

Exclusion Mutuelle hiérarchique Composition de protocoles

Baptiste Canton

Pierre Sens - Luciana Arantes - Julien Sopena

 L'exclusion mutuelle distribuée.

 Brique de base de nombreuses applications réparties.

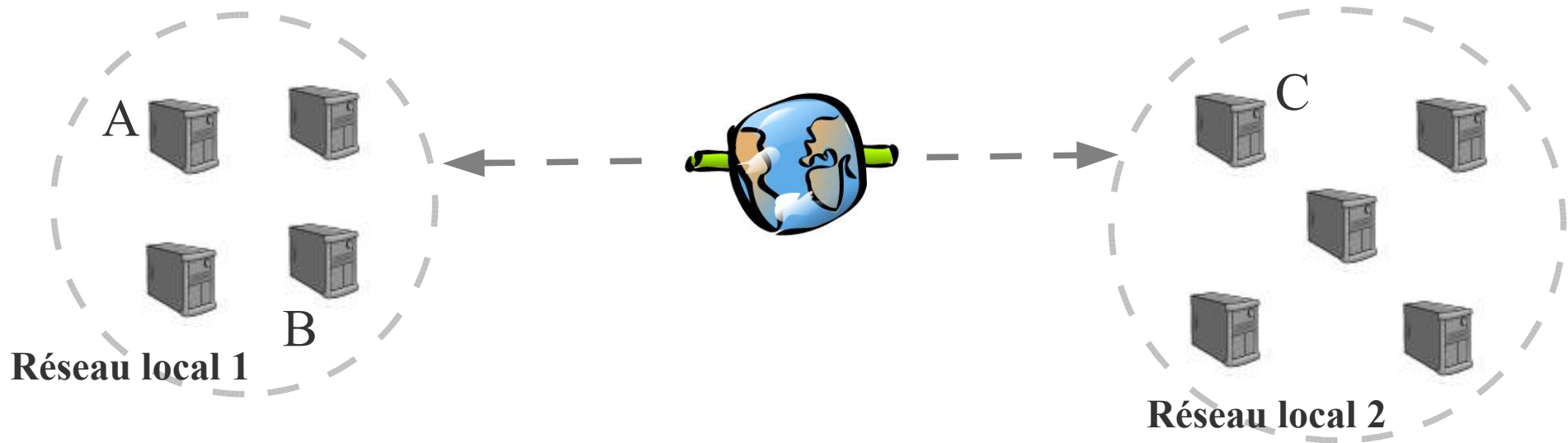
 Partager une ressource unique entre plusieurs sites.

 Tous les sites jouent le même rôle (symétrie).

 La Grille.

 Partager les ressources matérielles distribuées géographiquement.

 Structure physique hiérarchique.



Hiérarchisation due à l'hétérogénéité des communications

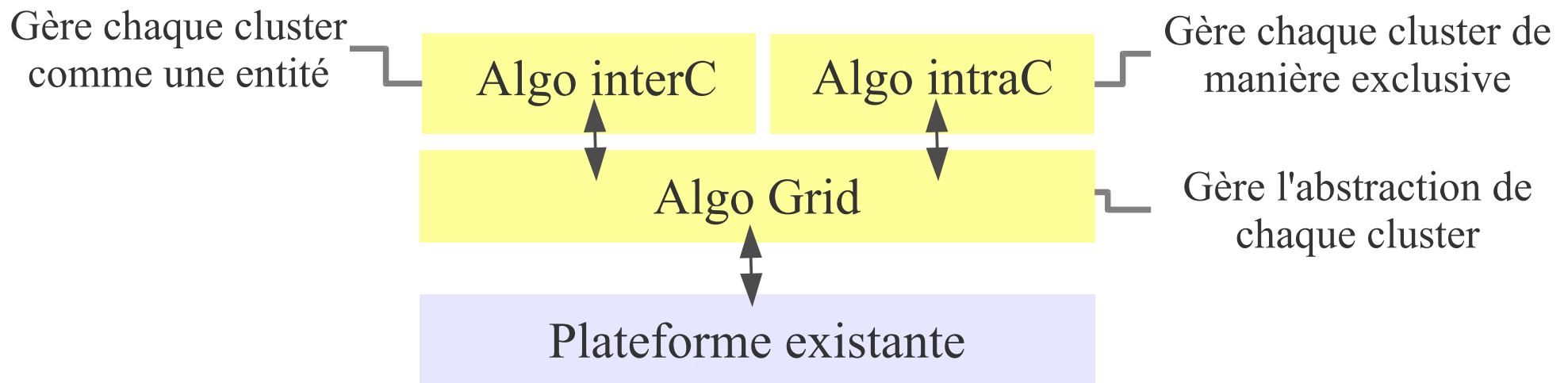
- intra-cluster → faible latence, débit élevé, peu de pertes, beaucoup de sites.
- inter-cluster → forte latence, débit moindre, plus de pertes, moins de noeuds.

Repenser les algorithmes existants supposant une topologie « à plat », tenter de les adapter pour prendre en compte l'infrastructure nouvelle.

Naimi-Trehel hiérarchique [BAS04], pour les grilles, montre de bons résultats.

- Combiner 2 algorithmes différents, chacun plus adapté au lien de communication utilisé.
- intra-cluster: peu sensible au nombre de messages simultanés, diffusion facile.
- inter-cluster: nécessite d'économiser le nombre de messages envoyés.

- Intégration du concept de GRID dans la plateforme expérimentale d'exclusion mutuelle développée par J.Sopena.
- Conception de bibliothèques aux interfaces génériques.



Il est ainsi possible de tester différentes combinaisons d'algorithmes.

Abstraction du cluster peut suivre différentes approches sans changer les algorithmes inter et intra:

 Proxy fixe représentant le cluster

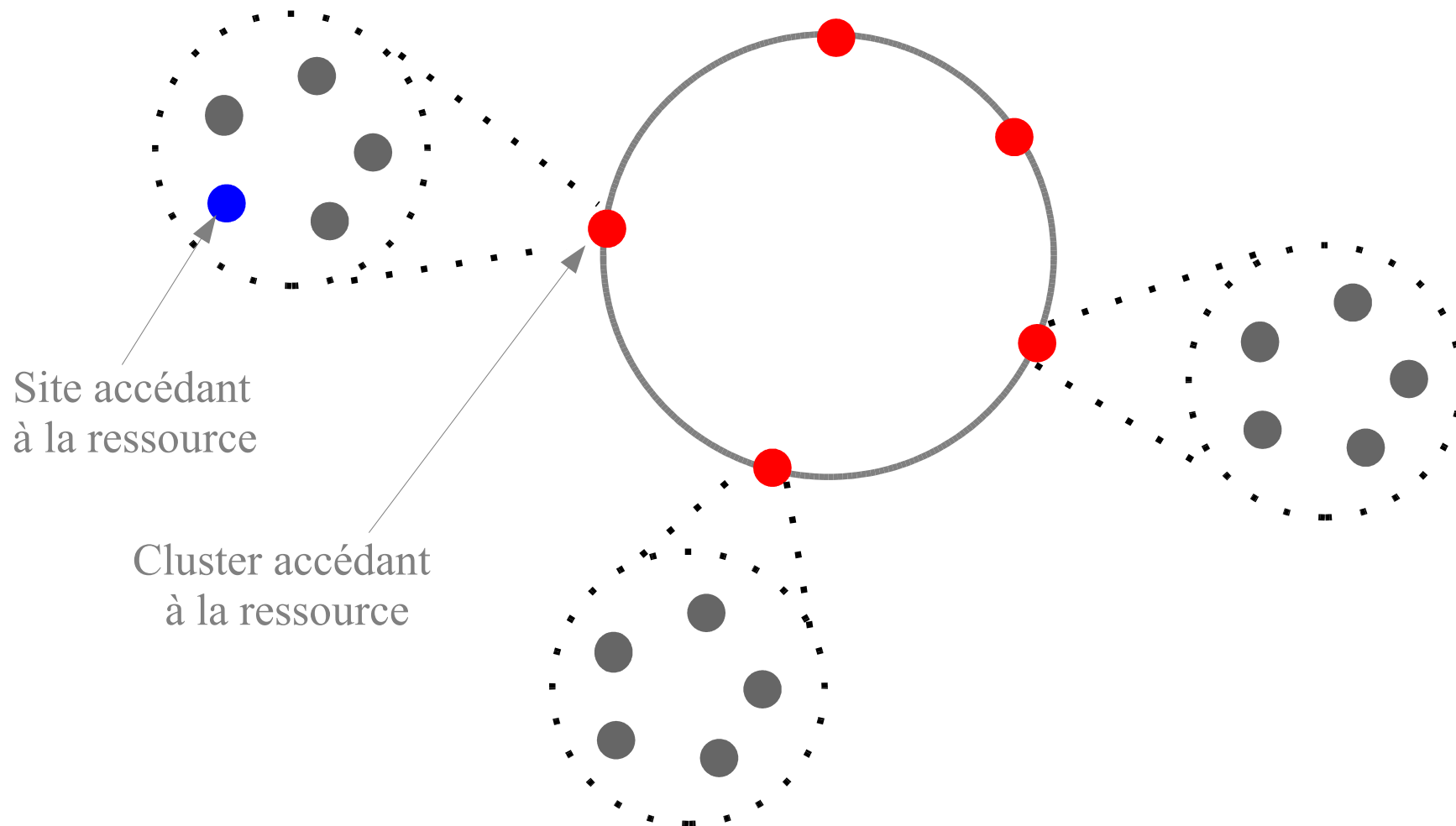
 Proxy dynamique

 Broadcast dans le cluster

Gère le changement de cluster de l'accès à la ressource critique en appelant les primitives de AlgoInter.

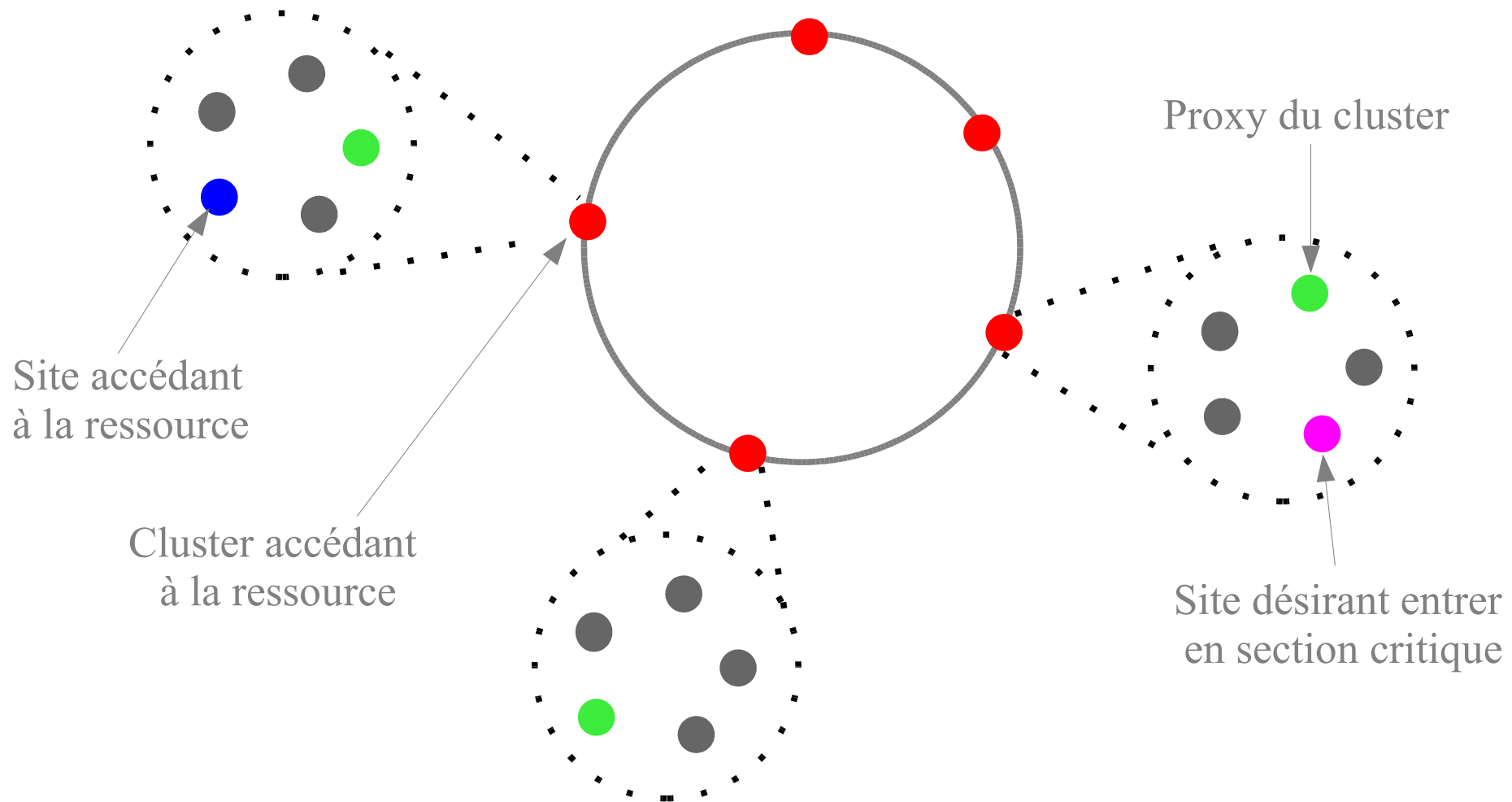
- Permet la préemption en favorisant les demandes locales.
- Évite la famine en permettant de « forcer » le changement de cluster.

Exemple de réalisation



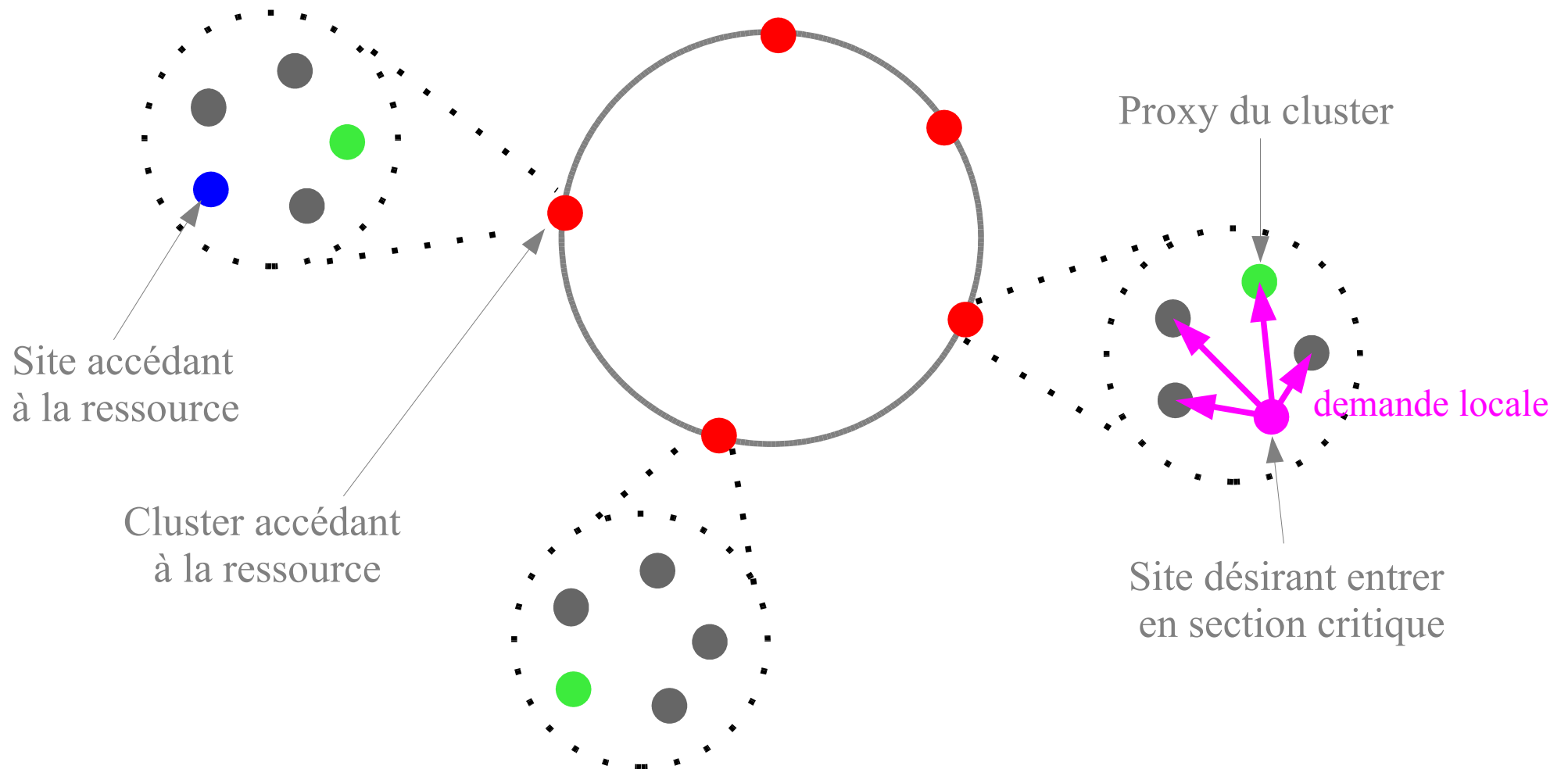
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



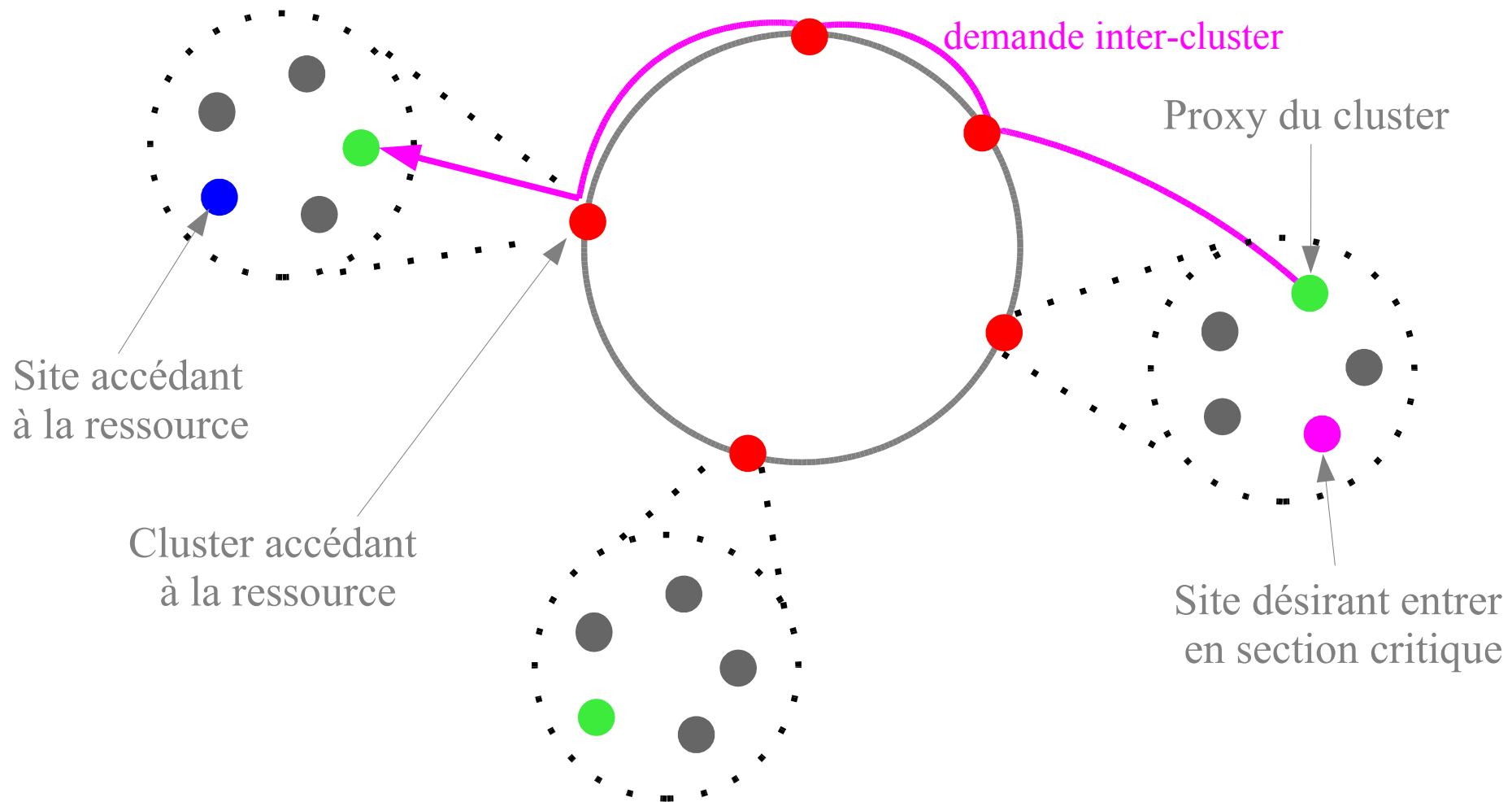
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



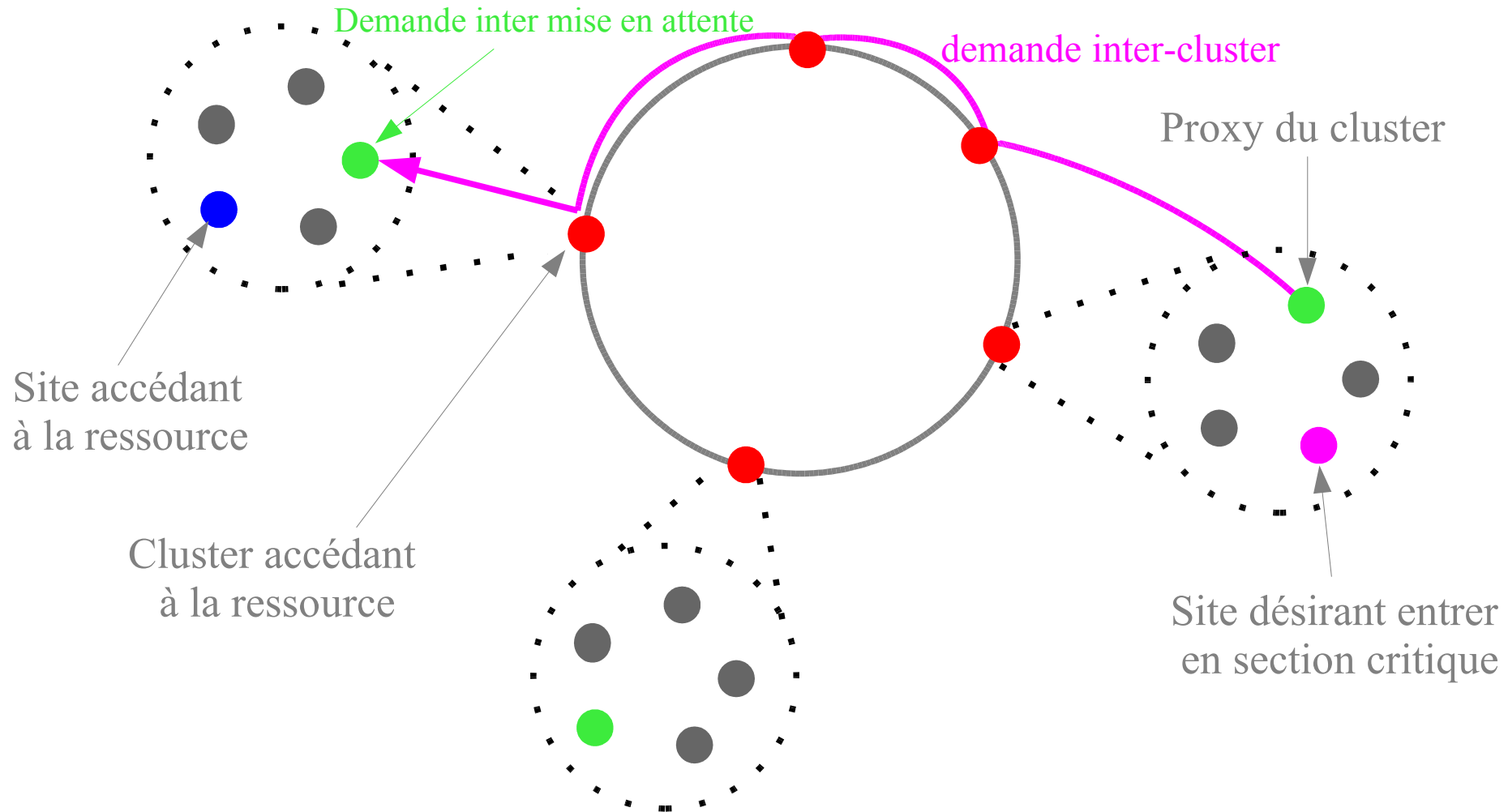
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



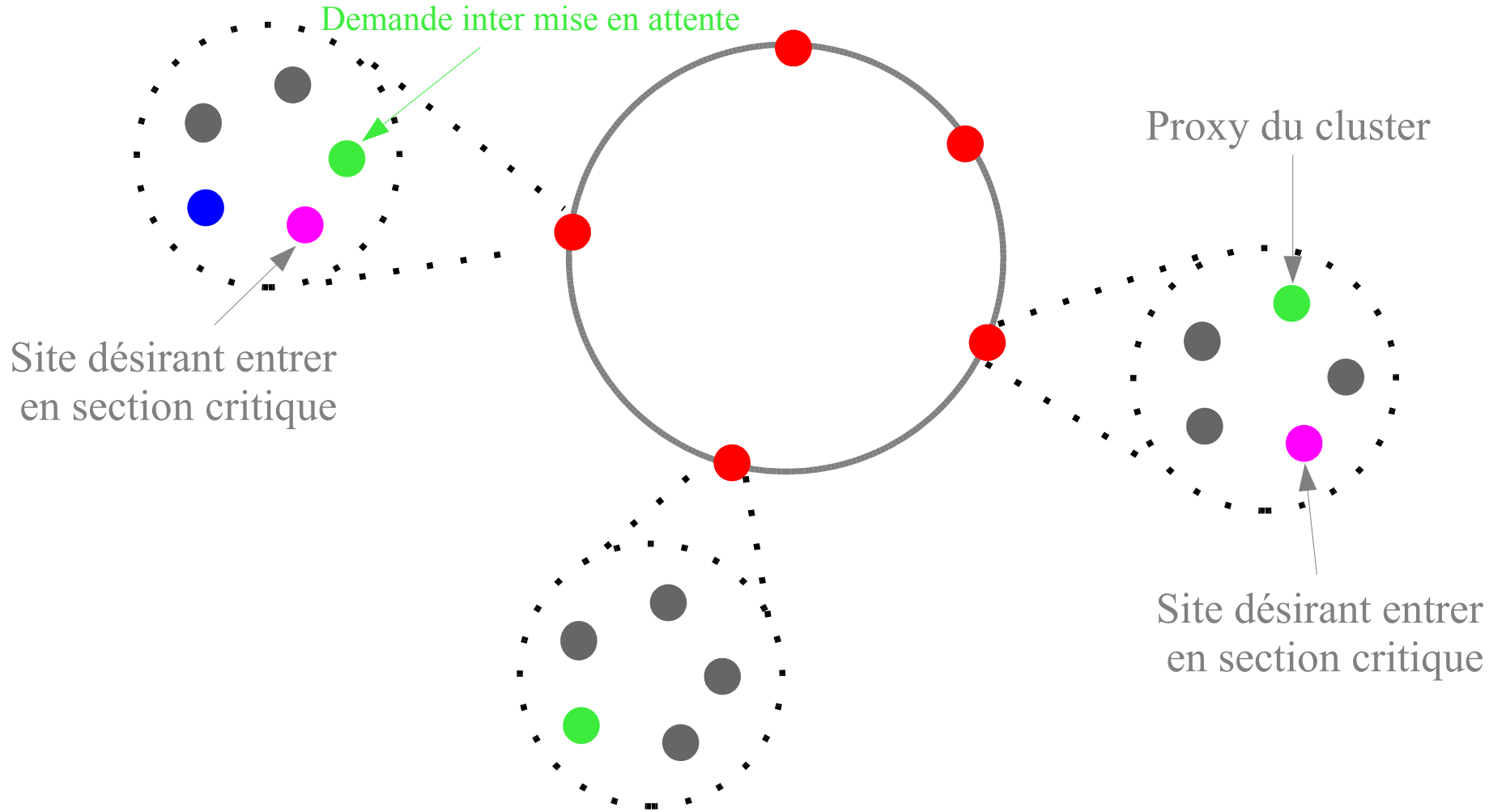
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



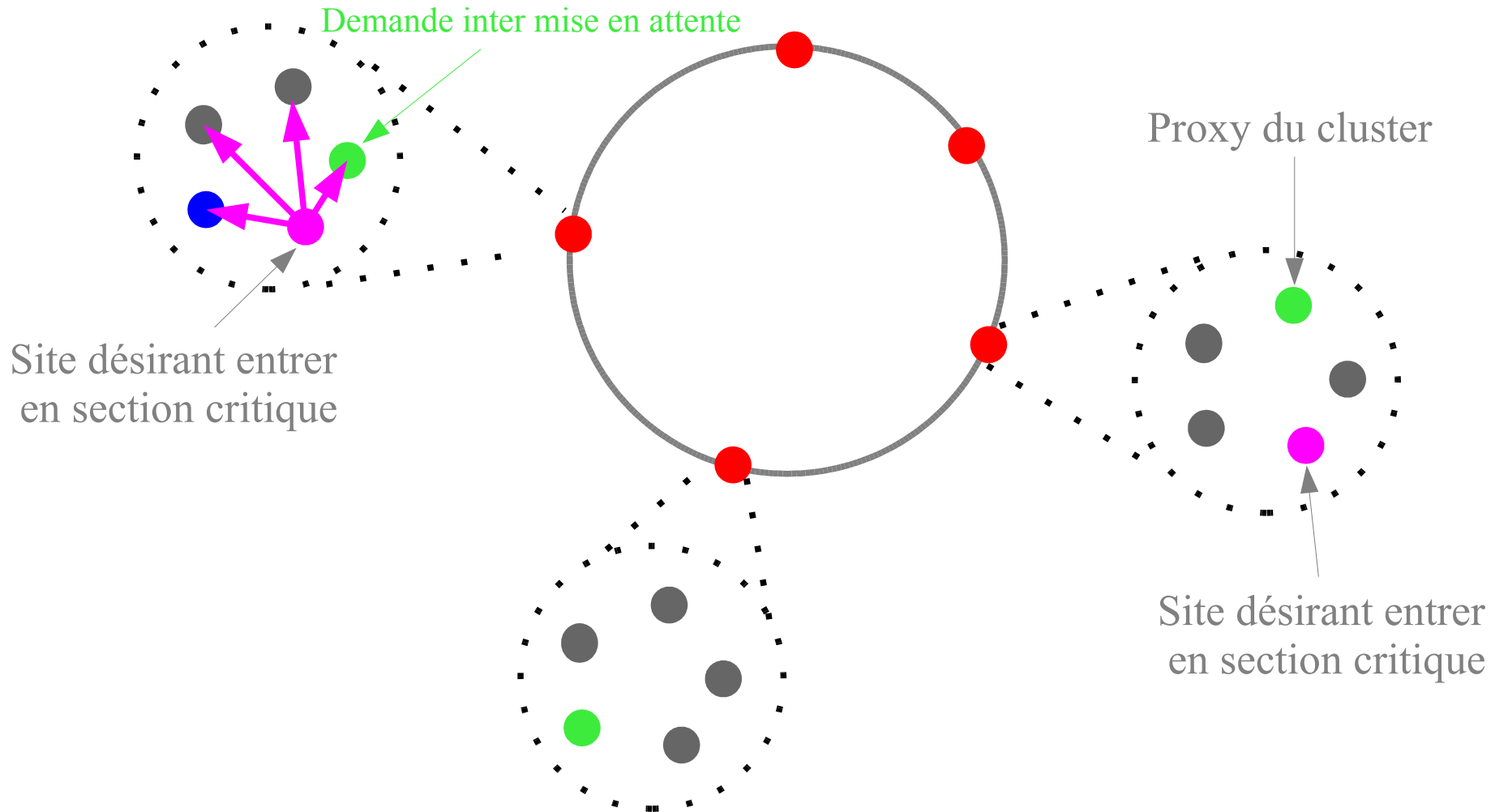
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



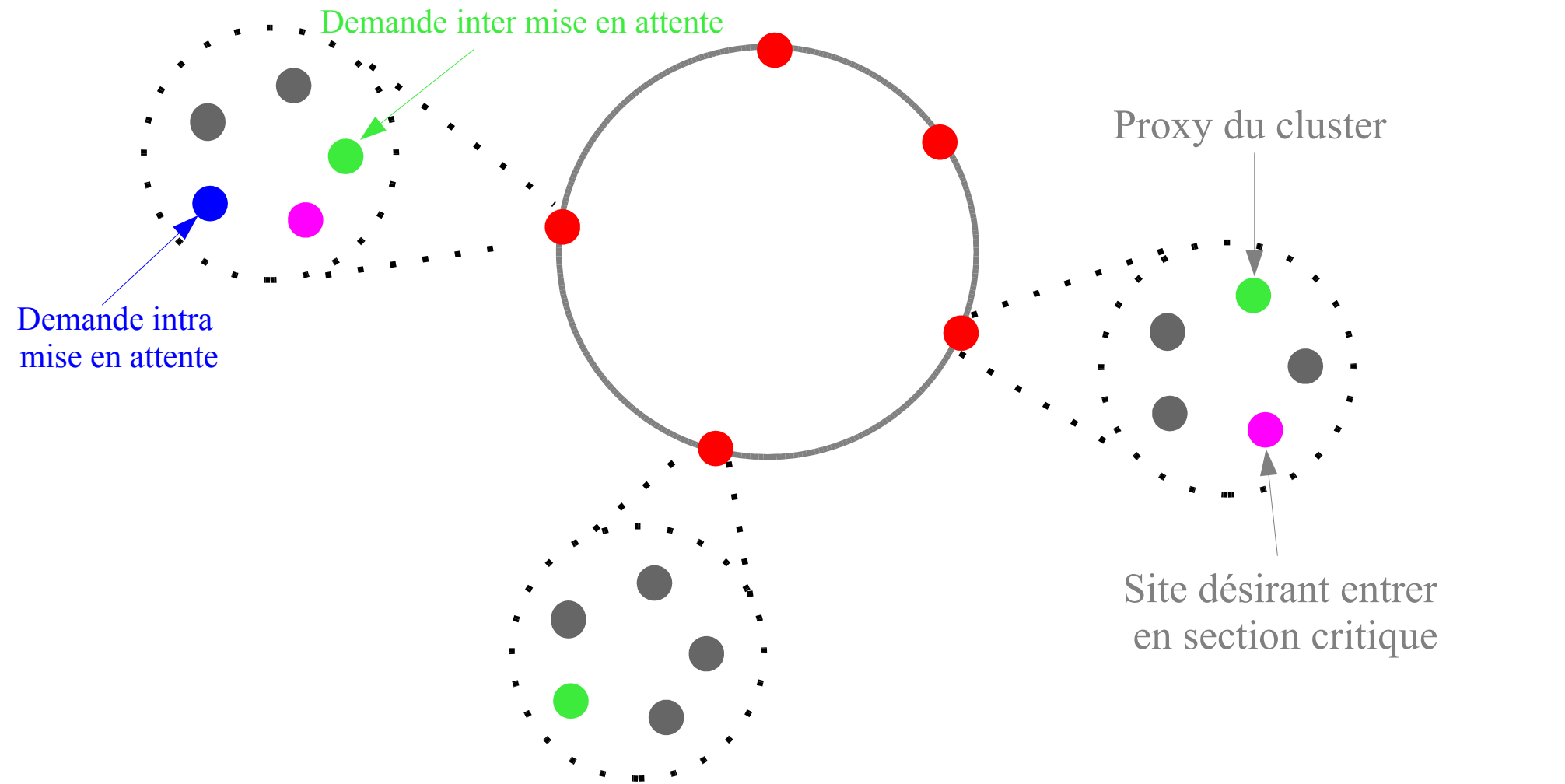
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



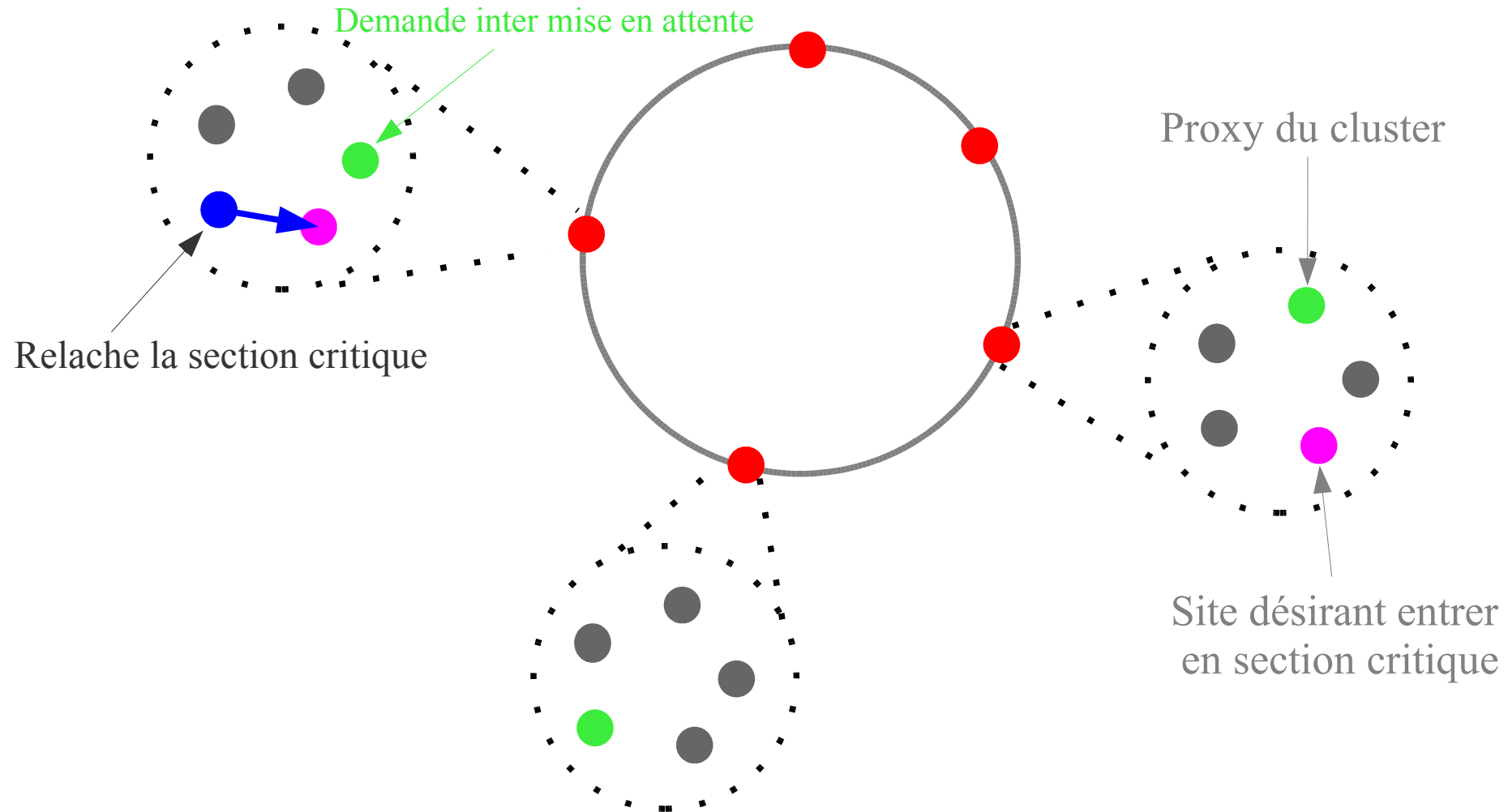
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



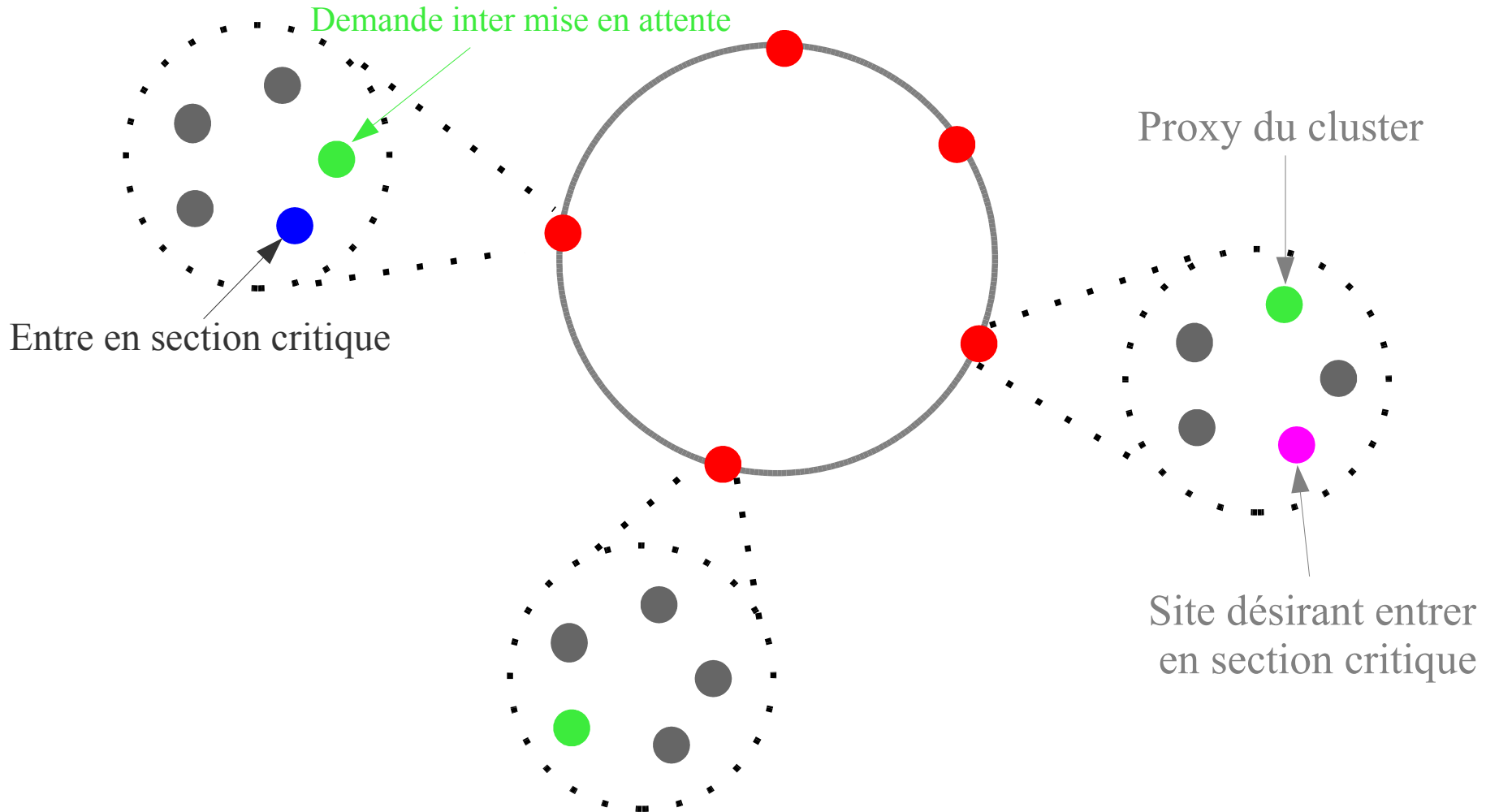
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



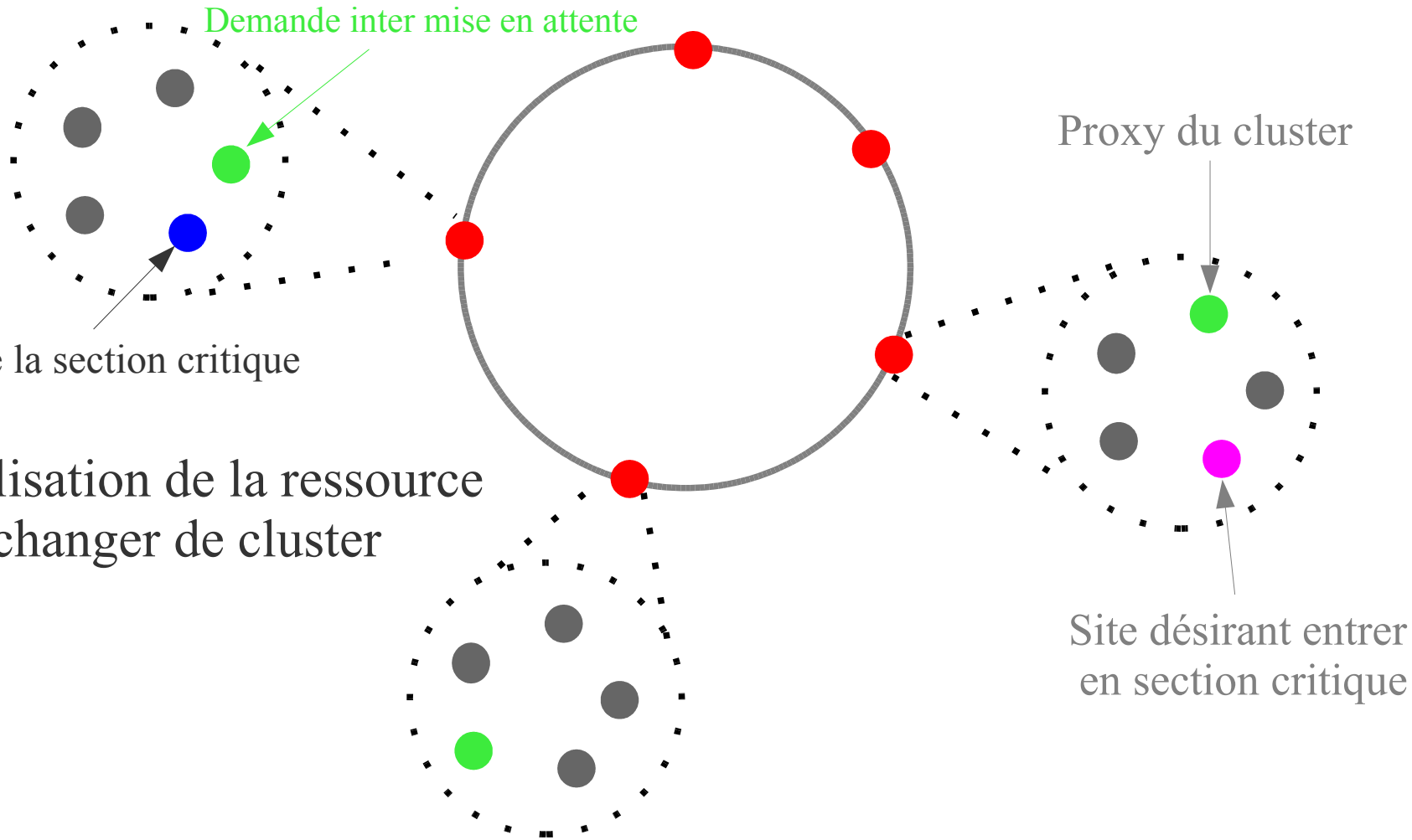
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



Relache la section critique

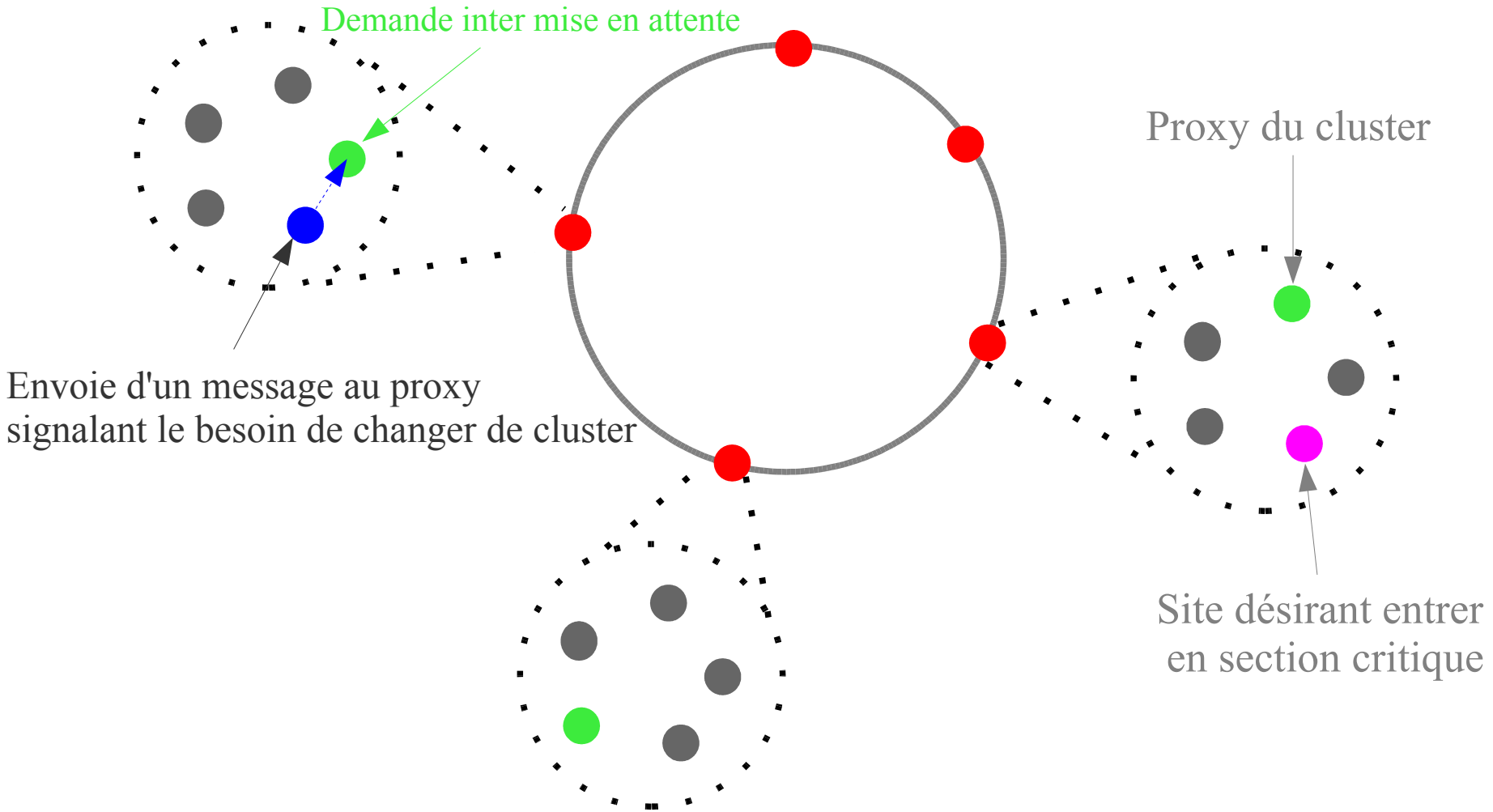
L'utilisation de la ressource doit changer de cluster

Proxy du cluster

Site désirant entrer en section critique

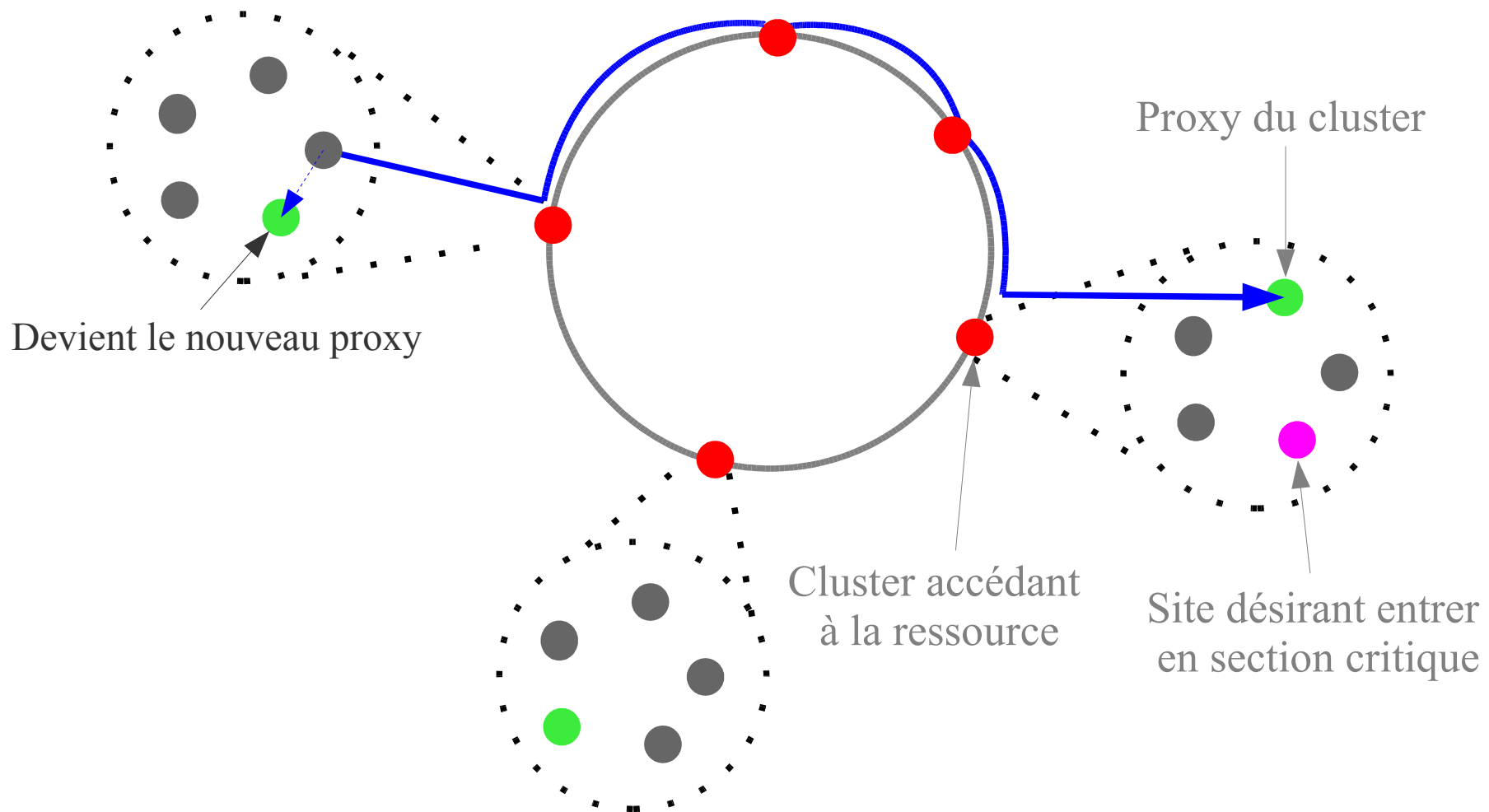
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



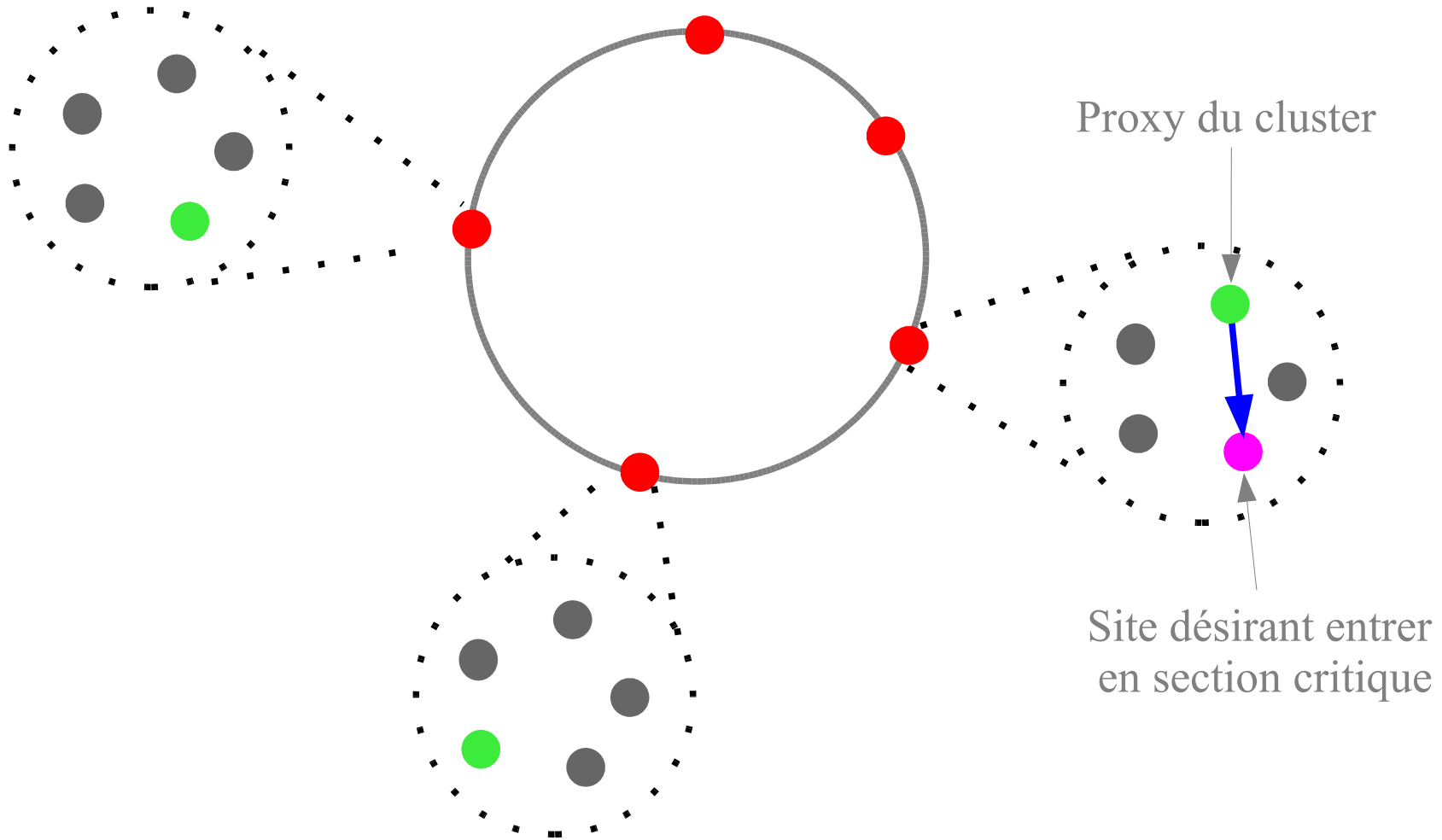
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation



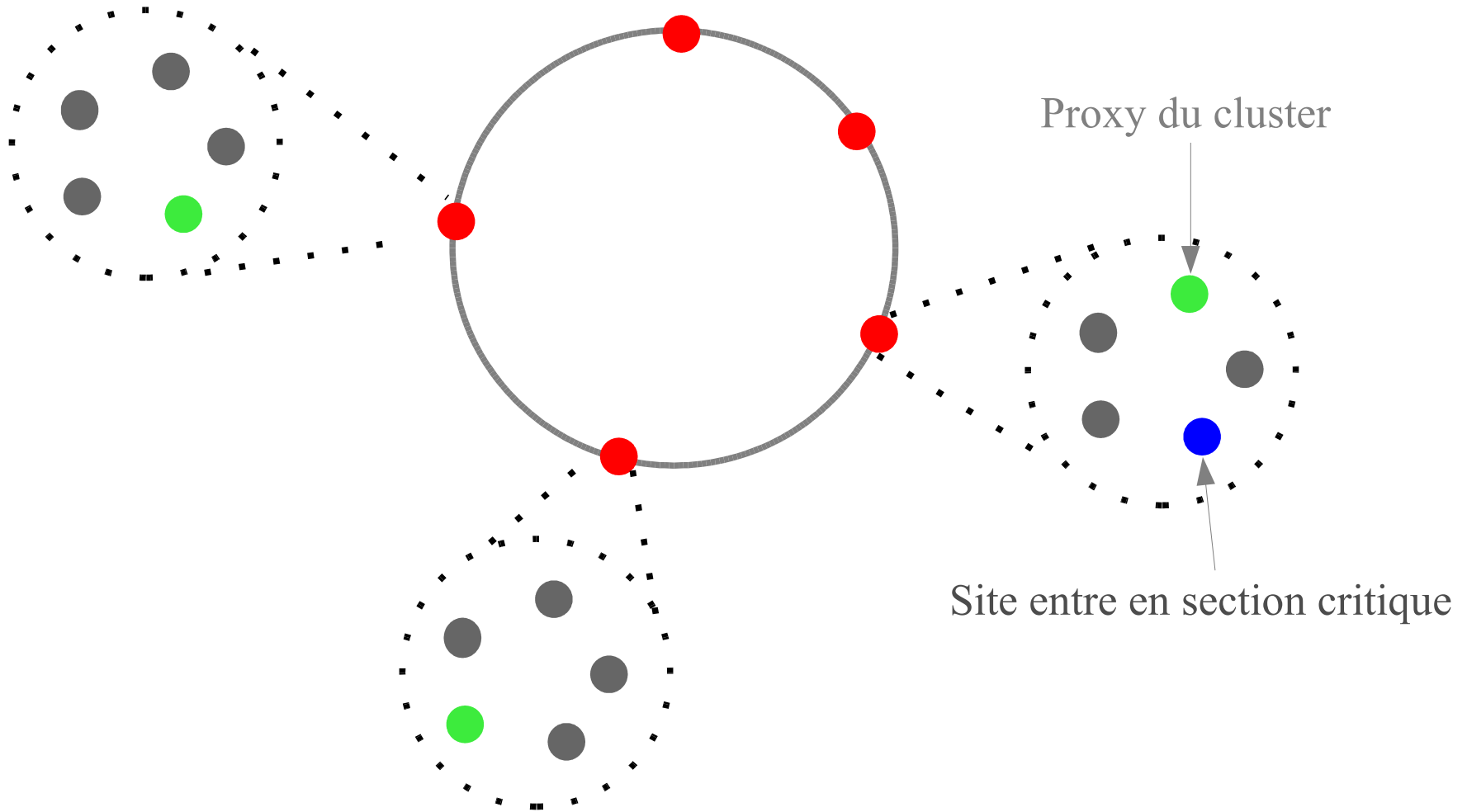
Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami

Exemple de réalisation





Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami


Exemple de réalisation





Algo inter-cluster: Martin
 Algo intra-cluster: Suzuki-Kasami


- 
 Approche totalement décentralisée, les sites n'occupent un rôle spécialisé que durant une période limitée.

- 
 L'information des changements de proxy est contenue dans les messages inter-cluster et ne nécessite donc pas de nouveaux messages.

- 
 Pour l'algorithme intra-cluster, le proxy temporaire est vu comme celui ayant accès à la ressource critique lorsque ce dernier est dans un autre cluster.

- 
 Approche générique permettant d'expérimenter une grande variété de combinaison d'algorithmes.

- 
 Approche compatible avec la tolérance aux fautes.

- 
 Permet d'envisager une architecture à plus de 2 niveaux avec possibilité de choisir l'algorithme utilisé sur chaque niveau de hierarchie.

Questions ?