Firewalls et autres éléments d'architecture de sécurité

ORGANISATION INTERNATIONALE DE la francophonie

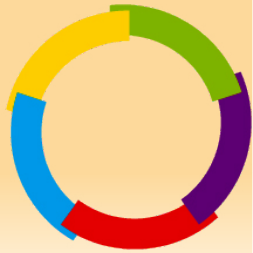
cedric.foll@(education.gouv.fr|laposte.net)
Ministère de l'éducation nationale

Atelier sécurité
Rabat – RALL 2007

Infrastructure de sécurité

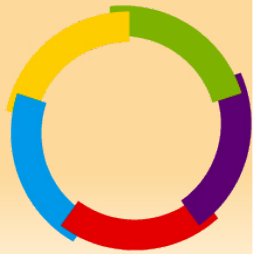
« If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology. »

Bruce Schneier



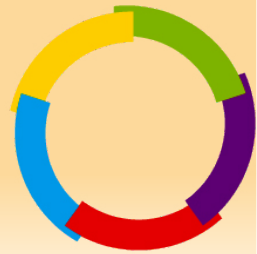
Plan

- Les firewalls
  - Netfilter
- Les IDS
  - HIPS, NIDS, IPS
- Les proxy
- Les reverses proxy
- Architecture réseau



Les Firewalls

- A quoi ça sert ?
  - Élément permettant de trier parmi les flux réseaux en bloquant certains et autorisant d'autres.
- Différents types de Firewalls:
  - Firewalls personnels
    - Protège la machine sur laquelle il est installé.
  - Firewalls (tout court)
    - Effectue du routage inter-zones tout en appliquant des règles de filtrage.
    - En général se place en coupure entre le réseau de l'entreprise et Internet.

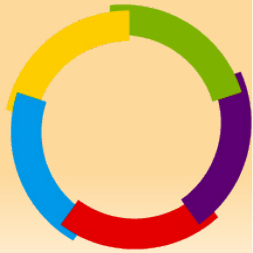


Qu'est ce que filtre un firewall ?

- Un firewall est un équipement de couche 4.
  - Il laisse passer ce qui est explicitement autorisé, se basant sur:
    - Les informations de couche 3
      - IP source et IP destination, protocole de couche 4 (ie icmp, udp, tcp, ...)
    - Les informations de couche 4
      - port source, port destination, message icmp, ...
  - Il filtre tout le reste.

Exemple de règle

- Considérons un site web d'adresse 10.0.0.1 auquel nous voulons autoriser l'accès:
  - « autoriser le trafic venant de n'importe où vers 10.0.0.1 en tcp sur le port 80 »
  - « rejeter tout le reste »

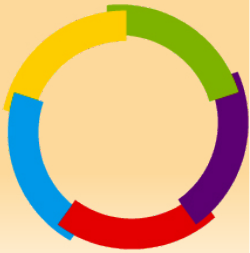


Notions de statefull/stateless

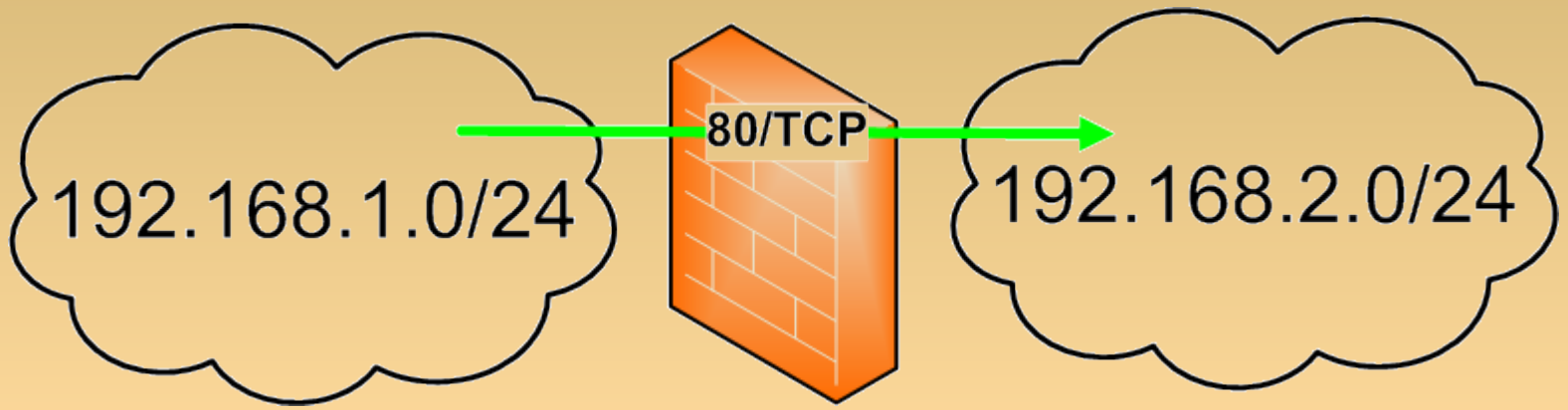
- Un firewall stateless ne sait pas si un paquet appartient à une connexion déjà établie.
  - Pour un flux TCP, la « solution » consiste à analyser les champs TCP SYN et ACK.
  - Pour chaque flux autorisé il faut explicitement autoriser les paquets entrant et sortant.
  - Incomplet en terme de sécurité, en particulier ne sait pas gérer les flux UDP, ICMP et complexe à maintenir.
- Un firewall statefull connaît l'état de chaque connexion.
  - Plus sécurisé et plus simple à gérer.
  - Mais demande plus de ressource CPU et de mémoire.

Exemples

- Firewall stateless:
  - les routeurs d'entrée de gamme voire certains firewalls « tout en un » destinés aux particuliers
  - ipchains (firewall des linux 2.2.X).
- Firewall statefull:
  - netfilter (firewall des linux 2.4 et suivant).
  - tous les firewalls modernes.



FireWall: IPChains (Linux 2.2)



#On autorise tous les flux de 192.168.1.0/24 vers 192.168.2.0/24 en 80 TCP

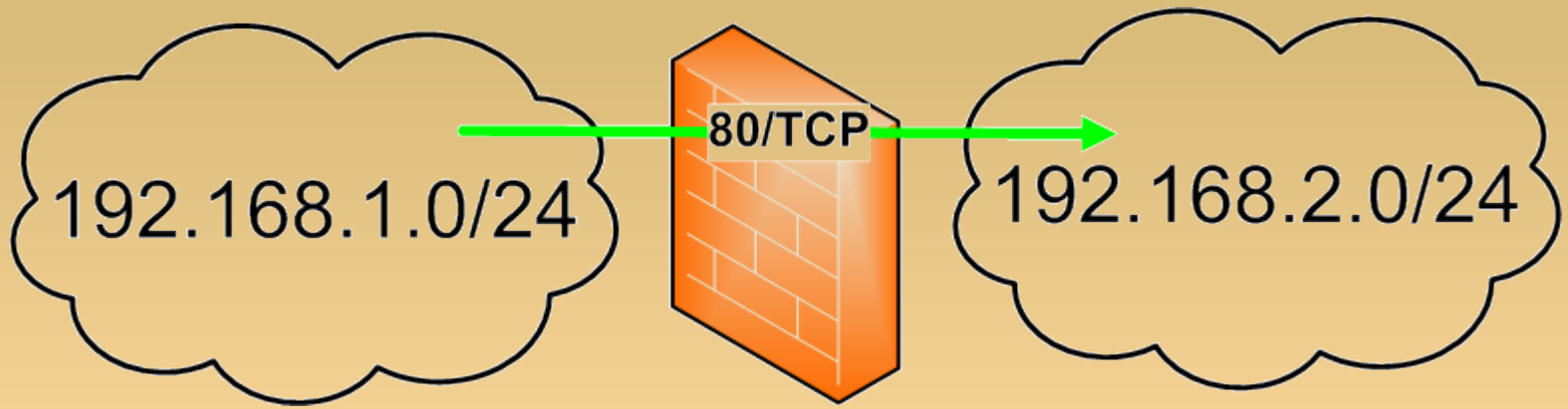
ipchains -A forward -p TCP
-s 192.168.1.0/24 1024: -d 192.168.2.0/24 80 -j ACCEPT

#On autorise tous les flux de 192.168.2.0/24 vers 192.168.1.0/24 sauf les SYN (-y)

ipchains -A forward -p TCP
-s 192.168.2.0/24 80 -d 192.168.1.0/24 :1024 -j !-y ACCEPT

- Tous les paquets venant de 192.168.2.0 avec comme port source 80 et sans SYN sont autorisés.
- Pour chaque règle autorisant un flux il faut écrire une règle autorisant la réponse.

Firewall: NetFilter (Linux 2.4, 2.6)



#On autorise tous les paquets appartenant à une connexion déjà établie:

iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

#On autorise tous les flux de 192.168.1.0/24 vers 192.168.2.0/24 en 80 TCP

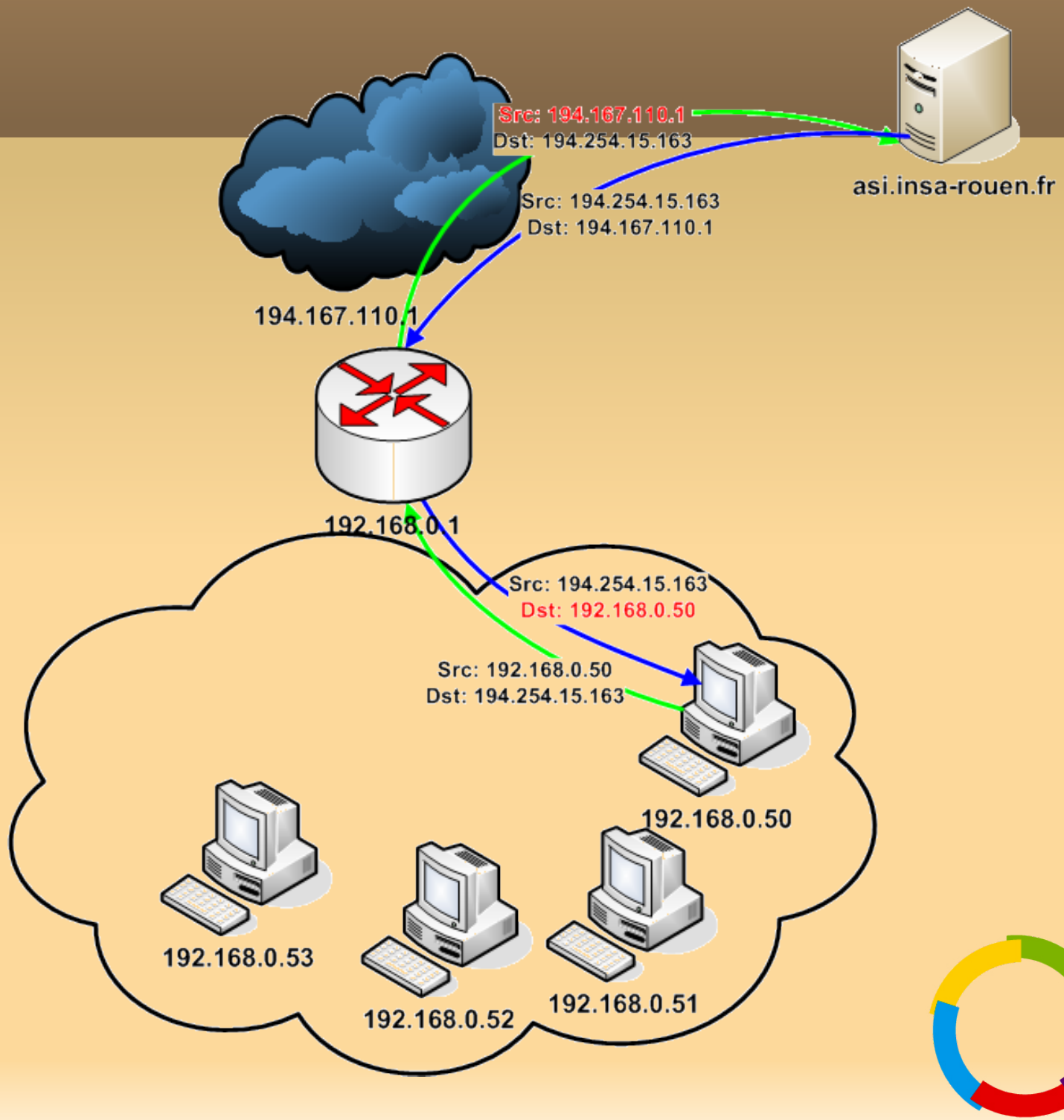
iptables -A FORWARD -p tcp -m tcp -m state --state NEW -s 192.168.1.0/24 -d 192.168.2.0/24 --dport 80 -j ACCEPT

- Seuls les paquets appartenant à une connexion établie sont autorisés.

Network address translation

- Problèmes:
  - Il n'y a plus assez d'adresses IP disponibles pour le nombre de machines reliées à internet sur la planète.
  - Les adresses IP coûtent cher.
  - On ne voudrait pas que l'on puisse accéder à tout le réseau interne depuis internet.
- Solution, le NAT:
  - Le réseau interne est en adressage privé (RFC1918), un routeur/firewall translate les adresses des connexions sortantes.

NAT


Src: 194.167.110.1
Dst: 194.254.15.163
asi.insa-rouen.fr
Src: 194.254.15.163
Dst: 194.167.110.1
194.167.110.1
192.168.0.1
Src: 194.254.15.163
Dst: 192.168.0.50
Src: 192.168.0.50
Dst: 194.254.15.163
192.168.0.50
192.168.0.53
192.168.0.52
192.168.0.51

NAT

- Inconvénients:
  - Certains protocoles fonctionnent mal ou pas avec de la translation d'adresse (H323, SIP, ...)
  - Il est impossible d'initier une connexion vers une machine NATée depuis l'extérieur
- Avantages:
  - On économise le nombre d'adresses IP publiques.
  - Il est impossible d'initier une connexion vers une machine NATée depuis l'extérieur

Netfilter

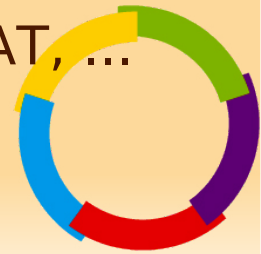
- Apparu dans le noyau 2.4
  - Firewall statefull.
  - Fonctions de translation d'adresse très évoluées.
  - Très répandu et concurrence sans problème des firewalls commerciaux (PIX par exemple de cisco).
  - Il existe une excellente interface graphique, fwbuilder.

Netfilter en pratique

- iptables: commande permettant d'interagir avec netfilter.
  - iptable-save: affiche la configuration du firewall
  - iptables ajoute/modifie/supprime des règles
- Dans la pratique on utilise souvent une GUI pour gérer la configuration des règles:
  - fwbuilder: http://www.fwbuilder.org/

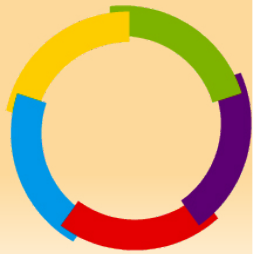
Netfilter: interface iptables

- iptables -A CHAINE REGLES -j ACTIONS
  - CHAINES
    - INPUT (trafic à destination du fw)
    - OUTPUT (trafic émis par le firewall)
    - FORWARD (trafic routé)
    - Pour les translations d'adresse:
      - -t nat POSTROUTING|PREROUTING
  - REGLES
    - Comment est caractérisé le paquet (ip src/dst, port src/dst, proto, interface in/out) ? Appartient-il à la connexion intialisée ou pas ?
  - ACTIONS
    - ACCEPT|DROP
    - Manipulation de paquet (pour le nat): SNAT, DNAT, ...



Netfilter: interface iptables

- Police par défaut, option -P
  - ex: iptables -P FORWARD DROP
- Effacer toutes les règles d'une chaîne, option -F
  - ex -F iptables -F FORWARD
- Effacer une règle donnée -D
  - iptables -D num règle
  - iptables -D toute la règle
- Lister les règles
  - iptables-save

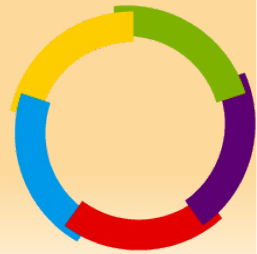


netfilter: cookbook

- Police par défaut sur le forward DROP
  - iptables -P FORWARD DROP
- Accepter toutes les connexions actives
  - iptables -A FORWARD -m state -state ESTABLISHED,RELATED -j ACCEPT
- Accepter les ssh de pc1 vers pc2
  - iptables -A FORWARD -m state -state NEW -s pc1 -d pc2 -p tcp -dport 22 -j ACCEPT

netfilter: cookbook

- Masquerade (pour partager une IP publique sur eth0)
  - iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
- SNAT (comme le précédent mais avec une autre IP)
  - Altérer l'ip source du trafic routé (et/ou sortant)
    - iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 192.168.0.1

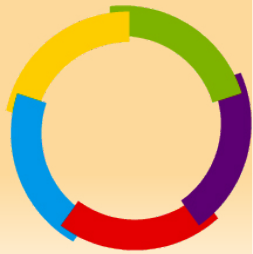


netfilter: cookbook

- REDIRECT (redirige un flux en local), utile pour faire du proxy transparent ou des attaques man in the middle.
    - iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080
- DNAT (transformation de l'IP destination), utile pour Man In The Middle:
    - iptables -t nat -A PREROUTING -p tcp -d www.google.com --dport 80 -j DNAT --to-destination www.faux-google.com

Les IDS

- Types d'IDS
  - Les NIDS
  - Les HIDS
    - Analyse de logs
    - Empreinte
- Ce qu'ils détectent et ce qu'ils ne détectent pas.



Les NIDS

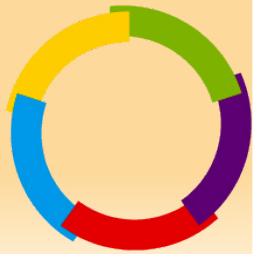
- Les NIDS (Network Intrusion Detection System).
  - Analyse les flux réseau et recherche des signatures d'attaques
    - Ex: /etc/passwd dans une urls, User-Agent: nikto,...
    - La plupart des NIDS fonctionnent ainsi, dont snort.
  - Vérification du respect de la conformité des protocoles aux RFC
    - Ex: paramètre de GET dans une requête HTTP possède moins de 1024 bytes
    - Ne nombreux IDS fonctionnent ainsi.
  - Analyse statistique (ou analyse comportementale)
    - Mesure de la déviation entre le trafic réseau habituel et le trafic à un instant T.

Les NIDS

- Qu'est ce qu'ils vont détecter ?
  - Toutes les attaques venant des outils de tests automatiques:
    - Nessus, nikto, whisker, nmap, ...
  - Certaines attaques visibles
    - En particulier certaines failles webs génériques
      - xss
    - L'exploitation de failles sur des applis web connues (phpbb, webcalendar, phpnuke, ...)
    - L'exploitation de faille touchant un logiciel connu (apache, IIS, Cisco, ...)

Les NIDS

- Qu'est ce qu'ils ne vont pas détecter ?
  - Les attaques webs faites à la main.
    - Les SQL injections, Remote include PHP
    - Les forces brutes sur des applis webs (sauf celles réalisant une analyse statistique)
  - La plupart des forces brutes (POP3, FTP, telnet, ...).
  - Les flux chiffrés
    - HTTPS
    - SSH
    - IPSec



Les NIDS

- Chez les logiciels libres:
  - Snort, très actif et efficace. Fonctionne par signature d'attaque.
  - BRO, vérification de conformité protocolaire

Les NIDS: Pourquoi un relatif insuccès ?

- Les réseaux deviennent trop gros
  - « Impossible » d'analyser un lien Giga.
- Les attaques sont trop fréquentes
  - On ne va pas réagir à chaque attaque.
  - Seules les attaques les plus visibles donnent lieu à un signalement (en gros l'utilisation de softs de scan)
- Les flux chiffrés (tels que HTTPS) ne peuvent être analysés.
- L'exploitation demande des personnes qualifiées (et c'est un travail ingrat).

Les HIDS

- Deux grandes familles:
  - Logiciels fonctionnant par scellement de fichiers (ex: tripewire, samhain)
  - Logiciels analysant les logs et les remontant
    - Détecte les forces brutes et les crash/redémarrage de processus.

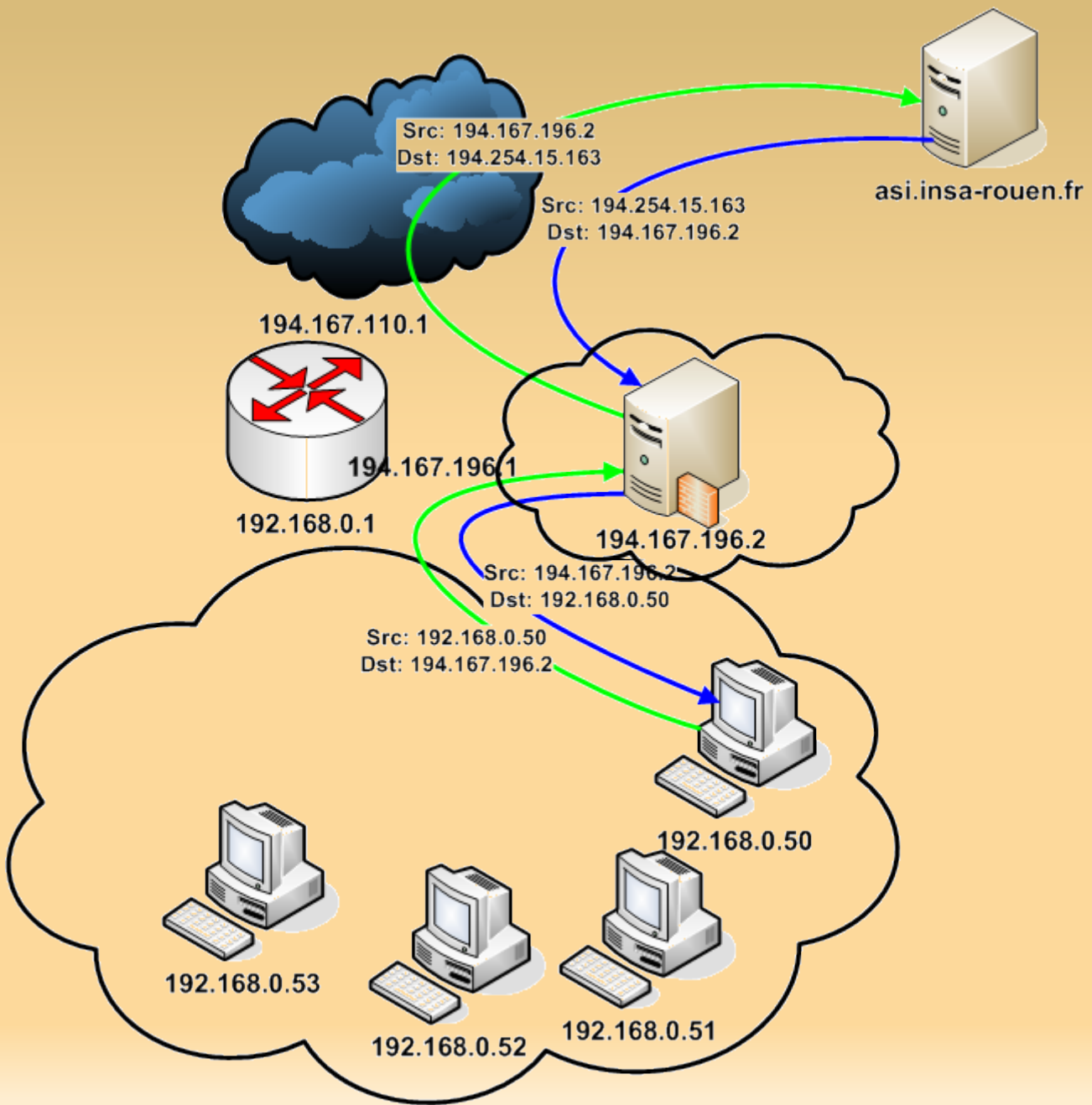
Les IPS

- Intrusion Prevention System
  - Grand succès marketing
    - Tous les firewalls propriétaires sont aujourd'hui des IPS
  - En fait un firewall qui réalise de l'analyse de couche 7 (comme un IDS) et qui bloque sur la base de signatures d'attaques ou de conformité protocolaire
    - Solution libre: snort-inline
      - Une version de snort compilée pour intéragir avec netfilter

Les Proxys

- Pour sortir sur internet, le client passe par un serveur mandataire
- Avantages:
  - Possibilité d'analyser plus finement le trafic.
  - Des journaux de connexions.
  - Sur les flux HTTP, possibilité de filtrer certains sites, de faire du contrôle anti-virus.
- Inconvénients:
  - Moins performant en terme de débit.
  - Ne fonctionne que pour quelques protocoles.

Les Proxys

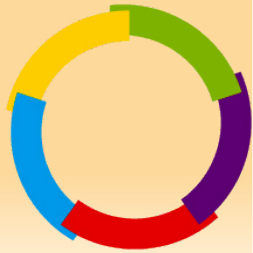

Src: 194.167.196.2
Dst: 194.254.15.163
asi.insa-rouen.fr
Src: 194.254.15.163
Dst: 194.167.196.2
194.167.110.1
194.167.196.1
192.168.0.1
194.167.196.2
Src: 194.167.196.2
Dst: 192.168.0.50
Src: 192.168.0.50
Dst: 194.167.196.2
192.168.0.50
192.168.0.53
192.168.0.52
192.168.0.51

Proxy

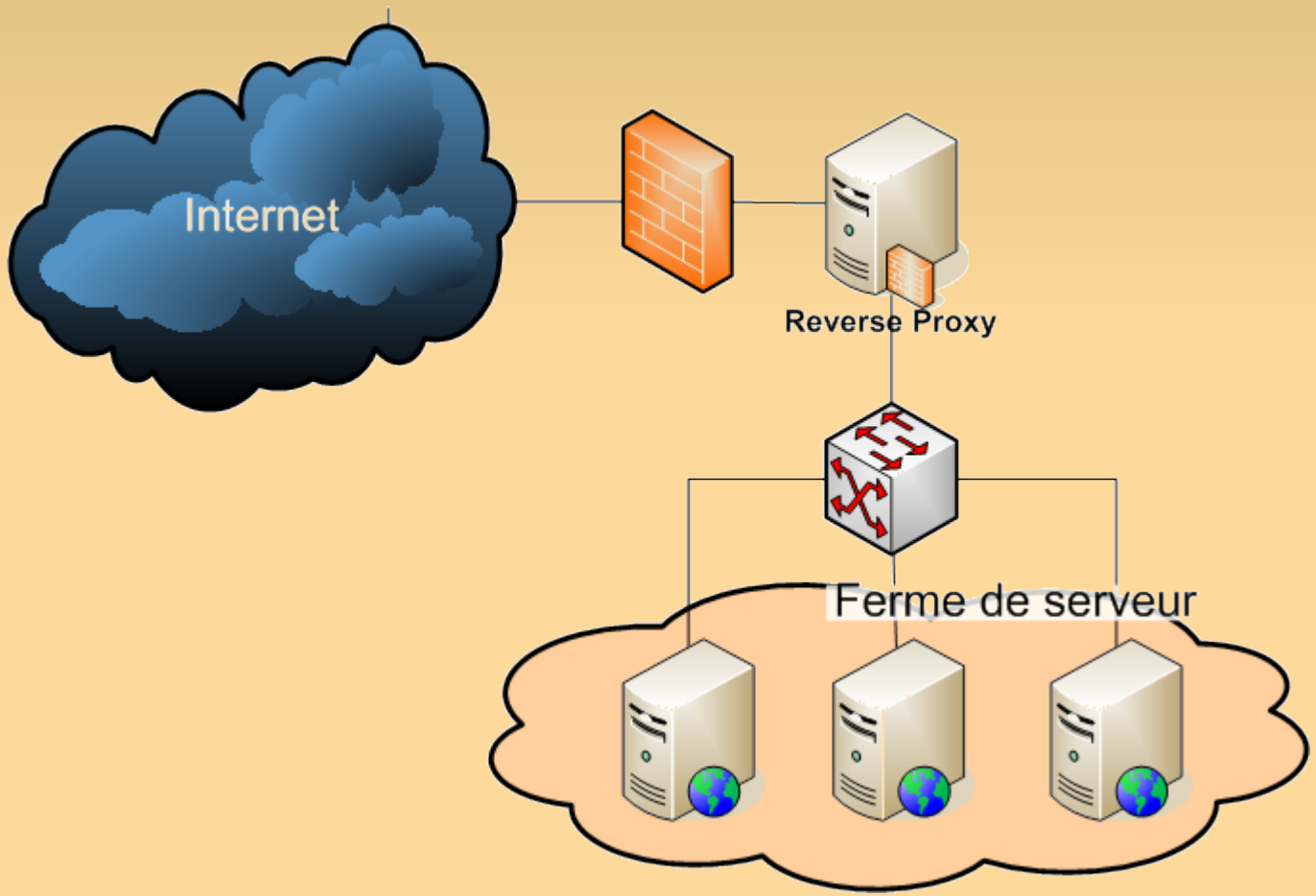
- Solutions libres
  - Squid, très largement utilisé
  - Pour le filtrage
    - Dansguardian
      - Filtrage par listes noires
      - Filtres de contenu (par mots clefs)
      - Filtrage anti-virus (par ClamAV ou anti-virus propriétaire)

Reverse proxy

- Permettre à des machines en adressage privé d'être accessibles de l'extérieur
- Faire de la répartition de charge entre plusieurs serveurs webs.
- Filtrer les attaques.
- Réaliser de l'accélération HTTP
  - Pré-loading
  - Mise en cache des pages dynamiques
  - ...

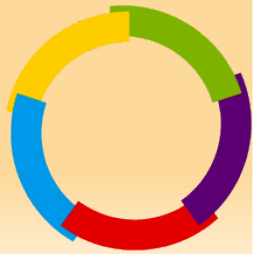


Reverse Proxy



Proxy

- Solutions libres
  - Squid
  - Apache
    - mod_proxy
    - mod_security
  - Varnish
    - Le plus abouti, orienté accélération

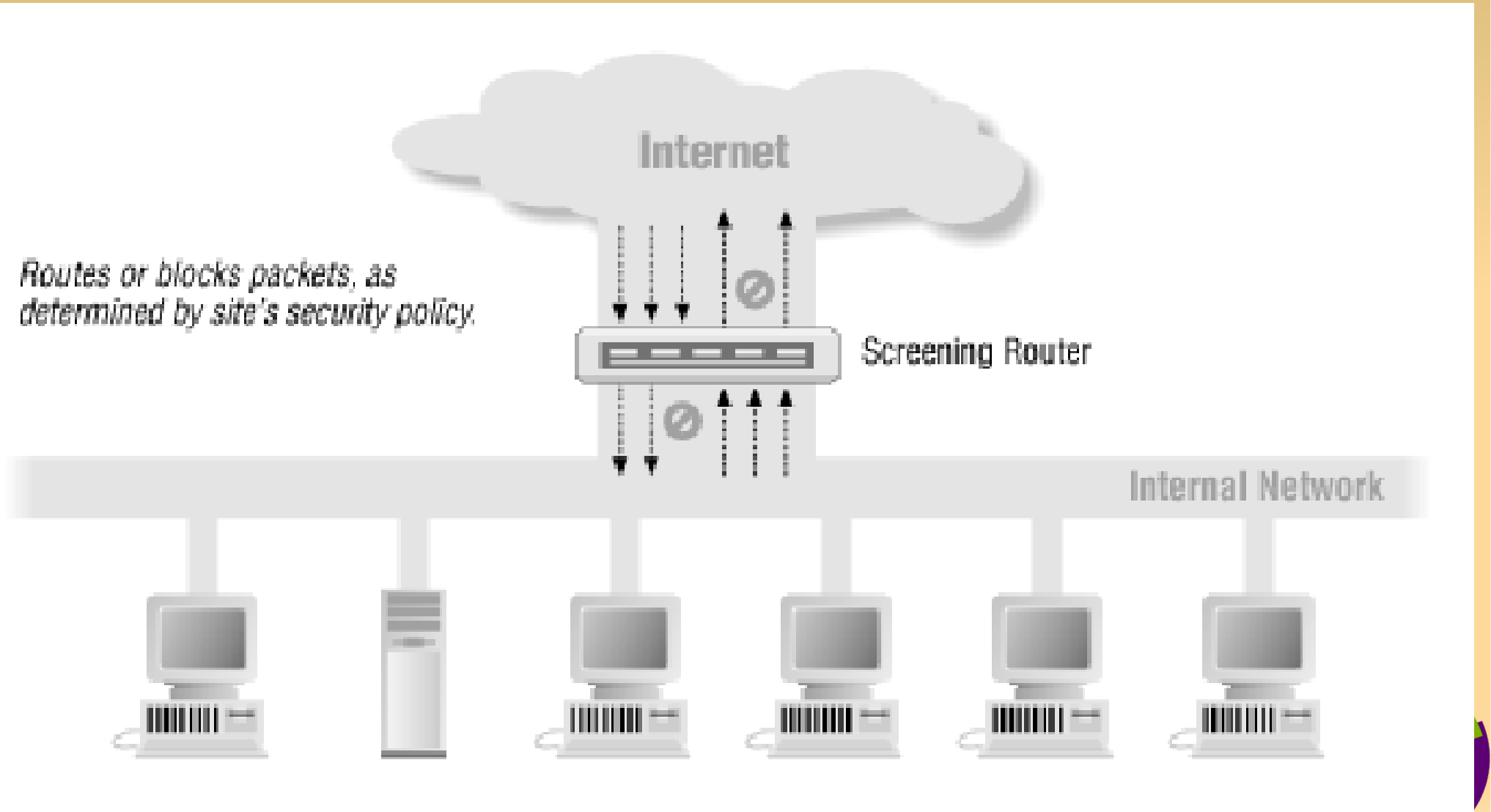


Architectures

- Comment concevoir son réseau en termes de sécurité ?
  - Compartimentation
  - Répartition des équipements de sécurité(1)
- Buts
  - Maximiser l’impact des protections
  - Minimiser l’effet d’escalade en cas d’intrusion

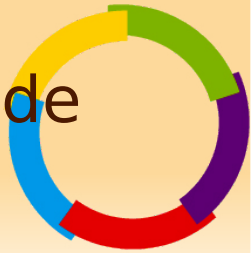
(1)Illustrations extraites de ”Building Internet Firewalls”, éditions O’Reilly

Filtre Simple

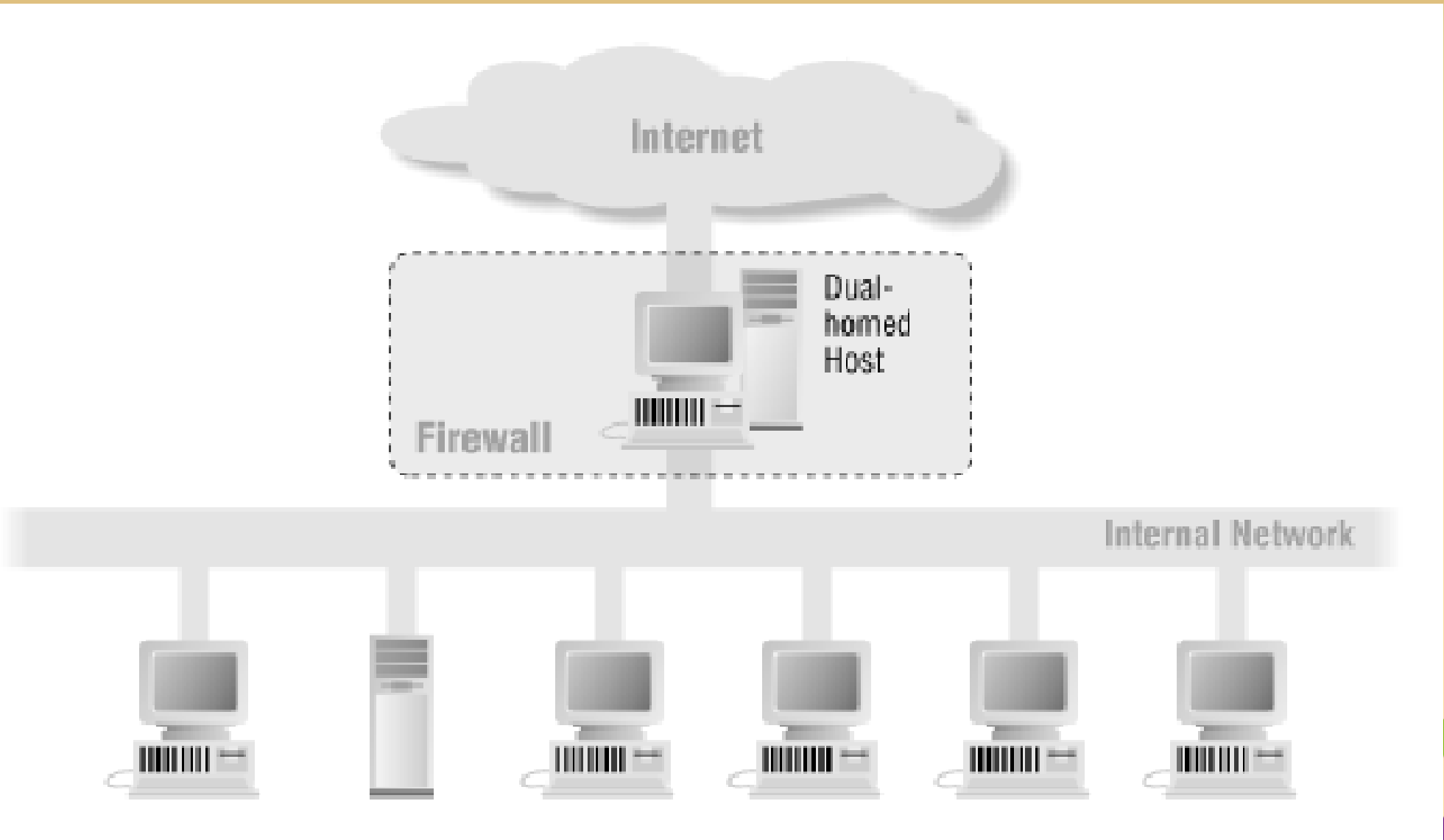


Filtre simple

- Seul le trafic sortant est autorisé, système de la diode
  - Avantage
    - Simple
    - On ne risque pas d'attaques externes
  - Limitations
    - Pas de possibilité d'offrir l'accès à des services depuis l'extérieur
    - Pas de traçabilité sur ce que font les utilisateurs en interne (sites visités, ...)
    - Pas de protection contre la récupération de code hostile (web, pop3, ...)

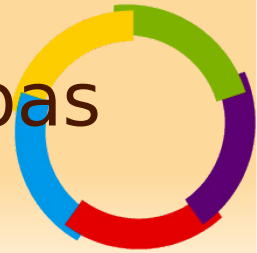


Bastion filtrant (ie proxy)

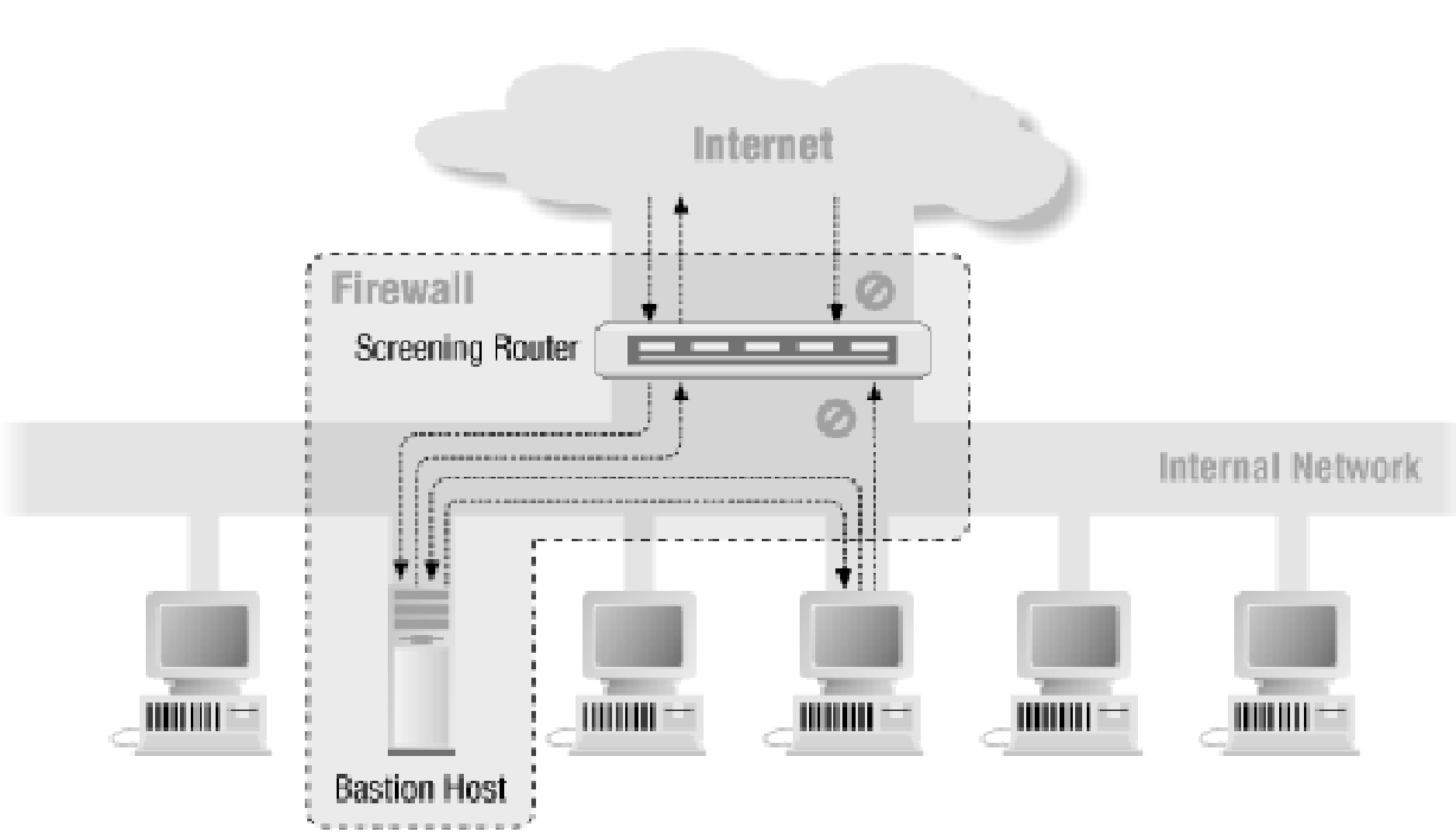


Bastion filtrant (ie proxy)

- Avantages
  - Un seul équipement
  - Filtrage applicatif possible (proxy web, pop3, ...)
- Inconvénients
  - Ne fonctionne qu'avec les flux proxyfiables.
  - Un serveur avec beaucoup de services (les différents proxy) très exposé (branché en direct sur internet sans firewall).
  - L'hébergement est possible mais risqué (pas de protection par un firewall).

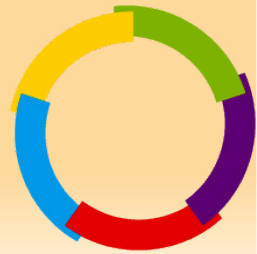


Filtre & Bastion

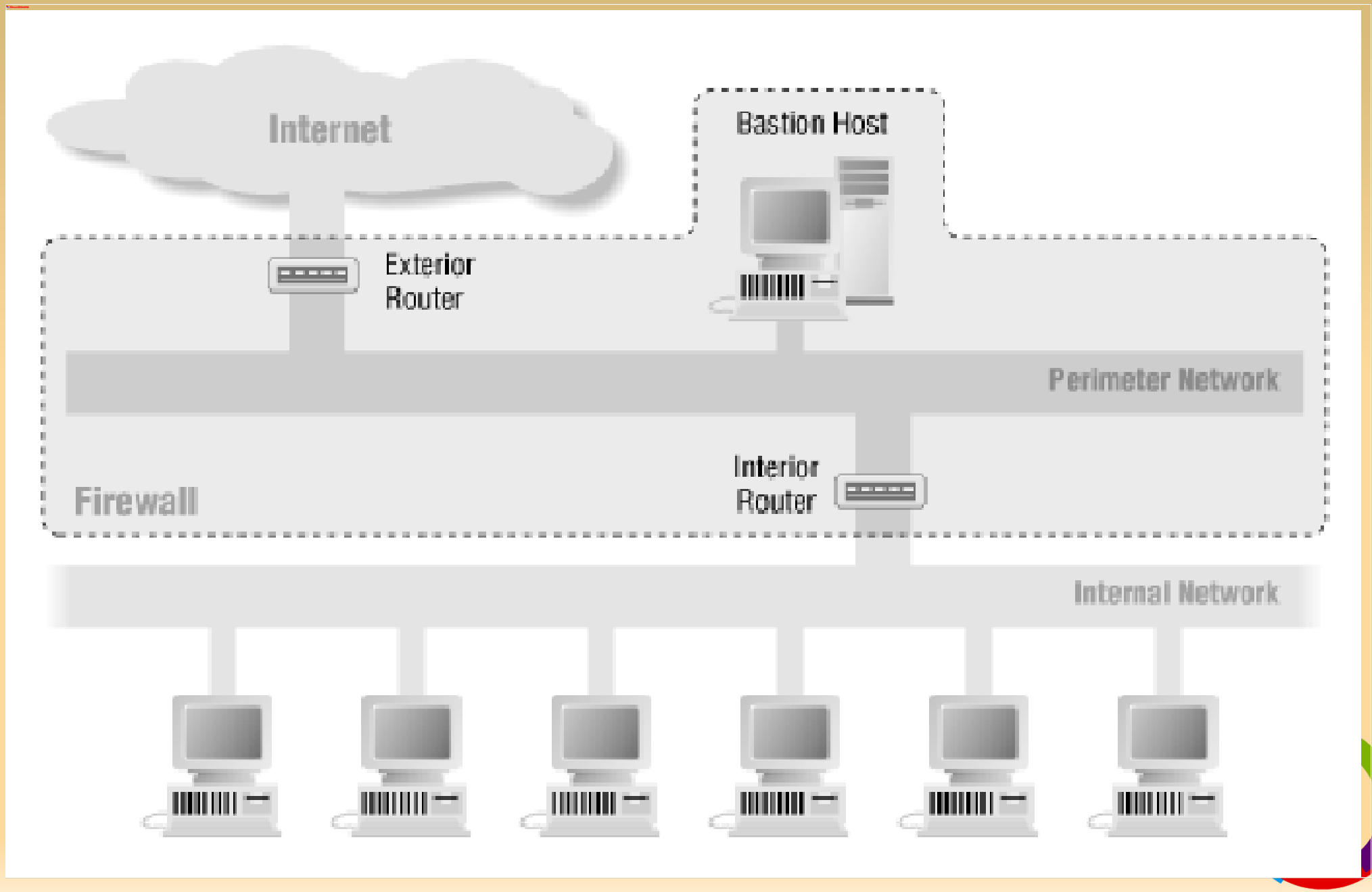


Filtre et bastion

- Avantages
  - Le bastion est mieux protégé
- Inconvénients
  - Pas d'hébergement possible (ou alors via une redirection de port qui rend extrêmement vulnérable le réseau interne).
  - Les postes de travail peuvent contourner la politique de filtrage/log du bastion via des attaques de couche 2 (arp cache poisoning et NAT pour se faire passer pour le bastion).

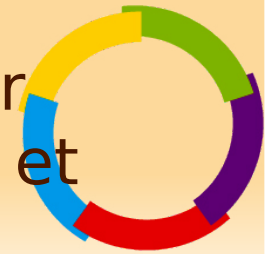


Bastion en sandwich

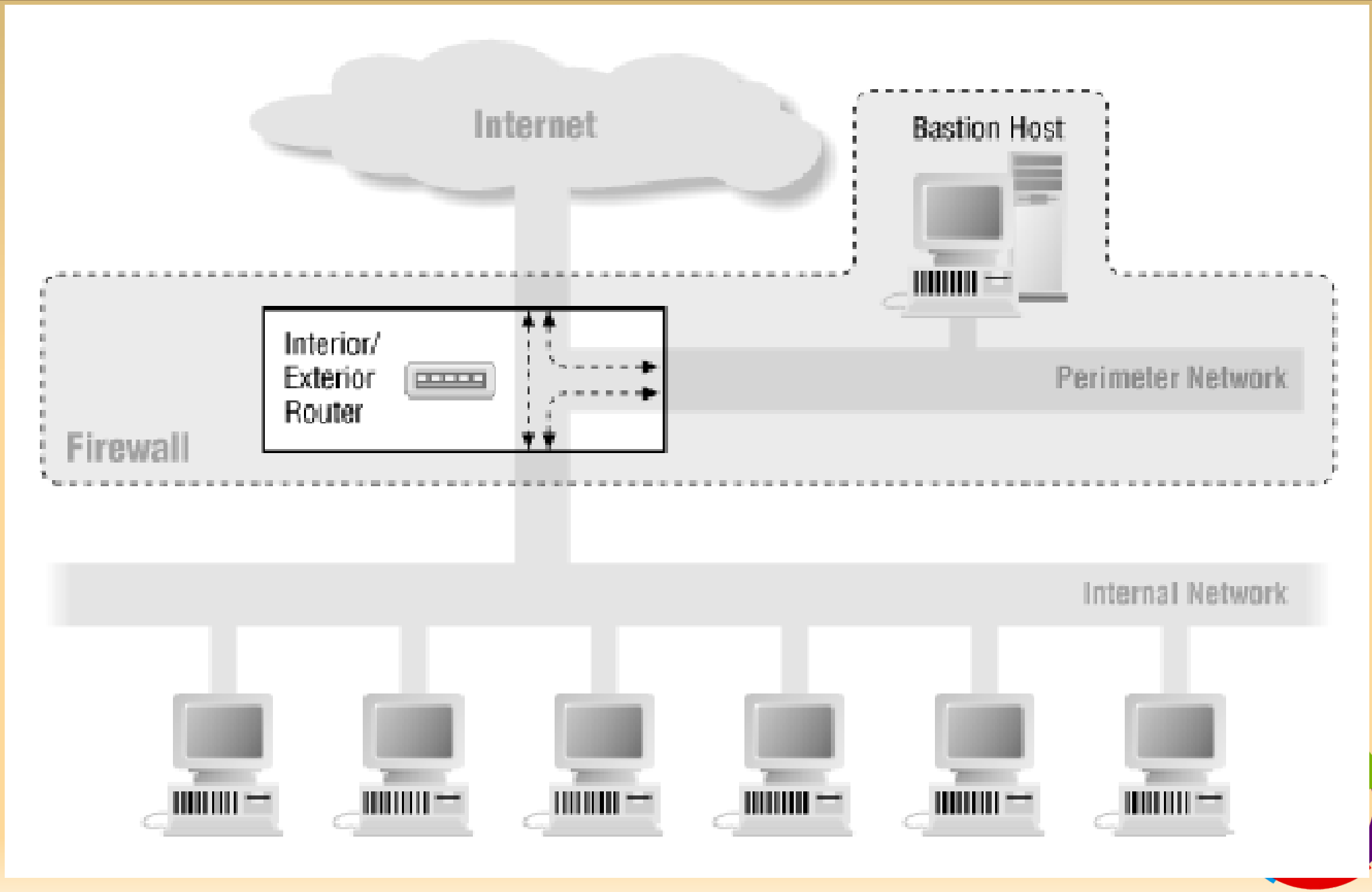


Bastion en sandwich

- Avantages
  - Le bastion est protégé à la fois des utilisateurs internes et de l'exterieur.
  - On peut commencer à envisager un hébergement de services
- Inconvénients
  - Deux firewalls...
  - Besoins d'une table de routage avec deux passerelles sur le bastion (trop compliqué pour un admin système ;-)).
  - En cas de compromission du serveur d'hébergement, le pirate a accès à la zone par laquelle transite tous les flux entre l'extérieur et l'intérieur (très grave!)



Le bastion en DMZ



Le bastion en DMZ

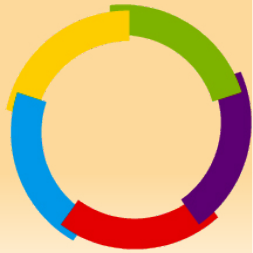
- Avantages
  - Comme précédent en plus simple
- Inconvénients
  - Les mêmes que dans le précédent si ce n'est que la compromission est un peu moins grave car elle ne permet l'écoute et l'altération « que » des flux proxysés.

Pistes de meilleurs solutions solution ?

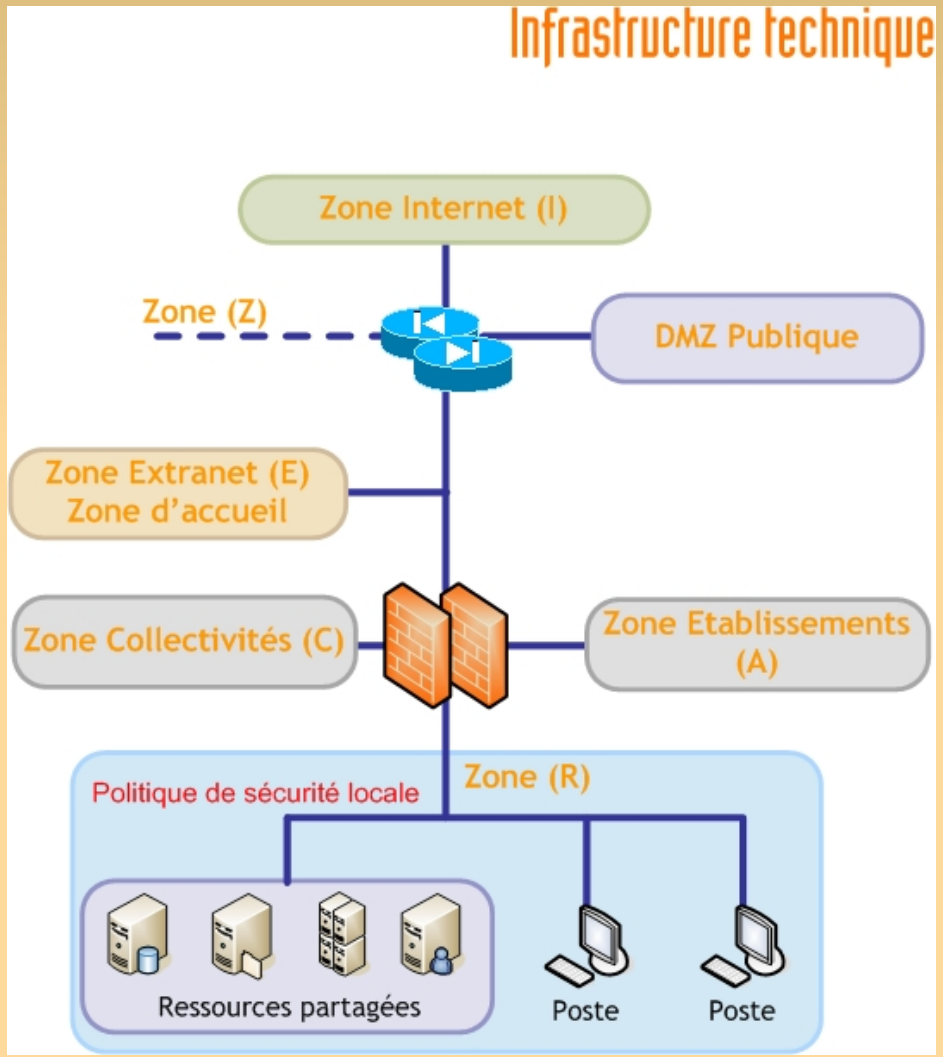
- Un firewall avec deux DMZ
  - Une DMZ pour le(s) serveur(s) hébergeant des services
    - Impact d'une compromission faible, pas d'accès aux postes internes ou aux flux sortant des postes.
  - Une DMZ pour le(s) proxy(s)
    - Protection contre les compromissions éventuelles des serveurs d'hébergement.
    - Pas de risque d'attaque de couche 2 des postes de travail pour contourner la protection.

Des exemples

Architecture des Rectorats de l'éducation nationale



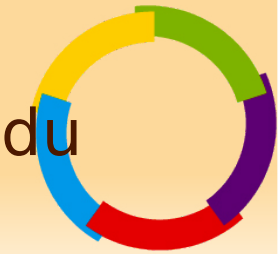
Modèle Bi-Firewalls



- Le bastion est dans la zone « ressources partagées ».

Architecture bi-DMZ

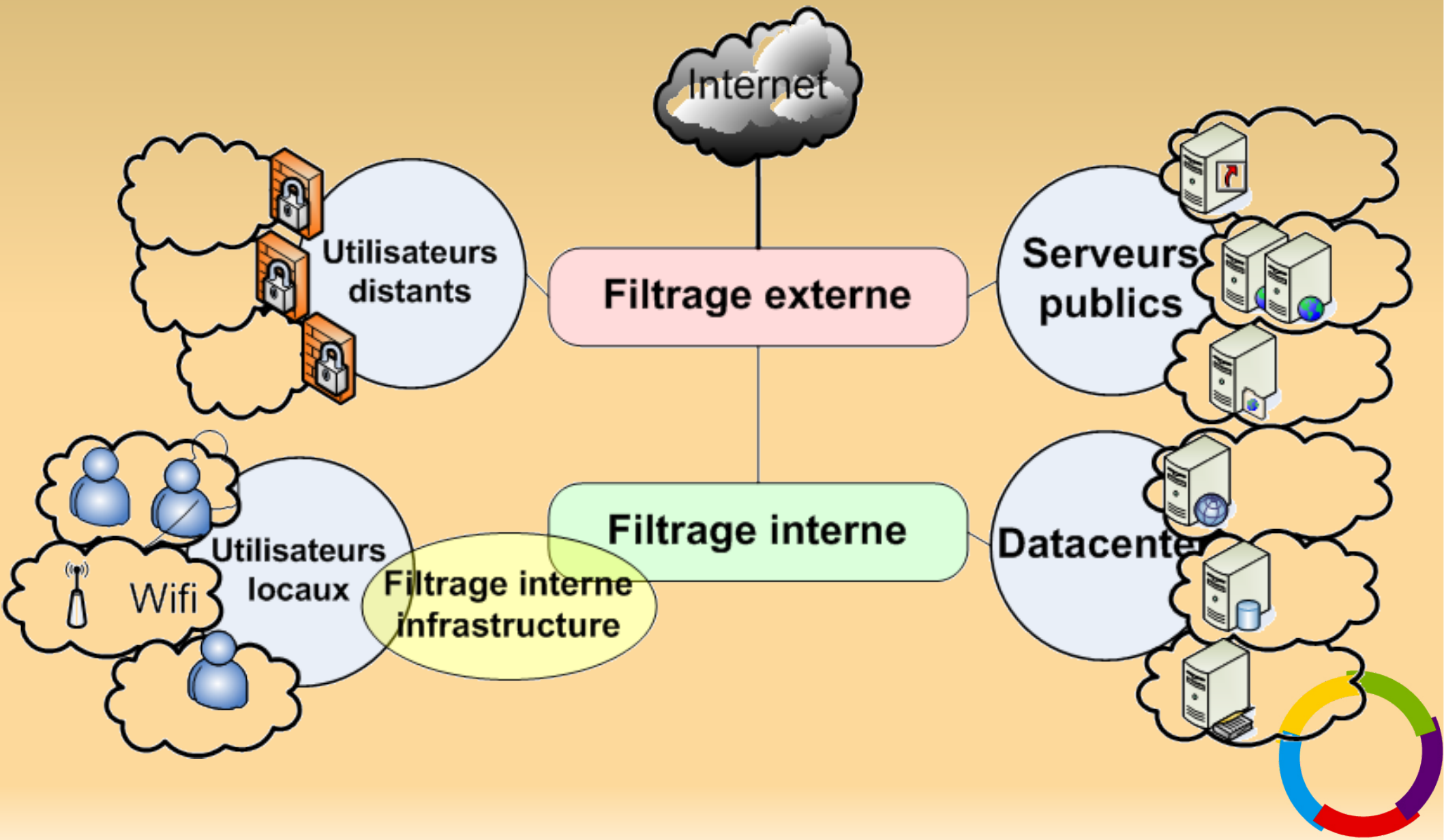
- Avantages
  - On commence à avoir une belle architecture
  - Tous les équipements sont redondés
  - Deux niveaux de firewalls de deux constructeurs différents
- Inconvénients
  - Manque de segmentation des DMZ
  - Pas de filtrage entre les utilisateurs et le datacenter (ie la zone ressources partagées).
  - Le proxy est dans le même LAN que les utilisateurs.
  - Pas de prise en compte des connexions VPN, du wifi



Exemples

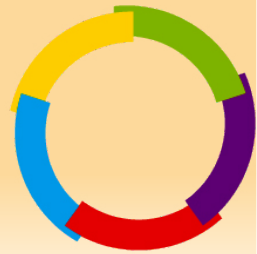
Architecture cible des Rectorats de l'éducation nationale

Architecture BI-DMZ resegmentée



Architecture BI-DMZ resegmentée

- Avantages
  - Comme précédents
  - Bien meilleure granularité dans la segmentation (zones d'hébergements, zones d'utilisateurs, proxy séparés des users).
  - Prise en compte du wifi et des terminaisons VPN
- Inconvénients
  - Ca commence à être compliqué (mais il faut bien justifier nos revenus...)
  - Avec autant de LAN, le coût s'en ressent (et heureusement que l'on utilise des VLAN!)



Conclusion sur l'architecture

- Trouver un compromis entre sécurité (segmentation maximum), coût (minimiser le nombre d'équipements), disponibilité (doublement des équipements), performance, ...
- On peut faire de très belles architectures de sécurité avec les logiciels libres:
  - Netfilter est un excellent firewall et offre tout ce qu'offre un firewall graphique (sans la GUI de gestion des logs et l'analyse de couche 7).
  - Idem pour les proxy, reverse proxy, IDS, ...