AD Miner

(aka Bloodhound on steroids)

Offensive & defensive assessment of Active Directory infrastructures

Jean-Michel Besnard

July 2024

About me

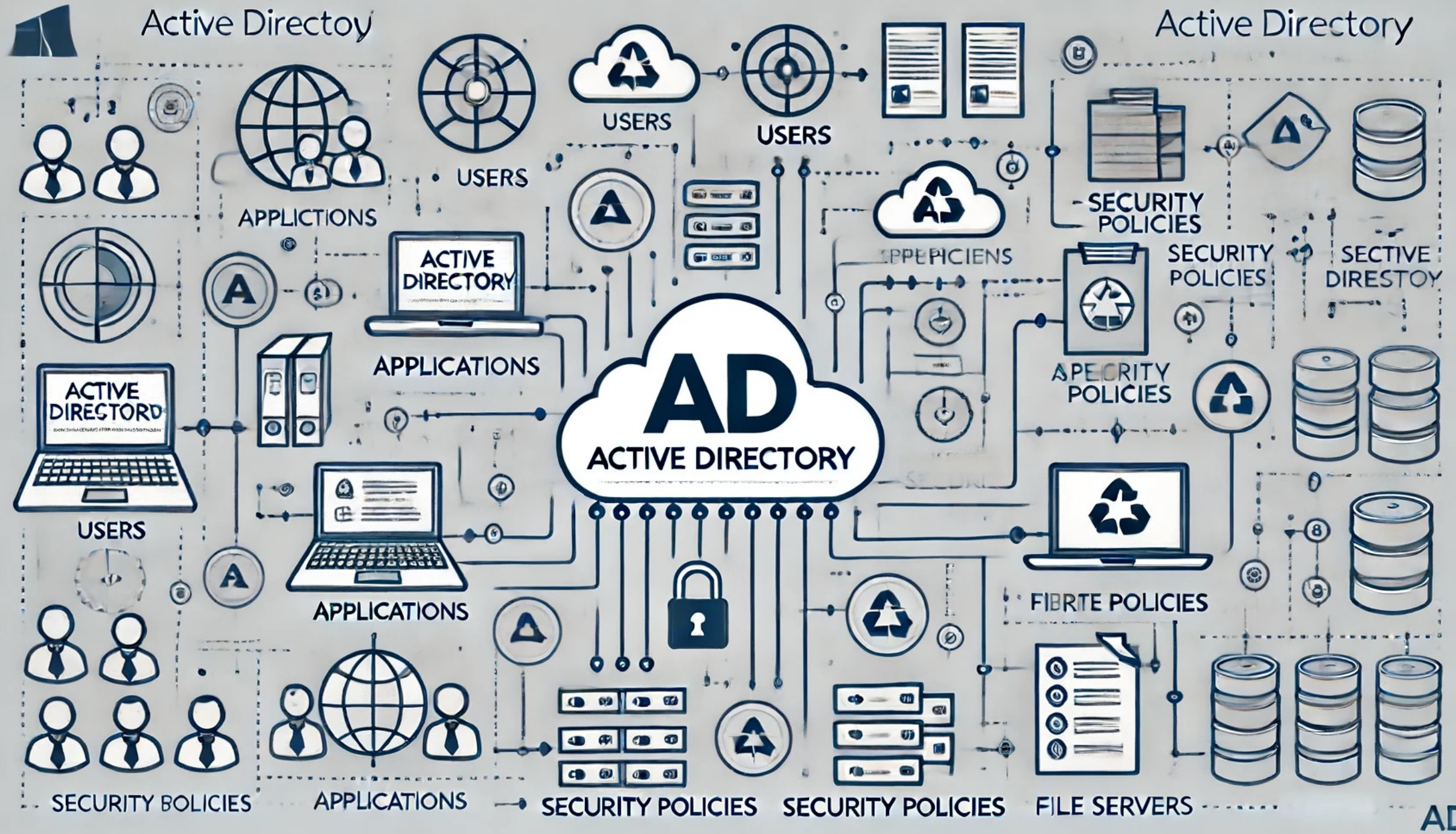
Jean-Michel Besnard

@jmbesnard_maz

Grant Thornton

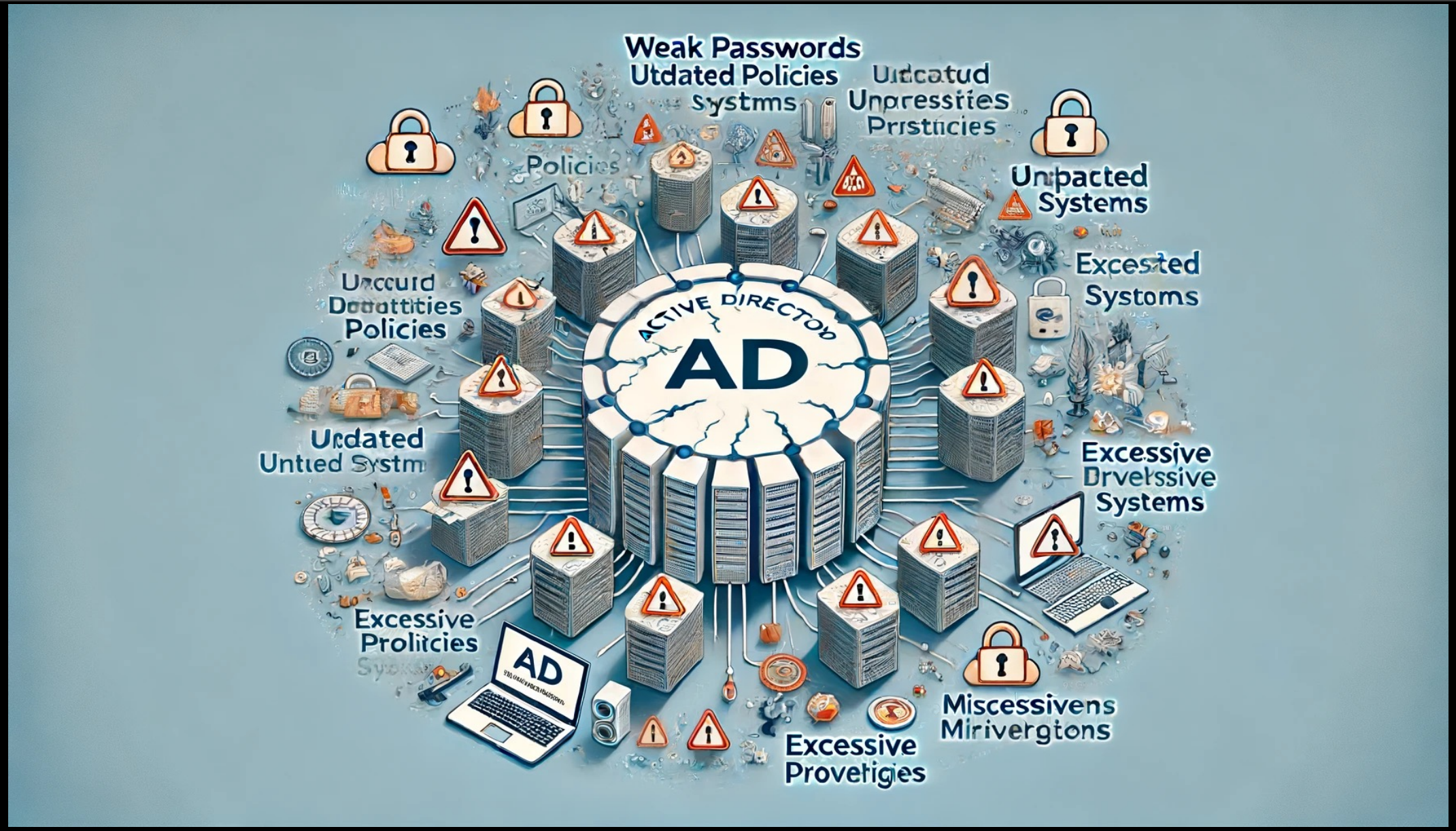
Why securing Active Directory is key

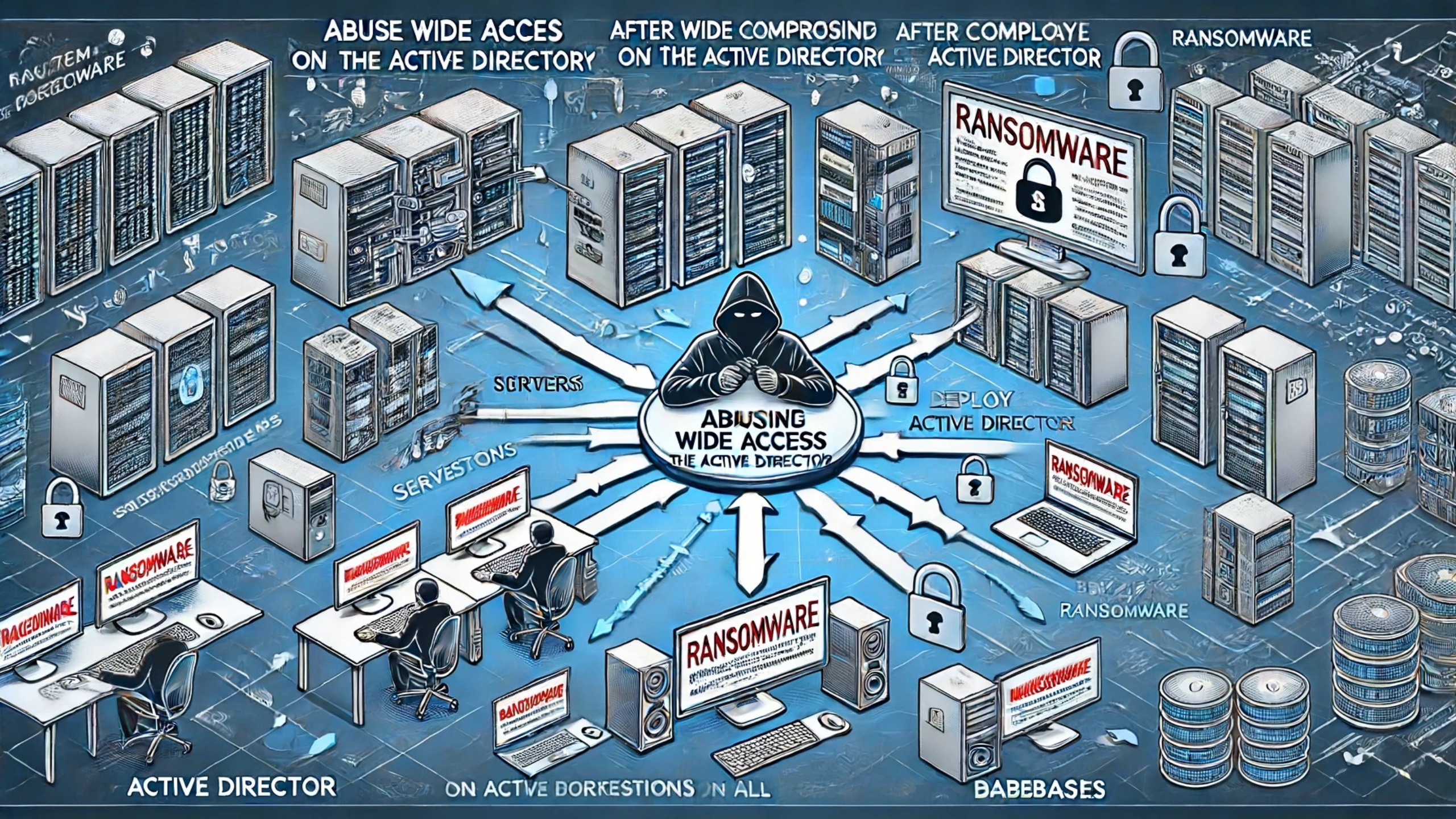
- Active Directory (on-prem & Azure/Entra) is :
  - Central to most IT infrastructures
  - Obviously critical in terms of security


Active Directoy
Active Directory
USERS
USERS
USERS
APPLICTIONS
ACTIVE DIRECTORY
APPLICATIONS
SECURITY POLICIES
SECURITY POLICIES
SECTIVE DIRESTOY
AD
ACTIVE DIRECTORY
ACTIVE DIRECTORD
USERS
APPLICATIONS
APECRITY POLICIES
FIBRTE POLICIES
SECURITY BOLICIES
APPLICATIONS
SECURITY POLICIES
SECURITY POLICIES
FILE SERVERS
AD

What you may think AD looks like

What it actually looks like




ABUSE WIDE ACCES
ON THE ACTIVE DIRECTORY
AFTER WIDE COMPROSIND
ON THE ACTIVE DIRECTOR
AFTER COMPLOAYE
ACTIVE DIRECTOR
RANSOMWARE
RANSOMWARE
SERVERS
ABUSING
WIDE ACCESS
THE ACTIVE DIRECTOR
DEPLOY
ACTIVE DIRECTOR
SERVESTONS
RANSOMWARE
RANSOMWARE
ACTIVE DIRECTOR
ON ACTIVE BORKESTIONS ON ALL
DABEBASES

Why securing Active Directory is key

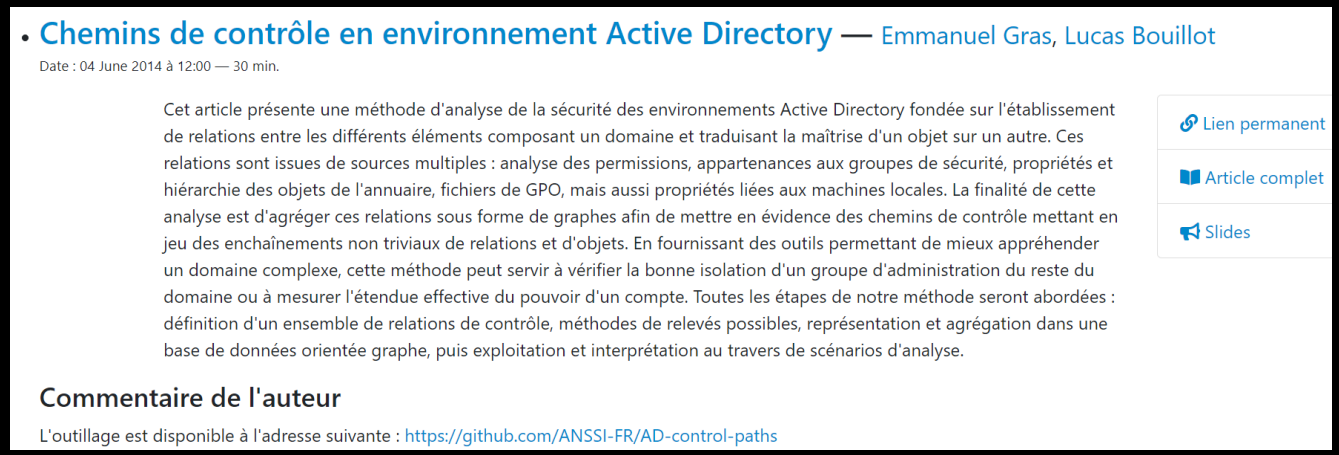
- Active Directory (on-prem & Azure/Entra) is :
  - Central to most IT infrastructures
  - Obviously critical in terms of security
- For short, if you own the AD, you own almost the whole Information System
- Yet it often exposes a LOT of weaknesses (that, often, survived decades of bad practices)
- Therefore, is involved in an estimated 90% of ransomware attacks (*source: Mandiant*)
- Hence, requires :
  - Assessments to mitigate the risks
  - Regular monitoring to spot new issues

Existing tools to assess Active Directory

- Commercial tools : probably good but VERY **expensive**
- ANSSI ADS : closed source
- Free/opensource solutions (not a thorough list):
  - PingCastle : good but mostly for **defensive** side (and almost no graphs)
  - Purple Knight : same same
  - BTA : great but requires NTDS in the first place (and again, no graphs)
  - Bloodhound : excellent but mostly offensive + other **limitations** (listed after)

Graph representation

- 10 years ago, at SSTIC 2014 :
  - First academic paper leveraging the power of graph representation for AD auditing
  - Implementation in a tool name **AD Control Paths** (GH repo is now archived)



• **Chemins de contrôle en environnement Active Directory** — Emmanuel Gras, Lucas Bouillot

Date : 04 June 2014 à 12:00 — 30 min.

Cet article présente une méthode d'analyse de la sécurité des environnements Active Directory fondée sur l'établissement de relations entre les différents éléments composant un domaine et traduisant la maîtrise d'un objet sur un autre. Ces relations sont issues de sources multiples : analyse des permissions, appartenances aux groupes de sécurité, propriétés et hiérarchie des objets de l'annuaire, fichiers de GPO, mais aussi propriétés liées aux machines locales. La finalité de cette analyse est d'agréger ces relations sous forme de graphes afin de mettre en évidence des chemins de contrôle mettant en jeu des enchaînements non triviaux de relations et d'objets. En fournissant des outils permettant de mieux appréhender un domaine complexe, cette méthode peut servir à vérifier la bonne isolation d'un groupe d'administration du reste du domaine ou à mesurer l'étendue effective du pouvoir d'un compte. Toutes les étapes de notre méthode seront abordées : définition d'un ensemble de relations de contrôle, méthodes de relevés possibles, représentation et agrégation dans une base de données orientée graphe, puis exploitation et interprétation au travers de scénarios d'analyse.

Lien permanent

Article complet

Slides

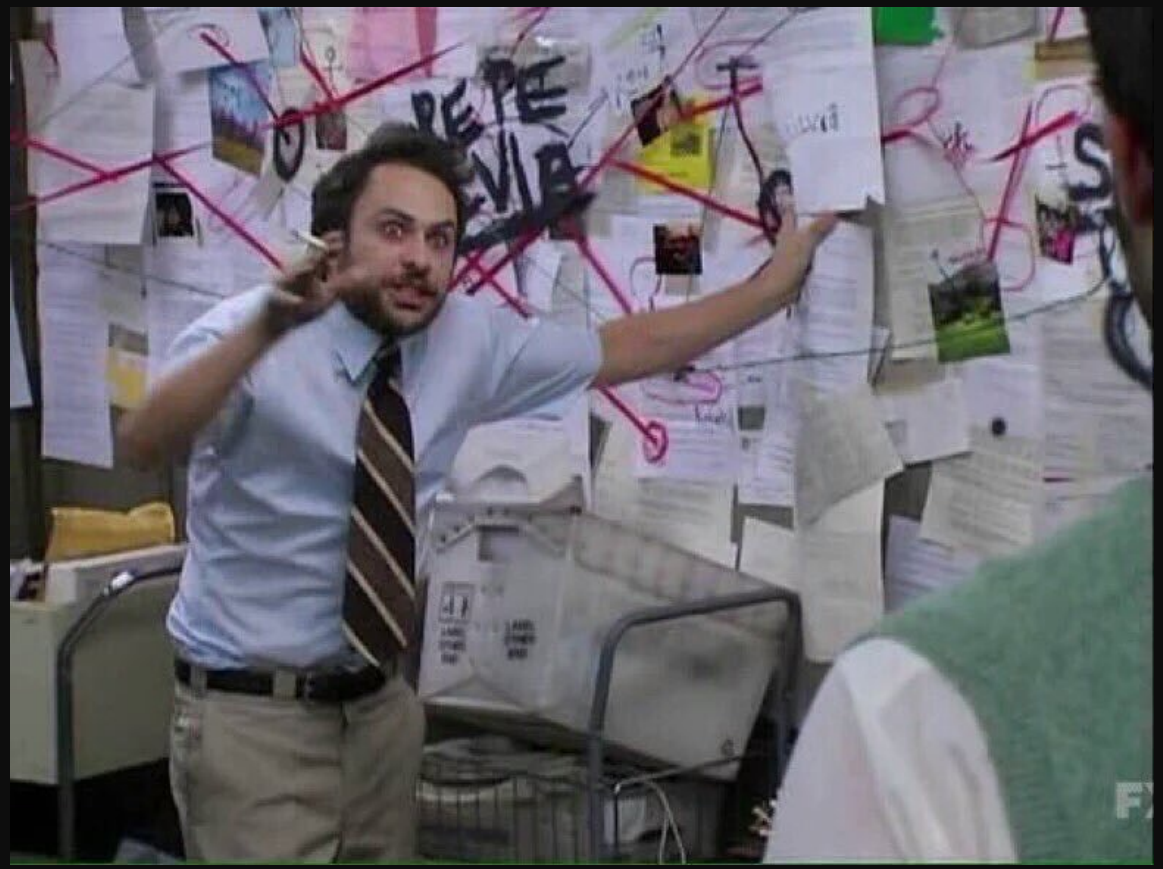
Commentaire de l'auteur

L'outillage est disponible à l'adresse suivante : https://github.com/ANSSI-FR/AD-control-paths

- 2 years later, in Nov. 2016, at BlackHat Arsenal Europe :
  - First release of **Bloodhound** by Andy Robins (@_wald0)
  - Also relies on graph representation
  - Receives a lot of attention and success

Graph representation

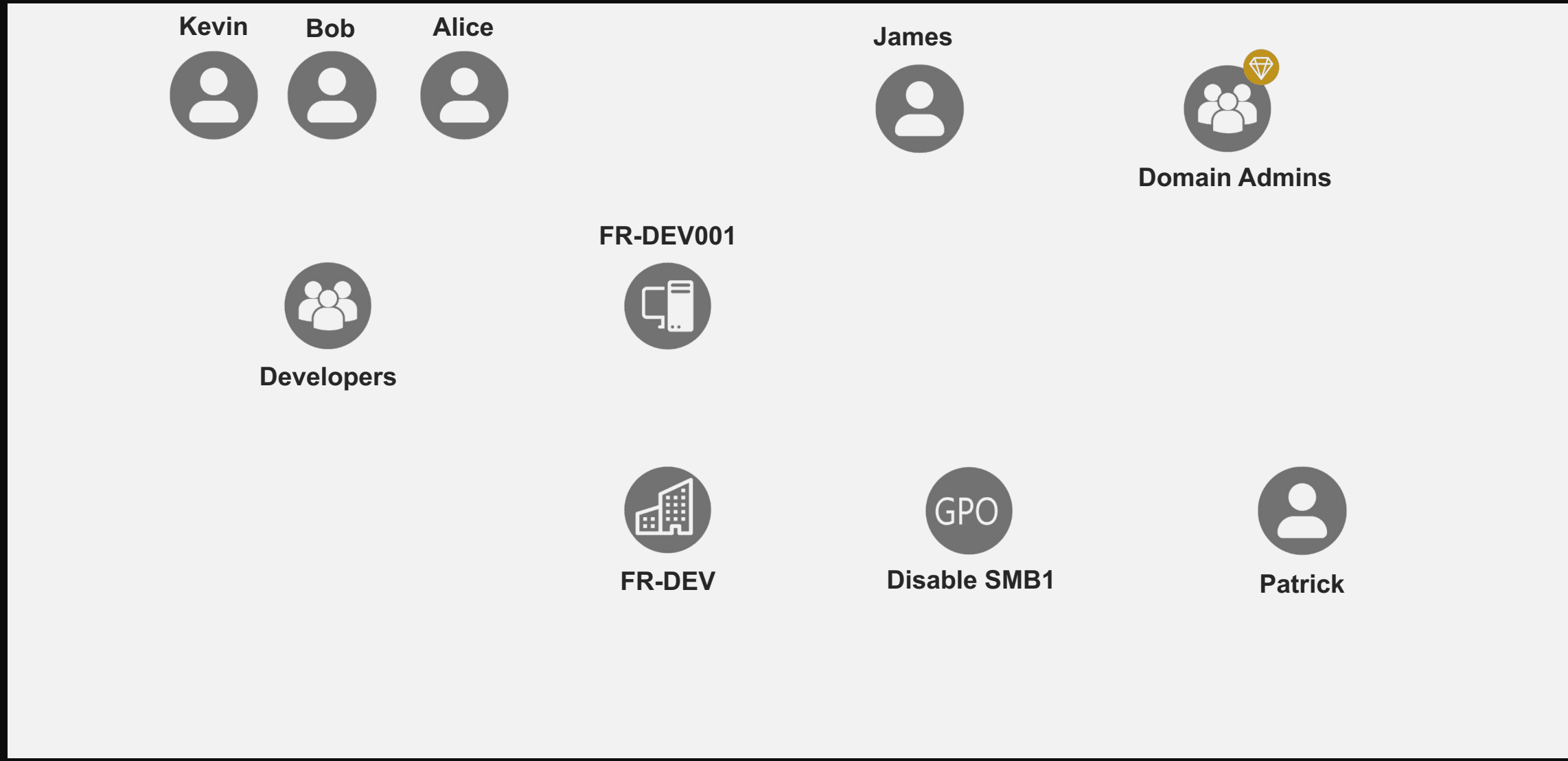
- What is graph representation and how can we use it in the context of AD ?



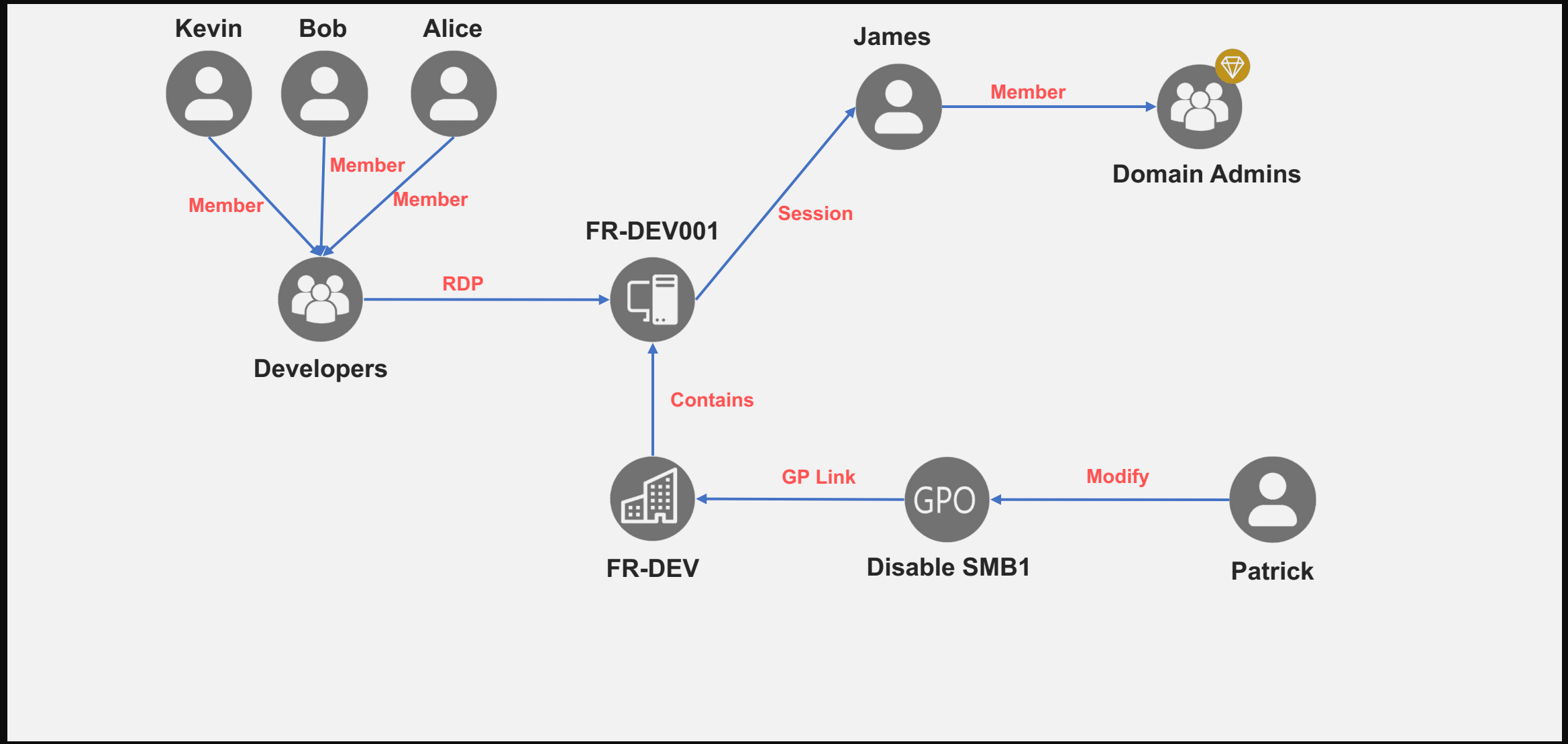
Graph representation

- What is graph representation and how can we use it in the context of AD ?
- Consider the following enumerated information :
  - **Kevin, Bob and Alice** are members of the **AD group “Developers”**
  - Members of that group can RDP to computer **FR-DEV001**
  - **James** has an active session on **computer FR-DEV001**
  - **James** is member of **Domain Admins**
  - **Patrick** can modify a **GPO named “Disable SMB1”**
  - That GPO is linked to **OU “FR-DEV”**
  - **OU “FR-DEV”** contains **computer FR-DEV001**

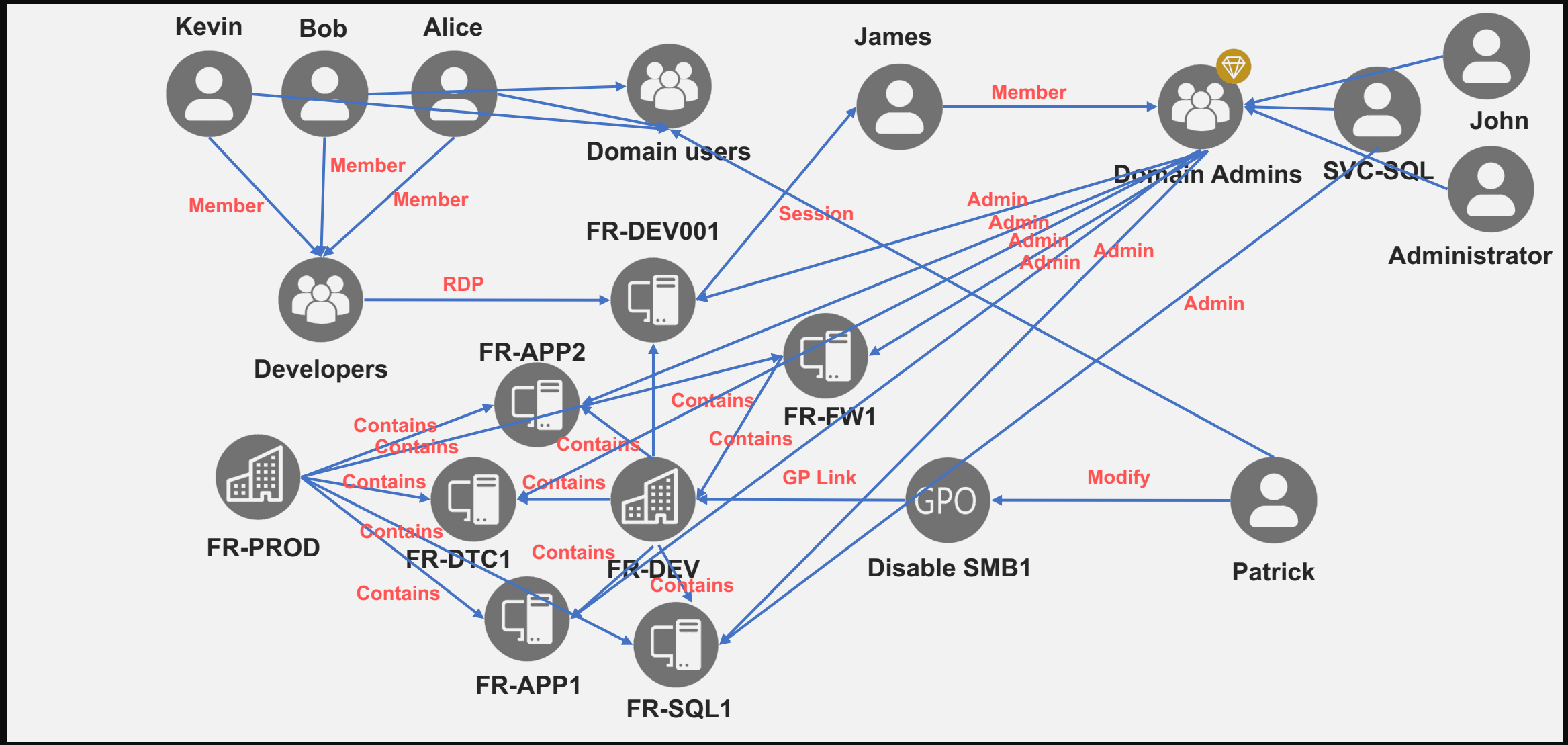
Graph representation



Graph representation



Graph representation

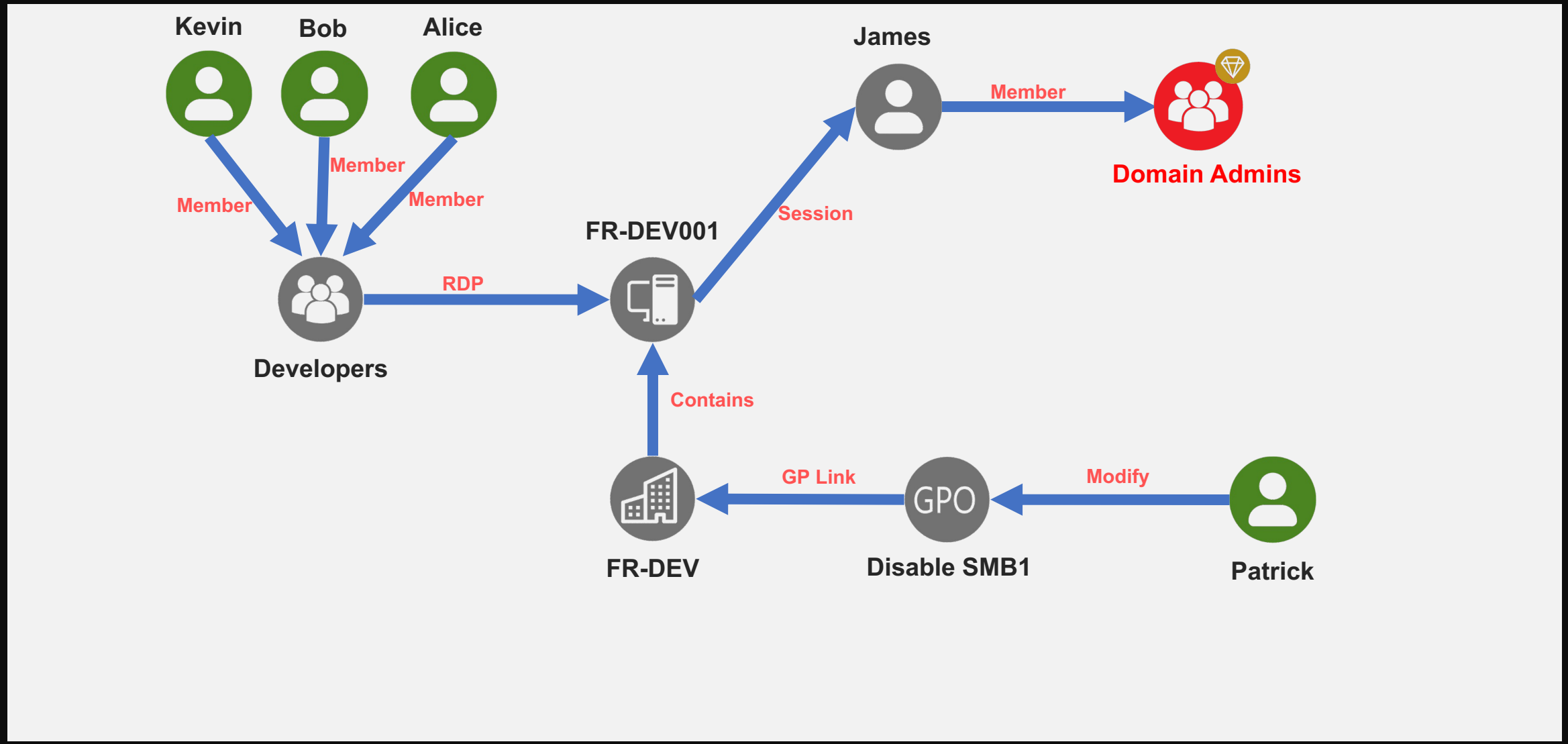


Graph representation

- These nodes and relations can be created in a graph database (ex: Neo4j)
- Then, a simple cypher query can reveal all control paths

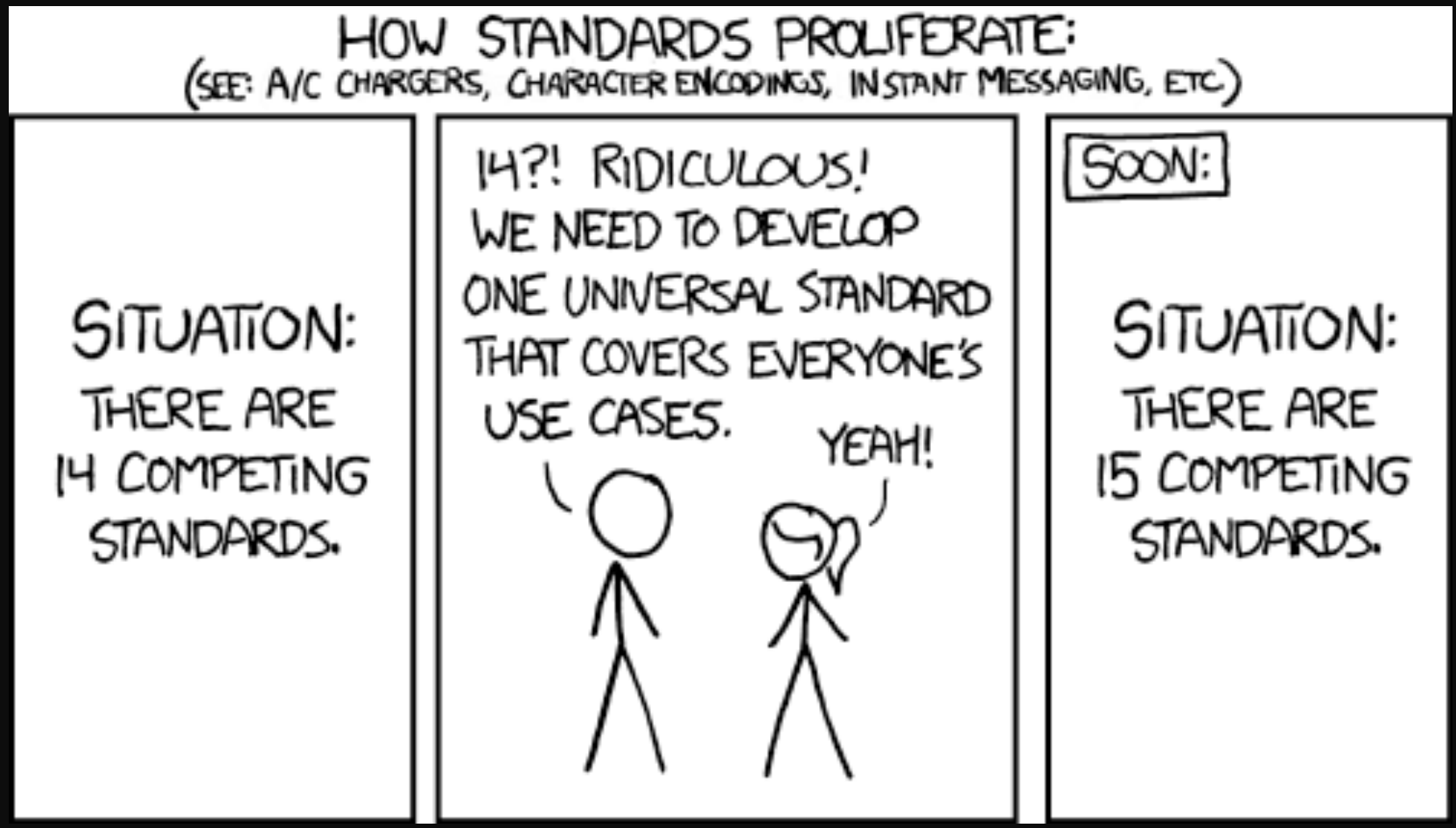
```
MATCH paths=(n)-[Member|Session|Admin|GP Link|Contains|Modify*1..]->(m{name:”Domain Admins”})
RETURN paths
```

Graph representation



A new tool to rule them all

- Why developing yet another tool ?



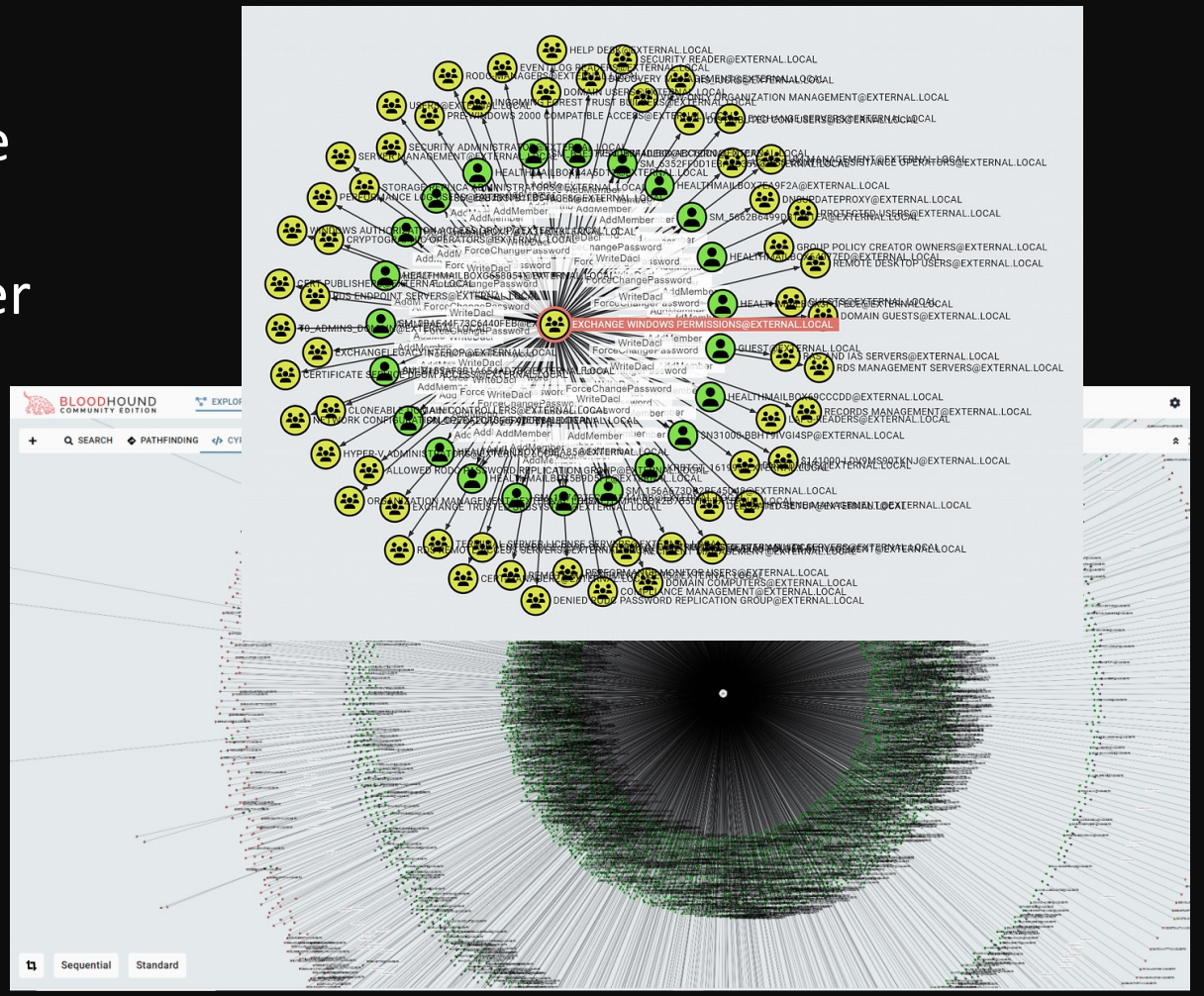
A new tool to rule them all

- Graph representation has a lot a potential
- As-is, Bloodhound is great but :
  - does not make the most out of its potential
  - can only plot **graphs**
  - is missing out on things where lists are better suited
  - can be difficult to use (unless you get your hands deep into cypher queries to fine-tune results)
  - Not really fit for defense/control activities

> "Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win."
>
> - John Lambert, General Manager, Microsoft Threat Intelligence Center

A new tool to rule them all

- As-is, Bloodhound is great but (continued…):
  - is not ideal for non/less-tech savvy people
  - includes some controls (i.e., default cypher queries) but those are
    - rather limited
    - very generic
    - may show all problems at once
      - **very heavy graphs** that may be difficult to navigate
  - requires to run queries for each control
    - Can take a **very long time**



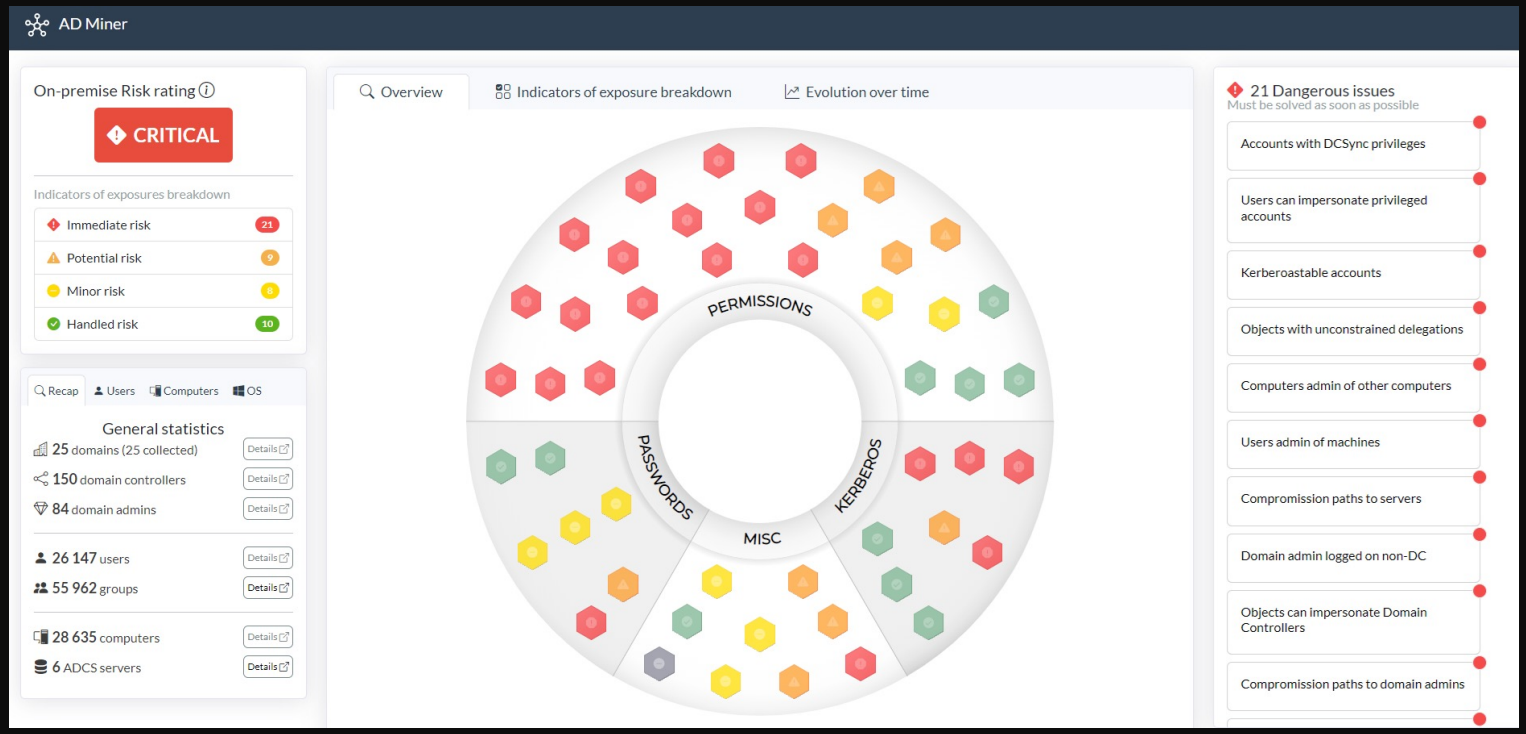
A new tool to rule them all

- What we wanted
  - Leverage the great aspects of Bloodhound (and Sharphound for data collection)
  - Run a constant series of fine-tuned controls **once and for all** (as of now 60+ controls)
  - Cover all controls already available in other tools (PingCastle, ADS, etc.)
  - Make something useful for:
    - **pentesters** who audit ADs and need to document/demo weaknesses
    - **defenders** who mitigate risks (KPI, ratings, evolution over time)
  - A good-looking, dynamic, web-based report that can be accessed offline and without a web server/database (aka static HTML)

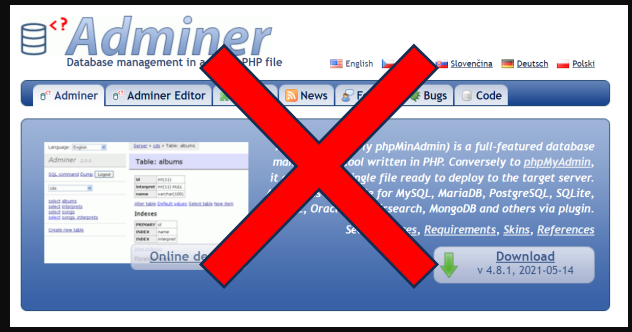
A new tool to rule them all

AD Miner

- aka Bloodhound on steroids



- Yes, we did not make the smartest choice when naming the tool
  - And have learned it the hard way

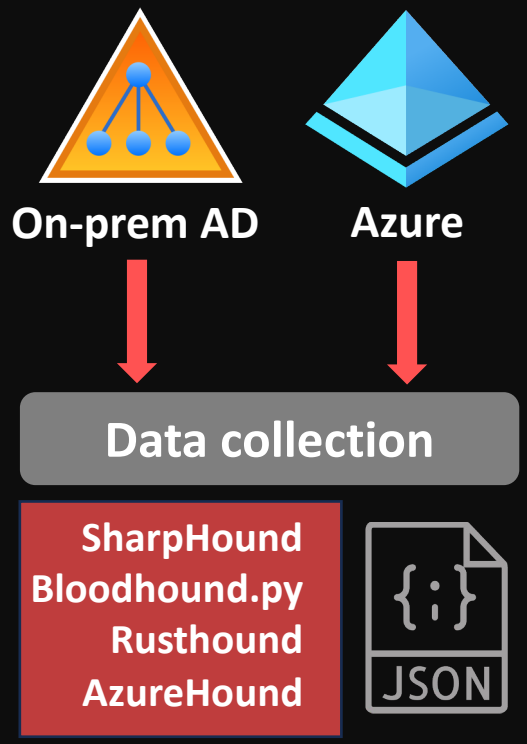


Architecture

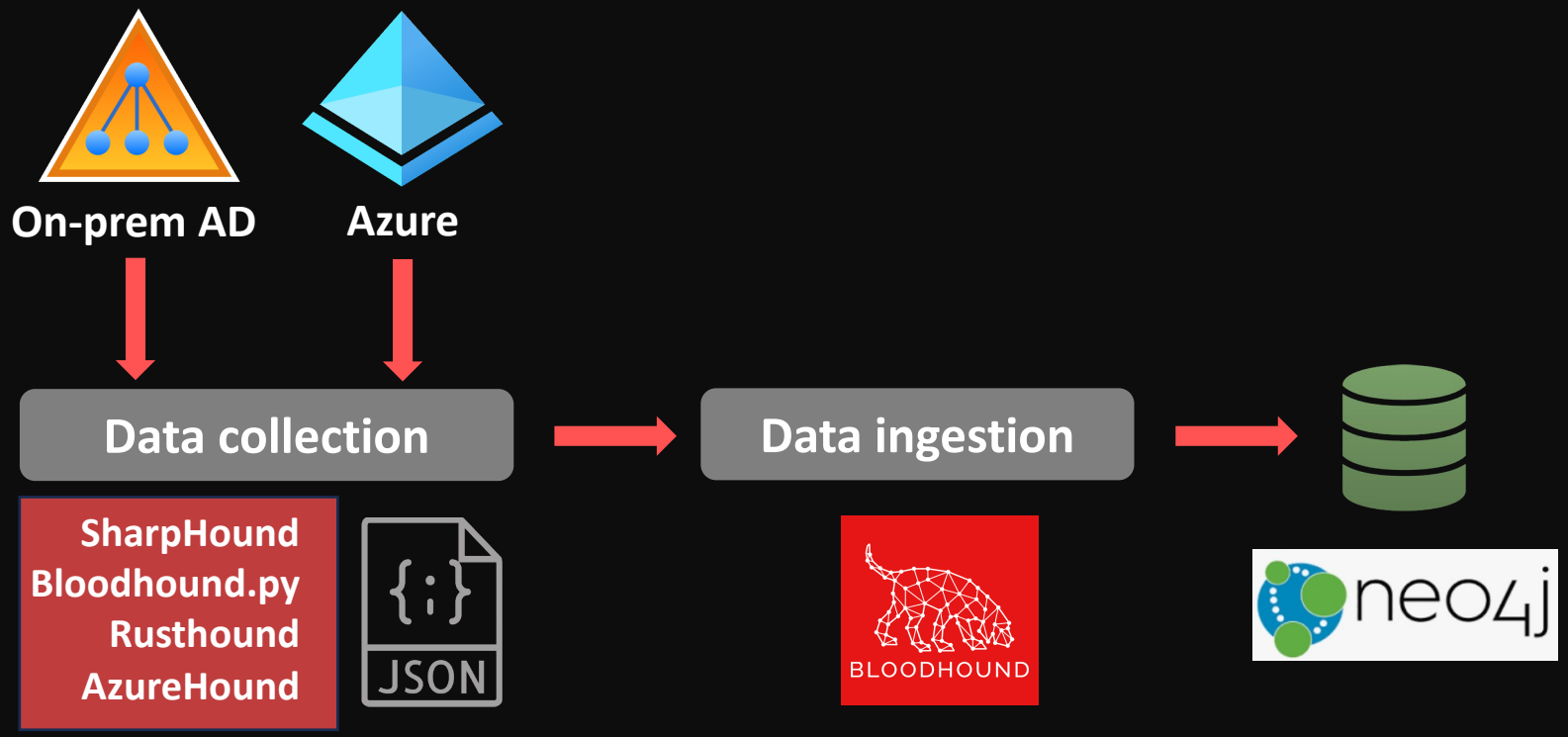
On-prem AD

Azure

Architecture



Architecture



Architecture

On-prem AD

Azure

Data collection

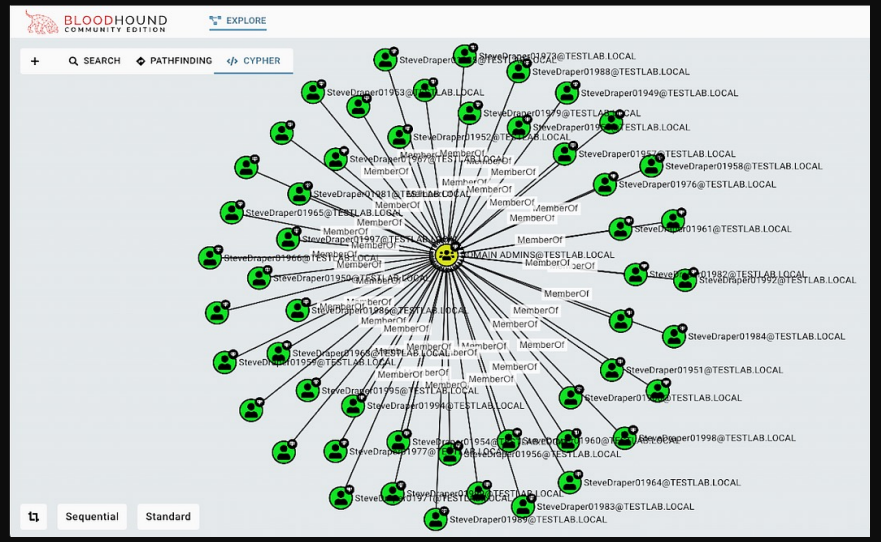
SharpHound
Bloodhound.py
Rusthound
AzureHound

JSON

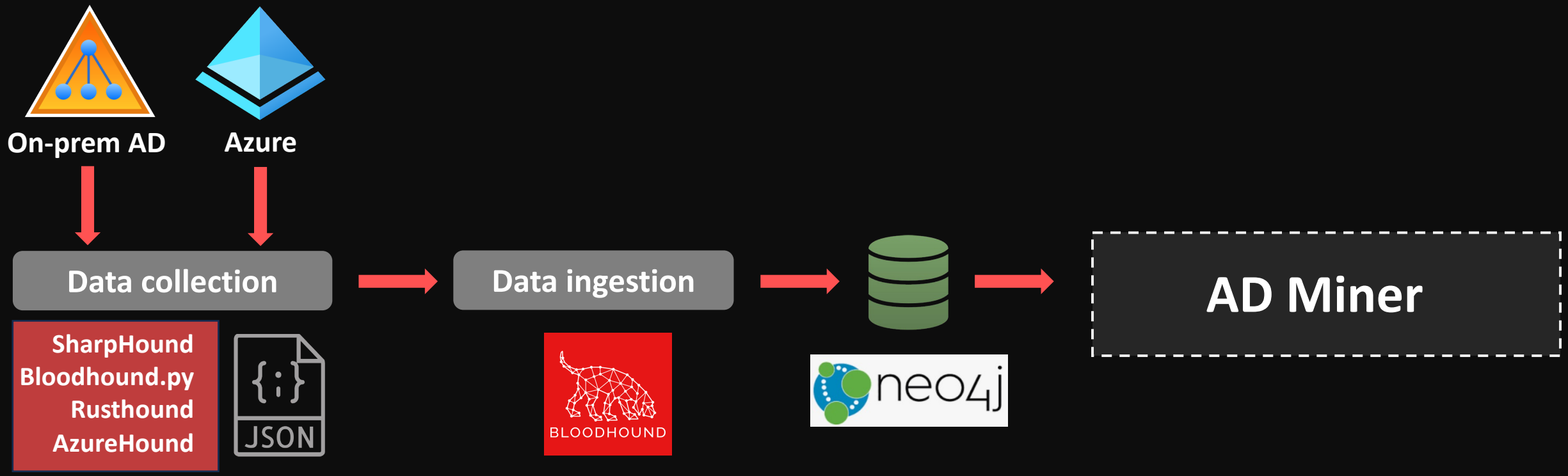
Data ingestion

BLOODHOUND

neo4j



Architecture



Architecture

AD Miner

python

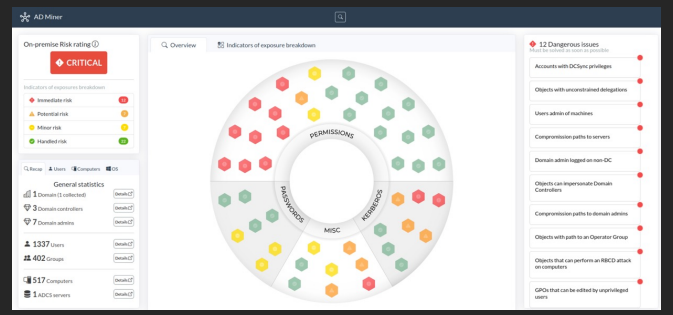
150+ queries

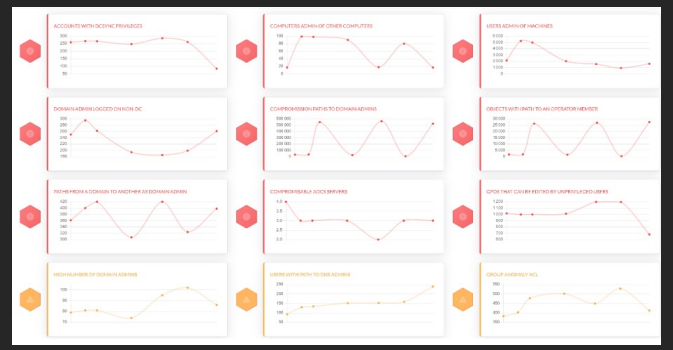
- → Cleaning
- → Custom edges creation
- → Custom nodes labelling
- → Compromission paths
- → Dangerous configuration

python

Data analytics

HTML report



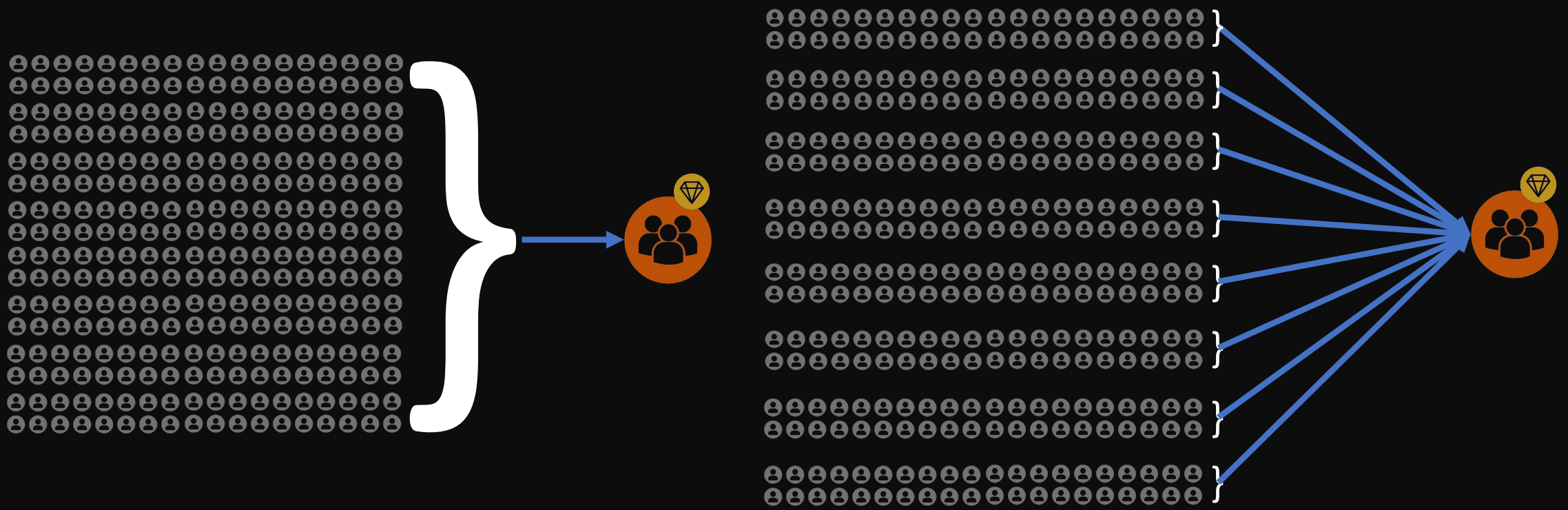


Innovations

- Graph database multi-threading
  - Cypher queries can be very CPU-intensive and take a long time to execute
  - Neo4j Community Edition uses only 1 CPU core per client request

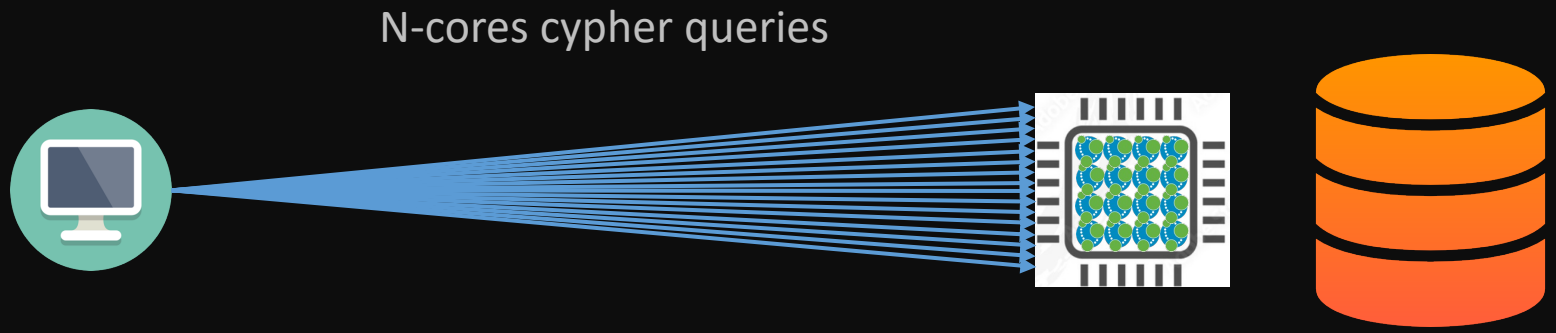
Innovations

- Graph database multi-threading
  - To get around this limitation, we have developed an easy solution:
    - Split cypher into smaller chunks (e.g., split source nodes space into 1000 chunks)



Innovations

- Graph database multi-threading
  - To get around this limitation, we have developed an easy solution:
    - Split cypher into smaller chunks (e.g., split source nodes space into 1000 chunks)
    - Run each chunk in N concurrents client queries

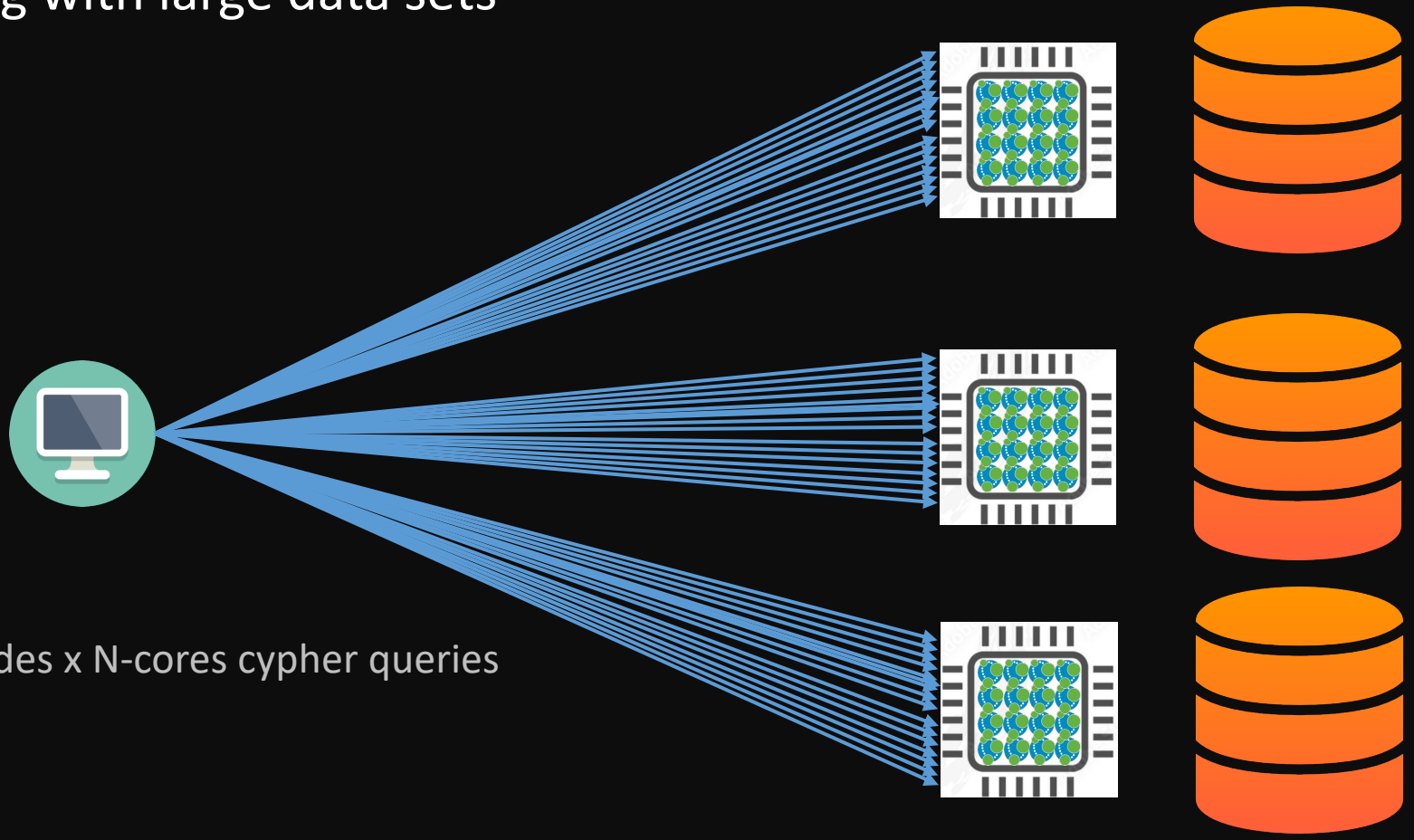


AD forest with 25K users on a 32-core CPU (e.g., Intel Core i9 13900)

- ~ 45 hours without multi-threading
- ~ 2 hours with multi-threading

Innovations

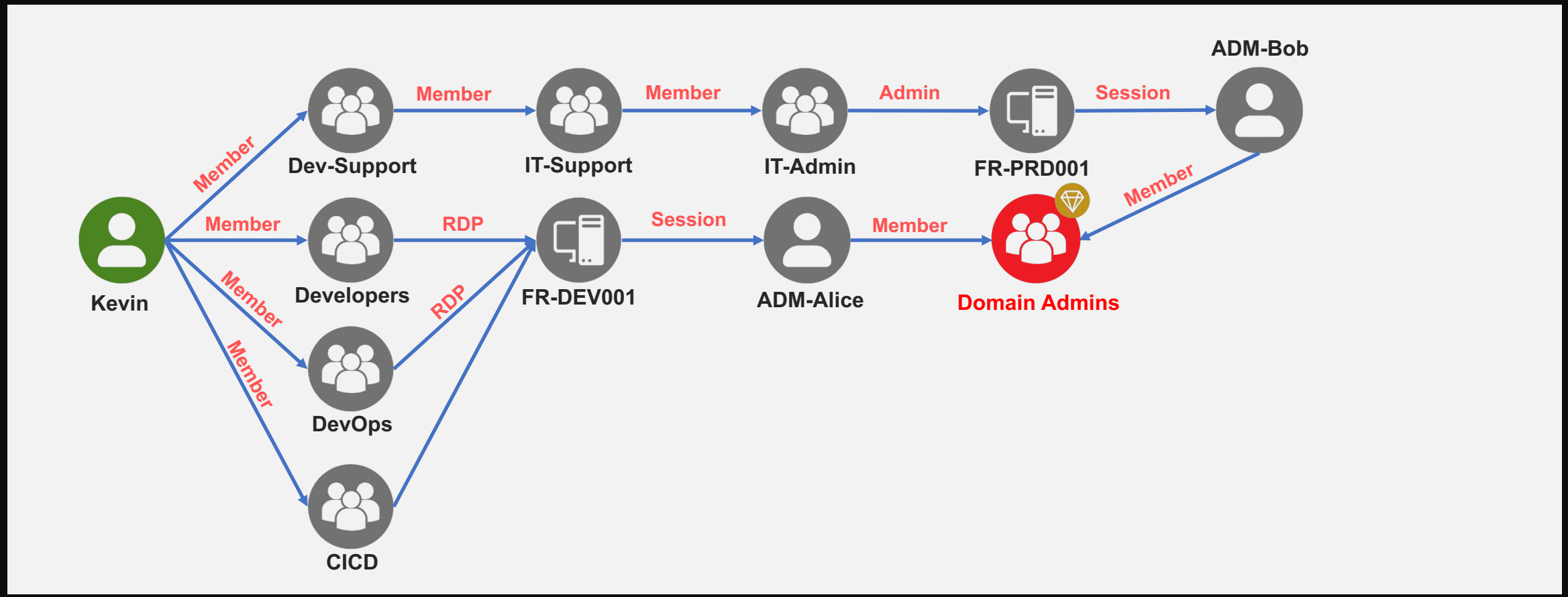
- Graph database multi-threading
  - And while we are at it, implement clustering to further maximize throughput when dealing with large data sets



N-cluster nodes x N-cores cypher queries

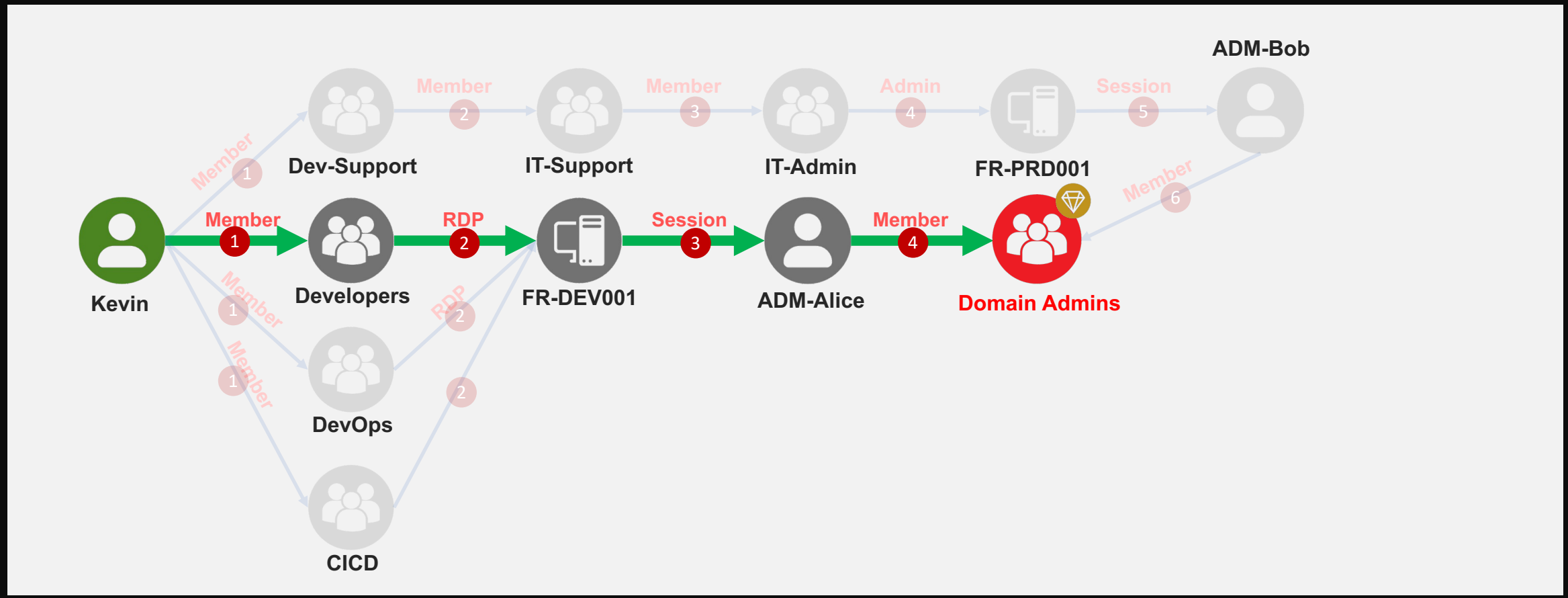
Innovations

- Path computation functions



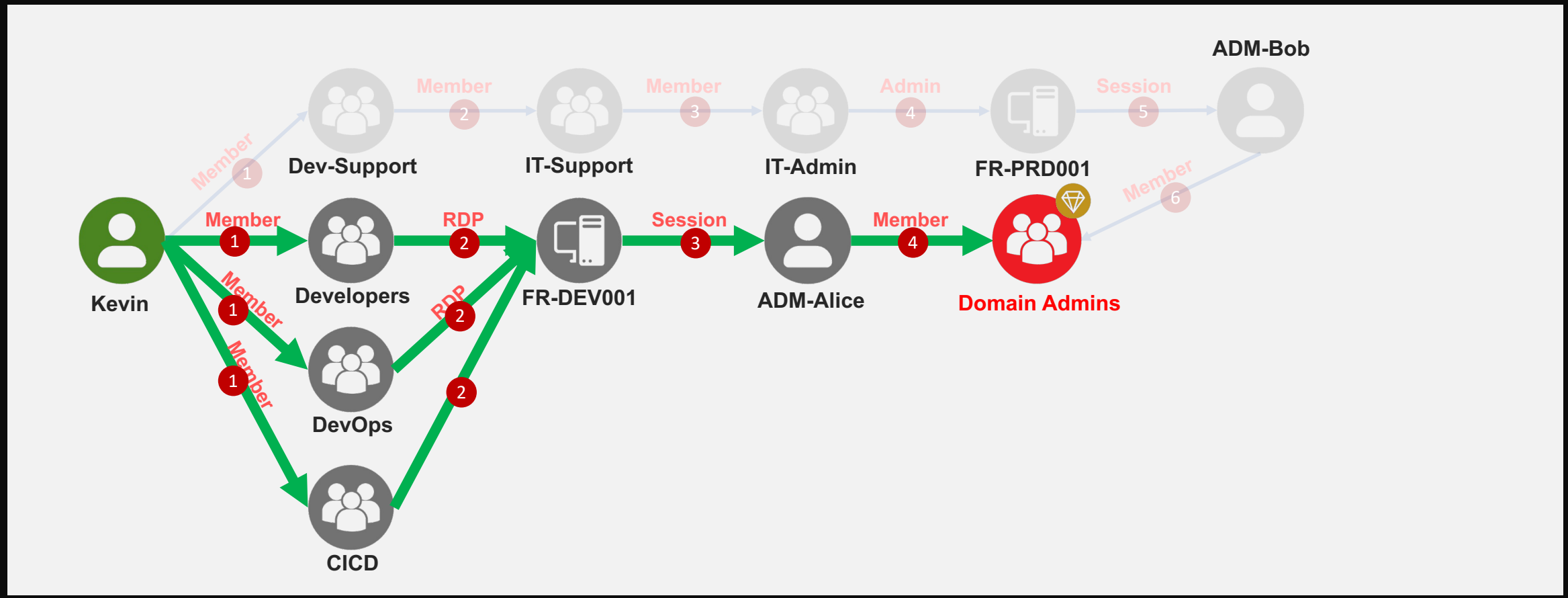
Innovations

- Path computation functions: **shortestPath()**
  - Return one of the paths with least hops



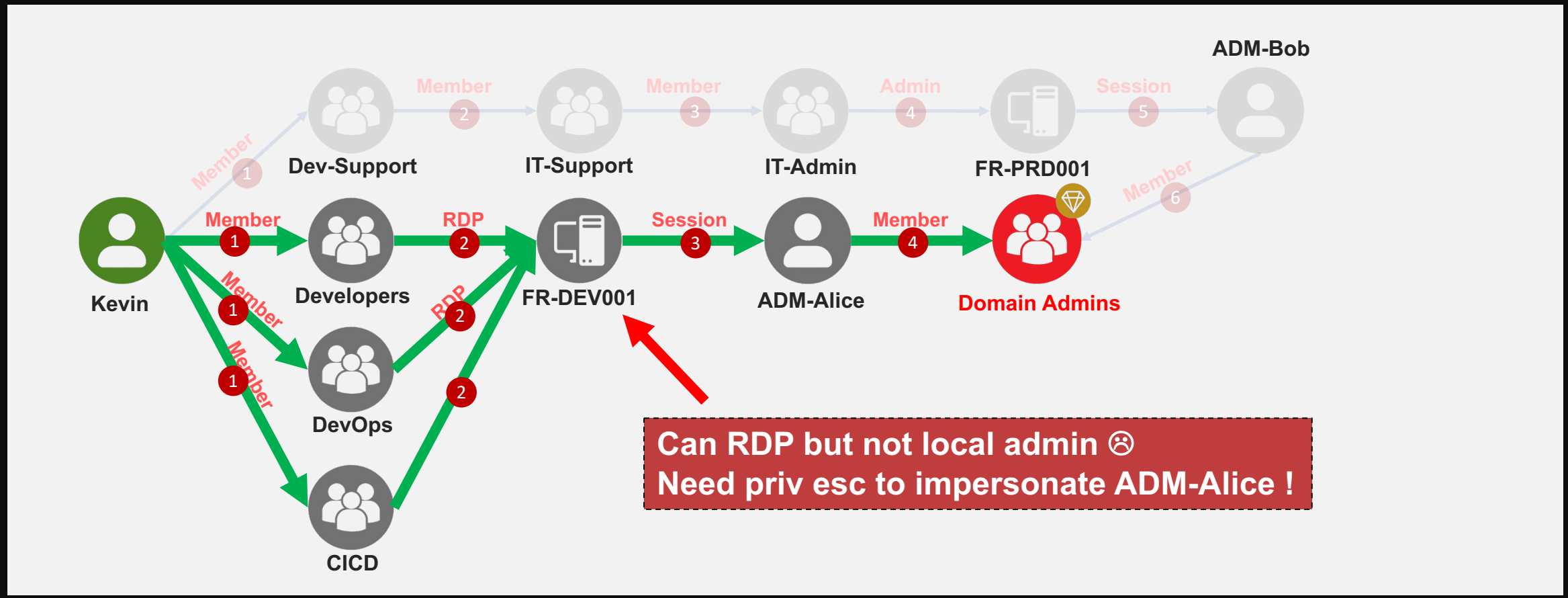
Innovations

- Path computation functions: **allShortestPaths()**
  - Returns all paths of length equal to shortest path



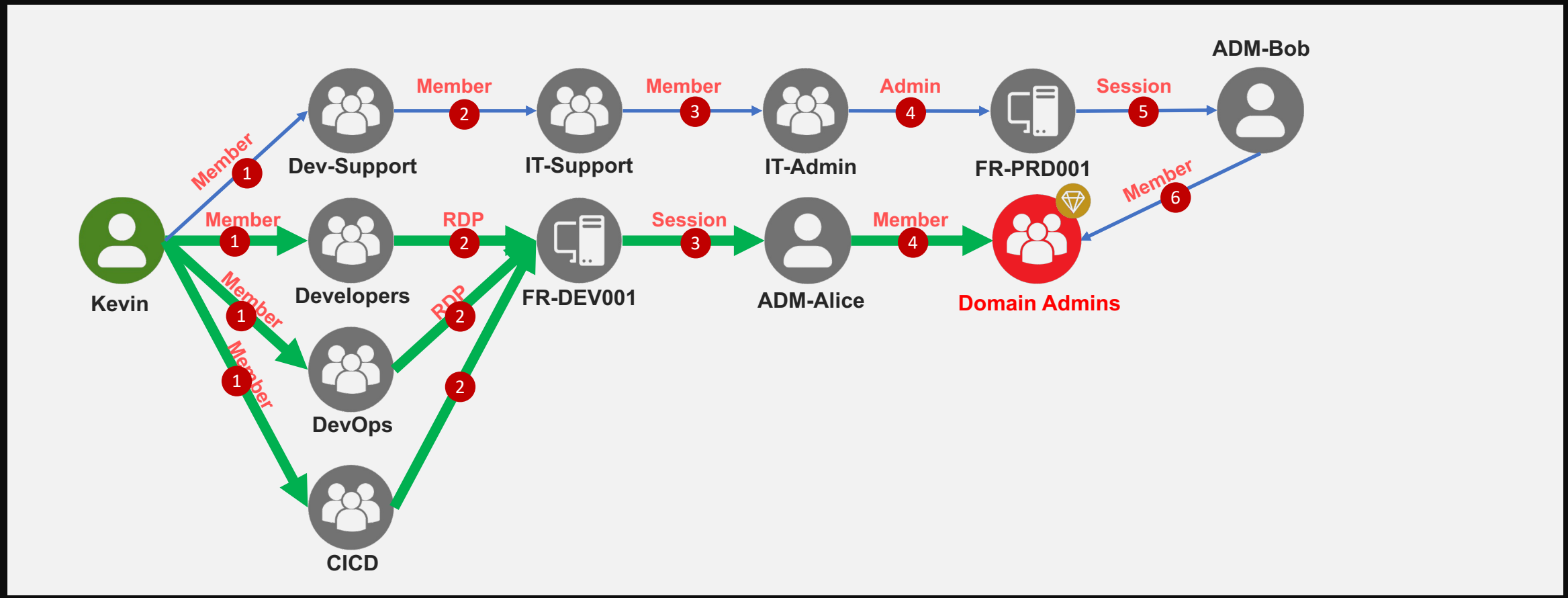
Innovations

- Path computation functions: **allShortestPaths()**
  - Returns all paths of length equal to shortest path



Innovations

- Path computation functions: allShortestPaths()
  - Returns all paths of length equal to shortest path



Innovations

- How to spot paths that are **easier to exploit** (even though these may cross more hops) ?

- November 8th, 2019 blog post by Riccardo Ancarani on using weighted relations:
  - https://riccardoancarani.github.io/2019-11-08-not-all-paths-are-equal/
- Also illustrated here :
  - https://www.linkedin.com/pulse/graph-theory-assess-active-directory-smartest-vs-shortest-besnard-0qgle/

A new tool to rule them all

- Apply weights to relations (values can be modified in AD Miner config file). For example,
  - **MemberOf = 0** | **CanRDP = 40** | **AdminTo = 10** | **etc…**
- Install/Load Graph Data Science (GDS) plugin for Neo4j (*)
- Create a graph projection

```
CALL gds.graph.project.cypher('graph_objects_to_domain_admin', 'MATCH (n)
RETURN id(n) AS id', 'MATCH (n)-[r:$properties$]->(m) RETURN id(m) as
source, id(n) AS target, r.cost as cost', {validateRelationships: false})
```

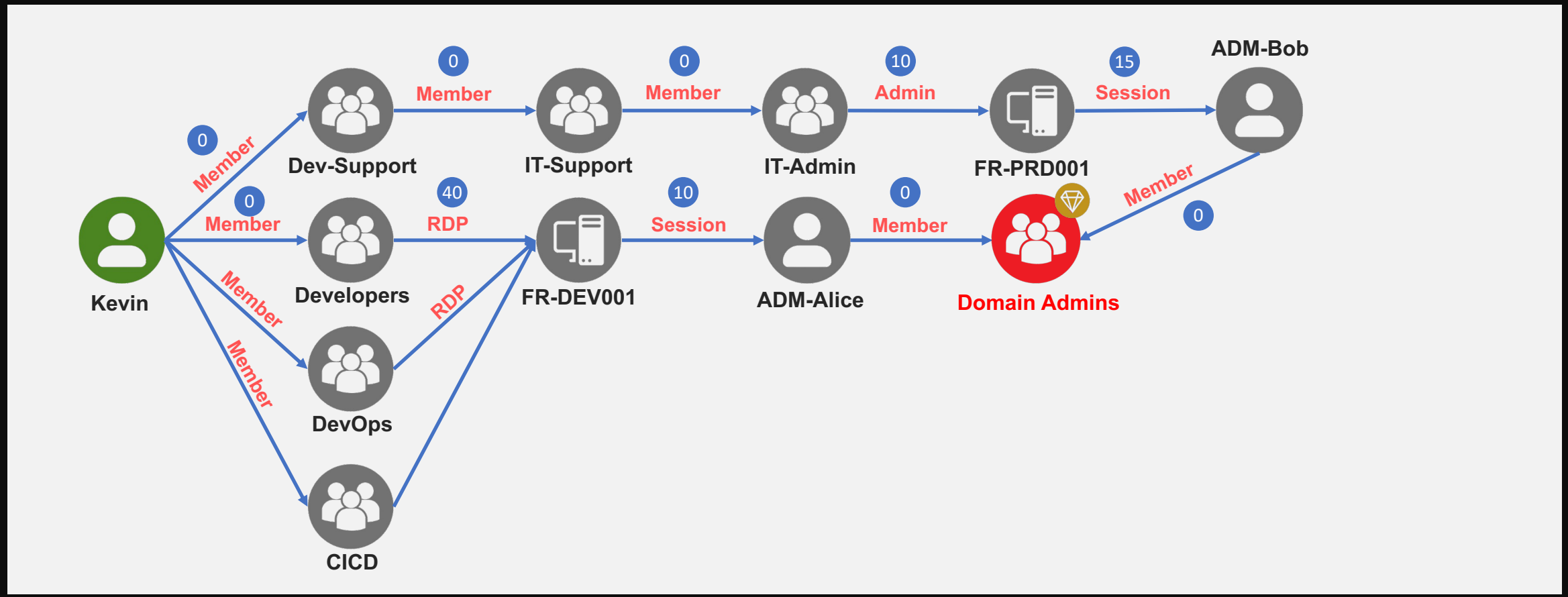
- Use **Dijkstra algorithm** to query path with lower cost

```
MATCH (target:User{name:"Admin-jane@DOM"}) CALL
gds.allShortestPaths.dijkstra.stream('graph_objects_to_domain_admin',
{sourceNode: target, relationshipWeightProperty: 'cost', logProgress:
false}) YIELD path WITH nodes(path)[-1] AS starting_node, path WHERE
starting_node.name = "Joe@DOM"
RETURN path as p
```

(*) Installed by default if you create your Bloodhound environment with github.com/Tanguy-Boisset/bloodhound-automation

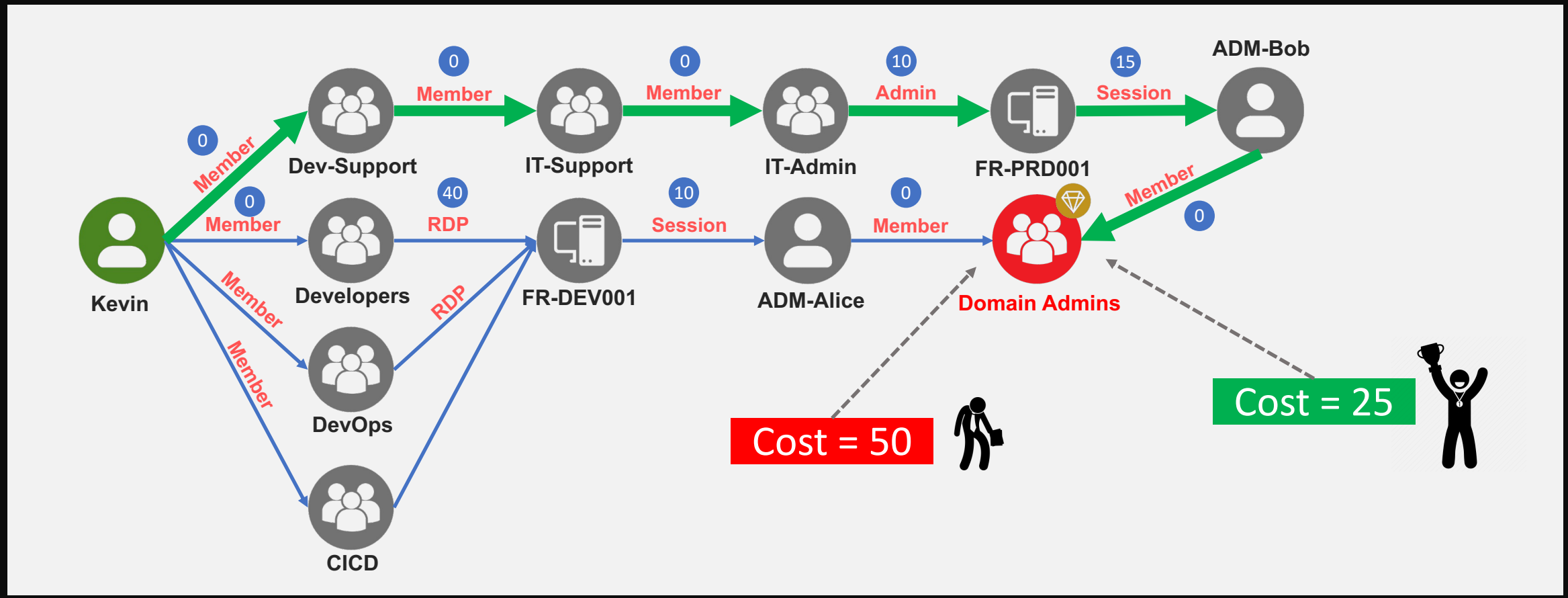
Innovations

- Smartest path (more hops but way easier to exploit)



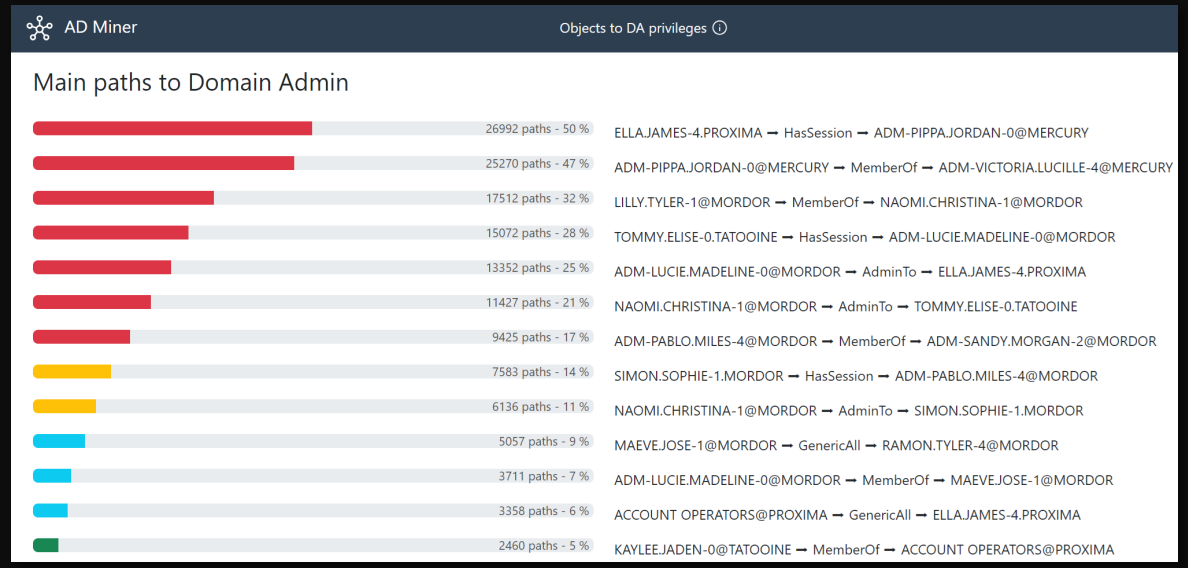
Innovations

- Smartest path (more hops but way easier to exploit)



Things that BH can not do out of the box

- Computing choke-points:
  - i.e., issues that are top contributors to attack paths
  - Fairly easy to do with basic data analytics
  - Shows mitigation quick wins



- Evolution over multiple extractions

Things that BH can not do out of the box

- Show deviant objects within comprehensible lists

AD Miner — Path to OU Handlers

OU Name	Inbound Graph	Inbound List	Targets Interest	Outbound List	Outbound Graph
ŚWIDWIN@PERSEUS.COM	42 paths	42 objects	★★★	1 object	1 path
ĮHAJJAH@MERCURY.COM	42 paths	42 objects	★☆☆	6 objects	6 paths
ĮHALABJAH@PERSEUS.COM	42 paths	42 objects	★☆☆	3 objects	3 paths
ÜBERHERRN@SPACEROCK.COM	44 paths	44 objects	★★☆	4 objects	4 paths
İSLAHIYE@SPACEROCK.COM	42 paths	42 objects	★☆☆	7 objects	7 paths
ZUSHI@PERSEUS.COM	42 paths	42 objects	★☆☆	6 objects	6 paths
ZHUFENG@AURORA.COM	42 paths	42 objects	★☆☆	170 objects	170 paths
ZHUOZHOU@MERCURY.COM	42 paths	42 objects	★☆☆	225 objects	225 paths

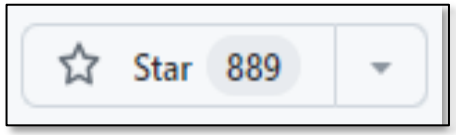
AD Miner — Users that are administrator of computers

User	Kerberoastable	Last Password Change	List Of Computers	Path To Computers	Path To DA
JASON.TOBY@SPACEROCK.COM	-	1 year, 1 month and 16 days	154 computers	path to 154 computers	28 paths to DA (4 domains)
MAXIM.TOBY@SPACEROCK.COM	YES	8 years, 6 months and 16 days	155 computers	path to 155 computers	4 paths to DA (4 domains)
ADM-ISAIAH.RACHEL@SATURN	-	1 month and 29 days	416 computers	path to 416 computers	20 paths to DA (4 domains)
ADM-LANA.IVY@SPACEROCK.COM	-	1 month and 3 days	155 computers	path to 155 computers	-
ADM-JULIAN.MIGUEL@SPACEROCK.COM	-	29 days	159 computers	path to 159 computers	-

Demo

Links

github.com/Mazars-Tech/AD_Miner/




```
> $  pipx install 'git+https://github.com/Mazars-Tech/AD_Miner.git'
```

Q&A

Questions ?