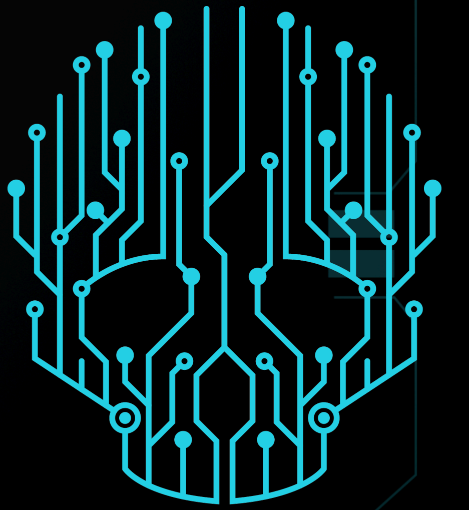
3 Years On: Open Standards, Open Source, Open Loop



David Ward dward@cisco.com

2017.11.16

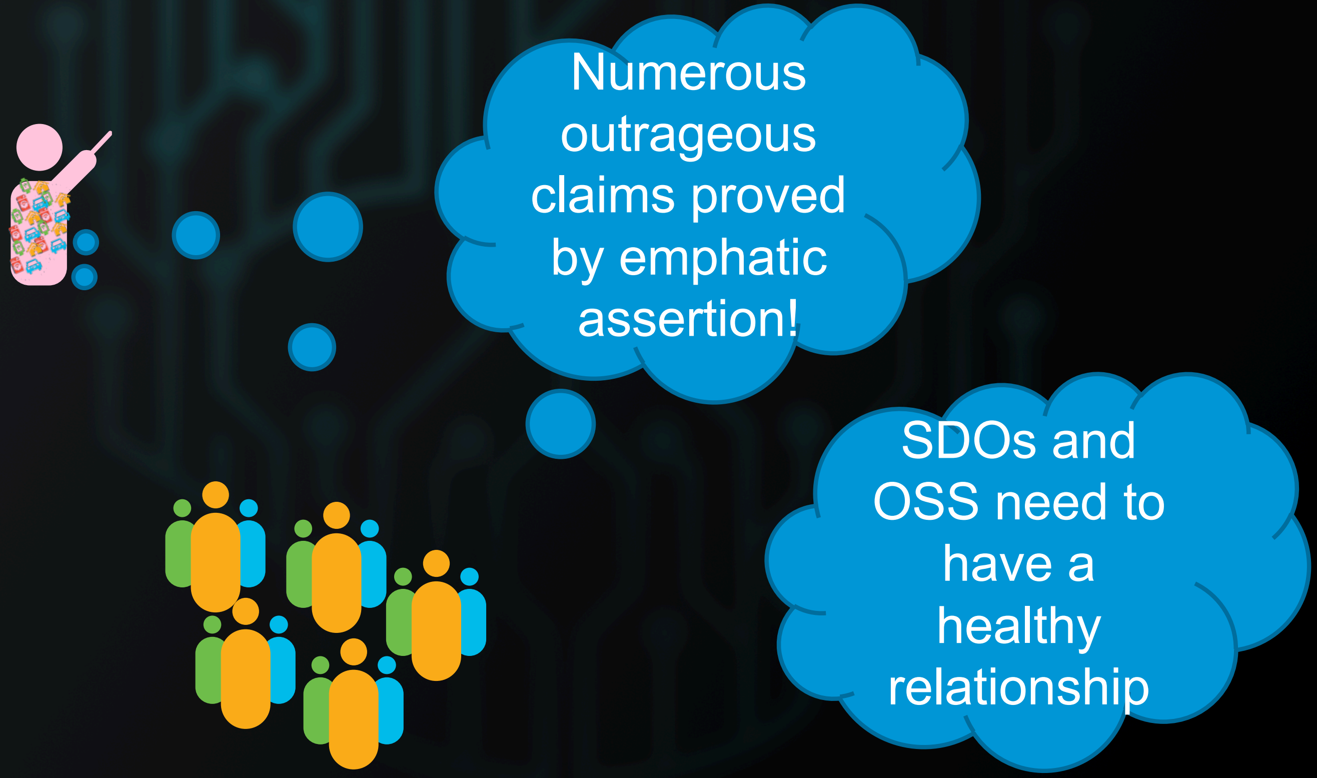
WHAT HAPPENED TO DWARD WRT OSS IN 3 YEARS?

Board Member: ATIS, Linux Foundation, Cloud Foundry, Open Networking Automation Platform (ONAP), OpenDayLight, OpenNFV, SNAS, FD.IO, Advanced Imaging Society, Entertainment Technology Center, ONOS, Platform for Network Data Analytics

Catalyzed: Critical Infrastructure Initiative, Hyperledger, Magen, CNFC, OCI

- Create a community of developers working on networking
- Bottomline: How can we move industry through infrastructure phase faster by producing communities, code and standards more efficiently. Industry and operators were stalling.
  - Run a number of different structures, funding models, community development

I. WHERE WE LEFT THINGS AT IETF91



Numerous outrageous claims proved by emphatic assertion!

SDOs and OSS need to have a healthy relationship

I. WHERE THEY WENT

- Still Potential for IETF to Engage w/ OSS communities
- OSS communities formed around networking
- “Stacks” “Controllers” “Virtual Fubars” quite fractured
- Community Aggregation
- Other SDOs working towards new trajectories that include OSS, direct contributions
- Industries have moved: OSS == SDO
- Stacks have new DIY cycle
- Career paths forged in OSS and now part of job satisfaction

POTENTIAL OUTCOME: IETF TAKES LEADERSHIP ROLE

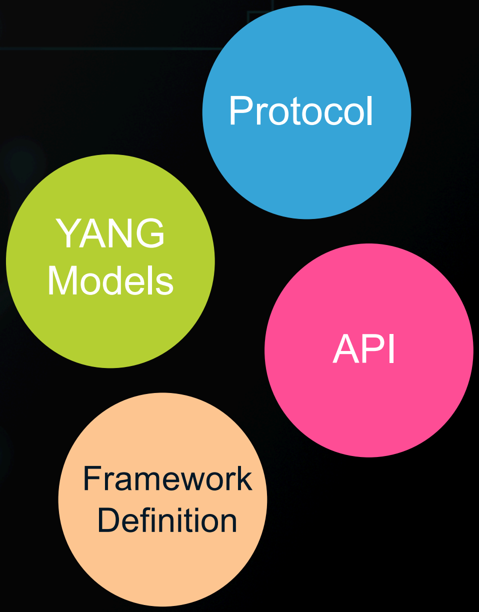
IETF91

The IETF has leverage-able experience in:

- Protocol definition
- Architecture definition
- Modeling languages

The IETF has RIGHT FOCUS

- Not too broad (e.g. not Health & Safety, etc.)
- Not too narrow (e.g. single service domain)



- APIs Platforms and Frameworks WILL be the future standards front for software driven network architectures
- Same standardization reasoning applies to these higher level concepts – system design, interoperability, and choice.

WAS: MAKING THE IETF AGILE

IETF91

Reform and restructure

- Cut the cycle time on EVERYTHING
- Fail fast and finish faster!
  - More BOF -> WG, WG->DONE
- Fewer “dead” drafts but more tangible/usable output

Adapt the liaison process

Generate more code and ideas

- Sponsor more research
- Encourage more demo (functionality and interoperability e.g. SRv6)

“rough consensus”
NOT
“parliamentary procedure”

Best Standards written as code written

“running code” NOW not “LATER”

STILL TRUE: SDO GEEKS != OSS GEEKS

IETF91

Realize you can’t do everything and augment IETF capabilities

- You don’t NEED to own everything – learn to depend on collaborators
- You probably don’t WANT to own everything – it will fundamentally change your core community
- Enable the organization to shift focus beyond what’s hot or deposited on the doorstep (be strategic)
- Open door, we’ll help standardize … must be more proactive

- OSS - Code may be “coin of the realm” but code isn’t normative.
- SDO - It’s hard to define APIs if you are not generating code!
- Modern Consensus is represented in the code by coders not at the mic or in .txt

- IETF consensus != Coder consensus model

SHOULD HAVE: FORM A COLLABORATIVE LOOP

IETF91

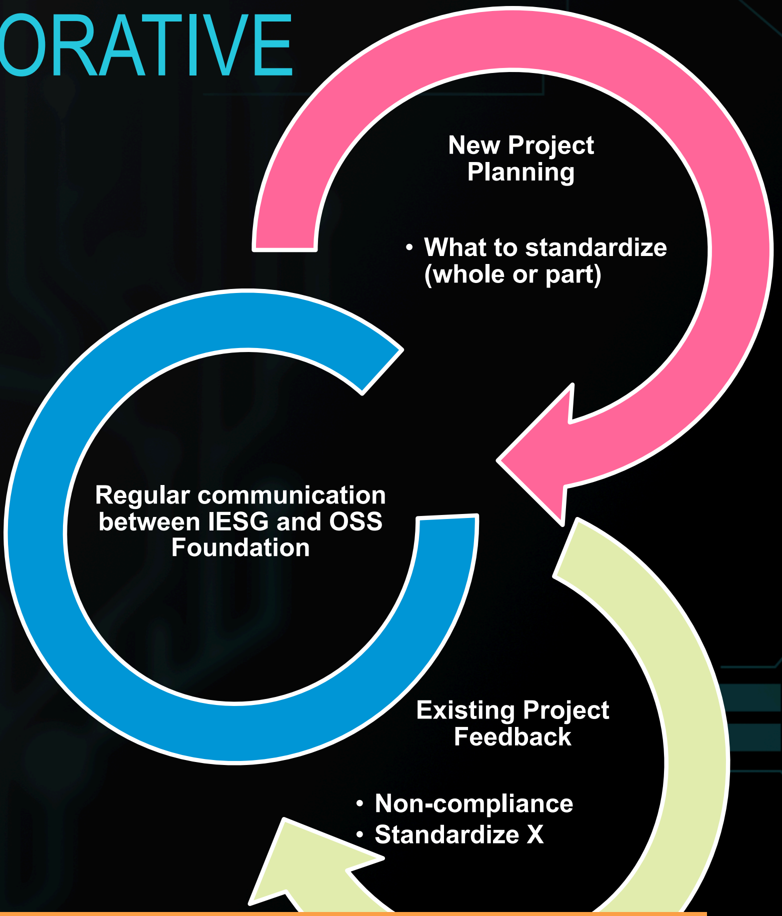
What the SDO/OSS relationship could look like

- Regular communication between IESG and reputable OSS Foundations
- Solicit OSS leaders to standardize (engage on reference implementations)?
- New Project Planning

  Relationship to existing or new standards
- Existing Project Review

  Standards compliance

  Standards potential



New Project Planning
• What to standardize (whole or part)
Regular communication between IESG and OSS Foundation
Existing Project Feedback
• Non-compliance
• Standardize X

Minimally Tooling built to have standards and dependencies in a RCS that is open and accepts contributions

EMBRACE ”GOOD” OPEN SOURCE

IETF91

Proven, Neutral 3rd Party Mgmt

Linux Foundation
Apache Foundation
OpenStack Foundation

Well-aligned, Productive Projects

Utility vs. “Dead-Code Repository”
Integrating IETF Protocols
Using IETF Tools (YANG)

Does this even apply to open source and how:

Is it stable, mature, and immutable (except for errata)? Stable means that there are not expected to be frequent updated versions. Mature is equivalent to being at least similar to a Proposed Standard RFC. Immutable means that the referenced content is not expected to change after RFC publication, except for minor error corrections. This might be achieved by referencing a particular dated version or a subsection of the document.

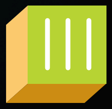
OPEN SOURCE PROJECTS

TAXONOMY

THE LINUX FOUNDATION

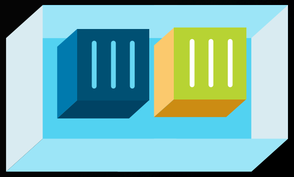
- Components

  Projects that address a narrowly defined problem whose output may be consumed as an atomic entity. Examples: VPP (virtual switch), a platform plug-in to integrate new hardware or software.



- Platforms

  Projects whose scope encompasses multiple components to yield a framework that can be adapted to meet a range of different user needs. Examples: OpenDaylight, FD.IO, PNDA and OpenStack



- Open Reference Platforms

  Projects that focus on the integration of platforms and components, and are primarily used to test, demonstrate, and validate broader solutions. Examples: OPNFV NFV reference platform and MEF OpenLSO reference platform, ONAP.

WHAT HAPPENED SINCE?
LINUX SDO/OSS DESIRE TO HARMONIZE

Harmonizing

Open Source

And Standards

In the Telecom World

MAY 2017

https://www.linuxfoundation.org/blog/new-linux-foundation-white-paper-harmonizing-open-source-and-standards-in-sdn/

HARMONIZATION

DRIVERS AND RECOMMENDATIONS – BIT TOO HIGH LEVEL

SDN & NFV DRIVEN

- Rise of the “Software Defined” Operator
- Software IS Eating the World
- Waterfall is giving way to Agile
- Internet Time is giving way to Cloud Time

RECOMMENDATIONS

- ***Communications***

  … communications, communications to resolve the cultural differences between standards and open source, with a focus on convergence
- ***Multi-SDO/open source activities***

  …such as the Information Modeling initiative involving TMForum, ONF, ETSI NFV, OSM and OPNFV, among others
- Less formality

  …and renewed attention on definitive outcomes
- Cooperation on a revised technology adoption methodology

  …that blends standards, open source, operator-contributed use cases, and vendor-technical contributions

LF NETWORK PROJECT

EXAMPLE OF CONSOLIDATION AND ORG EVOLUTION

CURRENT PROBLEM(s)

Too many networking foundations at Linux Foundation

- Too many checks, checks too big
- Too much total expenditure
- Too many yearly “events”

SOLUTION: NETWORK PROJECT

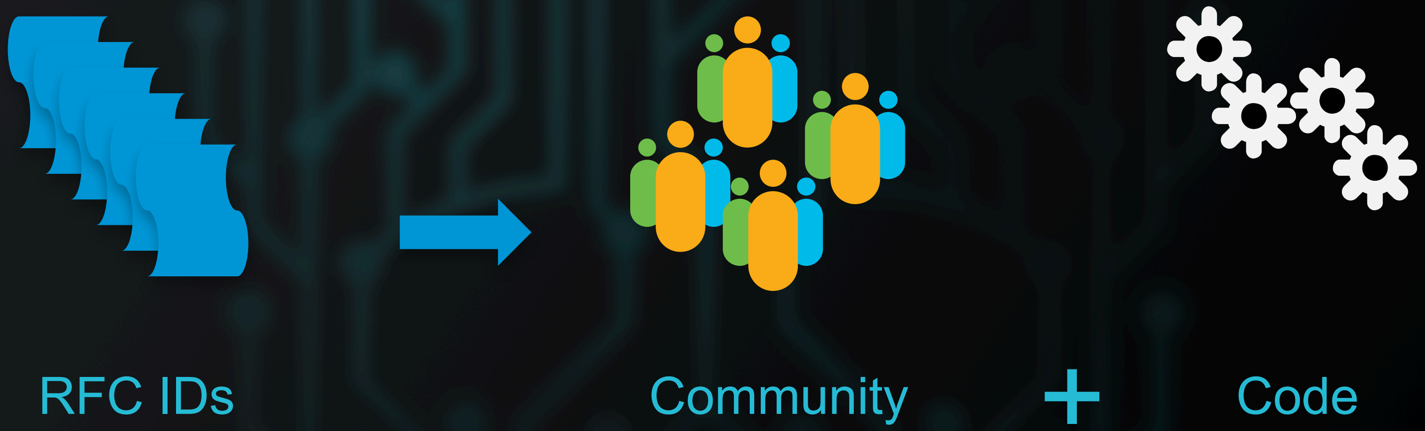
- ODL, FD, OPNFV, ONAP are expected to roll in initially
  - Others may or may not follow
- The technical governance of those communities stays unchanged
- One foundation and board making the business decisions in the background

II. WHAT SPECIFICALLY HAS HAPPENED SINCE IETF 91 IN THE IETF?

- Referencing OSS in drafts/RFCs
- Live Standards - YANG Catalogue

NORMATIVE REFERENCES PROPOSAL

SPEED MISMATCH



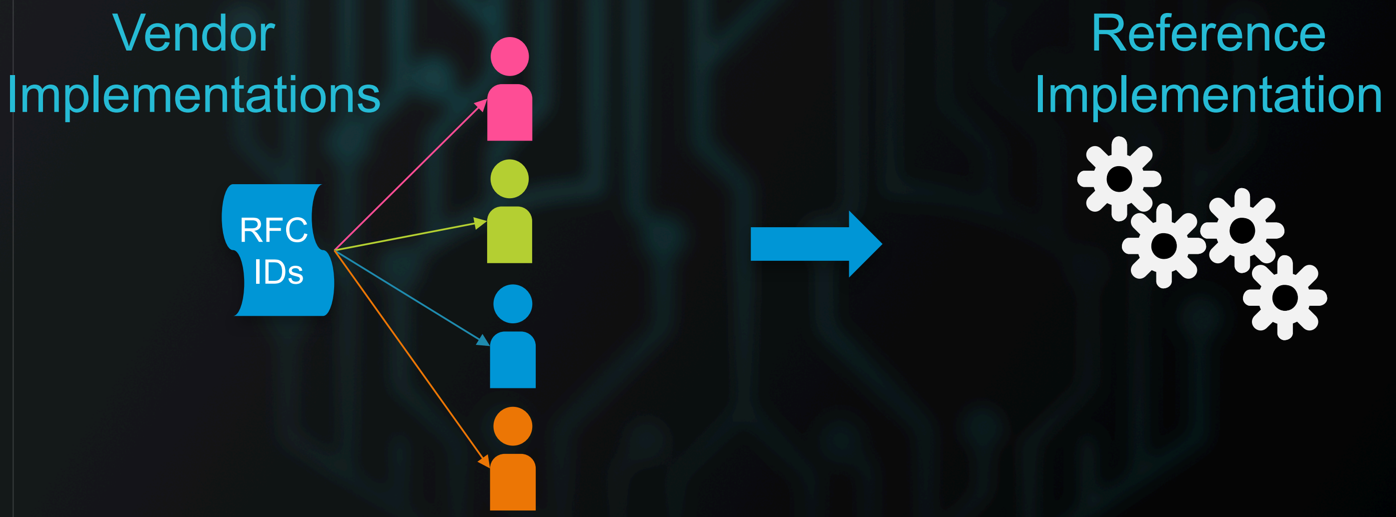
- OSS projects are not necessitating formal documents or structure. Their ”authority” are in their community and code and viable functionality
- There is a speed mis-match in these “instruments”.

Pitfalls To Avoid

Becoming “scribe” for existing code OR missing shift

NORMATIVE REFERENCES PROPOSAL

IDENTITY CRISIS/CULTURAL CHANGE



Vendor Implementations

RFC IDs

Reference Implementation

- OSS projects do not always generate “multiple interoperable instances” but instead one iteratively derived reference implementation

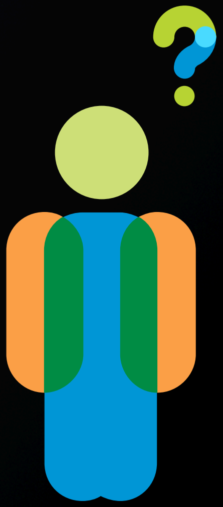
First to broadly acceptable solution will generally win. And an alternate solution has to be better to displace it or new/better implementation to overcome issues

NORMATIVE REFERENCES PROPOSAL

FINISHING THE DIALOGUE

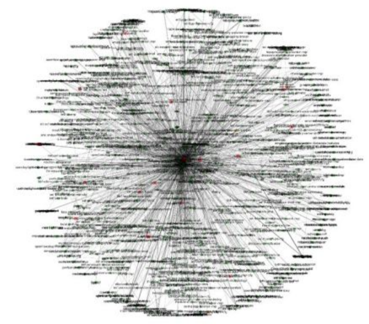
Other questions that arise:

- How do WG pick projects? Purely draft to draft?
- What are the health/longevity metrics for tech or community?
- What is an OSS reference? Is it one-size-fits-all or are there nuances: The transport protocols are one thing, APIs are another, Schemas are another, state machines, events, tooling, etc.
- There may be different aspects of an OSS project that you want to reference and not others (at the very least)



https://datatracker.ietf.org/doc/draft-atlas-external-normref/

POST91: YANGCATALOG - IETF EXPERIMENTS WITH LIVE, OPEN ORGANIZATION AND OSS TOOLING



YANGCATALOG.ORG

A SUCCESS STORY OF MODEL DRIVEN NETWORKING

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.

UTILITY of Tooling

- NETCONF and REST (not RESTCONF-compliant yet) server loaded with the YANG module from draft-clacla-netmod-model-catalog

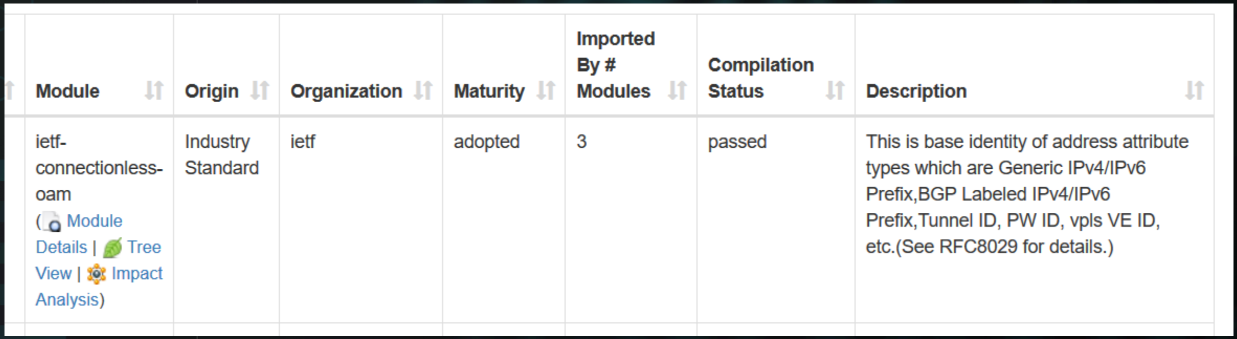
1. YANG Validator, a web frontend that allows for validation of YANG modules & IETF drafts.
2. YANG Search, a web frontend for searches over the content of the module catalog.
3. YANG Metadata. View a module's metadata details.
4. YANG Impact Analysis tool.
5. YANG Suite that includes a YANG browser and RPC-builder application
6. YANG Regex Validator to experiment with W3C YANG "pattern" statements

- APIs accessible via REST with JSON results

Programming and monitoring the PNF (and VNF)!

YANGCATALOG.ORG

METADATA AND HEATH METRIC



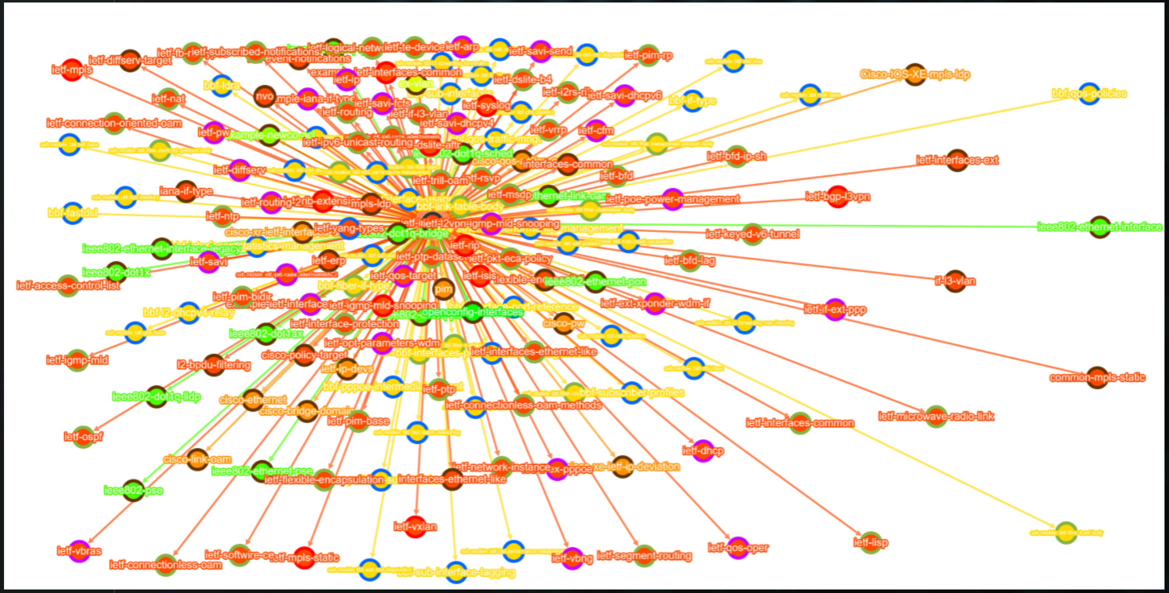
Module	Origin	Organization	Maturity	Imported By # Modules	Compilation Status	Description
ietf-connectionless-oam (Module Details | Tree View | Impact Analysis)	Industry Standard	ietf	adopted	3	passed	This is base identity of address attribute types which are Generic IPv4/IPv6 Prefix,BGP Labeled IPv4/IPv6 Prefix,Tunnel ID, PW ID, vpls VE ID, etc.(See RFC8029 for details.)

Extractable or manually maintained metadata, to assess a YANG module

Are there similar health metrics for OSS projects?

YANGCATALOG.ORG

CROSS SDO/OSS/VENDOR DEPENDENCY MAP



- 🟧 IETF
- 🟧 CISCO
- 🟨 BBF
- 🟨 INDEPENDENT
- 🟩 NEWCO
- 🟩 IEEE
- 🟩 OPENCONFIG

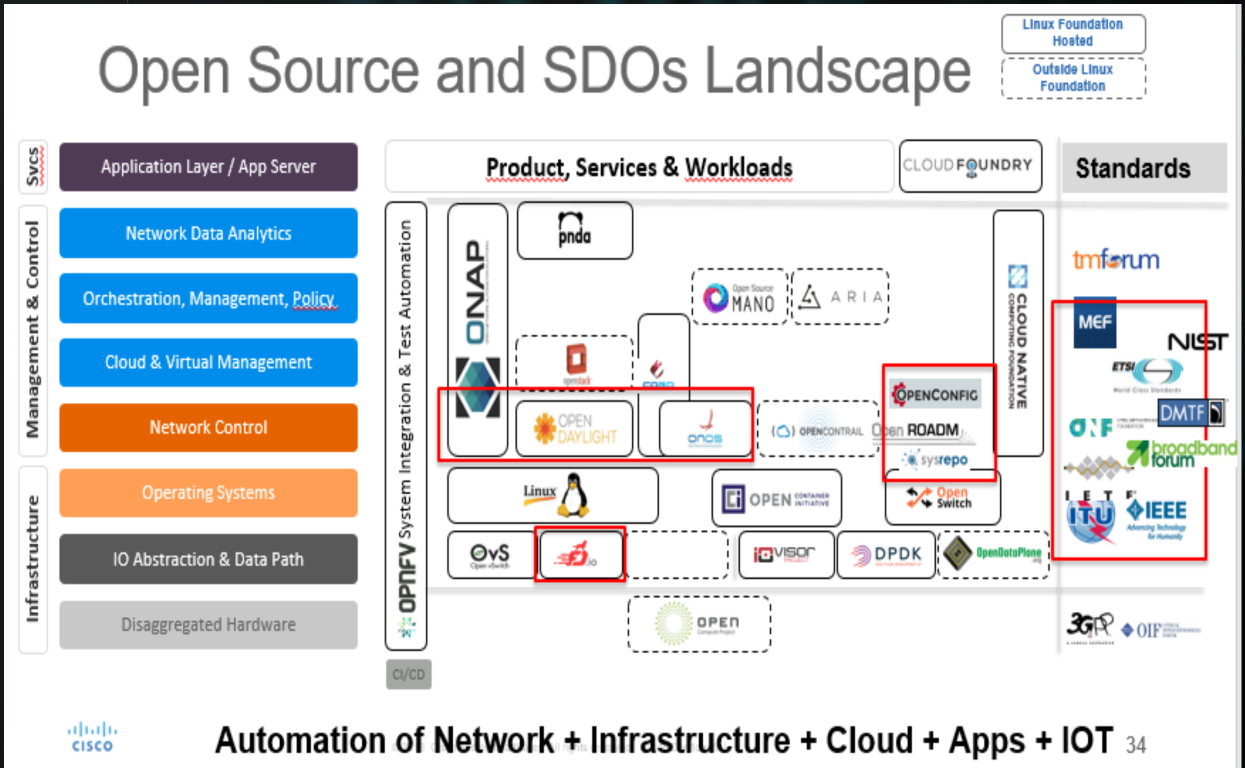
Rim Colors

- ○ Maturity: ADOPTED
- ○ Maturity: N/A
- ○ Maturity: RATIFIED
- ○ Maturity: INITIAL
- ○ Maturity: COMPILATION FAILED
- ○ Bottleneck to Ratification

- Impact analysis visualisation in case of (non backward compatible) changes
- Organizations: IETF, IEEE, BBF, MEF, openconfig, vendor, etc.
- Operators: for service composition

MANUAL SDO/OSS DEPENDENCY MAP

EVERY AD/IESG/IAB SHOULD HAVE ONE, PER TECHNOLOGY



Open Source and SDOs Landscape

Automation of Network + Infrastructure + Cloud + Apps + IOT 34

- If it can’t be done via tools, it must be manual.
- The ones in red (here) are YANG-based, for data modeling driven management.

YANGCATALOG LESSON

THE RFC # SHOULD NOT BE THE ONLY METRIC FOR IETF SUCCESS

- In OPS, the product of the IETF is not just the RFC ids but the YANG models inside
- Adopt an operations focus - deliver the goods in a non-ambiguous manner
- Add value in toolchain and metadata
- Encourage community interaction through Collaborative work externally

"automation is a good as the data models, the model metadata, and the toolchain"

Number of standard-based YANG modules : 300
Number of unique YANG modules into yangcatalog: 2675

YANGCATALOG LESSON

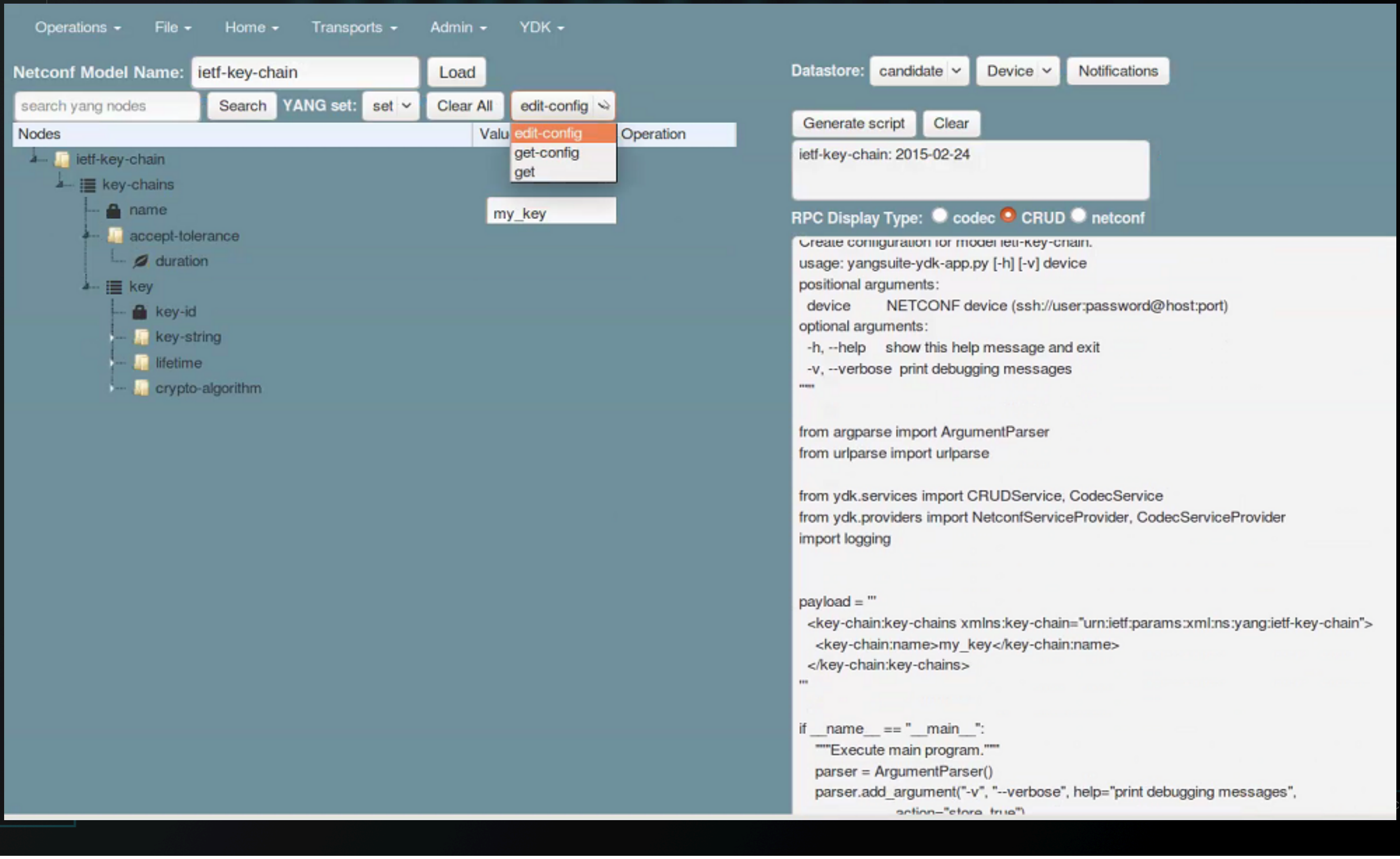
DEVELOP FOR YOUR SDO AND OSS “CUSTOMERS”

- Operators, who want to automate following the data-model driven management paradigm
- SDOs/OSS projects that want to integrate YANG models and related technologies
- All WGs in the IETF, who wants to create YANG modules
- Everybody wants to understand the technology
- Developers/vendors who want to do YANG module testing

YANG catalog created for all these audiences in mind: module users, module designers, module testers, with an educational goal

YANGCATALOG LESSON

SIMPLIFICATION THROUGH TOOLS, EDUCATION + TESTING



Netconf Model Name: ietf-key-chain

YANG set: set

Operation: edit-config

name: my_key

Datastore: candidate

ietf-key-chain: 2015-02-24

RPC Display Type: ○ codec ● CRUD ○ netconf

```
Create configuration for model ietf-key-chain.
usage: yangsuite-ydk-app.py [-h] [-v] device
positional arguments:
  device        NETCONF device (ssh://user:password@host:port)
optional arguments:
  -h, --help     show this help message and exit
  -v, --verbose  print debugging messages
"""

from argparse import ArgumentParser
from urlparse import urlparse

from ydk.services import CRUDService, CodecService
from ydk.providers import NetconfServiceProvider, CodecServiceProvider
import logging


payload = '''
  <key-chain:key-chains xmlns:key-chain="urn:ietf:params:xml:ns:yang:ietf-key-chain">
    <key-chain:name>my_key</key-chain:name>
  </key-chain:key-chains>
'''

if __name__ == "__main__":
    """Execute main program."""
    parser = ArgumentParser()
    parser.add_argument("-v", "--verbose", help="print debugging messages",
                        action="store_true")
```

YANGCATALOG

PROCESS AND FUNDING == FUTURE SUCCESS

- Process: Closer integration
  - Ex: Run the IETF process on YANG modules
  - Ex: No obsolete tags in YANG modules
  - Tighter integration in the IETF datatracker and IETF process
- Funding
  - Developed during IETF hackathon and with private funding
  - How to move from experimentation to maintenance?
    Note: The YANG catalog is as good as its content
  - How to fund the next set of tools?

Is the IETF able to maintain the tools or is the catalyst to build an open source community and/or non-profit foundation

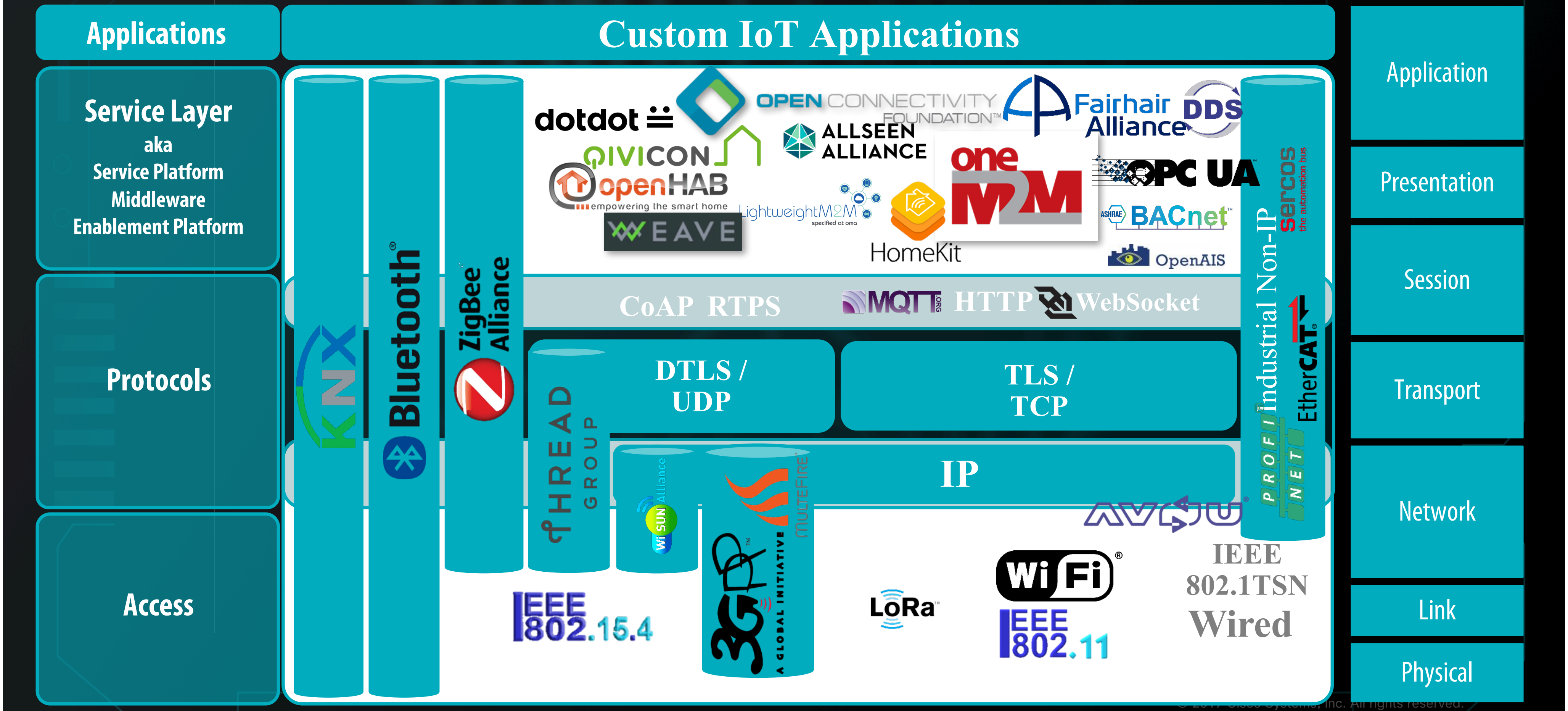
POST91: ONEM2M EXPERIENCE:
ANOTHER EXPERIMENT A LA YANG CATALOGUE

PATCHWORK JUNGLE OF CONSORTIA, STANDARDS, OS-PROJECTS

- Which groups actually specify technology, which are just doing marketing & promotion?
- Which technologies are used / will be used in M2M/IoT?
- Which technologies are overlapping or complementing each other?

openHAB empowering the smart home
OCEAN
WEAVE
MQTT.ORG
W3C
LightweightM2M specified at oma
sercos the automation bus
QIVICON
dotdot
ioTivity
OASIS Advancing open standards for the information society
oneM2M
iot eclipse.org
OPEN CONNECTIVITY FOUNDATION
AVNU
DDS
Ethernet
ZigBee Alliance
CoAP
HomeKit
OpenAIS
ALLSEEN ALLIANCE
WebSocket
RTPS
OMG WE SET THE STANDARD
HTTP
IETF
ASHRAE BACnet
industrial internet CONSORTIUM
ISO
IEC
OM2M Connecting things
KNX
Bluetooth
THREAD GROUP
MULTEFIRE
PLATTFORM INDUSTRIE 4.0
Fairhair Alliance
IEEE
CABA Continental Automated Buildings Association
OPEN DAYLIGHT
IoTDM
3GPP A GLOBAL INITIATIVE
atis
OS-IoT
PROFI NET
Fieldbus Foundation
Wi SUN Alliance
Wi Fi
LoRa
HART
FIELDCOMM GROUP Connecting the World of Process Automation

ORGANIZATIONS IN IOT STACK



Applications

Service Layer
aka
Service Platform
Middleware
Enablement Platform

Protocols

Access

Custom IoT Applications

CoAP RTPS

HTTP

WebSocket

DTLS /
UDP

TLS /
TCP

IP

Industrial Non-IP

IEEE
802.1TSN
Wired

Application

Presentation

Session

Transport

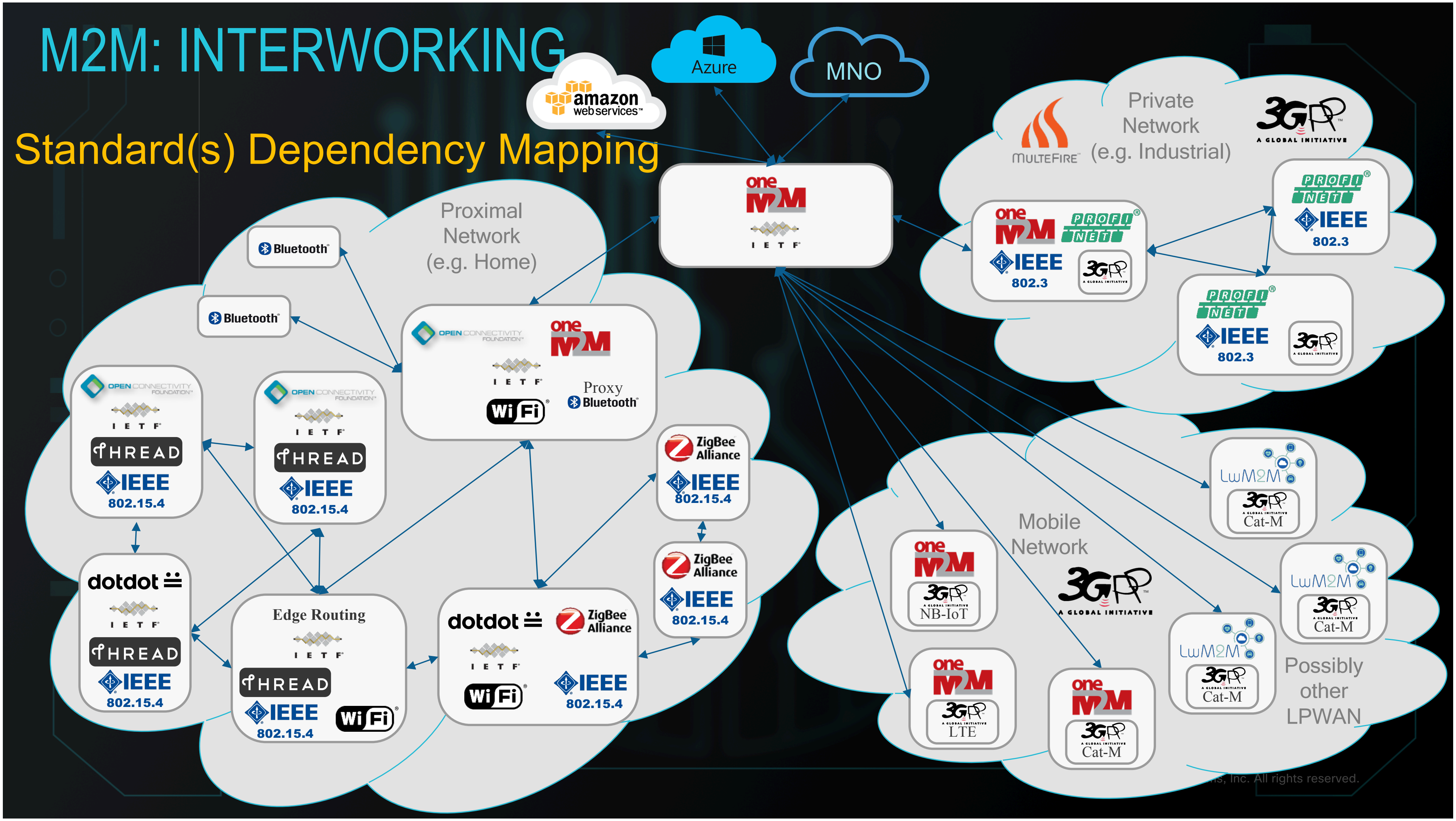
Network

Link

Physical

M2M: INTERWORKING

Standard(s) Dependency Mapping



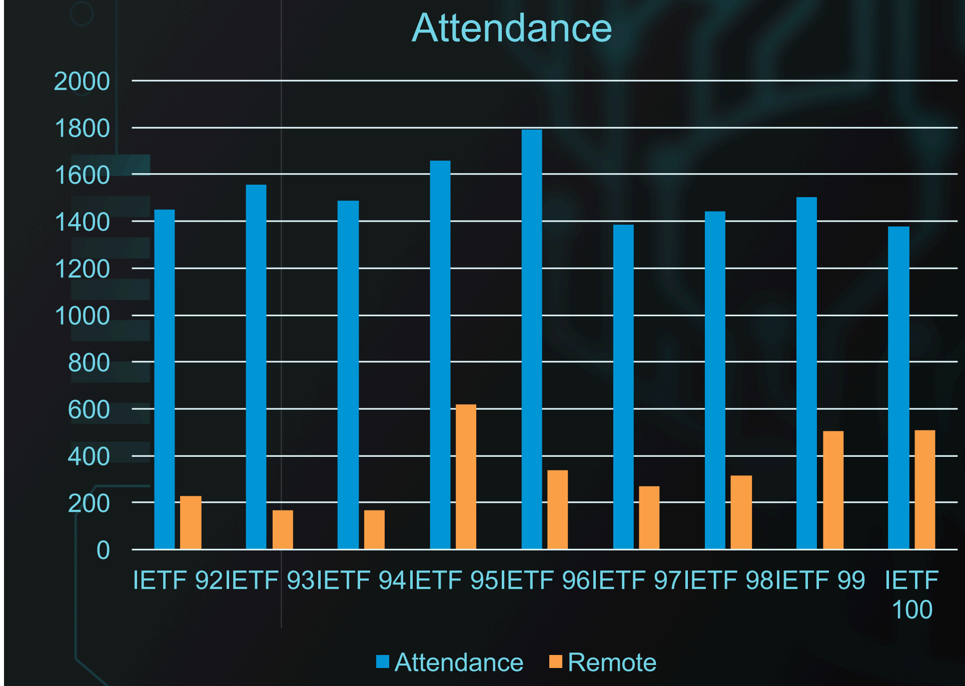
III. POST-IETF 91 IETF HACKATHON - EXPERIMENTS WITH RELATED OSS PROJECTS

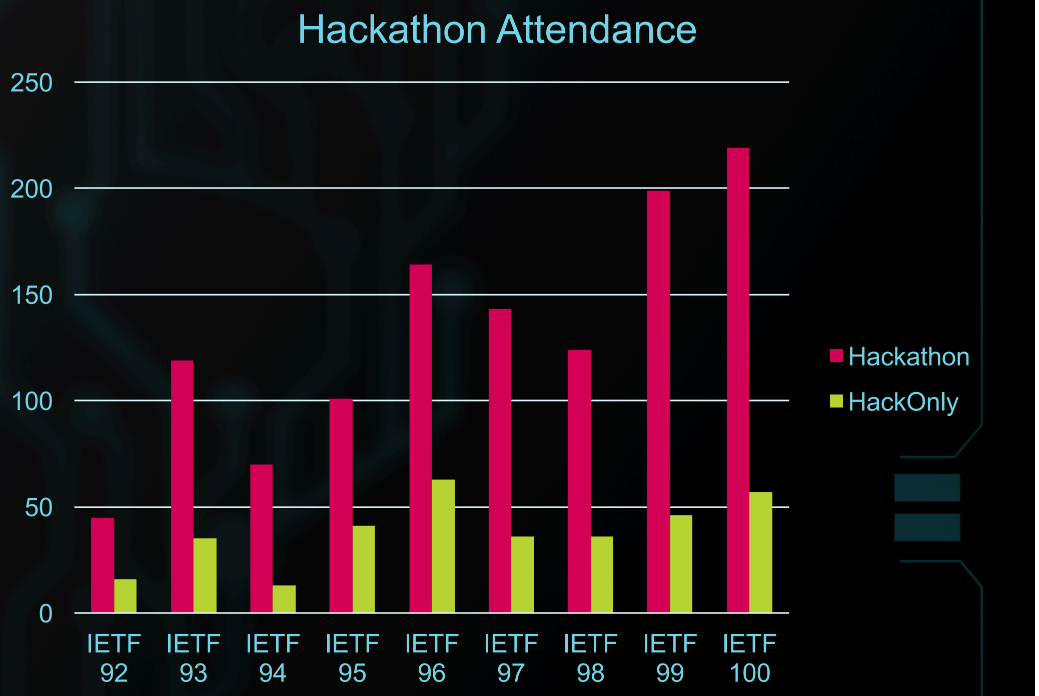
IETF HACKATHON

SUCCESSFUL EXPERIMENT

- Cisco DevNet brought to IETF 92, March 2015
- Funded and ran for 2015 (3 per year)
- Advance pace and relevance of IETF standards
- Leader of Hacks from Private funding
- Attract new/young people to IETF
- Open Source – though repo/community outside IETF gubnance
- Hackathon adopted as part of IETF schedule
- Semi - Rotating sponsorship for funding
- Running Code (RFC 6982)

IETF HACKATHON
HISTORICAL PARTICIPATION

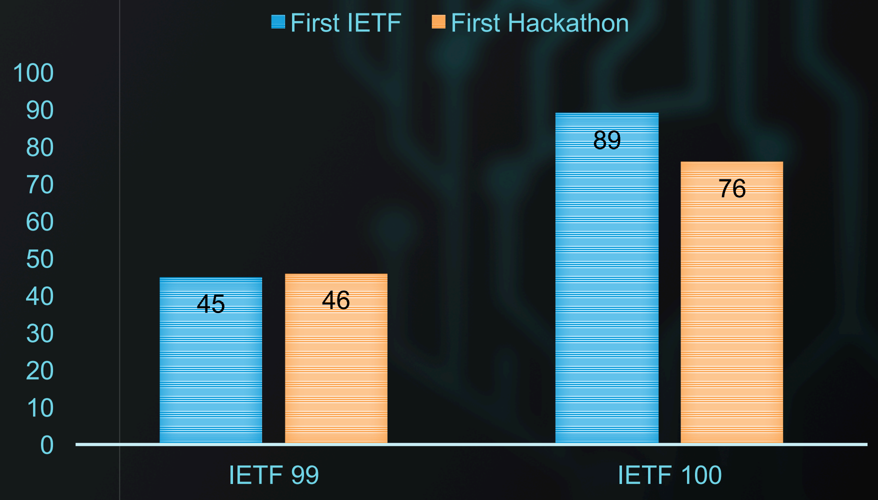




IETF HACKATHON

DRIVING ATTENDANCE AND FUNDING

DRIVING ATTENDANCE



- Survey not taken before IETF 99.
- Hackathon appears to be driving new IETF participation.
- A growing % of Hackathon participants are Hackathon-only.

FUNDING

- IETF 92-94: Cisco sponsorship, covered actual costs
- IETF 95-97: Huawei $40k/meeting
- IETF 98: Ericsson $10k, Mozilla $5k, Cisco DevNet t-shirts
- IETF 99: No sponsor, Cisco DevNet t-shirts HAD TO CLOSE REG EARLY BECAUSE OF SPACE & FOOD FUNDING
- IETF 100: Cisco $40k, Cisco DevNet t-shirts
- No sponsors identified yet for IETF 101 onward.

SPONSOR MODEL IS FAILING!

POPULAR HACKATHON COMMUNITIES

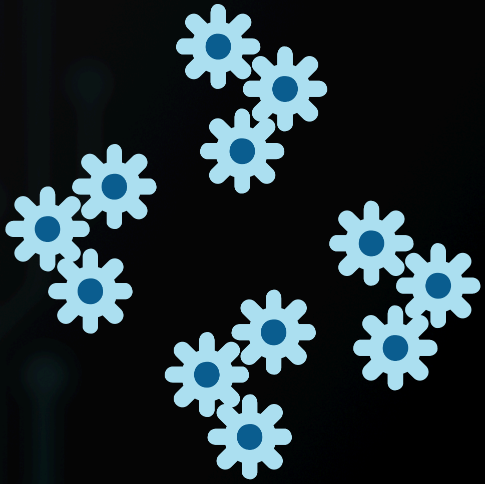
http://pnda.io Simple, scalable open source data platform provides a common set of services for developing network and service analytics applications.

http://fd.io Simple, scalable open source data plane toolkit enables early code development and rapid deployment. Avoid nominally specifying protocol with REAL implementation residing elsewhere (e.g. OpenFlow). Interop becomes (nearly) free (e.g. Segment Routing).

https://www.opendaylight.org Simple, scalable open source control and management plane toolkit (SDN platform) for developing networking applications (e.g. LISP). Used as middleware in projects like ONAP.
Model-driven modular software design.

AND A LONG LIST OF OTHERS

- Thor video codec https://github.com/cisco/thor
- Daala video codec: https://github.com/xiph/daala
- NAT Tools: https://github.com/NATTools
- DNS Utilities: https://github.com/getdnsapi
- OpenDNSSEC: https://www.opendnssec.org/
- Kea DHCP server: https://www.opendnssec.org/
- OpenWSN: https://openwsn.atlassian.net/wiki/spaces/OW/overview
- RIOT: https://riot-os.org/
- YDK YANG Development Kit: https://github.com/CiscoDevNet/ydk-py
- SCTP Lab: https://github.com/sctplabONOS: http://onosproject.org/
- OPNFV: https://www.opnfv.org/Sysrepo: https://github.com/sysrepo
- COSE working group implementations: https://github.com/cose-wgNsh-sf-devkit
- NSH Service Function Dev Kit: https://github.com/dcdolson/nsh-sf-devkit
- Joy: https://github.com/cisco/joy
- WebRTC E911 PSAP: https://github.com/IETF-Hackathon/webrtc-e911-psap
- The Trusted Domain Project: https://github.com/trusteddomainprojectI2
- NSF Framework: https://github.com/kimjinyong/i2nsf-framework
- Let’s Encrypt: https://github.com/letsencrypt
- NEAT https://github.com/neat-project
- DDOS Open Threat Signaling: https://github.com/nttdots/go-dots
- IPv6 Multiple Provisioning Domains: https://github.com/IPv6-mPvD
- Multipath TCP: https://github.com/multipath-tcp
- RPKI RTC Client C Library: http://rtrlib.realmv6.org/
- Magen https://github.com/magengit



THAT’S A LOT OF STUFF!

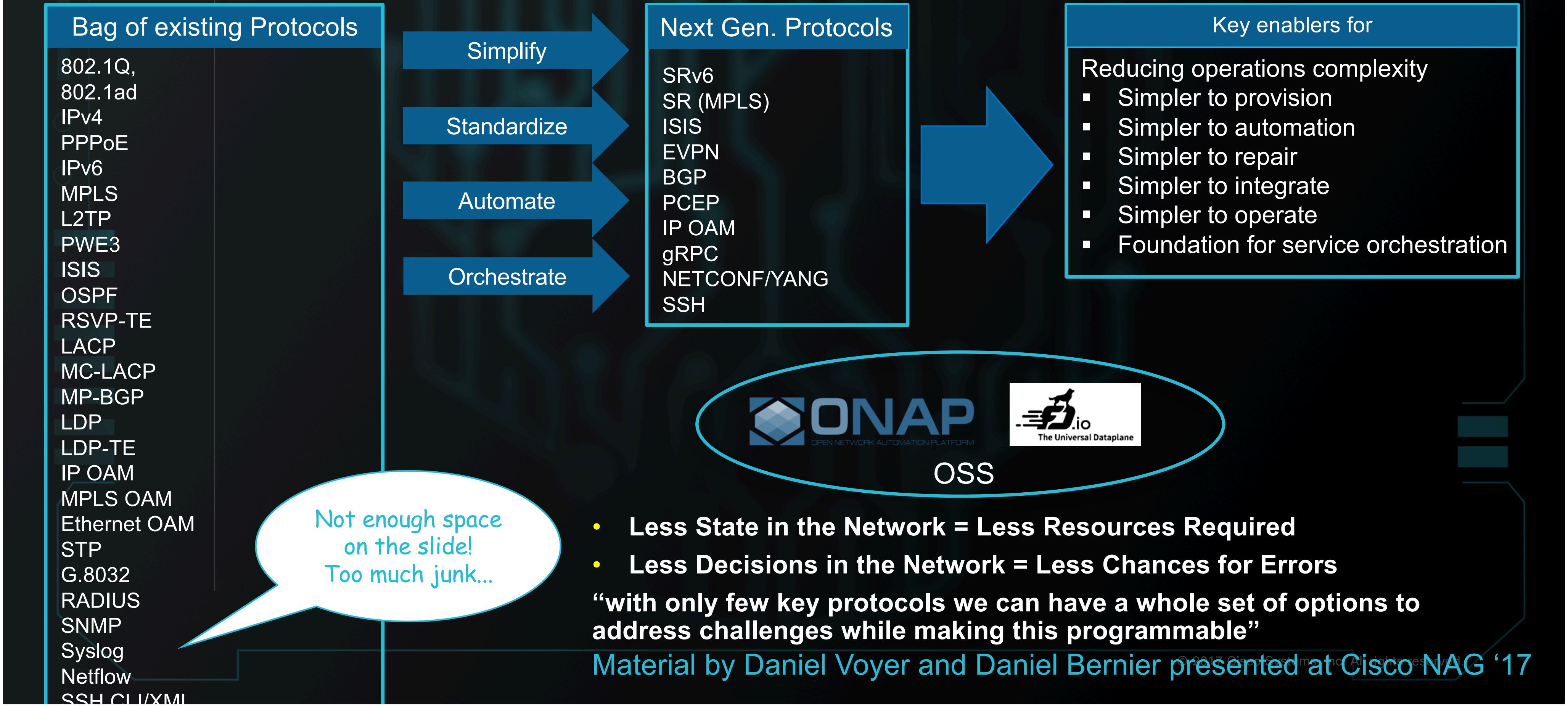
WHAT TRAJECTORY ARE IETF AND OSS CUSTOMERS/OPERATORS ON?

SIMPLIFICATION & EXPERIMENTATION

- Simplification via rejection of unnecessary technology (a
  - Fewer protocols to deliver services
  - Less state to manage
  - More control
  - More automation
- Move to open source “to be able to learn how it works”

ARCHITECTURE CHANGE

DRASTIC NETWORK PROTOCOLS REDUCTION @ BELL CA



Bag of existing Protocols

802.1Q,
802.1ad
IPv4
PPPoE
IPv6
MPLS
L2TP
PWE3
ISIS
OSPF
RSVP-TE
LACP
MC-LACP
MP-BGP
LDP
LDP-TE
IP OAM
MPLS OAM
Ethernet OAM
STP
G.8032
RADIUS
SNMP
Syslog
Netflow
SSH CLI/XML

Next Gen. Protocols

SRv6
SR (MPLS)
ISIS
EVPN
BGP
PCEP
IP OAM
gRPC
NETCONF/YANG
SSH

Key enablers for

Reducing operations complexity

- Simpler to provision
- Simpler to automation
- Simpler to repair
- Simpler to integrate
- Simpler to operate
- Foundation for service orchestration

ONAP
OPEN NETWORK AUTOMATION PLATFORM
.io
The Universal Dataplane
OSS
Not enough space on the slide! Too much junk...

- **Less State in the Network = Less Resources Required**
- **Less Decisions in the Network = Less Chances for Errors**

“with only few key protocols we can have a whole set of options to address challenges while making this programmable”

Material by Daniel Voyer and Daniel Bernier presented at Cisco NAG ‘17

PROVIDER PERCEPTIONS

SDO VS OSS

- OSS
  “We use Open source in projects because it’s simple and its fast. Its fast because you can download the code and get to work straight away. Most open source is relatively simple because it was usually designed to solve a specific problem, but if the software is complex, we have source so we can extract the parts we need to create a simple solution” -Adam Dunstan (CenturyLink)

- SDO
  “People tent and expect to solve everything within a protocol which makes the protocols overly complex and “slow” to develop.” -Daniel Voyer (Bell Canada)

POST91 INDUSTRY COMPARISON:
OTHER SDO'S MAKING TRAJECTORY CHANGE

MEF

EXPANSION DUE TO MEMBER REQUEST

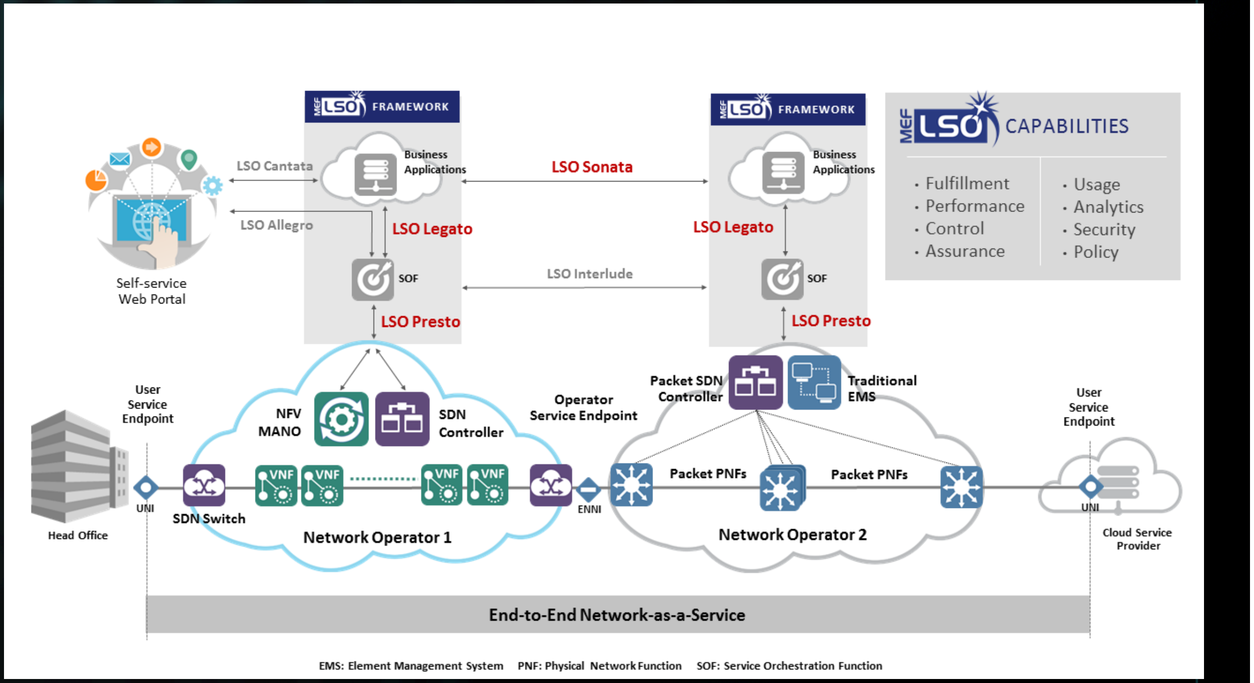
- Global Deployment of Carrier Ethernet Networks Services
- Found in 2001
- 210 + member companies
- Certification Programs
- Multi carrier interworking is key

*The MEF is the driving force accelerating the industry transition to agile, assured, and orchestrated services … that o*focus on what license rights are expected to properly engage with open source community projects*ffer user-directed control over service capabilities and cloud connectivity.*

MEF CHALLENGES OVER LAST 7 YEARS

RE-INVENTION

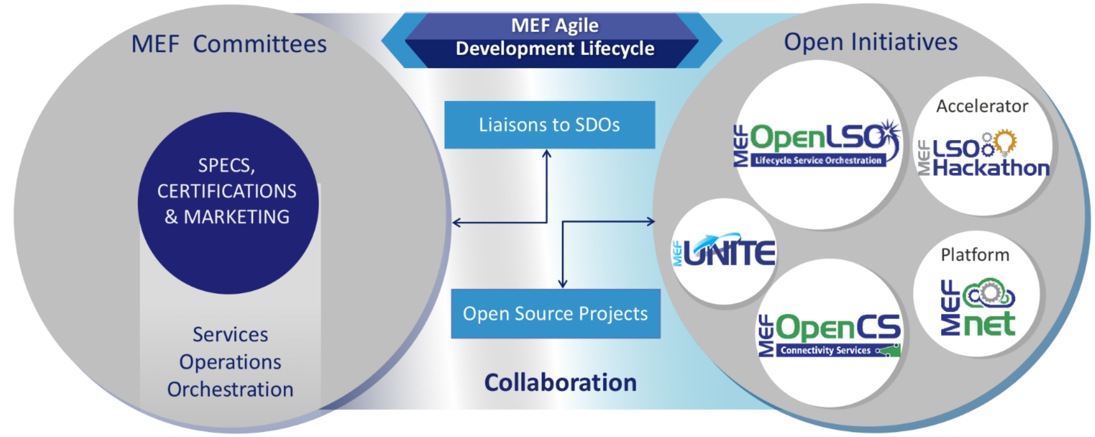
- Victim of own success
- Carrier Ethernet Network Services deployed globally
- Now what?
- Move up the stack to L3-L7
- Lifecycle Service Orchestration (LSO) for Next-Gen Networks Services
- LSO architecture and APIs



EMS: Element Management System PNF: Physical Network Function SOF: Service Orchestration Function

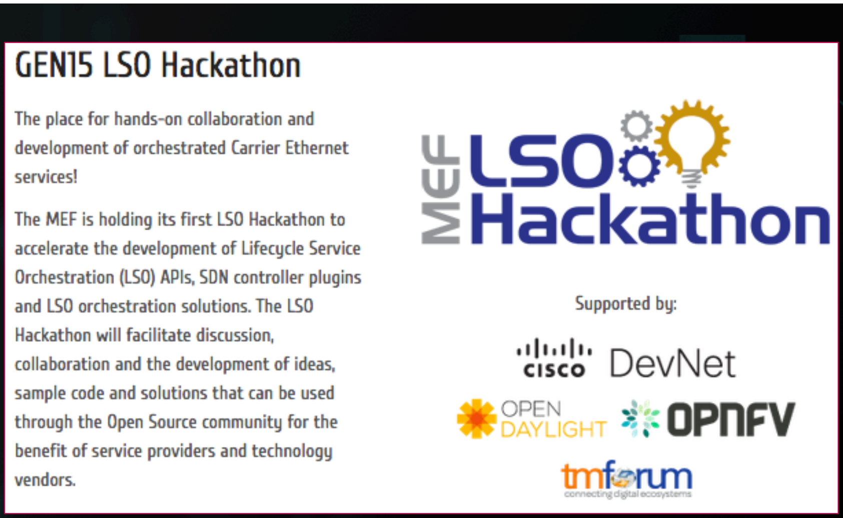
MEF OPEN INITIATIVES
LEADERSHIP DRIVEN

- Run by MEF Office of the CTO, Advisory Board, Members
- Includes OpenLSO and OpenCS projects, MEFnet, LSO Hackathons and the MEF UNITE program
- Mission: Benefit industry by creating reference implementations for standards defined components for Next Generation network services


MEF Committees
SPECS, CERTIFICATIONS & MARKETING
Services Operations Orchestration
MEF Agile Development Lifecycle
Liaisons to SDOs
Open Source Projects
Collaboration
Open Initiatives
MEF OpenLSO Lifecycle Service Orchestration
Accelerator
MEF LSO Hackathon
MEF UNITE
MEF OpenCS Connectivity Services
Platform
MEF net

LSO HACKATHON
OSS KEY TO RESTRUCTURE

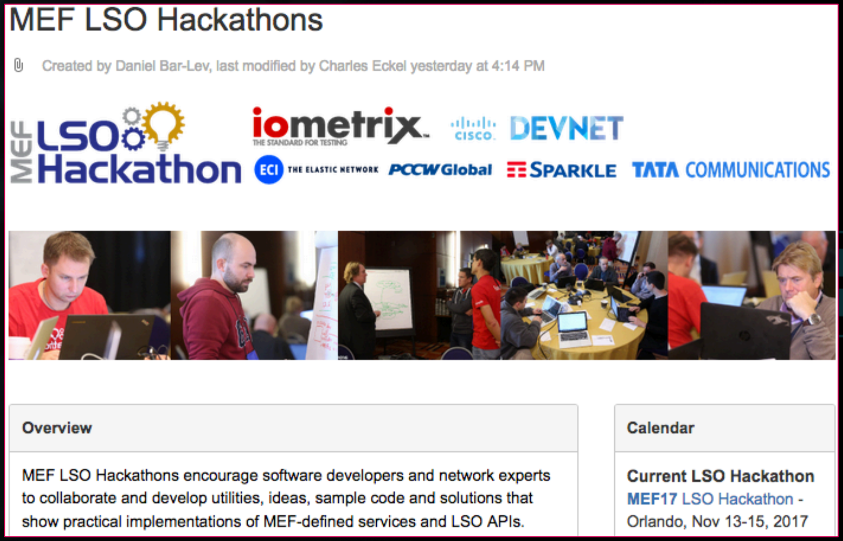
- Cisco DevNet introduced MEF to hackathon at GEN15, Nov 2015
- Funded by MEF, run by DevNet
- Transformed LSO architecture and APIs into running code
- MEF restructured with hackathon and open source as key components
- Privately funded



GEN15 LSO Hackathon

The place for hands-on collaboration and development of orchestrated Carrier Ethernet services!

The MEF is holding its first LSO Hackathon to accelerate the development of Lifecycle Service Orchestration (LSO) APIs, SDN controller plugins and LSO orchestration solutions. The LSO Hackathon will facilitate discussion, collaboration and the development of ideas, sample code and solutions that can be used through the Open Source community for the benefit of service providers and technology vendors.



MEF LSO Hackathons

Created by Daniel Bar-Lev, last modified by Charles Eckel yesterday at 4:14 PM

Overview

MEF LSO Hackathons encourage software developers and network experts to collaborate and develop utilities, ideas, sample code and solutions that show practical implementations of MEF-defined services and LSO APIs.

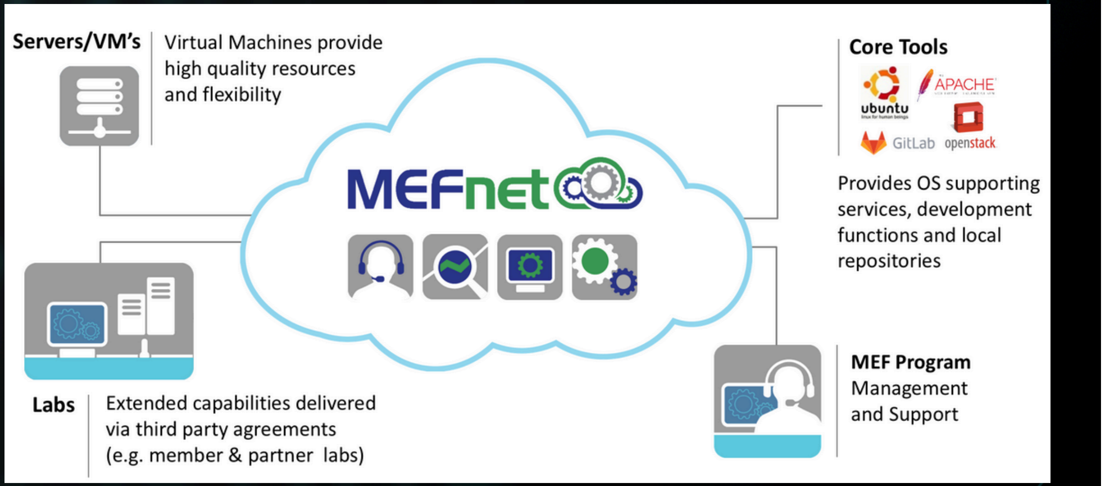
Calendar

Current LSO Hackathon
MEF17 LSO Hackathon - Orlando, Nov 13-15, 2017

MEFNET

INFRASTRUCTURE INVESTMENT

- Storage and compute platform
- Hosts reference implementations based on open and commercial software
  - OpenLSO projects, OpenCS projects
  - LSO Hackathons
  - MEF Software Developer Community
- OpenStack deployment



Servers/VM's Virtual Machines provide high quality resources and flexibility

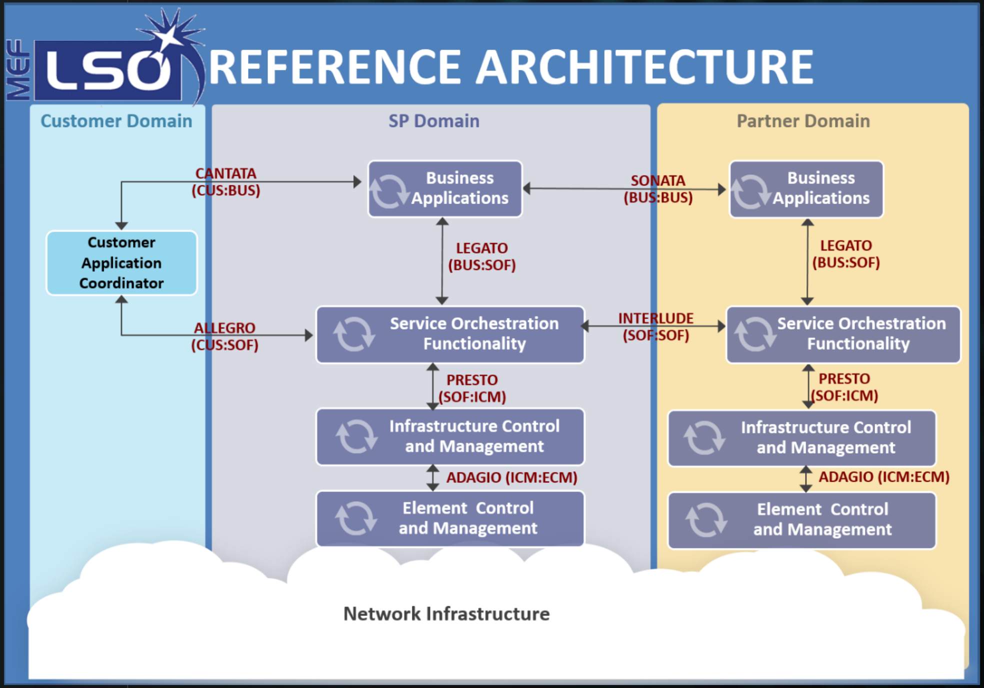
Core Tools Provides OS supporting services, development functions and local repositories

Labs Extended capabilities delivered via third party agreements (e.g. member & partner labs)

MEF Program Management and Support

WHAT MAKES MEF DIFFERENT?

BOUNDED ARCHITECTURE, BOUNDED TARGET



Pascal Menezes MEF 55

- MEF 3.0 Global Services Framework
  Service Descriptions
- Multi-domain API framework(s)
- Intra-Operator Open APIs Model
  ONF partner (TAPI)
- Inter-Operator API
  ONAP, TMForum partner
- Customer/Business Interface
  Sonata (MEF) SDK
- Multi-level interconnection
  Biz level
  Orchestration level

CROSSOVER OSS INTEREST

MOST POPULAR HACK TOOLS IN MEF & IETF HACKS

TOP 3 TOOLS	OVERLAPPING USE
IETF MEF	OpenDaylight https://www.opendaylight.org/
IETF	ONOS: http://onosproject.org/
	OPNFV: https://www.opnfv.org/
	YDK YANG Development Kit: https://github.com/CiscoDevNet/ydk-py
IETF	Sysrepo: https://github.com/sysrepo
MEF	PNDA: http://pnda.io/
	SNAS: http://www.snas.io/
	I2NSF Framework: https://github.com/kimjinyong/i2nsf-framework
MEF	ONAP: https://www.onap.org/

POST91: FRACTURING OF THE INDUSTRY, FIGHT FOR SDO RELEVANCY, DEFEND “TERRITORY”

Other Examples of attempted “restructure” or “resetting trajectory”

THERE ARE A LOT OF SDOS

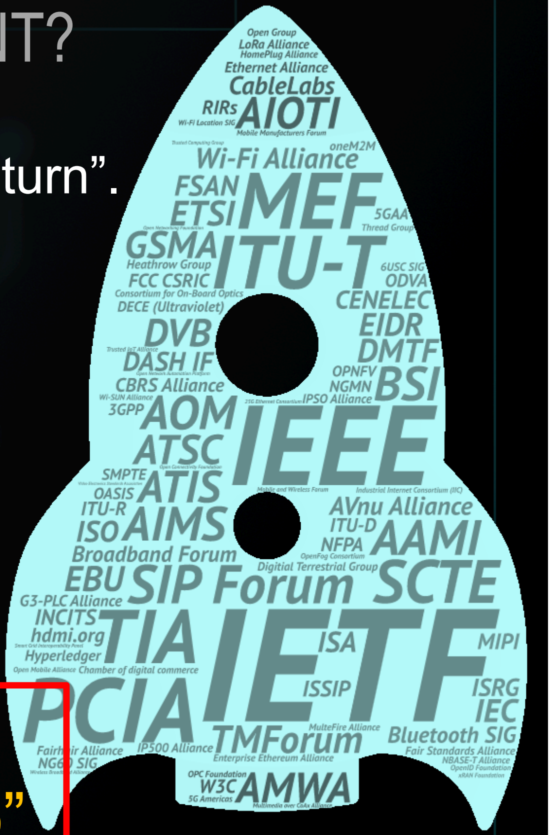
HOW LONG WILL THEY BE INDIVIDUALLY RELEVANT?

Lots of SDOs. Minimal success in making the SDO + OSS ”turn”.

- Where do we make the most impact?
- What will be relevant going forward?
- Are gaps forming we should be addressing?

What works for the IETF may not work for ALL.
The answer to “have we done enough?” is ***NO***

Claiming standardization responsibility for technology that has already been “defacto” standardized by an OSS community.



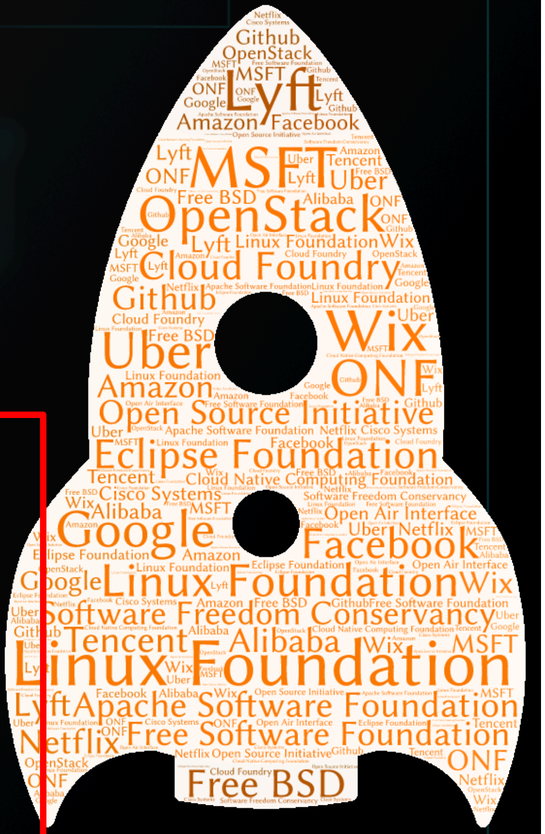
OPEN SOURCE PARTNERSHIPS

CHOOSE WISELY

There are a LOT of OSS “sources” and partners:

- Foundations - at least 8 major foundations relevant to our work
- Loose Projects - lots of unaffiliated projects under Github
- Massive Organizations - use open source as a market moving force

- OSS as strategic market and tech development tool by large organizations is unstoppable
- Interdependency requires SDOs to develop competencies, cultures and communities and outreach of their own
- Liaison Mechanism Failure



SDO STRUGGLES: OTHER ATTEMPTS

BUILDING COMMUNITY + CODE

🟢 Positive

🟡 Neutral

🔴 Negative

CableLabs

- Has code and sponsored projects:
  - C3 - PNM,
  - TruView,
  - OpenRPD
  - LoRaServer.io
- Not traditional OSS.
  - Community is small/specific
  - CL IPR wrapper “a little different” than common OSS

ETSI World Class Standards

- Difficult architecture specification (API?) hinders OPNFV (System Install/Test)
- Multiple project merge hinders ONAP (Automation Platform)
- LF pulling together projects and building community.

ITU

- SG12 QoS/QoE effort with PNDA just launched
- Alliance w/ NGMN still not attracting code or community (OpenAirInterface?)

ONF OPEN NETWORKING FOUNDATION

- Standardized OF - didn’t control data plane reference implementation (OVS).
- CORD - ONOS control plane codebase. Open Source License
- Tightly controlled (closed) community for controller core

tmforum

- ”Catalyst” 5G PoC in 2017 (2nd year) is collaboration of existing members.
- Two ”Open Hacks” in 2016. Mostly developing on vendor or forum provided APIs.
- Attempts like OpenOSS (APIs on SourceForge) are essentially APIs w/o software.

MODEL: CERTIFICATION AND TESTING

HISTORIC - APACHE/JCP FAIL

- Loosely coupled “certification and test” relationship between SDO and OSS.

- The reputation for JCP specs became so bad that the community would ignore them until Apache (open-source implementations of JCP specifications) had fixed the problems.

- Apache leaves Java SE/EE Exec Committee in dispute with Sun over JCP “open specification process” in Dec 2010.

THE APACHE® SOFTWARE FOUNDATION

Java Community Process

Bad Outcome #1: “Blind” specification can lead to SDO irrelevance. OSS project becomes authority.

MODEL: CERTIFICATION AND TESTING

CURRENT - ETSI/OPNFV FAIL

- Arguably, even looser “certification and test” relationship.
- OPNFV has a certification program in the works for years now. They are ready to launch and are currently in beta. https://cvp.opnfv.org/#/

  For the initial release, all they test is Openstack APIs (“Refstack ++”??).
- OPNFV does NOT certify against ETSI NFV framework per se (weak untestable specification - see JCP). OPNFV have chosen their own set of tests.

  OPNFV DOES implement a set of ETSI test cases, i.e. the “Yardstick” test tool implements parts of ETSI TST 001, but this is the minority of work - there are many more tests that go beyond ETSI.

World Class Standards

Bad Outcome #2: “Blind” specification can lead inability to certify or test (more irrelevance).

MODEL: CERTIFICATION AND TESTING

CURRENT - 3GPP/GCF SUCCESS?

- April 2016 GCF signs partnership project agreement with 3GPP

“GCF Certification includes conformance, interoperability and field testing for the following 3GPP radio access technologies (RATs), and their extensions, in the frequency bands.”

- Stated Benefits:

  Reduce manufacturer testing costs
  Shortens time-to-market for new handsets and devices
  Improves product quality
  Raises the overall quality of device interoperability

What factors make the loosely coupled “certification and test” relationship work? Tighter specifications, more upfront vendor involvement and tooling/infrastructure?

MODEL: REFERENCE DESIGN DRIVEN

OPEN SOURCE HARDWARE

TELECOM INFRA
PROJECT

RPHY
OpenAirInterface
nFAPI
OpenROADM

CONSOLIDATED

SCATTERED

- New approach moves beyond loosely coupled certification and test.
  Success coupled to reference designs
  Jumpstart ecosystems, promote interoperability
  Move at pace of SW, defining new system architectures and solutions
- TIP consolidates a number of projects under its umbrella, but many other similar project instances exist - with scattered ties to multiple SDOs.

Good ideas, lots of potential, Operator Driven

Need track the dependencies and references and use cases

OPEN CONNECTIVITY FOUNDATION

SINGLE FOCUS (M2M) USING ALL TECHNIQUES

OPEN CONNECTIVITY FOUNDATION™

1. Standard Partnerships
   - W3C, Genivi, Zigbee, OMA, HDMI, IPSO, CEA…
2. Ref designs
3. Certification - OCF, UPnP, AllJoyn
4. OSS Community
5. Device models

   OneIoTA.org contains live data models in a web-based tool that automates the process of quickly adding new devices to a network, regardless of location. (RAML & JSON Schemas)
6. Membership funding

https://openconnectivity.org/

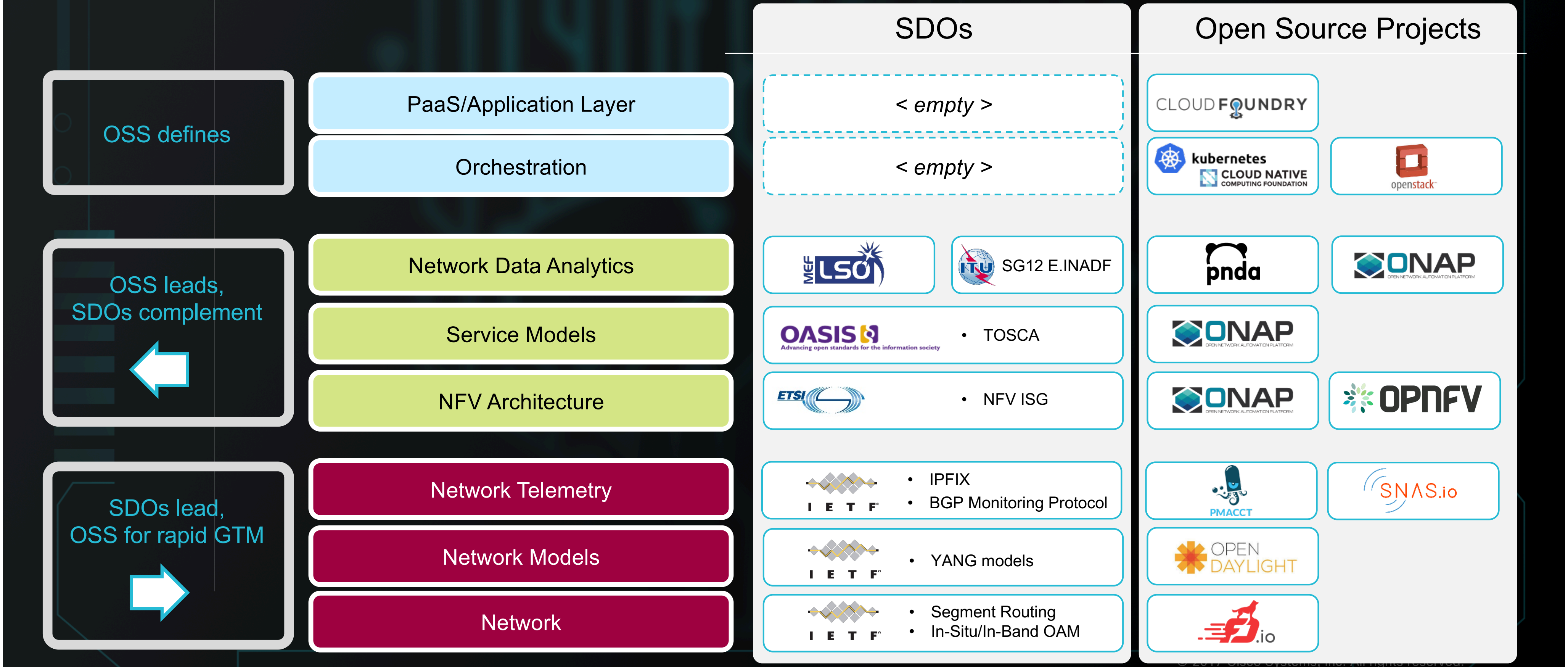
MILD TANGENT –
A FEW WORDS ABOUT PICKING AN OSS PARTNER

Because it’s OSS, it’s not magic and noone is riding unicorns on rainbows … it’s hard work by v diverse engineering communities with differing incentives and strategies

Slides of project logos, Garner Hype-cycles and Make-me-a-unicorn marketing machines are extremely deceiving

STANDARDS + OPENSOURCE

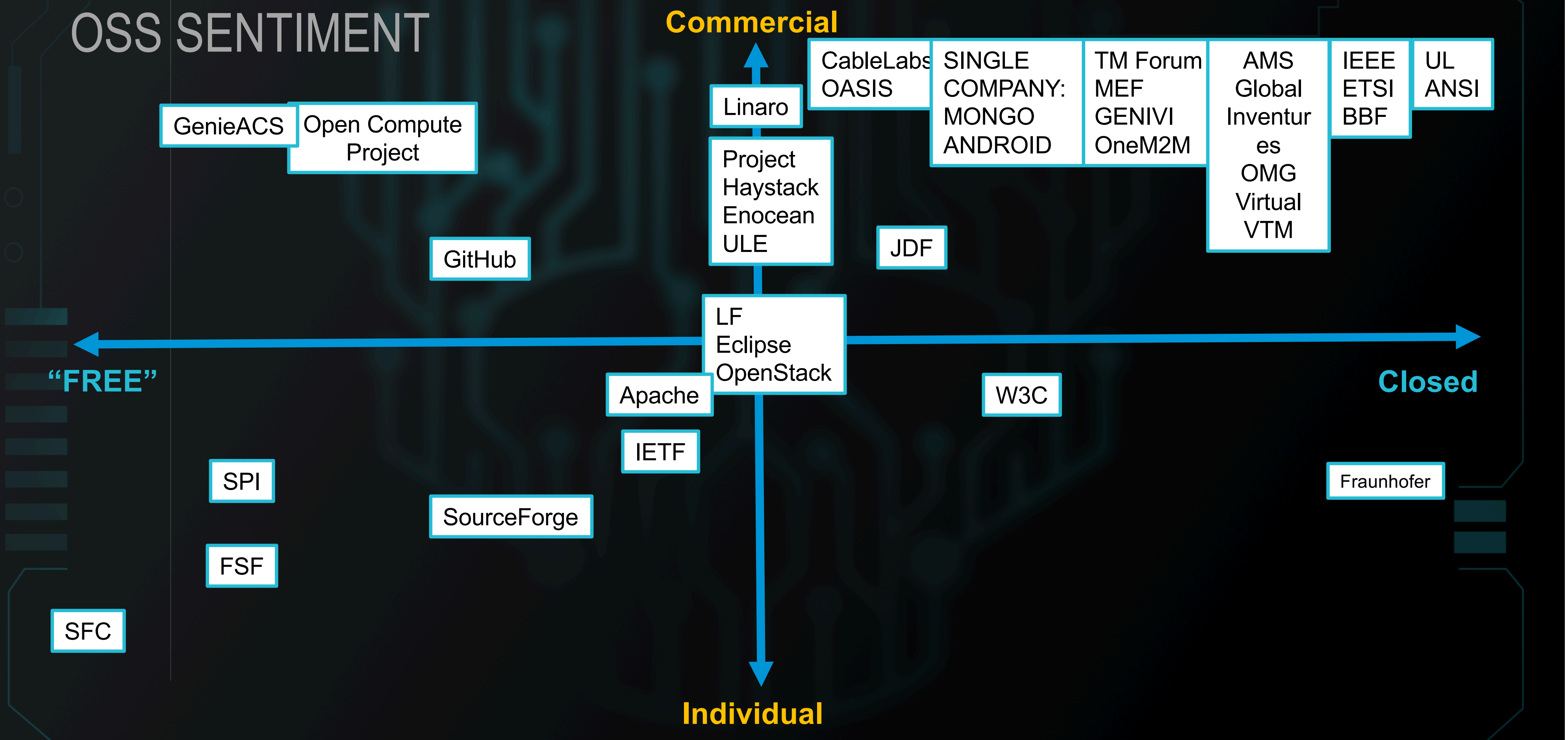
INDUSTRY BASELINES WITH LF



		SDOs	Open Source Projects
OSS defines	PaaS/Application Layer	*< empty >*	CLOUD FOUNDRY
	Orchestration	*< empty >*	kubernetes CLOUD NATIVE COMPUTING FOUNDATION openstack
OSS leads, SDOs complement ←	Network Data Analytics	MEF LSO ITU SG12 E.INADF	pnda ONAP OPEN NETWORK AUTOMATION PLATFORM
	Service Models	OASIS Advancing open standards for the information society • TOSCA	ONAP OPEN NETWORK AUTOMATION PLATFORM
	NFV Architecture	ETSI • NFV ISG	ONAP OPEN NETWORK AUTOMATION PLATFORM OPNFV
SDOs lead, OSS for rapid GTM →	Network Telemetry	IETF • IPFIX • BGP Monitoring Protocol	PMACCT SNAS.io
	Network Models	IETF • YANG models	OPEN DAYLIGHT
	Network	IETF • Segment Routing • In-Situ/In-Band OAM	FD.io

SDO/OSS PARTNERSHIP

OSS SENTIMENT



Commercial

Individual

“FREE”

Closed

CableLabs
OASIS

SINGLE
COMPANY:
MONGO
ANDROID

TM Forum
MEF
GENIVI
OneM2M

AMS
Global
Inventur
es
OMG
Virtual
VTM

IEEE
ETSI
BBF

UL
ANSI

Linaro

Project
Haystack
Enocean
ULE

JDF

GenieACS

Open Compute
Project

GitHub

LF
Eclipse
OpenStack

Apache

IETF

W3C

Fraunhofer

SPI

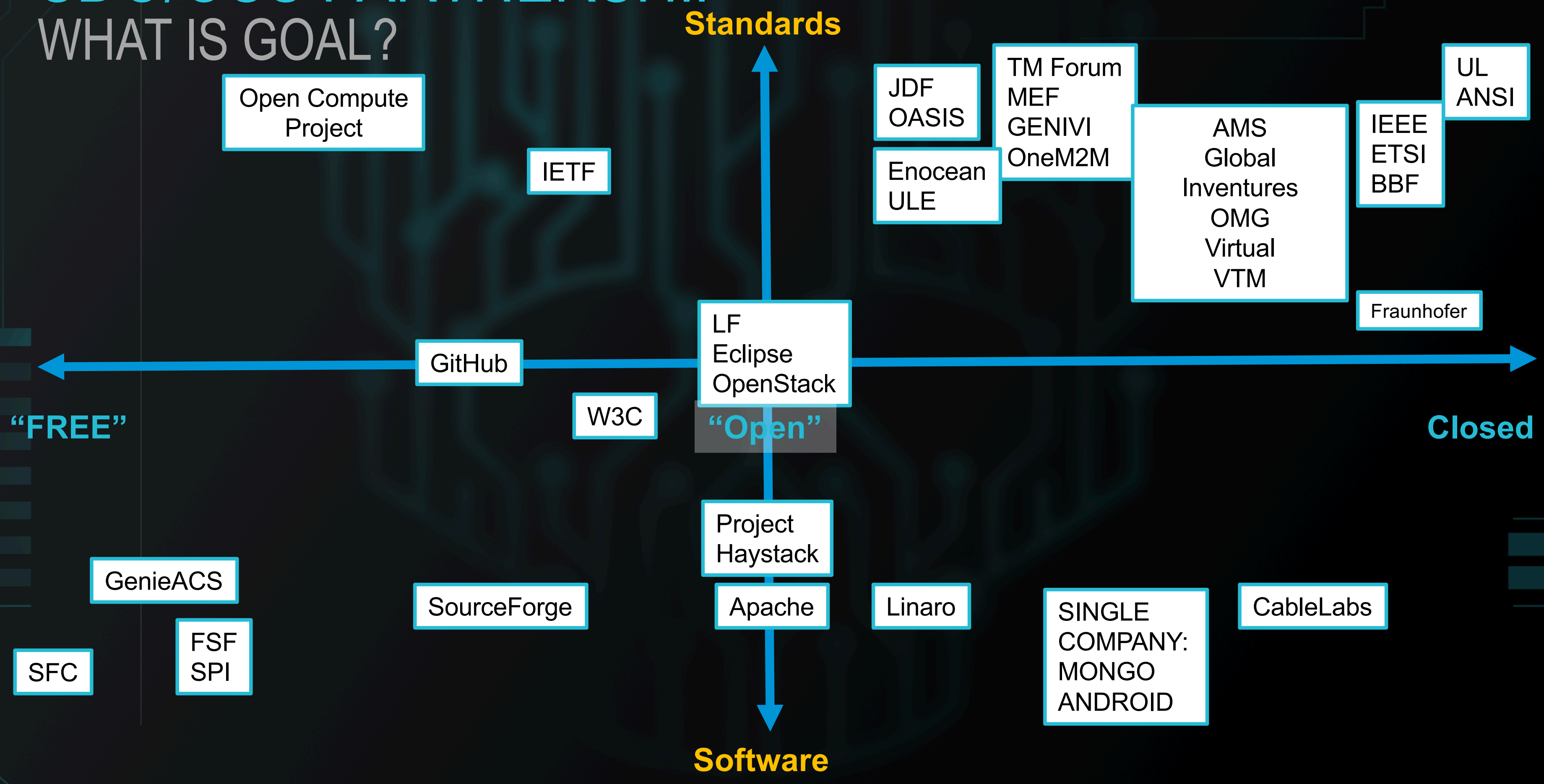
SourceForge

FSF

SFC

SDO/OSS PARTNERSHIP

WHAT IS GOAL?



Standards

Open Compute Project

IETF

JDF
OASIS

Enocean
ULE

TM Forum
MEF
GENIVI
OneM2M

AMS
Global
Inventures
OMG
Virtual
VTM

IEEE
ETSI
BBF

UL
ANSI

Fraunhofer

GitHub

LF
Eclipse
OpenStack

W3C

"FREE"

"Open"

Closed

Project
Haystack

GenieACS

SourceForge

Apache

Linaro

SINGLE
COMPANY:
MONGO
ANDROID

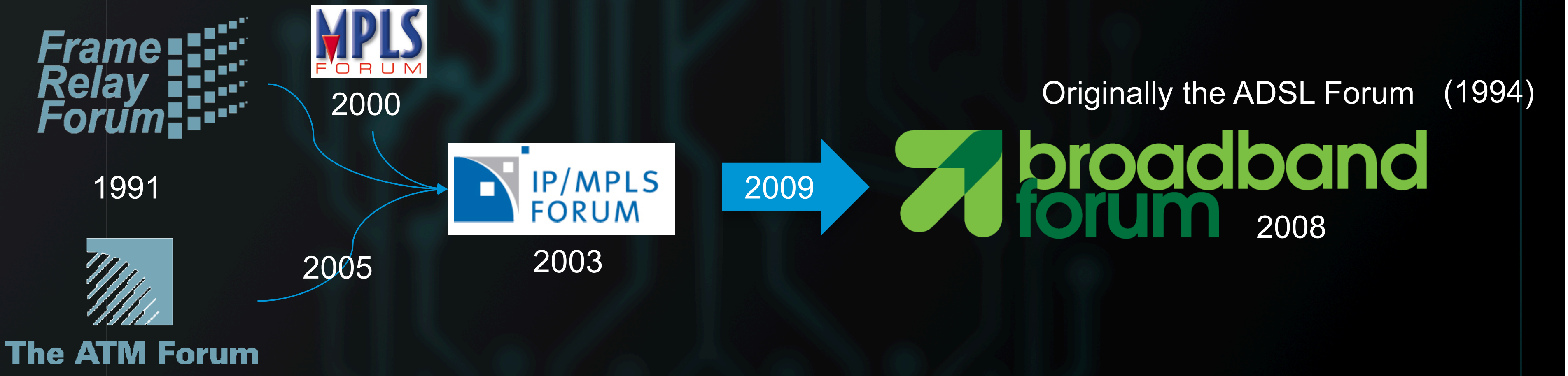
CableLabs

SFC

FSF
SPI

Software

POTENTIAL FOR AGGREGATION OF SDOS



1991

2000

2005

2003

2009

Originally the ADSL Forum (1994)

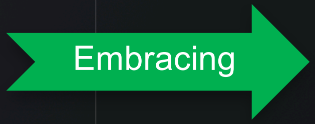
2008

- If not irrelevant, SDOs may become niched and probably aggregate together.
- Small communities of folks that come together in a bigger meeting because their own can’t be afforded.
- Example: Frame Relay Forum and MPLS Forum (founded 1991) collapse into MFA (2003), rebrand as IP/MPLS Forum to be joined in 1995 by ATM Forum. Ultimately they all join the BBF (rebranded ADSL Forum) in 2009.

POTENTIAL FOR SMALL FISH IN LARGE AQUARIUM

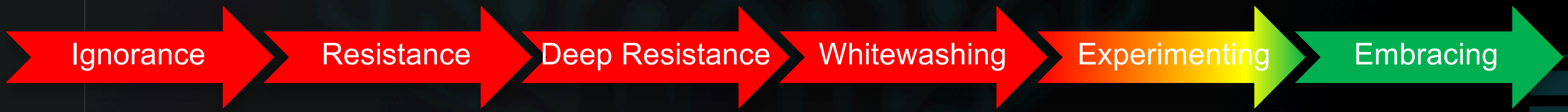
JOURNEY TO OSS BASED PRODUCTS & SERVICES

- Some companies had a short journey: Google, Netflix, Lyft



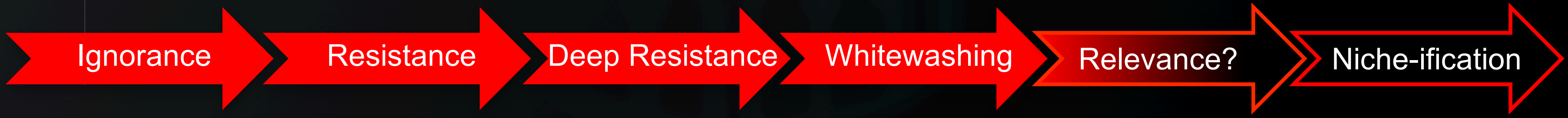
Embracing

- Some were successful users of OSS in transforming business: MSFT



Ignorance → Resistance → Deep Resistance → Whitewashing → Experimenting → Embracing

- Most traditional (paper) SDOs



Ignorance → Resistance → Deep Resistance → Whitewashing → Relevance? → Niche-ification

SDO + OSS

PARTNERSHIP NARRATIVES

- Potential SUCCESS - Friendly & Aligned IP policies (W3C, IETF)
- Potential failure - Can’t break from confidential development of standards and traditional IP licensing (RAND) (ETSI, IEEE, ITU)
- Potential failure - Can’t solve financial model (e.g. IEEE charges $40k to setup a new project, nothing after)

  No marketing, events, DevOps, etc.
- Real threats - New IPR model (e.g. ISC, BSD, MIT) VS RAND Undermines open source distribution, challenges open participation

  RAND is conceding software patents

OSS isn’t in the SDO DNA. Hire or use experts to guide and manage community/developer outreach

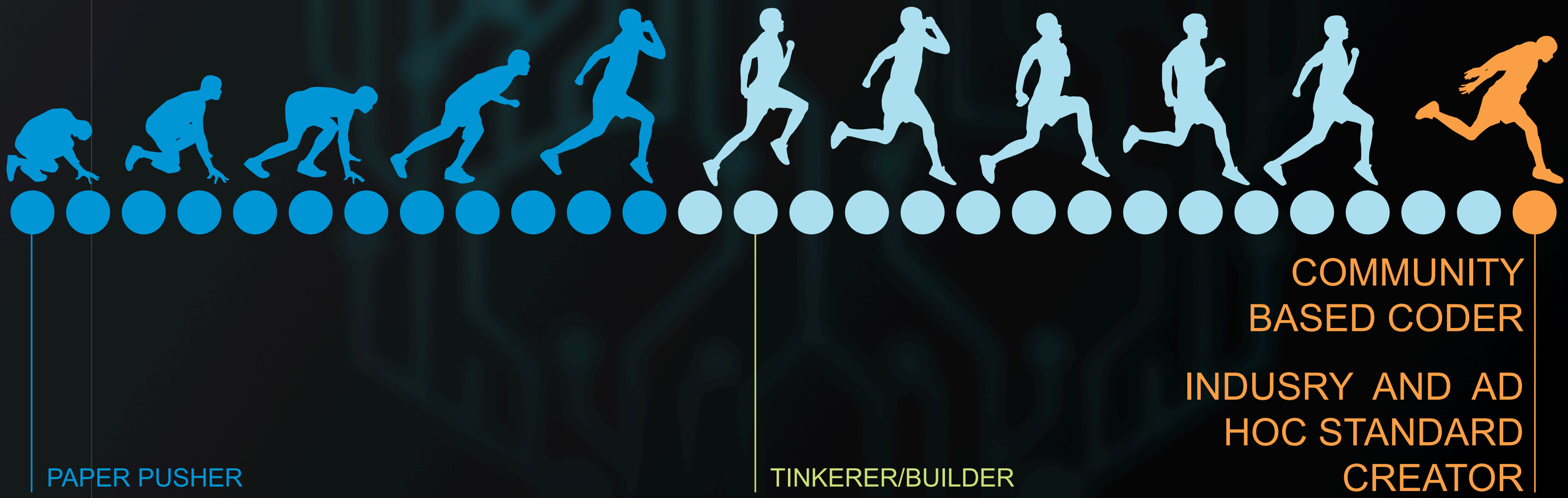
MANAGING IPR

- For most OSS projects only IPR terms are the OSS license
  - Pros: Lightweight, fast, low barrier to contribution
  - Cons: Rights conditioned on license compliance, license can’t be changed, multiple owners, no check on 3rd party IPR issues
  - Trend away from strong “copyleft” like GPL toward more permissive licenses like Apache 2.0 and BSD/MIT in recent years
- Major projects usually govern IP with additional terms like a Contributor License Agreement (CLA) or formal IP Policy
  - Pros: More flexibility, control, assurance (assignment, relicensing, provenance warranties, rights not terminable, patent policies)
  - Cons: Creates more overhead around contributing
  - But, orgs like Linux Foundation beginning to offer template CLAs, IP Policies, and other governing docs to help scale

IPR PITFALLS

- OSS community expects “open source” means sw will be usable by anyone for free without additional license
- Standards world beginning to see value in OSS methods
- RAND licensing conflicts with open source principles
- Some pushing for copyright-only “open source” projects that will require RAND patent royalty to use
- Highly controversial with OSS community and most SDOs, redefines open source and induces users to incur liability
- Recent problems surrounding Facebook’s React.js patent rider highlight why alignment is key

IV. MOVING FORWARD



PAPER PUSHER

TINKERER/BUILDER

COMMUNITY
BASED CODER

INDUSRY AND AD
HOC STANDARD
CREATOR

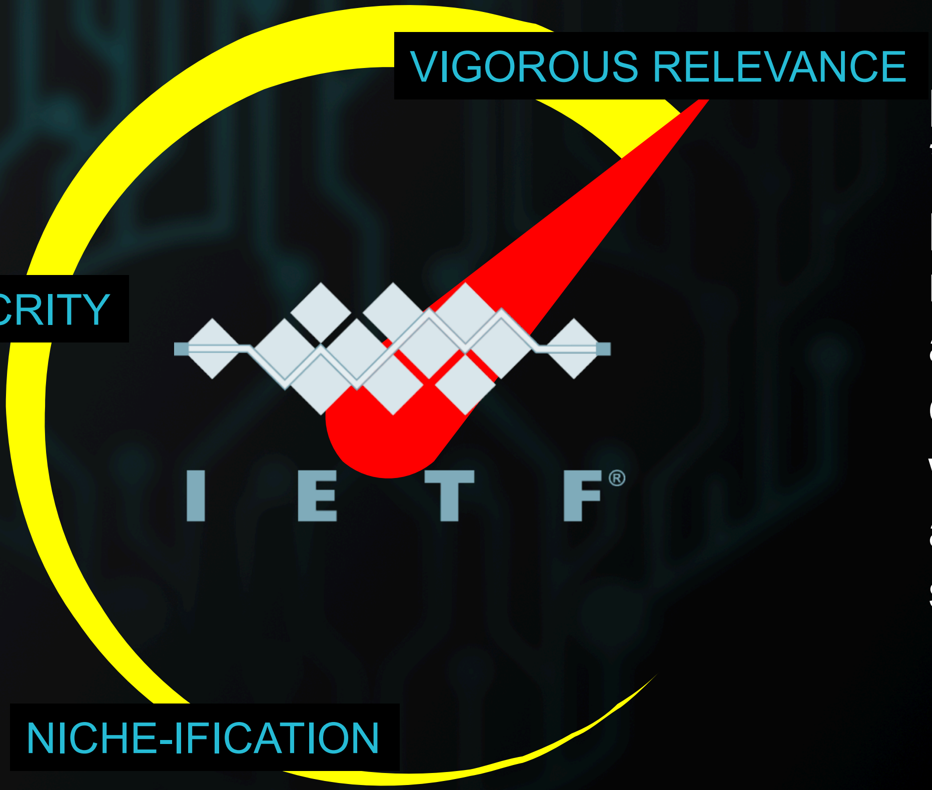
INDUSTRY
CATALYST

MULTI-COMMUNITY
INFLUENCER

SDO + OSS

IETF CHOICES

Actively reach out to engage with OSS or wait for them to find IETF and help them figure out their problems on their terms



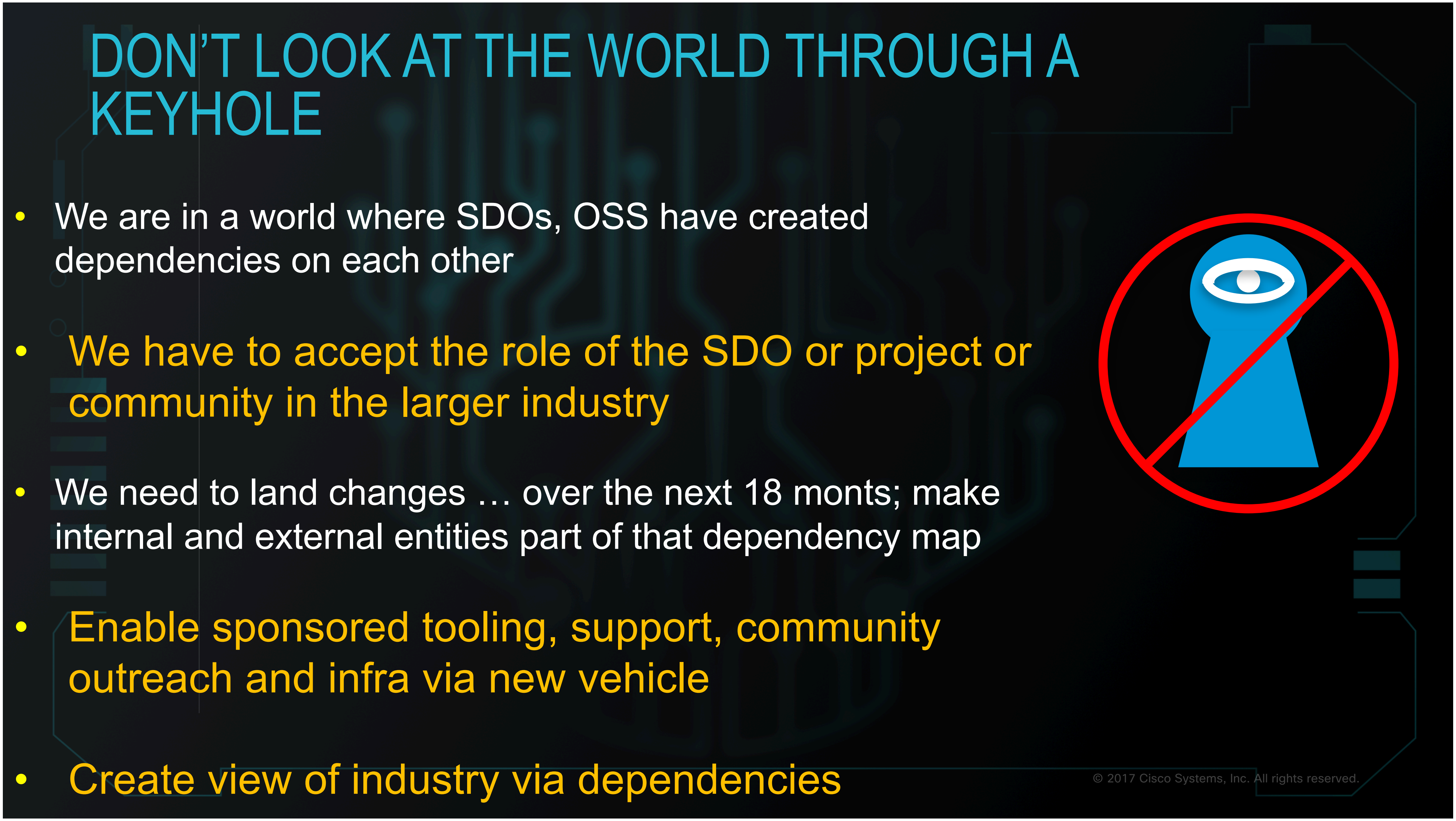
VIGOROUS RELEVANCE

MEDIOCRITY

NICHE-IFICATION

Identify emerging problems and *start* open source projects and other related SDOs with applicable communities to write code to solve and standardize solutions

Do nothing



DON'T LOOK AT THE WORLD THROUGH A KEYHOLE

- We are in a world where SDOs, OSS have created dependencies on each other
- We have to accept the role of the SDO or project or community in the larger industry
- We need to land changes … over the next 18 monts; make internal and external entities part of that dependency map
- Enable sponsored tooling, support, community outreach and infra via new vehicle
- Create view of industry via dependencies

OBSERVATIONS ON OSS PROJECT GENESIS

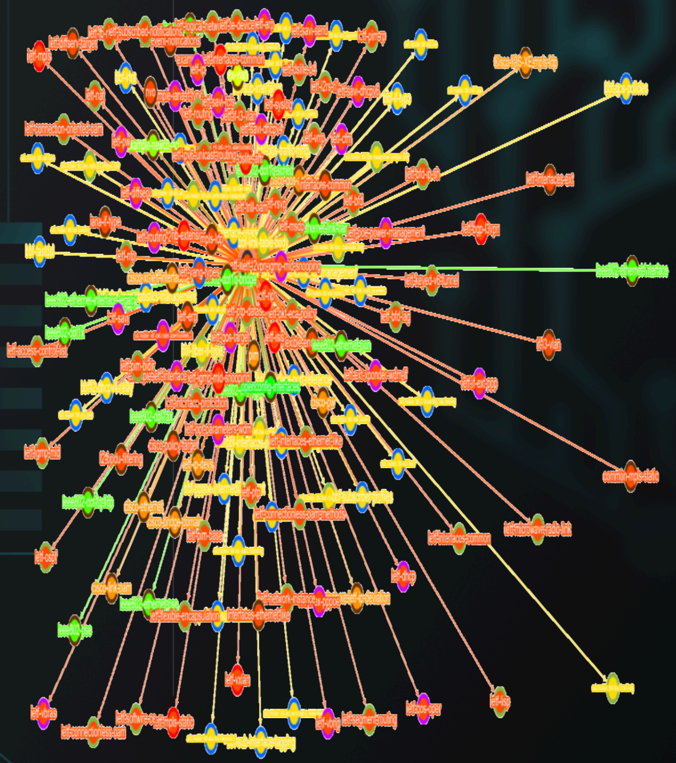
LESSONS APPLICABLE TO IETF?

- LF Harmonization is placed in an SDN/NFV context
  IETF put entire topic into IRTF and completely missed the boat, produced little
- OSS/SDO projects (ETSI/ONF) create partial/incomplete specifications, leave spotty implementations and closed groups
- IETF COULD seek out projects, (re)organize in an agile way, stop irrelevant projects

SPIFFE/SPIRE blow right past IETF consensus process. Standardizing and OSS'ing in the community

A PLATFORM OF PLATFORMS

PULLING IT ALL TOGETHER



- Yangcatalog and M2M are models for how we organize the fracturing of the industry, and stop the brownian motion between multiple SDOs and OSS

  Intuit how/if SDOs and OSS projects are working together … see the dependencies in a shared solution space.

- We need metadata to enrich those models:
  - We need a health metric of OSS projects. How can a consumer predict survival, assess efficacy?
  - We need a health metric of SDOs. We certainly need one for the quality and relevancy of specifications

IETF100 RECOMMENDATIONS

IASA2.0 IS THE STARTING LINE

- Publishing a RFC SHOULD not be the metric for IETF success
- A technology is successful when it's deployed
- If there is no way to automate a feature, it doesn't exist
- Develop tooling & metadata at the same time as specification
- Be faster or iterate
- Community and dependencies larger than email list
- Create your dependency map and reach out to your IETF "customers"
- Fund or partner or expand to build/maintain tools

Change the IETF process to be centered around the products of the RFCs NOT the ID#

ATCS

ALL THE COOL SH*T

David Ward | Chief Architect & CTO of Engineering
dward@cisco.com