energies

MDPI

Article

Centralized and Decentralized Distributed Energy Resource Access Control Implementation Considerations

Georgios Fragkos [1], Jay Johnson [2] and Eirini Eleni Tsiropoulou [1,*]

[1] Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, USA
[2] Sandia National Laboratories, Albuquerque, NM 87185, USA
* Correspondence: eirini@unm.edu

Abstract: A global transition to power grids with high penetrations of renewable energy generation is being driven in part by rapid installations of distributed energy resources (DER). New DER equipment includes standardized IEEE 1547-2018 communication interfaces and proprietary communications capabilities. Interoperable DER provides new monitoring and control capabilities. The existence of multiple entities with different roles and responsibilities within the DER ecosystem makes the Access Control (AC) mechanism necessary. In this paper, we introduce and compare two novel architectures, which provide a Role-Based Access Control (RBAC) service to the DER ecosystem's entities. Selecting an appropriate RBAC technology is important for the RBAC administrator and users who request DER access authorization. The first architecture is centralized, based on the OpenLDAP, an open source implementation of the Lightweight Directory Access Protocol (LDAP). The second approach is decentralized, based on a private Ethereum blockchain test network, where the RBAC model is stored and efficiently retrieved via the utilization of a single Smart Contract. We have implemented two end-to-end Proofs-of-Concept (PoC), respectively, to offer the RBAC service to the DER entities as web applications. Finally, an evaluation of the two approaches is presented, highlighting the key speed, cost, usability, and security features.

Keywords: distributed energy resources; Role-Based Access Control; blockchain; Ethereum; web application

Citation: Fragkos, G.; Johnson, J.; Tsiropoulou, E.E. Centralized and Decentralized Distributed Energy Resource Access Control Implementation Considerations. *Energies* **2022**, *15*, 6375. https://doi.org/10.3390/en15176375

Academic Editor: Pavlos S. Georgilakis

Received: 20 July 2022
Accepted: 28 August 2022
Published: 1 September 2022

1. Introduction

The advent of the Internet of Things (IoT) paradigm expanded the number of connected devices. The projected expansion of the IoT to the Smart Grid by enabling the interoperability functions in distributed energy resources (DER), distributed energy equipment communications to cloud infrastructures, and associated collection of sensitive data can lead to malicious control, critical information leakage, and privacy issues. A representative example of the increasing scale of interoperable distributed energy infrastructure is that the number of Smart Meters (SMs) within the IoT-based US Smart Grid is almost 90 million and the corresponding expected penetration in the market by 2021 is 90% [1]. Regardless, distributed energy resources connected to the internet like IoT devices can be considered a robust ecosystem and safely adopted by the Smart Grid community depending on its capability to prevent cyberattacks.

1.1. Background & Motivation

Many well-documented IoT security failures have plagued the industry including the creation of botnets (e.g., Mirai) and compromises of home networks [2]. To address these issues there has been significant work in risk management frameworks, securing logical access with, e.g., blockchain technologies [3], incorporating embedded hardware-based authentication with, e.g., Physical Unclonable Functions [4], and key management

for IoT Public Key Infrastructure (PKI) [5]. Authentication and access control is one of the areas highlighted as a key security requirement in the IoT environment [6]. Access Control (AC) can selectively restrict access to data via authorization after authenticating the user. Authentication refers to identifying users within the Smart Grid network by utilizing username-password credentials, security tokens, and other mechanisms. Then, the AC mechanism enables the authorization process to specify the users' privileges to the resources [7].

Focusing on the Smart Grid systems, the distributed energy resources (DER), such as diesel engines, microturbines, fuel cells, photovoltaic, small wind turbines, and others [8], along with the integrated communications and computing systems, can manage countless operations of the smart grid system [9]. Due to the constantly evolving IoT-based Smart Grid domain with a large number of heterogeneous interconnected devices, there has been increasing interest from the research and industrial communities in terms of proposing both sophisticated centralized and decentralized AC models. Towards mitigating potential attacks and ensuring the smooth operation of such a complex environment, several AC models have been introduced in the recent literature, such as Mandatory Access Control (MAC) [10], Discretionary Access Control (DAC) [11], Role-Based Access Control (RBAC) [12], Attribute-Based Access Control (ABAC) [13], and others. Under the MAC, a centralized authority, e.g., a security policy administrator, control the access privileges of the individual users, while the latter ones cannot override or modify the access policy. Thus, a central security policy is defined and is guaranteed to be enforced for all individual users. On the other hand, the DAC provides the ability to the individual users to perform policy decisions and assign security attributes, thus, providing greater flexibility to the overall system. Furthermore, the ABAC model considers the attributes associated with the users, objects that they try to access, requested operations, and the examined environment, in order to provide access privileges.

The main advantage of the RBAC model over newer alternatives, such as ABAC, is that it efficiently simplifies the user-to-rights mapping in the environment by establishing the role abstraction layer between the users and objects. In particular, defining and implementing roles and policies is much simpler and faster than assigning distinct attributes to every user within a complex organization. A common direction for organizations is to first design and build an RBAC model as a high-level authorization mechanism before including additional environmental variables through an ABAC system since implementing the latter often requires more resources and processing power. Moreover, some considered RBAC models with attributes over ABAC with role names as attributes, since ABAC fails to audit user access to specific permissions [14]. Thus, the RBAC model is also recommended in power system application spaces as indicated in IEC 62351-8 [15]. For these reasons, we focus our current study on the RBAC model. Notably, this is the first research paper in which such a model is implemented both in a centralized and decentralized manner and evaluated with respect to multiple metrics for the DER environment to effectively provide access to users to the equipment control settings or data in an automatic manner. The evaluation metrics from this work, i.e., security, speed, cost, and usability, revealed tradeoffs between the two approaches that would help system administrators choose an implementation based on their business needs. In the future, the same methodology and design choices could be followed for an ABAC implementation using users, environment, and resource attributes.

1.2. Related Work

Focusing on the *centralized AC models* for power systems applications, in [16], the authors introduce a certificate-based authentication scheme for securing demand response management, which is considered secure by utilizing the Real-Or-Random model. In [17], the authors present an XACML-based ABAC framework, which was extended in [18] to a global authorization component consisting of a hierarchical architecture that considered the Smart Grid's cloud security state. In [19], the authors have integrated an RBAC model along

with the Meter Data Management (MDM) database, which resides at the utility control center of the smart grid service provider, in order to guarantee the protection of the smart meters' data. Similarly, an RBAC model supporting the wind power systems is developed in [20]. The proposed model aims at mitigating the attacks from different external entities by complementing two existing Internal Energy Commission (IEC) standards, i.e., IEC 61400-25 and IEC 62351. The two IEC standards represent elements of the wind power infrastructure in a software domain and establish secure communication and authentication of the other parties in electrical power infrastructures, without addressing the problem of access control. All of the aforementioned research papers that examine centralized AC models, do not apply them to the complex DER environment where the power systems are geographically and logically distributed nor do they provide a comparative evaluation with alternative distributed solutions. Our research work aims to fill this research gap by designing and implementing an end-to-end centralized architecture for an RBAC model within a DER ecosystem by utilizing already existing open-source frameworks and also comparing it with a corresponding decentralized architecture. The comparative evaluation takes into consideration multiple metrics that can assist the DER system administrators to select a single approach based on their needs.

Decentralized AC models are mainly based on the blockchain data structure. A blockchain-based AC protocol is introduced in [21], where the data of the consumers' smart meters are stored in a private blockchain and securely transferred to the service providers. A comparable approach is adduced in [22] but consumer information for energy trading is encrypted by deploying a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) algorithm. A similar approach is followed in [23], where the decryption of the CP-ABE is performed at the edge nodes instead of the cloud, in order to further improve the efficiency and safety of the distributed environment. In [24], a proxy re-encryption technique is proposed based on the blockchain to enable secure data sharing. In [25], the utilization of the blockchain data structure is introduced in combination with an identity-based combined encryption, signature, and signcryption scheme for a secure AC scheme. A three-layer blockchain-based architecture is examined in [26] to produce a privacy-preserving AC model for federated Smart Grid domains. The Permissioned Blockchain Edge Model for Smart Grid Network (PBEM-SGN) is presented in [27], where blockchain is combined with edge computing to guarantee privacy protection and energy security. A decentralized keyless signature scheme is proposed in [28], based on a consortium blockchain to manage the keys of the service providers and smart meters. However, all of the aforementioned decentralized approaches utilize the blockchain data structure to only encrypt, store and transfer the users' data between the different entities of the system under examination. This research work not only focuses on preserving the anonymity of the users within the DER environment but also implements the whole RBAC policy in a distributed manner, i.e., through a single Smart Contract that resides in the blockchain data structure and is accessible by the system administrators. Finally, this approach is further evaluated and compared with a centralized RBAC-based architecture.

1.3. Contributions & Outline

The inclusion of renewable energy in the electric energy systems is of critical importance due to environmental reasons. The shift towards this new paradigm of energy generation is mainly driven by the wide adoption of DER, such as solar photovoltaics, energy storage systems, and fuel cells. Core cybersecurity principles embedded in traditional large generating plant Supervisory Control and Data Acquisition (SCADA) control networks must be accomplished in new ways in the case of DER, as the underlying communication architectures are different [29]. Specifically, there are multiple entities within the non-federated DER ecosystem with varying roles and responsibilities, each needing specific levels of access to DER data and/or control modes [30]. For instance, DER vendors will likely push firmware updates and may advise maintenance schedules by monitoring operations; grid operators need to change the operating modes of the DER equipment, but

DER owners only need access to a subset of operational data like solar generation power and DER status.

In this paper, we present an RBAC implementation for the DER ecosystem. To the best of our knowledge, this is the first research work in which an RBAC model is implemented and evaluated for the DER environment to selectively provide access to users to the equipment control settings or data. Prior work by the SunSpec/Sandia DER Cybersecurity Workgroup concluded that RBAC was a natural design choice for the DER AC ecosystem because there are distinct roles for the users based on their operational responsibilities defined by their job position and company of employment [30]. We introduce two full-stack, end-to-end, Proof of Concepts (PoCs) that provide ecosystem-wide authorization mechanisms. These implementations adopt zero-trust principles and allow users from multiple, dissociated organizations to interact with DER equipment with the appropriate levels of access. The first PoC is centralized based on the open-source OpenLDAP [31] and JXplorer frameworks. The second approach uses an Ethereum blockchain decentralized database in a real-time Ethereum test network not only to preserve the users' anonymity within the system but also to host the RBAC policy as a single Smart Contract within the blockchain data structure. Both approaches constitute the first attempt to build end-to-end authorization architectures for a non-federated DER environment. Also, another main novelty of this research work is the detailed comparison of the two architectures in order to thoroughly present their advantages and disadvantages with respect to multiple evaluation metrics, i.e., security, speed, cost, and usability. The evaluation results revealed multiple tradeoffs regarding those axes which can be considered as criteria for the system administrators to select a single approach for their DER environment. A brief summary of the main contributions is presented in Table 1.

The rest of the paper is organized as follows. In Section 2, we adduce an overview of the considered use case and system components. Sections 3 and 4 present the implementation details of the centralized and decentralized architectures. Section 5 provides experimental results and a discussion of the tradeoffs of the approaches. Section 6 concludes the paper.

Table 1. Summary of the main contributions.

Approaches	*Contributions*
Centralized	- Utilized already existing open-source authorization frameworks, i.e., OpenLDAP and JXplorer, for the DER environment
Decentralized	- Utilized Ethereum blockchain as an authorization medium for the DER environment - Designed and implemented the whole RBAC policy for the DER environment as a Smart Contract within the Ethereum blockchain
Overview	- RBAC model is implemented and evaluated for the DER environment - Comparative evaluation between the centralized and decentralized approaches - Tradeoffs are discussed with respect to the security, cost, speed, and usability of the approaches

2. Use Case Overview & System Components

This paper introduces two novel and efficient architectures that utilize a variety of software technologies to produce DER RBAC web services. We consider a DER topology with several Utility companies that have connections to DER devices; DER Vendors who manufacture the DER equipment; DER Installers who commission the equipment; DER Aggregators who monitor and control DER devices to provide grid services to the utilities, e.g., DER aggregation; DER Service Providers gather DER operational data to share with DER users typically through smartphone applications or other web services; and DER owners who may access the DER data directly or through a DER vendor or DER Service

Provider system. Multiple installers, vendors, service providers, and aggregators may operate within the jurisdictional regions of the utilities. These companies will also be installing equipment in other regions, so there are no clear hierarchical structures that can be built for DER access management, as shown in Figure 1.

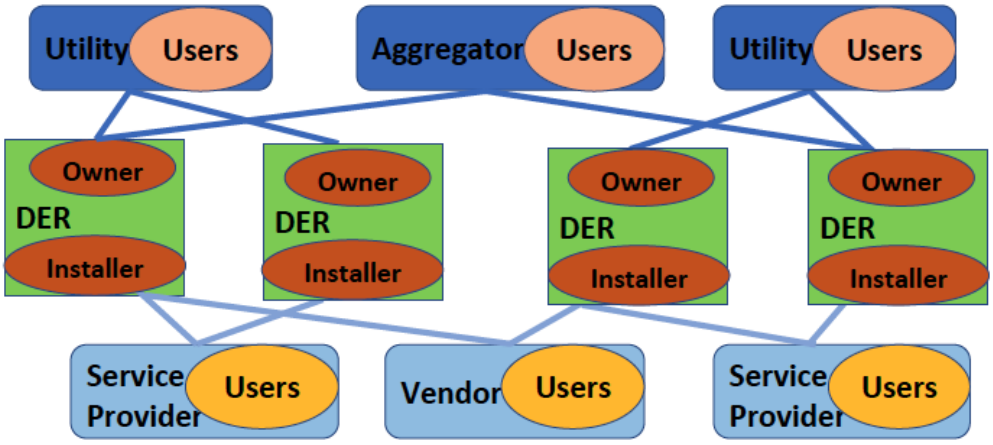

Figure 1. DER Ecosystem Topology.

We model a flat RBAC model to ensure secure AC for the users. Each user may have one or more of the following roles: Utility DER Management System Team, Utility Billing, Utility Auditing, Utility Software, DER Installer, DER Aggregator, DER Firmware/Patching, DER Service Provider Billing, and DER Owner. Based on these roles, each user acquires the corresponding set of permissions, which can be described as read/write operations to the registers of the DER devices with respect to the DER communication protocols. For example, a user is assigned to the Aggregator role and has direct read/write access permissions to the DER Modbus registers, IEEE 1815 analog and binary outputs, or IEEE 2030.5 server on the DER vendor portal to provide grid services to the utility [30]. Figure 2 shows the considered RBAC model.

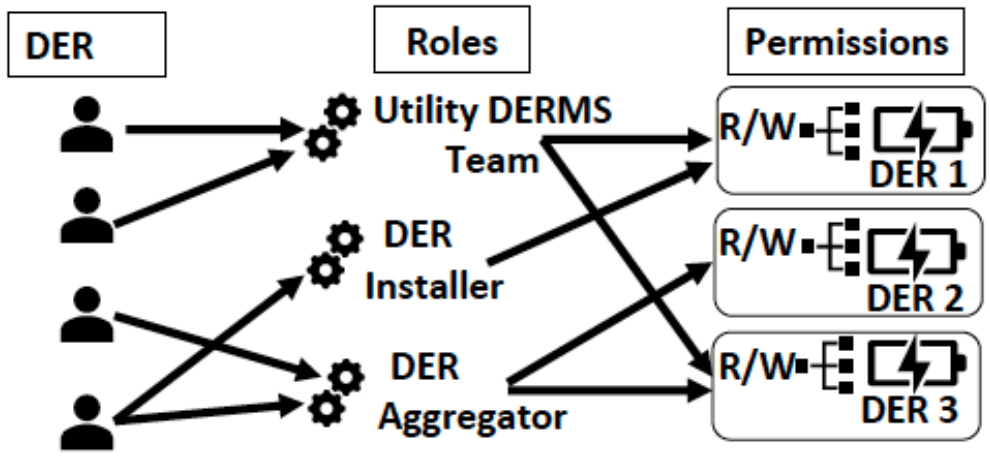

Figure 2. RBAC Model.

The users who could benefit the most from the introduced frameworks are both the DER administrators, who can perform Create, Read, Update, and Delete (CRUD) operations to the existing RBAC model, and those who have one or more of the aforementioned roles and ask for access authorization. In the latter case, we assume a Push RBAC model, where the successfully authenticated users request an authorization token from the RBAC provider, e.g., OpenLDAP or Blockchain, and they submit it to the DER object that either validates it or not. Upon successful validation, the user is allowed to perform the requested

operations on the object, based on the permissions that are acquired from the corresponding role. Although, this assumption does not exclude the utilization of a Pull RBAC model, since the only change would be that the authorization request is made by the object to the RBAC provider on behalf of the user. Thus, the proposed architectures provide a universal authorization mechanism independent of the organizations operating in the DER ecosystem.

The proposed system's core architectural components are:

- Web-service: In both the centralized and decentralized architectures, a web application is offered to the DER users, which runs as a Service. A GUI (Figures 3–5), based on the open-source VueJS and Bootstrap frameworks [32], is provided and hosted in a NodeJS web server. The users interact with the RBAC model in a way that offers transparency to their HTTP CRUD or authorization requests. The CRUD interfaces for the DER administrators include the following: (a) update the information entities, (b) revoke roles, (c) show or confirm user permissions, (d) search, add, or delete users, (e) search or add DER device, (f) verify a user-to-role assignment, and (g) find information and statistics about the RBAC provider. Their HTTP requests are sent asynchronously to a Python Flask RESTful endpoint via the JavaScript library AxiosJS [33].
- Centralized Approach: We utilized an open-source implementation of the Lightweight Directory Access Protocol (LDAP) called OpenLDAP. All the user-to-roles and role-to-permissions assignments of the RBAC model were efficiently stored and queried in the standalone centralized server provided by OpenLDAP. The respective codebase is open-sourced and can be found in [34].
- Decentralized Approach: We deployed a private Ethereum blockchain test network, where each entity within the DER ecosystem is assigned a unique Ethereum account. RBAC logic of the DER environment was stored in Smart Contracts. The RBAC model could be recalled using Ethereum search functions. The respective codebase is open-sourced and can be found in [35].

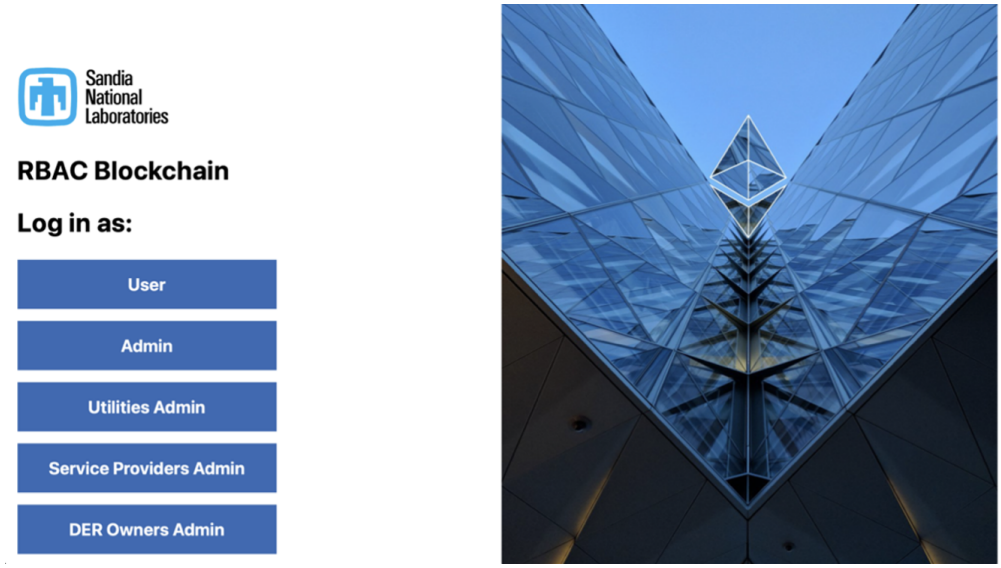

Figure 3. Ethereum Blockchain GUI: A user can determine how they want to login into the system (User or Administrator)

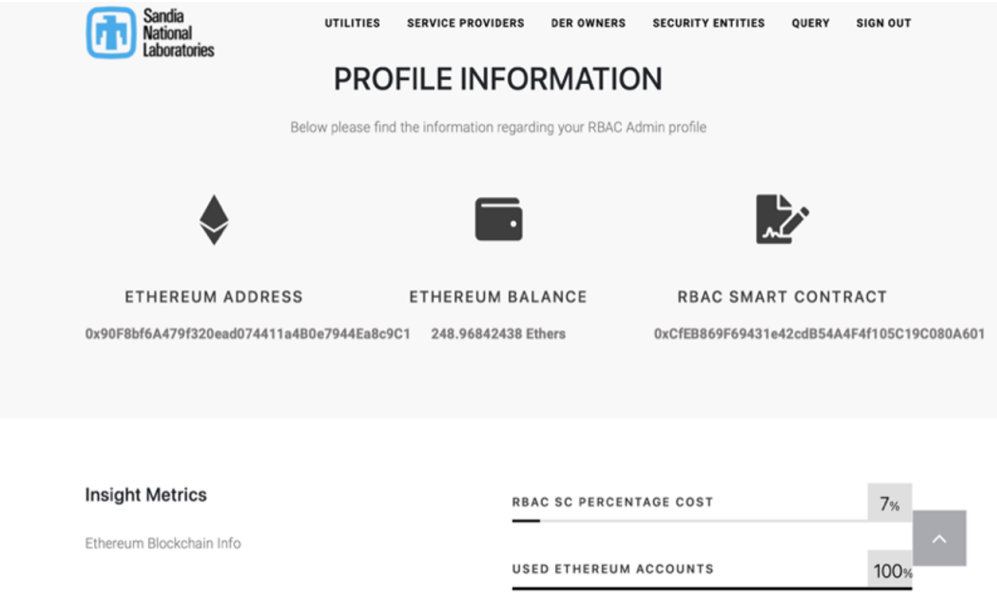

Figure 4. Ethereum Blockchain GUI: The logged user sees information about the Ethereum Blockchain

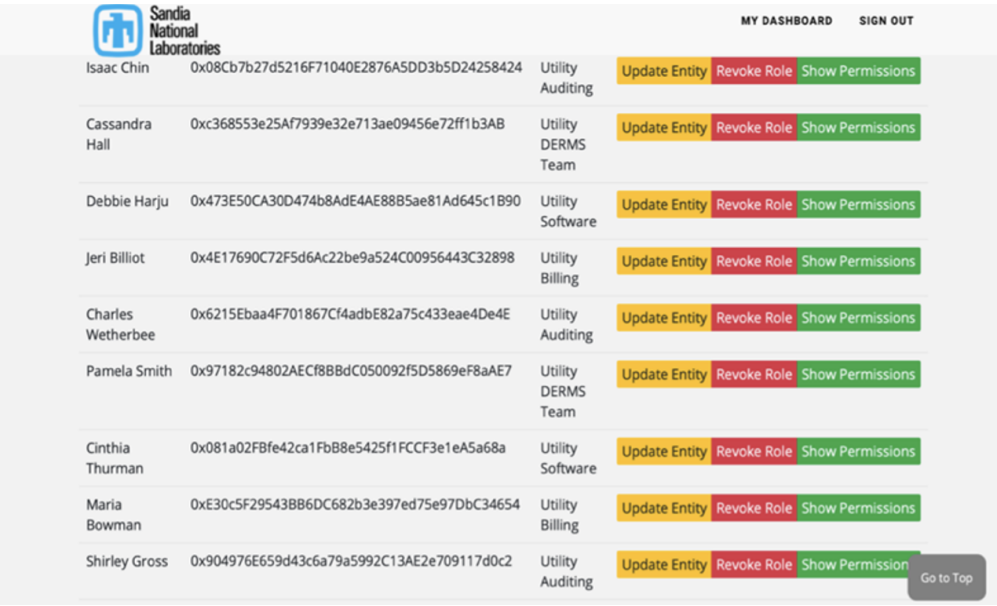

Figure 5. Ethereum Blockchain GUI: The user chooses a Utility or a Service Provider company to perform CRUD operations to the associated users

3. Implementing RBAC in DER Using OpenLDAP

In the centralized approach, we deployed OpenLDAP in the back-end of our architecture (Figure 6), in combination with the front-end and Flask RESTful API. This provides the centralized AC mechanism, which mediates read/write access to the registers of the DER resources. OpenLDAP utilizes the Oracle Berkeley DB, which is a family of embedded key-value database libraries storing the AC policy regarding the DER resources in a hierarchical structure, i.e., the Directory Information Tree (DIT). DIT consists of nodes, each of which has only one parent node and multiple child nodes. There are intermediate nodes that exist to schematize the aforementioned nodes as logical groups. Every node at the DIT is characterized by an identifier, i.e., "Object Class = id", such as *dc* (domain component), *ou* (organizational unit) or *cn* (canonical name). However, every child node includes the identifier of its parent in its own identifier, which is known as a Distinguished Name (DN). We can think of the DN as the absolute path of a resource within the OpenLDAP server.

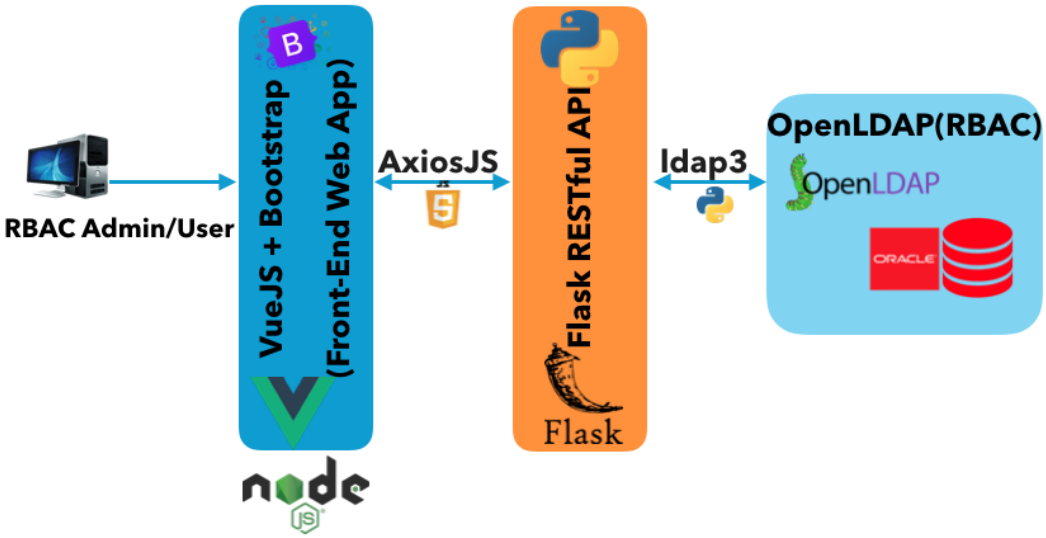

Figure 6. Centralized RBAC Architecture.

We have represented the roles, users, and DER resources in a DIT format (Figure 7). We have the root node, i.e., "dc=my-domain,dc=com", which is the domain of the OpenLDAP server and its child is a Utility company with a DN: "ou=Utility,dc=my-domain,dc=com". The Utility company has two associated users with DNs in the format "cn=Alice,ou=Utility, dc=my-domain,dc=com". Each user has an assigned role; for instance Alice has the Utility Software role ("cn=Utility Software,cn=Alice,ou=Utility,dc=my-domain,dc=com"). The leaf nodes are the DER models that define the corresponding permissions of the RBAC model.

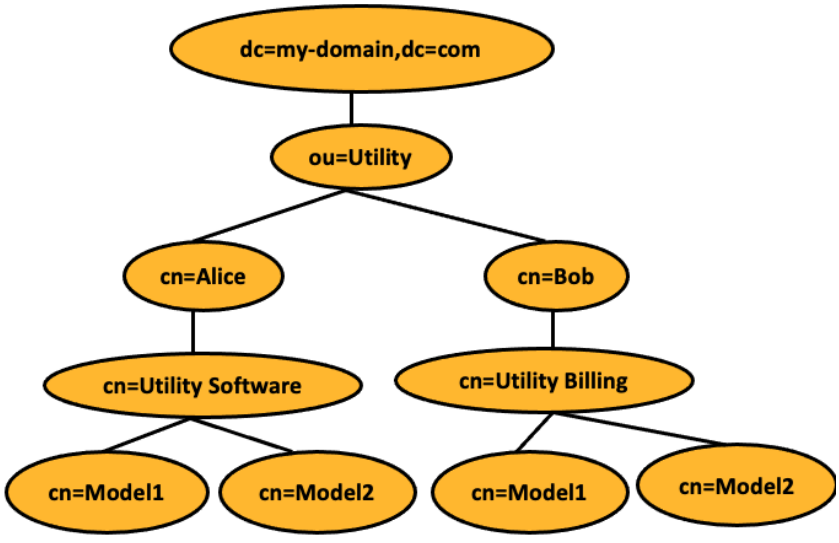

Figure 7. DIT Format Example of a DER ecosystem.

A screenshot of the OpenLDAP DER DIT structure is presented in Figure 8, where we have utilized JXplorer. JXplorer is an open-source implementation of the LDAP developed in JAVA, which enables us to visualize the DIT of the DER RBAC model that resides in the OpenLDAP server. We observe that the tree format that was described earlier, is applied to our DER ecosystem use case. Also, the communication between the Flask RESTful Application Programming Interface (API) and the OpenLDAP server is achieved based on the ldap3 Python library (Figure 6) [36]. Specifically, it is a client library, strictly conforming to RFC4510 [37] and supports CRUD operations to the RBAC model that is hosted in an LDAP server. The interaction between the users and the RBAC model is straight-forward in the centralized approach, e.g., the RBAC administrator creates an HTTP request to revoke

a user's role using the front-end, this request is handled by the Flask API and served by the OpenLDAP server that hosts the RBAC model. Thus, we can automate the process of populating the OpenLDAP server with the RBAC model by utilizing the ldap3, and making the universal DER AC mechanism transparent to the users.

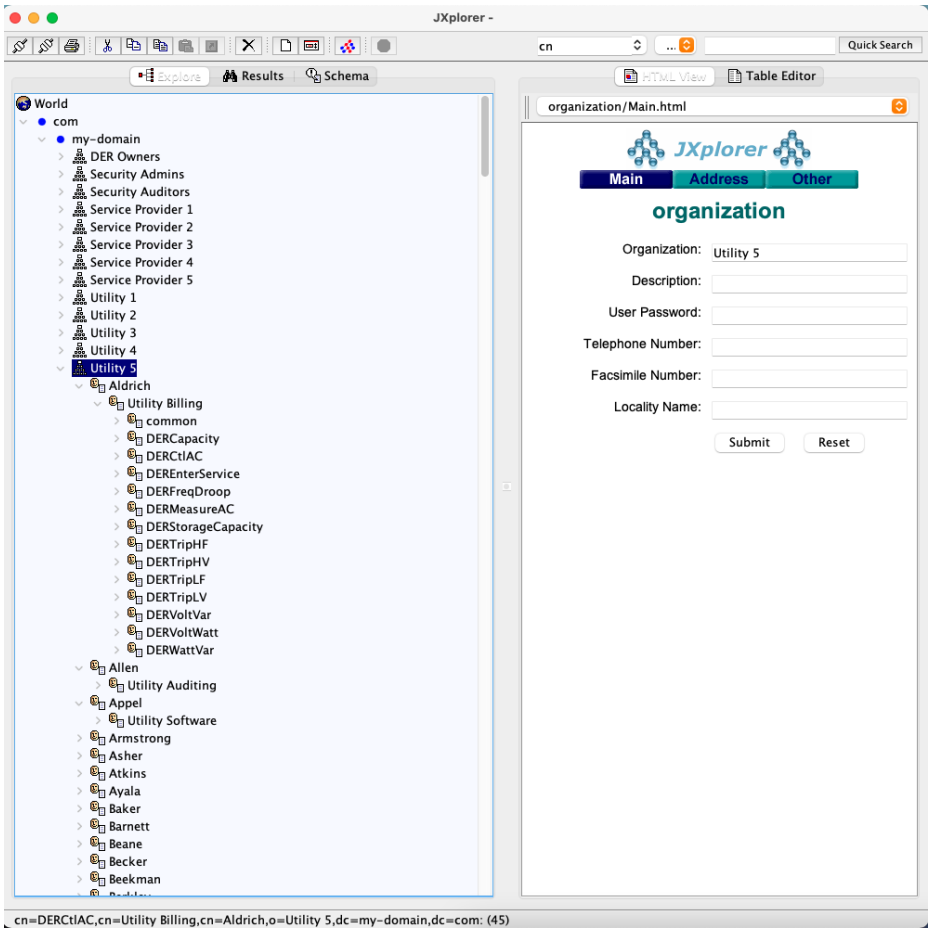

Figure 8. JXplorer & the DIT format.

4. Implementing RBAC in DER Using a Smart Contract

For the decentralized approach, a private Ethereum blockchain test network was deployed using the Truffle development framework [38]. Truffle provides a lightweight development environment and it is an asset pipeline for blockchains by utilizing the Ethereum Virtual Machine (EVM) [39]. It offers the essential functionality of built-in Smart Contracts compilation, linking, and binary deployment on the Ethereum blockchain. Also, we utilized Ganache CLI [40], which provides all the necessary Test Remote Procedure Calls that are required to simulate a full Ethereum client behavior, without the computing and storage overheads of running an actual Ethereum node. This tool enables us to generate all the Ethereum accounts needed for simulating a complete DER ecosystem with the DER devices, companies, and users.

The key idea of implementing an RBAC mechanism for the DER ecosystem is based on Smart Contracts. Ethereum utilizes Solidity [41], a JavaScript-like programming language, which supports features such as libraries, data structures, and variable or function definitions. RBAC system operators can compile the Smart Contracts into bytecode and deploy them automatically on the EVM. We created a single Smart Contract named *RBAC.sol*, which depicts the whole DER RBAC logic, by including functions that perform CRUD operations to the RBAC model. The main operations are: (a) adding or deleting users and DER devices by the administrator, (b) searching information about the DER entities, e.g., their Ethereum accounts or roles, (c) adding, deleting, and verifying user-to-role and

role-to-permission assignments by the administrator, (d) enabling users to ask for access to specific registers of DER devices based on their roles, (e) searching for transaction information that is stored in the blockchain based on the transactions' hashes, and (f) granting access to users by the administrator. The RBAC.sol is deployed to the aforementioned private Ethereum blockchain test network by the DER administrator, and acquires an Ethereum public address to be accessible by all users. The administrator adds all the DER entities, user-to-roles, and role-to-permissions assignments to the blockchain in the form of transactions, storing thus the whole RBAC model in the Ethereum blockchain. By creating just one Smart Contract, we can provide a universal decentralized AC mechanism to multiple DER organizations.

We consider a use case where a user with a certain role requests access/authorization to a DER device, i.e., a read/write operation to a DER Modbus register. The user logs into the web application and creates the authorization request (askAccess function), which is stored in the blockchain as a transaction with a unique transaction hash. Then, the decentralized system receives a notification regarding the access request by utilizing the Ethereum events, and it automatically pulls the transaction data from the blockchain by providing as an argument the respective transaction hash (pullRequest function). Thus, it acquires the Ethereum public address and role of the user, and the operation that the latter one wants to perform to the register of the DER device. The introduced framework automatically checks if it can grant access to the user, based on the existing RBAC model that resides in the blockchain. Specifically, it verifies the user with that Ethereum address has the requested role by querying the Smart Contract's function hasRole, and then verifies the requested role-to-permissions assignment (queryPerm function). An Ethereum transaction is created to store the user's access token in the blockchain using the grantAccess function. Finally, the user is able to fetch the access token by utilizing the pullToken function and present it to the DER device to get access. This shows how an AC Push Model [30] can also be deployed as a decentralized scheme using blockchain and Smart Contracts. The above process is shown in Figure 9.

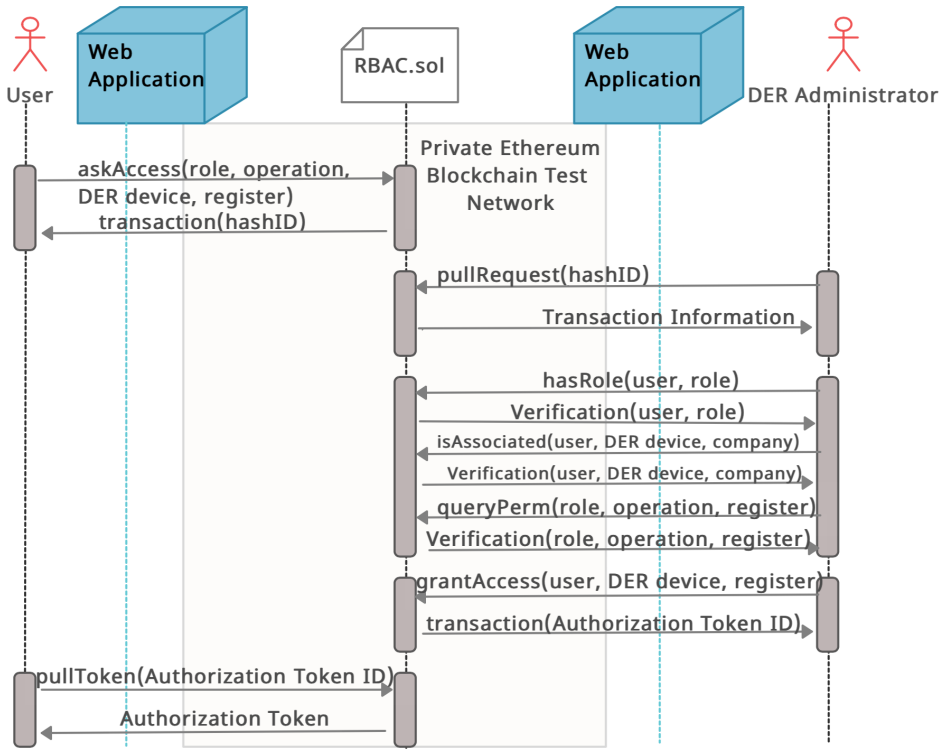

Figure 9. Sequence diagram of user Smart Contract interactions with the RBAC private Ethereum blockchain test network.

The full stack architecture of the decentralized PoC, shown in Figure 10, includes communications between the Flask RESTful API and the private Ethereum blockchain network with a web3 Python API. This approach is widely used in decentralized applications (dapps) to send transactions, deploy Smart Contracts, and other Ethereum blockchain-related functionalities.

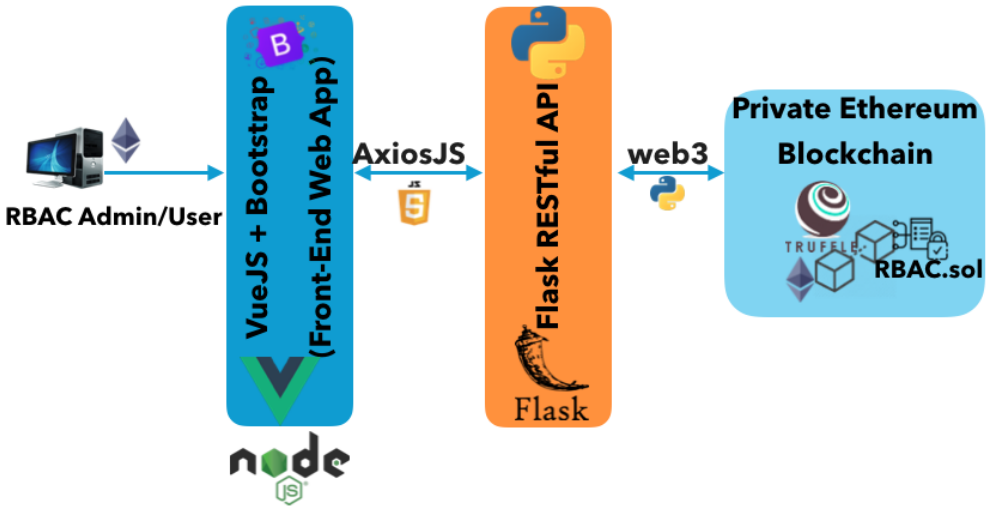

Figure 10. Decentralized RBAC Architecture.

5. Operational Demonstrations

Both the proposed RBAC mechanisms can be efficiently deployed in the DER ecosystem to restrict unauthorized access to DER devices. However, they are characterized by different advantages and disadvantages. In Table 2, we present a brief summary of the advantages and disadvantages of both the centralized and decentralized approaches regarding the *security, speed, cost, and usability*. These metrics are actual criteria that could be considered by the system administrators in order to select a single approach according to their business needs and goals. Ultimately, it will be an industry consensus or DER RBAC policy maker's decision to select the type of AC system to deploy. Conceivably, a national or regional DER interconnection standard such as IEEE 1547-2018 [42] or California Electric Rule 21 [43] could include requirements to use a particular AC system for DER communications. Alternatively, a more grassroots, the bottom-up approach could be taken with each of the stakeholders to select an implementation to deploy for the United States or other jurisdictions. In this section, we analyze a number of factors with respect to the aforementioned axis that should be considered in that decision.

For the Ethereum blockchain approach, the main advantage is there is no Single Point Of Failure (SPOF) since it is a peer-to-peer (P2P) network distributed across multiple Ethereum nodes and the centers of authority are distributed across the whole network. Blockchain is characterized as highly resistant to Distributed Denial of Service (DDoS) attacks [44], since the decentralized nature of blockchain allows the appropriate allocation of data and bandwidth. Additionally, transactions are fully transparent because every user acquires the same exact data from the blockchain, while there is not any third-party authority controlling the data centrally. However, public-private key cryptography is utilized in Ethereum, which masks the true identity of the users by assigning them their unique hexadecimal Ethereum address. Thus, anonymity is preserved. The Elliptic Curve Digital Signature Algorithm [45] is deployed, enabling the obfuscation of the user behaviors in regard to the recorded transactions. This characteristic is also known as pseudonymization. Blockchain guarantees immutability since the data written on it as transactions are not susceptible to malicious changes [46]. Ethereum blockchain is slightly faster than traditional blockchain networks, e.g., Bitcoin, while Ethereum 2.0 will enable sharding for load balancing to accelerate transaction processing [47]. The deployment of a DER AC ecosystem on the Ethereum blockchain is easy, since it can be implemented as a simple Smart Contract

without worrying about the underlying infrastructure. Thus, low deployment costs are experienced, as shown in Table 3.

Table 2. Summary of the main advantages and disadvantages of the proposed RBAC approaches with respect to the aspects of *Security, Speed, Cost, and Usability*.

RBAC Approach	Advantages	Disadvantages
Decentralized (Ethereum Blockchain)	Security • No Single Point Of Failure (SPOF) • Highly resistant to DDoS attacks • Fully transparent • Immutable, i.e., not susceptible to malicious change Speed • Faster than traditional blockchain networks Cost • Small deployment costs Usability • Ease of deployment	Security • Vulnerable to 51% cyberattacks and routing cyberattacks • Self-maintenance (users are responsible for their private keys) • Phishing attacks Speed • Slow transaction processing Cost • Energy inefficient PoW protocol for consensus • Scalability Usability • Smart Contracts implementation risks
Centralized (OpenLDAP)	Security • It has support for Simple Authentication, Security Layer, and Transport Layer Security • It can run over TCP/IP and SSL directly • Interoperable and secure storage scheme Speed • Generally, fast CRUD operations Cost • OpenLDAP is an open-source implementation of the LDAP. • Scalability Usability • Transferable with a flexible architecture • Wide support across the industry	Security • Single Point of Failure (SPOF) • Not fault-tolerant • Vulnerable to man-in-the-middle and eavesdropping attacks • Spoofing of directory Speed • Support of nested groups leads to slow query response times Cost • High infrastructure, management, and integration costs • High costs to rent infrastructure for a cloud-based solution Usability • Time-consuming configurations

Table 3. Smart Contract-Related Costs.

Functions	Gas	Total Cost (Gwei)	Ethers	US Dollars
Smart Contract Deployment	4,948,242	84,120,114	0.084120114	$213.62
Add User Query (Simple)	138,825	2,360,025	0.002360025	$6.05
Add User Query (DER Owner)	202,888	3,449,096	0.003449096	$8.83
Delete User Query	88,745	1,508,665	0.001508665	$3.86
Add DER Device Query	60,084	1,021,428	0.001021428	$2.62
Delete DER Device Query	132,754	2,256,818	0.002256818	$5.78
Revoke Role Query	21,123	359,091	0.000359091	$0.91
Update Entity Query	42,419	721,123	0.000721123	$1.85

On the other hand, blockchain is prone to 51% cyberattacks [48], meaning that a malicious attacker who controls more than 50% of the P2P network's mining hash rate,

can cause a network disruption by modifying for instance the ordering of transactions or perform double-spending. Routing attacks are another prominent security concern, since large blockchain networks, such as Ethereum, are based on the high volume of data transfer. Another drawback is that the users have to maintain their own private keys, since if they lose them, they cannot perform any transaction. The importance of keeping private keys safe is also demonstrated by the often occurrence of phishing cyberattacks, where malicious actors try to steal the blockchain credentials of the users. Finally, the Proof-of-Work (PoW) protocol is utilized as a consensus mechanism in the Ethereum blockchain, which enables the Ethereum nodes to agree on the state of all the recorded transactions. However, the PoW protocol is energy inefficient [49] regarding the verification of the transactions. Another issue is the scalability problems of the blockchain. An increased number of users leads to slow transaction processing since the average time for a transaction to be validated and published in the Ethereum blockchain is more than 3 min [50]. From the users' perspective, it is not scalable, as the peers should be synchronized with the current version of the blockchain, and thus the required storage capacity is increased. Examining the usability aspect, there are risks derived from the deployment of smart contracts. Specifically, their implementation should be careful, since some features can alter the smart contracts' modeled behavior and lead to security risks.

For the centralized OpenLDAP approach, it incorporates a common well-vetted information technology RBAC technology and the learning curve will be gentle for management practitioners. OpenLDAP supports Simple Authentication, Security Layer, and Transport Layer Security (TLS) protocols, which can be utilized to provide strong data integrity and confidentiality protection. Also, it inherently offers a big range of password-based authentication mechanisms, e.g., DIGEST-MD5. This makes it one of the most interoperable storage schemes. Moreover, the fact that it is open-sourced makes it low-cost, while it is also scalable since the AC model can be shared and replicated to multiple servers. By using the OpenLDAP data structure and libraries, the RBAC policy is transferable as an LDAP Data Interchange Format (LDIF) format across alternative LDAP platforms, e.g., Microsoft Azure. OpenLDAP can run over TCP/IP and SSL directly, supporting fast CRUD queries in the centralized Oracle Berkeley database and it is compliant with the IEC 62351-8 RBAC standard [15].

The most important disadvantage of OpenLDAP is that there is a SPOF, making it vulnerable to cyberattacks, e.g., DDoS or SQL injection attacks, and it is not fault-tolerant. The default setting for OpenLDAP data traffic is to be unencrypted, i.e., without SSL or TLS, which makes it prone to man-in-the-middle and eavesdropping attacks. Another common security issue in DIT-based AC mechanisms is the spoofing of the directory, where the user is tricked that the data is derived from the directory, but in fact, it was maliciously modified. The directory-based storage scheme that OpenLDAP provides leads to storing nested groups of data, which eventually may lead to slow query response time. Also, the physical infrastructure, management, and integration cost may be high for an on-premise AC mechanism. Even for cloud-based solutions, the aggregated costs for renting the cloud services would be very high. Finally, from our experience developing both PoCs, the OpenLDAP approach includes more time-consuming configurations than the decentralized one.

In the following analysis, we consider 2500 DER devices, 12 roles, and 3748 users for the experimentation environment, unless otherwise stated. Each role can contain up to 5 permissions, e.g., add DER device, delete user, and others, and the users randomly request specific role tokens from the authorization system every time instance to perform specific operations based on their respective roles' permissions. We also note that we utilized the aforementioned experimentation environment for both the centralized and decentralized approaches to achieve a fair comparison between them. The RBAC-based authorization process for the centralized and decentralized methods takes place as described in Section 3 and Section 4, respectively.

In Figure 11, we present the execution time (sec) of the main queries for both the centralized (OpenLDAP) and decentralized (Ethereum) approaches as an average of 100 runs for each query. In the latter, queries are handled by the RBAC.sol Smart Contract that resides in the deployed private Ethereum blockchain. We observe that the OpenLDAP mechanism is faster than the blockchain approach. The average query execution time of the centralized and the decentralized approach is 0.0784 and 1.081 s with average standard deviations equal to 0.000381 and 0.078, respectively. At this point, we would like to highlight that the decentralized approach is characterized by a higher standard deviation compared to the centralized approach, because of the stochasticity of the Ethereum mining process in terms of publishing a new block in the blockchain. This happens because most of the CRUD queries are transformed into transactions that need to be stored in the blockchain. For instance, when an "Update Entity" query is executed, all the new information of the DER entity has to be recorded in the distributed ledger. Thus, extra time is needed for the signatures verification of the new transactions and the mining of the blocks that include them, based on the PoW consensus mechanism. "Delete DER Device" and "Add User (DER Owner)" queries consume the most time because the DER user, all the DER models, registers, and DER permissions have to be deleted or added. This means a lot of recursive CRUD queries for the OpenLDAP mechanism and nested transactions for the blockchain-based RBAC approach.

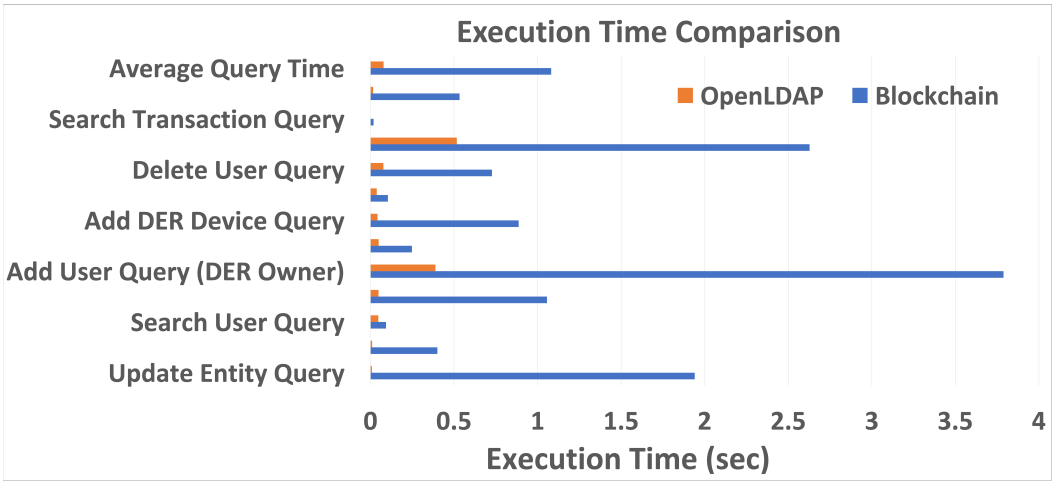

Figure 11. Centralized and Decentralized Query Execution Times.

In Table 3, the most important Smart Contract-related costs are presented, in regards to deployment and functions that would result in transactions in the public Ethereum blockchain. The costs were initially measured in Ethereum gas, which is a measure of the computational power needed to execute a certain function. We acquired the total cost using a representative Ethereum gas price of 17 Gwei. The total cost represents the actual fees that the miners will receive from the DER RBAC administrator validate and publish the transactions [51]. Since 1 eth $= 10^9$ Gwei, we derive the cost in Ethers and US Dollars. We observe that the RBAC Smart Contract deployment in the private Ethereum blockchain test network is the most expensive action ($213.62), because of the compilation process and converting the Solidity code into bytecode. However, the deployment happens only once, and then, the corresponding functions can be fetched/called using the public Ethereum address of the Smart Contract. Also, the "Delete DER Device" and "Add User (DER Owner)" queries result in the most expensive transactions, i.e., $5.78 and $8.83, respectively. This happens due to the nested transactions that these queries provoke, resulting in a higher amount of gas. Queries that are responsible for editing the RBAC model are less than $10. This is an acceptable cost to an organization for onboarding new staff, removing users, or commissioning a new DER. However, one risk with this cost is that organizations will not remove users when they no longer should have access to the DER equipment

due to the associated costs. Strict policies and auditing mechanisms would need to be established to prevent these security risks. The rest of the queries presented in Figure 11 are omitted because they do not result in transactions since they perform read operations on the blockchain data.

The $213.62 price tag for deploying the RBAC Smart Contract for the considered experiment environment, i.e., 2500 DER devices, 12 roles, and 3748 users can be extrapolated for generalization purposes to estimate the cost of deploying a national RBAC blockchain for the U.S. Even though this is an overestimate, let us assume 50% of legacy DER equipment is interoperable (1.25 M devices with 1.25 M owners with one role each), all 3300 American utilities were participating with three roles each, and there were 100 vendors with one role, 25 aggregators with one role, and 100 installers with one role. In that case, a national Smart Contract-based RBAC deployment would cost approximately $71,821.74. In contrast, the costs associated with creating and renting a single, cloud-based Linux LDAP server (i.e., storage optimized with 48 CPUs, 348 GBs temporary storage and an 8TB standard SSD for fast I/O operations) with a service level agreement with less than 1 hour of downtime per year, or 99.99%, would be $2.63 per hour or equally $3013.44 per month according to the Azure pay-as-you-go pricing model. There is an extra monthly charge of $614.40 for the 8 TB storage, concluding to a total of $3627.84 for the per month operational costs. As far as the respective RBAC deployment costs are concerned, a rough estimate for the centralized Azure-based LDAP approach is $7560.75, since each write operation (or transactional unit) costs $0.0020. After deployment, the cost for the centralized, cloud database would be constant every month, whereas the blockchain costs would be amortized based on the number and type of transactions. If the number and types of RBAC policy changes were known ahead of time, a more complete financial calculation could be completed. However, assuming the RBAC system will remain relatively constant, it is expected the operational costs will favor the distributed system after almost 17 months.

6. Conclusions

In this paper, we present end-to-end centralized and decentralized RBAC implementations for a DER ecosystem, which combine multiple software technologies. In the centralized approach, OpenLDAP was utilized to store the RBAC model in a DIT format on a standalone OpenLDAP server. In the decentralized RBAC mechanism, a private Ethereum blockchain test network with RBAC logic stored in a Smart Contract was deployed using the Truffle suite. In both approaches, we implemented a web application, based on the VueJS and Flask frameworks, that offers the aforementioned mechanisms as a Service. We compared the two approaches in terms of cost, speed, usability, and security to identify their pros and cons. At this point we should underline that the same design and implementation choices would be followed in the case of an alternative access control model, e.g., ABAC. Specifically, both the centralized and decentralized systems would remain the same since the only variable that changes in this scenario is that instead of the roles and permissions we would have the respective attributes. Consequently, the proposed APIs as well as the utilized storage spaces (e.g., Oracle database, Ethereum blockchain) would serve the needs of this scenario in the same way showing also a similar behavior. Thus, a system administrator should still determine if a centralized or a decentralized approach should be adopted for an alternative access control model based on the same criteria presented in Table 2 and the corresponding needs of the organization. Future work should investigate the deployment of alternative blockchain platforms, e.g., Hyperledger Fabric, Multichain, as well as focus more on the security performance analysis of both approaches.

Author Contributions: Conceptualization and writing, G.F. and E.E.T.; methodology, supervision, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the U.S. Department of Energy Solar Energy Technologies Office. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly-owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Chakraborty, S.; Das, S.; Sidhu, T.; Siva, A. Smart meters for enhancing protection and monitoring functions in emerging distribution systems. *Int. J. Elect. Power Energy Syst.* **2021**, *127*, 106626. [CrossRef]
2. Stellios, I.; Kotzanikolaou, P.; Psarakis, M.; Alcaraz, C.; Lopez, J. A Survey of IoT-Enabled Cyberattacks: Assessing Attack Paths to Critical Infrastructures and Services. *IEEE Commun. Surv. Tutorials* **2018**, *20*, 3453–3495. [CrossRef]
3. Huh, S.; Cho, S.; Kim, S. Managing IoT devices using blockchain platform. In Proceedings of the 19th International Conference on Advanced Communication Technology (ICACT), Pyeong Chang, Korea, 19–22 February 2017; pp. 464–467. [CrossRef]
4. Wachsmann, C.; Sadeghi, A.R. Physically Unclonable Functions (PUFs): Applications, Models, and Future Directions. *Synth. Lect. Inf. Secur. Priv. Trust* **2014**, *9*, 1–91.
5. Roman, R.; Alcaraz, C.; Lopez, J.; Sklavos, N. Key management systems for sensor networks in the context of the Internet of Things. *Comput. Electr. Eng.* **2011**, *37*, 147–159. [CrossRef]
6. Nandy, T.; Idris, M.Y.I.B.; Md Noor, R.; Mat Kiah, L.; Lun, L.S.; Annuar Juma'at, N.B.; Ahmedy, I.; Abdul Ghani, N.; Bhattacharyya, S. Review on Security of Internet of Things Authentication Mechanism. *IEEE Access* **2019**, *7*, 151054–151089. [CrossRef]
7. Patel, C.; Doshi, N., Security Challenges in IoT Cyber World. In *Security in Smart Cities: Models, Applications, and Challenges*; Hassanien, A.E., Elhoseny, M., Ahmed, S.H., Singh, A.K., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 171–191.
8. Jiayi, H.; Chuanwen, J.; Rong, X. A review on distributed energy resources and MicroGrid. *Renew. Sustain. Energy Rev.* **2008**, *12*, 2472–2483. [CrossRef]
9. Kumar, N.M.; Chand, A.A.; Malvoni, M.; Prasad, K.A.; Mamun, K.A.; Islam, F.; Chopra, S.S. Distributed energy resources and the application of AI, IoT, and blockchain in smart grids. *Energies* **2020**, *13*, 5739. [CrossRef]
10. Osborn, S. Mandatory access control and role-based access control revisited. In Proceedings of the ACM Workshop on RBAC, Fairfax, VA, USA, 6–7 November 1997; pp. 31–40.
11. Moffett, J.; Sloman, M.; Twidle, K. Specifying discretionary access control policy for distributed systems. *Comput. Commun.* **1990**, *13*, 571–580. [CrossRef]
12. Sandhu, R.S. Role-based access control. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 1998; Volume 46, pp. 237–286.
13. Hu, V.C.; Kuhn, D.R.; Ferraiolo, D.F.; Voas, J. Attribute-based access control. *Computer* **2015**, *48*, 85–88. [CrossRef]
14. Coyne, E.; Weil, T.R. ABAC and RBAC: Scalable, flexible, and auditable access management. *IT Prof.* **2013**, *15*, 14–16. [CrossRef]
15. IEC Webstore, IEC 62351-8:2020. 2020. Available online: https://webstore.iec.ch/publication/61822 (accessed on 15 May 2022).
16. Chaudhry, S.A.; Alhakami, H.; Baz, A.; Al-Turjman, F. Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure. *IEEE Access* **2020**, *8*, 101235–101243. [CrossRef]
17. Suciu, G.; Istrate, C.I.; Vulpe, A.; Sachian, M.A.; Vochin, M.; Farao, A.; Xenakis, C. Attribute-based access control for secure and resilient smart grids. In Proceedings of the 6th International Symposium for ICS & SCADA Cyber Security Research, Athens, Greece, 10–12 September 2019; pp. 67–73.
18. Suciu, G.; Istrate, C.; Sachian, M.A.; Vulpe, A.; Vochin, M.; Farao, A.; Xenakis, C. FI-WARE authorization in a Smart Grid scenario. In Proceedings of the 2020 Global Internet of Things Summit (GIoTS), Dublin, Ireland, 3 June 2020; pp. 1–5.
19. Barka, E.; Hussien, N.A.; Shuaib, K. Securing Smart Meters Data for AMI Using RBAC. In Proceedings of the 2016 11th Asia Joint Conference on Information Security (AsiaJCIS), Fukuoka, Japan, 4–5 August 2016; pp. 1–8. [CrossRef]
20. Nagarajan, A.; Jensen, C.D. A Generic Role Based Access Control Model for Wind Power Systems. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **2010**, *1*, 35–49.
21. Bera, B.; Saha, S.; Das, A.K.; Vasilakos, A.V. Designing blockchain-based access control protocol in IoT-enabled smart-grid system. *IEEE Internet Things J.* **2021**, *8*, 5744–5761. [CrossRef]

22. Guan, Z.; Lu, X.; Yang, W.; Wu, L.; Wang, N.; Zhang, Z. Achieving efficient and Privacy-preserving energy trading based on blockchain and ABE in smart grid. *J. Parallel Distrib. Comput.* **2021**, *147*, 34–45. [CrossRef]
23. Yang, W.; Guan, Z.; Wu, L.; Du, X.; Guizani, M. Secure Data Access Control With Fair Accountability in Smart Grid Data Sharing: An Edge Blockchain Approach. *IEEE Internet Things J.* **2021**, *8*, 8632–8643. [CrossRef]
24. Agyekum, K.O.B.O.; Xia, Q.; Sifah, E.B.; Cobblah, C.N.A.; Xia, H.; Gao, J. A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of Things Based on Blockchain. *IEEE Syst. J.* **2021**, *16*, 1685–1696. [CrossRef]
25. Zhou, Y.; Guan, Y.; Zhang, Z.; Li, F. A blockchain-based access control scheme for smart grids. In Proceedings of the Internaltional Conference on Networking and Network Applications, Daegu, Korea, 10–13 October 2019, pp. 368–373.
26. Alcaraz, C.; Rubio, J.E.; Lopez, J. Blockchain-assisted access for federated Smart Grid domains: Coupling and features. *J. Parallel Distrib. Comput.* **2020**, *144*, 124–135. [CrossRef]
27. Gai, K.; Wu, Y.; Zhu, L.; Xu, L.; Zhang, Y. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. *IEEE Int. Things J.* **2019**, *6*, 7992–8004. [CrossRef]
28. Zhang, H.; Wang, J.; Ding, Y. Blockchain-based decentralized and secure keyless signature scheme for smart grid. *Energy* **2019**, *180*, 955–967. [CrossRef]
29. Saleem, D.; Johnson, J. *Distributed Energy Resource (DER) Cybersecurity Standards*; Technical Report; National Renewable Energy Laboratory: Golden, CO, USA, 2017.
30. Johnson, J.T. *Recommendations for Distributed Energy Resource Access Control*; Technical Report; Sandia National Lab.: Albuquerque, NM, USA, 2021. [CrossRef]
31. Howes, T.A.; Howes, T.; Smith, M.; Good, G.S. *Understanding and Deploying LDAP Directory Services*; Addison-Wesley Prof.: Reading, MA, USA, 2003. Available online: https://tinyurl.com/3ztjm4ps (accessed on 14 May 2022).
32. Hong, P. *Practical Web Design: Learn the Fundamentals of Web Design with HTML5, CSS3, Bootstrap, jQuery, and vue. js*; Packt Publ.: Birmingham, UK, 2018. Available online: https://tinyurl.com/mr3b5wzh (accessed on 14 May 2022).
33. AxiosJS. 2022. Available online: https://axios-http.com/docs/intro (accessed on 14 May 2022).
34. Fragkos, G.; Johnson, J. Centralized LDAP Codebase. GitHub, 2021. Available online: https://github.com/geofragkos/RBAC_Centralized (accessed on 25 May 2022).
35. Fragkos, G.; Johnson, J. Decentralized LDAP Codebase, GitHub, 2021. Available online: https://github.com/geofragkos/RBAC_Decentralized (accessed on 25 May 2022).
36. ldap3 Python Library. 2022. Available online: https://ldap3.readthedocs.io/en/latest/welcome.html (accessed on 13 May 2022).
37. RFC4510. 2006. Available online: https://www.ietf.org/rfc/rfc4510.txto (accessed on 13 May 2022).
38. Mohanty, D. Frameworks: Truffle and Embark. In *Ethereum for Architects and Developers*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 181–195.
39. Hildenbrandt, E.; Saxena, M.; Rodrigues, N.; Zhu, X.; Daian, P.; Guth, D.; Moore, B.; Park, D.; Zhang, Y.; Stefanescu, A.; et al. Kevm: A complete formal semantics of the ethereum virtual machine. In Proceedings of the 31st Computer Security Foundations Symposium, Oxford, UK, 9–12 July 2018; pp. 204–217.
40. Lee, W.M. Testing smart contracts using ganache. In *Beginning Ethereum Smart Contracts Progr.*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 147–167.
41. Dannen, C. *Introducing Ethereum and Solidity*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 318.
42. *IEEE Std 1547-2018 (Rev. of IEEE Std 1547-2003)*; IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces; IEEE: Piscataway, NJ, USA, 2018; pp. 1–138. [CrossRef]
43. California Public Utilities Commission. *Electric Rule No. 21 Generating Facility Interconnections*; California Public Utilities Commission: San Francisco, CA, USA, 2018.
44. Wani, S.; Imthiyas, M.; Almohamedh, H.; Alhamed, K.M.; Almotairi, S.; Gulzar, Y. Distributed Denial of Service (DDoS) Mitigation Using Blockchain—A Comprehensive Insight. *Symmetry* **2021**, *13*, 227. [CrossRef]
45. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
46. Saraf, C.; Sabadra, S. Blockchain platforms: A compendium. In Proceedings of the International Conference on Innovative Research and Development (ICIRD), Bangkok, Thailand, 11–12 May 2018; pp. 1–6.
47. Park, D.; Zhang, Y.; Rosu, G. End-to-end formal verification of ethereum 2.0 deposit smart contract. In *Proceedings of the International Conference on Computer Aided Verification*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 151–164.
48. Saad, M.; Spaulding, J.; Njilla, L.; Kamhoua, C.; Shetty, S.; Nyang, D.; Mohaisen, A. Exploring the attack surface of blockchain: A systematic overview. *arXiv* **2019**, arXiv:1904.03487.
49. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 3–16.
50. Weber, I.; Gramoli, V.; Ponomarev, A.; Staples, M.; Holz, R.; Tran, A.B.; Rimba, P. On availability for blockchain-based systems. In Proceedings of the 36th Symposium on Reliable Distributed Systems (SRDS), Hong Kong, China, 26–29 September 2017; pp. 64–73.
51. Pierro, G.A.; Rocha, H. The influence factors on ethereum transaction fees. In Proceedings of the IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain, Montreal, QC, Canada, 27 May 2019; pp. 24–31.