Protocoles d'administration Réseau

(SNMP - NETCONF)

Administration des réseaux

- La gestion des anomalies (Fault Management)
- La gestion de la configuration réseau
- La gestion des performances
- La gestion de la sécurité
- La gestion de la comptabilité (Accounting Management)

La gestion des anomalies

(Fault Management).

- On doit être en mesure de localiser le plus rapidement possible toute panne ou défaillance.

- Pour cela, on surveille les alarmes émises par le réseau, on localise un incident par un diagnostic des alarmes, on journalise les problèmes...

La gestion de la configuration réseau (Configuration Management).

- Il convient de gérer la configuration matérielle et logicielle du réseau pour en optimiser l'utilisation.
- Il est important que chaque équipement, chaque compteur... soit parfaitement identifié de façon unique à l'aide d'un nom ou identificateur d'objet OID (Object Identifier).

La gestion des performances

- Il convient de contrôler à tout moment le réseau pour voir s'il est en mesure d'écouler le trafic pour lequel il a été conçu.

La gestion de la sécurité

- On gère ici les contrôles d'accès au réseau, la confidentialité des données qui y transitent, leur intégrité et leur authentification.

La gestion de la comptabilité (Accounting Management).

- L'objectif est de gérer la consommation réseau par abonné en vue d'établir une facture.

Protocoles d'administration Réseau

- Il existe plusieurs protocoles capables de répondre aux attentes de l'administration réseau, le plus connu est SNMP (simple network managment protocol) mais plusieurs alternatives sont possibles dont le protocole NETCONF (network configuration) qui est potentiellement le successeur de SNMP.

SNMP

simple network managment protocol

Les différentes versions de SNMP

- **SNMPv1** (ancien standard) : Première version apparue en 1989.
- **SNMPsec** (historique): Ajout de sécurité par rapport à la version 1
- **SNMPv2p** (historique) : Ajout de nouveau type de données.
- **SNMPv2c** (expérimental) : Amélioration des opérations du protocole
- **SNMPv2u** (expérimental) : Implémente la version 2c en ajoutant la sécurité utilisateurs.
- **SNMPv2*** (expérimental) : Combinaison des meilleures parties de v2u et v2p.
- **SNMPv3** (nouveau standard) : La sécurité avant tout.

Architecture de SNMP

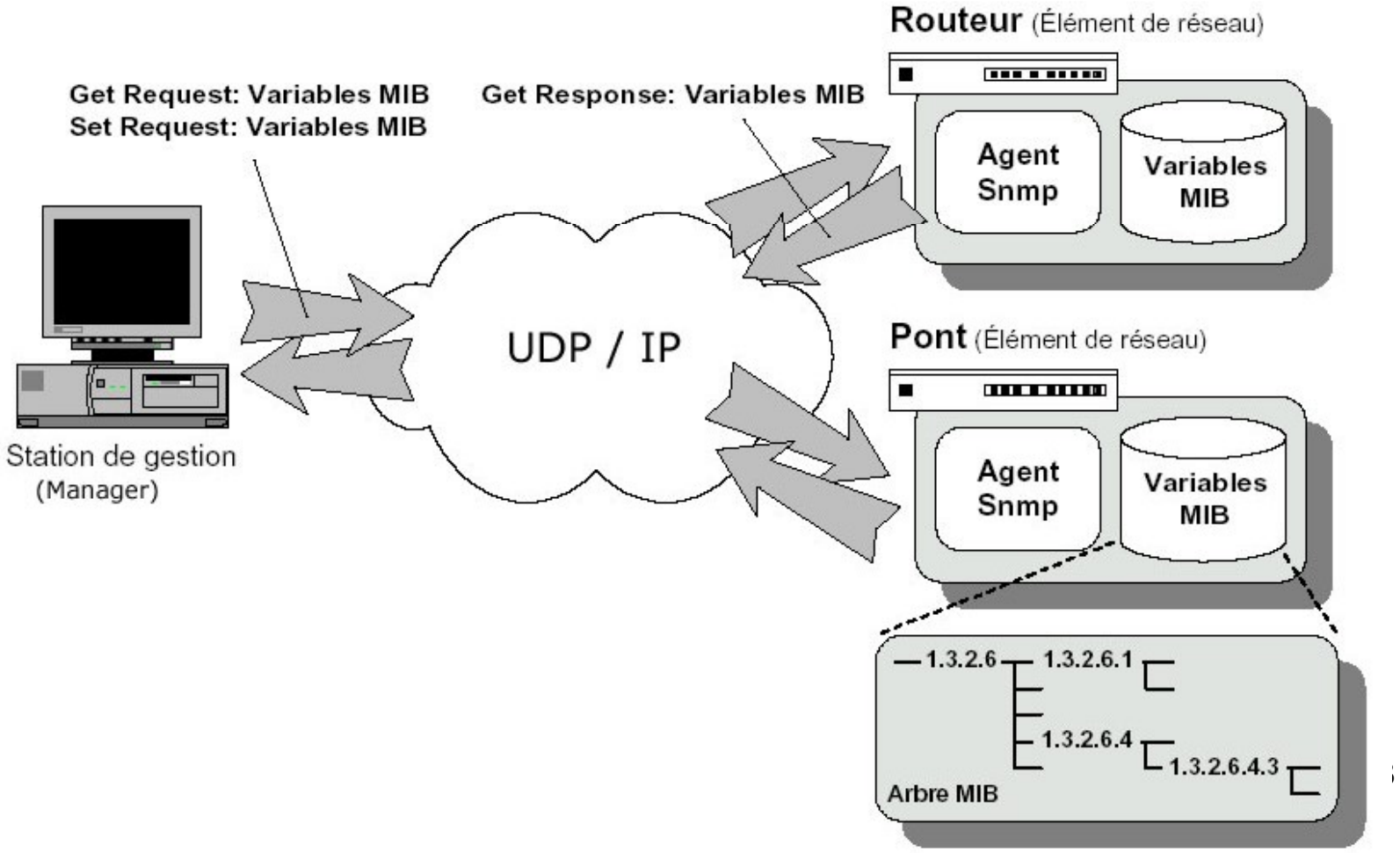
Le modèle OSI

<table>
<tr><th></th><th>Modèle OSI</th><th>Modèle TCP/IP (protocole)</th></tr>
<tr><td>7</td><td>Application</td><td rowspan="3">SNMP</td></tr>
<tr><td>6</td><td>Présentation</td></tr>
<tr><td>5</td><td>Session</td></tr>
<tr><td>4</td><td>Transport</td><td>UDP</td></tr>
<tr><td>3</td><td>Réseau</td><td>IP</td></tr>
<tr><td>2</td><td>Liaison</td><td rowspan="2">Interface réseau</td></tr>
<tr><td>1</td><td>Physique</td></tr>
</table>

La remontée d'informations

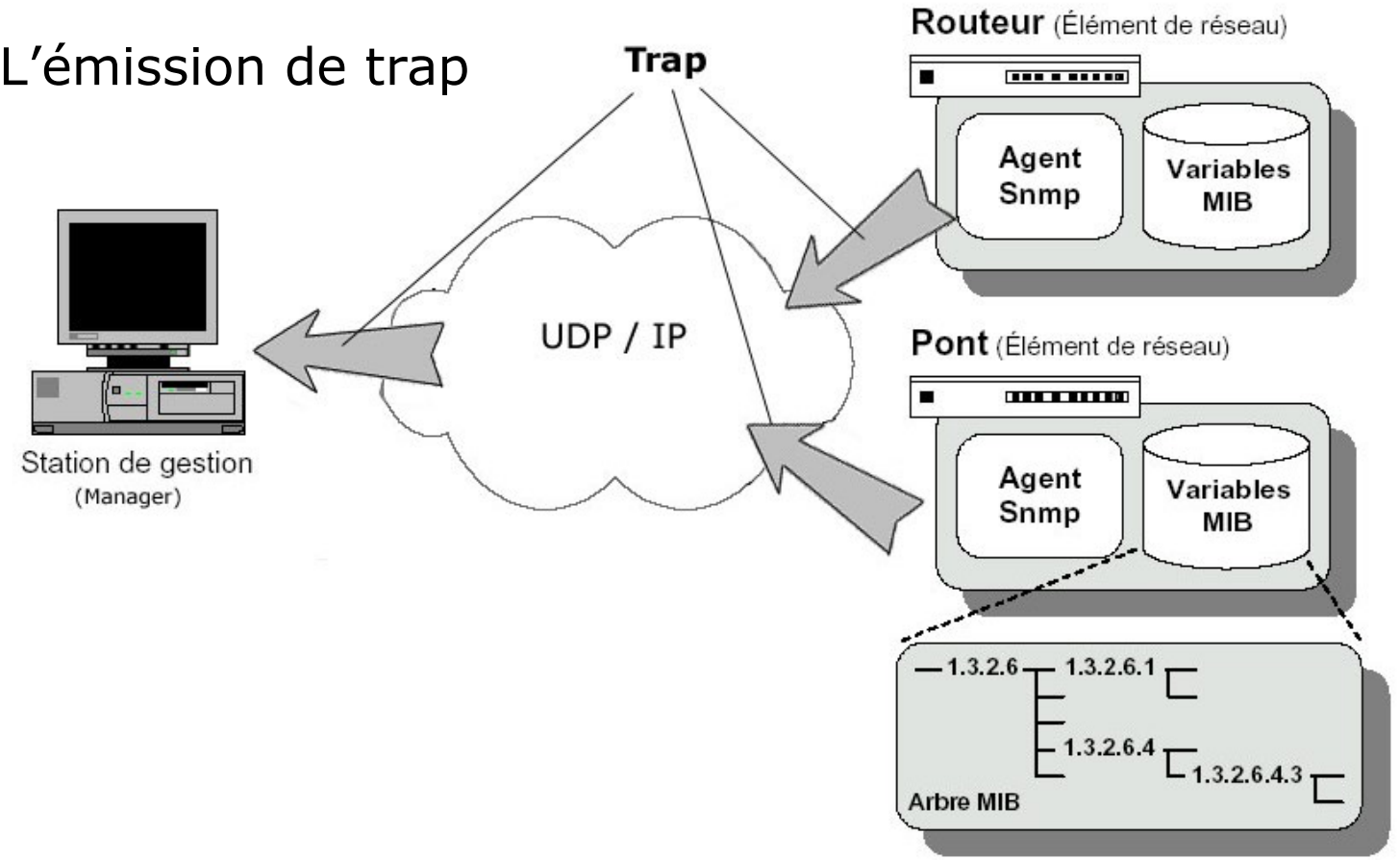
➢ Nous avons le choix entre deux méthodes complètement différentes mais qui peuvent être complémentaires :

✓ Le polling



La remontée d'informations

- ✓ L'émission de trap



Les requêtes SNMP

- Recherche d'informations :
  - ✓ **GetRequest** : recherche d'une variable sur un agent.
  - ✓ **GetNextRequest** : recherche de la variable suivante.
  - ✓ **GetBulkRequest** : recherche d'un groupe de variables
- Envoie d'informations
  - ✓ **Trap** : détection d'un incident
- Modification de valeurs :
  - ✓ **SetRequest** : permet de changer la valeur d'une variable d'un agent.

A titre d'information, d'autres requêtes existent (ex: InformRequest…) mais ne seront pas abordées dans cette présentation.

Les réponses SNMP

Une seule réponse existe.

Elle est différente s'il y a une erreur ou non.

- Aucune erreur :

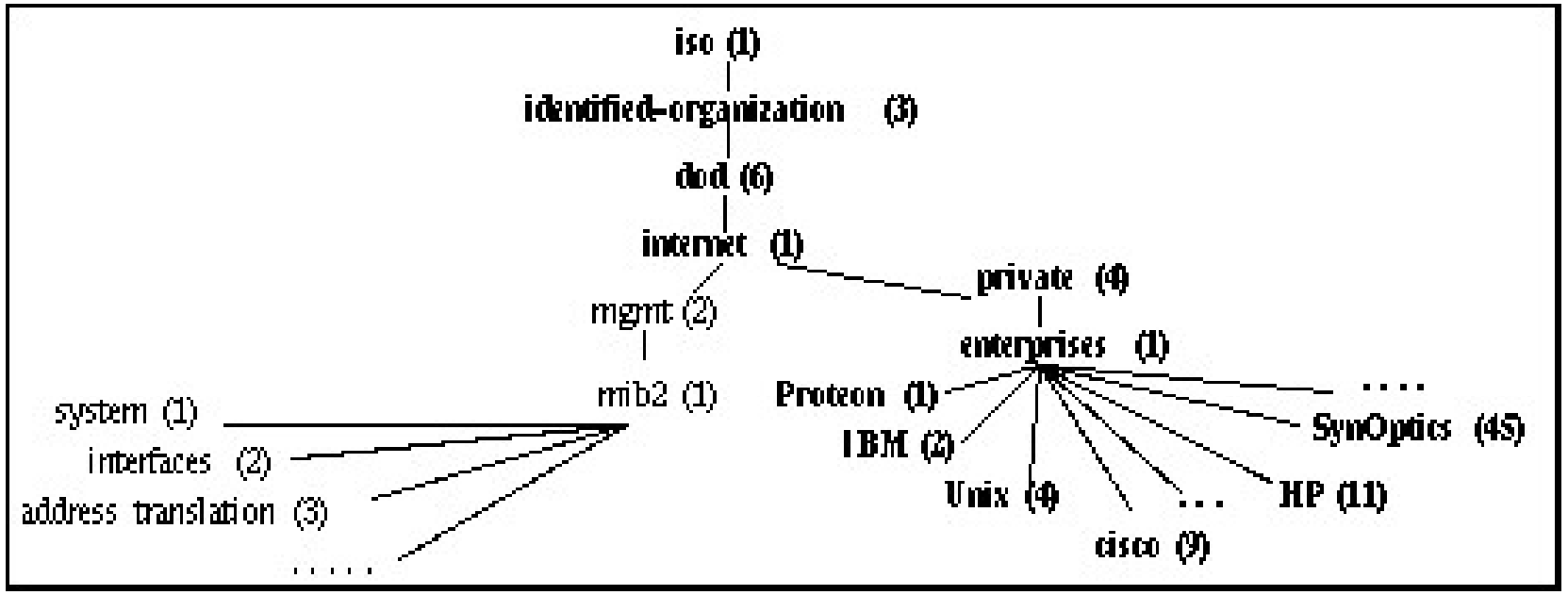
  GetResponse : renvoie la ou les valeurs souhaitées

- En cas d'erreur :

  GetResponse mais accompagné d'un *NoSuchObject*

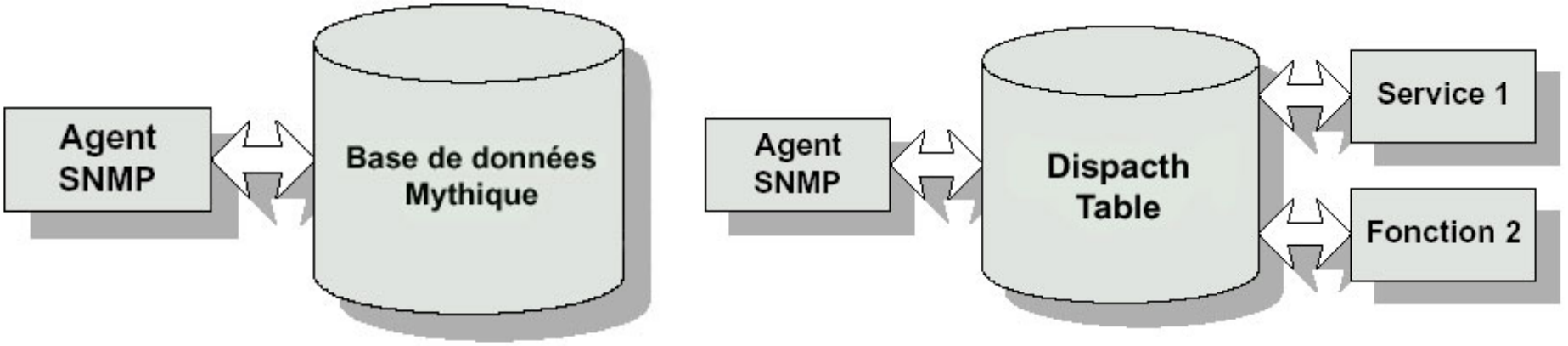
MIB (Management Information Base)

- Ensemble d'objets structurés de manière arborescente



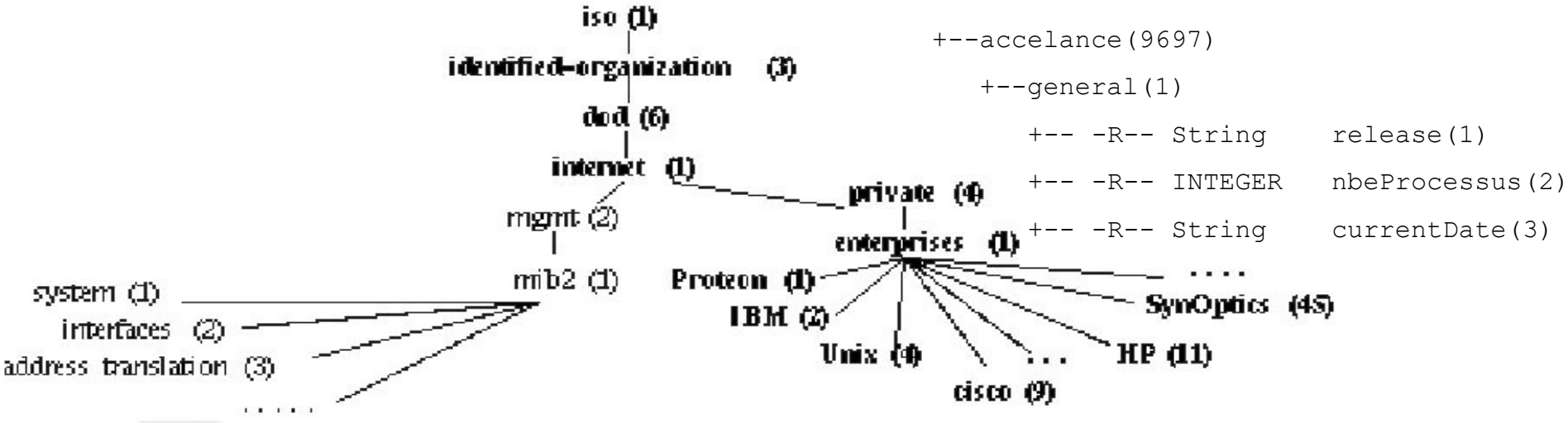
MIB (Management Information Base)

- Accès à un objet via un Object Identifier (OID)
- Deux MIB normalisées: MIB I et MIB II
  - ✓ Références des informations réseau (interfaces, ip …)
- MIB privée sous le nœud enterprises
- **Une MIB n'est pas une base de données mais une « dispatch table »**

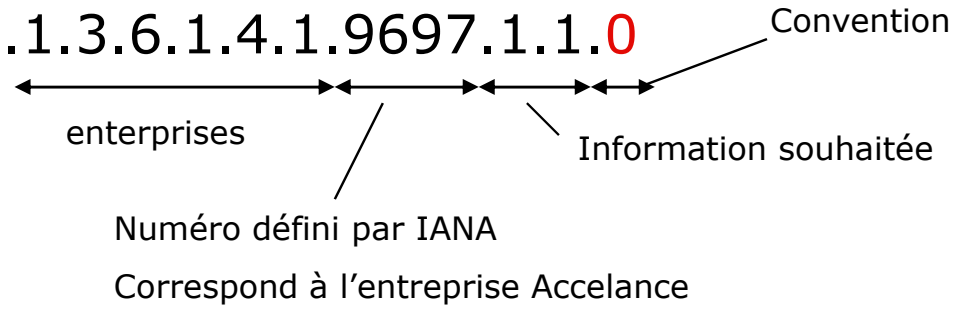


Exemple d'accès à un objet d'une MIB

- Objet de type simple (une seule variable)

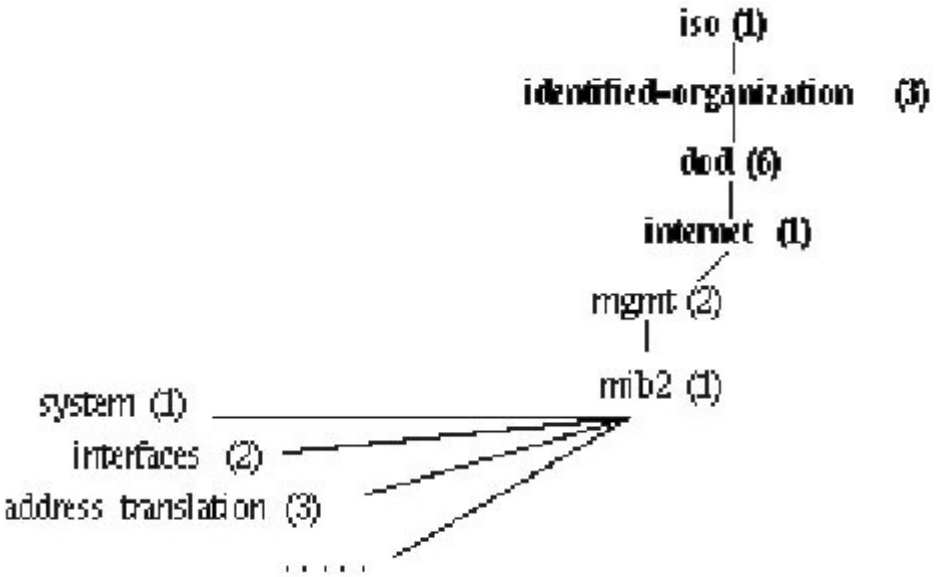


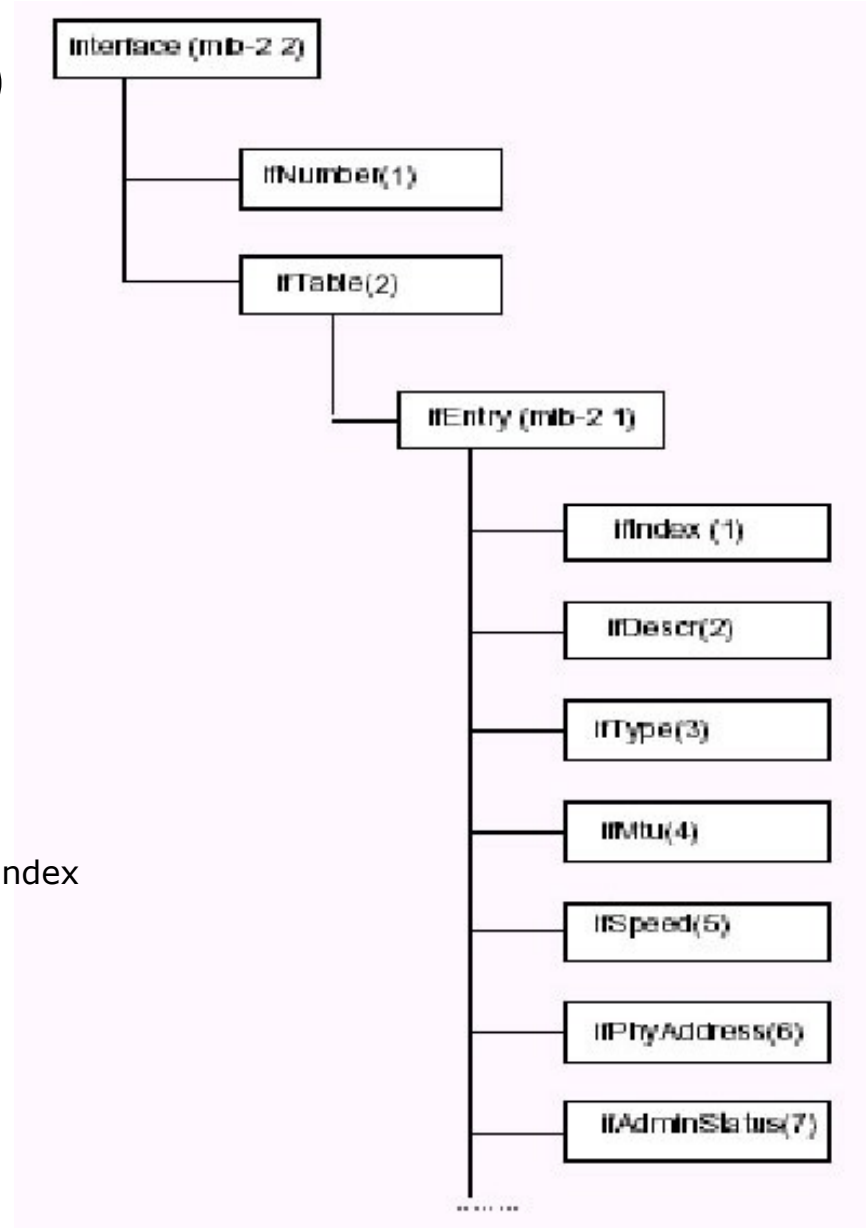
Accès à la variable ***realease*** :



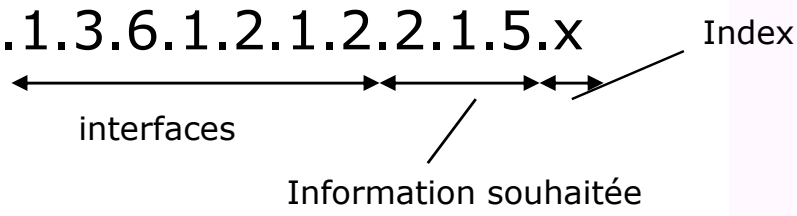
Exemple d'accès à un objet d'une MIB (2)

- Objet complexe (ex : Tableau)





Accès à la variable ***ifSpeed*** :



ASN.1 (Abstrat Syntax Notation One)

- Standard ISO (ISO 8824) qui définit plusieurs types autorisés dans SNMP (ex : INTEGER, DisplayString …)

- Effectue la correspondance entre un OID et un nom

- ASN.1 se localise au niveau de la couche *Présentation* dans le modèle OSI.

Exemple de fichier ASN.1

```
ACCELANCE-MIB DEFINITIONS ::= BEGIN
          …
accelance MODULE-IDENTITY
          …
    ::= { enterprises 9697 }

general OBJECT IDENTIFIER ::= { accelance 1 }
          …
release  OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
          "Type d'OS utilisé"
    ::= { general 1 }
```

Identification d'un fichier ASN.1

Rattachement de la branche *Accelance* à la branche *enterprises* via l'OID 9697

Affectation de la branche *general* à la branche *accelance* via l'OID 1

Définition de la variable *realease*

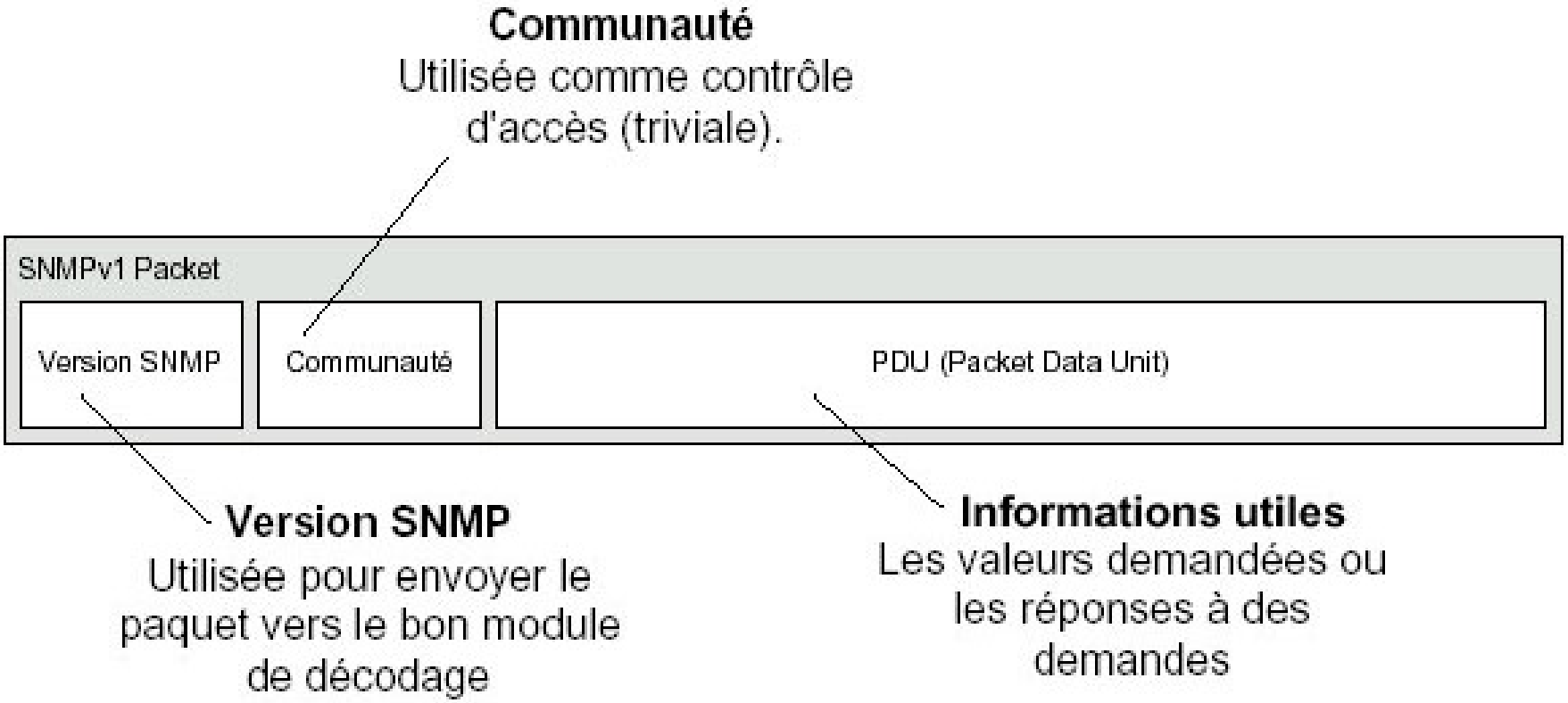
&

Rattachement de celle-ci à la branche *general*

- Désormais nous pouvons accéder à la variable *release* de la manière suivante :

.iso.dod.internet.private.enterprises.accelance.general.release.0

La Trame


Communauté
Utilisée comme contrôle d'accès (triviale).
SNMPv1 Packet
Version SNMP
Communauté
PDU (Packet Data Unit)
Version SNMP
Utilisée pour envoyer le paquet vers le bon module de décodage
Informations utiles
Les valeurs demandées ou les réponses à des demandes

- $ sudo apt-get update
- $ sudo apt-get install snmpd
- $ sudo apt-get install snmp
- $ sudo apt-get install snmp-mibs-downloader
- $ sudo dpkg -i snmpb_0.8_i386.deb
- $sudo snmpb

- # snmpwalk -v1 -cpublic localhost 1.3.6.1.2.1.1.5
- SNMPv2-MIB::sysName.0 = STRING: rafik-laptop

- # snmpwalk -v1 -cpublic localhost sysName
- SNMPv2-MIB::sysName.0 = STRING: rafik-laptop

NETCONF
(network configuration)

NETCONF

- NETCONF est un protocole de gestion de la configuration des dispositifs de données en réseau.
- Il est conçu pour couvrir les insuffisances de SNMP et Command-Line Interface (CLI), dans les fonctions de configurations réseaux.

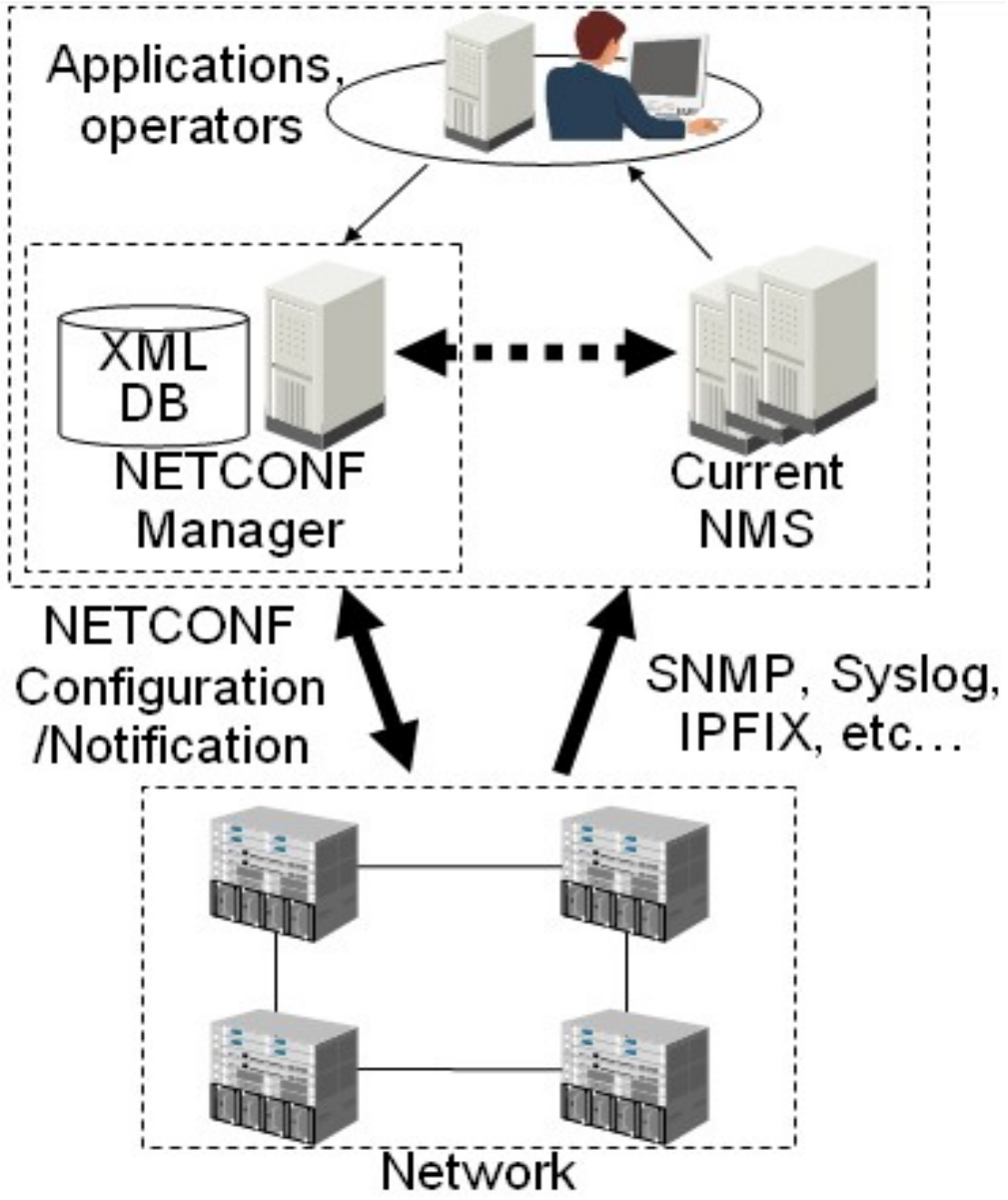
NETCONF

- Le protocole prévoit des mécanismes d'installation, manipulation, et suppression de la configuration des périphériques réseaux.

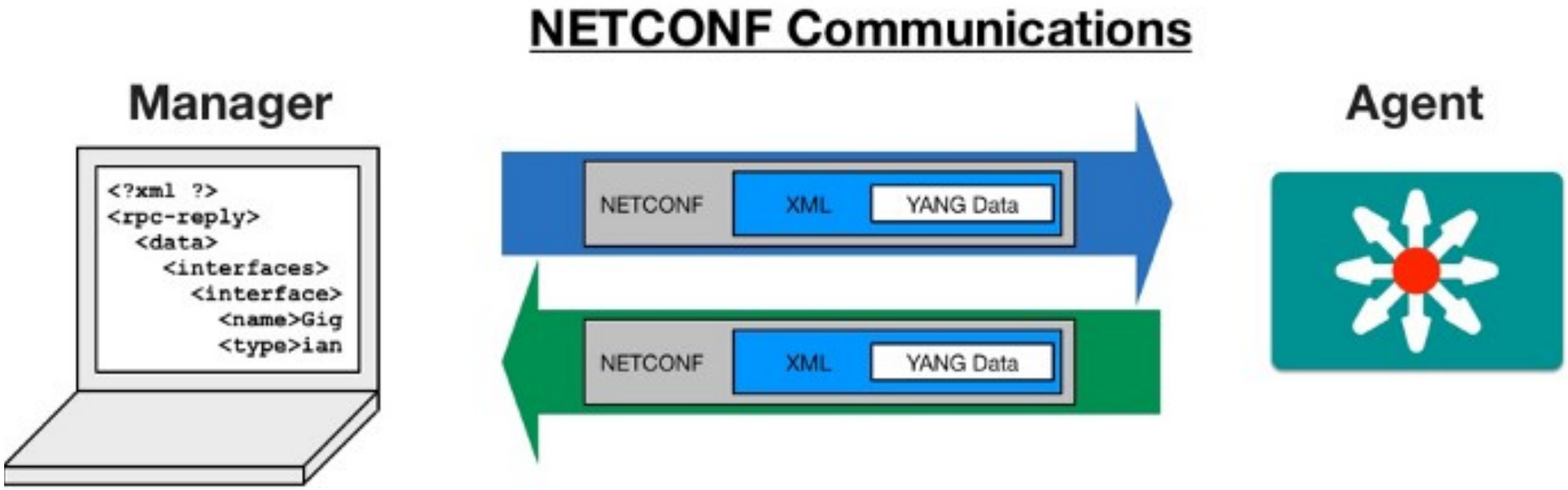
- Il utilise un langage de balisage extensible (XML) codant les données de configuration ainsi que les messages du protocole.

- Il assure la Séparation entre un état de configuration et un état opérationnel
- Il assure aussi la persistance des configurations

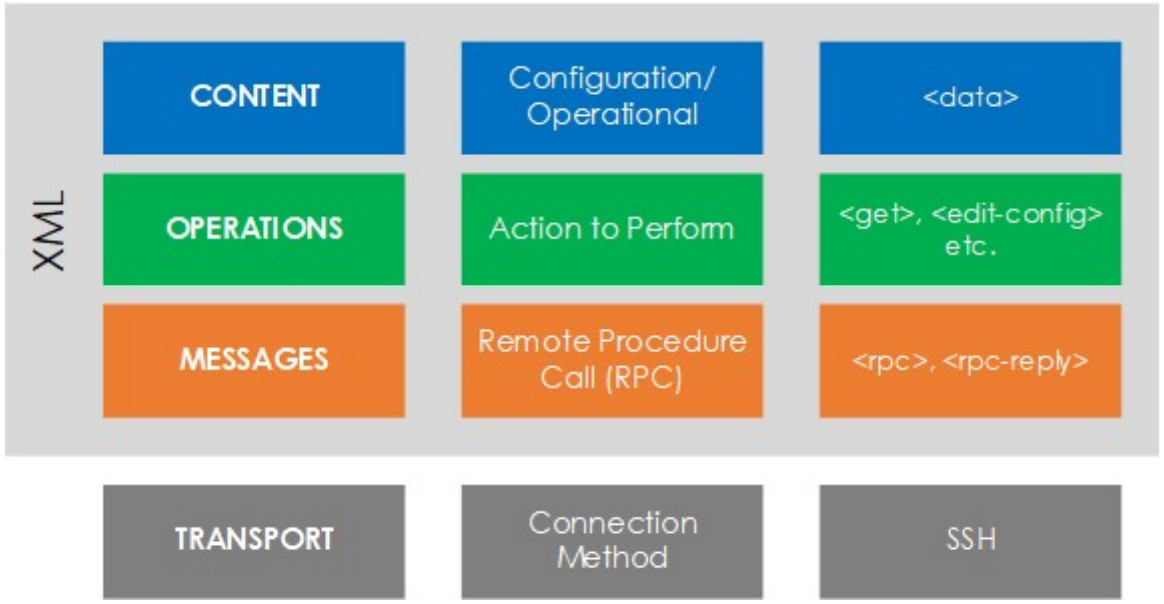
NETCONF


Applications,
operators
XML
DB
NETCONF
Manager
Current
NMS
NETCONF
Configuration
/Notification
SNMP, Syslog,
IPFIX, etc…
Network

Communication sous NETCONF



NETCONF: protocole en couches



NETCONF Data

<running config/>

<candidate config/>

<Startup config/>

<URL/>

<running/>

- Représente l'état active des configurations actuelles
- Permet à cette base de donnée d'être directement modifée
- Contient les informations sur l'état de l'équipement

<candidate/>

- Regroupe les configurations à appliquer après qu'elle soient validé par le serveur
- Les changements fait sur cette BDD ne s'applique pas immédiatement
- Utilisation d'opérations: <lock>, <commit> pour validation .

<Startup/>

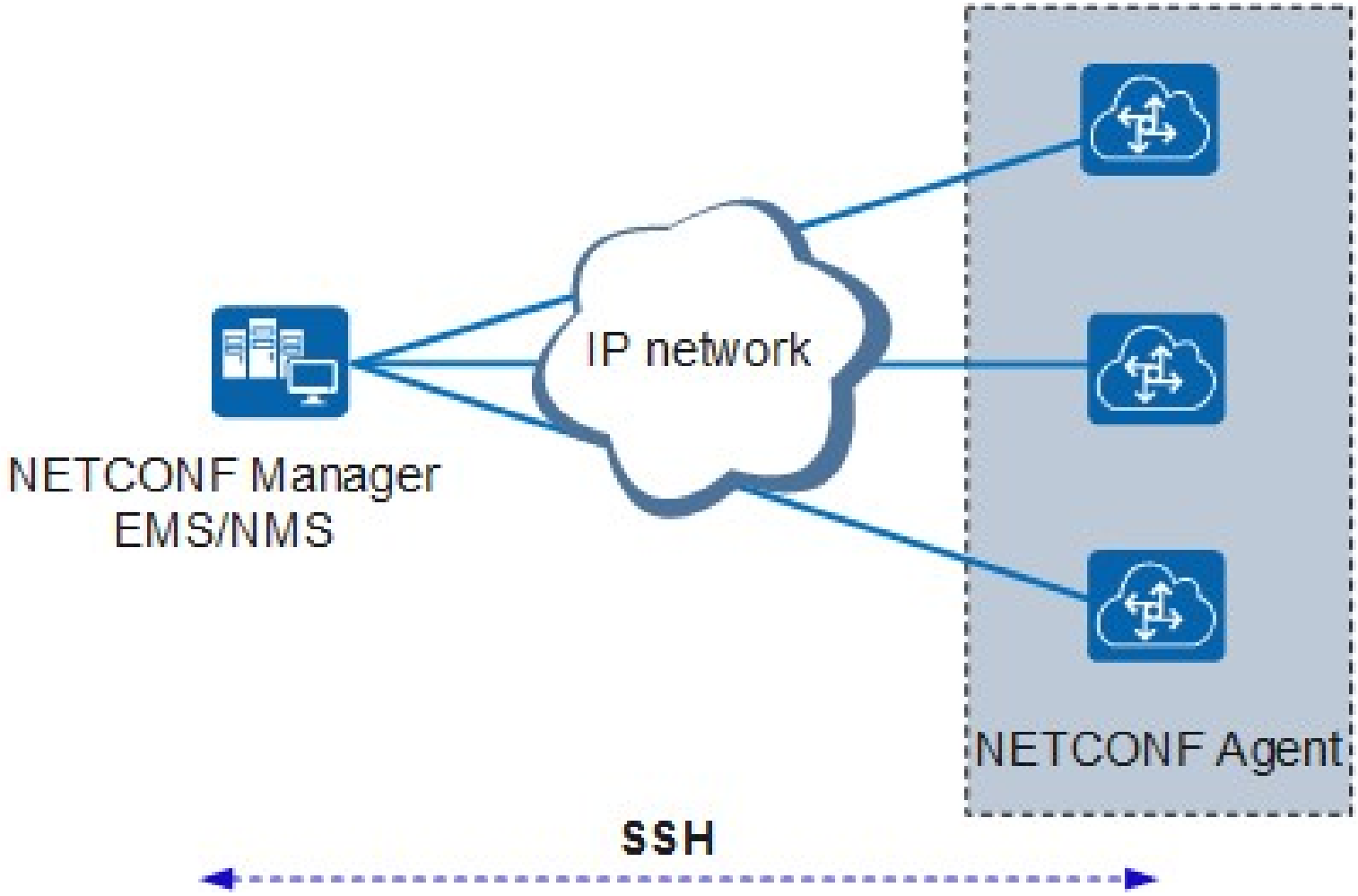
Représente les Configs à appliquer lors du prochain redémarrage

Opération <copy-config> pour copier la dernière sauvegarde de config

NETCONF Opérations de base

Opérations	Description
get	Récupérer les infos de configs à partir de la running database ou des statistiques
get-config	Récupérer les infos de configs à partir de la running database
edit-config	Modifier les configurations dans la database
copy-config	Copier les configurations
delete-config	Supprimer les configurations
commit	Commit du contenu de la config de <candidate/> ver <running/> database
lock	Bloquer l'écriture sur la database par d'autres sessions
unlock	Débloquer l'écriture sur la database par d'autres sessions
validate	Valider tout le contenu de la database
close-session	Fermer la session active
kill-session	Fermer d'autres sessions

Connexion SSH



Netconf messages **(Echanges RPC)**

- Une session NETCONF via SSH consiste à un échange, entre le manager et l'agent, de documents XML complets.
- Une fois la session établie et les capacités échangées, le manager envoie les documents XML contenant des éléments < rpc > au serveur, et l'agent répondra par les documents XML contenant l'élément < rpc-reply >.
- Ces < rpc > contiennent eux-mêmes des requêtes Netconf de différents types.

```
C: < ?xml version="1.0" encoding="UTF-8"? >
C: < rpc message-id="105"
C: xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
C: < get-config >
C: < source >< running/ >< /source >
C: < config xmlns="http://example.com/schema/1.2/config" >
C: < users/ >
C: < /config >
C: < /get-config >
C: < /rpc >
C: ]]>]]>
```

RPC client

```
S: < ?xml version="1.0" encoding="UTF-8"? >
S: < rpc-reply message-id="105"
S: xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
S: < config xmlns="http://example.com/schema/1.2/config" >
S: < users >
S: < user >< name >root< /name >< type >superuser< /type >< /user >
S: < user >< name >fred< /name >< type >admin< /type >< /user >
S: < user >< name >barney< /name >< type >admin< /type >< /user >
S: < /users >
S: < /config >
S: < /rpc-reply >
S: ]]>]]>
```

RPC serveur

YANG

(Yet Another Next Generation)

- Langage pour la modélisation des données
- Utilisé par NETCONF (couche content)
  - ✓Configuration data
  - ✓State data
- Description hiérarchique des données
- Interaction entre les modules et sous-modules

YANG

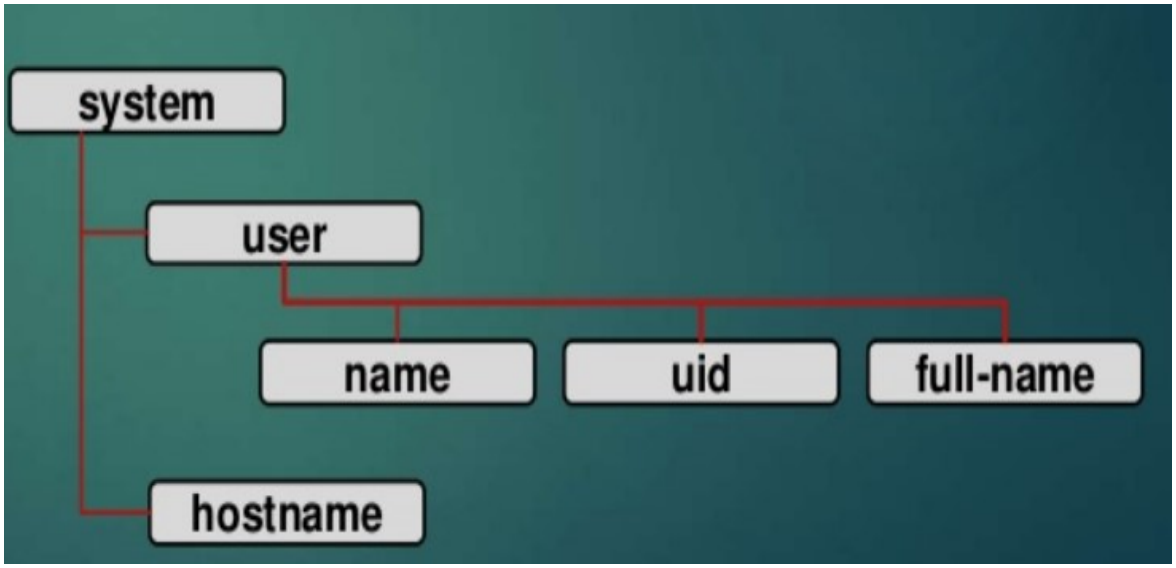
- Ces données sont représentées sous forme arborescente.
- Yang est modulaire, un module Yang pouvant se référer à d'autres modules.
- Yang définit un ensemble de types pour décrire les données.
- Il permet également d'indiquer les contraintes que doivent respecter les données.
- Une MIB peut être traduite en Yang, l'inverse n'étant pas vrai (Yang étant plus riche)

YANG: types de données et structures

YANG	Description
Module	Un fichier YANG complet
Data Type	String, Integer, Boolean, ... Pour YANG on peut définir son propre type en utilisant typedef
Structures	
Leaf	Une seule variable pouvant contenir une seule valeur
Leaf-list	Une collection de Leaf du même type de données
List	Une collection de paires clé / valeur, la clé est la Leaf et la valeur peut être n'importe quel type de données
Container	Hiérarchie de haut niveau qui regroupe Leaf, Leaf-list et list, et peut regrouper d'autres conteneurs

Exemples

```
container system {
  list user {
     key name;
     leaf name {
        type string;
     }
     leaf uid {
        type uint32;
     }
     leaf full-name {
        tyoe string;
     }
  leaf hostname{
     type string;
     mandatory true;
     config true;
}
```



```
module acme-system {
 namespace "http://acme.example.com/system";
 prefix "acme";
 organization "ACME Inc.";
 contact "joe@acme.example.com";
 description
 "The module for entities implementing the ACME system.";
 revision 2007-11-05 {
 description "Initial revision.";
 }
container system {
        leaf host-name {
                type string;
                description "Hostname for this system";
        }
        leaf-list domain-search {
                type string;
                description "List of domain names to search";
        }
        list interface {
                key "name";
                description "List of interfaces in the system";
                leaf name {
                        type string;
                }
                leaf type {
                        type string;
                }
                leaf mtu {
                        type int32;
                }
        }
        }
}
```