Conception et réalisation d'une architecture tolérant les intrusions pour des serveurs Internet

Ayda Saidane, Yves Deswarte, Vincent Nicomette

Tolérance aux fautes et Sûreté de Fonctionnement informatique
LAAS-CNRS

Projet DIT (Programme DARPA OASIS)

Internet : un réseau ouvert…

Environnement hétérogène

- Différents types d'utilisations d'Internet : Commerciales (B2C), professionnelles (B2B), administratives (B2A, C2A, e-Gov), sociales, culturelles, …
  - ✓ Toutes ces utilisations sont légitimes
  - ✓ Tout le monde n' a pas les mêmes besoins de sécurité

    => Beaucoup de systèmes mal administrés

Attaques de plus en plus efficaces

- Des attaques de grande envergure en un minimum de temps
  - ✓ le ver Slammer (2003) → 6 mois après l'annonce de la vulnérabilité;
  - ✓ le ver Witty (2004) → 24 heures après l'annonce de la vulnérabilité

Connexion des systèmes critiques à Internet

Approches classiques insuffisantes

- ✓ Renforcer l'authentification et l'autorisation ➔ inapplicable pour des serveurs accessibles au public
- ✓ Inefficaces contre les vulnérabilités ➔ correction ("patches")…

Les systèmes sont trop complexes

- ✓ Fonctionnalités standard
- ✓ Utilisation des COTS
- ✓ Rentabilité/sécurité

Objectif :

↪ Des systèmes capables de fournir le service pendant les attaques

Systèmes cibles

Serveurs publiant des informations ± critiques

- Propriétés critiques : intégrité, disponibilité
- Les données ne sont pas très confidentielles

Par exemple:

- Serveur publiant des alertes de sécurité
  - ✓ Fortes exigences en intégrité et disponibilité
  - ✓ Informations rarement mises à jour
  - ↪ Les mises à jours peuvent être faites hors-ligne

- Serveur d'une agence de voyage sur Internet
  - ✓ Fortes exigences en intégrité et disponibilité
  - ✓ Système très dynamique
  - ↪ Les mises à jour doivent être faites en temps réel

Objectifs

- Concevoir un serveur Internet tolérant aux intrusions à base de COTS
- Fournir un bon compromis performance/sécurité
- Assurer, en permanence, l'intégrité et la disponibilité des données

Caractéristiques des attaques

- Contrairement aux fautes accidentelles
  - ✓ Les attaques ne sont pas indépendantes
  - ✓ Leur distribution n'est pas uniforme dans le temps

- Les attaques visent des vulnérabilités spécifiques à un système exploitation, un logiciel d'application ou une plateforme matérielle

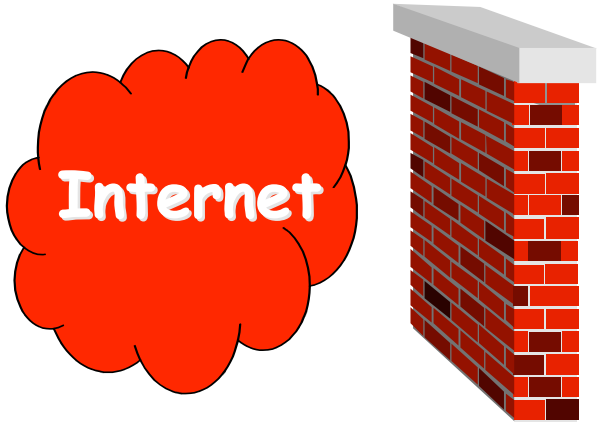
Un serveur tolérant les intrusions

Comportement en cas d'attaques

Mise en œuvre et mesures de performances

Bilan et perspectives

Un serveur tolérant les intrusions



1. Diversification
   . Matériel
   . OS
   . Logiciel

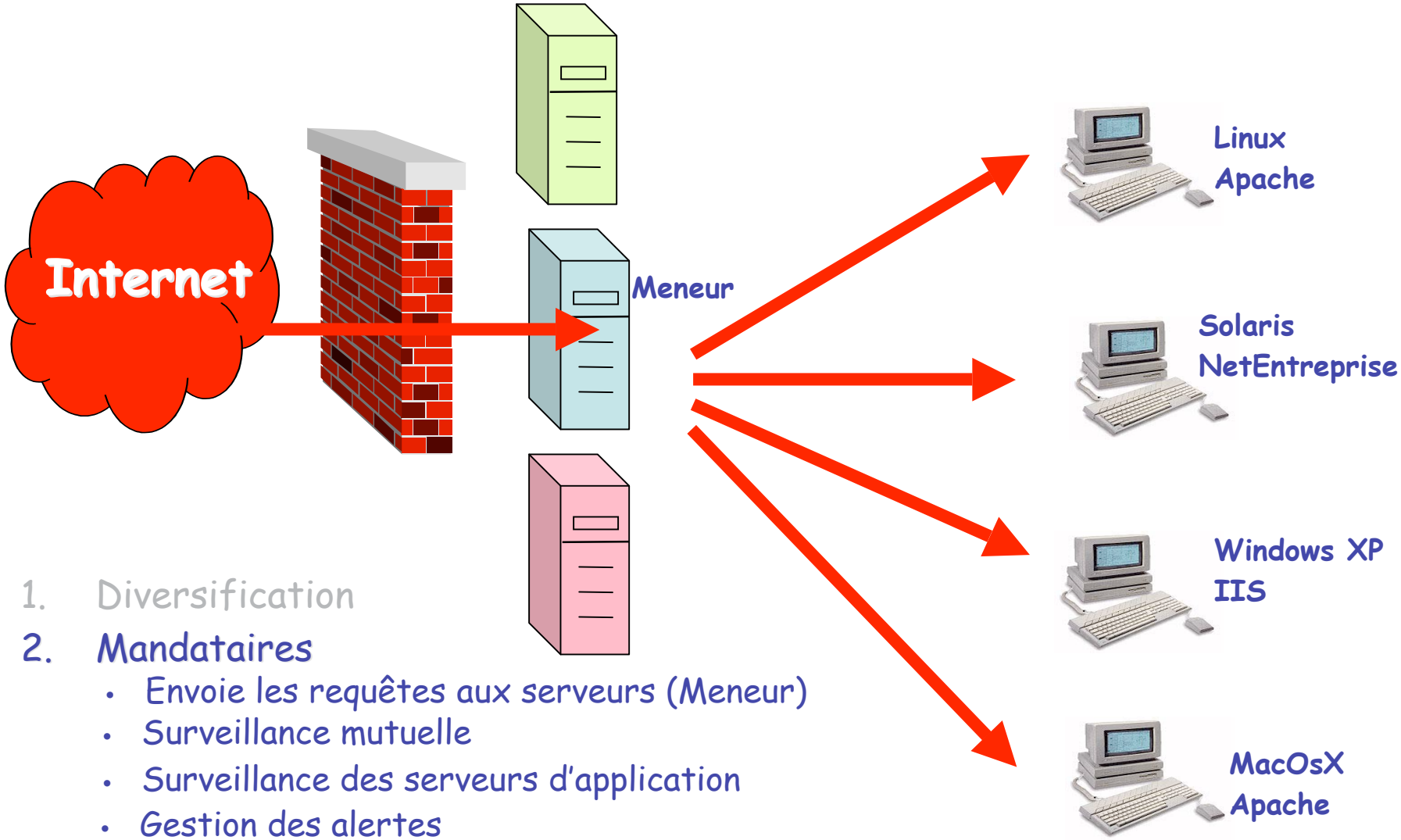

Linux
Apache


Solaris
NetEntreprise


Windows XP
IIS

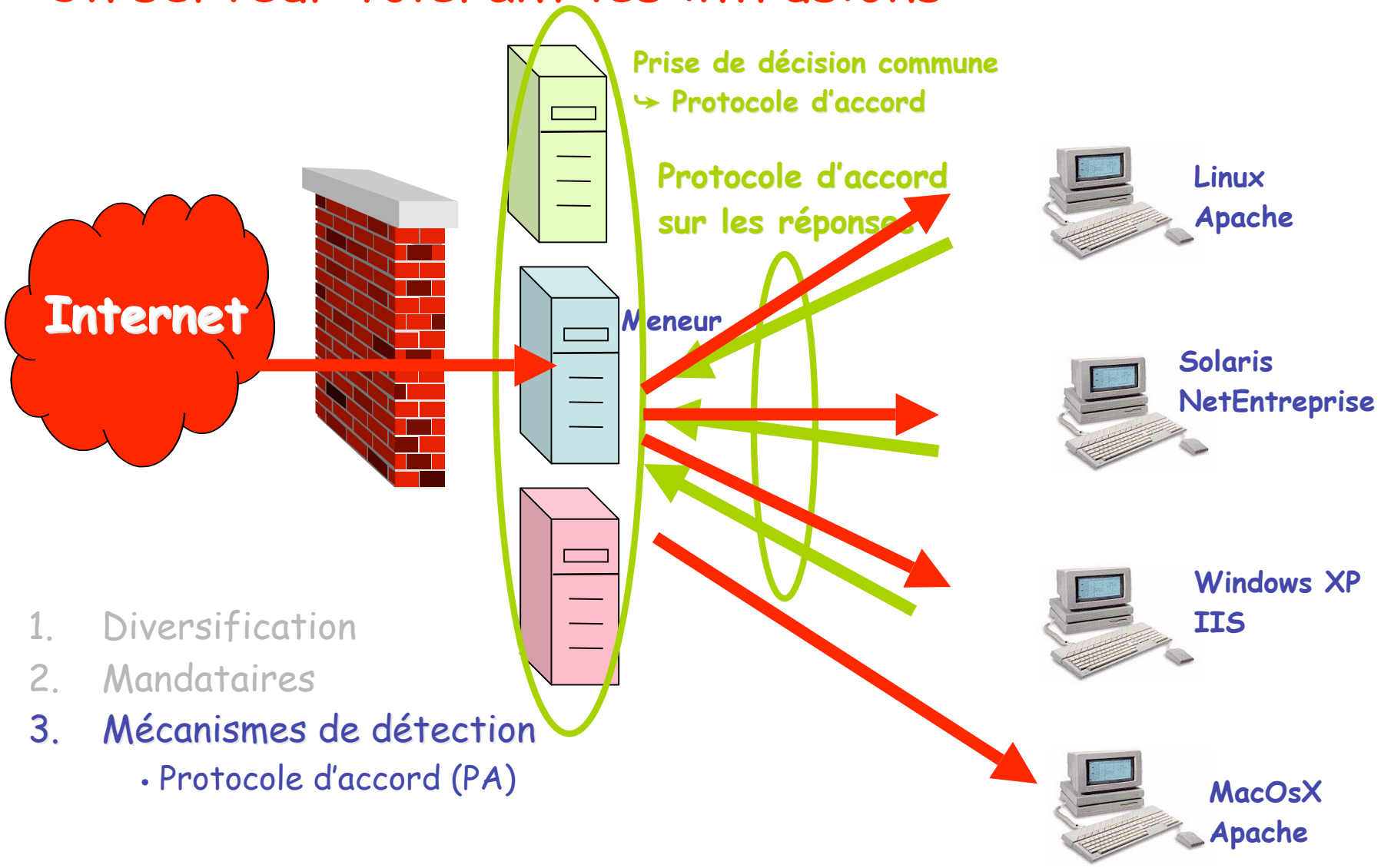

MacOsX
Apache

Un serveur tolérant les intrusions



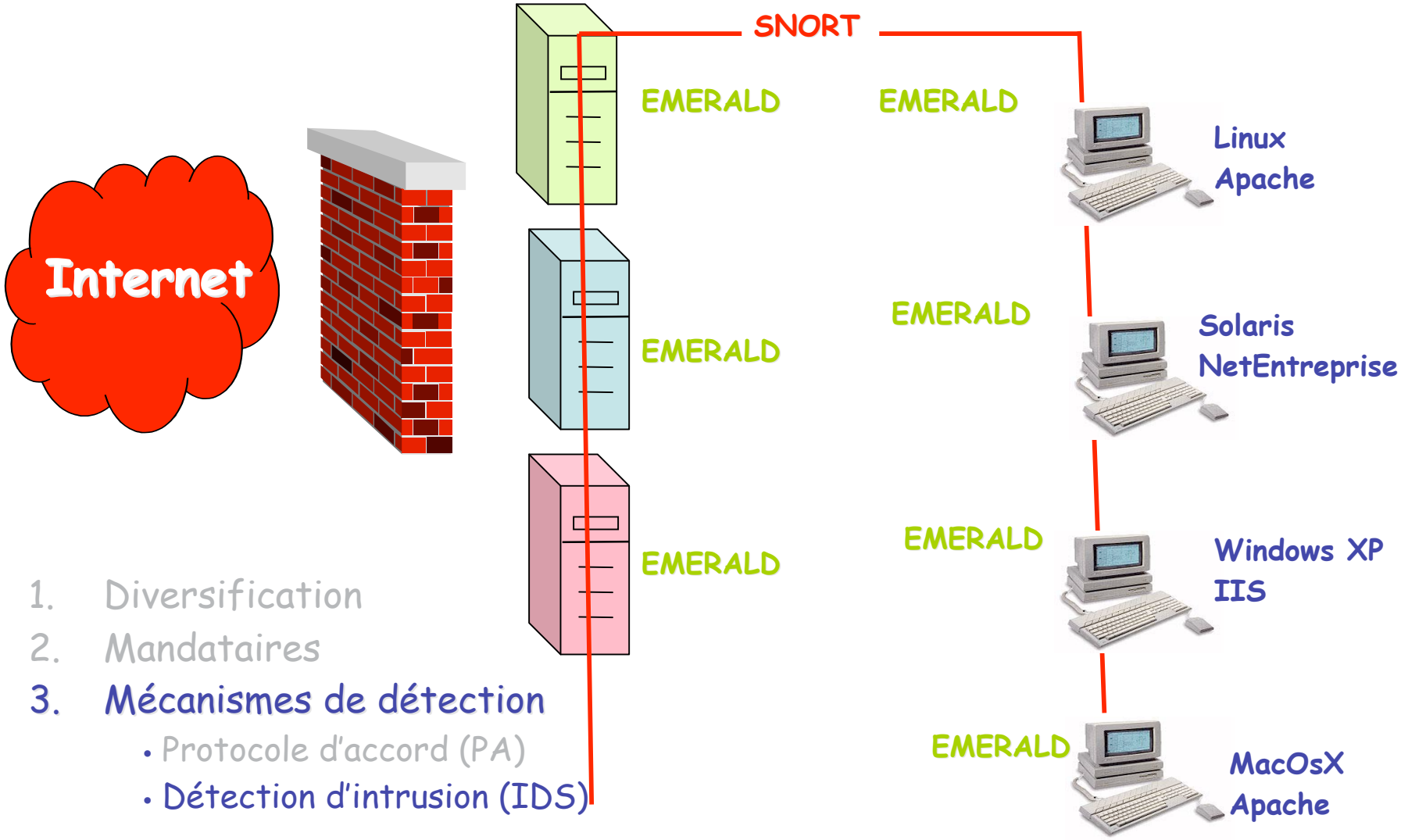
1. Diversification
2. Mandataires
   - Envoie les requêtes aux serveurs (Meneur)
   - Surveillance mutuelle
   - Surveillance des serveurs d'application
   - Gestion des alertes

Un serveur tolérant les intrusions



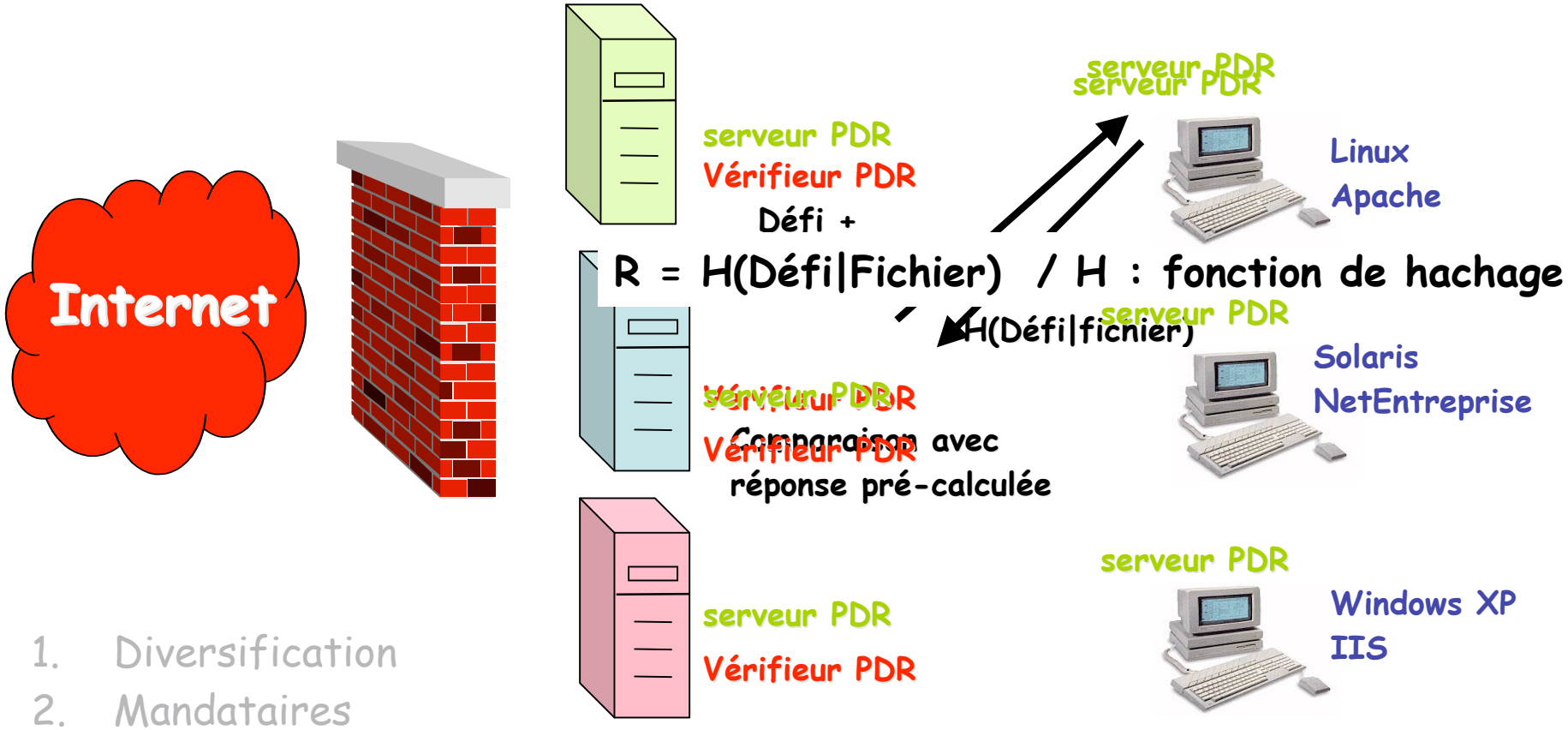
1. Diversification
2. Mandataires
3. Mécanismes de détection
   - Protocole d'accord (PA)

Un serveur tolérant les intrusions



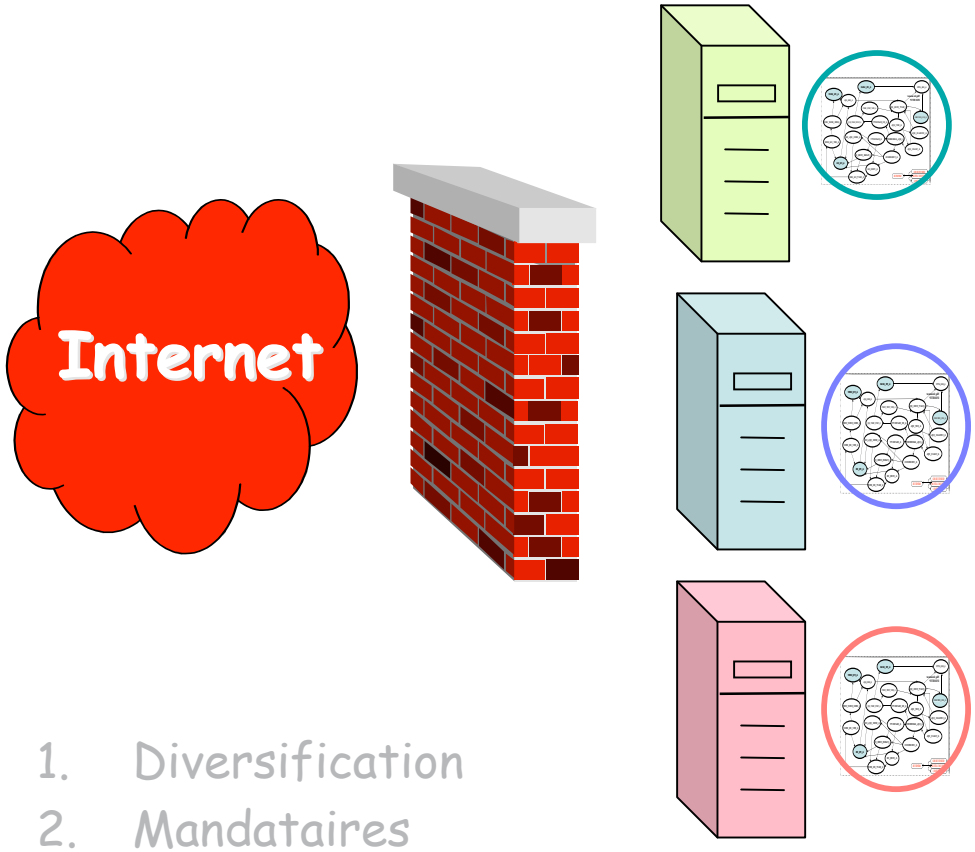
1. Diversification
2. Mandataires
3. Mécanismes de détection
   - Protocole d'accord (PA)
   - Détection d'intrusion (IDS)

Un serveur tolérant les intrusions



1. Diversification
2. Mandataires
3. Mécanismes de détection
   - Protocole d'accord (PA)
   - Détection d'intrusion (IDS)
   - Protocole de Défi-Réponse

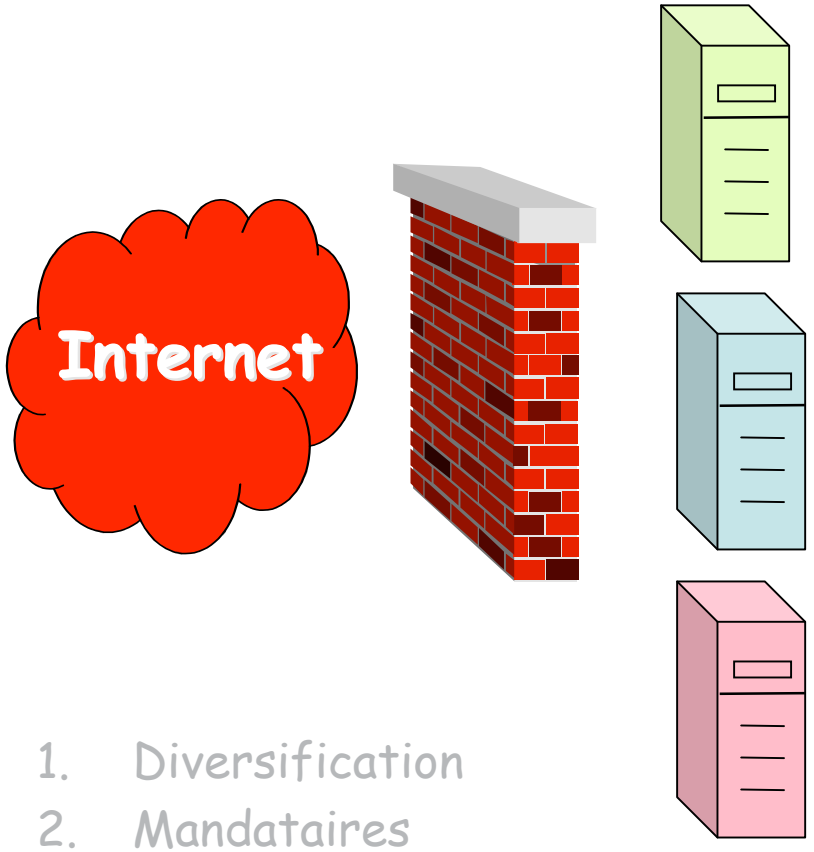
Un serveur tolérant les intrusions



Modèle de comportement = automate à états finis

1. Diversification
2. Mandataires
3. Mécanismes de détection
   - Protocole d'accord (PA)
   - Détection d'intrusion (IDS)
   - Protocole de défi-réponse
   - Vérification de modèle en ligne

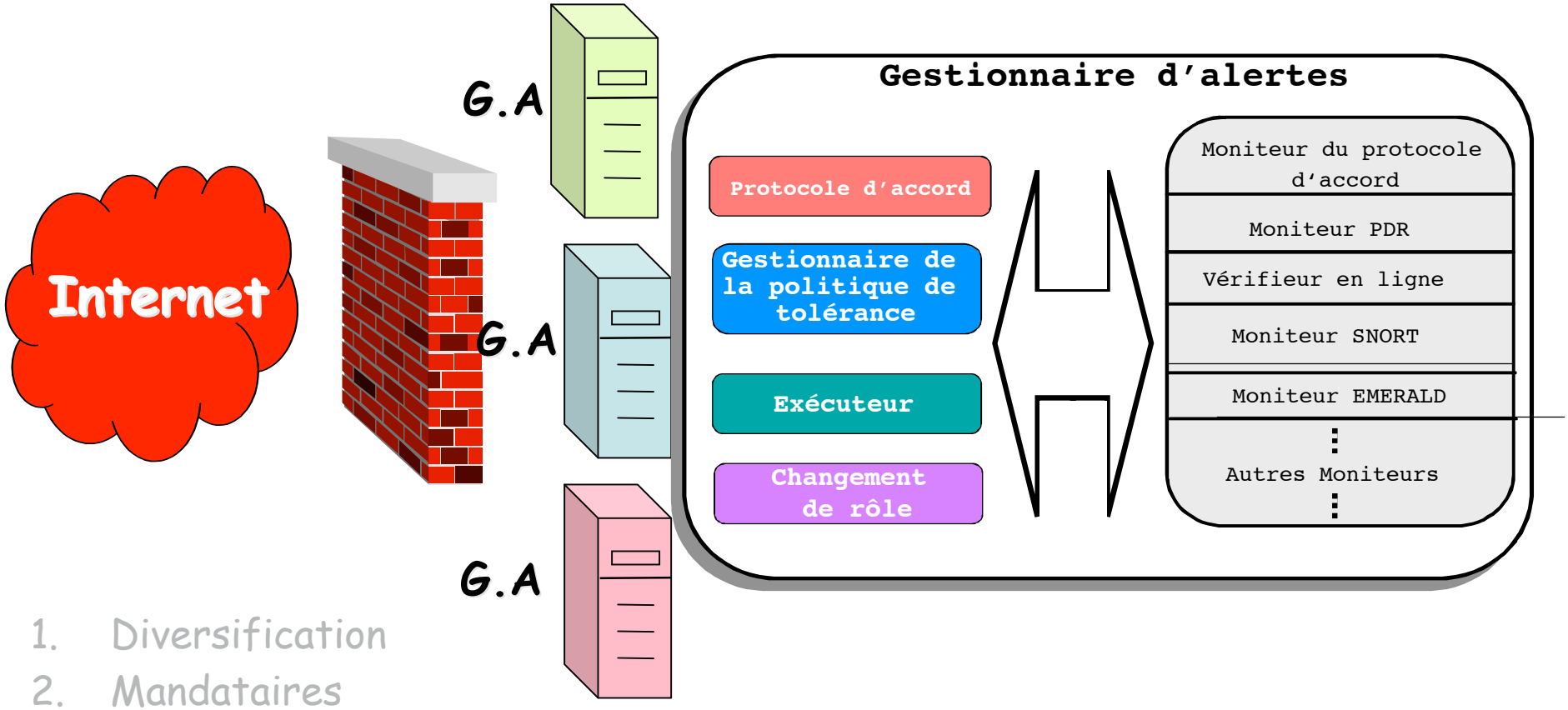
Un serveur tolérant les intrusions



1. Diversification
2. Mandataires
3. Mécanismes de détection
4. Gestionnaire des alertes (G.A)

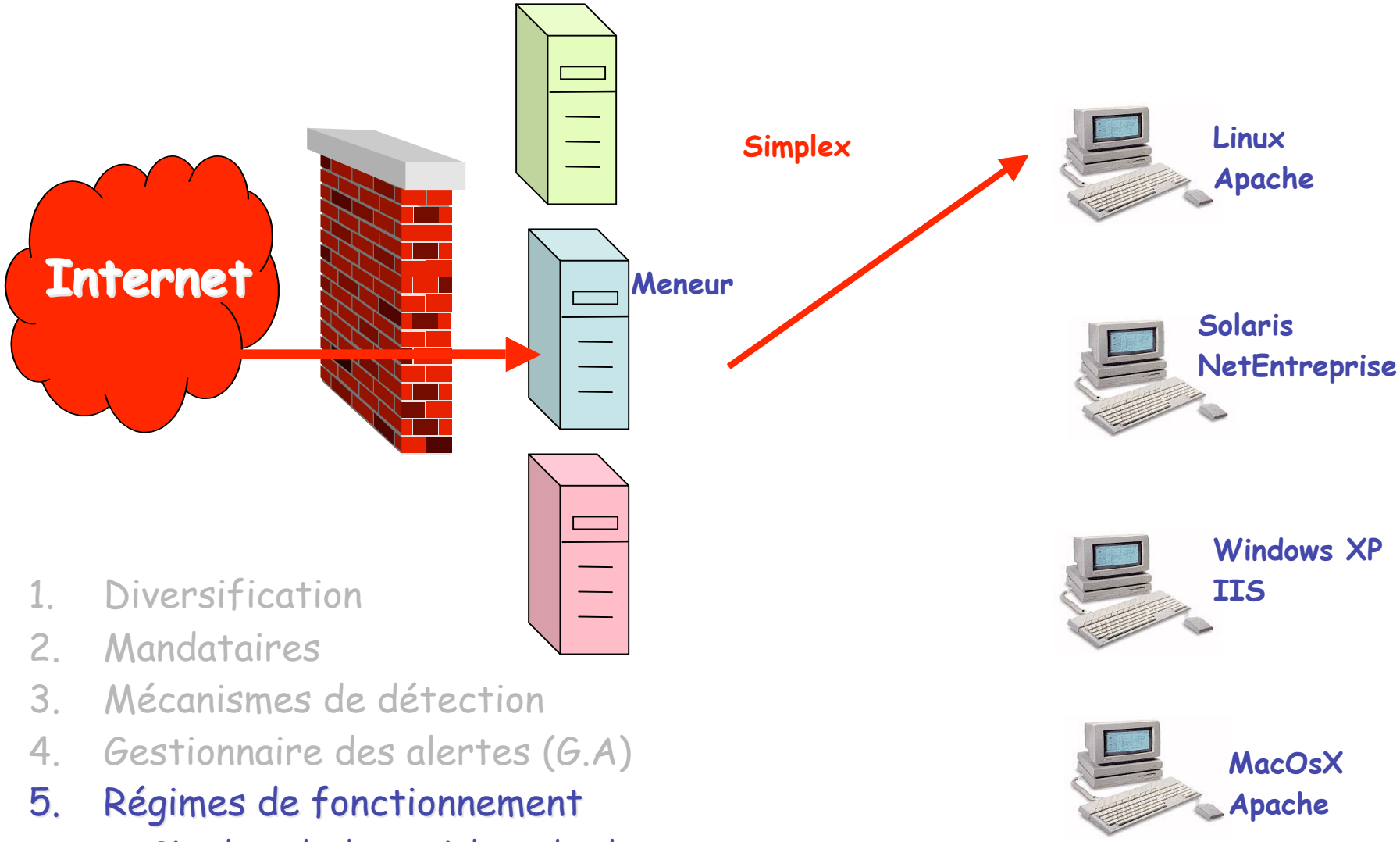
1. Reçoit et traite les alertes
2. Décide des contre-mesures pour réagir à la détection d'une attaque / anomalie

Un serveur tolérant les intrusions



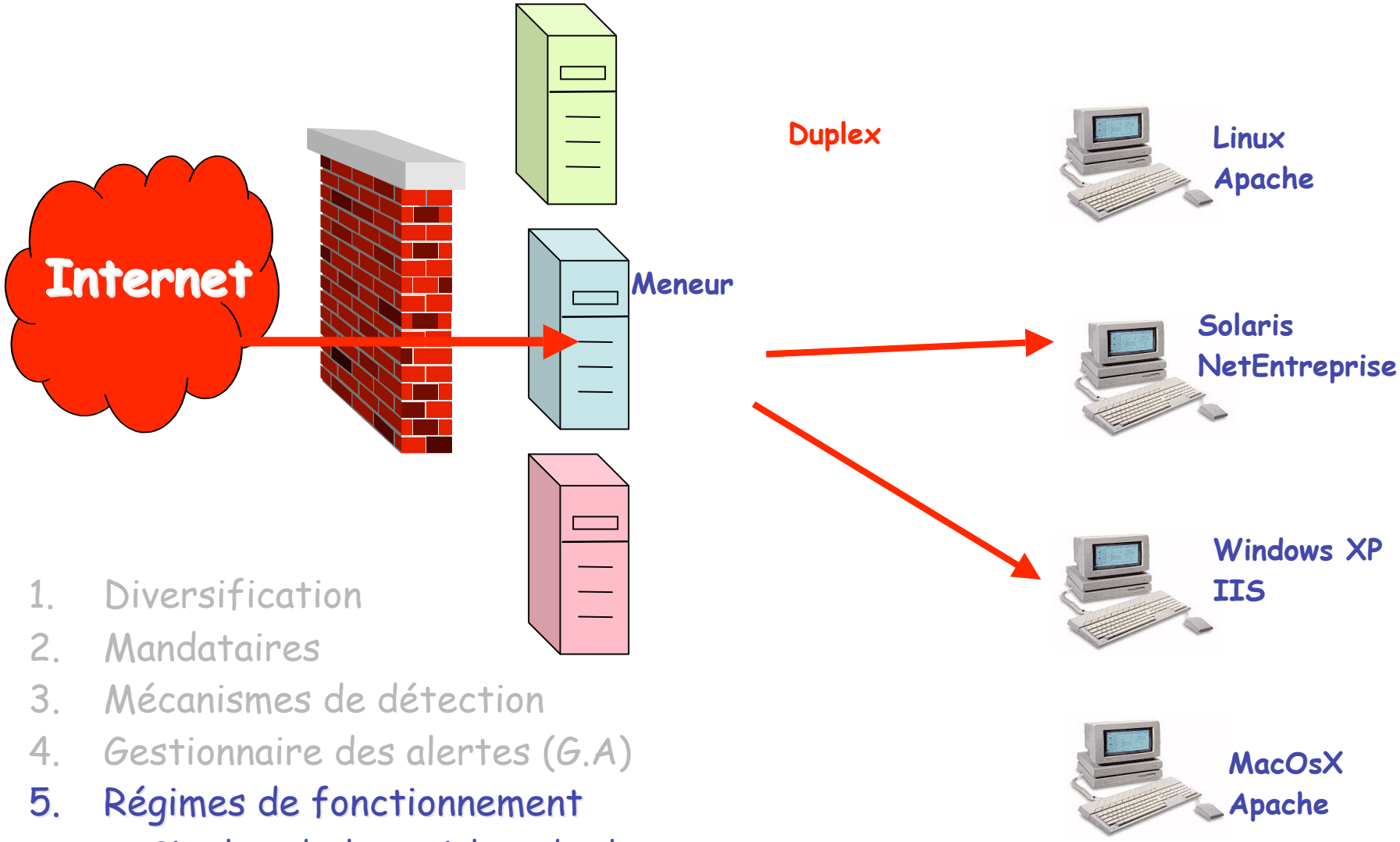
1. Diversification
2. Mandataires
3. Mécanismes de détection
4. Gestionnaire des alertes (G.A)

Un serveur tolérant les intrusions



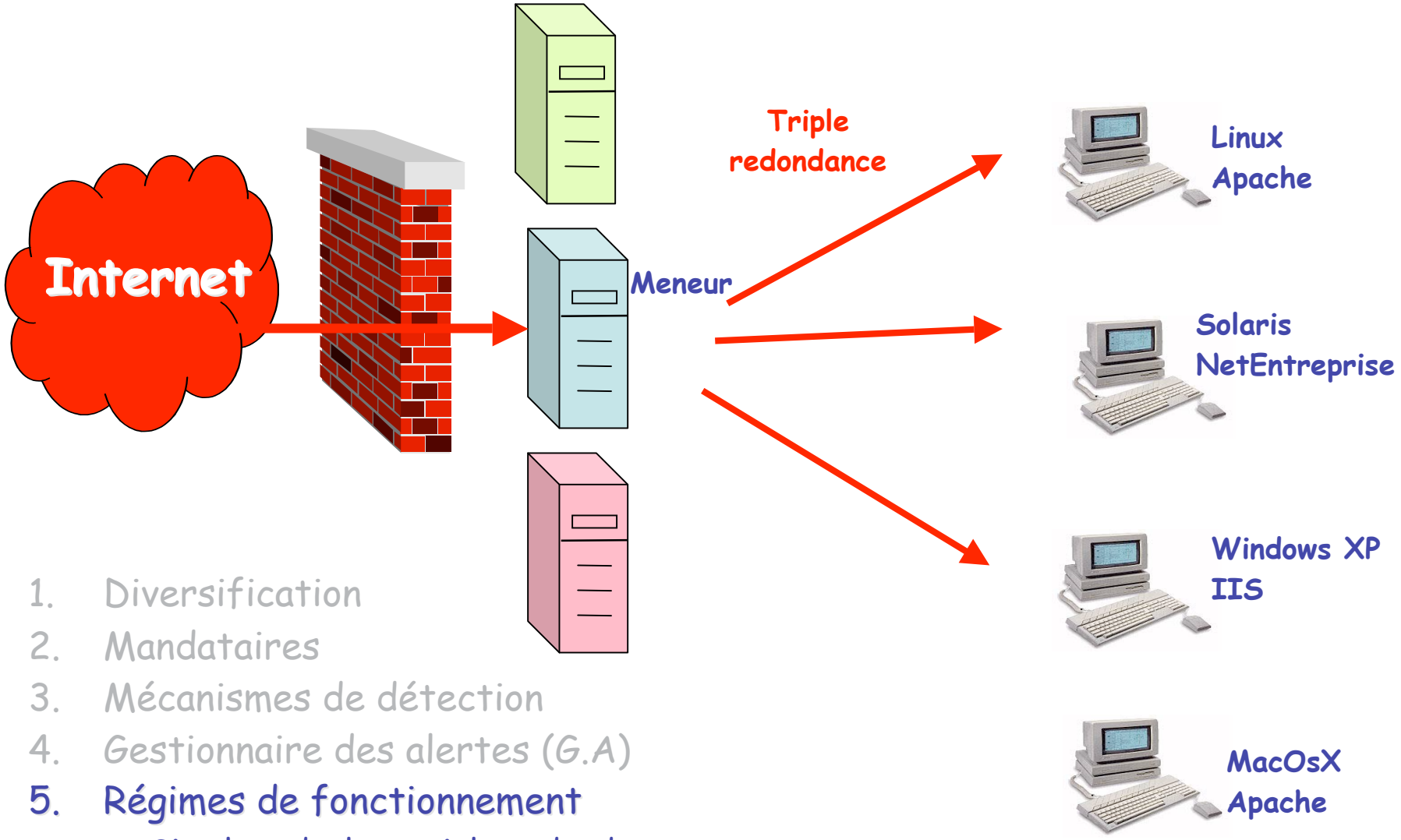
1. Diversification
2. Mandataires
3. Mécanismes de détection
4. Gestionnaire des alertes (G.A)
5. Régimes de fonctionnement
   - Simplex, duplex, triple redondance…

Un serveur tolérant les intrusions



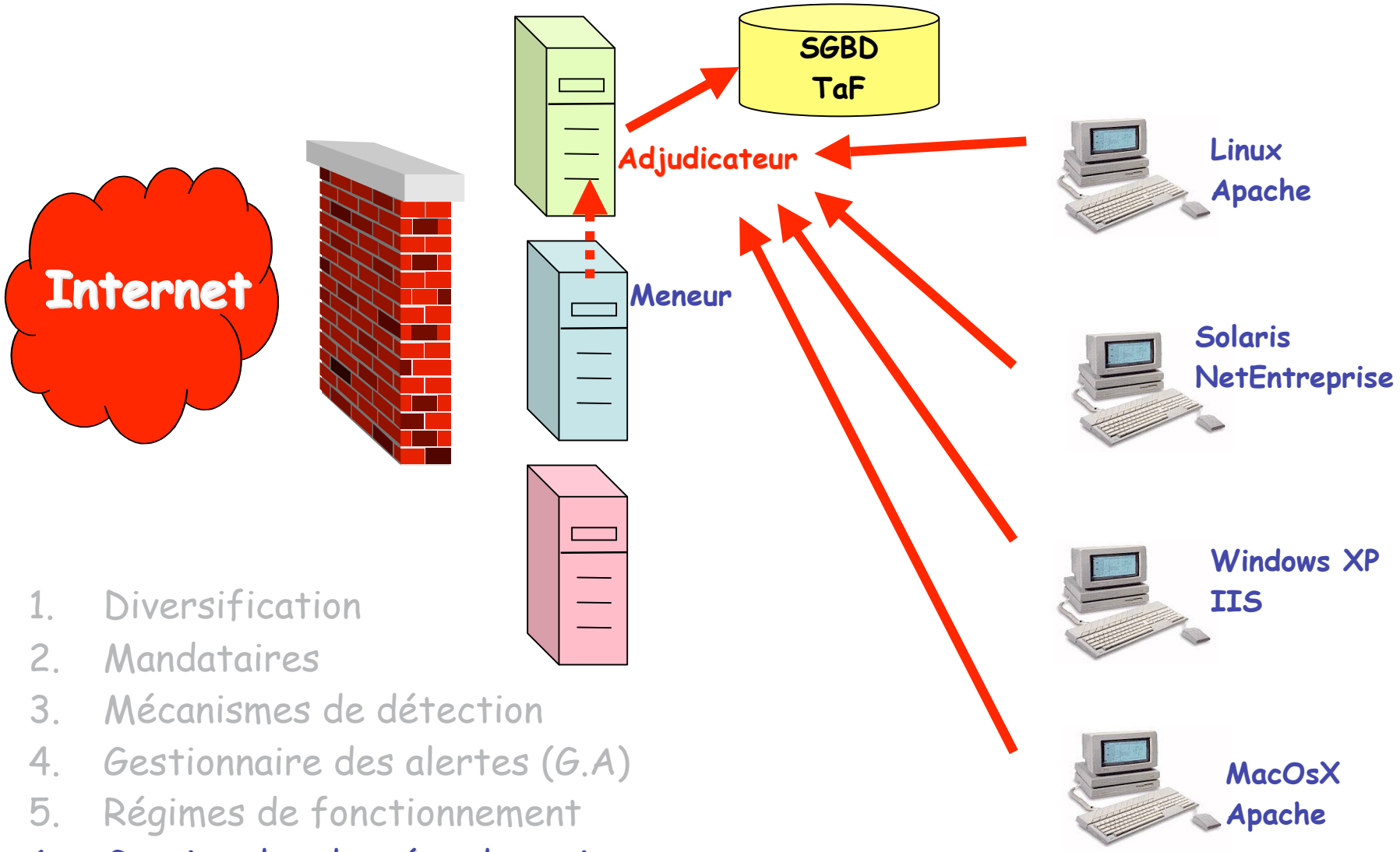
1. Diversification
2. Mandataires
3. Mécanismes de détection
4. Gestionnaire des alertes (G.A)
5. Régimes de fonctionnement
   - Simplex, duplex, triple redondance…

Un serveur tolérant les intrusions



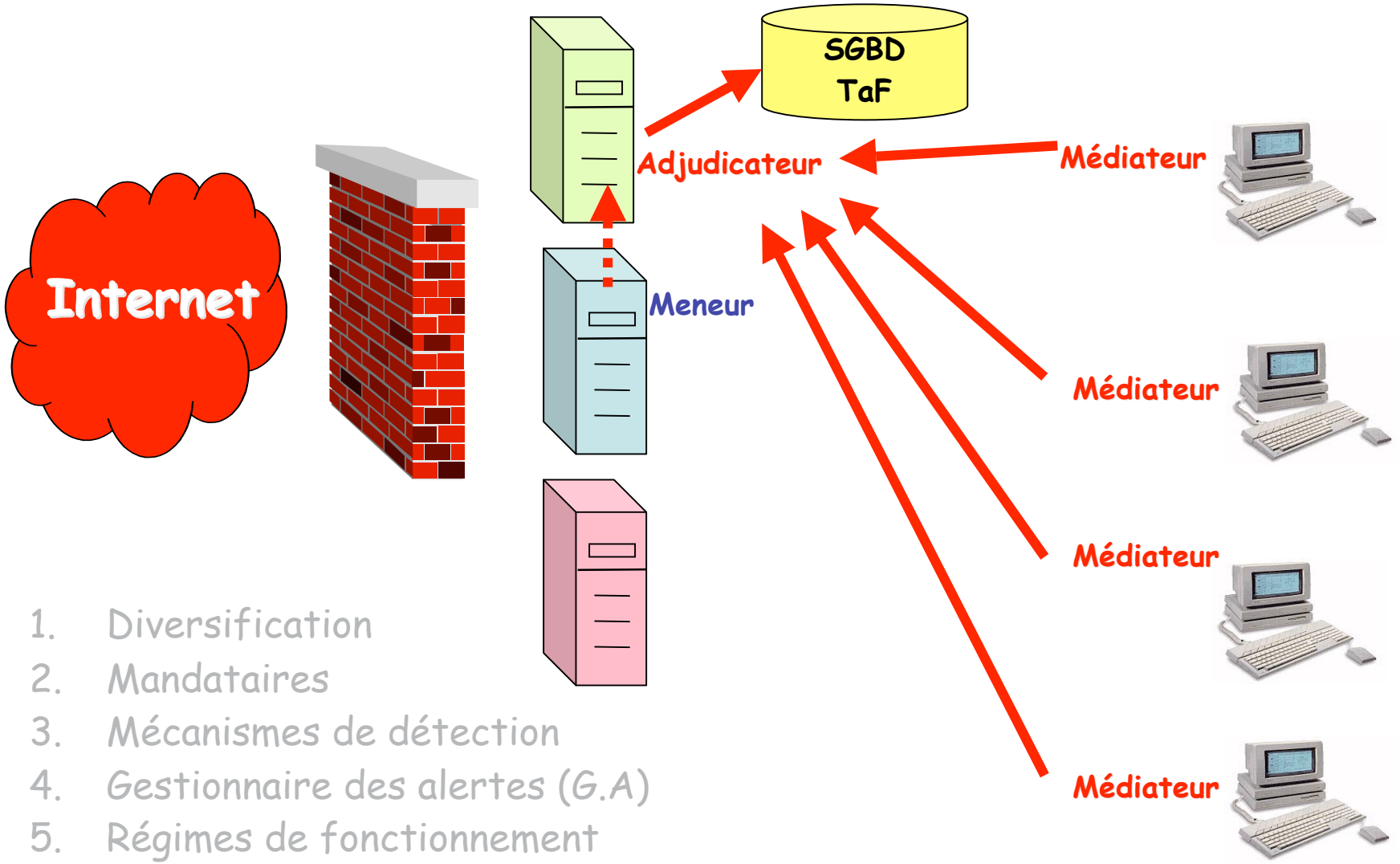
1. Diversification
2. Mandataires
3. Mécanismes de détection
4. Gestionnaire des alertes (G.A)
5. Régimes de fonctionnement
   - Simplex, duplex, triple redondance…

Un serveur tolérant les intrusions



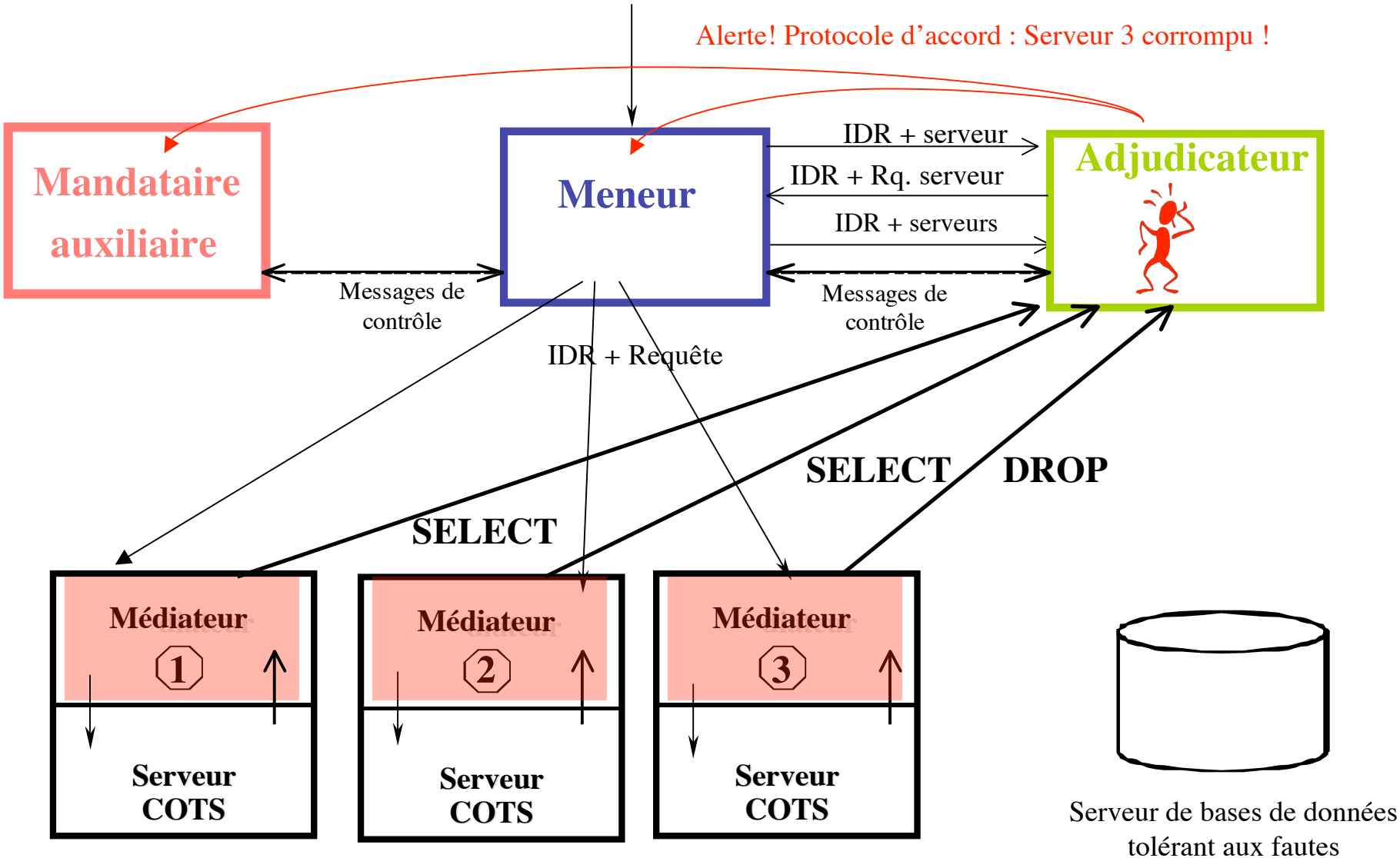
1. Diversification
2. Mandataires
3. Mécanismes de détection
4. Gestionnaire des alertes (G.A)
5. Régimes de fonctionnement
6. Gestion des données dynamiques

Un serveur tolérant les intrusions

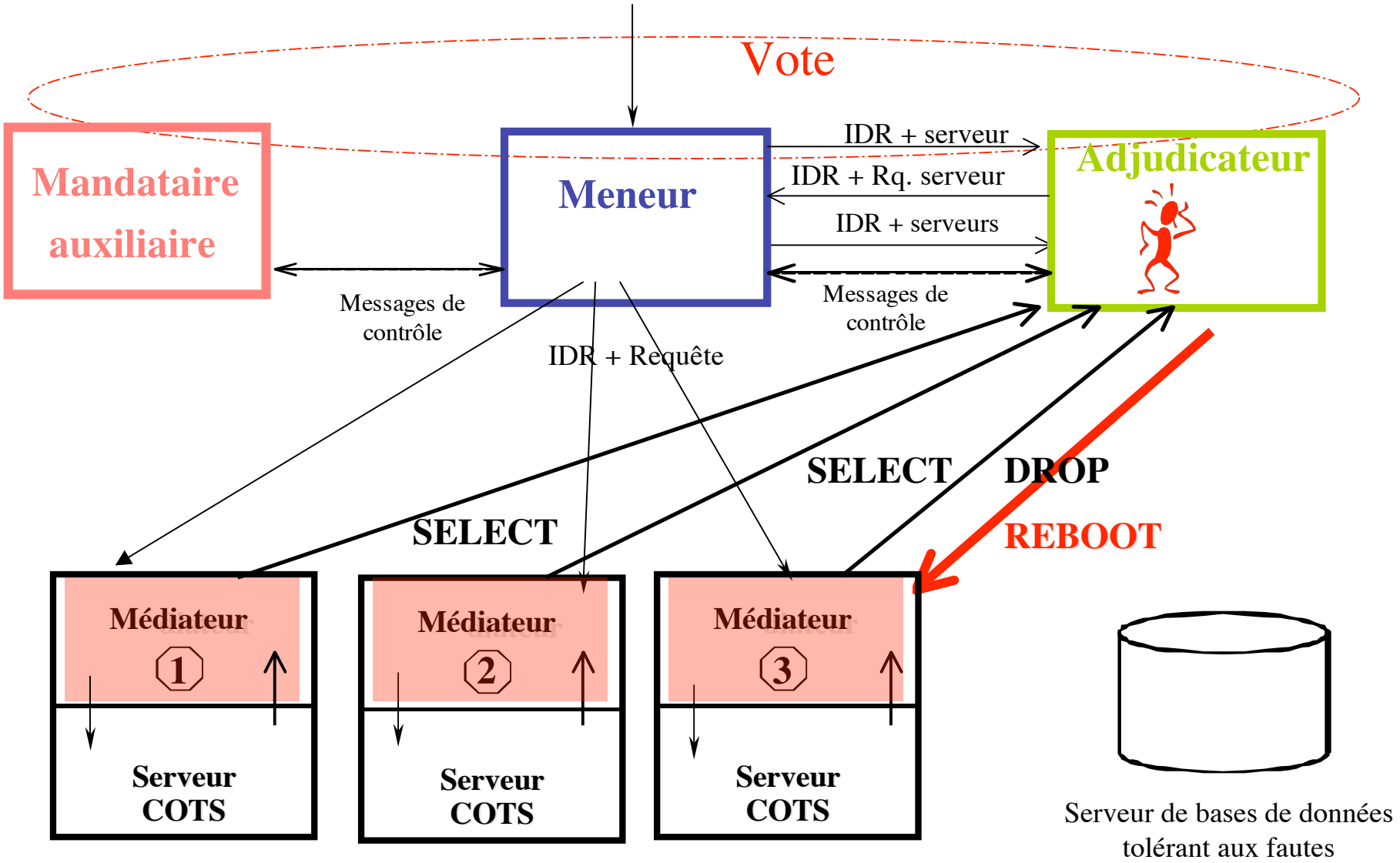


1. Diversification
2. Mandataires
3. Mécanismes de détection
4. Gestionnaire des alertes (G.A)
5. Régimes de fonctionnement
6. Gestion des données dynamiques

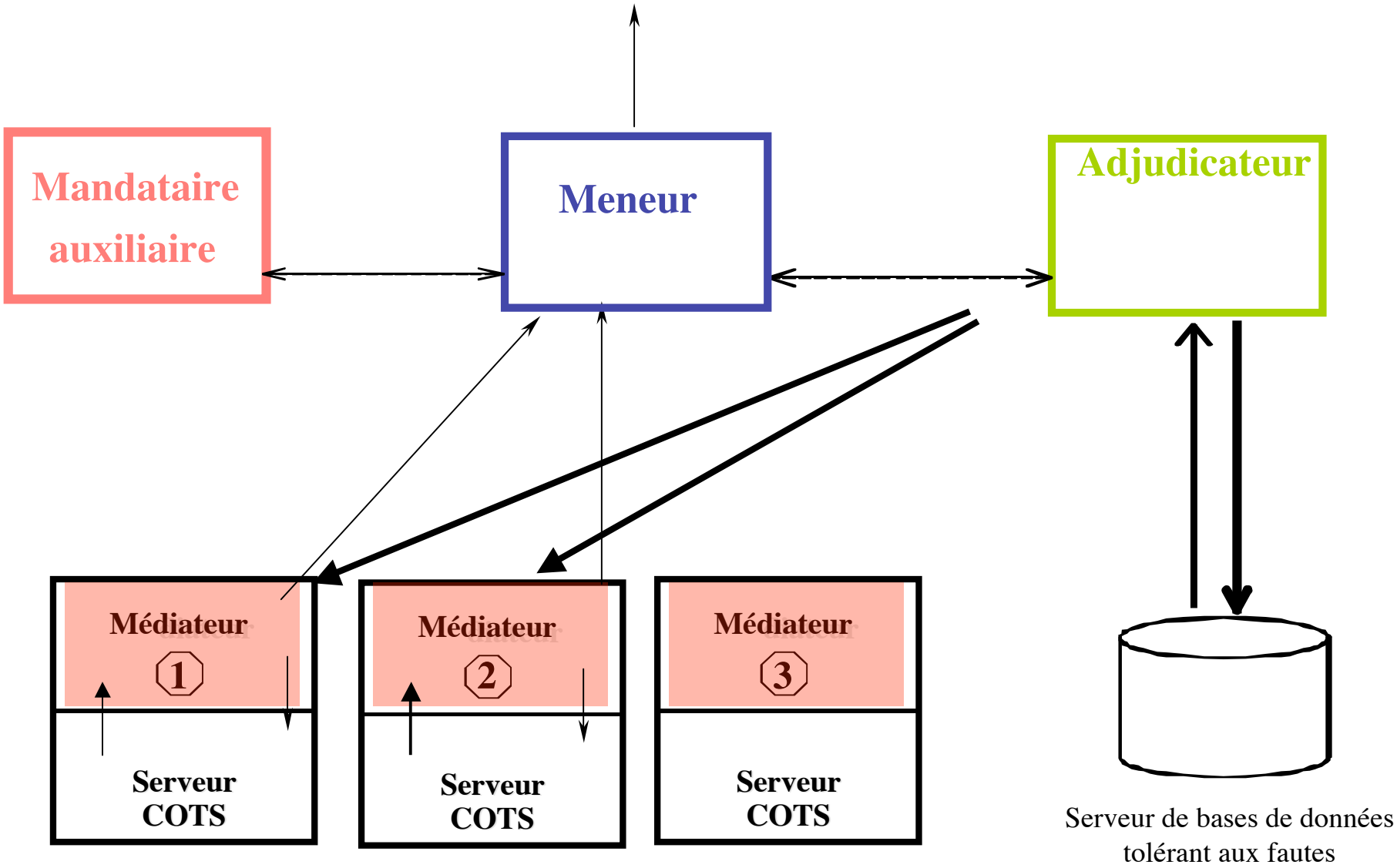
Un serveur tolérant les intrusions



Un serveur tolérant les intrusions



Un serveur tolérant les intrusions



Objectifs

Un serveur tolérant les intrusions

Comportement en cas d'attaques

Mise en œuvre et mesures de performances

Bilan et perspectives

Comportement en cas d'attaques

Quelle est le composant ciblé par l'attaque ?

1. Mandataires
2. Serveurs d'application ou
3. Base de données

Attaque contre les mandataires

- ✓ **Mécanismes de prévention**
  - a. Des techniques de développement de logiciels critiques sont utilisés pour le développement des codes des mandataires
  - b. Système d'exploitation « allégé »
  - c. Un seul mandataire accessible de l'extérieur
  - d. Pare-feu

- ✓ **Mécanismes de tolérance**
  - a Mécanismes de détection diversifiés
  - b Les mandataires se surveillent mutuellement

Attaque contre les serveurs d'application

- ✓ Mécanismes de prévention
  a. Les types et nombre de serveurs répondants à une requête donnée sont inconnus
  b. Le Meneur rejette les requêtes suspectes

- ✓ Mécanismes de tolérance
  a. L'attaque ne peut réussir que sur une petite minorité (1 seul) des serveurs (diversification matériel/OS/logiciel)
  b. Les mécanismes de détection diversifiés

Attaque contre la base de données

- ✓ Mécanismes de prévention
  a. Pas d'accès direct à la base de données
  b. Bibliothèque spécifique d'accès à la base de données
  c. Vérification syntaxique sur le Médiateur/Adjudicateur

- ✓ Mécanismes de tolérance
  a. Base de données tolérante aux fautes accidentelles
  b. Protocole d'accord sur l'Adjudicateur
  c. Recouvrement d'un Adjudicateur corrompu

Recouvrement d'une intrusion

1. Recouvrement du Meneur ou de l'Adjudicateur
   - ✓ Élection d'un nouveau Meneur/Adjudicateur
   - ✓ Redémarrage du Meneur/Adjudicateur corrompu
   - ✓ Si l'adjudicateur est corrompu, défaire les dernières modifications sur la base de données

2. Recouvrement d'un mandataire auxiliaire
   - ✓ Redémarrage du mandataire corrompu

3. Recouvrement d'un serveur d'application
   - ✓ Redémarrage du serveur corrompu
   - ✓ Renvoi des requêtes en cours à d'autres serveurs (si besoin)

Objectifs

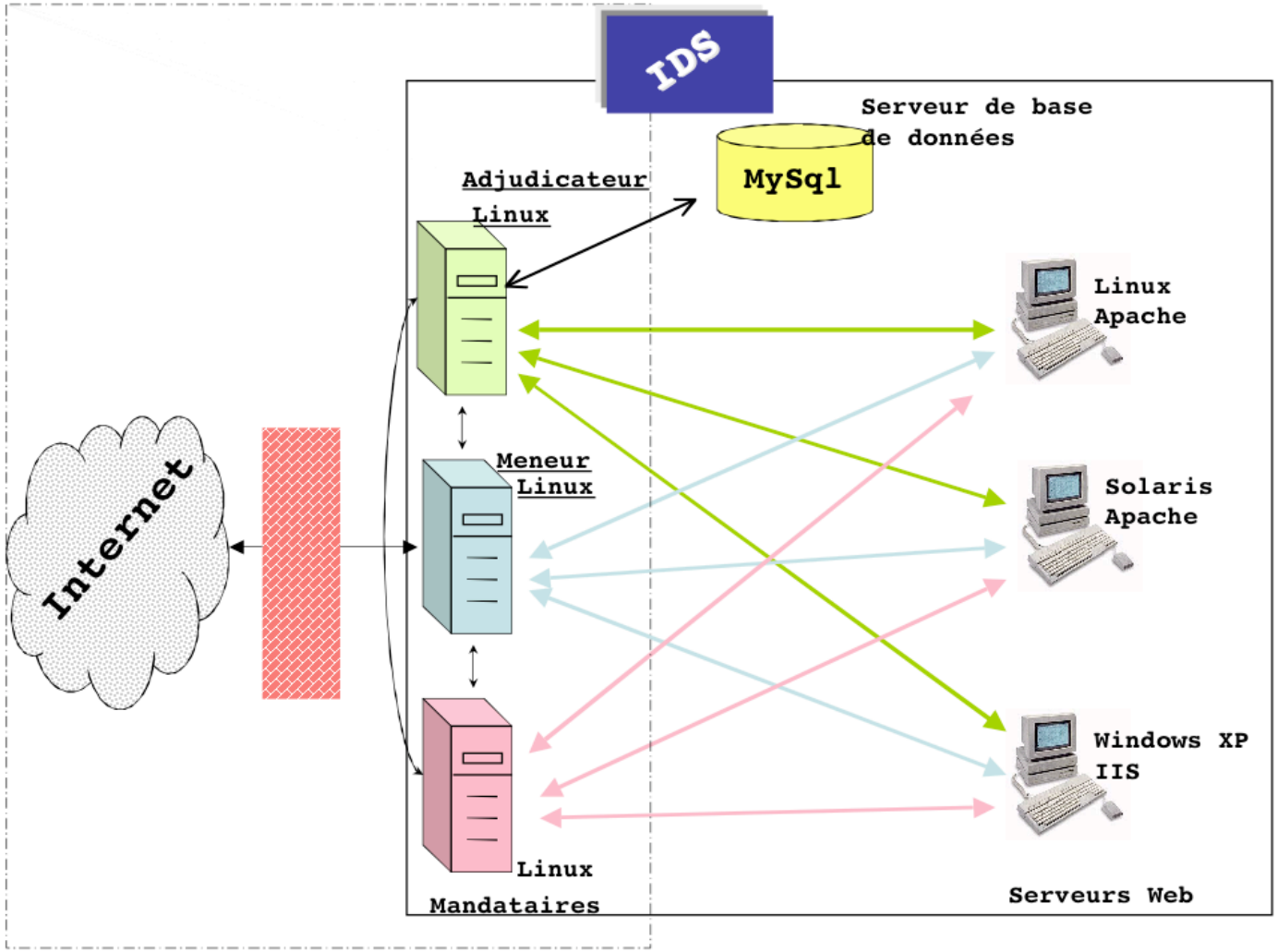
Un serveur tolérant les intrusions

Comportement en cas d'attaques

Mise en œuvre et mesures de performances

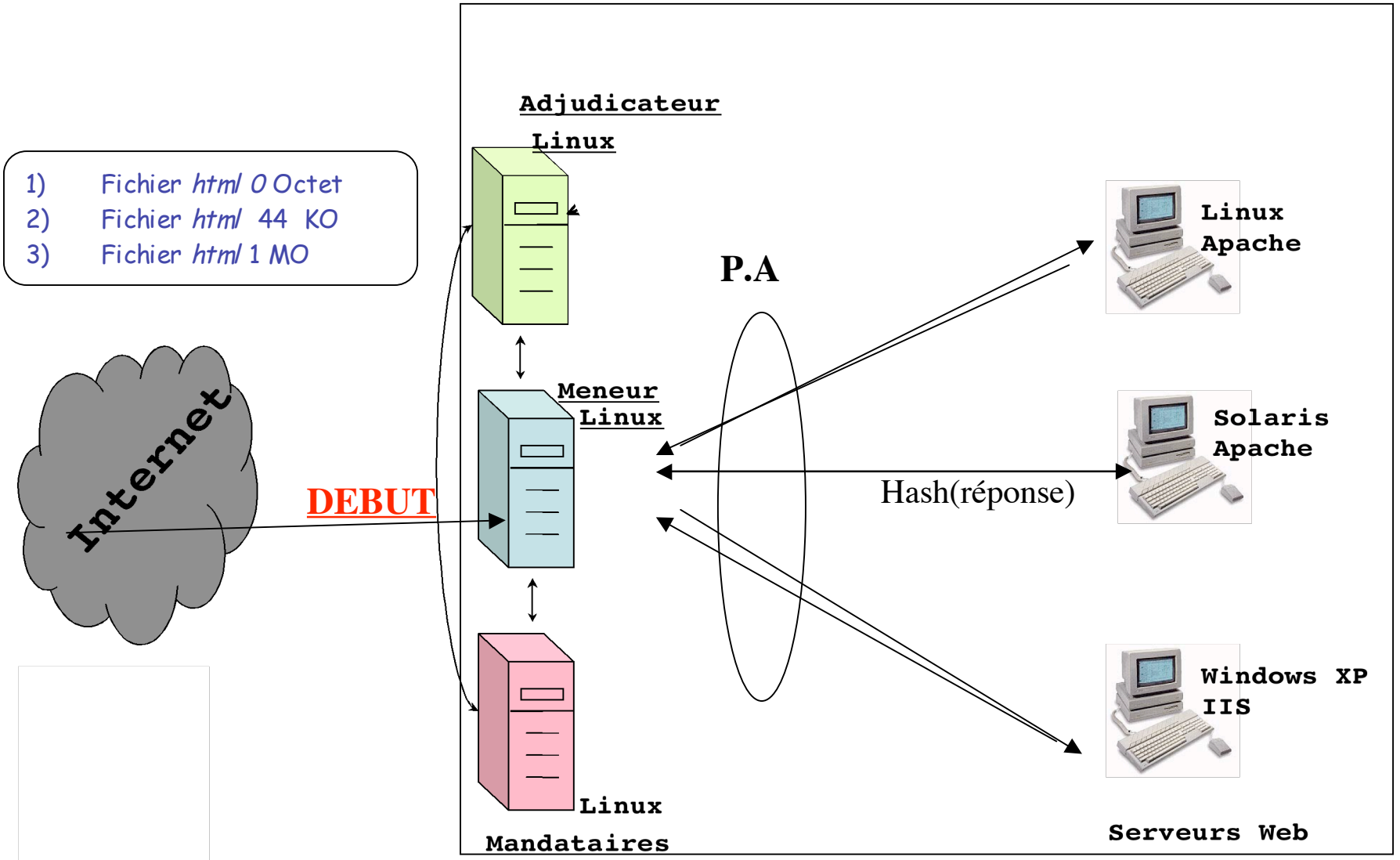
Bilan et perspectives

Plateforme expérimentale



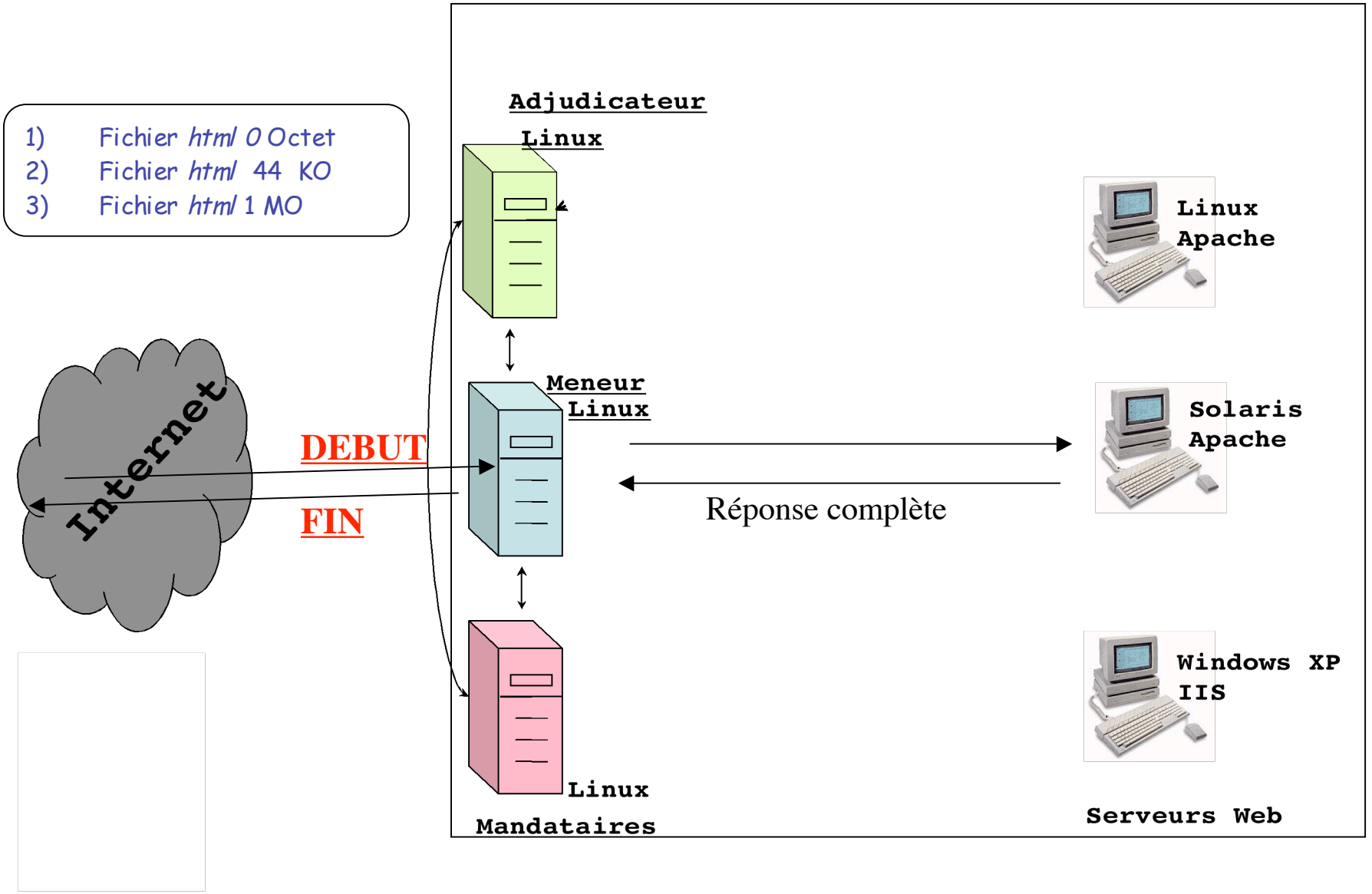
Première mesure

Temps de traitement d'une requête sans accès à la base de données



Première mesure

Temps de traitement d'une requête sans accès à la base de données



Première mesure

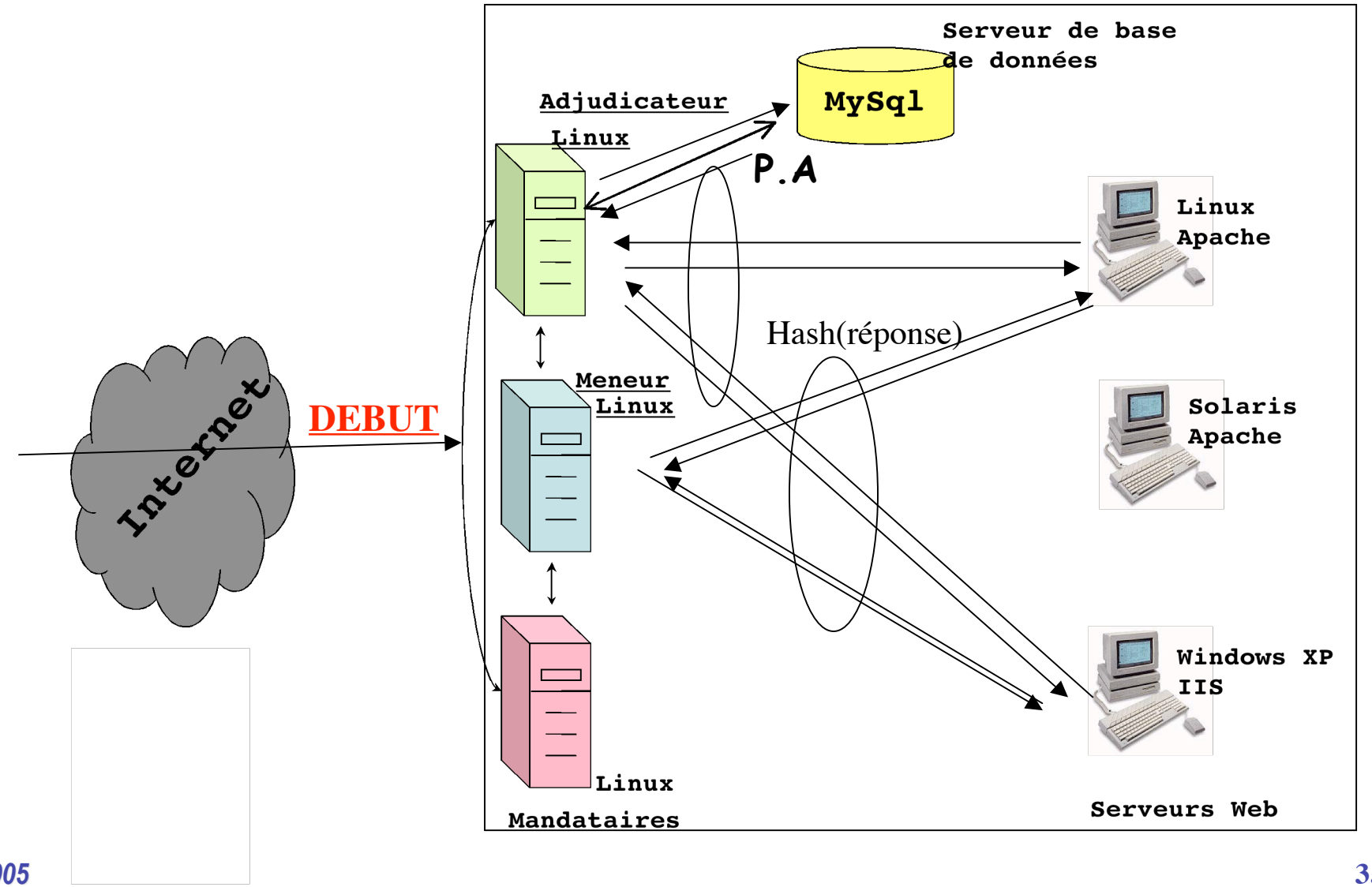
Temps de traitement d'une requête sans accès à la base de données

	Direct	Simplex	Duplex	Triplex
0 octet	0,0037 sec	0,0074 sec	0,0087 sec	0,0096 sec
44 Ko	0,0115 sec	0,0145 sec	0,0167 sec	0,0170 sec
1 Mo	0,14 sec	0,316 sec	0,321 sec	0,322 sec

	Direct → Simplex	Simplex → Duplex	Duplex → Triplex
0 octet	100%	18%	10%
44 Ko	26%	15%	2,2%
1 Mo	131%	1,49%	0,34%

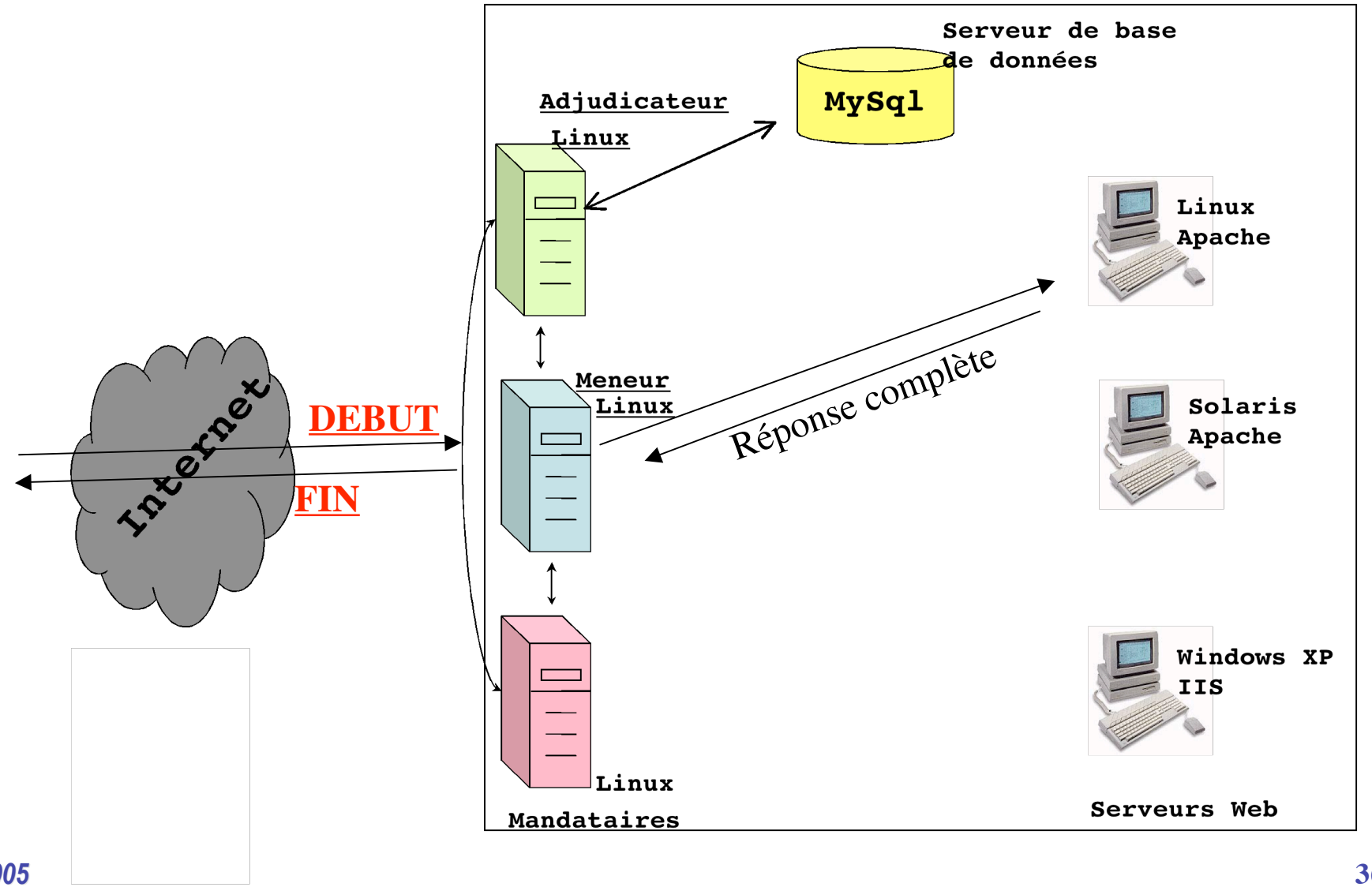
Deuxième mesure

Temps de traitement d'une requête avec accès à la base de données



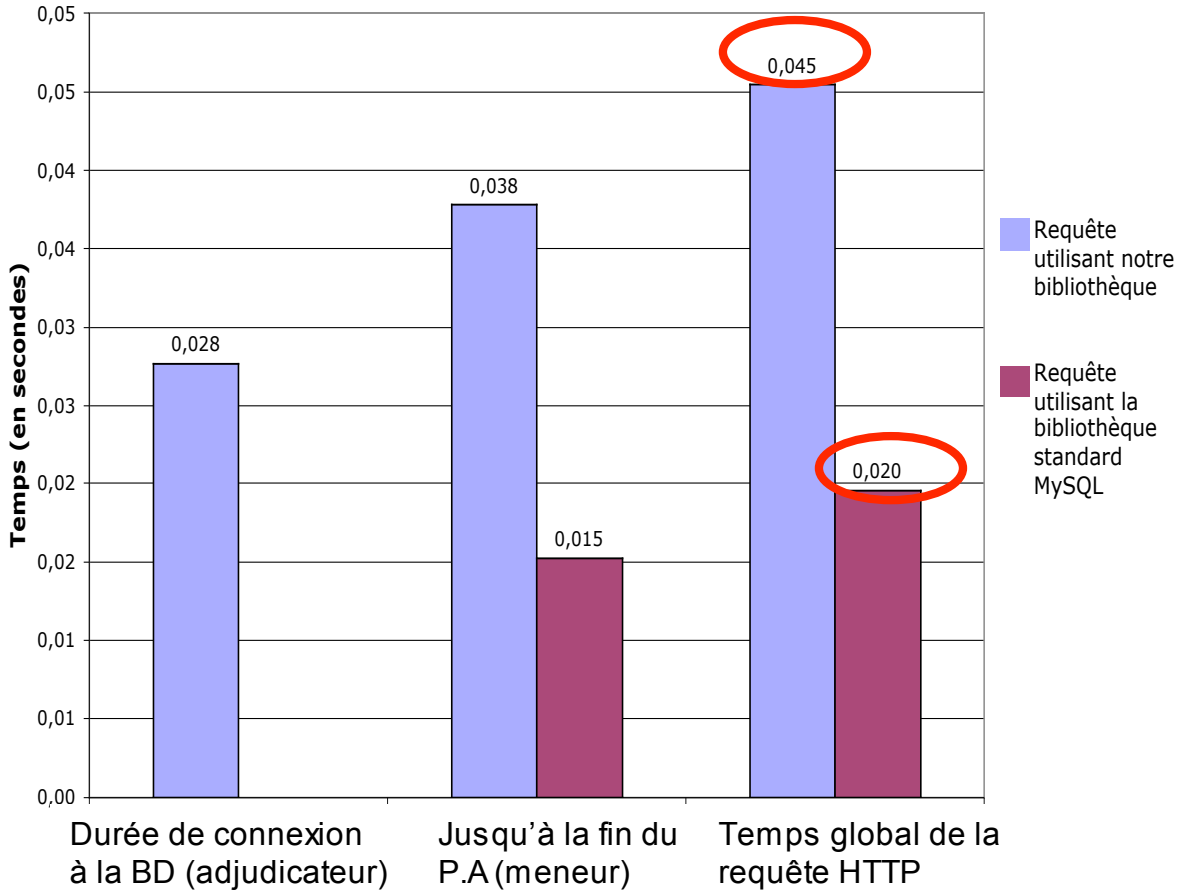
Deuxième mesure

Temps de traitement d'une requête avec accès à la base de données



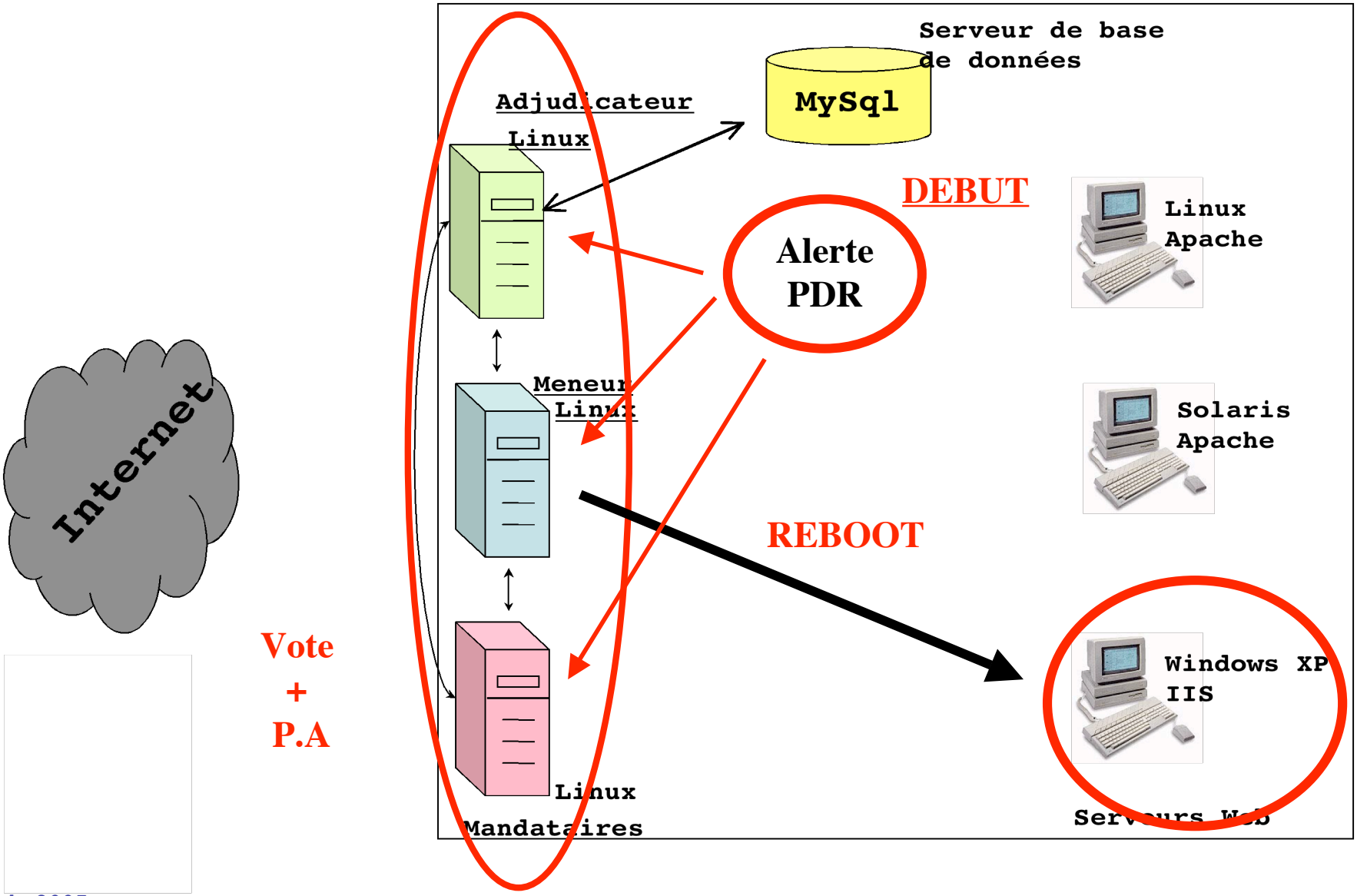
Deuxième mesure

Temps de traitement d'une requête avec accès à la base de données



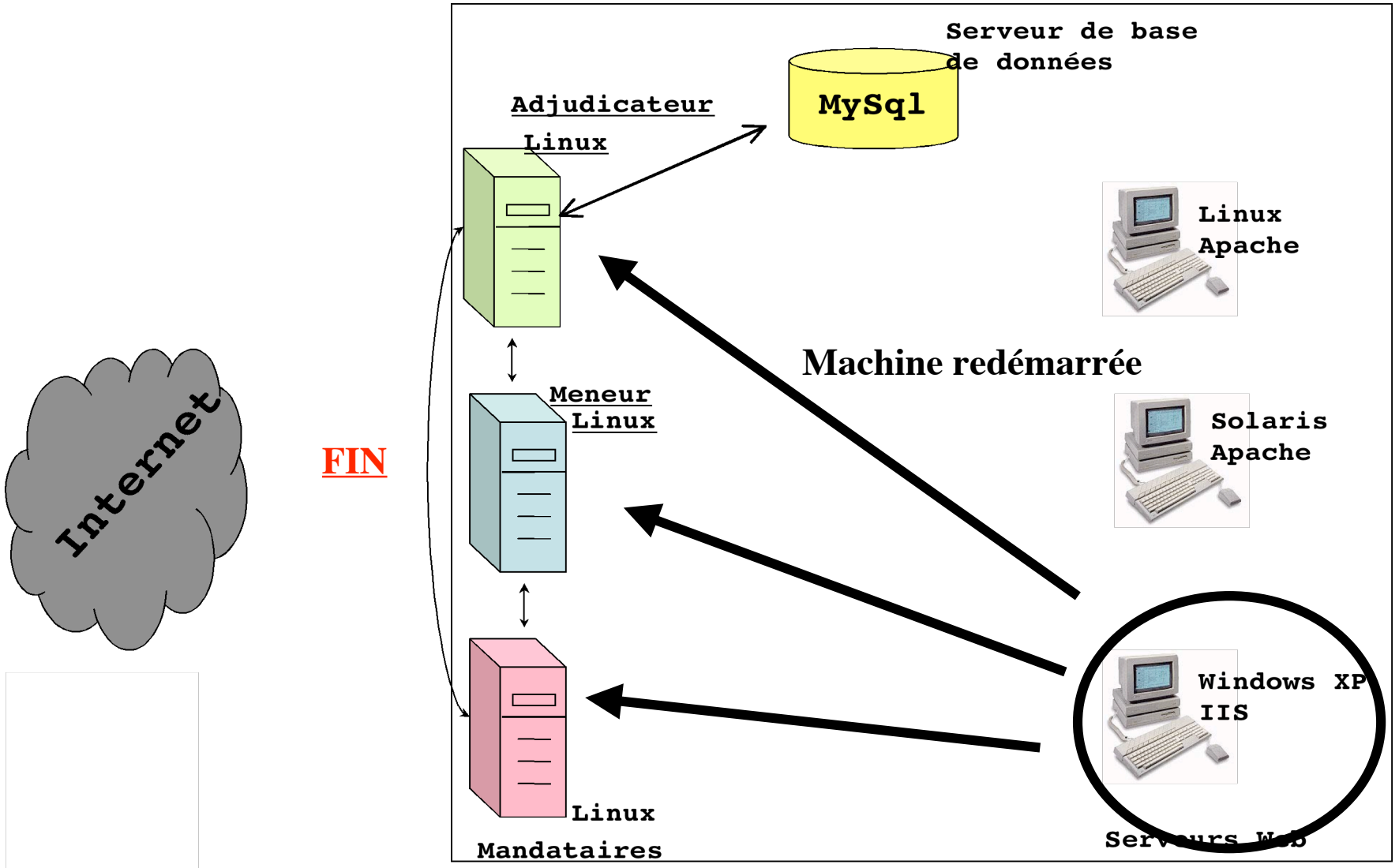
Troisième mesure

Temps de réponse à une alerte avec redémarrage du serveur corrompu



Troisième mesure

Temps de réponse à une alerte avec redémarrage du serveur corrompu



Troisième mesure

Temps de réponse à une alerte avec redémarrage du serveur corrompu

	Temps jusquà isolation du serveur corrompu	Temps global de traitement de l'alerte
Moyenne	0,344 sec	73,2 sec
Min	0,00876 sec	70,3 sec
Max	1,015 sec	75,8 sec

Détails du traitement d'une alerte concernant un serveur corrompu

Objectifs

Un serveur tolérant les intrusions

Comportement en cas d'attaques

Mise en œuvre et mesures de performances

Bilan et perspectives

Bilan

Une architecture générique tolérant les intrusions, répondant à trois exigences fixées au préalable :

- utilisation des composants COTS
- assurer l'intégrité des réponses et disponibilité du service, même pendant les attaques
- un bon compromis entre sécurité et performances.

- ✓ Intégration de différentes techniques de prévention, détection et tolérance pour les fautes accidentelles les malveillances
- ✓ Redondance avec diversification / Niveau de redondance variable
- ✓ Gestion des alertes de différents niveaux de crédibilité
- ✓ Un protocole de défi-réponse pour l'intégrité de serveurs distants
- ✓ Performances prometteuses

Perspectives

- Evaluation de la généricité de l'architecture
  - ✓ Application de l'architecture à d'autres services
- Application de l'architecture à des systèmes de très grande taille
  - ✓ Evaluer les limites de l'architecture de point de vue robustesse et performance
- Évaluation quantitative et expérimentale de la sécurité
- Des expérimentations plus poussées pour les mesures de performances
- Validation formelle du déterminisme