OConsent: Open Consent Protocol for Privacy and Consent Management with Blockchain

Subhadip Mitra
WILP, BITS Pilani, India
2017ht12635@wilp.bits-pilani.ac.in, contact@subhadipmitra.com

Abstract— **In the current connected world - Websites, Mobile Apps, IoT Devices collect a large volume of users' personally identifiable activity data. These collected data is used for varied purposes of analytics, marketing, personalization of services, etc. Data is assimilated through site cookies, tracking device IDs, embedded JavaScript, Pixels, etc. to name a few. Many of these tracking and usage of collected data happens behind the scenes and is not apparent to an average user. Consequently, many Countries and Regions have formulated legislations (e.g., GDPR, EU) - that allow users to be able to control their personal data, be informed and consent to its processing in a comprehensible and user-friendly manner.**

This paper proposes a protocol and a platform based on Blockchain Technology that enables the transparent processing of personal data throughout its lifecycle from capture, lineage to redaction. The solution intends to help service multiple stakeholders from individual end-users to Data Controllers and Privacy Officers. It intends to offer a holistic and unambiguous view of how and when the data points are captured, accessed and processed. The framework also envisages how different access control policies might be created and enforced through a public blockchain including real time alerts for privacy data breach.

Keywords— *Privacy, Blockchain, Distributed Ledger (DLT), Ethereum, GDPR, Privacy, Security, PII, Cryptography*

I. Introduction

Analysis of the Users' activity and behaviour on the websites and mobile apps provide unique insights to help businesses improve their products, service offerings and general user experience. Users' privacy and trust are key for any successful business - and thus user's consent must be sought before their data is used to maintain the said sustained trust and transparency. Given the volume of web traffic, geographies, prevalent sovereign privacy laws and multiple ways that the data points are used (e.g. Analytics, Recommendations, A/B Testing and personalization, Conversion tracking, Marketing Automation, Remarketing and User Feedback) - it is important to design a unified, open and extensible framework for Privacy and Consent Management. The framework must be able to capture consent, track lineage and enforce redaction (when consent is withdrawn).

Blockchains (and Distributed Ledger Technologies) by their very design provide trust and immutability of data . These two key features provide the building blocks of such Technology enabled Privacy Framework.

In this paper I intend to present a new protocol, and a platform architecture implementing the protocol - built on top of permissionless blockchain technology that can transparently address the Data Privacy and Consent Management concerns of digital businesses and legislators. The platform intends to provide a transparent and non-repudiable protocol for full lifecycle management of consent for the end users as well as business and organization. The platform would also provide an audit track for the consent usage as per the agreed norms between end users and organization. In short, the platform intends to empower the end users to make informed decisions and provide full control of their consent; and enable businesses to use such consent with confidence and in compliance of the prevailing legislations.

A. *Definition of Consent*

Following are the key consent definitions as per GDPR[1] and DPA. GDPR is considered the foremost and all-encompassing regulation for Data Privacy and Consent Management that is modelled by other legislations across different geographies. Throughout this paper, discussions are aligned to GDPR regulations.

1995 DPA definition - "... any freely given specific and informed indication of his wishes by which the data subject signifies his agreement to personal data relating to him being processed" [2]

The GDPR definition - "... any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her" [3]

B. *Maintaining the Integrity of the Specifications*

Stage 1: Collection - Consent is first collected from the Data Subject (DS).

Stage 2: Storage - Collected consent is then securely stored.

Stage 3: Process - The stored consent then is processed based on the context that it was obtained for by Data Controller (DC) and a Data Processor (DP).

Stage 4: Modification - Consent may be modified to accommodate a change in scope.

Stage 5: Revocation - Consent may be revoked by DS owing to expiry or agreement breach.

Stage 6: Archive - Consent data may be archived for regulatory and audit needs.

Stage 7: Destruction - Consent data may be completely destroyed as per prevailing legislative needs.

C. *Blockchain for Consent Management*

Blockchain by its inherent design elements like decentralization, distributed peer-to-peer (P2P) network and implementation of an immutable ledger – enforces trust. The

DOI: 10.13140/RG.2.2.26751.12967

following key characteristics of a Blockchain makes it suitable for Consent Management.

Distributed - All transactions (monetary and non-monetary) that is included in a block is shared and updated across all nodes of the blockchain ledger network.

Secure - Security is enforced through various cryptographic functions.

Transparent - As all nodes and miners can access all the transactions on the chain, thereby enabling complete transparency on the blockchain.

Consensus Based - All participants in the network must agree to validate a transaction using consensus protocols, thereby eliminating any monopoly. As more participants join a network the robustness continues to increase.

Flexible - Event or condition-satisfiability based executions of custom codes (Smart contracts on Ethereum Virtual Machine (EVM) or Chaincode on Hyperledger Fabric) allows for flexibility of employing various logics, including Consent lifecycle management. Smart Contracts are self-verifying, self-enforcing and tamperproof.

D. Related Work

As our digital footprint has multiplied manifold over the past decades, and with organizations widely adopting the use of such personal data - there has been a growing acknowledgement that better data management practices must be devised, so that the control of one's own personal data remains with the data subject. Furthermore, with the wider adoption of Machine Learning and Artificial Intelligence among business there has been a surge in the demand for data collection for behavioral analytics. As discussed earlier, multiple legislations across the world are now trying to define standards around managing user's personal data and the necessary consent for its use, e.g., EU's GDPR. Consequently, there has been significant research and design of solutions that allow consent management recently.

One of the first uses of embedding attribution data onto blockchain was by the Blockstack domain name registration service. It used a Distributed Hash Table on a virtual crossover chain that separated the storage and blockchain operations. It stored the hashed key value pairs relating to the ownership and domain name details on the blockchain.

In 2018, Wang, Zhang and Zhang[4] proposed an access control mechanism with Ethereum, for managing entitlements of the files in the distributed Inter Planetary File System[5]. It employed a fine-grained customized attribute-based encryption. The keys for the attributes were generated and maintained by the data owner and disseminated to requesters.

The framework ADvoCATE [6] proposed the use of Blockchain and smart contracts for managing consent and preferences for IoT devices. ADvoCATE extends the concept from the 2018 paper by the same authors [7]. ADvoCATE uses Smart Contracts for directly embedding consents onto Ethereum public blockchain. Admittedly, this is not a cost-efficient solution as the price of Ether continues to rise. The paper uses XACML (eXtensible Access Control Markup Language) [8] based markup language as a standard policy language. XACML has had a mixed adoption in the industry [9]. There have been multiple improved markup languages [10] to XACML, e.g., Policy Machine (PM) [11] based New Generation Access Control (NGAC). NGAC computes decision through a linear algorithm over non-conflicting policies, thereby making it operationally efficient over XACML that requires collecting attributes and running computations (matching conditions, rules and conflict resolutions) across a minimum of two different data stores - leading to extended complex computation steps. The proposed OConsent platform recognizes the clear advantages of NGAC over XACML and hence uses NGAC based markups to handle incoming consent and data access requests from Data Controllers. One of the key components of ADvoCATE is the Intelligence Component, that uses Fuzzy Cognitive Maps (FCMs)[12] to resolve conflicting policies for access requests. Fuzzy Cognitive Maps are popular for modeling complex systems but are known to be plagued by time lags between causes and observed effects. Consequently, Generalized Fuzzy Cognitive Maps (GFCM)[13] and Generalized Rules Fuzzy Cognitive Maps (GRFCM) have been recently proposed to overcome such challenges. ADvoCATE also proposes a recommendation module, based on Cognitive Filtering for recommending personalized rules.

Consentio [14] is another platform that looks to address the management of consent with blockchain. Consentio uses Hyperledger, which is a permissioned blockchain. Hyperledger Fabric [15] is known to be faster while processing transactions when compared to the Permissionless blockchains like Bitcoin and Ethereum. However, having a permissioned blockchain inhibits the wider adoption of the platform, and arguably is against the inherent idea of a decentralized blockchain – where the admission on the platform is tightly governed. The platform also maintains a World State Store – which is a key value store maintained by Hyperledger Fabric, with simplistic GET and PUT requests. This provides a high throughput over and above the conventional Hyperledger Fabric's gains. OConsent uses an Open Source Distributed In-Memory Key Value Store - Apache Ignite[16], that provides extremely fast Global State Store for the platform with simple PUT/GET requests as well as fully compliant ANSI SQL interface with strict transactions and complex analytical querying needs. Consentio does not propose any standardize markup languages for access control policies.

Truong, Sun, Lee and Guo proposed [17] to use a permissioned blockchain based on Hyperledger Fabric for consent management and provenance, similar to Consentio. Consequently, although the platform produces a higher throughput [18] as exhibited by the benchmarks in the paper – it may not be widely adopted unlike Bitcoin and Ethereum. It must be noted that, similar throughputs are possible using Sidechains, State-channels and Plasma – that OConsent uses. The platform uses access tokens and log ledgers for controlling access and tracking usage. It uses MongoDB as a backend for its profile management webservice. The platform uses the built-in Hyperledger Fabric ordering service with Apache Kafka. The platform does not account for any anonymity or pseudo anonymity concerns.

In the studies encountered, many designs included either using direct public blockchains for embedding consent hashes or using a permissioned Hyperledger Fabric based blockchain. Both of these approaches have limitations, in throughput and adoption, respectively. OConsent proposes a mixed approach, using a Local Ethereum based sidechain for granular embedding of consent hashes and versioning, while using a combination of Ethereum Main net and Bitcoin Main net for capturing the state of the local platform. This approach enables a high likelihood of adoption as the local chain is Permissionless and guaranteed high throughput as it uses a Sidechain. OConsent also explores the use of State Channels and Plasma based 2nd Layer Scaling. An In-memory Distributed Global State Store also forms a key component of the platform enabling high throughput and low latency.

None of the explored platform offers any anonymity or pseudo anonymity options. OConsent provide Surrogate ID [19] and Zk-SNARKs [20] based zero-knowledge proof for anonymity needs. Another key feature that is only attributed to OConsent platform is the embedding of the Trusted Timestamps Proofs. Provable and trusted timestamping is important as it enables non-repudiable assertions that a consent was generated at a particular point-in-time. This is vital as we move towards increasingly real time interactions and consequently, must cater time-exactness of a consent availability or revocation. OConsent also includes Time Leasing of Consent – which is a powerful option to make sure consents are not awarded perpetually and that expirations can be enforced.

II. Design and Architecture

A. *Key functional requirement*

i. Freely given: Consent must be provided by the Data Subject (DS) freely and completely optionally without any coercion.

ii. Informed, Granular and separate: Purpose for which a consent is sought must be clear, atomic and definitive. Separate consent must be sought for separate scope. A consent requirement and context must be concise and specific. E.g., a consent pursued for marketing must not automatically be reused for analytics.

iii. Unambiguous grant: It must be clearly demonstrated that an individual (Data Subject) has granted consent. There must not be any ambiguity on the affirmative action.

iv. Named: Consent agreement must clearly define the Data Controller and Data Processing organization and any third parties involved. The platform must establish and manage verifiable identities.

v. Avoid default opt-in: The Data Controller or consent seeker must avoid using prefilled checkboxes or forms for seeking consent. The Data Subject must explicitly demonstrate affirmative opt-in actions.

vi. Right to withdraw consent: End Users (Data Subjects) must be clearly notified at the time of obtaining consent that they may revoke consent any time, and that there will not be any residual consent based actions subsequent to the withdrawal.

vii. Regular Review: Consent validity and usage must be continually reviewed. A consent management platform must thereby account for scheduled checks. The platform must also allow 3rd party auditors and reviewers to validate such consent usage claims.

viii. Time based lease : Consent granted must not be indefinite, and should include some time bound default expiry, if not explicitly overridden. This also requires that a platform must ensure a trusted timestamp, so that time based validity may be enforced.

ix. Right to Forget: The End User may choose to exercise his/her Right to Forget, which would entail a complete destruction of stored personal data from the platform and/or the Data Controller and Data Processor.

B. *Key non-functional requirements*

Security

i. Confidentiality and Privacy : The platform must ensure that necessary controls are included so that confidentiality and privacy is maintained for all stakeholders. These may include segregating roles and actions within the platform as well as segregation of duties among the platform and blockchain node operators. The platform must operate with the notion of least-privileges.

ii. Anonymity: The platform should provide options to Data Subjects (end-users) to operate with necessary anonymity when desired.

iii. Non-Repudiation: Trust in the platform can be established only when it operates transparently and all actions are supported by verifiable proofs. These proofs include verification of digital signatures, timestamping, fingerprinting, etc.

Performance:

i. Latency: The platform must operate with low latency for the majority of the processes and associated actions. As real time processing needs become centre stage – its paramount that the platform should be able to support actions like consent querying and consent revocations actions withing a few seconds. This may entail employing distributed in-memory cache for Consent Queries responses and Circuit Breakers for immediate consent revocation.

ii. Throughput: The platform must be able to handle high throughputs, order of at least 500 tps (transactions per second) as is demonstrated by contemporary implementation.

iii. Scalability: The platform should be ideally horizontally and linearly scalable. A microservices based architecture must be embraced for granular scalability.

Reliability: The platform should be able to operate reliably with a reasonably expected performance.

Availability: The platform should be fault tolerant, and must continue to operate even if there are node failures and network partitioning.

Modifiability: As the platform will continue to evolve, it must support extensibility and modifiability. These would require that Smart Contracts must be properly versioned and

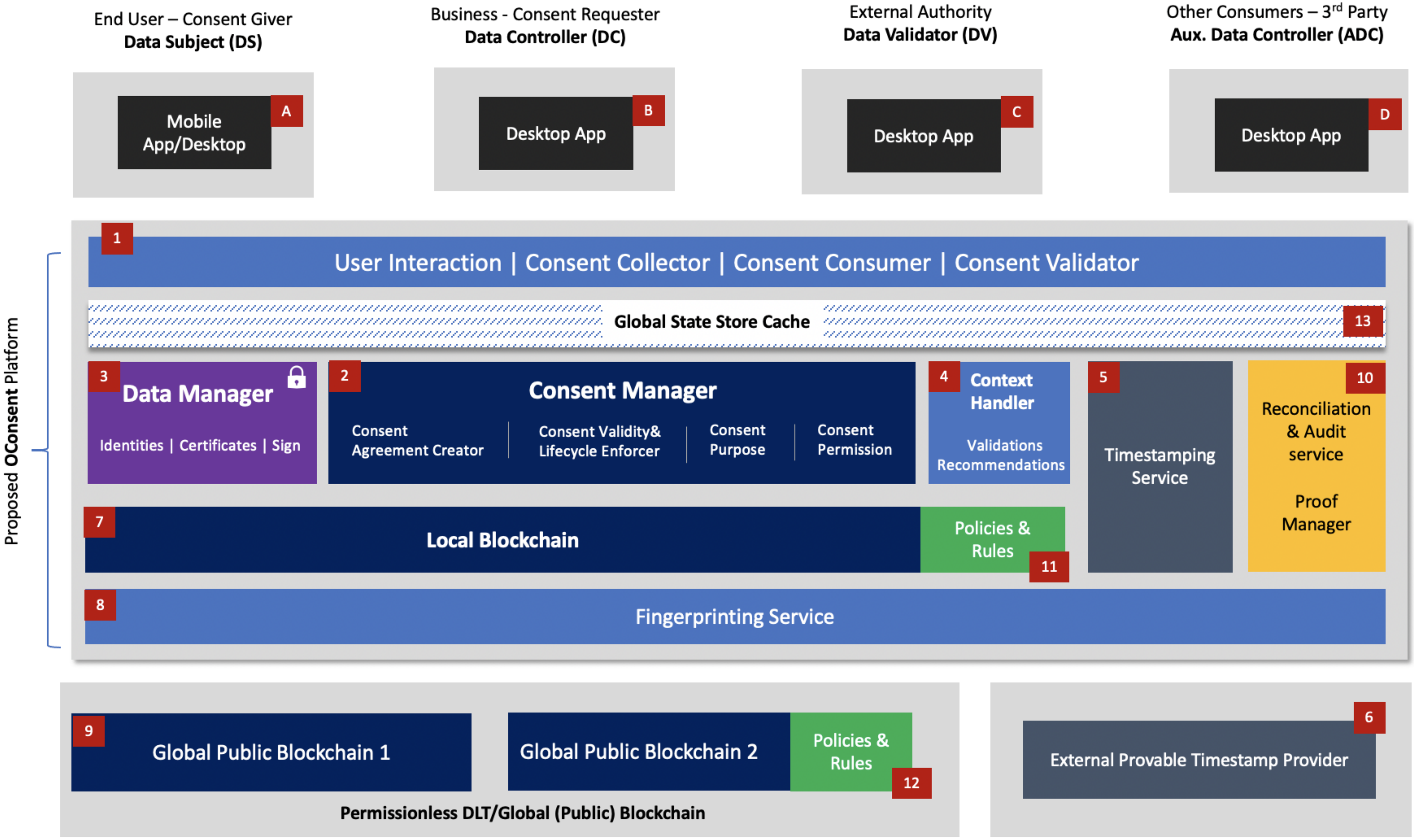

End User – Consent Giver
Data Subject (DS)
Mobile App/Desktop
A
Business - Consent Requester
Data Controller (DC)
Desktop App
B
External Authority
Data Validator (DV)
Desktop App
C
Other Consumers – 3rd Party
Aux. Data Controller (ADC)
Desktop App
D
Proposed OConsent Platform
1
User Interaction | Consent Collector | Consent Consumer | Consent Validator
Global State Store Cache
13
3
Data Manager
Identities | Certificates | Sign
2
Consent Manager
Consent Agreement Creator
Consent Validity& Lifecycle Enforcer
Consent Purpose
Consent Permission
4
Context Handler
Validations Recommendations
5
Timestamping Service
10
Reconciliation & Audit service
Proof Manager
7
Local Blockchain
Policies & Rules
11
8
Fingerprinting Service
9
Global Public Blockchain 1
Global Public Blockchain 2
Policies & Rules
12
Permissionless DLT/Global (Public) Blockchain
6
External Provable Timestamp Provider

Fig 1. Logical Architecture of OConsent Platform. The above figure represents the various logical components that make up the OConsent Platform.

designed so that newer and latest versioned Smart Contracts can be deployed without breaking changes. All interfaces and APIs must support extensibility for integrating with 3rd Party Service Providers.

Maintainability: The platform should be easy to maintain, i.e., installing upgrades and patches, without extensive downtimes.

Usability: Providing a simple, consistent and engaging UI/UX is key to attracting and retaining Users.

Cost: The design should cater for reducing operational cost. Infrastructure should be based on commodity non-specialized hardware. Where applicable, Open Source tools and frameworks should be adopted. Special attention must be given to reduce the transaction cost on the Global Public Blockchain, e.g., Ethereum and Bitcoin. This may entail deciding on the right batch size to include for fingerprinting on Bitcoin/Ethereum.

C. Key terms and definitions

Consent Agreement: Contract that lists all the details of a consent, e.g., parties involved – data subjects, data controllers, time period validity of the consent, context/purpose of the consent.

Consent Proof: Consent Proof is a collection of cryptographic proofs that guarantees the non-repudiation of the Consent Agreement. Proofs include provable timestamp, snapshot fingerprinting and/or full Consent Agreement's hash sum fingerprinting, consent versions lineage, etc. This is a JSON-LD [21] document.

File Hash: A fixed length string that is the output of passing a file's content through a hashing function, e.g., SHA 256. Every file with a different content produces a different hash value, whereas a file with same content will definitely produce the same hash value. The Hash value generated is thus essentially the fingerprint or identity of the file and its contents.

Signature: A file may be signed with a Private Key, to establish the ownership of the key and to prove that a file has not been modified. A signature is usually a fixed length string of characters. A user's Private Key is used to sign a file, whereas its Public Key is used to verify the ownership of the file.

Data Access Key (DAK): Data Access Key is used to access the Data Subject's (End User's) Data stored external to the OConsent Platform, after the Data Controller (or data requester) has proven that he/she has the necessary consent permissions to access such data.

D. Key terms and definitions

Data Subject (DS) - Data Subjects (or DS) are End Users who provide consent. DS are the primary actors on the platform and have full control on the consent lifecycle from creation to usage to deletion.DS interact with the platform through Mobile Apps available on iOS and Android Devices. The Apps serve as one-stop source for information on all active consents as well as well as their management.

Data Controller (DC) - Data Controllers (or DC) are the Business or Organizations that seek consent from the end users. DCs interact with the platform through a web portal. Every DC has tiered accounts – starting with the primary admin account, followed by other secondary accounts with various permissions. These secondary accounts may have varying access rights based on the different business units they belong too. An example of the tiered account would be a Bank – that has one primary account with super privileges, while multiple secondary accounts for Consumer Banking, Institutional Banking, and Digital Banking.

Data Validator (DV) - Data Validators (or DV) are independent actors who may validate if an organization or business is using a DS's consent in accordance with the DS's permission. Typically, DVs can be external auditors (both governmental as well as non-governmental). DVs requests for consent validations and proofs are served through the immutable fingerprints on the public blockchains.

Auxiliary Data Controller (ADC) - Auxiliary Data Controllers (or ADC) are third party entities that may inherit consents from Data Controllers (DCs). Propagation of consent via DCs must be in-accordance with the Data Subjects (DS) explicit permission and must not be assumed. ADCs are typically DC business partners. Before a consent is federated or propagated to ADCs it undergoes validations for rules conflicts.

Other Actors (OA)

- Platform Operators (PO)
- Local Blockchain Miners/Participants (LB): These are users who operate an instance of the Local OConsent Blockchain Node. These may other DS, DC, DV or ADCs.
- Global (Public) Blockchain Miners (GB): These are miners from the general public who may or may not be participating in the OConsent Platform.

Other Actors (OA) do not have direct access to Personally Identifiable Information (PII) and the platform operates strictly on the principal of least privilege. Do note that, PII stored data and its Hash are decoupled, and that only the hashed identities of the datasets are fingerprinted.

E. Key Components

1. **Interactions Layer** - This is the interface layer with which the various actors interact with the platform. This is also the interface that users use to capture and manage their consent and data. This layer provides the full suite of actions governing the consent lifecycle from definition and enforcement of data rights, data erasure and data/consent modification aligned to PM [22] markups.
2. **Consent Manager** - Consent Manager is the heart of the platform and undertakes multiple functions, Consent Agreement Creator, Consent Validity and Lifecycle Enforcer, Consent Purpose and Consent Permissions. The Consent Manager takes in the "Consent Request" from the Data Controller and the "Expression of Consent" from the Data Subject and enforces that the data is handled according the consent terms and privacy statutes. The Consent Manager also maintains multiple versions of the consent for audit and tracking purposes. Only the current (latest) version of the consent is enforced. It also coordinates with other modules to trigger the metadata captures associated with the consent lifecycle, e.g., who

created the consent, for whom was the consent created, the unique hash associated with the consent, timestamping requirements, data vaulting, etc. The consent manager is responsible for maintaining the "Consent Proof".

3. **Data Manager** - Data Manager is responsible for securely storing various data and only allow authorized access. The data types include, User's PII and non-PII Attribute Lists, Consent Metadata, Surrogate IDs, signature keys.

   Note that the platform does not physically store the Data Subject's data. Only the column metadata is retained. The Data Subject (DS) is responsible for storing and maintaining his/her data off-OConsent Platform either on AWS S3, GCP GCS, Azure Blob Store , Storj or some other decentralized store. DS or the platform that hosts DS's data must release the data only after Data Controller's demonstrated proof, e.g., the Data Access Key (DAK)

4. **Context Handler -** Every action performed by the actors on the platform have an associated context. The Context Handler is a reactive service responsible for interpreting the context and triggering a relevant action. For example: Data Subject (DS) may respond to a consent request from a Data Controller (DC). A context handler provides the following key functionalities:

   - Logically validate the context and associated rules for correctness or conflict.
   - Trigger Policies and Rules.
   - Recommend rules associated with the contexts (e.g., recommend rules of Consent Agreement based on the Data Controller's domain – ecommerce site)

5. **Timestamping Service** - This service invokes the External Timestamp Providers and embeds the timestamps into the generated Consent Proof.

6. **External Provable Timestamp Provider** - Multiple external Timestamp providers may be used to prove that an action ("Consent Agreement") happened after a certain point in time. This ensures the non-repudiation of the Consent Agreement.

7. **Local Blockchain** - A local blockchain is maintained to capture the Consent Agreement and Consent Proof details. It also embeds Smart Contracts/Chaincode that are executed in response to various events incoming from the Context Handler. The blockchain is generally a compatible local implementation of a public global blockchain, like Ethereum. Miners of the local blockchain include multiple DCs, DVs and ADCs. A DS may also choose to participate in the local blockchain by running a node.

8. **Fingerprinting Service** - This service takes the snapshot of the local blockchain and periodically publishes it to the global public blockchains like Ethereum and Bitcoin. The fingerprinting service may be scheduled by time or by volume of processed Consent Agreements.

9. **Global Public Blockchain** - These are public blockchains e.g., Bitcoin and Ethereum.

10. **Reconciliation and Proof Manager** - This service provides the necessary cryptographic and finite proofs for Data Validators. Proofs contain the validity of a Consent Agreement and its current usage as per the stipulations in the agreement.

11. **Policies and Rules (Local Blockchain)** - These are policies that are executed automatically based on the incoming legally relevant events and actions according to the terms of the Consent Agreement. These rules modify the state of Consent Proofs on the Local Blockchain of the platform.

12. **Policies and Rules (Global Blockchain)** - Similar to policies and rules for Local Blockchain.

13. **Global State Store Cache** - This is used to increase throughput and reduce the latency of the platform. It maintains a key-value store of Data Subject and Data Controller's agreement state in memory for fast retrievals. All front facing API requests are also served through the cache where applicable.

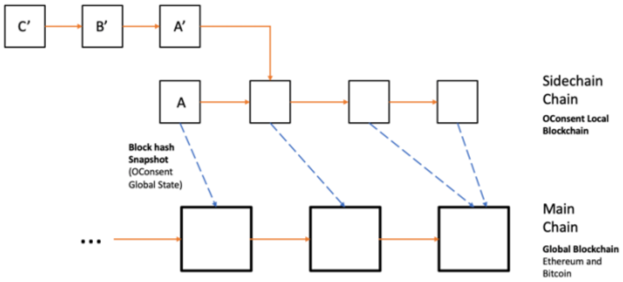
F. *Trusted and Provable Timestamping*

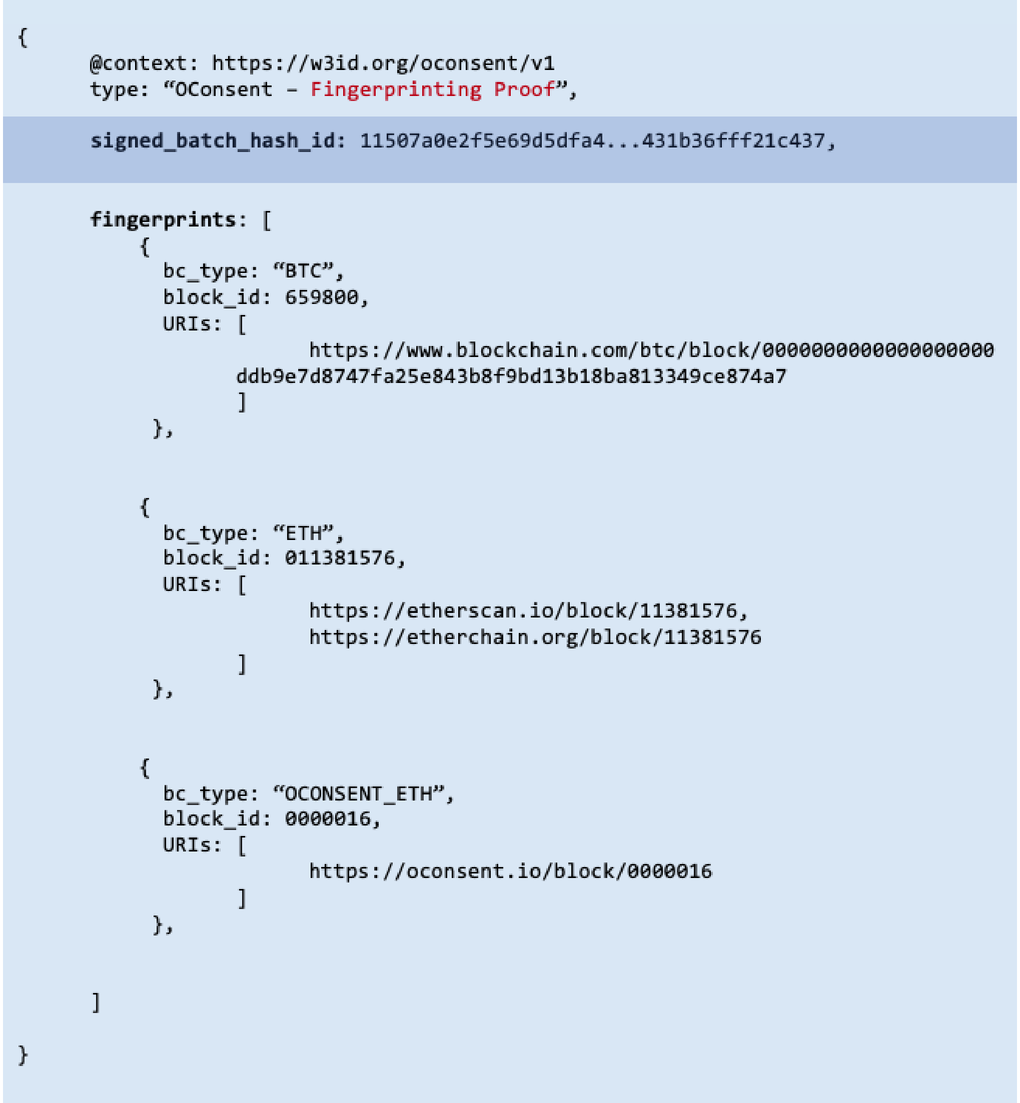
Trusted Timestamping helps to track when a Consent Agreement was created, modified or cancelled. Trusted Timestamping authorities provide the necessary cryptographic proof that makes repudiation of a consent event on OConsent Platform highly unlikely. None of the related work in blockchain based consent management employ a Trusted Timestamp Anchor. OConsent is the first Platform/Protocol that leverages provable timestamping for point-in-time validations. It is extremely useful for purposes of administration and audit. As the timestamp proofs can be publicly validated, the stampers integrity is unrepudiated.

G. *Fingerprinting on Global Public Blockchains*

The local blockchain forms a key component of the OConsent Platform. It's on the local blockchain that the Consent Agreements are embedded, including the Timestamp proofs. The local blockchain also maintains all versions of a Consent Smart Contract.

Fig 2. Sidechain interactions with a Main Chain and Security Implications.





```
{
      @context: https://w3id.org/oconsent/v1
      type: “OConsent – Fingerprinting Proof”,

      signed_batch_hash_id: 11507a0e2f5e69d5dfa4...431b36fff21c437,


      fingerprints: [
          {
            bc_type: “BTC”,
            block_id: 659800,
            URIs: [
                        https://www.blockchain.com/btc/block/0000000000000000000
                  ddb9e7d8747fa25e843b8f9bd13b18ba813349ce874a7
                  ]
           },


          {
            bc_type: “ETH”,
            block_id: 011381576,
            URIs: [
                        https://etherscan.io/block/11381576,
                        https://etherchain.org/block/11381576
                  ]
           },


          {
            bc_type: “OCONSENT_ETH”,
            block_id: 0000016,
            URIs: [
                        https://oconsent.io/block/0000016
                  ]
           },


      ]

}
```

Fig 3. Sample JSON-LD OConsent Fingerprinting Proof

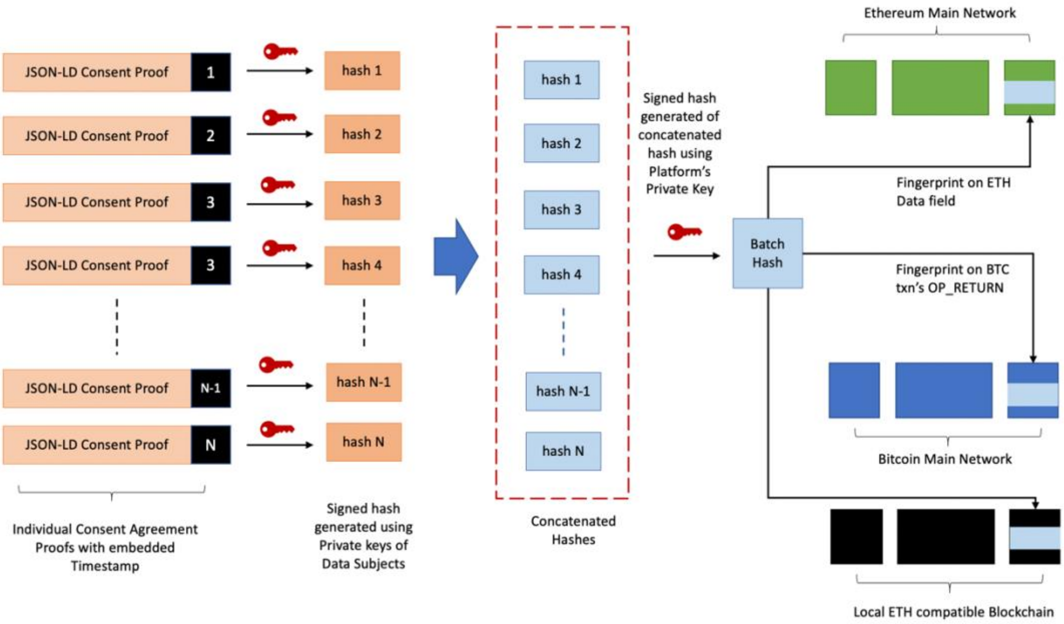

JSON-LD Consent Proof
1
JSON-LD Consent Proof
2
JSON-LD Consent Proof
3
JSON-LD Consent Proof
3
JSON-LD Consent Proof
N-1
JSON-LD Consent Proof
N
Individual Consent Agreement Proofs with embedded Timestamp
hash 1
hash 2
hash 3
hash 4
hash N-1
hash N
Signed hash generated using Private keys of Data Subjects
hash 1
hash 2
hash 3
hash 4
hash N-1
hash N
Concatenated Hashes
Signed hash generated of concatenated hash using Platform's Private Key
Batch Hash
Ethereum Main Network
Fingerprint on ETH Data field
Fingerprint on BTC txn's OP_RETURN
Bitcoin Main Network
Local ETH compatible Blockchain

Fig 4. How consent hashes are fingerprinted onto the public blockchains, Ethereum and Bitcoin

The OConsent platform operates a Sidechain. Simply put, Sidechains are a completely separate blockchain with its own set of actors, e.g., validators and operators. The Sidechain frequently transfers assets to main chain and back. One of the key purposes, (and also the same purpose of OConsent) is to capture the snapshot of the block headers to Main net in order to provide necessary guardrails against forking by bad actors on the sidechain.

In the figure below, if malicious validators of the sidechain conspire and collude to produce a different and longer chain with C'-->B'-->A' after the Block A has been mined and the OConsent Local Block A has been snapshotted onto the Main Chain – the longer chain would be discarded by the sidechain participant.

Furthermore, where a OConsent Local Sidechain participant wants to download a consent proof from the Main Chain, he/she must lock the 'batch consents' on the main chain and provide a proof of the lock to the side chain. To unlock the 'batch consents' on the main chain, the participant must initiate an exit on the local sidechain and publish a proof of the exit after the 'batch consents' have been added to the local sidechain block.

H. Anonymity on OConsent Platform

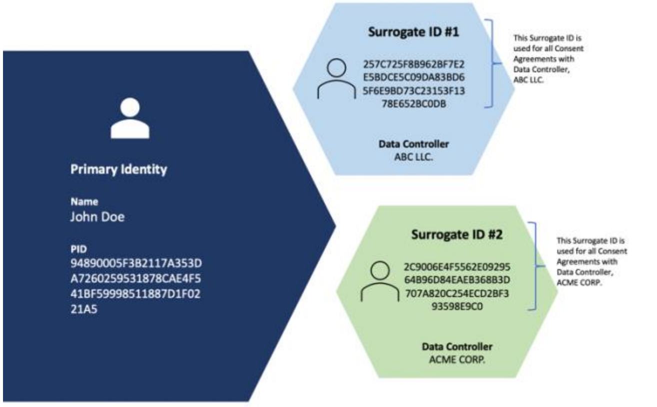


Fig 5. Surrogate IDs per Consent Contract

Surrogate Identities

The platform maintains a one-to-many mapping of Data Subjects primary and surrogate identities. This provides the DS's with the ability to anonymously share their data. A particular use case would be when a DS would not want to be tracked by advertisers using their real identities. It also provides the necessary guard against Data Controllers bypassing the OConsent Platform and the DS to collude among each other to share the data.

Zero Knowledge Proofs

There are multiple Zero Knowledge Proofs available, e.g., Zk-SNARKs, Zk-Starks, and Bulletproofs. Zk-SNARKs have been implemented successfully in production [23] and hence is the choice of Zero Knowledge Proof for the OConsent Platform. Zk-SNARKs [24] provides a proof construction whereby the Data Subject can prove possession of certain information without revealing that information, a without any explicit interaction between the prover (Data Subject) and verifier (Data Controller and Data Processor). Zk-SNARKs can help to verify proofs within a few milliseconds and "succinctly" provide proof within a few hundred bytes.

A possible use case of Zero Knowledge Proofs would be, if a Data Controller wants to know if a Data Subject is 18+ years old. Zk-SNARKs based non-interactive proofs may provide the answer without having to reveal the actual date of birth, thereby providing anonymity.

I. Interoperability Standard and Integration Requests Formats

OConsent platform uses the New Generation Access Control (NGAC) as a standard markup language for handling data access request:

a. During the initiation phase of agreement proposal by DC
b. For accessing additional data attributes access as an addendum to the consent agreement by the DC.
c. For audit and proofs access by the Data Validators (DVs)
d. Interoperations across other NGAC supported Data Storage Providers and Processors.
e. Other data access requests.

NGAC is a reference implementation of the Policy Machine (PM) and has clear advantages over the XACML (Extensible Access Control Markup Language). NGAC computes decision through a linear algorithm over non-conflicting policies, thereby making it operationally efficient over XACML that requires collecting attributes and running computations (matching conditions, rules and conflict resolutions) across a minimum of two different data stores - leading to extended complex computation steps. The proposed OConsent platform recognizes the clear advantages of NGAC over XACML and hence uses NGAC based markups to handle incoming consent and data access requests from Data Controllers.

NGAC also includes a standardized set of administrative operations with a unified interface. It also provides the same interface for decision making function for accessing data assets, which is remarkably amiss in XACML.

J. Resolving Classification Conflict

OConsent incorporates a module for resolving conflicting rules or policies of a Data Subject's consent. Traditionally, Fuzzy Cognitive Maps have been used for modelling complex systems but are known to be marred by time lags between causes and observed effects. Consequently, Generalized Fuzzy Cognitive Maps (GFCM) and Generalized Rules Fuzzy Cognitive Maps (GRFCM) have been recently proposed to overcome such challenges.

For OConsent platform, Double Induction [25] is proposed to be used. The idea behind Double Induction is that it induces unordered rules defined on the instances that are covered by the rules in conflict. By following this approach, new non-conflicting rules (as a result of separating the classes) are obtained by focussing on a smaller sub-space. This approach performs better over traditional Fuzzy Maps, Naïve Bayes and frequency-based classifications. Double Induction method does include a higher computation cost but the same is offset by the remarkable accuracy it attains – which is one of the key proponents of having this module on the OConsent Platform.

III. Summary

In this paper, I proposed the OConsent (Open Consent) framework that provides a comprehensive Consent Management System aligned to GDPR and other data privacy legislations using blockchain technology. The key goal of the platform is to provide a user-friendly solution that provides a one-stop solution for end users to reliably and confidently manage their consent. Optionally, the platform provides anonymity to the users using surrogate IDs or Zero Knowledge Proof – a first of its kind. Furthermore, a Double Induction based conflict resolution service is provided to better guide and advice Data Subjects (end users) while entering into Consent Agreements.

OConsent takes a practical approach to managing the Consent lifecycle with a Permissionless local sidechain. It provides multiple authoritative proofs for Consent receipt and validity for Auditors and Data Subjects. OConsent is also the only platform that implements a Trusted Timestamp proof to establish a non-repudiable point-in-time validity of a Signed Consent Agreement. OConsent also uses "multiple" Public Blockchains e.g., Bitcoin and Ethereum for fingerprinting the state of the local sidechain and thereby redundant proofs.

OConsent uses a standardized and most efficient access control policy mark-up language, NGAC. The platform has been designed to address scalability and performance needs from the initial.

Additionally, costs can be controlled by offloading the same to Data Controllers and Data Processors. As the platform is based on a sidechain approach, and only global states are fingerprinted on Bitcoin and Ethereum – the operating costs would be considerably lower that solutions that imprint all consent agreements on the main chain.

A. *Further Work*

As a future work, I intend setup a working solution of the OConsent platform. The implementation would be open-sourced, contributions are welcomed - github.com/OConsent

Other alternatives to a sidechain implementation, e.g., Plasma and Hyperledger Besu – would be explored, as well as designing of an adaptive intelligent scheduler for fingerprinting on Bitcoin/Ethereum at the most optimal time of lower Ether costs.

Finally, I would enhance the design and concept to allow Data Subject's to monetize their data for their consented data usage on the platform by Data Processors and Data Controllers.

References

[1] General Data Protection Regulation (GDPR) EU, Retrieved Aug 28, 2020, from https://gdpr.eu/

[2] UK Legislations, "Data Protection Act 1998". The National Archives on behalf of HM Govt. Retrieved Nov 3, 2020, from https://www.legislation.gov.uk/ukpga/1998/29/contents

[3] General Data Protection Regulation (GDPR), "Art. 4 GDPR Definitions", Retrieved Sep 10, 2020, from https://gdpr.eu/article-4-definitions/, Article 4 point (11)

[4] Wang S., Zhang Y., and Zhang Y., "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," IEEE Access, vol. 6, pp. 38 437–38 450, 2018.

[5] Benet J., "Ipfs-content addressed, versioned, p2p file system," arXiv preprint arXiv:1407.3561, 2014.

[6] Rantos K., Drosatos G., Demertzis K., Ilioudis C., Papanikolaou A., Kritsas A. (2019) ADvoCATE: A Consent Management Platform for Personal Data Processing in the IoT Using Blockchain Technology. In: Lanet JL., Toma C. (eds) Innovative Security Solutions for Information Technology and Communications. SECITC 2018. Lecture Notes in Computer Science, vol 11359. Springer, Cham. https://doi.org/10.1007/978-3-030-12942-2_23

[7] Rantos,K., Drosatos,G., Demertzis,K., Ilioudis,C., Papanikolaou,A.: Blockchain - based consents management for personal data processing in the IoT ecosystem, 15th International Conference on Security and Cryptography (SECRYPT2018), part of ICETE, page 572–577. SciTePress, Porto(2018). https://doi.org/10.5220/0006911005720577

[8] OASIS Standard, "eXtensible Access Control Markup Language (XACML) Version 3.0", Retrieved Sep. 23, 2020, from http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

[9] F Cser, A., Forrester, "XACML is dead". Retrieved Oct. 5, 2020, from https://go.forrester.com/blogs/13-05-07-xacml_is_dead/

[10] Harel, O., "Next-generation access control vs XACML: What's Really the Difference?", plainID.com. Retrieved, Oct. 5,˒ 2020, from https://blog.plainid.com/ngac-vs-xacml-whats-really-the-difference

[11] Computer Security Resource Center, "Policy Machine, Project Overview", NIST. Retrieved Oct 6, 2020, from https://csrc.nist.gov/Projects/Policy-Machine

[12] Bart Kosko, Fuzzy cognitive maps,International Journal of Man-Machine Studies, Volume 24, Issue 1, 1986, Pages 65-75, ISSN 0020-7373, https://doi.org/10.1016/S0020-7373(86)80040-2.

[13] Abhishek Nair, Diana Reckien, M.F.A.M van Maarseveen, Generalised fuzzy cognitive maps: Considering the time dynamics between a cause and an effect, Applied Soft Computing, Volume 92, 2020, 106309, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2020.106309

[14] Borisov, Vadim V. and Fedulov, Alexander S.,"Generalized Rule-Based Fuzzy Cognitive Maps: Structure and Dynamics Model", Neural Information Processing,2004, Springer Berlin Heidelberg, ISBN 978-3-540-30499-9, pp.918-922

[15] Agarwal R., Kumar D., Golab L. and Keshav S., "Consentio: Managing Consent to Data Access using Permissioned Blockchains", CoRR, Volu. abs/1910.07110, 2019, http://arxiv.org/abs/1910.07110

[16] Hyperledger Farbric, Retrieved Oct. 1, 2020, from https://www.hyperledger.org/use/fabric

[17] Apache Ignite, "In-Memory Computing Platform", Retrieved Nov. 10, 2020, from https://ignite.apache.org/

[18] Amritraj, "PUBLIC BLOCKCHAIN SCALABILITY: ADVANCEMENTS, CHALLENGES AND THE FUTURE" (2019). Master of Science in Software Engineering Theses. 2. https://digitalcommons.kennesaw.edu/msse_etd/2

[19] "What is a Surrogate Key? - Definition from Techopedia". Techopedia.com. Retrieved Oct. 21, 2020, from https://www.techopedia.com/definition/22403/surrogate-key

[20] Bitansky, Nir; Canetti, Ran; Chiesa, Alessandro; Tromer, Eran (January 2012). "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again". Proceedings of the 3rd Innovations in Theoretical Computer Science Conference on - ITCS '12. ACM. pp. 326–349. ISBN 9781450311151. S2CID 2576177

[21] JSON-LD, "JSON for Linking Data", Retrieved, Nov 2, 2020, from https://json-ld.org/

[22] Computer Security Resource Center, "Policy Machine", NIST, Retrieved on Nov 5, 2020, from https://csrc.nist.gov/Projects/Policy-Machine

[23] ZCash, "What are zk-SNARKs?", Retrieved Nov. 20, 2020, from https://z.cash/technology/zksnarks/

[24] Reitweissner C., "ZKSNARKS in a nutshell", Ethereum Blog. Retrieved on Nov. 20, 2020, from https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/

[25] Tony Lindgren and Henrik Boström. 2004. Resolving rule conflicts with double induction. Intell. Data Anal. 8, 5 (October 2004), 457–468