

ΔΙΑΧΕΙΡΙΣΗ ΔΙΚΤΥΩΝ - NETWORK MANAGEMENT

Αρχιτεκτονικές Διαχείρισης Δικτύων

Network Management Architectures: Automation, Data Modeling, NETCONF

**Network Programmability, Automation
Information Models (IM), Data Models (DM)
NETCONF (Network Configuration Protocol)
YANG (Yet Another Next Generation)
Data Modeling Languages**

Δ. Καλογεράς

Ερευνητής ΕΠΙΣΕΥ – Ε.Μ.Π.

dkalo@nocntua.gr

www.netmode.ntua.gr

Νέα Κτίρια ΣΗΜΜΥ - Αίθουσα 013

4/12/2023

RFC 3535: Απαιτήσεις?



Fault



Configuration



Accounting



Performance



Security

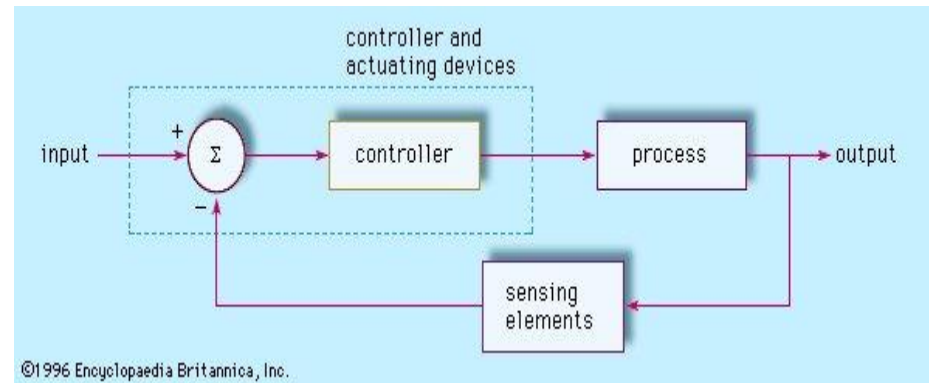
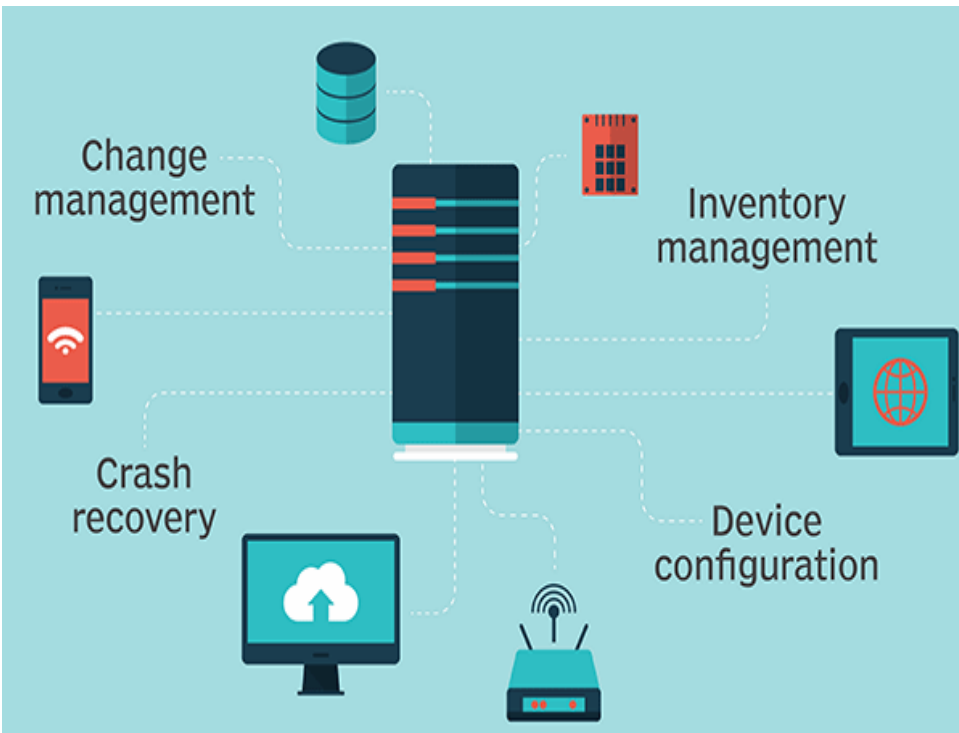
Τι χρειάζομαι;

- Μια προγραμματιζόμενη διεπαφή (**API**) για διάρθρωση (configuration) συσκευών, **OXI** ένα **CLI** (αυτά αλλάζουν εύκολα από τους κατασκευαστές, παράμετροι γίνονται default / εξαφανίζονται. *Δεν έχουν κύκλο ζωής*)
- Διαχωρισμός Configuration και State
- Δυνατότητα για *configuration* των υπηρεσιών (*services*), **OXI** μόνο των συσκευών (*devices*)



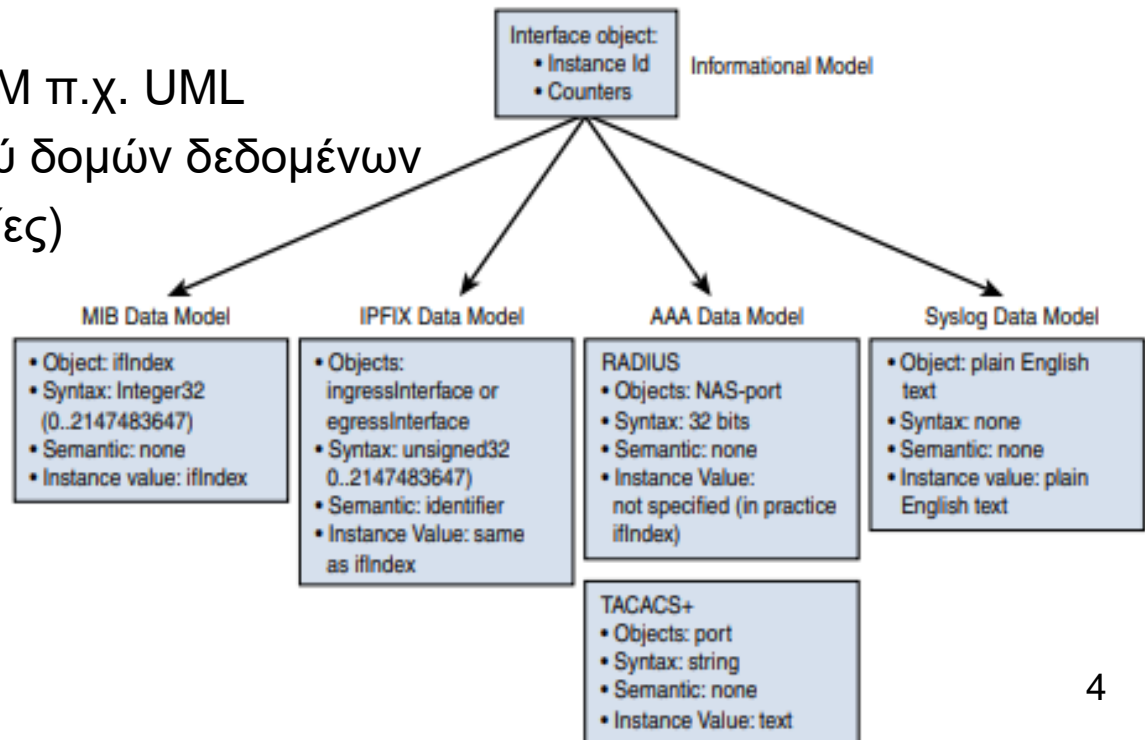
Διαχείριση Διάρθρωσης (Configuration Management): Ένας συνεχής αγώνας

Αυτοματισμός



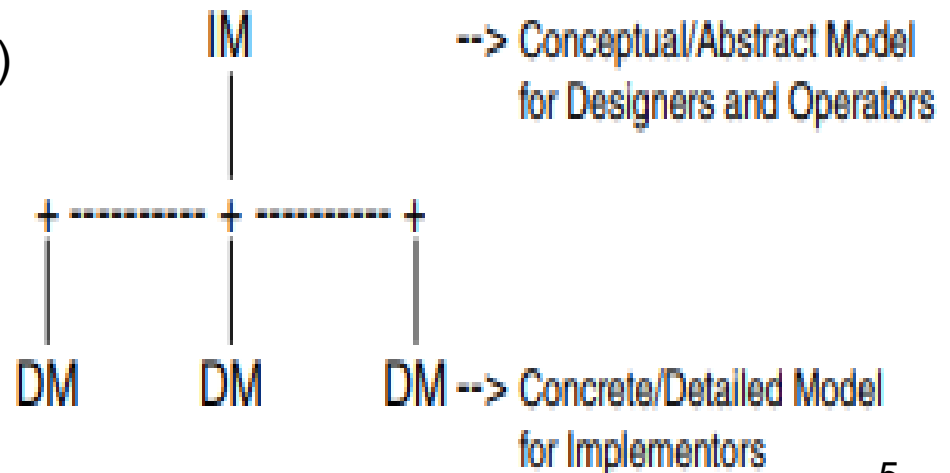
Διαχείριση ΙΔΙΩΝ οντοτήτων ΔΙΑΦΟΡΕΤΙΚΩΝ συστημάτων/υπηρεσιών

- Περιπτώσεις παροχής υπηρεσιών
 - π.χ. storage μεταξύ δύο σημείων του δικτύου ή ετερογενών συστημάτων
 - π.χ. ενεργοποίηση xDSL μεταξύ διαφορετικών στοιχείων του δικτύου (DSLAM, βάσης χρηστών, authentication κλπ.)
 - π.χ. ενεργοποίηση IP service εντός profile χρήσης
- Ίδιου σκοπού «συναρτήσεις» με διαφορετικά ορίσματα
 - Δημιουργία abstraction
 - Abstract class ή χρήση IM π.χ. UML
 - Μετασχηματισμός μεταξύ δομών δεδομένων
 - Μεταδεδομένα (κατηγορίες)

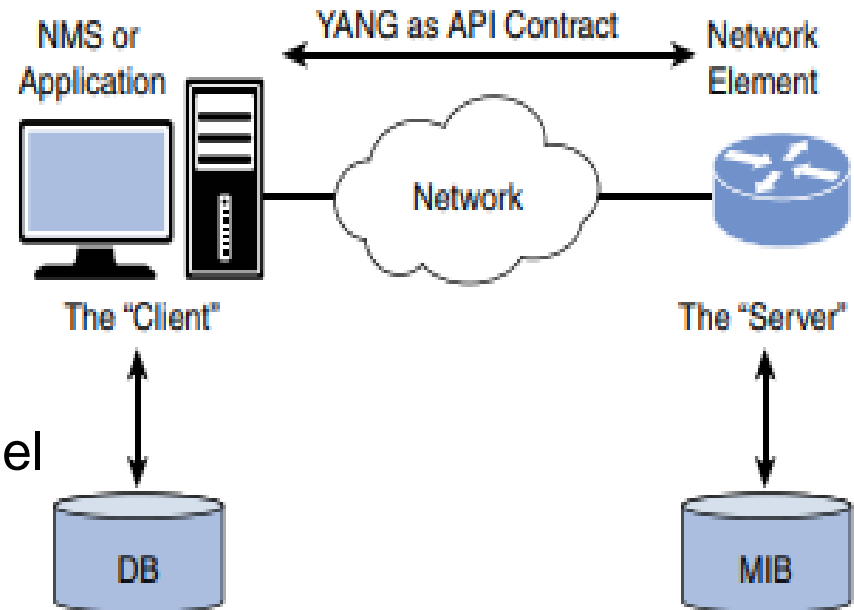


Information Model (IM) vs Data Model (DM)

- Ανάγκη για περιγραφή οντοτήτων (π.χ. Δίκτυο, χρήστης, κόμβος)
 - με τα χαρακτηριστικά τους (ιδιότητες - attributes)
 - και τις σχετικές συναρτήσεις χειρισμούς τους (π.χ. CRUD, search, list κλπ)
- Παρόμοια με τις ανάγκες στις Βάσεις Δεδομένων
- DM περιγράφει αντικείμενα ΚΑΙ μηχανισμούς/πρωτόκολλα
- Γνωστά DM από SDO (StanDard Org. bodies)
 - SMI-SNMP (IETF) , CIM (DMTF), NETCONF (IETF)
- Ανάγκη για διαχείριση οντοτήτων (πάνω από διαφορετικά DM -> πρωτόκολλα)
 - Ενθυλάκωση CIM (ενθυλάκωση)
 - Abstraction (Χρήση IM)



YANG (RFC 6020)



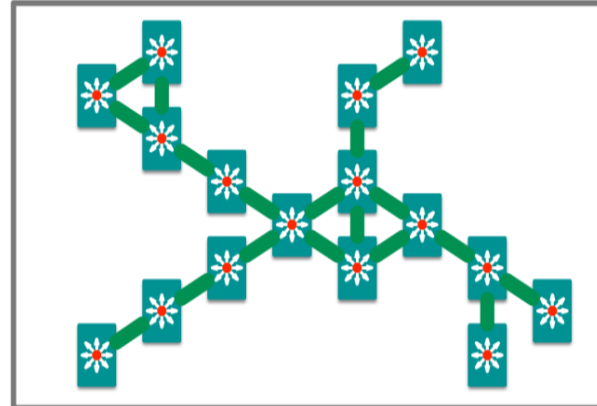
- ..a data modeling language used to model configuration and state data
- Q: Πως αλλάζουμε configuration?
- A: via Network Configuration Protocol (NETCONF),
 - NETCONF remote procedure calls (rpc), NETCONF notifications
 - Get, get-config, edit-config, copy-config, delete-config
- NETCONF: μηχανισμοί και κωδικοποιήσεις για χειρισμό «*configuration datastores* (συλλογή συγκεκριμένου configuration) π.χ. Interface, BGP κλπ (RFC 6244)
- RESTCONF (RFC 8040) υποσύνολο του NETCONF με χρήση HTTP verbs για operational status
- Openconfig (Χρήση protobufs, από το model -> API)

Διάφορα Μοντέλα YANG



Device Data Models

- Interface
- VLAN
- Device ACL
- Tunnel
- OSPF
- etc



Service Data Models

- L3 MPLS VPN
- MP-BGP
- VRF
- Network ACL
- System Management
- Network Faults
- etc

Πηγή των μοντέλων?

Πρότυπα
Standard

Ορισμός από SDO

IETF, ITU, OpenConfig...

- Συμβατά με πρότυπα
`ietf-diffserv-policy.yang` `ietf-diffserv-classifier.yang` `ietf-diffserv-target.yang`

Κατασκευαστές

Vendor definition

Cisco, Juniper, HP...

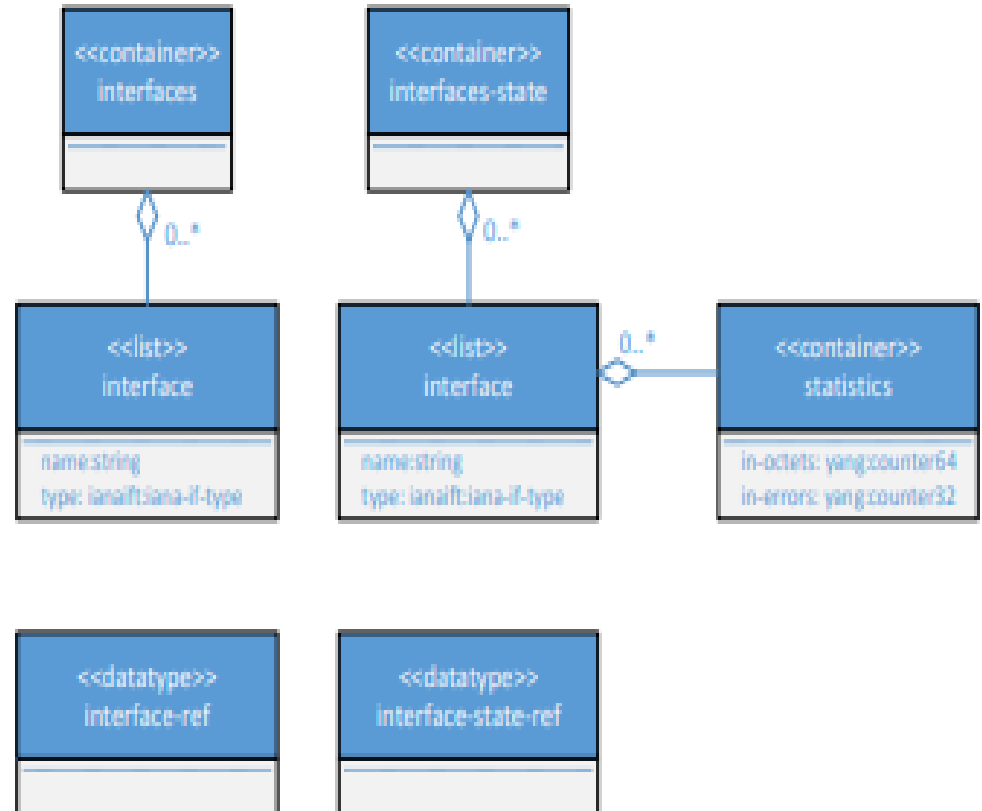
- Ειδικά για το υλικό των Κατασκευαστών π.χ. Cisco
 - `cisco-memory-stats.yang`
 - `cisco-flow-monitor`
 - `cisco-qos-action-qlimit-cfg`

YANG vs UML

```

1 module ietf_interfaces_parts {
2   prefix if;
3   organization "IETF NETMOD Working Group";
4   revision 2013-07-04 {
5     description "Initial revision.";
6     reference "...";}
7   container interfaces {
8     description "configuration parameters";
9     list interface {
10      key "name";
11      description "configured interfaces on device"
12      leaf name {
13        type string;
14        description "name of the interface"}
15      leaf enabled {
16        type boolean;
17        default "true";
18        description "state of the interface"
19        reference "RFC 2863";}
20    }
21  }
22  container interfaces_state {
23    config false;
24    description "Data nodes for the operational state"
25    list interface {
26      key "name";
27      description "interfaces on the device"
28      leaf name {
29        type string;
30        description "name of the interface";}
31    }
32  }
33 }

```



Δενδρική δομή των YANG

```
user$ pyang -f tree ietf-  
interfaces.yang
```

```
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}?
```

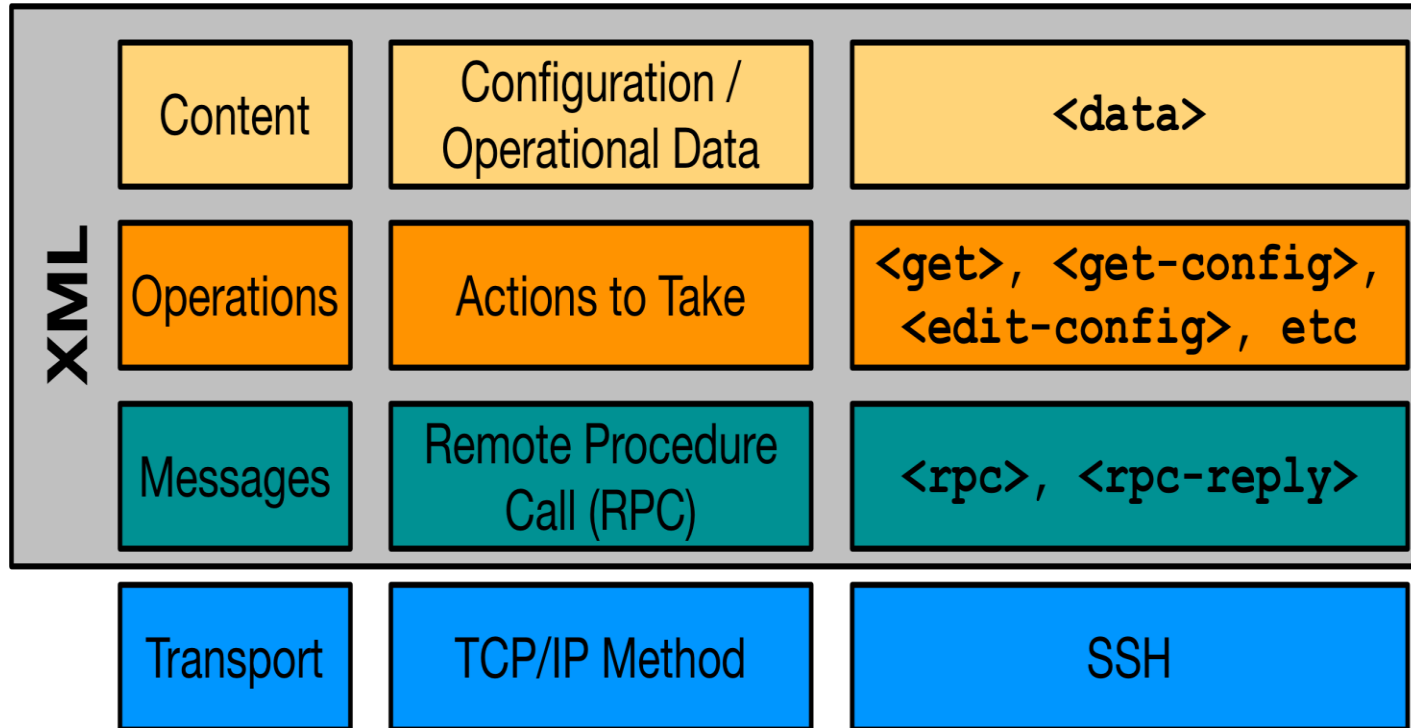
Χρήση pyang

- Python YANG Library
- Επικύρωση και προβολή αρχείων YANG
- Διαφορετικά formats:
 - Text: Δενδρικά
 - HTML: jstree

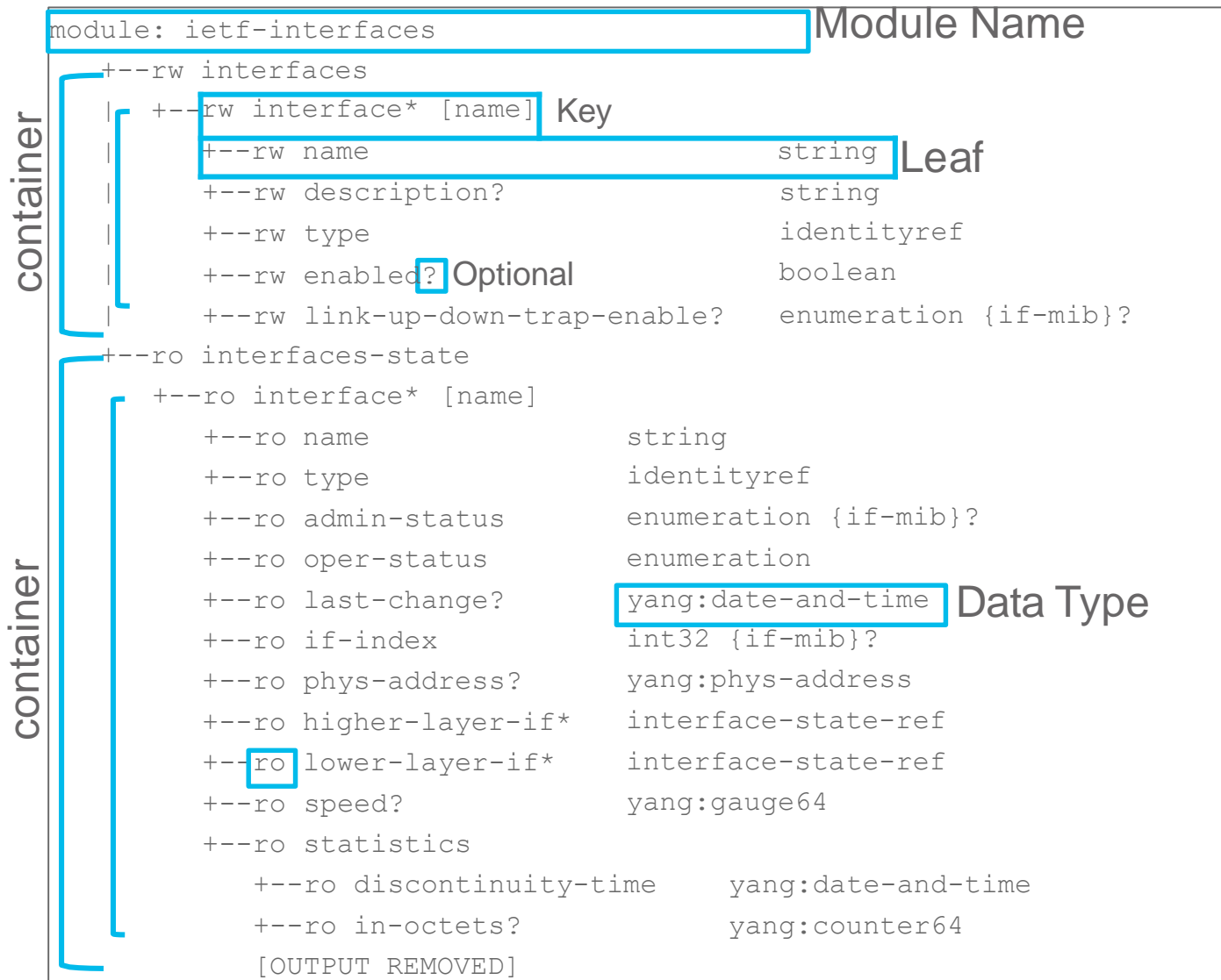
```
module: ietf-interfaces
```

container	+--rw interfaces		
	+--rw interface* [name]	Key	
	+--rw name	string	Leaf
	+--rw description?	string	
	+--rw type	identityref	
	+--rw enabled?	boolean	Optional
	+--rw link-up-down-trap-enable?	enumeration {if-mib}?	
container	+--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	Data Type
	+--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		
	[OUTPUT REMOVED]		

Στοιβά Πρωτοκόλλων NETCONF

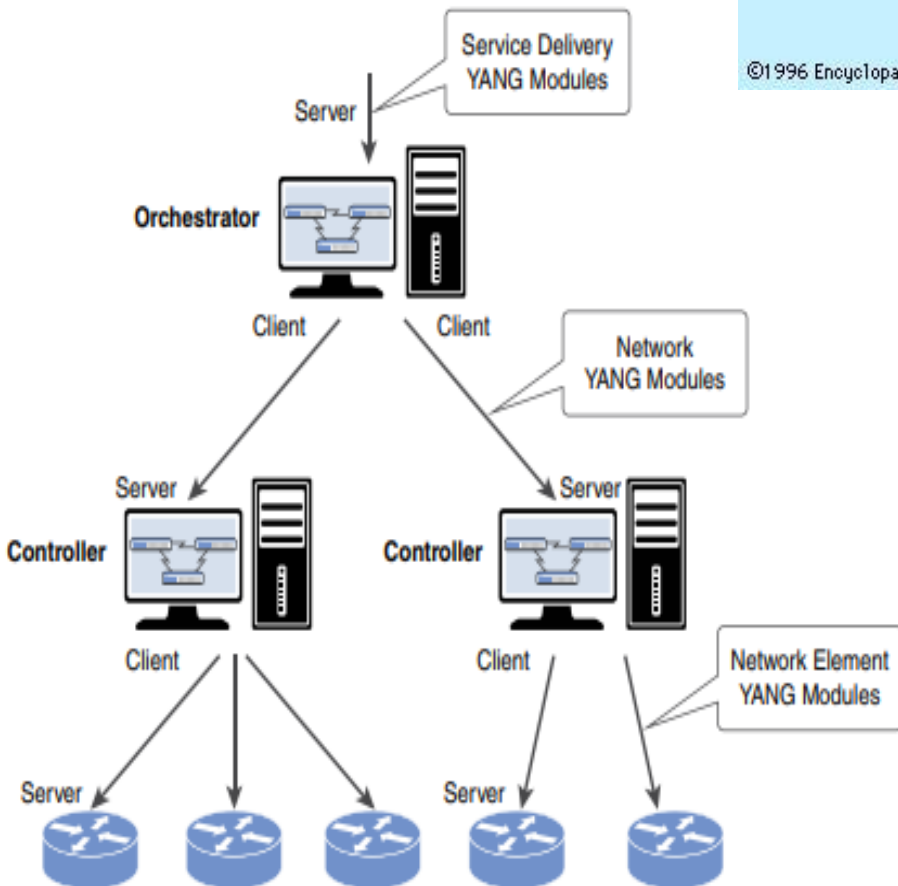
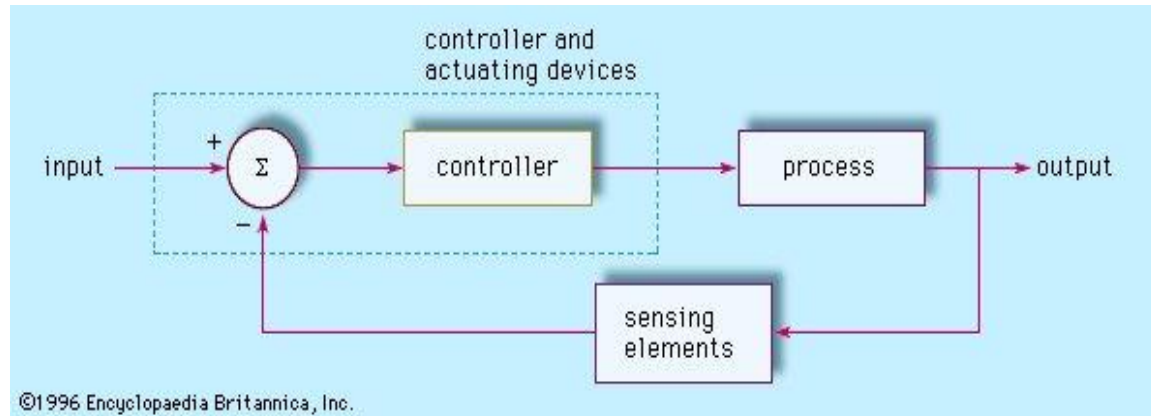


Χρήση NETCONF για λήψη μοντέλου ietf-interfaces



Εναλλαγή μεταξύ configuration και status

Αυτοματισμός



Data Modeling Language (Schema Language)

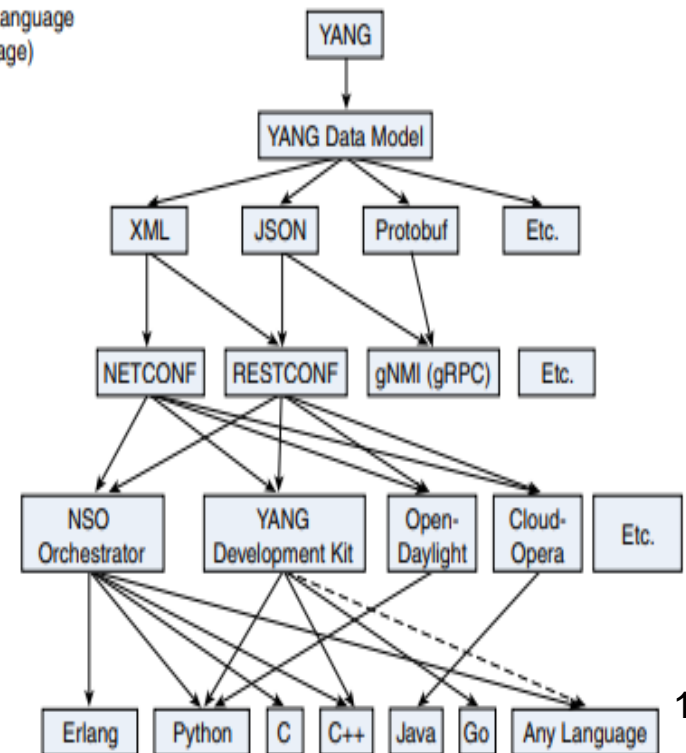
Data Modeling (Schema)

Encoding (Serialization)

Protocol

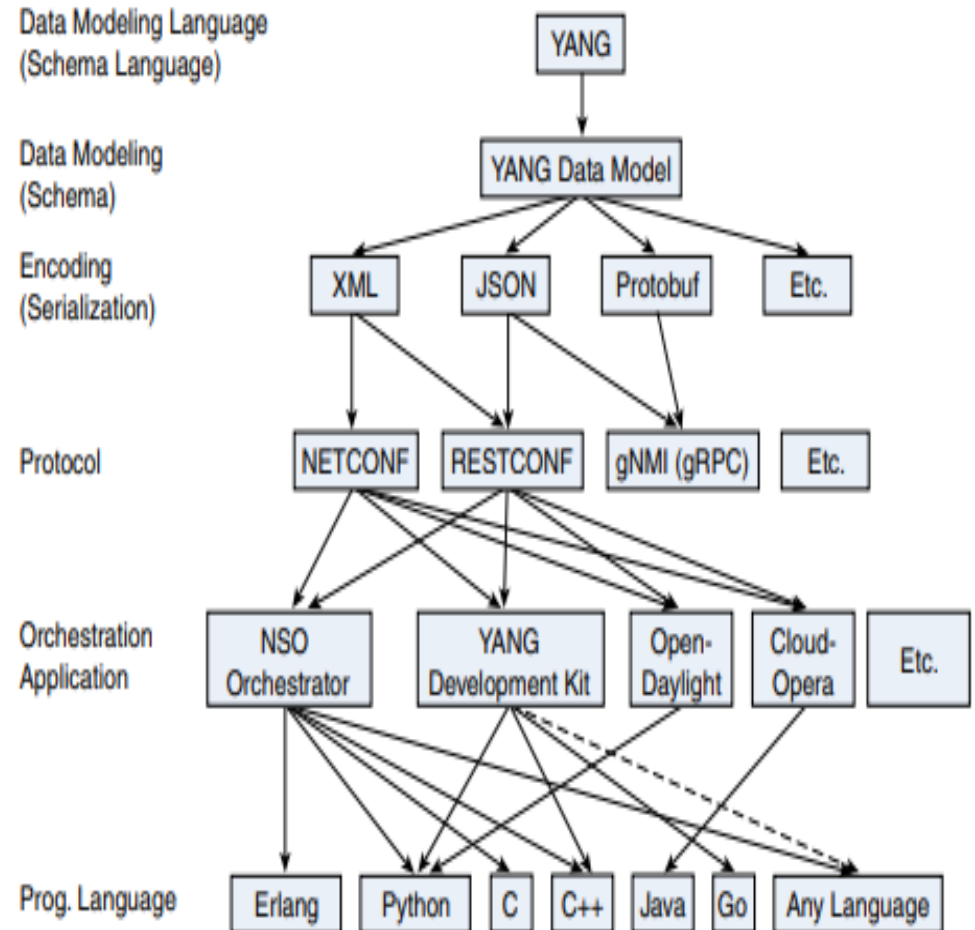
Orchestration Application

Prog. Language



Data Modeling Languages (1)

- Μοντέλο περιγραφής data sources → API
- Περισσότερο τεχνολογία λογισμικού παρά διαχείριση δικτύου
- Αυτοματοποίηση ΠΑΡΑΓΩΓΗΣ ΚΩΔΙΚΑ !!
- Από το YANG model στην κλάση της γλώσσας προγραμματισμού
- Τεχνικές προγραμματισμού για την συλλογή δεδομένων & εκτέλεση εντολών



Data Modeling Languages (2)

- YANG Data model - Schema
- From DM → API
- API -> interconnection -> API
- Automation

```
module my-interfaces {  
  namespace "urn:example";  
  
  container interfaces {  
    list interface {  
      key name;  
      leaf name { type string; }  
      leaf admin-status { type enumeration; }  
    }  
  }  
  
  rpc flap-interface {  
    input {  
      leaf name { type string; }  
    }  
    output {  
      leaf result { type boolean; }  
    }  
  }  
}
```

GET : Gets a resource

```
GET /restconf/data/my-interfaces:interfaces  
GET /restconf/data/my-interfaces:interfaces/interface/<some  
name>
```

POST : Creates a resource or invoke operation

```
POST /restconf/operations/my-interfaces:flap-interface  
+ JSON/XML Form Data (including name)  
Response will have JSON/XML result
```

PUT : Replaces a resource

```
PUT /restconf/data/my-interfaces:interfaces/interface/<some  
name> + JSON/XML Form Data (name, admin-status)
```

DELETE : Removes a resource

```
DELETE /restconf/data/my-  
interfaces:interfaces/interface/<some name>
```


Data Modeling Languages (3)

- Μοντέλο περιγραφής data sources → API
- Yangify
- Αυτοματοποίηση ΠΑΡΑΓΩΓΗΣ ΚΩΔΙΚΑ !!
- Από structured δεδομένα (JSON) → YANG model → κλάση της γλώσσας προγραμματισμού και parser
- Τεχνικές προγραμματισμού για την συλλογή δεδομένων και εκτέλεση εντολών