Conditional Access Engineering
Od architektury k testovatelnému policy enforcementu

Jan Šeda; O2 Czech Republic, a.s.

Credits:

Martin Vokurek, Microsoft; Robert Tibor Gajdzica, O2 IT Services s.r.o.

13. října 2026

ISACA
Czech Republic Chapter

Agenda

1	Conditional Access Policies Internals
2	Persona-based Conditional Access Policies Framework 🖥 demo
3	Conditional Access Policies jako kód: Microsoft365DSC, IaC a CI/CD 🖥 demo
4	Automatizované testování: What If, Pester, Maester 🖥 demo
5	AI v Conditional Access Policies engineeringu 🖥 demo
6	Tři roky provozu: drift, exception debt, více tenantů

ISACA Czech Republic Chapter

Pár otázek na úvod

Kdo chce mít Zero Trust přístup pro CA politiky?

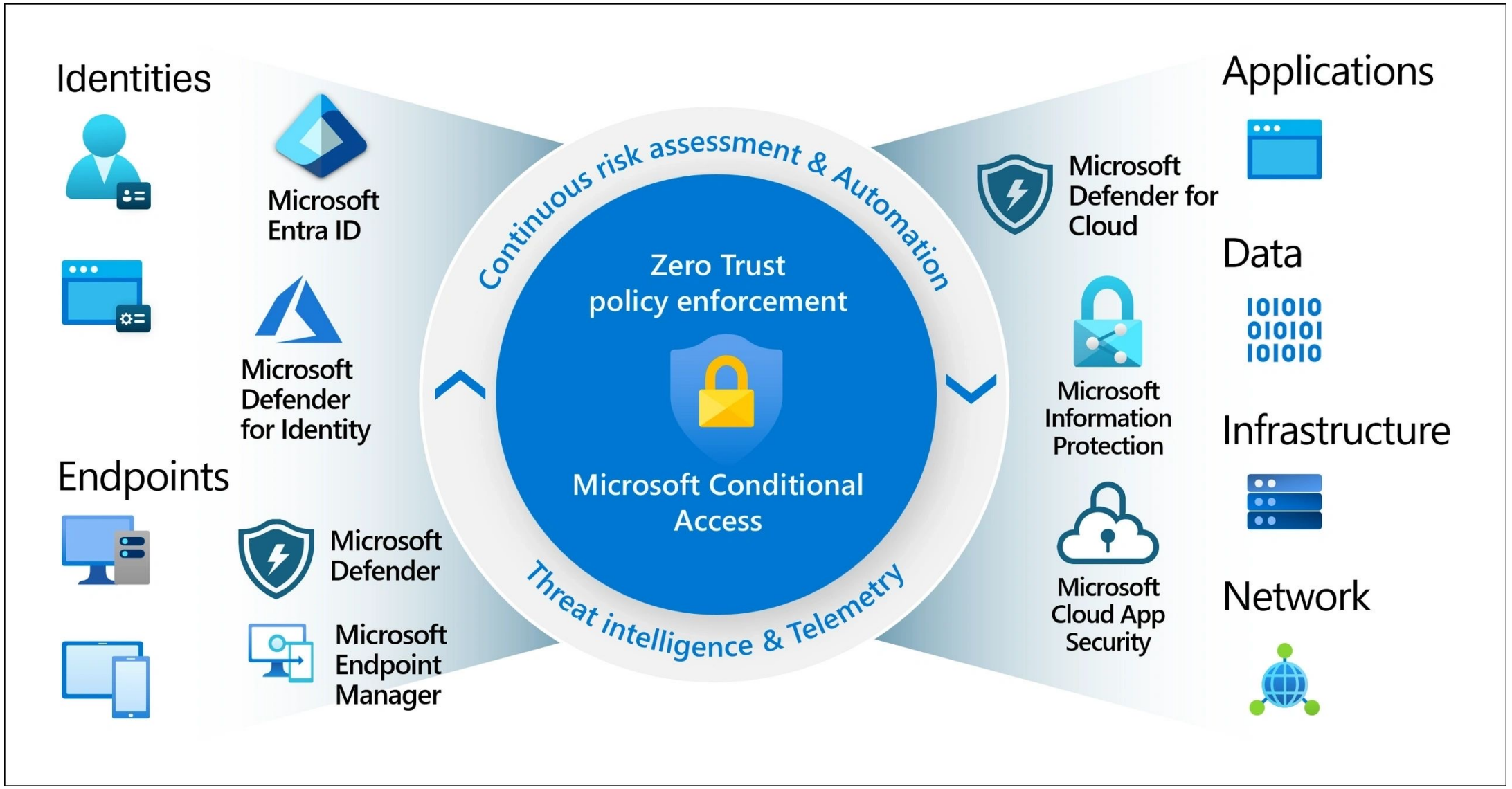
Kdo má ve firmě alespoň 10 a víc CA politik?

Kdo chce CAP verzovat a mít tracking a auditing změn?

Kdo chce řídit lifecycle CAP a zaměřovat je na specifické skupiny uživatelů?

Pokud aspoň jedna otázka s vámi rezonovala, je tenhle tutorial pro vás.

ISACA Czech Republic Chapter


Identities
Microsoft Entra ID
Microsoft Defender for Identity
Endpoints
Microsoft Defender
Microsoft Endpoint Manager
Continuous risk assessment & Automation
Zero Trust policy enforcement
Microsoft Conditional Access
Threat intelligence & Telemetry
Microsoft Defender for Cloud
Microsoft Information Protection
Microsoft Cloud App Security
Applications
Data
Infrastructure
Network

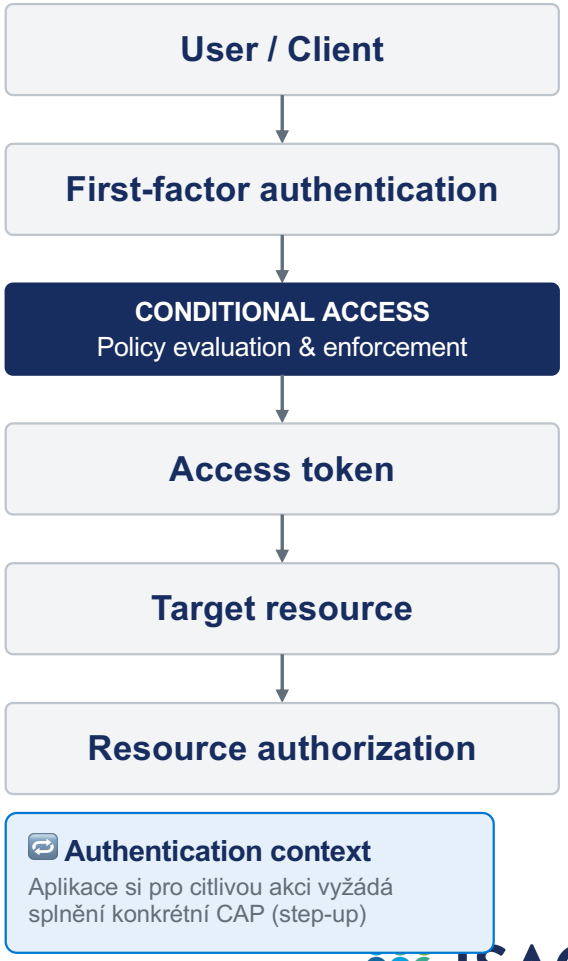
CAP Internals

Jak CAP engine rozhoduje nad konkrétním requestem

ISACA
Czech Republic Chapter

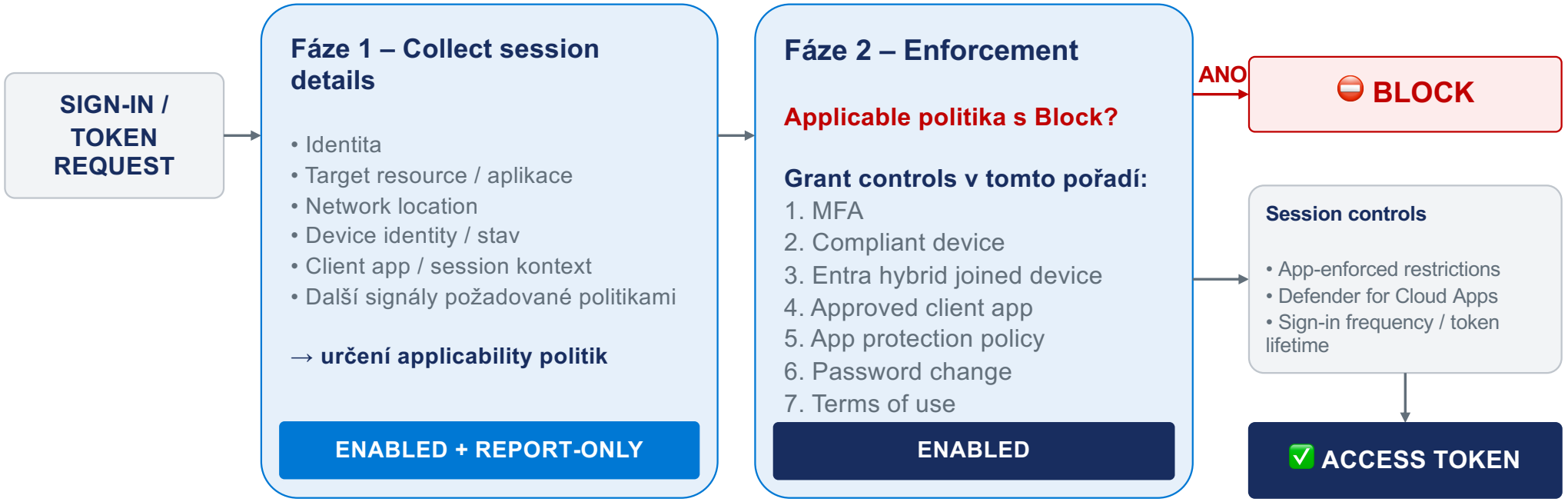
Kde sedí Conditional Access v procesu přihlášení

- 🎟️ Aplikace nekonzumují přihlášení, ale tokeny
  - Pokud CAP zabrání vydání access tokenu, přístup z pohledu aplikace vůbec nevznikne
  - Vyhodnocuje se při vydání tokenu pro konkrétní resource, ne jednou „za přihlášení“
  - Po vydání tokenu CAP již nerozhoduje – výjimkou je Continuous Access Evaluation (CAE) u podporovaných služeb
- 📍 CAP sedí mezi first-factor autentizací a vydáním tokenu
  - Není to „dodatek po přihlášení“, ale rozhodovací bod na cestě k tokenu
  - První faktor sám neprovádí, jen využívá signály vzniklé při autentizaci
  - Není první linií obrany proti DoS ani útoku na ověření identity
- ⚠️ Dvě časté mylné představy
  - Úspěšný první faktor ≠ přístup k resource
  - Úspěšný CAP ≠ autorizace uvnitř aplikace
- 🔍 Typické situace z provozu
  - „Přihlásil jsem se, ale nemám přístup“ → CAP přerušil nebo zablokoval vydání tokenu
  - Stejný uživatel, jiné chování v různých aplikacích → jiný target resource = jiné applicable politiky
  - Uživatel přidán do skupiny, ale politika na něj „nepůsobí“ → drží starý token
- Hard limity pro CAP
  - Max **počet 240 CAP per tenant** (zvýšeno ze 195)
  - **2048 skupin na uživatele – nad limit hrozí zablokování**
  - **195 Named locations**
  - **2000 IP ranges** v jedné Named location



ISACA Czech Republic Chapter

Dvoufázový model vyhodnocení CAP



👉 Report-only politiky projdou fází 1, ale nikdy se neenforcují
👉 Více applicable politik = všechny musí být splněny (AND)
👉 Stačí jedna applicable Block politika a ostatní grant controls už nehrají roli
👉 „Require one of the selected controls“ → výzvy v definovaném pořadí, po splnění je přístup udělen

⚠️ Chybí-li signál z fáze 1 (např. device state z nepodporovaného klienta), grant control nemusí jít splnit → uživatel je zablokován, i když politika Block nemá

ISACA Czech Republic Chapter

Kam všude CAP dosáhne

🌐 Multi-cloud a SaaS federace

• AWS, GCP, Salesforce, SAP, ServiceNow …
• CAP se uplatní i na přístup do prostředí mimo Microsoft, pokud se federuje přes Entra ID

🏢 On-premise aplikace

• Application Proxy – publikace interních aplikací
• Entra ID Private Access (ZTNA) jako moderní náhrada VPN
• AD FS je z pohledu Microsoftu obsolete a rizikové

🖥️ Virtuální desktopy

• Azure Virtual Desktop – CAP řídí přístup k VDI
• Stejné persony a baseline jako u fyzických zařízení

🔒 VPN a síťový přístup

• Cisco, Palo Alto, Fortinet, Checkpoint … přes SAML nebo OIDC
⚠️ Neplatí pro legacy VPN nad RADIUS – NPS se použije jen pro MFA

🤝 Externí identity

• B2B hosté a partneři, Entra External ID
• CAP platí i na guest účty, s vlastní personou

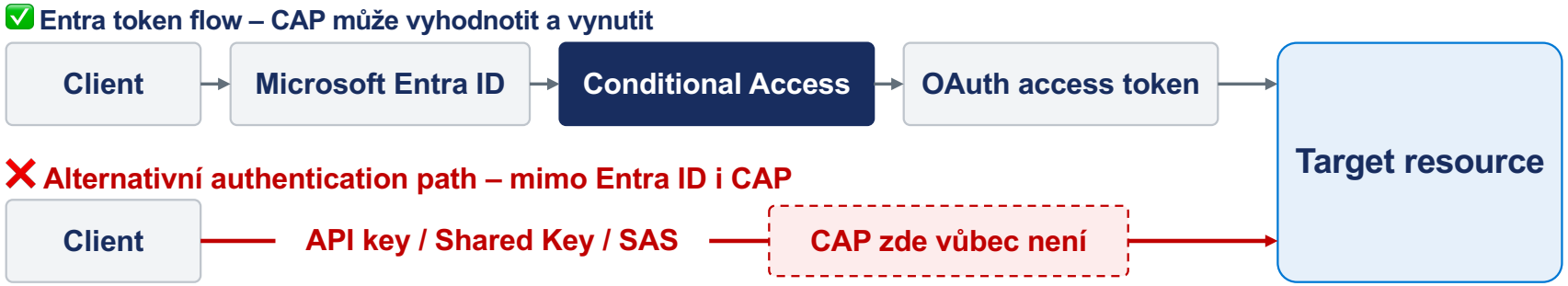
📡 Non-human identity

• Service principals a agenti
• Jiné podmínky a jen Block – viz slide o non-human identitách

Podmínka dosahu: platí to všude, kde aplikace získává token přes Entra ID.
Architektonický přínos je jednotná enforcement vrstva pro cloud i on-premise – kde tato podmínka neplatí, viz předchozí slide o hranicích působnosti.

ISACA Czech Republic Chapter

Hranice působnosti CAP a ochrana alternativních cest



- 1️⃣ **Alternativní cestu odstranit** (preferováno)
  - Vypnout API keys, Shared Key a lokální autentizaci; vynutit Entra ID, OAuth nebo workload identity
  - Nejdřív zmapovat, kdo klíče používá; plošně hlídat přes Azure Policy
- 2️⃣ **Pokud musí zůstat, obalit vlastní vrstvou**
  - API Gateway s OAuth, JWT validací nebo mTLS; network restrictions a private endpoint
  - Rate limiting, logging, secrets ve vaultu, minimální scope a rotace
- 🔍 Co se skutečně používá, ukážou logy resource, ne konfigurace

📌 Příklady z Azure

- Azure Storage – Shared Key a SAS → allowSharedKeyAccess: false
- Azure AI Foundry, AI Search – API key → disableLocalAuth: true
- Azure Service Bus, Cosmos DB → disableLocalAuth: true

👉 **CAP konfigurace nechrání vstupy mimo Entra token flow.**
Dokud má služba dvě paralelní cesty, Conditional Access chrání jen jednu z nich.

⛔ **Mimo Entra token flow patří také:**
connection strings s klíčem · lokální účty služby (SQL auth, SSH) · service-specific tokeny (PAT) · certifikát ověřený mimo Entra · anonymní přístup

ISACA Czech Republic Chapter

Non-human identity v Conditional Access

	Uživatel	Workload identity (service principal)	Agent identity	Agent user account
Cílení politiky	Users, groups, role, guests	Přímo přiřazený single-tenant SP registrovaný v tenantu – ne přes skupinu	Všechny agent identity, konkrétní agenti nebo blueprint (šablona agentů)	Agent users; blueprint je nepokrývá
Grant controls	Plná sada (MFA, authentication strength, compliant device …)	Jen Block	Jen Block – agent nemůže nic interaktivně splnit	Block; compliant device / network jen u session z managed endpointu
Typické podmínky	Risk, location, device, client app …	Location (veřejné IP), service principal risk, authentication context	Agent risk (High / Medium / Low) z ID Protection	Agent execution environments (Preview)
Mimo rozsah	–	Managed identity, Microsoft a multitenant aplikace	Token blueprintu při vytváření agentů	Session přímo v cloudu nemusí dát device signál
Licence	Entra ID P1 / P2	Workload ID Premium	P1 / P2 + Agent 365 (vynucování licence teprve přijde)	P1 / P2 + Agent 365

- 👉 On-behalf-of: agent jedná za uživatele → politika cílí na uživatele a platí jeho grant controls
- ⚠️ Politiky cílené na uživatele nezastaví service principal / workload identity ani autonomního agenta – potřebují vlastní politiky

ISACA Czech Republic Chapter

Persona-based CAP Framework

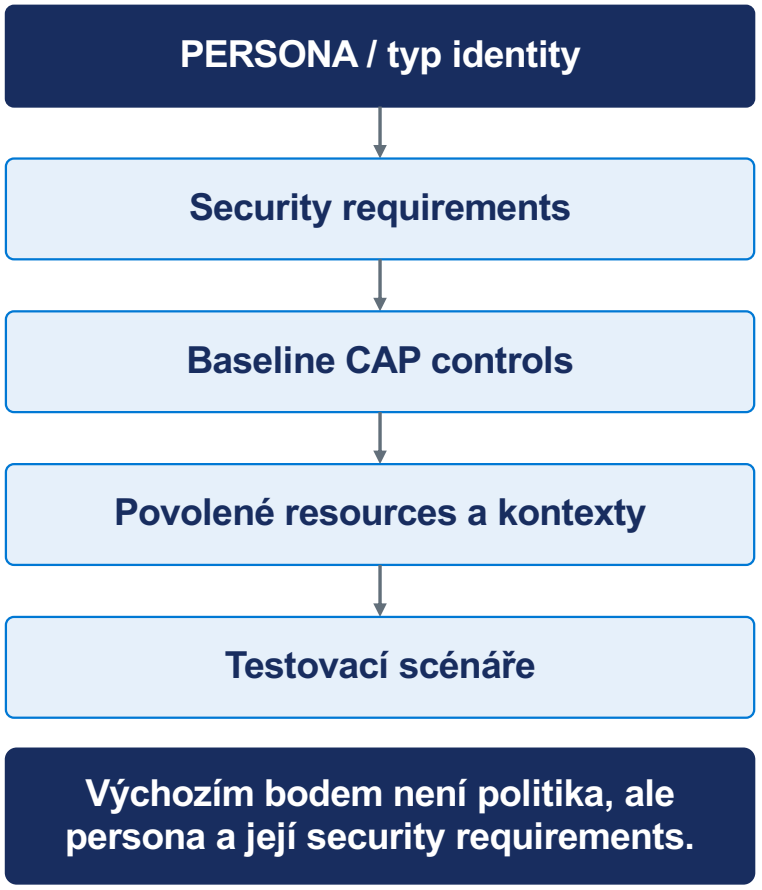
Persona jako bezpečnostní třída identity

Vychází z Microsoft guidance Conditional Access for Zero Trust – rozšířeno o číselné tiery, non-human persony a testovatelný deployment

ISACA
Czech Republic Chapter

Proč persony: strukturovaná architektura CAP

- 🧨 Jak prostředí vypadá bez person
  - Politika 1 – vznikla kvůli incidentu
  - Politika 2 – kvůli nové aplikaci
  - Politika 3 – historická výjimka
  - Politika 4 – jiný tým
  - Politika 5 – nový regulatorní požadavek …
- ⚠️ Důsledky
  - Není strukturovaná a segmentovaná identitní architektura
  - Není jasné, které identity má která politika chránit
  - Politiky se překrývají a zároveň vznikají mezery v pokrytí
  - Politiky a zároveň výjimky se kumulují a nejsou jasné hranice
  - Změna jedné skupiny změní chování více politik
  - Nikdo neumí jednoduše vysvětlit výsledný security model
- Existují také jiné CAP frameworky
  - Podle úrovně ochrany (Microsoft Zero Trust Identity & Device Access), podle resource, podle rizika, podle security tieru
  - Personu volíme proto, že odráží požadavky security contractu i testovací matici a zároveň odpovídá licenčnímu modelu Microsoftu



ISACA Czech Republic Chapter

Persona a její security contract

- 🧠 Persona je architektonická abstrakce
  - Jedná se o bezpečnostní třídu identity
  - Skupina identit s podobným účelem, risk profilem, trust assumptions a governance modelem
  - Není to objekt v Entra ID – je to design pattern
  - Do jedné třídy patří identity se stejnými:
    - Trust assumptions
    - Auth požadavky
    - Device / Network požadavky
    - Resource scope
    - Governance pravidly
- 🔧 Technická realizace
  - Entra security group, directory role, administrative unit
  - Custom security attributes, workload identity scope, explicitní přiřazení
- 🎯 Cíl
  - Malý počet stabilních bezpečnostních tříd identit, ne desítky person

Security contract persony – co musí být explicitně definované

Identity scope
Kdo do persony patří, kdo ji vlastní, jak vzniká a zaniká členství

Authentication
MFA, phishing-resistant MFA, authentication strength, sign-in frequency

Device
Compliant, hybrid joined, approved client app, omezení platforem

Network
Trusted a compliant network, named locations, Global Secure Access

Resource scope
M365, Azure Management, admin portály, vybrané SaaS, on-prem přes App Proxy

Prohibited
Nespravované zařízení, legacy auth, device code flow, neznámý kontext

Exceptions
Owner, důvod, datum expirace, test, monitoring

Owner
Kdo odpovídá za contract, baseline, výjimky a review

Persona je bezpečnostní abstrakce nad identitou, CAP je její technický enforcement.

ISACA Czech Republic Chapter

Vrstvy politik a číselné rozsahy person

Rozsah	Tier	Typ identity	Úroveň kontroly
CA001–099	Global – baseline vrstva, ne persona, tenant wide	všechny identity	Block Baseline
CA100–199	Admins	human	nejvyšší
CA200–299	Internals	human	střední
CA300–399	Externals	human	vysoká
CA400–499	Guests	human	vysoká
CA500–599	GuestAdmins	human	nejvyšší
CA600–699	Developers	human	vysoká
CA700–799	ServiceAccounts	human-operated	nejvyšší
CA800–899	Agents	non-human	jen Block
CA900–999	WorkloadIdentities	non-human	jen Block

Proč číselný index, když je v názvu i persona

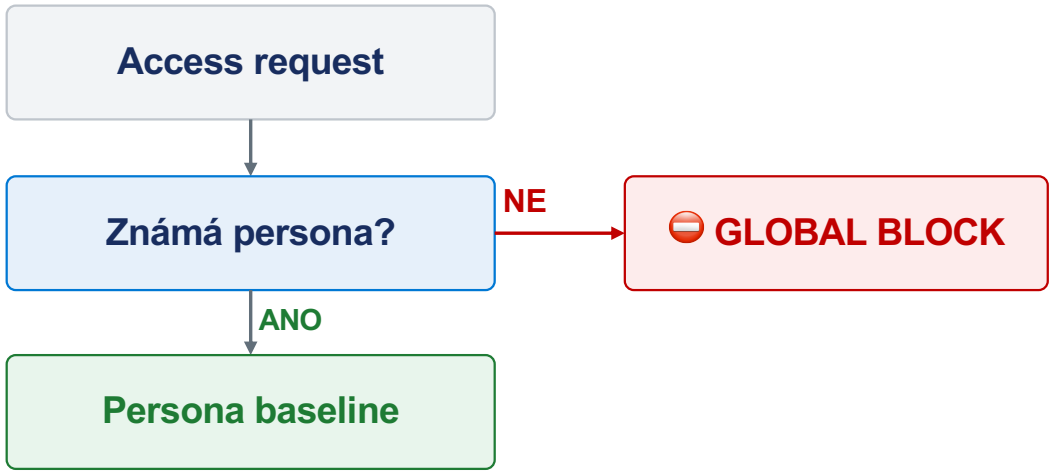
```
CA100-Admins-AllApps
     -AnyPlatform-MFA
```

- 🔎 Orientace v portálu i v kódu – politiky se řadí podle person samy
- 🛡️ SIEM: pravidla nad sign-in logy se píšou regexem na prefix, ne na výčet názvů
- 🧪 DSC a testy: výběr sady podle rozsahu, ne podle seznamu politik
- 👥 Sdílený jazyk se SOC – z čísla je persona zřejmá bez znalosti frameworku

⚠️ Model je cílový stav frameworku. Persona nemusí být v každém tenantu implementovaná – platí konfigurace tenantu.

ISACA Czech Republic Chapter

Global Block Policy: co s identitou mimo persony



- 🚪 Conditional Access není default-deny model
  - Pokud request nespadne do scope žádné enforcement politiky, nic ho automaticky nezablokuje
- 👉 Neznámá ani neřízená identita nemá implicitně získat přístup (fail-safe defaults)

- ⚠️ Co Global Block nesmí zablokovat
  - **Emergency access / break-glass identity**
  - Infrastrukturu potřebnou pro bootstrap a recovery tenantu
  - Workload identity, pokud jsou řízené jiným modelem

- 🧪 Jak ho nasadit
  - Vlastní definice, vlastní testy a dedikovaná pipeline s přísnějším schválením
  - Nejdřív report-only, pak What If, positive i negative testy
  - Řízené schválení a monitoring po nasazení
  - **Pořád si kontrolovat break-glass účty**

ISACA Czech Republic Chapter

Non-human persony

WorkloadIdentities · CA900–999

Hrozba
Ukradené credentials service principalu s vysokými oprávněními

Kontroly
Jen Block – nedůvěryhodná lokace, vysoké service principal risk

Cílení
Service principal přiřazený přímo do politiky, nikdy přes skupinu

Podmínka
Je nutné zakoupit Entra Workload ID Premium per jednotlivá identita

Agents · CA800–899

Hrozba
Autonomní jednání, volání nástrojů, dlouhý běh, měnící se kontext

Kontroly
Jen Block – podle agent risk z ID Protection

Cílení
Tenant-wide: všechny agent identity, konkrétní agenti nebo blueprint

Podmínka
Agent user accounts se cílí samostatně; blueprint je nepokrývá. Vyžaduje Agent 365.

Managed identity · mimo CAP

Hrozba
Žádné secrets, ale také žádné vynucení Conditional Access

Kontroly
CAP se na ni nevztahuje

Cílení
Nelze – řídí se RBAC, síťovými omezeními a access reviews

Podmínka
Samostatná bezpečnostní třída; v Entra reprezentována service principalem, ale workload identity CA nepokrývá

ISACA Czech Republic Chapter

Licence jako architektonická hranice

Co chcete vynutit	Co to vyžaduje	Na co pozor
CAP pro uživatele – MFA, device, session	Entra ID P1	Licenci potřebuje každý uživatel ve scope politiky
Risk-based podmínky – user a sign-in risk	Entra ID P2 (ID Protection)	Bez P2 nelze riziko použít jako podmínku vůbec
CAP pro workload identity (service principal)	Entra Workload ID Premium	Nutné k vytvoření i úpravě politiky; bez licence existující politiky běží, ale nejdou měnit
CAP pro agent identity	Entra ID P1/P2 + Agent 365	Vynucování licencí Microsoft teprve zavádí – stav ověřit
Doporučení od CA Optimization Agenta	Security Copilot – SCU	Vypnutím Copilotu přijdete o doporučení; hlídat spending limits
Cokoli pro managed identity	nelze	Conditional Access ji nepokrývá – řešit RBAC, sítí a access reviews

Licence rozhoduje, co lze vůbec vynutit.

Kde chybí, musí nastoupit jiná kontrola – a to je architektonické rozhodnutí, ne nákupní.

💳 **Licenční model je identity-based: P1/P2 per user, Workload ID per workload identity, Agent 365 per user (agenti jsou kryti licencí vlastníka)**

Persona proto pomáhá mapovat enforcement na licenční požadavky.

⚠ **Licenční model se rychle mění**

Zvlášť u agentů – ověřit těsně před nasazením

ISACA Czech Republic Chapter

Persona jako zdroj pro kód, testy a dokumentaci

Persona se propisuje do repozitáře

```
M365TenantConfig_<tenant>.ps1
   # politiky CA<XXX>-<Persona>-...

Datafiles/
   ConfigurationPart_CAPGroups.psd1
   # Azure_Restricted_CA_<Persona>

Tests/CAWhatIf/
   CA<XXX>.tests.ps1     # test politiky
   <Persona>.tests.ps1   # test persony
```



Naming

```
CA[Tier][SubTier]
  -[Persona]
  -[TargetApps]
  -[Platform]
  -[ControlType]

CA100-Admins-
  AllApps-
  AnyPlatform-MFA
```

- 🔢 Index nese personu; z názvu je zřejmý scope, resource, platforma a typ kontroly
  - Důsledkem je výborné propojení s AI – z názvu pochopí kontext politiky
- 🧪 Každá persona má testovací matici odvozenou ze security contractu
  - Expected allow, block, MFA nebo authentication strength, device requirement, rozsah resources, negativní scénáře a povinný break-glass test
- 🚀 Rollout persony
  - Definice → contract → členství → resources → baseline → report-only → What If → analýza sign-in logů → pilot → enforce → monitoring
- 🔎 Z názvu musí být zřejmý scope, persona, resource, účel a stav
  - Podpora auditu, troubleshootingu, automatizace a reportingu

ISACA Czech Republic Chapter

Anti-patterny persona frameworku

⛔ Persona explosion

- User-CZ, User-DE, User-Mac, User-Sales …
- Persona má vyjadřovat security posture, ne region, platformu nebo oddělení
- Region a platforma patří do conditions, ne do názvu persony

⛔ Persona = jedna politika

- Jedna persona se vždy rozpadá do více politik
- Baseline, resource-specific politiky, session controls a společné globální politiky

⛔ Exclusions jako hlavní design

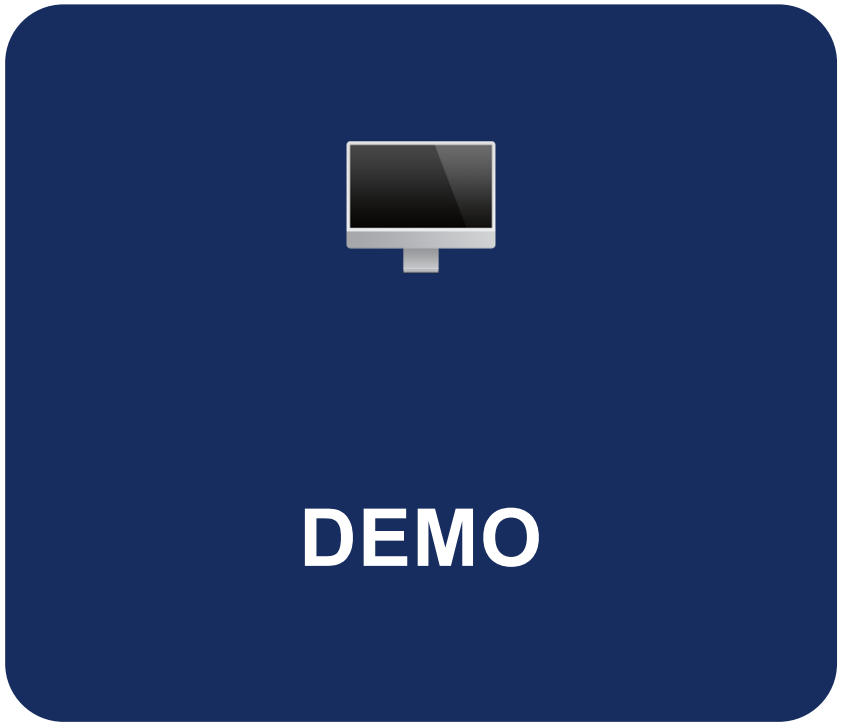
- Include: All users, Exclude: skupina A, B, C, D, E …
- Výsledná logika přestává být auditovatelná
- Lepší je explicitní persona scope a málo dobře zdokumentovaných výjimek

⛔ Break-glass pro pohodlí

- Emergency access slouží k recovery tenantu, ne k obcházení frameworku
- Jakmile se používá provozně, vzniká trvalý bypass
- Výjimka není nový standard, je to evidovaný security debt

ISACA Czech Republic Chapter

Demo: Od persony a security contractu k politikám



ISACA Czech Republic Chapter

CAP jako kód

Microsoft365DSC, IaC a CI/CD governance změn

ISACA
Czech Republic Chapter

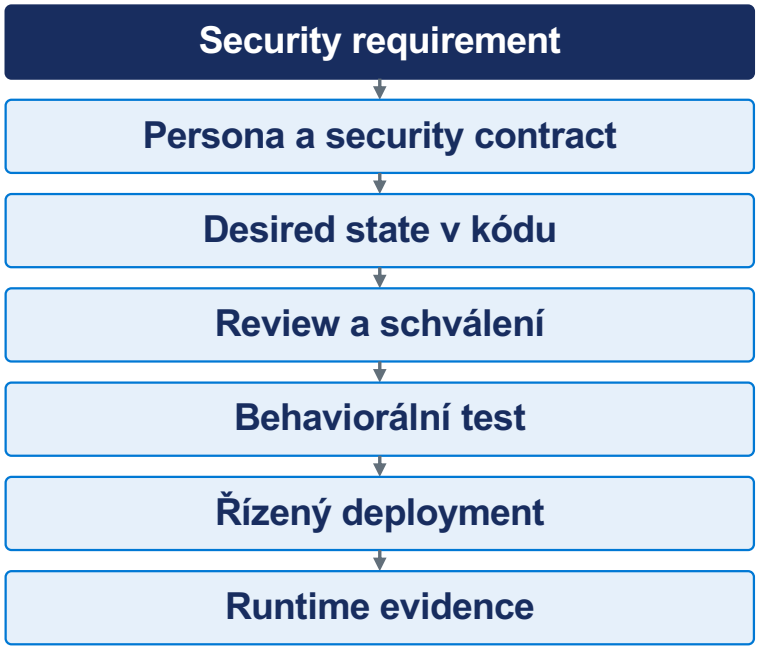
Proč CAP jako kód: od portálu k řízenému procesu

Bez IaC – ruční správa v portálu

- Změny se obtížně verzují a revidují, historie je oddělená od důvodu změny
- Politiky se liší mezi tenanty, desired state nejde doložit
- Ruční změny vytvářejí configuration drift
- Testování je ad hoc, rollback ruční a nejistý
- Změna není napojená na approval workflow
- AI lze zapojit jen pasivně, do konzumace

S CAP jako kód

- Desired state je doložitelný; architektura a persona model jsou sjednocené napříč tenanty
- Deterministická sada testů před každou změnou
- Rollback je revert commitu – jednoduchý a plně řízený
- Každá změna prochází approval workflow a zanechá evidenci
- AI navrhuje změny i testy uvnitř kontrolovaného procesu
- AI agenti míří směrem k DevOps principům



IaC samo o sobě bezpečnost nezvyšuje.
IaC je předpoklad pro automatizované testy, schvalování, least privilege a runtime validace a kontrolované zapojení AI.

ISACA Czech Republic Chapter

Repository jako source of truth

🔑 Repozitář je jediný zdroj pravdy

Co není v repozitáři, není součástí desired state

Tenant je odvozený stav, ne místo, kde konfigurace vzniká → **ruční změna v portálu nepřežije příští deployment**

Jediná výjimka je emergency zásah, který vyžaduje okamžitý efekt

Co z toho plyne

- Každá změna má **autora**, **test**, **schválení**, **audit** a je **dohledatelná**
- Stejný model lze nasadit do libovolného tenantu
- Rollback je revert commitu, ne rekonstrukce z paměti
- Testy a dokumentace žijí vedle konfigurace, ne mimo ni
- Naming convention a skupiny platí stejně ve všech tenantech

Co to stojí

- Změna trvá déle než klik v portálu a vyžaduje Git a review
- Část objektů zůstává mimo IaC – skupiny, named locations, licence

Struktura repozitáře

```
M365TenantConfig_<tenant>.ps1    # konfigurace tenantu

Datafiles/
   TenantData_<tenant>.psd1      # parametry tenantu
   ConfigurationPart_CAPGroups.psd1

Tests/CAWhatIf/
   CA<XXX>.tests.ps1             # testy politiky
   <Persona>.tests.ps1           # testy persony

docs/Implementation-guide.md

.github/copilot-instructions.md
```

ISACA Czech Republic Chapter

Pipeline a Pull Request jako security control

Pull Request = change record + peer review + security evidence

- Diff z přehledu commitů a důvod změny
- Dotčené persony a resources
- Risk impact a test evidence
- Výjimky
- Rollout a rollback plán

Review má dvě vrstvy

- Technická: syntax, properties, resource IDs, state, dependencies
- Bezpečnostní: identity a resource scope, exclusions, grant a session controls, persona baseline, lockout scénáře, coverage
- „Kód se zkompiluje“ není security review

Chain of evidence: Git, pipeline a audit logy tenantu dohromady doloží, kdo, proč, co, kdy, kdo schválil a co bylo nasazeno

WHY – [work item] → **WHAT – PR a diff** → **WHO – autor a reviewer** → **TEST RESULT** → **APPROVAL** → **DEPLOYMENT**

- 🔁 DevSecOps je výchozí princip – bezpečnostní kontroly běží uvnitř pipeline, ne až po nasazení
  - Pipeline není jen automatizace, ale enforcement: kdo smí změnu navrhnout a schválit, jaké testy musí projít, jaká identita ji nasazuje a jak se ukládá evidence
- 👥 Contributor → Reviewer → Approver → Pipeline identity → Tenant
  - Separation of duties – autor změny ji nesmí sám schválit a nasadit

🤖 **AI-assisted review je jedna z kontrol pipeline a do budoucna očekávaný sekundární reviewer.**

ISACA Czech Republic Chapter

Staged enforcement a governance podle rizika změny



Každá enforcement změna jde zleva doprava. Jak daleko musí projít, určuje riziko.

NÍZKÉ RIZIKO

- Description
- Naming
- Dokumentace
- Komentáře a formátování
- Úprava popisu a odkazu na work item

→ Standardní schválení
Flow: netýká se, mění se jen metadata

STŘEDNÍ RIZIKO

- Nová report-only politika
- Rozšíření pilot scope
- Změna session controls
- Přidání resource do existující politiky

→ Security review
Flow: report-only → pilot → enforce

VYSOKÉ RIZIKO

- Global Block, All users, All resources
- Admin persona, emergency exclusions
- Authentication strength, delete politiky
- Přístup workload identit

→ Multi-party approval + staged rollout
Flow: všechny fáze, bez zkratky

Global Block má v repozitáři zvláštní režim – největší blast radius celého frameworku

Vlastní definice, vlastní sada testů, dedikovaná pipeline a přísnější schvalování – odděleně od ostatních CAP
Povinné What If regresní testy, křížové ověření break-glass účtů a report-only před každou změnou scope.

ISACA Czech Republic Chapter

Anti-patterny v CAP jako kód

⛔ Portál je source of truth

Git slouží jen jako občasná záloha

⛔ Merge → okamžitý enforcement

Bez report-only, testingu a staged rolloutu

⛔ Jedna pipeline identita pro vše

S nadměrnými oprávněními pro všechny fáze

⛔ Client secret roky v CI/CD

Dlouhodobý credential s velkým dopadem úniku

⛔ Noční export přepisuje main

Neautorizovaný drift se stane desired state

⛔ IaC diff = security test

Diff neukáže změnu výsledného chování

⛔ Branch protection lze obejít

Všichni vývojáři mohou pushnout přímo do main

Cíl

Reprodukovatelný a dokazatelný security change process, ne maximum automatizovaných změn

- 🤖 AI (GitHub Copilot) smí v lifecycle navrhovat kód, testy i popis PR – ale **vždy přes Git diff, deterministické testy, review podle rizika a lidské schválení**
- ⛔ Nikdy AI → produkce bez HITL

ISACA Czech Republic Chapter

Čím řídit CAP jako kód: kritéria a kompromisy

Terraform

azuread / msgraph provider

- State file = záznam o stavu spravovaném Terraformem: nástroj ví, co sám vytvořil → umí smazat, přejmenovat, přesunout
- terraform plan ukáže před nasazením co vznikne, změní se a zanikne; zamykání state brání souběžnému nasazení

⚠️ State file je další chráněný artefakt s citlivou konfigurací a může se rozejít s realitou

⚠️ Pokrytí vlastností CAP zaostává za Graphem (např. application filter)

👉 **Pro obecnou infrastrukturu velmi silný nástroj**

Bicep

ARM a rozšíření pro Graph

- **CAP není ARM resource**
- Rozšíření pro Microsoft Graph pokrývá jen vybrané Entra objekty (aplikace, skupiny, service principals)
- Pro Azure IaaS a PaaS zůstává výborné řešení

👉 **Pro Conditional Access dnes ne**

Microsoft365DSC

Naše volba

- Desired state v .ps1 konfiguraci, kompilovaný do MOF (build artefakt – chránit, může nést credentials)
- Aktuální stav čte při každém běhu z tenantu – žádný state file navíc
- Nativní resource pro CAP i Exchange, Teams a SharePoint; export; více tenantů

⚠️ Neví, co sám vytvořil – smazání politiky je explicitní krok: Ensure = Absent

⚠️ Slabší plán před změnou (Test-M365DSCConfiguration není terraform plan)

👉 **Rozhodlo nativní pokrytí objektů CAP**

Microsoft Graph TCM

Sledovat

- Má nabízet v MS prostředí, co nabízí Terraform
- Nativní deklarativní vrstva přímo nad Graphem
- Baseline a monitory driftu; CA mezi podporovanými resources
- API obecně dostupné, produktová nadstavba Entra Tenant Governance
- Feature parity s Microsoft365DSC zatím neověřená

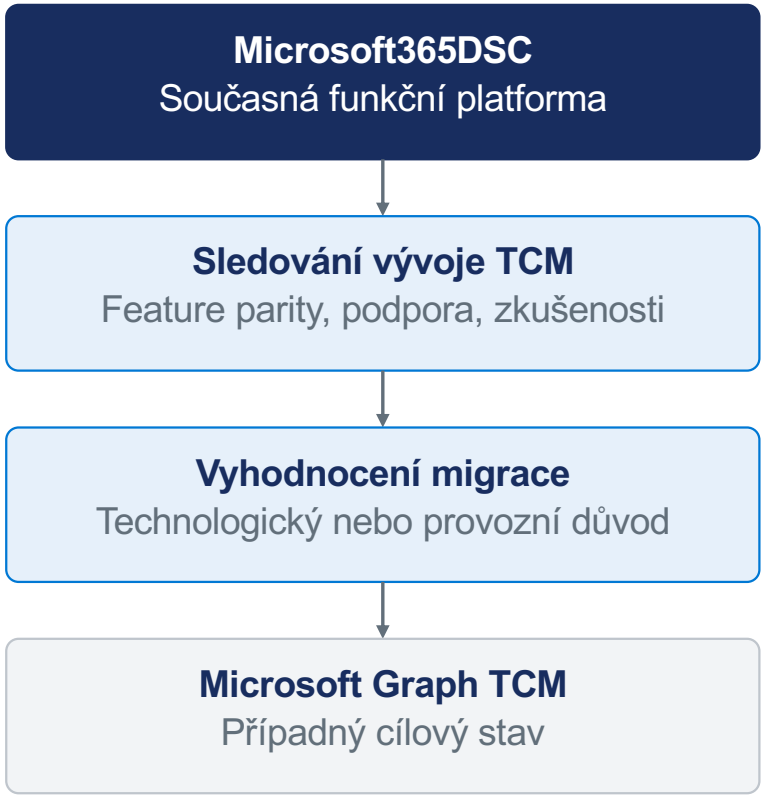
👉 **Vyhodnotit později jako samostatný projekt**

Není to volba lepšího nástroje, ale specializovaného před obecným. U Conditional Access je navíc automatické mazání spíš riziko než přednost – ztráta kontroly bývá tichá.

ISACA Czech Republic Chapter

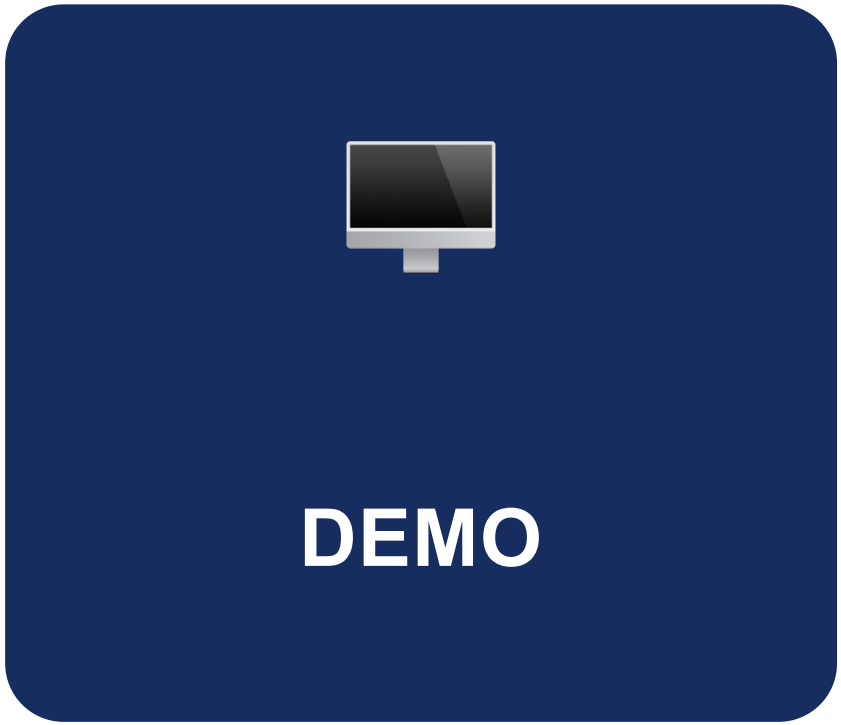
Microsoft365DSC dnes a výhled na Microsoft Graph TCM

- 🧰 Microsoft365DSC je funkční Configuration-as-Code nástroj pro Microsoft 365 a Entra ID
  - Naše implementace stojí na PowerShellu a Microsoft365DSC
  - Export, deklarace desired state, deployment, porovnání konfigurací a reporting driftu
  - Nezaměňovat s Microsoft DSC v3 – jiná technologie a jiná vývojová větev
  - ⚠️ Modul se aktivně vyvíjí – pinovat verze a nové verze testovat mimo produkci
- 🌐 Tenant Configuration Management (TCM) – deklarativní správa konfigurace tenantu přes Microsoft Graph
  - Microsoft staví nativní Configuration-as-Code vrstvu přímo nad Graph API; produktovou nadstavbou je Microsoft Entra Tenant Governance
  - Conditional Access už mezi podporovanými resources je, ale pokrytí všech scénářů Microsoft365DSC zaručené není – ověřit feature parity
- 🧭 Náš postoj: sledovat vývoj, nemigrovat bez důvodu
  - Existence nové platformy sama o sobě není důvod k migraci; případnou migraci řešit později jako samostatný projekt
- ⏩ Souběžně běží přechod na PowerShell 7
  - Microsoft365DSC vyžaduje od října 2026 PowerShell 7.6 a vyšší; v pipeline jde o přepnutí kroků na pwsh a instalaci modulu na agenta
  - Microsoft ukončuje podporu Graph PowerShell SDK na Windows PowerShell 5.1 – dvanáctiměsíční retirement období od 16. 9. 2026, verze v3 jen pro PS 7



ISACA Czech Republic Chapter

Demo: Implementace politik v kódu



ISACA Czech Republic Chapter

Automatizované testování CAP

What If, Pester a Maester

ISACA
Czech Republic Chapter

Pre-build testy: brána před kompilací MOF

Kdy to běží
Gate v build pipeline
- Běží před kompilací MOF
- Neprojde → build selže, deploy se nespustí
- Statická analýza .ps1, bez volání do tenantu

Kontrola souboru
Integrita a supply chain
- Kódování UTF-8 BOM
- Import-DscResource Microsoft365DSC
- Připnutá ModuleVersion, ne „latest“

Kontrola každé politiky
Bezpečnostní invarianty
- ExcludeGroups obsahuje break-glass skupinu
- ExcludeUsers obsahují emergency účty
- Chybí-li vyloučení, build spadne

Pre-build test v Pesteru (zkráceno)

```
Describe 'Pre-Build Tests' -Tag 'PreBuild' {
 Context 'CAP Policy "<_.PolicyName>"' -Foreach $policies {

  It 'Has ExcludeGroups containing break-glass group' {
    $m = [regex]::Match($_.PolicyBody,
         'ExcludeGroups\s*=\s*@\((?<Values>.*?)\)')
    $m.Success | Should -BeTrue
    $m.Groups['Values'].Value |
         Should -Match '\$AADGroups_CA\.Breakglass'
  }
 }
}
```

Nejlevnější místo, kde chybu zachytit

Chybějící vyloučení break-glass účtu se najde v textu konfigurace – dřív, než vznikne MOF a dřív, než se cokoli dotkne tenantu.

Deploy běží jen po úspěšném CompileAndTest a nasazuje přesně ten artefakt, který prošel testy.

ISACA Czech Republic Chapter

Nástroje a průběh automatizovaného CAP testu

Pester
Test execution engine pro PowerShell

- Describe → Context → It, assertions přes Should
- Dvě fáze běhu: Discovery a Run (Pester 5+)
- Neřeší logiku Conditional Access

Maester
Security testing framework nad Pesterem

- Používá Pester jako engine
- Hotové testy pro Entra ID a M365, vlastní testy
- Test-MtConditionalAccessWhatIf volá Graph

What If
Conditional Access evaluation engine

- Graph API nad reálnou CA logikou
- Simuluje konkrétní sign-in scénář
- Vrací applicable politiky a controls

Persona contract → **Maester test** → **Pester: Discovery → Run** → **What If API** → **Applicable politiky a controls** → **Assertion → PASS / FAIL**

Co se ve skutečnosti testuje

```
Given:
  Persona     = Administrator
  Application = Azure Management
  Device      = Unmanaged
  User Risk   = High

Expected:
  Success = Block
```

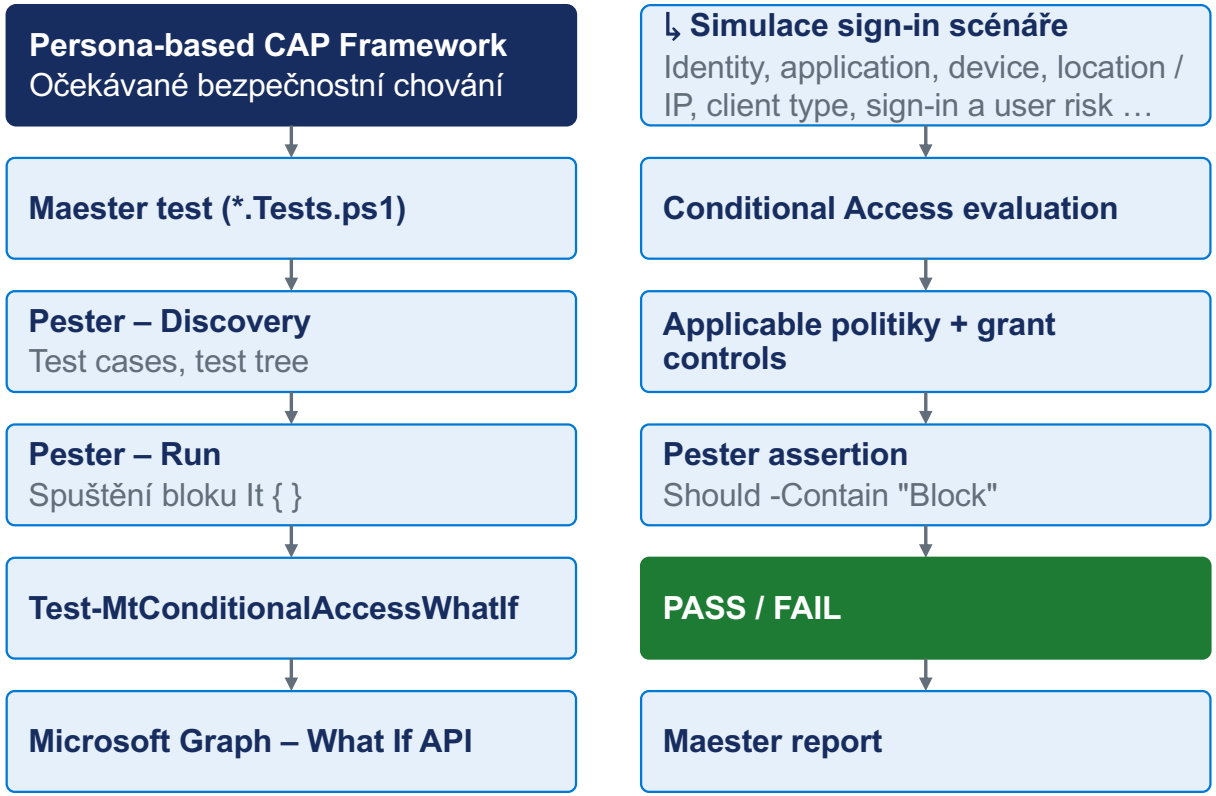
Posun v otázce

Z „Máme správně nakonfigurované politiky?“
na „Chová se výsledná sada politik správně pro definované scénáře?“

Regresní testy při každé změně politik; napojení na deployment proces.

ISACA Czech Republic Chapter

Jak probíhá automatizovaný CAP test



Co testujeme

```
Given:
  Persona     = Administrator
  Application = Azure Management
  Device      = Unmanaged
  User Risk   = High

Expected:
  Success = Block
```

Posun v otázce

Z „Máme správně nakonfigurované politiky?“

na „Chová se výsledná sada politik správně pro definované scénáře?“

- 🔁 Regresní testy při každé změně politik
- ⚙️ Napojení na deployment a IaC proces

ISACA Czech Republic Chapter

Persona model jako testovací matice

Rozsah	Persona	Scénář	Očekávaný výsledek	Typ testu
CA100–199	Admins	Azure Management, compliant PAW, phishing-resistant	allow	✅ positive
CA100–199	Admins	Azure Management, nespravované zařízení	block	⛔ negative
CA100–199	Admins	break-glass účet	politika se neaplikuje	🚨 povinný
CA200–299	Internals	M365, compliant device	MFA → allow	✅ positive
CA200–299	Internals	legacy auth klient	block	⛔ negative
CA400–499	Guests	SharePoint Online, browser	MFA a session control	✅ positive
CA900–999	WorkloadIdentities	SP mimo povolené IP rozsahy	block	⛔ negative
CA001–099	Global	identita mimo persony	block (Global Block)	⛔ negative

- 🧪 Dvě úrovně testů: **CA<XXX>.tests.ps1** pro politiku, **<Persona>.tests.ps1** pro celou personu
- 🔢 Sada se vybírá podle rozsahu, ne podle seznamu politik – proto je index v názvu
- ⚙️ Behaviorální testy neblokují merge, ale deploy ano – běží jako samostatná pipeline s auditní stopou; další tvrdou bránou je pre-build
- 🔐 Kdo změní test, změní bezpečnostní specifikaci – testy pod stejným review jako politiky; identita volající What If má minimální oprávnění a liší se od deployment identity

🚨 Break-glass test je povinný

U každé politiky cílené na uživatele se ověřuje, že emergency účet do politiky nespadá.

Co dělá z testu bezpečnostní kontrolu

- 🔴 Failed test má dvě možné příčiny
  - Zestárnul test – změna členství, jiná testovací identita, nové App ID, změna named location, výpadek API
  - Změnilo se chování – politika, vyloučení nebo podmínka mění efektivní výsledek
- 🧭 Postup: ověřit vstupy testu → spustit scénář ručně ve What If → porovnat se sign-in logem

Pět podmínek, aby test byl kontrola

1. Test umí správně selhat – ověřit rozbitím funkčního CAP v testovacím tenantu
2. Failed výsledek musí někdo vyšetřit a opravit
3. Test se udržuje – stárne stejně jako politika
4. Test je chráněný – stejný review jako politiky
5. Výsledek se archivuje – test, výsledek i schválení

Bez těchto pěti věcí je test jen zelená fajfka v pipeline. S nimi je to doložitelná kontrola.
Kde se změna zastaví: pre-build a security scan blokují build, What If je podmínka pro deploy, DSC test po kompilaci se jen zaznamená.
Opakovatelný test, archivovaný výsledek a doložené schválení tvoří důkaz pro ISO 27001 i NIS2.

Co What If ověří a co ne

- ✅ Co WIF spolehlivě odchytí
  - Chyba v rozsahu – politika zasáhne víc nebo míň identit, než jste chtěl
  - Chybějící vyloučení – break-glass účet se dostal do scope
  - Špatná kombinace – dvě politiky zvlášť v pořádku, dohromady blokují
  - Regrese po změně – úprava CA210 rozbije personu, kterou pokrývá CA205
  - Špatné grant controls – MFA tam, kde být nemá, nebo chybí tam, kde má
- ⚠️ Co WIF evaluace neumí
  - Co je jen v Gitu nebo Disabled
  - Chybějící vstup při definici WIF = neúplná simulace (lokace, riziko, device)
  - Resource jen jako App ID, skupiny aplikací shodu nevrátí
  - Netestuje service dependencies – test pro Teams nezahrne Exchange Online

⚠️ **Report-only nevyhodnocuje politiky na User actions**

- Jsou dvě: Register security information (registrace MFA a SSPR údajů) a Register or join devices. Report-only u nich negeneruje žádná data – What If funguje, chybí měření.
- → Pilotuje se scopem: Enabled na malou skupinu a postupné rozšiřování.

What If je silný test CA logiky, ale není 100% ekvivalent end-to-end sign-inu.

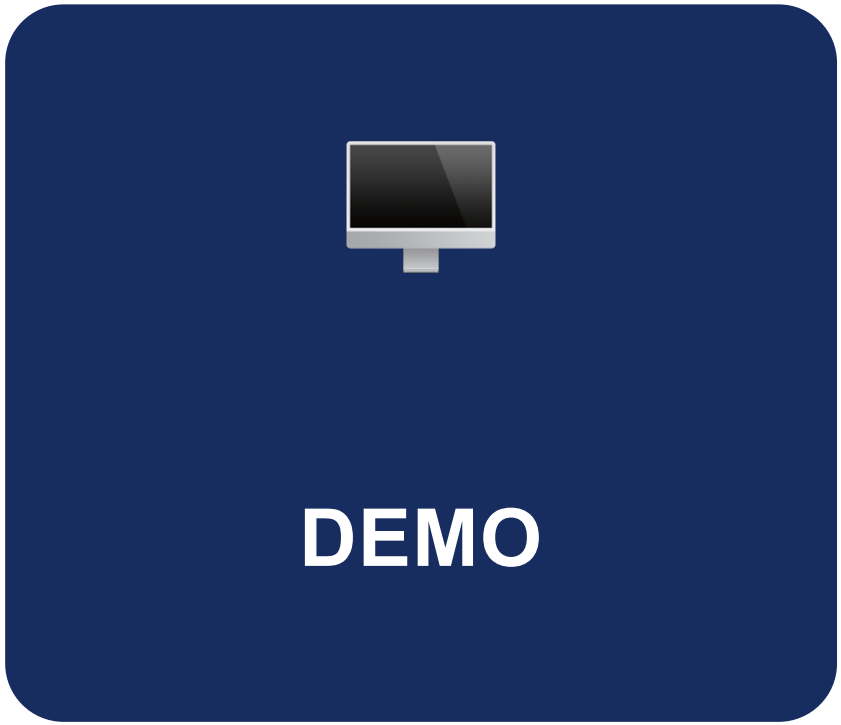
Co o reálném přihlášení neví

- Jestli uživatel MFA skutečně projde
- Jestli má vyhovující metodu zaregistrovanou
- Jaký compliance stav vrátí Intune
- Jaké riziko spočítá Identity Protection
- Jaký protokol klient použije
- Jaký token se vystaví a na jak dlouho
- Co udělá Continuous Access Evaluation
- Jestli přístup povolí sama aplikace

Zelený What If znamená, že se politiky rozhodly správně. Ne že se uživatel přihlásí.

ISACA Czech Republic Chapter

Demo: Conditional Access a testování



ISACA Czech Republic Chapter

AI v CAP engineeringu

AI asistuje. Deterministické kontroly ověřují. Člověk rozhoduje.

ISACA
Czech Republic Chapter

Zero Trust platí i pro AI agenty (obzvlášť pro ně)

Nedůvěryhodné vstupy

- Obsah repozitáře
- Komentáře v pull requestu a issues
- Dokumentace a instrukční soubory
- Výstupy nástrojů a MCP serverů
- Externí data

Co agent podle režimu umí

- Číst repozitář a měnit soubory
- Spouštět příkazy
- Vytvářet branch a pull request
- Používat nástroje a MCP
- Získávat externí kontext

Rizika

- Prompt injection
- Škodlivý obsah repozitáře
- Zneužití nástrojů
- Únik secrets
- Eskalace oprávnění, lateral movement, supply chain

🔁 **Jak s agentem pracujeme: agent začíná až po schváleném plánu a nikdy nekončí u sebe**

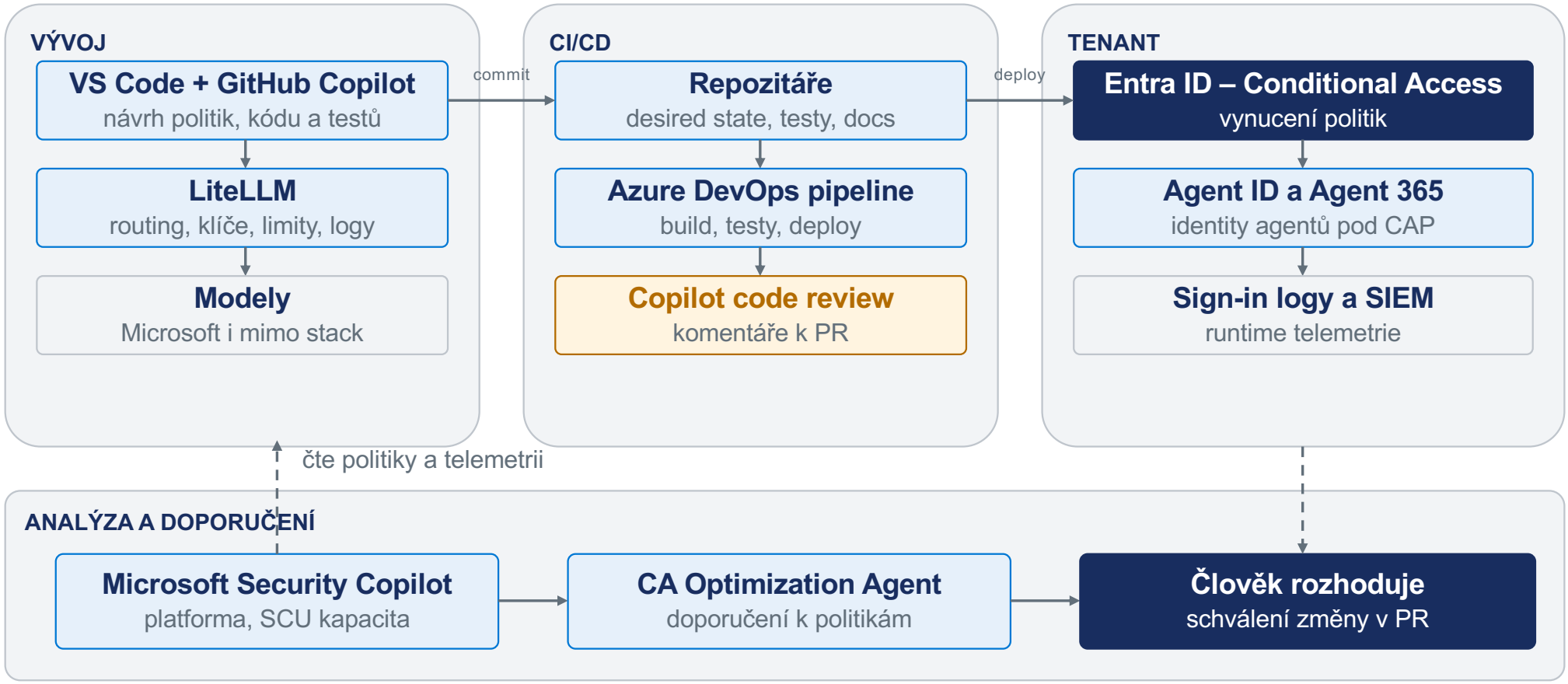
Plan → **Architecture Review** → **Agent** → **Code / Pester / Maester** → **Pull Request** → **Code Review**

- 🛡️ Guardrails
  - Least privilege; read-only tam, kde AI jen reviewuje
  - Sandboxované spouštění, omezené nástroje a síťový přístup
  - Runtime protection pro celý sandbox environment
  - Žádné produkční credentials v dosahu agenta, žádné přímé nasazení do produkce
  - Git diff jako auditovatelný artefakt a deterministické brány v pipeline

AI agent je privilegovaná non-human identita. Musí podléhat Zero Trust stejně jako pipeline identity.

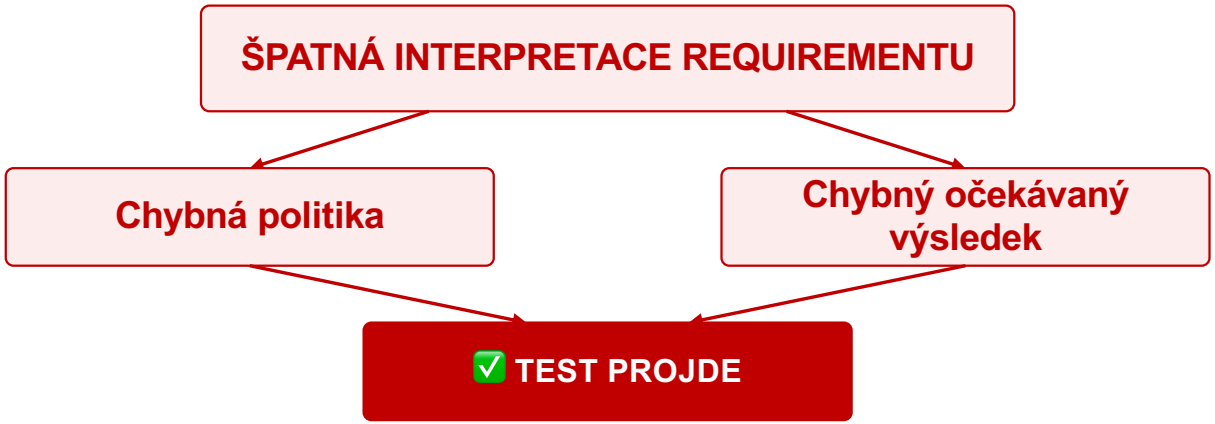
ISACA Czech Republic Chapter

Jak to spolu souvisí: komponenty a toky

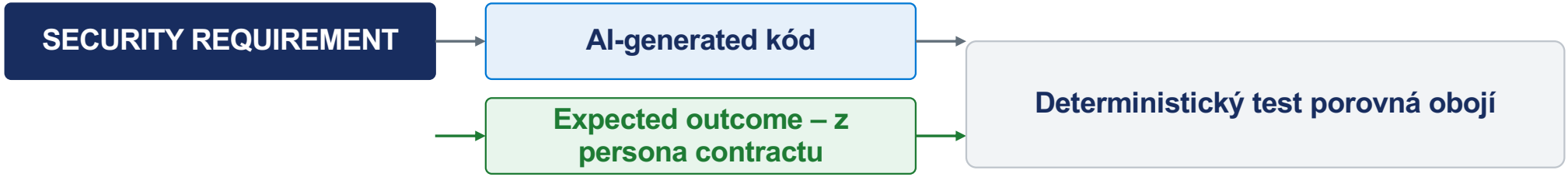


ISACA Czech Republic Chapter

Correlated failure: AI napíše chybný kód i chybný test



Správný model: očekávaný bezpečnostní výsledek pochází odjinud než generovaný kód



- 🧭 Zdroj očekávaného výsledku: persona contract, architektonický standard, explicitní test, lidmi schválené invarianty
- 🔀 Totéž platí napříč nástroji: rozpor mezi modely je signál k ověření u zdroje, shoda není důkaz

AI může test implementovat. Nemá sama určovat, co je správný security outcome.

ISACA Czech Republic Chapter

Conditional Access Optimization Agent

✅ Co agent nabízí

- Doporučení k pokrytí a kvalitě politik nad reálným tenantem
- Přizpůsobení přes knowledge base: .docx nebo PDF, 1 dokument na tenant, max 5 MB; MS avizuje .md
- Je doplňující detekční vrstva
- Je nutné provést napromptování agenta a naučit ho chápat logiku frameworku a kontext prostředí
- Pro report-only politiku cílenou na All users navrhne pětifázový rollout
- Fáze skládá podle velikosti skupin, historie politik a aktivity uživatelů
- Pojistky: pauza nad 10 % blokovaných, report-only pod 90 % úspěšnosti, zrušení při změně grant controls

⛔ Proč to u nás [zatím] není cesta k nasazení

- Agent vytvoří novou politiku přímo v tenantu → druhá deployment cesta mimo repozitář
- Z pohledu IaC je to drift; rozpadne se i řetězec evidence – změna bez commitu, PR a testu
- Agent aktuálně postrádá sílu CI/CD deploymentu
- Neškáluje na více tenantů – scénář s testovacími a dceřinými tenanty vyžaduje IaC
- Neumí cílenou změnu na míru – např. nová politika pro VPN skupinu a konkrétní aplikaci, kterou je potřeba navrhnout, odladit a otestovat
- Aktuálně vhodný pro jednoduché scénáře založené na telemetrii

👉 Doporučení od agenta ano. Druhá cesta do produkce ne.

Až agent místo zápisu do tenantu otevře pull request, zapadne do stejného procesu vedle Copilot code review → zúročí se síla CAP Zero Trust frameworku i AI. Modely jsou komplementární: agent pozoruje jeden tenant, IaC přenáší stejný model přes všechny.

Copilot code review v Azure DevOps

- ⚙️ Jak se zapíná
  - Limited preview – zapíná Project Collection Admin pro organizaci, Project Admin pro projekt, správce repozitáře pro jednotlivé repo
  - Copilot se pak objeví jako reviewer v pull requestu
- 🔁 Jak běží
  - Na vyžádání, nebo automaticky u nových PR – politiku lze omezit jen na vybrané cílové větve
  - Effort level low / medium / high – vyšší úroveň znamená víc tokenů a vyšší cenu
  - Běží na Azure Pipelines poolu nebo Managed DevOps Pool (Ubuntu); self-hosted agenti podporovaní nejsou
- 💸 Účtování
  - Přes Azure subscription jako GitHub AI credits (1 credit = 0,01 USD), meter GitHub Copilot for AzDO; budget alerty v Cost Management

✅ **AI je nehlasující reviewer**

Copilot vždy zanechá jen komentář. Nikdy PR neschválí ani nevyžádá změny.
→ **Neplní roli povinného schvalovatele a neblokuje merge – aspoň zatím ne**

⚠️ **Na co si dát pozor u CAP**

• Po novém commitu se review neopakuje samo – jinak schvalujete podle review staršího kódu
• Copilot nečte odpovědi na své komentáře a nenavazuje
• Limity: repozitář do 10 GB, nejvýš 100 změněných souborů a 100 změn, jen aktivní PR bez konfliktů
• Souběžně 5 review na organizaci a 2 na uživatele

ISACA Czech Republic Chapter

AI v review změn CAP: rozdělení rolí

- ⏱️ Čtyři oči na každý merge a deployment jsou v dnešním tempu blokátor
  - Nezvládnutelné už před nástupem AI, tím spíš dnes, kdy se zbytek procesu automatizuje
- 🧍 Formální druhý schvalovatel kvalitu nezvýší
  - Nemá znalost prostředí ani dopadů změny a nedokáže ji nabrat ad hoc
  - Vzniká difuze odpovědnosti – autor spoléhá na schvalovatele, schvalovatel na autora
  - Razítko vytváří falešný pocit jistoty, což je horší než přiznaná absence kontroly
  - **A celé to vede k neefektivitě a obrovskému nárůstu neproduktivní latence**
- 👉 Proto jeden člověk, který změnám rozumí, plus AI jako reviewer
  - Začít s nehlasujícím reviewerem a případně přejít na hlasujícího reviewera.
- 🔧 Nástrojově: Copilot Code Review nad pull requestem s kontextem repozitáře
  - V Azure DevOps v limited preview – jak přesně funguje, viz slide o Copilot code review

🤖 AI – [nehlasující] reviewer

Konzistentní kontrola celého diffu, pokaždé stejně

- Odpovídá změna důvodu ve work itemu?
- Sedí změna s persona contractem?
- Nová nebo rozšířená exclusion bez zdůvodnění
- Změna scope: All users, All resources, admin persona
- Odpovídá klasifikace rizika skutečné změně?
- Možné lockout scénáře k prověření
- Výčet toho, co nekontrolovala

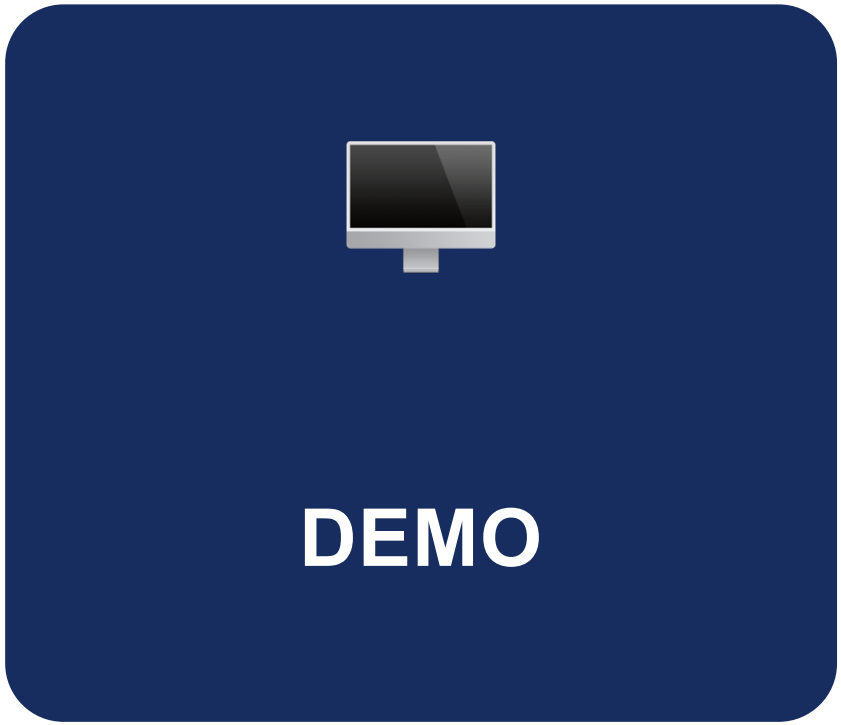
🧠 Člověk – rozhoduje

Kontext, záměr a odpovědnost

- Provozní realita, kterou v datech není vidět
- Posouzení, zda výjimka dává smysl
- Vyhodnocení zjištění od AI
- Konečné schválení nasazení

Člověk, kdo nese odpovědnost

Demo: AI v našem repozitáři



ISACA Czech Republic Chapter

Tři roky provozu

Drift, exception debt a více tenantů

ISACA
Czech Republic Chapter

Projekt v číslech: tým, vznik a provoz

3
členové týmu
O2, ITS a Microsoft v rámci Unified support

1000+
commitů
v hlavní vývojové větvi repozitáře

denně
běžící pipelines
Maester testy a backup celého DSC repozitáře do tier 0 prostředí

Jak to vznikalo

Polovina 2023
první přenesení frameworku do tenantu, společně s ITS

→

Q1/2024
repozitář s Microsoftem, kostra projektu, jednoduché CAP definice a jedna testovací pipeline

→

2024–2025
optimalizace kódu a nasazování, nasazení Maesteru

→

Dnes
sada pipeline: testy, Maester, backup, security scanning

👥 Tým

- Tři lidé: kombinace O2, ITS a Microsoftu
- Spolupráce s Microsoftem probíhá v rámci Unified support
- Nikdo není fulltime developer – vývoj je komplementární k ostatní pracovní náplni
- Náklad je primárně na utilizaci lidských zdrojů, IaC/DSC a testovací frameworky jsou OSS
- Náklad vzniká při častějším běhu pipelines a zapojením AI (Copilot Reviewer)

Framework nevznikl jako jednorázový projekt s rozpočtem a fulltime týmem.
De facto to byl zpočátku projekt s nulovým nákladem na technologie.
Rostl inkrementálně vedle běžné práce, proto je dlouhodobě udržitelný.

ISACA Czech Republic Chapter

Tři roky provozu

- 📈 Persony: ze 3 na 9 typových bezpečnostních tříd
  - Růst nevznikl primárně přidáváním politik, ale zpřesňováním a rozšiřováním tříd identit
- ⚙️ Změny prakticky denně a žádná CAP ručně
  - Vše přes Microsoft365DSC; ruční zásah v portálu příští deployment přepíše
  - Mohou existovat emergency zásahy (dočasná výjimka např. Device Code flow)
- 🧩 Kde má model hranici
  - DSC aktualizuje jen politiky, které řídí – nemaže nic navíc
  - Ručně vytvořená nebo historická politika mimo evidenci proto není automaticky odhalena
  - Reziduální legacy politiky revidujeme a odstraňujeme ručně; cíl je 100 % CAP jen přes DSC
- ♻️ Výjimky nezanikají, vyvíjejí se
  - Mění se technologický základ Entra ID i potřeby firmy – exception debt je průběžná práce, ne jednorázový úklid

Drift vzniká jinde, než se čeká

Ne v politikách řízených kódem, ale:
• v politikách mimo evidenci
• v objektech, na které politiky odkazují – skupiny, lokace, aplikace
• ve výjimkách

🌀 Persona drift

Posture identity se může změnit změnou členství, role nebo typu identity, aniž se změnila jediná CAP.

ISACA Czech Republic Chapter

Více tenantů: sdílený model, různé konfigurace

- 🏢 Role tenantů
  - Firemní tenanty pro IT a telco část, tenanty dceřiných firem a O2 Slovensko
- 🎛️ Řízení je centrální, konfigurace je per tenant
  - Každý tenant má vlastní konfigurační soubory s předpisem politik
  - Vlastníkem politik ve všech tenantech je core tým; žádný lokální tým si je neřídí sám
  - DSC deployment zůstává jediným zdrojem pravdy – nikde již není ruční správa CAP v portálu
- 👥 Jak se lokální týmy zapojují
  - Přístup do repozitářů v Azure DevOps a účast na vývoji, jen v explicitně definovaných částech

✅ Co je společné – páteř frameworku

• Persona model, naming convention a číselné rozsahy
• Pravidla pro exclusions a povinné vyloučení break-glass
• Způsob testování a governance změn

🔀 Co se legitimně liší

• Sada politik – některý tenant má jen podmnožinu
• Persony mohou částečně divergovat podle toho, jaké identity v tenantu existují
• Skupiny, named locations, authentication strengths, aplikace a licence jsou vázané na tenant

Sdílí se model a způsob řízení, ne konkrétní sada politik.
Divergence je řízená a viditelná v kódu – ne výsledek toho, že se někde zapomnělo.

ISACA Czech Republic Chapter

Zavedení IaC: lidé, adopce a náklady

- 👥 **Lidé a adopce**
  - Zero Trust není jednorázový projekt, ale kontinuální proces
  - Zavedení IaC je hlavně změna myšlení administrátorů – trvá roky a nejde tlačit
  - **Začít s Persona ZT Framework třeba jen v UI naklikáním**
  - Začít prací v IDE a napsat třeba jen jednu personu
  - Zapojit ihned AI a přejít na Agentic Engineering

- 💸 **Licence a náklady**
  - Platformy zkoušet na menších projektech; DSC v POC jako nástroj nativně zaměřený na Entra ID a M365
  - ⚠️ Pozor na AI hype a SCU – s Copilot Agents a Agent 365 se licencování znásobilo
  - Vypnutím Copilot for Security přijdete o doporučení agenta → platit aspoň 1 SCU a hlídat spending limits
  - Prioritou jsou licence Entra ID P2; nákladné jsou federační scénáře s Entra External ID

ISACA Czech Republic Chapter

Zavedení IaC: na co myslet od začátku

🧩 Nezávislá architektura

Persony, security contracty, naming, exclusions, break-glass, testy a governance přežijí výměnu enginu.

- Naming convention a číselné rozsahy nesoucí personu
- Exclusions a povinné vyloučení break-glass
- Testovací matice ze security contractu
- Staged enforcement: report-only → pilot → plné vynucení
- Pravidla pro zapojení AI do návrhu i review

Přenáší se model, ne konkrétní implementace.

⚙️ Volba enginu je implementační rozhodnutí

DSC, Graph TCM i Terraform vedou ke stejnému desired state – rozhoduje feature parity a provozní realita.

- Na zelené louce: TCM jako nativní řešení Microsoftu, nebo Terraform, pokud už ve firmě běží Terraform tým
- Microsoft365DSC tam, kde je potřeba širší pokrytí Microsoft 365 než jen Conditional Access
- **Behaviorální testy zůstávají v PowerShellu s Pesterem a What If – nezávisle na zvoleném enginu**

🤖 AI poroste ve vývoji, běhu i kontrolách

Zapojit ji s deterministickými kontrolami a odpovědností člověka. Až budou agenti otevírat pull requesty místo zápisu do tenantu, zapadnou do stejného procesu – framework je na to připravený.

- Zapojit ji ihned do vývoje testů a návrhu implementace CAP
- **Očekávaný výsledek v testu ale schvaluje člověk proti security contractu**
- Postupně zapojit do Code review a jako Review při deploymentu

Q&A

ISACA
Czech Republic Chapter