Chemins de contrôle en environnement Active Directory

Chacun son root, chacun son chemin

Lucas Bouillot, Emmanuel Gras

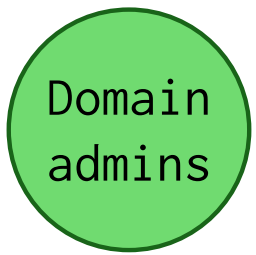
Agence **N**ationale de la
Sécurité des **S**ystèmes
d'**I**nformation

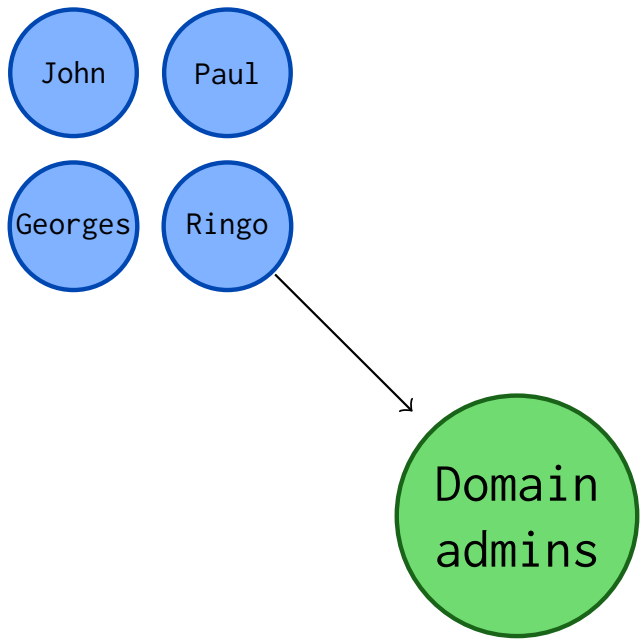
SSTIC 2014 - 4 juin 2014

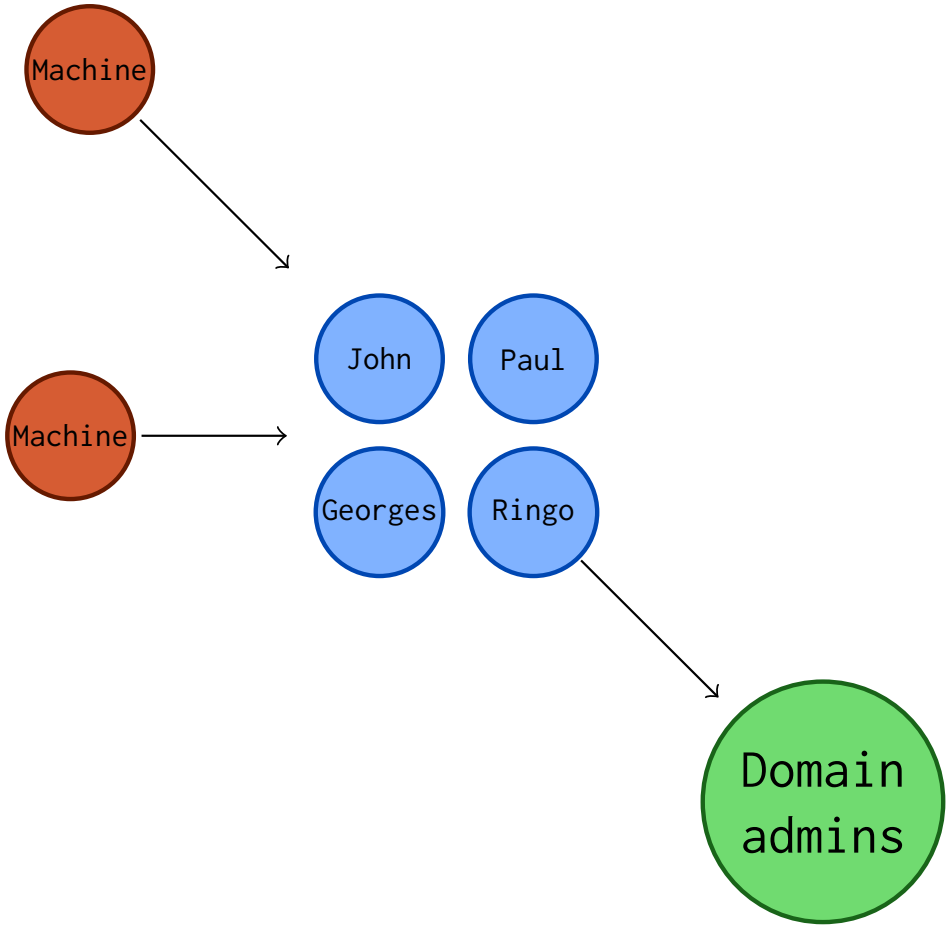
Section 1

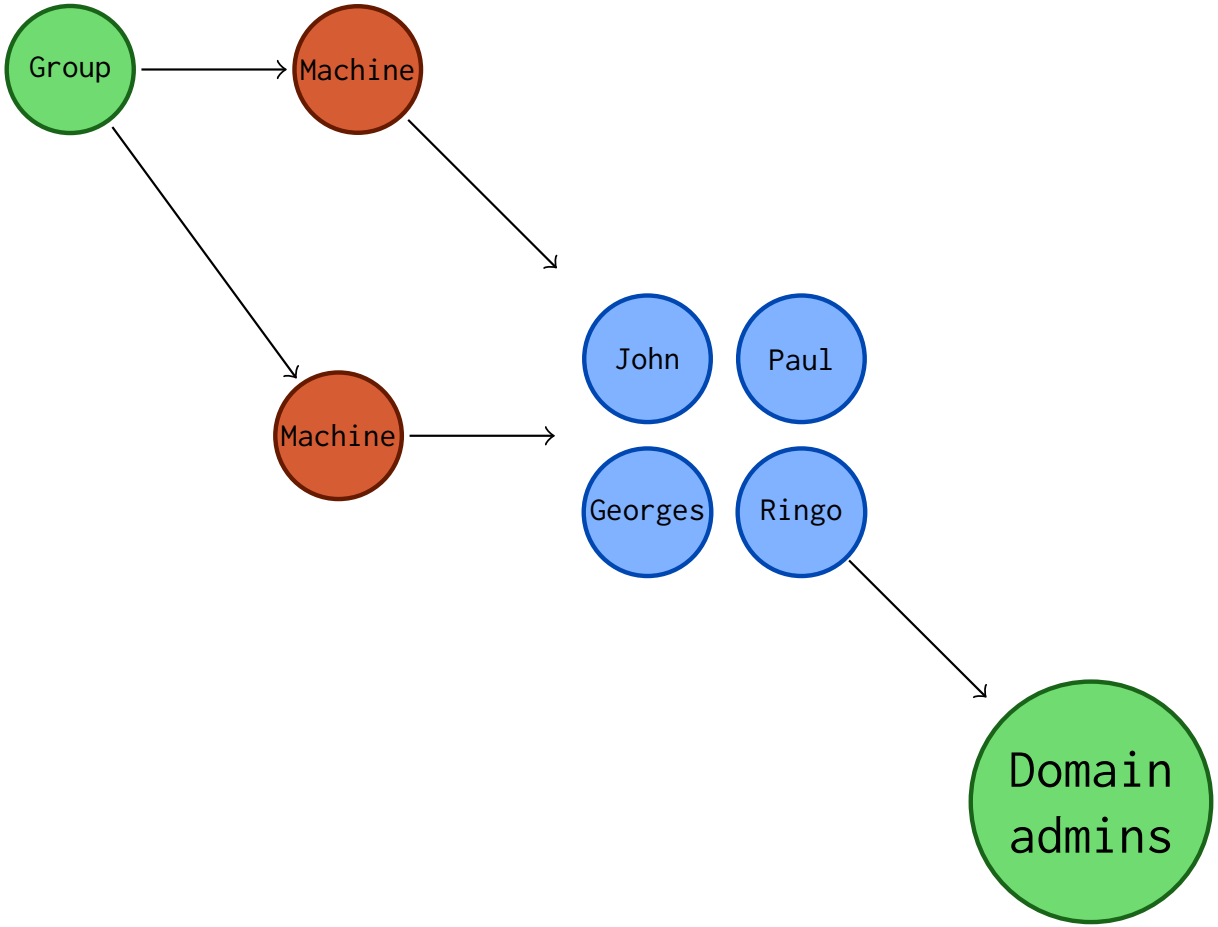
Introduction

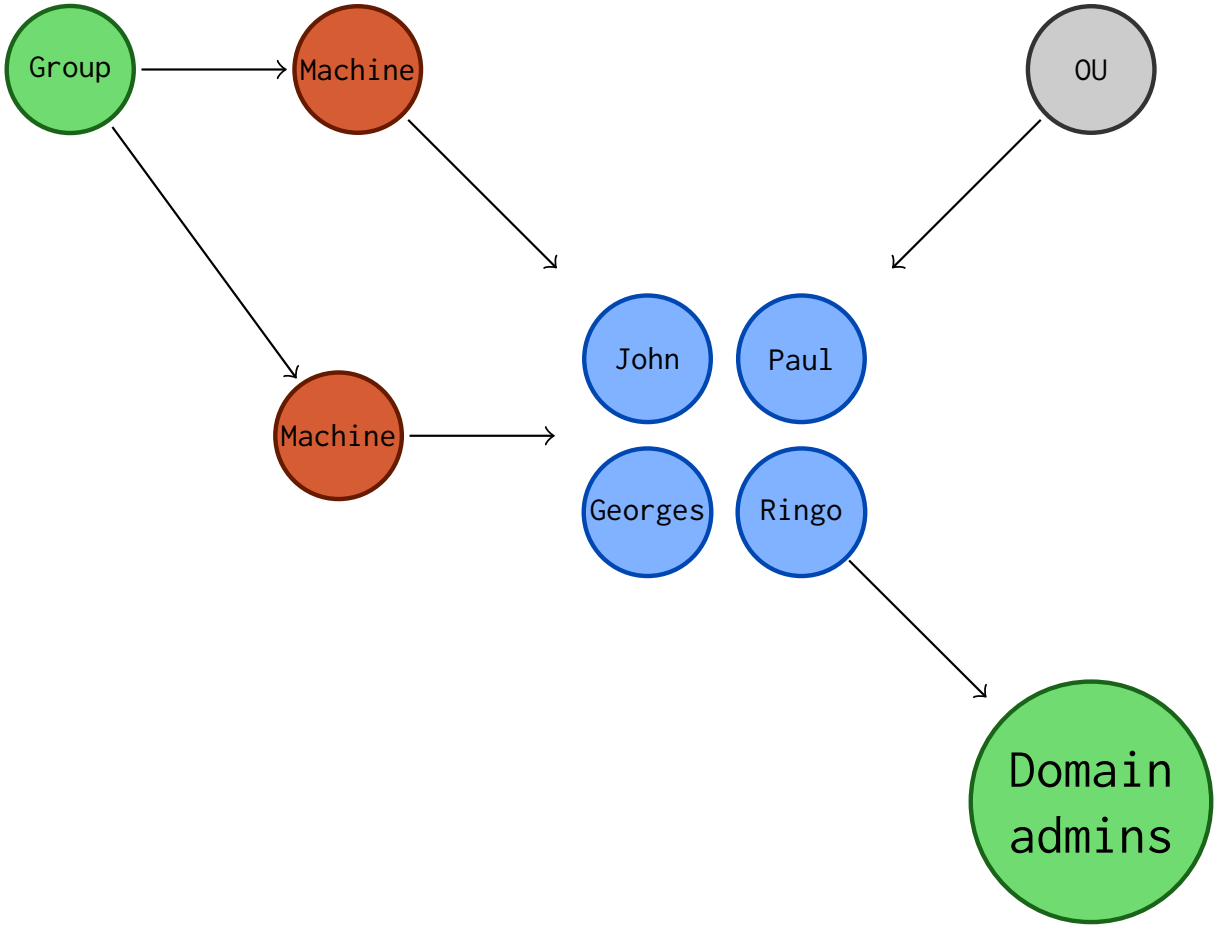
Qui est administrateur de mon domaine ?

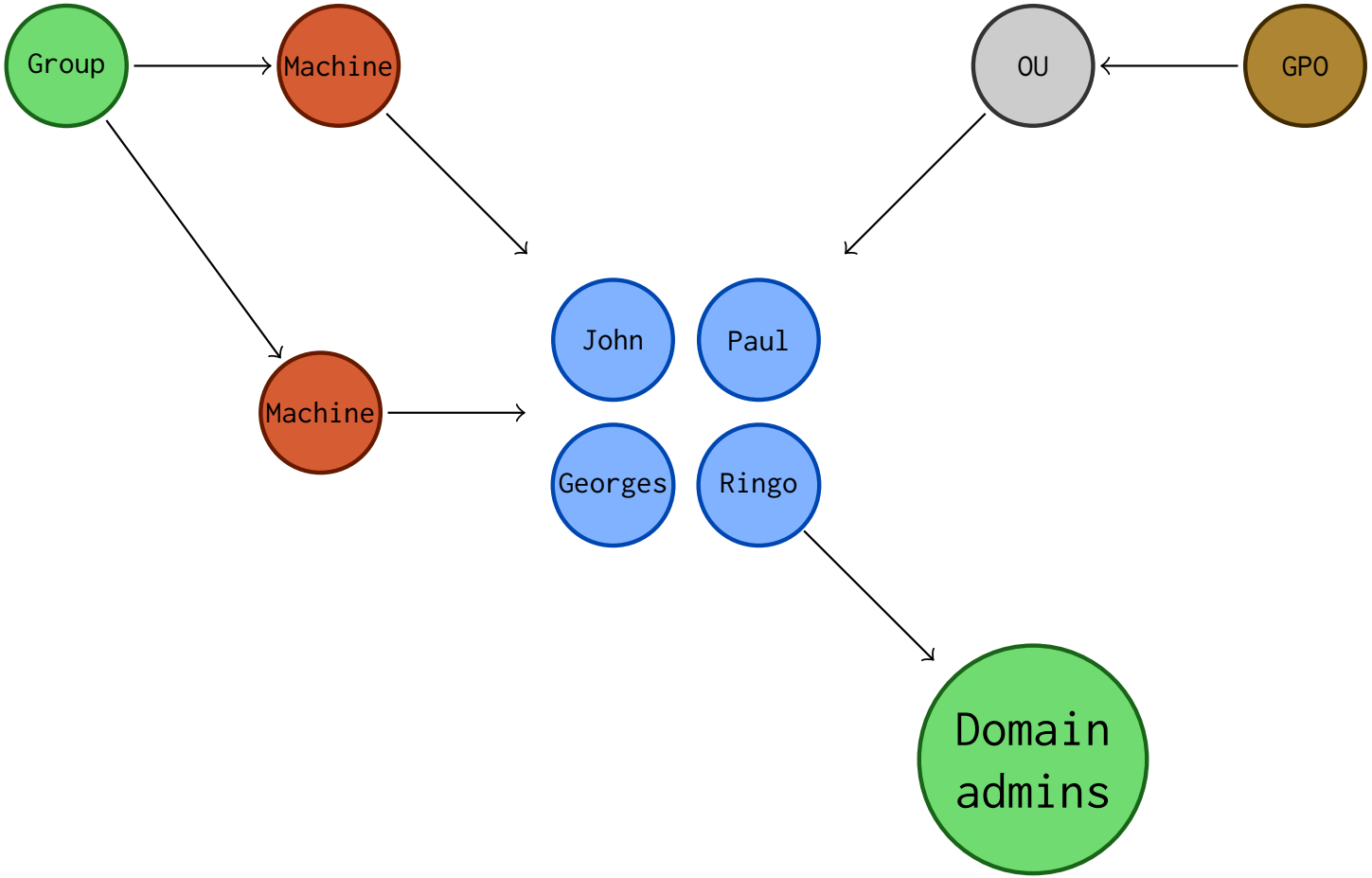

Domain
admins

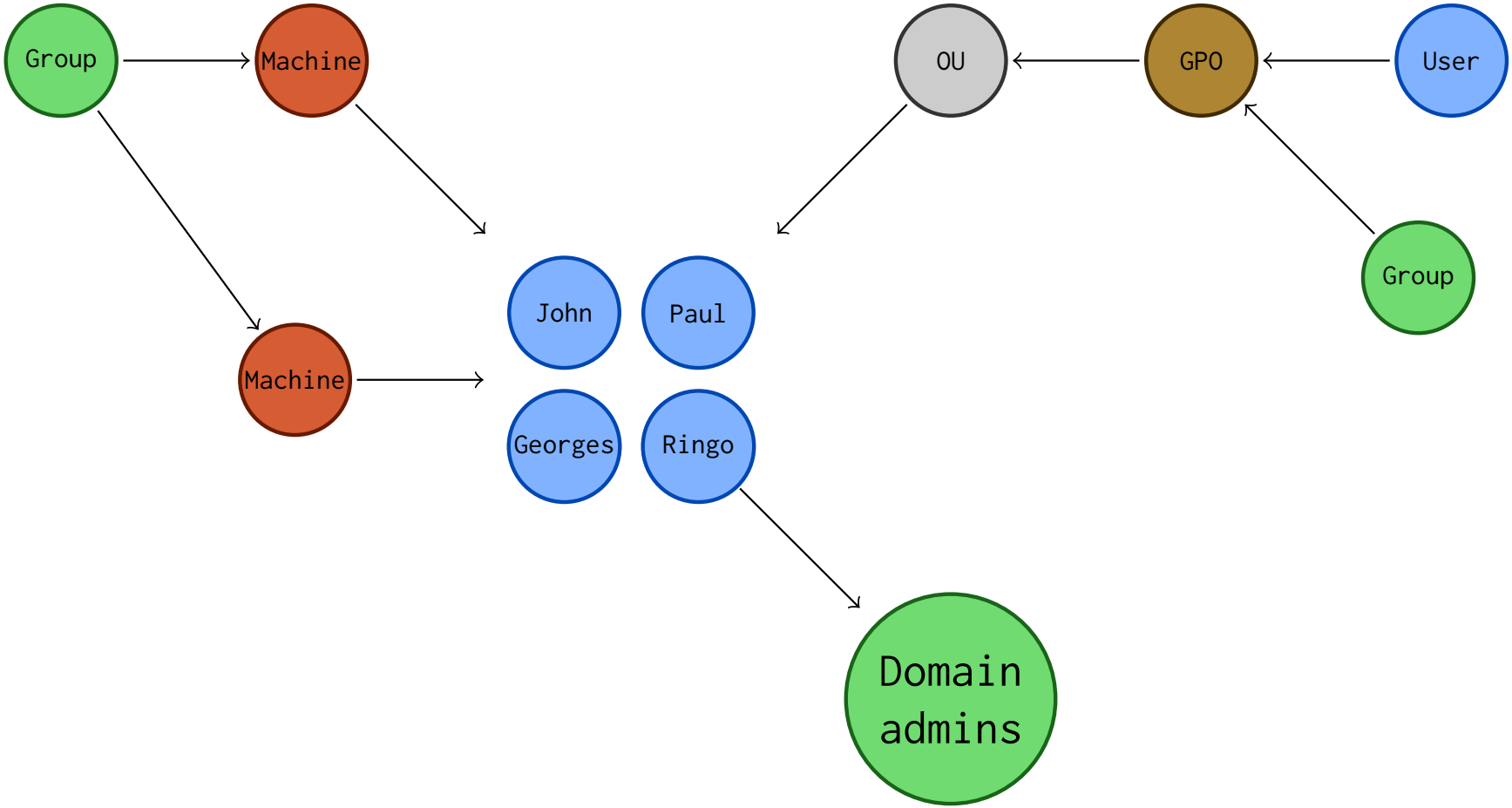

John
Paul
Georges
Ringo
Domain admins

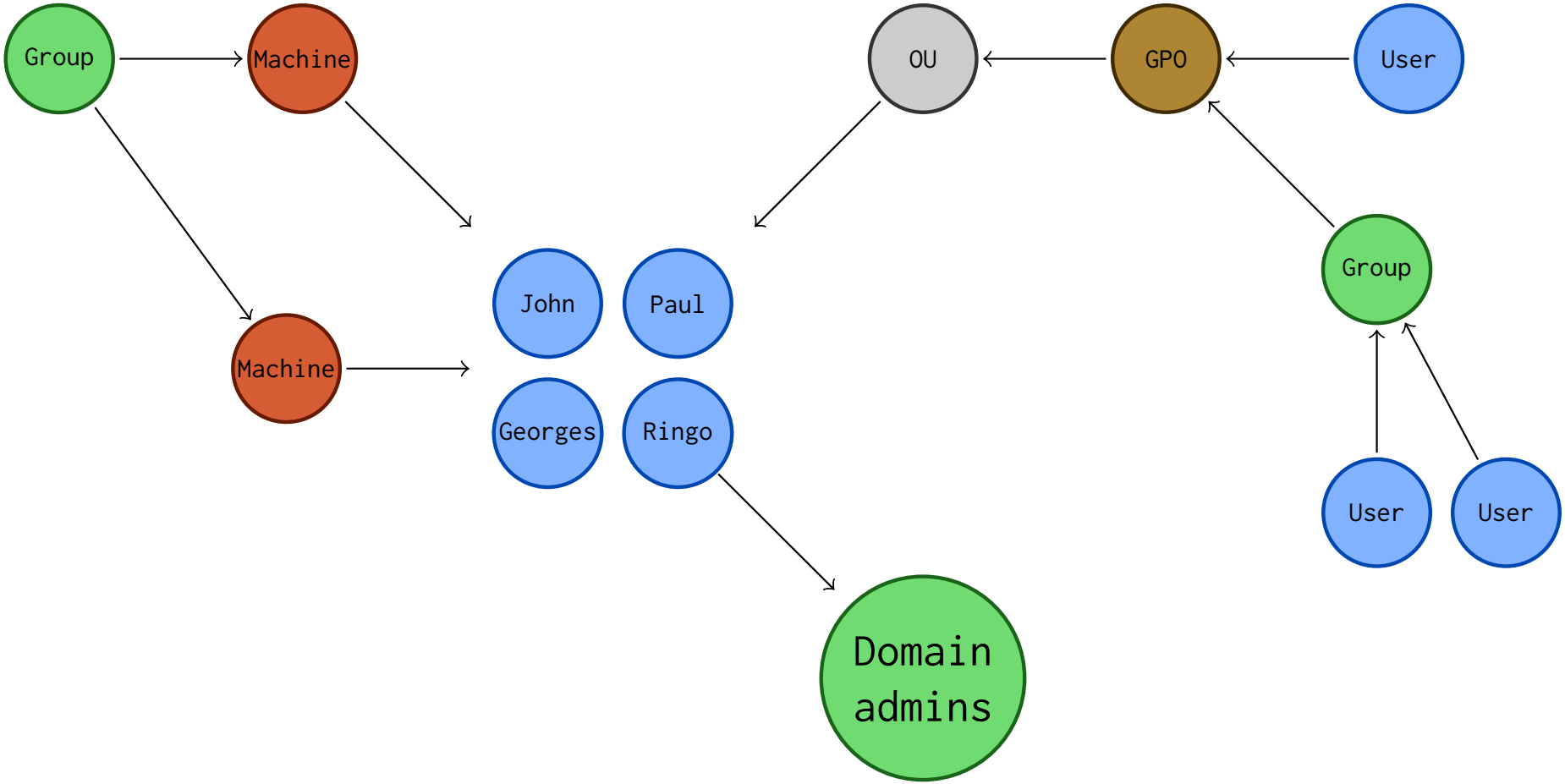

Machine
Machine
John
Paul
Georges
Ringo
Domain admins

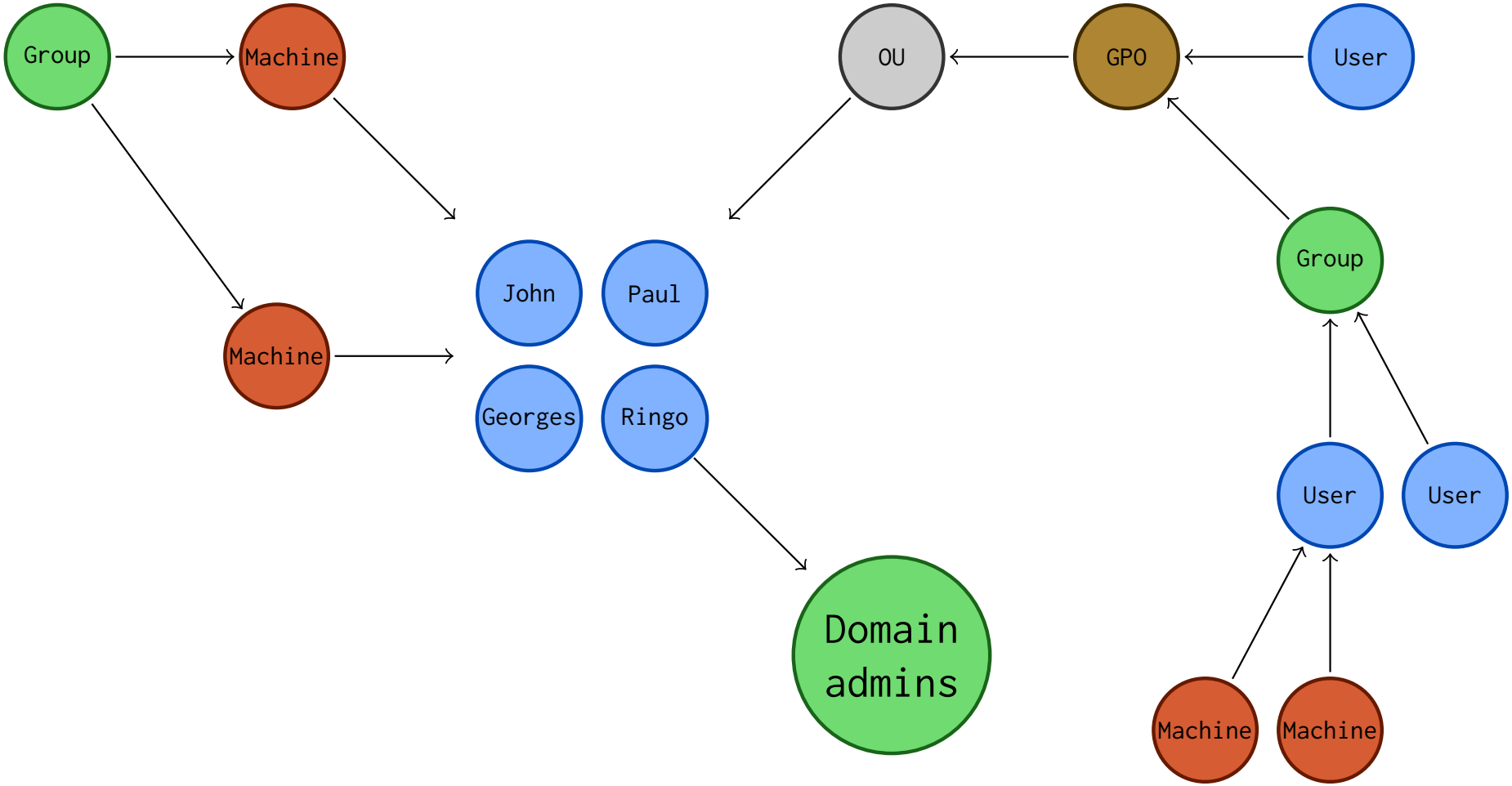

Group
Machine
Machine
John
Paul
Georges
Ringo
Domain admins

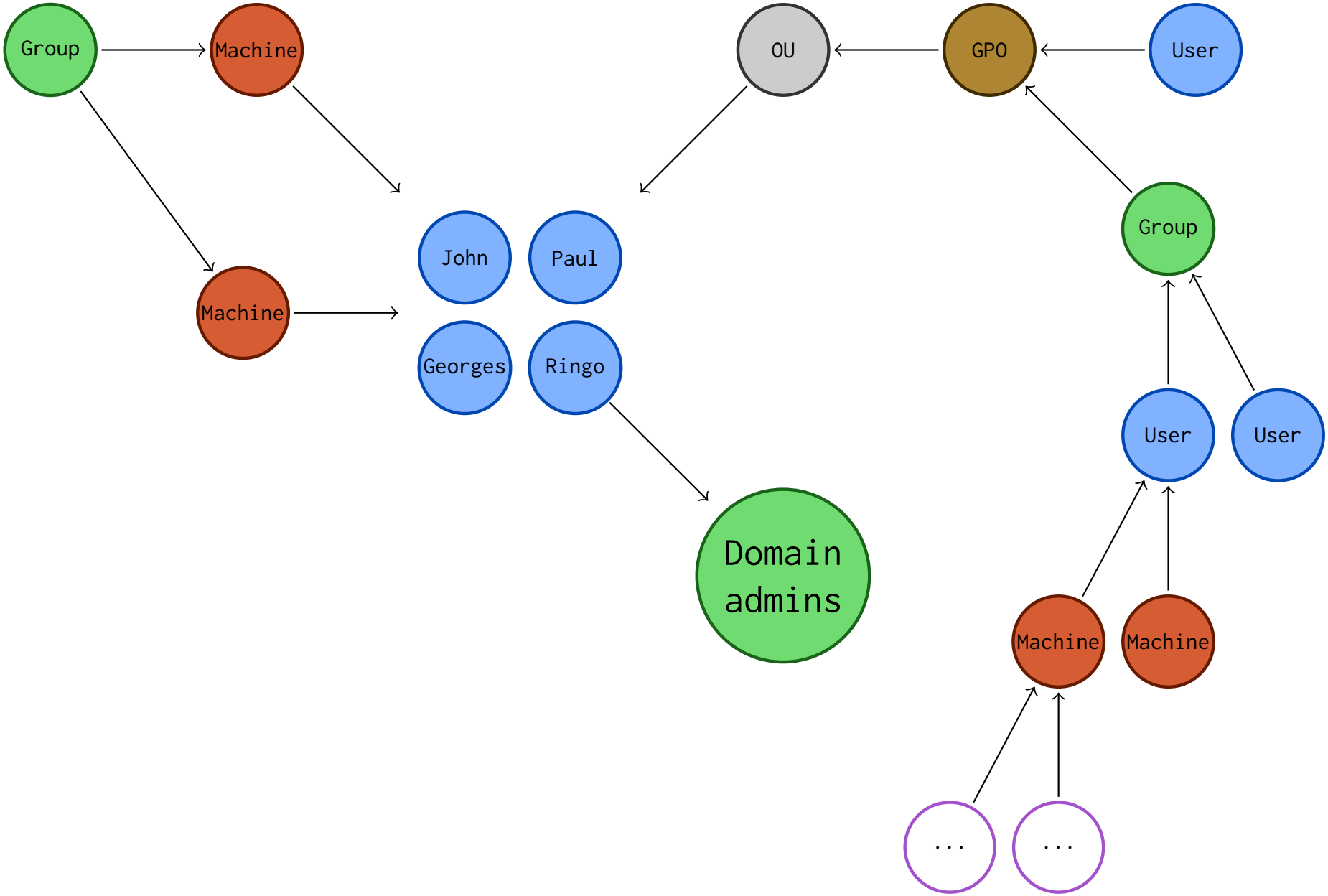

Group
Machine
OU
John
Paul
Machine
Georges
Ringo
Domain admins

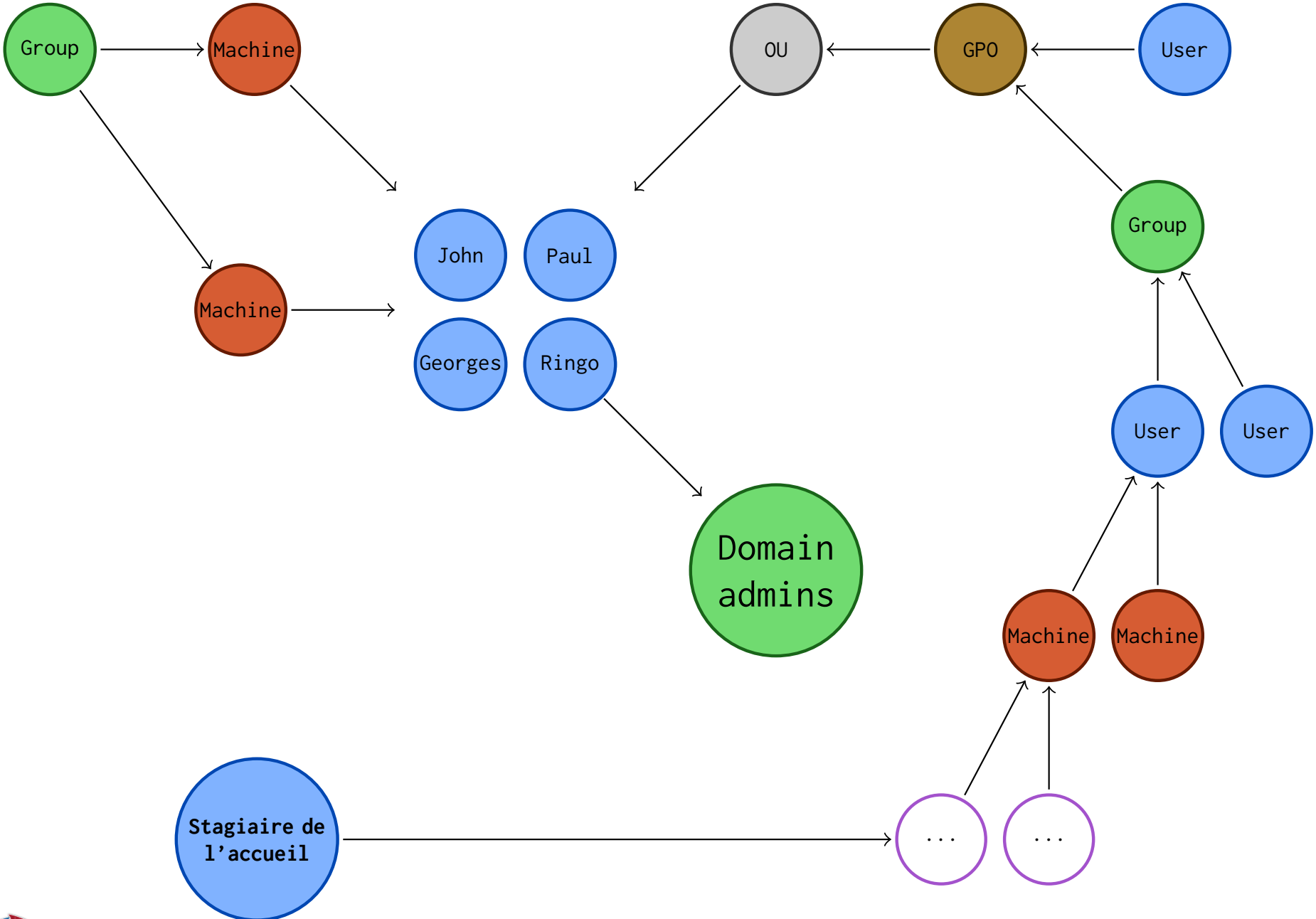

Group
Machine
OU
GPO
John
Paul
Machine
Georges
Ringo
Domain admins

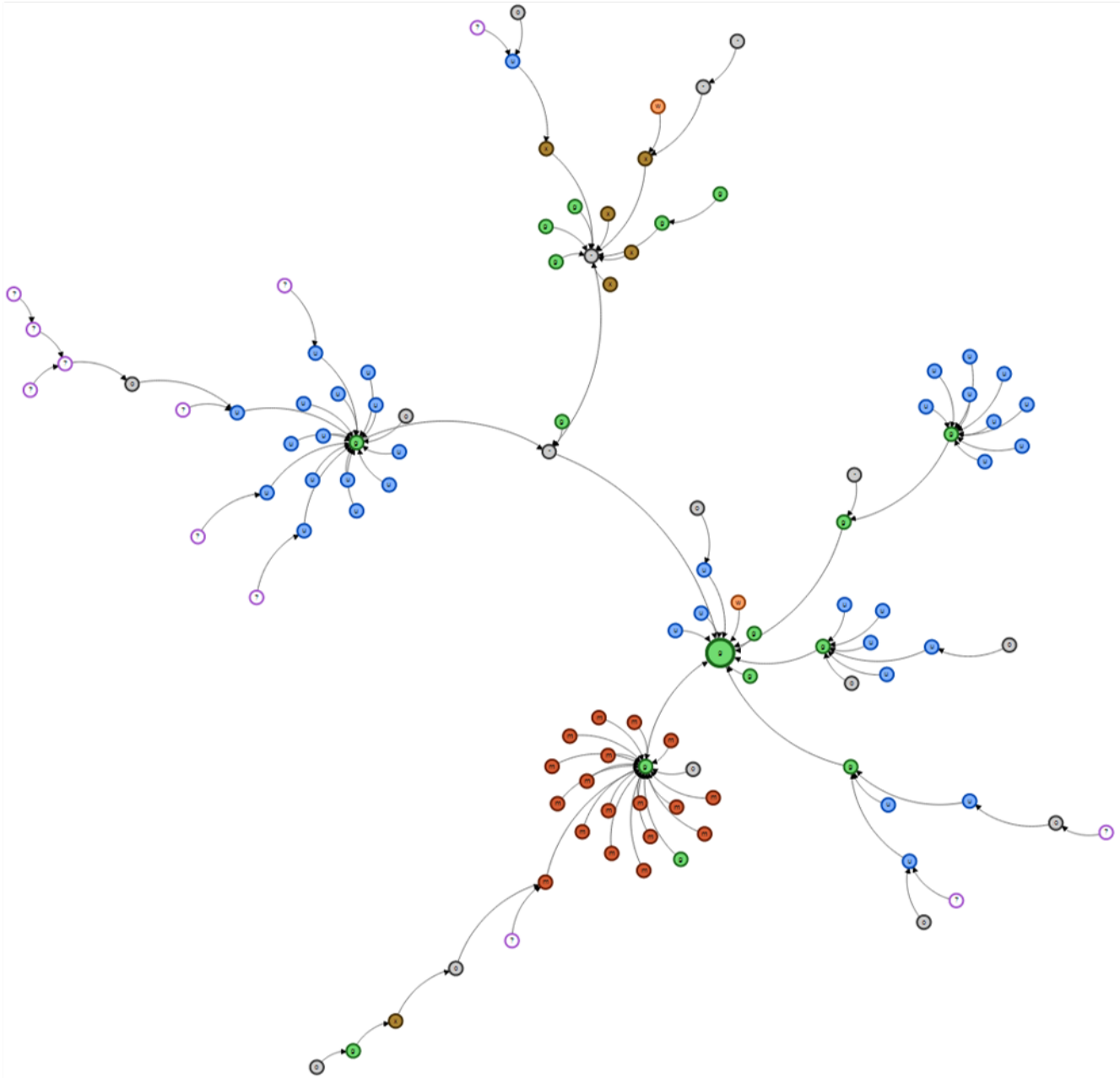

Group
Machine
OU
GPO
User
Group
John
Paul
Machine
Georges
Ringo
Domain admins


Group
Machine
OU
GPO
User
Group
John
Paul
Machine
Georges
Ringo
User
User
Domain admins


Group
Machine
OU
GPO
User
Machine
John
Paul
Georges
Ringo
Group
Domain admins
User
User
Machine
Machine


Group
Machine
OU
GPO
User
Group
John
Paul
Machine
Georges
Ringo
User
User
Domain admins
Machine
Machine
...
...


Group
Machine
OU
GPO
User
Group
John
Paul
Machine
Georges
Ringo
User
User
Domain admins
Machine
Machine
Stagiaire de l'accueil
...
...

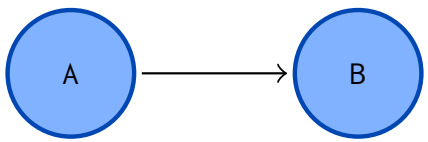


Section 2

Relations et chemins de contrôle

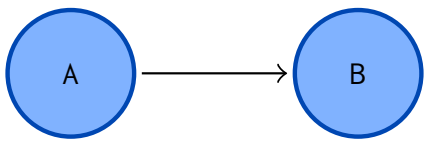
Relation de contrôle :

- Traduit la maîtrise d'un objet sur un autre
- Orientée de l'objet maître vers l'esclave
- Issue de propriétés particulières, de permissions, d'événements, ...



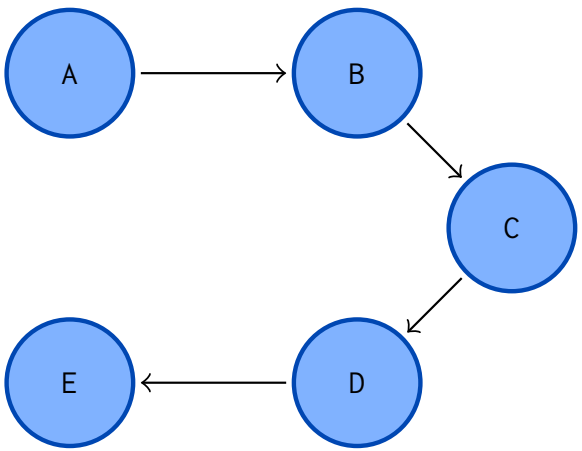
Relation de contrôle :

- Traduit la maîtrise d'un objet sur un autre
- Orientée de l'objet maître vers l'esclave
- Issue de propriétés particulières, de permissions, d'événements, ...



Chemin de contrôle :

- Agrégation de relations
- Enchaînement successif d'actions non-triviales
- Partant d'un noeud, ou arrivant à un noeud

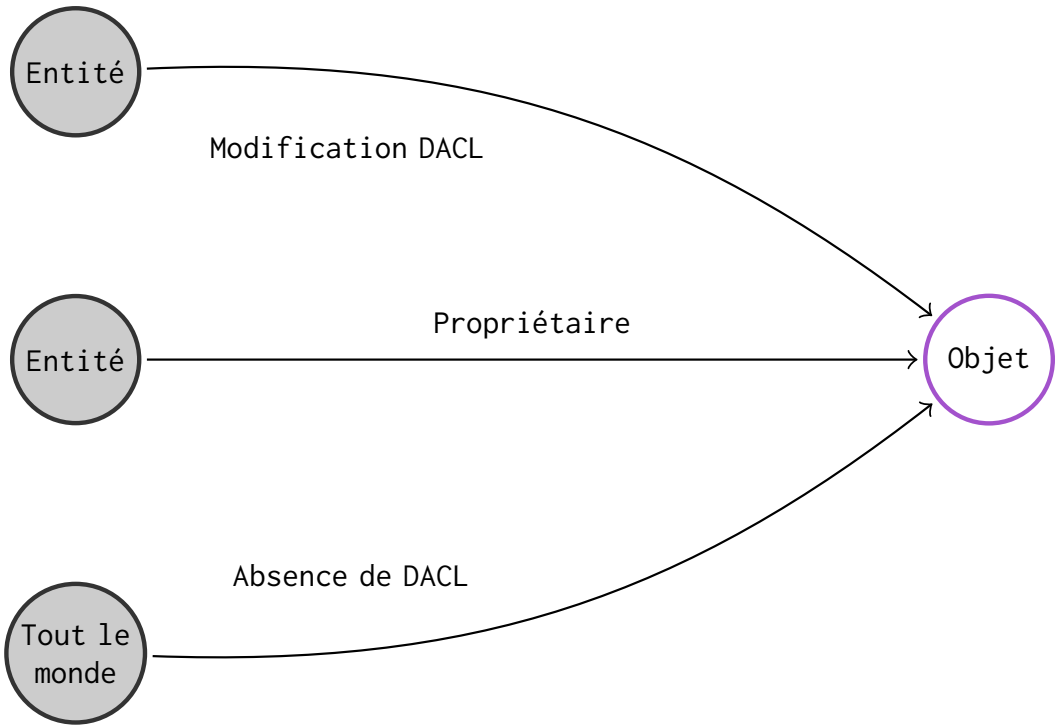


Types de relations :

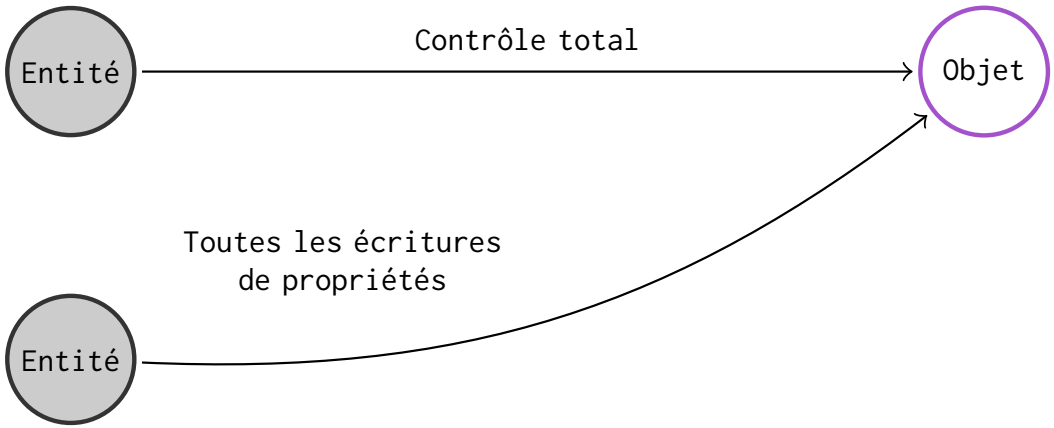
- Génériques pour tous les securable objects
- Génériques pour tous les objets de l'annuaire
- Particulières, de hiérarchie
- Spécifiques à des classes d'objets : User, Computer, Group, OU, ...
- Liées aux stratégies de groupe
- Liées aux machines locales

- Plus on considère de types d'objet et de relations, plus les chemins trouvés seront exhaustifs et complexes

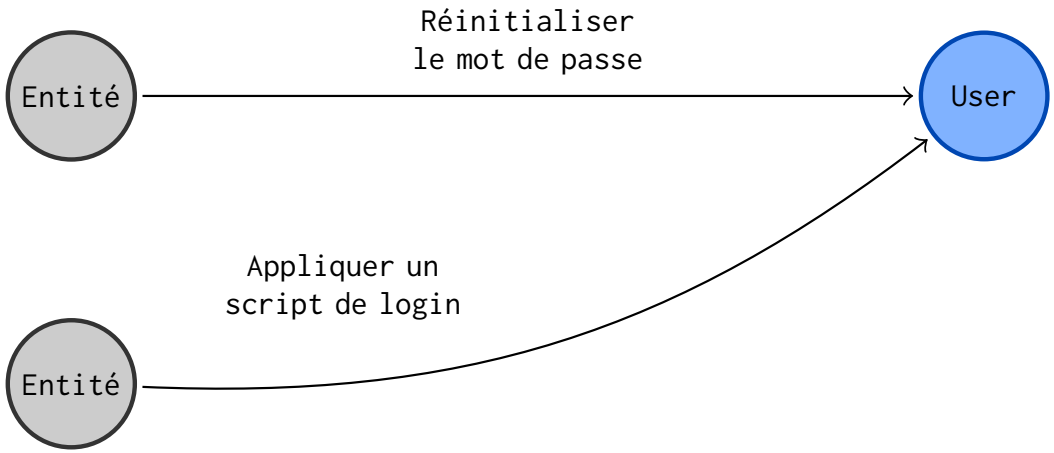
Relations applicables à tous les securable objects :



Relations liées à l'annuaire :

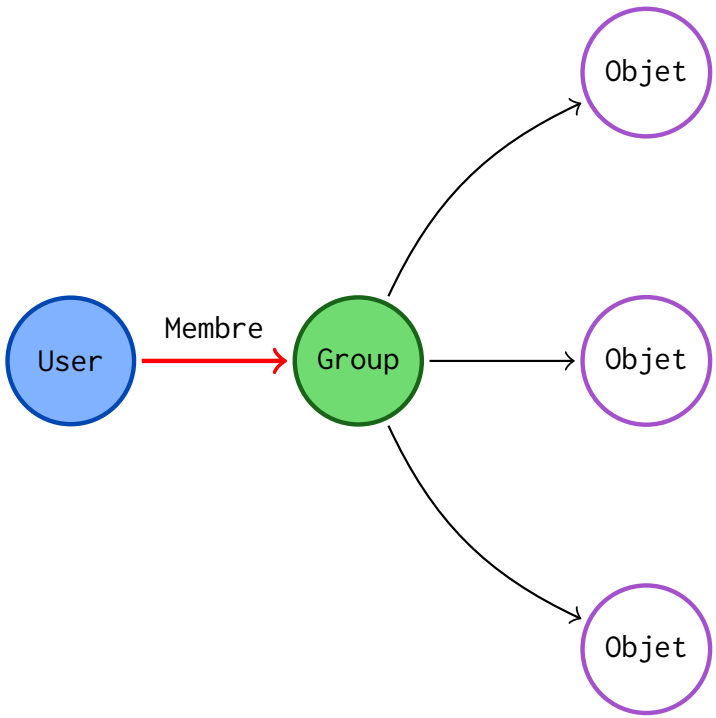


Relations applicables aux utilisateurs :

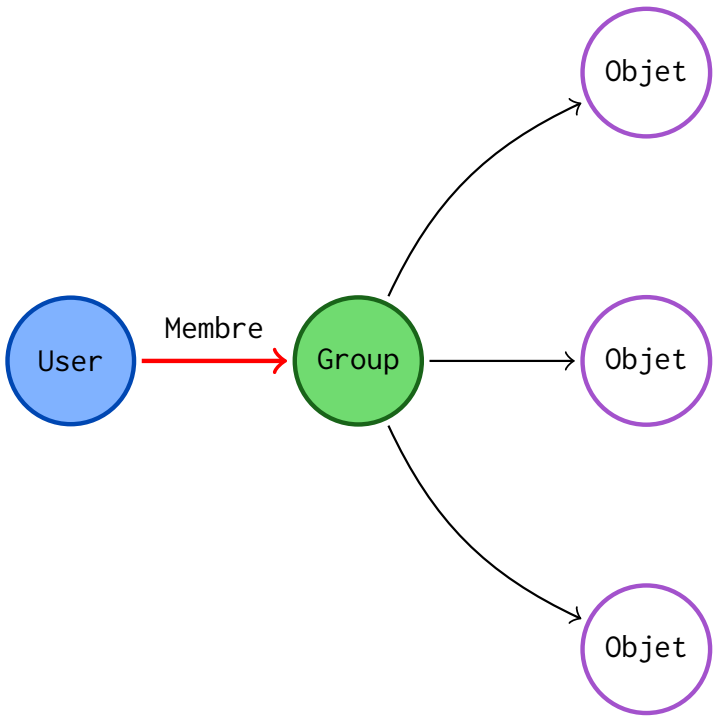


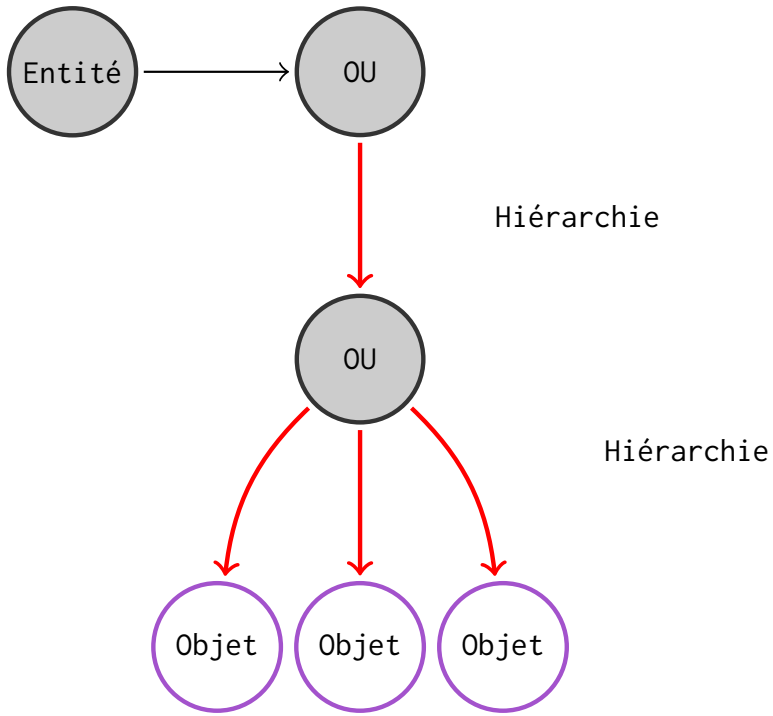
Relations liées aux hiérarchies :

Relations liées aux hiérarchies :

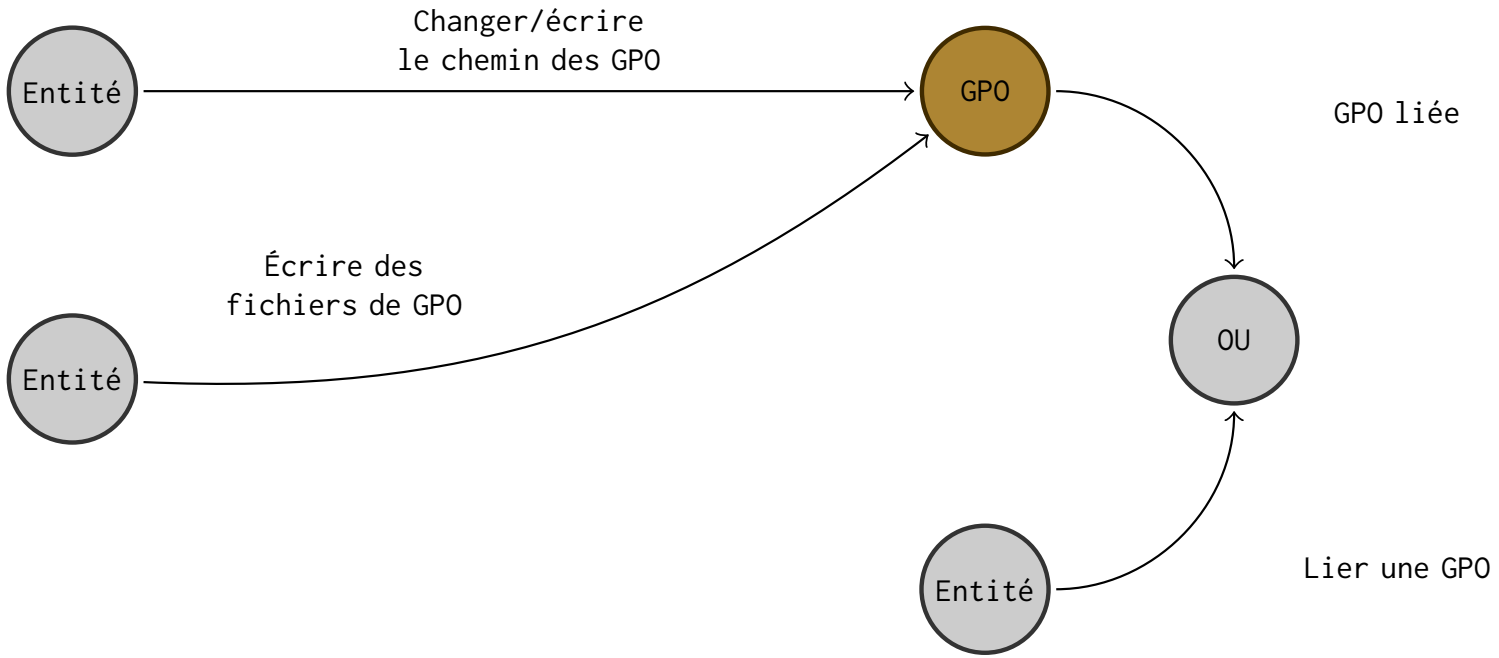


Relations liées aux hiérarchies :

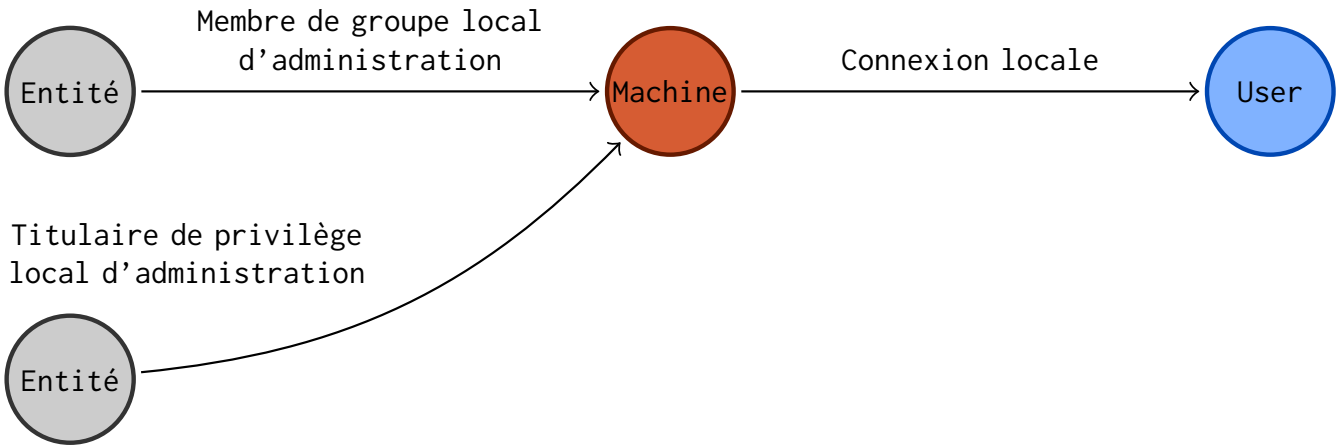




Relations liées aux stratégies de groupe :



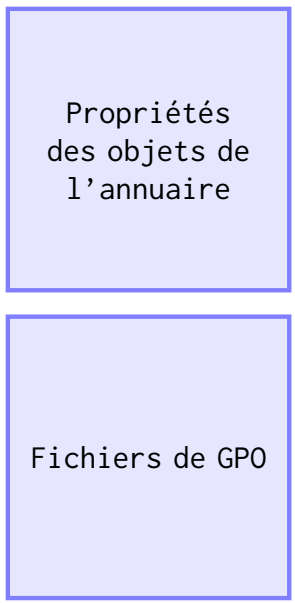
Relations liées aux machines locales :



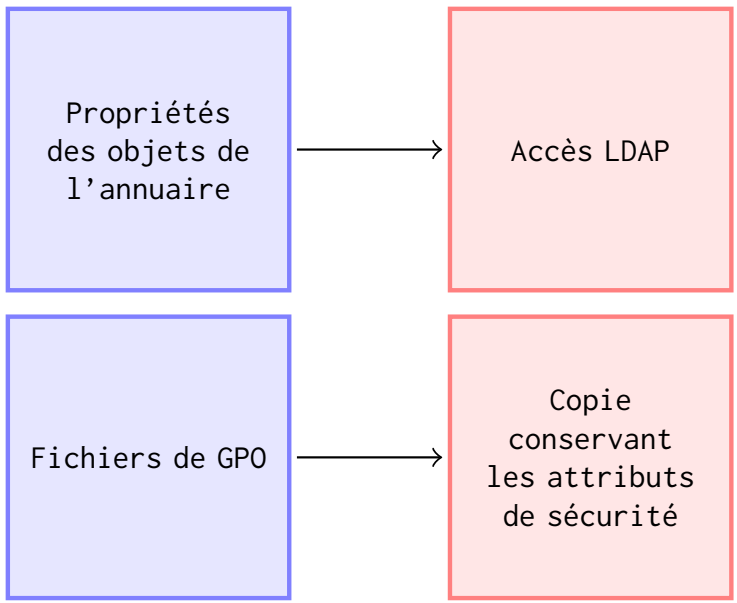
Section 3

Modélisation et outillage

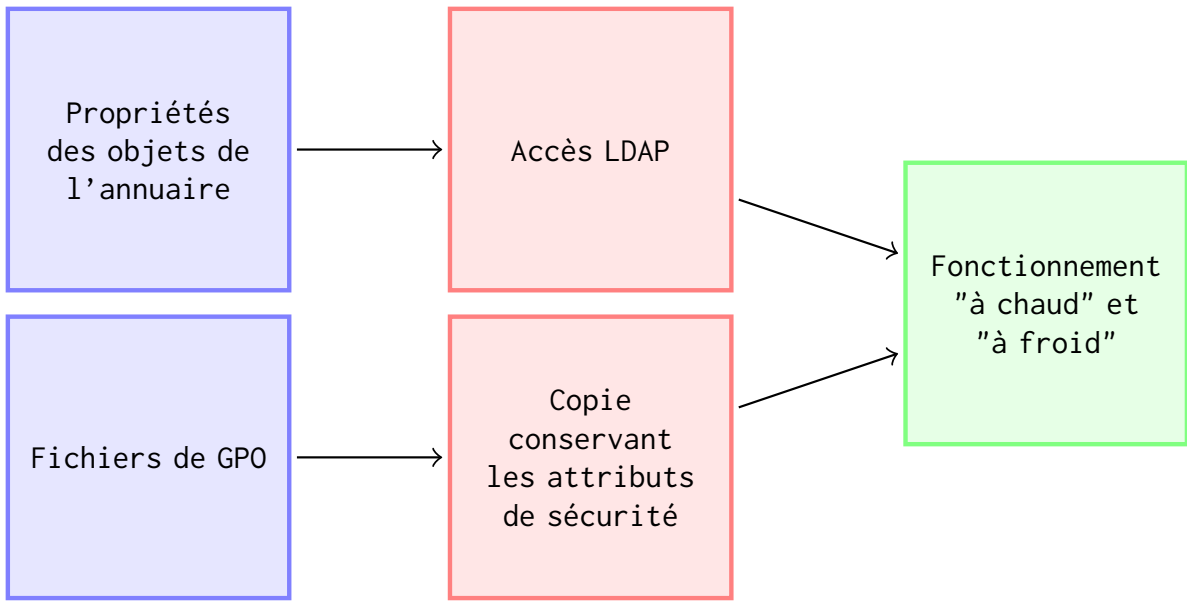
- La majorité des relations de contrôle provient des :



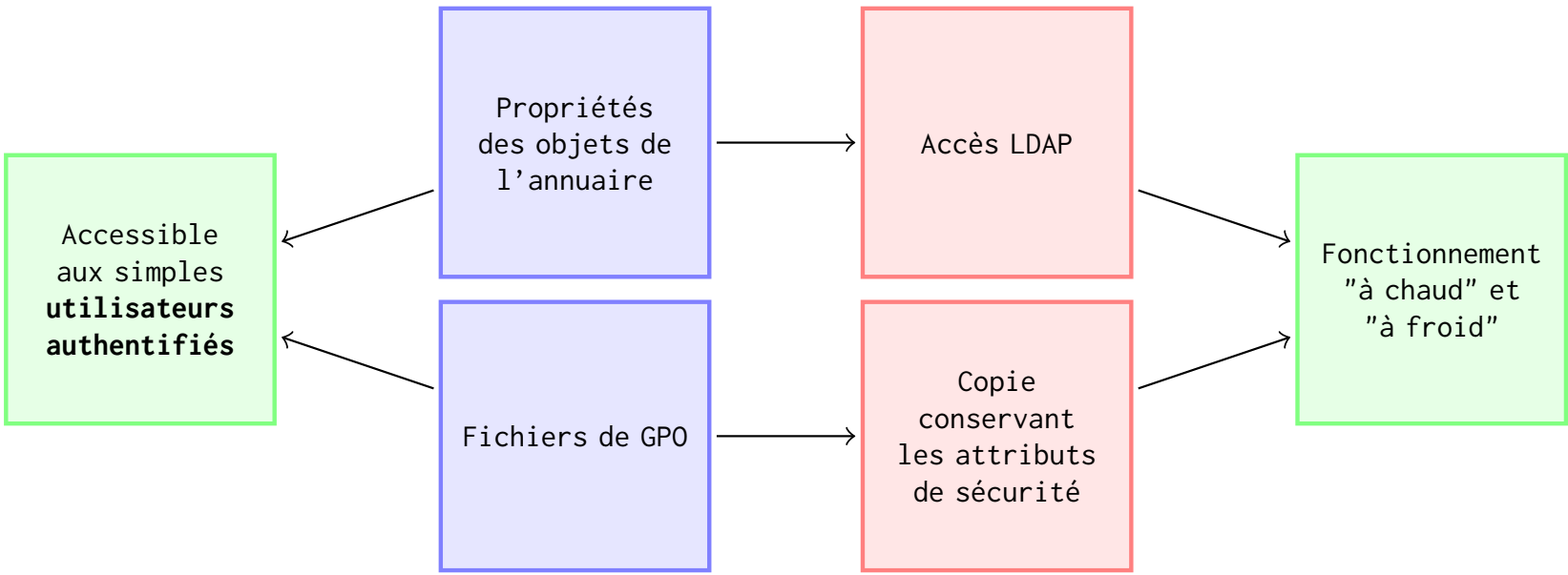
- La majorité des relations de contrôle provient des :



- La majorité des relations de contrôle provient des :



- La majorité des relations de contrôle provient des :



- Fichiers CSV listant les relations regroupées par famille
  - ACE, hiérarchie de conteneurs, appartenance à un groupe
  - Format : MAITRE,ESCLAVE,RELATION
  - Objets représentés par leur Distinguished Name (DN)
- Taille importante
  - Plusieurs dizaines de millions de relations dans un domaine de production

MAITRE	ESCLAVE	RELATION
CN=Domain Controllers,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	REPLICATION_GET_CHANGES_ALL
CN=Administrators,CN=Builtin,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	REPLICATION_GET_CHANGES_ALL
CN=Domain Admins,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	STAND_RIGHT_WRITE_DAC
CN=Domain Admins,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	STAND_RIGHT_WRITE_OWNER
CN=Domain Admins,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	EXT_RIGHT_ALL
CN=Domain Admins,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	WRITE_PROP_ALL
CN=Enterprise Admins,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	STAND_RIGHT_WRITE_DAC
CN=Enterprise Admins,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	STAND_RIGHT_WRITE_OWNER
CN=Enterprise Admins,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	EXT_RIGHT_ALL
CN=Enterprise Admins,CN=Users,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	WRITE_PROP_ALL
CN=Administrators,CN=Builtin,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	STAND_RIGHT_WRITE_DAC
CN=Administrators,CN=Builtin,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	STAND_RIGHT_WRITE_OWNER
CN=Administrators,CN=Builtin,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	EXT_RIGHT_ALL
CN=Administrators,CN=Builtin,DC=dom2012R2,DC=local	DC=dom2012R2,DC=local	WRITE_PROP_ALL

- Base de données orientée graphe
  - Stocke les données sous forme de graphe
  - Optimisée pour parcourir les connexions entre objets du graphe
  - Choix d'implémentation : **Neo4j**
    - Open-source
    - Java
    - Version gratuite
    - Nombreuses API : REST, webapp, Java, Ruby, Python

- Problème : comment importer des millions de noeuds et de relations en un temps raisonnable ?
  - Import à froid (serveur Neo4j éteint)
  - Programme indépendant utilisant l'API Java
  - Très rapide

```
$ time java ControlPathImporter neo4j-community-2.0.3/data/graph.db ~/audits/poneyclub/data/mapping.tsv \
       ~/audits/poneyclub/data/{ace.tsv,gplink.tsv,group.tsv,ou.tsv,owner.tsv,sysvol_ace.tsv}
[+] loading mapping file /home/manu/audits/poneyclub/data/mapping.tsv
[+] importing file /home/manu/audits/poneyclub/data/ace.tsv
[+] importing file /home/manu/audits/poneyclub/data/gplink.tsv
[+] importing file /home/manu/audits/poneyclub/data/group.tsv
[+] importing file /home/manu/audits/poneyclub/data/ou.tsv
[+] importing file /home/manu/audits/poneyclub/data/owner.tsv
[+] importing file /home/manu/audits/poneyclub/data/sysvol_ace.tsv
[+] done: 302243 nodes / 8682912 relationships
java  60.56s user 2.56s system 109% cpu 57.771 total
```

```
neo4j-sh (?)$ MATCH voisin-->(adm {name:"cn=domain admins,cn=users,dc=dom2012r2,dc=local"})
              RETURN DISTINCT voisin;
+-------------------------------------------------------------------------------------------------------------+
| voisin                                                                                                      |
+-------------------------------------------------------------------------------------------------------------+
| Node[3]{name:"cn=users,dc=dom2012r2,dc=local"}                                                              |
| Node[147]{name:"cn=administrator,cn=users,dc=dom2012r2,dc=local"}                                           |
| Node[151]{name:"cn=administrators,cn=builtin,dc=dom2012r2,dc=local"}                                        |
| Node[182]{name:"cn=enterprise admins,cn=users,dc=dom2012r2,dc=local"}                                       |
| Node[184]{name:"cn=domain admins,cn=users,dc=dom2012r2,dc=local"}                                           |
| Node[1978]{name:"cn=system,cn=wellknown security principals,cn=configuration,dc=dom2012r2,dc=local"}       |
+-------------------------------------------------------------------------------------------------------------+
6 rows
159 ms
```

```
neo4j-sh (?)$ MATCH
                path=shortestPath(voisin-[*..5]->(adm {name:"cn=domain admins,cn=users,dc=dom2012r2,dc=local"}))
              RETURN DISTINCT voisin, length(path) AS len ORDER BY len;
+-------------------------------------------------------------------------------------------------------------------+
| voisin                                                                                                            | len |
+-------------------------------------------------------------------------------------------------------------------+
| Node[184]{name:"cn=domain admins,cn=users,dc=dom2012r2,dc=local"}                                                 | 0   |
| Node[3]{name:"cn=users,dc=dom2012r2,dc=local"}                                                                    | 1   |
| Node[147]{name:"cn=administrator,cn=users,dc=dom2012r2,dc=local"}                                                 | 1   |
| Node[151]{name:"cn=administrators,cn=builtin,dc=dom2012r2,dc=local"}                                              | 1   |
| Node[182]{name:"cn=enterprise admins,cn=users,dc=dom2012r2,dc=local"}                                             | 1   |
| Node[1978]{name:"cn=system,cn=wellknown security principals,cn=configuration,dc=dom2012r2,dc=local"}             | 1   |
| Node[2]{name:"dc=dom2012r2,dc=local"}                                                                             | 2   |
| Node[150]{name:"cn=builtin,dc=dom2012r2,dc=local"}                                                                | 2   |
| Node[22]{name:"cn={31b2f340-016d-11d2-945f-00c04fb984f9},cn=policies,cn=system,dc=dom2012r2,dc=local"}            | 3   |
| Node[180]{name:"cn=domain controllers,cn=users,dc=dom2012r2,dc=local"}                                            | 3   |
| Node[21]{name:"cn=policies,cn=system,dc=dom2012r2,dc=local"}                                                      | 4   |
| Node[1979]{name:"cn=creator owner,cn=wellknown sp,cn=configuration,dc=dom2012r2,dc=local"}                        | 4   |
| Node[5]{name:"cn=system,dc=dom2012r2,dc=local"}                                                                   | 5   |
+-------------------------------------------------------------------------------------------------------------------+
14 rows
66 ms
```

```
neo4j-sh (?)$ START n=node(22), adm=node(184) MATCH path=shortestPath(n-[*..3]->adm) RETURN path;
+-------------------------------------------------------------------------------------------------------------------+
| path                                                                                                              |
+-------------------------------------------------------------------------------------------------------------------+
|                                                                                                                   |
| [                                                                                                                 |
|   Node[22]{name:"cn={31b2f340-016d-11d2-945f-00c04fb984f9},cn=policies,cn=system,dc=dom2012r2,dc=local"},         |
|     :gplink[1958]{},                                                                                              |
|   Node[2]{name:"dc=dom2012r2,dc=local"},                                                                          |
|     :container_hierarchy[1]{},                                                                                    |
|   Node[3]{name:"cn=users,dc=dom2012r2,dc=local"},                                                                 |
|     :container_hierarchy[181]{},                                                                                  |
|   Node[184]{name:"cn=domain admins,cn=users,dc=dom2012r2,dc=local"}                                               |
| ]                                                                                                                 |
+-------------------------------------------------------------------------------------------------------------------+
1 row
35 ms
```

- Les résultats des requêtes ne sont pas franchement esthétiques :
  - Export des sous-graphes en JSON
  - Utilisation de la bibliothèque **d3.js** pour visualiser les graphes

Section 4

Scénarios d'analyse

Cibles d'administration

Groupes d'administration, groupes d'opérateurs, contrôleurs de domaine, etc.

Cibles d'administration	Groupes d'administration, groupes d'opérateurs, contrôleurs de domaine, etc.
Cibles ayant accès à des données	Issues de l'analyse des permissions NTFS des filers. Accès à des données métier, à de grandes quantités, etc.

Cibles d'administration	Groupes d'administration, groupes d'opérateurs, contrôleurs de domaine, etc.
Cibles ayant accès à des données	Issues de l'analyse des permissions NTFS des filers. Accès à des données métier, à de grandes quantités, etc.
Cibles VIP	Objets d'attaques ciblées, accès à des données particulières.

Cibles d'administration	Groupes d'administration, groupes d'opérateurs, contrôleurs de domaine, etc.
Cibles ayant accès à des données	Issues de l'analyse des permissions NTFS des filers. Accès à des données métier, à de grandes quantités, etc.
Cibles VIP	Objets d'attaques ciblées, accès à des données particulières.
Sources connues comme compromises	Vérification de l'étendue possible de la compromission.

- **Question** : Qui est administrateur de mon domaine ?
- **Réponse** : Graphe des chemins de contrôle arrivant jusqu'au nœud central "administrateurs du domaine"

- **Question** : Qui est administrateur de mon domaine ?
- **Réponse** : Graphe des chemins de contrôle arrivant jusqu'au nœud central "administrateurs du domaine"

- Différents domaines :
  - domaine vierge
  - domaine "relativement simple"
  - domaine complexe

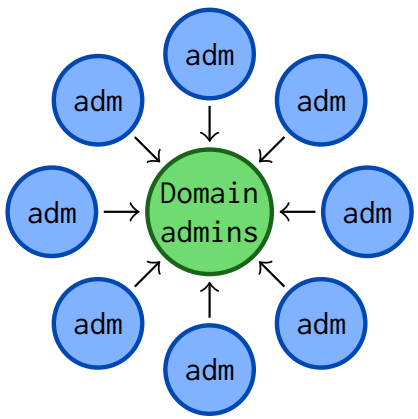
- **Question** : Qui est administrateur de mon domaine ?
- **Réponse** : Graphe des chemins de contrôle arrivant jusqu'au nœud central "administrateurs du domaine"

- Différents domaines :
  - domaine vierge
  - domaine "relativement simple"
  - domaine complexe

- Buts :
  - déterminer une situation de référence
  - identifier des déviances
  - identifier des backdoors
  - illustrer la complexité de l'identification d'un périmètre critique

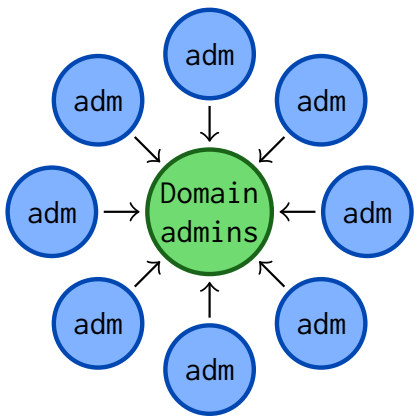
Qui est administrateur de mon domaine ?

- **Réponse du RSSI :**



Qui est administrateur de mon domaine ?

- **Réponse du RSSI :**

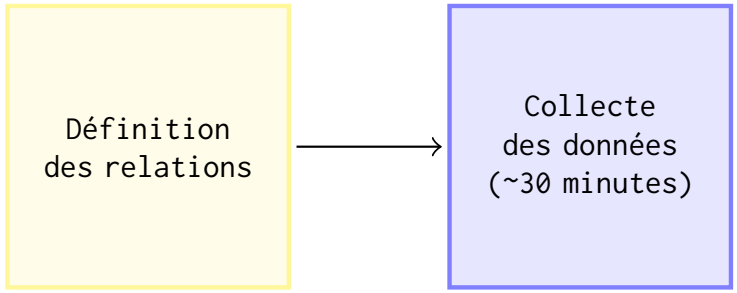


- **Réponse de l'outil :**
  - Vidéo #1 : domaine vierge
  - Vidéo #2 : domaine simple
  - Vidéo #3 : domaine complexe

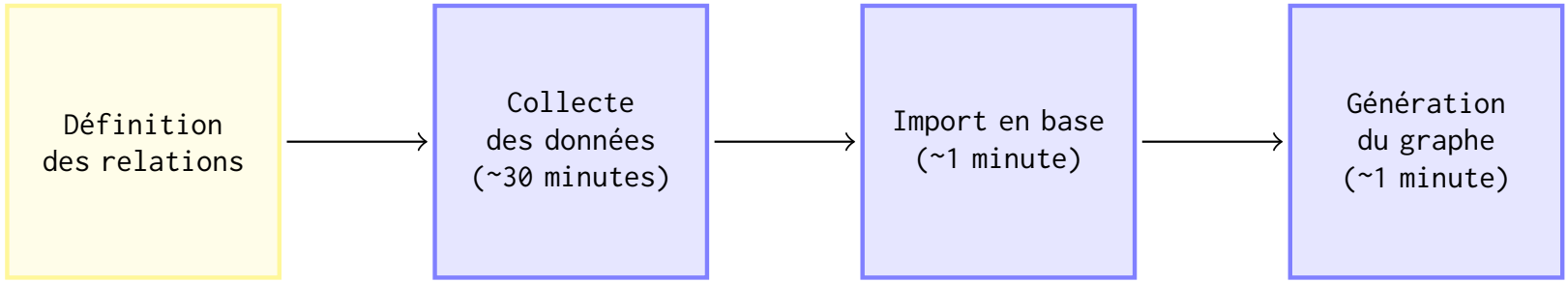
Section 5

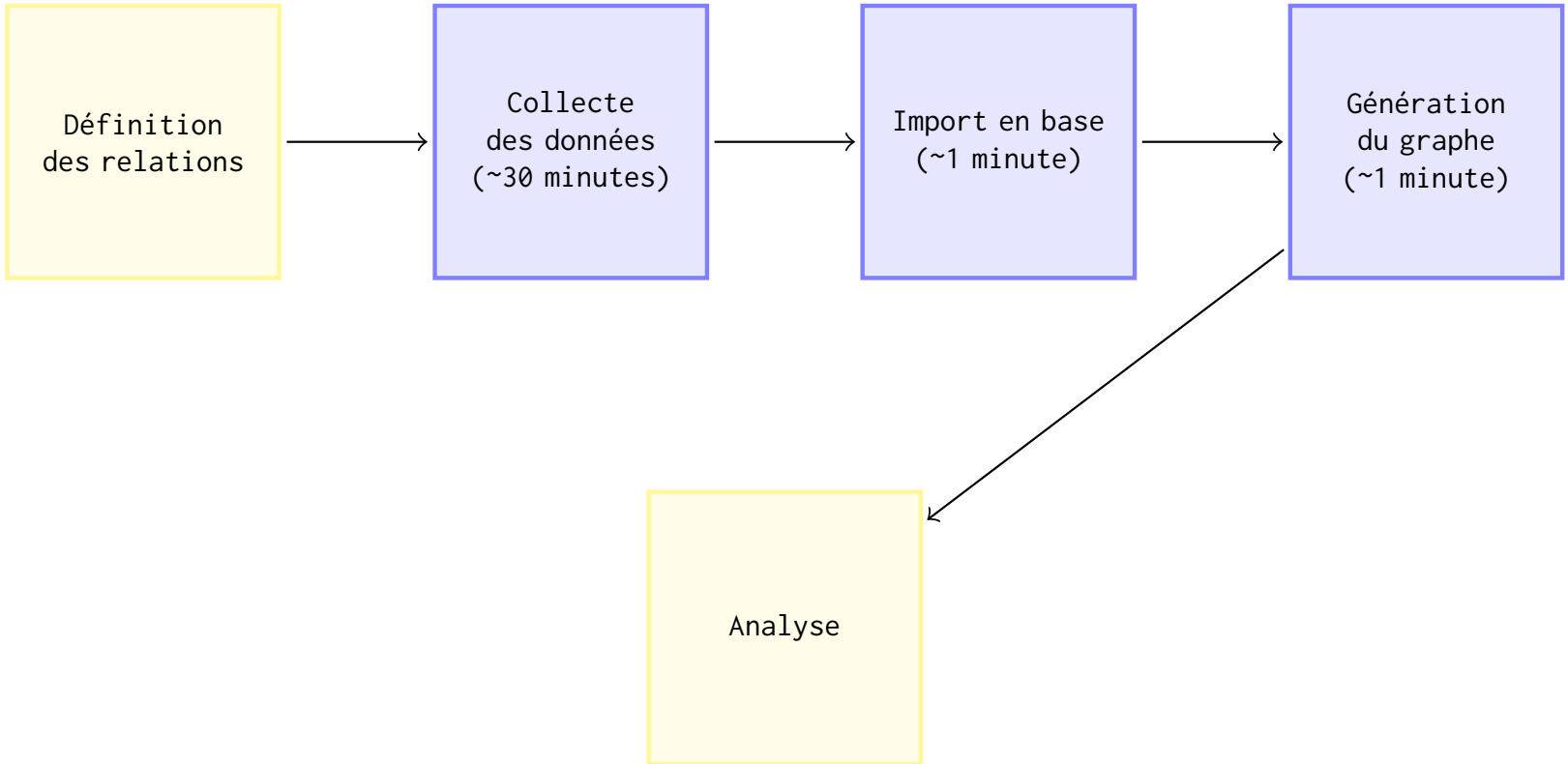
Conclusion

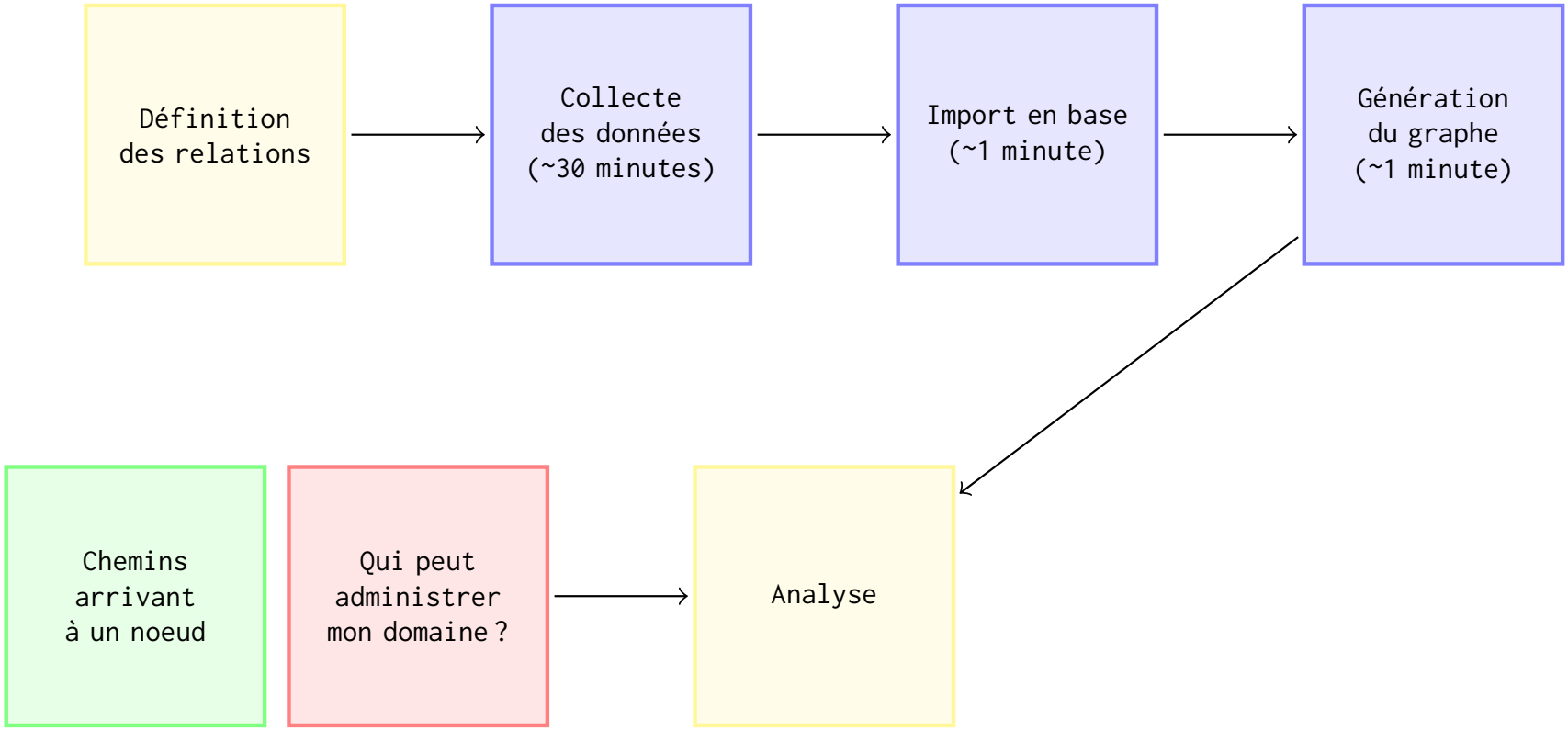
Définition
des relations

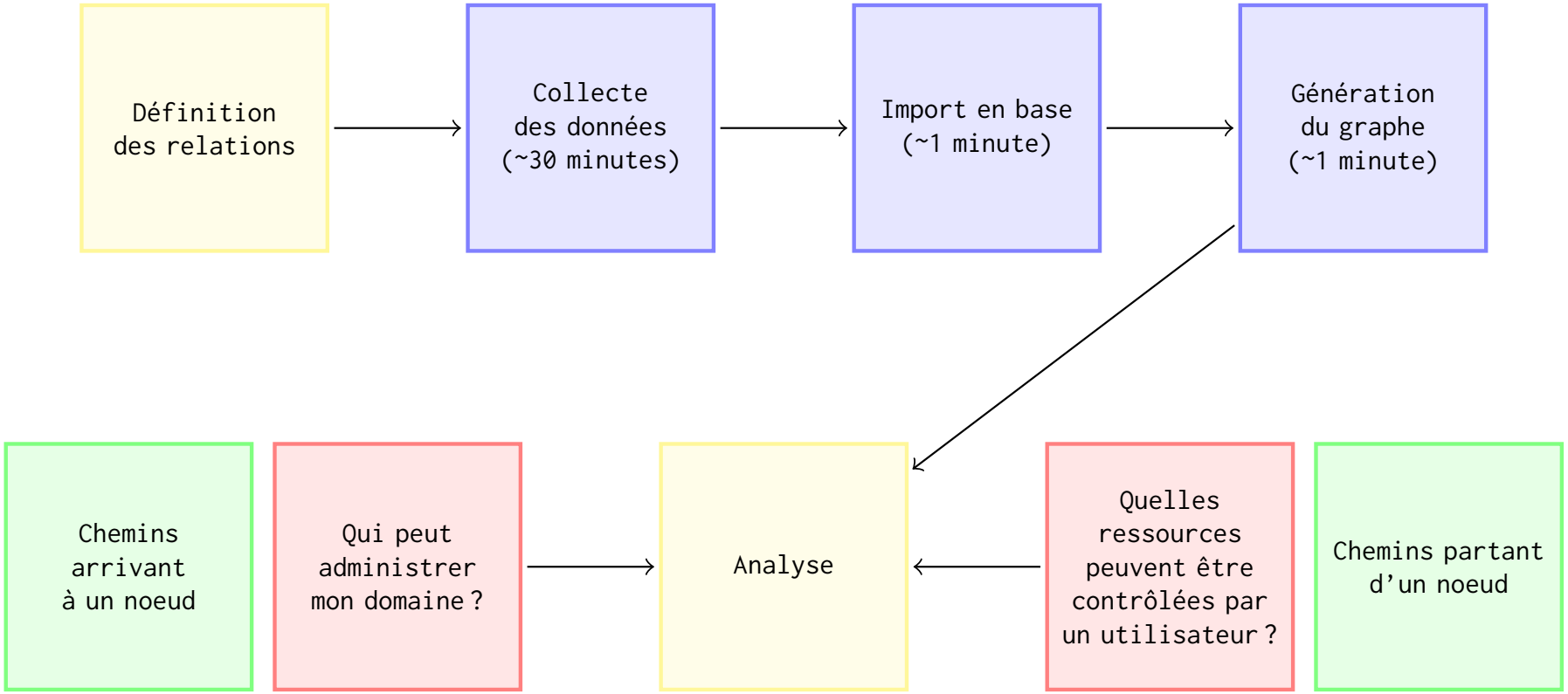

Définition
des relations
Collecte
des données
(~30 minutes)

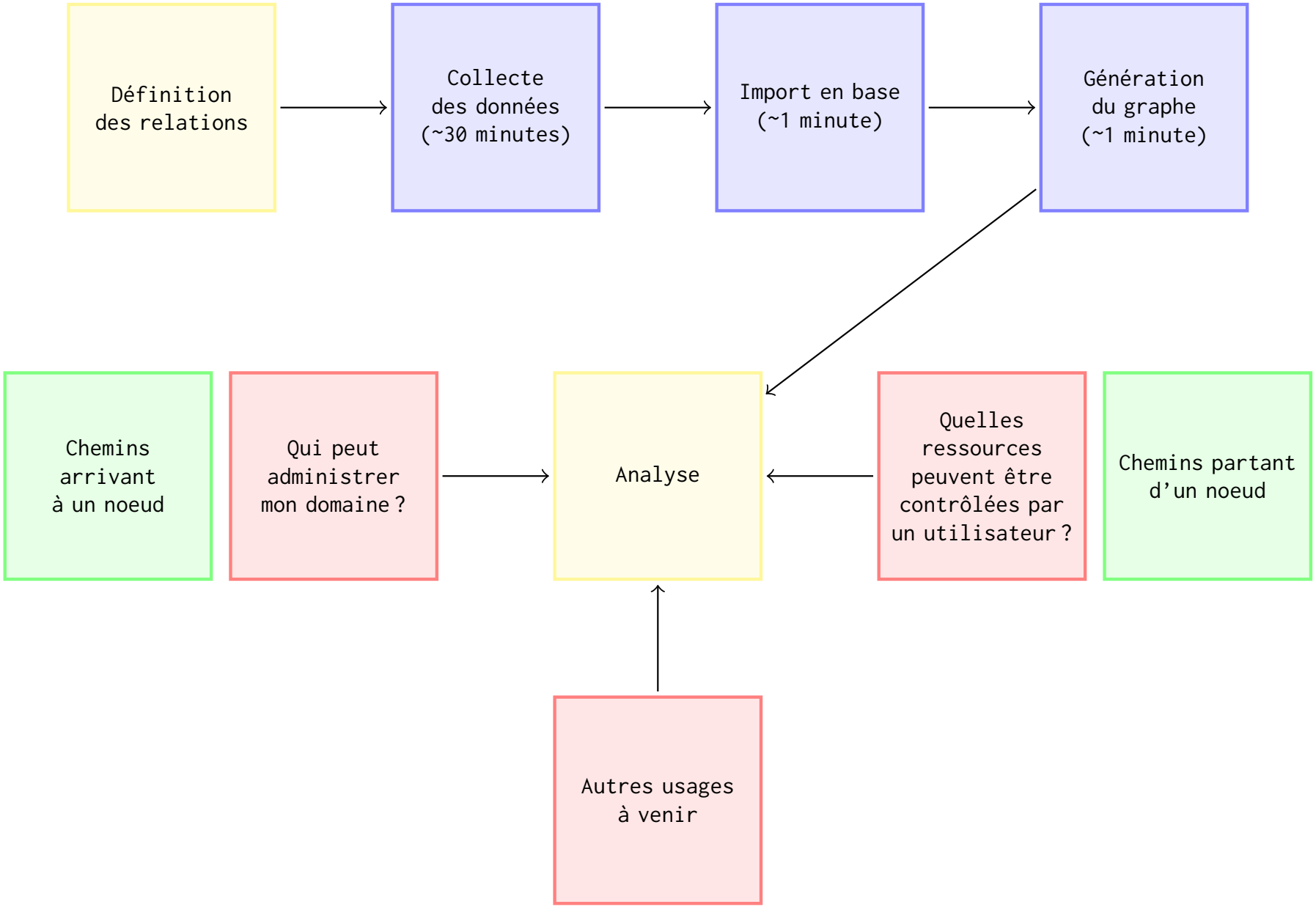

Définition
des relations
Collecte
des données
(~30 minutes)
Import en base
(~1 minute)


Définition
des relations
Collecte
des données
(~30 minutes)
Import en base
(~1 minute)
Génération
du graphe
(~1 minute)


Définition
des relations
Collecte
des données
(~30 minutes)
Import en base
(~1 minute)
Génération
du graphe
(~1 minute)
Analyse


Définition des relations
Collecte des données (~30 minutes)
Import en base (~1 minute)
Génération du graphe (~1 minute)
Chemins arrivant à un noeud
Qui peut administrer mon domaine ?
Analyse


Définition des relations
Collecte des données (~30 minutes)
Import en base (~1 minute)
Génération du graphe (~1 minute)
Chemins arrivant à un noeud
Qui peut administrer mon domaine ?
Analyse
Quelles ressources peuvent être contrôlées par un utilisateur ?
Chemins partant d'un noeud


Définition des relations
Collecte des données (~30 minutes)
Import en base (~1 minute)
Génération du graphe (~1 minute)
Chemins arrivant à un noeud
Qui peut administrer mon domaine ?
Analyse
Quelles ressources peuvent être contrôlées par un utilisateur ?
Chemins partant d'un noeud
Autres usages à venir

Contextes d'utilisation :

- Audit pour durcissement
- Réponse à incident après compromission
- Contrôle régulier (détection de mauvaises pratiques)

Évolution de l'outil :

- Trouver et relever de nouvelles relations
- Améliorer la modélisation des mécanismes d'héritage (GPO, ACE)
- Développer des outils d'aide à l'analyse

Application de la méthode à d'autres périmètres :

- Qui contrôle les boîtes aux lettres Exchange ?
- Qui contrôle les données sur les serveurs de fichiers ?
- Linux ?
  - crontab qui appelle un script, stocké sur un partage sur lequel un utilisateur a les droits d'écriture, ...

Merci de votre attention