Services et protocoles pour l'exécution fiable d'applications distribuées dans les grilles de calcul

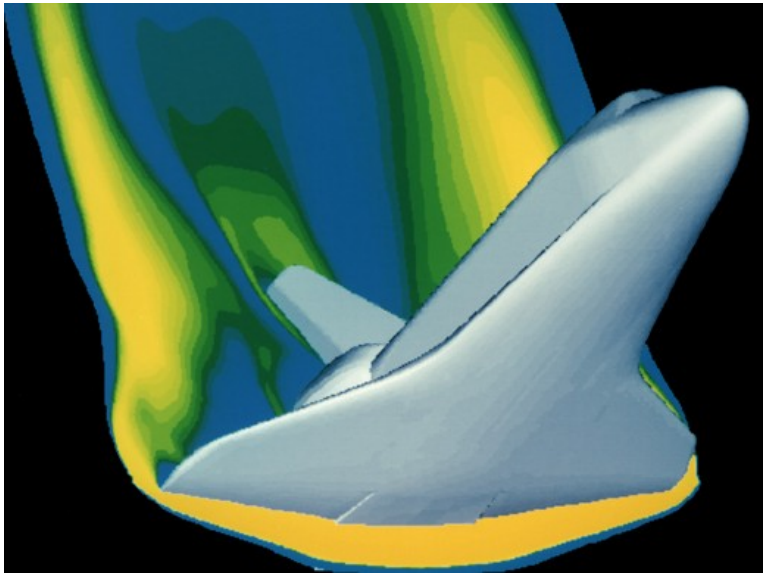
Thomas Ropars

Équipe-projet PARIS

INRIA IRISA UNIVERSITÉ DE RENNES 1

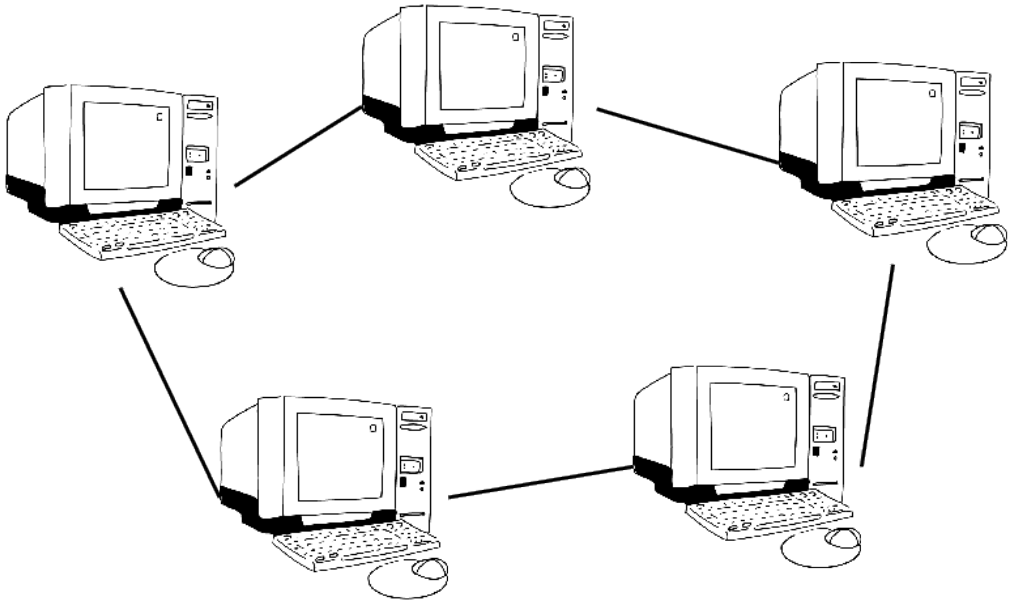
Les applications de calcul scientifique

- Objectifs :
  - Simuler des phénomènes physiques complexes
    - Météorologie
    - Dynamique des fluides
  - Analyser de grandes quantités de données
    - Génomique



Des besoins en ressources de calcul toujours plus grands

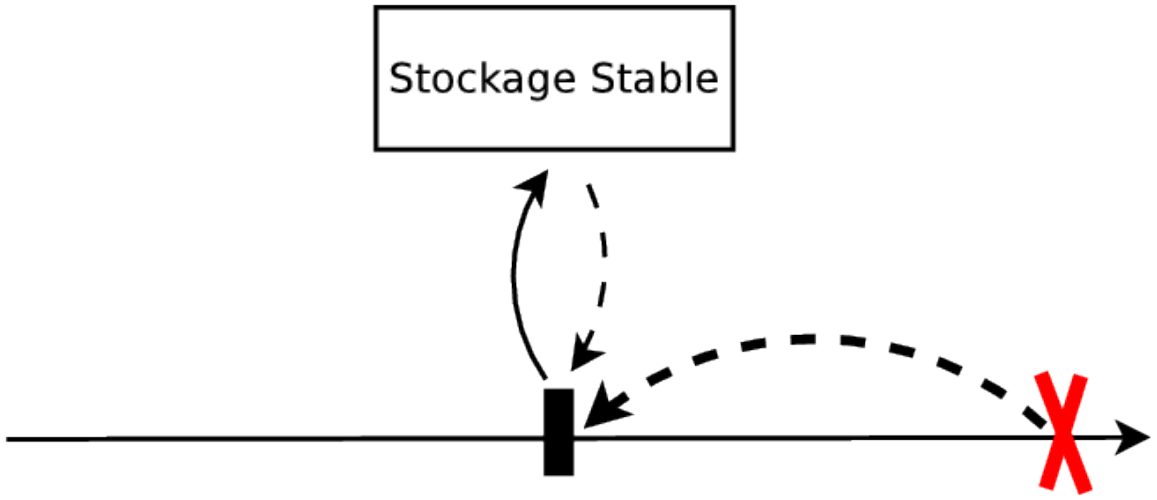
- Des systèmes distribués
  - Machines physiques interconnectées (nœuds)



- Des applications distribuées
  - Plusieurs processus s'exécutant en parallèle

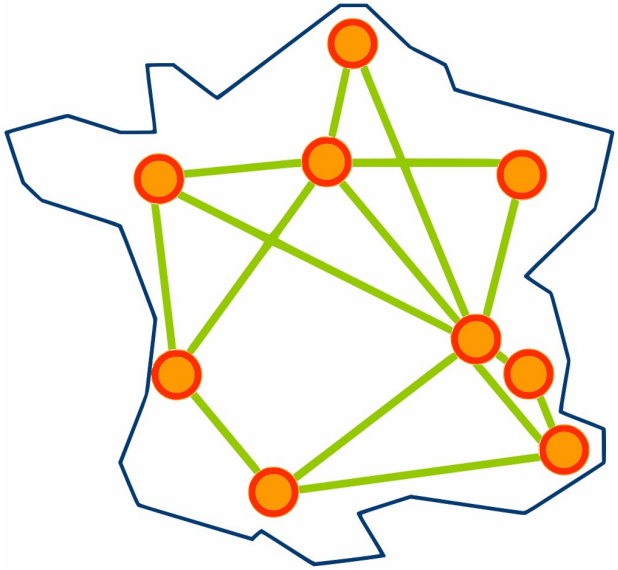
Des défaillances fréquentes

- Plus le nombre de ressources composant un système distribué est grand, plus la probabilité de subir des défaillances est élevée.
  - Plusieurs dizaines de milliers de nœuds
  - Temps moyen entre deux défaillances = quelques heures
- Besoin de mécanismes de tolérance aux fautes
  - Recouvrement arrière

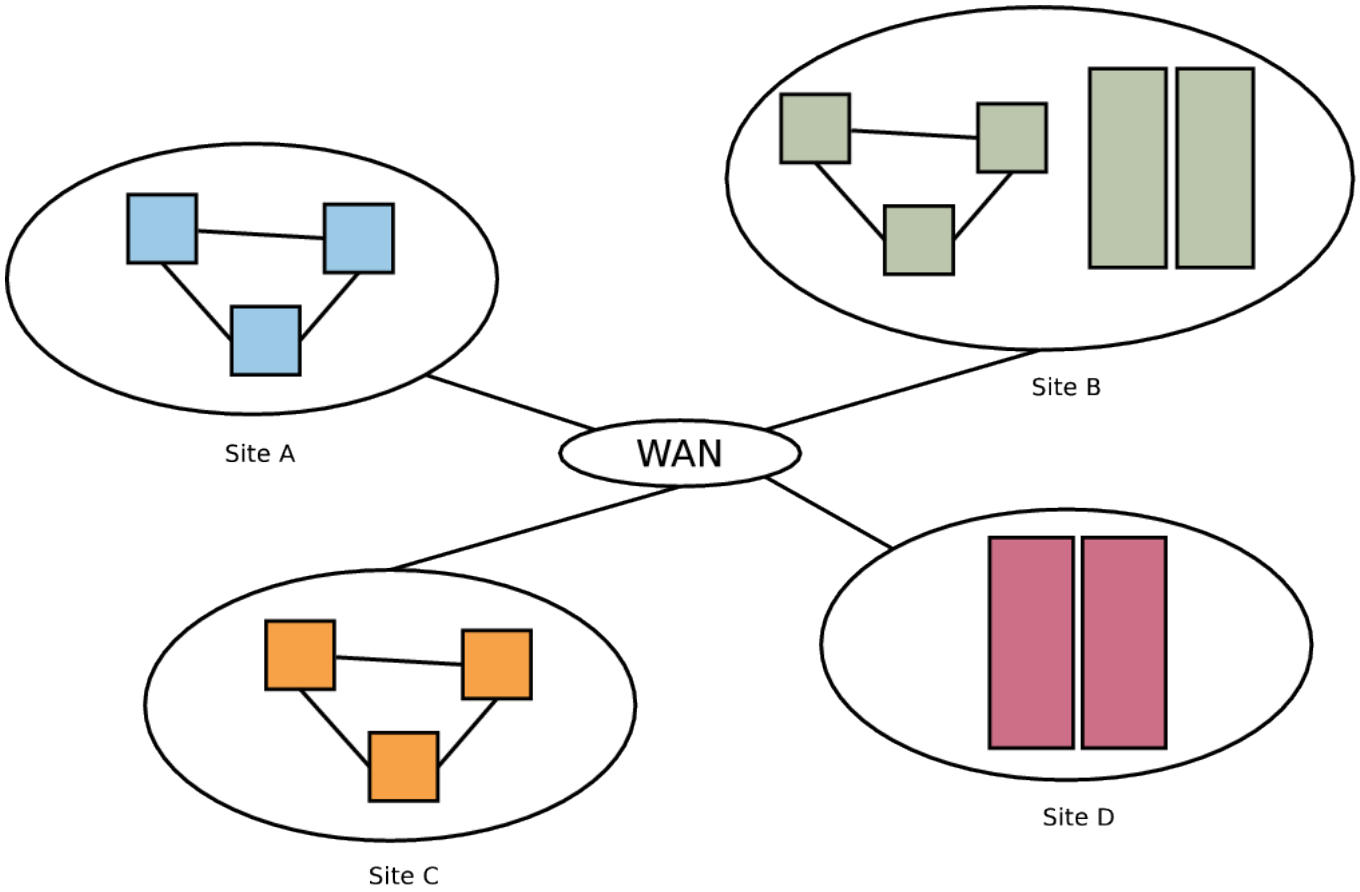


Les grilles de calcul

- Partage de ressources entre plusieurs institutions
- Système distribué regroupant des ressources pouvant appartenir à différents domaines d'administration

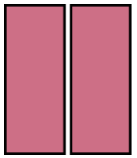


Les grilles de calcul



- Grande taille
- Hétérogénéité
- Volatilité

• Ordinateurs personnels 

• Grappes de calcul 

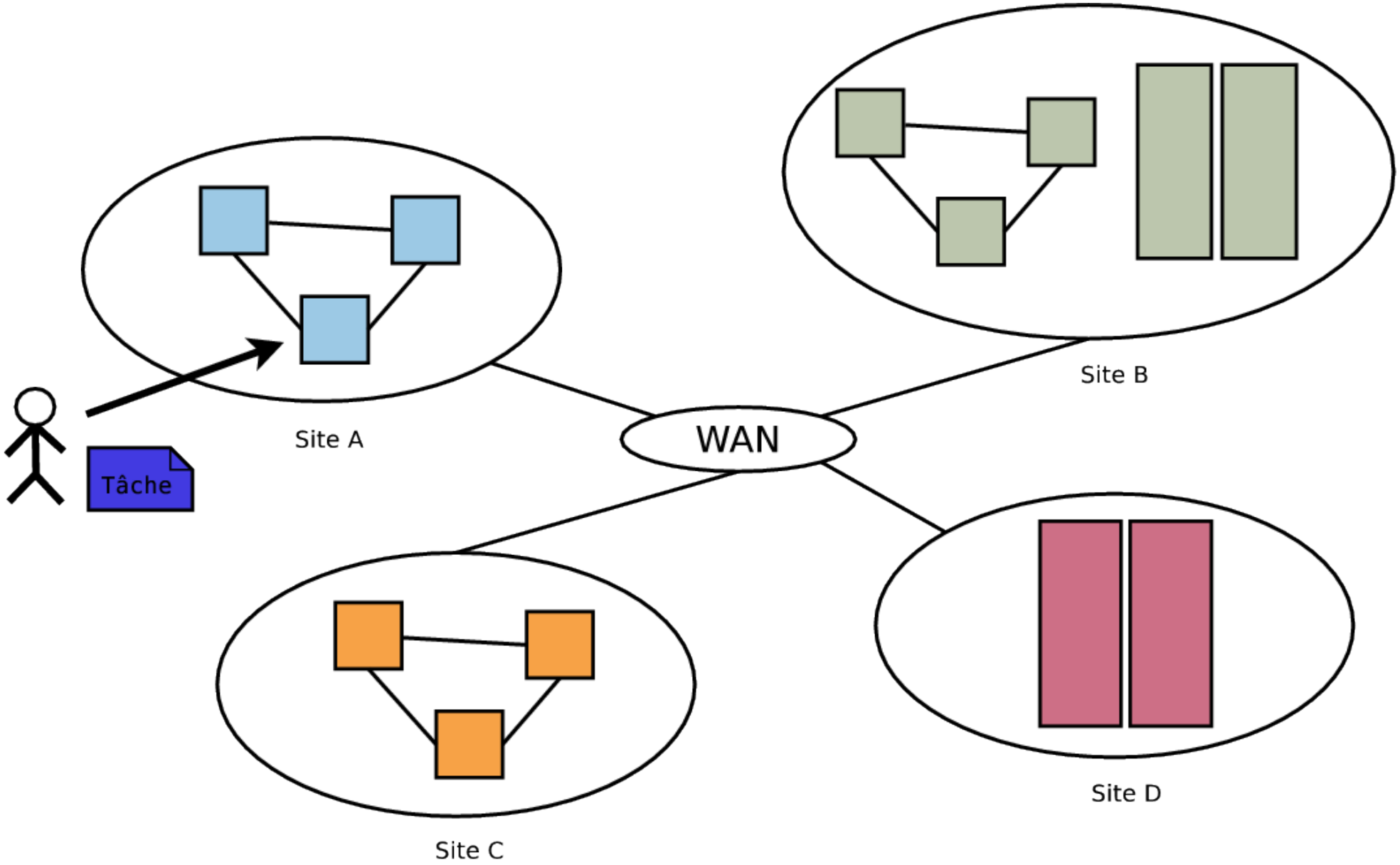
Les systèmes de grille

- Ensemble de services pour simplifier l'utilisation des grilles de calcul
  - XtreemOS
    - Projet européen

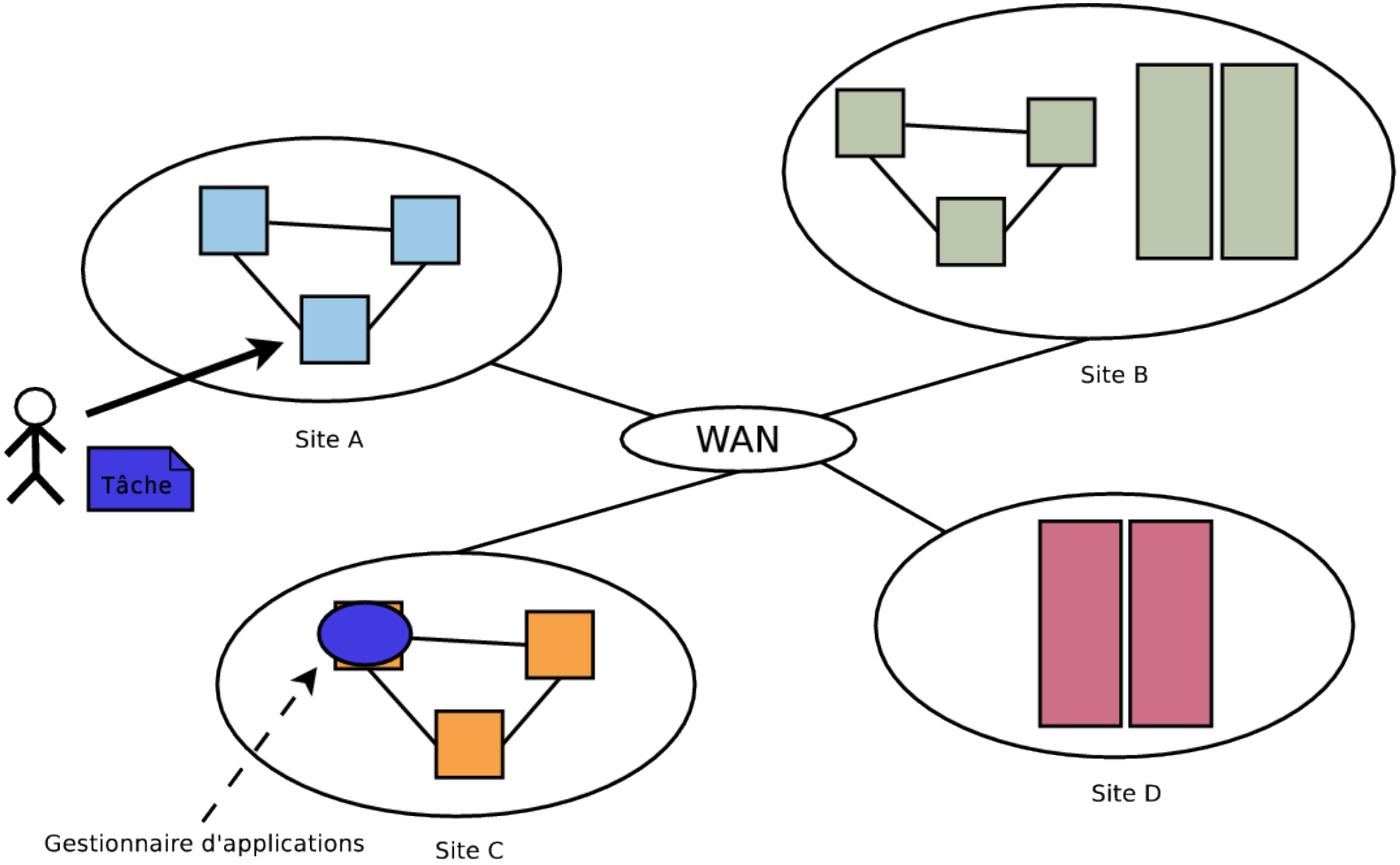
  - Vigne
    - Équipe-projet PARIS
    - EDF R&D

- Exemple :
  - Un service de gestion des applications

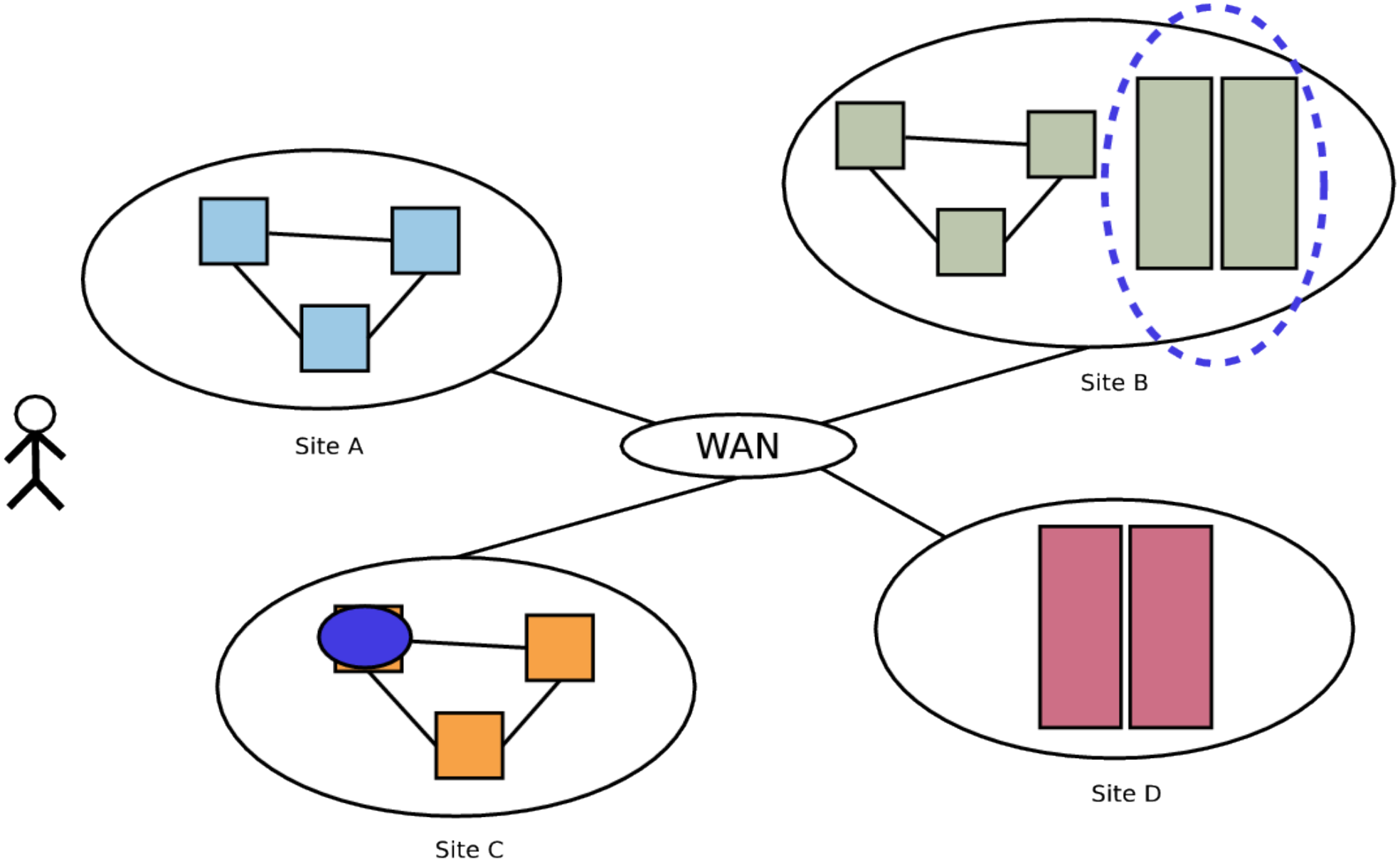
Soumission d'une application par l'utilisateur



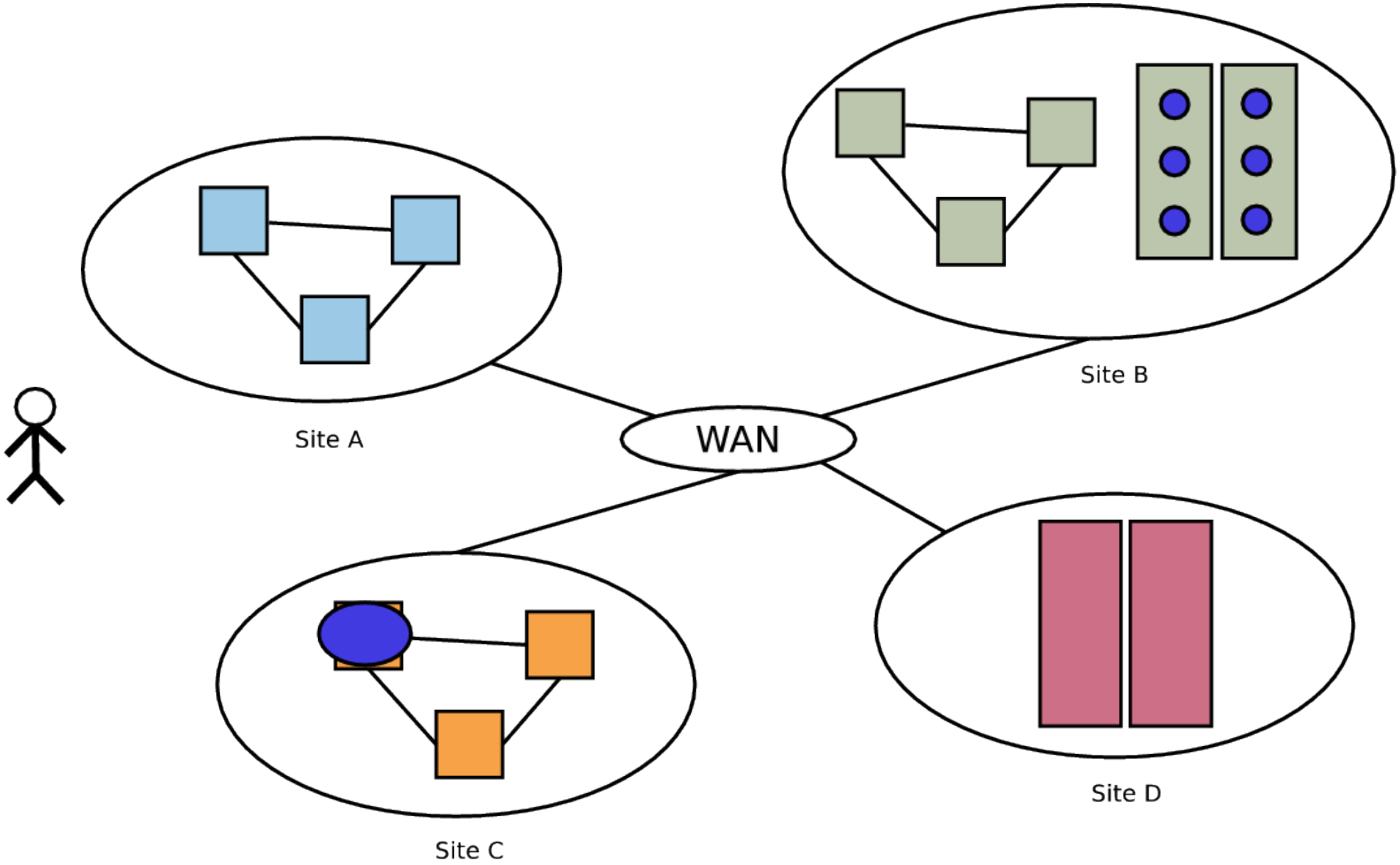
Soumission d'une application par l'utilisateur



Découverte et allocation de ressources

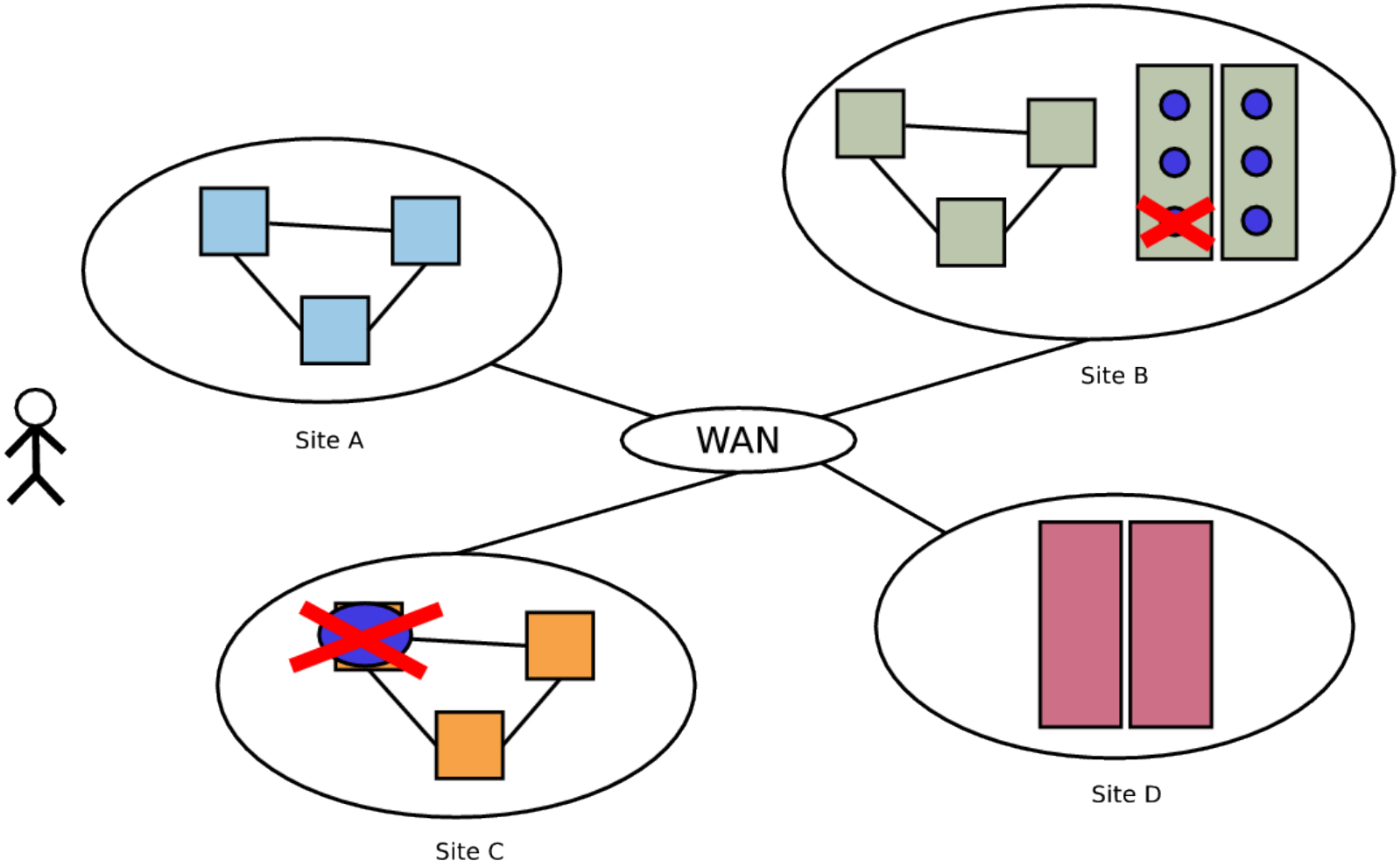


Exécution de l'application



Objectifs

- Exécution fiable des applications et des services
  - Assurer la bonne terminaison des applications
  - Assurer la disponibilité des services critiques

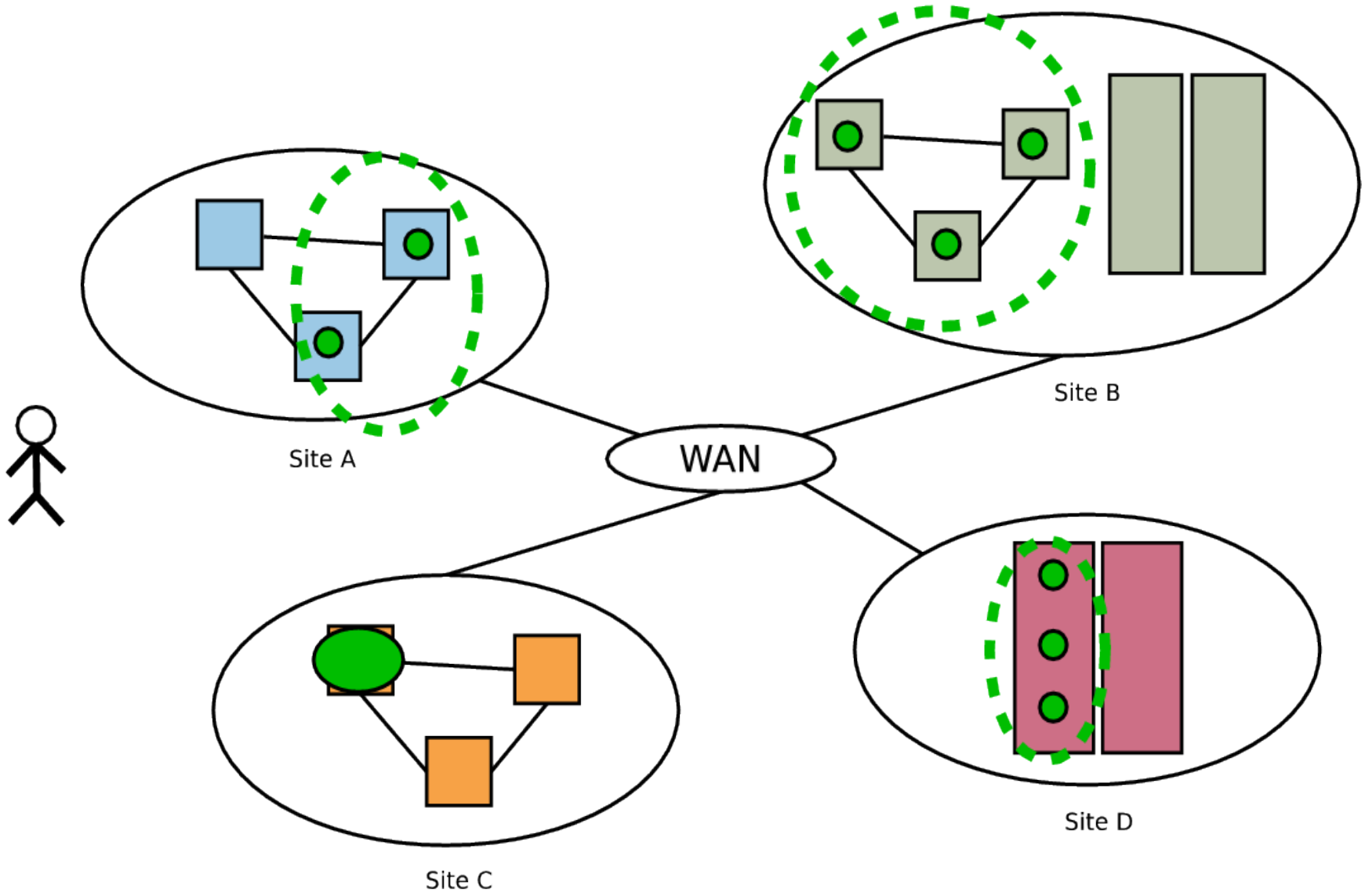


Objectifs

- Solutions adaptées aux caractéristiques des grilles
  - Hétérogénéité
  - Grande taille
- Simplicité d'utilisation
  - Traitement automatique des défaillances
- Coût
  - Ressources consommées
  - Performances des applications

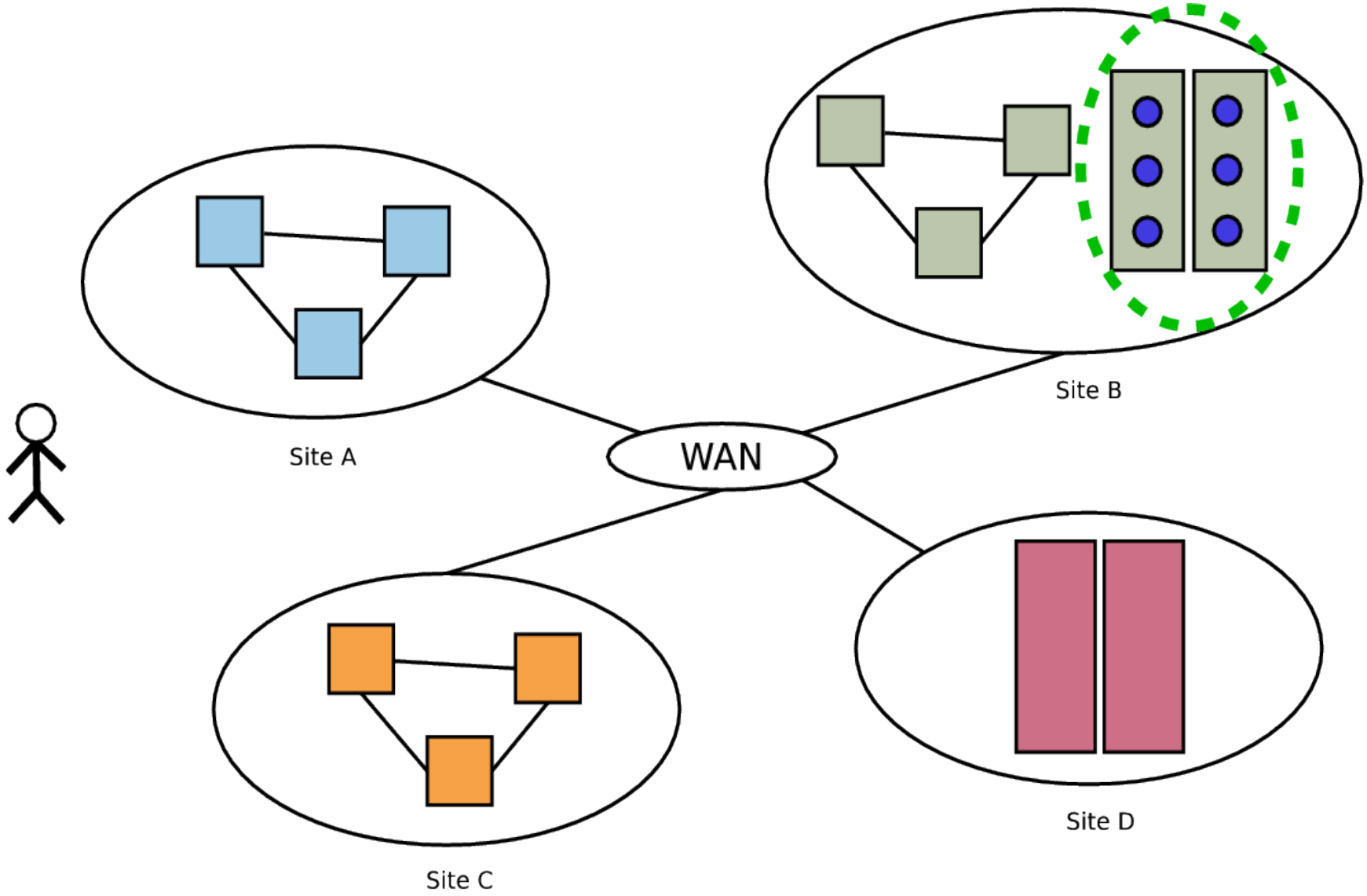
Contributions

XtreemGCP : Un service générique de recouvrement arrière pour le redémarrage automatique d'applications distribuées



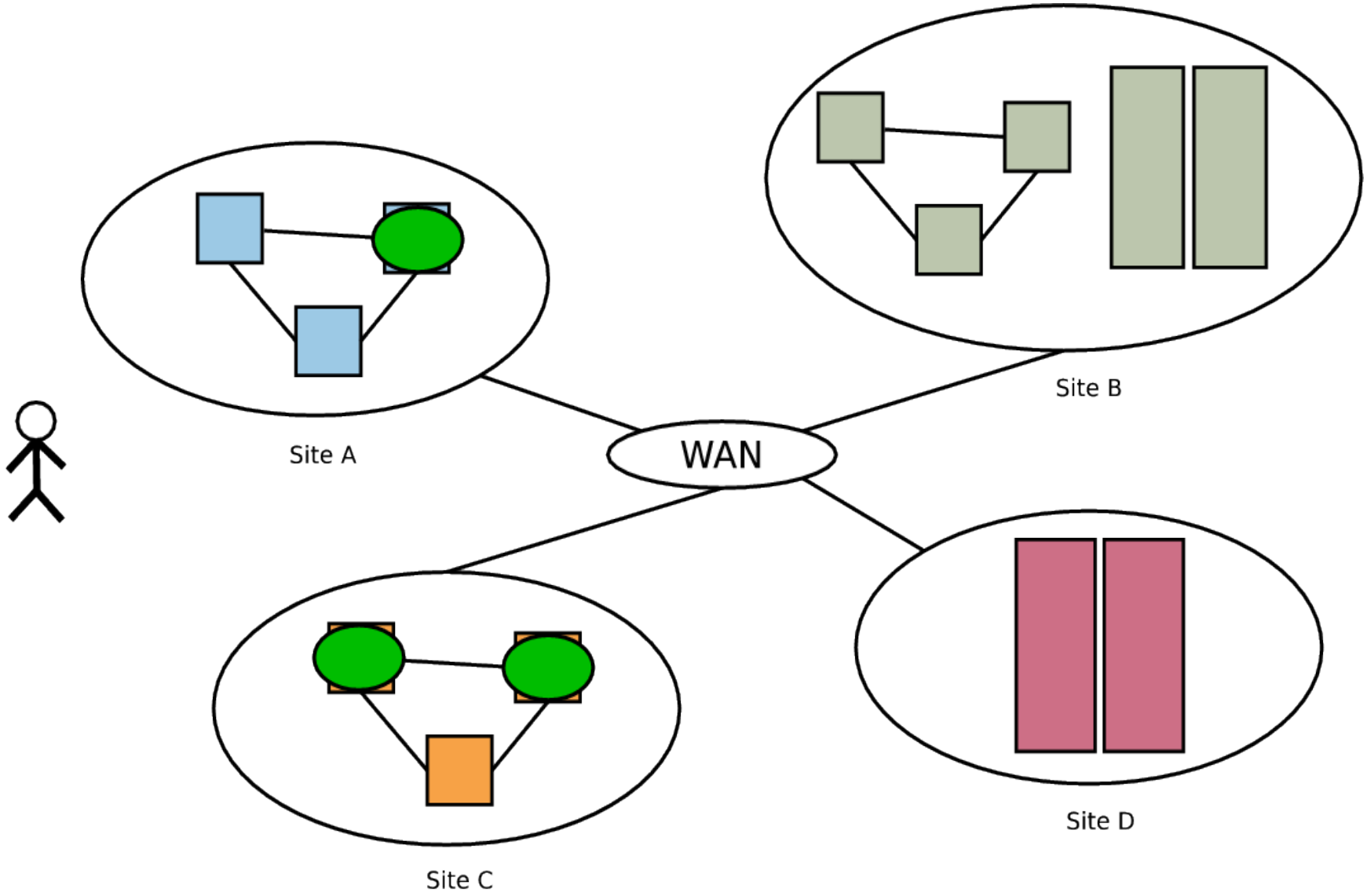
Contributions

O2P : une solution de recouvrement arrière passant à l'échelle pour applications à échange de messages

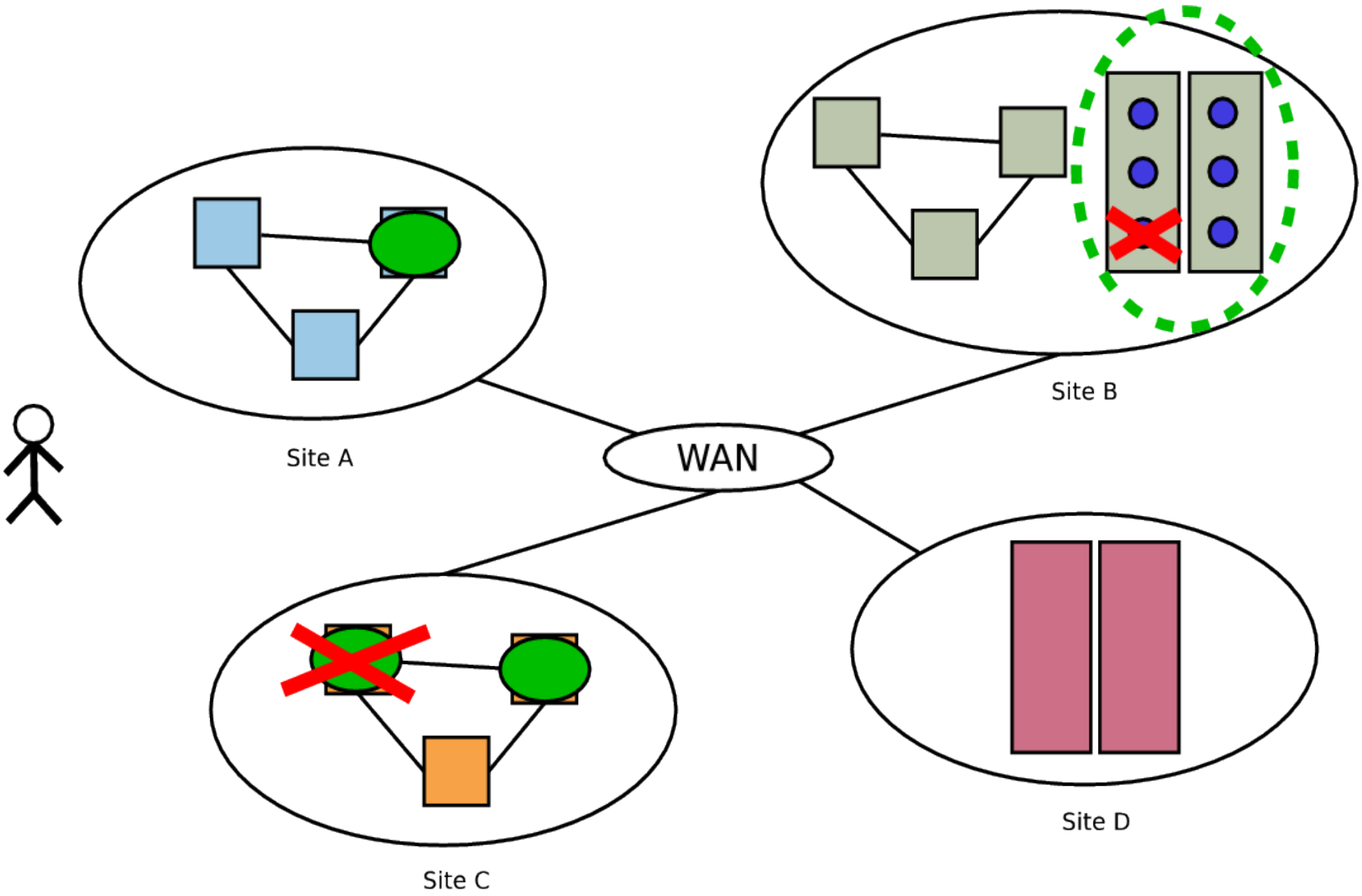


Contributions

Semias : un cadre pour la haute disponibilité et l'auto-réparation de services de grille



Contributions



Modèle de fautes

- Système distribué asynchrone
- Canaux fiables et FIFO
- Fautes par arrêt total
  - Possibilité de fautes corrélées
- Pas de fautes byzantines

Semias

Haute disponibilité de services de grille

Stefania Costache, Rajib Nath, Sébastien Gillot

Semias

- Objectifs
  - Haute disponibilité des services de grilles
  - Simplicité d'utilisation
    - Auto-réparation
    - Transparence des reconfigurations
  - Simplicité de programmation
    - Pas de modification des services

- Approche
  - Duplication active de services dans un réseau logique structuré

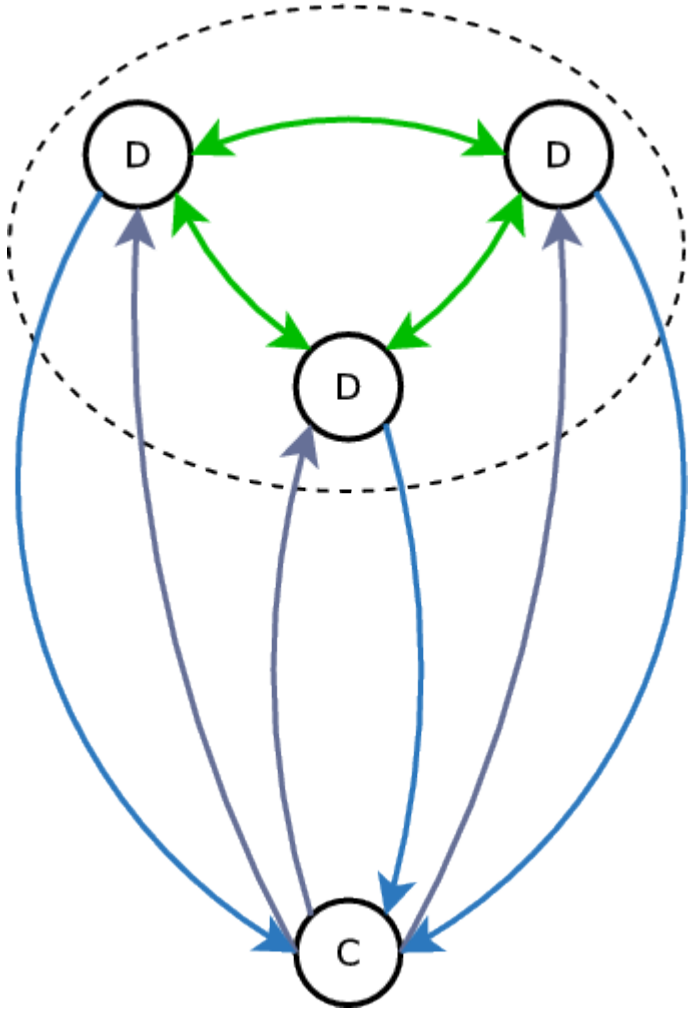
Semias

- Principes de fonctionnement
- Auto-réparation des services
- Évaluation
- Travaux apparentés

Duplication Active

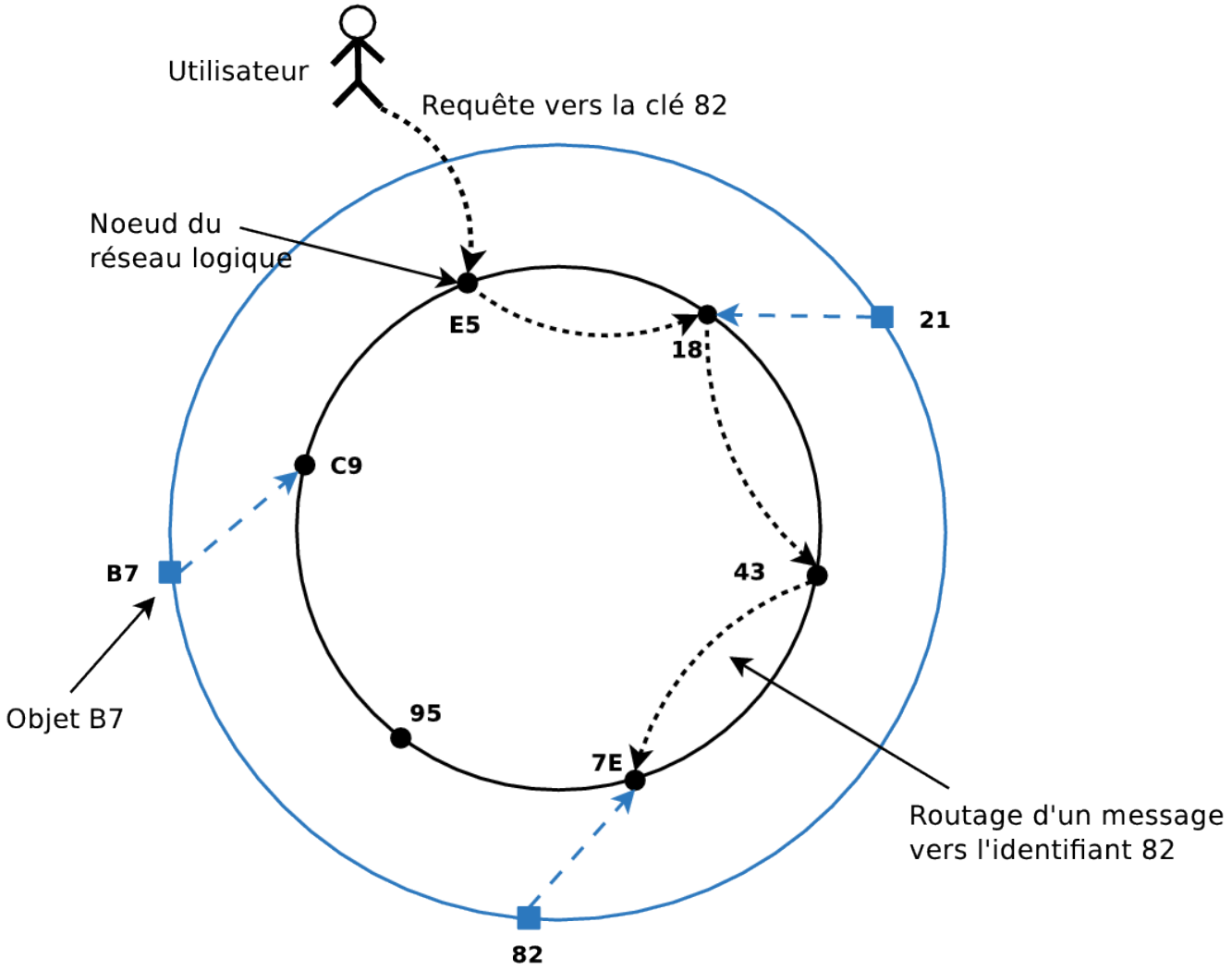
- Tous les duplicatas d'un service traitent les requêtes

- Cohérence des duplicatas
  - Services déterministes
  - Diffusion atomique des requêtes
    - Algorithme de consensus

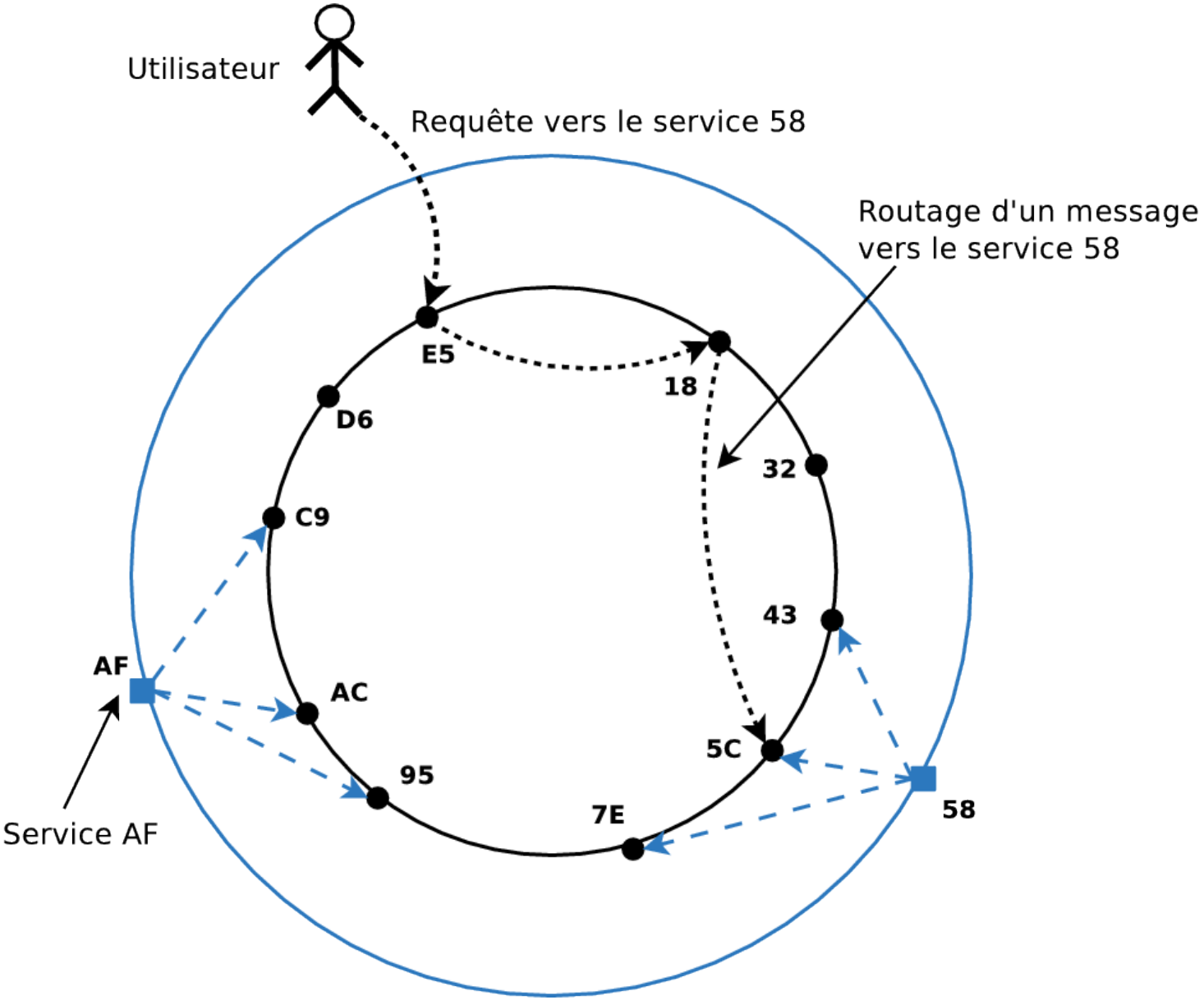


Les réseaux logiques structurés

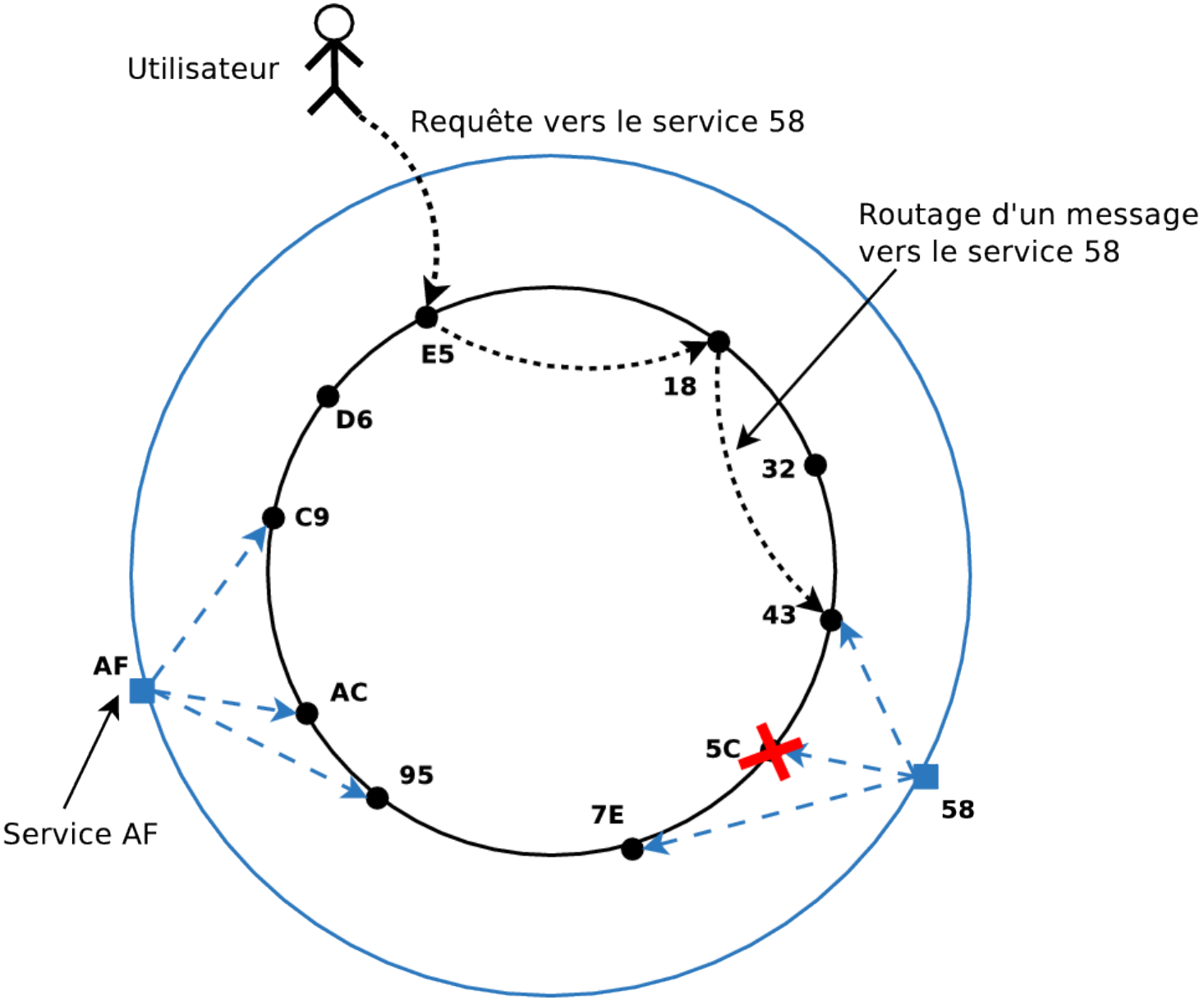
- Routage passant à l'échelle et tolérant aux fautes
- Adressage des objets par une clé



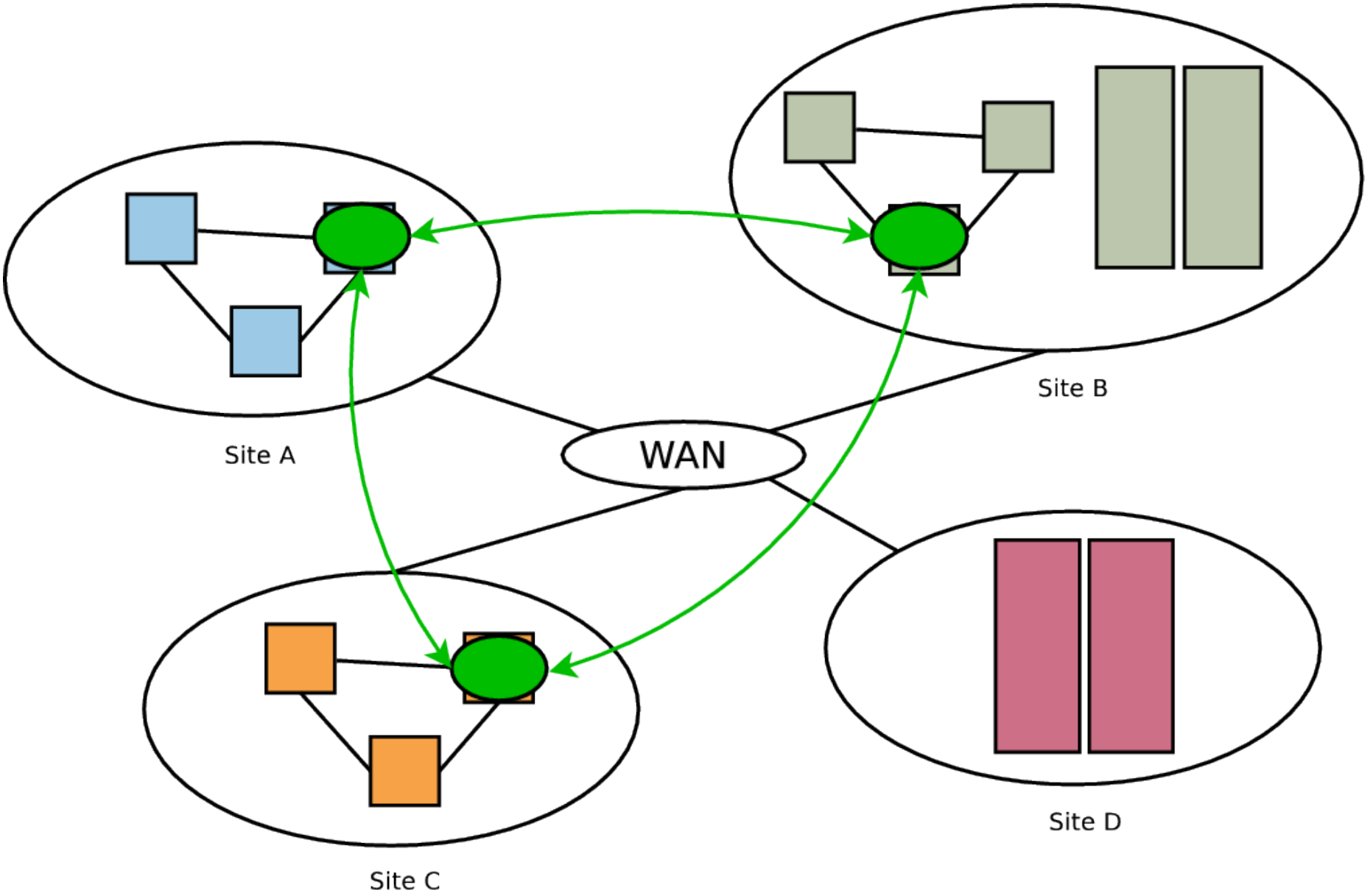
Duplication active de services dans un réseau logique structuré



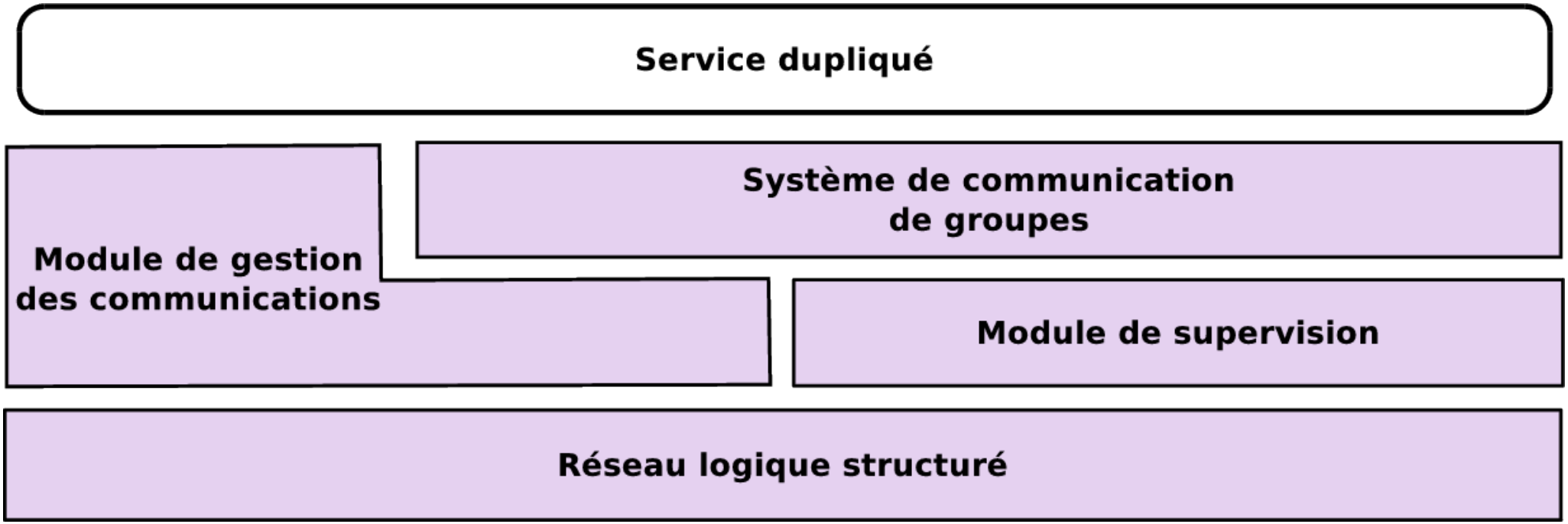
Duplication active de services dans un réseau logique structuré



Duplication active de services dans un réseau logique structuré

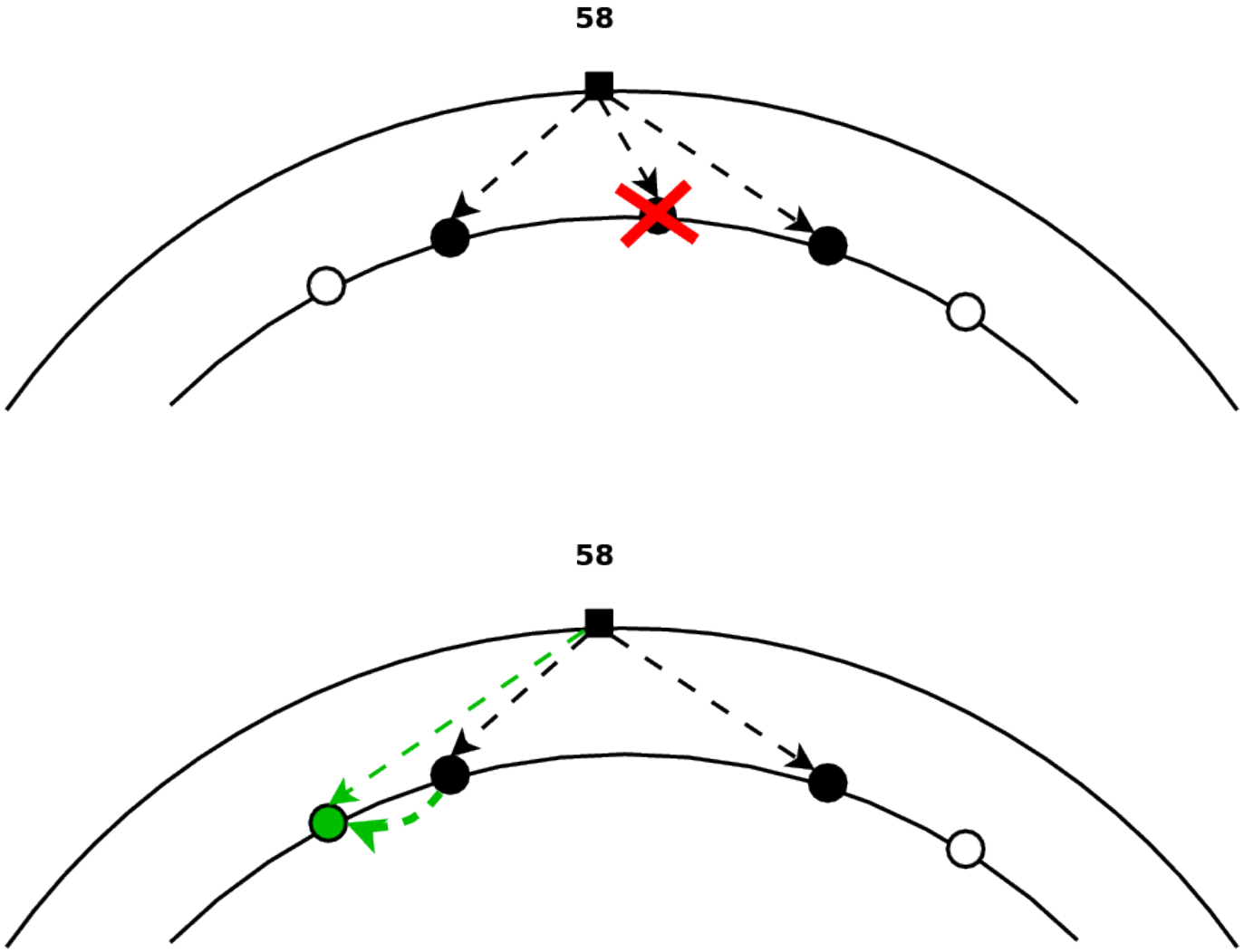


Architecture de Semias



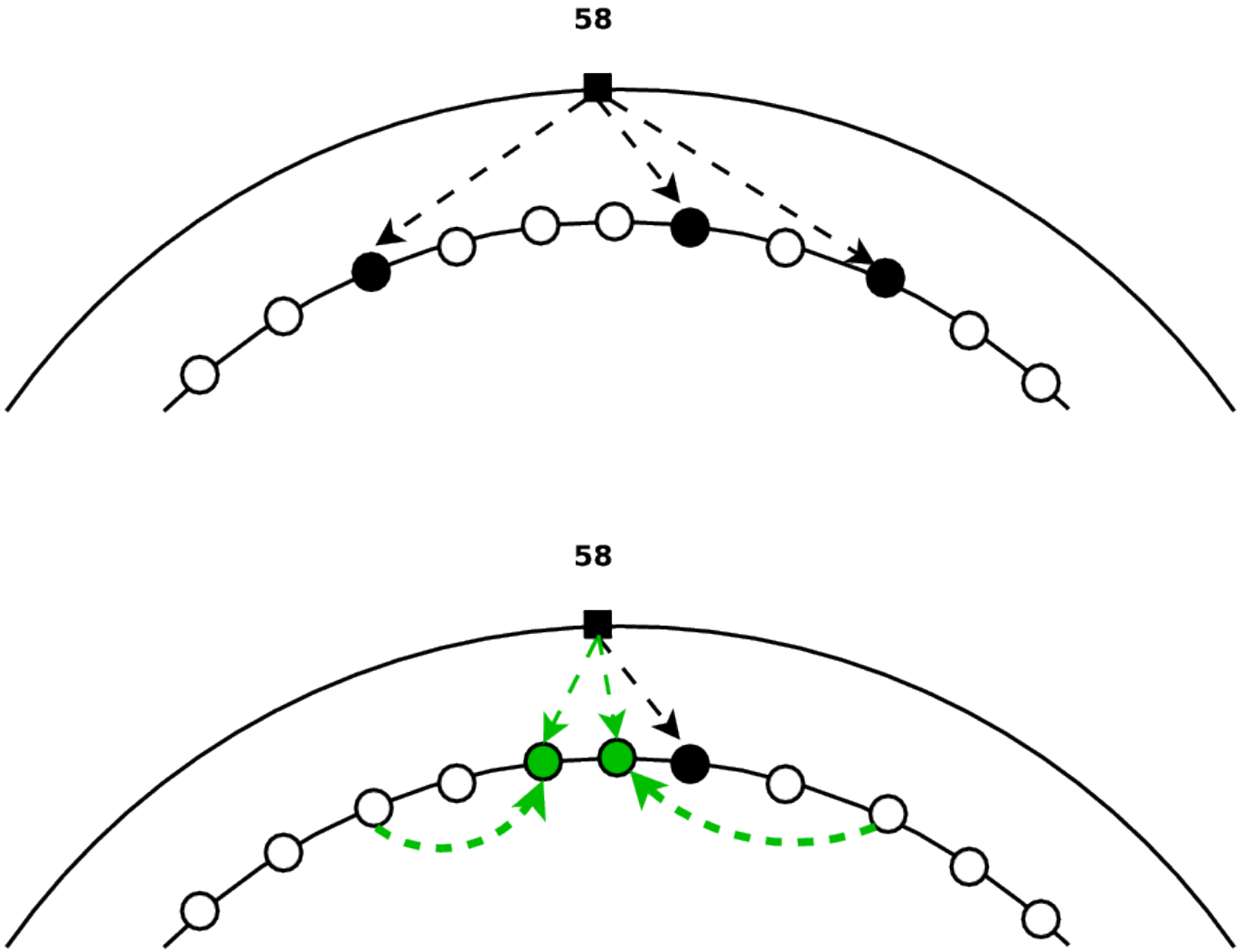
Auto-réparation

- Perte de duplicatas

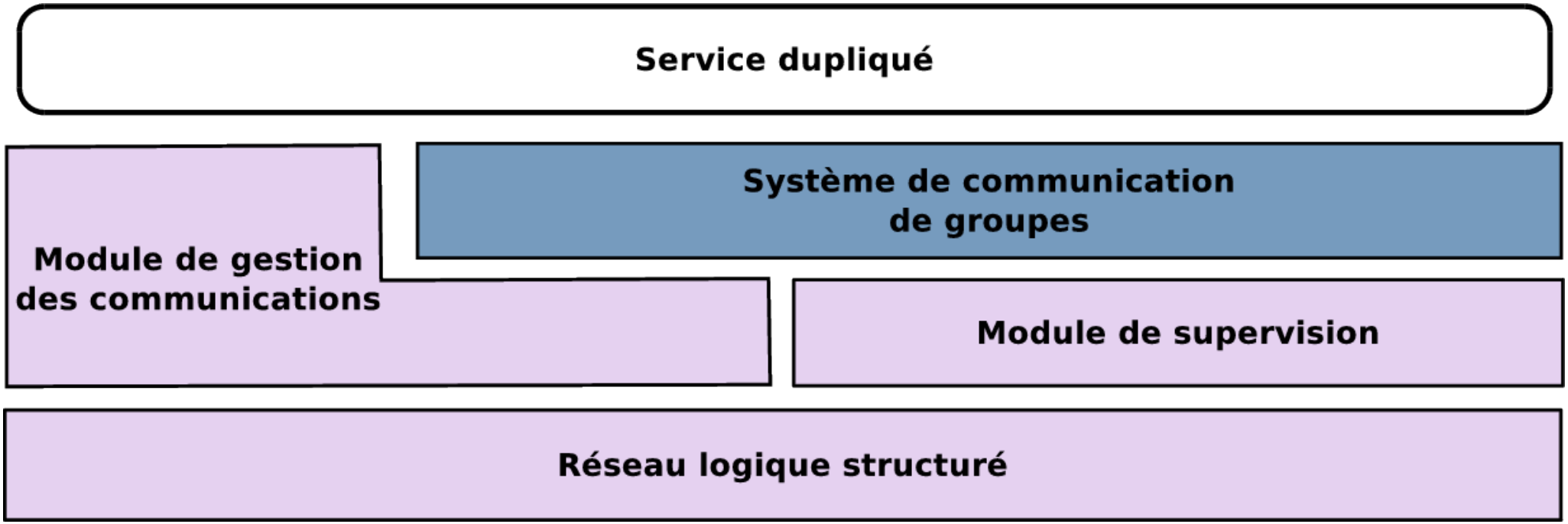


Auto-réparation

- Ajout de nouveaux nœuds

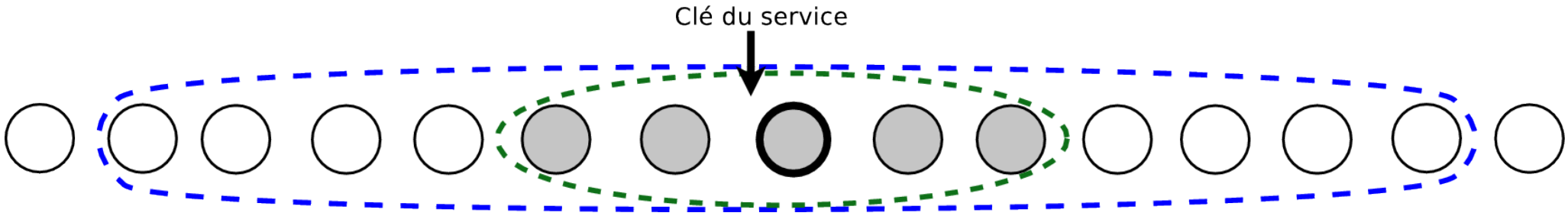


Auto-réparation

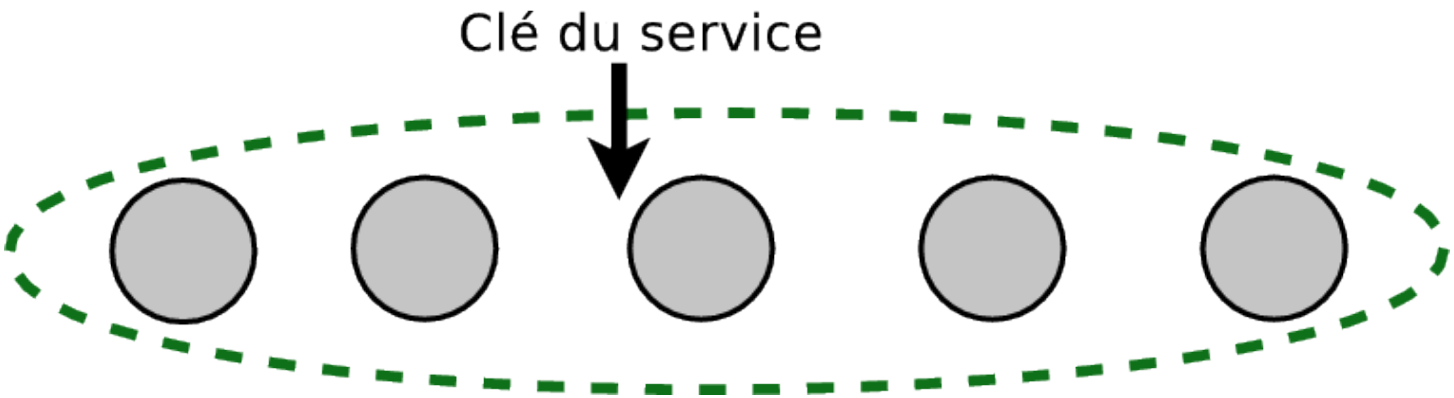


Évaluation de l'état du système

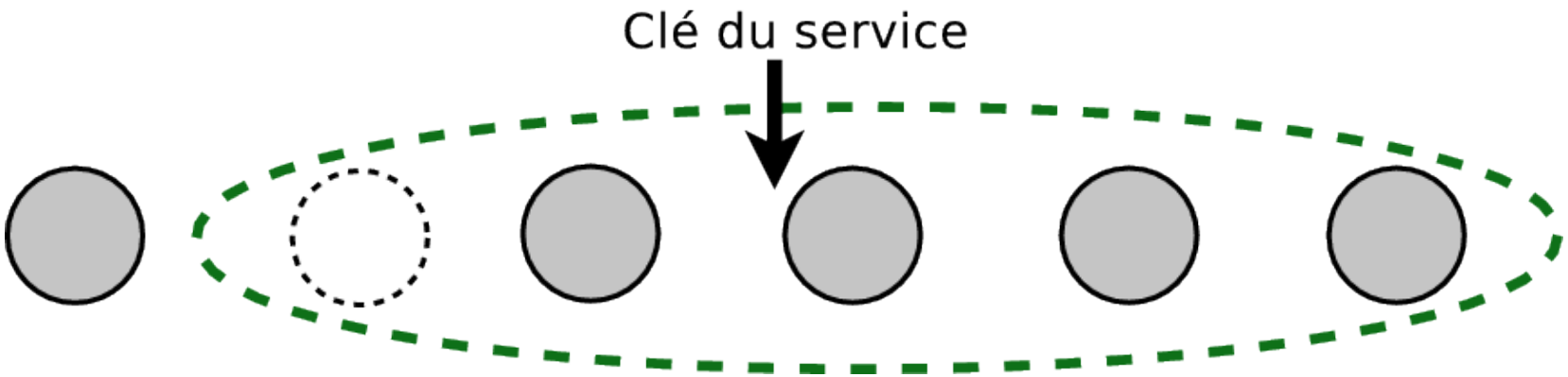
- Exploitation des algorithmes de maintenance des tables de routage du réseau logique
  - Défaillance d'un nœud
  - Arrivée d'un nœud



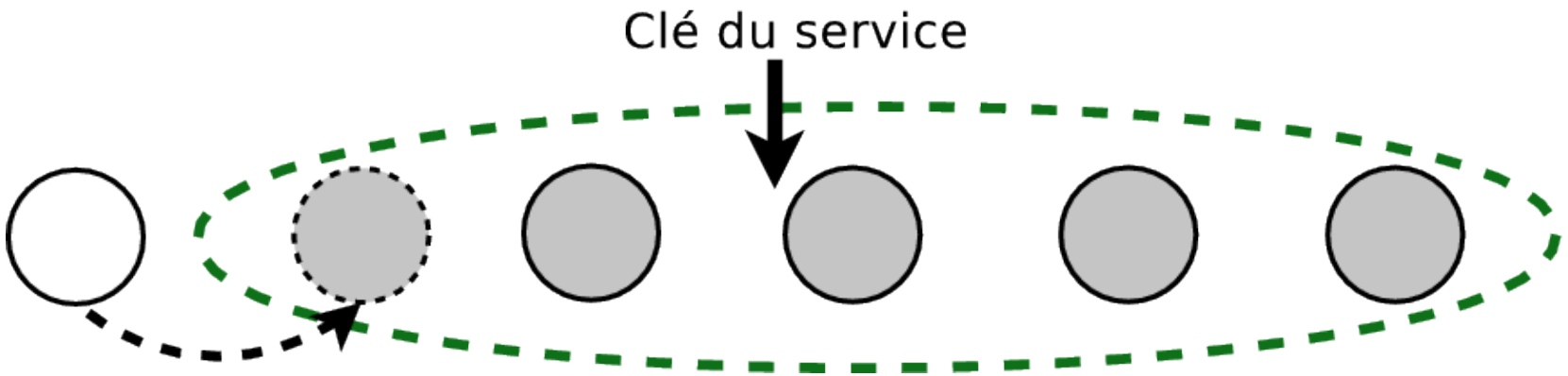
Limiter le nombre de reconfigurations



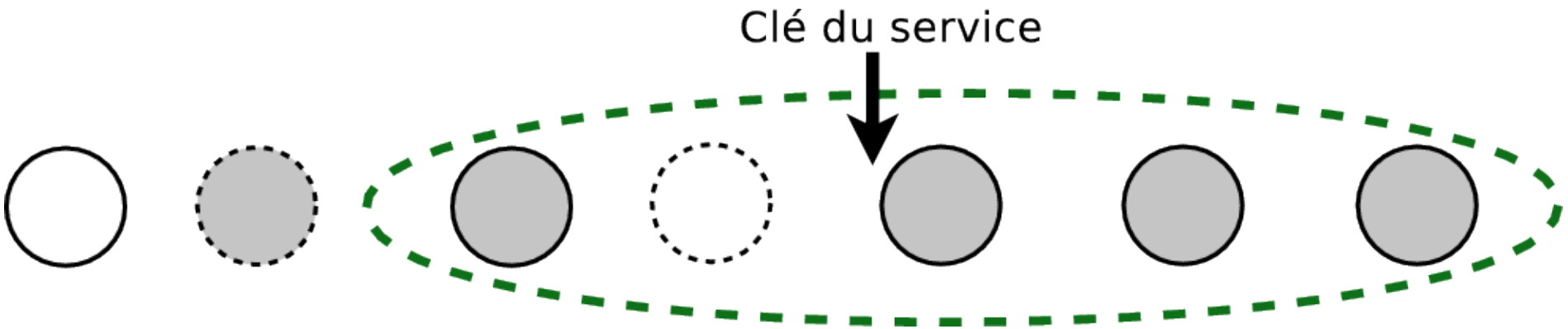
Limiter le nombre de reconfigurations



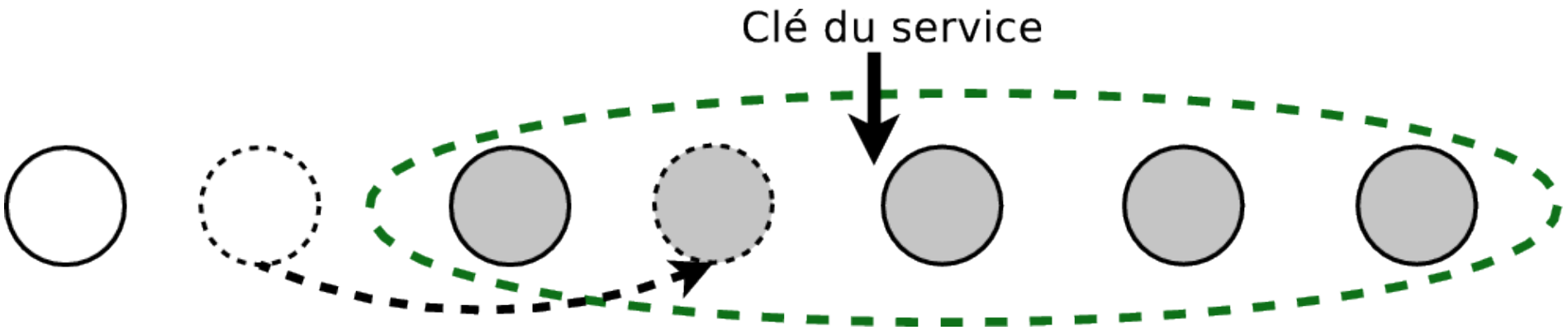
Limiter le nombre de reconfigurations



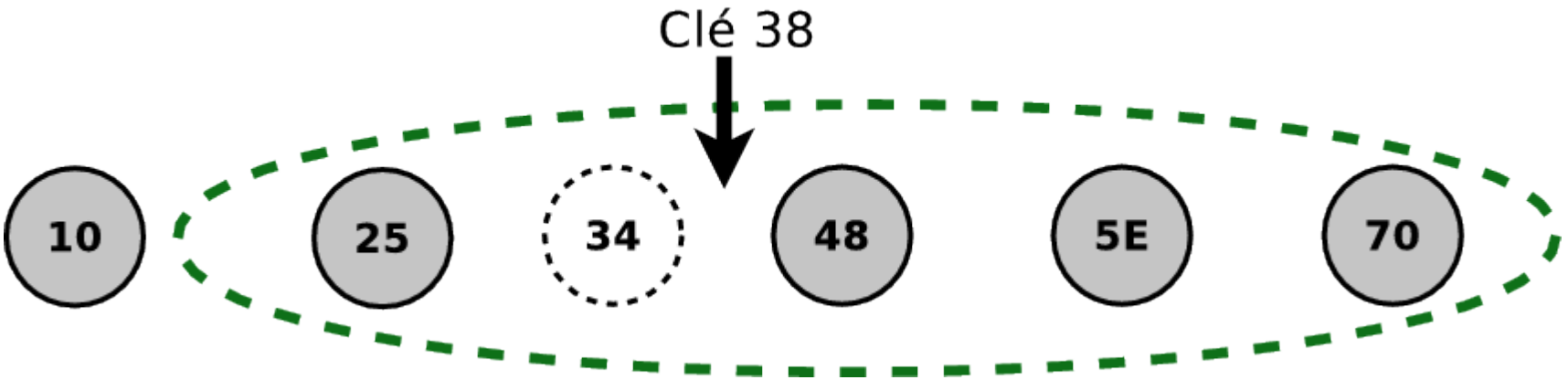
Limiter le nombre de reconfigurations



Limiter le nombre de reconfigurations

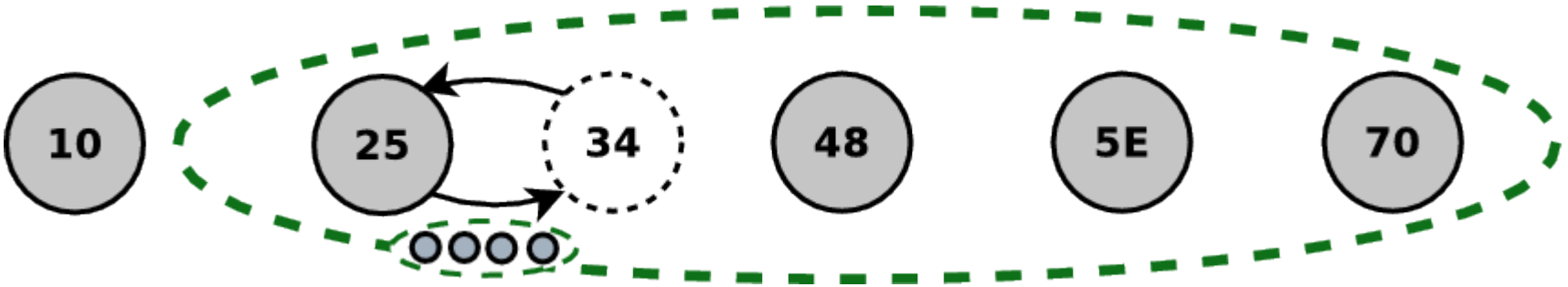


Des reconfigurations périodiques



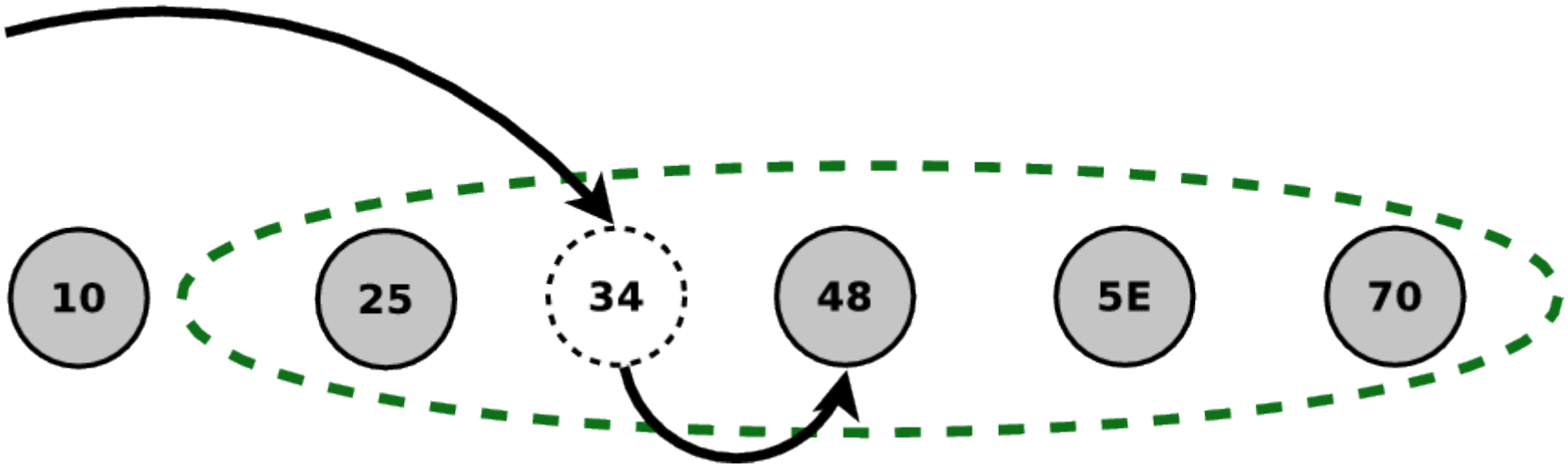
Utilisation de nœuds retransmetteurs

Des reconfigurations périodiques



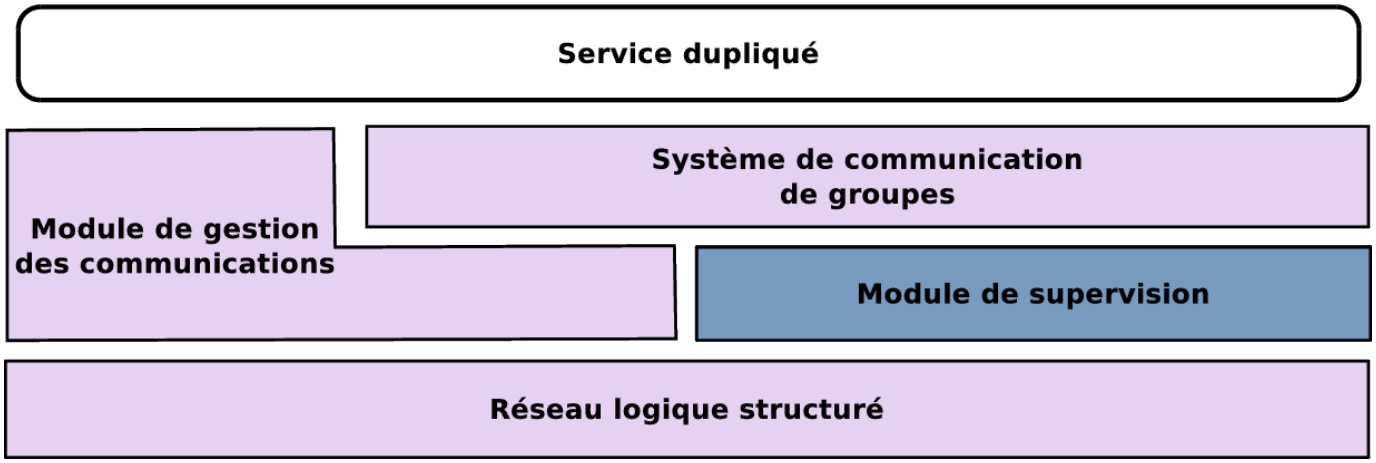
Utilisation de nœuds retransmetteurs

Des reconfigurations périodiques



Utilisation de nœuds retransmetteurs

Module de supervision

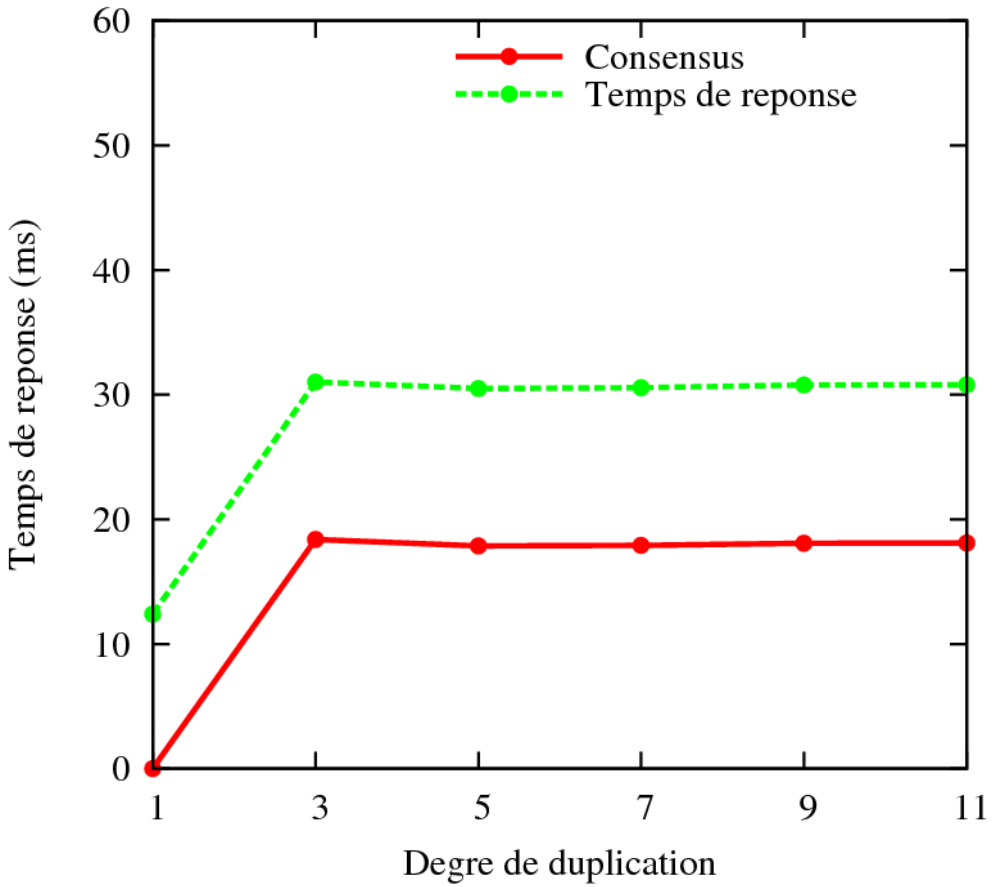


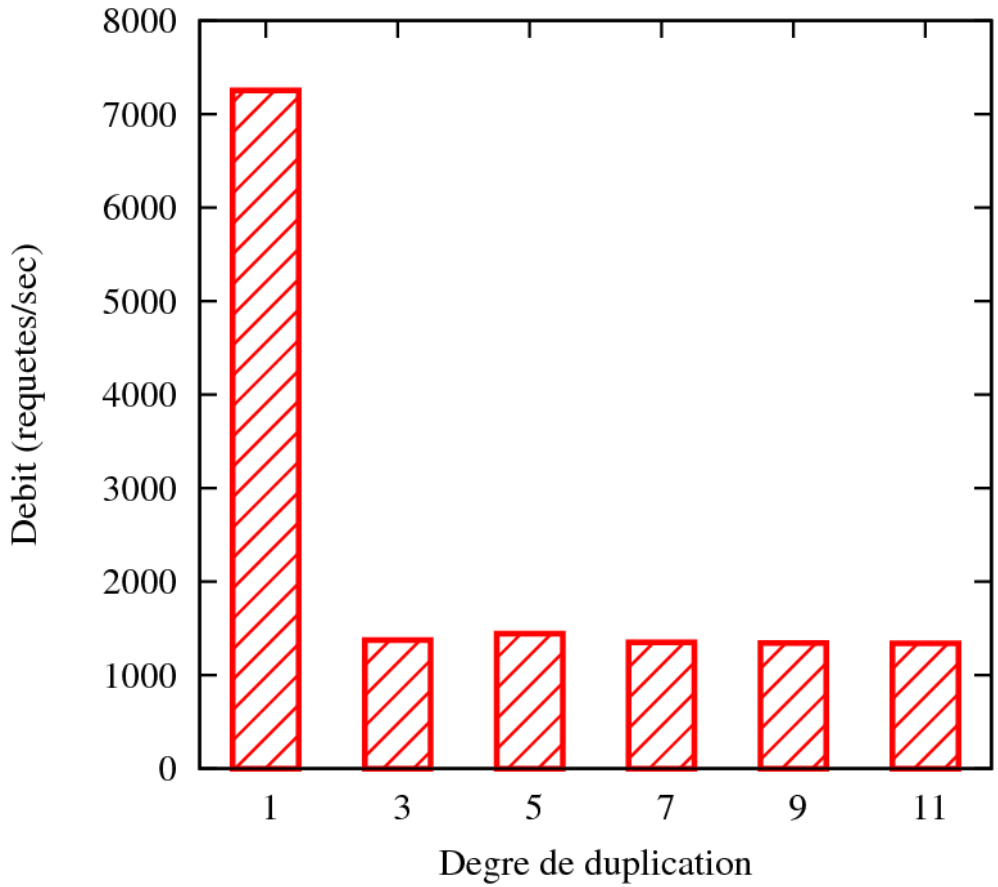
- Détecter les cas critiques pour le bon fonctionnement de Semias
- 3 règles de validité
  - Nombre de défaillances
    - Il doit toujours y avoir une majorité de duplicatas non défaillants
  - Position des duplicatas
    - Il doit toujours y avoir un duplicata de chaque coté de la clé du service
    - Un nœud hébergeant un duplicata d'un service doit avoir dans sa liste de voisins tous les nœuds hébergeant des duplicatas de ce service

Mise en œuvre

- Prototype
  - Réseau logique : Pastry
  - Algorithme de consensus : Paxos
  - Algorithme de diffusion atomique : [Schiper2006]
- Appliqué au service de gestion d'applications de Vigne
  - Peu de modifications du service
    - Enregistrement / chargement de l'état d'un service
- Évaluation sur Grid'5000
  - Performances en fonctionnement normal
  - Auto-réparation dans un environnement dynamique

Performances des services dupliqués





- Sites utilisés : Grenoble (leader), Lille (client), Lyon, Sophia, Bordeaux, Orsay, Nancy

Expérience à grande échelle

- Description
  - 100 nœuds répartis sur 7 sites
  - 70 gestionnaires d'application avec un degré de duplication de 5
  - 1 défaillance ou arrivée d'un nouveau nœud toutes les 6 minutes pendant 6 heures
  - Reconfigurations périodiques toutes les 300 secondes
- Résultats
  - Tous les gestionnaires d'application sont disponibles à la fin de l'exécution
  - Diminution de 26% du nombre de reconfigurations
    - 537 reconfigurations potentielles
    - 394 reconfigurations effectives

Travaux apparentés

- Duplication active dans un réseau logique structuré
  - PaxonDHT [Temkow2006]
    - Simulation
    - Problème de sûreté de l'algorithme de consensus
  - Systèmes de grille
    - Vigne [Rilling2006], Vishwa [Reddy2006], Zorilla [Drost2006]

- Duplication active sur Ipv6 Mobile
  - XtreemOS [Pierre2009]

- Duplication passive
  - XtremWeb : RPC-V [Djilali2004]
  - [Zhang2004]

Synthèse

- Première mise en œuvre de duplication active au-dessus d'un réseau logique structuré
  - Haute disponibilité et auto-réparation de services de grille
  - Reconfigurations transparentes pour les clients
  - Pas de modification des services existants
- Performances acceptables pour des services de grille
- Propriétés d'auto-réparation
  - Limitation du nombre de reconfigurations
  - Haute disponibilité des services dans un environnement dynamique
    - Architecture du module de communication de groupe [Mena2003]
    - Découplage suspicions / défaillances
    - Limitation de l'impact des défaillances sur le temps de réponse

O2P

Recouvrement arrière passant à l'échelle pour applications à échange de messages

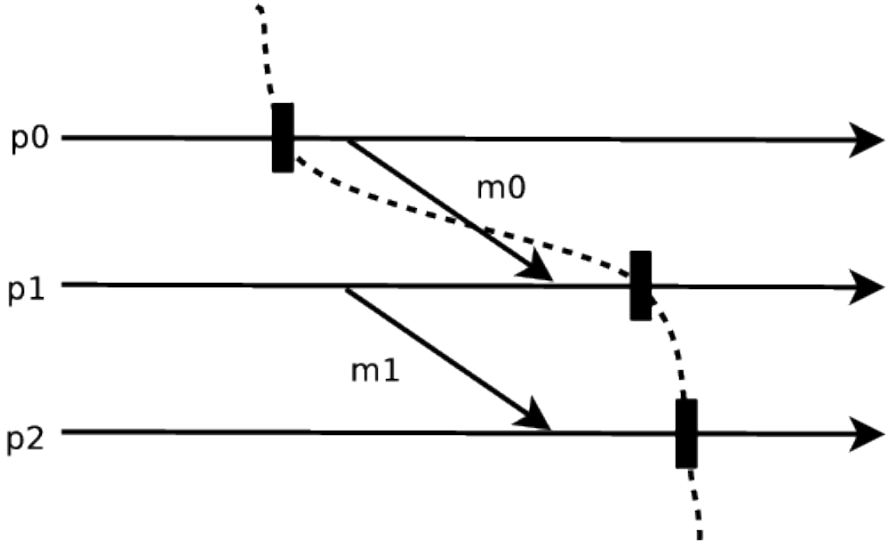
O2P

- Contexte
  - Applications à échange de messages
  - Grappe de calcul
- Objectifs
  - Passage à l'échelle
  - Performances
- Approche
  - Un protocole à enregistrement de messages optimiste actif

O2P

- Pourquoi un protocole à enregistrement de messages optimiste ?
- Enregistrement de messages optimiste actif
- Gestion distribuée de l'enregistrement des messages
- Évaluation

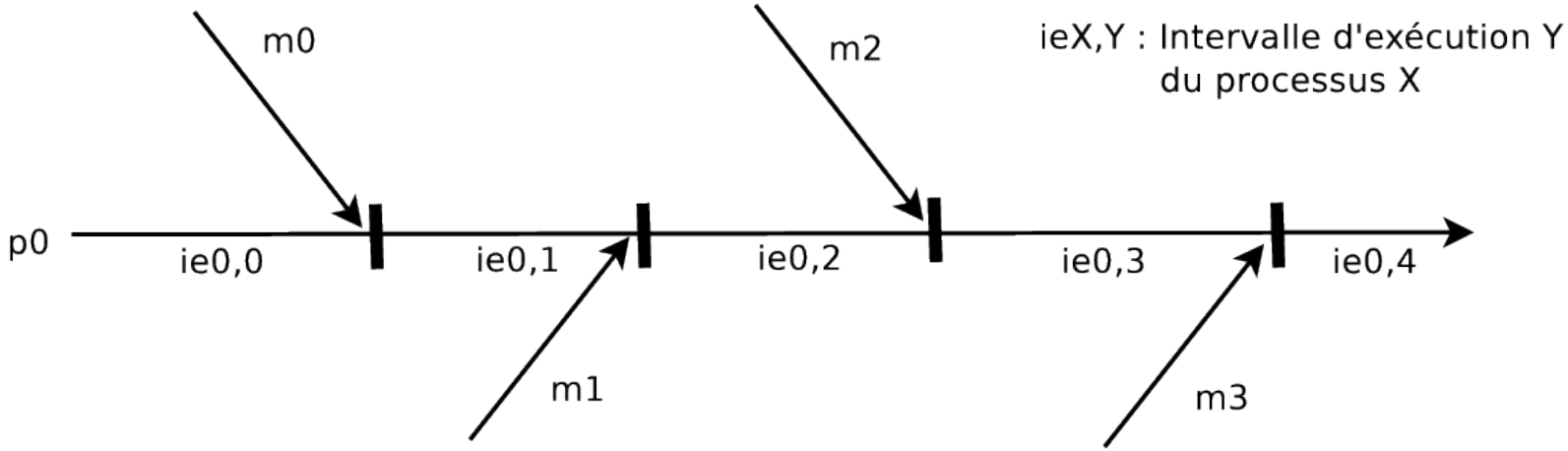
Pourquoi l'enregistrement de messages ?



- Sauvegarde de points de reprise coordonnés
  - Retour arrière de tous les processus à chaque défaillance
  - Peut empêcher l'application de progresser
- Protocoles à enregistrement de messages
  - Tolère des fréquences de défaillances plus élevées

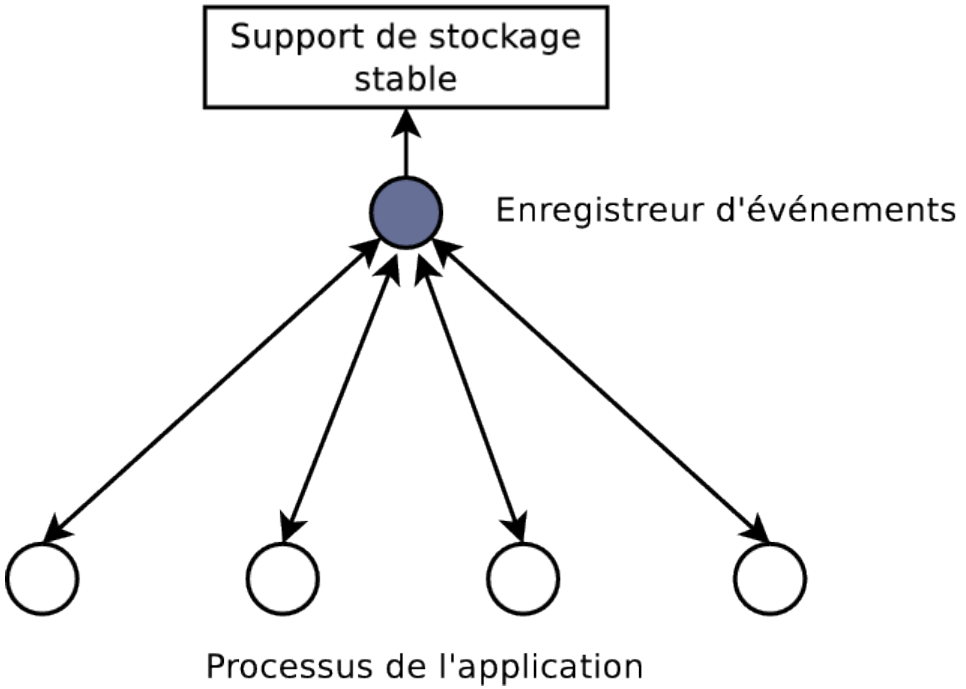
Les principes de l'enregistrement de messages

- Exécution des processus déterministe par morceaux



- Rejouer les mêmes messages dans le même ordre permet d'atteindre le même état
- Enregistrement de messages fondé sur l'émetteur
  - Contenu du message : dans la mémoire de l'émetteur
  - Déterminant : sur support stable
    - Déterminant = identifiant du message + ordre de réception

Enregistreur d'événements



- Interface entre les processus de l'application et le support de stockage stable
- Permet d'exécuter du code spécifique au protocole
- Introduit dans MPICH-V

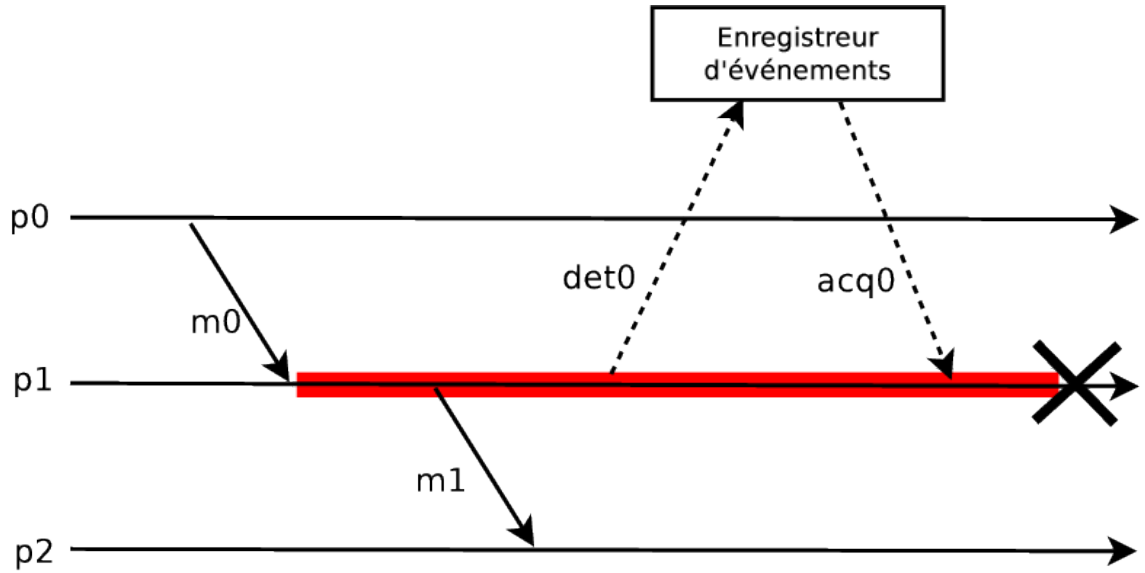
Pourquoi l'enregistrement de messages optimiste ?

- 3 familles de protocoles
  - Pessimiste
    - Sauvegarde synchrone des déterminants
  - Optimiste
    - Sauvegarde asynchrone des déterminants
  - Causal
    - Déterminants attachés sur les messages

- Performances équivalentes au redémarrage [Rao1998]

- Les protocoles optimistes sont les plus performants en fonctionnement normal

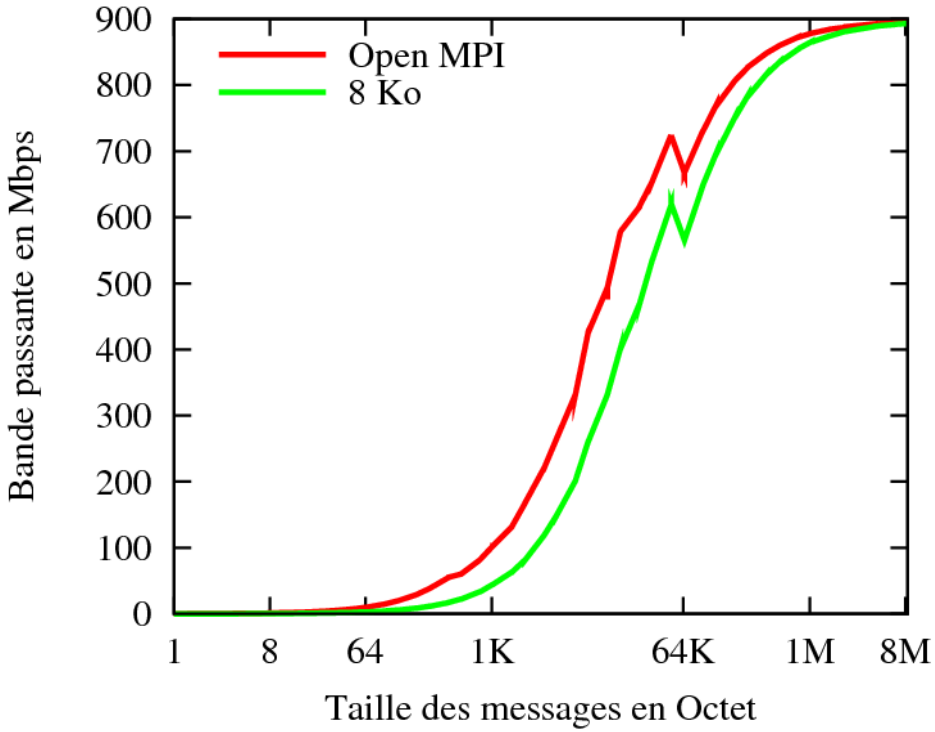
Enregistrement de messages optimiste

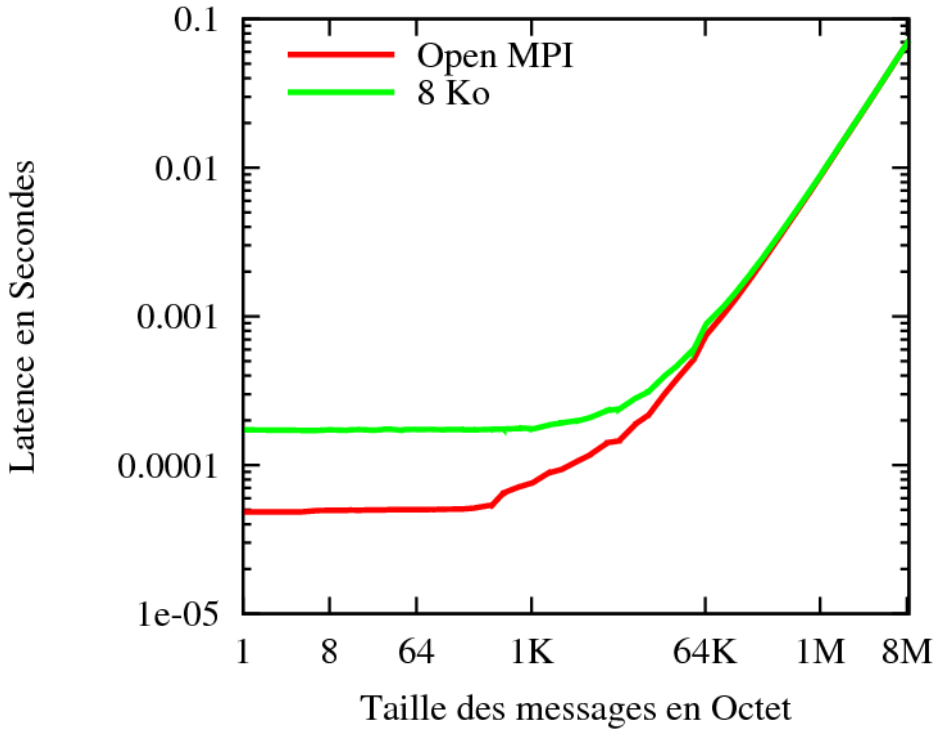


- Enregistrement asynchrone des déterminants sur support stable
- Optimisation des performances en fonctionnement normal
- Risque de création de processus orphelins

Passage à l'échelle des protocoles optimistes

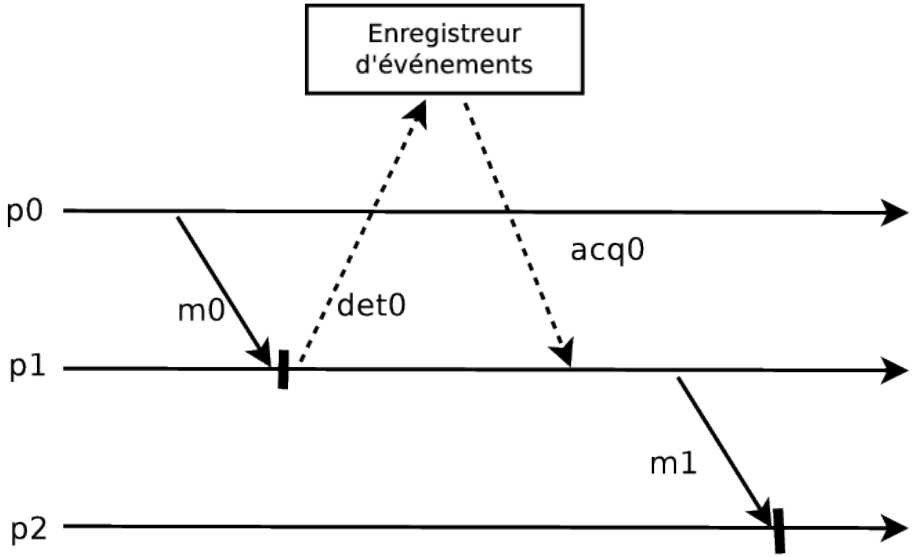
- Détection des processus orphelins
  - Les dépendances entre les processus de l'application doivent être tracées
  - Les protocoles optimistes existants
    - Utilisation de vecteurs de dépendances de taille *n* [Damani2003, Peterson1993, Sistla1989, Strom1985, Smith1996]

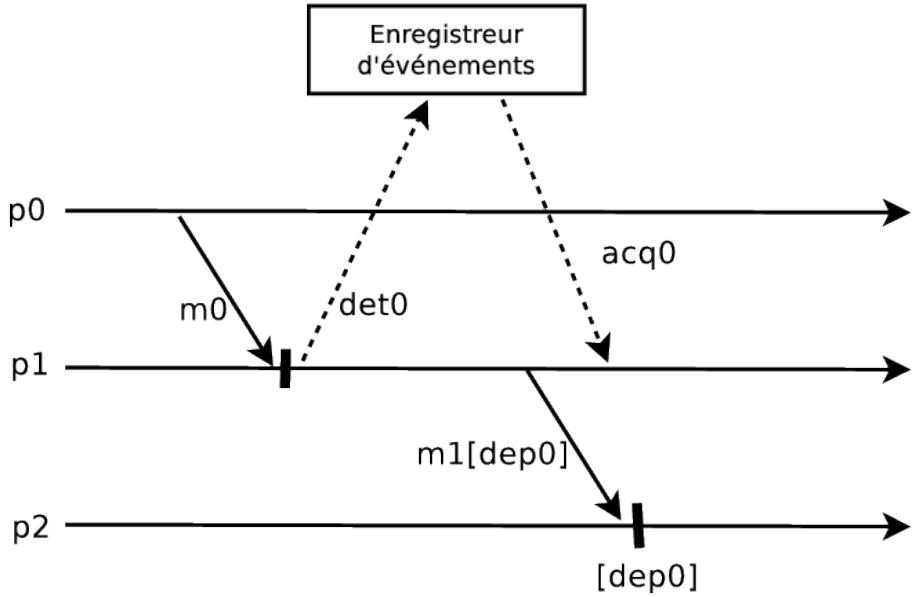




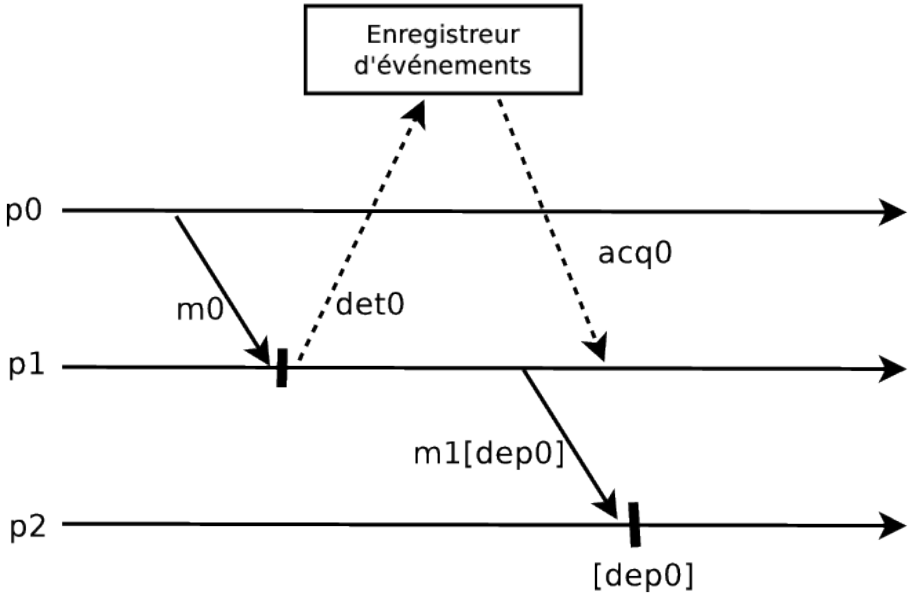
Enregistrement de messages optimiste actif

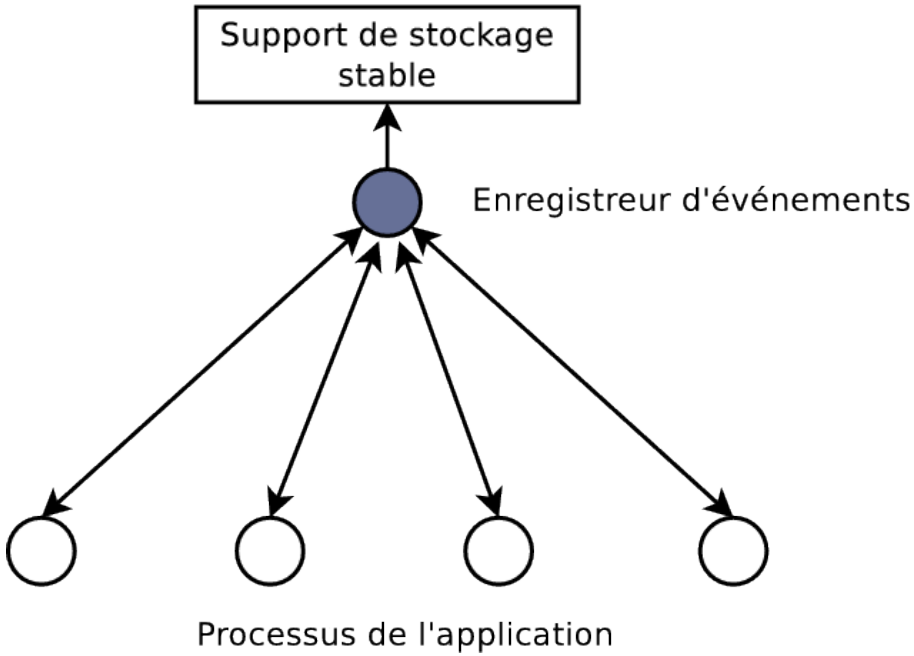
- Enregistrer les déterminants au plus tôt sur support stable





- Preuves :
  - O2P est capable de tolérer plusieurs fautes simultanées
  - O2P permet de rétablir l'application dans son état global cohérent maximum après une défaillance



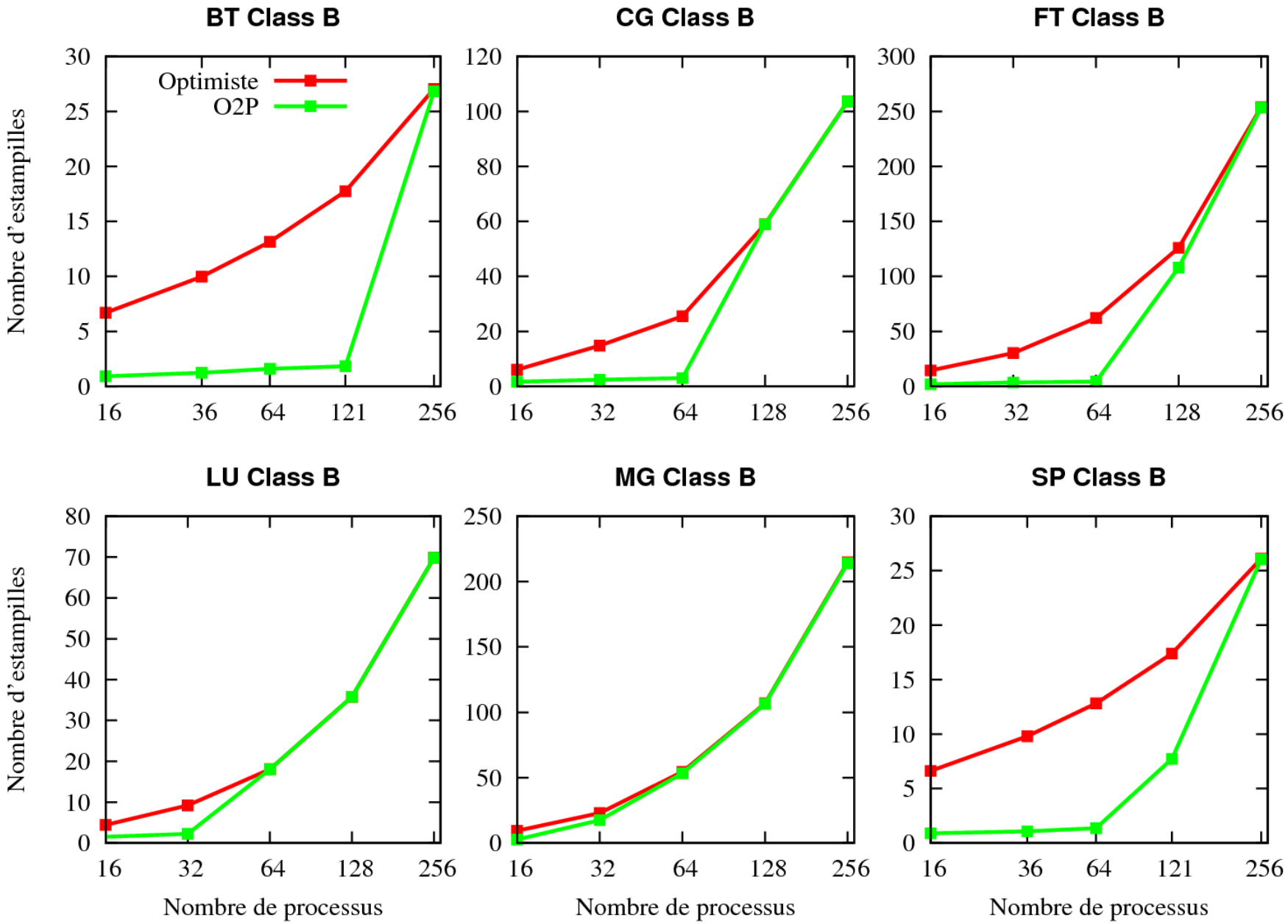


- Utilisation d'un vecteur des états stables
  - Tenu à jour par l'enregistreur d'événements
  - Envoyé comme acquittement

Évaluation

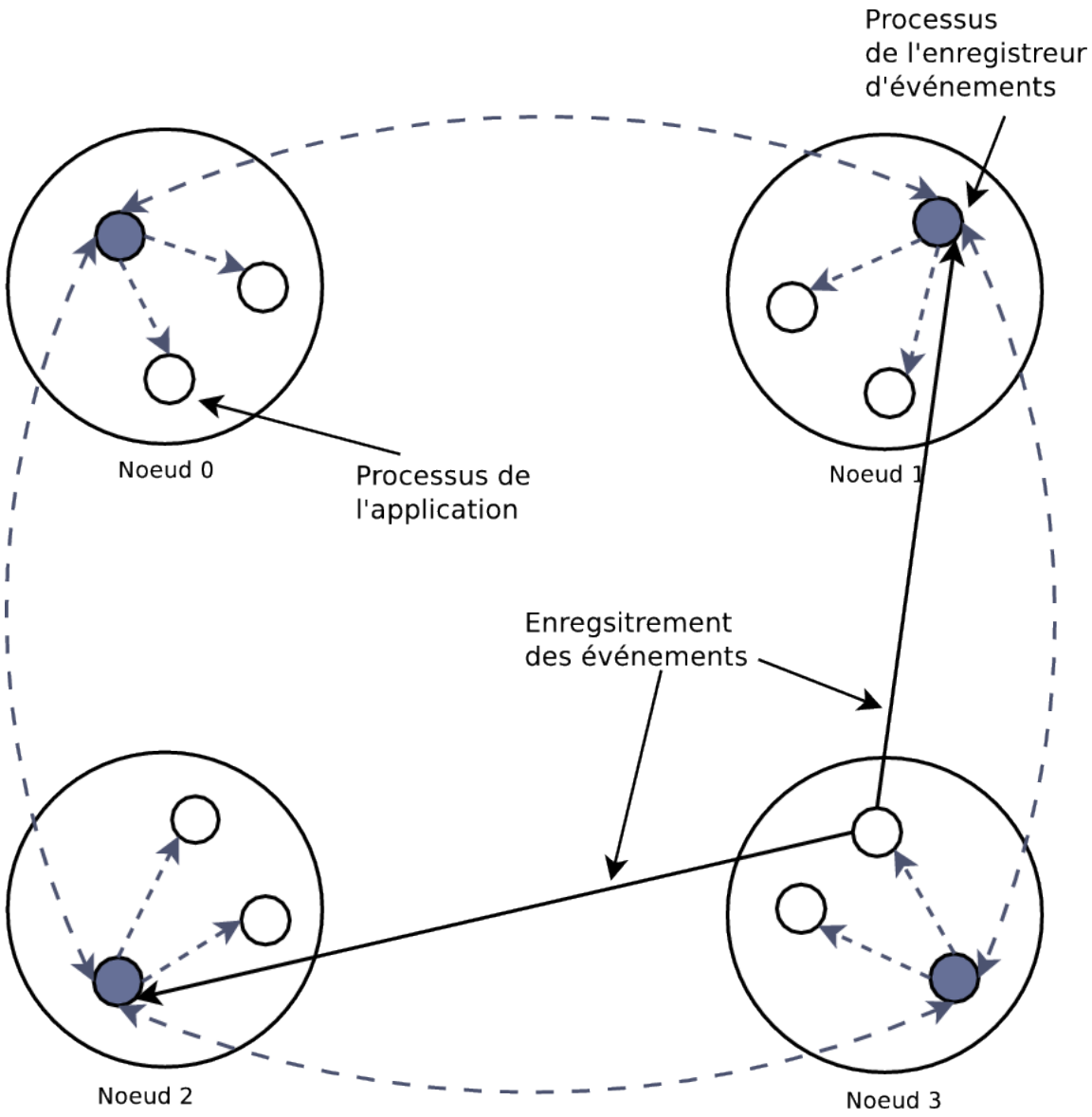
- Mise en œuvre dans Open MPI
- Évaluation sur Grid'5000 (site de Rennes)
- Applications
  - NAS Parallel Benchmark Suite
- Protocoles
  - O2P
  - Optimiste « classique »
  - Pessimiste

Taille des données attachées sur les messages

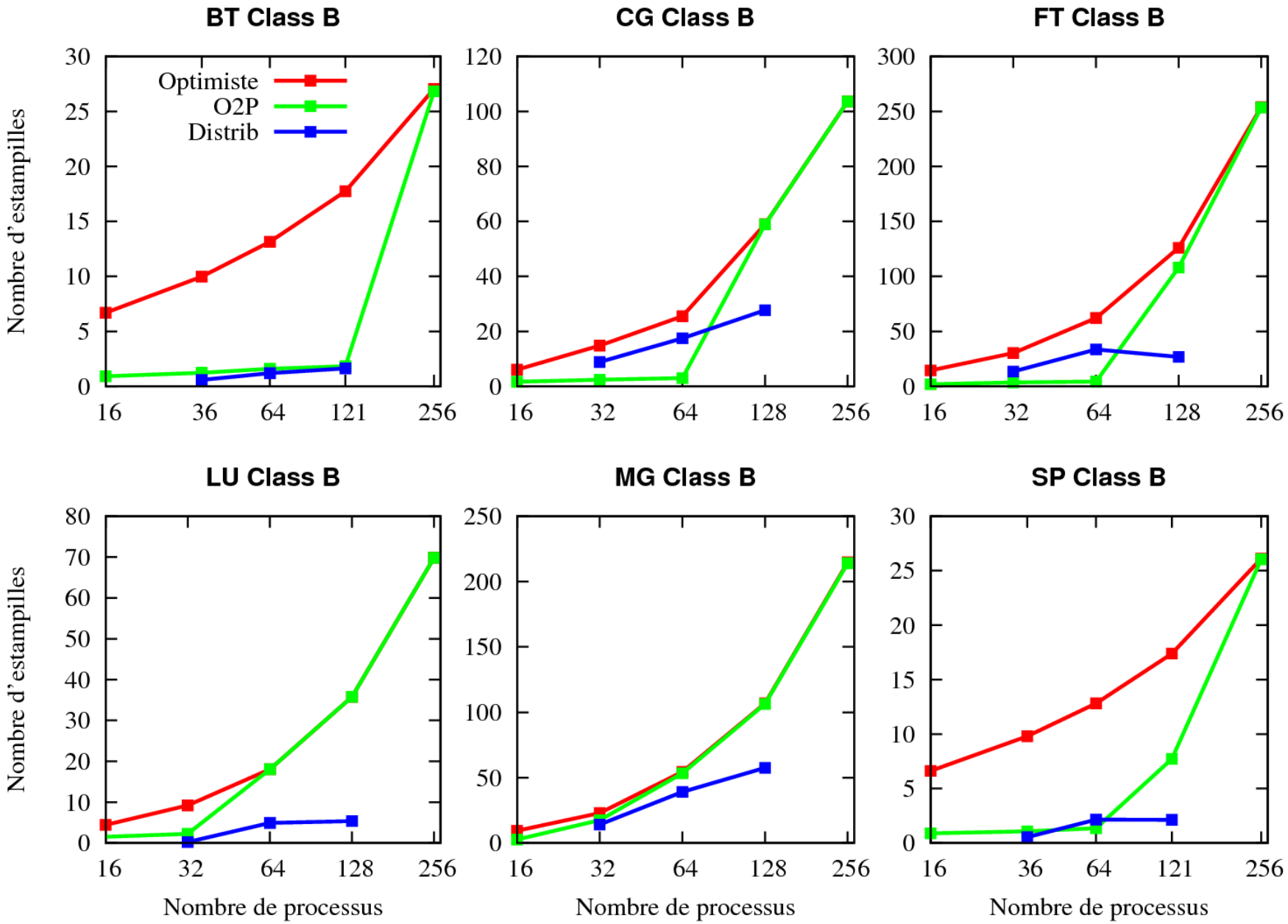


Enregistreur d'événements distribué

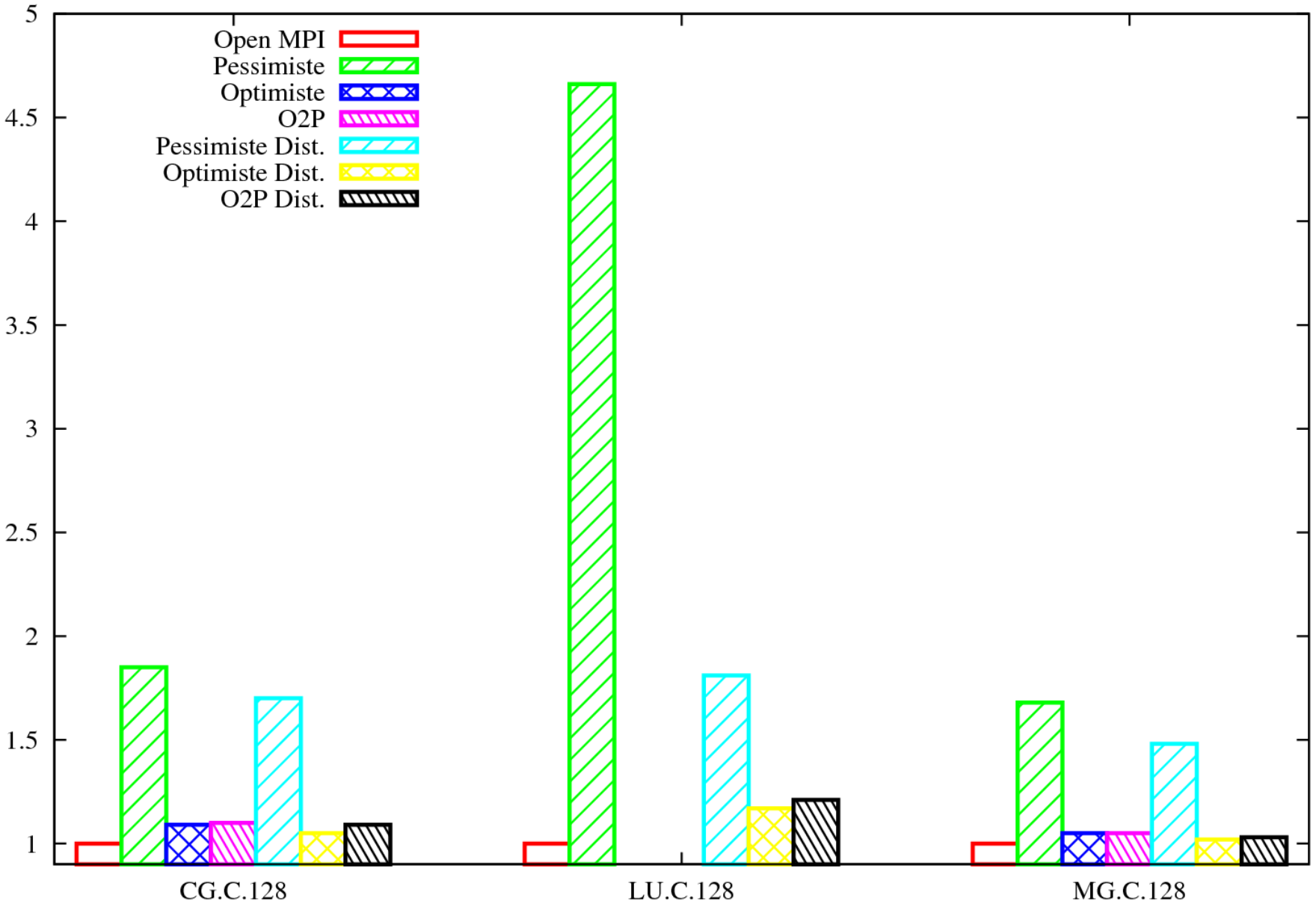
- Exploitation de la mémoire des nœuds sur lesquels s'exécute l'application
  - Paramètre *f* = nombre de fautes
- Algorithme épidémique pour diffuser les informations
  - Paramètre g = Degré de diffusion
- Chaque processus accède au vecteur des états stables local



Taille des données attachées sur les messages



Performances



Synthèse

- Gestion distribuée de l'enregistrement des messages
  - Meilleur passage à l'échelle des protocoles à enregistrement de messages

- Un protocole à enregistrement de messages optimiste actif
  - Limite la taille des informations de dépendances attachées sur les messages applicatifs
    - Meilleur passage à l'échelle
    - Performances équivalentes aux protocoles optimistes existants
  - Diminution du risque de création de processus orphelins
    - Limitation du nombre de processus devant effectuer un retour arrière
    - Validation plus rapide des messages vers le monde extérieur

Conclusion et perspectives

- XtreemGCP
  - Service de recouvrement arrière pour applications de grille
    - Redémarrage automatique des applications défaillantes
    - Adapté à l'hétérogénéité des grilles de calcul
  - Collaboration avec l'équipe de Michael Schöttner (*Heinrich-Heine University of Düsseldorf*)
  - Intégré à XtreemOS 2.0
    - Intégration d'un protocole de sauvegarde de points de reprise coordonnés
    - Intégration d'un protocole de sauvegarde de points de reprise non coordonnés

- Perspectives
  - Intégration de notre protocole O2P
  - Évaluation avec des applications réelles

Conclusion et perspectives

- Protocole de recouvrement arrière passant à l'échelle
  - O2P : un protocole à enregistrement des messages optimiste actif
    - Assure de bonnes performances en fonctionnement normal
    - Limite le risque de création de processus orphelins
  - Gestion distribuée de l'enregistrement des messages
    - Passage à l'échelle des protocoles à enregistrement de messages

- Perspectives
  - Évaluation avec des applications de grande taille
  - Évaluation au redémarrage
  - O2P-CF : un protocole hiérarchique pour les fédérations de grappes de calcul
  - Évaluation de O2P dans de nouveaux modèles d'exécution

Conclusion et perspectives

- Semias : un cadre pour la haute disponibilité de services de grille
  - Première mis en œuvre de duplication active dans un réseau logique structuré
  - Appliqué à un service de Vigne
    - Utilisation simple pour le programmeur
  - Propriétés d'auto-réparation
    - Solution adaptée aux environnements dynamiques
    - Limitation du nombre de reconfigurations

- Perspectives
  - Étendre son utilisation
    - À d'autres services de Vigne
    - À d'autres systèmes de grille (XtreemOS)
    - À d'autres contextes d'utilisation
  - Diffusion en *Open Source*

Perspectives

- Adaptation des mécanismes de tolérance aux fautes
  - Évolution de l'environnement d'exécution et des applications
  - XtreemGCP
    - Sélectionner le protocole de tolérance aux fautes le mieux adapté pour l'application
    - Changer de protocole de recouvrement arrière dynamiquement
  - O2P
    - Sélectionner les meilleurs paramètres
  - Semias
    - Adapter le degré de duplication et les règles de validité
- Exploitation des architectures multi-cœurs
  - Exécuter les mécanismes de tolérance aux fautes en parallèle des processus des applications

Services et protocoles pour l'exécution fiable d'applications distribuées dans les grilles de calcul

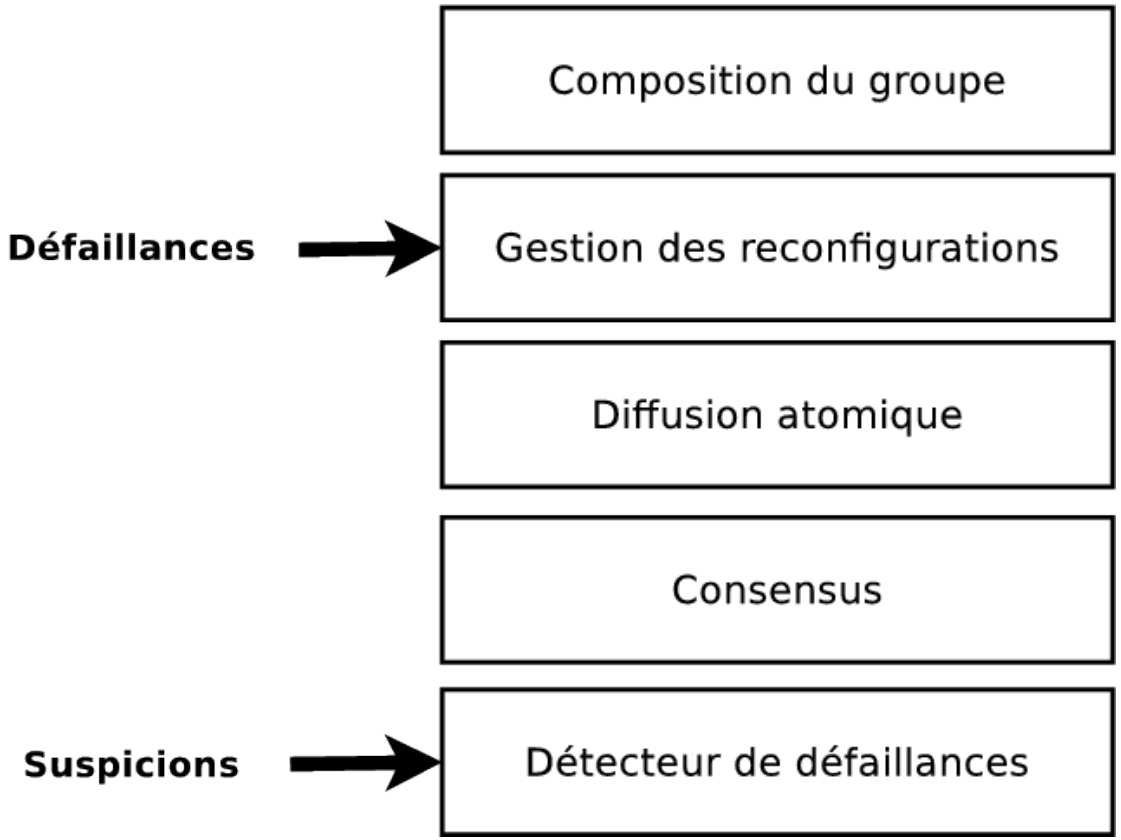
Thomas Ropars

Équipe-projet PARIS

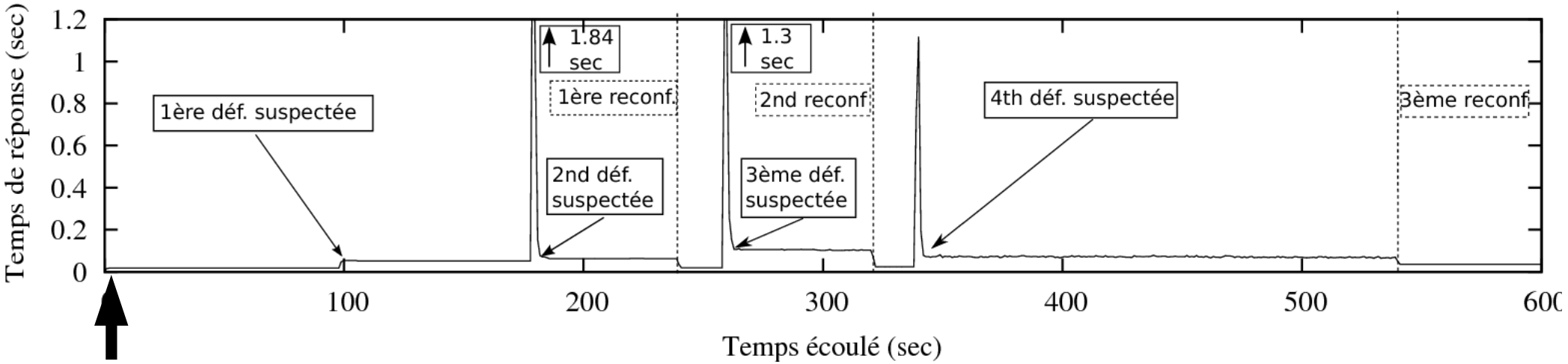
INRIA IRISA UNIVERSITÉ DE RENNES 1

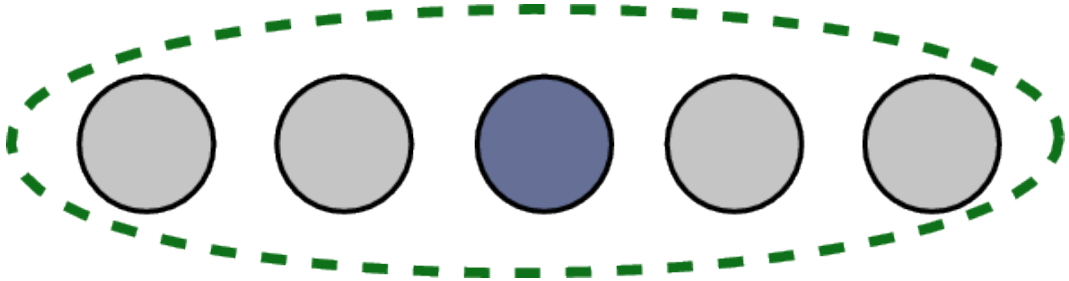
Communications de groupe

- Mena *et al.* 2003

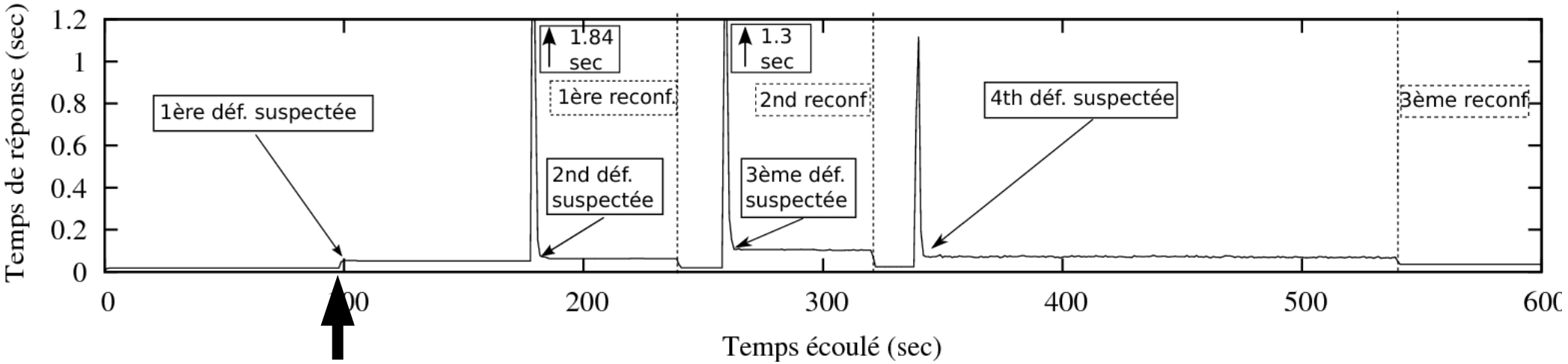


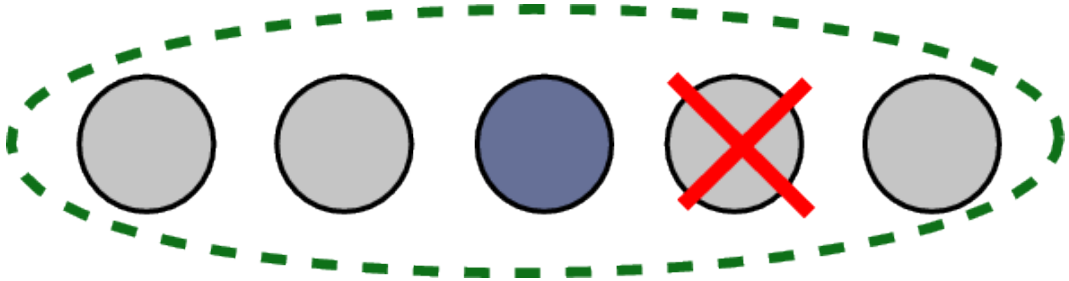
Performances dans un environnement dynamique



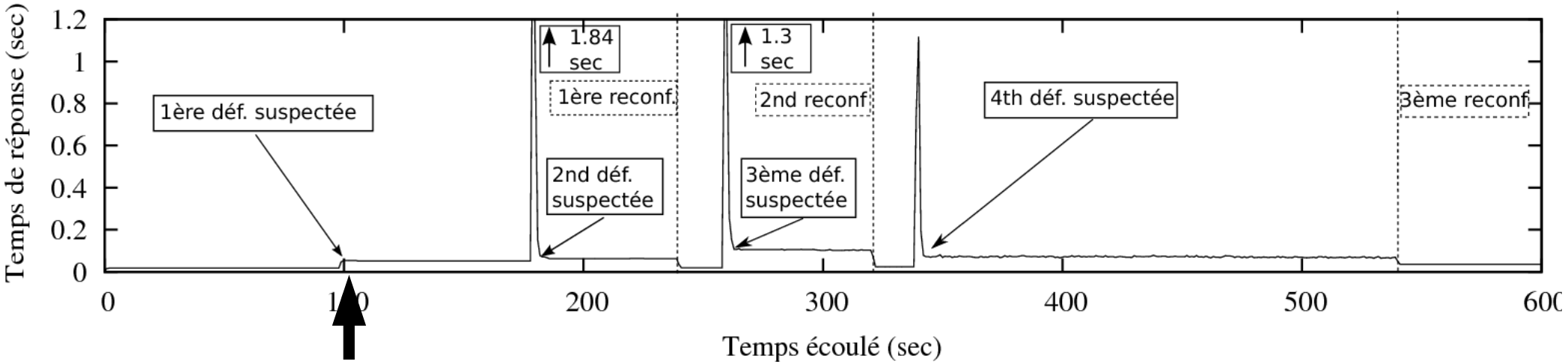


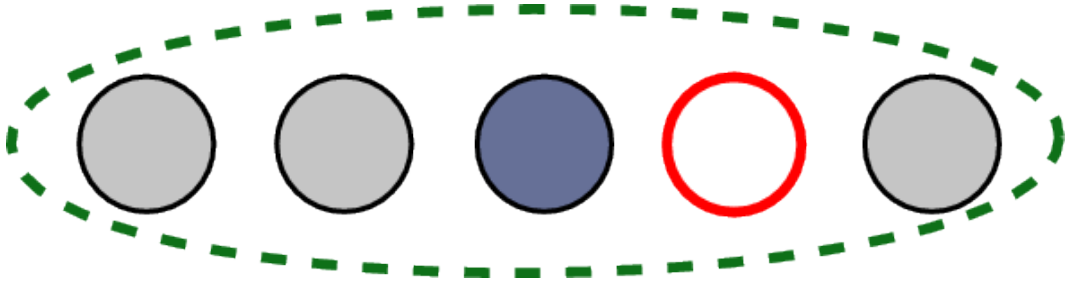
Performances dans un environnement dynamique



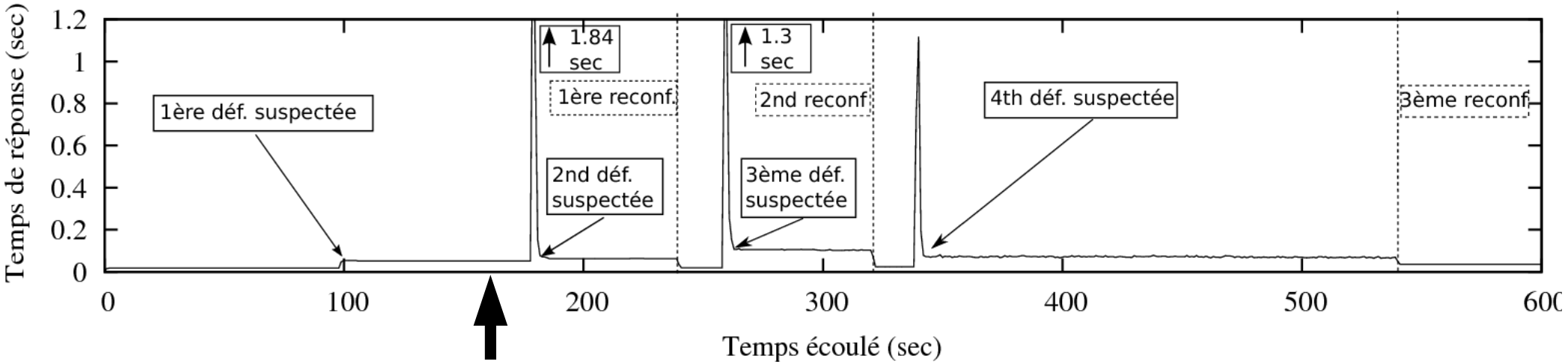


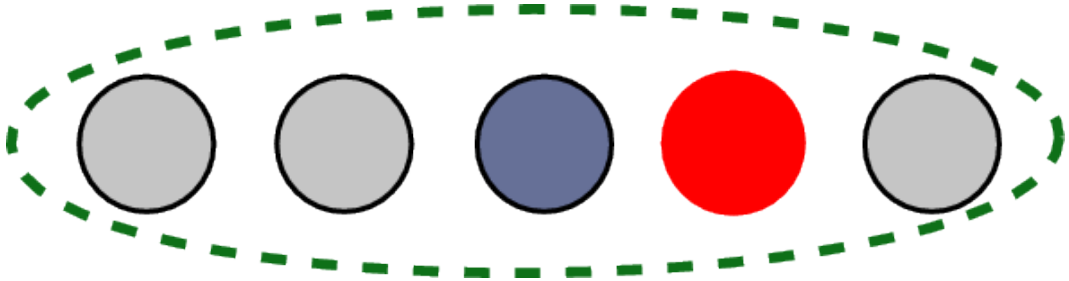
Performances dans un environnement dynamique



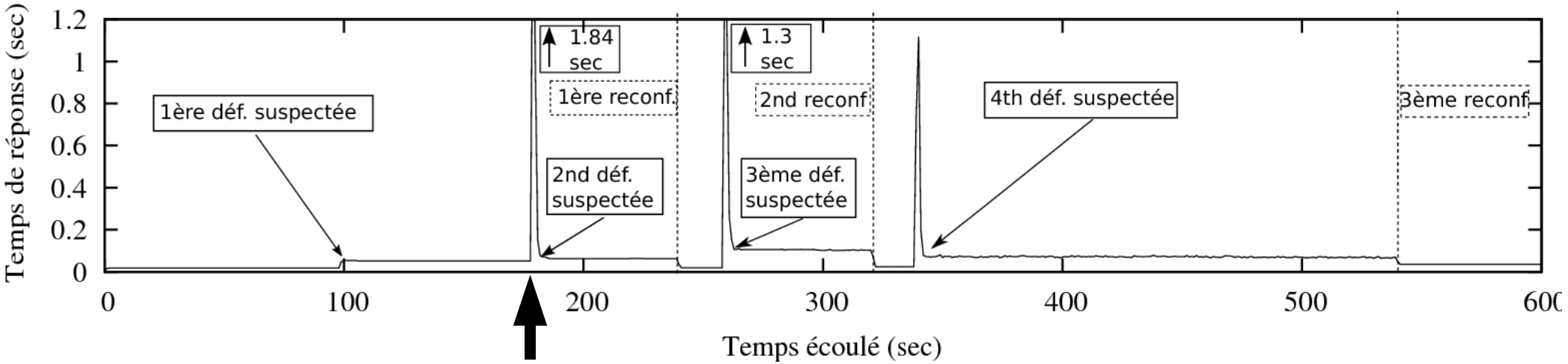


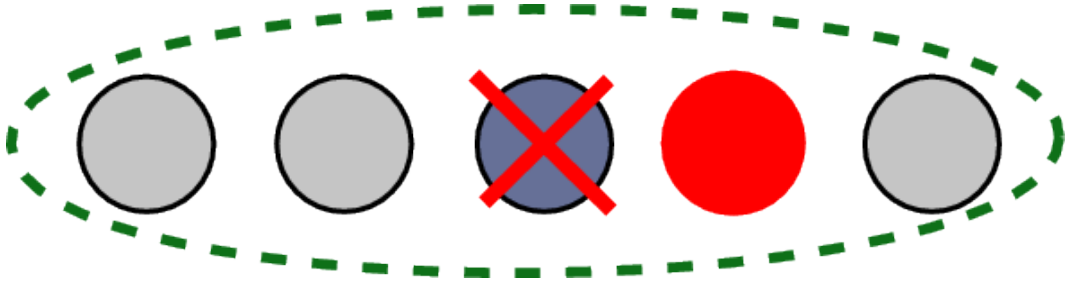
Performances dans un environnement dynamique



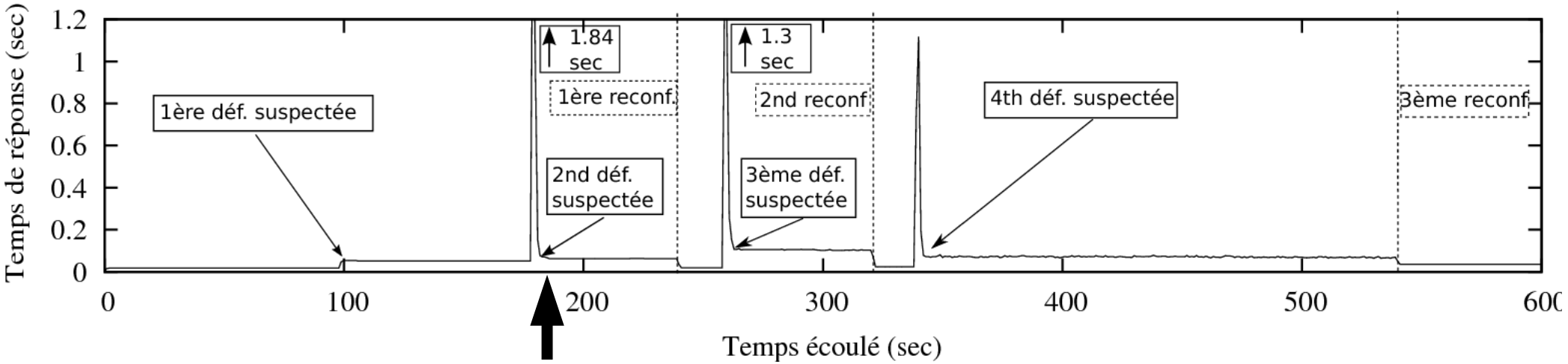


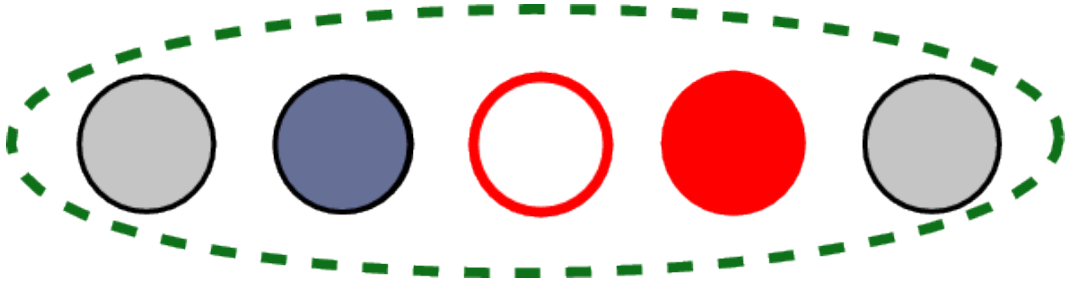
Performances dans un environnement dynamique



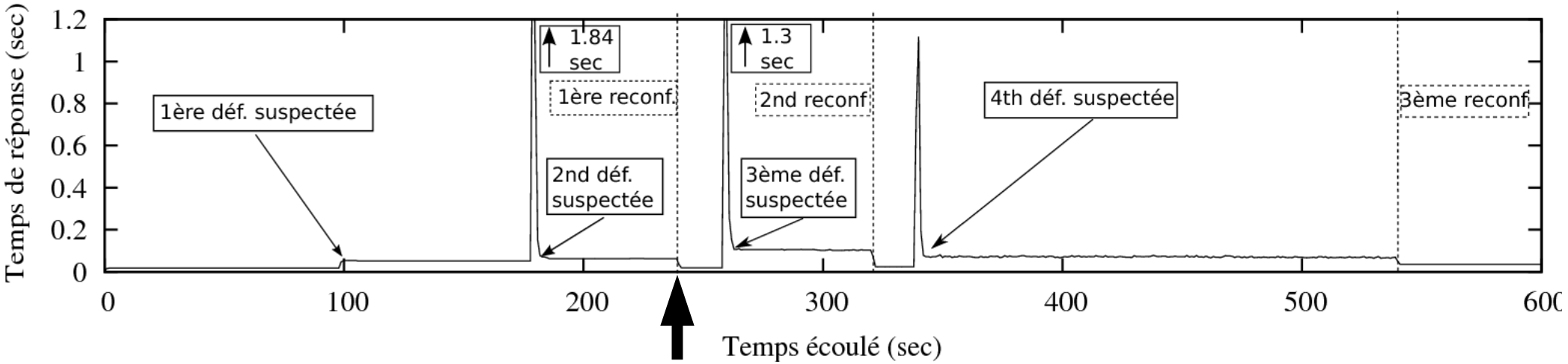


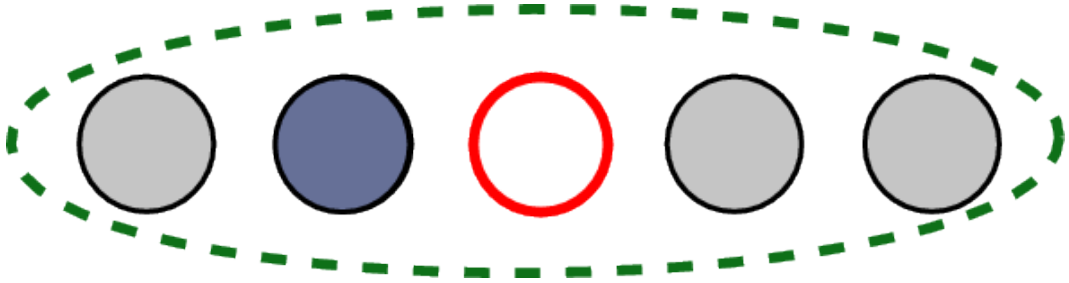
Performances dans un environnement dynamique



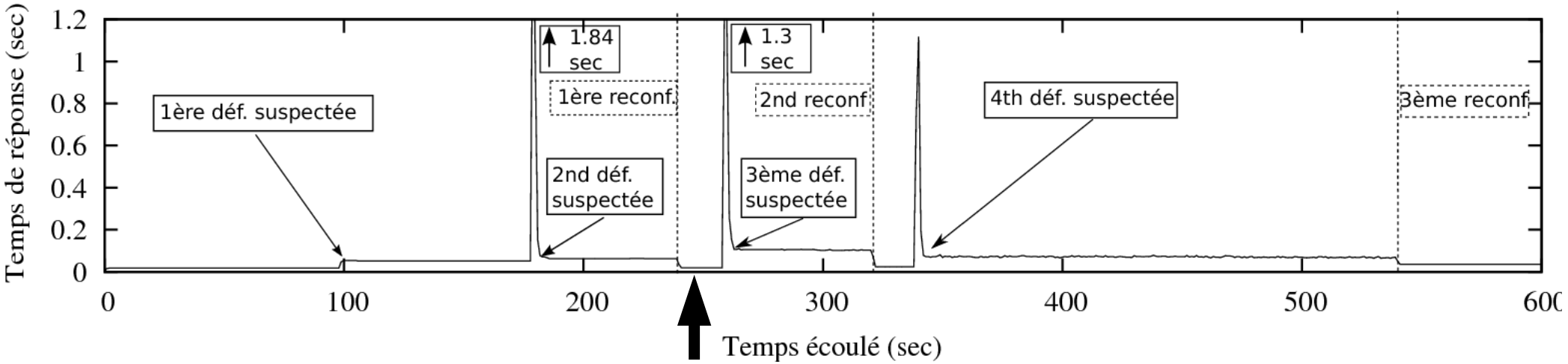


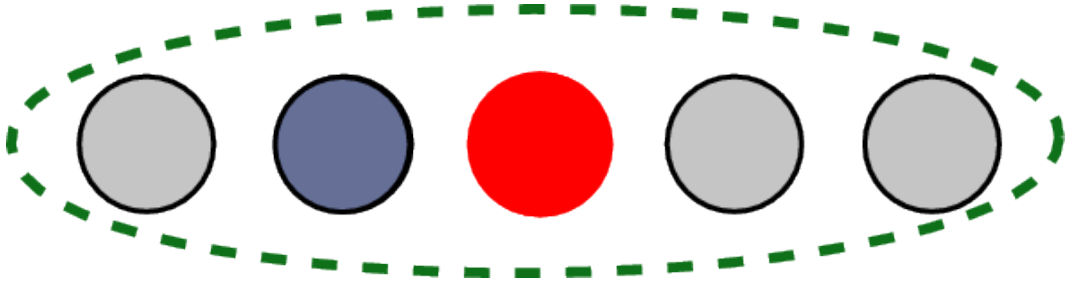
Performances dans un environnement dynamique



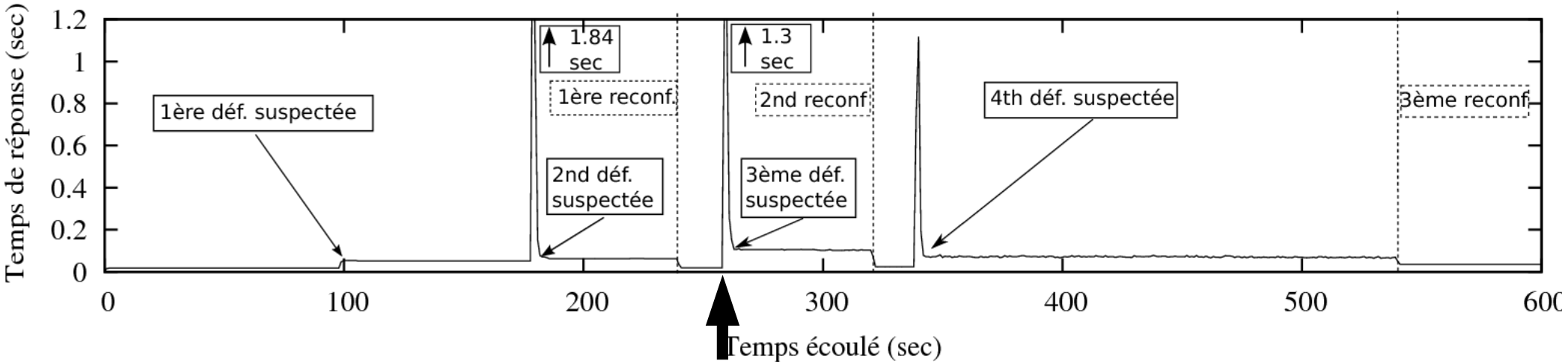


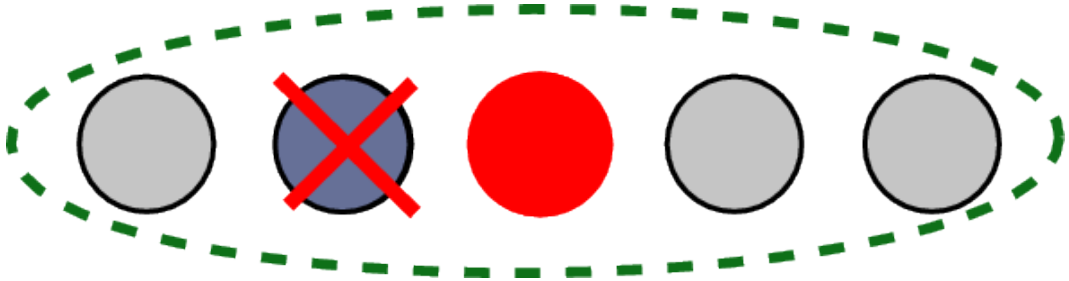
Performances dans un environnement dynamique



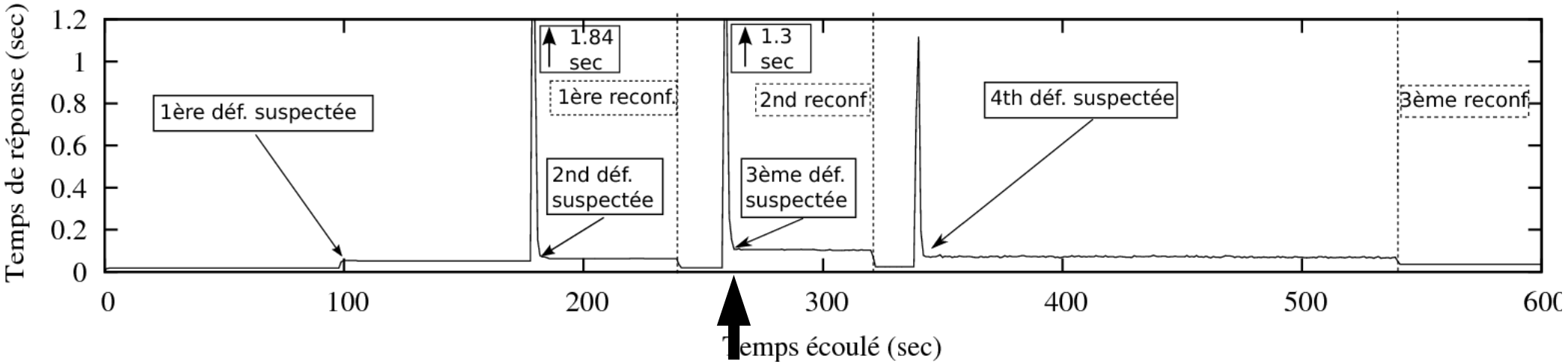


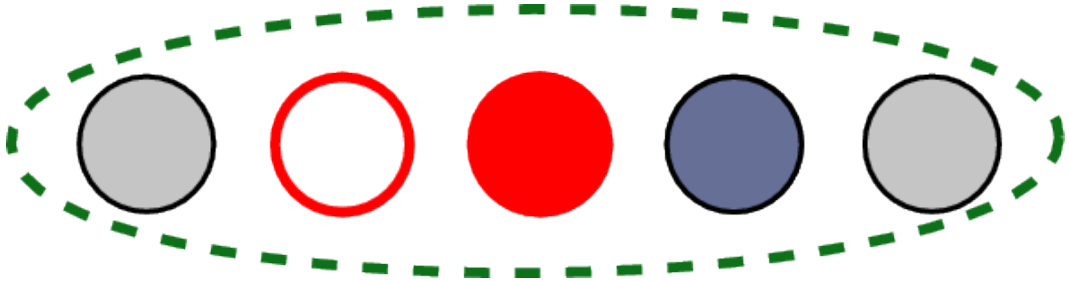
Performances dans un environnement dynamique



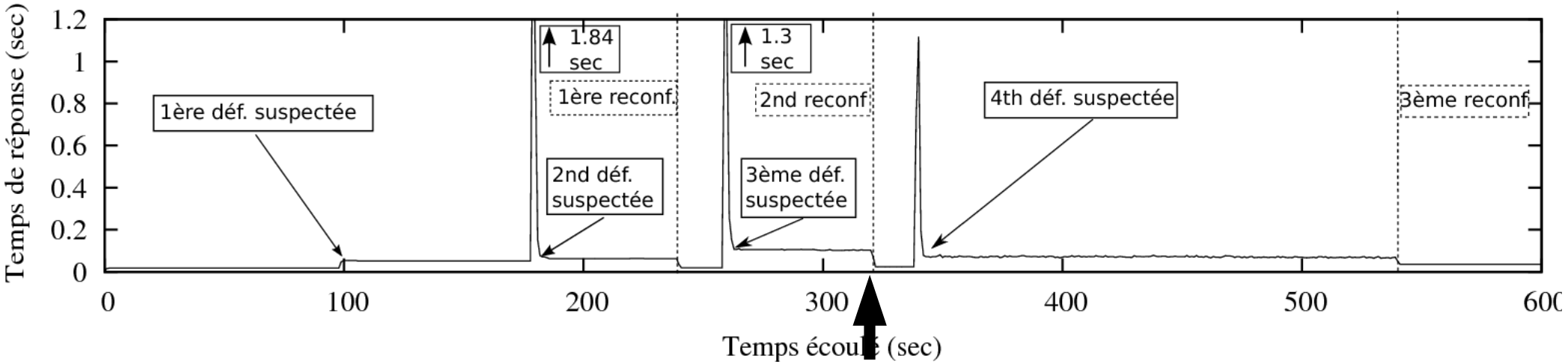


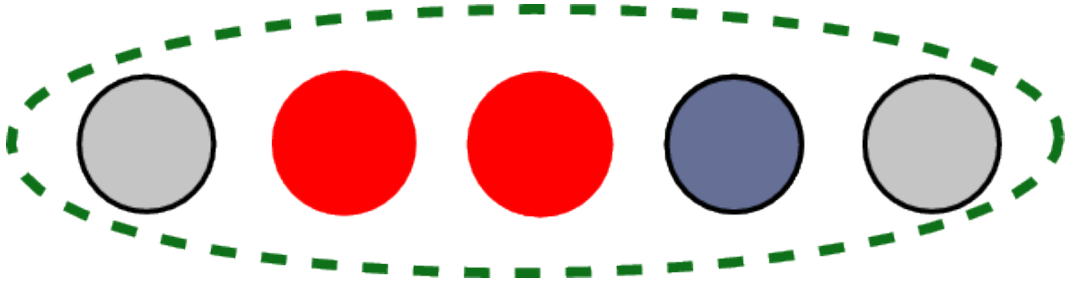
Performances dans un environnement dynamique



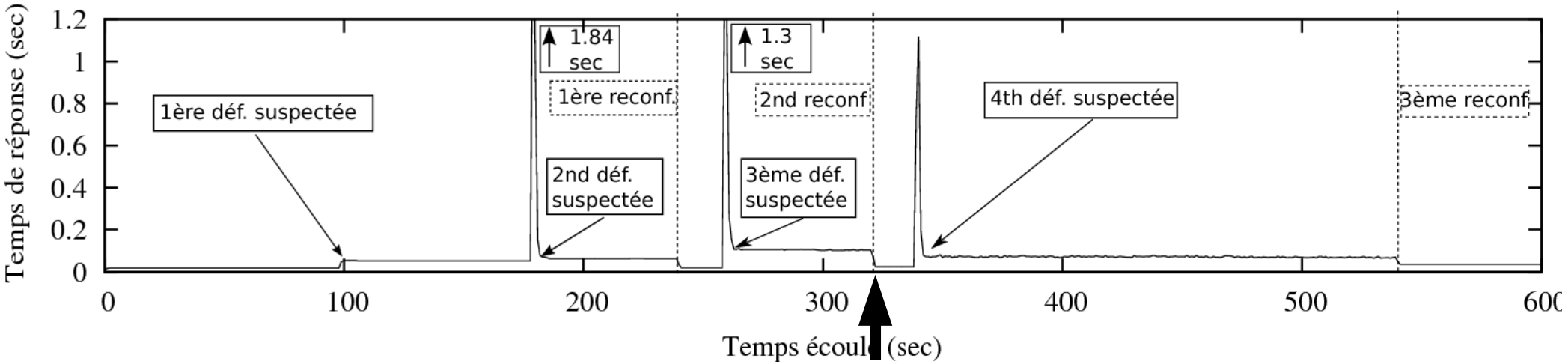


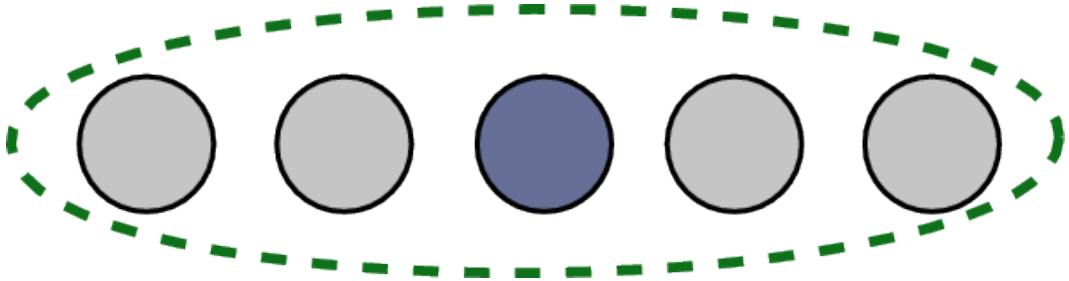
Performances dans un environnement dynamique



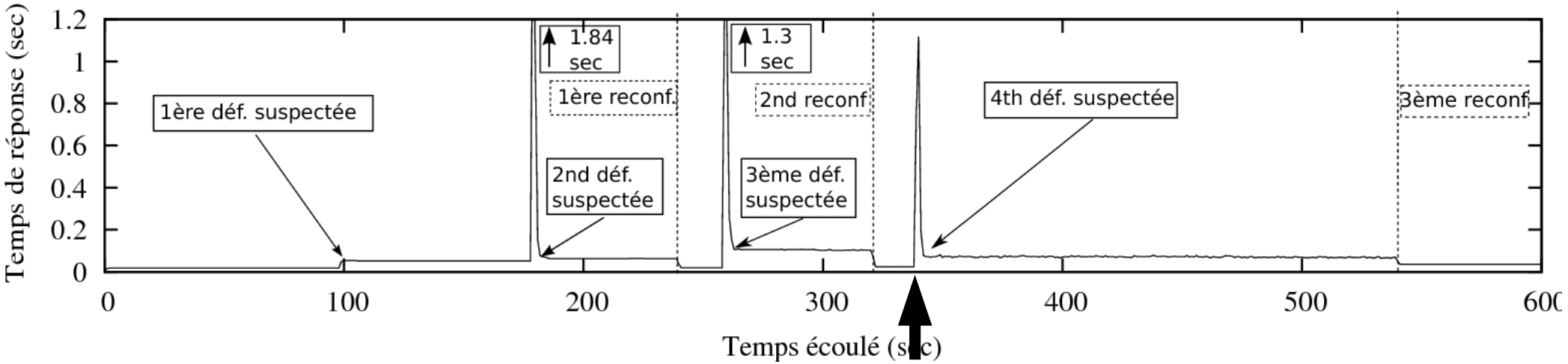


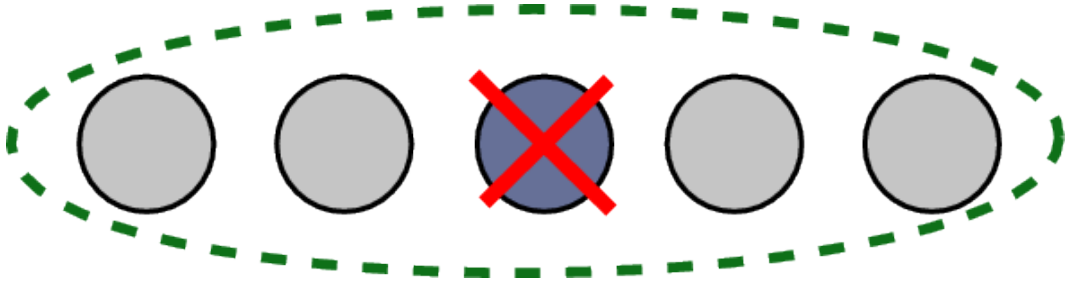
Performances dans un environnement dynamique



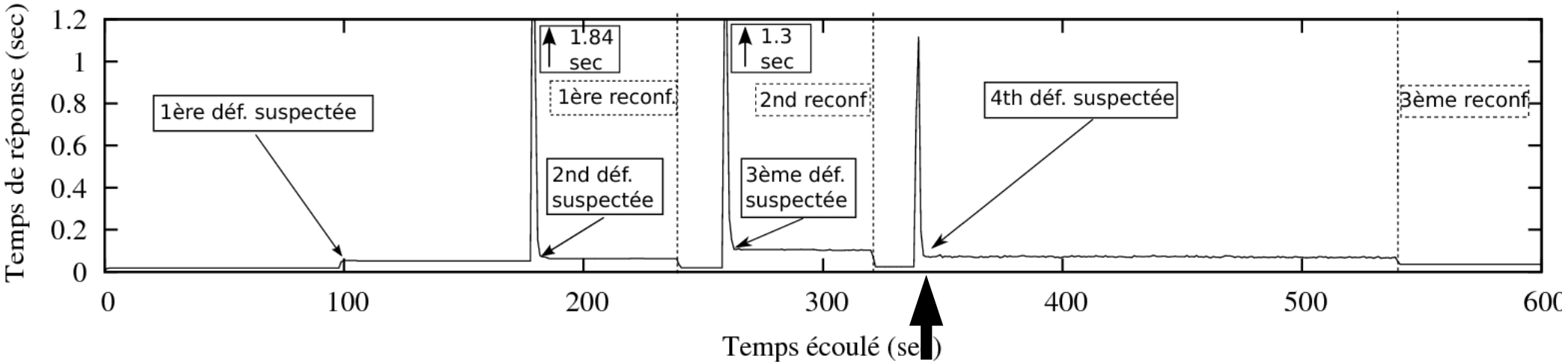


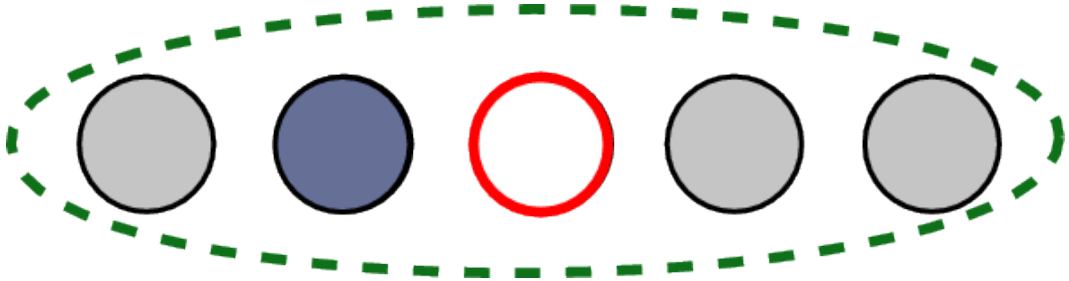
Performances dans un environnement dynamique



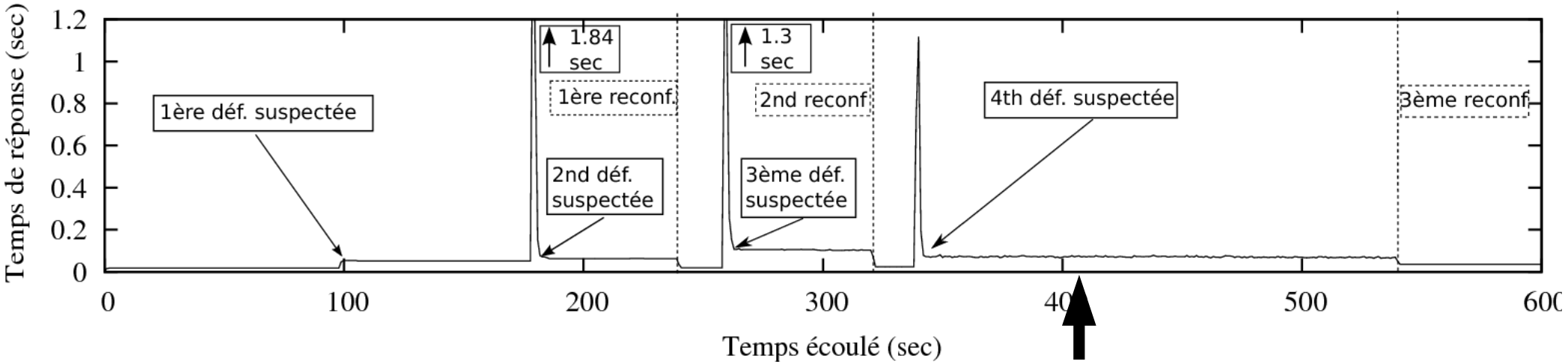


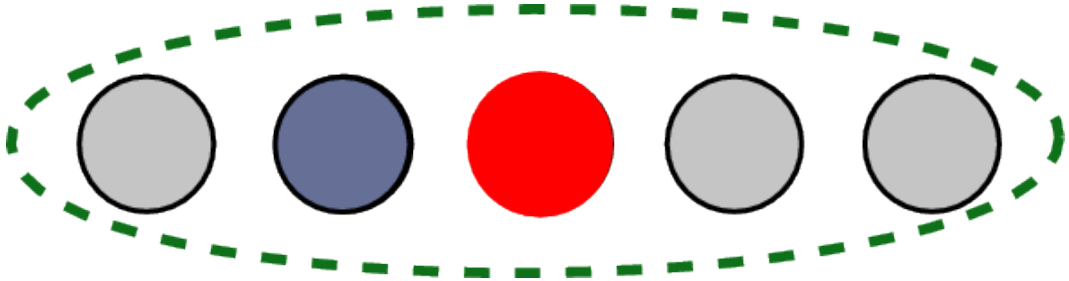
Performances dans un environnement dynamique





Performances dans un environnement dynamique





Performances dans un environnement dynamique