



PROGRAMMABLE NETWORK SYSTEMS THROUGH SDK

JunosSDK (including JunosVEE) , JunosSpaceSDK

Hidetsugu Sugiyama

Director of R&D Support, APAC/Senior Architect
Juniper Networks Inc.

March 5th, 2012

WTC2012, Miyazaki, Japan
WORLD TELECOMMUNICATIONS CONGRESS

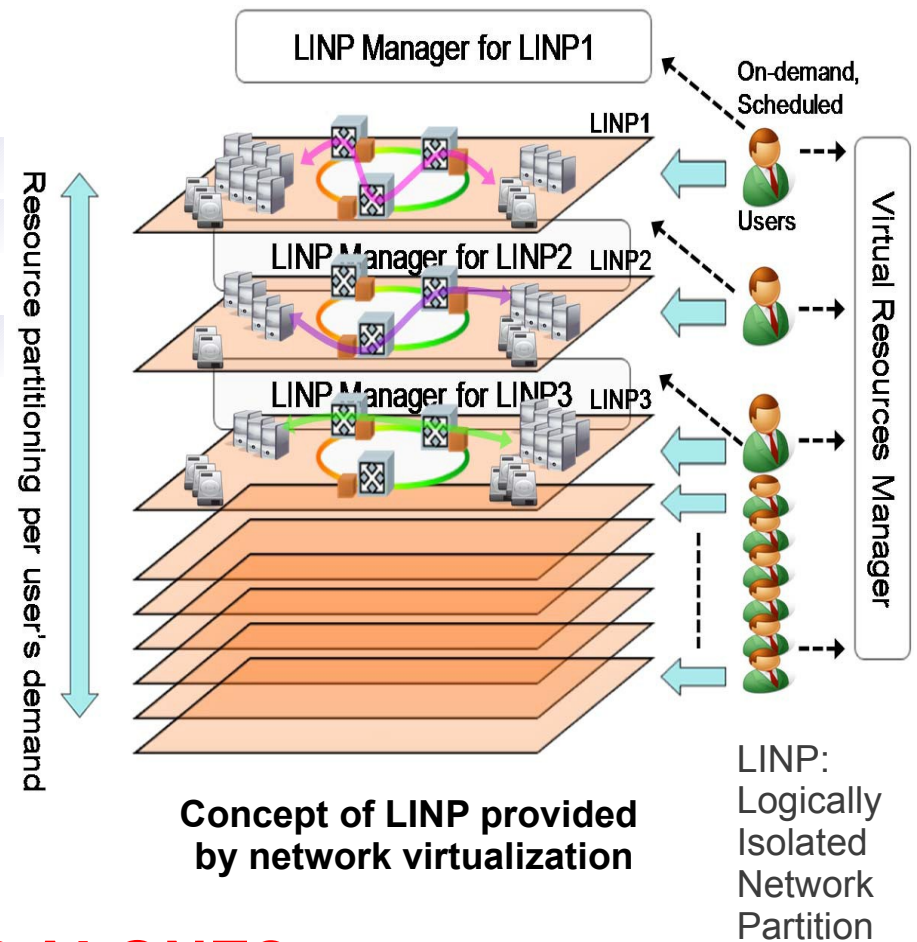
TS-B3: Network Virtualization

FRAMEWORK OF NETWORK VIRTUALIZATION

ITU-T Y.3011

◆ Design goals

- 1) Isolation
- 2) Programmability
- 3) Performance
- 4) Network abstraction
- 5) Topology awareness and quick reconfiguration
- 6) Management
- 7) Mobility
- 8) Wireless

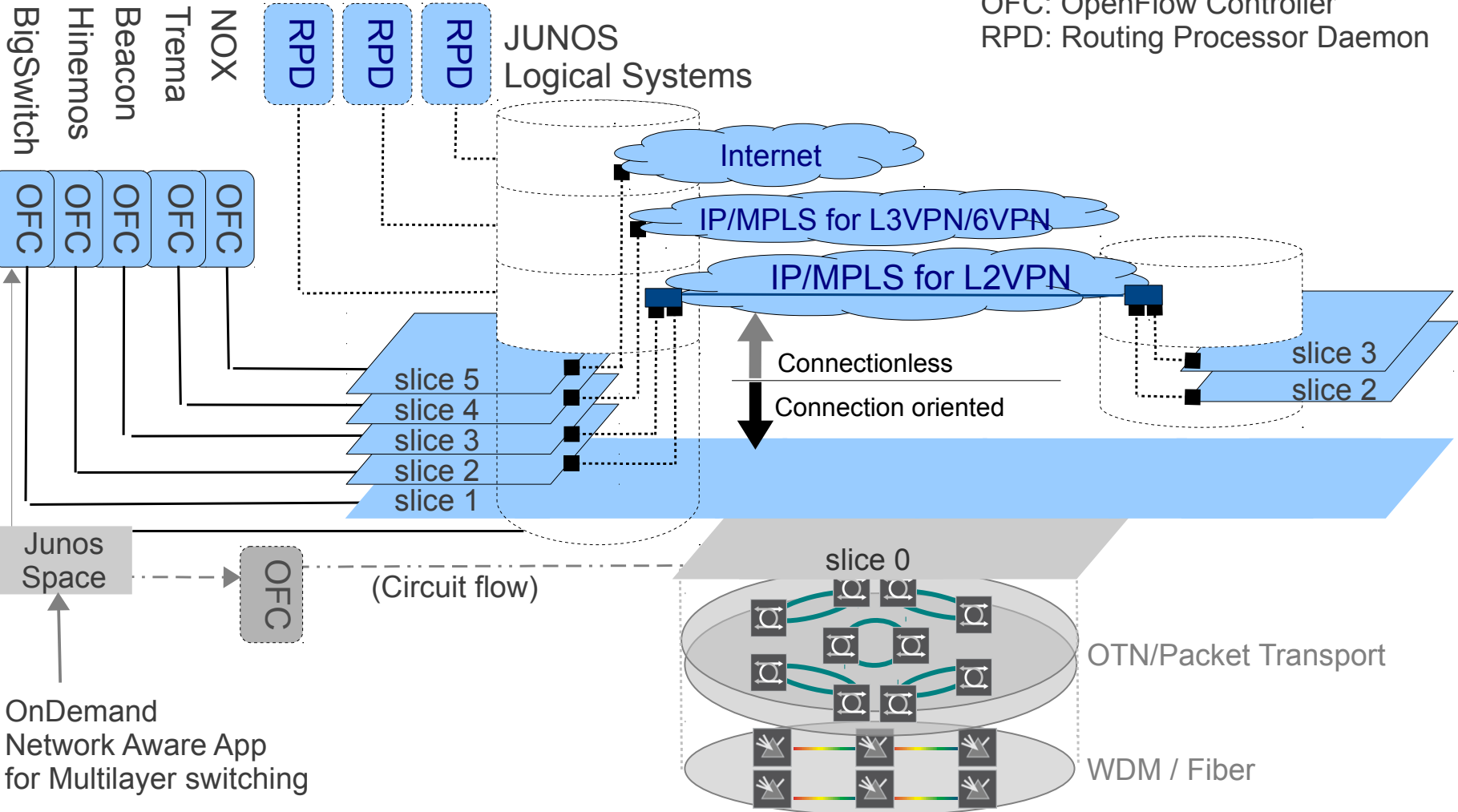


DO WE HAVE TO DEVELOP ALONE?

JUNOS CAPABILITIES FOR NETWORK VIRTUALIZATION

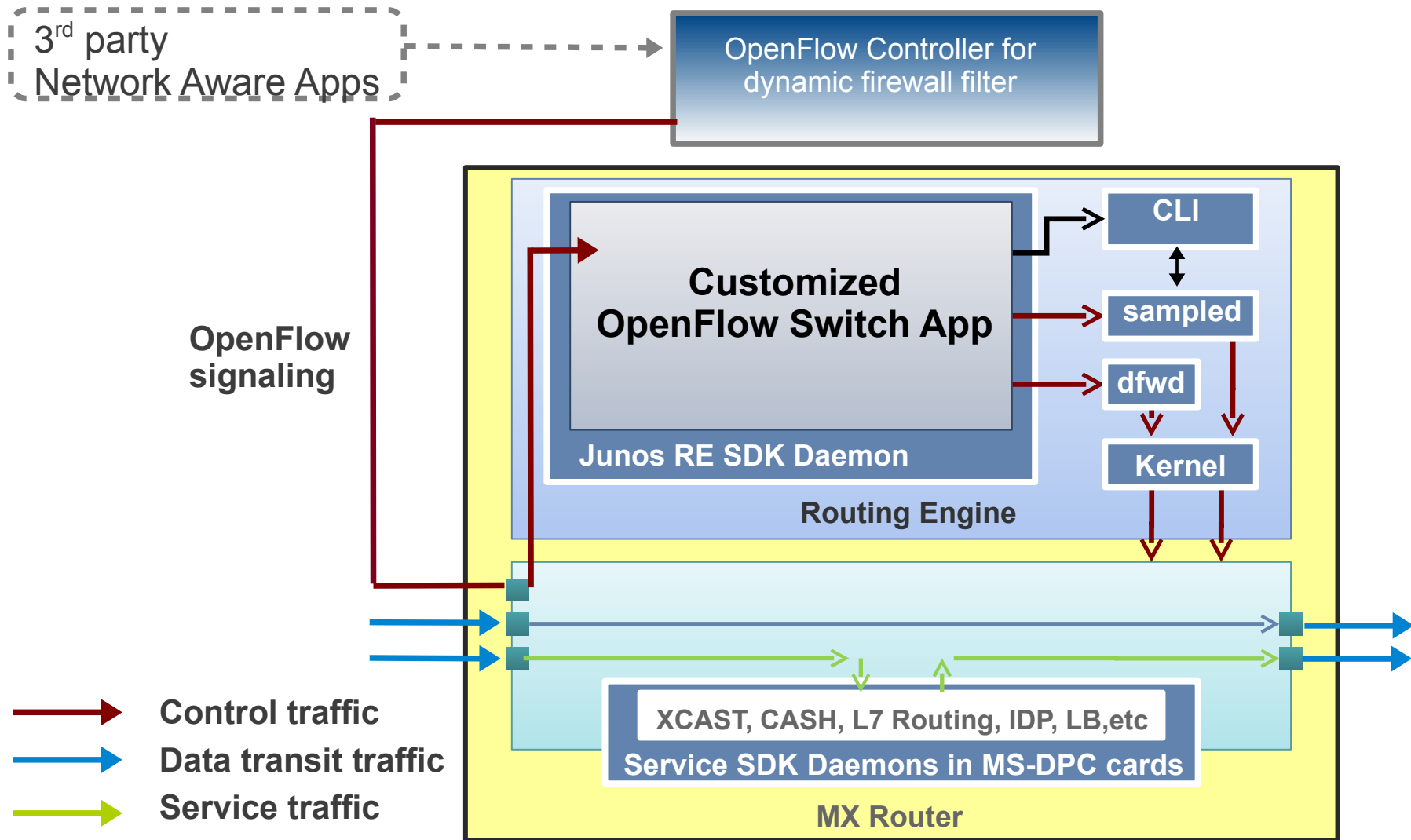
OpenFlow Ver1.0

JUNOS SDK version 11.4 prototype



OFC: OpenFlow Controller
RPD: Routing Processor Daemon

OPENFLOW ENHANCEMENT WITH JUNOS SDK FOR INCREASING PROGRAMMABILITY



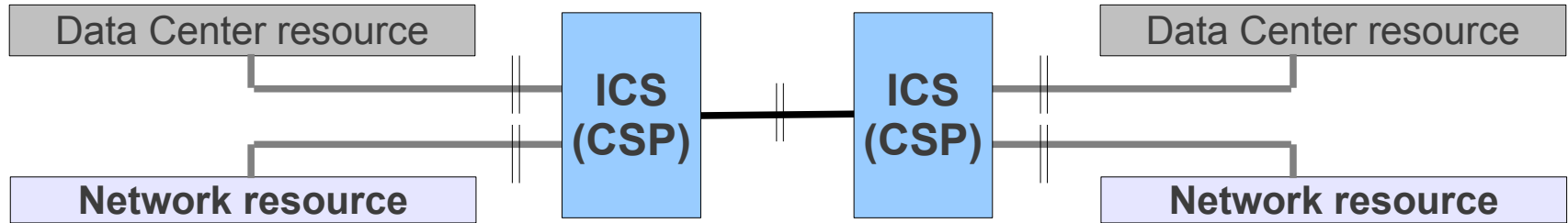
JUNOS SDK IDE FOR OPENFLOW SWITCH CODE

The screenshot displays the Junos SDK IDE environment. The main editor window shows the file `of-switch.cnf.dd` with the following code:

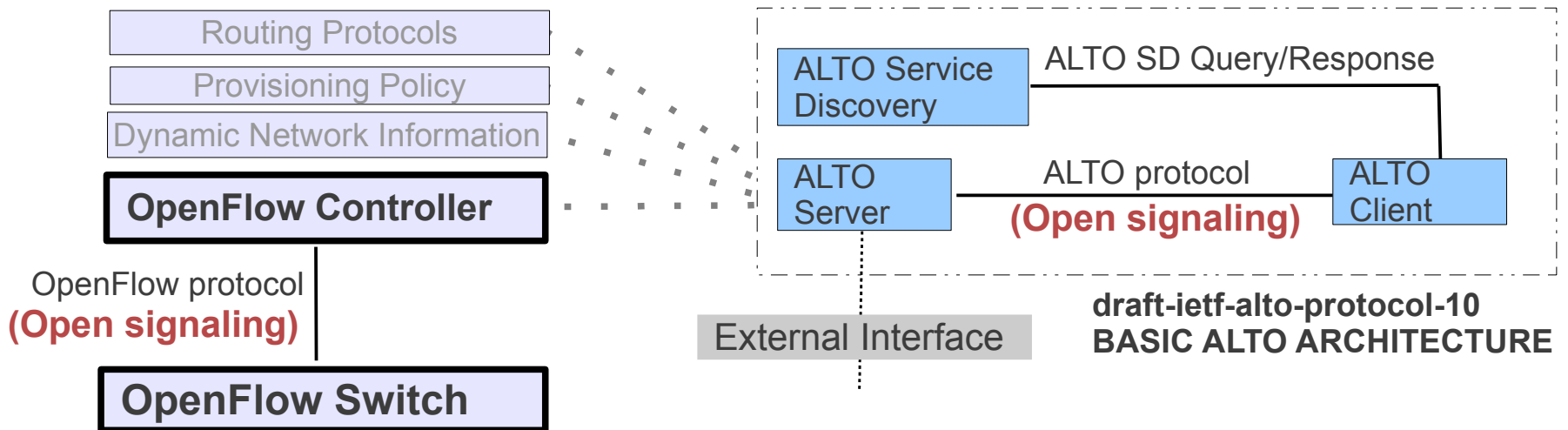
```
cname "of_switch_ctrl_protocol";
type enum uint {
    choice tcp {
        help "Set protocol type to 'TCP' (default)";
        define OF_CTRL_PROTO_TCP;
    }
    choice ssh {
        help "Set protocol type to 'ssh'";
        define OF_CTRL_PROTO_SSH;
    }
    default tcp;
}
attribute address {
    flag mandatory;
    type ipv4addr;
    cname "of_switch_ctrl_address";
    help "Controller's IPv4 address";
}
attribute port {
    type ranged int 1024 .. 65535;
    cname "of_switch_ctrl_port";
    help "Controller's port number (default 6633)";
}
}
attribute no-vpls {
    help "Do not use VPLS address family";
    cname "of_switch_no_vpls";
    type toggle;
}
}
/*
 * Trace options configuration:
 */
object traceoptions {
    help "OpenFlow switch daemon trace options";
    require trace;
    flag remove-empty;
```

The left sidebar shows a project tree for `of-switchd` with files like `of-switchd-cmn.c`, `of-switchd-config.c`, and `of-switchd-dfw-clnt.c`. The bottom status bar indicates the current file is `of-switch.cnf.dd` with a 1:1 zoom level.

GICTF INTER CLOUD RESOURCE MODEL, BASIC ALTO ARCHITECTURE

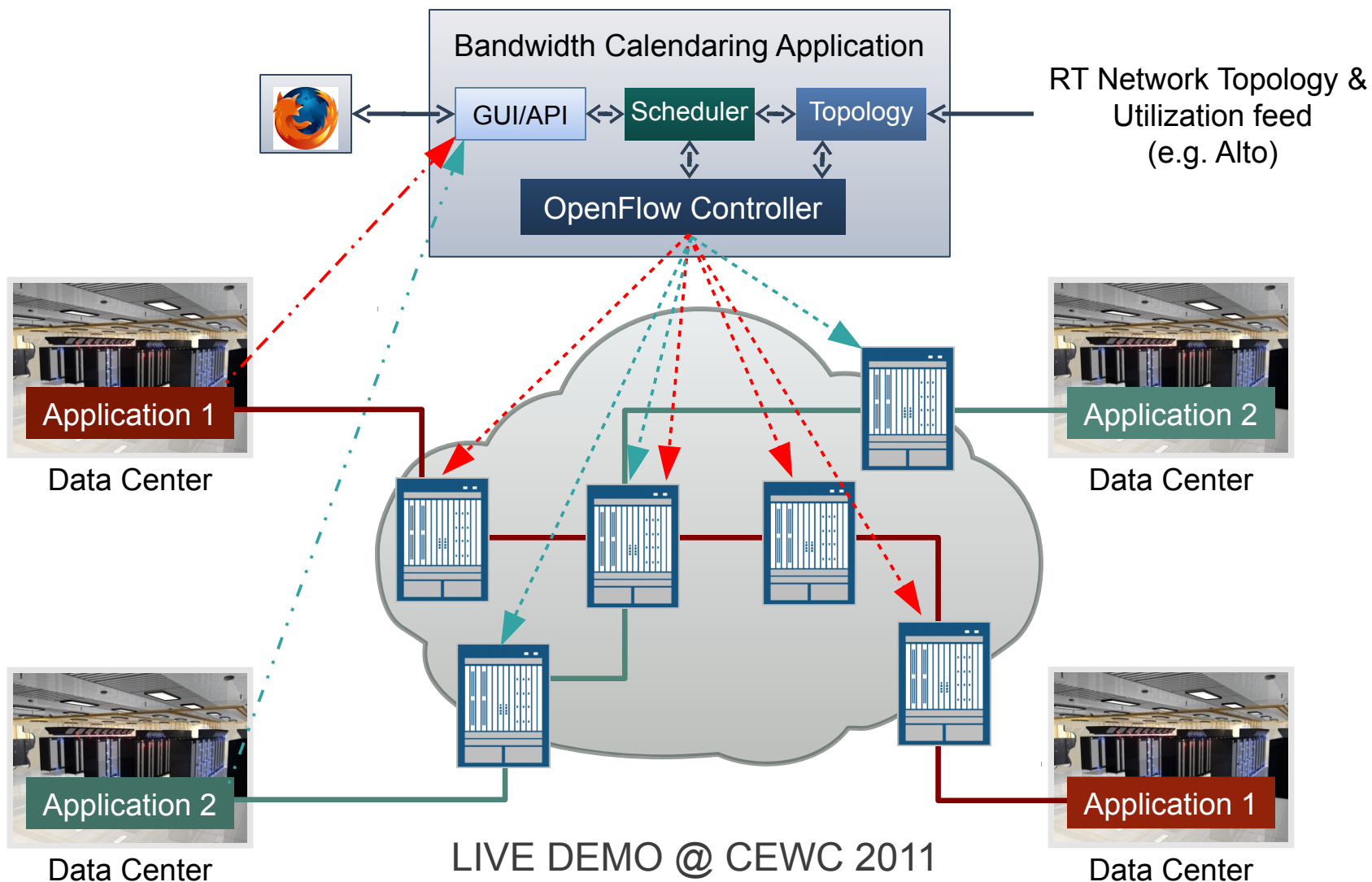


<http://www.gictf.jp/documents.html>
GICTF Inter Cloud resource model

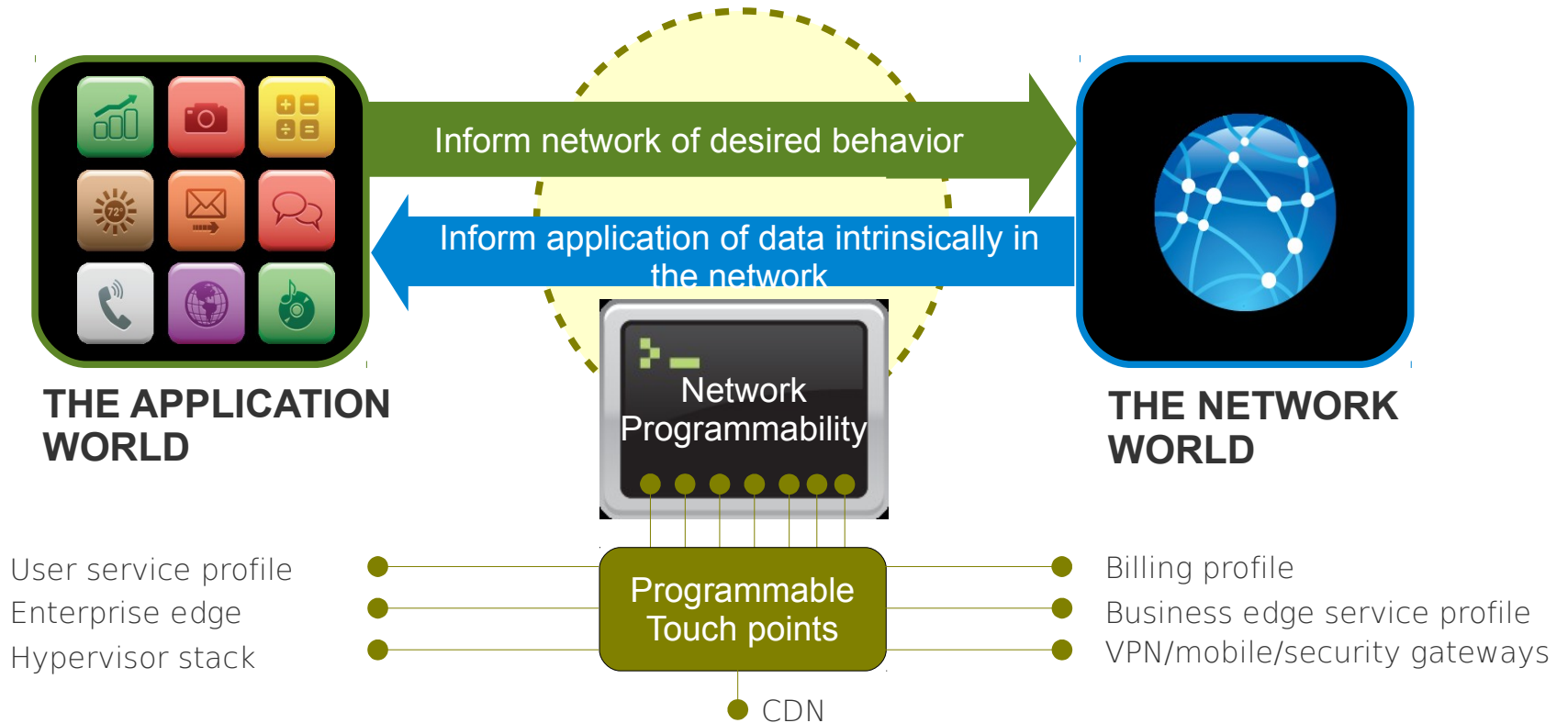


GICTF : Global InterCloud Technology Forum
ICS : InterCloud System
ALTO : Application Layer Traffic Optimization

USE CASE: CALENDAR BANDWIDTH APPLICATION



BI-DIRECTIONAL INTERACTION AND PROGRAMMABILITY

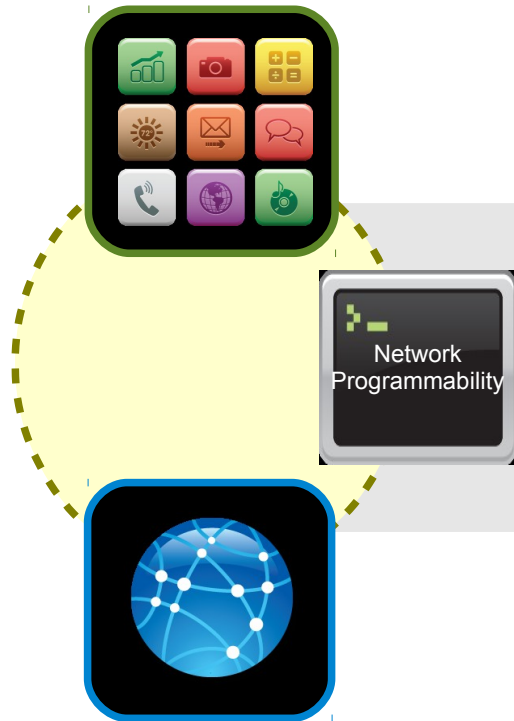


Touchpoints to extract information or influence behavior. Platforms use touch points. Developers use platforms.

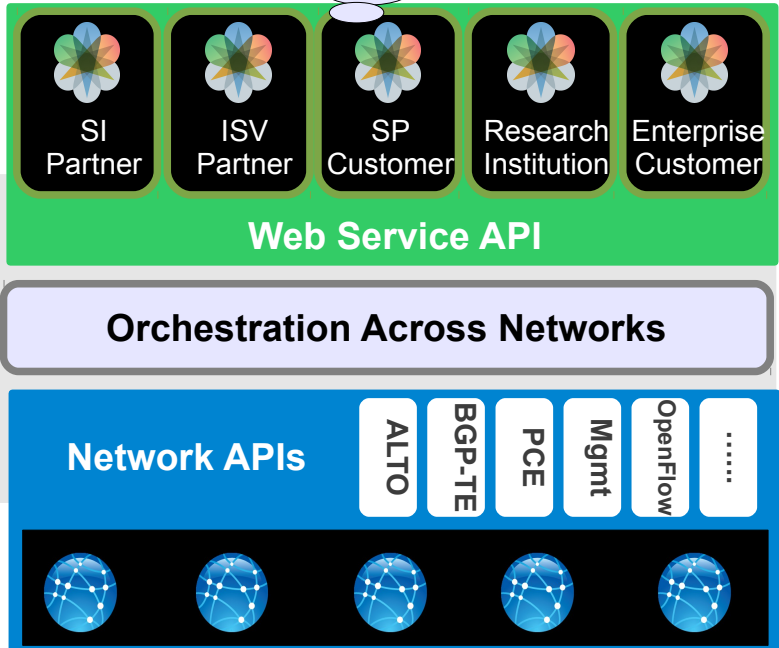
PROGRAMMABILITY FOR NETWORK ABSTRACTION



THE APPLICATION WORLD



- Real-time topology understanding (ALTO, BGP-TE)
- Steering traffic through optimal paths (PCE)
- Selecting specific traffic (OpenFlow)
- New touch points: gateways, billing collectors, service appliances, CDN, DPI



THE NETWORK WORLD

Bridge the divide from both sides: Augment and Enable

WHAT IS OMOTENASHI?

- **The heart of Japanese hospitality.**
- **The host (Service provider) anticipates the needs of the guests (Users) in advance and offers a pleasant service that guests (Users) don't expect.**
- **“Omotenashi” has a similar meaning to hospitality in English, but it suggests a deeper part of the human consciousness.**



IMPROVING USER EXPERIENCE BY PROGRAMMING THE NETWORK

**SOFTWARE
DEFINED
NETWORKING**

**PROGRAMMABLE
NETWORKING**

**Real Problems.
Real Attention.
Real Contribution to User.**

Bringing together and influence the user experience

Bi-directional interaction between App & Net

JUNOS VEE SDK

JUNOS SPACE SDK

JUNOS SDK

UNLEASH THE POTENTIAL!
Today the two worlds are not interlocked

**Application
World**

**Network
World**



Virtual Engine SDK APIs IN JAVA, C, C++, MORE...

Junos Systems



Line card



Appliance



Routing Engines - Control Plane

Control Applications
(built with RE SDK)

UI Extensions
(built with RE SDK)

Service Broker
(native platform)

Services over MOM over IP

VEE Manager
(native platform)

Packet Forwarding Engine - Data Plane

Service Applications
(built with Services SDK)

Service Engines - Services Plane

Virtual Engine Environment - Virtual Plane

Virtual Engine Guest Systems
(Linux, Windows, Junos, etc.)

Junos-integrated Application
(built with VE SDK)

Other Applications

VEE KVM Hypervisor

Host Manager
(native platform)

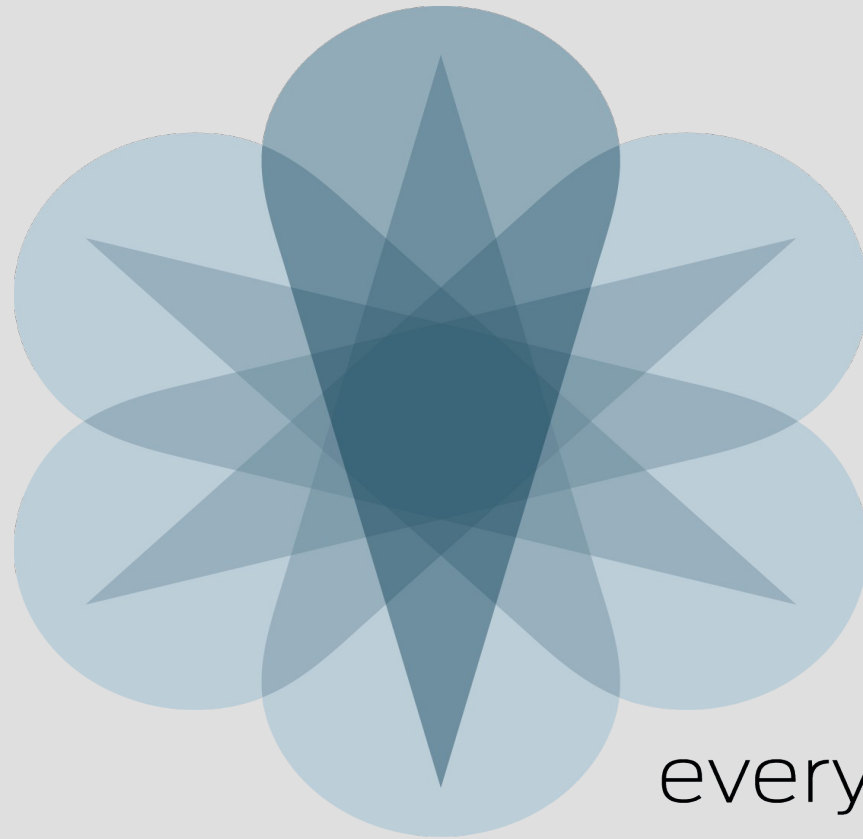
Cluster Manager
(native platform)

SUMMARY

- Further work for developing network virtualization (ITU-T Y.3011)
 - 1) Isolation
 - 2) Programmability
 - 3) Performance
 - 4) Network abstraction
 - 5) Topology awareness and quick reconfiguration,
 - 6) Management, 7) Mobility, 8) Wireless
- Further leveraging the network device and the server device
- Developer ecosystem
 - Shift from vendor driven to developers driven

**THE NEW NETWORK
MUST OPEN TO
SOFTWARE APPS
& DEVELOPERS**





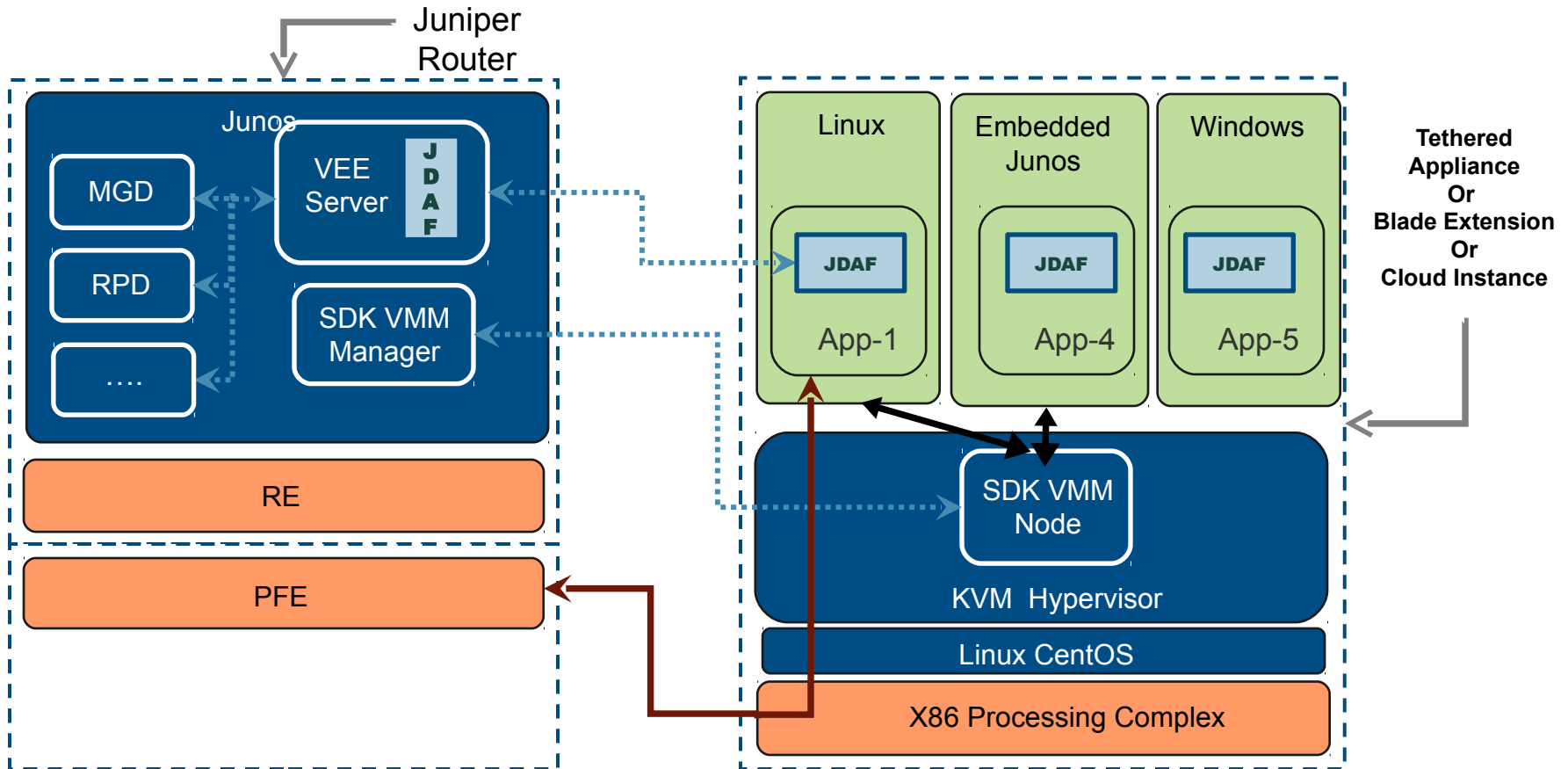
everywhere

Thank you!

jdn@juniper.net

VEE – high level VIEW

VEE allows non-Junos applications to integrate with Junos-based Platforms



↔ Mirrored or Inline Data Traffic

↔ Junos SDK- JDAF

↔ App Specific information

↔ VM creation information

Junos VEE long-term vision

Router acting as the central entity - interacting with a distributed set of applications, which are both existing 3rd party apps as well as VE-SDK Apps that run tethered to the router and deployed in the cloud.

