



Comprendre la programmabilité des réseaux et les modèles de dispositifs

8.3.1

Qu'est-ce que la programmabilité pilotée par un modèle ?



La programmabilité est la capacité d'un appareil à accepter des instructions pour contrôler ou modifier le comportement. Pour les périphériques réseau, cela signifie souvent être configuré et géré à l'aide de protocoles logiciels. Contrairement aux interfaces orientées vers l'homme telles que l'interface de ligne de commande et les interfaces graphiques, les interfaces programmables sont explicitement conçues pour être consommées par les machines.

La programmabilité pilotée par un modèle hérite de la puissance des modèles, en faisant correspondre les capacités et les services d'un appareil à des modèles standardisés. Il est ainsi plus facile de configurer les périphériques réseau et de surmonter les inconvénients posés par les techniques traditionnelles de gestion des périphériques réseau.

Qu' est-ce qu'un modèle de données?

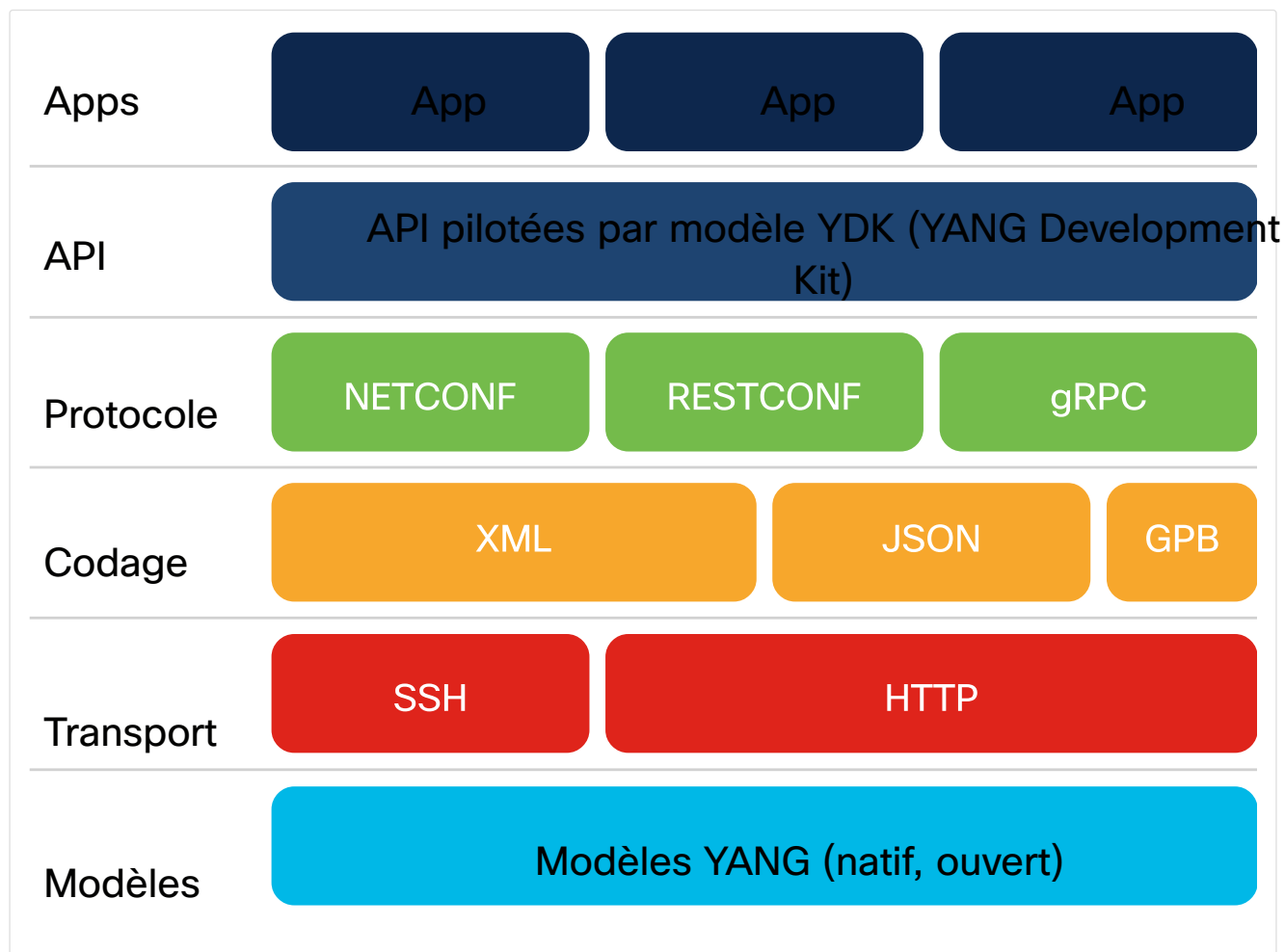
En général, un modèle de données est une méthode structurée pour décrire n'importe quel objet, qu'il s'agisse d'une personne, d'une voiture, d'un bâtiment ou d'un autre objet. Par exemple, les données personnelles figurant sur un passeport ou un permis de conduire peuvent décrire une personne d'une manière individuelle, de sorte que ce sont les deux "modèles de données".

En réseau, les "modèles de données" sont une manière programmatique et basée sur des normes d'écriture de configurations pour n'importe quel périphérique réseau. Il peut remplacer la configuration manuelle, implémentant YANG comme langage de modélisation de données de facto.

Dans le monde actuel des réseaux, un ensemble de modèles de données est basé sur YANG et utilise le langage de modélisation YANG. YANG version 1 \ [RFC6020] ou YANG version 1.1 \ [RFC7950] sont utilisées pour spécifier la configuration et l'état de fonctionnement pris en charge par un périphérique.

La pile de programmabilité basée sur le modèle YANG ressemble à ceci:

Pile de programmation pilotée par un modèle



Avantages de la programmabilité basée sur un modèle

Traditionnellement, les CLI ou les fonctions de bibliothèque SDK sont utilisées pour configurer les périphériques réseau. Les CLI et les scripts sont presque toujours spécifiques à l'appareil ; ils ne peuvent être utilisés que sur le même type de périphériques qui utilisent les mêmes CLI du même fournisseur. De plus, ces fonctions peuvent ne pas exister pour toutes les commandes de configuration, et lorsqu'elles existent, il peut être difficile de programmer une configuration complexe.

Cependant, la programmation basée sur un modèle fournit un moyen standard de décrire la configuration souhaitée des périphériques réseau.

Bien sûr, l'équipement cible lui-même doit prendre en charge la configuration pilotée par le modèle. Pour les modèles basés sur YANG, il doit prendre en charge YANG et comprendre les protocoles de niveau supérieur basés sur YANG, tels que NETCONF et RESTCONF.

Les configurations basées sur le modèle doivent toujours être fournies au périphérique réseau. La livraison de modèle peut être encapsulée dans une API ou gérée dans un wrapper SDK.

En résumé, la programmabilité pilotée par les modèles :

- Fournit un langage de configuration lisible par l'homme
- est basé sur un modèle, structuré et convivial pour l'ordinateur

- Prise en charge de plusieurs types de modèles, y compris natif, OpenConfig et IETF
- Utilise la spécification qui est découplée du transport, encodage de fin de protocole
- Utilise des API basées sur des modèles pour l'abstraction et la simplification
- Exploite l'open source et bénéficie d'une large prise en charge

Afin de comprendre la programmabilité basée sur un modèle, nous devons comprendre ses composantes clés:

- YANG
- NETCONF
- RESTCONF

YANG est un langage de modélisation. NETCONF et RESTCONF sont des protocoles utilisés pour les interfaces programmables des modèles de données.

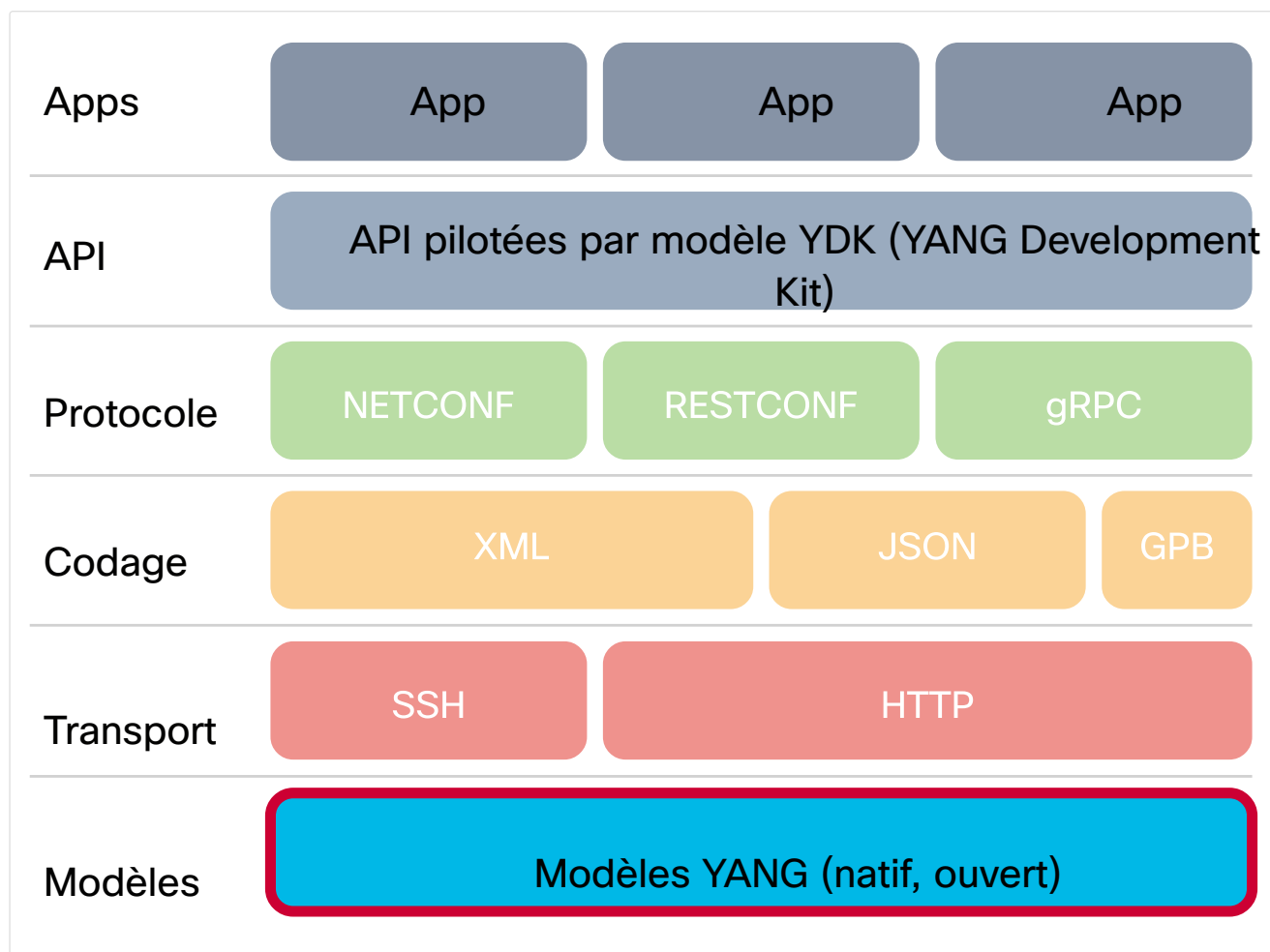
8.3.2

Qu' est-ce que YANG?



YANG, un acronyme de Yet Another Next Generation, tel que défini dans RFC7519, est "un langage de modélisation de données utilisé pour modéliser les données de configuration et d'état manipulées par le Network Configuration Protocol (NETCONF), les appels de procédure à distance NETCONF et les notifications NETCONF."

YANG dans la pile de programmabilité pilotée par modèle



Un module YANG définit des hiérarchies de données pouvant être utilisées pour les opérations NetConf, y compris la configuration, les données d'état, les RPC et les notifications. Cela permet une description complète de toutes les données envoyées entre un client NETCONF et un serveur. YANG peut également être utilisé avec des protocoles autres que NETCONF.

Bien que YANG puisse décrire n'importe quel modèle de données, il a été initialement conçu pour les modèles de données réseau.

Dans le monde réel, il existe deux types de modèles YANG, ouverts et natifs.

- **Open YANG Models:** Développé par des fournisseurs et des organismes de normalisation, tels que l'IETF, l'UIT, OpenConfig, etc. Ils sont conçus pour être indépendants de la plate-forme sous-jacente et normaliser la configuration par fournisseur des périphériques réseau.
- **Native Models:** Développé par des fournisseurs tels que Cisco. Ils se rapportent et sont conçus pour intégrer des fonctionnalités ou des configurations qui ne concernent que cette plate-forme.

Pourquoi nous avons besoin de YANG pour la modélisation d'appareils?

YANG fournit une norme tout en permettant une description plus détaillée. Nous avons besoin d'une méthode standard pour modéliser la configuration des périphériques réseau. YANG permet aux différents fournisseurs de périphériques réseau de décrire leur type de périphérique, leur configuration et leur état pour les mapper au fonctionnement de l'appareil de manière programmatique.

Les termes utilisés dans YANG sont définis à la section 3 de la RFC6020. Voici quelques termes

couramment utilisés à comprendre avant de plonger dans YANG:

- **anyxml**: Nœud de données pouvant contenir un morceau inconnu de données XML.
- **augment**: ajoute de nouveaux nœuds de schéma à un nœud de schéma précédemment défini.
- **container**: nœud de données intérieur qui existe dans, au plus, une instance de l'arborescence de données. Un conteneur n'a pas de valeur, mais plutôt un ensemble de nœuds enfants.
- **data model**: Un modèle de données décrit la façon dont les données sont représentées et accessibles.
- **data node**: nœud dans l'arborescence du schéma qui peut être instancié dans une arborescence de données. L'un des conteneurs, leaf, leaf-list, list et anyxml.
- **data tree**: arborescence instanciée des données de configuration et d'état sur un périphérique.
- **derived type**: type dérivé d'un type intégré (tel que uint32) ou d'un autre type dérivé.
- **grouping**: ensemble réutilisable de nœuds de schéma. Le regroupement peut être utilisé localement dans le module, dans les modules qui l'incluent, et par d'autres modules qui en importent. L'instruction grouping n'est pas une instruction de définition de données et, en tant que telle, ne définit aucun nœud dans l'arborescence du schéma.
- **identifier**: Utilisé pour identifier différents types d'éléments YANG par nom.
- **leaf**: nœud de données qui existe dans au plus une instance dans l'arborescence de données. Une feuille a une valeur mais aucun nœud enfant.
- **leaf-list**: Comme le nœud leaf, mais définit un ensemble de nœuds identifiables de manière unique plutôt qu'un seul nœud. Chaque nœud a une valeur, mais aucun nœud enfant.
- **list**: nœud de données intérieur qui peut exister dans plusieurs instances dans l'arborescence de données. Une liste n'a pas de valeur, mais plutôt un ensemble de nœuds enfants.
- **module**: Un module YANG définit une hiérarchie de nœuds pouvant être utilisés pour les opérations basées sur NetConf. Avec ses définitions et les définitions qu'il importe ou inclut d'ailleurs, un module est autonome et "compilable".
- **RPC**: appel de procédure distante, tel qu'il est utilisé dans le protocole NETCONF.
- **state data**: données supplémentaires sur un système qui ne sont pas des données de configuration, telles que les informations d'état en lecture seule et les statistiques collectées \ [RFC4741].

YANG définit quatre types de nœuds pour la modélisation des données. Le détail est décrit à la section 4.2.2 de la RFC6020 ou à la section 4.2.2 de la RFC 7950.

- Leaf Nodes
- Leaf-List Nodes
- Container Nodes
- List Nodes

Un module YANG contient une séquence d'instructions. Chaque instruction commence par un mot-clé, suivi de zéro ou d'un argument, suivi d'un point-virgule (";") ou d'un bloc de sous-déclarations entre accolades ("{" "}").

```
statement = keyword [argument] (";" / "{" *statement "}");
```

Il y a quatre déclarations principales dans un module YANG:

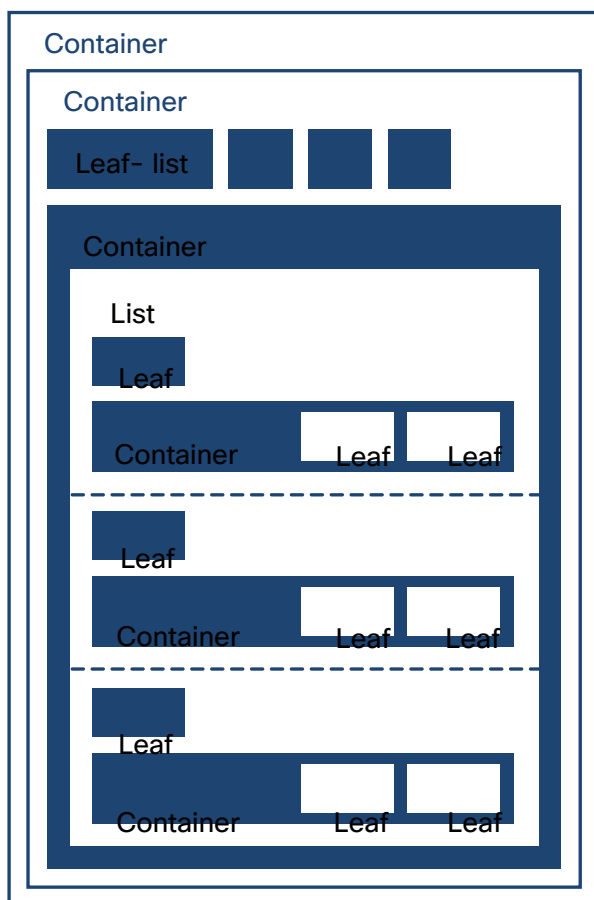
- **Container**: Un regroupement d'autres instructions (n'a pas de "Valeur").

- **List**: plusieurs enregistrements contenant au moins une "clé" de feuille et une hiérarchie arbitraire d'autres instructions
- **Leaf**: une seule paire clé/valeur.
- **Leaf-list**: paire clé/valeurs multiples du même type.

Construction de modules YANG

Les modules prennent généralement racine à partir d'un ou de plusieurs conteneurs, et l'arborescence du schéma s'étend à un autre niveau de structures de données. Une branche se termine par des feuilles (leaves) et/ou des listes de feuilles (leaf-list).

Structure du module YANG



YANG en action

Examinons maintenant un exemple de fichier YANG dans la vie réelle, `ietf-interfaces.yang`. Ce module ouvert YANG contient une collection de définitions YANG pour la gestion des interfaces réseau.

Quand vous connaissez la terminologie et la structure de YANG, il n'est pas difficile de comprendre le contenu d'un fichier YANG, qui contient des commentaires et des descriptions très détaillés. Mais ces descriptions rendent également le fichier très long. Heureusement, il existe des outils pour extraire le contenu de manière plus lisible et concise, et l'outil `pyang` en est l'un.

Vous pouvez installer `pyang` en utilisant la commande `pip` dans un environnement virtuel.

Comme vous pouvez le voir ci-dessous, l'utilisation de l'outil `pyang` peut convertir un fichier YANG en une structure arborescente facile à suivre.

1. Créez un fichier nommé `ietf-interfaces.yang` à l'aide d'un éditeur de texte local.
2. Copiez un exemple de fichier YANG à partir de GitHub. Par exemple, vous pouvez accéder à <https://github.com/YangModels/yang/blob/master/standard/ietf/RFC/ietf-interfaces%402018-02-20.yang> pour un modèle ouvert, ou <https://github.com/YangModels/yang/blob/master/vendor/cisco/xr/602/ietf-interfaces.yang> pour un modèle natif spécifique au fournisseur.
3. Cliquez sur le bouton **Raw** et copiez le texte de la page dans le fichier local `ietf-interfaces.yang` dans un éditeur de texte.
4. Enregistrez localement le fichier `ietf-interfaces.yang` dans lequel vous avez installé `pyang`.
5. Dans le répertoire où vous avez stocké le fichier YANG, exécutez la commande `pyang -f tree ietf-interfaces.yang` pour générer une liste *tree-formatted* des ressources ietf-interface.

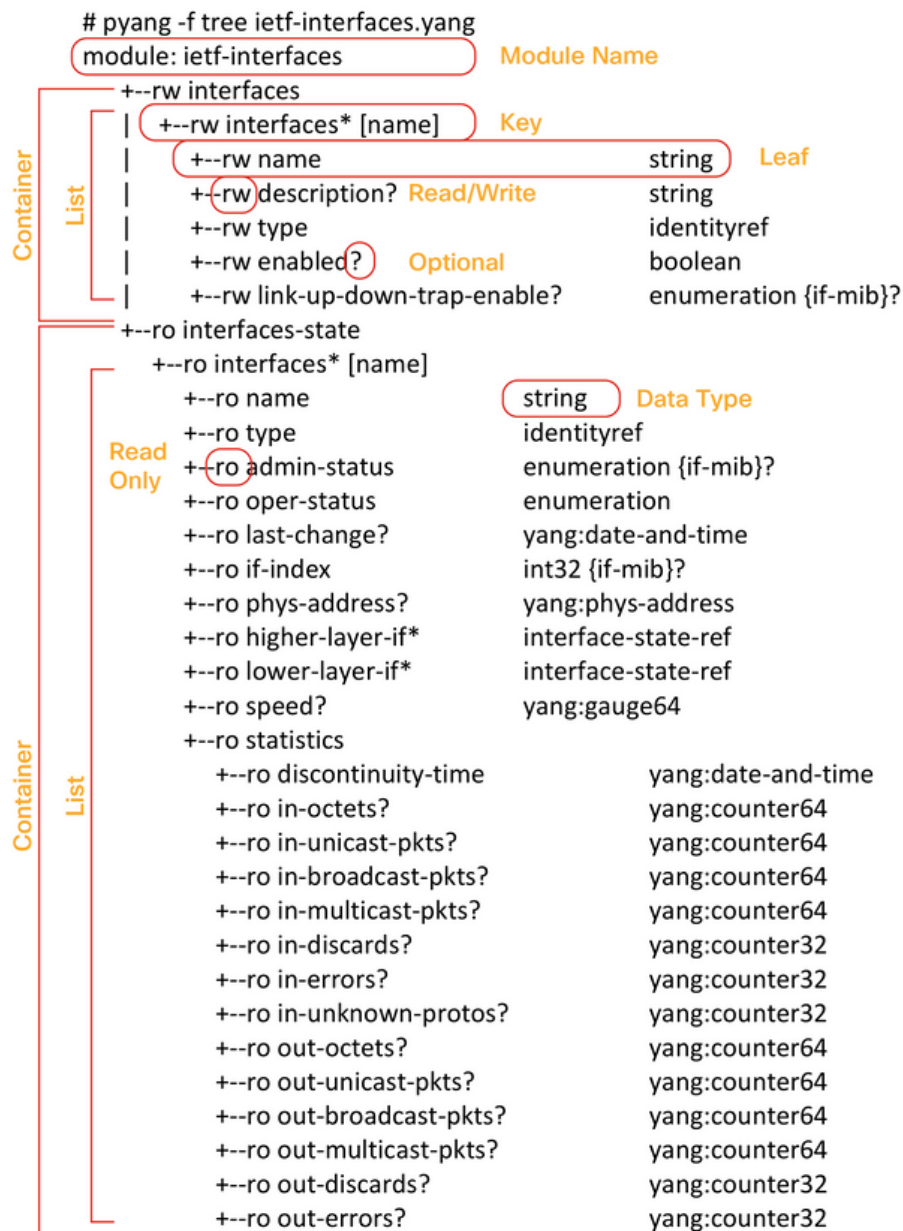
Exemple de sortie Pyang

```
module: ietf-interfaces
  +--rw interfaces
  |   +--rw interface* [name]
  |       +--rw name                string
  |       +--rw description?        string
  |       +--rw type                 identityref
  |       +--rw enabled?             boolean
  |       +--rw link-up-down-trap-enable? enumeration {if-mib}?
  +--ro interfaces-state
      +--ro interface* [name]
          +--ro name                string
          +--ro type                 identityref
          +--ro admin-status         enumeration {if-mib}?
          +--ro oper-status          enumeration
          +--ro last-change?         yang:date-and-time
          +--ro if-index             int32 {if-mib}?
          +--ro phys-address?        yang:phys-address
          +--ro higher-layer-if*     interface-state-ref
          +--ro lower-layer-if*      interface-state-ref
          +--ro speed?               yang:gauge64
          +--ro statistics
              +--ro discontinuity-time yang:date-and-time
              +--ro in-octets?         yang:counter64
              +--ro in-unicast-pkts?   yang:counter64
              +--ro in-broadcast-pkts? yang:counter64
              +--ro in-multicast-pkts? yang:counter64
              +--ro in-discards?       yang:counter32
              +--ro in-errors?         yang:counter32
              +--ro in-unknown-protos? yang:counter32
              +--ro out-octets?        yang:counter64
              +--ro out-unicast-pkts?  yang:counter64
```

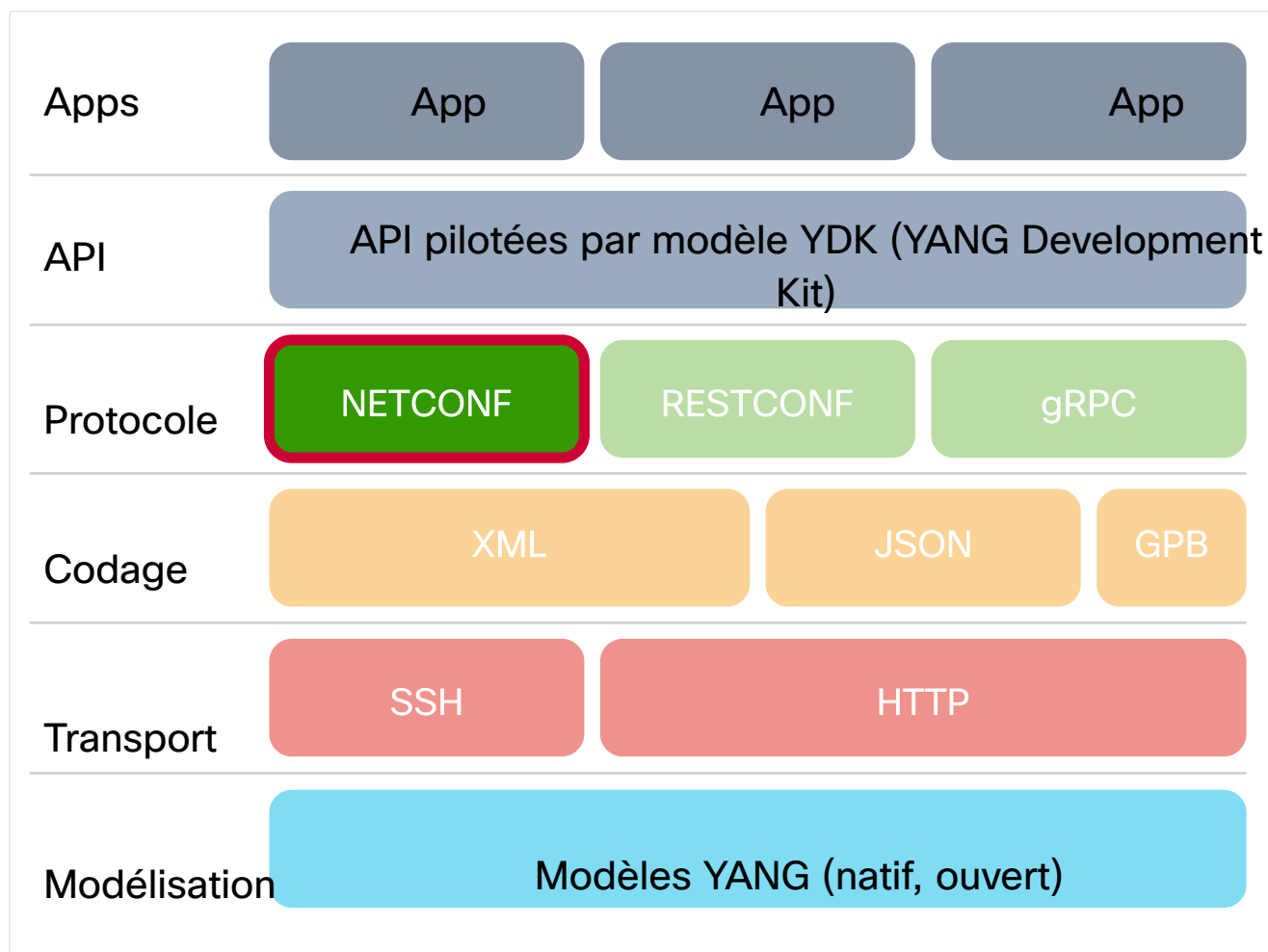
```

+--ro out-broadcast-pkts? yang:counter64
+--ro out-multicast-pkts? yang:counter64
+--ro out-discards?      yang:counter32
+--ro out-errors?        yang:counter32

```



Depuis que YANG a été utilisé pour la première fois avec NETCONF, regardez de plus près NETCONF et voyez comment les deux se rapportent.



Le protocole NETCONF utilise le codage de données XML pour les données de configuration et les messages de protocole.

Le protocole NETCONF fournit un petit ensemble d'opérations pour gérer les configurations des périphériques et récupérer les informations sur l'état des périphériques. Le protocole de base fournit des opérations de récupération, configuration, copie et suppression des stockages de données de configuration.

Opérations du protocole NETCONF

Le protocole NETCONF fournit un ensemble d'opérations pour gérer les configurations des périphériques et récupérer les informations sur l'état des périphériques. Le protocole de base comprend les opérations de protocole suivantes:

Operation	Description
get	récupérer les informations de configuration en cours d'exécution et d'état du périphérique.
get-config	récupérer tout ou partie d'une configuration spécifiée.
edit-config	modifier la configuration d'un appareil.
copy-config	créer ou remplacer une banque de données de configuration complète par une autre banque de données de configuration complète.
delete-config	supprimer une configuration dans un magasin de données.

Operation	Description
lock	verrouiller l'ensemble du système de banque de données de configuration d'un périphérique.
déverrouiller	libérer un verrou de configuration, précédemment obtenu avec l'opération lock .
close-session	demander la fin gracieuse d'une session NETCONF.
kill-session	forcer la fin d'une session NETCONF.

NETCONF contre SNMP

Les protocoles NETCONF et SNMP sont tous deux définis pour configurer à distance les périphériques.

Comparaison des caractéristiques

SNMP:

- Utilise le modèle pull lors de la récupération de données à partir d'un périphérique qui ne s'évolutionne pas bien pour les plates-formes haute densité
- Ne dispose pas d'un processus de découverte permettant de rechercher la base d'informations de gestion (MIB) prise en charge par un périphérique
- Ne supporte pas le concept de transactions
- Manque de sauvegarde et de restauration de la configuration des éléments
- Support limité de l'industrie pour les MIB de configuration

NETCONF a été conçu pour permettre:

- Stockages de données de configuration multiples (candidat, en cours d'exécution, démarrage)
- Transactions au niveau de l'appareil et à l'échelle du réseau
- Test et validation de la configuration
- Distinction entre les données de configuration et d'exploitation
- Récupération sélective des données avec filtrage
- Diffusion et lecture des notifications d'événements
- Appels de procédure distante extensible
- Échange de capacités intégrées

Le NETCONF RFC dispose de trois magasins de données distincts qui sont la cible des lectures et des écritures de configuration. Les incitations sont les suivantes:

- en cours d'exécution (obligatoire)
- candidat (facultatif)
- démarrage (facultatif)

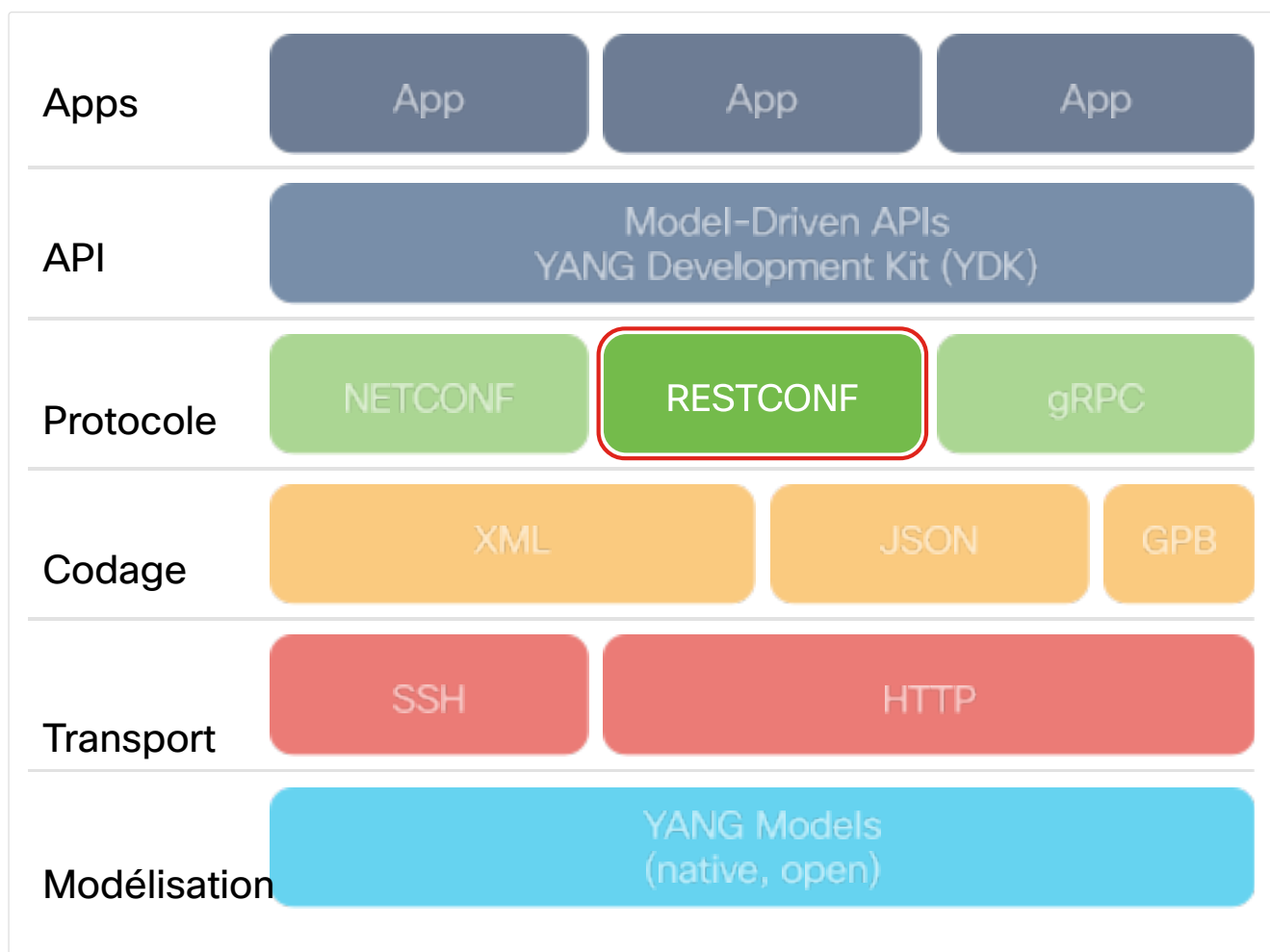
Fonctionnalités NETCONF par rapport à SNMP

Exemple d'utilisation	NETCONF	SNMP
-----------------------	---------	------

Exemple d'utilisation	NETCONF	SNMP
Récupère la collection des champs d'état	Oui	Oui
Définir la collection des champs de configuration	Oui	Oui
Définir les champs de configuration dans la transaction	Oui	Non
Transactions sur plusieurs éléments de réseau	Oui	Non
Envoyer des notifications d'événements	Oui	Oui
Protocole sécurisé	Oui	Oui
Test de la configuration avant la validation finale	Oui	Non

Le RESTCONF RFC 8040 définit un protocole et un mécanisme pour un accès de type REST aux informations de configuration et au contrôle. Similaire à NETCONF, il utilise les modèles de banque de données et les verbes de commande définis dans le protocole NETCONF (Network Configuration Protocol), encapsulés dans les messages HTTP. Comme avec NETCONF, le langage YANG est utilisé pour définir la syntaxe de la structure de commande, en tant que sémantique de la configuration, de l'état et des événements de banque de données de configuration.

RESTCONF dans la pile de programmabilité pilotée par modèle



RESTCONF utilise des données structurées (XML ou JSON) et YANG pour fournir des API de type REST, permettant un accès programmatique aux périphériques. Les commandes HTTP GET, POST, PUT, PATCH et DELETE sont dirigées vers une API RESTCONF pour accéder aux ressources de données représentées par les modèles de données YANG. Ces opérations de modification de base permettent d'afficher et de modifier la configuration en cours d'exécution par un client RESTCONF.

RESTCONF n'est pas destiné à remplacer NETCONF, mais plutôt à fournir une interface HTTP qui respecte les principes REST et est compatible avec le modèle de banque de données NETCONF. Un périphérique réseau peut servir à la fois NETCONF et RESTCONF simultanément, ou peut fournir seulement l'un ou l'autre.

RESTCONF vs NETCONF

Dans l'ensemble, NETCONF est plus complet, flexible et complexe que RESTCONF. RESTCONF est plus facile à apprendre et à utiliser pour les ingénieurs ayant une expérience antérieure de l'API REST. Voici les différences entre NETCONF et RESTCONF:

- NETCONF prend en charge les stockages de données en cours d'exécution et candidats, tandis que RESTCONF ne prend en charge qu'une banque de données en cours d'exécution, car toutes les modifications du magasin de données candidat sont immédiatement validées.
- RESTCONF ne prend pas en charge l'obtention ou la libération d'un verrou de stockage de données. Si une banque de données a un verrou actif, l'opération de modification RESTCONF échoue.
- Une modification RESTCONF est une transaction limitée à un seul appel RESTCONF.
- RESTCONF ne prend pas en charge les transactions sur plusieurs appareils.
- La validation est implicite dans chaque opération d'édition RESTCONF, qui réussit ou échoue.

Mappage des opérations NETCONF et des méthodes RESTCONF

Description	NETCONF	RESTCONF
Créer une ressource de données	<code><edit-config></code> , <code></edit-config></code>	POST
Récupérer les données et les métadonnées	<code><get-config></code> , <code><get></code> , <code></get-config></code>	GET
Créer ou remplacer une ressource de données	<code><edit-config></code> (nc:operation="create/replace")	PUT
Supprimer une ressource de données	<code><edit-config></code> (nc:operation="delete")	SUPPRIMER

Chaque serveur RESTCONF de périphérique est accessible via des méthodes API. Où pouvez-vous trouver l'API RESTCONF? La réponse est que les API de périphérique individuelles sont rarement publiées. Au lieu de cela, les URL de la méthode sont déterminées dynamiquement.

Le RESTCONF RFC 8040 indique que la syntaxe URI de base RESTCONF est `/restconf/<resource-type>/<yang-module:resource>`. `<resource-type>` et `<yang-module:resource>` sont des variables et les valeurs sont obtenues à l'aide de fichiers de modèles YANG spécifiques.

Le format de base d'une URL RESTCONF est `https://<hostURL>/restconf<resource><container><leaf><options>` where any portion after `restconf` could be omitted.

Une requête GET initiale avec seulement la partie `restconf` devrait renvoyer une réponse avec les ressources disponibles sur ce serveur. Par exemple, un GET construit comme suit:

```
GET /restconf HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

Retournerait quelque chose de similaire à:

```
HTTP/1.1 200 OK
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
Content-Type: application/yang-data+json
{
  "ietf-restconf:restconf" : {
    "data" : {},
    "operations" : {},
    "yang-library-version" : "2016-06-21"
  }
}
```

Cette réponse nous indique que `data` et `operations` sont des types de ressources RESTCONF pris en charge par ce périphérique.

Remarque: Vous apprendrez comment utiliser RESTCONF dans le laboratoire, Utiliser RESTCONF pour accéder à un périphérique IOS XE.

8.3.5


Travaux pratiques – Explorez les modèles YANG



Dans ce TP, vous apprendrez à utiliser l'outil pyang open source pour transformer les modèles de données YANG à partir de fichiers utilisant le langage YANG, en un format beaucoup plus facile à lire. En utilisant la transformation de vue "tree", vous identifierez les éléments clés du modèle YANG ietf-interfaces.

Vous atteindrez les objectifs suivants:

- Partie 1: Lancez les machines virtuelles DEVASC et CSR1000V
- Partie 2: Explorez un modèle YANG sur GitHub
- Partie 3: Explorez un modèle YANG en utilisant pyang

 Explorer les modèles YANG

8.3.6

Travaux pratiques – Utiliser NETCONF pour accéder à un périphérique IOS XE



Dans ce TP, vous utiliserez un client NETCONF, ncclient, qui est un module Python pour les scripts côté client. Vous utiliserez ncclient pour vérifier que NETCONF est configuré, récupérer une configuration de périphérique et modifier une configuration de périphérique.

Vous atteindrez les objectifs suivants:

- Partie 1: Création du réseau et vérification de la connectivité
- Partie 2: Utiliser une session NETCONF pour recueillir des informations
- Partie 3: Utiliser ncclient pour se connecter à NETCONF
- Partie 4: Utiliser ncclient pour récupérer la configuration
- Partie 5: Utiliser ncclient pour configurer un périphérique
- Partie 6: Défi: Modifier le programme utilisé dans ce laboratoire

 Utiliser NETCONF pour accéder à un périphérique...

8.3.7


Travaux pratiques – Utiliser RESTCONF pour accéder à un périphérique IOS XE



Dans la première moitié de ce laboratoire, vous utiliserez le programme Postman pour construire et envoyer des requêtes API au service RESTCONF qui s'exécute sur R1. Dans la seconde moitié du laboratoire, vous allez créer des scripts Python pour effectuer les mêmes tâches que votre programme Postman.

Vous atteindrez les objectifs suivants:

- Partie 1: Création du réseau et vérification de la connectivité
- Partie 2: Configurer un périphérique IOS XE pour l'accès RESTCONF
- Partie 3: Ouvrir et configurer Postman
- Partie 4: Utiliser le facteur pour envoyer des requêtes GET
- Partie 5: Utiliser Postman pour envoyer une demande de PUT
- Partie 6: Utiliser un script Python pour envoyer des requêtes GET
- Partie 7: Utiliser un script Python pour envoyer une requête PUT

 Utiliser RESTCONF pour accéder à un périphériqu...

 8.2
Les SDK Cisco

8.4
Gestion des réseaux Cisco 