# Bringing privacy, security and performance to the Internet of Things using IOTA and usage control

Nathanaël Denis, Sophie Chabridon, Maryline Laurent

# Bringing Privacy, Security and Performance to the Internet of Things using IOTA and Usage Control

## Preprint - To be published in Annals of Telecommunications

Nathanael Denis[1,2], Sophie Chabridon[1,2], and Maryline Laurent[1,3]

[1] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France
nathanael.denis@telecom-sudparis.eu, sophie.chabridon@telecom-sudparis.eu, maryline.laurent@telecom-sudparis.eu
[2] Associate researchers of the Chair Values and Policies of Personal Information
[3] Cofounder of the Chair Values and Policies of Personal Information

**Abstract.** The Internet of Things (IoT) is bringing new ways to collect and analyze data to develop applications answering or anticipating users' needs. These data may be privacy-sensitive, requiring efficient privacy-preserving mechanisms. The IoT is a distributed system of unprecedented scale, creating challenges for performance and security. Classic blockchains could be a solution by providing decentralization and strong security guarantees. However, they are not efficient and scalable enough for large scale IoT systems, and available tools designed for preserving privacy in blockchains, e.g. coin mixing, have a limited effect due to high transaction costs and insufficient transaction rates.
This article provides a framework based on several technologies to address the requirements of privacy, security and performance of the Internet of Things. The basis of the framework is the IOTA technology, a derivative of blockchains relying on a directed acyclic graph to create transactions instead of a linear chain. IOTA improves distributed ledger performance by increasing transaction throughput as more users join the network, making the network scalable. As IOTA is not designed for privacy protection, we complement it with privacy-preserving mechanisms: merge avoidance and decentralized mixing. Finally, privacy is reinforced by introducing usage control mechanisms for users to monitor the use and dissemination of their data. A Proof of Concept is proposed to demonstrate the feasibility of the proposed framework. Performance tests are conducted on this Proof of Concept, showing the framework can work on resource-constrained devices and within a reasonable time. The originality of this contribution is also to integrate an IOTA node within the usage control system, to support privacy as close as possible to the objects that need it.

**Keywords:** IoT · Privacy · DAG · IOTA · PET · Usage Control

## 1 Introduction

The Internet of Things (IoT) is a ubiquitous network where connected devices exchange data between each other, as well as with users [11]. The devices collect data about their environment and usually transfer them to centralized cloud service providers, also known as CSPs. The CSPs process the data in order to provide a real-time and customized service to customers. Due to the number of devices concerned, their heterogeneity and the personal nature of the data collected, privacy and security are at risk in IoT systems, thus resulting in the need for new privacy-preserving solutions, well-tailored for the Internet of Things [2,24].

Currently, the most common model centralized around CSPs is troublesome for the IoT both for privacy and security reasons. Indeed, cloud service providers must not be automatically trusted and may snoop on users' data [29]. Besides, they can be vulnerable to internal attacks, from malicious employees, as well as accidental disclosures or external attacks [29]. Availability can be a matter of concern too, as physical infrastructure can be damaged, e.g. because of a fire or a natural disaster [3]. Furthermore, centralization hinders performance, specifically by increasing the cost of deployment and maintenance [35], which limits scalability.

Blockchain has been drawing attention as a solution to security issues, because of its properties regarding decentralization and the removal of intermediate third parties (cf. Section 2.1). However, conventional blockchains are not suitable for IoT systems, as they are computationally expensive, not scalable enough and introduce memory

and bandwidth overhead [11]. Besides, while conventional blockchains address security issues, they provide no more than pseudonymity. Privacy in blockchains is a specific and challenging topic, different from security, that must be addressed using dedicated tools.

Distributed ledgers more suitable for the Internet of Things than conventional blockchains have been designed, such as IOTA. IOTA is a distributed ledger technology that aims to power the Internet of Things (IoT) ecosystem by facilitating secure communications and transactions between devices [27]. Unlike traditional blockchain architectures, IOTA operates on a directed acyclic graph (DAG) known as the Tangle, which does not require blocks or miners. The Tangle enables zero-fee microtransactions and provides scalability in terms of transaction, making it more suitable for the Internet of Things. Its structure is designed for lightweight, efficient data transfer and high transaction throughput to integrate various IoT devices without the bottlenecks and high transaction fees often associated with public blockchain networks.

This paper is an extended version of a previous work published in IFIP Privacy and Identity Management [10]. The new contributions in this article are the following:

- update of the related work section (Section 2);
- a Proof of Concept of the proposed framework;
- performance tests to evaluate the feasibility of the solution under our assumptions and scenario. Tests are conducted using a remote node of the IOTA technology and then with a local node optimized, in particular by integrating an IOTA node;
- an extended security analysis using the STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) model, taking the example of our former privacy analysis based on LINDDUN (Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance).

This article is structured as follows: Section 2 summarizes the current state of the art about blockchains, usage control and privacy in the Internet of Things. Section 3 describes the car sharing use case over which both system and threat models are elaborated. Our framework for supporting privacy, security and performance in the IoT, is explained in Section 4. A Proof of Concept of a usage control system based on the IOTA technology is detailed in Section 5, focusing on performance aspects. The security and privacy analysis is carried out in section 6 before concluding in Section 7.

## 2 Related work

Considering the need for decentralization, security and privacy in the Internet of Things, this section identifies specific distributed ledgers (2.1) and usage control (2.2) technologies as candidate solutions and discusses their current limitations and state of the art. We eventually discuss the privacy of blockchain transactions in Section 2.3.

### 2.1 Distributed ledgers

Regarding the specific requirements of the Internet of Things, distributed ledgers, in particular blockchains, have been actively examined as an appealing solution. A blockchain is a "distributed and immutable ledger made out of an unalterable sequence of blocks" [35]. This technology provides several properties of interest for the Internet of Things [7]: 1) decentralization; 2) ability to audit the data; 3) disintermediation; and 4) transparency. Decentralization and disintermediation are particularly relevant for large scale deployments and for security, as they limit the extent of data leaks and prevent potential misbehavior from CSPs.

**Blockchain topology.** Blockchains can be of three types: public, private or consortiums [35]. *Public blockchains* do not control access and are called permissionless, while private and consortiums blockchains do have a control layer and are called permissioned blockchains. Public blockchains are distributed and tamper-proof ledgers that are not controlled by a single entity and are open to anyone. New entries can be appended to the ledgers as long as the network participants agree. To this end, the participants use a consensus method in order to determine

who can add a new block to the chain. Conversely, *private blockchains* restrict access to the public. Access to the network and involvement in the consensus protocol rely on authorizations, and require a third-party [35]. In particular, participants can join a private blockchain network only through an invitation where their identity or other credentials are authentic and verified. The validation is done by the network operator or by a clearly defined set of protocols implemented by the network through smart contracts or other automated approval methods.

Overall, security and privacy are better in a private blockchain from the perspective of nodes' owners since they control both the network and data access. Private blockchains are consequently more appropriate than public blockchains for most IoT use cases, due to better performance and privacy [35].

Finally, *consortium blockchains* are partially private blockchains, shared between several institutions instead of a single one. All these institutions are directly involved in the consensus protocol. The only concrete difference between consortium and private blockchains is the number of governing institutions. As a consequence, they will be considered as private blockchains in this paper.

**Consensus methods for the Internet of Things.** Blockchains implement consensus methods to agree on which data can be appended to the ledger and by whom. Consensus methods are paramount in blockchains as they enable the nodes to reach an agreement on the order and the validity of transactions and update the distributed ledger without the need for a central entity [15]. Moreover, the blockchain network is as secure as its consensus method is robust. Therefore, modifying the consensus method allows trading security for performance, and the parameters of the blockchain network are deeply impacted by the selected consensus method. The performance of blockchains can be qualified as follows [35]:

- *throughput*, generally measured in transactions per second (TPS);
- *latency*, also referred to as block time, the time between the creation of two blocks on the blockchain;
- *network overhead*;
- *storage overhead*;
- *scalability*, that can be defined as "the ability of the network to support an increasing load of transactions" [32].

Conventional blockchains heavily rely on *Proof of Work* (PoW) mechanisms, which are computationally expensive and not suitable for resource-constrained devices of the Internet of Things. The main alternative to proof of work in mainstream blockchains is the *Proof of Stake* (PoS), where the node responsible for block creation is chosen at random based on its proportional stake in the network. While this removes the resource-hungry computational race, it still introduces new issues. It is based on a monetary concept, the stake, which excludes many IoT use cases, including sensors, that do not require the use of currencies. The Proof of Stake gives the power to the most important holders, partially centralizing the blockchain network. Finally, *Proof of Elapsed Time* (PoET) is another IoT-friendly consensus method. While miners still have to solve the computation puzzle, the winner is chosen based on a random wait time. The next block is created by the miner whose timer expires first, and miners are not competing. However, the verification of the right timer execution is done with a *Trusted Environment Execution* provided by Intel. Consequently, this consensus depends upon Intel which goes against the decentralization property.

To make a blockchain network suitable for large scale IoT deployments, all these properties must be achieved simultaneously. To this end, the current literature is looking for specific consensus methods for the Internet of Things. Raghav *et al.* [30] propose a lightweight consensus mechanism for blockchains in the IoT. This consensus method is called *Proof of Elapsed Work And Luck* (PoEWAL). Its performance, energy consumption and latency are compared to those of several consensus methods, including Proof of Work and Proof of Stake. It turns out its performance is overall better than Proof of Stake considering different parameters, without introducing monetary concepts, making it suitable for the IoT. Another line of research focuses on the use of artificial intelligence to integrate IoT with blockchains, especially to improve the consensus method. Salimitari *et al.* [34] propose a framework for consensus in blockchain-based IoT systems with the support of machine learning. Actually, their solution consists of a 2-step consensus protocol, first detecting anomalies with machine learning, then using the *Practical Byzantine Fault Tolerance* (PBFT) consensus. The PBFT consensus method allows a distributed system to reach a consensus even though a small number of nodes demonstrate malicious behavior.

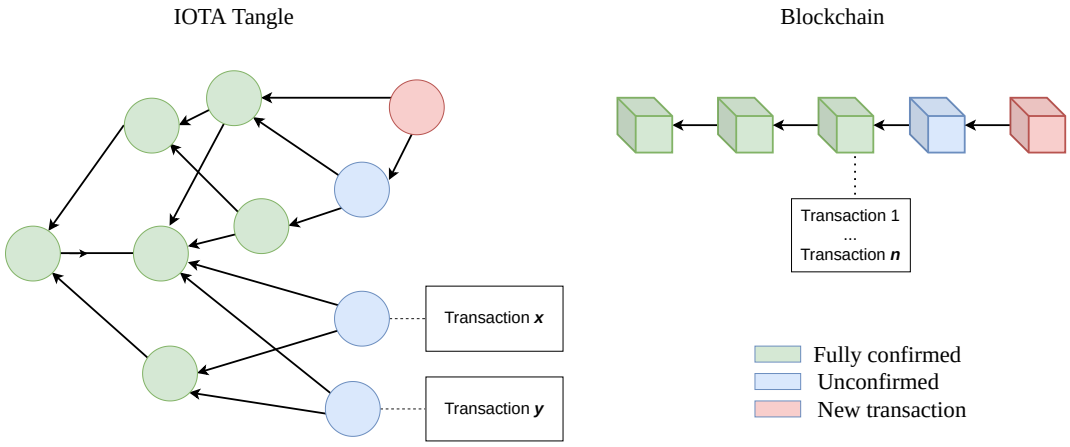


**Fig. 1.** Tangle transaction graph compared to traditional blockchains. Blocks contain several transactions while vertices in the Tangle contain only one transaction.

**Directed acyclic graphs are an alternative to blockchains.** They are used by the IOTA technology [27] to build the Tangle, IOTA's graph of transactions. To issue a new transaction, a node of the IOTA network has to validate two pending transactions known as tips (cf. Figure 1). In blockchains, blocks can be composed of several transactions. However, in the Tangle, each block is composed of only one transaction. A transaction is pending until confirmed by another transaction.

Finally, the node processes a light proof of work to prevent spam. This unique system ensures scalability, as more transactions mean faster tip validations [1], whereas common blockchains tend to saturate when the number of transactions increases. This creates two network regimes for IOTA, called respectively *low load* and *high load* regime depending on the current amount of transactions being pushed to the network [27]. In the low load regime, the transactions take longer to be validated, waiting for future transactions.

IOTA does not require a computationally expensive proof of work for strong security, but uses a proof of work affordable for IoT devices to protect from spam. Moreover, there are no rewards for the proof of work which implies the transactions are free, thus making micropayments possible, a boon for many IoT use cases. Storing a potentially huge ledger on devices is another issue to consider. For nodes with insufficient storage capacity, local snapshots can be created removing some transaction data. This process is known as *pruning* and enables the deployment of lightweight nodes [25]. Yet, the IOTA technology has a major flaw, because it is at the moment partly centralized. Indeed, IOTA relies on a *coordinator node* run by the IOTA Foundation, i.e. the foundation that created and has been developing IOTA, whose mission is to directly or indirectly validate transactions [39]. It does not completely centralize the network as all the nodes verify that the coordinator node does not break the consensus rules, yet it can freeze funds, ignore transactions and is a single point of failure, i.e. if the coordinator stops, after an attack or by purpose, transactions are no longer validated. In order to solve the coordinator issue, the IOTA Foundation is developing a new IOTA 2.0 network, whose test network (1.5) is already available. The removal of the Coordinator is likely to be achieved by introducing new components, particularly a new consensus method called *Fast Probabilistic Consensus* (FPC) and a node accountability system to protect against basic attacks [28].

### 2.2 Usage control

Usage control, as an extension of access control, monitors how the data can be used after initial access. It was first proposed by Sandhu and Park as the UCON model [26]. This model extends traditional access control by introducing attribute mutability, as well as new decision factors, namely obligations and conditions. Obligations are requirements to be fulfilled by the subject to be granted access. Conditions are subject-independent environmental requirements for allowing access. Since attributes are mutable, authorizations and obligations can be done before or during the access. They are referred to as pre-authorizations and ongoing-authorizations, or respectively pre-

obligations and on-going obligations. Improving user control over the data is crucial to achieve privacy in IoT systems [6], and UCON provides the technical basis to enable this control.

**Modern Usage Control Systems (UCS) integrate Data Flow Control (DFC) to complement UCON.** To actually control the usage, another concept was introduced to complement UCON: Data Flow Control [17], [23]. Data Flow Control (DFC), also referred to as Information Flow Control (IFC), aims at controlling the flow of information and ensuring the data is not disseminated to irrelevant actors. Therefore, DFC trackers are components of modern data usage control systems (UCS), whose purpose is to improve their behavior, especially when multiple copies of the data are distributed over numerous devices.

To achieve a reliable and controllable enforcement of the usage control rules, usage control may rely on a Trusted Execution Environment (TEE) [38]. A trusted execution environment is an area on the main processor of a device that is separated from the system's main operating system (OS). It ensures that data is stored, processed and protected in a secure environment. The TEE is installed on the monitored user's machine, to prevent undue processing or dissemination of the monitored data.

**The integration of usage control with blockchains is a recent topic of research.** Most existing works rely only on permissioned ledgers for usage control. Khan *et al.* [21] propose to integrate UCON in blockchains relying on the Hyperledger Fabric, a permissioned blockchain. For the authors, the purpose of introducing UCON is to monitor assets continuously to cover all possible access control models. Shi *et al.* [38] propose a Distributed Usage Control Enforcement (DUCE) for the Internet of Things, to solve privacy issues existing in the Cloud-Enabled Internet of Things (CEIoT). DUCE distributes some usage control components and relies on private blockchains to store tamper-proof data on the permissioned ledger. The policies are written using the XACML policy specification language and then converted into the Solidity language for smart contracts. DUCE relies on a Trusted Execution Environment (TEE), thus the protection of users' data depends on the strength of the TEE. Rizos *et al.* [31] suggest extending UCON to distributed systems in order to strengthen IoT security. More precisely, they adapt UCON to the MQTT and CoAP protocols. Finally, Kelbert and Pretschner [20] developed a fully decentralized usage control for distributed systems, including data flow tracking. In several situations, their decentralized policy enforcement outperforms a centralized one.

### 2.3 Transaction privacy

While blockchain transactions are thought to be anonymous, the reality is more nuanced. Public blockchains do not require identifying information to make a transaction worldwide. Yet, transactions are publicly broadcast. The transaction content, as well as the operation itself disclose information about the individuals involved. Interested third parties automatically collect and analyze this information, for several purposes including law enforcement [22]. By default, public blockchains only provide pseudonymity, and anonymity provided the linkage between the pseudonym and the real identity is not possible. Yet, two behaviors facilitate significantly the re-identification analysis: address reuse and super-clusters with high centrality. Using address clustering, i.e. partitioning the addresses into subsets likely controlled by the same entity, combined with address tagging and graph analysis, it is possible to re-identify more than 69% of wallets stored by Bitcoin lightweight clients [22].

**Privacy-preserving techniques have been designed to mitigate the effectiveness of de-anonymisation.** The most well-known tools for enforcing privacy in transactions are coin mixing and merge avoidance, which can theoretically be added on top of any blockchain [36]. Both aim at obfuscating the transactions by adding new fictional ones. In merge avoidance, a single transaction between two users is split into numerous sub-transactions for both users, hiding the amount of the original transaction. A new address must be created for each sub-transaction. Otherwise, it will be easy to rebuild the original transaction using a blockchain explorer with either the sender or the receiver address.

Cryptocurrency mixing services, also known as cryptocurrency tumblers or coin mixers, are designed to remove the linkage between the sender and the receiver of a transaction. To achieve this task, the cryptocurrency mixing

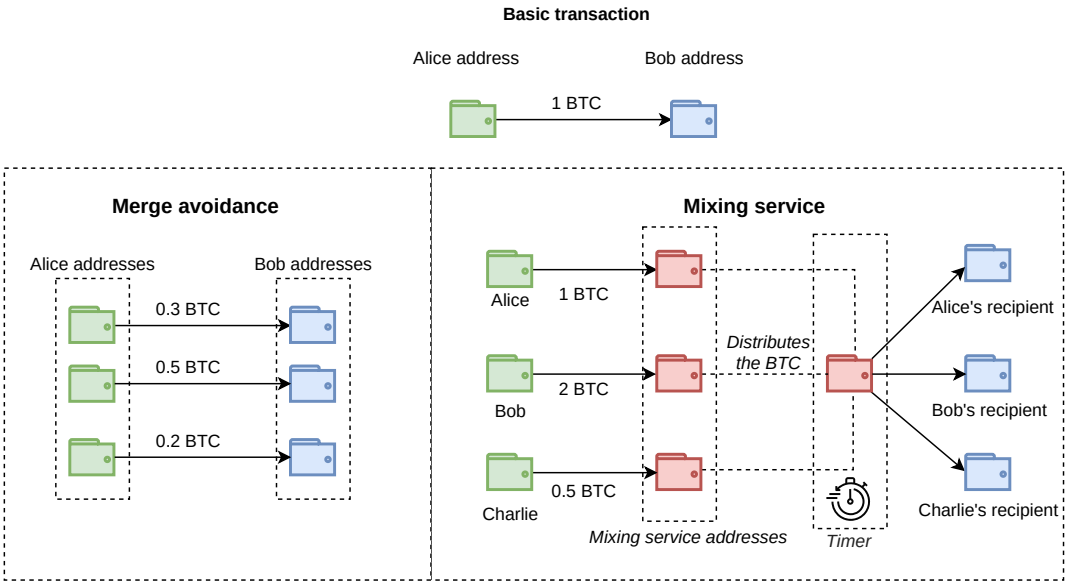


**Fig. 2.** Obfuscation with merge avoidance and mixing on the Bitcoin blockchain. The mixing service regroups every coin to be mixed, then sends the due coins to the recipients in a random order and after the random timer is exhausted.

service gathers coins from different users, whose identities are linked to these coins. The mixing service then keeps the coins for a long, potentially random period of time before randomly assigning the coins to the users. The coins, after being regrouped, are distributed at random which removes the linkability between the coins and the users. The purpose of randomness and keeping the coins for a long time is to avoid re-identification of the sender by using the timestamps. The mixing process is shown in Figure 2, in a version where each destination address receives its coin in a random order and at random time, using a timer.

Besides, some cryptocurrencies have been specifically designed to enforce privacy in their transactions, such as Zcash (ZEC) [5] and Monero (XMR) [33], obfuscating the transactions with several Privacy-Enhancing Technologies (PETs) and cryptographic tools.

**Privacy in the IOTA technology.** Apart from using a directed acyclic graph instead of a blockchain, IOTA has several features that change the concerns related to privacy. Its main asset is the free transaction cost, making merge avoidance particularly relevant as transactions can be virtually split into infinite sub-transactions. Decentralized mixing is then relevant as the network does not rely on financial motivation. To this end, Sarfraz [36] designed a decentralized mixing service for the IOTA network, which requires no mixing fees. Mixing consists in joining coins from different senders before swapping their receivers, in order to remove linkage. Conversely, IOTA has some properties harmful to privacy. Indeed, the removal of the mining process prevents from creating tokens without taint. A token is considered as tainted if it belongs to at least one identifying address on the IOTA ledger. All IOTA tokens were created in the first *genesis* transaction. Only iotas that have never been linked to any identifiable address, i.e. an address belonging to someone who has been re-identified, can be considered as untainted [41].

## 3 Illustrating scenario and threats

To identify the needs in terms of performance, security and privacy for large scale deployments of IoT systems, a car club illustrating scenario is first proposed in Section 3.1. The following sections 3.3 and 3.4 highlight respectively the privacy and security threats based on this scenario.

### 3.1 Scenario

Car clubs (UK) or car sharing (US) is a model of car rental where people rent cars for a short period of time, often by the hour. They differ from classic rental models in that the owners of the cars are individuals themselves, instead of an agency. The context is highly dynamic, as many users may enter the car club or leave it on the same day. In order for the users to interact with the system, an application is responsible for registration and asking or granting access to the vehicles.

Mainly for security reasons, the car owners have the right to watch over their cars and know where they are, almost in real-time. The position of the cars as well as their navigation produce data about the car renters which are sent to the car owners. Car owners use a mobile phone application to define the access policy for their car. Similarly, car renters define usage control policies using the same application. Car renters use a public distributed ledger technology to make transactions in a decentralized, auditable fashion. Car renters as well as car owners have one or several pseudonyms i.e., addresses, where they can send their coins as a payment.

### 3.2 Agents

The agents of the system in this scenario can therefore be summarized as follows:

- the car owners, who propose their vehicles on the renting market;
- the car renters, who pay for renting the vehicles;
- the car itself, which sends data to the owners such as location, and whose access must be monitored;
- the Access Server (AS), which is responsible for managing the access to the cars;
- the Usage Control System (UCS), which monitors the data generated by other agents;
- the mixing server, responsible for obfuscating the transactions to preserve privacy.

Actually, both the Access Server and the Usage Control System control access, respectively to a physical object - the car - and to the data. The UCS also prevents the dissemination of the data to irrelevant actors.

### 3.3 Privacy threat model

Depending on the data obtained by the attackers, and following the LINDDUN threat evaluation framework [9], we discuss the threat analysis for our proposed scenario hereafter.

- *linkability*(L): an attacker can link the car renter and the car owner, respectively the sender and the receiver of a transaction, thus simplifying re-identification and inference. Besides, an attacker can also link coins to its holders if they are tainted;
- *identification*(I): the attacker can link the pseudonym to the real identity of the car renters or the car owners;
- *Non-repudiation and repudiation*(N): With repudiation, an attacker can exfiltrate information and deny it did. Note that this threat is actually *a security goal* in our system, contrary to other threats. Non-repudiation can conversely be a threat to legitimate users, when an attacker has the possibility to prove a user has done some sensitive actions e.g., an illicit transaction [9]. In our scenario, non-repudiation is not considered a threat, but repudiation is;
- *disclosure of information*(Di): an attacker can get data about a user without having the access rights. We also include inference attacks in this category, which can be defined as "attacks where the attacker has used existing knowledge to aid the attack" [18]. An inference attack occurs when an attacker is able to infer information from apparently harmless information. For example, in our scenario, an attacker could infer working hours by gathering transaction timestamps;
- *unawareness*(U): a car renter is not aware of the collection, processing, sharing and storage of their geolocation data;

Detectability(D) is not considered a threat as data is publicly registered on the ledger. Both the existence and the content of the data are already known to the attackers. Rather than preventing Detectability, the focus is given to preventing its most important consequence, inference [43]. Non-compliance(N) is considered as an orthogonal issue since the regulations are country-dependent. However, distributed ledgers may have several compliance issues, such as their immutability which contravenes articles 16, 17 and 18 of the GDPR about the right to data deletion and modification [42]. Finding technical solutions to compliance issues is an active field of research [16].

### 3.4 Security threat model

Considering the agents defined in the scenario, the threat model identifies *four attacker types*:

1. the single car owner, who has a legitimate access to some sensitive data of the car renters;
2. several car owners colluding with each other to gather big sets of data;
3. the mixing server, who may keep for itself the addresses of senders and receivers, or put another way, secretly keep the links it is supposed to remove. It can use this information to carry out re-identification attacks;
4. external attackers, who wish to disable the UCS to help car owners disseminate data to other agents.

The car owners are considered *honest-but-curious*, which means they will fulfill their mission, but will snoop on the data of the users requesting their services. Honest-but-curious attackers are assumed to rely on transaction contents only, rather than network-level information, e.g. IP addresses, to re-identify users. External attackers are conversely *malicious* and may try actively to neutralize the UCS to enable car owners to disseminate their data. The main motivation of honest-but-curious attackers is to gather as much data as possible.

Concurrently, there are *risks to security* because a single agent of the system - namely the UCS - is responsible for the data protection. The UCS itself is considered as *trusted*. External attackers can however be interested in neutralizing the UCS, e.g. by disabling or modifying the UCS, to enable car owners to collude. Similarly to the privacy threat analysis using LINDDUN, we rely on the STRIDE security threat model [19] to identify the security threats weighing on the usage control system:

- *Spoofing*(S): an external attacker could masquerade as a legitimate user to be granted access to unauthorized data, or as the control system to collect the car renters' data;
- *Tampering*(T): an external attacker could modify either the data or the infrastructure of the usage control system. Besides, an attacker could try to modify the binaries of the usage control system to make it ineffective [20];
- *Denial of service*(D): the external attacker can temporarily disable the UCS, threatening the availability of the system and disabling the usage control mechanisms.

The Repudiation(R) and Information disclosure(I) risks are already tackled as privacy threats by the LINDDUN privacy threat model, and therefore excluded from the security threat modeling. Finally, an external attacker can conduct an Elevation of privilege(E) by leveraging vulnerabilities as illustrated in [4] to bypass the UCS restrictions. These attacks are very diverse and implementation-dependent, therefore considered out of the scope of this paper.

## 4 Proposed framework

Regarding the different challenges for large scale deployments of IoT systems, as illustrated by the car sharing scenario (cf. Section 3.1), a set of complementary tools is needed to match privacy, security and performance requirements simultaneously. To this end, the originality of our work is to design a framework with the following features (cf. Figure 3):

1. IOTA technology, as the most promising solution to match IoT performance requirements;
2. IOTA Access, an open-source framework used to control access to IoT devices. It is developed by the IOTA Foundation to complement the IOTA technology;
3. a Usage Control System, for car renters to monitor the usage of the data they produce. The UCS relies on a Trusted Environment Execution on the device of the car owner;
4. a decentralized mixing service coupled with merge avoidance, to obfuscate the transactions and improve users' privacy.

IOTA and its Tangle are introduced along with IOTA Access, the framework developed by the IOTA Foundation to control the access to devices. IOTA Access is meant for any device, ranging from sensors to vehicles. The Usage Control System, which controls the data and how they are disseminated, is embedded into IOTA Access. The mixing service is external to the Tangle and they interact with one another. Merge avoidance can be programmed directly by the user, when sending the transactions to the mixing service.

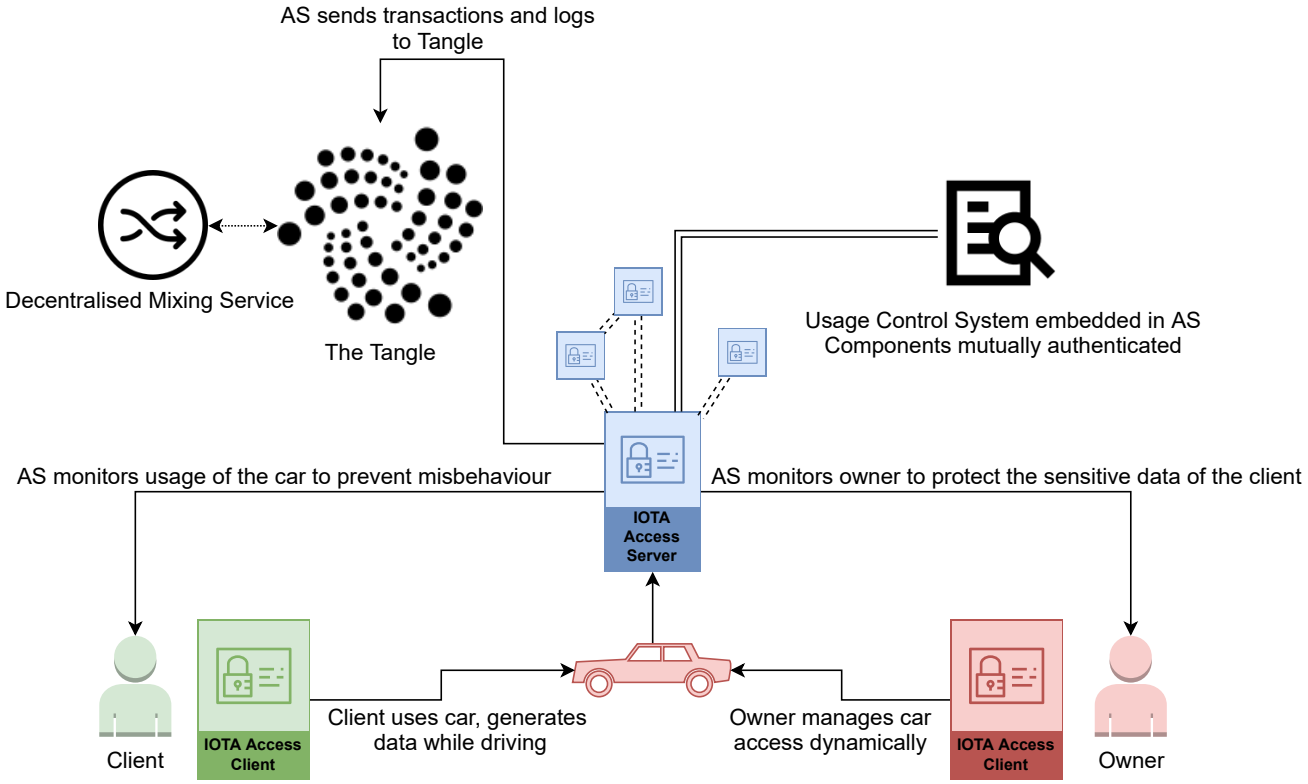


**Fig. 3.** Framework and relationships between agents

The IOTA Access framework is composed of three main components: a policy database to store the access control policies, a client so that the car owners can define their policies and can grant access to their cars, and finally a server monitoring the access and interacting with the Tangle. As the Access Server (AS) already controls the access to vehicles, the UCS is embedded into the AS even if the controlled objects differ. Indeed, the AS controls access to a physical device, the car, while the UCS monitors access to the data and prevents dissemination. The UCS uses a Trusted Execution Environment to monitor how the car owner uses the geolocation data. The Access Clients are deployed on car renter and car owner mobile devices to manage their respective policies.

**Decentralizing the framework.** First, we emphasize that the IOTA Access server is already decentralized, as it can be deployed by anyone. In our use case, the most suitable solution is to pick one external trustworthy server to connect to, which is realistic as a list of trustworthy IOTA nodes is maintained by the community[4]. The same principle could be extended to IOTA Access servers, with a list of the top public ones.

Merge avoidance and mixing are used jointly to increase the effect of obfuscation. The effectiveness of merge avoidance is increased due to free transactions on the IOTA network. For the same reason, mixing is more efficient as the nodes involved in the mixing service do not have to pay for the transactions. Indeed, if IOTA nodes were encouraged to participate for money, decentralized mixing services would become vulnerable to edge insertion attacks [40] where nodes can claim undue rewards. Therefore, our framework uses a decentralized mixer to remove the threat of linkage and re-identification, and without introducing the edge insertion issue.

The Usage Control System must be decentralized as well in order to benefit from the IOTA 2.0 (without the coordinator) and to be resilient to some attacks like denial of service. Kelbert and Pretschner [20] implemented a decentralized usage control system. It is achieved by distributing the components of the UCS responsible for the

[4] https://trinity.iota.org/nodes

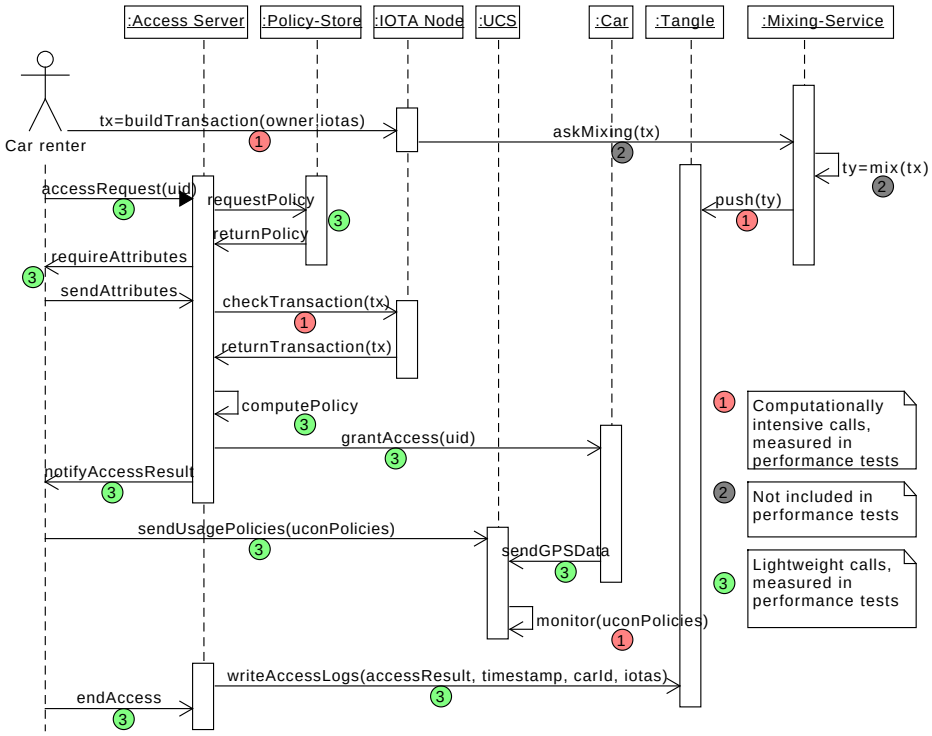


**Fig. 4.** The sequence diagram related to our framework. Calls are classified into different categories, according to how they are tested: 1) the computationally intensive calls; 2) the calls that are not included in the tests, related to the mixing service and its security level; 3) lightweight calls, essentially communications.

policy enforcement, and it addresses both the UCON and the data flow tracking aspects. Additionally, decentralizing the UCS reduces the communication and performance overhead compared to a centralized policy enforcement. In our framework, the Usage Control System is deployed along the IOTA Access servers which are decentralized as well, enabling the integration of Kelbert and Pretschner's solution.

**Sequence of interactions.** Figure 4 details the sequence of interactions in our framework through the use case presented in Section 3.1. These interactions unfold as follows. First, before the access can be granted, the car renter must ask the usage control system to send iotas to the car owner's wallet and sign the transaction, using `buildTransaction`. To avoid re-identification, the usage control system sends the transaction to the mixing service and requires mixing with `askMixing`. The mixing service removes the linkage between the car renter and owner, creates false intermediate transactions to obfuscate the transactions, and finally holds the iotas for a random period of time. These three processes are referred to as the `mix` call. At the end of the waiting process, the mixer sends the coins to the car owner's address using the `push` call. The `accessRequest` is then sent to the Access server, which then requests the car renter attributes to be able to take an access decision. The Access Server checks if the transaction is successful with the call `checkTransaction` to be able to grant or refuse access. The IOTA node returns the transaction and its result to the Access Server using the call `returnTransaction`. Then, it also fetches the car access policies from the policy store, before evaluating them with `computePolicy` and returning the access result to the car with `notifyAccessResult`. If the evaluation is positive, the car is unlocked with `grantAccess` and the car renter can get inside. Afterwards, the car renter sends its data usage policies to the UCS with `sendUsagePolicies`, before using the car. These data usage policies are composed of the authorizations, the obligations to be fulfilled by the car owner and finally the conditions on the system. For example, a pre-obligation of the car owner is to agree

to send the logs to the UCS, e.g. by reading instructions and ticking a box to actually agree. When driving, the car renter generates navigation data, relayed by the car to the car owner with the call `sendGPSData`. To comply with the car renter policies, the UCS continuously monitors the data usage of the car owner with `monitor`. In the sequence diagram, only one monitor call is considered i.e., one verification of conformity with a given policy, for simplification. The monitoring can actually be represented by several calls between the different UCS components. The Usage Control system being composed of a owner-side TEE, the `monitor` call is shown as a self message, masking actual interactions between the different components of the UCS. Once the monitoring is over, the Access server writes logs on the Tangle using `writeAccessLogs`. The logs detail the result of the access policies evaluation, the time information, the amount of iotas spent and finally a pseudonym for the car to simplify car management when a car owner has several vehicles. The addresses of the car renters i.e. their pseudonyms on the Tangle, are written in the logs as well when they request access to a car. Finally, the car renter leaves the system by calling `endAccess` on the Access server.

## 5 Proof of Concept

This section introduces a Proof of Concept (PoC) of the proposed framework. The purpose is to demonstrate the feasibility of the solution and to assess the computation and network overhead introduced by the different components of the framework. Firstly, it focuses on the design of the solution in Section 5.1, before assessing the performance of the usage control system in Section 5.2.

### 5.1 Generalities

The PoC focuses on usage and access control, excluding the mixing service for two reasons: 1) well-documented performance tests are provided by Sarfraz *et al.* [36] in addition to the proposed decentralized mixing service. In particular, the mixing service needs approximately a minute to sign the transactions of 10 participants simultaneously, much more than the time needed to monitor the access or the usage; 2) the mixing service is expected to hold the iotas for a potentially long period of time in order to prevent re-identification of the senders using timestamps (cf. Section 2.3). The mixing service is a tool designed for privacy purposes, but it is not necessary to wait for the mixing process to end in order to monitor the usage, as the UCS pushes the transactions itself (cf. Figure 3). Moreover, we rely on a *Cassandra* distributed NoSQL database as a decentralized storage for off-chain monitored data, notably car renters geolocation. Cassandra is horizontally scalable which means it can properly handle increasing traffic demands by adding more machines [8]. Cassandra can also work on a low-power cluster making it suitable for the Internet of Things [8]. The node is powered by *Hornet*, a community-driven IOTA node software written in Go. This software is maintained by a community of developers alongside the IOTA foundation.

IOTA has several networks, the main network is called *mainnet* and the development network is called *devnet*. Contrary to the mainnet, the devnet has free tokens and is designed for testing. We have conducted experiments using the development network that runs IOTA in its first version IOTA 1.0.

For the usage control system, the code is written in Java, and the interaction with the Tangle is managed using the Rust library bindings for Java. The usage control policies are defined by the users and written using the XACML language [14]. During the tests, policies are not specified by the users, but automatically generated for convenience. All the files related to the Proof of Concept are available on a public repository[5] for reproducibility purposes. The Proof of Concept is depicted in Figure 5.

### 5.2 Performance analysis

The proposed framework is designed to answer the performance, security and privacy needs of the Internet of Things. Therefore, an evaluation is necessary to ensure the system is actually consistent with the performance requirements. In its coordinator-less form, IOTA is decentralized and provides a high throughput and low latency; it scales well and handles storage overhead by using pruning on lightweight IOTA nodes. However, the introduction of several

[5] https://zenodo.org/record/6632102

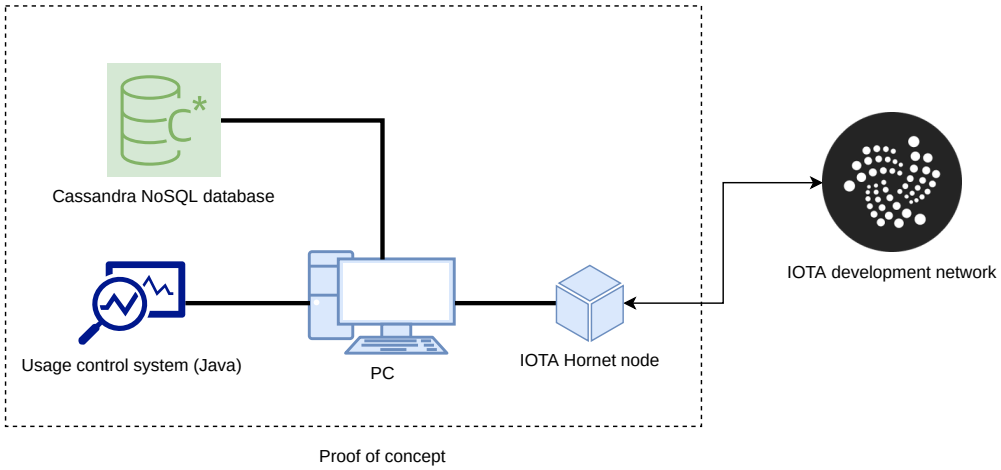


**Fig. 5.** Architecture of the Proof of Concept with local IOTA Hornet node that interacts with the development (test) network.

tools may introduce computation and network overhead, requiring further testing to demonstrate the viability of the framework.

Network performances e.g., in terms of scalability and throughput, (cf. Section 2.1), are not the purpose of this evaluation. This section aims to demonstrate the framework feasibility despite the stacking of several technologies, which will be evaluated in terms of:

- *Quality of service*: the entire chain of events leading to the access being granted should be reasonably short to be acceptable;
- *Computational power*: the type of hardware is important to run an efficient usage control system.

Regarding the Quality of Service, the focus is given to determining how long it takes for a user to be granted access after initiating a payment. Average and maximum times will be considered as metrics to assess the feasibility of the solution. Additionally, several hardware configurations are tested according to the Internet of Things specifications, to assess whether the UCS can be deployed on resource-constrained devices.

**Usage control optimization** To limit the computation and network overhead introduced by the usage control system, several optimizations are considered. Firstly, the usage control system deploys a node itself to integrate the IOTA network. This integration provides several benefits. The UCS can prioritize its own transactions and perform local analysis on its ledger without querying the other IOTA nodes. Secondly, the node itself is optimized by refusing to compute delegated proofs of work for other users and by relying on *spammers* to speed up the network when the network is in a low load regime. Spammers are useful for reproducible testing as well, whose output is different whether tests are conducted in low load or high load regime.

As IOTA validates transactions faster in high load regime, i.e. when many users push new transactions, it is relevant to use spammers that create zero-value transactions and validate two pending transactions from other users in the process. Spammers are implemented to ensure transactions do not take too long to be validated during low load regime. Small devices with very poor computation capacities or with energy constraints can delegate their proof of work to a node. Our node is configured to refuse delegations in order to dedicate its computation power to usage control.

**Methodology** The objective is to measure the time needed for a transaction to be validated and pushed to the network, and the time to fetch the transaction from an IOTA node, which is not null for the IOTA node. These operations correspond respectively to the calls `buildTransaction`, `push` and `checkTransaction` from the sequence diagram of Figure 4. Tests are conducted in three different configurations: (1) the IOTA remote node which is a

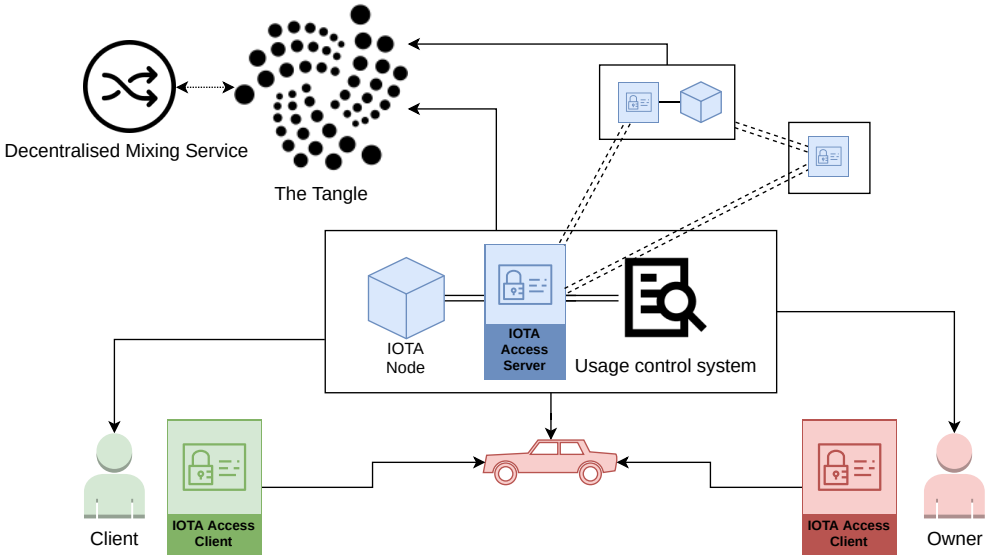


**Fig. 6.** The architecture for the local node: an IOTA node is deployed on the same device running the UCS, thus prioritizing transactions and locally analyzing the ledger

resource-constrained node, to help understand the behavior of the solution in a fully constrained IoT environment, (2) the IOTA remote node which is no longer resource-constrained to measure the gain from lifting the resource limitation, and (3) a local node which supports both the UCS and IOTA node, as illustrated in Figure 6. For each test, one thousand samples ($N = 1000$) are used. The resulting experimental measurements are summarized in Table 1.

**Resource-constrained remote testing.** To demonstrate the possibility of a usage control system interacting with IOTA on resource-constrained devices, the performance tests are first conducted on a virtual machine with 4096MB of RAM memory and an Intel Core i5-10210U CPU @ 1.60GHz (1 core). The number of transactions per second *on the test network* was oscillating between 3 TPS and 11 TPS on the test network, up to 16 with the spammer. The delegated proof of work is removed as part of the optimizations.

Pushing a transaction on a remote resource-constrained node (RCN), takes on average $\bar{t}_{push,rcn} = 5271ms$. Additionally, the usage control system takes an average $\bar{t}_{fetch,rcn} = 45ms$ to fetch the transaction result from the remote node, accounting for a total $\bar{t}_{rcn} = 5316ms$ on average as arithmetic mean is linear. The time needed to create and push a transaction can tremendously vary, from a minimum $m_{rcn} = 364ms$ to a maximum $M_{rcn} = 26851ms$, which is reflected by a standard deviation of $\sigma_{rcn} = 4629ms$. This difference is mostly due to the synchronization between peer nodes, which increases the transaction time significantly when the node is lagging behind or one of the peers is disconnected. The confidence interval is $I_{rcn} = \bar{t}_{rcn} \pm 1.96\frac{\sigma_{rcn}}{\sqrt{N}} = 5316 \pm 287ms$.

The results demonstrate that the solution can be deployed on a machine with low computation capacities. However, with delays up to 26 seconds to create, validate and push a transaction to the network, this might be unsatisfying according to the use cases, e.g. access to a vehicle.

**Resource-unconstrained remote testing.** The IOTA remote node is now run on a computer with more computing power, with an Intel Core i5-10210U CPU @ 1.60GHz (4 cores) and 8192MB of RAM memory, supporting the optimizations. This corresponds to the high-end Raspberry Pi 4 Model B specifications[6].

Pushing a transaction on a remote node (RN) with more computing capacity, takes on average $\bar{t}_{push,rn} = 1867ms$. Additionally, the usage control system takes on average $\bar{t}_{fetch,rn} = 45ms$ to fetch the transaction result from the

[6] https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

remote node, accounting for a total $\bar{t}_{rn} = 1912ms$ on average. The time needed to create and push a transaction is still very variable but spreads out less, from a minimum of $m_{rn} = 363ms$ to a maximum of $M_{rn} = 12209ms$, with a standard deviation of $\sigma_{rn} = 1499ms$. The samples express a significant impact of the UCS computation power when creating and pushing transactions to a node. The confidence interval is $I_{rn} = \bar{t}_{rn} \pm 1.96\frac{\sigma_{rn}}{\sqrt{N}} = 1912 \pm 93ms$.

The tests are also conducted using a much more powerful device, with 32GB RAM Memory and an Intel Core i5-1021UCPU @ 1.60GHz (8 cores). The purpose is to see if there is a limit in speed improvement as the UCS computation power increases. The results are actually very similar with the 8GB RAM setup, in the same confidence interval.

**Local testing.** The IOTA node is deployed on the local node (LN) running the UCS, as illustrated in Figure 6. The network connection, expressing the capacity of the local node to quickly get updates from other nodes, provides 98 Mbps in downlink and 77 Mbps in uplink. The node and the UCS runs on the same device with 8192MB of RAM Memory and with the Intel Core i5-10210U CPU @ 1.60GHz (4 cores). The optimizations are also enabled. The average time for a node to validate a transaction drops from $\bar{t}_{rn} = 1912ms$ to an average $\bar{t}_{ln} = 1579ms$. The minimum transaction time on a local node dropped from $m_{rn} = 363ms$ to $m_{ln} = 10ms$, while the maximum changed from $M_{rn} = 12209$ to $M_{ln} = 9830s$. The standard deviation is $\sigma_{ln} = 1544ms$. The confidence interval is $I_{ln} = \bar{t}_{ln} \pm 1.96\frac{\sigma_{ln}}{\sqrt{N}} = 1579 \pm 96ms$.

As a result, using a local node has the following benefits, compared to a remote node without computation power constraints (RN):

1. a 17.5% decrease on the average transaction time;
2. occasional very fast transactions, taking a minimum of 10ms instead of a minimum $363ms$;
3. a steady maximum time to push a transaction, only dropping from $12209ms$ to $9830ms$, which remains satisfactory;
4. 48% of transactions are processed within a second, compared to 34.5% for transactions using the remote node.

| **Test category** | **Min** | **Max** | **Average** | Standard deviation $\sigma$ |
|---|---|---|---|---|
| Remote (constrained) | 364ms | 26851ms | 5316ms | 4629ms |
| Remote (unconstrained) | 363ms | 12209ms | 1912ms | 1499ms |
| Local | 10ms | 9830ms | 1579ms | 1544ms |

**Table 1.** Performance measurements for different test configurations

**Additional calls** While the above-mentioned performance tests are conducted on the computationally intensive calls, the other category called *lightweight calls* (cf. Figure 4) have also been measured. The calls `accessRequest`, `requestPolicy`, `requireAttributes`, their corresponding return values `notifyAccessResult`, `sendAttributes`, `returnPolicy` as well as `grantAccess` and `endAccess` consist in messages exchanged between the actors. They are strongly dependent on the time to communicate between the car renters, the access server, and the policy store which was measured as under 1ms in our setup, since they were all running locally on the same device. These three entities can be realistically considered as close and the time for all the above-mentioned calls negligible compared to the `buildTransaction`, `push` and `checkTransaction` calls.

The remaining calls have a different behavior. `computePolicy` is composed of several boolean operations, taking a negligible time. `sendUsagePolicies` and `sendGPSData` are continuous operations, they are repeated until the access is terminated using the `endAccess` call or if `monitor` detects a violation of the policy. The time needed to `monitor` the access according to a given policy was measured and takes an average $\overline{p} = 5ms$ for a simple policy made of three rules, but this time may increase if the policy becomes more complex. Finally, the call `writeAccessLogs` is very similar to `buildTransaction` as a message on IOTA is built as a zero-value transaction. However, it does not

require checking balances and the construction of the transaction is simpler. A log takes an average $\bar{l}_{lo} = 473ms$ to be built and pushed to the network, on a local node using 1000 samples. Besides, the call `writeAccessLogs` does not impact the Quality of Service of the users as it is performed after the access is terminated.

In conclusion, the experiments have shown that the framework fulfills the performance requirements, regarding the quality of service and the computational power. The time needed to validate the access to a user requiring it is acceptable, and resource-constrained devices can run a usage control system and interact with an IOTA node. The IOTA node itself can run on a machine corresponding to Raspberry Pi Model B, as we did in the local testing section.

### 5.3 Extension to other technologies

While the Proof of Concept focuses on usage control, the integration principle can be generalized to any security or privacy technology that requires processing distributed ledger transactions and that can deploy a node. The network's security is improved with more nodes that validate the ledger. Throughput increases for directed acyclic graphs, including IOTA, as more transactions are submitted to the network [27].

Examples of such technologies are:

- the mixing service, which requires checking the ledger state to make transactions. Instead of querying IOTA nodes, the mixing service can query its local transaction ledger to avoid further communication costs;
- self-sovereign identities (SSI). SSI is an approach in which subjects are in full control of their own digital identities [13]. Self-sovereign identities are made of 3 main actors: the issuer who can issue digitally-signed identity attributes, the user who monitors and presents his or her identity attributes, and the verifier who wants to check the user's identity attributes. Decentralized identifiers are commonly stored on blockchains [13] due to their distributed nature. Deploying an IOTA node along the user can reduce the time needed to register its identity attributes on the blockchain.

## 6 Privacy and security analysis

This section analyzes the privacy and security risks in the system. It distinguishes the risks to privacy for the car renters, and the risks to security if the usage control system is compromised as well as a protection system to compensate the car owners if their cars are damaged.

### 6.1 Privacy threats and mitigations.

Firstly, Table 2 describes, for the inference attacks, each combination of attackers, the data types they have access to, where data is stored in the system and an example of a privacy leakage associated with this risk. Secondly, Table 3 describes other threats to privacy and how they are mitigated.

| Attacker type | Data type | Data storage | Example |
|---|---|---|---|
| **Honest-but-curious** | Transaction | Tangle | Purpose of payment |
| **Car owner (alone)** | Location | Owner's device | Renter's job |
| **Car owners (colluding)** | Location | Owners' devices | Renter's job |
| **Ext. attacker & car owners** | Location | Owners' devices | Data sets on renters |

**Table 2.** Inference attacks according to the attackers' profile

Any user has access to the transactions on the Tangle, which are public and contain privacy-sensitive timestamps, users' addresses and values, i.e. how many iotas are sent to a car owner. Based on these elements, any honest-but-curious attacker can attempt to use the blockchain transactions for inference attack, e.g. use the amount of tokens in the transactions to infer for which purpose the payment is done. The merge avoidance mechanism integrated

in our framework can help reduce the risk of inference by splitting the transactions into several smaller ones, thus making it harder to guess the purpose of the transactions.

Additionally, car owners may infer privacy-sensitive data from the car renters' location data. For instance, the location of the car renters might reveal their driving habits, their jobs, their religion, their hobbies, or partially their social graph. Besides, when colluding, car owners can 1) merge their data about a given user to increase the quality of the inference; 2) increase the number of users in their databases thus improving their value. If a colluding external attacker successfully neutralizes the UCS, as reported in Section 6.2, car owners can freely share user data through the system and can disseminate their data to a shared database for processing.

| Attacker type | Data type | Threat | Mitigation |
|---|---|---|---|
| **Honest-but-curious** | Transaction | Linkability | Mixing |
| **Honest-but-curious** | Transaction | Identification | No address reuse |
| **Curious Mixer** | Addresses | Linkability | Mixer decentralization |
| **External attacker** | Geolocation data | Disclosure | Usage and Data Flow Control |
| **Car owner** | Renters' data | Repudiation | Data flow control, auditability |
| **Honest-but-curious** | Renters' data | Unawareness | Usage control |

**Table 3.** Threats to privacy and their mitigation

Table 3 summarizes the privacy threats for the car renters with the exception of inference attacks, here above presented. By observing the transactions, an honest-but-curious attacker may attempt to make a link between the sender and the receiver. This risk can be mitigated by using the mixing server. Furthermore, when car renters use the same address multiple times for outward transactions, they are exposed to identification (cf. section 2.3). A new address is generated in our framework for each outward transaction to forbid address reuse. Moreover, as the mixing service is decentralized, following the Sarfraz [36] procedure, a node involved in the mixing process is not able to make links between any input or output addresses. Disclosure of information is prevented by the Usage Control System, as it monitors the access to the data and prevents the dissemination to unauthorized users. The non-repudiation property is provided as the car owners are continuously monitored by the UCS. Finally, usage control provides a solution to unawareness as car renters' have to explicitly specify how they want their data to be used, which emphasizes the privacy risks they face.

### 6.2 Security threats and mitigation

The UCS is paramount for usage control and data flow transfers between the agents. It is consequently an attractive target, vulnerable to specific attacks which can be partially mitigated [20]. The proposed countermeasures to the security threats established using the STRIDE model (cf. section 3.4) are:

- *Spoofing*(S): legitimate users and the usage control system mutually authenticate e.g., using SSH or TLS. The channel between the different agents is considered authentic i.e., resistant to tampering;
- *Tampering*(T): the data processing is monitored by the UCS, excluding modifications on the data. However, an attacker could try to modify the binaries of the usage control system to make it ineffective [20]. This threat can be mitigated by using digital signatures [20];
- *Denial of service*(D): modern denial of service attacks are hard to mitigate, but the decentralization of the UCS, as designed in our framework, alleviates the risk, as well as mutual authentication of all the infrastructure components, e.g. using certificates [20];

In all likelihood, the car renters may damage the car, it is therefore paramount to design a compensatory measure to make sure the car owners actually get involved in the network. Indeed, as the framework provides a fair level of privacy, the car renter is encouraged to flee without paying after a material damage to the car. This is a strong deterrent to the car owners involvement in the system as it seriously worsens the benefit-risk balance. As a solution, we introduce a stake that has to be locked by the car renter during a given amount of time, like a UCON obligation

to be granted access. This obligation has to be fulfilled before, during and after access to leave time for arbitration in case of legal conflict. This principle is very similar to the Proof of Stake, but is used for access decision instead of consensus making. In Proof of Stake, smart contracts are needed to automatize both the rewards and the penalties, respectively for right or wrong behaviors. In our stake guarantee system, smart contracts can be used to withdraw the deposit or conversely to give it back to the car renters once the access and the arbitration time are over. However, smart contracts are not yet fully implemented in IOTA, and can only be used in the test network [12]. An alternative is to send the deposit to an address belonging to the Usage Control System while smart contracts are not available on the main network. Although less satisfying, this is a convenient workaround under the trusted UCS assumption.

## 7 Conclusion

In this paper, we devise a framework to guarantee simultaneously the requirements in terms of scalability and security of large scale IoT systems, as well as the privacy of the users. To do so, we rely on several technologies. IOTA guarantees high transaction processing capacities and scalability with a balanced security fitting IoT needs. Usage control empowers the users with a tool to monitor how their data are used, while coin mixing and merge avoidance introduce obfuscation on the network to protect from re-identification and inference. Using a car sharing scenario, we highlight the threats faced by the agents in using the system, and we analyze the security and privacy of our solution.

A Proof of Concept of the solution is proposed, on which we conduct performance tests to demonstrate the feasibility of our solution in the Internet of Things context. The performance tests show that the framework enables users to make transactions in an acceptable time, using devices with computational constraints in accordance with the scenario.

As soon as the version 2.0 of IOTA will be available, offering both the removal of the coordinator node and the availability of smart contracts, other perspectives will be opened for privacy with cross-chain transactions [37]. Cross-chain transactions rely on smart contracts to lock the coins simultaneously on two different blockchain networks. As IOTA mixing brings two properties of interest - its free transactions and the support for decentralization - there is a significant interest to mix the coin of other blockchains on IOTA network, to avoid payments of centralized mixing fees on their own networks.

Some practical issues remain, such as the legal obligation in several countries to verify the driving license before renting a car. The car owner should be able to verify that the car renter has the right to rent the car and to drive, without the disclosure of their identity to the car owner. This should be achieved in a privacy-preserving way for the car owner, which is an orthogonal, yet paramount research question for the car sharing use case.

Scalability is the main notion used to assess the suitability of the proposed solution for the Internet of Things. However, scalability is a multi-faceted notion. In this work, we considered scalability in terms of *number of transactions*, i.e., transaction throughput.

Besides, our car rental use case focuses on access to physical objects. The concept could be taken further and applied to data-centric use cases, closer to the UCON philosophy of controlling access to data and not only to objects. Finally, our framework can be applied to any IoT use case involving large scale deployments, decentralization and a demand for high processing capacities, all requirements taken together or separately. For instance, a widespread network of vending machines could benefit from this framework, especially for its zero-fee transactions.

## Acknowledgments

This paper is supported by the Future & Ruptures program of Fondation Mines-Télécom, the Institut Mines-Télécom VP-IP Chair on Values and Policies of Personal Information (https://cvpip.wp.imt.fr) and the 3rd Programme d' Investissements d'Avenir (ANR-18-EUR-0006-02) within the framework of Energy4Climate Interdisciplinary Center (E4C) (https://www.e4c.ip-paris.fr/). It is an extended version of the article called "Bringing privacy, security and performance to the Internet of Things through usage control and blockchains" published in IFIP's Privacy and Identity Management [10].

## Conflicts of interest

The authors have no conflicts of interest to declare.

## References


1. Alshaikhli, M., Elfouly, T., Elharrouss, O., Mohamed, A., Ottakath, N.: Evolution of internet of things from blockchain to iota: A survey. IEEE Access **10**, 844–866 (2022). https://doi.org/10.1109/ACCESS.2021.3138353
2. Alwarafy, A., Al-Thelaya, K.A., Abdallah, M., Schneider, J., Hamdi, M.: A survey on security and privacy issues in edge-computing-assisted internet of things. IEEE Internet Things J. **8**(6), 4004–4022 (2021). https://doi.org/10.1109/JIOT.2020.3015432, https://doi.org/10.1109/JIOT.2020.3015432
3. Ayoub, O., De Sousa, A., Mendieta, S., Musumeci, F., Tornatore, M.: Online virtual machine evacuation for disaster resilience in inter-data center networks. IEEE Transactions on Network and Service Management **18**(2), 1990–2001 (2021). https://doi.org/10.1109/TNSM.2021.3056766
4. Babil, G.S., Mehani, O., Boreli, R., Kaafar, M.: On the Effectiveness of Dynamic Taint Analysis for Protecting against Private Information Leaks on Android-based Devices. In: 2013 Int. Conference on Security and Cryptography (SECRYPT). pp. 1–8 (2013)
5. Bowe, H.S., Hornby, T., Wilcox, N.: Zcash Protocol Specification (2016), https://github.com/zcash/zips/blob/main/protocol/protocol.pdf
6. Cha, S., Hsu, T., Xiang, Y., Yeh, K.: Privacy Enhancing Technologies in the Internet of Things: Perspectives and Challenges. IEEE Internet of Things Journal **6**(2), 2159–2187 (2019)
7. Christidis, K., Devetsikiotis, M.: Blockchains and Smart Contracts for the Internet of Things. IEEE Access **4**, 2292–2303 (2016)
8. Da Silva, L.F., Lima, J.V.F.: An Evaluation of Cassandra NoSQL Database on a Low-power Cluster. In: Int. Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW). pp. 9–14 (2021). https://doi.org/10.1109/SBAC-PADW53941.2021.00012
9. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering **16**(1), 3–32 (03 2011)
10. Denis, N., Chabridon, S., Laurent, M.: Bringing privacy, security and performance to the internet of things through usage control and blockchains. In: Friedewald, M., Krenn, S., Schiering, I., Schiffner, S. (eds.) Privacy and Identity Management. Between Data Protection and Security - 16th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2.2 International Summer School, Privacy and Identity 2021, Virtual Event, August 16-20, 2021, Revised Selected Papers. IFIP Advances in Information and Communication Technology, vol. 644, pp. 57–72. Springer (2021). https://doi.org/10.1007/978-3-030-99100-5_6, https://doi.org/10.1007/978-3-030-99100-5_6
11. Dorri, A.: A Scalable Lightweight Blockchain-based Framework for IoT Security and Anonymity. Ph.D. thesis, UNSW, http://handle.unsw.edu.au/1959.4/65030 (2020)
12. Evaldas Drąsutis: IOTA Smart Contracts (2021), https://files.iota.org/papers/ISC_WP_Nov_10_2021.pdf
13. Fedrecheski, G., Rabaey, J.M., Costa, L.C.P., Calcina Ccori, P.C., Pereira, W.T., Zuffo, M.K.: Self-sovereign identity for IoT environments: A perspective. In: 2020 Global Internet of Things Summit (GIoTS). pp. 1–6 (2020). https://doi.org/10.1109/GIOTS49054.2020.9119664
14. Godik, S., Moses, T.: eXtensible Access Control Markup Language (XACML). OASIS Standard (01 2003)
15. Gramoli, V.: From blockchain consensus back to byzantine consensus. Future Gener. Comput. Syst. **107**, 760–769 (2020). https://doi.org/10.1016/J.FUTURE.2017.09.023, https://doi.org/10.1016/j.future.2017.09.023
16. Haque, A.B., Islam, A.K.M.N., Hyrynsalmi, S., Naqvi, B., Smolander, K.: Gdpr compliant blockchains–a systematic literature review. IEEE Access **9**, 50593–50606 (2021). https://doi.org/10.1109/ACCESS.2021.3069877
17. Harvan, M., Pretschner, A.: State-Based Usage Control Enforcement with Data Flow Tracking using System Call Interposition. In: Int. Conf. on Network and System Security. pp. 373–380 (2009)
18. Henriksen-Bulmer, J., Jeary, S.: Re-identification Attacks—A Systematic Literature Review. Int. Journal of Info. Management **36**(6, Part B), 1184 – 1192 (2016)
19. Howard, M., Lipner, S.: The security development lifecycle, vol. 8. Microsoft Press Redmond (2006)
20. Kelbert, F., Pretschner, A.: Data Usage Control for Distributed Systems. ACM Trans. Priv. Secur. **21**(3) (Apr 2018)
21. Khan, M., et al.: BlockU: Extended usage control in and for Blockchain. Expert Systems **37** (01 2020)
22. Martin, H., Christoph, F.: The Unreasonable Effectiveness of Address Clustering. IEEE UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld (2016)
23. Myers, A.C., Liskov, B.: A Decentralized Model for Information Flow Control. In: ACM Symp. on Operating Systems Principles. pp. 129–142 (1997)

24. Ogunniye, G., Kökciyan, N.: A survey on understanding and representing privacy requirements in the internet-of-things. J. Artif. Intell. Res. **76**, 163–192 (2023). https://doi.org/10.1613/JAIR.1.14000, https://doi.org/10.1613/jair.1.14000
25. Palm, E., Schelén, O., Bodin, U.: Selective blockchain transaction pruning and state derivability. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). pp. 31–40 (2018). https://doi.org/10.1109/CVCBT.2018.00009
26. Park, J., Sandhu, R.: The UCON ABC Usage Control Model. ACM Trans. Inf. Syst. Secur. **7**(1), 128–174 (Feb 2004)
27. Popov, S.: The Tangle (2017), https://iotatoken.com/IOTA_Whitepaper.pdf
28. Popov, S.: The Coordicide (2020), https://files.iota.org/papers/Coordicide_WP.pdf
29. Qin, X., Huang, Y., Yang, Z., Li, X.: A Blockchain-based Access Control Scheme with Multiple Attribute Authorities for Secure Cloud Data Sharing. Journal of Systems Architecture p. 101854 (2020)
30. Raghav, Andola, N., Venkatesan, S., Verma, S.: PoEWAL: A lightweight consensus mechanism for blockchain in IoT. Pervasive and Mobile Computing **69**, 101291 (2020)
31. Rizos, A., Bastos, D., Saracino, A., Martinelli, F.: Distributed UCON in CoAP and MQTT Protocols. In: ESORICS Int. Workshops, CyberICPS, SECPRE, SPOSE, and ADIoT. LNCS, vol. 11980, pp. 35–52. Springer (2019)
32. Rožman, N., Corn, M., Škulj, G., Diaci, J., Podržaj, P.: Scalability solutions in blockchain-supported manufacturing: A survey. Strojniški vestnik - Journal of Mechanical Engineering **68**, 585–609 (10 2022). https://doi.org/10.5545/sv-jme.2022.355
33. van Saberhagen, N.: Cryptonote Monero Whitepaper (2013), https://github.com/monero-project/research-lab/blob/master/whitepaper/whitepaper.pdf
34. Salimitari, M., Joneidi, M., Chatterjee, M.: AI-Enabled Blockchain: An Outlier-Aware Consensus Protocol for Blockchain-Based IoT Networks. In: 2019 IEEE Global Communications Conference (GLOBECOM). pp. 1–6 (2019)
35. Salimitari, M., Chatterjee, M., Fallah, Y.P.: A Survey on Consensus Methods in Blockchain for Resource-constrained IoT Networks. Internet of Things **11**, 100212 (2020)
36. Sarfraz, U., Alam, M., Zeadally, S., Khan, A.: Privacy Aware IOTA Ledger: Decentralized Mixing and Unlinkable IOTA Transactions. Computer Networks **148**, 361–372 (2019)
37. Shadab, N., Houshmand, F., Lesani, M.: Cross-chain Transactions. In: 2020 IEEE Int. Conference on Blockchain and Cryptocurrency (ICBC). pp. 1–9 (2020)
38. Shi, N., Tang, B., Sandhu, R., Li, Q.: DUCE: Distributed Usage Control Enforcement for Private Data Sharing in Internet of Things. In: Data and Applications Security and Privacy XXXV (DBSec). Springer (2021)
39. Silvano, W.F., Marcelino, R.: IOTA Tangle: A Cryptocurrency to Communicate Internet-of-Things Data. Future Generation Computer Systems **112**, 307–319 (2020)
40. Simões, J.E., Ferreira, E., Menasché, D.S., Campos, C.A.V.: Blockchain Privacy Through Merge Avoidance and Mixing Services: a Hardness and an Impossibility Result. SIGMETRICS Perform. Evaluation Rev. **48**(4), 8–11 (2021)
41. Tennant, L.: Improving the Anonymity of the IOTA Cryptocurrency (2017), https://laurencetennant.com/papers/anonymity-iota.pdf
42. General Data Protection Regulation (2018), http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679
43. Wuyts, K., Joosen, W.: Linddun privacy threat modeling: a tutorial (2015), https://www.linddun.org/publications