Architectures Réseau

Architecture d'un réseau

- Vous avez travaillé avec l'infrastructure du cours depuis quelque jours, et êtes un peu plus comfortables avec celle-ci
- Vous avez probablement la responsabilité pour tout ou partie du réseau de votre registre, et avez de l'expérience dans la gestion d'un réseau – mais peut-être n'avez vous jamais eu l'occasion de le concevoir correctement depuis le départ, ou alors le réseau a grandi autour de vous.

Architecture d'un réseau (2)

- Dans cette présentation nous allons donner les grandes lignes pour concevoir un réseau qui soit plus robuste et fiable
- Nous parlerons des couches 2-7
- On intégrera les Architectures de Registres, en indiquant, en indiquant ou chaque morceau sera placé
- Nous espérons que ceci vous fera réfléchir d'une manière neuve sur comment structurer le réseau de votre registre

Aperçu

- Couche 1 – Physique
  - Commutateur, Câblage
- Couche 2 – Lien
  - VLANs, ARP, NDP
- Couche 3 - Réseau
  - IP, IPv6, IPsec, certains protos de routage
- Couche 4 - Transport
  - TCP, UDP
- Couche 7 - Application
  - DNS, WHOIS, HTTP, SMTP, DB, ...

Règles de base

- Utiliser des équipements réseau actifs et gérables
  - Les commutateurs et répéteurs qui ne peuvent être gérés activement rendent la résolution d'un problème réseau plus difficle – ils sont effectivement transparents.
- Installer un serveur syslog pour vos équipements réseau
  - L'étape no.1 pour identifier une panne réseau est de regarder dans les journaux

Règles de base (2)

- Pratiquer la séparation des services
  - Il peut-être tentant de tout faire tourner sur une seule machine, mais cela rend difficile la sécurisation et le grandissement de vos services
  - Pas besoin d'investir dans de nouvelles machines immédiatement – la virtualisation comme par exemple les vservers sous Linux et les jails sous FreeBSD rendent les choses aisées

Règles de base (3)

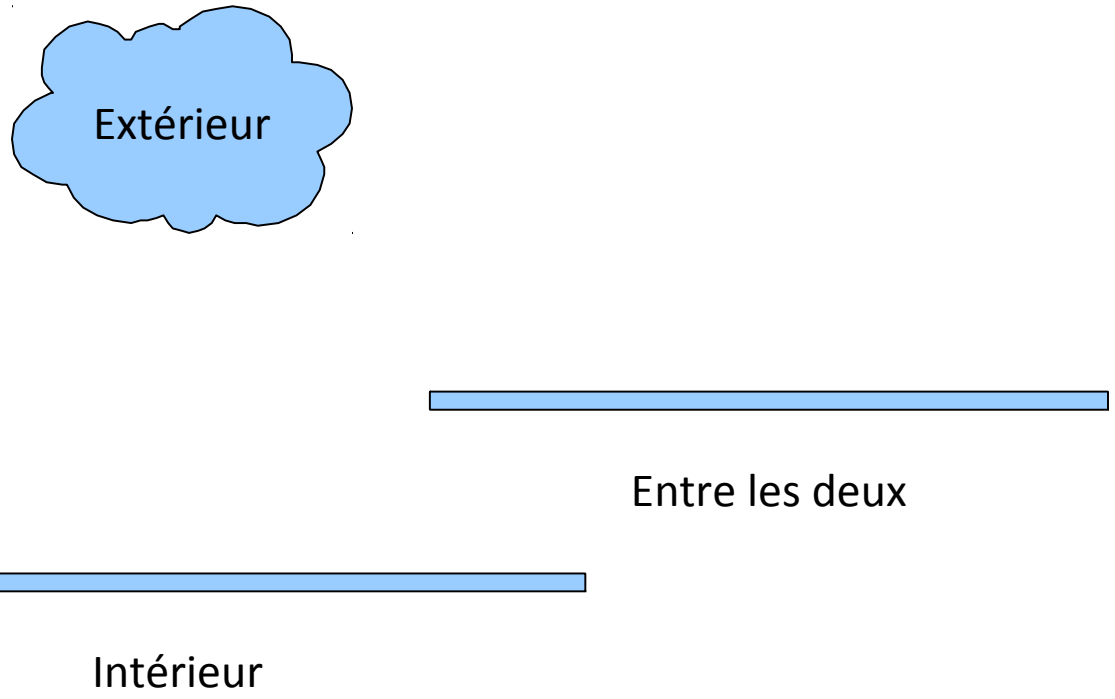
- Sécurité d'un réseau
  - Le filtrage se passe aux couches 3 et 4
  - Commencer par tout bloquer, et n'ouvrir que pour ce qui est strictement nécessaire
  - Bien plus simple que de tenter de sécuriser un réseau en production à priori
  - Il faut connaître les protocoles que vous filtrez...
    - Oui, le DNS utilise TCP...
    - Quel port source déjà ?

Topologie du réseau

- Dans la phase de conception du réseau, penser en termes de niveaux de privilège:
  - Qu'est-ce qui est accessible par tout le monde ?
  - Qu'est-ce qui doit être limité à quelques machines ou quelques personnes ?
  - Qu'est-ce qui est strictement interne et ne doit jamais être accessible de l'extérieur ?
- Cette classification n'est bien sûr pas hermétique, et des exceptions devront être faites pour communiquer

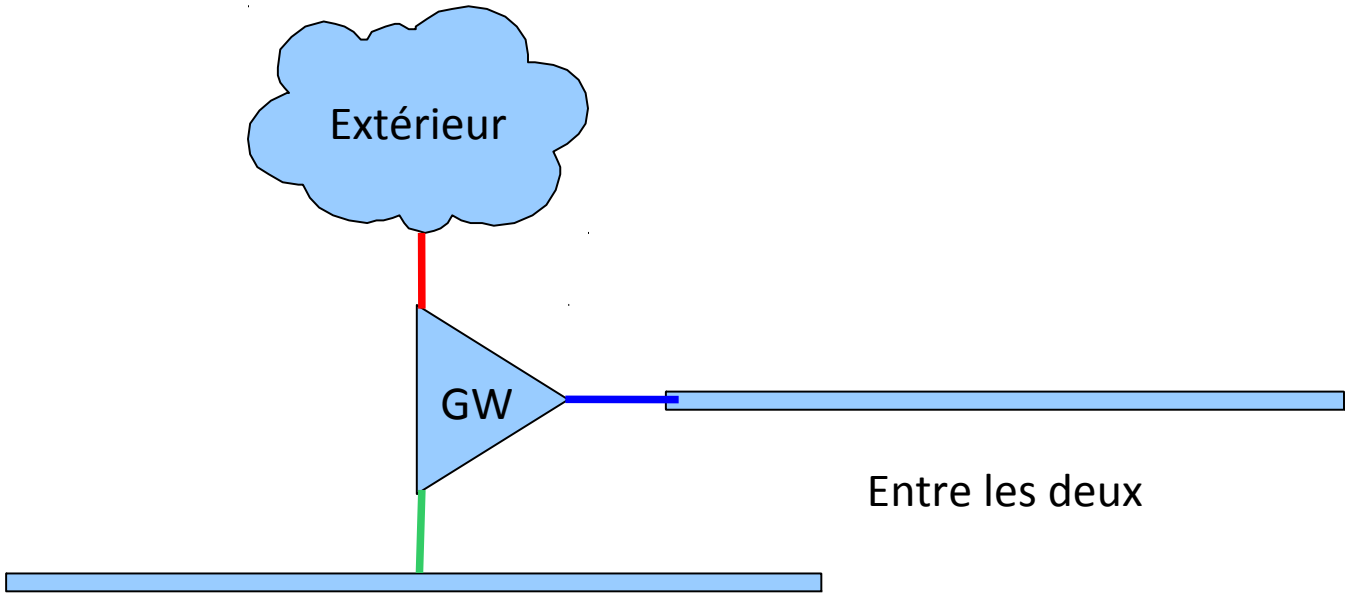
Topologie du réseau (2)

- En appliquant les principes ci-dessus, on arrive à la simplification suivante:



Topologie du réseau (3)

- Il va bien évidemment raccorder ces différentes parties du réseau ensemble



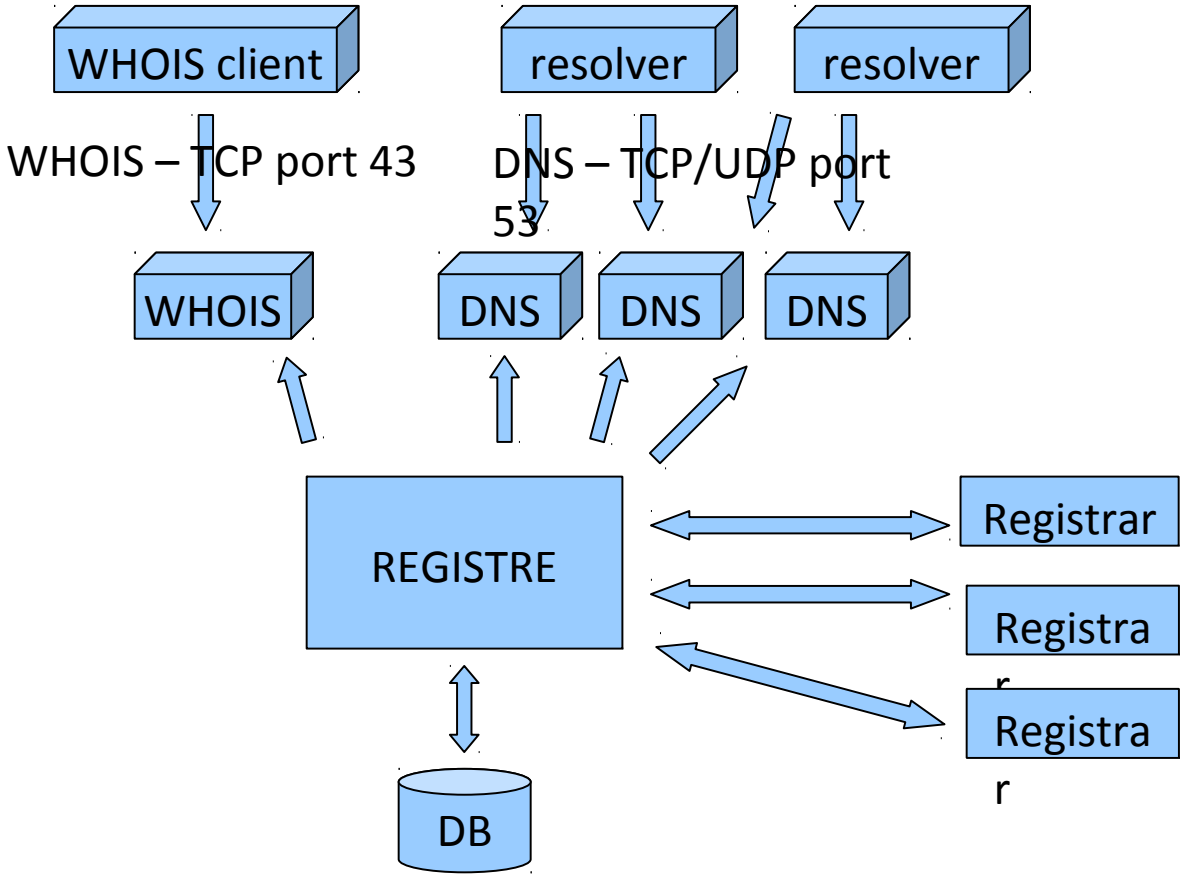
GW est le point de contrôle où l'on peut appliquer la sécurité et contrôler le trafic

Topologie du réseau (4)

- La passerelle (GW) peut-être:
  - Un routeur avec des listes d'accès
  - Un firewall dédié
  - Une machine configurée spécialement (FreeBSD, Linux)
- Utiliser un filtre à paquets pour contrôler:
  - Ce qui est permis (protocole)
  - Depuis qui/où (adresse et port source)
  - Dans quelle direction (interface, machine)

Et notre registre dans ceci ?

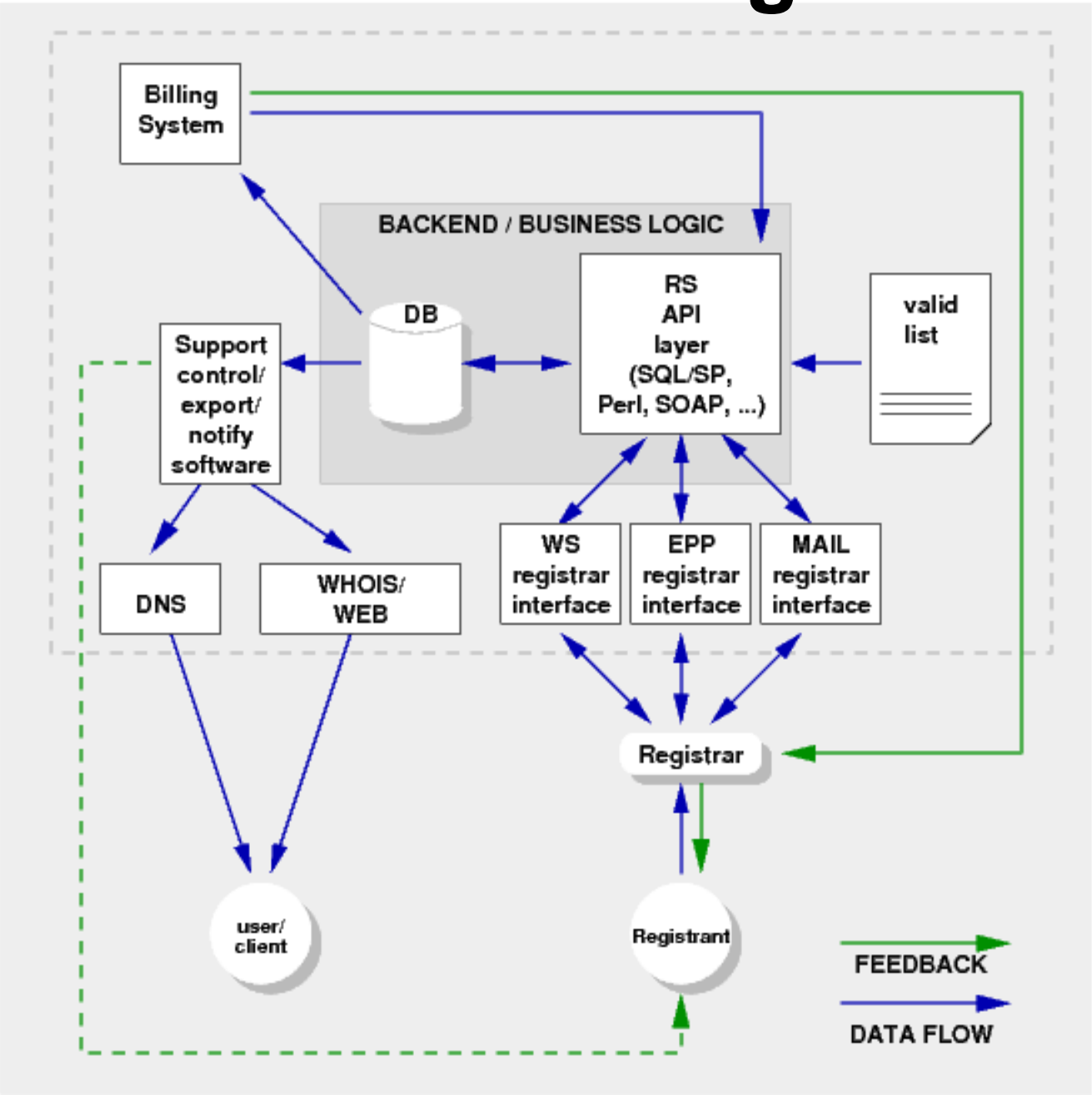
- Un peu plus tôt nous avons parlé des architectures de registre:



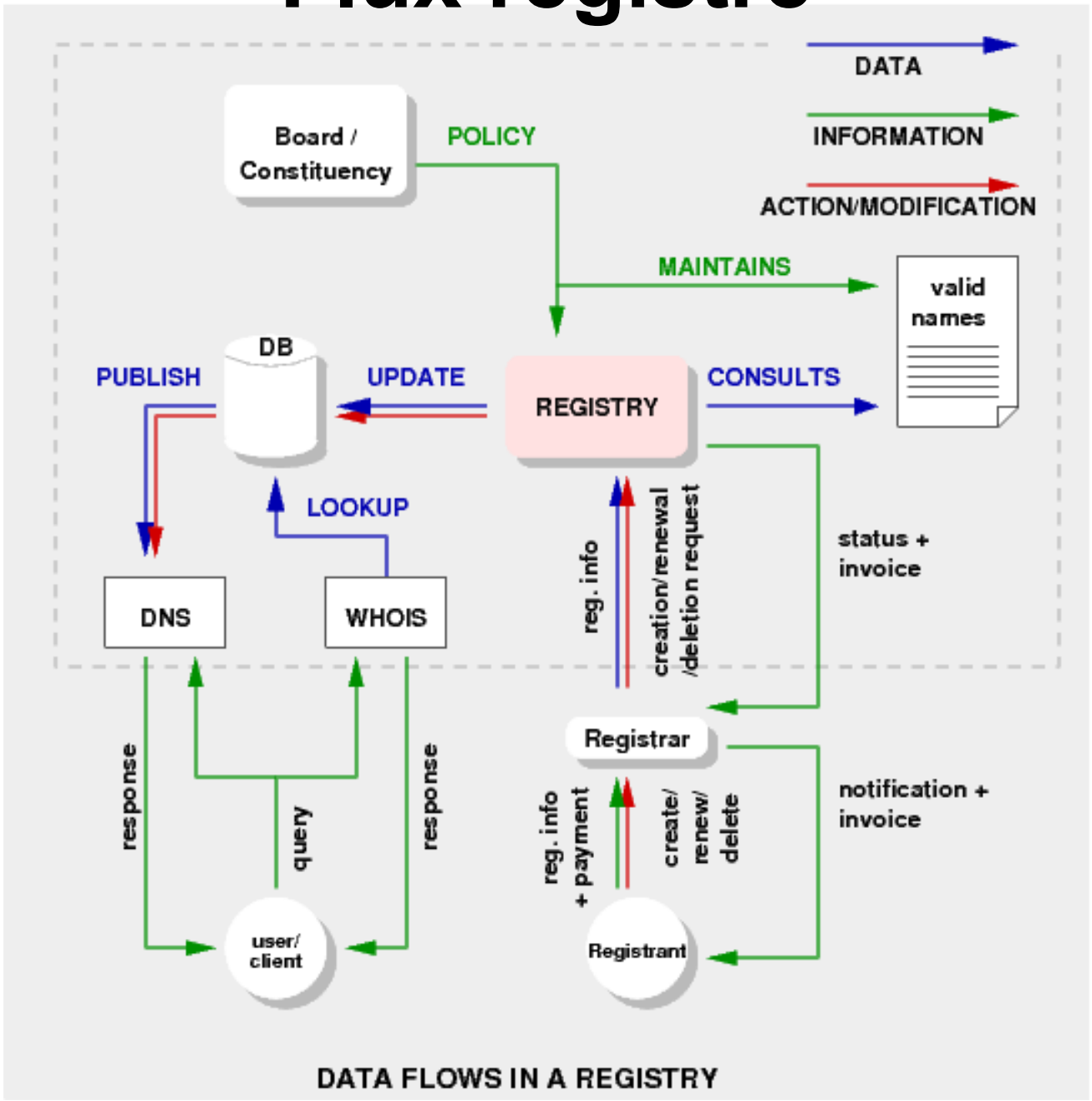
Comment assembler le tout

- Ceci est une représentation logique des composants d'un registre
- Il ne va pas en détail pour illustrer les fonctions et les flux entres les différents composants
- Voici un schéma un peu plus détaillé...

Architecture registre



Flux registre



Représentation du réseau

- En regardant les diagrammes, on a une meilleure idée de l'interaction
- Essayez de produire un schéma similaire pour voitre registre, en utilisant un modèle ”de boites”
- Certains des flux et actions peuvent facilement être converties à des listes d'accès ou des filtres:
  - Les requêtes DNS sont souvent des questions, et les serveurs esclaves ne doivent pas autoriser les mises à jour!

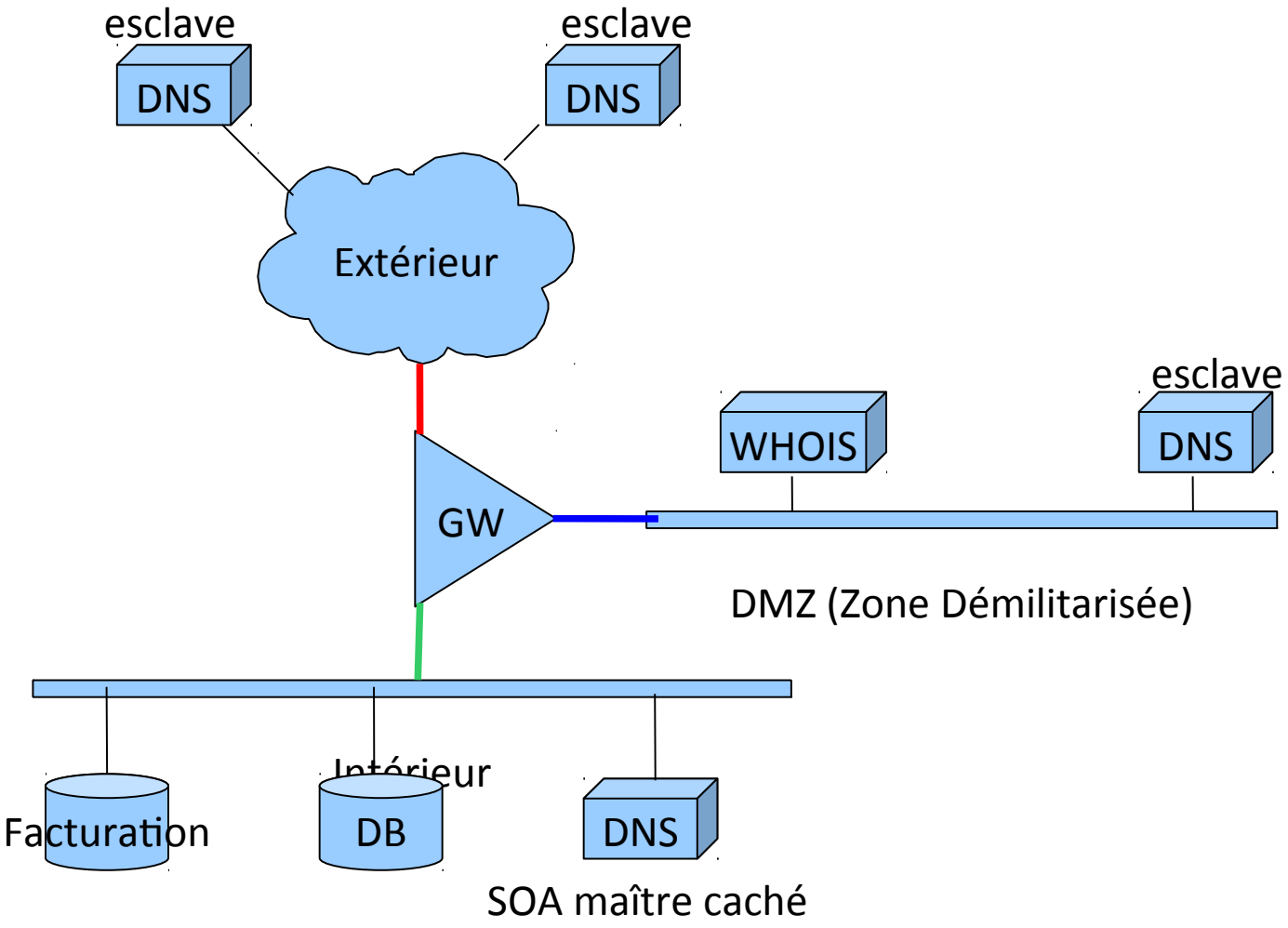
Représentation du réseau (2)

- Les autres flux peuvent être plus difficiles à isoler, car les protocoles qui les transportent peuvent encapsuler *beaucoup* d'actions différentes:
  - HTTP, *SQL, SSH
- Un vendeur de firewall peut vous vendre des systèmes de protection applicatifs ou en profondeur – ceux-ci peuvent aider dans certains cas (injection SQL, ...)
  - Mais vous connaissez vos applications mieux que le vendeur!

Représentation du réseau (3)

- Une fois que vous avez défini un modèle ”en boites” pour votre registre, et vous connaissez les flux et actions qui y prennt place, alors on peut passer à la phase de conception du réseau – dans cet ordre!
- Pensez au différent composants (boites) et les flux qui les relient en terme de niceaux de privilège (comme décrit précédemment)
- Essayons d'assemble tout ceci

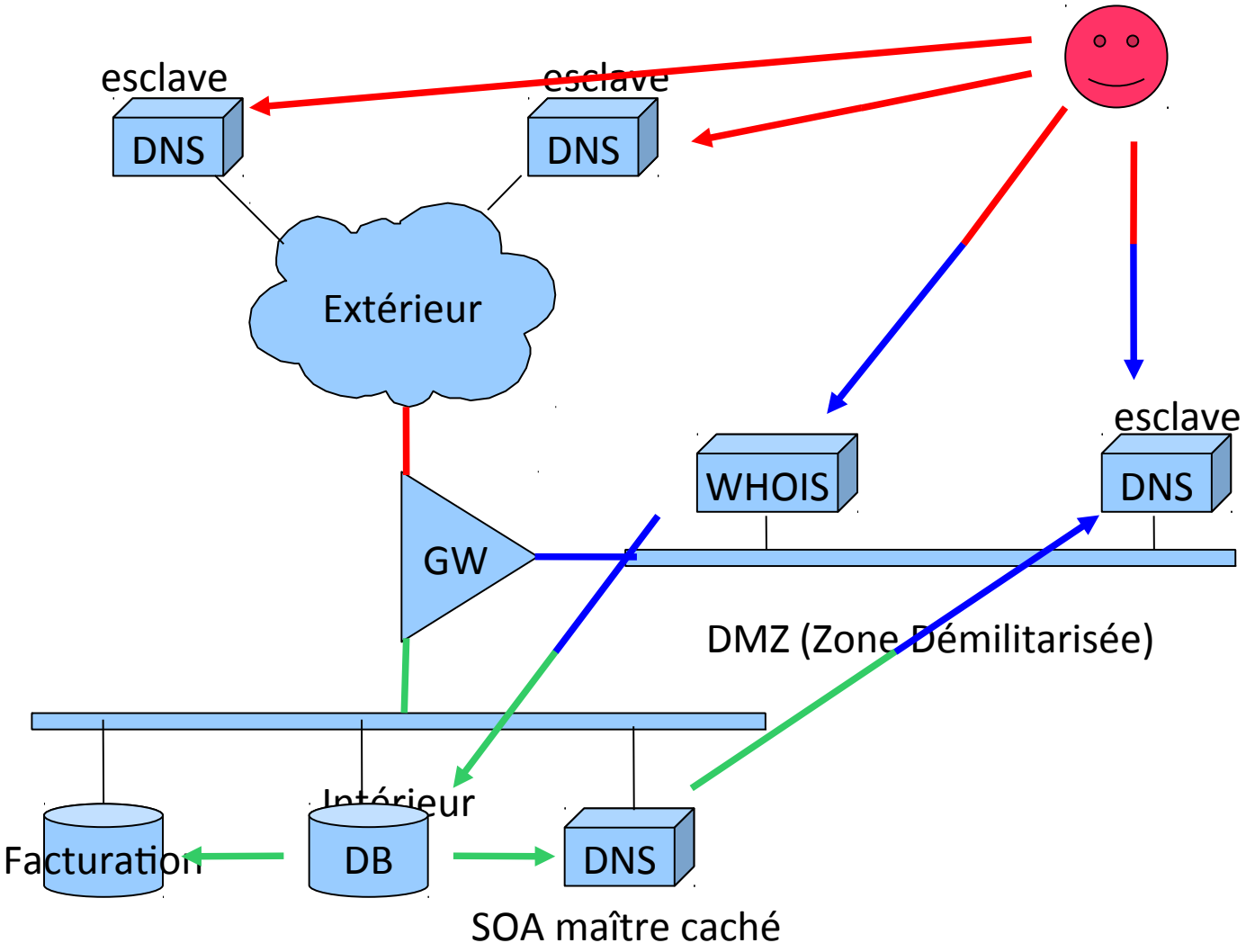
Représentation du réseau (4)



Représentation du réseau (5)

- Ceci est un réseau de registre assez simple
- Néanmoins, les composant sensibles sont placés à l'intérieur, derrière le firewall
- Ceci est toujours une vue logique – en pratique cette architecture peut s'implémenter avec des VLANs

Flux/actions



Aspect sécuritaires

- Si vous observez les diagrammes précédent, vous verrez qu'aucun flux provenant de l'Internet accède directement au réseau interne.
- Les requêtes pour des informations publiques se font soit aux serveurs DNS accessibles publiquement, ou semi-publiques (comme par exemple le serveur WHOIS sur la DMZ)

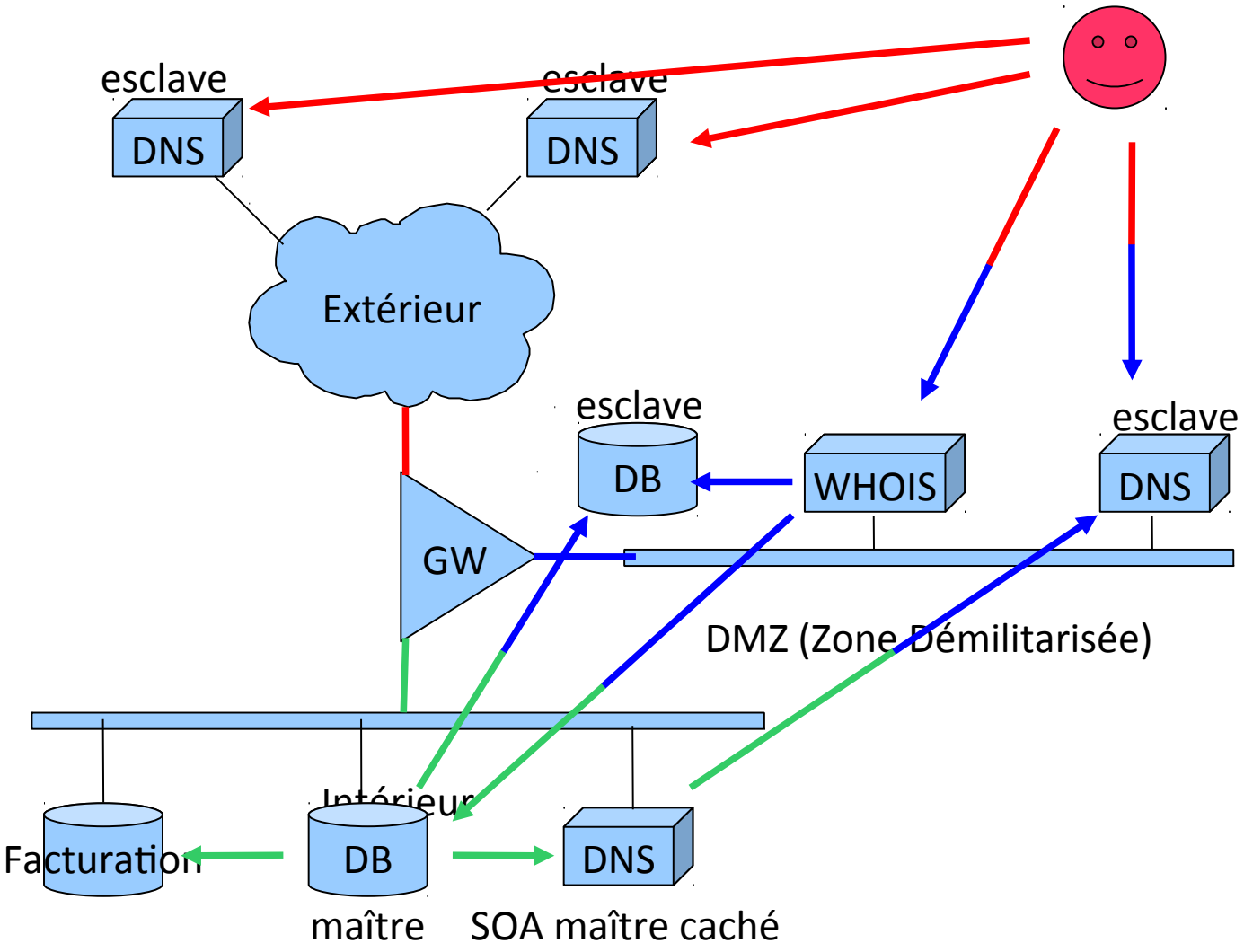
Aspect sécuritaires (2)

- Pourquoi ne pas simplement placer le serveur WHOIS à l'extérieur ?
  - Le serveur WHOIS a une connexion privilégiée avec la BD, où résident les données de zones et les donnés admin.
  - C'est un point sensible – l'accès à la BD depuis un serveur WHOIS doit se faire via un utilisateur en lecture seule sur les tables concernées.
  - Ceci limit les dégât en cas d'attaque type injection SQL

Servons-nous de l'architecture

- Peut-on améliorer ce modèle ?
- Et si quelqu'un cassait la sécurité de la BD ? Alors ils auraient accès à TOUT le réseau interne!
- C'est ici que la conception du réseau et la séparation des services devient utile.
- Pourquoi ne pas placer une copie synchrone de la base de données (BD) à côté du serveur WHOIS, et en lecture seule ?

Servons-nous de l'architecture (2)



Servons-nous de l'architecture (3)

- Nous avons utilisé l'architecture pour supprimer un problème de sécurité potentiellement grave
- Si le firewall est configuré correctment, alors il y a très peu de chances que quiconque exploite une connexion entrante vers les réseau interne pour monter une attaque.
- Utiliser des ACLs / filtres pour limite l'accès au login à vos utilisateurs et seulement eux!

Choix de l'emplacement

- Tous ces composants n'ont pas besoin de se trouver sur le même site
  - Les serveurs DNS et WHOIS pourraient se situer dans deux centres différents
  - Les esclaves DOIVENT être hors-site!
  - Pour des raison de sécurité et de performance, il vaut mieux placer les services qui communiquent beaucoup entre eux proche l'un de l'autre
    - Par exemple le serveur WHOIS et la BD

Questions

?