WASD en environnement haute disponibilité

Jean-Pierre PETIT

WASD

Le produit

WASD

- WASD signifie
  - Wide Area Surveillance Division
- C’est le nom d’une division de la défense australienne qui a accepté en 1996 que ce produit développé à usage interne soit mis librement à disposition de la communauté VMS (licence GNU GPL).

L’auteur

- Mark Daniel a depuis 1996 employé beaucoup de son temps libre pour faire évoluer ce produit.
- Il doit être remercié pour:
  - la qualité de son travail (la qualité du code est exemplaire)
  - sa grande réactivité
  - son soucis de prise en compte des demandes d’évolution

WASD / environnement

- WASD est disponible pour les plateformes
  - VAX
  - Alpha
  - Itanium
- Fonctionne à partir de VMS 6.0
- Plusieurs piles IP acceptées:
  - Compaq/HP: TCPIP et UCX toutes versions
  - Process Software: MultiNet et TCPware

WASD et performances

- WASD est conçu spécifiquement pour VMS.
- Il n'utilise pas de *threads*.
- Il est entièrement basé sur le mécanisme d'AST.
- Il apporte ses propres mécanismes d'optimisation:
  - cache de fichiers
  - cache DNS
  - cache d'authentification (partagé entre instances)

WASD

Utilisation à l'ESME-Sudria

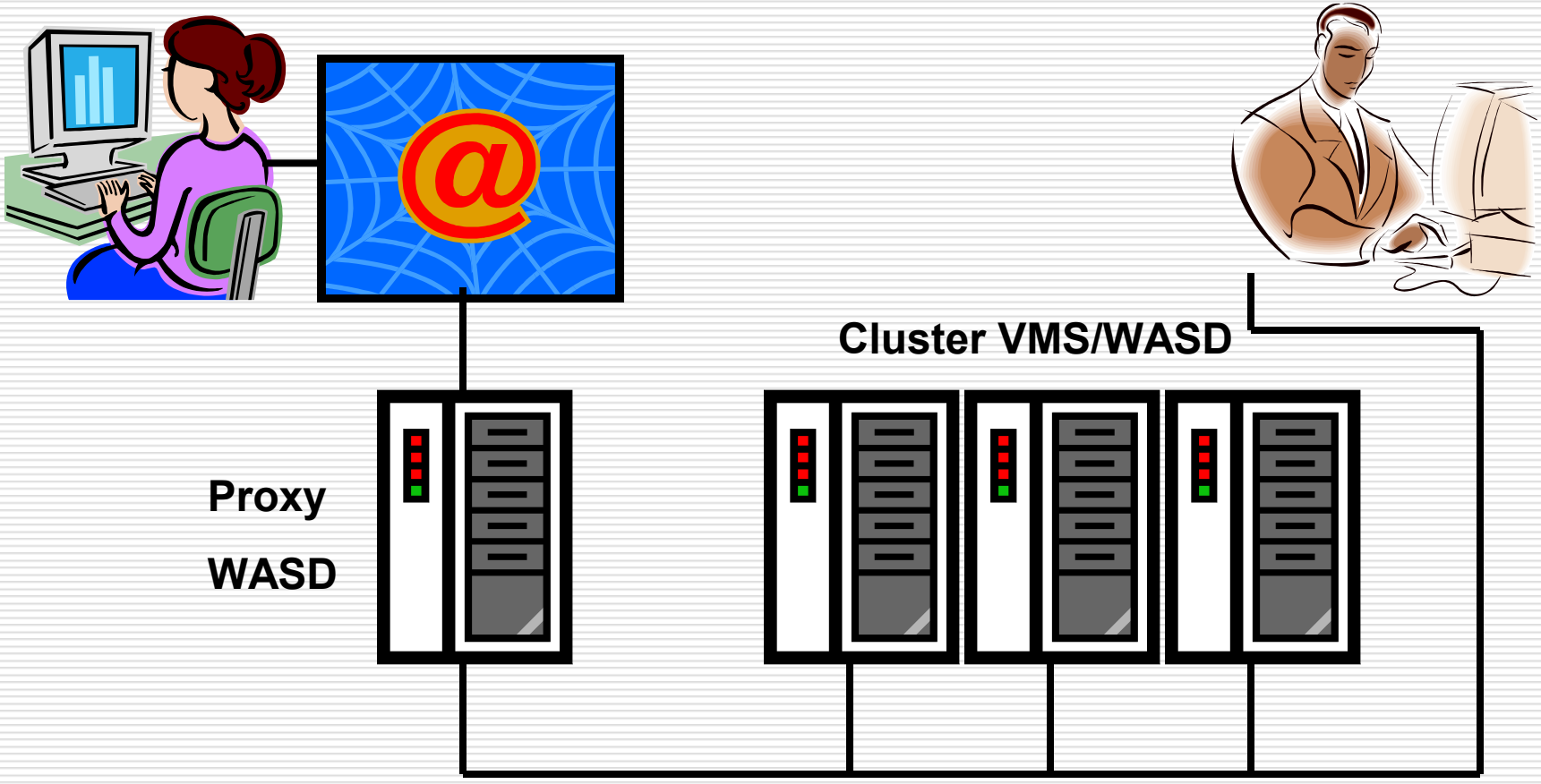
Les services Web à l'ESME-Sudria

- L'Intranet est depuis des années la référence d'information de l'école, son corps enseignant et ses étudiants.
- Il doit être à la fois sécurisé et accessible depuis Internet de façon à permettre la consultation par les étudiants depuis leur domicile.

Les services Web à l'ESME-Sudria

- Différents produits ont été testés/utilisés par le passé:
  - Purveyor
  - Netscape FastTrack
  - Apache
- WASD a été mis en place début 2001 à la plus grande satisfaction des utilisateurs.

Les services Web à l'ESME-Sudria

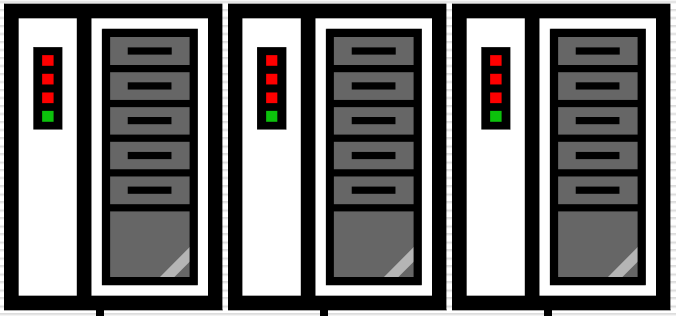


Serveurs Intranet

Site interne

intranet.esme.fr

WASD



Proxy

WASD

Serveurs Intranet

- 3 serveurs dans le même cluster
  - 2 serveurs Alpha / 1 serveur Itanium
- Les 3 serveurs partagent les mêmes disques donc les mêmes:
  - fichiers de configuration
  - pages statiques
  - scripts portables (DCL, PHP, Python,…)
- Un jeu de scripts compilés (.EXE) par architecture.

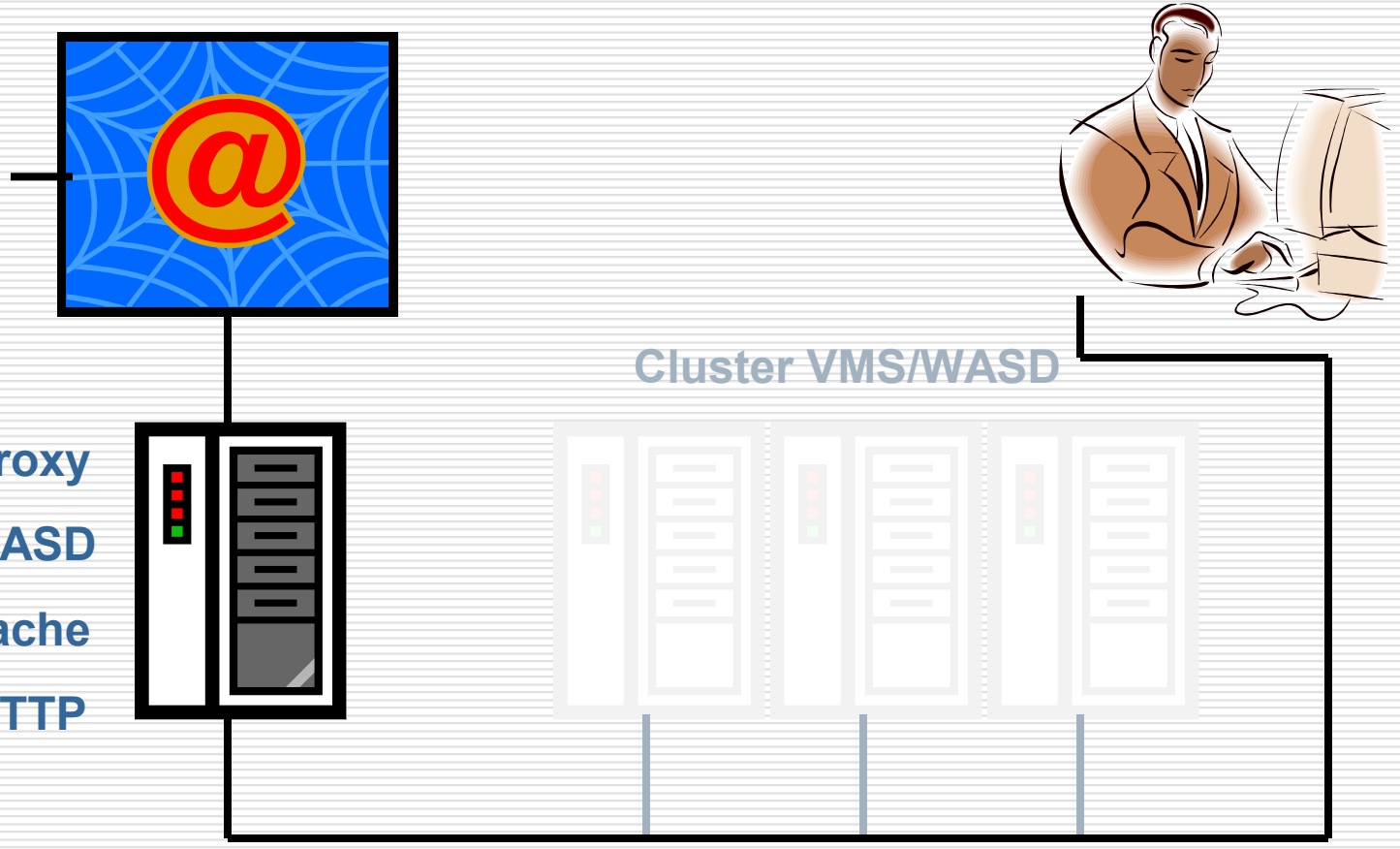
Serveurs Intranet (suite)

- La répartition des clients est de type *DNS round robin*.
- WASD autorise le déclenchement d'actions sur l'ensemble des serveurs d'un cluster via:
  - L'option /CLUSTER de la ligne de commande
  - La cache à cocher CLUSTER de l'interface web

Serveur proxy

- Il cumule plusieurs fonctions qui seront détaillées dans la suite.
- Deux instances de serveur HTTP tournent sur la même machine pour assurer:
  - la reprise immédiate en cas de crash d'une instance
  - la possibilité de redémarrage totalement transparent lors de:
    - changement de configuration
    - changement de version de WASD

Proxy standard d'accès à Internet



Proxy standard d’accès à Internet

- Outre l’accès à Internet, le proxy assure plusieurs fonctions:
  - cache global de contenu web
  - filtrage de sites indésirables
  - élimination d’éléments publicitaires
  - filtrage de ports pour la méthode CONNECT (restriction au port HTTPS)

Cache de contenu Web

- En régime établi, le cache contient environ 200000 fichiers (renouvellement 15%/jour).
- Le nombre d'E/S générées sur le cache peut être très important.
- Il est donc plus que souhaitable d'utiliser un disque dédié au cache.
- La solution retenue pour obtenir de bonnes performances est de type RAID 0.
- Le cache est défragmenté toutes les nuits.

Cache de contenu Web

- Le cache n'utilise pas d'index spécifique.
- Les attributs RMS sont utilisés pour stocker les dates de création, et de consultation.
- Les noms de fichiers sont générés par *hashing* de l'URL.
- La recherche d'un document dans le cache est donc une simple recherche de fichier menée par XQP.

Cache et XQP

ACP_DIRCACHE parameter information:

- Old value was 10240, New value is 10240
- Hit percentage: 94%
- Attempt rate: 392 attempts per 10 sec.

ACP_DINDXCACHE parameter information:

- Old value was 2048, New value is 2048
- Hit percentage: 96%
- Attempt rate: 137 attempts per 10 sec.

Cache et XQP

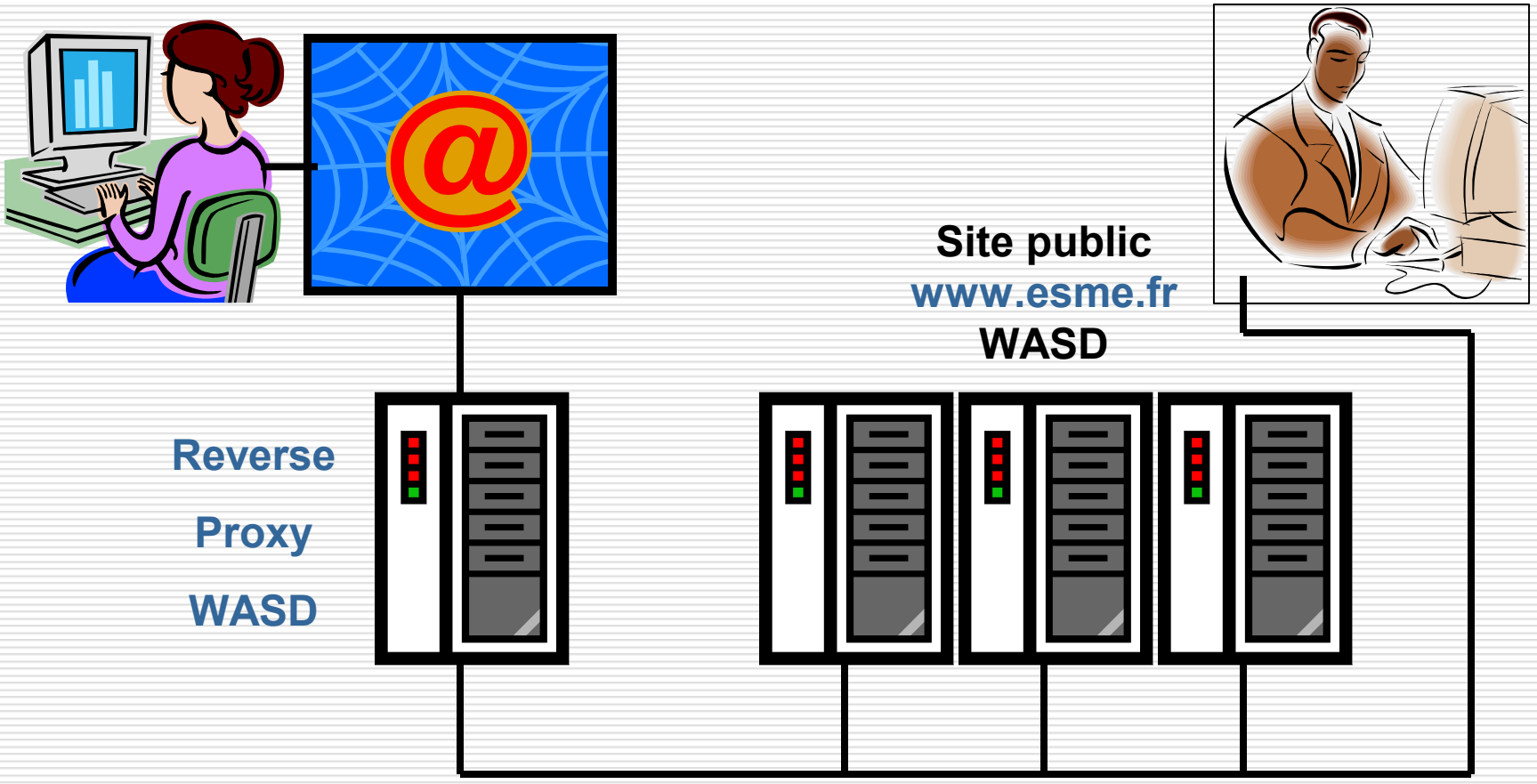
ACP_HDRCACHE parameter information:

- Old value was 16384, New value is 16384
- Hit percentage: 71%
- Attempt rate: 126 attempts per 10 sec.

ACP_MAPCACHE parameter information:

- Old value was 1024, New value is 1024
- Hit percentage: 97%
- Attempt rate: 13 attempts per 10 sec.

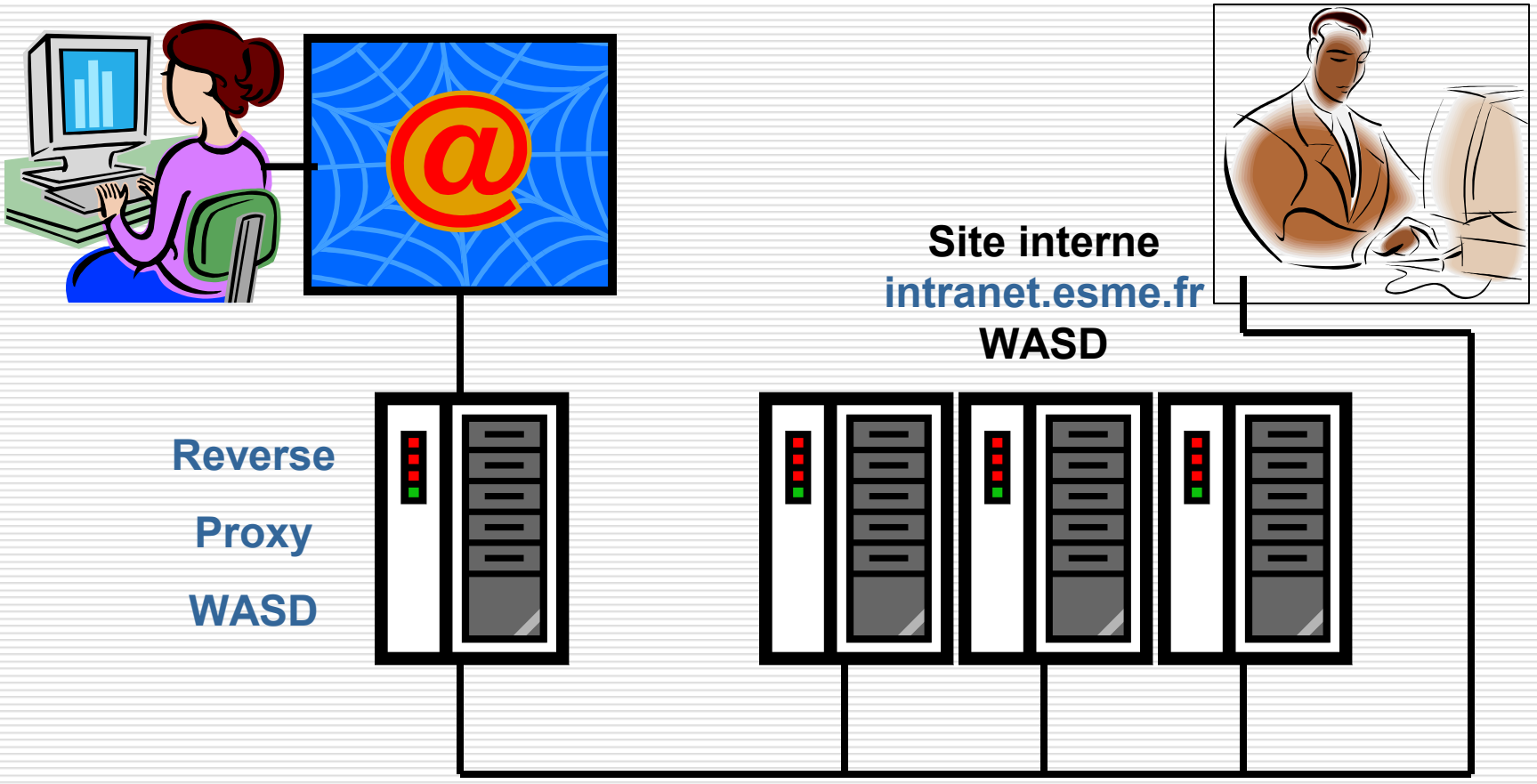
Reverse proxy HTTP



Reverse proxy HTTP

- 2 serveurs du cluster partagent la même configuration pour le site www.esme.fr
- Le reverse proxy assure le *failover* automatique d'un serveur vers l'autre.
- Les serveurs sont protégés du monde extérieur mais directement accessibles pour les utilisateurs internes.
- La répartition de charge est de type *DNS round robin*.

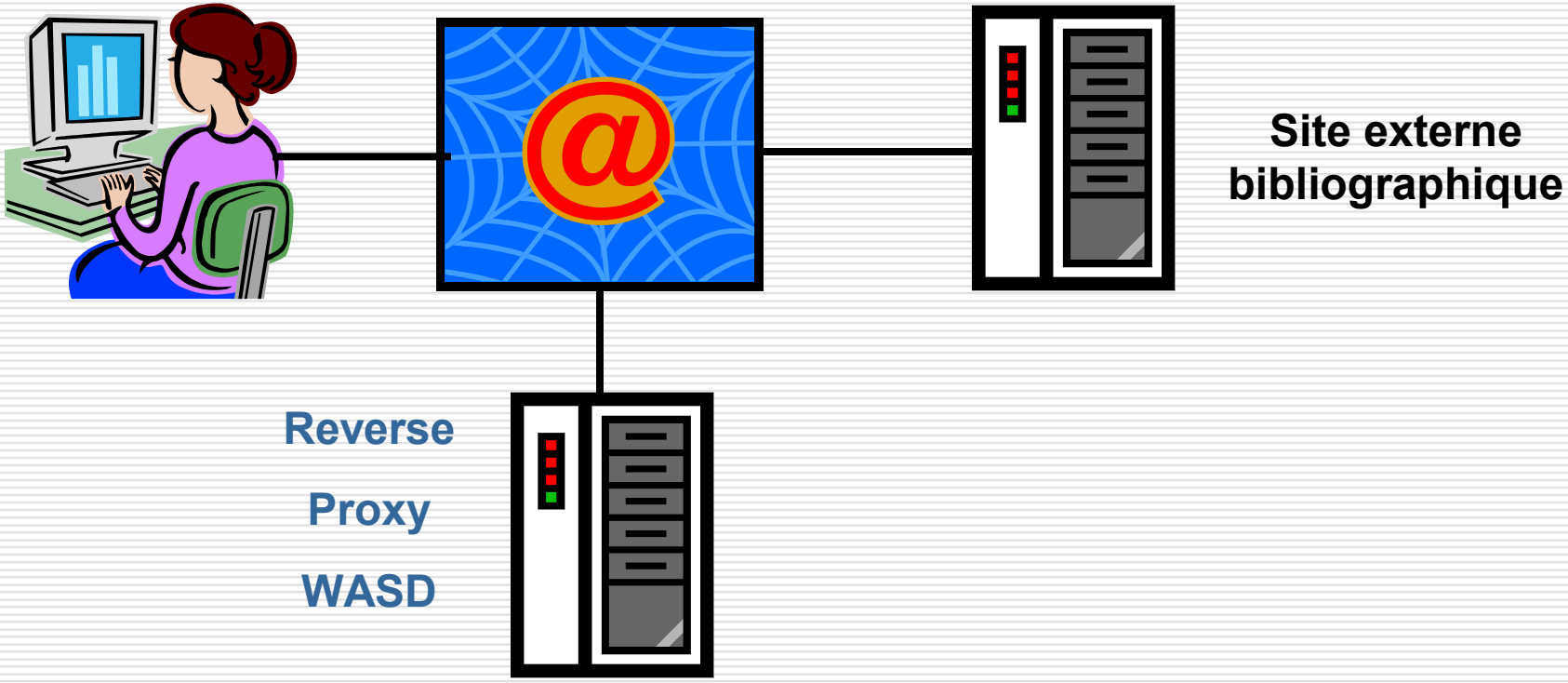
Reverse proxy HTTPS



Reverse proxy HTTPS

- Le *proxy* contrôle l'accès et assure le dialogue sécurisé avec les clients Internet.
- L'authentification est assurée par un module spécifique qui effectue une requête IMAP sur le cluster (le proxy n'appartient pas au cluster et n'a pas d'accès à l'UAF de celui-ci).
- Le *reverse proxy* assure le *failover* automatique sur les 3 serveurs Intranet.

DNS wildcard proxy



DNS wildcard proxy

- L'école est abonnée à un site encyclopédique qui autorise l'accès uniquement à partir de l'adresse IP du proxy.
- La fonction de *DNS wildcard proxy* permet d'entrer dans le DNS une équivalence du type:
  - *.esme.fr adresse externe du proxy

DNS wildcard proxy

- Le site www.site.fr est accédé au travers d'une URL du type www.site.fr.esme.fr qui nécessite bien sûr une authentification.
- La page peut être trouvée dans le cache de contenu web. Si ce n'est pas le cas, le proxy fait la requête vers le site d'origine.
- La réponse est retournée au client avec éventuellement compression GZIP.

Proxy transparent vers IIS

- Les serveurs Intranet servent de *proxy* transparents à des serveurs IIS.
- L'authentification est assurée par WASD à partir de l'UAF.
- Les applications IIS récupèrent le *username* (HTTP_REMOTE_USER).
- Le *reverse proxy* assure le filtrage contre les attaques IIS classiques.

WASD

Les fonctionnalités les plus appréciées

Scripts CGIplus

- WASD offre une extension très importante au concept de script CGI.
- Les scripts CGIplus sont des scripts CGI persistents qui restent en attente d'une nouvelle activation.
- Il n'y a donc pas de latence liée à:
  - l'activation de l'exécutable
  - l'attachement à une base de données

Scripts CGIplus

- Il est très facile d'écrire des scripts qui s'exécutent au choix en mode CGI ou CGIplus.
- Le choix du mode d'activation se fait par une règle de *mapping* EXEC ou EXEC+.
- Ceci peut aussi s'appliquer à des interpréteurs de scripts: PHP, Python, Java…

Scripts CGIplus

- void main (int argc, char *argv[])
- {
- CgiLibInit(argc, argv, script);
- }

- void script (void)
- {
- ... corps du script ici
- ... appelé à chaque requête
- }

Impersonnation

- Un script CGI ou CGIplus peut s'exécuter sous l'identité:
  - HTTP$NOBODY (compte par défaut)
  - d'un utilisateur spécifié par une règle de *mapping*
  - de l'utilisateur VMS authentifié par le dialogue navigateur/serveur

Modifications dynamiques

- Toute modification du fichier de *mapping* (HTTPD$MAP) peut être prise en compte dynamiquement par une simple commande:
  - HTTPD$EXE /DO=MAP
  - clic sur un bouton de l'interface web
- La prise en compte est immédiate sans autre effet visible par les utilisateurs.

Modifications dynamiques

- Toute modification du fichier de d'autorisation (HTTPD$AUTH) peut être prise en compte dynamiquement par une simple commande:
  - HTTPD$EXE /DO=AUTH=LOAD
  - clic sur un bouton de l'interface web
- La prise en compte est immédiate mais implique une réauthentification de tous les clients.

WATCH

- WASD possède un outil sans égal permettant de suivre en temps réel toute activité du serveur.
- La totalité du code est instrumentée.
- Démonstration…

Questions