



# PBAC Platform


## Retail Banking Use Case

## Introduction:

Banks handle sensitive data for a large set of clients. They operate in a highly regulated industry and need to comply with both external and internal compliance frameworks (GDPR, PSD2, AML, ISO27000, NIST...)

Banks must know exactly who has what access, but the reality is that this is a major gap in most banks' security architectures.

Permissions and authorizations are often managed in siloed IT platforms and in applications that might provision and even enforce access control, but the siloed nature makes this approach cumbersome, error prone and inefficient. In order for a bank manager to know that a bank teller has the correct access to a certain customer portal, for example, they likely need to ask the IT department, who would need to research the specific request. Many companies attempt to subvert the inefficiency by simply granting broad brush strokes of access to employees, which creates **unprecedented security risk**. Errors that stem from manual provisioning only add to this risk. Finally, the elephant in the room is the time and money being spent by IT to administer these siloed platforms. Often companies have multi-million dollar maintenance contracts that could be replaced by the capabilities of PlainID PBAC Platform.



# Using PlainID for policy based access control (PBAC):

PBAC offers a centralized approach to streamline secure business processes and to simplify back-office and IT permission management processes. This approach ensures that the right users have access to the right data at the right time without the hold up of slow, sometimes even manual, internal processes.

## The retail banking use case:

In Retail Banking there are a multitude of business rules that could govern how bank client data can and should be accessed. These can relate to basic client profile data such as home address, type of relationship with the bank and active client contracts.

The client's financial situation such as bank accounts, bank cards and bank transactions typically have specific rules for access. This type of data is what we can consider basic bank data that a Branch Manager or Branch Clerk might need to access in their daily business activities. There is also other data that needs even more restrictions such as a Client Risk Score where the bank has more detailed access rules.

## A policy example in natural language:

***"Branch Managers and Bank clerks can access the Client Basic Profile, Bank Accounts and Card Data of clients that belong to the same Line of Business (LoB) and same Branch as themselves."***



## How the basic policy looks in PlainID:

Note that there are no references to specific LoB's or Branches in the policy. Neither are there any references to specific Bank Client Profiles, Bank Accounts or Bank Cards. The PlainID visual representation of the policy allows for a simplified and understandable view of who can access which data, down to the fine-grained layer. It also shows under which circumstances and what reasons the users are able to access the data. You can also see that one policy can govern the access control through multiple applications. In the picture to the right two different applications, the Internal Bank Portal and the Call Center Application, use the same policy ensuring the same user experience across multiple applications.

This example policy demonstrates the efficiencies and flexible access control that PBAC offers over a traditional Role Based Access Control (RBAC) model. Implementing an RBAC structure to support the policy requirements above would be nearly impossible or at least force the bank to implement hundreds or even thousands of Roles causing a massive administrative burden.

# Supporting more advanced policy requirements:



Even though the Basic Policy above is very effective in segregating access to a Bank Client's data and thereby supporting basic privacy requirements aligning with the "need-to-know" principle, there are far more sophisticated privacy rules that must be incorporated. Below we have outlined some of these rules for our use case.

## Restrictive access policies:

- Not access any Client Data where the client is a "VIP" or a "PEP" (Politically Engaged Person)
- Not access any of my colleague's Client Data
- Not access my own Client data
- Not access any of my close relative's Client Data





# The authorization decision:

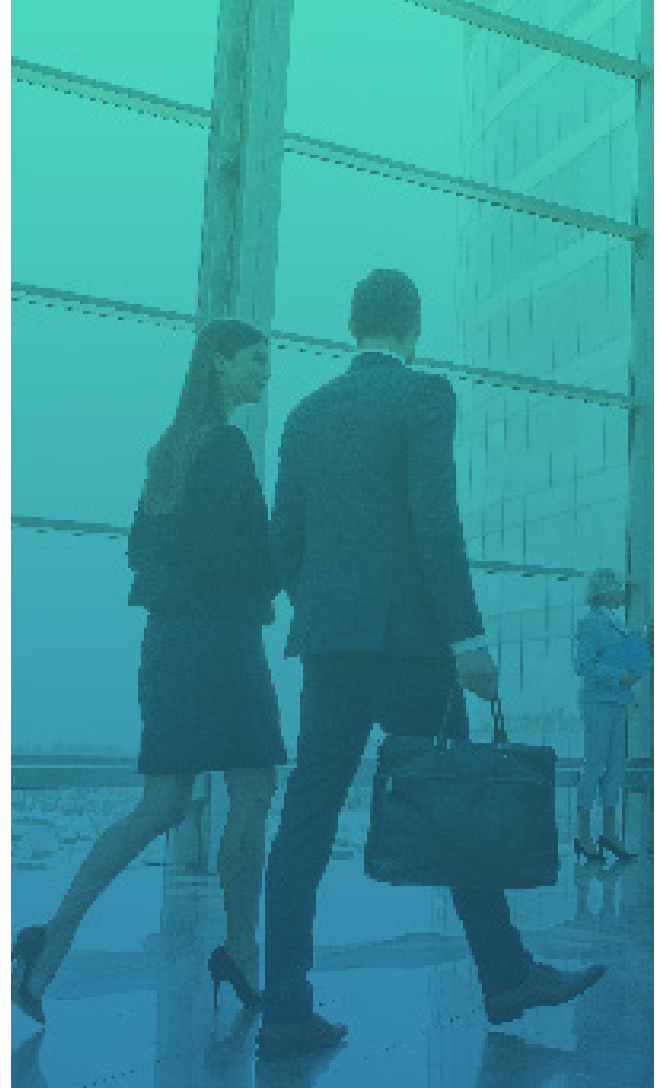
At run-time when the Branch Manager tries to access a Client Profile through a Bank Application, the application sends a simple request to the PlainID authorization rule engine.

In order for the rule engine to be able to provide an "informed decision" the authorization service needs to access the underlying bank data. This includes the identity data and the asset data, but also the referential data, meaning the relationships between relatives, block listed bankers and VIPs/PEPs. This data could be retrieved from a variety of systems including SQL, LDAP, SCIM, REST API's etc.

The rule engine evaluates the current user's (in this use case, the Branch Manager) role and also LoB and Branch to compare this data with the current Client Profile data to be able to reach an access decision for the basic policy.

In addition, the rule engine also evaluates the restrictive rules that are part of the advanced restricted policies e.g. the Branch Manager's colleagues, relatives' potential blockings.

Of course, we can also use the policy decision to enable appropriate access. This approach can support access based on a "white list" or, for example, if the banker is assigned as a "Personal Banker" to a client or a set of clients.



## The enforcement of access:

After evaluating all the granting access policies and the restrictive policies, the rule engine can provide an authorization decision and/or the information that Bank Application needs to enforce the correct access for the Branch Manager/Clerk.

The application may use this information in-application for many purposes. Maybe it is used to provide a decision if the user needs to have access to a functional tab in the application (coarse-grained resource) or maybe the access should provide more fine-grained access due to a specific Client Risk Score.



## The business benefit and value:

By unifying and externalizing the authorization logic from the Banking Application, we can now start maintaining the life-cycle of the policies in a separate process from maintaining the application logic.

The benefit of having two processes is that they now can change independently from each other. This is often very important for retail banks that struggle with supporting the ever-changing compliance frameworks, both internal and external.

The visibility offered through the PlainID solution ensures that no one is able to access beyond what they should, and that the business knows at all times who has access to what.

The financial benefits of PBAC include a cost reduction in application development and application maintenance efforts but also in the user management processes where PBAC provides efficiency and clarity to the business. In the longer term, significant financial and strategic benefits relate to the efforts of being able to stay compliant with regulations over time, and to be able to continuously onboard new technologies in a secure and efficient fashion.

Furthermore, the compliance control gains can decrease data breaches as well as the number of records affected by each breach. The accounting of access decisions and their enforcement is complex and voluminous. Adoption of PBAC forces an organisation to standardize the dimensions of the authorization landscape to identify and define users, roles and resources (data, applications, other). With standardized taxonomies on which to pivot access control policies the accrued benefits include not only simplification, but greater transparency around who is accessing what, when, how and why. This makes control attestation for effectiveness a much easier process in order to self-assure compliance and audit teams that information control compliance requirements are being met.