

Université Abdelhamid Mehri Constantine 2- Algérie
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Informatique Fondamentale et ses Applications



1^{ère} Année Master Réseaux et Systèmes Distribués

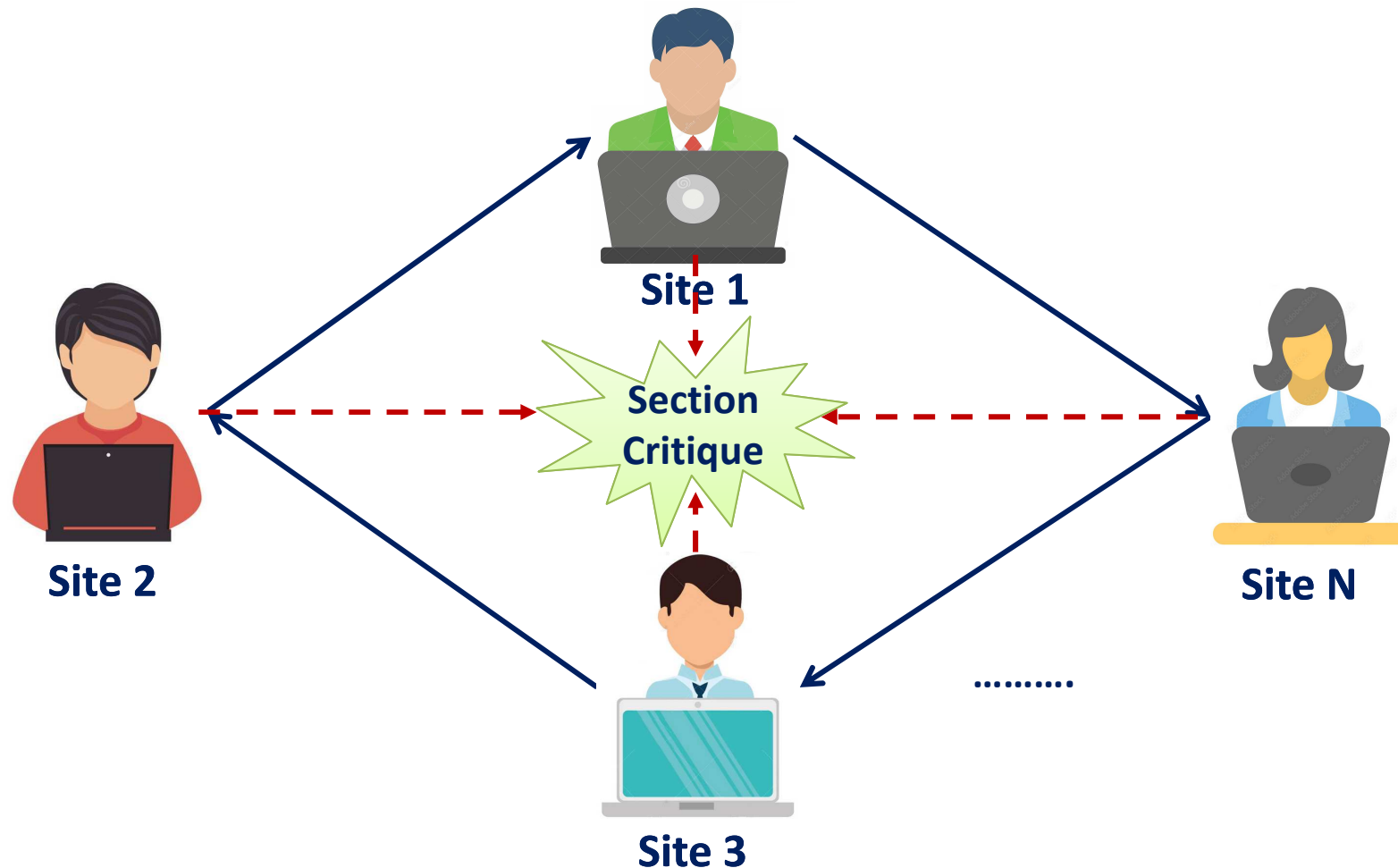
TP ALGORITHMES DISTRIBUÉS (ALDI)

**Correction du TP N°5 : Implémentation du 2ème
algorithme distribué en utilisant la plateforme JADE
(Algorithme basé sur le jeton)**

Année universitaire : 2021/2022

Enoncé de l'algorithme

Soit un système distribué composé de **N sites** reliés par un anneau unidirectionnel. On suppose que les liaisons sont fiables et chaque **site** connaît seulement le nom de son **successeur**.

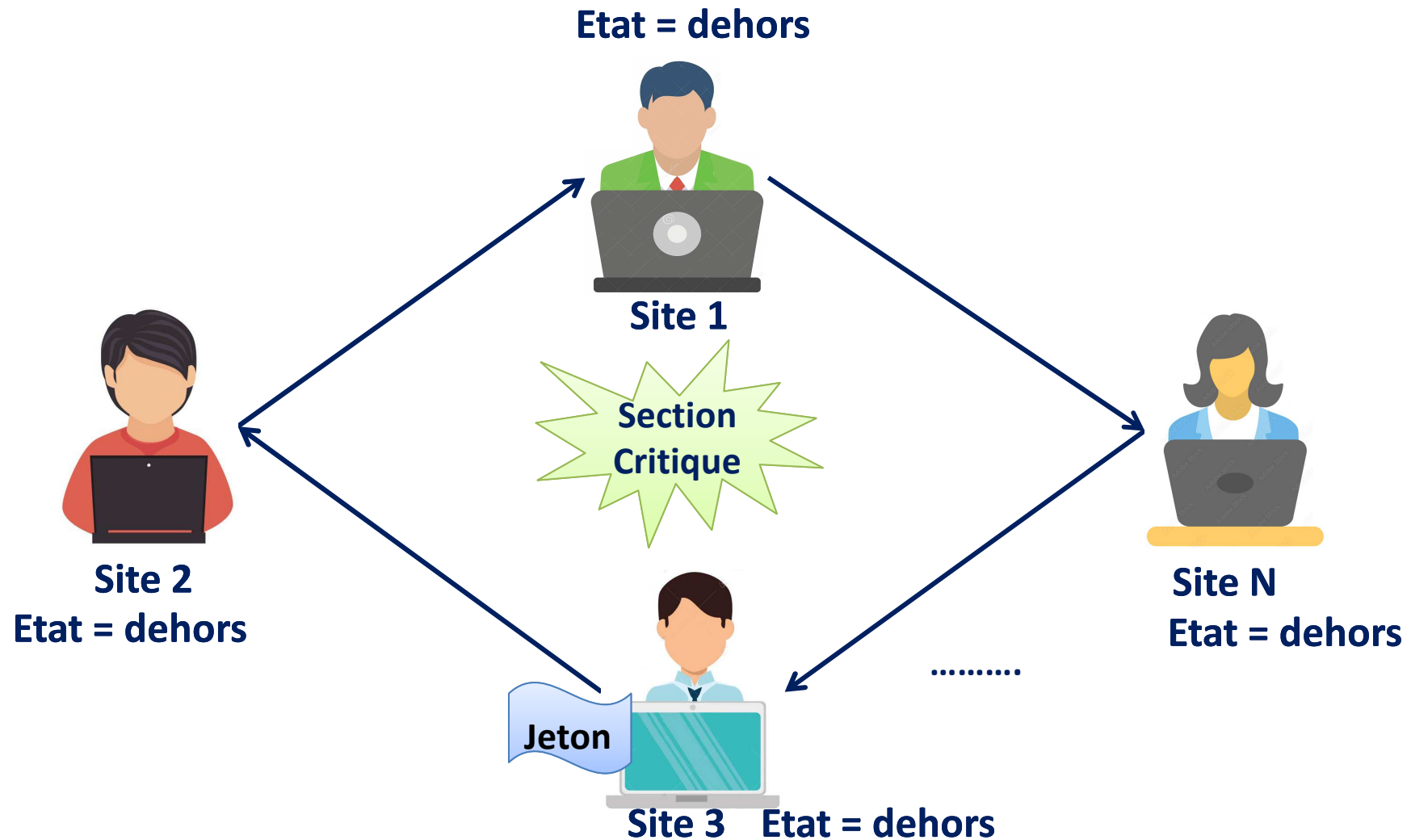


Enoncé de l'algorithme

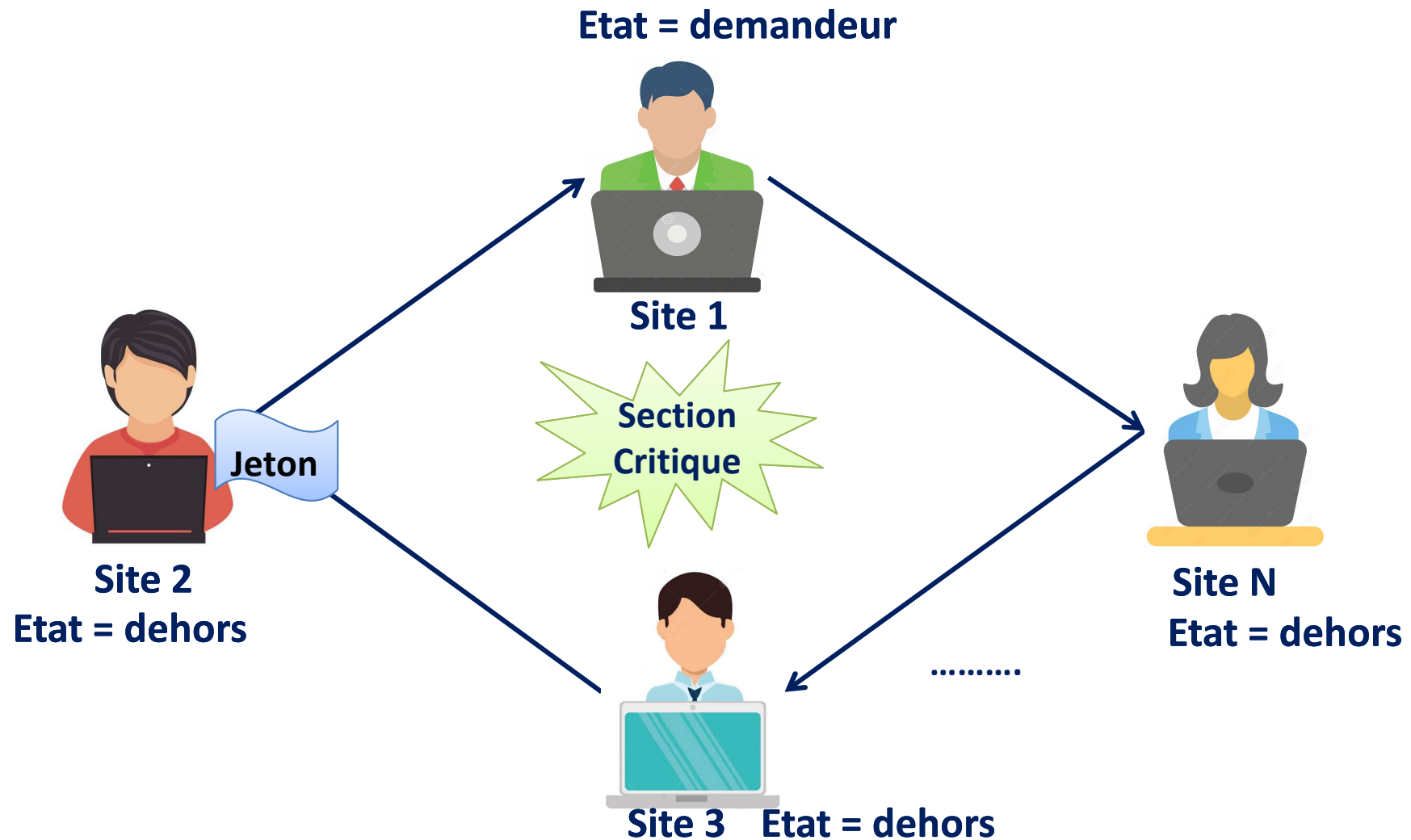
On considère l'algorithme de **Le Lann** qui permet de résoudre le problème de la section critique. Ce dernier est basé sur l'existence d'un message appelé **Jeton** qui est en mouvement circulaire sur l'anneau. Le principe de fonctionnement de cet algorithme est le suivant :

- Initialement, le **Jeton** est placé sur un **site** et tous les **sites** sont dans l'état **dehors**.
- Lorsqu'un site **Pi** désire entrer en section critique (**SC**), il passe à l'état **demandeur** et attend l'arrivée du **Jeton**. Lorsque le **Jeton** arrive au **site Pi** alors **Pi** passe à l'état **dedans** et accède à la **SC**.
- Au bout d'un temps fini, **Pi** libère la **SC** en passant vers l'état **dehors** et en envoyant le **Jeton** à son **successeur**.
- Lorsqu'un **site** reçoit le **Jeton** et il est dans l'état **dehors** alors il envoie le **Jeton** à son **successeur**.

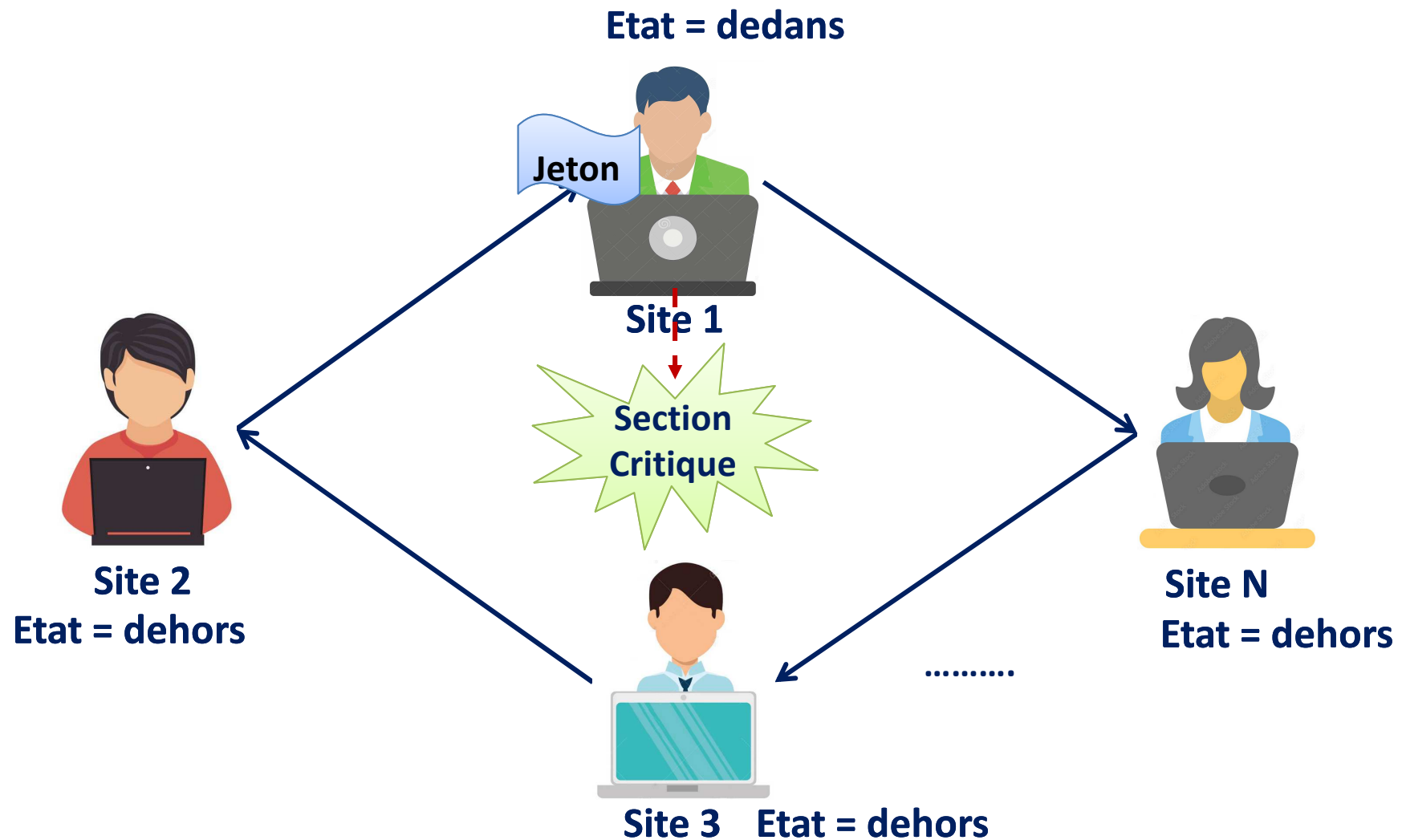
Enoncé de l'algorithme



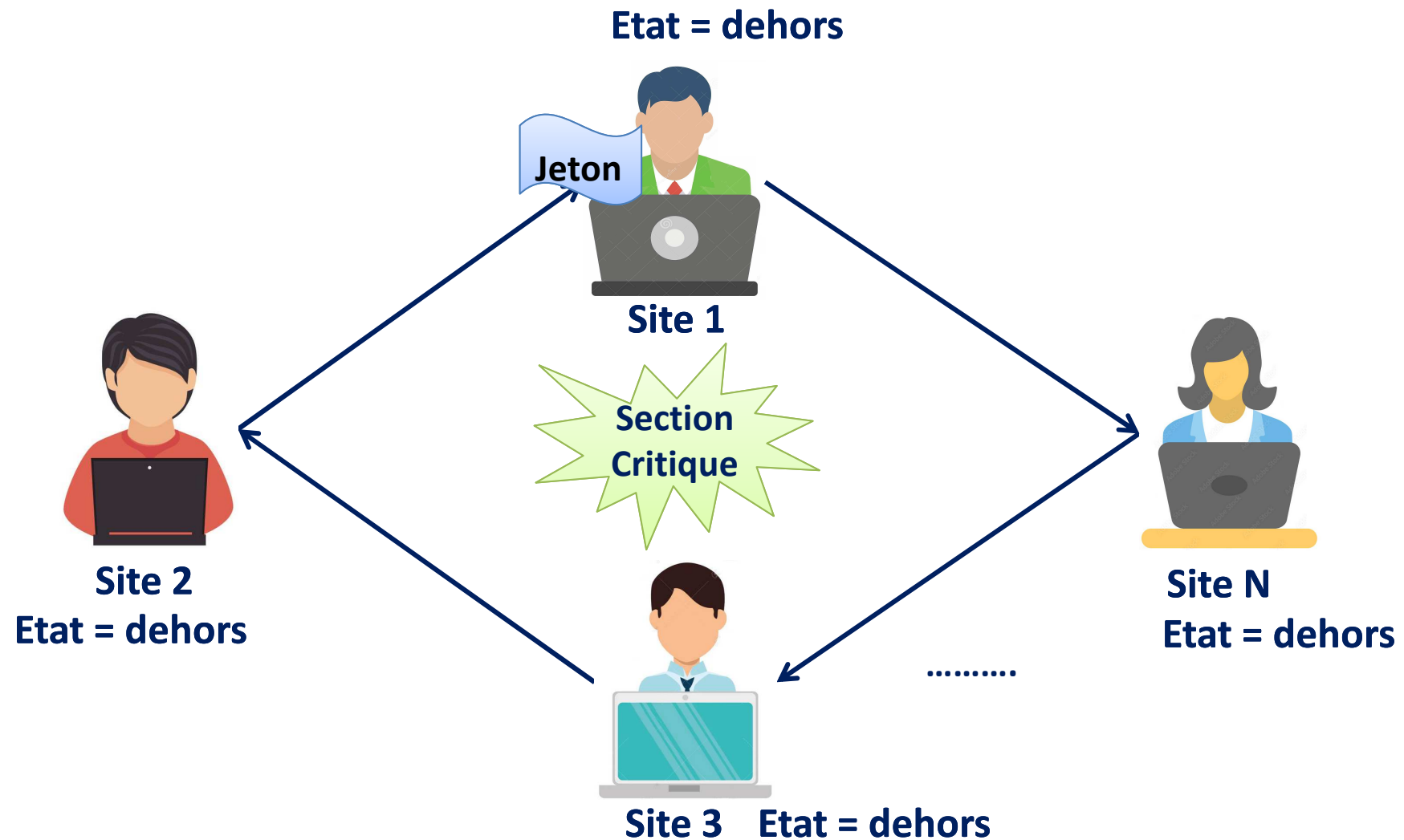
Enoncé de l'algorithme



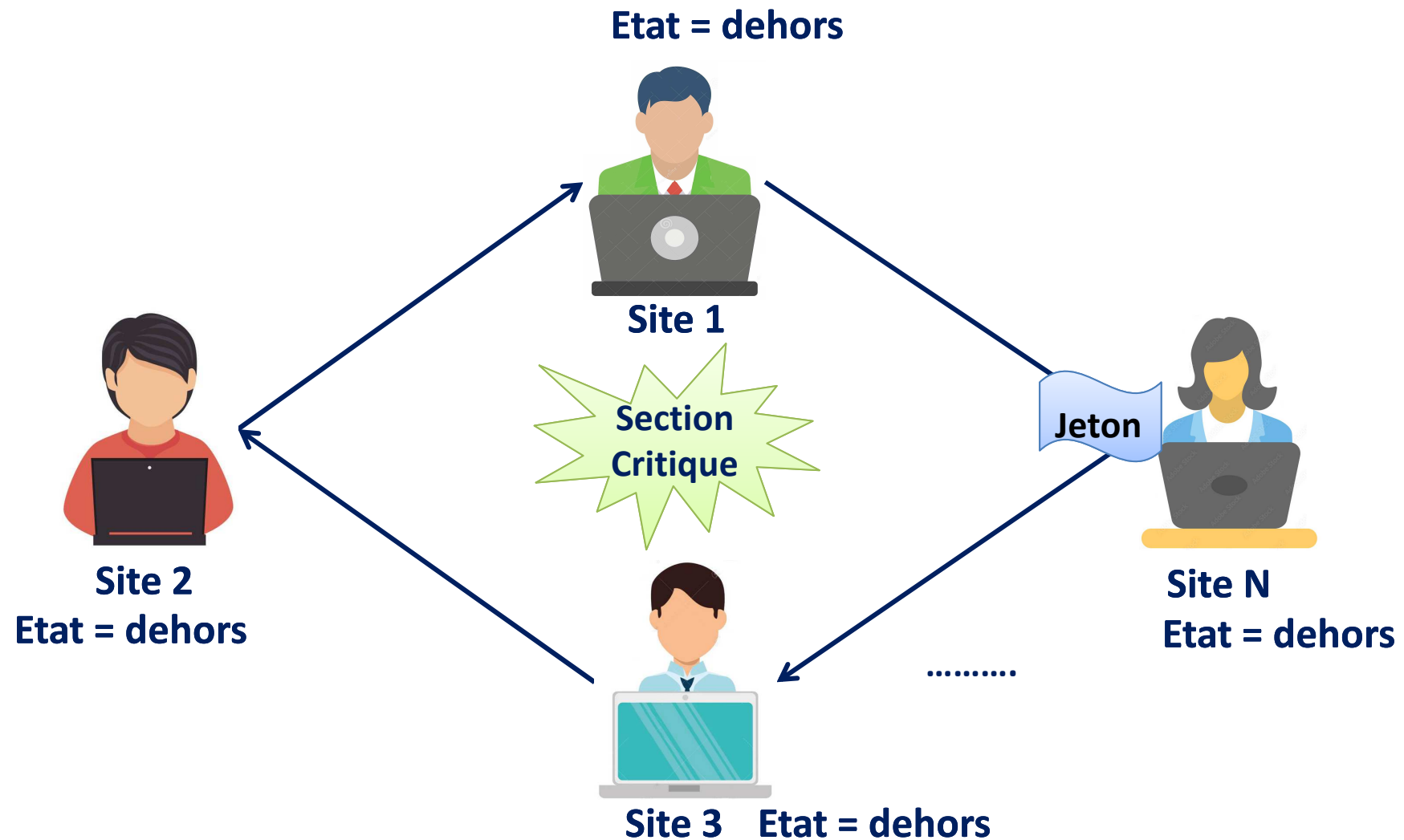
Enoncé de l'algorithme



Enoncé de l'algorithme



Enoncé de l'algorithme



Enoncé de l'algorithme

Variables locales pour un processus P_i :

- **suivant** : Nom du site successeur dans l'anneau;
- **état** = {dehors, demandeur, dedans} initialisé à dehors;
- **jetonPrésent** : booléen initialisé à **faux** sauf sur le site j sur lequel est initialement placé le jeton;

Enoncé de l'algorithme

Lors d'un appel à acquérir

état = demandeur;

Attendre (jetonPrésent == vrai);

état = dedans;

Enoncé de l'algorithme

Lors d'un appel à libérer

étati = dehors;

jetonPrésent = faux;

Envoyer ***Jeton*** au **suivant**;

Enoncé de l'algorithme

Lors de la réception du *jeton*

Si (état == dehors) alors

Envoyer *Jeton* au suivant;

Sinon

jetonPrésent = vrai;

Fin Si

Démarche d'implémentation d'un algorithme distribué sous la plateforme JADE

1. Identifier les types d'agents qui seront utilisés dans l'algorithme distribué.
2. Identifier les comportements de chaque agent ainsi que leur type puis tracer un graphe qui représente le lien entre les différents comportements.
3. Identifier les différents messages échangés entre les agents et donner leur contenu.
4. Identifier les arguments qui seront passés à chaque agent.
5. Implémenter la classe de chaque agent (la méthode setup et les différents comportements).
6. Tester les classes implémentées.
7. Analyser le contenu de la console après exécution, s'il y a des anomalies alors il faut revoir l'étape 5.

Implémentation de l'algorithme

1. Identifier les types d'agents qui seront utilisés dans l'algorithme distribué :


Dans cet algorithme, nous avons un seul type d'agents :

- Un agent (site) qui désire accéder à une section critique.

Implémentation de l'algorithme

2. Identifier les comportements de chaque agent ainsi que leur type puis tracer un graphe qui représente le lien entre les différents comportements :


Comportement de l'Agent (site)	Type de Comportements
Dehors	GenericBehaviour
↓	
DevenirDemandeur	OneShotBehaviour
↓	
AttendreJetonPresent	GenericBehaviour
↓	
EnSc	OneShotBehaviour
↓	
LibererSC	OneShotBehaviour



Implémentation de l'algorithme

2. Identifier les comportements de chaque agent ainsi que leur type puis tracer un graphe qui représente le lien entre les différents comportements :

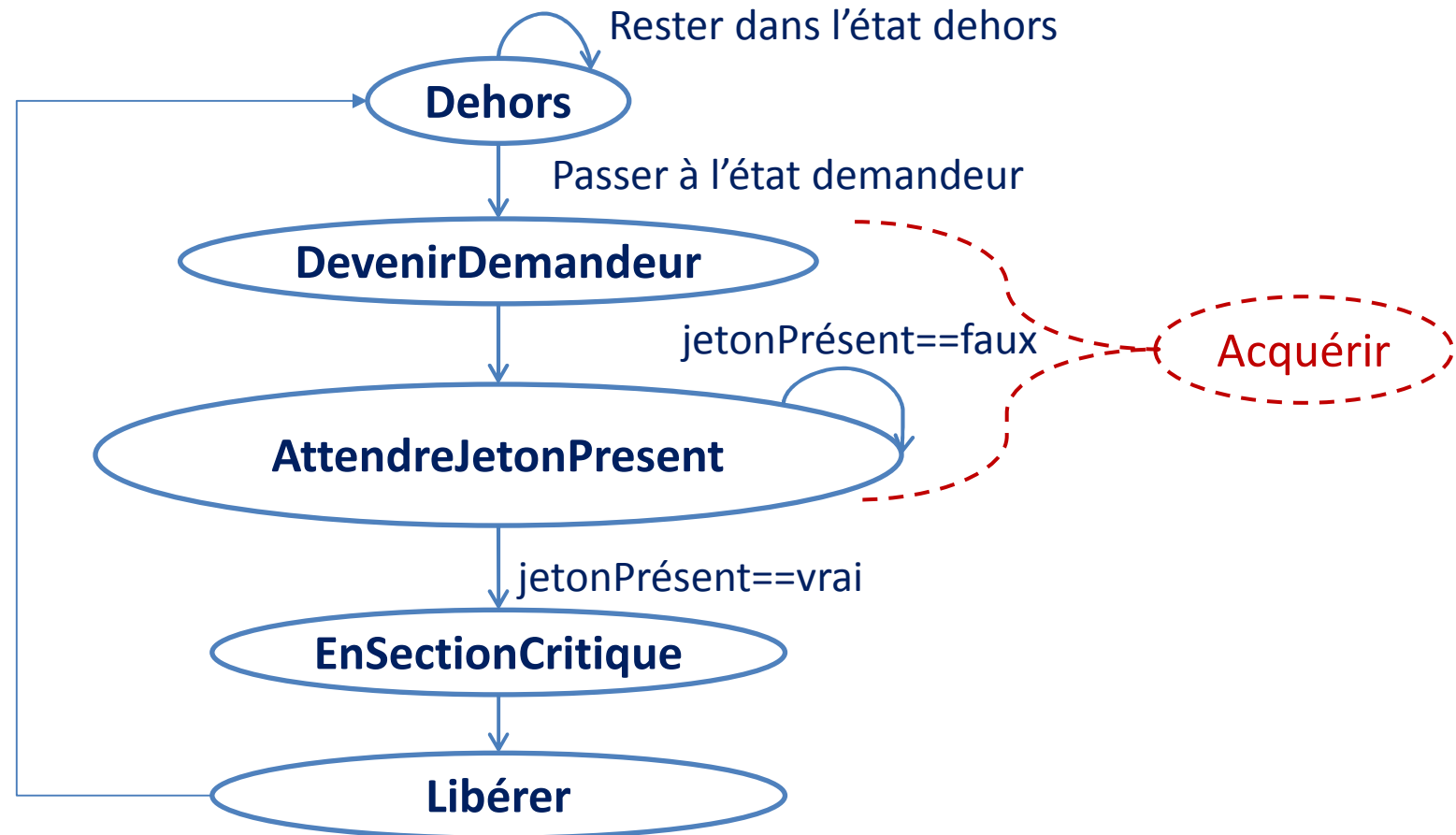
Comportement de l'Agent (site)	Type de Comportements
Dehors	GenericBehaviour
↓	
DevenirDemandeur	OneShotBehaviour
↓	
AttendreJetonPresent	GenericBehaviour
↓	
EnSc	OneShotBehaviour
↓	
LibererSC	OneShotBehaviour



Remarque : la consultation de la boîte aux lettres se fera dans une méthode qui sera appelée dans certains comportement

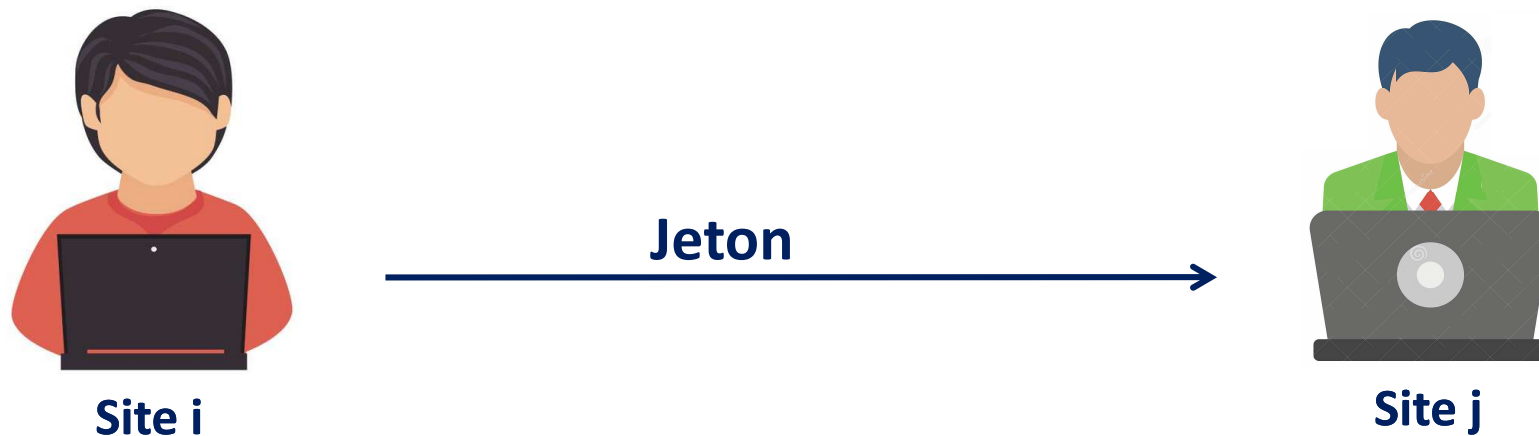
Implémentation de l'algorithme

2. Identifier les comportements de chaque agent ainsi que leur type puis tracer un graphe qui représente le lien entre les différents comportements :



Implémentation de l'algorithme

3. Identifier les différents messages échangés entre les agents et donner leur contenu :



Implémentation de l'algorithme

4. Identifier les arguments qui seront passés à chaque agents

- Chaque agent (site) doit connaître :
 - ✓ Le nom de son successeur
 - ✓ S'il possède le jeton (true/false, 0/1, ...)
- ⇒ La liste des arguments sera la forme suivante :
(NomSucc, jetonPrésent)

Exemple :

a:site(c,false)
b:site(a,true)
c:site(b,false)

Exemple :

a:site(c,0)
b:site(a,1)
c:site(b,0)

Implémentation de l'algorithme

5. Implémenter la classe de chaque agent (la méthode setup et les différents comportements)

- Nous allons implémenter 1 seule classe :

Classe Site