

# **YANG - Yet Another Next Generation**

**A.J. Ragusa**

# What is YANG?

---

YANG is a data modeling language, and it is used heavily for NETCONF/RESTCONF protocols but it can be used in other mechanisms as well.

YANG is an IETF RFC (6020 and 7950)

Lots of different systems use YANG internally (NSO, any NetConf/RestConf device) but it is more flexible than just for those

You can model almost anything in YANG

# What does it look like

— — —

```
module example-sports {

    namespace "http://example.com/example-sports";
    prefix sports;

    import ietf-yang-types { prefix yang; }

    typedef season {
        type string;
        description
            "The name of a sports season, including the type and the year, e.g,
            'Champions League 2014/2015'.";
    }

    container sports {
        config true;

        list person {
            key "name";
            leaf name { type string; }
            leaf birthday { type yang:date-and-time; mandatory true; }
        }

        list team {
            key "name";
            leaf name { type string; }
            list player {
                key "name season";
                unique number;
                leaf name { type leafref { path "/sports/person/name"; } }
                leaf season { type season; }
                leaf number { type uint16; mandatory true; }
                leaf scores { type uint16; default 0; }
            }
        }
    }
}
```

# Why do I care? - Let's have a fun use case!

---

You have a new David-Co router and there is a new cool feature on every interface called the “David” bit. All it does is replace “aluminum” with “aluminium” in every packet.

Let's presume you are using an SQL based Source of Truth (NetBox/GlobalNOC DB, etc...)

To utilize the “David” bit via your source of truth you have to modify the schema, modify the APIs, modify the WebUI, create a pull request and then argue with the devs to get your pull request pulled back in (or work off your own fork until the “David” bit becomes an RFC. Potentially this is a long time to get this rolled out.

# How YANG can help fix this problem

---

JavaScript modules exist to allow you to render YANG models as forms. So if you have a YANG model you can render it as a form in a webUI.

When the form is saved you save a JSON object that can be validated by your YANG Model and stores all of the data.

YANG is simple enough for even Network Engineers to be able to manipulate directly.

To add the “David” bit, we add a single leaf of type Boolean

Finally we update our template to handle the “David” bit. This updates our WebUI, Source of Intent, and can now push out the “David” bit to any interface on the network.

# Putting all the pieces together

---

Using YANG as your Source of Intent while you have a Source of “Operational” Truth (ie... NetBox/GlobalNOC DB) allows you to quickly modify your intent to support new protocols without requiring upstream projects to add support for it.

You can utilize the Source of Operational Truth to help fill out your Source of Intent

You can put many YANG models together to create a full configuration model for a device (depending on how you store the data)