

ABAC Mastery: Axiomatics Policy Companion

October 2024

ABAC to School The Basics of Policy Modeling

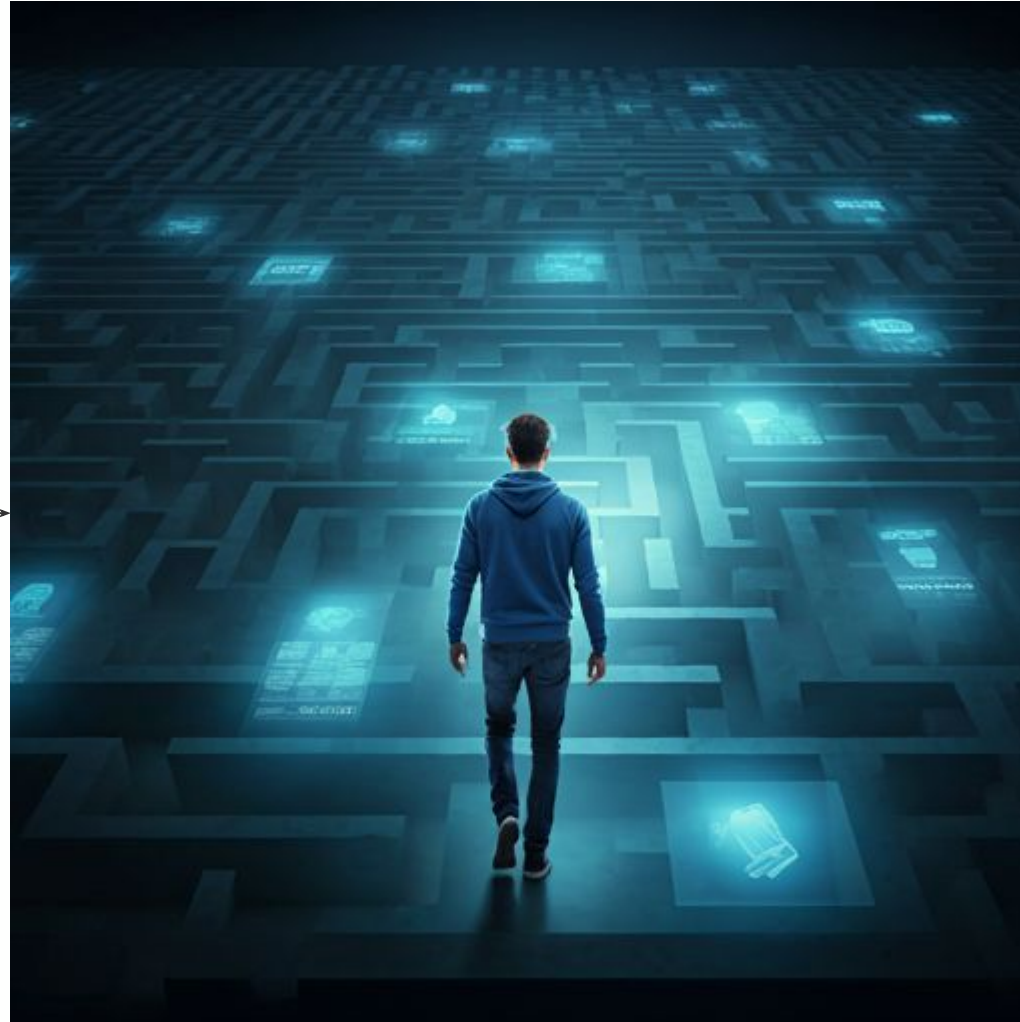
September 2024



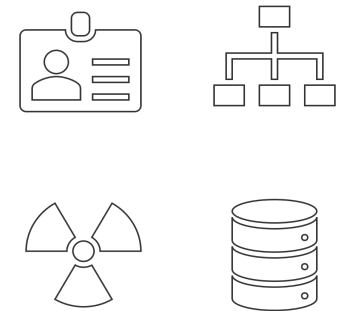
The Challenge of Authorization

Use Cases

- Compliance Standards
- Data Protection
- Reduced Risk
- Open Banking
- Etc.



Attribute Sources



Attribute Values

- Department
- Citizenship
- Risk Score
- Security Clearance
- Training
- Etc.

Identify key natural attributes

Students can view their own grades

Identify key natural attributes

Students can view their own grades

Identify key natural attributes

Students can view their own grades

- Attribute: user.role with Value: Student
- Attribute: user.identity with Value: Biff
- The verb is usually the action, The values view, edit, delete could be values for the attribute action.identity
 - Attribute: action.identity with Value: view
- Resources are the noun being accessed by/protected from the subject (student).
 - Attribute: resource.identity with Value: Grade
 - Attribute: grade.student with Value: Biff
- There is a relationship between the student and the grades. their own grades becomes:
 - $\text{grade.student} == \text{user.identity}$



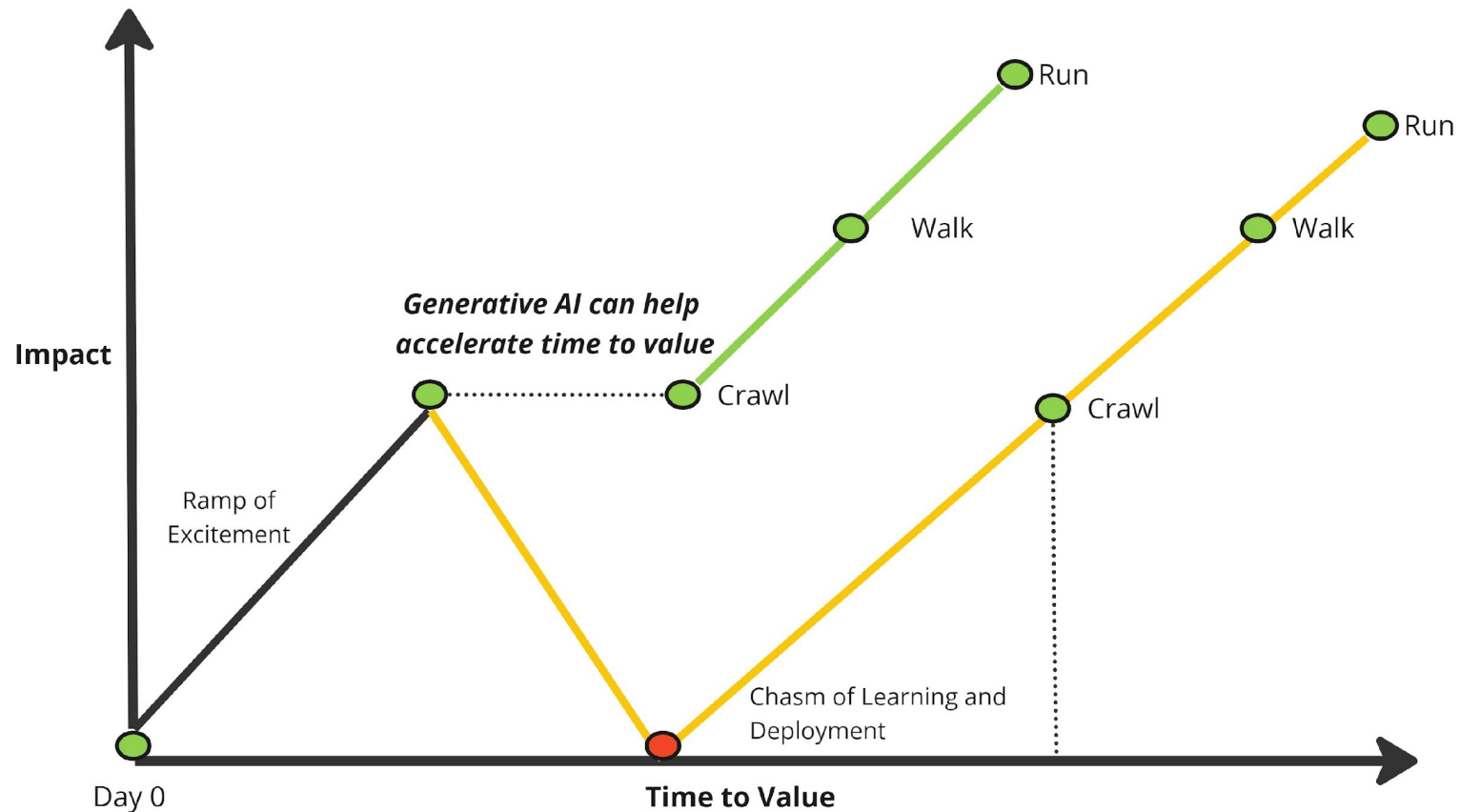
Vision: To enable every organization in the world to enforce Zero Trust access



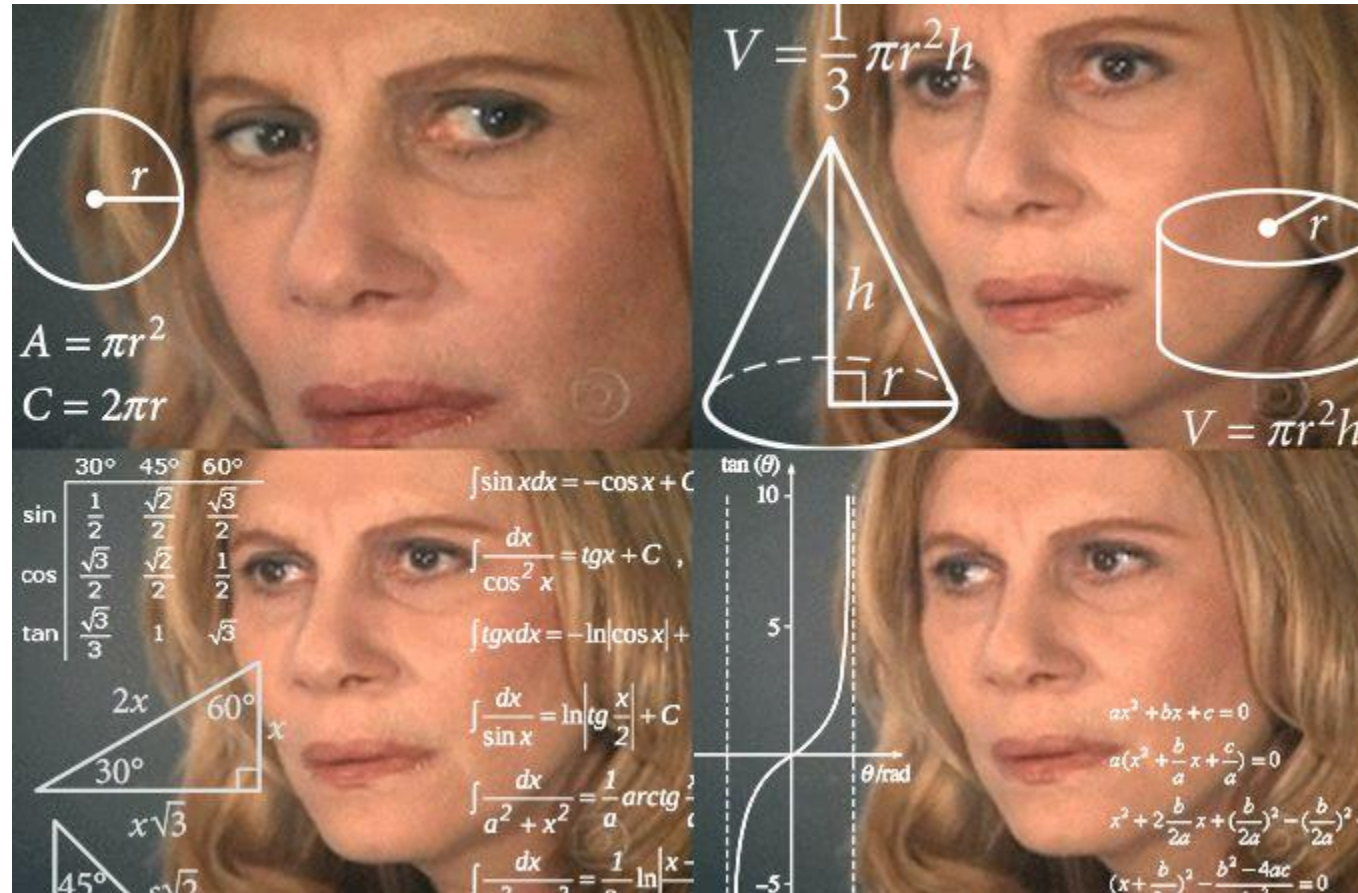
Mission: To securely connect people and information using orchestrated authorization

How Generative AI can help accelerate time to value

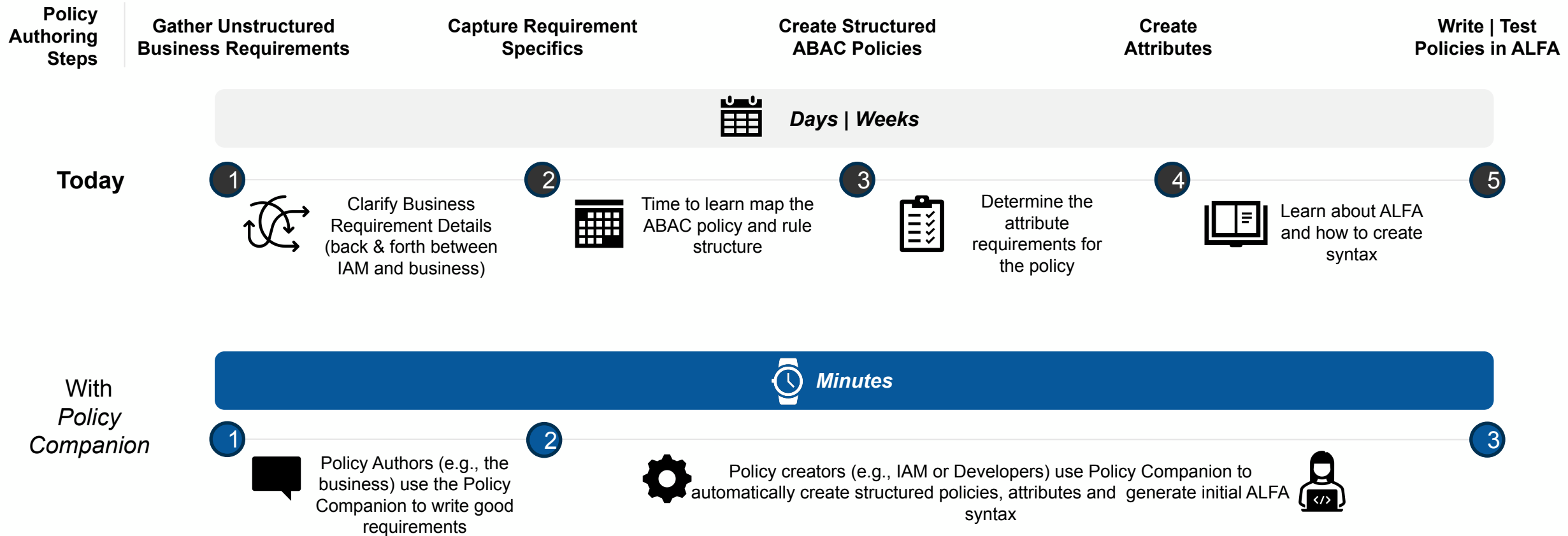
Enterprise software is rarely a linear deployment as production teams need to ramp up and learn how to use and adopt solutions that are often procured by different teams. **Generative AI can accelerate and automate the learning curve** for teams that inherit the responsibility of deploying software to avoid the “deployment chasm”.



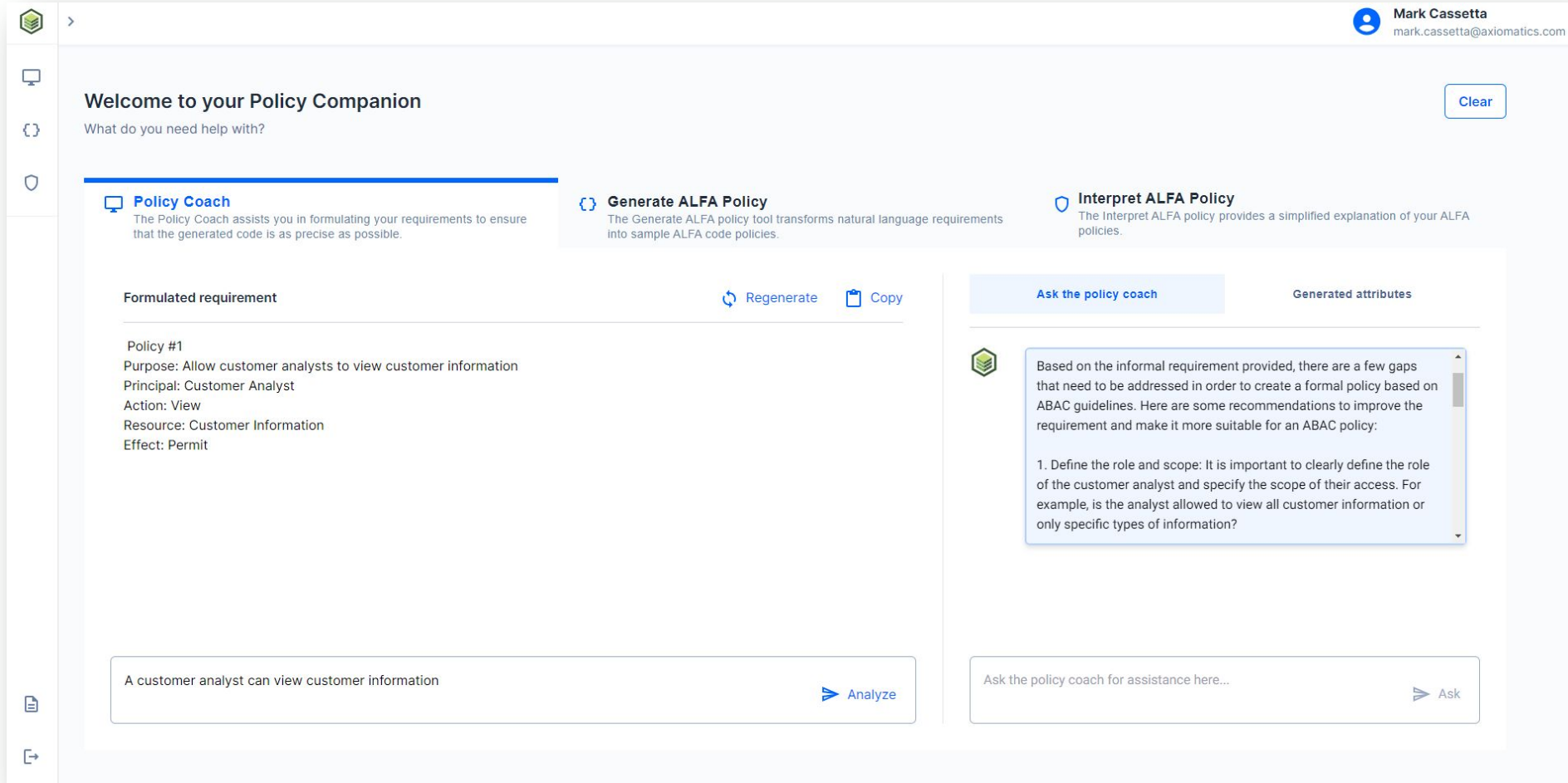
When a manager describes their perfect authorization scenario



Generative AI can accelerate authorization time to value



Axiomatics Policy Companion



The screenshot displays the Axiomatics Policy Companion web application. At the top right, a user profile for Mark Cassetta (mark.cassetta@axiomatics.com) is shown. The main header reads "Welcome to your Policy Companion" with a "Clear" button. A sidebar on the left contains icons for a monitor, code, and a shield. The main content area is divided into three sections: "Policy Coach" (with a description of its role in formulating requirements), "Generate ALFA Policy" (describing the tool's function in transforming natural language into ALFA code), and "Interpret ALFA Policy" (describing the tool's function in providing simplified explanations of ALFA policies). The "Policy Coach" section is active, showing a "Formulated requirement" with details: Policy #1, Purpose: Allow customer analysts to view customer information, Principal: Customer Analyst, Action: View, Resource: Customer Information, and Effect: Permit. Below this, a text input field contains "A customer analyst can view customer information" and an "Analyze" button. To the right, the "Interpret ALFA Policy" section shows a "Generated attributes" box with a message about gaps in the requirement and a recommendation to define the role and scope. Below this, a text input field contains "Ask the policy coach for assistance here..." and an "Ask" button.

Mark Cassetta
mark.cassetta@axiomatics.com

Clear

Policy Coach
The Policy Coach assists you in formulating your requirements to ensure that the generated code is as precise as possible.

Generate ALFA Policy
The Generate ALFA policy tool transforms natural language requirements into sample ALFA code policies.

Interpret ALFA Policy
The Interpret ALFA policy provides a simplified explanation of your ALFA policies.

Formulated requirement [Regenerate](#) [Copy](#)

Policy #1
Purpose: Allow customer analysts to view customer information
Principal: Customer Analyst
Action: View
Resource: Customer Information
Effect: Permit

A customer analyst can view customer information [Analyze](#)

Ask the policy coach **Generated attributes**

Based on the informal requirement provided, there are a few gaps that need to be addressed in order to create a formal policy based on ABAC guidelines. Here are some recommendations to improve the requirement and make it more suitable for an ABAC policy:

1. Define the role and scope: It is important to clearly define the role of the customer analyst and specify the scope of their access. For example, is the analyst allowed to view all customer information or only specific types of information?

Ask the policy coach for assistance here... [Ask](#)

Policy Coach Defines the Requirement

Welcome to your Policy Companion
What do you need help with?

 **Policy Coach**
Assists you in formulating your requirements to ensure that the generated policy is as precise as possible.

 **Generate Policy Syntax**
Converts natural language requirements into machine policies.

Formulated requirement  Regenerate  Copy

...

Policy #1
Purpose: Allow students to view their own grades
Principal: Student
Action: View
Resource: Grades
When: The student is viewing their own grades
Effect: Permit
...

Students can view their own grades  Analyze

 Policy Companion is an AI-based tool. Verify critical information before use. Check the terms.

Policy Coach defines the policy elements

- ✓ Purpose: Policy summary
- ✓ Principle: Who does this policy apply to?
- ✓ Action: What can they do?
- ✓ Resource: Where is the data?
- ✓ When: What are the access conditions?
- ✓ Effect: What is the expected outcome?

Policy Coach Defines the Requirement

Ask for proposals to improve your requirements



Students can view their own grades



grade record, ensuring students can only view grades that belong to them.
- **Effect**: The effect is `permit`, granting access when the specified conditions are met.

This policy framework captures the requirement accurately by ensuring that students are only allowed to view their own grades, providing a secure and well-defined access control mechanism.

Students can view their own grades

➤ Ask

Policy Coach can improve your requirement

- ✓ Provides a clear definition of the attributes to consider
- ✓ Gives you an ALFA policy recommendation
- ✓ Shows a detailed explanation of the defined requirement
- ✓ Summarises the policy back to you

Policy Generator Creates the Code

Policy Coach

Assists you in formulating your requirements to ensure that the generated policy is as precise as possible.

Generate Policy Syntax

Converts natural language requirements into machine readable policies.

Translate Policy Syntax

Converts machine readable policies into natural language

Here's an example of an ALFA authorization policy for the given requirement:  Regenerate  Copy

```
* Rule permitting students to view their own grades.
*/
rule {
  permit
  condition userRole == "student" && actionType == "view"
}
}
```

Students can view their own grades

 Generate ALFA

Attributes

 Update ALFA

 Copy

```
/*
 * Attribute representing the role of the user, specifically if they
 * are a student.
 */
attribute userRole {
  id = "education.userRole"
  category = subjectCat
  type = string
}

/*
 * Attribute representing the type of resource being accessed, in
 * this case, grades.
 */
attribute resourceType {
  id = "education.resourceType"
  category = resourceCat
  type = string
}
```

Policy Interpreter Translates Existing Policies



Policy Coach

Assists you in formulating your requirements to ensure that the generated policy is as precise as possible.



Generate Policy Syntax

Converts natural language requirements into machine readable policies.



Translate Policy Syntax

Converts machine readable policies into natural language

ALFA Code

```
/* The target specifies that this policy applies to resources of type "grades".
*/
target clause resourceType == "grades"
apply denyOverrides

/*
* Rule permitting students to view their own grades.
*/
rule {
  permit
  condition userRole == "student" && actionType == "view"
}
}
```

 Interpret

Interpretation

 Regenerate

 Copy

The policy can be summarized as follows:

- Students are allowed to view their own grades.
- All other users are denied access to student grades.

The namespace education defines the following ALFA authorization policy:

- The target clause is `resourceType == "grades"`, which means that the policy applies to all resources of type "grades".
- The apply `denyOverrides` rule means that if any rule denies access, the access will be denied, even if other rules permit it.
- The rule permits students to view their own grades, provided their role is "student" and the action type is "view".

Policy Companion Demo

Policy Coach

Structures your requirement

Policy Generator

Converts natural language into ALFA

Policy Translator

Converts policies into plain text

Questions

Thank you