies are covered. Finally, privacy-related attacks of decentralized systems are discussed.

5.1.1 Legal bases for processing personal data (GDPR, etc.)

The processing and use of personal data is increasingly being controlled by legislation throughout the world. In the EU, the new General Data Protection Regulation (GDPR), enforced in May 2018, sets some of the strictest requirements in the field. It is based on the EU's strong fundamental rights protection, particularly for privacy, as unlimited collection and use of information on individuals can have negative consequences both for the individuals themselves and for the society as a whole. It will also have a global impact, as GDPR extends beyond EU borders, namely to all services and goods whose data processing concerns EU citizens.

The GDPR defines personal data as "any information relating to an identified or identifiable natural person" (Art. 4 (1)[92], which covers a significant part of all information due to the inclusion of the word "identifiable". The GDPR also defines a number of legal roles relating to personal data processing, all of which are subject to differing rights and obligations. Data subject refers to the natural person to whom the personal data relates. Data controllers and data processors, on the other hand, can be either physical or legal persons, public authorities, agencies, or other bodies (Art. 4 (7)-(8)). A data controller is the entity which "alone or jointly with others determines the purposes and means of the processing of personal data", whereas the data processor "processes personal data on behalf of the controller". The notion of *processing* is understood widely: according to Art. 4 (2) GDPR, processing signifies "any operation or set of operations which is performed upon personal data or sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction". In addition, all personal data processing operations have to adhere to the general principles of data protection (for instance, the principles of limited collection or purpose limitation) as outlined in national and international data protection conventions and regulations.

[92] Regulation (EU) 2016/679 of the European Parliament and of the Council. Official Journal of the European Union L119:1–88

As any collection and processing of personal data is viewed to interfere with the related individual's rights, the GDPR presents an exhaustive list of the six legal bases for processing personal data. Data processing is legal only if: (i) it is based on an individual's consent, (ii) necessary for the performance of a contract, (iii) justified in the legitimate interest of a controller, as well as (iv) if processing is in the vital interest of the data subject, (v) in public interest, or (vi) for the fulfillment of a legal obligation.

According to the GDPR, consent means an indication of the data subject's wishes by which they signify agreement to the processing of their personal data, either by statement or by "clear affirmative action". Compared to previous EU personal data legislation, the GDPR sets stricter requirements for the validity of a consent: it must be freely given, specific, informed, and unambiguous. Therefore, consent should not be used when individuals have no genuine free choices, e.g., when there is a significant power imbalance between a data subject and a controller, or where consent to personal data processing is made conditional for the provision of a service.

Article 7 of the GDPR provides the framework for utilising consent. Firstly, the controller must be able to demonstrate the existence of consent. Secondly, in the context of written declarations containing also other matters, consenting must be clearly distinguishable, accessible, and understandable in order to be valid. Thirdly, the data subject can withdraw their consent at will, and this must be as easy as consenting. Fourthly, in assessing the free nature of consent, particular account is to be taken of whether the performance of a contract, including the provision of a service, is made conditional on the consent to processing of data, which is not necessary for the performance of the contract. The GDPR also requires parental oversight for children's consent (below the age of 13-16, depending on national definition) in cases where information society services are directly offered to children. Furthermore, some special categories of personal data require stronger justifications, including explicit consent for processing.

Consenting is also a separate action from agreeing to a contract. If only a contract is made, personal data can be processed only as far to what is necessary for the performance of the contract. Another option for processing personal data on a lawful basis is by referring to a legitimate interest of a data controller. This means that personal data can be processed when there is a present, lawful, and clearly specified interest of a controller, and the personal data processing does not disproportionally interfere with individuals' rights. Legitimate interest requires sound justification and reasoning as well as a sophisticated balancing test on the side of the controller. Personal data can also be processed if this is of vital interest (in life-or-death scenarios) to the data subject or other natural persons. Finally, personal data can be processed by a public authority in case of justified public interest.

The GDPR also brings many rights to the data subject, including data portability and the right to be forgotten. Data portability (Art. 20 (1)) means that the data subject has "the right to receive the personal data concerning him or her, which he or she has provided to a controller, in a structured, commonly used and machine-readable format" and "the right to transmit those data to another controller without hindrance from the controller to which the personal data have been provided". This right should make it much easier for the individual to reuse existing data in other services. Finally, the right to be forgotten (Art. 17 (1)) means that the data subject has the right to have the personal data relating to them erased, for instance when the consent is withdrawn or when the data is no longer required – a requirement which means that storing personal data on an immutable platform, such as a blockchain, can be quite problematic and should, consequently, be avoided.

5.1.2 Introduction to MyData and related solutions for privacy management

MyData is a human-centered approach in personal data management that aims to combine industry needs for data with digital human rights. The core idea in MyData is to let individuals be in control of their own data – give people an easy way to see where their data goes, to specify who can use it, and to alter these decisions over time. MyData's vision is that with the right technologies in place the human-centric approach would simplify data flows and open new opportunities for businesses to develop innovative personal-data based services while preserving privacy and data protection.

Figure 7 positions the MyData concept with respect to the personal data and data protection dimensions.

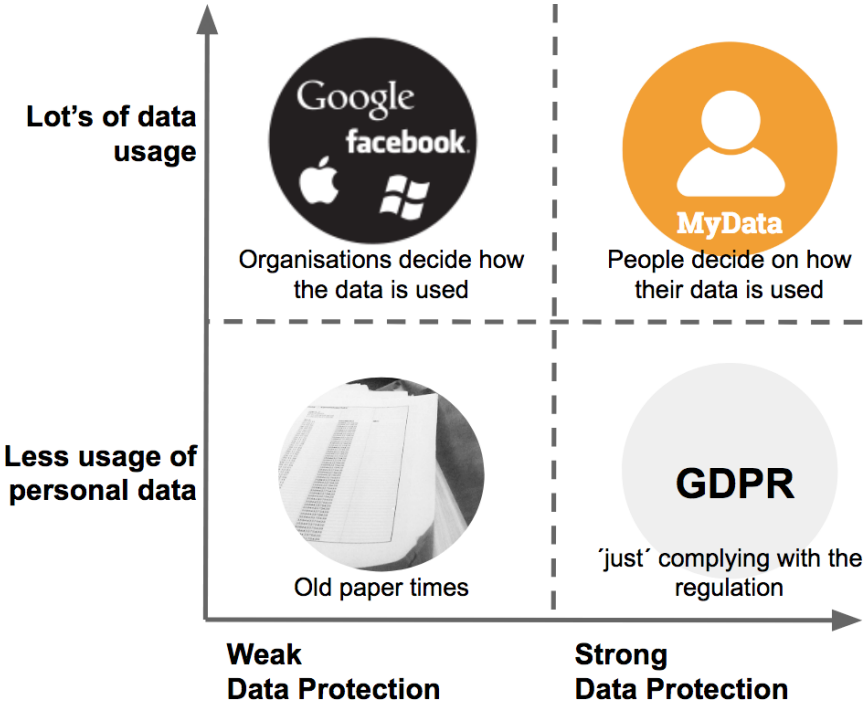

Figure 7: MyData concept in the personal data / data protection field

In the core of traditional data-protection thinking lies the notion that all personal data collection may potentially harm the privacy of individuals – "the less data collected, the smaller the risk". In contrast, industry highlights the benefits that the collection and use of personal data could provide. Solving this perceived contradiction between data protection and data utility is one of the main shifts that the MyData community aims at. The MyData Declaration[93] specifies: *"Data protection regulation and corporate ethics codes are designed to protect people from abuse and misuse of their personal data by organisations. While these will remain necessary, we intend to change common practices towards a situation where individuals are both protected and empowered to use the data that organisations hold about them. Examples of such uses include simplifying administrative paperwork, processing data from multiple sources to improve one's self-knowledge, personalised AI assistants, decision-making, and data sharing under the individual's own terms."*

[93] *MyData Declaration*. URL: https://mydata.org/declaration

The MyData concept of human-centric personal-data management has been developed collaboratively in a loosely coupled MyData Global Network. The international network was born out of the Personal Information Management Services (PIMS) industry round table organised by the European Commission in the end of 2015. After the initial meeting, Aalto University and FING have facilitated several follow-up meetings for the key stakeholders and have organised larger international MyData conferences[94,95]. In the last MyData conference in 2017 the network published the MyData Declaration, which describes the shared understanding of the high level ideology.

5.2 Decentralised Identity Technologies

Traditionally, most of the identifiers used by both individuals and devices to operate with networked services have been issued by a central party, e.g., the employer of an individual or the manufacturer or administrator of a device. Lately, an increasingly popular type of issuers of identifiers are social networks and other large internet companies, such as Facebook and Google, which can simplify the individual's process of using multiple services by federating the use of identifiers, but at the cost of ceding much control over the identifier and how it is being used to the issuing organisation.

The movement of Decentralised Identifiers[96] (DIDs) has emerged as the counterforce. It aims to make the owner of the identifier its sole controller, thus creating self-sovereign identities. This is achieved by building the system on a distributed ledger or a blockchain, a technology that provides the necessary immutable registry of identifiers and related information (e.g., attributes describing the owner of the identifier or technical information related to using the identifier), without relinquishing control of the platform to any single party.

Currently, there are dozens of initiatives of launching decentralised identifiers[97]. Some build their solutions by adapting existing blockchains, such as Bitcoin and Ethereum, while others build new dedicated blockchains. All major initiatives are part of the Decentralized Identity Foundation[98]. Many initiatives also support using verifiable credentials[99], an upcoming standard from W3C, or similar mechanisms to express and utilise the attribute data. Some systems even support zero-knowledge proofs, a privacy-enhancing way of using (suitably constructed) credentials so that only the requested information is revealed, while keeping all other information in the credentials hidden.

There are so far two public studies evaluating this plethora of (oftentimes self-claimed) self-sovereign identities. The first one, an Austrian Self-Sovereign Identity evaluation [Abr17] from December 2017, compares five identity technologies (Sovrin, Blockstack [Ali+17], Multichain, Ethereum and uPort), and concludes that "Sovrin is the most promising technology because of its key management, no expensive Proof-of-Work has to be calculated and the support of identity data import even though some trust is required. All other technologies require the Proof-of-Work calculation, which is a huge disadvantage. Besides Sovrin is uPort the only other technology that supports identity data import".

[94] *MyData Conference 2016*. URL: https://mydata2016.org
[95] *MyData Conference 2017*. URL: https://mydata2017.org
[96] *Decentralized Identifiers*. URL: https://decentralized.id
[97] *Peacekeeper*. URL: https://github.com/peacekeeper/blockchain-identity
[98] *Identity Foundation*. URL: http://identity.foundation
[99] *Verifiable Credentials Data Model*. URL: https://w3c.github.io/vc-data-model

The second one, a more extensive evaluation [KM18] from ING in May 2018 goes over 50 identity technologies, of which it shortlists only nine (Blockauth, Cryptid, DIMS, Sovrin, IPv8, IRMA, Selfkey, OpenID Connect, and uPort) for a detailed evaluation as they are the only ones that meet the three key self-sovereignty principles[100], namely *Control*, *Transparency*, and *Portability*. The detailed evaluation, which adds the rest of Allen's self-sovereignty principles, as well as support for attribute life-cycle and implementation success criteria, concludes that only three technologies, Sovrin, uPort, and IRMA, are worth recommending. Of these, IRMA is currently not based on distributed ledgers, although this may change in future releases. Finally, one identity technology with their own identity-tailored blockchain, Veres One, which was introduced too recently to be included in the aforementioned comparisons, appears to have the relevant properties, and thus deserves a further look.

Based on the evaluations and remembering the MyData guiding principles and practical implementability of the solutions, this document will focus on the following three technologies: *Sovrin*, *uPort*, and *Veres One*. The following table summarises some of the key differences between the solutions:

	Sovrin	**uPort**	**Veres One**
Permissioned / Permissionless	Permissioned (= requires trust in node selection)	Permissionless (= requires mining)	Permissionless (= requires mining)
Selective disclosure	Yes	Yes	No
Interoperability	Support for W3C DIDs and verifiable credentials on the roadmap. Zero-Knowledge Proofs and supporting credentials are Sovrin-specific.	Complies with W3C DIDs and verifiable credentials	Complies with W3C DIDs and verifiable credentials
Cost of an identifier	?	$ 5-10 per DID	$ 2 per DID

Devising evaluation criteria for assessing the scalability and applicability of identity technologies is clearly a key component to taking the right design decisions. This need is pushed even further when IoT ecosystems become massive, and when individuals are incentivized to issue distinct identifiers per linked connection to protect their privacy.

To be able to scale to a global solution, the identifier solution would have to be able to handle 10^9 identifiers for a moderately successful global system and 10^{12} identifiers for a highly successful system. Further, each identifier has multiple transactions over its lifetime: the identifier is created, the related cryptographic keys can be rotated multiple times, the related information can be repeatedly updated, and finally the identifier is revoked. Each transaction requires a separate write to the ledger. To appreciate the magnitude of the issue, let us look at identifier creation: a typical identifier creation requires ca. 0,5 kB of information to be written on the ledger, so creating 10^9 identifiers (a moderately successful global system) would require writing 0,5 TB of data to the ledger, and creating 10^{12} identifiers results in 500TB of data. Using Bitcoin (with the

[100] *Self-Sovereign Identity Principles*. URL: https://github.com/WebOfTrustInfo/self-sovereign-identity/blob/master/self-sovereign-identity-principles.md

throughput of 1 MB every 10 minutes) as an example ledger, it would take 9,5 years and 9500 years, respectively, just to create the identifiers - clearly the throughput is not enough considering all the other DID transactions also required. Furthermore, some systems use the ledger to store other types of information further exacerbating the issue.

Applicability includes a large number of criteria for the identifier solution. Some are technical and relate to the IoT devices' or other agent-carrying mobile devices' limited resources, the ability to operate with legacy solutions etc. Another group of criteria are financial and relate to topics such as the cost of a transaction. This is strongly related to the consensus mechanism used in the system. For instance, solutions relying on Proof-of-Work to establish consensus will naturally gravitate towards higher transaction costs. Yet another group relates to the performance of the system including the latency of transactions. Currently, Bitcoin transactions can have a latency of over an hour, which naturally precludes application areas where transactions have to complete within seconds. Zero-knowledge proof creation requires heavy resources too, rendering them inadequate for some real-world use case scenarios.

The criteria about support infrastructure relate both to the technical ledger infrastructure and the organisations and trust frameworks required to maintain a smoothly running identifier system.

5.2.1 Sovrin

Sovrin[101] is a decentralised identity network for self-sovereign identities. Sovrin uses *Hyperledger Indy*[102] as its underlying distributed ledger. Indy is a permissioned ledger that relies on the Plenum[103] protocol to reach consensus. All verifier nodes used to run the Plenum-protocol (called *steward nodes*) are chosen by the Sovrin Foundation, which is responsible for governing the Sovrin network.

Sovrin has a strong focus on privacy and on minimising correlation attacks. In addition to verifiable credentials it also supports zero-knowledge proofs that enable one to prove things about oneself without revealing the credentials used to prove the property. Both credentials and proofs are currently specific to the Indy/Sovrin ecosystem, though support for W3C DIDs and verifiable credentials is on the Indy project's roadmap for later on in 2018. As a privacy enhancing principle, individuals are assumed to be using multiple Sovrin identifiers (one for each party they are operating with) to reduce the chance of correlation attacks.

Sovrin is currently only an emerging technology as all basic functionality is not yet available and the Sovrin Foundation is only running a provisional Sovrin network.

5.2.2 Veres One

Veres One[104] aims at providing a globally interoperable blockchain for identity. It builds on a new identity-specific permissioned blockchain that uses the *Continuity* consensus algorithm. Veres One DIDs[105] and verifiable credentials are based on the W3C standards.

[101] *Sovrin*. URL: https://sovrin.org
[102] *Hyperledger Indy*. URL: https://www.hyperledger.org/projects/hyperledger-indy
[103] *Plenum*. URL: https://github.com/hyperledger/indy-plenum
[104] *Veres One*. URL: https://veres.one
[105] *Veres One DIDs*. URL: https://w3c-ccg.github.io/didm-veres-one

Veres One is an emerging technology and currently the network is running in a pre-production mode. Veres One claims that it will have higher throughput (x10), lower latency (x1/10), and cheaper DIDs (x1/5) than Ethereum-based solutions, such as uPort, though these estimates are based on benchmarks by Veres One and remain to be verified once the network has entered production.

5.2.3 uPort

uPort[106] uses the Ethereum blockchain and IPFS as the underlying technical solution, making it a permissionless solution. uPort DIDs are essentially Ethereum addresses. uPort's design target has been to minimise the footprint on the blockchain. uPort users will have fewer identifiers (DIDs) compared to systems like Sovrin, but it is possible to create identities for different roles (work, free time, etc.). Anonymity is improved by uPort's design decision not to store any personally identifiable attributes in the the system. uPort supports W3C's DIDs and verifiable credentials.

uPort is more established than the other two technologies and has focused on also providing supporting elements, such as identity and messaging protocols. The uPort Identity Protocol describes a generalized identity model capable of expressing natural or legal persons, applications and IoT devices, while the uPort Claims Protocol describes a standard message format that enables source attribution, facilitating interoperability between various blockchains and identity networks.

5.3 Vulnerabilities

This section discusses potential vulnerabilities with respect to privacy, to the consensus mechanism, and to smart contracts.

5.3.1 Vulnerabilities in Privacy

Blockchain users are usually identified by public keys. In theory, it should not be possible to map a public key to a real user, nevertheless in practice this is possible in many cases, e.g., by utilizing off-network information, by performing TCP/IP-layer attacks [Kam11], or by analyzing P2P traffic [KKM14]. Using anonymizing networks, such as *Tor*[107], is a possible solution, nevertheless, they do not always provide the necessary bandwidth and an attacker can trigger a ban of Tor connections to the Bitcoin network [BKP14]. In order to mitigate this privacy threat, users often result in creating multiple public/private key pairs. However, Reid and Harrigan [RH11], demonstrated that by monitoring the transactions graph, and especially those transactions that have multiple inputs, it is possible to identify keys belonging to the same user. A number of follow-up research works used similar techniques and achieved the same results [TS16]. Another solution to the privacy problem is the so called "laundry service", which exchanges different users' bitcoins. These services, however, have severe limitations [Mie+13]: operators can steal funds, track coins, or simply go out of business, taking users' funds with them. Perhaps in recognition

[106] *uPort*. URL: https://www.uport.me
[107] *The Tor Project*. URL: https://www.torproject.org/

of these risks, many services offer short laundering periods, which lead to minimal transaction volumes and hence to limited anonymity.

Blockchains are vulnerable to various kinds of attacks. These include general cryptography-related issues, such as weaknesses in the underlying cryptographic hash and digital signature functions, as well as weaknesses in the cryptographic protocol used (how to apply cryptographic primitives in a secure way). In general, these issues are well documented and researched, and it is unlikely that major blockchains will suffer from these in the near future, unless a weakness is found in a major algorithm, such as SHA256. Finally, blockchain innovation in distributed consensus mechanisms and smart contracts may also create new types of vulnerabilities.

5.3.2 Vulnerabilities in Consensus Mechanisms

Distributed, permissionless blockchains based on Proof-of-Work are vulnerable to the "51% attack", where the malicious entity gains the majority of computational resources of the network. In such a situation, the attacker may double-spend tokens, prevent transactions from other users from getting through, and cause arbitrary disturbance to the blockchain network. While blockchain solutions may include various means to lessen the impact of the "51% attack", such an attack has the potential to cause a significant disturbance to the day-to-day operation of the network. In some cases even a smaller share of the network's resources may be enough for launching an attack [ES14].

The 51% attack can be executed in various ways. If the target blockchain is relatively small, the attacker can gain control of the network with a modest investment in mining hardware. For larger blockchain networks, directly deploying sufficient computational resources to launch a 51% attack may not be feasible. In this case network- and application-level attacks against the blockchain network can be used, including hijacking existing mining nodes (and therefore gaining their resources) and launching denial-of-service attacks against the mining nodes or their networks to decrease the overall capacity of the target blockchain network.

While attacks against well-established blockchains may seem too expensive to carry out, they can actually be very cost-effective if the target blockchain is used for critical purposes (business processes of marge companies, basic infrastructure of countries such as utilities, electronic payments, etc.). For state actors, conventional warfare is very expensive both in terms of money and diplomatic consequences, therefore spending billions of euros on cyber attacks against a major adversary is inexpensive in comparison. Criminal organizations could use such attacks to profit through blackmailing, while major corporations would have an incentive to disrupt the operations of their competitors.

5.3.3 Vulnerabilities in Smart Contracts

Smart contracts (Section 2.2.2) are computer programs running inside the blockchain (i.e., executed by blockchain miners). Since the execution duration of a program cannot be accurately estimated in advance, smart contracts could invite denial-of-service (DoS) attacks, where the attacker drains miners' resources by submitting a resource-intensive smart contract. Blockchains offering smart contracts require a separate payment to execute the smart contract in order to alleviate the risks of DoS attacks, however such mechanisms do not always work perfectly.

Ethereum is the most popular blockchain that supports smart contracts. It charges for the use

of miners' resources, introducing the term "gas", a metric for the amount of resources used by a smart contract invocation. In order to invoke a smart contract, the caller has to pay for the respective gas in ether, Ethereum's internal currency. Gas price is not fixed. Instead, it is shaped by the supply-and-demand law of free markets. Several DoS attacks against Ethereum smart contracts have been identified and executed [Che+17], [ABC17]. The Ethereum foundation has responded by adjusting the estimation of gas to better reflect the resources used.

6. Conclusions

We provided an overview of the State of the Art in blockchains and Distributed Ledger Technologies (DLTs) with the intent of using them to support trust and automatic operations in the Internet of Things (IoT). For this reason, we also briefly reviewed IoT characteristics and architectures and IoT systems that have embraced DLTs, as well as security and privacy aspects.

DLTs provide an append only, tamper-proof log of entries typically representing transactions. Trust is supported through the immutability of the entries and the consensus mechanism, usually in Byzantine fault-tolerant manner over a large network of nodes, rather than centrally through a single or a small group of institutions. Blockchains are one type of distributed ledgers where data records are grouped into blocks that are linked through cryptographic hashes, thus forming a chain of blocks. The main elements of a blockchain are the consensus mechanism and the programming language potentially supported; the latter can range from simple scripting to more powerful smart contract programming languages that support Turing-complete computation, can maintain state, and support the interaction with other contracts. Blockchain systems can belong to two broad categories that differ in the policy that defines which nodes can participate in the blockchain's distributed network and the roles that they can perform. Noteworthy blockchain systems that have been presented here include Bitcoin, Ethereum, Cardano, IOTA, which are public (open) blockchains, and Hyperledger Fabric, MultiChain, and Corda, which are permissioned (private) blockchains. The consensus mechanism is central in providing the guarantees and properties of the DLT and is still a topic of active research. Given the energy hungry and slow consensus mechanism of the Bitcoin blockchain, it is important for IoT applications and now it appears possible, that new consensus mechanisms and permissioned blockchains, in addition to sidechains and other interledger approaches, allow different trade-offs to effectively address these two key problems.

Interledger approaches and mechanisms have been developed to connect different DLTs and payment networks because no one technology seems to prevail and new ones supporting new features and benefits continue being introduced. Interledger approaches differ in whether they provide support for trading value between two blockchains, in which case the total amount of value in each blockchain remains the same, or whether they support the transfer of value among blockchains. Moreover, different interledger approaches have varying support for transferring information across different blockchains. Atomic cross-chain transactions form an important subclass of interledger approaches and are based on more basic mechanisms, namely hashlocks and timelocks. Atomic cross-chain transactions allow trading of value across blockchains, between two parties that do not trust each other, without requiring the presence of a trusted third party. Sidechains enable the transfer of value from one blockchain to another. The main motivation for implementing sidechains is to achieve higher performance, such as lower block confirmation times, lower transaction costs, or to support for more flexible smart contracts, which are not supported by the main chain. An important advantage of sidechains is that they can offload the main chain from handling all transactions, hence can enhance the scalability of blockchains. One approach for implementing sidechains is based on the federated peg, which relies on semi-trusted functionaries that achieve agreement through Byzantine consensus. Another approach for implementing sidechains, on which Etherium's Plasma proposal is based, is to create a hierarchical tree of sidechains. Each sidechain can be governed by its own set of rules and constraints, while full security is provided only by the root chain. Bridging approaches typically support the transfer of data, in addition to the transfer of value between blockchains. Unlike bridging, ledger-of-ledger approaches rely on a super ledger for interconnecting sidechains or

other ledgers. Proposals such as the Lightning Network and Raiden seek to increase the scalability of blockchains by performing specific transactions among two parties off-chain; moreover, they support transferring payments across a network of payment channels between entity pairs. Payments across a network of DLT systems is also the focus of the Inter-ledger Protocol (ILP), which aims to provide a coherent, standards-based payment infrastructure. Finally, smart contracts (on blockchains) cannot directly interface with the outside world. This interfacing of blockchains with the outside world is provided by oracles that can obtain data from external sources or call external APIs, which we also discuss in the first section. Note that our emphasis on cross-chain transactions, interledger operations, and ILP in particular is not accidental. It is a critical area for our work in SOFIE for various reasons: first, in order to realize an open federation rather than dictate or expect a specific DLT; second, it is important to interface with the real-world and external systems; and third, it was a key choice from the beginning, since it provides a level of technology independence and future-proofs our approach, by allowing the selection of different DLT solutions to different domains based on their specific requirements and by enabling the slow migration to newer DLTs with new features.

Reviewing next IoT technologies, one realizes immediately that there is a large number of IoT platforms, many proprietary and either fully or partially closed. There have been various attempts for defining or describing in their generality IoT architectures and systems using a layered and modular approach for facilitating interoperability. There are also proposals that aim to achieve interoperability at a higher, semantic layer, compared to technical (connectivity-oriented) and syntactic interoperability, in order to enable different entities to exchange information, data, and knowledge in a meaningful way. Some IoT systems have already adopted blockchain technology. We discussed such systems focusing in particular on how they support IoT data and transactions. These systems include industrial solutions, but also blockchain-community originating proposals that specifically target IoT data handling.

Finally, we discussed privacy related issues, including the General Data Protection Regulation (GDPR) and the MyData approach, especially those that arise when a large amount of data needs to be handled from different sources. Decentralized identity solutions are necessary for large distributed systems with key evaluation criteria including the trust model, degree of interoperability, and cost. Sovrin is an ambitious and promising emerging approach in this area, based on a permissioned blockchain (Hyperledger Indy), but other efforts exist and were also presented. We concluded this Section with a discussion of vulnerabilities that target specifically the consensus mechanism of blockchains.

Open blockchains are important for the creation of open 4th generation business platforms, where any player can participate without requiring permission from anyone to do so and without any one player having a dominant position in the platform (except if they obtain control of the majority of the network nodes). The path to realizing this aspect seems clear, even though many details remain to be sorted out. Among the key issues to be worked out are the automatic realization of semantic interoperability over multiple domains and the trade-off between privacy and transparency. Even less developed is the issue of the distribution to the involved parties of the added value created through such platforms, which will be a strong incentive for cooperation and their long-term stability.

References

[ABC17] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. "A Survey of Attacks on Ethereum Smart Contracts SoK". In: *Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204*. New York, NY, USA: Springer-Verlag New York, Inc., 2017, pp. 164–186. ISBN: 978-3-662-54454-9. DOI: 10.1007/978-3-662-54455-6_8. URL: https://doi.org/10.1007/978-3-662-54455-6_8.

[Abr17] Andreas Abraham. *Self-Sovereign Identity*. 2017. URL: https://www.egiz.gv.at/files/download/Self-Sovereign-Identity-Whitepaper.pdf.

[Aga+16] Rachit Agarwal, David Gomez Fernandez, Tarek Elsaleh, Amelie Gyrard, Jorge Lanza, Luis Sanchez, Nikolaos Georgantas, and Valerie Issarny. "Unified IoT Ontology to Enable Interoperability and Federation of Testbeds". In: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)(WF-IOT)*. Dec. 2016, pp. 70–75. DOI: 10.1109/WF-IoT.2016.7845470. URL: http://doi.ieeecomputersociety.org/10.1109/WF-IoT.2016.7845470.

[Ali+16] Muneeb Ali, Jude C Nelson, Ryan Shea, and Michael J Freedman. "Blockstack: A Global Naming and Storage System Secured by Blockchains." In: *USENIX Annual Technical Conference*. 2016, pp. 181–194.

[Ali+17] Muneeb Ali, Ryan Shea, Jude Nelson, and Michael J. Freedman. *Blockstack: A New Internet for Decentralized Applications*. Version 1.1. Oct. 2017. URL: https://blockstack.org/whitepaper.pdf.

[And+18] Elli Androulaki et al. "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains". In: *Proceedings of the Thirteenth EuroSys Conference*. EuroSys. Porto, Portugal: ACM, 2018, 30:1–30:15. ISBN: 978-1-4503-5584-1. DOI: 10.1145/3190508.3190538. URL: http://doi.acm.org/10.1145/3190508.3190538.

[Ark] Ark. *ARK Whitepaper: A Platform for Consumer Adoption*. URL: https://ark.io/Whitepaper.pdf.

[Atk+17] Rob Atkinson, Raúl García-Castro, Joshua Lieberman, and Claus Stadler. *Semantic Sensor Network Ontology*. Ed. by Armin Haller, Krzysztof Janowicz, Simon Cox, Danh Le Phuoc, Kerry Taylor, and Maxime Lefrançois. Dec. 2017. URL: https://www.w3.org/TR/vocab-ssn/.

[Bac+14] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. *Enabling Blockchain Innovations with Pegged Sidechains*. 2014. URL: http://www.blockstream.com/sidechains.pdf.

[Baj+17] Garvita Bajaj, Rachit Agarwal, Pushpendra Singh, Nikolaos Georgantas, and Valérie Issarny. "A study of existing Ontologies in the IoT-domain". In: *CoRR* abs/1707.00112 (2017).

[Bar+12] Payam Barnaghi, Wei Wang, Cory Henson, and Kerry Taylor. "Semantics for the Internet of Things: Early Progress and Back to the Future". In: *Int. J. Semant. Web Inf. Syst.* 8.1 (Jan. 2012), pp. 1–21. ISSN: 1552-6283. DOI: 10.4018/jswis.2012010101. URL: http://dx.doi.org/10.4018/jswis.2012010101.

[Bas+16] Louay Bassbouss et al. *Semantic Interoperability for the Web of Things*. Tech. rep. Aug. 2016. DOI: 10.13140/RG.2.2.25758.13122.

[Ber+17] Maria Bermudez-Edo, Tarek Elsaleh, Payam Barnaghi, and Kerry Taylor. "IoT-Lite: A Lightweight Semantic Model for the Internet of Things and Its Use with Dynamic Semantics". In: *Personal Ubiquitous Comput.* 21.3 (June 2017), pp. 475–487. ISSN: 1617-4909. DOI: 10.1007/s00779-017-1010-8. URL: https://doi.org/10.1007/s00779-017-1010-8.

[BKP14] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. "Deanonymisation of Clients in Bitcoin P2P Network". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS. Scottsdale, Arizona, USA: ACM, 2014, pp. 15–29. ISBN: 978-1-4503-2957-6. DOI: 10.1145/2660267.2660379.

[Bul+14] Ahto Buldas, Ahto Truu, Risto Laanoja, and Rainer Gerhards. "Efficient Record-Level Keyless Signatures for Audit Logs". In: *Secure IT Systems*. Springer International Publishing, 2014, pp. 149–164. ISBN: 978-3-319-11599-3.

[But13] Vitalik Buterin. *A Next-Generation Smart Contract and Decentralized Application Platform*. Nov. 2013. URL: https://github.com/ethereum/wiki/wiki/White-Paper.

[But16] Vitalik Buterin. *Chain Interoperability*. Sept. 2016.

[C+99] Miguel Castro, Barbara Liskov, et al. "Practical Byzantine fault tolerance". In: *OSDI*. Vol. 99. 1999, pp. 173–186.

[Che] Jollen Chen. *Flowchain: A Distributed Ledger Designed for Peer-to-Peer IoT Networks and Real-time Data Transactions*. URL: https://www.flowchain.co/Flowchain-WhitePaper.pdf.

[Che+] Zhi-dong Chen, Zhuo Yu, Zhang-bo Duan, and Kai Hu. "Inter-Blockchain Communication". In: 2nd International Conference on Computer Science and Technology (CST 2017). URL: http://dpi-proceedings.com/index.php/dtcse/article/download/12539/12074.

[Che+17] Ting Chen, Xiaoqi Li, Ying Wang, Jiachi Chen, Zihao Li, Xiapu Luo, Man Ho Au, and Xiaosong Zhang. "An Adaptive Gas Cost Mechanism for Ethereum to Defend Against Under-Priced DoS Attacks". In: *Information Security Practice and Experience*. Ed. by Joseph K. Liu and Pierangela Samarati. Springer International Publishing, 2017, pp. 3–24. ISBN: 978-3-319-72359-4. DOI: 10.1007/978-3-319-72359-4_1.

[CM] Arlyn Culwick and Dan Metcalf. *The Blocknet Design Specification*. URL: https://www.blocknet.co/wp-content/uploads/2018/04/whitepaper.pdf.

[Cro+16] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. "On Scaling Decentralized Blockchains". In: *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer, 2016, pp. 106–125. ISBN: 978-3-662-53357-4.

[Dil+16] Johnny Dilley, Andrew Poelstra, Jonathan Wilkins, Marta Piekarska, Ben Gorlick, and Mark Friedenbach. "Strong Federations: An Interoperable Blockchain Solution to Centralized Third Party Risks". In: *CoRR* abs/1612.05491 (2016). arXiv: 1612.05491. URL: http://arxiv.org/abs/1612.05491.

[EOA16] S. Matthew English, Fabrizio Orlandi, and Sören Auer. "Disintermediation of Inter-Blockchain Transactions". In: *CoRR* abs/1609.02598 (2016). arXiv: 1609.02598. URL: http://arxiv.org/abs/1609.02598.

[ES14] Ittay Eyal and Emin Gün Sirer. “Majority Is Not Enough: Bitcoin Mining Is Vulnerable”. In: *Financial Cryptography*. Vol. 8437. Lecture Notes in Computer Science. Springer, 2014, pp. 436–454. ISBN: 978-3-662-45471-8. DOI: 10.1007/978-3-662-45472-5_28. URL: https://arxiv.org/pdf/1311.0243.pdf.

[Gan+17] Maria Ganzha, Marcin Paprzycki, Wiesław Pawłowski, Paweł Szmeja, and Katarzyna Wasielewska. “Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective”. English. In: *Journal of Network and Computer Applications* 81.Complete (2017), pp. 111–124. DOI: 10.1016/j.jnca.2016.08.007.

[Gan+18] Maria Ganzha, Marcin Paprzycki, Wiesław Pawłowski, Paweł Szmeja, and Katarzyna Wasielewska. “Towards Semantic Interoperability Between Internet of Things Platforms”. In: *Integration, Interconnection, and Interoperability of IoT Systems*. Ed. by Raffaele Gravina, Carlos E. Palau, Marco Manso, Antonio Liotta, and Giancarlo Fortino. Cham: Springer International Publishing, 2018, pp. 103–127. ISBN: 978-3-319-61300-0. DOI: 10.1007/978-3-319-61300-0_6. URL: https://doi.org/10.1007/978-3-319-61300-0_6.

[Gmb16] IoT Analytics GmbH. *IoT Platforms: Market report 2015–2021 / IoT Analytics*. Jan. 2016. URL: https://iot-analytics.com/product/iot-platforms-market-report-2015-2021-3/.

[H2016] H2020 UNIFY-IoT. *D3.1 Report on IoT platform activities*. Sept. 2016.

[Her18] Maurice Herlihy. “Atomic Cross-Chain Swaps”. In: *CoRR* abs/1801.09515 (2018). arXiv: 1801.09515. URL: http://arxiv.org/abs/1801.09515.

[Hyu17] Hyundai Corporation. *Hdac: Transaction Innovation - IoT Contract & M2M Transaction Platform based on Blockchain, Whitepaper*. Nov. 2017. URL: https://github.com/Hdactech/doc/wiki/Whitepaper.

[Jar+14] Antonio J. Jara, Alex C. Olivieri, Yann Bocchi, Markus Jung, Wolfgang Kastner, and Antonio F. Skarmeta. “Semantic Web of Things: An Analysis of the Application Semantics for the IoT Moving Towards the IoT Convergence”. In: *Int. J. Web Grid Serv.* 10.2/3 (Apr. 2014), pp. 244–272. ISSN: 1741-1106. DOI: 10.1504/IJWGS.2014.060260. URL: http://dx.doi.org/10.1504/IJWGS.2014.060260.

[Kam11] Dan Kaminsky. *Black Ops of TCP/IP Presentation*. 2011. URL: https://dankaminsky.com/2011/08/05/bo2k11.

[KK18] Sebastian Kaebisch and Takuki Kamiya, eds. *Web of Things (WoT) Thing Description*. Apr. 2018. URL: https://www.w3.org/TR/wot-thing-description.

[KKD17] Kazuo Kajimoto, Matthias Kovatsch, and Uday Davuluru. *Web of Things (WoT) Architecture*. Mar. 2017. URL: https://w3c.github.io/wot-architecture.

[KKM14] Philip Koshy, Diana Koshy, and Patrick McDaniel. “An Analysis of Anonymity in Bitcoin Using P2P Network Traffic”. In: *Proceedings of Financial Cryptography and Data Security (FC’14)*. Mar. 2014. URL: http://fc14.ifca.ai/papers/fc14_submission_71.pdf.

[KM18] Tommy Koens and Stijn Meijer. *Matching Identity Management Solutions to Self-Sovereign Identity Principles*. 2018. URL: https://www.slideshare.net/TommyKoens/matching-identity-management-solutions-to-selfsovereign-identity-principles/1.

[KMZ17] Aggelos Kiayias, Andrew Miller, and Dionysis Zindros. “Non-Interactive Proofs of Proof-of-Work”. In: *IACR Cryptology ePrint Archive* 2017 (2017), p. 963.

[Ler16] Sergio Demian Lerner. *Drivechains, Sidechains, and Hybrid 2-way Peg Designs*. working paper, revision 9. Apr. 2016. URL: https://bravenewcoin.com/assets/Whitepapers/Drivechains-Sidechains-and-Hybrid-2-way-peg-Designs-R9.pdf.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem". In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4.3 (1982), pp. 382–401.

[LZS18] Chengnian Long, Bin Zhao, and Qingwei Shi. *Decentralized Infrastructure for Next Generation Internet of Things*. Jan. 2018. URL: https://www.cpchain.io/CPChain_Whitepaper_English.pdf.

[Mic17] Microsoft Corporation. *The Coco Framework Technical Overview*. Aug. 2017. URL: https://github.com/Azure/coco-framework/blob/master/docs/Coco%20Framework%20whitepaper.pdf.

[Mie+13] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. "Zerocoin: Anonymous Distributed E-Cash from Bitcoin". In: *Proceedings of the 2013 IEEE Symposium on Security and Privacy*. SP '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 397–411. ISBN: 978-0-7695-4977-4. DOI: 10.1109/SP.2013.34. URL: http://dx.doi.org/10.1109/SP.2013.34.

[Min+15] Julien Mineraud, Oleksiy Mazhelis, Xiang Su, and Sasu Tarkoma. "Contemporary Internet of Things platforms". In: *CoRR* abs/1501.07438 (2015). URL: https://arxiv.org/abs/1501.07438.

[Min+16] Julien Mineraud, Oleksiy Mazhelis, Xiang Su, and Sasu Tarkoma. "A Gap Analysis of Internet-of-Things Platforms". In: *Computer Communications* 89.C (Sept. 2016), pp. 5–16. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2016.03.015. URL: http://dx.doi.org/10.1016/j.comcom.2016.03.015.

[Nak01] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008-11-01. URL: http://bitcoin.org/bitcoin.pdf.

[one18] onem2m. *The oneM2M Base Ontology, TS-0012-V3.7.1*. Mar. 2018.

[PB17] Joseph Poon and Vitalik Buterin. *Plasma: Scalable Autonomous Smart Contracts*. Aug. 2017. URL: https://plasma.io.

[PD16] Joseph Poon and Thaddeus Dryja. *The Bitcoin Lightning Network: Scalable off-chain instant payments*. Jan. 2016. URL: https://lightning.network/lightning-network-paper.pdf.

[Pop17] Serguei Popov. *The Tangle, White paper*. Oct. 2017. URL: https://iotatoken.com/IOTA_Whitepaper.pdf.

[PSL80] Marshall Pease, Robert Shostak, and Leslie Lamport. "Reaching Agreement in the Presence of Faults". In: *Journal of the ACM (JACM)* 27.2 (1980), pp. 228–234.

[RAC16] Dave Raggett, Kazuyuki Ashimura, and Yingying Chen, eds. *White Paper for the Web of Things*. Jan. 2016. URL: http://w3c.github.io/wot/charters/wot-white-paper-2016.html.

[RH11] Fergal Reid and Martin Harrigan. "An Analysis of Anonymity in the Bitcoin System". In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. Oct. 2011, pp. 1318–1326. DOI: 10.1109/PASSAT/SocialCom.2011.79.

[Ser+15] M. Serrano, P. Barnaghi, F. Carrez, P. Cousin, O. Vermesan, and P. Friess. *IoT Semantic Interoperability: Research Challenges, Best Practices, Recommendations and Next Steps*. Ed. by European Research Cluster on the Internet of Things (IERC). Mar. 2015.

[Sey+17] Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. "Capturing the Contributions of the Semantic Web to the IoT: a Unifying Vision". In: *CoRR* abs/1709.03576 (2017). URL: http://arxiv.org/abs/1709.03576.

[Sol+15] John Soldatos, Nikos Kefalakis, Manfred Hauswirth, Martin Serrano, Jean-Paul Calbimonte, Mehdi Riahi, Karl Aberer, Prem Prakash Jayaraman, Arkady Zaslavsky, Ivana Podnar Žarko, Lea Skorin-Kapov, and Reinhard Herzog. "OpenIoT: Open Source Internet-of-Things in the Cloud". In: *Interoperability and Open-Source Solutions for the Internet of Things*. Ed. by Ivana Podnar Žarko, Krešimir Pripužić, and Martin Serrano. Cham: Springer International Publishing, 2015, pp. 13–25.

[Sun+17] H. Sun, H. Mao, X. Bai, Z. Chen, K. Hu, and W. Yu. "Multi-Blockchain Model for Central Bank Digital Currency". In: *2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. Dec. 2017, pp. 360–367. DOI: 10.1109/PDCAT.2017.00066.

[SW16] I. Szilagyi and P. Wira. "Ontologies and Semantic Web for the Internet of Things - a survey". In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. Oct. 2016, pp. 6949–6954.

[TJF17] Mary Tate, David Johnstone, and Erwin Fielt. "Ethical issues around crowdwork: How can blockchain technology help?" In: *28th Australasian Conference on Information System*. Hobart, 2017. URL: https://eprints.qut.edu.au/115042/.

[TSa] Stefan Thomas and Evan Schwartz. *A Protocol for Interledger Payments*. URL: https://interledger.org/interledger.pdf.

[TSb] Stefan Thomas and Evan Schwartz. *A Protocol for Interledger Payments, White paper*. URL: https://interledger.org/interledger.pdf.

[TS16] Florian Tschorsch and Björn Scheuermann. "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies". In: *IEEE Communications Surveys Tutorials* 18.3 (2016), pp. 2084–2123. ISSN: 1553-877X. DOI: 10.1109/COMST.2016.2535718.

[Vel15] Laurent Velez. *oneM2M Showcase demos*. Dec. 2015. URL: https://docbox.etsi.org/Workshop/2015/201512_M2MWORKSHOP/S01_SETTINGTHESCENE/ETSI_VELEZ.pdf.

[Woo14] Galvin Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. 2014. URL: https://github.com/ethereum/yellowpaper.

[Woo16] Laurent Wood. *Polkadot: Vision for a Heterogenous Multi-Chain Framework, White Paper*. Nov. 2016. URL: https://polkadot.network/PolkaDotPaper.pdf.

[Zha+16] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. "Town Crier: An Authenticated Data Feed for Smart Contracts". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: ACM, 2016, pp. 270–282. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978326. URL: https://eprint.iacr.org/2016/168.pdf.