Modern Management based on Big Data III
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220078

Elastic Trust Management in Decentralized Computing Environments

Hiroyuki SATO [1] and Yepeng DING
The University of Tokyo, Japan.

Abstract. Today, the Internet trust is a critical framework that forms a fundamental premise for network security. Several identity federations are deployed to embody the classic theory of trust framework. However, in an emerging blockchain-based decentralized system, we need to model dynamically grown/shrunken trust. This paper models the trust elasticity by formalizing PDP(policy decision point)s of enrolling entities and introducing Trust PDP. In the proposed formalization, assertions are exchanged among participating nodes to express the knowledge of the PDP of nodes. The knowledge may grow by exchanging assertions. Furthermore, we propose exchanging assertions on the trust relationship to express the elasticity of trust. The judgment on trust is operated by Trust PDP that affects the elasticity of trust. The proposed theory is applied to the trust establishment of blockchain and an authorization management system based on blockchain. The trust in the blockchain environment is expressed in terms of Trust PDP.

Keywords. elastic trust, trust establishment, trust framework, PDP (policy decision point), Trust PDP, blockchain, decentralized system

1. Introduction

Today, the Internet trust is a critical framework that forms a fundamental premise for network security. A site or an organization accepts communications from those trusted using a contract or a framework in advance. Namely, to making the Internet trust effectively work, the policies of trust must be explicitly specified. Furthermore, entities enrolling into the community are bound by contract. Servers accept communications by obeying thus specified policies. This is modeled by the classic theory of trust framework. In addition, typical criteria of trust have been developed [1] that are the basis of the deployment of several identity federations such as eduGain (https://www.edugain.org), inCommon (https://incommon.org/federation), and GakuNin (https://www.gakunin.jp).

Generally, modern distributed or decentralized systems obey certain logic to determine whether a designated server accepts a service request. They may be explicit or implicit, but one certain thing is that the complexity is becoming uncontrollable. In a classic model, this logic is managed and operated by given PDPs (policy decision point) and PEPs (policy enforcement point). However, in an emerging blockchain-based decentralized system, we cannot assume such a centralized trust framework. The trust is local to an enrolling node that must extend its trust by consensus with peers.

Considering the trust of blockchain-based decentralized systems, we see that the trust is never static, but elastically grows or shrinks depending on dynamic factors such

[1]Corresponding Author: Hiroyuki Sato, The University of Tokyo; Email: schuko@satolab.itc.u-tokyo.ac.jp.

as communications history, consensus with other nodes and environmental change by join/leave of nodes. In order to model this kind of trust construction, we need to model the elasticity of trust.

Modern Internet architectures have evolved so that services are provided to a large number of participants, data is produced by a number of IoT nodes, and managed in a completely distributed way. Blockchain is a typical system used for such a scenario. In such a new kind of scenario, trust is managed in a decentralized way, and therefore negotiations for trust become complicated. To resolve this kind of possible trust chaos, the trust construction must still obey pre-specified rules that include those how nodes enroll into the network, how they leave it, and how trustworthy the produced data is. We have to note that thus constructed trust is still local to the enrolling nodes that may lack the capability of global trustworthy communications. Elevating the local trust to the global trust is mandatory. In this meaning, classic trust framework cannot be applied to this scenario because the evaluation of global trust levels must be based on dynamic trust-related behaviors of enrolling entities.

In the classic trust framework, the policies on service, security and trust are decided and enforced by participating RP (Relying Party)s. An RP receives a service request under its own service policies. A received request is evaluated by its PDP in the context of request for the policies of the RP. Along this line, we consider their extension of this scheme so that an RP can accept arbitrary data for making policy decision. This kind of additional data is generally called "assertion." In the classic trust framework, assertions are mainly used for security reasons. On the side of an RP, Sato[2] has proposed a model in which an RP accepts a service request accompanied with related assertions. The accepting RP runs its policies under which the received assertions are evaluated. If the evaluation are trustworthy, the collected assertions will be elevated to evidence. Assertions may include general documents that may represent operation policies or current behavior of a client, which an RP can also use to make an allow/deny decision.

In the same way, we extend the logic that express the range of trusted nodes whose assertions will be elevated to evidence. In a distributed or decentralized system, nodes may arbitrarily join and/or leave. This extend logic is a meta-logic to PDP.

We summarize our scenario in Fig. 1. A PDP of a node can enrich its policies by exchanging assertions with other nodes in a given trust circle **(a)**. Assertions can include policies that are evaluated and included by the PDP to enrich the inference of the node. Second, the trust circle can grow or shrink when a node joins or leaves it. This elasticity is expressed to be consistent with the inference within the trust circle **(b)**. This framework of elasticity is applied to a blockchain-based system **(c)** to show its effectiveness.

We start by modeling the knowledge exchange among the nodes that trust each other (modeling of classic trust frameworks) and extend the model to express the elasticity of trust. This theory is from [3], and will be applied to a decentralized system.

The rest of this paper is organized as follows. In Sect. 2, we give a formalization of knowledge exchange by using assertions in a general distributed system. In Sect. 3, we extend it to express the trust elasticity. In Sect. 4, we apply the proposed theory to blockchain scenarios. Sect. 5 surveys related work. Finally, Sect. 6 concludes this paper.

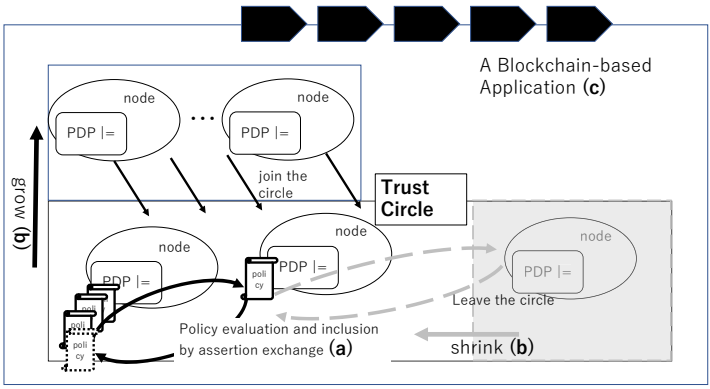

Figure 1. Goal and Application Scenario of the Elastic Trust

2. Formal Definitions of Trust and PDP

In [3], we have modeled a distributed system in which each knowledge and trust of an enrolling node grows or shrinks by exchanging "assertions." We first sketch it, emphasizing the role of PDPs.

A distributed environment is defined as a collection of entities e that communicates with other nodes in $e.env$. Its policy decision *Allow* or *Deny* is inferred and made on the basis of $e.pol$, the policy set of PDP, and $e.trust$, the set of trusted nodes and assertions. In the inference, a property can be an assertion of knowledge of e. $assert(e, \lceil P \rceil)$ for entity e and property P represents that the entity e claims the property P. Here, $\lceil P \rceil$ is a term for the internal representation of a given property P.

Definition 1 (Trust) The trust of e is a set of pairs of an entity and assertion: $\{(e_1, \lceil assert(e_1, \lceil P_1 \rceil) \rceil), \cdots, (e_n, \lceil assert(e_n, \lceil P_n \rceil) \rceil)\}$. Using A, a set of assertions, an entity e infers a property P. We denote it by $A \models^e P$. The inference is the first order logic that assumes $P \in e.pol$ and the following rule of assertions:

if $e.trust \ni (f, \lceil assert(f, \lceil P \rceil) \rceil), A \ni \lceil assert(f, \lceil P \rceil) \rceil$, then $A \models^e assert(f, \lceil P \rceil)$ and $A \models^e P$.

The intention of rules of *assert* is that when a node e accepts an assertion, its internal representation as data is translated into a policy. In this way, a policy is sent to another node. As its result, a set of assertions as policies trusted by an entity may grow during the communications with other entities in $e.env$ or shrink by invalidation of assertions (e.g. expiration). The rule on *assert* claims that the data $\lceil P \rceil$ is expanded to be used as property P in the inference of an accepting PDP. We note that the premise of inference may grow by communications with other entities.

It is often the case that an entity sends its policy set to another entity whose PDP is expected to evaluate it. Assuming that all policies are available in the a machine-readable form, Sato [4] gives a model in which policies of a peer such as security and privacy can be directly processed as data, and sent to a peer PDP. In our formalization, the machine-readable policy set $\{P_1, P_2, \cdots, P_n\}$ of an entity e is maintained as a collection of assertion

data $\{\lceil assert(e,\lceil P_1\rceil)\rceil,\cdots,\lceil assert(e,\lceil P_n\rceil)\rceil\}$ and can be exchanged within $e.env$. This enables the dissemination of policies among nodes to share the same policy set.

However, we must note that the decision "policy" of PDP whether an entity accepts an assertion from a specific entity is not yet discussed, which will be discussed in the next section.

3. Elastic Trust and Logic of Trust PDP

So far, an entity is assumed to communicate with a fixed set of peer entities in a given trust circle. In decentralized systems, no entity knows the set of entities existing in the world. An entity may come into or leave another entity's view at any time. In such an environment, dynamically controlling the trust, monitoring the environment is indispensable. In our system, we consider this policy for monitoring and controlling the environment of a given entity as a "meta" policy of PDP that is dynamically controlled by Trust PDP (TPDP). This dynamically growing/shrinking trust is called *Elastic Trust* in this paper, and controlled by TPDP.

Under the control of TPDP, by specifying a set of trustworthy assertions S, and executing $create(S,TP,P)$, an entity e can create a uniquely identifiable entity in $e.env$ where TP represents the TPDP policy, or TPDP properties defined below and P the PDP policy of the created entity. The name of a created entity belongs to at least one name space where the created name is unique.

A TPDP property is defined as a formula of the first order logic on $trust(t,S)$, $join(t,S)$, $created(e,t,S,TP,P)$ and $leave(t,S)$ for a entity t and a set of assertions S for $trust$, $join$ and $leave$ and for entities e,t, S a set of assertions, TP a set of TPDP properties, and P a set of properties for $Created$. As in a PDP property, $assert(t,\lceil P\rceil)$ for an entity t and a TPDP property P is also used.

As the entailment in TPDP properties in an entity e, we use the same logic for PDP. The properties *join* and *leave* in TPDP policies correspond to $Allow$ and *Deny* in PDP policies, respectively.

Actions taken by some entity (i.e. *join* and *leave*) affects the trust status of other entities. Trust status is represented by a set of assertions that the entity trusts. It shows elasticity, that is, it can grow or shrink dynamically. An entity claims its own trust status by continuously issuing its TPDP assertions. These assertions determine growth and shrink of the entity's trust $e.trust$. We define the elasticity as the trust status transition as:

Definition 2 (Elasticity)
- A trust status of an entity e is defined as a tuple (e,TP,T,TA) where TP represents a trust policy comprising of a set of TPDP properties, T the trust of e, and TA a sequence of TPDP assertions.
- We denote by $(e,TP,T,TA)\Rightarrow(e,TP,T',TA')$ that TA, changes to TA', and T to T' accordingly,

$(e,TP,T,TA)\Rightarrow(e,TP,T',TA')$ is defined reflecting the semantics of *created*, *join* and *leave*. Typically, If TA' is TA with $\{\lceil assert(t,\lceil join(s,B)\rceil)\rceil\}$ concatenated, and $join(s,B)$ can be inferred in TP, then $T'=T\cup(\{s\}\times B)$.

Receiving TPDP assertions. e modifies $e.trust$. In other words, the trust status (e,TP,T,TA) (T = $e.trust$) of a given environment is continuously updated. When $e.trust$ is modified, the inference of PDP is modified. In this meaning, the trust status of e shows

$$\begin{array}{ccc} \textbf{TPDP} : (e, TP, T, TA) & \Longrightarrow & (e, TP, T', TA') \\ \downarrow(\textbf{determines}) & & \downarrow(\textbf{determines}) \\ \textbf{PDP} : T = e.trust \models^e & \overset{(update)}{\longrightarrow} & T' = e.trust \models^e \end{array}$$

Figure 2. Hierarchical Transition of Trust Status of TPDP and PDP

transition. According to *join* and *leave* actions, $e.env$, the set of entities that can communicate with e, is updated to contain or remove s with its related assertions. Figure 2 summarizes the two-layer trust status transition of PDP and TPDP. In the figure, we note that $\longrightarrow$ and $\Longrightarrow$ are commutative.

As applications of this system, typical scenarios of blockchain and trust update by environment monitoring are analyzed in [3].

4. Elastic Trust for Blockchain Scenarios

Recently, blockchain technologies have been widely used to construct decentralized computing environments to enhance integrity and availability. A blockchain is an immutable distributed ledger based on a consensus mechanism enforced by all decentralized nodes to agree on transactions, which can be generally categorized into permissionless blockchain and permissioned blockchain. The trust among nodes varies in blockchain types.

Unlike permissionless blockchains, permissioned blockchains are governed by authorities to partially decentralize systems with trade-offs for the administration that provide trust among nodes. With the node trust assumption, permissioned blockchains can adopt consensus mechanisms like Raft[5] that have better performance than permissionless blockchains.

Our elastic trust model can be used to analyze trust in blockchain scenarios from the infrastructure level, like formalizing the trust establishment in permissioned blockchains, to the application level, like reasoning about decentralized authorization frameworks. In this section, we analyze the authorization system on the blockchain bearing [6] in mind.

4.1. Trust Establishment of Consortium Blockchains

A consortium blockchain is a type of permissioned blockchain that allows the existence of authorities to govern the network. Depending on the consortium blockchain platform, authorities can usually authenticate nodes for consensus to introduce trust.

In our scenario, a set of authorities (governors) $g_i \in G, i \in N$ governs a consortium blockchain. Each authority g_i provides and maintains a set of nodes $C_i, |C_i| > 1$ for consensus, which composes an authenticated node set $\hat{C}$. All nodes in $\hat{C}$ only process transactions from authenticated sources and reject any other transactions, i.e., trust is established only among all authenticated nodes. We also allow authorities to control the num-

ber of provided nodes dynamically. An authority g_i can authenticate new nodes to join the consensus and remove old nodes to mark them as unauthenticated.

For brevity, we formulate the trust establishment process for an authority $g_i \in G$ as follows. All other authorities in G follow the same process.

1. g_i is created with its PDP policy $\{g_j \mid j \in N, j < |G|\} \times \top$ and trust environment $g_i.env = G \setminus g_i$;
2. For each $c \in C_i$, g_i issues an assertion set $\{\lceil assert(g_i, \lceil created(g_i, c, \top, \{g_i\} \times \top, \emptyset)\rceil)\rceil, \lceil assert(g_i, \lceil join(c, \top)\rceil)\rceil\}$ to itself, which creates a node c with the TPDP policy $\{trust(g_i, \top)\}$ and the PDP policy $\{g_i\} \times \top$. Meanwhile, the PDP policy and trust environment of g_i are updated;
3. For all $c \in C_i$, c has the initial trust status $trust_{init} = \{g_i\} \times \top$ and dynamically updates its trust status by continually receiving TPDP assertions from g_i as follows.

   $$(c, TP_c, trust_{init}, \emptyset) \implies$$
   $$(c, TP_c, trust_{init} \cup (\{c_j \mid j \in N, j < |C_i|\}), TA_{g_i}) \implies \ldots$$

4. g_i publishes assertion sets $\{\lceil assert(g_i, \lceil created(g_i, c, \top, \{g_i\} \times \top, \emptyset)\rceil)\rceil \mid c \in C\}$ and $\{\lceil assert(g_i, \lceil join(c, \top)\rceil)\rceil \mid c \in C\}$ to all $g_j \in G, i \neq j$;
5. For all assertion sets received from $g \in G$, g_i updates its trust status and sends them to all $c \in C_i$;
6. For all $c \in C_i$, c updates its trust status with TPDP issued by $g_j \in G, i \neq j$ from g_i and update its trust environment.

After the trust is fully established, authorities trust each other and nodes trust authenticated node set $\hat{C}$ because of their trust to owners. A node can judge the source of a transaction by evaluating its TPDP policies and PDP policies. When a node c_m receives a transaction from a client, it will transmit the processed transaction information τ to other nodes as an assertion $\lceil assert(c_m, \lceil received = \tau\rceil)\rceil$. When the received transaction is processed by a node $c_n \in \hat{C}$, c_m will have an assertion $\lceil assert(c_n, \lceil received = \tau\rceil)\rceil$. c_m will issue $\lceil assert(c_m, \lceil assert(c_n, \lceil received = \tau\rceil)\rceil)\rceil$ and send to other nodes that have established trust with c_m. If the trust between c_m and c_n has been established, then c_m can simplify the assertion as $\lceil assert(c_m, \lceil received = \tau\rceil)\rceil$.

In the case when an authority g_i appends a new node c to the network, g_i will issue assertions $\lceil assert(g_i, \lceil created(g_i, c, \top, \{g_i\} \times \top, \emptyset)\rceil)\rceil$ and $\lceil assert(g_i, \lceil join(c, \top)\rceil)\rceil$. Similar to the trust establishment process, authorities in A and nodes in $\hat{C}$ will update their trust status and trust environments.

On the contrary, if an authority g_i removes a node c from the network, g_i will issue an assertion set $\{\lceil assert(g_i, \lceil leave(c, \top)\rceil)\rceil\}$ to all authorities in G including itself. When g_i delivers the TPDP assertion set, the trust status of g_j changes $(g_j, TP, T, TA) \implies (g_j, TP, T \setminus (\{c\} \times \top), TA + \{\lceil assert(g_i, \lceil leave(c, \top)\rceil)\rceil\})$.

In this manner, the consortium blockchain can establish and dynamically change trust among nodes.

4.2. Blockchain-Based Authorization Framework

Authorization frameworks have evolved into a new stage with the support of blockchain technologies. Studies [6,7] have proposed practical approaches to implementing autho-

rization mechanisms with smart contracts, a type of program deployed and executed on blockchains. Here, we show how our elastic trust model can be used to reason about decentralized authorization frameworks.

A decentralized authorization framework mechanizes trust between resource providers and resource consumers into a smart contract running on either a permissionless blockchain or a permissioned blockchain. A state of the smart contract can be abstracted by a function $\mathscr{S} : R \mapsto S$, where R is a set of resources registered by resource providers and S is a set of trust status. For a resource $r \in R$, we have $\mathscr{S}(r) = (r, TP, T, TA)$. Here r is identified by its wallet address $W(r)$, TP and T are defined in the contract storage, and TA can be constructed by assertion change logs stored in the event storage.

A resource provider p_i can register a set of resources R_i by issuing assertion sets $\{\lceil assert(p_i, \lceil created(p_i, r, \top, \{p_i\} \times \top, \emptyset)\rceil)\rceil, \lceil assert(p_i, \lceil join(r, \top)\rceil)\rceil\}$ where $r \in R_i$. For each registered resource r, r is created with its TPDP policy $\{trust(p_i, \top)\}$ and its PDP policy $\{p_i\} \times \top$. To authorize resource consumer u to access resource r, p_i can issue a TPDP assertion $\lceil assert(p_i, \lceil join(u, Allow(u, r))\rceil)\rceil$. On the contrary, p_i can issue an assertion $\lceil assert(p_i, \lceil leave(u, Allow(u, r))\rceil)\rceil$ to prevent u from accessing resource r. Because, all resources $r \in R_i$ trust TPDP assertions from p_i, resources can give correct authorization responses by using records on the blockchain to evaluate access requests.

5. Related Work

For constructing a trust framework, policies play a critical role in establishing the trust relationship among participants. P3P[8] was the framework that evaluated the published policy of Web servers. Partially inspired by the idea of P3P, the direct evaluation of security and privacy policies are proposed[2]. The policy evaluation framework is another facet of this study. In addition to ordinary audit and assessment, evidence based policy evaluation has been studies especially in health care systems[9].

In modern distributed computing environments where stakeholders enroll and leave dynamically and the network architecture is based on local connections, the security context and the related security policy assessment may dynamically change. One typical example includes supply chain. The hierarchical evaluation of policies [10] has been proposed to handle this locality. IoT environments are the second typical example. IoT device management is discussed with blockchain combined[11]. [12] is a hierarchical construction of blockchain. IoTeX (https://iotex.io) is another hierarchical blockchain-based management of IoT devices in which several functions have been proposed [13, 14]. Its trust is also discussed[15]. [16] uses the Ethereum platform instead of IoTeX. [17] is another example that adopts independent platform for controlling IoT. Together with our blockchain-based system, the trust analysis of these systems needs dynamism, which is one of the application fields of our theory.

6. Concluding Remarks

In this paper, we have extended the theory of classic trust frameworks in order to express the elastic trust typically seen in blockchain based decentralized system. The trust is formalized for an environment where enrolling nodes exchange properties by using asser-

tions. An assertion of a given property is represented by using the internal representation. The trust elasticity is represented by join and leave of nodes.

Furthermore, we have applied the proposed theory to consortium blockchain and to a blockchain-based authorization system. The system of trust elasticity is represented, and how it is used in the authentication has been shown. In such a scenario, consensus is the source of trust that is reflected in TPDP. The formal analysis of TPDP together with its relation to consensus is our next step.

In collecting a huge number of data, the heterogeneity of the related trust directly affect the construction of trust. Trust elasticity is one of key factors in the management of data and trust.

References

[1] Grassi P, Fenton J. Digital authentication guideline. Technical Report SP800-63-3: NIST; 2017.
[2] Sato H, Tanimoto S, Kobayashi T, Kanai A. Adaptive Policy Evaluation Framework for Flexible Service Provision. In: Proceedings of 2018 IEEE Symposium on Service-Oriented System Engineering; 2018 Mar; Oxford, UK: IEEE; p. 124–131.
[3] Sato H, Yamamoto N. Elastic Trust Model for Dynamically Evolving Trust Frameworks. Trans IEICE. 2019 Sep; E102-D(9):1617–1624.
[4] Sato H, Tanimoto S, Kanai A. A Policy Consumption Architecture that enables Dynamic and Fine Policy Management. In: Proceedings of 3rd ASE International Conference on CyberSecurity; 2014 Jun; Palo Alto, CA: ASE.
[5] Howard H, Schwarzkopf M, Madhavapeddy A, Crowcroft J. Raft Refloated: Do We Have Consensus? SIGOPS Operating Systtems Review. 2015 Jan; 49 (1):12-–21.
[6] Ding Y, Sato H. Bloccess: Towards Fine-Grained Access Control Using Blockchain in a Distributed Untrustworthy Environment. In: Proceedings of 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering; 2020 Mar; Oxford, UK: IEEE; p.17–22.
[7] Ding Y, Sato H. Dagbase: A Decentralized Database Platform Using DAG-Based Consensus. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference; 2020 July; Madrid, Spain: IEEE; p.798–807.
[8] Olurin M Adams C Logrippo L. Platform for privacy preferences (P3P): Current Status and Future Directions. In: Proceedings of 2012 Tenth Annual International Conference on Privacy, Security and Trust; 2012 Jul; Paris, France: IEEE; p. 217–220.
[9] Kuchenmüller T, Chapman E, Takahashi R, Lester L, Reinap M, Ellen M, Haby M. A Comprehensive Monitoring and Evaluation Framework for Evidence to Policy Networks. Evaluation and Program Planning. 2022 April; 91:102053. Volume 91, 2022,
[10] Tanimoto S, Watanabe Y, Sato H, Kanai A. Two-Tier Trust Structure Model for Dynamic Supply Chain Formulation. In: Proceedings of Advanced Informaion Networking and Applications 2022 (LNNS 451); 2022 April; Sydney, Australia:Springer; p. 324—333.
[11] Dai H.-N, Zheng Z, Zhang Y. Blockchain for Internet of Things: A Survey. IEEE Internet of Things Journal. 2019 Oct; 6(5):8076–8094.
[12] Lei K, Du M, Huang J,Jin T. Groupchain: Towards a Scalable Public Blockchain in Fog Computing of IoT Services Computing. IEEE Trans on Services Computing, 2020 March-April; 13(2):252–262.
[13] Fan X, Chai Q, Li Z, Pan T. Decentralized IoT Data Authorization with Pebble Tracker. In; Proceedings of 2020 IEEE 6th World Forum on Internet of Things (WF-IoT); 2020 June; New Orleans, LA: IEEE; p. 1–2.
[14] Partida A, Criado R, Romance M. Identity and Access Management Resilience against Intentional Risk for Blockchain-Based IOT Platforms. Electronics. 2021 Feb; 10(4):378.
[15] Boncea R, Petre I, Vevera V. Building Trust among Things in Omniscient Internet using Blockchain Technology. Romanian Cyber Security J. 2019 Spring; 1(1):25–33.
[16] Leal M, Pisani F, Endler M. A Blockchain-based Service for Inviolable Presence Registration of Mobile Entities. J. Brazil Comput Soc. 2021 Jan; 27(1).
[17] Hu J, Reed MJ, Al-Naday M, Thomos N. Hybrid Blockchain for IoT-Energy Analysis and Reward Plan. Sensors. 2021 Jan; 21(1):305.