WHITEPAPER

# SECURITY INFORMATION GOVERNANCE

## */BAC RE-INVENTED

HELISOFT

INFORMATION MANAGEMENT

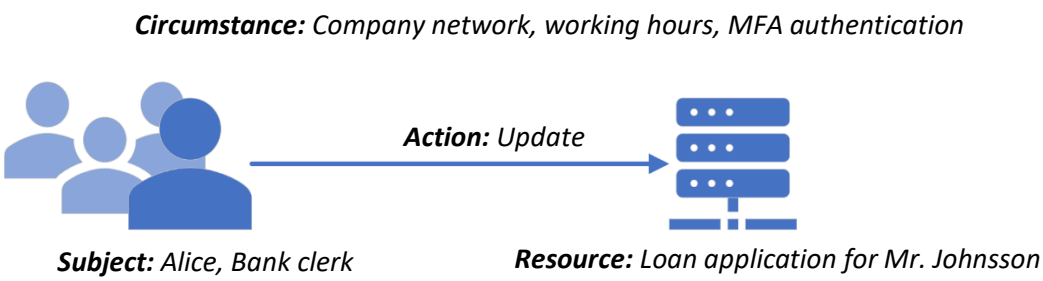
# A holistic perspective on */BAC

## Executive summary

*/BAC refers to the domain of attribute-based access control that enables organizations to meet a broad set of regulatory requirements using an access control where authorization is based on information dynamically evaluated at runtime.

Before the */BAC revolution, organizations statically assigned permissions and entitlements to users and stored them in a common, central catalogue, governance over the security information (who has access to what) was much easier as it was converged into a single attribute store, e.g., Microsoft Active Directory.

With */BAC, access rules deciding who has access to what and under what circumstances, is no longer pre-defined in a static assignment, such as adding a user to a security group. Instead, the access rules, or policies if you like, are based on attributes in different perspectives, typically "***Subject***", "***Resource***", "***Action***" and "***Circumstance***".

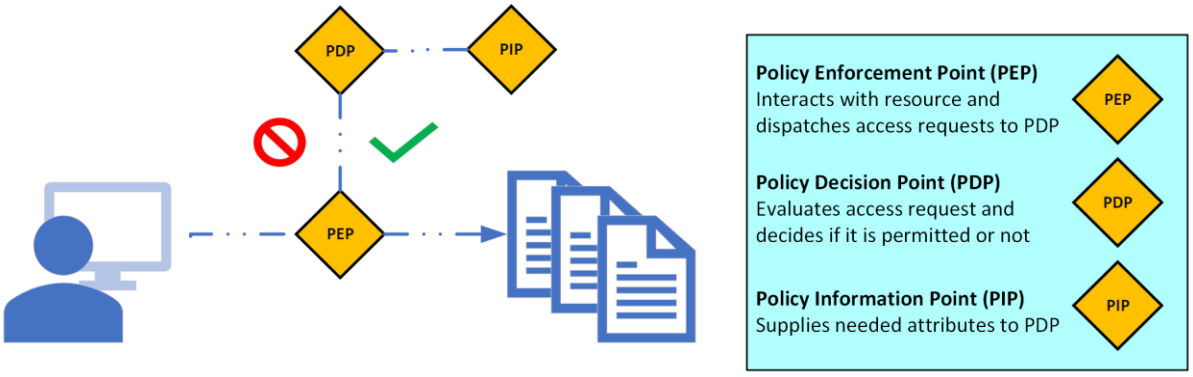



As this simple model get more sophisticated, maybe by levering complex and synthetic attributes, it might counteract the compliance goal of the implementation due to heterogeneous, or in worst case, complete lack of, a common attribute governance model.

This paper discusses the changing requirements and suggests that the necessary */BAC evolution needs to be complemented with governance model effectively implemented in a modern architecture allowing organizations to fully leverage the capabilities delivered by */BAC.

## */BAC implementation

The typical implementation model for */BAC is based on the scheme set forward by OASIS, almost twenty years ago, called eXtensible Access Control Markup Language, XACML.




The model is still valid, though vendors over the years have a found the need to make their own ways of implementing the model.

The implementation model doesn't really matter when it comes to the governance perspective, the attributes could be collected during end-user authentication by an Identity provider (IdP) or they can be fetched from the PDP when evaluating an access request at runtime.

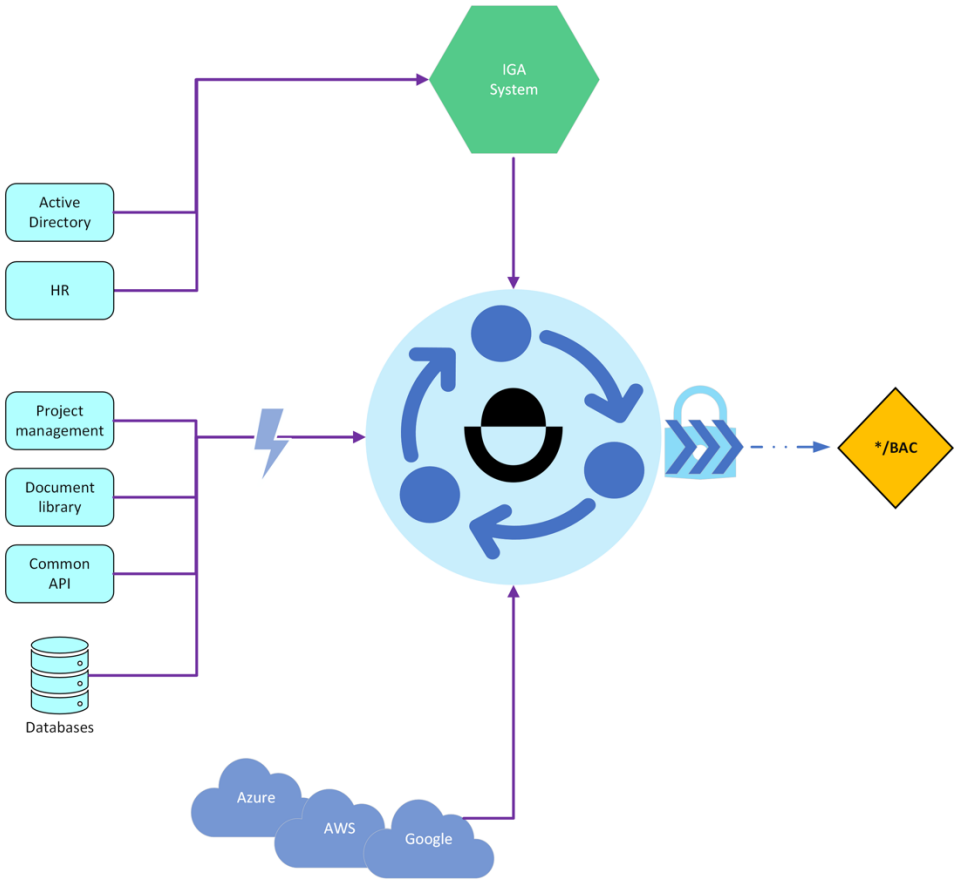
## Changing requirements

Working in the domain of */BAC certainly has some challenges, especially for organizations about to endeavor the implementation journey, where one of the first questions usually is attribute governance and */BAC vendors today solely focus access control leaving customers with only part of the compliance solution they most probably are looking for.

As the challenges with data normalization and cleaning yet alone are enough to derail any implementation, going forward with synthetic/virtual attributes in many cases needed for a full proof access solution will be impassable for most.

## Completing the puzzle

The solution to effectively solve this Gordian knot is to implement a normalization point, effectively acting as a Policy Information Point (PIP) to any kind of */BAC implementation. Many organizations have successfully implemented a solution for Identity Governance & Administration (IGA), providing governance over entitlements and identity information this isn't enough for */BAC as it, in most cases, only covers the "***Subject***" perspective of */BAC.




With the Helios™ platform from any organization can easily implement a normalization point taking the performance and security bottlenecks from the production systems, effectively creating a super PIP with capabilities such as approval flows on attribute change, distributed cache, arbitrary models from any source and dynamic APIs on every model that's created.

Helios™ is an event driven, container-based platform, built on a microservice architecture enabling both horizontal and vertical scaling, avoiding performance bottlenecks.

With a scalable graph database and our user-friendly dashboards, that are seamlessly updated as soon an object gets updated thru our message-driven data ingestion, users always stay in control.

## Conclusion

By leveraging the capabilities of Security Information Governance, organizations can collect the highly sought-after capabilities of */BAC. In most organizations implementing */BAC, the attributes is usually scattered over the IT-landscape with various quality due to the fact that governance is applied atomistic, at best.

By using a normalization point with capabilities allowing organization to build their own security information model and apply governance, normalization and data cleaning over it at the same time, the rough edges of an */BAC implementation is suddenly not so rough anymore.

When holistic governance is applied those security related bits and pieces used by */BAC, organization can truly embrace the paradigm shift of access control.