INSTITUT POLYTECHNIQUE DE PARIS

NNT : 20XXIPPAXXXX

Thèse de doctorat

TELECOM SudParis

IP PARIS

For a Private and Secure Internet of Things with Usage Control and Distributed Ledger Technology

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 Dénomination (EDIPP)
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Palaiseau, le Date, par

NATHANAËL DENIS

Composition du Jury :

Philippe Pucheral Professeur (HDR), Université de Versailles/St-Quentin	Président
Vincent Roca Chercheur (HDR), INRIA-Privatics	Rapporteur
Mawloud Omar Professeur (HDR), Université Bretagne Sud	Rapporteur
Sara Tucci Chef de laboratoire, CEA-LIST	Examinatrice
Maryline Laurent Professeure, Télécom SudParis	Directrice de thèse
Sophie Chabridon Directrice d'Études, Télécom SudParis	Encadrante de thèse

"Civilization is the progress toward a society of privacy. The savage's whole existence is public, ruled by the laws of his tribe. Civilization is the process of setting man free from men. "

— Ayn Rand, *The Fountainhead*, 1943

Abstract

IoT devices represent one of the major targets for malicious activities. The grounds for this are manifold: first, to reduce the cost of security, manufacturers may sell vulnerable products, leaving users with security concerns. Second, many IoT devices have performance constraints and lack the processing power to execute security software. Third, the heterogeneity of applications, hardware, and software widens the attack surface. As a result, IoT networks are subject to a variety of cyber threats. To counter such a variety of attacks, the IoT calls for security and privacy-preserving technologies. For privacy concerns, *usage control* grants the users the power to specify how their data can be used and by whom. Usage control extends classic access control by introducing *obligations*, i.e., actions to be performed to be granted access, and *conditions* that are related to the system state, such as the network load or the time. This thesis aims at providing answers to the challenges in the Internet of Things in terms of performance, security and privacy. To this end, *distributed ledger technologies* (DLTs) are a promising solution to Internet of Things constraints, in particular for micro-transactions, due to the decentralization they provide. This leads to three related contributions: 1. a framework for zero-fee privacy-preserving transactions in the Internet of Things designed to be scalable; 2. an integration methodology of usage control and distributed ledgers to enable efficient protection of users' data; 3. an extended model for data usage control in distributed systems, to incorporate decentralized information flow control and IoT aspects. A proof of concept of the integration (2) has been designed to demonstrate feasibility and conduct performance tests. It is based on IOTA, a distributed ledger using a directed acyclic graph for its transaction graph instead of a blockchain. The results of the tests on a private network show an approximate 90% decrease of the time needed to push transactions and make access decisions in the integrated setting.

Résumé

Les objets connectés représentent l'une des principales cibles de la cybercriminalité. Les raisons en sont multiples : d'abord, pour des raisons commerciales, les fabricants peuvent vendre des produits vulnérables qui posent des problèmes de sécurité. Deuxièmement, de nombreux appareils IoT sont soumis à des contraintes de performance et ne disposent pas de la puissance nécessaire pour exécuter des logiciels de sécurité. Enfin, l'hétérogénéité des applications, du matériel et des logiciels élargit la surface d'attaque. Pour parer à ces menaces, l'IoT a besoin de technologies de sécurité et de préservation de la vie privée sur mesure.

En ce qui concerne la protection de la vie privée, *le contrôle d'usage* donne aux utilisateurs la possibilité de spécifier comment leurs données peuvent être utilisées et par qui. Le contrôle d'usage étend le contrôle d'accès classique en introduisant des *obligations*, c'est-à-dire des actions à effectuer pour obtenir l'accès, et des *conditions* qui sont liées à l'état du système, comme la charge du réseau ou le temps. Cette thèse vise à apporter des réponses aux défis de l'internet des objets en termes de performance, de sécurité et de respect de la vie privée. Pour cela, les registres distribués (DLT) constituent une solution prometteuse aux contraintes de l'internet des objets, en particulier pour les micro-transactions, notamment par leur caractère décentralisé. Cela se traduit par trois contributions : 1. un ensemble de technologies pour des transactions sans frais préservant la vie privée, conçu pour passer à l'échelle ; 2. une méthode d'intégration du contrôle de l'utilisation et des registres distribués pour permettre une protection efficace des données des utilisateurs ; 3. un modèle étendu pour le contrôle d'usage dans les systèmes distribués, afin d'y ajouter le contrôle de flux décentralisé et les aspects liés à l'internet des objets. Une preuve de concept de l'intégration (2) a été mise en place pour démontrer la faisabilité et effectuer des tests de performance. Il s'appuie sur IOTA, un registre distribué qui utilise un graphe orienté acyclique pour son graphe de transactions au lieu d'une *blockchain*. Les résultats des tests de performance sur un réseau privé montrent une diminution d'environ 90% du temps nécessaire pour effectuer des transactions et pour évaluer des politiques de contrôle d'usage, dans le cas où ce dernier est intégré au réseau.

Synopsis en français

This synopsis is provided in compliance with the 1994 law on the use of the French language. It outlines the structure of the thesis and summarizes its chapters and contributions.

Ce synopsis est fourni en conformité avec la loi de 1994 relative à l'emploi de la langue française. Il reprend la structure de la thèse et résume les chapitres et les contributions de la thèse.

Introduction

L'internet des objets (IdO) est l'interconnexion entre l'Internet et les objets (connectés), et prend une importance croissante avec le nombre de d'objets connectés qui augmente. Le nombre d'appareils actifs est estimé à 15,1 milliards en 2023 (cf. Figure 1.1), se connectant et échangeant des données via différents réseaux de communication [Insights, 2023]. Avec un nombre prévu de dispositifs actifs qui devrait atteindre 29,4 milliards d'ici 2030 [Insights, 2023], les exigences en matière de performance, de sécurité et de respect de la vie privée dans l'internet des objets seront de plus en plus pressantes. De nombreux domaines d'activité seront touchés, notamment la santé, l'industrie, les *smart cities*, la logistique, l'agriculture ou encore la construction (cf. Figure 1.1).

Risques liés à la sécurité et à la protection de la vie privée. L'internet des objets offre de nouveaux moyens de collecter des données, de les analyser et de prendre des décisions pour développer des applications qui permettent de répondre aux besoins des utilisateurs, parfois même de les anticiper. La nature sans précédent de l'IdO a des conséquences sur les données générées, qui sont très détaillées et potentiellement intrusives et en quantité importante. Pour ces raisons, les données sont particulièrement à risque pour la vie privée, ce qui nécessite des mécanismes efficaces de protection. En outre, l'IdO présente des caractéristiques uniques en raison de l'hétérogénéité entre les objets et de la grande quantité d'objets qu'il peut interconnecter, ce qui en fait un système distribué d'une ampleur sans précédent. Il en résulte plusieurs défis en matière de sécurité, car certains objets connectés, par exemple les capteurs, peuvent ne pas disposer de la puissance de calcul ou du stockage nécessaires pour mettre en oeuvre des solutions basées sur la cryptographie. En outre, les dispositifs IdO peuvent présenter

des failles de sécurité dans leur logiciel ou leurs composants matériels. Ces vulnérabilités peuvent être exploitées pour prendre le contrôle des appareils, perturber leur fonctionnement ou lancer des attaques sur d'autres appareils ou réseaux. [Omolara et al., 2022].

Réglementation. Le règlement général sur la protection des données (RGPD) de l'Union Européenne [The European Parliament and the Council of the European Union, 2018] introduit plusieurs obligations légales, parmi lesquelles la protection des données par défaut, la gestion du consentement de l'utilisateur et la définition des responsabilités. En effet, les entreprises - en dehors de l'intérêt légitime - doivent demander explicitement à l'utilisateur un consentement clair et explicite avant toute collecte de données. Une entreprise doit pouvoir prouver à tout moment que le traitement des données est toujours effectué de manière légitime, soit en fonction de l'intérêt de l'utilisateur, soit en fonction de son consentement ou soit dans un but légitime pour l'entreprise.

Exigences pour un internet des objets sécurisé. En raison des risques propres à l'internet des objets, les exigences pour un internet des objets sûr et préservant la vie privée sont les suivantes. Premièrement, la solution doit prendre en compte *les objets aux capacités restreintes*, et veiller à ce que les mesures pour parer les menaces pour la sécurité et la vie privée tiennent compte des capacités réelles de l'IdO. Par exemple, les solutions basées sur la cryptographie sont souvent inapplicables aux objets connectés les moins puissants. Deuxièmement, les données étant sensibles du point de vue de la vie privée, il est indispensable pour l'utilisateur d'appliquer un *contrôle d'accès* sur ses données et de *contrôler l'usage* qui en est fait. Troisièmement, pour des raisons de performance, de sécurité et de confidentialité, la décentralisation est un aspect important de l'internet des objets. Les solutions centralisées peuvent espionner les données des utilisateurs [Qin et al., 2020] et être vulnérables aux fuites de données accidentelles ou aux attaques externes [Qin et al., 2020]. Les dénis de service peuvent également être un sujet de préoccupation, car l'infrastructure physique peut être endommagée, par exemple en raison d'un incendie ou d'une catastrophe naturelle [Ayoub et al., 2021]. En outre, la centralisation nuit aux performances, notamment en augmentant les coûts de déploiement et de maintenance [Salimitari et al., 2020].

Utilisation des registres distribués pour l'Internet des Objets. Les registres distribués (DLTs), en raison de leurs propriétés, constituent une solution prometteuse pour répondre aux exigences de sécurité de l'internet des objets. Les DLTs offrent en effet un certain degré de *décentralisation* et sont *immutables* ce qui est utile pour un large éventail d'applications de sécurité allant de la gestion de la confiance [Liu et al., 2023] aux transactions anonymes et sécurisées [Bothra et al., 2023]. Les DLTs peuvent également être utilisés pour fournir un contrôle d'accès de manière automatisée et transparente à l'aide de *contrats intelligents* [Bao et al., 2023]. Toutefois, les technologies de registres distribués ne sont pas toujours conçues pour répondre aux exigences de l'internet des

objets. Les exigences en matière de performance, de sécurité et de confidentialité nécessitent des registres distribués adaptés, qui permettent des transactions anonymes, efficaces et peu coûteuses pour les objets connectés.

Contrôle d'usage. Le contrôle d'usage est une technologie qui permet de contrôler l'utilisation qui est faite des données. Il permet d'accorder ou de refuser l'accès en fonction d'autorisations, d'*obligations*, qui doivent être remplies pour obtenir l'accès, et enfin de *conditions* liées à l'état du système. En conséquence, le contrôle d'usage est une technologie d'intérêt pour la vie privée des utilisateurs puisqu'ils peuvent décider qui peut accéder à leurs données et comment celles-ci sont utilisées. Mais le problème de l'utilisation conjointe des registres distribués et du contrôle d'usage se pose. Des travaux existants proposent d'intégrer le contrôle d'usage dans des *blockchains* privées, mais ce type de *blockchain* n'est pas adapté aux cas d'usage impliquant un très grand nombre d'objets.

Objectifs de recherche. Ayant identifié les problématiques de performance, de sécurité, et de protection de la vie privée ainsi que des technologies pouvant potentiellement y répondre, nous pouvons formuler des objectifs de recherche qui forment la ligne conductrice des travaux de recherche présentés dans cette thèse. Les objectifs de recherche sont les suivants :

- *Objectif 1* : Permettre des transactions gratuites ou à très bas coût, respectueuse de la vie privée pour répondre aux besoins de l'internet des objets ;
- *Objectif 2* : Identifier les registres distribués adaptés aux contraintes de l'IdO ;
- *Objectif 3* : Mettre en place une méthodologie pour intégrer efficacement le contrôle d'usage et les registres distribués adaptés ;
- *Objectif 4* : Identifier les concepts utiles à l'internet des objets et qui ne sont pas traités dans l'état de l'art du formalisme du contrôle d'usage dans les systèmes distribués.

En plus de ces quatre objectifs, nous ajoutons d'autres objectifs méthodologiques, pour la validation des résultats présentés :

- *Objectif 5* : Analyser les aspects sécurité et protection de la vie privée des méthodes proposées, à l'aide d'une évaluation des menaces pour la sécurité et la vie privée ;
- *Objectif 6* : Valider la faisabilité des méthodes proposées à l'aide d'une preuve de concept.

Dans ce travail de thèse, les solutions proposées prendront également en compte que les scénarios de l'internet des objets peuvent impliquer de très nombreux objets

(*large-scale networks*), créant des problématiques de passage à l'échelle qui limitent l'utilisation de certaines technologies existantes.

Contributions. Pour répondre à ces objectifs de recherche, ce travail de thèse propose les contributions suivantes :

- un *framework* pour répondre aux besoins de vie privée, de sécurité et de performances de l'internet des objets (chapitre 3). Le *framework* s'appuie en particulier sur la technologie IOTA, un registre distribué utilisant un graphe orienté acyclique pour effectuer des transactions sans frais, au lieu d'une *blockchain* (*Objectif 1* et *Objectif 2*) ;
- une méthode d'intégration du contrôle d'usage avec les registres distribués (chapitre 4). Les registres appropriés sont identifiés en fonction des paramètres adaptés, et une preuve de concept est mise en place pour évaluer les performances (*Objectif 3*) ;
- une extension du modèle pour le contrôle d'usage et du flux d'information décentralisé (DIFC) dans les systèmes distribués (chapitre 5), en introduisant une politique définie conjointement sur les données personnelles collectives et la disponibilité des systèmes présents dans le réseau (*Objectif 4*).

Contexte scientifique

Cette partie introduit toutes les technologies utiles à la compréhension de ce document, ainsi que leurs caractéristiques en termes de performance et de protection de la vie privée quand cela est nécessaire. Nous résumons rapidement cette partie en présentant de manière succincte les technologies en question.

Le contrôle d'usage. Le contrôle d'usage est une extension du contrôle d'accès, décrivant la manière dont les données peuvent être utilisées après l'accès initial. Il a été proposé pour la première fois par Sandhu et Park sous la forme du modèle *UCON* [Park and Sandhu, 2004]. Ce modèle introduit la *mutabilité* des attributs, ainsi que de nouveaux facteurs de décision décrits par le modèle ABC (Figure 2.1) : *Autorisations, oBligations, Conditions*. Les *attributs mutables* sont modifiés à la suite d'un accès, tandis que les *attributs immutables* sont modifiés à la suite d'une action administrative. Les obligations sont des conditions à remplir par le sujet pour se voir accorder l'accès. Les conditions sont des exigences indépendantes du sujet et liées au système, par exemple l'heure. Les attributs étant mutables, les obligations et les conditions peuvent être effectuées avant ou pendant l'accès.

Si le contrôle d'usage impose des limites sur la façon dont les données sont utilisées, il ne fournit aucune garantie sur la propagation de l'information. L'utilisation

d'un mécanisme dédié de contrôle de flux des données (IFC) est crucial pour la sécurité de l'information afin de prévenir les fuites de données. Un tel mécanisme est utile aussi pour le contrôle d'usage, car les informations peuvent potentiellement être diffusées en dehors de la zone de surveillance du système de contrôle. Dans les systèmes de contrôle d'usage modernes, ces deux technologies de contrôle de flux et de contrôle d'usage sont donc utilisées conjointement.

Les registres distribués. Les registres distribués (DLT) constituent la deuxième technologie d'intérêt pour ces travaux de thèse. Leur caractère distribué est bénéfique pour les performances et la sécurité du réseau, et ils sont aussi étudiés de manière active dans la littérature scientifique pour la protection de la vie privée des utilisateurs [Rifi et al., 2017, Ma et al., 2021, Goyat et al., 2022, Rajasekaran et al., 2023, Bao et al., 2023].

Les registres distribués se distinguent notamment par une *méthode de consensus*, qui impacte significativement les performances et la sécurité du réseau. Les deux méthodes de consensus principales sont la preuve de travail (PoW) utilisée dans la *blockchain* Bitcoin, et la preuve d'enjeu (PoS) utilisée dans son principal concurrent Ethereum. La preuve de travail s'appuie sur un défi calculatoire difficile, dont le gagnant obtient le droit d'écrire le prochain bloc dans le réseau et est récompensé financièrement pour ses efforts. Ce processus, malgré ses apports en termes de sécurité, est très gourmand en ressources et consomme beaucoup d'énergie (voir les illustrations C.2 et C.3). La preuve d'enjeu cherche à atténuer ce coût en utilisant un enjeu économique sous la forme d'un montant en cryptomonnaie, qui augmente proportionnellement les chances d'être choisi comme mineur. En plus des variantes de la preuve d'enjeu - preuve d'enjeu déléguée, preuve d'enjeu liquide...-, il existe aussi des méthodes de consensus pour les *blockchains* dites privées, dont l'accès au registre est contrôlé. Les deux méthodes principales sont la preuve du temps écoulé (PoET) et la tolérance pratique aux fautes byzantines (PBFT). Dans un réseau basé sur la méthode PoET, chaque noeud participant du réseau doit attendre une période de temps aléatoire, et le premier à terminer est désigné comme mineur. PBFT est un algorithme de consensus introduit à la fin des années 90 par Barbara Liskov et Miguel Castro [Castro and Liskov, 1999] conçu pour fonctionner efficacement dans les systèmes asynchrones. Le consensus sur les transactions est obtenu via des échanges nombreux entre les noeuds, ce qui fait que cette méthode ne passe pas à l'échelle et ne peut être utilisée dans des *blockchains* publiques.

Si les *blockchains* sont les exemples les plus connus de registres distribués pour les cryptomonnaies, la notion de DLT est plus large et inclut plusieurs autres technologies d'intérêt. Tout d'abord, un registre distribué peut être complètement déconnecté de la notion de cryptomonnaie, comme par exemple les bases de données distribuées. Certaines cryptomonnaies n'utilisent pas de *blockchains* pour leur graphe de transactions, mais des structures mathématiques différentes. Les alternatives les plus utilisées dans les cryptomonnaies sont les graphes orientés acycliques (DAG) et les graphes de

hachage (*hashgraphs*).

Protection de la vie privée dans les registres distribués. Les *blockchains* publiques ne demandent pas d'informations d'identification pour effectuer une transaction, et un pseudonyme est utilisé. Cependant, l'accès aux transactions et à leur contenu n'est pas limité. Les transactions révèlent des informations sur les différentes parties impliquées et créent des risques d'inférence. Des tiers intéressés collectent et analysent automatiquement ces informations, pour plusieurs raisons incluant l'analyse à des fins judiciaires [Harrigan and Fretter, 2016]. Par défaut, les *blockchains* publiques n'offrent que le pseudonymat, ou l'anonymat si et seulement si le lien entre le pseudonyme et l'identité réelle de l'utilisateur n'est pas possible. Cependant, plusieurs comportements facilitent considérablement la ré-identification, notamment la réutilisation d'une même adresse pour effectuer plusieurs transactions.

Des méthodes existent pour protéger l'identité des utilisateurs. L'une des plus utilisées est le mélangeur de cryptomonnaie. Les mélangeurs de cryptomonnaie permettent d'empêcher le traçage des utilisateurs qui envoient et ceux qui reçoivent de la cryptomonnaie. La facilité d'intégrer les nouveaux utilisateurs et la compatibilité avec les technologies existantes sans modification sont des caractéristiques attrayantes de ce service. Bien qu'utiles pour la préservation de la vie privée, les mélangeurs de cryptomonnaie sont confrontés à plusieurs défis techniques tels que la décentralisation et le coût du service, car le mélangeur génère lui-même des nouvelles transactions pour lesquelles il doit souvent payer. Le mélangeur s'appuie souvent sur un mécanisme de *merge avoidance*, en séparant la transaction en plusieurs sous-transactions pour éviter d'inférer le motif de la transaction.

Performances des registres distribués. Comme les méthodes de consensus sont gourmandes en ressources et en temps, les performances des *blockchains* et des registres distribués sont beaucoup étudiées dans la littérature scientifique [Brotsis et al., 2021, Fan et al., 2021, Chen et al., 2022, Okegbile et al., 2022]. Les performances constituent la base utilisée pour comparer les méthodes de consensus entre elles et suggérer ou exclure l'utilisation d'une méthode de consensus pour un cas d'usage précis. Les critères pour mesurer les performances sont les suivants. Le *débit*, souvent mesuré en transactions par secondes, et qui traduit la capacité du réseau à traiter beaucoup de transactions simultanément. La *latence*, qui est la mesure du temps nécessaire à la validation d'une transaction. Le *passage à l'échelle*, une notion qui peut s'exprimer de plusieurs manières - en termes de nombre d'objets connectés, de transactions simultanées...- et qui est très liée à la décentralisation. Finalement, les *surcoûts* liés au stockage ou à la *communication* entre les noeuds du réseau, qui impactent la taille du registre comme la possibilité de passer à l'échelle.

Infrastructure logicielle pour répondre aux besoins de l'IdO

Dans ce chapitre qui correspond à la première contribution, un *framework* est proposé pour permettre des micro-transactions dans l'internet des objets (Objectif 1) respectant les besoins de performance, de sécurité et de respect de la vie privée. En particulier, le *framework* permet de contrôler l'accès aux dispositifs physiques et l'usage des données, n'impose pas de frais de transaction, pour permettre les micro-transactions et est respectueux de la vie privée pour les deux participants à la transaction.

Le cadre proposé se compose des éléments suivants (cf. Figure 3.2) :

1. La technologie IOTA, en tant que registre distribué approprié pour répondre aux exigences de performance de l'IdO et au besoin de transactions sans frais ;
2. IOTA Access, un logiciel open-source utilisé pour contrôler l'accès aux appareils de l'IdO. Il est développé par la Fondation IOTA pour compléter la technologie IOTA ;
3. un système de contrôle d'usage, pour contrôler l'utilisation et la dissémination des données dans le système. Le système de contrôle d'usage repose sur l'exécution d'un environnement de confiance (TEE) présent sur l'appareil de l'utilisateur contrôlé ;
4. un mélangeur de cryptomonnaie décentralisé couplé au *merge avoidance* (cf. section 2.3.3), pour l'obfuscation des transactions et améliorer la vie privée des utilisateurs.

Intérêt de l'utilisation de IOTA. IOTA est un registre distribué utilisant un graphe orienté acyclique plutôt qu'une *blockchain* pour son graphe de transactions. Il a été conçu pour l'internet des objets [Popov, 2017] et possède de nombreux atouts pour répondre aux besoins en termes de performance. D'abord, IOTA ne possède pas de *mineurs* responsables de la création des nouvelles transactions, qui est déléguée aux utilisateurs eux-mêmes. Cela permet d'avoir des transactions sans frais, et un débit plus élevé grâce à la structure du graphe qui permet des insertions simultanées à plusieurs endroits du graphe, contrairement aux *blockchains*. IOTA permet aussi aux objets avec des contraintes sur les capacités de calcul ou de stockage de contribuer au réseau, en partie en déléguant certaines opérations. Il faut noter que dans l'état actuel (IOTA 1.0), une partie de ces avantages ne sont pas encore visibles en pratique, notamment à cause de la présence du noeud *coordinateur*. Ce composant centralisé est chargé de valider les transactions régulièrement en posant des jalons (*milestones* en anglais), mais réduit le débit et est un point de défaillance unique qui permet à la fondation IOTA d'arrêter le réseau si elle le souhaite.

Validation de la solution. Conformément aux objectifs de recherche *Objectif 5* et *Objectif 6*, le *framework* proposé est validé par :

- une analyse de performance pour démontrer la faisabilité de la solution, en s'appuyant sur une preuve de concept. Les tests prennent en compte des optimisations en utilisant le processus d'intégration décrit dans le chapitre 4 ;
- une analyse des risques sur la vie privée des utilisateurs, en s'appuyant sur un cas d'usage de location de voitures entre particuliers. La méthode d'analyse de risques LINDDUN [Wuyts et al., 2018] est utilisée pour identifier précisément les risques sur les données personnelles dans le cadre de ce scénario ;
- une analyse des risques de sécurité, également dans le cadre du scénario sur la location de voitures, en utilisant la méthode STRIDE [Howard and Lipner, 2006]

Ces analyses montrent que le nécessaire pour faire des transactions et prendre des décisions vis-a-vis des politiques de contrôle d'usage est réaliste, et que les outils utilisés pour parer les menaces de sécurité et de vie privée sont efficaces quand ils sont utilisés conjointement.

Intégration du contrôle d'usage et des registres distribués

Le chapitre 4 propose d'intégrer finement le système de contrôle d'usage dans les technologies de registres distribués. Le but de cette intégration est de faire fonctionner les deux technologies - le contrôle d'usage et les registres distribués - en synergie pour augmenter leur efficacité. Intuitivement, les registres distribués gagnent à avoir plus de noeuds dans le réseau, car cela augmente le nombre d'utilisateurs qui vérifient la validité des transactions. Dans le cas des registres basés sur un DAG, augmenter le nombre d'utilisateurs augmente souvent en conséquence le nombre de transactions, ce qui permet en retour d'augmenter le débit. Le système de contrôle d'usage dispose en contrepartie d'une version locale du registre de transactions, sur lequel il peut s'appuyer pour traiter les politiques. Des travaux existants [Khan et al., 2020, Shi et al., 2021, Ma et al., 2021] proposent d'utiliser les registres distribués avec le contrôle d'usage, mais :

- ils se limitent aux *blockchains* privées et excluent les registres publics, ce qui empêche son utilisation pour les cas d'usage IdO impliquant un grand nombre d'objets ;
- aucun travail de généralisation de ce processus d'intégration a été proposé, ce qui ne permet pas de voir quels sont les registres appropriés pour les différents les cas d'usage.

Identification des technologies adaptées. Cette contribution propose donc de différencier les registres entre eux en utilisant plusieurs critères : la méthode de consensus, la méthode de construction du graphe de transactions et la méthode utilisée pour inciter les utilisateurs à participer au fonctionnement du réseau. Ces trois critères ont un impact sur deux paramètres, la *décentralisation* et l'*équitabilité*, qui garantit que tous les objets y compris ceux avec des capacités limitées peuvent contribuer au réseau de manière significative, typiquement dans le cadre d'une élection. L'analyse des différents types de registre conduit à la conclusion que les *blockchains* privées et les registres basés sur des DAGs (privés et publics) sont particulièrement adaptés pour intégrer le contrôle d'usage à leur réseau.

Analyse de performance. Pour valider le fait que l'intégration a des effets positifs sur le contrôle d'usage, et conformément à l'objectif de recherche *Objectif 6*, les gains en performance sont évalués dans le cadre d'une preuve de concept. Contrairement aux tests de performance effectués dans le cadre de la première contribution (Chapitre 3), l'accent est mis dans cette partie sur la *reproductibilité des tests*. Pour cela, les tests sont réalisés sur un réseau IOTA privé, en faisant varier le nombre de noeuds (de 3 à 10 noeuds). Les tests mesurent la différence entre le temps mis pour valider et transmettre une transaction au reste du réseau dans les deux configurations - avec ou sans l'intégration du contrôle d'usage en tant que noeud du réseau. Les tests montrent que l'intégration diminue jusqu'à 94% (3 noeuds) le temps nécessaire pour transmettre une transaction valide sur le réseau, accélérant grandement les prises de décisions liées aux paiements.

Analyse de risques sur la vie privée. Contrairement à la *Contribution 1*, l'analyse de privacy est ici conduite non pas pour un scénario spécifique, mais dans un cadre générique de transactions pour acheter des données. L'analyse de risques est faite en utilisant LINDDUN [Wuyts et al., 2018]. L'analyse permet de montrer que dans un cas général, le contrôle d'usage seul permet de parer 4 des 7 familles de menace de LINDDUN, et partiellement pour 6 sur 7 d'entre elles. Seule la non-répudiation n'est pas fournie, c'est-à-dire la capacité de l'utilisateur à nier des actions qui lui sont attribuées. C'est un résultat attendu, car le contrôle d'usage surveille étroitement les actions de l'utilisateur des données pour pouvoir empêcher les actions interdites.

Formalisme du contrôle d'usage dans les registres distribués

Un modèle formel du contrôle d'usage des données peut aider à garantir que les objectifs de sécurité et de confidentialité du système sont atteints en fournissant une spécification claire des politiques. Bien que cette modélisation soit régulièrement proposée dans l'état de l'art dans des contextes centralisés [Pretschner et al., 2011, Kelbert and Pretschner, 2013, Fromm, 2020], elle est moins souvent abordée dans des contextes dis-

tribués, pourtant plus adaptés à l'internet des objets. En particulier, les modèles actuels ne prennent pas en compte la possibilité pour les utilisateurs de définir des politiques sans passer par le système distribué lui-même, ni les aspects spécifiques aux réseaux IdO, où des sous-parties du réseau peuvent être momentanément déconnectées.

Contrôle de flux décentralisé. Le contrôle de flux décentralisé (DIFC) [Myers and Liskov, 1997] permet à des utilisateurs de définir des politiques sur la dissémination des données en appliquant directement des *étiquettes* sur les données, mais aussi les conteneurs des données eux-mêmes. Cela permet de ne pas passer par une entité centrale, qui peut être corrompue ou neutralisée dans le cadre d'un déni de service. Cependant, le contrôle de flux n'est pas très utilisé en pratique à cause du surcoût en développement qu'il entraîne. Cependant, cela a changé en raison de l'évolution des pratiques de développement, incluant notamment la télémétrie et la journalisation des événements, qui permettent d'utiliser DIFC dans la plupart des systèmes [Liu et al., 2022]. Il est donc devenu intéressant d'intégrer DIFC dans les modèles de contrôle d'usage pour les systèmes distribués.

Contribution aux modèles actuels. Dans le but d'intégrer DIFC au modèle de contrôle d'usage dans les systèmes distribués, il est nécessaire d'intégrer les éléments suivants dans le modèle :

- des composants du système de contrôle d'usage dédiés au traitement du contrôle de flux décentralisé, à savoir un composant pour l'étiquetage, la conversion des étiquettes en autorisations, et un composant de pré-traitement qui détermine si un élément doit être étiqueté ou non ;
- des fonctions définies formellement, qui permettent de récupérer les étiquettes associées aux données et à leur conteneurs, et de détecter les conflits qui peuvent survenir entre ces éléments ;

En plus des éléments DIFC, nous introduisons dans la contribution des éléments liés à l'état des sous-parties du système distribué. En particulier, certaines parties peuvent se déconnecter, être indisponibles si le réseau est instable, etc. Des fonctions sont formalisées pour déterminer si une partie précise du système est accessible, et par extension, si une donnée ou un conteneur est présent dans un système actuellement accessible.

Conclusion

Le chapitre de conclusion récapitule les contributions de cette thèse, avant d'introduire les limites des travaux présentés, notamment en ce qui concerne les technologies clés utilisées dans les différentes contributions. Enfin, différentes pistes de recherche sont proposées.

Limites des travaux. Nous identifions plusieurs limites dans nos travaux de thèse :

- des limites liées à l'utilisation de IOTA, qui est une technologie encore en développement. En particulier, la version actuelle de IOTA (1.0) repose encore sur un mécanisme centralisé de contrôle, le coordinateur. Cela limite le débit de transaction, et sa neutralisation ou son interruption provoquent l'arrêt du réseau ;

- la capacité à passer à l'échelle, importante dans le cadre de cette thèse où il y a potentiellement beaucoup d'objets dans le réseau, n'est pas simple à évaluer intégralement. Si le débit de transactions est un moyen d'évaluer cette capacité à passer à l'échelle, d'autres aspects existent pour mesurer cette capacité, comme le nombre de machines connectées au noeud, la taille du registre...

- des limites liées à l'adoption du contrôle d'usage dans les systèmes. Définir les politiques de contrôle d'usage nécessite d'établir des obligations et des conditions avec différents niveaux de temporalité, ce qui est complexe. Il faut parfois interagir avec d'autres mécanismes existants de contrôle d'accès ce qui limite l'utilisation du contrôle d'usage dans les systèmes existants ;

- des aspects pratiques du contrôle d'usage, qui peuvent avoir un impact significatif sur les performances et être limitants dans certains scénarios, en particulier si les objets à contrôler sont nombreux. Par ailleurs, le contrôle d'usage est partiellement déployé sur les objets contrôlés (cf. Section 2.1.2). Cela peut créer des problèmes de vie privée pour les utilisateurs accédant aux données, car le contrôle nécessite d'intercepter les processus y compris au niveau réseau. Souvent, un *Trusted Execution Environment* est déployé pour pouvoir contrôler l'utilisateur en conservant l'intégrité et la confidentialité des données, mais le TEE lui-même peut être vulnérable.

Pour compléter les travaux fournis dans le cadre du doctorat, des pistes de recherche ont été identifiées, pour renforcer la validité des solutions proposées.

Validation du modèle formel. Les extensions proposées pour compléter le modèle de contrôle d'usage dans les registres distribués pourraient faire l'objet de validation. La validation peut être expérimentale pour vérifier que les fonctions (pour déterminer l'étiquette DIFC, le statut de la connexion des parties du système...) ne sont pas trop coûteuses et donc inapplicables en pratique. Il est aussi possible de faire de la vérification de modèle pour s'assurer que les fonctions sont correctes du point de vue logique. Étant donnée la nature distribuée du modèle proposée, TLA$^+$ [Lamport, 1992] est un outil qui semble particulièrement adapté pour traduire le modèle théorique [Lazouski et al., 2010]. TLA$^+$ dispose notamment d'un outil de vérification de modèle intégré, *TLC*.

Les travaux de recherche présentés dans cette thèse ouvrent par ailleurs plusieurs perspectives. En particulier, les propriétés des DAGs sont intéressantes pour un en-

semble de sujets de recherche orthogonaux liés à l'internet des objets, que nous présentons maintenant.

Génération de politiques XACML pour les tests. Les contributions de cette thèse s'appuient sur l'évaluation de politiques de contrôle d'usage, écrites dans le langage XACML. Générer des politiques pour les tests est difficile, mais nécessaire à l'évaluation des performances du système de contrôle d'usage. Ensuite, traduire les politiques de haut niveau en politiques XACML est également fastidieux.

Plusieurs travaux dans la littérature ont développé des outils pour rendre la génération des politiques XACML plus conviviale. Bertolino *et al.* a proposé deux outils différents pour dériver les requêtes XACML permettant la génération automatique de requêtes XACML pour les tests de politiques [Bertolino et al., 2012]. De même, Xu *et al.* utilisent des *tests basés sur la mutation* où les demandes d'accès sont dérivées d'une politique originale [Xu et al., 2020]. Les travaux futurs pourraient inclure des tests basés sur une dérivation rigoureuse de la politique.

L'apprentissage fédéré basé sur IOTA. L'apprentissage fédéré est une méthode d'apprentissage automatique qui distribue l'apprentissage sur les objets sans partager les données personnelles avec le serveur central. Dans l'apprentissage fédéré, le serveur central ne fait que l'orchestration du processus d'apprentissage, et seules les mises à jour des paramètres du modèle sont partagées entre les objets et l'orchestrateur central. Le serveur central n'a pas besoin d'accéder aux données réelles pour entraîner son modèle, ce qui réduit les risques pour la vie privée.

L'apprentissage fédéré étant distribué par nature, son utilisation conjointe avec la technologie *blockchain* a fait l'objet d'une attention particulière, comme le montrent des travaux récents [Hou et al., 2021, Lee and Kim, 2021, Issa et al., 2023, Qu et al., 2023, Qammar et al., 2023]. Les *blockchains* sont utilisées dans l'apprentissage fédéré pour les raisons suivantes :

- L'utilisation des contrats intelligents pour coordonner l'apprentissage fédéré. Les contrats intelligents peuvent valider les contributions des noeuds (pour empêcher les manipulations des noeuds malveillants), calculer le modèle global ou enregistrer les performances des noeuds [Issa et al., 2023] ;

- L'amélioration de la sécurité et de la confidentialité en supprimant le serveur central [Issa et al., 2023] ;

Cependant, l'apprentissage fédéré utilisant les *blockchains* doit encore résoudre plusieurs défis pour préserver la vie privée et gérer les contraintes de l'internet des objets [Issa et al., 2023]. Les registres distribués basés sur les DAGs ne sont pas mentionnés dans la littérature comme une solution potentiellement plus performante que les *blockchains* pour l'apprentissage fédéré [Issa et al., 2023, Qu et al., 2023]. IOTA pourrait être mis à profit car il peut intégrer efficacement les objets connectés dans son réseau, ce qui

n'est pas le cas des *blockchains* en général. Les résultats de ce travail de thèse pourraient donc également être étendus à l'apprentissage fédéré.

Identité numérique auto-souveraine basée sur IOTA. L'identité auto-souveraine ou SSI est une approche dans laquelle les sujets contrôlent pleinement leurs propres identités numériques [Fedrecheski et al., 2020] contrairement aux solutions actuelles d'identité numérique qui sont centralisées et posent des problèmes de confidentialité et de sécurité [Fedrecheski et al., 2020].

Les identités souveraines présentent plusieurs avantages pour l'internet des objets. Les identités des utilisateurs et de leurs appareils sont stockées localement et sont divulguées de manière sélective par les utilisateurs, ce qui protège mieux la vie privée. La suppression de la nécessité d'un tiers de confiance pour gérer les identités accroît la décentralisation du réseau et supprime un point de défaillance unique du réseau [Fedrecheski et al., 2020].

Pourtant, l'adoption du paradigme SSI dans les réseaux IdO se heurte à plusieurs problèmes, techniques et non techniques, comme la standardisation. Les aspects techniques sont notamment les suivants :

- *objets avec capacités restreintes* : les objets doivent être en mesure de mettre en place les outils cryptographiques et de gérer les communications ;
- *traçabilité* : un suivi global n'est souvent pas possible sans une autorité centrale.

La technologie IOTA, en raison de sa capacité à intégrer des dispositifs contraints dans son consensus et son réseau, est une technologie prometteuse pour répondre à la première limite technique. En particulier, IOTA offre la possibilité de déployer des noeuds pour les utilisateurs. Des travaux existants ont proposé d'utiliser IOTA comme base d'un SSI [Gebresilassie et al., 2020] et IOTA lui-même possède un module pour générer des identités décentralisées [Yarger, 2020]. Gebresilassie *et al.* proposent d'utiliser les DAGs comme éléments de base d'un système de gestion des identités des noeuds du DAG, en particulier pour gérer la réputation des noeuds. Cependant, la contribution reste très évasive sur de nombreux aspects techniques clés comme les conditions d'enrôlement des noeuds dans le SSI, les propriétés de sécurité souhaitées, le contenu des transactions, l'analyse de sécurité, ce qui laisse encore beaucoup d'inconnues avant une possible mise en oeuvre au sein d'une preuve de concept.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Contents

1.1 Motivation

The Internet of Things (IoT) is a unique paradigm, with an estimated 15.1 billion active devices in 2023 (cf. Figure 1.1) connecting and exchanging data through different communication networks [Insights, 2023]. With a forecast number of active devices reaching 29.4 billion by 2030 [Insights, 2023], the requirements regarding performance, security and privacy in the Internet of Things will be increasingly pressuring. Numerous domains of activities are to be impacted, including, but not limited to, healthcare, industries, cities, logistics, agriculture or construction (cf. Figure 1.1 categories).

Security and privacy risks. The Internet of Things (IoT) is bringing new ways to collect data, analyze them and take decisions for developing applications that answer or even anticipate the users' needs. The unprecedented nature of the IoT has consequences on the data generated, which are *fine-grained* and in *quantity*.The collected data enables the monitoring of users' actions and locations while users are often unaware of the data collection. For these reasons, the data are particularly privacy-sensitive, requiring efficient privacy-preserving mechanisms. Moreover, the IoT has unique characteristics due to the extreme heterogeneity and large quantity of objects it can interconnect, as

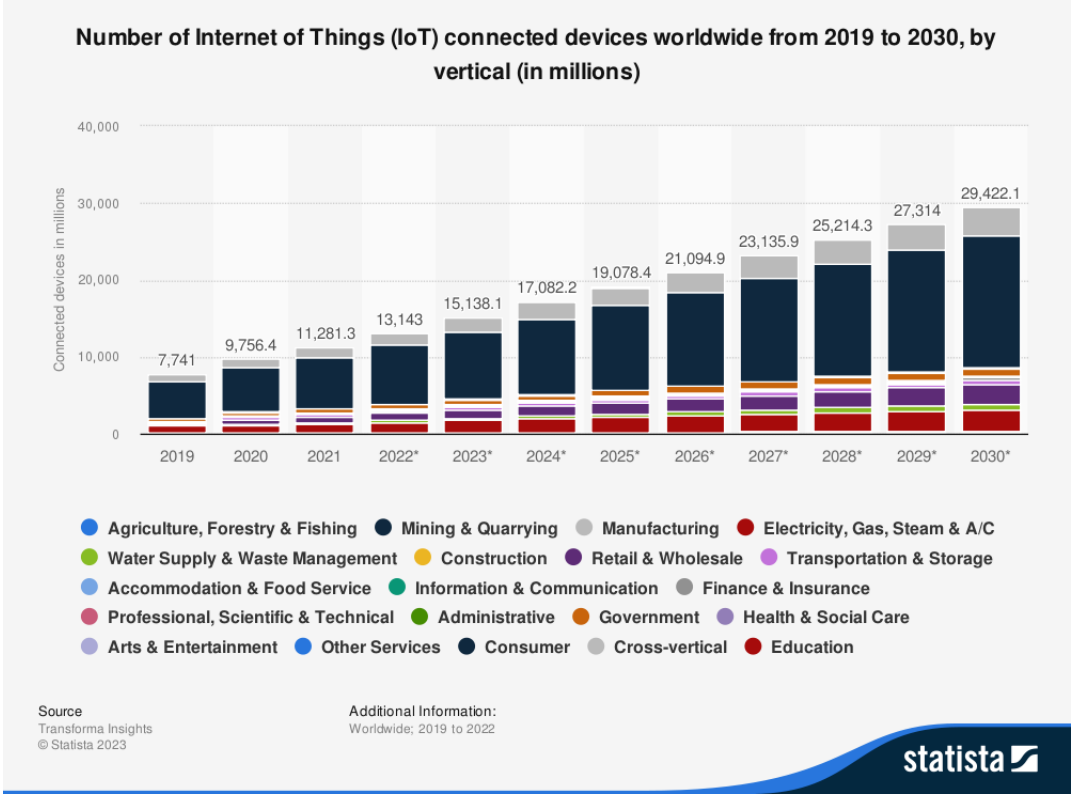


Figure 1.1 – Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030, by vertical. The single largest use case in terms of the number of Internet of Things (IoT) connected devices is consumer internet and media devices, accounting for a third of all devices worldwide in 2030. The other two largest use cases are smart grid (e.g. smart meters) and connected vehicles - Statista [Insights, 2023]

well as the spontaneous nature of their interactions, making it a distributed system of unprecedented scale. It results in several security complications, as some IoT devices, e.g., sensors, may not have the computation power or storage needed to implement cryptographic primitives. Additionally, IoT devices may have security vulnerabilities in their firmware, software, or hardware components. These vulnerabilities can be exploited to gain control over devices, disrupt their functionality, or launch attacks on other devices or networks [Omolara et al., 2022].

Regulation. Furthermore, the EU General Data Protection Regulation (GDPR) [The European Parliament and the Council of the European Union, 2018] introduces several legal obligations, among which *privacy-by-default*, *consent management* and *accountability*. Indeed, companies - outside of the legitimate interest - have to explicitly ask the user, as a data owner, for clear, positive and explicit consent before any data collection. Whatever the interests at stake, a company has to prove at any time that data process-

ing is always performed legitimately, either according to the user's consent or for a legitimate purpose.

Requirements for a secure Internet of Things. As a consequence of the security and privacy risks peculiar to the Internet of Things, the requirements for a secure and privacy-preserving Internet of Things are the following. First, the solution must consider *constrained devices*, and ensure the security and privacy threat mitigations are not disconnected from IoT's actual capacities. For example, cryptography-based solutions are often inapplicable to the Internet of Things devices. Second, as data are privacy-sensitive, it is compulsory for user's privacy to enforce *access control* on their data and to *monitor the usage of their data*. Third, for performance, security and privacy purposes, decentralization is an important aspect in the Internet of Things. Centralized entities such as cloud servers may snoop on users' data [Qin et al., 2020] and can be vulnerable to accidental disclosures or external attacks [Qin et al., 2020]. Availability can be a matter of concern too, as physical infrastructure can be damaged, e.g. because of a fire or a natural disaster [Ayoub et al., 2021]. Furthermore, centralization hinders performance, specifically by increasing the deployment and maintenance cost [Salimitari et al., 2020], which in turn hinders scalability.

Leveraging distributed ledgers for the Internet of Things. Distributed ledger technologies (DLTs), due to inherent properties, are a promising solution to address the Internet of Things requirements for security. DLTs indeed provide a certain degree of *decentralization* and are *tamper-proof* which is useful for a wide range of security applications ranging from trust management [Liu et al., 2023] to anonymous and secure transactions [Bothra et al., 2023]. DLTs can also be used to provide access control in an automated and transparent fashion using *smart contracts* [Bao et al., 2023]. However, distributed ledger technologies are not always designed for the Internet of Things requirements. Performance, security and privacy requirements call for well-tailored distributed ledgers, that provide anonymous, efficient and cheap transactions for IoT-constrained devices.

1.2 Problem statement

In this part, several challenges related to the Internet of Things are highlighted and are used to identify *research objectives*. These objectives will provide the connecting thread throughout this document.

Micro-transactions. In the Internet of Things, data collection often serves business purposes and requires financial transactions afterward to buy and sell the data. Data trading is based on *micro-transactions*, which should be processed with the minimum possible fee as regards the data value. Additionally, several Internet of Things real-

life scenarios without data trading require micro-transactions. Such use cases include *parking meters* in smart cities, where the drivers would be charged automatically for the amount of time they park their vehicles. In *Industrial Internet of Things* (IIoT), IoT devices could monitor machinery and equipment and automatically order replacement parts for improved maintenance. The manufacturer could be charged a small fee for the replacement parts, with the payment occurring automatically via micro-transactions.

Overall, micro-transactions can provide new revenue streams for businesses by enabling them to monetize data generated by IoT devices and increase the efficiency of business operations by enabling automated payments. However, it is important to ensure that privacy and security are maintained to protect sensitive data due to the Internet of Things context and financial transactions involved. The latter issues as well as the need for efficient, micro-transactions in the IoT leads to the first research objective of this thesis:

Objective 1: Achieve privacy-preserving zero-fee transactions for the Internet of Things

Distributed ledgers and usage control. To process transactions, blockchains or distributed ledgers in general are technologies of interest in IoT settings due to their inherent distribution. Besides, due to its unique properties, the use of blockchains jointly with usage control has been widely discussed in the literature. Usage control can be considered as an extension of access control, which continuously monitors data once access has been granted. It grants or denies access based on *authorizations*, *obligations*, which have to be fulfilled to be granted access, and finally on *conditions* related to the system state. As a consequence, usage control is beneficial to users' privacy as they can decide who may access their data and how the data are used.

Several works combining usage control and blockchains have been conducted in private blockchain settings [Khan et al., 2020, Shi et al., 2021, Ma et al., 2021, Zhang et al., 2022]. While private blockchains benefit from low latency, they lack some beneficial features of public distributed ledgers such as total distribution with highly secure and immutable data storage. Besides, the network overhead increases rapidly with the number of nodes in the PBFT consensus protocol, commonly used by private blockchains. [Salimitari et al., 2020]. Distributed ledgers often have performance constraints that do not meet IoT requirements, such as the number of transactions per second [Salimitari et al., 2020] which is around 6 transactions per second (TPS) for the Bitcoin blockchain excluding use cases with numerous simultaneous transactions. The size of the transaction ledger can be a hurdle to IoT adoption as well. The Bitcoin blockchain has grown to 475 GB as of June 8, 2023, which is impossible to store on lightweight devices [Blockchain.info, 2023]

Even though the literature proposed integration schemes for usage control in distributed ledgers [Khan et al., 2020, Shi et al., 2021], the general principles of integration are not properly identified. In particular, solving the potential bottleneck issues, as well as the identification of criteria for integration, are not well addressed by the current research. Considering these current limitations, this thesis identifies the following research objectives:

Objective 2: Identify the distributed ledgers appropriate for the Internet of Things

Objective 3: Design a method to integrate usage control in distributed ledger technologies efficiently

Data usage control modeling in distributed systems. A formal model of data usage control can help ensure that the system's security and privacy goals are achieved by providing a clear specification of the access control policies, the resources to be protected, and the roles and responsibilities of the participants in the system. While this modeling has been repeatedly discussed in the state-of-the-art in centralized settings [Pretschner et al., 2011, Kelbert and Pretschner, 2013, Kelbert and Pretschner, 2014, Fromm, 2020], it is still incomplete in distributed systems [Gil et al., 2022], considering only the distribution of the usage control system (UCS) components, but not of the other system users, such as the policy makers and the data readers [Kelbert and Pretschner, 2015, Kelbert and Pretschner, 2018]. In the Internet of Things, due to performance, security and privacy concerns, distributed systems are widely used, requiring an appropriate model to describe information flows and data policies. This leads to the introduction of the fourth research objective:

Objective 4: Identify concepts relevant to the Internet of Things and missing from the current usage control formalism in distributed systems

Validation of the proposed solutions. As part of the previous research objectives (Objectives 1 to 4), several solutions will be proposed in this thesis work to address specific issues in the Internet of Things. As regards the performance, security and privacy constraints of the IoT, it is necessary that: 1) the security and the privacy of these solutions are assessed; 2) the solutions are tested using *proof of concepts* to ensure their feasibility, in particular, as regards performance.These requirements lead to the last set of research objectives:

Objective 5: Analyze the security and privacy of the proposed methods with security and/or privacy threat assessment

Objective 6: Validate the feasibility of the proposed methods with a proof of concept

1.3 Assumptions

The Internet of Things encompasses a wide range of use cases with different requirements and specific research questions. As a consequence, this section explains the context considered in this thesis. Several definitions are first provided to clarify some notions such as the Internet of Things. The precise context considered is then given to define the scope of the conducted research.

Definitions. The *Internet of Things*, i.e., IoT for short, has been first used in 1999 by MIT's Kevin Ashton when he promoted the RFID technology [Zhang et al., 2020]. The Internet of Things has since disrupted the classic Internet by introducing devices embedded in everyday objects, sending and receiving data by themselves. However, the actual definition of the Internet of Things is loose and is often referred to using other words such as *smart home* outside academia and industry [Berte, 2018]. The National Institute of Standards and Technology (NIST) agrees with the lack of a universal definition of the Internet of Things but proposes the two following *fundamental concepts* of the IoT [National Institute of Standards and Technology, 2018]:

(1) *IoT components are connected by a network providing the potential for a many-to-many relationship between components (the network capability may or may not be Transmission Control Protocol/Internet Protocol (TCP/IP) based)*;

(2) *Some IoT components have sensors and actuators that allow the components to observe (collect data about) and affect the physical world.*

These two concepts are of interest as they reflect the *high number of devices*, the *heterogeneity* both in terms of protocols and devices, and the link between the IoT devices and the physical world.

Similarly, *privacy* is repeatedly mentioned in this thesis but is not an easy concept to define. It is recurrently discussed in the literature [Moore, 2008, Alibeigi et al., 2019, Tang et al., 2021, Elmimouni et al., 2023] and derived to more precise concepts,

such as *usable privacy* [Malkin, 2023] which refers to the implementation of user-friendly privacy-preserving mechanisms. For privacy, the two following definitions are proposed, first to emphasize the interdisciplinary aspect of privacy, then a definition more related to data and computer science, also called *information privacy*:

(1) *Broadly speaking, privacy is "the right to be let alone" [Warren and Brandeis, 1890], or freedom from interference or intrusion*;

(2) *Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others* [Westin and Solove, 1968]

To further clarify the notion of privacy, we list the privacy properties of interest for this thesis work. Users tend to provide excessive information to service providers and to lose control of their personal information. Therefore, the *content awareness (1)* property is proposed [Wuyts et al., 2018] to make sure that users are aware of their personal data and that only the minimum necessary information should be shared and used for data processing. The more personally identifiable information a data subject discloses, the higher the risk is for a privacy violation. To ensure content awareness, several technical enforcement tools have been developed such as the personal information feedback tools [Patil and Kobsa, 2009] to help users gain privacy awareness and self-determine which personal data to disclose.

Unlike the user-centric content awareness property, the *policy and consent compliance* property (2) [Wuyts et al., 2018] requires the whole system, e.g., including data flows, data stores, and processes, as a data controller to inform the data subjects about the system's privacy policy, or allow the data subjects to specify consent in compliance with the legislation. These two properties are often referred to as *soft privacy* [Wuyts et al., 2018]. *Hard privacy* focuses on providing as little data as possible to reduce the need to trust other entities, which is referred to as *data minimization*. On the contrary, soft privacy is based on the assumption that data subjects have already no control over their personal data and must have confidence in data controllers. The data protection purpose of soft privacy is therefore to provide data security and process data with specific purpose and consent, using policies, access control, and audit.

Context. In this thesis, only *large-scale IoT deployments* are considered, as use cases involving a moderate amount of devices are already well addressed by the state of the art. Large-scale IoT deployments have specific issues regarding scalability, heterogeneity and performance constraints on devices that require dedicated, well-tailored solutions. For instance, private blockchains, often considered as an IoT-friendly technology

[Asheralieva and Niyato, 2021], are actually not well-suited for large-scale deployments of devices. Second, the scenarios considered will require *payments*, as IoT devices can be used in real-life to generate and sell data or buy actual goods, e.g., vending machines. Finally, the Internet of Things features a high diversity of devices. This *heterogeneity* of devices is considered in this thesis, which implies that constrained devices, e.g., in terms of computing power, memory, energy storage..., are involved in the network. This last assumption markedly restricts the use of several security and privacy tools, such as encryption.

1.4 Contributions

This thesis addresses the defined research objectives with the following contributions:

- the design of a framework to address the requirements of privacy, security and performance of the Internet of Things (Chapter 3). The basis of the framework is the IOTA technology, a distributed ledger relying on a directed acyclic graph to create zero-fee transactions, instead of a blockchain (*Objective 1* and *Objective 2*). A performance evaluation on a proof of concept, as well as security and privacy threat assessments using an illustrative scenario, are conducted on the proposed solution (*Objective 5* and *Objective 6*) ;
- an integration method of usage control with distributed ledgers (Chapter 4). Suitable DLTs are determined according to specified parameters, and a proof of concept is proposed for performance evaluation (*Objective 3* and *Objective 6*). A privacy threat assessment using LINDDUN is also conducted (*Objective 5*);
- an extension of a decentralized model for usage and information flow control in distributed systems, introducing a policy jointly defined over collective personal data and considering the availability of the different individual parts of the distributed system (*Objective 4*);

Framework for performance, privacy and security in the Internet of Things. This first contribution (*Contribution 1*) proposes a newly designed framework to address the requirements of privacy, security and performance of the Internet of Things. The basis of the framework is the IOTA technology. IOTA unlocks distributed ledger performance by increasing throughput as more users join the network thus making the network scalable. Additionally, IOTA does not rely on miners to add transactions to the ledger, enabling zero-fee transactions. Not designed for privacy protection, IOTA is complemented in the framework by privacy-preserving mechanisms: merge avoidance and decentralized mixing. Finally, usage control mechanisms are introduced so that users can monitor the use and dissemination of their data.

Integration of usage control with distributed ledgers. This contribution (*Contribution 2*) proposes to integrate usage control with distributed ledgers based on directed acyclic graphs (DAG). The benefits of integration are: 1) the components of the usage control system contribute to the network security, as a node; 2) the usage control system processes the transaction data without intermediaries, which is faster and more reliable. An analysis of distributed ledgers is provided based on their features, to determine which ledgers are suitable for the proposed integration. An implementation of usage control integration is proposed on IOTA. Performance tests are conducted on this implementation, showing approximately a 90% decrease in the time needed to push transactions and make an access decision when the UCS is integrated.

Data usage control modeling in distributed systems. The last contribution (*Contribution 3*) of this thesis work is the proposition of an extended usage control model going beyond the state-of-the-art that integrates decentralized information flow control (DIFC). DIFC enables users to decide collectively which policy to apply to their common data, further distributing the network by decentralizing the policy definition to the users. Architectural aspects and formal definitions to enable decentralized policies for shared data are proposed as a novelty, resulting from the DIFC integration. Unreliable and intermittent distributed systems are also considered, as a distinctive characteristic in some Internet of Things use cases, e.g., low-energy devices.

1.5 Acknowledgments

This thesis work and the associated Ph.D. has been supported and funded by the Futures & Ruptures program of Foundation Mines-Télécom. It is also supported by both the Institut Mines-Télécom VP-IP Chair on Values and Policies of Personal Information (`https://cvpip.wp.imt.fr`) and Energy4Climate (E4C) interdisciplinary center (`https://www.e4c.ip-paris.fr/`).

1.6 Thesis outline

This document is structured as follows. Chapter 2 provides the necessary background on usage control and distributed ledgers, then the state-of-the-art and its current limitations in addressing the different objectives of the problem statement (cf. Section 1.2). Chapter 3 details the design of a framework (*Contribution 1*) to simultaneously address performance, security and privacy requirements in the Internet of Things and enable micro-transactions (*Objective 1*). The framework is based on the IOTA distributed ledger, and its security and privacy are evaluated on a car-sharing use case. Chapter 4 details the integration of usage control with distributed ledgers (*Contribution 2*) to further improve the performances of both the distributed ledger and usage control system.

Chapter 5 addresses the issues in defining a formalism for usage control in distributed systems and justifies the use of decentralized information flow control to complete the state-of-the-art formalism (*Contribution 3*). The final Chapter 6 concludes this thesis with a summary of the thesis, the future works and the limitations considering both the research objectives and the scope of this research work. A glossary (Appendix A) provides definitions and explanations for some notions for better understanding. The complementary appendices (Appendix C, Appendix D) provide additional general information on the different technologies addressed in this thesis.

Chapter 2

Background on Usage Control and Distributed Ledgers

Contents

In the previous chapter, usage control and distributed ledgers are identified as key technologies for Internet of Things privacy, provided that the distributed ledger can handle requirements in terms of performance. In this chapter, we will thoroughly introduce *usage control* (Section 2.1), a privacy-enhancing technology enabling fine-grained dynamic control over the data.

Modern usage control systems now integrate *Information Flow Control* (IFC), a technology that enables the monitoring of the information flow in a system to prevent undue dissemination. The control of information flow, which is compulsory for efficient usage control as it prevents data from escaping the monitoring scope of the usage control system, will be introduced in Section 2.1.3.

Afterward, a general introduction of distributed ledgers will be provided (Section 2.2), before focusing on specific IoT aspects. Then, distributed ledger privacy (Section 2.3) and performance (Section 2.4) will be introduced, as these two topics are closely related to the Internet of Things concerns.

2.1 Usage control

In the first chapter, we mentioned usage control as a potential privacy-enhancing technology for the Internet of Things, which could be used jointly with distributed ledgers with the proper integration scheme (Objective 3). In this section, the needed notions of usage control are introduced for a better understanding of this document (Section 2.1.1), before introducing the system architecture of usage control systems (Section 2.1.2).

2.1.1 Usage control elementaries

ABC model. Usage control is an extension of access control, describing how the data can be used after initial access. It was first proposed by Sandhu and Park as the *UCON* model [Park and Sandhu, 2004]. This model extends traditional access control by introducing attribute mutability, as well as new decision factors described by the ABC model (Figure 2.1): *Authorizations, oBligations, Conditions*. In addition, the ABC model considers *subjects and objects*, classic components of traditional access control, and their attributes i.e., *subject attributes* and *objects attributes*. *Subject* and *object* are notions taken from access control, such that a *right* enables a subject to access an object in a particular mode e.g., read, write. Note that in usage control, the right does not exist as such, but is determined when the access is initiated by the subject and considering other ABC components. *Usage decision functions* are designed to make the determination of a right's existence based on subject and object attributes, authorizations, obligations and conditions. *Subject* and *object attributes* are properties that can be used during the access decision process. Subject identity is an example of useful subject attributes, but is not

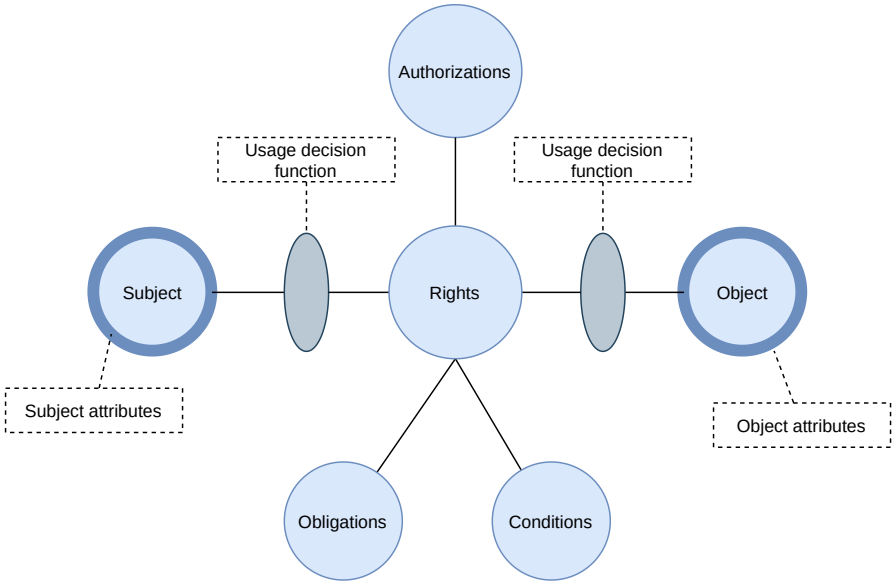


Figure 2.1 – ABC model as defined by Sandhu and Park [Park and Sandhu, 2004]

mandatory as it would exclude the possibility of anonymous services. Examples of object attributes are security labels and access-control lists.

One of the most important innovations of UCON is the introduction of *mutable attributes* that are modified as a consequence of access, while *immutable attributes* are changed as a result of administrative action. This is a critical differentiator compared to most access control models and is particularly relevant in IoT contexts due to the dynamicity of the network.

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 can be checked and obligations fulfilled 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 achieving privacy in IoT systems [Cha et al., 2019], and usage control provides the technical basis to do so.

Related technologies. Modern usage control systems rely on information flow control, particularly to prevent data dissemination to uncontrolled areas. Information flow control is extensively discussed in Section 2.1.3. Besides, to monitor the use of the data, the usage control system needs to access system calls. This process is intrusive and is a threat to data readers' privacy which must be protected as much as the monitored data. To achieve privacy-preserving enforcement of the usage control rules, usage control usually relies on a *Trusted Execution Environment* (TEE) [Shi et al., 2021]. A trusted

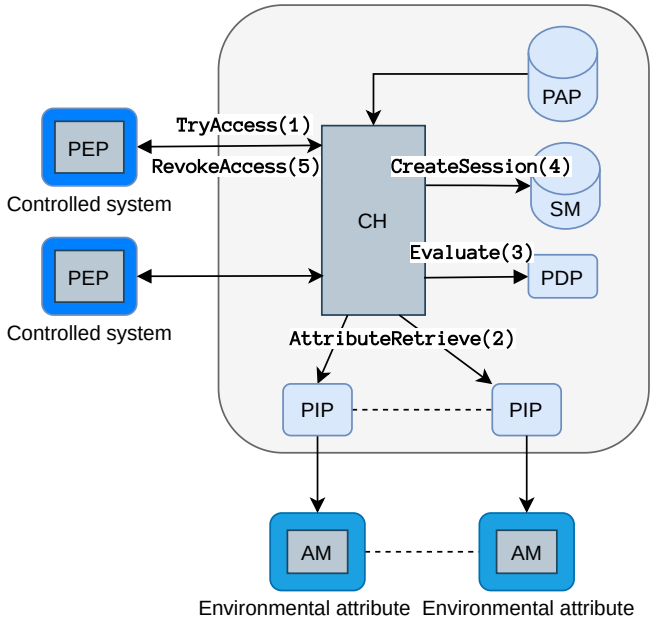


Figure 2.2 – Usage control architecture, based on [Rizos et al., 2019]

execution environment is an area on the main processor of a device that is separated from the system's main operating system. It ensures that data is stored, processed and protected in a secure environment. In particular, the data loaded inside a TEE is guaranteed to be protected as regards *integrity* and *confidentiality*. The TEE is installed on the monitored user's machine, to prevent undue processing or dissemination of the monitored data.

2.1.2 Usage control architecture

The components of the usage control system are depicted in Figure 2.2. Note that different implementations of a usage control system may not implement every component, or conversely design additional ones. For instance, Martinelli *et al.* [Martinelli et al., 2019] introduces components dedicated to the evaluation of obligations called *Policy Obligation Point* to manage distinct kinds of obligations.

The *Usage Control System* (UCS) interacts with *Environmental Attributes* through *Attribute Managers* (AM) to recover the values of the attributes, and with the *Controlled Systems*.

- *PDP*: the Policy Decision Point in charge of the policy evaluation. It takes as input an access request, the corresponding policy and the attributes of both users and context. Then it returns the result of the evaluation: *Permit*, *Deny*, *Undetermined*;

- *PAP*: the Policy Administration Point, which is responsible for creating, modifying, and deleting policies based on the needs of the system and the requirements of the users. It is also the storage point of the policies in the usage control system;
- *PEP*: the Policy Enforcement Points are embedded in the Controlled Systems, and intercept access requests and trigger the UCS decision-making process. The PEPs then enforce the policy evaluation result on the Controlled System which requires an access;
- *PIP*: the Policy Information Point, an interface so that the UCS can retrieve the values of the attributes from the system environment;
- *CH*: the Context Handler, in charge of routing the different processes;
- *SM*: the Session Manager stores all active sessions and the information needed for monitoring their status. The SM is composed of an *Access Table* that stores information about the current ongoing sessions to be able to continuously enforce policies. Usually, Access Tables are implemented by a database whose entries refer to a session and contain at least the session identifier, the access request and the session status i.e., pending, active, revoked or ended.

Workflow. The workflow between the different components of the usage control system after receiving an access request is shown in Figure 2.2 and is composed of the following messages [Rizos et al., 2019]. When a user requires access, the PEP sends a `TryAccess` message to the CH. The CH fetches the environmental attributes from the PIP (`AttributeRetrieve`) and the policy from the PAP, and forwards them to the PDP to evaluate the policy (`Evaluation`). The PDP decides to accept or reject the request with a *PERMIT* or a *DENY* message, then sends the decision to the CH which forwards it to the PEP. If the answer is *PERMIT*, a unique SessionId is assigned to the access request (`createSession`) and the Session Manager (SM) is updated. The PEP then sends a `startAccess` message to the CH, to signal the start of the access. If the UCS detects a policy violation during the continuous monitoring, the CH informs both the PEP and the SM that the session is terminated by sending the `revokeAccess` message. It is also possible that the user asks to end the session itself, by sending the `endAccess` message. The CH tells the SM to delete the session details, and the PIP to unsubscribe the attributes related to this session.

2.1.3 Information flow control

We introduced usage control as a dynamic and flexible access control technology. It enables data owners to enforce policies for their data, by defining authorizations, but

also obligations, which are actions to be performed before, during or after being granted access, and conditions bearing on the system and environment attributes, e.g., the time.

Several methods exist to limit information disclosure, including usage control but also access control lists, firewalls or cryptography. However, although these methods do impose limits on the information that is released by a system, they do not provide any guarantees about information *propagation*. Monitoring data dissemination is valuable in information security to prevent leaks, but also for usage control as information could be disseminated outside the scope of the usage control system. *Information Flow Control* (IFC) is a mechanism introduced to enforce information flow policies in a system. Several methods to implement information flow control have been proposed in the literature:

- Run-time mechanisms that tag data with *information flow labels* have been employed at the operating system level and the programming language level [Hu et al., 2022];
- *Static program analyses* have also been developed to ensure information flows within programs conform with policies [Zheng and Myers, 2007];

Taxonomy. Information flow control techniques can be characterized and compared using the following features [Denning, 1976, Hu et al., 2022].

1. *Operator precision*: How are security classes updated?
    - Conservative: IFC uses a least upper bound;
    - Precise: IFC considers the value of the data;
    - Hybrid: IFC performs a tradeoff between accuracy and computational cost.
2. *Security properties*: Which security properties are guaranteed?
    - Confidentiality: IFC prevents leakage of sensitive information;
    - Integrity: IFC prevents undue modification of data;
    - Isolation: IFC forbids communication between two agents with different trust levels;
    - Constant time: IFC captures information flows through runtime variations;
    - Design integrity: IFC technique detects malicious information flows triggered by design modifications;
3. *Level of Abstraction*: What is the level of abstraction of the information flow model?
    - System: IFC considers flows at the system level;

- Algorithmic: IFC is deployed during high-level synthesis (HLS). HLS involves transforming an abstract, algorithmic-level description into a lower-level, hardware-specific implementation that can be synthesized into digital circuits;
- Architecture: IFC is deployed at the Instruction Set Architecture (ISA) level. The Instruction Set Architecture is the interface between a computer's software and hardware components, defining the set of instructions that a processor can execute and the memory model it operates on;
- RTL: IFC targets register transfer level (RTL);
- Gate: IFC occurs at the gate level. A gate-level netlist is a representation of a digital circuit at the lowest level of abstraction;
- Circuit: IFC targets analog and mixed-signal hardware designs.

4. *Verification Technique*: Which verification techniques are supported?

   - Simulation: IFC determines information flows using simulation tools;
   - Formal: IFC checks security properties using formal methods, e.g., theorem proving, equivalence checking...;
   - Emulation: IFC relies on hardware emulation of information flow behaviors;
   - Virtual prototyping: IFC creates a software version of the hardware to monitor information flow;
   - Runtime: Dynamic IFC during runtime.

2.1.4 Conclusion on usage control

As a conclusion, we summarize the key elements of usage control needed for the rest of this document. First, usage control is an extension of access control introducing attribute mutability, obligations and conditions. These concepts are useful to handle IoT aspects, in particular the dynamicity of the network, and to design fine-grained policies that consider the users' actions and the system's state, during the whole access phase (including before and after the access). The usage control system (UCS) is responsible for enforcing usage control, and its components are partly located on the end devices under the form of policy enforcement points. Finally, modern usage control systems are complemented by information flow control, to monitor the data dissemination and prevent malicious users from sending the data outside of the UCS scope.

2.2 Distributed ledgers

Distributed ledgers, particularly blockchains, have been actively studied as a promising technology for the Internet of Things privacy [Rifi et al., 2017, Ma et al., 2021, Goyat et al., 2022, Rajasekaran et al., 2023, Bao et al., 2023] notably for access control, auditing and data storage [Cha et al., 2019]. As a potentially relevant solution to *Objective 1*, the focus will be given in this section to distributed ledger technologies and why they can be useful for the Internet of Things.

General principles of distributed ledger technologies (Section 2.2.1) and common consensus methods (Section 2.2.2) will first be introduced in this section, before distinguishing distributed ledgers from blockchains (Section 2.2.3).

Privacy issues in distributed ledgers and current means to mitigate them are discussed in Section 2.3. Distributed ledger performance is finally considered in Section 2.4 along with the relevant metrics to measure it.

2.2.1 Blockchain elementaries

History. In 1996 the NSA published a report "How to make a mint: the cryptography of anonymous electronic cash" [Law et al., 1997], to express its concerns about electronic currencies. This document raises several potential problems, which are still relevant today:

1. The problem of *double spending*. In the case of offline payments, it is not possible to guarantee that a coin will only be spent once;

2. To transfer money, the user must use an electronic *wallet*, developed by a company. This can be seen as a trusted third party that can potentially act against the will of its owner, e.g., by tracing the transactions. The latter can also be in charge of checking that there is no double spending;

3. All the security of the protocol lies in the chosen *cryptographic functions*, which may be insecure.

In 1997, Adam Back proposed a *proof-of-work* method, under the name Hashcash [Back, 2002]. The principle is to perform a given amount of work, requiring CPU resources, to access a service. If a user is legitimate, this work is negligible, while in the case of a malicious person accessing the service repeatedly, this amount of work becomes considerable. This technique is used for example to fight against *spam* or *denial of service*.

In 2008, Satoshi Nakamoto wrote the founding paper for Bitcoin [Nakamoto, 2009]. This name appears to be a pseudonym, and the real identity of this person, or group of people, remains a mystery to this day. This article describes the structure and operation

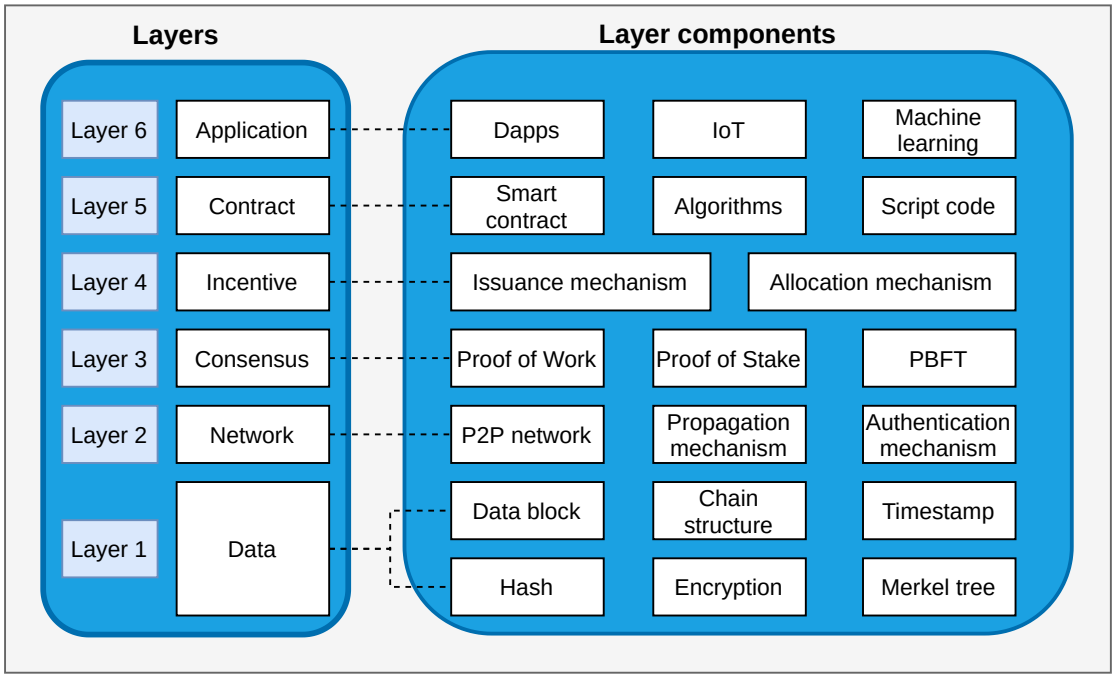


Figure 2.3 – Blockchain architecture in six layers, based on [Yuan and Wang, 2018]

of the Bitcoin protocol. The implementation of Bitcoin came the following year, in 2009, under the name Bitcoin-QT. This is also the first time the word *blockchain* is used.

Five years later, in 2013, Vitalik Buterin introduces Ethereum [Buterin, 2014], a novel blockchain network with the possibility to execute *decentralized applications* (DApps) *via* smart contracts. *Smart contracts* are "computerized transaction protocol that executes the terms of a contract" [Szabo, 1994], enabling decentralized execution of code with predetermined conditions, without the intervention of a third party.

Blockchain architecture. Blockchains are usually decoupled in several layers, depending on the model, to fully describe the underlying architecture of a blockchain. While the models can differ, the most common model is composed of six layers [Wen et al., 2021, Issa et al., 2023], discussed next and shown in Figure 2.3.

The first *data layer* involves the techniques required for storing transaction records. Each block in the ledger stores a set of verified, timestamped and hashed transactions, in the form of a Merkle tree [Yuan and Wang, 2018, Issa et al., 2023].

The *network layer* (layer 2) includes the network aspects, such as the communication and verification mechanisms, including authentication. Nodes in a blockchain network usually interact forming a peer-to-peer (P2P) decentralized network. The network layer helps in broadcasting, forwarding and verifying transactions. When a node creates a transaction, it signs it with its private key then broadcasts it to its neighbors. The

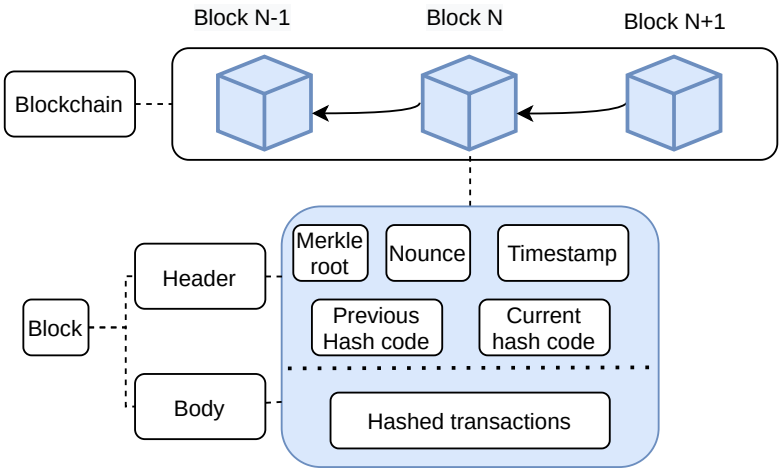


Figure 2.4 – Block composition - based on [Issa et al., 2023]

neighbors verify this transaction with its public key. If the transaction is valid, it is added to the blockchain and broadcast to the other nodes of the network, or discarded otherwise [Yuan and Wang, 2018, Issa et al., 2023].

The *consensus layer* (layer 3) is used by the network of nodes to reach an agreement on the state of the blockchain. The blockchain state includes the valid transactions, the blocks in the ledger and their order, as well as the next block to be added. Numerous consensus algorithms have been designed for blockchains, which are extensively discussed in Section 2.2.2.

The *incentive layer* (layer 4) is a fundamental part of the blockchain architecture that encourages users to contribute to the network, usually with economic incentives. The incentives are given to specific nodes so that they verify the blocks and keep the decentralized nature of the blockchain network.

The *contract layer* (layer 5) includes smart contracts, but also algorithms and script codes to extend the logic of transactions. This layer enables to devise more complex business rules, automatically performed when predetermined conditions are fulfilled.

The *application layer* (layer 6) includes a wide range of applications that leverage blockchain properties. It can include generic applications, such as machine learning or the Internet of Things. Decentralized applications (DApps) are encompassed in this layer as well.

Block composition. The blocks composing the transaction ledger are composed of a header and a body, shown in Figure 2.4. The body part of the block holds hashed transactions, while the header is constituted of these parts [Nakamoto, 2009, Issa et al., 2023]:

1. *Merkle Root*: The Merkle root is a hash value computed from the set of all transactions included in the block. It is used to cryptographically prove that all transactions within the block are valid and unchanged [Merkle, 1987].

2. Nonce: a nonce is an arbitrary number that can be used just once in a cryptographic communication. Nonces are used in proof-of-work systems to vary the input to a cryptographic hash function to obtain a hash for a certain input that fulfills certain arbitrary conditions, notably on difficulty;

3. Previous Hash Code: This field contains the hash value of the previous block in the blockchain. It forms the link between blocks and ensures the integrity and chronological order of the blockchain.

4. Timestamp: The timestamp indicates the exact time when the block was created. It provides a reference point for ordering blocks and contributes to the overall consensus mechanism of the blockchain.

2.2.2 Consensus methods

Consensus methods (layer 3, as introduced in Section 2.2) are very diverse and have different advantages and drawbacks regarding performance, security and privacy. Due to their significant impact on the network characteristics, the most popular consensus mechanisms are discussed next.

Proof of Work. The proof of work (PoW) is a consensus method based on a computation race. The first participant who solves the computation puzzle wins the right to add the next block to the transaction ledger. It was popularized by the Bitcoin blockchain as the first consensus method for blockchains [Nakamoto, 2009]. The user appending the new block to the ledger is called the *miner*. The computation puzzle is most commonly based on a hash function, where the users have to find a nonce to solve the given problem. It consumes a lot of computational resources, raising environmental concerns [Stoll et al., 2019, Sharma et al., 2023].

Blockchain networks based on proof of work are vulnerable to the 51% attack, which is a potential security vulnerability in blockchain networks where a single entity or group of entities gains control of more than 50% of the network's mining power [Aponte-Novoa et al., 2021]. If an attacker gathers the majority of the mining power, it allows him to spend its funds multiple times, reorganize the blockchain by creating a longer chain with alternative transactions, or block the incoming transactions.

Proof of Stake. The proof of stake (PoS) was proposed to overcome the issues of the proof of work, in particular the high energy and computational resource consumption. In contrast with the proof of work, the PoS does not rely on a computation race but rather on an economic stake as proof to select the next validator. The first cryptocurrency to use proof of stake is Peercoin (PPCoin)[King and Nadal, 2012]. Ethereum, Bitcoin's main competitor, has replaced proof of work with proof of stake in its current version Ethereum 2.0. While proof of stake is not threatened by the 51% attack[1], it is endangered by the *Nothing at Stake* problem.

The *Nothing at Stake* problem arises when validators in a PoS network are not financially incentivized to follow the rules and behave honestly. Since validators do not have to spend computational resources or energy to create blocks in PoS, there is no direct cost associated with attempting to create multiple competing blockchains. This creates a scenario where validators can potentially create multiple valid chains in an attempt to double-spend or disrupt the network without any economic consequences. However, the *Nothing at Stake* problem is not a practical vulnerability in most PoS protocols. This is because PoS protocols usually have mechanisms implemented to prevent or mitigate this problem. For example, penalties or slashing mechanisms can be used to punish validators who attempt to create multiple competing chains.

There are several variations of the proof of stake that are designed to adjust PoS performance metrics to fit specific use cases. The **Delegated Proof of Stake** (DPoS) is a version of PoS where stakeholders vote to elect a set of delegates or validators who are responsible for producing blocks and maintaining the blockchain. These elected validators take turns creating blocks on behalf of the stakeholders who elected them. The **Leased Proof of Stake** (LPoS) allows users to lease their stake or balance to a chosen delegate, who then includes their stake in their own validator's pool. This allows users with smaller stakes to participate in the block production [Salimitari et al., 2020]. The **Liquid Proof of Stake** (also LPoS) introduces a concept of liquidity, where stakeholders can freely transfer or trade their staked tokens while still participating in the consensus mechanism. This allows for more flexibility and liquidity compared to traditional staking mode [Breitman and Breitman, 2014].

The **Proof of Authority** (PoA) is a variant of the proof of stake where the *identities* and the *reputation* of the nodes *are at stake* rather than a cryptocurrency asset. The time to reach the consensus and the latency is better compared to the proof of work, but not as good as the actual proof of stake [Raghav et al., 2020]. One significant contribution to its popularity was the introduction of the PoA consensus algorithm by Gavin Wood, the co-founder of Ethereum, in the Ethereum network's *Parity* client [Ekparinya et al., 2020]. In the following, only the conventional PoS, the delegated and the proof

[1]Yet, if a single entity or a group were to control the majority of the stake, they would have disproportionate influence over the network's consensus and decision-making process

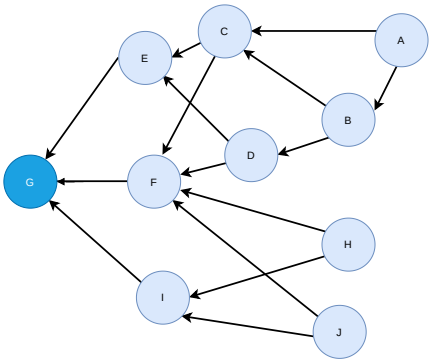
of authority are considered, as both liquid and leased proofs do not have an impact on network metrics, but have an economic rationale.

Proof of Elapsed Time. In the Proof of Elapsed Time (PoET), the miner is chosen at random based on a timer. The user whose timer expires first becomes the miner. This consensus method has several benefits, including a higher throughput and a low latency. However, its main drawback is its reliance on Intel's SGX, as the correctness of the timer execution must be verified within a *trusted execution environment*, implying PoET's governance is centralized.

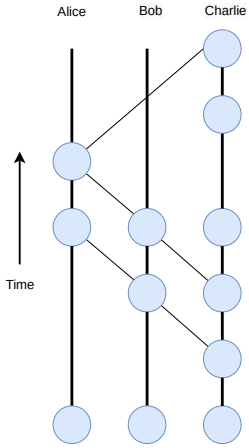
The Proof of Elapsed Time consensus mechanism was popularized through its implementation in the Hyperledger Sawtooth blockchain platform, developed by the Linux Foundation. Intel, the company behind the creation of PoET, actively contributed to its promotion and adoption by highlighting its energy efficiency and scalability benefits compared to traditional consensus mechanisms like Proof of Work (PoW) [Wang et al., 2022a].

Practical Byzantine Fault Tolerance. The Practical Byzantine Proof Tolerance (PBFT) is a voting-based consensus method. All the nodes are involved in the voting process and the consensus is reached when more than two-thirds of the nodes agree upon the next block. As a consequence, the network can handle malicious behavior from at most a third of the nodes, which is low compared to the 51% assumption in the proof of work networks [Aponte-Novoa et al., 2021]. PBFT is consequently efficient in private blockchains, but not for public blockchains which have a lower tolerance to malicious nodes [Salimitari et al., 2020]. In addition, the voting process does not scale well, as PBFT generates a significant network overhead [Salimitari et al., 2020]. Practical Byzantine Fault Tolerance (PBFT) was introduced by Miguel Castro and Barbara Liskov [Castro and Liskov, 1999]. The paper introduced the PBFT algorithm as a solution for achieving consensus in distributed systems under the presence of Byzantine faults.

A delegated version of the practical byzantine fault tolerance is used as well in cryptocurrencies called *dPBFT*. dPBFT stands for *Delegated Proof of Byzantine Fault Tolerance*. It is a consensus algorithm that builds upon the original PBFT algorithm. dPBFT was introduced to improve the scalability and performance of PBFT by allowing a smaller subset of nodes, known as *delegates* or *representatives*, to handle the consensus process on behalf of the entire network. This delegation reduces the communication and computational overhead compared to traditional PBFT, making it more suitable for large-scale distributed systems. dPBFT and the concept behind it have been popularized by the NEO cryptocurrency project, which implemented the dPBFT consensus algorithm as its underlying consensus mechanism [Zhan et al., 2021].



(a) Directed acyclic graph (DAG). Transaction *A* directly approves *B* and *C*, indirectly approves *E*, *F* and *G*. *G* is the original transaction, approved by all other transactions [Popov, 2017].



(b) Hashgraph, representing the gossip history. If Alice receives gossip from Bob, the event is marked by a vertice in the Alice column, with two edges going downward to the preceding gossip events by Alice and Bob [Baird et al., 2018].

Figure 2.5 – Alternatives to blockchains to build transaction graphs.

2.2.3 From blockchains to distributed ledgers

While blockchains are the most well-known instances of distributed ledgers for cryptocurrencies, the notion of *Distributed Ledger Technology* (DLT) is broader and includes several other technologies of interest that are introduced in this section.

First, a distributed ledger can be completely disconnected from the notion of cryptocurrency, e.g., distributed databases. Besides, some cryptocurrencies do not build their transaction ledger using blockchains, but rather using different mathematical structures. The most used alternatives to blockchain in cryptocurrencies are the *directed acyclic graphs* (DAGs) and the *hashgraphs*.

Directed acyclic graph. Unlike a traditional blockchain, where transactions are organized in linear blocks, a directed acyclic graph allows multiple transactions to be confirmed concurrently, enabling greater scalability and potentially faster transaction processing times. Several cryptocurrencies rely on directed acyclic graphs to build the transaction ledger such as IOTA [Popov, 2017], Obyte [Churyumov, 2017] and Nano [LeMahieu, 2017]. IOTA is by far the most studied DAG-based distributed ledger in the scientific literature [Conti et al., 2022, Carelli et al., 2022, Guo et al., 2023, Naresh et al., 2023]. Though DAG-based technologies slightly differ, they share several interesting properties:

- small transactions fees, or even no transaction fees at all;
- writing transaction is not energy-intensive;
- throughput is high after the bootstrapping stage and increases with the number of users;
- users add their transactions directly to the network without relying on any intermediaries, such as miners or gateways.

In general, a DAG-based distributed network builds its transaction ledger as detailed next. The transactions issued by nodes constitute the vertices of the transaction graph. There is only one transaction by vertice, in contrast with blockchains that usually store multiple transactions in one block. The edges of the graph are obtained as follows: when a new transaction arrives, it must approve two previous transactions. These approvals are represented by directed edges, as shown in Figure 2.5a. If there are no directed edges between two transactions A and B, but there is a directed path of length at least two from A to B, we say that A indirectly approves B. There is an original transaction, which is approved directly or indirectly by all the transactions.

Hashgraph. Hashgraph is a distributed ledger technology designed as an alternative to blockchains. The hashgraph technology is currently patented and used by the public ledger Hedera [Baird et al., 2018]. Hashgraphs have been studied in the literature as possible alternatives to blockchains, in particular in the context of the Internet of Things [Bansal and Bhatia, 2020, Gao et al., 2022, Jha et al., 2022, Tarlan et al., 2022]. The claimed features of Hedera's hashgraph are the following:

- very high transaction throughput and low latency;
- *fairness* in the ordering of transactions by using a consensus algorithm that considers the timestamp of each event. This approach aims to prevent unfair advantages or biases among participants, in particular in use cases involving real-time pricing such as stock markets;
- resilience against denial of service attacks thanks to asynchronous byzantine fault tolerance (aBFT) consensus. aBFT is a version of PBFT for asynchronous networks;

While these claims are attractive, Hedera's hashgraph lacks evaluation and peer review for the scientific community. In particular, the claim regarding throughput, up to 500.000 transactions per second [Baird et al., 2018], is way above the current standards (24.000 TPS for Visa). Hedera claims that the network throughput is only limited by bandwidth, which requires each network node to be able to download and upload the transactions, which set high requirements on network nodes. Besides, the claims

should be verified based on empirical evidence, performance benchmarks, and independent research.

Hashgraph consensus protocol. The core of the hashgraph consensus is called *gossiping*. The gossip protocol unfolds as follows. The first member (Alice) chooses another member at random (Bob), and then Alice tells Bob all of the information she has about the current status of the ledger. Alice then repeats with a different random member. Bob repeatedly does the same, as well as all other members of the network. Consequently, if a single member becomes aware of a new transaction, the information is spread exponentially fast through the community until every member is aware of it.

In hashgraph, the participants do not only gossip transactions, but also the hashgraph itself, a process which is referred to as *gossip about gossip*. Gossiping a hashgraph gives the participants a great deal of information. If a new transaction is placed in the payload of an event, it will quickly spread to all members, until every member knows it. A participant can learn about new transactions, know when exactly a participant learned about a given transaction, and also know exactly when a participant learned the fact that another participant had learned of that transaction by transitivity.

The gossip history is therefore represented as a directed graph. The history of any gossip protocol can be represented by a graph where each member is a line composed of vertices. When Alice receives gossip from Bob, telling her everything he knows, that gossip event is represented by a vertex in Alice's line, with two edges going downward to the immediately-preceding gossip events by Alice and Bob. The hashgraph is represented in Figure 2.5b.

2.2.4 Conclusion on distributed ledgers

Distributed ledgers, whose most famous instances are blockchains, are in fact more diverse and encompass distributed databases as well as ledgers built using hashgraphs or directed acyclic graphs. Apart from the structure of the transaction graph, the main distinctive feature of distributed ledger is the consensus method, used to decide who can write the transactions and how. The consensus method has a strong impact on the network's security and performance, which we will discuss in Section 2.4. Besides, distributed ledgers are confronted to privacy challenges that are discussed in the next Section 2.3

2.3 Preserving privacy in distributed ledgers

While distributed ledger transactions are thought to be anonymous, the reality is more balanced. Public blockchains do not require identifying information to make a transaction. Yet, access to transactions and their content is not restricted. The transactions dis-

close information about the different parties involved and create risks of inference. Interested third parties also automatically collect and analyze this information, for several purposes including law enforcement [Harrigan and Fretter, 2016]. By default, public blockchains only provide pseudonymity, or anonymity provided the linkage between the pseudonym and the real identity is not possible. However, several behaviors drastically make re-identification easier, which are discussed in the section 2.3.2. In the next section, we detail the privacy threats for distributed ledger technology, based on LINDDUN privacy threat modeling [Wuyts et al., 2018].

2.3.1 Privacy threat modeling.

LINDDUN is a privacy threat modeling framework [Wuyts et al., 2018] to reason about potential privacy concerns in a systematic and structured way. It is structured according to seven threat types captured in the acronym LINDDUN. LINDDUN is used in this section to identify the main privacy threats in distributed ledgers so that the risks addressed by the privacy-preserving tools detailed in the following (Section 2.3.3) are clearly expressed.

Linking. Linking is the association of data items or user actions to learn more about an individual or a group. In blockchains, linking threats can be done by linking several transactions together to deduce the consumption habits of a user, or by linking senders and receivers of transactions.

Identifying. The identifying threat is the risk of learning the identity of an individual, even though it wants to remain anonymous. In distributed ledger technologies, the main risk is breaking the pseudonym, i.e., the address of a user in the blockchain, to disclose its true identity. This is usually achievable by combining blockchain data with external information, such as network data, by identifying from which IP address a transaction originates.

Non repudiation. The non-repudiation threat is the ability to attribute a claim to an individual, e.g., read a message. The system maintains evidence regarding some actions of facts, which implies deniability claims, like log files or metadata. Non-repudiation is a counter-intuitive notion when it comes to privacy threat assessment, as in *security threat modeling* such as STRIDE [Howard and Lipner, 2006], repudiation is conversely considered as the security risk. In security, it is more relevant to prevent a malicious user from denying it performed a forbidden action, while for privacy preservation, it is better to be unable to attribute actions to individuals. An example of non-repudiation risk for a blockchain user is the impossibility to deny it performed a transaction.

Detecting. Detecting is the ability to deduce the involvement of an individual through observation. Notably, as the ledger is public, it is possible to see if a user made transactions in a blockchain network as its address is recorded in the transaction. It is an issue when combined with identification attacks, or if the motive of the transaction

can be disclosed.

Data disclosure. Data disclosure is the excessive collection, storage, processing or sharing of personal data. This is not a general threat in blockchains and distributed ledger technologies and is rather implementation-dependent.

Unawareness. Unawareness corresponds to insufficient information and involvement of individuals in the processing of their personal data. It is also not a general risk.

Non-compliance. The last non-compliance threat reflects the deviation from security and data management best practices, standards, and legislation. Distributed ledger technologies can be legal conundrums regarding the EU General Data Protection Regulation (GDPR) [The European Parliament and the Council of the European Union, 2018, Haque et al., 2021]:

1. Article 3 - *Territorial scope*. Article 3 is about controlling the user data from being processed and stored outside the geographical area of the EU. In the case of the public blockchain, it is difficult since the nodes are distributed worldwide. Private blockchains can conversely have nodes that are located in the same region;

2. Article 16 and 17 - *Right to deletion and modification*. One of the most important legal issues for blockchains is Article 17 of GDPR, about the *right to be forgotten*. That means the concerned organizations should delete the user data if the users request it. Since information inside the blockchain cannot be removed, it directly contradicts Article 17. This is also true for Article 16 *right to rectification* as data cannot be edited either.

3. Article 4 states the definition of *personal data*, data *controller*, and data *processors*. Articles 24 and 28 describe data controllers' and processors' purposes and responsibilities. Identifying data controllers and processors is hard for blockchains since no centralized authority is controlling the nodes or is accountable for them.

2.3.2 Breaking pseudonymity in blockchains

Most public blockchains including the two most popular, Bitcoin and Ethereum, only provide pseudonymity under the form of an address. Anonymity would be provided if it is not possible to link the pseudonym and the real identity. However, several behaviors of users and techniques significantly ease the re-identification process.

Clustering. Address clustering in the context of blockchain refers to the process of identifying multiple addresses that belong to the same user or entity by analyzing their transactional behavior on the blockchain [Huang et al., 2017]. It is a method used to de-anonymize blockchain transactions and can potentially reveal the identity of the user or entity behind the transactions. Address clustering is a common technique used

by law enforcement [Harrigan and Fretter, 2016] agencies to investigate illicit activities such as money laundering and terrorist financing.

Address reuse. Address reuse in blockchains occurs when the same public address is used for multiple transactions. This can be problematic for several reasons. First, for privacy, when a user reuses an address for multiple transactions, it becomes easier for anyone to track the user's transactions as it requires to link one address to an identity. Conversely, using a different address for each transaction requires repeating the process of re-identification several times, which is time-consuming. Additionally, address reuse is a problem for security, as the leak of only one private key enables a malicious agent to steal all the funds of a user.

Network analysis. By monitoring network traffic and analyzing the timing, size and value of transactions, it is possible to infer relationships between addresses and potentially identify the real-world identities hidden behind the pseudonym address. For instance, heuristics have been developed to identify ownership relationships between Bitcoin addresses and IP addresses [Koshy et al., 2014], significantly facilitating re-identification.

2.3.3 Obfuscation using coin mixing and merge avoidance

Coin mixing services allow users to mix cryptocurrency coins to enable unlinkable payments and prevent tracking of honest users' coins by both the service provider i.e. the *coin mixer* or *coin tumbler*, and the users themselves. The easy bootstrapping of new users and backward compatibility with cryptocurrencies are attractive features of this service, which has recently drawn the attention of both academia and industry. While useful for privacy preservation, coin mixers face several technical challenges such as decentralization to be able to meet their requirements as regards privacy.

Principles of coin mixing. Cryptocurrency mixing services are designed to remove the linkage between senders and receivers of transactions. To achieve this task, the cryptocurrency mixing service gathers coins from different users, whose identities are linked to these coins. The mixing service then keeps the coins for a random time before assigning the coins to the users. The coins 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 the re-identification of the sender by using timestamps. The mixing process is shown in Figure 2.6. Mixing services should have the following guarantees:

- *anonymity*: usually considered as the unlinkability between the senders and the receivers of the transactions [Sarfraz et al., 2019, Seres et al., 2019, Glaeser et al., 2022];

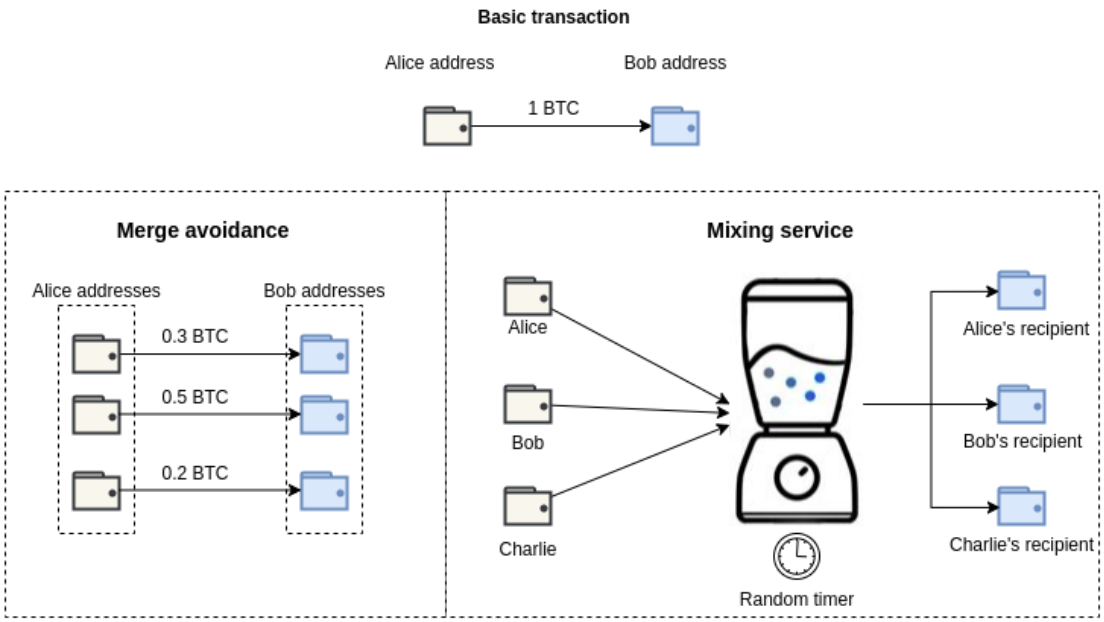


Figure 2.6 – Coin mixing and merge avoidance for transaction obfuscation.

- *availability*: users must be able to withdraw their cryptocurrencies from the mixer [Seres et al., 2019];
- *correctness*: no malicious participant should be able to steal other participants' cryptocurrencies [Sarfraz et al., 2019, Seres et al., 2019], or their private keys including partial private keys [Sarfraz et al., 2019];

Limitations of centralized coin mixing. Centralized coin mixing services, although important privacy-enhancing parts in blockchains have several limitations which create significant risks for users:

- *availability*: being centralized and responsible for obfuscating transactions, coin mixing services are likely targets of denial-of-service attacks as single points of failure (SPOF) [Sarfraz et al., 2019];
- *trust*: the centralized mixing service needs to be trusted, as it still can link the sender and the receiver of the transaction, and also can keep the coins for itself, or take a fee without actually providing the service.

These two limitations are direct risks to the security and privacy of the users. This justifies the introduction of decentralized mixing schemes.

Decentralized coin mixing. Coin mixing services, when centralized, become a single point of failure [Sarfraz et al., 2019, Andola et al., 2021] and are targets of choice

for attackers willing to break the privacy of users or to embezzle the coins. Therefore, the first natural step to improve coin mixing would be to decentralize the mixing service. Decentralized mixing faces a major issue nonetheless, known as the *bootstrapping problem*[Glaeser et al., 2022]. The bootstrapping problem is the difficulty to find a set of initial participants to execute the protocol. While a high number of participants is desirable to improve the anonymity guarantees provided by the coin mixing protocol, it is at the same time undesirable as it results in poor scalability and makes bootstrapping harder [Glaeser et al., 2022].

While decentralized mixing services are more resilient against denial of service attacks, the decentralization is troublesome as well as it exposes the mixing service to *edge insertion attacks*. In such attacks, a malicious user claims that additional mock nodes were involved in the mixing process to receive the corresponding rewards on their behalf [Simões et al., 2021]. While this kind of attack is easy to mitigate with a central entity accountable for node identity verification, it becomes a challenge in a distributed setting. This attack is hard to prevent in distributed ledgers with transaction fees, as the miner, and consequently, the mixing service must be paid to fulfill their role and cover their costs.

Merge avoidance. Merge avoidance is a technique first designed for Bitcoin [Hearn, 2013, Simões et al., 2021] by Mike Hearn, a Bitcoin developer, but it can be generalized to other cryptocurrencies [Sarfraz et al., 2019]. 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, and the addresses must not be linked to the identity of the sender. Otherwise, it will be easy to rebuild the original transaction using a blockchain explorer with either the sender or the receiver address.

The purpose of merge avoidance is to prevent inference attacks, in which an attacker could use the amount of cryptocurrency sent to infer the purpose of the transaction. For instance, let's consider a case where Alice is getting paid her wage in the Ethereum cryptocurrency (ETH). Each month, she gets her wage to the same wallet, at a similar time of the month. An attacker can easily guess the purpose of the transaction by observing the different transactions with Alice's wallet as outputs. By sending each individual wage in different sub-transactions to multiple wallets belonging to Alice, it is much harder to identify the purpose of the payments. However, the merge avoidance strategy has downsides [Hearn, 2013]:

- as a receiver, your level of privacy depends on the sender and how it crafts transactions. However, the senders do not have a strong incentive to protect your privacy and may just ignore the merge avoidance process if it is not compulsory;
- it increases the number of transactions, which increases the size of the ledger, but

the overhead is quite limited [Hearn, 2013];

2.3.4 Privacy-oriented cryptocurrencies

While privacy-enhancing technologies have been devised to preserve the privacy of blockchain users, other cryptocurrencies endeavor to implement privacy by design. Here are a few technologies of interest focusing on privacy protection:

Monero (XMR): Monero [van Saberhagen, 2013] is a privacy-focused cryptocurrency that uses a technology called *ring signatures* to mask the sender's identity. It relies on stealth addresses to conceal the recipient's address and *ring confidential transactions* (RingCT) to obfuscate the transaction amount. These techniques make it difficult to trace Monero transactions and provide a high level of privacy.

Zcash (ZEC): Zcash [Bowe et al., 2016] is a privacy-oriented cryptocurrency that uses a technology called *zero-knowledge proofs* (ZKPs) to keep transactions private. ZKPs allow users to prove that a statement is true without revealing any additional information beyond what is necessary [Goldwasser et al., 1985]. In Zcash, ZKPs are used to hide transaction details such as the sender's address, the recipient's address, and the transaction amount.

Dash (DASH): Dash [Duffield and Diaz, 2014]is a privacy-focused cryptocurrency that offers optional privacy features. Dash seeks to improve upon Bitcoin (BTC) by providing stronger privacy and faster transactions. Its *PrivateSend* feature allows users to mix their transactions with other users to make them more difficult to trace. PrivateSend uses a decentralized network of *masternodes* to mix transactions, ensuring that no single entity can track the transaction history.

2.3.5 Conclusion on privacy

While blockchain transactions are often considered privacy-preserving, distributed ledgers are confronted with challenges when protecting users' anonymity. Distributed ledgers often only provide pseudonymity, which can be broken by some actors notably by using network information. Some privacy-enhancing technologies have been designed to safeguard users' anonymity, such as the coin mixer, whose purpose is to prevent the linkage between the sender and the receiver of a transaction. Additionally, some cryptocurrencies have been designed to specifically provide anonymous transactions (ZCash, Monero...). In the next section, we will discuss the performance aspects of DLT, such as the metrics used to describe the performance of a distributed ledger, as well as the impact of the consensus methods on these metrics.

2.4 Performance considerations of distributed ledger technologies

Due to resource-consuming and time-consuming consensus methods, blockchains and distributed ledger performances have been critically important in the literature. [Brotsis et al., 2021, Fan et al., 2021, Chen et al., 2022, Okegbile et al., 2022]. Performance aspects are the basis used to compare consensus methods and suggest or preclude the use of a consensus method for a use case. In this section, we will first introduce the metrics used to assess the performance of a consensus method (Section 2.4.1), before detailing the expected performance of the most important methods and how they answer the Internet of Things requirements (Section 2.4.2).

2.4.1 Performance metrics

To assess the performance of distributed ledger technologies, a wide range of parameters can be considered. Security aspects must also be considered as increasing performance may impact security negatively. In distributed ledger technologies, this translates into a relatively low adversary tolerance. *Adversary tolerance* is achieved through designing consensus protocols that guarantee security even in the presence of an adversary who may control up to a certain fraction of the resources of the system. This fraction expresses the robustness of the consensus method. We now look at the different metrics used to measure the performance of a consensus method.

Throughput, or in the context of distributed ledger, transaction throughput, is the number of transactions that a blockchain network can process per unit of time. It is usually expressed in *transaction per second* (TPS). It must be distinguished from *network throughput*, which measures the amount of data that can be transmitted from one point to another within a given time. It is usually expressed in bits per second (bps) or bytes per second (Bps) but is a less relevant notion for blockchain networks. Transaction throughput can greatly vary when comparing DLTs. The Bitcoin blockchain has a maximum throughput of around 3-7 transactions per second, while Ethereum, the 2^{nd} most important blockchain in capitalization has a throughput between 15 and 45 TPS in its current version [2]. Compared with traditional payment networks, Visa can process up to 24.000 transactions per second, while Paypal has a 193 TPS throughput. A final thing to consider when discussing throughput is the range. A network processing under 100 transactions per second can be considered as having a low throughput, while 100 to 1000 TPS will be considered average and over 100 TPS a high throughput [Salimitari et al., 2020].

Latency, in blockchain networks, is the amount of time between initiating a trans-

[2] Ethereum 2.0 is expected to drastically increase the throughput

action or payment and receiving confirmation that it is valid [Kokoris-Kogias, 2022]. It is sometimes referred to as *block time*, which covers a slightly different notion as blocks are not transactions, but groups of transactions. For instance, a block of the Bitcoin blockchain can be made of 2000 transactions [Data, 2023]. Latency is usually expressed in seconds. While latency may seem to be the inverse of throughput, these two notions are not related. Indeed, when the system is at a low load, the number of transactions per second can increase until reaching the high load regime for the network, with no impact on the latency. Conversely, on high load, if new transactions are pushed to the network, the throughput stays the maximum, but the latency keeps on increasing. Both notions are therefore necessary to reflect the transaction processing capacity of a given distributed ledger network.

Scalability, which actually covers two different concepts [van Steen et al., 2021]. First, scalability in terms of transaction processing capacity, and can be considered as throughput. Then, scalability in terms of the number of users, which is positively impacted by the open membership, the lack of any centralized component and the absence of trust assumptions on any third-party [van Steen et al., 2021]. Scalability is consequently strongly related to decentralization. As set in assumptions (cf. Section 1.3), only large-scale IoT networks are considered in this thesis, which implies that high scalability is compulsory. Despite being crucial, it is also one of the most difficult metrics to evaluate accurately.

Network overhead. In the context of distributed ledgers, network overhead refers to the amount of additional data transmitted over the network that is required to support the consensus protocol, verify transactions, or maintain the distributed ledger system. This additional data can include message headers, authentication and encryption data, transaction data, and consensus-related data, and can increase the total amount of network traffic required to operate the system. This is a metric of interest for the Internet of Things, as a high network overhead will limit the scalability in terms of nodes.

Storage overhead is the additional storage space required to maintain the ledger, including transaction data and metadata, on each node in the network. Internet of Things devices e.g., sensors, may have very low storage. A high storage overhead excludes low storage devices from contributing to the consensus method.

2.4.2 Performance of consensus methods

Using the above-mentioned performance metrics (cf. Section 2.4.1), we discuss how consensus methods compare with each other. In particular, we highlight which distributed ledger technologies are suitable for the Internet of Things from a performance perspective, i.e., regardless of security and privacy concerns. The discussion is summarized in Table 2.1.

Proof of Work. The PoW is a consensus method based on a computation-intensive cryptographic puzzle. The difficulty of the cryptographic challenge directly defines the security and the performance of the network. By design, PoW-based blockchains usually have a low throughput and a high latency to allow time for the nodes to check the new transactions' correctness. Storage is troublesome as well, as the size of the ledger is continuously increasing and can not be stored on a device with low capacity. PoW is consequently not adapted for the Internet of Things.

Proof of Stake. The proof of stake removes the computation race between nodes, alleviating computation requirements compared to the proof of work. However, throughput is still low considering IoT requirements [Raghav et al., 2020]. As it is a monetary-based approach, it is often excluded for IoT use cases [Raghav et al., 2020, Salimitari et al., 2020], as it requires constrained devices to hold a cryptocurrency. The proof of stake also tends to concentrate the power into the hands of the richest nodes, centralizing the network to some extent [Salimitari et al., 2020]. The **Proof of Authority** is very similar to the proof of stake but does not introduce monetary concepts.

Delegated Proof of Stake. Electing miners and nodes responsible for the network management significantly improves throughput and latency, but centralizes the network. Besides, it also requires holding a cryptocurrency to participate in the voting, limiting its adoption in the Internet of Things.

Proof of Elapsed Time. The proof of elapsed time has a low latency and a high throughput and can be considered IoT-friendly [Salimitari et al., 2020]. The only drawback is the need to verify the timer's execution using a dedicated *Trusted Execution Environment*. For PoET, it is usually done using Intel's SGX software, which makes it partly centralized and limits its use in large-scale networks.

Practical Byzantine Fault Tolerance. Similarly to PoET, PBFT features high throughput, low latency and low computational overhead making it appealing for IoT networks. However, PBFT requires a lot of messages to achieve consensus and has a high network overhead, strongly limiting its scalability and restricting its use to private blockchains and small IoT networks. The **delegated Byzantine Fault Tolerance** does not require the participation of all nodes in the consensus, aiming at solving the network overhead issue, but at the cost of higher latency making it unsuitable for the Internet of Things [Salimitari et al., 2020].

IOTA. IOTA has been specifically designed for the Internet of Things requirements. IOTA provides high throughput and low latency for transactions, as well as high scalability [Alshaikhli et al., 2022]. Nodes do not require to store the whole ledger to run, rather relying on automated snapshotting, limiting storage overhead.

Consensus method	Scalability	Latency	Throughput	Adversary tolerance	Network overhead	Storage overhead	IoT suitable
PoW	High	High	Low	<51% computing power	Low	High	✗
PoS	High	Medium	Low	<51% stakes	Low	High	?
PoA	High	Medium	Low	<51% stakes	Low	High	?
DPoS	High	Medium	High	<51% validators	n/a	High	?
PoET	High	Low	High	n/a	Low	High	✓(small scale)
PBFT	Low	Low	High	<33% faulty replicas	High	High	✓(small scale)
dBFT	High	Medium	High	<33% faulty replicas	High	High	✗
IOTA	High	Low	High	<33% computing power	Low	Low	✓

Table 2.1 – Performance of consensus methods - based on [Salimitari et al., 2020]

2.5 Conclusion

In this chapter, we introduced the background on usage control and on distributed ledger technologies needed to understand the context and the contributions of this thesis. First, we highlighted the purpose of usage control and information flow control to monitor data usage and data dissemination (Section 2.1). Then, distributed ledger technologies have been extensively discussed, from general considerations (Section 2.2.1) to privacy challenges (Section 2.3). Finally, we addressed performance aspects of distributed ledgers, which are paramount considering Internet of Things constraints. The performance of a distributed ledger technology is deeply correlated to the method used to achieve consensus.

In the following chapters, the different contributions of this thesis are described. The next chapter 3 introduces a set of tools to design a framework for efficient, privacy-preserving zero-fee transactions for the Internet of Things (*Objective 1*). The framework is based on the aforementioned technologies, notably usage control, distributed ledgers and privacy-enhancing technologies for transactions, motivating this background chapter.

Chapter 3

Solving the Internet of Things Trilemma: Performance, Security and Privacy

Contents

In the context of the Internet of Things, as defined in Section 1.3, a framework for zero-fee transactions in the Internet of Things (*Objective 1*) should consider transactions to be granted access to physical devices , e.g., cars and doors, but also to the data generated by the devices, as those data are valuable. To this end, we will focus in this chapter on designing a framework that:

- provides access control on physical devices and goods;
- provides access and usage control on data;
- has no fee, or very low fees, to enable micro-transactions;
- preserve the privacy of both participants of a transaction;
- considers the performance and the security requirements of the Internet of Things.

This chapter is structured as follows. First, the IOTA technology, a distributed ledger with zero-fee transactions is introduced in Section 3.1. IOTA uses a DAG to build its transaction graph (cf. Section 2.2.3), and was designed to answer the performance requirements of the Internet of Things. IOTA is the key technology of the framework, from which the other tools are derived. Then, the framework and its components are detailed in Section 3.2. Due to the Internet of Things performance constraints, an optimization scheme based on the integration of usage control is proposed and evaluated in Section 3.3. The process of integration is fully detailed in the following Chapter 4. Finally, a security and privacy threat assessment is conducted in Section 3.4. A conclusion of the chapter is provided in Section 3.5.

3.1 IOTA distributed ledger

In this section, we introduce IOTA as a key technology of the proposed framework, since most of the tools presented are derived or adapted to this distributed ledger. As discussed in Section 2.4, blockchain technology has several security and performance drawbacks for the Internet of Things which limits its adoption. Besides, transaction fees in some public blockchains can be greater than the actual transaction value, making micro-transactions impossible. Removing transaction fees in blockchains is an intricate issue since transaction fees are used as an incentive for creators of blocks to contribute to the network. These different issues justify the need to introduce a new paradigm for cryptocurrency transactions.

3.1.1 A DAG-based transaction ledger

IOTA builds its transaction graph using a directed acyclic graph (cf. Section 2.2.3) called the *Tangle*. The difference between the tangle and a blockchain is represented in Figure 3.1. IOTA has properties common to DAG-based distributed ledgers, i.e., low transaction fees, disintermediation, and high throughput. As a reminder of Section 2.2.3, transactions in a network based on a directed acyclic graph are built as follows. The transactions are the vertices of the transaction graph. There is only one transaction by vertice. When a new transaction arrives, it must approve two previous transactions. The approval is materialized by an edge. These approvals are represented by directed edges. If there are no directed edges between two transactions A and B, but there is a directed path of length at least two from A to B, we say that A indirectly approves B. There is an original transaction, which is approved directly or indirectly by all the transactions, called *the Genesis transaction* in IOTA.

The genesis is described in the following way. At the beginning of the tangle, there was a single address with a balance that contained all IOTA tokens. The Genesis transaction sent these tokens to several other addresses belonging to the founders. All of the tokens were created in the genesis transaction and as there is no mining, it is impossible to create iota tokens without *taint*. A token is considered tainted if it belongs to at least one identifying address on the IOTA ledger. 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 [Tennant, 2017].

Finally, for a transaction site on the IOTA ledger, there is an associated *weight* w. The general idea behind this notion is that a transaction with a significant weight is more important and trusted. Along weight, the notion of *cumulative weight* w_{cum} of a transaction t is also introduced, as the sum of its weight plus the weight of all transactions validating (including indirectly) the transaction t. In Figure 3.1, both weights and cumulative weights (respectively w and w_{cum}) are given inside the sites of the DAG. For instance, the transaction F has a weight $w_F = 2$, is directly validated by B and E and indirectly by A, C, and D. Therefore, the cumulative weight of F is $w_{cum,F} = (w_A + w_B + w_C + w_D + w_E) + w_F = 1 + 1 + 1 + 1 + 4 + 2 = 10$.

3.1.2 Consensus method

In IOTA, to issue a transaction, users must work to approve other transactions, so that users themselves are contributing to the network's security. The nodes check in particular if the approved transactions are not conflicting. If a node finds a transaction conflicting with the tangle history, the node will not approve the conflicting transaction directly or indirectly. In case a node issues a new transaction that validates conflicting transactions, it is at risk that other IOTA nodes do not approve its transaction. As a

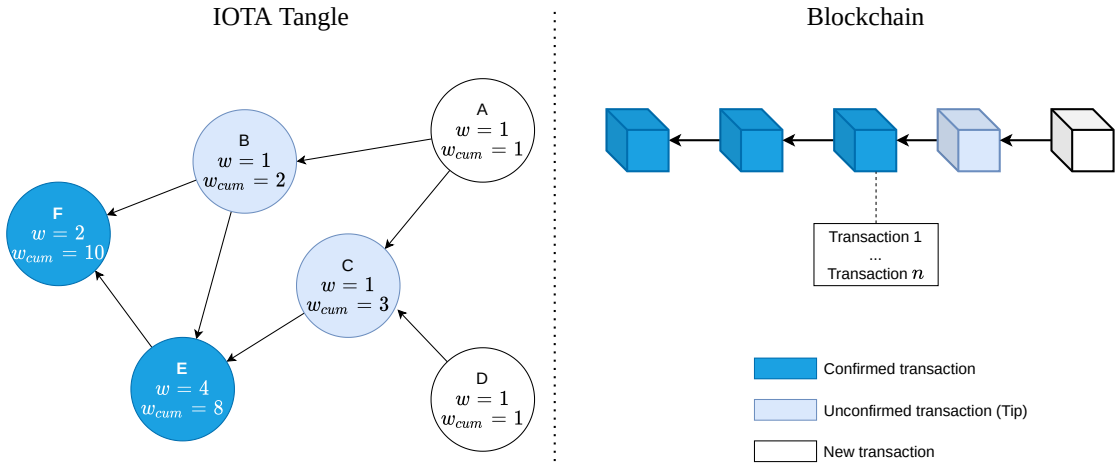


Figure 3.1 – Transaction ledger in the Tangle (directed acyclic graph) and a blockchain. Each transaction site on the DAG has a weight w and an indirect cumulative weight w_{cum}

transaction receives additional approvals, i.e., its weight is increasing, it gets a higher level of confidence. It consequently becomes harder for the system to accept double-spending transactions.

A new transaction, i.e., not approved by other transactions, in the IOTA network, is called a *tip*. To issue a new transaction, an IOTA node proceeds to the following steps:

- The node picks two other transactions to approve according to a *tip selection algorithm* (TSA);
- The node checks if the two transactions are conflicting;
- The node solves a cryptographic challenge to prevent spam. It works exactly like Bitcoin's proof of work, but the challenge is not designed to pick the next miner, but only to make sure a node can not spam the network by pushing too many transactions simultaneously.

The IOTA network is asynchronous, which means that the tangle may contain conflicting transactions. However, the nodes *do not have to achieve consensus* on which transactions are valid and should be in the tangle, which is a significant difference with blockchains. That implies that all the transactions can be in the ledger. However, if there are two conflicting transactions, the nodes have to decide which transaction will be *orphaned*, i.e., not validated indirectly by new transactions.

Tip selection algorithm. In IOTA, the tip selection algorithm (TSA) determines which tips should be approved and referenced in new transactions. It aims to achieve

a balance between security and network efficiency. The tip selection algorithm in IOTA is based on a *Monte Carlo Markov Chain* (MCMC). The MCMC algorithm takes into account various factors such as the cumulative weight of transactions, the transaction arrival rate, and the network topology. By considering these factors, the algorithm selects tips that are more likely to be included by other transactions in the Tangle, increasing the chances of approval for the users and overall transaction confirmation. The tip selection algorithm plays a crucial role in the security and performance of the IOTA network. It helps prevent the concentration of approvals on a single branch of the Tangle, improving the resilience against potential attacks. Additionally, the algorithm aims to maintain a balanced distribution of transaction approvals across the network, ensuring efficient propagation and confirmation of transactions.

3.1.3 Benefits of IOTA

IOTA is a DAG-based distributed ledger and therefore has the attractive features of this technology. These benefits have been listed in Section 2.2.3 and are the following:

- writing transaction is not energy-intensive, enabling low-power devices to participate in the network;
- throughput is high after the bootstrapping stage and increases with the number of users;
- users add their transactions directly to the network without relying on any intermediaries, such as miners or gateways.

In particular, the high throughput answers the scalability requirements of large-scale networks (cf. Section 1.3) as it enables to process the high volume of transactions. In addition, IOTA does not have any transaction fee due to the removal of miners (*Objective 1*), while other distributed ledgers using a DAG have small fees [Churyumov, 2017, LeMahieu, 2017]. IOTA is also largely studied and used in the literature [Alshaikhli et al., 2022, Conti et al., 2022, Guo et al., 2023, Sadi et al., 2023], and the IOTA foundation, the main actor supporting the development of IOTA, also contributes to the scientific literature[1]. Finally, IOTA is built to be resistant to an adversary with a quantum computer [Popov, 2017], which is an interesting property from long-term security that distinguishes IOTA from its competitors.

3.1.4 Limits of IOTA

Reliance on the Coordinator. The IOTA network, in its current version (1.0, June 2023), relies on an entity called *the coordinator*. The coordinator is a centralized component that

[1] https://www.iota.org/foundation/research-papers

was initially introduced to ensure the security of the network during the bootstrapping phase. The coordinator's purpose is to issue *milestones* that validate transactions and to help prevent certain types of attacks, notably the double-spending attack. Even though the coordinator is a temporary device that is supposed to be removed in the IOTA 2.0 [Popov, 2020], it is still currently used and creates the following issues:

- *Centralization*: The coordinator gives a single entity control over the network's security, making it a point of failure and potential vulnerability. Indeed, if the coordinator is stopped or under a denial of service attack, the transactions can not be validated anymore;
- *Trust*: the coordinator is run by the IOTA Foundation, which should be trusted by the participants. It goes against the trustless and transparent nature of blockchain networks;
- *Timeline*: the timeline for the removal of the coordinator has not been set yet, and the network has been running since 11 July 2016.

Scalability issues. IOTA distinguishes (in version 1.0) two different regime types, based on the number of simultaneous tips in the network [Popov, 2017]. The *low load regime* has a small number of tips, typically one tip. The flow of transactions is so small that it is unlikely that two different transactions approve the same tip. The low load regime is characterized by a low latency. Conversely, the *high load regime* has a large number of tips. The flow of transactions is large, and both computational delays, as well as higher network latency, increase the likeliness of two simultaneous transactions approving the same tip.

3.1.5 IOTA 2.0 and the Coordicide

The removal of the coordinator is a process called *the coordicide* [Popov, 2020] that is due to thoroughly change the IOTA network mechanisms. We provide an overview of the anticipated changes in IOTA 2.0, the post-coordicide version of IOTA.

Node accountability. In a network where the coordinator has been removed, several applications require to associate transactions and messages with the node which issued them. This is true for the rate control mechanism and the voting-based consensus detailed next. To make the node accountable, IOTA 2.0 requires the introduction of a *global node identity*. Node identities are achieved using common public key cryptography to sign data and link it to the issuing node. The issuing node adds its public key to every signed message so that every node can verify its authenticity.

The introduction of node identities is troublesome as it exposes IOTA to *sybil attacks*, in which attackers create multiple fake identities to get a disproportionate weight in the network. To solve this issue, IOTA 2.0 relies on *mana*, which is obtained when a node

issues transactions. Mana is the basis of a reputation system, to identify the reliable nodes which contribute the most to the network.

Rate control mechanism. In an overload scenario, where the nodes are trying to issue more transactions than the overall network can handle, e.g., due to its physical limits, particular transactions originating from the most heavily contributing nodes should be either limited or penalized. It is achieved in IOTA 2.0 by using an *adaptive proof of work*. The adaptive proof of work is determined thanks to three parameters: the base difficulty, an adaptation rate that depends on the mana owned by the node, and finally, a time window that defines the granularity of the rate control mechanism. The shorter the time window, the quicker the network reaction is when the node issues too many transactions.

Consensus and voting. In IOTA 1.0, the consensus is achieved by applying the tip selection algorithm, i.e., the mechanism used by nodes to select the transactions to approve, based on a biased random walk. IOTA 2.0 is based on a new consensus method, called *Shimmer*. The idea behind this new consensus mechanism is to care only about the knowledge of a very small subset of nodes, instead of the opinion of every other node. The actual consensus method used by IOTA 2.0 is called *Fast probabilistic consensus* [Cooper et al., 2015, Mallmann-Trenn, 2017], and relies on the idea that randomized voting, i.e., random queries, are sufficient for good performance, and due to the small message complexity, makes the protocol scalable. Another advantage of this randomness is the improved robustness in less reliable networks and situations with dynamicity, where nodes join and leave the network frequently.

3.2 A framework for privacy, performance and security in the Internet of Things

To answer the performance, privacy and security needs of the Internet of Things, and to enable zero-fee transactions (*Objective 1*), the following framework is proposed. It is designed to address the different needs simultaneously, as well as to cover the different categories of Internet of Things use cases, by considering both data and physical accesses. The different components are detailed next, after a general overview.

3.2.1 Framework overview

The proposed framework is made up of the following components (cf. Figure 3.2):

1. IOTA technology, as a suitable distributed ledger technology to answer IoT performance requirements and its zero-fee transactions;

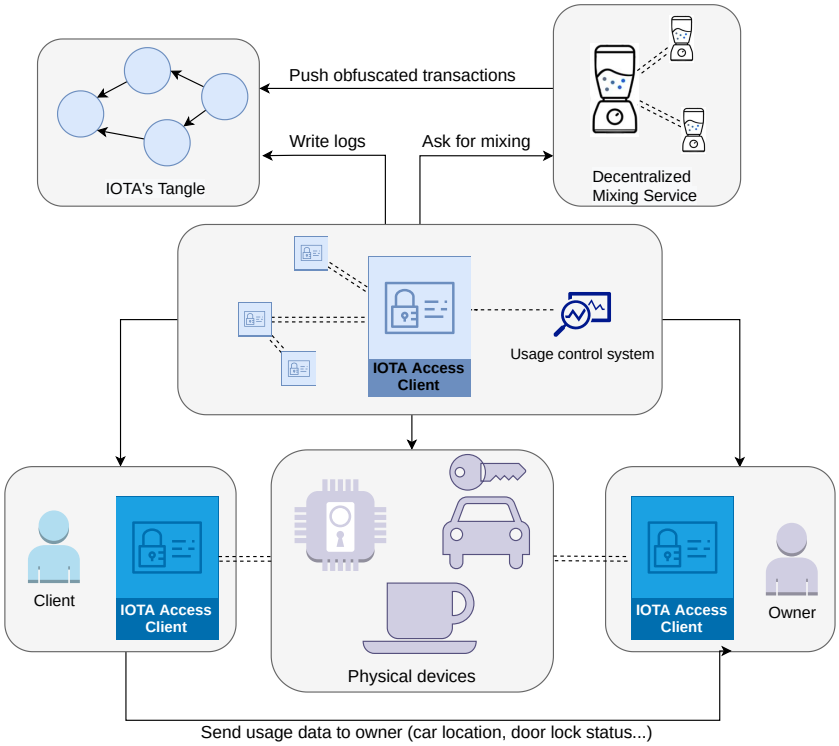


Figure 3.2 – Framework to monitor data usage and physical access to IoT devices based on privacy-preserving transactions

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, to monitor the usage and dissemination of the data in the system. The UCS relies on a Trusted Environment Execution of the device of the monitored user;

4. a decentralized mixing service coupled with merge avoidance (cf. Section 2.3.3), to obfuscate the transactions and improve the privacy of users.

IOTA and usage control have been already described in previous sections (respectively, Section 3.1 and Section 2.1). In the following, we focus on the two other components of the framework, IOTA Access and the decentralized mixing service for IOTA.

3.2.2 IOTA Access

IOTA Access is a lightweight access control framework tailored for resource-constrained networks. It is based on XAIN's FROST project, which is the byproduct of Leif-Nissen

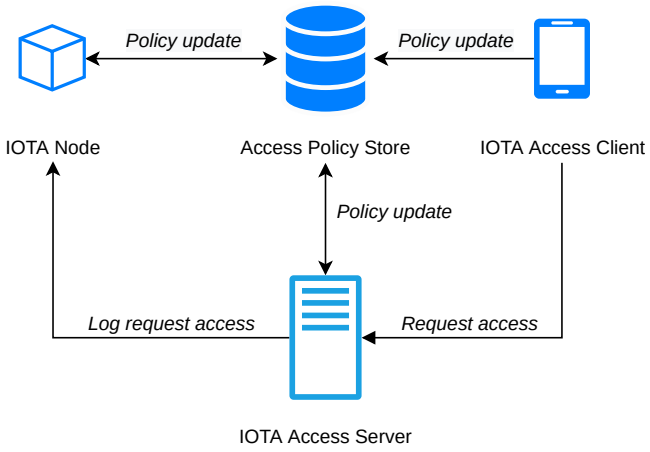


Figure 3.3 – IOTA Access framework representation

Lundbaek's Ph.D. Thesis at Imperial College London [Lundbaek, 2020]. The framework is also expanded with relevant concepts, such as obligations and the delegation of access-control policies, to particularly address the need for reliable and secure human-machine interactions in the IoT. Notably, IOTA Access introduces the concept of *actions*, which are all the authorized actions that a user can perform. Actions are based on *attributes*, *obligations* and *conditions*, very similar to the UCON notions with the same terminology (cf. Section 2.1), but used in the context of physical access instead of data access. In particular, empowered by obligations and conditions, it is possible with IOTA Access to: 1) grant or deny access at any time; 2) charge users for physical access with zero-fee transactions; 3) set complex access restrictions based on conditions and obligations. To do so, IOTA Access is divided into three components, represented in Figure 3.3:

- the *IOTA Access client*, which is a mobile access client used as the user interface, both for creating policies and initiating access requests. IOTA Access contains an Android-based reference implementation;
- the *IOTA Access server*, the embedded software executed on the device for which access will be delegated;
- the *IOTA Access policy store*. It consists of interface servers for managing policies;

The interactions between the IOTA Access components are as follows, also represented in Figure 3.3. Users deploying an IOTA Access client can either request access to devices or update access policies enforced on their own devices. In the first case, if a user requires access to a device, the IOTA Access server evaluates the request against the policy, then logs the request on the IOTA ledger, possibly using a private Tangle

rather than the public ledger. In the second case, a user-triggered policy update causes a chain of network exchanges between the IOTA Access components. First, the IOTA Access clients directly update the policies in the IOTA Access policy store, which forwards the update to the IOTA Access server and notifies the IOTA node of the update for logging purposes.

IOTA Access is used in the proposed framework to extend the zero-fee privacy-preserving transactions to the Internet of Things devices, instead of restricting them to data-sharing use cases.

3.2.3 Decentralized mixing for IOTA.

Coin mixing has been introduced in Section 2.3.3 as one of the most common tools to obfuscate transactions in distributed ledgers. Coin mixing aims at removing the link between the sender and the receiver of a transaction, to complicate re-identification attacks. However, centralized coin mixing creates significant security and privacy risks. The centralized mixing service is a likely target for denial of service attacks and must be trusted, as it may steal the funds or keep the transaction records for itself (cf. Section 2.3.3). However, decentralization is troublesome due to edge insertion attacks. While this is true for most public blockchains, which require transaction fees, it is different for IOTA due to its transaction mechanisms.

Indeed, Sarfraz *et al.* [Sarfraz et al., 2019] designed a decentralized mixing scheme for IOTA that leverages zero-fee transactions and does not require changes in the IOTA protocol. The mixing scheme has the following attractive features:

- protection against signature forgery and guarantee that even in the presence of malicious adversaries during mixing, no participant can reveal portions of his/her private key of the input address. This property is guaranteed by *multi-signatures*;
- fully decentralized mixing operation;
- no mixing fees from participant;
- anonymity, availability and correctness are guaranteed (cf. Section 2.3.3);

Mixing protocol design. We now detail the actual decentralized mixing protocol. It is composed of three different phases: *settlement*, then *output shuffling* and finally the *transaction* [Sarfraz et al., 2019]. In case the protocol can not be completed, another additional *fallback* phase in introduced. The protocol is detailed with n peers, but to achieve anonymity, the protocol requires at least two participants. The first settlement phase unfolds as follows:

(1) each peer $i \in \{1, ..., n\} |\ k_i, ...k_n \in 0, 1, 2, ...$ determines key indexes $k_1, ..., k_n$ and security levels $s_1, ..., s_n \mid s_i, ..., s_n \in 1, 2, 3$. s is a security level that results in a

key of length $l = s * 27 * 81$ trytes. k_i and s_i are used to generate digests d_i for $i \in 1, ..., n$ and private keys $pr_1, ..., pr_n$;

(2) each peer shares the digests $d_1, ..., d_n$ to generate $M_1, ..., M_n$ multi-signature addresses such that the addresses-digests mapping M is

$$M = \left\{ \begin{array}{l} M_1 \rightarrow D_{i(1)}, \ldots, D_{n(1)} \\ M_2 \rightarrow D_{i(2)}, \ldots, D_{n(2)} \\ \ldots \\ M_n \rightarrow D_{i(n)}, \ldots, D_{n(n)} \end{array} \right\}$$

(3) for each generated multi-signature address, every $n - i$ peers share their private keys $pr_1, pr_2, ..., pr_n$ to make 1-N mapping for address ownership;

(4) Each peer makes a transaction T_i to the multi-signature address M_i;

Mixing peers validate the multi-signature address by sharing digests. If the address validation fails, then a malicious participant may be involved in the settlement process. The protocol aborts and all participants receive a notification. The settlement phase is followed by the output shuffling phase. The output shuffling phase aims to randomize the set of output addresses declared by the peers, preventing peers from mapping input addresses to output addresses. Shuffling unfolds as follows:

(1) each peer creates an IOTA output address $O_1, ..., O_n$ corresponding to input addresses $A_1, ..., A_n$;

(2) each participant generates a key pair (Enc_i, Dec_i) and broadcasts its public key:

(3) once all keys have been broadcast, the first peer creates a layered encryption of its output address, which means that it encrypts its output address with all the keys of the other peers. The first peer then forwards the ciphertext to the next participant;

(4) each following participant i up to the last one n decrypts the outermost layer of encryption for all ciphertexts with its corresponding decryption key Dec_i. Each peer receives a list of ciphertexts with $i - 1$ size;

(5) after decrypting the outermost layer of all the ciphertexts, each participant i randomly shuffles the ciphertexts and then creates a nested encryption for its output address O_i;

(6) the last participant decrypts all the ciphertexts and shuffles the final decrypted list with its output address. Finally, the final list of output addresses is forwarded to the other participants.

Upon receiving the decrypted list, the participants check if their output addresses are actually in the list. If there is a duplication of output addresses or if any output address is missing from the list, the protocol interrupts and peers enter the fallback phase. Additionally, if the decryption fails or multiple decryptions lead to the same output, the corresponding participant aborts the protocol and notifies every participant and all the participants enter the fallback phase as well. If the shuffling phase unfolded correctly, the peers can enter the last transaction phase.

The purpose of the transaction phase is to build and push the final transaction to the network, i.e., the iotas from each input address $A_1, ..., A_n$ will be transferred to each corresponding output address $O_1, ..., O_n$. As input addresses are multi-signature, it is required that all the participants sign each transaction. The transaction phase is made up of the next steps:

(1) every participant creates a mixing transaction that spends IOTA tokens from each of the input addresses in $(A_1, ..., A_n)$ to $(O_1, ..., O_n)$.

(2) the participant i signs the transaction according to the specification of the IOTA protocol and broadcasts the signature;

(3) when it receives a valid signature from another peer j, the peer i adds all the signatures in the bundle. Participant i checks if any other participant has spent their money saved for mixing. If a participant spent its funds, the protocol is aborted and the fallback phase is triggered;

(4) the participant i receives two transactions to approve as part of the IOTA transactional protocol (cf. Section 3.1), performs the proof of work to prevent spamming, then generates the transaction hash and broadcast the bundle to the IOTA network.

If the transaction phase is completed, the mixing process is achieved and the mixing desired properties, i.e., anonymity, availability and correctness, are provided. However, at any step of the mixing process, the protocol can enter the fallback phase in case of misbehavior. The protocol needs to be run again with new unused output addresses for correct protocol execution. The full protocol with a correct execution, i.e., excluding the fallback phase, is represented in Figure 3.4.

3.3 Performance optimization

The proposed framework is designed to answer the performance, security and privacy needs of the Internet of Things. To ensure the system is consistent with the performance requirements, we assess the performance of the proposed framework in this section. In

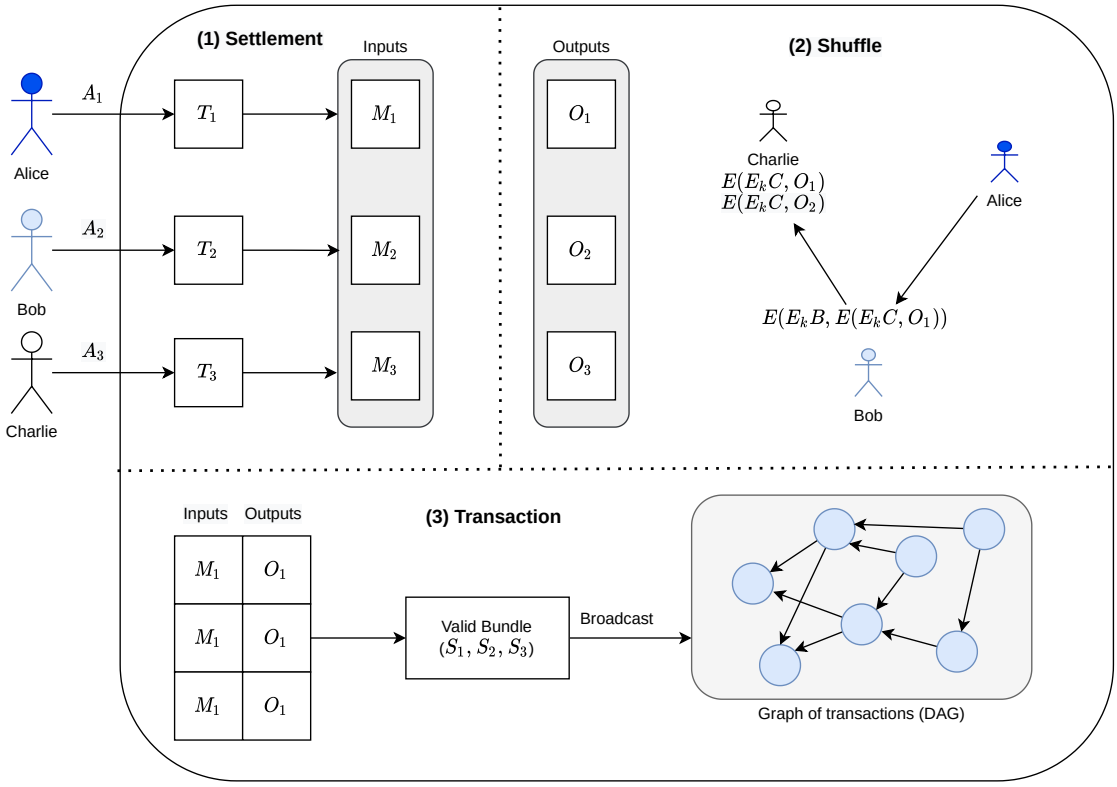


Figure 3.4 – Three phases of decentralized mixing on the IOTA network with three peers [Sarfraz et al., 2019].

IOTA 2.0 [Popov, 2020], IOTA is decentralized and has a high throughput, low latency, high scalability and low storage overhead due to pruning. However, the introduction of the UCS, IOTA Access and the mixing service likely introduces computation and network overheads, requiring further testing.

The evaluation of IOTA's network metrics, such as scalability and throughput is not addressed by this evaluation. This section aims at demonstrating the framework is viable even though it relies upon numerous technologies, and is evaluated in terms of:

- *Quality of service*: the entire chain of events leading to access is completed in a reasonable time frame;
- *Computational power*: the hardware requirements are reasonable as regards Internet of Things constraints.

Regarding the Quality of Service, the focus is given to determining the time required for a user to be granted access after its payment. Average, minimum and maximum times will be considered. Additionally, several hardware configurations are tested, to assess whether the UCS can be deployed on resource-constrained devices.

3.3.1 Configuration

Configuration of the IOTA node. To reduce computation and network overheads introduced by the usage control system, several optimizations are set up: the usage control system deploys an IOTA node to integrate the IOTA network. This integration provides several benefits. The UCS can prioritize its transactions and perform local analysis on its ledger without querying the other IOTA nodes. Secondly, the IOTA node is configured, by removing the possibility to compute proofs of work for other users and by using *spammers* to speed up the network by validating tips (cf. Section 3.1), in particular when the network is in a low load regime. Spammers are useful for testing as well, as the results are different whether the tests are conducted in a low load or high load regime. The principles of integration, as well as its benefits, are fully detailed in the dedicated Section 4.2. As IOTA transaction throughput increases with the number of users, i.e. when many users push new transactions, it is relevant to use spammers that create zero-value transactions[2] 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 to focus on usage control.

Methodology and network configurations. We 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. These operations correspond respectively to the calls `buildTransaction`, `push` and `checkTransaction` in the sequence diagram of Figure 3.5. Tests are conducted in three different configurations: (1) the IOTA remote node which is a 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 3.6. For each test, one thousand samples ($N = 1000$) are used. The resulting experimental measurements are summarized in Table 3.1.

Note on reproducibility. The tests provided in this section are performed on the IOTA public test network. This is troublesome for reproducibility reasons (cf. Section 4.3.1), but these tests have the following benefits:

- it is a first evaluation on a large-scale network to show that integration is a relevant process for performance. Addressing the challenges of large-scale networks is one of the purposes of this thesis work, justifying an evaluation on an appropriate network (cf. Section 1.3) ;

[2] zero-value transaction do not transfer iotas and the transaction corresponding amount is zero, hence their name

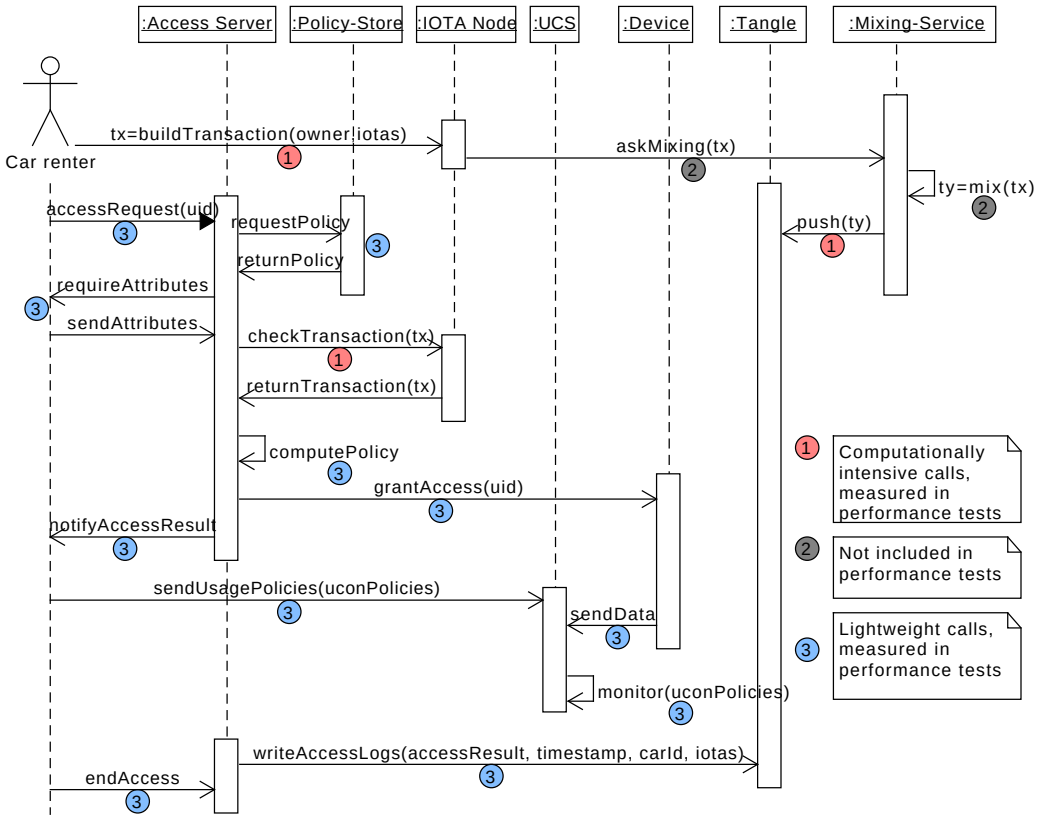


Figure 3.5 – Workflow of a data access request using IOTA, IOTA Access and the mixing service.

- it enables estimating the outcomes of integration on a public network with a subsequent high number of nodes, rather than on a test network with fewer nodes.

As a consequence, these tests are still considered in this chapter, not to demonstrate the validity of integration, but for informative purposes. Testing to demonstrate the validity of the integration process is provided in the dedicated section (Section 4.3) of Chapter 4.

3.3.2 Evaluation results.

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 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 or one of the peers disconnects. The confidence interval is $I_{rcn} = \bar{t}_{rcn} \pm 1.96\frac{\sigma_{rcn}}{\sqrt{N}} = 5316 \pm 287ms$.

The results show that the solution can be deployed on a machine with low computation capacities. However, with delays of up to 26 seconds to create, validate and push a transaction to the network, this can be unsatisfying in some use cases, e.g. accessing a vehicle or opening a door lock.

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 supporting the optimizations. This corresponds to the high-end Raspberry Pi 4 Model B specifications[3].

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 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 very similar to 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 3.6. 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 run 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

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

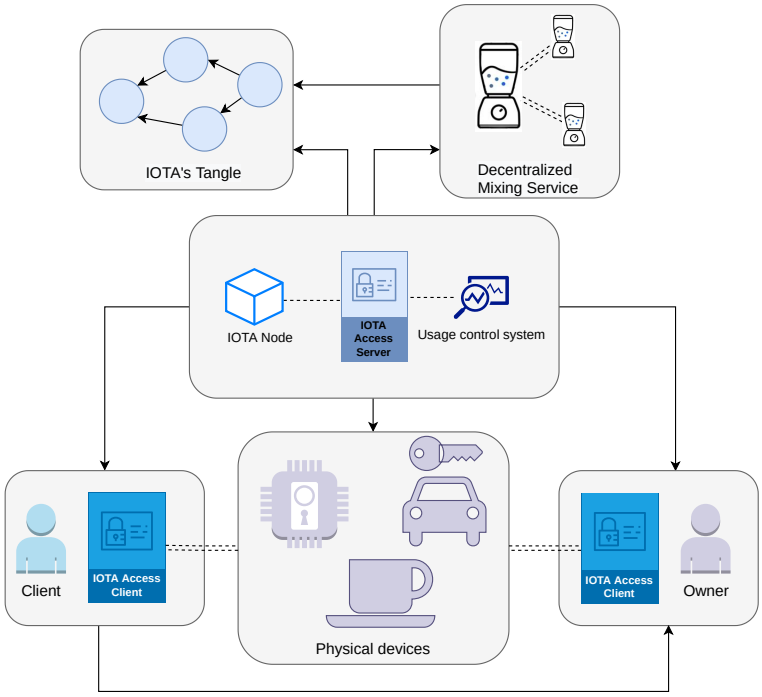


Figure 3.6 – IOTA node deployment for optimization

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 outcome:

1. a 17.5% decrease on the average transaction time;
2. transactions can be processed very quickly, taking a minimum of 10ms instead of a minimum $363ms$;
3. the maximum time only drops from $12209ms$ to $9830ms$, which remains convenient for real-life applications;
4. almost half (48%) of the transactions are processed within a second, compared to 34.5% for transactions using a remote node.

Additional calls. While the performance tests are conducted on the three computationally intensive calls, the other category called *lightweight calls* (cf. Figure 3.5) has 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 correlated to the time needed to communicate between the car renters, the access server, and the policy store. These calls took under 1ms to be achieved in our setup since they were all running locally on the same device. These three entities can be considered as close (in space) and the time for all these calls is negligible compared to the `buildTransaction`, `push` and `checkTransaction` calls.

The remaining calls have different behavior. `computePolicy` is composed of several boolean operations, taking a negligible time. `sendUsagePolicies` and `sendGPSData` are operations that are continuously 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. 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 (cf. Section 3.3). 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.

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

Table 3.1 – Performance measurements for different test configurations

3.4 Security and privacy evaluation

As part of the research *Objective 5* (cf. Section 1.2) , this section analyzes the privacy and security risks in the system. It distinguishes the risks to privacy and the risks to security if the usage control system is compromised. To better identify the privacy and security threats and how they are mitigated by the proposed framework, we first introduce a car sharing illustrative scenario.

3.4.1 Illustrative scenario

Car sharing is a model of car rental where people rent cars for a short period. They differ from classic rental models in that the owners of the cars are individuals instead of an agency. The context is dynamic as many users may enter or exit the car club during the day. For the users to interact with the system, a mobile application is used for registration and requiring access to the cars.

For security purposes, car owners have the right to monitor their cars and collect their real-time location. However, the location of the car also produces 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 cars. Similarly, car renters define usage control policies using the same application. Car renters use public distributed ledger technology to make decentralized and transparent transactions. Car renters as well as car owners have one or several addresses on the IOTA network, to either send or receive transactions.

3.4.2 System agents

The agents of the car renting system can be summarized as follows:

- the car owners put their vehicles on the renting market;
- the car renters pay for renting the vehicles;
- the cars themselves send data to the owners such as location, and whose access is monitored;
- the Access Server (AS) is responsible for managing the access to the cars;
- the Usage Control System (UCS) monitors the data generated by other agents;
- the mixing server obfuscates the transactions to preserve privacy.

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 monitors the information flow, blocking data dissemination to forbidden actors.

3.4.3 Security and privacy threat model

Privacy threat model. Depending on the data the attackers are obtaining, and using the LINDDUN threat evaluation framework [Wuyts et al., 2018] (cf. Section 2.3.1), 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;

- *identification*(I): the attacker can link the pseudonym to the real identity of the car renters or the car owners, breaking anonymity;

- *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. Conversely, non-repudiation can be a threat to legitimate users if an attacker can prove that a user has done some sensitive actions e.g., an illicit transaction [Wuyts et al., 2018]. In our scenario, non-repudiation is not considered a threat, but repudiation is;

- *disclosure of information*(D): an attacker can access data about a user without having the proper access rights. Inference attacks can be included in this category and are defined as "attacks where the attacker has used existing knowledge to aid the attack" [Henriksen-Bulmer and Jeary, 2016]. An inference attack occurs when an attacker can infer information from apparently harmless information. For example, in our scenario, an attacker could infer working hours by gathering transactions timestamps;

- *unawareness*(U): unawareness occurs if car renters are 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 aftermath, inference attacks [Wuyts et al., 2018]. Non-compliance(N) is considered 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 [The European Parliament and the Council of the European Union, 2018]. Finding technical solutions to compliance issues is an active field of research [Haque et al., 2021].

Security threat model Considering the agents defined in the car renting system, the threat model distinguishes *four attacker types*:

1. the single car owner, who has legitimate access to some sensitive data of the car renters;

2. several car owners colluding to gather bigger sets of data;

3. the mixing server that may secretly keep the links between car renters and car owners that 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*. They will respect the system rules by granting access to their cars upon receiving payments but will snoop on the data of the users requesting their services. Honest-but-curious attackers are assumed to rely on transactional content only to re-identify users, rather than network-level information, e.g. IP addresses. External attackers are conversely *malicious* and actively try to neutralize the UCS to enable car owners to disseminate their data.

Concurrently, there are *security risks* because a single agent of the system - namely the UCS - is responsible for data protection. The UCS itself is considered as *trusted*. External attackers can however be interested in neutralizing the UCS to enable the collusion of car owners. Similarly to the privacy threat analysis using LINDDUN, we use the STRIDE security threat modeling [Howard and Lipner, 2006] to identify the security threats for 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 [Kelbert and Pretschner, 2018];

- *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 by the LINDDUN privacy threat model and are excluded from the security threat modeling. Finally, an external attacker can conduct an Elevation of privilege(E) by leveraging vulnerabilities as illustrated in [Babil et al., 2013] to bypass the UCS restrictions. These attacks are very diverse and implementation-dependent, therefore considered out of the scope of this thesis work.

3.4.4 Privacy threats and mitigations.

Table 3.2 describes for inference attacks each possible combination of attackers, the data types they have access to, where data is stored and an instance of a privacy leakage

associated with this risk. Table 3.3 describes other privacy threats 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 3.2 – Inference attacks according to the attackers' profile

Any user has access to the Tangle's transactions, which are public and contain privacy-sensitive timestamps, users' addresses and values, i.e. how many iotas are sent to a car owner. Any honest-but-curious attacker can attempt to use the blockchain transactions for inference attacks, e.g. use the number of tokens in the transactions to infer the purpose of the payment. The merge avoidance mechanism integrated into our framework reduces the risk of inference by splitting the transactions into several smaller ones, thus making it harder to guess the purpose of the transactions (cf. Section 2.3.3).

Additionally, car owners may infer privacy-sensitive data from the car renters' location data. The location of the car renters may reveal their driving habits, their jobs, their religion or their hobbies. Besides, colluding car owners can merge their data about a given user to increase the inference's quality or increase the number of users in their databases to improve their value. If an external attacker successfully neutralizes the UCS, as reported in Section 3.4.5, car owners can freely share users' data and disseminate them 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 3.3 – Threats to privacy and their mitigation

Table 3.3 summarizes the privacy threats for the car renters excluding inference attacks, here above presented. By observing transactions, an honest-but-curious attacker may try to link the sender and the receiver. This risk is mitigated by the mixing server. Furthermore, when car renters use the same address multiple times for outward transactions, they are exposed to re-identification (cf. section 2.3.2). We generate a new address in our framework for each outward transaction and forbid address reuse to mit-

igate this risk. Moreover, as the mixing service is decentralized (cf. Section 3.2.3), a node involved in the mixing process can not make links between any input or output addresses. The information disclosure is prevented by the usage control system, as it monitors access to the data and prevents dissemination to unauthorized users. The repudiation threat 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.

3.4.5 Security threats and mitigation

The UCS is a critical agent for controlling data usage and data flows between the system agents. It is an attractive target, vulnerable to specific attacks which can be partially mitigated. The proposed countermeasures to the security threats established using the STRIDE model (cf. section 3.4.3) are:

- *Spoofing*(S): legitimate users and the usage control system mutually authenticate, e.g., using SSH or TLS;
- *Tampering*(T): the data processing is monitored by the UCS, excluding illegal modifications to the data. However, an attacker can modify the usage control system's binaries to make it ineffective [Kelbert and Pretschner, 2018]. This threat can be mitigated by digital signatures [Kelbert and Pretschner, 2018];
- *Denial of service*(D): the modern denial of service attacks can be hard to mitigate, but decentralizing the UCS alleviates the risk, as well as mutual authentication of all the infrastructure components using certificates [Kelbert and Pretschner, 2018];

Car renters may damage the car and it is relevant to design a compensatory measure to make sure the car owners do not take disproportionate risks by getting involved in the network. Indeed, the framework provides a fair level of privacy and car renters are encouraged to run away without paying compensation for damaging the car. This is a strong deterrent to the car owners' involvement in the system. As a solution, we introduce a stake that has to be locked by the car renter during a given amount of time in the form of a UCON pre-obligation. This principle is very similar to the proof of stake but is used to make access decisions rather than to achieve consensus. 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 are 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 [Evaldas Drąsutis, 2021]. An alternative is to send the deposit to an address belonging to the usage control

system as long as smart contracts are not available. Although less satisfying, this is a convenient workaround under the trusted UCS assumption.

3.5 Conclusion

In this chapter, we detailed and evaluated a framework to address the requirements of privacy, security and performance of the Internet of Things, while providing a basis for zero-fee micro-transactions (*Objective 1*). The basis of the framework is the IOTA DAG-based distributed ledger. IOTA is complemented by privacy-preserving mechanisms: merge avoidance and decentralized mixing. Finally, we added usage control so that users can monitor the usage of their data, and data providers sell the data generated by their devices according to pre-defined policies, including rules on transactions.

We conducted performance, security and privacy evaluations as part of the *Objective 5* (cf. Section 1.2) to demonstrate the validity of the proposed framework and highlight the security and privacy guarantees it provides. In the next chapter 4, the integration process we used in this chapter to improve performance is detailed. We analyze the necessary conditions to ensure integration is relevant, provide a methodology to integrate usage control before evaluating the outcome of the proposed integration.

Chapter 4

Integration of Usage control with Distributed Ledger Technologies

Contents

As detailed in the problem statement 1.2, one of the motivations of this thesis is to identify the methods to integrate usage control correctly in distributed ledgers, notably for performance issues (*Objective 3*). Indeed, adding the usage control system, or any privacy-enhancing technologies, to an existing distributed ledger creates additional bottlenecks. Firstly, the usage control system itself has performance limitations that sometimes exceed the ledger constraints. Besides, the interactions between the usage control system and the distributed ledger can create network or storage bottlenecks. Finally, the large number of devices in the Internet of Things creates significant

network interactions between devices and the usage control system e.g., between the Policy Enforcement Points and the Policy Decision Point. This chapter focuses on the integration process. First, the existing integration schemes in the literature are introduced (Section 4.1). Then, we focus on identifying the most suitable technologies to integrate, as well as the purpose and methods of integration (Section 4.2). The integration process is evaluated to measure its performance benefits accurately (Section 4.3). A privacy threat assessment is also conducted in the general context of data trading, highlighting the benefits of usage control to preserve privacy (Section 4.4), before concluding (Section 4.5).

4.1 State-of-the-art usage control with distributed ledgers

Integration of usage control with distributed ledgers, often referred to as *incorporation*, has been proposed in the literature to benefit from distributed ledger properties. In particular, private blockchains are often leveraged for the following reasons:

- *auditability* and the possibility to monitor data while controlling data access simultaneously [Khan et al., 2020, Ma et al., 2020];
- *smart contracts* to enforce usage control and process data automatically [Ma et al., 2020, Zhang et al., 2022];
- *performance* with better network response time and low computation requirements [Salimitari et al., 2020] in comparison with public blockchains.

Khan *et al.* [Khan et al., 2020] incorporated the components of the UCON framework into the Hyperledger Composer, to form the BlockU model. The main reason stated by the authors is the continuous monitoring of the data access, thanks to attribute mutability. UCON is incorporated by using the Hyperledger Composer layer, a layer dedicated to modeling applications. Therefore, this incorporation can be seen as peripheral as the UCON components do not contribute to the blockchain network, and performance aspects are ignored. Besides, this work is not focused on IoT use cases.

Ma *et al.* [Ma et al., 2020] proposed a decentralized usage control on both a public and a permissioned blockchain, respectively Ethereum and Hyperledger. A clear focus is given to privacy and decentralization to prevent accidental or intentional data leaks. All data operations are processed directly on the private blockchain with smart contracts, the degree of integration is consequently quite high. The public blockchain is used to introduce a reward mechanism, to encourage users to provide quality usage control data. However, the integration of usage control into the blockchain benefits usage control but not the blockchain network.

Shi *et al.* [Shi et al., 2021] designed a Distributed Usage Control Enforcement (DUCE) solution for the Internet of Things, to tackle privacy issues related to the Cloud-Enabled Internet of Things (CEIoT). DUCE uses distributed PDP and PEP components and leverages private blockchains to store tamper-proof and traceable data on the ledger. The policies are written in the XACML language and then converted by an algorithm into the Solidity language for smart contracts. DUCE relies on a *Trusted Environment Execution* (TEE) to enforce trustworthy processing of the rules.

Zhang *et al.* [Zhang et al., 2022] devised an efficient data trading scheme with usage control for Industrial IoT (IIoT). The scheme is based on Hyperledger Fabric and uses Intel SGX to define protected private regions of memory. Similarly to the abovementioned related works, the authors utilize smart contracts to automatically trade data. To tackle usage control overhead on the data trading system, the authors rely on *Policy Monitors* (PM) on the user side, using the SGX as a TEE. This enables the offloading usage policy enforcement on the user side, addressing scalability issues.

4.2 Integration of usage control with distributed ledgers

In this section, the principles of usage control integration with distributed ledger technologies are discussed. First, we propose criteria to determine if a technology is suitable for integration in Section 4.2.1. Then Section 4.2.2 details the different benefits of integration. Finally, Section 4.2.3 describes the method to integrate usage control system with the distributed ledger technology.

4.2.1 Integration suitability criteria

The distributed ledgers are heterogeneous because of diverse consensus methods, incentives, access control methods or even transaction graphs. To design a relevant integration scheme of usage control, it is first necessary to identify the requirements of this integration to find out the suitable distributed ledger technologies. Therefore, we propose a classification (cf. Table 4.1) relying on three categories of features: the consensus method, the incentive to contribute to the network, and the graph of transactions.

Method of analysis. To determine whether a DLT instance can integrate properly with the usage control system, we consider two properties of distributed ledger features: decentralization and equitability. These properties are directly influenced by other ledger features which are the basis of our analysis: 1) the consensus method; 2) the incentive to contribute to the network; 3) the transaction graph structure. The method of analysis is schematized in Figure 4.1. *Equitability* describes the possibility for every user to have a fair part in the decision-making and consensus processes. Particularly, it includes devices with poor computation or storage capacities. This property

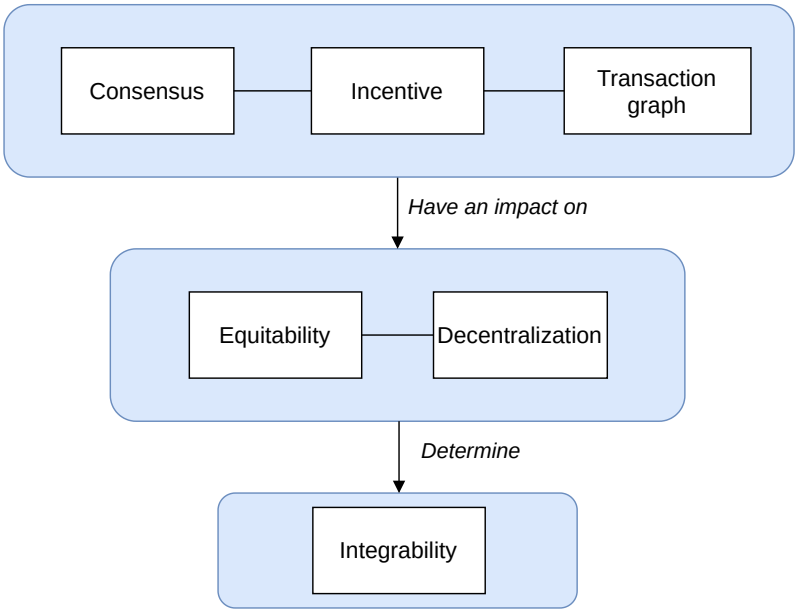


Figure 4.1 – Criteria used for integration suitability - schematized

ensures small devices in the network have an actual impact, and that a small fraction of users do not monopolize the power in the network. *Decentralization*, besides providing performance and security benefits, is an interesting property to assess the integration suitability. Indeed, decentralization partially reflects the above-mentioned equitability. A fully decentralized network is likely to rely on local consensus to make decisions, which gives a bigger impact on participating devices [Popov, 2020, van Steen et al., 2021]. Decentralization can take several forms, all expressing power asymmetries [Bodó et al., 2021]: 1) decentralization in the governance; 2) decentralization in terms of power, where nodes, or more likely node aggregates, have disproportionate power over the rest of the network. For instance, the proof of work consensus method gives power to the most powerful mining pools, while the proof of stake creates another form of centralized capital power. Highly centralized governance is often a prerequisite for a distributed network to work, which means a network can be highly centralized regarding only one centralization aspect, i.e., governance or power, but not the other. The integration is more relevant as equitability and decentralization increase. The distributed ledger features have an impact on these two properties of equitability and decentralization. To determine the characteristics that favor or hinder integration, three categories of features are next analyzed.

Consensus method. Consensus methods are very diverse, but the following are considered in our analysis:

- *Proof of Work*: due to computation concerns, a challenging computation race will

exclude low-power devices from participating in the consensus. Even though the UCS could make a moderate profit while fulfilling its role, it is not an interesting application for our integration project;

- *Proof of Stake*: PoS better fits the IoT computational requirements, but has several drawbacks for integration. First, a device can not contribute to the consensus unless it has a stake. Sensors in particular do not have the motivation or the possibility to hold a cryptocurrency asset, either for the consensus or to make transactions [Raghav et al., 2020]. The proof of stake also tends to centralize the network in the hands of a few users, either the delegates (DPoS) or the richest users. Equitability is therefore not achieved with this consensus method. Integrating the UCS in a distributed ledger based on PoS is not relevant, as devices can not contribute to the network;

- *Proof of Authority*: PoA, contrary to the proof of stake, does not require holding a cryptocurrency asset. Nevertheless, PoA, by delegating the power to a few nodes considered trustworthy, limits the possibility to integrate the UCS components;

- *Proof of Elapsed Time*: in PoET, miners are chosen at random using timers. This consensus method ensures equitability. In this setting, the components of the usage control system could contribute to the consensus, and integration is relevant. However, PoET is limited to private ledgers, as users must first join the network and gain a membership certificate to be allowed to start the timer;

- *Practical Byzantine Fault Tolerance*: all nodes take part in the voting process with equal power, providing the equitability property. Besides, this consensus method is suitable for the IoT but only on private ledgers due to scalability issues in terms of the number of users, as it causes high network overhead [Salimitari et al., 2020].

- *dPBFT*: delegating the consensus process to a smaller subset of nodes centralizes partly the network and reduces equitability, but the delegates are chosen by the nodes. The communication overhead is also reduced.

Relying only on consensus methods, it appears that only the practical byzantine fault tolerance (PBFT and dPBFT) and the proof of elapsed time (PoET) are suitable for integration, but both are used only in private settings. This explains why the related works (cf. Section 4.1) mostly focus on private ledgers when the requirements for the IoT use cases are not considered.

Incentive. Most consensus methods rely on a financial incentive to encourage users to contribute to the network. The proof of work rewards the miner when it adds a block to the ledger, while the proof of stake rewards the users when they stake their cryptocurrencies for network security. However, rewards have negative side effects. When

using a proof of work, it not only encourages miners to contribute to the network but also to group in mining pools. This phenomenon is the cause of the centralization of the network and users can no longer act independently [Ketsdever and Fischer, 2019]. In proof of stake networks, the staking reward incentives create the *Nothing at Stake* issue. When a fork occurs, i.e., when two versions of the ledger are competing, the validators have an interest in maintaining both versions to avoid taking the risk of maintaining the wrong one and earning worthless rewards [Ketsdever and Fischer, 2019].

In the Internet of Things context, where small devices with energy constraints are involved, the contribution to the network may not be conditioned by financial incentives. IoT-oriented projects would rather focus on operational and energy savings, in particular for battery-powered devices. This is the case for IOTA, which does not introduce any kind of financial rewards for operating a node or validating transactions. The usage control system, as a security device, does not contribute to the network for financial rewards but seeks to deliver fast access decisions at the minimum cost.

Ledger type. Considering consensus methods, it appears that private blockchains are more suitable for integration than public blockchains. However, directed acyclic graphs have several properties of interest, among which some significantly ease the integration process. The removal of gateways in directed acyclic graphs enables the users to push their transactions directly to the network. This is also true for the usage control system, which may actively contribute to the network. The overall metrics of directed acyclic graphs enable more users to contribute to the network. The usage control system can make its decisions faster by processing the transactions locally.

Selection of the suitable distributed ledgers. To sum up, the possibility to integrate usage control with distributed ledgers is mostly determined by the equitability of the protocol and its decentralization. Three main criteria have a direct impact on equitability and decentralization: the consensus method, the incentive and the transaction graph.

The consensus method has a direct impact on the possibility for small devices to contribute to the network. The incentive is paramount as well because usual financial incentives tend to centralize the network. Finally, the ledger type, either a blockchain or a directed acyclic graph, has a deep impact on the network features. Since DAGs are meant to allow users to push and check transactions themselves, they enable the contribution of the usage control system to the network.

According to this classification, summarized in Table 4.1, it is possible to depict two categories of fitting technologies, both without a financial incentive: directed acyclic graphs and private blockchains. However, only directed acyclic graphs consider the large-scale IoT requirements, while private blockchains are not scalable and do not have cryptocurrencies. Consequently, *we will consider directed acyclic graphs* for the integration of usage control in the following.

Parameter	Instance	IoT Suitable	Equitability	Decentralized*	Integration	Notes
Consensus	PoW	✗	✗	✗(.pow)	✗	Compute-intensive
	PoS	✗	✗	✗(.pow)	✗	Stake needed
	PoA	✗	✗	✗	✗	Similar to PoS
	PBFT	✓	✓	✓	✓	Private blockchains only
	dPBFT	✓	✓	✓	✓	Less equitable, low communication overhead
	PoET	✓	✓	✗(.gov)	✓	Private blockchains, relies on Intel SGX
Incentive	Financial	✗	✗	✗	✗	Deterrence for small devices
	Savings	✓	✓	?	✓	Incentive for small devices
Ledger	Private blockchain	✓	✓	✗(.gov)	✓	Promising but scalability, currency issues
	Public blockchain	?	?	?	?	Very diverse, not determining
	Directed acyclic	✓	✓	✓	✓	Significantly favors integration

Table 4.1 – DLT parameters and their impact on integration. Question marks (?) mean the parameter is not determining.
*Decentralized can take several forms, governance (.gov) or power asymmetries (.pow)

4.2.2 Integration benefits

A logical way to integrate the usage control system with a distributed ledger based on a DAG is to run a node. Nodes are indeed critical components of distributed ledgers. They differ depending on the technology, but their purpose is at least to check the validity of transactions. The usage control system could run a node with the following expected benefits for itself:

- *disintermediation*: running a node avoids relying on a third party. The bandwidth is secured for the node transactions without delay. Usage control with policies based on transactions is faster, based on the local ledger analysis;
- *storage control*: the node may keep all transaction records to enforce the policies if necessary. Nodes usually rely on local snapshots to reduce the size of the local ledger;
- *node configuration*: the node can be configured to adjust both security and performance parameters to the UCS needs;
- *network security*: the UCS will contribute to the ledger verification, reducing the probability of failure resulting from a low number of nodes [Khacef et al., 2021]. Besides, a higher number of nodes makes some attacks harder, such as the 51% attack [Aponte-Novoa et al., 2021];
- *throughput increase*: the UCS increases the throughput as it pushes transactions on the network, due to DAG properties;

To fulfill its mission, the usage control system has to monitor system and network calls, which is an intrusive process. *Transparency* and *auditability* are paramount in this context. Transparency in usage control can be considered as the fact that the usage control operations, e.g., allowing access or preventing dissemination, are communicated to others, while auditability anticipates the storage of these usage control data for a future audit. Transparency and auditability can be achieved by writing usage control data on the ledger such as the operations performed by the UCS on the users. Auditability incites the usage control system to behave correctly, as any misbehavior will be recorded publicly on the ledger.

4.2.3 Integration methodology

The integration methodology concerns two aspects: the global system architecture, considering how the usage control system can integrate the distributed network, and data protection, as data written on the ledger becomes visible and exposed to privacy risks.

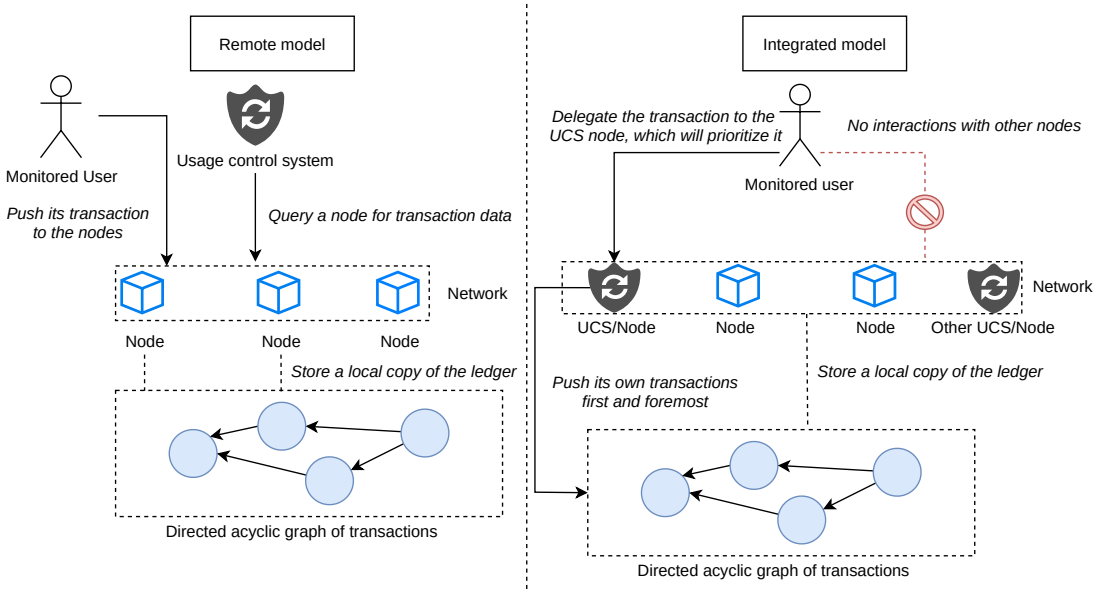


Figure 4.2 – Differences between the remote model and the integrated model when using directed acyclic graphs.

Integration as a Node. Directed acyclic graphs, by design, alleviate the requirements on nodes to maximize the number of devices contributing to the network. Therefore, the usage control system itself can deploy a node and push transactions itself. By deploying a node, the usage control system will 1) push transactions including some related to usage control use cases, accelerating the access decision process; 2) prioritize its transactions on the network, avoiding a potential queue; 3) store a local copy of the ledger, to process the ledger faster when necessary for its access decision. The integration model is represented in Figure 4.2, showing the links between the UCS, the nodes, and the monitored users. Data can be partially written on the public ledger as long as they do not create inference risks, as described in the following.

Data management. For public ledgers, one key motivation for integration is the capacity to write immutable data on the ledger for transparency purposes (cf. Section 4.2.2). Data must be classified to determine whether they can be displayed publicly or not. Data can be of four types, summarized in Table 4.2:

- *protected data*: data whose access is monitored by the UCS;
- *usage control data*: data concerning the usage control, including the processes performed by the UCS as well as the results of the evaluation;
- *users' data*: data needed by the UCS about the users, such as their attributes to make access decisions;

- *metadata*: data about the other data, e.g., data that states the kind of processed attributes, but not the content of the attributes.

Protected data must not be stored in clear text on the ledger, and can either be encrypted on the public distributed ledger or stored in a distributed database. However, encryption produces computational overhead and the management of encryption keys is a real issue in large-scale contexts; that is why we resort to using a database management system. Usage control data describe the operations performed by the usage control system on the users. They are composed of a data identifier, the pseudonyms of the users and the action performed. They are published on the public ledger for transparency and auditability purposes. Data about users, such as their attributes, are needed only by the usage control system and are stored in the database. Finally, metadata is published on the ledger, such as timestamps and data identifiers to keep track of the data. Metadata can pose a *detectability threat* when revealing its existence can lead to privacy issues, even without providing the actual content of the data, e.g., knowing the existence of a police record associated with an identity, without the content inside the record, is a sensitive information. Metadata must therefore be processed carefully, using a privacy threat analysis framework such as LINDDUN [Deng et al., 2011].

Data type	Data storage
Protected	Database
Users' data	Database
Usage control	Directed acyclic graph
Metadata	Directed acyclic graph*

Table 4.2 – Data types and their respective storage area
* If detectability is not an issue

4.3 Performance evaluation

In this section, we detail the results of a performance evaluation conducted on IOTA to demonstrate that the integration is efficient and to assess its estimated performance outcomes. To this end, we will measure the time needed for the UCS to make an access decision in the integrated setting compared to the remote node setting.

4.3.1 Testbed

Testing environment. To assess our contribution, we test our integration using the IOTA technology (cf. Section 3.1). Since data are partially stored off-chain (cf. Section 4.2.3), we relied on a *Cassandra* distributed NoSQL database as decentralized stor-

age. Cassandra is *horizontally scalable*, meaning it easily handles increasing traffic demands by adding more machines [Silva and Lima, 2021]. Cassandra also works on low-power clusters, which is particularly suitable for the Internet of Things [Silva and Lima, 2021]. The IOTA node relies on *Hornet*, a community-driven IOTA node software written in the *Go* language. The usage control system is written in Java. The usage control policies are defined by the users and written using the XACML language [Godik and Moses, 2003]. An example of an XACML policy is provided in the appendix (Figure D.1). During the tests, policies are not specified by users, but automatically derived for convenience.

Network selection. IOTA has a public development network called *devnet* for testing. The devnet has free tokens and is meant for testing. The public devnet could be convenient because its network is already deployed and designed to conduct tests, in particular, to measure resistance in high-load scenarios. However, it has significant drawbacks for testing. First, the network is subject to complete overhauls with no backward compatibility, preventing the reproduction of the tests conducted in our experiments on the same network. In particular, the introduction of IOTA 2.0 will lead to the removal of the IOTA 1.0 testing network. Second, the network is public, it is not possible to control the network topology including the number of nodes.

Consequently, the tests on the public network are not sufficient to ensure the efficiency of integration. To ensure our tests can be reproduced, we deploy a *private Tangle* instead. This methodology has been used by Dong *et al.* [Dong et al., 2019] to benchmark different DAGs including IOTA. Each node is deployed on AWS instances and constitutes a part of a private IOTA network. The network architecture of the private Tangle with AWS instances is given in Figure 4.3, for 5 nodes.

Nodes settings. The tests are conducted using AWS t2.micro instances with 8 Gio[1] storage capacity, 1 vCPU and 1 Gio RAM, consistent with the Internet of Things constraints. Another computer is used to run the usage control system, with better storage and computational power: 8Go of RAM memory, 32Go of storage and 4 CPUs. The nodes are located in Amazon's default US East (North Virginia) zone. Each Hornet node *when running on the private Tangle* uses the default spammer, spamming 5 messages by second for each node. Spamming is a desired behavior, as it theoretically speeds up the transaction validation in IOTA (cf. Section 3.1).

4.3.2 Methodology

System model. The agents of the system can be summarized as follows. First, the *data providers* that sell data collected from their devices. Then the *data buyers* pay a certain amount of cryptocurrency to be granted access to these protected data. The *usage control*

[1] 1 Gio=1024 Mo

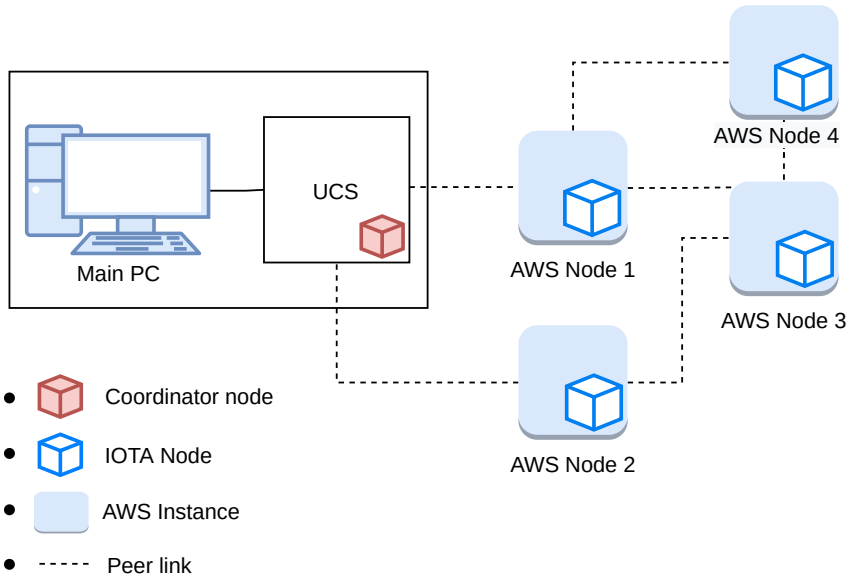


Figure 4.3 – Private Tangle architecture with AWS instances - 5 nodes. Each instance runs an IOTA node, and a PC runs the Coordinator node to orchestrate the network.

system (UCS) is responsible for monitoring the data access rights and for preventing dissemination, based on data buyers' attributes and actions. In particular, it processes the transactions on the *distributed ledger* to grant access if the payments are duly performed. A Cassandra *distributed database* shared between the data providers stores the protected data. *Network nodes* validate the data buyers' transactions and propagate them to the other network nodes. The system model is depicted in Figure 4.4.

Workflow. To determine which calls are of interest to assess performance outcomes of integration, the system and network calls needed for granting or denying access are detailed and classified into two categories according to the intensity of computations required, as depicted in the sequence diagram of Figure 4.5. Either a call is *computationally-intensive* (1) or *lightweight* (2). This notion is comparative, such that lightweight calls are not time-consuming compared to computationally-intensive calls. The workflow following an access request is as follows. First, the data buyer requires access to data by sending an `accessRequest` along with its user identifier. Then it sends a `transaction` to an IOTA node to perform the payment. This node checks if the transaction is correct, e.g., no double spending, by checking it against the local state of the ledger. If the transaction is indeed correct, the node computes a light proof of work only to prevent spam. The node then forwards the transaction to the rest of the network, using the `push` call. The UCS then begins the monitoring of the access and sends a `requestPolicy` to its policy store, fetching the XACML policy specified by the data

Number of nodes	Setting	`build` and `push`	`fetchTransaction`	Total decrease
3 nodes	Integrated Remote	65±3 ms 1020±21 ms	✗ 85 ms	94.12 %
5 nodes	Integrated Remote	61±3 ms 868±38 ms	✗ 84 ms	93.59 %
7 nodes	Integrated Remote	66±3 ms 773±16 ms	✗ 86 ms	92.31 %
10 nodes	Integrated Remote	58±3 ms 845 ms±32*ms*	✗ 86 ms	93.77 %

Table 4.3 – Measures of transaction time (averages) for each configuration with different networks sizes

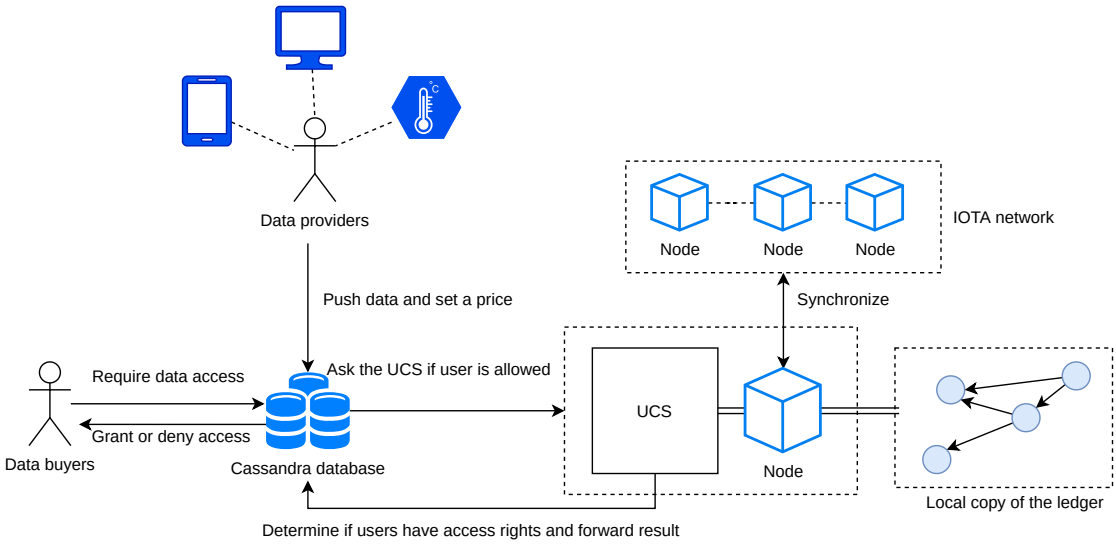


Figure 4.4 – Interactions within the system model during the pre-access phase

provider. The UCS asks for attribute values of the data buyer (`requireAttributes`) to be able to check policy compliance. The attribute values are used for data access control, but also include values resulting from the analysis of network calls to monitor the information flow. When the UCS receives the attributes from the data buyer (`sendAttributes`), it fetches the transaction status on the network by calling `fetchTransaction`, then determines the policy compliance (`checkPolicy`) to make an access decision. If the user is authorized to access the data, the UCS sends a `grantAccess` call to the Cassandra data provider database and notifies the data buyer with a `notifyAccessResult`. Following the `grantAccess` call, the data provider sends the requested data to the data buyer with the `sendData` call. The data buyer is monitored by the UCS for the ongoing obligations and ongoing conditions (cf. Section 2.1.1) with `ongoingMon-`

itoring calls. For convenience, only one call is represented in the sequence diagram (Figure 4.5), but the monitoring usually requires numerous calls due to the continuous nature of the monitoring. The monitored data buyer must reply to `ongoingMonitoring` with `sendOngoingData` so that the UCS can make its access decisions and interrupt access if the policy is violated. The UCS writes usage control logs and metadata for transparency on the ledger while considering privacy threats (cf. Section 4.2.3 and Table 4.2) by calling `writeAccessLogs`. The access is finally revoked by the UCS with `revokeAccess`, should the data user request it or contravene the data provider's policy.

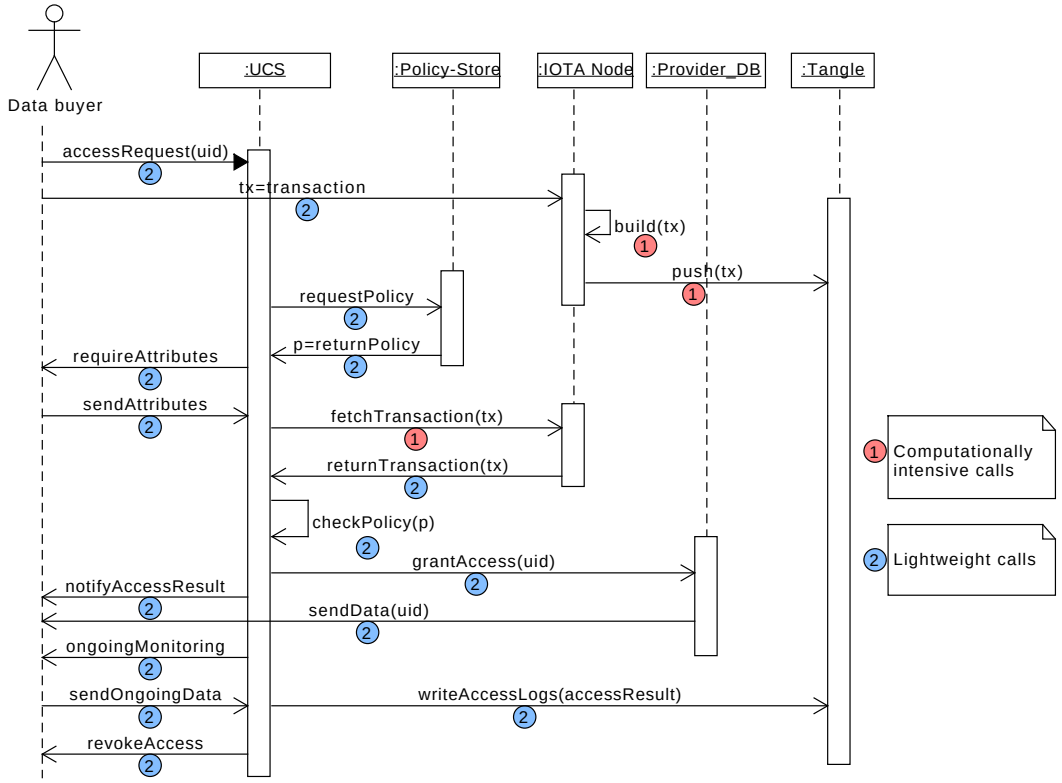


Figure 4.5 – Sequence diagram for pre-access and ongoing access - UCS not integrated with IOTA

Measured calls. Only the `build`, `push` and `fetchTransaction` calls will be measured as the only computationally intensive calls. They are referred to as *transaction time*. Our first guess would be to also include the `checkPolicy` calls; however first tests showed that a policy with 1000 rules takes an average 7 ms to be evaluated (1500 samples), which is lightweight compared to the transaction time. Other calls are also lightweight when measured. The three calls `build`, `push` and `fetchTransaction` are the

most time-consuming for the reasons detailed next. To build and push a transaction, the node must compute a proof of work and check that the transaction is consistent with its known ledger state. The *fetchTransaction* call is time-consuming due to several varying parameters: 1) the node is processing other transactions which delays the `fetchTransaction` call by adding it to its processing queue; 2) the (physical) distance between the node and the UCS without integration, which creates network latency; 3) the time needed to query the ledger of the node, which increases with the number of simultaneous requests.

Transaction time. To demonstrate actual performance improvements, we measure the time needed for a transaction to be validated and pushed to the network, and the time required to fetch the transaction from an IOTA node. These operations are respectively the calls `build`, `push` and `fetchTransaction` of the sequence diagram of Figure 4.5. The `build` and `push` calls are grouped under a single API call in the Java IOTA library and correspond to only one common measure. Tests are conducted in the two different configurations, respectively the remote and the integrated settings: (1) the UCS interacts with a remote IOTA node running in an AWS instance; (2) an integrated node that runs both the UCS and an IOTA node concurrently, as illustrated in Figure 4.2. For each test, 1500 samples ($N = 1500$) are used and confidence intervals are given with a 95% confidence level.

4.3.3 Results

The results of experimental measurements are summarized in Table 4.3. For convenience, only the experiments with 5 nodes $N = 5$ are first discussed. We then detail the impact of the number of nodes on the results and provide further explanations.

Remote setting. When the UCS interacts with a remote node, the `build` and `push` calls need an average $\bar{t}_{transaction,r} = 868 \pm 31ms$ with 5 nodes in the network. Building and pushing a transaction requires a minimum $m_r = 646ms$ and a maximum $M_r = 7649ms$. This wide range is caused by nodes which are often desynchronized and have to update the state of the ledger before pushing their transactions. Additionally, the UCS has to recover the result of the transaction requiring an additional $\bar{t}_{fetch,r} = 84 \pm 0.38ms$.

Integrated setting. When using an integrated node on the UCS device, the transaction time improvements are significant. Indeed, the `fetchTransaction` call does not involve any network calls and requires less than 1 ms to be completed when measured. Our own transactions are prioritized on our node, substantially decreasing the time to complete the `build` call. The average time needed to `build` and `push` a transaction drops to $\bar{t}_{transaction,l} = 61 \pm 2.86ms$.

Number of nodes. As the number of nodes increases, the impact of integration in percentage remains steady, varying from a 92.31% to a 94.12% transaction time de-

crease. The transaction time in our integration scheme is consequently 13 to 18 times faster compared to the remote node setting.

The time required to build and push transactions decreases before 7 nodes, then starts to go up when using 10 nodes. Firstly, the number of messages spammed increases with the number of nodes. We can therefore observe the expected behavior of IOTA [Popov, 2017]: as the number of transactions per second increases, transactions are validated faster. Secondly, when reaching 10 nodes, there are approximately 50 messages spammed per second as each node spams 5 messages per second. There are two possible explanations for this behavior. First, the network could saturate due to the Coordinator milestones validation. This issue is well-identified by both the IOTA foundation and academics [Conti et al., 2022, Wang et al., 2022b] and is the motivation for the development of IOTA 2.0, willing to remove the Coordinator. However, the Coordinator runs on the same device in both integrated and remote settings, but the transaction time does not increase for 10 nodes in the integrated setting. Therefore, we can conclude that it is not the Coordinator that saturates, but the remote node used for the testing. Indeed, the nodes running on AWS are resource-constrained, and processing 50 transactions per second is time-consuming for an instance with 1 Gio of RAM and 1 vCPU.

4.4 Privacy evaluation

In this section, we conduct a privacy threat analysis, to highlight the risks for the users either selling or buying data in the system as described in Section 4.3.2. Based on this analysis, we then detail how usage control mitigates the different privacy risks.

4.4.1 Threat model

The *usage control system* is considered *honest-but-curious*. It fulfills its usage control tasks but may be interested in collecting undue data about the users it monitors. This behavior may be financially motivated, i.e., to sell the users' data afterward. In particular, the UCS has an interest in gathering the system calls and network calls of the monitored users, both carrying valuable, privacy-sensitive data.

Data buyers and *data providers* are considered honest-but-curious as well. Notably, data buyers try to infer more data from the protected data they buy, but do not try to bypass the UCS monitoring or to compromise the database. The *Cassandra distributed database* is considered trusted and not compromised.

4.4.2 Privacy risks

The *LINDDUN privacy threat modeling framework* [Deng et al., 2011] (cf. Section 2.3.1) is used to describe the different privacy risks faced by data buyers and data renters. **Items of interest.** Items of interest (IOI) in LINDDUN refer to any data element, including users' personal data or transaction data, that is considered privacy-sensitive. It is paramount to exhaustively identify which items are of interest before conducting the privacy threat assessment, to ensure the full listing of the threats. Items of interest can be subjects, messages, actions or data. In scenarios involving transactions, the items of interest can be:

- the geolocated protected data generated by data providers;
- the transaction data, including but not limited to users' addresses and transaction values;
- users' data, i.e., data buyer and data provider personal data;
- usage control data, such as the results of access requests;
- metadata, such as the time the data was created or added to the Cassandra database, that can be used to infer other data.
- data buyers, data providers, the UCS and the Cassandra database, i.e, subjects;
- network messages between the subjects, and the messages between the UCS components (cf. Section 2.1.2);
- actions carried out by the subjects, such as an access request by the data buyer, entry insertion on the database by the data provider or the beginning of an access request evaluation by the UCS.

In the following, transaction data are ignored, as preserving privacy in distributed ledgers is a specific, orthogonal research topic. Legal compliance in particular is troublesome for distributed ledgers [Haque et al., 2021], and privacy-enhancing technologies have been designed to address privacy threats, such as cryptocurrency mixers for linkability [Sarfraz et al., 2019], [Glaeser et al., 2022].

Linking (L). Linking occurs when two *items of interest* are associated to learn more about an individual or a group. Due to the diversity of agents and technologies involved, linking threats are numerous in the electricity consumption prediction scenario:

1. linkage between protected data and its owner (user data). The location in the protected data can be used to facilitate the re-identification of the data providers (I);

2. linkage between a data buyer and a data provider. It simplifies re-identification when one of them is identified (I);

3. linkage between metadata and usage control data, as both are written on the ledger.

Linking leads to other identifying (I), detection (D) and non-repudiation (N) threats and is triggered by data disclosure (D) - the more data available, the more likely are linking threats.

Identifying (I). Identification is occurring when an attacker learns the identity of an individual, breaking its anonymity or pseudonymity. This threat distinguishes *identified data* where the identity is explicitly maintained, and *identifiable data* which enables to derive the identity indirectly. Users' attributes, e.g., IP address or processed by the usage control system to make an access decision can be used to infer the user's true identity. In the scenario, the risks are to re-identify data buyers and data providers, using protected data, usage control data or metadata.

Non-repudiation (N). Non-repudiation, in LINDDUN privacy threat assessment, is the ability to attribute a claim to an individual. For example, the impossibility for a data buyer to deny they bought and accessed protected data, or for data providers to deny they generated a given data, are non-repudiation threats.

Detecting (D). Detecting is the ability to deduce the involvement of an individual in an action with observation. Detecting does not require being able to read the actual data. Knowing that the data exists is enough to infer more sensitive information. Detecting can be done by observing *communications* or *application side-effects*. Detecting threats in the electricity consumption prediction scenario includes:

1. detecting that a user is monitored by analyzing the communications of the UCS. An attacker learns that the user has likely bought valuable data;

2. detecting what are the protected data and where they are disseminated, without the actual content of the data, by analyzing the UCS logs;

3. detecting users' attributes by intercepting the communications between the policy information points (PIP) and the context handler (CH) (cf. Section 2.1.2);

Data disclosure (D). Data disclosure is the immoderate collecting, storing, processing or sharing of personal data. This generic threat focuses on four characteristics:

1. *unnecessary data types*, if the data granularity or sensitivity is too detailed;

2. *excessive data volume*, if the amount or the frequency of data processing is too high;

3. *unnecessary processing*, if the data is processed or disseminated out of necessity;

4. *excessive exposure*, refers to how widely accessible the data is and to whom the data is shared.

In the energy consumption prediction scenario, data disclosure can occur if a data provider has access to protected data while not fulfilling the policy's conditions, or can disseminate it outside of the UCS monitoring perimeter. Similarly, data disclosure occurs if the UCS collects too detailed data about the data buyers to monitor them.

Unawareness (U). Unawareness occurs when an individual is insufficiently informed and involved in the processing of personal data. Unawareness occurs in the scenario if the data providers are not informed about the privacy threats they face by selling their data with associated geolocation. The user may also be unaware of the privacy risks of accepting to be monitored by the UCS, which writes usage control data on the ledger, that are potentially privacy-sensitive (cf. Section 4.3.2)

Non-compliance (N). Non-compliance is the deviation from security and data management best practices, standards and legislation. This risk occurs if the processing of any item of interest is considered unlawful based on the specified policies and the applicable regulations, e.g., GDPR [The European Parliament and the Council of the European Union, 2018]. In the scenario, the risks are that although the data provider specifies a policy, it is not properly enforced by the usage control system.

Threat	Scenario example	Mitigated by the UCS
Linking	Link between data owner and buyer	?
Identifying	Data provider is re-identified	?
Non-repudiation	Data provider can not deny data generation	✗
Detecting	Detect a user is monitored	✓
Data disclosure	Protected data dissemination	✓
Unawareness	Inference risks of location unknown	✓
Non-compliance	Data provider policy not enforced	✓

Table 4.4 – LINDDUN privacy threat analysis, based on the illustrative scenario.
(?) marks means the threat is only partly mitigated by the UCS.

4.4.3 Threat mitigation with usage control

As a privacy-enhancing technology, usage control is designed to address a significant part of the above-mentioned privacy threats. We next detail how these threats are addressed, and which ones can not be mitigated by the UCS. The results are summarized in Table 4.4.

Linkability. (?) The usage control system has only a partial impact in preventing this threat. Notably, the linkage between metadata and usage control data, which are both unrestricted data available publicly on the ledger, can be accessed without

monitoring in the first place. However, the UCS monitors the dissemination and the processing of the protected data, limiting the risks to link them to their owners.

Identifying. (?) Similarly, the usage control system must be able to identify the data buyer to fulfill its mission. While it can prevent users from disclosing *identified data*, it is much harder for it to prevent inference from *identifiable data*.

Non-repudiation. ✗ To fulfill its task, the UCS needs to ensure that a monitored data buyer can not decline having disseminated the data, or having processed it in an unlawful manner. Therefore, the UCS not only does not guarantee non-repudiation, but also writes usage control data on the distributed ledger, making non-repudiation impossible.

Detecting. ✓ To mitigate this threat, all communications are secured using TLS, notably the communication between the usage control system and the users, the context handler and the PIPs, as well as the context handler and the PEPs. Besides, unless the UCS is compromised, an attacker does not have access to the logs of the UCS.

Data disclosure. ✓ The data processing is monitored by the usage control system, mitigating the *excessive data volume* threat. The usage control system, by monitoring both the information flow and the usage of the data, also prevents *unnecessary processing*. *Excessive exposure* is prevented as part of the access control to the data.

Unawareness. ✓ The usage control system asks data owners to design data policies themselves, directly addressing the unawareness threat.

Non-compliance. ✓ The usage control system monitors the data buyers, stopping them from processing the data unlawfully. The usage control system, considered honest-but-curious, enforces the proper usage control policy specified by the data provider.

Usage control addresses four categories of threats directly, i.e., detecting, data disclosure, unawareness and non-compliance. Due to the diversity of the data and agents considered, linking and identifying threats are only partially mitigated. Only the non-repudiation threat is not considered by usage control.

4.5 Conclusion

In this chapter, we introduced and evaluated the integration of usage control with distributed ledger technologies. After identifying the possible compatible DLTs (Section 4.2.1), we developed a proof of concept using IOTA before conducting performance tests (Section 4.3). Even though similar works exist in the state-of-the-art that propose the integration (also referred to as incorporation) of usage control with blockchains, the integration is always conducted in private blockchain settings. The work in this thesis therefore comes as a novelty, by studying the integration process itself, and by identifying DAG-based distributed ledgers as potential permissionless solutions for large-scale

IoT networks.

While this chapter focused on usage control, the integration principle can be generalized to any system that requires to process distributed ledger transactions and that can be deployed on a node. The network security always benefits from having more nodes (cf. Section 4.2.2). For example, the mixing services can benefit from integration (cf. Section 2.3.3), especially when decentralized as used in Chapter 3 as the mixing peers can be numerous. Similarly, a device will benefit from deploying a node if it has to analyze the transaction, which is the case for the UCS so as to monitor access. In the next Chapter 5, we will digress from practical optimization aspects and discuss modeling. We will see the limitations of the current usage control formalism and how it can be better adapted to the Internet of Things.

Chapter 5

Modeling Distributed Data Usage Control

Contents

5.1 Introduction

Data usage control enables data owners to enforce policies for their data, by defining authorizations, but also obligations, which are actions to be performed before, during or after being granted access, and conditions bearing on the system and environment attributes such as the time or the system load. Information flow control (IFC) is a security

mechanism that regulates the movement of data within a system or network, to prevent undue dissemination of the data [Denning, 1976]. A decentralized version called *Decentralized Information Flow Control* (DIFC), was introduced by Myers and Liskov in 1997. Contrary to traditional IFC, it enables users to define policies individually or as a group without relying on a central entity. Data owners add security labels to the data directly, to indicate who is allowed to read the data [Myers and Liskov, 1997]. While usage control and information flow control are two different technologies, they are used jointly by modern usage control systems (UCS) [Harvan and Pretschner, 2009]. Therefore, recent models [Kelbert and Pretschner, 2018, Fromm, 2020] tend to encompass both aspects as usage control needs to prevent dissemination in areas outside of the monitoring scope. Introducing usage control can therefore lead to unclear statements, as it is hard to distinguish whether usage control refers to classic usage control, i.e., UCON and derived models, or classic usage control with information flow control. For disambiguation, the following terminology is used in this chapter:

- *data usage control* (UCON): refers to the policies which define the users' rights to access data, based on authorizations, obligations, and conditions;
- *information flow control* (IFC): refers to the policies about the movement of data within the system, to monitor dissemination;
- *usage control*: refers to both data usage control and information flow control. It is the general term, corresponding to modern usage control;
- *a usage control system* (UCS): an entity, centralized or distributed, that is responsible for enforcing usage control.

A formal model of data usage control can help ensure that the system's security and privacy goals are achieved by providing a clear specification of the access control policies, the resources to be protected, and the roles and responsibilities of the participants in the system. While this modeling has been abundantly addressed in the state-of-the-art in centralized settings [Pretschner et al., 2011, Kelbert and Pretschner, 2013, Kelbert and Pretschner, 2014, Fromm, 2020], it is still incomplete in distributed systems, considering only the distribution of the usage control system (UCS) components, but not of the other system users, notably the policy makers and the data readers [Kelbert and Pretschner, 2015, Kelbert and Pretschner, 2018].

Kelbert and Pretschner have proposed a formal model for usage control in distributed systems in 2018 [Kelbert and Pretschner, 2018] that constitutes the state-of-the-art. It focuses on the distribution of the UCS components, to enable local policy evaluation. In this chapter, we propose to add the DIFC model [Myers and Liskov, 1997] to this state-of-the-art formalism. DIFC can be used in contexts where data are

derived from several sources and all sources must agree to grant or deny access to the data [Myers and Liskov, 1997]. It further distributes the system by enabling the definition of policies on data owned by several owners. To this end, we propose the following contributions:

- the definition of data usage control and information flow control policies in a distributed fashion, further enabling a better consideration of the distribution dimension;
- fine-grained information flow tracking with better granularity as DIFC harnesses application-layer semantics [Liu et al., 2022], which is a known limit of the state-of-the-art [Kelbert and Pretschner, 2018];
- the enrichment of the state-of-the-art formalism for distributed systems [Kelbert and Pretschner, 2018], introducing novel functions and dedicated DIFC components to design and manage decentralized policy making.

The integration of DIFC is made possible by recent advances in coding practices, highlighted by Liu *et al.* [Liu et al., 2022]. Past approaches to DIFC have required dedicated instrumentation efforts or developer support, which caused little adoption. Nowadays, DIFC can be leveraged by relying on application event logging, a best practice in software development used for telemetry[1] and debugging [Liu et al., 2022].

The chapter is structured as follows. First, decentralized information flow control elementaries and the state-of-the-art are provided (Section 5.2), before introducing the current modeling of usage control for distributed systems (Section 5.3). We then detail the extension of the existing modeling as our contribution (Section 5.4), before concluding (Section 5.5).

5.2 Background on decentralized information flow control

In this section, after the introduction of the DIFC model (Section 5.2.1), we detail the reasons why decentralized information flow control, while being a canonical approach to access control in systems, was not actually implemented and largely adopted (Section 5.2.2). Recent advances in software development enable the use of DIFC without additional development costs, which is explained in Section 5.2.3

5.2.1 DIFC model

In addition to information flow control, Myers and Liskov [Myers and Liskov, 1997] proposed a decentralized version of information flow control (DIFC). The main moti-

[1]Telemetry is the *in situ* collection of measurements or other data at remote points and their automatic transmission to receiving monitoring equipment

vation behind DIFC is to provide a security mechanism for distributed systems. The DIFC model allows users to declassify information in a decentralized way, by delegating security to users and groups rather than a monolithic organization or entity, such as the usage control system. In DIFC, the users in the system assign *labels* to data, and then the system enforces access control policies based on these labels. DIFC labels are defined as pairs of owner-readers p: $\{O : R\}$, where owner O allows the set of readers R to read the information on which the label is attached. The labels represent the sensitivity or confidentiality of the data and the clearance level of the users. The labels are attached to the data by users and are propagated through the system as the information flows from one component to another. Myers and Liskov [Myers and Liskov, 1997] also introduce the following objects, independently of the existing information flow control modeling:

- *values* that are used in computations;
- *slots* that are storage locations for values that can take different forms like variables in a code as well as a memory space in the hard drive;
- *channels* which can be either *input channels* to read data or *output channels* to write data. Input channels are read-only slots that allow information to enter the system monitored by DIFC mechanisms, while output channels are write-only slots that serve as an information sink to transmit data outside the system;

In DIFC, labels can be attached to values, but also to slots (and channels which are a particular type of slots).

Operations on labels. In DIFC, the label on a value cannot change, but a new copy of the value can be created with a new label. [Myers and Liskov, 1997] When this happens we say the value is *relabeled* even though only the copy has a new label. The key to secure flow is to ensure that any relabeling is consistent with the security policies of the original labeling. Only values can be relabeled, not slots and channels. There are two types of relabeling. *Restrictions* are relabeling allowing fewer accesses than the original label. *Declassifications* conversely add more readers to the label. If a label attached to a value can not change, it is possible to have *multiple labels* corresponding to different owners.

5.2.2 DIFC implementation challenges

While decentralized flow control enables to make data policies whose ownership is shared by several owners, it has several shortcomings which severely limited its adoption in real systems [Liu et al., 2022]. A major factor limiting the use of DIFC is the tremendous cost in terms of development. DIFC implementations assume that programs are modified to be able to add, remove and handle labels (Cf. Section 5.2.1).

Besides, DIFC implementations can operate at different levels of abstraction like IFC implementations, e.g., operating system level or algorithmic level. DIFC implementations relying on run-time mechanisms (cf. Section 2.1.3) are employed at the operating system and programming language levels, which requires additional modifications at this level, making DIFC even more cumbersome [Liu et al., 2022].

5.2.3 Advances in DIFC

The above-mentioned limitations to DIFC adoption (cf. Section 5.2.2) have been addressed indirectly by evolving best coding practices. Indeed, applications now tend to have a security context in the form of *application event logs*, for debugging, fault detection and telemetry. Events are at the root of information flow control and can be used by DIFC at the application system layer, without requiring additional development costs. Liu *et al.* [Liu et al., 2022] consequently presented their solution *T-DIFC*[2] to define DIFC policies based on the inspection of application logging, i.e., using the application layer. Verifying policies at the application level spares the administrator from the need for additional development.

5.3 Existing usage control modeling

The following sections introduce the state-of-the-art on data usage control modeling in distributed systems. First, the data usage control formalism as designed for centralized settings (Section 5.3.1, Section 5.3.2) is detailed, before the specific aspects of distribution (Section 5.3.3). We finally detail the rationale behind the need to improve the existing model, to meet IoT requirements regarding the connection status of the network and to integrate DIFC concepts and functions (Section 5.3.4)

5.3.1 Information flow control model

An information flow control model is defined by a tuple $(\mathcal{D}, \mathcal{C}, \mathcal{E}, \mathcal{I}, \mathcal{F}, \Sigma)$ [Kelbert and Pretschner, 2018, Fromm, 2020]:

- $\mathcal{D}$ are all the *data* to be protected in the system. Data are the equivalent of value in DIFC modeling;
- $\mathcal{C}$ are all representations where the data can be stored in the system and are therefore named *containers*. For example, containers can represent variables, computer memory cells or a blockchain block. Containers are the equivalent of slots in DIFC modeling;

[2] T-DIFC stands for transparent decentralized information flow control

- $\mathcal{E}$ are *events*, e.g., method invocations or system calls, that may cause a flow of data and change the system state;
- $\mathcal{I}$ are the *principals*, all the active entities in a system, i.e., a process or a thread;
- $\mathcal{F}$ is the set of all naming identifiers that are used to identify containers in a system, such as the process-id;
- Σ are all the possible *data flow states*;

For any set S, $\mathcal{P}(S)$ will denote all the possible subsets of S. For example, if $\mathcal{E} = \{e_1, e_2\}$ is a set of events, $\mathcal{P}(\mathcal{E}) \stackrel{def}{=} \{\emptyset, e_1, e_2, \{e_1, e_2\}\}$ is composed of all the possible subsets of $\mathcal{E}$.

System runs are modeled as a set of timed *traces* $\mathcal{T} : \mathbb{N} \rightarrow \mathcal{P}(\mathcal{E}))$ that maps abstract time points t to a set of events $\mathcal{E}$.

An *event* $e \in \mathcal{E}$ itself is composed of a name e.name $\in \mathcal{N}_\mathcal{E}$ ($\mathcal{N}_\mathcal{E}$ is the set of event names) and a set of parameters e.p $\in \mathcal{J}_\mathcal{E} \subseteq \mathcal{P}(\mathcal{J})$, $\mathcal{J}$ being the set of all possible parameters. Each parameter $j \in \mathcal{J}$ is in turn defined by a name j.name $\in \mathcal{N}_\mathcal{P}$ and a value j.value $\in \mathcal{V}_\mathcal{P}$.

Each event has two special parameters, called (e.obj, e.actual) $\in {\mathcal{J}_\mathcal{E}}^2$. e.obj denotes the primary object of e, such as a file. The parameter e.actual is a boolean stating if the event has already happened if true or is only intended if false[3]. Besides, a parameter e.time $\in \mathcal{J}_\mathcal{E}$ indicates the event's time of observation. Note that e.time is not a special parameter, as an event may not have been observed yet if it is intended (e.obj and e.actual are always defined).

We say that e_1 refines e_2 if their names are the same and if the parameters of e_1 are a superset of e_2. An event refinement is detected using the operation *refines* $\subseteq \mathcal{E} \times \mathcal{E}$:

$$\forall e_1, e_2 \in \mathcal{E} : e_1 \; refines \; e_2 \stackrel{\text{def}}{\Longleftrightarrow} e_1\text{.name} = e_2\text{.name} \wedge e_1\text{.p} \supseteq e_2\text{.p}$$

where $e_1.name$, $e_2.names$ are the names of the events e_1 and e_2, and $e_1.p$, $e_2.p$ are the parameters of the events e_1 and e_2. Refinement is useful to specify only relevant event parameters and exclude the parameters that are not useful for the evaluation of a given policy, or conversely to gather additional parameters, depending on the use case. As an example, if we consider the following events:

$$e_1 = (send, (obj, mail), (actual, true), (to, "user@xyz.com"))$$
and
$$e_2 = (send, (obj, mail), (actual, true), (to, "user@xyz.com"), (from, "admin@xyz.com"))$$

[3]this distinction is important for information flow control, as an intended event can be blocked before the data is disseminated outside the system.

then, we can say that e_2 refines e_1 as the parameters e_2.p are a superset of e_1.p

The *traces* $\mathcal{T}$ model system runs by mapping abstract points in time $i \in \mathbb{N}$ to all system events that happened since $i-1$. Traces are formally defined as:

$$\mathcal{T} : \mathbb{N} \to \mathcal{P}(\mathcal{E}), \text{such that } \forall t \in \mathcal{T}, \forall i \in \mathbb{N}, i > 0, \forall e \in t(i) : i-1 < e.\text{time} \leq i$$

The set of all possible *data flow states* Σ is defined as $\Sigma \stackrel{def}{=} s \times a \times n$ where s, a and n are three mappings. Each state $\sigma \in \Sigma$ is defined by

- s is a *storage function*, $s : \mathcal{C} \to \mathcal{P}(\mathcal{D})$ returns the data stored in a container, capturing which containers *potentially* store which data;
- a is an *alias function*, $a : \mathcal{C} \to \mathcal{P}(\mathcal{C})$ which returns the list of containers that may be updated when a given container is updated. It captures the fact that some containers are automatically updated when others do, which is the case when containers store a copy of the same data and those data are modified.
- n is a naming function, $n : \mathcal{F} \to \mathcal{C}$ mapping identifiers to containers. For each state $\sigma \in \Sigma$, σ.n, σ.a and $\sigma.n$ refers to those mappings.

With the introduction of data flow states, it is now possible to extend the *refines* operator to $refines_\Sigma$ which describes the refinement of an event in the presence of a given state. The need for the introduction of $refines_\Sigma$ is that system events always operate on containers $\mathcal{C}$, while policies may be specified in terms of data $\mathcal{D}$. We say that (e_1, σ) refines e_2 if either:

1. both e_1 and e_2 operate on the same container and if e_1 *refines* e_2;
2. if e_1 operates on some container $c \in \mathcal{C}$ and e_2 on some data $d \in \mathcal{D}$ within c and e_1 *refines* e_2 when ignoring the parameter e.obj.

Therefore, the definition of (e_1, σ) *refines* e_2 is :

$$\begin{aligned} \forall e_1 \in \mathcal{S}, e_2 \in \mathcal{E}, \sigma \in \Sigma : (e_1, \sigma)\, refines_\Sigma\, e_2 \stackrel{\text{def}}{\iff} \quad & \exists c \in \mathcal{C} : e_1.obj = c \wedge e_2.obj = c \wedge e_1 refines\ e_2 \\ & \vee \exists c \in C, d \in \mathcal{D} : e_1.n = e_2.n \wedge e_1.obj = c \wedge e_2.obj = d \\ & \wedge d \in \sigma.s(c) \wedge e_{1..}p \backslash \{(obj, c)\} \supseteq e_2.p \backslash \{(obj, d)\} \end{aligned}$$

The different elements that have been introduced in this section enable us to model the information flow. However, it is also necessary to model usage control policies, i.e., how data can be used. Besides, it is necessary to model the reaction of the system when information flow policy violations are detected, which we discuss next.

5.3.2 Modeling usage control policies with ECA rules

While in the preceding section (Section 5.3.1), we introduced the model of information flow control, we now detail the modeling around data usage control policies, using ECA rules.

ECA rules. Usage control policies can be translated into Event-Condition-Action rules (ECA) to facilitate their enforcement at the PEP [Kumari and Pretschner, 2013]. The semantics of the ECA rules is as follows: if an event e' refining the trigger Event e (E) is observed at timestep i and if the execution of this event would make the Condition C true, then the additional Action A may be performed at timestep $i+1$. Actions include allowance, inhibition or delaying the event e' as well as the execution of other events.

ECA rules are expressed using four different operators with the following syntax:

$$\Psi \stackrel{def}{=} \underline{true}|\underline{false}|\mathcal{E}$$
$$\Omega \stackrel{def}{=} \underline{notIn}(\mathcal{D},\mathcal{P}(\mathcal{C}))|\underline{combined}(\mathcal{D},\mathcal{D},\mathcal{P}(\mathcal{C}))|\underline{maxIn}(\mathcal{D},\mathbb{N},\mathcal{P}(\mathcal{C}))$$
$$\Phi \stackrel{def}{=} (\Phi)|\Psi|\Omega|\underline{not}(\Phi)|\Phi\underline{and}\Phi|\Phi\underline{or}\Phi|\Phi\underline{since}\Phi|\Phi\underline{before}\mathbb{N}|\ \underline{repmin}(\mathbb{N},\mathbb{N},\mathcal{E})|\underline{repmax}(\mathbb{N},\mathbb{N},\mathcal{E})|\underline{always}(\Phi)|\underline{evalExt}(\Gamma)$$
$$\Gamma \stackrel{def}{=} \Psi|\mathbb{N}|op|String|...$$

In the above syntax, $\mathcal{D}$ represents a set of data items, $\mathcal{C}$ a set of containers and $\mathcal{E}$ a set of events. Γ is designed to incorporate external specifications and evaluation logic. It is left unspecified, but usually leverages XPath[4] for this purpose [Feth and Pretschner, 2012, Wüchner and Pretschner, 2012]. This part of the formalism is meant to enable generalization. Operator $\underline{evalExt}$ allows to incorporate external specification and evaluation logics, to specify conditions that refer to subject and object attributes or environmental conditions, such as time and location. $\underline{evalExt}$ as a general operator could be used for DIFC incorporation but requires additional evaluation to ensure the incorporation's correctness [Kelbert and Pretschner, 2018].

Ω defines state-based operators that constrain the data flow state, which is directly related to the place of data within the containers. Notably, $\underline{notIn}(d,\mathcal{C})$ is true if the data d is not in any of the containers of $\mathcal{C}$, $\underline{maxin}(d,m,\mathcal{C})$ if d is contained in at most m containers of $\mathcal{C}$. In other words, Ω constrains the allowed states and by combination determines in which containers D must or must not be contained. The $\underline{combined}(d_1,d_2,\mathcal{C})$ operator is true if there exists at least one container $c \in \mathcal{C}$ containing d_1 and d_2

Ψ intuitively refers to the constants true and false, but also to events $\mathcal{E}$. For instance, an event operator refine(e) where e is in $\mathcal{E}$ evaluates to true if and only if an event refining e is happening in the current timestep and the current data flow state. Finally, Φ defines the propositional, temporal and cardinality operators. not, and, or and always are intuitive. α since β is true if β was true at a moment earlier, i.e., during a former

[4]XPath (XML Path Language) is an expression language designed to support the query or transformation of XML documents. It enables the extraction of external specifications from XML documents.

timestep and regardless of the current status, and α is true since then, or always true. α before $j \in \mathbb{N}$ is true if α was true exactly j timesteps ago. repmin and repmax correspond respectively to an event appearing at least or at most n times during the j last timesteps.

5.3.3 Existing usage control model for distributed systems

We now introduce the distribution aspects of the state-of-the-art formalism on data usage control. These aspects are threefold. First, the selective distribution of the UCS components. Second, the separation of the whole distributed system into individual systems, that can be merged into sets of individual systems. Finally, the introduction of functions to identify the relevant system sets to evaluate usage control policies.

Local PDP/PIP components. The first step to the distribution is the introduction of local PDP/PIP couples (cf. Section 2.1.2), to take policy decisions locally [Kelbert and Pretschner, 2018]. Distributing these couples however requires handling the communication between the distributed instances, which would otherwise create a significant overhead. In the worst-case scenario, all instances of PDP/PIP couples must share their local environment with all the other instances. The communication cost increases exponentially with the number of couples. Optimization effort is compulsory to make the distributed policy evaluation viable.

Individual systems. To optimize policy enforcement in distributed systems, it is necessary to identify the relevant set of systems where a given policy must be enforced. This is useful as 1) it avoids monitoring the whole network for every policy; 2) it reduces the network exchanges between the different PDP instances.

Individual systems $\mathcal{Y}$ constitute the autonomous parts of the distributed system [Kelbert and Pretschner, 2018]. A *system* is a non-empty set of individual systems whose PEPs share the same PDP/PIP couple. Each individual system $y \in \mathcal{Y}$ has its own events $\mathcal{E}_y$, its own containers $\mathcal{C}_y$, its own set of identifiers $\mathcal{F}_y$, its own system runs $\mathcal{T}_y$, and its own possible data flow states Σ_y. Each event e is required to carry a parameter e.sys with sys $\in \mathcal{N}$ ($\mathcal{N}$ is a set of names) to identify the system in which it is happening. In practice, individual systems run in parallel and have independent traces and flow states. For policy enforcement, the reasoning must be done on sets of individual systems, called *sets of systems* $Y \subseteq \mathcal{Y}$ or on the whole distributed system. Sets of systems also have :

1. a set of events defined as the union of individual system's events: $\mathcal{E}_Y \stackrel{def}{=} \cup_{y \in Y} \mathcal{E}_y$;
2. a set of containers $\mathcal{C}_Y \stackrel{def}{=} \cup_{y \in Y} \mathcal{C}_y$;
3. a set of identifiers $\mathcal{I}_Y \stackrel{def}{=} \cup_{y \in Y} \mathcal{I}_y$;

4. a set of possible system runs (modeled by traces) $\mathcal{T}_Y : \mathbb{N} \rightarrow \mathcal{P}(\mathcal{E}_Y)$. At the difference of the other parameters, traces can not be formed as the union of the possible system runs of the individual systems composing the set of systems.

Relating individual systems with system sets. From individual system traces, it is possible to define the combined traces of systems [Kelbert and Pretschner, 2018]. Let $\prod$ be the Cartesian product of sets. We can define $\tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y$ as a set of traces of all individual systems. $t_y^\tau \in \mathcal{T}_y$ refers to the trace of the system $y \in \mathcal{Y}$ while σ_{t_y} refers to the corresponding data flow state. $t_Y^\tau \in \mathcal{T}_Y$ is defined as the combined traces of systems $Y \subseteq \mathcal{Y}$. For each timestep $i \in \mathbb{N}$, the set of events observed in Y is the union of the events in all individual systems $y \in Y$, such that:

$$\forall Y \subseteq \mathcal{Y}, i \in \mathbb{N}, \tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y, t_Y^\tau \in \mathcal{T}_Y : t_Y^\tau(i) \stackrel{def}{=} \bigcup_{y \in Y} t_y^\tau(i)$$

This corresponds to what a centralized PDP would observe. Analogously, we combine the data flow states of individual systems to form a global data flow state $\sigma_{t_Y^\tau}$, representing what a central PIP would observe. For convenience, $\forall i \in \mathbb{N}$ a given timestep, $\sigma_{t_y^\tau}(i)$ will be noted $\sigma^i_{t_y^\tau}$ in the following. $\sigma_{t_y^\tau}$ is constructed by unifying the mappings of the individual systems' storage, alias, and naming functions:

$$\forall Y \in \mathcal{Y}, i \in \mathbb{N}, \tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y, y \in Y, t_Y^\tau \in \mathcal{T}_Y, t_y^\tau \in \mathcal{T}_y, \sigma^i_{t_Y^\tau} \in \Sigma_Y, \sigma^i_{t_y^\tau} \in \Sigma_y, c \in \mathcal{C}_y, j \in \mathcal{I}_y:$$
$$\Sigma_Y \subseteq \Sigma : (\mathcal{C}_Y \rightarrow \mathcal{P}(\mathcal{D})) \times (\mathcal{C}_Y \rightarrow \mathcal{P}_Y) \times (\mathcal{I}_Y \rightarrow \mathcal{C}_Y)$$

which leads to the definition of $\sigma^i_{t_Y^\tau}.s(c)$ such that:

$$\sigma^i_{t_Y^\tau}.s(c) \stackrel{def}{=} \sigma^i_{t_y^\tau} \wedge \sigma^i_{t_Y^\tau}.a(c) \stackrel{def}{=} \sigma^i_{t_y^\tau}.a(c) \wedge \sigma^i_{t_Y^\tau}.n(j) \stackrel{def}{=} \sigma^i_{t_y^\tau}.n(j)$$

To recapitulate, t_Y^τ and $\sigma^i_{t_y^\tau}$ of set of systems Y are the reflection of the trace (t_Y^τ) and the state ($\sigma^i_{t_y^\tau}$) of Y.

Identification functions. To avoid unnecessary communication, functions are needed to identify the relevant systems for evaluating the policy p. First, the three following auxiliary functions are defined [Kelbert and Pretschner, 2018]:

1. *know*C: $\mathcal{P}(\mathcal{C}) \rightarrow \mathcal{P}(\mathcal{Y})$. Given a set of containers $\mathcal{C}$, *know*C returns the set of systems in which one of the containers $c \in \mathcal{C}$ resides. The formal definition of *know*C is: $\forall C \in \mathcal{C} : knowC(C) \stackrel{def}{=} \{y \in \mathcal{Y} | \mathcal{C}_y \cap C \neq \emptyset\}$;

2. *knowD*: $\mathcal{P}(\mathcal{D}) \times \mathbb{N} \times \prod \mathcal{T} \rightarrow \mathcal{P}(\mathcal{Y})$. Given a set of data items $\mathcal{D}$, a point in time $t \in \mathbb{N}$ and a tuple of concurrently executing traces $\mathcal{T}$, *knowD* relies on the systems' information flow to return the set of systems where there are one or more containers store at least one of the data items $d \in \mathcal{D}$:

$\forall D \subseteq \mathcal{D}, i \in \mathbb{N}, \tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y : knowD(D, i, \tau) \stackrel{def}{=} \{y \in \mathcal{Y} | \exists c \in \mathcal{C}_y, t_y^\tau \in \mathcal{T}_y, \sigma^i_{t_y^\tau} \in \Sigma_y : D \cap \sigma^i_{t_y^\tau}.s(c) \neq \emptyset\}$

knowD is parameterized in time and requires traces as input, as data are continuously propagated across systems;

3. *happens*: $\mathcal{E} \times \mathbb{N} \times \prod \mathcal{T} \rightarrow \mathcal{P}(\mathcal{Y})$. Given an event $e \in \mathcal{E}$, a point in time, and a tuple of concurrently executing traces, *happens* returns the set of systems where any event refining e happens. $\forall e \in \mathcal{E}, i \in \mathbb{N}, \tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y : happens(e, i, \tau) \stackrel{def}{=} \{y \in \mathcal{Y} | \exists t_y^\tau \in \mathcal{T}_y, e' \in t_y^\tau(i), \sigma^i_{t_y^\tau} \in \Sigma_y : (e', \sigma^i_{t_y^\tau})\ refines_\Sigma\ e\}$

Using these three auxiliary functions, the *relevant* function is composed so as to detect individual systems potentially interesting for the evaluation of policies. For a policy p, *relevant*: $\Phi \times \mathbb{N} \times \prod \mathcal{T}$ identifies the set of systems that may contribute to the evaluation of conditions $\varphi_p \in \Phi$ at a given timestep $i \in \mathbb{N}$ and given a set of concurrently executing traces $\tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y$. *relevant* is defined using the operators Ψ, Ω, Φ *and* Γ (cf. Figure 5.1).

$$
\begin{aligned}
&\forall \varphi \in \Phi, i \in \mathbb{N}, \tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y : \text{relevant}(\varphi, i, \tau) \stackrel{\text{def}}{=} \{y \in Y \mid \\
&\left(\left(\varphi = \underline{true} \vee \varphi = \underline{false}\right) \Longrightarrow Y = \emptyset\right) \textbf{ (1)} \\
&\vee \exists e \in \mathcal{E} \cdot (\varphi = e \Longrightarrow Y = \text{ happens } (e, i, \tau)) \textbf{ (2)} \\
&\vee \exists d \in \mathcal{D}, m \in \mathbb{N}, C \subseteq \mathcal{C} \cdot ((\varphi = \underline{notIn}(d, C) \vee \varphi = \underline{maxin}(d, m, C)) \Longrightarrow \quad Y = \text{knowD}(\{d\}, i, \tau) \cap \text{knowC}(C)) \textbf{ (3)} \\
&\vee \exists d_1, d_2 \in \mathcal{D}, C \subseteq \mathcal{C} \cdot (\varphi = \underline{combined}\,(d_1, d_2, C) \Rightarrow Y = \text{knowD}\,(\{d_1\}, i, \tau) \cap \text{ knowD }\,(\{d_2\}, i, \tau) \cap \text{knowC}(C)) \textbf{ (4)} \\
&\vee \exists \alpha \in \Phi \cdot (\varphi = \underline{not}(\alpha) \Longrightarrow Y = \text{relevant}(\alpha, i, \tau)) \textbf{ (5)} \\
&\vee \exists \alpha, \beta \in \Phi \cdot ((\varphi = \alpha\ \underline{and}\ \beta \vee \varphi = \alpha\ \underline{or}\ \beta) \Rightarrow \quad Y = \text{relevant}(\alpha, i, \tau) \cup \text{relevant}(\beta, i, \tau)) \textbf{ (6)} \\
&\vee \exists a, \beta \in \Phi \cdot (\varphi = \alpha\ \underline{since}\ \beta \Rightarrow \quad Y = \bigcup_{j=0}^{i} (\text{relevant}(\alpha, j, \tau) \cup \text{relevant}(\beta, j, \tau))) \textbf{ (7)} \\
&\vee \exists \alpha \in \Phi, j \in \mathbb{N} \cdot (\varphi = \alpha\ \underline{before}\ j \Rightarrow \quad Y = \text{relevant}(\alpha, i - j, \tau)) \textbf{ (8)} \\
&\vee \exists j, m \in \mathbb{N}, e \in \mathcal{E} \cdot (\varphi = \underline{repmin}(j, m, e) \Longrightarrow \quad Y = \bigcup_{k=0}^{\min(i,j)-1} \text{happens(e}, i - k, \tau))\} \textbf{ (9)}
\end{aligned}
$$

Figure 5.1 – Definition of the function *relevant* to identify systems of interest to evaluate a policy

5.3.4 Rationale for extending the existing model

Distributed systems are computer systems composed of multiple independent components, such as processors, storage devices, and communication channels, that cooperate to perform a unified task. Their purpose is to provide high performance, scalability, fault tolerance, and transparency to the users. They are consequently used in several

fields including the Internet of Things (IoT), Cloud computing and peer-to-peer networks.

The current state-of-the-art formalism for usage control in distributed systems of Kelbert and Pretschner [Kelbert and Pretschner, 2018] (cf. Section 5.3.3) indeed considers the distribution of the UCS components, notably the PDP/PIP couple. However, the dimension of user-side distribution is important, where groups of data owners define possibly conflicting policies for access. We propose to use the DIFC to define these group policies. In general, data usage control in distributed systems is a recent field of research, and actual challenges that must be addressed are not clearly identified nor formalized [Gil et al., 2022].

Kelbert and Pretschner [Kelbert and Pretschner, 2018] mention DIFC in the related work, claiming that 1) data usage control policies can express similar constraints than DIFC; 2) DIFC mechanisms can be integrated via the evalExt operator. While the former affirmation is partly true regarding the constraints in the policies, data usage control policies do not consider the owner-side distribution of the data, and ignore the architectural consequences for the UCS. For the second affirmation, the evaluation of external specification is indeed considered, but the evalExt operator requires additional analyses when a concrete framework realizing evalExt is implemented as stated by Kelbert and Pretschner [Kelbert and Pretschner, 2018] These limitations, the need to consider distribution as a whole and the recent advancements in DIFC enabling its practical use justify the integration of DIFC in distributed usage control modeling.

Besides, individual systems as defined in Section 5.3.3 may be disconnected from the network at a given timestep $i \in \mathbb{N}$. This is particularly true in the Internet of Things networks, as devices can be intermittent, on battery, or unreliable. The formalism must be able to take into account the possibility that a data$d$ is present in a container, but that the system where the container is located can not be reached at a given timestep.

To sum up, the extension of the existing model is justified by:

1. the need for users to define policies jointly in a decentralized manner. This requires the introduction of DIFC in the existing formalism, as the $\underline{evalExt}$ operator is too generic to ensure the incorporation's correctness;

2. the need to consider the connection status of individual systems.

In the following Section 5.4, we introduce auxiliary functions and dedicated components to evaluate DIFC policies in addition to data usage control policies. This results in different functions, where the focus is on detecting conflicts between labels. Additionally, we define functions to monitor the status of a system and determine if data and containers are currently accessible

5.4 Proposed extension

We propose the introduction of DIFC to the existing state-of-the-art formalism for distributed data usage control. This results in new formal definitions, and new components in the usage control architecture. In this section, an illustrative scenario is provided first to highlight the need for DIFC, before the actual details of the contribution.

5.4.1 Illustrative scenario

The illustrative scenario is again around car sharing as introduced in Section 3.4.1, but with a significative difference as several passengers can now be in the car. They share the same geolocation data but have different privacy needs.

In this scenario, *car owners* put their cars on a rental market, and *car renters* pay to rent the cars. The car renter may not be alone in the car. Therefore, *passengers* are also considered. Technically, these people share the *same geolocation data*, i.e., the location of the car. However, they may have different privacy needs, and the usage control policy should be defined by every one of them, with the most restrictive policy being applied. For instance, a passenger p authorizes the car owner c_{own} to monitor the car, and the car renter c_r authorizes the car owner c_{own} as well as third-party responsible for advertising third-parties adv. If c_{own} logically can access the data, should adv be allowed to access the data as well? This is precisely what DIFC is designed for, as it enables users to define policies as a group by applying labels to data, without requiring a central authority. This scenario is represented in Figure 5.2.

5.4.2 Additional functions for DIFC and network status aspects

As DIFC is only mentioned as an external specification in the state-of-the-art, DIFC objects, namely values, slots, channels and labels are not formalized in usage control models *per se*. However, data $d \in \mathcal{D}$ can be regarded as DIFC values. Similarly, as containers hold the data, they can be considered as slots as defined in the DIFC specification and have labels attached to them. A set of labels is noted $\mathcal{L}$, and can be composed of labels on any of the three DIFC objects that may be labeled, i.e., data, slots, and values. For convenience, we keep track of the objects associated with a label, such that $l.objects$ returns the list of data, slots or values associated with the label l.

Having different labels, data and containers may have conflicting policies. Detecting conflicts is crucial to enforce information flow policies, as a decision must be taken to prioritize either the container labeling or the data labeling. Detecting the conflicts requires additional functions, to fetch the labels associated to containers and data, which are defined here as part of the contribution:

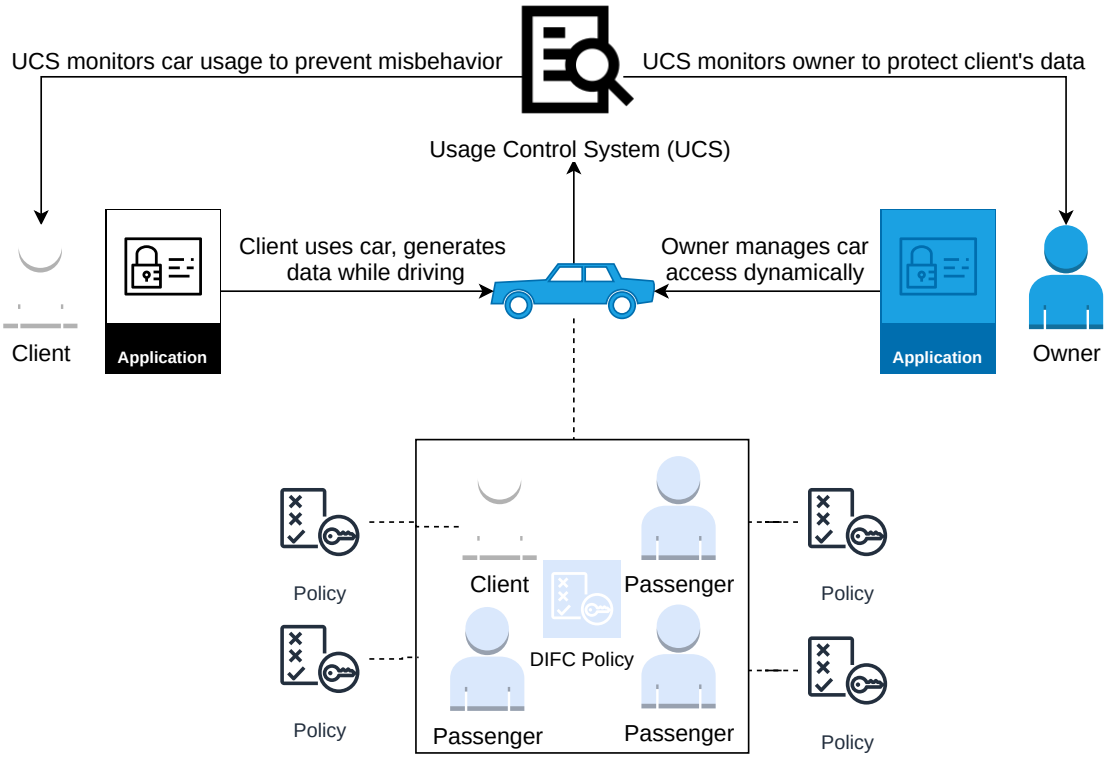


Figure 5.2 – Car sharing scenario and policies by car renter and passengers

(1) *labelC*: $\mathcal{P}(\mathcal{C}) \rightarrow \mathcal{P}(\mathcal{L})$. Given a set of containers, returns the labels associated with the containers: $\forall c \in \mathcal{C}, labelC(c) \stackrel{def}{=} \{l \in \mathcal{L} \mid c \in l.objects\}$;

(2) *labelD*: $\mathcal{P}(\mathcal{D}) \rightarrow \mathcal{P}(\mathcal{L})$. Given a set of data, returns the labels associated with the data items: $\forall d \in \mathcal{D}, labelD(d) \stackrel{def}{=} \{l \in \mathcal{L} \mid d \in l.objects\}$

(3) *conflict*: $\mathcal{P}(\mathcal{C}) \times \mathcal{P}(\mathcal{D}) \rightarrow \mathcal{P}(\mathcal{Y})$. Given a set of containers $\mathcal{C}$ and data $\mathcal{D}$, returns the set of systems where there is a difference between the label associated with a container and the label of one of their known data. This function relies on *knowD* (cf. Section 5.3.3) to identify the containers C knowing d in the first place, then, by comparing the labels of all $c \in C$ with the labels of d in $\mathcal{D}$: $\forall c \in \mathcal{C}, \forall d \in \mathcal{D}$, $conflict(c,d) \stackrel{def}{=} \{y \in \mathcal{Y} | \exists (c,d) : labelC(c) \neq labelD(d)\}$;

The *conflict* function, based on the *labelC* and *labelD* auxiliary functions, is dedicated to detecting individual systems $\mathcal{Y}$ where there is a conflict between the data label and the label of the container holding the data. These systems are of interest because this situation requires deciding which label should be prioritized, e.g., by considering the label representing the most restrictive policy.

Functions for network status. Distributed systems are used in a diversity of networks, including the Internet of Things and peer-to-peer networks. These networks are inherently dynamic, with devices and peers connecting and disconnecting very frequently. Besides, the network may be intermittent or unreliable. For proper policy evaluation, it is consequently required to determine whether an individual system or a set of systems $\mathcal{Y}$ are currently connected to a system. More accurately, an individual (or a set of) system is considered disconnected if it can not be reached by the usage control system. The main concern in our case is the possibility to evaluate policies, and it avoids introducing more intricate notions, like relative disconnection between two individual systems. By extension, it makes it possible to identify the subset of systems $\mathcal{Y}$ where a container $\mathcal{C}$ resides and that is connected to the distributed network. This leads to the definition of the following functions:

(1) *connected* : $\mathcal{Y} \times \mathbb{N} \times \prod \mathcal{T} \rightarrow \mathcal{P}(\mathcal{Y})$. Given a set of individual systems $\mathcal{Y}$, a point in time $i \in \mathbb{N}$ and a tuple of traces, return the subset of individual systems that are reachable in the network at this point in time. Traces are needed because the connection status depends on the timesteps considered. An individual system is considered connected if there is no disconnection $e_{disconnect}$ in the traces, i.e., an event whose name e.name = disconnect. *connected* is defined as:

$$\forall y \in \mathcal{Y}, i \in \mathbb{N}, \tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y,$$
$$connected(y, i, \tau) \stackrel{def}{=} \{y \in \mathcal{Y} \mid \forall e \in t_y^\tau(i), \text{e}.name \neq disconnect\}$$

(2) *availableC* : $\mathcal{P}(\mathcal{C}) \times \mathbb{N} \times \prod \mathcal{T} \rightarrow \mathcal{P}(\mathcal{Y})$. Given a set of containers, a point in time and a tuple of concurrently executing traces $\mathcal{T}$, *availableC* returns the set of currently reachable systems in which one of the containers $c \in \mathcal{C}$ resides. The formal definition of *availableC* is:

$$\forall C \in \mathcal{C}, i \in \mathbb{N}, \tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y : availableC(C) \stackrel{def}{=} \{y \in \mathcal{Y} \mid \mathcal{C}_y \cap C \neq \emptyset \wedge$$
$$connected(y, i, \tau) \cap knowC(C) \neq \emptyset\};$$

(3) *availableD* : $\mathcal{P}(\mathcal{D}) \times \mathbb{N} \times \prod \mathcal{T} \rightarrow \mathcal{P}(\mathcal{Y})$. Given a set of data items $\mathcal{D}$, a point in time $t \in \mathbb{N}$ and a tuple of concurrently executing traces $\mathcal{T}$, *availableD* returns the set of currently reachable systems where one or more containers are storing at least one of the data items $d \in \mathcal{D}$. *availableD* is formally defined as follows:

$$\forall D \subseteq \mathcal{D}, i \in \mathbb{N}, \tau \in \prod_{y \in \mathcal{Y}} \mathcal{T}_y :$$
$$availableD(D, i, \tau) \stackrel{def}{=} \{y \in \mathcal{Y} \mid \exists c \in \mathcal{C}_y, t_y^\tau \in \mathcal{T}_y, \sigma_{t_y^\tau}^i \in \Sigma_y : D \cap \sigma_{t_y^\tau}^i.s(c) \neq \emptyset \wedge$$
$$connected(y, i, \tau) \cap knowD(D, i, \tau) \neq \emptyset\}$$

In this section, we introduced several formal functions to handle DIFC labels (*labelC* and *labelD*) and detect conflicts between the data and their containers (*conflict*). Additionally, we introduced functions to handle the status of the network, detecting if a set of individual systems is connected or not (*connected*), and consequently if a container can be accessed and data fetched depending on the system's state. In the next section, we introduce the new components needed in the usage control system architecture to handle DIFC aspects, as part of our contribution.

5.4.3 Integrating DIFC components in the usage control system

Distributing the system requires the introduction of several components for the UCS to handle cross-system communication, information flow tracking and policy deployment. In addition to the existing UCS components in centralized settings (cf. Section 2.1.2), the usage control system has the following components:

- The *Distribution Management Point* (DMP) organizes the information flow between systems and the policy propagation while relying on a distributed database to coordinate the policy decisions,
- the *PDP server* provides the interface so that PEPs can request data usage control decisions.
- The *PIP Server* provides the interface to fetch environment attributes from the PIPs.
- The *DMP Server* allows communication with remote DMPs.
- the *Context Handler* orders and processes requests sent to the servers, notably to avoid event reordering and race conditions. The architecture is represented in Figure 5.3;
- The *communication manager* manages all external communication and runs servers to show the components' functionalities to the outside. It is not a component itself and is composed of the Context Handler, PDP server, PIP server and DMP server.

Additional components for DIFC processing. Decentralized information flow control is often processed using a dedicated instance, called the DIFC platform [Schultz and Liskov, 2013] or DIFC center [Lu et al., 2022]. There are different ways to process DIFC tasks: either use already existing components, e.g., the PDP, or introduce new dedicated components to manage DIFC aspects. Since DIFC policies differ from usage control policies, it is better to introduce dedicated components to handle DIFC aspects.

For distribution purposes, DIFC platform components are added as a novelty to the existing architecture [Lu et al., 2022], and interact only with the DMP to distribute the

policies to other individual systems $\mathcal{Y}$ or system set Y. The DIFC platform is responsible for pre-processing (DPP), privilege extraction (PP) and data labeling (DLP). Each operation has a dedicated component, as depicted in Figure 5.3:

- the *Data Labeling Point* (DLP) is responsible for labeling data, slots and channels;
- the *Privilege Point* (PP) is responsible for privilege extraction, converting labels into actual access rights;
- the *Data Pre-processing Point* (DPP) routes the data to other components of the DIFC platform. If the data is unlabeled, the data are sent to the DLP. Conversely, labeled data are sent to PP for privilege extraction (cf. Section 5.2.1).

Existing interfaces. Interfaces are associated with the components to provide several methods for usage control. The PDP server provides two interfaces `IPmp2Pdp` and `IPep2Pdp`, providing methods to deploy or retrieve policies, to signal system events and to await policy decisions. The PDP uses interface `IPdp2Pip` to evaluate state-based operators, retrieve all data stored in a container $\mathcal{C}$, and signal system events associated with information flow. The PMP uses interface `IPmp2Pip` to inform the PIP about initial data representations. Interface `IDmp2Pip` is used by the DMP if there are cross-system data flows. The DMP provides interface `IDmp2Pmp`, which is used by the local DMP to deploy policies and to retrieve currently deployed policies. The DMP additionally provides four interfaces:

1. `IPip2Dmp` allows the PIP to inform the DMP about data flows to the remote socket containers;
2. `IPdp2Dmp` enables the coordination of policy decisions across PDPs. This includes methods to notify that an operator's state has changed, to query whether the state of some operator has changed at remote PDPs, and to synchronize the points in time in which policies are evaluated;
3. `IPmp2Dmp` allows the PMP to register policies at the DMP;
4. `IDmp2Dmp` provides functionalities for cross-system information flow tracking and policy propagation between remote DMPs.

DIFC interfaces. As part of the contribution, the introduction of DIFC components requires to add four new interfaces (represented in Figure 5.3):

1. `IDPP2DLP` is an interface implemented so that the data can be labeled by the data labeling point (DLP) if the component DPre detects that it has not been labeled yet;

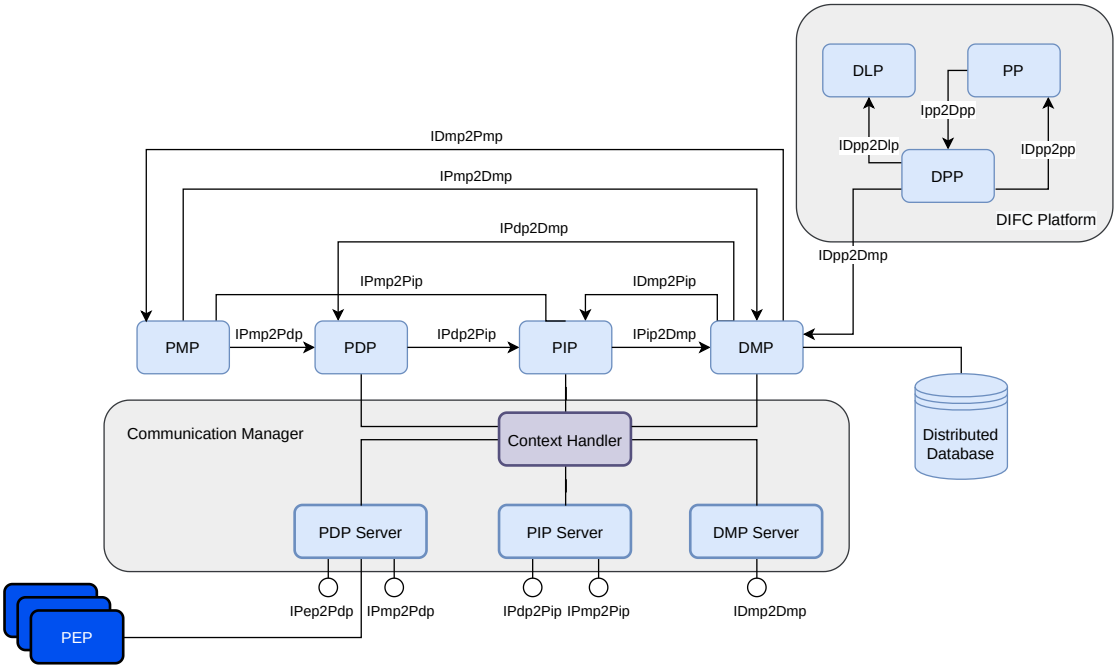


Figure 5.3 – Architecture of the usage control system [Kelbert and Pretschner, 2018], with the DIFC platform for policy labeling as a contribution.

2. `IPP2DPP` is needed to answer a request from the data processing point (DPP) and to notify it with updates of privileges;
3. `IDPP2PP` is an interface that enables the data pre-processing point (DPP) to fetch the required privileges from the dedicated privilege point;
4. `IDPP2DMP` provide methods so that the DIFC policies are propagated to all individual systems.

5.5 Conclusion

In this chapter, we propose an extended usage control model for distributed systems, by introducing elements of decentralized information flow control (DIFC) into the state-of-the-art formalism [Kelbert and Pretschner, 2018]. The extension of the former modeling is justified by the need to consider all the aspects of the distribution, including the user-side policy definition, as well as specificities of IoT networks that require a more sophisticated distributed system modeling. Recent advances in DIFC, based on evolving coding practices [Liu et al., 2022], make DIFC usable and justify its consideration in distributed system modeling. Although this formalism enables the addition of external

specification, using the dedicated evalExt operator, we state that DIFC actually requires dedicated components and functions. As a consequence, we extend the existing usage control architecture adding a privilege point (PP), a data pre-processing component (DPre) and a last component dedicated to data labeling (DL). Interfaces are also considered to enable cross-component communication. Additionally, we define new functions to identify the relevant parts of the distributed system. The main function considered, *conflict*, detects individual systems $\mathcal{Y}$ (cf. Section 5.3.3) of the distributed system where the security policy on the data and their storage location are conflicting.

Chapter 6

Conclusion and Future Works

Contents

This concluding chapter first summarizes the contributions of this thesis (Section 6.1), before introducing the limitations of the presented works, notably concerning the key technologies used in the different contributions (Section 6.2). Finally, different research directions are provided, either to improve the existing work or to extend the contributions to solve other research issues (Section 6.3).

6.1 Summary

Throughout this document, we proposed several contributions that addressed four different research objectives. These objectives fall within the general topic of privacy preservation, security and performance in the Internet of Things, which are paramount for the nearing massive adoption of IoT devices. The contributions of this thesis can be summarized as follows. A key aspect regarding these contributions is that they addressed these requirements altogether, i.e., performance, privacy and security simultaneously. Besides, both practical aspects, such as integration with performance tests (Section 4.3) and theoretical, formal ones (Chapter 5) were presented in this work. These contributions are summarized as follows.

Framework for performance, privacy and security in the Internet of Things. (*Objective 1, Objective 2*). This first contribution is the design of a framework whose purpose is to address the privacy, security and performance requirements of the Internet of Things. The basis of the framework is the IOTA technology. The framework is centralized around the IOTA technology, used for its zero-fee transactions and attractive metrics for the Internet of Things. Other components are also related to IOTA, such as IOTA Access to monitor physical access to the Internet of Things devices, such as cars or doors. A decentralized mixer to obfuscate IOTA transactions is also part of the framework, as defined by Sarfraz [Sarfraz et al., 2019]. Mixing was complemented with merge avoidance, to further improve the privacy of the users. Finally, usage control mechanisms are introduced so that users can monitor the use and dissemination of their data, by defining policies.

Integration of usage control with distributed ledgers. (*Objective 3*). This contribution proposes to integrate usage control with distributed ledgers based on directed acyclic graphs (DAG), after an analysis of suitable distributed ledger technologies. Private blockchains (or DAGs) are also suitable, but have limited scalability due to centralized governance, and were not selected for the experiments. The benefits of integration are to make both the usage control system and the distributed ledger work in synergy: 1) the components of the usage control system contribute to the network security, by verifying the state of the ledger as a DAG node; 2) the usage control system processes the transaction data without intermediaries, which is faster and more reliable. An implementation of usage control integration is proposed on IOTA. Performance tests are conducted on this implementation, showing approximately a 90% decrease of the time needed to push transactions and make an access decision when the UCS is integrated.

Data usage control model for distributed systems. (*Objective 4*). In the last contribution, we proposed an extended model for data usage control in distributed systems. The need to define an all-encompassing model for distributed systems is motivated by their use in Internet of Things networks. To improve the existing model, we added dedicated functions and components dedicated to *decentralized information flow control*. This enables users to define policies jointly, removing the need for a centralized entity responsible for policy management. Decentralized information flow control has seen recent improvements, notably because of current coding practices [Liu et al., 2022], which justifies its introduction as a relevant, usable tool in distributed systems. The model is also complemented by new functions considering the state of the individual entities of the network, as IoT networks can be unreliable, intermittent, and subject to frequent connections and disconnections.

6.2 Limitations

There are several limitations in the thesis work that we discuss next, inherent to the technologies as part of the solutions such as IOTA, or due to validation aspects that could increase the confidence in the proposed solutions.

Limitations of IOTA. IOTA is a key technology used in this thesis as a basis to address the research objectives, notably *Objective 1* (privacy-preserving zero-fee transactions), *Objective 2* (suitable distributed ledger for the IoT) and to a lesser extent *Objective 3* as DAG-based distributed ledger, including IOTA, are particularly fitted for integration (cf. Chapter 4). The main benefits of using IOTA are to benefit from the properties provided by DAG-based distributed ledgers, including:

1. zero-fee transactions with the removal of the miner, or fundamentally, by trading a financial incentive with a cost-saving incentive;
2. attractive metrics for IoT use cases, such as high transaction throughput;
3. involvement of resource-constrained IoT devices.

Yet, these claims are partly inaccurate in the first version of IOTA. While transactions are actually zero-fee, current performance metrics are limited compared to the expected metrics. In theory, the more transactions in the network, the faster transactions are confirmed, but more transactions also create congestion. This impacts negatively the confirmation rate and confirmation time [Dong et al., 2019]. The phenomenon is due to the coordinator, the central component in charge of validating transactions during the early stages of IOTA. The second version of IOTA 2.0 (Coordicide [Popov, 2020]) is designed to remove this component which should provide the expected network metrics in IOTA. Yet, it is not possible to anticipate accurately the release of IOTA 2.0.

Besides, the reliance of IOTA on a non-financial incentive could be troublesome in the future, as the IOTA network requires stable, full nodes to be reliable, while full nodes are expensive to run [Dong et al., 2019]. The lack of motivation may hinder the network's ability to properly grow in the future, which is required to achieve the claimed high throughput.

Evaluating the scalability of distributed ledgers. Scalability has been identified as a key metric throughout this thesis work (cf. Section 2.4.1). Scalability is one of the expected benefits of decentralization, which was identified as one of the two criteria of interest to determine integrability (cf. 4.2.1). Nevertheless, as discussed in Section 2.4.1, scalability is complex to define in the first place, and requires handling any possible bottleneck. Dong *et al.* [Dong et al., 2019] have proposed to evaluate scalability as "the changes in throughput and latency when increasing the number of nodes", using the *confirmation rate*. The confirmation rate measures the number of transactions

confirmed[1] in a second, which is related to both throughput and latency. The results of Dong *et al.*'s evaluation show that the confirmation rate is at most $0.8 tx.s^{-1}$, much lower than IOTA's throughput. The Coordinator node of IOTA(1.0) is still a major obstacle to scalability considering the confirmation rate.

Adoption of usage control. If IOTA is the main technology used to answer IoT requirements in terms of performance, usage control is its counterpart concerning privacy needs in this thesis work. While it is a multipurpose technology with regard to the mitigation of privacy threats (cf. Section 4.4.3), it has several limitations preventing its adoption:

- the complexity of implementing and managing UCON policies. UCON systems require detailed specification and configuration of access control rules, obligations, and conditions, which can be challenging to design and maintain, especially in large and dynamic systems. Additionally, UCON may face resistance or challenges in adoption due to compatibility issues with existing access control mechanisms and the need for integration with different components and platforms within an organization's infrastructure;
- the policy enforcement points are distributed on the monitored devices, which creates risks to the security and privacy of users. While these risks are usually mitigated using a trusted execution environment, this solution has shortcomings as the TEE itself may not be secured;
- the potential impact on system performance. Usage control systems involve additional computational overhead to evaluate and enforce access control policies at runtime, which may be limiting in several use cases.

6.3 Future works

In this section, different prospects following this thesis work are developed. First, different works that could complete the proposed contribution are proposed. Additionally, orthogonal research topics are mentioned, that were not directly addressed in this work but which could benefit from its findings.

6.3.1 Automatic generation of XACML policies for testing

The contributions of this thesis often required to evaluate usage control policies, written in the XACML language. For instance, we assessed the time needed to evaluate

[1] A transaction is confirmed if it is referenced by a milestone from the coordinator in IOTA (1.0).

an XACML policy in Chapter 4, which is actually lightweight compared to other system and network calls. However, defining XACML policies is a cumbersome process, which hinders the adoption of usage control in practice. First, the clear and exhaustive definition of the policies themselves can be difficult, including for testing purposes. Besides, the translation from high-level policies to XACML policies is also tedious to do by hand (cf. Listing D.1).

Several works in the literature have been developing tools to make the generation of XACML policies more user-friendly. Bertolino *et al.* have proposed two different ways to derive XACML requests automatically for policy testing [Bertolino et al., 2012]. The authors also underline the impossibility of manual specification of a set of test cases in XACML testing. Similarly, Xu *et al.* use *mutation-based tests* where access requests are derived from an original policy [Xu et al., 2020]. While the same methodology was used in the presented thesis, the derivation was very basic and consisted of value substitution. Future works may include tests based on rigorous policy derivation.

6.3.2 Model checking on usage control model

For the *Contribution 3*, we proposed an extension of the state-of-the-art usage control model in distributed systems. However, the model can have flaws in its definition, including the existing model we used as a basis. Model-checking tools exist to ensure the model's validity.

TLA+ is a formal specification language developed by Leslie Lamport. It is used for designing, modeling, documentation, and verification of programs, especially concurrent systems and distributed systems [Lamport, 1992]. TLC is a model checker designed for TLA+ that can be used to detect errors in TLA+ specifications. The validation of the model could be of interest in particular when using the model in well-defined use cases. Besides, some functions such as the *conflict* function, responsible for detecting collisions between the labels of data and their containers, might have side effects and not be doable in practice, e.g., due to communication costs. An implementation with performance testing could exclude this risk and would fit the research objective *Objective 6* regarding validation using a proof of concept.

6.3.3 IOTA-based privacy-preserving machine learning

The Internet of Things can leverage advanced machine learning algorithms for its applications [Ali et al., 2021]. Machine learning (ML) and deep learning (DL) algorithms have significantly been improved and used in diverse applications, including computer vision, natural language processing and automated speech recognition. Several Internet of Things systems, such as autonomous vehicles, UAVs, drones or security robots, heavily rely on ML/DL-based technologies [Bian et al., 2022]. However, considering

the huge quantity of data stored at a central cloud server, adopting centralized machine learning algorithms is not a viable option due to computation cost and privacy leakage issues. Yet, leveraging distributed data for application purposes is still a challenging task [Ali et al., 2021]. To address this challenge, *federated learning* (FL) is a promising solution that distributes learning to the end devices without sharing personal data with the central server. In federated learning, the central server only orchestrates the learning process, and only the updates of model parameters are shared between end devices and the central orchestrator. Therefore, the central server does not need access to actual data to learn, reducing privacy risks.

Blockchain-based federated learning. Federated learning being distributed by nature, blockchain-enabled federated learning has been gathering significant attention, which is shown by frequent and recent surveys [Lee and Kim, 2021, Issa et al., 2023, Qu et al., 2023] or systematic literature reviews [Hou et al., 2021, Qammar et al., 2023]. Blockchains are leveraged in federated learning for the following reasons:

- *smart contracts* can be used to coordinate federated learning. Smart contracts can validate node contributions, compute the global model, record the performance of nodes on the ledger and provide incentives to nodes based on performance [Issa et al., 2023];
- improving security and privacy by removing the central server [Issa et al., 2023], mitigating the *single point of failure* threat;
- enhanced auditability and accountability of the nodes [Issa et al., 2023].

Yet, federated learning using blockchains still has to address several challenges to preserve privacy and to handle the Internet of Things constraints, e.g., low computation and storage capabilities [Issa et al., 2023]. These challenges are very similar to what has been addressed in Chapter 3 and Chapter 4) and most of the research objectives of this paper (*Objective 2*, and indirectly *Objective 3* but integrating federated learning instead of usage control). The results of this thesis work could also be extended to federated learning.

IOTA-based federated learning. Distributed ledgers based on DAGs are not mentioned in the literature as a potential solution outperforming blockchains for federated learning [Issa et al., 2023, Qu et al., 2023]. To my knowledge, only one research article from Lu *et al.* [Lu et al., 2020] suggests the use of a DAG-based distributed ledger combined with a blockchain for secure data sharing and federated learning. In the authors' setting, both the DAG and the blockchain are permissioned, which prevents its adoption in large-scale networks. IOTA could be leveraged as it can incorporate Internet of Things devices into its network efficiently, which is not the case of blockchains in general (cf. Section 2.2).

6.3.4 IOTA for supporting Self-Sovereign Identities

Self-Sovereign Identity (SSI) is an approach in which subjects are in full control of their own digital identities [Fedrecheski et al., 2020]. SSI contrasts with current digital identity solutions that are centralized which creates privacy and security issues [Fedrecheski et al., 2020].

Self-Sovereign Identities have several benefits to the Internet of Things. In particular, SSIs are *owner-centric* and enable users to be the root of trust of their devices instead of a third party. The identities of users and their devices are stored locally on their own devices and are disclosed selectively by the users themselves, improving privacy in contrast with centralized identity management. Removing the need for a third party to manage identities increases the decentralization in the network and removes a single point of failure from the network [Fedrecheski et al., 2020].

Yet, the adoption of the SSI paradigm in IoT networks faces several hurdles, both technical and non-technical e.g., standardization [Fedrecheski et al., 2020]. The technical aspects include the limitations of *constrained devices*. To implement Self-Sovereign Identities, the devices must be able to run asymmetric cryptography and handle the communication overhead.

The IOTA technology, due to its ability to integrate constrained devices into its consensus and its network, is a promising technology to address the first technical limit. In particular, IOTA gives the possibility to deploy nodes to the users, to which the computation-intensive aspects can be delegated. Existing works have proposed to use IOTA as the basis of a SSI system for the Internet of Things devices [Gebresilassie et al., 2020] and IOTA itself has a module to generate decentralized identities [Yarger, 2020]. Gebresilassie *et al.* propose to use DAGs as building blocks for a DAG node identity management system, in particular to manage node reputation. However, the contribution remains very elusive on many key technical aspects such as conditions for enrolling nodes in the SSI system, desired security properties, transaction content and security analysis which still leaves many unknowns before a possible implementation within a proof of concept.

Bibliography

[Ali et al., 2021] Ali, M., Karimipour, H., and Tariq, M. (2021). Integration of blockchain and federated learning for Internet of Things: Recent advances and future challenges. *Comput. Secur.*, 108:102355. 139, 140

[Alibeigi et al., 2019] Alibeigi, A., Munir, A., and Karim, M. (2019). Right to privacy, a complicated concept to review. *SSRN Electronic Journal*. 36

[Alshaikhli et al., 2022] Alshaikhli, M., Elfouly, T., Elharrouss, O., Mohamed, A., and Ottakath, N. (2022). Evolution of internet of things from blockchain to IOTA: A survey. *IEEE Access*, 10:844–866. 65, 73

[Andola et al., 2021] Andola, N., Raghav, Yadav, V. K., Venkatesan, S., and Verma, S. (2021). Anonymity on blockchain based e-cash protocols - A survey. *Comput. Sci. Rev.*, 40:100394. 60

[Andoni et al., 2018] Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., Mccallum, P., and Peacock, A. (2018). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 100:143–174. 157

[Aponte-Novoa et al., 2021] Aponte-Novoa, F. A., Orozco, A. L. S., Villanueva-Polanco, R., and Wightman, P. M. (2021). The 51% attack on blockchains: A mining behavior study. *IEEE Access*, 9:140549–140564. 51, 53, 100

[Asheralieva and Niyato, 2021] Asheralieva, A. and Niyato, D. (2021). Throughput-efficient Lagrange coded private blockchain for secured IoT systems. *IEEE Internet Things J.*, 8(19):14874–14895. 38

[Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805. 158

[Axiomatics, 2023] Axiomatics (2023). eXtensible Access Control Markup Language (XACML) — axiomatics.com. `https://axiomatics.com/resources/`

reference-library/extensible-access-control-markup-language-xacml. [Accessed 18-Jun-2023]. 159

[Ayoub et al., 2021] Ayoub, O., De Sousa, A., Mendieta, S., Musumeci, F., and Tornatore, M. (2021). Online virtual machine evacuation for disaster resilience in inter-data center networks. *IEEE Transactions on Network and Service Management*, 18(2):1990–2001. 10, 33

[Babil et al., 2013] Babil, G. S., Mehani, O., Boreli, R., and Kaafar, M. (2013). On the effectiveness of dynamic taint analysis for protecting against private information leaks on Android-based devices. In *2013 International Conference on Security and Cryptography (SECRYPT)*, pages 1–8. 89

[Back, 2002] Back, A. (2002). Hashcash - a denial of service counter-measure. http://www.hashcash.org/papers/hashcash.pdf. 48

[Baird et al., 2018] Baird, L., Harmon, M., and Madsen, P. (2018). Hedera: A public hashgraph network & governing council. "https://hedera.com/hh_whitepaper_v2.1-20200815.pdf". 54, 55, 157

[Bansal and Bhatia, 2020] Bansal, G. and Bhatia, A. (2020). A fast, secure and distributed consensus mechanism for energy trading among vehicles using hashgraph. In *2020 International Conference on Information Networking, ICOIN 2020, Barcelona, Spain, January 7-10, 2020*, pages 772–777. IEEE. 55

[Bao et al., 2023] Bao, Z., He, D., Khan, M. K., Luo, M., and Xie, Q. (2023). PBidm: Privacy-preserving blockchain-based identity management system for industrial internet of things. *IEEE Trans. Ind. Informatics*, 19(2):1524–1534. 10, 13, 33, 48

[Berte, 2018] Berte, D.-R. (2018). Defining the IoT. *Proceedings of the International Conference on Business Excellence*, 12:118–128. 36, 158

[Bertolino et al., 2012] Bertolino, A., Daoudagh, S., Lonetti, F., and Marchetti, E. (2012). Automatic XACML requests generation for policy testing. In Antoniol, G., Bertolino, A., and Labiche, Y., editors, *Fifth IEEE International Conference on Software Testing, Verification and Validation, ICST 2012, Montreal, QC, Canada, April 17-21, 2012*, pages 842–849. IEEE Computer Society. 20, 139

[Bian et al., 2022] Bian, J., Arafat, A. A., Xiong, H., Li, J., Li, L., Chen, H., Wang, J., Dou, D., and Guo, Z. (2022). Machine learning in real-time Internet of Things (IoT) systems: A survey. *IEEE Internet of Things Journal*, 9(11):8364–8386. 139

[Blockchain.info, 2023] Blockchain.info (2023). Size of the Bitcoin blockchain from January 2009 to June 8, 2023. https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/. 27, 34, 165

[Bodó et al., 2021] Bodó, B., Brekke, J. K., and Hoepman, J.-H. (2021). Decentralisation in the Blockchain Space. *Internet Policy Review*, 10(2):1–12. 96

[Bothra et al., 2023] Bothra, P., Karmakar, R., Bhattacharya, S., and De, S. (2023). How can applications of blockchain and artificial intelligence improve performance of internet of things? - A survey. *Comput. Networks*, 224:109634. 10, 33

[Bowe et al., 2016] Bowe, H. S., Hornby, T., and Wilcox, N. (2016). Zcash Protocol Specification. `https://github.com/zcash/zips/blob/main/protocol/protocol.pdf`. 62

[Boyes et al., 2018] Boyes, H., Hallaq, B., Cunningham, J., and Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. *Comput. Ind.*, 101:1–12. 158

[Breitman and Breitman, 2014] Breitman, A. and Breitman, K. (2014). Tezos — a self-amending crypto-ledger. White paper. `https://tezos.com/whitepaper.pdf`. 52

[Brotsis et al., 2021] Brotsis, S., Limniotis, K., Bendiab, G., Kolokotronis, N., and Shiaeles, S. (2021). On the suitability of blockchain platforms for IoT applications: Architectures, security, privacy, and performance. *Comput. Networks*, 191:108005. 14, 63

[Buterin, 2014] Buterin, V. (2014). Ethereum: A next-generation smart contract and decentralized application platform. Whitepaper. 49

[Carelli et al., 2022] Carelli, A., Palmieri, A., Vilei, A., Castanier, F., and Vesco, A. (2022). Enabling secure data exchange through the IOTA tangle for iot constrained devices. *Sensors*, 22(4):1384. 54

[Castro and Liskov, 1999] Castro, M. and Liskov, B. (1999). Practical byzantine fault tolerance. In Seltzer, M. I. and Leach, P. J., editors, *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 22-25, 1999*, pages 173–186. USENIX Association. 13, 53

[Cha et al., 2019] Cha, S., Hsu, T., Xiang, Y., and Yeh, K. (2019). Privacy enhancing technologies in the internet of things: Perspectives and challenges. *IEEE Internet Things J.*, 6(2):2159–2187. 43, 48

[Chen et al., 2022] Chen, X., Nguyen, K., and Sekiya, H. (2022). On the latency performance in private blockchain networks. *IEEE Internet Things J.*, 9(19):19246–19259. 14, 63

[Churyumov, 2017] Churyumov, A. (2017). Byteball whitepaper. `https://obyte.org/Byteball.pdf`. 54, 73

[Conti et al., 2022] Conti, M., Kumar, G., Nerurkar, P., Saha, R., and Vigneri, L. (2022). A survey on security challenges and solutions in the IOTA. *Journal of Network and Computer Applications*, 203:103383. 54, 73, 108

[Cooper et al., 2015] Cooper, C., Elsässer, R., Radzik, T., Rivera, N., and Shiraga, T. (2015). Fast consensus for voting on general expander graphs. In Moses, Y., editor, *Distributed Computing - 29th International Symposium, DISC 2015, Tokyo, Japan, October 7-9, 2015, Proceedings*, volume 9363 of *Lecture Notes in Computer Science*, pages 248–262. Springer. 75

[Data, 2023] Data, N. (2023). Bitcoin number of transaction per block. `https://data.nasdaq.com/data/BCHAIN/NTRBL`. 64

[Deng et al., 2011] Deng, M., Wuyts, K., Scandariato, R., Preneel, B., and Joosen, W. (2011). A Privacy Threat Analysis Framework: Supporting the Elicitation and Fulfillment of Privacy Requirements. *Requir. Eng.*, 16(1):3–32. 102, 109

[Denning, 1976] Denning, D. E. (1976). A lattice model of secure information flow. *Commun. ACM*, 19(5):236–243. 46, 116, 157

[Digiconomist, 2023] Digiconomist (2023). Bitcoin average energy consumption per transaction compared to that of visa as of may 1, 2023 (in kilowatt-hours) [graph]. https://www.statista.com/statistics/881541/bitcoin-energy-consumption-transaction-comparison-visa/. 27, 164

[Dong et al., 2019] Dong, Z., Zheng, E., Lee, Y. C., and Zomaya, A. Y. (2019). DAGBENCH: A performance evaluation framework for DAG distributed ledgers. In Bertino, E., Chang, C. K., Chen, P., Damiani, E., Goul, M., and Oyama, K., editors, *12th IEEE International Conference on Cloud Computing, CLOUD 2019, Milan, Italy, July 8-13, 2019*, pages 264–271. IEEE. 103, 137

[Duffield and Diaz, 2014] Duffield, E. and Diaz, D. (2014). Dash : A payments-focused. `https://github.com/dashpay/dash/wiki/Whitepaper`. 62

[Ekparinya et al., 2020] Ekparinya, P., Gramoli, V., and Jourjon, G. (2020). The attack of the clones against proof-of-authority. In *27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020*. The Internet Society. 52

[Elmimouni et al., 2023] Elmimouni, H., Shusas, E., Skeba, P., Baumer, E. P. S., and Forte, A. (2023). What makes a technology privacy enhancing? laypersons' and experts' descriptions, uses, and perceptions of privacy enhancing technologies. In Sserwanga, I., Goulding, A., Sandy, H. M., Du, J. T., Soares, A. L., Hessami, V., and

Frank, R. D., editors, *Information for a Better World: Normality, Virtuality, Physicality, Inclusivity - 18th International Conference, iConference 2023, Virtual Event, March 13-17, 2023, Proceedings, Part II*, volume 13972 of *Lecture Notes in Computer Science*, pages 229–250. Springer. 36

[Evaldas Drąsutis, 2021] Evaldas Drąsutis (2021). IOTA Smart Contracts. 91

[Fan et al., 2021] Fan, C., Ghaemi, S., Khazaei, H., Chen, Y., and Musílek, P. (2021). Performance analysis of the IOTA dag-based distributed ledger. *ACM Trans. Model. Perform. Evaluation Comput. Syst.*, 6(3):10:1–10:20. 14, 63

[Fedrecheski et al., 2020] Fedrecheski, G., Rabaey, J. M., Costa, L. C. P., Calcina Ccori, P. C., Pereira, W. T., and Zuffo, M. K. (2020). Self-sovereign identity for IoT environments: A perspective. In *2020 Global Internet of Things Summit (GIoTS)*, pages 1–6. 21, 141

[Feth and Pretschner, 2012] Feth, D. and Pretschner, A. (2012). Flexible data-driven security for android. In *Sixth International Conference on Software Security and Reliability, SERE 2012, Gaithersburg, Maryland, USA, 20-22 June 2012*, pages 41–50. IEEE. 122

[Fromm, 2020] Fromm, A. (2020). *Enhancing Data Flow Tracking for Data Usage Control*. PhD Dissertation, Technische Universität München. 17, 35, 116, 119

[Gao et al., 2022] Gao, N., Huo, R., Wang, S., Huang, T., and Liu, Y. (2022). Sharding-hashgraph: A high-performance blockchain-based framework for industrial internet of things with hashgraph mechanism. *IEEE Internet Things J.*, 9(18):17070–17079. 55

[Gebresilassie et al., 2020] Gebresilassie, S. K., Rafferty, J., Morrow, P., Chen, L., Abu-Tair, M., and Cui, Z. (2020). Distributed, secure, self-sovereign identity for IoT devices. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pages 1–6. 21, 141

[Gil et al., 2022] Gil, G., Arnaiz, A., Higuero, M., and Díez, F. J. (2022). Assessment framework for the identification and evaluation of main features for distributed usage control solutions. *ACM Trans. Priv. Secur.*, 26(1):10:1–10:28. 35, 126

[Glaeser et al., 2022] Glaeser, N., Maffei, M., Malavolta, G., Moreno-Sanchez, P., Tairi, E., and Thyagarajan, S. A. K. (2022). Foundations of coin mixing services. In Yin, H., Stavrou, A., Cremers, C., and Shi, E., editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 1259–1273. ACM. 59, 61, 109

[Godik and Moses, 2003] Godik, S. and Moses, T. (2003). eXtensible Access Control Markup Language (XACML). *OASIS Standard*. 103

[Goldwasser et al., 1985] Goldwasser, S., Micali, S., and Rackoff, C. (1985). The knowledge complexity of interactive proof-systems (extended abstract). In Sedgewick, R., editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 291–304. ACM. 62

[Goyat et al., 2022] Goyat, R., Kumar, G., Alazab, M., Conti, M., Rai, M. K., Thomas, R., Saha, R., and Kim, T. (2022). Blockchain-based data storage with privacy and authentication in internet of things. *IEEE Internet Things J.*, 9(16):14203–14215. 13, 48

[Guo et al., 2023] Guo, F., Xiao, X., Hecker, A., and Dustdar, S. (2023). A theoretical model characterizing tangle evolution in IOTA blockchain network. *IEEE Internet Things J.*, 10(2):1259–1273. 54, 73

[Haque et al., 2021] Haque, A. K. M. B., Islam, A. K. M. N., Hyrynsalmi, S., Naqvi, B., and Smolander, K. (2021). GDPR compliant blockchains-a systematic literature review. *IEEE Access*, 9:50593–50606. 58, 88, 109

[Harrigan and Fretter, 2016] Harrigan, M. and Fretter, C. (2016). The unreasonable effectiveness of address clustering. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, July 18-21, 2016*, pages 368–373. IEEE Computer Society. 14, 57, 59

[Harvan and Pretschner, 2009] Harvan, M. and Pretschner, A. (2009). State-based usage control enforcement with data flow tracking using system call interposition. In *2009 Third International Conference on Network and System Security*, pages 373–380. 116

[Hearn, 2013] Hearn, M. (2013). Merge avoidance. A note on privacy-enhancing techniques in the Bitcoin protocol. `https://blog.plan99.net/merge-avoidance-7f95a386692f`. 61, 62

[Henriksen-Bulmer and Jeary, 2016] Henriksen-Bulmer, J. and Jeary, S. (2016). Re-identification attacks— A systematic literature review. *International Journal of Information Management*, 36(6, Part B):1184 – 1192. 88

[Hou et al., 2021] Hou, D., Zhang, J., Man, K. L., Ma, J., and Peng, Z. (2021). A systematic literature review of blockchain-based federated learning: Architectures, applications and issues. In *2021 2nd Information Communication Technologies Conference (ICTC)*, pages 302–307. 20, 140

[Howard and Lipner, 2006] Howard, M. and Lipner, S. (2006). *The security development lifecycle*, volume 8. Microsoft Press Redmond. 16, 57, 89

[Hu et al., 2022] Hu, W., Ardeshiricham, A., and Kastner, R. (2022). Hardware information flow tracking. *ACM Comput. Surv.*, 54(4):83:1–83:39. 46

[Huang et al., 2017] Huang, B., Liu, Z., Chen, J., Liu, A., Liu, Q., and He, Q. (2017). Behavior pattern clustering in blockchain networks. *Multim. Tools Appl.*, 76(19):20099–20110. 58

[Insights, 2023] Insights, T. (2023). Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030, by vertical (in millions). `https://www.statista.com/statistics/1194682/iot-connected-devices-vertically/`. 9, 31, 32

[Institute, 2021] Institute, T. P. E. (2021). Annual energy consumption for bitcoin, ethereum and selected european union countries in 2019 (in terawatt hour) [graph]. https://www.statista.com/statistics/1243388/eu-annual-energy-consumption-for-bitcoin-ethereum/. 27, 164

[Issa et al., 2023] Issa, W., Moustafa, N., Turnbull, B. P., Sohrabi, N., and Tari, Z. (2023). Blockchain-based federated learning for securing Internet of Things: A comprehensive survey. *ACM Comput. Surv.*, 55(9):191:1–191:43. 20, 26, 49, 50, 51, 140

[Jha et al., 2022] Jha, S., Jha, N., Prashar, D., Ahmad, S., Alouffi, B., and Alharbi, A. (2022). Integrated IoT-based secure and efficient key management framework using hashgraphs for autonomous vehicles to ensure road safety. *Sensors*, 22(7):2529. 55

[Kelbert and Pretschner, 2013] Kelbert, F. and Pretschner, A. (2013). Data usage control enforcement in distributed systems. In Bertino, E., Sandhu, R. S., Bauer, L., and Park, J., editors, *Third ACM Conference on Data and Application Security and Privacy, CODASPY'13, San Antonio, TX, USA, February 18-20, 2013*, pages 71–82. ACM. 17, 35, 116

[Kelbert and Pretschner, 2014] Kelbert, F. and Pretschner, A. (2014). Decentralized distributed data usage control. In Gritzalis, D., Kiayias, A., and Askoxylakis, I. G., editors, *Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings*, volume 8813 of *Lecture Notes in Computer Science*, pages 353–369. Springer. 35, 116

[Kelbert and Pretschner, 2015] Kelbert, F. and Pretschner, A. (2015). A fully decentralized data usage control enforcement infrastructure. In Malkin, T., Kolesnikov, V., Lewko, A. B., and Polychronakis, M., editors, *Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers*, volume 9092 of *Lecture Notes in Computer Science*, pages 409–430. Springer. 35, 116

[Kelbert and Pretschner, 2018] Kelbert, F. and Pretschner, A. (2018). Data usage control for distributed systems. *ACM Trans. Priv. Secur.*, 21(3):12:1–12:32. 27, 35, 89, 91, 116, 117, 119, 122, 123, 124, 126, 132

[Ketsdever and Fischer, 2019] Ketsdever, S. and Fischer, M. J. (2019). Incentives don't solve blockchain's problems. In *10th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON 2019, New York City, NY, USA, October 10-12, 2019*, pages 873–876. IEEE. 98

[Khacef et al., 2021] Khacef, K., Benbernou, S., Ouziri, M., and Younas, M. (2021). Trade-off between security and scalability in blockchain design: A dynamic sharding approach. In Awan, I., Benbernou, S., Younas, M., and Aleksy, M., editors, *The International Conference on Deep Learning, Big Data and Blockchain (Deep-BDB 2021), Virtual Event, 23-25 August, 2021*, volume 309 of *Lecture Notes in Networks and Systems*, pages 77–90. Springer. 100

[Khan et al., 2020] Khan, Y., Syed, T. A., Fariz, M., Moreira, F., Branco, F., Martins, J., and Gonçalves, R. (2020). BlockU: Extended usage control in and for blockchain. *Expert Syst. J. Knowl. Eng.*, 37(3). 16, 34, 35, 94

[King and Nadal, 2012] King, S. and Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. 52

[Kokoris-Kogias, 2022] Kokoris-Kogias, L. (2022). Understanding Blockchain Latency and Throughput. `https://www.paradigm.xyz/2022/07/consensus-throughput`. 64

[Koshy et al., 2014] Koshy, P., Koshy, D., and McDaniel, P. D. (2014). An analysis of anonymity in bitcoin using P2P network traffic. In Christin, N. and Safavi-Naini, R., editors, *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, volume 8437 of *Lecture Notes in Computer Science*, pages 469–485. Springer. 59

[Kumari and Pretschner, 2013] Kumari, P. and Pretschner, A. (2013). Model-based usage control policy derivation. In Jürjens, J., Livshits, B., and Scandariato, R., editors, *Engineering Secure Software and Systems - 5th International Symposium, ESSoS 2013, Paris, France, February 27 - March 1, 2013. Proceedings*, volume 7781 of *Lecture Notes in Computer Science*, pages 58–74. Springer. 122

[Lamport, 1992] Lamport, L. (1992). Hybrid systems in TLA$^+$. In Grossman, R. L., Nerode, A., Ravn, A. P., and Rischel, H., editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 77–102. Springer. 19, 139

[Law et al., 1997] Law, L., Sabett, S., and Solinas, J. A. (1997). How to make a mint: The cryptography of anonymous electronic cash. *The American University law review*, 46:6. 48

[Lazouski et al., 2010] Lazouski, A., Martinelli, F., and Mori, P. (2010). Usage control in computer security: A survey. *Comput. Sci. Rev.*, 4(2):81–99. 19

[Lee and Kim, 2021] Lee, H. and Kim, J. (2021). Trends in blockchain and federated learning for data sharing in distributed platforms. In *Twelfth International Conference on Ubiquitous and Future Networks, ICUFN 2021, Jeju Island, South Korea, August 17-20, 2021*, pages 430–433. IEEE. 20, 140

[LeMahieu, 2017] LeMahieu, C. (2017). RaiBlocks: A Feeless Distributed Cryptocurrency Network. `https://docs.nano.org/whitepaper/english/`. 54, 73

[Liu et al., 2022] Liu, J., Kandikuppa, A., and Bates, A. (2022). Transparent DIFC: harnessing innate application event logging for fine-grained decentralized information flow control. In *7th IEEE European Symposium on Security and Privacy, EuroS&P 2022, Genoa, Italy, June 6-10, 2022*, pages 487–501. IEEE. 18, 117, 118, 119, 132, 136

[Liu et al., 2023] Liu, Y., Wang, J., Yan, Z., Wan, Z., and Jäntti, R. (2023). A survey on blockchain-based trust management for internet of things. *IEEE Internet Things J.*, 10(7):5898–5922. 10, 33

[Lu et al., 2022] Lu, J., Sun, J., Xiao, R., and Jin, S. (2022). DIFCS: A secure cloud data sharing approach based on decentralized information flow control. *Comput. Secur.*, 117:102678. 130

[Lu et al., 2020] Lu, Y., Huang, X., Zhang, K., Maharjan, S., and Zhang, Y. (2020). Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Trans. Veh. Technol.*, 69(4):4298–4311. 140

[Lundbaek, 2020] Lundbaek, L. (2020). *Energy-efficient and decentralized access control: a framework for embedded systems and mobility*. PhD thesis, Imperial College London, UK. 77

[Ma et al., 2021] Ma, Z., Meng, J., Wang, J., and Shan, Z. (2021). Blockchain-based decentralized authentication modeling scheme in edge and iot environment. *IEEE Internet Things J.*, 8(4):2116–2123. 13, 16, 34, 48

[Ma et al., 2020] Ma, Z., Wang, L., Xiaochang, W., Wang, Z., and Zhao, W. (2020). Blockchain-enabled decentralized trust management and secure usage control of IoT big data. *IEEE Internet Things J.*, 7(5):4000–4015. 94

[Malkin, 2023] Malkin, N. (2023). Contextual integrity, explained: A more usable privacy definition. *IEEE Security & Privacy*, 21(1):58–65. 37

[Mallmann-Trenn, 2017] Mallmann-Trenn, F. (2017). *Probabilistic analysis of distributed processes with focus on consensus. (Analyse probabiliste de processus distribués axés sur les processus de consensus)*. PhD thesis, PSL Research University, Paris, France. 75

[Martinelli et al., 2019] Martinelli, F., Mori, P., Saracino, A., and Cerbo, F. D. (2019). Obligation management in usage control systems. In *27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2019, Pavia, Italy, February 13-15, 2019*, pages 356–364. IEEE. 44

[Merkle, 1987] Merkle, R. C. (1987). A digital signature based on a conventional encryption function. In Pomerance, C., editor, *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, volume 293 of *Lecture Notes in Computer Science*, pages 369–378. Springer. 51

[Moore, 2008] Moore, A. (2008). Defining privacy. *Journal of Social Philosophy*, 39. 36

[Myers and Liskov, 1997] Myers, A. C. and Liskov, B. (1997). A decentralized model for information flow control. In Banâtre, M., Levy, H. M., and Waite, W. M., editors, *Proceedings of the Sixteenth ACM Symposium on Operating System Principles, SOSP 1997, St. Malo, France, October 5-8, 1997*, pages 129–142. ACM. 18, 116, 117, 118, 157

[Nakamoto, 2009] Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at https://metzdowd.com*. 48, 51, 159

[Naresh et al., 2023] Naresh, V. S., Allavarpu, V. V. L. D., and Reddi, S. (2023). Blockchain IOTA sharding-based scalable secure group communication in large vanets. *IEEE Internet Things J.*, 10(6, March 15):5205–5213. 54

[National Institute of Standards and Technology, 2018] National Institute of Standards and Technology (2018). Nistir 8200: Interagency report on the status of international cybersecurity standardization for the internet of things (IoT). Technical report, U.S. Department of Commerce, National Institute of Standards and Technology. 36

[NISO, 2004] NISO (2004). Understanding metadata. 158

[Okegbile et al., 2022] Okegbile, S. D., Cai, J., and Alfa, A. S. (2022). Performance analysis of blockchain-enabled data-sharing scheme in cloud-edge computing-based IoT networks. *IEEE Internet Things J.*, 9(21):21520–21536. 14, 63

[Omolara et al., 2022] Omolara, A. E., Alabdulatif, A., Abiodun, O. I., Alawida, M., Alabdulatif, A., Alshoura, W. H., and Arshad, H. (2022). The internet of things security: A survey encompassing unexplored areas and new insights. *Comput. Secur.*, 112:102494. 10, 32

[Park and Sandhu, 2004] Park, J. and Sandhu, R. S. (2004). The UCON_{abc} usage control model. *ACM Trans. Inf. Syst. Secur.*, 7(1):128–174. 12, 26, 42, 43, 159

[Patil and Kobsa, 2009] Patil, S. and Kobsa, A. (2009). Privacy considerations in awareness systems: Designing with privacy in mind. In Markopoulos, P., de Ruyter, B. E. R., and Mackay, W. E., editors, *Awareness Systems - Advances in Theory, Methodology and Design*, Human-Computer Interaction Series, pages 187–206. Springer. 37

[Popov, 2017] Popov, S. (2017). The Tangle. `https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf`. 15, 54, 73, 74, 108, 158

[Popov, 2020] Popov, S. (2020). The Coordicide. `https://files.iota.org/papers/20200120_Coordicide_WP.pdf`. 74, 81, 96, 137

[Pretschner et al., 2011] Pretschner, A., Lovat, E., and Büchler, M. (2011). Representation-independent data usage control. In García-Alfaro, J., Navarro-Arribas, G., Cuppens-Boulahia, N., and di Vimercati, S. D. C., editors, *Data Privacy Management and Autonomous Spontaneus Security - 6th International Workshop, DPM 2011, and 4th International Workshop, SETOP 2011, Leuven, Belgium, September 15-16, 2011, Revised Selected Papers*, volume 7122 of *Lecture Notes in Computer Science*, pages 122–140. Springer. 17, 35, 116

[Qammar et al., 2023] Qammar, A., Karim, A., Ning, H., and Ding, J. (2023). Securing federated learning with blockchain: a systematic literature review. *Artif. Intell. Rev.*, 56(5):3951–3985. 20, 140

[Qin et al., 2020] Qin, X., Huang, Y., Yang, Z., and Li, X. (2020). A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing. *Journal of Systems Architecture*, page 101854. 10, 33

[Qu et al., 2023] Qu, Y., Uddin, M. P., Gan, C., Xiang, Y., Gao, L., and Yearwood, J. (2023). Blockchain-enabled federated learning: A survey. *ACM Comput. Surv.*, 55(4):70:1–70:35. 20, 140

[Quiniou and Debonneuil, 2019] Quiniou, M. and Debonneuil, C. (2019). *Glossary - Blockchain*. Chaire UNESCO - ITEN. 157, 158, 159

[Raghav et al., 2020] Raghav, Andola, N., Venkatesan, S., and Verma, S. (2020). PoE-WAL: A lightweight consensus mechanism for blockchain in IoT. *Pervasive and Mobile Computing*, 69:101291. 52, 65, 97

[Rajasekaran et al., 2023] Rajasekaran, A. S., Azees, M., Rajagopal, M., and Lorincz, J. (2023). Blockchain enabled anonymous privacy-preserving authentication scheme for internet of health things. *Sensors*, 23(1):240. 13, 48

[Rifi et al., 2017] Rifi, N., Rachkidi, E., Agoulmine, N., and Taher, N. C. (2017). Towards using blockchain technology for iot data access protection. In *17th IEEE International Conference on Ubiquitous Wireless Broadband, ICUWB 2017, Salamanca, Spain, September 12-15, 2017*, pages 1–5. IEEE. 13, 48

[Rizos et al., 2019] Rizos, A., Bastos, D., Saracino, A., and Martinelli, F. (2019). Distributed UCON in CoAP and MQTT Protocols. In *ESORICS Int. Workshops, CyberICPS, SECPRE, SPOSE, and ADIoT, Luxembourg*, volume 11980 of *LNCS*, pages 35–52. Springer. 26, 44, 45

[Sabt et al., 2015] Sabt, M., Achemlal, M., and Bouabdallah, A. (2015). Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64. 159

[Sadi et al., 2023] Sadi, A. A., Mazzocca, C., Melis, A., Montanari, R., Prandini, M., and Romandini, N. (2023). P-IOTA: A cloud-based geographically distributed threat alert system that leverages P4 and IOTA. *Sensors*, 23(6):2955. 73

[Salimitari et al., 2020] Salimitari, M., Chatterjee, M., and Fallah, Y. P. (2020). A Survey on Consensus Methods in Blockchain for Resource-constrained IoT Networks. *Internet of Things*, 11:100212. 10, 28, 33, 34, 52, 53, 63, 65, 66, 94, 97, 157, 158

[Sarfraz et al., 2019] Sarfraz, U., Alam, M., Zeadally, S., and Khan, A. (2019). Privacy aware IOTA ledger: Decentralized mixing and unlinkable IOTA transactions. *Comput. Networks*, 148:361–372. 26, 59, 60, 61, 78, 81, 109, 136

[Schultz and Liskov, 2013] Schultz, D. and Liskov, B. (2013). IFDB: Decentralized information flow control for databases. In *Proceedings of the 8th ACM European Conference on Computer Systems*, EuroSys '13, page 43–56, New York, NY, USA. Association for Computing Machinery. 130

[Seres et al., 2019] Seres, I. A., Nagy, D. A., Buckland, C., and Burcsi, P. (2019). Mixeth: Efficient, trustless coin mixing service for ethereum. In Danos, V., Herlihy, M., Potop-Butucaru, M., Prat, J., and Piergiovanni, S. T., editors, *International Conference on Blockchain Economics, Security and Protocols, Tokenomics 2019, May 6-7, 2019, Paris,*

France, volume 71 of *OASIcs*, pages 13:1–13:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. 59, 60

[Sharma et al., 2023] Sharma, A., Sharma, P., Bamotra, H., and Gaur, V. (2023). An extended approach to appraise electricity distribution and carbon footprint of bitcoin in a smart city. *Frontiers Big Data*, 6. 51

[Shi et al., 2021] Shi, N., Tang, B., Sandhu, R. S., and Li, Q. (2021). DUCE: distributed usage control enforcement for private data sharing in Internet of Things. In Barker, K. and Ghazinour, K., editors, *Data and Applications Security and Privacy XXXV - 35th Annual IFIP WG 11.3 Conference, DBSec 2021, Calgary, Canada, July 19-20, 2021, Proceedings*, volume 12840 of *Lecture Notes in Computer Science*, pages 278–290. Springer. 16, 34, 35, 43, 95

[Shirey, 2007] Shirey, R. W. (2007). Internet Security Glossary, Version 2. RFC 4949. 157

[Silva and Lima, 2021] Silva, L. F. D. and Lima, J. V. F. (2021). An evaluation of Cassandra NoSQL database on a low-power cluster. In *33rd International Symposium on Computer Architecture and High Performance Computing, SBAC-PAD 2021 Workshops, Belo Horizonte, Brazil, October 26-29, 2021*, pages 9–14. IEEE. 103

[Simões et al., 2021] Simões, J. E., Ferreira, E., Menasché, D. S., and Campos, C. A. V. (2021). Blockchain privacy through merge avoidance and mixing services: a hardness and an impossibility result. *SIGMETRICS Perform. Evaluation Rev.*, 48(4):8–11. 61

[Stoll et al., 2019] Stoll, C., Klaaßen, L., and Gallersdörfer, U. (2019). The carbon footprint of bitcoin. *Joule*, 3. 51

[Szabo, 1994] Szabo, N. (1994). Smart contracts. *Unpublished*. 49

[Tang et al., 2021] Tang, J., Shoemaker, H., Lerner, A., and Birrell, E. (2021). Defining privacy: How users interpret technical terms in privacy policiess. *Proc. Priv. Enhancing Technol.*, 2021(3):70–94. 36

[Tarlan et al., 2022] Tarlan, O., Safak, I., and Kalkan, K. (2022). DiBLIoT: A distributed blacklisting protocol for IoT device classification using the hashgraph consensus algorithm. In *International Conference on Information Networking, ICOIN 2022, Jeju-si, Republic of Korea, January 12-15, 2022*, pages 84–89. IEEE. 55

[Tennant, 2017] Tennant, L. (2017). Improving the Anonymity of the IOTA Cryptocurrency. `https://laurencetennant.com/papers/anonymity-iota.pdf`. 71

[The European Parliament and the Council of the European Union, 2018] The European Parliament and the Council of the European Union (2018). General Data Protection Regulation. http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679. 10, 32, 58, 88, 111, 158

[TradingView, 2022] TradingView (2022). Bitcoin (BTC), Ethereum (ETH) dominance - their market cap relative to the market cap of all other cryptocurrencies in the world - on November 15, 2022. https://www.statista.com/statistics/1269302/crypto-market-share/ Graph]. In Statista. 27, 163

[van Saberhagen, 2013] van Saberhagen, N. (2013). Cryptonote Monero Whitepaper. https://github.com/monero-project/research-lab/blob/master/whitepaper/whitepaper.pdf. 62

[van Steen et al., 2021] van Steen, M., Chien, A. A., and Eugster, P. (2021). The difficulty in scaling blockchains: A simple explanation. *CoRR*, abs/2103.01487. 64, 96

[Wang et al., 2022a] Wang, H., Chen, G., Zhang, Y., and Lin, Z. (2022a). Multi-certificate attacks against proof-of-elapsed-time and their countermeasures. In *29th Annual Network and Distributed System Security Symposium, NDSS 2022, San Diego, California, USA, April 24-28, 2022*. The Internet Society. 53

[Wang et al., 2022b] Wang, T., Wang, Q., Shen, Z., Jia, Z., and Shao, Z. (2022b). Understanding characteristics and system implications of DAG-based blockchain in IoT environments. *IEEE Internet Things J.*, 9(16):14478–14489. 108

[Warren and Brandeis, 1890] Warren, S. D. and Brandeis, L. D. (1890). The right to privacy. *Harvard Law Review*, 4(5):193–220. 37

[Wen et al., 2021] Wen, Y., Lu, F., Liu, Y., and Huang, X. (2021). Attacks and countermeasures on blockchains: A survey from layering perspective. *Comput. Networks*, 191:107978. 49

[Westin and Solove, 1968] Westin, A. and Solove, D. (1968). *Privacy and Fzreedom*. Ig Publishing. 37

[Wüchner and Pretschner, 2012] Wüchner, T. and Pretschner, A. (2012). Data loss prevention based on data-driven usage control. In *23rd IEEE International Symposium on Software Reliability Engineering, ISSRE 2012, Dallas, TX, USA, November 27-30, 2012*, pages 151–160. IEEE Computer Society. 122

[Wuyts et al., 2018] Wuyts, K., Landuyt, D. V., Hovsepyan, A., and Joosen, W. (2018). Effective and efficient privacy threat modeling through domain refinements. In Haddad, H. M., Wainwright, R. L., and Chbeir, R., editors, *Proceedings of the 33rd An-*

nual ACM Symposium on Applied Computing, SAC 2018, Pau, France, April 09-13, 2018, pages 1175–1178. ACM. 16, 17, 37, 57, 87, 88

[Xu et al., 2020] Xu, D., Shrestha, R., and Shen, N. (2020). Automated strong mutation testing of XACML policies. In Lobo, J., Stoller, S. D., and Liu, P., editors, *Proceedings of the 25th ACM Symposium on Access Control Models and Technologies, SACMAT 2020, Barcelona, Spain, June 10-12, 2020*, pages 105–116. ACM. 20, 139

[Yarger, 2020] Yarger, M. (2020). The case for a unified identity. our vision for a unified identity protocol on the tangle for things, organizations,and individuals. `https://files.iota.org/comms/IOTA_The_Case_for_a_Unified_Identity.pdf`. 21, 141

[Yuan and Wang, 2018] Yuan, Y. and Wang, F. (2018). Blockchain and cryptocurrencies: Model, techniques, and applications. *IEEE Trans. Syst. Man Cybern. Syst.*, 48(9):1421–1428. 26, 49, 50

[Zhan et al., 2021] Zhan, Y., Wang, B., Lu, R., and Yu, Y. (2021). DRBFT: delegated randomization byzantine fault tolerance consensus protocol for blockchains. *Inf. Sci.*, 559:8–21. 53

[Zhang et al., 2020] Zhang, W. E., Sheng, Q. Z., Mahmood, A., Tran, D. H., Zaib, M., Hamad, S. A., Aljubairy, A., Alhazmi, A. A. F., Sagar, S., and Ma, C. (2020). The 10 research topics in the Internet of Things. In *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, pages 34–43. 36

[Zhang et al., 2022] Zhang, X., Li, X., Miao, Y., Luo, X., Wang, Y., Ma, S., and Weng, J. (2022). A data trading scheme with efficient data usage control for industrial IoT. *IEEE Trans. Ind. Informatics*, 18(7):4456–4465. 34, 94, 95

[Zheng and Myers, 2007] Zheng, L. and Myers, A. C. (2007). Dynamic security labels and static information flow control. *Int. J. Inf. Sec.*, 6(2-3):67–84. 46

Appendix A

Glossary

DApp: DApps are open-source, decentralized applications that can operate autonomously and without human intervention. DApps make use of cryptocurrencies or tokens, are executed in a network of computers and store outputs in public ledgers [Andoni et al., 2018];

DIFC: Decentralized information flow control (DIFC) allows users to control the flow of their information without imposing the rigid constraints of a traditional multilevel security system [Myers and Liskov, 1997];

DLT: Distributed Ledger Technology. A distributed ledger is a register containing a set of transactions. This ledger instead of being stored in a single place, the central server, is duplicated on a set of machines. The fact that the ledger is copied identically multiple times makes it expensive to modify it: it requires to be changed on every node of the network. [Quiniou and Debonneuil, 2019];

DPoS: Delegated Proof of Stake (DPoS) is a variant of PoS in which "all the stakeholders vote to choose some nodes as witnesses and delegates. Witnesses are responsible and rewarded for creating new blocks. The delegates are responsible for maintaining the network and proposing changes such as block sizes, transaction fees, or reward amount" [Salimitari et al., 2020];

Hashgraph: Hedera Hashgraph is a distributed ledger technology that has a new form of distributed consensus [Baird et al., 2018]. It provides fast and ordered transactions, as well as secure infrastructure to run decentralized applications;

IFC: Information Flow Control (IFC) [Denning, 1976] is "a concept requiring that information transfers within a system be controlled so that information in certain types of objects cannot, via any channel within the system, flow to certain other types of objects" [Shirey, 2007] ;

IIoT: Industrial Internet of Things, refers to the use of IoT technologies in the manufacturing industry [Boyes et al., 2018];

IoT: Internet of Things, the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems [Berte, 2018]. It is to be noted that the Internet of Things has manifold definitions [Atzori et al., 2010];

IOTA: IOTA is a cryptocurrency for the Internet-of-Things (IoT) industry. The main feature of this novel cryptocurrency is the tangle, a directed acyclic graph (DAG) for storing transactions. It offers features that are required to establish a machine- to-machine micropayment system [Popov, 2017].

GDPR: European *General Data Protection Regulation* [The European Parliament and the Council of the European Union, 2018];

Metadata: Metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Metadata is often called data about data or information about information [NISO, 2004];

PoA: The proof of authority (PoA) is a consensus algorithm used most of the time for private blockchains. It makes it possible to designate nodes of the network as validators, these nodes having the role of determining the state of the ledger for the entire network [Quiniou and Debonneuil, 2019]. It significantly increases the speed of transaction validations, even if centralization is increased [Quiniou and Debonneuil, 2019];

PBFT: In Practical byzantine fault tolerance (PBFT), all the nodes should participate in the voting process in order to add the next block and the consensus is reached when more than two-thirds of all nodes agree upon that block. PBFT can tolerate malicious behavior from up to one-third of all nodes to perform normally [Salimitari et al., 2020];

PoET: Proof of elapsed time (PoET) is a consensus method proposed by Intel that works similarly to PoW but with significantly lower energy consumption. In this method, miners have to solve a hash problem similar to that of PoW. However, instead of a competition between miners to solve the next block, the winning miner is randomly chosen based on a random wait time. The winning miner is the one whose timer expires first [Salimitari et al., 2020];

PoS: the proof of stake (PoS) is an alternative consensus algorithm to the proof of work which provides the right to create the next block to an active validator on the network that has deposited units of the cryptocurrency of this blockchain. The proof of stake is less energy intensive than proof of work [Quiniou and Debonneuil, 2019];

PoW: Proof of Work is a computation race taking the form of a cryptographic puzzle, used by the Bitcoin cryptocurrency to elect the node responsible for writing the next transaction on the ledger [Nakamoto, 2009];

TEE: a Trusted Execution Environment (TEE) is a tamper-resistant processing environment that runs on a separation kernel. It guarantees the authenticity of the executed code, the integrity of the runtime states (e.g. CPU registers, memory and sensitive I/O), and the confidentiality of its code, data and runtime states stored on a persistent memory [Sabt et al., 2015];

UCON is a model for data access control that allows for dynamic decision-making based on attributes such as context, time, and user behavior [Park and Sandhu, 2004];

UCS: Usage Control System, an entity responsible for monitoring the data usage and dissemination, according to user-defined policies;

XACML: The eXtensible Access Control Markup Language (XACML) is an XML-based standard markup language for specifying access control policies. The standard, published by OASIS, defines a declarative fine-grained, attribute-based access control policy language, an architecture, and a processing model describing how to evaluate access requests according to the rules defined in policies [Axiomatics, 2023].

Appendix B

Publications

Conferences and journals

N. Denis, M. Laurent and S. Chabridon, *Integrating Usage Control into Distributed Ledger Technology for Internet of Things Privacy* in IEEE Internet of Things Journal, `https://doi.org/10.1109/JIOT.2023.3283300`, June 2023;

N. Denis, S. Chabridon, M. Laurent, *Bringing Privacy, Security and Performance to the Internet of Things through Usage Control and Blockchains*, IFIP Summer School on Privacy and Identity Management 2021, IFIP Advanced in Information and Communication Technologies Series (Eds. I. Schiering, M. Friedewald, S. Krenn, S. Schiffner), `https://doi.org/10.1007/978-3-030-99100-5_6`, March 2022.

Presentations

N. Denis, M. Laurent, S. Chabridon, *A Decentralized Model for Usage and Information Flow Control in Distributed Systems*, Atelier sur la Protection de la Vie Privée (APVP 2023), Bourgogne Franche-Comté, France, June 12-15, 2023;

N. Denis, M. Laurent, S. Chabridon, *Integrating Usage Control into Distributed Ledger Technology for Internet of Things Privacy*, Seminar Thales (french company), Palaiseau, France, June 8, 2023;

N. Denis, M. Laurent, S. Chabridon, *Integrating Usage Control into Distributed Ledger Technology for Internet of Things Privacy*, Seminar Cybersecurity on a Plate (CoAP), Palaiseau, France, April 18, 2023;

N. Denis, S. Chabridon, M. Laurent, *Bringing Privacy, Security and Performance to the Internet of Things using IOTA and Usage Control*, Atelier sur la Protection de la Vie Privée (APVP 2022), Châtenay-sur-Seine, France, June 13-16, 2022;

N. Denis, S. Chabridon, M. Laurent, *Bringing Privacy, Security and Performance to the Internet of Things using IOTA and Usage Control*, Seminar ACMES, Lisses, France, December 13, 2021;

N. Denis, S. Chabridon, M. Laurent, *Internet des Objets: Conjuguer Sécurité, Protection de la Vie Privée et Performances*, Atelier sur la Protection de la Vie Privée (APVP 2021), Remote presentation, June 15-17, 2021.

Appendix C

Cryptocurrencies

C.1 General data

Figure C.1 depicts the percentage of market capitalization for major cryptocurrencies in relation to the total market capitalization of the cryptocurrency market. The figure showcases the prominent dominance of Bitcoin and Ethereum, which collectively account for nearly 60% of the cryptocurrency market. The third most important cryptocurrency in market capitalization, Tether (USDT), is a *stable coin* following the value of the US dollar and only accounts for 8% of the whole market.

C.2 Energy consumption

Cryptocurrencies based on proof of work, such as BTC, are known for their high electricity consumption. Elements of comparison are provided in this section to quantify the actual energy consumption. From the first figure (Figure C.2), it appears that Bitcoin is consuming more electricity for one transaction, than the Visa network *for 400.000 transactions*. The second figure (Figure C.3) highlights the energy consumption of BTC compared to entire countries. Notably, BTC consumes more electricity yearly, that Czechia alone.

C.3 Storage

The Bitcoin blockchain has grown significantly since its inception to 475 GB as of June 8, 2023, and it continues to grow as new transactions are added to the network. Figure C.4 shows the evolution of Bitcoin's ledger size from 2009 to June, 2023. This continuous growth poses challenges for storage and synchronization for network participants. Running a full node, which stores a complete copy of the Bitcoin blockchain, requires

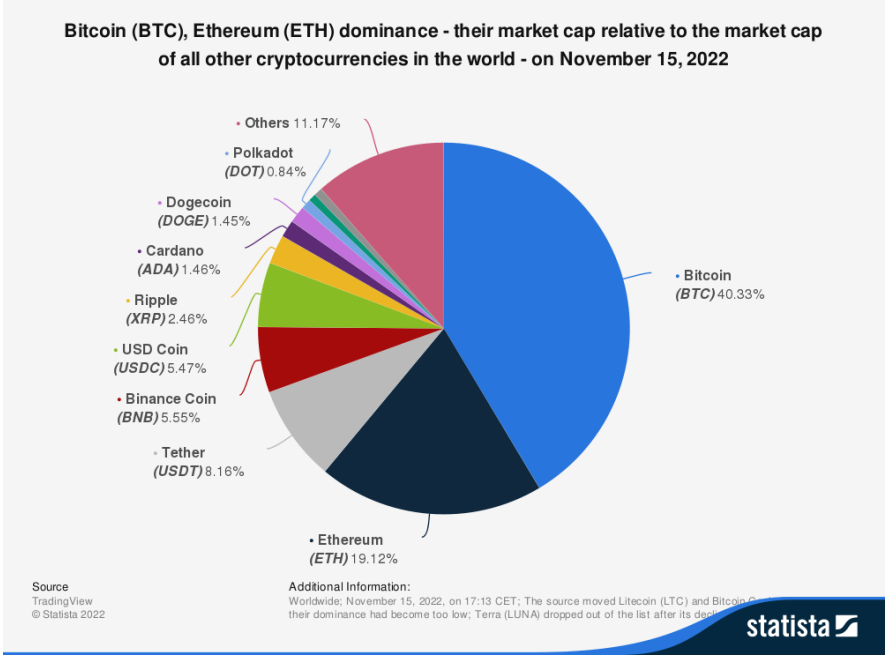


Figure C.1 – Bitcoin (BTC), Ethereum (ETH) dominance - their market cap relative to the market cap of all other cryptocurrencies in the world - on November 15, 2022 - Statista [TradingView, 2022]

substantial storage space. In addition to the blockchain data itself, a full node also needs to store additional data, such as transaction indexes and the UTXO set. As of now, a full node typically requires several terabytes of storage space.

Pruning. Bitcoin introduced a feature called pruning to address the storage requirements of running a full node. Pruning allows nodes to discard older blockchain data while still maintaining the ability to verify new transactions. This reduces storage requirements for nodes that do not need to maintain a complete history of all transactions.

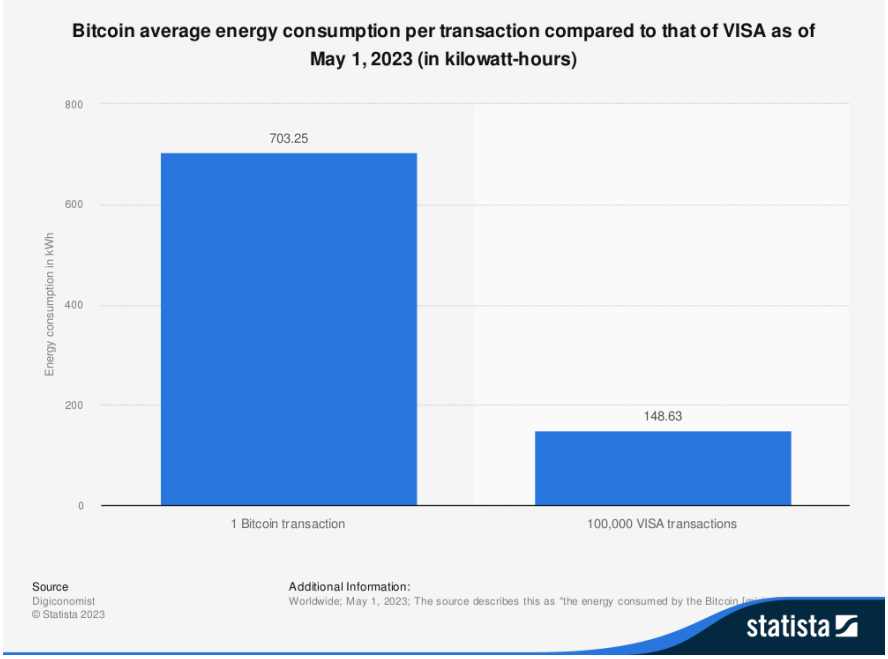


Figure C.2 – Bitcoin average energy consumption per transaction compared to that of VISA as of May 1, 2023 in kilowatt-hours - Statista [Digiconomist, 2023]

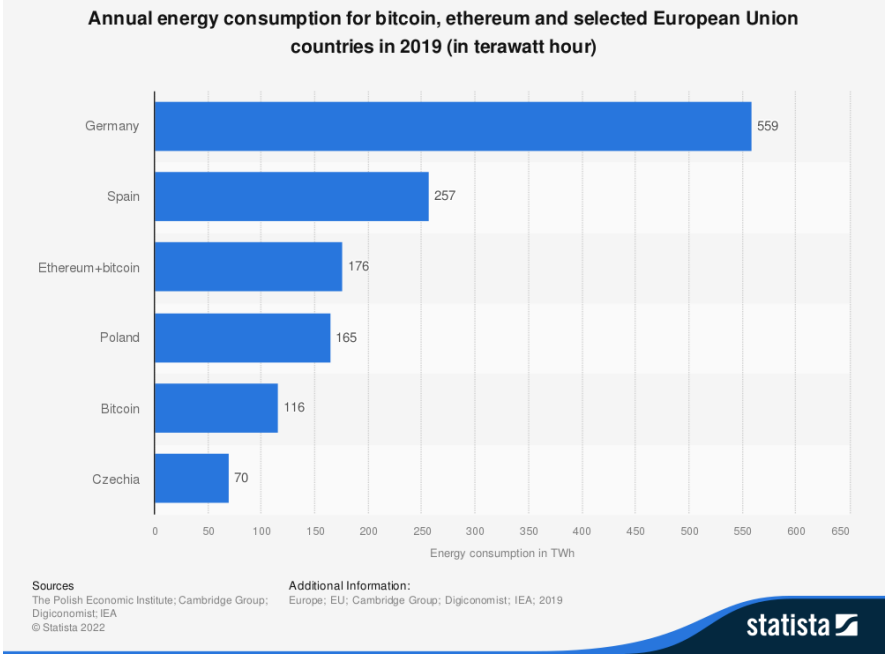


Figure C.3 – Annual energy consumption for bitcoin, ethereum and selected European Union countries in 2019 in terawatt hour [Institute, 2021]

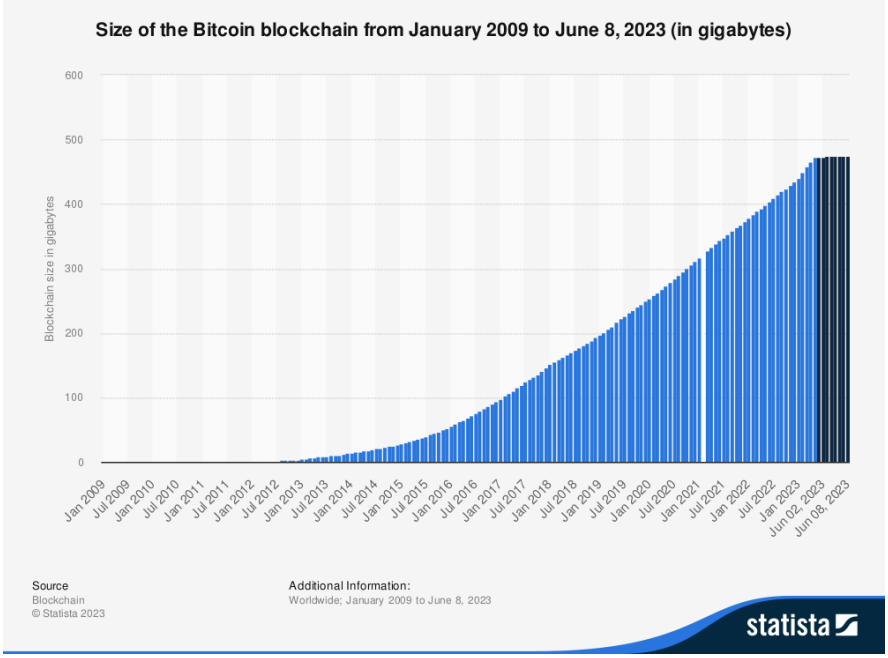


Figure C.4 – Size of the Bitcoin blockchain from January 2009 to June 8, 2023(in gigabytes) [Blockchain.info, 2023]

Appendix D

Usage control

The provided XACML policy (cf. Listing D.1) checks if a data buyer whose address is **atoi1qztx22lp5n69f4etlr6600qcdswrj8terzkga65msddc3sh4sdx02rxjqsq** paid the right amount of cryptocurrency (1000000) to the data provider **atoi1qpty45svr0r5s9nxatzas-zkde8syt4etrsmea50pmc3l2swyskc0q8wlpjs**

Listing D.1 – XACML policy

```
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os" PolicyId="
    HumidityPolicy" RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-
    combining-algorithm:permit-overrides" Version="1.0">
<Target><AnyOf><AllOf>
    <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
        atoi1qztx22lp5n69f4etlr6600qcdswrj8terzkga65msddc3sh4sdx02rxjqsq</
        AttributeValue>
    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:
        subject-id" Category="urn:oasis:names:tc:xacml:1.0:subject-category:
        access-subject" DataType="http://www.w3.org/2001/XMLSchema#string"
        MustBePresent="true"/>
</Match>
</AllOf></AnyOf></Target>
<Rule Effect="Permit" RuleId="check-amount">
<Description>Check if transaction value matches the data value</Description>
<Target><AnyOf><AllOf>
    <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer"
        >1000000</AttributeValue>
    <AttributeDesignator AttributeId="http://iotawucon/nbiotas" Category="http
        ://iotawucon/category" DataType="http://www.w3.org/2001/XMLSchema#string
        " MustBePresent="true"/>
    </Match></AllOf></AnyOf></Target>
```

```
15      <AdviceExpressions><AdviceExpression AdviceId="permit-amount" AppliesTo="
            Permit">
16          <AttributeAssignmentExpression AttributeId="urn:oasis:names:tc:xacml
                :2.0:example:attribute:text">
17          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">The
                amount of iotas paid is correct</AttributeValue>
18      </AttributeAssignmentExpression></AdviceExpression></AdviceExpressions>
19  </Rule>
20  <Rule Effect="Permit" RuleId="check-owner">
21      <Description>Check if the data owner is the transaction destination</
            Description>
22      <Target><AnyOf><AllOf>
23         <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:integer-equal">
24         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
               atoi1qpty45svr0r5s9nxatzaszkde8syt4etrsmea50pmc3l2swyskc0q8wlpjs</
               AttributeValue>
25         <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:
               subject-id" Category="urn:oasis:names:tc:xacml:1.0:subject-category:
               access-subject" DataType="http://www.w3.org/2001/XMLSchema#string"
               MustBePresent="true"/>
26         </Match></AllOf></AnyOf></Target>
27  </Rule>
28  <Rule Effect="Permit" RuleId="no-dissemination" Fulfill-phase="ongoing-access">
29      <Description> Check if data is disseminated during access </Description>
30      <Target> <Subjects><AnySubject/></Subjects>
31      <Resources><AnyResource/></Resources>
32          <Action> <ActionMatch MatchId="equal">
33          <AttrValue>Dissemination</AttrValue>
34          <ActionAttrDesignator AttrId="action-diss"/>
35          </ActionMatch></Action></Target>
36  </Rule>
37   <Rule RuleId="deny-rule" Effect="Deny"/>
38  </Policy>
```

The XACML request (Listing D.2)

Listing D.2 – XACML request

```
1       public static String createSimpleXACMLrequest(int nbIotas, String
            userAddress) {
2           return "<Request xmlns=\"urn:oasis:names:tc:xacml:3.0:core:schema:wd
                -17\" CombinedDecision=\"false\" ReturnPolicyIdList=\"false\">\n"+
3                   "<Attributes Category=\"urn:oasis:names:tc:xacml:1.0:subject-
                        category:access-subject\">\n" +
4                   "<Attribute AttributeId=\"urn:oasis:names:tc:xacml:1.0:subject:
                        subject-id\" IncludeInResult=\"true\">\n" +
```

```
                "<AttributeValue DataType=\"http://www.w3.org/2001/XMLSchema#
                    string\">" + userAddress +"</AttributeValue>\n" +
                "</Attribute>\n" +
                "</Attributes>\n" +
                "<Attributes Category=\"http://iotawucon/category\">\n" +
                "<Attribute AttributeId=\"http://iotawucon/nbiotas\"
                    IncludeInResult=\"true\">\n" +
                "<AttributeValue DataType=\"http://www.w3.org/2001/XMLSchema#
                    integer\">" + nbIotas + "</AttributeValue>\n" +
                "</Attribute>\n" +
                "</Attributes>\n" +
                "</Request>";
    }
```

Titre: Pour un internet des objets sécurisé et respectueux de la vie privée basé sur le contrôle d'usage et les registres distribués

Mots clés: Internet des Objets, Systèmes distribués, Vie privée, Contrôle d'usage, RGPD

Résumé: Les objets connectés représentent l'une des principales cibles de la cybercriminalité. Les raisons en sont multiples : d'abord, pour des raisons commerciales, les fabricants peuvent vendre des produits vulnérables qui posent des problèmes de sécurité. Deuxièmement, de nombreux appareils IoT sont soumis à des contraintes de performance et ne disposent pas de la puissance nécessaire pour exécuter des logiciels de sécurité. Enfin, l'hétérogénéité des applications, du matériel et des logiciels élargit la surface d'attaque. Pour parer à ces menaces, l'IoT a besoin de technologies de sécurité et de préservation de la vie privée sur mesure.
En ce qui concerne la protection de la vie privée, *le contrôle d'usage* donne aux utilisateurs la possibilité de spécifier comment leurs données peuvent être utilisées et par qui. Le contrôle d'usage étend le contrôle d'accès classique en introduisant des *obligations*, c'est-à-dire des actions à effectuer pour obtenir l'accès, et des *conditions* qui sont liées à l'état du système, comme la charge du réseau ou le temps. Cette thèse vise à apporter des réponses aux défis de l'internet des objets en termes de performance, de sécurité et de respect de la vie privée. Pour cela, les registres distribués (DLT) constituent une solution prometteuse aux contraintes de l'internet des objets, en particulier pour les micro-transactions, notamment par leur caractère décentralisé. Cela se traduit par trois contributions: 1. un ensemble de technologies pour des transactions sans frais préservant la vie privée, conçu pour passer à l'échelle; 2. une méthode d'intégration du contrôle de l'utilisation et des registres distribués pour permettre une protection efficace des données des utilisateurs; 3. un modèle étendu pour le contrôle d'usage dans les systèmes distribués, afin d'y ajouter le contrôle de flux décentralisé et les aspects liés à l'internet des objets. Une preuve de concept de l'intégration (2) a été mise en place pour démontrer la faisabilité et effectuer des tests de performance. Il s'appuie sur IOTA, un registre distribué qui utilise un graphe orienté acyclique pour son graphe de transactions au lieu d'une *blockchain*. Les résultats des tests de performance sur un réseau privé montrent une diminution d'environ 90% du temps nécessaire pour effectuer des transactions et pour évaluer des politiques de contrôle d'usage, dans le cas où ce dernier est intégré au réseau.

Title: For a Private and Secure Internet of Things with Usage Control and Distributed Ledger Technology

Keywords: Internet of Things, Distributed Systems, Privacy, Usage Control, GDPR

Abstract:
IoT devices represent one of the major targets for malicious activities. The grounds for this are manifold: first, to reduce the cost of security, manufacturers may sell vulnerable products, leaving users with security concerns. Second, many IoT devices have performance constraints and lack the processing power to execute security software. Third, the heterogeneity of applications, hardware, and software widens the attack surface. As a result, IoT networks are subject to a variety of cyber threats. To counter such a variety of attacks, the IoT calls for security and privacy-preserving technologies. For privacy concerns, *usage control* grants the users the power to specify how their data can be used and by whom. Usage control extends classic access control by introducing *obligations*, i.e., actions to be performed to be granted access, and *conditions* that are related to the system state, such as the network load or the time.
This thesis aims at providing answers to the challenges in the Internet of Things in terms of performance, security and privacy. To this end, *distributed ledger technologies* (DLTs) are a promising solution to Internet of Things constraints, in particular for micro-transactions, due to the decentralization they provide. This leads to three related contributions: 1. a framework for zero-fee privacy-preserving transactions in the Internet of Things designed to be scalable; 2. an integration methodology of usage control and distributed ledgers to enable efficient protection of users' data; 3. an extended model for data usage control in distributed systems, to incorporate decentralized information flow control and IoT aspects. A proof of concept of the integration (2) has been designed to demonstrate feasibility and conduct performance tests. It is based on IOTA, a distributed ledger using a directed acyclic graph for its transaction graph instead of a blockchain. The results of the tests on a private network show an approximate 90% decrease in the time needed to push transactions and make access decisions in the integrated setting.

Institut Polytechnique de Paris
91120 Palaiseau, France