



# ESnet

ENERGY SCIENCES NETWORK

# Philosophy of Network Operating Systems and Intent APIs

Inder Monga  
CTO, ESnet

Technology Exchange,  
October 2015  
Cleveland, Ohio



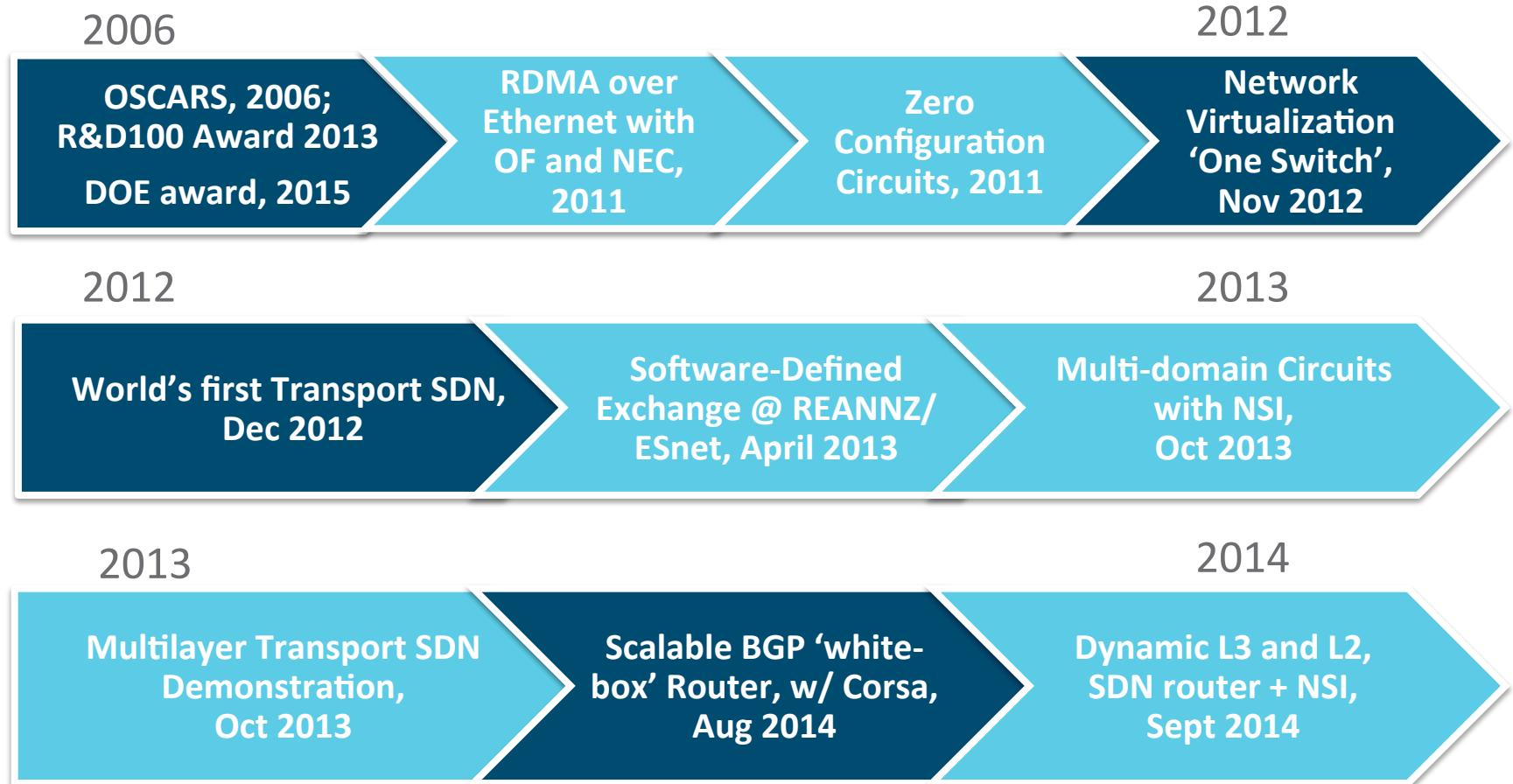
U.S. DEPARTMENT OF  
**ENERGY**  
Office of Science



# Agenda

- Background
- ESnet's Network Operating System
- What is your Intent?
- Summary and Next Talk!

# ESnet SDN investigations, in a nutshell:



Thanks to ESnet's SDN team



# ENOS: A Network Operating System for ESnet Testbed by Eric Pouyoul

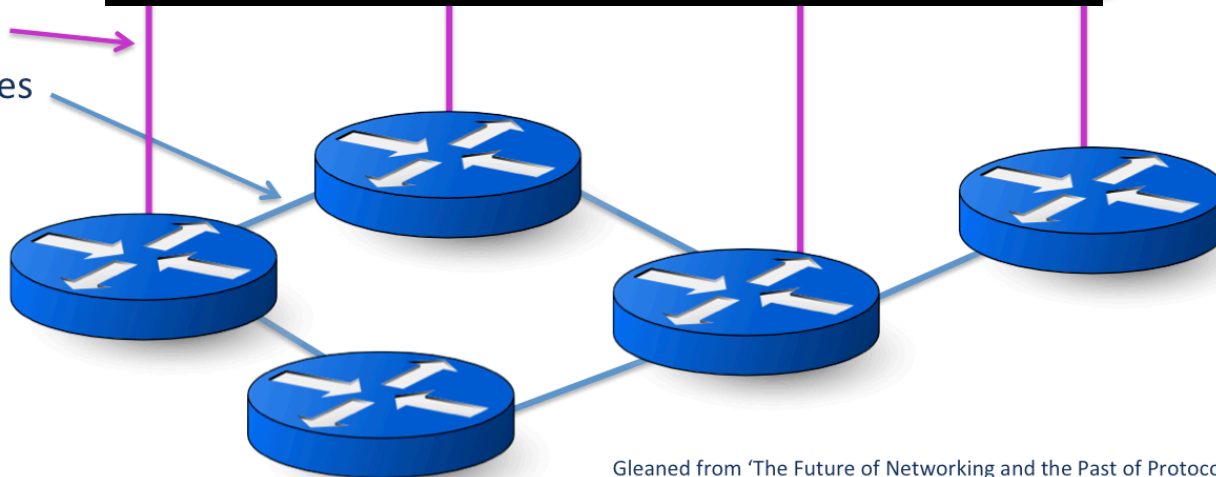
# Network OS – classic SDN picture

## Four Implicit SDN Assumptions

1. Control program configures all network switches
2. Switches relatively homogenous in role/function
3. Switches all use hardware (ASICs) for forwarding
4. Network dataplane is fairly simple (just forwarding)

**We were wrong on all of them.....**

Separate  
Control &  
Data Planes



Gleaned from 'The Future of Networking and the Past of Protocols'  
based on a presentation given by Scott Shenker in June of 2011

**ESnet**

# Confusion between SDN controller and Network Operating System

- The previous picture equates implementation of NOX (or a simple SDN controller like Ryu) to Network Operating System (NOS)
- If one draws an equivalence to a Computer Operating System,
  - a NOS needs more features than **Network State, Topology and OpenFlow message management**

- We started to design an equivalent Network OS + hypervisor (end of 2014)
- ..with the assumption that SDN controller is something we would leverage from open source....



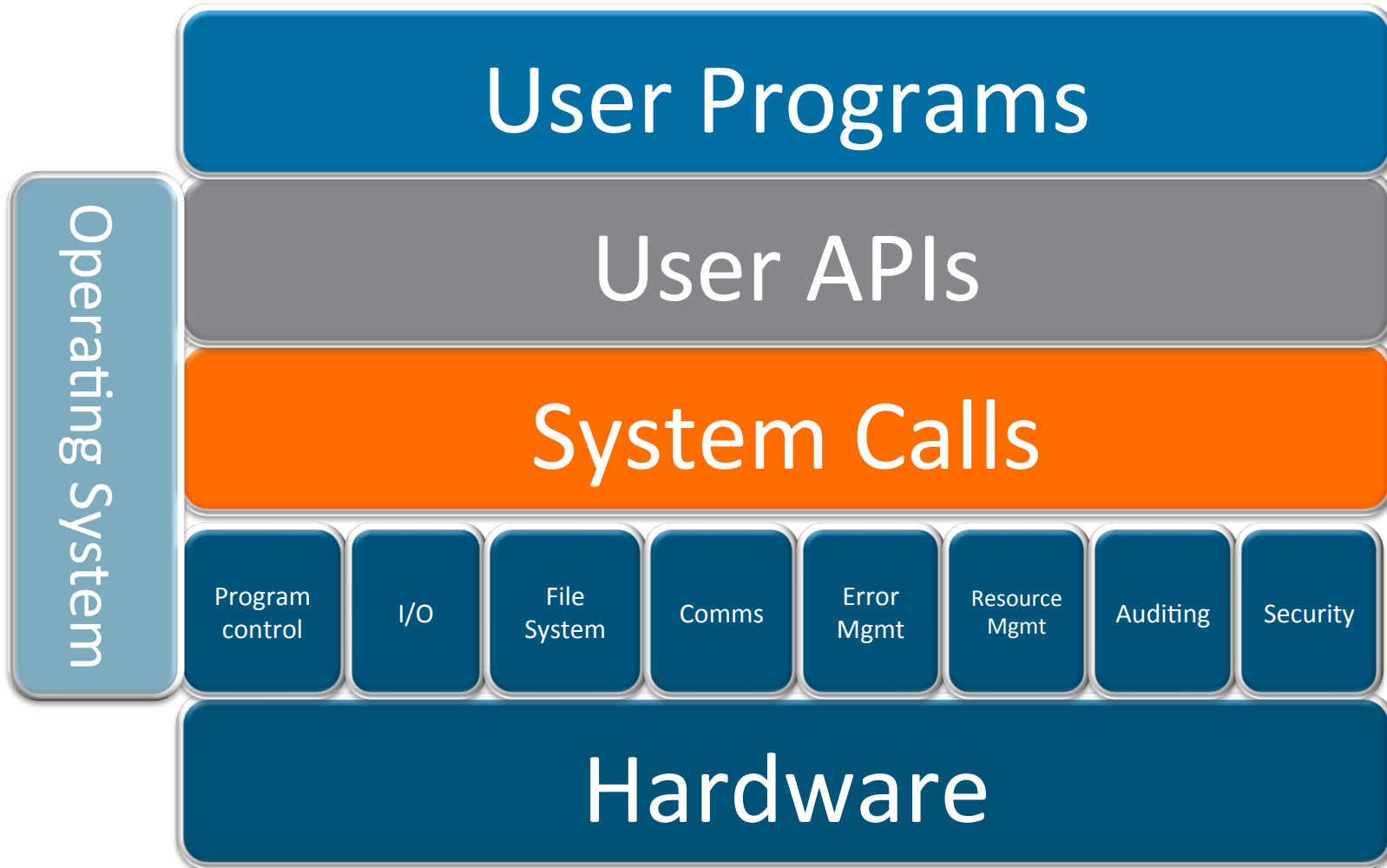
# Requirements when designing our Network OS

Philosophy of our SDN approach was to focus on

- **Multiple applications**
  - Support multiple, concurrent applications on the network
  - Agile resource allocation and resource management
- **Network operator**
  - Design with network operations and the current network operator skills
  - CLI access to the entire network
- **Security**
  - Enable different roles: users, applications, network operators, network collaborators, testbeds etc.
- **Multi-domain compatible**
  - NSI and BGP protocol support
- **Legacy internet compatible**
- **Simplicity of use and development**
  - Scripting interface, Python programming model
  - Focus on Intent APIs as research



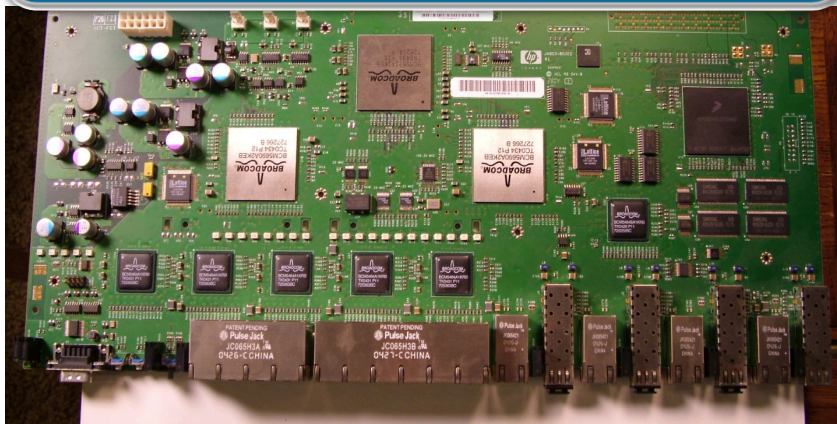
# Computer Operating System



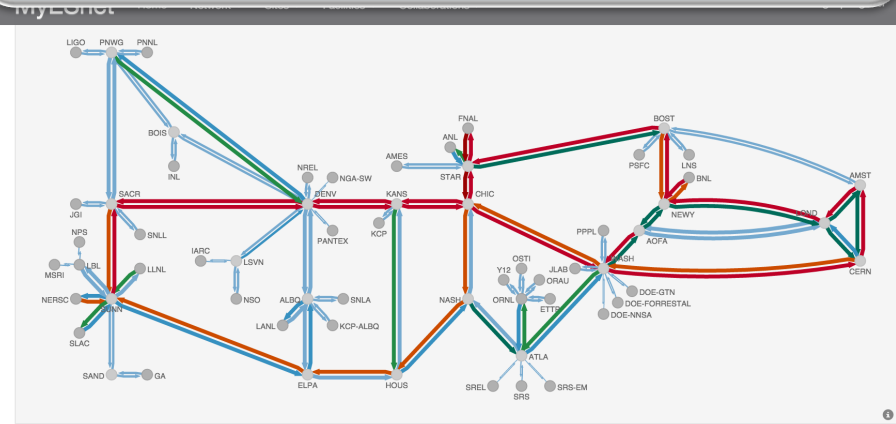


# Of course, Network Operating System runs on distributed sets of motherboards

## Computer OS

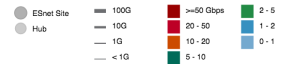


## Network OS

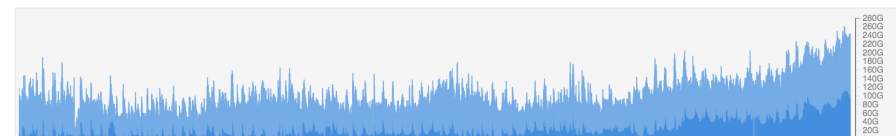


### Network Wide Total Traffic

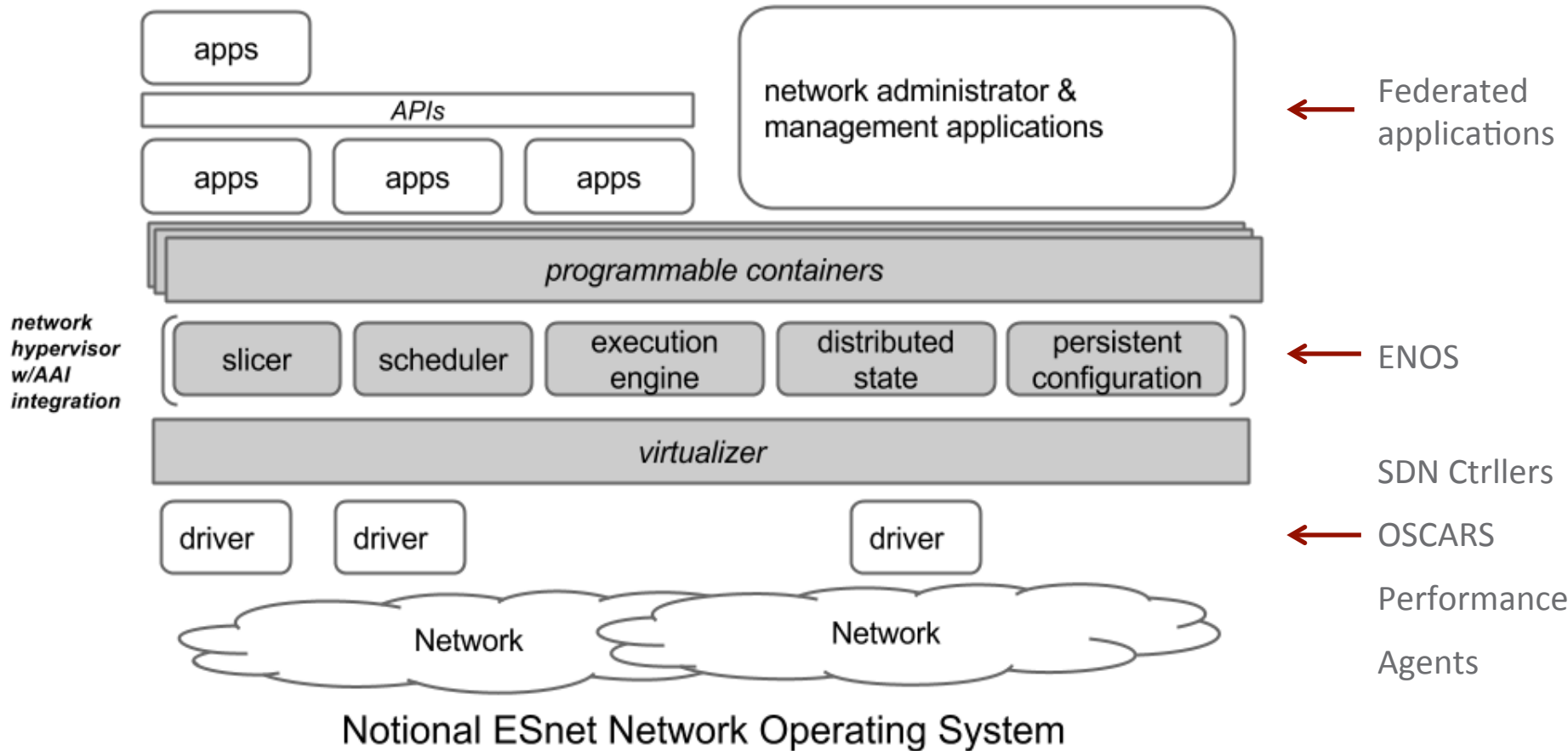
This map highlights the sites ESnet serves, the structure of the network and the current traffic load. Clicking on a node or edge will show details. Please note that this map does make some simplifications, click ⓘ on the map or chart for more details. The map of the transatlantic extension is a work in progress, more details are available in the info box.



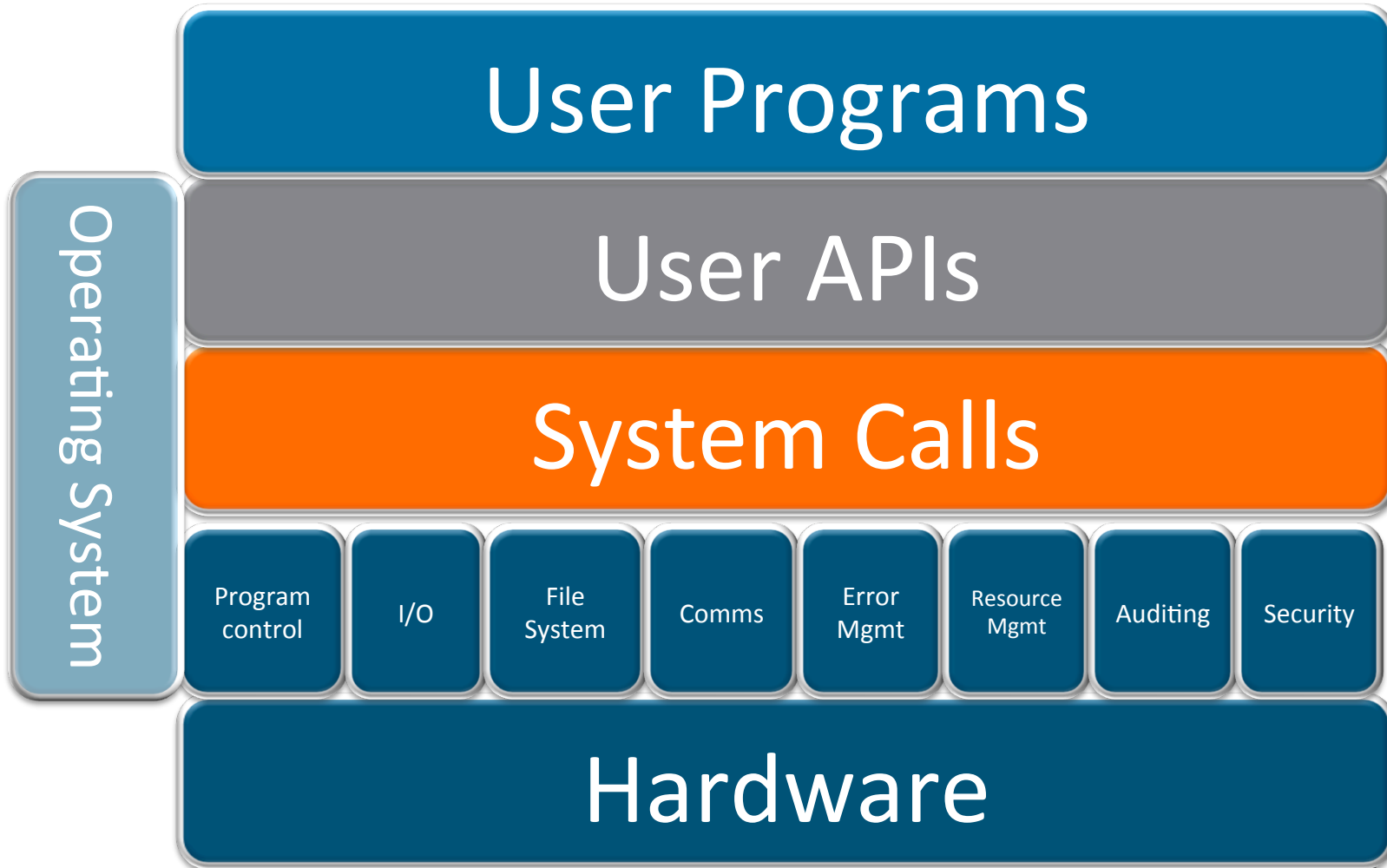
■ OSCARS ■ Total



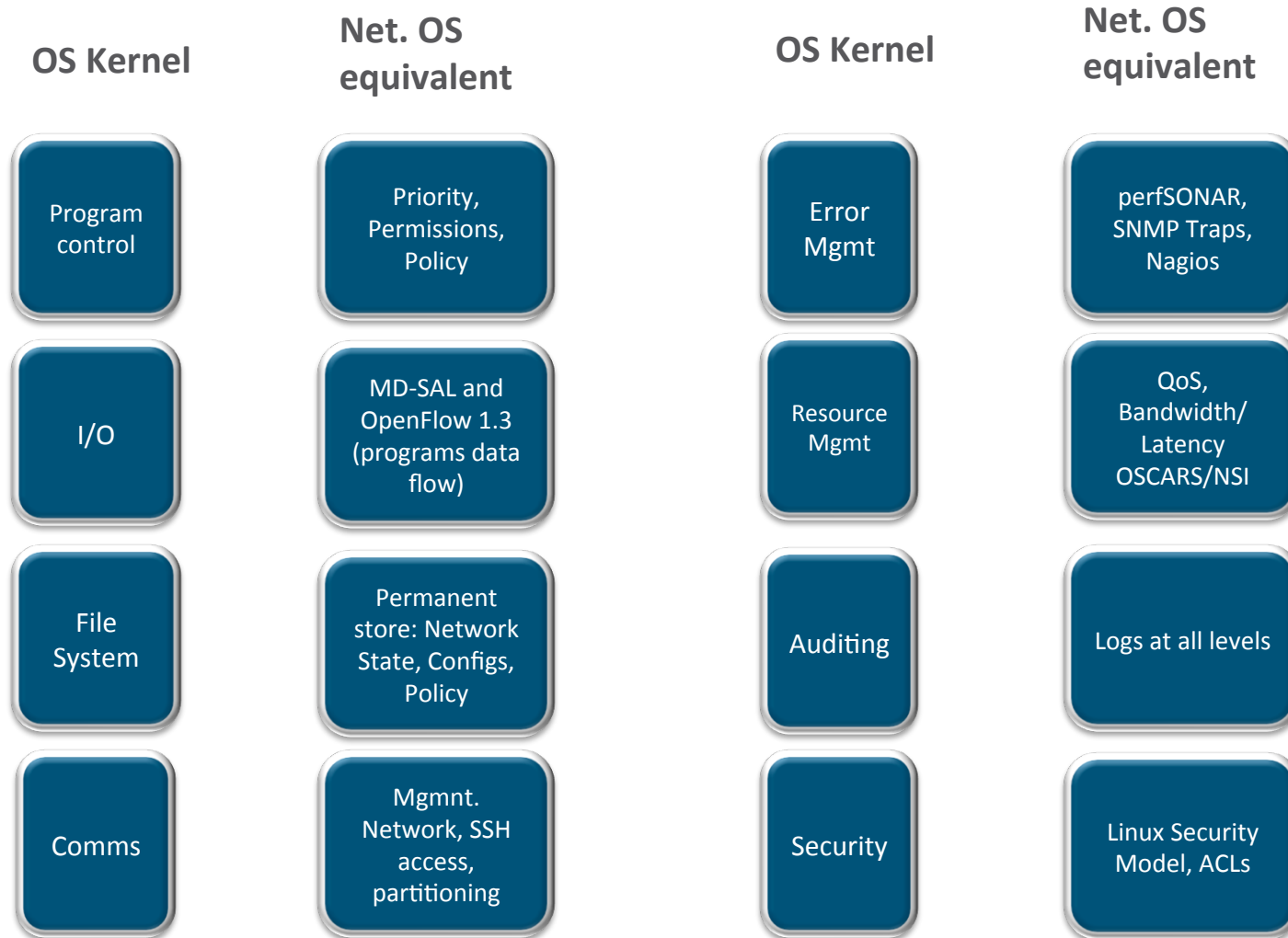
# High-level architecture of the Network OS



# Computer Operating System



# ESnet's Network Operating System



# What do the User Program's interact with?

User APIs

'declarative' interfaces or  
Intent-based Interfaces

System Calls

List of composable network services  
for example: point to multi-point  
service

# This is not an 'Intent' interface

Now that you have logged on, click on the Start button and navigate to your email account. Take five minutes to read, then make a cup of tea...



To increase productivity, Rob's boss decided to micro-manage his employees.

# Intent versus Prescription

## Intent

- **What I want, not how to do it**
- Portable, independent of protocol, vendor, media, etc.
- **“I want my headache to stop”**
- **“Bob is allowed to access the internet”**
- **“Please cut my lawn”**

## Prescription

- **How to do it** (Commands, rules, settings)
- Non-portable, dependent on protocol, vendor, media, etc.
- **“Give me two aspirin”**
- **“Send packets matching this {5-tuple} out port 11”**
- **“Take mower out of truck, fill gas and oil, pull starter cord, push onto lawn, ...”**

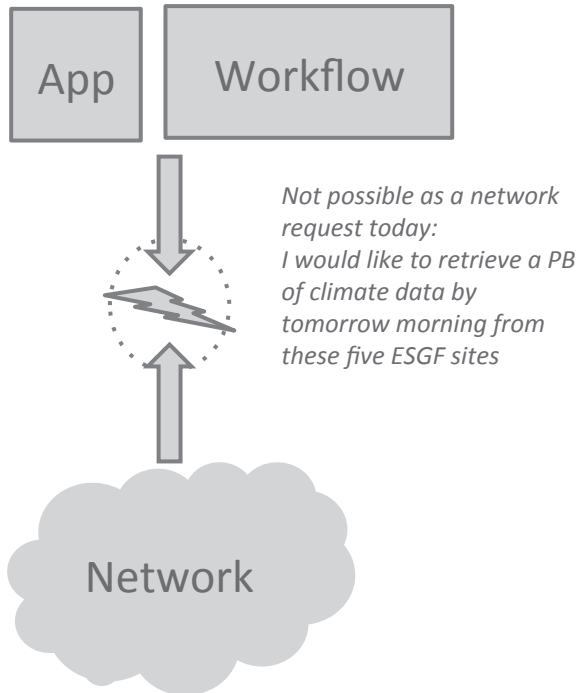
# How does it relate to the network?

- Multipoint VPN
  - ‘setup connectivity with isolation between points A, B and C on the topology’
- Traffic Tap
  - ‘Tap all TCP traffic going between Site A and Site B’
- Errors
  - ‘Alert me when packet loss exceeds 0.1%’
- Resiliency
  - ‘Create service backup, tolerant till 40% service degradation in bandwidth”

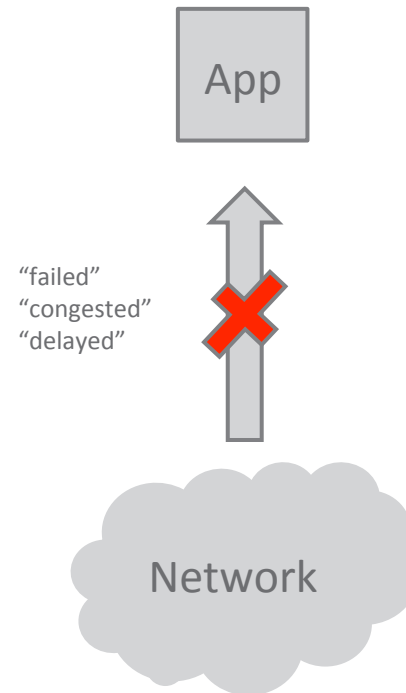


# Constrained application-network interaction prevents benefits of automation, orchestration, optimization

High-level network abstractions for applications-network to have a 'meaningful' conversation



No abstractions for networks to feedback 'service state' to the application



# Intent's are next SDN frontier, lots of community interest and efforts

- **Congress** - Policy as a Service, uses SQL Policy Language, Interfaces to OpenStack modules
- **Group-based Policy (GBP)** - Implementations in Docker (preliminary) and in OpenStack (advanced).
- **Network Intent Composition (NIC)** - OpenDayLight project that can be implemented used independently and over GBP as well. *Boulder* in Open Source SDN / ONF
- **ONOS** – Carrier-grade SDN network operating system that enables intent via JAVA modules, accessed via REST APIs.
- **NEMO** - Network Modeling language which could serve as a Domain Specific Language (DSL) providing abstraction of network models and operation patterns. Uses JAVA, JS, YANG.
- **Frenetic** – Class Programmable-friendly domain specific language to build modular network applications providing powerful abstractions. Pyretic - implementation of frenetic in python

# SDN Take-Away

- SDN is a journey R&E networks have been on for a while, but recently commercially formalized
- Innovator's dilemma gaps between established and the new ways
  - industry and R&E momentum will close those gaps
  - Maturity will still take some time
  - Focus on the problem being solved
- Plan for the SDN future – skillsets, trying, training and hiring

# Breaking News: L2 QoS enabled on ESnet's SDN Testbed controlled with OF 1.3

- ESnet worked with Corsa to create a joint L2/L3 pipeline with L2 QoS controlled via OF 1.3 protocol
  - <http://www.corsa.com/corsa-provides-metering-and-qos-capability-to-esnets-sdn-testbed/>
- Features enabled
  - Traffic isolation (guaranteed bandwidth circuits, best-effort circuits, best-effort IP, scavenger IP)
  - Prevent over-subscription of guaranteed bandwidth (using meters)
- After extensive testing, the SDN testbed has been upgraded with this feature
- Demonstration @ SC15



# Thank you!

- For more details contact, imonga at es dot net