Owner-centric sharing of physical resources, data, and data-driven insights in digital ecosystems

Kwok Cheung
kwok@extropy.io
Extropy Ltd.
Oxford, UK

Michael Huth
michael.huth@xain.io
XAIN AG & Imperial College London
London, UK

Laurence Kirk
laurence@extropy.io
Extropy Ltd.
Oxford, UK

Leif-Nissen Lundbæk
leif@xain.io
XAIN AG
Berlin, Germany

Rodolphe Marques
rodolphe.marques@xain.io
XAIN AG
Berlin, Germany

Jan Petsche
jan.petsche@xain.io
XAIN AG
Berlin, Germany

ABSTRACT

We are living in an age in which digitization will connect more and more physical assets with IT systems and where IoT endpoints will generate a wealth of valuable data. Companies, individual users, and organizations alike therefore have the need to control their own physical or non-physical assets and data sources. At the same time, they recognize the need for, and opportunity to, share access to such data and digitized physical assets. This paper sets out our technology vision for such sharing ecosystems, reports initial work in that direction, identifies challenges for realizing this vision, and seeks feedback and collaboration from the academic access-control community in that R&D space.

CCS CONCEPTS

• **Security and privacy** → **Human and societal aspects of security and privacy**; *Access control*; • **Social and professional topics** → *Computing / technology policy*; • **Computer systems organization** → *Embedded and cyber-physical systems*.

KEYWORDS

FROST Language, Policy Life Cycle, Software Development Kit, Belnap Logic, Distributed Ledger

1 INTRODUCTION

There is little doubt that the increased digitization of our physical and social world has already had profound impact – economically,

socially, and technically. The adoption rate of smart-phones and their increased capabilities and richness of user interactions through apps are just one source of evidence for this claim. Digitization is also changing the manufacturing plants and their processes, vehicle function and interaction with the environment, and general consumer expectations of service provisions. For the latter, progressive web apps are an example of how today's users wish to interact with products and services – where the ability of an app to adapt to user-specific interaction history and needs is seen as a competitive advantage.

The increased connectedness that this digitization of our lives brings, and its shift from product purchase to user-centric service consumption drives, in part, the creation of nascent sharing economies. For example, throughout the globe there are a host of providers for e-bikes, e-scooters, and cars operating locally where clients can use these resources on demand for flexible periods of time. And with the growth rate of sales of new cars predicted to fall to about 2 percent globally by 2030 [32], Original Equipment Manufacturers (OEM) are incentivized to find means of increasing the monetary value of a manufactured vehicle. One way in which an OEM can do this is to not sell the car at all, but rather operate a service by which the car can be used by many, thus increasing the active usage time and the revenue stream generated by usage of the vehicle.

The sharing economy is not limited to the consideration of Business-To-Consumer (B2C) business models. A decentralized app for accommodation sharing, e.g., may enable Consumer-To-Consumer (C2C) business models for the sharing of rooms, apartments or other facilities such that clients can directly formulate the access conditions to their own data and physical resources without a central organization having already access to such data and resources.[1] Such an approach requires, though, that clients as owners of physical resources can freely delegate the access of their own resources to others – under specific conditions that typically include or trigger financial transactions.

Of course, the sharing economy is neither confined to mobility and transportation nor to B2C and C2C business models. A typical Business-To-Business (B2B) case is found in ecosystems in which companies already have established working relationships

---

[1]The governance of this platform would be exercised by some parties that would hold power over running that platform, but that should not enable these parties to circumvent access controls that operate within that platform itself.

and work-flows across organizations. Generally, such an ecosystem lacks the technical means of judiciously sharing access to physical resources, data or data-driven insights *across* organizational boundaries in a digital, fine-grained, and (semi)-automated manner.

This gap presents a business opportunity as the provision of such digital sharing processes can reduce cost (e.g. removal of manual data entry on the receiving end), improve security (e.g. prevention of ad-hoc data sharing such as through email attachments), and create economic value (e.g. by leasing under-utilized physical resources).

For example, a company that operates train infrastructure may want to share some data sets with a train manufacturer, and share different data sets and access to physical devices deployed on tracks with a service company that maintains these tracks. But even if these parties had already understood the specifics of what should be shared with whom under which circumstances, they would have neither the technical means of operationalizing this understanding, nor of verifying the trustworthiness of such an operationalization.

It is clear that a technical solution to this needs to provide sufficient interoperability. Non-technical issues such as social, regulatory or cognitive ones also may need to be overcome; e.g., company culture may make it hard to contemplate that there is value in sharing data or resources with other, perhaps competing, companies. Regulation may prevent such sharing or put demands on its operationalization and auditing. And understanding what should be shared and exactly how and with whom can create cognitive complexity that needs to be managed – regardless of what specification language is being used for access controls.

The engagement with our clients and potential customers has made us realize that there is a definite need for technology that can provide such owner-centric means of sharing physical resources, data, and data-driven insights such that participating parties can agree on the specifics of their sharing topology and its controls. The realization of the vision we offer in this paper faces several challenges, some of which we will highlight subsequently. The intent of this paper is to share this vision and its challenges in order to gain feedback and to encourage collaboration on the needed research and development (R&D) in order to make this vision a reality.

Outline of Paper. Our vision and its guiding principles are the subject of Section 2. Our work done so far towards realizing this vision is discussed in Section 3. In Section 4, we explore some challenges that attaining this vision faces. Section 5 details research topics in which we think the academic access-control community can make vital contributions for our vision of sharing ecosystems. Representative related work is summarized in Section 6 and the paper concludes in Section 7.

2 OUR VISION AND ITS PRINCIPLES

We now discuss our vision by formulating its guiding principles and technical ambition, while also explaining the values that determine these principles.

First, we observe that there is an apparent tension between centralization of service provision and genuine control of owners over their digital or physical assets. This can be seen, e.g., in the manner in which big tech companies such as Facebook, Apple,

Amazon, Netflix, and Google give their users only limited control over data generated by them.

However, it is the ability to control access and to manage the life cycle of such data that amounts to what most of us would understand to be *data ownership.* In UK law, e.g., there seems to be nothing on the Statute Books that refers to data *ownership* directly but rather to the *control* of data. Indeed, it is unclear whether users, owners or manufacturers (say of a car) could *own* a piece of data from that asset – say under current EU law. It is therefore conceivable that Internet of Things (IoT) data may have multiple owners. Moreover, future owners may neither be human subjects nor legal entities – e.g. autonomous machines.

Our first principle assumes that such legal clarity of ownership and data creation will be established in the future:

PRINCIPLE 1 (OWNER-CENTRIC DATA HANDLING). *The creators of data should be in genuine control over their data.*

Note that this principle does not rule out that data owners may give away their own data or even control over it – be it for free as a social good, for money or for some other motive. But it does mean that the owners of such data, e.g. of sensor data from a vehicle, are in full control of that data as soon as it gets created – and that these owners can decide to delegate to others access of such data. We emphasize that this applies not only to owners as individuals but equally when the owner is a legal entity such as a limited company that hosts its data in a cloud environment but on premise.

An owner-centric approach also places great importance onto data privacy: data controls should default to using access-control policies that protect the user, and where deviations from defaults are at the discretion of data owners.

For *physical* assets, it is often said that "possession is nine-tenths of the law". We believe that this expression will need to be revised in the digital future, since most physical assets of value will have micro-controllers embedded in them that guard access to those assets and their functions based on digital credentials. This leads us to the following principle:

PRINCIPLE 2 (OWNER-CENTRIC ACCESS TO PHYSICAL ASSETS). *The owners of physical assets should be in control over these assets and over the ability to delegate such access controls.*

This principle seems like a truism. But it should also apply when the owner is not in physical proximity of the asset and when the owner is a legal entity (e.g. as might be the case for a car that "owns itself"). The delegation mechanism stated in this principle gives rise to social and technical complexity, manifesting itself in two types: delegation of

- basic access rights such as "my daughter may use my Porsche on weekends before 8pm",
- specifying entire access-control policies, e.g. when the OEM would say that "this car dealer may install and run an access-control policy on that vehicle as well".

Transfer of ownership may be seen as a special case of that second type of delegation. In the future, the OEM will certainly want to maintain a numerical passport of the vehicle it manufactured, e.g. in order to do predictive maintenance and to improve future versions of that particular car model. But such access to data needs

to be consistent with whatever policy the car dealer, e.g. as legal owner of the car, would want to put onto the vehicle. Similarly, the policy of a leaseholder (to whom the car dealer may have delegated the right to install such a policy on the vehicle) would need to be consistent with that of the car dealer and with the one of the OEM. This leads us to another principle:

PRINCIPLE 3 (PROGRAMMABLE SHARING ECOSYSTEM). *Technology should enable the expression of delegation and access relationships that are consistent with each other and reflect the needs of owners and consumers of data and other assets.*

Let us explore this principle in a case in which two companies (X and Y) want to share access to data sets about driving behavior over their own vehicle models. They have an incentive to share the knowledge that resides within such data sets, e.g. in order to improve the quality of their machine-learning algorithms for semi-automatic driving.

Yet companies X and Y will never agree to actually share these data sets as in a "may read/may copy" modality that would move such data onto the other company's premises. But they may agree to running a machine-learning algorithm on their own data sets on behalf of the other company and to then report back the learned mathematical model. The principle says that the technology should allow these companies to leave their data sets on premise, to run machine-learning algorithms from the other company on their own data sets, and to have a fair way of reporting those results back.

Fairness may amount to selling the model for an adequate price, to using an escrow service if both companies expect to receive models from the other company's data sets or to relying on some other mutually agreed upon trust mechanism. We believe that the *"may train"* modality for this use case of data access – which is weaker than "write" and "read" – is of independent interest for the access-control community.

This principle also applies when actors share a physical resource, as our use case of the OEM, the car dealer, and the leaseholder of a vehicle illustrated. In that use case, the principle assures that these stakeholders have the technological means of mapping their social or contractual understanding of access to vehicle data and function consistently onto the vehicle technology and its communication interfaces.

That use case also demonstrates that more data will be generated from IoT endpoints, and more computation will be pushed to IoT edges. This technology trend, and general security considerations, lead us to our next principle:

PRINCIPLE 4 (PHYSICAL PROXIMITY OF ACCESS-CONTROL). *Access-control, including the verification of both delegation and change management of controls, should be enforced close to or embedded into the assets that are subject to these controls.*

Vehicles would thus compute and enforce access-control decisions within their internal vehicle systems, potentially also with offline capabilities for communicating and verifying policy state. And enterprises may host a server that provides similar functionality in controlling access to local data sets and the reporting of data-driven insights to external parties.

From a business perspective, there is an incentive to realize such access controls with the same software for the micro-controller and the server environments; this would decrease development costs, would ease maintenance, and so forth. We therefore formulate a principle rooted in considerations of code assurance and economics:

PRINCIPLE 5 (SOFTWARE IS EXECUTABLE ON IOT AND ICT). *Software that can reliably realize such access-control for both micro-controllers and servers has a competitive advantage.*

The importance of this principle should not be underestimated; e.g., it rules out the use of eXtensible Markup Language (XML) technology for building such software, since XML is not supported or too demanding in most micro-controller environments. It also suggests that a balance needs to be struck between the expressiveness of programming languages running on servers and the restrictions in terms of storage and available technology stacks and toolings found on commercial micro-controllers.

The access-control software we envision serves as an abstraction layer on top of existing controls, e.g. those within a vehicle network or in an enterprise network. Therefore, this principle is consistent with the use of server-side technology such as eXtensible Access Control Markup Language (XACML) at lower layers of abstraction.

The use cases we described so far, Company X/Company Y and OEM/Car Dealer/Leaseholder, may be interpreted as closed systems in and of themselves. We prefer to call them *biotopes* that may choose to connect with other such sharing biotopes within larger sharing ecosystems, e.g. emerging data marketplaces. From a technology perspective, this means that we also want to accommodate *open* sharing systems. For example, a company that manufactures batteries may want to share battery usage data with a provider of electric charging stations for e-scooters, and e-scooter users may share their location data with advertisers for free battery charging – should they deem this deal to be acceptable.

PRINCIPLE 6 (OPEN SHARING ECOSYSTEMS). *Sharing systems can connect to other sharing systems in an open network with trustworthy consensus mechanism.*

We believe that decentralized consensus architectures as found in Distributed Ledger Technology (DLT) have good potential in this regard: important system state needs to have consensus in order to make the system function reliably (under faults) and resiliently (under attacks from active adversaries). Known consensus algorithms provide such reliability and resiliency at high cost: they may not scale to desired network or transaction size (e.g. Byzantine algorithms such as PBFT [8]), they may be proprietary (e.g. Hashgraph at hedera.com) or they may be wasting energy (e.g. Proof of Work [17, 33]). This leads us to another principle:

PRINCIPLE 7 (SUSTAINABLE TECHNOLOGY). *Sharing ecosystems should use technology that makes these systems sustainable.*

For example, we do not believe that consuming energy at the scale of a major industrial economy is a good way of implementing consensus in an open network for sharing ecosystems, as seems to be the case for Proof of Work in the Bitcoin cryptocurrency [33]. And we also do not accept that the incentive mechanisms that such open systems typically require should lead to unfair treatment of some network participants over others, e.g. as seen in the dynamics of mining for cryptocurrencies based on Proof of Work – where it is no longer feasible to mine as a private citizen with standard computing equipment. This motivates our next principle:

PRINCIPLE 8 (FAIR AND DEMOCRATIC SHARING IN OPEN NETWORKS). *Technology should enable a democratic and fair participation in open sharing ecosystems.*

This principle is one based on values that are of a political or civic nature. It does not mean that network participants would all be equal, but it does imply that such networks should have transparent and fair governance structures in which network participants may gain influence based on the good contributions they have made to that network over time. For example, network participants may gain influence by contributing to the creation of consensus in the network state. However, the principle is also consistent with an open network that has primarily a commercial purpose and whose governance and membership models were co-designed with initial commercial stakeholders.

Unlocking the economic value of such networks will certainly require sufficient interoperability between IoT and ICT environments that allows for the flow of data and data-driven insights as well as for the evaluation and enforcement of access-control policies that regulate such flow [30, 34].

The lack of interoperability in IoT is a well recognized problem. Standardization efforts, through national and international bodies, are under way in order to address this issue (see e.g. the International Organization for Standardization and its Technical Committee 307 on blockchain and distributed ledger technologies). While we applaud such initiatives, we also recognize that it is difficult (politically and technologically) to create a globally recognized official standard that provides such interoperability all the way from IoT edges to ICT servers and storage environments.

Rather, we deem it to be more realistic to build lightweight software development kits (SDK) that can be installed on servers and micro-controllers alike, so that these SDKs become a de facto standard for connecting physical and digital infrastructures to application programming interfaces (APIs) that facilitate data sharing, access to physical function, and sharing of data-driven insights.

These SDKs would also allow for the specification of executable access-control policies that can refer to existing APIs (identity management systems, distributed databases, blockchains, standardized communication protocols, and so forth) through an expressive and *extensible* attribute-based language. The idea of extending domain-specific languages (DSL) in this manner has gained momentum in recent years, for example with the development and successful uptake of programming languages such as Scala [35] and Haskell [20]. These considerations suggest our next principle:

PRINCIPLE 9 (HIGH INTEROPERABILITY AT LOW COST). *Sharing ecosystems thrive if the software for its supporting infrastructure is lightweight, expressive, and extensible.*

Before we study challenges that a realization of our vision and its principles face, we first describe some initial technical work we did in this larger space and how it informed the vision set out in this paper.

3 OUR WORK DONE SO FAR

We now describe our past and ongoing efforts in shaping this vision with R&D partners and in developing the FROST technology. The latter serves as a foundation for realizing at least the first five

principles above and it can play a key role in enabling the remaining principles we set out above.

3.1 Porsche Pilot

Our in-house experts in Cybersecurity and Blockchain were selected as winners of a competition organized by the German luxury automobile manufacturer Porsche. The prize resulted in a pilot project to bring blockchain functionality into the modern car, with a view of exploring what possibilities this would bring – e.g. richer user experiences, novel services or better cybersecurity of vehicles; see e.g. [29] for a technical account of that project.

Our team did bring blockchain functionality into a Porsche Panamera and demonstrated not only how this approach can indeed strengthen the cybersecurity of accessing vehicles, but also how this could enable a suite of new service offerings for Porsche customers, with the resiliency that blockchain state can offer. For example, a car owner could delegate – through an app on a smartphone – to a designated delivery service access to the trunk in specific contexts such as time intervals. At youtu.be/KvyF78RTj18 an illustrative video shows those pilot outcomes.

These capabilities were brought about by integrating a Raspberry Pi into the vehicle's internal network and the delegation and access functionality was programmed in a somewhat ad-hoc manner, given the rapid pace of the pilot project itself. Of course, we realized at that time already that we needed a more programmable means of articulating and enforcing such controls in the medium term.

3.2 FROST Language

It was this recognition and the lessons learned during that pilot project that made us research policy-based access-control in the literature. One of the authors of this paper, together with Glenn Bruns (who was at Bell Labs at the time), developed the PBel policy-composition language based on Belnap bilattices [4, 5]. Their joint work, which began in 2006, also showed how satisfiability checkers can be used to answer a range of policy analyses that can support design, change management, and composition of policies.

This work was therefore a natural starting point for assessing suitable approaches. There are of course many other techniques and excellent contributions to policy-based access-control, composition of policies, and policy verification in the extent literature. But we soon recognized that PBel and its analysis framework would offer many benefits if adapted judiciously to the needs of the access-control infrastructure of our envisioned sharing ecosystem. Our team then worked out these revisions, which led to the development of the FROST language, its compiler, and its bespoke code analysis. "FROST" stands for **F**lexible, **R**esilient, **O**pen, **S**ervice-Enabling, and **T**rusted.

The FROST language expresses policies that also take on meaning over the 4-valued Belnap lattice: **grant**, **deny**, **conflict**, and **undef**. The two additional decision values **conflict** and **undef** were needed to adequately reflect the open nature in which policies may be composed in such ecosystems: the former captures that there is evidence for granting *and* for denying a request; the latter expresses that the policy has neither evidence for granting nor for denying a request. We appreciate that the access-control community has argued for the use of even more decision values, e.g. in

$$
\begin{aligned}
\textbf{grant if} \quad &(\textbf{object} \;\; == vehicle) \,\&\& \\
&(\textbf{subject} \;\; == vehicle.owner.daughter) \,\&\& \\
&(\textbf{action} \;\; == driveVehicle) \,\&\& \\
&(owner.daughter.driversLicense == valid) \,\&\& \\
&(0900 \leq localTime) \,\&\& \\
&(localTime \leq 2000)
\end{aligned}
$$

Figure 1: Example FROST policy that allows the daughter of a vehicle owner to drive that vehicle under specific conditions pertaining to time windows, valid driving license, and potential authentication implicit in the "subject" condition

the standard XACML 3.0 of the Organization for the Advancement of Structured Information Standards. But our engagement with business partners made us move additional values and their complexity into constraints whose truth values influence decisions by triggering rules within policies.

Like PBel, the FROST language is rule-based and its composition primitive is a **case**-statement in which guards capture decision scenarios of one or more sub-policies. FROST is also *attribute-based* in that its atomic conditions are formed from relations of attributes. The latter accommodates integration with other technology through APIs; e.g. we have developed such an API for the Security Assertion Markup Language. Attributes and their verification are a powerful tool. For example, an attribute **subject** can resolve to a more complex meaning such as multiple agents who issue a request based on a majority vote.[2]

Let us illustrate this with examples of policy and policy composition in FROST. The policy in Figure 1 consists of a sole rule that **grant**s the access when its condition is true and returns **undef** otherwise. That condition is a conjunction of constraints that provide mappings to the object, subject, and action of the access request and where additional conditions such as time windows constrain that access. We note that the semantics of these mappings is implementation-specific. For example, evaluating whether the subject is the daughter of the owner may be done through multifactor authentication (live AI for face recognition, biometrics of the driver seat, and so forth).

Figure 2 shows a **case**-statement, a priority composition of policies P and Q. FROST cases are of the form [*guard* : *policy*] where *guard* is either **true** or a conjunction of expressions of form

$$policy \;\; \textbf{eval} \;\; decision$$

saying that *policy* evaluates to a *decision*, an element of the set {**grant, deny, conflict, undef**}. The composition in Figure 2 therefore denies if P reports a **conflict**, returns the decision of Q if P reports **undef**, and returns the decision of P in all other cases. Deep embeddings of FROST into other languages such as Haskell would also support naming of conditions and policies, as well as function declarations and invocations whose return types are FROST conditions or policies.

---

[2]We appreciate that access-control *models* such as UCON [37] distinguish between subjects and subject attributes. FROST allows the use of more abstract DSLs that compile into FROST and that do make such distinctions.

```
case {
    [P eval conflict : deny]
    [P eval undef : Q]
    [true : P]
}
```

Figure 2: Example policy composition specified with a case-statement: this encodes a binary operator $P \gg Q$ where policy parameter P has priority over policy parameter Q

The FROST language also accommodates obligations, in an approach similar to one of [25]; but other obligation semantics are consistent with the FROST language as well. We have not yet implemented obligations, but are interested in exploring their enforcement aspects (e.g. for payments) through an open FROST network that makes use of DLT.

Finally, a FROST policy is compiled into two Boolean circuits that capture the 4-valued decision space and whose atomic expressions are actually atomic conditions occurring in the compiled FROST policies. It is these circuits that will be executed in the access-control architecture, making this execution environment simple and efficient enough for micro-controller environments as well. We refer to the FROST Yellow Paper [11] for further details.

3.3 Access-control Architecture

The access-control architecture for FROST is akin to that of XACML, making use of policy decision points, policy enforcement points, policy administration points, policy information points, and so forth [11].

We developed a reference client for this architecture, at present with minimal functionality, that can also run in a micro-controller environment. One thing we realized in its development is that the policy decision point can be very generic but the code for the policy enforcement point may vary as a function of the host environment (e.g. whether it is a micro-controller or a server or whether the decisions would be enforced in a moving vehicle).

3.4 Delegation Chains

Our vision foresees that actors can program their own sharing ecosystem. One aspect of that is the articulation and enforcement of a *delegation graph* that expresses the intent of actors in regard to the sharing of their assets.

So far, we designed cybersecurity protocols that an asset owner can initiate in order to build up a delegation *chain* of actors, where the chain starts with the asset owner and ends with the software agent that controls access to that asset. Intermediate actors are then permitted to submit their own policies. The overall policy executed on the asset is the composition of these policies, but where the choice of the composition operator is controlled by the asset owner. In FROST, we decouple syntax for conditions and policies from syntax and semantics of delegation; this simplifies reasoning and allows for the adoption of other approaches to delegation.

3.5 Open Sourcing

We have already published parts of the FROST technology as open source, please see the GitLab repositories

gitlab.com/xain-foundation/frost
gitlab.com/xain-foundation/proofofkernelwork

The second repository contains our initial client for a consensus algorithm *Proof of Kernel Work* (PoKW) [28, 29], an energy-efficient version of Proof of Work (PoW) to be used in open networks in which mining is also meant to become a more democratic process. PoKW uses a publicly verifiable random function on a blockchain to select a committee of network nodes for the next mining race. In that selection process, only selected nodes will know that they have been selected, and they can keep that information secret until and if they announce their solution of the mining race puzzle.[3]

Some of our FROST open-source code is written in Haskell, including code pertaining to language semantics, obligation semantics, compilation and related tools. As such, it serves to accompany the language specification in our Yellow Paper [11] but also stands on its own as a FROST DSL implementation. Other open-source code is our current reference client for FROST written in Rust and in the early stages of development; safety claims of Rust are now being verified on realistic language fragments by others (see e.g. [24]).

3.6 Infineon Project

We also entered a partnership with Infineon Technologies, after we demonstrated successfully in a minimal viable product how the capabilities of the Porsche pilot could be realized not on a Raspberry Pi but on an Electronic Control Unit that is part of the vehicle's product platform, an Aurix™ micro-controller:

www.infineon.com/cms/en/about-infineon/press/press
-releases/2018/INFATV201810-005.html

The execution of that project made us appreciate the extent and types of demands that processing and storage constraints of microprocessors place on realizing use cases of our vision of sharing ecosystems.

Through this partnership, we hope to be able to mature the FROST technology so that it will become production-ready for integration in the automotive sector, where lead times for such technology changes range from 3-5 years. A key aspect will be to integrate our technical approach with a de facto software standard, Autosar, for interfacing with the internal network of vehicles.

4 CHALLENGES FOR OUR VISION

In this section, we outline some of the key challenges that have to be addressed in order to realize our vision.

4.1 From Closed to Open Ecosystems

From our experience, it seems prudent to build up sharing systems and their supporting technologies bottom-up, rather than top-down. The latter approach would first build an entire economic platform and has then the problem of drawing parties into using that platform

as a marketplace for access to data, physical resources, and data-driven insights.

A bottom-up approach ensures that potential users of such a platform are actively involved in defining and shaping the technical aspects of that platform. This also seems to be a better fit for commercial parties, since they are interested in trying this first out in a (semi-)closed environment in order to fully control with whom to engage in sharing interactions.

An important part of our vision is an open sharing network, populated by many sharing biotopes that may interact with each other. One challenge faced here is to build up a sufficient number of biotopes to allow for network effects, and that at present the setup of a biotope requires social and technical on-boarding that should ideally be supported with as much automation as possible.

4.2 Building a Technical Community

Almost all of our code development for FROST will be open-sourced and its license models are meant to encourage open-source contributions so that others can freely use such code, also for further derived Intellectual Property that they may wish to exploit commercially.

One challenge in this is to build up a community of code developers that can make such contributions. The Rust programming language has its own ecosystem of developers but this is still a fairly small community, which limits the amount of qualified programmers as contributors and as potential recruitment targets. In automotive, use of C instead of Rust is currently preferred as this makes certification and compliance activities faster and cheaper.

We hope our Haskell-based DSL will help draw some interest from developers (and potentially academics) at the intersection of the access-control and functional programming communities.

4.3 Geographical Dynamics

Our vision and its supporting technology are about infrastructure and, as such, are meant to be global in reach. One challenge, certainly within Europe, is that moving from a pilot project to a scalable production case may take 3-5 years – and in some sectors even longer than that (e.g. automotive platforms last 10-15 years). This is a challenge for start-ups such as XAIN AG. We therefore are keen to also engage with partners who can integrate our technology in large production systems in 2019 or 2020 already.

4.4 Optimal Trade-offs

Access-control was invented as a means for making systems more secure. In sharing ecosystems, security and privacy demands are certainly very important. But trade-offs may have to be made. For example, it is not feasible to make an electronic control unit within a car into a full node of some blockchain, as the storage and compute resources of these devices won't allow for this.

This means that we need to develop hybrid approaches, for example one in which such an Electronic Control Unit only stores partial state of a blockchain that is pertinent to its own security state. Working this out in practice, with an appropriate threat and mitigation model, poses a challenge.

---

[3]We here acknowledge Algorand [10] as a pioneer of this use of Cryptographic Sortition for committee elections in their own, very different, consensus protocol.

5 ENGAGING ACADEMIC COMMUNITY

The academic community of access-control has indubitably a tremendous wealth of knowledge, tools, and general expertise to offer in order to support the shaping and realization of our vision. We here sketch some of the ways in which this community may help. But we want to stress that members of that community are best placed to come forward with ideas and proposals for how they can engage in this long-term project. We therefore are eager to hear from that community directly about what it thinks it could offer here.

Below, we feature some topics whose development would be of value to realizing our vision and that we think would be of interest to that academic community:

5.1 Policy Life Cycle Management

A trusted life cycle of an access-control policy and its potential updates and retirement are important. How this life cycle is specified and enforced can be seen as an aspect of policy administration. We seek models for this life cycle (e.g. automata-based models of permitted administration actions) that can be stored and executed on resource-constrained micro-controllers as well as on enterprise servers.

5.2 Policy Integrity

How best can we ascertain the integrity of an access-control policy, not just its provenance but also its freshness? DLT solutions suggest themselves, so that integrity tags of policies and their update history may be recorded on a distributed ledger. We have some initial concepts in that space, using a block*tree* rather than a blockchain, to reflect that many of these policies are independent of each other and so state can be sharded for higher scalability. This approach is based on a similar proposal for Bitcoin found at

lists.linuxfoundation.org/pipermail/bitcoin-dev/
2014-March/004797.html

5.3 Policy Verification

We realize that the academic community has done a lot of work on the formal verification of access-control policies and their administration. Here, we seek tools that make use of such established methods and techniques, e.g. Satisfiability Modulo Theories solvers (see e.g. [16]). We already have a good understanding of the first analyses we want to support (e.g. dead code analysis on policies). But we certainly stand to benefit from making use of existing tools of the community. And we think that FROST can offer interesting formal verification problems and impact opportunities to the academic community.

5.4 Policy Privacy

Privacy considerations are of paramount importance in many use cases. We would welcome collaborations with the academic community to ensure that FROST can be adapted to varying privacy needs. For example, our current delegation protocol only hides policies of delegatees from other delegatees and not from the device that stores and executes the composed policy. And this limited privacy guarantee is based on encryption, which may be too weak for some use cases that require higher privacy levels – e.g. as those associated with the "Right to be Forgotten" in the EU's General Data Protection Regulation (GDPR).

It is clear that the academic research community has a lot to offer in understanding which architectures for access-control and its administration can support very strict privacy demands such as those imposed by the GDPR. As the report by the French CNIL, found at

www.cnil.fr/sites/default/files/atoms/files/
blockchain.pdf

suggests, it is challenging to make blockchains GDPR-compliant but there seem to be ways forward in achieving this. We are particularly interested in solutions where blockchains (or blocktrees) manage the administrative and security state of access-control policies and their composition, rather than managing the history of financial transactions.

5.5 Access-Control on a Chip

In the medium to long term, we are interested in putting such complex access-control architectures on a chip that may host a secure database (which e.g. may store partial blocktree state). We are therefore eager to engage with academics who offer expertise in chip design for access-control.

5.6 App Store

Part of our vision is that others can use this sharing infrastructure in order to program applications on top of it, and where these applications make use of the sharing mechanisms of that infrastructure. For example, one could imagine an Artificial Intelligence app that is installed in vehicles that perform local reinforcement learning but where the models are updated across these vehicles for better accuracy and learning speed; and where FROST would facilitate the access restrictions but also the incentives for such sharing. We are thus also interested in what the academic community can offer in that regard. Topics of interest here seem to be algorithmic foundations for writing such apps, interface technology that can connect such apps with our infrastructure, user interfaces for specifying the access controls pertinent to such an app, to name a few.

5.7 Academic Outreach

We are keen to engage with the academic community through established means. For example, we would be prepared to organize a technical workshop dedicated to our vision – perhaps affiliated with an established conference. We would also welcome academics to visit our headquarters in Berlin, Germany, for a face-to-face chat about our vision and ongoing work and possible research collaborations.

6 RELATED WORK

In this section, we feature related work that pertains to the vision we have set out in this paper. We group this work into four themes rather than giving a many-to-many mapping of such work to the nine principles of our vision.

6.1 Sharing and Collaboration

First, we relate our vision to research on access-control for sharing and collaboration. In [42], XACML is used to refine existing enterprise access-control systems so that they allow for collaboration in distributed environments, also between different organizations. Role-based access-control is combined with attribute-based access-control in [27] to improve the security of multi-tenant cloud environments. Multi-tenancy architectures for Software as a Service can benefit from support for sub-tenancies. But this requires secure management of such complexity. An access-control model based on Administrative RBAC is proposed in [51] in order to address this. In [26], key distribution is combined with the authorization of users and a procedure for data sharing to allow for secure data sharing without the need of a trusted third party. Open challenges for access-control of jointly owned or shared documents are discussed in [47], a solution for the cloud environment is presented in [46], and a data-centric cloud solution is offered in [39]. Cross-organizational tracking of assets also is in need of access-control in order to avoid unintended confidentiality leaks [23]. Access-control for the provisioning of attributes and policies in collaborative network environments is developed in [44]. Work that uses domain knowledge and semantic relations between data and data usage, see e.g. [36], can improve policies for sharing and their enforcement. The sound and resilient recovery of policy state from missing attribute information, see e.g. [11, 15], is important in sharing ecosystems. Relationship-based access-control [3, 14, 31, 43], its policy negotiation [31], and its modelling languages [38] seem to fit well user experiences in sharing ecosystems.

6.2 Safety, Resiliency, and Trust

Second, we discuss work regarding the aspects of safety, resiliency, and trust. Access-control for sharing displays in a vehicle can ensure the safety of display sharing [19]. In [49], decentralized storage and blockchain technology are combined with attribute-based encryption to provide data privacy and fine-grained access-control while avoiding a single point of failure. Attestable trust anchors on clients can help with usage control of data across devices [48]. Combining reputation anchors with obligation handling [2] or with DLT mechanisms such as obligation chains [41] can make trust more resilient in open systems. Hardware isolation of the access-control functionality from the normal application layers, e.g. as done in [22], is of great interest for secure sharing technology. Access-control enforcement itself carries risks that may be mitigated through runtime tracking of risk estimates [40]. The need for access-control protection measures at different abstraction layers of systems is well recognized (see e.g. [45]). And the increasingly distributed and intelligent nature of connected systems brings both research challenges and opportunities for policy-based system management [6].

6.3 Compute Constraints

Third, we discuss work pertaining to the compute constraints in IoT systems. Access-control is an important aspect of vehicular cloud environments [21]. The challenge of compute demands for secure access-control within vehicle systems is recognized in [50], a fog-to-cloud architecture is proposed to address this limitation while also reducing latency of cloud-based solutions. The Policy Machine [18] is a framework that can host access-control enforcement with minimal code requirements, a generic mechanism, and expressiveness for refining constraints in high-assurance domains.

6.4 Policy Analysis and Synthesis

Fourth, we briefly relate to work on synthesis or analysis of access-control policies. The high degree of dynamics in IoT networks may benefit from use of event-driven policies [41] and a less static generation and enforcement of access-control policies (e.g. through machine learning) [7]. Machine learning can help with converting natural language policies into formal access-control policies, e.g. through the identification of attributes in [1]. Policy analysis tools can identify excessive privileges and redundancies [9], establish privacy properties [12], and improve revocation schemes for delegation chains [13].

7 CONCLUSIONS

This paper articulated our ambitious vision of technology for sharing ecosystems and its relevance to the access-control research community. Furthermore, we discussed exploratory and more concrete work that we already undertook towards realizing this vision. Then we talked about several challenges that we face in bringing this vision to fruition. We pointed out that we can benefit greatly from the broad and deep expertise of the academic community in access-control in overcoming these challenges in the medium to long term. We also reviewed research from the academic literature in order to illustrate pertinent connections between that extant work and our socio-technical vision.

As we said already, we would be very grateful for gaining any feedback on this vision, receiving any expressions of interest in research collaborations, and in obtaining any contributions to our open-source project for the FROST access-control and infrastructure technology.

8 ACKNOWLEDGMENTS

REFERENCES

[1] M. Alohaly, H. Takabi, and E. Blanco. A Deep Learning Approach for Extracting Attributes of ABAC Policies. In *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2018, Indianapolis, IN, USA, June 13-15, 2018*, pages 137–148, 2018.

[2] B. A. P. Botelho, D. G. Pelluzi, and E. T. Nakamura. A versatile access control implementation: secure box. In *18th ACM Symposium on Access Control Models and Technologies, SACMAT '13, Amsterdam, The Netherlands, June 12-14, 2013*, pages 249–252, 2013.

[3] G. Bruns, P. W. L. Fong, I. Siahaan, and M. Huth. Relationship-based access control: its expression and enforcement through hybrid logic. In *Second ACM Conference on Data and Application Security and Privacy, CODASPY 2012, San Antonio, TX, USA, February 7-9, 2012*, pages 117–124, 2012.

[4] G. Bruns and M. Huth. Access-Control Policies via Belnap Logic: Effective and Efficient Composition and Analysis. In *Proceedings of the 21st IEEE Computer Security Foundations Symposium, CSF 2008, Pittsburgh, Pennsylvania, USA, 23-25 June 2008*, pages 163–176, 2008.

[5] G. Bruns and M. Huth. Access control via Belnap logic: Intuitive, expressive, and analyzable policy composition. *ACM Trans. Inf. Syst. Secur.*, 14(1):9:1–9:27, 2011.

[6] S. B. Calo, D. C. Verma, and E. Bertino. Distributed Intelligence: Trends in the Management of Complex Systems. In *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2017, Indianapolis, IN, USA, June 21-23, 2017*, pages 1–7, 2017.

[7] S. B. Calo, D. C. Verma, S. Chakraborty, E. Bertino, E. Lupu, and G. H. Cirincione. Self-Generation of Access Control Policies. In *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2018, Indianapolis, IN, USA, June 13-15, 2018*, pages 39–47, 2018.

[8] M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. In *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 22-25, 1999*, pages 173–186, 1999.

[9] S. Chari, I. Molloy, Y. Park, and W. Teiken. Ensuring continuous compliance through reconciling policy with usage. In *18th ACM Symposium on Access Control Models and Technologies, SACMAT '13, Amsterdam, The Netherlands, June 12-14, 2013*, pages 49–60, 2013.

[10] J. Chen and S. Micali. ALGORAND. *CoRR*, abs/1607.01341, 2016.

[11] K. Cheung, M. R. A. Huth, L. M. Kirk, L. Lundbæk, R. Marques, and J. Petsche. The FROST Language: A trusted and user-centric access control language: Enabling delegation of fine-grained policies in shared ecosystems. Yellow Paper freely available online at https://xain.foundation, October 2018. Version 0.9.

[12] O. Chowdhury, A. Gampe, J. Niu, J. von Ronne, J. Bennatt, A. Datta, L. Jia, and W. H. Winsborough. Privacy promises that can be kept: a policy analysis method with application to the HIPAA privacy rule. In *18th ACM Symposium on Access Control Models and Technologies, SACMAT '13, Amsterdam, The Netherlands, June 12-14, 2013*, pages 3–14, 2013.

[13] M. Cramer, D. A. Ambrossio, and P. V. Hertum. A Logic of Trust for Reasoning about Delegation and Revocation. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, pages 173–184, 2015.

[14] M. Cramer, J. Pang, and Y. Zhang. A Logical Approach to Restricting Access in Online Social Networks. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, pages 75–86, 2015.

[15] J. Crampton, C. Morisset, and N. Zannone. On Missing Attributes in Access Control: Non-deterministic and Probabilistic Attribute Retrieval. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, pages 99–109, 2015.

[16] L. M. de Moura and N. Bjørner. Z3: An Efficient SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, pages 337–340, 2008.

[17] C. Dwork and M. Naor. Pricing via Processing or Combatting Junk Mail. In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, pages 139–147, 1992.

[18] D. F. Ferraiolo, V. Atluri, and S. I. Gavrila. The policy machine: A novel architecture and framework for access control policy specification and enforcement. *Journal of Systems Architecture - Embedded Systems Design*, 57(4):412–424, 2011.

[19] S. Gansel, S. Schnitzer, A. Gilbeau-Hammoud, V. Friesen, F. Dürr, K. Rothermel, and C. Maihöfer. An access control concept for novel automotive HMI systems. In *19th ACM Symposium on Access Control Models and Technologies, SACMAT '14, London, ON, Canada - June 25 - 27, 2014*, pages 17–28, 2014.

[20] J. Gibbons. Functional Programming for Domain-Specific Languages. In *Central European Functional Programming School - 5th Summer School, CEFP 2013, Cluj-Napoca, Romania, July 8-20, 2013, Revised Selected Papers*, pages 1–28, 2013.

[21] M. Gupta and R. S. Sandhu. Authorization Framework for Secure Cloud Assisted Connected Cars and Vehicular Internet of Things. In *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2018, Indianapolis, IN, USA, June 13-15, 2018*, pages 193–204, 2018.

[22] S. K. Haider, H. Omar, I. A. Lebedev, S. Devadas, and M. van Dijk. Leveraging Hardware Isolation for Process Level Access Control & Authentication. In *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2017, Indianapolis, IN, USA, June 21-23, 2017*, pages 133–141, 2017.

[23] W. Han, Y. Zhang, Z. Guo, and E. Bertino. Fine-Grained Business Data Confidentiality Control in Cross-Organizational Tracking. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, pages 135–145, 2015.

[24] R. Jung, J. Jourdan, R. Krebbers, and D. Dreyer. RustBelt: securing the foundations of the Rust programming language. *PACMPL*, 2 (POPL):66:1–66:34, 2018.

[25] N. Li, Q. Wang, W. H. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin. Access control policy combining: theory meets practice. In *14th ACM Symposium on Access Control Models and Technologies, SACMAT 2009, Stresa, Italy, June 3-5, 2009, Proceedings*, pages 135–144, 2009.

[26] Z. Li, M. Zhao, H. Jiang, and Q. Xu. Data sharing with fine-grained access control for multi-tenancy cloud storage system. In *Cloud Computing, Security, Privacy in New Computing Environments - 7th International Conference, CloudComp 2016, and First International Conference, SPNCE 2016, Guangzhou, China, November 25-26, and December 15-16, 2016, Proceedings*, pages 123–132, 2016.

[27] N. Lo, T. C. Yang, and M. Guo. An attribute-role based access control mechanism for multi-tenancy cloud environment. *Wireless Personal Communications*, 84(3):2119–2134, 2015.

[28] L.-N. Lundbæk, D. J. Beutel, M. Huth, S. Jackson, and L. Kirk. Proof of Kernel Work: A Resilient & Scalable Blockchain Consensus Algorithm for Dynamic Low-Energy Networks. *Yellow Paper*, v1.3, 2018.

[29] L.-N. Lundbæk, D. J. Beutel, M. Huth, S. Jackson, L. Kirk, and R. Steiner. Proof of Kernel Work: a democratic low-energy consensus for distributed access-control protocols. *Royal Society Open Science*, 5(8), 2018.

[30] J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon. Unlocking the potential of the Internet of Things, McKinsey Report, published online, June 2015.

[31] P. Mehregan and P. W. L. Fong. Policy Negotiation for Co-owned Resources in Relationship-Based Access Control. In *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies, SACMAT 2016, Shanghai, China, June 5-8, 2016*, pages 125–136, 2016.

[32] D. Mohr, H.-W. Kaas, P. Gao, D. Wee, and T. Möller. Automotive revolution – perspective towards 2030, published online, January 2016.

[33] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, May 2008. Published under pseudonym.

[34] R. Nicolescu, M. Huth, P. Radanliev, and D. D. Roure. Mapping the values of IoT. *Journal of Information Technology*, Mar 2018.

[35] M. Odersky and T. Rompf. Unifying functional and object-oriented programming with Scala. *Commun. ACM*, 57(4):76–86, 2014.

[36] F. Paci and N. Zannone. Preventing Information Inference in Access Control. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, pages 87–97, 2015.

[37] J. Park and R. S. Sandhu. The UCON$_{ABC}$ usage control model. *ACM Trans. Inf. Syst. Secur.*, 7(1):128–174, 2004.

[38] E. Pasarella and J. Lobo. A Datalog Framework for Modeling Relationship-based Access Control Policies. In *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2017, Indianapolis, IN, USA, June 21-23, 2017*, pages 91–102, 2017.

[39] T. F. J. Pasquier, J. Bacon, J. Singh, and D. M. Eyers. Data-Centric Access Control for Cloud Computing. In *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies, SACMAT 2016, Shanghai, China, June 5-8, 2016*, pages 81–88, 2016.

[40] G. Petracca, F. Capobianco, C. Skalka, and T. Jaeger. On Risk in Access Control Enforcement. In *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2017, Indianapolis, IN, USA, June 21-23, 2017*, pages 31–42, 2017.

[41] R. D. Pietro, X. Salleras, M. Signorini, and E. Waisbard. A blockchain-based Trust System for the Internet of Things. In *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2018, Indianapolis, IN, USA, June 13-15, 2018*, pages 77–83, 2018.

[42] E. Rissanen, D. Brossard, and A. Slabbert. Distributed access control management - A XACML-based approach. In *Service-Oriented Computing, 7th International Joint Conference, ICSOC-ServiceWave 2009, Stockholm, Sweden, November 24-27, 2009. Proceedings*, pages 639–640, 2009.

[43] M. Röscheisen and T. Winograd. A Network-Centric Design for Relationship-Based Security and Access Control. *Journal of Computer Security*, 5(3):249–254, 1997.

[44] C. E. Rubio-Medrano, Z. Zhao, A. Doupé, and G. Ahn. Federated Access Management for Collaborative Network Environments: Framework and Case Study. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, pages 125–134, 2015.

[45] A. Sadeghi. Mobile security and privacy: the quest for the mighty access control. In *18th ACM Symposium on Access Control Models and Technologies, SACMAT '13, Amsterdam, The Netherlands, June 12-14, 2013*, pages 1–2, 2013.

[46] C. Soriente, G. O. Karame, H. Ritzdorf, S. Marinovic, and S. Capkun. Commune: Shared Ownership in an Agnostic Cloud. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies, Vienna, Austria, June 1-3, 2015*, pages 39–50, 2015.

[47] A. C. Squicciarini, S. M. Rajtmajer, and N. Zannone. Multi-Party Access Control: Requirements, State of the Art and Open Challenges. In *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2018, Indianapolis, IN, USA, June 13-15, 2018*, page 49, 2018.

[48] P. G. Wagner, P. Birnstill, and J. Beyerer. Distributed Usage Control Enforcement through Trusted Platform Modules and SGX Enclaves. In *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, SACMAT 2018, Indianapolis, IN, USA, June 13-15, 2018*, pages 85–91, 2018.

[49] S. Wang, Y. Zhang, and Y. Zhang. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access*, 6:38437–38450, 2018.

[50] K. Xue, J. Hong, Y. Ma, D. S. L. Wei, P. Hong, and N. Yu. Fog-aided verifiable privacy preserving access control for latency-sensitive data sharing in vehicular cloud computing. *IEEE Network*, 32(3):7–13, 2018.

[51] Q. Zuo, M. Xie, G. Qi, and H. Zhu. Tenant-based access control model for multi-tenancy and sub-tenancy architecture in software-as-a-service. *Frontiers Comput. Sci.*, 11(3):465–484, 2017.