Practica Final

Control d’accés dinàmic basat en intel·ligència artificial: cap a un sistema autoadaptatiu i segur

Alumnes: Hamza El Haddad Sabri i Oscar Saborido Valdes

Data: 11/05/2025-01/06/2025

ÍNDEX

1. Introducció

Actualment, la **ciberseguretat** s'enfronta al repte de gestionar l'accés als sistemes en un entorn digital en constant evolució. **Els models de control d'accés tradicionals** (DAC, MAC, RBAC), basats en regles estàtiques, són insuficients a causa de l'auge del **cloud computing, l'IoT, el teletreball i els atacs avançats** que comprometen credencials. De fet, les credencials robades van ser la principal causa de bretxes de seguretat el 2023.

Aquesta situació ha impulsat el paradigma **Zero Trust** ("Mai confiar per defecte, verificar sempre"), popularitzat per la **NIST SP 800-207**. Aquest model exigeix la **validació contínua** de cada sol·licitud d'accés utilitzant **informació de context en temps real** (estat del dispositiu, ubicació, hora, historial de comportament, etc.). No obstant això, implementar un control d'accés contextual i dinàmic requereix una **automatització i intel·ligència** que va més enllà de les capacitats humanes o les regles estàtiques.

És aquí on la **intel·ligència artificial (IA)** esdevé crucial. Aplicada a la gestió d'identitats i accessos (**IAM**), la IA, mitjançant tècniques d'aprenentatge automàtic, permet **extreure patrons de dades d'accés, detectar desviacions del comportament habitual i predir sol·licituds malicioses o d'alt risc.** Conceptes com l'**Anàlisi del Comportament d'Usuaris i Entitats (UEBA)** utilitzen algoritmes per identificar activitats inusuals que puguin indicar la presència d'un intrús amb credencials legítimes. A més, productes comercials com Azure AD Conditional Access, Okta Adaptive MFA i Google Cloud Context-Aware Access ja implementen **autenticació adaptativa**, ajustant dinàmicament els requisits de verificació segons factors de risc analitzats per models d'IA.

Aquest treball proposa un model de control d'accés híbrid que combina **ABAC (basat en atributs)** amb **intel·ligència artificial**. L'objectiu és superar les limitacions dels models tradicionals en entorns complexos com el núvol o l'IoT, i la manca de reacció contextual o la dificultat de gestió dels models existents. La IA analitzarà factors com l'hora, la ubicació o el comportament per calcular un risc i **ajustar les polítiques d'accés ABAC en temps real**. El treball inclou un exemple pràctic i aborda els **reptes ètics, legals i tècnics** d'aquesta implementació, com la privadesa, la transparència, el biaix algorítmic i la integració amb infraestructures existents.

2. Marc teòric: Models de control d'accés

Cada model té uns principis de disseny particulars i s'adapta millor a determinats escenaris. Els models que tractarem són:

- **DAC (Discretionary Access Control)** – Control d'accés discrecional.
- **MAC (Mandatory Access Control)** – Control d'accés obligatori.
- **RBAC (Role-Based Access Control)** – Control d'accés basat en rols.
- **ABAC (Attribute-Based Access Control)** – Control d'accés basat en atributs.
- **PBAC (Policy-Based Access Control)** – Control d'accés basat en polítiques.
- **Zero Trust** – Paradigma modern de "confiança zero" (no és exactament un model clàssic d'AC, però és rellevant considerar-lo en aquest marc teòric pel seu enfoc dinàmic).

Cada subapartat definirà el model, en destacarà el funcionament, **avantatges** i **limitacions**, i establirà la base per a discussions posteriors sobre la seva evolució i aplicació conjunta amb IA.

2.1 Discretionary Access Control (DAC)

El **control d'accés discrecional (DAC)** és un dels models més antics i senzills. S'anomena "discrecional" perquè deixa certa discrecionalitat o llibertat a l'**propietari dels recursos** per determinar qui hi pot accedir. En un sistema DAC, cada objecte (per exemple, un fitxer, una base de dades, etc.) té un propietari, i és aquest propietari qui pot concedir o revocar permisos a altres subjectes (usuaris) sobre el seu objecte. En altres paraules, la política d'accés es basa en **llistes de control d'accés (ACLs)** definides per l'amo de la informació.

Un exemple típic de DAC és el sistema de fitxers d'Unix: cada fitxer té un propietari i un grup associat, i es defineixen permisos de lectura, escriptura i execució per al propietari, el grup i altres. El propietari (i l'administrador del sistema) pot modificar aquests permisos a voluntat. Un altre exemple seria compartir un document en serveis com Google Drive: l'usuari creador decideix qui més pot veure o editar el document (aquest és un control discrecional atorgat pel propietari).

Avantatges del DAC: la simplicitat i la **flexibilitat**. En tant que es recolza en decisions descentralitzades (cada propietari gestiona els accessos als seus objectes), el model és intuïtiu i fàcil d'implementar en entorns senzills. Permet un **grau de granularitat** elevat, ja que es poden assignar permisos a nivell d'usuari individual per cada recurs. Aquesta flexibilitat és útil

en entorns on els usuaris generen informació i volen controlar-ne la difusió (per exemple, en sistemes de compartició de fitxers dins d'una organització).

Limitacions del DAC: Precisament la seva dependència en les decisions dels propietaris pot derivar en **mancances de control centralitzat**. En organitzacions grans, DAC pot dificultar la visibilitat global de qui té accés a què, ja que les permissivitats es poden dispersar i canviar constantment sense coneixement dels administradors de seguretat. Això facilita problemes de **privilegis excessius** (overprivilege) i **incoherències**: un usuari pot acabar tenint accés a un recurs crític simplement perquè algun propietari li va concedir, fins i tot si això contradiu les polítiques generals de l'organització. DAC també és vulnerable a **atacs de troià**: un usuari amb dret de lectura sobre un fitxer pot, per exemple, copiar-lo i atorgar accés a un altre usuari no autoritzat originalment, ja que el model ho permet (l'usuari pot redistribuir la informació al seu criteri). Aquesta possibilitat de **transferència d'informació** és inherent al DAC i és precisament el que models com MAC busquen evitar.

En resum, DAC es caracteritza per un enfocament basat en **ACLs gestionades pels propietaris**. És adequat per a entorns col·laboratius petits o moderats on la facilitat de compartir informació és prioritària i els riscos són controlats. Però en sistemes d'alta seguretat o grans corporacions, DAC pot resultar insuficient, ja que no garanteix enforcement uniforme ni restriccions que vagin més enllà del criteri dels usuaris individuals.

2.2 Mandatory Access Control (MAC)

El **control d'accés obligatori (MAC)** adopta una filosofia gairebé oposada al DAC. En un sistema MAC, els accessos es regulen segons **polítiques centralitzades** definides per l'organització (o l'autoritat central) i *no poden ser alterades pels usuaris individuals*. El nom "obligatori" reflecteix que tots els subjectes i objectes estan *sotmesos obligatòriament* a les regles de control definides, sense discrecionalitat per part dels propietaris de la informació.

Històricament, MAC s'ha associat als sistemes militars i governamentals, on la informació es classifica per **nivells de sensibilitat** (per exemple: Confidencial, Secret, Altament Secret) i les persones tenen **nivells de classificació** o autoritzacions de seguretat (clearance). En un model MAC clàssic, a cada document se li assigna una etiqueta de classificació, i a cada usuari una autorització; l'accés només es concedeix si l'autorització de l'usuari és igual o superior al nivell de classificació del document, i alhora hi ha una necessitat de conèixer (need-to-know). Un exemple conegut és el model de seguretat de Bell-LaPadula per a **confidencialitat**, que estipula regles com "*no read up, no write down*": un usuari no pot llegir informació per sobre del seu nivell, ni escriure informació a un nivell per sota (per evitar filtracions).

Característiques clau del MAC:

- **Uniformitat i centralització:** Les polítiques afecten tot el sistema de manera consistent. Els usuaris no poden canviar les etiquetes de seguretat dels objectes, ni concedir permisos a altres arbitràriament.

- **Política basada en etiquetes de seguretat:** Sovint implementat mitjançant etiquetatge de subjectes i objectes amb metadades de seguretat (p. ex. les etiquetes de classificació).

- **Nondiscrecionalitat:** De fet, MAC és considerat un tipus de control d'accés **no discrecional** (de vegades així es denomina, contraposant-lo al discrecional) perquè elimina la discreció de l'usuari en la propagació de permisos.

Avantatges del MAC: Proporciona un alt nivell de **control i seguretat** en entorns on això és crític. En restringir que els usuaris canviïn permisos o transfereixin informació a voluntat, es redueixen fuites accidentals o malintencionades. MAC fa que el sistema sigui coherent amb les polítiques d'alt nivell: per exemple, en un entorn governamental, cap document classificat com "Secret" podrà ser llegit per un usuari sense acreditació "Secret", punt. També evita l'**escalat de privilegis** per social engineering o errors humans, ja que ni tan sols el propietari d'un document pot saltar-se les regles (ell no pot rebaixar la classificació del document per compartir-lo amb algú sense permís, per exemple).

Limitacions del MAC: És un model **inflexible i costós de gestionar** en entorns comercials o oberts. La rigidesa de MAC, que és avantatge en seguretat, resulta en desavantatge en *usabilitat*. Pocs entorns civils funcionen amb MAC pur, perquè implica que totes les dades i usuaris han de ser classificats i controlats per una autoritat central i qualsevol canvi (p. ex., donar accés a un col·laborador extern a un document) esdevé burocràtic i difícil. A més, definir correctament les etiquetes i mantenir un sistema MAC consistent requereix una planificació complexa. En entorns dinàmics (pensem en una empresa on els projectes canvien i els equips col·laboren de forma transversal), MAC pot entorpir l'agilitat. Un altre inconvenient és que MAC tracta bé escenaris de **confidencialitat** (evitar filtracions), però pot ser menys natural per altres criteris d'autorització (per exemple, polítiques basades en context, o restriccions temporals) a menys que es complementi amb mecanismes addicionals.

Un exemple modern on es percep influència de MAC és en sistemes operatius mòbils o virtualització: per exemple, l'**Android Security Model** etiqueta aplicacions i recursos de manera que una app no pugui accedir a dades d'una altra (sandboxing), sense intervenció de l'usuari. El mateix passa en sistemes Linux amb *SELinux* o *AppArmor*, que implementen polítiques obligatòries de quins processos poden accedir a quins arxius més enllà dels permisos Unix estàndard.

En resum, MAC ofereix **màxima seguretat per mitjà de polítiques centrals i invariants**, però a costa de la flexibilitat. És adequat quan la protecció de la informació és prioritària i s'accepta sacrificar comoditat (p. ex. entorns governamentals, infraestructures crítiques). En la majoria d'entorns corporatius, però, un MAC pur no és pràctic, i per això han sorgit models com RBAC i ABAC que busquen un equilibri.

2.3 Role-Based Access Control (RBAC)

El **control d'accés basat en rols (RBAC)** va ser proposat formalment a inicis dels anys 90 (Ferraiolo i Kuhn, 1992) i ràpidament es va convertir en el paradigma dominant en la indústria per a la gestió de privilegis. En RBAC, les decisions d'accés es prenen en funció de **rols** que representen funcions o llocs de treball dins d'una organització. En lloc d'assignar permisos directament a cada usuari (com es faria en DAC), en RBAC els permisos s'assignen als **rols**, i els usuaris són associats (assignats) a un o diversos rols segons la seva feina. Això reflecteix més de prop l'estructura organitzativa i simplifica l'administració: per exemple, es pot crear un rol "Administratiu de base de dades" amb permisos de lectura/escriptura a les bases de dades X i Y; llavors n'hi ha prou d'assignar aquest rol a cada nou administrador de BD contractat, en comptes d'haver de configurar-li desenes de permisos individuals.

Funcionament bàsic:

- Definició de rols (p. ex. *administrador*, *analista*, *operador*, *visor*, etc.).
- Assignació de **permisos** (accions permeses sobre determinats recursos) a cada rol.
- Assignació de **usuaris** a rols (cada usuari hereta els permisos dels rols que posseeix).

El RBAC suporta també conceptes més avançats com **jerarquies de rols** (rols pares i fills, on un rol sènior inclou permisos dels rols subordinats), restriccions de separació de funcions (**SoD, Separation of Duties**, per evitar que un usuari tingui dos rols que combinats suposin un risc, com per exemple rol de comptable i rol d'auditor alhora), i rols dinàmics activats per sessió. El model RBAC estandarditzat (ANSI INCITS 359-2004) defineix diferents nivells de refinament d'aquests conceptes.

Avantatges del RBAC:

- **Simplicitat d'administració en escenaris grans:** en reduir dràsticament el nombre d'assignacions. Normalment, els rols es defineixen segons les responsabilitats laborals, i això encaixa amb el que *de facto* es fa servir per decidir qui pot fer què. Així, si un empleat canvia de departament, simplement se li canvien els rols enlloc de reconfigurar tots els accessos.
- **Principi de mínim privilegi:** bé dissenyat, RBAC facilita que els usuaris tinguin només els permisos necessaris, ja que aquests vénen determinats pel seu rol, que correspon a les tasques que han de realitzar. A més, és més fàcil auditar i explicar els privilegis ("l'usuari X té accés a aquest recurs perquè és *Responsable de Projecte* i aquest rol ho permet").
- **Compliment i regulacions:** molts estàndards i lleis demanen controls d'accés basats en rols per assegurar separació de funcions i traçabilitat. RBAC es va fer famós en part perquè encaixava amb requisits d'auditoria (p. ex. SOX, HIPAA, etc. requereixen control

sobre qui pot accedir a dades sensibles; tenir rols ben definits ajuda a demostrar-ho).

Limitacions del RBAC:

- **Manca de granularitat i context:** RBAC per si sol considera només l'atribut "rol" de l'usuari, i potser alguns conceptes d'estructura (rols jeràrquics). No té en compte altres atributs com la ubicació, hora, estat del sistema, projectes específics, etiquetes de dades, etc. Per tant, no pot expressar polítiques del tipus "Només permet accés si l'usuari és *Administrador* **i** està connectat des de la xarxa interna en horari laboral" (això RBAC pur no ho contempla, caldria complementar-ho o anar a ABAC).

- **Problemes de creixement de rols:** A mesura que les organitzacions creixen i les tasques es diversifiquen, RBAC pot patir **explosió de rols**. Si es defineixen rols molt específics per cobrir cada combinació de permisos necessària, aviat es pot acabar amb centenars de rols, fent complexa la gestió (qui té quin rol) i propiciant errors. Alternativament, si es fan rols massa amplis, llavors els usuaris acaben amb més privilegis dels necessaris (problema de *privilege creep*).

- **Manteniment i actualització:** RBAC reflecteix l'estructura organitzativa en el moment de dissenyar els rols. Però les organitzacions són dinàmiques; nous projectes, noves responsabilitats i reorganitzacions impliquen revisar constantment els rols i les assignacions. Això pot esdevenir un procés costós, i de vegades els departaments de TI no aconsegueixen mantenir els rols perfectament actualitzats al ritme dels canvis de negoci, la qual cosa porta a desviacions (gent que conserva rols antics que ja no necessita, etc.).

Malgrat aquests reptes, RBAC és **extremadament popular** i està implementat en multitud de sistemes: sistemes operatius (p. ex. Windows AD a través de grups de seguretat equival a rols), gestors de bases de dades, eines de planificació de recursos (ERP), etc. Molts productes cloud també parteixen d'RBAC (per exemple, *Azure* i *AWS* tenen rols o grups als quals s'assignen permisos sobre recursos cloud). Precisament, part de la discussió posterior serà com RBAC s'estén o complementa per guanyar dinamisme (per exemple, RBAC + condicions context = naixement de ABAC).

2.4 Attribute-Based Access Control (ABAC)

El **control d'accés basat en atributs (ABAC)** representa una generalització i evolució del RBAC per aportar més **flexibilitat i granularitat** a les decisions d'autorització. En ABAC, en lloc de basar-nos únicament en rols o en identitats específiques, es defineixen polítiques d'accés que avaluen un conjunt d'**atributs** del subjecte (usuari o procés que sol·licita accés), de l'objecte (recurs al qual vol accedir), de l'acció (el tipus d'operació sol·licitada) i de l'entorn (context, com l'hora, localització, etc.). Aquest model també es coneix com a **control d'accés**

basat en polítiques (PBAC en sentit ampli), tot i que com veurem, PBAC es pot considerar un terme gairebé sinònim o almenys molt proper.

En ABAC, una **política d'accés** típicament té la forma de regles lògiques sobre atributs, per exemple:

- "Permetre l'accés als fitxers amb etiqueta 'Projecte Alfa' a usuaris el departament dels quals = 'Projecte Alfa' i el nivell de *clearance* ≥ 3, excepte si l'entorn *localització* = extern."

- "Autoritzar la invocació de l'API X només si *rol* del subjecte = *admin* **i** *MFA autenticada* = true **i** *hora* dins [8:00-18:00]."

Els atributs poden ser qualsevol propietat pertinent: dades de l'usuari (edat, departament, rol, antiguitat, certificacions...), propietats dels objectes (propietari, etiquetes de classificació, tipus de recurs...), paràmetres d'entorn (IP d'origen, zona geogràfica, estat de la xarxa, nivell d'amenaça global, etc.). Això permet decisions **molt més contextuals i dinàmiques** que RBAC. De fet, sovint es diu que ABAC implementa millor el principi de *need-to-know* i de *least privilege*, perquè podem refinar exactament sota quines condicions s'autoritza cadascuna de les accions.

Avantatges de l'ABAC:

- **Granularitat i expressivitat:** Com ja s'ha apuntat, amb ABAC es poden expressar polítiques complexes que incloguin múltiples factors. Això permet controlar accessos de forma **fina i condicional** sense haver de crear rols específics per a cada combinació de condicions (un problema d'RBAC). Per exemple, si demà l'organització vol afegir la condició "i l'usuari ha completat la formació anual de seguretat" als accessos a dades sensibles, en ABAC només cal afegir aquest atribut a la política; en RBAC potser caldria crear un rol nou per distingir els que han fet la formació.

- **Adaptabilitat:** Com que les polítiques es basen en atributs i aquests es poden actualitzar fàcilment, ABAC s'adapta bé a canvis. Afegir un nou usuari amb determinades característiques implica que automàticament s'aplicaran les polítiques pertinents sense haver de reconfigurar permisos específics – n'hi ha prou que se li assignin els atributs adequats (p. ex., departament = X, funció = Y) i ja heretarà els accessos definits per la política.

- **Reducció de rols i combinacions explosivess:** En molts casos, ABAC pot eliminar la necessitat de tenir desenes de rols, perquè les diferents casuístiques es representen amb condicions d'atributs. Es passa de pensar en "qui és l'usuari i quin rol té" a "quines propietats compleix l'usuari i el recurs, i en base a això decideixo". Això simplifica l'administració a gran escala (tot i que introdueix altres reptes, com veurem).

Desafiaments i limitacions de l'ABAC:

- **Complexitat de definició i gestió:** La potència d'ABAC ve amb el preu de major complexitat en definir i mantenir les polítiques. A diferència de RBAC on moltes coses es poden fer textualment ("rol Administrador pot fer X"), en ABAC cal especificar condicions lògiques que poden ser complexes, i assegurar que hi ha coherència entre elles. Sovint es requereix utilitzar llenguatges com **XACML (eXtensible Access Control Markup Language)** per expressar les polítiques, que és potent però verbós i difícil de manegar manualment a gran escala. Algunes fonts indiquen que ABAC "requereix més recursos de TI i desenvolupament a mesura que creix el nombre d'atributs" – fer referència a l'esforç de gestionar esquemes d'atributs i polítiques.

- **Decisions difícils d'explicar:** Amb ABAC pur, pot ser menys intuïtiu per a un usuari o administrador comprendre per què a algú se li va denegar l'accés. Amb RBAC és fàcil dir "no tens el rol necessari". Amb ABAC, la decisió pot dependre de múltiples condicions (potser l'usuari tenia 3 atributs correctes però li en falta un quart). Això pot requerir eines de diagnòstic per depurar polítiques i comunicar clarament els motius d'una denegació.

- **Rendiment i avaluació contínua:** ABAC implica avaluar múltiples condicions en temps real per cada petició. En entorns distribuïts i de molt alt volum, pot ser un repte optimitzar aquest procés. Tot i que normalment l'avaluació lògica és ràpida, cal un bon disseny de **infraestructura (PDP, PEP)** per no introduir latència. També s'ha de tenir els atributs actualitzats i disponibles a l'hora de la decisió (p. ex., si l'atribut "estat del dispositiu" es consulta a un sistema extern, podria generar retards).

Malgrat aquests reptes, ABAC ha estat reconegut com un model capaç de resoldre necessitats de **seguretat moderna**. La publicació **NIST SP 800-162 (Guide to ABAC)** defineix ABAC i en promou l'adopció per a entorns federats i heterogenis on RBAC queda curt. ABAC es troba en sistemes com la gestió d'IAM de **Google Cloud** (que permet afegir condicions als permisos IAM basades en atributs de context, com la ubicació de la petició o etiquetes de recurs), també en solucions de bases de dades (p. ex. SQL: es poden fer vistes condicionals per atributs d'usuari), i en general és la base conceptual sota llenguatges de política utilitzats en microserveis i APIs (com ara Open Policy Agent, que és una implementació de motor de polítiques tipus ABAC per a infraestructura cloud).

Exemple: Suposem un entorn de sanitat. Amb RBAC, potser definiríem un rol "Metge" que té accés als expedients mèdics. Però potser la política real desitjada és "un metge només pot veure expedients dels pacients que estan sota la seva atenció i només mentre estiguin ingressats". Expressar això amb RBAC és molt difícil (hauríem de crear rols per metge/pacient o fer mil combinacions), mentre que amb ABAC es pot fer: atribut *pacient.assignat_a* = *ID_metge* i *pacient.estat* = *ingressat*. Així, quan un metge demana accés a un expedient, el motor comprova si l'atribut "metge_assignat" del pacient coincideix amb l'ID del metge sol·licitant i que el pacient estigui ingressat; només en cas afirmatiu s'autoritza. Quan l'estat del

pacient canvia o es reassigna el metge, aquestes propietats canviaran i automàticament les polítiques actuaran diferent. Aquest nivell de granularitat és exactament el punt fort d'ABAC.

2.5 Policy-Based Access Control (PBAC)

El terme **control d'accés basat en polítiques (PBAC)** sovint es fa servir de manera solapada amb ABAC, ja que ABAC és en el fons un tipus de PBAC. En general, PBAC indica qualsevol model on les decisions es prenen segons **polítiques declaratives** definides, en lloc de llistes discrecionals o rols estrictes. Alguns autors i proveïdors utilitzen PBAC per referir-se a enfocaments híbrids que combinen rols i atributs. De fet, la **NIST CSRC Glossary** defineix PBAC com "una forma de control d'accés flexible en els tipus de paràmetres que avalua (identitat, rol, autorització, necessitat operativa, risc, heurístiques…)". Això és força ampli: inclou RBAC, ABAC i més.

Tanmateix, per donar-li un context més específic, aquí parlarem de PBAC com aquella filosofia on un sistema utilitza un motor central de **polítiques** per determinar els accessos, possiblement combinant múltiples criteris, incloent-hi rols. Alguns veuen PBAC com un **RBAC augmentat amb regles**. Per exemple, una política PBAC podria dir: "Si l'usuari té el rol *Enginyer* **i** està en el grup de projecte X **i** el directiu del projecte ha aprovat l'accés, llavors permet l'accés al repositori Y". Això involucra rols (Enginyer), atributs (grup = X) i una condició externa (aprovació).

En la literatura, també existeix una variant anomenada **Purpose-Based Access Control** (a vegades també PBAC) que es refereix a controlar l'accés segons la *finalitat* per la qual s'accedeix a la dada (això es veu en privacitat: un metge pot accedir a l'historial mèdic d'un pacient **amb la finalitat** de tractament, però no per curiositat – i el sistema ha de registrar o controlar la finalitat declarada). No confondrem amb aquest concepte, tot i ser interessant, sinó que ens mantindrem en PBAC com a *Policy-Based*.

Característiques i beneficis de PBAC (Policy-Based):

- **Flexibilitat:** Igual que ABAC, permet regles fines. De fet, es pot considerar que ABAC és un tipus de PBAC on els paràmetres són atributs. PBAC podria englobar també polítiques basades en *riscos*, *contextos*, *regles expertes*, etc.

- **Visibilitat i consistència:** Un punt destacat per alguns venedors és que les polítiques en PBAC es poden escriure en un format relativament **humà-llegible** (segons implementació) i centralitzar en un repositori, proporcionant millor **governança**. Per exemple, es pot tenir un repositori on es puguin consultar totes les polítiques i saber quina relació hi ha entre identitats i recursos, millorant el compliment normatiu.

- **Adaptabilitat a canvis:** Afegir, treure o modificar una política pot ser més senzill que reprovisionar molts rols o permisos individuals. A més, si l'entorn canvia (nous tipus de recursos, noves condicions de seguretat), només cal actualitzar o afegir polítiques sense

reestructurar tot el model.

Consideracions i reptes de PBAC:

- **Solapament amb ABAC:** Realment PBAC no aporta res conceptualment nou respecte a ABAC, més aviat és un terme que recalca l'ús de polítiques com a centre. Moltes implementacions d'ABAC es publiciten com PBAC per subratllar aquesta naturalesa.

- **Eines de gestió:** Per implementar PBAC es requereixen eines capaces d'avaluar polítiques en temps real. Existeixen estàndards com **XACML** (esmentat) o **Rego** (llenguatge de polítiques d'Open Policy Agent) que faciliten aquest enfoc. Però llavors l'organització necessita integrar aquestes eines amb les aplicacions existents.

- **Rendiment i escalabilitat:** Similar a ABAC, ja que PBAC pot implicar regles complexes, cal assegurar que el motor de polítiques està ben optimitzat i pot escalar horitzontalment, sobretot en entorns de microserveis on cada petició pot invocar una avaluació de polítiques.

Podem veure PBAC en acció en solucions com **Next-Generation Access** de vendors que combinen RBAC amb condicions. Un exemple: **AWS** progressa cap a PBAC oferint *AWS IAM policies* que són essencialment regles JSON que combinen atributs de recursos, condicions d'hora, IP, etc. (tenen claus de condició). **Okta** parla de "polítiques d'accés adaptatives" també, on es defineixen condicions (com que un login sigui des d'un dispositiu gestionat) per permetre l'accés. Això són casos de PBAC.

En conjunt, PBAC reforça la idea que la gestió de l'accés ha evolucionat d'assignar permisos explícits o implícits via rols, a *escriure regles altes* que el sistema aplica uniformement. Aquesta transició és important de cara a la incorporació de IA: un sistema de control d'accés basat en polítiques pot ser més fàcilment alimentat per decisions o recomanacions d'IA, ja que hi ha un punt centralitzat (el motor de polítiques) on es pot injectar lògica adaptativa.

2.6 Zero Trust i control d'accés continu

Tot i no ser un "model de control d'accés" clàssic en el sentit de DAC, RBAC, etc., **Zero Trust** és un paradigma modern estretament relacionat que cal incloure en aquest marc teòric perquè introdueix els conceptes de **verificació contínua i dinàmica** que justament motiven l'ús d'IA en control d'accés.

Zero Trust Architecture (ZTA), formalitzat per NIST SP 800-207 el 2020, es basa en el principi que *cap petició d'accés, interna o externa, ha de ser confiada per defecte*. Tradicionalment, molts sistemes han confiat en un **perímetre de seguretat**: un cop un usuari o dispositiu es valida i entra a la xarxa corporativa, hi ha implícitament més confiança (es pot moure lliurement per segments interns). Zero Trust trenca amb això i diu: *"autentica i autoritza cada accés com si*

vingués d'una xarxa insegura". Cada vegada que un usuari vol accedir a un recurs, cal **tornar a avaluar-ho tot**: identitat, credencials, context, postura de dispositiu, etc., i aplicar polítiques de menor privilegi.

En termes de control d'accés, Zero Trust impulsa varis aspectes:

- **Autenticació i autorització contínues i dinàmiques:** no n'hi ha prou amb autenticar-se un cop a l'inici de sessió. El sistema ha de re-validar constantment, especialment a cada canvi de context o accés a un recurs nou. NIST ho expressa així: *"All resource authentication and authorization are dynamic and strictly enforced before access is allowed"*. És a dir, decisions dinàmiques i enforcement estricte en tot moment.

- **Visibilitat total i criteris contextuals:** Zero Trust recomana recollir tanta informació com es pugui sobre l'estat de dispositius, xarxa, activitat, i usar-la per millorar les decisions d'accés. Això significa que atributs d'entorn i comportament (com hem vist a ABAC) esdevenen centrals.

- **Polítiques centrades en actius i identitats:** cada accés depèn de *qui* (identitat, incloent usuari i procés), *què* (recurs al que s'accedeix, valorat com a actiu), *com* (mètode de connexió), *estat del qui i el què* (estat del dispositiu, postura de seguretat del recurs). Sona molt semblant a ABAC + altres elements.

Zero Trust no prescriu exactament com han de ser les polítiques d'accés, però en general implementacions Zero Trust opten per models **ABAC/PBAC amb fort ús de context i sovint amb **avaluació de risc en temps real**. Per exemple, sistemes coneguts com **Google BeyondCorp** (pioner de Zero Trust) classifiquen el nivell de confiança de cada dispositiu d'un empleat i atorguen o deneguen accessos a aplicacions internes en funció d'això i de l'identitat de l'usuari. Un dispositiu sense parchejar o fora de la gestió corporativa tindrà restriccions malgrat que l'usuari sigui legítim.

IA i Zero Trust: Zero Trust genera un gran volum de dades a analitzar (telemetria contínua de xarxa, autenticacions, estat de paràmetres de seguretat) i decisions que poden ser complexes. La IA encaixa aquí com a eina per:

- **Detectar anomalies** dins d'aquesta telemetria (per exemple, identificar que tot i complir polítiques, un usuari es comporta de manera atípica podria desencadenar mesures addicionals).

- **Avaluar risc en temps real:** algunes implementacions parlen de *"risk engines"* que fan scoring de cada accés. Per exemple, si un usuari fa 5 accions ràpides que solen precedir un exfiltration (com accedir a molts arxius sensibles en poc temps), un motor d'IA podria assignar un risc alt a la sessió i temporalment reduir permisos o tancar la sessió (això seria un comportament autoadaptatiu ideal).

- **Orquestrar respostes automàtiques:** en línia amb *CARTA (Continuous Adaptive Risk and Trust Assessment)* de Gartner, on el sistema de seguretat adapta els controls segons puja o baixa el nivell de confiança.

En resum, Zero Trust aporta els conceptes que justifiquen la necessitat d'un control d'accés **més dinàmic i intel·ligent**. És un catalitzador que combina els models ja vistos (perquè en Zero Trust seguirem tenint rols, atributs, etc.) però demana portar-los al següent nivell: cada accés condicionat pel context i reevaluat contínuament. La unió de ABAC/PBAC amb IA és, en gran mesura, una manera de donar vida als principis de Zero Trust de forma escalable i automatitzada.

Abans de concloure aquest marc teòric, cal notar que també existeixen models emergents o complementaris, com el **Risk-Adaptive Access Control (RAdAC)**, que formalitza la idea d'incloure la noció de *risc* en la decisió d'accés. RAdAC essencialment diu que l'autorització no ha de dependre només de si tens permisos (p. ex. rol) sinó també del context de risc: potser normalment tens accés, però si avui el risc és alt (diguem, la teva conducta és anòmala o hi ha alertes de seguretat) se't limiti. Aquest model, proposat inicialment pel DoD, no ha estat àmpliament implementat encara, però conceptualment és molt proper al que discutirem en integrar IA per avaluar risc.

Amb aquests conceptes teòrics clars – DAC i MAC com a extrems oposats de discrecionalitat, RBAC aportant gestió basada en rols, ABAC/PBAC aportant flexibilitat basada en atributs i polítiques, i Zero Trust/RAdAC introduint el context dinàmic de risc – ja tenim el vocabulari i la comprensió bàsica per analitzar l'estat actual de la IA en control d'accés i posteriorment descriure la nostra proposta de model híbrid.

3. Estat actual de la IA aplicada al control d'accés

Els avenços en **intel·ligència artificial** i aprenentatge automàtic han trobat una aplicació natural en l'àmbit de la seguretat, i concretament en la gestió d'identitats i control d'accés. En els darrers anys hem vist proliferar tant iniciatives de recerca com solucions comercials que incorporen tècniques d'IA per **millorar la presa de decisions d'accés, detectar usos anòmals i automatitzar la configuració de permisos**. En aquesta secció revisarem l'estat actual: les **tècniques principals** emprades, alguns **casos d'ús reals** destacats i productes disponibles al mercat, i fins a quin punt aquestes solucions han resolt els reptes descrits o quin grau de maduresa tenen.

3.1 Tècniques d'IA en control d'accés i identitats

Les tècniques d'IA aplicades en aquest camp es poden agrupar en diverses categories:

- **Aprenentatge supervisat per classificació de risc:** Consisteix a entrenar models (p. ex. xarxes neuronals, arbres de decisió, random forest, etc.) que **classifiquin les sol·licituds d'accés en categories** com "permès", "bloquejat", o assignar-los un nivell de risc numèric (0-100, o baix/mitjà/alt). Aquests models s'entrenen amb dades històriques on hi ha exemples de comportaments legítims i maliciosos. Per exemple, Microsoft ha indicat que els seus sistemes d'Identity Protection utilitzen **algoritmes adaptatius** per detectar inici de sessió arriscats basant-se en milions d'intents recollits globalment. Característiques que alimenten aquests classificadors podrien ser: geolocalització de l'IP, hora del dia, coincidència amb patrons coneguts d'atac (IP llistada com a maliciosa, impossible travel, etc.), tipus de dispositiu i reputació, etc. Un cop entrenat, el model pot, en temps real, puntuar cada intent d'accés. Azure AD Identity Protection, per exemple, assigna un *user risk level* i *sign-in risk level* a cada usuari i sessió, respectivament, emprant **machine learning i heurístiques**. Si el risc supera un llindar, una política condicional pot bloquejar o exigir MFA.

- **Aprenentatge no supervisat i detecció d'anomalies:** Aquí no es classifica en categories predefinides, sinó que es busca identificar patrons inusuals respecte a la normalitat. En control d'accés, això pren la forma de **User and Entity Behavior Analytics (UEBA)**, que construeix un perfil de comportament normal per a cada usuari (i possiblement dispositiu o compte de servei) i llança alarmes quan hi ha desviacions significatives. Algunes tècniques inclouen clustering, estimació de densitat (p. ex. algoritmes com Isolation Forest, LOF - Local Outlier Factor) o models estadístics (modelatge Gaussià, etc.). Per exemple, **Okta** ha desenvolupat un sistema de machine learning que aprèn un perfil individual per a cada usuari en termes d'aplicacions accedides, horaris, ubicacions i dispositius usuals, de manera que pot reconèixer quan un intent d'autenticació *no encaixa* amb el patró d'aquell usuari. Això complementa els controls estàtics: potser un usuari té permisos per accedir a una aplicació, però si sobtadament ho intenta des d'un país diferent i en horari estrany, el sistema ho marca com a anòmal i pot demanar una verificació addicional. L'objectiu de la detecció

d'anomalies és trobar **compromisos de comptes** o ús indegut de credencials legítimes. De fet, molts incidents de seguretat interns (per exemple, un empleat abusant dels seus privilegis) només es poden descobrir així, ja que no es viola cap regla d'accés sinó un patró de comportament.

- **Aprenentatge automàtic per a *role mining* i revisió de permisos:** Abans d'abordar IA en l'execució de l'accés (enforcement), cal esmentar que IA també s'aplica a la **configuració d'accés**. Aquí l'ús típic és analitzar els registres d'accés i les assignacions actuals de permisos per extreure coneixement. Un exemple és l'**Automatic Role Mining**: algoritmes (sovint clustering o factorització de matrius) que analitzen quins permisos té cada usuari i tracten de trobar agrupacions òptimes, que corresponen potencialment a rols candidats. També hi ha treballs per extreure **polítiques ABAC automàticament** a partir de logs: Karimi et al. (2021) proposen un mètode per inferir regles ABAC a partir dels accessos realitzats, utilitzant tècniques de clustering sobre vectors d'atributs dels accessos concedits per descobrir patrons que puguin generalitzar-se en forma de polítiques. Aquest tipus d'eines d'IA ajudaria als administradors a definir millors rols o polítiques, identificant també *outliers* (p. ex., detectant que un usuari té un conjunt de permisos únic no compartit – pot ser un cas de privilegi innecessari).

- **Aprenentatge per reforç (reinforcement learning) aplicat a l'accés:** És un camp encara experimental, però hi ha investigació sobre l'ús d'algoritmes de reforç per *aprendre* polítiques òptimes d'autorització que equilibren seguretat i usabilitat. En un entorn simulat, un agent d'aprenentatge per reforç podria rebre recompenses per encertar en permetre accessos legítims i bloquejar accés maliciosos, i càstigs quan es genera un fals positiu o fals negatiu. Amb prou simulació (o fins i tot en línia, si el sistema permet corregir errors), l'agent podria ajustar una política d'accés. Ara com ara, això és teòric i complicat d'entrenar, però en futur podria donar lloc a sistemes adaptatius que ajustin polítiques de forma autònoma.

- **Processament de llenguatge natural (NLP)** per a anàlisi de polítiques i ordres: Un altre angle és usar IA per facilitar la gestió: per exemple, prototips on un administrador pot escriure en llenguatge natural "Permet només als enginyers de QA accedir als repos X i Y fora de l'horari laboral" i un sistema NLP ho tradueix a una política formal. Fins i tot, amb la popularitat dels **xatbots i assistents amb IA generativa** (com ChatGPT o similars), s'està explorant la integració d'aquests en eines IAM perquè ajudin a revisar qui té accés a què ("Mostra'm quins usuaris fora del departament IT tenen accés a la base de dades financera") o recomanar canvis de configuració. Encara és incipient, però mostra com l'IA pot entrar en tots els aspectes: no només en la decisió runtime, sinó també en el *govern* de l'accés.

En resum, a nivell tècnic l'IA pot aparèixer tant en la fase de **disseny/administració** del control d'accés (ajudant a definir rols i polítiques òptimes) com en la fase d'**execució** (decidint o

modulant cada sol·licitud en temps real) i fins i tot en la fase de **monitoratge i resposta** (detectant usos indeguts post-factum i trigant accions).

A continuació veurem casos concrets i productes que implementen aquestes tècniques.

3.2 Casos reals i productes existents

En aquesta subsecció, examinarem alguns exemples representatius de com s'està aplicant la IA en el control d'accés, tant en l'àmbit comercial com en projectes i estudis reals.

Microsoft Entra ID (Azure AD) – Identity Protection i Access Management: Microsoft ha integrat IA en diverses capes del seu servei d'identitat al núvol. Azure AD Identity Protection utilitza **machine learning adaptatiu** per detectar sign-ins de risc. Les fonts de dades inclouen tot l'ecosistema Microsoft (centenars de milions d'autenticacions diàries) i intel·ligència de seguretat global (per exemple, llistes d'IP compromeses, credencials filtrades a la xarxa fosca, etc.). En temps real, quan un usuari inicia sessió, el sistema avalua: hi ha indicis que el compte ha estat compromès? (p. ex., es va trobar la contrasenya en una filtració), la sign-in prové d'un dispositiu infectat conegut, d'una localització estranya, o presenta un patró anòmal? Si així és, Azure AD pot elevar el risc de la sessió a *medium* o *high*. A partir d'aquí, ve l'integració amb el control d'accés: els administradors poden definir **polítiques condicionals de control d'accés basades en aquest nivell de risc**. Per exemple: "bloqueja inici de sessió automàticament si RiskLevel = High" o "si RiskLevel >= Medium, requereix MFA i resetejar contrasenya". Aquest és un clar cas d'IA (detectant anomalies) combinada amb ABAC/PBAC (política condicional) per adaptar dinàmicament l'accés. Cal destacar que Microsoft reporta que aquestes deteccions combinen ML i **heurístiques** (regles expertes), i que són prou sofisticades per detectar fins i tot variacions subtils (per exemple, ús de VPN per emmascarar localització, patrons de bot vs humà en intents de login, etc.).

Además, Microsoft té la funcionalitat de **Privileged Identity Management (PIM)** amb accessos just-in-time on l'IA pot notificar si hi ha elevacions de privilegi inusuals. Encara en vista, també han parlat d'aplicar IA generativa (ChatGPT) per analitzar logs de Azure AD i respondre preguntes de seguretat, la qual cosa evidencia la direcció de mercat: cada cop més *insights* automàtics en IAM.

Okta – Adaptive Multi-Factor i ThreatInsight: Okta, un dels líders en identitat com a servei, va llançar ja al 2019 el seu sistema de **Risk-Based Authentication**. Aquest sistema recull "panoramic insights" de tots els inicis de sessió als seus clients (de manera anonimitzada) per detectar patrons maliciosos. A nivell de producte, Okta ofereix **Adaptive Multi-Factor Authentication (MFA)**: això significa que la necessitat de MFA pot dependre de si l'intent es considera d'alt risc segons l'algoritme. Un usuari que inicia sessió des del seu equip corporatiu habitual i xarxa coneguda pot no rebre desafiaments addicionals (experiència d'usuari suau), mentre que si ho fa des d'un dispositiu nou i IP estranya, se li exigeix MFA. Okta va declarar que utilitza **machine learning amb el gran volum de dades de la seva plataforma** per construir un "perfil robust de comportament" per usuari. A més, ofereixen *Okta ThreatInsight*, que és un servei que aprofita l'aprenentatge de dades agregades de tots els clients d'Okta per

identificar **IP malicioses conegudes o patrons d'atac** i automàticament bloquejar aquests intents abans que arribin a demanar credencials. ThreatInsight empra ML per distingir trànsit legítim de scanners, bots, etc., i permet a les organitzacions aprofitar la intel·ligència col·lectiva (per exemple, si un conjunt d'IP fa atac de *credential stuffing* a diverses companyies, Okta ho veu i pot bloquejar-les per a tots).

Un cas real reportat per Okta va ser la mitigació del frau de peatge telefònic (**toll fraud**) on han emprat ML per detectar patrons d'ús anòmal en trucades VoIP que comprometien comptes – tot i que és un cas lleugerament diferent, demostra la versatilitat d'aplicar IA en seguretat d'identitat.

Google – BeyondCorp Enterprise i Context-Aware Access: Google des de 2014 va implementar el seu propi model zero trust anomenat BeyondCorp per accés intern, i actualment ofereix productes derivats. En l'àmbit de IA aplicada, Google Cloud Identity-Aware Proxy (IAP) i Context-Aware Access permeten definir polítiques en base a **atributs de context** (ubicació, seguretat del dispositiu, grups d'usuari, etc.), que és ABAC. Google també incorpora **Trust scores** per a dispositius: per exemple, un endpoint management de Google assigna un nivell de confiança al dispositiu (basat en si està actualitzat, xifrat, etc.) i això serveix de base a la política. Pel que fa a IA, Google recull una quantitat ingent de telemetria d'usuari (pensem en Gmail, logins a comptes Google, Android, etc.) i entrena models per, per exemple, **detectar sessions sospitoses** (fa anys van reportar que Google Accounts podia detectar amb 99.7% precisiÓ un login fraudulent abans d'obligar a pas addicional). Encara que Google no ho vengui explícitament com a producte, internament i en Google Cloud's Identity Platform hi ha aquesta intel·ligència. Es pot mencionar també **Chronicle** (de Google, abans si fossin Chronicle Security) i productes de Cloud AI que analitzen logs, però això és més en l'àmbit SIEM/UEBA general.

IBM – Watson i access control cognitiu: IBM va promocionar fa uns anys el concepte de *cognitive security*. En particular, **IBM Cloud Identity** oferia una característica anomenada **Adaptive Access** que prometia usar IA per *scoring* de riscos en autenticació, similar als altres. També IBM té recerca en "Watson for cybersecurity" on el sistema pot ajudar a analitzar decisions d'accés i correlacionar incidents. Tot i que la presència al mercat no ha estat tan notòria com Microsoft o Okta, IBM sí ha integrat coses com assessors de risc i monitors de comportament en les seves solucions d'IAM corporatiu.

Productes especialitzats en analítica i governança: Hi ha un segment d'eines que fan servir IA per **Governança d'Identitats** (IGA, Identity Governance and Administration). Per ex., *SailPoint Predictive Identity* clama usar ML per suggerir quins accessos s'han de recovar a un usuari o quin recurs probablement necessita segons gent similar (fent clustering de rols o entitats). També eines de **Seguretat d'entitlements en núvol** (Cloud Infrastructure Entitlement Management, CIEM) usen anàlisi intel·ligent per detectar permisos orfes o perillosos en entorns multinuvol – no tant IA complexa sinó regles i correlacions, però s'hi encaminem.

Recerca acadèmica i prototips reals: A part dels productes comercials, també val la pena esmentar alguns treballs de recerca que demostren casos reals:

- Investigadors han creat prototips on un sistema ABAC aprèn de logs. L'esmentat **Karimi et al. (2021)** van implementar un prototip que, aplicat a logs d'un entorn sanitari, va ser capaç d'extreure regles ABAC que replicaven en bona part les decisions realitzades manualment.

- **Incorporating Behavior in ABAC** (2020) – proposta d'Attribute/Behavior-Based AC que combinava atributs estàtics amb mètriques de comportament derivades per ML. És a dir, afegir a ABAC un atribut com "riskScore" que prové d'un model de comportament.

- **Deep Learning Based Access Control (DLBAC)**: hi ha un article (Zhang et al. 2018) suggerint emprar xarxes neuronals profundes per avaluar decisions d'accés en IoT en base a múltiples senyals. L'anomenen DLBAC i van mostrar en simulació que podia bloquejar comportaments no autoritzats sense definir regles dures a priori.

- **Access control en entorns serverless amb ML**: Un treball de Chad (2025) explora mecanismes dinàmics per a entorns *serverless*, on les instàncies de microserveis són efímeres i escalen. Proposen recollir logs de Kubernetes, fer detecció d'anomalies amb ML i ajustar polítiques en calent per protegir aquests entorns. Això mostra que fins i tot en arquitectures cloud natives s'està pensant en controls adaptatius.

Indústria financera i sanitària: Sectors regulats han començat a provar aquestes aigües. En bancs, per exemple, implementen **fraud detection** als canals digitals que no deixa de ser control d'accés adaptatiu (si es detecta activitat inusual al compte, es bloqueja o demana revalidació). Empreses de targetes de crèdit fan anys que fan servir IA per bloquejar transaccions sospitoses – un paral·lel clar: la targeta pot ser vàlida (credencial vàlida), però la transacció es denega per anomalia (control adaptatiu). Moltes d'aquestes tecnologies s'estan portant a l'IAM intern.

Per últim, mencionar **estàndards i recomanacions**: Organismes com **NIST** també s'han pronunciat. A la publicació **NIST IR 8314 (2021)** sobre *AI for Access Control*, es discuteix com l'aprenentatge automàtic pot ajudar en la presa de decisions i recomana tenir en compte coses com biaixos, etc. També es parla en fòrums de Cloud Security Alliance del concepte de **Continuous Access Evaluation Protocol (CAEP)** per fer avaluació contínua d'accés, on es pot encabir la IA per alimentar senyals contínues.

4. Limitacions dels models tradicionals en entorns moderns

En aquesta secció analitzem per què els models tradicionals de control d'accés (DAC, MAC, RBAC, ABAC estàtic) resulten **insuficients o presenten dificultats** quan s'apliquen a entorns de TI moderns, caracteritzats per l'ús extensiu del **núvol (sovint múltiple)**, la presència massiva de dispositius de l'**Internet de les Coses**, i arquitectures híbrides i altament dinàmiques. Entendre aquestes limitacions ajudarà a motivar la necessitat de models més dinàmics i intel·ligents.

4.1 Entorns multinuvol i híbrids: dispersió de polítiques i identitats

Actualment moltes organitzacions no es limiten a una única infraestructura; poden tenir aplicacions en un **núvol públic (o diversos)** – Amazon Web Services, Microsoft Azure, Google Cloud, etc. – a més de mantenir sistemes **on-premises**. Cada entorn ve sovint amb el seu propi esquema de control d'accés:

- AWS té IAM (Identity and Access Management) amb un model principalment basat en polítiques JSON (rols i polítiques attachades als recursos o principal).

- Azure té RBAC per recursos d'Azure + Azure AD per identitats, amb rols integrats i condicions.

- Google Cloud IAM és similar a AWS amb rols i policies, i també permet condicions (ABAC).

- Sistemes on-premises poden seguir amb Active Directory (RBAC via grups de seguretat) per aplicacions tradicionals.

Problema 1: Coherència de polítiques. Quan una empresa distribueix les seves càrregues en múltiples plataformes, és molt complex garantir que les polítiques d'accés es tradueixin i apliquin de forma coherent a tot arreu. Un usuari pot tenir un cert rol que li dóna permisos en aplicacions on-prem, però cal crear-li un mirall equivalent a cada núvol. Això sovint es fa amb sincronització d'identitats (p. ex. Azure AD Connect sincronitzant AD local amb Azure AD), però igualment, els permisos a recursos cloud (com accés a instàncies, bases de dades al núvol, etc.) s'assignen separadament. Els models tradicionals com RBAC no van ser dissenyats per ser **federats** en molts dominis administratius diferents.

Problema 2: Visibilitat i control centralitzat. Amb tants entorns, els administradors de seguretat perden visibilitat central: què passa si un usuari marxa de l'empresa? Amb RBAC tradicional, li treus els rols en l'LDAP corporatiu, però potser encara té un access key actiu a AWS, o és part d'un grup en un entorn SaaS extern. Aquest fet ha portat a forçar més **governança** (IGA) i integracions IAM (CIEM per cloud), però evidència que el model estàtic

distribuït no escala bé sense molta automatització. Idealment, un model dinàmic podria fer **enforcement universal**: independentment d'on vagi la petició (núvol X o Y), es consulta un cervell central de polítiques. Però avui, sense aquest cervell, s'acaba duplicant regles i havent de confiar en processos manuals o semiautomàtics de *provisioning*/*deprovisioning*.

Problema 3: Entorns efímers i de microserveis. A la multinuvol s'hi afegeix l'arquitectura de microserveis i *DevOps*. Abans, els recursos (servidors, aplicacions) eren força estables, i s'hi podien assignar rols definits. Avui, tenim contenidors que neixen i moren en minuts, funcions *serverless* que s'executen sota demanda. Com assignem permisos en aquests casos? Molts cops es tendeix a donar rols amplis als microserveis per no bloquejar funcionalitats, però això va contra el mínim privilegi. Els models tradicionals podrien aplicar-se (p. ex. crear identitats de servei per microservei), però la quantitat i dinamisme és tan gran que la gestió manual és impracticable. S'han vist casos de **role explosion** no ja per usuaris humans, sinó per comptes de servei i entitats de microserveis. Això genera configuracions d'accés molt complexes on errors de configuració són comuns – de fet, un percentatge alt d'incidents al núvol provenen de configuracions errònies de permisos S3, colls massa permissius, etc., en part perquè mantenir regles acurades a mà és difícil en entorns que canvien tan ràpid.

4.2 Entorns IoT: escala i heterogeneïtat

A IoT ens trobem amb milers o milions de dispositius i sensors que necessiten comunicar-se amb serveis o entre ells. Controlar l'accés aquí és crític (per exemple, evitar que un node maliciós enviï ordres a un actuador industrial), però **quin model fem servir?** RBAC? Seria impossible crear rols per cada sensor... Normalment s'opta per **ACLs senzilles o certificats**, però llavors hi ha poca adaptabilitat.

Limitacions de RBAC/ABAC en IoT:

- **Escala massiva:** assignar atributs o rols a cada dispositiu a mà és inviable. Es tendeix a fer-ho per tipus de dispositiu (rol "sensor temperatura"), però tots els sensors temperatura reben mateix permís – si un és compromès, pot fer les mateixes accions que un altre.

- **Context canviant:** potser un sensor només ha de comunicar-se en cert horari o certes condicions. Programar això en polítiques estàtiques seria complex, i sovint no es fa, deixant portes obertes.

- **Capacitat limitada dels dispositius:** molts dispositius IoT tenen poc processament, no poden córrer clients complexos d'autenticació. Sovint es recorre a clau precompartida o certificats simples. Implementar ABAC allà dins seria costós. Això obliga a traslladar el control d'accés a la passarel·la o al servei central, però llavors la comunicació fins allà pot estar exposada si no hi ha xifrat fort.

- **Simplicitat de protocols:** protocols IoT (MQTT, CoAP) no tenen integrat conceptes de rol/atributs; solen confiar en nivells de confiança de la xarxa (un concepte ja antiquat).

Per tant, es necessiten solucions add-on.

Tot plegat fa que en IoT moltes vegades s'opti per seguretat perimetral (restringir xarxa) i funcionalitats mínimes, i no es pugui aplicar un control fi. Això és una limitació real: els models clàssics no encaixen bé i caldria potser control d'accés més **adaptatiu i autònom** (perquè no pots manar un humà que reguli permís a permís per milers de sensors, potser un algoritme ha de decidir en viu si un dispositiu està actuant dins de paràmetres normals i si no, tallar-lo).

4.3 Limitacions davant amenaces modernes i APTs

Més enllà dels entorns tècnics, és important veure com les **tàctiques d'atac modernes** esquiven o exploten els defectes dels models tradicionals:

- **Credencials robades i Moviment lateral:** Els atacants, especialment en **APT (Amenaces Persistents Avançades)**, sovint aconsegueixen credencials legítimes (via phishing o dumping). Un cop tenen un usuari i contrasenya, si aquest usuari té privilegis segons RBAC, entraran sense soroll. Casos com *SolarWinds* o *Colonial Pipeline* van implicar ús de credencials vàlides. Un cop dins, fan moviments laterals cercant comptes amb més permisos. *Limitació tradicional:* RBAC/ABAC estàtic no detectarà res estrany – el sistema veu "usuari X amb rol Y accedint a recurs Z, que té permís". És a dir, no hi ha cap mecanisme intrínsec de seguretat que digui "aquest ús de la credencial és sospitós". Els únics remeis tradicionals són MFA (dificulta robar credencial, però no impossible) i monitoratge manual de logs (SIEM) on humans busquin coses rares. Clarament no és suficient. **Zero Trust** aborda això amb "revalida tot, confia zero" però per fer-ho bé cal analítica.

- **Malinsiders i ús indegut:** Un empleat pot fer servir els seus accessos per extreure informació sensible (casos d'espionatge corporatiu, o simplement curiosos veient dades que no haurien de veure tot i tenir permís general). Models tradicionals confien en que la política inicial hagi estat ben configurada per evitar-ho – però sovint, per facilitats, es dona accés ampli. Per ex., un administrador de BD té accés a totes les dades de clients (perquè és la seva feina), si decideix copiar-les, no hi ha control d'accés que ho impedeixi (ja que tenia permís). Només un sistema que entengués context o detectés volum anòmal de consultes podria saltar. Per tant, la limitació aquí és que **els models clàssics no contemplen la intenció ni la conseqüència, només l'autorització bàsica**.

- **Entorns DevOps i CI/CD:** Els entorns de desenvolupament-continu integració (CI) solen requerir que molts serveis s'accedeixin entre si (repos, entorns de prova, deploy a producció). Sovint, per no entrebancar, es configuren credencials d'ampli abast (API keys amb molts permisos) per agilitat. Això crea **exposicions** enormes si una d'aquestes claus s'escapa (ha passat en multitud de breaches on claus AWS S3 es publiquen per error). Per què passa? Perquè RBAC/ABAC estrictes serien massa feixucs en entorns de pipeline àgil, i es relaxa la seguretat. L'alternativa seria tenir un

sistema adaptatiu que *entengués* l'entorn: ex., un container de build no hauria d'esborrar bases de dades, encara que la seva clau ho permeti; si ho intenta, potser és hackeig, i un AI podria detectar-ho. Sense això, s'opta per la solució subòptima: claus àmplies i confiar que res dolent passi.

4.4 Complexitat administrativa i errors de configuració

Un altre aspecte limitant és la **fal·libilitat humana**: Els models tradicionals, per molt que tinguin principis clars, acaben depenent de configuracions manuals i processos administratius. En entorns moderns on tot canvia de pressa, la probabilitat d'**error de configuració** puja. Molts incidents provenen no de falles del model, sinó d'errors en aplicar-lo:

- Exemples: un bucket S3 exposat públicament (perquè algú va posar permissos *AllUsers* per error), un *firewall rule* mal posat, un usuari no revocat després de deixar la companyia, etc. No és culpa de RBAC o ABAC com a concepte, però sí de la **càrrega que suposa gestionar-los** en entorns grans.

Això esdevé una limitació pràctica: si un model requereix tant esforç que els humans no ho fan bé, el model de fet no compleix la seva fi. Necessitem més **automatització intel·ligent** per compensar la fatiga i error humans en governar milers de entitlements.

4.5 Compliment i regulacions en entorns híbrids

Moltes regulacions (GDPR, HIPAA, PCI DSS) exigeixen controls d'accés estrictes i auditables. Tradicionalment, RBAC es feia servir per demostrar compliment (mostrar qui té accés a dades personals, per exemple). En entorns moderns, amb dades moguent-se a SaaS, PaaS, IoT, etc., fer un **tracking complet** de qui pot accedir a dades regulades és molt complicat. Els models tradicionals no comparteixen informació entre sí fàcilment. Necessites capes de meta-governança. Això és una limitació quan es vol garantir, per exemple, *principi de menor privilegi universal*: potser en sistemes centrals ho tens, però resulta que un servei en el núvol replicava dades i allà per defecte tots els enginyers hi tenien accés – una bretxa de model conceptual.

Resum: Els models tradicionals van ser concebuts en entorns controlats, estàtics (servidors on-prem, usuaris corporatius a la LAN, dispositius confiats). Avui tot és més **obert, distribuït i dinàmic**. Les seves limitacions es fan paleses en:

- Falta de **context en temps real**: no poden respondre adequadament a situacions canviants (risc temporal, anomalies).

- Problemes d'**escalabilitat humana**: massa entitats i permisos per gestionar manualment.

- Silos entre plataformes que trenquen la **coherència**.

- Amenaces que **exploten la confiança implícita** que aquests models donen quan es satisfan les credencials inicials.

5. Anàlisi crítica de les solucions existents i identificació de buits

Després d'haver explorat tant els fonaments com l'estat actual de la IA en control d'accés, és pertinent fer una **anàlisi crítica**: fins a quin punt les solucions actuals (tant les basades en models tradicionals com les primeres integracions d'IA) solucionen els problemes esmentats? Quines són les seves limitacions intrínseques i **buits** pendents d'omplir? Aquesta secció posarà en relleu aquests aspectes, servint de pont cap a la proposta de la secció 7.

5.1 On arriben (i on no) les solucions actuals basades en IA

En la secció 4 hem vist diverses solucions que ja fan servir IA, com Azure AD Identity Protection, Okta Adaptive MFA, etc. Si bé aporten millores evidents, és important notar que moltes actuen com a **capess addicionals** més que no pas substituir el mecanisme central d'autorització:

- **IA com a sistema d'assessorament o alerta:** En molts casos, la IA detecta una anomalia i **genera una alerta** o recomanació, però *no forçosament pren la decisió final*. Per exemple, Azure Identity Protection marca un inici de sessió com arriscat, però és la política condicional (configurada per humans) la que diu "bloqueja'l" o "força MFA". Si aquesta política no s'ha configurat o es configura malament, la intel·ligència no s'aplica. Moltes organitzacions, per por a frenar operacions, configuren respostes "suaus" (p. ex., només alertar en comptes de bloquejar). Això redueix l'efectivitat.

- **IA centrada sobretot en l'autenticació, no tant en l'autorització continuada:** Actualment, el punt principal d'inserció de la IA és durant l'autenticació inicial (login) o en la validació de sessions inicials. Un cop concedit l'accés, hi ha menys sistemes monitorant activament en temps real totes les accions. Algunes solucions de monitoreig de sessió existeixen (p. ex. Microsoft Cloud App Security pot vigilar accions dins d'aplicacions SaaS i aplicar polítiques en viu), però són més aviat DLP/monitoratge que control d'accés formal. **Buit:** Falta un enforcement continu integrat a les aplicacions que, per exemple, talli una sessió si sobtadament es detecta anomalia després del login. Zero Trust ho postula, però la implementació real es complica, i sovint es cau en que l'agent de seguretat només avisa.

- **Limitacions en l'abast de dades:** Molts sistemes d'IA actuals depenen de tenir prou dades d'entrenament i context. Una empresa petita, per exemple, que usi Okta, potser no té un volum de trànsit suficient perquè l'algoritme "aprengui" bé el normal de cada usuari (sobretot si l'usuari és nou). Okta llavors es recolza en dades agregades de tota la xarxa, la qual cosa és bona per atacs coneguts, però pot no captar peculiaritats de l'empresa. És a dir, hi ha un **buit en personalització**: IA global detecta bé atacants externs (IP dolentes, etc.) però potser no tant un insider lentament exfiltrant informació,

cosa que potser un sistema entrenat específicament amb logs interns detectaria.

- **Focus restringit a identitat, deixant de banda altres pilars:** La majoria d'eines IA/IAM es focalitzen en l'usuari i el login, però què passa amb l'**estat del dispositiu** o la **xarxa**? Zero Trust diu que tot importa. Microsoft i Cisco, per exemple, tenen plataformes que integren senyals (Cisco TrustSec, etc.), però en la pràctica molts d'aquests subsistemes no es parlen prou. Un buid detectat: la integració de **Threat Intelligence general amb IAM** està en inicis. Imaginem que hi ha una alerta de que la màquina d'un usuari està infectada (ex. per l'EDR). El sistema hauria de revocar automàticament credencials o intensificar controls per a l'usuari. Algunes solucions de MDM/EDR ho fan (p. ex. si un dispositiu no és complient, Azure AD li pot bloquejar recursos via Conditional Access). Però és limitat als que estan dins l'ecosistema. Si tens multi-núvol i diferents eines, pot fallar la comunicació. Això indica que l'**arquitectura global** de seguretat sovint és fragmentada.

5.2 Limitacions dels models tradicionals quan s'envolten d'IA

Una altra perspectiva: què passa amb DAC/RBAC/ABAC un cop apliquem pegats d'IA al voltant? Segueixen arrossegant certs problemes:

- **RBAC estàtic i rol explosion:** Cap IA màgica evita que les organitzacions encara hagin de definir rols i grups sensats. Moltes empreses continuen patint de *privilege creep* (usuaris amb massa privilegis) perquè netejar això requereix anàlisi i voluntat política. IA pot suggerir rols, però pocs confien cegament i a la fi un gestor ha d'aprovar. Sovint aquests projectes es queden a mig camí per la complexitat organitzativa (no tècnica). **Buit**: manquen eines de *role mining* que siguin fàcils d'usar i integrin directament canvis, i manquen processos per acceptar canvis suggerits per IA (p. ex., si l'algoritme diu que 10 usuaris no haurien de tenir cert permís, hi ha reticència per por que perjudiqui operacions).

- **ABAC complexitat:** Com s'ha esmentat, ABAC és poderós però difícil d'escriure. Encara avui moltes companyies es queden en RBAC per simplicitat. Tot i tenir plataformes que ho suporten, configuren tot via rols i grups, i no usen condicions context (més enllà de bàsics com "requereix MFA fora oficina"). **Buit**: Falta adopció d'ABAC malgrat avantatges, possiblement perquè les eines no el fan prou assequible. L'IA podria ajudar generant polítiques (com GPT escrivint XACML a partir d'una descripció), però això tot just comença. Mentrestant, hi ha un gap entre capacitat i ús real.

- **DAC vs protecció de dades en col·laboració**: En entorns com col·laboració en núvol (Google Drive, O365 SharePoint), la filosofia és DAC (cada usuari comparteix). Com evitem fugues? Es posen DLPs i alertes fora del sistema d'accés. Buit: no hi ha un bon mecanisme de "MAC" adaptat per dades al núvol sense dificultar la feina. IA pot

monitorar qui comparteix què i alertar, però de nou, a posteriori.

5.3 Resistència cultural i de confiança cap a l'IA

Un factor no tècnic però crític: la **confiança**. Implementar control d'accés basat en IA demana que els responsables de seguretat confiïn en les recomanacions o decisions de sistemes automatitzats. Sovint hi ha resistència per:

- **Por a falsos positius:** Bloquejar un usuari legítim en un moment crític pot suposar pèrdues (imagineu un metge no podent accedir a història clínica en emergència per un algoritme paranoic). Això frena l'adopció de mesures completament automàtiques. Les empreses solen configurar per segur: millor deixar passar algun atac que bloquejar operativa bona. Això és un **gran buit**: com assegurar *zero falsos positius* o minimitzar-los? Explicabilitat i ajust fi dels models és clau, però encara falta recorregut. Moltes solucions IA són una mica opaques o almenys complexes d'afinar (p. ex. com ajustar la sensibilitat d'un detector d'anomalies?).

- **Compliance i responsabilitat:** En algunes indústries, es requereix poder explicar per què es va denegar un accés (p. ex. decisions que afecten persones subjectes a drets). Si un model d'IA no pot justificar clarament la decisió, podria infringir regulacions o polítiques internes. Per això, sovint es limita la IA a recomanar i un humà aprova (ex.: systems que suggereixen revocar accessos però un gestor ha d'aprovar).

- **Manca d'estàndards i referències:** No hi ha encara un marc universalment acceptat per "AI-driven access control". Això fa les empreses prudents, ningú vol ser el primer en adoptar totalment. NIST i altres treballen en directrius, però està verd. Per tant, l'**ecosistema normatiu** és un buit – i sense benedicció dels estàndards, moltes empreses no arrisquen a innovar radicalment en com controlen accessos.

5.4 Integració insuficient entre productes de seguretat

Moltes organitzacions han anat comprant solucions ad hoc: un per MFA, un per SIEM, un per DLP, etc. Sovint aquestes eines no estan ben integrades. Per exemple, un SIEM pot detectar activitat anòmala però no té com directament tallar l'accés – cal que algú manualment suspengui l'usuari. Algunes plataformes de **SOAR** (orquestració de resposta) automaitzen accions, però configurar-les és complex i propens a error (que torna al punt de confiança).

Aquest mosaic de solucions significa que tot i que potser hi ha la capacitat tècnica, els fluxos no estan unificats. **Buit identificat:** la manca d'un sistema unificat de **Policy Enforcement dinàmic** que rebi inputs de totes les eines. Zero Trust diria: un Policy Engine que tot ho decideix segons context. Però actualment, tens decisions locals en cada aplicació. Fins i tot si un sistema central diu "aquest usuari risc alt", si tens 50 aplicacions legades, com li comuniques? Sovint no es pot en temps real, i l'usuari segueix loguejat a elles.

5.5 Casuístiques no cobertes i atacs adversarials

Finalment, pensem en el **atacant adaptant-se a la IA**. Un cop sap que hi ha detectors, pot intentar evadir-los: moure's més lent, mesurar conducta per no sobresortir (atacants interns poden *fingir normalitat* fins al moment de l'exfiltració). També poden fer soroll per saturar alertes (falsos positius generats expressament). Els sistemes actuals de IA en accés encara no han enfrontat molts atacs adversarials declarats (tot i que en camps com anti-frau financers, els criminals ja proven d'evadir models). Un **buit potencial**: necessitem que els models d'IA siguin robustos a atacants que entenen el seu funcionament parcialment (p. ex. si saben que 5 intents de login fallit bloquegen, en faran 4 per mantenir-se sota el llindar – i potser IA actual no ho veu com prou anòmal perquè mai hi ha error 5 seguit). Caldrà evolució constant.

5.6 Resum dels buits

Recapitulant els **principals buits** identificats en solucions existents:

- **Adaptació parcial, no completa:** La IA sol aconsellar, però l'enforcement final recau en regles estàtiques configurades; manca tancament automàtic del cicle.

- **Cobertura fragmentada:** Ens enfoquem molt en login i ús de credencials, però menys en autorització contínua i altres contextos (dispositiu, dades).

- **Usabilitat i adopció pendent:** ABAC i altres són poderosos però infrautilitzats per complexitat; IA pot ajudar però encara no hi ha integracions prou amigables.

- **Confiança i explicabilitat:** Resistència a l'automatització per por a errors i dificultat d'explicar decisions als stakeholders o compliance.

- **Integració i orquestració:** Eines aïllades no comparteixen prou informació per fer un control d'accés realment global i dinàmic.

- **Atacs evolutius:** Necessitat de preveure que un adversari pot provar de derrotar els mecanismes d'IA, i tenir plans per això (sistemes multi-capa, etc.).

6. Proposta pròpia: model híbrid ABAC + IA

Arribem al nucli propositiu del treball: es planteja un **model híbrid** que combina l'enfocament de **control d'accés basat en atributs (ABAC)** amb la potència adaptativa de la **intel·ligència artificial**. L'objectiu del model és superar les limitacions identificades, proporcionant un sistema de control d'accés **autoadaptatiu, segur i explicable**. En aquesta secció es detallarà la proposta, incloent-hi la descripció funcional, la seva arquitectura mitjançant un diagrama, el flux de dades o pseudocodi que il·lustra el funcionament i un exemple concret d'aplicació en un context realista (una entitat financera amb sistemes al núvol).

6.1 Descripció funcional del model híbrid

El model proposat es pot entendre com una extensió intel·ligent d'un sistema ABAC clàssic. Recordem que en ABAC tradicional tenim:

- Un **Policy Decision Point (PDP)** que avalua sol·licituds d'accés contra un conjunt de polítiques (regles que involucren atributs).

- Un **Policy Enforcement Point (PEP)** que intercepta les peticions a recursos i demana al PDP una decisió (permitir/denegar, o altres com obligar MFA).

- Fonts d'atributs (bases de dades d'usuari, serveis que proveeixen info de context, etc.) i un **Policy Administration Point (PAP)** on es defineixen les polítiques.

En el nostre model híbrid, a aquest esquema hi afegim un component crucial:

- Un **Motor d'Intel·ligència Artificial** integrat, al qual anomenarem aquí **AI Risk Engine** per simplificar, que col·labora amb el PDP en la presa de decisions.

La idea és que les polítiques ABAC poden incloure condicions que siguin resoltes pel AI Risk Engine. Per exemple, una política podria dir:
"Permet l'accés a l'aplicació X si **rol == "manager"** i **risk_score < 70** i **device_trust_level >= 3**."
Aquí **risk_score** és un valor calculat pel modul d'IA en base a múltiples senyals, i **device_trust_level** podria venir d'un sistema de gestió de dispositius.

Així, el **PDP delega part de l'avaluació a la IA**: quan rep la petició, recull els atributs coneguts (identitat, rol, recursos, etc.), i també crida l'AI Risk Engine per obtenir atributs dinàmics com el **risk_score** actual de la sessió, o una classificació del comportament. Un cop els té tots, aplica la política completa.

Què fa exactament l'**AI Risk Engine**? Les seves funcions principals:

- **Anàlisi de comportament de l'usuari i entorn en temps real:** A partir de logs d'accés recents, historials de comportament de l'usuari (hores d'activitat típiques, patron de recursos accedits, velocitat de peticions, etc.), estat del dispositiu client (extret via integració amb EDR/MDM), estat de la xarxa (p. ex., si la IP d'origen té mala reputació), i possiblement informació de transaccions específiques (ex: import d'una transacció financera que s'està intentant realitzar), calcula una mètrica de risc i/o categoritza la sol·licitud. Es poden fer servir models de ML com detallat abans: combinació de classificadors i detecció d'anomalies.

- **Feedback contínu: actualització de perfils:** El motor d'IA no és estàtic; aprèn amb el temps. Cada validació d'accés (sigui permesa o denegada) s'utilitza per reajustar el perfil de normalitat o detectar nous patrons. Això permet que si l'organització introdueix una nova eina i de sobte tots comencen a accedir-hi (canvi de patró global legítim), el sistema s'adapti per no marcar-ho erròniament.

- **Explicació del risc:** Per fer el sistema utilitzable, l'AI Risk Engine també genera **indicadors explicatius** junt al risc. Exemple: **risk_score = 85, factors: new_geolocation + unfamiliar_device**. Aquestes etiquetes poden ser usades després per logs o per mostrar als administradors per què certa decisió va ser presa.

Un cop el PDP té la resposta de l'AI (p. ex. risc alt), aplica la política:

- Si risc alt i política diu denegar, torna DENY.

- Si risc mitjà i política diu requerir MFA addicional, li indica al PEP que challenge l'usuari amb MFA.

- Si risc baix i resta de condicions OK, permet.

Característica clau: autoadaptació de polítiques: Més enllà de la decisió individual, el model inclou un mecanisme perquè l'IA **suggereixi actualitzacions de la política ABAC**. És a dir, hi ha un bucle d'aprenentatge:

- El AI Risk Engine pot detectar regles emergents. Ex: veu que repetidament està marcant risc alt per a cert tipus d'accés i aquests accessos acaben sent denegats per polítiques o per intervenció manual. Llavors podria recomanar formalitzar això en una política explícita (reduint dependència del ML en aquest cas). Potser descobreix que "tots els accessos de servei des de IP estrangera estan sent bloquejats; hauria de ser una condició de política".

- A l'inrevés, si constantment marca risc alt però els administradors veuen que són falsos positius i acaben permetent accés, el sistema podria aprendre a ajustar el model (baixar

pes d'aquella condició) **o** suggerir que es relaxi una política massa estricta si s'escau.

Aquest component d'aprenentatge de polítiques és semiautomàtic: probablement les suggerències passen per revisió humana (al menys en entorns molt regulats). Es podria fer via un *dashboard*: "El sistema suggereix afegir condició X a la política Y per reduir incidències", i un analista pot acceptar-ho amb un clic.

Visió 360º de context: El model proposat centralitza en gran mesura la decisió en un punt (PDP + AI Engine). Això és important per entorns híbrids: significa que independentment de si la sol·licitud ve d'un recurs on-prem, d'una API al núvol, o d'una aplicació SaaS, totes passen pel mateix cervell que aplica coherència de polítiques i intel·ligència. En la implementació, això pot ser un servei d'autorització centralitzat (per exemple, un servei web que actua de PDP universal via APIs, al qual apunten tots els PEPs desplegats en proxies, gateways, etc.). Evidentment cal integrar-se bé: potser usar proxies inversos per aplicacions web, adaptadors PAM per a servidors legats, etc., però conceptualment, **tots consulten al mateix cervell**.

Resposta adaptativa i actuació en temps real: A diferència dels sistemes tradicionals on un cop donat accés poca cosa dinàmica passa, aquí es pot incloure:

- **Reautenticació o elevació de nivell durant la sessió:** Si el context canvia, el PDP/AI pot reavaluar. Per exemple, suposem que l'usuari inicia sessió (risc baix) i se li permet fer operacions bàsiques. De sobte, comença a demanar gran volum de dades confidencials (això es detecta com anòmal). El PEP pot demanar al PDP de reavaluar (a través d'un agent monitor). El PDP veu que ara risc = alt per canvi de comportament i llança una acció: potser força l'usuari a reintroduir credencials o directament li revoca la sessió ("logout forçat per seguretat"). Aquest tipus de flux s'alinea amb Zero Trust – *continual monitoring and enforcement* – i l'AI és qui dóna la capacitat de decidir quan cal saltar.

- **Aïllament progressiu:** El model també podria suportar respostes gradacionals: no tot és binari permet/denega. Per exemple, si es detecta risc moderat, pot canviar la *postura d'accés*: "et deixem seguir però amb privilegis reduïts fins que es clarifiqui". Un cas podria ser en un entorn cloud: si un compte de servei es comporta estrany, encara permetem algunes operacions de lectura però bloquegem accions crítiques fins verificar. Això equival a un *circuit breaker* que l'AI pot activar.

Amb tot això, la descripció funcional es resumeix així: **un sistema d'autorització central que fa complir polítiques declaratives (ABAC) enriquides amb valoracions d'IA en temps real**, retroalimentat per l'aprenentatge continu i capaç de modificar la seva conducta sobre la marxa per mantenir un equilibri òptim entre seguretat i usabilitat.

6.2 Diagrama d'arquitectura

Figura 6.1: Diagrama d'arquitectura del model híbrid ABAC + IA.

Figura 6.1: Arquitectura proposada del sistema de control d'accés dinàmic ABAC+IA, mostrant el flux entre l'usuari (subjecte), el PEP (punt d'aplicació, per exemple una passarel·la de seguretat), el PDP central (amb motor de polítiques) i el motor d'IA que analitza context i comportament en temps real per produir un risc que influeix en la decisió. Es destaca també el bucle de retroalimentació on els logs alimenten l'IA per reentrenament i el mòdul d'administració de polítiques rep suggeriments de millora.

En el diagrama es poden observar els següents components i fluxos numerats:

1. **Sol·licitud d'accés:** Un subjecte (usuari o servei) intenta accedir a un recurs protegit (p. ex. una aplicació, una API, una base de dades). Aquesta petició passa pel **Policy Enforcement Point (PEP)** – podria ser un proxy, un agent al servidor, un plugin a l'aplicació, etc. El PEP captura la petició i la remet al PDP central per a decisió (això correspon al flux 2).

2. **Consulta al PDP:** El **Policy Decision Point**, implementat com un servei d'autorització central, rep la consulta amb els detalls: identitat del subjecte, recurs sol·licitat, acció, etc. Recupera els atributs bàsics del subjecte (rols, grups, atributs de perfil) d'un magatzem d'identitats (base LDAP/AD o base de dades d'usuari) i atributs de l'objecte (metadades del recurs) i entorn (IP, hora, ubicació).

3. **Crida al AI Risk Engine:** Abans d'avaluar les polítiques, el PDP envia aquests contextos al **Motor d'IA de Risc i Comportament** (flux 3). Li proporciona: identificador d'usuari, atributs coneguts, historials de logs disponibles, etc., i espera una resposta.

4. **Anàlisi i resposta de l'AI:** El motor d'IA processa la informació. Combinant el que sap de l'usuari (patrons previs), la situació actual (per ex., IP no habitual, device posture = no complient) i altres entrades, retorna al PDP un conjunt d'**atributs calculats**. Els principals:

    - **risk_score = X** (numèric 0-100 o categòric Baix/Mig/Alt).

    - Possiblement **anomaly = true/false** si va detectar una anomalia concreta, i una classificació de tipus d'anomalia (p. ex. **anomaly_type = "impossible_travel"**).

    - Qualsevol altre atribut derivat que les polítiques facin servir, com **user_trust_level** actualitzat, o **session_risk_level.**

5. **Avaluació de Polítiques ABAC/PBAC:** Ara el PDP té tots els atributs (estàtics + dinàmics de l'AI). Llavors carrega les regles de polítiques rellevants (per exemple, troba la política que s'aplica a "aplicació X") i avalua les condicions. Això inclou expressions possiblement com (rol == "manager" AND risk_score < 70...). El motor de regles determina la decisió d'accés i també pot adjuntar obligacions (p. ex., "permitir però

notificar" o "permetre però fer logging extra").

6. **Resposta al PEP:** El PDP envia la decisió al PEP (flux 5 al diagrama). Això pot ser: Permit, Deny, MFA required, etc., segons un esquema de decisions (per exemple, XACML suporta *permit*, *deny*, *permit with obligations*). Al nostre model, incloem decisions com "permit with step-up" (requereix MFA abans de finalitzar) o "permit with reduced scope" (un concepte on potser no donem tots els privilegis, però això és més difícil d'implantar generalment tret que l'aplicació suporti nivells d'accés dinàmics).

7. **Enforcement i resposta a l'usuari:** El PEP aplica la decisió. Si és *deny*, retorna un error d'autorització a l'usuari. Si és *permit*, li permet passar i accedir al recurs. Si hi ha obligació de MFA, el PEP gestionarà un cicle extra demanant MFA (ell mateix o trucant un servei d'autenticació) abans de finalitzar l'accés. Després d'això, l'usuari obté accés.

8. **Monitoratge continu (opcional al moment inicial):** Un cop l'usuari està dins, el PEP pot seguir monitorant la sessió. Si detecta un esdeveniment significatiu (p. ex., volum de dades descarregades molt gran, un canvi d'adreça IP a mig sessió via VPN, etc.), pot tornar a trigger el flux de decisió. L'AI Risk Engine també pot estar consumint logs en streaming i si ell detecta que la sessió d'un usuari es torna de risc (sense ni que el PEP ho demani), podria notificar el PDP/PEP. Al diagrama, això s'il·lustra amb una fletxa de retroalimentació 8.1 on els logs arriben a l'AI i 8.2 on pot avisar el PDP.

9. **Feedback i aprenentatge:** Totes les decisions i dades associades es loguegen a un **magatzem de logs segur**. El Motor d'IA les incorpora per reentrenar models (flux 8.1). Alhora, hi ha un **Policy Administration Point (PAP)** on un administrador defineix les polítiques. Aquest PAP rep del AI Engine suggeriments (flux 9) com mencionat. Un administrador pot veure recomanacions com: "S'ha detectat que cap accés fora de la UE hauria de ser permès per aplicació Y; actualitza política per reflectir-ho.". Si accepta, es modifica la política al repositori i es propaga al PDP.

6.3 Flux de dades i pseudocodi del procés

Per il·lustrar el funcionament, presentem un pseudocodi simplificat per a l'algoritme que executaria el PDP en conjunció amb l'AI engine, aquest pseudocodi degut al poc nivell que tenim de programació es generat mitjançant l'ajuda de IA ***(chatgpt 4.0)***:

```
// Pseudocodi per maneig d'una petició d'accés en el model ABAC+IA

function authorizeRequest(user, resource, action, context):
    # Obtenir atributs estàtics
    user_attrs = IdentityStore.getAttributes(user)        # e.g. rol, departament, clearance
    resource_attrs = ResourceDB.getAttributes(resource)    # e.g. owner, sensitivity_level, tags
    env_attrs = context                                    # e.g. ip_address, time_of_day, location
```

```
    # Combinar atributs bàsics
    attrs = user_attrs ∪ resource_attrs ∪ env_attrs

    # Consultar l'AI Risk Engine per atributs dinàmics
    ai_input = {
        user: user,
        resource: resource,
        action: action,
        context: env_attrs,
        history_window: Last30minLogs[user]
    }
    ai_output = AIRiskEngine.assess(ai_input)
    # ai_output pot retornar, per exemple:
    # ai_output.risk_score, ai_output.anomaly_flags, ai_output.confidence

    attrs = attrs ∪ ai_output    # afegim risk_score i altres atributs calculats

    # Avaluar polítiques ABAC
    decision = "deny"   # valor per defecte si cap política s'aplica
    applicable_policies = PolicyStore.retrievePolicies(resource, action)
    obligation = None

    for policy in applicable_policies:
        if evaluateConditions(policy.condition, attrs):
            decision = policy.effect  # e.g. "permit" o "deny"
            obligation = policy.obligation  # e.g. "MFA" o "NOTIFY"
            break   # assumim primera política coincident (poden haver-hi estratègies merge)

    # En cas que la política demani MFA i l'usuari no hagi fet MFA en aquesta sessió:
    if decision == "permit" and obligation == "MFA":
        if not context.mfa_performed:
            triggerMFA(user)
            context.mfa_performed = true
                return authorizeRequest(user, resource, action, context)  # reavaluem amb context
actualitzat (MFA fet)
        else:
            # MFA ja realitzat, procedir a permetre
            decision = "permit"
            obligation = None

    # Log de la decisió
    EventLog.write({
        user: user, resource: resource, action: action,
```

```
    decision: decision, obligation: obligation,
    risk_score: ai_output.risk_score, anomalies: ai_output.anomaly_flags
  })

  return decision, obligation
```

En aquest pseudocodi podem veure:

- Obtenció d'atributs de subjecte, objecte i entorn.

- Crida al **AIRiskEngine.assess** amb inputs (usuari, recurs, acció, context actual i potser històric curt termini). L'engine retorna un **risk_score** i possiblement banderes d'anomalia.

- Els atributs retornats s'afegeixen al conjunt d'atributs.

- Després s'avaluen les polítiques aplicables. La condició de cada política pot incloure expressions com **risk_score < 70, anomaly_login == false**, etc., les quals **evaluateConditions** resol fàcilment perquè tots els valors estan a **attrs**.

- Si la política té efecte *permit* però obliga MFA, es comprova si ja s'ha fet (potser es pot saber per context de sessió). Si no s'ha fet, es llança MFA (**triggerMFA**) i després de completar es torna a cridar a authorizeRequest amb **mfa_performed** marcat per reavaluar (aquest cop possiblement **attrs** inclourà un atribut **mfa = true** que la política podria requerir). Això garanteix que MFA s'integra en el cicle de decisió i registre.

- Finalment, es registra l'esdeveniment complet amb tota la informació, incloent el risc calculat. Aquests logs alimenten l'entrenament futur.

Detalls addicionals:

- L'algoritme mostra un reenviament recursiu per MFA. En implementació real es podria manejar de forma no recursiva però la idea és fer re-check després de MFA.

- **applicable_policies** es basa en algun mecanisme (ex. cada política té camps target de quin recurs o tipus d'acció s'aplica, similar a XACML target matching).

- La combinació de polítiques aquí per simplicitat para quan troba una (first match). En ABAC real potser n'hi ha vàries i s'han de combinar (p. ex. deny pot prevaldre sobre permit). Són detalls del motor de polítiques.

Aquest flux deixa clar com la IA entra en bucle de decisió però **no reemplaça la política** sinó que la complementa amb atributs nous. Les polítiques segueixen essent definides per l'organització (amb l'ajuda de la IA). Això aporta **control i explicabilitat**: un auditor pot veure la política (p. ex. "denegar si risc alt i no és xarxa corporativa") i entendre-la, i també verificar els valors que l'IA va donar (p. ex. risc=90, ip_externa=true) per justificar la denegació. Així es compleix requisits de traçabilitat.

6.4 Exemple d'aplicació: entitat financera en entorn híbrid

Per fer tangible el model, considerem una **entitat financera** (un banc) que té infraestructura híbrida:

- Utilitzen aplicacions al núvol per a gestió de clients (CRM al Salesforce, per exemple), i tenen aplicacions financeres crítiques on-prem (sistemes core bancari).

- Tenen també desplegaments en un núvol públic (Azure) on corren algunes funcions serverless per analítica.

- Disposen d'empleats a oficines i d'altres en teletreball, i col·laboradors externs (auditors, etc.) amb accés limitat.

- Han adoptat principis Zero Trust i volen garantir que l'accés a dades sensibles (com dades de comptes bancaris, transferències) sigui estrictament controlat.

Situació abans del model: Abans, el banc tenia un RBAC clàssic:

- Rols com "Gestor de comptes", "Director d'oficina", "Analista de riscos", etc., definien qui podia veure o modificar certs informes.

- Tothom passava per VPN per accedir a recursos interns, i via SSO a apps cloud.

- Tenien MFA en l'accés VPN i SSO, però era fix (sempre que no fossis a la xarxa interna).

- Les polítiques eren estàtiques: per exemple, un gestor de comptes (rol) podia veure dades de clients del seu segment sempre que estigués logat. No hi havia diferenciació per context, només es pressuposava "rol adequat => accés".

- Patrons abusiu potencial: si robaven credencials d'un gestor, podies entrar via VPN (MFA potser saltat via phishing push) i extreure milers de dades sense alarmes immediates (només logs per a revisar tardana).

Amb el model ABAC+IA implementat:

- El banc defineix polítiques ABAC avançades. Per exemple:
    - *Política 1:* “Permetre a un **Gestor de comptes** accedir a les dades detallades d’un client **només si**: el client està assignat a la seva cartera **i** la petició prové d’un entorn de confiança **i** el risc de sessió és baix. En cas de risc mitjà, requerir una confirmació addicional (MFA) per obrir les dades; en cas de risc alt, denegar.”
    - *Política 2:* “Un **Analista de riscos** pot generar informes agregats (no PII) sense restricció horària, però per accedir a dades personals raw de clients requereix estar en horari laboral i risc baix.”
    - *Política 3:* “Col·laboradors externs (amb atribut usuari.extern=true) només accediran a l’aplicació X des de dispositius corporatius gestionats i amb risc molt baix, mai a dades de comptes reals (només entorns de test).”
    - *Política 4:* “Si un usuari realitza més de N consultes de saldo de comptes en menys de 5 minuts (patró anòmal comparat amb operativa normal), elevar automàticament el seu risc de sessió a alt i notificar seguretat; bloquejar transaccions fins a revisió.”
- Ara imaginem un escenari: **Anna**, Gestora de Comptes, treballa des de casa avui. Intenta accedir al perfil d’un client VIP per revisar les seves inversions.
    - Anna obre l’app CRM via el portal securitzat. El PEP (un proxy zero trust) intercepta la petició.
    - PDP rep petició “Anna vol GET dades client #123”.
    - Atributs: rol=Gestor, client_solicitat=123, cartera_Anna inclou 123? (aquest es un atribut calculat: el sistema sap quins clients té Anna assignats, diguem que sí).
    - Entorn: IP = adreça de casa (no a l’oficina), DeviceID = portàtil corporatiu (marcat com gestionat amb antivirus OK). Hora = 10:00 del matí.
    - AI Risk Engine: avalua context:
        - Anna ha iniciat sessió amb MFA fa 30 min.
        - IP de casa seva s’ha vist abans ocasionalment, però avui es connecta des d’una altra ciutat (suposem que està de viatge). Això és un factor una mica anòmal però no extrem; el model ho marca com a risc moderat.

        - Dispositiu: complient, cap alerta, bon senyal.

        - Conducta: en l'última hora Anna ha consultat 2 clients, això és normal.

- Conclusió AI: **risk_score = 40** (baix-mitjà), factors = {nova_localitzacio}.

    - PDP avalua Polítiques:
      Política 1: rol=Gestor (OK), client assignat (OK), entorn de confiança? (Comprova, definim entorn de confiança = xarxa corporativa o VPN corporativa; aquí està fora VPN, així que NO es compleix entorn de confiança). Això ja falla condició. Potser hi ha part: "si no entorn de confiança, requerir risc extra baix + MFA". Continuem:
      Potser Polítiques estan estructurades per casos:
      Trobem potser *Política 1b:* "Gestor fora oficina => ha de tenir MFA recent i risc <50".
      Anna té MFA (sí, fa 30 min) i risc 40 (<50), tot Ok. Llavors efecte: Permit amb obligació "NOTIFY: external access".

    - Per tant, decisió: Permetre però s'envia una notificació al supervisor que Anna va accedir fora oficina (a efectes d'auditoria).

    - Anna obté l'accés a les dades del client i ni s'adona de la mecànica (ja havia fet MFA).

    - Tot queda registrat: risc 40, ubicació X diferent, etc.

- Imaginem ara un cas dolent:

    - **Brian**, un atacant que ha obtingut credencials d'un gestor similar, intenta fer extracció massiva:

    - Es connecta amb credencials de **Carles** (un altre gestor) des d'un altre país i dispositiu no corporatiu.

    - PEP intercepta, PDP consulta:

        - Device_trust = 0 (no device check possible o no conegut).

        - IP geolocalització = país estranger on Carles mai ha estat.

        - AI Risk Engine: veu inici de sessió estrany (impossible travel, ja que Carles ahir estava a l'oficina a Barcelona i avui algú intenta des de Rússia, i sense VPN corporativa).

        - risk_score = 95, anomaly_flags = {impossible_travel, unmanaged_device}.

    - Política aplicable: “Gestor fora entorn corporatiu i risc >=80 => Deny directament”.

    - Decisió: Denegar. Potser fins i tot bloqueja compte i envia alerta immediata a seguretat (això pot ser part de obligacions: **obligation = "LOCK_ACCOUNT_AND_ALERT"**).

    - L’atacant no entra. El SOC rep avís i pot investigar (descobreixen credencial robada, fan reset).

- Un altre escenari:

    - **Diana**, empleada interna, ha iniciat sessió al sistema de core bancari (on prem).

    - Normalment consulta uns 10 comptes al dia. Avui, un malware al PC la controla i comença a enumerar comptes (fa 100 consultes en 5 minuts).

    - AI Risk Engine (o un mòdul seu especialitzat en ràtios) detecta desviació forta respecte patró de Diana i llindars definits (Política 4: >N consultes en <5min).

    - Eleva `risk_score` de la sessió a 90 i marca anomaly = "data_exfil_pattern".

    - Potser fins i tot el motor pot enviar directament un senyal al PEP del core (si està monitoritzant) via un canal push.

    - PDP revalua o PEP aplica directament una regla de emergència: bloqueja les següents consultes de Diana, la desloga de la aplicació i marca el seu compte temporalment sospès.

    - Els administradors de seguretat reben alerta crítica. S’evita possiblement una fuga massiva.

Beneficis observats en l’exemple:

- Granularitat: es diferencia accés en oficina vs fora, clients assignats vs no, accés normal vs massiu.

- Adaptació: es va permetre a Anna amb una condició (ja tenia MFA, es confia però es notifica). En canvi, l’atacant va ser bloquejat per risc alt. Diana va ser parada a mig

abús.

- Millora de usabilitat: Carles, si hagués volgut treballar viatjant, potser li hagués saltat MFA addicional o fins i tot denegat si no avisava – però això és per seguretat. Per a la majoria, si fan servir dispositiu corporatiu i VPN, la feina flueix sense traves (risc roman baix).

- Explicabilitat i compliance: totes les decisions van quedar justificades amb regles: "denegat perquè risc alt per impossible travel", "permet però logueja perquè dispositiu no corporatiu tot i risc baix". En auditories, poden presentar aquestes regles i registres com a evidència de control robust (p. ex. PCI DSS demana monitorar accessos administradors – aquí es fa i es bloquegen si sospitosos).

- Reducció de superfície d'error humà: Moltes d'aquestes reaccions són automàtiques, no cal que un administrador vegi un log i manualment desconnecti un usuari (que potser ho faria tard). El sistema actua en segons.

7. Consideracions ètiques, tècniques i legals

En implementar un sistema de control d'accés autoadaptatiu basat en IA com el proposat, cal atendre un seguit de **consideracions transversals** per assegurar que la solució sigui no només efectiva, sinó també responsable, ètica i conforme a la normativa vigent. En aquesta secció abordem tres eixos de consideracions:

- **Ètiques:** relacionades amb aspectes com la privacitat, la transparència, la no discriminació i la responsabilitat en l'ús d'IA en decisions que afecten persones.

- **Tècniques:** reptes d'implementació, complexitat, fiabilitat i seguretat del propi sistema (resiliència davant atacs, manteniment, etc.).

- **Legals i de compliment normatiu:** adequació a regulacions de protecció de dades (p. ex. GDPR), a estàndards de seguretat (ISO, NIST) i a requisits sectorials (p. ex. en finances, salut, etc.).

7.1 Consideracions ètiques i de privacitat

Privacitat de dades i mínim accés a informació personal: Un sistema d'IA per control d'accés pot necessitar analitzar gran quantitat de dades dels usuaris: horaris de connexió, adreces IP (que poden indicar localització), comportaments d'ús d'aplicacions, etc. Això planteja un repte de privacitat, ja que s'està recopilant i processant informació potencialment sensible per inferir comportaments. És essencial aplicar **principis de minimització i anonimització**: recollir només les dades necessàries per a la seguretat i, en la mesura possible, tractar-les de forma agregada o pseudonimitzada. Per exemple, si s'avalua el "comportament típic" d'un usuari, potser no cal emmagatzemar cada URL que visita sinó mètriques agregades (nombre d'accessos per hora, variància, etc.). També convé limitar la retenció temporal de dades de comportament – un cop entrenades les models, conservar només el necessari per recalibrar, esborrant registres antics per no crear un històric massa invasiu.

Transparència i explicabilitat en decisions d'IA: Quan un sistema automatitzat denega a algú l'accés a un recurs o li imposa traves (com requerir MFA addicional), s'està prenent una decisió que afecta directament l'individu. Des d'un punt de vista ètic (i recolzat legalment en algunes jurisdiccions, com veurem) l'usuari té dret a saber almenys en termes generals per què se li ha denegat l'accés. Un sistema opac on "la màquina ha dit no" pot generar indefensió i desconfiança. Per això cal incorporar **mecanismes d'explicació** de la decisió: per exemple, si un usuari és bloquejat, mostrar-li un missatge indicant "Accés bloquejat per seguretat: inici de sessió des d'ubicació desconeguda" en lloc de només "Accés denegat". Internament, l'empresa també ha de poder auditar i entendre les decisions: tenir logs que indiquin "usuari bloquejat perquè risc alt degut a IP estranya i dispositiu no registrat". Aquesta transparència és

fonamental per mantenir la responsabilitat i corregir errors (p. ex., si un usuari creu que se l'ha discriminat, es podrà investigar la causa precissa).

Absència de biaix i no discriminació: Els algoritmes d'IA corren el risc d'heretar biaixos dels dades amb què s'entrenen. En context de control d'accés, això podria traduir-se en tractament desigual a usuaris d'una certa regió, o amb cert perfil, si històricament s'han produït més incidències de seguretat associades a ells. Per exemple, podria passar que el sistema marqués amb més freqüència com a alt risc un inici de sessió des d'un país estranger perquè en el passat molts atacs provenien d'allí. Però hi pot haver usuaris legítims viatjant a aquell país que es vegin perjudicats. Cal vigilar que les mètriques de risc no incorporin factors com la nacionalitat, raça, gènere, etc., de forma que penalitzin sistemàticament un grup (encara que sigui indirectament). Un exemple: si dones empleats d'un departament (p. ex. seguretat) es comporten diferent i això confon l'algoritme, podria incitar-lo a marcar-los, creant un biaix per departament. Solució: entrenar i validar amb conjunts diversos de dades, i fer tests de **"biaix algorítmic"** periòdics. Incloure humans en la revisió de falsos positius i negative.

Consentiment i percepció de vigilància: Monitorar el comportament dels usuaris pot acostar-se a una línia tènue on els empleats es sentin constantment vigilats. Tot i que l'objectiu és seguretat, èticament s'ha de considerar la *dignitat i confiança*. És important que l'organització informi clarament els usuaris (empleats) que s'implementa aquest sistema, explicant-ne l'objectiu (protegir-los a ells i a les dades) i les mesures de privacitat adoptades. Sovint, les polítiques internes de l'empresa han de contemplar i fer signar als empleats que certes activitats poden ser monitoritzades. L'**ús legítim** de la IA ha d'estar alineat amb la finalitat declarada: no es poden fer servir les dades recollides per al control d'accés per altres propòsits (ex: avaluació de rendiment laboral, tret que això estigués explícit i consentit, la qual cosa seria controvertida). Mantenir un estricte *firewall* entre seguretat i altres usos reforça l'ètica.

Responsabilitat i supervisió humana: Malgrat l'alt grau d'automatització, es recomana mantenir una **supervisió humana** dels casos crítics. Això vol dir que davant decisions especialment sensibles (p. ex. bloquejar un compte crític) pot haver-hi un procediment de revisió urgent per part d'un administrador. Èticament, no es vol que la IA tingui l'última paraula en absolut sobre decisions que podrien fins i tot afectar la carrera d'una persona (imagineu un fals positiu repetit bloquejant un directiu en moments clau). Com diu la màxima, la IA ha d'**augmentar** la intel·ligència humana, no substituir-la completament en tot. Mantenir humans "in the loop" o "over the loop" per vigilar i fer *override* quan cal és una recomanació fonamental.

7.2 Consideracions tècniques i de seguretat del sistema

Complexitat i fiabilitat: Un sistema híbrid ABAC+IA és, per definició, força complex: integra motors de polítiques, models d'ML, diverses fonts de dades… Això introdueix superfícies de fallada. Cal dissenyar l'arquitectura amb **alta disponibilitat** en ment: rèpliques del PDP i AI engine, fallback modes (p. ex. si l'AI engine no respon, el PDP ha de tindre una política per defecte – potser optant per la via segura: "assume risk alt" o, en entorns menys crítics, "assumeix risc normal però logueja"). També s'ha d'assegurar la **eficiència**: la latència afegida

per la consulta a l'AI ha d'estar controlada. Si el sistema tarda massa a autoritzar, pot impactar experiència d'usuari. Tècniques com caching de decisions de curt termini (per no recalcular risc a cada clic, si no cada X minuts o en certs esdeveniments) poden ajudar, però s'ha de vigilar no perjudicar seguretat (potser es cau en "cachear" decisions que haurien de reevaluar-se).

Resiliència contra atacs adversarials: El sistema mateix pot ser objectiu d'atacants sofisticats. Per exemple, un agent maliciós podria intentar "enganyar" l'AI engine enviant comportaments calculats per semblar normals (adversarial examples). O pitjor, si coneixen en part com es calcula el risc, podrien intentar manipular atributs: imaginant que saben que massa peticions seguides alçen alarmes, poden just quedar-se sota el llindar (atac per evasió). S'ha de dur a terme **testing de seguretat específic per IA**: examinar si petits canvis en l'input (ex. spoofar adreces MAC, o injectar soroll als logs) podrien fer que l'algoritme calculi risc incorrecte. La IA aplicada a ciberseguretat també pot patir atacs de poisoning: algú podria infiltrar-se generant molts events falsos perquè l'algoritme "aprengui" malament (ex.: saturar el sistema amb accessos estranys que al final no són atacs, per tal que esdevingui tolerant amb aquests). Contra això cal filtrar dades d'entrenament, monitorar qualitativament si hi ha canvis dràstics en el model i, si és possible, fer que les actualitzacions de model no siguin completament automàtiques sense certa validació. És prudent mantenir versions anteriors del model i si una nova versió es comporta de manera sospitosa (p. ex., sobtadament classificacions de risc baix per coses que abans eren altes), poder fer rollback.

Seguretat de la infraestructura IAM+AI: Més enllà dels atacs a l'algoritme, hi ha els atacs clàssics: el **PDP** i l'**AI engine** esdevenen components crítics de seguretat; si un atacant els comprometís, podria autoritzar-se tot (per ex., manipulant la base de polítiques o fent que l'AI posi risc 0 a tot). Per tant, cal aplicar controls forts: executar aquests components en servidors segurs, aïllats, amb mínim accés. Autenticar cada interacció (PEP-PDP han de fer mútua autenticació, per evitar impersonació). Emmagatzemar logs de manera íntegra (signats, per evitar que un intrús esborri el rastre). El codi del motor d'IA s'ha d'escriure seguint estàndards, i preferentment no ha de tenir acces a secrets (no hauria de poder canviar polítiques directament, només recomanar). Tambe, tests de penetració regulars i auditories al codi són recomanables, ja que és codi complex i susceptible a vulnerabilitats com qualsevol altre (buffer overflows, injections si tracta data, etc.).

Escalabilitat i manteniment: Tècnicament, cal pensar en l'escalabilitat: en una organització gran, pot haver-hi milers de decisions per segon. L'AI engine ha de suportar-ho – possiblement amb computació paral·lela o pre-càlculs. Per exemple, podria recalcular el risc base d'un usuari cada minut i guardar-lo, en comptes de fer un heavy compute a cada sol·licitud (depèn de la frescor requerida). El sistema ha de poder *escalar horitzontalment*: instàncies múltiples del PDP (amb repositori de polítiques coherent entre elles, potser centralitzat o replicat), i instàncies del motor d'AI amb un balanç de càrrega intel·ligent (potser assignant usuaris determinats a instàncies en particular per mantenir caches locals). També manteniment: models de ML deprecien amb temps (concept drift). S'ha de planificar tasques de retuning: potser cada mes, un equip de seguretat revisa la performance (falsos positius/negatius) del model, i ajusta paràmetres o reentrena amb noves dades. Fins i tot es pot fer *red-team* exercicis interns:

persones provant de burlar el sistema per veure com respon i millorar-lo. Tot això requereix recursos especialitzats (data scientists, enginyers) contínuament, que cal considerar.

7.3 Consideracions legals i de compliment normatiu

Protecció de dades personals (GDPR, etc.): Si el sistema processa dades personals d'usuaris (empleats o clients), entra en l'àmbit de regulacions com la **GDPR europea**. Aquesta exigeix diversos aspectes:

- **Base legal per al processament:** l'empresa ha de tenir un motiu legítim per recollir i analitzar aquestes dades. En el cas d'empleats, la seguretat de la xarxa podria entrar com interès legítim de l'empresa, però possiblement caldria fer una avaluació d'impacte de protecció de dades (DPIA) perquè es tracta d'un processament novedós i potencialment intrusiu.

- **Drets dels interessats:** la GDPR inclou el dret a rebre explicacions sobre decisions automàtiques significatives que els afectin (article 22). El nostre sistema en certa manera pren decisions automàtiques (denegar accés) que podrien "afectar significativament" l'empleat si li impedeixen fer la seva feina. Per tant, la implementació ha de complir que es proporcionin explicacions (com hem comentat abans) i possiblement oferir la possibilitat de *revisió humana* de decisions contestades (per ex., un empleat pot reclamar si creu que se'l bloqueja injustament).

- **Minimització i retenció limitada:** correlat amb la part ètica, legalment s'ha de limitar molt la quantitat de dades recollides i el temps que es guarden. Per exemple, log de localitzacions: potser es poden guardar només durant X mesos i després anonimitzar. I s'ha de documentar aquestes polítiques de retenció i informar els usuaris.

Compliance amb estàndards de seguretat (NIST, ISO): L'ús d'un sistema adaptatiu no eximeix de seguir controls clàssics. De fet, frameworks com **NIST 800-53 Rev.5** o ISO 27001 no diuen "fes servir IA", però sí demanen coses com: control d'accés basat en rol amb privilegi mínim (AC-2, AC-6), monitoratge continu (SI-4) i resposta a incidents. El nostre sistema pot ajudar a complir-ho millor, però cal assegurar que:

- Les polítiques definides encara respecten *least privilege* (no donar mai més del necessari per defecte).

- Hi ha separació de funcions clara: qui pot canviar les polítiques? (hauria d'estar restringit a Security Admins). Qui pot canviar els models d'AI? (potser Data Scientists, però amb validació de Security).

- Auditories: Tenir tot loguejat i disponible per auditors interns/externs. Moltes regulacions (PCI DSS, SOX, etc.) requereixen proves de que els accessos a dades sensibles estan controlats i registrats. El sistema automàtic ha de proporcionar logs comprensibles a

auditors. Podria incloure funcionalitats de reporting: ex., generar un informe mensual de tots els accessos elevats bloquejats, o de com s'estan aplicant polítiques de segregació.

Sectors regulats específics:

- **Finances (PSD2, SOX)**: Normatives com PSD2 imposen MFA per certs accessos de clients; en el cas intern d'un banc, no tant, però la governança d'usuaris privilegiats (S-OX) és important. El sistema pot facilitar demostrar qui accedeix a què i sota quin context. Caldrà possiblement configurar-lo per *no impedir* aquelles accions requerides legalment – ex.: PSD2 requereix re-autenticació cada X temps per certs accessos a dades de clients; el nostre sistema, adaptatiu, potser no ho faria perquè confia en context, però la llei ho exigeix, s'ha de complementar: "sigui com sigui, per transacció > Y, demanar MFA per regulació". Això s'integra en polítiques.

- **Sanitat (HIPAA, RGPD sector salut):** Molt enfoc a privacitat pacient. El sistema pot ajudar blocant accessos no autoritzats, però compte: si un metge es bloqueja injustament d'una història clínica i això afecta atenció, podria tenir conseqüències legals. Probablement en entorns de vida o mort s'establiran llindars més permissius i més monitoratge a posteriori en comptes de tallar en sec. Legalment, la confidencialitat és important però també el *duty of care*.

- **Sector públic i defensa:** Podria ser exigible auditar l'algoritme per assegurar-se que no discrimina. En alguns casos, hi ha requisits de *transparència algorítmica* per a IA usada en govern. Potser s'hauria de documentar l'algoritme, fins i tot fer-lo validable per tercers.

Contractes i responsabilitats amb tercers: Si es fan servir solucions de tercers (per ex. integrar Okta, Azure, o un producte de ML extern), cal atendre a la contractació:

- Assegurar en contracte nivells de servei (SLA) adequats (si el servei cau, pot bloquejar tota la companyia).

- Temes de propietat de dades: on es guarden? Compliment si es puja a un cloud (transferència internacional de dades, etc. han de complir GDPR).

- Responsabilitat en cas de fallades: és un terreny nou, però es podria definir fins on arriba la responsabilitat del proveïdor de l'algoritme vs l'empresa usuària, en cas d'un incident de seguretat o d'un dany causat per una decisió errònia del sistema.

Adopció progressiva per garantir compliment: Legalment (i per prudència) pot ser millor implementar el sistema de forma escalonada, a mode de *pilot*, validant que no viola cap

regulació ni causa efectes no desitjats, abans d'expandir-lo. Fer consultes amb comitès d'ètica, delegats de protecció de dades (DPO) i representants d'empleats pot evitar litigis o sorpreses.

8. Conclusions

Al llarg d'aquest treball s'ha explorat la intersecció entre els models tradicionals de control d'accés i la intel·ligència artificial, argumentant la necessitat i la viabilitat d'un **sistema de control d'accés dinàmic, autoadaptatiu i segur** enfront dels reptes dels entorns actuals. A continuació, resumim els punts clau i conclusions obtingudes:

Evolució dels models d'accés: S'ha revisat el recorregut des del DAC i MAC – models rígids nascuts en entorns tancats – fins a RBAC i ABAC, que aportaren major flexibilitat i granularitat. No obstant, fins i tot ABAC, en la seva implementació estàtica, pot quedar curt davant la complexitat dels sistemes distribuïts i la sofisticació de les amenaces. L'aparició del paradigma Zero Trust (NIST SP 800-207) representa un canvi de mentalitat: es passa de controls puntuals a **controls continus i contextuals**, cosa que els models tradicionals per si sols no podien assolir plenament.

Limitacions identificades: En entorns multinuvol, IoT i híbrids, els mecanismes tradicionals pateixen problemes de coherència, escalabilitat i visibilitat. S'ha evidenciat que atacants moderns exploten les bretxes d'aquests models (per exemple, moure's lateralment amb credencials vàlides, sabent que no hi ha reavaluació un cop dins). Encara que s'han introduït millores (MFA, SIEM, etc.), persistia un **buit**: la manca d'una capa d'autorització capaç d'*aprendre* i *adaptar-se* a condicions canviants sense dependre enterament de regles predefinides.

IA aplicada: situació actual i deficiències: Les solucions actuals que incorporen IA (Azure AD Identity Protection, Okta Adaptive MFA, Google BeyondCorp, etc.) confirmen els beneficis de la intel·ligència adaptativa: moltes amenaces són detectades i blocades proactivament. Tanmateix, la majoria actuen com a capes addicionals al marge dels sistemes de control d'accés principals, generant alertes o ajustant certs paràmetres (com exigir MFA) però sense integrar-se completament en la lògica d'autorització. Això comporta que segueix havent-hi decisions preses amb informació incompleta (sense context de risc) o, a l'inrevés, deteccions de risc que no sempre es tradueixen en accions automàtiques. Aquest divorci parcial entre IAM i AI limita l'efectivitat global.

Proposta de model híbrid ABAC+IA: Per superar aquestes limitacions es proposa un model integrat on la IA forma part intrínseca del procés d'autorització. El model combina la **consistència i claredat d'ABAC** – polítiques declaratives basades en atributs i condicions lògiques – amb la **capacitat d'aprenentatge i adaptació de l'IA** – aportant atributs dinàmics com puntuacions de risc, detecció d'anomalies i decisions basades en patrons. S'ha demostrat amb exemples concrets que aquest model pot:

- Reforçar el principi de **privilegi mínim dinàmicament**, permetent per exemple accés condicional a informació sensible només en context de risc baix i requerint validació extra en situacions de risc.

- **Bloquejar proactivament** accions malicioses fins i tot si provenen de comptes legítims, gràcies a l'anàlisi de comportament (com el cas de l'atacant Carles en l'exemple, detectat per "impossible travel" i bloquejat).

- **Reduir falsos positius** respecte enfocaments de regles fixes: el sistema aprèn què és normal per a cada usuari i s'adapta a canvis (per exemple, si un empleat comença a teletreballar regularment des d'un nou lloc, l'IA ajustarà el risc a la baixa amb el temps, evitant que constantment se li bloquegi l'accés).

- **Millorar la visibilitat i auditoria:** cada decisió es pren amb més context, i queda registrada amb motius concrets (atributs i valors). Això facilita complir requisits d'auditoria i comprendre les incidències.

Beneficis col·laterals: Un cop vençudes les reticències inicials, un sistema així també pot aportar **eficiència operativa**. Per exemple, pot reduir la càrrega del personal de TI en recertificacions d'accés, ja que molts accessos innecessaris es podrien detectar i tallar automàticament (el sistema "sap" que cert usuari no ha fet servir mai una aplicació que té assignada i podria suggerir revocar-li). Així mateix, un control d'accés adaptatiu pot donar més **confiança per obrir sistemes** a entorns moderns: per exemple, permetre connectivitat des de mòbils i núvols externs sense comprometre seguretat, ja que es disposa d'un control fi segons el context.

Consideracions i precaucions: Malgrat els avantatges, hem subratllat la importància de desplegar aquest model de manera responsable:

- Garantint **privacitat i ètica**: adoptant un plantejament "Privacy by Design" per no convertir la seguretat en un mecanisme de vigilància excessiva, fent transparent a la plantilla el funcionament (fins al punt que sigui possible sense comprometre seguretat).

- Mantenint humans en el bucle decisional per a casos crítics, i vetllant per evitar biaixos discriminadors en les decisions algorísmiques.

- Assegurant **robustesa tècnica**: no subestimar la complexitat; cal invertir en infraestructures d'alta disponibilitat, en professionals qualificats per entrenar i supervisar els models, i en mesures de ciberseguretat per protegir la pròpia eina (hardening del PDP/AI, proves d'intrusió, etc.).

- Complint el marc legal: alineant la implementació amb GDPR i normatives sectorials, i preparant procediments d'explicació i reclamació de decisions automatitzades, d'acord amb les exigències legals i de bon govern corporatiu.

Contribució del treball i línies futures: Aquest treball ha aportat una visió integradora i actualitzada, recolzada tant en fonts bibliogràfiques com en tendències de mercat, sobre com la

IA pot transformar el control d'accés. S'ha proposat una arquitectura i flux que poden servir de referència per a organitzacions que vulguin adoptar un enfoc similar. Com a línies futures, es poden identificar diverses extensions o aprofundiments:

- **Refinament de models de risc:** Investigació en algoritmes més avançats (p. ex. ús de deep learning seqüencial per predir comportaments futurs, o aprenentatge federat per aprofitar dades d'atacs en múltiples organitzacions sense compartir dades sensibles).

- **Experiència d'usuari adaptativa:** No només restringir quan hi ha risc, sinó potser adaptar l'entorn: per exemple, si es detecta que un usuari està cansat (via patrons de tecleig) potser el sistema li mostra opcions més senzilles en comptes de fer-lo passar per un procés complex de seguretat – això entra en l'àmbit de l'**UX adaptatiu** combinat amb seguretat.

- **Estandardització:** Contribuir al desenvolupament d'estàndards oberts per a polítiques dinàmiques i intercanvi de senyals de risc (alguna cosa ja s'insinua amb iniciatives com CAEP – Continuous Access Evaluation Protocol). Estàndards clars facilitarien a productes heterogenis comunicar-se, beneficiant entorns multivendor.

- **Cost-benefici i impacte organitzatiu:** Estudis de cas quantificant la reducció d'incidents gràcies a sistemes d'IA i comparant-ho amb el cost d'implementació. Això ajudaria a convèncer executius i reguladors de la inversió en aquestes tecnologies.

En conclusió, la integració de la intel·ligència artificial amb controls d'accés és un camí gairebé natural i necessari donades les condicions actuals: expandeix la capacitat humana per gestionar entorns molt més dinàmics i amenaces més subtils. Ja no es tracta de preguntar "té X permís sí o no?", sinó "**aquest accés té sentit en aquestes circumstàncies?**". La IA és l'eina que permet respondre aquesta pregunta complexa en fraccions de segon, basant-se en gran quantitat de dades. El treball present demostra que és factible arribar a un **model autoadaptatiu i segur**, on s'equilibren automatització i governabilitat. Encara queden reptes per polir i resistències que vèncer, però els beneficis en termes de seguretat proactiva i flexibilitat operativa fan que aquesta evolució sigui, probablement, la direcció en què convergirà la gestió d'identitats i accessos en els propers anys. Com deia un principi de la seguretat, "la millor autenticació és la que gairebé no es nota però sempre hi és": en el futur proper, la visió és que el control d'accés sigui gairebé invisible per a l'usuari legítim (perquè l'IA entén el context i no molesta quan tot quadra) i alhora infranquejable per a l'intrús (perquè sempre hi haurà alguna condició contextual que el traeixi). Aquest treball s'emmarca dins d'aquest propòsit, aportant un gra de sorra acadèmic i pràctic per fer-lo realitat.

9. Bibliografia

[1] **Oladoyin Akinsuli (2025).** "Adaptive Access Control: Navigating Cybersecurity in the Era of AI and Zero Trust." *ISACA Now Blog*, 22 abril 2025.

[2] **NIST Special Publication 800-207 (2020).** *Zero Trust Architecture.* National Institute of Standards and Technology.

[3] **Portnox (2016).** "Risk Adaptive Access Control and Why You Should Care About It." *Portnox Blog*, 11 maig 2016.

[4] **Microsoft Learn Documentation (2023).** "Microsoft Entra ID Protection – Adaptive machine learning for risk detections." *Microsoft Learn*, actualitzat 2025.

[5] **Okta, Inc. (2019).** "Okta Launches Risk-Based Authentication Solution with Machine Learning Capabilities, Enhancing Adaptive MFA and SSO." *Nota de premsa*, 2 abril 2019.

[6] **The Hacker News (2025).** "Stolen credentials were the #1 breach vector in 2023/24 (80% of web app attacks)." *Article: The $10 Cyber Threat Responsible for the Biggest Breaches of 2024.*, 16 gener 2025.

[7] **IBM Security (2024).** "Navigating the ethics of AI in cybersecurity." *IBM Security Intelligence Blog*, 16 octubre 2024.

[8] **StrongDM (2023).** "Difference between RBAC vs. ABAC vs. ACL vs. PBAC vs. DAC." *StrongDM Blog*, 2023.

[9] **NIST CSRC Glossary (2015).** Definicions de *Discretionary Access Control (DAC)* i *Mandatory Access Control (MAC)*.

[10] **Arxiv – Karimi et al. (2021).** "An Automatic Attribute Based Access Control Policy Extraction from Access Logs." *arXiv:2003.07270v4*.

[11] **Felix Chad (2025).** "Dynamic Access Control Mechanisms in Serverless Environments for Enhanced Security." *Article de recerca (pre-print)*.

[12] **NIST Special Publication 800-53 Rev.5 (2020).** *Security and Privacy Controls for Information Systems*, controls AC i SI (referència conceptual en text).

11. Annexos

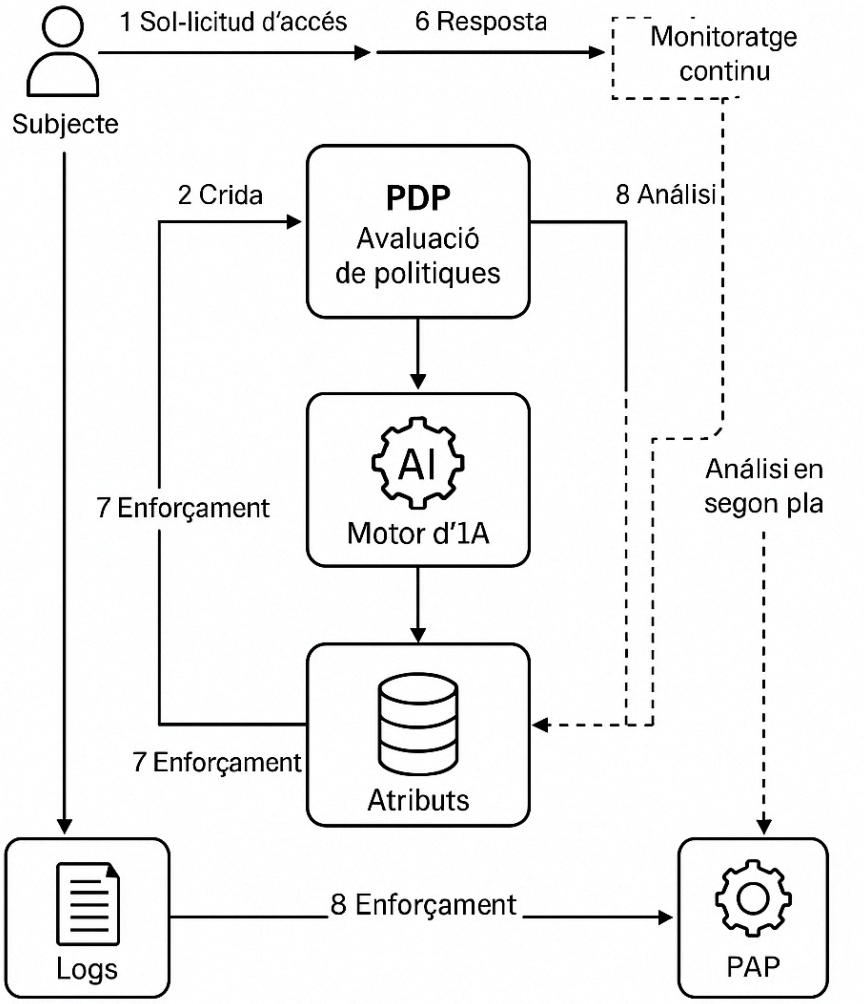

Figura 6.2 Diagrama de'arquitectura del model hibrid ABAC + IA