Managing the evolution of IETF YANG Modules
Today

IETF is slowly publishing lots of YANG Modules in RFCs

- Protocol specific device modules
- Some work on service and network-wide YANG modules (LxSM, LxNM, topology, UNI)
Goal

Also the goal for OpenConfig

YANG

Produce a standard common API for configuring and managing devices
Problems

- IETF is not focused on a cohesive complete mgmt. API (i.e., DMs)
- RFC 7950 rules prevent fixing mistakes
- IETF is way too slow
- OpenConfig is getting much more market traction
- Fracture in the market slows adoption
OpenConfig

... has its own issues too

- Frequent churn
- Unclear governance path
- Technically less flexible (vs NMDA)
In the IETF pipeline

- Semantic Versioning
- Allow non-backwards-compatible changes
- YANG packages
Remaining Problems

- IETF is not focused on a cohesive complete mgmt. API
- RFC 7950 rules prevent fixing mistakes
- IETF is slow to standardize
Idea

Fundamentally change how the IETF manages code assets like YANG

I.e., stop treating them as documents.

- Store canonical YANG files somewhere else (not in RFCs), e.g., IANA, RFC Editor, Github, YANG Catalog?
- Develop YANG in Github, version using Semver
- Informational YANG RFCs describe module + short stable URL reference to canonical YANG
- Allow the YANG to evolve with bugfixes, errata, features
- Allow YANG RFC to evolve (e.g., living documents)
Idea 2

Manage the YANG as if it is a programmatic API

- Develop in Github
- Update RFCs as living docs

Stable Branch:
- New changes (pull requests)
- Must compile & be compatible
- Must achieve WG level consensus before merge

PS Branch:
- Periodically publish experimental branch
- Also store archival copies (IANA/RFC Editor)?
- IETF LC/IESG review?
Questions/Comments

- How would the IETF consensus process work?
- Where do we store canonical PS copy (RFC editor, IANA, Yang Catalog)?
- Will IETF have an existential crisis that these are not RFCs?
- Next steps: Experimental RFC to propose process?