Towards a Zero Trust Based Hybrid Access Control Model for Medical Data

Colin Wilcox*, Kristopher Welsh*, Soufiene Djahel†, Nicholas Costen* and Vasileios Giagos‡
* Department of Computing and Mathematics, Manchester Metropolitan University, UK
† Centre for Future Transport and Cities, Coventry University, UK
‡ School of Mathematics, Statistics and Actuarial Science, Essex University, UK
{colin.r.wilcox@stu.mmu.ac.uk; (k.welsh, n.costen)@mmu.ac.uk; soufiene.djahel@coventry.ac.uk; v.giagos@essex.ac.uk}

***Abstract*—This paper addresses some of the limitations of current hybrid access control models, in particular the issues surrounding managed rights inheritance, a model's consistency and its ability to reflect changes in environment without losing integrity or flexibility. Our approach describes a layered model design which allows more or less detail to be added, in real-time, to reflect changes in the real world. The model provides time-based inherited access control following the principles of zero trust to minimise risk whilst preserving its integrity and flexibility. This paper concludes by providing a comparison of how our approach differs from traditional techniques and the areas in which improvements can be seen.**

***Keywords* – access control, modelling, least privilege, RBAC, ABAC, MAC, DAC, zero trust.**

I. Introduction

The ever-tighter integration of technology in everyday life increases the scope of individuals' digital footprint. The scope, sensitivity & volume of data stored about individuals raises increasing concerns over privacy and security. Data stored about individuals needs to be accurate, consistent and secure [1]. Data security concerns have led to a wider use of access control mechanisms to add protection to computer systems and the people that use them, especially when the data is highly sensitive such as in a medical context. Personal medical data is used as the basis for making medical decisions for individuals and as such the data is expected to be accurate, complete and secure from unauthorised access or potential manipulation [2]. The risks associated with unauthorized access to personal data can affect any domain where potentially valuable information is stored and so there is a need to ensure that such data is kept secure. Advances in technology and the corresponding rise in sophistication of cyber attacks has seen an increase in attempts to gain illegitimate system access [3], driving the need to develop more secure systems. Access control systems have evolved over the decades to consider the specific needs of businesses as well as to address and limit certain types of attacks. Such an evolution creates the potential for new vulnerabilities to be introduced [4]. A key aim of any access control system is to protect the data in the system and ensure its accuracy [5]. Many systems were developed before the need to consider access control and the problems it may bring. Historically, many medical facilities had no notable or consistent access control methods beyond physically locking records away, leaving them prone to tampering or manipulation [6]. The smart healthcare era has introduced new access control methods, however these tend to have been developed with minimal standardisation [7]. Newly developed access control approaches need to be flexible enough to accommodate new technologies with new vulnerabilities. This will only emphasise the importance of access control.

This paper proposes a flexible, modular, hybrid access control solution which improves on existing models by allowing dynamic changes to the model's structure and operation in real-time to reflect changes context. The solution also provides a time-based rights mechanism to allow rights to be inherited for controlled periods to allow better workflow and resource management. A pipeline-centric approach has been adopted with each stage of the pipeline being able to exist independently of the others, allowing each stage to be replaced with a more effective solution as necessary, without impacting the rest of the pipeline.

The remainder of this paper is organized as follows. In Section II, we present a brief background and history on past approaches to access control and the issues they presented together with related work. Section III compares and highlights the deficiencies of existing access control models in which our new hybrid model could improve upon. Section IV gives an overview of how the evolution of access control models has given rise to the zero trust philosophy. Section V provides an overview of our proposed approach outlining the key design components and architecture together with a brief analysis in Section VI. Section VII concludes the paper.

II. Background and related work

The term *access control* describes the process and ability for a system to restrict access to locations, systems, and information. The emergence of cloud based solutions has blurred the boundaries of what constitutes a traditional system, creating new challenges with regards to who can make use of what, and who should be allowed to access what. To prevent unauthorised access it is required to monitor and track all access requests and apply sophisticated rules to ensure those making access requests are correctly identified and authorised to access the resource they require. Depending on the nature and structure of an organization several broad concepts will need to be considered namely;

- What level of ownership do you want over the system?
- How do you decide on the criteria that allows access?

Access control is one of the key security factors providing both data confidentiality, by restricting access to sensitive information only to those who are authorised [8] and integrity by defining and controlling who can access and modify data, preventing unauthorized exposure or alterations to be made to sensitive information. The symbiotic processes of *authentication* and *authorization*, together with the application of suitable access control policies ensure that users are firstly correctly identified and then confirms that they have sufficient and appropriate rights to access company data.

There are several widely used types of access control model within organisations. The choice of model is dependant on its suitability for the business, based on their unique security and compliance requirements. The most common access control models are outlined below.

A. *Mandatory access control* ***(MAC)***

MAC-based models are the most restrictive and the most secure of the four main models discussed. Here, the power to permit access lies entirely with the administrators of the system, making it a highly centralised approach. A MAC approach sets mandatory rules which enforce all access rights requests in order to maintain compliance. There is no user control to manipulate these rules as they are owned and fixed by the business or organisation. This security means that not only does a user require access to a system but also needs explicit access to secure resources, making it the hardest access control method to manage and often burdensome for users. Due to these tight controls there are many widely-used models that are based on mandatory access control to restrict the flow of data from a higher to lower level of access including well known models such as the Bell-LaPadula Model [9] and Clark Wilson Model [10]. MAC systems occur within military security, where an individual data owner does not decide who has a top-secret clearance, nor can they change the classification of an object from top-secret to secret [11].

B. *Discretionary access control* ***(DAC)***

DAC-based models are less restrictive than MAC based models by restricting access to resources based on the identity of subjects and groups to which they belong. Every resource has an owner who decides who gets access to that resource. There are three fundamental elements of access control policy; a *subject* as an active entity that accesses system data, an *object* which is a passive entity that represents the data to be protected, and an *action* which represents the action that subject performs on the object. This model is the least restrictive allowing access control to be decentralized and managed by teams and individuals giving more control to the people on the ground doing the work by allowing them to determine who can access which resources. Access to any given resource is a matter of having the right credentials at the right time compared against an access control list. These controls are discretionary in the sense that a user who has been given such access to information is capable of passing that information along to another subject. While DAC can provide a high level of flexibility for organizations, it also brings a lack of visibility for identity administrators. To provide this discretionary control, DAC policies usually include a concept of resource ownership, where the resource owner has control permission to grant access to the resource for other parties [12]. This approach has the obvious disadvantage of giving the end-user control over security levels and may provide gaps in the wider business security schema if not closely monitored. A technological example of a DAC system are the file and folder permissions common in many modern operating systems, including Windows and Unix, which define the operations that may be performed on these entities. In such a system users may transfer object ownership to another user as required.

C. *Role-based access control* ***(RBAC)***

RBAC-based models aim to facilitate access control rights based on roles through a model that simplifies resource and access administration and offering higher performance and easier scalability [13]. RBACs are therefore amongst the most widely used type of access control mechanism and determine access based on your role within the company. Such an approach ensures that lower-level employees within the same part of the business are not granted access to high-level information compared to their more senior department colleagues. Employee access can be managed by grouping similar employees together and granting *group level* access profile to all members of the group. The *RBAC* approach provides a flexible model that increases visibility while maintaining protection against breaches and data leaks. The RBAC approach suffers from several drawbacks including potential *role explosion* caused when the level of granularity needed for access control requires so many roles as to make access control too complex. This reduces the effectiveness of the access control system. RBAC solutions suffer from issues of scalability and dynamism with changes in role descriptions and company structure adding to the maintenance overhead and cost by requiring regular systems reviews to streamline access control and to plug any security holes that may have been inadvertently created. These issues, together with the evolution of technology has allowed established practices in access control to move towards more customised approaches. *Financial institutions* are a good example of RBAC systems where secure access to sensitive data is crucial. Such systems would include roles such as account managers, bank tellers, branch managers, and auditors each of which can perform their duties without compromising security or violating privacy regulations.

D. *Attribute based access control* ***(ABAC)***

ABAC-based models provide a dynamic and risk-intelligent control based on attributes given to a specific user. These attributes may be considered as part of a user-centric profile and together they define a user's access level and permissions.

Once policies are set, these attributes can be used to determine whether or not a user should have access. ABAC can be seen to extend the RBAC based approach by being able to delegate attribute authority and the decentralisation of attributes [13]. Access to a particular resource is determined based on certain attributes associated with the entity making the access request at the time and place from where the request is being made. Attributes of the resource may be business, project related, or based on personally identifying information. These attributes, when considered together, are used to determine the rules governing access to data and resources. A common known example of an ABAC system is that used by streaming video providers who use sharing attributes to authorize external users and enforce their household sharing policies.

The role based approach of RBAC systems is too granular, inflexible, and scales poorly as the size of the business and number of roles increases. ABAC systems make an access determination based on more environmental, time-based and situational information at the moment of request.

III. Comparative study

The evolution of access control models has been driven by business need rather than being directed by technology. Each model iteration has built upon existing ideas and concepts rather than looking for a general-purpose, technology-focused framework. Each generation of model has solved specific issues highlighted by their predecessors, for instance RBAC solves the problem of managing the security administration complexity of large networks by introducing roles, solving the inflexibility problems of previous MAC and DAC based approaches, but itself introduces the problem of managing role explosion resulting in a largely manual and time consuming process. The state of the art in access control models has often been outpaced by the emergence of new access control threats, leading to continual firefighting to model development. Modern approaches now tend to use a hybrid design [14], leveraging many of the insights gained from past iterations without necessarily aiming for a multi-purpose, generic framework. Hybrid ABAC models aim to address these issues by finding a balance between a model's generality and its flexibility, or by removing the identity-less nature of ABAC. This leaves a number of unsolved problems with regards to ABAC-based solutions with a lot of focus on the inheritance of access rights [15], increased flexibility of models [16] and separation of duties needed to complete complex tasks [17] .

A. Role explosion as jobs change

The main drawback of RBAC based approaches is that of role explosion and the security weaknesses that it can introduce to a system. Our proposed model does not associate any individual with a specific role but instead has a series of time controlled attributes which can be dynamically assigned or removed depending on the scenario, location or other external factors. This provides the necessary flexibility for medical workers to carry out their daily activities involving the short term adoption of responsibilities from other employees.

B. No emulation of traditional models

More research is needed to show how ABAC can be developed as an extension of previous access control models as well as its suitability to be used as a basis for developments in the future. Our proposal is to develop a more domain-agnostic solution without extending the issues with historic access control methods.

C. Hierarchical ABAC

The inherent structure of an organisation influences the relationships between the defined roles in a business allowing for easier administration. This is a fundamental feature that is missing from traditional ABAC systems. Although a given role can be represented as a single attribute of a subject, traditional ABAC models are unable to represent the hierarchical nature of RBAC models without introducing burdening complexity. The concept of attribute groups may provide a solution to this limitation with a group by allowing inheritance of attribute values by subgroups implicitly without the need for complex relationships being explicitly defined. This idea is extended within our model by allowing dynamic grouping using zones which may change over time. The general nature of our proposed solution removes the need to rigidly model an organisation by allowing relationships to change dynamically, or be added and/or removed as needed.

D. Separation of duties and access delegation

Separation of duties arises from the idea that more than one entity is needed to complete a task in a system, as a mechanism to reduce error, fraud and ideally increase security. RBAC systems implement this by preventing conflicting rules from being applied in the same session. This area seems to be largely unexplored in an ABAC context. One high profile exception to this is the work of Alipour and Sabbari [18] who introduced the idea of *do not perform* rules defining actions that are not applicable to specific resources but this is in itself problematic, requiring knowledge of resources and potential conflicting scenarios *a priori*. In a medical context, an often overlooked aspect of an individual's role is the ability to multitask by sharing responsibility to complete a task. Rigid inheritance rules and structures of existing approaches conflict with the need for *dynamic* delegation of responsibility and authority seen in real world situations. Our model moves the inheritance decisions away from the organisation and onto the medical staff themselves as they are best placed to decide what control needs to be given at any particular moment, which staff should inherit these abilities, and for how long.

IV. The evolution towards zero trust

The increasing difficulty of maintaining security and integrity for ever-more complex systems has led to the adoption of the Zero Trust (ZT) concept, whereby access to resources is denied unless identity is authenticated, regardless of access origin or location. This contrasts with traditional approaches where firewalls differentiate between secure (trusted) and potentially insecure entities. Without ZT, attackers once within

	MAC	RBAC	DAC	ABAC	Our Model
Access Rights Owner	Administrator	Business	Creator	Attribute Controlled	Attribute Controlled
User's Convenience	Variable	High	High	High	High
Node Overhead	Low	Low	Low	Varies	Low
Performance	Variable	High	Variable	High	High
Reusability	N/A	Multiple Users	Multiple Users	Multiple Users	Multiple Users
Role Assignment	Controlled by policies	Controlled by policies	No, User/Group based	No, access determined by user characteristics	User, group or role based
Information flow	Global	Restricted	Restricted	Flexible	Flexible
Maintenance Level	Medium	High	Low	Medium	Low
Secure	Yes	Sometimes	Yes	Yes	Yes
Flexible	No, rigid and less flexible	Balanced between flexible and secure.	Yes, but can lead to insecurity	Yes	Yes
Scaleable	Low	Medium	Medium	High	High
Reflects Real World	No	Sometimes	Sometimes	Yes	Yes
Generalised Use?	No	No	Partially	Partially	Yes
Supports Dynamic Changes	Limited	Limited	No	Yes	Yes
Include Incomplete Data Sets	No	No	No	No	Yes
Supports Inheritance	No	Limited	Limited	Yes, rule based	Yes, both rule and location based

Table I: Metrics for comparing access control model approaches

the firewall can gain access to systems or data that would not be granted outside, due to the implicit trust placed in their access origin. With ZT, trust is established between entities at the moment of each request for access.

Increasing the level of security around sensitive computer systems acts as a deterrent to would-be attackers, who turn their attentions to less well protected, more vulnerable, targets. The overall process is an attempt to raise the *software trust* within an organisation by reducing the number of system attacks, and thus breaches, while raising the quality of the software on which the business depends.

A. What is zero trust?

It is realistically impossible to determine all potential threats to a system in advance. The practical approach is to start with the worst case scenario and assume everything is a potential threat. Zero trust is based on this simple concept, with three axiomatic premises to maintaining system security namely *assume nothing*, *believe no one* and *check everything*. This means that existing security systems used to protect critical information and resources should not be implicitly trusted, but that any request for access should be assumed potentially malicious. The decision whether access is granted is solely based on the information available at the time of the request and uses nothing about any previous access attempts or historical information. The National Institute of Standards and Technology (NIST) has formally defined ZT as *"an evolving set of cybersecurity paradigms that move defenses from static and network-based perimeters to focus on users, assets and resources"* [19]. Despite acknowledging its obvious security benefits, many organizations are reluctant to implement a fully ZT based approach. Typically, concerns for legacy environments which have too much technical debt to overcome often make such a transition untenable [20]. A complete ZT based approach is, therefore, considered more suitable for new environments and greenfield projects which are not hampered by historic technical debt. A more incremental transition is usually applied to such legacy environments [21].

V. Proposed hybrid access control model

This section describes a high level overview of the design of our proposed hybrid access control model. The key features will be highlighted to show how our model would be an improvement over other hybrid and traditional approaches and how it addresses some of the specific limitations of these solutions. Our platform architecture design includes:

- A domain-agnostic solution.
- A pluggable and modular development framework.
- Templated functions and service solutions aimed at the medical domain.
- Support a high degree of configuration.

The platform architecture will use a service oriented architecture approach, specifically a microservice design, allowing it to be extended in a programming language independent manner. The ethos of the design will to be environmentally agnostic not aligning the design to any particular business structure or problem domain using an incremental ZT approach. Any specifics needed to accurately model a specific environment will be defined as part of the model itself rather than the framework. The completed data set will be seen as the source of truth on which decisions made within the access controlled environment will be based and so it is essential that it not only maintains its integrity but also remains secure, accurate and as complete as possible. Our design uses a two phased pipeline approach with each phase being reliant on the other. The modified data set from the first phase will be used as the source of truth in the second phase, namely the access control model.

Our proposed design offers a two stage solution. The first stage is concerned with taking an incomplete medical data set and applying a set of transformation algorithms in an attempt to *complete* the data, with a goal of creating a pool of information that is sufficiently accurate and consistent that it can be used as a source of truth in an access control model. The second stage in our solution is a simulated access control model which although generic in nature, will be specifically tailored to a medical scenario. The architecture define a simulation containing an access control model that will be responsible for defining the core system elements and how they interact with one another, rules of behaviour and navigation.

The model's design uses a layered approach (Figure 1). In order to keep the design as flexible and extendable as possible there are two types of layer; those which are mandatory and those which are optional. There are really only two mandatory layers namely the physical layer (which defines the physical environment upon which the model will be based) and the top level resource layer (which defines both the physical and logical objects that are allowed to exist within the model). Both of these exist for the entire lifetime of the simulation although both can be changed to reflect environmental updates as needed. There can be as many optional layers as required by the scenario being modelled. With no optional layers, actors can move around a controlled space without restriction. Optional layers are used to define features of the model such as access control points, resource locations, zonal definitions or whatever else is needed to accurately reflect reality. These optional layers can be disabled, updated or even removed completely as part of a periodic update so long as no inconsistencies are created as a result of their removal.

A. Modelling the physical environment

The role of the *physical layer* in our hybrid model is to represent a real world environment digitally within a computer model. This is done by representing a physical space in the form of a unidirectional graph. Each node is considered as a location where actors within this space may need particular access permissions to proceed. Locations may be both physical and virtual in their nature. Nodes are connected by unidirectional transition links (arcs) each of which encompasses a concept of distance and weight to represent the physical separation of nodes within the real world. Arcs are unidirectional to provide the flexibility to allow the direction of movement between nodes to have different attributes to distinguish between flats and sloping surfaces. Actors, human or otherwise, can move within the confinement of the space by transitioning between nodes along arcs with any access permissions being satisfied to allow progression.

B. Modelling the logical environment

The *access layer* is superimposed, together with any optional layers, onto the physical layer to define any possible restrictions to movement (doors, access systems, access rights, time-factors). The ability to travel between any two nodes is determined by an individual's access rights defined in their profile and should include rules specific to the node *at the time of request*. The topological mapping of physical space is maintained by the distance attributes of arcs between nodes and in those situations where any given node is disabled or removed for some reason, distance is preserved.

The optional *zonal layer* defines how the physical space is segmented into self-managing regions. Zones will vary depending on their use and nature but will adhere to a number of basic rules. Various rules are imposed to ensure that the consistency and integrity of the model's topology and integrity and are maintained at all times. Zones are a powerful feature that may be used when managing co-located inheritance between resources. Since any model is just a representation of a real world scenario, the better the model the more accurately it mimics the real world, those interactions and situations that may happen within its confines. In order to produce a model with sufficient flexibility and scalability it is necessary to allow as many of the model's parameters to be configured allowing it to be used in as wide a range of scenarios as possible.

C. Configuration files

Configuration files are used to configure both the model's initial configuration and startup parameters as well as to accommodate dynamic changes to the operational parameters of a running model without the need to stop or restart it. These dynamic changes may include such tasks as the addition of resources and other assets, changes to the structure of a physical environment or the reflection and imposition of changes to external rules and policies on a running model environment. The frequency of these updates is model specific and defined as part of the active refresh cycle for a given model allowing different model instance to be updated at different rates independently of each other. The use of these file is kept deliberately open ended to allow their use to be extended in the future as needed.

D. Periodic updates and refreshing the model

On startup the model will look for a suitable configuration file, exiting gracefully if non can be found. If one is found it will be read, parsed and the details needed to construct the model will be used to create the initial model's state. The creation timestamp will be stored for later comparison and a periodic callback timer will be initiated to start the period update process. A valid configuration file contains information specifying how often model updates are to be checked. Each running model instance may be updated independently, with each instance having its own configuration file defining all active parameters necessary for that instance at any given time. During the startup process the model engine watches for changes to a file in a specific system folder. When the periodic update timer expires the model's periodic callback functionality checks the watched file's timestamp to see whether it has been touched since the last update. This provides the flexibility and adaptability to reflect changes in the real world in close to real time. When there are no changes the model remains unchanged but in either case the periodic update timer restarts.

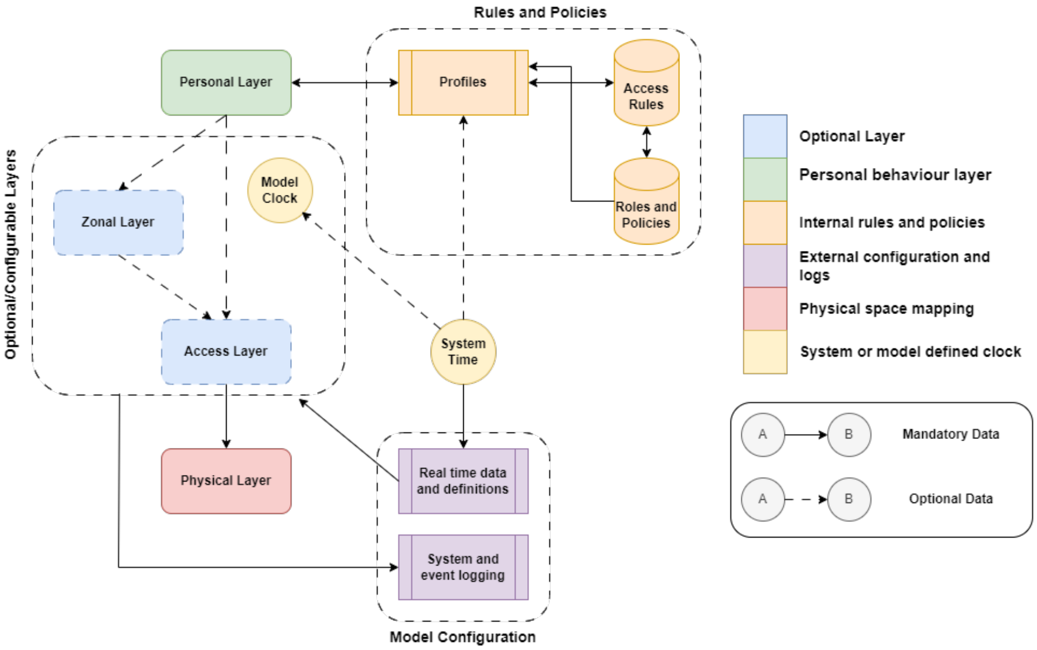

Figure 1: Layers of the proposed access control model

Each time an update is detected the model engine performs a *sanity check* to ensure that none of the proposed changes leaves the model in a broken or inconsistent state. What constitutes a *broken* state will evolve as part of the design, but should include any change that negatively impacts the functionality and behaviour of the model after the update changes have been made. If any of the potential new changes would cause such a broken state the entire update is ignored and discarded. Partial updates are not supported in order to make the update mechanisms simple and streamlined as possible. Updates are applied atomically ensuring the model configuration is always in a known stable state.

VI. Proposal Analysis

Table I shows how each of the historic access control models discussed in this paper compares with our proposed hybrid solution. The table is an amalgamation of many different viewpoints and provides a subset of the regularly occurring metrics used for comparison within these sources. The main criteria for inclusion was to identify some of those open issues with each model which could be targeted as areas for potential improvement by our solution. Several of these open issues were chosen based on a combination of limitations discussed in [22] as well as other factors such as time limitations and problem complexity. These criteria were used as a comparative baseline as well as a direction of future progression but still represent a small proportion of the open issues with historic access control models leaving plenty of scope for investigation. The idea of creating a single, general hybrid solution is neither a small, quick or simple problem to solve. It has many different, sometimes conflicting, facets all of which need to be addressed to truly get close to a generalised solution. Our model has been developed to be as flexible and configurable as possible to distance itself from the recurring issue of problem-centric solutions. Flexibility is offered through dynamic, runtime configurations and updates allowing it to be used in any domain that can be represented in the form of a graph and whose moving components can be defined as resources within this environment. The model must not only represent the physical layout of a controlled space preserving its topology but also any environmental dynamics that may exist, such as inclinations and spaces split across different levels. Any physical access restrictions that exist in the real world also need to be modeled not just the logical restrictions imposed by access control. The representation of relationships between resources within the model are independent of business structure or hierarchy and have been made such that these relationships can be altered as needed by the problem and the needs of the domain. Table II shows how our model improves on histor

VII. Conclusion

Common problems with the traditional models discussed are the covert channels of control and an inability to describe restrictions and prohibitions in control policy. Each historic model type has focused on solving specific issues. Models have traditionally not considered the bigger problem of a general purpose solution and its possible broader application. There are two broad directions for future model development; the first direction consists in developing a generic model that is widely adopted without domain influence; the second direction is to build a series of iterative improvement models based on ZT trust principles in a way similar to that of previous models. Regardless of the future direction, consideration will

	Our Model	**MAC**	**DAC**	**RBAC**	**ABAC**
Model flexibility and consistency	**Yes**	No	Partly	Partly	Yes
Hierarchical inheritance	**Yes**	Yes	No	Yes	No
Proximity limitations	**Partly**	No	No	No	No
Model reflects business structure	**Yes**	Yes	No	Yes	Partly
Time interval access restrictions	**Yes**	No	No	No	No
Model reflects physical environment	**Partly**	No	No	No	No
Role-based support	**No**	No	No	Yes	No
Real-time updates to model	**Yes** **Yes**	No	No	No	No
Location grouping	**Partly**	No	No	No	No

Table II: Feature comparison with historic models

need to be given to the obfuscation/encryption of data, both at rest and in transit, since any practical application will need to ensure the data is stored and transmitted in a secure and integrity preserving manner. The ideal solution would be to develop a generic non-business/domain specific model which can be used equally well in any situation. This may result in an over complicated solution which may leave itself open to unforeseen security and integrity issues. The best offering now is that of incremental ZT and the need to validate any and all requests for access. As systems become more complex this could lead to ever increasing performance issues which in turn could possibly force the need to return to the path of developing a more generic solution.

References

[1] Z. Li et al. Preserving data privacy via federated learning: Challenges and solutions. *IEEE Consumer Electronics Magazine, 9(3):8–16.*, 2020.

[2] K. Kaloyanova et al. Addressing data quality in healthcare. In *Information Systems and Grid Technologies, ISGT, CEUR Workshop Proceedings, 2933*, 2021.

[3] C. Wilcox, S. Djahel, and V. Giagos. Identifying the main causes of medical data incompleteness in the smart healthcare era. *International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates, 2021, pp. 1-6*, 2023.

[4] L. Zhong. A survey of prevent and detect access control vulnerabilities. *University of California, San Diego*, 2023.

[5] E. Krastev et al. Software implementation of the eu patient summary with archetype concept. *The Eighth International Conference on Global Health Challenges, Porto, Portugal, Sep 2-26, pp. 8-13*, 2019.

[6] A. Thakur et al. Data encoding for healthcare data democratization and information leakage prevention. *Nat Commun 15, 1582*, 2024.

[7] Cai L. and Y. Zhu. The challenges of data quality and data quality assessment in the big data era. *Data Science Journal*, pages 1–10, 2019.

[8] O. Popoola et al. A critical literature review of security and privacy in smart home healthcare schemes adopting iot & blockchain: Problems, challenges and solutions. *Blockchain: Research and Applications, 27th April*, 2023.

[9] D.E. Bell and L.J. LaPadula. Secure computer systems: Mathematical foundations. *National Technical Information Service, ADD 760-768*, 1973.

[10] X. Ge et al. Secure databases: An analysis of clark-wilson model in a database environment. *Biomedical Sciences Instrumentation 3084:234-247*, 2004.

[11] V.C. Hu, R. Kuhn, and D. Yaga. Verification and test methods for access control policies/models. *NIST Special Publication 800-192*, 2017.

[12] S. Parkin and S. Khan. A survey on empirical security analysis of access-control systems: A real-world perspective. *ACM Computer Survey, Vol. 55, No. 6, Article 123*, 2022.

[13] G. Sahani et al. Scalable rbac model for large-scale applications with automatic user-role assignment. *International Journal of Communication Networks and Distributed Systems 28(1):76*, 2022.

[14] M.U. Aftab et al. Traditional and hybrid access control models: A detailed survey. *Security and Communication Networks, vol. 2022, Article ID 1560885*, 2022.

[15] D.F. Ferraiolo et al. Proposed nist standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)4, 3*, page 224–274, 2001.

[16] D. Servos and S. Osborn. Current research and open problems in attribute based access control. *ACM Computing Surveys*, 49, 2017.

[17] H. Wang and S.L Osborn. Static and dynamic delegation in the role graph model. *IEEE Transactions on Knowledge and Data Engineering*, pages 1569–1582, 2011.

[18] H.S. Alipour and M. Sabbari. Definition of action and attribute based access control rules for web services. *Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management Istanbul, Turkey, July 3 – 6*, 2012.

[19] S.W. Rose et al. Zero trust architecture (nist). *NIST website*, 2022.

[20] L. Rosser and Norton J.H. A systems perspective on technical debt. *IEEE Aerospace Conference*, 2021.

[21] P. Petrov et al. Incremental transformation of legacy informationsystems. *Conference: 21st SGEM International Multidisciplinary Scientific GeoConference Proceedings.*, 2021.

[22] N. Soveizi et al. Security and privacy concerns in cloud-based scientific and business workflows: A systematic review. *Future Generation Computer Systems*, pages 184–200, 2023.