FUJITSU

shaping tomorrow with you

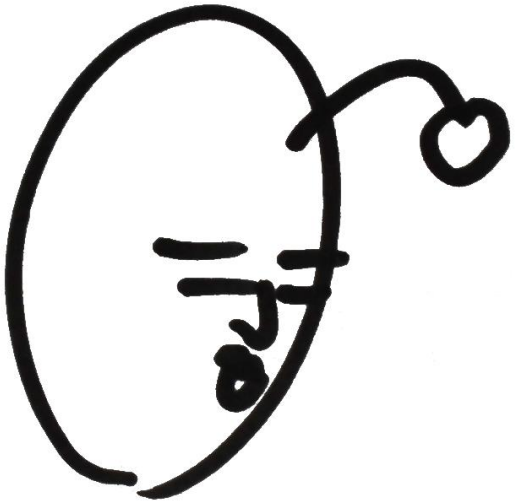
P4, eBPF and SmartNIC Data Plane Acceleration

Open Source Networking Days Tokyo (2018/10/23)

富士通アドバンストテクノロジ株式会社
システム技術統括部
谷所 基行

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

自己紹介



自画像	
画数	8[画]
描画時間	5[s]

名前

- 谷所 基行（たにしょ もとゆき）

仕事

- NPU-μCode、ASSPドライバー
- SmartNIC/WBS評価

（最近）

- TSN、OTN

（その他）

- たまーに、ハッカソンに参加

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

アジェンダ

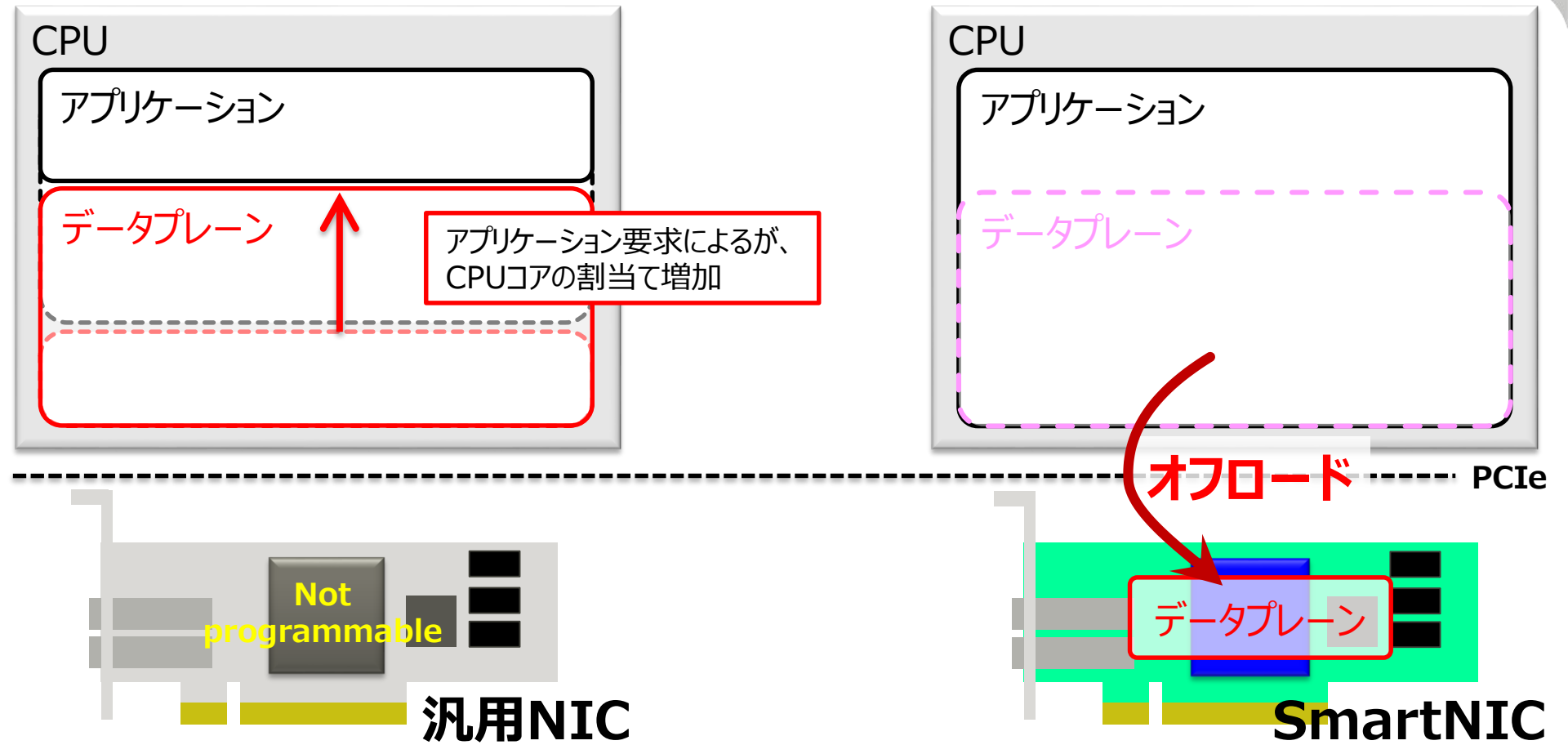
FUJITSU

- データプレーンアクセラレーションの動向
- SmartNIC アクセラレーション
- P4の紹介
- P4 → eBPF アクセラレーション
- P4 → eBPF 検証
- まとめ

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

データプレーンアクセラレーションの動向

FUJITSU



CPUによる
アクセラレーション

コア割当て増加
（アプリ割当て困難）

SmartNICによる
アクセラレーション

- **性能向上**
- **アプリ割当て最適化**

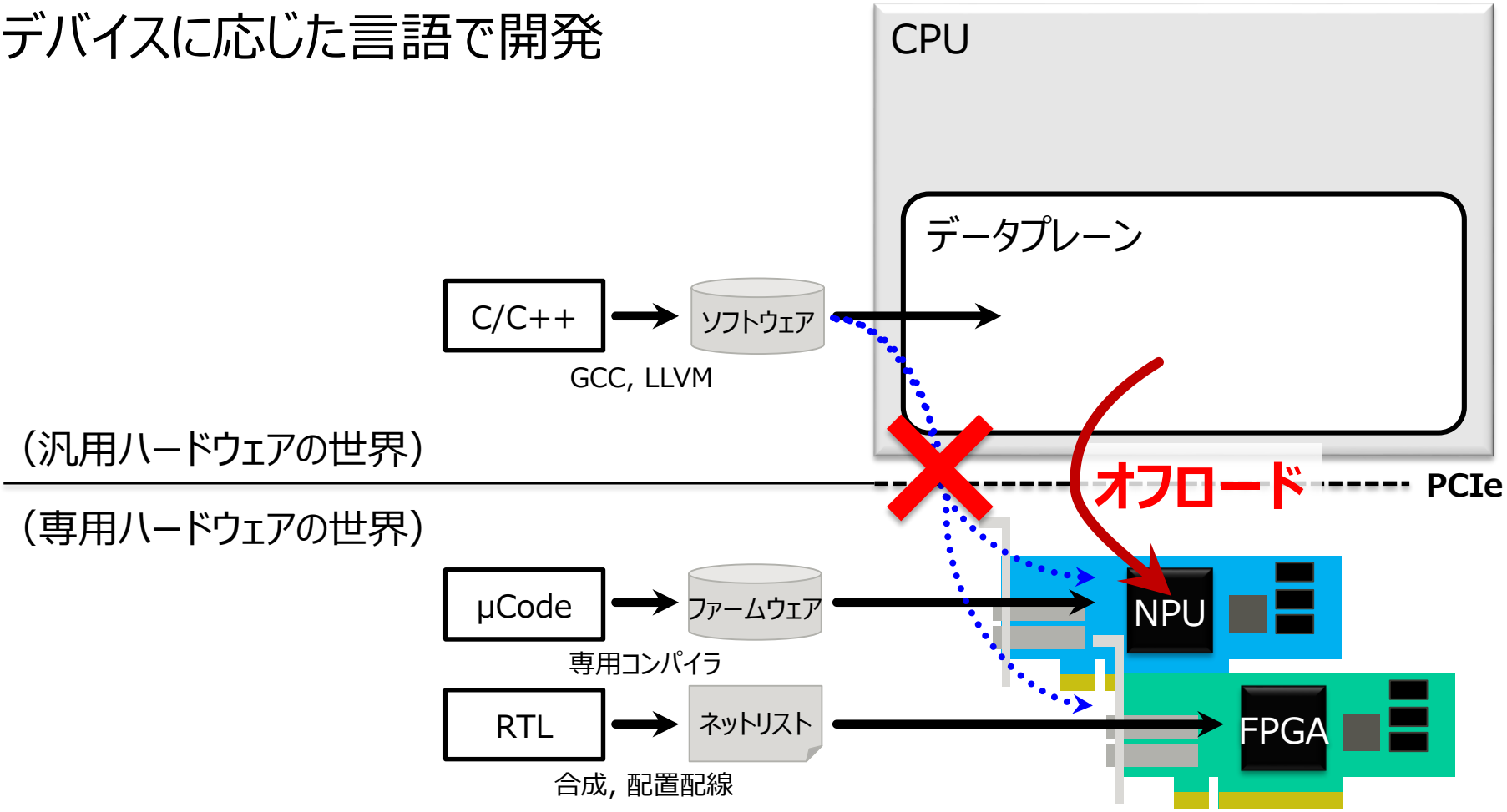
SR-IOV 2007~　**DPDK** 2011~　**XDP** 2016~

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

これまでのオフロード方法

FUJITSU

■ デバイスに応じた言語で開発



1. デバイスのほぼ全般の知識が必要
   → 熟練度に依存する
2. 設計資産が流用できない

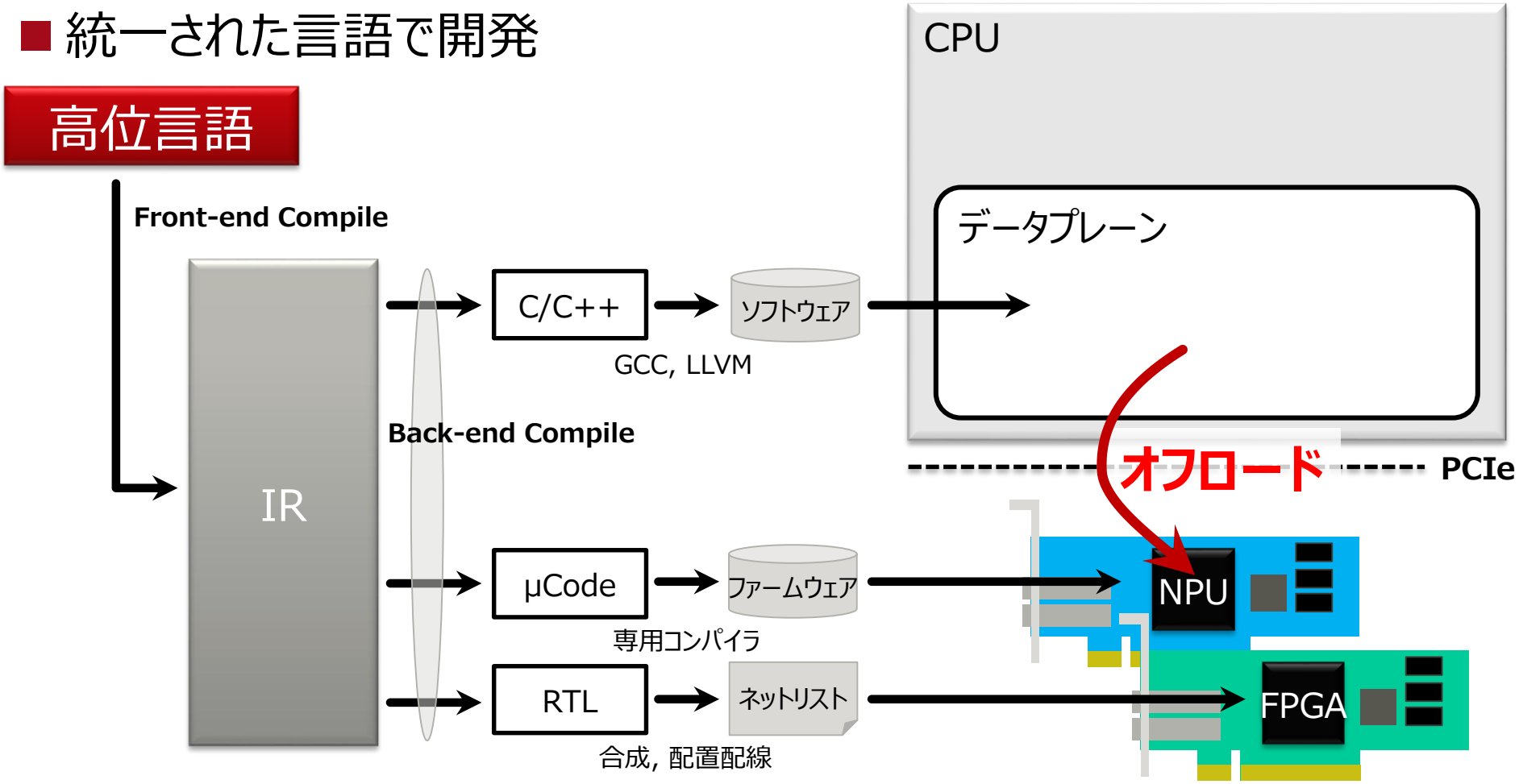
容易にオフロードできる開発手法が必要

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

これからのオフロード方法

FUJITSU

■ 統一された言語で開発



1. 開発手法が統一され、必要なスキルセットが少なく済む
   → 熟練度に依存しない
2. 設計資産が流用できる

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

P4の特徴

FUJITSU

- 標準化団体：P4.org（https://p4.org/）50社以上
  - デバイスベンダーや装置ベンダー、DC事業者など

- ドメイン特化型高位言語：データプレーンを記述
  - プロトコル非依存、デバイス非依存
  - バージョン：$P4_{14} \rightarrow P4_{16}$
    - 少しずつ、$P4_{16}$で動くハードウェアが出てきています
- Workshop：開催頻度年1,2回（https://p4.org/events/）
- P4を使った取組み
  - アプリケーション　：INT, DNS, Firewall, etc.
  - コミュニティ　　：GitHub上で公開されている(Open-NFPなど)

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

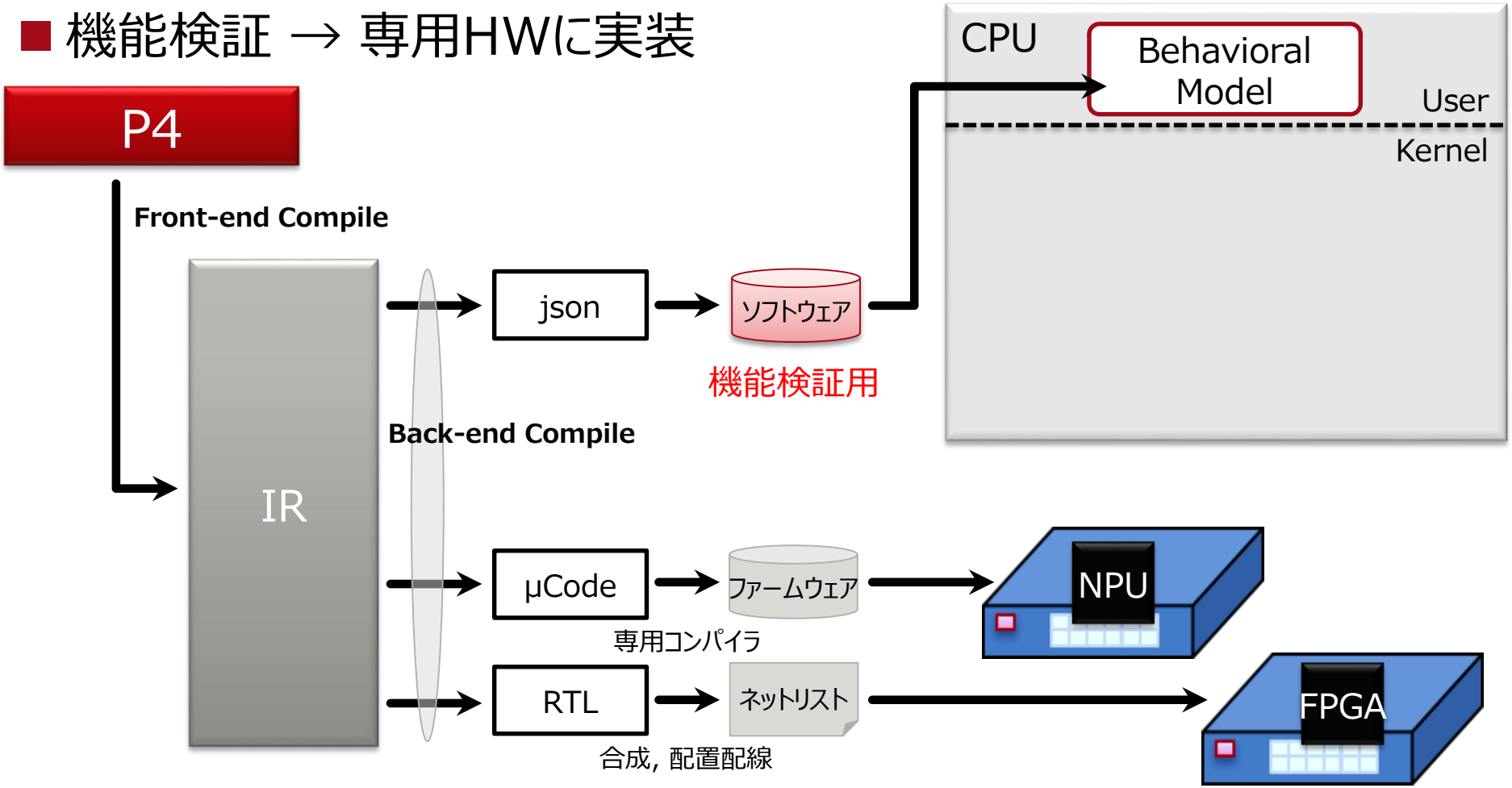
P4 → eBPF
データプレーンアクセラレーション

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

P4を使った開発プロセス

FUJITSU

■ 機能検証 → 専用HWに実装



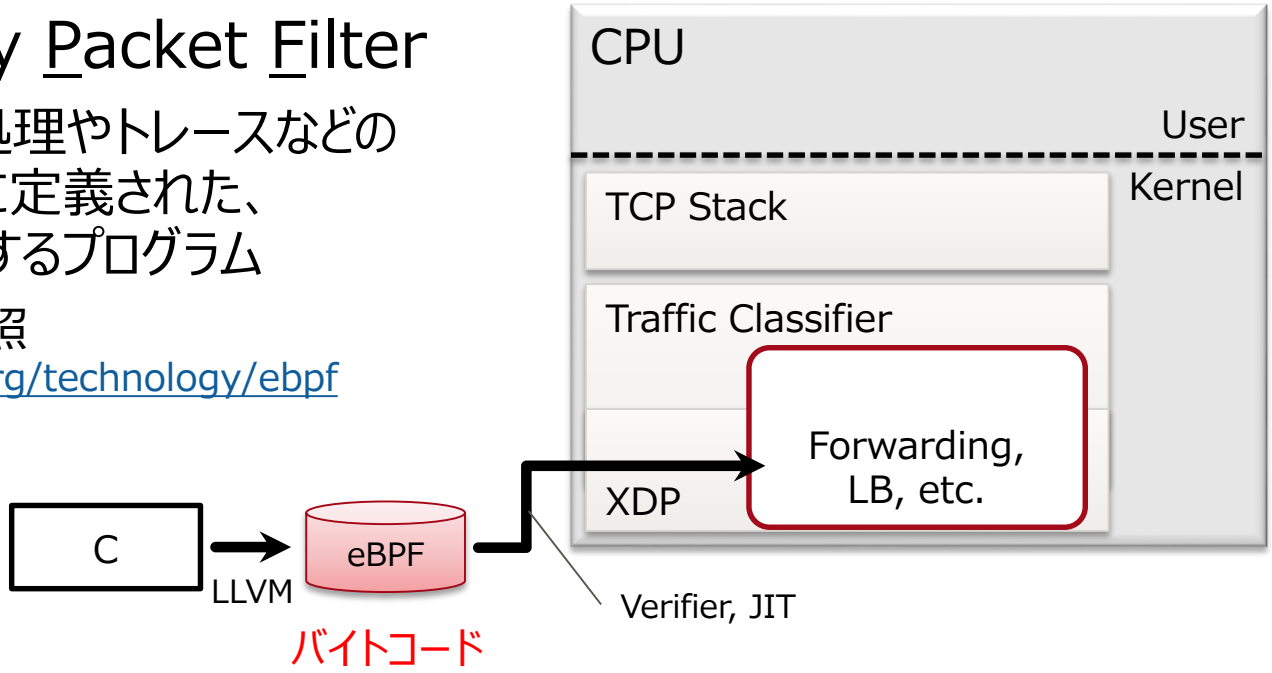
Behavioral Model：CPUをターゲットとしたアクセラレーションは想定していない

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

CPUによるアクセラレーション：eBPF

FUJITSU

- extended Berkley Packet Filter
  - Kernel内のパケット処理やトレースなどの機能を拡張するために定義された、仮想マシン上で動作するプログラム
    - 詳細はIOVisorを参照 https://www.iovisor.org/technology/ebpf



➢ XDP(eXpress Data Path)と呼ばれる、Kernel内でパケットを受信した直後のドライバーで動作させることで性能向上を実現している

(eBPFを含む)Linuxカーネルの知識が必要
→ 特殊な知識

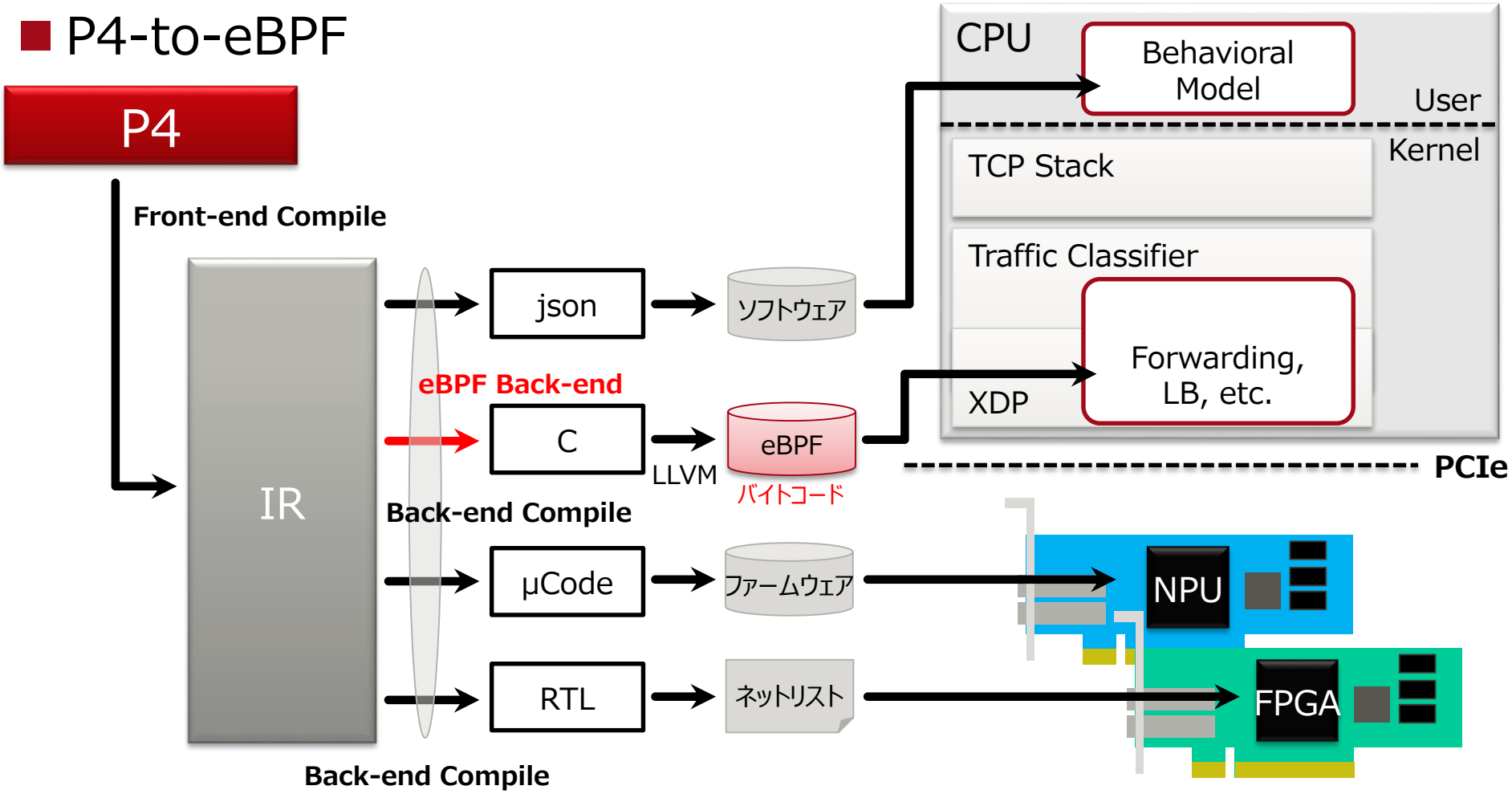
容易にアクセラレーションできる開発手法が必要

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

P4からeBPFに変換するルート

FUJITSU

■ P4-to-eBPF

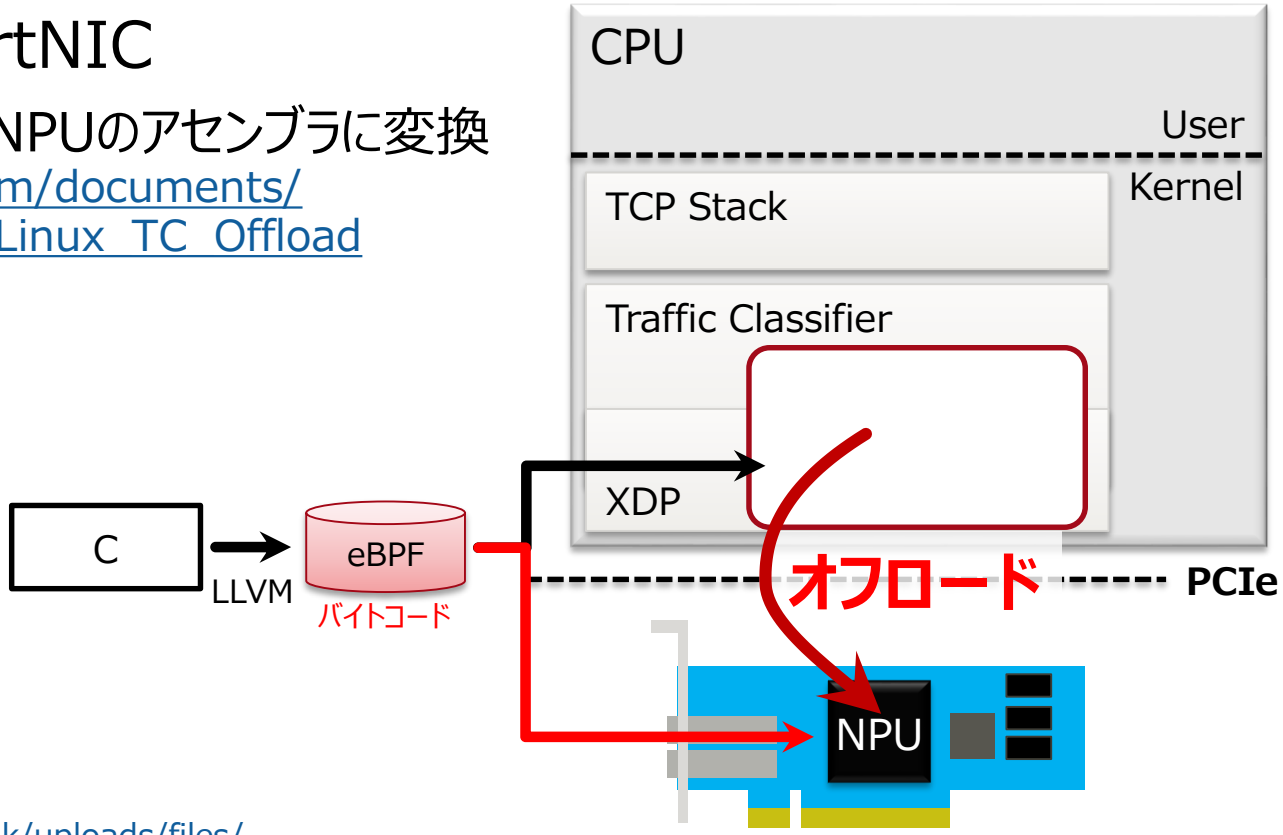


P4-to-eBPFによって、CPUのアクセラレーションも可能

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

eBPFのオフロード

- NPU-based SmartNIC
  - eBPFバイトコードを、NPUのアセンブラに変換
    https://open-nfp.org/m/documents/Open_NFP_P4_EBPF_Linux_TC_Offload_FINAL_5JHLETS.pdf



- その他
  - FPGA-based
    - https://netlab.dcs.gla.ac.uk/uploads/files/d99abd5bbadbed8c0f29808ee812bd26.pdf

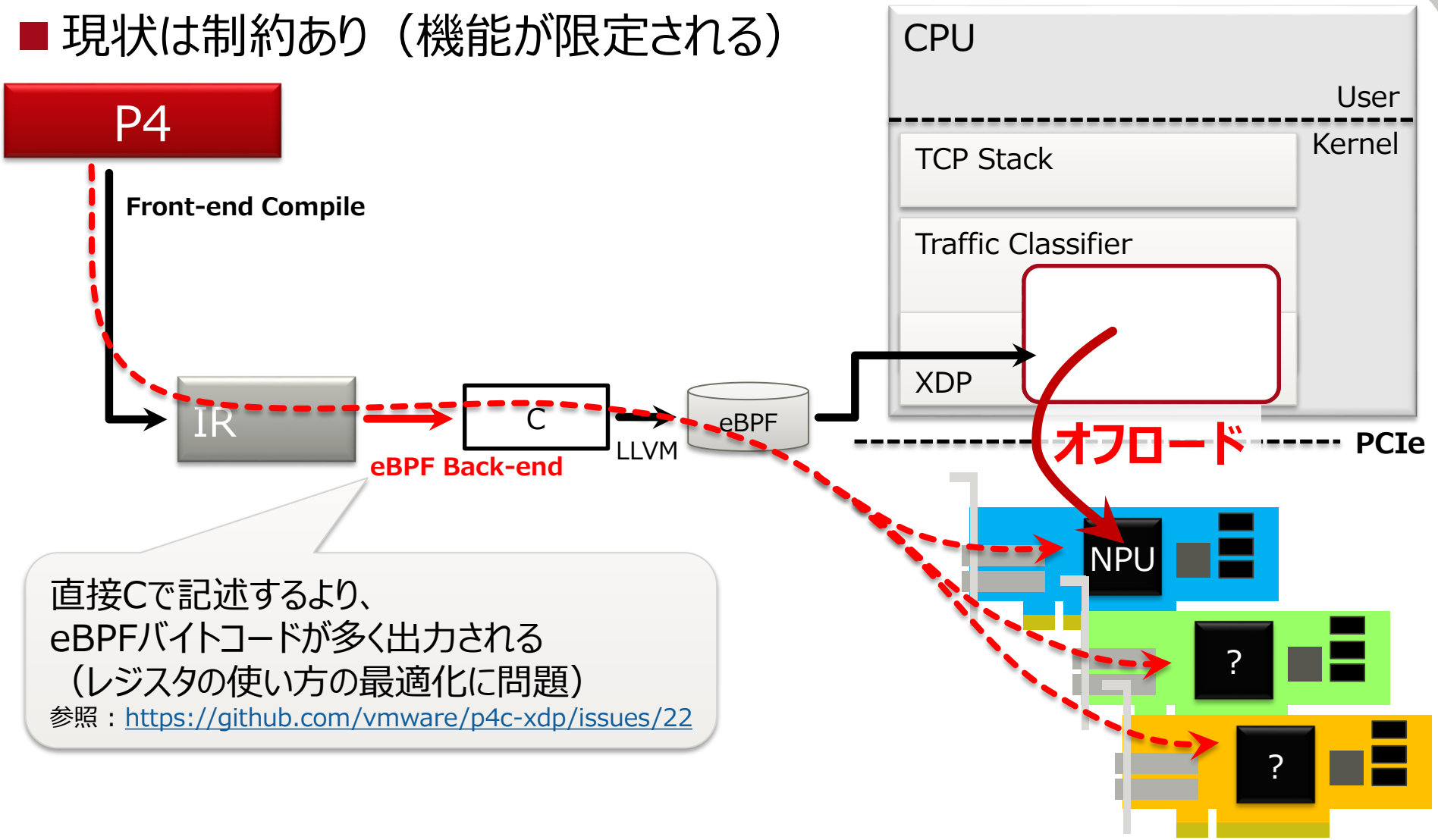
eBPFのオフロードによる性能向上、およびCPUリソースの効率化が可能

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

P4 → eBPF → オフロード

FUJITSU

■ 現状は制約あり（機能が限定される）



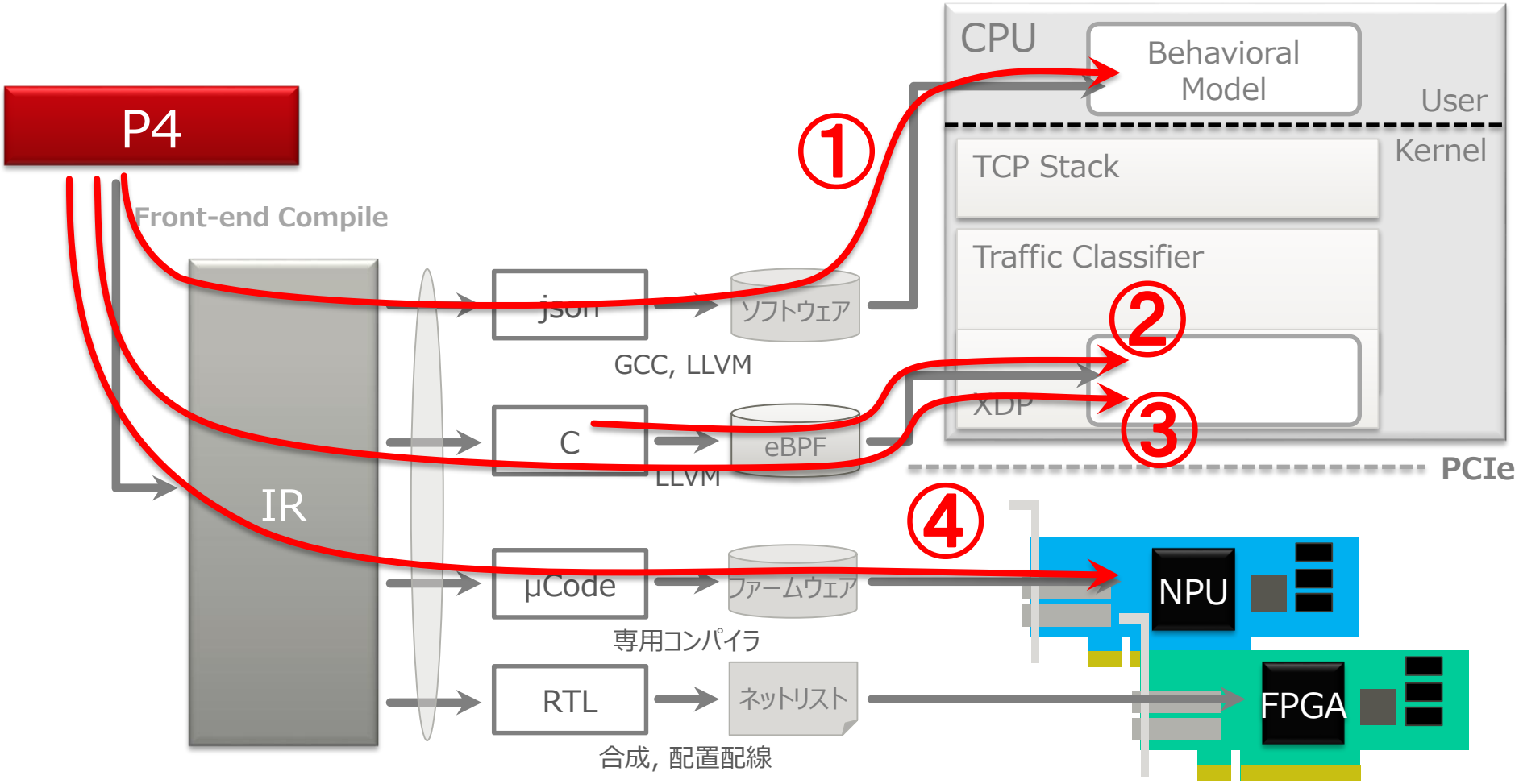
直接Cで記述するより、
eBPFバイトコードが多く出力される
（レジスタの使い方の最適化に問題）
参照：https://github.com/vmware/p4c-xdp/issues/22

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

P4→eBPF 検証

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

検証パターン



Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

検証コード

FUJITSU

- ①、③、④は同一のP4コード（②はCで記述）
- L4ロードバランサー
  - IPアドレス, TCP/UDP Portで振り分け。宛先ごとにIPv4でカプセリング
  - 受信したポートに折り返す。

```
#include <core.p4>
#include <xdp_model.p4>

struct local_metadata_
    bit<48> dstMAC;
    bit<48> srcMAC;
    bit<4>  version;
    bit<4>  ihl;
    bit<8>  diffserv;
    bit<16> totalLen;
    bit<16> identifica
    bit<3>  flags;
    bit<13> fragOffset
    bit<8>  ttl;
    bit<8>  protocol;
    bit<16> hdrChecksu
    bit<32> srcIP;
    bit<32> dstIP;
```

ヘッダー定義

```
control ingress(inout headers hdr, in xdp_input xin, out xdp_output xout) {
    bool xoutdrop = true;
    metadata meta;
    action _drop() {
        xoutdrop = true;
    }
    @name(".gen_hash") action gen_l
        hdr.inner_ipv4.setValid();
        meta.local_metadata.versio
        meta.local_metadata.ihl = l
        meta.local_metadata.diffse
        meta.local_metadata.totalL
        meta.local_metadata.identi
        meta.local_metadata.flags =
        meta.local_metadata.fragOf
        meta.local_metadata.ttl = l
        meta.local_metadata.protoc
```

Ingress処理

```
    @name(".loadbalance") action loadbalance(bit<32> dstIP, bit<48> dstMAC) {
        hdr.outer_ipv4.srcIP = 32w0xa000001;
        hdr.outer_ipv4.dstIP = dstIP;
        hdr.ethernet.dstMAC = dstMAC;
        xoutdrop = false;
    }
    @name(".ig_def") table ig_def {
        actions = {
            _drop;
        }
        default_action = _drop;
    }
    @name(".udp_hash") table udp_hash {
        actions = {
            gen_hash;
        }
```

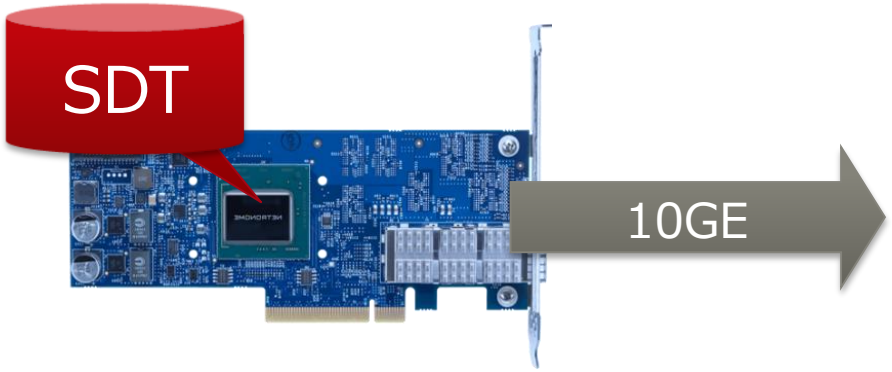
Match Action

P4で記述するメリット：eBPF(XDP)などカーネルの知識は不要

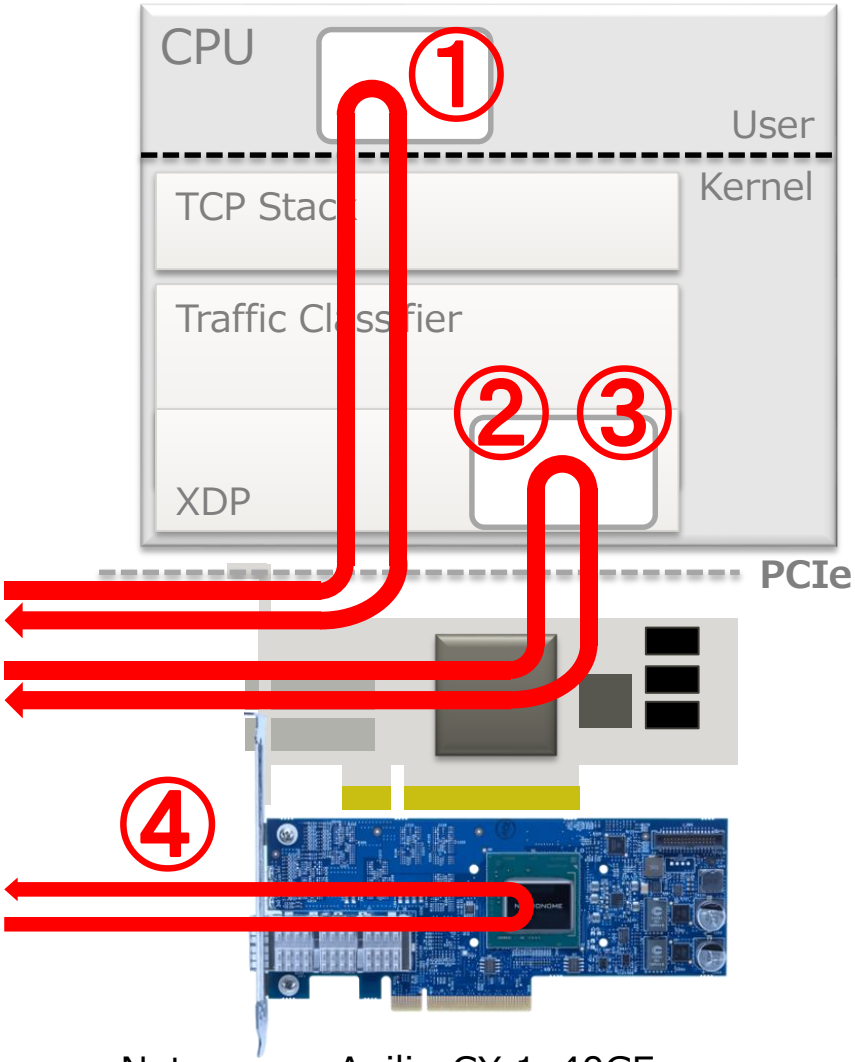
Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

検証構成

① P4 → Behavioral Model
② C → eBPF(XDP)
③ P4 → C → eBPF(XDP)
④ P4 → オフロード



パケットジェネレーター（Software-Defined Tester）



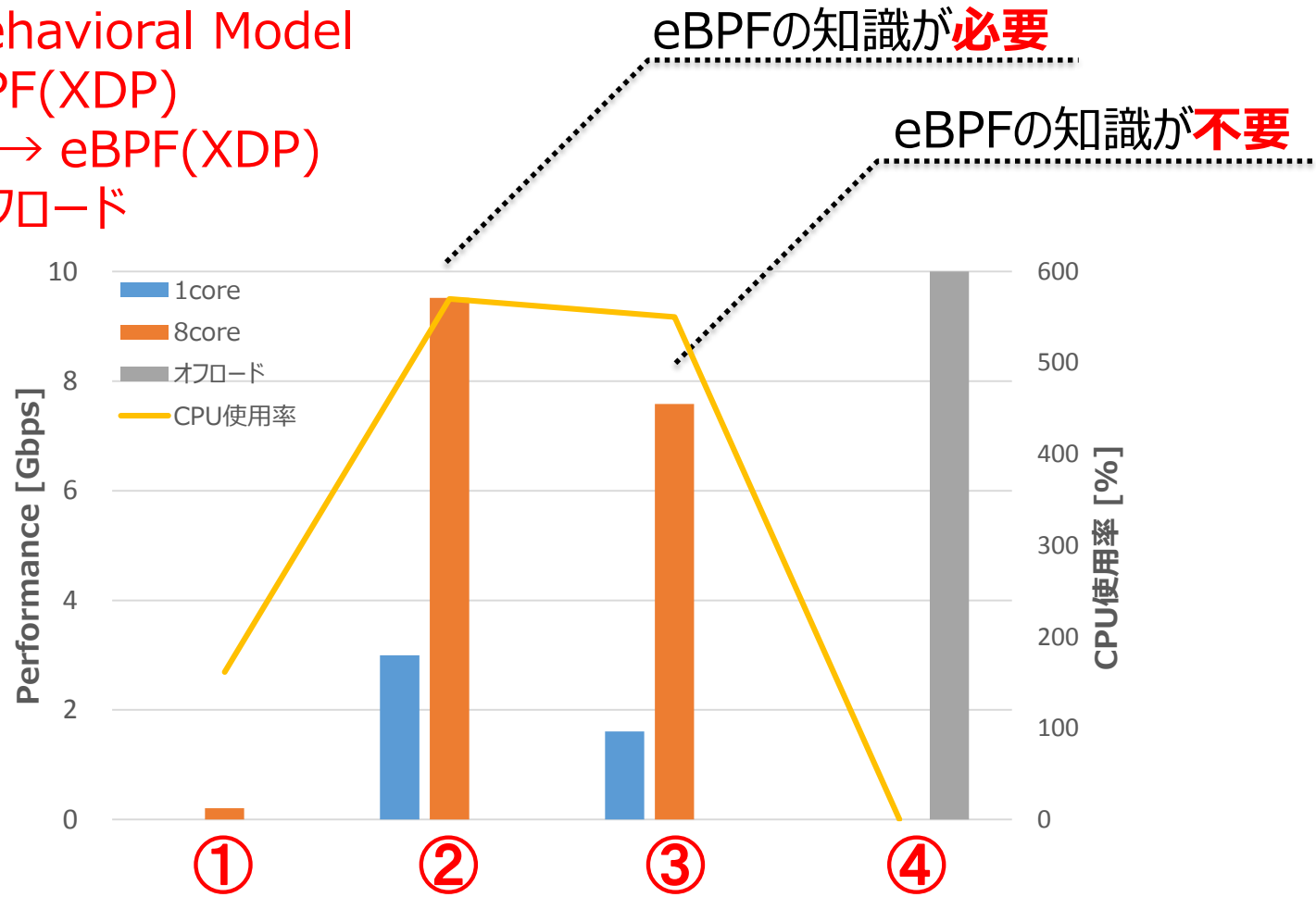
Software-Defined Tester
http://www.fujitsu.com/jp/group/fatec/services/platform/software-defined-tester/

Netronome Agilio-CX 1x40GE
https://www.netronome.com/media/documents/PB_Agilio_CX_1x40GbE.pdf

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

検証結果

① P4 → Behavioral Model
② C → eBPF(XDP)
③ P4 → C → eBPF(XDP)
④ P4 → オフロード



1. eBPFの知識がなくても、性能向上が得られる：P4の方が効率的
2. オフロードすることで、CPUリソースの消費もなくなる

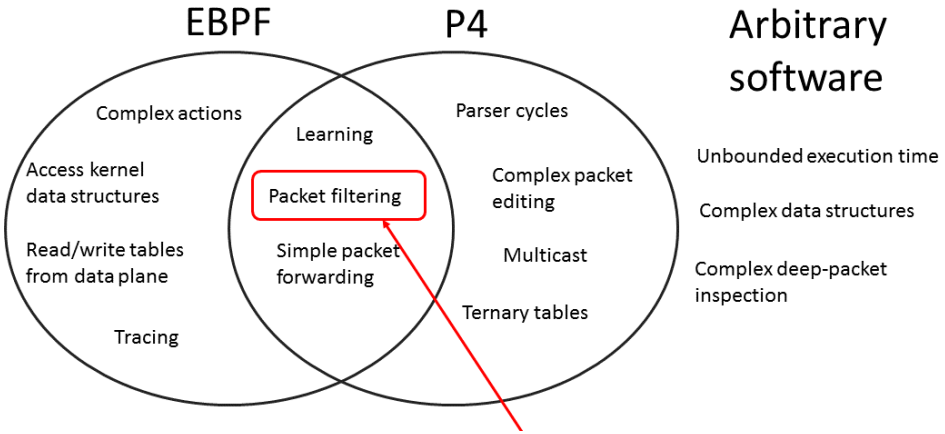
Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

P4は万能か?

FUJITSU

- P4
  - できること：プロトコル処理
  - できないこと：デバイスに依存する機能（QoSなど）
  - 参考：Okinawa Open Days 2017
    https://www.okinawaopenlabs.com/ood/2017/wp-content/uploads/sites/4/2017/12/fujitsu_3.pdf

- P4-to-eBPF
  - できること：プロトコル処理
  - できないこと：カーネル固有の機能



https://github.com/p4lang/p4c/tree/master/backends/ebpf

P4でできることから共通資産化し、開発効率化につなげていくもの

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

まとめ

FUJITSU

■ P4

できることから共通資産化し、
データプレーンアクセラレーションを
効率化する技術

新しいサービス、新しいニーズに
対する迅速な対応につながる

Copyright 2018 FUJITSU ADVANCED TECHNOLOGIES LIMITED

FUJITSU

shaping tomorrow with you