Software-Defined Data Plane Enhancing SDN and NFV

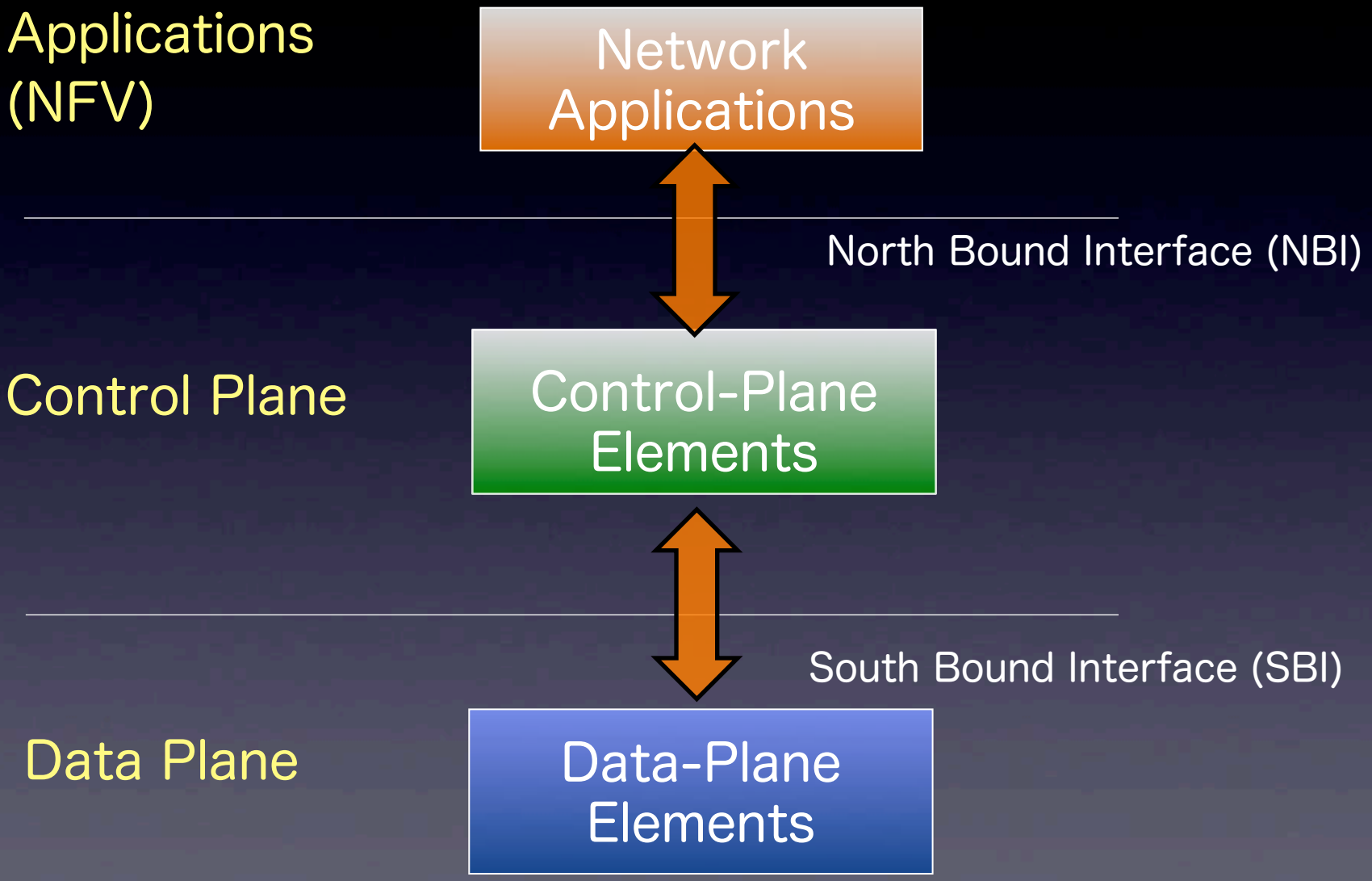
Aki Nakao

The University of Tokyo

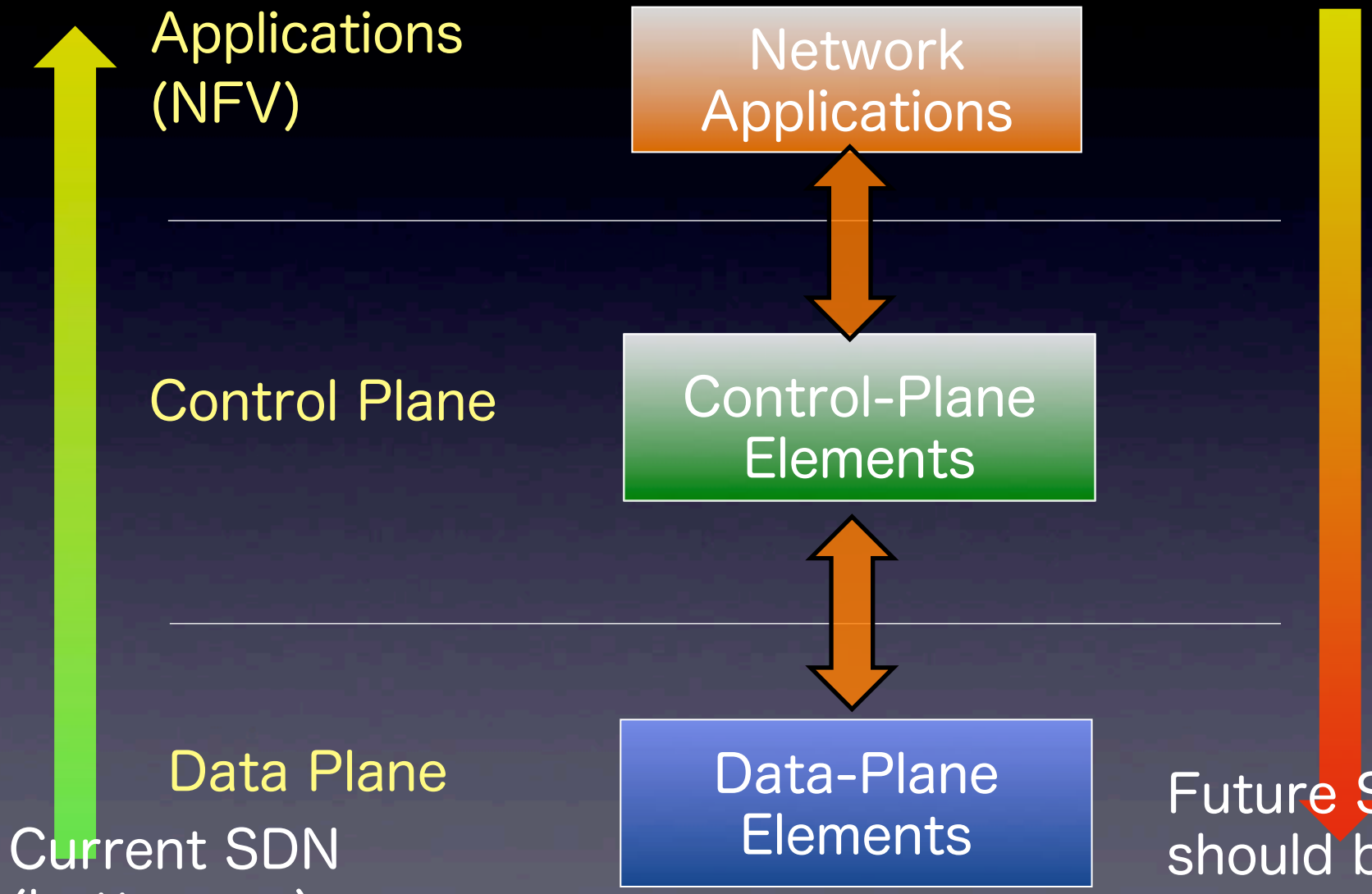
GSC 18

2014/7/22

SDN Architecture



Application Driven Thinking



Application-Driven Thinking

Premise：

Programmable networking has been enabled by SDN and NFV

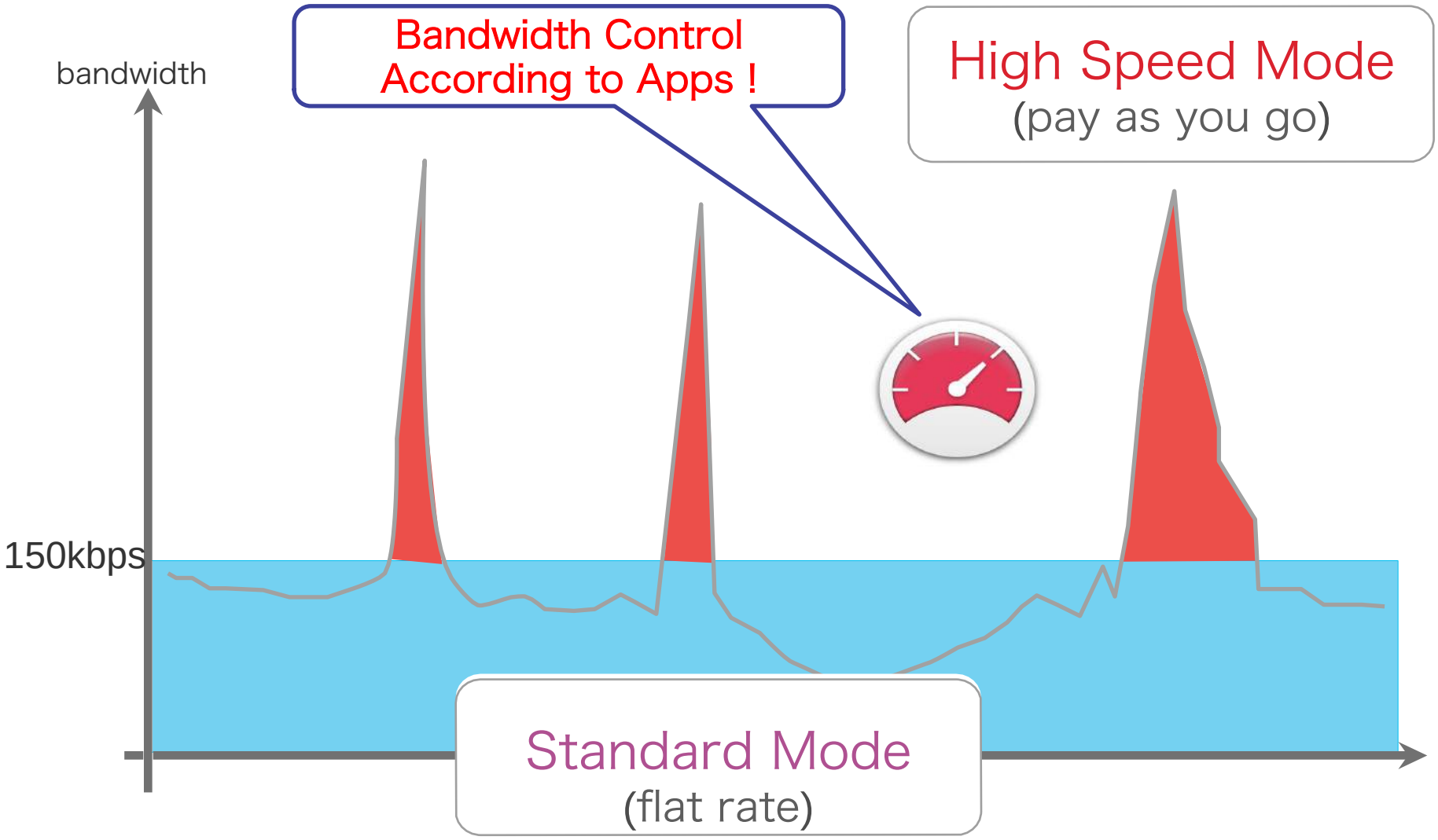
App-Driven Thinking:

- Think of “(killer) applications” first and then design network functions and interfaces APIs for SDN and NFV
- Not just OPEX/CAPEX reduction but create new values via SDN and NFV

Lets start with an example application!

“Next-Gen MVNO”

MVNO Customers Need More Flexible Subscription



App-specific traffic control enables more fined grained subscription plans that can get an MVNO out of the “low cost” competition

Application Driven SDN

There is a gap between application developers and SDN's programming interface

- The current SDN is targeted at network operators (researchers) and not at application developers
- Flow abstraction in Southbound Interface is for operators

  <Flow Pattern> <Action> <Stat>

- App/Device abstraction is useful and intuitive for application developers

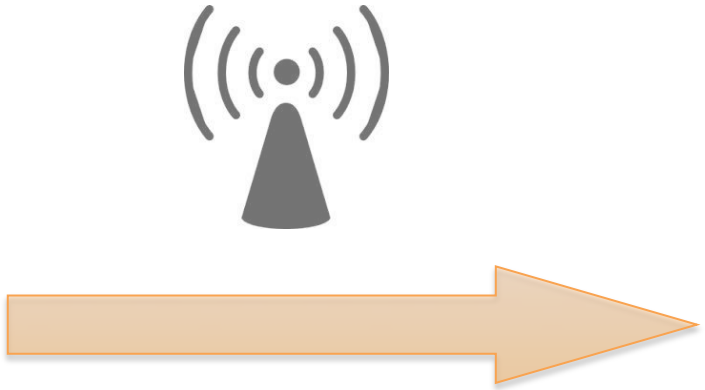
  <App/Device> <Action><Stat>

Solution: Application-Driven Software Defined Networking

GENI Engineering Conference 20 Demo:
App-Driven SDN QoS

Smartphone

App

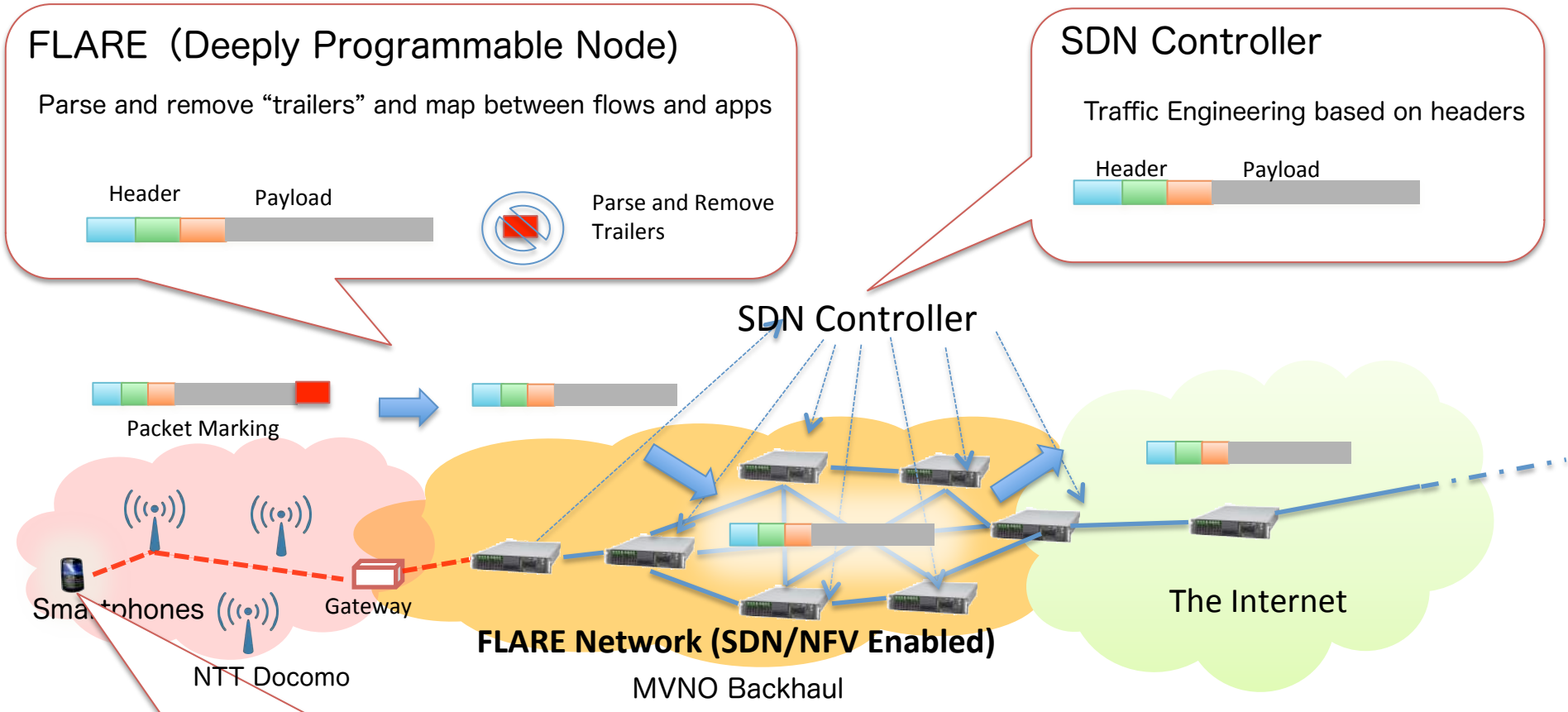


Packet

FLARE
(Data Plane Programmable)

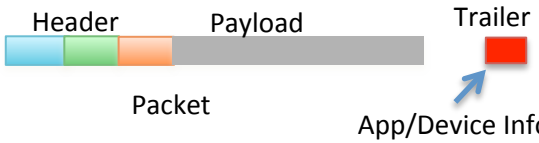
Chrome : Pass Thru
Firefox: Block
YouTube: Rate Limit

Our Proposal



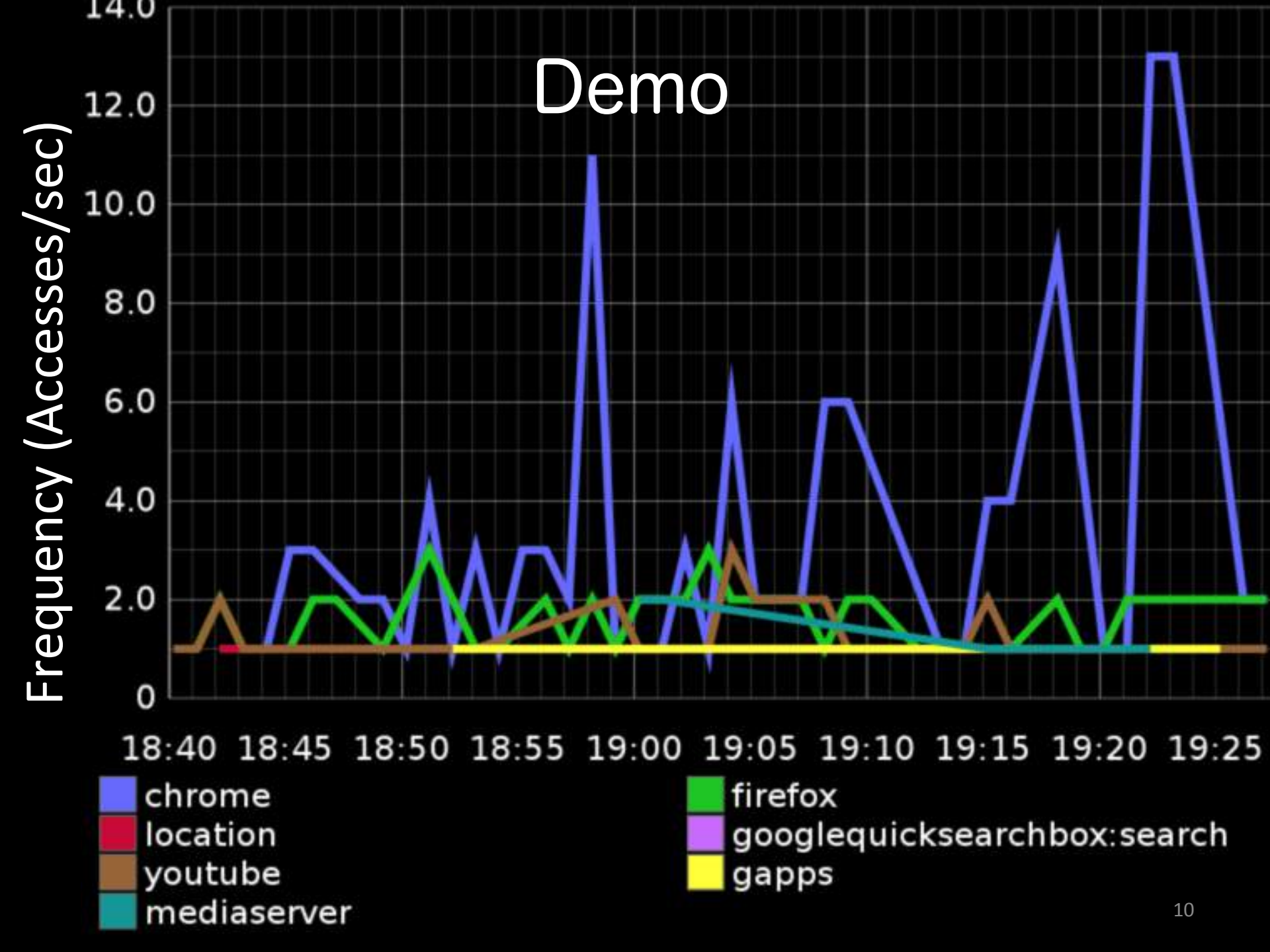
Smartphones (wearables)

Add app/device information to packet trailers

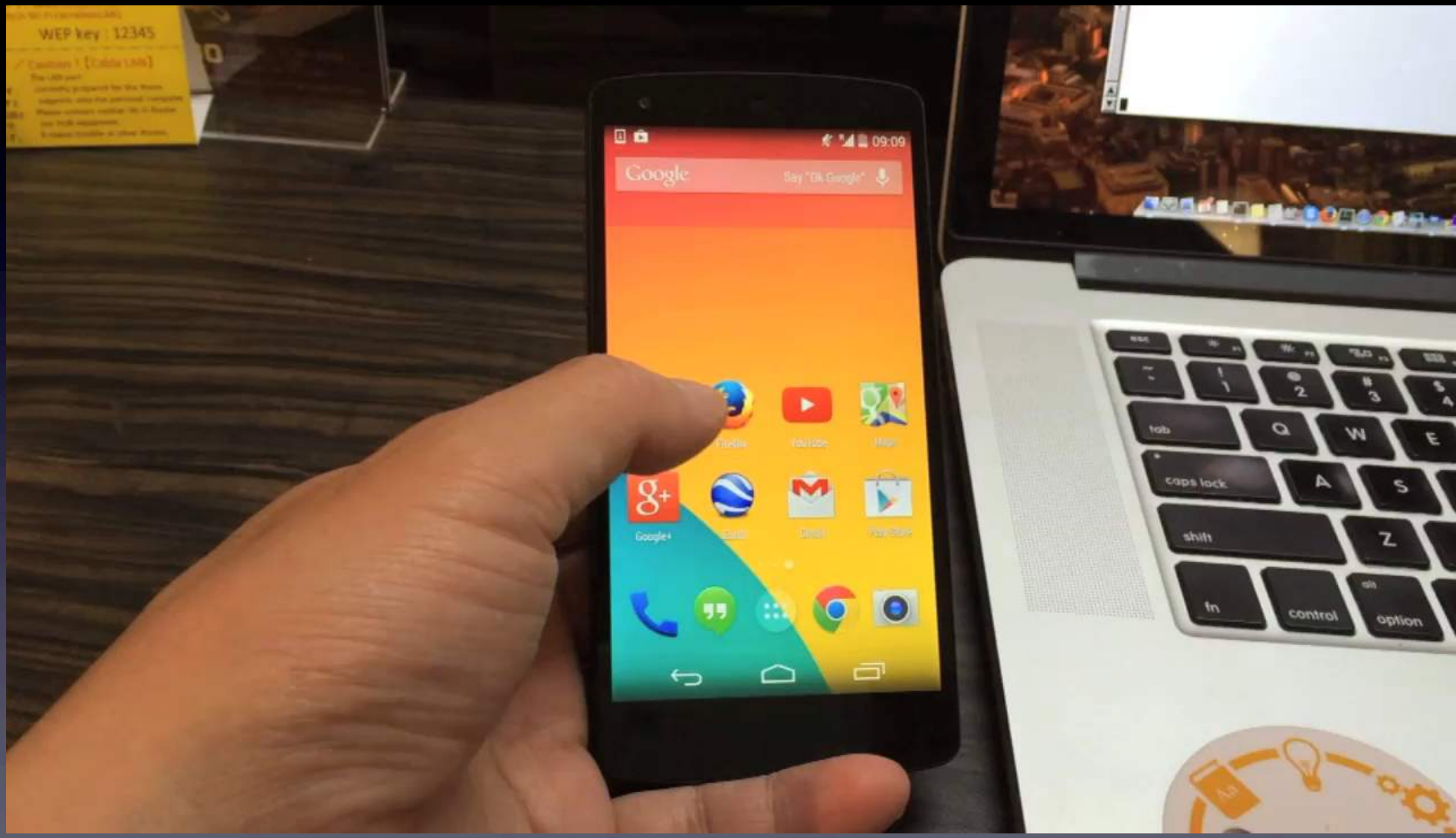


Smartphones attach app/device information to packets
FLARE detects app/device information and creates mapping between flows and apps/devices

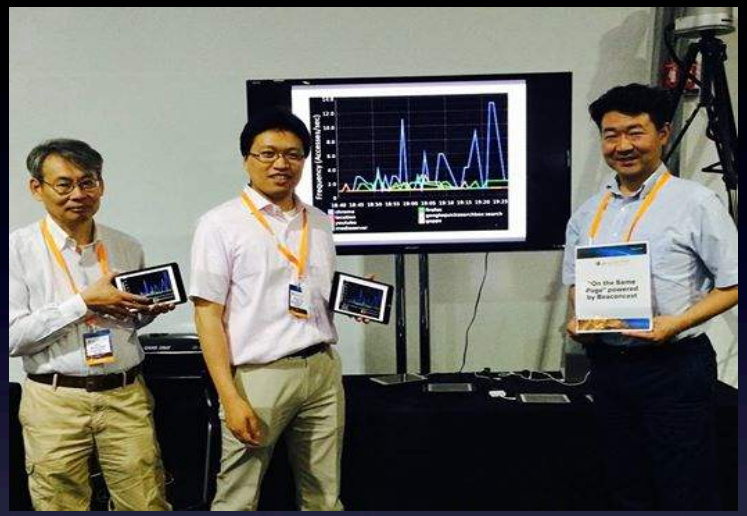
Demo


Frequency (Accesses/sec)
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0
18:40
18:45
18:50
18:55
19:00
19:05
19:10
19:15
19:20
19:25
chrome
location
youtube
mediaserver
firefox
googlequicksearchbox:search
gapps

Demo


WEP key : 12345
09:09
Google
Say "Ok Google"
Google+
tab
caps lock
shift
fn
control
option

We won the best demo award! GEC20@UC Davis



Benefits

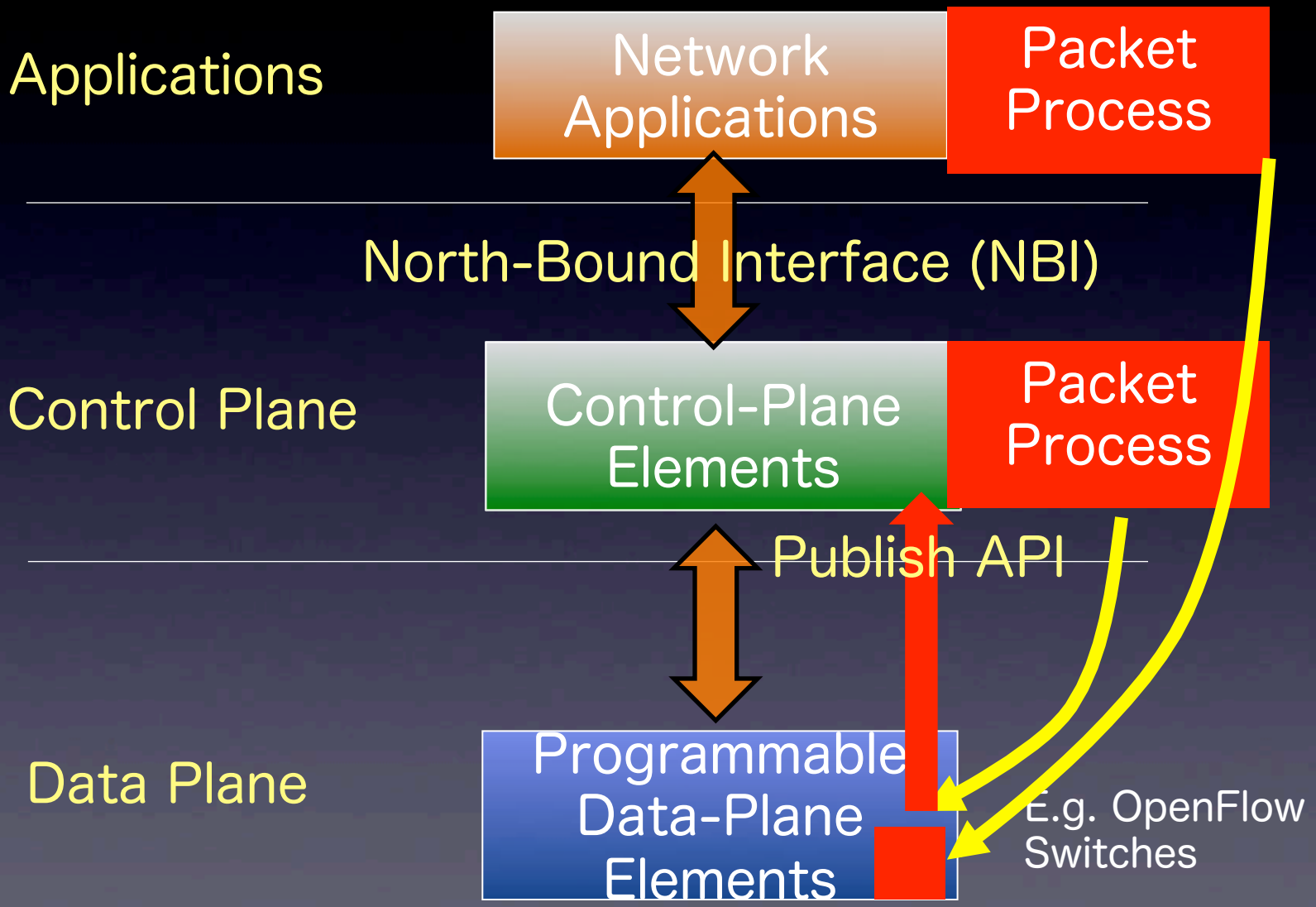
- Application Specific Traffic Engineering for MVNO
  - Application Name Based
  - Application Process Based (Fore/Background)
  - Device Type Based
  - Device State Based (Context / Location Aware)
- Parental Control
  - Not by apps on devices, but by networking
- Additional Value-Add services for specific applications
  - Differentiation for competing apps

    (e.g., Chrome vs. Firefox)

Contributions

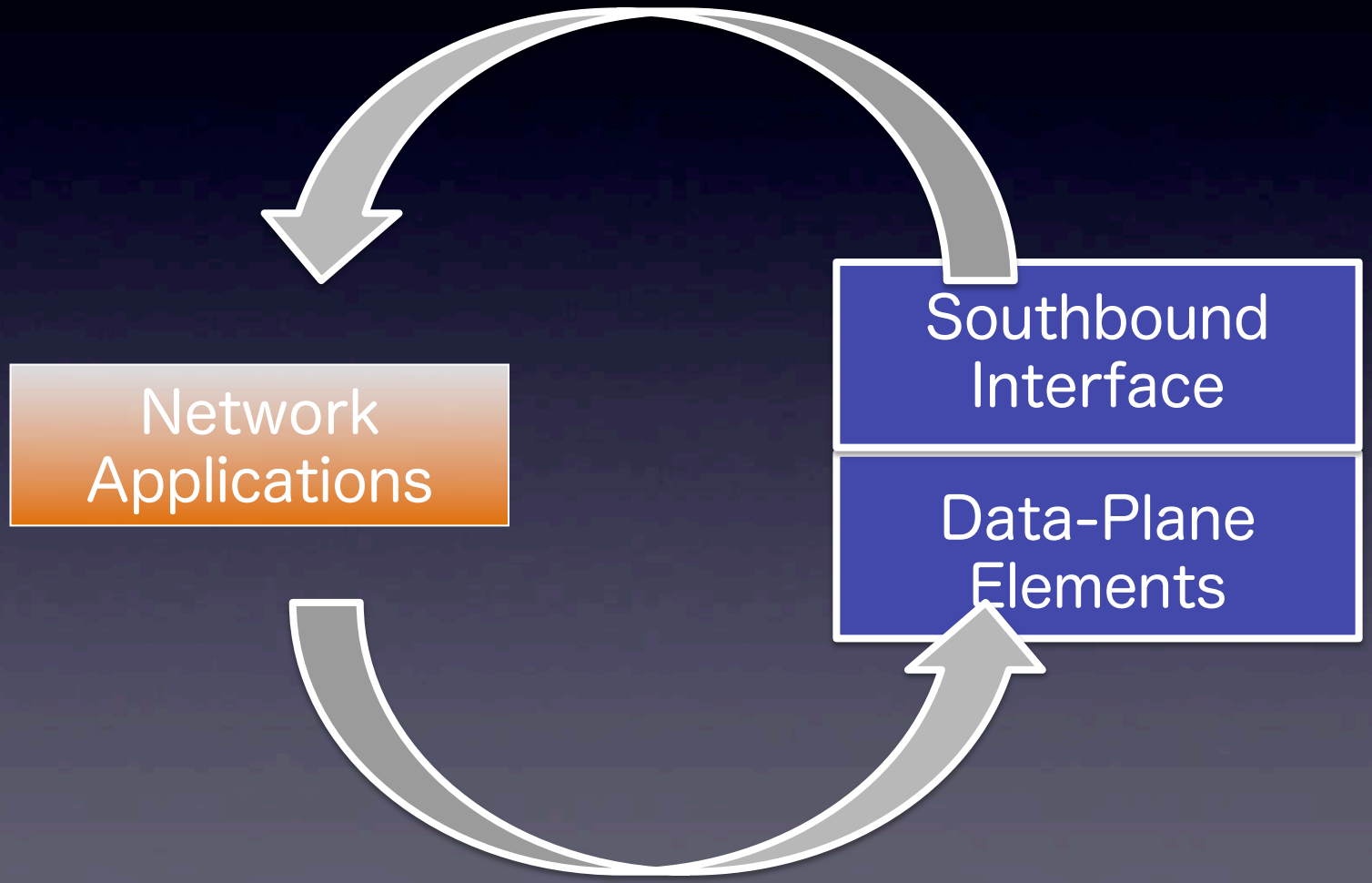
- Insert application information in trailers of packets (e.g., TCP SYN)
  - Extensible to other protocols than TCP
  - Similar to “Google Fast TCP Open” but for different purpose
  - More bits usable than TCP/IP options
- Determine applications with 100% Accuracy
  - Cooperation between end-systems and programmable nodes
- Extensible to supervised learning without app
  - Machine learning using sampled data with app

Software Defined Data Plane

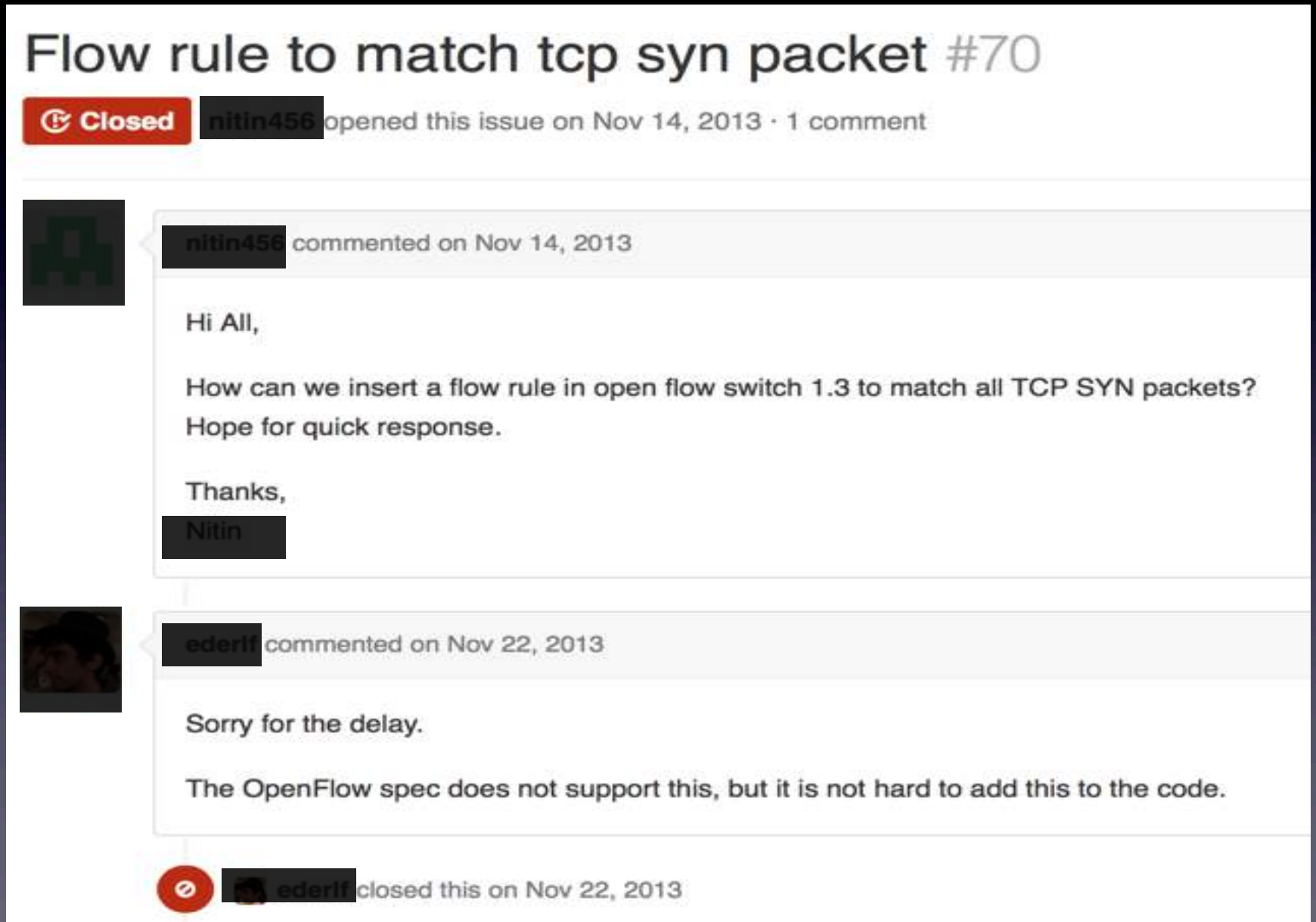


Accelerate SDN Standardization

Operation and Evaluation Feedback -> Accelerate Standardization



Example



Flow rule to match tcp syn packet #70

Closed [redacted] opened this issue on Nov 14, 2013 · 1 comment

[redacted] commented on Nov 14, 2013

Hi All,

How can we insert a flow rule in open flow switch 1.3 to match all TCP SYN packets?
Hope for quick response.

Thanks,
[redacted]

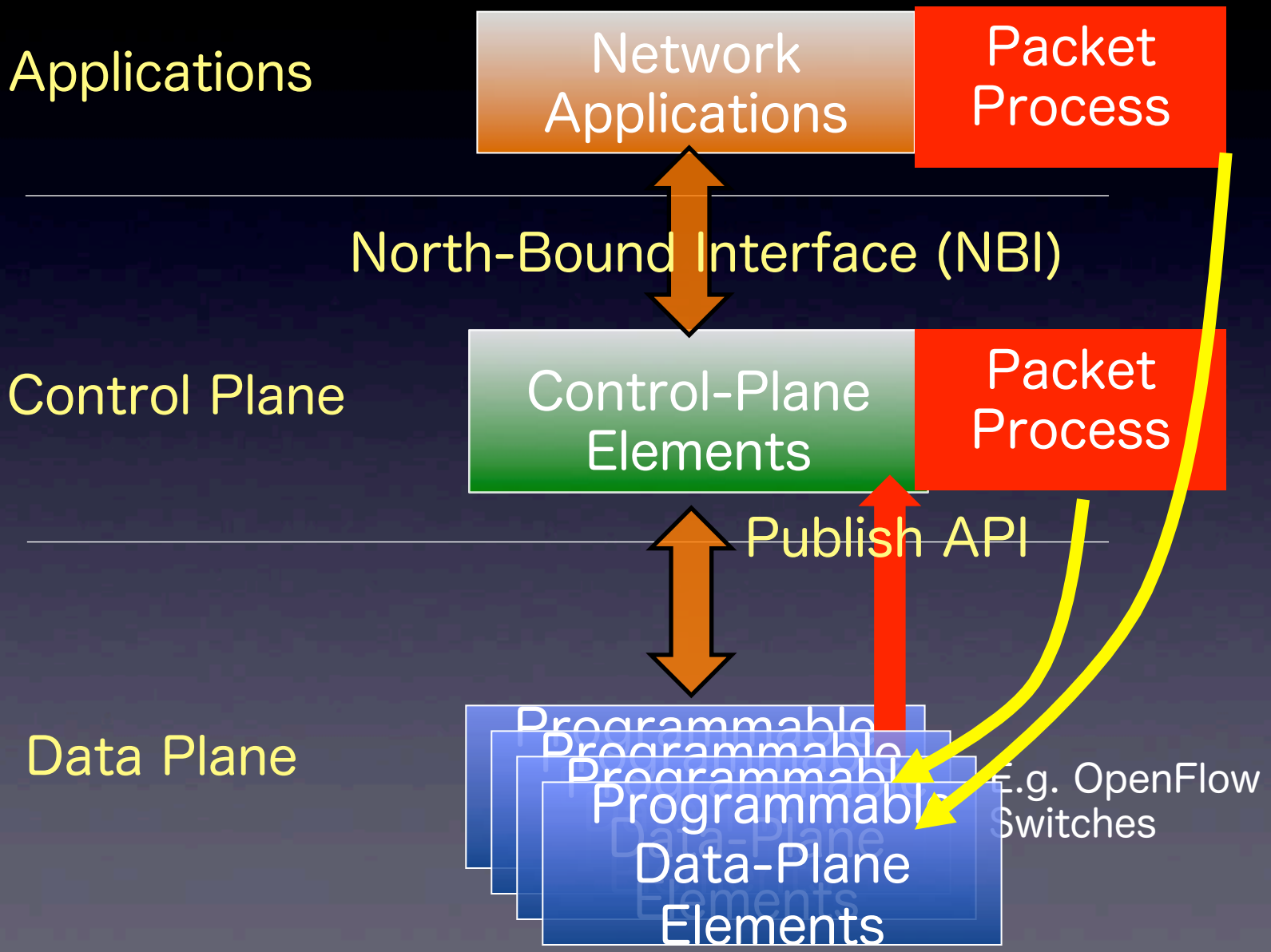
[redacted] commented on Nov 22, 2013

Sorry for the delay.

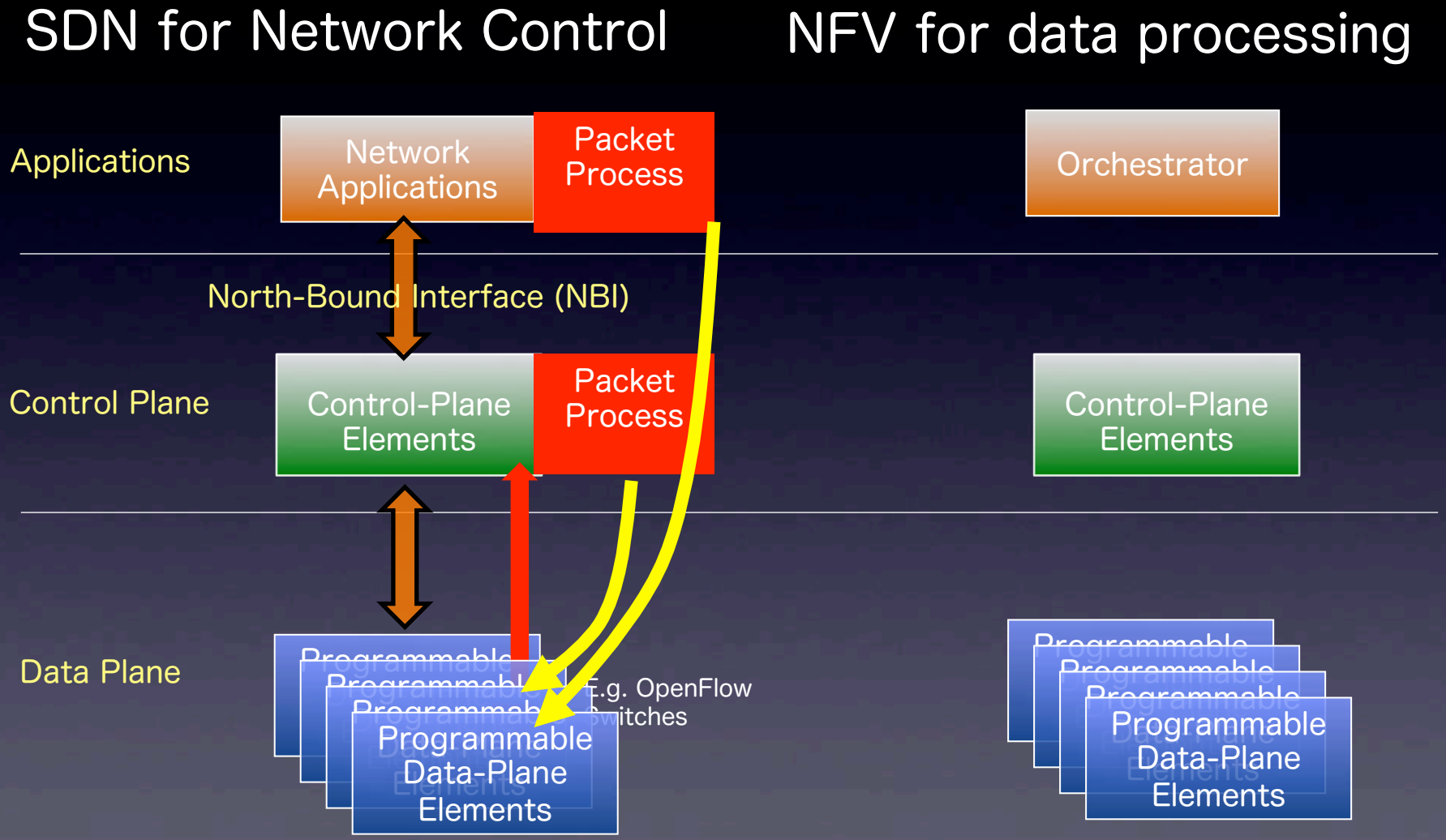
The OpenFlow spec does not support this, but it is not hard to add this to the code.

[redacted] closed this on Nov 22, 2013

Sliceable Software Defined Data Planes



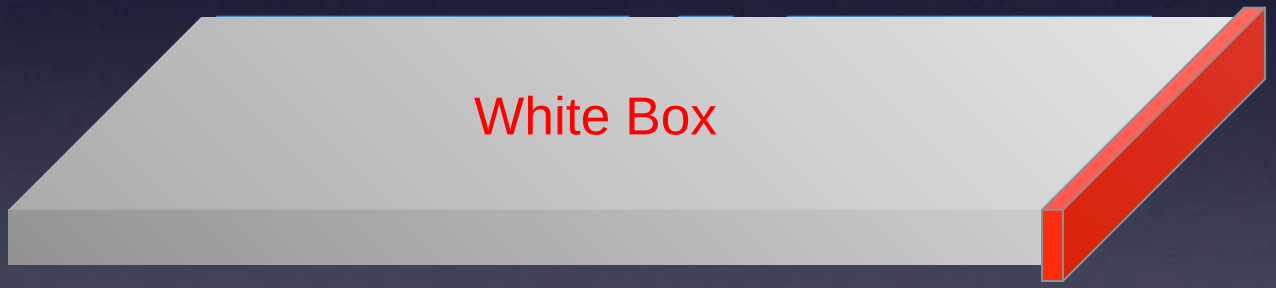
SDN data plane and NFV could be unified



FLARE

(Unifying SDN Software Defined Data Plane and NFV)

Physical Resources: CPU + NPU



THE UNIVERSITY OF TOKYO

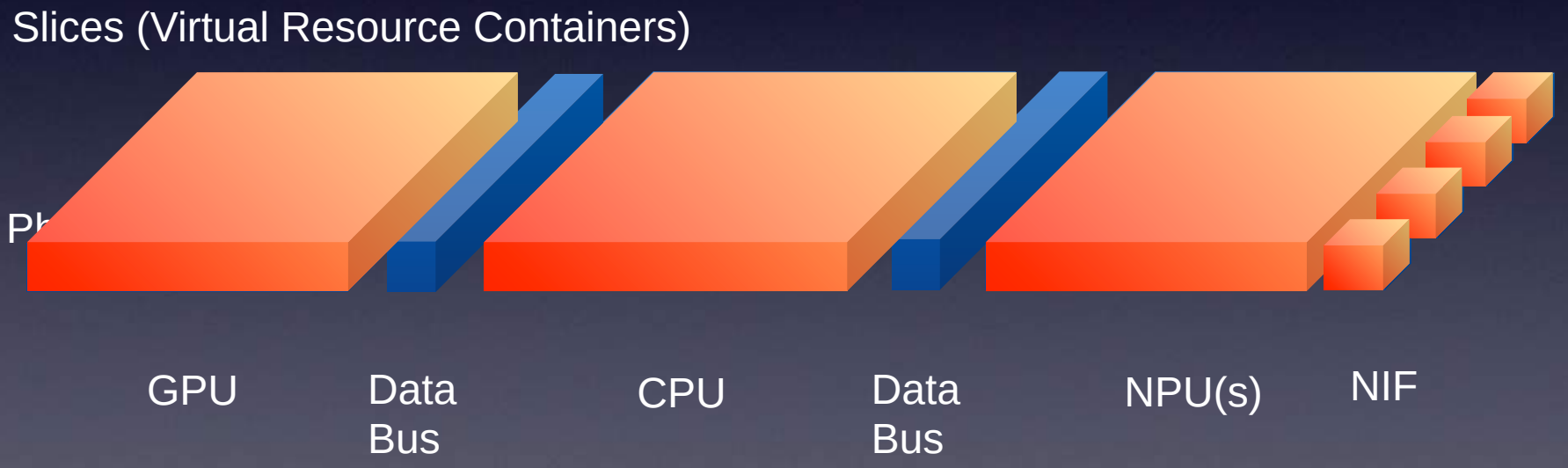
FLARE

Physical Resources: CPU + NPU (+ GPU)

GPU	Data Bus	CPU	Data Bus	NPU(s)	NIF
Intel Phi	PCIe Gen3	Intel x86	PCIe Gen3	Tilera	

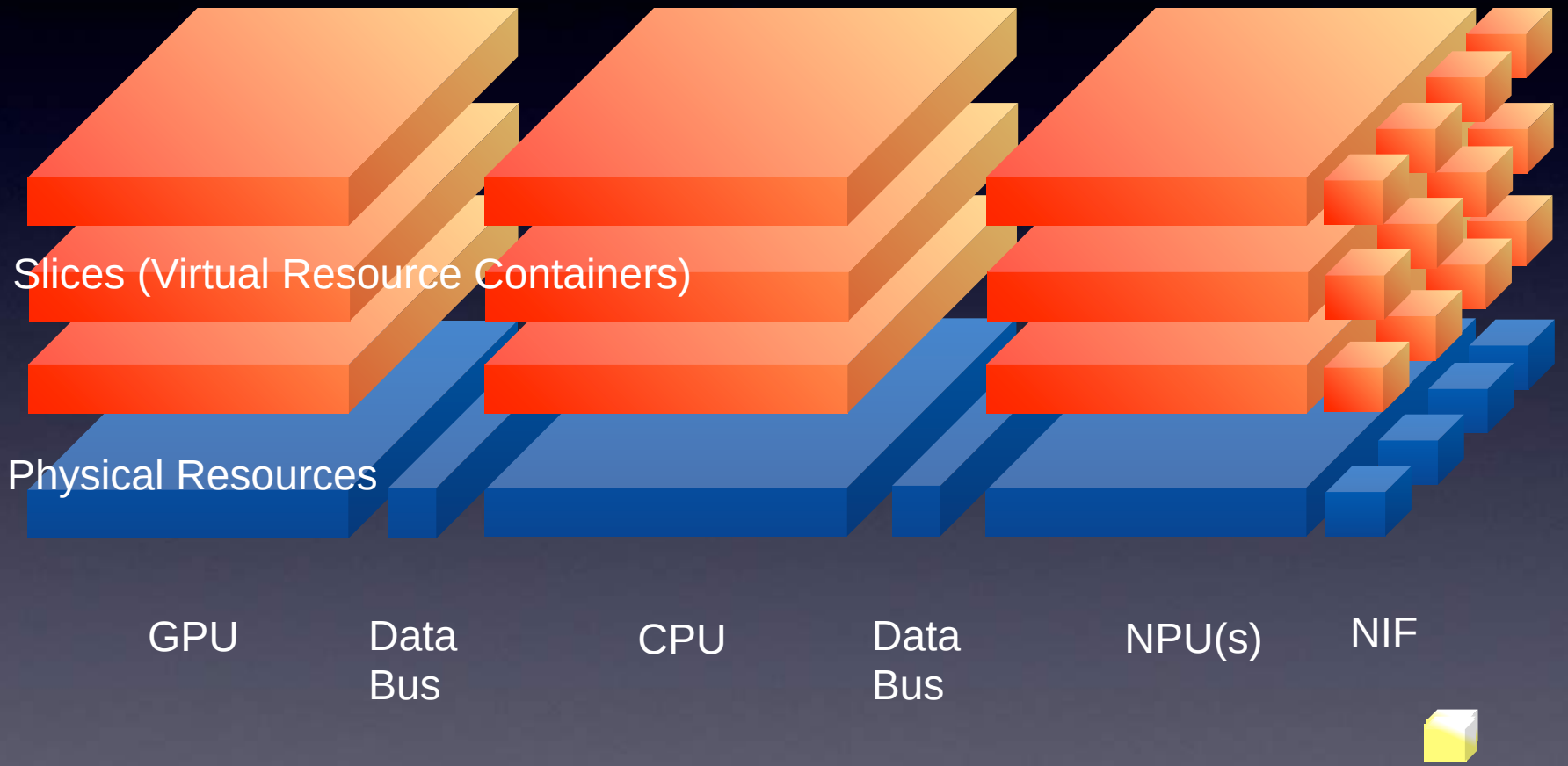
FLARE

Virtualization (Resource Container) -> Slices of Resources

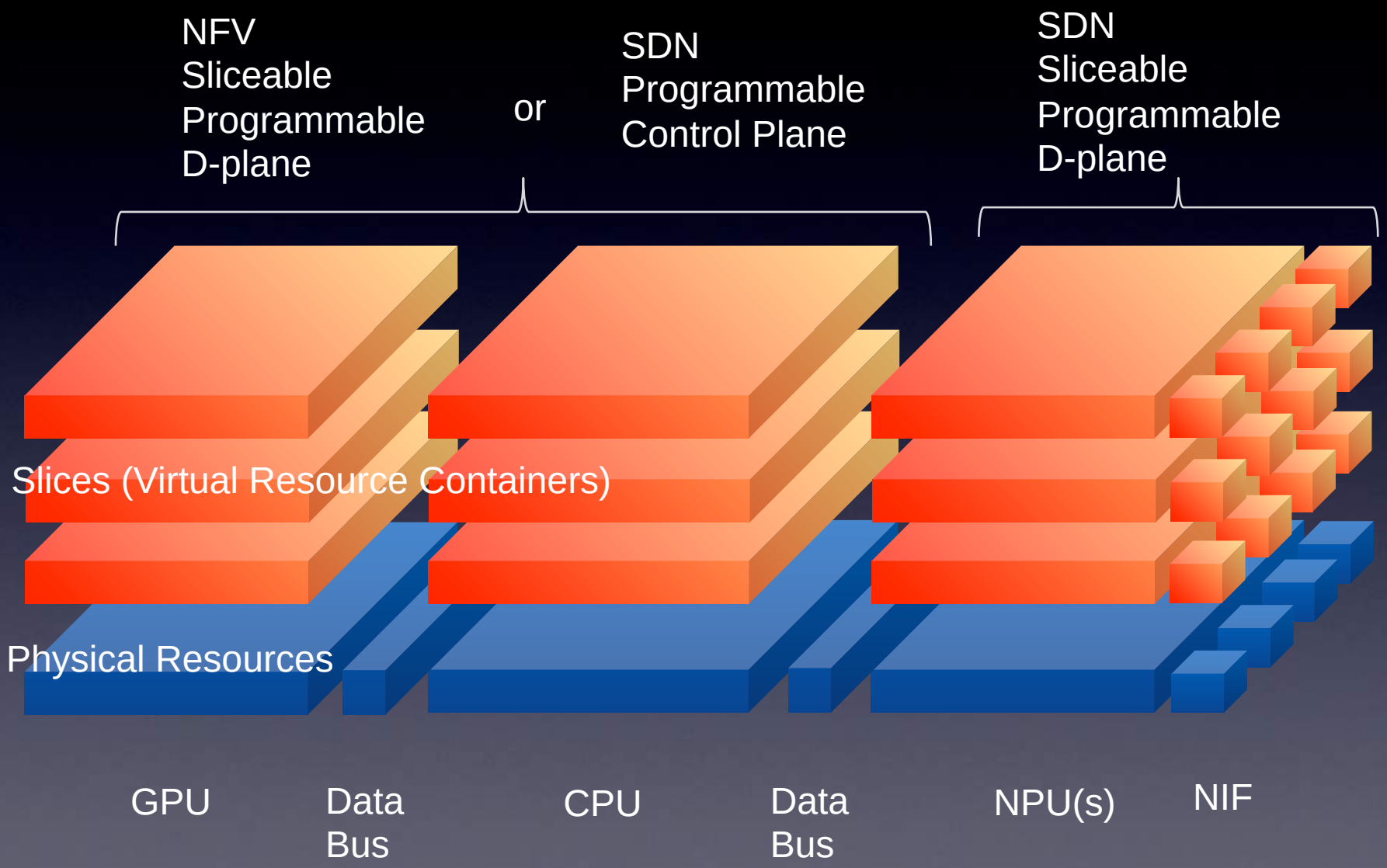


Sliceable !

FLARE

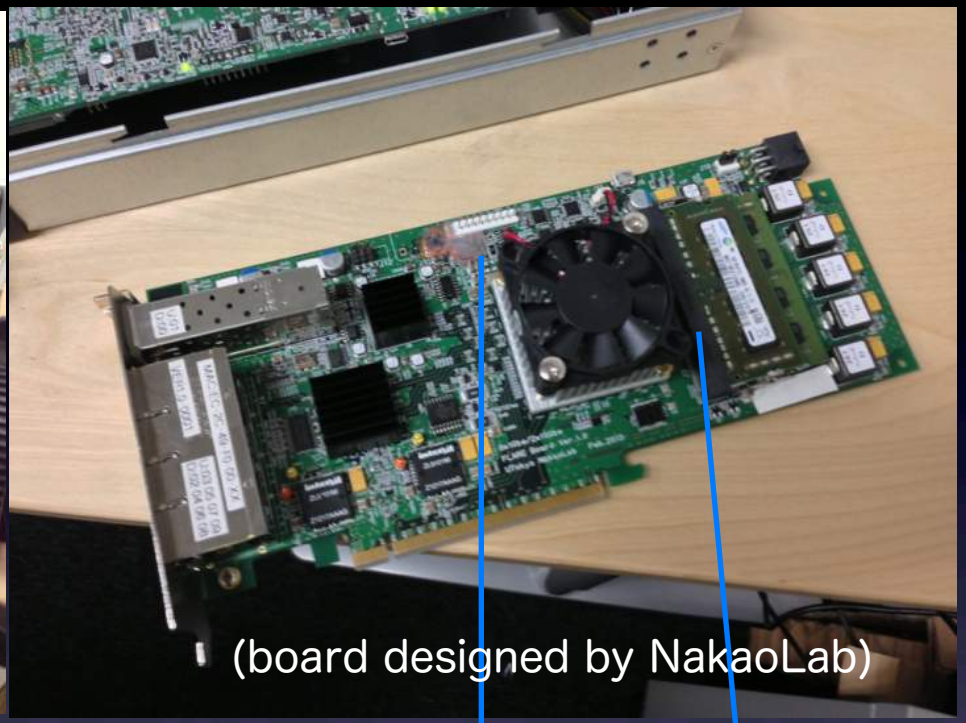

Slices (Virtual Resource Containers)
Physical Resources
GPU
Data Bus
CPU
Data Bus
NPU(s)
NIF

FLARE

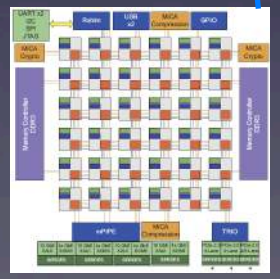


THE UNIVERSITY OF TOKYO

FLARE Node Implementation

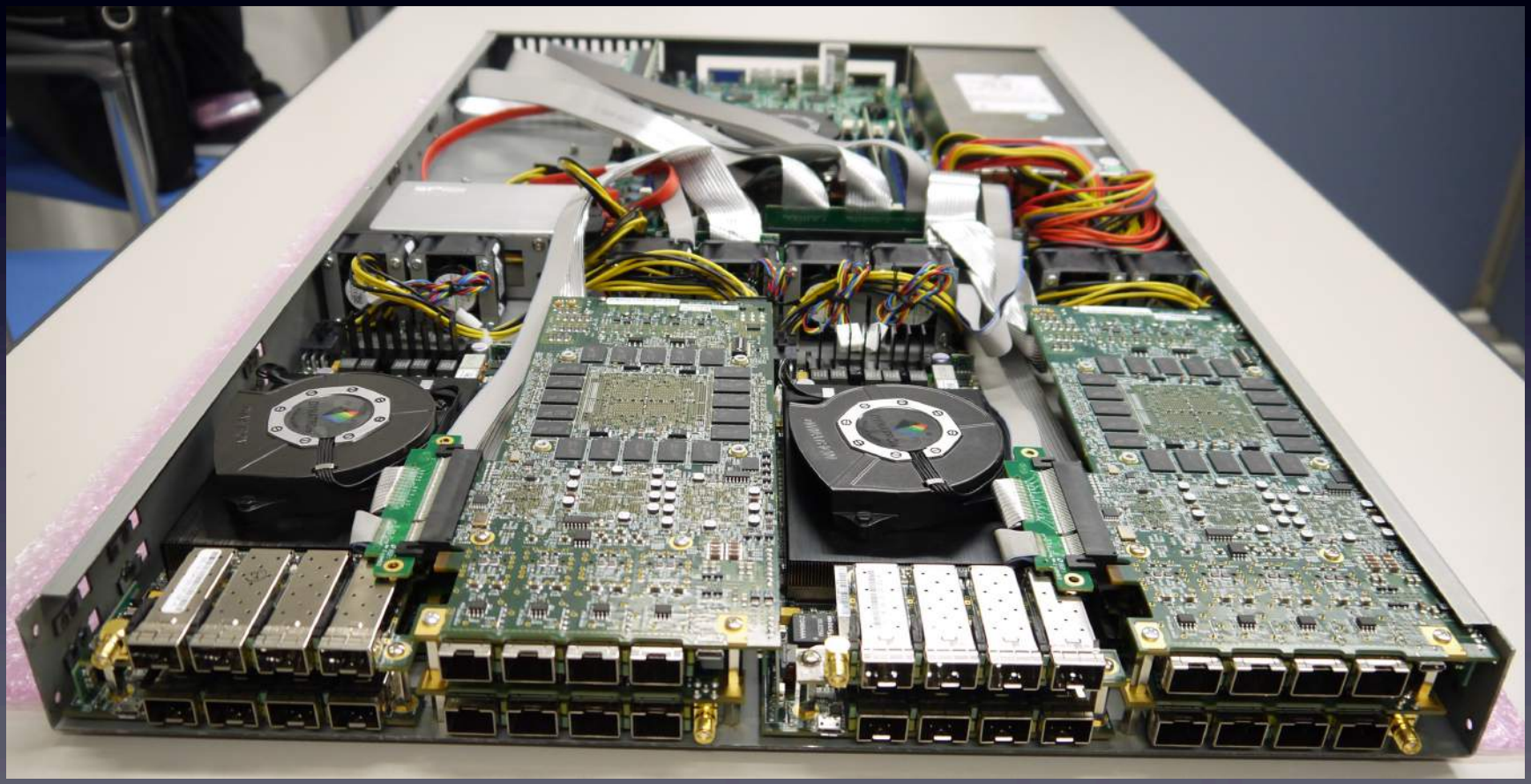


36–72 cores
(upto 100-200 cores in future)

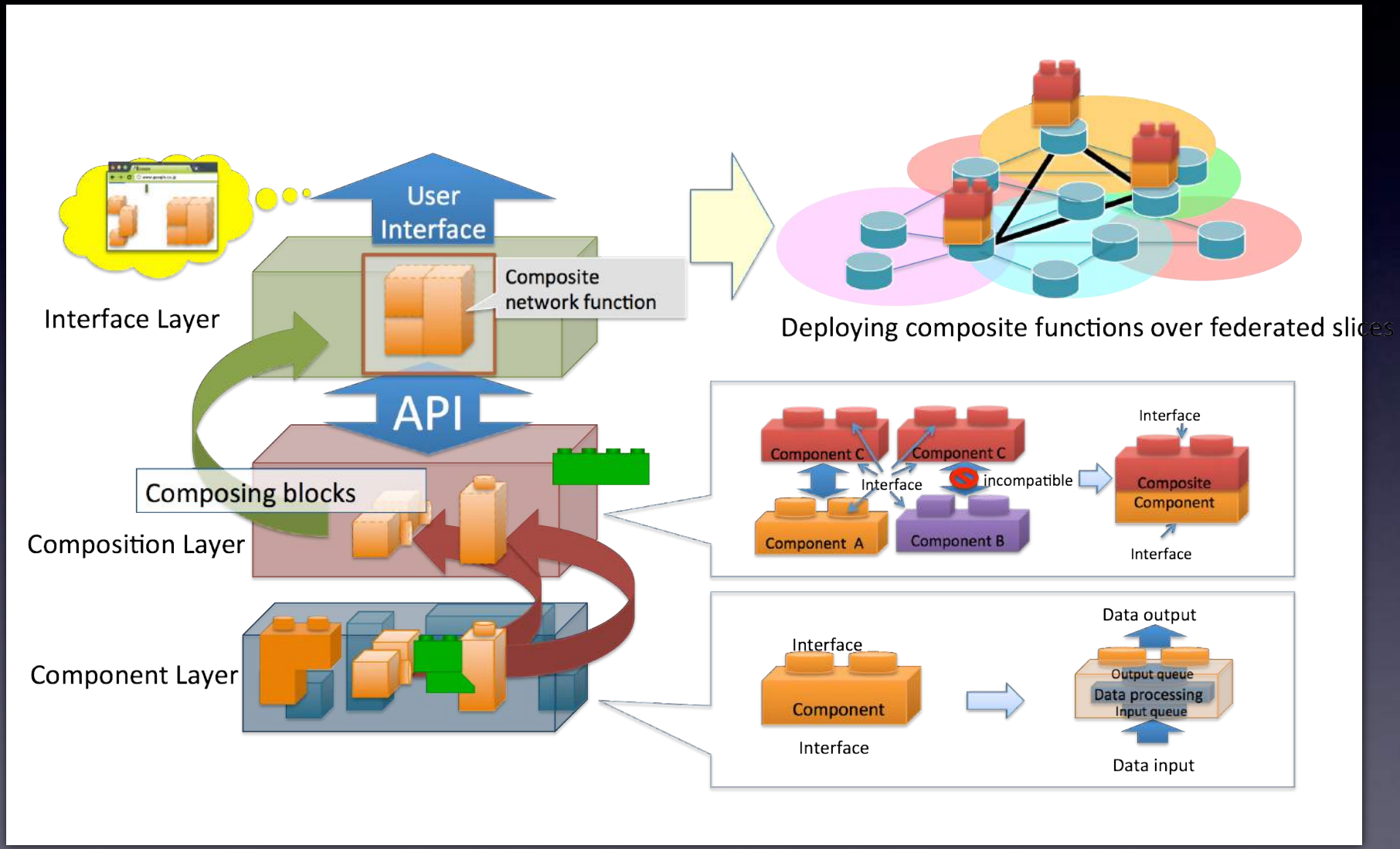


Hierarchical Resource Management
- General Purpose Processor(s)
- Network Processor(s)
- ...and more types of processors

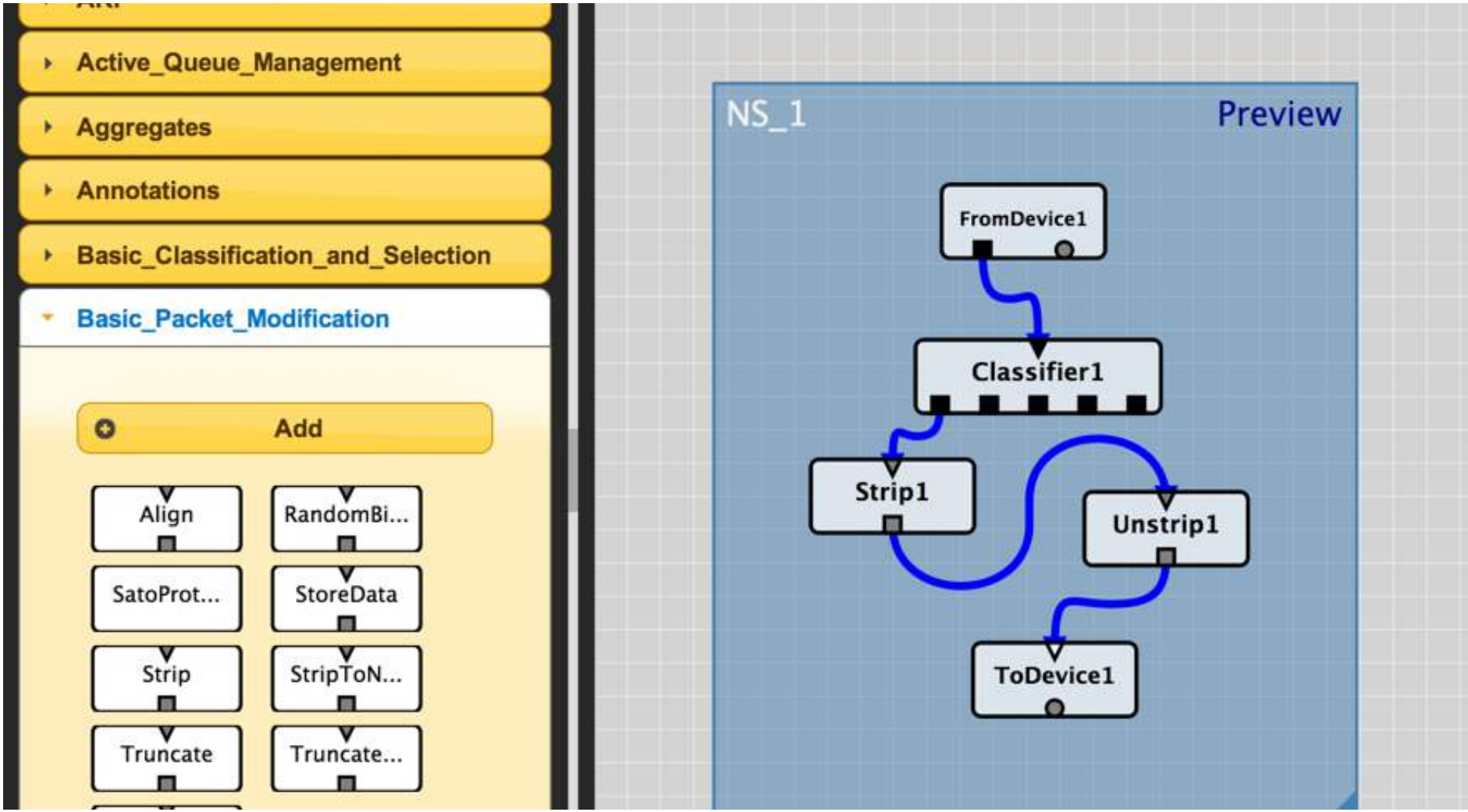
FLARE-EX Prototype (32x10Gbps)



Toy-Block Networking



ToyBlock Networking GUI



Standardization Possibility

- Deep (Data Plane) Programmability
  - Slicing (Virtualization)
  - Evolvable APIs
  - Viewed as extension to SDN
  - Also viewed as extension to NFV
  - Rethinking architecture of SDN/NFV for enabling “Deep programmability”
- Application Driven Thinking
  - Top-Down approach is necessity
- Toy Block Networking
  - Accommodate a wide range of programmers
  - Marketing of reusable network function “Toy Blocks”

Conclusion

"Software Eats Everything"

"Application Driven Thinking
empowered with
Software Defined Data Plane
will extend SDN/NFV further
and accelerate standardization!"