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

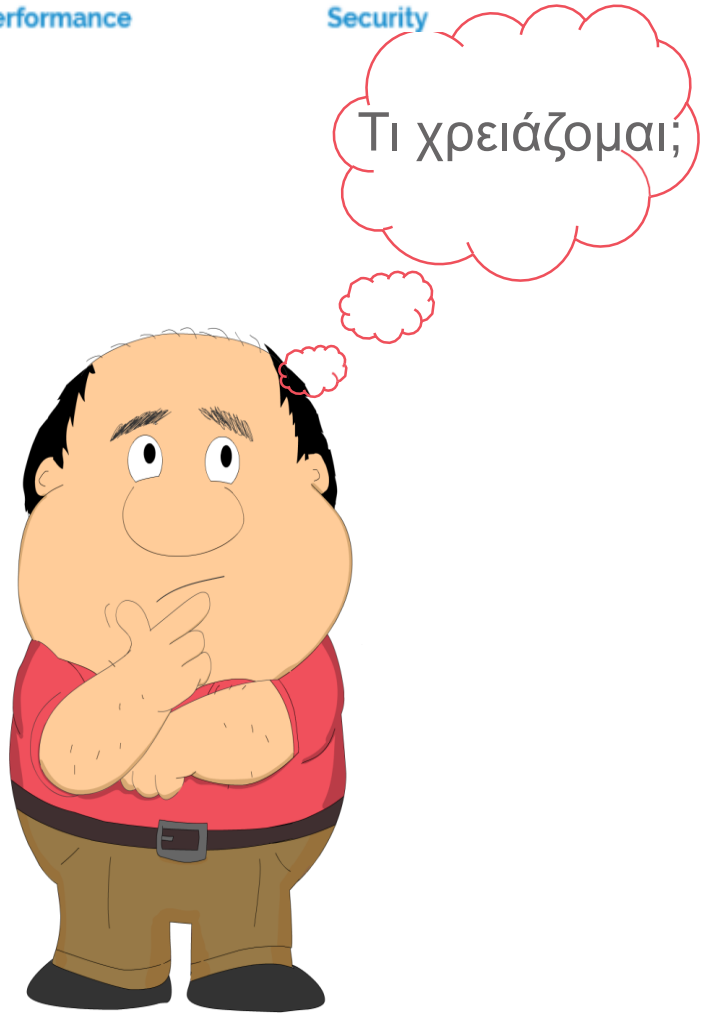
Δ. Καλογεράς
dkalo@noc.ntua.gr

www.netmode.ntua.gr

29/11/2021

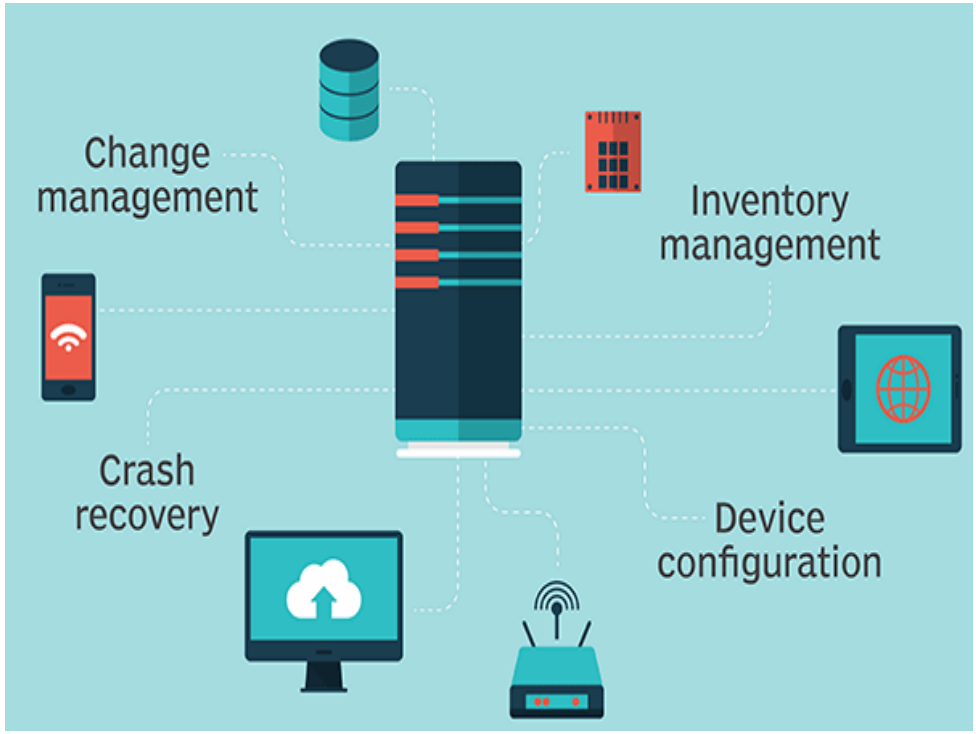
RFC 3535: Απαιτήσεις?

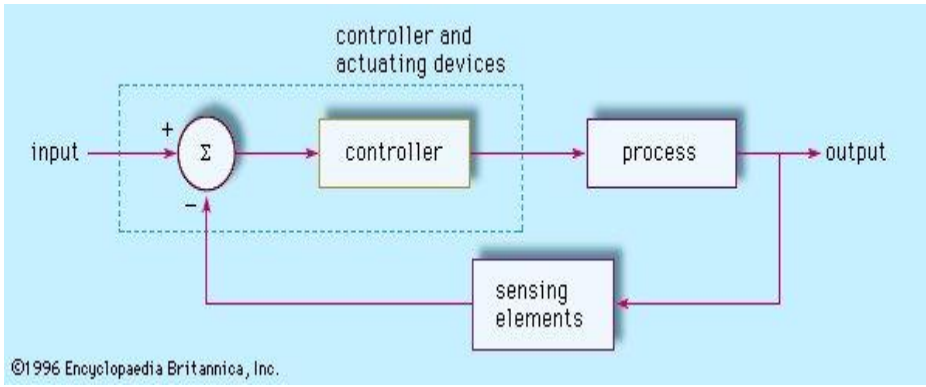
- Μια προγραμματιζόμενη διεπαφή (**API**) για διάρθρωση (configuration) συσκευών, **ΟΧΙ** ένα **CLI** (αυτά αλλάζουν εύκολα από τους κατασκευαστές, παράμετροι γίνονται default / εξαφανίζονται. ***Δεν έχουν κύκλο ζωής***)
- **Διαχωρισμός Configuration και State**
- Δυνατότητα για ***configuration*** των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



Διαχείριση Διάρθρωσης (Configuration Management): Ένας συνεχής αγώνας

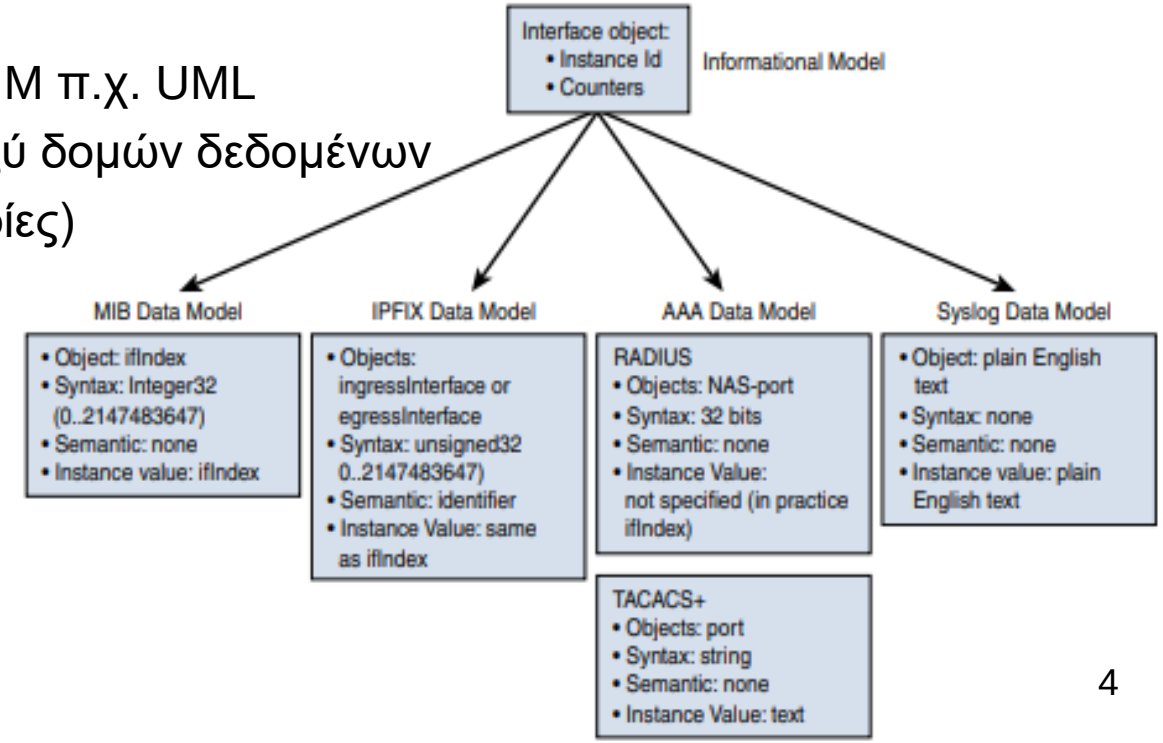
Αυτοματισμός





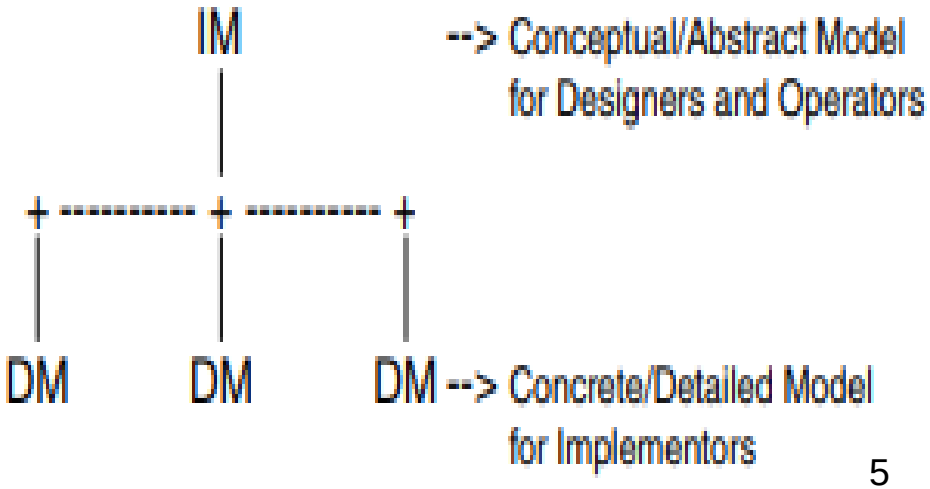
Διαχείριση ΙΔΙΩΝ *οντοτήτων ΔΙΑΦΟΡΕΤΙΚΩΝ συστημάτων/υπηρεσιών*

- Περιπτώσεις παροχής υπηρεσιών
  - π.χ. storage μεταξύ δύο σημείων του δικτύου ή ετερογεννών 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)

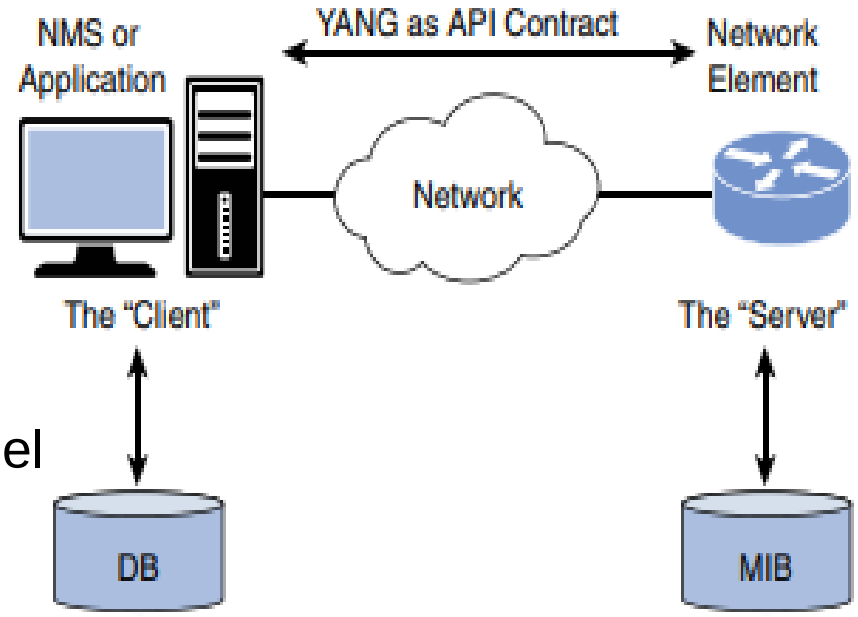


Information Model (IM) vs Data Model (DM)

- Ανάγκη για περιγραφή οντοτήτων (π.χ. Δίκτυο, χρήστης, κόμβος)
  - με τα χαρακτηριστικά τους (ιδιότητες - attibutes)
  - 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)

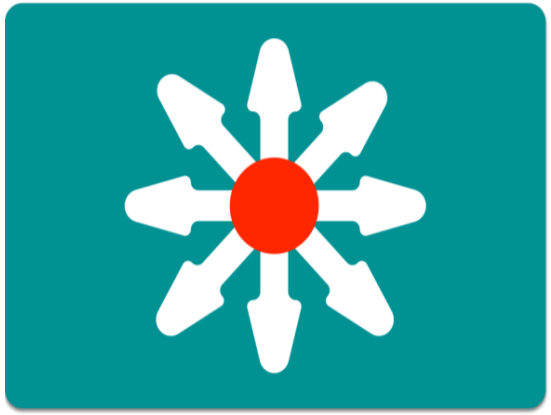


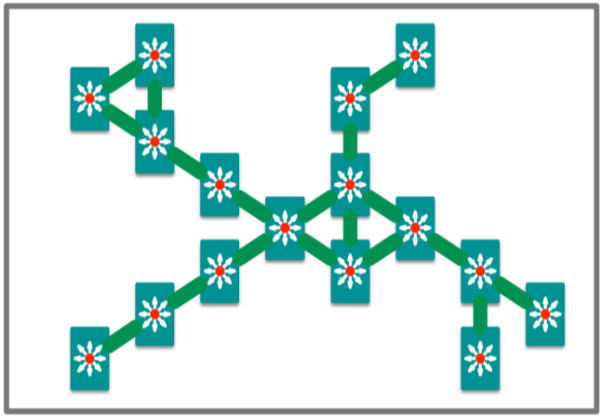
YANG (RFC 6020)



- ..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?

Ορισμός από SDO

IETF, ITU, OpenConfig…

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

Vendor definition

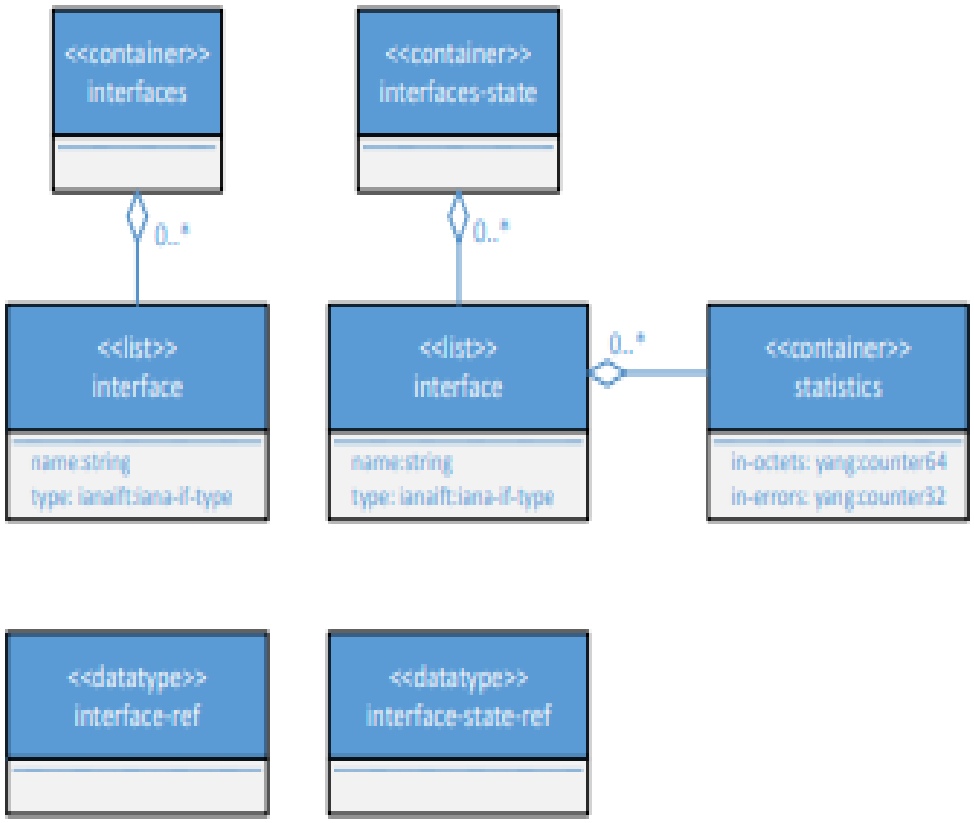
Cisco, Juniper, HP…

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

https://github.com/YangModels/yang

YANG vs UML

```
module ietf_interfaces_parts {
    prefix if;
    organization "IETF NETMOD Working Group";
    revision 2013-07-04 {
        description "Initial revision.";
        reference "...";}
    container interfaces {
        description "configuration parameters";
        list interface {
            key "name";
            description "configured interfaces on device"
            leaf name {
                type string;
                description "name of the interface"}
            leaf enabled {
                type boolean;
                default "true";
                description "state of the interface"
                reference "RFC 2863";}
        }
    }
    container interfaces_state {
        config false;
        description "Data nodes for the operational state'
        list interface {
            key "name";
            description "interfaces on the device"
            leaf name {
                type string;
                description "name of the interface";}
        }
    }
}
```

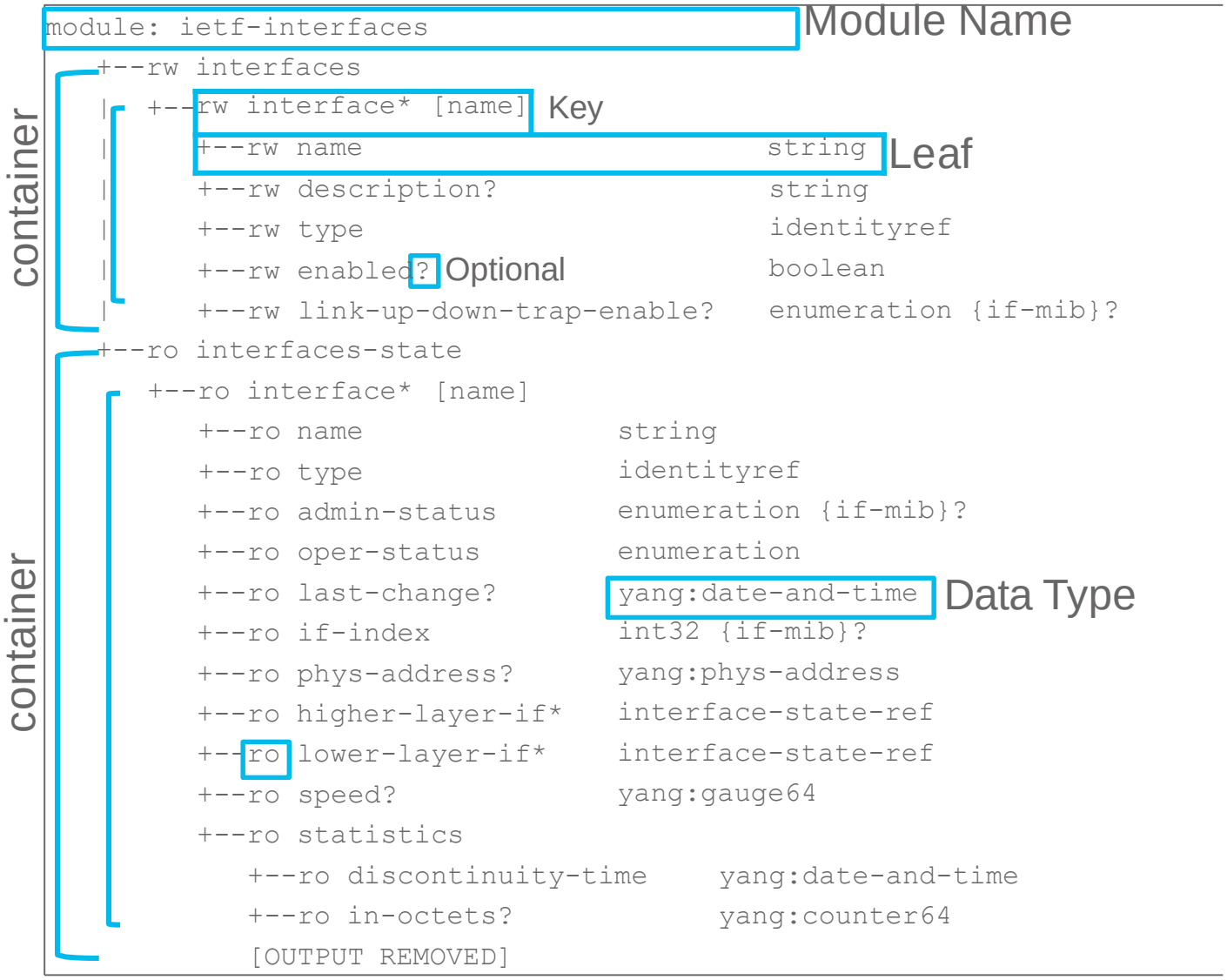


Δενδρική δομή των YANG

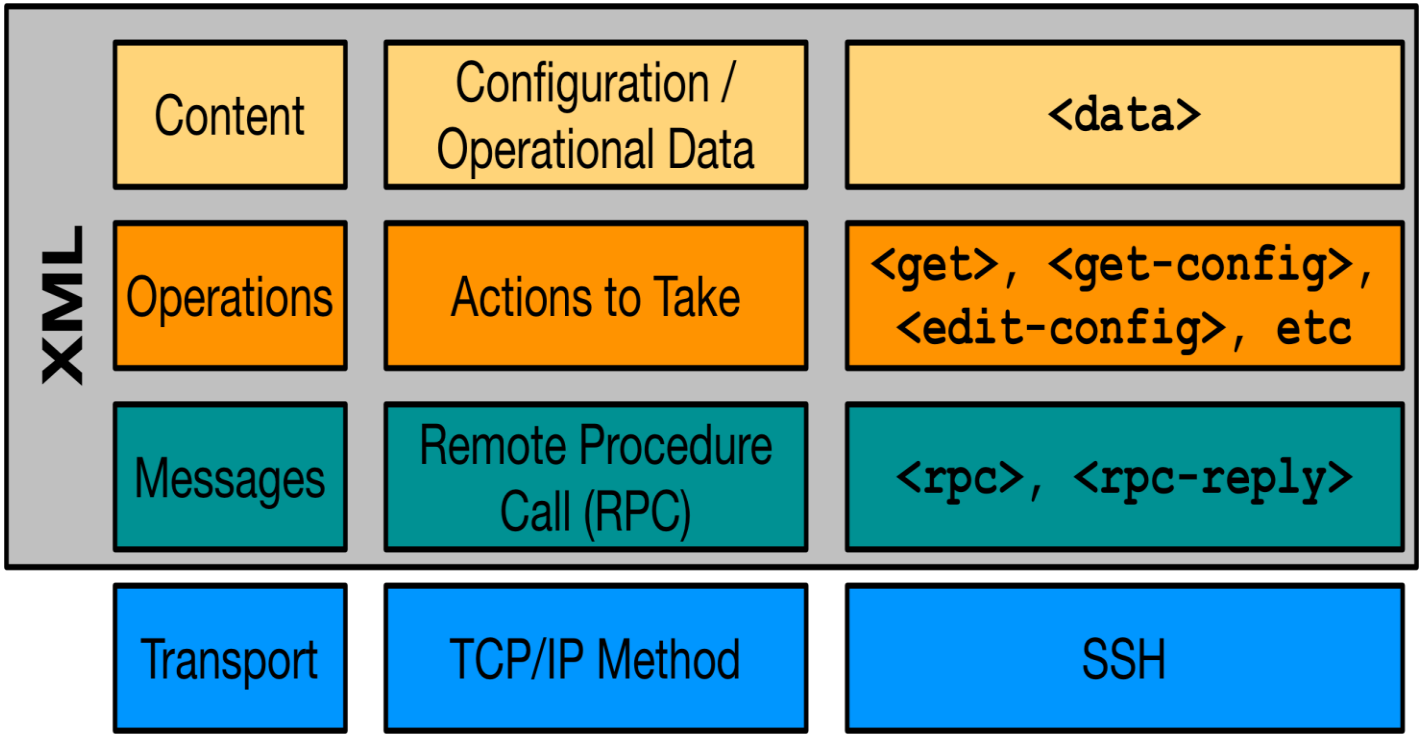
```
user$ pyang -f tree ietf-
interfaces.yang


module: ietf-interfaces
   +--rw interfaces

   |  +--rw interface* [name]
   |     +--rw name                string
   |     +--rw description?        string
   |     +--rw type                identityref
   |     +--rw enabled?            boolean
   |     +--rw link-up-down-       enumeration
   |        trap-enable?           {if-mib}?
```

Χρήση pyang

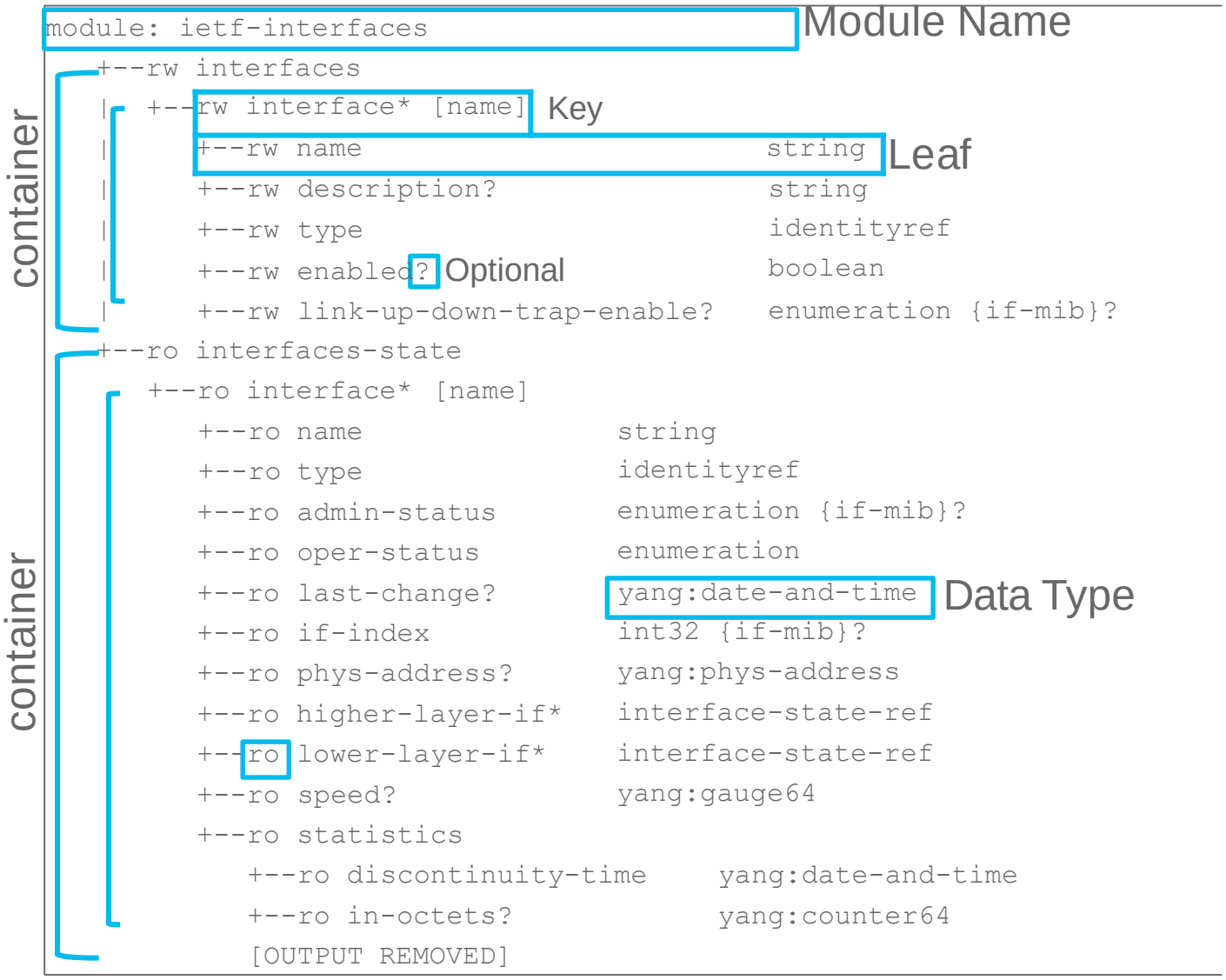
- Python YANG Library
- Επικύρωση και προβολή αρχείων YANG
- Διαφορετικά formats:
  - Text: Δενδρικά
  - HTML: jstree



Στοίβα Πρωτοκόλλων NETCONF



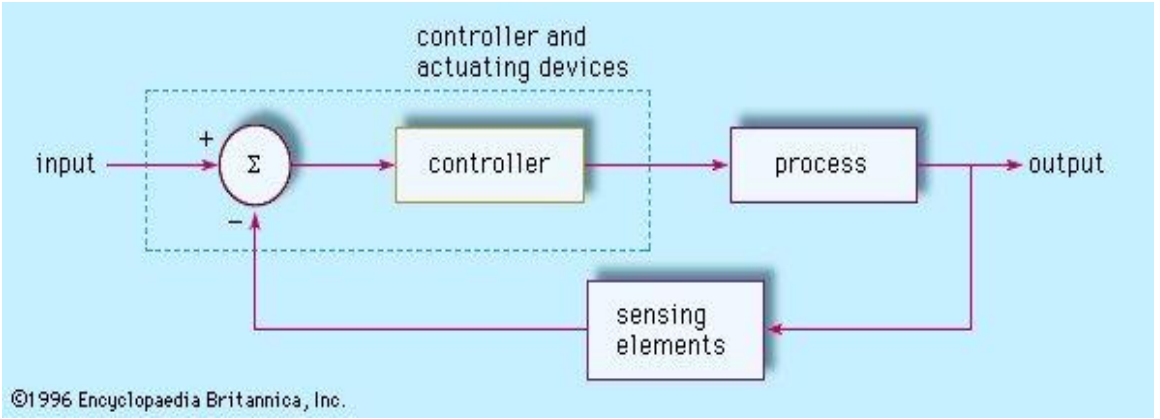
Χρήση NETCONF για λήψη μοντέλου ietf-interfaces

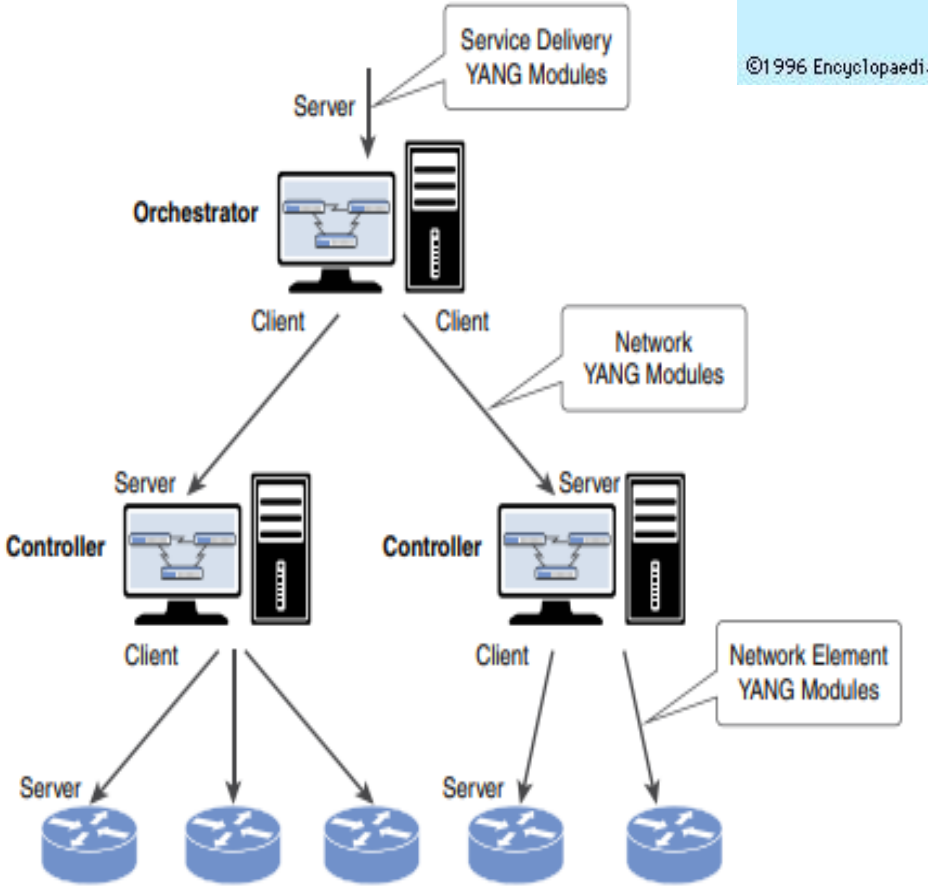


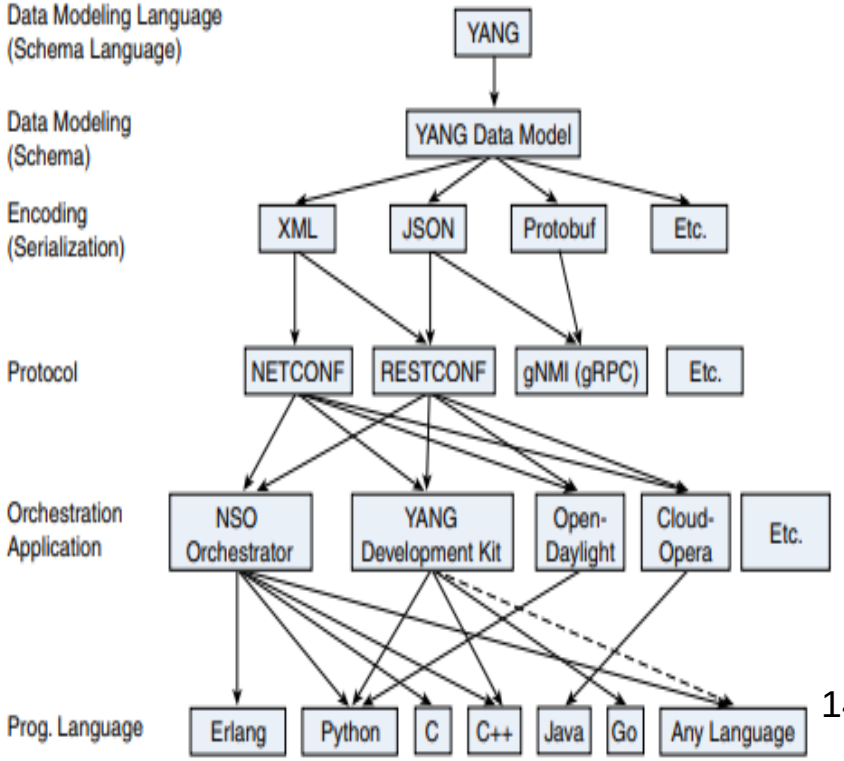
BRKDEV-1368/yang/device_info.py BRKDEV-1368/yang/example1.py

Εναλλαγή μεταξύ configuration και status

Αυτοματισμός

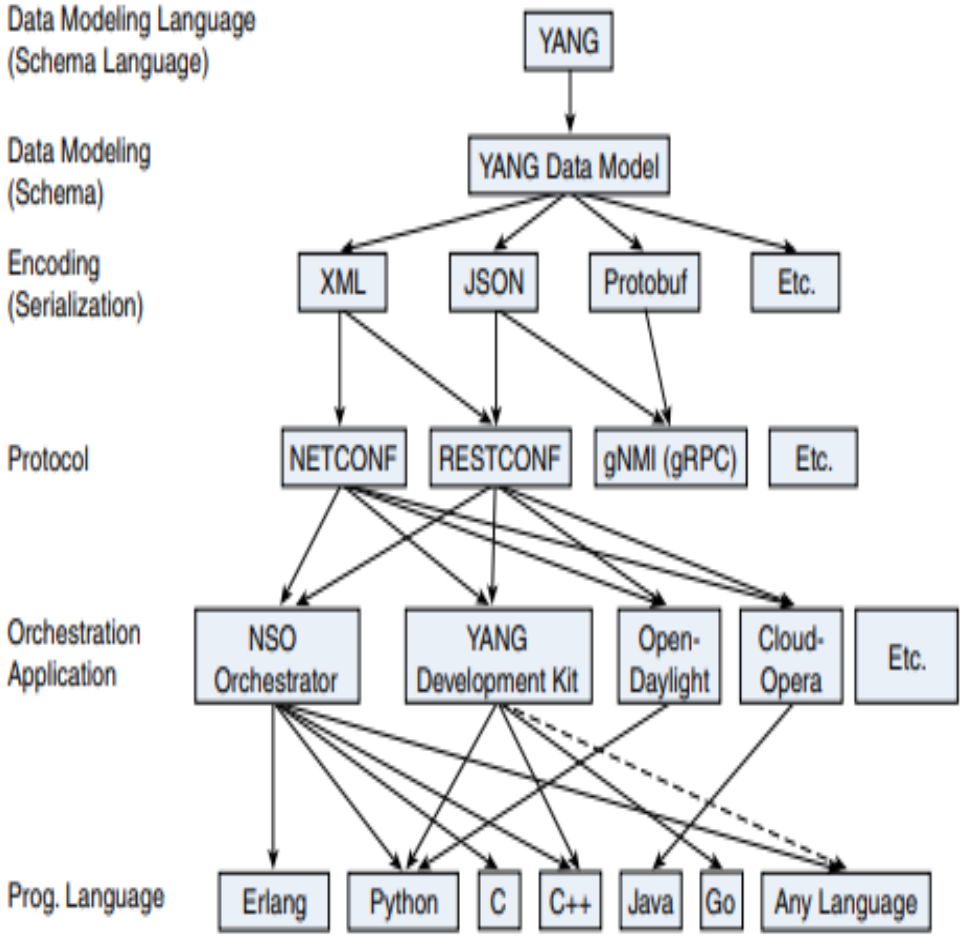






Data Modeling Languages (1)

- 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