JANOG44 Day3 @ 神戸国際展示場

オープンソースソフトウェアを活用して Disaggregated Routerを"1"から開発した話

2019年 7月 26日

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

自己紹介

●鈴木　雄一郎 (KDDI株式会社 IPネットワーク部)

＜経歴、担当業務＞

Softbank Telecom (2006-2012)：法人向けIPNW(solteria)の設計/構築

NTTドコモ (2012-2018)：LTE/3G用のIPバックボーンの構築、EPC開発

KDDI (2018-)：5G用NW設計/構築、ホワイトボックススイッチ開発

●仲山　裕也　(KDDI株式会社 IPネットワーク部)

＜経歴、担当業務＞

KDDI (2017-2018)：IP設備の監視、作業

(2018-)：FTTH/5G NW設計/構築、ホワイトボックススイッチ開発

自己紹介

●**Hans Tseng（Delta Eelectronics, Inc)**

＜経歴、担当業務＞

CipherLab Co., Ltd(2009-2011) ： RFID antenna design, Embedded Linux system

Delta Eelectronics, Inc. (2011-)：Wireless AP development

ONL, SONiC, Linux Driver development

●野崎　忠雄　**(Delta Electronics Japan Inc.)**

＜経歴、担当業務＞

富士通（～2014)

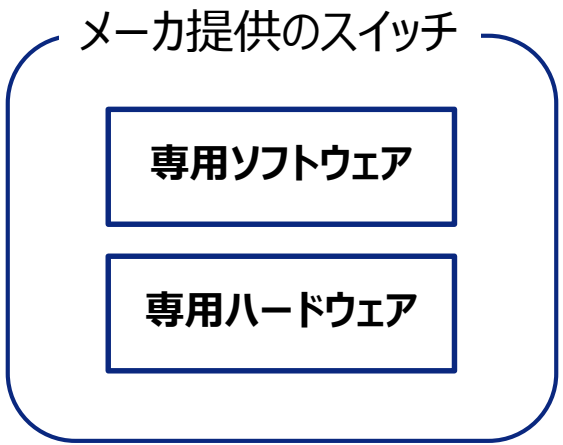
Delta Electronics Japan Inc (2014-)： White Box Switchの拡販活動にFAEとして従事

Agenda

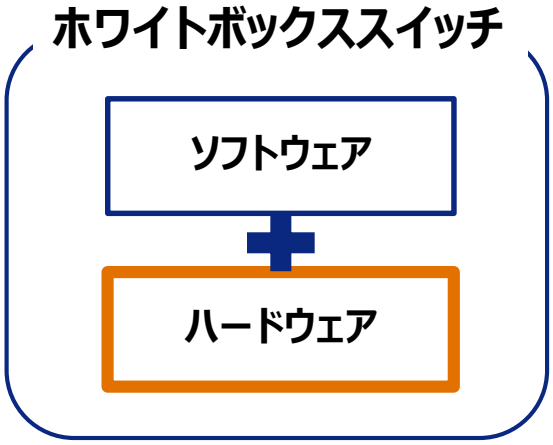
1. ホワイトボックススイッチとは
2. ルータ開発の背景
3. **ThalarctOS** アーキテクチャ
4. **ThalarctOS Demo**
5. 開発中の苦労や失敗談/今後の展望
6. ディスカッション、質疑＆応答

ホワイトボックススイッチとは

- ソフトウェアが含まれないスイッチ
- ユーザがハードウェアとソフトウェアを選択可能



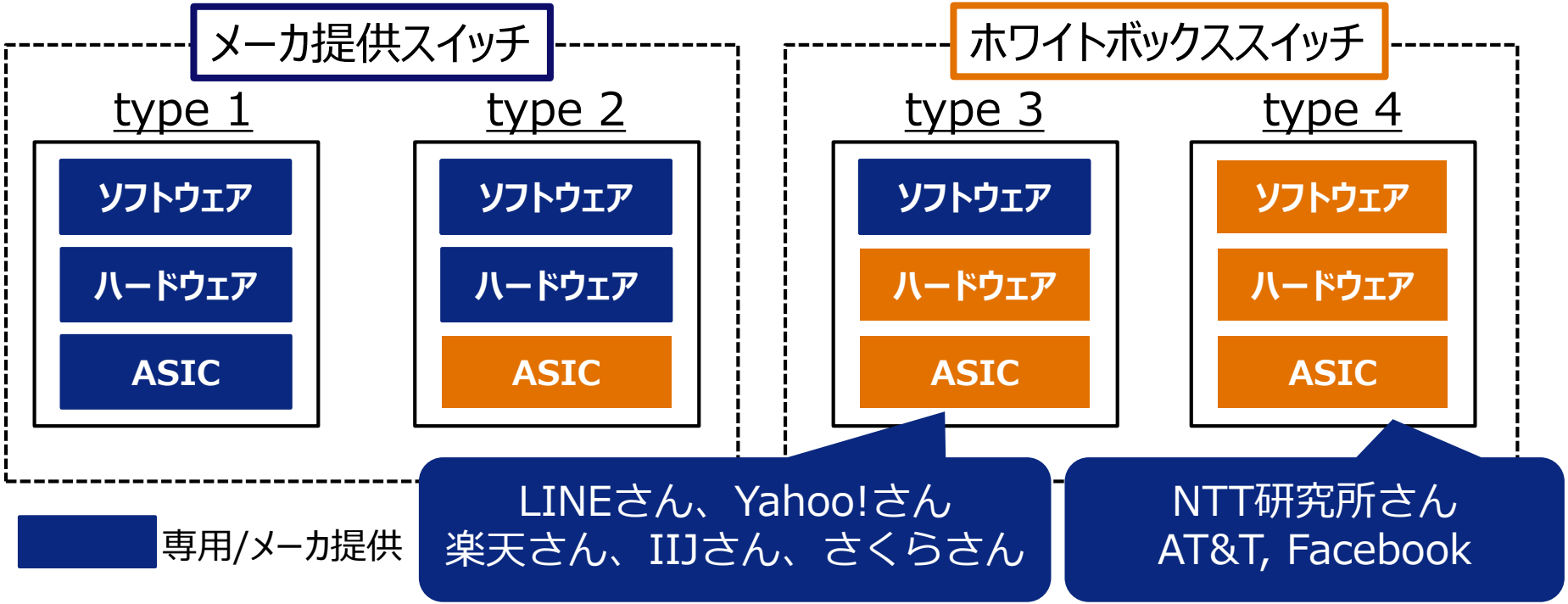
専用ハードウェアと
専用ソフトウェアの
垂直統合型スイッチ



ソフトウェアと
ハードウェアが分離
組み合わせを自由に選択可能

ホワイトボックススイッチとは

- ASIC/ハードウェア(ASIC除く)/ソフトウェアの組み合わせにより4種類に分類可能
- Type 3/4を本資料ではホワイトボックススイッチ(以降WBSと表記)と呼ぶ。



開発背景

WBSは自由な機能実装やコストの削減において、大きなポテンシャルを秘めていると考えられている。しかし、「機能実装の難しさ」や「導入費用以外のコスト」については不明確な点が多い。そこで、「商用化に向けた課題の確認」および「適用領域の見極め」を行うために自社での開発を行った。

背景		
	オープンソースのネットワークOSの登場	● オープンソースのネットワークOSやRoutingアプリケーションが登場し、自由に使用できるようになった。
	チップセットの大容量・オープン化の進展	● 大容量チップセットのAPIが一部オープンになり、メーカ以外でもチップセットを制御することが可能になった。
	SP向けのWBSラインナップの増加	● 世界の様々なキャリアからの要望により、SP向けのWBSラインナップが増加している。

- オープンソースソフトウェアを活用したプロトタイプ機の開発を実施し、WBS導入による効果を確認する。

ThalarctOS アーキテクチャ

ハードウェア：Broadcom社Tomahawk搭載スイッチ 100G×32ポート(Delta:AG9032v1)
ソフトウェア：ベースOS/ルーティングアプリ/マネジメント系アプリはオープンソースを使用
ルーティング情報をASICに書き込むソフトウェア(Switch Fabric Daemon)を開発

機能
snmp,syslog,IF管理

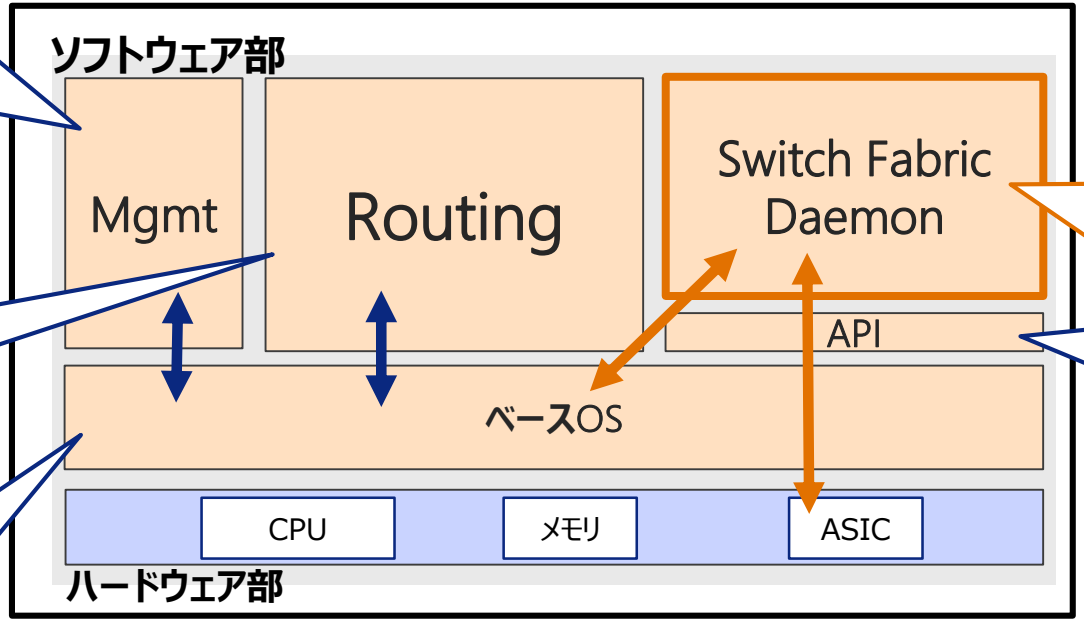
オープンソース
Net-snmp
rsyslog
iproutes2

機能
ルーティングアプリ

オープンソース
FRRouting

機能
ベースOS,ハードウェア管理

オープンソース
Open Network Linux



開発対象

機能
OS上のルーティング情報をASICに書き込む。

オープンソース
なし

機能
ASICのAPI

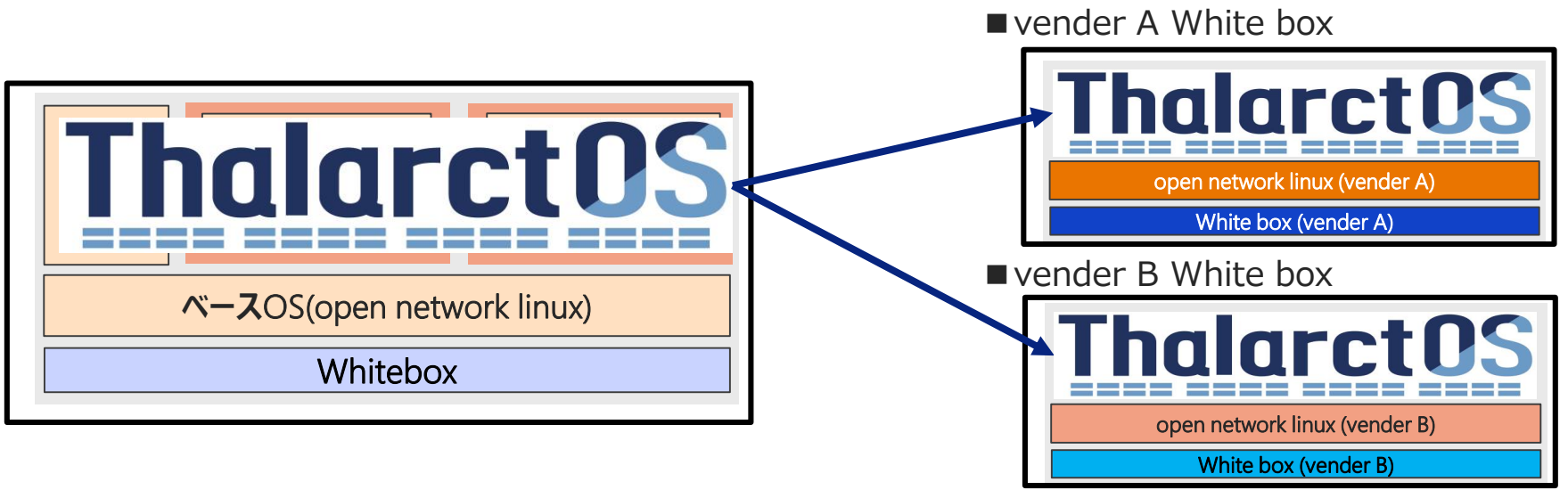
オープンソース
open NSL

ソフトウェアデザイン

目指した事

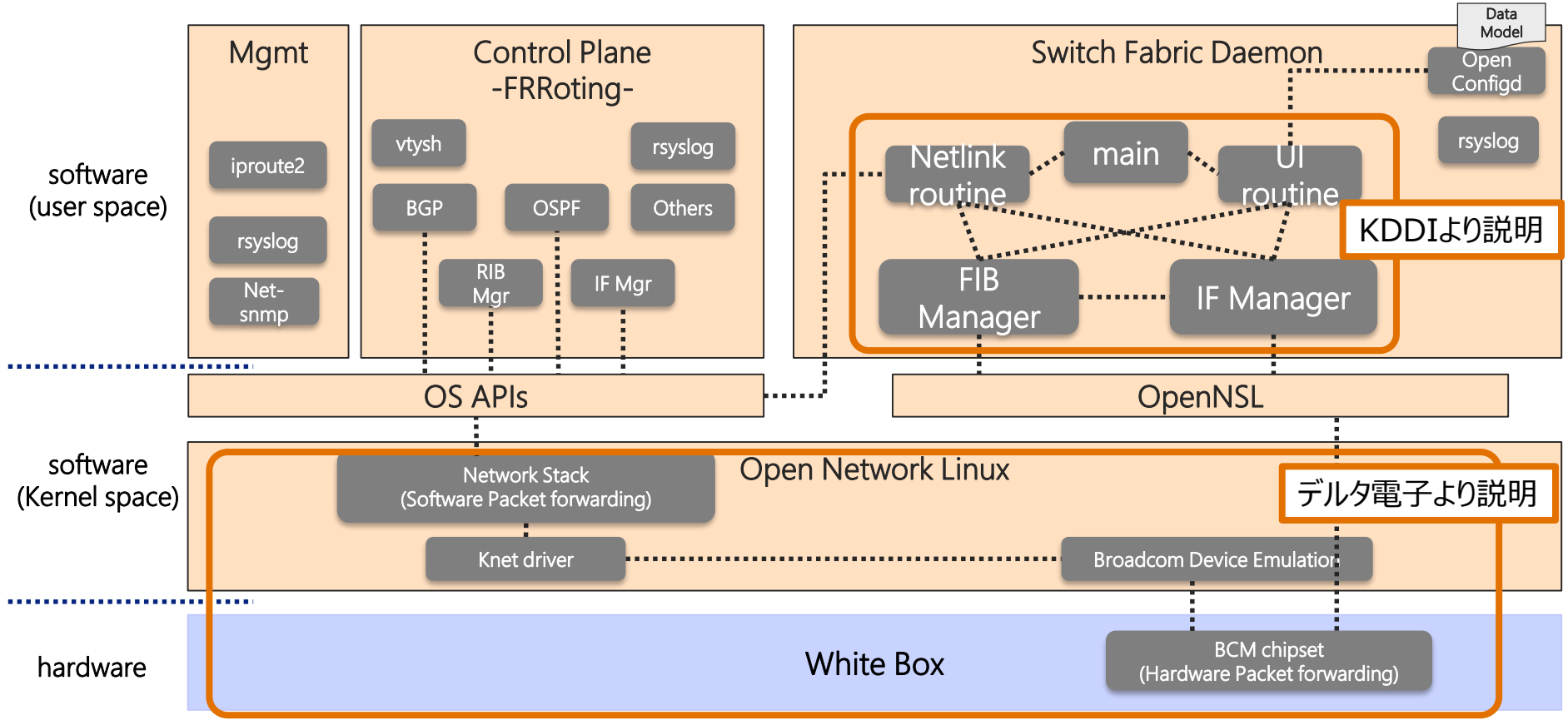
- ホストOSに依存しない柔軟なデーモン設計
- デプロイ・ロールバックのスピードup
- ソフトウェアのポータビリティ性向上

⇒コンテナ化によりHW/ベースOSとの依存関係を減らし、DisaggregatedなOSへ！

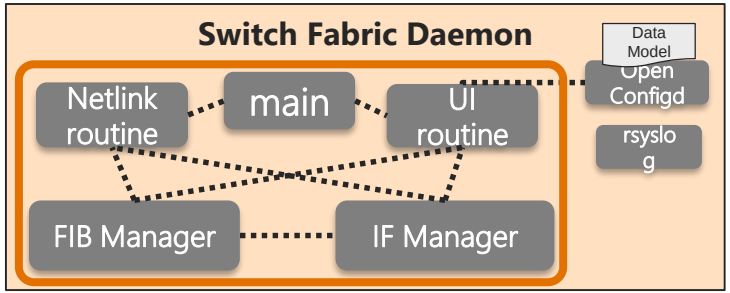


ThalarctOS 詳細アーキテクチャ

ThalarctOS 詳細アーキテクチャ



Switch Fabric Daemon 詳細コンポーネント



- **Switch Fabric Daemonの役割**

「OS上のルーティング情報をASICに書き込む。」

↑ん、なんか簡単そう！？

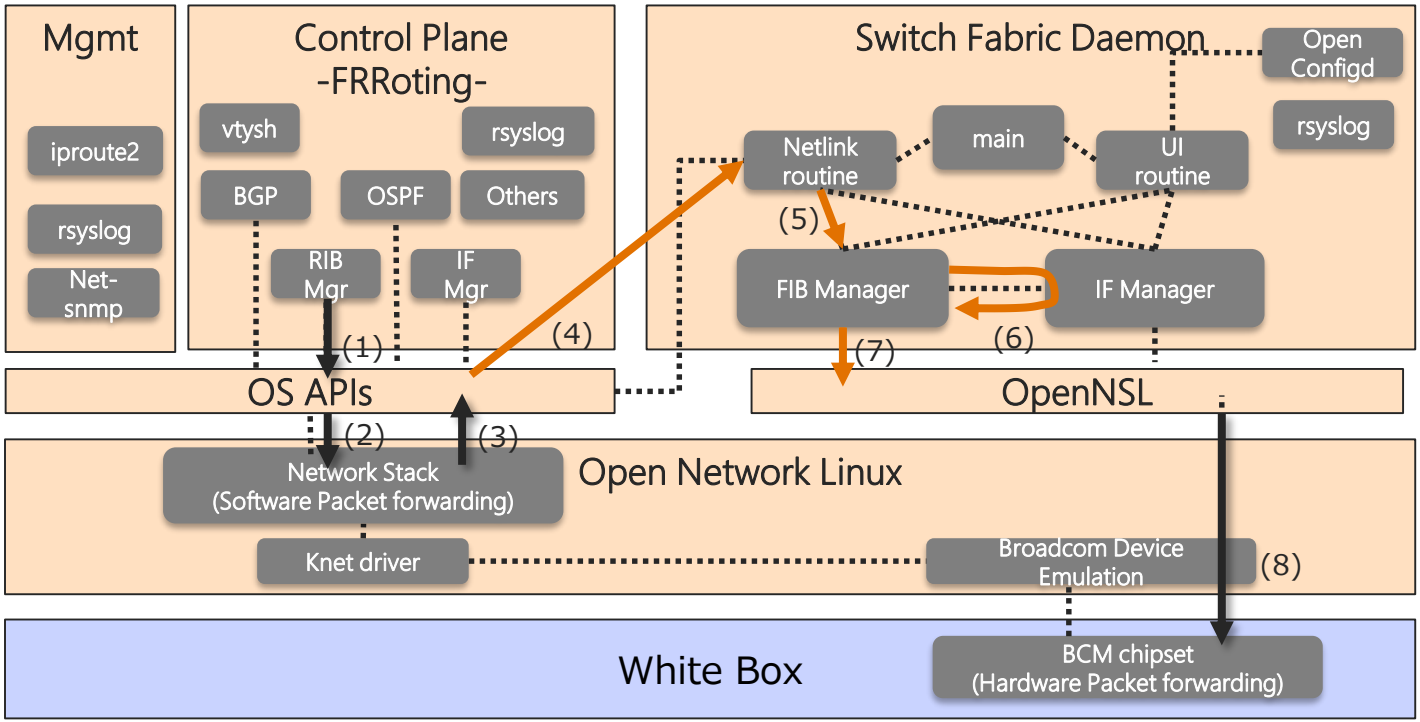
-ASICの初期コンフィグレーション
(L2/L3モード、CPUにパントするパケットの定義) → IF Manager

-OSからの情報を受信、解析 → Netlink routine

-ASICが理解できるメッセージに整形 → IF Manager

-経路情報の管理、経路情報のASICへの書き込み → FIB Manager

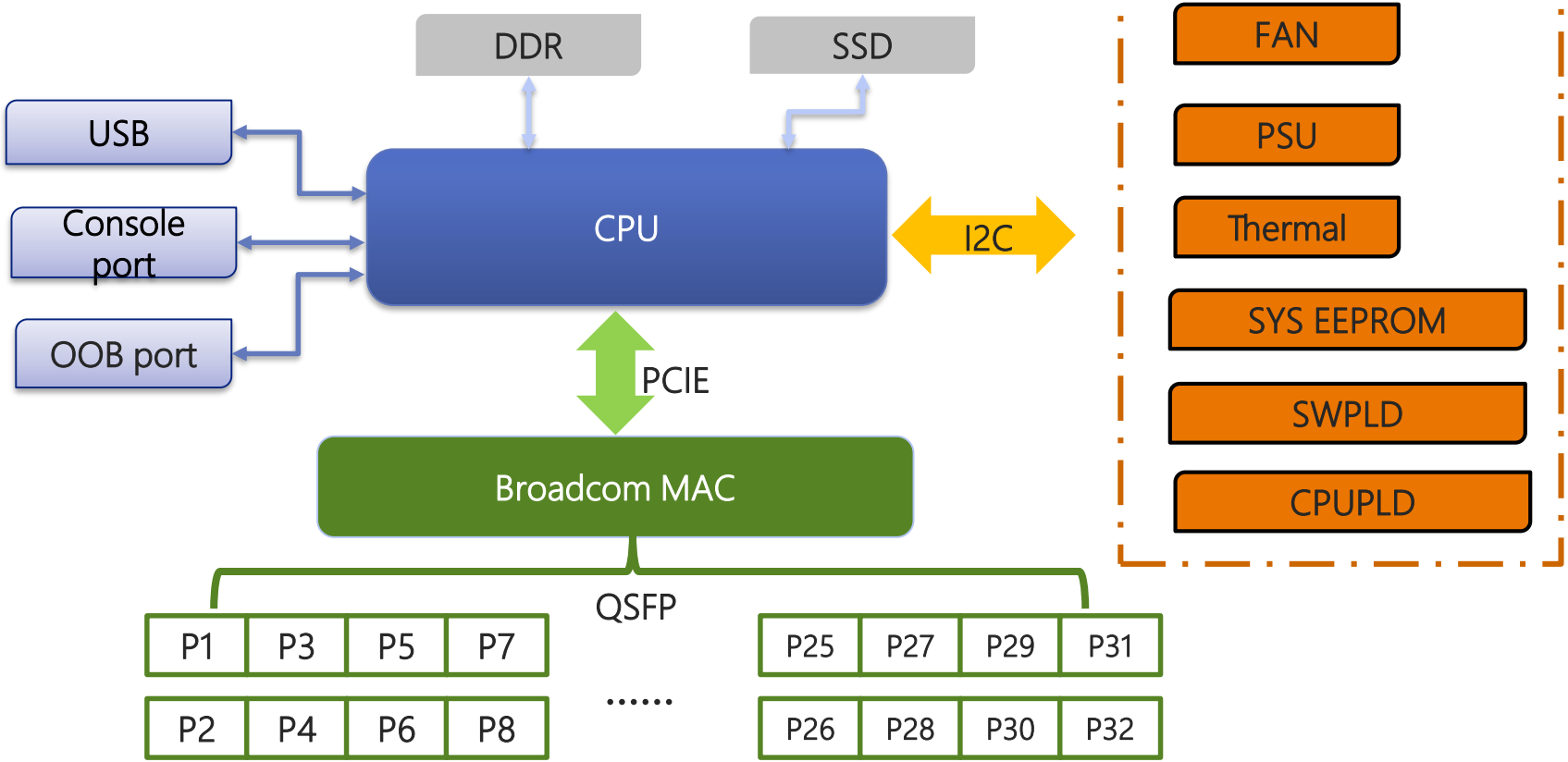
プロシージャ紹介 （経路update）



1. 経路更新発生
2. カーネル上の経路テーブル(FIB)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. メッセージを解析し内部テーブルをアップデート
6. 該当の経路をフォワードするIF state/IDを確認
7. 経路情報をASICに書き込むAPIをcall
8. 経路情報をASICに書き込む

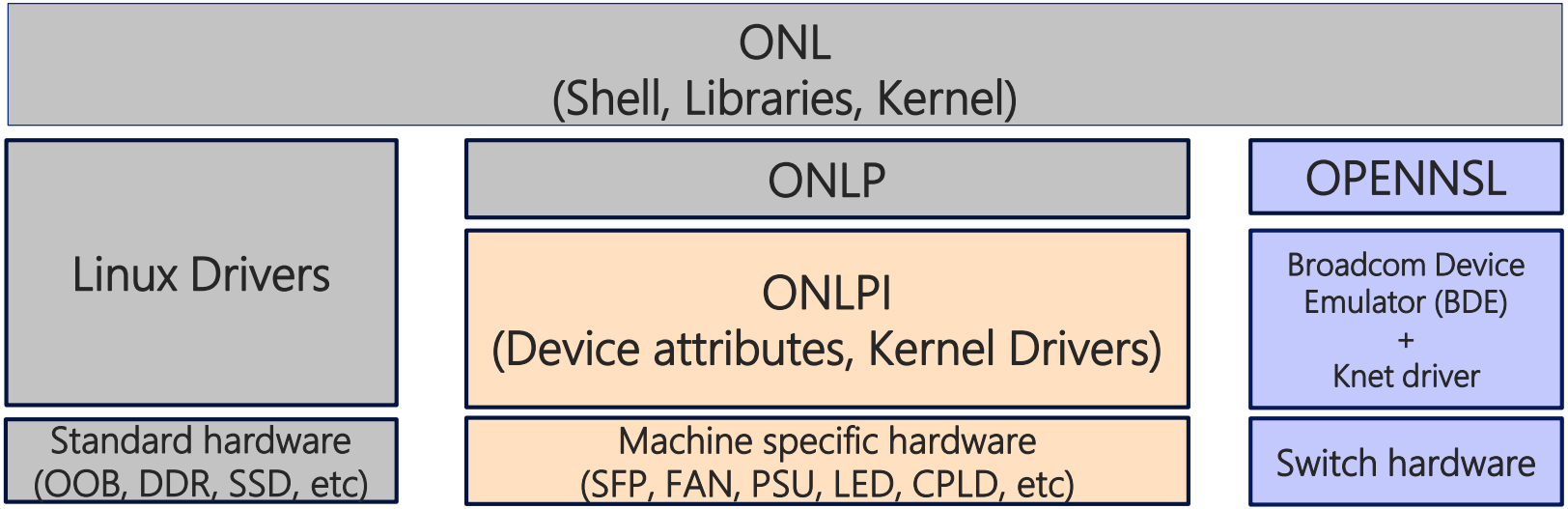
HW/ONLについて

Brock Diagram of AG9032v1



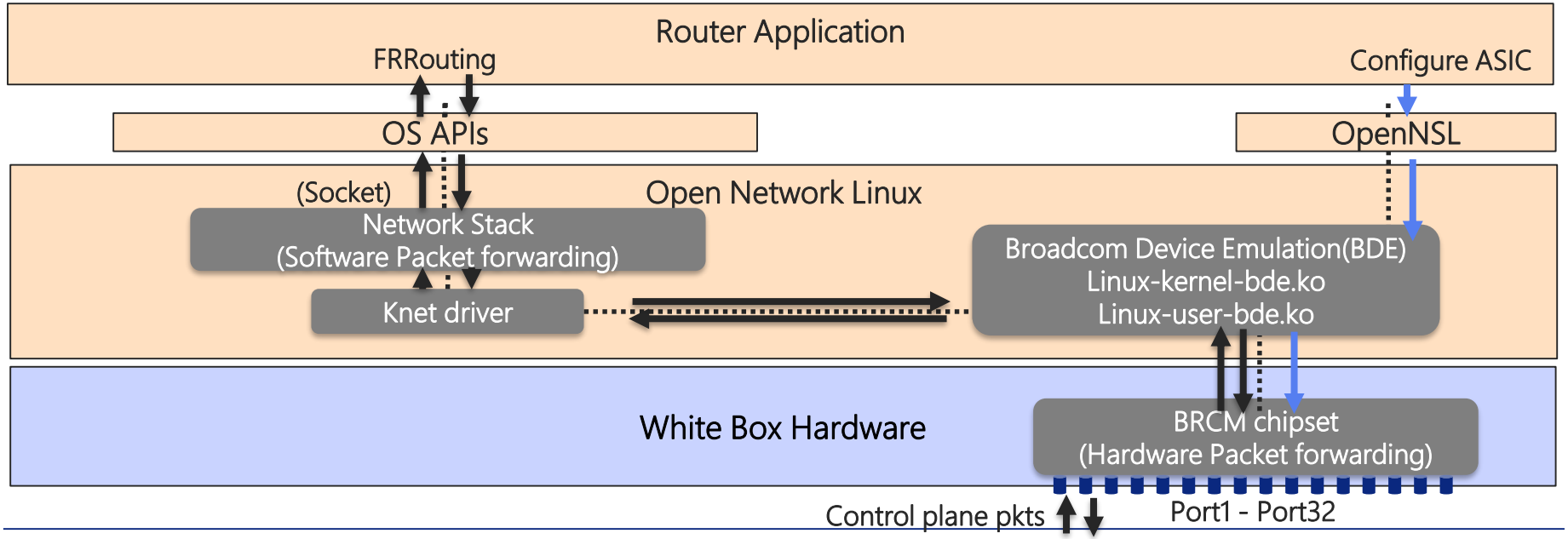
ONL and Hardware Integration

- ONL: An Operating System for Bare Metal Switches by OCP
- ONL: A complete Debian distribution + ONLP + associated drivers
- ONLP: provide abstraction interfaces to SFP, PSU, FAN, LED, etc.
- ONLPI: provides ONLP with hardware mapping and drivers
- OpenNSL: external to ONL, provided by Broadcom separately



OpenNSL kernel modules

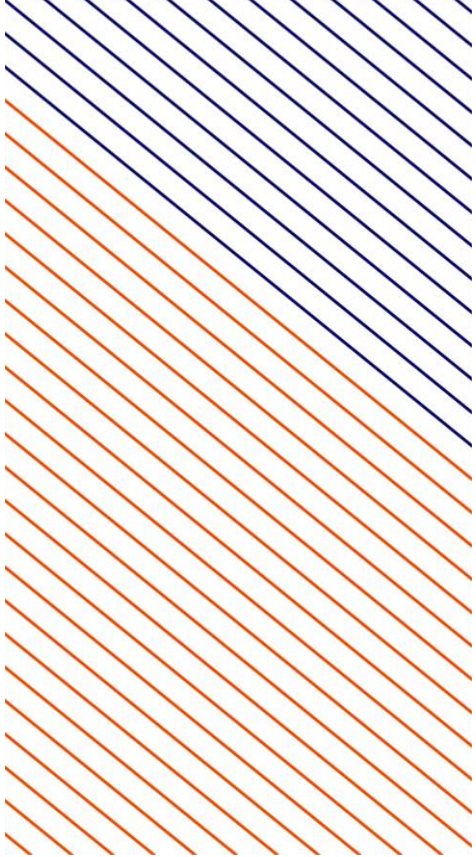
- Network stack: standard Linux Network Stack
- Knet driver: create/manage multiple virtual network interfaces, relay transactions from/to Linux Network Stack and BDE
- Broadcom Device Emulation: provides Switch ASIC hardware abstraction



Knet driver

- Create multi network interfaces(32 ports)
- Trap the control packets(BGP, OSPF, ARP, ND, ICMP) to Linux network stack
- Send out the control packets generated locally to the outside of the box
- Physical Link status monitoring and statistics

- Filtering mechanism can work in conjunction with the switch device field processor
- Can manage multiple network interfaces using KNET “filters” to direct CPU bound traffic to the desired network interface
- Utilize Linux Socket Buffers

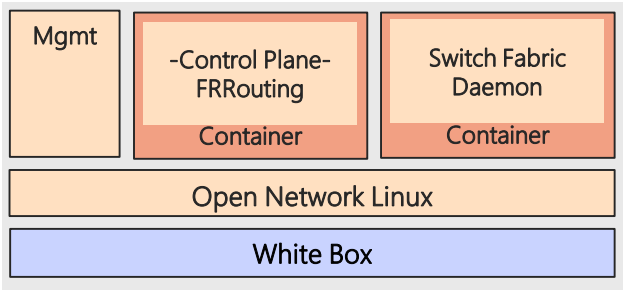


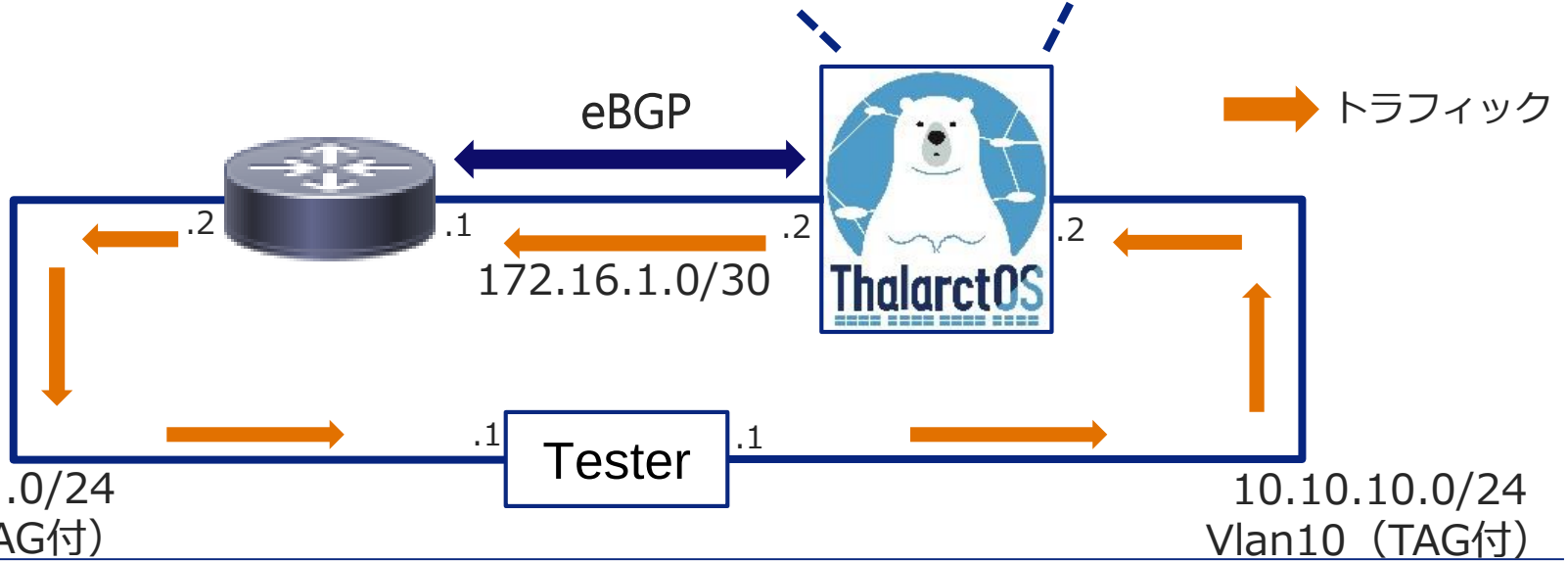
ThalarctOS デモ

ThalarctOSデモシナリオ

■ デモ

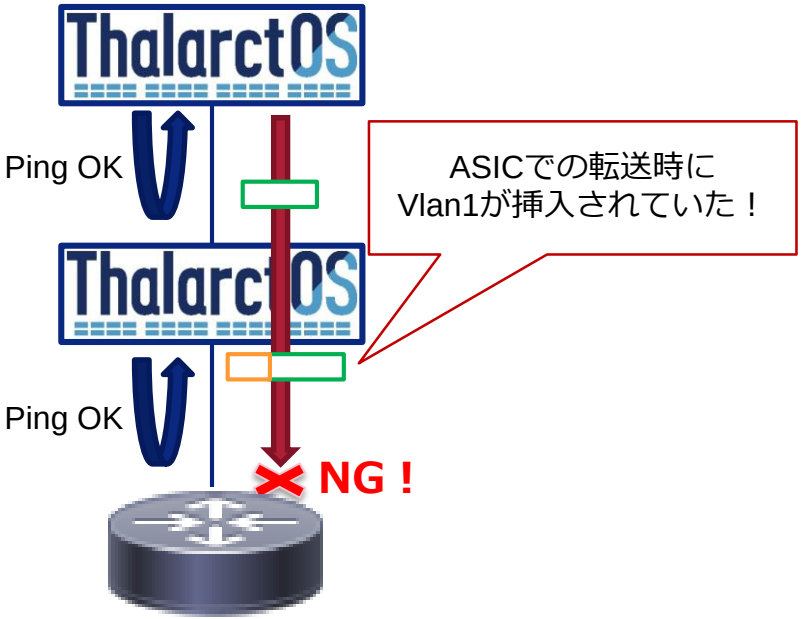
1. ThalarctOS起動
2. 対向ルータとBGP接続
3. トラフィック疎通確認





失敗談①

■ Vlan1が挿入されてしまう
→ OpenNSLの理解不足



OpenNSLの理解不足

- OpenNSLのドキュメント
  - 関数、引数の説明が詳しくない
  - 更新もされていない（最終更新2017年）

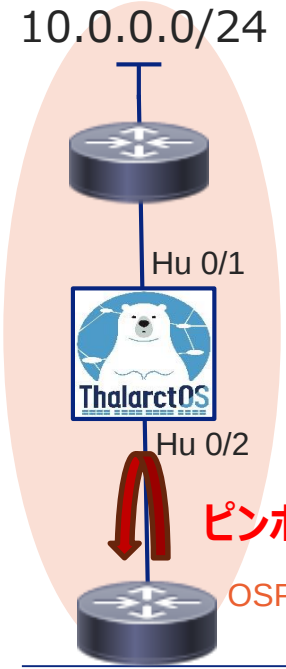
➡ Black Boxテスト

謎のFlag

```
 /*** create egress object 1 ***/
flags = 0;
rv = example_create_l3_egress(unit, flags, out_sysport,
out_vlan,ing_intf_out,next_hop_mac, &l3egid);
if (rv != OPENNSL_E_NONE) {
    printf("Error, create egress object, out_sysport=%d. Return code %s ¥n",
                        out_sysport, opennsl_errmsg(rv));
return rv;
}
```

失敗談②

■ FRRとASICの経路に差分発生
→ FRR、Netlinkの動作理解不足



■10.0.0.0/24の経路情報

	FRR	ASIC
OUT IF	Hu 0/1	Hu 0/2

FRRからの経路情報が正しく
ASICに書き込まれていない。。

FRR、Netlinkの動作理解不足

- 様々なパターンのメッセージやフラグ
  - 単純な経路追加、経路変更
  - Single Path → Multi Path　等々
  - v4とv6でメッセージに差分あり

Netlinkメッセージの例

タイミング		想定	実際
経路変更	v4	DelRoute→AddRoute	AddRoute（REPLACE FLAG）
	v6	DelRoute→AddRoute	DelRoute→AddRoute

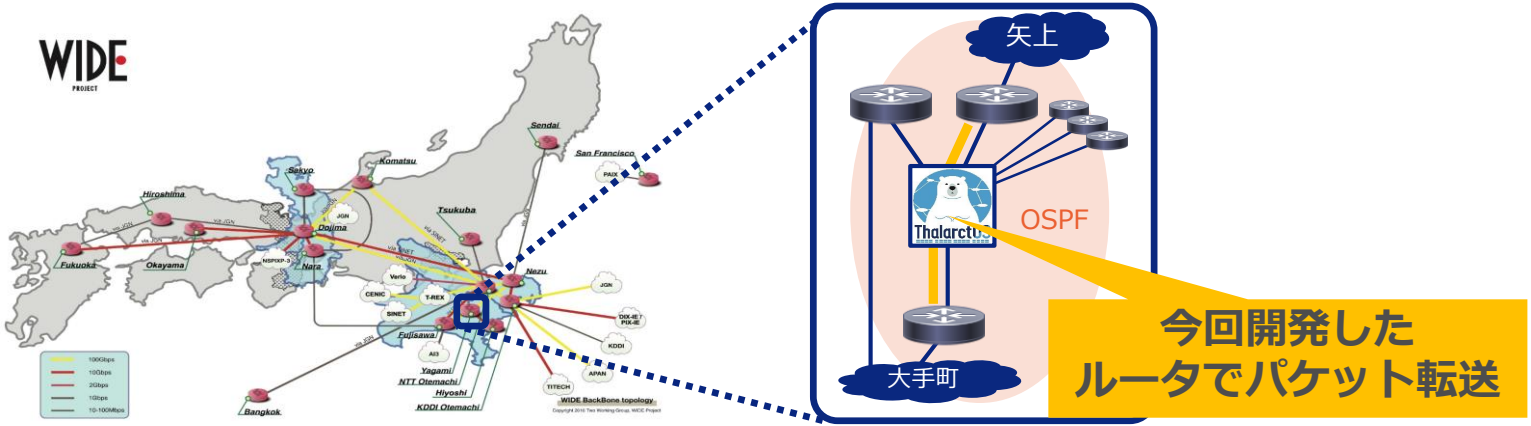
今後の展望

今後の展望の前に…

■ WIDE網で実験を開始しました

- 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



今後の展望

- **展望1**
  運用機能の拡充 (telemetry的な！)

- **展望2**
  UIの統一

- **展望3**
  新ASIC(Jericho2)への対応

<参考>Jericho2 特徴
・9.6Tb/s forwarding capacity
・SR/MPLS/VPLS/E-OAM etcサポート
・フルルート保持可能※ 1200万(v4)、600万(v6) ※外部メモリ利用時
・25GE/50GE/100GE/200GE/400GE IFサポート
・ディープバッファ/H-Qos など

まとめ

- WBS活用方法の検討のためにルータ開発を実施
  - -オープンソースを最大限活用
  - -HW/ベースOSとの依存関係を減らしたソフトウェアデザイン

- WBSを使ったルータ開発により知見を得た
  - -ASICの扱い方
  - -WBS周りの構成技術

ディスカッション_Q&A

●ディスカッションしたい事

・オープンソースを利用した開発について

オープンソースを使用することで開発ボリュームを抑えることができるが、
機能追加やbug Fixはコントロールできない。

・ルータ用途のホワイトボックススイッチ(ルータ)について

導入を検討されている方いますか？
どんな領域が適切と考えていますか？
(セルサイト、エッジルータ、コアルータ、ピアルータなど)


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

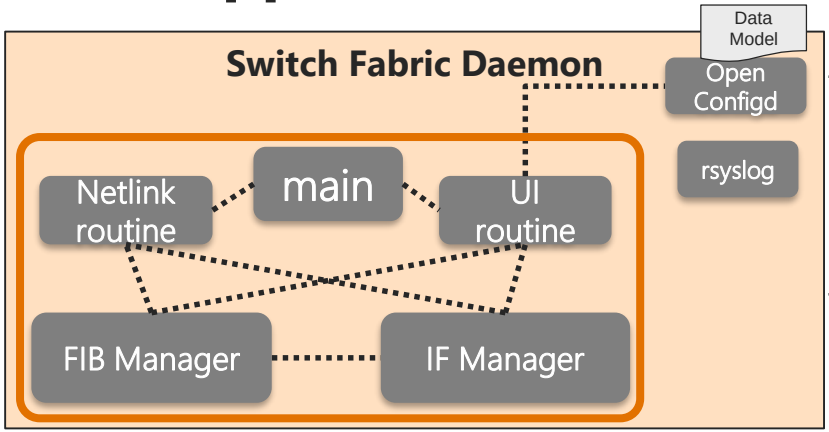
Appendix1-ThalarctOSロゴ

ロゴ作りました！

ホワイトボックススイッチ用のOSなので、
“白”熊をモチーフに！

愛称：白熊くん、タラちゃん、鱈

Appendix2-Switch Fabric Daemon 各コンポーネントの役割



【コンポーネントを分けた理由】
・依存関係を減らして、メンテナンス性を向上
・複数のチームで分担して開発
・コンポーネント単位での試験実施→生産性向上

■IF manager

ASIC初期化、パント処理の定義
WBSのL3IFのテーブル管理(アドレス、Link status、MTU、vlan等)

■Netlink routine

kernelからのnetlinkメッセージを受信/解析
FIB/interface Managerに情報書き込み

■FIB manager

IPv4/v6のFIBの維持/管理
ARP/NDテーブルの管理
ASIC上の経路情報の書き込み/消し込み

■UI routine

Command line インターフェースの提供