

プログラマブル・データプレーン時代に向けた ネットワーク・オペレーション・スタック

ONIC Japan 2017

海老澤 健太郎 @ Ponto Networks, Inc.

Ponto Networks, Inc.

Locations

Head Quarters

San Diego (USA)

Development

Tokyo (Japan) + San Jose (USA)

Investors

Ex-executives of Internet and Mobile industry.

アジェンダ

プログラマブル・データプレーンの時代とは？

「現在」

「課題」と「未来」

ネットワーク・オペレーション・スタックの実装例

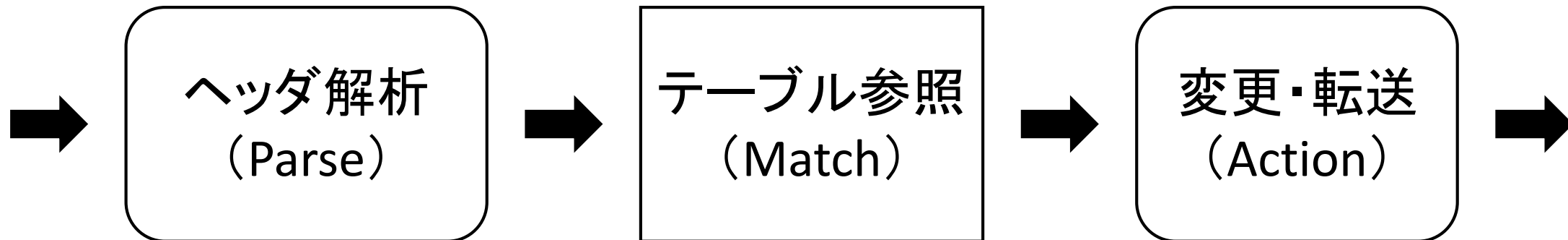
データプレーン

とは？

イーサネットヘッダを解析

宛先アドレスをKEYに
MACテーブルを参照

学習済みポートへ転送

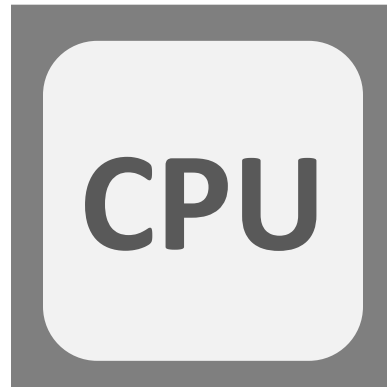


パケット処理 パイプライン

最も柔軟な

データプレーン

とは？



どんな処理も実装できる(プログラマブル)

ハードウェア・データプレーン(ASIC)を使う利用

	CPU	ASIC
ピーク性能	300Gbps	6.5Tbps
Gbps単価	10,000円	461円

※ 数字は概算(桁感の例)です

※ CPU/ASIC 1個搭載のサーバー/スイッチ価格を300万円として比較。

※ CPU: XEON® PLATINUM 8180 (PCIe 3 x 48 = 300Gbps)

※ ASIC: Barefoot Tofino (100GbE x 65 port) を Layer 2/3 switch として利用

※ 数字は概算(桁感の例)です。CPU/ASIC 1個搭載のサーバー/スイッチ価格を300万円として比較。

機能追加時に必要な開発体制とコスト

	CPU	ASIC
ピーク性能	300Gbps	6.5Tbps
Gbps単価	10,000円	461円
開発体制	数人	数十人
開発コスト (人件費除く)	数百万円	数億円



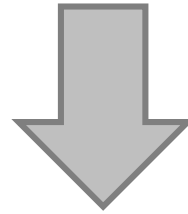
※ 数字は概算(桁感の例)です。CPU/ASIC 1個搭載のサーバー/スイッチ価格を300万円として比較。

ASICのスケールラビリティ + CPUの開発コスト

	CPU	ASIC	プログラマブル ASIC
ピーク性能	300Gbps	6.5Tbps	6.5Tbps
Gbps単価	10,000円	461円	461円
開発体制	数人	数十人	数人
開発コスト (人件費除く)	数百万円	数億円	数百万円

※ 数字は概算(桁感の例)です。CPU/ASIC 1個搭載のサーバー/スイッチ価格を300万円として比較。

「プログラマブル・データプレーン」の時代



イノベーションがスケールする時代

試作・試行コストが小さくなる ⇒ サーバーで起きたイノベーションをネットワークへ
コスト & 性能の良いプラットフォームでイノベーションが実現

個人で開発への参加が可能 ⇒ オープンな活動の活性化
より多くの人新しい試みに参加できるように

プログラマブル・データプレーンを用いた試行(例)

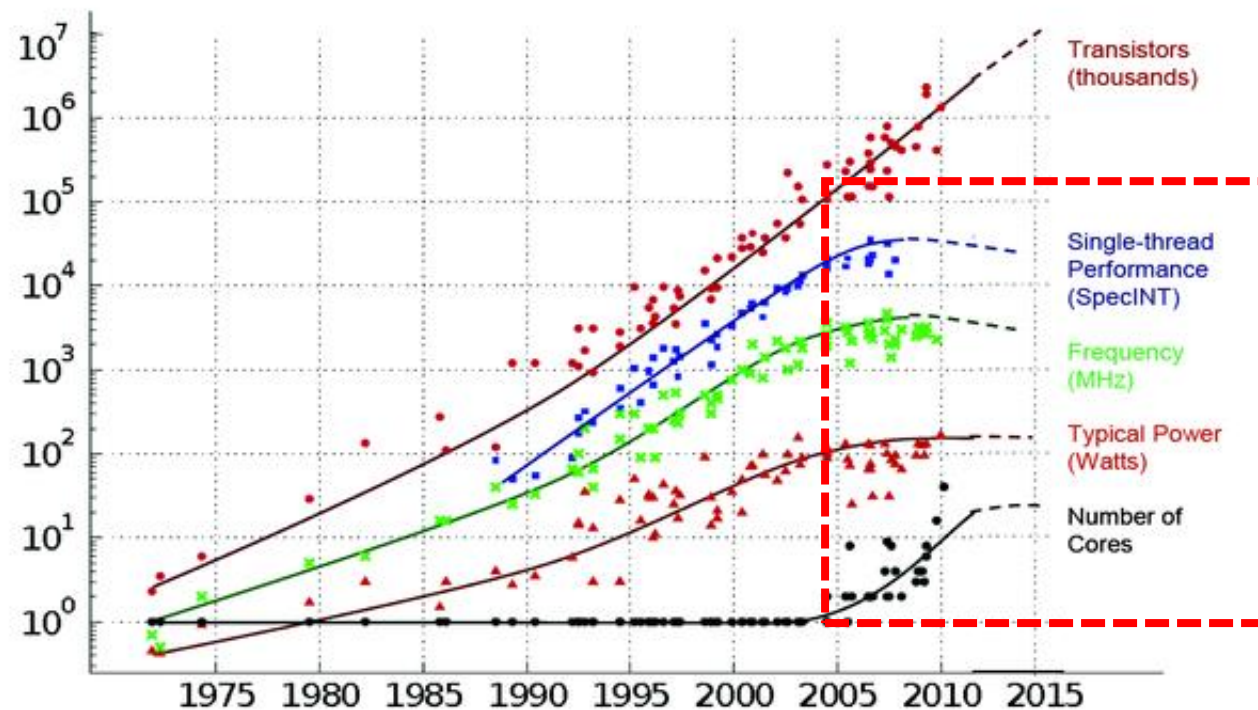
- In-band Network Telemetry (INT)
 - パケットがマッチしたルールやQueueの情報をパケット自身に含める事により、SNMP等ではモニタリングできなかつた粒度でネットワークの状況を把握
 - 遅延や特定アプリケーションの性能劣化原因を特定
 - トラフィックエンジニアリングへの利用
- Tbps Class Load Balancer
 - ラックあたり Tbps クラスのトラフィックに対応したLoad Balancerを低コストで構築
 - トラフィック種別により振り分けに利用するフィールドを変更
- Pre-processing & Content aware routing for Massive IoT
 - 様々な Client からのデータをサーバーが処理しやすいように前処理
 - データ種別に応じた計算。bit -> Byte データ展開。
 - ペイロード中のIDに応じたサーバーへの振り分け、ストレージへの直接保存
 - アプリケーション毎のペイロード・データフォーマットの定義

プログラマブル・データプレーンが必要とされる もう1つの理由

ムーアの法則の終焉

待っていても性能は向上しない時代
いかに高価なCPUをオフロードするか？

35 YEARS OF MICROPROCESSOR TREND DATA



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

出典 : <https://www.karlsruh.net/2015/06/40-years-of-microprocessor-trend-data/>

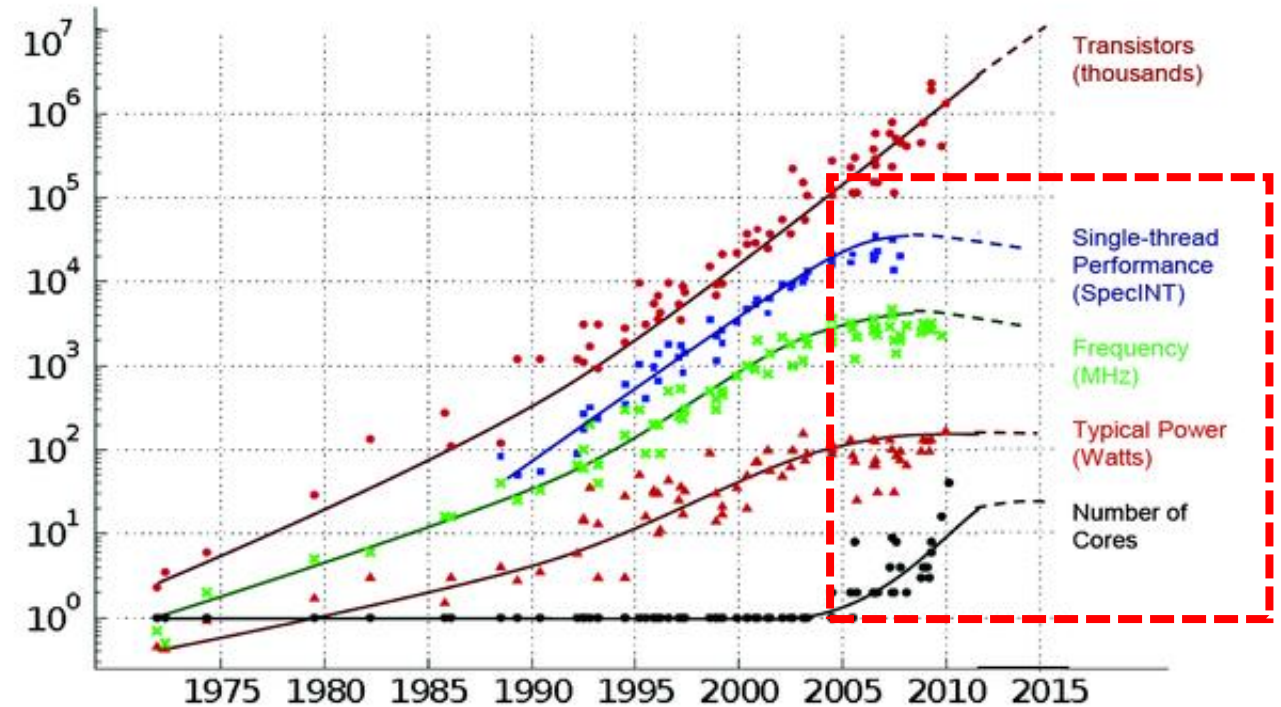
プログラマブル・データプレーンが必要とされる もう1つの理由

ムーアの法則の終焉

待っていても性能は向上しない時代
いかに高価なCPUをオフロードするか？

「ユースケース最適」な
ハードウェアの選択が必要に
(ASIC / NPU / FPGA)

35 YEARS OF MICROPROCESSOR TREND DATA



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

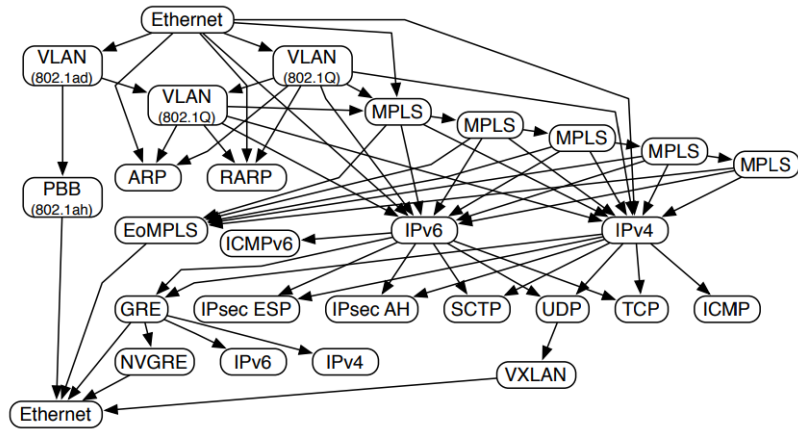
出典 : <https://www.karlsruh.net/2015/06/40-years-of-microprocessor-trend-data/>

プログラマブル・データプレーンの現在

ハードウェア・データプレーンに関して

「プログラマブル・データプレーン」で「プログラム」できること

ヘッダフォーマットの定義
パースグラフの構築



マッチフィールドの定義
テーブルタイプの定義

(Exact / Masked)

- MAC address
- IPv4 address
- proto + TCP ports
- (any header fields)

アクションの定義
フィールド操作ロジック

- drop
- forward
- copy
- push / pop
- add(+) sub(-) multiple(*)
- bit shift (<<) (>>)



プログラマブル・データプレーン(ハードウェア)の現在

メーカー・製品名称	タイプ	開発環境	
Cavium XPliant	ASIC	XDK	製品出荷実績多数(クラウド事業者) OEM: Arista, Brocade
Barefoot Tofino	ASIC	P4	大規模事業者を中心とした限定出荷 検証・開発用筐体は一般入手可能
Netronome NFP	NPU	C-based	SmartNIC 1枚から広く入手可能 OEM多数(非公開・アプライアンス製品等)
NetFPGA (Xilinx)	FPGA	SDNet	NetFPGAは研究・教育目的が中心 Xilinx FPGA 搭載ボード(+SDNet)としても入手可能

※ 代表的な製品のみ記載。他多数のメーカーもプログラマブル ASIC/NPU/FPGA リリースを予定している。

amazon.co.jp prime

パソコン・周辺機器 Edgecore AS7512

Amazonポイント: 27

マイストア タイムセール ギフト券 Amazonで売る

パソコン・周辺機器 セール&キャンペーン パソコン本体 タブレットPC アクセサリ・サプライ

1件の結果 パソコン・周辺機器ストア: "Edgecore AS7512"

カテゴリ

すべてのカテゴリ

パソコン・周辺機器ストア

無線LAN・ネットワーク機器 (1)

絞り込み

配達日

本日中にお届け

明日お届け

コンディション

新品 (1)

中古品

新着



Edgecore AS7512ホワイトボックススイッチ(Cavium Xpliant CNX88091 3.2 Tbps)

Edge-Core Networks Corporation

新品 (1 出品)

<https://www.amazon.co.jp/Edgecore AS7512> で検索

Cavium Xpliant
CNX88091 3.2Tbps



COLFAX DIRECT

HPC and Data Center Gear

[Home](#)

[AboutUs](#)

[ContactUs](#)

[Search](#)

[Checkout](#)

[MyAccount](#)

Browse by Category

[Adapters](#)

[Switches](#)

[Cables](#)

[NVMe SSDs](#)

[SDN Appliance](#)

[Gateways](#)

[Transceivers](#)

[Accessories](#)

[Software](#)

[Warranty / Support](#)

Netronome Agilio-CX Single-Port 40 Gigabit Ethernet Intelligent Server Adapter - Part ID: ISA-4000-40-1-2

SKU: ISA-4000-40-1-2

Manufacturer: [Netronome](#)

Single-port 40GbE, PCIe Gen3 x8, 2GB of onboard memory

[More details...](#)

Price: \$555.00

Agilio-CX OVS Software and Support for 1 Year:

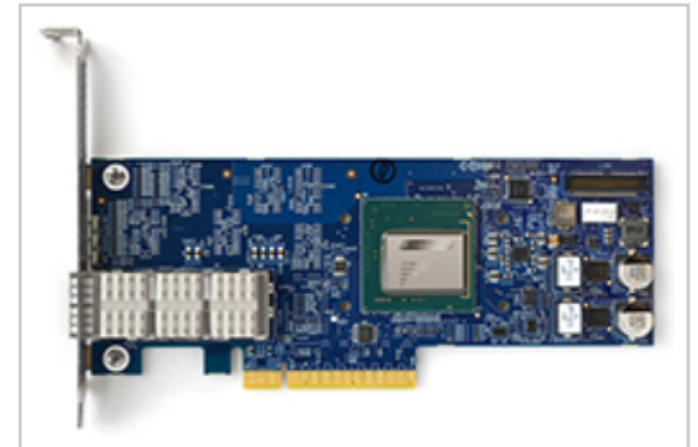
Select One

1

[Add To Cart](#)

[Add to Wishlist](#)

[Tell a Friend](#)



(最もハードルが高い)

ハードウェアは入手可能に

(最もハードルが高い)
ハードウェアは入手可能に

プログラマブル・データプレーン 「課題」と「未来」

プログラマブル・データプレーンの課題（1）

メーカー・製品名称	タイプ	開発環境
Cavium XPliant	ASIC	XDK
Barefoot Tofino	ASIC	P4
Netronome NFP	NPU	C-based
NetFPGA (Xilinx)	FPGA	SDNet

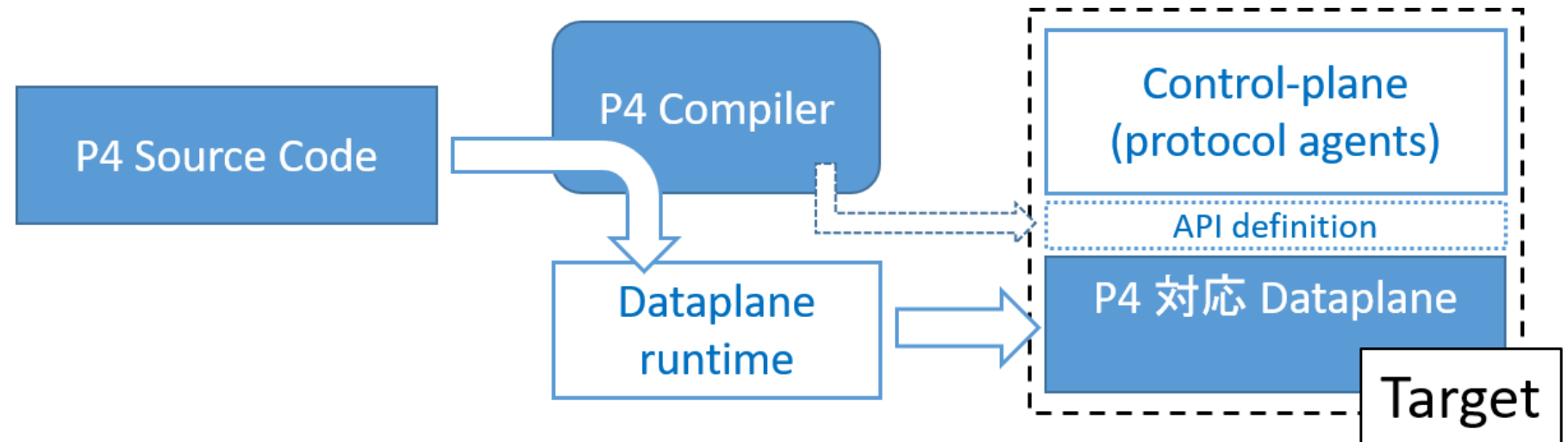
「自由にプログラム」
からは、やや遠い

言語もツールも異なる
過去の学習が生かされない
SDK入手に別途契約が必要な場合も

P4: 汎用データプレーン・プログラミング言語の登場

“Programming Protocol-Independent Packet Processors”

P4 Source Code	パケット処理パイプラインの定義 パーサーやテーブル、アクション、など
P4 Compiler	P4をTarget上で実行可能な形式にコンパイル Target毎に提供される
Target (P4対応Dataplane)	P4 Dataplane runtime に従いパケットを処理 Hardware: ASIC, NPU, FPGA Software: CPU



ヘッダ定義

```

16 // header defintitions
17
18 header_type ethernet_t {
19     fields {
20         dstAddr    : 48;
21         srcAddr    : 48;
22         etherType  : 16;
23     }
24 }
25
26 header_type ipv4_t {
27     fields {
28         version    : 4;
29         ihl        : 4;
30         diffserv   : 8;
31         totalLen   : 16;
32         identification : 16;
33         flags      : 3;
34         fragOffset : 13;
35         ttl        : 8;
36         protocol   : 8;
37         hdrChecksum : 16;
38         srcAddr    : 32;
39         dstAddr    : 32;

```

パーサー定義

```

22 parser start {
23     return parse_ethernet;
24 }
25
26 parser parse_ethernet {
27     extract(ethernet);
28     return select(latest.etherType) {
29         ETHERTYPE_IPV4 : parse_ipv4;
30         default        : ingress;
31     }
32 }
33
34 parser parse_ipv4 {
35     extract(ipv4);
36     return select(latest.protocol) {
37         IP_PROTOCOLS_TCP : parse_tcp;
38         IP_PROTOCOLS_UDP : parse_udp;
39         default          : ingress;
40     }

```

テーブル定義

```

103 table tbl_rewrite_dstAddr {
104     reads {
105         standard_metadata.ingress_port : exact;
106         ipv4.dstAddr    : exact;
107         ipv4.srcAddr    : exact;
108         ipv4.protocol   : exact;
109         lb_metadata.srcL4Port : exact;
110         lb_metadata.dstL4Port : exact;
111         lb_metadata.hash : exact;
112     }
113     actions {
114         rewrite_dstAddr;
115         rewrite_dstAddr_mac;
116         rewrite_dstAddr_ipv4;
117     }
118 }

```

パイプライン定義

```

120 control ingress {
121     apply(tbl_lb_calc_hash);
122     apply(tbl_rewrite_dstAddr);
123     apply(tbl_forward_packet);
124 }

```

- C 構造体のようにプロトコルヘッダを定義
- パーサー、テーブル、パイプラインを簡単な構文で記述可能

P4 対応状況

メーカー・製品名称	タイプ	開発環境	P4 対応状況(入手状況)
Cavium XPliant	ASIC	XDK	対応予定 (エンドースメント・プレスリリース有り)
Barefoot Tofino	ASIC	P4	Capilano SDE (製品購入者＋契約)
Netronome NFP	NPU	C-based	Agilio P4C SDK (製品購入者)
NetFPGA (Xilinx)	FPGA	SDNet	P4→NetFPGA / P4-SDNet (登録必要)

Netronome: <https://www.netronome.com/products/datapath-programming-tools/>

NetFPGA: <https://github.com/NetFPGA/P4-NetFPGA-public/wiki>

<https://github.com/p4lang>

The screenshot shows the GitHub organization page for p4language. The page features a header with navigation links (Pull requests, Issues, Marketplace, Explore) and a search bar. Below the header, the organization name 'p4language' is displayed with a profile picture. The page is divided into sections for 'Repositories' (21) and 'People' (4). The 'Pinned repositories' section lists three repositories:

- behavioral-model**: Rewrite of the behavioral model as a C++ project without auto-generated code (except for the PD interface). Language: C++. Stars: 68. Forks: 81.
- tutorials**: P4 language tutorials. Language: P4. Stars: 61. Forks: 71.
- p4c**: P4_16 prototype compiler. Language: P4. Stars: 55. Forks: 48.

Red annotations with arrows point to these repositories:

- A box labeled 'P4 対応 Software Switch (サンプルCLI付き)' points to the **behavioral-model** repository.
- A box labeled 'P4 チュートリアル&サンプル (P4 source code, Protocol agent ...)' points to the **tutorials** repository.
- A box labeled 'P4 コンパイラ' points to the **p4c** repository.

P4 ソフトウェア実装 | ビルド・実行手順サンプル

<https://www.slideshare.net/kentaroebisawa/how-to-run-p4-bmv2>

プログラマブル・データプレーンの課題（2）

データプレーンをどうコントロールするか？

プログラマブル・データプレーン時代に必要な
「ネットワーク・オペレーション・スタック」とは？

「ネットワーク・オペレーション・スタック」に何が求められるか？

異なるデータプレーン・ハードウェアへの対応

パケット処理パイプラインの管理

- (Match/Action Table)

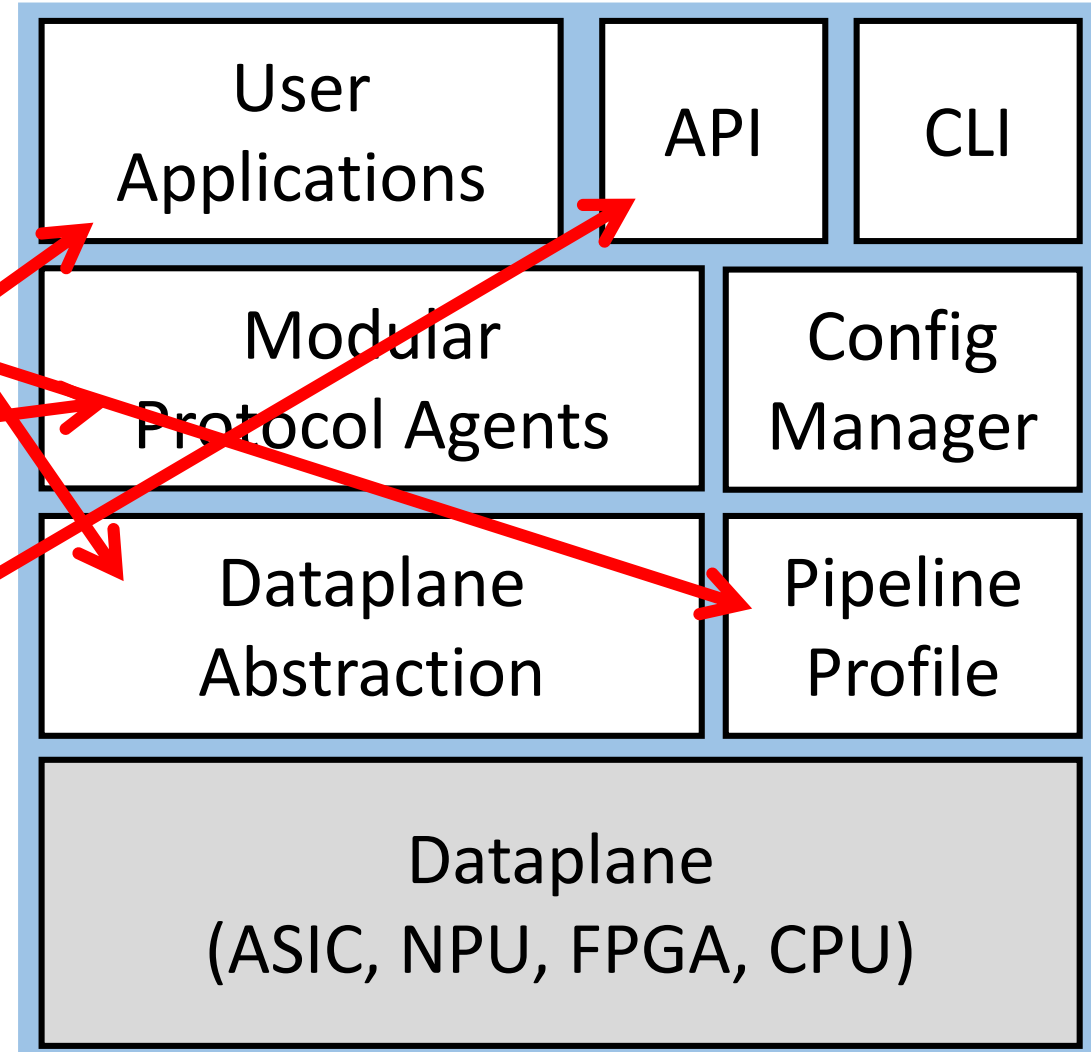
プロトコル・エージェントの入れ替え

ユーザーアプリケーションの動作

- シェルアクセス
- サーバー同様の開発ツールチェーン

アプリケーションのためのAPI

- 都度コンパイルでなく、gRPC/REST等APIで制御可能



Open Network OS の現状

	Dataplane Abstraction	Pipeline Profile	Modular Protocol Agents	User Apps	APIs for Apps
OpenSwitch SnapRoute + Dell	○	×	○	○	×
SONiC	○	×	○	○	×
ONL Open Network Linux	—	×	—	—	—

プログラマブル・データプレーン時代の ネットワーク・オペレーション・スタックの実装例

PontOS² コンセプト & アーキテクチャ

 Application
Friendly API

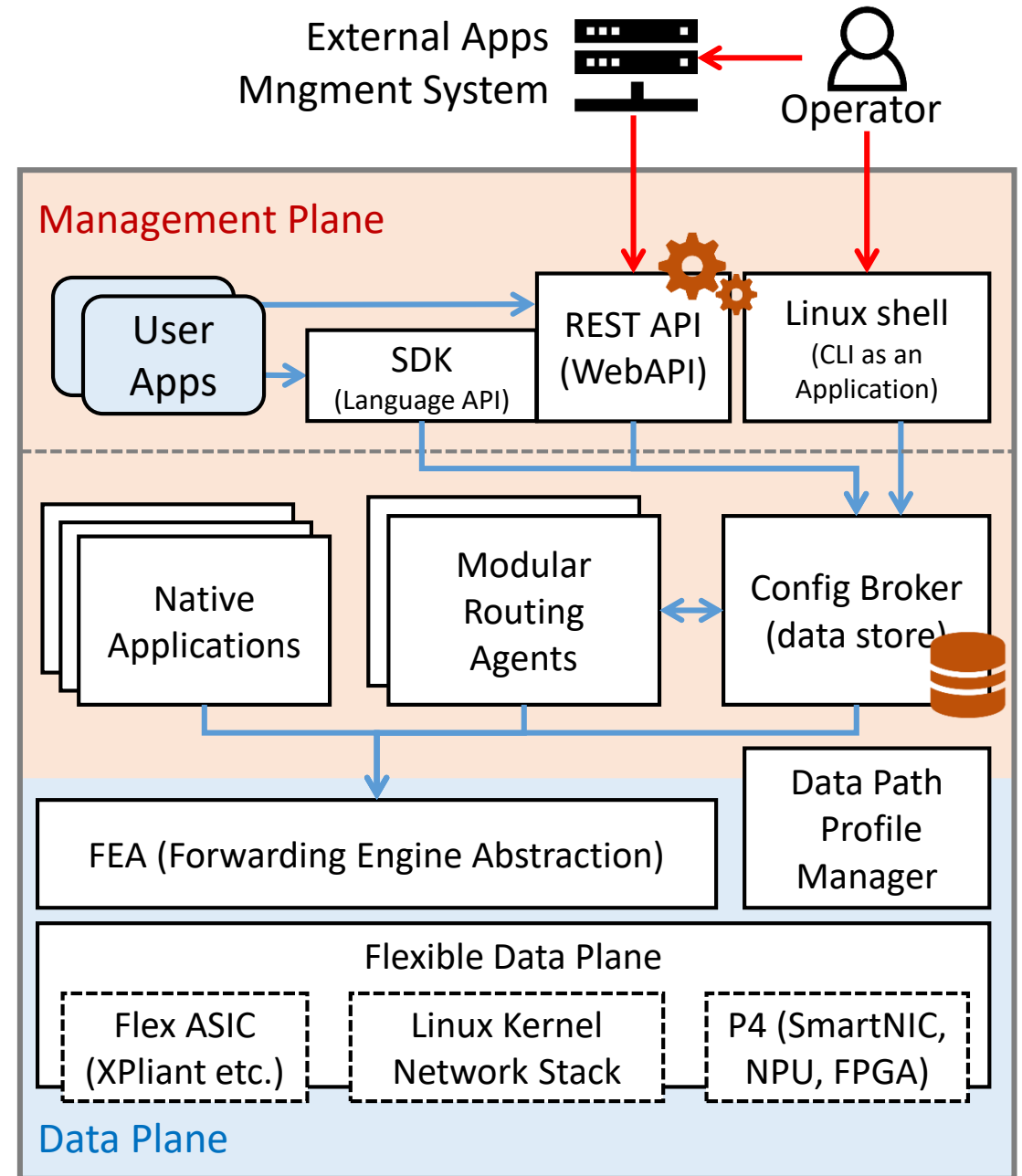
ネットワークプログラマビリティを実現する
アプリケーション・フレンドリーな開発環境

Open Control Plane


オープンソースにより構成された
ビルディング・ブロック

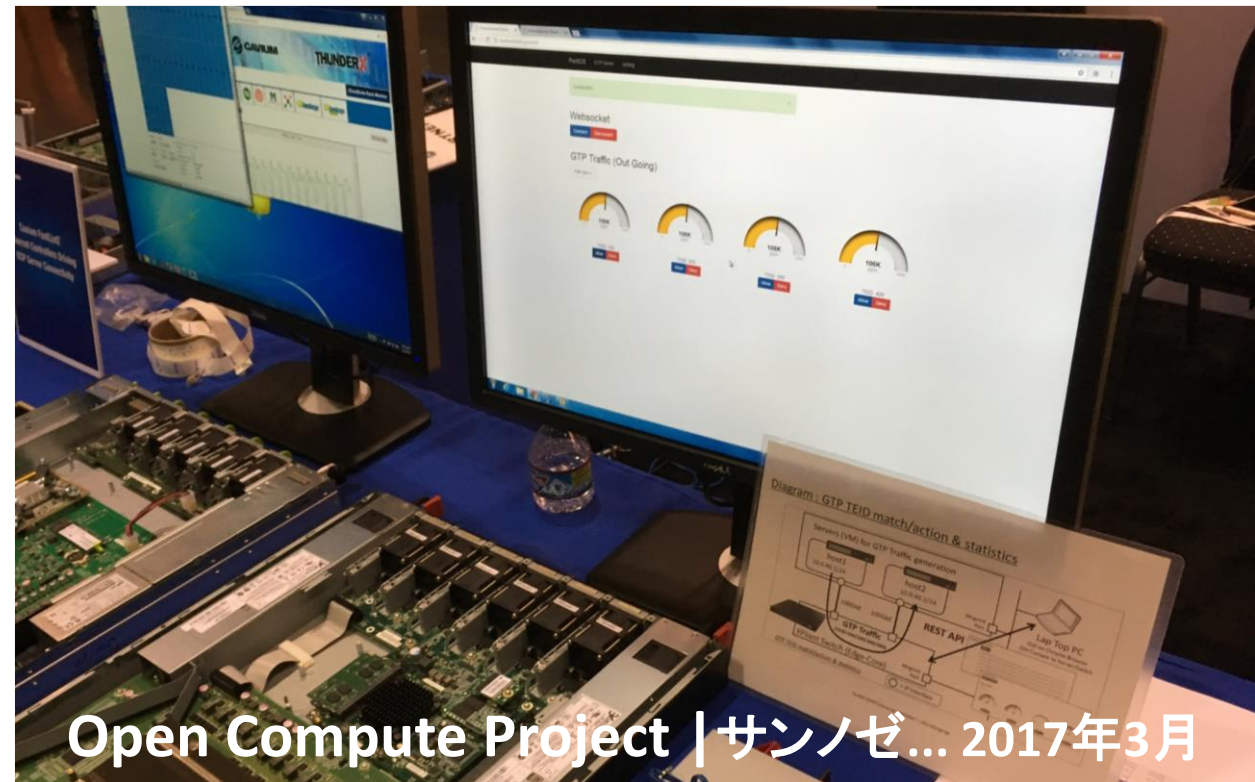
 Flexible
Data plane

プロファイル選択により入れ替え可能な
パケット処理エンジン

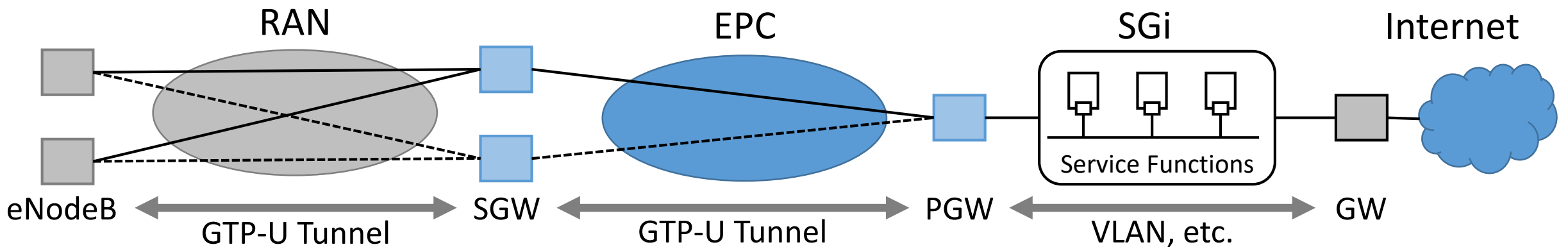




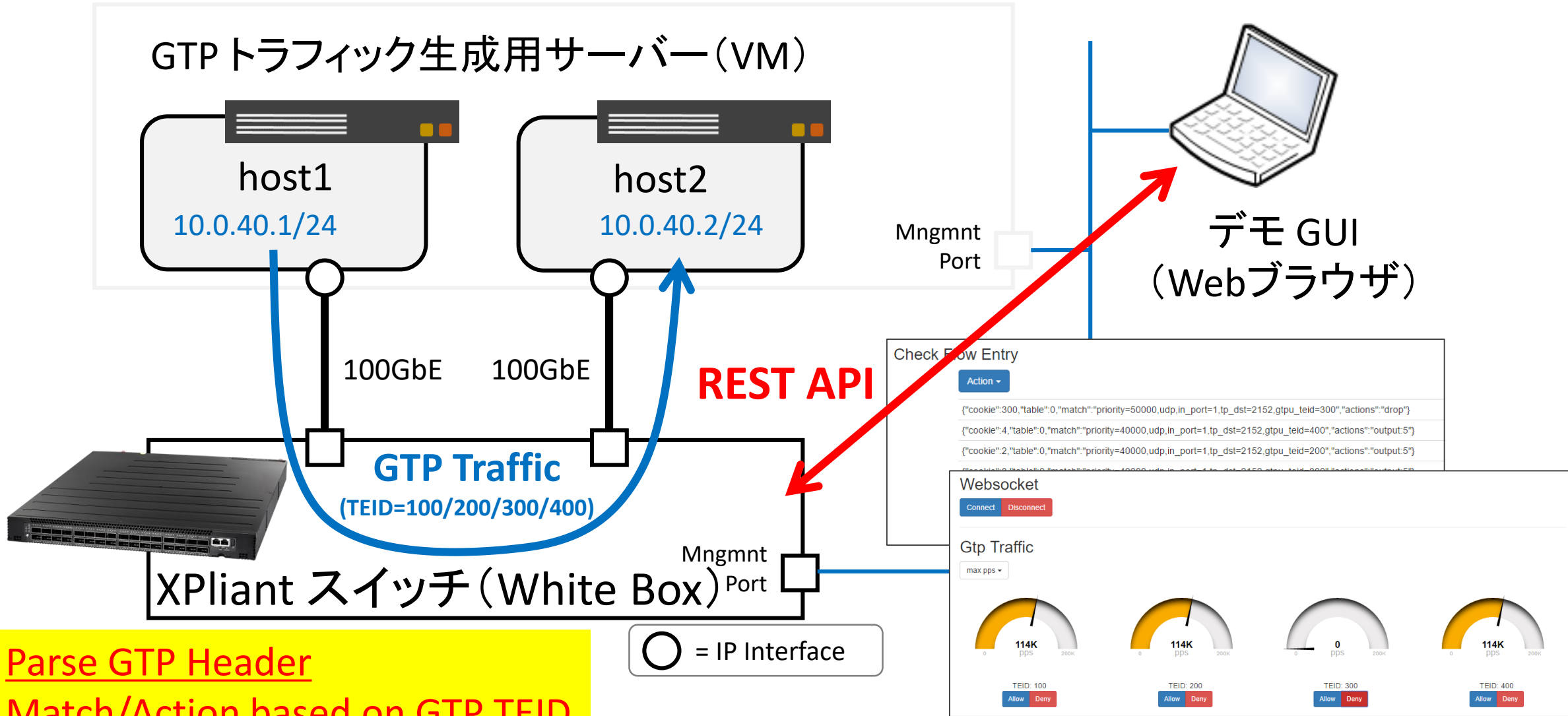
Computex Taipei | 台湾・台北 ... 2017年6月



Open Compute Project | サンノゼ... 2017年3月



GTP TEID match/action & statistics (デモ構成図)



- Parse GTP Header
- Match/Action based on GTP TEID
- stats via REST API

PontOS² Implementation (実装)

Zebra 2.0

Open Source Network Stack

Fresh rewrite of Zebra/Quagga

Data Plane agnostic NetOS Stack

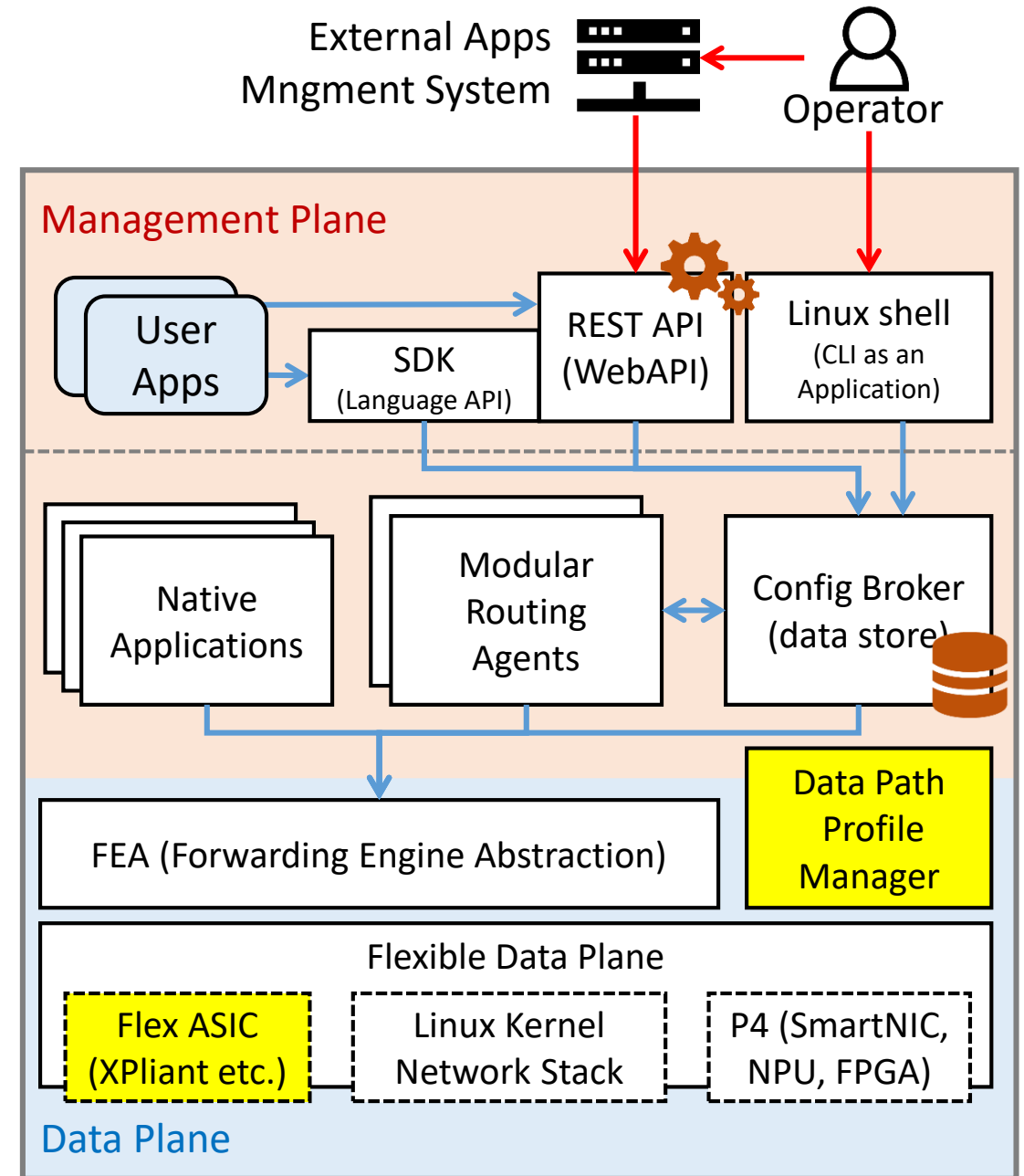
PontOS²

Seamless integration with multiple

Proprietary Data Plane platforms

Performance and quality control

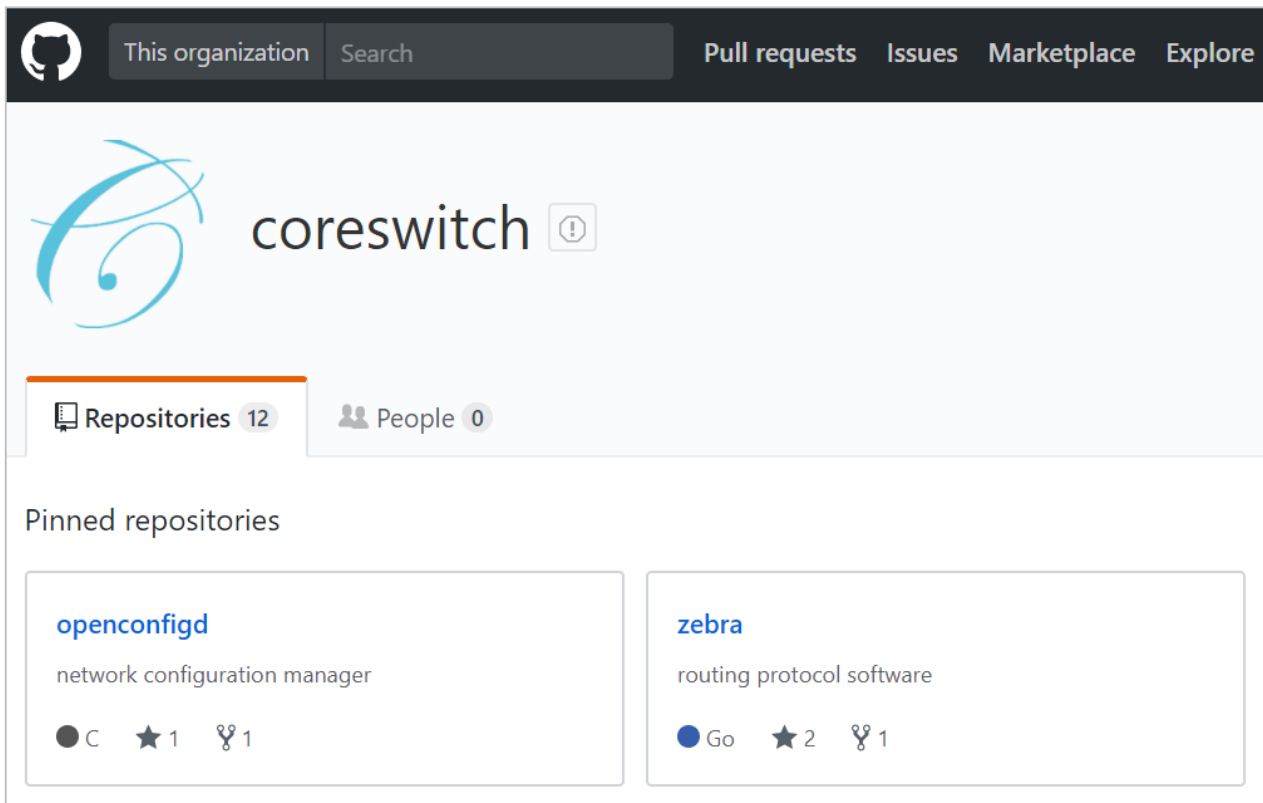
for service providers



Zebra 2.0 on GitHub !!

<https://github.com/coreswitch/zebra>

<https://github.com/coreswitch/openconfigd>



This organization Search Pull requests Issues Marketplace Explore

coreswitch

Repositories 12 People 0

Pinned repositories

- openconfigd**
network configuration manager
C ★ 1 🍴 1
- zebra**
routing protocol software
Go ★ 2 🍴 1

Zebra 2.0 Installation

- Install openconfigd

```
$ go get github.com/coreswitch/openconfigd/openconfigd
```

- Install CLI

```
$ go get github.com/coreswitch/openconfigd/cli_command
```

```
$ cd $GOPATH/src/github.com/coreswitch/openconfigd/cli
```

```
$ ./configure; make
```

```
$ sudo make install
```

```
$ cd $GOPATH/src/github.com/coreswitch/openconfigd/bash_completion.d
```

```
$ sudo cp cli/etc/bash_completion.d/
```

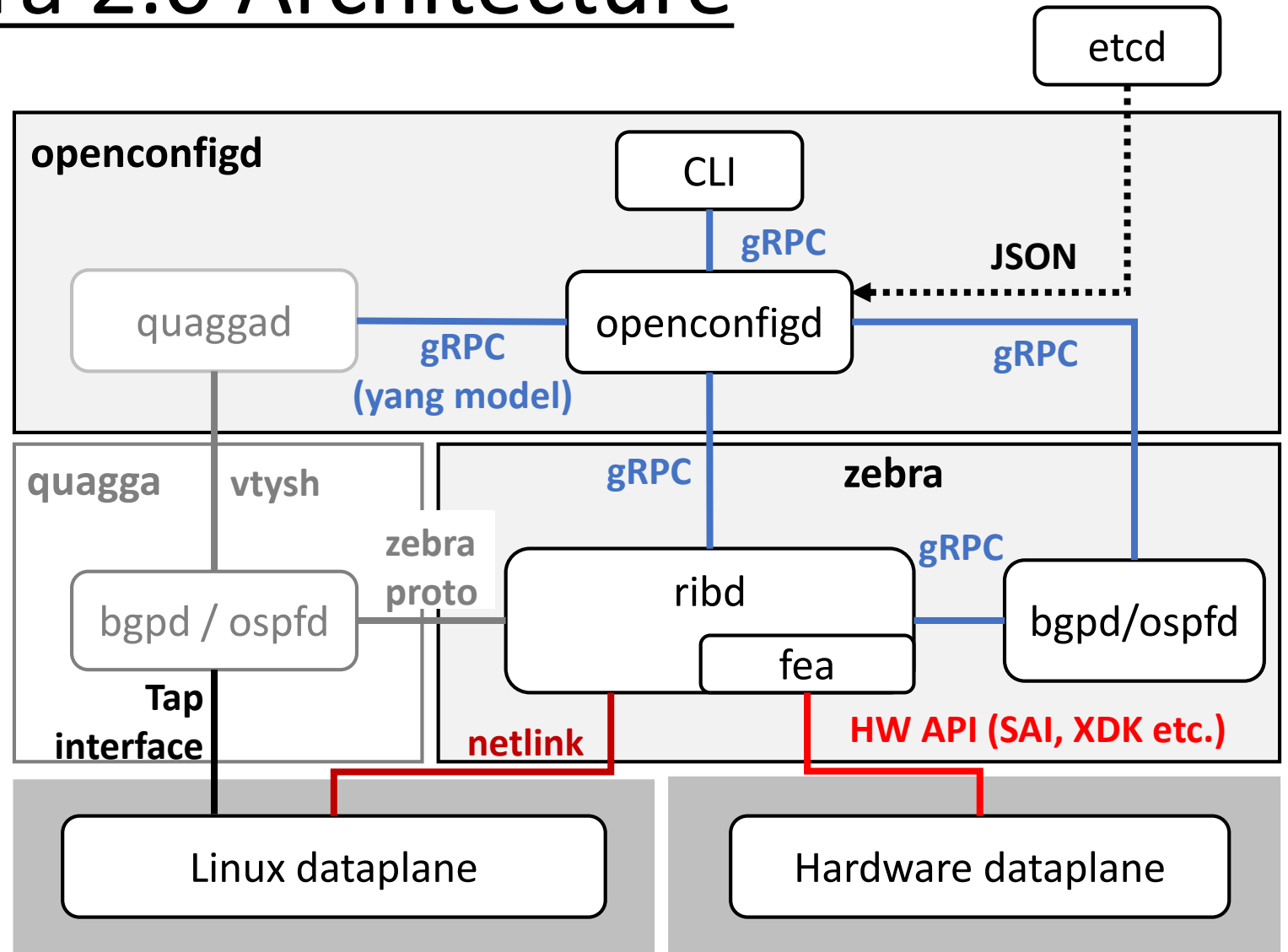
- Install zebra

```
$ go get github.com/coreswitch/zebra/rib/ribd
```

Zebra 2.0 Architecture

Written from scratch in Go

- openconfigd
 - configuration system
 - yang model
 - CLI (Junos like)
 - etcd for scalability
- zebra/ribd
 - dataplane management (ex: FIB)
- zebra/fea
 - multiple dataplane support
 - link/port, bridge domain etc.
- New bgpd/ospfd
 - multi-core support
- quaggad & zebra protocol
 - for backward compatibility



Zebra2.0 + OpenConfigd + Quagga ospfd/bgpd On LXC Containers

<https://github.com/coreswitch/zebra/blob/master/docs/quagga-lxc.md>

```

+-----+ +-----+
| host1 | | host2 |
+---+---+ +---+---+
      |           |
+---+-----+---+
| lxcbr0 10.0.3.1/24 |
+-----+

```

- host1 IP address
 - eth0 : 10.0.3.61/24
 - lo: 10.10.0.1/32, 10.10.10.1/31
- host2 IP address
 - eth0 : 10.0.3.62/24
 - lo: 10.10.0.2/32, 10.10.10.2/31

```

host1>show ip ospf route
===== OSPF network routing table =====
N   10.0.3.0/24          [10] area: 0.0.0.0
                        directly attached to eth0
N   10.10.10.1/32       [10] area: 0.0.0.0
                        directly attached to lo

===== OSPF router routing
===== OSPF external routin

```

```

host1>show ip bgp
BGP table version is 0, local router ID is 10.0.3.61
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.10.0.1/32	0.0.0.0	0		32768	i
*> 10.10.0.2/32	10.0.3.62	0		0	65002 i

```

host1>show ip route
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area

K       0.0.0.0/0 via 10.0.3.1, eth0
C       10.0.3.0/24 is directly connected eth0
C       10.10.0.1/32 is directly connected lo
B       10.10.0.2/32 [200/0] via 10.0.3.62
C       10.10.10.1/32 is directly connected lo
C       127.0.0.0/8 is directly connected lo

```

Zebra 2.0 (future roadmap)

- Basic routing/switching features (ACL, NAT etc.)
- New Protocols
 - Segment Routing (SRv6)
- Forwarding Engine Abstraction
 - ASIC support (via SAI)
 - P4 dataplane (via SAI or P4-PI)
- New Protocol Agents
 - BGP, OSPFv2, OSPFv3, IS-IS



データプレーンに自由を

スケーラブルなデータプレーンを「個人」がプログラムできる時代に

データプレーンにイノベーションを

~~プログラマブルになると何ができるの？~~
(昨日は) 想像もできなかった事を
何度も試行できるプラットフォーム