



WHITE PAPER

The Ultimate Guide to Choosing the Right Authorization Language

Controlling access to resources, ensuring data privacy, and enforcing security policies are more critical than ever. The key to achieving these goals lies in choosing the right authorization language. With a variety of options available, each with unique strengths, understanding which one best suits your specific needs is essential. This guide provides a comprehensive overview of popular authorization languages, their ideal use cases, and key considerations to help you make an informed decision.

Overview of popular authorization languages

When it comes to authorization, several languages are commonly used, each tailored to specific scenarios and requirements. Whether you're dealing with infrastructure management, cloud services, or complex access control needs, understanding the capabilities and limitations of each language is critical.

In this document, we'll explore the most prominent languages in this space:

1. **ALFA (Abbreviated Language for Authorization)**
2. **Open Policy Agent's Rego**
3. **AWS Cedar**
4. **XACML (eXtensible Access Control Markup Language)**

ALFA: The familiar syntax

ALFA is a domain-specific language designed to simplify the process of creating and maintaining authorization policies. It provides a more intuitive and less verbose syntax compared to traditional languages like XACML. With its syntax resembling popular programming languages like C# and Java, ALFA offers a developer-friendly alternative for those who find verbose languages challenging.

Use cases

- **Fine-grained access control:** ALFA is ideal for scenarios requiring detailed access control, such as granting access to medical records or managing Kubernetes resources.
- **Negative policies:** ALFA excels in expressing negative policies, enabling dynamic segregation of duties (SoD).
- **Relationship-based access control:** ALFA allows for nuanced access decisions by comparing attributes of users and resources.
- **Hierarchical use cases:** ALFA has unique native built-in support for hierarchies, policy nesting, and policy conflict resolution. This allows for a broader range of use cases not possible with other languages.

- **Cross-team collaboration:** Leverage ALFA's built-in hierarchical structure to enable cross-team collaboration. This allows a team to write global authorization policy checks (e.g. compliance) and combine those with application-specific policies.

Benefits

- **Rich and expressive:** ALFA is built on attributes and is designed for fine-grained access control, supporting common patterns like attribute-based access control (ABAC), role-based access control (RBAC), and relationship-based access control (ReBAC).
- **Efficient:** ALFA's policy structure enables the implementation of complex authorization scenarios with minimal policies, making it efficient at both runtime and design time.
- **Compliance-driven:** ALFA allows organizations to demonstrate compliance by enabling the design of policies that deny access by default and simplify conflict resolution.

Challenges

- **Limited attribute schema:** ALFA relies on a basic attribute schema (attribute name and datatype) that doesn't provide additional information such as attribute relationship or cardinality.

Rego: The powerhouse

Rego is a powerful programming language based on Datalog, designed to handle complex access control scenarios. Unlike ALFA, Rego can change the system state, although this is generally discouraged to maintain predictability. Rego operates on JSON data structures, making it highly versatile and suitable for a wide range of use cases.

Use cases

- **Infrastructure scenarios:** Rego is particularly strong in managing access control within Kubernetes environments and other infrastructure use cases.
- **Complex policies:** Like ALFA, Rego can implement complex authorization use cases.

Benefits

- **Flexible and powerful:** Rego's flexibility allows it to handle a wide variety of policy models and logics, making it suitable for complex, dynamic systems.
- **Inline data manipulation:** Rego policies can also be used to manipulate data (reason over JSON documents) making it versatile.

Challenges

- **Learning curve:** Mastering Rego and its features requires significant effort, especially for those unfamiliar with declarative logic languages.
- **Statefulness:** While Rego can change system states, this flexibility can introduce unpredictability if not managed carefully.
- **No clear delineation:** Rego can do it all and while it sounds like a blessing, it's also a trap. Rego can be used to fetch attribute values and process complex JSON payload thus blurring the line between authorization policy logic and data management.

Cedar: AWS' lightweight approach

Cedar is a domain-specific language created by AWS to simplify permissions within its ecosystem. It is lightweight, with a flat policy structure, and focuses on making policy authoring straightforward.

Use cases

- **AWS ecosystem:** Cedar is best suited for AWS-specific use cases, seamlessly integrating with AWS services.
- **Simplified permissions:** Cedar is ideal when straightforward policies without complex hierarchies are needed.

Benefits

- **Lightweight and constrained:** Cedar's simplicity makes it easy to implement and maintain, especially in AWS environments.
- **Rich attribute schema:** Cedar comes with a rich schema to define attributes that are to be used in the policies.

Challenges

- **AWS-specific:** Cedar's utility is largely confined to AWS, limiting its applicability in other environments.
- **Lacks hierarchical policies:** Cedar's flat policy structure can be limiting in complex scenarios requiring hierarchical policy definitions.
- **Scale:** With Cedar it's hard to scale the management of policies unless a structure is defined externally.

XACML (eXtensible Access Control Markup Language): The standard bearer

XACML is a widely-adopted standard language and framework for expressing and enforcing authorization policies. It uses extensible markup language (XML) to define policies, consisting of rules, policies, and policy sets. A Policy Decision Point (PDP) evaluates XACML policies to grant or deny access based on attributes like subject, resource, action, or environment.

Use cases

- **Interoperable and standardized access control:** XACML is ideal for managing access control in heterogeneous and distributed systems, such as web applications, cloud services, and enterprise systems.
- **ABAC and RBAC models:** XACML supports the implementation of attribute-based access control and role-based access control models.
- **Policy Analysis and Verification:** XACML's structured format allows for thorough policy analysis and verification, ensuring robust access control.

Benefits

- **Comprehensive and expressive:** XACML offers a highly detailed way of writing authorization policies, supporting complex access control scenarios.
- **Standardized approach:** As a widely-adopted standard, XACML ensures interoperability across different platforms and systems.
- **Policy modularity and reuse:** XACML supports the creation of modular policies that can be reused and combined, simplifying policy management.

Challenges

- **Verbose and complex:** XACML is often criticized for its verbosity and complexity, which can make policy authoring and management cumbersome.
- **Steep learning curve:** The XML-based syntax and comprehensive nature of XACML require significant expertise and effort to master.

Conclusion

Selecting the right authorization language requires careful consideration of your system's architecture, policy requirements, and the specific challenges you face. Whether you opt for ALFA's familiarity, Rego's power, Cedar's simplicity, or XACML's standardization, your choice will significantly impact your system's security, scalability, and maintainability.

When making your decision, consider the following key factors:

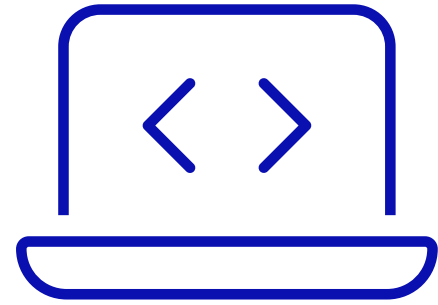
- **Compatibility:** How well does the language integrate with your existing systems and frameworks?
- **Expressiveness:** Does the language offer the flexibility needed to meet your policy requirements?
- **Performance:** Performance: Will the policy engine scale effectively with your system's demands?
- **Ecosystem support:** Are there tools and resources available to support development and testing?
- **Learning curve:** How steep is the learning curve, and what resources are required to implement the language effectively?

By evaluating these factors, you'll be better equipped to select the authorization language that best fits your needs, ensuring robust and scalable access control for your systems.

Take the next steps

Learn how a security vendor can help secure your access control while allowing you to focus on your core business initiatives.

[Meet with our solution experts for a demo](#) of our Axiomatics Authorization Management Platform and discuss how our approach can save your organization time and money.



Stay connected and informed

Get the latest insights, announcements, and assets from our thought leaders and customer success team:



[Blog articles](#)



[Press releases and announcements](#)



[Resources, solution briefs, and tools](#)



[Join us on LinkedIn](#)



[Subscribe to our LinkedIn Newsletter](#)



[Subscribe to our YouTube channel](#)

Axiomatics - a Leonardo Company is the originator and leading provider of runtime, fine-grained authorization delivered with attribute-based access control (ABAC) for applications, data, APIs, and microservices.

The company enables enterprises to effectively and efficiently connect Axiomatics' award-winning authorization platform to critical security implementations, such as Zero Trust or identity-first security. The world's largest enterprises and government agencies continually depend on Axiomatics' award-winning authorization platform to share sensitive, valuable, and regulated digital assets – but only to authorized users and in the right context.

Follow us on LinkedIn: <https://linkedin.com/company/axiomatics>

Join us on YouTube: <https://youtube.com/@axiomatics>

Learn more or request a demo of our solution:

axiomatics.com

info@axiomatics.com

US Office: +1 (312) 374-3443 | Europe Office: +46 8 51 510 240