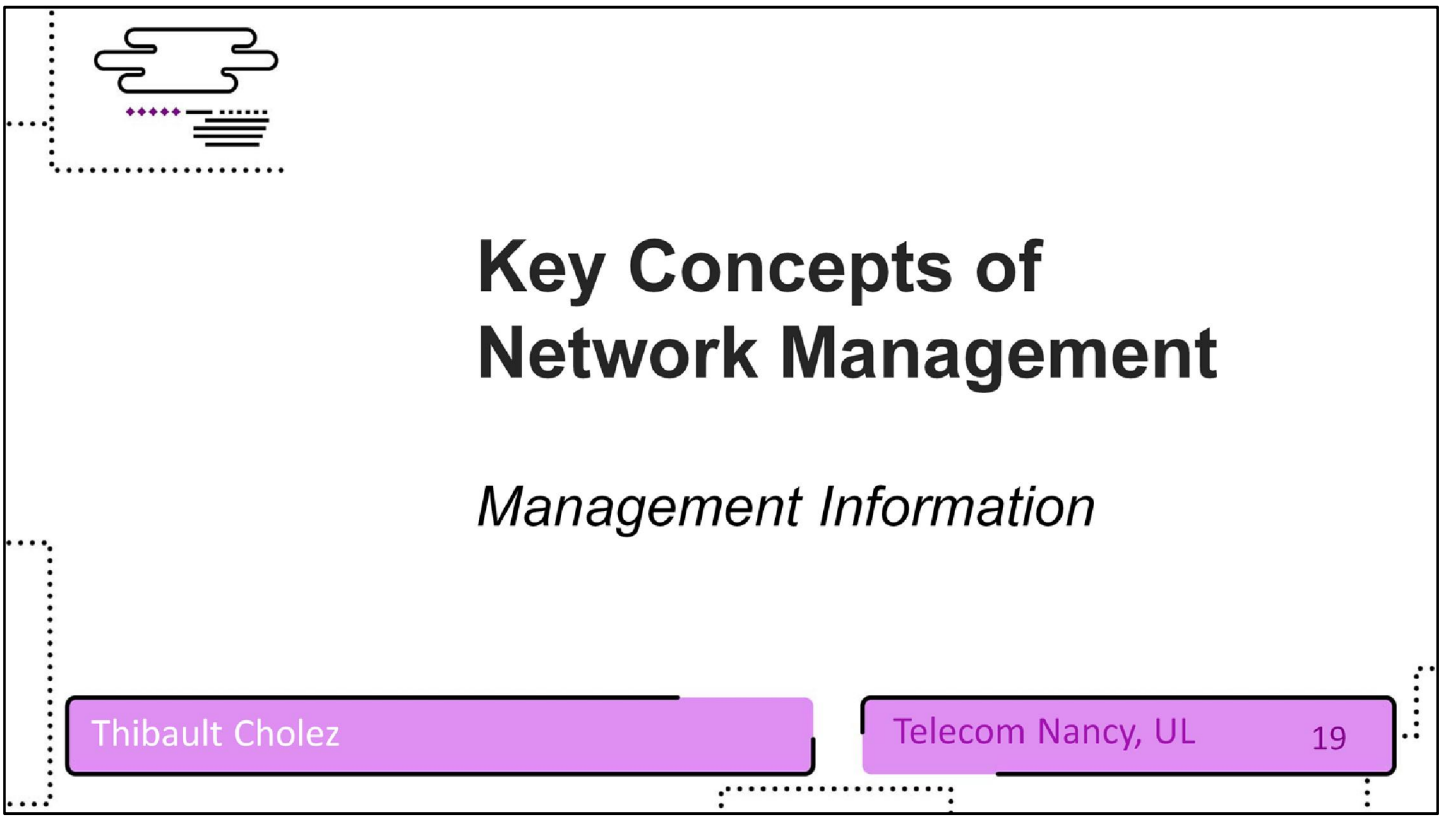
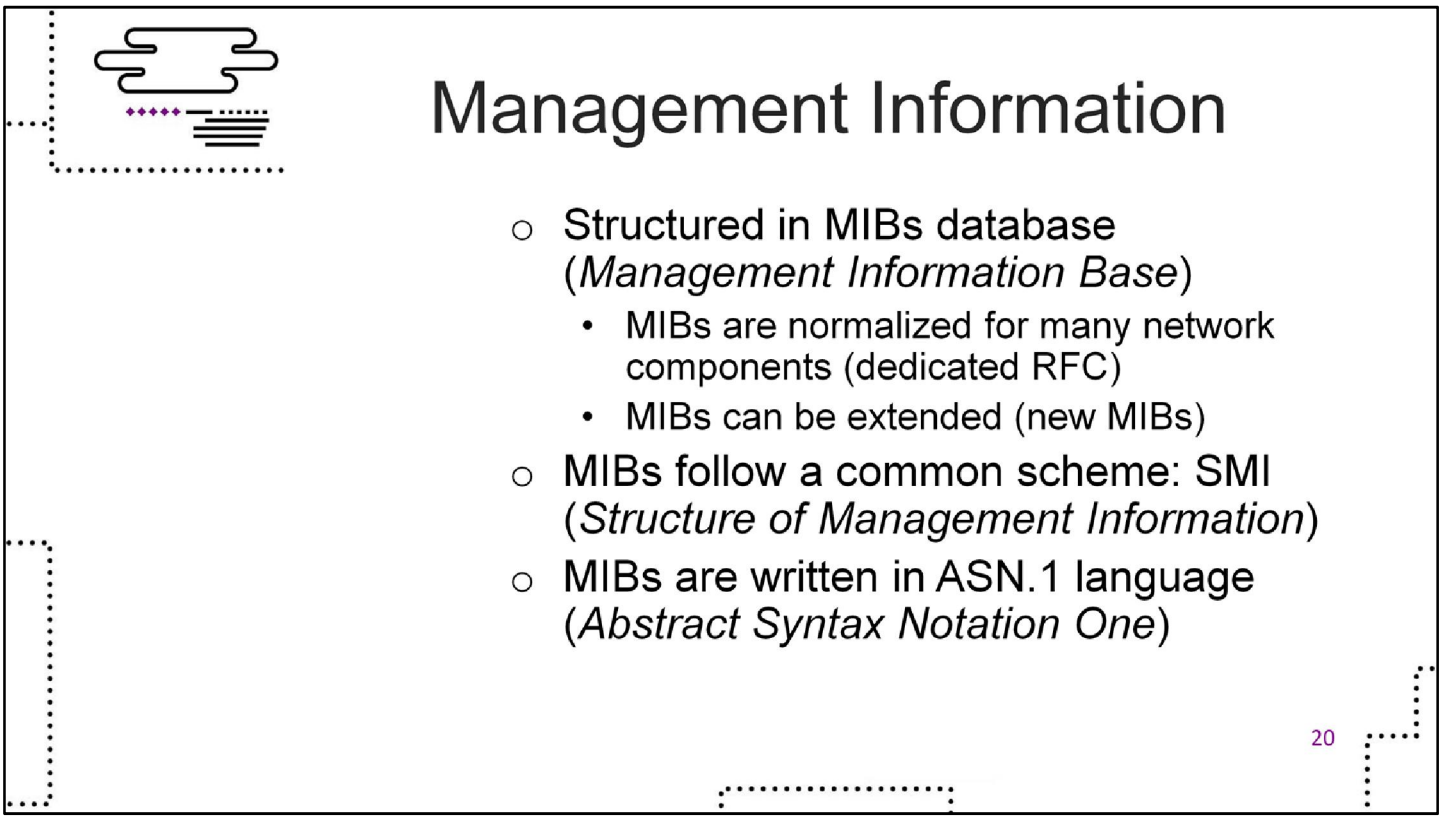
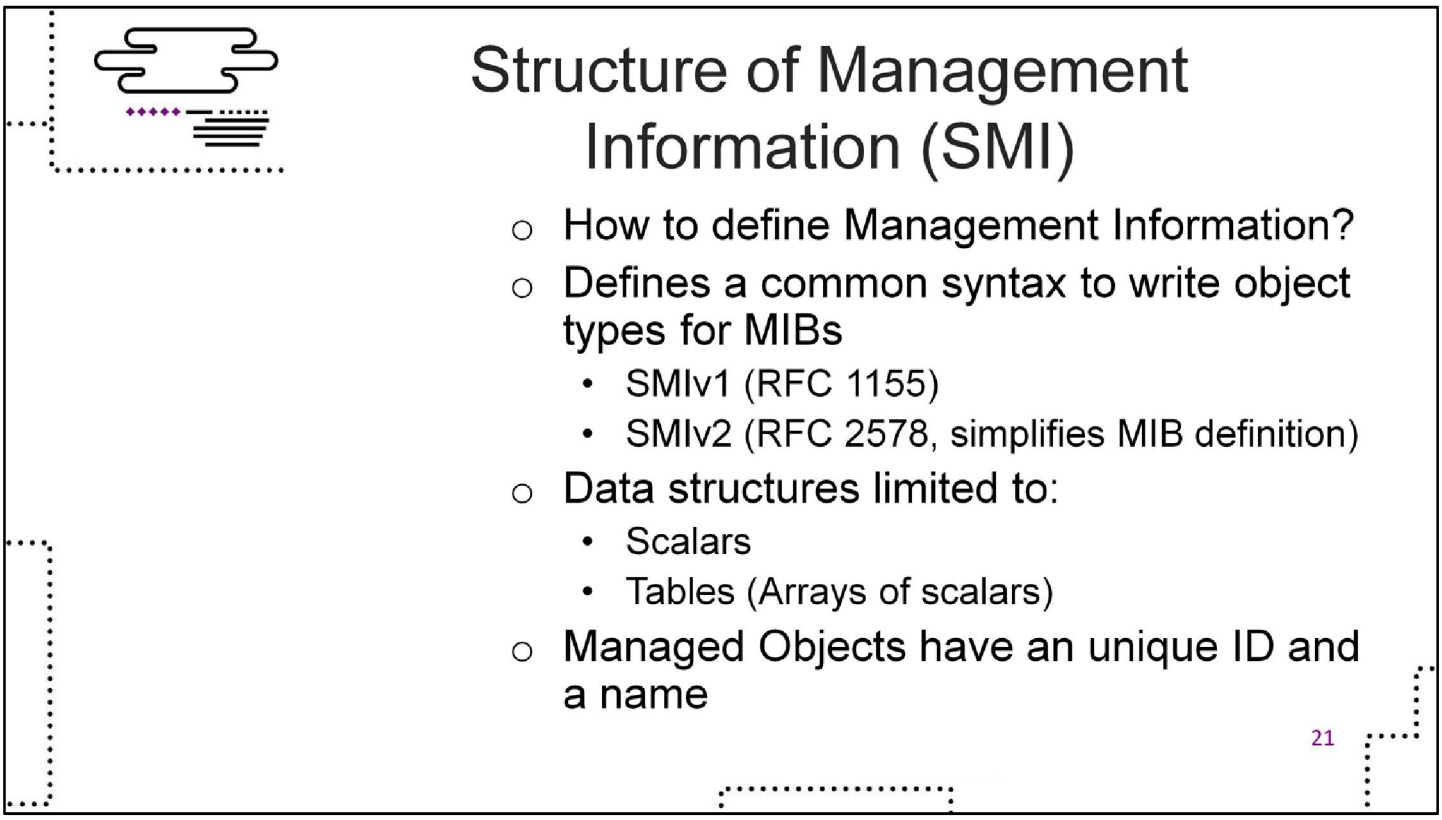


Nous allons voir dans cette leçon comment sont organisées les informations de gestion grâce à un modèle d'information spécifique.



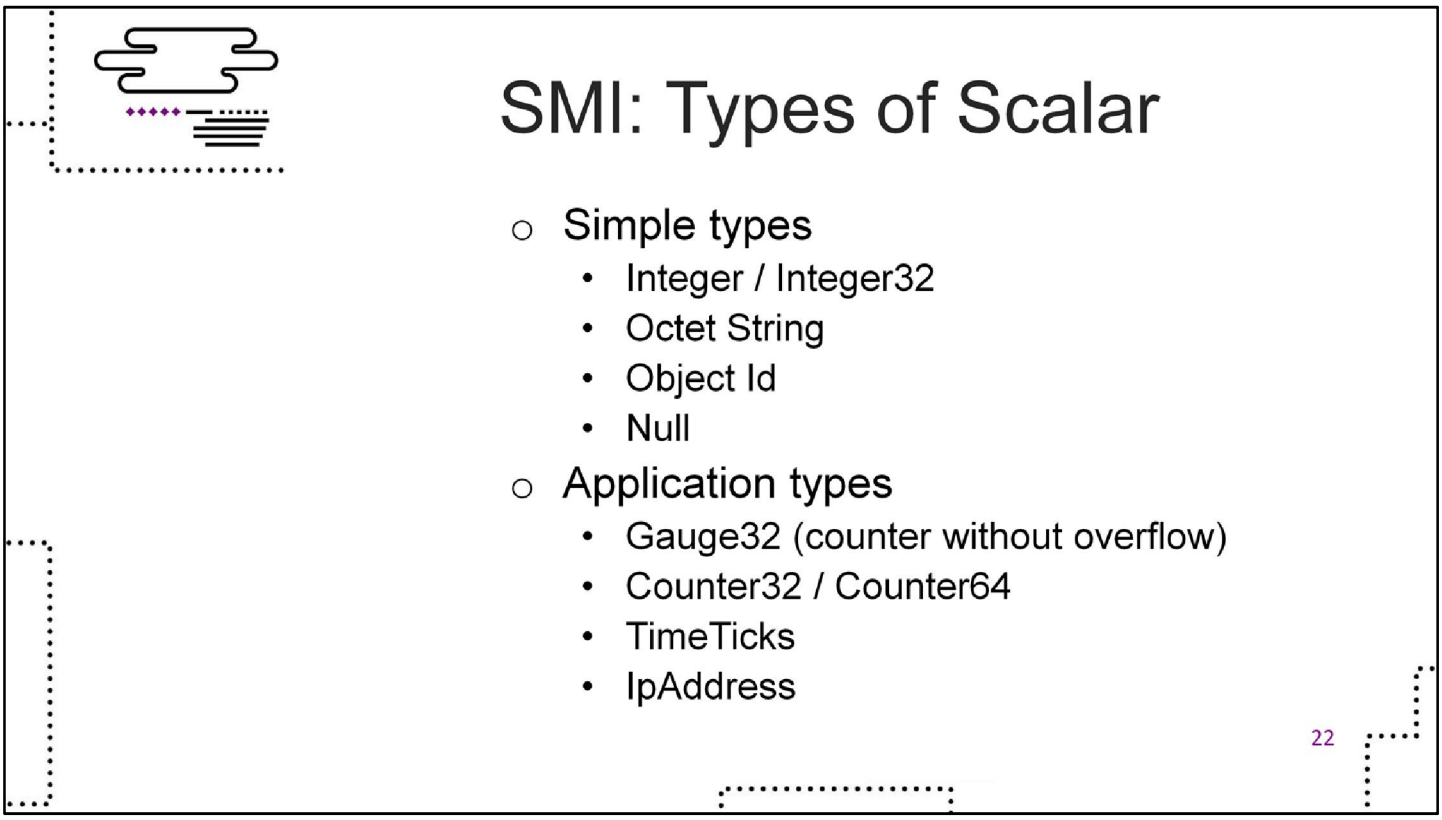
Les objets utilisés pour représenter une même ressource (par exemple : la pile TCP) doivent être identiques pour tous les systèmes afin de faciliter la gestion du réseau. Il y a donc besoin de définir quelles sont les informations qui caractérisent une pile TCP et qui doivent être exposées par les agents devant gérer ce protocole.

Ainsi, les informations relatives à la gestion sont structurées en bases de données spécifiques qui sont appelées MIB pour « Management Information Base ». Les MIBs sont spécifiques à un type d'objet géré et sont standardisées par l'IETF pour de nombreux équipements courants. Par exemple, le document de standardisation RFC 1759 définit une MIB pour une imprimante, RFC 1696 pour un modem, etc.
De nouvelles MIBs peuvent être créées pour répondre à de nouveaux besoins. Les MIBs respectent une structure commune appelée SMI pour « Structure of Management Information », et sont écrites dans un langage appelé ASN.1 pour « Abstract Syntax Notation One ».



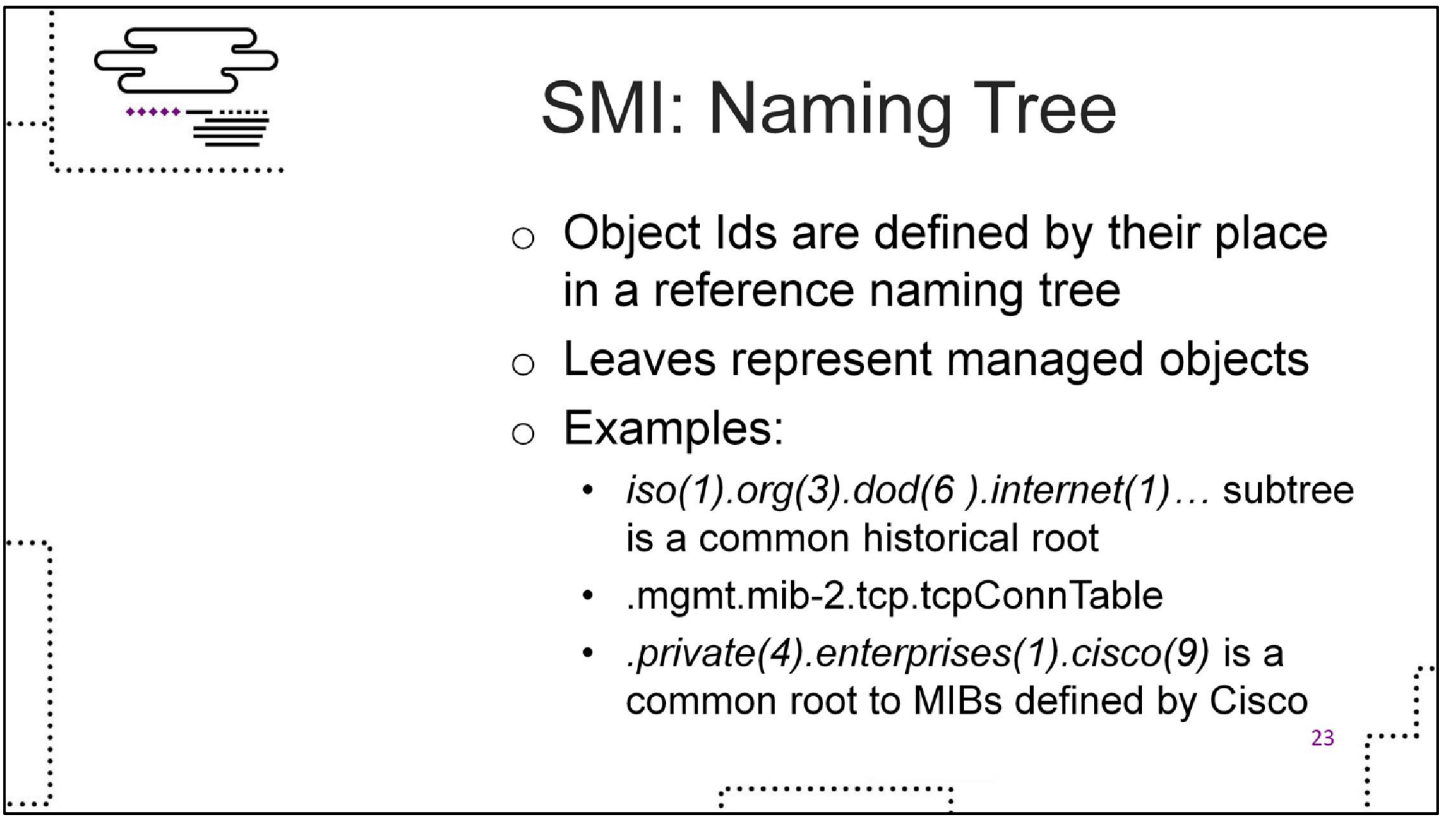
La structure des informations de gestions (SMI) fut initialement définie dans le document de standardisation RFC 1155 qui décrit SMIv1, avant d'être améliorée dans la version 2 qui simplifie notamment la définition des MIBs. Les contraintes imposées par SMI limitent les données pouvant être contenues dans les MIBs aux seules valeurs scalaires et aux tableaux (tableaux qui sont en réalité des listes de scalaires).

Les objets définis dans une MIB ont un nom complet et un identifiant uniques au sein d'un arbre de nommage global, ce qui permet d'identifier rapidement et sans ambiguïté n'importe quelle information de gestion.



SMI définit plusieurs types possibles pour les valeurs scalaires. Il y a tout d'abord les types simples qui sont présents par défaut dans le langage ASN.1. Ceux-ci comportent les entiers codés sur 16 ou 32 bits, les octets, les identifiants d'objet qui sont en réalité des séquences d'entiers et enfin le type null.

Il y a ensuite des types spécifiques à certaines applications comme les compteurs, qui sont des entiers positifs codés sur 32 ou 64 bits et dont la valeur ne peut pas décroître, les jauges qui sont des compteurs qui peuvent croître ou décroître, mais sans dépassement possible de la valeur maximale.
Les timeticks représentent le nombre de millisecondes écoulées depuis un temps de référence. Un exemple de type applicatif est, par exemple, le type IpAddress qui stocke une adresse IP sur 32 bits suivant le format consacré.

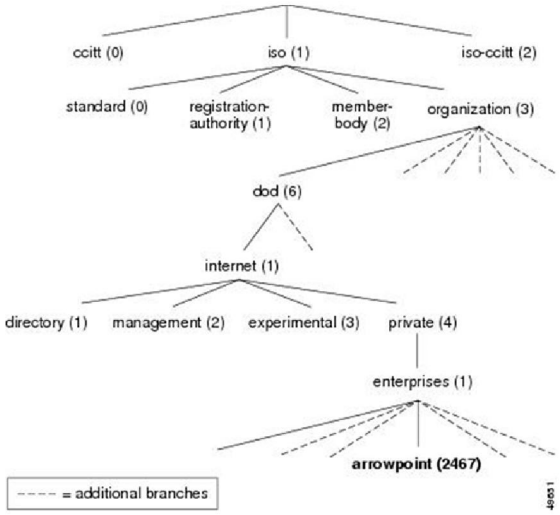


Les MIBs, et a fortiori, les éléments les composant sont tous numérotés et organisés au sein d'un schéma de nommage global ayant une forme arborescente. Les feuilles de cet arbre sont les objets de gestion.

L'arbre de nommage n'est pas du tout équilibré et certains sous-arbres spécifiques sont très utilisés. Le plus connu est celui identifié sous l'arborescence iso.org.dod.internet sous laquelle de nombreuses MIBs sont standardisées sous l'arborescence « .mgmt.mib-2 ».

Un autre exemple est l'arborescence .private.enterprises.cisco où sont définies les MIBs créées par le constructeur d'équipements réseaux CISCO.

SMI: Naming Tree



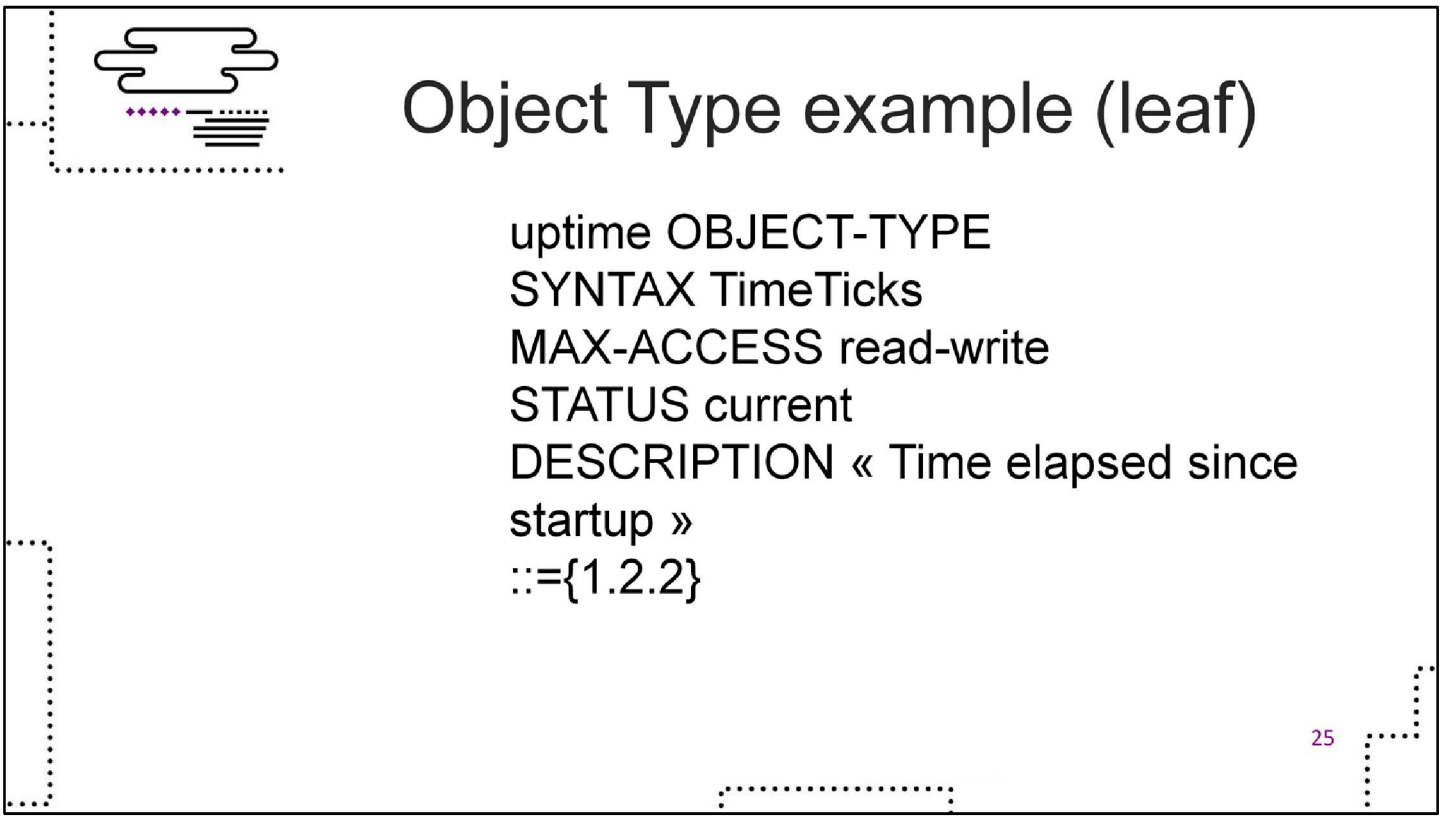
Source: cisco.com - Configuring Simple Network Management Protocol (SNMP)

24

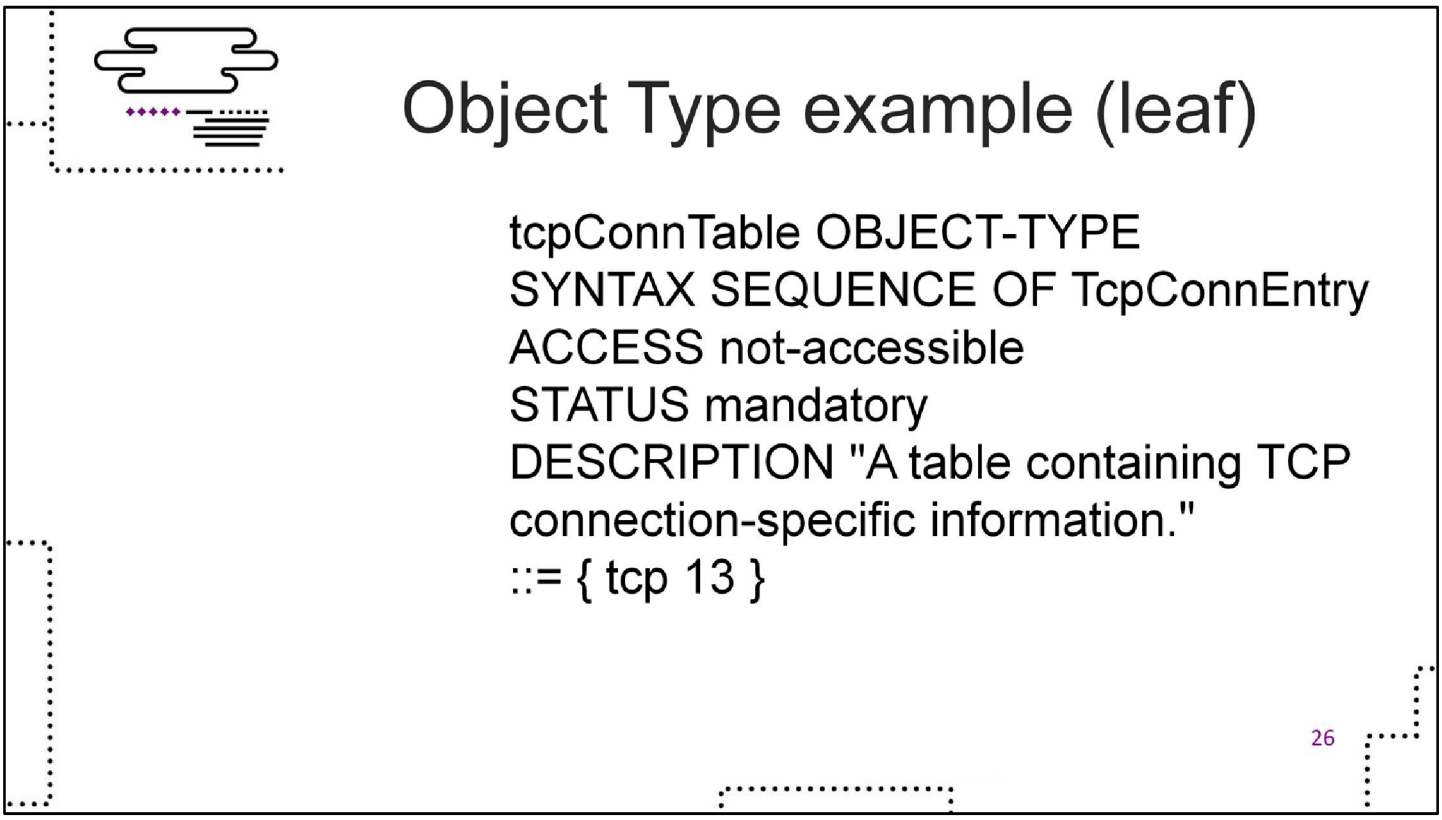
Ce schéma illustre cet arbre de nommage permettant d'identifier tout objet de gestion.

On peut y retrouver l'arborescence précédemment citée : 1.3.6.1 correspondant à « iso.organization.dod.internet », ainsi que les sous-arbres management (2) ou private (4) suivi d'enterprise (1).

Des informaticiens expérimentés dans le domaine de la gestion de réseaux ont l'habitude de manipuler ces références.

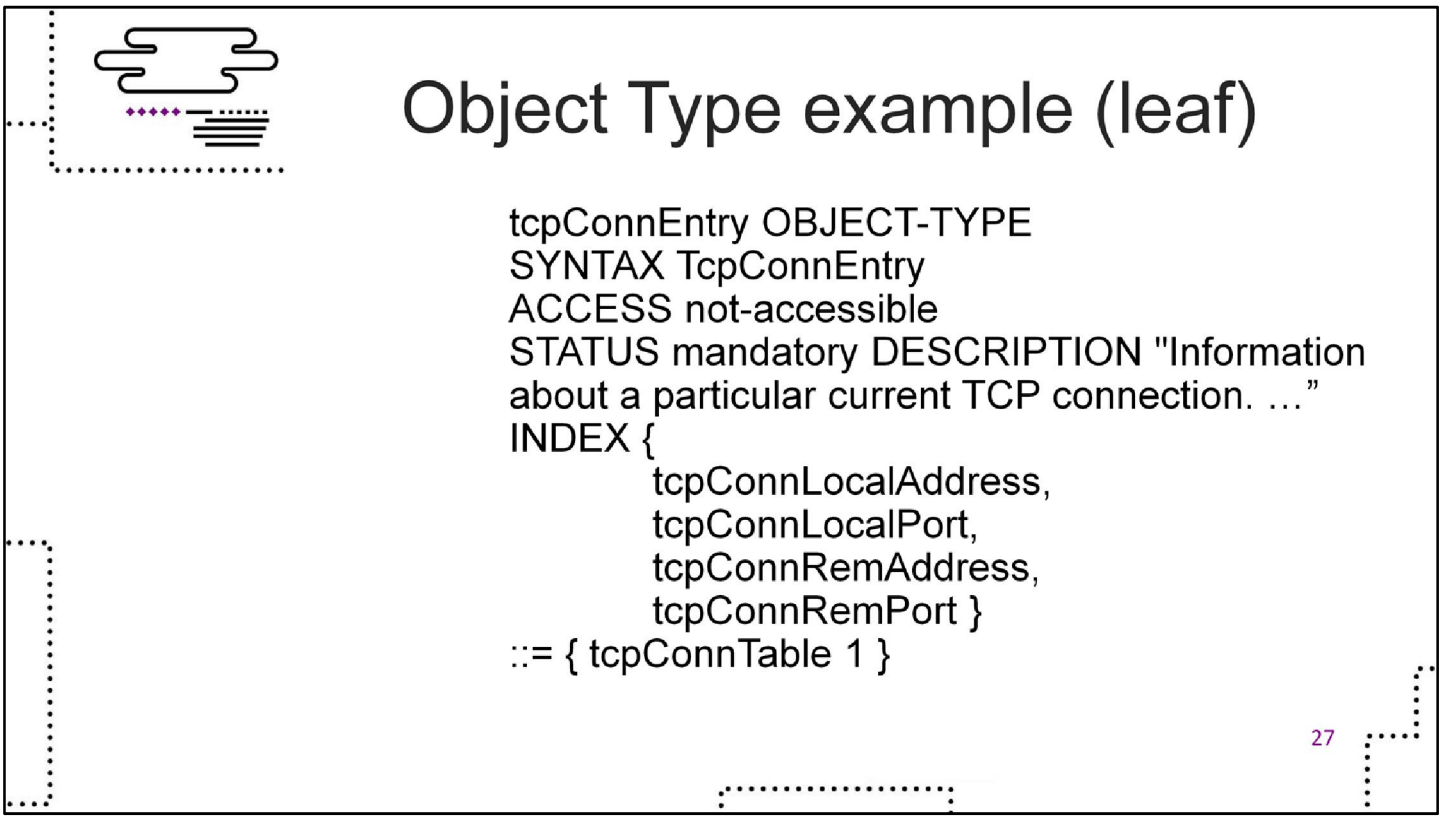


Voici maintenant un exemple de code définissant un objet de gestion, c'est-à-dire une feuille de l'arbre.
Le nom de l'objet (uptime) est suivi du mot clé OBJECT-TYPE. Suit le type de la donnée, ici on utilise le type applicatif TimeTicks, puis les modes d'accès, en l'occurrence en lecture et en écriture (read-write). Le mot clé STATUS indique la validité de cette définition d'objet car plusieurs anciennes versions peuvent co-exister. La DESCRIPTION est un champ textuel décrivant l'objet de gestion, ici « le temps écoulé depuis le démarrage ». Pour finir, l'identifiant de l'objet est donné de manière absolue ou relative ({1.2.2}).

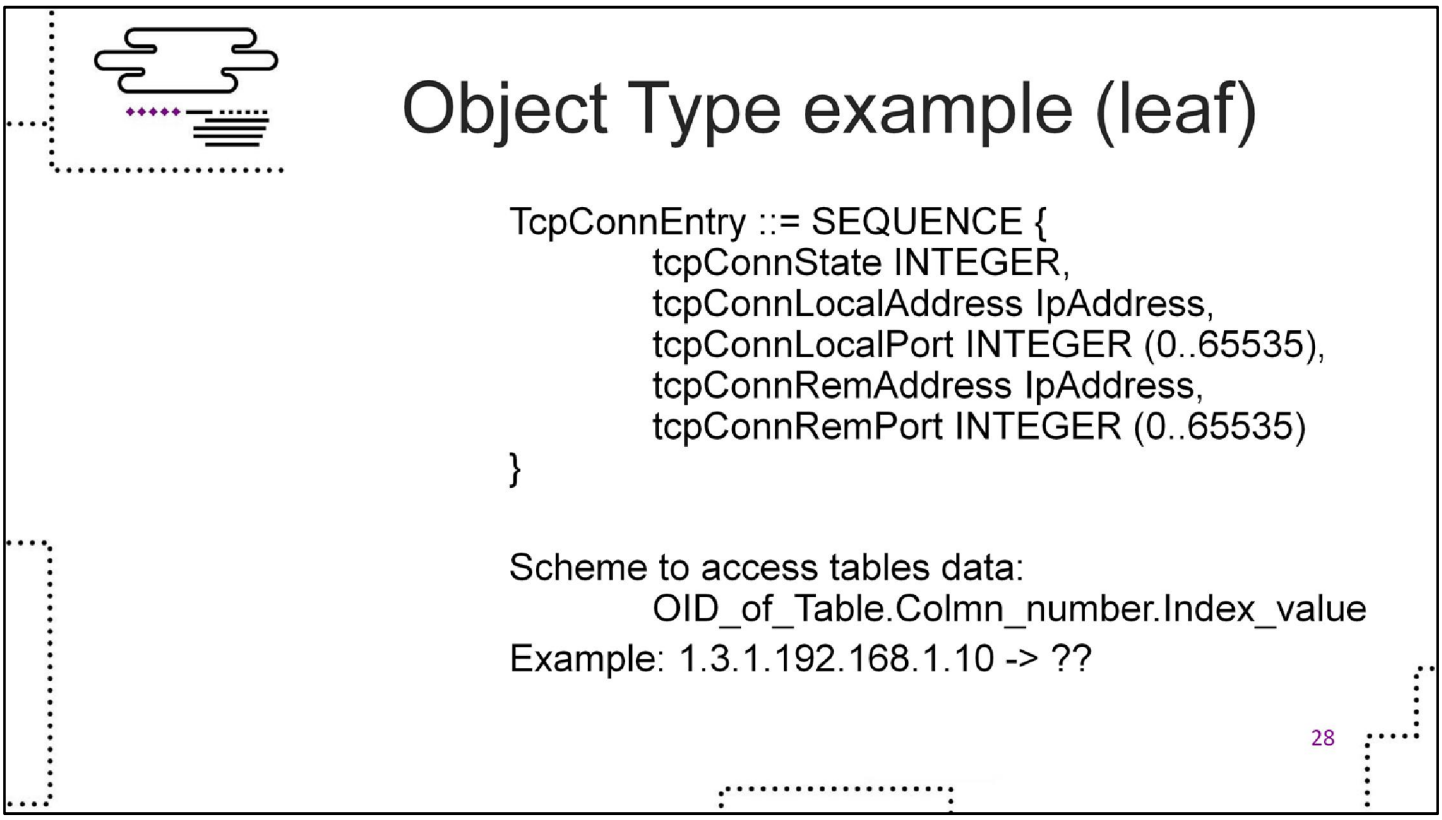


Voici un autre exemple de définition d'objet de gestion qui représente un tableau de connexions TCP.

Le mot clé « SEQUENCE OF » permet de définir une liste contenant entre 1 et N éléments d'un même type, en l'occurrence du type « TcpConnEntry » qui est lui-même défini dans le transparent suivant.

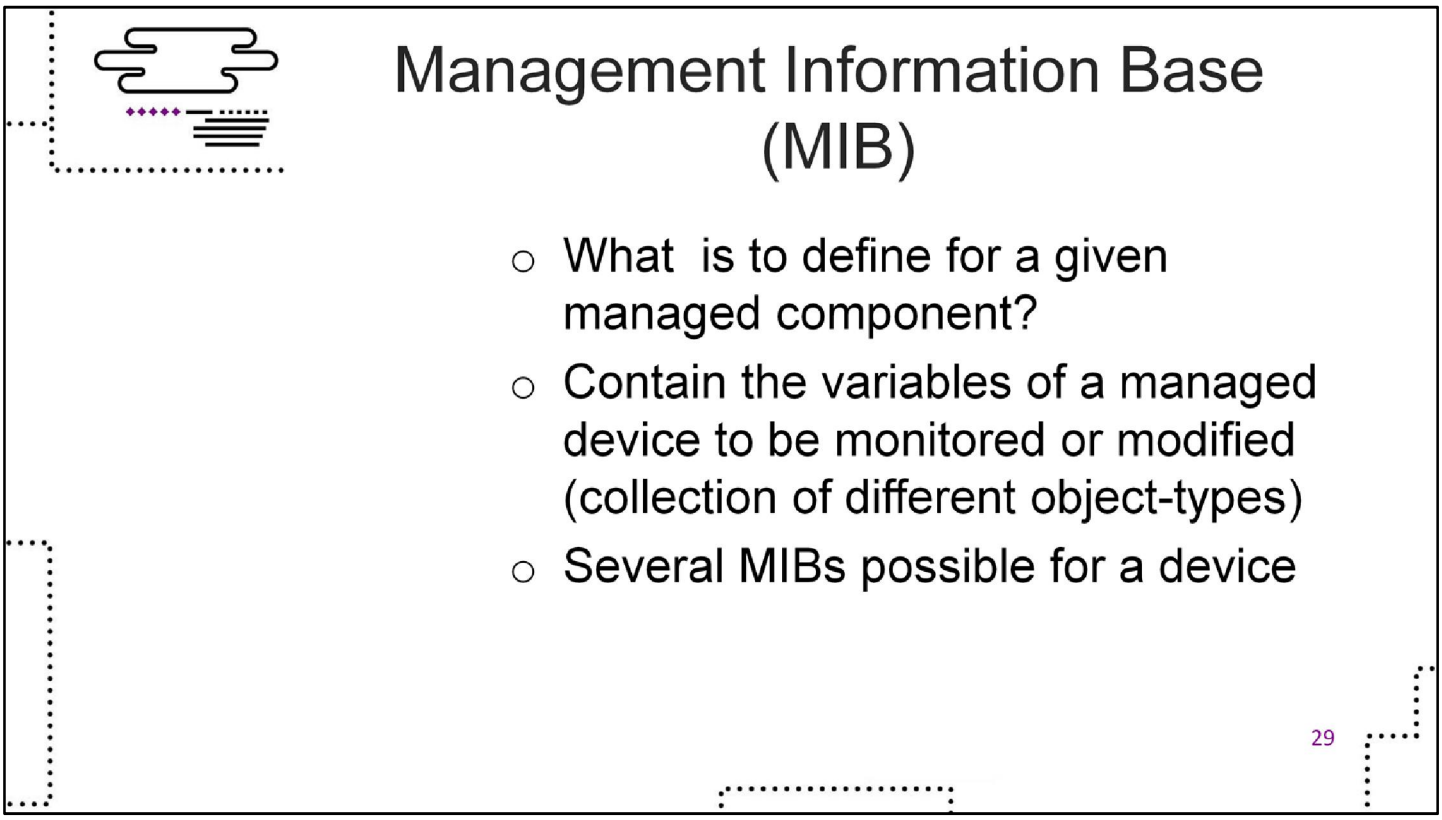


Dans cet exemple est défini le contenu d'une ligne stockant les informations d'une connexion TCP.
Le mot clé SYNTAX suivi du nom de l'objet avec une première lettre en majuscule fait référence au type.
Le mot clé INDEX définit la clé qui permet d'identifier les éléments stockés dans une ligne du tableau. On voit ici le nom des variables, mais pas leurs types. Pour cela, il faut se référer au type TcpConnEntry (avec un « T » majuscule).



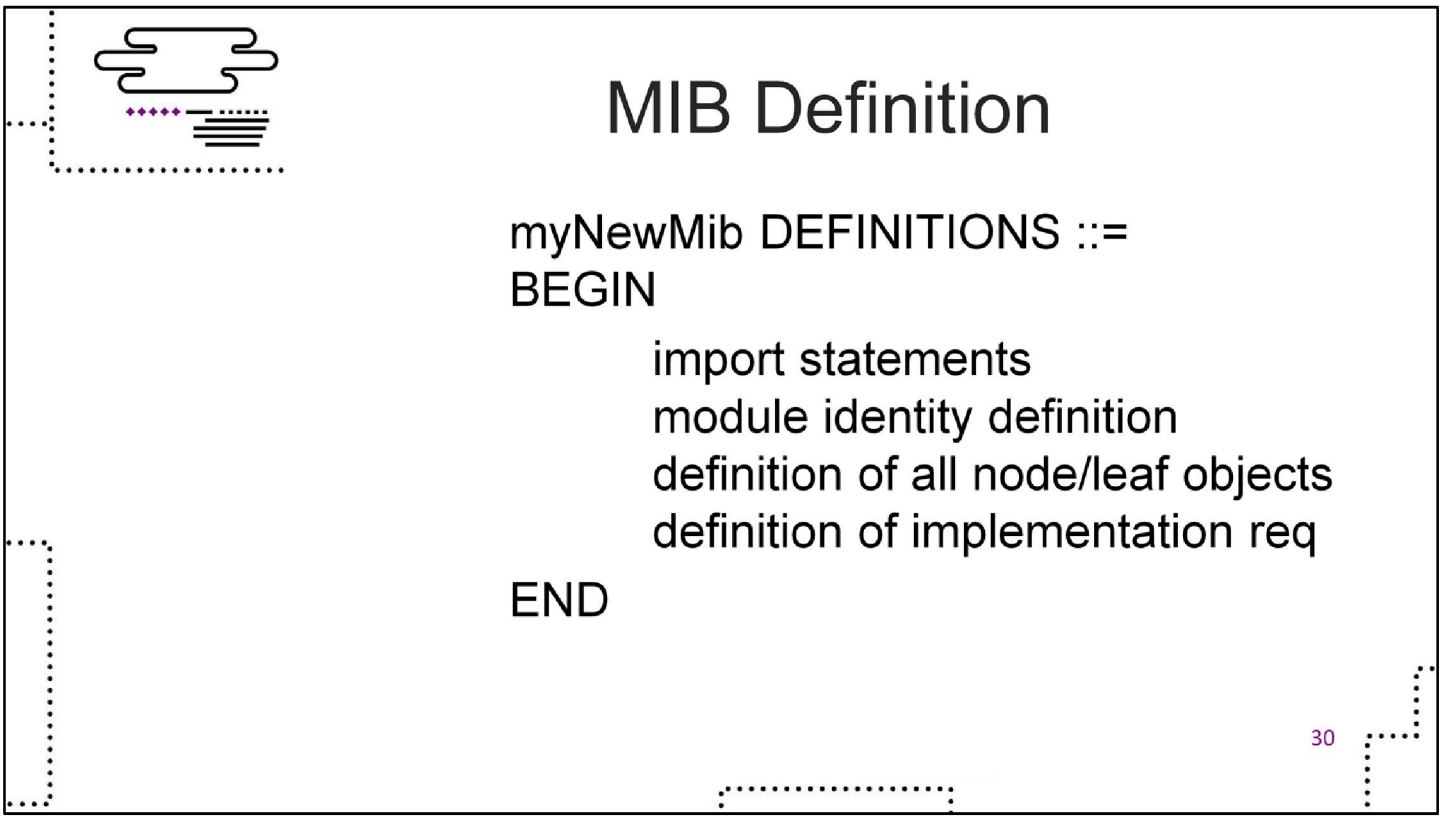
Le nouveau type TcpConnEntry est défini par ce code. Le mot clé SEQUENCE (sans le OF) fixe exactement le nombre d'éléments qui est défini entre les accolades. Pour accéder à un élément du tableau, il faut ensuite connaître son identifiant dans l'arbre, puis le numéro de la colonne, et enfin la valeur d'index qui correspond à une clé comprenant un ou plusieurs éléments de la ligne. Attention, ces éléments servant d'index ont, suivant leur type, des valeurs qui ne sont pas forcément des entiers comme on en a l'habitude en programmation.

Par exemple, on demande dans l'exemple « 1.3.1.192.168.1.10 » la valeur contenue dans la première colonne du tableau d'identifiant 1.3 avec pour identifiant de ligne « 192.168.1.10 ». Si plusieurs éléments sont retournés, on peut préciser d'avantage l'index (OID_of_Table.Colmn_number.Index_value1.Index_Value2).



Les MIBs sont donc au cœur de la gestion de réseaux en définissant et structurant les informations qui peuvent être manipulées.

Lorsque l'on souhaite pouvoir gérer un nouveau composant pour lequel aucune MIB adaptée n'existe, il convient de réfléchir aux variables et aux types de données représentant le mieux cet objet, sachant qu'il n'y a pas de solution unique et que plusieurs MIBs sont possibles pour représenter un composant.



Nous n'avons pas vu en détail la définition d'une MIB complète. Un exemple complet est trop volumineux pour être convenablement illustré ici, et sera vu directement en travaux pratiques. En attendant, voici la structure générale d'une définition de MIB. Celle-ci est composée de références à d'autres MIBs dont on peut réutiliser les types internes, d'une définition du module, puis des différents nœuds et feuilles constituant la MIB, pour terminer par les contraintes d'implémentation.