


CISSP Domaine 5 : Gestion des accès et des identités.

À Propos de ce Contenu

 Guide Open Source pour la Cybersécurité






Démarche Open Source

Ce contenu est conçu dans une démarche open source et réalisé dans le cadre de l'entraide en cybersécurité. Son objectif est de partager des connaissances et d'encourager la collaboration au sein de la communauté.





Réutilisation Autorisée

-  Libre de réutiliser et partager
-  Adaptation autorisée
-  Mention de l'auteur requise



Usage Commercial Interdit

-  Commercialisation interdite
-  Contenu gratuit uniquement



Ma Vision

Promouvoir une communauté cyber solidaire, où le partage des connaissances reste gratuit et accessible à tous.



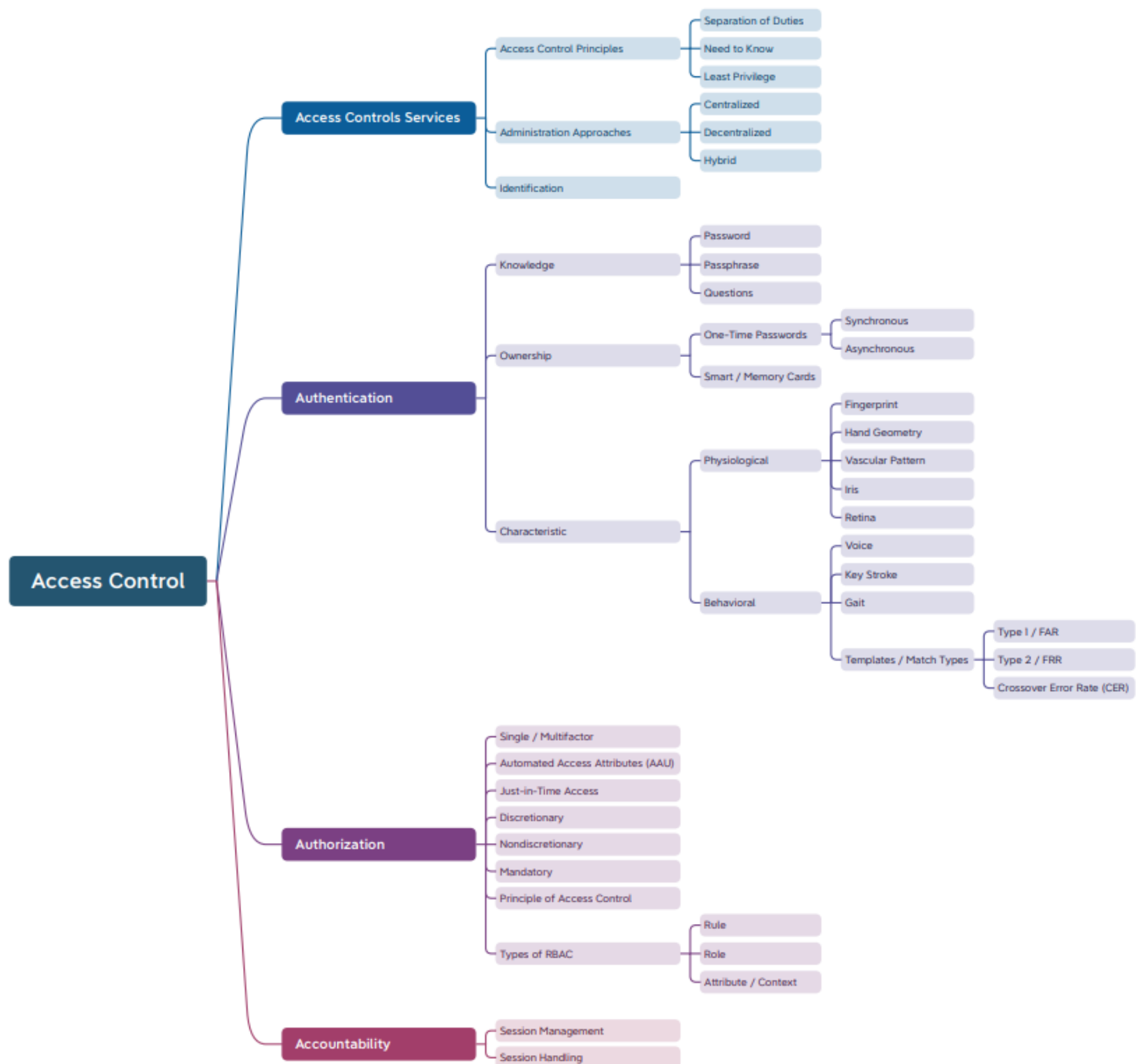
Ce document a été conçu dans le cadre de la révision du Domaine 5 de la certification CISSP.

Note importante : Il se peut que ce document contienne quelques erreurs. Il est essentiel de **vérifier les sources** si besoin.

N'hésitez pas à me suivre pour plus de contenu lié au domaine de la cyber sécurité :

<https://www.linkedin.com/in/nathan-lemaire-cyber/>

Mindmap des sujets abordés



Sommaire

1. Contrôle d'accès et gestion des identités

1.1 Contrôle d'accès

1.1.1 Principes fondamentaux du contrôle d'accès

- Les principes fondamentaux du contrôle des accès
- Applicabilité du contrôle d'accès
- Fonctionnement d'un système de contrôle d'accès
- Groupes vs Rôles

1.1.2 Approches d'administration du contrôle d'accès

- Administration centralisée
- Administration décentralisée
- Approche hybride

1.2 Concevoir une stratégie d'identification et d'authentification

1.2.1 Services de contrôle d'accès

1.2.2 Identification

1.2.3 Authentification par connaissance

1.2.4 Authentification par possession

- OTP : Processus asynchrone
- OTP : Processus synchrone
- Smart Card
- Memory Card

1.2.5 Authentification par caractéristiques

- Caractéristiques physiologiques
- Caractéristiques comportementales
- Types d'erreurs dans les systèmes biométriques (FRR, FAR, CER)

1.2.6 Facteurs d'authentification

1.2.7 Systèmes de gestion des identifiants (Credential Management Systems)

- Coffres-forts de mots de passe (Password Vault)

1.2.8 Single Sign-On (SSO)

- Processus de SSO
- Kerberos
- SESAME

1.2.9 CAPTCHA

1.2.10 Gestion de session

1.2.11 Enregistrement et vérification d'identité

1.2.12 Niveaux d'Assurances des Authentificateurs

- AAL1, AAL2, AAL3

1.2.13 Gestion Fédérée des Identités (Federated Identity Management - FIM)

1.2.14 Les Normes d'Accès Fédéré

- SAML
- WS-Federation
- OpenID
- OAuth

1.2.15 Responsabilité : Principe de Contrôle d'Accès

1.2.16 Accès "Just-in-time" (JIT)

1.3 Identité fédérée avec un service tiers

1.3.1 Identity as a Service (IDaaS)

1.4 Implémenter et gérer les mécanismes d'autorisation

1.4.1 Exploration des mécanismes de contrôle d'accès

- DAC, RBAC, Rule BAC, ABAC, MAC, Risk BAC

1.4.2 Description du Contrôle d'Accès Obligatoire (MAC)

1.4.3 Application des politiques d'accès

- Policy Enforcement Point (PEP)
- Policy Decision Point (PDP)

1.5 Gestion du cycle de vie des identités et des accès

1.5.1 Accès des fournisseurs

1.5.2 Cycle de vie des identités (Identity Life Cycle)

1.5.3 Revue des accès utilisateurs

1.5.4 Privilège Escalation

- Élévation de privilèges
- Utilisation de sudo
- Audit

1.6 Systèmes d'authentification

1.1.1 Contrôle d'accès

Le contrôle de l'accès physique et logique aux ressources est une composante essentielle de la sécurité en entreprise. Il garantit que seules les personnes ou les systèmes autorisés peuvent accéder aux ressources sensibles, tout en protégeant ces dernières contre les menaces internes et externes.



Qu'est-ce que le contrôle d'accès ?

Le contrôle d'accès repose sur un ensemble de mécanismes conçus pour protéger les ressources d'une organisation, tout en autorisant un accès contrôlé et ciblé aux utilisateurs ou systèmes légitimes (appelés sujets).

En d'autres termes, il s'agit de trouver l'équilibre parfait entre sécurité et fonctionnalité :

- **Sécurité** : Empêcher les accès non autorisés.
- **Fonctionnalité** : Permettre aux bonnes personnes d'accéder aux ressources dont elles ont besoin pour accomplir leur travail.


Le contrôle d'accès permet à la direction de :

1. Spécifier quels utilisateurs peuvent accéder au système (Qui?).
2. Définir quelles ressources ils peuvent utiliser. (Quoi?)
3. Déterminer quelles opérations ils peuvent effectuer (Comment?)
4. Assurer une responsabilité individuelle, c'est-à-dire savoir qui fait quoi et comment il le fait.

Mais comment y parvenir ? C'est là qu'interviennent les principes fondamentaux du contrôle d'accès.

Les principes fondamentaux du contrôles des accès :

Pour garantir un contrôle efficace, trois principes clés sont appliqués :




Besoin de savoir

Restreindre l'accès uniquement au personnel nécessitant cet accès pour accomplir ses tâches.

✓ **Application correcte**
Un médecin accède uniquement aux dossiers de ses patients

✗ **À éviter**
Accès à tous les dossiers médicaux sans distinction




Moindre privilège

Limiter les permissions au strict nécessaire pour l'utilisateur ou le système.

✓ **Application correcte**
Droits de lecture seule pour la consultation de documents

✗ **À éviter**
Droits administrateur pour tous les utilisateurs



Séparation des tâches

Impliquer plus d'une personne dans un processus pour éviter les erreurs et fraudes.

✓ **Application correcte**
Une personne initie le paiement, une autre l'approuve

✗ **À éviter**
Même personne pour créer et approuver les paiements

Point essentiel : Ces trois principes doivent être appliqués de manière coordonnée pour assurer une sécurité optimale. Chaque principe renforce les autres et contribue à créer un système de contrôle d'accès robuste.

Applicabilité du contrôle d'accès

Le contrôle d'accès s'applique à tous les niveaux d'une organisation et à tous les types de ressources :

- Les installations physiques : Locaux, bureaux, entrepôts, etc.
- Les systèmes/dispositifs : Serveurs, postes de travail, équipements réseaux.
- Les informations : Données sensibles, bases de données.
- Le personnel : Contrôle des identités et des rôles.
- Les applications : Logiciels et plateformes numériques utilisées par l'organisation.

Ainsi, il ne s'agit pas seulement de protéger un serveur ou un fichier, mais d'avoir une approche globale qui sécurise tous les aspects de l'organisation.

Question : C'est bien tout ces principes, mais comment les accès sont-ils gérés ?

Réponse : Avec des systèmes de contrôle d'accès.

Comment fonctionne un système de contrôle d'accès ?

Un système de contrôle d'accès repose sur le Concept de Moniteur de Référence (RMC), qui agit comme un intermédiaire entre les sujets (actifs) et les objets (passifs).



Imaginez un gardien de sécurité sophistiqué qui vérifie non seulement qui peut entrer, mais aussi ce que chaque personne peut faire une fois à l'intérieur. Le système de contrôle d'accès joue ce rôle crucial en protégeant les ressources de l'organisation.



Les composants principaux :

- **Sujet (actif) :** L'utilisateur ou le système qui demande l'accès (par exemple, un employé cherchant à ouvrir un fichier).
- **Objet (passif) :** La ressource protégée (par exemple, le fichier que l'employé souhaite ouvrir).
- **Règles :** Les conditions qui régissent l'accès, par exemple, "cet utilisateur peut uniquement lire ce fichier, mais pas le modifier".
- **Journalisation et surveillance :** Toutes les actions sont enregistrées pour garantir la traçabilité et identifier d'éventuels comportements malveillants.

Résumé de la mise en œuvre :



Résumé de la mise en œuvre :

Le système applique les règles via un noyau de sécurité, qui décide :

1. Si l'accès est autorisé ou non.
2. Comment cet accès doit être consigné pour assurer la responsabilité.

Ce modèle permet de centraliser la prise de décision, tout en maintenant un suivi rigoureux des accès aux ressources.

Transition vers les modes d'accès logiques

Une fois que le RMC a validé l'accès d'un utilisateur, les modes d'accès logiques précisent les actions autorisées sur la ressource. Ces modes sont essentiels pour garantir que chaque utilisateur ne fait que ce qu'il est autorisé à faire.

Ci-dessous les principaux modes d'accès logiques :



Créer (Create)

Ce mode permet d'ajouter de nouvelles données ou ressources dans le système. C'est comme un bibliothécaire qui peut ajouter de nouveaux livres à la collection.

Exemples concrets :

- Créer un nouveau document dans un dossier
- Ajouter un nouvel utilisateur dans le système
- Créer une nouvelle base de données



Lire (Read)

Ce mode permet de consulter les informations sans pouvoir les modifier. Comme un visiteur de la bibliothèque qui peut lire les livres mais ne peut pas les modifier.

Exemples concrets :

- Consulter un rapport
- Afficher le contenu d'un dossier
- Visualiser des données dans une application



Mettre à jour (Update)

Ce mode permet de modifier des données existantes. C'est comme un restaurateur de livres qui peut réparer et mettre à jour les informations des ouvrages existants.

Exemples concrets :

- Modifier un document existant
- Mettre à jour les informations d'un utilisateur
- Changer la configuration d'un système



Exécuter (Execute)

Ce mode permet de lancer des applications ou des scripts. C'est comme utiliser les équipements de la bibliothèque (ordinateurs, scanners) sans pouvoir les modifier.

Exemples concrets :

- Lancer un programme
- Exécuter un script
- Démarrer un service système



Supprimer (Delete)

Ce mode permet de retirer des données du système. C'est comme un bibliothécaire qui peut retirer des livres obsolètes ou endommagés de la collection.

Exemples concrets :

- Supprimer un fichier
- Effacer un compte utilisateur
- Retirer une application

⚠ Point important : Ces modes d'accès sont souvent combinés pour créer des profils d'utilisateurs spécifiques. Par exemple, un utilisateur standard pourrait avoir les droits de lecture et d'exécution, tandis qu'un administrateur aurait tous les droits. La bonne gestion de ces droits est essentielle pour la sécurité du système.

Groupes vs Rôles

Les Groupes :

- Une collection d'utilisateurs partageant des permissions similaires.
- Flexible, mais pas nécessairement aligné sur une fonction ou un poste précis.

Les Rôles :

- Ensemble de permissions associées à une fonction spécifique dans l'organisation.
- Permet une gestion précise et cohérente des accès selon les responsabilités.

Groupes	Rôles
Collection d'utilisateurs.	Basé sur un poste ou une fonction.
Adapté pour gérer des permissions temporaires.	Idéal pour des permissions alignées à une mission.

Exemple pratique :

- Un groupe "Équipe Marketing" peut avoir accès aux outils de gestion de campagnes publicitaires.
- Un rôle "Responsable RH" aura des droits spécifiques comme l'accès aux dossiers des employés.

1.1.2 Approches d'administration du contrôle d'accès

L'administration du contrôle d'accès est cruciale pour garantir que les utilisateurs accèdent aux bonnes ressources tout en minimisant les risques de sécurité. Il existe trois principales approches pour gérer cette administration : centralisée, décentralisée, et hybride. Chaque méthode a ses avantages, ses inconvénients et des cas d'utilisation spécifiques.

Administration centralisée



Dans un système centralisé, toutes les décisions et configurations liées aux accès sont gérées depuis un point unique, souvent par une équipe ou un outil central (exemple : un Active Directory ou un IAM - Identity and Access Management System).

Caractéristiques clés :

- Un système central contrôle les accès aux différentes ressources (systèmes, applications, données).
- Les permissions sont uniformes et appliquées globalement.
- Les utilisateurs se connectent généralement avec un identifiant unique (SSO - Single Sign-On).

Avantages :

1. **Simplicité administrative** : Toutes les règles et permissions sont centralisées, facilitant les audits et la gestion des accès.
2. **Cohérence** : Les mêmes politiques d'accès sont appliquées sur l'ensemble de l'organisation.
3. **Réduction des coûts** : Moins de duplication de tâches entre équipes.

Inconvénients :

1. **Point de défaillance unique** : Si le système central est compromis ou indisponible, cela peut affecter l'ensemble de l'organisation.
2. **Rigidité** : Moins de flexibilité pour des besoins spécifiques à des équipes ou départements locaux.
3. **Cible privilégiée pour les attaquants** : Le système central devient un objectif stratégique.

Exemple pratique :

- Dans une entreprise utilisant Active Directory, toutes les permissions (lecture, écriture, modification) pour les fichiers partagés sont gérées depuis un serveur centralisé.

Administration décentralisée



Avec une approche décentralisée, chaque équipe ou département gère ses propres permissions et décide localement qui a accès à quelles ressources.

Caractéristiques clés :

- Les droits d'accès sont configurés localement, souvent au niveau des systèmes ou des applications.
- Chaque département ou groupe peut avoir des règles adaptées à ses besoins.
- Pas de dépendance forte à un système central.

Avantages :

- 1. **Flexibilité** : Les équipes peuvent ajuster rapidement les permissions selon leurs besoins.
- 2. **Réactivité** : Les décisions d'accès sont prises localement, ce qui peut accélérer certaines opérations.
- 3. **Adaptation spécifique** : Permet de mieux répondre à des besoins variés au sein d'une organisation.

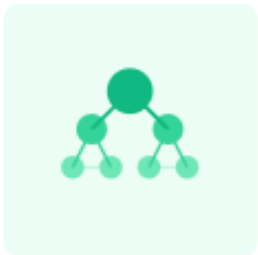
Inconvénients :

- 1. **Manque de standardisation** : Risque de politiques d'accès incohérentes entre départements.
- 2. **Répétition des tâches** : Chaque équipe doit gérer ses propres règles, ce qui peut augmenter la charge administrative.
- 3. **Lacunes de sécurité** : Des erreurs humaines ou un manque de coordination peuvent créer des failles.

Exemple pratique :

- Dans une entreprise avec plusieurs filiales, chaque filiale gère indépendamment les permissions des employés sur ses serveurs locaux, sans dépendre du siège.

Approche hybride



L'approche hybride combine les avantages des systèmes centralisés et décentralisés. Certaines permissions sont gérées globalement par une entité centrale, tandis que d'autres sont laissées à la discrétion des équipes locales.

Caractéristiques clés :

- Une **administration partagée** : Les politiques critiques sont centralisées, mais les équipes locales peuvent gérer certains aspects spécifiques.
- Un équilibre entre **uniformité globale** et **adaptabilité locale**.

Avantages :

- 1. **Équilibre optimal** : Offre à la fois la cohérence d'un système centralisé et la flexibilité d'un système décentralisé.
- 2. **Réduction des risques** : Minimisation des impacts d'un point de défaillance unique, car certaines permissions sont locales.
- 3. **Meilleure efficacité** : Les ressources critiques restent sous contrôle central, tandis que les détails opérationnels peuvent être gérés localement.

Inconvénients :

- 1. **Complexité de gestion** : Nécessite une coordination efficace entre les équipes centrales et locales.
- 2. **Formation accrue** : Les équipes locales doivent être formées pour comprendre et appliquer correctement les politiques définies par l'administration centrale.

Exemple pratique :

- Dans une organisation multinationale, l'accès aux applications globales (ex. : CRM, ERP) est géré de manière centralisée, mais chaque bureau régional gère l'accès à ses serveurs locaux ou applications spécifiques.

Résumé des approches

Approche	Avantages	Inconvénients
Centralisée	Simplicité, cohérence, gestion globale.	Point de défaillance unique, rigidité.
Décentralisée	Flexibilité, rapidité, adaptation locale.	Manque de standardisation, risques accrus.
Hybride	Équilibre, uniformité et flexibilité.	Gestion plus complexe, formation nécessaire.

Quand choisir quelle approche ?

1. **Centralisée** : Idéal pour les petites et moyennes organisations où la cohérence et la simplicité sont prioritaires.
2. **Décentralisée** : Adaptée aux grandes entreprises très segmentées, avec des besoins locaux spécifiques.
3. **Hybride** : Convient aux entreprises complexes (multinationales ou organisations avec de nombreux services) ayant besoin d'une gouvernance globale tout en maintenant une flexibilité opérationnelle.



À retenir pour cette partie :


Contrôle d'accès : Le contrôle d'accès fait référence à un ensemble de mécanismes qui travaillent ensemble pour protéger les ressources d'une organisation tout en permettant un accès contrôlé aux sujets autorisés.

- **Principes fondamentaux** : Les principes clés du contrôle d'accès incluent :
 - **Besoin de savoir**
 - **Moindre privilège**
 - **Séparation des tâches**
- **Applicabilité** : Le contrôle d'accès s'applique à tous les niveaux d'une organisation et couvre tous les types de ressources.

1.2 : Concevoir une stratégie d'identification et d'authentification


Les Sept Lois de l'Identité constituent un cadre fondamental pour la gestion de l'identité numérique, établissant les principes essentiels de la protection des données personnelles et de la confidentialité.

Les sept lois de l'identité ont été élaborées par Kim Cameron et une série d'autres experts en sécurité.




1. Contrôle par l'utilisateur et consentement

Maîtrise des données personnelles et consentement explicite



Points Clés

- Contrôle total des données
- Consentement explicite requis
- Transparence des utilisations



Implications

- Mécanismes de contrôle
- Processus de validation
- Documentation claire

Les systèmes d'identité doivent être conçus de manière à ce que les informations identifiant un utilisateur ne soient révélées qu'avec le consentement de l'utilisateur.



2. Divulgaration minimale

Limitation stricte des données collectées et partagées



Points Clés


- Collecte minimaliste
- Usage défini et limité
- Protection des données



Implications

- Audit régulier des données
- Justification des collectes
- Processus d'épuration

La solution d'identité la plus stable à long terme est celle qui révèle le moins d'informations d'identification et limite l'utilisation de ces données.



3. Parties justifiables

Partage limité aux entités légitimes

Les solutions d'identité doivent être conçues de manière à ne divulguer des informations d'identification qu'aux parties qui ont une raison justifiable et nécessaire de faire partie de la relation d'identité.



4. Identité dirigée

Contrôle précis des informations partagées



Points Clés

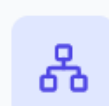
- Sélection précise des données
- Contrôle granulaire
- Gestion contextuelle



Implications

- Interface de sélection
- Paramètres avancés
- Options de partage

Les systèmes d'identité doivent prendre en charge les identifiants omnidirectionnels pour les entités publiques, ainsi que les identifiants unidirectionnels pour les entités privées.



5. Pluralisme des opérateurs et technologies

Interopérabilité et compatibilité des systèmes



Points Clés

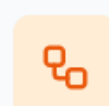
- Standards ouverts
- Compatibilité étendue
- Flexibilité technique



Implications

- Tests d'interopérabilité
- Documentation technique
- Support multi-plateforme

Les systèmes doivent être interopérables et compatibles avec différents fournisseurs.



6. Intégration humaine

Simplicité et intuitivité des systèmes



Points Clés

- Interface intuitive
- Accessibilité maximale
- Design centré utilisateur



Implications

- Tests utilisateurs
- Retours d'expérience
- Améliorations continues

Les systèmes doivent être conçus pour être simples et intuitifs pour les utilisateurs.



7. Expérience cohérente

Uniformité sur tous les systèmes et technologies



Points Clés

- Navigation fluide
- Cohérence visuelle
- Expérience unifiée



Implications

- Guidelines de design
- Standards d'interface
- Validation cross-platform

Le système d'identité doit fournir aux utilisateurs une expérience cohérente et simple.

💡 Les **Sept Lois de l'Identité** fournissent un cadre essentiel pour concevoir des systèmes d'identité qui respectent les utilisateurs, protègent leurs données personnelles, et garantissent la sécurité des organisations. En les appliquant, les entreprises peuvent offrir une expérience utilisateur fluide tout en respectant les exigences réglementaires et les attentes en matière de confidentialité.

1.2.1 Services de contrôle d'accès

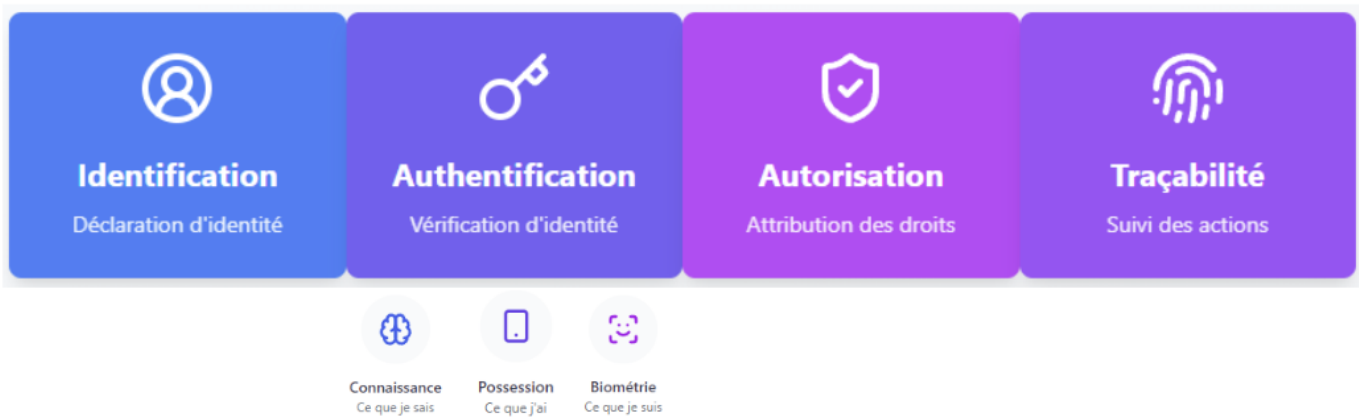
Le contrôle d'accès et les services associés sont des éléments fondamentaux de la sécurité organisationnelle. Les ressources et les utilisateurs peuvent être mieux protégés et tenus responsables de leurs actions. En fait, ce dernier point — la notion de **responsabilité (accountability)** — est le moteur principal des services de contrôle d'accès.

En tant que professionnel de la sécurité, il est essentiel de comprendre les implications d'un contrôle d'accès inexistant, faible ou inefficace, en particulier en ce qui concerne le **Principe de contrôle d'accès**.

Les composantes des services de contrôle d'accès sont illustrées dans la **Figure 5-2** et expliquées en détail dans la section suivante. Ces composantes sont :

- **Identification,**
- **Authentification,**
- **Autorisation,**
- **Responsabilité (Accountability).**

Parfois, les trois derniers processus sont regroupés sous l'acronyme **AAA** (Authentication, Authorization, Accountability).





À retenir pour cette partie "Services de contrôle d'accès" :

- Les services de contrôle d'accès comprennent :
 - **Identification,**
 - **Authentification,**
 - **Autorisation,**
 - **Responsabilité (Accountability).**
- **Identification :**

Fait référence à l'affirmation de l'identité d'un utilisateur ou d'un processus dans un système.
- **Authentification :**

Correspond à la vérification d'une identité au moyen de connaissances, d'une possession, ou d'une caractéristique (par exemple, mot de passe, carte ou empreinte biométrique).
- **Autorisation :**

Désigne le niveau d'accès défini pour l'utilisateur ou le processus identifié et authentifié.
- **Responsabilité (Accountability) :**

Concerne la bonne identification, authentification et autorisation, et inclut leur enregistrement et leur surveillance.
- **Responsabilité :**

Aussi connue sous le nom de **Principe du contrôle d'accès**, elle garantit la traçabilité et l'auditabilité des actions.

1.2.2 Identification

La notion d'identification est simple : pour accéder à un système, une identité unique doit être présentée, permettant ainsi de tracer toute activité jusqu'à un individu. L'utilisation d'identités partagées peut potentiellement contourner le Principe de contrôle d'accès et ne doit donc pas être autorisée.

Quelques exemples de méthodes d'identification :

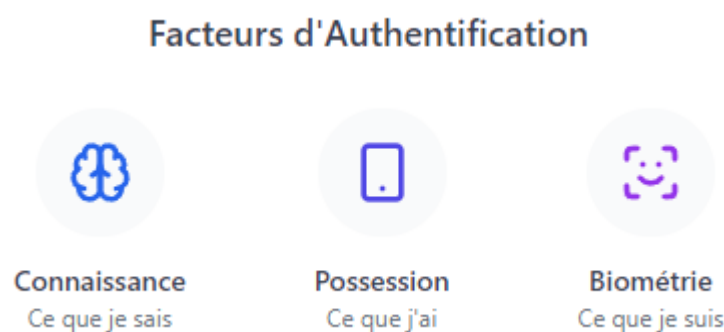
- Identifiant utilisateur (par exemple, prénom, nom de famille, ou les deux).
- Identifiant de compte.
- Carte d'accès.
- Biométrie.

Les identifications des utilisateurs doivent être :

1. **Univoques** : Relatives à un individu ou processus unique.
2. **Non descriptives d'un rôle** :
 - Par exemple, les comptes administrateurs ne devraient pas inclure le mot "admin".
 - Les comptes liés à des fonctions financières ne doivent pas pointer vers un rôle ou un poste spécifique.
3. **Créées et utilisées de manière sécurisée** :
 - Par exemple, utiliser un gestionnaire de mots de passe pour générer et stocker des mots de passe au lieu de les noter sur un document papier.

Authentification

En ce qui concerne l'authentification, il existe trois facteurs d'authentification permettant de vérifier l'identité d'un utilisateur. Ceux-ci sont résumés dans le tableau suivant :



1.2.3 Authentification par connaissance



L'**authentification par connaissance** est un composant des services de contrôle d'accès qui vérifie une identité à travers quelque chose que l'utilisateur **sait** (par exemple, un mot de passe, une phrase secrète, ou des questions de sécurité).

Explication

L'authentification par connaissance, également appelée "quelque chose que vous savez", consiste simplement à demander à une personne d'utiliser :

1. Un **mot de passe** :

- Un mot de passe peut être aussi simple que "password", ou quelque chose de plus complexe comme "HPv%ZWd%V79%#c\$U!Ke8".
- Les mots de passe complexes sont souvent difficiles à mémoriser, ce qui pousse les utilisateurs à les noter sur des Post-it ou à les stocker de manière non sécurisée.

2. Une **phrase secrète (passphrase)** :

- Une phrase secrète est similaire à un mot de passe, mais elle est généralement plus longue, plus complexe, et plus facile à mémoriser.
- Exemple : Une citation, une phrase de chanson, ou une ligne d'un livre peut servir de phrase secrète.

3. Une ou plusieurs **questions de sécurité** :

- Ces questions, aussi appelées "questions cognitives", sont définies par l'utilisateur.
- Elles peuvent inclure des réponses inattendues ou sans rapport direct avec la question (par exemple, une réponse non véridique).

Résumé des formes d'authentification par connaissance



Résumé :

- **Mot de passe** : Doit être unique et complexe pour éviter qu'il soit deviné ou piraté.
- **Phrase secrète** : Plus longue et généralement plus sécurisée qu'un mot de passe.
- **Questions de sécurité** : Conçues pour être mémorables pour l'utilisateur mais imprévisibles pour un attaquant.

Points importants

Les méthodes d'authentification doivent être uniques à l'utilisateur et difficiles à deviner ou à déterminer.

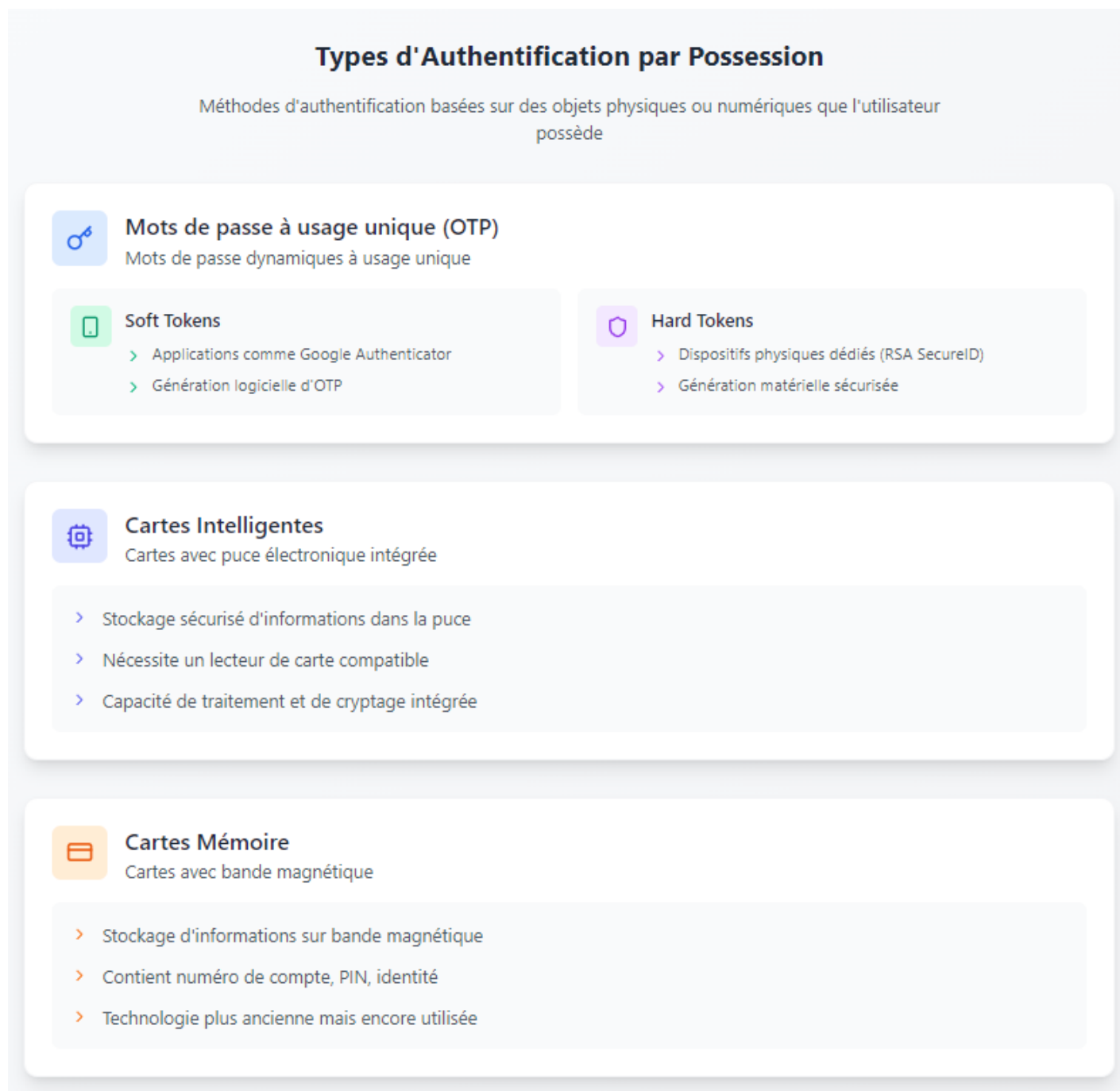
La complexité et l'unicité augmentent la sécurité et réduisent les risques d'usurpation d'identité.

1.2.4 Authentification par Possession



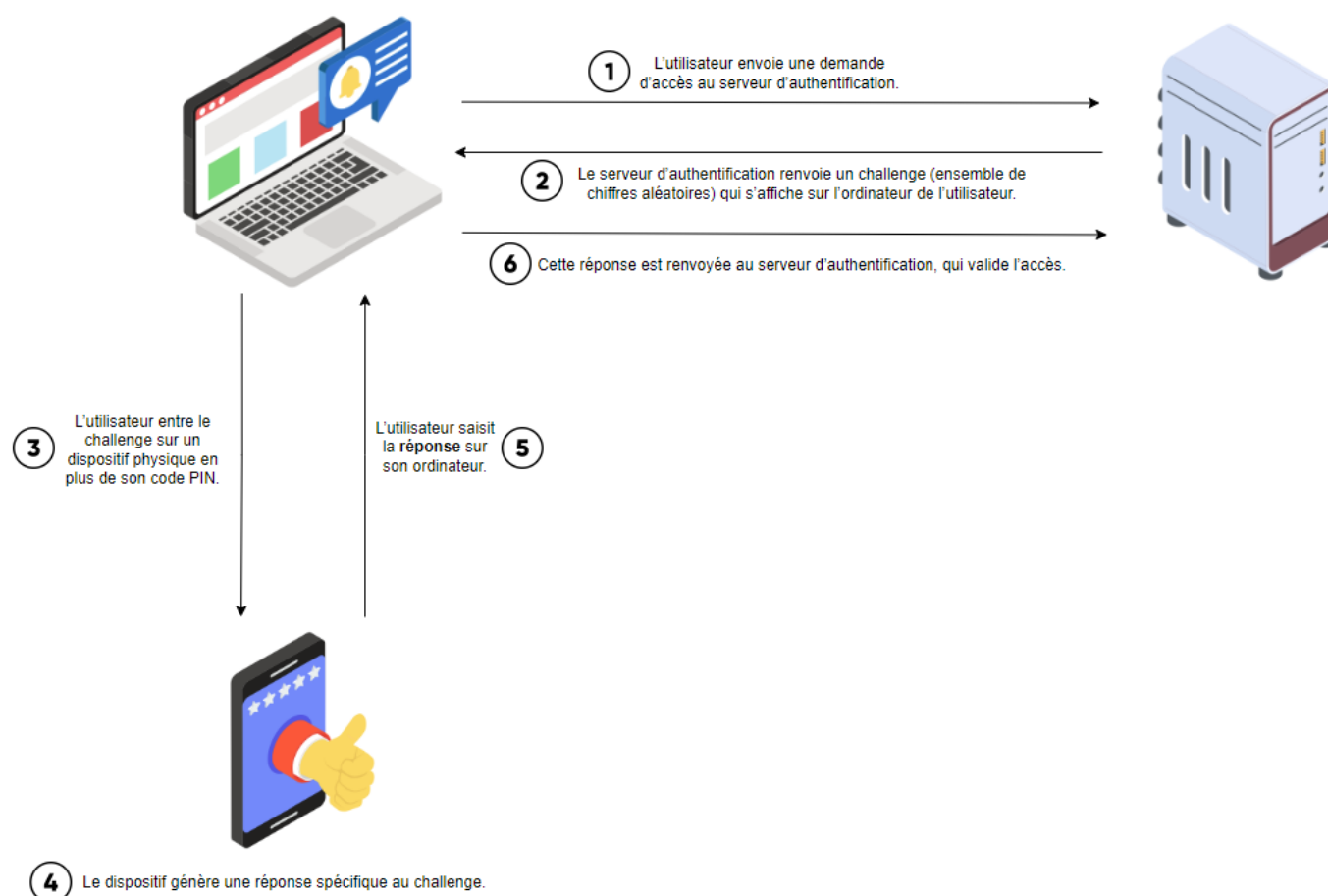
L'authentification par possession est une méthode de contrôle d'accès qui vérifie une identité à travers un objet que l'utilisateur possède.

Cette méthode repose sur des éléments physiques ou numériques qui prouvent l'identité de l'utilisateur.

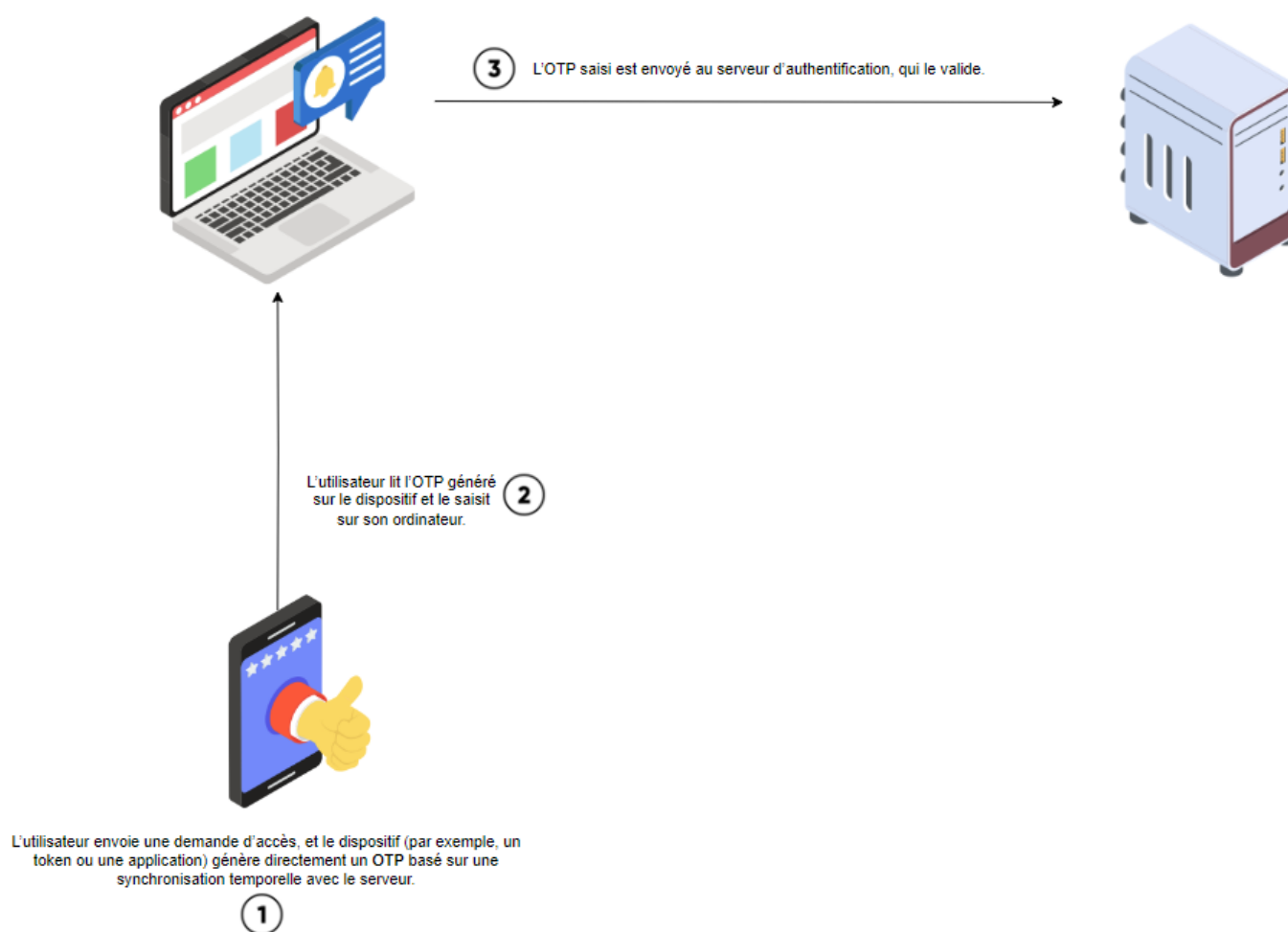


Description plus détaillée de ces processus :

OTP Asynchrones



OTP Synchrones



Différences entre cartes intelligentes (Smart Cards) et cartes mémoire (Memory Cards)

Les cartes intelligentes et cartes mémoire sont des éléments d'authentification par possession. Bien qu'elles aient des similitudes, elles se distinguent par leur fonctionnement et leur technologie :

Carte intelligente (Smart Card)	Carte mémoire (Memory Card)
Contient une puce intégrée capable d'effectuer des calculs et de générer des données d'authentification uniques à chaque transaction.	Contient une bande magnétique où les données sont stockées. Ces données sont lues à chaque transaction.

Détails sur les cartes mémoire :

- La bande magnétique située au dos de la carte stocke les informations.
- Les anciennes cartes utilisant uniquement la bande magnétique sont vulnérables à la fraude (par exemple, le skimming ou copie de bande magnétique).

Détails sur les cartes intelligentes :

- Contiennent une puce électronique qui agit comme un moteur de traitement pour accepter, stocker et envoyer des données.
- Les cartes modernes combinent souvent une puce **intelligente** et une bande magnétique pour améliorer la sécurité.

Résumé : Pourquoi utiliser l'authentification par possession ?

- **Avantages :**
 - Sécurité accrue grâce à l'utilisation de dispositifs ou applications uniques.
 - Les OTP réduisent les risques de piratage, car ils ne sont valides qu'une seule fois.
- **Inconvénients :**
 - Dépendance à un objet physique ou numérique (risque de perte ou de vol).
 - Les hard tokens peuvent entraîner des coûts supplémentaires pour l'organisation.

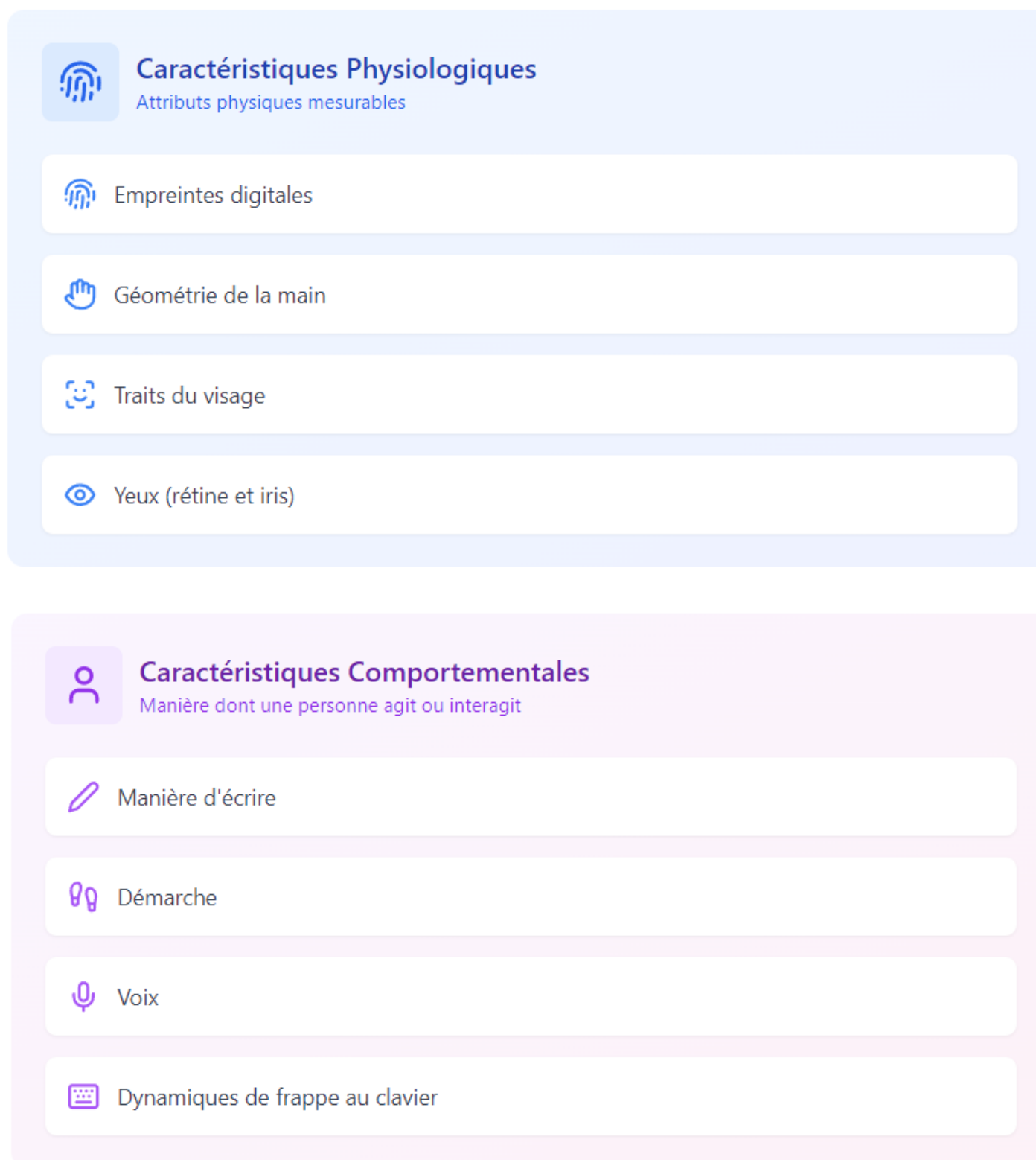
Ce type d'authentification est souvent utilisé comme **second facteur (2FA)** pour renforcer la sécurité des systèmes.

1.2.5 Authentification par Caractéristiques



L'authentification par caractéristiques fait référence à l'utilisation de données biométriques physiologiques (attributs physiques) ou comportementales (façon de se comporter) pour vérifier une identité.

La précision des dispositifs biométriques peut varier et n'est pas toujours garantie à 100%



Les caractéristiques physiologiques sont généralement plus stables dans le temps.

Les caractéristiques comportementales peuvent varier selon le contexte

Facteurs à considérer pour les dispositifs biométriques

1. **Vitesse de traitement** : Certains systèmes biométriques sont plus lents que d'autres types d'authentification.
2. **Acceptation par l'utilisateur** : Les utilisateurs peuvent être réticents si les systèmes biométriques sont intrusifs ou peu fiables.
3. **Protection des données biométriques** : La sécurité des données collectées est essentielle.
4. **Précision** : Les systèmes biométriques ne sont pas infallibles.

Types d'erreurs dans les systèmes biométriques

Les systèmes biométriques peuvent générer deux types d'erreurs :

1. Type 1 — Faux rejet (False Rejection Rate, FRR) :

- Un utilisateur légitime est rejeté à tort par le système.
- **Impact** : Résultat frustrant pour l'utilisateur, qui ne peut pas accéder au système.

2. Type 2 — Fausse acceptation (False Acceptance Rate, FAR) :

- Un utilisateur non autorisé est accepté par le système.
- **Impact** : Grave, car cela peut permettre à un individu malveillant d'accéder aux ressources.

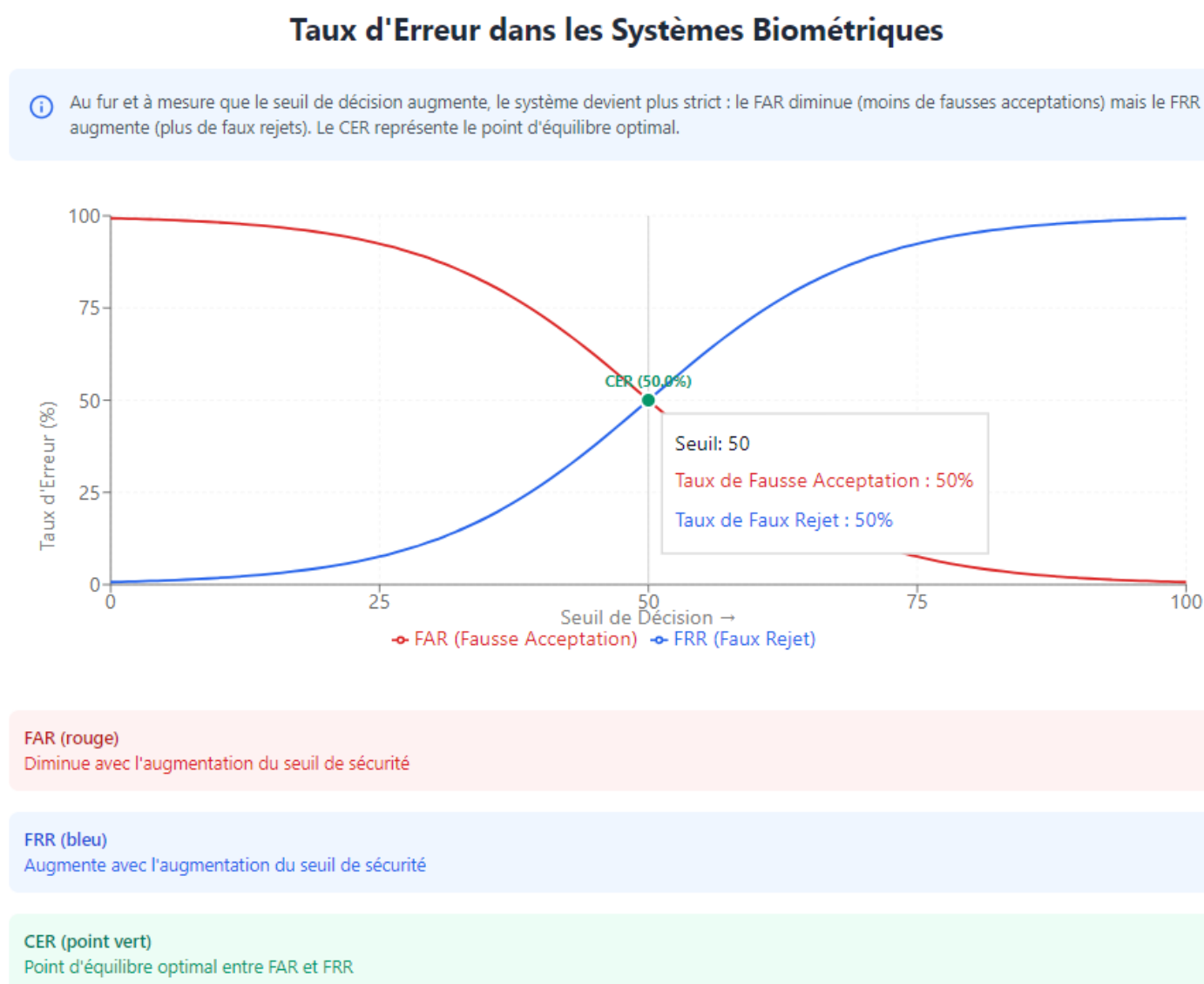
Importance relative des erreurs :

- Les erreurs de Type 2 (fausse acceptation) sont beaucoup **plus dangereuses** que les erreurs de Type 1, car elles compromettent la sécurité en donnant un accès à des utilisateurs non autorisés.

CER (Crossover Error Rate)

- Le **CER (taux d'erreur croisée)** représente le point d'intersection entre les taux d'erreur de Type 1 (FRR) et Type 2 (FAR).
- **Qu'est-ce que cela signifie ?**
 - Lorsque le taux de faux rejets diminue (Type 1), le taux de fausses acceptations (Type 2) augmente, et vice-versa.
 - Le **CER** est un indicateur de l'équilibre global et de l'efficacité d'un système biométrique.
 - Un **CER faible** signifie un système plus précis.

Voici un graphe représentatif des types d'erreurs et du CER :



Résumé sur les systèmes biométriques

1. **Précision** : Les dispositifs biométriques ne sont pas infaillibles et peuvent être sujets à des erreurs.
2. **Sécurité** : Les erreurs de Type 2 (fausse acceptation) présentent un risque majeur de compromission.
3. **CER** : Une métrique utile pour évaluer et comparer les systèmes biométriques.

Les systèmes biométriques offrent une sécurité avancée mais nécessitent un équilibre entre fiabilité, rapidité et acceptabilité pour les utilisateurs.

Les Modèles Biométriques

Importance de la protection des données biométriques :

- La protection des données biométriques est cruciale car leur exposition a des conséquences bien plus graves que l'exposition d'un mot de passe.
 - Si un mot de passe est compromis, l'utilisateur peut simplement le changer.
 - Si des données biométriques (comme une empreinte digitale ou un scan de l'œil) sont compromises, elles ne peuvent pas être remplacées — vous ne pouvez pas "faire pousser" une nouvelle empreinte digitale ou un nouvel œil.

Que sont les modèles biométriques ?

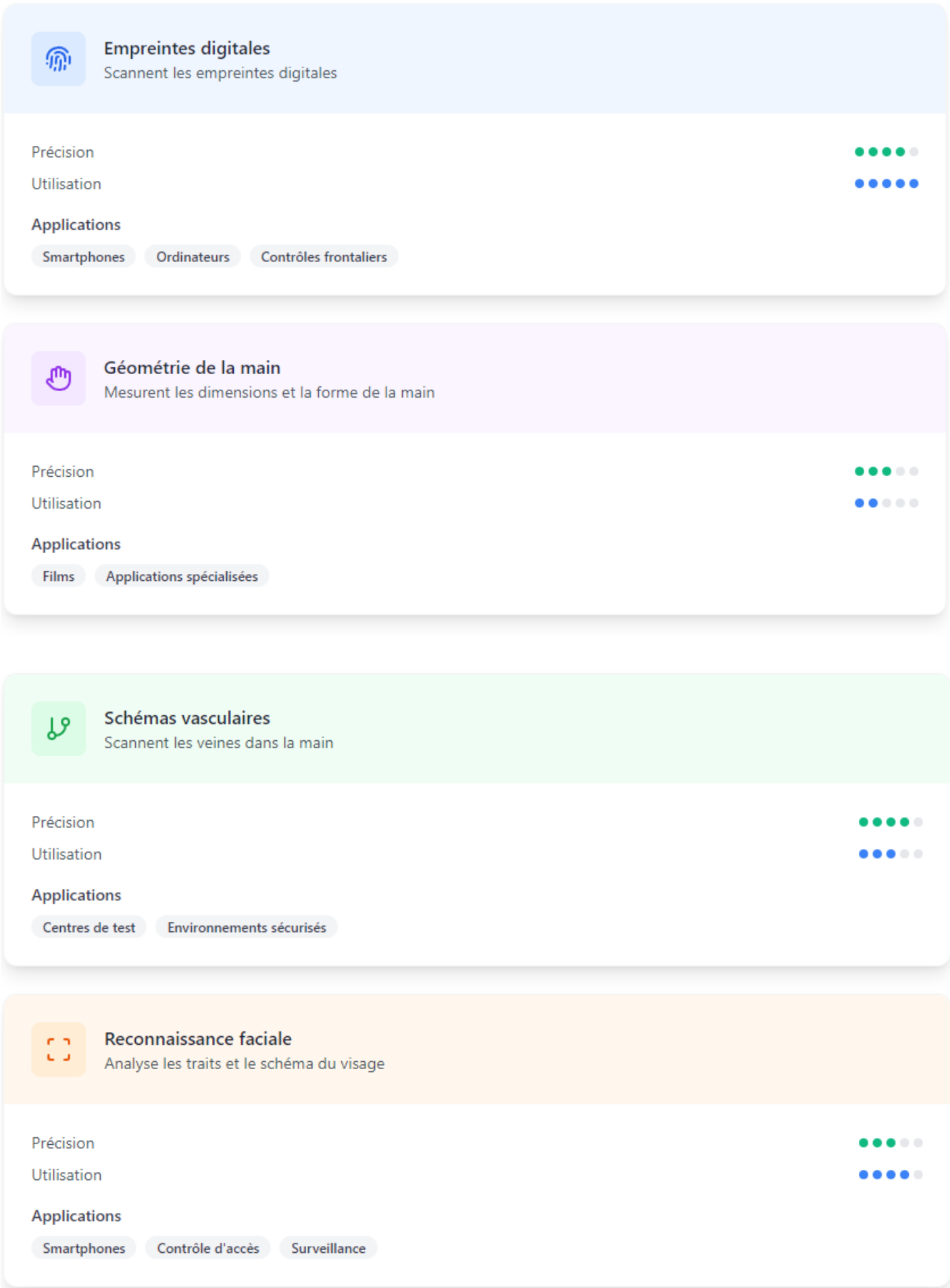
- Les systèmes biométriques ne stockent pas les données brutes ou originales (comme une simple image d'une empreinte digitale).
- À la place, ils utilisent des fonctions mathématiques pour transformer les caractéristiques biométriques en une **représentation numérique unique** appelée **modèle biométrique (template)**.

Utilisation des modèles biométriques :

- Les modèles biométriques sont utilisés de deux manières :
 1. **1 : N (Identification)** :
 - L'utilisateur présente ses données biométriques (ex. : pose son doigt sur un scanner) pour être comparé à une **base de données contenant plusieurs modèles biométriques** afin d'identifier qui il est.
 - Exemple : Déverrouillage d'une porte.
 2. **1 : 1 (Authentification)** :
 - L'utilisateur s'identifie d'abord (ex. : nom d'utilisateur) et son modèle biométrique est comparé **uniquement à son propre modèle** dans la base de données.
 - Exemple : Connexion à un ordinateur portable après avoir tapé un mot de passe.

Types de dispositifs biométriques (Biometric Devices)

Les dispositifs biométriques peuvent être regroupés en deux grandes catégories : physiologiques et comportementaux.





Scans de l'iris
Examens du cercle coloré autour de l'œil

Précision

Utilisation

Applications

Aéroports

Zones hautement sécurisées



Scans de la rétine
Analysent les schémas des veines dans la rétine

Précision

Utilisation

Applications

Installations militaires

Laboratoires sécurisés

Invasif et peut révéler des informations médicales



Échelle de Précision

 Extrêmement précis - Taux d'erreur minimal

 Très précis - Peu d'erreurs

 Précision moyenne - Acceptable pour usage standard

 Précision limitée - Erreurs fréquentes

 Peu précis - Non recommandé seul



Échelle d'Utilisation

 Usage très répandu - Standard de l'industrie

 Largement utilisé - Applications courantes

 Usage modéré - Applications spécifiques

 Usage limité - Cas particuliers

 Rare - Technologies émergentes ou obsolètes



Résumé des dispositifs biométriques et précision

1. Les **scanners de rétine** sont les plus précis, mais aussi les plus controversés en raison de leur caractère intrusif.
2. Les dispositifs **comportementaux** (comme la voix ou la démarche) sont généralement moins précis que les dispositifs physiologiques, mais peuvent être utilisés pour des scénarios spécifiques (par exemple, une authentification passive).

Les modèles biométriques et leurs dispositifs associés doivent être soigneusement choisis en fonction du niveau de sécurité requis, de l'acceptation des utilisateurs et des préoccupations éthiques ou juridiques.

1.2.6 Facteurs d'authentification



- Les facteurs d'authentification se divisent en trois catégories principales :
 1. **Authentification par connaissance** : Quelque chose que vous savez.
 2. **Authentification par possession** : Quelque chose que vous possédez.
 3. **Authentification par caractéristique** : Quelque chose que vous êtes (physiologique ou comportemental).
- **Authentification à facteur unique** :
 - Implique l'utilisation d'un seul de ces trois facteurs.
- **Authentification multi-facteurs (MFA)** :
 - Implique l'utilisation de deux facteurs ou plus issus de différentes catégories.

Différence entre l'authentification à facteur unique et l'authentification multi-facteurs

- Si un système utilise un seul type de facteur d'authentification (par exemple, "quelque chose que vous savez"), même avec plusieurs étapes, c'est une authentification à facteur unique.
- Si un système combine deux ou plusieurs facteurs provenant de différentes catégories, cela devient une authentification multi-facteurs (MFA).

Exemple :

1. Si un utilisateur utilise une clé RSA (possession) et un token Microsoft (possession), cela reste une authentification à facteur unique.
2. Si le processus combine un mot de passe (connaissance) et une clé RSA (possession), cela constitue une authentification multi-facteurs.

Authentification sans mot de passe (Password-less authentication)

Concepts clés

- L'authentification sans mot de passe permet aux utilisateurs de s'authentifier sans avoir à entrer de mot de passe.
- Avantages :
 - Réduit la friction pour les utilisateurs.
 - Limite les risques liés aux mots de passe faibles ou volés.
 - Aide à atténuer les attaques de phishing.

Comment ça fonctionne ?

- Les options d'authentification sans mot de passe incluent :
 1. **Biométrie** (ex. : empreinte digitale, reconnaissance faciale).
 2. **Appareils mobiles** appartenant à l'utilisateur.
 3. **Tokens de sécurité** (ex. : passkeys).

Exemple : Passkeys

- Les utilisateurs s'authentifient avec un PIN ou des données biométriques directement sur leur appareil, qui communique avec le système sécurisé.
- **Avantages** : Plus pratique et résistant au phishing.
- **Inconvénients** :
 - Si un utilisateur perd son appareil ou son token, il peut être verrouillé hors de son compte.
 - Les tokens matériels peuvent avoir des coûts d'implémentation élevés.



À retenir pour cette partie :

- **Facteurs d'authentification** : Divisés en **connaissance**, **possession**, et **caractéristique**.
- **MFA** : Combine deux facteurs ou plus pour renforcer la sécurité.
- **Authentification sans mot de passe** : Réduit les risques liés aux mots de passe et améliore l'expérience utilisateur, mais nécessite une infrastructure sécurisée.

1.2.7 Systèmes de gestion des identifiants (Credential Management Systems)



Les systèmes de gestion des identifiants permettent aux organisations de gérer efficacement, à grande échelle, l'accès aux ressources en s'assurant que :

1. **Les utilisateurs**,
2. **Les processus**,
3. **Les appareils**... disposent d'identifiants uniques.


Ces systèmes sont conçus pour gérer, attribuer et révoquer des identifiants, souvent en utilisant une authentification forte à deux facteurs (2FA) intégrant une infrastructure à clé publique (PKI).

Ils unissent les programmes, processus, technologies et personnels pour créer :

- Des représentations numériques fiables des identités (utilisateurs et entités non humaines comme des processus).
- Une association sécurisée de ces identités à leurs identifiants.

Coffres-forts de mots de passe (Password Vault)






Mot de Passe Principal

La clé unique qui déverrouille votre coffre-fort


Un seul mot de passe à retenir pour accéder à tous vos comptes de manière sécurisée

✓ Avantages



Sécurité Renforcée


Création et gestion de mots de passe forts et uniques, limitant les attaques par réutilisation



Facilité d'Utilisation

Plus besoin de mémoriser de multiples mots de passe complexes

✗ Inconvénients



Point de Défaillance Unique

Si le mot de passe principal est compromis, tous les comptes sont potentiellement exposés

Solution Recommandée :

Utiliser l'authentification multi-facteurs (MFA) en complément pour chaque compte important

Fonctionnement d'un coffre-fort de mots de passe

1. Les mots de passe sont stockés dans une base de données chiffrée, protégée par un mot de passe principal.
2. Les utilisateurs se connectent au coffre-fort avec ce mot de passe principal.
3. Une fois connecté :
 - Le coffre-fort peut remplir automatiquement les champs de mot de passe sur les sites et applications.
 - Les utilisateurs peuvent générer de nouveaux mots de passe forts directement via l'outil.

Résumé

Les **systèmes de gestion des identifiants**, comme les coffres-forts de mots de passe, offrent une solution efficace pour protéger les identités et les comptes en ligne. Ils facilitent la création de mots de passe forts et uniques, réduisant ainsi les risques de compromission. Toutefois, leur **fiabilité dépend du mot de passe principal** et d'une mise en œuvre sécurisée (notamment avec le MFA).

1.2.8 Single Sign-On (SSO)

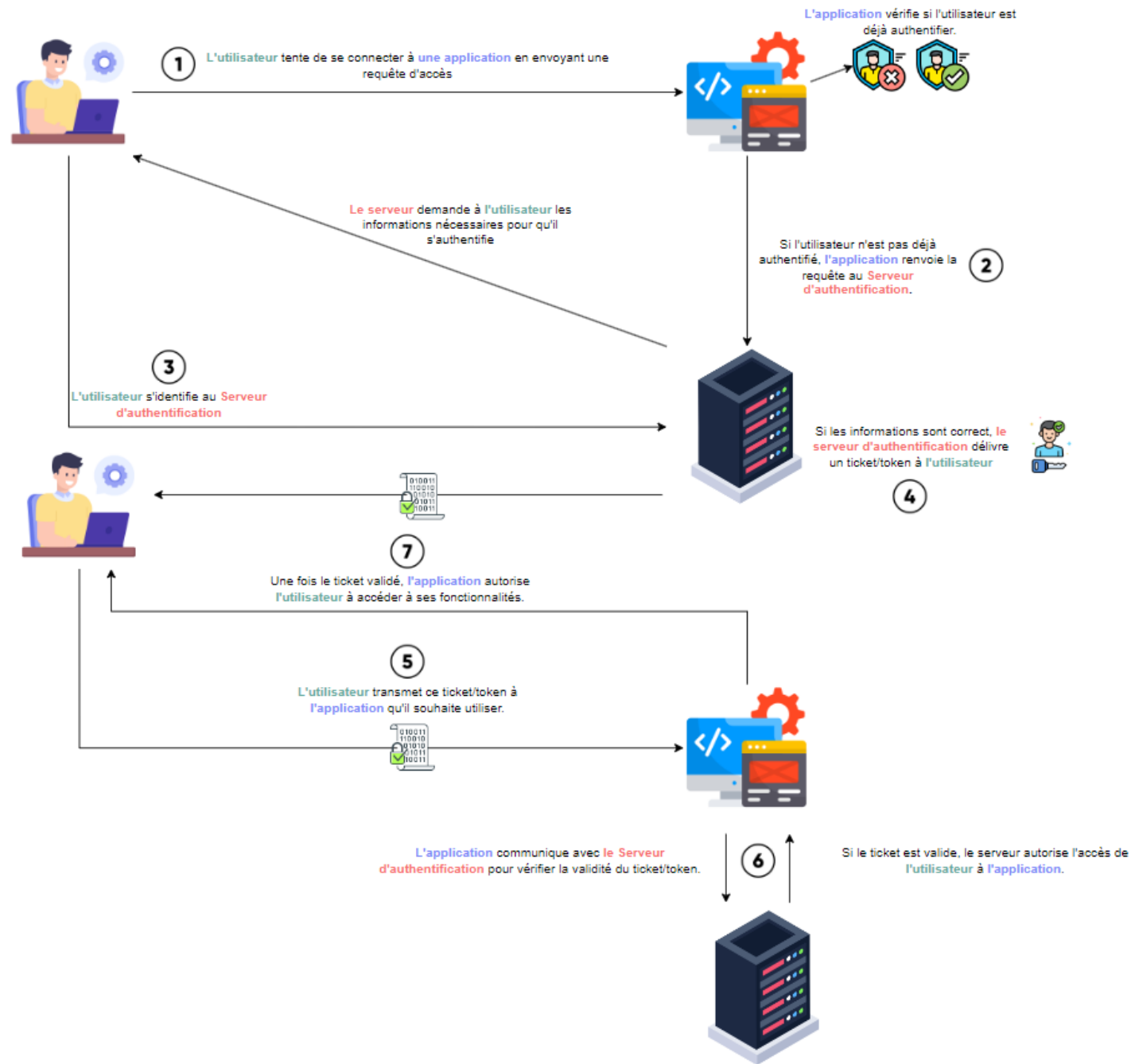
Le **Single Sign-On (SSO)** permet à un utilisateur de s'authentifier une seule fois pour accéder à plusieurs systèmes ou applications sans avoir à se reconnecter à chaque fois.

- **Avantage principal** : Simplifie l'expérience utilisateur, encourage l'utilisation de mots de passe plus forts et réduit les multiples saisies d'identifiants.
- **Inconvénient majeur** : Crée un **point de défaillance unique**, car si le système SSO est compromis, un attaquant pourrait accéder à toutes les ressources autorisées pour l'utilisateur.

CISSP Domaine 5 : Gestion des accès et des identités.

28

Processus de SSO



1. L'utilisateur demande l'accès à une application

- **Action :** L'utilisateur (appelons-le *Client*) tente de se connecter à une application en envoyant une requête d'accès.
- **Direction :**
 - La requête part du *Client* vers l'*Application*.
 - Exemple : L'utilisateur clique sur un lien pour accéder à un service en ligne.

2. L'application redirige vers le serveur d'authentification

- **Action :** Si l'utilisateur n'est pas déjà authentifié, l'application renvoie la requête au **Serveur d'authentification**.
- **Direction :**
 - L'Application redirige la requête vers le **Serveur d'authentification (Auth Server)**.
- **Pourquoi ? :** L'application ne gère pas l'authentification elle-même ; elle délègue cette tâche au serveur SSO.

3. L'utilisateur s'authentifie auprès du serveur d'authentification

- **Action :** L'utilisateur s'identifie au **Serveur d'authentification**, en fournissant :
 - Un mot de passe,
 - Un code MFA (exemple : un OTP généré par une application),
 - Ou des données biométriques (empreinte digitale, reconnaissance faciale, etc.).
- **Résultat :**
 - Si les informations sont correctes, le serveur génère un **ticket ou un token**.

4. Le serveur d'authentification délivre un ticket/token

- **Action :** Le **Serveur d'authentification** envoie un **ticket d'authentification** (ou **token**) à l'utilisateur.
- **Contenu du ticket/token :**
 - Le ticket contient des informations chiffrées confirmant que l'utilisateur a été authentifié.
 - Il peut inclure un délai d'expiration.

5. Le ticket/token est présenté à l'application

- **Action :** L'utilisateur transmet ce **ticket/token** à l'application qu'il souhaite utiliser.
- **Direction :**
 - Le ticket est envoyé du *Client* vers l'*Application*.
- **Pourquoi ? :** L'application utilise ce ticket pour valider l'accès auprès du serveur.

6. L'application vérifie le ticket/token auprès du serveur

- **Action :** L'application communique avec le **Serveur d'authentification** pour vérifier la validité du ticket/token.
- **Résultat :**
 - Si le ticket est valide, le serveur autorise l'accès de l'utilisateur à l'application.

7. Accès accordé à l'application

- **Action :** Une fois le ticket validé, l'application autorise l'utilisateur à accéder à ses fonctionnalités.
- **Résultat :** L'utilisateur peut désormais naviguer dans l'application sans avoir à se reconnecter.

Optionnel : Réutilisation du ticket pour d'autres applications

- **Action :** Si l'utilisateur souhaite accéder à une autre application (intégrée au SSO), il peut réutiliser le même **ticket/token**, sans avoir à s'authentifier de nouveau.
- **Résultat :** Cela réduit la friction pour l'utilisateur et lui permet d'accéder à plusieurs systèmes après une seule authentification.

Avantages et inconvénients du SSO

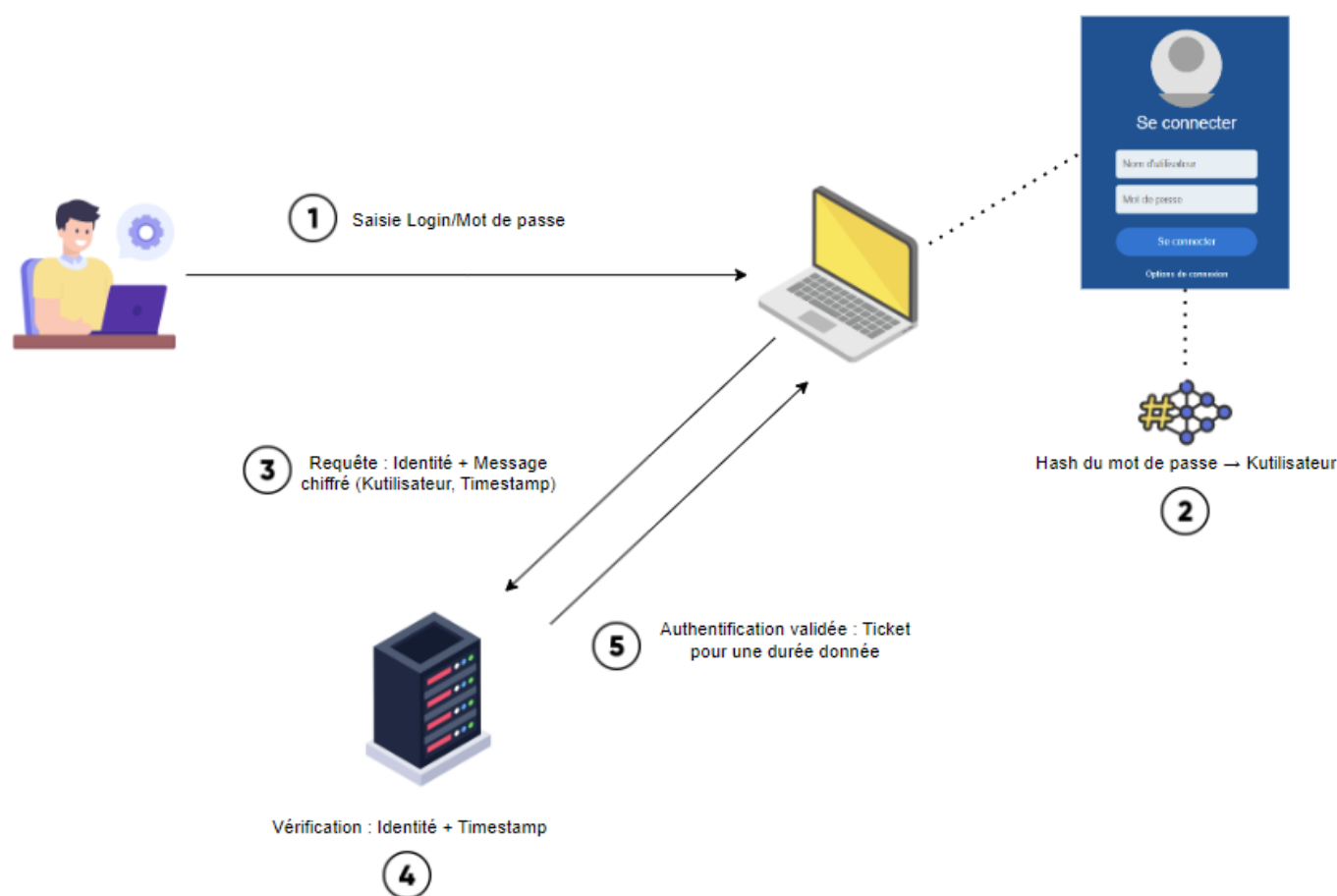
Avantages	Inconvénients
Améliore l'expérience utilisateur.	Crée un point de défaillance unique .
Encourage l'utilisation de mots de passe plus forts.	En cas de panne, les utilisateurs perdent l'accès à tout.
Administration centralisée des identifiants.	Difficulté d'intégration avec des systèmes anciens ou non standardisés.

Kerberos : son fonctionnement et ses spécificités

L'un des principaux protocoles d'authentification SSO est Kerberos. En référence à la mythologie grecque, le nom Kerberos provient du chien à trois têtes, Cerbère, qui gardait les portes des Enfers. Inspiré de ce mythe, Kerberos protège l'accès aux ressources et offre trois fonctionnalités principales :

- Comptabilité (Accounting)
- Authentification (Authentication)
- Audit (Auditing)

Kerberos est un protocole ancien et complexe. Heureusement, il n'est pas nécessaire d'être un expert pour comprendre son fonctionnement. Le schéma ci-dessous vous offre une vue simplifiée des étapes clés du processus d'authentification Kerberos.



1. Interaction : Utilisateur → Client Kerberos

L'utilisateur commence par saisir son **Login/Mot de passe** sur sa machine.

- **Objectif** : Transmettre les informations nécessaires pour lancer le processus d'authentification.
- **Résultat** : Le client Kerberos récupère les identifiants de l'utilisateur pour les traiter.

2. Interaction : Client Kerberos → Algorithme de hachage

Le **Client Kerberos** utilise un **algorithme de hachage** pour transformer le mot de passe de l'utilisateur en une **clé secrète Kutilisateur**.

- **Objectif** : Protéger le mot de passe en évitant de le transmettre directement. À la place, seule une clé dérivée est utilisée.
- **Résultat** : La clé secrète **Kutilisateur** est prête pour chiffrer les échanges avec le serveur.

3. Interaction : Client Kerberos → Serveur d'Authentification (AS)

Le client Kerberos envoie une **requête d'authentification** au **Serveur d'Authentification (AS)**. Cette requête contient :

- **L'identité de l'utilisateur** (transmise en clair pour l'identifier).
- **Un message chiffré avec Kutilisateur** contenant un **timestamp**.
- **Objectif** :
 - L'identité informe le serveur sur l'utilisateur qui tente de se connecter.
 - Le **chiffrement** avec Kutilisateur sécurise le message.
 - Le **timestamp** empêche les attaques par rejeu en garantissant que la requête est récente.

Résultat : Le Serveur AS reçoit les informations nécessaires pour valider l'identité de l'utilisateur.

4. Interaction : Serveur d'Authentification (AS) → Validation interne

Le **Serveur AS** effectue les vérifications suivantes :

- Il **déchiffre** le message reçu avec la clé secrète Kutilisateur (qui doit correspondre à celle dérivée du mot de passe).
- Il vérifie :
 - **L'identité de l'utilisateur** pour confirmer qu'elle est correcte.
 - Le **timestamp** pour s'assurer que la requête est légitime et récente (ce qui protège contre les attaques par rejeu).
- **Objectif** : Garantir que l'utilisateur possède bien le mot de passe correct sans jamais le transmettre en clair.
- **Résultat** : Si ces vérifications réussissent, l'utilisateur est authentifié.

5. Interaction : Serveur d'Authentification (AS) → Client Kerberos

Une fois l'identité de l'utilisateur validée, le **Serveur AS** délivre un **ticket d'authentification** au **Client Kerberos**.

- **Objectif** : Le ticket prouve que l'authentification a réussi et permet à l'utilisateur d'accéder aux services sans avoir besoin de se reconnecter pendant une **durée prédéfinie**.
- **Résultat** : Le Client Kerberos reçoit le ticket d'authentification et peut l'utiliser pour accéder aux ressources du système.

<https://blog.devensys.com/2018/07/18/kerberos-principe-de-fonctionnement/>

SESAME

Le Système Européen Sécurisé pour Applications dans un Environnement Multi-Vendeur, mieux connu sous le nom de SESAME, est une version améliorée de Kerberos. Tout comme Kerberos, SESAME est un protocole permettant la mise en place du Single Sign-On (authentification unique).

De plus, l'un des principaux avantages de SESAME par rapport à Kerberos est qu'il prend en charge la cryptographie symétrique et asymétrique, ce qui résout naturellement le problème de distribution des clés. Il délivre également plusieurs tickets, ce qui atténue les vulnérabilités aux attaques TOCTOU (*Time Of Check To Time Of Use*).

Bien que SESAME soit un meilleur protocole, Kerberos est de loin plus répandu, car il est intégré à de nombreux systèmes courants, notamment :

- Les systèmes d'exploitation Windows,
- MacOS,
- Et diverses distributions Linux et Unix.

Il est important de noter que pour utiliser Kerberos dans un environnement Windows, Active Directory doit être activé.

1.2.9 CAPTCHA



- CAPTCHA est une mesure de sécurité qui fonctionne en demandant à un utilisateur généralement un visiteur d'un site web ou portail de réaliser un test simple pour prouver qu'il est humain et non un robot ou un programme automatisé.
- CAPTCHA est utilisé pour prévenir la création automatique de comptes, le spam et les attaques par force brute visant à décrypter des mots de passe.

Comprendre ce qu'est un CAPTCHA et pourquoi il est couramment utilisé

Lorsqu'un utilisateur accède à un site web, les fournisseurs utilisent souvent ce qu'on appelle un test CAPTCHA (*Completely Automated Public Turing Test to tell Computers and Humans Apart*) comme mesure de sécurité pour protéger contre :

1. La création automatisée de comptes.
2. Les attaques de spam.
3. Les attaques par force brute visant à déchiffrer des mots de passe.

Fonctionnement du CAPTCHA :

- Le CAPTCHA demande à l'utilisateur de compléter un test simple pour prouver qu'il est humain et non un programme automatisé tentant d'accéder ou de pirater un compte ou une zone protégée.
- Sous sa forme la plus simple, le CAPTCHA affiche une image contenant des lettres et des chiffres déformés.
 - L'utilisateur doit saisir ces lettres et chiffres dans un champ prévu à cet effet.
 - Si les informations saisies sont correctes, l'utilisateur obtient l'accès à la zone protégée.
 - En cas d'erreur, l'utilisateur a généralement une seconde chance pour réaliser le test.



En résumé :

Le CAPTCHA est utilisé pour :

- Empêcher les robots de créer des comptes multiples sur un système.
- Réduire le spam.
- Éviter les accès non autorisés à des zones protégées.

1.2.10 Gestion de session



La gestion de session fait référence à la création, à la gestion et à la sécurisation des sessions utilisateur. Une session est créée après une authentification réussie d'un utilisateur, représentant la connexion et l'interaction entre cet utilisateur et le système.

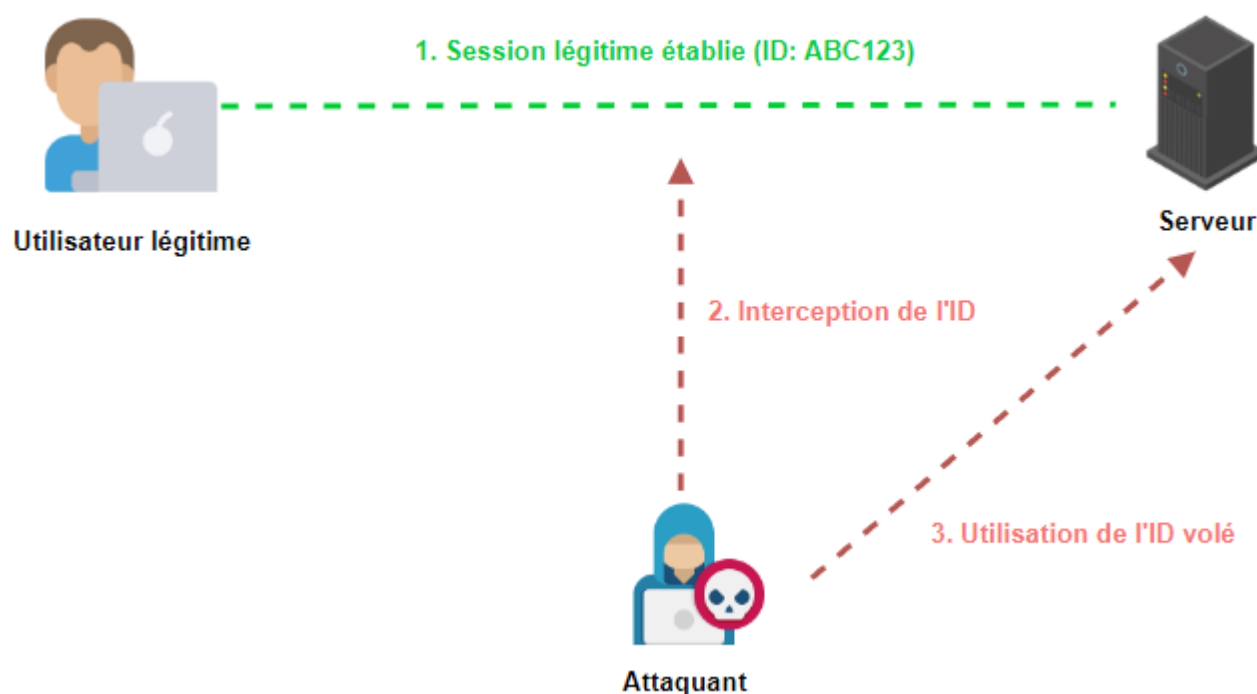
- Étapes clés :
 1. L'utilisateur fournit des identifiants valides.
 2. L'utilisateur est authentifié et autorisé par le système.
 3. Une session est créée et reste active jusqu'à sa terminaison manuelle ou automatique.

Objectif : Assurer la gestion efficace et sécurisée des sessions pendant toute leur durée.

Détournement de session (Session Hijacking)

Qu'est-ce que le détournement de session ?

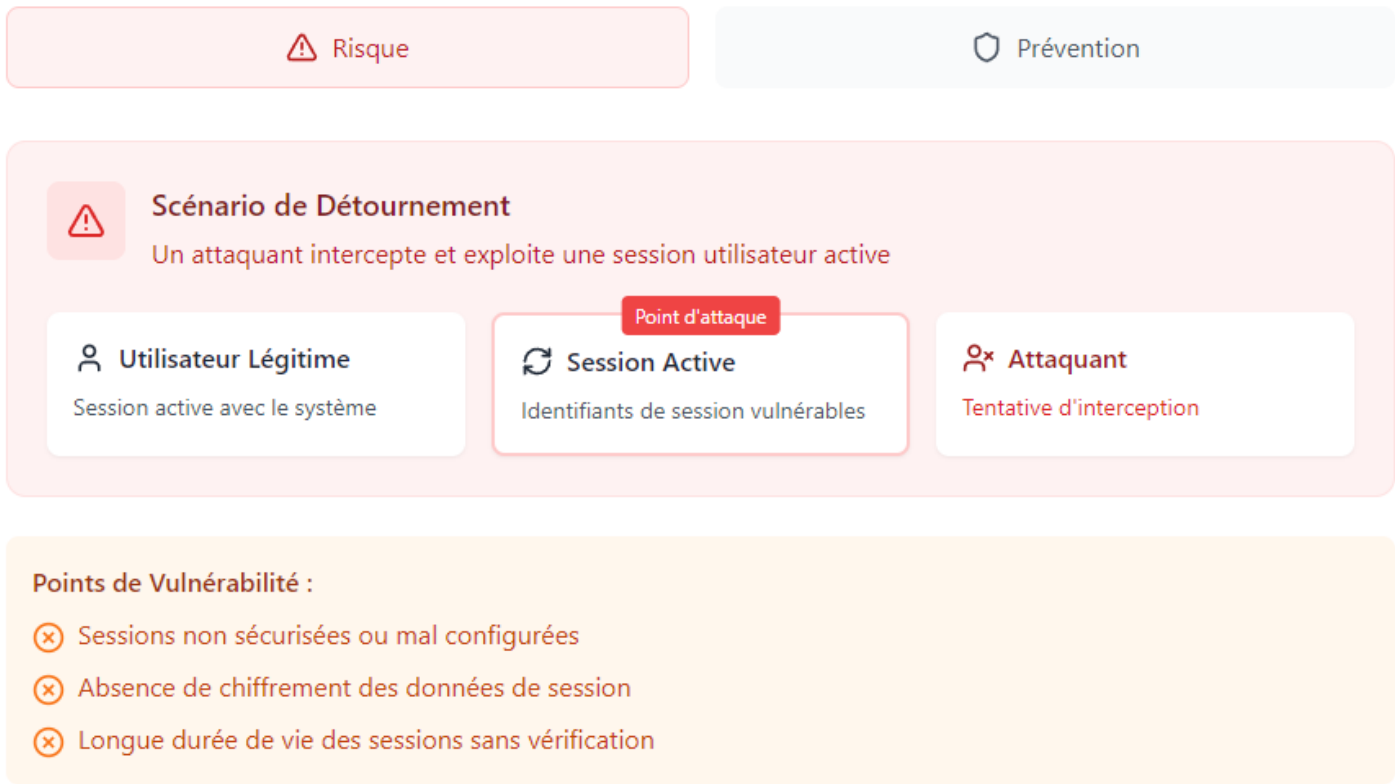
- Le détournement de session est un risque majeur lorsqu'une gestion de session appropriée n'est pas mise en place.
- Cela se produit lorsqu'une personne non autorisée (comme un attaquant) obtient l'accès à une session active d'un utilisateur et l'exploite à des fins malveillantes.



Exemple :

- Un attaquant utilise des méthodes techniques ou des failles (comme un manque de sécurisation) pour intercepter ou usurper une session active.

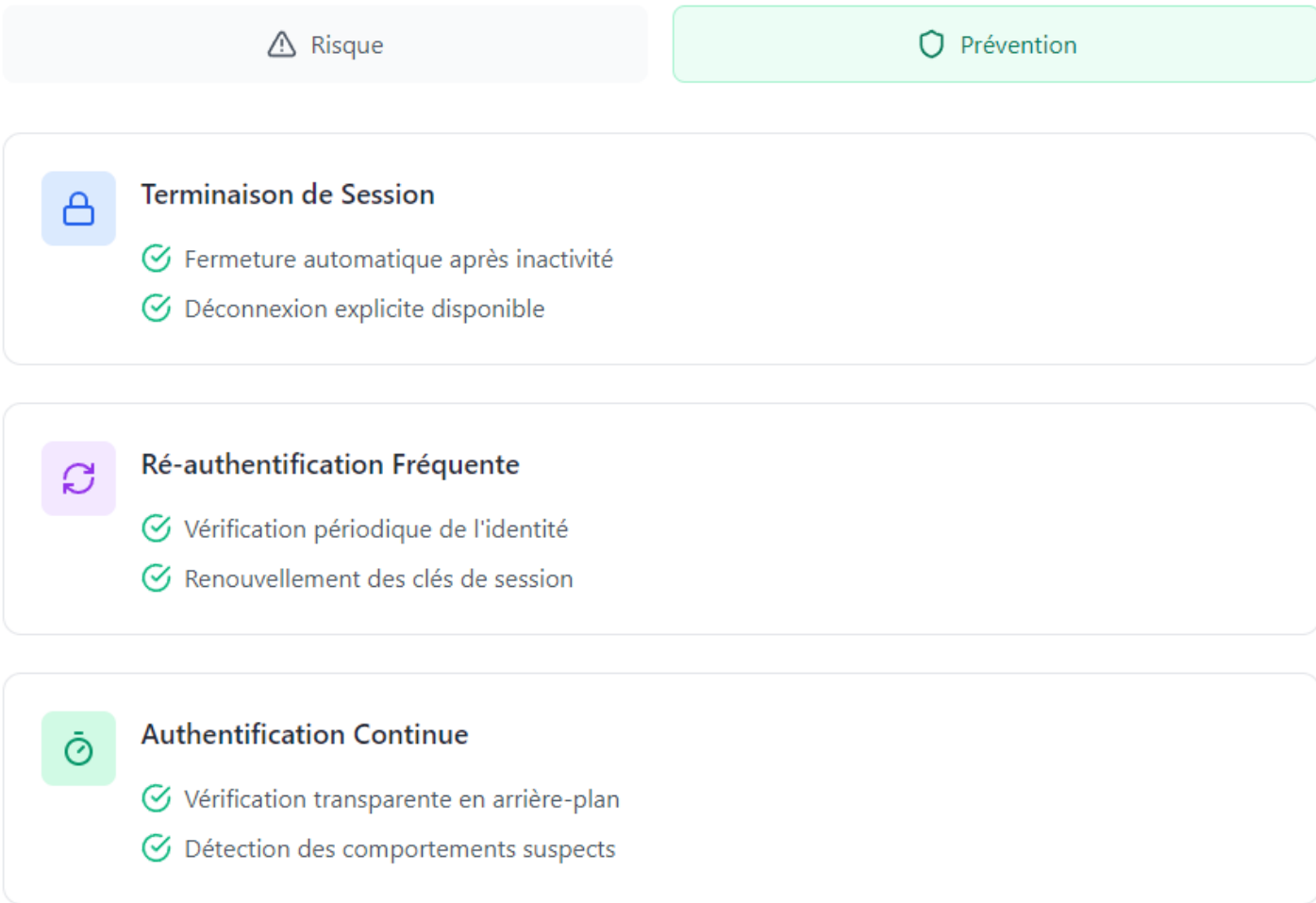
Détournement de Session (Session Hijacking)



Comment prévenir le détournement de session ?

- **Terminaison de session** : Une session doit être fermée correctement après utilisation pour réduire les risques.
- **Ré-authentification fréquente** :
 - Les systèmes doivent exiger une ré-authentification régulière pour s'assurer que l'utilisateur actuel est légitime.
 - Par exemple, les solutions VPN utilisent des clés de chiffrement de session qui sont renouvelées périodiquement en arrière-plan.
- **Authentification continue** : Le système peut ré-authentifier l'utilisateur de manière transparente, rendant difficile l'exploitation d'une session active.

Détournement de Session (Session Hijacking)



Terminaison de session (Session Termination)

En plus de la ré-authentification continue, il existe plusieurs méthodes pour terminer une session :

Méthodes de terminaison de session

1. Limites de planification (Schedule Limitations) :

- Contrôle administratif permettant de déconnecter les utilisateurs automatiquement à un moment défini, par exemple :
 - Tous les jours à 17h.
 - Pendant le week-end, si les connexions ne sont pas autorisées.

2. Limitation des connexions simultanées (Login Limitation) :

- Empêche plusieurs connexions simultanées avec le même identifiant utilisateur.
- But : Éviter le partage de comptes ou leur utilisation non autorisée par d'autres personnes.

3. Time-outs :

- Si aucune activité n'est détectée pendant un laps de temps défini, la session expire automatiquement (timeout).
- Cela permet de réduire les risques liés aux sessions inactives.

4. Économiseurs d'écran (Screensavers) :

- Lorsque l'économiseur d'écran apparaît après une période d'inactivité, l'accès à la session est bloqué.
- Pour reprendre l'accès, une ré-authentification est requise.



Résumé des points clés :

- **Gestion de session** : Création et maintien d'une session utilisateur après authentification.
- **Détournement de session** : Exploitation d'une session active par un attaquant.
 - **Prévention** : Terminaison de session, ré-authentification continue et sécurisation des clés de session.
- **Terminaison de session** : Utilisation de **timeouts**, limitations de connexions et de planification pour protéger contre les sessions non sécurisées ou inactives.

Ces mesures garantissent une sécurité renforcée contre les accès non autorisés et les attaques de type session hijacking.

1.2.11 Enregistrement et vérification d'identité



La vérification d'identité (ou enregistrement) est le processus qui consiste à confirmer ou établir qu'une personne est bien celle qu'elle prétend être.

La vérification d'identité est une composante essentielle du cycle de vie de l'identité, notamment lors de la phase de provisionnement.

Qu'est-ce que la vérification d'identité et quand intervient-elle ?

La vérification d'identité, parfois appelée enregistrement, est le processus par lequel une organisation confirme l'identité d'une personne avant de lui accorder l'accès à des ressources précieuses ou des actifs critiques.

Exemple :

- Avant qu'une autorité de certification (CA) délivre un certificat numérique, elle vérifie l'identité du demandeur.

Rôle de l'Autorité d'Enregistrement (RA)

- L'autorité d'enregistrement (RA) est responsable de vérifier l'identité des candidats.
- Application pratique :
 - Lorsqu'une personne commence un emploi, l'organisation vérifie son identité avant de lui fournir :
 - Un badge d'employé,
 - Des identifiants de compte,
 - Ou tout autre accès aux systèmes internes.

Méthodes de vérification d'identité



Pièce d'Identité Officielle

- ✓ Délivrée par l'État
- ✓ Photo officielle
- ✓ Haute sécurité



Permis de Conduire

- ✓ Document officiel
- ✓ Photo récente
- ✓ Validité vérifiable



Autres Identifications

- ✓ Passeport
- ✓ Carte professionnelle
- ✓ Documents spécifiques



En résumé :

La vérification d'identité garantit que la personne est

réellement celle qu'elle prétend être avant de lui accorder un accès ou un privilège. Cela constitue une **mesure de sécurité fondamentale** dans la gestion des identités.

1.2.12 Niveaux d'Assurances des Authentificateurs

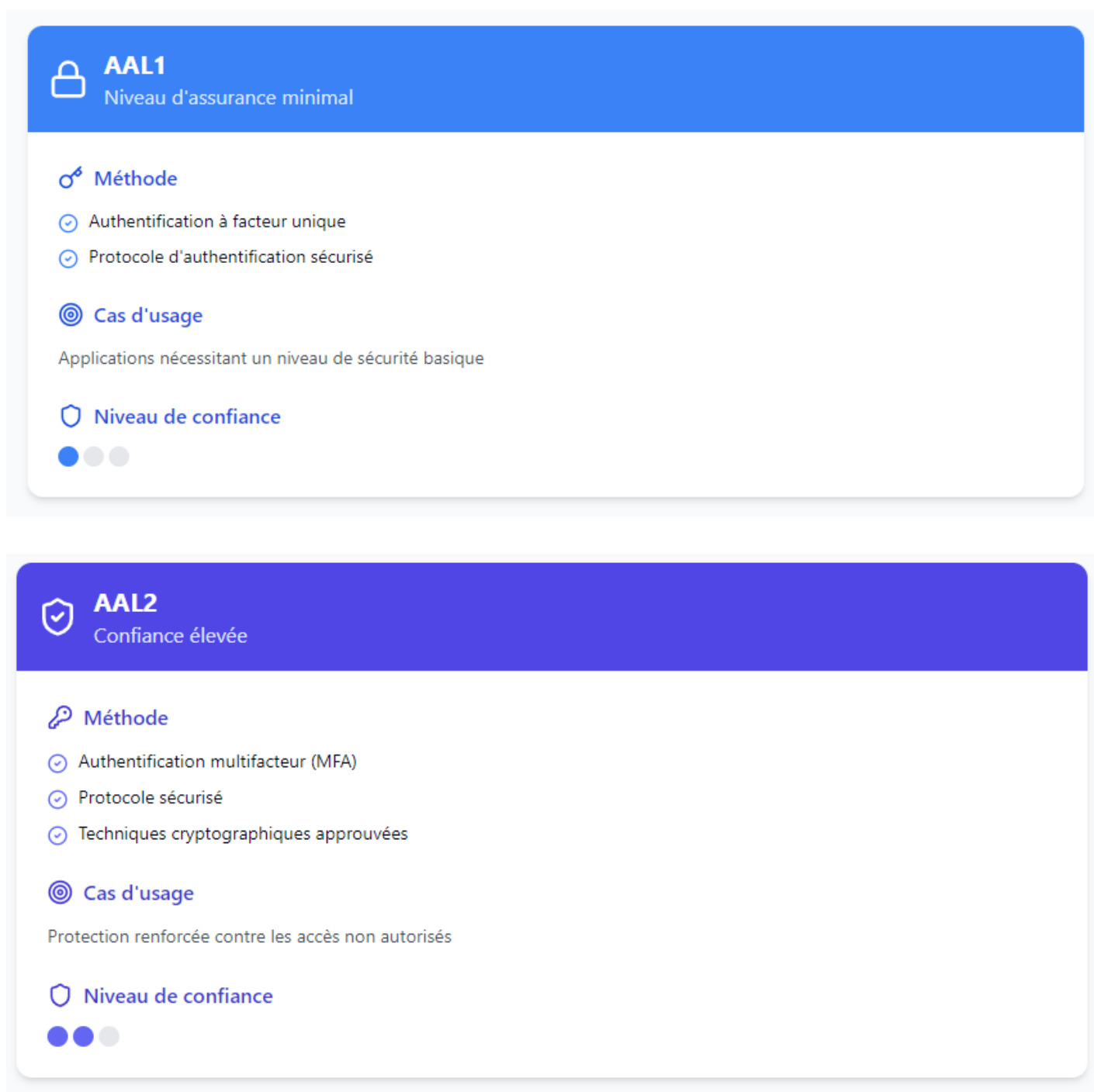



Les Niveaux d'Assurance des Authentificateurs (AAL) font référence à la robustesse des processus et systèmes d'authentification.


Les niveaux AAL vont de AAL1 (le moins robuste) à AAL3 (le plus robuste).

Comprendre les niveaux AAL et leurs éléments


Les niveaux AAL définissent la **force** et la **sécurité** des méthodes d'authentification. Plus le niveau est élevé, plus l'authentification est robuste et fiable.




**AAL3**
Très haute confiance


 **Méthode**

- ✓ MFA avec exigences strictes
- ✓ Authenticateur cryptographique "difficile"
- ✓ Protection contre l'usurpation d'identité


 **Cas d'usage**

Systemes critiques necessitant une securite maximale

 **Niveau de confiance**

 Le choix du niveau AAL depend des exigences de securite specifiques et des risques associes a l'application ou au systeme.

Niveau	Assurance	Methodes utilisees
AAL1	Faible	Authentication a facteur unique, protocole securise.
AAL2	Elevee	Authentication multifacteur, cryptographie approuvee.
AAL3	Très elevee	Authentication multifacteur stricte, authenticateur cryptographique "dur".

 En resume :

Les AAL permettent d’évaluer et de définir la robustesse des processus d'authentification en fonction des besoins de sécurité. AAL1 convient à des scénarios moins critiques, tandis qu'AAL3 est utilisé pour des environnements nécessitant une sécurité maximale.

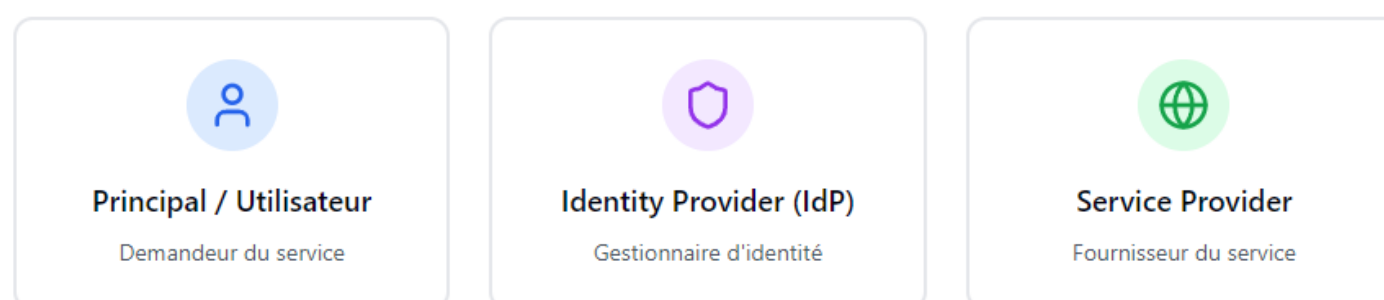
1.2.13 Gestion Fédérée des Identités (Federated Identity Management - FIM)



La Gestion Fédérée des Identités (FIM) permet d'établir une authentification unique (Single Sign-On - SSO) sur plusieurs systèmes appartenant à des entités différentes. Elle repose sur une relation de confiance entre les organisations, où l'une d'elles effectue l'authentification et les autres acceptent les résultats de cette authentification.

- **Authentification unique (Single Sign-On - SSO) :**
 - **Définition :** L'utilisateur s'authentifie une seule fois pour accéder à plusieurs systèmes.
 - **Limite :** Dans un contexte traditionnel, l'authentification unique fonctionne **au sein d'une seule organisation**.
- **Gestion Fédérée des Identités (FIM) :**
 - **Définition :** Étend le concept de SSO au-delà des frontières d'une organisation pour inclure des systèmes d'autres entités (par exemple, d'autres entreprises ou services).
 - **Objectif :** Permettre un accès transparent et sécurisé aux utilisateurs tout en **conservant l'identité unique** dans des systèmes hétérogènes.
- **Relations de confiance :**
 - FIM repose sur des **accords de confiance** entre plusieurs entités qui acceptent les authentifications réalisées par un tiers de confiance (Identity Provider).

Les trois composantes de la FIM



1. Le Principal / Utilisateur :

- **Définition :** La personne qui souhaite accéder à un service ou à un système.
- **Rôle :** L'utilisateur fournit ses informations d'identification à un **Identity Provider** pour être authentifié.
- **Exemple :** Un utilisateur qui se connecte à Pinterest via son compte Google.

2. L'Identity Provider (IdP) :

- **Définition :** L'entité qui **détient l'identité** et réalise l'**authentification** de l'utilisateur.
- **Rôle :**
 - Vérifier l'identité de l'utilisateur.
 - Fournir une **preuve d'authentification** (comme un token sécurisé) à l'entité qui en a besoin.
- **Exemple :** Google est l'Identity Provider lorsque l'utilisateur se connecte à Pinterest avec son compte Google.

3. Le Service Provider (Relying Party) :

- **Définition :** L'entité qui fournit le service demandé par l'utilisateur et **fait confiance** à l'Identity Provider pour l'authentification.
- **Rôle :**
 - Accepter les **preuves d'authentification** fournies par l'Identity Provider.
 - Autoriser l'accès à ses services sur la base de cette authentification.

- **Exemple** : Pinterest est le Service Provider dans l'exemple précédent.

Exemple pratique de la FIM dans le monde réel

Exemple d'un voyage aérien

1. Un voyageur passe par un **contrôle de sécurité** à l'aéroport d'origine.
2. Lorsqu'il arrive dans un autre aéroport, il n'a pas besoin de repasser par un contrôle complet, car le **nouvel aéroport fait confiance** au contrôle effectué par le premier aéroport.
3. Cela représente une **relation de confiance fédérée** entre deux entités distinctes (les aéroports).

Exemple dans le monde numérique


- Un utilisateur souhaite créer un compte sur **Pinterest**, mais préfère utiliser son compte **Google** existant :
 1. L'utilisateur sélectionne l'option "**Se connecter via Google**" sur Pinterest.
 2. Une fenêtre s'ouvre, demandant les identifiants Google de l'utilisateur.
 3. **Google authentifie** l'utilisateur et confirme cette authentification à Pinterest.
 4. **Pinterest** accepte l'authentification effectuée par Google et permet à l'utilisateur d'accéder à ses services.

Explication :

- Google joue le rôle de **Identity Provider (IdP)**.
- Pinterest joue le rôle de **Service Provider (Relying Party)**.
- L'utilisateur est le **Principal** qui souhaite accéder au service.
- **Confiance fédérée** : Pinterest **fait confiance** à Google pour l'authentification de l'utilisateur.


Avantages de la Gestion Fédérée des Identités

Une solution moderne pour une gestion efficace des identités


Expérience utilisateur simplifiée
 Authentification unique pour plusieurs services

- ✓ Single Sign-On (SSO) pour tous les services
- ✓ Réduction du nombre d'identifiants à mémoriser
- ✓ Processus de connexion simplifié

Exemple : Un utilisateur se connecte une fois avec son compte Google pour accéder à plusieurs applications


Réduction des coûts administratifs
 Gestion centralisée et efficace

- ✓ Moins de ressources pour la gestion des comptes
- ✓ Réduction du support technique
- ✓ Automatisation des processus d'identité

Exemple : Une entreprise délègue l'authentification à Microsoft Azure AD plutôt que de gérer ses propres systèmes



👉 La Gestion Fédérée des Identités (FIM) repose sur des relations de confiance établies entre trois entités clés :

1. L'utilisateur (Principal),
2. L'Identity Provider (qui authentifie l'utilisateur),
3. Le Service Provider (qui fait confiance à l'Identity Provider).

Elle étend le principe de Single Sign-On (SSO) au-delà des limites d'une organisation pour permettre un accès transparent et sécurisé à des systèmes appartenant à plusieurs entités distinctes. Cela améliore à la fois l'expérience utilisateur et la sécurité globale.

1.2.14 Les Normes d'Accès Fédéré



Les protocoles clés d'accès fédéré incluent :

- Security Assertion Markup Language (SAML),
- WS-Federation,
- OpenID (pour l'authentification),
- OAuth (pour l'autorisation).

SAML est fréquemment utilisé dans les solutions FIM (Federated Identity Management) et fournit des fonctionnalités d'authentification et d'autorisation.

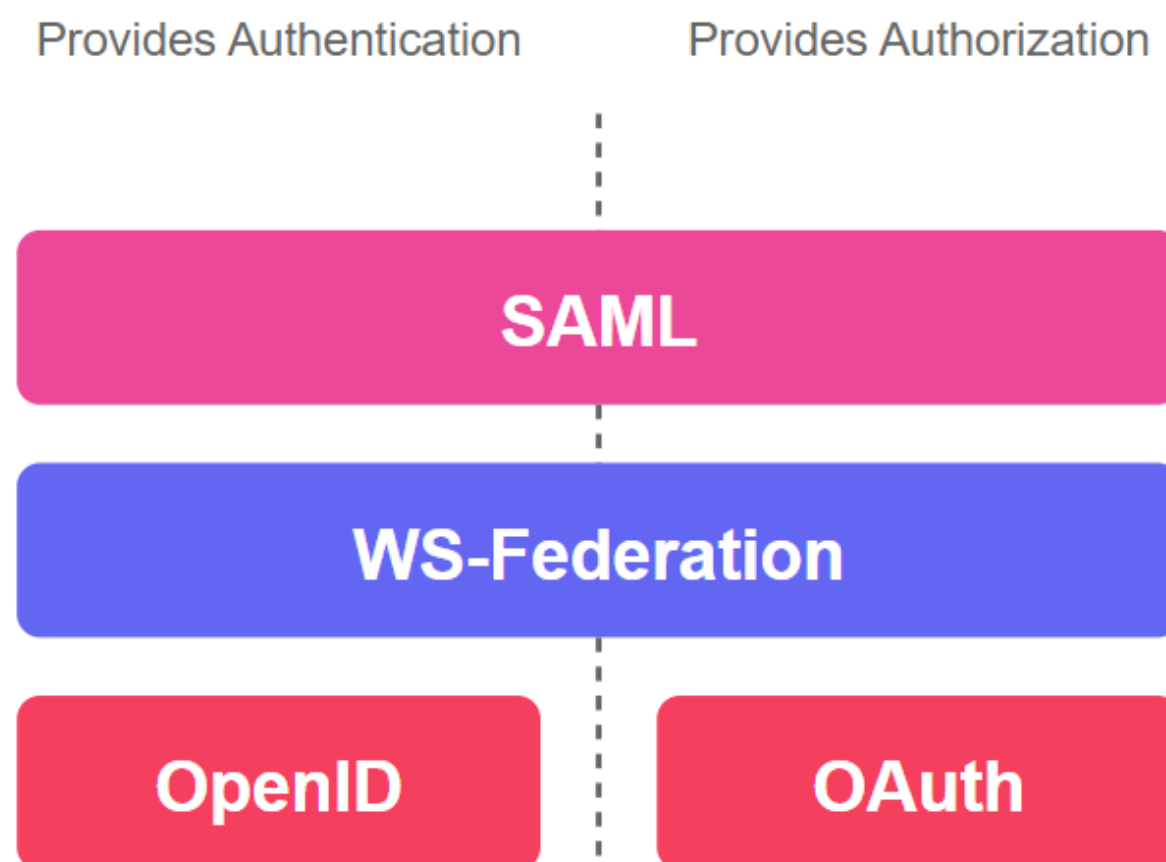
OpenID et OAuth sont des protocoles fédérés ouverts qui fournissent :

- Authentification via OpenID,
- Autorisation via OAuth.

Les assertions SAML sont écrites dans un langage appelé XML (Extensible Markup Language).

XML est une méthode de communication à la fois lisible par les machines et par les humains.

Plusieurs protocoles majeurs permettent l'accès fédéré, avec SAML comme norme clé. D'autres normes incluent WS-Federation, OpenID, et OAuth, qu'il est important de connaître.



WS-Federation

- À l'instar de SAML, WS-Federation offre des fonctionnalités d'authentification et d'autorisation.
- Objectif principal : Permettre la fédération des identités pour l'authentification et l'autorisation.
- Origine : WS-Federation a été créé par un consortium d'entreprises, incluant IBM, Microsoft et Verisign, puis standardisé par OASIS.

OpenID et OAuth

- OpenID et OAuth sont des protocoles complémentaires souvent utilisés ensemble :
 - OpenID fournit la composante d'authentification.

- OAuth fournit la composante d'autorisation.

Exemple :

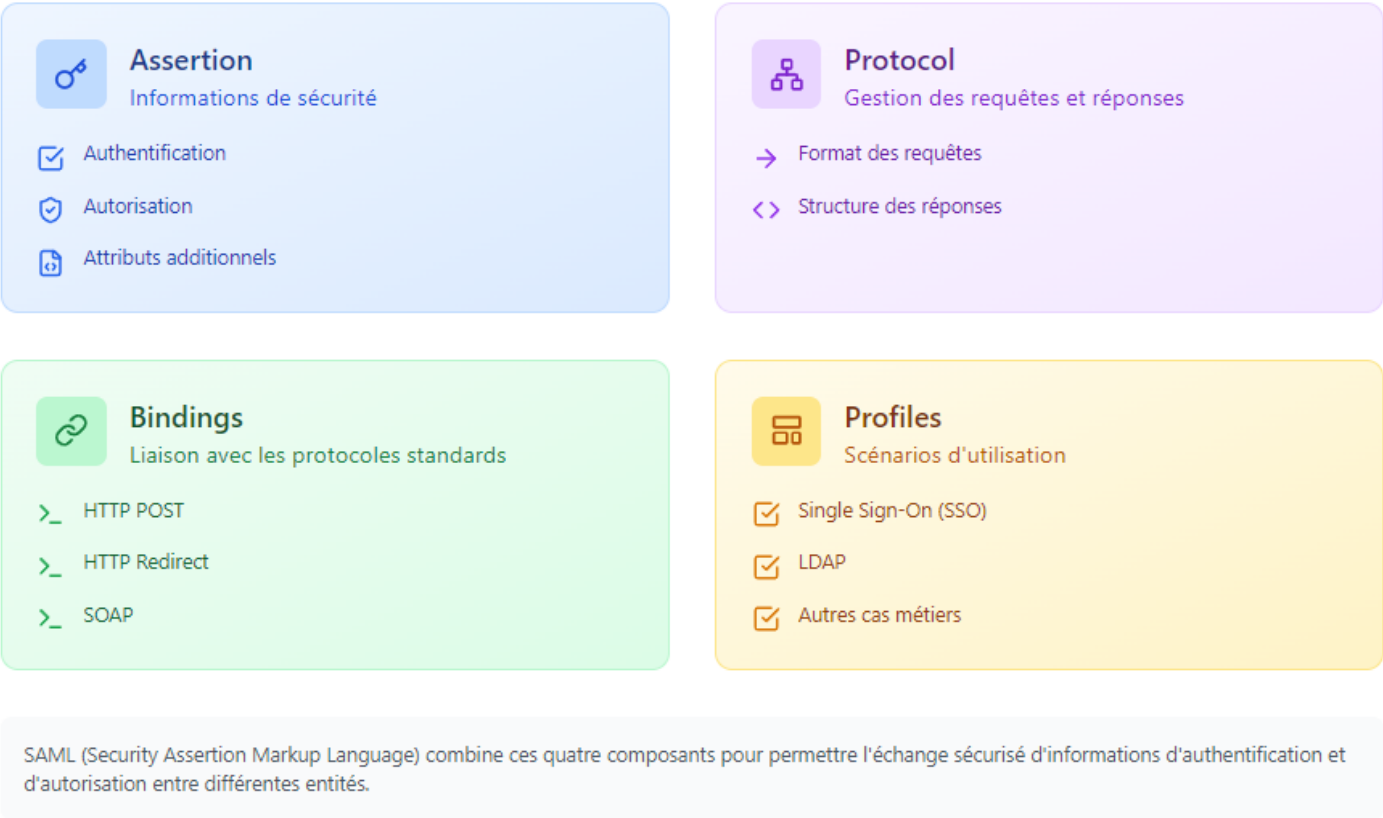
- OpenID permet à un utilisateur d'utiliser un compte existant (ex. compte Microsoft) pour s'authentifier sur plusieurs services, sans créer de nouveaux mots de passe.
- OAuth permet à l'utilisateur d'être autorisé à accéder à des ressources.
- Point clé : Bien qu'ils puissent fonctionner indépendamment, OpenID et OAuth sont souvent déployés ensemble pour offrir des fonctionnalités complètes.

Security Assertion Markup Language (SAML)

Fonctionnement de SAML

- 1. Authentification de l'utilisateur :**
 - L'utilisateur (principal) doit d'abord s'authentifier via l'Identity Provider (IdP).
 - Si l'utilisateur n'est pas connecté et demande l'accès à un service, la requête est redirigée vers l'Identity Provider.
- 2. Processus d'authentification via l'Identity Provider :**
 - L'Identity Provider effectue l'authentification et génère un ticket d'assertion SAML.
 - Note importante : Le ticket SAML ne contient pas le nom d'utilisateur ni le mot de passe.
 - Ce ticket contient des déclarations (assertions) sur l'utilisateur, telles que :
 - Nom d'utilisateur,
 - Rôle,
 - Niveau d'accès.
- 3. Transfert du ticket au Service Provider :**
 - Une fois le ticket SAML délivré à l'utilisateur, il est transmis au Service Provider.
 - Le Service Provider lit les déclarations SAML contenues dans le ticket et prend une décision d'autorisation.
 - Fonctionnement similaire à Kerberos : SAML utilise des tickets ou tokens pour autoriser l'accès.

Composants Clés de SAML



Composant	Fonction
Assertion	Transmet des informations d'authentification, d'autorisation et d'autres attributs.


Protocole	Définit la méthode par laquelle les entités demandent et répondent aux requêtes
Bindings	Associe SAML aux protocoles de communication standards (ex : HTTP, SOAP).
Profiles	Précisent les cas d'usage de SAML pour différentes applications métiers (ex : SSO, LDAP).

Caractéristiques essentielles de SAML

- 1. **Utilisation de tokens ou tickets d'assertion :**
 - Ces tokens servent de preuve d'authentification et d'autorisation.
- 2. **Langage XML :**
 - Les assertions SAML sont écrites en XML (Extensible Markup Language), ce qui permet une communication claire, à la fois lisible par les machines et compréhensible par les humains.
- 3. **Interopérabilité :**
 - SAML est une norme largement acceptée et utilisée pour l'authentification fédérée entre des systèmes et organisations différents.

Pourquoi SAML est important dans la FIM ?

- 1. **Sécurité renforcée :**
 - L'utilisateur n'a pas besoin de transmettre ses identifiants au Service Provider. Seul un ticket sécurisé (SAML Assertion) est utilisé.
- 2. **Expérience utilisateur fluide :**
 - L'utilisateur peut accéder à plusieurs systèmes ou services après une authentification unique (SSO).
- 3. **Interopérabilité entre organisations :**
 - SAML permet aux organisations d'établir des relations de confiance fédérée pour gérer l'accès aux ressources de manière sécurisée.
- 4. **Diminution des risques :**
 - Moins de dépendance aux mots de passe grâce aux tickets sécurisés, réduisant les risques d'attaques comme le **phishing** ou le vol d'identifiants.

 **Résumé :**

SAML joue un rôle central dans la gestion fédérée des identités (FIM) en permettant l'authentification unique (SSO) et en facilitant les relations de confiance entre des entités différentes. Sa capacité à utiliser des assertions sécurisées et son format XML en font une solution robuste pour l'authentification et l'autorisation dans des environnements complexes et multi-organisations.

1.2.15 Responsabilité = Principe de Contrôle d'Accès

💡 La responsabilité est au cœur du principe de contrôle d'accès.

Le principe de contrôle d'accès repose sur la responsabilité, c'est-à-dire la capacité à attribuer les actions effectuées dans un système à des utilisateurs spécifiques.



👉 En résumé :

Lorsque tous ces composants sont en place (identification, authentification, autorisation et journalisation) le principe de contrôle d'accès est pleinement respecté, et la responsabilité des utilisateurs peut être garantie.

1.2.16 Accès "Just-in-time" (JIT)

💡 Accès Just-in-time

Fait référence à l'élévation temporaire des privilèges d'un utilisateur autorisé pour une courte période afin qu'il puisse réaliser des tâches nécessaires mais peu fréquentes.

L'accès Just-in-time réduit le besoin d'élévation prolongée des privilèges, minimisant ainsi les risques potentiels de sécurité.

Le terme **"Just-in-time"** provient du monde de la production industrielle :

- Une entreprise reçoit les composants nécessaires à la production au moment précis où ils sont requis, sans stockage excessif.
- Cela permet à l'organisation de gagner en efficacité en réduisant les besoins de gestion d'inventaire.

Application au domaine de la sécurité

L'accès Just-in-time fonctionne de manière similaire mais dans un contexte de sécurité informatique :

- Par exemple, un utilisateur peut avoir besoin d'accéder à une base de données sensible une fois par mois pour générer un rapport.
- L'accès Just-in-time lui accorde des privilèges élevés uniquement pendant cette fenêtre temporelle limitée. Une fois la tâche accomplie, les privilèges sont révoqués automatiquement.

Avantages de l'accès Just-in-time



Réduction des risques

Minimisation de la surface d'attaque



Limitation de la durée des privilèges élevés



Protection contre les cyberattaques et abus internes



Élévations non-permanentes

Évitement des droits permanents inutiles



Suppression des droits administratifs prolongés



Droits accordés uniquement pour la durée nécessaire



Automatisation

Processus automatisé et efficace



Réduction des erreurs manuelles



Processus plus efficace et fiable



Efficacité opérationnelle

Accès précis et opportun



Accès accordé au moment exact du besoin



Équilibre entre sécurité et opérations



L'accès Just-in-time (JIT) est une stratégie efficace qui équilibre les besoins en productivité et en sécurité. En limitant l'élévation des privilèges à des périodes courtes et précises, il permet de réduire les risques liés aux droits excessifs, tout en maintenant l'efficacité opérationnelle des utilisateurs et des systèmes.

1.3 Identité fédérée avec un service tiers

1.3.1 Identity as a Service (IDaaS)



Identity as a Service (IDaaS) :

- Fait référence à l'implémentation ou à l'intégration des services d'identité dans un environnement basé sur le cloud.
- Cela inclut l'identification, l'authentification, l'autorisation et l'accès fédéré dans le cloud.

Fonctionnalités clés d'IDaaS

L'IDaaS offre diverses fonctionnalités pour la gestion des identités dans le cloud, notamment :

1. **Provisionnement** : Création, mise à jour et suppression des comptes utilisateurs.
2. **Administration** : Gestion des politiques d'accès et des identités.
3. **Single Sign-On (SSO)** : Authentification unique pour accéder à plusieurs services.
4. **Authentification multifacteur (MFA)** : Amélioration de la sécurité en exigeant plusieurs méthodes d'authentification.
5. **Services d'annuaire** : Intégration avec des annuaires tels qu'Active Directory.
6. **Prise en charge sur site (On Premises) et dans le cloud** : Flexibilité pour gérer des environnements hybrides.

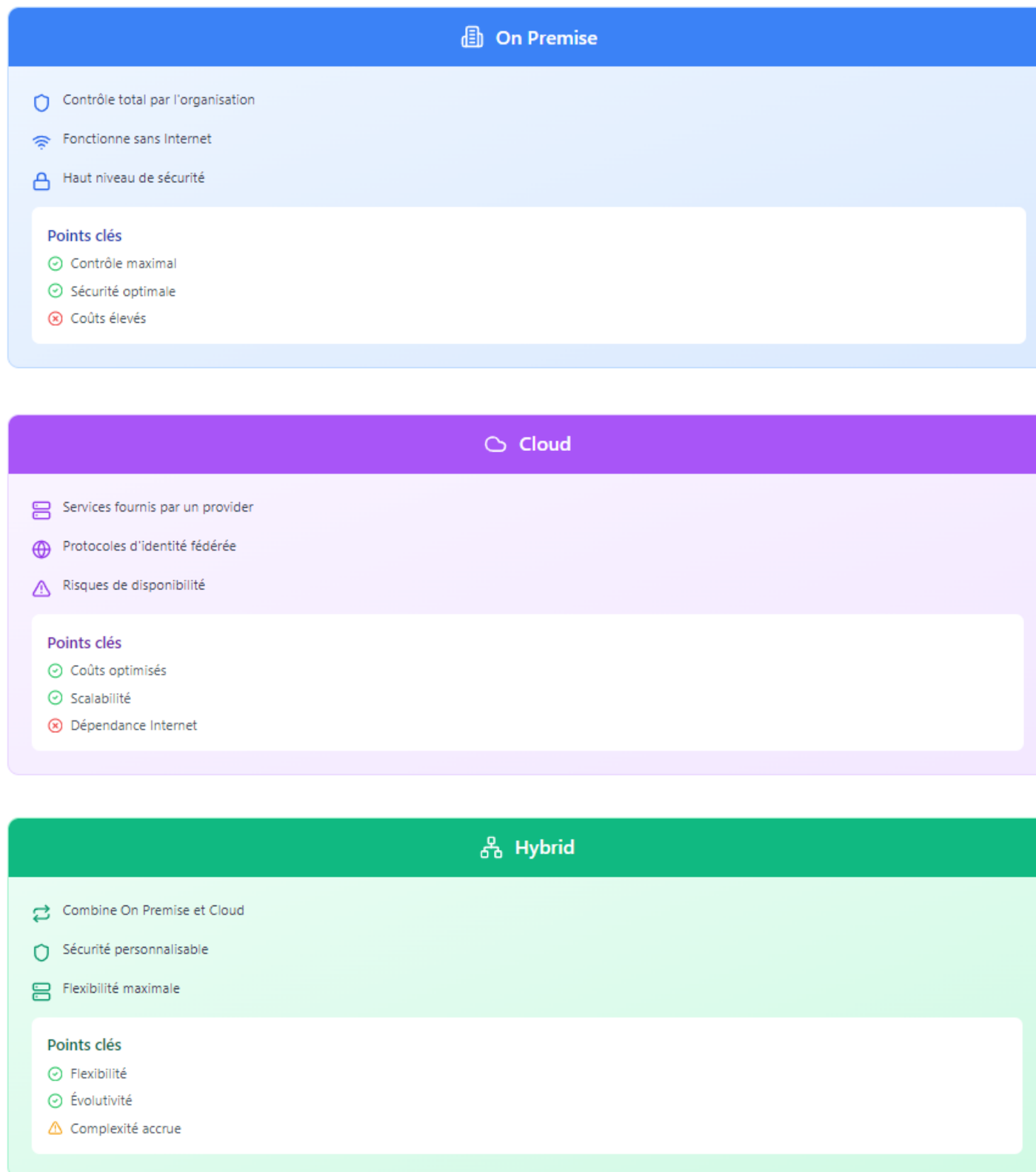
Types d'identités IDaaS

Les services IDaaS prennent en charge plusieurs types d'identités :

Cloud Identity
Créée et gérée directement dans le cloud
Authentification par service cloud
Synced Identity
Synchronisée entre stockage local et cloud
Authentification locale ou cloud
Linked Identities
Deux comptes distincts liés ensemble
Authentification locale ou cloud
Federated Identity
Authentifiée via un Identity Provider externe
Authentification par Identity Provider

Solutions de gestion des identités et des accès (IAM)

Les solutions IAM utilisent l'un des trois modèles suivants :



Le choix du modèle de déploiement dépend des besoins spécifiques de l'organisation en termes de sécurité, de flexibilité et de budget.

Risques potentiels liés à IDaaS

1. Disponibilité du service :

- Si le fournisseur cloud subit une panne ou une interruption, les utilisateurs seront incapables d'accéder aux systèmes.

2. Protection des données d'identité critiques :

- Les informations sensibles (comme les **PII** - *Personally Identifiable Information*) seront stockées chez le fournisseur cloud.
- Une protection adéquate dépend des mécanismes de sécurité mis en place par le fournisseur.

3. Confiance en un tiers avec des données sensibles ou propriétaires :

- Les données partagées avec le fournisseur cloud peuvent inclure des informations sensibles sur l'organisation.

- Des protections doivent être mises en place pour empêcher les fuites ou le partage non autorisé de ces informations.



En résumé :

L'Identity as a Service (IDaaS) permet une gestion centralisée des identités dans le cloud avec des fonctionnalités avancées comme le SSO, l'authentification multifacteur et le provisionnement. Toutefois, il est essentiel d'évaluer les risques associés, tels que la disponibilité du service, la sécurité des données d'identité et la confiance accordée à un tiers fournisseur. Les modèles On Premise, Cloud et Hybride offrent des solutions adaptées selon les besoins spécifiques de l'organisation.

1.4 Implémenter et gérer les mécanismes d'autorisation



Un mécanisme d'autorisation est un ensemble de règles, politiques et procédures utilisées pour déterminer qui peut accéder à une ressource, à quel moment, et à quel niveau. Il s'agit d'une composante essentielle de la gestion des accès qui vise à sécuriser les systèmes et données en s'assurant que seuls les utilisateurs autorisés peuvent effectuer des actions spécifiques.

Il existe plusieurs modèles de mécanisme d'autorisations.



Contrôle d'accès discrétionnaire (DAC)

Contrôle par le propriétaire de la ressource

✓ Accès géré à la discrétion du propriétaire



Contrôle d'accès basé sur les rôles (RBAC)

Accès basé sur les rôles des utilisateurs

✓ Attribution des droits par rôle



Contrôle d'accès basé sur les attributs (ABAC)

Utilisation des attributs utilisateurs

✓ Multiple critères d'évaluation



Contrôle d'accès contextuel

Analyse du contexte et des risques

✓ Évaluation du contexte de connexion




Le choix du type de contrôle d'accès dépend des besoins spécifiques de l'organisation en termes de granularité et de flexibilité.

Types de contrôle d'accès

Les mécanismes d'autorisation se divisent en trois catégories principales :

- 1. **Discrétionnaire (Discretionary) :**
 - Le propriétaire décide qui a accès à la ressource.
- 2. **Obligatoire (Mandatory) :**
 - Le système décide en fonction de règles strictes (ex : étiquettes de classification).
- 3. **Non-discrétionnaire :**
 - Une entité autre que le propriétaire (comme un administrateur ou un système automatisé) décide des droits d'accès.


Type de contrôle	Caractéristiques
Discretionary Access Control (DAC)	Le propriétaire détermine les règles d'accès.
Role-Based Access Control (RBAC)	L'accès est basé sur les rôles des utilisateurs (ex : administrateur, employé).
Rule-Based Access Control	L'accès dépend d'un ensemble de règles (ex : ACL - Liste de Contrôle d'Accès).
Attribute-Based Access Control (ABAC)	L'accès est basé sur des attributs des utilisateurs (ex : OS, version du navigateur).
Mandatory Access Control (MAC)	Le système détermine l'accès en fonction des étiquettes de classification.
Risk-Based Access Control	Analyse des éléments de connexion (ex : adresse IP, heure) pour évaluer le niveau de risque.

 La gestion des mécanismes d'autorisation est essentielle pour garantir un contrôle sécurisé des accès aux ressources. Chaque modèle de contrôle d'accès — qu'il soit discrétionnaire, obligatoire ou basé sur des règles et attributs — présente des caractéristiques adaptées à différents besoins organisationnels et niveaux de sécurité.

1.4.1 Exploration de chaque mécanisme de contrôles d'accès

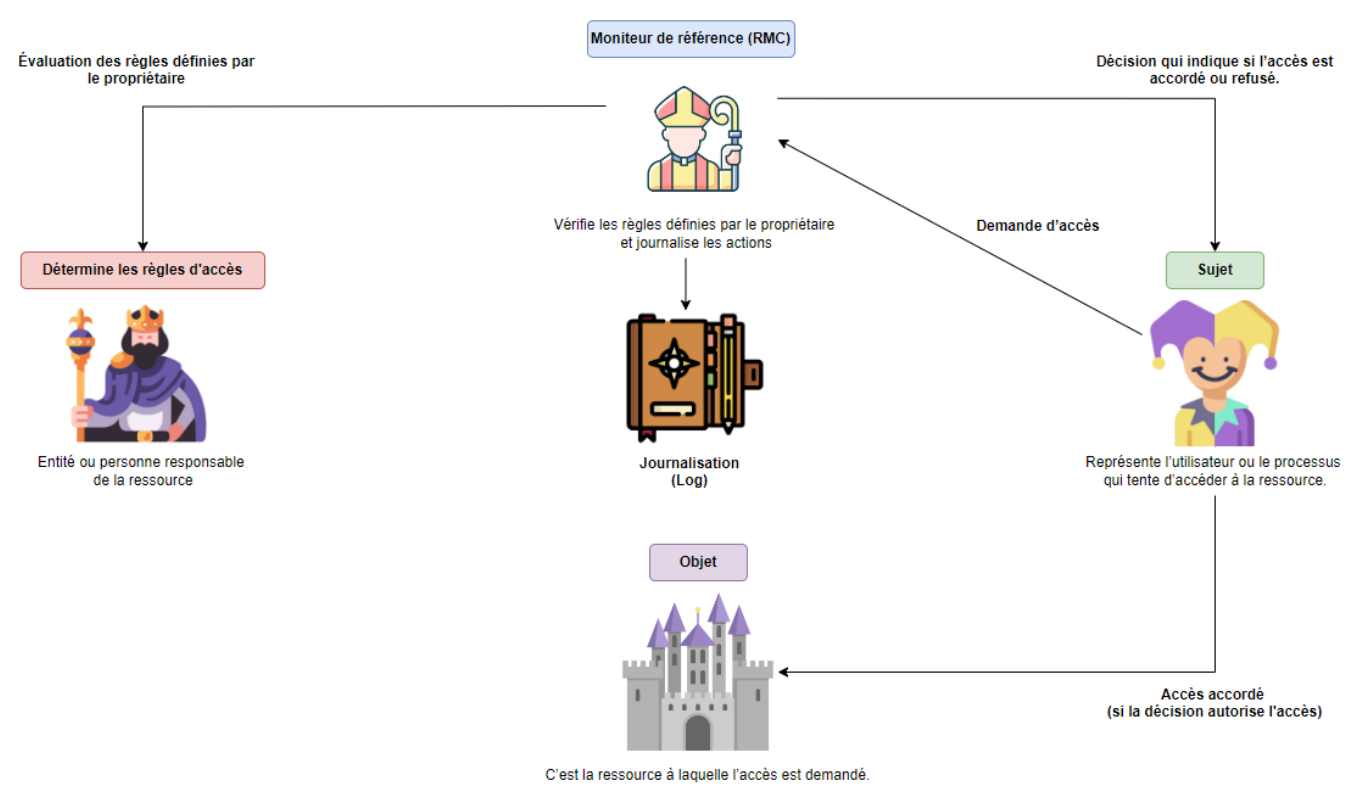
Commençons par le **DAC** : **Contrôle d'accès discrétionnaire**.

1. Contrôle d'accès discrétionnaire (DAC)

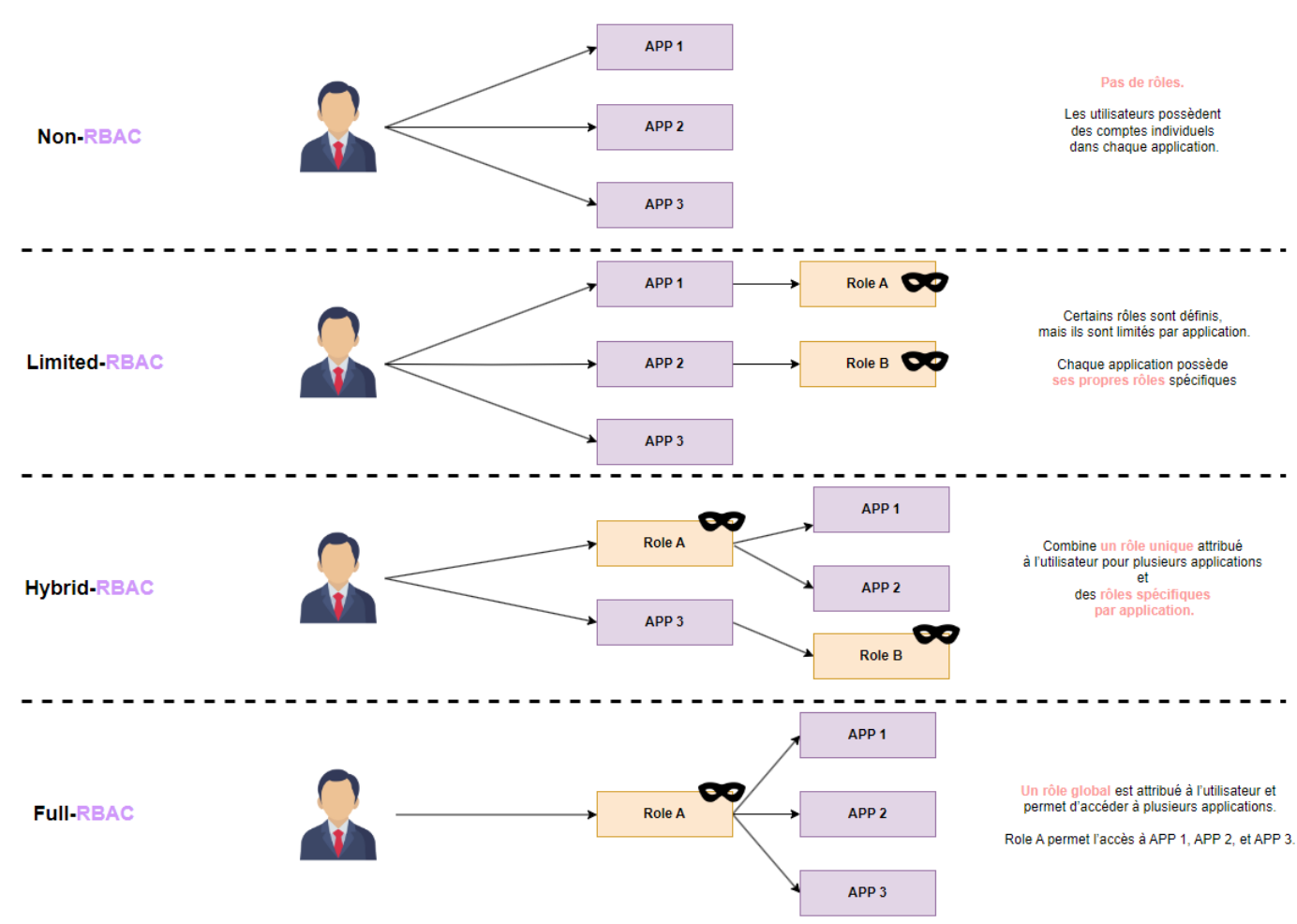


Le contrôle d'accès discrétionnaire est un modèle où le propriétaire de la ressource décide qui peut accéder à une ressource et quels privilèges (lecture, écriture, exécution) sont accordés.

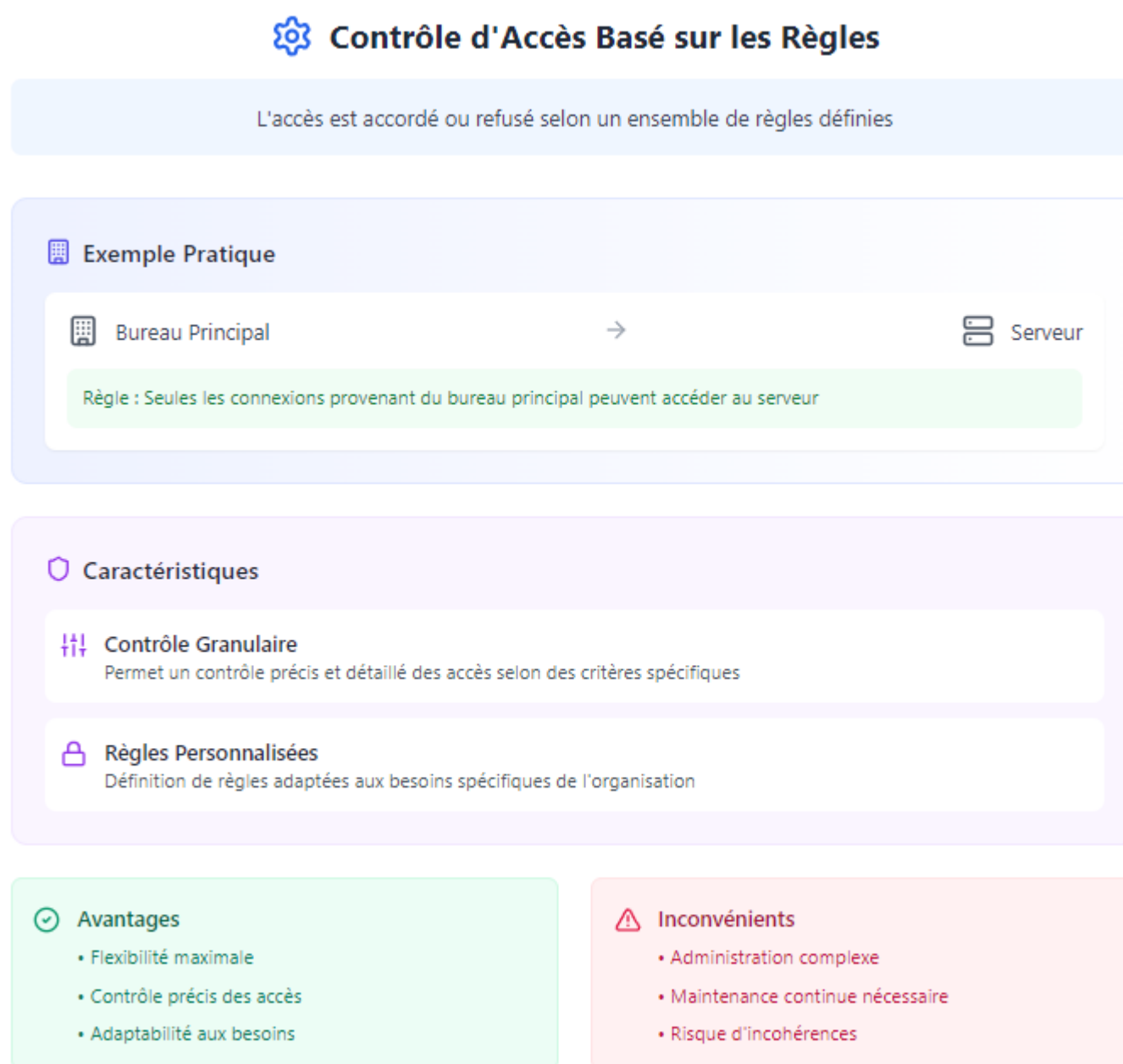
L'accès est accordé à la discrétion du propriétaire :



2. Contrôle d'accès basé sur les rôles (RBAC)



3. Contrôle d'accès basé sur les règles (Rule-Based Access Control)



4. Contrôle d'accès basé sur les attributs (ABAC)

- **Définition :** L'accès est accordé en fonction des **attributs** associés à l'utilisateur, à l'environnement, à la ressource et à l'action demandée.
- **Attributs pris en compte :**
 - **Utilisateur :** Identité, rôle, expérience, département.
 - **Environnement :** Version OS, localisation géographique, type d'appareil.
 - **Ressource :** Type et classification de la ressource.
 - **Action :** Lecture, écriture, exécution (RWX).
- **Exemple :**
 - Un utilisateur peut accéder à un fichier **uniquement depuis un appareil approuvé et pendant les heures de bureau.**

Avantage : Granularité extrême et flexibilité.

Inconvénient : Nécessite un moteur d'autorisation performant et une **gestion des politiques complexe.**

5. Contrôle d'accès obligatoire (MAC)

- **Définition** : Le système détermine l'accès en fonction de **niveaux de classification** et d'étiquettes attribuées aux utilisateurs et aux ressources.
- **Principe** : L'accès est strictement contrôlé par des **politiques de sécurité centralisées**.
- **Exemple** : Un document classifié « Confidentiel » ne peut être ouvert que par un utilisateur disposant d'un niveau d'habilitation « Confidentiel ».

Avantage : Sécurité élevée grâce à un contrôle centralisé.

Inconvénient : Moins flexible, difficile à gérer pour les grandes organisations.

6. Contrôle d'accès basé sur les risques (Risk-Based Access Control)

- **Définition** : Analyse les **éléments d'une connexion utilisateur** pour évaluer un **niveau de risque** associé à la demande d'accès.
- **Éléments évalués** :
 - Adresse IP,
 - Heure de connexion,
 - Localisation géographique,
 - Type d'appareil.
- **Exemple** : Une tentative d'accès depuis une adresse IP suspecte entraîne une **demande d'authentification supplémentaire** (ex. : MFA).

Avantage : Prend en compte le contexte pour une sécurité adaptative.

Inconvénient : Nécessite des algorithmes d'évaluation du risque précis.

7. eXtensible Access Control Markup Language (XACML)

- **Définition** : Une norme qui définit et permet l'implémentation du **contrôle d'accès basé sur les attributs** (ABAC).
- **Fonctionnement** :
 - **XACML** fournit un langage pour définir les politiques, l'architecture et le modèle de traitement des demandes d'accès.
 - Il permet une mise en œuvre **standardisée et automatisée** du contrôle d'accès basé sur les attributs.

Exemple : Une politique XACML peut spécifier qu'un utilisateur du département « RH » ne peut accéder aux données de paie qu'à partir d'un appareil sécurisé.

1.4.2 Description du Contrôle d'Accès Obligatoire (MAC)



Le Mandatory Access Control (MAC) est un modèle où les règles d'accès sont strictement déterminées par un système central.

Le propriétaire n'a pas de contrôle direct sur les autorisations.

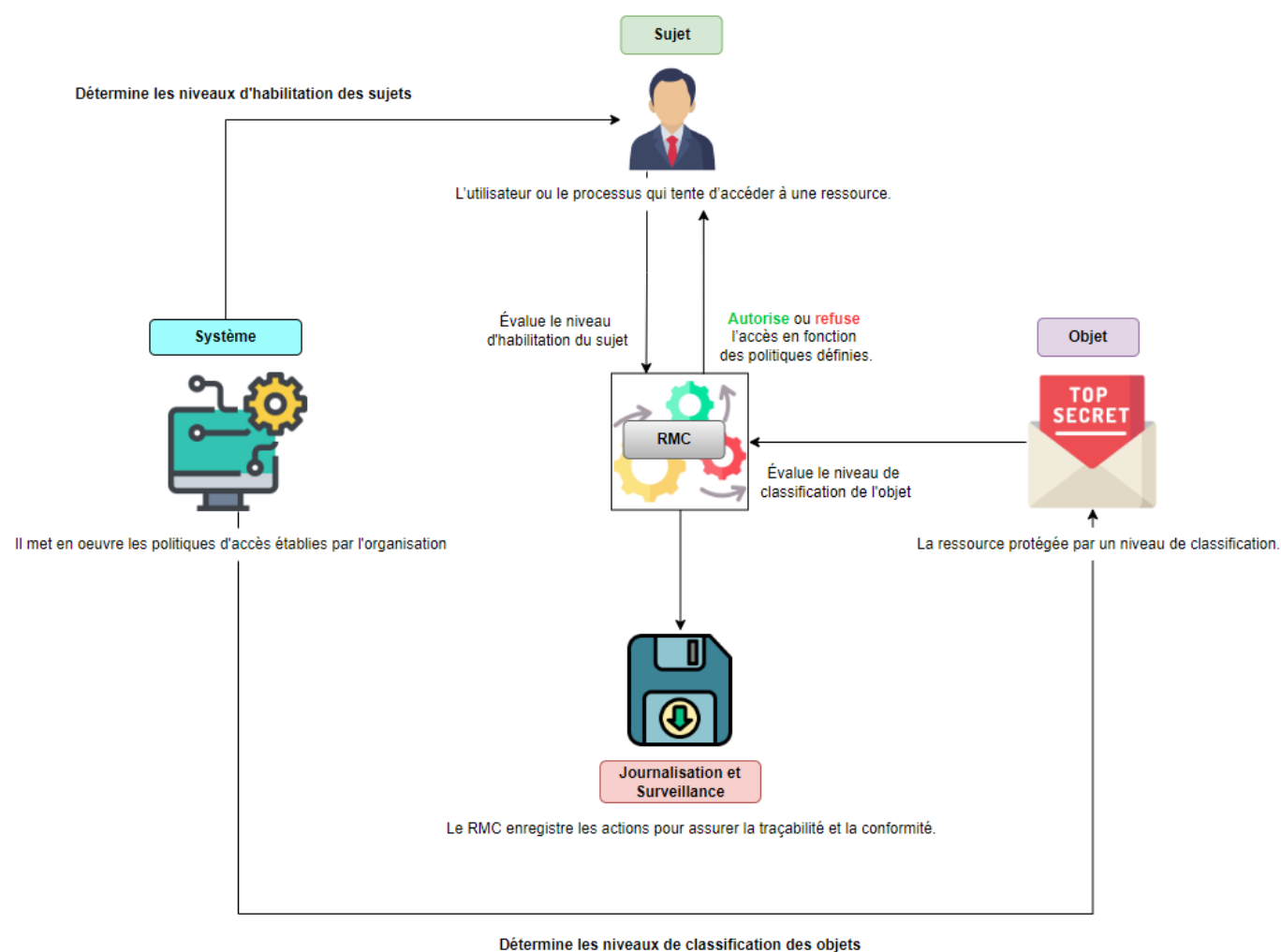
Principes fondamentaux :

- **Classification des objets** : Chaque ressource est étiquetée avec un niveau de classification (ex : Public, Secret, Top Secret).
- **Niveaux d'habilitation des sujets** : Chaque utilisateur (ou sujet) est associé à un **niveau de sécurité** (ou habilitation).
- **Décision d'accès** : L'accès est accordé ou refusé en fonction de la **correspondance** entre :
 - Le niveau de classification de la ressource.
 - Le niveau d'habilitation de l'utilisateur.

Utilisation typique :

- Principalement utilisé dans des organisations gouvernementales ou militaires où la confidentialité est primordiale.
- Moins courant dans le secteur privé en raison de la complexité de mise en œuvre.

Exemple en schéma :



- **Système** :
 - Le système **détermine l'accès** en fonction de la **classification** de l'objet et du **niveau d'habilitation** de l'utilisateur.
 - **Label dans le schéma** : "System determines based on clearance of subject and sensitivity of object".
- **Sujet (Subject)** :
 - L'utilisateur ou le processus qui tente d'accéder à une ressource.
 - **Exemple** : Un utilisateur avec un niveau **"Public"** tente d'accéder à une ressource classifiée **"Secret"**.
- **Objet (Object)** :
 - La ressource protégée par un **niveau de classification**.
 - **Exemple** : Fichier classé **"Top Secret"**.

- **Moniteur de Référence (RMC) :**
 - Composant central qui évalue la **demande d'accès** en comparant :
 - Le **niveau d'habilitation** du sujet.
 - Le **niveau de classification** de l'objet.
 - **Fonctionnalité :**
 - Accorde ou refuse l'accès.
 - Journalise et surveille toutes les tentatives d'accès.
- **Journalisation et Surveillance (Log & Monitor) :**
 - Le RMC **enregistre les actions** pour assurer la traçabilité et la conformité.

Caractéristiques clés du MAC

1. **Système centralisé** : Le contrôle est déterminé par des **politiques** définies par l'organisation ou le système.
2. **Classification obligatoire** : Chaque ressource (objet) doit avoir une étiquette de classification.
3. **Habilitation obligatoire** : Chaque utilisateur (sujet) doit avoir un niveau d'habilitation clairement défini.
4. **Strict et rigide** : Le modèle est non flexible car les utilisateurs ne peuvent pas modifier les permissions.



En résumé :

Le Contrôle d'Accès Obligatoire (MAC) repose sur un système centralisé pour attribuer ou refuser l'accès aux ressources en fonction de :

1. La classification de l'objet (Public, Secret, Top Secret).
2. Le niveau d'habilitation du sujet (Utilisateur autorisé).

Ce modèle est particulièrement adapté aux organisations gouvernementales et aux environnements nécessitant un haut niveau de sécurité et de confidentialité. Le schéma présenté montre clairement l'importance du Moniteur de Référence (RMC) comme composant clé pour contrôler, surveiller et journaliser les décisions d'accès.

1.4.3 Application des politiques d'accès

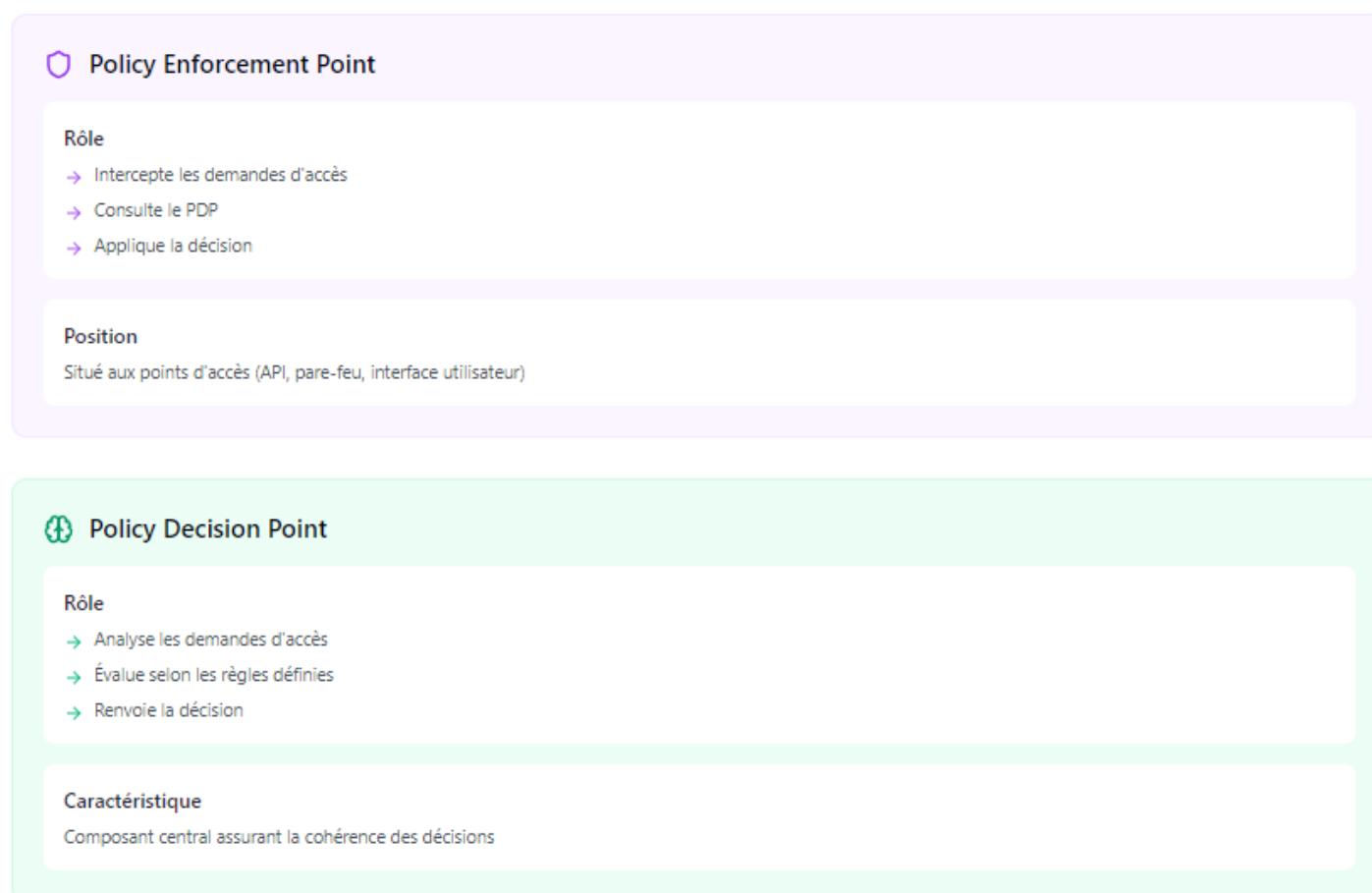


L'Access Policy Enforcement est un processus permettant de contrôler et appliquer les règles d'accès aux ressources protégées d'un système, d'une application ou d'un réseau. Elle garantit que les utilisateurs, les processus ou les systèmes ne peuvent accéder aux ressources qu'en respectant des politiques d'accès préalablement.

Il existe deux aspects critiques pour l'application des politiques d'accès :

1. Policy Enforcement Point (PEP)
2. Policy Decision Point (PDP)

Composants clés de l'Access Policy Enforcement



Fonctionnement global d'une Access Policy Enforcement

1. Interception de la demande :

- Un utilisateur ou un système tente d'accéder à une ressource protégée (exemple : fichier, base de données).
- Le **PEP** intercepte la requête et agit comme un **portier**.

2. Transmission au PDP :

- Le PEP transmet la demande au **Policy Decision Point (PDP)** pour évaluation.

3. Évaluation par le PDP :

- Le PDP évalue la demande selon les **politiques d'accès** prédéfinies.
- Les politiques prennent en compte divers critères :
 - **Rôle** de l'utilisateur (RBAC),
 - **Attributs** (ABAC),
 - **Contexte** (heure, localisation, appareil utilisé),
 - **Règles spécifiques** (ex : listes de contrôle d'accès).

4. Décision et retour au PEP :

- Le PDP renvoie une **décision d'autorisation** (Permit) ou de **refus** (Deny) au PEP.

5. Application de la décision :

- Le PEP applique la décision :
 - Si l'accès est **autorisé**, l'utilisateur accède à la ressource.
 - Si l'accès est **refusé**, l'utilisateur est bloqué.
- Toutes les actions sont **journalisées** pour garantir la traçabilité et l'audit.

Exemple concret d'une Access Policy Enforcement

1. Scénario : Un utilisateur tente d'accéder à un fichier sensible via une application d'entreprise.

- **Étape 1** : L'utilisateur fait une demande d'accès via l'interface.
- **Étape 2** : Le **PEP** intercepte la demande et l'envoie au **PDP**.
- **Étape 3** : Le **PDP** évalue la demande selon les politiques suivantes :
 - **Rôle** : L'utilisateur est-il un manager ?
 - **Localisation** : Est-il connecté depuis le bureau ?

- Heure : La demande est-elle faite pendant les heures de travail ?
- **Étape 4** : Le **PDP** retourne une décision "**Refusé**" car l'utilisateur n'a pas les privilèges requis.
- **Étape 5** : Le **PEP** applique la décision et bloque l'accès à la ressource. L'événement est journalisé.

Avantages de l'Access Policy Enforcement

1. Cohérence des décisions d'accès :

- Grâce à un **PDP centralisé**, les politiques sont appliquées uniformément sur l'ensemble du système.

2. Sécurité renforcée :

- Le **PEP** agit comme un **point de contrôle** pour toutes les demandes d'accès.
- Seules les décisions autorisées sont appliquées.

3. Audit et traçabilité :

- Toutes les demandes d'accès et leurs décisions sont **journalisées**, facilitant les audits de sécurité.

4. Flexibilité :

- Les politiques peuvent être ajustées pour s'adapter aux **règles métiers** et aux besoins de sécurité (par exemple, RBAC, ABAC).



En résumé :

L'Access Policy Enforcement est un processus qui permet de faire respecter les politiques d'accès à des ressources protégées via deux composants essentiels :

1. Policy Enforcement Point (PEP) : Applique les politiques d'accès aux points d'accès.
2. Policy Decision Point (PDP) : Prend les décisions d'accès en évaluant les règles et politiques prédéfinies.

En travaillant ensemble, ces deux composants assurent un contrôle rigoureux et centralisé des accès, renforçant ainsi la sécurité globale des systèmes et des données.

1.5.1 Gestion du cycle de vie des identités et des accès : Accès des fournisseurs



Importance de la gestion des accès pour les fournisseurs :

- Les fournisseurs tiers (ex. : prestataires de services IT, marketing, finance, chaînes d'approvisionnement) nécessitent parfois un accès aux systèmes et données d'une organisation.
- Cet accès peut présenter des risques significatifs pour la sécurité.
- Par conséquent, l'attribution et la gestion des identités et des accès pour les fournisseurs doivent être traitées avec autant, voire plus de rigueur que celles des employés internes.

Provisionnement des accès des fournisseurs :

- Le processus de provisionnement doit inclure :
 - Validation de l'identité et des besoins d'accès du fournisseur.
 - Revue de sécurité approfondie du fournisseur, qui peut inclure :
 - Analyse des systèmes et processus utilisés par le fournisseur.
 - Inspections sur site des installations du fournisseur.
 - Vérification de la conformité avec les politiques de sécurité.

- Une fois les accès attribués, des activités comme la revue régulière et la révocation des accès doivent être planifiées pour garantir que les droits d'accès restent appropriés.

Pourquoi est-ce important ?

1. **Risque accru** : Les relations avec des tiers élargissent la surface d’attaque de l’organisation. Une mauvaise gestion des accès des fournisseurs peut entraîner des fuites de données ou des violations de sécurité.
2. **Criticité des fonctions** : Les fournisseurs jouent un rôle clé dans des fonctions critiques (ex. : support IT, gestion financière). Leur accès doit donc être limité et surveillé.
3. **Contrôle rigoureux** : L’accès des fournisseurs doit être :
 - Restreint aux ressources nécessaires (principe du moindre privilège).
 - Temporaire, avec des révisions régulières.
 - Traçable pour garantir une responsabilité et une conformité totale.

Résumé

La gestion des identités et des accès des fournisseurs doit :

👉 Pour résumé :

La gestion des identités et des accès des fournisseurs doit :

1. Suivre les mêmes standards de sécurité que pour les employés internes, voire plus rigoureux.
2. Inclure des évaluations de sécurité approfondies, des inspections et des revues régulières des accès.
3. Mettre en place des processus pour le provisionnement, la surveillance et la révocation des accès afin de réduire les risques liés aux tiers.

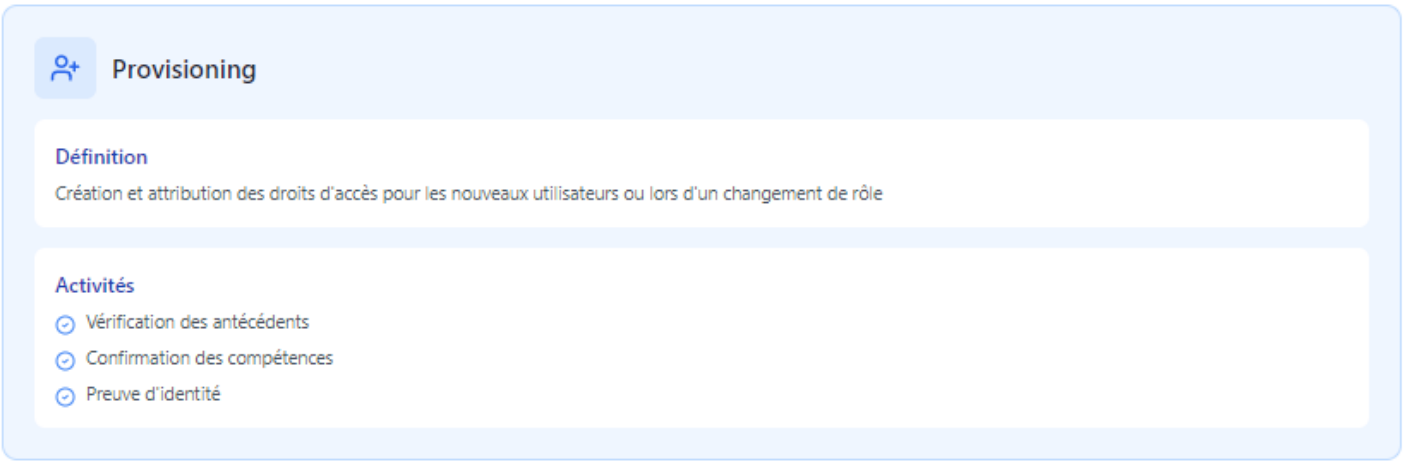
Un contrôle strict garantit que les fournisseurs n'accèdent qu'aux systèmes nécessaires et que les accès sont sécurisés, surveillés et révisés en continu.

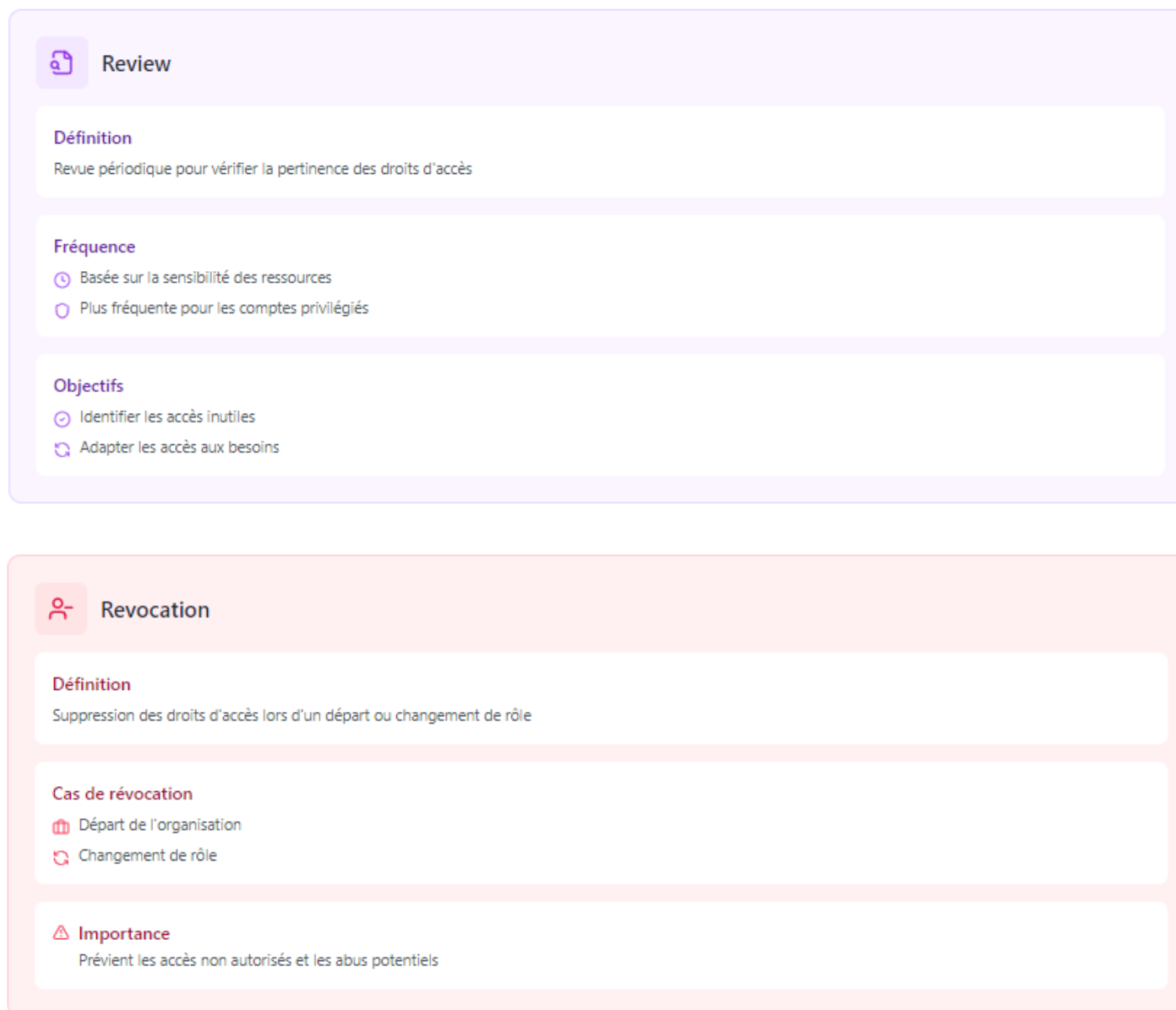
1.5.2 Cycle de vie des identités (Identity Life Cycle)



Le cycle de vie des identités est un processus qui permet de gérer l'accès des utilisateurs à des systèmes et ressources tout au long de leur parcours dans une organisation.

Le cycle de vie des identités comporte trois étapes principales :





Importance du Cycle de vie des identités

1. Amélioration de la sécurité :

- Réduit les risques d'accès non autorisés ou dormants.
- Limite les abus potentiels liés aux privilèges excessifs.

2. Conformité :

- Assure la conformité avec les normes et réglementations (ex : RGPD, HIPAA) exigeant une gestion stricte des accès.

3. Efficacité opérationnelle :

- Facilite l'intégration rapide des nouveaux employés grâce à un provisionnement efficace.
- Optimise la gestion des accès pour éviter les erreurs humaines.

4. Réduction des coûts :

- Évite les accès inutiles qui peuvent alourdir la charge du système.

👉 Pour résumé :

Le Identity Life Cycle est un processus essentiel dans la gestion des identités et des accès (IAM). En respectant les trois étapes principales — Provisioning, Review et Revocation — les organisations assurent une sécurité renforcée, une gestion efficace des accès, et une conformité aux exigences réglementaires.

1.5.3 Revue des accès utilisateurs



Qu'est-ce qu'une revue des accès utilisateurs ?

- La **revue des accès** est un **processus continu** permettant d'évaluer si les accès d'un utilisateur, d'un système ou d'un service sont **toujours appropriés** et nécessaires.
- Elle s'applique à tous types de comptes :
 - **Comptes utilisateurs.**
 - **Comptes systèmes.**
 - **Comptes de services.**

Pourquoi effectuer des revues d'accès ?

- Les droits d'accès attribués à un utilisateur lors de son intégration ne doivent pas être permanents sans vérification.
- Objectifs :
 - S'assurer que les accès restent appropriés et nécessaires.
 - Révoquer les accès inutiles pour éviter les risques de "privilège excessif" (Privilege Creep).
 - Maintenir la sécurité des systèmes et des données sensibles.
- Fréquence des revues :
 - La fréquence dépend de la valeur des ressources et des risques associés.
 - Les comptes privilégiés (ex : administrateurs, super utilisateurs) doivent être examinés plus fréquemment que les comptes standards.

Revue d'Accès : Nécessité et Fréquence

Pourquoi les revues sont nécessaires



Validation périodique

Confirmation que les accès restent appropriés malgré les changements



Réduction des risques

Prévention de l'accumulation de privilèges et des accès non révoqués



Responsabilité de l'asset owner

Évaluation de la nécessité des accès par le propriétaire de la ressource

🕒 Fréquence des revues

📅 Revue périodique

Au moins une fois par an pour les accès standards

👤 Changement de rôle

🕒 Retrait des anciens accès

🕒 Attribution des nouveaux accès

👤 Départ de l'organisation

Révocation immédiate de tous les accès

🛡️ Comptes privilégiés

Revue hebdomadaire ou mensuelle pour les super utilisateurs

Quels comptes doivent être examinés le plus fréquemment ?

- **Comptes privilégiés :**
 - Accès administrateur (admin, root).
 - Comptes de service critiques.
 - Comptes avec des privilèges étendus.
- **Ressources sensibles :**
 - Les comptes ayant accès à des données sensibles ou à des systèmes critiques.
- **Accès risqués :**
 - Les comptes utilisés pour des tâches temporaires ou des projets spécifiques doivent être examinés plus régulièrement.

Résumé des bonnes pratiques pour la revue des accès

1. **Automatiser les revues d'accès :**
 - Utiliser des outils d'Identity and Access Management (IAM) pour faciliter les revues et générer des rapports d'audit.
2. **Prioriser les comptes sensibles :**
 - Réaliser des revues plus fréquentes pour les comptes privilégiés et à haut risque.
3. **Inclure les changements de rôle :**
 - Chaque changement de poste doit déclencher une revue immédiate pour éviter des accès inappropriés.
4. **Documenter les résultats :**
 - Conserver des preuves des revues réalisées pour répondre aux exigences réglementaires et d'audit.
5. **Révoquer immédiatement les accès non nécessaires :**
 - Agir rapidement pour supprimer les accès inutiles après une revue.

👉 En résumé :

La **revue des accès utilisateurs** est un processus essentiel pour maintenir une **sécurité optimale** et une **gestion efficace des privilèges**. En réexaminant régulièrement les accès, notamment ceux des comptes privilégiés, les organisations minimisent les risques d'accès non autorisés et assurent une conformité avec les politiques de sécurité internes et externes.

1.5.4 Privilège Escalation (Élévation de privilèges, utilisation de sudo, audit)

💡 L'élévation de privilèges consiste à utiliser un compte ou des permissions avec droits élevés pour exécuter des tâches administratives critiques. Cela s'applique généralement aux administrateurs, aux comptes "root", ou aux utilisateurs avec des privilèges élevés.

⚠ Utilisation stricte des comptes privilégiés

⊗ À ne pas faire avec un compte privilégié

- ✉ Vérifier les emails
- 🌐 Naviguer sur le web
- 👥 Participer aux réunions

✅ Utilisation appropriée

- ⚙ Tâches administratives critiques
- 📋 Configuration système

👤 Utilisation de deux comptes distincts

👤 **Compte Standard**
Pour les tâches quotidiennes ne nécessitant pas de privilèges élevés

🛡 **Compte Privilégié**
Uniquement pour les tâches administratives spécifiques

>_ Exemples d'élévation temporaire de privilèges

Unix/Linux (sudo)

```
sudo apt-get update
```

Windows (RunAs)

```
runas /user:admin "programme.exe"
```

Pourquoi cette séparation est-elle cruciale ?

1. Réduction des risques de compromission :

- Les activités courantes (emails, web) exposent le compte à des **menaces** comme le phishing ou les malwares.
- Si un **compte standard** est compromis, l'impact est **limité**. En revanche, la compromission d'un compte privilégié pourrait entraîner des **dommages majeurs**.

2. Sécurité renforcée :

- Limiter le temps d'utilisation des comptes privilégiés réduit la fenêtre d'opportunité pour un attaquant potentiel.

3. Audit et traçabilité :

- L'utilisation de commandes telles que **sudo** est **journalisée**, permettant de suivre :
 - **Qui** a exécuté une commande.
 - **Quand** elle a été exécutée.
 - **Quelle commande** a été utilisée.
- Cela simplifie les audits et la détection d'activités suspectes.

Exemple d'application pratique

1. Connexion avec un compte standard :

- L'administrateur utilise son compte utilisateur habituel pour les tâches courantes (email, web).

2. Exécution de commandes privilégiées :

- Pour effectuer une mise à jour ou modifier la configuration système, il utilise `sudo`

```
sudo systemctl restart apache2
```

Audit des activités :

- Les commandes exécutées via `sudo` sont enregistrées dans des fichiers de **journalisation** (par exemple `/var/log/sudo.log` sous Linux).

Avantages clés

1. Sécurité accrue :

- Limite les risques d'utilisation abusive ou de compromission des comptes privilégiés.

2. Principe du moindre privilège :

- Les administrateurs utilisent **seulement les privilèges nécessaires** pour leurs tâches.

3. Visibilité et traçabilité :

- Chaque action nécessitant des privilèges élevés est **journalisée** pour des fins d'audit et de conformité.

4. Prévention des attaques :

- En cas de compromission d'un compte standard, les privilèges élevés restent sécurisés.

Pour résumé :

L'élévation de privilèges via des outils comme

`sudo` (Unix/Linux) ou RunAs (Windows) est une bonne pratique essentielle pour sécuriser les comptes administratifs. En séparant l'utilisation des comptes standard et privilégié, les organisations appliquent le principe du moindre privilège et réduisent le risque de compromission. Les commandes privilégiées doivent être auditable pour garantir une traçabilité des actions critiques.

1.6.1 Systèmes d'authentification



Un système d'authentification permet de prouver ou de vérifier une identité ou une assertion relative à un système, un utilisateur ou un service.

L'objectif principal est de protéger les actifs critiques d'une organisation contre des accès non autorisés.

1. Systèmes d'authentification populaires :

- OpenID Connect (OIDC)/Open Authorization (OAuth)
- Security Assertion Markup Language (SAML)
- Kerberos
- Remote Authentication Dial-In User Service (RADIUS)
- Terminal Access Controller Access Control System Plus (TACACS+)

Description des systèmes d'authentification



OAuth

Standard de délégation d'accès sécurisé

Caractéristiques

- ✓ Utilise des tokens pour l'accès
- ✓ Ne nécessite pas les identifiants directs

Exemple

Accès aux fichiers Google Drive via un token, sans mot de passe



OpenID Connect

Couche d'identité sur OAuth 2.0

Différence avec OAuth

→ Focus sur l'authentification plutôt que la délégation d'accès

Exemple

Authentification via Google Login avec récupération du profil



SAML

Protocole XML pour authentification fédérée

Caractéristiques

Permet le Single Sign-On pour accéder à plusieurs applications



Kerberos

Protocole d'authentification basé sur les tickets

Composants

🔑 Clés secrètes

🛡️ Tiers de confiance



RADIUS et TACACS+

Protocoles d'authentification réseau

RADIUS

Authentification, autorisation et audit des connexions distantes

TACACS+

Séparation fine entre authentification et autorisation



Pour résumer :

Les systèmes d'authentification sont essentiels pour garantir que seules les identités vérifiées peuvent accéder aux systèmes et ressources critiques. Chaque protocole (OAuth, OIDC, SAML, Kerberos, RADIUS, TACACS+) joue un rôle spécifique dans des contextes différents (authentification locale, fédérée ou distante). En particulier, OAuth permet la délégation d'accès sécurisée, tandis qu'OpenID Connect se concentre sur l'authentification utilisateur.