Network Management

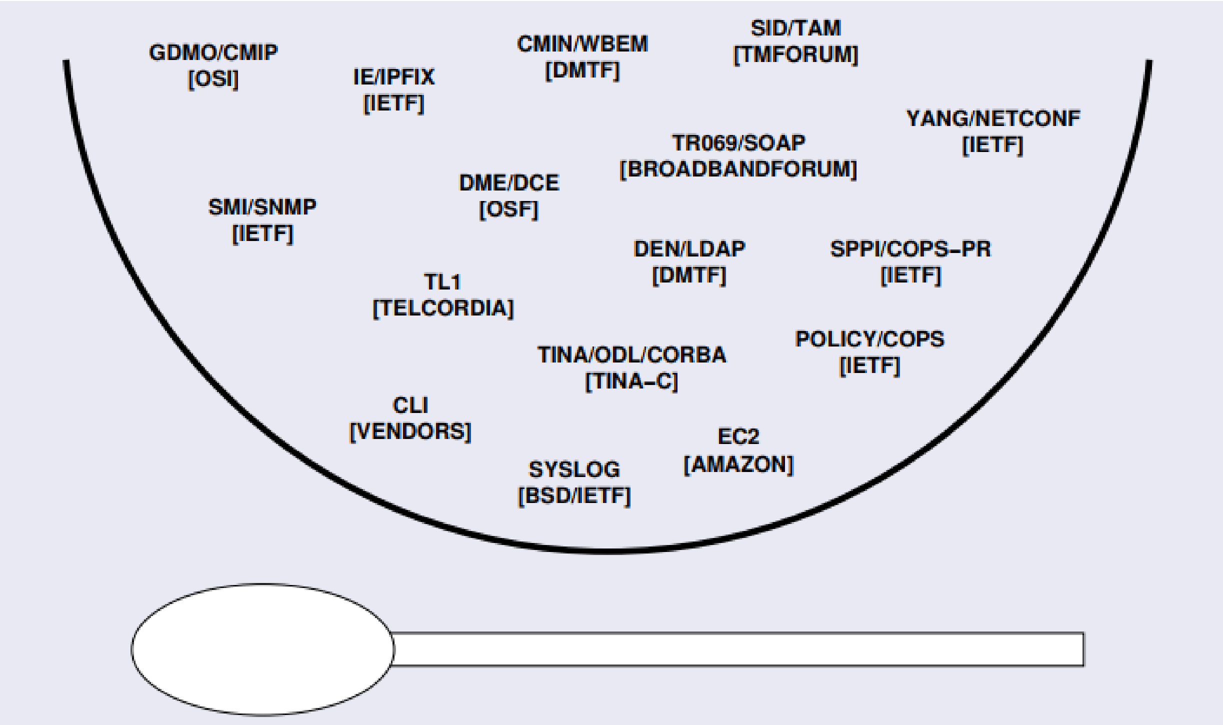
YANG as a motivation for open source?

Glenn Parsons, IEEE 802.1 WG chair

December 2017

Network Management protocol soup

- **Network Management** is the process of administering and managing the networks of one or many organizations.
  - fault analysis
  - performance management
  - provisioning of networks
  - maintaining the quality of service
- Several SDOs have defined an architecture:
  - ISO – FCAPS
  - TMF – FAB
- ...and protocols:



SID/TAM [TMFORUM]

GDMO/CMIP [OSI]

CMIN/WBEM [DMTF]

IE/IPFIX [IETF]

YANG/NETCONF [IETF]

TR069/SOAP [BROADBANDFORUM]

DME/DCE [OSF]

SMI/SNMP [IETF]

DEN/LDAP [DMTF]

SPPI/COPS–PR [IETF]

TL1 [TELCORDIA]

POLICY/COPS [IETF]

TINA/ODL/CORBA [TINA–C]

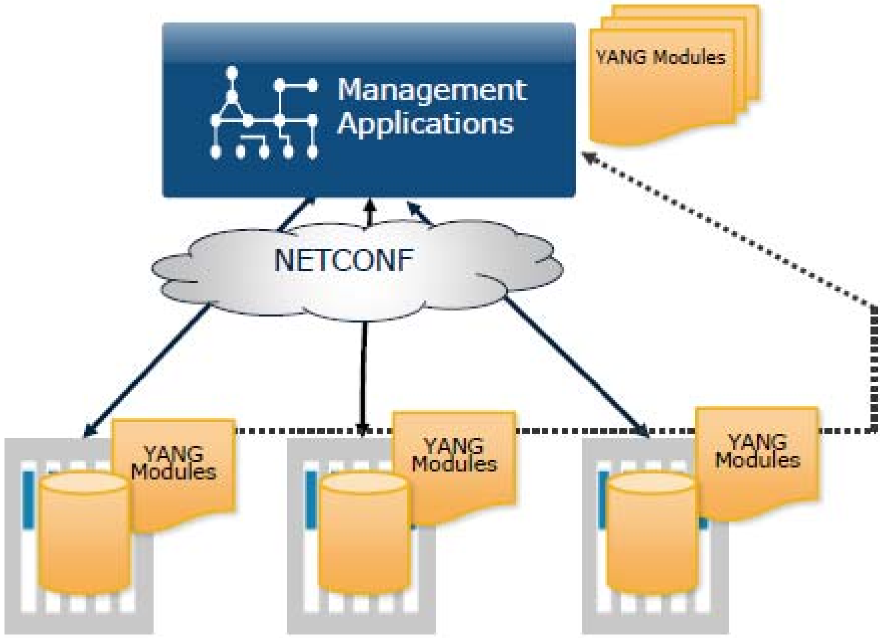
CLI [VENDORS]

EC2 [AMAZON]

SYSLOG [BSD/IETF]

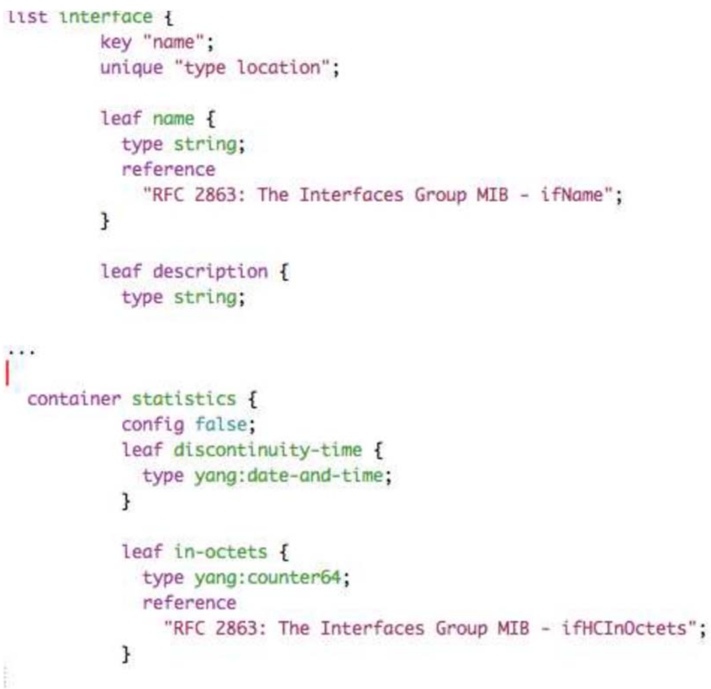
What is network management for?

- YANG
  - *YANG is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols (e.g., NETCONF, RESTCONF, etc.)*
- NETCONF
  - The Network Configuration Protocol (NETCONF) provides mechanisms to install, manipulate, and delete the configuration of network devices.
  - It is an example network configuration protocol



YANG –A Data Modeling Language for Networking

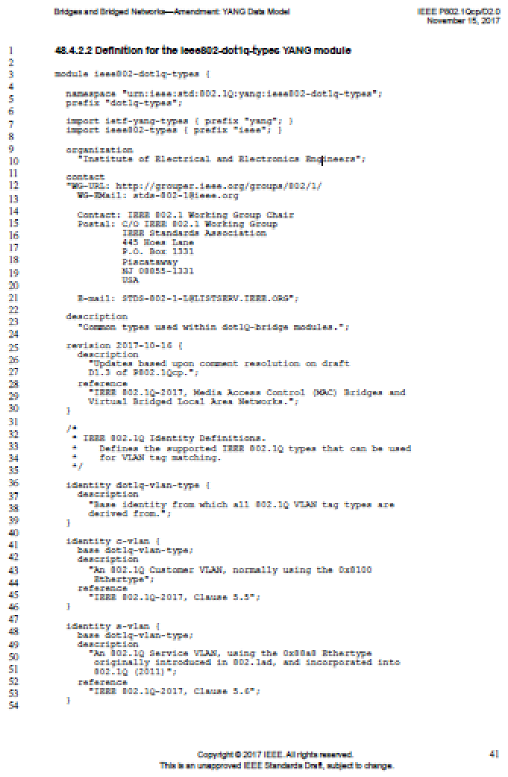
- Motivation in development was to satisfy Network Operator requirements (RFC 3535)
  - Ease of use, clear distinction between configuration data and operation state & stats, ability to fetch separately configuration/operational data from the device, etc.
- YANG is
  - Human readable and easy to learn
  - Hierarchical configuration
  - Reusable types and groupings
  - Extensibility through augmentation
  - Formal constraints for configuration validation
  - Data modularity through modules and sub-modules



```
list interface {
         key "name";
         unique "type location";

         leaf name {
           type string;
           reference
             "RFC 2863: The Interfaces Group MIB - ifName";
         }

         leaf description {
           type string;

...

  container statistics {
           config false;
           leaf discontinuity-time {
             type yang:date-and-time;
           }

           leaf in-octets {
             type yang:counter64;
             reference
               "RFC 2863: The Interfaces Group MIB - ifHCInOctets";
           }
```

Approved YANG model PARs in IEEE 802

- P802.1Xck – Port authentication
- P802.1Qcp – VLAN Bridges
- P802.1Qcw – TSN
- P802.1Qcx – CFM
- P802.1ABcu – LLDP
- P802.3.2 – Ethernet
- …



Bridges and Bridged Networks—Amendment: YANG Data Model

IEEE P802.1Qcp/D2.0
November 15, 2017

48.4.2.2 Definition for the ieee802-dot1q-types YANG module

```
module ieee802-dot1q-types {

  namespace "urn:ieee:std:802.1Q:yang:ieee802-dot1q-types";
  prefix "dot1q-types";

  import ietf-yang-types { prefix "yang"; }
  import ieee802-types { prefix "ieee"; }

  organization
    "Institute of Electrical and Electronics Engineers";

  contact
  "WG-URL: http://grouper.ieee.org/groups/802/1/
    WG-EMail: stds-802-1@ieee.org

    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
            P.O. Box 1331
            Piscataway
            NJ 08855-1331
            USA

    E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG";

  description
    "Common types used within dot1Q-bridge modules.";

  revision 2017-10-16 {
    description
      "Updates based upon comment resolution on draft
      D1.3 of P802.1Qcp.";
    reference
      "IEEE 802.1Q-2017, Media Access Control (MAC) Bridges and
      Virtual Bridged Local Area Networks.";
  }

  /*
   * IEEE 802.1Q Identity Definitions.
   *     Defines the supported IEEE 802.1Q types that can be used
   *     for VLAN tag matching.
   */

  identity dot1q-vlan-type {
    description
      "Base identity from which all 802.1Q VLAN tag types are
      derived from.";
  }

  identity c-vlan {
    base dot1q-vlan-type;
    description
      "An 802.1Q Customer VLAN, normally using the 0x8100
       Ethertype";
    reference
      "IEEE 802.1Q-2017, Clause 5.5";
  }

  identity s-vlan {
    base dot1q-vlan-type;
    description
      "An 802.1Q Service VLAN, using the 0x88a8 Ethertype
       originally introduced in 802.1ad, and incorporated into
       802.1Q (2011)";
    reference
      "IEEE 802.1Q-2017, Clause 5.6";
  }
```

Copyright © 2017 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

41

GitHub – YANG repository

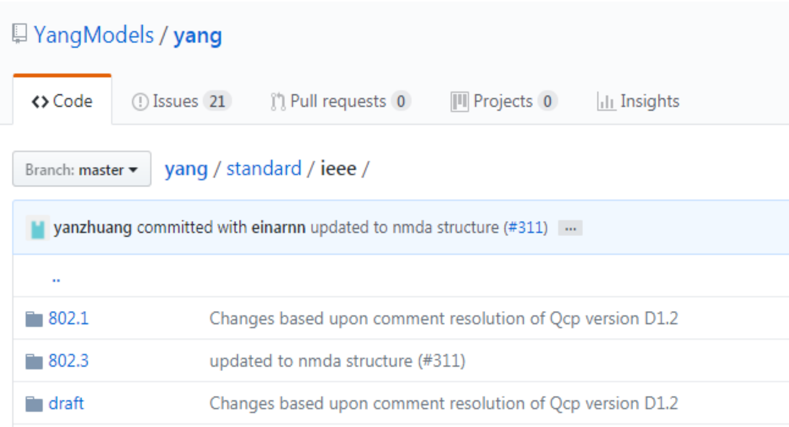
- **YANG Model directory for IETF, IEEE, MEF, BBF, …**
- **IEEE License:**
  - All files contained within this sub-directory are considered to be intended as IEEE Contributions.
  - All issues entered into the trouble ticket system for this directory are considered to be intended as IEEE Contributions.
  - All pull requests submitted for this directory are considered to be intended as IEEE Contributions.
  - All contributions to IEEE standards development (whether for an individual or entity standard) shall meet the requirements outlined in the IEEE-SA Copyright Policy
  - Copyright release for YANG modules: Users may freely reproduce the YANG modules contained under /experimental/ieee/ so that they can be used for their intended purpose.

YANG catalog – IETF led tooling

- A YANG model catalog and registry that allows users to find models relevant to their use cases from the large and growing number of YANG modules being published.
  - A YANG Validator, a web frontend that allows for validation of YANG modules.
  - A YANG Search, a web frontend that allows for searches over the content of the module catalog.
  - A YANG impact analysis tool.
  - View a module's metadata details.
  - An interactive YANG exploration tool that includes a YANG browser, RPC builder, and a YDK script generator to experiment with YANG modules
  - A YANG Regex Validator, a YANG regular expression validator to experiment with W3C YANG "pattern" statements

Source Code in IEEE standards

- Tooling from the Open Source community has provided a significant improvement in code development
- Current process has been enhanced to support this:
  - Development of code in GitHub with IEEE license indication – all code contributions are considered contributions to the standard
  - Publication of code in IEEE standard – pasted inline and attached to PDF as text code files
  - Publication of code on website or in GitHub
  - Copyright release to freely reproduce the YANG modules so that they can be used for their intended purpose.



yang / standard / ieee /

yanzhuang committed with einarnn updated to nmda structure (#311) ...

..	
802.1	Changes based upon comment resolution of Qcp version D1.2
802.3	updated to nmda structure (#311)
draft	Changes based upon comment resolution of Qcp version D1.2

What can full open source offer?

- Faster and continuous (aka agile) updating of YANG models
  - Errors in code, Additional enumerations in lists, Augmentations
- Flexibility on release cadence
- Much larger contributor / developer pool
- Early adoption by developers and network operators
- Added complexity for contributors?
  - Open Source Contribution License Agreement ***vs***
  - Standards Contribution Copyright Policy & Patent Policy
- Uncertainty on how to control direction of the open source specification
  - Consensus by balloting ***vs*** benevolent dictator
- A different revenue model