"Software-Defined Data Plane Enhancing SDN and NFV"

Pursuing "Deeper Programmability"

Aki Nakao

TTC

(The University of Tokyo)

GSC 18

2014/7/23

SDN Architecture

Applications

Network Applications
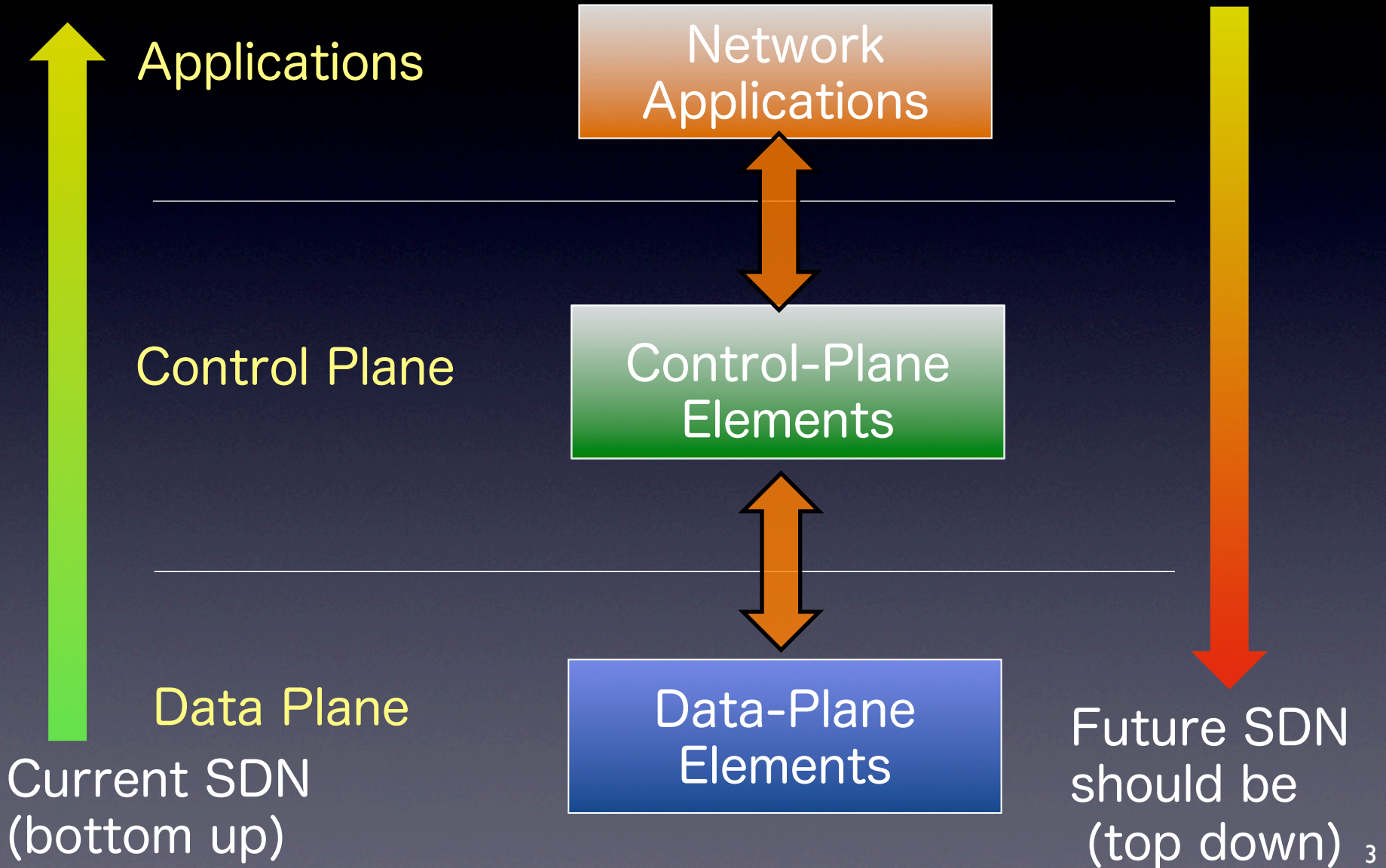
North Bound Interface (NBI)

Control Plane

Control-Plane Elements

South Bound Interface (SBI)

Data Plane

Data-Plane Elements

Application Driven Thinking

Applications

Network
Applications

Control Plane

Control-Plane
Elements

Data Plane

Data-Plane
Elements

Current SDN
(bottom up)

Future SDN
should be
(top down)

3

Application-Driven Thinking

Premise:

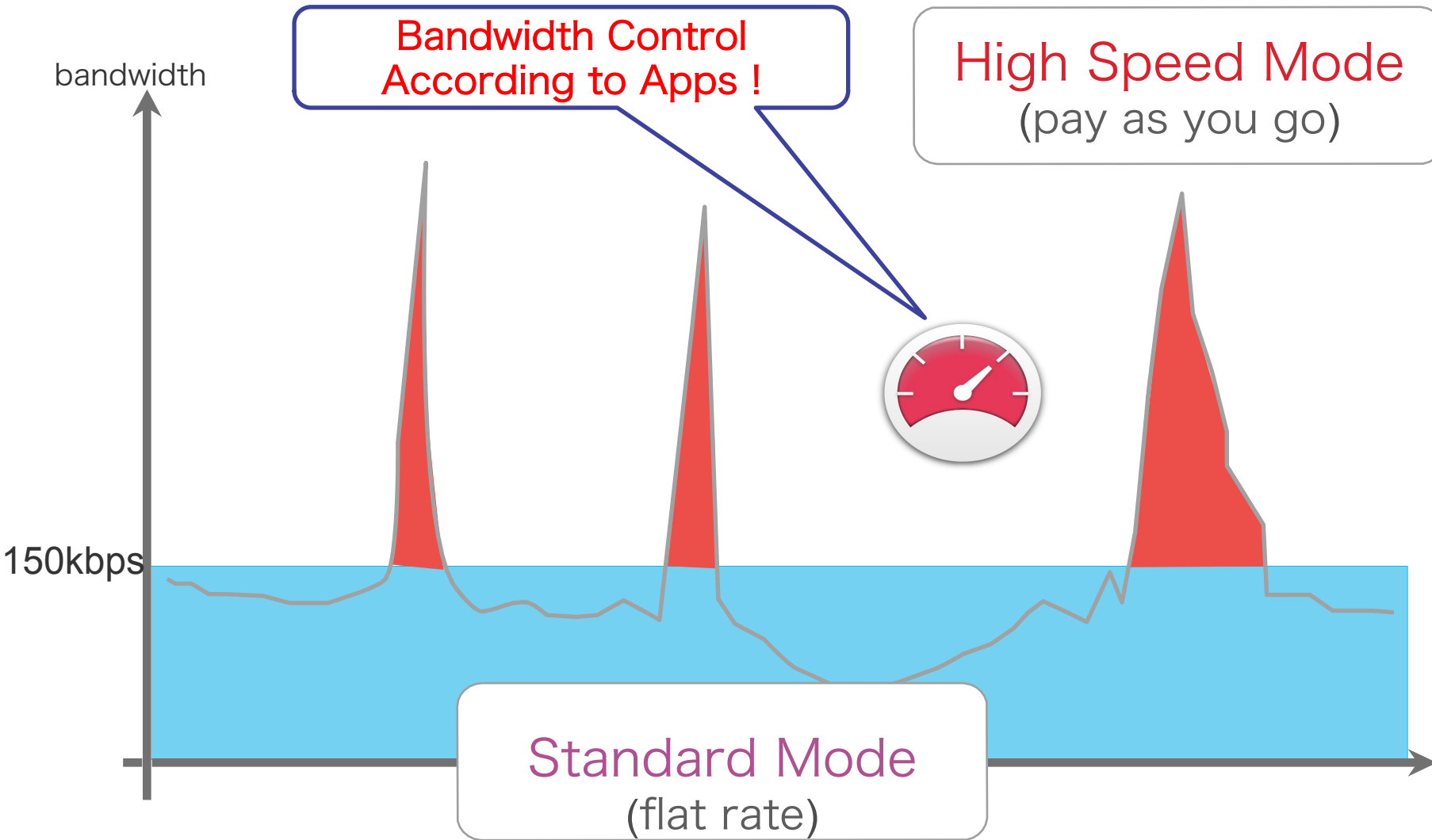Programmable networking has been enabled by SDN and NFV

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

bandwidth

Bandwidth Control
According to Apps !

High Speed Mode
(pay as you go)

150kbps

Standard Mode
(flat rate)

App-specific traffic control enables more fine-grained subscription plans that can get an MVNO out of the "ever-lower-cost" competition

Our Proposal
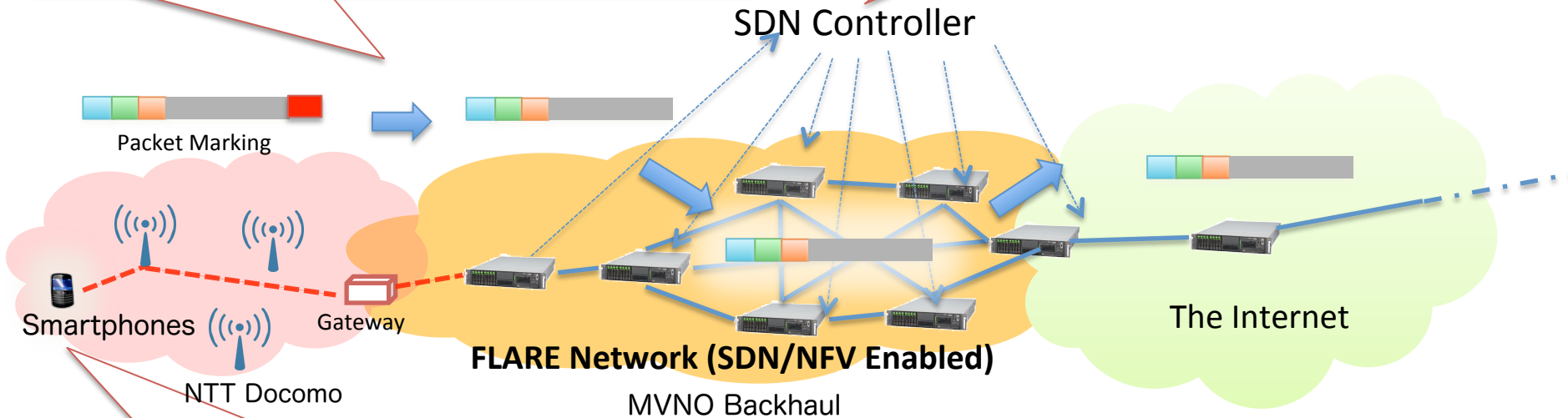
FLARE（Deeply Programmable Node）

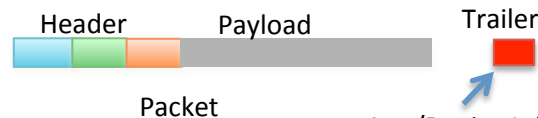Parse and remove "trailers" and map between flows and apps

Header   Payload

Parse and Remove Trailers

SDN Controller

Traffic Engineering based on headers

Header   Payload

SDN Controller

Packet Marking

Smartphones

NTT Docomo

Gateway

FLARE Network (SDN/NFV Enabled)

MVNO Backhaul

The Internet

Smartphones (wearables)

Add app/device information to packet trailers

Header   Payload   Trailer

Packet

App/Device Information

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

App-Specific Traffic Control

Remote console of programmable network node (FLARE)

Smartphone connected to our MVNO

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)

Application Driven SDN

Some ISPs need more direct SDN Southbound Interface

- Flow abstraction in Southbound Interface is for operators

  <Flow Pattern> <Action> <Stat>

- App/Device abstraction is useful and intuitive

  <App/Device> <Action><Stat>

Application Driven SDN QoS

Bandwidth Control
According to Apps !

Smartphone

FLARE
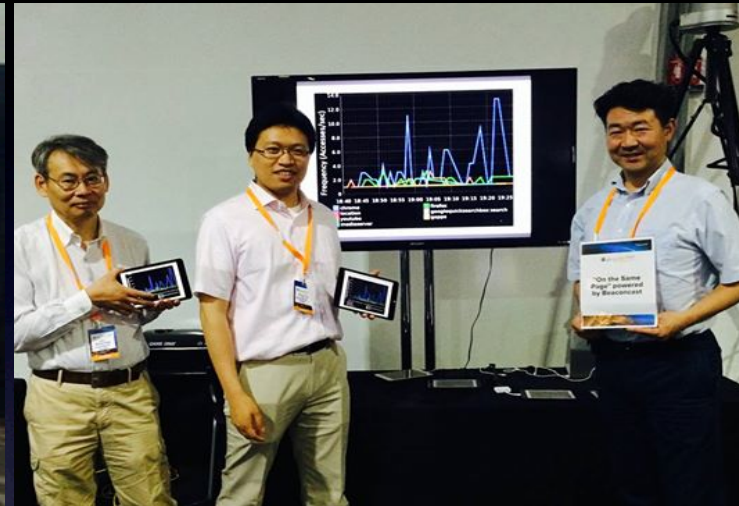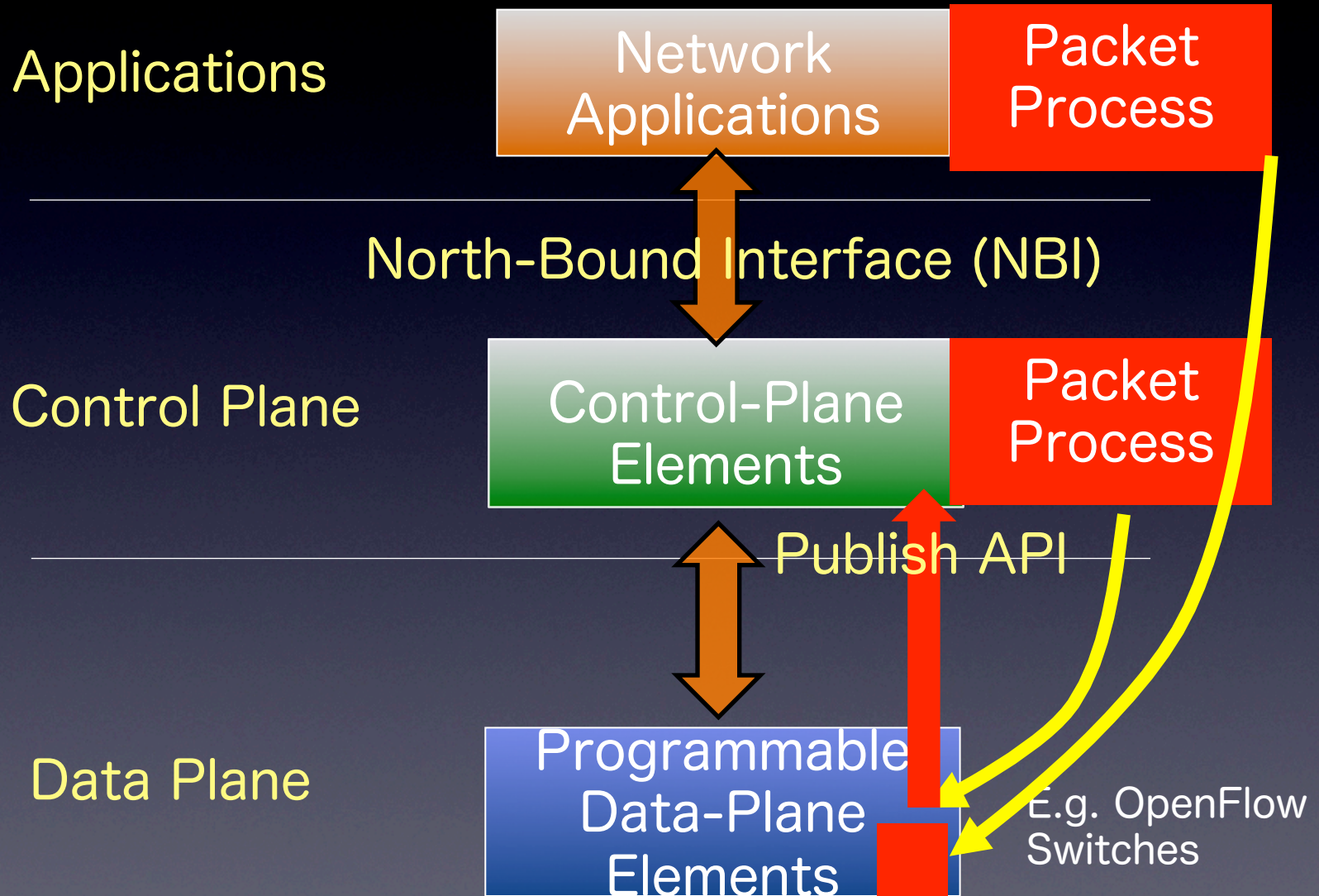Deeply programmable network node
With software defined data plane

Packet

App

Chrome : Pass Thru
Firefox: Block
YouTube: Rate Limit

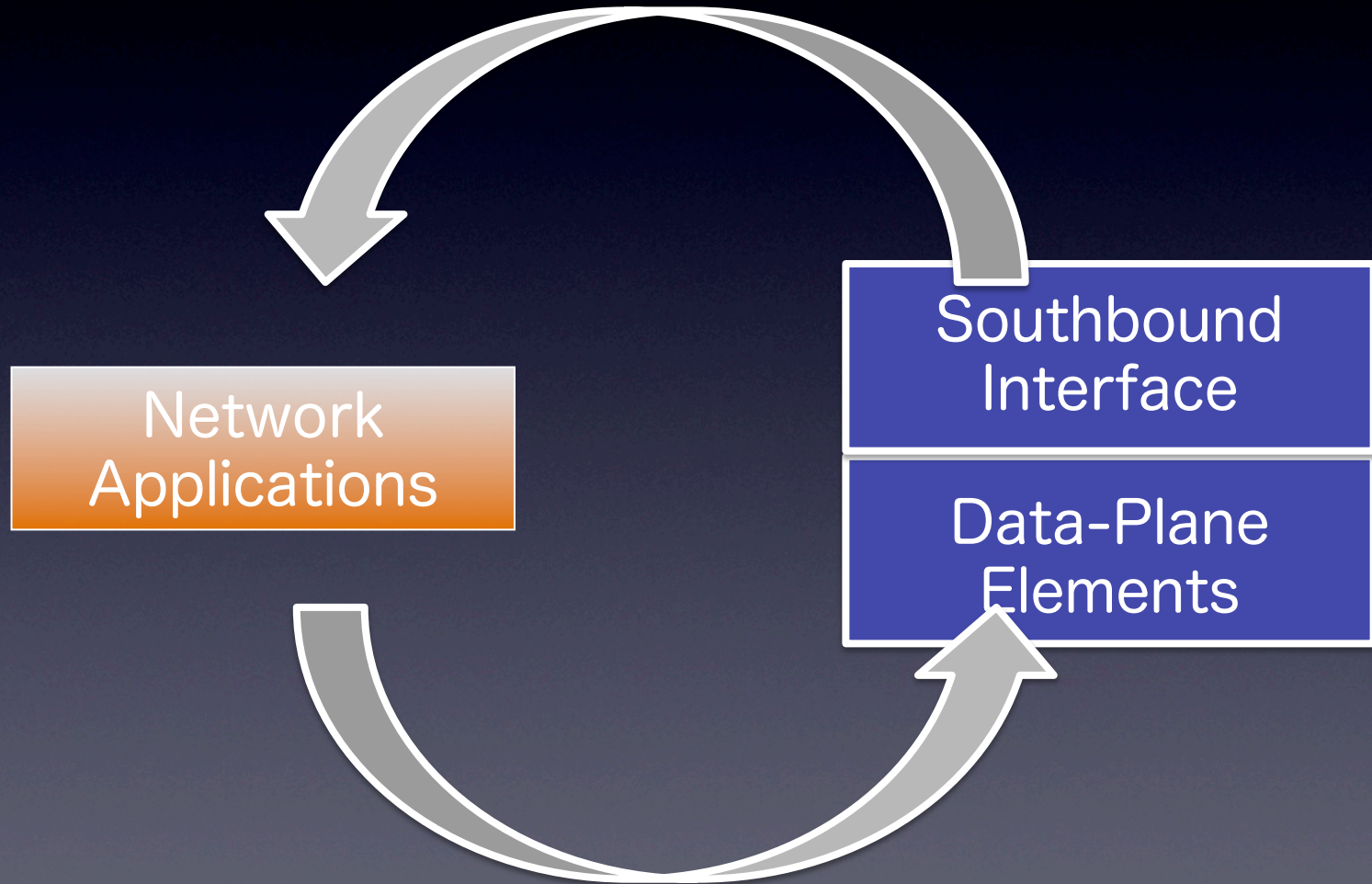# We won the best demo award!
GEC20@UC Davis

Software Defined Data Plane

Applications

| Network Applications | Packet Process |

North-Bound Interface (NBI)

Control Plane

| Control-Plane Elements | Packet Process |

Publish API

Data Plane

Programmable Data-Plane Elements

E.g. OpenFlow Switches

13

Innovation Cycle

Operation and Evaluation Feedback

Network Applications

Southbound Interface

Data-Plane Elements

Application Driven Thinking

ITU-T Y.3300 (Y.SDN-FR)
"Framework of software-defined networking,"
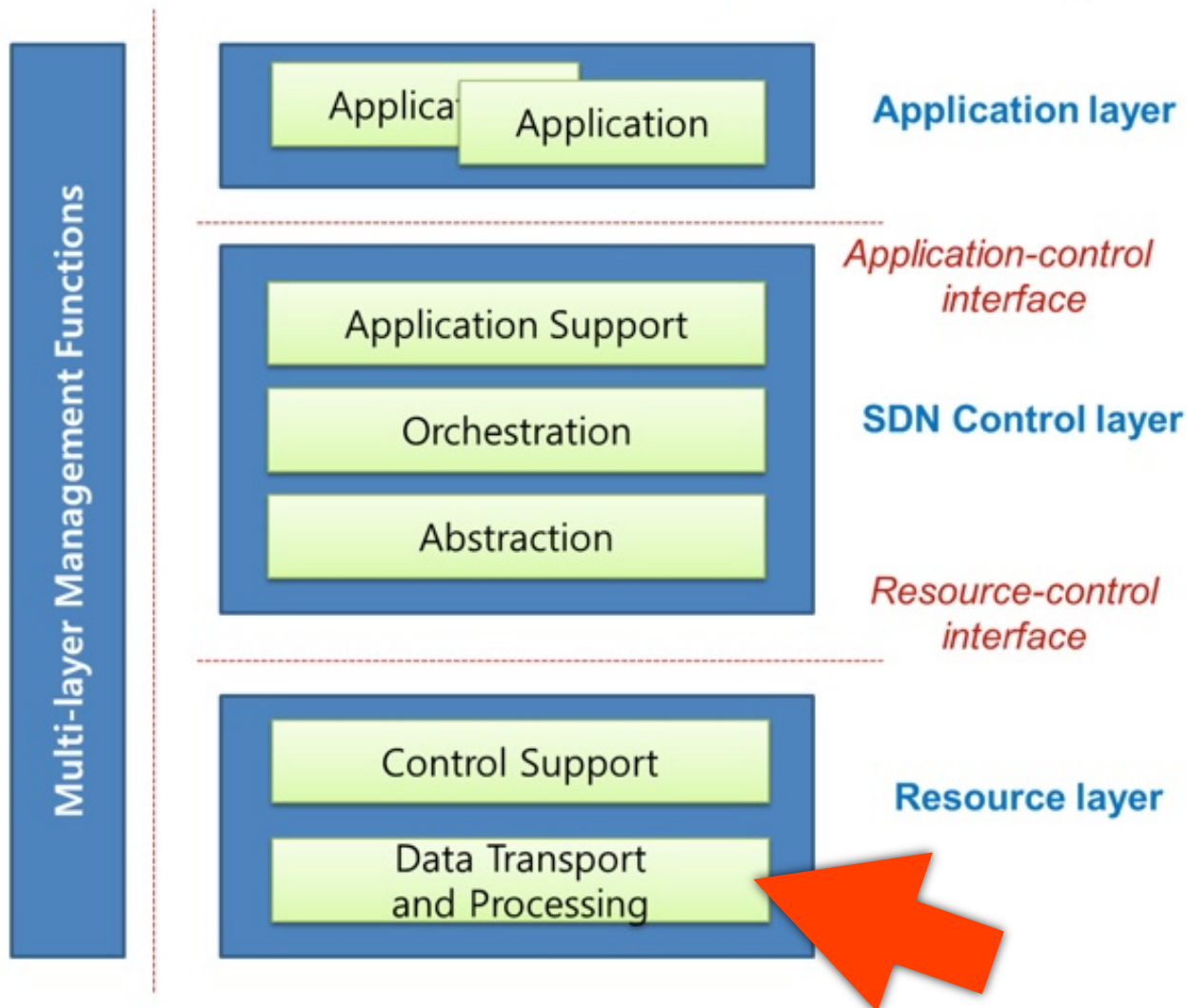
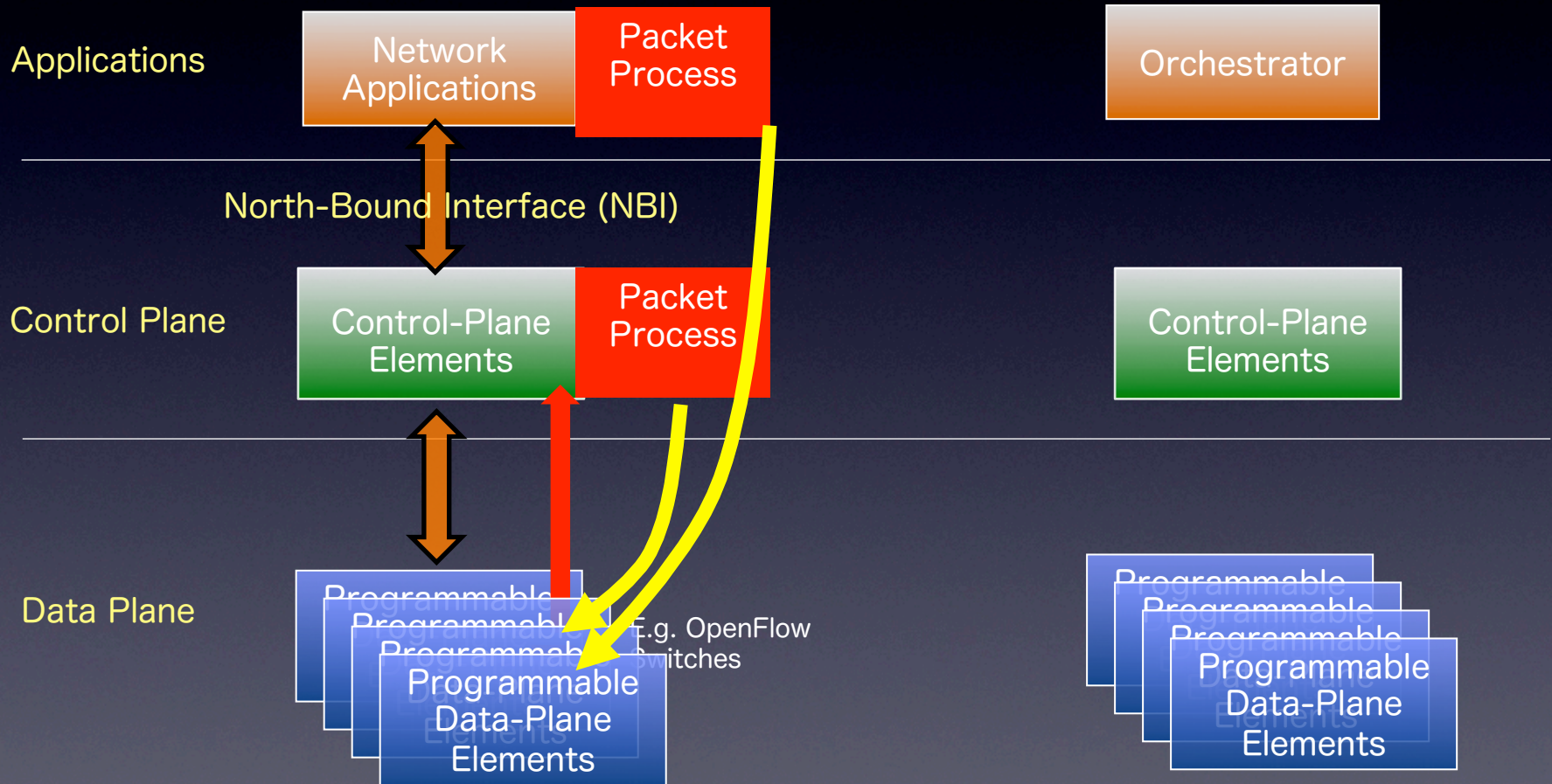**Figure 2 – High-level architecture of SDN**

Sliceable Software Defined Data Planes

Applications

Network Applications

Packet Process

North-Bound Interface (NBI)

Control Plane

Control-Plane Elements

Packet Process

Publish API

Data Plane

Programmable Data-Plane Elements

E.g. OpenFlow Switches

16
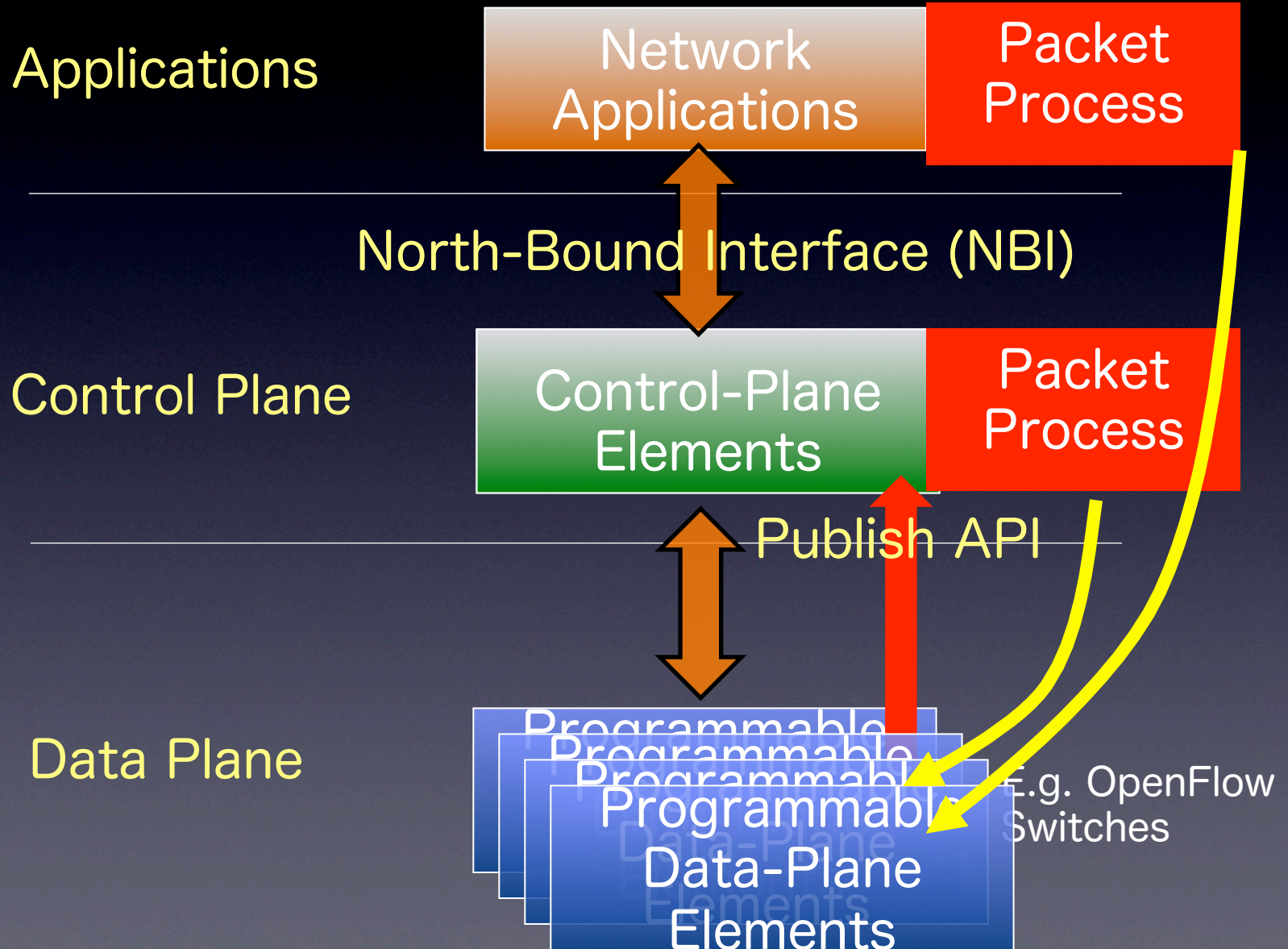
SDN data plane and NFV could be unified

SDN for Network Control

NFV for Data Processing

Applications

Network Applications	Packet Process

Orchestrator

North-Bound Interface (NBI)

Control Plane

Control-Plane Elements	Packet Process

Control-Plane Elements

Data Plane

Programmable Data-Plane Elements

e.g. OpenFlow Switches

Programmable Data-Plane Elements
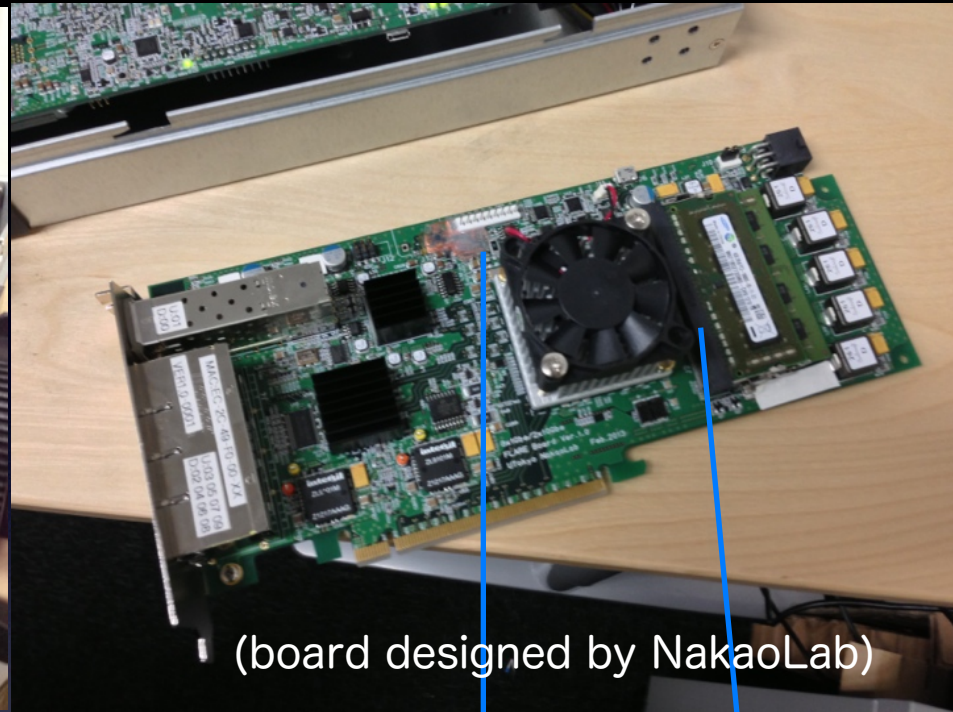
FLARE Node Implementation

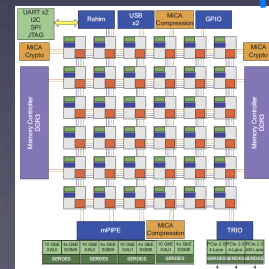x86 Processor

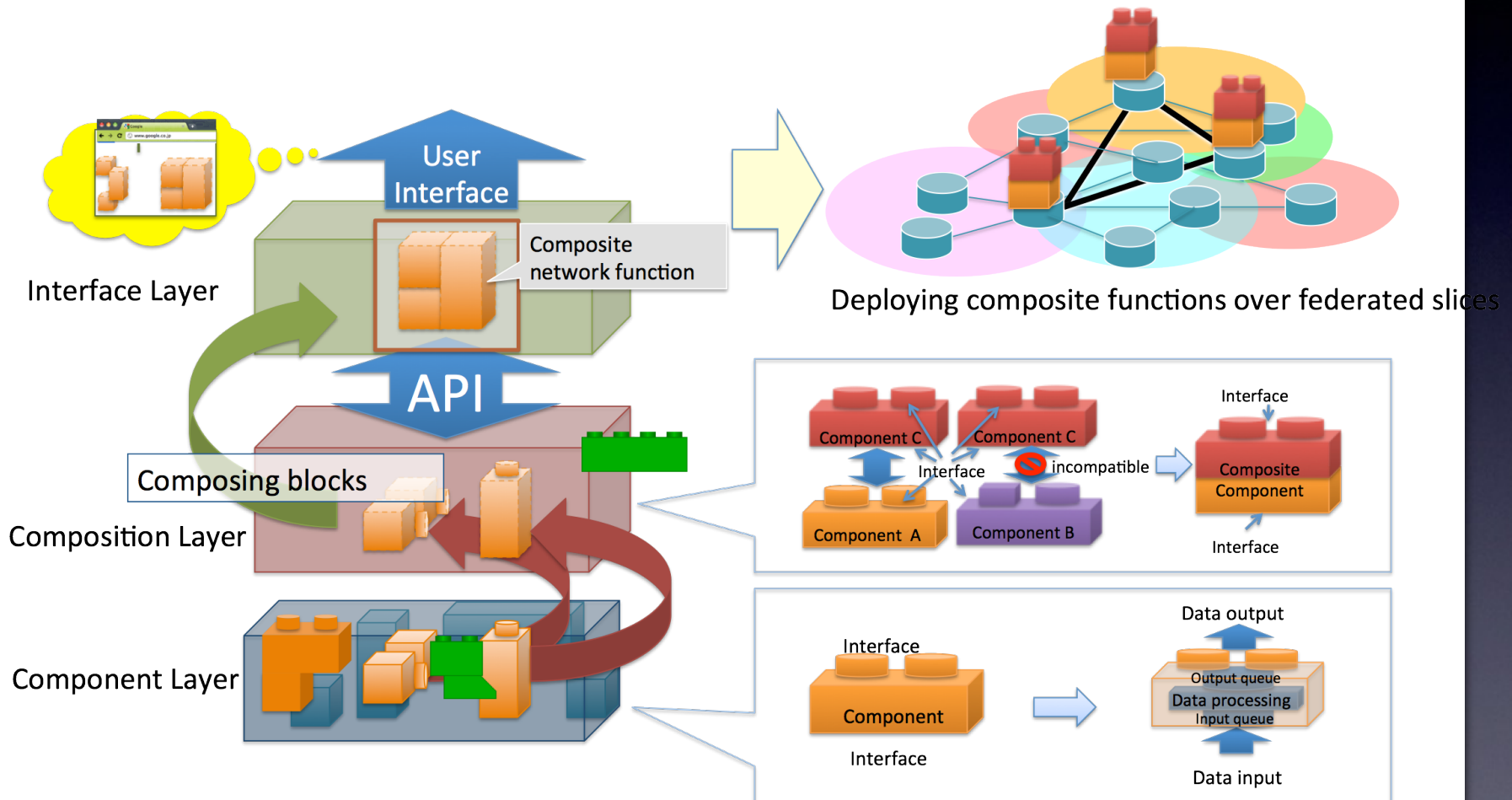Many Core Processor

(board designed by NakaoLab)

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

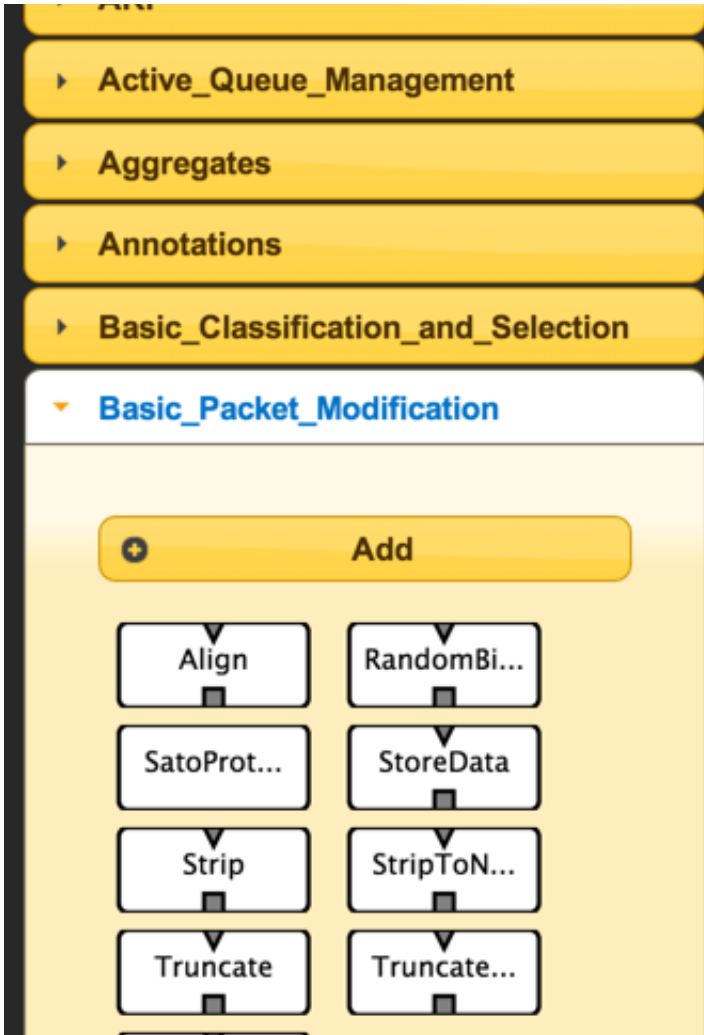## Hierarchical Resource Management
- General Purpose Processor(s)
- Network Processor(s)
- ...and more types of processors

Programming Model
Toy-Block Networking

Interface Layer

User Interface

Composite network function

Deploying composite functions over federated slices

API

Composing blocks

Composition Layer

Component C   Component C   Interface

Interface   incompatible   Composite Component

Component A   Component B   Interface

Component Layer

Interface

Component

Interface

Data output

Output queue
Data processing
Input queue

Data input

Toy-Block Networking GUI
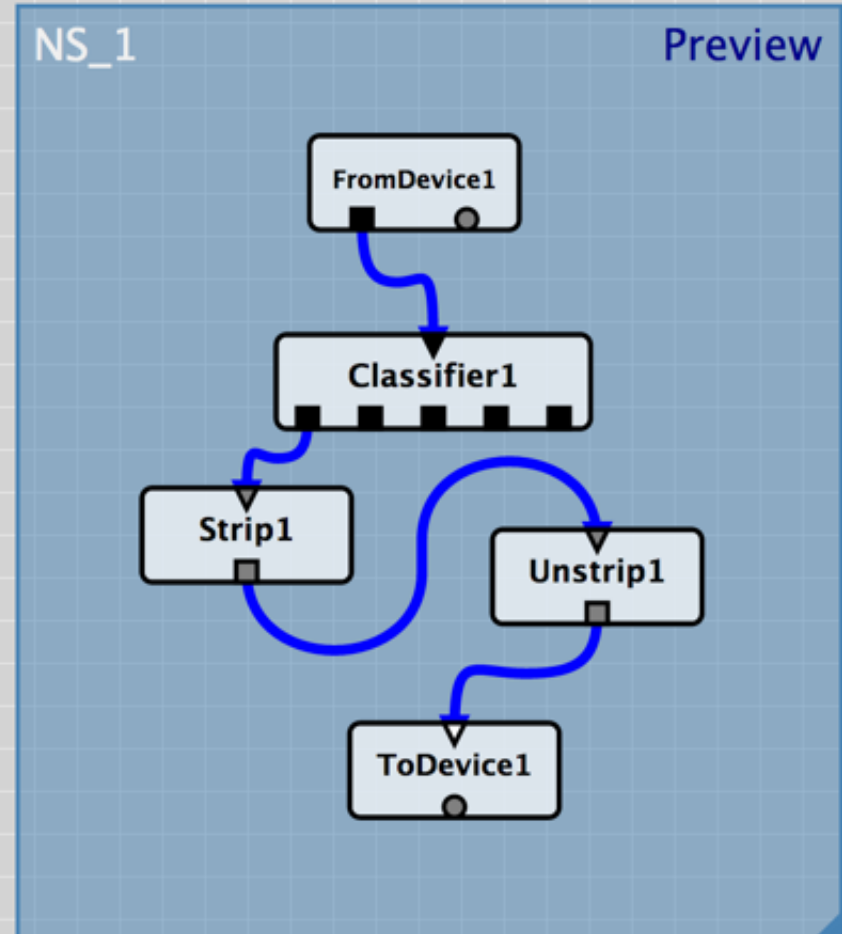
Summary

Missing from the current landscape of SDN and NFV

- Application Driven Thinking
  - Top-down, dynamic update of software
  - User, app, device, service oriented modeling
- Deep (Data Plane) Programmability
  - SDN data plane as a network function in NFV
  - Data plane slicing (virtualization)
  - Evolve-able APIs
  - New protocol handling
- Programming Model
  - Toy-Block Networking
  - Accommodate a wide range of programmers
  - Marketing of reusable network function blocks