AYI NEDJIMI Consultants

RBAC, ABAC, PBAC : modèles de contrôle d'accès comparés

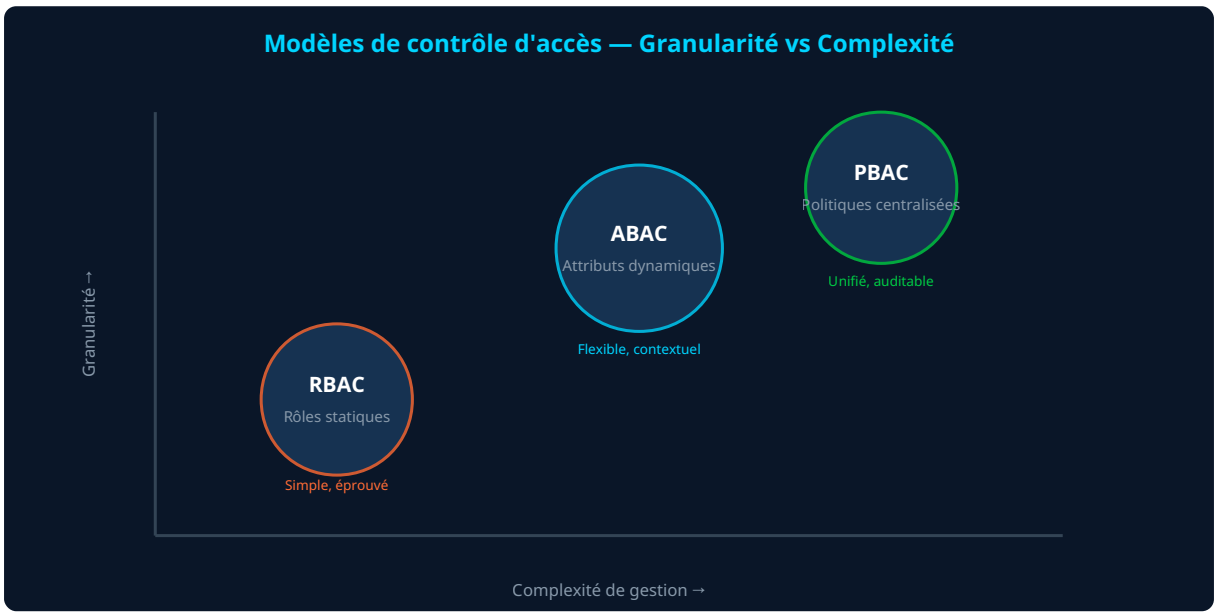
Catégorie : IAM et Gestion des Identités Lecture : 6 min Publié le : 12/03/2026 Auteur : Ayi NEDJIMI

RBAC, ABAC et PBAC : comparaison détaillée des modèles de contrôle d'accès avec cas d'usage, avantages et guide de choix pour votre architecture IAM.

Qui peut accéder à quoi, quand et dans quelles conditions ? Cette question fondamentale du contrôle d'accès se décline en plusieurs modèles, chacun avec ses forces et ses limites. Le RBAC (Role-Based Access Control) attribue des permissions via des rôles prédéfinis. L'ABAC (Attribute-Based Access Control) évalue des attributs contextuels en temps réel. Le PBAC (Policy-Based Access Control) unifie les règles dans un moteur de politiques centralisé. Pour l'architecte sécurité, le choix du bon modèle — ou de la bonne combinaison de modèles — détermine la granularité, la flexibilité et la maintenabilité de toute la chaîne d'autorisation. Ce guide vous propose une analyse comparative approfondie de ces trois approches, illustrée par des cas d'usage concrets issus d'environnements de production. Nous aborderons aussi les modèles émergents comme le ReBAC (Relationship-Based Access Control) et leur pertinence dans les architectures modernes. L'objectif est de vous donner les clés pour concevoir un modèle de contrôle d'accès adapté à la complexité de votre organisation, sans tomber dans le piège du sur-engineering ou du sous-dimensionnement.

Points clés à retenir

- **RBAC** est simple et mature mais manque de granularité pour les besoins complexes
- **ABAC** offre une granularité maximale mais sa complexité de gestion peut devenir un frein
- **PBAC** centralise les décisions d'accès dans un moteur de politiques externalisé
- La majorité des organisations utilisent un **modèle hybride RBAC + ABAC**
- Le modèle choisi doit s'aligner sur la maturité de votre gouvernance des identités



RBAC : le modèle par rôles statiques

Le *Role-Based Access Control* est le modèle le plus répandu et le plus simple à comprendre. Chaque utilisateur se voit attribuer un ou plusieurs rôles (Administrateur, Éditeur, Lecteur, Manager). Chaque rôle est associé à un ensemble de permissions (lire, écrire, supprimer, administrer). L'utilisateur hérite des permissions de ses rôles. C'est le modèle natif d'Active Directory (groupes de sécurité), d'**Azure RBAC** et de la plupart des applications enterprise.

Les forces du RBAC : simplicité de compréhension et d'administration, auditabilité directe (qui a quel rôle = quelles permissions), compatibilité universelle avec les applications. Les limites : le RBAC ne gère pas le contexte. Un utilisateur avec le rôle « Manager RH » a les mêmes permissions à 3h du matin depuis un café Wi-Fi public qu'à 10h depuis son bureau. Et quand le nombre de rôles explose pour couvrir toutes les combinaisons possibles (role explosion), l'administration devient un cauchemar. Les attaques sur Active Directory exploitent souvent cette prolifération de rôles et de groupes imbriqués.

ABAC : le contrôle d'accès contextuel

L'*Attribute-Based Access Control* évalue les décisions d'accès en fonction d'attributs évalués en temps réel. Quatre catégories d'attributs interviennent : les attributs du **sujet** (département, niveau d'habilitation, pays), les attributs de la **ressource** (classification, propriétaire, date de création), les attributs de l'**action** (lire, modifier, approuver) et les attributs de l'**environnement** (heure, localisation, type de terminal, niveau de risque).

Une règle ABAC typique : « Un utilisateur du département Finance, avec un niveau d'habilitation Confidentiel, peut lire les rapports financiers classifiés Confidentiel, uniquement depuis un terminal géré par l'entreprise, pendant les heures ouvrées ». Cette granularité est impossible à atteindre avec le RBAC seul. L'accès conditionnel d'Entra ID est une implémentation d'ABAC : il évalue des attributs (utilisateur, terminal, localisation, risque) pour décider d'autoriser, bloquer ou renforcer l'authentification.

PBAC : les politiques centralisées avec OPA et Cedar

Le *Policy-Based Access Control* externalise les décisions d'accès dans un moteur de politiques dédié, séparé de l'application. L'application demande au moteur « cet utilisateur peut-il effectuer cette action sur cette ressource ? » et reçoit une réponse binaire (allow/deny). Les politiques sont écrites dans un langage dédié et gérées comme du code (policy-as-code), versionnées dans Git, testées et déployées via CI/CD.

Open Policy Agent (OPA) avec le langage Rego est la solution PBAC open source de référence. **Cedar** (développé par AWS pour Amazon Verified Permissions) est une alternative récente avec un langage plus accessible. Google utilise **Zanzibar** en interne, et son implémentation open source **SpiceDB** gagne en popularité pour les modèles ReBAC. Le PBAC brille dans les architectures microservices où chaque service a besoin de prendre des décisions d'autorisation cohérentes sans dupliquer les règles. L'intégration avec une architecture Zero Trust est naturelle : le policy engine est le Policy Decision Point (PDP) central.

Critère	RBAC	ABAC	PBAC
Granularité	Rôle	Attribut	Politique
Contexte dynamique	Non	Oui	Oui
Complexité admin	Faible	Élevée	Moyenne
Auditabilité	Directe	Complexe	Excellente (policy-as-code)
Scalabilité	Role explosion	Bonne	Excellente
Adoption marché	Universelle	Croissante	Émergente
Outils	AD, Azure RBAC, IAM	XACML, Entra CA	OPA, Cedar, SpiceDB

Le modèle hybride RBAC + ABAC : la réalité du terrain

En pratique, les organisations matures combinent RBAC et ABAC. Le RBAC définit les permissions de base via les rôles métier (un comptable accède aux modules comptables). L'ABAC ajoute les contraintes contextuelles (uniquement depuis un poste conforme, pendant les heures ouvrées, avec un MFA valide). Cette approche hybride offre le meilleur compromis entre simplicité d'administration et granularité de contrôle.

Dans Entra ID, ce modèle hybride est natif : les rôles Azure RBAC (RBAC) sont modulés par les politiques d'accès conditionnel (ABAC). Un utilisateur avec le rôle Contributor sur une subscription Azure ne peut exercer ce rôle que si les conditions d'accès sont remplies (MFA, terminal conforme, localisation autorisée). La gouvernance des identités (IGA) gère le cycle de vie des rôles RBAC, tandis que l'accès conditionnel applique les politiques ABAC en temps réel. Les solutions PAM ajoutent une couche supplémentaire pour les accès à privilèges.

ReBAC : le modèle émergent basé sur les relations

Le *Relationship-Based Access Control* (ReBAC) est un modèle émergent qui définit les droits d'accès en fonction des relations entre entités. Au lieu de « l'utilisateur X a le rôle Admin sur la ressource Y », ReBAC modélise « l'utilisateur X est propriétaire du document Y, et les propriétaires peuvent partager ». Google Drive, GitHub et Notion utilisent des modèles ReBAC en interne.

Le ReBAC excelle pour les applications collaboratives où les permissions dépendent de la structure organisationnelle et des relations entre utilisateurs et ressources. **SpiceDB** (open source, inspiré de Google Zanzibar) et **Amazon Verified Permissions** (basé sur Cedar) sont les implémentations les plus matures. Pour les architectures IAM enterprise classiques, le ReBAC reste complémentaire au RBAC/ABAC plutôt qu'un remplacement. Le NIST propose des lignes directrices pour intégrer ces modèles dans une architecture cohérente.

Guide de choix selon votre contexte

Le choix du modèle dépend de quatre facteurs. La **taille de l'organisation** : une PME de 200 utilisateurs n'a pas besoin d'ABAC complexe — le RBAC avec quelques politiques d'accès conditionnel suffit. La **complexité réglementaire** : les secteurs réglementés (finance, santé, défense) nécessitent la granularité de l'ABAC pour implémenter les contrôles de classification et d'habilitation. L'**architecture applicative** : les microservices bénéficient du PBAC centralisé, les applications monolithiques s'accommodent du RBAC intégré. La **maturité IAM** : démarrez par le RBAC, ajoutez l'ABAC progressivement et envisagez le PBAC quand la complexité des politiques le justifie.

Pour préparer le business case COMEX d'un projet de contrôle d'accès, quantifiez les risques actuels : nombre de violations de la séparation des tâches, comptes avec des privilèges excessifs, temps de traitement des demandes d'accès. Un audit ANSSI identifiera les gaps entre votre modèle actuel et les bonnes pratiques.

Questions fréquentes sur les modèles de contrôle d'accès

Peut-on migrer progressivement de RBAC vers ABAC ?

Oui, et c'est l'approche recommandée. Conservez vos rôles RBAC existants comme base et ajoutez des conditions ABAC progressivement. Par exemple, maintenez le rôle « Comptable » mais ajoutez la condition « terminal conforme + horaires ouvrés » pour accéder aux données financières sensibles. Cette approche hybride évite le big bang tout en augmentant la granularité du contrôle d'accès. La migration complète vers l'ABAC pur n'est d'ailleurs ni nécessaire ni souhaitable dans la majorité des cas.

Comment éviter le phénomène de role explosion en RBAC ?

La role explosion survient quand on crée un rôle pour chaque combinaison de permissions. Trois stratégies : la hiérarchie de rôles (rôles enfants qui héritent des rôles parents), la composition de rôles (un utilisateur cumule plusieurs rôles fins plutôt qu'un rôle monolithique) et le complément ABAC pour les conditions contextuelles au lieu de créer des rôles dédiés. Le role mining — analyse des permissions effectives pour consolider les rôles — est aussi un exercice de nettoyage périodique recommandé.

OPA est-il adapté aux petites organisations ?

OPA a été conçu pour les architectures cloud-native à grande échelle (Kubernetes, microservices). Pour une PME avec un SI classique (AD, quelques applications SaaS), OPA est surdimensionné. L'accès conditionnel d'Entra ID ou les politiques IAM d'AWS couvrent les besoins ABAC/PBAC sans infrastructure additionnelle. OPA devient pertinent à partir du moment où vous gérez des dizaines de microservices avec des politiques d'accès hétérogènes qu'il faut unifier.

Sources et références : ANSSI · MITRE ATT&CK

Synthèse et recommandations pratiques

Le contrôle d'accès est un sujet fondamental qui mérite une réflexion architecturale sérieuse. Ne choisissez pas un modèle par mode technologique — choisissez-le en fonction de vos contraintes réelles. Le RBAC reste le socle incontournable. L'ABAC enrichit ce socle avec du contexte dynamique. Le PBAC unifie les politiques dans les architectures distribuées. Et le ReBAC adresse les cas d'usage collaboratifs. Commencez simple, mesurez les gaps et évoluez progressivement. La maturité du contrôle d'accès se construit par itérations, pas par révolution.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr