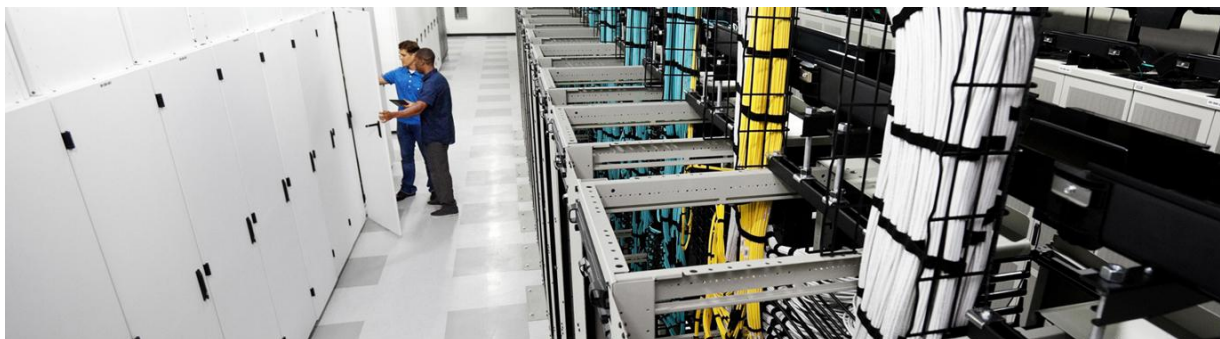


# Programmabilité du réseau avec l'infrastructure axée sur les applications (ACI) de Cisco



## Présentation

Ce document examine la prise en charge de la programmabilité sur l'infrastructure axée sur les applications (ACI) de Cisco®. Le modèle de programmabilité de l'ACI Cisco autorise un accès en programmation complet à cette infrastructure. L'ACI Cisco fournit un accès en lecture et en écriture, via des API REST (Representational State Transfer, transfert d'état représentatif), au modèle d'objet sous-jacent, lequel représente tous les attributs logiques et physiques du système. Grâce à cet accès, les clients peuvent intégrer le déploiement réseau dans leurs outils de gestion et de surveillance et déployer de nouvelles charges de travail au moyen de programmes.

## Les difficultés causées par les approches actuelles de la programmabilité réseau

La plupart des réseaux actuels reposent sur des équipements matériels dotés de logiciels étroitement liés et administrés au moyen d'une interface de ligne de commande. Ces systèmes étaient performants lorsque les configurations réseau et les charges de travail étaient statiques, et que le taux de renouvellement des applications était prévisible et plus lent. Alors que les réseaux de data centers ont été virtualisés et ont commencé leur transition vers le cloud et des modèles IT flexibles, ce modèle n'a plus cours.

Désormais, les fournisseurs entendent proposer plusieurs couches de capacité de programmation dans leurs offres et systèmes d'exploitation existants. Si cette approche permet de gagner en capacité, elle n'est toutefois pas idéale en termes d'intégration de la programmabilité. De fait, elle génère de la complexité en introduisant un tout nouveau point de gestion, le contrôleur réseau, dont le but est de mapper artificiellement les politiques des applications et des utilisateurs sur des composants réseau rigides. Ce contrôleur et les modèles qu'il expose sont limités aux fonctions réseau et ne sont pas aptes à prendre en charge le reste de l'infrastructure. Il importe donc d'intégrer une véritable programmabilité dans l'infrastructure réseau en amont : les composants de l'infrastructure et les éléments de construction qu'ils exposent doivent intégrer la programmabilité à la base, à l'aide d'un modèle compréhensible et rapidement exploitable par les développeurs.

## Programmabilité de l'ACI Cisco : modèle de données orienté objet et API REST

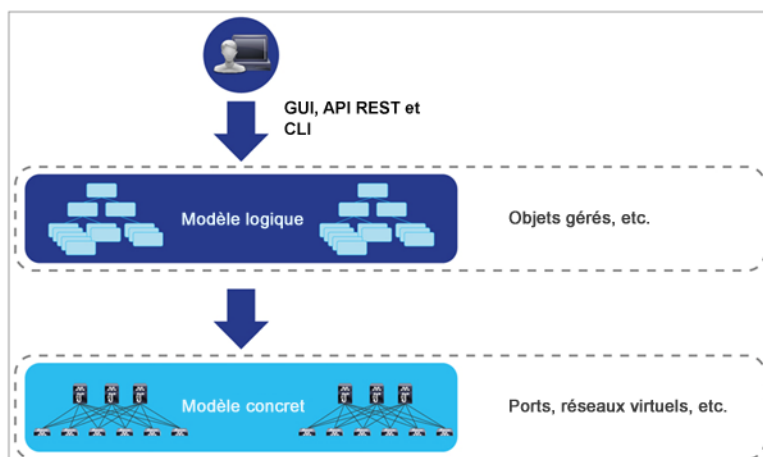
Cisco a adopté une approche visant à construire une infrastructure réseau programmable avec la solution ACI Cisco. Cette infrastructure fonctionne comme un système unique au niveau du fabric et est régulée par le contrôleur d'infrastructure des politiques d'applications Cisco (contrôleur APIC). Avec cette approche, le réseau de data center est lié et traité comme un système de transport intelligent pour les applications qui assurent la prise en charge du secteur opérationnel. Le cœur du système d'exploitation des périphériques réseau qui font partie de ce fabric a été écrit de façon à prendre en charge cette vue du système et à fournir une architecture programmable à la base.

Au lieu de commander l'ouverture d'un sous-ensemble de la fonctionnalité réseau via des interfaces spécifiques de programmes, comme pour les solutions SDN de la génération précédente, l'ensemble de l'infrastructure est accessible en programmation. Cela est rendu possible grâce au modèle d'objet de l'ACI Cisco, qui représente la configuration complète et l'état d'exécution de chacun des composants logiciels et matériels de l'infrastructure. Disponible via des interfaces REST standard, il peut être plus facilement modifié, tout comme la configuration et l'état d'exécution du système.

Au niveau supérieur, le modèle d'objet de l'ACI Cisco est basé sur la théorie de la promesse, qui fournit une architecture de contrôle évolutive dans laquelle des objets autonomes sont chargés de mettre en œuvre les changements d'état souhaités, fournis par le cluster de contrôleurs. Cette approche est plus évolutive que les systèmes de gestion descendants traditionnels, qui obligent à bien connaître l'état actuel et les configurations des niveaux inférieurs de l'infrastructure. La théorie de la promesse a permis la diffusion des changements d'état souhaités, puis leur mise en œuvre par les objets et la transmission d'éventuelles défaillances.

Le modèle d'objet constitue le cœur de la programmabilité de l'ACI Cisco. Ce modèle peut être divisé en deux parties, la partie logique et la partie physique. Les structures basées sur le modèle permettent de représenter les données avec clarté. Le modèle de l'ACI Cisco offre un accès complet au modèle d'information sous-jacent (abstraction des politiques, modèles physiques et données de mise en œuvre et de débogage). La figure 1 illustre sa structure. Le modèle est accessible via des API REST, ce qui rend le système programmable.

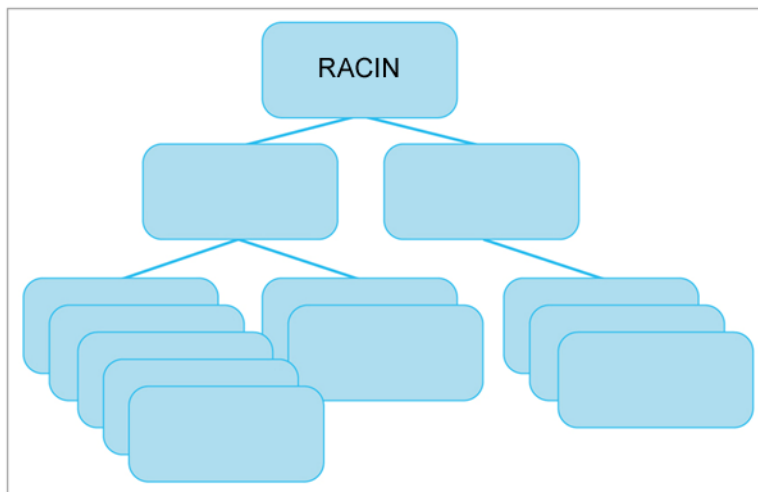
**Figure 1.** ACI Cisco : modèle de données orienté objet et API REST



Comme illustré dans la figure 1, le modèle logique est l'interface avec le système. Les administrateurs ou les systèmes de gestion dans le Cloud de niveau supérieur interagissent avec le modèle logique via les API, les interfaces de ligne de commande ou les interfaces utilisateur graphiques. Les changements apportés au modèle logique sont ensuite transmis au modèle physique, qui devient généralement la configuration matérielle.

Le modèle logique se compose des objets (configuration, politiques et état d'exécution) sujets à modification et des attributs de ces objets. Dans la structure de l'ACI Cisco, ce modèle est appelé arbre d'information de gestion (management information tree, MIT). Chaque nœud de cet arbre représente un objet ou un groupe d'objets gérés. Ces objets sont organisés de façon hiérarchique, dans des conteneurs d'objet logiques. La figure 2 illustre la hiérarchie logique du modèle d'objet du MIT.

**Figure 2.** Arbre d'informations de gestion (MIT)

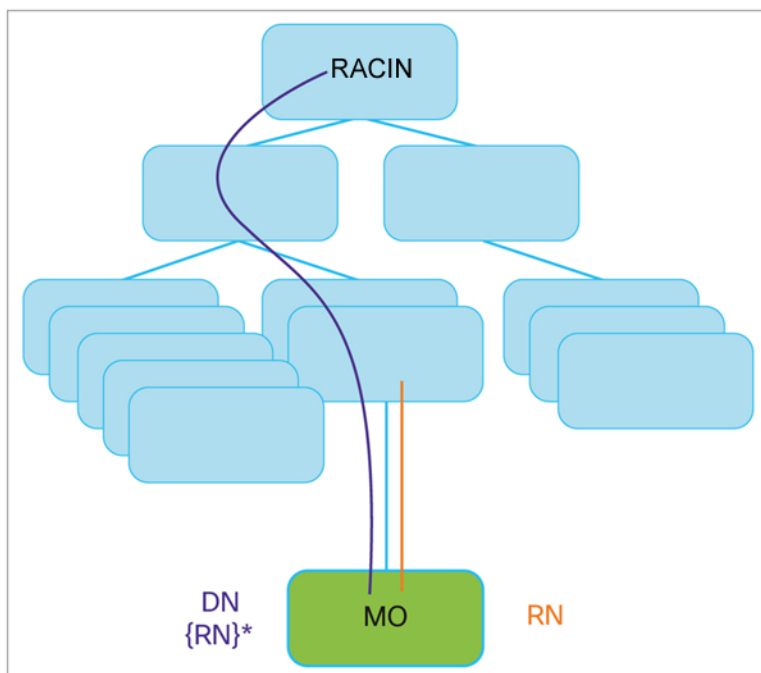


### Objets dans le MIT

L'ACI Cisco utilise une architecture basée sur les modèles d'information, dont la structure décrit toutes les informations qui peuvent être contrôlées par un processus de gestion. Les instances d'objet sont appelées objets gérés (MO). Chaque objet géré dans le système peut être identifié par un nom distinctif unique (DN). Cette approche permet de créer une référence globale à l'objet.

Il est possible de faire référence à un objet par son nom distinctif ou par son nom relatif (RN). Le nom relatif identifie un objet par rapport à son objet parent. Le nom distinctif d'un objet se compose de son nom relatif, ajouté au nom unique de son objet parent. Les noms distinctifs sont mappés directement sur des URL. Il est possible d'utiliser soit le nom relatif, soit le nom distinctif d'un objet pour y accéder, selon son emplacement actuel dans le MIT. La figure 3 illustre la relation entre objets gérés, noms relatifs et noms distinctifs.

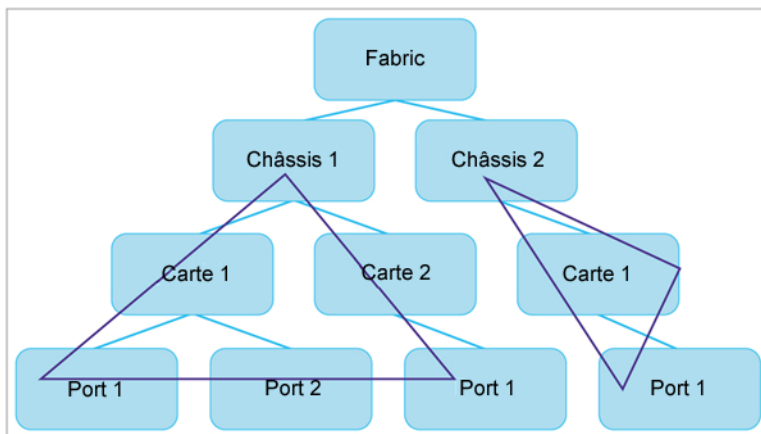
**Figure 3.** Objets gérés, noms relatifs et noms distinctifs



La figure 3 illustre le nom distinctif, qui représente de manière unique toute instance d'un objet géré spécifique, et le nom relatif, qui le représente localement, sous son objet parent géré. Tous les objets de l'arbre existent sous l'objet racine (root).

Du fait de la nature hiérarchique de l'arbre et du système d'attributs utilisé pour identifier les classes d'objets, il est possible d'interroger l'arbre de différentes manières pour obtenir des informations sur les objets gérés. Ces requêtes peuvent porter sur un objet proprement dit, identifié par son nom distinctif, sur une classe d'objets (châssis de commutateur par exemple), ou sur un niveau de l'arbre. Dans ce cas, elles permettent de découvrir tous les membres d'un objet. La figure 4 montre 2 requêtes portant sur un niveau de l'arbre.

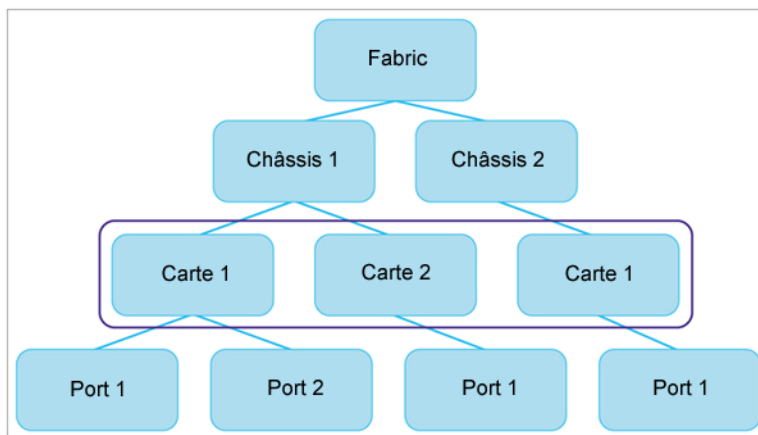
**Figure 4.** Requêtes au niveau de l'arbre



La figure 4 montre des requêtes envoyées au niveau de deux châssis dans l'arbre. Les deux requêtes renvoient l'objet référencé et ses objets enfants. Cette approche est utile pour détecter les composants d'un système de plus grande envergure.

La requête de la figure 4 permet de détecter les cartes et ports d'un châssis de commutateur spécifique. La figure 5 illustre un autre type de requête : la requête au niveau de la classe.

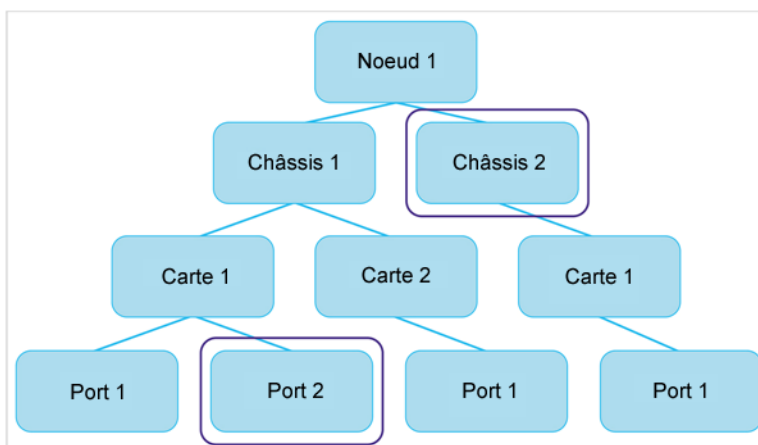
**Figure 5.** Requêtes au niveau de la classe



Comme illustré dans la figure 5, les requêtes au niveau de la classe renvoient les objets d'une classe donnée. Cette approche permet de détecter tous les objets d'un certain type disponibles dans le MIT. Dans cet exemple, la requête porte sur la classe Cartes, laquelle renvoie tous les objets de type Carte.

Le troisième type de requête porte sur le niveau objet. Une requête au niveau de l'objet utilise un nom distinctif pour renvoyer un objet spécifique. La figure 6 illustre deux requêtes au niveau de l'objet : l'une pour le nœud 1 (Node 1) dans le châssis 2, l'autre pour le nœud 1 dans le châssis 1 sous la carte 1 reliée au port 2.

**Figure 6.** Requêtes au niveau de la classe

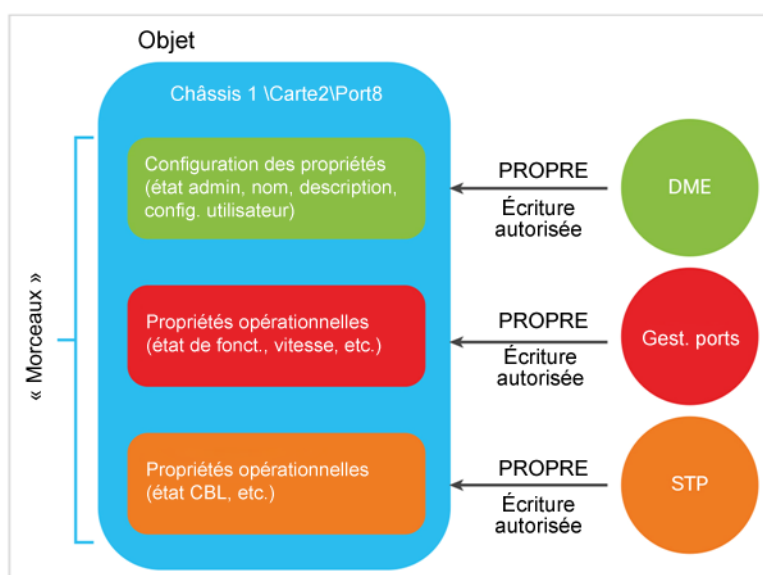


Quel que soit le type de requête utilisé, vous pouvez choisir de renvoyer le sous-arbre dans son intégralité ou en partie. Le mécanisme de contrôle d'accès à base de rôles (RBAC) du système détermine quels objets sont renvoyés. Seuls les objets que l'utilisateur a le droit de visualiser sont renvoyés.

## Propriétés des objets gérés

Dans l'ACI Cisco, chaque objet géré contient des propriétés qui le définissent. Les propriétés d'un objet géré sont divisées en plusieurs fragments, eux-mêmes gérés par des processus définis au sein du système d'exploitation. Il est donc possible que plusieurs processus accèdent à un même objet. Toutes ces propriétés sont compilées au moment de l'exécution et présentées à l'utilisateur sous forme d'objet unique. La figure 7 montre un exemple de cette relation.

**Figure 7.** Propriétés des objets gérés



Dans la figure 7, trois processus écrivent dans des fragments de propriétés de l'objet exemple. Le moteur de gestion de données (DME), qui sert d'interface entre le contrôleur APIC (donc l'utilisateur) et l'objet, le gestionnaire de ports qui gère la configuration des ports et le Protocole Spanning Tree (STP) interagissent tous les trois avec des fragments de cet objet. L'objet proprement dit est présenté à l'utilisateur via l'API sous forme d'entité unique compilée au moment de l'exécution.

## Accès aux données de l'objet via des interfaces REST

REST est un style d'architecture logicielle utilisé pour des systèmes distribués tels que le World Wide Web. L'architecture REST s'est imposée ces dernières années comme un modèle prédominant de conception de services Web. Son style plus simple lui a permis de surpasser d'autres modèles de conception tels que le protocole SOAP (Simple Object Access) et le langage WSDL (Web Services Description Language). Le contrôleur APIC Cisco prend en charge les interfaces REST pour fournir un accès en programmation à l'ensemble de la solution ACI de Cisco.

Le modèle d'information basé sur les objets de l'ACI Cisco explique son excellente compatibilité avec les interfaces REST. En effet, les URL et URI sont mappées directement sur des noms distinctifs identifiant des objets de l'arbre, et toute donnée de l'arbre peut être décrite sous la forme d'un texte structuré autonome, dans un document encodé en XML ou JSON (JavaScript Object Notation). Les objets ont des relations parent-enfant identifiées au moyen de propriétés et de noms distinctifs qui sont lus et modifiés par un ensemble d'opérations de création, de lecture, de mise à jour et de suppression (CRUD).

Pour récupérer et modifier les données d'objets de l'APIC Cisco, il est possible d'accéder aux objets directement à leur adresse définie, via leur URL REST, ou par le biais de commandes HTTP standard. Le format d'URL utilisé peut être représenté comme suit :

```
<système>/api/[mo|classe]/[dn|classe][:méthode].[xml|json]?{options}
```

Les différents blocs de construction de cette URL sont les suivants :

- **Système** : identifiant du système ; adresse IP ou nom d'hôte résoluble par DNS
- **mo | classe** : indique s'il s'agit d'une requête portant sur un objet géré, au niveau de l'arbre ou au niveau de la classe
- **classe** : classe d'objet géré (tel que spécifié dans le modèle d'information) des objets sur lesquels porte la requête ; le nom de la classe est représenté ainsi : <pkgName><ManagedObjectClassName>
- **dn** : nom distinctif (nom hiérarchique unique de l'objet dans l'arbre d'informations de gestion) de l'objet sur lequel porte la requête
- **méthode** : indication (facultative) de la méthode invoquée sur l'objet ; s'applique uniquement aux requêtes HTTP POST
- **xml | json** : format d'encodage
- **options** : options, filtres et arguments de la requête

L'URL REST vous permet d'adresser et d'accéder à un objet individuel ou à une classe d'objets, et donc à disposer d'un accès en programmation complet à l'arbre des objets et, par extension, à l'ensemble du système.

## Kits de développement logiciel pour les environnements de programmation

Les API REST disponibles pour l'ACI Cisco facilitent l'intégration dans tout environnement de programmation, indépendamment du langage et de la méthodologie de développement utilisés. Afin d'accélérer le développement dans les environnements de programmation couramment utilisés, des kits de développement logiciel seront disponibles prochainement pour l'ACI Cisco, comme le kit de développement logiciel Cisco ACI-pysdk, basé sur Python et destiné aux environnements de programmation Python. Les bibliothèques Python et les API intégrées dans ce SDK extraient les appels de l'API REST sous-jacente et assurent une intégration facile et rapide dans les suites logicielles basées sur Python.

## Conclusion

Le modèle de données orienté objet de l'ACI Cisco a été conçu à la base pour offrir une programmabilité réseau. Le système d'exploitation des périphériques a été réécrit pour l'ACI Cisco en tant que système d'exploitation pour commutateurs entièrement basé sur des objets. Les composants de l'ACI Cisco sont gérés par le contrôleur APIC, qui fournit des API REST entièrement fonctionnelles. Au-dessus de ces API, une interface de ligne de commande et une interface utilisateur graphique assurent l'administration au quotidien.

Le modèle d'objet offre une programmabilité fluide et un accès complet aux composants sous-jacents de l'infrastructure, par l'intermédiaire des API REST. Les objets sont organisés de manière logique dans un modèle hiérarchique et stockés dans l'arbre d'informations de gestion (MIT). Cette approche offre un cadre de programmabilité et de contrôle du réseau dont le degré d'ouverture est inédit.

## Informations complémentaires

<http://www.cisco.com/go/aci>.



**Siège social aux États-Unis**  
Cisco Systems, Inc.  
San Jose. CA

**Siège social en Asie-Pacifique**  
Cisco Systems (États-Unis) Pte. Ltd.  
Singapour

**Siège social en Europe**  
Cisco Systems International BV Amsterdam.  
Pays-Bas

Cisco compte plus de 200 agences à travers le monde. Les adresses, numéros de téléphone et de fax sont répertoriés sur le site Web de Cisco, à l'adresse : [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco et le logo Cisco sont des marques commerciales ou des marques déposées de Cisco Systems, Inc. et/ou de ses filiales aux États-Unis et dans d'autres pays. Pour consulter la liste des marques commerciales Cisco, visitez le site : [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Les autres marques mentionnées dans les présentes sont la propriété de leurs détenteurs respectifs. L'utilisation du terme « partenaire » n'implique pas de relation de partenariat commercial entre Cisco et d'autres entreprises. (1110R)