




5 MYTHS ABOUT

POLICY-BASED ACCESS CONTROL

A PlainID Publication

An abstract graphic of a network or web structure, composed of numerous small white dots connected by thin, light blue lines. The dots are scattered across the dark blue background, with a higher density of connections in the lower right quadrant, forming a complex, interconnected mesh. The lines vary in length and orientation, creating a sense of dynamic connectivity.

IAM needs have significantly changed over the last 5 years. In just a short time, we've gone from enclosed networks, accessed by employees only, to a cloud that is so open that [Gerry Grealish](#) of Symantec writes, "the web [itself] has become the biggest threat vector that companies now face."

Granted, the cloud isn't just a threat; it makes many B2B services possible. These services also need Authorization solutions, especially to support [delegated administration](#). The best solution is Policy-Based Access Control (PBAC), which supports fine-grained Authorization based on policy statements written in natural language. This is an important set of properties which point to both the great flexibility PBAC offers as well as how easy it is to use when implemented correctly. PBAC also makes each Authorization decision in real-time, not at login, meaning that a company can implement a new policy and have it take effect immediately. PBAC is the most recent of Authorization approaches, and is starting to see a surge of interest from both the B2B and B2B sectors, as well as B2E, which leverages the delegated administration capability to offer partners or downstream businesses the flexibility and security that PBAC offers.



MYTH #1

RBAC IS ENOUGH

Role-Based Access Control (RBAC) has been the most popular Authorization solution since the 1990s. It's easy to understand: you create roles with specific permissions, create system users, and then assign user roles. The permissions can vary widely in scope from one that allows one to open an application to another one that controls a specific field in an application. For example, a company with a call center might have two roles, a regular service rep and a supervisor. The permission sets for each could then specify that both could log on to the call center application, but the regular worker could only view a customer's balance, whereas the supervisor could change it.

Although RBAC was an important step forward, replacing access lists, it simply isn't powerful or flexible enough to meet today's needs. First, it doesn't support fine-grained Authorization, meaning it cannot vary permissions by attributes such as time of request. RBAC makes its Authorization decisions at login, and therefore cannot support mid-session changes. PBAC, however, decides at the time of each request, supporting immediate Authorization changes. This allows for dynamic policies to be the norm in an enterprise, changing a person's access in real time, as needed.

Additionally, RBAC is cumbersome to maintain, requiring roles to be changed whenever an asset is added to a system, [making scalability a serious issue](#). At the same time, once an IT person creates one role, which may contain dozens of permissions, companies often then request a role or two that differs from the original by only one or two permissions. If this phenomenon is repeated too often, the company suffers from the "role explosion" phenomenon with dozens of very similar roles, all of which need maintenance. Finally, if users have more than one role, "access creep" can occur as people retain access rights they no longer need as they switch jobs. PBAC avoids all these problems by supporting the creation of policy statements that can include more than one role. This eliminates both "role explosion" and "access creep," while also reducing maintenance.



MYTH #2

PBAC AND ABAC ARE THE SAME THING

Although PBAC and Attribute-Based Access Control (ABAC) do have some things in common, they also have important differences, which makes PBAC more flexible and user-friendly.

ABAC was the first major attempt to go beyond RBAC. It eliminated roles, replacing them with attributes, such as time of day, as part of the inputs considered in Authorization decisions. The language eXtensible Access Control Markup Language ([XACML](#)) was developed for ABAC to support the creation of logical rules that apply Boolean logic to these attributes; later PBAC began to also extend support beyond XACML, such as supporting JSON. By making these changes, ABAC supported fine-grained Authorization, just like PBAC. But that's pretty much where the similarities end.

ABAC platforms depend on rules written in computer code, often using XAMCL or JSON combined with Boolean logic. Although PBAC can make use of policies written in these languages, one of PBAC's main advantages is that it supports writing rules in natural language, making it much easier to use. This is particularly important when an organization [scales up](#): the rule "Sales managers can update their project data" can be applied as easily in companies with 5, 50, or 500 sales managers. All in all, PBAC is more flexible and easier to use than ABAC.



MYTH #3

DEVELOPERS SHOULD BUILD POLICIES, NOT BUSINESS LEADERS

Much as we appreciate developers, it's bad practice for management to assign policy development to IT in every scenario. **There are two main issues - delaying of crucial development across the enterprise and exposing of internal resources. Developers are great at building and implementing services, but asking them to create business policies leads to delays in delivering crucial enterprise needs, even when PBAC is being used.** If the company is using RBAC or ABAC, the problem is compounded by the complexity of the languages used. **Developers also tend to externalize data as their attention shifts to new projects so that others within the organization can consume their work, leading to internal assets being used externally.** As Synopsys's [Gary McGraw](#), notes, resources "are set up for 'internal use' and then over time start being used 'externally' as well."

And that's the crux of the matter. Authorization decisions are business decisions; they must fit business logic, rather than being confined and complicated by code. Your company's data is its most important asset, aside from its people, thus [access control must be a management decision](#). With this principle in mind, it's clear that you need an Authorization solution that enables management to create clear policies. Only PBAC solutions that offer an easy to use User Interface that brings to life the natural language capabilities of PBAC provide these features. In a best case scenario, IT will implement the PBAC solution for the business, and train management on how to administer the PBAC solution, needing IT only for standard help requests.



MYTH #4

A “HOME-MADE” IAM SOLUTION IS BETTER THAN A VENDOR’S SOLUTION

It is tempting to ask your development team to build your company’s IAM solution. After all, no one knows your company’s security needs, assets, and network structure better. While no one knows your business better than you, this approach is similar to reinventing the wheel or SQL. There are already working PBAC IAM solutions on the market that:

- Know how to [take any type of data and turn it into a usable attribute](#)
- Support predefined or configured data sources, and enable flexible mapping of identities and Authorization data
- Support interaction with cloud, mobile, and legacy applications
- Are technology-agnostic, supporting existing RBAC and ABAC solutions and have no problem adding future IAM solutions
- Include a GUI that uses [graph technology](#) that makes it easy to see and work with the following aspects of policy creation:
 - **Who:** The user identities
 - **What:** The object or resource being accessed
 - **When:** The conditions required by the user identities to access the object or resource
 - **How:** How is the object or resource being accessed

By choosing a solution from an experienced IAM vendor, you not only gain from their experience, you can direct your resources to letting your business excel at what it does best. You also leverage the vendor’s implementation experience across all of their customers, which is translated into new features that constantly make the solution better, as opposed to a home-grown solution becoming stale, lagging behind new technology and ideas.



MYTH #5

FINE-GRAINED PBAC AUTHORIZATION DOESN'T HAVE THE ROI TO MAKE IT WORTHWHILE

Finally, what about ROI? Can the cost of switching to a dynamic, fine-grained PBAC solution be justified? Certainly. First, PBAC provides the best IAM solution, supplying the strongest and most flexible access control in the market. Avoiding breaches with their direct and indirect costs should already settle the question.

Even when considering actual costs, PBAC is the winner because it avoids the development and maintenance costs of both RBAC and ABAC. In a recent [case study](#), a company that switched from RBAC to PBAC went from having 1000 separate roles to only 50. The savings in maintenance costs as well as the reduction in development time were profound.

Fine-grained PBAC solutions offer more than eliminating the drawbacks of RBAC and ABAC. It supports [B2B delegation and use of a company portal](#), services that are growing more important as businesses make greater use of the cloud. In a second [case study](#), the same leading PBAC platform enabled a pharmaceutical company with a set of partners/resellers to:

- Manage the partners/resellers
- Allow partners/resellers controlled access to assets (products)
- Allow partners/resellers to manage their employees
- Audit, investigate, and analyze all data related to authorization

Additionally, today's PBAC solutions are designed with regulatory [compliance](#) in mind, saving companies both money and worry. All told, PBAC provides strong ROI.

Of course, the biggest ROI of all is the insurance policy that PBAC provides, allowing the Boardroom of a company to rest easy knowing that the Identity Management is under control, leaving fewer scenarios for major data breach, which can be fiscally catastrophic for a company.

PBAC: Stronger Than The Myths

Fine-grained PBAC Authorization solutions are better than other IAM approaches. PBAC provides greater flexibility, reduced maintenance, and more precise control than RBAC, avoiding “role explosion” and “access creep.”

It is simpler to use than ABAC, supporting the creation of policies without writing code, while finding a middle ground between RBAC’s dependence on roles and ABAC’s avoidance of them.

In addition, PBAC’s support of natural language and graph technology makes it possible for business leaders to make and implement crucial Authorization decisions, rather than leaving them to developers. PBAC offers significant ROI, so using an established vendor’s platform is more cost-effective than creating an in-house one. All in all, PBAC is the best IAM solution.

INTERESTED IN LEARNING MORE? SCHEDULE
A DEMO WITH THE PLAINID TEAM.

[REQUEST A DEMO](#)