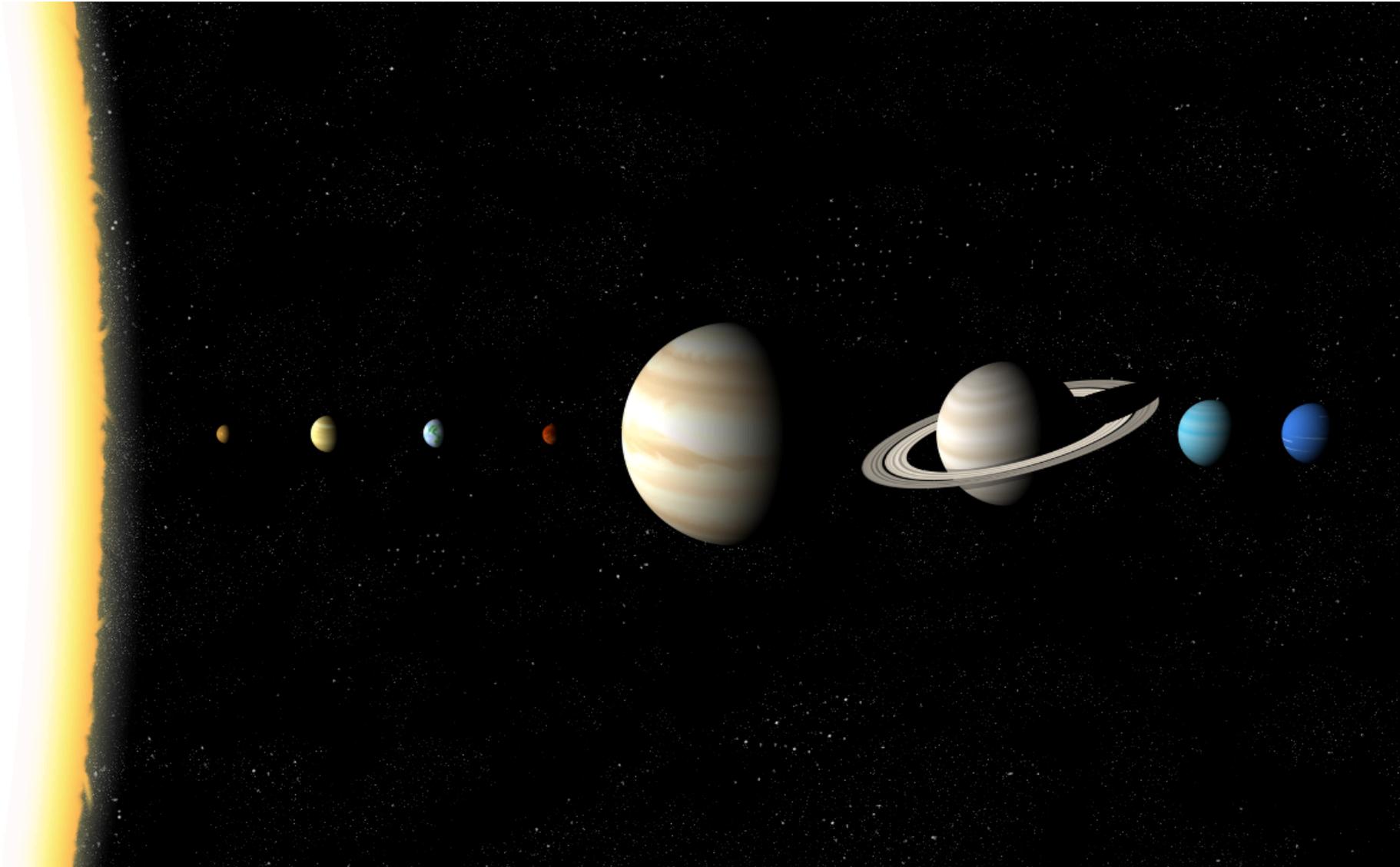


# Alignment of issues



# Half and Half important!



Eat your own Dog Food

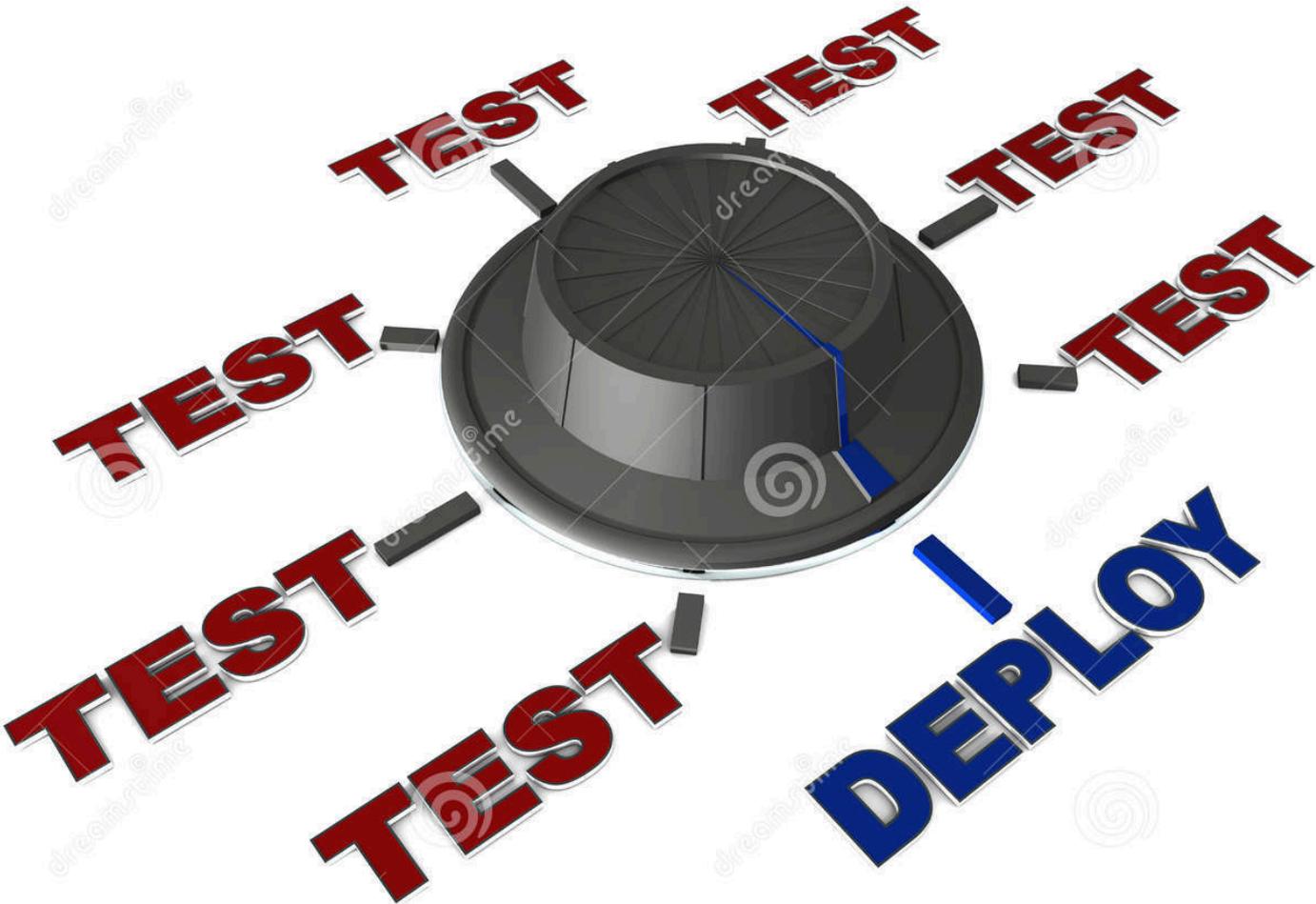


**IMPORTANT**



**Trouble Shooting**

# Automate Test – network IS SI



## SDN Futures:

What's missing today that we need to fix  
- Group 2

SDN Workshop LBNL 14-16 July, 2015

# Summary

- New Functionality
  - Abstractions
  - Crisp standards
  - Multi-domain
  - Multi-tenancy
- Manageability
  - Troubleshooting
  - Verification
  - Instrumentation
  - Common definitions
- Skills
  - Workforce development (devOps culture...)

# Abstractions

- Vertical abstractions
  - Inter-layer like App-Cntrl, Net-Net, Cntrl – Net
- Horizontal abstractions (can span multi-domain)
  - Service
  - Resource
  - Topology

# Standards are Squishy, e.g. OF

- How do we test interoperability between....
  - vendors, within SDN islands
  - SDN islands directly connected
  - SDN island and legacy network
- System level view – switch behavior
- Black box tests
- Best practices for hardware vendors
  - Based on use-cases, like TTPs but system view
- Behavior consistency tests
- Metrics for OF hardware

# Multi-domain SDN

- End-to-end services
  - Abstraction and description of services
- Model based
  - Resources
- Security and Trust model
  - Hard!
- Need to deploy!
  - Existing NSI model or a new way?

# Multi-tenancy

- Cater to multiple applications with likely conflicting priorities
  - Flow governance
    - From abstractions to implementation of prioritized flow entries.
    - Conflict resolution
    - Policy driven

# Troubleshooting

- Debug required at all levels: Service, Network, Device
  - Protocol coherence and correctness
  - End-to-end service coherence and correctness
- Fundamental tool development required for SDN
  - Best Practices (O)
  - Implementation (S, V)
  - Must work for multi-domain SDN

# Verification

- Sanity check, state consistency
- Verification of
  - Flow policy
  - Flow state
  - Abstraction → Instantiation
- Requires models to verify against

# Instrumentation

- Flow instrumentation covered in OF specification
- Missing
  - Instrumentation of the controller
    - What standard things need to be measured, and how?
  - Instrumentation of the NBI
    - What service characteristics needs to be measured, and how?
  - Instrumentation of the switch
    - In addition to flow statistics, what standard instrumentation is baseline for a SDN hardware device?

# Definitions

- We differ on common understanding of 'end-to-end', 'SDN', etc.
  - Ontology needed

# Workforce Development

# SDN Futures

What's missing today that  
we need to fix

SDN Workshop  
LBNL  
14-16 July, 2015

# “In the next 1 1/2 years, what do we want the SDN world to look like?”

- A clearer definition of the problems that SDN can solve for us must be assembled.
- From a large view:
  - Abstraction at all levels defined
  - Policy that is interwoven with abstraction assembled
  - Security across all layers addressed at least initially
- From the next level down:
  - Hardware based on more clearly defined standards
  - Programming, from the user down into the network, as both tools and methods specified and working
  - Service description and setup based on user feedback created
  - Troubleshooting, at all layers, including tools, practices, and environments implemented
  - Validation and performance testing, tools and techniques, defined
  - Monitoring of flows, tables, changes, errors, implemented end to end

# More details

- Abstraction at all levels defined
  - Device - network devices, end user participation, interoperability - multi-protocol...
  - End-to-End vs core vs single domain...
  - Across domains - what is multi-domain at different layers - policy controller...
  - Delegated Trust - how do authentication and access control work across these abstractions through a common model...
- Policy that is interwoven with abstraction assembled
  - Peering in SDX - peering of policy and data planes...
  - Policy around resource allocation for multi-domain uses...
  - Service level policy and implementation...
- Security across all layers addressed at least initially
  - Security of SDN infrastructure
  - Security through SDN infrastructure

- Hardware based on more clearly defined standards
  - Open systems SDN vs. SDN
  - Clarification of specs, and profiles, (“can you implement a router with your specs?”)
  - Compartmentalization of controller vs switch
  - Compliance specifications for testing
  - Cohesive system of parts instead of a bag of tools
  - Service Evolution to allow migration over time instead of fork lift upgrades
  - Quality of implementation
- Programming, from the user down into the network, as both tools and methods specified and working
  - Dev/Ops culture of engineering and development needed
  - Troubleshooting capabilities built in, and expertise in the developers to aid this
  - Quality of implementation

- Service description and setup based on user feedback created
  - End-to-end service specification - core/last mile/last inch,
  - multi-domain orchestration and stitching
  - monitoring included as part of the spec for the service (a method included)
  - Layer abstractions - can drill down, but don't have to
  - Ability to have different views of the same flow
  - User capabilities to set priority - (auctions as mechanism?)
  - Should describe more than just bandwidth - latency, jitter, etc.
  - Hard isolation guaranteed for performance and security

- Troubleshooting, at all layers, including tools, practices, and environments implemented, debugging, isolation, repair
  - Test harness abstraction with generic implementation through defined interfaces and peer level interactions
  - Develop expertise for troubleshooting and train professionals in these methods
  - Telemetry - streaming... for devices
- Validation and performance testing, tools and techniques defined
  - Discovery of capabilities and available services
  - Exchange mechanisms for brokering these services
  - Between layers, from the switch to the transport
  - End-to-end performance, dynamic PerfSonar, a fully formed model to use for validation
- Monitoring of flows, tables, changes, errors, implemented end to end