CognitivaTech: Ingeniería de Software Inteligente y Sistemas Adaptativos. 2024; 1(1):3
doi: 10.63688/cognitivatech.v1.i1.3

ORIGINAL

Methodological model for authorization management in microservices-based applications

Modelo metodológico para la gestión de autorización en aplicaciones basadas en microservicios

Daniel Oswaldo Ramírez Guevara[1]

[1] Instituto Sapiens de Investigación Científica Multidisciplinar. Milagro, Ecuador.

Recibido: 20-05-2024 | Revisado: 03-06-2024 | Aceptado: 20-07-2024 | Publicado: 22-07-2024

Autor de correspondencia: Daniel Oswaldo Ramírez Guevara

ABSTRACT

The growing adoption of microservices-based architectures has transformed software development by enabling distributed, scalable, and loosely coupled systems; however, this evolution has also increased the complexity of access control management, particularly in terms of authorization. In this context, this research proposes a methodological approach to systematically integrate authorization policies into the software development lifecycle of microservice-based applications. The study follows a descriptive–analytical design structured into four stages: authorization requirements analysis, policy formalization, implementation within a distributed architecture, and validation through a case study. The results show that the use of the Attribute-Based Access Control (ABAC) model allows the definition of more precise and flexible rules compared to traditional role-based approaches, reducing redundancy and improving system adaptability. Additionally, separating authorization logic from business logic proved essential for enhancing maintainability and scalability, enabling policy modifications without affecting the services. The validation phase demonstrated that the early integration of authorization helps maintain consistency across requirements, design, and implementation, avoiding inconsistencies and rework. However, the findings also indicate that properly defining attributes requires a rigorous initial analysis. In conclusion, the proposed approach improves authorization management in microservices by providing a structured, coherent, and adaptable framework, contributing to the development of more secure and efficient systems.

Keywords: microservices; access control; ABAC; software security.

RESUMEN

La creciente adopción de arquitecturas basadas en microservicios ha transformado el desarrollo de software al favorecer sistemas distribuidos, escalables y desacoplados; sin embargo, esta evolución ha incrementado la complejidad en la gestión del control de acceso, especialmente en lo referente a la autorización. En este contexto, la presente investigación propone un enfoque metodológico para integrar de manera sistemática políticas de autorización dentro del ciclo de desarrollo de aplicaciones basadas en microservicios. El estudio adopta un diseño descriptivo–analítico, estructurado en cuatro etapas: análisis de requisitos de autorización, formalización de políticas, implementación en una arquitectura distribuida y validación mediante un caso de estudio. Como resultado, se identificó que el uso del modelo basado en atributos (ABAC) permite definir reglas más precisas y flexibles en comparación con enfoques tradicionales basados en roles, reduciendo la redundancia y mejorando la adaptabilidad del sistema. Asimismo, la separación de la lógica de autorización respecto a la lógica de negocio demostró ser clave para mejorar la mantenibilidad y escalabilidad, facilitando la modificación de políticas sin afectar los servicios. La validación evidenció que la integración temprana de la autorización contribuye a mantener la coherencia entre requisitos, diseño e implementación, evitando inconsistencias y retrabajos. No obstante, se identificó que la correcta definición de atributos requiere un análisis inicial riguroso. En conclusión, el enfoque propuesto mejora la gestión de la autorización en microservicios al proporcionar un marco estructurado, coherente y adaptable, contribuyendo al desarrollo de sistemas más seguros y eficientes.

Palabras clave: microservicios; control de acceso; ABAC; seguridad de software.

INTRODUCCION

La adopción de arquitecturas basadas en microservicios ha transformado de manera significativa el desarrollo de sistemas de software modernos, permitiendo sustituir estructuras monolíticas por conjuntos de servicios independientes, especializados y altamente escalables(Newman, 2015; Sidler et al., 2022). Este enfoque facilita la modularidad, la reutilización de componentes y la evolución continua de las aplicaciones, ya que cada microservicio puede desarrollarse, desplegarse y mantenerse de forma autónoma. Asimismo, la interacción entre estos servicios se realiza mediante interfaces bien definidas, generalmente a través de APIs, lo que promueve la interoperabilidad y la integración en entornos distribuidos(Fielding, 2000).

No obstante, esta descentralización de la lógica de negocio introduce nuevos desafíos, especialmente en lo relacionado con la gestión de la seguridad. A medida que el número de servicios crece y se incrementa la comunicación entre ellos, también aumenta la complejidad para garantizar un control de acceso adecuado(Yarygina& Bagge, 2018). En este contexto, aspectos como la autenticación y la autorización adquieren un papel fundamental, ya que determinan quién puede acceder a los recursos del sistema y bajo qué condiciones(Gollmann, 2010). La naturaleza distribuida de los microservicios dificulta la implementación de mecanismos de control de acceso centralizados, lo que obliga a adoptar enfoques más flexibles y granulares(Chandramouli, 2019).

Uno de los principales retos radica en definir políticas de autorización que sean suficientemente expresivas para adaptarse a distintos contextos, pero a la vez estructuradas y sistemáticas para facilitar su implementación y mantenimiento. En particular, los modelos de control de acceso basados en atributos surgen como una alternativa adecuada en entornos dinámicos, ya que permiten tomar decisiones considerando múltiples características del usuario, del recurso y del entorno(Huet al., 2015;Yuan& Tong, 2005). Sin embargo, su integración dentro del ciclo de desarrollo de software aún presenta limitaciones, debido a la falta de metodologías claras que orienten su definición desde las etapas tempranas del diseño(Sänger&Abeck, 2023).

En muchos casos, los mecanismos de seguridad son incorporados de manera tardía en el proceso de desarrollo, lo que puede generar inconsistencias, vulnerabilidades y mayores costos de implementación(Devanbu& Stubblebine, 2000). Por ello, resulta necesario abordar la autorización como un requisito no funcional desde las primeras fases del desarrollo, integrándola de forma transversal en el análisis, diseño e implementación del sistema(Firesmith, 2003). Esto implica no solo definir qué debe ser protegido, sino también establecer cómo se formulan, implementan y validan las políticas de acceso en entornos distribuidos.

En este sentido, el presente enfoque propone una visión integral para la gestión de la autorización en aplicaciones basadas en microservicios, orientada a estructurar la definición de políticas, facilitar su implementación técnica y promover su incorporación sistemática dentro del proceso de desarrollo de software. De esta manera, se busca reducir la complejidad inherente a estos sistemas, mejorar la seguridad y garantizar una gestión coherente del acceso a los recursos en arquitecturas modernas.

METODOS

La presente investigación adopta un enfoque aplicado con un diseño descriptivo–analítico, orientado a estructurar y evaluar un proceso sistemático para la definición, implementación e integración de políticas de autorización en arquitecturas basadas en microservicios. La metodología se construye alineando conceptos de ingeniería de software, seguridad informática y control de acceso, con énfasis en modelos basados en atributos (ABAC) y su incorporación dentro del ciclo de vida del desarrollo.

En una primera fase, se realiza una revisión dirigida de literatura técnica y científica enfocada en tres ejes: arquitecturas de microservicios, mecanismos de control de acceso y estrategias de seguridad en sistemas distribuidos. Esta revisión permite identificar limitaciones en la formulación de políticas de autorización y su integración en procesos de desarrollo, particularmente en entornos donde predominan servicios desacoplados y comunicación mediante APIs.

A partir de este análisis, se define un modelo metodológico estructurado en cuatro etapas, que responde a la necesidad de integrar la autorización como un requisito no funcional desde fases tempranas del desarrollo:

Análisis de requisitos de autorización

Se identifican los actores del sistema, recursos protegidos y operaciones asociadas (por ejemplo, operaciones tipo CRUD en servicios REST). En esta etapa se establecen atributos relevantes del sujeto, recurso y contexto, que servirán como base para la construcción de políticas ABAC. La especificación se realiza en lenguaje natural estructurado, buscando claridad y trazabilidad con los requisitos funcionales.

Formalización de políticas de acceso

Los requisitos definidos se transforman en políticas formales mediante una estructura sintáctica basada en reglas. Esta formalización permite reducir ambigüedades y facilitar la posterior implementación técnica. Se emplea una representación intermedia que organiza condiciones, atributos y decisiones de acceso, permitiendo su traducción hacia lenguajes de políticas como los utilizados en motores de autorización.

Implementación en arquitectura de microservicios

Validación mediante caso de estudio

El modelo propuesto se evalúa a través de un caso de estudio representativo, en el cual se implementa una aplicación basada en microservicios con requerimientos de control de acceso granular. Se analizan aspectos como consistencia de políticas, facilidad de integración, mantenibilidad y capacidad de adaptación ante cambios en los requisitos. La validación se centra en comprobar la viabilidad del enfoque y su contribución a la reducción de la complejidad en la gestión de autorización.

Finalmente, los resultados obtenidos se analizan de forma cualitativa, considerando criterios de coherencia entre requisitos y políticas, separación de responsabilidades y alineación con principios de seguridad en sistemas distribuidos. Este enfoque metodológico permite no solo estructurar la autorización en microservicios, sino también integrarla de manera sistemática dentro del proceso de desarrollo, abordando una de las principales limitaciones identificadas en el estado actual.

RESULTADOS

La aplicación del modelo metodológico permitió estructurar de forma consistente la definición e implementación de políticas de autorización en una arquitectura basada en microservicios. Los resultados se presentan en función de las etapas propuestas: especificación, formalización, implementación y validación.

Resultados del análisis de requisitos

Se identificaron tres tipos principales de actores (usuario estándar, administrador y servicio interno), así como cinco recursos críticos (usuarios, órdenes, reportes, configuración y auditoría). A partir de estos elementos se definieron atributos relevantes para el modelo ABAC, diferenciando entre atributos del sujeto, del recurso y del entorno.

Estos atributos permitieron construir reglas más expresivas en comparación con modelos tradicionales basados únicamente en roles.

Resultados de la formalización de políticas

Las políticas de autorización fueron transformadas desde descripciones en lenguaje natural hacia estructuras formales basadas en reglas condicionales. Esto permitió eliminar ambigüedades y facilitar su implementación en motores de política.

Tabla 1. Ejemplo de atributos definidos para políticas ABAC

Categoría	Atributo	Descripción
Sujeto	rol	Tipo de usuario (admin, user, service)
Sujeto	id_usuario	Identificador único del usuario
Recurso	tipo_recurso	Tipo de entidad (orden, reporte, etc.)
Recurso	propietario	Usuario propietario del recurso
Entorno	hora_acceso	Hora en la que se realiza la solicitud
Entorno	ip_origen	Dirección IP de origen

Tabla 2. Ejemplo de transformación de política

Lenguaje natural	Política formal (estructura lógica)
Un usuario puede ver su propia orden	permitir si (rol=user) ∧ (id_usuario = propietario)
Un administrador puede acceder a todos los reportes	permitir si (rol=admin)
Acceso restringido fuera de horario laboral	denegar si (hora_acceso < 08:00 ∨ hora_acceso > 18:00)

La formalización evidenció que el uso de atributos permite definir condiciones más precisas, reduciendo reglas redundantes.

Resultados de la implementación

Las políticas fueron implementadas utilizando un componente externo de autorización desacoplado de los microservicios. Este componente evaluó las solicitudes entrantes antes de permitir el acceso a los servicios.

Se observó que:

- La separación entre lógica de negocio y autorización redujo la complejidad del código en los microservicios.
- Las políticas pudieron modificarse sin necesidad de redeplegar los servicios.
- La validación de tokens y atributos permitió aplicar control de acceso en cada solicitud de manera uniforme.

Resultados de la validación

El caso de estudio demostró que el modelo permite gestionar de forma coherente políticas de autorización en sistemas distribuidos. Se evaluaron escenarios con múltiples servicios interactuando entre sí, donde cada solicitud debía ser validada de acuerdo con las políticas definidas.

Tabla 3. Evaluación de implementación

Criterio	Resultado observado
Acoplamiento	Bajo (lógica de autorización desacoplada)
Escalabilidad	Alta (componente reutilizable)
Mantenibilidad	Alta (políticas centralizadas)
Flexibilidad	Alta (adaptación a nuevos atributos)

Se identificaron los siguientes hallazgos:

- La coherencia entre requisitos y políticas se mantuvo durante todo el ciclo de desarrollo.
- La incorporación temprana de la autorización evitó inconsistencias en fases posteriores.
- El uso de ABAC permitió manejar escenarios complejos sin incrementar significativamente el número de reglas.

Tabla 4. Comparación antes vs después del modelo

Aspecto	Enfoque tradicional	Modelo propuesto
Definición de políticas	No estructurada	Estructurada y trazable
Integración	Tardía	Desde análisis
Control de acceso	Basado en roles	Basado en atributos (ABAC)
Adaptabilidad	Limitada	Alta

En conjunto, los resultados evidencian que el modelo propuesto mejora la claridad en la definición de políticas, facilita su implementación en entornos de microservicios y reduce la complejidad asociada a la gestión del control de acceso en sistemas distribuidos.

DISCUSION

Los resultados obtenidos evidencian que la incorporación estructurada de políticas de autorización dentro del ciclo de desarrollo de microservicios no solo mejora la organización del sistema, sino que también reduce problemas típicos asociados a la gestión de seguridad en entornos distribuidos. A diferencia de enfoques tradicionales donde el control de acceso se implementa de forma reactiva, el modelo aplicado permitió establecer una relación directa entre los requisitos iniciales y las políticas finalmente desplegadas.

Uno de los aspectos más relevantes es la efectividad del modelo basado en atributos (ABAC) para representar escenarios reales. Los resultados muestran que el uso de atributos del sujeto, recurso y entorno permitió definir reglas más precisas sin necesidad de incrementar significativamente el número de políticas. Esto contrasta con modelos basados únicamente en roles, donde la escalabilidad suele verse afectada por la proliferación de combinaciones de permisos. En este sentido, la reducción de redundancia observada en la formalización confirma que ABAC se adapta mejor a la naturaleza dinámica de los microservicios.

Por otro lado, la separación de la lógica de autorización respecto a la lógica de negocio demostró ser un factor clave en la mantenibilidad del sistema. Los resultados de implementación indican que los cambios en políticas no requirieron modificaciones en los servicios, lo que reduce el riesgo de introducir errores y facilita la evolución del sistema. Este hallazgo es coherente con los principios de desacoplamiento característicos de las arquitecturas de microservicios, donde cada componente debe poder evolucionar de manera independiente.

Sin embargo, los resultados también ponen en evidencia ciertos desafíos. La necesidad de definir atributos adecuados desde la fase de análisis implica un mayor esfuerzo inicial en comparación con enfoques más simples. Si los atributos no son correctamente identificados, las políticas pueden volverse incompletas o excesivamente complejas. Además, aunque la centralización de políticas mejora la gestión, introduce una dependencia crítica en el componente de autorización, lo que requiere garantizar su disponibilidad y rendimiento.

En relación con la validación, el caso de estudio confirmó que la integración temprana de la autorización contribuye a mantener la coherencia del sistema a lo largo del desarrollo. La trazabilidad entre requisitos, políticas y su implementación permitió identificar inconsistencias de manera anticipada, evitando retrabajos en etapas avanzadas. Este resultado sugiere que tratar la autorización como un requisito no funcional desde el inicio no solo mejora la seguridad, sino también la calidad global del proceso de desarrollo.

En conjunto, la discusión de los resultados indica que el modelo propuesto logra un equilibrio entre flexibilidad y control en la gestión de autorización en microservicios. Si bien introduce cierta complejidad en las fases iniciales, los beneficios en términos de mantenibilidad, escalabilidad y coherencia justifican su adopción en sistemas distribuidos que requieren control de acceso granular.

CONCLUSION

El desarrollo de esta investigación permitió demostrar que la gestión de la autorización en arquitecturas basadas en microservicios puede abordarse de manera más efectiva cuando se integra de forma estructurada dentro del ciclo de desarrollo de software. A través del modelo propuesto, se logró establecer una relación clara entre los requisitos de autorización, su formalización y su implementación, reduciendo ambigüedades y mejorando la coherencia del sistema.

Los resultados evidencian que el uso de un enfoque basado en atributos (ABAC) proporciona una mayor capacidad de adaptación frente a escenarios dinámicos, permitiendo definir políticas más precisas sin incrementar innecesariamente su complejidad. Asimismo, la separación entre la lógica de negocio y la lógica de autorización contribuyó significativamente a la mantenibilidad y escalabilidad de la arquitectura, facilitando la evolución del sistema sin afectar sus componentes principales.

La integración temprana de la autorización como requisito no funcional se consolidó como un factor determinante para evitar inconsistencias y retrabajos en etapas posteriores del desarrollo. Este enfoque no solo fortalece la seguridad del sistema, sino que también mejora la calidad del proceso de ingeniería de software al promover una visión más integral y sistemática.

No obstante, se identificó que la correcta definición de atributos y políticas requiere un esfuerzo inicial considerable, lo que implica la necesidad de contar con lineamientos metodológicos claros y personal capacitado. A pesar de ello, los beneficios obtenidos en términos de control, flexibilidad y coherencia justifican la adopción del modelo en entornos distribuidos.

En conclusión, el enfoque propuesto constituye una alternativa viable para abordar uno de los principales desafíos en arquitecturas de microservicios: la implementación de mecanismos de autorización robustos, escalables y alineados con el proceso de desarrollo, contribuyendo así a la construcción de sistemas más seguros y sostenibles.

Referencias

1. Chandramouli, R., Butcher, Z., & Chetal, A. (2021). Attribute-based access control for microservices-based applications using a service mesh. NIST. https://doi.org/10.6028/NIST.SP.800-204B
2. Devanbu, P., & Stubblebine, S. (2000). Software engineering for security: A roadmap. https://doi.org/10.1145/336512.336584
3. Ferraiolo, D., Kuhn, R., & Chandramouli, R. (2016). Role-based access control. Artech House. https://doi.org/10.1201/9781315369440
4. Ghotbi, S. H., & Fischer, B. (2013). Fine-grained role- and attribute-based access control for web applications. https://doi.org/10.1007/978-3-642-38709-8_12
5. Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute-based encryption for fine-grained access control. https://doi.org/10.1145/1180405.1180418
6. Hu, V. C., Kuhn, D. R., Ferraiolo, D. F., & Voas, J. (2015). Attribute-based access control. Computer, 48(2), 85–88. https://doi.org/10.1109/mc.2015.33
7. Hu, V. C., Kuhn, D. R., & Ferraiolo, D. F. (2018). Access control for emerging distributed systems. Computer, 51(10), 100–103. https://doi.org/10.1109/mc.2018.3971365
8. Khare, S., Thakur, P., Talwandi, N. S., & Yadav, V. (2024). Securing microservice architecture: Load balancing and role-based access control. https://doi.org/10.63503/j.ijaimd.2024.7
9. Madkaikar, G., Sural, S., Vaidya, J., & Atluri, V. (2024). Queuing theoretic analysis of dynamic attribute-based access control systems. https://doi.org/10.1007/978-3-031-65175-5_23
10. Nehme, A., Jesus, V., Mahbub, K., & Abdallah, A. (2019). Fine-grained access control for microservices. https://doi.org/10.1007/978-3-030-18419-3_19
11. Newman, S. (2015). Building microservices. O'Reilly Media.
12. Salehi, S. A., Han, R., Rudolph, C., & Grobler, M. (2023). DACP: Enforcing a dynamic access control policy in cross-domain environments. Computer Networks, 237, 110049. https://doi.org/10.1016/j.comnet.2023.110049
13. Sänger, N., & Abeck, S. (2023). User authorization in microservice-based applications. Software, 2(3), 400–426. https://doi.org/10.3390/software2030019
14. Sandhu, R., Coyne, E., Feinstein, H., & Youman, C. (1996). Role-based access control models. https://doi.org/10.1109/2.485845
15. Sandhu, R., & Samarati, P. (1994). Access control: Principle and practice. https://doi.org/10.1109/35.312842
16. Schneider, M., Zieschinski, S., & Abeck, S. (2021). A test concept for microservice-based applications. https://doi.org/10.1109/sose52839.2021.00025
17. Sidler, J., Braun, E., Schmitt, C., Schlachter, T., & Hagenmeyer, V. (2022). Microservice-based architecture for integration systems. https://doi.org/10.1007/978-3-030-88063-7_3
18. Teerakanok, S., Uehara, T., & Inomata, A. (2021). Migrating to zero trust architecture: Reviews and challenges. https://doi.org/10.1155/2021/9947347
19. Wang, H., Chen, Y., & Xu, Z. (2023). Decentralized access control for secure microservices cooperation with blockchain. ISA Transactions, 141, 44–51. https://doi.org/10.1016/j.isatra.2023.07.018
20. Yuan, E., & Tong, J. (2005). Attribute-based access control (ABAC) for web services. https://doi.org/10.1109/icws.2005.25

Financiación

Los autores no recibieron financiamiento para el desarrollo de esta investigación.

Conflictos de interés

Los autores afirman que no existen conflictos de intereses en este estudio y que se han seguido éticamente los procesos establecidos por esta revista. Además, aseguran que este trabajo no ha sido publicado parcial ni totalmente en ninguna otra revista.

Contribución de autoría

Conceptualización: DORG
Curación de datos: DORG
Análisis formal: DORG
Investigación: DORG
Metodología: DORG
Administración del proyecto: DORG
Recursos: DORG
Software: DORG
Supervisión: DORG
Validación: DORG
Visualización: DORG
Redacción – Borrador original: DORG
Redacción – Revisión y edición: DORG