NEXTLABS®

White Paper

# Zero Trust Policy Engine: An In-Depth Analysis

Today's organizations face the task of securing a digital core beyond traditional network boundaries, while also ensuring that data can seamlessly traverse through various environments, from cloud infrastructures to mobile technologies. Coupled with the rise in data volumes and the sophistication of cyberattacks, the IT landscape calls for a paradigm shift in data security, propelling the adoption of the Zero Trust Architecture (ZTA).

However, the implementation of ZTA is incomplete without a policy engine. Central to the efficacy of ZTA, the policy engine serves as a software component or system that is responsible for evaluating and enforcing policies or rules within an organization or application.

This white paper delves into the crucial role of the policy engine within a ZTA. We will dive into its function, underlying architecture, benefits, and the challenges associated with its implementation. Additionally, we explore its specific business use cases across various industries, and how the engine fits within NextLabs' data security solution.

## What is a Policy Engine and Why is it needed?

A policy engine is a software component or system that functions as a decision-making mechanism within an organization or application, playing a key role in the enforcement of policies and rules. Its operation is triggered by inputs or events such as user requests, system events, or data updates, to which it applies predefined policies to reach a decision or execute an action. The policies enforced can span across domains such as security, compliance, governance, and business rules.

Within the framework of Zero Trust Architecture, policy engines occupy a critical role in fulfilling the requirements of continuous verification, particularly through the enablement of Attribute-Based Access Control (ABAC) to enforce least privilege access principles. As opposed to the network-based access control where only a single validation is needed to access the network and its resources, ABAC ensures that every access request to a data resource is validated based on user, resource, and environmental attributes. This calls for a policy engine that can evaluate every access attempt dynamically based on a wide range of attributes.

Under Zero Trust and ABAC, a specialized policy engine that evaluates policies based on real-time attributes is referred to as a Dynamic Authorization policy engine. It makes fine-grained access control decisions by evaluating access requests and their subject, resource, and environment attributes against predefined policies. A Dynamic authorization policy engine can be used to implement the ZTA principle of "Least Privileged Access" to only provide the minimum level of access needed by the user at that specific time. Beyond controlling access to application data, files and documents, the policy engine can also enforce granular policies to segregate and obfuscate data as needed.

## How a Policy Engine Works

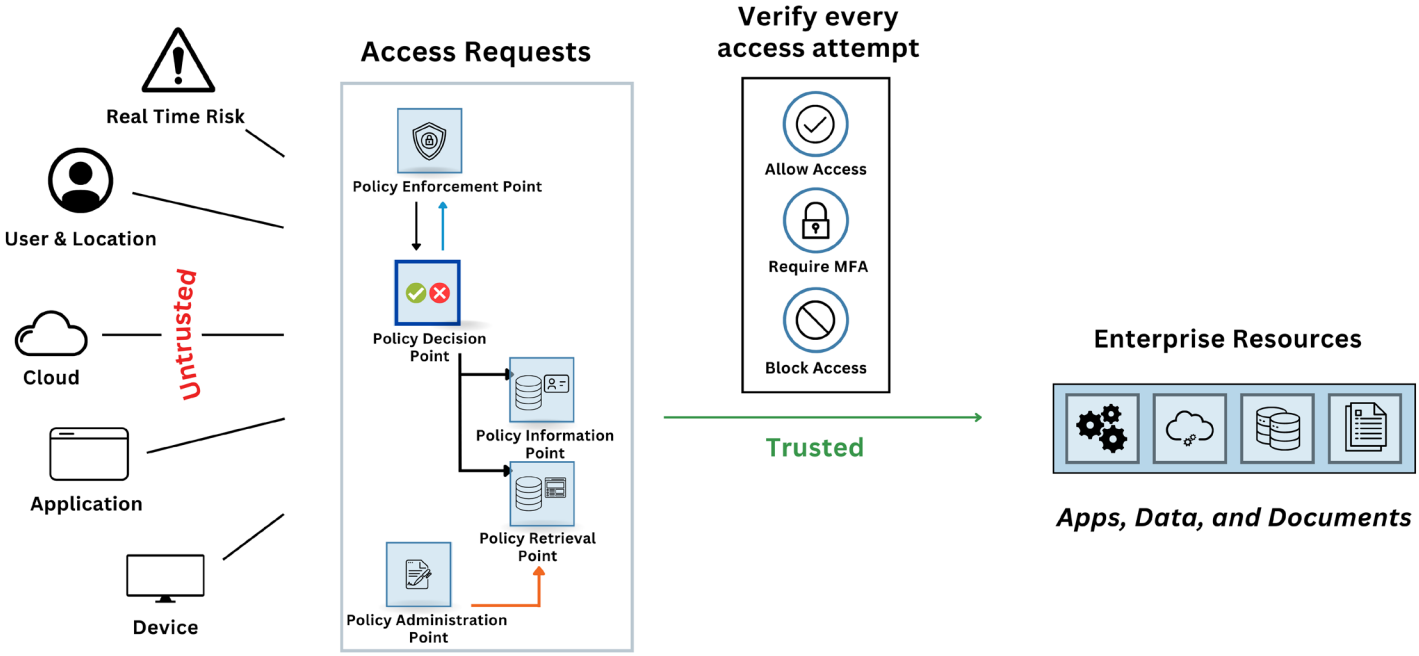
The operations of a zero trust policy engine span from its initial interactions with various inputs to its assessments and action implementations, all in strict adherence to established policies. Each phase in the process, outlined below, plays a crucial role in maintaining swift and reliable policy enforcement.

- Input Acquisition
    - Receives the inputs or events that call for policy evaluation. These inputs can stem from access requests, system events, data updates, that may instigate policy enforcement.
- Gathering Contextual Information
    - Collects contextual data around the input or event, such as subject, resource, environment attributes, and other essential information like user status, data classification, and device.
- Policy Evaluation
    - Evaluates inputs and gathered contextual information against pre-defined policies. Typically involves matching attributes, applying logical operators, and considering rule priorities or precedence.
- Decision-Making
    - Based on the policy evaluation, the policy engine determines an action to be taken. This can involve granting or denying access, triggering automated actions, providing recommendations, or any other response specified in the policies.
- Action Execution
    - If the policy engine decides that an action needs to be taken, it triggers the execution of the specified action(s). This may involve modifying data, sending notifications, invoking external services, enforcing access control, or any other relevant operation
- Audit Log
    - Records the decisions made and actions taken for auditing and compliance purposes. It maintains a log or audit trail that captures the policy evaluation process and the outcomes.
- Policy Updates and Management
    - The policy engine allows for policy updates and management to maintain the policy set efficiently. This includes modifying existing policies, adding new policies, or removing obsolete policies.

## Underlying Architecture

The policy engine is one of the core components of Zero Trust Architecture, which can be summarized as the following:

- Policy Administration Point (PAP)
- Policy Decision Point (PDP)
- Policy Information Point (PIP)
- Policy Retrieval Point (PRP)
- Policy Enforcement Point (PEP)




According to the PCIM (Policy Core Information Model), a collaborative effort by Internet Engineering Task Force (IETF) and Distributed Management Task Force, policies are predefined and stored in a repository point. Upon demand, the policies can be automatically retrieved and used in decision-making, eliminating the need for manual administrative countermeasures on a per-event basis.

Within this framework, as highlighted in NIST SP 800-207, a policy engine is the Policy Decision Point (PDP) - a component that evaluates access requests against defined access control policies to make authorization decisions.

The Policy Administration Point (PAP) manages policies within the policy database, or Policy Retrieval Point (PRP), and the deployment of policies to PDPs.

When a user or device seeks access to a resource, the Policy Enforcement Point (PEP) will form a request based on the subject's attributes, the resource in question, the action, and other relevant information.

Next, the PEP will send the request to the PDP, which will evaluate the request and the applicable policy. The Policy Information Point (PIP), consisting of Attribute Value Providers, gathers the necessary attributes that the policy requires. The PDP considers data from the PRP and the PIP before issuing access decisions.

Then, the decision is returned to the PEP, which enforces the authorization or denial of access to the requester.

## Benefits

From ensuring consistent policy application across distributed environments to enabling granular access controls and simplifying compliance processes, zero trust policy engines are instrumental in bolstering organizational integrity and business agility. Below are the principal benefits of implementing a zero trust policy engine:

1. Distributed Policy Enforcement: Policy engines ensure rapid and consistent evaluation of centrally managed policies which are enforced across applications in a distributed environment.
2. Fine-Grained Access Control: Policy engines enable granular control by dynamically evaluating policies based on multiple attributes and conditions in real time. Authorization decisions can be based on attributes such as user roles, resource classifications, time of day, location, and more. Fine-grained controls enhance security and help enforce the principle of least privilege access.
3. Agility and Manageability: Policy engines facilitate the separation of policy logic from application or system logic, allowing policies to be easily modified without making significant and costly changes to the underlying system or application. This improves manageability and reduces cost of maintenance, allowing organizations to adapt quickly to evolving business, security and compliance requirements.
4. Reusability and Scalability: Policies defined within a policy engine can be reused across different systems or applications, eliminating the need to duplicate policy logic. This reusability simplifies policy management and reduces development effort. Additionally, policy engines can scale to handle large volumes of policy evaluations efficiently, ensuring performance in complex and dynamic environments.
5. Audit and Compliance: Policy engines provide logging capabilities to streamline audit and compliance. They capture all access activity, policy evaluation results, decisions made, and actions taken to create an audit trail. This helps organizations demonstrate compliance with regulations, mitigate risk, and strengthen governance.

## Implementation Challenges and Considerations

When contemplating the implementation of a zero-trust policy engine, it is paramount to address certain intrinsic challenges:

- Clarify the primary requirements of the policy engine in conjunction with the environment and IT landscape in question. For instance, discerning whether the primary aim is to provide fine-grained access control in a cloud system or to secure data in a Product Lifecycle Management (PLM) system.
- Determine the type of data that the organization seeks to protect and how it is used
- Define policies in accordance with the organization’s security and governance needs
- Engage and socialize with key stakeholders throughout the process, to order to provide a mechanism for business input and align with their needs
- Establish a tangible and manageable scope for the implementation process, including a well-defined timeline and resource plan. This also ensures cost-effectiveness without compromising on results.
- Rigorous testing with detailed user acceptance criteria to ensure usability and high adoption rates

# Use Cases

Policy engines have use cases across various industries and domains. Here are some of the common applications of policy engines:

### Use Case: Automating Need-to-know Access in the A&D Industry

The Aerospace and Defense (A&D) industry grapples with stringent data security demands, facing complex regulatory requirements and ever-evolving cyber threats. Zero trust policy engines are pivotal in automating data security without impeding critical access and fortifying both compliance and defense in a high-risk sector.

#### Industry

Aerospace & Defense

#### Objective

Enforce stringent access controls to prevent data leakage and automate compliance with export control regulations, especially ITAR.

#### Scenario

A group of engineers in an aerospace company, consisting of teams across the globe, collaborating with multiple suppliers, is working on a jet design that involves the use of sensitive technology. The company needs to ensure that only those with the necessary clearance and who are located within approved regions (as per ITAR) have access to this data. The solution must operate in a highly secure, private cloud environment.

#### Role of Policy Engine

- Real-time Policy Evaluation:
    - Before any engineer can access sensitive data, the policy engine evaluates contextual attributes such as:
        - » User role (company, supplier) and clearance level
        - » The data's classification level
        - » Geolocation of where the access request is coming from
        - » Device from which the data is being accessed, ensuring it meets security requirements.
- Decision Making and Action Execution:
    - If all contextual attributes align with the policy rules, the engine grants access and the decision is transmitted to the PEP or the application
    - If even one attribute is out of policy bounds, access is denied.
- Audit Logging:
    - All decisions, whether access is granted or denied, are recorded in a tamper-proof log. Metadata, such as timestamp, reason for denial, and contextual attributes, are also logged.
- High Availability & Security
    - Policy vault is built on a High Availability architecture, supporting large scale deployment and uptime requirement while allowing offline policy evaluation, especially useful for disrupted, disconnected, intermittent and low-bandwidth (DDIL) environments
- Policy Orchestration:
    - As regulations or company policies change, administrators can update the policies without impacting system operations. If new threats or vulnerabilities are detected, quick policy modifications can be done to address them.
- Automated Policy Deployment and Policy Testing
    - The policy engine collaborates with distribution and testing features to seamlessly deploy and validate policies. Test plans can be reused across deployments, and optimized policy bundles are automatically distributed, enhancing performance.

### Conclusion

By employing a policy engine, the aerospace company effectively enforces stringent access controls, curtailing data leakage and seamlessly automating compliance with export control regulations. The engine's real-time policy evaluation, action execution, and logging render it a robut solution for achieving the set objectives.

## Use Case: Dynamic Data Masking for Financial Transactions

In the financial sector, companies need to manage data access, sharing and privacy, and client consent, in order to align with regulations such as the Sarbanes-Oxley Act, PCI DSS, and GDPR. Consistent data security is essential in helping firms avoid regulatory breaches and the associated penalties, reputational damage, and loss of customer trust.

### Industry

Financial Services

### Objective

Dynamically obfuscate sensitive data during customer management processes, based on the role, permissions, and context of the user, to ensure data privacy and compliance with regulations.

### Scenario

A financial services institution's customer service division, with customer service centers operating across multiple countries, serving clients across the world. Some representatives need to assist their clients with transaction histories -- while they need to view transaction details, they are not authorized to see the full credit card numbers, CVVs, or other sensitive customer data.

### Role of Policy Engine

- Real-time Policy Evaluation:
    - Before Lydia or any other bank employee accesses customer data, the policy engine evaluates user role and permissions, type of data being requested, and the access location (e.g., in-bank terminal, remote login, country where call center is located, country where the customer is).
- Decision Making
    - Based on the evaluated attributes, the policy engine determines which data fields need masking, and transmits the decision to PEP or application.
    - For instance, the last four digits of a credit card might be shown as "xxxx xxxx xxxx 1234". However, when authorized personnel such as an internal audit employee tries to access, they will be able to see the unmasked version of the data.
- Audit Logging:
    - Every data access and masking decision is logged with details such as user ID, timestamp, data fields accessed, and the masking decision made. This assists in future audits and ensures employees' accountability.
- Reliability & Performance
    - Policy vault is built on a High Availability architecture, supporting large scale deployment with stringent uptime requirements. Policy engine must be robust and always available, while meeting the response time requirements of high volumes of users across the world.
- Policy Orchestration:
    - As regulations (like GDPR, CCPA, GLBA etc.) evolve or the bank's internal privacy policies change, administrators can quickly update masking policies. The engine allows for flexibility to add new data fields or change masking rules for existing fields.

### Conclusion

By implementing a policy engine for data masking in the financial industry, banks and financial institutions can maintain a delicate balance between user accessibility and data privacy. It ensures that sensitive information remains obscured without compromising the efficiency of operations, meeting both privacy and compliance objectives.

## NextLabs' Solution: Unified Policy Platform with Policy Engine

NextLabs' CloudAz is a unified policy platform with real-time enforcement that centralizes administration and employs a zero-trust principle to enforce data-centric security measures and compliance in real time, by automating least privilege access and securing applications and data.

CloudAz's patented, dynamic authorization policy engine uses real-time contextual information to evaluate conditions in policy set to make authorization decision. These conditions are based on user, environment, and resource characteristics ("attributes"), which are evaluated in real-time to determine what permission a user or subject should be granted to applications, APIs / microservices, business transactions, and data.

CloudAz's policy engine is able to account for changes in user status or changes in the resource. For instance, if an employee moves to a different department within the company, no new policy needs to be created since policies are evaluated against the latest set of attributes without the need for manual intervention.

CloudAz can be deployed anywhere, be it on-premises, in private cloud, or as a SaaS. CloudAz runs natively on AWS, Azure, OpenShift and Google Cloud. With support for multiple deployment models, it gives you the freedom to choose the right cloud deployment strategy, whether it is hybrid or multi-cloud. With the ability to create new instances across multiple landscapes – set up development, test, and production environments can be done quickly. Policies can be transported between cloud and on-premise deployments, ensuring consistent policy enforcement across all environments.

## ABOUT NEXTLABS

NextLabs®, Inc. provides data-centric security software to protect business critical data and applications. Our patented dynamic authorization technology and industry leading attribute-based policy platform helps enterprises identify and protect sensitive data, monitor and control access to the data, and prevent regulatory violations – whether in the cloud or on premises. The software automates enforcement of security controls and compliance policies to enable secure information sharing across the extended enterprise. NextLabs has some of the largest global enterprises as customers and has strategic relationships with industry leaders such as SAP, Siemens, Microsoft, and IBM. For more information on NextLabs, please visit http://www. nextlabs.com.