


SECOURS 2030

Développement d'une application d'affectation de secouristes

A. BELMOUHCINE

2024-2025

A solid orange horizontal bar at the bottom of the slide.

Défi des JO 2024

- ❖ Les Jeux Olympiques de 2024 ont constitué un défi logistique considérable en matière d'affectation des secouristes :
 - Mobilisation de centaines de bénévoles et de professionnels sur différents sites et à des moments précis
 - Nécessité de garantir une couverture en personnel adaptée aux compétences requises

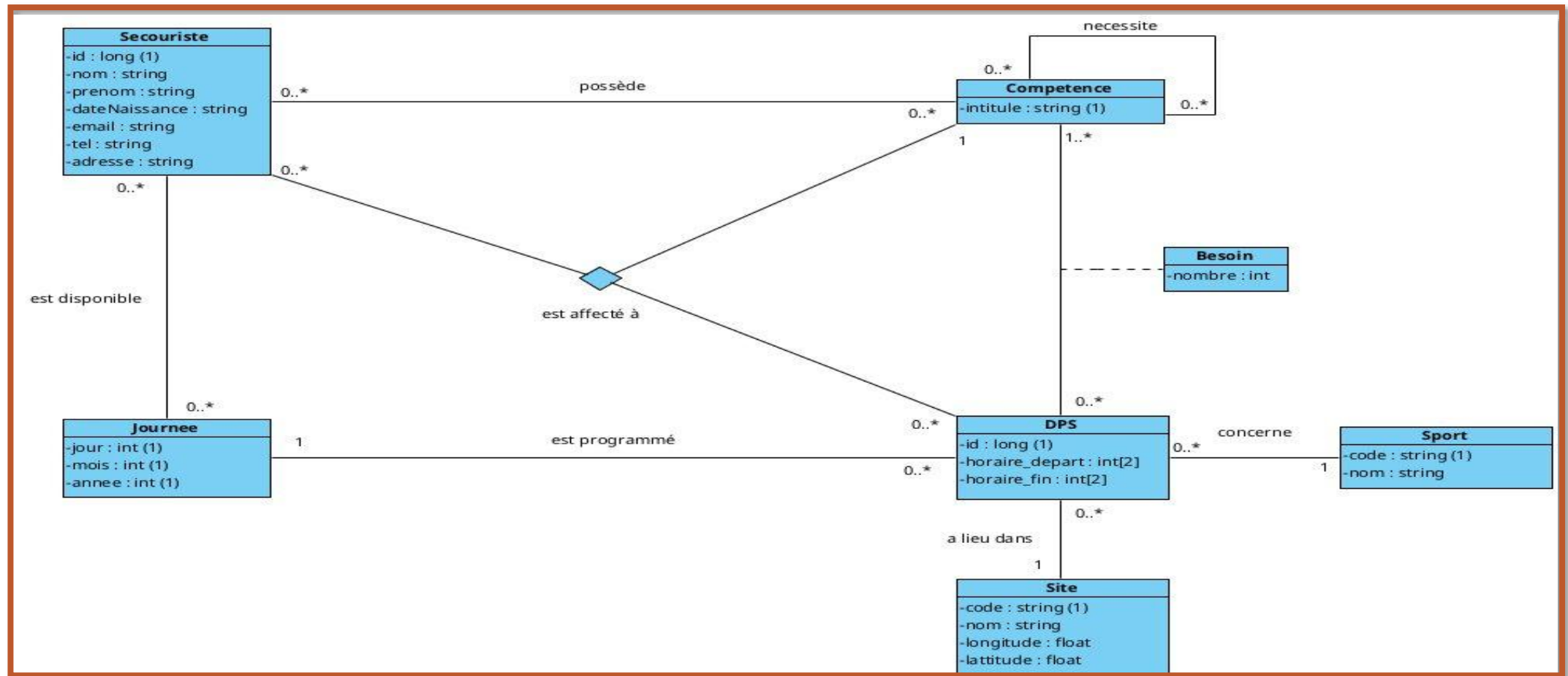
Compétences des secouristes

- ❖ **CO** : Cadre Opérationnel
- ❖ **CP** : Chef de Poste
- ❖ **CE** : Chef d'Équipe
- ❖ **PBC** : Pilote Bateau Côtier
- ❖ **PBF** : Pilote Bateau Fluvial
- ❖ **PSE1 / PSE2** : Premiers Secours en Équipe niveaux 1 et 2
- ❖ **SSA** : Secouriste, Sauveteur Aquatique
- ❖ **VPSP** : Conducteur de Véhicule de Premier Secours à Personnes

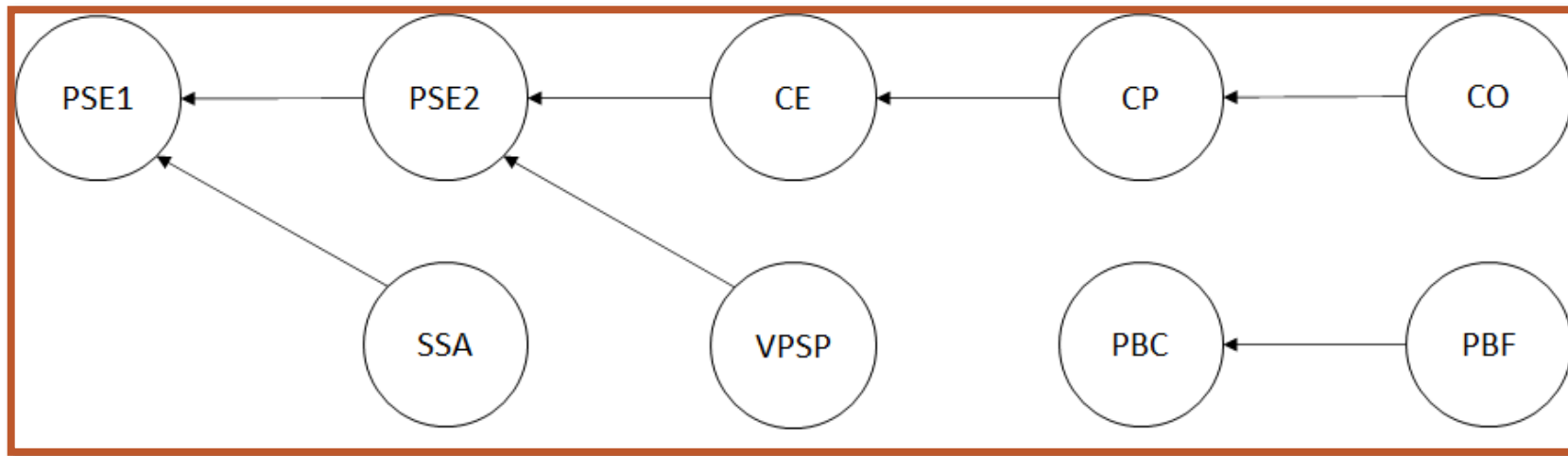
Perspectives pour 2030

- ❖ Forte des enseignements tirés des JO 2024, la Fédération des Secouristes s'est tournée vers les étudiants de **BUT1** pour développer un logiciel d'affectation des secouristes
- ❖ **Objectif** : Concevoir une solution automatisée et optimisée pour organiser efficacement les équipes de secours lors des JO d'hiver 2030

Domaine de l'application (base de données)



Prérequis des compétences



3 SAÉ de S2

SAÉ	Titre	Descriptif	Mots-clés	Ressources
S2.01	Développement d'une application	Création d'une application simple avec interface graphique	Concevoir Coder Tester Intégrer	R2.01 (POO) R2.02 (IHM) R2.03 (Qualité) R2.13 (Comm.)
S2.02	Exploration algorithmique d'un problème	Exploration et comparaison de plusieurs solutions algorithmiques	Explorer Comparer Optimiser	R2.07 (Graphe) R2.08 (Stats) R2.09 (M. Num.)
S2.03	Installation de services réseaux	Préparation d'un serveur en installant des services réseau	Installer Configurer Déployer Communiquer	R2.04 (Bas niveau) R2.05 (Services réseau) R2.12 (Anglais) R2.13 (Comm.)

Planning S2.01 : Développement d'une application

- ❖ Créneaux **R2.02** en **P3** (T. Morineau)
- ❖ **3 h** en **R2.01** lors de la **semaine 12** (R. Fleurquin, JF Kamp, C. Tibermacine)
- ❖ **6 h** en autonomie lors de la **semaine 14**
- ❖ **3 h** en **R2.02** lors des **semaines 21, 22 et 23** (A. Belmouhcine, H. Evenas)
- ❖ **1,5 h** en **R2.01** lors de la **semaine 22** (R. Fleurquin, JF Kamp, C. Tibermacine)
- ❖ **1.5 h** encadrées lors de la **semaine 23** (A. Belmouhcine, JF Kamp)
- ❖ **1.5 h** encadrées lors de la **semaine 25** (A. Belmouhcine, JF Kamp)
- ❖ **3 h** en autonomie lors de la **semaine 25**

Planning S2.02 : Exploration algorithmique d'un problème

- ❖ **45 min** de présentation lors de la **semaine 18 (A. Belmouhcine)**
- ❖ Un créneau en **R2.07** lors de la semaine **20 (T. Ferragut, T. Godin)**
- ❖ **1,5 h** encadrées lors de la semaine **23 (A. Belmouhcine)**
- ❖ **1,5 h** encadrées lors de la semaine **25 (A. Belmouhcine)**
- ❖ **4,5 h** en autonomie lors de la semaine **25**

Planning S2.03 : Installation de services réseau

- ❖ **45 min** de présentation lors de la **semaine 17** (A. Belmouhcine)
- ❖ **3 h** encadrées **entre** les **semaines 17** et **19** (E. Staddon)
- ❖ **4,5 h** en autonomie **entre** les semaines **19** et **23**



Gros coefficients

S2.01 : 38% de la C1 au S2 (Réaliser)

S2.02 : 38% de la C2 au S2 (Optimiser)

S2.03 : 38% de la C3 au S2 (Administrer)

Vos missions

- ❖ Développer une application de bureau avec interface graphique
- ❖ Permettre aux secouristes de visualiser leurs plannings et affectations
- ❖ Offrir aux administrateurs la possibilité de gérer efficacement les affectations des secouristes
- ❖ Concevoir une application en respectant le modèle MVC (Modèle-Vue-Contrôleur)

S2.01 : Application avec IHM

❖ **Fonctionnalités pour les secouristes:**

- Permettre aux secouristes de visualiser leurs plannings et affectations
- Permettre aux secouristes de mettre à jour leurs disponibilités

❖ **Fonctionnalités pour les administrateurs :**

- Visualisation des affectations de tous les secouristes
- Gestion des dispositifs de secours (DPS) : création, mise à jour et suppression
- Gestion des secouristes et de leurs compétences : ajout, suppression et modification.
- Affectation des secouristes pour chaque DPS

1. Premier livrable de la partie IHM (R2.02) : maquette de l'interface graphique (**31/03/25**)
2. Une page individuelle de contribution en **S2.01**. Chaque étudiant devra rédiger une page maximum listant les tâches qu'il a réalisées ou qu'il est entrains de réaliser dans la partie S2.01 (**13/06/25**)
3. Livrable de la partie Programmation orientée objets (R2.01) : diagrammes de conception de la partie modèle de l'application, code commenté, et jeux de tests (**19/06/25**)
4. Deuxième livrable de la partie IHM (R2.02): code complet de l'application (**19/06/25**)

SAÉ S2.02 : Algorithme d'affectation et algorithme de vérification DAG

- ❖ Vérification que le graphe des compétences est un graphe orienté acyclique (DAG), garantissant ainsi la cohérence des relations entre les compétences
- ❖ Utilisation de la théorie des graphes pour l'affectation des secouristes aux dispositifs de secours
- ❖ Modélisation des secouristes et des dispositifs de secours sous forme de nœuds dans un graphe d'affectation, avec des arêtes représentant les affectations possibles en fonction des compétences, des disponibilités et des besoins des dispositifs
- ❖ Comparaison de deux algorithmes (exhaustif et glouton) de correspondance bipartite pour le graphe d'affectation

1. Une page individuelle de contribution en **S2.02**. Chaque étudiant devra rédiger une page maximum listant les tâches qu'il a réalisées ou qu'il est entrains de réaliser dans la partie **S2.02 (13/06/25)**
2. Implémentation de l'algorithme de vérification DAG (**19/06/25**)
3. Implémentations des deux algorithmes d'affectation: **exhaustif** et **glouton (19/06/25)**
4. Rapport de présentation et de comparaison des deux méthodes d'affectation (**19/06/25**)

SAÉ S2.03 : Accès à centralisé la base de données

❖ Objectifs :

- Mettre en place un réseau local pour l'application
- Mettre en place des services DNS et DHCP
- Centraliser les données de l'application dans une base de données accessible via le réseau

❖ Tâches principales :

1. Manuel technique en anglais (09/06/25)
2. Compte rendu de déploiement et de tests en anglais (09/06/25)
3. Une page individuelle de contribution en **S2.03**. Chaque étudiant devra rédiger une page maximum listant les tâches qu'il a réalisées dans la partie S2.01 (09/06/25)
4. Vidéo de démonstration en anglais de 10 minutes maximum, où chaque membre d'un groupe prend la parole de manière équitable (09/06/25)

Organisation

- ❖ Groupe de **4 étudiants** (+/- 1) d'un même groupe de TP
- ❖ Utilisation des créneaux de SAE dédiés (indiqués à l'emploi du temps) et travail personnel hors créneaux si besoin
- ❖ Travail sur le projet pendant certaines ressources
- ❖ Les livrables devront être remis sur la plateforme Moodle en un exemplaire par groupe. Sauf pour les pages de contributions individuelles, qui devront être remises par tout le monde
- ❖ Les noms de tous les membres du groupe doivent être présents sur chacun des livrables
- ❖ Soutenance de projet le **20/06** !

Organisation: conseils

- ❖ Certaines ressources dont vous avez besoin (javafx ou graphe par exemple) commenceront en mars/avril : avancez progressivement en commençant par faire ce que vous pouvez faire (maquette IHM, conception orientée objet ...)
- ❖ Notez que les livrables n'ont pas tous la même date de rendu
- ❖ Inscrivez-vous dès maintenant sur le Moodle de **S2.01** et notez la composition de vos groupes
- ❖ **Groupes à constituer impérativement avant le lundi 27/01!**



Soutenance

- ❖ Pendant la soutenance, vous devez tous prendre la parole équitablement, répondre individuellement aux questions sur votre projet et réaliser une démonstration collective de votre application
- ❖ Des questions/réponses en anglais sont à prévoir pour la partie S2.03
- ❖ La soutenance donnera lieu à 3 notes :
 - Note individuelle de présentation. Chaque membre du groupe doit impérativement prendre la parole
 - Note individuelle de réponses aux questions
 - Note de groupe pour la démo

Evaluation

Notes de la contribution individuelle dans la SAE (**coeff. 2**), des livrables de Programmation Orientée Objets (**coeff. 2**) et d'IHM (**coeff. 3**), de la soutenance (**coeff. 2**) et de la démonstration, partie application (**coeff. 1**)

Notes de la contribution individuelle dans la SAE (**coeff. 2**), du livrable de la partie Graphe (**coeff. 3**), de la soutenance (**coeff. 2**), de la démonstration, partie affectation/vérification de DAG (**coeff. 1**), ainsi que des commandes de compilation et d'exécution de l'application (**coeff. 1**)

Notes de la contribution individuelle dans la SAE (**coeff. 2**), du manuel technique en anglais (**coeff. 1**), du compte rendu de déploiement et de tests en anglais (**coeff. 1**), ainsi que de la vidéo d'explication et de démonstration en anglais (**coeff. 1**).

La vidéo doit durer 10 minutes maximum, et chaque membre du groupe doit disposer d'un temps de parole à peu près équivalent

SAÉ S2.03 : Installation de services réseau

ACCÈS CENTRALISÉ LA BASE DE DONNÉES

Rappel - SAÉ S2.03 : Accès centralisé à la base de données

❖ Objectifs :

- Mettre en place un réseau local pour l'application
- Mettre en place des services DNS et DHCP
- Centraliser les données de l'application dans une base de données accessible via le réseau

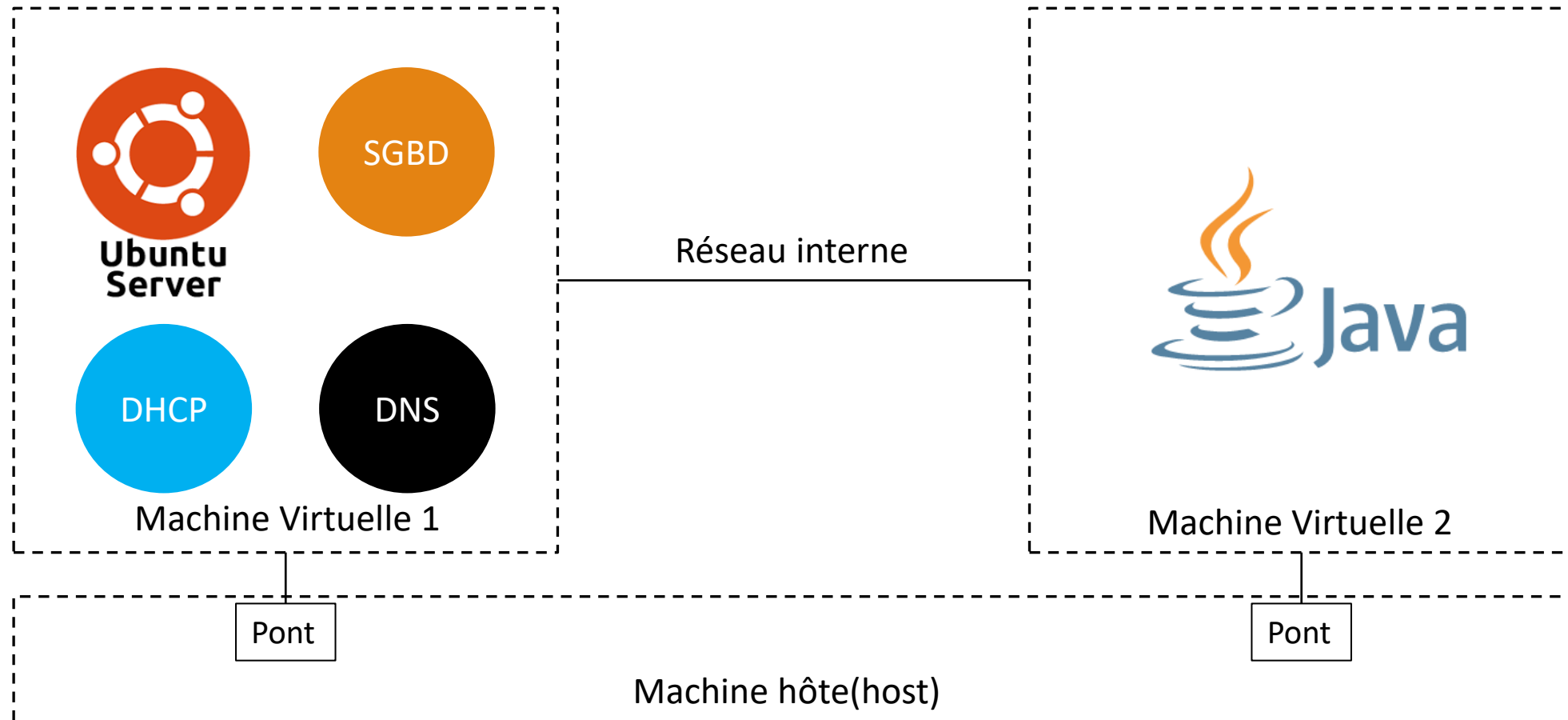
❖ Tâches principales :

1. Manuel technique en anglais (09/06/25)
2. Compte rendu de déploiement et de tests en anglais (09/06/25)
3. Une page individuelle de contribution en **S2.03**. Chaque étudiant devra rédiger une page maximum listant les tâches qu'il a réalisées dans la partie S2.01 (09/06/25)
4. Vidéo de démonstration en anglais de 10 minutes maximum, où chaque membre d'un groupe prend la parole de manière équitable (09/06/25)

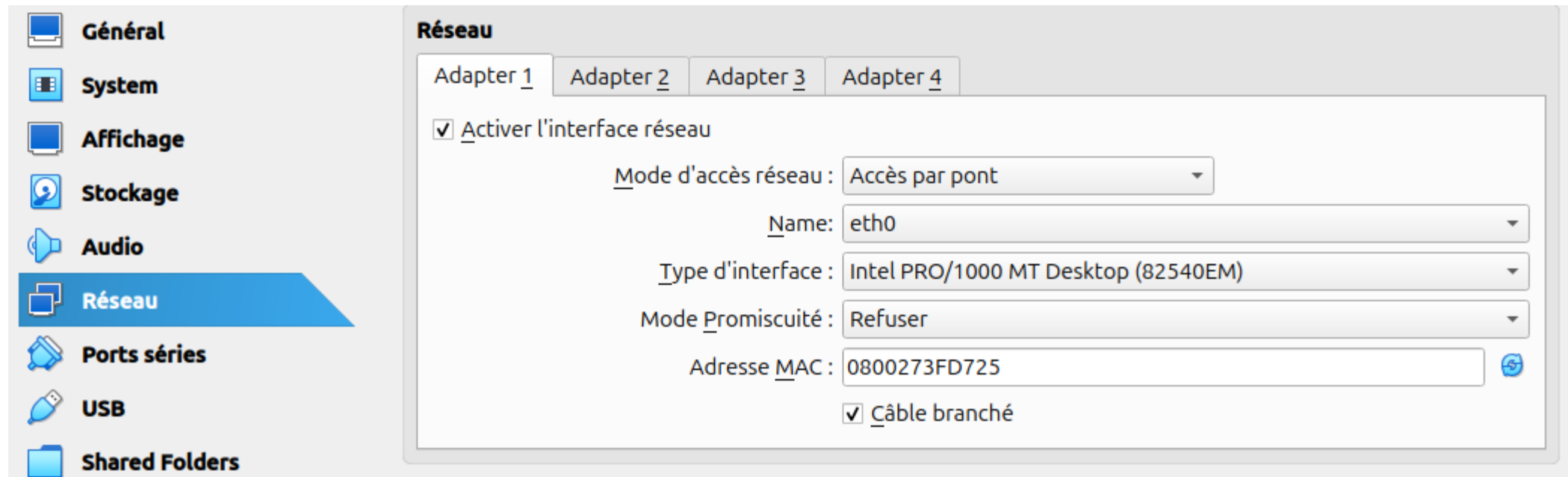
Rappel - Planning S2.03 : Installation de services réseau

- ❖ **45 min** de présentation lors de la **semaine 17 (A. Belmouhcine)**
- ❖ **3 h** encadrées **entre** les **semaines 17 et 19 (E. Staddon)**
- ❖ **4,5 h** en autonomie **entre** les semaines **19 et 23**

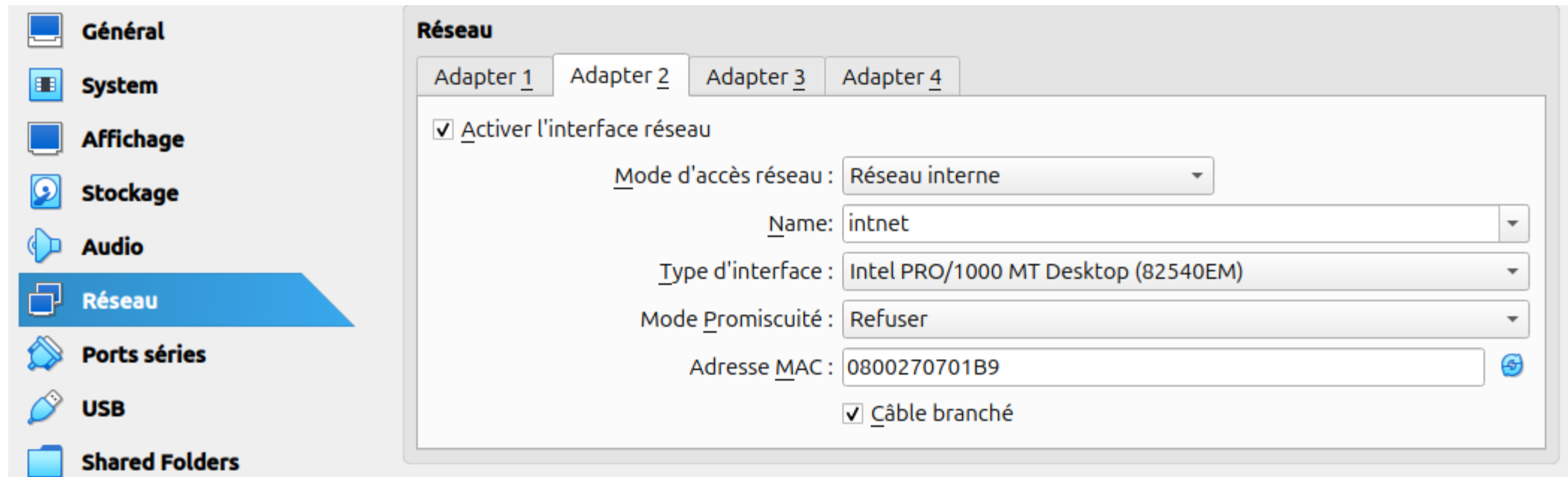
Simulation de l'infrastructure réseau



Accès en mode pont à la machine hôte



Connexion entre machines virtuelles via un adaptateur interne



Configuration réseau du serveur

- ❖ Le serveur doit disposer d'une **adresse IP statique**
- ❖ **Netplan** pour configurer les deux interfaces réseau :
 - Accès à Internet via la machine hôte en mode Pont (Bridge)
 - Connexion au réseau local interne (interne aux machines virtuelles)

DNS : Domaine Name Server

- ❖ Service de résolution des noms de domaine en adresses IP
- ❖ Fonctionne en mode client/serveur via le port 53
- ❖ Permet d'obtenir l'adresse IP à partir d'un nom de domaine
- ❖ Adresse IP du serveur DNS connue par le client (station hôte)

DHCP : Dynamic Host Configuration Protocol

- ❖ Protocole réseau permettant à un hôte d'obtenir dynamiquement une adresse IP depuis un serveur DHCP lorsqu'il se connecte au réseau
- ❖ Avantages
 - Réutilisation des adresses IP : une adresse n'est conservée que pendant que l'hôte est actif
 - Prise en charge des utilisateurs mobiles qui se déplacent entre différents réseaux
 - Administration simplifiée : pas de configuration manuelle pour chaque appareil
- ❖ En plus de l'adresse IP, le serveur DHCP peut transmettre :
 - Adresse du routeur passerelle (first-hop router)
 - Nom et adresse IP du serveur DNS
 - Masque de sous-réseau (network mask)

SAÉ S2.02 : Exploration algorithmique d'un problème

ALGORITHME D'AFFECTATION ET ALGORITHME DE VÉRIFICATION
DAG

Rappel - SAÉ S2.02 : Algorithme d'affectation et algorithme de vérification DAG

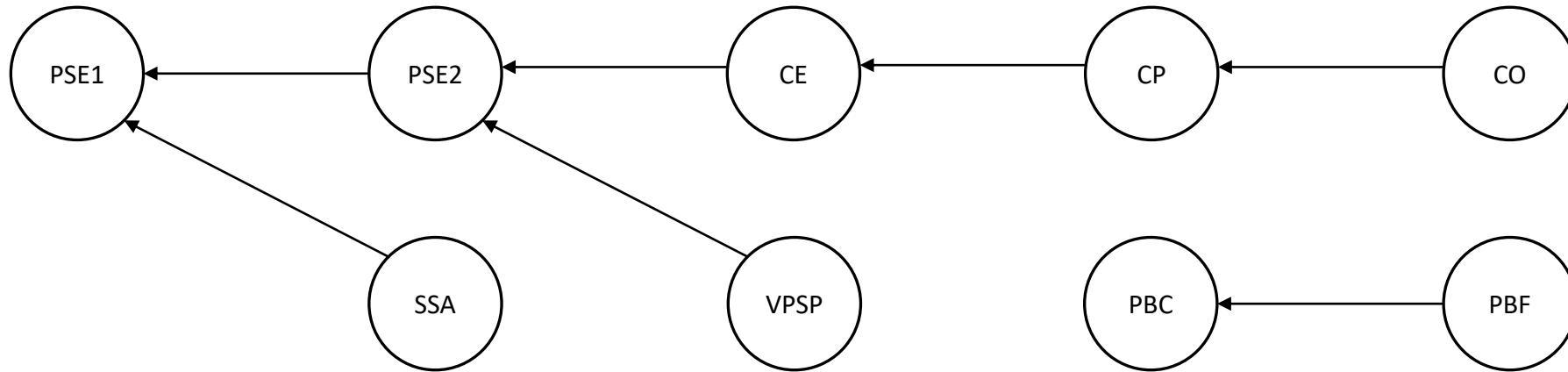
- ❖ Vérification que le graphe des compétences est un graphe orienté acyclique (DAG), garantissant ainsi la cohérence des relations entre les compétences
- ❖ Utilisation de la théorie des graphes pour l'affectation des secouristes aux dispositifs de secours
- ❖ Modélisation des secouristes et des dispositifs de secours sous forme de nœuds dans un graphe d'affectation, avec des arêtes représentant les affectations possibles en fonction des compétences, des disponibilités et des besoins des dispositifs
- ❖ Comparaison de deux algorithmes (exhaustif et glouton) de correspondance bipartite pour le graphe d'affectation

1. Une page individuelle de contribution en **S2.02**. Chaque étudiant devra rédiger une page maximum listant les tâches qu'il a réalisées ou qu'il est entrains de réaliser dans la partie **S2.02 (13/06/25)**
2. Implémentation de l'algorithme de vérification DAG (**19/06/25**)
3. Implémentations des deux algorithmes d'affectation: **exhaustif** et **glouton (19/06/25)**
4. Rapport de présentation et de comparaison des deux méthodes d'affectation (**19/06/25**)

Rappel - Planning S2.02 : Exploration algorithmique d'un problème

- ❖ **45 min** de présentation lors de la **semaine 18 (A. Belmouhcine)**
- ❖ Un créneau en **R2.07** lors de la semaine **20 (T. Ferragut, T. Godin)**
- ❖ **1,5 h** encadrées lors de la semaine **23 (A. Belmouhcine)**
- ❖ **1,5 h** encadrées lors de la semaine **25 (A. Belmouhcine)**
- ❖ **4,5 h** en autonomie lors de la semaine **25**

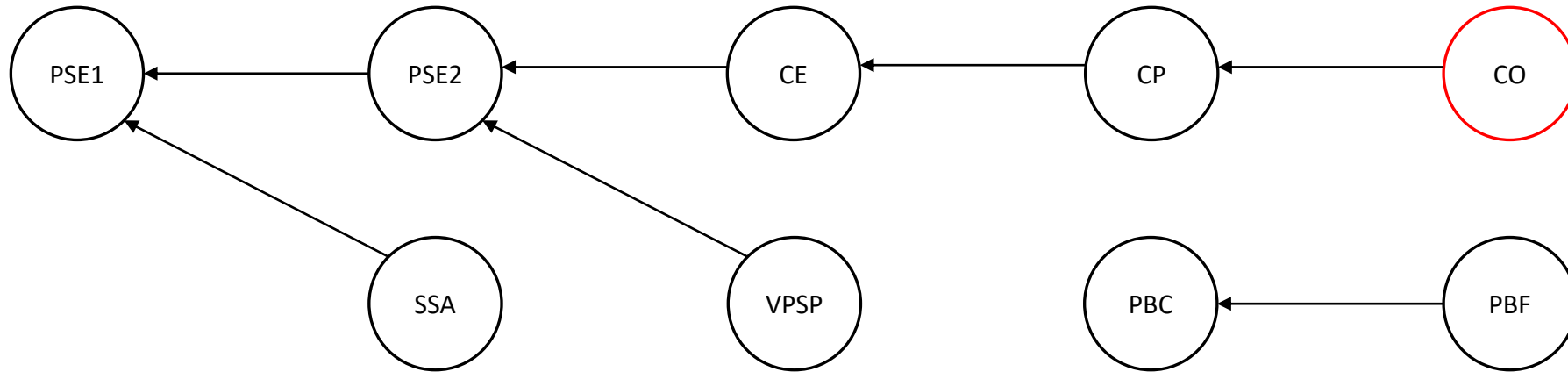
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

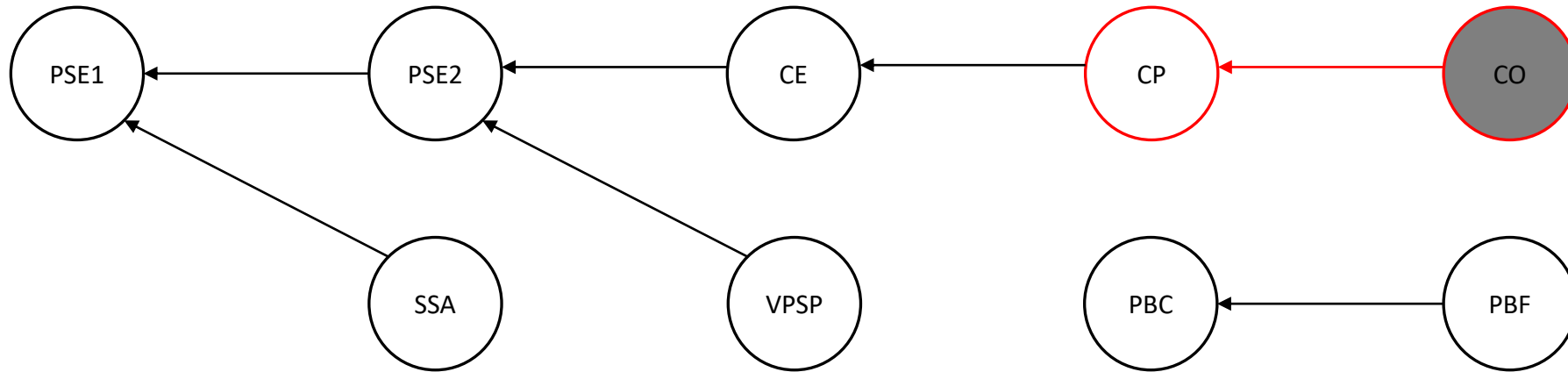
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

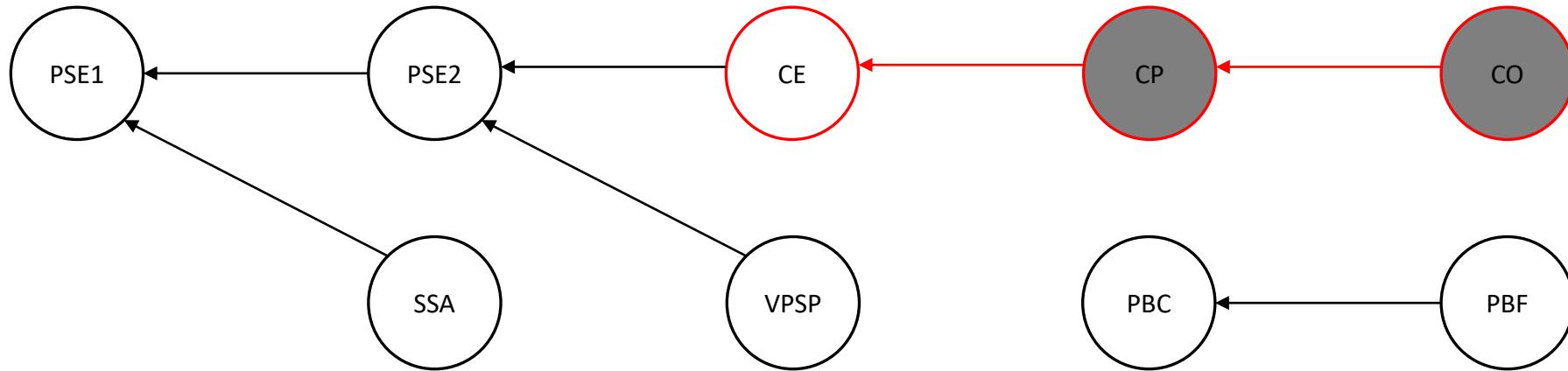
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

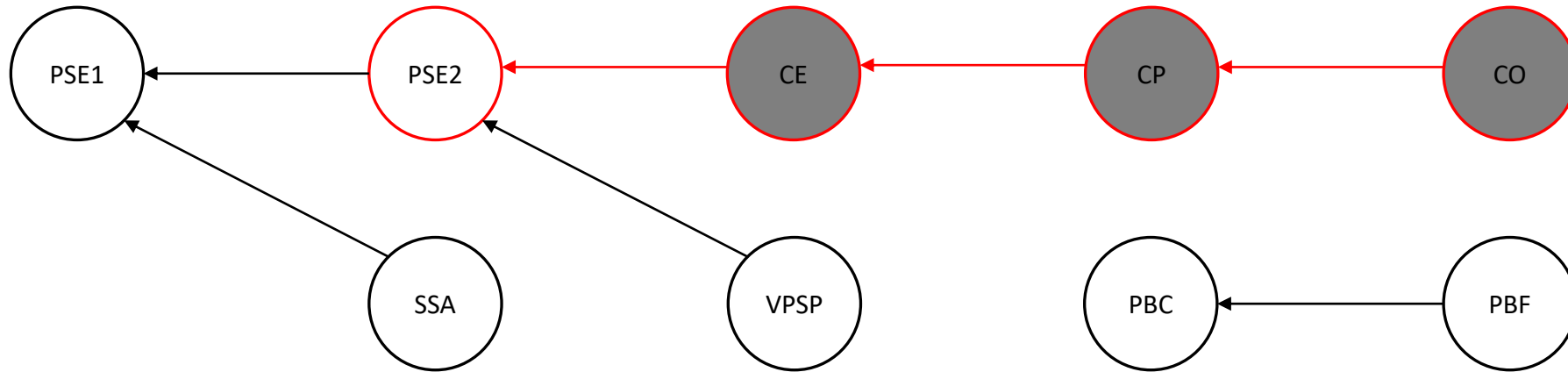
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

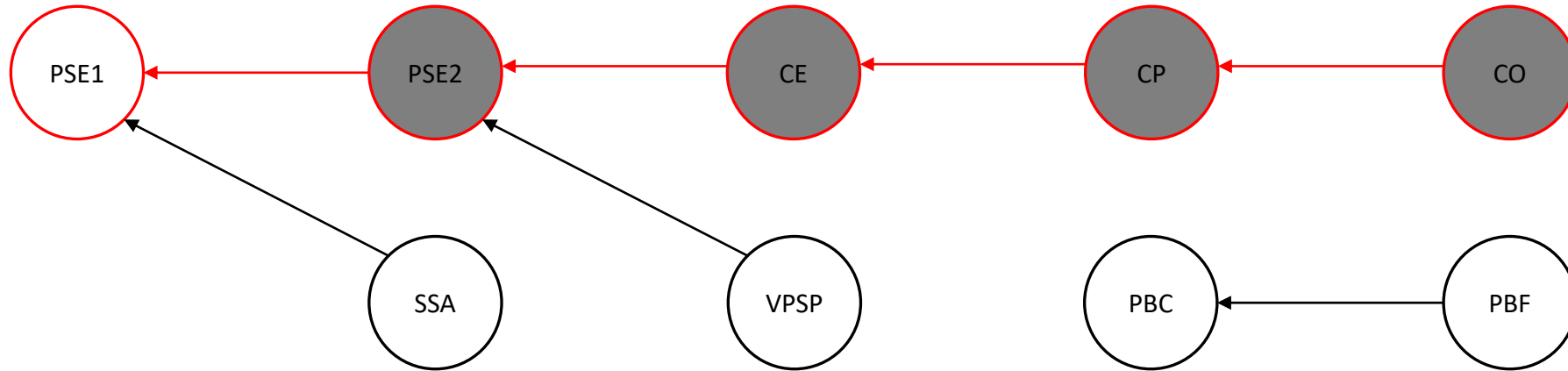
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

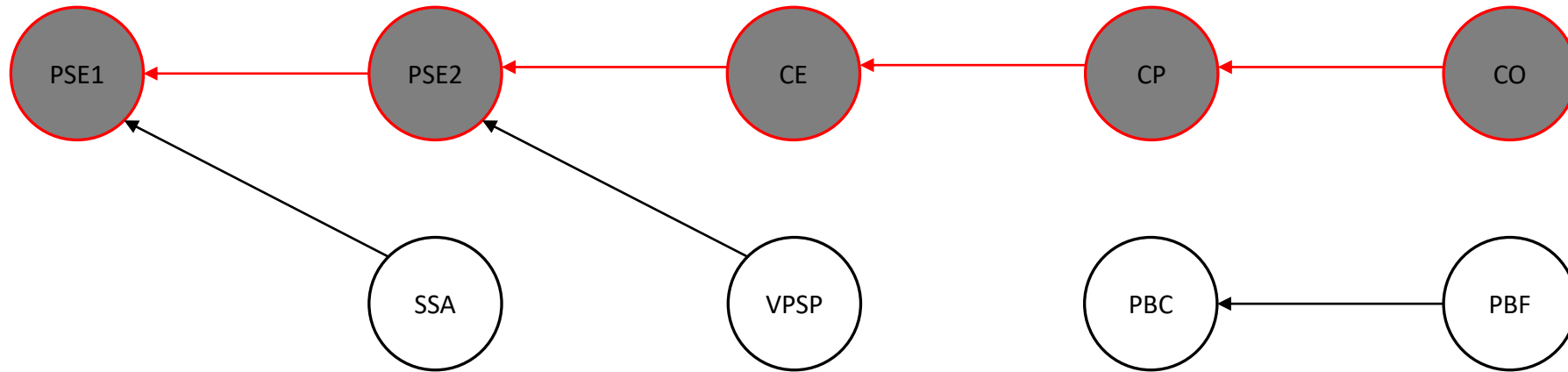
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

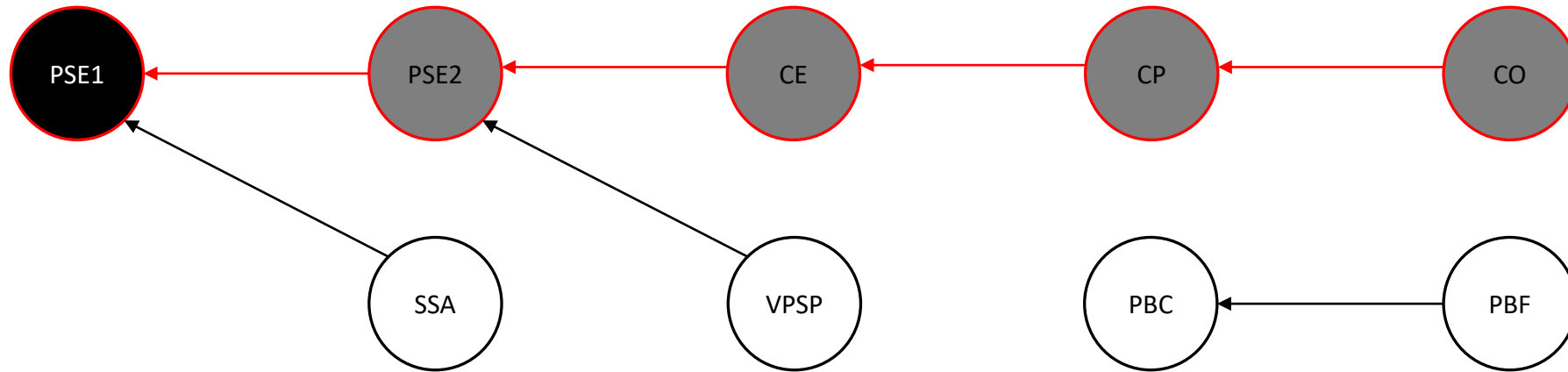
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

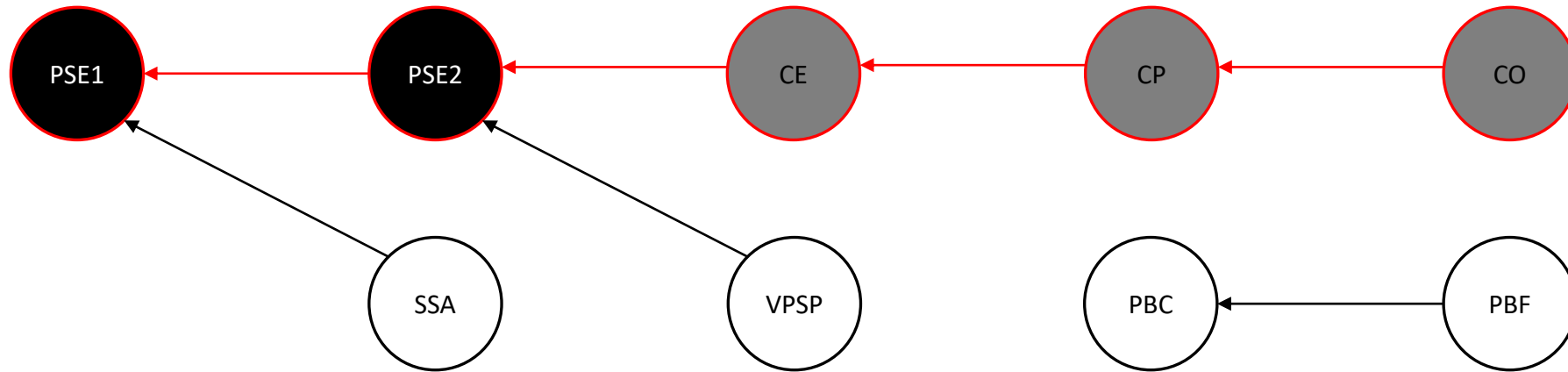
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

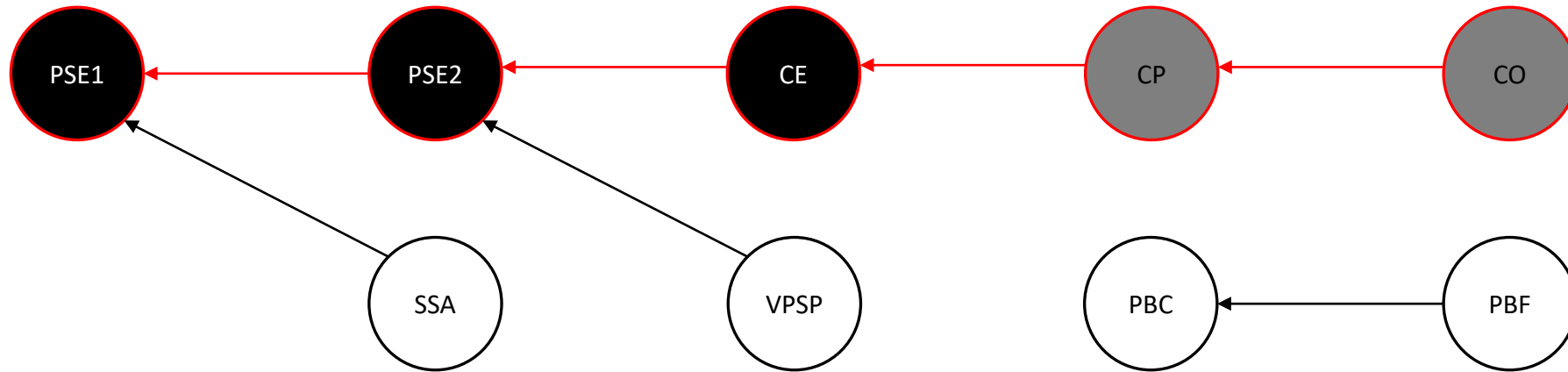
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

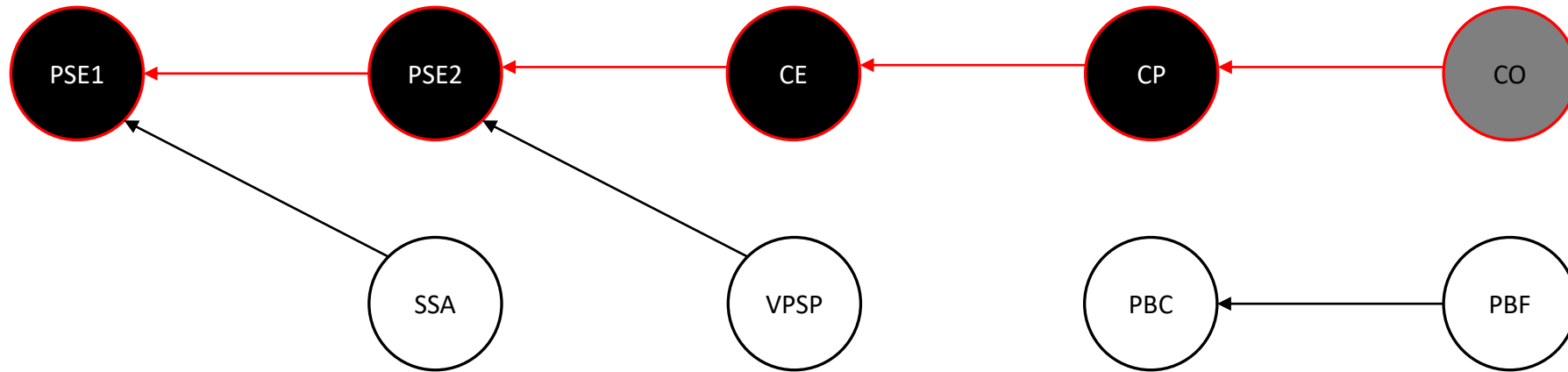
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

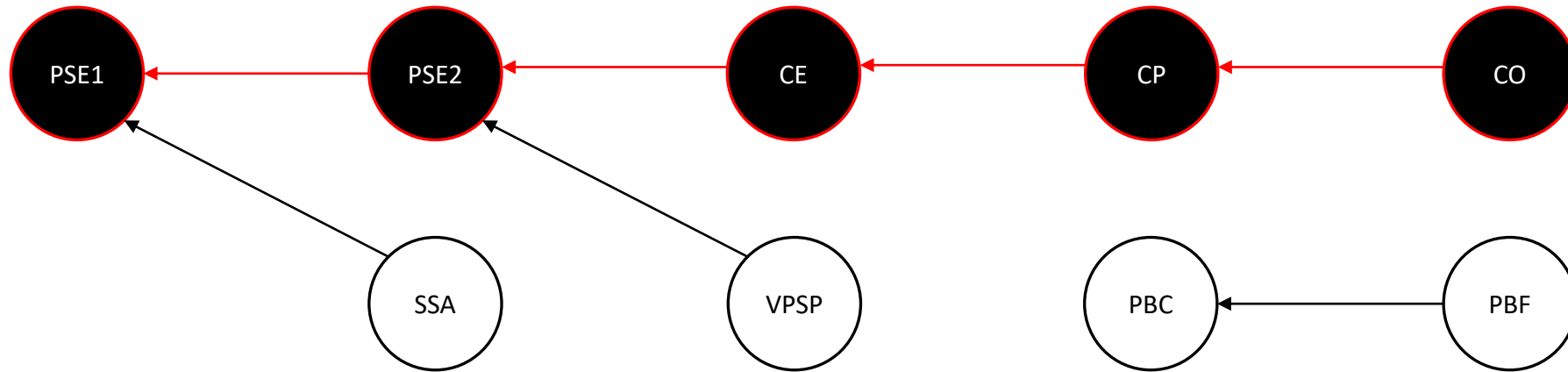
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

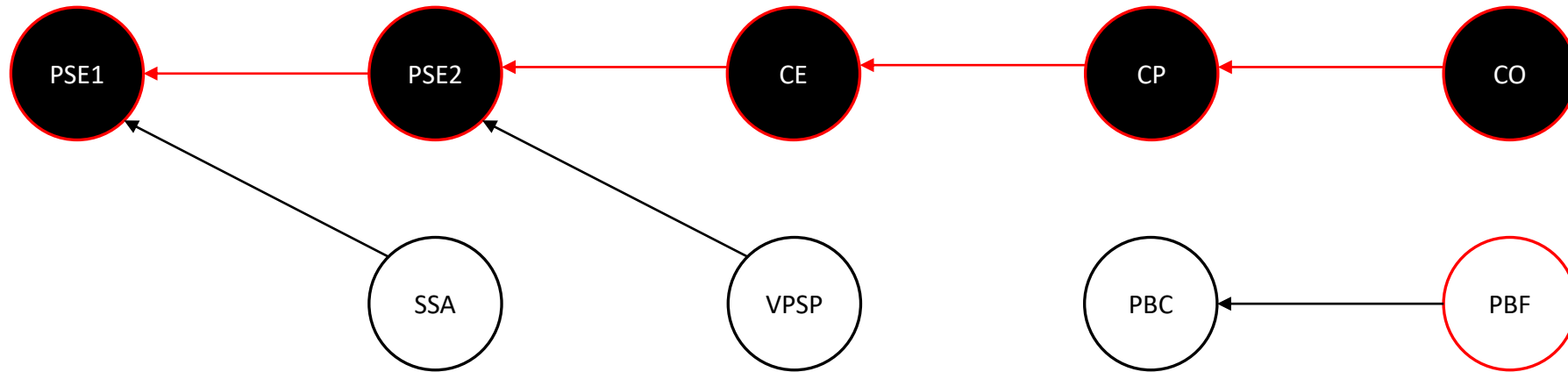
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

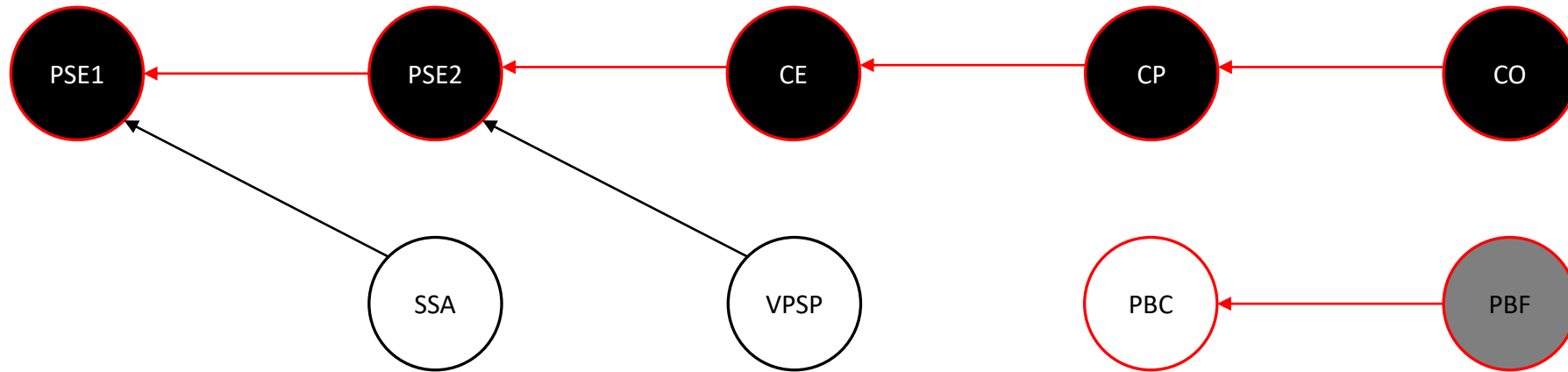
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

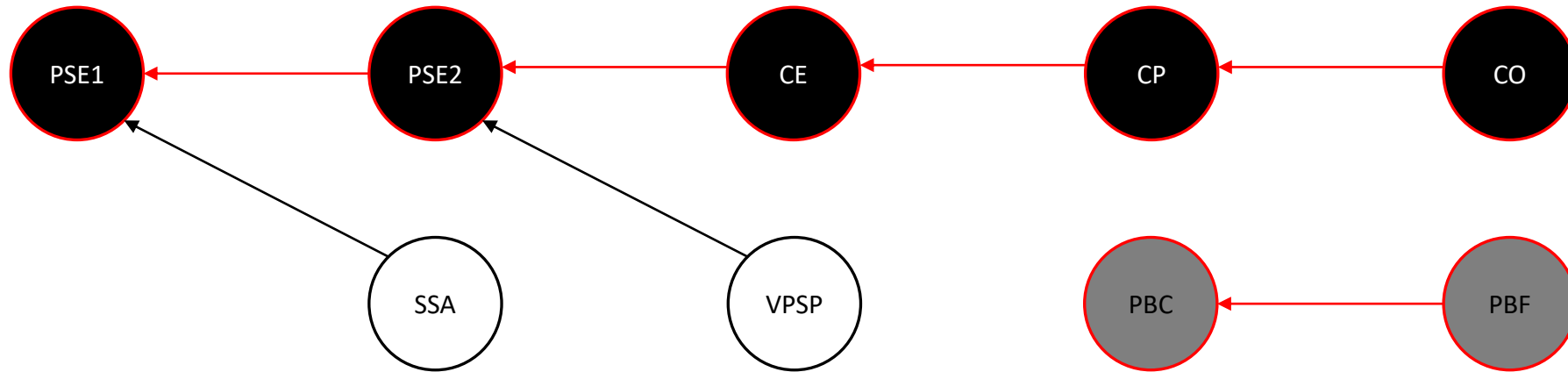
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

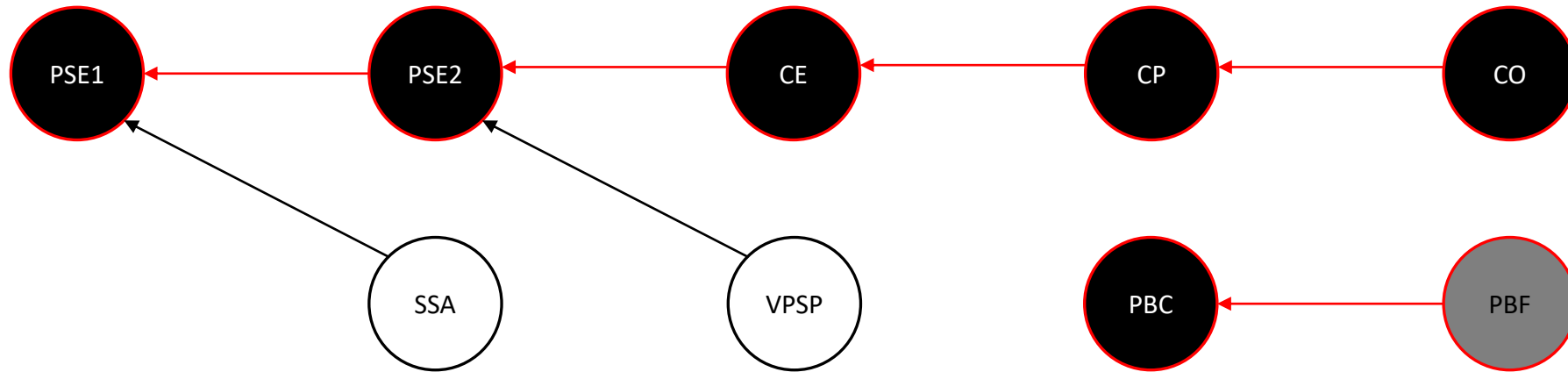
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

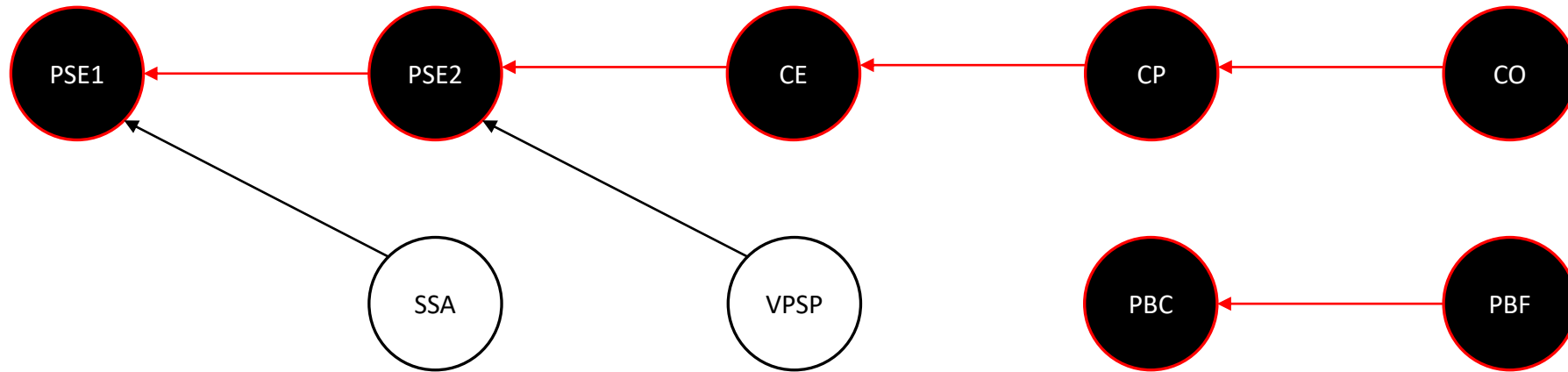
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

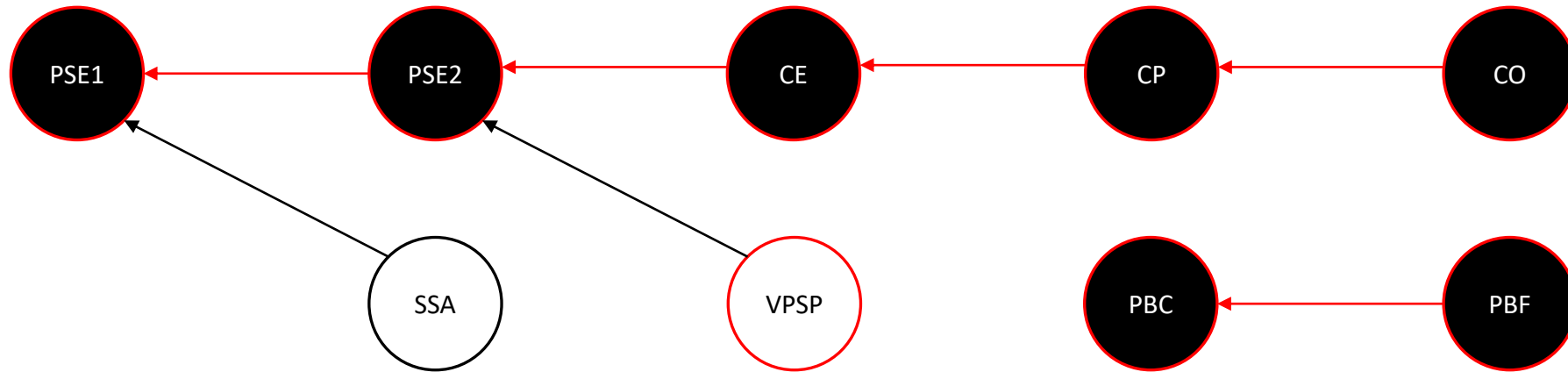
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

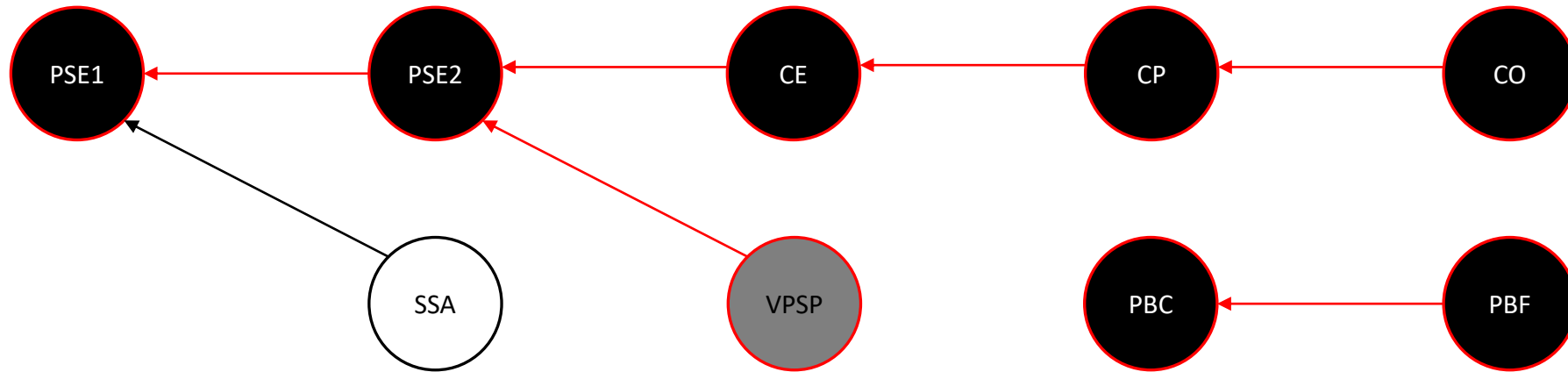
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

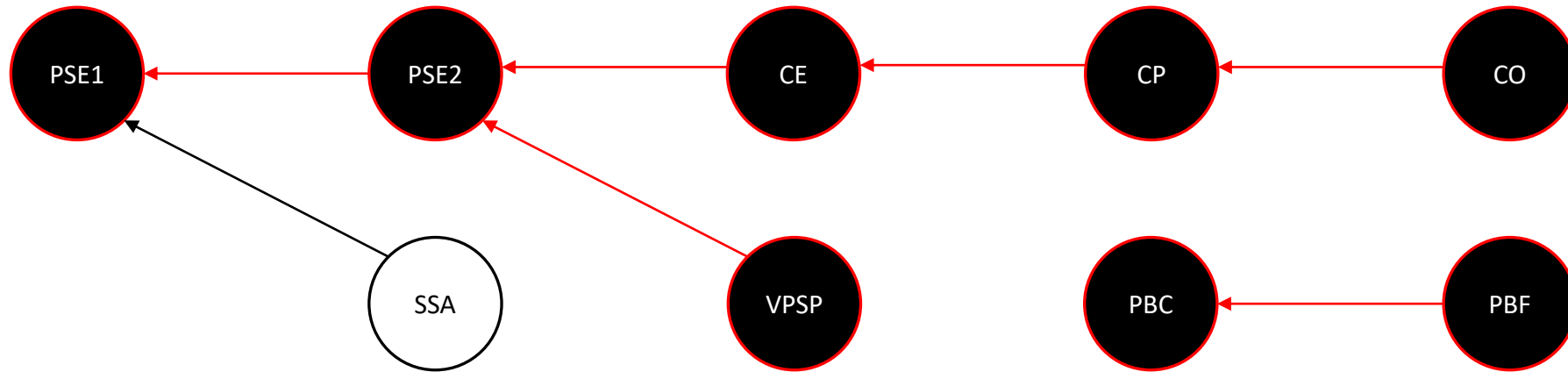
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

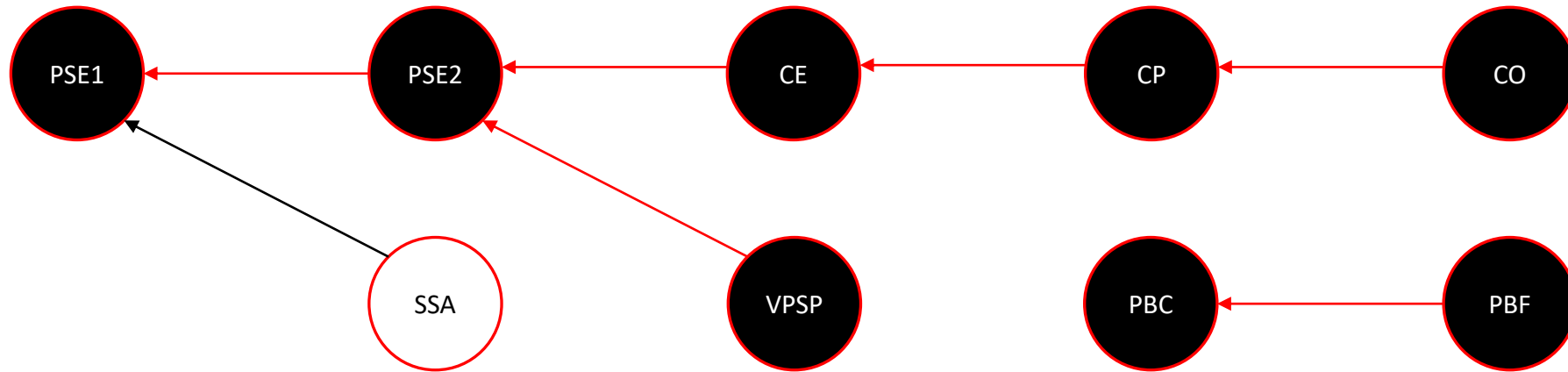
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

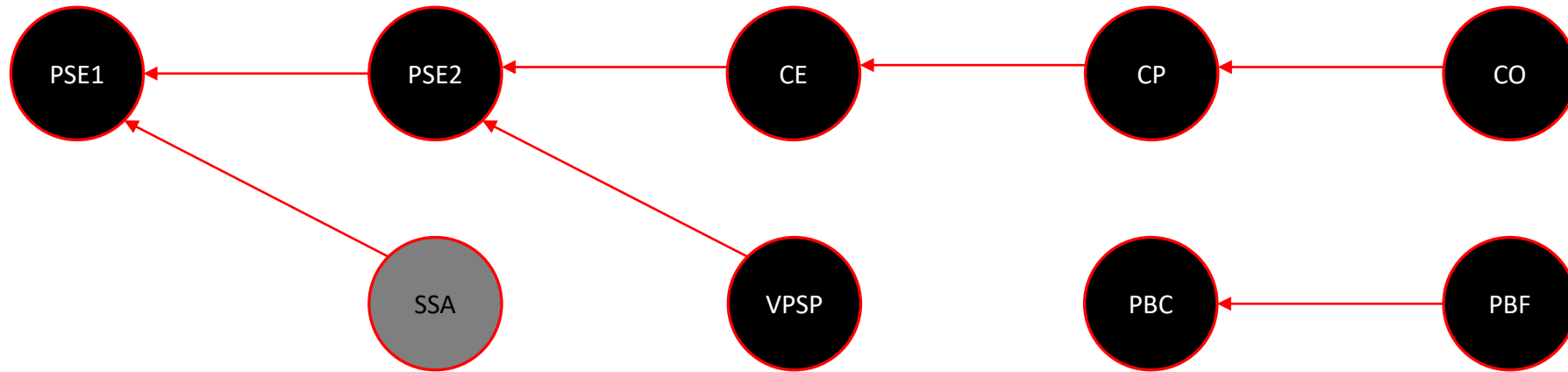
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

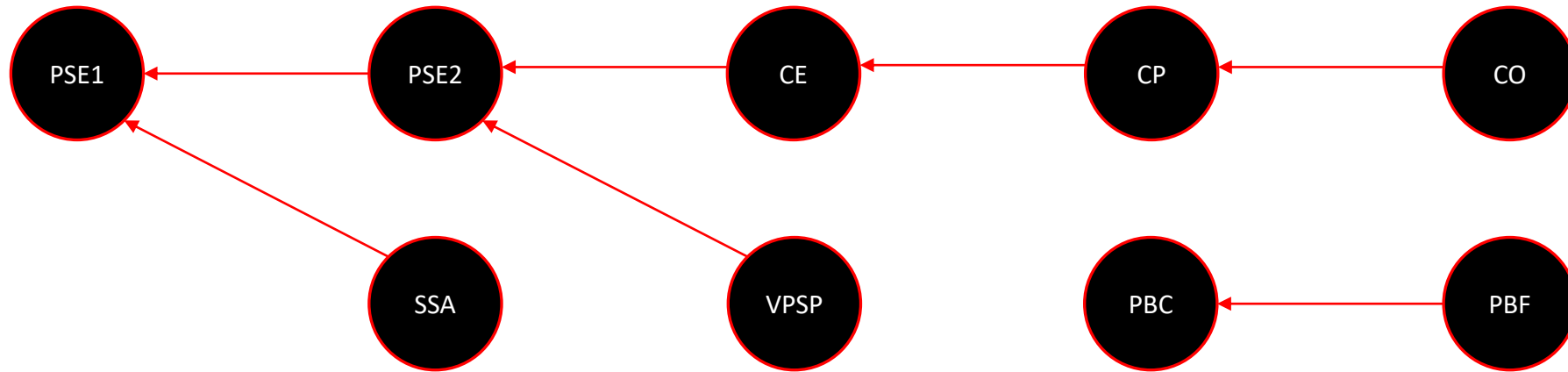
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

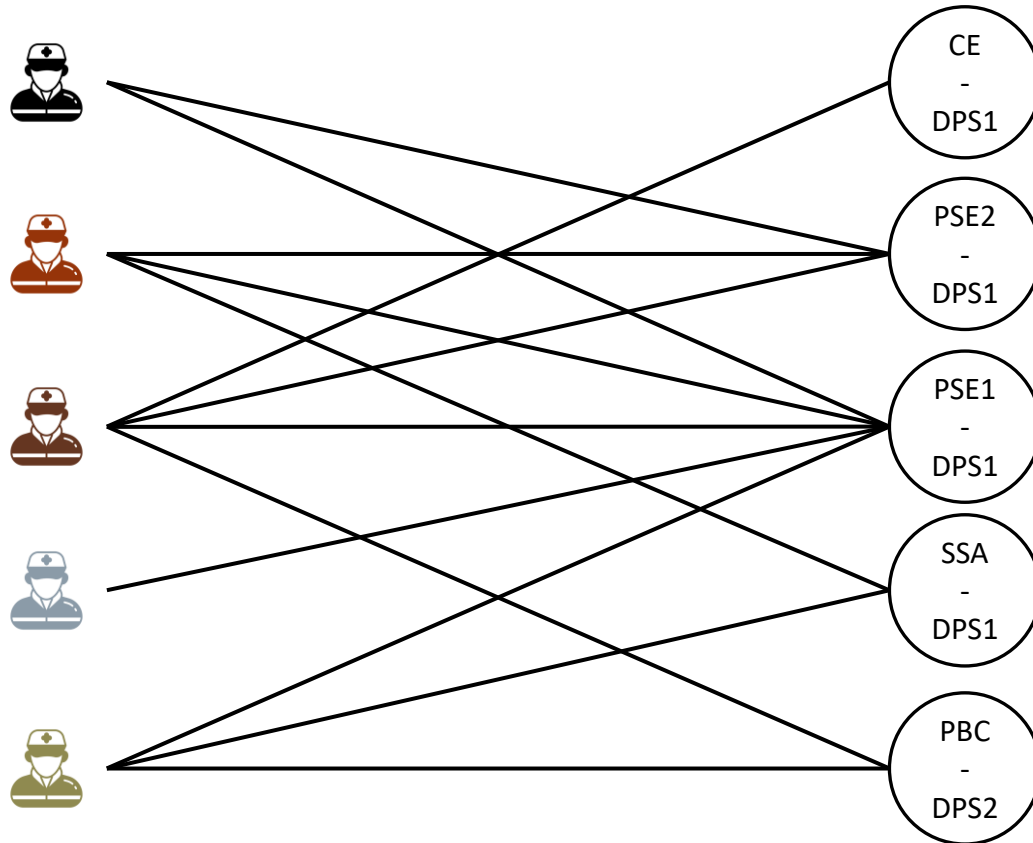
Vérification que le graphe des compétences est DAG



❖ Parcours en profondeur

- Chaque sommet a une couleur qui indique son état : **Blanc** : sommet non visité, **Gris** : sommet en cours d'exploration, **Noir** : sommet entièrement exploré
- Avant de commencer, tous les sommets sont marqués comme blancs
- Lorsqu'on commence à explorer un sommet, on le colore en gris
- Lorsqu'on a terminé l'exploration de tous ses voisins, on le colore en noir
- Si l'on découvre un sommet blanc, on commence une nouvelle exploration à partir de lui
- Si l'on rencontre un sommet noir, on n'a rien à faire (il a déjà été totalement exploré)
- Si l'on rencontre un sommet gris, cela signifie qu'on a trouvé un cycle => le graphe n'est pas un DAG

Affectation des secouristes – Graphe bipartite



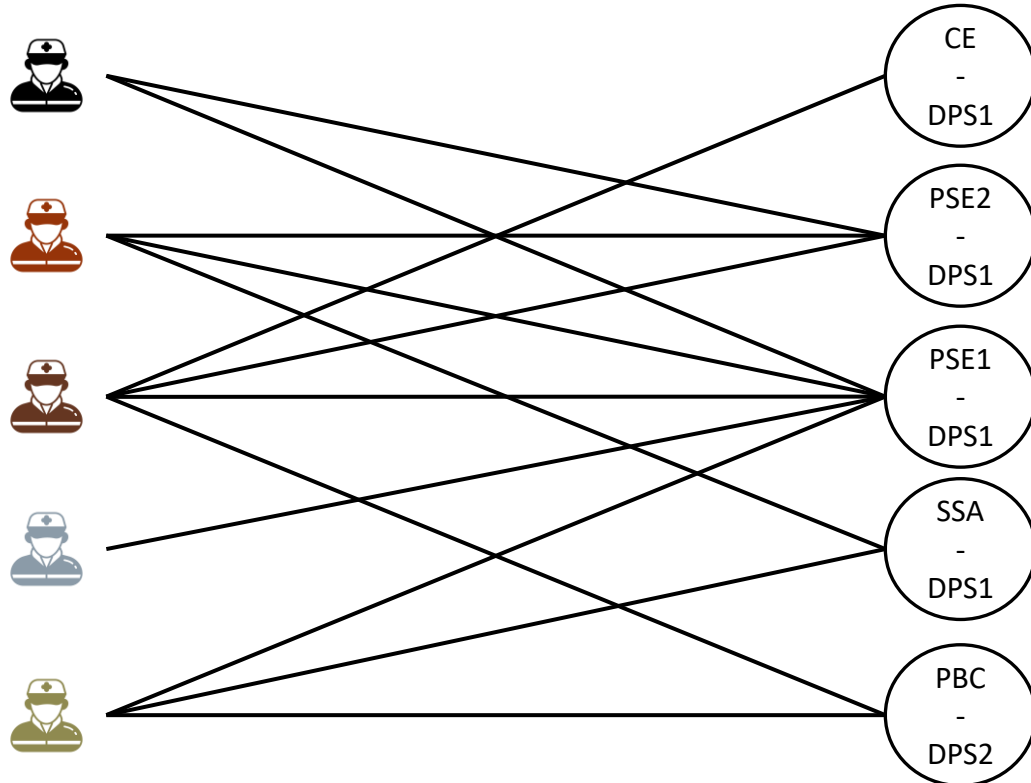
❖ L'affectation se fait par journée

❖ Définition de la matrice de coût :

- Si un secouriste n'a pas la compétence
=> coût = 0
- Si un secouriste possède la compétence
=> coût = 1
- Si la matrice de coût n'est pas carrée, il faut la compléter par des entrées fictives (secouristes ou besoins de DPS)

Stratégie d'affectation exhaustive

- ❖ Explorer toutes les affectations possibles, c'est-à-dire toutes les permutations
- ❖ Sélectionner celle qui maximise la somme des coûts

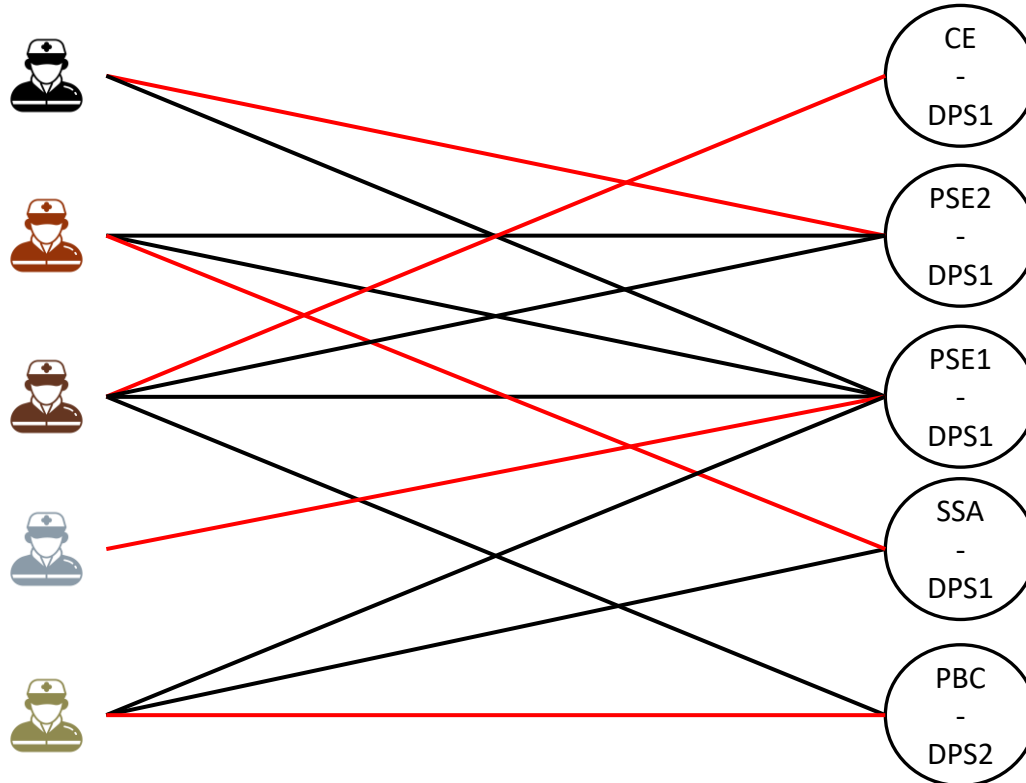


CE	PSE2	PSE1	SSA	PBC	Cout
					5
					2
					3
					2
					3
					3
					3

...

Stratégie d'affectation exhaustive

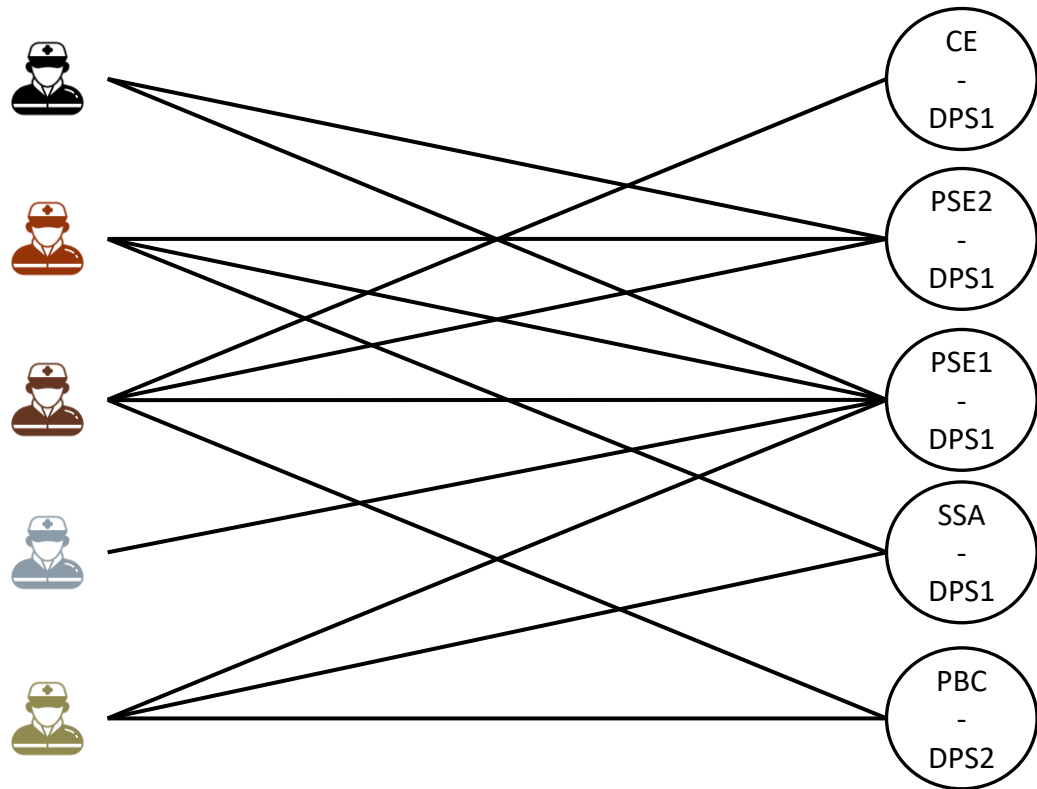
- ❖ Explorer toutes les affectations possibles, c'est-à-dire toutes les permutations
- ❖ Sélectionner celle qui maximise la somme des coûts



CE	PSE2	PSE1	SSA	PBC	Cout
					5
					2
					3
					2
					3
					3
					3

...

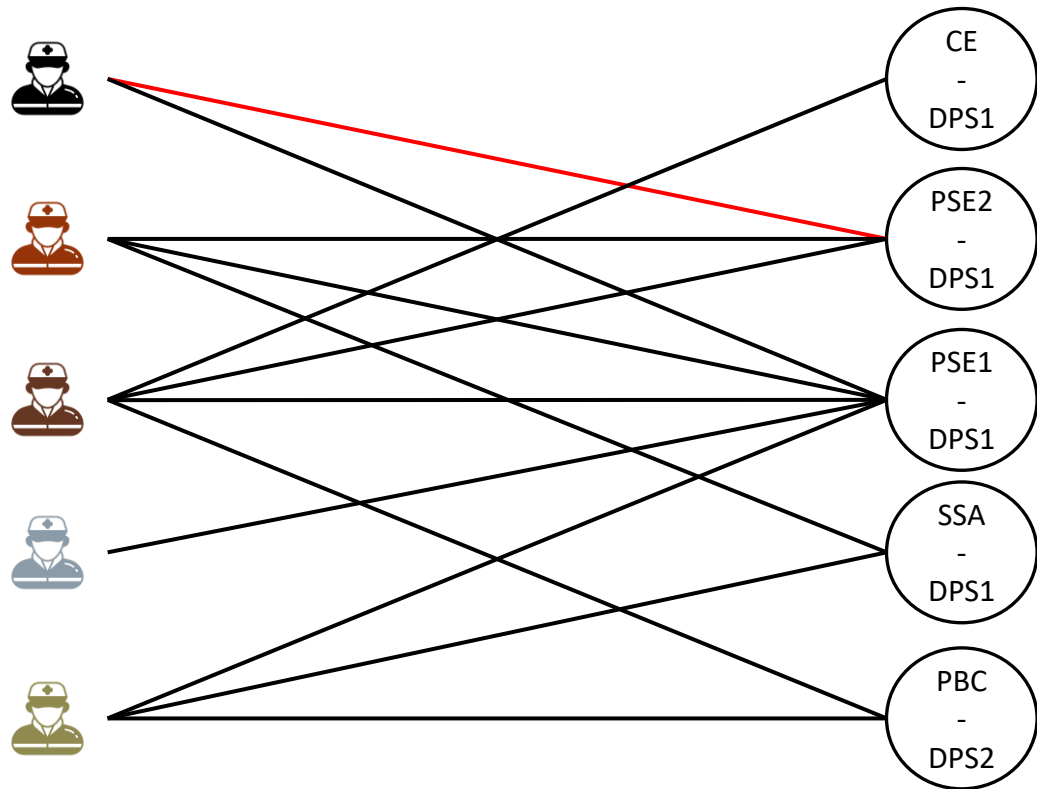
Algorithme glouton d'affectation - Version naïve



	CE	PSE2	PSE1	SSA	PBC
	0	1	1	0	0
	0	1	1	1	0
	1	1	1	0	1
	0	0	1	0	0
	0	0	1	1	1

❖ Ne garantit pas une solution optimale

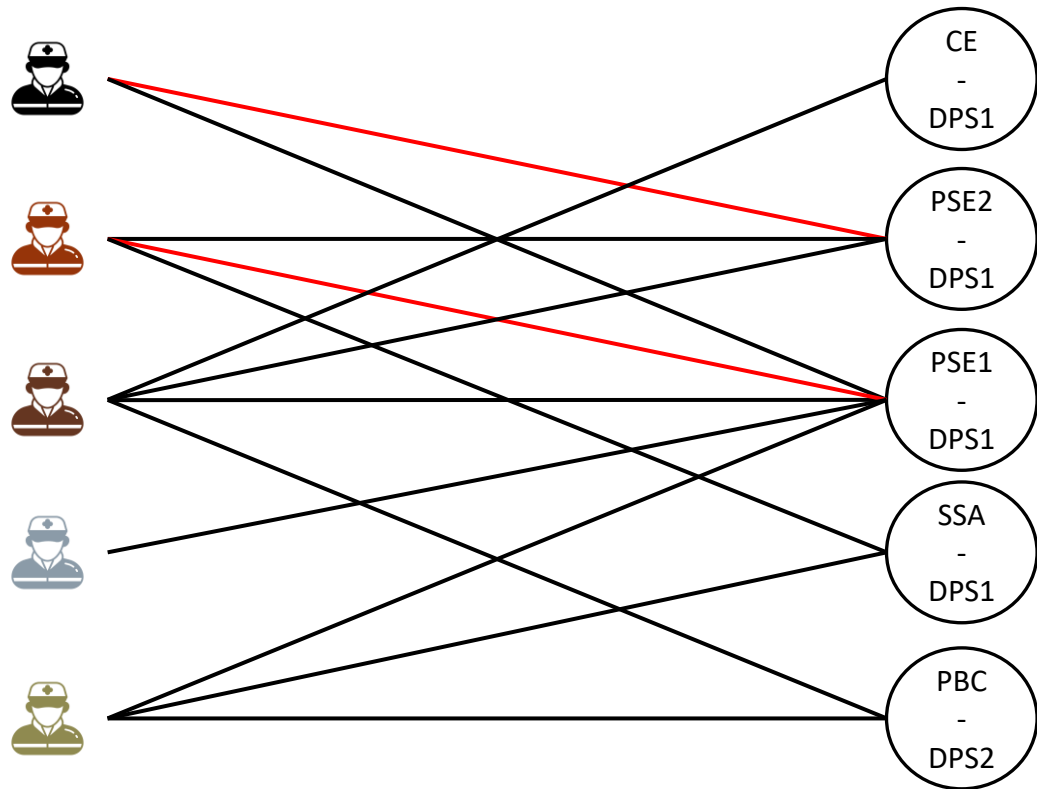
Algorithme glouton d'affectation - Version naïve



	CE	PSE2	PSE1	SSA	PBC
	0	1	1	0	0
	0	1	1	1	0
	1	1	1	0	1
	0	0	1	0	0
	0	0	1	1	1

❖ Ne garantit pas une solution optimale

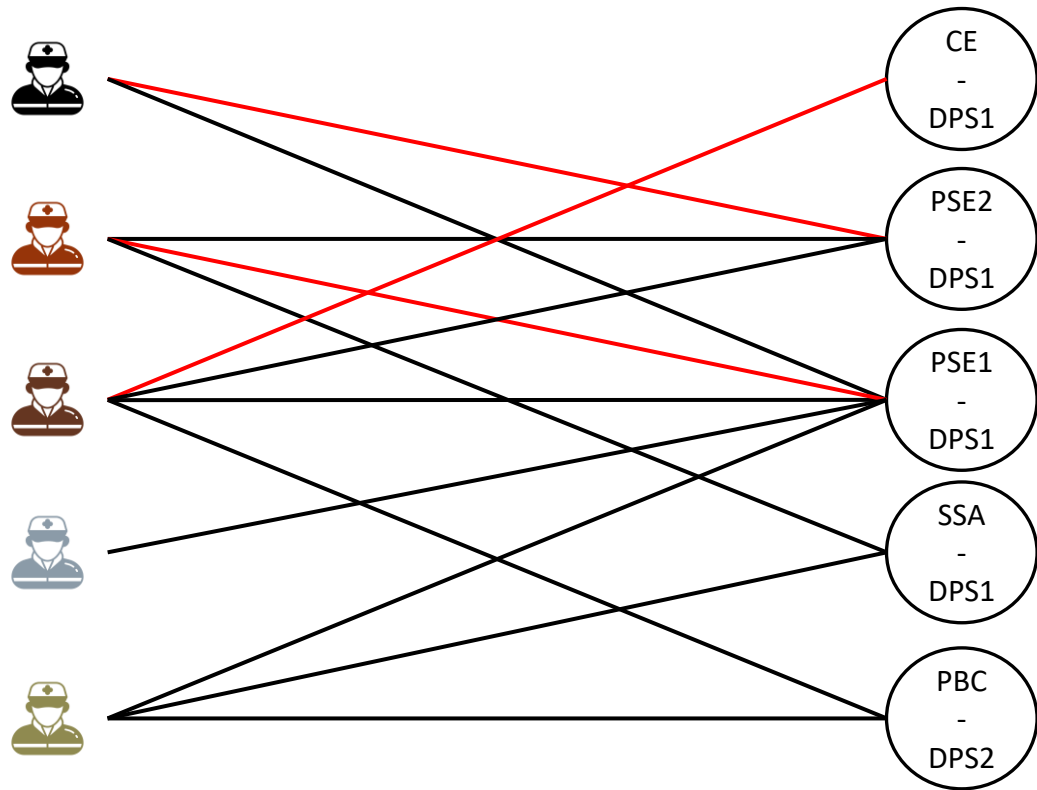
Algorithme glouton d'affectation - Version naïve



	CE	PSE2	PSE1	SSA	PBC
	0	1	1	0	0
	0	1	1	1	0
	1	1	1	0	1
	0	0	1	0	0
	0	0	1	1	1

❖ Ne garantit pas une solution optimale

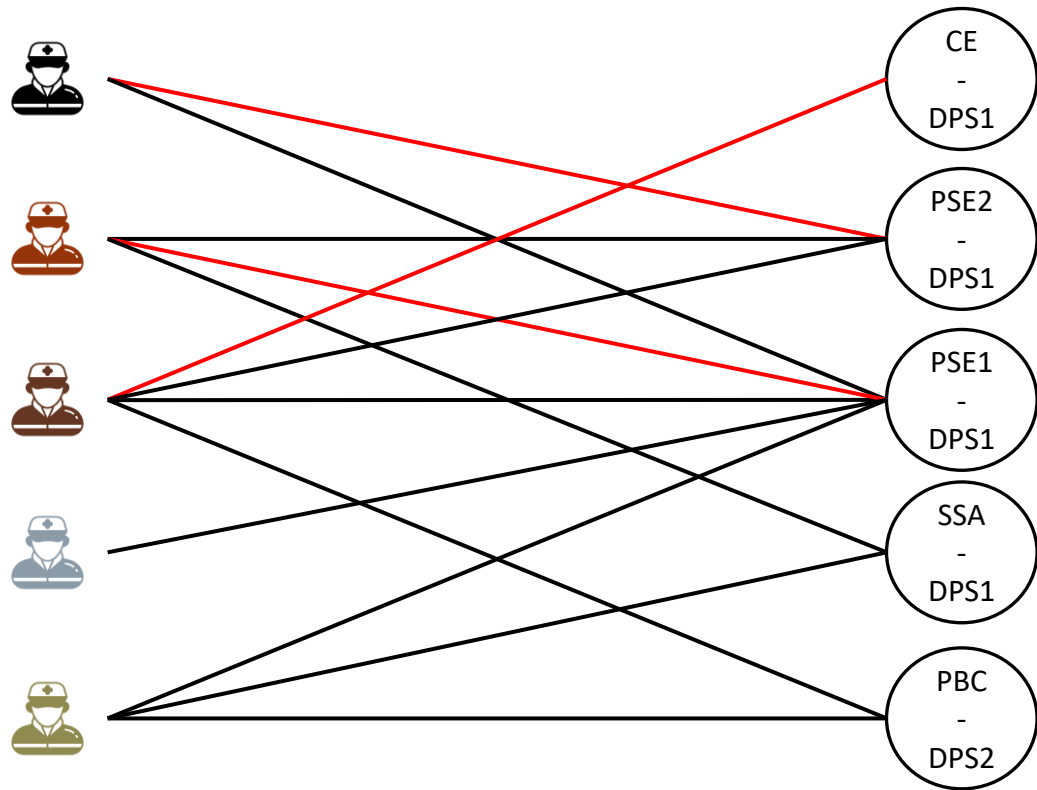
Algorithme glouton d'affectation - Version naïve



	CE	PSE2	PSE1	SSA	PBC
	0	1	1	0	0
	0	1	1	1	0
	1	1	1	0	1
	0	0	1	0	0
	0	0	1	1	1

❖ Ne garantit pas une solution optimale

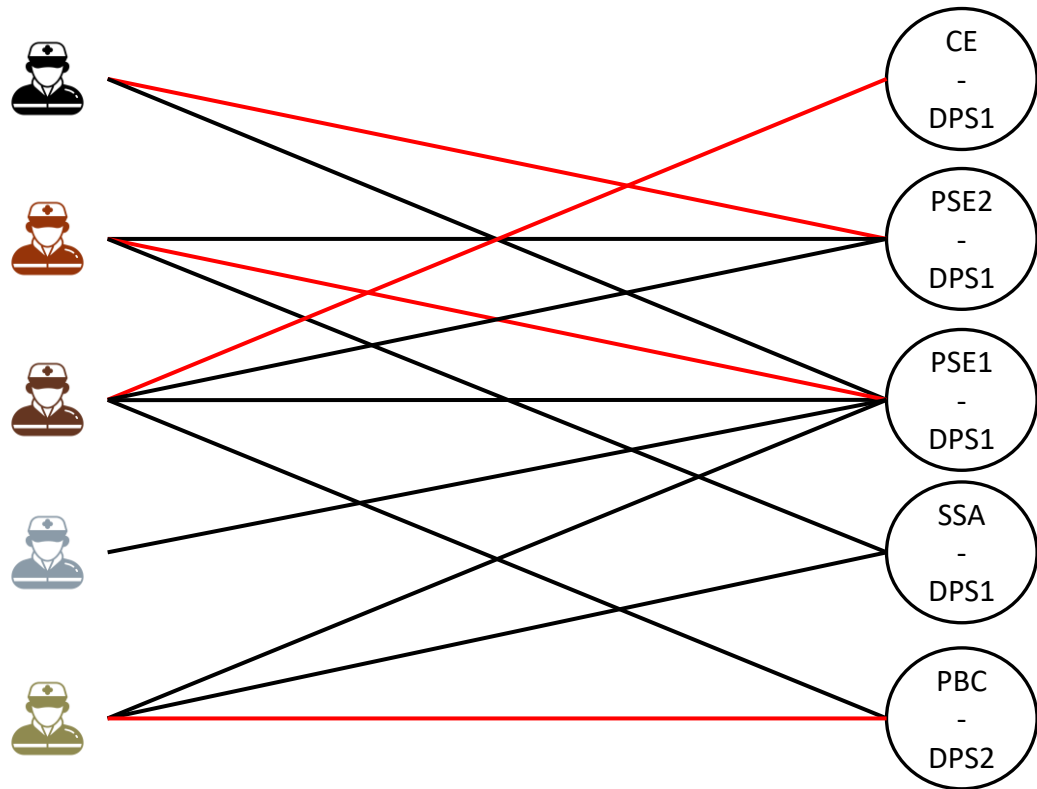
Algorithme glouton d'affectation - Version naïve



	CE	PSE2	PSE1	SSA	PBC
	0	1	1	0	0
	0	1	1	1	0
	1	1	1	0	1
	0	0	1	0	0
	0	0	1	1	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - Version naïve

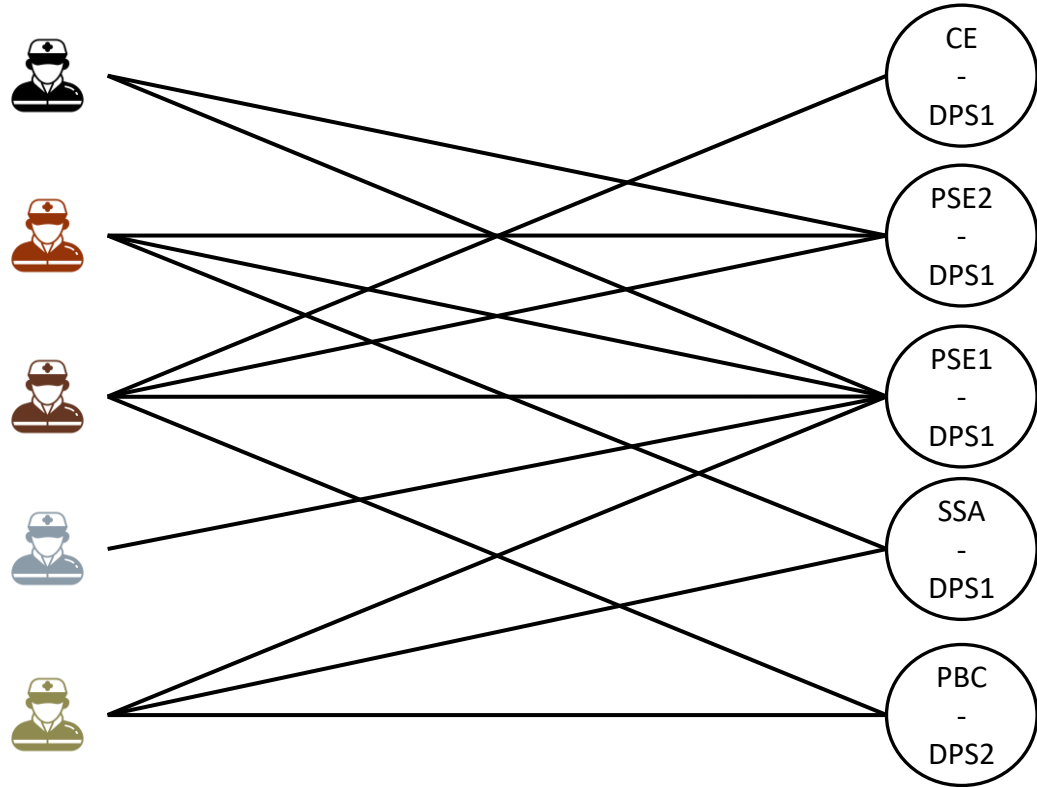







	CE	PSE2	PSE1	SSA	PBC
	0	1	1	0	0
	0	1	1	1	0
	1	1	1	0	1
	0	0	1	0	0
	0	0	1	1	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ On effectue d'abord un tri croissant des lignes selon la somme de leurs coûts

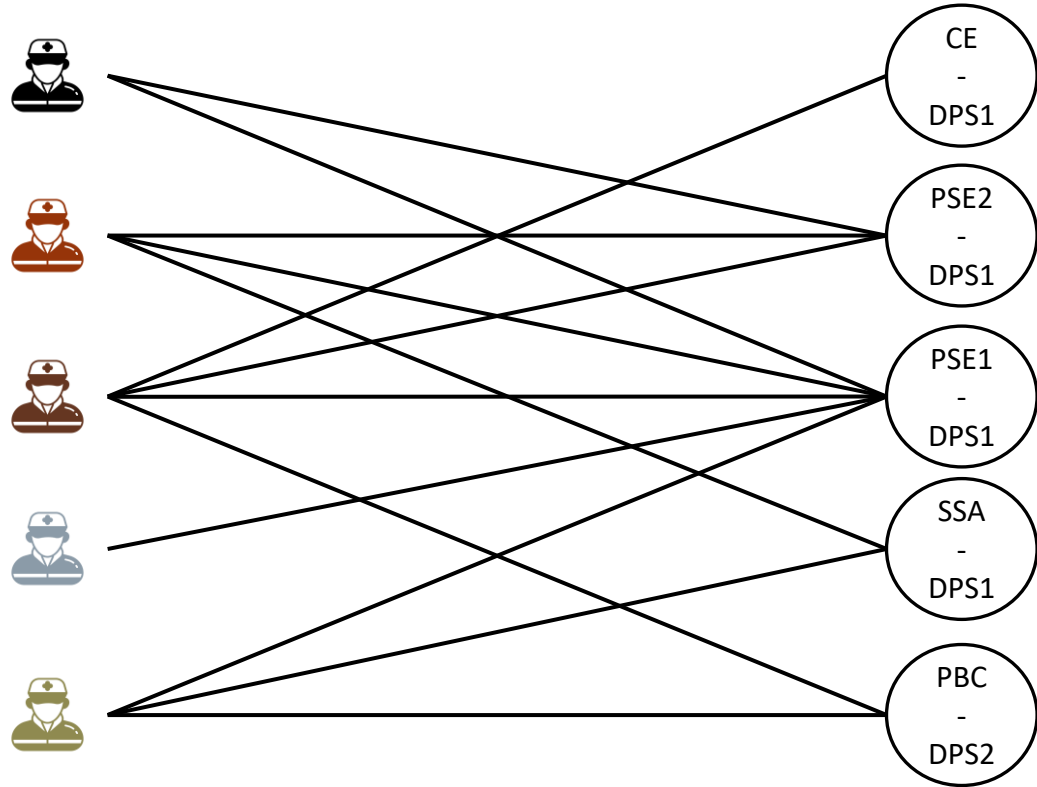


	CE	PSE2	PSE1	SSA	PBC
	0	1	1	0	0
	0	1	1	1	0
	1	1	1	0	1
	0	0	1	0	0
	0	0	1	1	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ On effectue d'abord un tri croissant des lignes selon la somme de leurs coûts

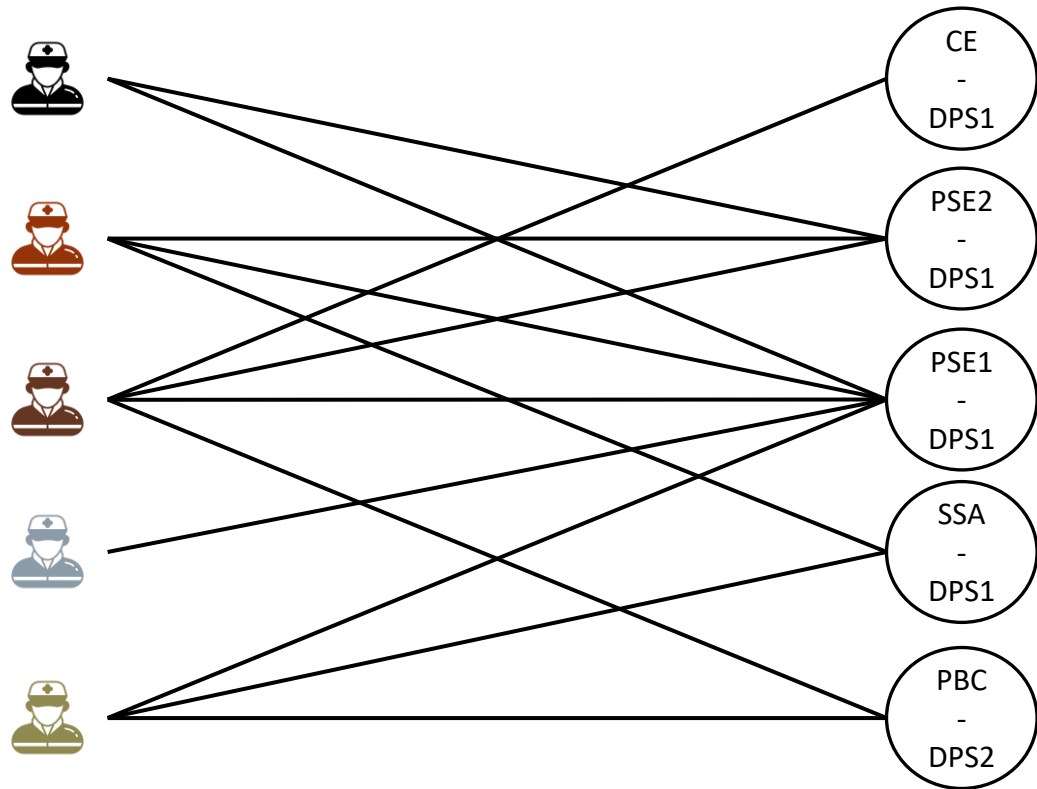







	CE	PSE2	PSE1	SSA	PBC
	0	0	1	0	0
	0	1	1	0	0
	0	1	1	1	0
	0	0	1	1	1
	1	1	1	0	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ Ensuite, on effectue un tri croissant des colonnes selon la somme de leurs coûts

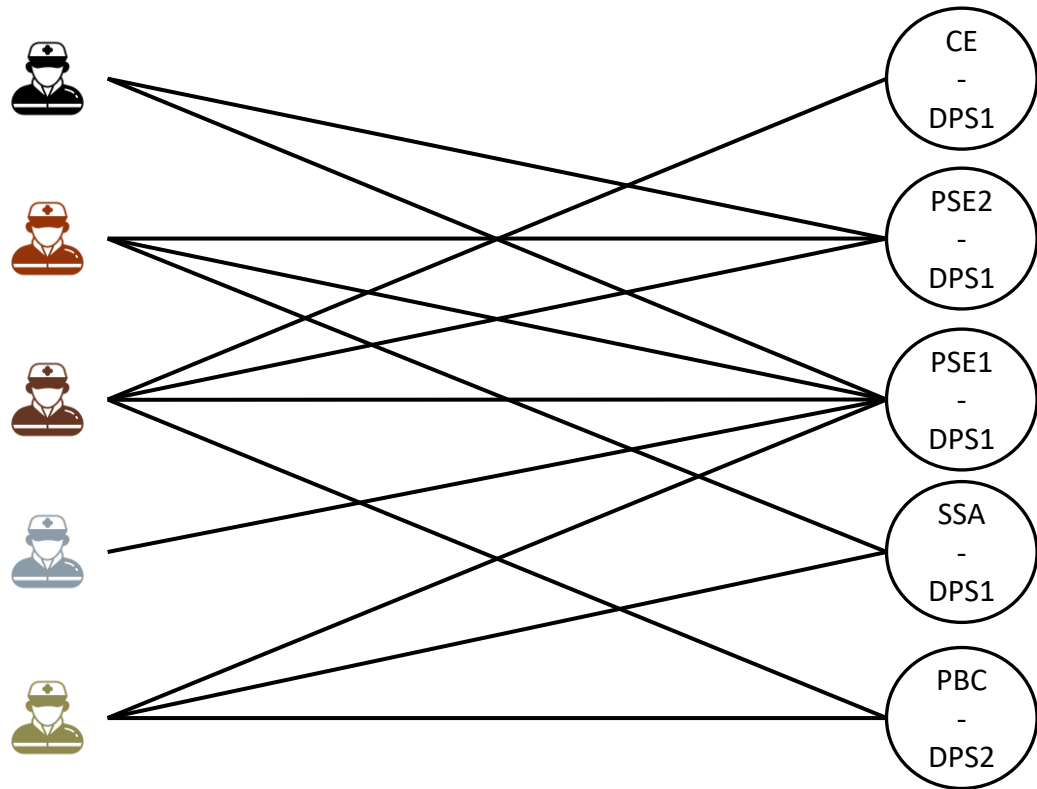


	CE	PSE2	PSE1	SSA	PBC
	0	0	1	0	0
	0	1	1	0	0
	0	1	1	1	0
	0	0	1	1	1
	1	1	1	0	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ Ensuite, on effectue un tri croissant des colonnes selon la somme de leurs coûts

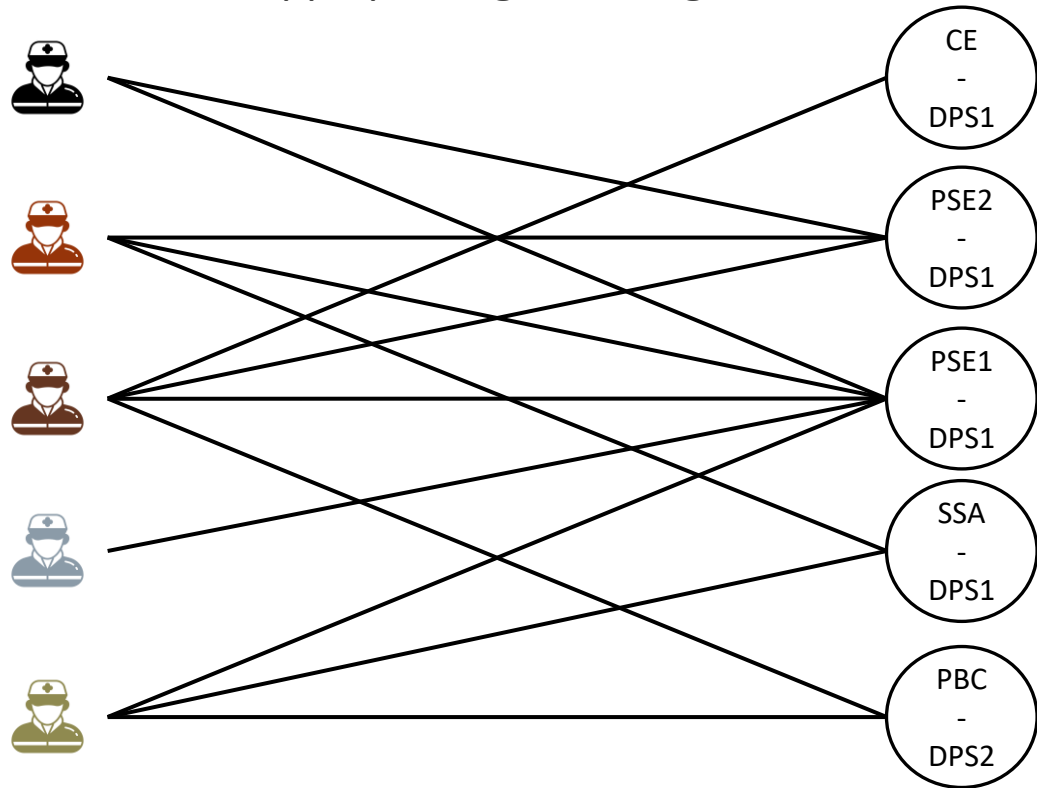







	CE	SSA	PBC	PSE2	PSE1
	0	0	0	0	1
	0	0	0	1	1
	0	1	0	1	1
	0	1	1	0	1
	1	0	1	1	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ Ensuite, on applique l'algorithme glouton

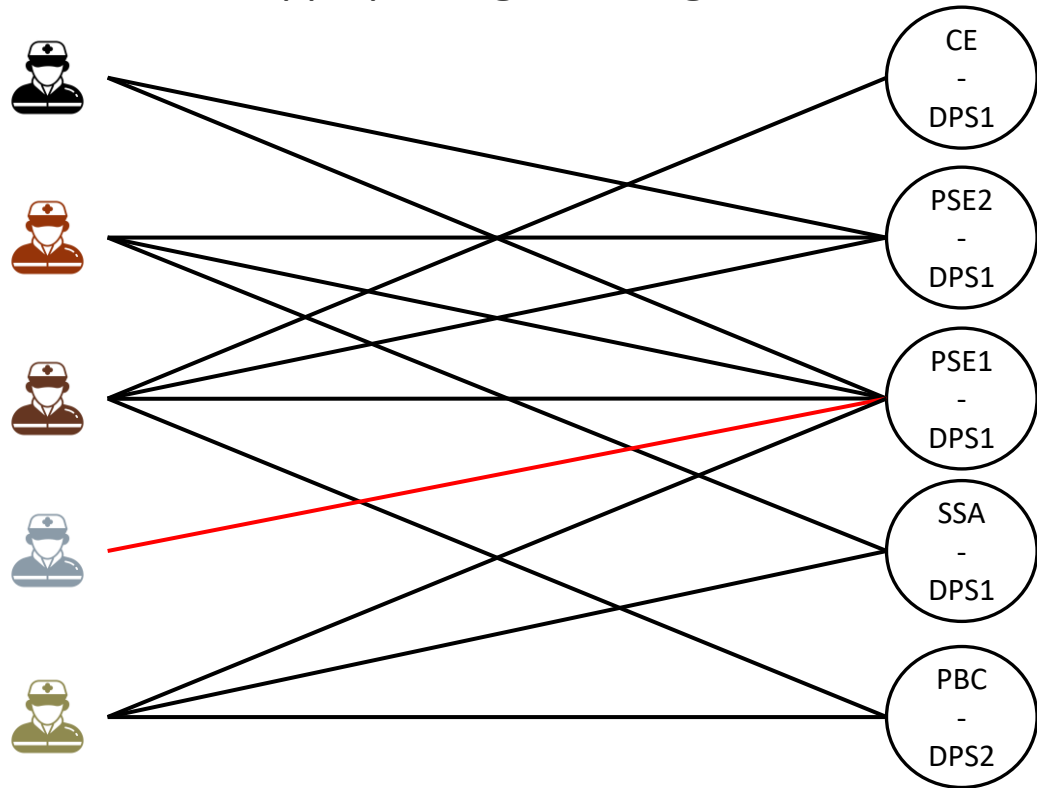


	CE	SSA	PBC	PSE2	PSE1
	0	0	0	0	1
	0	0	0	1	1
	0	1	0	1	1
	0	1	1	0	1
	1	0	1	1	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ Ensuite, on applique l'algorithme glouton

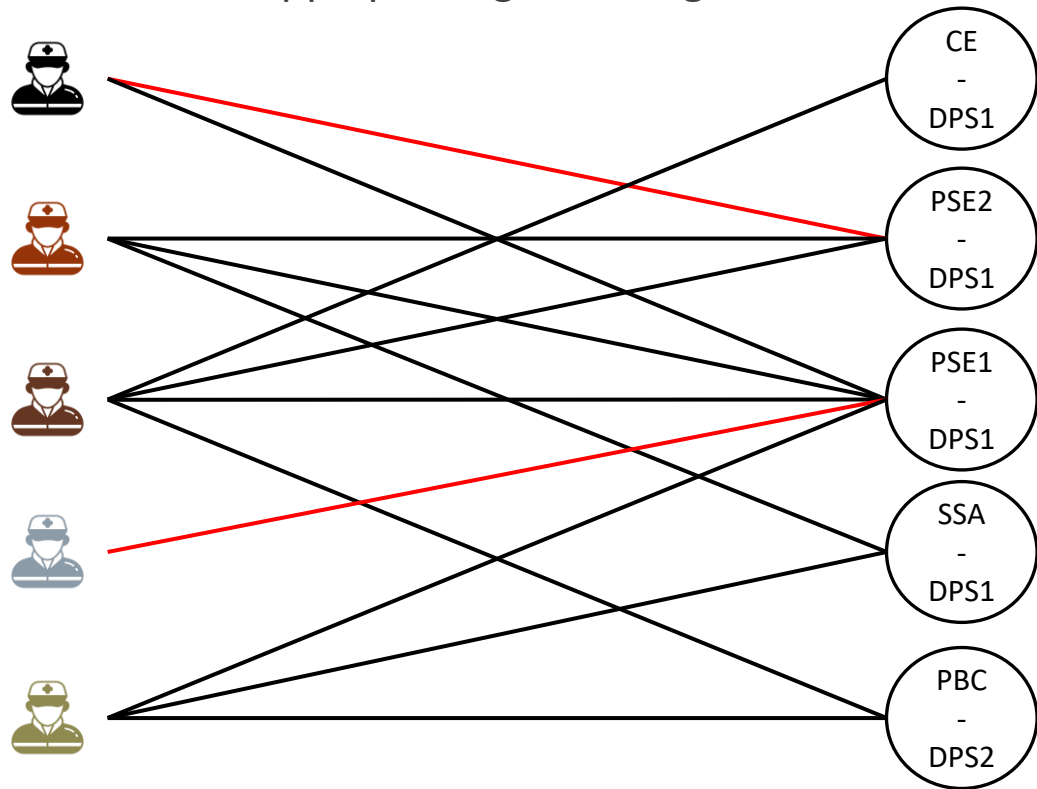


	CE	SSA	PBC	PSE2	PSE1
	0	0	0	0	1
	0	0	0	1	1
	0	1	0	1	1
	0	1	1	0	1
	1	0	1	1	1






❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ Ensuite, on applique l'algorithme glouton

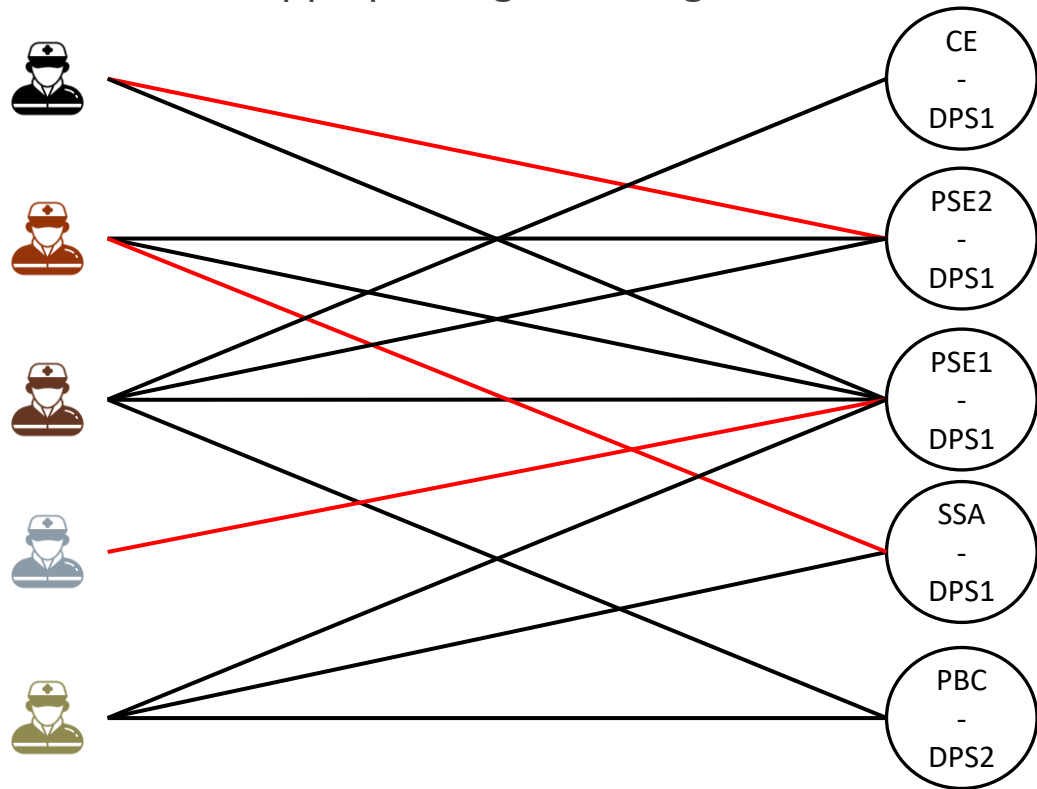


❖ Ne garantit pas une solution optimale

	CE	SSA	PBC	PSE2	PSE1
	0	0	0	0	1
	0	0	0	1	1
	0	1	0	1	1
	0	1	1	0	1
	1	0	1	1	1

Algorithme glouton d'affectation - version avec priorités

❖ Ensuite, on applique l'algorithme glouton

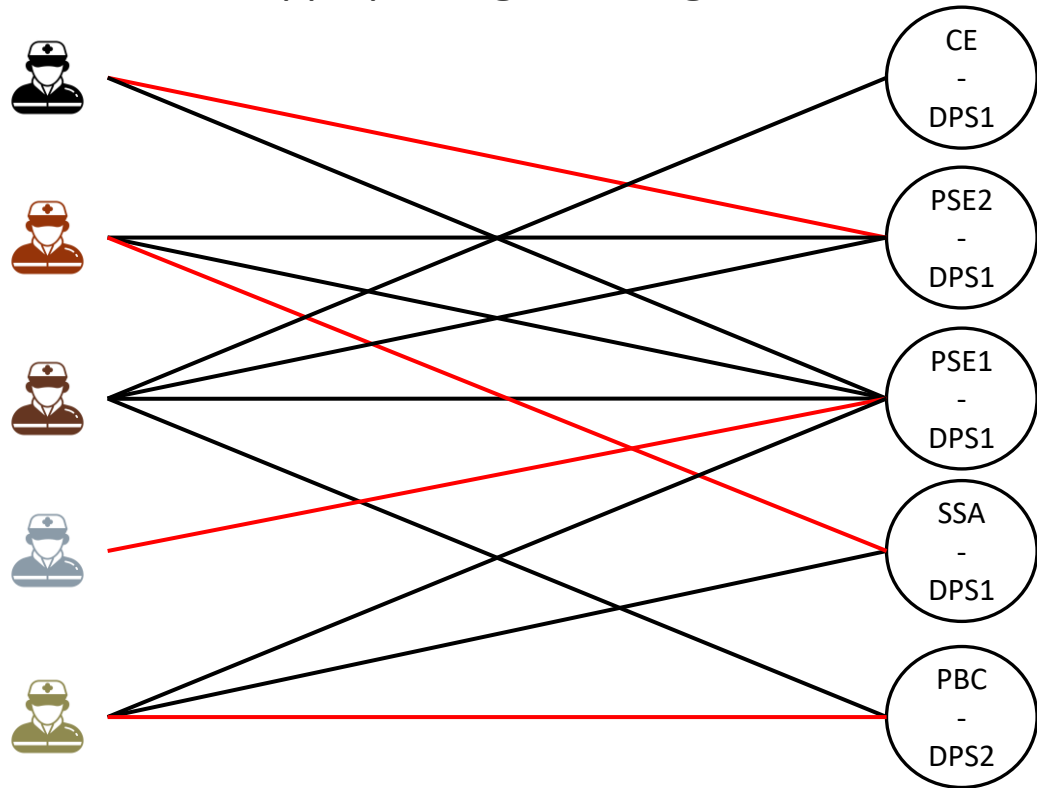







	CE	SSA	PBC	PSE2	PSE1
	0	0	0	0	1
	0	0	0	1	1
	0	1	0	1	1
	0	1	1	0	1
	1	0	1	1	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ Ensuite, on applique l'algorithme glouton

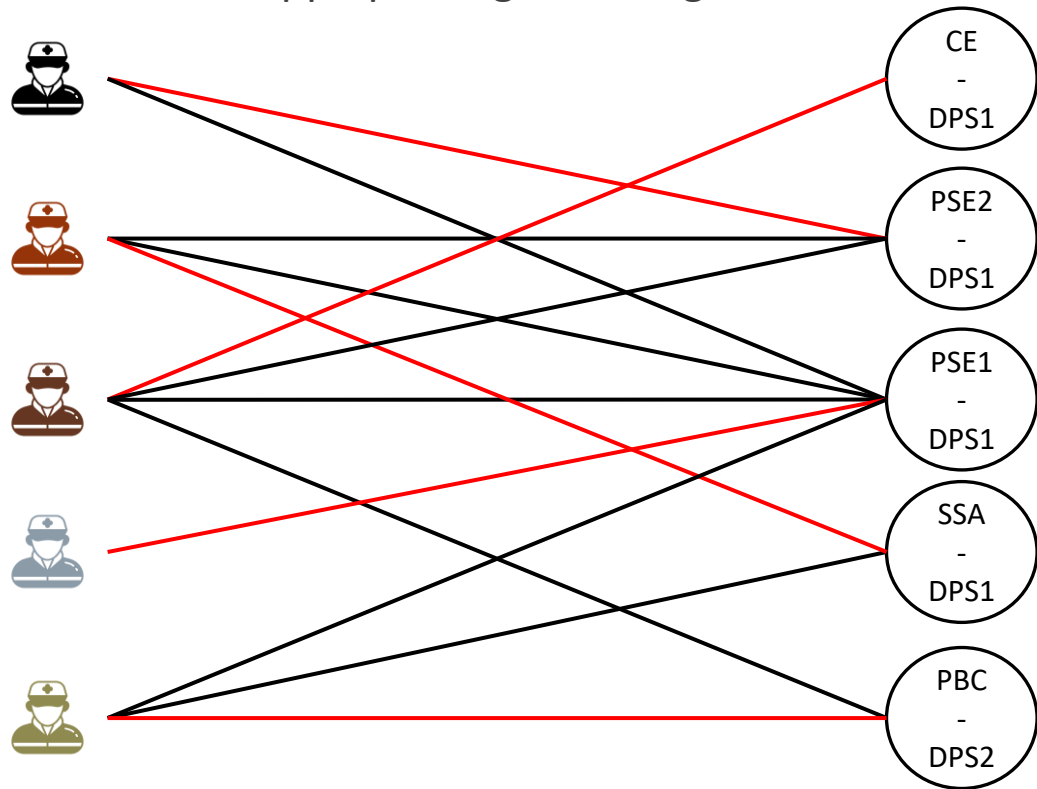







	CE	SSA	PBC	PSE2	PSE1
	0	0	0	0	1
	0	0	0	1	1
	0	1	0	1	1
	0	1	1	0	1
	1	0	1	1	1

❖ Ne garantit pas une solution optimale

Algorithme glouton d'affectation - version avec priorités

❖ Ensuite, on applique l'algorithme glouton



	CE	SSA	PBC	PSE2	PSE1
	0	0	0	0	1
	0	0	0	1	1
	0	1	0	1	1
	0	1	1	0	1
	1	0	1	1	1

❖ Ne garantit pas une solution optimale