ホワイトボックスハードウェアを用いたルータ開発の課題

2019年10月15日(火)　MPLS JAPAN 2019

KDDI株式会社　丹羽朝信

Tomorrow, Together
KDDI
おもしろいほうの未来へ。
au

自己紹介

丹羽 朝信（Niwa Tomonobu）

2013/4/1~

KDDI システム開発センター

- NW運用支援システム開発

2014/4/1~

KDDI総合研究所

- NFVやMECの研究開発

2018/4/1~

KDDI IPネットワーク部

- ホワイトボックスルータ開発

アジェンダ

KDDIがルータ向けソフトウェアを作ってみた話をします。

1. ホワイトボックスとは
2. ルータ開発の背景
3. アーキテクチャ
4. トライアル
5. 開発中の苦労や失敗談
6. まとめ

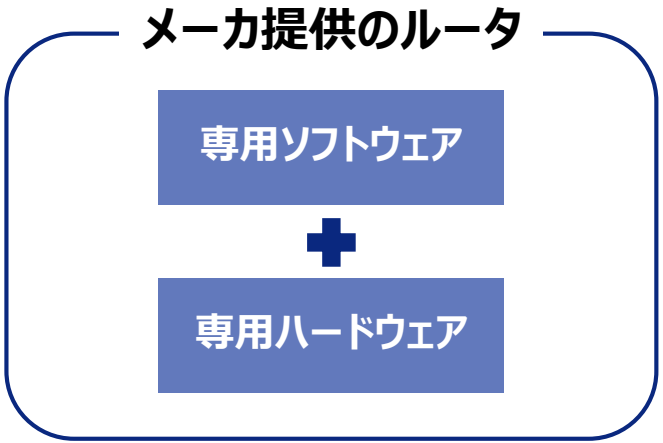
アジェンダ

KDDIがルータ向けソフトウェアを作ってみた話をします。

1. **ホワイトボックスとは**
2. ルータ開発の背景
3. アーキテクチャ
4. トライアル
5. 開発中の苦労や失敗談
6. まとめ

ホワイトボックスとは？（1/2）

- ソフトウェアにバンドルされない、ネットワークプロセッサ(ASIC)を備えるハードウェア
- 所望のソフトウェアに対し、ハードウェアを自由に選択できる世界



垂直統合型

ハードウェアを選択可能

ホワイトボックスとは？（2/2）

■ ASIC/ハードウェア/ソフトウェアの組み合わせから4種類に分類すると…

専用/メーカ提供　オープン/自前

メーカ提供のルータ

Type1

- ソフトウェア
- ハードウェア
- ASIC

Type2

- ソフトウェア
- ハードウェア
- ASIC

ホワイトボックス

Type3

- ソフトウェア
- ハードウェア
- ASIC

Type4

- ソフトウェア
- ハードウェア
- ASIC

アジェンダ

KDDIがルータ向けソフトウェアを作ってみた話をします。

1. ホワイトボックスとは
2. **ルータ開発の背景**
3. アーキテクチャ
4. トライアル
5. 開発中の苦労や失敗談
6. まとめ

ルータ開発の背景（1/3）

ホワイトボックスを使うとコストが下がるらしい。

導入費用は安いけど、トータルで見るとコストアップになるよ！

自由に機能実装できるよ！

SWとHWのインテグレーションって難しい。。。

実際どうなのかよく分からん。。。

よし、Type 4 やってみよう！

Type4

ルータ開発の背景（2/3）

ネットワークプログラミング素人集まる！

KDDIプロジェクトメンバ

SWの相談

HWの相談

社外メンター

ホワイトボックスベンダ

ルータ開発の背景（3/3）

タラルクトス

ロゴ作りました！

ホワイトボックス用のOSなので、
“白”熊をモチーフに！

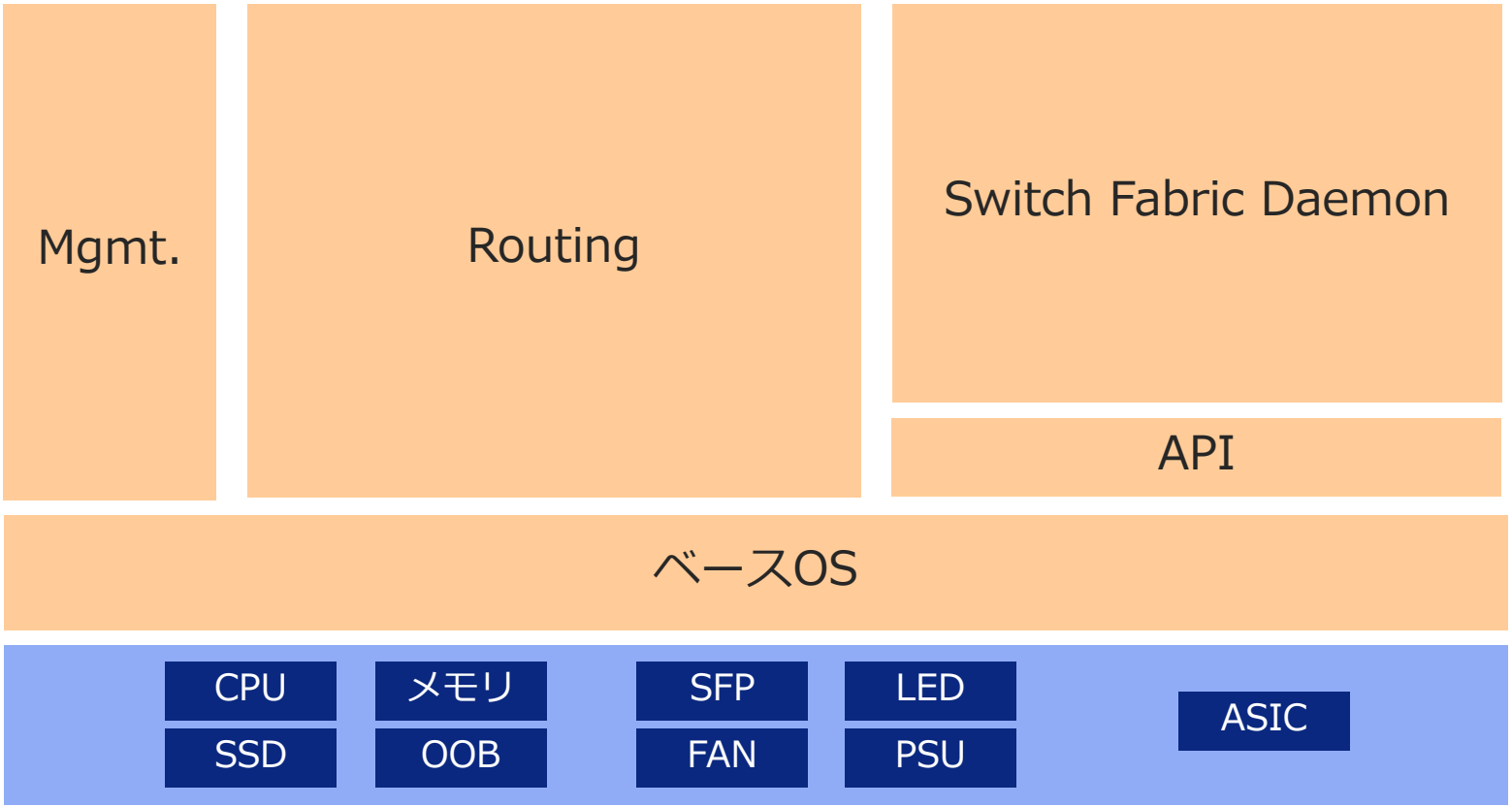
愛称：白熊くん、タラちゃん

アジェンダ

KDDIがルータ向けソフトウェアを作ってみた話をします。

1. ホワイトボックスとは
2. ルータ開発の背景
3. **アーキテクチャ**
4. トライアル
5. 開発中の苦労や失敗談
6. まとめ

アーキテクチャ

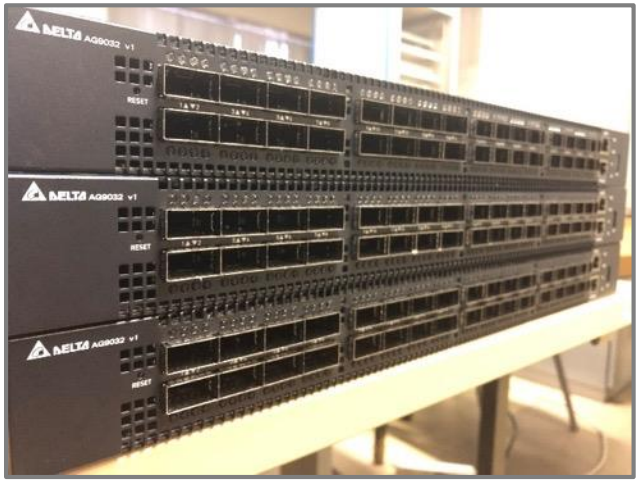


アーキテクチャ：概要（1/5）

Tomahawk搭載
ホワイトボックス
100GE x 32ポート

(Delta: AG9032v1)

OPEN Compute Project®



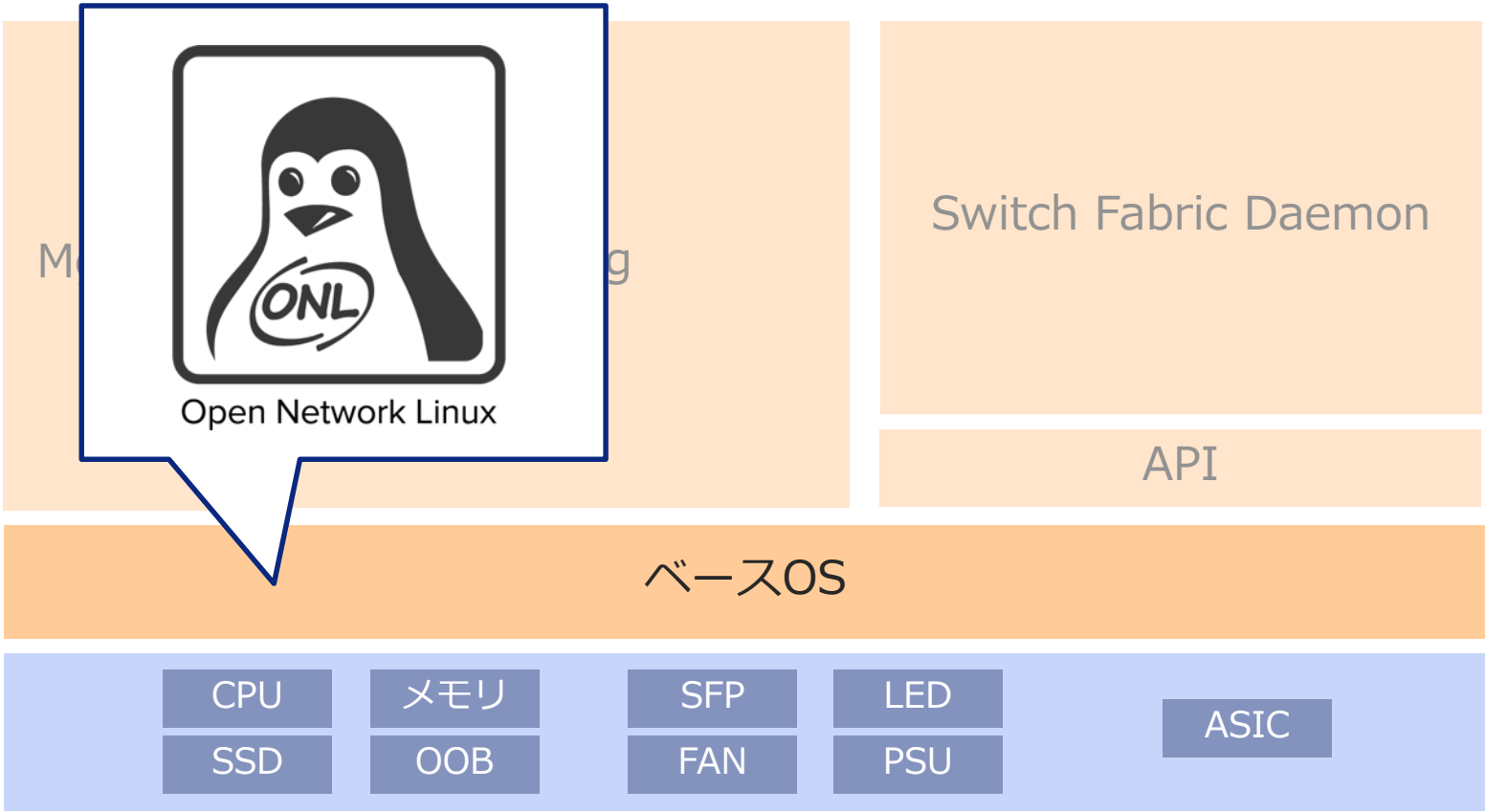
n Fabric Daemon

API

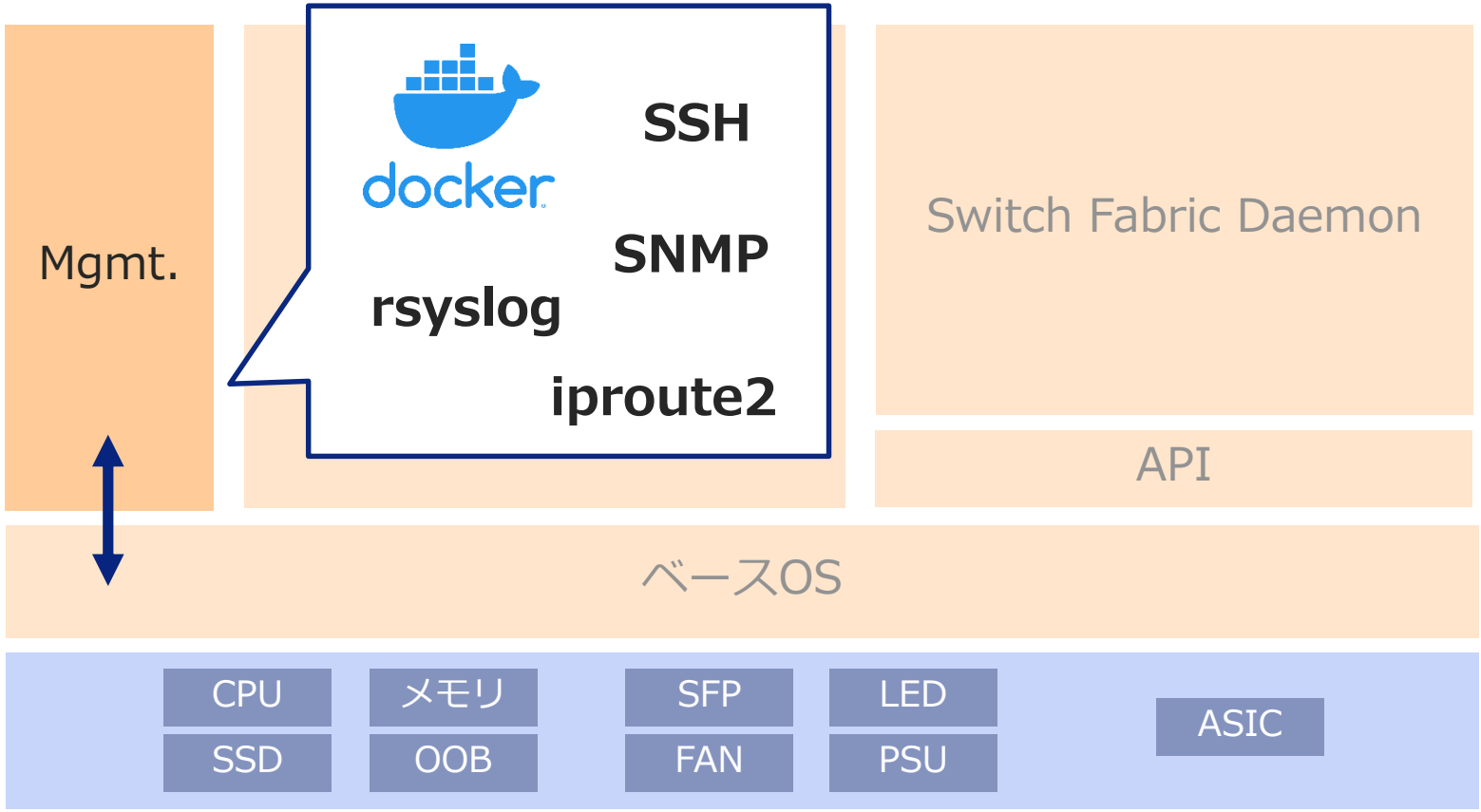
ベースOS

CPU	メモリ	SFP	LED	ASIC
SSD	OOB	FAN	PSU	

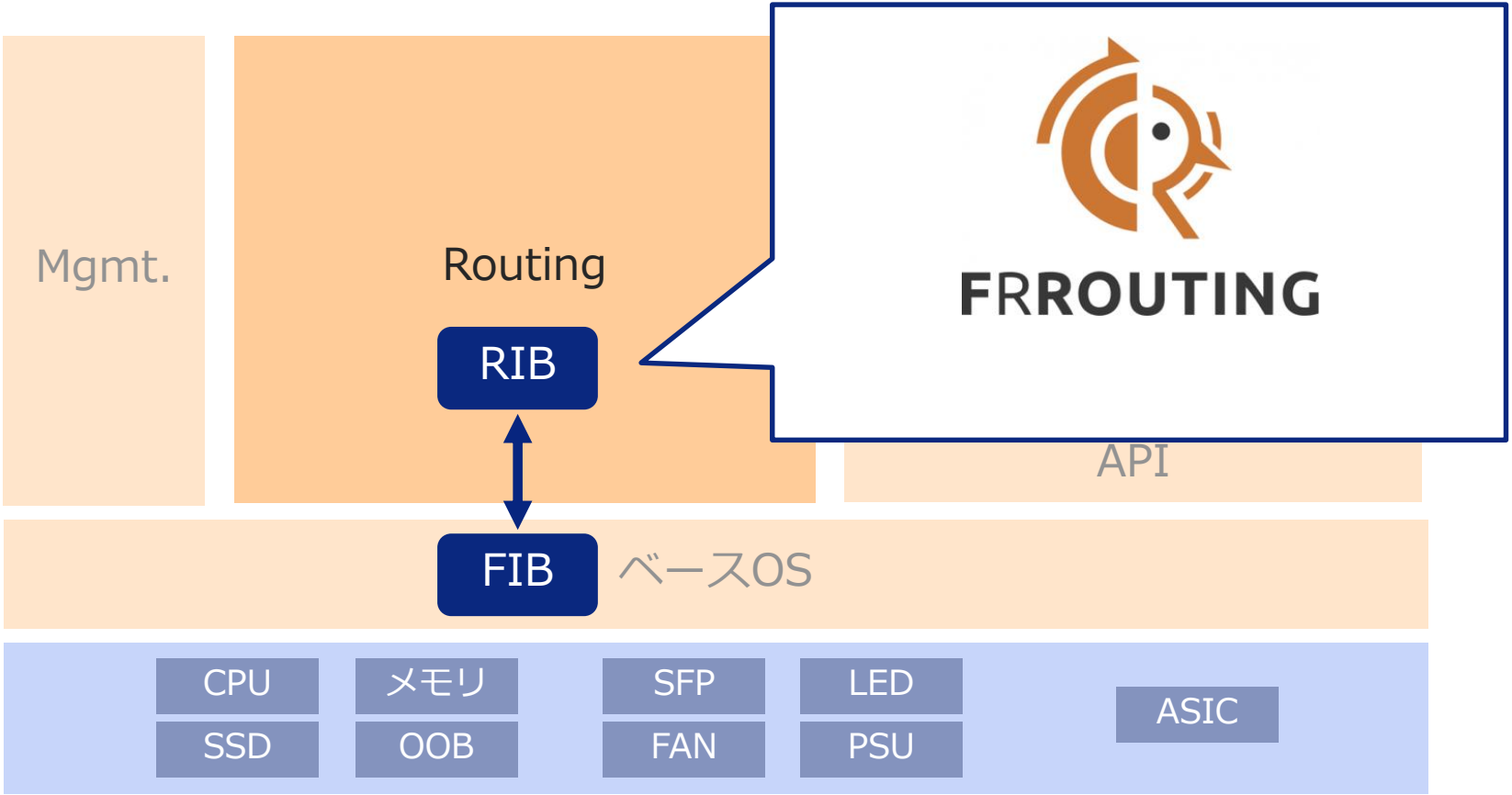
アーキテクチャ：概要（2/5）



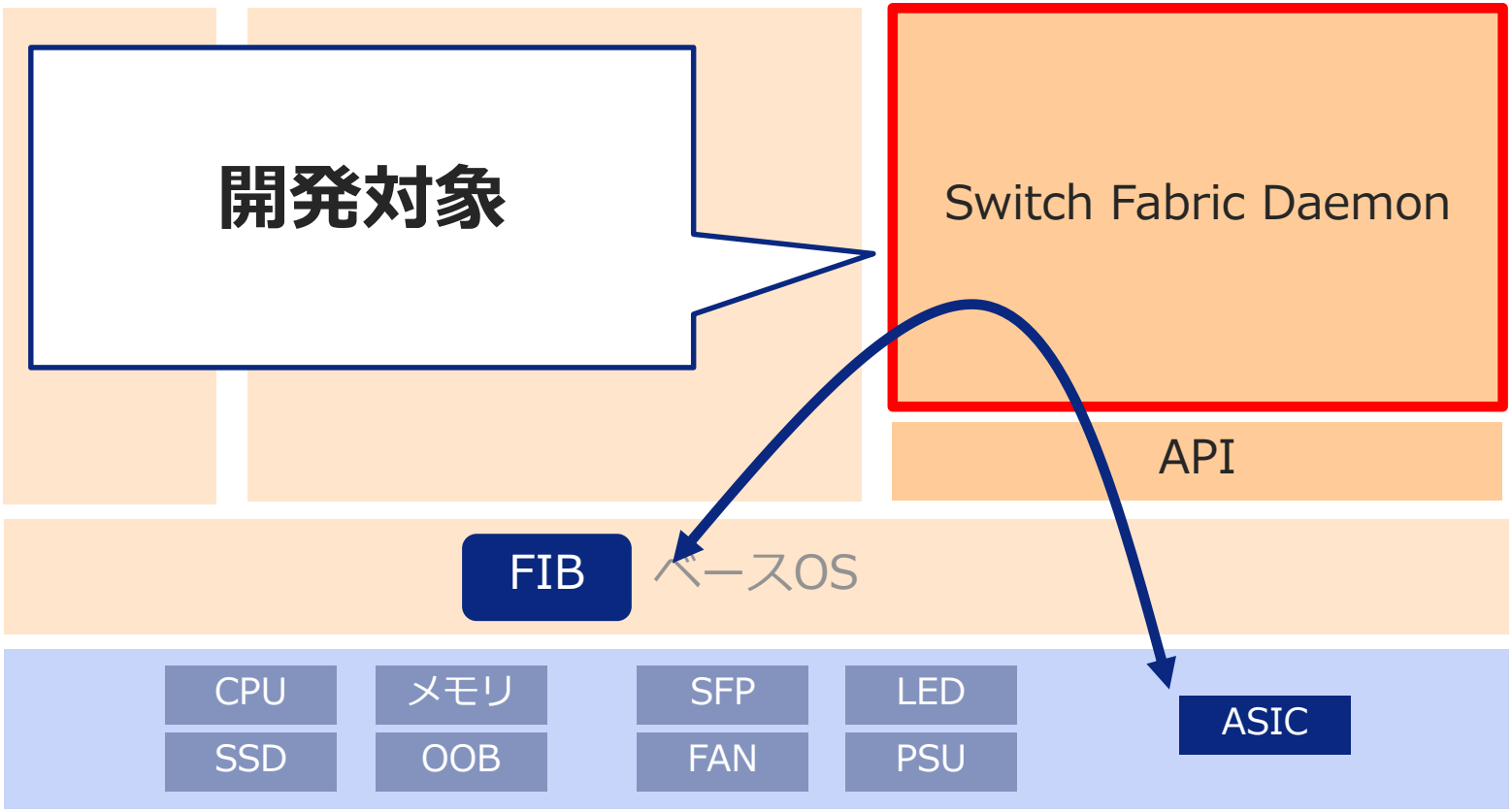
アーキテクチャ：概要（3/5）



アーキテクチャ：概要（4/5）

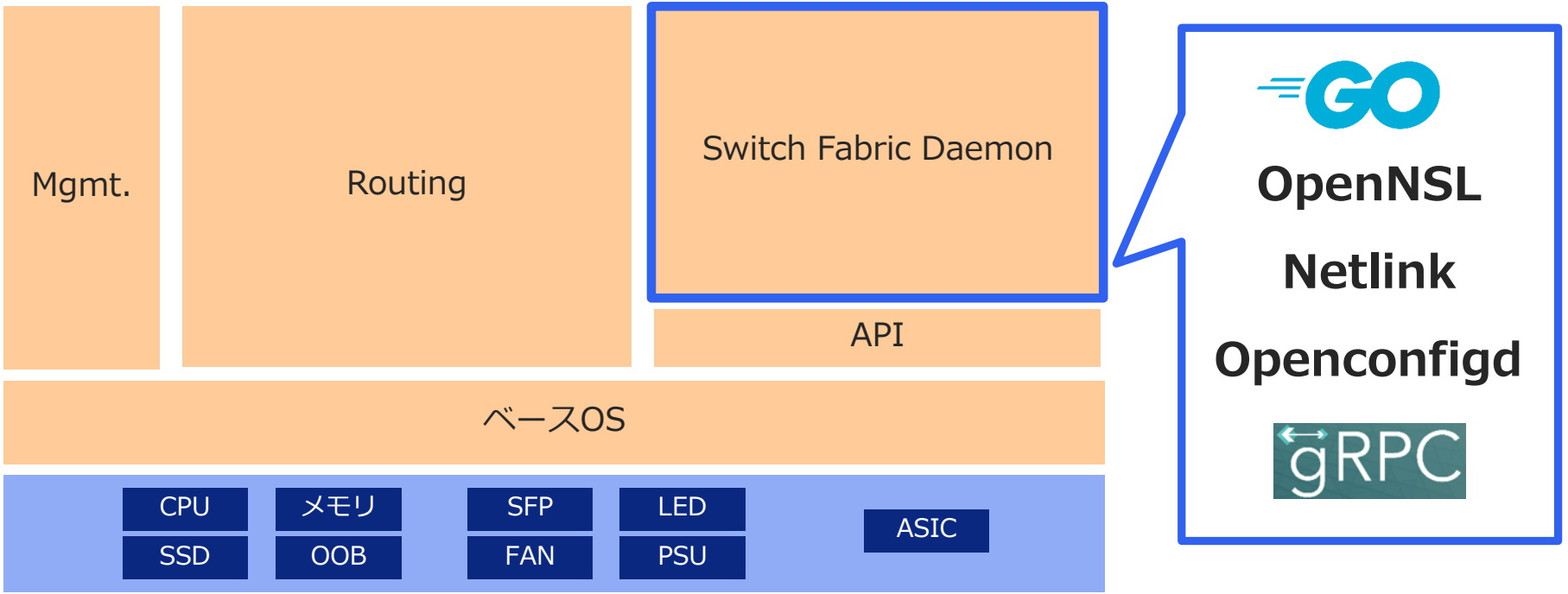


アーキテクチャ：概要（5/5）



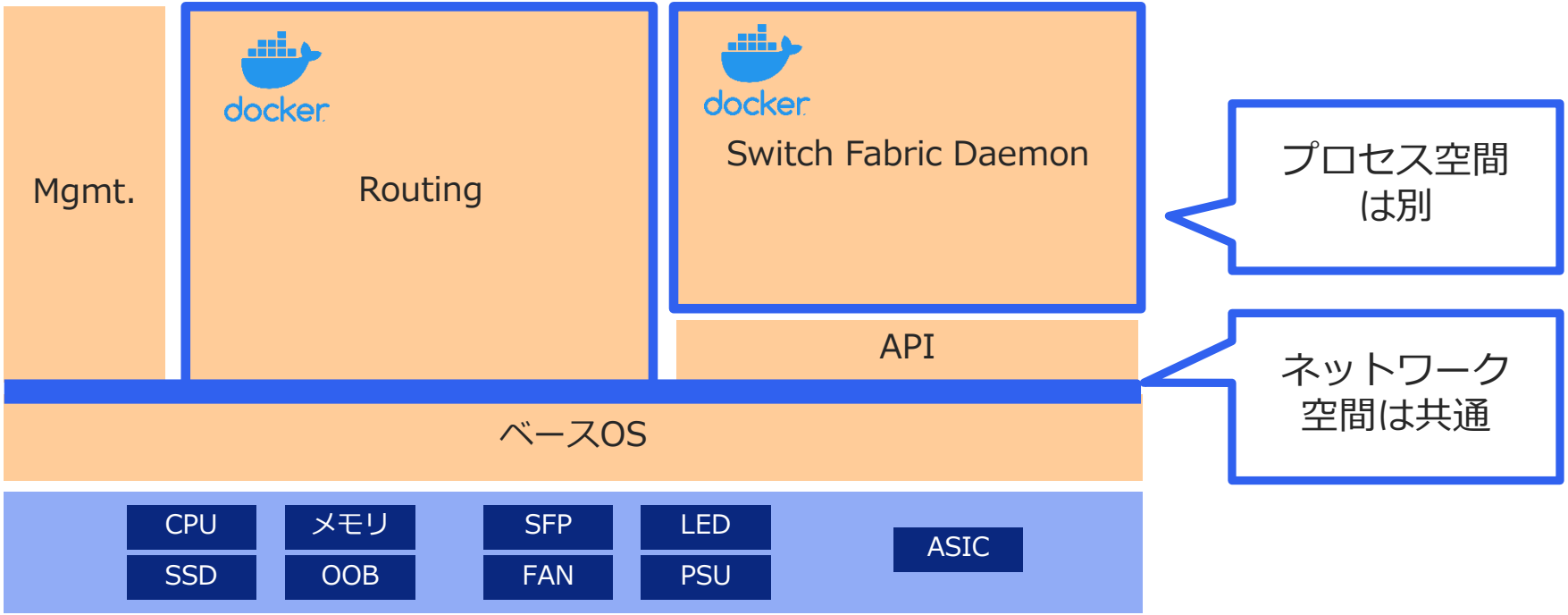
アーキテクチャ：設計思想（1/2）

- オープンなソフトウェア・技術を利用する
- ホストOSに(なるべく)依存しない柔軟なデーモン設計にする



アーキテクチャ：設計思想（2/2）

- オープンなソフトウェア・技術を利用する
- ホストOSに(なるべく)依存しない柔軟なデーモン設計にする

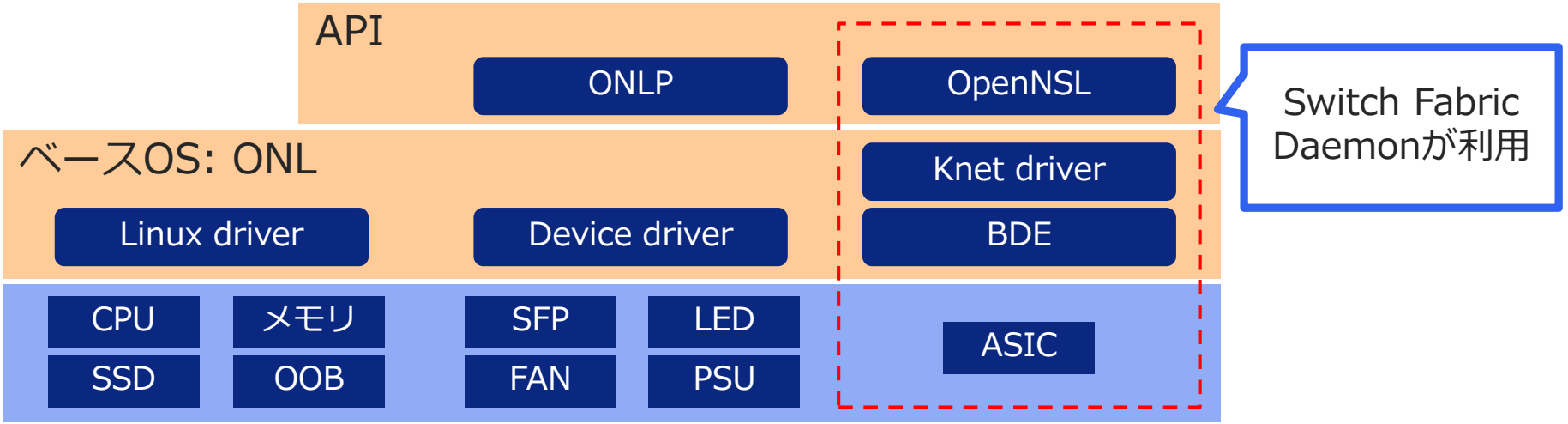


アーキテクチャ：サポート機能(2019年10月時点)

サポート済	制約
ARP/ND	Syslog・SNMP・SSH・ユーザ管理等はLinux機能を利用。
IPv4/IPv6	グローバルスコープのみ。 (VRFは未サポート)
Static	CLIは各コンポーネントで用意。
OSPFv2	
OSPFv3	
BGP4+	
SVI/VLAN	
ECMP (仮)	
Statistics (仮)	

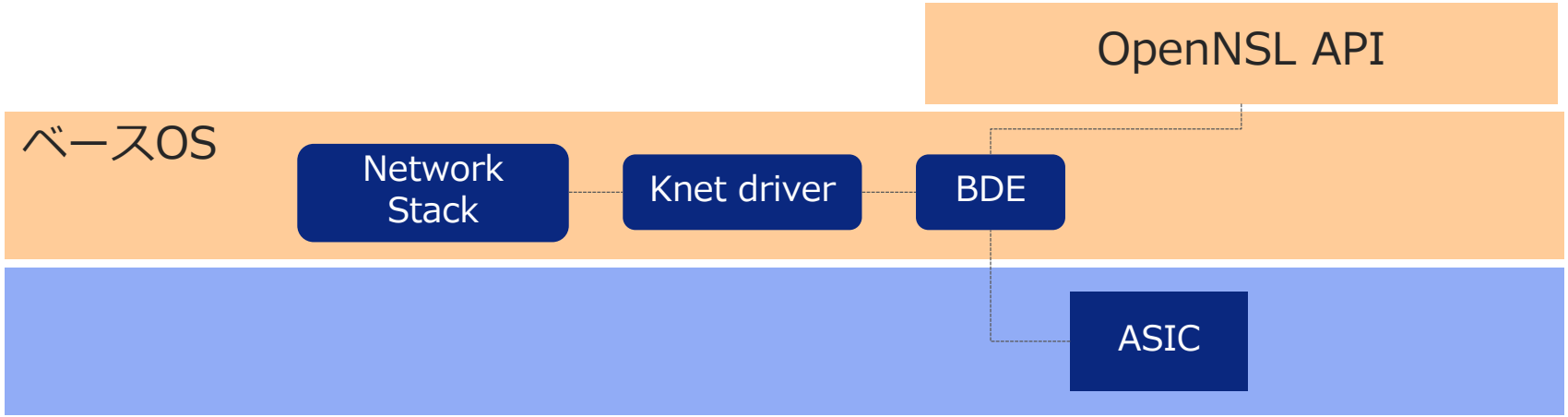
アーキテクチャ：設計（1/5）

ONL (Open Network Linux)	• Open Compute ProjectのOperating System。
ONLP(ONL Platform)	• ホワイトボックスのデバイスを管理するAPI。 • ONLに含まれる。
OpenNSL	• ASICを操作するAPI。 • Broadcomから提供される。



アーキテクチャ：設計（2/5）

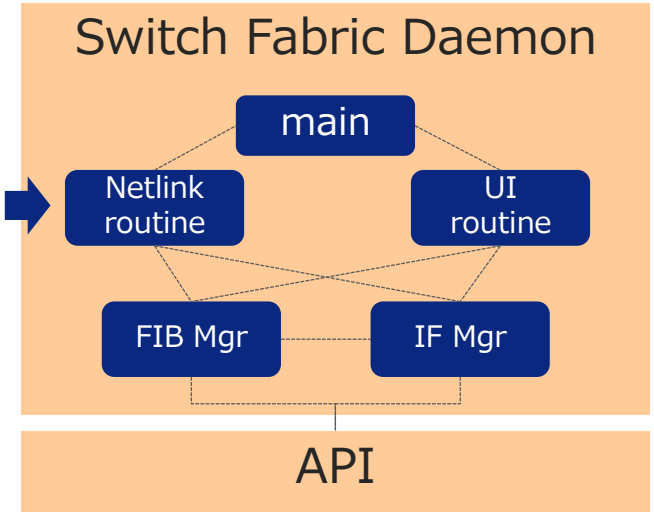
Network Stack	• Linuxのネットワークスタック。
Knet driver	• ネットワークインターフェースのモニタや管理。 • Network StackとBDEの連携（パケットの送受信）。
BDE (Broadcom Device Emulator)	• ASICのHardware abstractionを提供。



アーキテクチャ：設計（3/5）

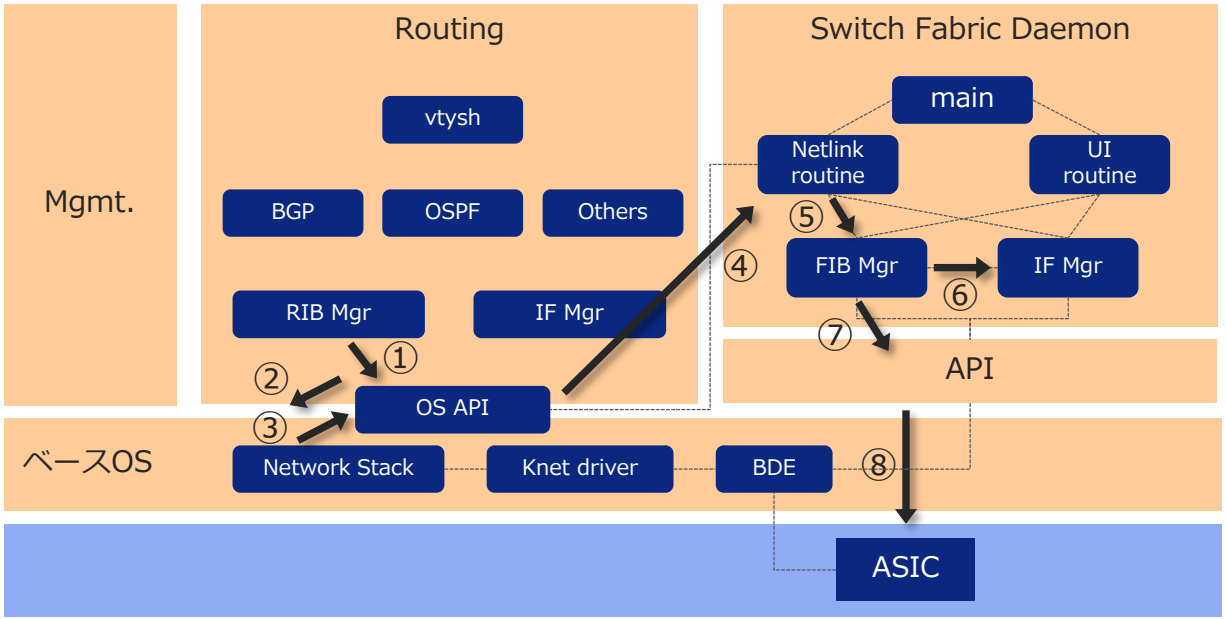
■ Switch Fabric Daemonの役割
「OS上のネットワーク情報を**ASICに書き込む**」

- AISCの初期設定
  (L2/L3モード、CPUにパントするパケットの定義等) → IF Mgr
- OSからの情報を受信、解析 → Netlink routine
- 受信した情報を整形 → Netlink routine
- 経路情報やインターフェース情報のASICへの書き込み → FIB Mgr, IF Mgr



アーキテクチャ：設計（4/5）

■ 動作例：経路更新



① 経路アップデート

② カーネル上のFIBを更新

③ カーネルがNETLINKメッセージ送信

④ NETLINKメッセージを受信

⑤ NETLINKメッセージを解析し、内部テーブルをアップデート

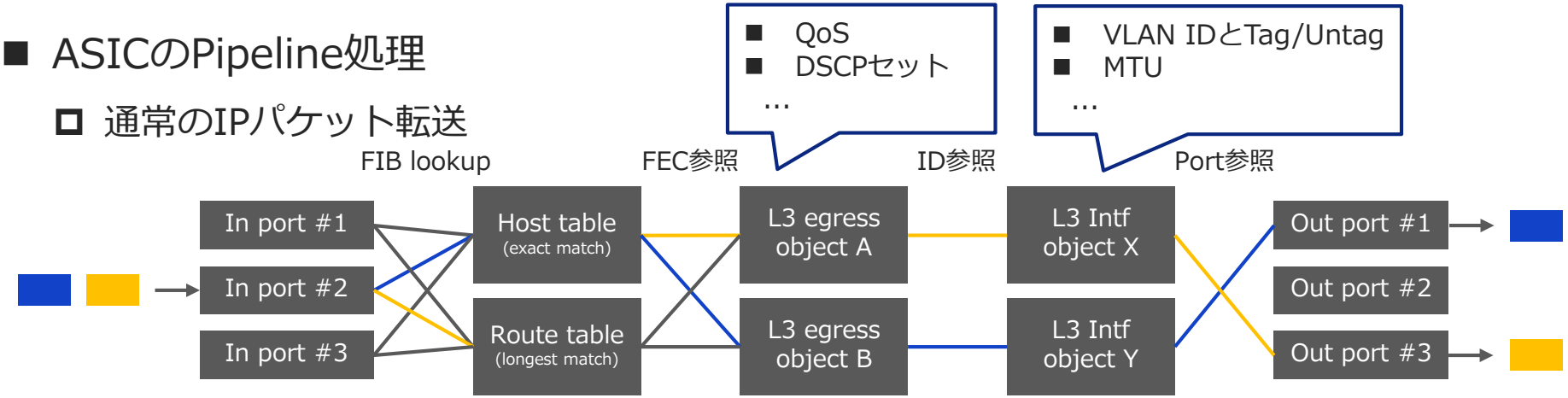
⑥ 該当経路をフォワードするインターフェースの状態を確認

⑦ **経路アップデート情報をASICに書き込むAPIをコール**

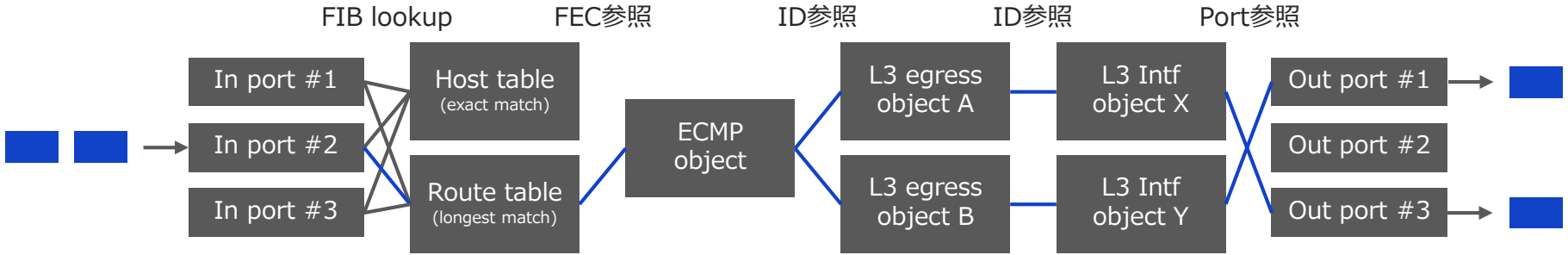
⑧ **経路アップデート情報をASICに書き込み**

アーキテクチャ：設計（5/5）

- ASICのPipeline処理
  - 通常のIPパケット転送



  - ECMPのケース



アジェンダ

KDDIがルータ向けソフトウェアを作ってみた話をします。

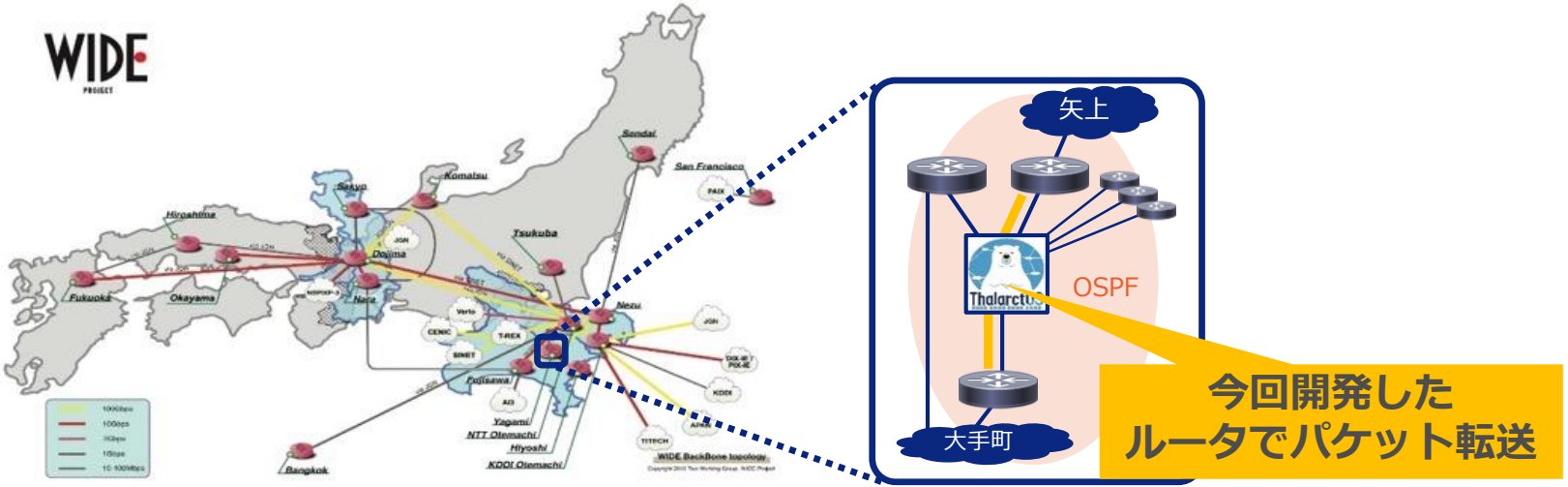
1. ホワイトボックスとは
2. ルータ開発の背景
3. アーキテクチャ
4. **トライアル**
5. 開発中の苦労や失敗談
6. まとめ

トライアル

■WIDE網で実験中です。

- **https://news.kddi.com/kddi/corporate/newsrelease/2019/06/11/3849.html**

WIDEプロジェクトとKDDI、オープンソースソフトウェアを活用した最大3.2テラビットのパケット転送が可能なルーターを導入



アジェンダ

KDDIがルータ向けソフトウェアを作ってみた話をします。

1. ホワイトボックスとは
2. ルータ開発の背景
3. アーキテクチャ
4. トライアル
5. **開発中の苦労や失敗談**
6. まとめ

開発中の苦労や失敗談（1/4）

Forwardingされるパケットにのみ、Default VLANが意図せず、挿入される。

- OpenNSLの動作理解には、ブラックボックステストが必要
  - ドキュメントの関数/変数の説明が詳しくない
  - ドキュメントが更新されていない

Function Documentation

http://broadcom-switch.github.io/OpenNSL/doc/html/pages.html

```
int example_create_l3_egress ( int          unit,
                               uint32       flags,
                               int          out_port,
                               int          vlan,
                               int          l3_eg_intf,
                               opennsl_mac_t nhop_mac_addr,
                               int *        intf,
                               int *        encap_id
                             )
```

Parameters

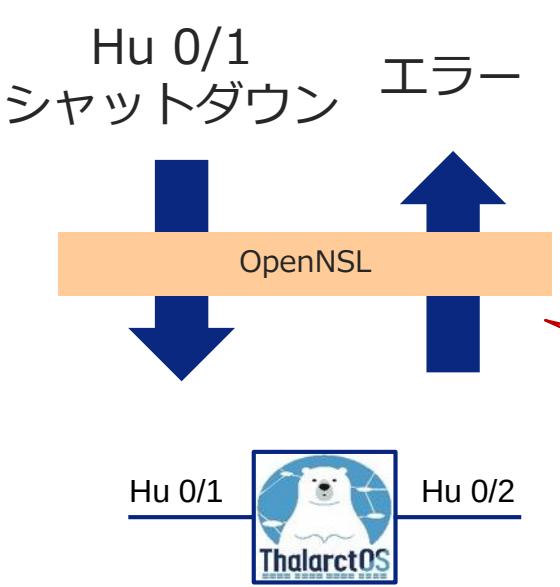
unit	[IN] Unit number
flags	[IN] special controls set to zero
out_port	[IN] egress port
vlan	[IN] VLAN identifier
l3_eg_intf	[IN] egress router interface will derive (VLAN, SA
nhop_mac_addr	[IN] next hop mac address
*intf	[OUT] returned interface ID
*encap_id	[OUT] returned encapsulation ID

Returns

OPENNSL_E_xxx OpenNSL API return code

開発中の苦労や失敗談（2/4）

インターフェースのshutdownがエラーになる。

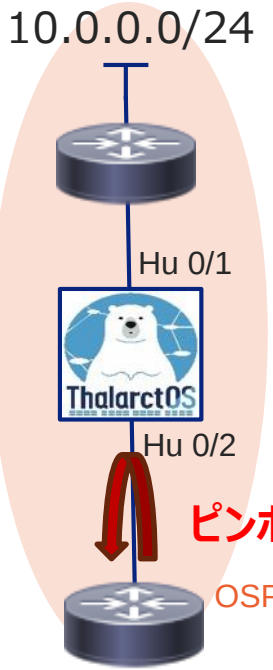


- OpenNSLの動作理解には、ブラックボックステストが必要
  - ドキュメントの関数/変数の説明が詳しくない
  - ドキュメントが更新されていない

パケットがキューにある状態で、L3 Egress Objectを削除しようとすると、エラーコードを返す

開発中の苦労や失敗談（3/4）

LinuxとASICの経路に差分が発生する。



■10.0.0.0/24の経路情報

	Linux	ASIC
OUT IF	Hu0/1	Hu0/2

経路情報を正しくASICに書き込めていない。

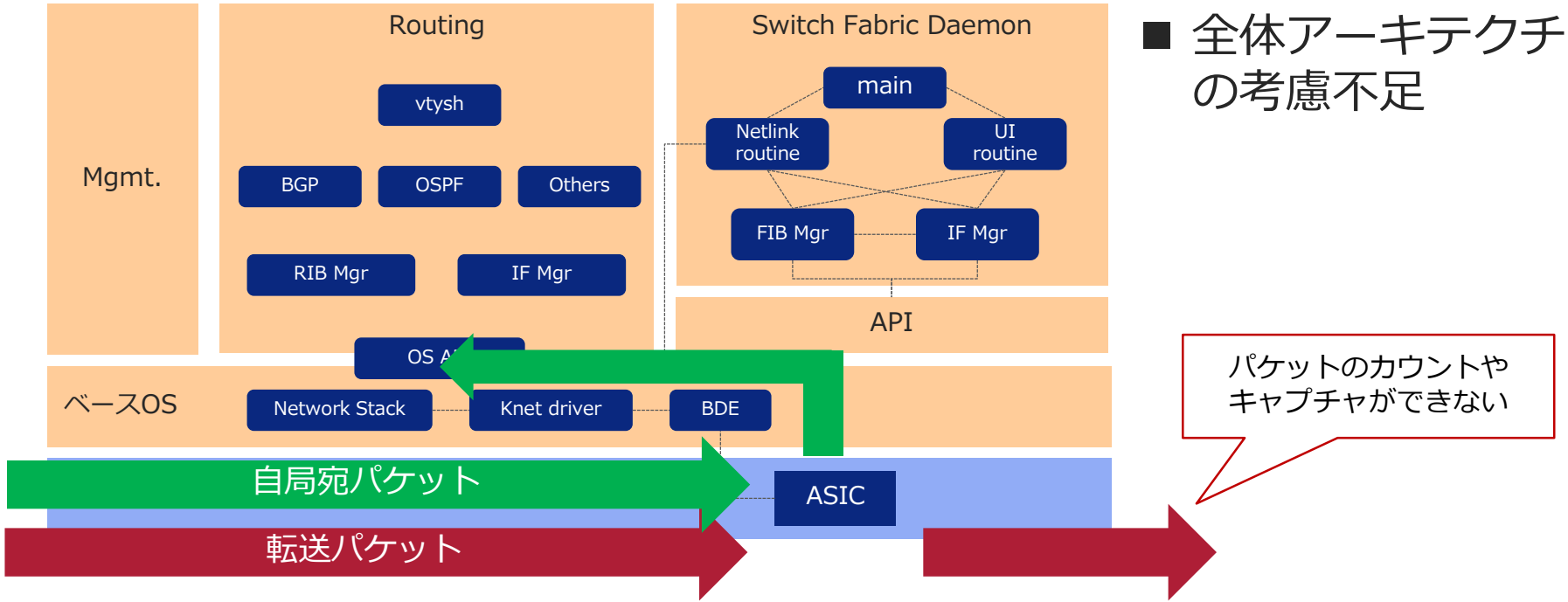
■ Netlinkの動作理解にも、ブラックボックステストが必要
- 様々なパターンのタイプとフラグ
- IPv4とIPv6で差分あり

Netlinkメッセージの例（経路変更）

	想定	実際
v4	DelRoute→AddRoute	AddRoute（REPLACE FLAG）
v6	DelRoute→AddRoute	DelRoute→AddRoute

開発中の苦労や失敗談（4/4）

パケットのカウンタが正しくない。



- 全体アーキテクチャの考慮不足

アジェンダ

KDDIがルータ向けソフトウェアを作ってみた話をします。

1. ホワイトボックスとは
2. ルータ開発の背景
3. アーキテクチャ
4. トライアル
5. 開発中の苦労や失敗談
6. **まとめ**

まとめ

- オープンなソフトウェア・技術を利用して、ルータを開発。
- ホワイトボックス開発にはある程度のブラックボックステスト(動かないことを前提にしたデバッグ環境)が必要。単体試験ではわからない。
  - OpenNSLの引数と返り値
  - Netlinkのパターンとフラグ
  - パケットキャプチャ (100GE)
- 今後の展望
  - コマンドラインの統一
  - ONLPへの対応
  - 機能（VRFやTelemetry）の追加


Tomorrow, Together
KDDI
おもしろいほうの未来へ。
au