# The Changing Face of Networks and Implications for Future HEP Computing Models

Michael Ernst, BNL/ATLAS

April 16, 2015

CHEP 2015

Okinawa

BROOKHAVEN NATIONAL LABORATORY

a passion for discovery

Office of Science
U.S. DEPARTMENT OF ENERGY

# Networking

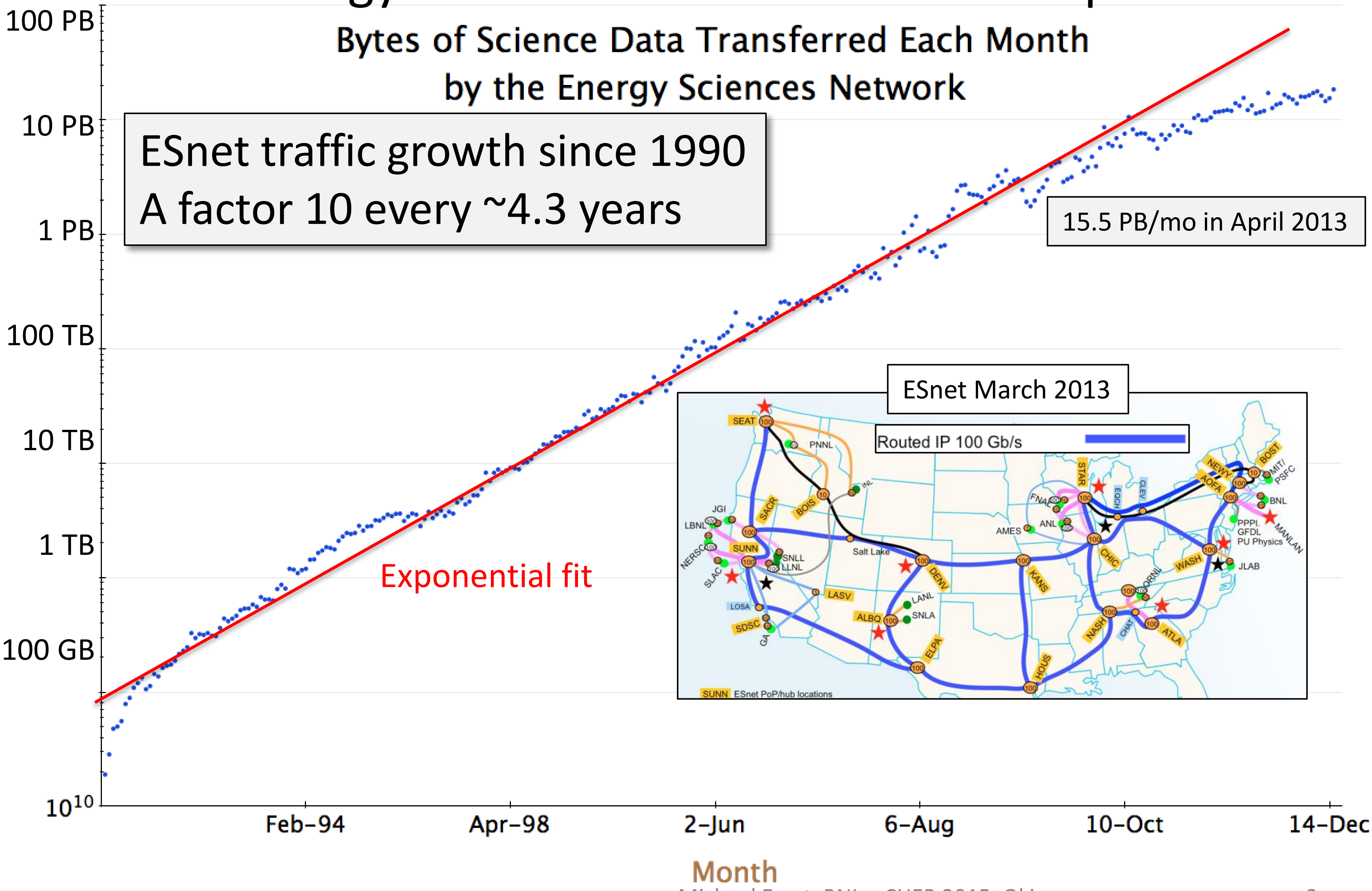
T. Wenaus @ CHEP2013

- Enabling backbone of LHC computing is reliable, high-bandwidth, feature-rich networks
- HEP was a pioneer in network-intensive science and international research networks, and continues to lead
  - Networks optimized for massive data flows, e.g. now testing the first 100Gb transatlantic production link – since Dec 2014 using the first 100Gb transatlantic production links
- **Making the most of the network translates to more science at lower computing cost**
  - Important that we design our workflows around this fact
- Next generation networks allow applications to interact with the network, reacting to conditions and proactively controlling it
  - e.g. work underway to integrate network awareness in job brokerage data distribution (PheDex) and job brokerage (PanDA)

**In general it's much cheaper to transport data than to store it**

# Networking growth has been dramatic

## US Energy Sciences Network as an example



Bytes of Science Data Transferred Each Month by the Energy Sciences Network

ESnet traffic growth since 1990
A factor 10 every ~4.3 years

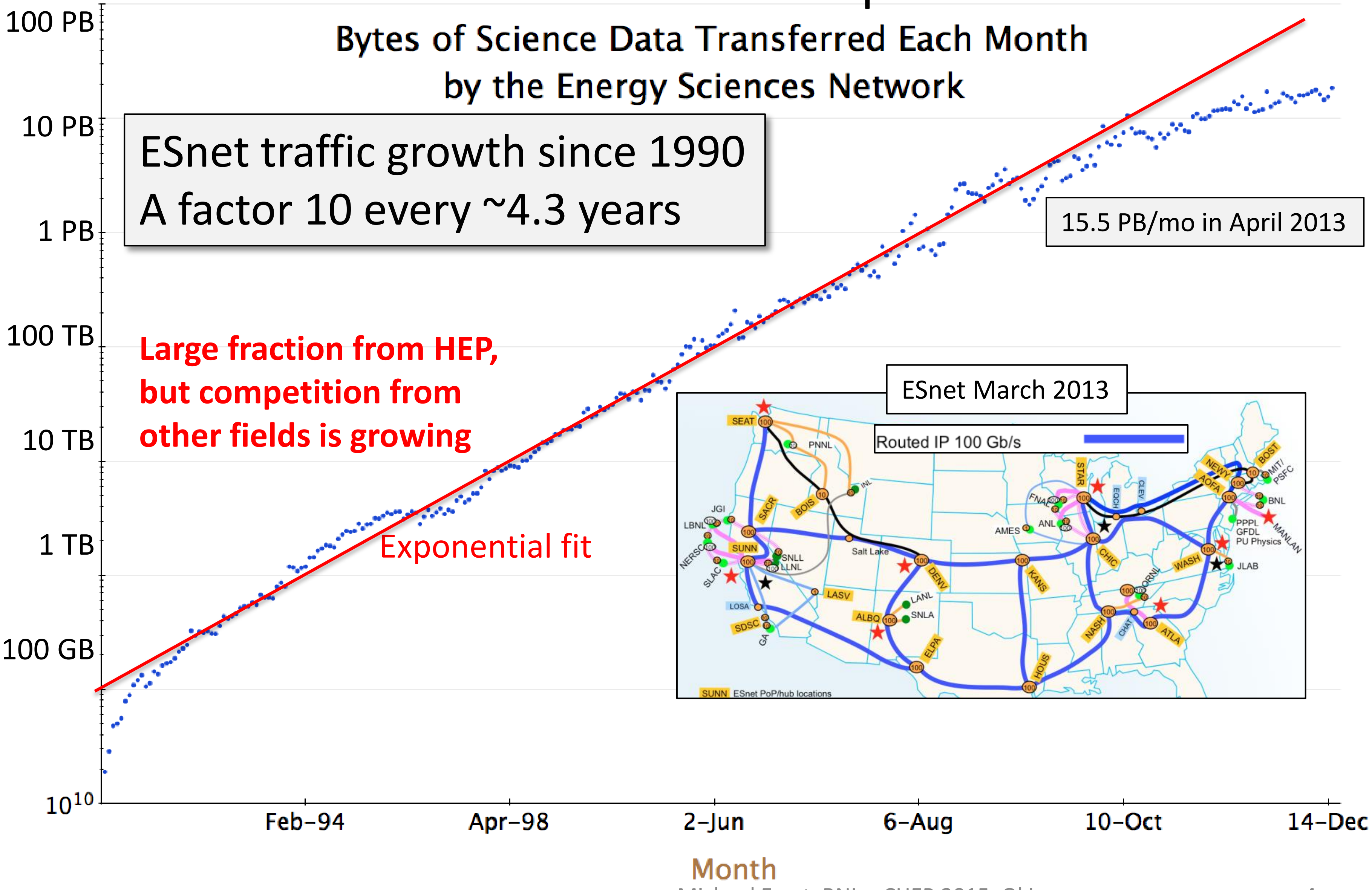
15.5 PB/mo in April 2013

Exponential fit

ESnet March 2013

# Networking growth has been dramatic

## US ESnet as an example



ESnet traffic growth since 1990
A factor 10 every ~4.3 years

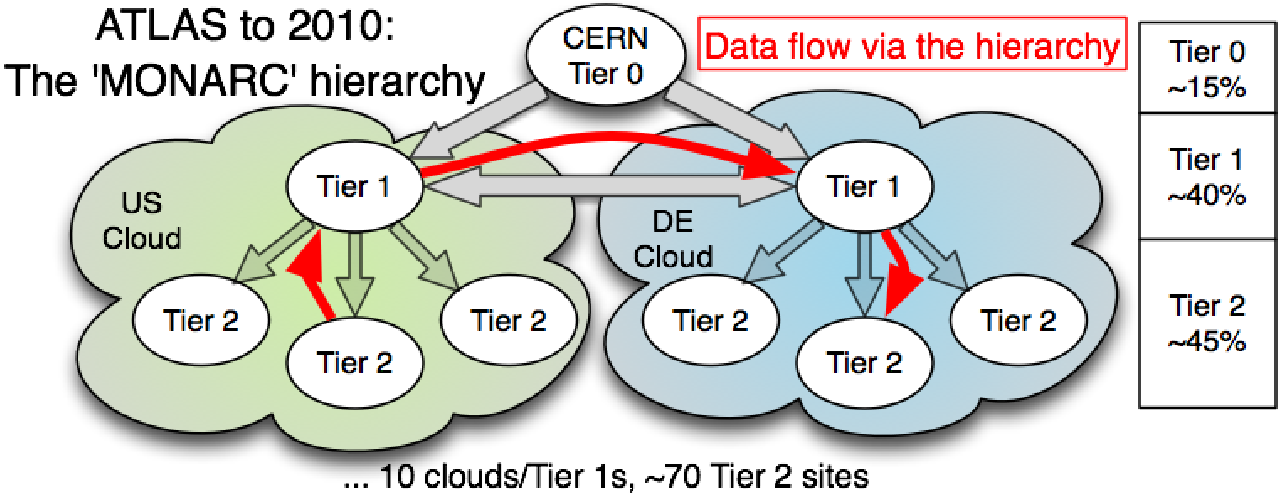
15.5 PB/mo in April 2013

Large fraction from HEP, but competition from other fields is growing

Exponential fit

ESnet March 2013

# Networking has been a critical enabler for evolving LHC computing models – ATLAS as example



**Original model:**

Static strict hierarchy

Multi-hop data flows

Lesser demands on Tier 2 networking

Virtue of simplicity

**Designed for <~2.5 Gb/s within the hierarchy**

**Today:**

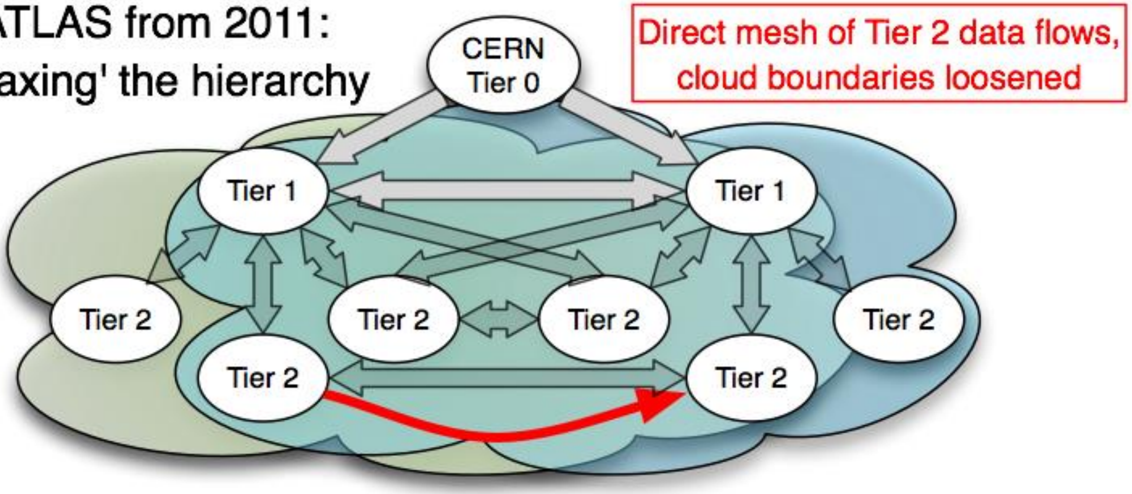
**Bandwidths 10-100 Gb/s, not limited to the hierarchy**

Flatter, mostly a mesh

Sites contribute based on capability

**Greater flexibility and efficiency**

**More fully utilize available resources**

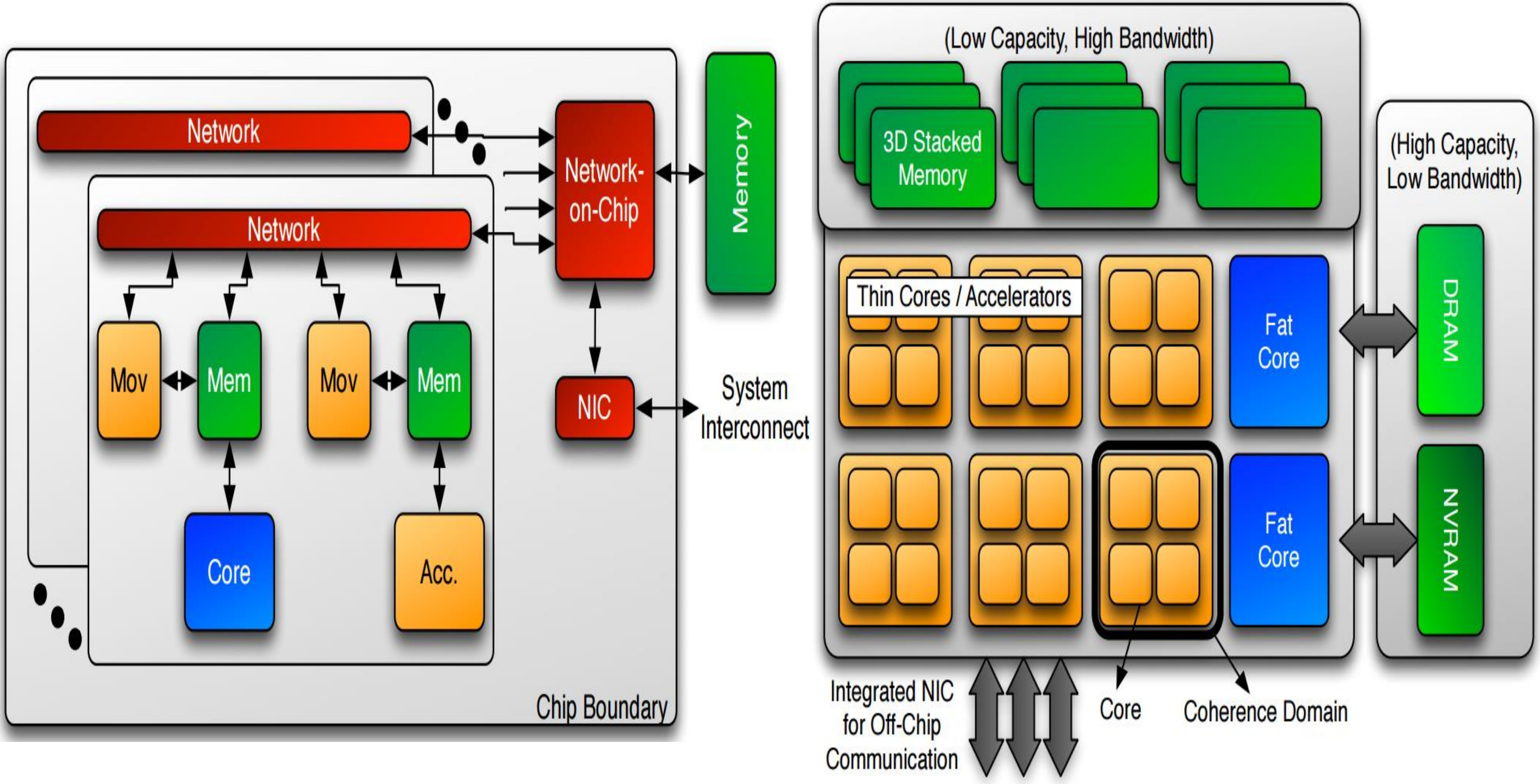


T. Wenaus @ CHEP2013

# Outline

- The changing landscape in facilities components and HEP/NP Applications
- Recap of HEP/NP Networking, how it has evolved over time and Issues
- Virtualizing Networking and how it could benefit HEP/NP Applications
- Data and Network Integration
- Conclusions

# Computer Architecture is evolving - Abstract Machine Model (HPC)



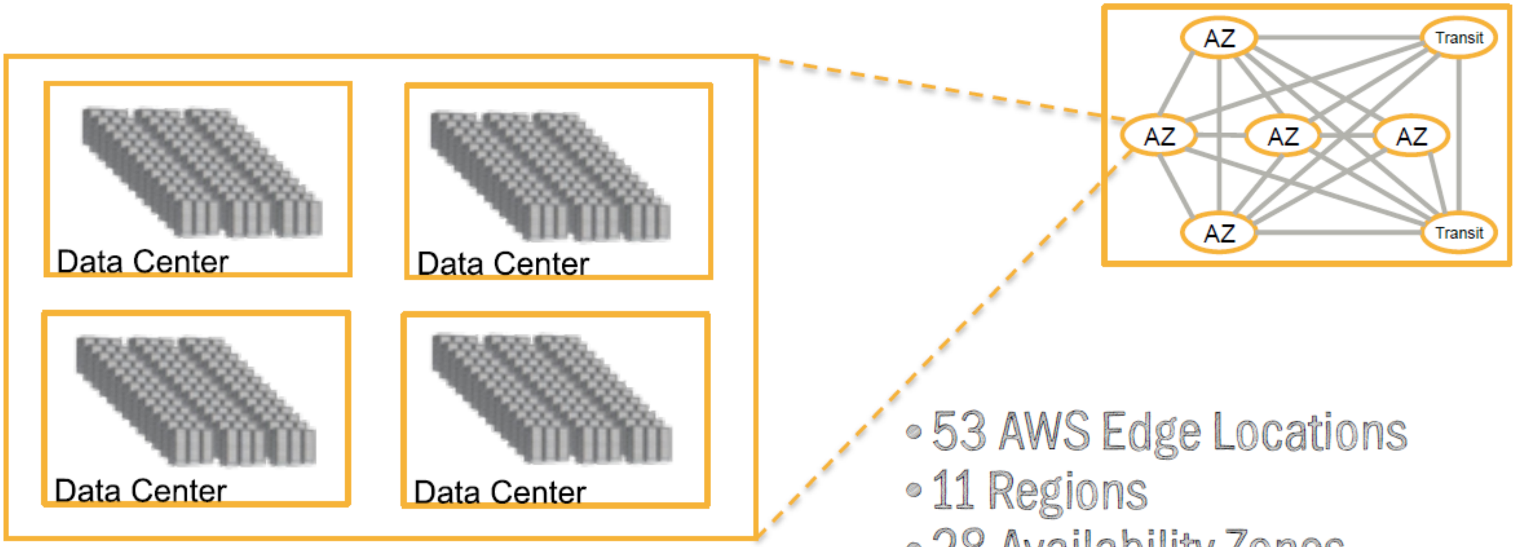
Networking becomes integral part of internal Chip Architecture, extending across machines and beyond

J. Shalf et al

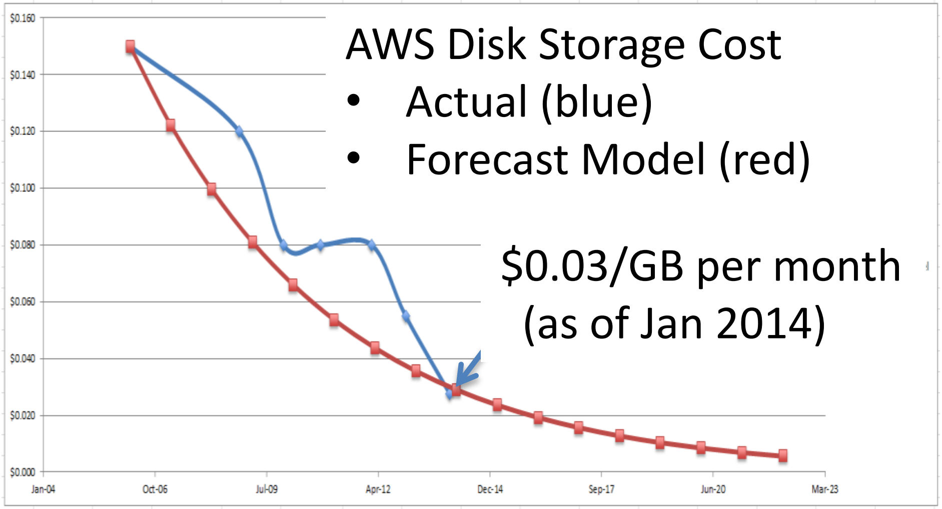
# The potential of Commercial Cloud Resources

## A Snapshot of AWS Global Capacity

J. Kinney (AWS)



- 53 AWS Edge Locations
- 11 Regions
- 28 Availability Zones
- 2 or more AZs per Region
- 1-6 Data Centers per AZ
- 50,000-80,000+ servers per DC
- Up to 102 Tbps provisioned to each DC



AWS Disk Storage Cost

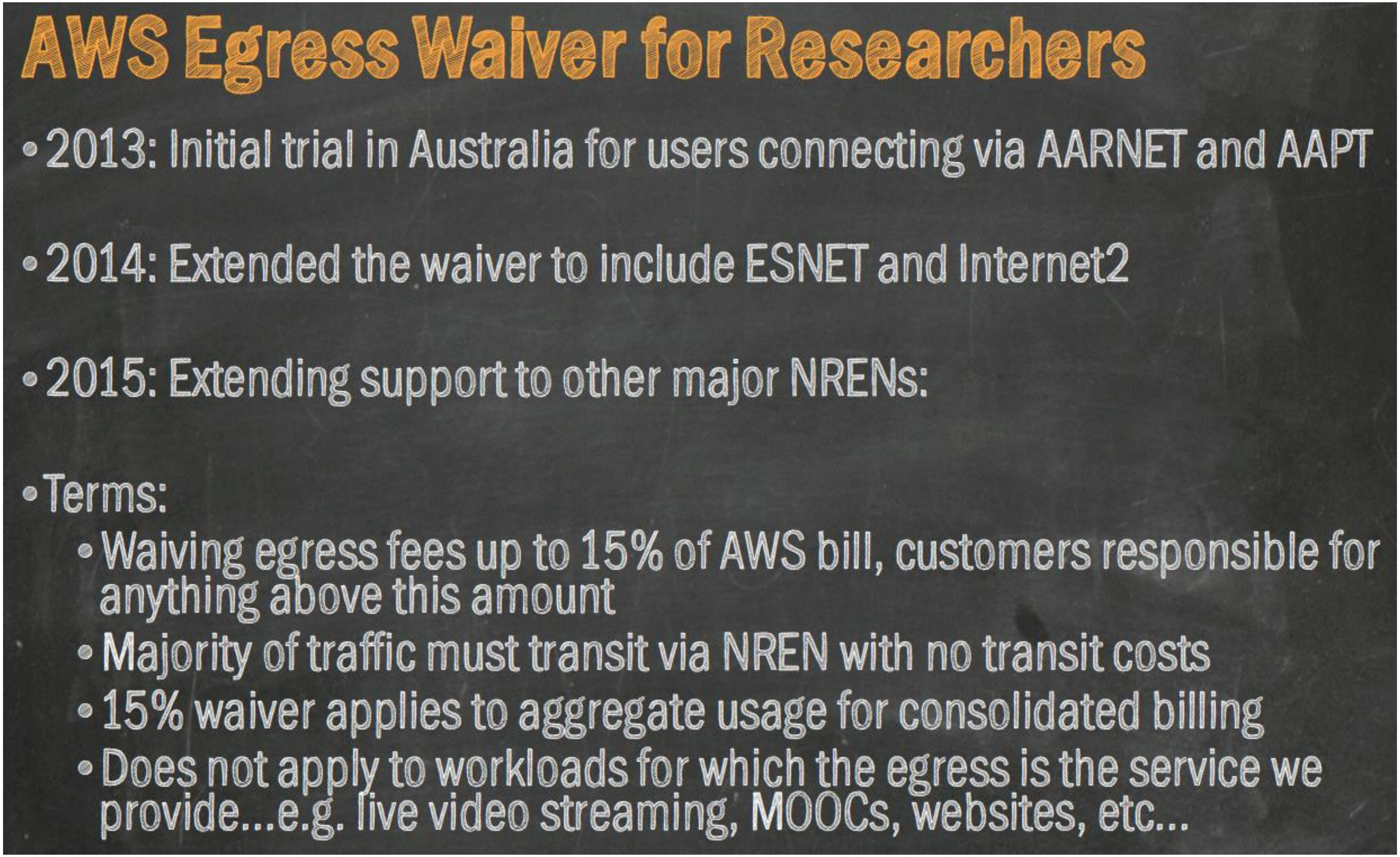
- Actual (blue)
- Forecast Model (red)

$0.03/GB per month (as of Jan 2014)

Cost for Compute (AWS Spot) quickly approaching cost for dedicated resources

- A cost-efficient way to serve peak demand

# AWS – ESnet – ATLAS Collaboration



## AWS Egress Waiver for Researchers

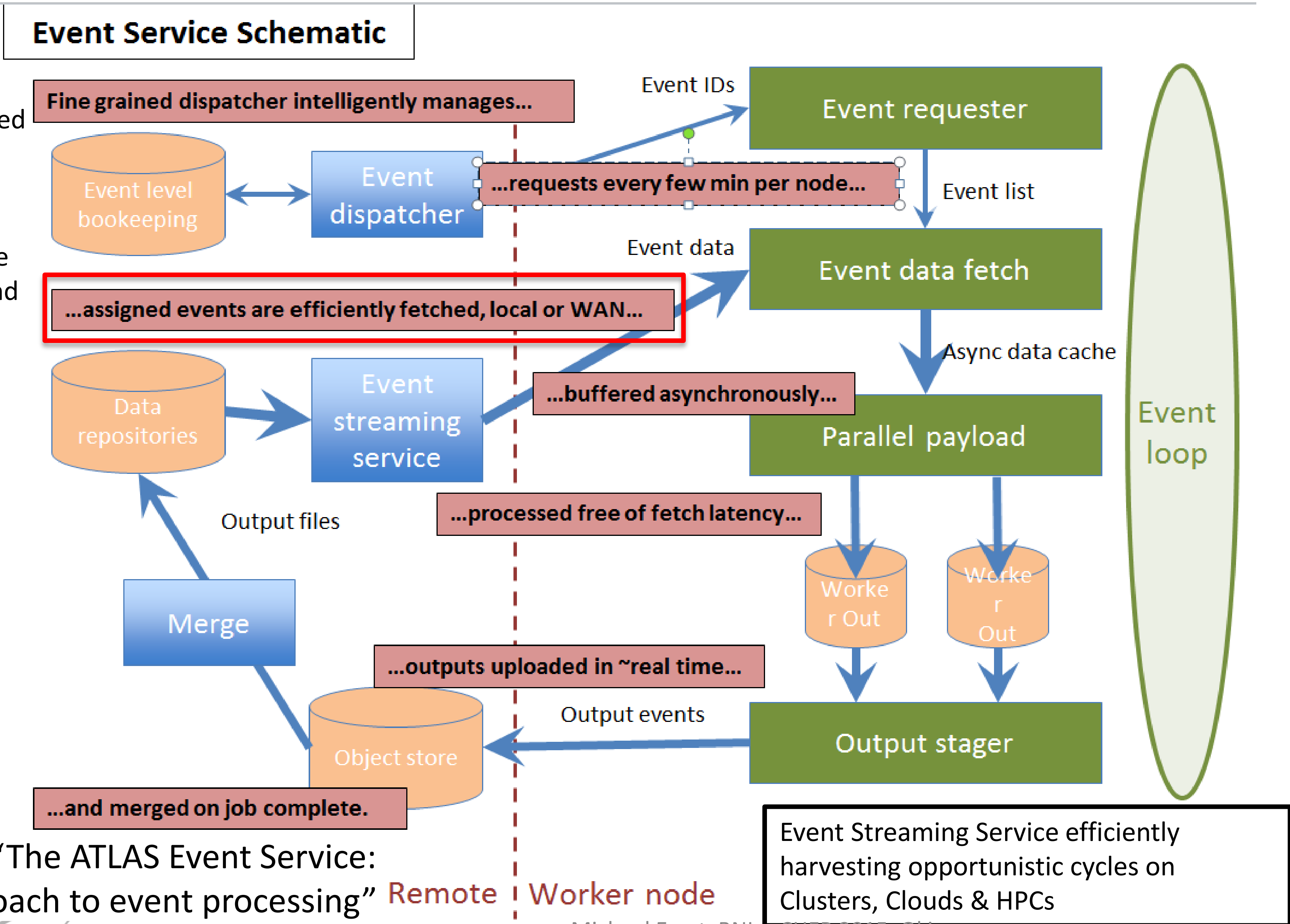
- 2013: Initial trial in Australia for users connecting via AARNET and AAPT
- 2014: Extended the waiver to include ESNET and Internet2
- 2015: Extending support to other major NRENs:
- Terms:
  - Waiving egress fees up to 15% of AWS bill, customers responsible for anything above this amount
  - Majority of traffic must transit via NREN with no transit costs
  - 15% waiver applies to aggregate usage for consolidated billing
  - Does not apply to workloads for which the egress is the service we provide...e.g. live video streaming, MOOCs, websites, etc...

J. Kinney (March 2015)

# Example: Leveraging Networks for fine-grained Event Processing

V. Tsulaia et al

Requested events delivered to a client agnostic as to event origin (cache, remote file, on-demand generation)



Event Streaming Service efficiently harvesting opportunistic cycles on Clusters, Clouds & HPCs

T. Wenaus: "The ATLAS Event Service: A new approach to event processing"

# Bandwidth Requirements to Storage growing by 50x from Run 1 to Run 4

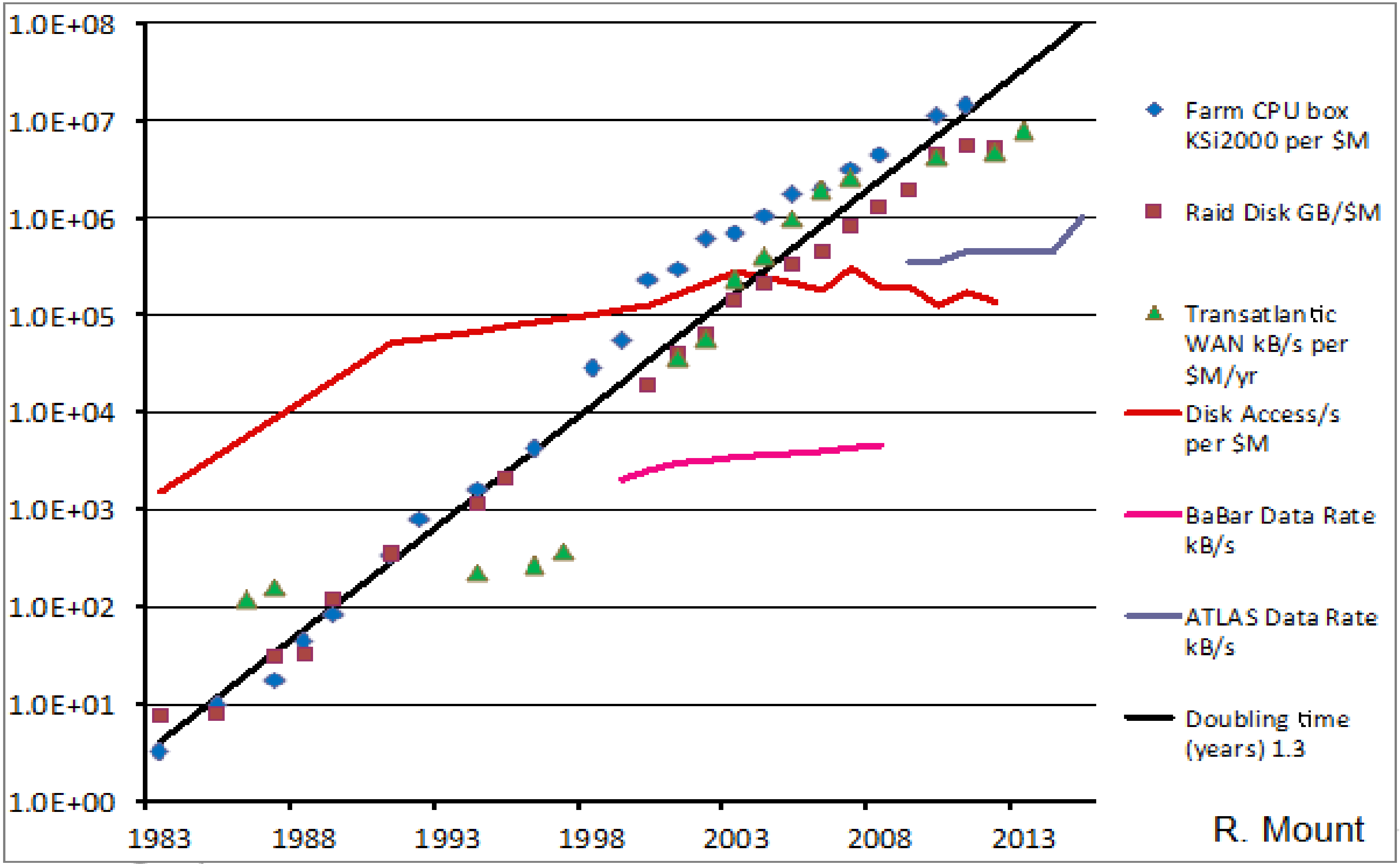
- Driving Parameters

| | # of Trigger levels | Level-xRate (kHz) | | Event Size (MB) | Network BW (GB/s) | Storage | |
|---|---|---|---|---|---|---|---|
| | | | | | | GB/s | kHz |
| Run 1 | 3 | Lvl-1<br>HLT | 75<br>~0.4 | ~1 | 10 | 0.5 | ~0.4 |
| Run 2 | 2 | Lvl-1<br>HLT | 100<br>1 | ~2 | 50 | 1 | 1 |
| Run 3 | 2 | Lvl-1<br>HLT | 100<br>1 | ~2 | 50 | 1 | 1 |
| Run 4 | 3 | Lvl-1<br>HLT | 400<br>10 | ~5 | 2000 | 25 | 10 |

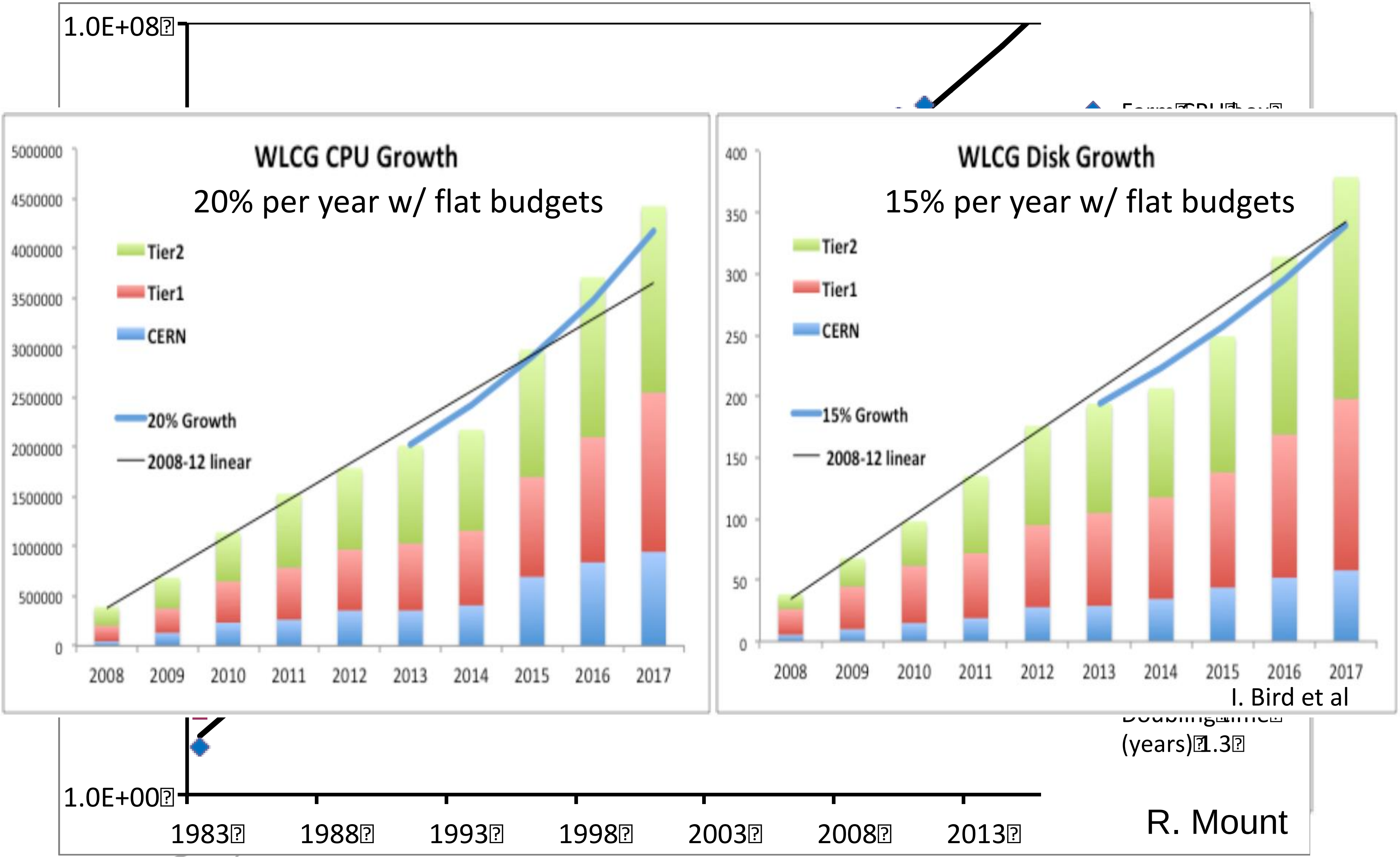
ATLAS DAQ/HLT Upgrade Plans

- ALICE even more challenging in Run 3
  - 100x Network rate compared to Run 1
  - 80 GB/s to storage

# The Past: Exponential growth of CPU, Storage, Networks – The Resource Dilemma



# The Past: Exponential growth of CPU, Storage, Networks – The Resource Dilemma



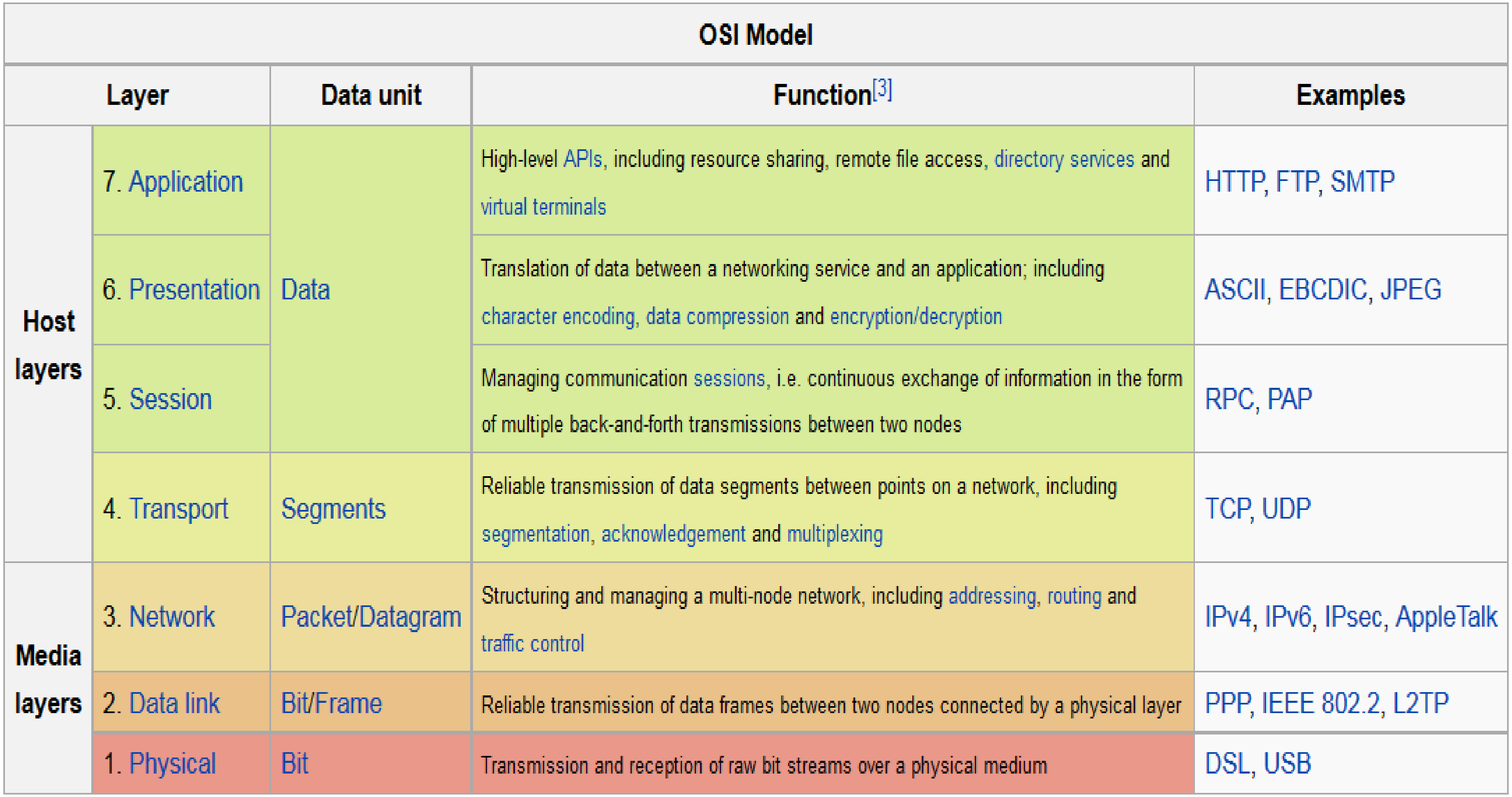
20% per year w/ flat budgets

15% per year w/ flat budgets

I. Bird et al

R. Mount

# Understanding the Network Protocol Stack - The OSI Reference Model for Communication

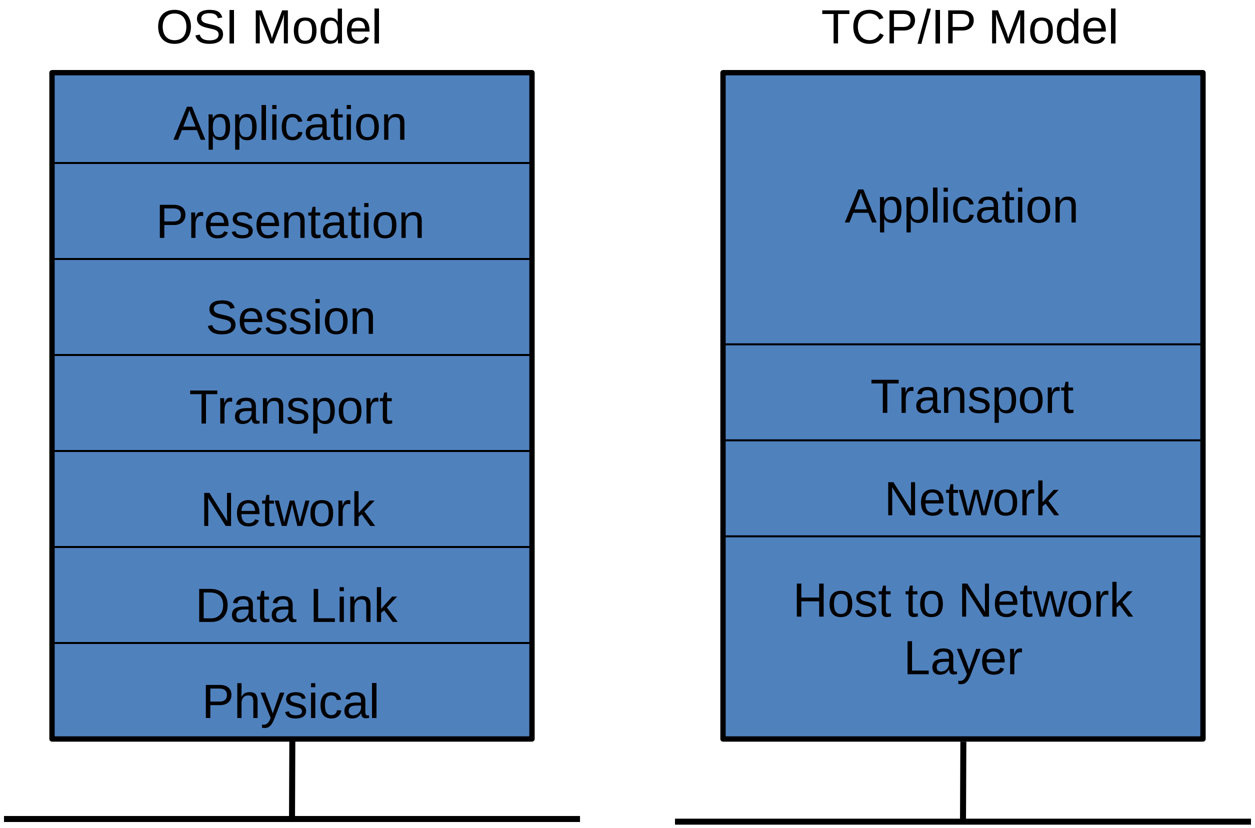


**OSI Model**

| Layer | | Data unit | Function[3] | Examples |
|---|---|---|---|---|
| Host layers | 7. Application | Data | High-level APIs, including resource sharing, remote file access, directory services and virtual terminals | HTTP, FTP, SMTP |
| | 6. Presentation | | Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption | ASCII, EBCDIC, JPEG |
| | 5. Session | | Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes | RPC, PAP |
| | 4. Transport | Segments | Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing | TCP, UDP |
| Media layers | 3. Network | Packet/Datagram | Structuring and managing a multi-node network, including addressing, routing and traffic control | IPv4, IPv6, IPsec, AppleTalk |
| | 2. Data link | Bit/Frame | Reliable transmission of data frames between two nodes connected by a physical layer | PPP, IEEE 802.2, L2TP |
| | 1. Physical | Bit | Transmission and reception of raw bit streams over a physical medium | DSL, USB |

Wikipedia

# Application – Network Interaction



OSI Model

Application

Presentation

Session

Transport

Network

Data Link

Physical

TCP/IP Model

Application

Transport

Network

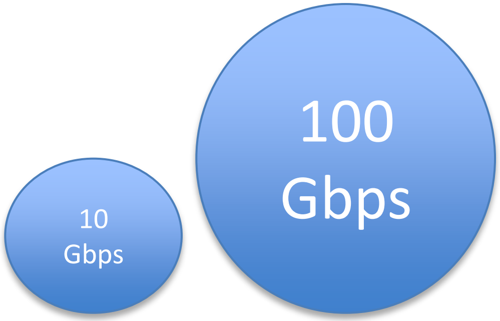
Host to Network Layer

What's widely used in HEP is a simplified Model

- Applications drop packets into the network layer, routes them towards the destination – at best effort
- Not much has changed in the last 2 Decades

# What HEP Applications care about

- Bandwidth:
  - – Capacity of a given network
- ➢ Network property



10 Gbps

100 Gbps

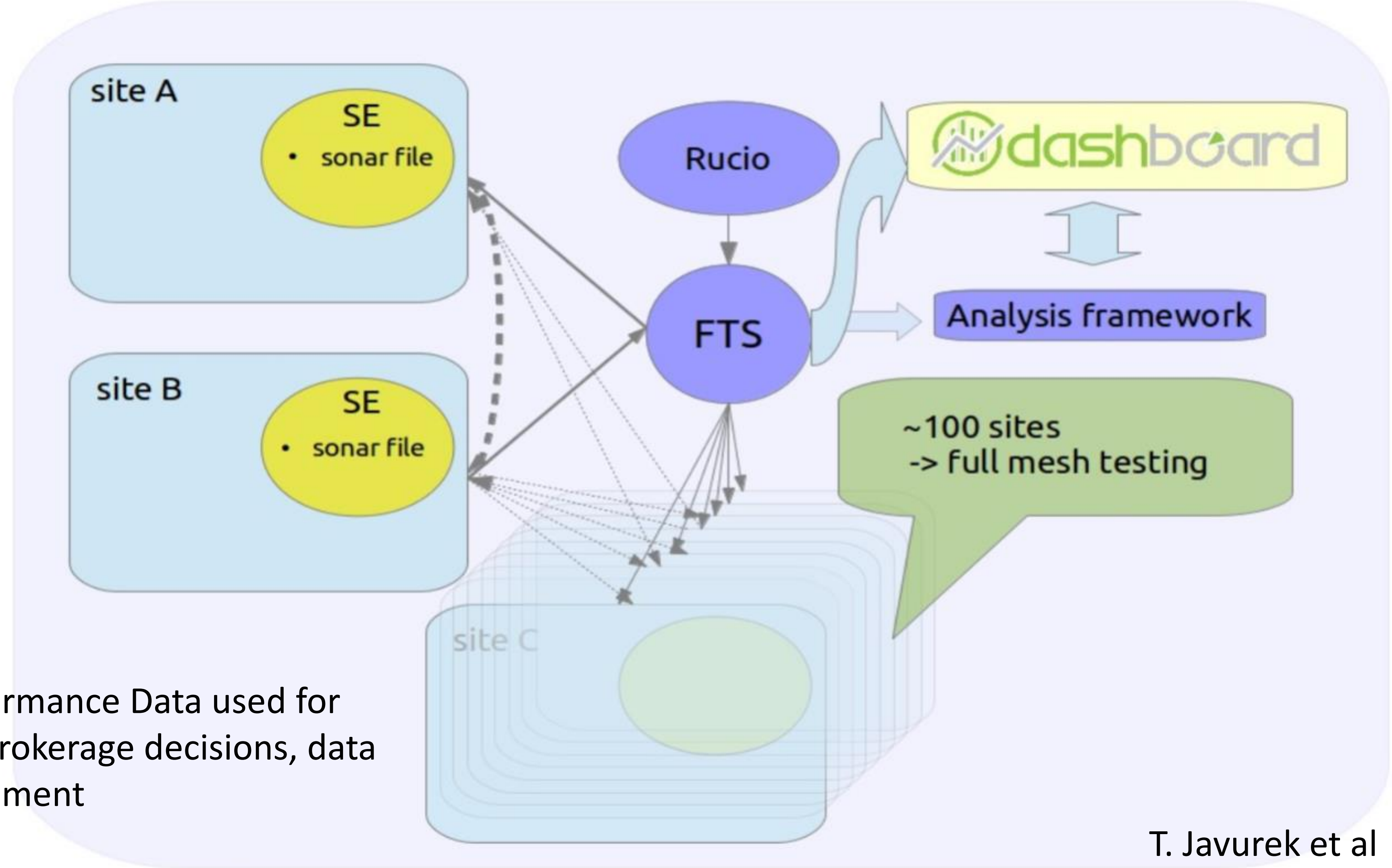
- *Predictable* Throughput:
  - – How many bits/second can be carried between any two points of the network
- ➢ Application or end-to-end property
- ➢ With quickly rising line rates applications running increasingly into problems utilizing available bandwidth
  - ➢ T. Maier: “ATLAS I/O Performance Optimization in As-Deployed Environments”

# Interacting with the Network

- Slowly moving from using an opaque service to
  - Understanding “the Network”
  - Making the Network an integral part of the Distributed Facilities and making Applications network-aware
- Historically HEP applications did not interact with Network Management/Control Plane entities
  - For the last ~5 years Providers offer the HEP community WAN Virtual Circuit technology to support Bandwidth on Demand (BoD)
  - Technically works well within a network domain but in the past many interoperability issues in multi-domain networks
    - Agreeing on standards, i.e. the Network Services Interface (NSI), and jointly working on implementations has helped to overcome some of the problems
  - Software Defined Networking (SDN) is another technology applications could benefit from by having an application-driven/dynamically created virtualized network environment optimized for a variety of HEP/NP workflows

# Understanding the "Network"- Application-level Monitoring

Example: ATLAS

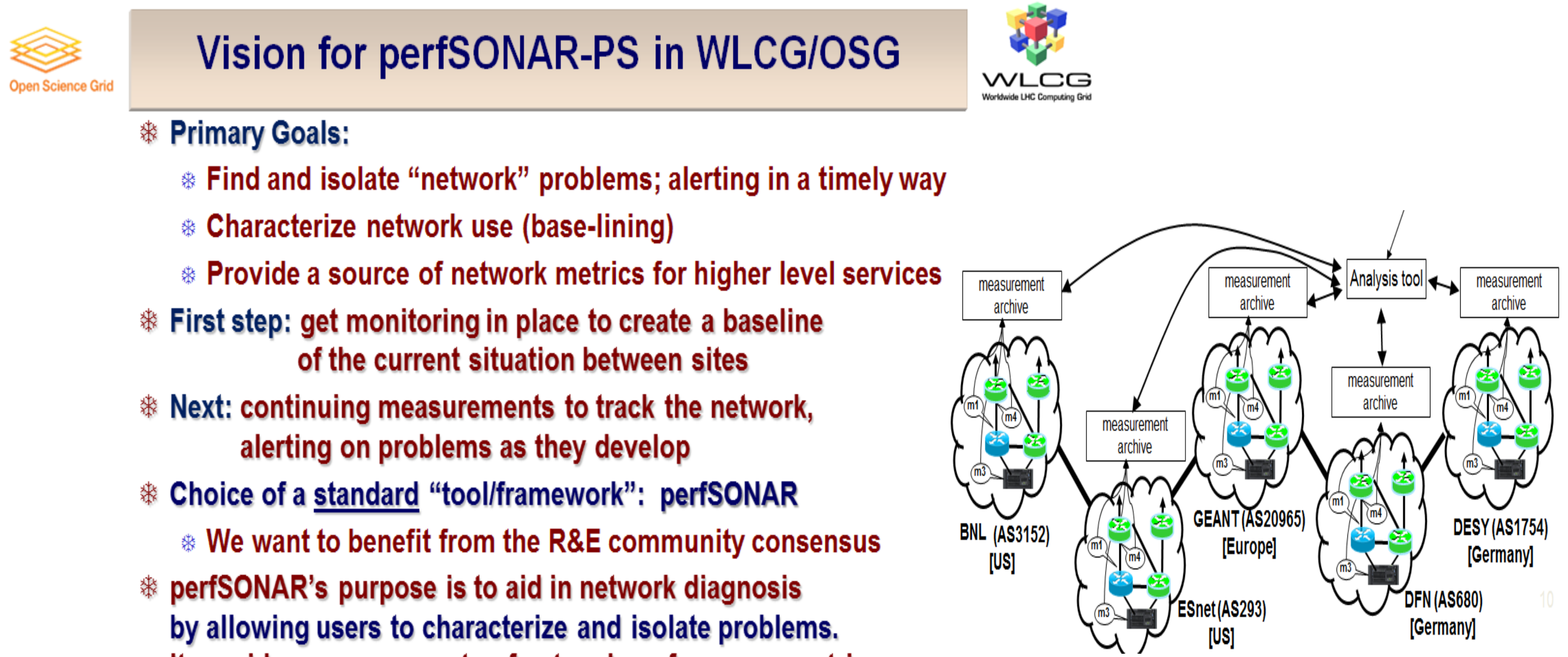


Performance Data used for job brokerage decisions, data placement

T. Javurek et al

# Understanding the “Network” - perfSONAR

S. McKee



Open Science Grid

## Vision for perfSONAR-PS in WLCG/OSG

WLCG
Worldwide LHC Computing Grid

- **Primary Goals:**
  - Find and isolate “network” problems; alerting in a timely way
  - Characterize network use (base-lining)
  - Provide a source of network metrics for higher level services
- First step: get monitoring in place to create a baseline of the current situation between sites
- Next: continuing measurements to track the network, alerting on problems as they develop
- Choice of a standard “tool/framework”: perfSONAR
  - We want to benefit from the R&E community consensus
- perfSONAR’s purpose is to aid in network diagnosis by allowing users to characterize and isolate problems. It provides measurements of network performance metrics over time as well as “on-demand” tests.

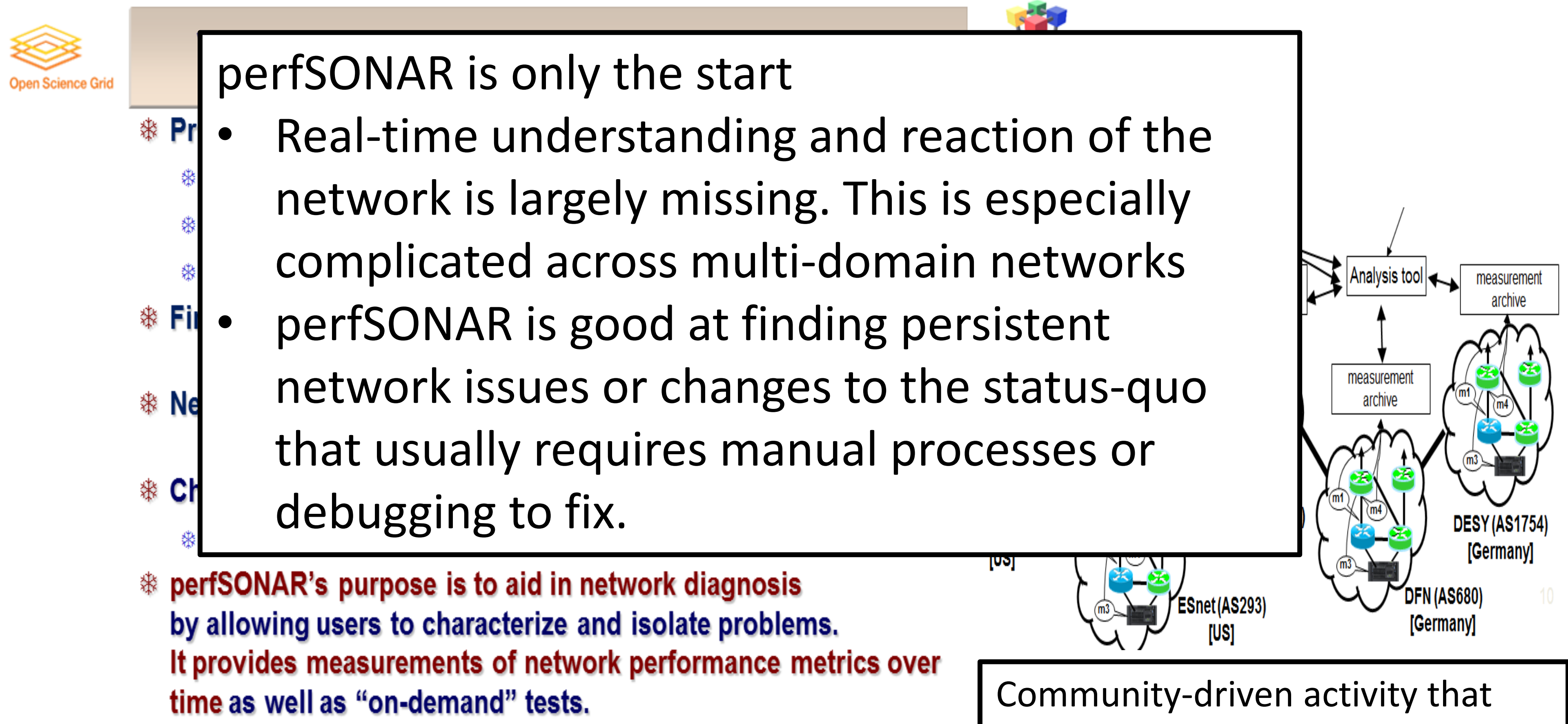
The perfSONAR development is supported by the perfSONAR Consortium (ESnet, I2, GEANT Indiana University and others)

Community-driven activity that requires non-negligible Effort

- Network-provided performance data would be advantageous

# Understanding the "Network" - perfSONAR

S. McKee



Open Science Grid
Pr
Fi
Ne
Ch
Analysis tool
measurement archive
measurement archive
m1
m4
m3
DESY (AS1754) [Germany]
DFN (AS680) [Germany]
ESnet (AS293) [US]
[US]
10

- perfSONAR's purpose is to aid in network diagnosis by allowing users to characterize and isolate problems. It provides measurements of network performance metrics over time as well as "on-demand" tests.

perfSONAR is only the start

- Real-time understanding and reaction of the network is largely missing. This is especially complicated across multi-domain networks
- perfSONAR is good at finding persistent network issues or changes to the status-quo that usually requires manual processes or debugging to fix.

Community-driven activity that requires non-negligible Effort

- Network-provided performance data would be advantageous

The perfSONAR development is supported by the perfSONAR Consortium (ESnet, I2, GEANT Indiana University and others)

# Interacting with the Network

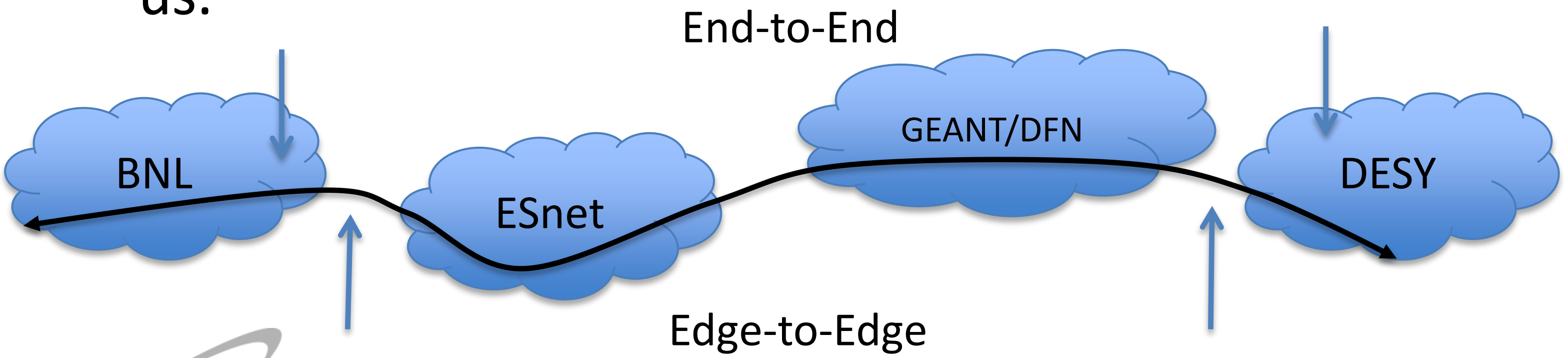
- Slowly moving from using an opaque service to
  - Understanding "the Network"
  - Making the Network an integral part of the Distributed Facilities and making Applications network-aware
- Historically HEP applications did not interact with Network Management/Control Plane entities
  - For the last ~5 years Providers offer the HEP community WAN Virtual Circuit technology to support Bandwidth on Demand (BoD)
  - Technically works well within a single network domain but in the past many interoperability issues in multi-domain networks
    - Agreeing on standards, i.e. the Network Services Interface (NSI), and jointly working on implementations has helped to overcome some of the problems
      - Works well edge-to-edge (networks) but not end-to-end (transfer hosts)
  - Software Defined Networking (SDN) SDN is moving towards providing the right abstractions and APIs to applications - networks have a greater chance of communicating productively - with feedback control

# Network Domains

The Network path between sites may traverse infrastructure provided and operated by different organizations

- Administrative boundaries are broad and can be arbitrary
- Single Domain
  - Single administrative management entity
- Multi-domain
  - Multiple administrative management entities
    - This is what's most relevant to HEP Workflows

➢ The multi-domain problem is deeply characteristic of large-scale science, and almost unknown in the commercial sphere. It's one of those problems the commercial sector will not solve for us.



# Limitations with BoD

- Most applications not familiar with the topology might assume Bandwidth on Demand capabilities which may not be  physically possible
- ‘Guaranteed Reservations’ of other applications might consume all resources
  - Even though the traffic profile indicates a lot of headroom
- An intermediate network domain might have resource constraints
  - Path finding needs to be intelligent
  - Path computation may take a lot of cycles if network is ‘reservation congested’

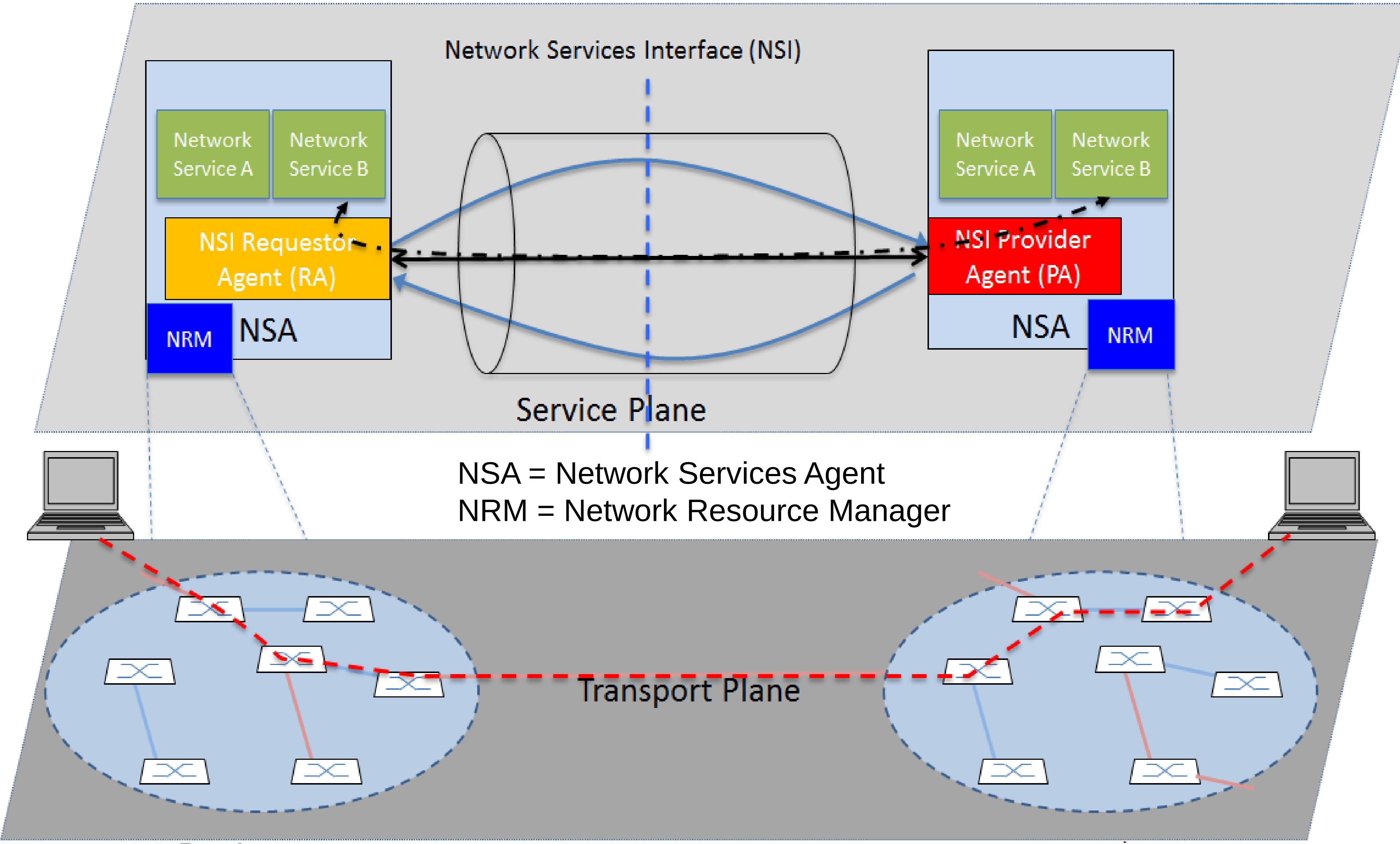
➢ Not really “end-to-end”, requires lots of manual configurations

# BoD: What we need is a Service !

Components include

- Authentication and Authorization
  - Global federated system that works well with applications
- Service Level Agreements
  - What is the lowest common denominator across the multi-domain network?
- Service Definition
  - Consistent view of the end-to-end service
  - Homogenous service over heterogeneous technologies
- Monitoring and measurement
  - End-to-end as well as along the network path between sites
- Multi-domain debugging
  - How do you find errors, report them so they can be debugged and fixed?

# Network Service Framework Concept



NSA = Network Services Agent
NRM = Network Resource Manager

# HEP Workflows need Agile Networking

- Requires traffic engineering for large flows and different science flows
- Custom security policies
- Constant network testing and monitoring
- ➢ Inherently multi domain

Today’s production services may be sufficient but need to evolve:

- Network-aware Workflow Engines and network-aware Data Placement
  - A. Klimentov: “Integrating Network Awareness in ATLAS Distributed Computing Using the ANSE Project”
  - T. Wildish: “Virtual Circuits in PhEDEx, an update from the ANSE project”
- Network Virtualization combines Hardware, S/W Network Resources into single Administrative Entity
- Scalability across a heterogeneous computing environment

# More Network Challenges: The Need for Cultural and Operational Change

(Findings from the Snowmass Community Planning Process in the US)

- In addition to basic network research, a number of important cultural and operational practices need to be changed:
  - Expectations for network performance need to be raised significantly, so that collaborations do not design workflows around a historical impression of what is possible.
    - Networking needs to be included into resource planning process, in addition to CPU and Storage, and determine how much/what is needed based on a comprehensive cost/benefit analysis
      - Possible consolidation of the worldwide distributed Facility
  - The large gap between peak and average transfer rates must be closed.
  - Campuses must deploy high performance Local Area Network Infrastructure, matching the capabilities of the Wide Area Network, and secure, science data enclaves – or Science DMZs – optimized for the needs of HEP.
  - Facility Storage System Performance must match Workflow I/O requirements

# Network Research and Innovation Agenda

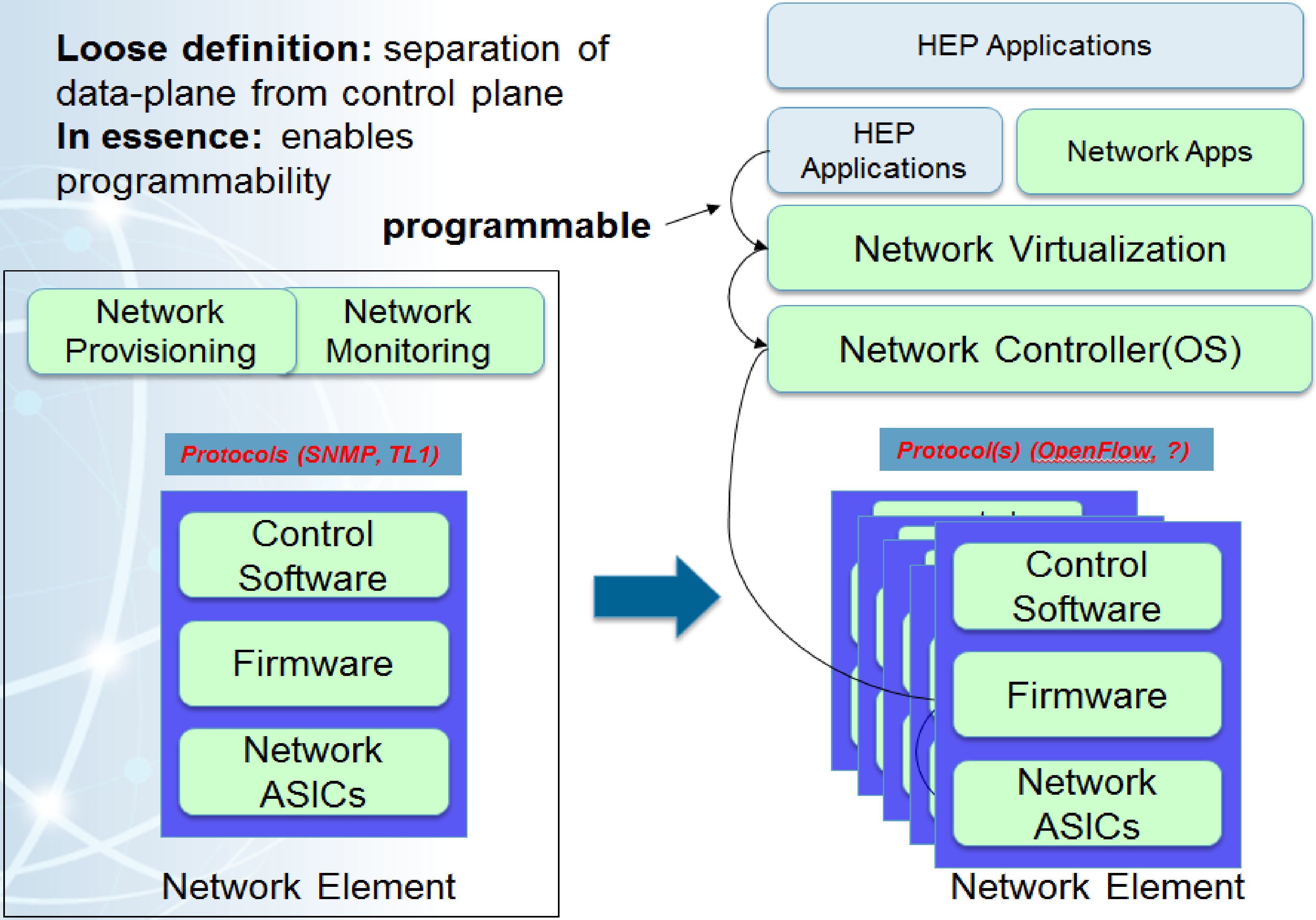
(Findings from the Snowmass Community Planning Process in the US)

- Evolving HEP Computing Models will heavily integrate communication, computing, and storage resources in order to support a wide range of discovery techniques and environments.
  - These will include HEP-specific science gateways and portals, cloud-based workflows, high-performance and high-throughput elements, and new data service capabilities.
- Research and innovation in many domains are necessary to support this evolution.
- Core question: can global research networks evolve into *adaptive, self-organizing, programmable systems* that quickly respond to requests of HEP science applications?
- Software Defined Networking is a promising research area
  - Closed, inflexible, proprietary hardware/software systems are re-imagined as open, programmable hardware/software components. Easier software evolution, plus potential cost savings through cheaper hardware.
  - – Software-defined networks have the potential to enable great innovation inside the network, and to benefit HEP by facilitating virtualization, programmability, integration.
  - – Analogous situation to VM - getting the service I want, no longer interfering
    - Predictability, > 80% that we get what we need

# What is SDN?

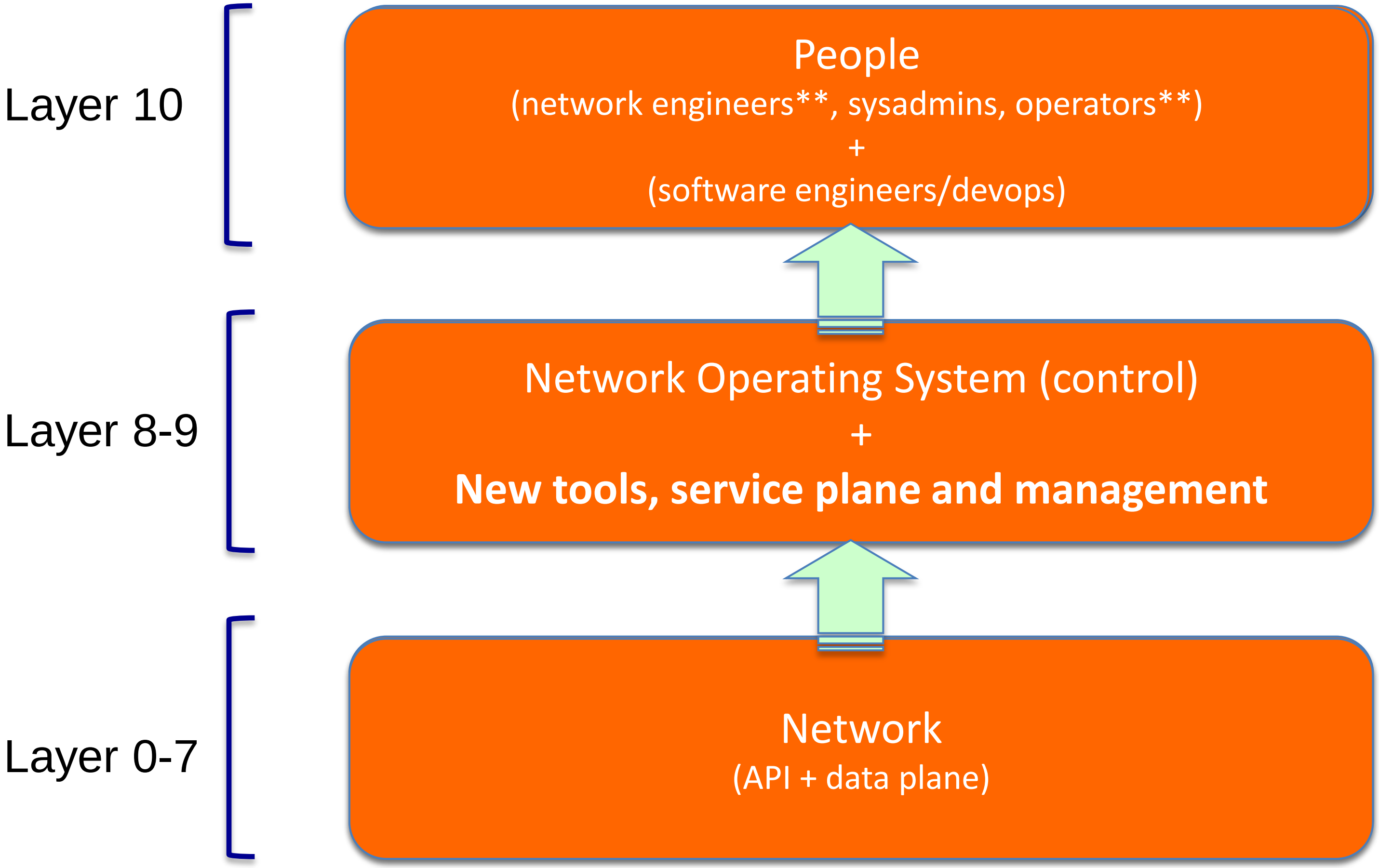
**Loose definition:** separation of data-plane from control plane

**In essence:** enables programmability



I. Monga @ CHEP2013

# SDN concepts come with People and Software Challenges



Layer 10

People
(network engineers**, sysadmins, operators**)
+
(software engineers/devops)

Layer 8-9

Network Operating System (control)
+
**New tools, service plane and management**

Layer 0-7

Network
(API + data plane)

I. Monga @ CHEP2013

# Programmability will lead to greater predictability

- HEP increasingly needs to deal with high performance, any-to-any bursts of data
  - E.g. ALICE, ATLAS (Event Streaming Service), CMS, LHCb
- Virtualization simplifies how HEP applications could program the network
  - The complexity is absorbed by ‘software hypervisor’ of the underlying multi-domain network
- SDN enables
  - Multi-layer control – packet and optical layer
  - Control over individual flows – e.g. route science flows around packet bottlenecks
  - **An opportunistic way to leverage all bandwidth without extra investment**
- Many NRENs have access to fiber, optical and packet platforms.

# Network Agility: Network Policies

An SDN controller does not do anything on its own: software implements the network behavior expected by the applications.

**Definition:**

***A network policy is a software or service that listens to events describing changes in the state of the network, and may decide on actions to be performed based on a pre-defined set of rules.***

# The issue of Coherence in SDN Environments (the “Multiple Writer Problem”)

- Currently there is no way to insure cooperation among independently developed SDN applications/services
- Asking S/W developers to understand all of the resource allocation in all the other applications is not scalable

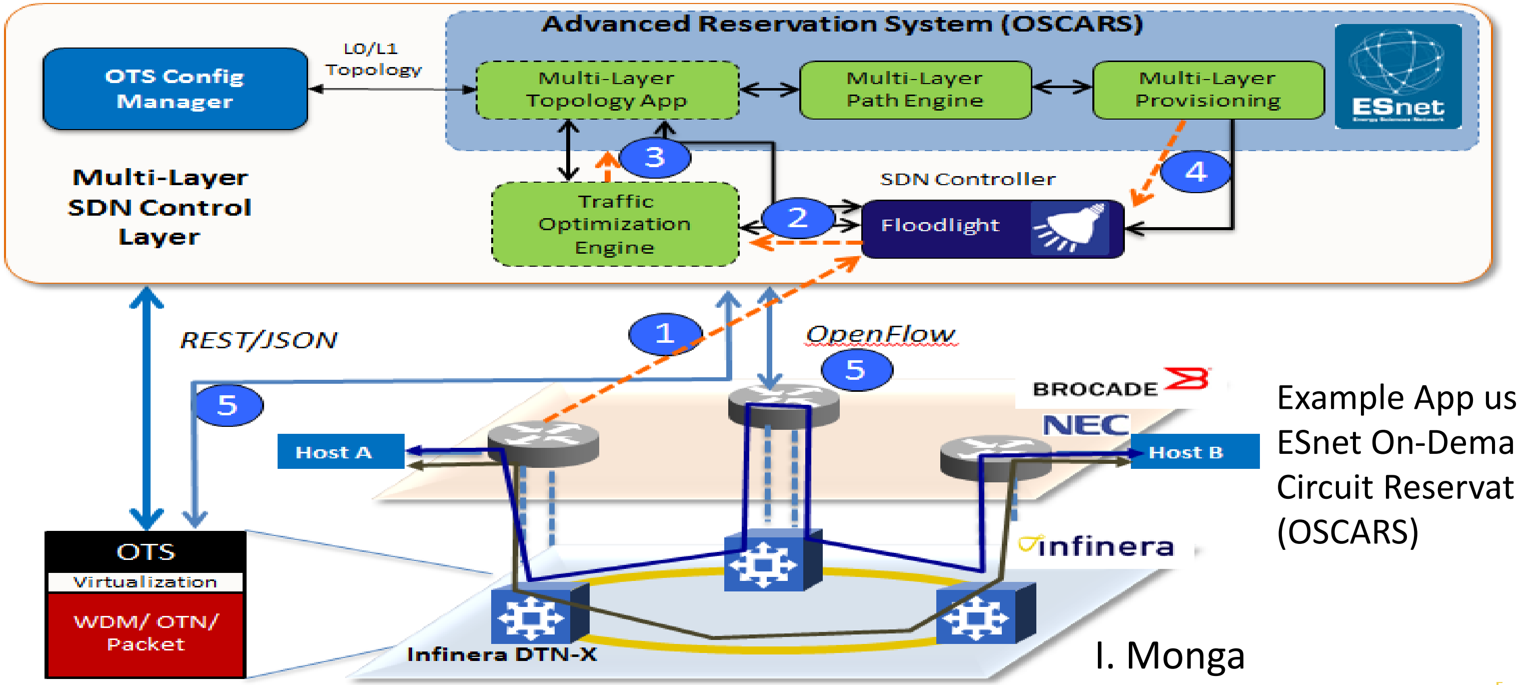
Eric Pouyoul (ESnet):

*“After a few years “playing” with SDN concepts and writing both production and prototype software, I started to wonder how all of this would fit together in a coherent system, and few questions came to mind:*

- *How to validate the correctness of complex, large, software before deploying it in production ?*
- *How will I debug the code ?”*

# Where are we going with "Intelligent Networking" ?

*Q: "Is it the right approach for HEP to deal with the details associated with dynamic network management services at the circuit and SDN level?"*



Example App using SDN: ESnet On-Demand Circuit Reservation System (OSCARS)

I. Monga

*"Wouldn't it be more appropriate for us to let the "Network" know about our needs, expressed as an "Intent" rather than using a prescriptive approach that requires us to know and digest lots of internal information we don't want/need to know about?"*

# Intent versus Prescription

## Intent

- What I want, not how to do it
- Portable, independent of protocol, vendor, media, etc.
- “I want my hunger to stop”
- Jane wants to communicate with the server at BNL
- “Please fix my fridge”

## Prescription

- How to do it (commands, rules, settings)
- Non-portable, dependent on protocol, vendor, media, etc.
- “Give me food”
- Send packets matching this 5-tuple out on port 24
- Request & schedule technician, analyze problem, identify and get spare part(s), replace part(s) ...

Example: DDM to Network: “I want to transfer 25 TB of data from MWT2 to BNL between 8 am and 4 pm”

- Intent is composed of “object” (xx amount of data) and “operation” (Transfer) w/ “Condition” (between 8 am and 4 pm)

Intent-based Networking is an activity within the larger SDN community, with Group-Based Policies in OpenStack, Network Intent Composition (NIC) in ODL & project KEYSTONE in OpenSourceSDN

# Intent – Some fundamentals

Thesis is that anything that can be conveyed to the controller via prescriptions can instead be described as Intent

- Intent is invariant and doesn't change as a result of
  - Link, switch, router, server, storage fault
  - Changing network providers, equipment manufacturers, protocol, devices
- Infrastructure is complex & complicated and it needs to be configured
  - "Intent" API completely abstracted to aid usability
- Intent is portable across implementation choices
  - Heterogeneous solutions w/o multiple investments in infrastructure-specific integration
- Disparate SDN services can be combined arbitrarily within an SDN domain
  - Intent is common language & only interface to SDN box
  - Rendering system understands how to translate intent into resource allocation, detecting/resolving conflicts

# Networking in a Virtualized Computing Environment

- Virtualization and mobility are driving what we call "Cloud Networking"
  - Different than enterprise networking
  - Needs to be built of a strong S/W foundation
    - Enterprise is about connectivity, cloud & SDN is about optimizing applications
  - No more one physical server w/ one network connection
  - Thousands of VMs instead
  - Storage technology transitioning to Object Stores (?)
  - Core count increasing – millions of cores in future

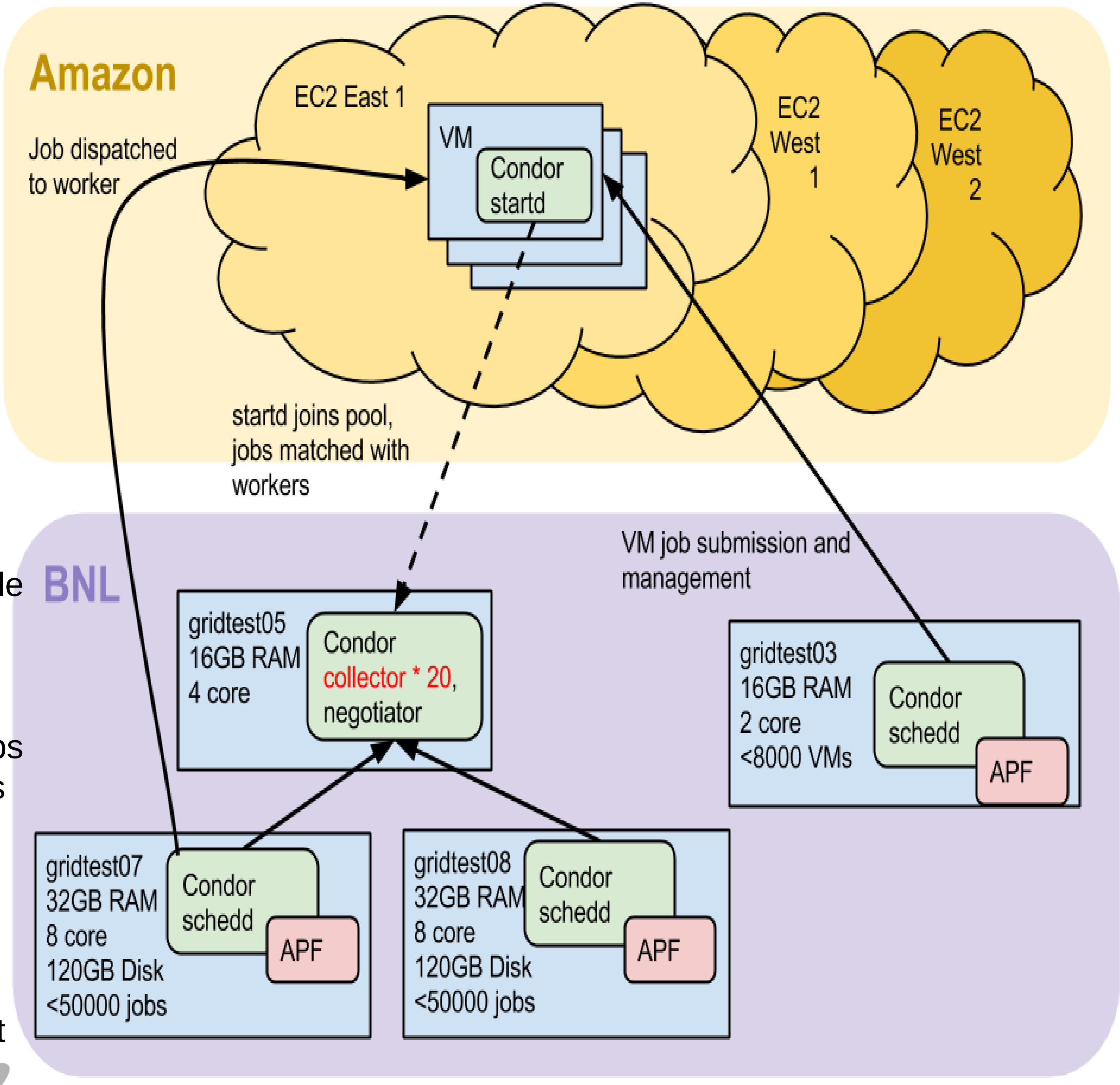
➢ With such change, network & I/O likely becomes the bottleneck

# Cloud Provisioning

Example: AutoPyFactory

The deployed hardware is expected to scale to 100k concurrent jobs

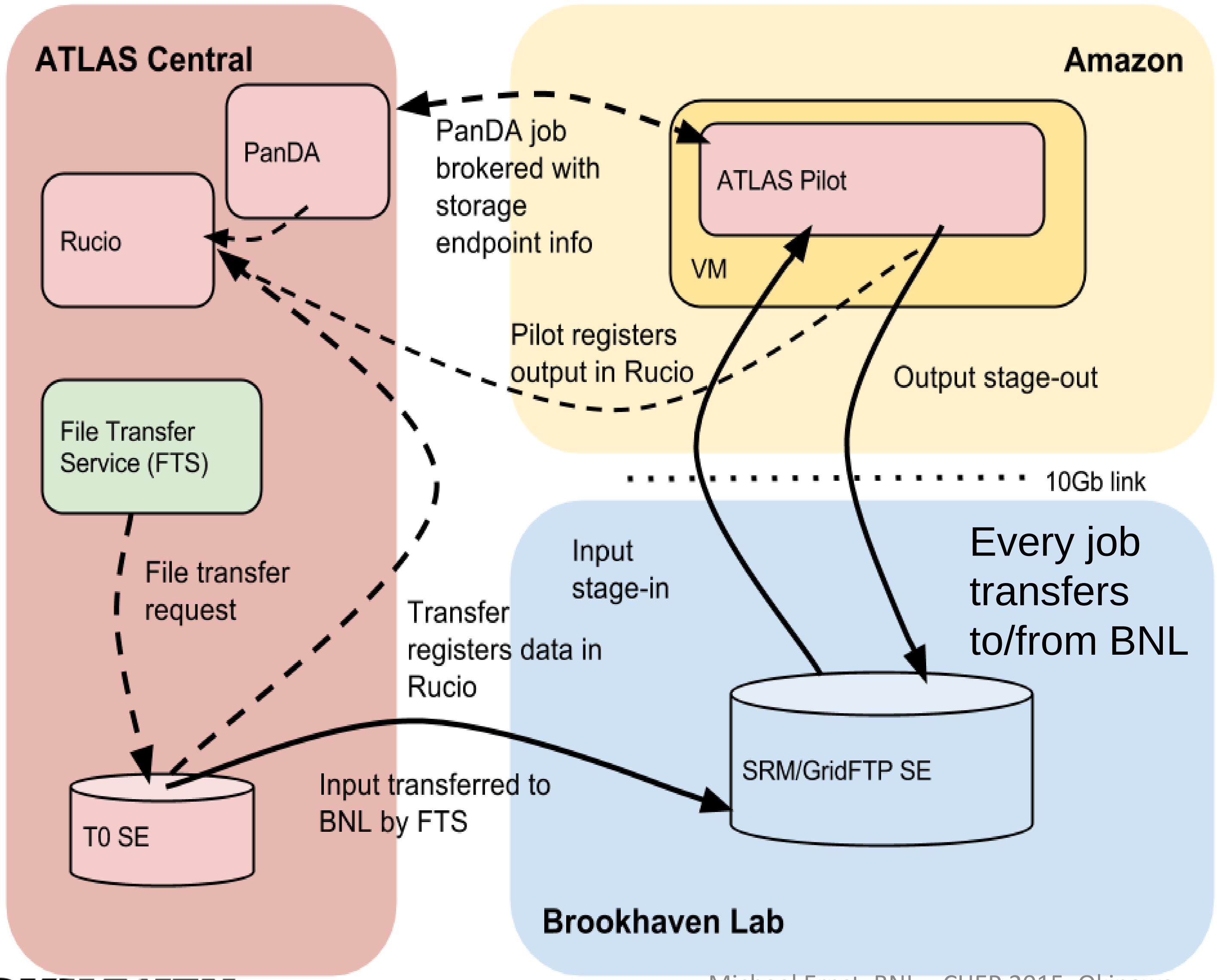
- Experience with ~30k jobs
- Setup serves serial and multi-core queues
- Policy-based VM lifecycle management



Amazon
EC2 East 1
EC2 West 1
EC2 West 2
Job dispatched to worker
VM
Condor startd
startd joins pool, jobs matched with workers
VM job submission and management
BNL
gridtest05 16GB RAM 4 core
Condor collector * 20, negotiator
gridtest03 16GB RAM 2 core <8000 VMs
Condor schedd
APF
gridtest07 32GB RAM 8 core 120GB Disk <50000 jobs
Condor schedd
APF
gridtest08 32GB RAM 8 core 120GB Disk <50000 jobs
Condor schedd
APF
John Hover, Brookhaven National Laboratory

# Utilize Cloud for Compute

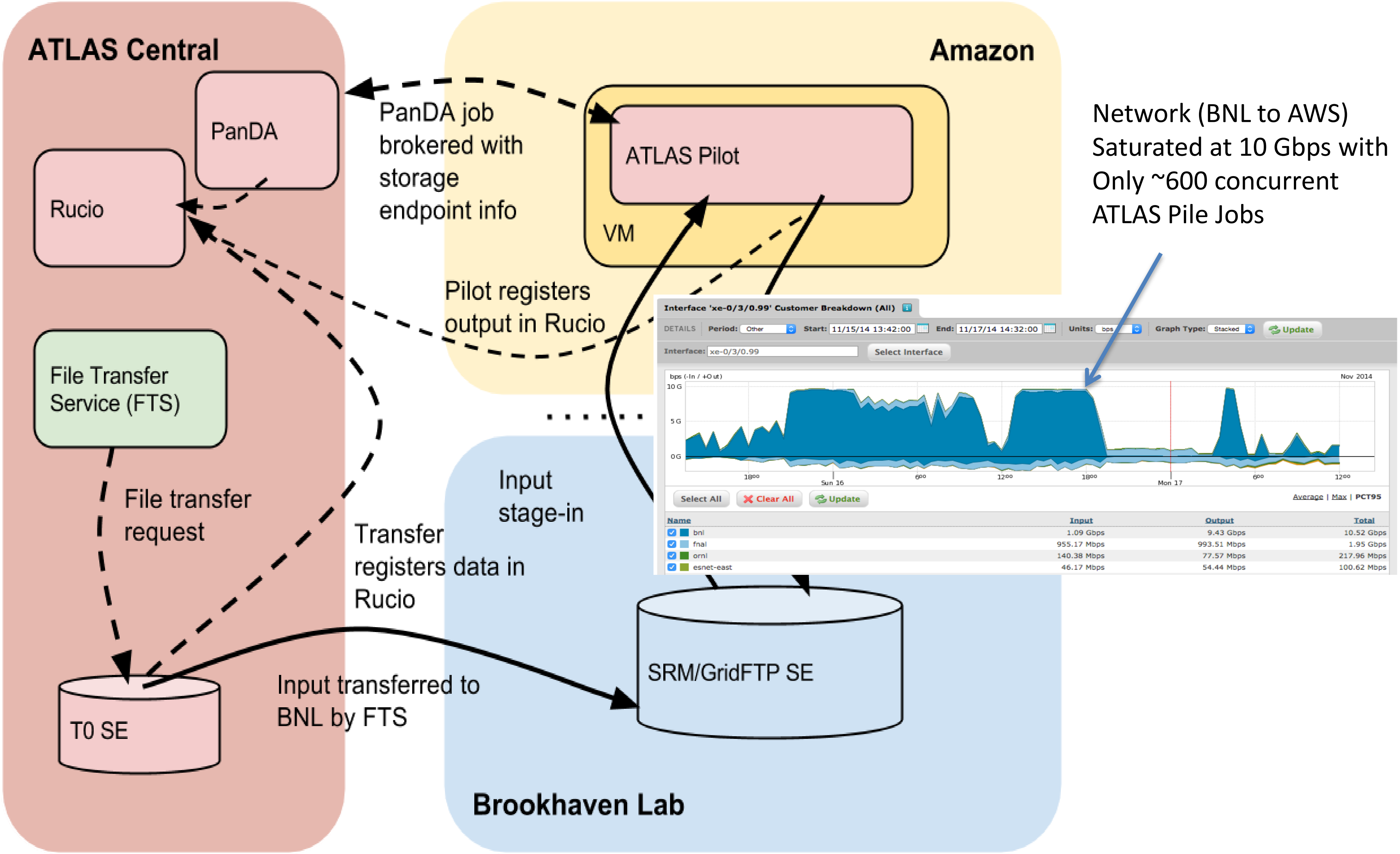
… the initial, easy step to using the "Cloud"


ATLAS Central
PanDA
Rucio
File Transfer Service (FTS)
File transfer request
T0 SE
PanDA job brokered with storage endpoint info
Amazon
ATLAS Pilot
VM
Pilot registers output in Rucio
Output stage-out
10Gb link
Input stage-in
Transfer registers data in Rucio
SRM/GridFTP SE
Input transferred to BNL by FTS
Brookhaven Lab

Every job transfers to/from BNL

# Utilize Cloud for Compute

## … the initial, easy step to using the “Cloud”


ATLAS Central
PanDA
Rucio
File Transfer Service (FTS)
T0 SE
Amazon
ATLAS Pilot
VM
Brookhaven Lab
SRM/GridFTP SE
PanDA job brokered with storage endpoint info
Pilot registers output in Rucio
File transfer request
Transfer registers data in Rucio
Input stage-in
Input transferred to BNL by FTS

Interface 'xe-0/3/0.99' Customer Breakdown (All)

DETAILS Period: Other Start: 11/15/14 13:42:00 End: 11/17/14 14:32:00 Units: bps Graph Type: Stacked Update

Interface: xe-0/3/0.99 Select Interface

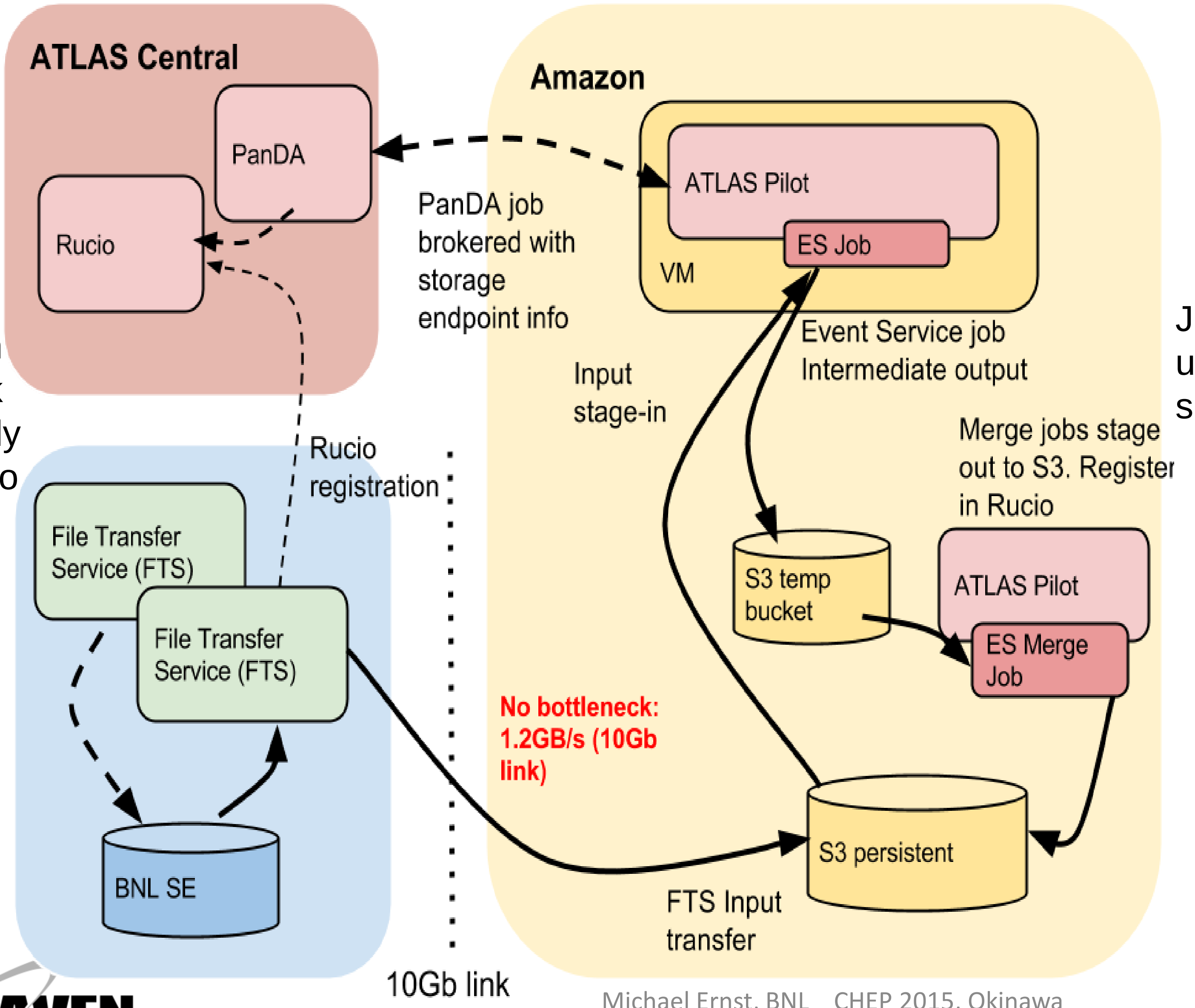
Select All ✗ Clear All Update    Average | Max | PCT95

| Name | Input | Output | Total |
|---|---|---|---|
| ☑ bnl | 1.09 Gbps | 9.43 Gbps | 10.52 Gbps |
| ☑ fnal | 955.17 Mbps | 993.51 Mbps | 1.95 Gbps |
| ☑ ornl | 140.38 Mbps | 77.57 Mbps | 217.96 Mbps |
| ☑ esnet-east | 46.17 Mbps | 54.44 Mbps | 100.62 Mbps |

Network (BNL to AWS)
Saturated at 10 Gbps with
Only ~600 concurrent
ATLAS Pile Jobs

# Using the Cloud for Compute and Storage

Long-term goal: Run entirely within EC2, with link limitations only affecting SE to SE transfers. "Site" stage-in/out via S3.

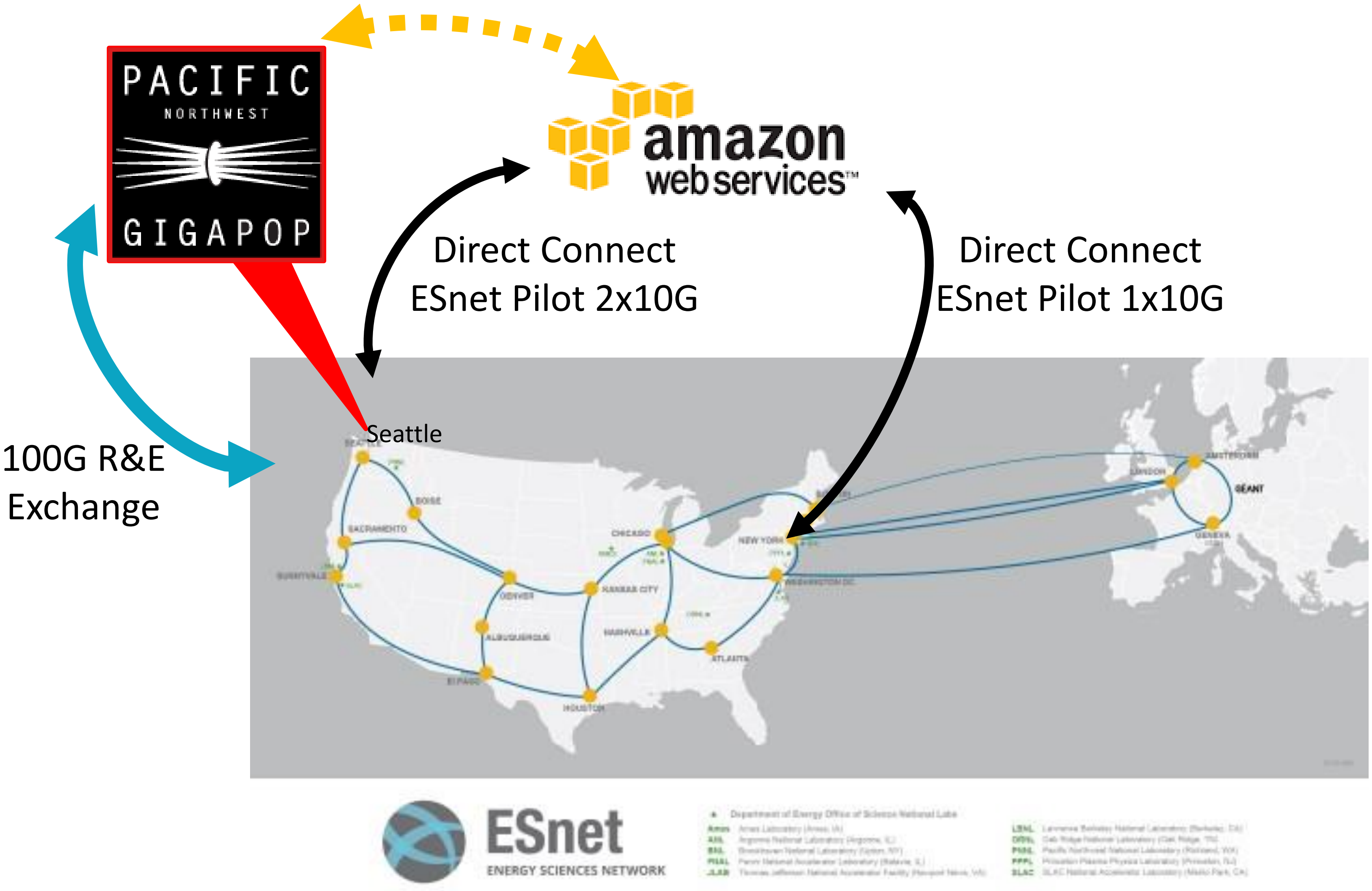

ATLAS Central
PanDA
Rucio
Amazon
ATLAS Pilot
ES Job
VM
PanDA job brokered with storage endpoint info
Event Service job Intermediate output
Input stage-in
Merge jobs stage out to S3. Register in Rucio
Rucio registration
File Transfer Service (FTS)
File Transfer Service (FTS)
S3 temp bucket
ATLAS Pilot
ES Merge Job
No bottleneck: 1.2GB/s (10Gb link)
BNL SE
S3 persistent
FTS Input transfer
10Gb link

Jobs only using S3, no scaling issues

# Example: AWS/NREN Networking

- AWS peering with ESnet (soon Internet2 and GEANT) to allow data flows between experiment dedicated storage and Amazon
  - Peering in Ashburn VA (10Gb), and Seattle WA and Sunnyvale CA (100Gb Test)
  - **AWS DirectConnect** to BNL via ESNet (10Gb)
- DirectConnect is the router/advertising and flow configuration; enables
  - QoS/congestion control
  - Virtual Private Clouds for custom topology (if needed).
  - Public internet IP addresses with host in EC2.

# Connecting AWS Facilities to the Research Community



AWS Planned 100G to PNWG

Direct Connect ESnet Pilot 2x10G

Direct Connect ESnet Pilot 1x10G

100G R&E Exchange

# The Network as a Partner in Cloud Computing

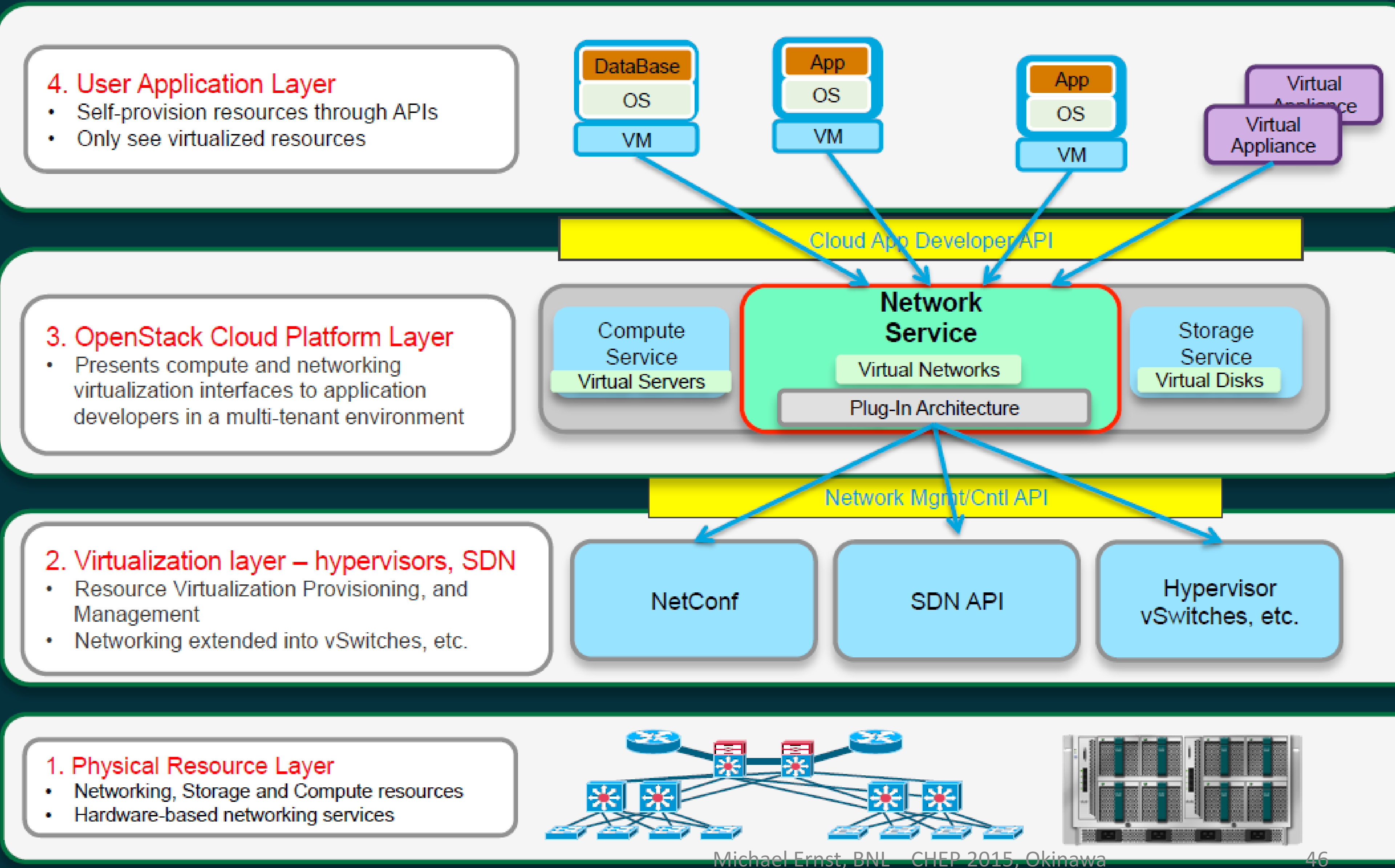
- Amazon’s “Elastic IP” is an application-driven approach to integrating the Network and the Cloud
    - Automation
        - Rich collection of network-related service primitives
        - Data Center Level combined with Wide Are Networking
        - Network virtualization critical to automate end-2-end system connectivity
            - Automation only way to save on operational effort
- OpenStack includes Network services as one resource it virtualizes as with CPU/server and Storage
    - OpenStack’s Neutron interface defines how a virtual network can be created to “host” CPU and other elements
        - Neutron does not define the technology to create virtual networks
        - Cloud provider is responsible for mapping their technology to virtual network models

# OpenStack Neutron

- Since Folsom release OpenStack users have OpenStack Neutron, an industry-standard, open API for Cloud Networking orchestration
  - Providing abstraction layer
  - Decouples operations from network mechanics
  - OpenStack had API for Compute Module called "Nova"
    - Networking was hardcoded
    - Neutron makes Networking Subsystem pluggable, modular, standard interface to run any number of virtual networking solutions
    - Neutron allows to create virtual Networks in a vendor-independent way, attach those networks to VMs and orchestrate configuration of services

# SDN and Cloud Computing

## OpenStack is a platform for application developers to automate provisioning of compute, storage, and networking virtual resources



### 4. User Application Layer

- Self-provision resources through APIs
- Only see virtualized resources

### 3. OpenStack Cloud Platform Layer

- Presents compute and networking virtualization interfaces to application developers in a multi-tenant environment

### 2. Virtualization layer – hypervisors, SDN

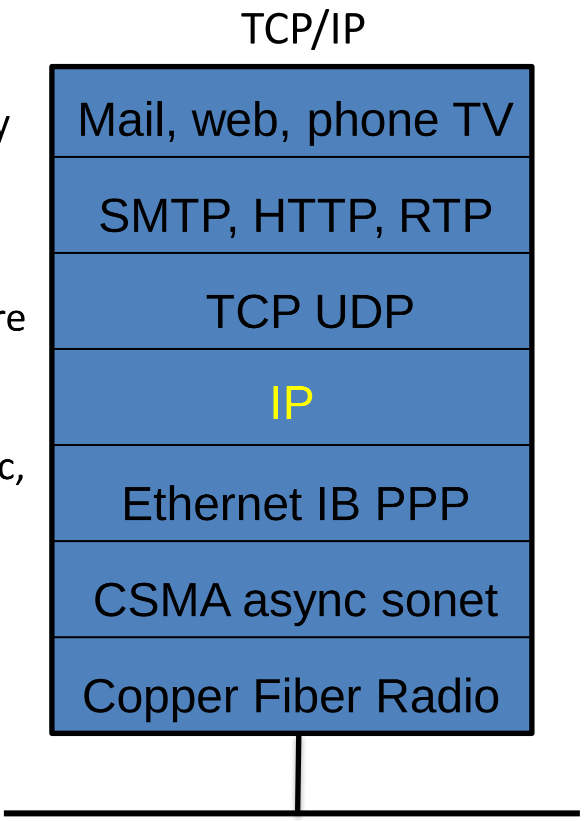
- Resource Virtualization Provisioning, and Management
- Networking extended into vSwitches, etc.

### 1. Physical Resource Layer

- Networking, Storage and Compute resources
- Hardware-based networking services

# Data Management in the Network

- Best effort IP packet delivery to destination IP addresses
- The anchor of the architecture is IP address space
- One-way traffic, stateless, no storage



TCP/IP

Mail, web, phone TV

SMTP, HTTP, RTP

TCP UDP

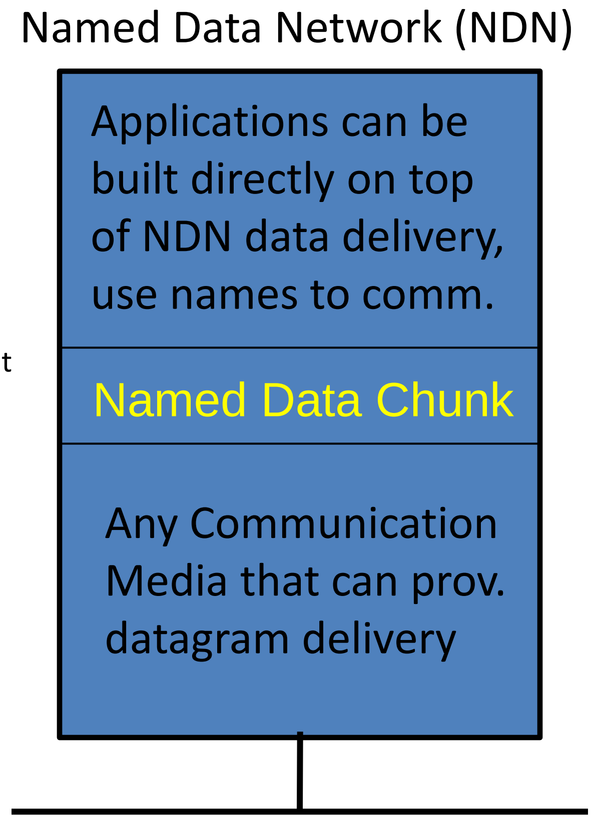
IP

Ethernet IB PPP

CSMA async sonet

Copper Fiber Radio

- Names are generated by applications,
- opaque to the network
- Packet granularity
- Hierarchical
  - identify content relationship & facilitate aggregation
- Every data packet carries a signature, binding the name to the content (security)



Named Data Network (NDN)

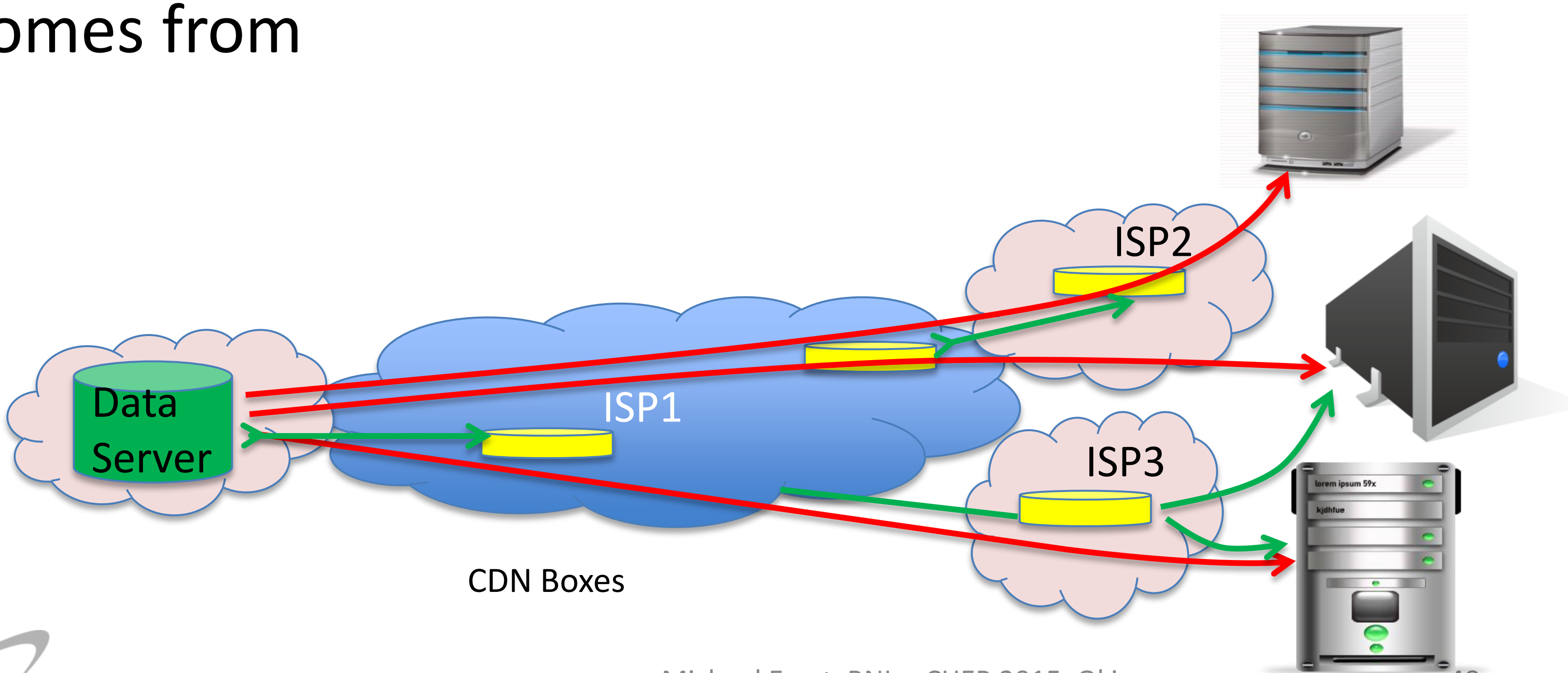
Applications can be built directly on top of NDN data delivery, use names to comm.

Named Data Chunk

Any Communication Media that can prov. datagram delivery

# Example: IP-based Content Delivery

- Applications request data by names, network name packets by IP address
- IP delivers data between two end points
  - Multiple users may request the same data, content delivery network optimizes where data comes from



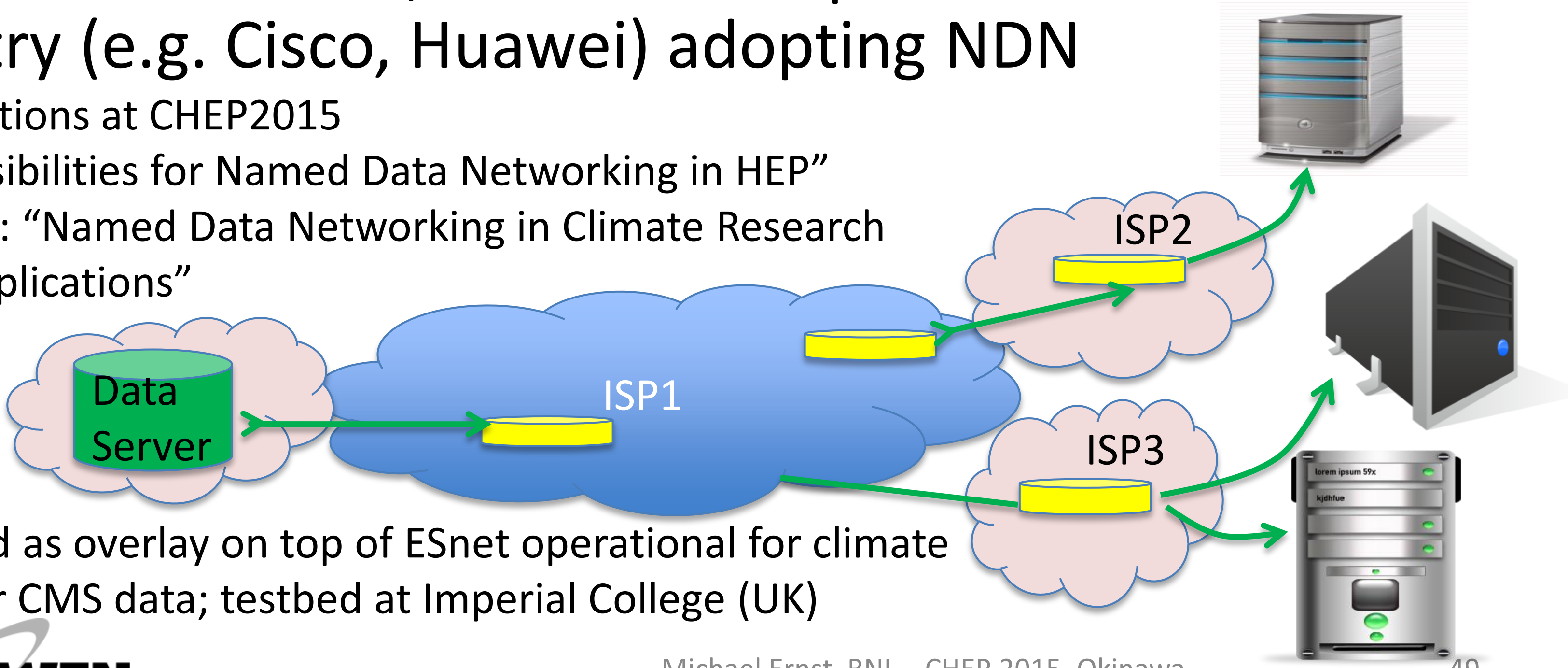
# Example: Content Delivery with Named Data Network (NDN)

- Network uses application data names for delivery
  - Multiple users request the same data: network can retrieve from nearby copy
  - Provides performance estimates (user provided metrics)
- Name + data-signature enables in-network storage (sec.)
- Caching happens automatically
- Broadcast “interest”, location-independent data retrieval
- Industry (e.g. Cisco, Huawei) adopting NDN

NDN presentations at CHEP2015

D. Rand: “Possibilities for Named Data Networking in HEP”

S. Shannigrahi: “Named Data Networking in Climate Research and HEP Applications”



NDN testbed as overlay on top of ESnet operational for climate and soon for CMS data; testbed at Imperial College (UK)

# Conclusions

- Regardless of the resource composition (distributed grid centers, consolidation within a few large data centers, consolidation within clouds, NDN) - high performance networking will continue to be critical to HEP
  - The network as a partner motivates why we should worry about networks
  - Regardless of Computing Models, HEP and network partners will need to work closely together to build the intent-based interfaces between applications and networks that can most effectively accelerate discovery
- Excellent networks, flexible and adaptable computing models and software systems are the foundation to fully exploiting resources such as Grids, Clouds and HPCs
  - Networks overcome limitations of geography
  - To optimize usage of excellent network infrastructure we have access to we need to interact with the control plane in an intelligent way
  - Network Virtualization - integration of storage, compute and network - in a seamless manner, including cloud and local resources. Leveraging efforts like OpenStack to instantiate VMs, allocate storage, and network dynamically
- Named-Data Networking – a new way of accessing content than worrying about where the data is located.

# Thank you!

- Many thanks to those who have helped with comments and whose materials I have drawn on
- Including but not limited to: J. Caballero, I. Fisk, R. Gardner, J. Hover, H. Ito, T. Javurek, S. McKee, R. Mount, C. Papadopoulos, V. Tsulaia, T. Wenaus, X. Zhao
- Special thanks to Gregory Bell and Inder Monga and the ESnet Team

" WE'D LOVE TO START A FAMILY, BUT WE'RE GOING TO WAIT UNTIL WE HAVE ENOUGH BANDWIDTH."

John Klossner