

BROUILLON

BROUILLON

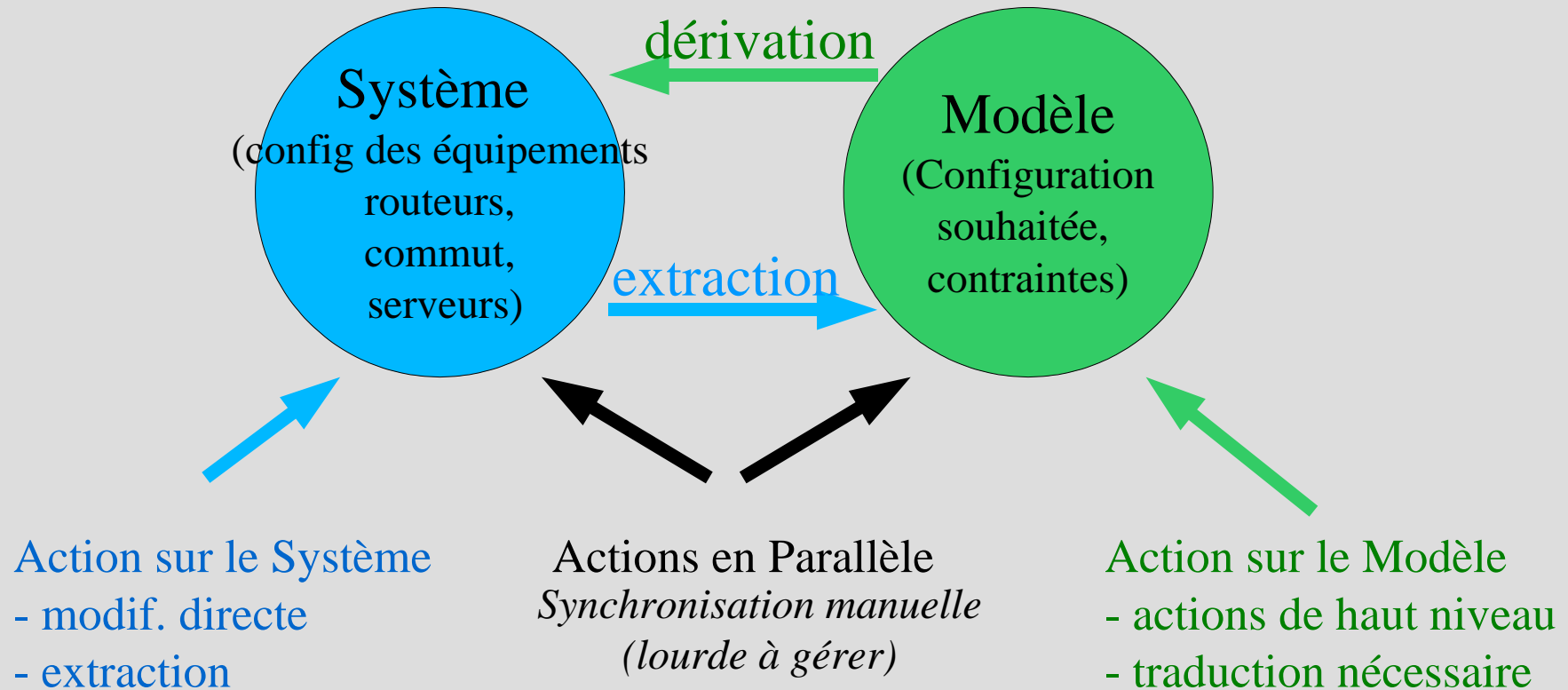
Réflexions sur l'administration réseau déléguée et
automatique

BROUILLON

L.Facq

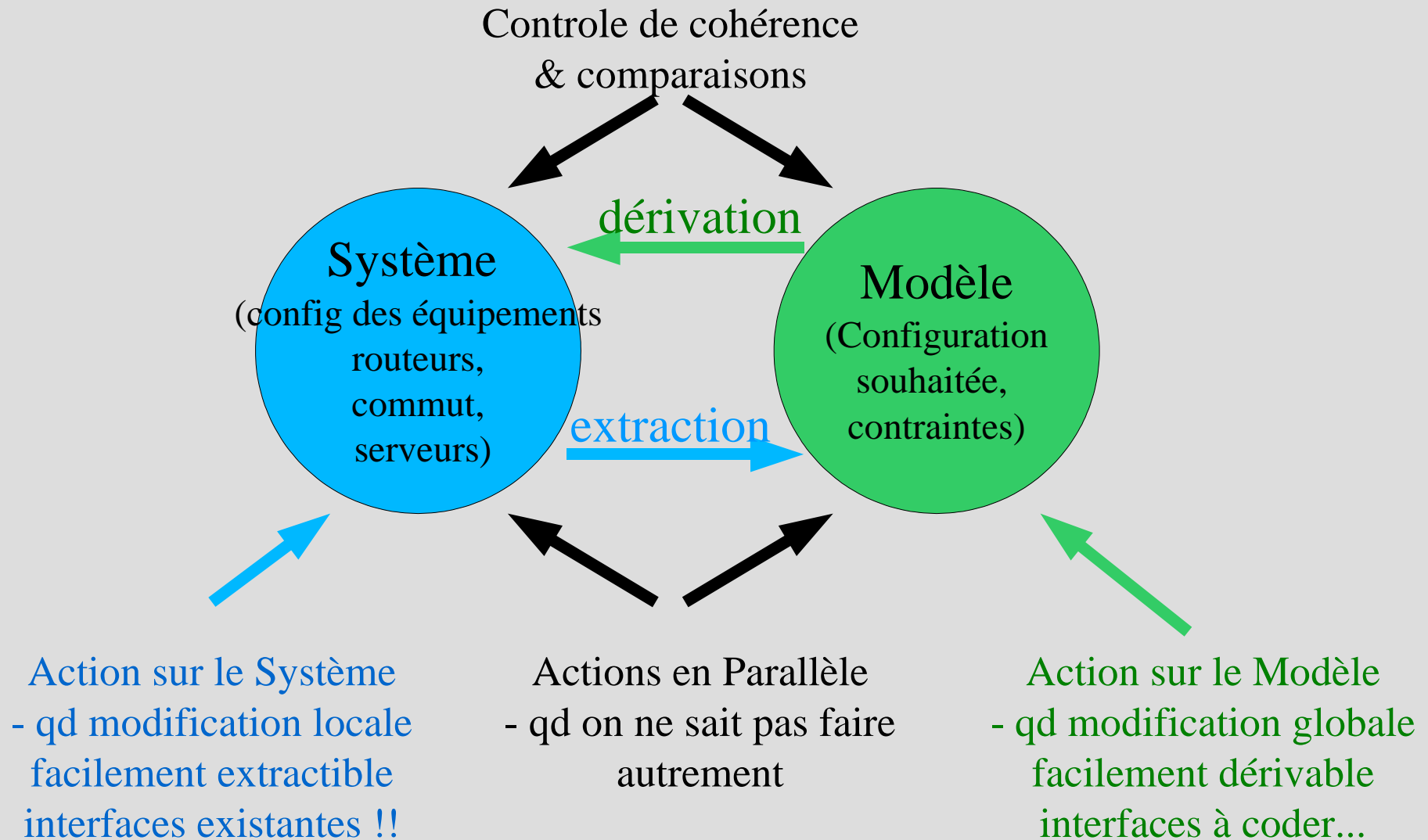
Comment synchroniser la réalité avec le modèle ?

*Synchronisation automatique
(lourde à coder)*



Synchronisation pragmatique

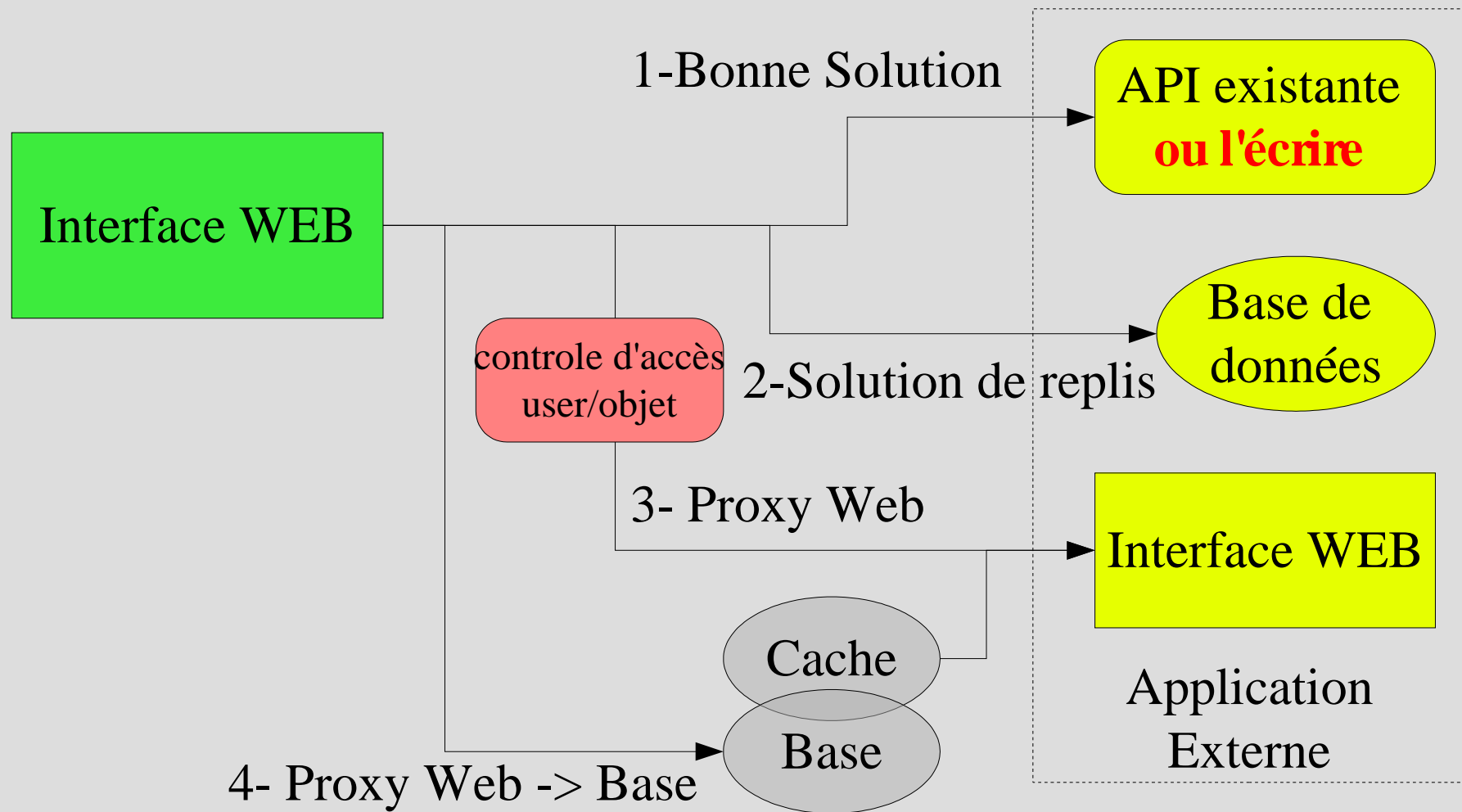
“au mieux avec un peu tout !”



Intégration de soft externes

- Réutiliser le maximum de softs existants
- “Meta” outil de configuration de ces softs
 - droit d'accès
- Se concentrer sur l'intégration de ces outils
 - Utiliser l'API existante ou en écrire une
 - Base de données = pseudo API
 - permet d'extraire les données du soft
 - Proxy Web (pseudo API) + Base de données !
 - extraction via le web et injection dans une base
 - Proxy Web tout seul
 - affichage direct des infos

Intégration de soft externes



Proxy Web

- Perl & LWP By [Sean M. Burke](#) 1st Edition June 2002
- HTML::TreeBuilder
- HTML::TokeParser
- <http://search.cpan.org/~gaas/libwww-perl/>

JFFNMS

- Orienté surveillance d'équipements de coeur
- Manque gestion de la relation CT (portail)
 - manque droit des CT sur liste d'objets (plage ip, interfaces, vlan, ...)
- Comment injecter une config existante ?
 - et comment maintenir dans le temps (réinjecter)
 - => comparer et appliquer différences (patch!)

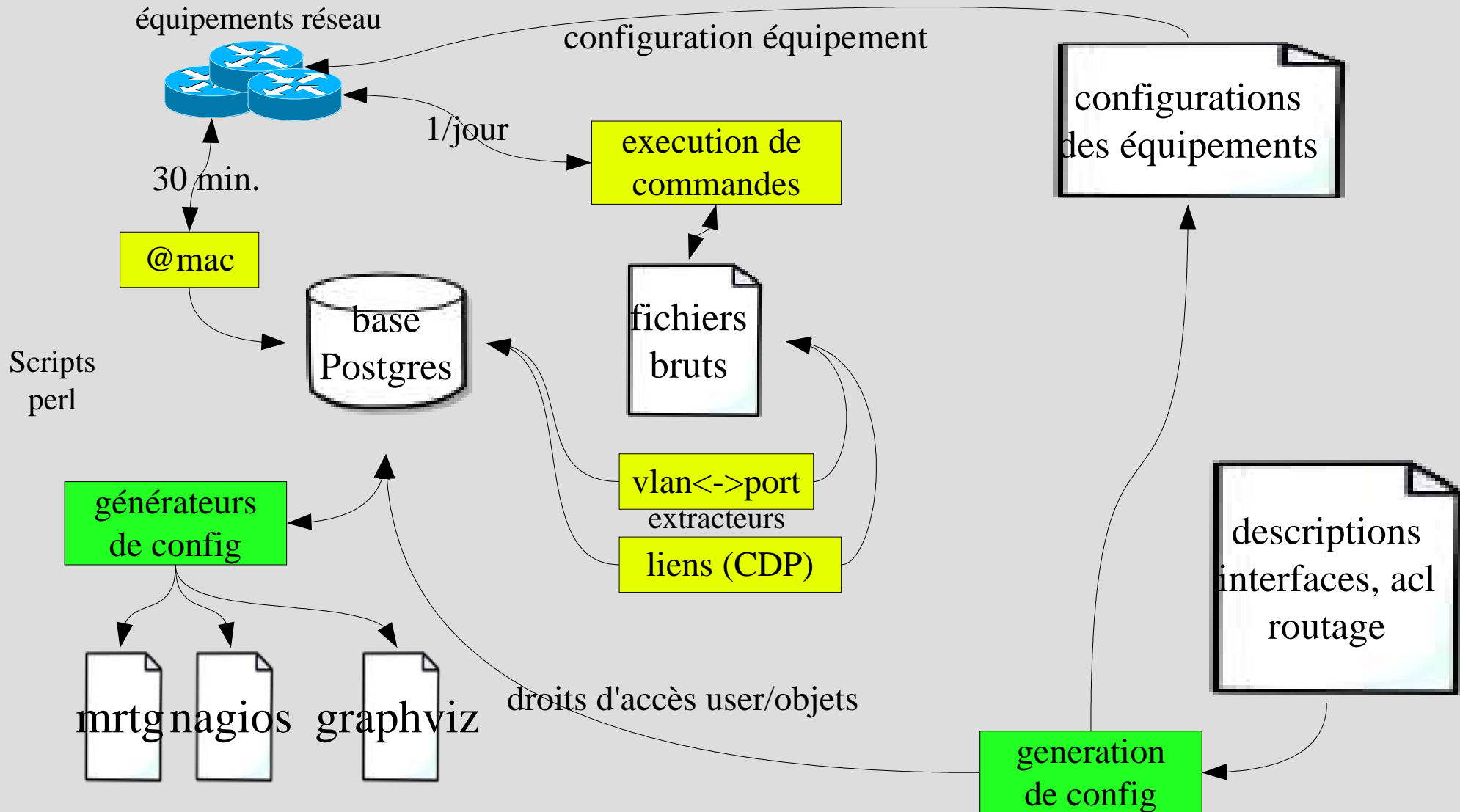
OpenNMS

router-conf / reabase

- Objectif : supervision de réseau “sans config”
- Statut: expérimental, non diffusé (manque de temps)
- Contraintes & philosophie :
 - config la plus auto possible
 - recup de la config réelle (physique->modèle)
 - plutot scripter que cliquer
- génération des config d'interfaces, acl, manipulation de plages d'adresses
- recup des config,etat interfaces, cdp -> fichiers
- extraction de ces infos => base postgres
- base postgres -> config nagios, et outils tiers

Extraction de la config

100% auto... 100% cisco



Objets Concernés

- Correspondants (personnes)
 - droits sur les objets
- Objets Physiques
 - actifs réseau (routeurs, commut)
 - passifs réseau (liaisons)
 - interfaces physiques (L2&3), ports (L2)
 - serveurs
 - machines utilisateurs
- Objets Logiques
 - @ip, subnets et routage
 - réseau & interfaces virtuels (L3) (vlan, vp, vc)

Elements à prendre en compte

- Collecte des événements
- Statistiques
- Métrologie (flux et par interface ou port)
- Niveau d'accès
- Diffusion & édition d'informations
 - tableau de bord (global, par CT)
 - looking glass
 - extractions (rapports/mail, CSV)
- Sécurité (blocages)
- Groupes (de CT, de machines, ...)

- Expression simple (compacte?) des droits
 - expressions / match
 - subnet => interfaces phys. => vlan
 - vlan => interfaces => subnets
 - => ports => @mac
- Mais
 - vlan sans subnet (tube)

Ports et interfaces

- Port : connecteur physique (Layer2)
- Interface virtuelle: connecteur logique(Layer3)
- Interface physiques : interface + port (L2+L3)
- Prendre en compte HSRP, Trunk
- Port avec attributs L2 et L3 ...
- ou objets différents (port,interface) reliés entre eux
-

Vrac

- gestion des droits
 - delegation : on ne peut que ce qu'on a
 - => règles écrite par correspondants
 - recompilation => verification/regeneration a chaque fois de la cohérence
- reflexion sur les greffons / plugins
 - strategique pour outils logiciel libre – permet de faire des modifs sans avoir a reecrire si modif dans le code
- monitorer la vitalité des domaines aussi (+log) => alarmes

Vrac

- Droits sur groupes de machines (groupe ou plustot filtre, selecteur, operation sur ensemble)
- prevoir des hook comme operateur ensemble
- fusion, exclusion, ajout, soustraction

Vrac

- génération auto de tables plates des droits par type d'objet (subnet, string)
 - exploitation facile avec routine du genre (simple select pour existence du quadruplet) :
 - `check_right_net(qui,quoi,typequoi,droit)`
 - `check_right_strings(qui,quoi,typequoi,droit)`

Vrac

- qui | quoi | type-quoi | droit
 - qui : id ou nom ? nom pour commencer
 - quoi | type quoi
 - subnet (inet6) | subnet ou host ? utile ?
 - strings | type-quoi
 - email : toto@domain.tld
 - emaildomain : domain.tld
 - acl : host@ @aclnameornum
 - interface : host@ @interface (syntaxe constructeur, @@ comme séparateur)
 - droit
 - view, read, write, delegate, view-mrtg, view-showver
 - view-command
- nécessite un séparateur universelle @@ ?

Vrac

- hostlistgroup|hostlist|what|compiled|comment
lastchange
- hostlistgroup|hostlist|description
- rights_string
 - hostlistgroup -> droit
 - hostlistgroup@ @hostlist

Plugins, Hook, Scriptable

- apache – mozilla
- design pattern

Concept “N2T”

- No Time To
 - .. administer this network
 - ... build a good software
 -
- Pas Le Temps d'Administrer Ce Réseau
- PaLTACR
-
-

Développer ensemble

- Pas trivial, surtout à distance
- Depasser ses besoins propres
- => travailler sur 2 sites <> permet d'avoir les bonnes hatibutes (pour passer a plus ensuite!)
- commencer par discuter
- puis faire converger les outils existants petit a petit et se mettant d'accord sur des points communs.
- developper ensemble des packages pour réécrire proprement l'existant.