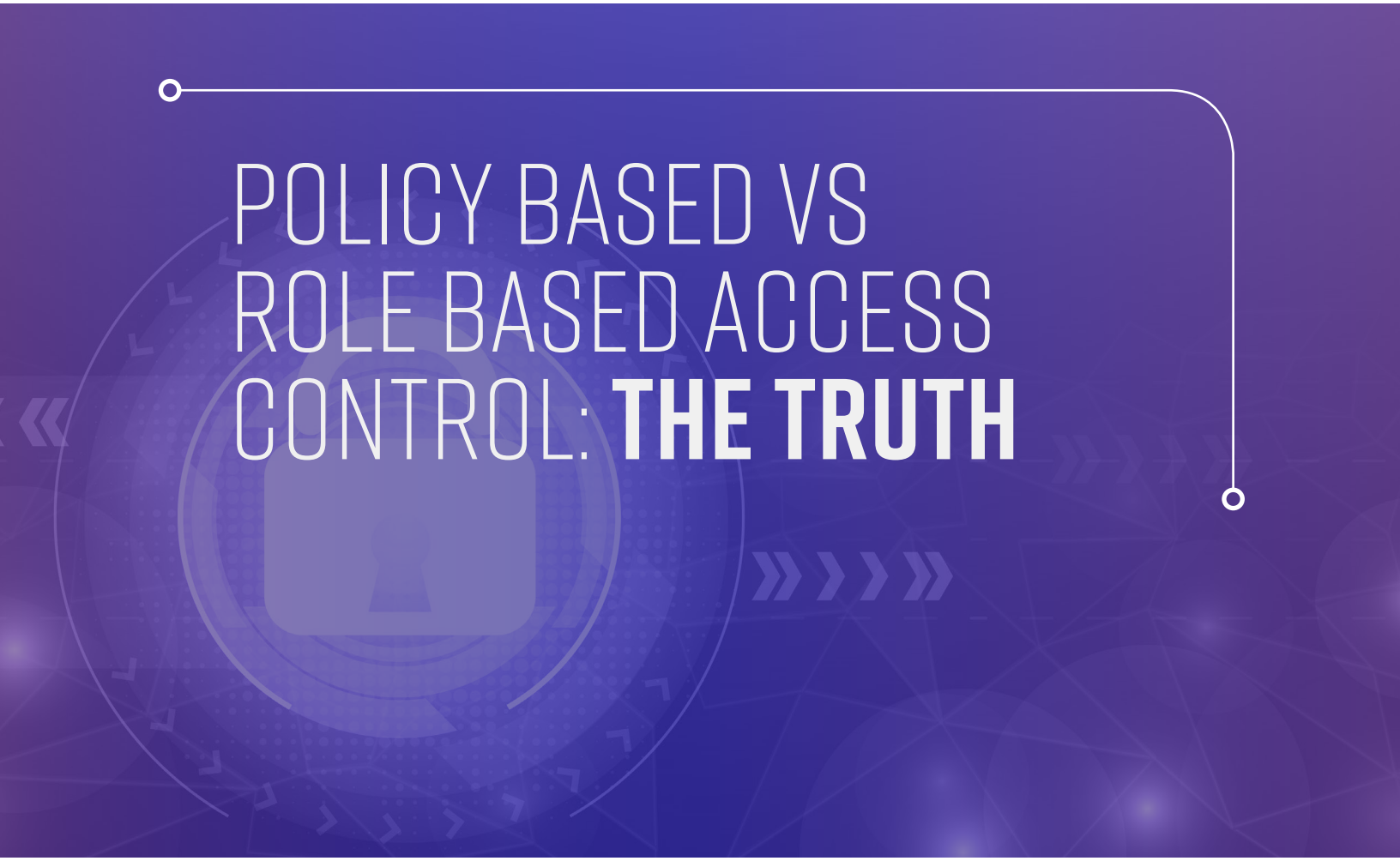
plainID
THE AUTHORIZATION COMPANY

POLICY BASED VS ROLE BASED ACCESS CONTROL: **THE TRUTH**



WWW.PLAINID.COM

Policy Based vs Role Based Access Control: **The Truth**

Identity and access management (IAM) is typically responsible of the identity life cycle management within various systems the user needs to access, including on-boarding, off-boarding and role changes. But, even though IAM solutions have been in the market for more than 30 years, they are considered to be extremely complex and highly time and resource consuming.

Adding to that is the ongoing expansion of the organization, both with its data and identities. Many organizations are considering or have already expanded their data to the cloud. What was once controlled and secured by the internal measures of the organization is now distributed and so is the control of its access.

Many organizations are also required to support distributed identities. The organizational systems are accessed by the employees from anywhere, office, home or mobile. Mergers, external contractors, extend even more the sources of identities those systems are required to support.

PlainID provides a simple and intuitive way for fast-paced organizations to create and manage their Authorization policies

The IAM Building Blocks

The IAM building blocks can be broken down to the following three categories:

- Identity – How is the online experience defined and managed?
- Authentication (AuthN) – How can the identity be proven?
- Authorization (AuthZ) – What can the identity do?

Existing technologies focus primarily on the first two, Identity life cycle management and Authentication. Authorization is usually left to the responsibility of the developer, the application owner. The result is no real control, and no visibility into what the user can do or see.

It's time for a change, it's time for a better way to handle Authorization.

The developments in the Authorization approach are aiming to simplify the Authorization process, enable it to scale faster and provide much better control and visibility to the organization of its assets.

info@plainid.com www.plainid.com

The Authorization Company

Policy Based vs Role Based Access Control: **The Truth**

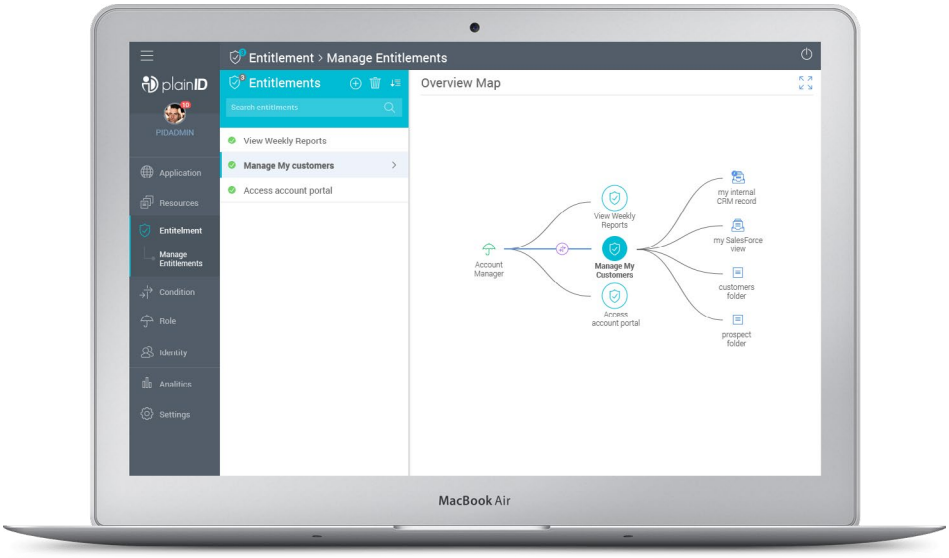
From RBAC to ABAC to PBAC

Assigning access controls is one of the foundational steps in information security and compliance. Enterprises commonly deal with data—personal, commercial, business internal —that absolutely cannot leak to the public at large nor internally within the organization. All the major compliance regimes, such as FISMA, PCI-DSS, HIPAA, and GDPR, have detailed requirements that lay out who can access certain data, when that data can be accessed, and how to keep records of that access. One last question remains, however—how do you ensure that no unauthorized persons can access that data?

Modulate Access Using Attributes

Under ABAC, access to a particular record or resource is moderated based on certain traits—attributes—of the person accessing the file (the subject), the resource itself (the object), and the time and place where the object is being accessed (the environment). Attributes of the subject may include their title, certifications, or training. Attributes of the object may include its related project, the personally identifying information (PII) it contains, and the sensitivity of that PII. Viewed holistically, these attributes can now be used to set the rules that would enable access to data and resources.

RBAC (Role-Based Access Control) on the other hand, involves creating a role for every organizational or business functionality, giving that role permission to access certain records or resources and assigning a user to the role. This system is entirely too granular, not flexible, and very limited in large scales.



info@plainid.com
www.plainid.com

The Authorization Company

Policy Based vs Role Based Access Control: **The Truth**

Simplify Access Controls and Smart up your Permissions with PBAC

Under Policy Based Access Control (PBAC), Authorization isn't reliant on any specific implementation (like XACML) and policies can be set in natural language. For example, a team lead can grant access to a project only to team members, on weekdays, between the hours of 9-5. This makes it much simpler to manage large amounts of users and data.

Additionally, PBAC supports environmental controls. So if there are sensitive files that should only be viewed on certain corporate computers, a policy can easily be set to limit access to on premises systems. The policy can also be quickly adjusted based on events. For example, if engineers need access in the event of an emergency, they can be granted immediate access for a limited amount of time, during the emergency.

PBAC also makes compliance with regulations such as GRC and GDPR easier by segregating data access, controlling access creep, and even preventing authorized users from accessing data in a risky manner.

Automation to Cope with Constant Challenges

Change of role, organizational structure, assignment to new projects – PBAC adapts immediately to what documents, medical records, servers and more the user can access. No additional action is required to enable access to the new project data, the new user in the department, or a freshly assigned accountant.

PBAC truly is a more efficient approach to support your access decisions.

> Policy-based Access Control (PBAC) is an emerging model that seeks to help enterprises address the need to implement actionable access control schemes based on corporate policy and governance requirements.
>
> - KuppingerCole

Policy Based vs Role Based Access Control: **The Truth**

How Do You Authorize:

The Old Static Way or the New Dynamic Way?

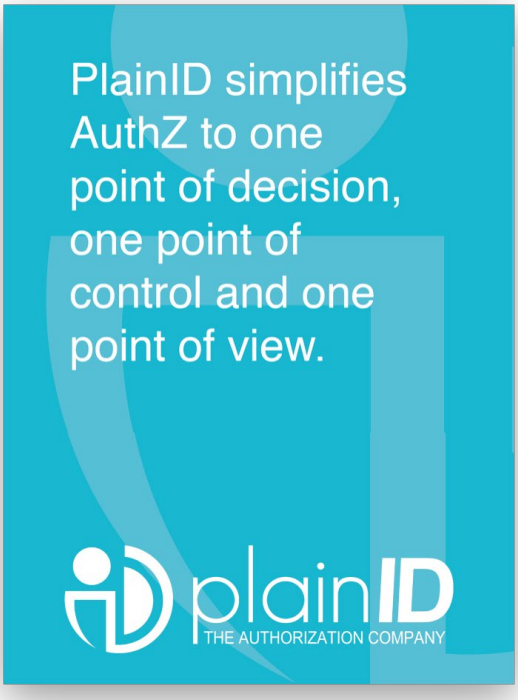
Authorization is the core of organizational security—which also means that it's central to productivity, Authorization determines what a digital identity can do within each and every application.

Employees need access to data and applications anytime and while on the go, but for security reasons, they can't have that access at all times. So, when should that decision be made? When is the right time to decide if access is permitted or denied?

Traditional Authorization – Once upon a time...

It's common for Authorization decisions to be triggered by HR movements, for example when on-boarding a new employee, changing an employee's role or terminating employment.

When a new employee joins an organization, they are likely to receive the same Authorization as a co-worker, gain additional permissions over time, and that would be the end of it. Authorizations aren't changed or even reviewed as more factors change.



Is that Enough? Of Course it's not Enough

The expectations of core security within organizations today are higher, they need to be smarter and to be able to take more factors into account, such as where the employee is accessing from, whether their role has changed, or if the parameters of the project have changed. A definition that was made a year ago, or even an hour ago may no longer be relevant. When gaining access to the most valuable assets of the organization, the decision needs to be made now!

Classic Authorization methods have limited functionality. They rely on repository-defined groups or roles that provide a link between users and resources. Those Authorization decisions are preconfigured and cannot change in real time. Giving a user Authorization to view and use a certain suite of files and applications means that unless the administrator manually revokes Authorization, the user will be able to use them forever.

info@plainid.com
www.plainid.com

The Authorization Company

Policy Based vs Role Based Access Control: **The Truth**

Smarten-up your Authorization

Dynamic Authorization represents access decisions that are made in real time. When the user requests access to data, tries to manipulate data, or attempts to use an application, Dynamic Authorization can consider what factors make those actions a potential risk. For example, the time of access, the location, the worker's employment or vacation status, and the security status of the system are all deterministic factors that delineate a user's ability to take certain actions.

What could this look like? Access to modify research data is permitted for authorized users until that data is approved; after approval, the data can only be viewed. Or, let's say that normally, access to IT resources in a production environment is limited, but, upon an incident that requires specific attention, access is approved for the authorized user. Dynamic Authorization can be used to enable access, to prevent access based on what happens in real time, who the user is, and where they are accessing from. It lets you make smart decisions on how data and resources will be available to be used.

Dynamic Authorization: Connect with the 'here and now' to protect your assets

The Unsudden Death of Group-based Access Control

It's true that traditionally, security groups are the most common way used to authorize access to data and actions. But what is truly their role? How do they operate? Security groups, or security roles act as a link, an intermediate between the users and the data, information or actions. They do not directly reflect what a user can actually do or see.

Groups Are Old School

For example: we want developers to access development data, and managers to access management data. So we create one group for developers and another for managers, and assign the users accordingly. But who is responsible for ensuring that the development group can only access development documents? And that the management group can access only management data? What happens if a development manager, who is assigned to both groups, accidentally places an "Employee Management Assessment" document in the development folder?

info@plainid.com www.plainid.com

The Authorization Company

Policy Based vs Role Based Access Control: **The Truth**

Diverted Responsibility

When using group based access, the responsibility is diverted. The IAM team usually connects users to groups, but the data and activities this group can access is under the responsibility of the application or the business owner. In practice, users often receive access to too many resources that they do not need and are prevented from receiving access to specific resources and tools that they do need.

This Approach is Not Good Enough

This approach leads to too many errors and does not scale.

Reverting to the developer example: Whenever that developer joins a new project, they need to be assigned to a new set of documents, tools, and permissions. In addition, access to their previous set of tools etc., will probably need to be revoked. This may be manageable if there are only a few developers in the organization, but what happens in places with hundreds or thousands of employees? The IAM team will spend a large and unnecessary amount of time unpacking and reassigning permissions even if only a single employee changes roles.

A Better Way: Connecting Users Directly to Data

Resource-based access allows you to connect users to the data they are entitled to access. Without a middle man, without giving away responsibility, without any unknown stages which could lead to potential errors.

For example, in resource-based access, access can be granted based on the matching project identifier for both user and document. It can also be based on the user's role in the project, with access determined according to project stage – Project A is in review stage, so its data is accessible to all reviewers that are assigned to this project.

Resource-based access enables direct connection and accurate visibility to what a user can see and do.

Empty the Repository: Why Virtual Tokens are Better for Authorization

If you're using a business application, it is very likely to have a user repository attached. This is usually a simple database containing an ID and a list of authorized actions for each user. It's a simple system, that requires IT resources to be managed.

Several solutions have been tried, with LDAP (Lightweight Directory Access Protocol) as the most popular. This is, in effect, is a single directory designed to share user and Authorization information between many applications. Its advantages are that it is an industry standard designed so that every developer can freely integrate it into their product. The drawback, however, is that it didn't fit all Authorization needs and so wasn't widely adopted.

Policy Based vs Role Based Access Control: **The Truth**

The Problems with Repositories Mimic those of Static Authorization

In addition to the problem of volume, repositories have drawbacks common with other traditional forms of Authorization.

- Administration: In order to change permissions for a given application, the repository needs to be updated. Either manually or by a provisioning system, in both cases it's a complicated task that requires time and resources.
- No Flexibility: Authorizations don't change based on any variables. For example, a cyber security event, or user login through a mobile device, won't remove any assigned permissions. Repositories are static, however, and their users & permissions must be programmed in advance.
- Inefficient Distribution: With over 500 repositories in the average enterprise, the problem isn't just a matter of scale. It is difficult to apply Authorization policy consistently over such a large volume of databases. If the Authorization policy isn't applied consistently – whether due to accident or indifference, then certain applications may become security risks.

> *"77% of enterprises have at least one application or a portion of their enterprise computing infrastructure in the cloud."*
>
> *https://www.forbes.com/sites/louiscolumbus/2018/08/30/state-of-enterprise-cloud-computing-2018/#2d66fc10265e*

Virtual Tokens Provide the Answer

Virtual tokens take one of the traditional aspects of Authorization and flips it on its head. What if, instead of storing Authorization information in large repositories within each application, you instead reduce it to a small repository fitted for an individual user? This is what a virtual token represents. Upon access, this token is sent to the application, which responds accordingly.

This approach displays some marked advantages over the traditional repository approach. For one, it's responsive -the data carried by the virtual token allows the application to respond dynamically based on conditions described by the Authorization token. Secondly, virtual tokens are allowed to be small, containing only the information that's necessary for the app to authenticate and authorize the user.

Say NO to Provisioning

Lastly, virtual tokens reduce the need to maintain all those repositories, so no more unmanaged Authorization, no more "ghost" IDs.

info@plainid.com www.plainid.com

The Authorization Company

Policy Based vs Role Based Access Control: **The Truth**

What PlainID Does

PlainID provides both Business AND Admin teams with a simple and intuitive means to control their organization's entire authorization process.The platform allows you to implement literally any kind of rules you could imagine, all without coding, and all in fine grained detail. PlainID simplifies Authorization so that thousands of Roles, Attributes and even Environmental Factors can be converted into a few logical SmartAuthorization policies using our Graph Database Decision Engine.

No Plain... No Gain

REQUEST YOUR DEMO

The rapid demands of business require a clear path to the tools and apps that enable growth. We don't put up roadblocks; our solution is not messy or hard to understand.

PlainID quickens new implementations by meshing emerging Authorization standards with existing technologies. On premise, in the cloud or mobile based implementations are supported.

Our PBAC approach gives business leaders unprecedented control over their Identity and Access Management.

www.PlainID.com

The Authorization Company