

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

Blockchain for Access Control Systems

Vincent C. Hu

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.IR.8403-draft>

Blockchain for Access Control Systems

Vincent C. Hu
*Computer Security Division
Information Technology Laboratory*

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.IR.8403-draft>

December 2021



U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
*James K. Olthoff, Performing the Non-Exclusive Functions and Duties of the Under Secretary of Commerce
for Standards and Technology & Director, National Institute of Standards and Technology*

57
58 National Institute of Standards and Technology Interagency or Internal Report 8403
59 31 pages (December 2021)
60

61
62 This publication is available free of charge from:
63 <https://doi.org/10.6028/NIST.IR.8403-draft>

64 Certain commercial entities, equipment, or materials may be identified in this document in order to describe an
65 experimental procedure or concept adequately. Such identification is not intended to imply recommendation or
66 endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best
67 available for the purpose.

68 There may be references in this publication to other publications currently under development by NIST in accordance
69 with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies,
70 may be used by federal agencies even before the completion of such companion publications. Thus, until each
71 publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For
72 planning and transition purposes, federal agencies may wish to closely follow the development of these new
73 publications by NIST.

74 Organizations are encouraged to review all draft publications during public comment periods and provide feedback to
75 NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at
76 <https://csrc.nist.gov/publications>.

77

78 **Public comment period: *December 20, 2021 through February 7, 2022***

79 National Institute of Standards and Technology
80 Attn: Computer Security Division, Information Technology Laboratory
81 100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
82 Email: ir8403-comments@nist.gov
83

84 All comments are subject to release under the Freedom of Information Act (FOIA).

85

Reports on Computer Systems Technology

86 The Information Technology Laboratory (ITL) at the National Institute of Standards and
87 Technology (NIST) promotes the U.S. economy and public welfare by providing technical
88 leadership for the Nation’s measurement and standards infrastructure. ITL develops tests, test
89 methods, reference data, proof of concept implementations, and technical analyses to advance the
90 development and productive use of information technology. ITL’s responsibilities include the
91 development of management, administrative, technical, and physical standards and guidelines for
92 the cost-effective security and privacy of other than national security-related information in federal
93 information systems.

94

95

96

Abstract

97 The rapid development and wide application of distributed network systems have made network
98 security – especially access control and data privacy – ever more important. Blockchain
99 technology offers features such as decentralization, high confidence, and tamper-resistance, which
100 are advantages to solving auditability, resource consumption, scalability, central authority, and
101 trust issues – all of which are challenges for network access control by traditional mechanisms.
102 This document presents general information for blockchain access control systems from the views
103 of blockchain system properties, components, functions, and supports for access control policy
104 models. Considerations for implementing blockchain AC systems are also included.

105

106

107

Keywords

108

109 access control; blockchain; authorization; ABAC; policy.

110

111

112

Acknowledgments

113

114 The author, Vincent C. Hu of the National Institute of Standards and Technology (NIST), wishes
115 to thank Dylan Yaga, David Ferraiolo, Isabel Van Wyk, Jim Foti (NIST), and Antonios Gouglidis
116 (Lancaster University, UK) who reviewed drafts of this document. The authors also gratefully
117 acknowledge and appreciate the comments and contributions made by government agencies,
118 private organizations, and individuals in providing direction and assistance in the development of
119 this document.

120

121

Call for Patent Claims

122 This public review includes a call for information on essential patent claims (claims whose use
123 would be required for compliance with the guidance or requirements in this Information
124 Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be
125 directly stated in this ITL Publication or by reference to another publication. This call also includes
126 disclosure, where known, of the existence of pending U.S. or foreign patent applications relating
127 to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

128

129 ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in
130 written or electronic form, either:

131

132 a) assurance in the form of a general disclaimer to the effect that such party does not hold and
133 does not currently intend holding any essential patent claim(s); or

134

135 b) assurance that a license to such essential patent claim(s) will be made available to
136 applicants desiring to utilize the license for the purpose of complying with the guidance or
137 requirements in this ITL draft publication either:

138

139 i. under reasonable terms and conditions that are demonstrably free of any unfair
140 discrimination; or

141 ii. without compensation and under reasonable terms and conditions that are
142 demonstrably free of any unfair discrimination.

143

144 Such assurance shall indicate that the patent holder (or third party authorized to make assurances
145 on its behalf) will include in any documents transferring ownership of patents subject to the
146 assurance, provisions sufficient to ensure that the commitments in the assurance are binding on
147 the transferee, and that the transferee will similarly include appropriate provisions in the event of
148 future transfers with the goal of binding each successor-in-interest.

149

150 The assurance shall also indicate that it is intended to be binding on successors-in-interest
151 regardless of whether such provisions are included in the relevant transfer documents.

152

153 Such statements should be addressed to: ir8403-comments@nist.gov

154

155

156 **Executive Summary**

157 Access control is concerned with determining the allowed activities of legitimate users and
158 mediating every attempt by a user to access a resource in the system. The objectives of an access
159 control system are often described in terms of protecting system resources against inappropriate
160 or undesired user access. From a business perspective, this objective could just as well be described
161 in terms of the optimal sharing of information. As current information systems evolve to be more
162 lightweight, pervasive, and interactive network architectures such as Cloud and Internet of Things
163 (IoT), there is need for an access control mechanism to support the requirements of
164 decentralization, scalability, and trust for accessing objects, which is challenging for traditional
165 mechanisms.

166
167 Blockchains are tamper evident and tamper resistant blocks (digital ledgers) implemented in a
168 distributed fashion (i.e., without a central repository) and usually without a central authority (i.e.,
169 a bank, company, or government). It uses replicated, shared, and synchronized digital blocks
170 between the users of a private or public distributed computer network located in different sites or
171 organizations. Blockchain can be utilized for access control systems as a trustable alternative for
172 a single entity/organization or a member of a large-scale system to enforce access control policies.
173 The robust, distributed nature of blockchain technology can address issues in overcoming the
174 limitations of traditional access control systems in a more decentralized and efficient way. It is
175 supported by the following infrastructural properties that are not included in traditional access
176 control mechanisms unless specifically implemented:

- 177
- 178 • Tamper evident and tamper resistant design prevents **access control data** (i.e., attributes,
179 policy rules, environment conditions, and access request) and **access control logs** (i.e.,
180 request permissions, and previous access control data) from alternation and reduces the
181 probability of frauds.
 - 182 • Decentralized control of authorization processing and the storage of access control
183 data/logs has no single point of failure, thus providing more system tolerance and
184 availability.
 - 185 • The traceability of blocks allows access control data/logs and system states to be seen and
186 tracked.
 - 187 • The execution of arbitrary programs in smart contracts allows for controls on distributed
188 access control data and authorization processes.
 - 189 • Consensus mechanisms and protocols regulate the participating access control
190 entities/organizations jointly in determining policy rules through blocks or smart contracts.

191

192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210

Table of Contents

Executive Summary v

1 Introduction 1

2 Blockchain System Components and Advantages for Access Control Systems 3

3 Access Control Functions of Blockchain AC Systems 6

4 Access Control Model Support 15

5 Considerations 18

6 Conclusion 21

References 22

List of Figures

Figure 1 – XACML Architecture..... 6

Figure 2 – Examples of access control function points implemented in blockchain systems 11

Figure 3 – Examples of Figure 2d with attribute source options 13

List of Tables

Table 1 Comparison of IoT AC system capabilities for general access control requirements by blockchain and traditional mechanisms enforcing RBAC, ABAC, and CBAC policy models 16

211 **1 Introduction**

212 Access Control (AC) is concerned with determining the allowed activities of legitimate users and
213 mediating every attempt by a user to access a resource in the system. The objectives of an AC
214 system are often described in terms of protecting system resources against inappropriate or
215 undesired user access. From a business perspective, this objective could just as well be described
216 in terms of the optimal sharing of information [IR7316]. As current information systems evolve to
217 be more lightweight, pervasive, and interactive network architectures such as Cloud and Internet
218 of Things (IoT), there is need for an AC mechanism to support the requirements of decentralization,
219 scalability, and trust for accessing objects, which is challenging for traditional mechanisms.

220
221 Blockchains are tamper evident and tamper resistant blocks (digital ledgers) implemented in a
222 distributed fashion (i.e., without a central repository) and usually without a central authority (i.e.,
223 a bank, company, or government). It uses replicated, shared, and synchronized digital blocks
224 between the users of a private or public distributed computer network located in different sites or
225 organizations. A block links to the previous blocks by containing a cryptographic hash summary
226 of the previous block's contents, thus making the blockchain tamper resistant and tamper evident
227 properties (because to change a block, one must then change all subsequent blocks that follow it).
228 A linked list of blocks i.e., a blockchain typically has no central control authority utilizes a
229 decentralized consensus mechanism for reliable data transaction. A smart contracts contract is a
230 transaction protocol that executes the terms of a contract (such as payment term, lien,
231 confidentiality, and even enforcement) on a blockchain via code that is deployed to and executed
232 by blockchain nodes. The main purpose of smart contracts is to satisfy common contractual
233 conditions, minimize exceptions (both malicious and accidental), and the need for trusted
234 intermediaries. [IR8020] Every blockchain node that executes the smart contract should arrive at
235 the same result given the same input.

236
237
238 Blockchain can be utilized for AC systems as a trustable alternative for a single entity/organization
239 or a member of a large-scale system to enforce AC policies. The robust, distributed nature of
240 blockchain technology can address issues in overcoming the limitations of traditional AC systems
241 in a more decentralized and efficient way. It is supported by the following infrastructural properties
242 that are not included in traditional AC mechanisms unless specifically implemented:

- 243
- 244 • Tamper evident and tamper resistant design prevents **AC data** (i.e., attributes, policy rules,
245 environment conditions, and access request) and **AC logs** (i.e., request permissions, and
246 previous AC data) from alternation and reduces the probability of frauds.
 - 247 • Decentralized control of authorization processing and the storage of AC data/logs has no
248 single point of failure, thus providing more system tolerance and availability.
 - 249 • The traceability of blocks allows AC data/logs and system states to be seen and tracked.
 - 250 • The execution of arbitrary programs in smart contracts allows for controls on distributed
251 AC data and authorization processes.
 - 252 • Consensus mechanisms and protocols regulate the participating AC entities/organizations
253 jointly in determining policy rules through blocks or smart contracts.
- 254

255 Blockchain properties provide improvements of security, flexibility, scalability. The integrity, and
256 confidentiality for AC data/logs and processes over traditional AC systems by allowing
257 organizations to verify and audit AC data transactions and processes to track the states of their AC
258 systems hosted on distributed sites [SP162].

259
260 This document presents analyses of blockchain AC systems from the perspectives of properties,
261 components, architectures, and model supports, as well as discussions on considerations for
262 implementation. Sections included are:

- 263 • Section 1 is the introduction.
- 264 • Section 2 describes blockchain system components and their advantages over traditional
265 AC systems.
- 266 • Section 3 illustrates the architecture of AC basic functions for blockchain systems.
- 267 • Section 4 demonstrates blockchain AC system supports for AC policy models.
- 268 • Section 5 discusses considerations for the implementation of blockchain AC systems.
- 269 • Section 6 is the conclusion.

2 Blockchain System Components and Advantages for Access Control Systems

Blockchain systems provide an alternative (or complimentary) system for reliability, security, accountability, and scalability for AC systems. Blockchain characteristics such as, transparency, distributed computing/storage and a tamper evident/tamper resistance design help to prevent AC data from being modified by malicious users for unauthorized accesses, and access logs are recorded in blocks which to allow for the detection of malicious activities. Blockchain system components and their advantages for AC systems are:

- **Node** is an individual computer system within a blockchain network. It can act as an AC system's entity or organization; it is called **AC node** within the AC network. AC nodes including lightweight nodes (i.e., a node that does not store or maintain a copy of the blockchain), full nodes (i.e., a node that stores the entire blockchain and ensures that transactions are valid), and publishing nodes (i.e., a full node that also publishes new blocks). Lightweight nodes must pass their transactions to full nodes. Depending on the design of the AC system, AC nodes can act as a host server for AC data (e.g., subjects/object attributes, environment conditions, and policy rules) or as administrators for AC policy management and enforcement.
- **Block** contains trustable and tamper resistant AC data as well as a history of access logs without third parties or centralized management. Distributed blocks solve the single point of failure problem and provide information for distributed architectures, which often involve a much larger set of AC entities or organization. Distributed ownership of blocks is necessary because of possible trust, security, and reliability concerns that are associated with the centralized management of AC enforcement or AC data ownership [IR8202].
- **Blockchain** servers are not only a repository of AC data and logs of blocks but can also store objects. Even though blockchain contents are tamper evident and tamper resistant, [Kuhn] proposed a data structure with similar features to a blockchain, the data block matrix data structure, that allows for the deletion of arbitrary records and preserves hash-based integrity assurance that other blocks are unchanged. Such a feature may be incorporated into AC systems that require integrity and privacy protection such that organizations or users are able to delete all information related to a particular access request.
- A **consensus mechanism** ensures that only valid transactions are recorded on the blockchain. Different kinds of consensus mechanisms can be used for AC systems, including proof of work (PoW), proof of stake (PoS), and single committee-based [LQLL]. For mandatory AC (MAC) policies, the integrity and consistency of AC administrations are maintained by consensus mechanisms configured for permissioned blockchains. Consensus mechanisms configured for permissionless blockchains are crucial for discretionary AC (DAC) policies due to the dynamic management requirement for scalability and decentralization of the system.

- A **smart contract** is an event-driven computer program distributed to and executed by AC nodes to facilitate and enforce **AC processes** (i.e., authorization processes and AC data transitions) between them without going through a trusted third party. A smart contract can perform calculations, store data in storage spaces, expose environment conditions to reflect the current system state via callable functions, and – if appropriate – automatically send data or function calls to other smart contracts [IR8202]. Adding a smart contract to a block means executing code and updating the **AC state** (i.e., previous access permissions, environment conditions, and system status) accordingly [DMMR]. The smart contract code is also tamper-evident and tamper-resistant. It is copied to each AC node to reduce human error and avoid disputation, thus providing a secure way to specify AC policies and transform the authorization process into a distributed execution [KLG]. Such a capability works especially well for a system that requires each distributed AC entity to perform local authorization so that the authorization chain can be verified in a decentralized manner.

Blockchain systems' decentralized storage of AC data and the delegation of authorization processes not only optimize performance and cost of an AC system but also help avoid single points of failure and many-to-one traffic problems for highly dynamic and scalable systems (e.g., cloud, grid, IoT). This is especially true for AC systems that enforce attribute-based policy models, such as RBAC [FK] and ABAC [SP162], where the AC data management and policy enforcement are traditionally administrated by a central server. Blockchain also allows for AC log information collection and replicates data among AC nodes in a transparent and trustworthy way with a verifiable and secure records. The following blockchain capabilities are not generally supported by traditional, centrally controlled AC mechanisms:

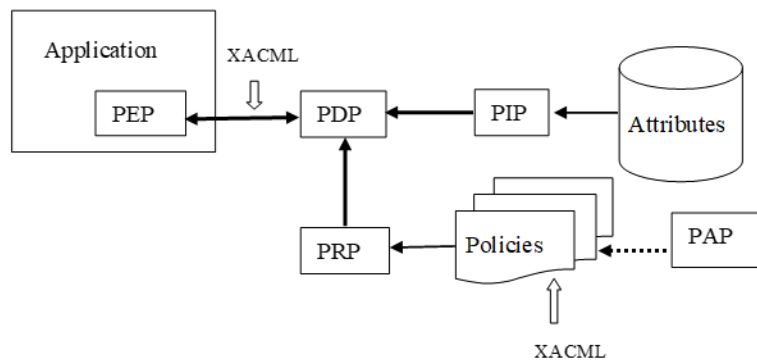
- Removes control from a centralized system and provides flexibility in AC data management and AC processes, such as workflow control or localization control, thus, avoids possible leakages or faults of access privileges by excessive powers of centralized server [LOLL]
- Increases performance for managing a large number of subjects and objects, such as IoT AC systems, where each IoT device is an AC node of an AC entity or organization
- Allows for the enforcement of flexible, fine-grained, and responsive policy by transferring or propagating access privileges from one AC node to others through smart contract functions
- Supports communication between subjects, AC administrators, and protocols for the administration of heterogeneous AC policies and security analysis
- Avoids tampering and single points of failure (e.g., caused by network attacks like DDoS) to increase integrity, availability, and traceability [GBHC] through recording, distributing, and storing AC data and log information in the blockchain. However, as all subjects can see all entries in the blockchain, privacy can be an issue for this capability.

- 357
- 358
- 359
- Dispenses heavy and complex authorization or management tasks between AC nodes to enhance performance and scalability, as well as decrease the cost and responsibility of administration traditionally assigned to central or third-party services.

3 Access Control Functions of Blockchain AC Systems

This section examines an integration of access control functional components and a blockchain framework in support of ABAC. Extensible Access Control Markup Language (XACML) [XACML] and Next Generation Access Control (NGAC) [INCITS] are two ABAC standards that could serve as a basis for this discussion. See [SP162] for a detailed comparison. Both standards encompass four layers of functional decomposition: Enforcement, Decision, Administration, and Access Control Data. Unfortunately, XACML and NGAC achieve this decomposition by involving components with often similar names but apply different access control data types, provide different interfaces, and result in different functional outcomes. To avoid confusion, the remainder of this section applies XACML's reference architecture, as an example ABAC integration use case.

The Organization for the Advancement of Structured Information Standards (OASIS) standard XACML proposes basic processing entities for AC systems. Each entity handles a different stage of processing a user's access request, as shown in Figure 1. These functional components may be physically and logically separated and distributed rather than centralized, such as several functional "points" that are the service node for retrieval and management of the policy, along with some logical components for handling the context or workflow of AC data retrieval and assessment.



379

381

Figure 1 – XACML Architecture

In a blockchain AC system, these function points can be performed by an individual or combination of blockchain system components. The following describes the five basic XACML function points and their implementations by blockchain AC components.

385

1. Policy Administration Point (PAP): Provides a user interface for creating, testing, and debugging policies, as well as storing these policies in the appropriate repository. PAP can be created and maintained by AC nodes or smart contracts that are coded to access AC policies, depending on where the source of the AC data is maintained.

390

2. Policy Information Point (PIP): Serves as the source of subject/object attributes or environment condition data required for policy evaluation to provide the information needed by the PDP to make the authorization decisions. PIP can be performed in AC nodes,

393

394 coded in smart contracts that are coded to access AC data, or hosted in an off-chain
395 processor, depending on where the source of the AC data is maintained.

396

397 3. Policy Retrieval Point (PRP): Where the policies are stored and fetched by the PDP. As
398 PIP depends on where the source of AC data is maintained, the PRP can be implemented
399 in AC nodes, coded in smart contracts, or hosted in an off-chain system, depending on
400 where the source of the policy rules is maintained [IR7874].

401

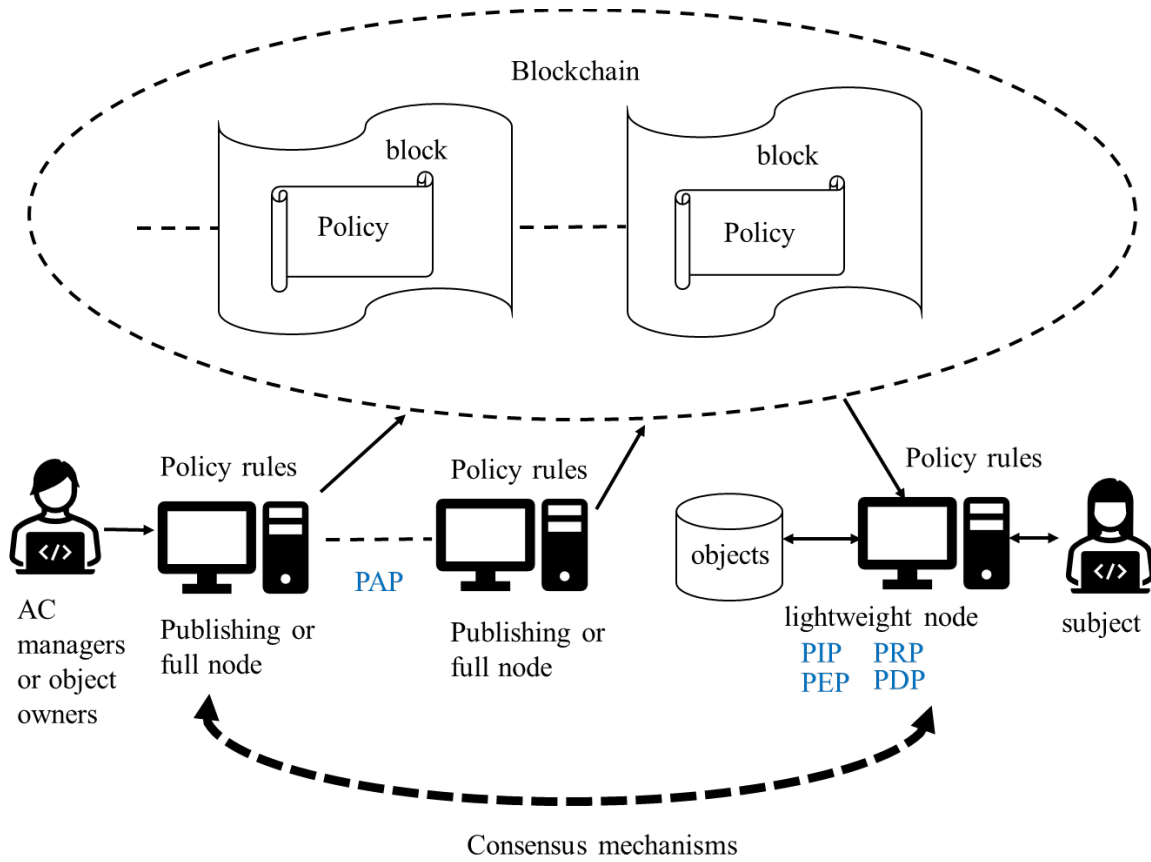
402 4. Policy Decision Point (PDP): Computes access decisions by evaluating the applicable
403 policies based on information provided by PIP and PRP. One of the main functions of the
404 PDP is to mediate or deconflict policy rules. PDP can be coded in a smart contract, into
405 distributed executions, or performed by AC nodes.

406

407 5. Policy Enforcement Point (PEP): Making decision requests and enforcing authorization
408 decisions made by PDP. PEP can be performed by AC nodes that contain objects, smart
409 contracts that are coded to access objects, or by an off-chain processor.

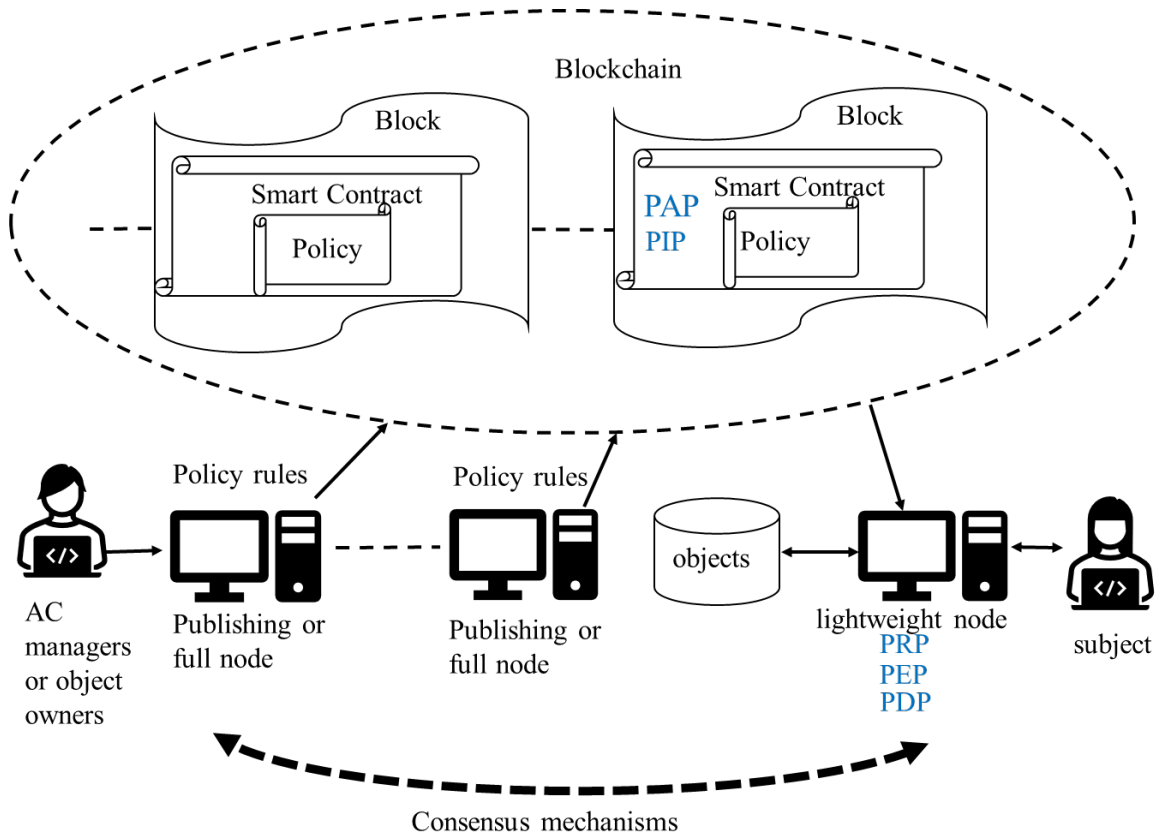
410

411 The basic AC function points can be processed through the uploading and updating of AC data to
412 execute AC processes, smart contract functions, or even off-chain processes, depending on
413 security/performance requirements and resource availability of the AC system. Figure 2 illustrates
414 examples of different assignments of function points in blockchain AC systems. Each function
415 point in a picture is labeled (in blue) alongside the performing blockchain component.



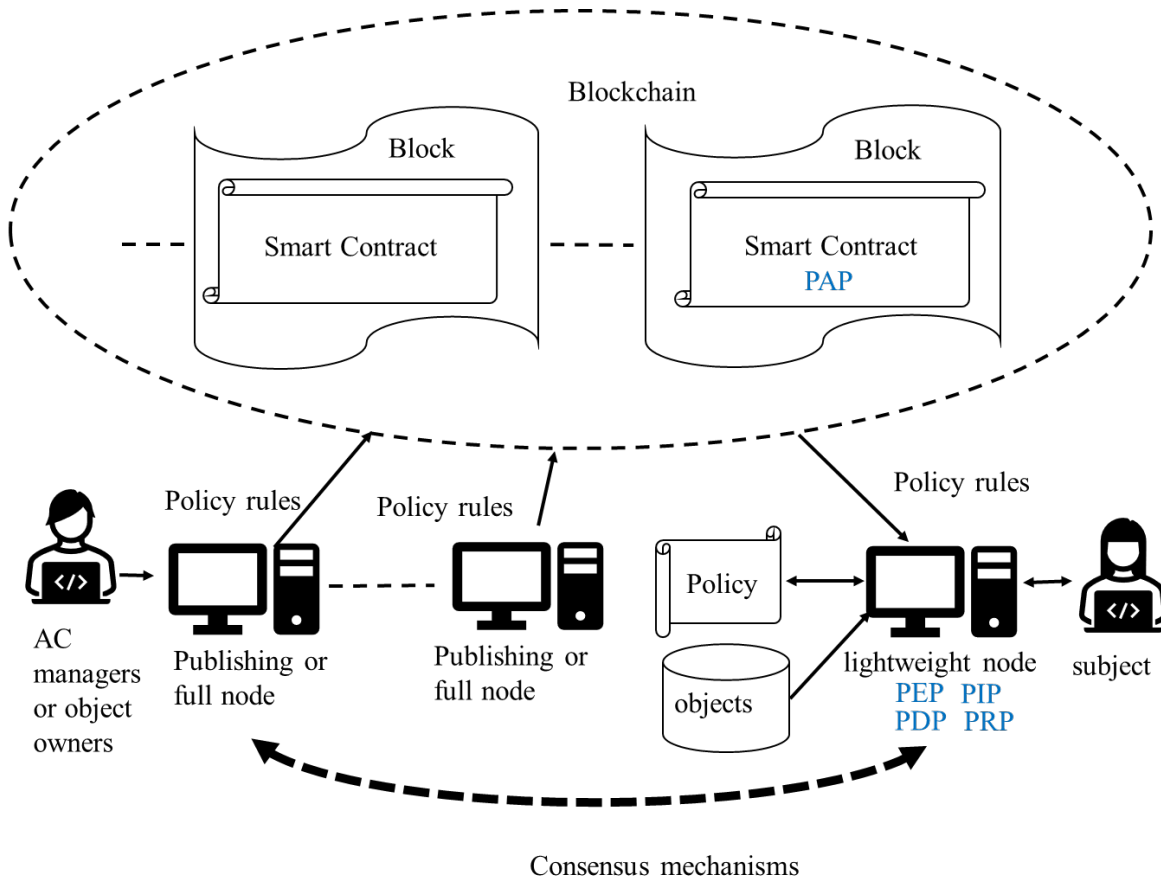
417
418
419

Figure 2a – Example 1 of access control function points implemented in a blockchain system



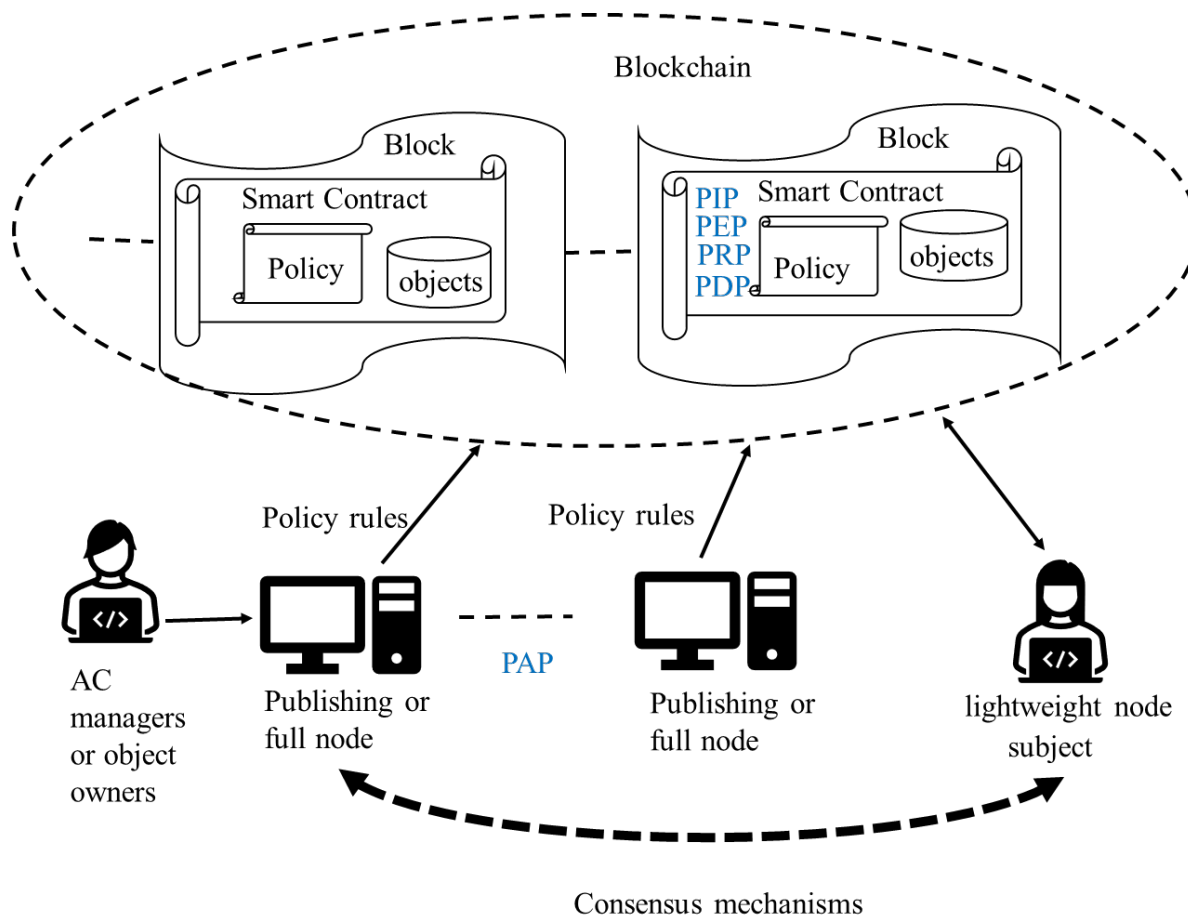
421
422
423

Figure 2b – Example 2 of access control function points implemented in a blockchain system



425
426
427

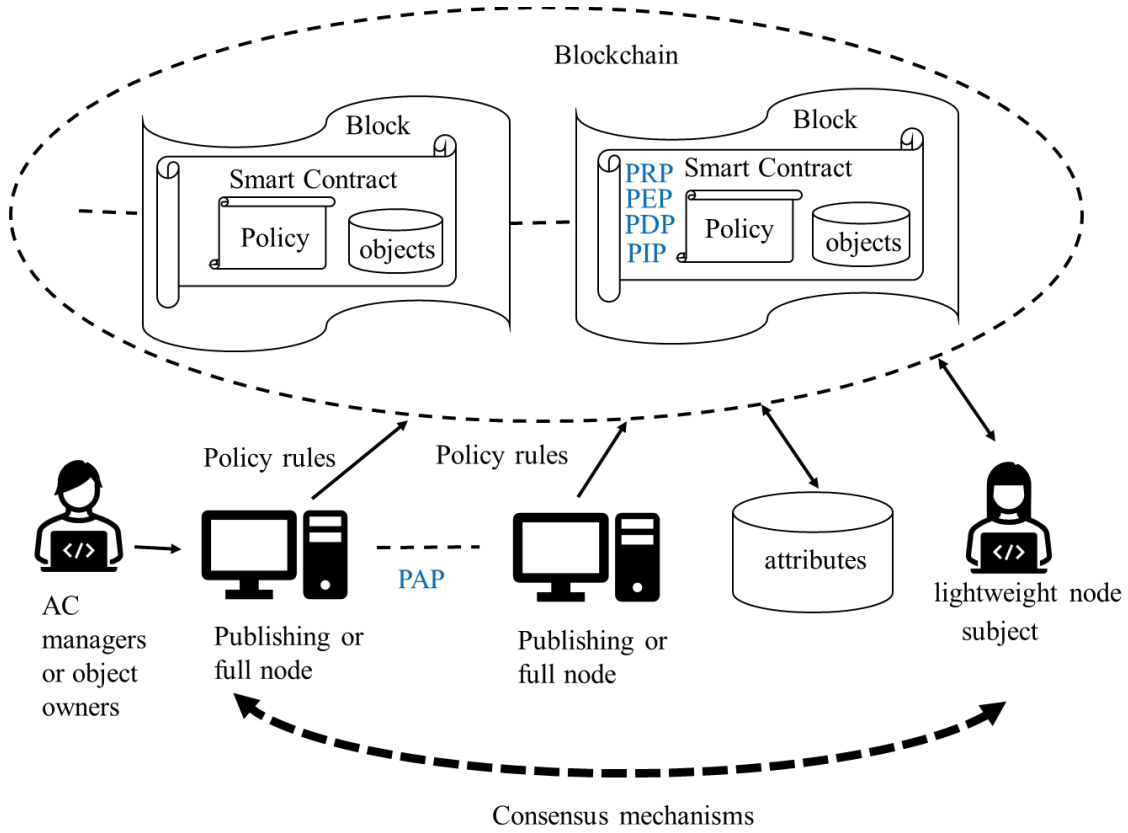
Figure 2c –Example 3 of access control function points implemented in a blockchain system



429 Figure 2d – Example 4 of access control function points implemented in a blockchain system

430 **Figure 2 – Examples of access control function points implemented in blockchain systems**

431 A centralized management AC system, as shown in Figure 2a, requires the blockchain to play the
 432 role of a trusted storage of AC data such that most function points are hosted in a lightweight node
 433 connected to the blockchain to obtain the AC data and current system states. However, it still
 434 inherits the shortcomings of centralization, such as the problem of a single point of failure. In
 435 contrast, Figure 2d shows how the AC system requires decentralized management and adopts
 436 blockchain as a trusted platform to maximize system availability and minimize the possibility of
 437 AC data forgery and tampering. All function points, except for PAP, are implemented in the
 438 blockchain (with smart contracts) that also stores AC data. For this implementation, AC policy
 439 administrators use a publishing node for policy management, and subjects use lightweight nodes
 440 for access requests that will be processed by smart contracts, thereby ensuring that it can be
 441 processed promptly. For conciseness, Figure 2 examples address both the subject and object
 442 attributes associated with the policies (i.e., stored and managed by the same authority). Otherwise,
 443 they can be separately administrated by PIP and PRP either in or out of the chain hosts, as shown
 444 in Figure 3 – an example of options for a federated AC system.
 445



447

Figure 3a – Attribute source is out of the chain

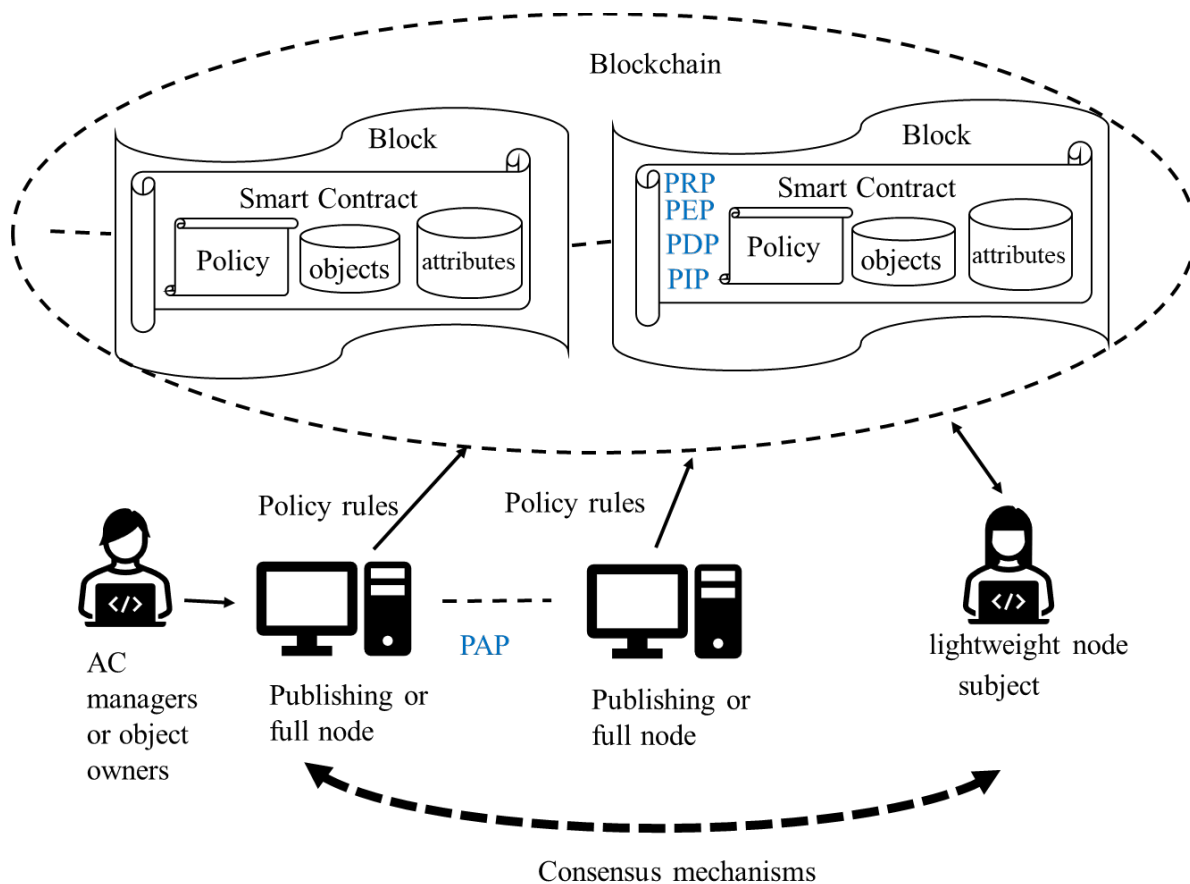


Figure 3b – Attribute source is in the chain

Figure 3 – Examples of Figure 2d with attribute source options

449
 450
 451 Architectures for blockchain AC systems offer flexibility based on the AC policy models enforced,
 452 such as separate blockchain networks for the separation of duty (SoD) policy model or external
 453 expansion of the AC system, which connects off-chain oracles for accessing AC data provided by
 454 third parties. Note that the architecture of a blockchain AC system is independent from the AC
 455 policy models (e.g., ABAC, RBAC, CBAC [Capability Bases AC] [GPR]) that the AC system
 456 intends to apply. For example, if a CBAC model is applied, then the policy rules in Figure 2 should
 457 be replaced by access tokens.

458 To ensure AC data security, functions to satisfy the following three security requirements may also
 459 need to be included [SP205]:

- 460 1. The semantic and syntactic correctness (i.e., veracity of AC data) needs to be ensured or
 461 trusted. If such data is from out-of-the-chain sources, an authority for oracle needs to be
 462 applied to validate and oversee the correctness of the data. However, if it is provided by
 463 different AC nodes, then multiple authorities working in coordination can take part in
 464 validating different sources or functions embedded in smart contracts need to be developed
 465 for the tasks [GMS].

466
467
468
469
470
471
472

473
474
475
476
477
478
479

2. In addition to secure transmission and repositories of AC data in the blockchain, to further avoid compromising the data integrity and confidentiality, inherited hash cypher schemes may be required to avoid exposing vulnerabilities or other types of malicious actions performed by unauthorized entities in AC notes or smart contracts. Smart contracts may also be created to define the secure communications between AC notes for AC data owners, creators, or managers.

3. Cache synchronization and failover/backup capabilities for readiness of the AC system, which refers to the frequency of refresh for AC data change. A blockchain AC system needs to adequately perform AC data update and retrieval frequencies to ensure that a recent set of AC data in question is cached in the blockchain if the most current AC data from authoritative sources or repositories cannot be accessed during an emergency (e.g., low bandwidth, loss of service).

4 Access Control Model Support

481 AC systems are basically categorized as Discretionary AC (DAC), which leaves a certain amount
482 of AC to the discretion of the object's owner or anyone else who is authorized to control the
483 object's access. In general, all AC policies other than DAC are grouped under the category of non-
484 discretionary AC (NDAC). As the name implies, policies in this category have rules that are not
485 established at the discretion of the user. NDAC establishes controls that can only be changed
486 through administrative action, not by subjects. For example, a capability list is a popular model of
487 DAC, and IBAC [IR7316], RBAC, ABAC, and CBAC models are popular examples of NDAC.
488 In general, permissionless blockchains are suitable for DAC implementations, and permissioned
489 blockchains are preferred for NDAC implementations for their control mechanisms.

490
491 As the mandatory nature of NDAC, consensus mechanisms of permissioned blockchains are
492 mostly required so that only permitted AC administrators or security officials are allowed to create
493 and modify AC rules through the restricted publishing of AC nodes, such that the consensus
494 mechanism is restricted to general subjects. Note that the coordination of the permitted AC nodes
495 can be centrally managed by a designed AC node, out-of-the-chain process, or through smart
496 contracts published by authorized administrators.

497
498 For DAC policy models, the consensus mechanism configured for permissionless blockchain
499 needs to be available to all authenticated subjects, who are usually also object owners and who can
500 use publishing AC nodes for managing policy for the authorized objects. However, for a large
501 number of subjects, the mechanism needs to consider performance and operation requirements.
502 For example, in general service environments, the consensus mechanism needs to be fair for
503 generating and updating AC data for all publishing or full AC nodes. The system also needs to
504 ensure that AC nodes can only manage policy rules associated with the object owned by the subject.

505
506 An example policy model that supports NDAC for resource-constrained devices (e.g., size, battery
507 energy, processing speed) on an IoT network is the CBAC, which is relatively lightweight because
508 it uses a communicable and unforgeable token for access rights associated with devices. If the
509 CBAC is implemented by a traditional AC mechanism, it is inefficient to satisfy AC data
510 management and AC processes due to the scale and heterogeneity of IoT devices. The reason for
511 this is that tokens can only be granted to one subject, which makes them difficult to specify
512 centrally and in advance. Further, devices have to use tools provided by a central AC server or a
513 third party to manage their AC data, which may end up with a single point of failure and privilege
514 leakages [BXANL]. These issues can be eliminated by the blockchain system where tokens for
515 AC data management and AC processes are distributed to each IoT device hosted in an AC node.

516
517 [PDA] published a survey of blockchain AC systems compared to traditional AC mechanisms for
518 the implementations of RBAC, ABAC, and CBAC policy models for the IoT AC system. As
519 shown in Table 1, the survey presents capabilities to satisfy the listed general requirements of IoT
520 networks, including scalability, ease of use, data trust, security, and cross-domain control, which
521 are also applicable to other AC systems that enforce the policy models.

522

523 **Table 1 Comparison of IoT AC system capabilities for general access control requirements by blockchain**
 524 **and traditional mechanisms enforcing RBAC, ABAC, and CBAC policy models**

525

AC Requirements	Capabilities of traditional AC mechanisms implementing AC policy models (in parenthesis)	Capabilities of Blockchain AC systems implementing any of the RBAC, ABAC, CBAC models
Scalability	Low (RBAC) , Medium (ABAC), High (CBAC)	High
Ease of use	Medium (RBAC), High (ABAC, CBAC)	High
Architecture	Centralized (RBAC, ABAC), Distributed (CBAC)	Distributed
Data Trust	Low (CBAC), Medium (ABAC), High (RBAC)	High
Continual Control	Medium (RBAC), High (ABAC, CABC)	High
Security	Low (CBAC), Medium (ABAC), High (RBAC)	High
Cross-domain AC	Yes (CBAC), No (RBAC, ABAC)	Yes

526

527 In addition to the static policy models listed in Table 1, dynamic policy models can also be
 528 supported through smart contracts. For example, historical policies regulate access permissions by
 529 historical access states or recorded and predefined series of events. The representative models for
 530 this type of AC policy are Chinese Wall and Workflow [IR7316], which can be best described by
 531 synchronous or direct specification and expressions of a finite state model. For instance, a
 532 synchronous algorithm specified a policy of Chinese Wall model where there are three conflict of
 533 interest groups – C_1 , C_2 , and C_3 – for the access of object groups O_1 and O_2 . Instead, implemented
 534 in a traditional AC mechanism that relies on a central process to monitor each transition of the
 535 entire AC states, the blockchain AC system can specify and enforce the policy via smart contracts,
 536 which every AC node can execute to maintain the policy states. The following is an example
 537 algorithm for the smart contract code for the Chinese Wall policy model.

538

```

539 Contract Chinese_Wall {
540     Public variables {
541         next_state {1,2,3}:= 1;
542         subject_attribute { $C_1$ ,  $C_2$ ,  $C_3$ };
543         object_attribute { $O_1$ ,  $O_2$ };
544         permission {grant, deny};
545         // a FSM of state, subject attribute, object attribute, and permission //
546     }
547     Function Public Access (state, subject attribute, object attribute) {
548         IF next_state == 1;
549         CASE {
550             subject_attribute ==  $C_1$  AND object_attribute ==  $O_1$ : next_state =2;
551             permission = grant;
552             subject_attribute ==  $C_2$  AND object_attribute ==  $O_1$ : next_state =2;
553             permission = grant;
554             subject_attribute ==  $C_3$  AND object_attribute ==  $O_1$ : next_state =2;
555             permission = grant;

```

```

556         subject_attribute == C1 AND object_attribute == O2: next_state =3;
557         permission = grant;
558         subject_attribute == C2 AND object_attribute == O2: next_state =3;
559         permission = grant;
560         subject_attribute == C3 AND object_attribute == O2: next_state =3;
561         permission = grant;
562         OTHERWISE: permission = deny;
563     }
564     IF next_state == 2;
565     CASE {
566         subject_attribute == C1 AND object_attribute == O1: next_state =2;
567         permission = grant;
568         subject_attribute == C2 AND object_attribute == O1: next_state =2;
569         permission = grant;
570         subject_attribute == C3 AND object_attribute == O1: next_state =2;
571         permission = grant;
572         OTHERWISE: permission = deny;
573     }
574     IF next_states == 3;
575     CASE {
576         subject_attribute == C1 AND object_attribute == O2: next_state =3;
577         permission = grant;
578         subject_attribute == C2 AND object_attribute == O2: next_state =3;
579         permission = grant;
580         subject_attribute == C3 AND object_attribute == O2: next_state =3;
581         permission = grant;
582         OTHERWISE: permission = deny;
583     }
584     ELSE permission = deny;
585     RETURN permission;
586 }
587 }
588

```


589 **5 Considerations**

590 This section discusses considerations for the implementation of the blockchain AC system from
591 the perspectives of management, security, privacy, performance, and standardization of AC
592 systems.

593 **5.1 Management Considerations**

594

595 Blockchain AC system management needs to coordinate with the business and resource
596 requirements of the system. For instance, a federated AC system may spread over multiple
597 organizations for cooperation and communication between participating organizations. Hence, AC
598 policies needs to be flexible and fine-grained. A blockchain AC system can transform the policy
599 evaluation process to executable smart contracts so that each organization can control its own
600 system while communicating with other federated organizations. Optionally, some federation
601 scheme may use the blockchain as a database for storing only the policies but not use the
602 blockchain for access enforcement, such that PDP and PEP functions are performed off-the-chain.
603 However, the main problems of the traditional mechanism, like single point of failure, will be
604 inherited [GBHC].

605
606 Another challenge of managing blockchain AC systems is to develop a trust management and
607 evaluation framework for the decentralization of constrained resource systems, such as an IoT
608 network, where each AC node embedded in a device has limited battery power, memory capacity,
609 and processing speed, and it is often impossible to store extensive interaction history or employ
610 heavy-weight security functions (e.g., microservice of mesh service for SecDevOps
611 implementation).

612
613 General AC management requirements, such as allowing runtime policy rule changes and policy
614 administration delegation, may further complicate the design of the blockchain AC system,
615 especially the consensus mechanisms and smart contract functions [IR7874].

616 **5.2 Security Considerations**

617

618 Any vulnerability of a blockchain AC system on the level of the entire system or an underlying
619 function of a smart contract can be hacked (e.g., reentrancy vulnerability) or misused. For instance,
620 the publicly available smart contract's byte code might generate erroneous system state data that
621 will be securely logged on the blockchain. The only way to fix errors is to delete, correct, and
622 redeploy the entire smart contract. Thus, it is necessary that smart contracts are correctly deployed
623 (i.e., they work as intended by the developer and cannot be exploited by attackers).

624
625 Optimizing smart contract codes can effectively reduce potential vulnerabilities and ensure the
626 efficient execution of contracts. For instance, running smart contracts in parallel can speed up
627 contract execution but requires the consideration of how to execute contracts that depend on each
628 other at the same time (especially for dynamic AC policy models). Further, smart contracts might
629 require communicating with out-of-chain services, such as receiving AC data from a PIP host, and
630 reliance on oracle of off-chain resources from trusted third parties to retrieve the data and then
631 push them to the blockchain at predetermined times. Although existing oracles are well-tested,

632 their use may introduce a potential point of failure (e.g., an oracle might be unable to push out or
633 provide erroneous data) [KLG].

634

635 Due to the tamper evident and tamper resistant design of blockchain systems, the system
636 performance evaluation should include extensive and possibly expensive reviews of the smart
637 contract performed by experts before its deployment [SGLSFB et al., KLG]. If smart contracts are
638 required to provide a way to report and correct any errors, then the system should allow actions to
639 nullify and replace a smart contract.

640

641 Protocol of the consensus mechanism is another security concern of vulnerability. For example,
642 PoS mechanisms are vulnerable to attacks such as nothing-at-stake, grinding, long-range, and stake
643 bleeding attacks. PoW and PoS mechanisms may cause low throughput and long transaction
644 confirmation delay, leading to weak consistency problems because an AC process cannot be
645 finalized until its block reaches a certain depth in the blockchain. These might degrade the
646 performance of AC process, so consistency – including common-prefix, chain growth, and chain
647 quality properties of the consensus mechanism – needs to be considered [PDA].

648

649 **5.3 Privacy Considerations**

650 Storing AC data and logs on the blockchain raises questions of privacy as all subjects can see all
651 entries, and auditable access history in the blockchain can violate user privacy. If regulations
652 require AC data owners who are accountable for all data privacy, then instead of storing private
653 data on the blockchain, consider storing index numbers that are tied to private data in an off-the-
654 chain system. Thus, subjects can own, secure, and even delete their privacy data. Otherwise,
655 methods or tools facilitating cryptography need to be considered for privacy protection [GT].

656

657 **5.4 Performance Considerations**

658 The performance of a blockchain AC system should consider process throughput and confirmation
659 delay. The former refers to the number of AC access requests/processes that the AC system can
660 confirm per unit time (e.g., the Ethereum blockchain can verify 14 transactions per second, which
661 is slow compared to Visa, which can handle up to 24,000 transactions per second), while the latter
662 measures the time it takes for them to be finalized. A blockchain AC system may generate a large
663 volume of access requests that need to be processed and handled or a large number of AC nodes
664 that generate a large amount of data to form an oversized chain (e.g., IoT AC system) [ZZH]. In
665 such cases, AC requests/processes may be constrained by the fact that blockchain data can only be
666 added, not deleted. Due to the scalability limitation of the block memory size, a reduction in
667 performance (bottleneck for the end users) is inevitable. The consequences will be increased
668 synchronization time, increased commission fees (if required), and increased time to confirm an
669 AC request/process [PDA, KLG].

670

671 Scalability is another performance concern, and one of the major affecting factors is the consensus
672 mechanism's consistency and liveness. Consistency means that legitimate AC nodes have an
673 identical view of the AC system state, and liveness means that a valid AC process is sure to be
674 processed and written on the blockchain for a certain period. To address these issues, consensus
675 mechanism can be adjusted to decrease resource consumption, especially for resource-constrained
676 AC systems such as IoT AC systems (it has some disadvantages on security regarding to

677 immutability). For example, for the AC system uses a permissionless type of blockchain. Although
678 the PoW algorithm enables security in the blockchain, it wastes resources. Thus, consider
679 switching from the PoW algorithm to others, such as proof-of-activity (PoA) or delegated proof-
680 of-stake (DPoS), that can improve scalability as well as lower fees and energy costs (if required)
681 for AC processes [GBHC, KLG]. The selection of consensus mechanisms also needs to comply
682 with the AC policy models applied.

683

684 As a result, a blockchain AC system must consider hardware limitations in order to economically
685 design an architecture for its memory that is lightweight with limited computing power and storage
686 capabilities, especially for systems with massive AC nodes. Considerations must also include a
687 fast response consensus mechanism to comply with performance requirements.

688

689 **5.5 Standardization Considerations**

690 A blockchain AC system may handle a variety of devices, infrastructures, and governments. For
691 example, a system has different types of AC data (e.g., types of subject or object attribute values)
692 or proprietary protocols between AC nodes that make it difficult to communicate using a single
693 blockchain platform. Thus, assurance and standardization of the guidelines allow for universal
694 acceptance of the AC data and smart contracts for AC processing [RGD, PDA].

6 Conclusion

696 The rapid development and wide application of distributed network systems have made network
697 security – especially access control and data privacy – ever more important. Blockchain
698 technology offers features such as decentralization, high confidence, and tamper-resistance, which
699 are advantages to solving auditability, resource consumption, scalability, central authority, and
700 trust issues – all of which are challenges for network access control by traditional mechanisms.

701 Blockchain is particularly applicable to access control for network systems, where authorization
702 processes are based on subject and object attribute data, because it improves security, flexibility
703 and scalability for management, and enforcement of access control data and processes. It also
704 improves the capability of organizations to verify and audit access control processes with function
705 calls to track the global access control system state. Blockchain system components can function
706 as a resource repository or executable process, allowing it to be neutral for access control policy
707 models. As blockchain access control systems address some challenges from traditional
708 mechanisms, the management, security, privacy, performance, and standardization of the
709 implementation need to be considered.

710
711 This document presents general information for blockchain access control systems from the views
712 of blockchain system properties, components, functions, and supports for access control policy
713 models. Considerations for implementing blockchain AC systems are also included.

References

- 714
- 715 [BXANL] Bouras MA, Xia B, Abuassba AO, Ning H, Lu Q (2021) IoT-CCAC: A
716 Blockchain-Based Consortium Capability Access Control Approach for IoT.
717 *PeerJ Computer Science* 7:e455. <https://doi.org/10.7717/peerj-cs.455>
718
- 719 [DMMR] Di Francesco Maesa D, Mori P, Ricci L (2019) A Blockchain Based Approach for
720 the Definition of Auditable Access Control Systems. *Computers & Security*
721 84(July):93-119. <https://doi.org/10.1016/j.cose.2019.03.016>
722
- 723 [GBHC] Ghaffari F, Bertin F, Hatin J, Crespi N (2020) Authentication and Access Control
724 Based on Distributed Ledger Technology: A survey. *2nd conference on*
725 *Blockchain Research & Applications for Innovative Networks and Services*
726 *(BRAINS 2020)* (IEEE, Paris), pp 79-86.
727 <https://doi.org/10.1109/BRAINS49436.2020.9223297>
728
- 729 [GMS] Guo H, Meamari E, Shen CC (2019). Multi-Authority Attribute-Based Access
730 Control with Smart Contract. *Proceedings of the 2019 International Conference*
731 *on Blockchain Technology (ICBCT 2019)* (ACM, Honolulu, Hawai'i), pp 6–11.
732 <https://doi.org/10.1145/3320154.3320164>
733
- 734 [GPR] Gusmeroli S, Piccione S, Rotondi D (2013) A Capability-Based Security
735 Approach to Manage Access Control in the Internet of Things. *Mathematical and*
736 *Computer Modelling* 58(5–6), pp 1189-1205.
737 <https://doi.org/10.1016/j.mcm.2013.02.006>
738
- 739 [GT] Grant Thornton US (2020) Blockchain and Privacy: How Do You Protect Data
740 That's Distributed?[video]. Available at
741 <https://www.youtube.com/watch?v=hcXz3EQoDF8>
742
- 743 [INCITS] InterNational Committee for Information Technology Standards (2020) *INCITS*
744 *565-2020 – Information technology – Next Generation Access Control* (INCITS,
745 Washington, DC). Available at
746 [https://standards.incits.org/apps/group_public/project/details.php?project_id=232](https://standards.incits.org/apps/group_public/project/details.php?project_id=2328)
747 [8](https://standards.incits.org/apps/group_public/project/details.php?project_id=2328)
748
- 749 [IR7316] Hu VC, Ferraiolo DF, Kuhn DR (2006) Assessment of Access Control Systems.
750 (National Institute of Standards and Technology, Gaithersburg, MD), NIST
751 Interagency or Internal Report (IR) 7316. <https://doi.org/10.6028/NIST.IR.7316>
752
- 753 [IR7874] Hu VC, Scarfone KA (2012) Guidelines for Access Control System Evaluation
754 Metrics. (National Institute of Standards and Technology, Gaithersburg, MD),

- 755 NIST Interagency or Internal Report (IR) 7874.
756 <https://doi.org/10.6028/NIST.IR.7874>
757
- 758 [IR8202] Yaga DJ, Mell PM, Roby N, Scarfone KA (2018) Blockchain Technology
759 Overview. (National Institute of Standards and Technology, Gaithersburg, MD),
760 NIST Interagency or Internal Report (IR) 8202.
761 <https://doi.org/10.6028/NIST.IR.8202>
762
- 763 [KLG] Khan SN, Loukil F, Ghedira-Guegan C, Benkhelifa E, Bani-Hani A (2021)
764 Blockchain Smart Contracts: Applications, Challenges, and Future Trends. *Peer-*
765 *to-Peer Networking and Applications* 14:2901-2925.
766 <https://doi.org/10.1007/s12083-021-01127-0>
767
- 768 [Kuhn] Kuhn DR, (2018) A Data Structure for Integrity Protection with Erasure
769 Capability. (National Institute of Standards and Technology, Gaithersburg, MD),
770 Draft NIST Cybersecurity White Paper. Available at
771 [https://csrc.nist.gov/CSRC/media/Publications/white-paper/2018/05/31/data-](https://csrc.nist.gov/CSRC/media/Publications/white-paper/2018/05/31/data-structure-for-integrity-protection-with-erasure-capability/draft/documents/data-structure-for-integrity-with-erasure-draft.pdf)
772 [structure-for-integrity-protection-with-erasure-capability/draft/documents/data-](https://csrc.nist.gov/CSRC/media/Publications/white-paper/2018/05/31/data-structure-for-integrity-protection-with-erasure-capability/draft/documents/data-structure-for-integrity-with-erasure-draft.pdf)
773 [structure-for-integrity-with-erasure-draft.pdf](https://csrc.nist.gov/CSRC/media/Publications/white-paper/2018/05/31/data-structure-for-integrity-protection-with-erasure-capability/draft/documents/data-structure-for-integrity-with-erasure-draft.pdf)
774
- 775 [LQLL] Liu Y, Qiu M, Liu J, Liu M (2021) Blockchain Based Access Control
776 Approaches. 8th IEEE International Conference on Cyber Security and Cloud
777 Computing (CSCloud)/2021 7th IEEE International Conference on Edge
778 Computing and Scalable Cloud (EdgeCom) (IEEE, Washington, DC), pp 127-
779 132. <https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00032>
780
- 781 [PDA] Pal S, Dorri A, Jurdak R (2021) Blockchain for IoT Access Control: Recent
782 Trends and Future Research Directions. *arXiv preprint*.
783 <https://arxiv.org/abs/2106.04808>
784
- 785 [RBAC] Ferraiolo D F, Kuhn D R (1992) Role-Based Access Controls. *Proceedings of the*
786 *15th National Computer Security Conference* (NIST, Baltimore, MD), pp 554-563.
787 Available at [https://csrc.nist.gov/CSRC/media/Publications/conference-](https://csrc.nist.gov/CSRC/media/Publications/conference-paper/1992/10/13/role-based-access-controls/documents/ferraiolo-kuhn-92.pdf)
788 [paper/1992/10/13/role-based-access-controls/documents/ferraiolo-kuhn-92.pdf](https://csrc.nist.gov/CSRC/media/Publications/conference-paper/1992/10/13/role-based-access-controls/documents/ferraiolo-kuhn-92.pdf)
789
- 790 [RGD] Rani PL, Guru Gokul AR, Devi N (2021) Blockchain-Based Access Control
791 System. *Transforming Cyber Security Solution Using Blockchain*, eds Agrawal R,
792 Gupta N (Springer, Singapore), pp 91-114. [https://doi.org/10.1007/978-981-33-](https://doi.org/10.1007/978-981-33-6858-3_6)
793 [6858-3_6](https://doi.org/10.1007/978-981-33-6858-3_6)
794
- 795 [SGLSFB] Schiff J, Grundmann M, Leinweber M, Stengele O, Friebe S, Beckert B (2021)
796 Towards Correct Smart Contracts: A Case Study on Formal Verification of
797 Access Control. *Proceedings of the 26th ACM Symposium on Access Control*

- 798 *Models and Technologies (SACMAT '21)* (ACM, [Virtual], Spain) pp 125–130.
799 <https://doi.org/10.1145/3450569.3463574>
800
- 801 [SP162] Hu VC, Ferraiolo DF, Kuhn DR, Schnitzer A, Sandlin K, Miller R, Scarfone KA
802 (2014) *Guide to Attribute Based Access Control (ABAC) Definition and*
803 *Considerations*. (National Institute of Standards and Technology, Gaithersburg,
804 MD), NIST Special Publication (SP) 800-162, Includes updates as of August 02,
805 2019. <https://doi.org/10.6028/NIST.SP.800-162>
806
- 807 [SP205] Hu VC, Ferraiolo DF, Kuhn DR (2019) *Attribute Considerations for Access*
808 *Control Systems*. (National Institute of Standards and Technology, Gaithersburg,
809 MD), NIST Special Publication (SP) 800-205.
810 <https://doi.org/10.6028/NIST.SP.800-205>
811
- 812 [XACML] OASIS eXtensible Access Control Markup Language (XACML) TC (2020)
813 *Organization for the Advancement of Structured Information Standards*.
814 Available at [https://www.oasis-](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml)
815 [open.org/committees/tc_home.php?wg_abbrev=xacml](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml)
816
- 817 [ZZH] Zhai P, Zhang L, He J (2021) A Review of Blockchain-Based Access Control for
818 the Industrial IoT. *CONVERTER* 2021(3):308-316.
819 <https://doi.org/10.17762/converter.62>