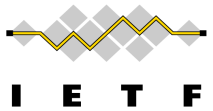
YANG, NETCONF, RESTCONF

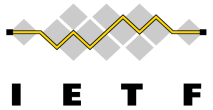
Hackathon 100

November 12, 2017

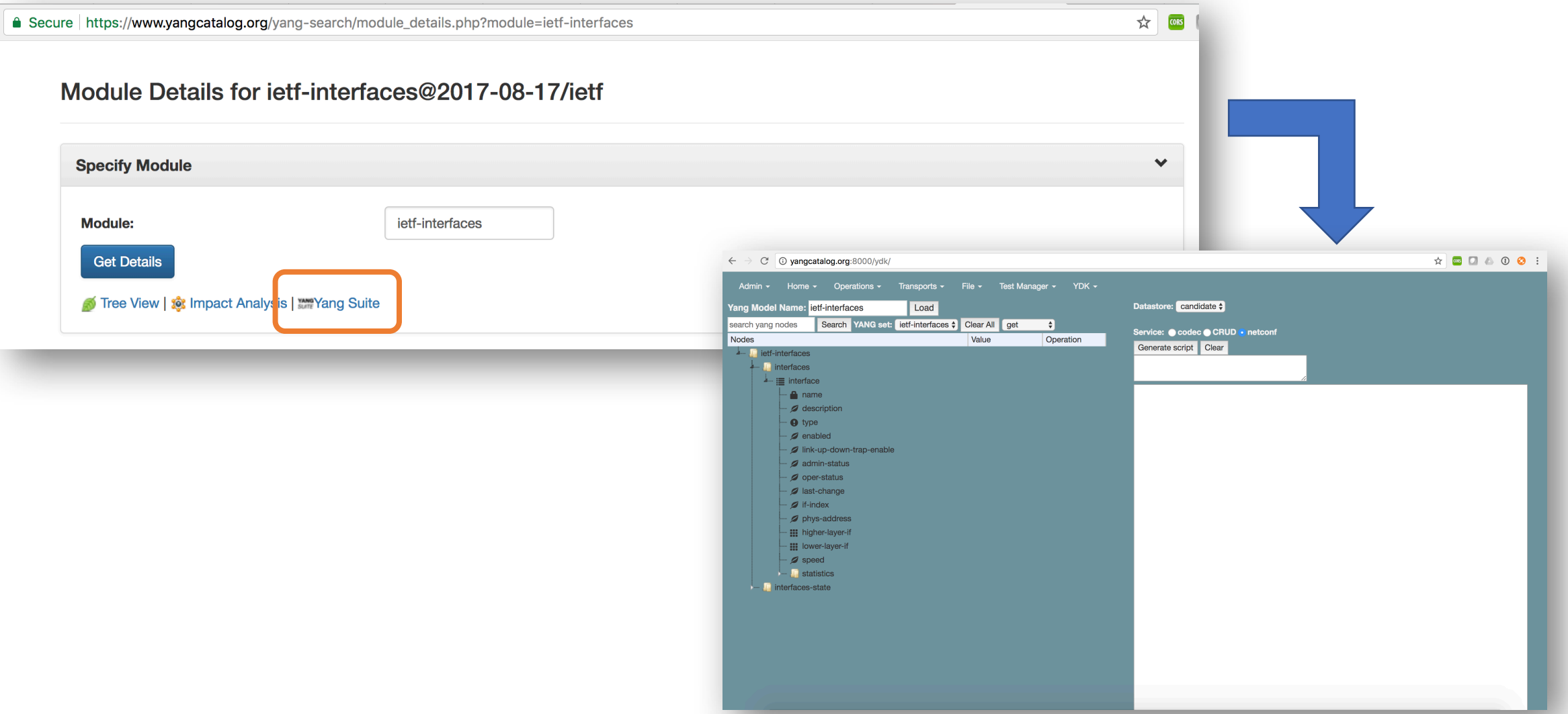


Summary

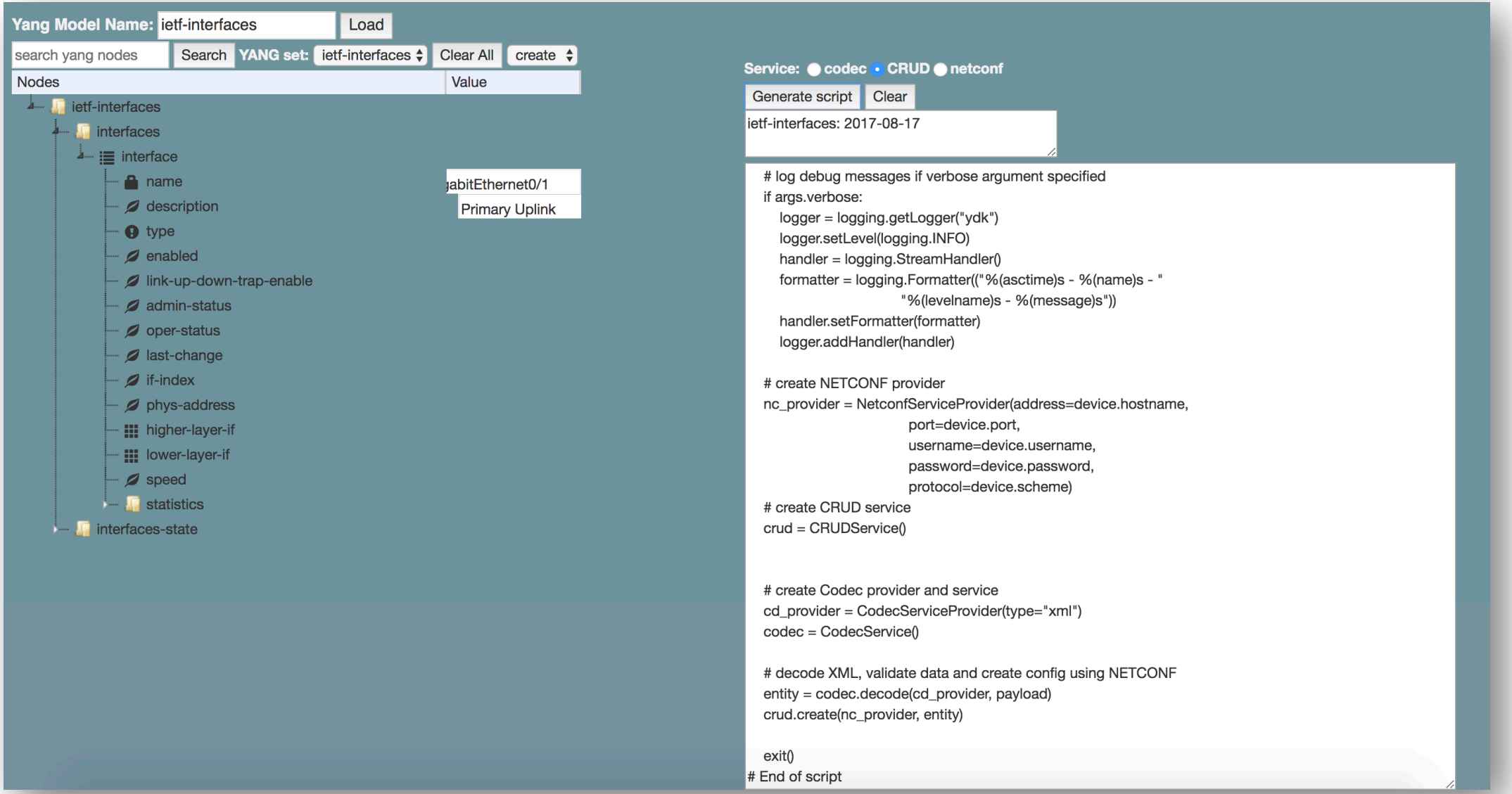
- Yang Suite integration with yangcatalog.org
- YDK integration with Yang Suite
- YANG module semantic versioning and diff views from YANG Catalog
- REST API for the YANG Regex validator
- Automated YANG test harness against confd and netconfd
- Integration of YANG Catalog with DataTracker (Henrik)



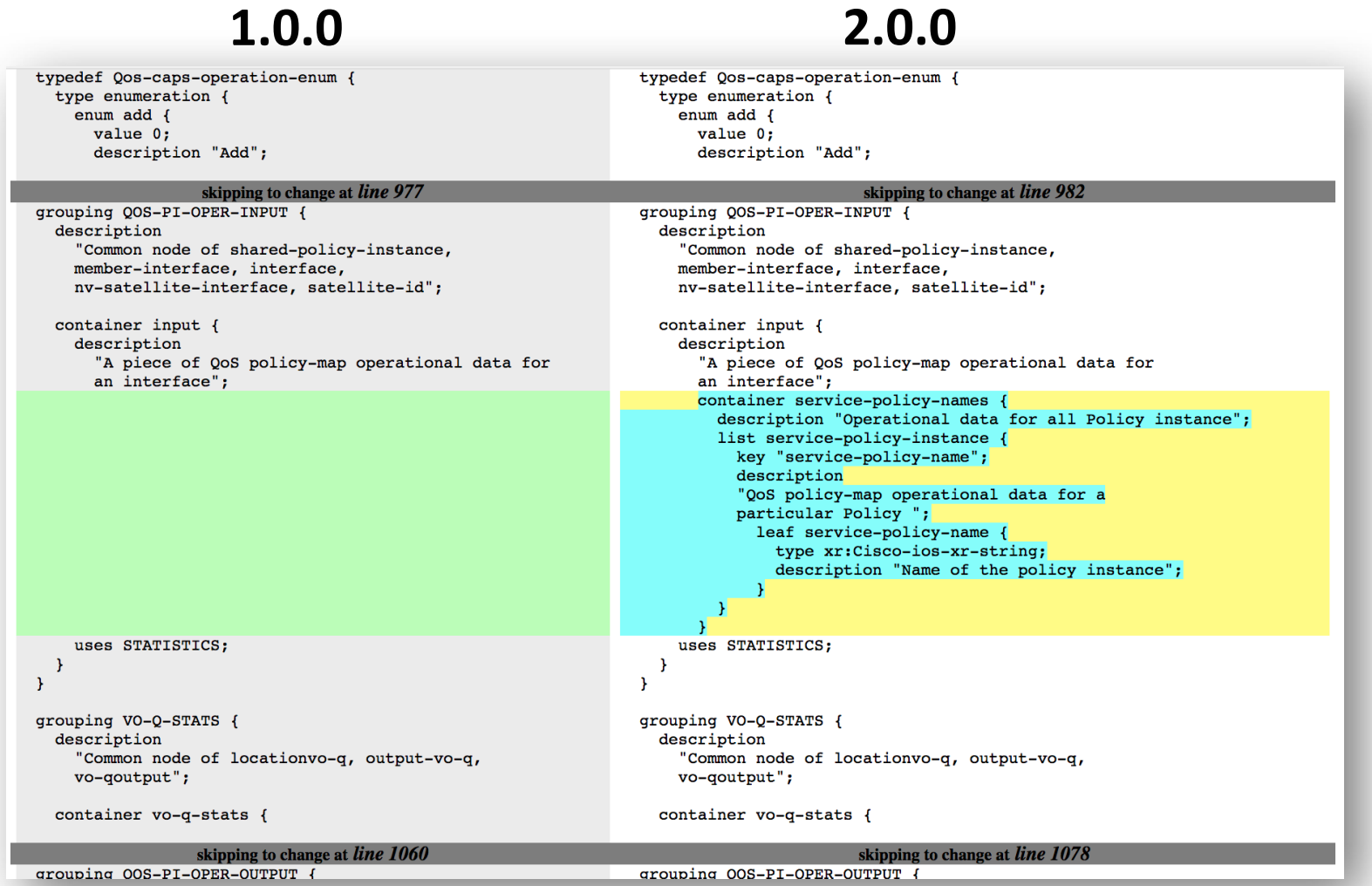
Yang Suite Integration With yangcatalog.org



YDK Integration With Yang Suite



Semantic Version Diffs



1.0.0

```
typedef Qos-caps-operation-enum {
  type enumeration {
    enum add {
      value 0;
      description "Add";
```

skipping to change at *line 977*

```
grouping QOS-PI-OPER-INPUT {
  description
    "Common node of shared-policy-instance,
    member-interface, interface,
    nv-satellite-interface, satellite-id";

  container input {
    description
      "A piece of QoS policy-map operational data for
      an interface";
```

```
    uses STATISTICS;
  }
}

grouping VO-Q-STATS {
  description
    "Common node of locationvo-q, output-vo-q,
    vo-qoutput";

  container vo-q-stats {
```

skipping to change at *line 1060*

```
grouping QOS-PI-OPER-OUTPUT {
```

2.0.0

```
typedef Qos-caps-operation-enum {
  type enumeration {
    enum add {
      value 0;
      description "Add";
```

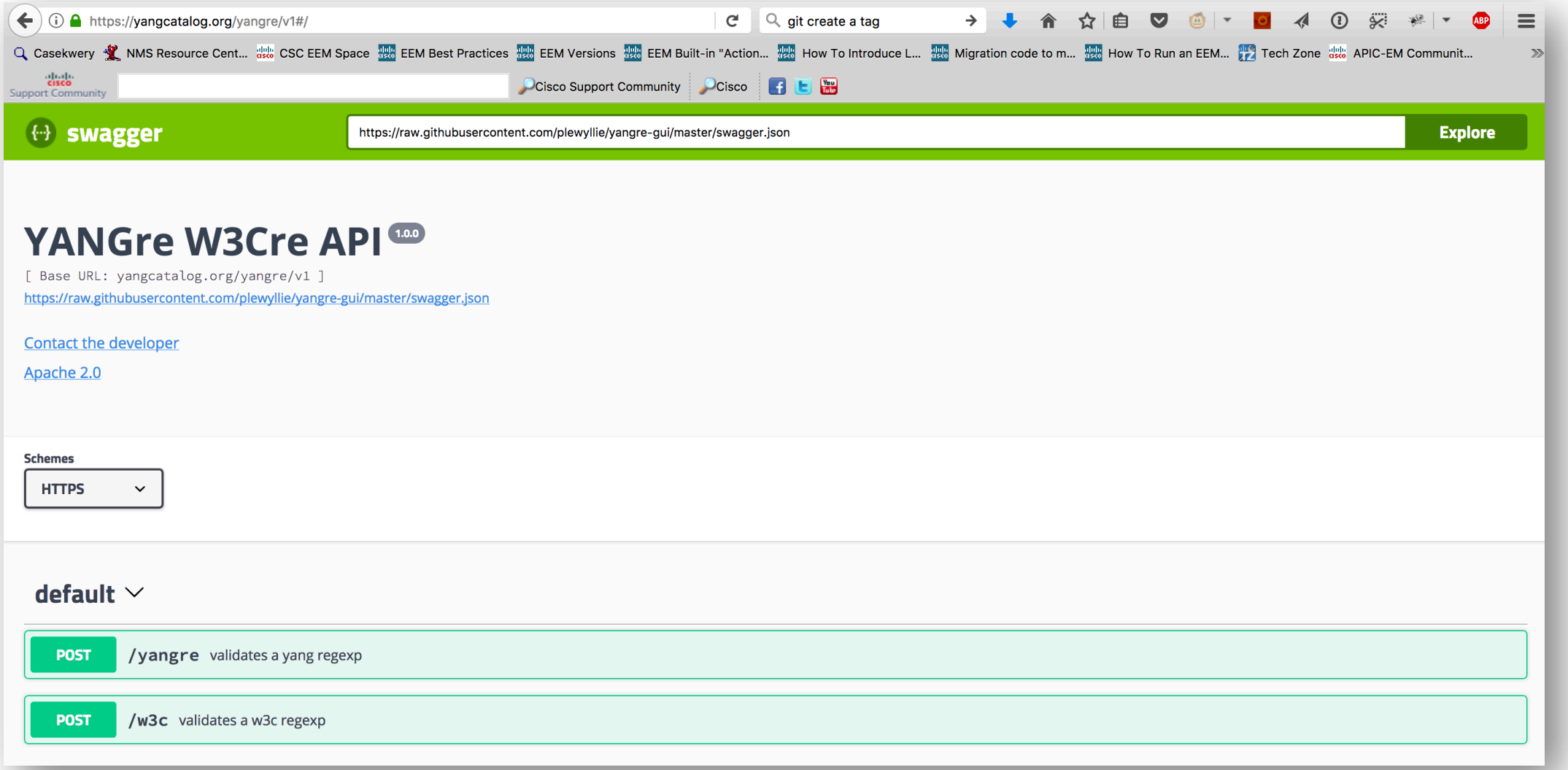
skipping to change at *line 982*

```
grouping QOS-PI-OPER-INPUT {
  description
    "Common node of shared-policy-instance,
    member-interface, interface,
    nv-satellite-interface, satellite-id";

  container input {
    description
      "A piece of QoS policy-map operational data for
      an interface";
      container service-policy-names {
        description "Operational data for all Policy instance";
        list service-policy-instance {
          key "service-policy-name";
          description
          "QoS policy-map operational data for a
          particular Policy ";
            leaf service-policy-name {
              type xr:Cisco-ios-xr-string;
              description "Name of the policy instance";
            }
        }
      }
    uses STATISTICS;
  }
}

grouping VO-Q-STATS {
  description
    "Common node of locationvo-q, output-vo-q,
    vo-qoutput";

  container vo-q-stats {
```

skipping to change at *line 1078*

```
grouping QOS-PI-OPER-OUTPUT {
```

REST API For Regex Validator



Automated Test Harness With Servers

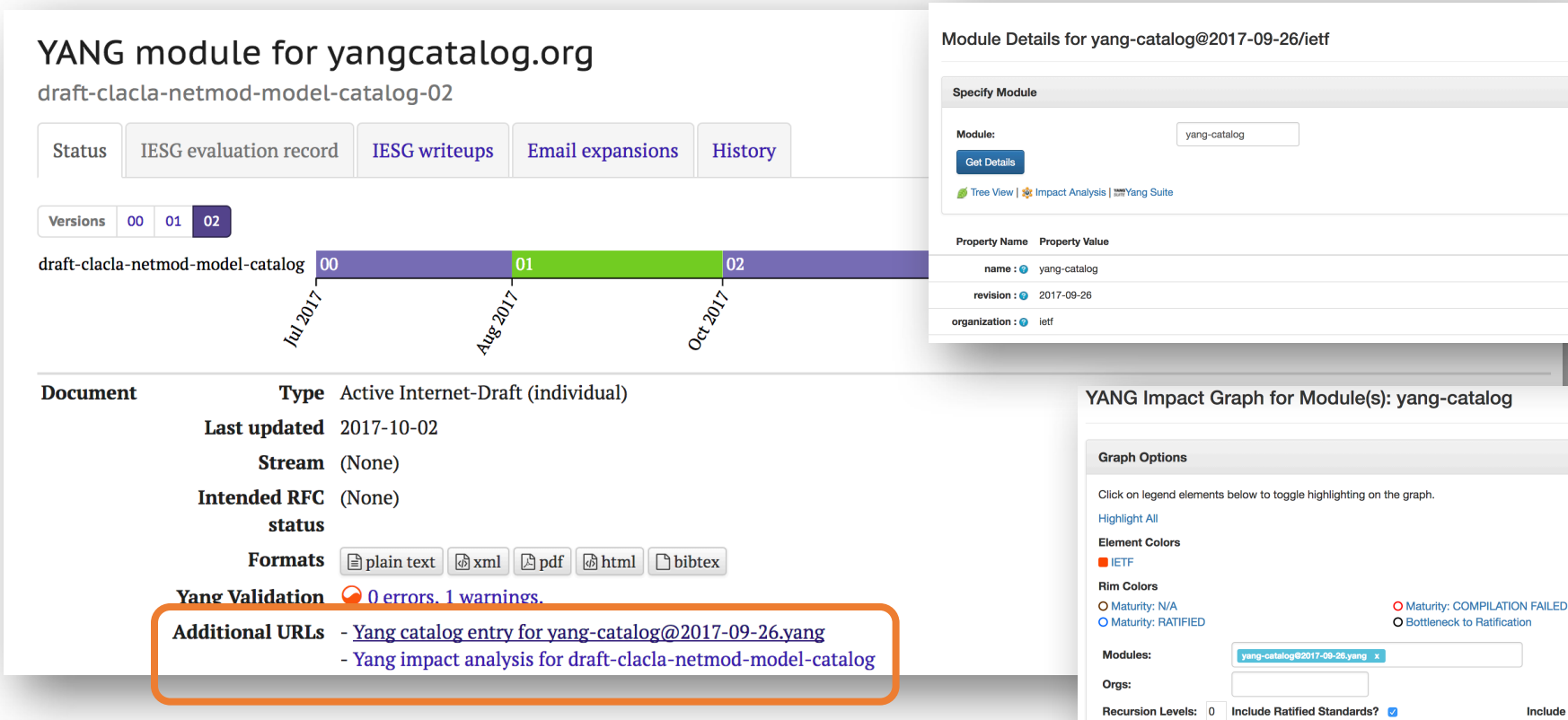
Objectives

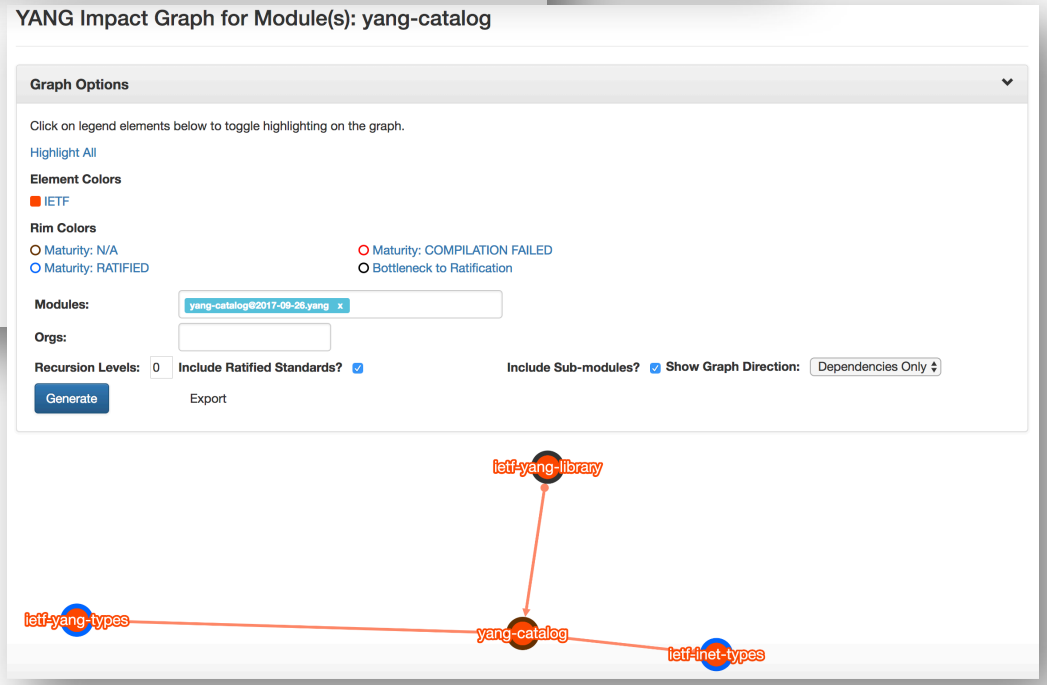
- Implement Network Management Datastore Architecture (NMDA) support for netconfd.
- NMDA support testcase based on litenc client and ietf-interfaces, ietf-ip, ietf-routing model.
- Add multisession support to libyangrpc.
- litenc: Implement netconf:base:1.1 capability (https://tools.ietf.org/html/rfc6241).

Progress

- The testcases (running against confd and netconfd based raw NETCONF session scripting):
  - Non-NMDA - https://sourceforge.net/p/yuma123/git/ci/master/tree/netconf/test/netconfd/ietf-routing/
  - NMDA - https://sourceforge.net/p/yuma123/git/ci/master/tree/netconf/test/netconfd/ietf-routing-bis/ (in progress)
- Server side implementation (in progress, complexity of more then 10 volatile drafts actively developed to implement support for the testcase)

Integration of YANG Catalog with DataTracker





How Much Was Done At This Hackathon?