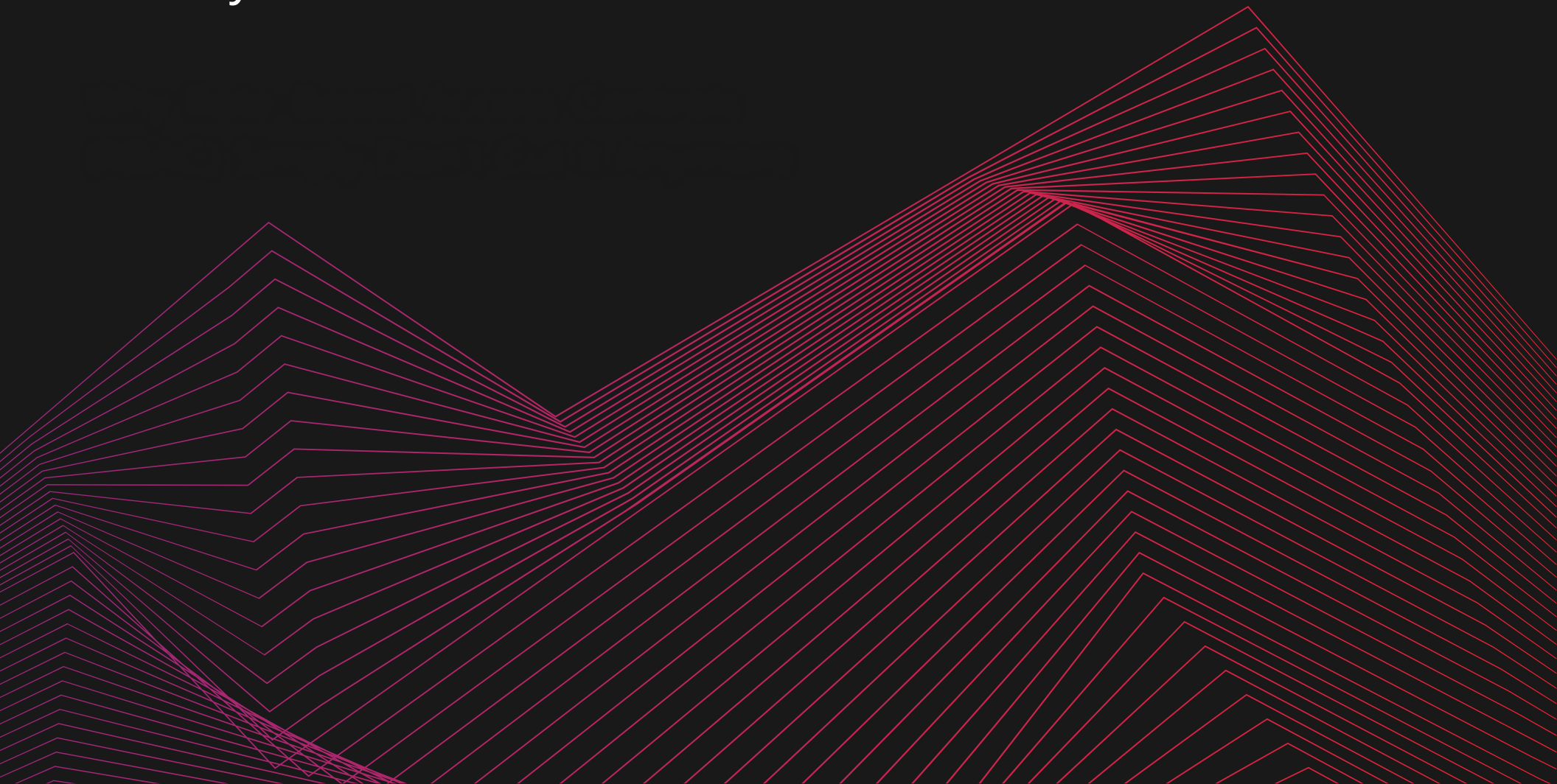
Solve Modern Policy Challenges with Attribute-Based Access Controls (ABAC)

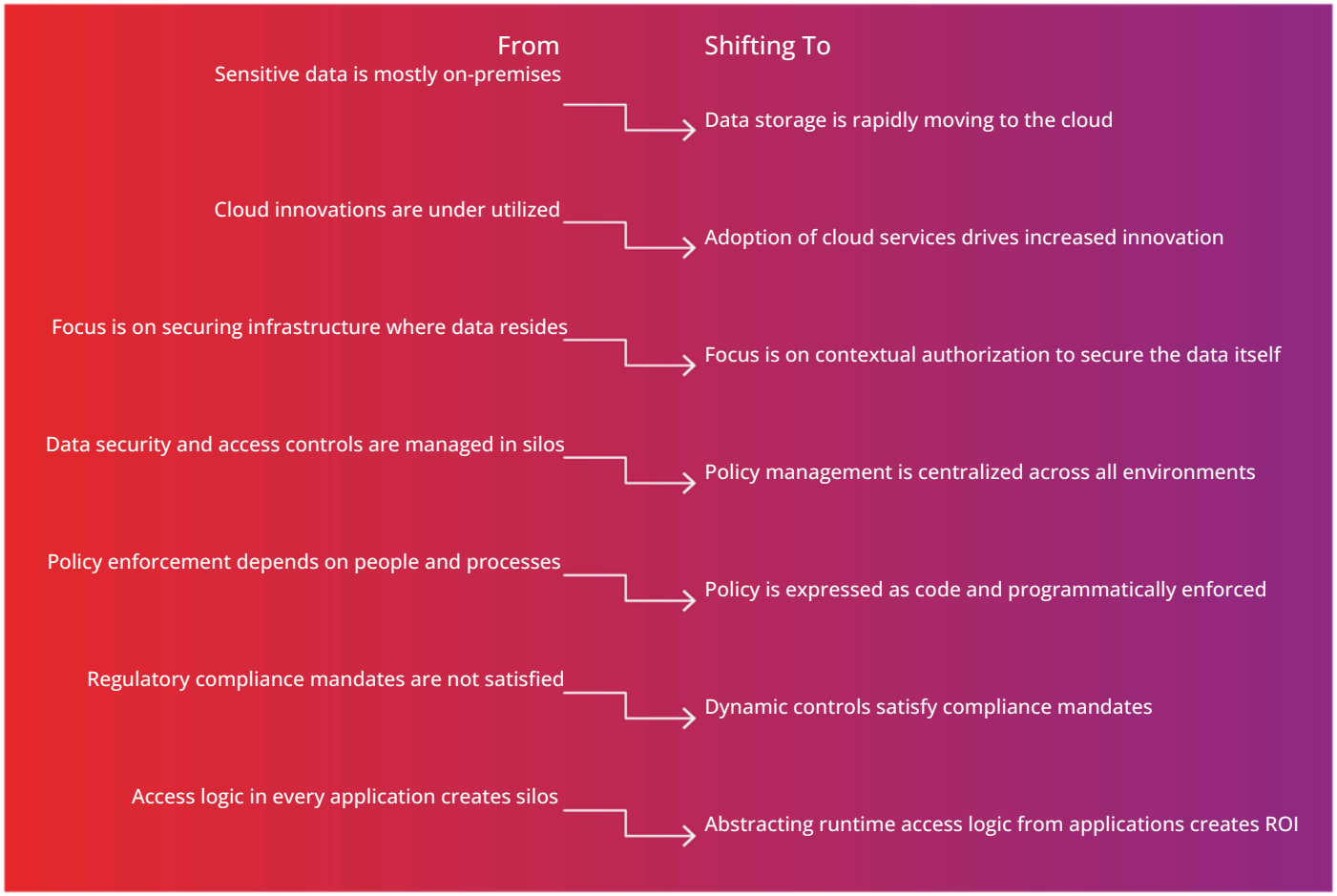
Why Role-Based Access Controls (RBAC) Simply Don’t Cut it Anymore



Data Security Paradigm Shift

Datais the most valuable asset in theworld today, and powerfulnew policy controls are needed to address modern challenges.

There is an explosion of data in terms of volume and complexity due to the exponential growth in cloud computing, mobile data traffic, and the development and adoption of technologies that depend on connected systems, processes, workflows, and applications.



Legacy Access Control Mechanisms are Inadequate

Accesscontrol systems are usedto control the actions, functions, applications, and operations oflegitimate userswithin an organization. They secure theassets wevalue — data, services, resources, workloads — by granting or denying access to them. The need for access controls crosses all industries, but it's most urgently felt by highly-regulated organizations.

Healthcare

Financial

Technology

Legal

Government

Consider the challenges presented by a few specific use cases: • Healthcare: A lab testing organization, expanding to meet growing needs, wants to limit the access to personal health information to only those employees who have completed HIPAA training. Rapid onboarding is creating gaps between what employees have and should have access to in lab systems.

• Technology: A technical services company, pushed to expand their remote workforce capabilities, is seeking to minimize dependence on their VPN for all but very high security use cases. The challenges presented by a global workforce are daunting.

• Legal: A court system creates a new role and data marking for every specific court case, but employees have gotten overwhelmed by the number of markings. Mistakes are being made and a new solution is needed.

Organizations attempt to solve these use cases with legacy methods:

- Template-based and static models
  Every object needs its own access controllist (ACL) – leading to over privileging or granting access morebroadly to users.
- Identity solutions
  These reference lessstatic information stores like active directories and assign privilegesto roles, instead of the users themselves.
- Role-based access control (RBAC)
  Most security measures depend on RBAC,which maps naturally to an organization's structure. The most common method for assigning access to information is based on the individual'sneed for the information, which is a function of their job or role within the organization.

Unfortunately, roles and groups fall short of solving modern security challenges in many ways...

Shortcomings of Role-Based Access Control (RBAC)

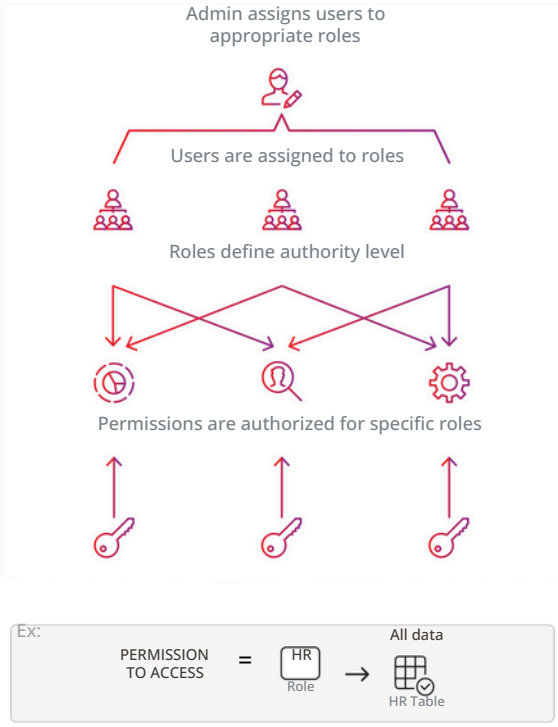
Inan RBACmodel, the role assignedto an individual implicitly grants them the predetermined levels of access based on that role. For example, a user assigned to an HR role, can only perform certain operations within HR applications. Access to finance apps will be denied (unless the user is also assigned to that role).

RBAC needs to be painstakingly managed, often involving significant manual intervention. It's not easy to represent cross-functional project access through roles or groups, and most organizations wind up with more roles to manage than they have actual employees!

When trying to secure sensitive data consistently across hybrid or cloud environments, RBAC falls short of being a productive, scalable, and flexible solution that can deliver authorization decisions based on the context of access requests. Considering additional attributes will make a tremendous difference in responding to modern challenges.

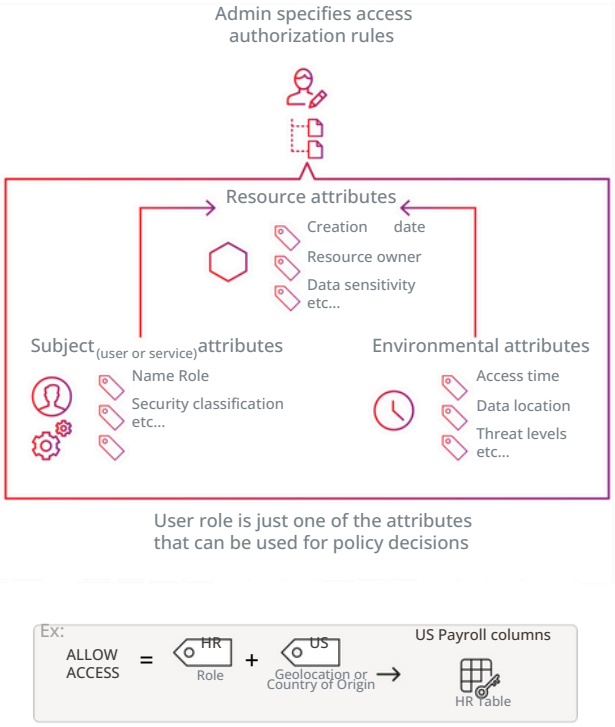
Role-based Access Control (RBAC)

RBACprovides access to resourcesor information based on userroles



Attribute-based Access Control (ABAC)

ABACprovidesaccessrights basedonuser, environment,orresource attributes



Shortcomings of Role-Based Access Control (RBAC)

Therateofchangein today'sbusiness environmentmakes RBACimplementations:

- **Complex**

  When thelevelofgranularityneededforauser's access controlistoodetailed,managingitbecomes complicated.Also,whenauserhastoomanyroles assignedtothem,anychangesnotimplemented by IT adminscanreducetheaccesscontroleffectiveness and createsecurityholes.Roledefinitioncanbe a contentiousandtime-consumingprocess,making it the costliestcomponentofRBACimplementation. These areintegraltoRBAC'ssuccess.

- **Inflexible**

  RBAC requiresintimateknowledgeofthesecurity layout ofyourorganizationandofhowpermissions are grantedtothedifferentroles.Organizations also needtocomplementtheirinformationaccess policies withgeneraladministrationpolicies.Once deployed,subsequentrealigningofworkflowand positions,towhateverextentnecessary,maybe very expensive,difficult,andtimeconsuming.This increasestheriskofdatabreach,withsignificant financial and reputational consequences.

- **Unscalabe**

  As rolesevolve andchange, the time to process and implement them can't keep up with the pace of business. Organizations begin drowning in more roles than actual employees, creating new ones to handle every edgecase,remote access scenario, or special project. RBACbecomes difficult to maintain and manage, andtheresulting work-around solutions becomemajorproblems in the long run.

- **Non-Contextual**

  With the explosionofdata and how it's accessed, context has becomecritical to securing sensitive data. Managing accessbased just on roles for an increasingly mobileand remote workforce represents increased riskthat is not well-managed.

Just consider the three specific use cases introduced at the beginning:

Healthcare

The lab testing organization wants to consider the completion of training modules as required steps before granting access to certain parts of their applications that handle personal health information.

Technology

The technical services company needs to consider information about where a user is located, risk scores of local devices, and the sensitivity of the data being requested before granting access.

Legal

The court system is determined to limit access to certain cases based on case assignments. They want to further constrain access to managed devices for approved locations.

Trying to solve these modern challenges with RBAC models presents significant shortcomings.

Context is King: Authentication vs Authorization

As data is shared internally and externally — across disparate applications, environments, data stores, devices, and services — securing and controlling access to it is critical. The context of every access point — identity, role, location, risk profile, and other attributes — needs to be considered to dynamically authorize access to your data based on policies that can consider roles in addition to other attributes.

Organizations often conflate authentication with authorization:

- Authentication

  Verifiesasubject'sidentity to grant or deny access

  » Role-based identities are used as a proxy for protecting data and resources since users are granted access to all resources once they authenticate

  » If additional authorization is used, it is implemented separately in each application

- Authorization

  Decision to allow or deny a verified subject access to data or resources based on the context of the access request

  » Roles can easily be implemented in ABAC as subject attribute values. Authentication is still required, but authorization is no longer automatically granted.

  » Policies constructed against multiple attributes can express more complex requirements. This makes them ideal for handling the dynamic and contextual needs of cloud environments with precision.

  » Abstracting policy to a separate layer allows it to be managed and enforced from a central location

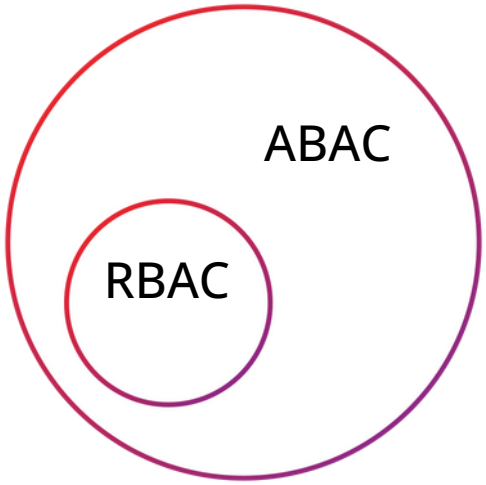
Example: As a member of the HR group, a user in the US is granted the ability to authenticate into an HR application, but in today's world of privacy regulations, the user may no lon- ger be authorized to view personal data for EU employees.

Multiple instances of this HR app with geographically partitioned datasets and matching roles can be created to tackle this problem, but that creates a mess for appdev, identity, and IT teams.

Now imagine thousands to millions of such requirements across your organization… managing the complexity quickly gets out of control. You get the picture!

Attribute-Based Access Control (ABAC) to the Rescue

Assigning users to roles and groups to manage access is a standard practice in virtually all organizations, but roles have been hyped and oversold as the only way to govern access. RBAC may indeed suffice for many use cases, but it's no longer sufficient to deal with all of the contextual and dynamic needs of today's business.



Organizations need to effectively manage: •

Turnovers and business reorganizations
• Workforce fluidity, with abrupt changes to roles, structures, and behaviors

• M&A and deal room activities that require management of who can access what information for how long

• Lots of temporary projects that require add/ remove requests to be processed urgently, es- pecially when they are tied to revenue

• Movement of data and workloads to the cloud creates complex and siloed access management processes

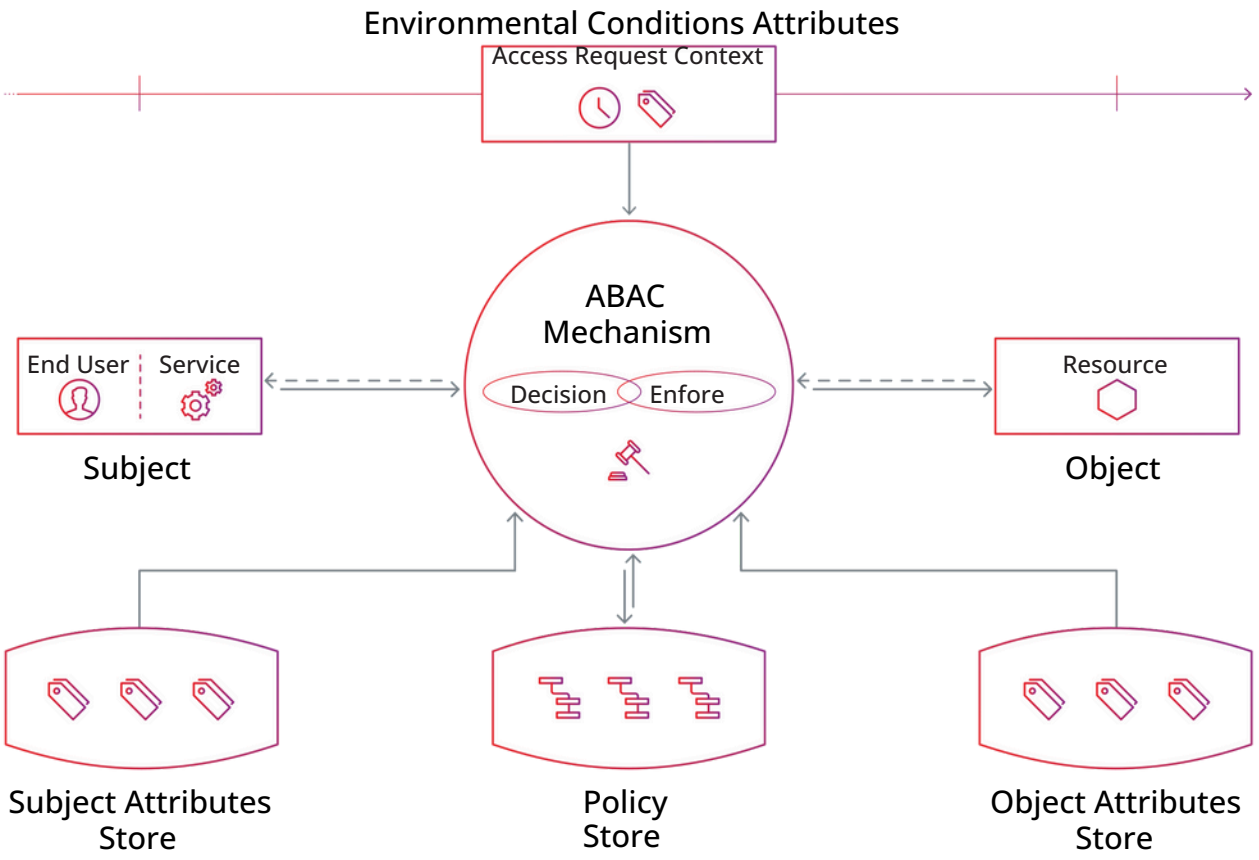
• Contextual access of an increasingly mobile and remote workforce
• Evolving and new privacy regulation mandates
Rather than proliferating the number of roles to manage context for making access control deci- sions, ABAC is capable of everything RBAC can do, plus it can use additional elements for authoriza- tion. In fact, ABAC augments RBAC in most cases.

ABAC Overview

ABAC goes beyond just users and roles to perform a comprehensive evaluation across multiple vectors:

ř What type of data or resource is it? ř Where is it located? ř Where is the subject requesting access located? ř How is access being requested? ř Over what network? ř What's the security profile of that network ř What's the risk profile of the subject or device involved?

Attribute-Based Access Control (ABAC) to the Rescue



NIST describes ABAC as:

> ***An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions.***
> ***NIST Special Publication 800-162***

ABAC components:

•Subject:User,device, service, or other entity requesting access to perform an operation

• Operation: Read, write, edit, execute, modify, copy, or delete

• Resource: Any object or service like a structured database, command-line interface to a compute service, workload, etc.

• Context: Environmental conditions like network details or software version that en- able situational and contextual awareness

• Attribute(s): One or more characteristics of all the above • Policies: Written rules and relationships that govern when access can be granted based on attributes

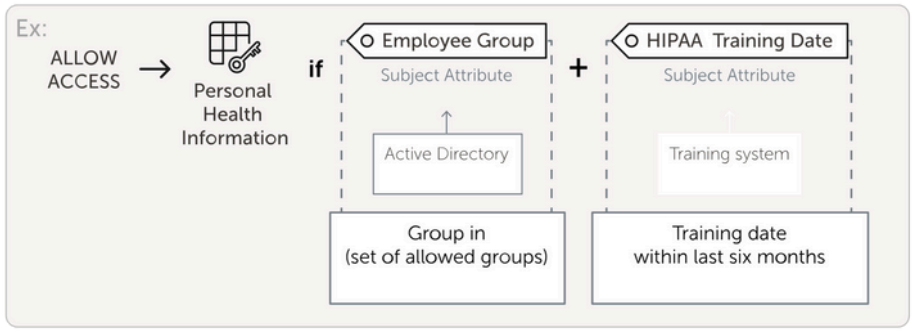
Attribute-Based Access Control (ABAC) to the Rescue

ABAC Myth Debunked

Implementing ABAC models is too complex — this is a common myth, but it couldn't be further from the truth!

The amount of work required to design and maintain an existing RBAC model may make you think that an ABAC model will magnify existing authorization and governance problems. But the perceived complexity of scaling and managing an ABAC model is just a myth. RBAC models, which require a tremendous amount of planning to think through every potential edge case in advance, are fundamentally far more complex. ABAC models require you to just describe the facts dynamically to provide real-time access decisions based on context. Let's look at those three example use cases and the attributes that will help solve them more easily:

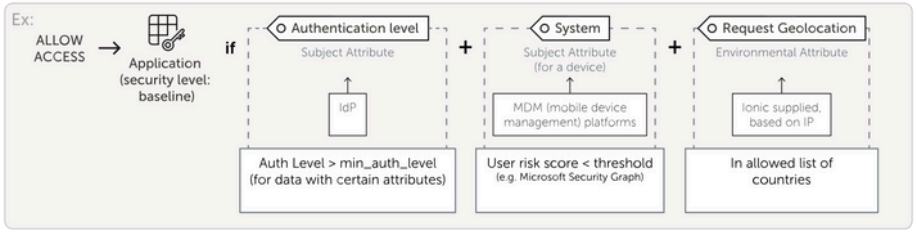
Healthcare

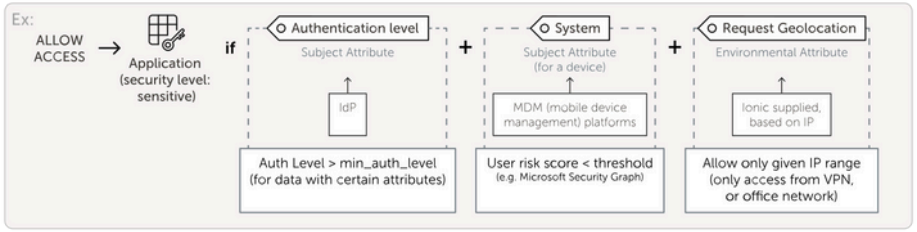



```
// Policy Rule A
subject:group-name
string-at-least-one-member-of
string:$allowed_groups
```

```
// Policy Rule B
subject:com.acme.hippa_training_completed_date
date-greater-than
add-yearMonthDuration(environment:current-dateTime,(0,-6))
```

Technology



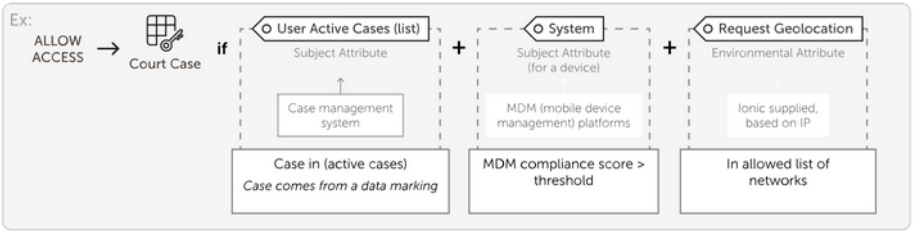



```
// Policy Rule A
subject:com.acme.case-mgmt.active-cases
string-at-least-one-member-of
data:com.acme.case-mgmt.allowed-cases
```

```
// Policy Rule B
environment:com.acme.device-mgmt.risk-score
double-less-than
double:$max_risk
```

```
// Policy Rule C
environment:ipAddress
ipAddress-regexp-match
string:$ip_regex
```

Legal




```
// Policy Rule A
subject:com.acme.case-mgmt.active-cases
string-at-least-one-member-of
data:com.acme.case-mgmt.allowed-cases
```

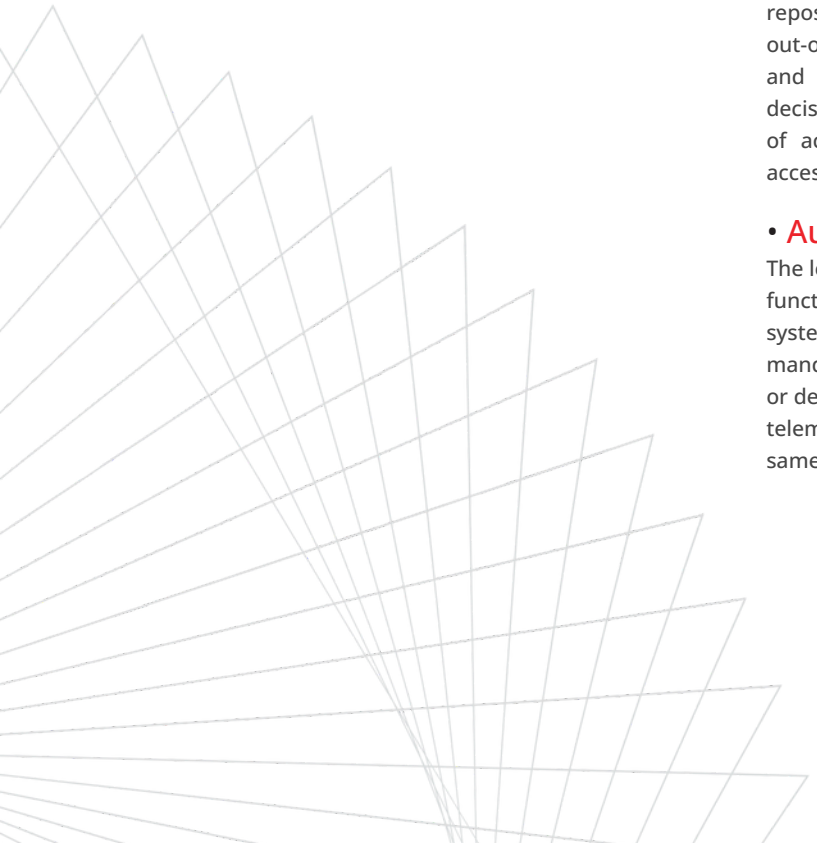
```
// Policy Rule B
environment:com.acme.device-mgmt.risk-score
double-less-than
double:$max_risk
```

```
// Policy Rule C
environment:ipAddress
ipAddress-regexp-match
string:$ip_regex
```

ABAC Benefits

If policy is the new perimeter, then it needs to be enforced not just against users and roles — which have been the domain of IAM systems — but against data and resources.

ABAC provides:



• Relevant Context

Considering morecontext in eachaccessdecision provides bettersecurity becauseyounolongerhave a single point ofcompromise. Thisisafoundational concept for a ZeroTrust securitystrategy,asrisk-based decision-makingcandeny accessbasedondetected threat scores oftheuser, device,network,andmore.

• Real-Time Attribute Sourcing

ABAC models can source attributesfromanysystem or repository in real-time, reducing theriskofincorrect or out-of-date information that existsacrossmultiple silos and layers of abstraction beforeitcanbeused in access decisions.ABAC captures amorecomplete understanding of access, with betterwaystodictate the conditions of accessand appropriateuseofinformation.

• Auditable Visibility

The levels of abstraction required to make RBAC models function expose sensitive information across multiple systems and make it challenging to address compli- ance mandates, let alone trace back why access was approved or denied. ABAC provides security teams with valuable telemetry to monitor authorization decisions, while the same visibility improves compliance efforts.

• Change Resiliency

RBAC abstraction levelsalso make it challenging to identify all the touch points where updates are required when organizational orsecurity changes need to be implemented. This increases administrative time spent and cost to maintain data security. ABAC simply requires rewriting policy against existing attributes or assigning new ones to accommodatethe required changes.

• Operational Flexibility

ABAC avoids the need forcapabilities to be assigned directly to roles or groups.This is subtle, but it means an administrator can manage policy without need- ing to directly modify (andlikely proliferate) roles to address change. More dynamic attribute-based policy ensures data and resources are made avail- able just in time to those who need them.

• Programmatic Enforcement of Compliance Mandates

Turn compliance mandates into code that can be enforcedprogrammatically across the orga- nization. ABAC policies require fewer levels of abstractionto enforce mandates more accurate- ly and consistently across any environment.