How does an SDO enable innovation?

Open Standards, Open Source, Open Loop

David Ward
dward@cisco.com
2014.11.13
Open Standards
What is the role of SDOs?

• A question of relevance and NOT existence

• Software has come to dominate what we perceive as "the Internet" and the "agile" development model has created a sharp knee in the rate of innovation over the past couple of years … innovation that NEEDS standardization

• While it's important to have SDOs and consensus based standards, SDOs need to realize the OSS cycle time can create a market-based consensus to fill a standards void (and this realization may be the key to our collective futures)
“If open APIs become the de-facto definition of interoperability requirements, the role of the standardization bodies, and the opportunity for operators to influence specifications, diminishes. As a result the functional interoperability (and interchangeability) of vendors and devices will decrease, potentially leading to a more proprietary and less open and global nature of the Internet.”
SDO Challenges

• Every organization is potentially (and dangerously) self perpetuating and few SDOs have a life-cycle plan that bounds their authority and scope as applied to new technologies.

• New technology study groups are exploding across multiple SDOs.
  • Choose any new area of technical endeavor – cloud, SDN, NFV, IoT, “APIs” and you will find one or more ready examples
  • This dilutes and confuses the effort for contributors and consumers

• Coordination across SDOs doesn’t really appear to be working – and productivity within SDOs isn’t looking good either
SDO Response

- MEF
  - Known previously for specific, service-related (Carrier Ethernet) standard
  - Proposes to become all services creation/definition and main OSS arbiter via “Lifecycle Service Orchestration” (API/Interface/Endpoint definition)
  - Suggests better SDO collaboration (UNITE liaison process)
SDO Response

• ITU-T
  • Resolution 102 - rejects massive expansion of powers regarding public policy and Internet (and potential “UN takeover of the internet”*)
  • Suggested revisions in CWG-Internet workgroup to make it more open

CONS
“Land grab” doesn’t change the dis-function of both large and small SDOs (will not succeed).

PROS
Universal recognition that transparency and collaboration are the only way forward.
IETF Challenges – *Aristocracy vs. Meritocracy*

- Are we re-arranging the deck chairs on the titanic or creating a structure that can cope with emerging “agile” requirements?
- How much time are we spending on old vs new technologies?
  - Why has the IETF not had a successful engagement around SDN or NFV?
  - What’s the relationship between OPS and Orchestration projects?
  - How have the APPS and TRANSPORT areas evolved to keep up?
- How do we handle issues, architectures and technologies that potentially span or break our structure (YANG modeling explosion)?
- How do new entrants, new invested consumers (operators and IT) get an equal voice?
- Do we request other groups (OSS projects) to standardize in the IETF? Do we have a way to give feedback if/when projects headed down garden path?
Conway's Law: organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations

Is the way we set up distributed data modeling with “Drs” a revisit of MIBs? Repeat history?

The IETF structure has been around for a while.
Open Source
Open Source Architectures

What do these projects imply about what might be missing in SDO community work?

- Network programmability (control models)
- Management and operations (orchestration & NFV)
- Systems/Solutions

Service assurance: Fault, performance, resource management and service analytics … coming soon
Open Source Risks

• The danger to us all is the co-opting of open source and the lack of governance.

• Resulting problems:
  • Security flaws (critical technology with few eyes on it)
  • Small communities (underfunded monocultures)
  • Fragmentation (many OSS projects that each solve 20% of a problem but cannot be used together)
Open Source Collaboration Requires New Skills

- **Strategic Analysis**
  - Choosing best open source projects to meet goals
  - License review boards
  - What code to share and what to keep

- **Intellectual Property**
  - Open Source License Management
  - Contribution Agreements
  - Trademark Licensing

- **Development Process**
  - Social coding style: Git, email, IRC, etc
  - Project governance: contribution, maintainers, etc.

- **Business Process**
  - Downstream product integration
  - Upstream code contributions
  - Release management
SDO:OSS Impedance Mismatch

• It takes > 2 years for SDOs to draft paper standards
• It takes ~ 1 year for an OSS project to generate a real product that creates a de-facto standard.

\[ Z_{(SDO)} \neq Z_{(OSS)} \]
OSS-as-SDO Hybrid

- ONF attempts to be both “open” architecture and “standard” protocol.
  - Controller market development delays NBI standardization
    - OSS isn’t partially open
  - Architectural vision originally: limited “SDN is OpenFlow”
  - Weak collaboration (IETF, ODL)
- OpenDaylight – OSS project
  - Polyglot architectural vision – a superset of ONF
  - Marketing used to develop community
    - Developer-centric
Open Loop: For Further Discussion
IETF Takes Leadership Role

The IETF has leverage-able experience in:
- Protocol definition
- Architecture definition
- Modeling languages

The IETF has the RIGHT FOCUS:
- Not too broad (e.g. not Health & Safety, etc.)
- Not too narrow (e.g. single service domain)

APIs 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.
IETF Existing Structure Revisited

• 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
    • //’ize Specifications and any need for Framework/arch/
      requirements/use case drafts

• Less Talk, More Action
  • Fix/adapt the liaison process to a model that’s applicable to OSS Governance Groups

• Generate more code and ideas
  • Sponsor more research
  • Encourage more demo (functionality and interoperability e.g. SRv6) – WHERE is our “bits and bytes”? 
SDO Geeks != OSS Geeks

- Realize you can’t do everything and augment IETF capabilities
  - You don’t NEED to own everything (FAIL) – learn to point at collaborators.
  - You probably don’t WANT to own everything – it will fundamentally change your core community (maybe NOT for the better)
  - Enable the organization to shift focus beyond what’s hot or deposited on the doorstep (be strategic)

Shared Paradox
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! Hard to define standards if you aren’t following where the communities have formed. Content/output defined by those that show-up may not be enough
Form A Collaborative Loop

• 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
Embrace “Good” Open Source

• Align with properly governed OSS projects
  • Proven, neutral third-party management, proper licensing, support infrastructure, public participation
    • Linux Foundation
    • Apache Foundation
    • OpenStack Foundation
    • Eclipse Foundation
• Reward the “right type of open” projects
  • Projects that produce utility for the internet vs “dead code repository”
  • Look for - projects that compliment the IETF e.g. OpenDaylight (Linux Foundation) – driving YANG modeling into IETF and other OSS
Realizing these “Laws” exist, we must understand that the roles of OSS and SDOs need to change. Embrace it, move faster, and focus on a building a bigger and better Internet.

Law of OpenSource: quality of the project is 100% dependent on the interests, energy and capability of the developer community.

Law of OpenStandards: importance, validity and timeliness of standardization is 100% dependent on interests, energy and compromises of the individuals who have been empowered to manage, organize and complete the work effort of the SDO.

Adapt to the modern world.
TOMORROW starts here.