TAPS YANG Model

draft-jholland-taps-api-yang

Jake Holland
Akamai
Motivation

• Code makes everything better
• "Abstract" has many levels
  • Finding the right level = very important
• Usable API needs some concreteness
  • More than taps-interface?
Basic Connection Example

```
{
   "ietf-taps-api:preconnection":{
      "properties":[
         {
            "type":"remote-host",
            "preference":"require",
            "host":["example.com"]
         }
      ]
   }
}
```

- Examples auto-parsed and validated during draft build
- Copy/paste for users & implementors
Config Format Benefits

- **Language Agnostic**
  - Portable Config Across Implementations
- **Runnable Test Suites**
- **Links Implementations to I-D**
  - YANG Draft becomes upstream for reference implementations
  - Anything missing/unimplementable gets more obvious
YANG Benefits

• Tool support
  • API generation
  • Parsing, validating config
  • State output -> downstream tool ecosystem

• Import Existing Models
  • 27 Existing Standards-track RFCs and counting
    • "relevant to taps": probably ~3-4 and counting
Next Steps

• Adopt?

• Relation to taps-interface?
  • Paired
    • Hard to keep sync’d
    • Many cross-references for descriptions
  • Integrated
    • Too big?
    • Weird ownership?

• Co-authors?