

～ P4から探る ～

# サーバーサイド高速パケット処理の現状と展望

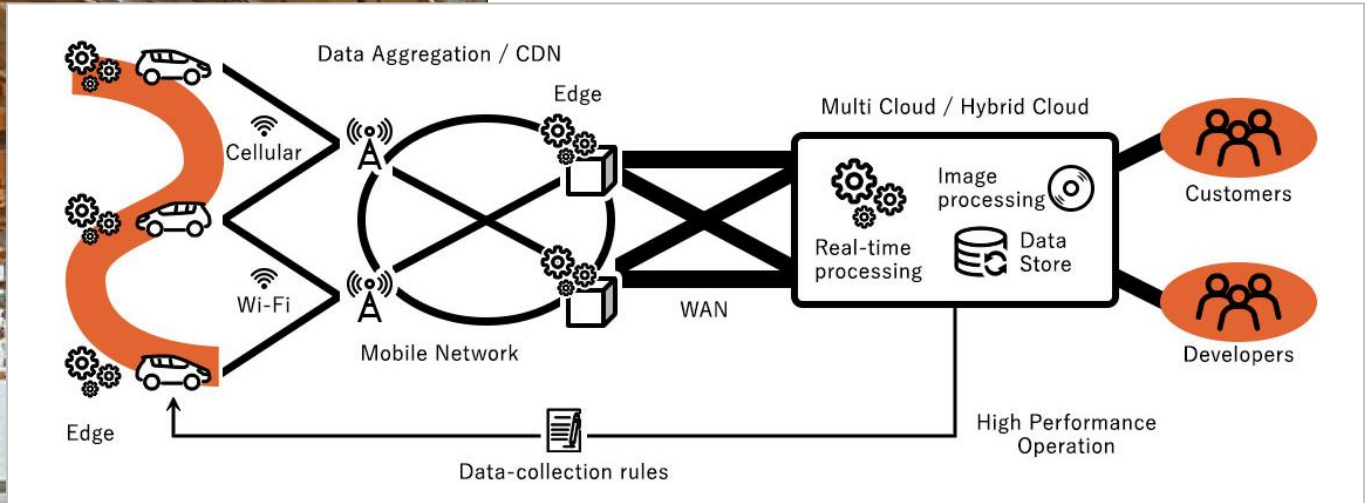
海老澤 健太郎 @トヨタ自動車 株式会社

コネクティッド先行開発部 | InfoTech | DCインフラG



TOYOTA@大手町  
 トヨタ自動車株式会社  
 コネクティッドカンパニー  
 コネクティッド先行開発部  
 InfoTech, DCインフラG

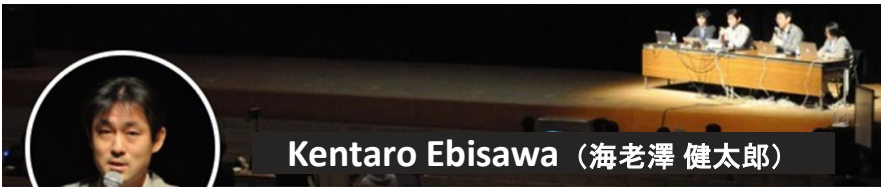
<https://www.toyota-tokyo.tech/>



## 研究・開発

「CASE」\*時代の技術革新を見据え、情報処理、AI技術、通信、データ分析などを活用したトヨタ独自の「つなぐ」技術の研究・開発を推進しています。それらの技術とトヨタのアイデンティティ・強みを融合することによって、お客様のニーズに応え、社会課題を解決する「モビリティサービス」の実現に挑戦しています。

\*C: Connected(コネクティッド)  
 A: Autonomous/Automated(自動化)  
 S: Shared(シェアリング)  
 E: Electric(電動化)



**Kentaro Ebisawa (海老澤 健太郎)**

<https://www.linkedin.com/in/ebiken/>

**10~20 years**


- Product Design and Development
- Management and Board member of Startup Companies
- Support Japan Market Entry
- Technical Consulting / Support
- Open Source Community

**~10 years**

- Product Intro and Support for Operators
- Data Center Operation (energy efficiency)
- Technical Team Management

**Network (Enterprise / Telecom / ISP)**

ATM, VPN(IPsec), xDSL, MPLS



Research & Support Engineer  
**Netmarks**  
Mar 1998 ~ Jun 2001

**Content Delivery Network & Storage**

Web/Streaming Cache & LB, NFS/SAN



Regional manager, Global Support Center  
**Network Appliance**  
Jul 2001 ~ Dec 2006

**Data Center & SaaS (MEX/SGI)**


Energy Efficiency, DesktopVPN (SaaS)



Director of Service Development Operation  
**SGI Japan**  
Feb 2007 ~ Sep 2008

**Switch Design & Development**

Flow Router, IPv6/v4 Translation (nat64)



Senior Product Manager  
**Sable Networks**  
Apr 2008 ~ Nov 2010

OpenFlow, FPGA, WhiteBox NOS



VP of Technology  
**Riava**  
Jul 2014 ~ Sep 2015



Co-Founder, CTO  
**Ponto Networks**  
Dec 2015 ~ Jun 2018




**Lagopus OF Switch**



**Service Automation & Container**


Operation/Business Support System



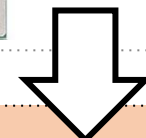
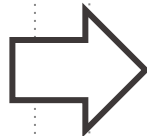
Solution Architect  
**Parallels**  
Dec 2010 ~ Mar 2014

**Network Automation**

SDN Controller, Orchestrator




Principal Engineer  
**Lumina Networks**  
Nov 2018 ~ Aug 2020



**current**

**Network Infra for Connected Cars**



Principal Researcher  
**TOYOTA Motor Corp.**  
Nov 2016 ~ Present

TOYOTA InfoTechnology (Nov 2016 ~ Mar 2019)  
Merged to TOYOTA Motor Corp on Mar 2019

**Operator Network Technology**



Research Professor  
**NTT Ltd.**  
Dec 2020 ~ Present

**Japan Market Entry & Tech Consultant**

**Terrasence**  
(a sole proprietor)  
Apr 2010 ~ Present

**P4の動向や、なぜサーバサイドが注目されているのか？を共有し、  
「我々はどのように活用可能か？」 「どのような課題の解決が必要なのか？」を議論する**

## P4の動向紹介

- P4の存在理由 (P4's raison d'être)
  - 2023 P4 Workshop @SanJose
  - データプレーンプログラミング言語 vs 定義言語
- データプレーンプログラミング言語 としてのP4
- データプレーン定義言語としての P4
  - データプレーン定義言語 としての役割
  - ハードウェア抽象化レイヤ (HAL) の重要性

## サーバーサイド・アクセラレーション

- サーバーサイド・アクセラレーション活性化の背景
- サーバーサイド・アクセラレーション 実現方式の分類
  - Software Based & Hardware Based
  - SmartNIC のタイプ
- サーバーサイド・アクセラレーションのユースケース
  - CPUのオフロード&セキュリティ向上
  - ネットワーク製品 (機能) の高速化・効率化
- サーバーサイド・アクセラレーション に向けたP4の拡張
- Server Side P4 の技術とコミュニティ

## まとめ&議論



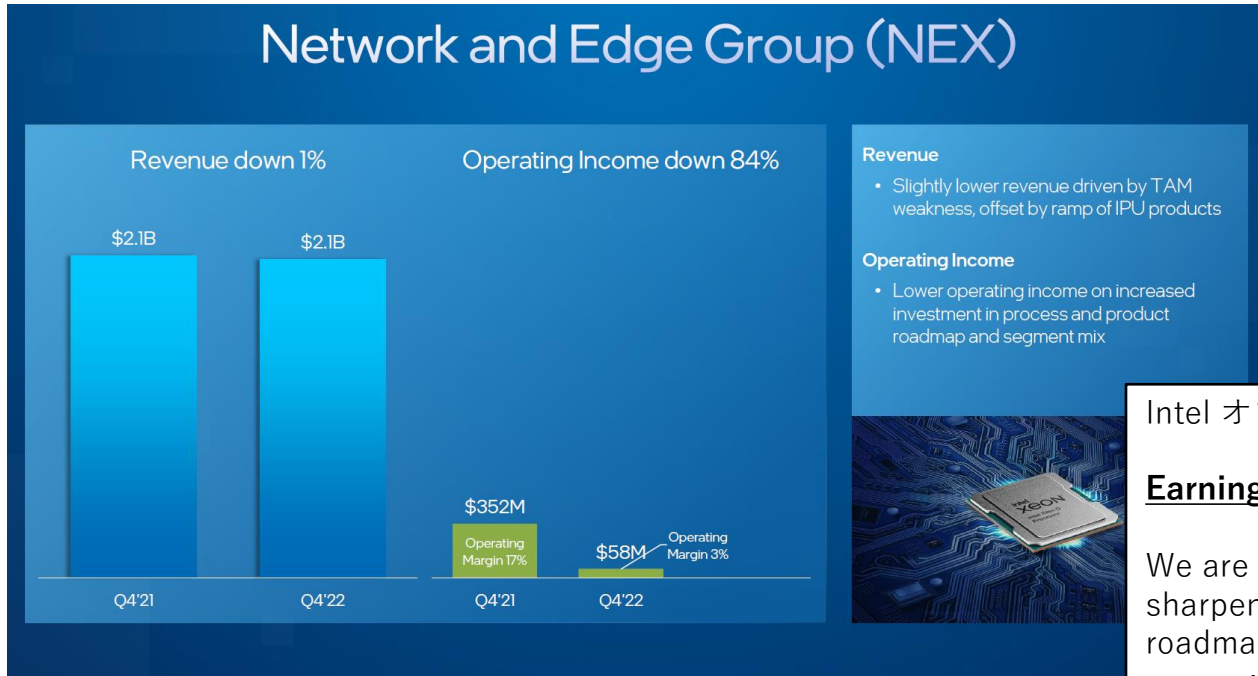
# P4の動向紹介

P4の存在理由 (P4's raison d'être)

2023 P4 Workshop @SanJose

# Intel, Tofino への投資凍結を発表

(2022年度 Q4 決算発表 : 2023/01/26 )



Intel オフィシャルコメント: <https://download.intel.com/newsroom/2023/corporate/q422-investor-call-remarks.pdf>

## Earning Comments from CEO Pat Gelsinger and CFO Dave Zinsner

We are making tough decisions to right-size the organization, and we further sharpened our business focus within our BUs (business units) by rationalizing product roadmaps and investments. NEX continues to do well and is a core part of our strategic transformation, but **we will end future investment on our network switching product line**, while still fully supporting existing products and customers. Since my return, we have exited seven businesses, providing in excess of \$1.5 billion in savings. We are also well underway to integrating AXG (Accelerated Computing Systems and Graphics) into CCG (Client Computing Group) and DCAI (Data Center and AI Group), respectively, to drive a more effective go-to-market capability, accelerating the scale of these businesses while further reducing costs.

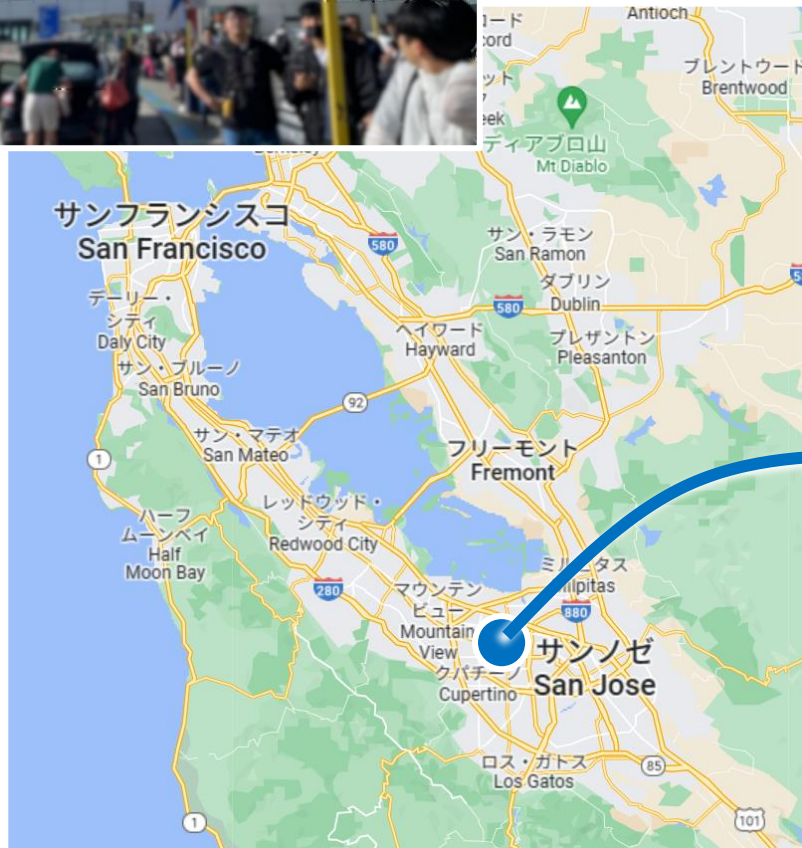
引用: "4<sup>th</sup> Quarter Earnings Presentation" @ <https://www.intc.com/>  
[https://d1io3yog0oux5.cloudfront.net/\\_2cead9b6413a1a91de449423742eea20/intel/db/887/8894/earnings\\_presentation/Q4%272022+Earnings+Deck\\_Final+PDF.pdf](https://d1io3yog0oux5.cloudfront.net/_2cead9b6413a1a91de449423742eea20/intel/db/887/8894/earnings_presentation/Q4%272022+Earnings+Deck_Final+PDF.pdf)

# 2023 P4 Workshop @SanJose (2023/04/25~26)



## 2019年@スタンフォード大学 以来 4年ぶりの現地参加のみのWorkshop

Tofino等P4対応ASICに限らず、Fixed Function ASICも含めたスイッチASIC及びIPU/DPU/FPGAを搭載した SmartNIC などのサーバーサイドターゲットを中心に、研究論文、技術解説、ユースケースの紹介など、多岐に渡るトピックが講演&議論された。特に今回は現地参加のみの開催という事もあり、質疑が活発で様々な立場での課題感や取組状況について把握できた。







Barefoot !? 😊



**Barebottle Brewing Company**

<https://www.barebottle.com/>





# P4's raison d'être (P4の存在理由)

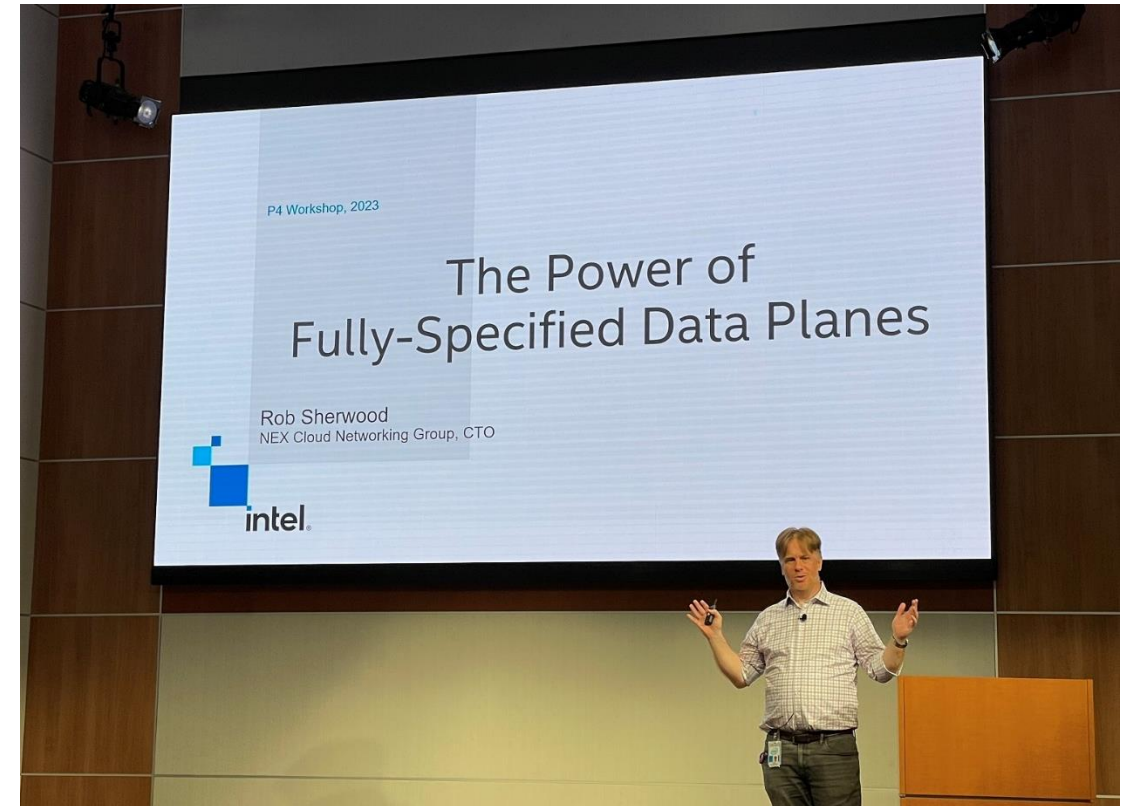
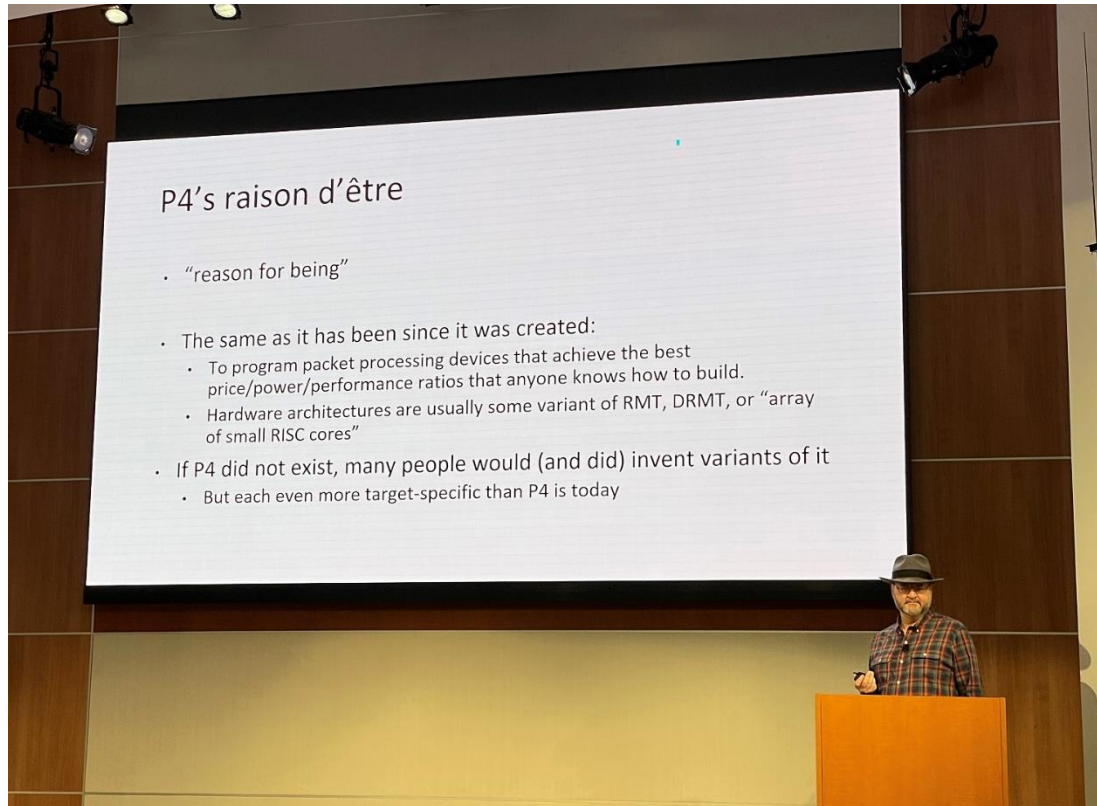
2023 P4 Workshop @SanJose ~ Key Note 1 & 2

## P4's raison d'etre, missing features and works done

Andy Fingerhut, Intel

## The Power of Fully-Specified Data Planes

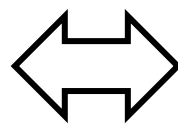
Rob Sherwood, NEX Cloud Networking Group, CTO



# P4の存在理由

## ~ P4's raison d'être ~

データプレーンプログラミング言語  
data plane programming language

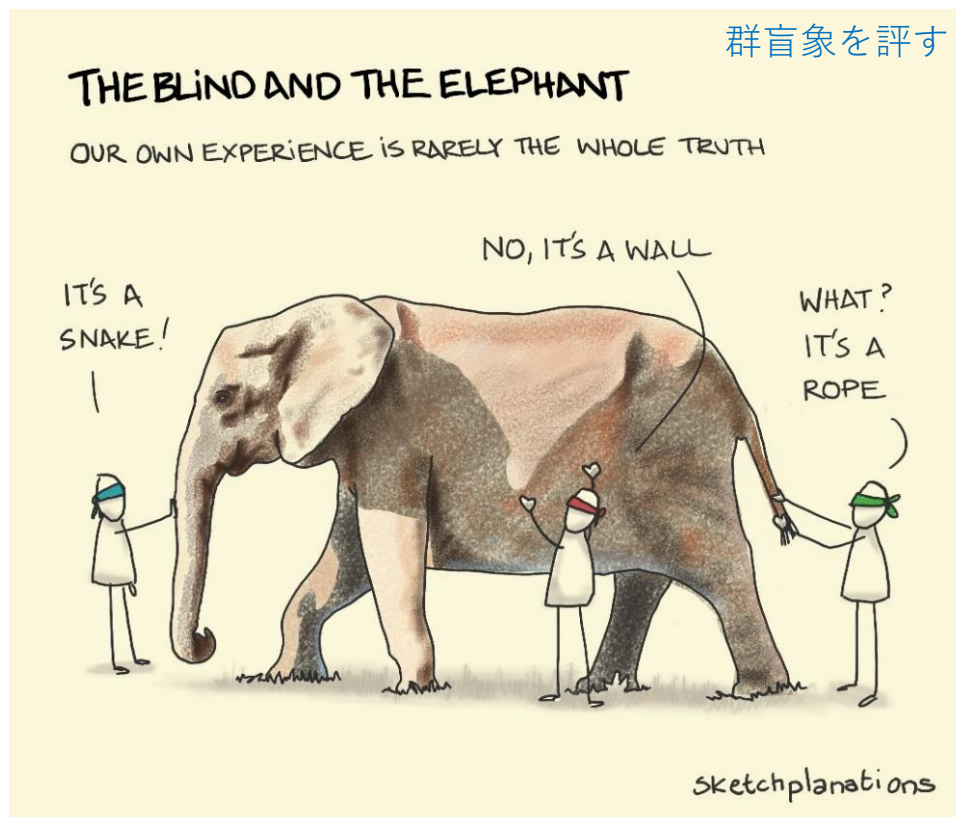


データプレーン（完全）定義言語  
fully-specified data plane language

高速パケット処理デバイス  
のプログラミング

ターゲット非依存の  
プログラミング  
(ASIC, FPGA)

データプレーン  
APIを自動生成  
(P4Runtime)



データプレーン  
抽象化レイヤ  
(Fixed Function ASIC への適用)

テストの自動生成

検証可能な形式言語  
Formal Language

The parable of the blind and the elephant  
<https://sketchplanations.com/the-blind-and-the-elephant>

# Slides & Video (Key Note)

<https://opennetworking.org/events/2023-p4-workshop/>

Title (amended by ebiken to make it easier to understand what's been talked)

	Key Note	In-depth Talk	Working Group	Lightning Talk	Demo	Poster
<b>Total 32 items</b>	7	8	1			16
Keynote 1 - P4's raison d'être, missing features and works done (Welcome + Recognition)	1					
Keynote 2 - The Power of Fully-Specified Data Planes (Intel)	1					
Keynote 3 - From Programmability to Fungibility (Rice University)	1					
Keynote 4 - AMD Pensando DPUs and Projects	1					
Keynote 5 - Fireside Chat with Nick McKeown	1					
Keynote 6 - Laconic: Streamlined LB for SmartNICs, Xenic: SmartNIC-Accelerated Distributed Transactions (University of Washington)	1					
Keynote 7 - P4 HAL for Network Virtualization (Google Cloud)	1					
I1 - Escaping Babel: The Flow Must Go On		1				
I2 - OpenConfig Co-Existence with P4 Using TDI		1				
I3 - Formalizing and Extending P4's Type System		1				
I4 - Effective DGA Family Classification using a Hybrid Shallow and Deep Packet Inspection Technique on P4 Programmable Switches		1				
I5 - Segment Routing Proxy Device Implemented Using P4 on FPGA with Zero CPU Overhead		1				
I6 - Hardware Offload Driver with P4-TC		1				
I7 - P4TC: Linux Kernel P4 Implementation Approaches And Evaluation		1				
I8 - Augmenting P4-DPDK software pipelines with accelerators: the IPsec use-case		1				
<b>P4 Working Groups - Reports &amp; Future of P4 Panel</b>			1			
L1 - Intent-based Platform leverages Programmable Networking for Optimizing Edge				1	1	1
L2 - A Language Engineering Approach to Support the P4 Coding Ecosystem				1		1
L3 - Enhancing Blockage Detection and Handover on 60 GHz Networks with P4 Programmable Data Planes				1	1	1
L4 - P4BS: Leveraging Passive Measurements from P4 Switches to Dynamically Modify a Router's Buffer Size				1	1	1
L5 - URRLC SLA Measurement Implemented Using P4 on FPGA Without Decreasing a Network Function Performance.				1		1
L6 - Enabling IPsec via P4Runtime and Openconfig				1		
L7 - Modular Code Parser for the P4 Language				1		
L8 - Sieve: Layered Network Defenses against Large-Scale Attacks				1		
L9 - A Testbench for Testing Programmable Traffic Managers in a Software Environment				1		1
L10 - Enabling P4 Hands-on Training using Hardware Switches in a Cloud System at the University of South Carolina				1	1	
Causal Network Telemetry					1	
Demo to Offload Networking Pipeline on Intel IPU E2000 Using P4 Control Plane					1	
HW Offload of L2 Forwarding P4 Program via P4-TC in Linux Kernel					1	
Kubernetes Networking Acceleration Using P4					1	
Extending the P4 Language to Facilitate the Use of Stateful Constructs						1
P4-Based Packet Tracing of Microservices for Service Mesh						1

表：タイプ毎のセッション件数集計（合計32件）

Session Type	件数
Key Note	7
In-depth Talk	8
Working Group	1
Lightning Talk (LT)	10
Demo/Poster (LT重複分を含む)	16



# 2023 P4 Workshop @SanJose ~ セッション内容

## • データプレーンプログラミング言語としての P4

- スイッチ ASIC ターゲットは Google が言及した程度
  - (参考) 2022 P4 Workshop
  - OPEN SRV6 PROJECT: OPEN SOURCE FOR P4-BASED EDGE ROUTER
  - [ALIBABA] THE JOURNEY TOWARDS PREDICTABLE NETWORK IN ALIBABA CLOUD
- サーバースイド (SmartNIC・CPU) ターゲットは多数あり

## • データプレーン定義言語としての P4

- データプレーン抽象化レイヤ ... スイッチ & SmartNIC
- テストの簡略化
- ハードウェア認証の簡略化
- ソフトウェア進化・オフロードの簡易化
- Formal Behavior Specification (形式仕様)

⇒ P4をデータプレーン抽象化レイヤとして利用したセッションが多数



# プログラミング言語としてのP4

スイッチASIC

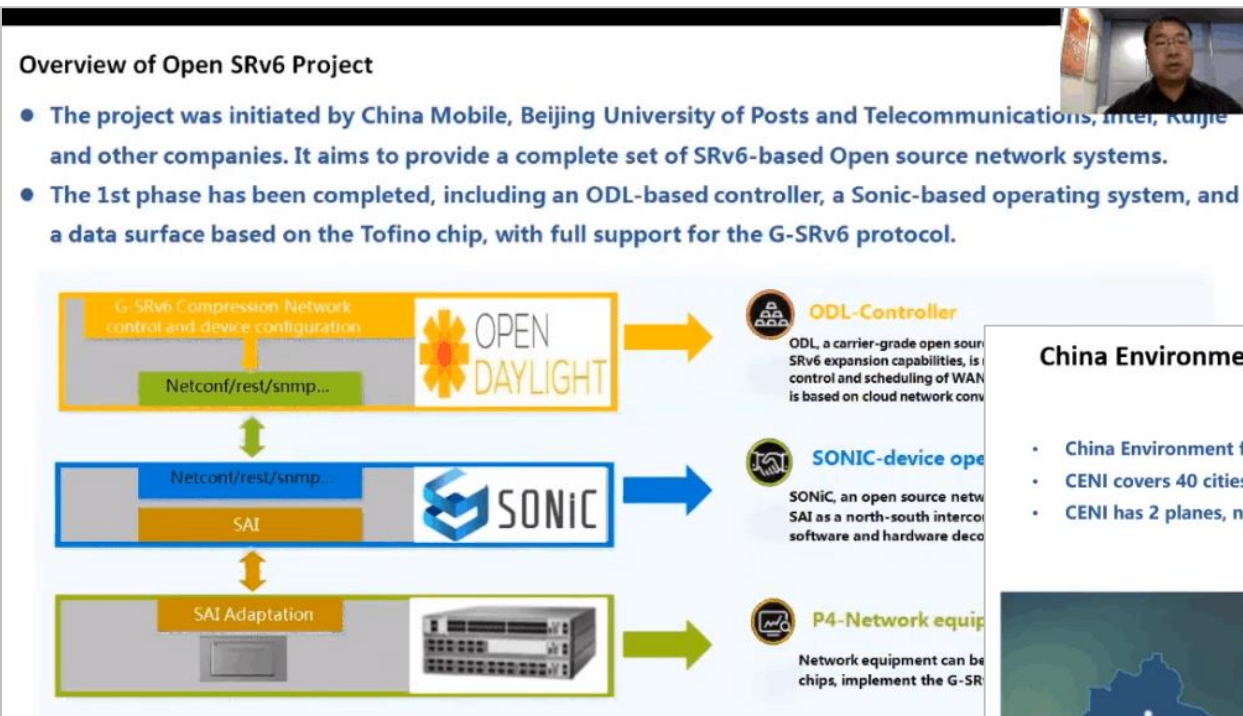
# データプレーンプログラミング言語としてのP4

ユーザが自由にプログラム可能な世界の実現  
(プログラマビリティの改善)

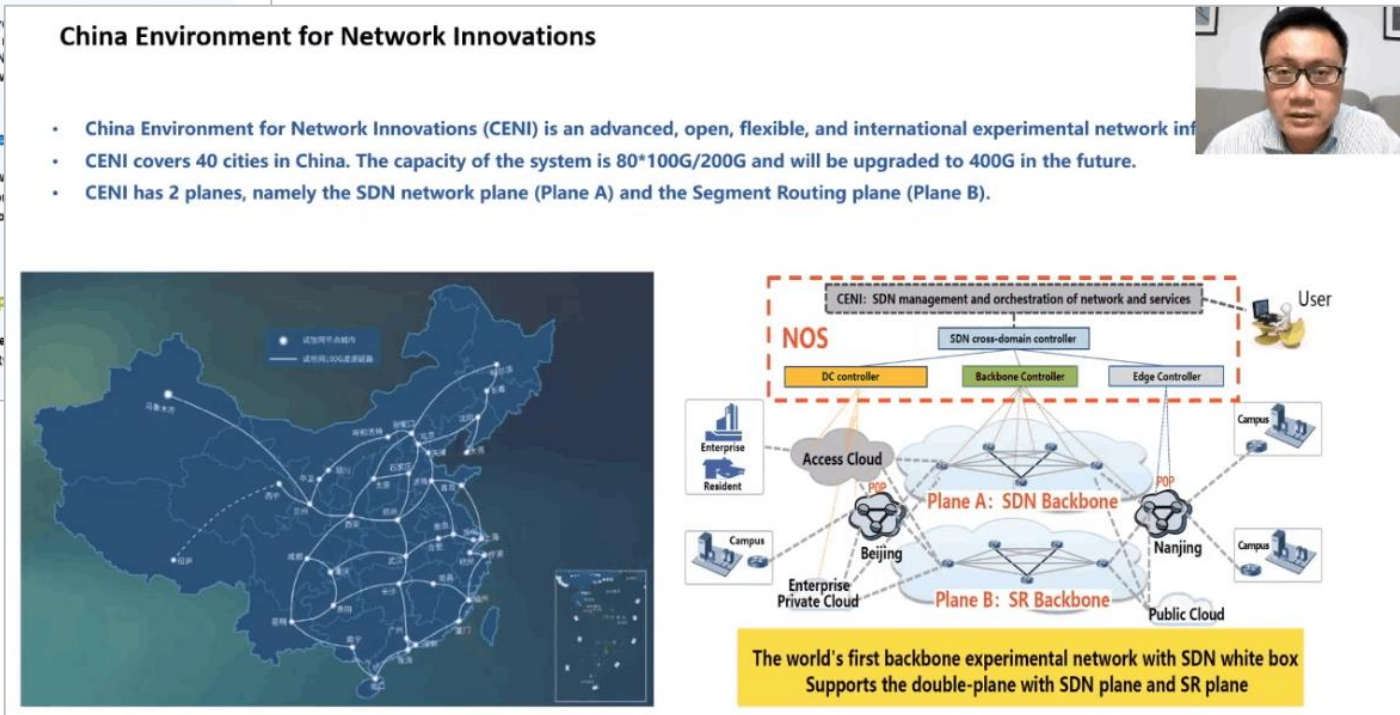
Target	課題
ASIC	<ul style="list-style-type: none"><li>機能追加や変更は限定的</li><li>変更にはマイクロコードプログラミングが必要</li><li>クローズド&amp;ベンダ依存 (ユーザはプログラム不可)</li></ul>
NPU	<ul style="list-style-type: none"><li>ベンダ毎に異なるプログラム言語やSDK</li></ul>
FPGA	<ul style="list-style-type: none"><li>HDL技術者の希少性</li><li>プログラミングの難易度 (高)</li></ul>
CPU	<ul style="list-style-type: none"><li>カーネルの変更 (Upstream) に時間が必要</li></ul>

# OPEN SRv6 PROJECT: OPEN SOURCE FOR P4-BASED EDGE ROUTER

CENI: China Environment for Network Innovations (中国40都市を繋いだ実験ネットワーク) G-SRv6 導入事例



- SRv6 ノード : P4 (Tofino ASIC) + SONiC
- コントローラ : OpenDaylight (ODL)
- L3VPN over G-SRv6 サービスの実証実験



発表動画(YouTube)

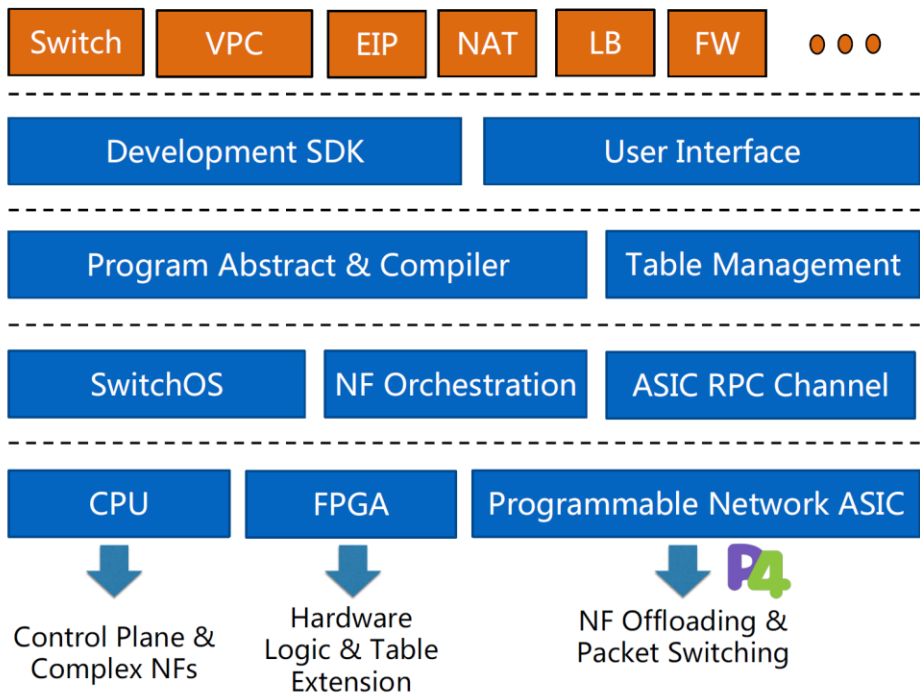
⇒ <https://www.youtube.com/watch?v=wHkqdd9HbVQ>

詳細解説 (Zenn記事)

⇒ [https://zenn.dev/ebiken\\_sdn/articles/e1ab7c9a803abd](https://zenn.dev/ebiken_sdn/articles/e1ab7c9a803abd)

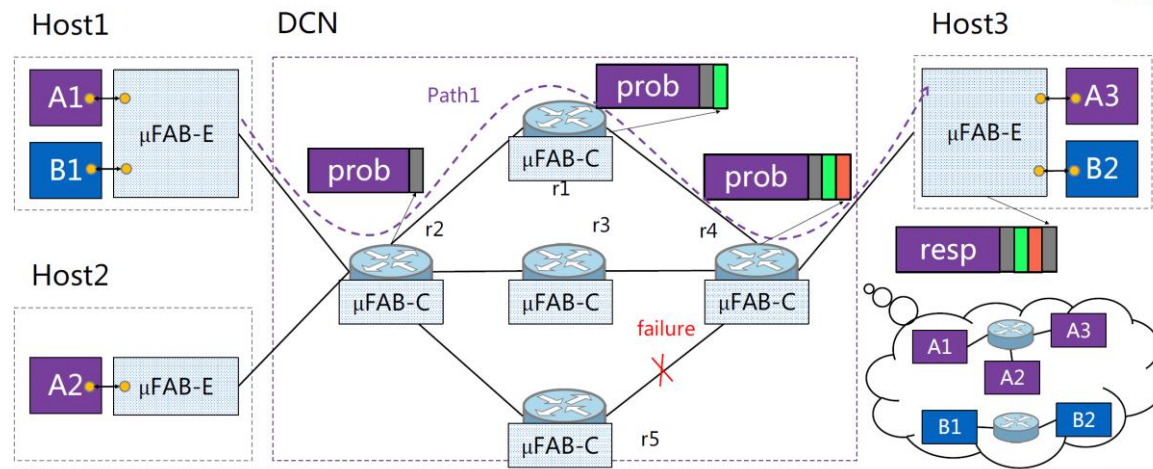
# THE JOURNEY TOWARDS PREDICTABLE NETWORK IN ALIBABA CLOUD

SNA\*: a hyper-converged programmable gateway



SNA ... Smart Network Appliance  
P4を用いた独自実装

μFAB: Predictable μFabric on Informative Data Plane



\* Smart

## 2022 P4 Workshop

"The Journey towards Predictable Network in Alibaba Cloud"

Dennis Cai, Head of Network Infrastructure

<https://opennetworking.org/2022-p4-workshop-gated/>

### uFab-E

- Send probes with fetch core information back
- Schedule packets to paths
- Control sending rate of each path with back-pressure

### uFab-C

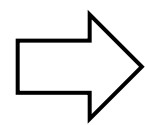
- Check failures locally and in neighbors
- Summarize total bandwidth subscription on each link
- Piggyback the preceding information with INT



# スイッチ ASIC のプログラミング言語としての P4

## 新しいプロトコル・機能の開発期間短縮

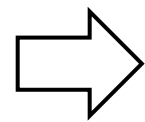
- 従来の Fix Function ASIC (3~5年) ⇒ Programmable ASIC (1~2年)



**Intel/Barefoot 以外の ASIC もプログラマビリティをサポート**  
**Cisco Silicon One (P4), NVIDIA Spectrum (P4?), Broadcom (SDKLT)**

## ユーザー固有のニーズを満たす機能の (早期) サポート

- G-SRv6 (CENI ... China Environment for Network Innovations )
- Smart Network Appliance (Alibaba)
- In-Network Computing for Hyperscale ML



**ユーザ自身による開発は Hyper Scaler か学術研究が中心**  
**多くの場合、ベンダがP4 (ASIC) プログラミングをサポート**  
⇒ **オープンな環境を維持できるエコシステムが構築できなかった**  
⇒ **ユーザサイドでのプログラマビリティはサーバサイドへ**

# Tofino 新規開発中止 ～ 背景と今後の対応

Nick McKeown Fireside Chat や Rob Sherwood (Intel NEX CTO) の Key Note 等で説明

- 新規開発中止の背景（従来アナウンス通り）
  - 景気後退により事業のリストラクチャリングが必要となった
  - IPU (Mount Evans) の売上が圧倒的に大きかった
  - 投資の優先度付けを迫られ、Tofino 3 の開発中止を決断
- Tofino 1, 2 の今後について（従来アナウンス通り）
  - 購入する人がいる限り Tofino 1, 2 製造 & サポートを継続
  - 機能や性能が要件満たす顧客からの新規受注が継続
  - Alibaba等、中国で圧倒的に売れている
- P4 研究開発への投資は継続
  - P4は既にスイッチ専用ではなく利用が広がっている
  - IPU (Mount Evans) はIntel戦略にとってのコア製品
  - DPDK (IPDK), P4TC など、ソフトウェアでのP4活用も継続



# Intel Vision: Common Programming Model with P4, 2023 ver

2021 P4 Workshop @online

"P4 AT INTEL"

Guido Appenzeller, CTO, Data Platforms Group, Intel

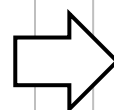
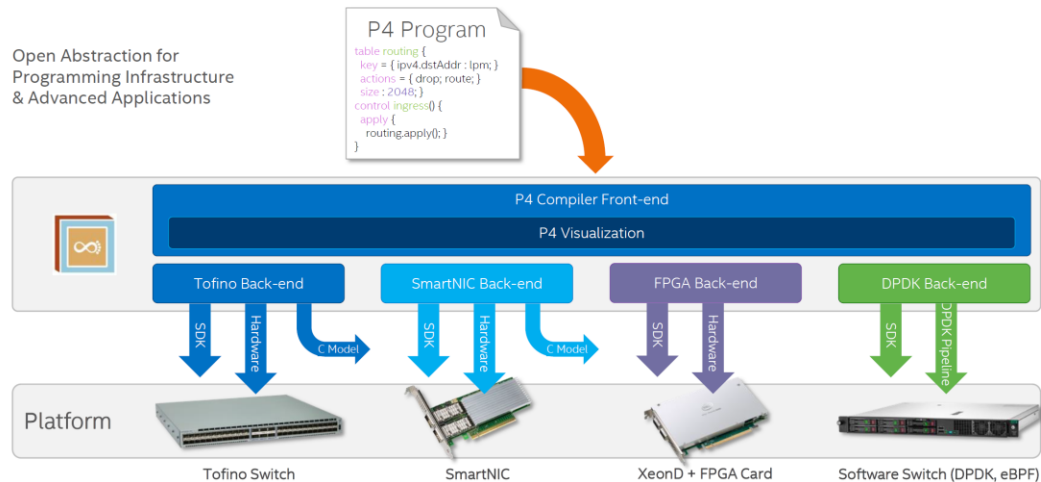
2023 P4 Workshop @SanJose

"The Power of Fully-Specified Data Planes"

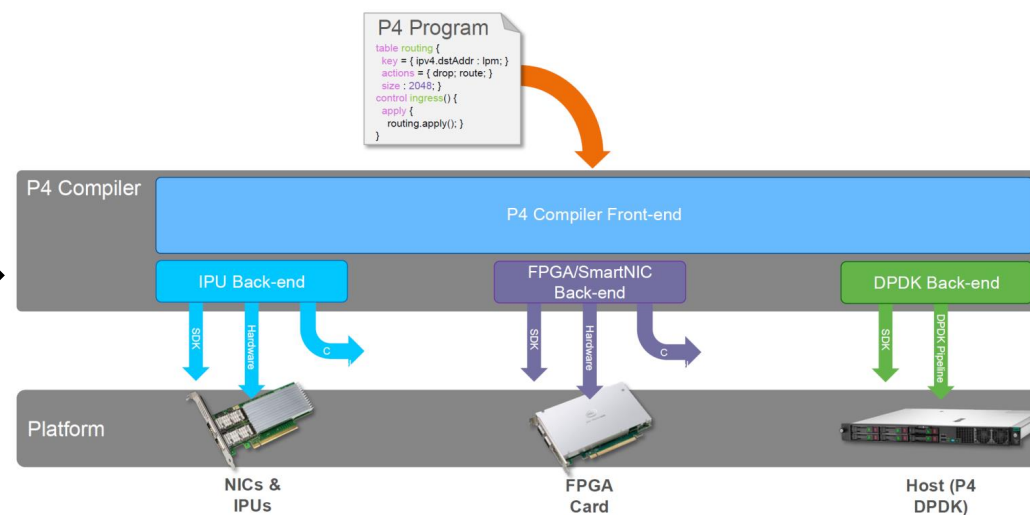
Rob Sherwood, CTO, NEX Cloud Networking Group

## P4 at Intel

Broad enablement across product families



## Intel Vision: Common Programming Model with P4



データプレーンをプログラマブルにする、という目標は達成（Cisco Silicon One, Broadcom, etc.）  
ユーザーニーズに合わせた柔軟なカスタマイズ ⇒ サーバサイド（SmartNIC/Software）が主流になる

# Tofino Architecture オープン化 & コミュニティでの維持



## Tofino を RISC-V のように公開するのは？

- 非常にポジティブ
- 但し、どのように維持発展させるリソースを確保するか？が課題
- RISC-V は数多くの PhD Team による研究開発リソースがあり、SiFiveが金銭的なバックアップ&商業展開を推進した
- Tofino に関して同様のエコシステムを構築・維持する方法は？？
- **維持可能なコミュニティを構築運営するアイデアは大歓迎**



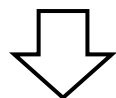
# データプレーン 定義言語 としてのP4

P4の存在理由 (P4's raison d'être)

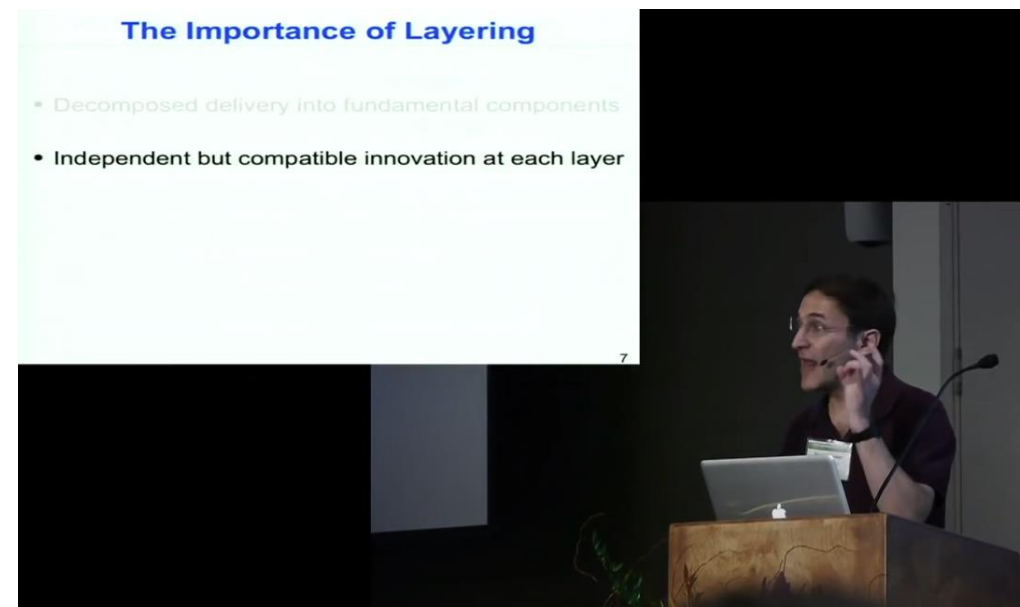
# ネットワークの革新には抽象化が必要

"The Power of Fully-Specified Data Planes", Rob Sherwood, CTO, NEX Cloud Networking Group, Intel

P4はコントロールプレーンとデータプレーンを分離する  
データプレーンの抽象化を実現する



アーキテクチャのコンポーネントを分離する事により  
コントロールプレーンとデータプレーンが  
独立してイノベーションを起こし進化する事が可能になる



YouTube: <https://www.youtube.com/watch?v=YHeyuD89n1Y&t=259s>

Scott Shenker, 2011 “Networking Needs Abstractions”

# データプレーン定義言語としての P4

## 何が可能になるか？

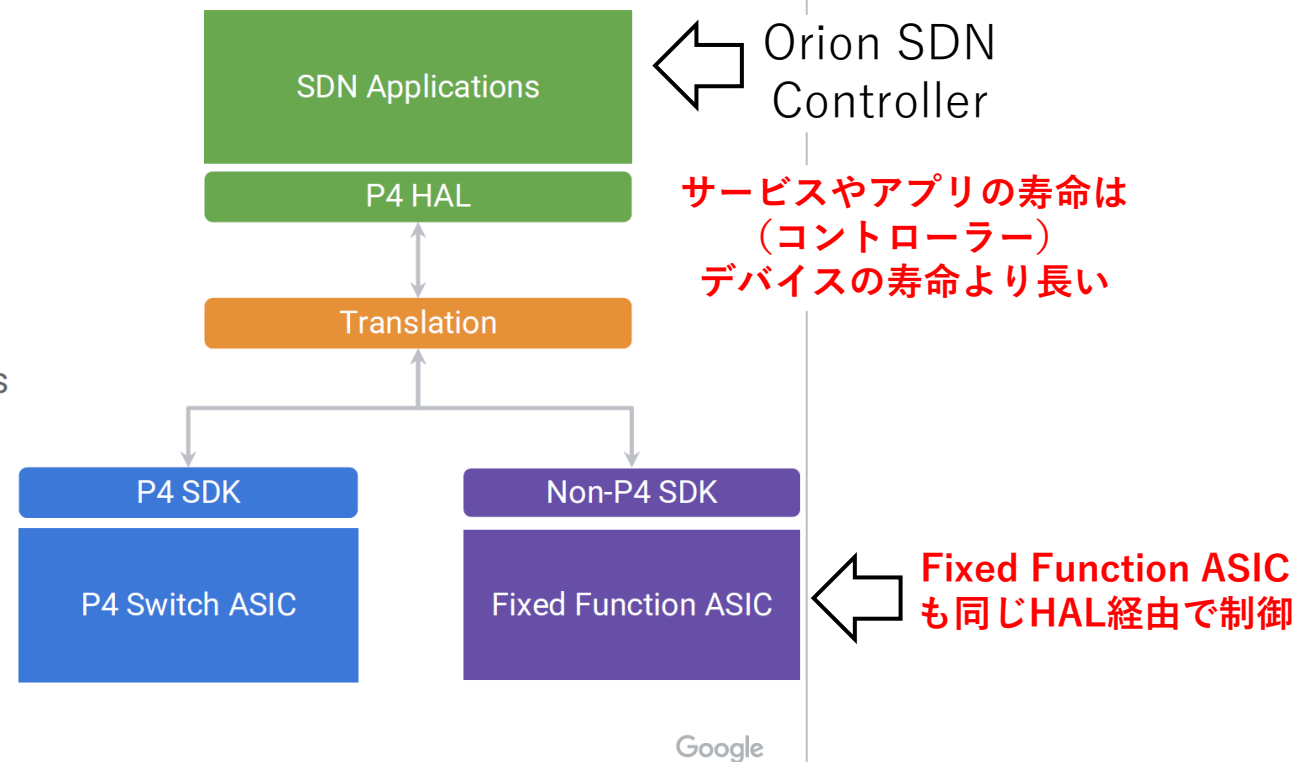
- データプレーン抽象化レイヤ ... スイッチ & SmartNIC
- テストの簡略化
- ハードウェア認証の簡略化
- ソフトウェア進化・オフロードの簡易化

# データプレーン抽象化レイヤ (Switch)

コントローラに対し、データプレーン抽象化レイヤ (API) を提供

## Lessons Learnt: P4 has two related yet distinct use-cases!

- HAL for SDN applications
  - Google SDN Controllers
  - Target-independent P4, avoid vendor extensions
- SDK for programmable hardware
  - Barefoot, Cisco, AMD
  - Target-specific P4, custom optimization extensions



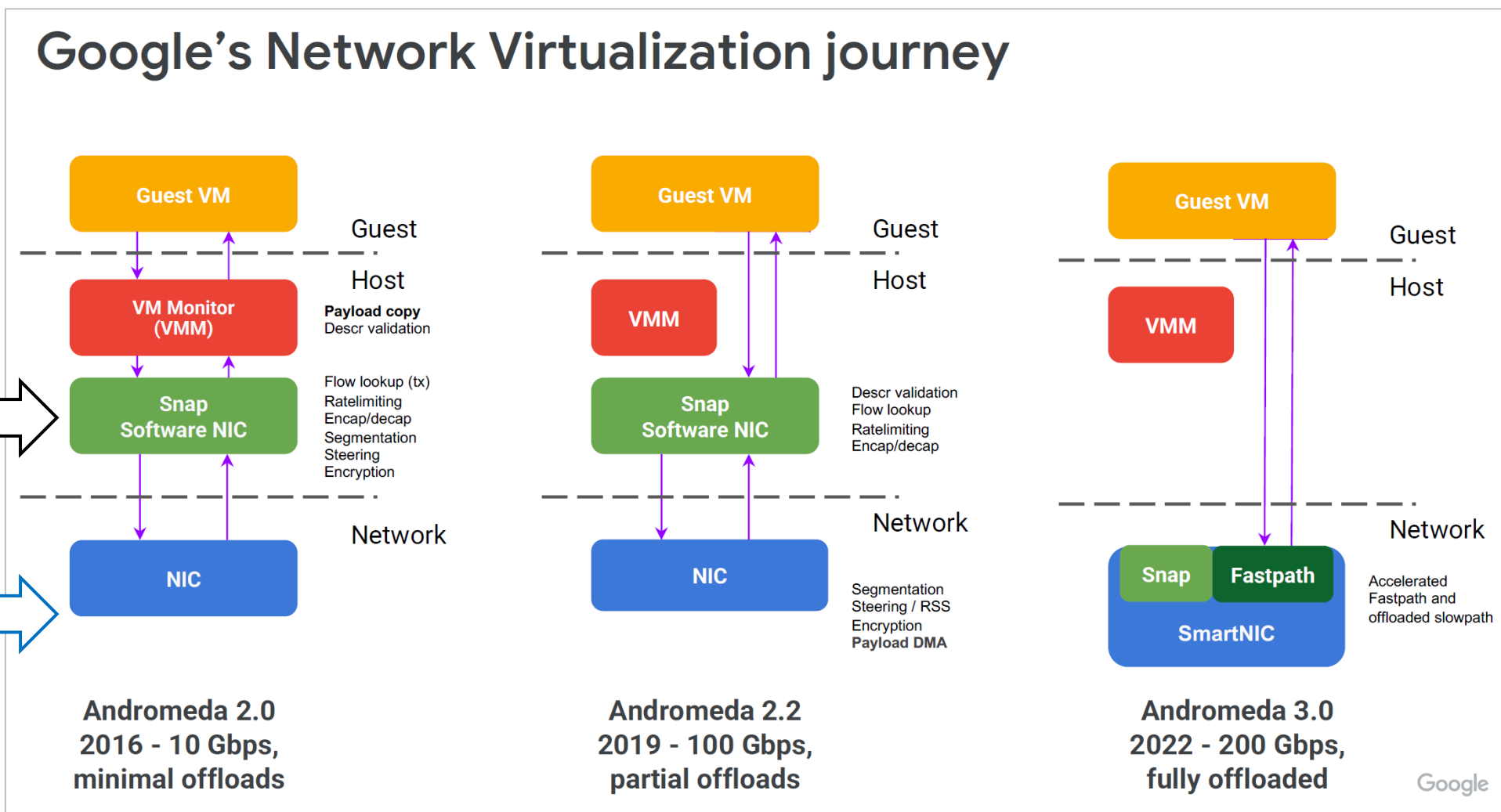


# データプレーン抽象化レイヤ (SmartNIC)

## Google's Network Virtualization journey

ネットワーク  
仮想化レイヤ

ハードウェア  
を交換可能に



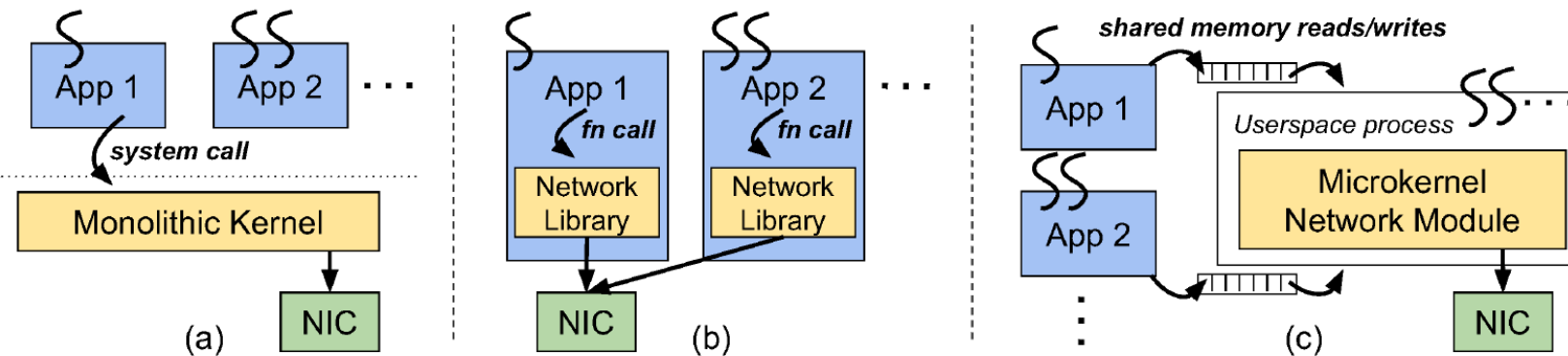
# Snap: a microkernel approach to host networking

<https://dl.acm.org/doi/10.1145/3341301.3359657>

Google @ SOSP'19

SOSP '19, October 27–30, 2019, Huntsville, ON, Canada

Marty and De Kruijf, et al.



**Figure 1.** Three different approaches to organizing networking functionality: (a) shows a traditional monolithic kernel where applications make system calls, (b) shows a library OS approach without centralization and with application-level thread scheduling of processing, and (c) shows the Snap microkernel-like approach leveraging multicore for fast IPC.

Blog: <https://hub.packtpub.com/google-ai-introduces-snap-a-microkernel-approach-to-host-networking/>  
OCP, Networking/NIC Software: [https://www.opencompute.org/wiki/Networking/NIC\\_Software](https://www.opencompute.org/wiki/Networking/NIC_Software)

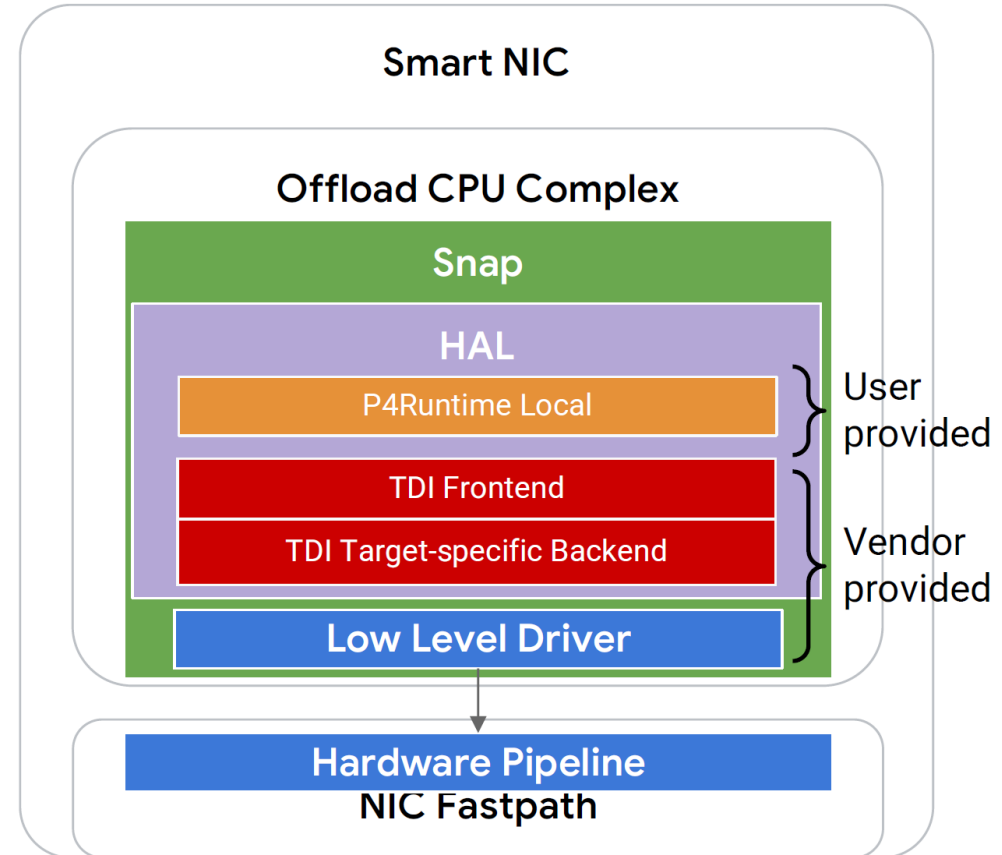
# ハードウェア抽象化レイヤ (HAL) の重要性

- 異なるハードウェア (SmartNIC) にポーティングするのは数年単位の作業
- 複数種類のNICを平行して利用するために "HAL" は必須
  
- P4 とは、最適なネットワークの "HAL"
- 適切なパーツが揃っている
  - APIs + behavioral model + validation tests
- 但し、不足するパーツもある (あった) ⇒ スイッチには無い新しい要件
  - 高速なAPIが必要 (High-performance P4Runtime)
    - Millions table ops/sec for per-connection insert/delete, bulk ops, counter reads
    - 制約の緩和: Local controller only, no need for the overhead due to serialization

# SmartNIC 向けP4関連実装の拡張 (Google)

## A sketch of a P4 HAL for Network Virtualization

- P4 HAL
  - ◆ [P4Runtime Local](#)
  - ◆ [Table Driven Interface \(TDI\)](#)
- Provides high performance
  - ◆ Millions table ops/sec
- Enable a variety of targets
  - ◆ Manual or compiler-driven
- Enable rapid development of features
  - ◆ No intermediary between HAL and target backend



Google



# テストの簡略化

- コントロールプレーンのテスト簡略化
  - データプレーンをソフトウェア（e.g. P4 DPDK）に入替テストを実施
  - CI/CD環境への統合が容易に
  - サービス環境ではハードウェアを利用（e.g. Switch ASIC, SmartNIC）
- データプレーンのテスト自動化
  - テストの自動実行（テーブル設定・変更の自動化）
  - テストパケットの自動生成
  - 結果の検証（生成パケット）

**p4testgen**: Automated Test Generation for Real-World P4 Data Planes  
<https://opennetworking.org/wp-content/uploads/2022/05/Fabian-Ruffy-Final-Slide-Deck.pdf>

# ハードウェア認証の簡略化

ある "ソフトウェア X" は "ハードウェア Y" でサポート（ポーティング）可能か？

## 従来の回答

- 試行錯誤でポーティング、もし解決できない課題があれば NO と結論
- ポーティング可能で、もし広範囲なテストを実施しパスしたら YES と結論

## より良い回答

- "ソフトウェア X" を "P4で抽象化されたデータプレーン" に対し開発する
- 質問をシンプルに変更 ⇒ "X.p4" は "ハードウェア Y" にマップ可能か？
- 完全なソフトウェア移植よりも、p4経由の手動マッピングの方が簡単
- より速く、より簡単で、より正しくなる可能性が高い

IPU E2000 (Mount Evans) と FPGA ベースの IPU で実績あり  
(モバイルユースケース?)

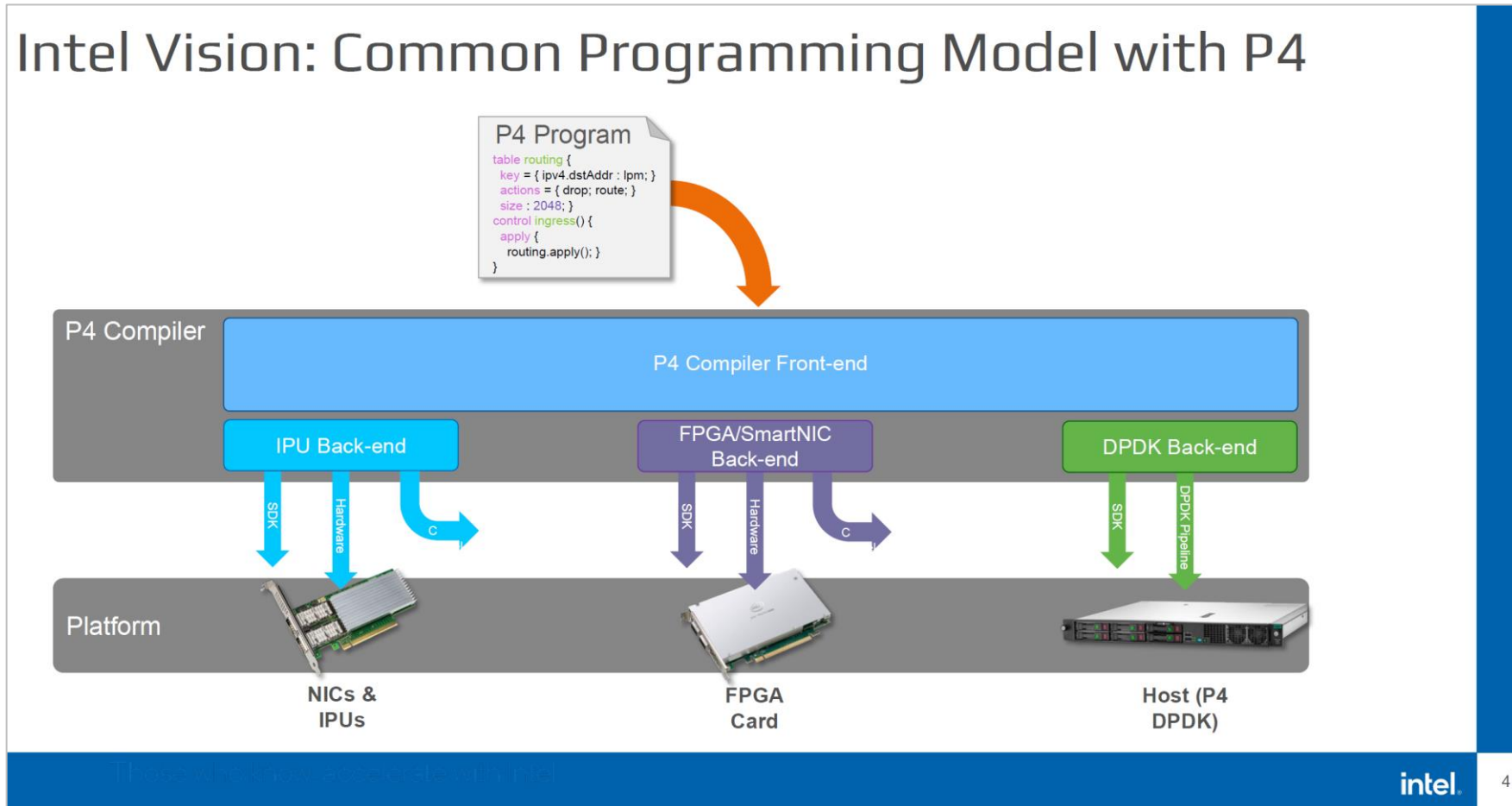
# ソフトウェア進化・オフロードの簡易化

## P4 を用いたソフトウェアデータプレーン開発

- 理解しやすい記述により、データプレーンの進化が容易に
- ハードウェアオフロードが容易に (Clearer semantics for hardware-offload)
- Example #1: Linux Kernel Traffic Classification (TC) System
  - Linux TC の P4 による記述
  - ソフトウェア実行 ⇒ プラットフォームに応じ部分的なオフロードが可能
- Example #2: P4 for Kubernetes
  - Calico P4 Plug-in, k8s.p4 を用いて、ソフトウェア実行
  - Kernel dataplane (iptables, iproute) 処理をIPUにオフロード可能
  - <https://ipdk.io/documentation/Recipes/PaaSOffloadKubernetes/>
  - <https://github.com/ipdk-io/k8s-infra-offload>



# 「プログラミング言語」「定義言語（抽象化レイヤ）」 両方が求められるサーバーサイドでの活用が活性化



2023 P4 Workshop @SanJose "The Power of Fully-Specified Data Planes", Rob Sherwood, CTO, NEX Cloud Networking Group

# サーバーサイド・アクセラレーション

P4に限定しない広い視点で

# サーバーサイド・アクセラレーションが必要となった背景

※ アクセラレーション ⇒ オフロード を含む

NextGenInfra.io PRESENTED BY CONVERGE! NETWORK DIGEST AND AVIDTHINK

## 2022 SmartNICs and Infrastructure Acceleration Report

Explore related Next-Gen sites

2021 Open RAN 2021 Service Assurance 2021 Private Mobile Networks

**Download Report**

### Infrastructure Acceleration Highlights from Industry Thought Leaders

PENSANDO Microsoft napa:tech THE LINUX FOUNDATION OLF NETWORKING JUNIPER NETWORKS

NETRONOME SmartNICs Summit ETHERNITY NETWORKS

Infrastructure Acceleration: SmartNICs, DPUs, FPGAs & more

SmartNICs, DPUs, FPGAs & more

**INFRASTRUCTURE ACCELERATION**

見る YouTube Convergence! AvidThink

Our SmartNICs and Infrastructure Acceleration highlight reel includes key insights from the industry's thought leaders including Soni Jiandani, Co-Founder and Chief Business Officer of Pensando (acquired by AMD); Arpit Joshipura, GM/SVP Networking, Edge and IoT at Linux Foundation Networking; Chuck Sobey, Conference Chair of the SmartNICs Summit; Charlie Ashton, Senior Director of Business Development at Napatech; Michael Bushong, Group VP of the Cloud-Ready Data Center at Juniper Networks; and Kevin Deierling, SVP of Networking at Nvidia.

Convergence! NETWORK DIGEST | AvidThink®

## SmartNICs and Infrastructure Acceleration Report 2022

Enabling the Next Generation of Digital Services

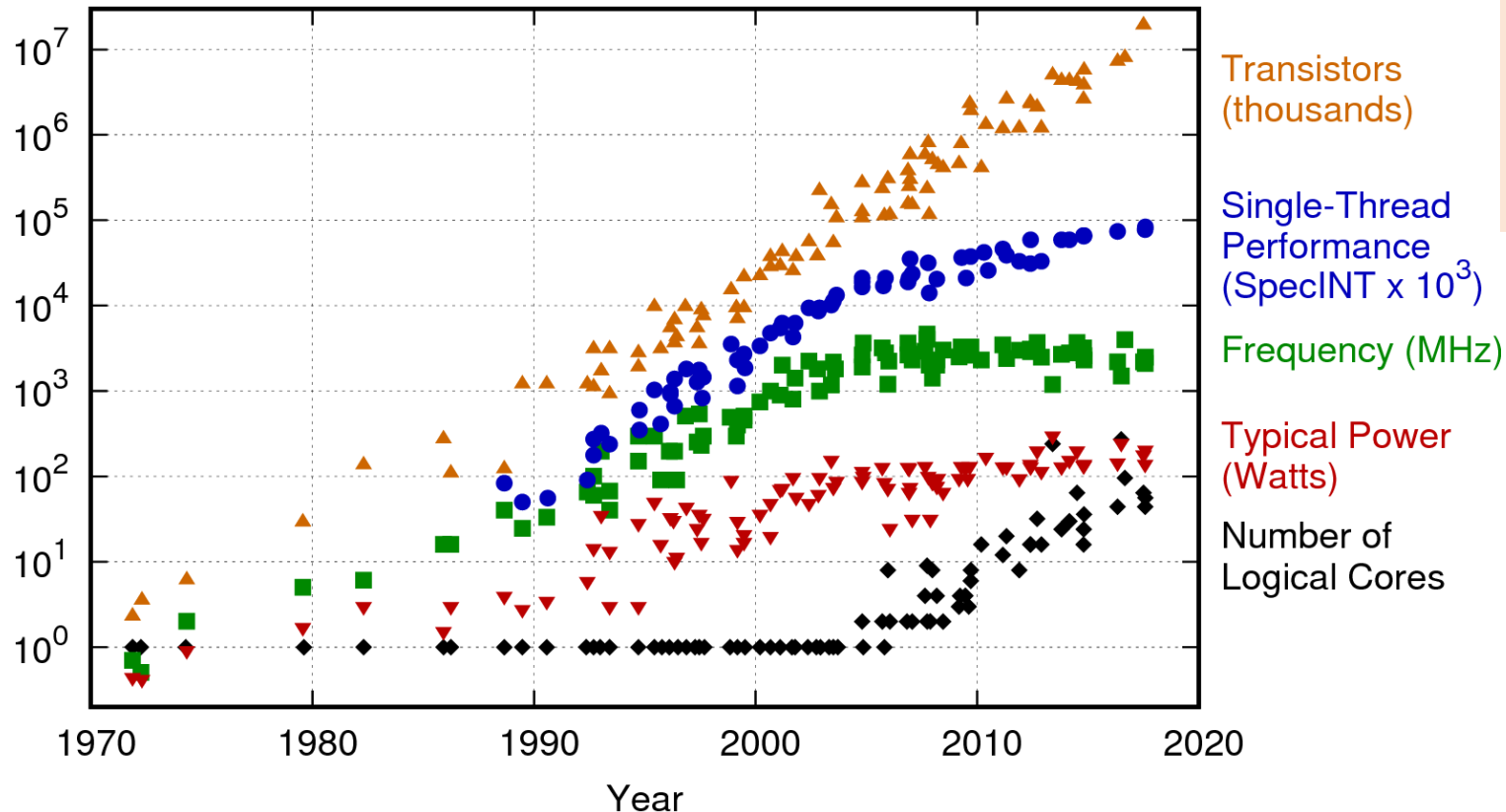
RESEARCH BRIEF

<https://nextgeninfra.io/smartnics-infrastructure-acceleration/>



# (CPU) 半導体トレンドの変化による、アプリケーション構成の変化

42 Years of Microprocessor Trend Data



Moore's Law  
Amdahl's Law  
Dennard Scaling Principles

← 周波数の頭打ち

← コア数の増加

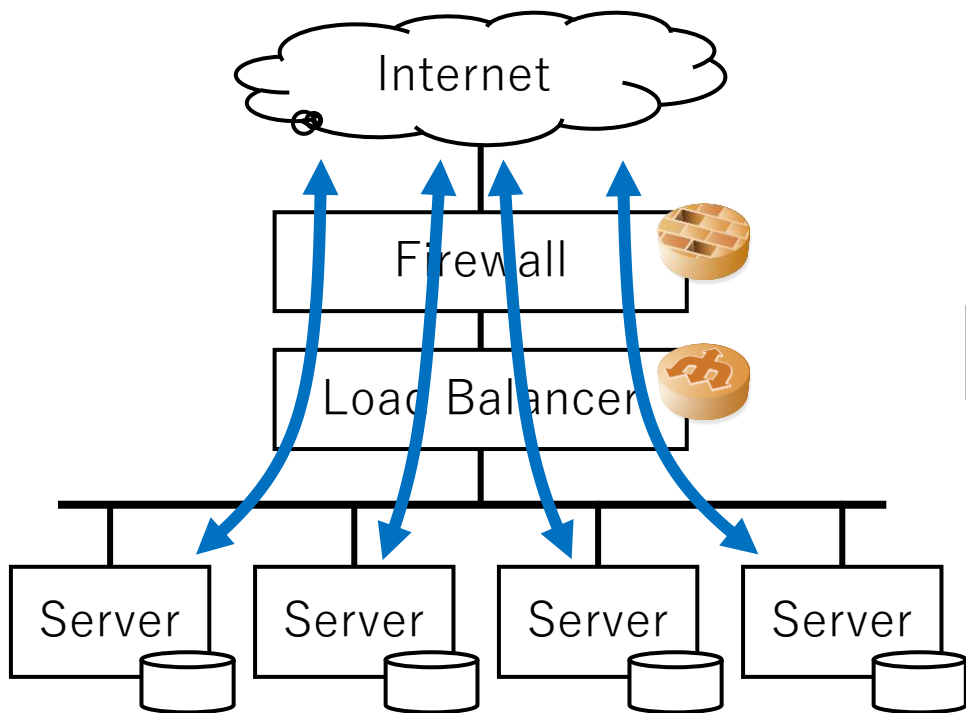
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten  
New plot and data collected for 2010-2017 by K. Rupp

<https://github.com/karlrupp/microprocessor-trend-data>

# アプリケーション構成の変化により、ネットワーク機能の場所も変化

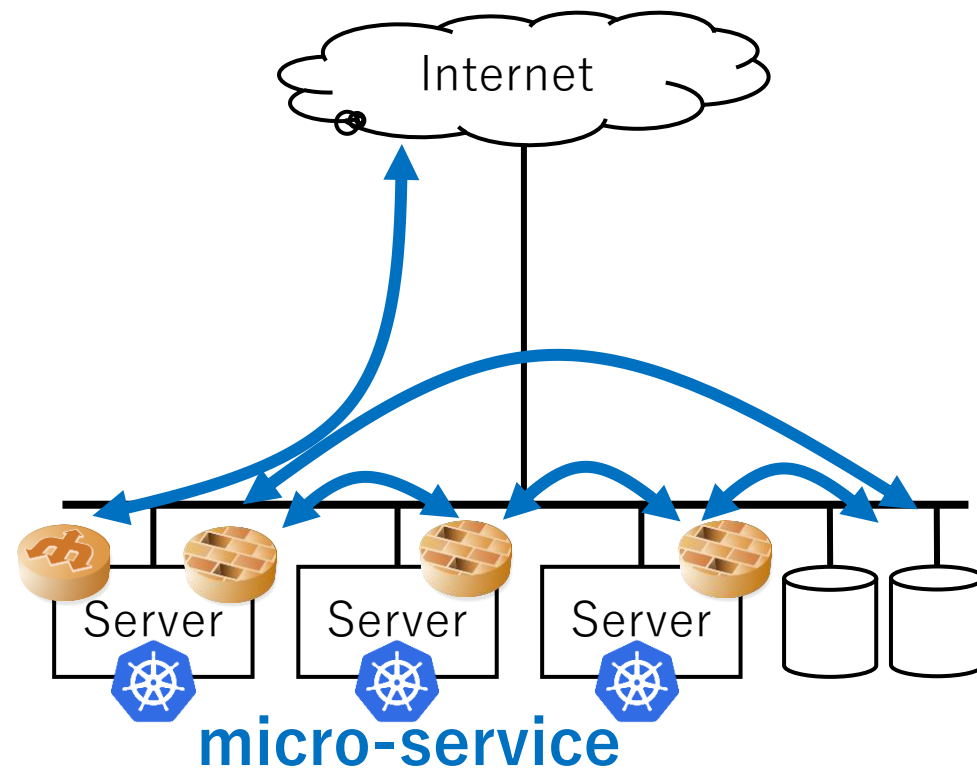
## North-South Traffic 中心

アプリケーションがゲートウェイとして  
ネットワーク機能を提供



## East-West Traffic 中心

サーバーに近い場所での  
ネットワーク機能提供が必要に

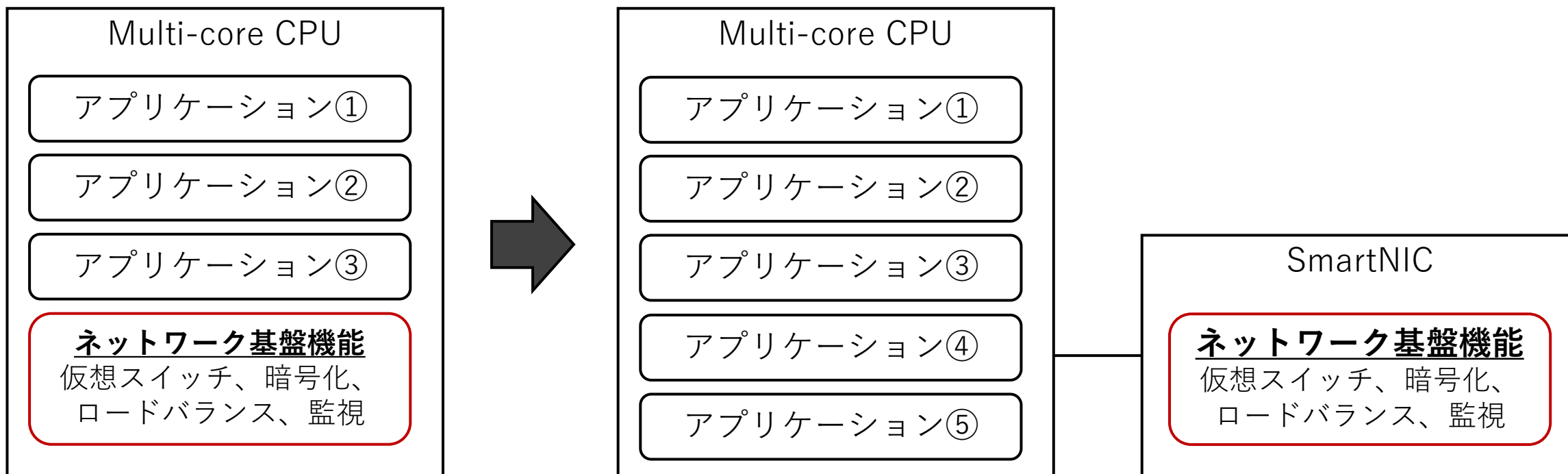


# アプリケーションが占めるCPU利用率の低下

- 2015 Google: Profiling a warehouse-scale computer
  - <https://dl.acm.org/doi/10.1145/2749469.2750392>
  - They found that the diversity of workloads would benefit from flexible architectures, and identify a “datacenter tax” in the lower layers of the software stack that comprises nearly 30% of cycles across jobs and that are prime candidates for hardware specialization and acceleration.
- 2020 Facebook: Accelerometer: Understanding Acceleration Opportunities for Data Center Overheads at Hyperscale
  - <https://dl.acm.org/doi/10.1145/3373376.3378450>
  - Another study by Facebook published in 2020 found that microservices spend as few as 18% of CPU cycles executing core application logic. The remaining cycles were spent in common operations not core to the application logic, including I/O processing, logging, and compression. Facebook believed that accelerating standard building blocks can significantly improve data center performance and they built a model to project possible hardware speedup in microservices.

# サーバーサイドで必要な ネットワーク機能のアクセラレーション

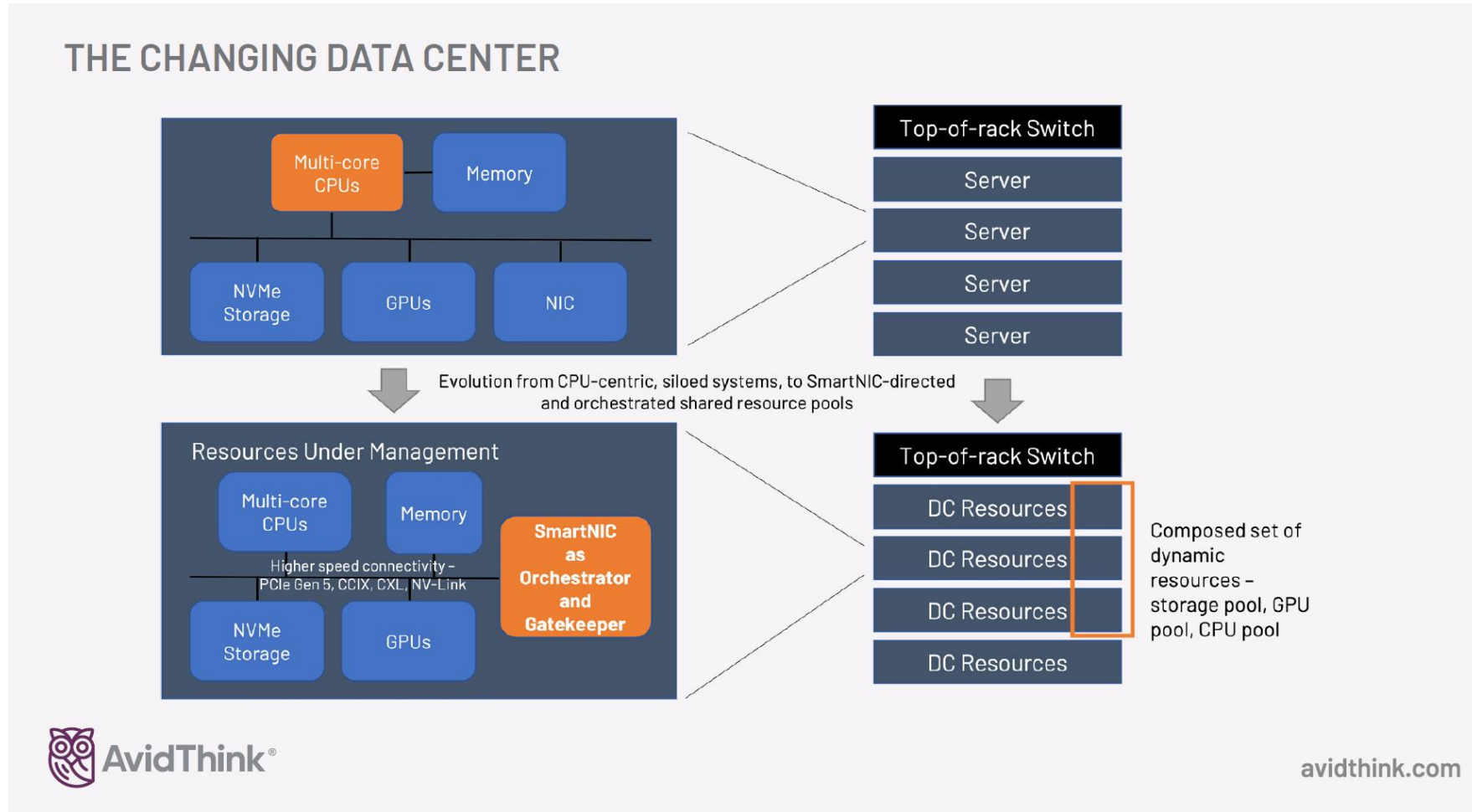
アプリケーションのCPU利用率向上  
& セキュリティの向上 (アイソレーション)



SmartNICによるCPUオフロード



# (将来) 計算リソースの分散配置



# サーバーサイド・アクセラレーション 実現方式の分類

## • ソフトウェア (Software Based)

- 主に Kernel Network Stack で実行される処理の軽減やバイパス
- Linux Kernel と親和性あり
  - **eBPF, XDP, AF\_XDP** (eXpress Data Path, Address Family XDP)
  - **vDPA** (virtio data path acceleration)
    - (コンテナなどアプリケーション基盤や、オブザーバビリティでの利用が多い)
- Linux Kernel と親和性無し
  - **DPDK** (VPP, IPDK, OVS-DPDK)
    - (VPP等、NFVワークロードでの利用が多い)

## • ハードウェア (Hardware Based)

- パケット処理に (CPUより) 適した、ドメイン特化型のハードウェアを利用
- **ASIC, NPU, FPGA, SoC** (Many-coreプロセッサ+暗号化・圧縮・特定演算用チップ)
- NIC型のフォームファクタに搭載され、**IPU/DPU**とも呼ばれる(**SmartNIC**)

注：IPU/DPU/SoC/ASIC/NPU etc. は厳密な定義がなく、ベンダにより異なる場合がある

# "SmartNIC" のタイプ

"Choosing the Best SmartNIC", NVIDIA, Sep 14, 2021 を元に情報を加えて作成  
<https://developer.nvidia.com/blog/choosing-the-best-dpu-based-smartnic/>




	IPU/DPU (ASIC/NPU)	FPGA	Many Core Processor (SoC)
プログラマビリティ	比較的高い (ユースケースに依存)	非常に高い	非常に高い
プログラムの しやすさ	容易 (SDKを利用)	難しい (HDLの経験が必要)	比較的容易 (C-likeな言語が多い)
価格性能比 [*]	非常に良い	高性能だが高価	良い
製品 (例)	Intel IPU E2000 AMD Pensando DPU	Intel IPU F2000X-PL Intel IPU C5000X-PL AMD Alveo SN1000, U25N 他にも多数のベンダが提供	NVIDIA BlueField DPU Marvell OCTEON DPU

[\*] 価格は調達量や方法により大きく変化するため都度確認が必要

FPGA: Intel (Altera), AMD (Xilinx)

# Intel IPU E2000 (MountEvans) & FPGA Cards

<https://www.intel.com/content/www/us/en/products/details/network-io/ipu.html>

Products	Features	Target Acceleration Workloads	Related Documents
<p><a href="#">Intel® IPU E2000</a></p> 	<ul style="list-style-type: none"> <li>• 2 x 100 GbE or 1 x 200 GbE connectivity</li> <li>• Up to 16 Arm Neoverse N1 Cores</li> <li>• PCIe 4.0 x16</li> <li>• Up to 48GB DRAM</li> </ul>	<ul style="list-style-type: none"> <li>• Packet processing</li> <li>• OVS</li> <li>• NVMeOF and Storage</li> <li>• RDMA/RoCEV2</li> <li>• Traffic shaping and QoS</li> <li>• Security: Inline and Lookaside Crypto with Compression</li> </ul>	<p><a href="#">White paper: IPU Based Cloud Infrastructure &gt;</a></p>
<p><a href="#">Intel® IPU Platform F2000X-PL</a></p> 	<ul style="list-style-type: none"> <li>• 2 x 100 GbE connectivity</li> <li>• Intel® Agilex-F FPGA</li> <li>• Intel® Xeon D-1736 Processor</li> <li>• 32GB DRAM</li> </ul>	<ul style="list-style-type: none"> <li>• Packet processing</li> <li>• OVS</li> <li>• NVMe-oF</li> <li>• Security/Isolation</li> <li>• Crypto</li> <li>• RDMA/RoCEV2</li> </ul>	<p><a href="#">Solution brief: Data Center Acceleration with Intel FPGAs &gt;</a></p> <p><a href="#">White paper: IPU Based Cloud Infrastructure &gt;</a></p>
<p><a href="#">Intel® IPU Platform C5000X-PL</a></p> 	<ul style="list-style-type: none"> <li>• 2 x 25 GbE connectivity</li> <li>• Intel® Stratix® 10 DX FPGA</li> <li>• Intel® Xeon D-1612 Processor</li> <li>• 20GB DRAM</li> </ul>	<ul style="list-style-type: none"> <li>• Packet processing</li> <li>• OVS</li> <li>• RDMA/RoCEV2</li> </ul>	<p><a href="#">Solution brief: Data Center Acceleration with Intel FPGAs &gt;</a></p>



# AMD Pensando DPU

Portal : <https://www.amd.com/en/accelerators/pensando>

## AMD Pensando DPUs and Projects

- Shipping 1st generation DPU in 2019, 2nd generation shipped 2021
- Full stack solution, native P4 hardware with a full P4 centric software
- Many years of DPUs in production at Cloud and Enterprise customers
- P4 Applications in Production
  - Multiple SDN Stacks in Clouds
  - Enterprise DPU Distributed Firewall
  - Storage Target Offload
  - Storage NVMeoF Initiator
  - SDN Disaggregation
  - VPN/NAT/Cloud GW
- Deployed in Various Physical Form Factors
  - PCIe card in server
  - Smartswitch
  - SDN Network Accelerator Appliance
  - Storage Target




AMD  
together we advance\_

"Developing Real World Applications", Krishna Doddapaneni, AMD, 2023 P4 Workshop

# AMD Alveo SN1000, U25N SmartNIC

<https://www.xilinx.com/products/boards-and-kits/alveo.html>


Accelerator Cards



**Alveo SN1000**

Industry's first fully software defined, fully hardware accelerated SmartNIC.


[Learn More >](#)



**Alveo U25N**

The Alveo U25N SmartNIC delivers a true convergence of network and security acceleration functions, including OVS and IPsec, into a single platform.


[Learn More >](#)



**Alveo U55C**

Built for HPC and Big Data applications, the Alveo U55C accelerator is the most powerful Alveo card ever from AMD.


[Learn More >](#)



**Alveo U50**

Delivers compute, networking, and storage acceleration in an efficient 75-watt, small form factor, and armed with 100 GbE networking, PCIe Gen4, and HBM2. Designed to deploy in any server.


[Learn More >](#)



**Alveo U30**

The Alveo U30 media accelerator card provides the industry's highest channel density, lowest cost per channel, and lowest power consumption for live video streaming workloads.


[Learn More >](#)





**Alveo U200**

Incredible compute, networking, and storage acceleration thanks to 890k LUTs, 5.9k DSP slices, 64GB of DDR4 memory, and dual 100Gbps network interfaces.

[Learn More >](#)







# NVIDIA DPU BlueField-1/2 & Converged Accelerators

<https://www.nvidia.com/en-us/networking/products/data-processing-unit/>



## NVIDIA BlueField-3 DPU

The NVIDIA BlueField-3 DPU is a 400 Gb/s infrastructure compute platform with line-rate processing of software-defined networking, storage, and cybersecurity. BlueField-3 combines powerful computing, high-speed networking, and extensive programmability to deliver software-defined, hardware-accelerated solutions for the most demanding workloads. From accelerated AI to hybrid cloud, high-performance computing to 5G wireless networks, BlueField-3 redefines the art of the possible.

[Explore BlueField-3 DPUs >](#)

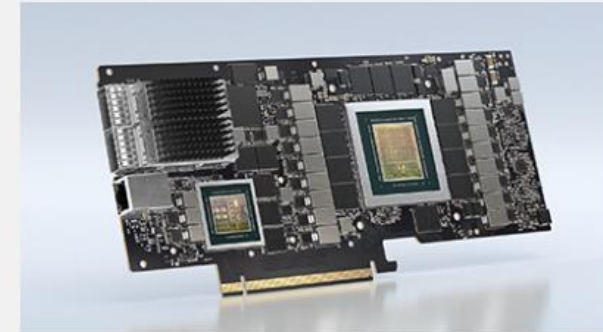


## NVIDIA BlueField-2 DPU

The NVIDIA BlueField-2 DPU provides innovative acceleration, security, and efficiency in every host. BlueField-2 data center infrastructure combines the power of the NVIDIA ConnectX®-6 Dx with programmable Arm® cores and hardware offloads for software-defined storage, networking, security, and management workloads.

NVIDIA BlueField-2 also delivers superior performance, security, and reduced total cost of ownership for cloud computing platforms, enabling organizations to efficiently build and operate virtualized, containerized, and bare-metal infrastructures at massive scale.

[Explore BlueField-2 DPUs >](#)



## NVIDIA Converged Accelerators

NVIDIA converged accelerators combine the power of the NVIDIA® Ampere GPU architecture with the enhanced security and networking capabilities of the NVIDIA BlueField DPU, all in a single high-performance package. This advanced architecture delivers unprecedented performance and strong security for AI-powered workloads in edge computing, telecommunications, and network security.

[Explore Converged Accelerators >](#)

# Marvell OCTEON DPU

<https://www.marvell.com/products/data-processing-units.html>

The screenshot shows the Marvell website's product page for OCTEON 10 DPU. The navigation bar includes the Marvell logo, 'PRODUCTS', 'COMPANY', 'SUPPORT', and icons for globe, user, and search. The main heading is 'EMPOWERING CARRIER, ENTERPRISE AND CLOUD DATA SERVICES Data Processing Units (DPUs)'. The text describes the devices' use in 5G wireless infrastructure and networking equipment. A diagram on the right shows the internal architecture of the OCTEON 10 DPU. At the bottom, there are buttons for various Marvell products.

**MARVELL** PRODUCTS COMPANY SUPPORT

EMPOWERING CARRIER, ENTERPRISE AND CLOUD DATA SERVICES  
**Data Processing Units (DPUs)**

Marvell's OCTEON and ARMADA devices are design for use in 5G wireless infrastructure and networking equipment including switches, routers, secure gateways, firewall, network monitoring, and SmartNICs (Smart Network Interface Cards) and are supported with comprehensive and unified software development kits (SDKs) and open source APIs for a wide range of networking, security and compute market applications.

**OCTEON 10 innovations**

- DDR 5 Memory Controllers
- Arm v9 N2 (64K I / d cache, 1MB L2)
- L3 cache 2MB per core
- Inline Crypto Processor
- System Virtualization
- Inline ML Processor
- Vector Packet Processing
- PCIe 5.0

Up to 400GE Ethernet  
16 x 50G Ethernet Switch

© 2021 Marvell. All rights reserved.

OCTEON 10 Fusion 5G Baseband

No-Compromise 5G Open vRAN Accelerators

OCTEON 10 DPU

OCTEON TX2 DPUs

OCTEON MIPS64 Multi-Core DPUs

OCTEON TX2 LiquidIO III SmartNIC

ARMADA DPUs

Software

<https://www.marvell.com/content/dam/marvell/en/company/media-kit/octeon-10/marvell-octeon-10-media-deck.pdf>



# サーバーサイド・アクセラレーション

ユースケース

# サーバーサイド・アクセラレーションのユースケース

## • CPUのオフロード&セキュリティ向上（主にデータセンター）

- サーバーサイドで必要なネットワーク機能
- クラウド基盤の高速化（オフロード）&セキュリティ向上
- ストレージアクセスの高速化（Target Offload + NVMEoF Initiator）
- 任意のパケットフォーマットでの高速暗号化（IPsec）
- 分散ファイヤーウォール

## • ネットワーク製品（機能）の高速化・効率化（主に通信事業者）

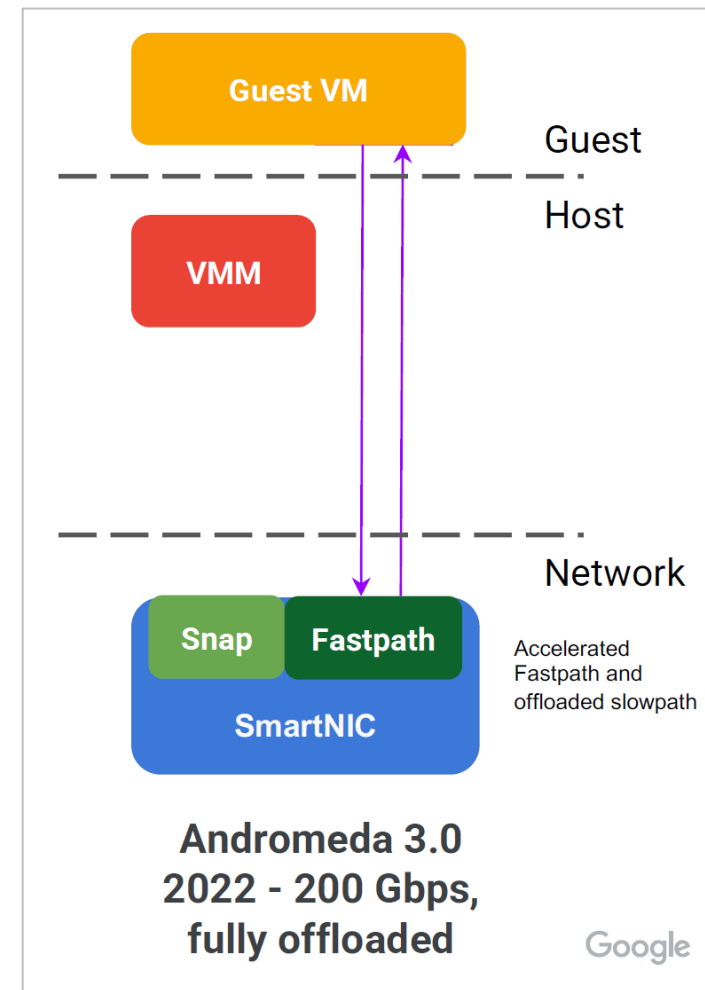
- ネットワーク仮想アプライアンス（NVA）
- VPN/NAT/Cloud GW
- ORAN, UPF（モバイル）
- 高速化&電力効率向上（MEC対応）

### In-Network Computing

- Hyperscale化する機械学習（ML）への対応
- Gradient aggregations offload
- 事例はまだ少ない？（単にサーベイ不足？）

# クラウド基盤の高速化 & セキュリティ向上

- ネットワーク処理のオフロード
  - ホストCPUからネットワーク処理負荷をオフロード
  - より多くのCPUリソースをユーザーに提供可能
  - 暗号化など、CPU負荷の高い処理を専用回路で実行
- セキュリティの向上
  - ユーザーアプリケーションとインフラの物理的な分離
  - DDoS 等の攻撃をCPUの手前で検知・フィルタ



"Keynote 7 - P4 HAL for Network Virtualization"  
Parveen Patel, **Google Cloud**, 2023 P4 Workshop @SanJose

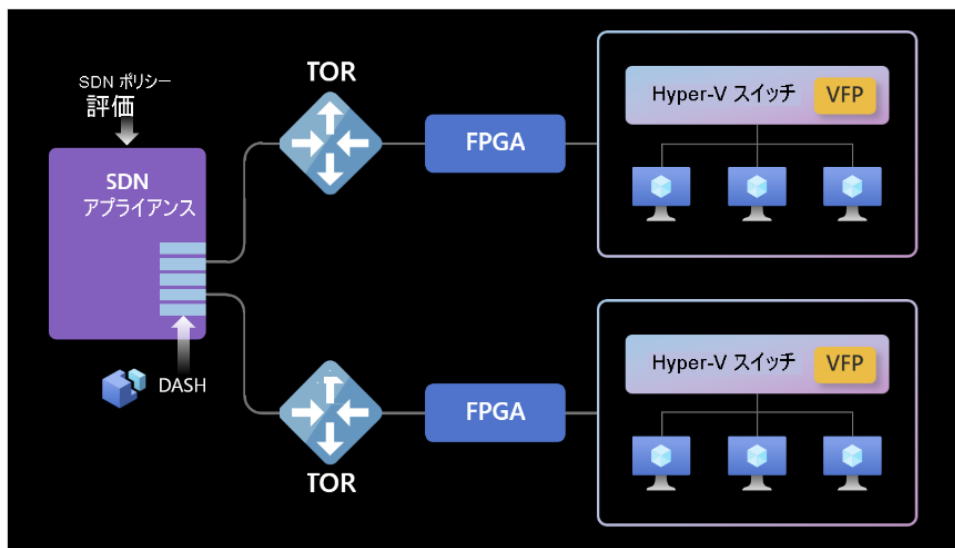
# ネットワーク仮想アプライアンスの高速化

## Microsoft Azure: 高速接続と NVA (プレビュー)

<https://learn.microsoft.com/ja-jp/azure/networking/nva-accelerated-connections>

### メリット

- 1秒あたりの接続数 (CPS) の増加
- 一貫性のあるアクティブな接続
- 高トラフィックネットワークの最適化されたVMのCPU容量や安定性の向上
- ジッターの削減
- CPU 使用率の削減



## NSDI23: Disaggregating Stateful Network Functions

<https://www.usenix.org/conference/nsdi23/presentation/bansal>

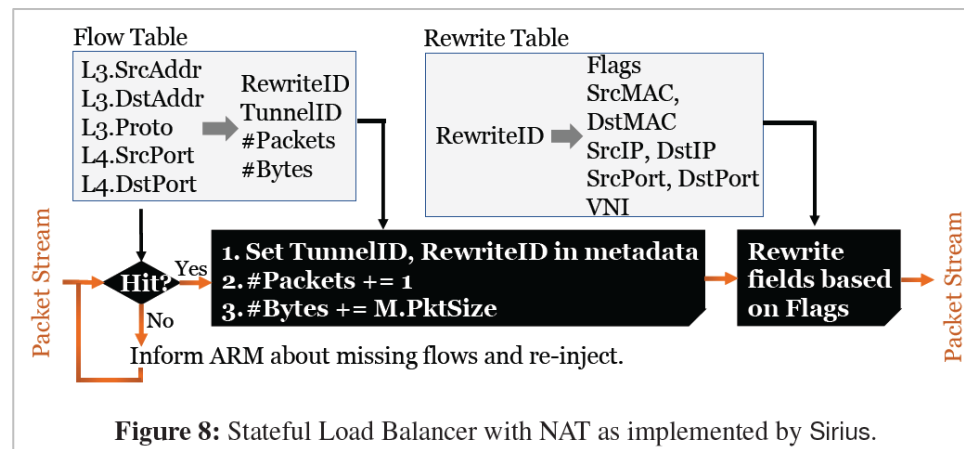


Figure 8: Stateful Load Balancer with NAT as implemented by Sirius.

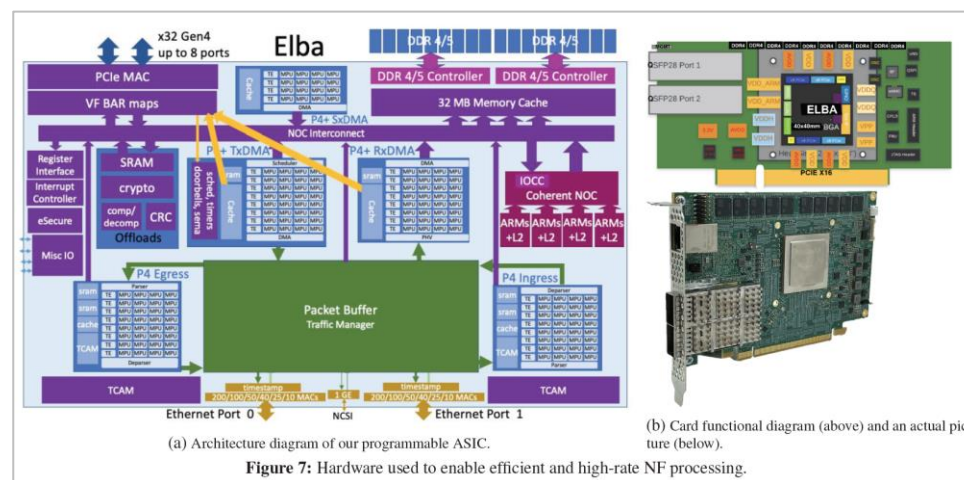
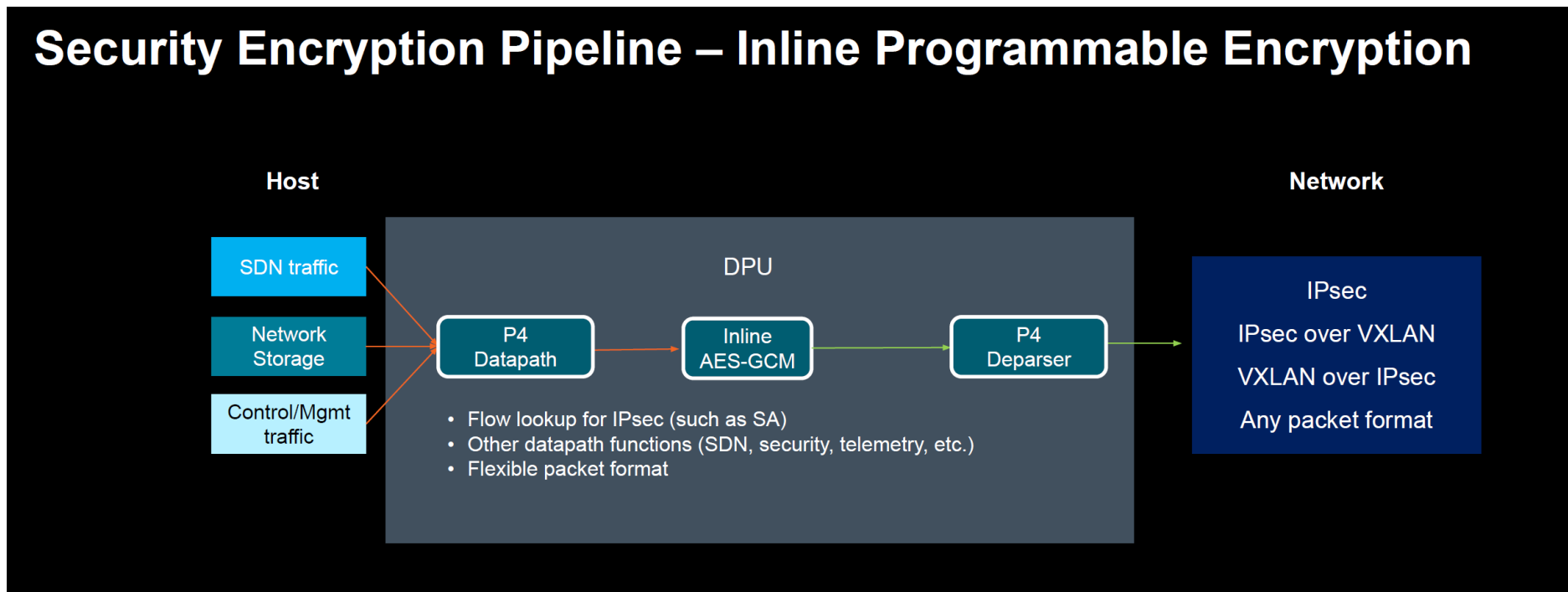


Figure 7: Hardware used to enable efficient and high-rate NF processing.



# 任意のパケットフォーマットでの高速暗号化 (IPsec)



- 低遅延 + 高スループット
- 任意の粒度の暗号化鍵
- 任意のレイヤでの暗号化 (Overlay vs Underlay)
- IPsec コントロールプレーン非依存

## その他のユースケース

- 分散ファイヤーウォール
- ストレージアクセスの高速化 (Target Offload + NVMeoF Initiator)
- VPN/NAT/Cloud GW

### AMD Pensando DPUs and Projects

- Shipping 1st generation DPU in 2019, 2nd generation shipped 2021
- Full stack solution, native P4 hardware with a full P4 centric software
- Many years of DPUs in production at Cloud and Enterprise customers
- P4 Applications in Production
  - Multiple SDN Stacks in Clouds
  - Enterprise DPU Distributed Firewall
  - Storage Target Offload
  - Storage NVMeoF Initiator
  - SDN Disaggregation
  - VPN/NAT/Cloud GW
- Deployed in Various Physical Form Factors
  - PCIe card in server
  - Smartswitch
  - SDN Network Accelerator Appliance
  - Storage Target



AMD  
together we advance

2

# プログラミング言語としてのP4

サーバーサイド・アクセラレーションに向けたP4の拡張

# SmartNIC で必要な P4 機能（スイッチと比較）

## Missing features in 2020

- Add-on-miss
  - Add new entries to tables at high rate *in data plane*
- Auto-delete
  - Delete old entries *in the data plane* when they have been unmatched for configurable duration.
  - The timeout duration of an entry is *modifiable* at packet processing time.
- Packet encryption
  - Data plane APIs and P4 architecture flow for encryption & decryption
  
- Now part of the PNA specification

## Missing features in 2023

- Data-plane-writable action data
  - e.g. maintain expected TCP sequence numbers independently for each table entry, in TCP connection tracking.
- High throughput control plane APIs
  - Adding millions of table entries per second to large tables.
- Configuring externs with P4Runtime API more consistently
  - See new GenericTable idea proposed in P4 API work group
  
- Updating P4 code with 0 down time
  - Implementation techniques are typically target-dependent, but sharing ideas on how is likely to make this more widely available.
- Support in Linux to load P4 code into kernel
  - Then offload into NICs that support it. See talk on P4-TC later in the workshop.
- Good IDE support
  - New open source repo: <https://github.com/p4lang/p4analyzer>
  
- In active discussion/development now

# SmartNIC で必要な P4 機能（スイッチと比較）

---

- 暗号化・複合化
- 高速なAPI（P4Runtime）
  - Adding millions of table entries per second to large tables.
- コントロールプレーン非依存なテーブル更新
  - Table lookup miss 時のエントリ追加
  - Timeout したエントリの削除
- ステートフルなプロトコルへの対応
  - Data-plane-writable action data
  - e.g. maintain expected TCP sequence numbers independently for each table entry, in TCP connection tracking.



# P4言語仕様の拡張 (PNA)

## P4 PNA (Portable NIC Architecture) 仕様で add\_on\_miss, auto\_delete (idle timeout) をテーブル属性として定義

### 8.1. Tables with add-on-miss capability

PNA defines the `add_on_miss` table property. If the value of this property is `true` for a table `t`, the P4 developer is allowed to define a default action for `t` that calls the `add_entry` extern function.

When `t.apply()` is invoked, `t`'s lookup key is constructed, and the entries of the table are searched. If there is no match, i.e. the lookup results in a miss, `t`'s default action is executed. So far, this is all standard behavior as defined in the P4<sub>16</sub> language specification.

If `t`'s default action makes a call to `add_entry`, it causes a new entry to be added to the table with the same key that was just looked up and resulted in a miss, and the action name and action parameters specified by the parameters of the call to the `add_entry` extern function. Thus, future packets that invoke `t.apply()` with the same lookup key will get a match and invoke the specified action (until and unless this new table entry is removed). The new table entry will be matchable when the next packet is processed that invoked `t.apply()`.

Some PNA implementations may allow the control plane software to add, modify, and delete entries of such a table, but any entries added via the `add_entry` function do not require the control plane software to be involved in any way. Other PNA implementations may choose not to support control plane modification of the entries of an add-on-miss table.

It is expected that PNA implementations will be able to sustain `add_entry` calls at a large fraction of their line rate, but it need not be at the same packet rate supported for processing packets that do not call `add_entry`.

### 8.2. Table entry idle timeout

PNA defines the table property `pna_idle_timeout` to enable specifying whether a table should maintain an idle time for each of its entries, and if so, what the data plane should do when a table entry has not been matched for a length of time at least its configured idle time.

The value assigned to `pna_idle_timeout` must be a value of type `PNA_IdleTimeout_t`:

```
/// Supported values for the pna_idle_timeout table property
enum PNA_IdleTimeout_t {
    NO_TIMEOUT,
    NOTIFY_CONTROL,
    AUTO_DELETE
};
```

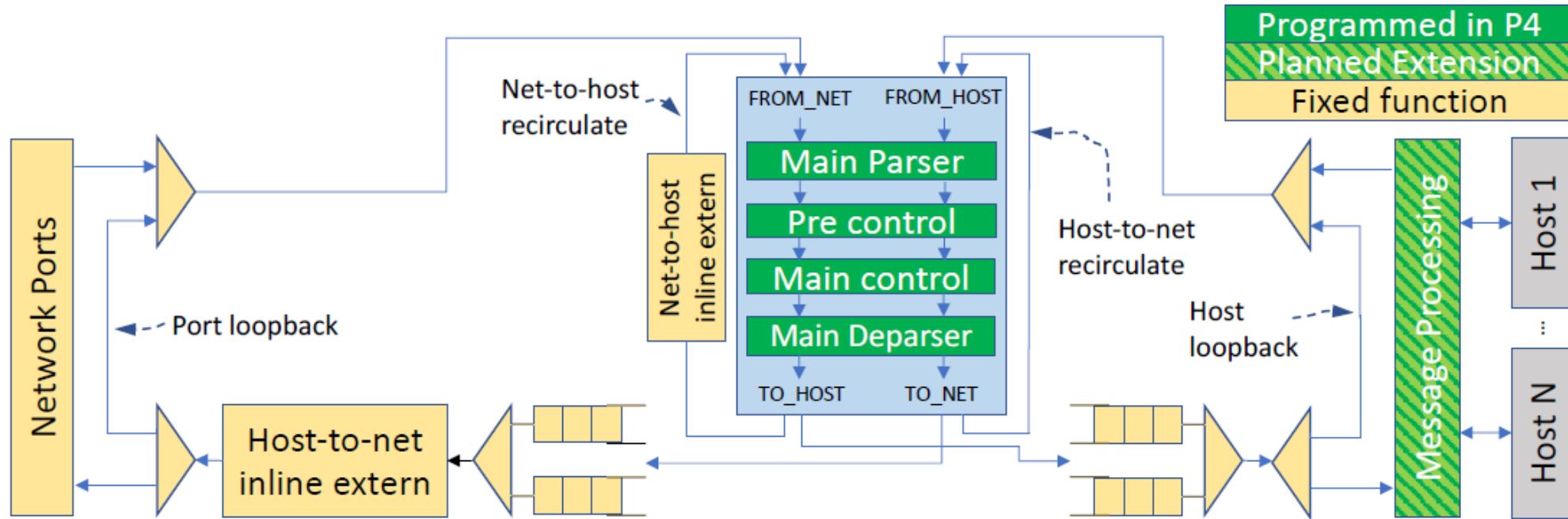
If the property `pna_idle_timeout` is not specified for a table, its default value is `NO_TIMEOUT`. Such tables need not maintain an idle time for any of its table entries, and will not perform any special action regardless of how long a table entry remains unmatched.

<https://p4.org/specs/>

#### P4<sub>16</sub> Portable NIC Architecture (PNA)

- v0.7 [HTML | PDF] (Dec 2022)
- Working draft: [HTML | PDF]

# P4-16 Portable NIC Architecture (PNA)



## PNA Architecture

Figure 2. Packet Paths in PNA

引用：P4 Portable NIC Architecture (PNA), version 0.5

# P4-16 Portable Switch Architecture (PSA)

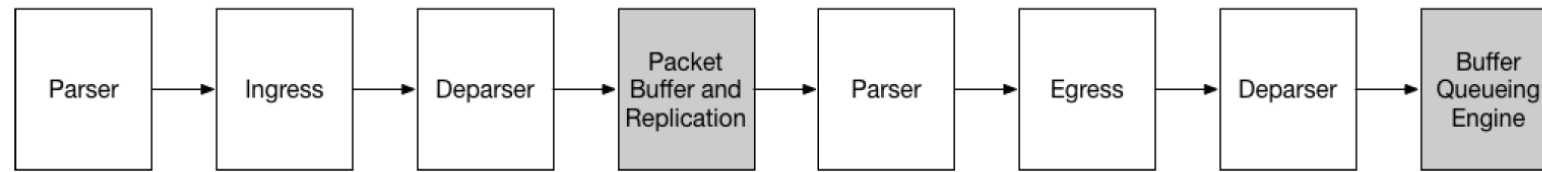


Figure 1. Portable Switch Pipeline

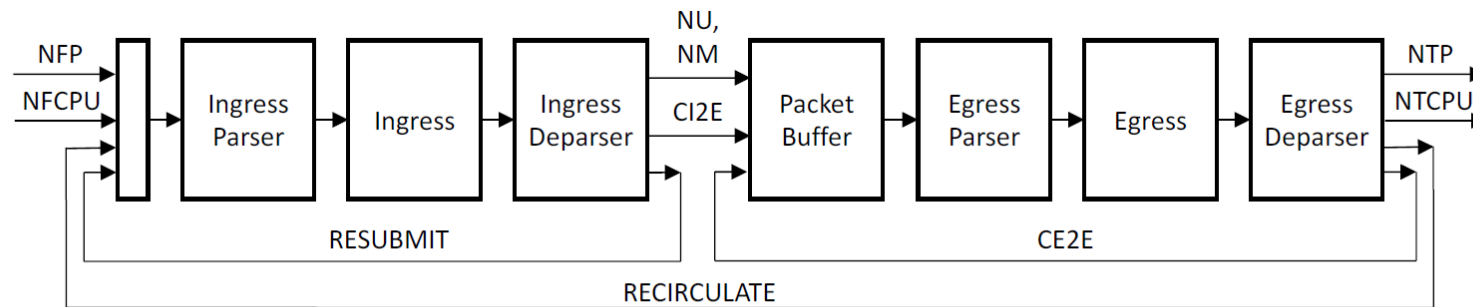


Figure 2. Packet Paths in PSA

引用：P4-16 Portable Switch Architecture (PSA), version 1.2

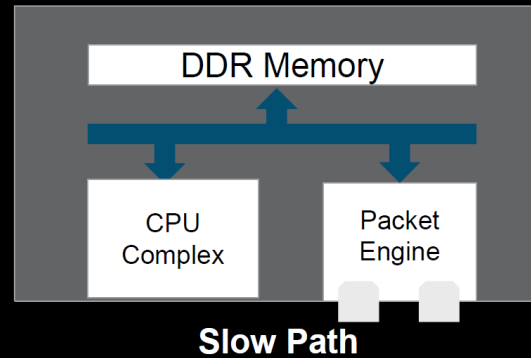
# Policy や Flow timeout Check の "DPU CPU" からのオフロード

## Pensando: P4による CPS の向上 Connection Per Sec

### Accelerate CPS with P4

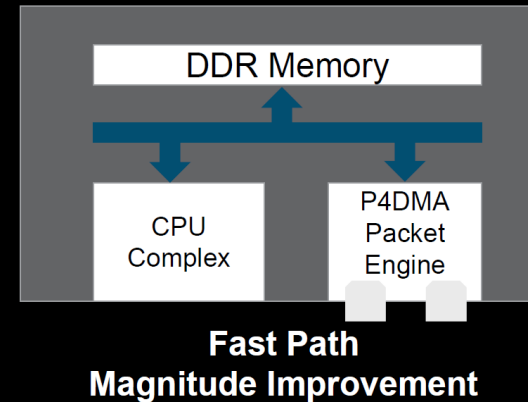
#### DPU CPU connection setup/tracking

1. New flow packet received
2. DPU CPU checks policy
3. DPU CPU performs table lookup
  - Next hop, metering, telemetry, NAT, LB etc.
4. DPU CPU create flow entry
5. Packet Forwarded
6. Flow timeout checked periodically by Arm® CPU



#### DPU P4 connection setup/tracking

1. New flow packet received
2. P4 performs table lookup
  - Next hop, metering, telemetry, NAT, LB etc..
3. DPU CPU creates flow entry
4. Packet Forwarded
5. Flow timeout checked periodically by P4



# SmartNIC での P4Runtime API の高速化

2022 P4 Workshop, NVIDIA, Evolving P4Runtime from Switch to DPU

<https://opennetworking.org/wp-content/uploads/2022/05/Alan-Lo-and-Milind-Chabbi-Final-Slide-Deck-1.pdf>

## Switch と SmartNIC の制約条件の違いを利用

コントローラがローカルに存在

⇒ APIがネットワークを経由せず、共有メモリを利用可能

## Bulk table update を追加

コントローラがリモートに存在しても利用可能

2022年4月頃に P4 API Working Group で議論  
その後情報無し (TDI, IPDK などのフレームワークに合流か? 要調査)

<https://groups.google.com/a/lists.p4.org/g/p4-api>

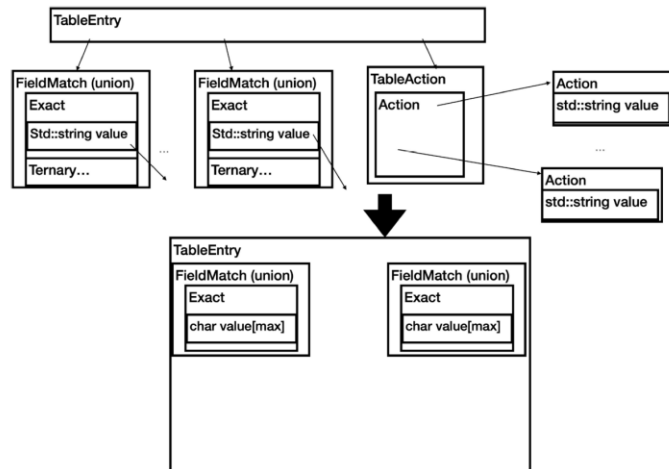


# 2022 P4 Workshop, NVIDIA, Evolving P4Runtime from Switch to DPU

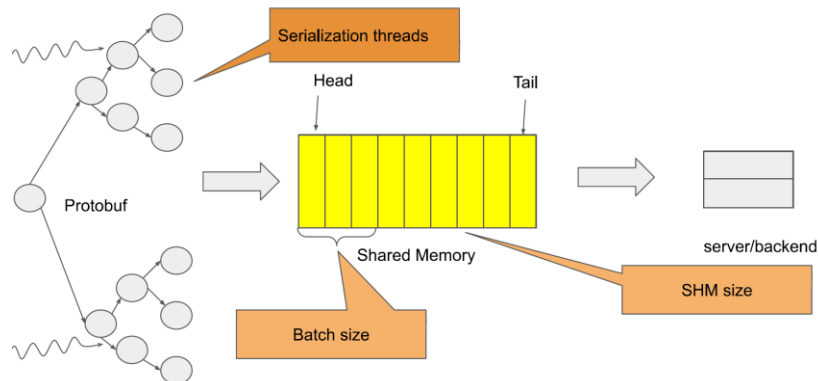
<https://opennetworking.org/wp-content/uploads/2022/05/Alan-Lo-and-Milind-Chabbi-Final-Slide-Deck-1.pdf>

## Shared-memory for local controller

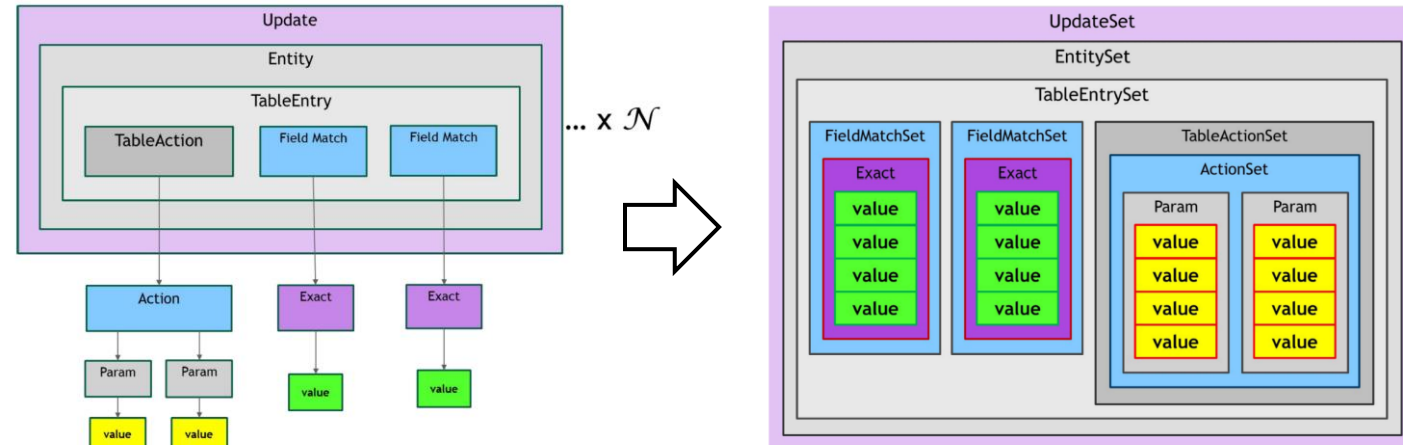
### Serialize Protobuf to C-style Struct



### Tunables



## Bulk table updates for remote controller



Layer	Operations / sec	Comments
gRPC	945,000	C, deserialize with arena
gRPC bulk	9,289,000	C, deserialize with arena
libPI	598,000	no gRPC, already deserialized
bulk PI	5,996,000	no gRPC, already deserialized

\* Broadwell - Intel(R) Xeon(R) CPU E5-2609 v4 @ 1.70GHz

# Server Side P4 の技術とコミュニティ

# Server Side P4 の技術とコミュニティ

P4TC

P4-DPDK

IPDK

Infrastructure Programmer Development Kit

OPI

Open Programmable Infrastructure

# P4TC

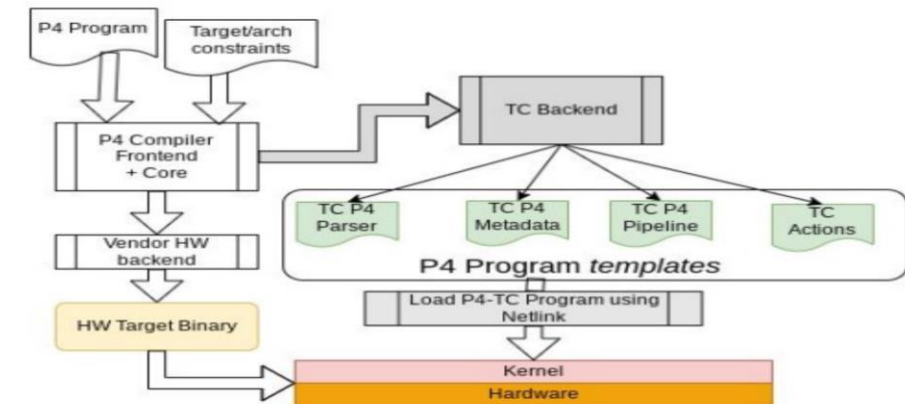
- P4 を Linux TC にコンパイル
- Netlink 経由で Kernel にロード
- オフロード無し有りの様々な方法で実行可能
  - Model 1: Scriptable P4TC (SW dpath via P4TC)
  - Model 2: eBPF Parser Only, rest of SW Dpath via P4TC
  - Model 3: SW dpath eBPF at TC+XDP independent of P4TC
  - Model 4: Integrated ebpf sw-dataplane P4TC control
- 状況
  - Linux Kernel へ Upstream 中
  - 2023 P4 Workshop の次の日に P4TC Workshop を実施
    - 10名程度 (Intel, NVIDIA, Mojatatu Networks, + ebiken 😊)
    - Intel E2000 (MEV) を用いたデモを紹介
  - デファクト化、コミュニティの拡大、等は始まったばかり

## References

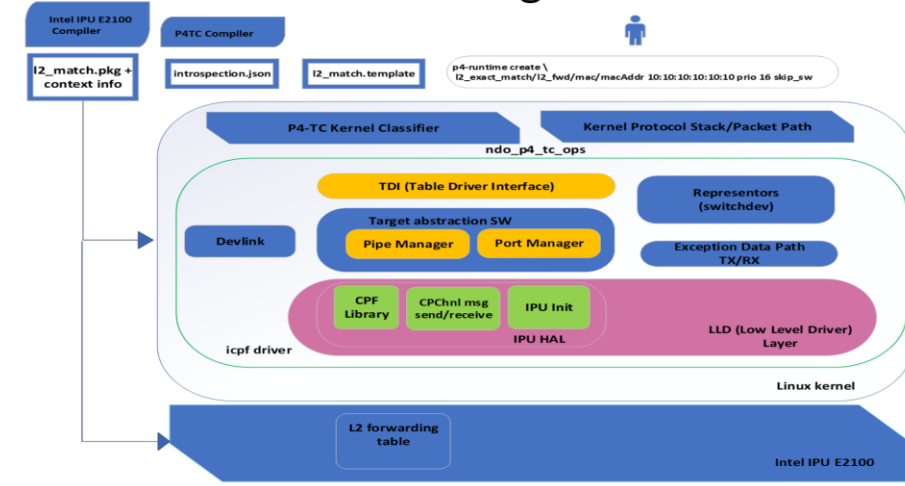
- What Is P4TC You Ask? (P4TCとは何か?の詳細な解説)
  - <https://github.com/p4tc-dev/docs/blob/main/why-p4tc.md>
- P4TCポータル: NetDevConf 0x16 等、過去のカンファレンス資料有り
  - <https://www.p4tc.dev/>
- 2023 P4 Workshop @SanJose
  - In-depth Talk - Hardware Offload Driver with P4-TC, Anjali Singhai Jain, Namrata Limaye
  - In-depth Talk - P4TC: Linux Kernel P4 Implementation Approaches And Evaluation, Deb Chatterjee, Jamal Hadi Salim

"Hardware Offload Driver with P4-TC", Anjali Singhai Jain, Namrata Limaye, 2023 P4 Workshop

## P4-TC in Kernel

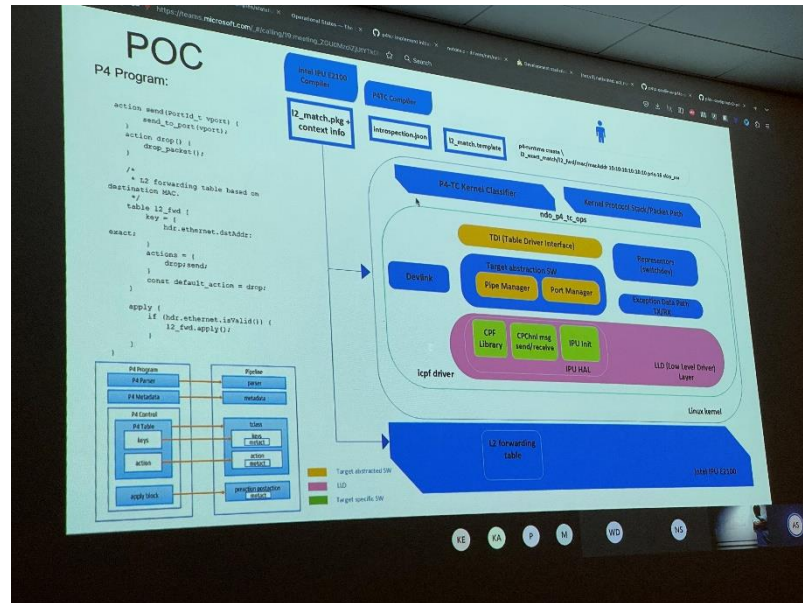
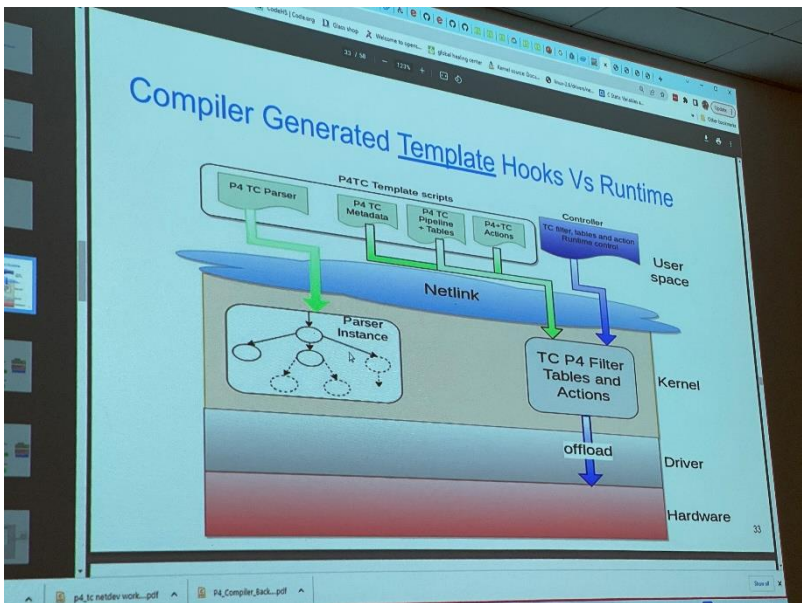


## P4-TC Offload Driver Design





# 2023 P4TC Workshop (Next day of 2023 P4 Workshop @SanJose)



```

17 $TC p4template create hdrfield/redirect_srcip/redirect_srcip_parser ethernet destAddr param:srcip \
18 @ Defining Metadata \
19 @ Defining Metadata \
20 @ Defining Metadata \
21 @ Defining Metadata \
22 echo "Creating Metadata" \
23 sleep 1 \
24 $TC p4template create metadata/redirect_srcip/global/drop mid 1 type bist \
25 @ Metadata for global port \
26 $TC p4template create metadata/redirect_srcip/output_port mid 1 type dev \
27 $TC p4template get metadata/redirect_srcip \
28 @ Dumping Metadata \
29 $TC p4template update action/redirect_srcip/mainControlI/drop mid 1 type dev \
30 @ Create Actions = defined externs \
31 @ Create Actions = defined externs \
32 @ Note, here we create the actions specified in the P4 program \
33 @ Note, here we create the actions specified in the P4 program \
34 @ Note, here we create the actions specified in the P4 program \
35 sleep 1 \
36 @ send_mh() just sets metadata which will be used on P4TC hooks \
37 @ We have a type "dev" which is not really a type \
38 @ Other possible types here are port, ifindex and port id \
39 @ Note, here we create the actions specified in the P4 program \
40 @ Note, here we create the actions specified in the P4 program \
41 $TC p4template create action/redirect_srcip/mainControlI/send_mh actid 1 \
42 param port type dev id 1 \
43 param smac type macaddr id 1 \
44 param dmac type macaddr id 1 \
45 cmd set hdrfield. redirect_srcip. redirect_srcip_parser ethernet destAddr param:srcip \
46 cmd set hdrfield. redirect_srcip. redirect_srcip_parser ethernet srcAddr param:srcip \
47 cmd set metadata. redirect_srcip.global.drop constant.bist 1 \
48 \
49 @ activate \
50 $TC p4template update action/redirect_srcip/mainControlI/send_mh state active \
51 \
52 $TC p4template create action/redirect_srcip/mainControlI/drop actid 1 \
53 cmd set metadata. redirect_srcip.global.drop constant.bist 1 \
54 \
55 @ activate \
56 $TC p4template update action/redirect_srcip/mainControlI/drop state active \
57 \
58 @ Tables \
59 @ Tables \

```



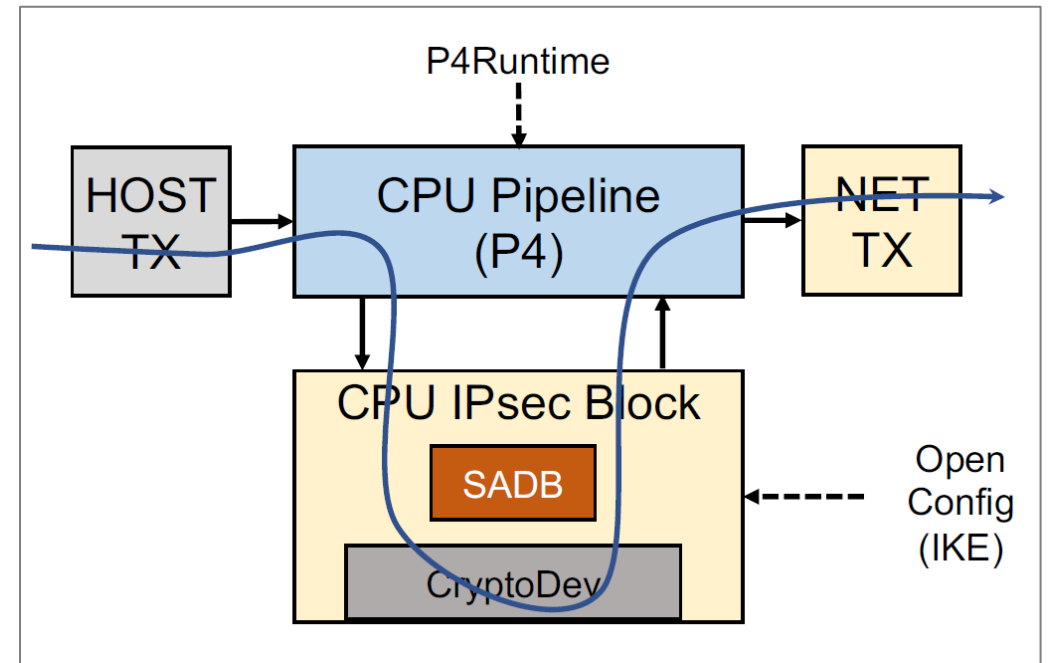
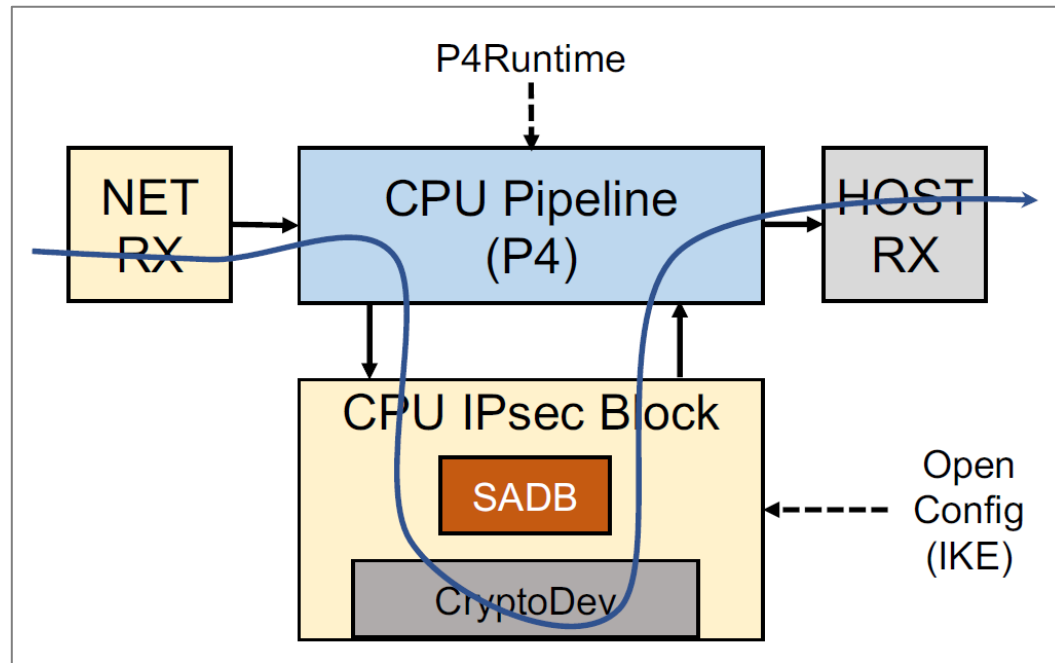


# P4-DPDK

- P4 program を multi-core CPU で実行するフレームワーク
- 目的：DPDKの性能に、P4の柔軟性（プログラムしやすさ）を組み合わせ、より良いソフトウェアスイッチを実現する
- IPDK の CPU Target
  
- オープンソースで公開
- P4 compiler back-end and TDI driver on p4.org
  - <https://github.com/p4lang/p4c/tree/main/backends/dpdk>
- P4 data plane engine on dpdk.org
  - <http://git.dpdk.org/dpdk/tree/lib/pipeline>

# P4-DPDK で高速な IPsec 処理を実現

- DPDK IPsec & crypto library を利用
- P4 extern により複雑さを開発者から隠蔽





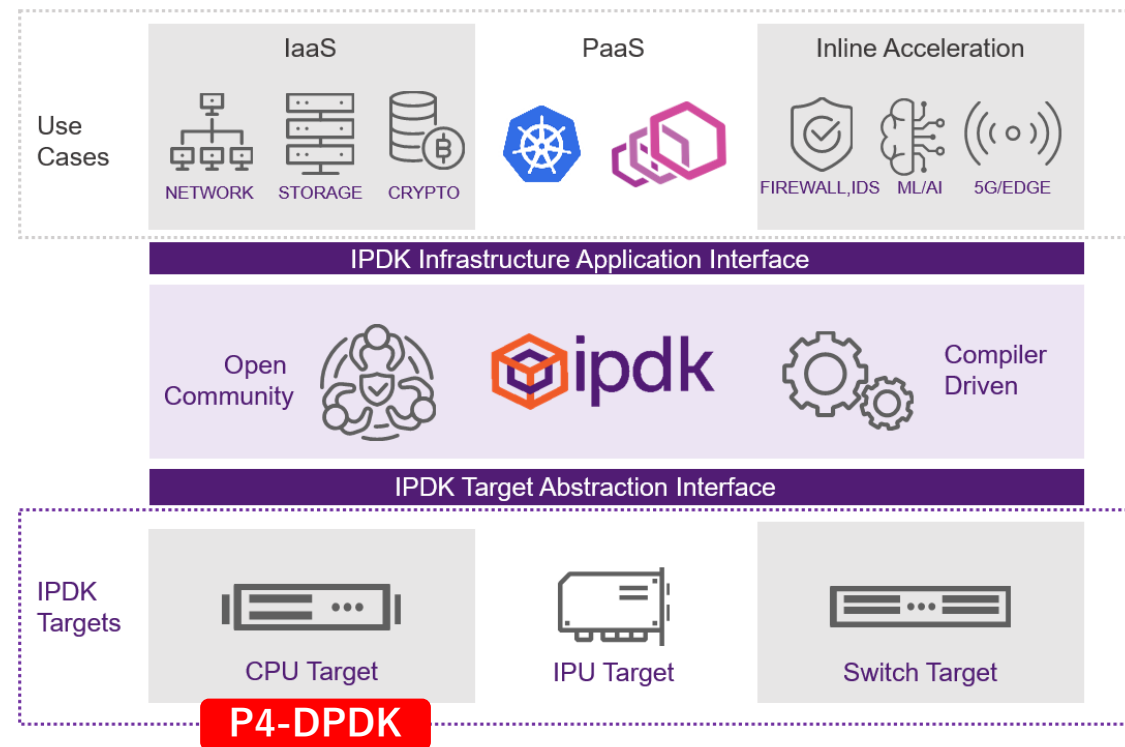
## Infrastructure Programmer Development Kit

<https://ipdk.io/>

- Infrastructure Offload のためのフレームワーク
- SmartNIC(IPU/DPU)だけでなく、CPU, Switch を含む様々なデバイスに対し、共通のコントロール方法を提供
- 中核となるのは TDI (Table Driven Interface)
  - <https://github.com/p4lang/tdi>
- 現在は OPI の Sub Project として活動

### IPDK お試し方法

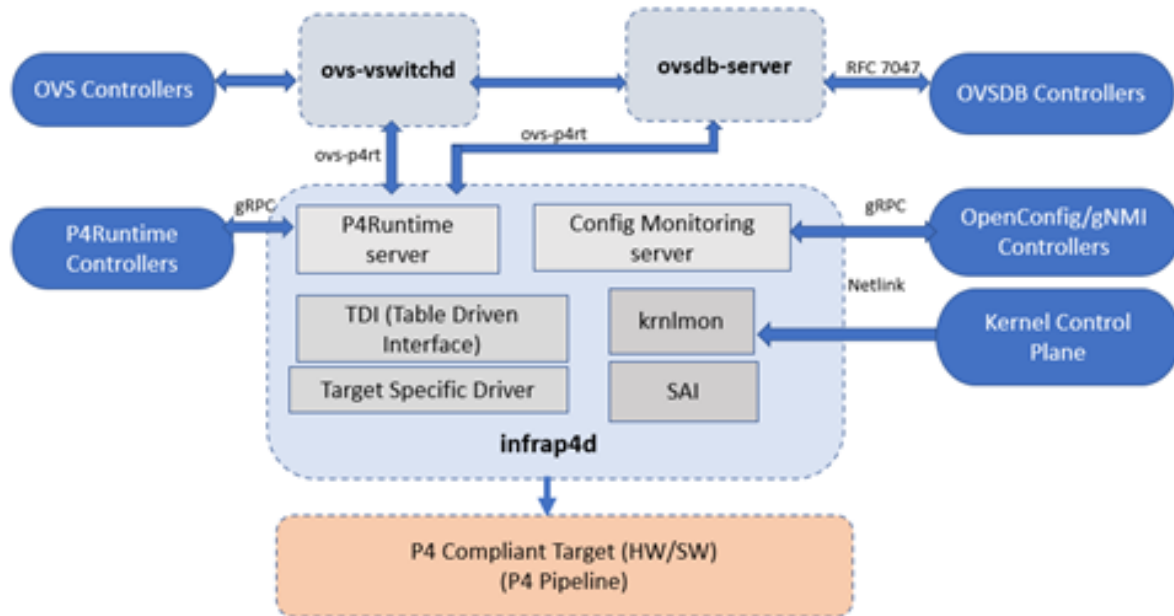
- IPDKをKVM仮想マシン環境で試してみる by Apresia Systems
  - <https://www.apresiatac.jp/blog/202207226950/>
- p4-guide by Andy Fingerhut
  - <https://github.com/jafingerhut/p4-guide/blob/master/ipdk/23.01/README-install-ipdk-networking-container-ubuntu-20.04-and-test.md>



# IPDK Networking Recipe (P4 Control Plane)

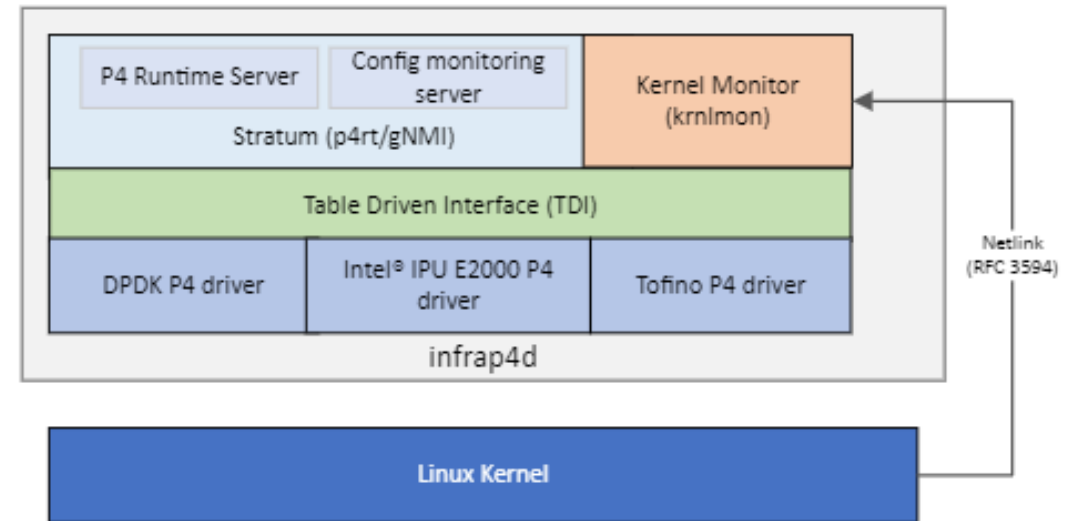
<https://github.com/ipdk-io/networking-recipe>

The IPDK Networking Recipe (originally P4-OVS Split Architecture) modularizes P4-OVS and reduces coupling between its components, making the code easier to maintain and more suitable for upstreaming. It moves the P4-specific components of the integrated architecture of P4-OVS to a separate process called `infrap4d`.



## infrap4d

Infrap4d integrates Stratum, the Kernel Monitor (**krnlmon**), Switch Abstraction Interface (**SAI**), Table Driven Interface (**TDI**), and a P4 target driver into a separate process (**daemon**).



# OPI Project

## Open Programmable Infrastructure

Members: ARM, Dell, F5, Intel, Keysight, Marvell, NVIDIA, RedHat, Tencent, ZTE

### 背景

- CPU性能（周波数・コア数）向上スピードの頭打ちや、省エネニーズによる消費電力あたりの処理性能向上を実現するため、ドメイン特化型プロセッサの利用拡大が進んでいる
- しかし、現状は各プロセッサやデバイス毎に異なるフレームワークやAPIが利用されているため、ハードウェアに応じたアプリケーションやミドルウェアの開発が必要となり、ユーザーには高い技術力とリソース（コスト）が求められる

### 目的

（上記課題を解決するため）

- DPU/IPU-like な技術をベースにした、次世代アーキテクチャやフレームワークのための、（デファクト）スタンダードを用いた、コミュニティ主導のオープンなエコシステムを育成する

<https://opiproject.org/>

### 組織（Governance）

2023年1月27日現在

#### Governing Board

<https://opiproject.org/board/>

Hao Chen (ZTE)  
Dror Goldenberg (NVIDIA)  
Michael Lynch (Intel)  
Shekhar Mishra (Dell)  
Joel Moses (F5)  
Kris Murphy (Red Hat)  
Venkat Pulella (Keysight)  
yachenwang(王亚晨)  
(Tencent)  
Cary Ussery (Marvell)

#### TSC members

<https://opiproject.org/tsc/>

Prasun Kapoor (Marvell)  
Kyle Mestery (Intel)  
Tim Michels (F5)  
Tzahi Oved (NVIDIA)  
Venkat Pulella (Keysight)  
Steve Royer (Red Hat)  
Chair: Joseph White (Dell)  
Richard Wu (Tencent)  
Songming Yan (TE)

#### Sub Groups

<https://opiproject.org/subgroups/>

Developer Platform/PoC/Reference Architecture  
Provisioning and Platform Management  
OPI API and Behavioral Model  
Use Case  
Outreach Committee



# OPI / IPDK の歴史

- 2021年7月：Diamond Bluff（後のOPI）GitHubレポジトリ作成 ⇒ <https://github.com/Diamond-Bluff/>
- 2021年10月頃：Intel, RedHat, F5 によるDiamond Bluff（後のOPI）の議論が始まった（[参考メール](#)）
- 2021年10月：Intel IPDK project Web&GitHub 公開 ⇒ <https://ipdk.io/> | <https://github.com/ipdk-io/>
- 2022年3月：Diamond Bluff 一般公開（GitHub, Slackなど）
  - 2022年3月15~16日：OPI イベントの開催（Co-organizers: F5, Intel, RedHat） -> Play List: [Day 1](#), [Day 2](#)
  - 2022年3月24日にオリエンテーション開催（テレカン） -> "[Diamond Bluff Orientation session \(2022-03-24 08\\_07 GMT-7\).mp4](#)"
- 2022年4月：Diamond Bluff と IPDK が一緒となり OPI に名称変更
  - IPDKはひとまずOPIのsub-project扱い
  - <https://github.com/Diamond-Bluff/> は閉鎖（アーカイブ）
- 2022年7月：Technical Charter 決定・公開
  - [https://opiproject.org/docs/Open\\_Programmable\\_Infrastructure\\_Technical\\_Charter\\_Final-06-9-2022.pdf](https://opiproject.org/docs/Open_Programmable_Infrastructure_Technical_Charter_Final-06-9-2022.pdf)
- 2022年8月：Mailing List 開設 & 参加募集 ⇒ <https://lists.opiproject.org/g/main>
- 2022年9月：sub-projects の在り方などがMailing Listにて議論される ⇒ <https://lists.opiproject.org/g/tsc/message/86>

主なコントリビューション：Intel によるIPDK提供、Pensando(AMD) によるAPI提供、Dellによる storage protobuf/gRPC specification の提供

# OPIの組織 & 活動状況

- 組織運営 (Governance)
  - Governing Board と Technical Steering Committeeによる運営
- Sub Groups (Working Groups)
  - 各Teamで仕様提案が活発に行われ、Team毎のWeekly Meetingで議論されている
  - Slackはほぼ毎日書き込みあり、成果物はGitHubで管理・更新されている
  - 広報活動はOutreach Teamによるホームページの更新、年2～3回のイベントでの講演 (OPI紹介) を実施
- リリース
  - 6ヶ月毎のリリース (実際に実行されるかは要確認)

[https://github.com/opiproject/opi/blob/main/Policies/OPI\\_RELEASE\\_APPROACH.md](https://github.com/opiproject/opi/blob/main/Policies/OPI_RELEASE_APPROACH.md)

## 組織 (Governance)

2023年1月27日現在

### Governing Board

<https://opiproject.org/board/>

Hao Chen (ZTE)  
Dror Goldenberg (NVIDIA)  
Michael Lynch (Intel)  
Shekhar Mishra (Dell)  
Joel Moses (F5)  
Kris Murphy (Red Hat)  
Venkat Pallela (Keysight)  
yachenwang(王亚晨)  
(Tencent)  
Cary Ussery (Marvell)

### TSC members

<https://opiproject.org/tsc/>

Prasun Kapoor (Marvell)  
Kyle Mestery (Intel)  
Tim Michels (F5)  
Tzahi Oved (NVIDIA)  
Venkat Pallela (Keysight)  
Steve Royer (Red Hat)  
Chair: Joseph White (Dell)  
Richard Wu (Tencent)  
Songming Yan (TE)

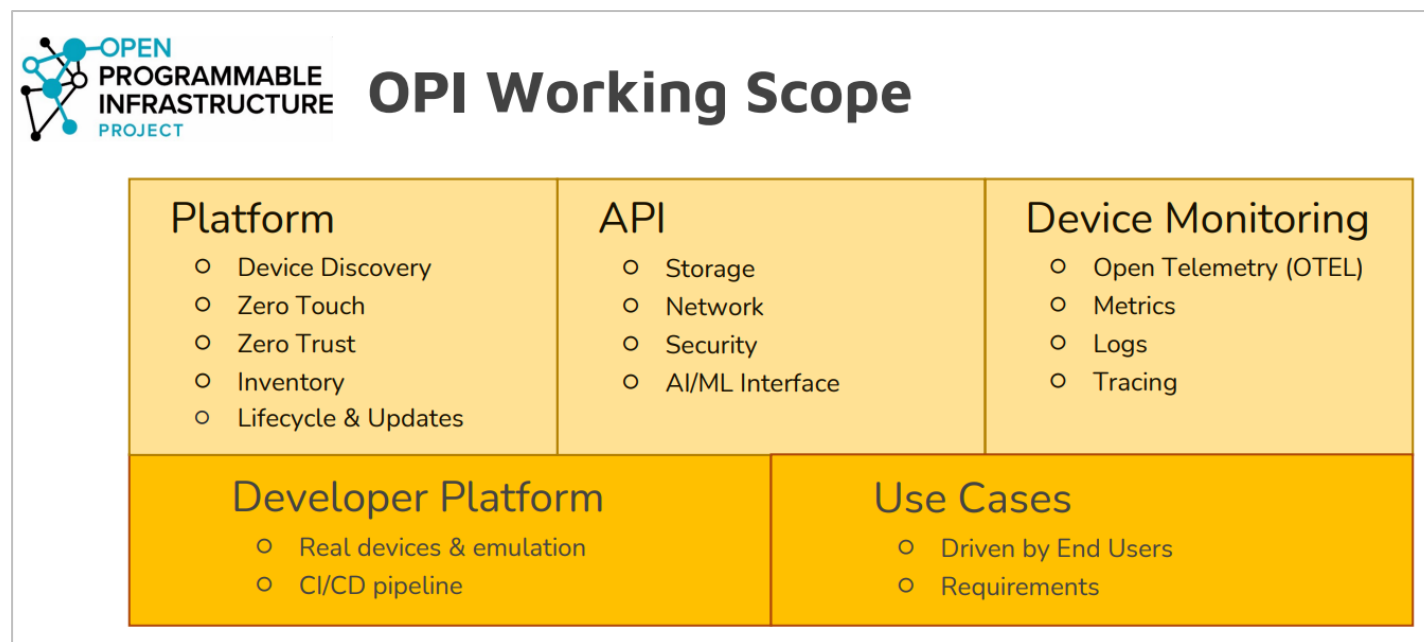
### Sub Groups

<https://opiproject.org/subgroups/>

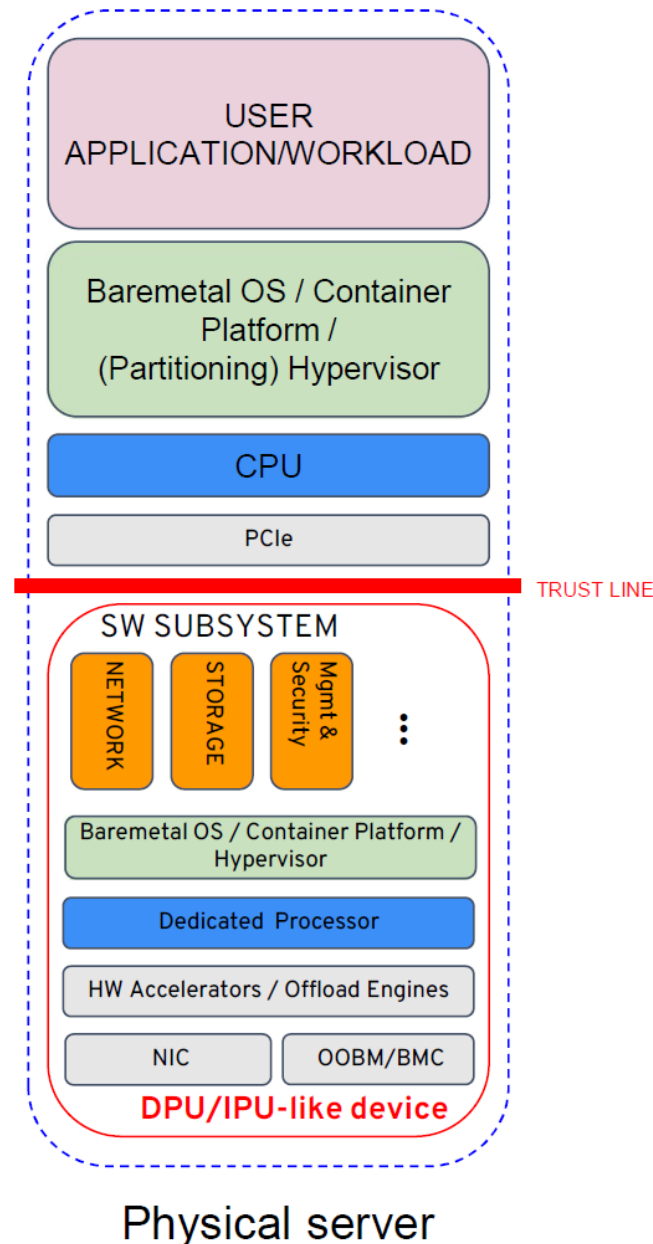
Developer Platform/PoC/Reference Architecture  
Provisioning and Platform Management  
OPI API and Behavioral Model  
Use Case  
Outreach Committee

# OPIのスコープ

- プラットフォームとして必要な、Zero Touch 機能や Lifecycle 管理
- 様々なユースケースに応じた API の定義
- デバイスマニタリング (OTELを採用)
- 開発環境の整備
- ユースケースの議論、POCの実施



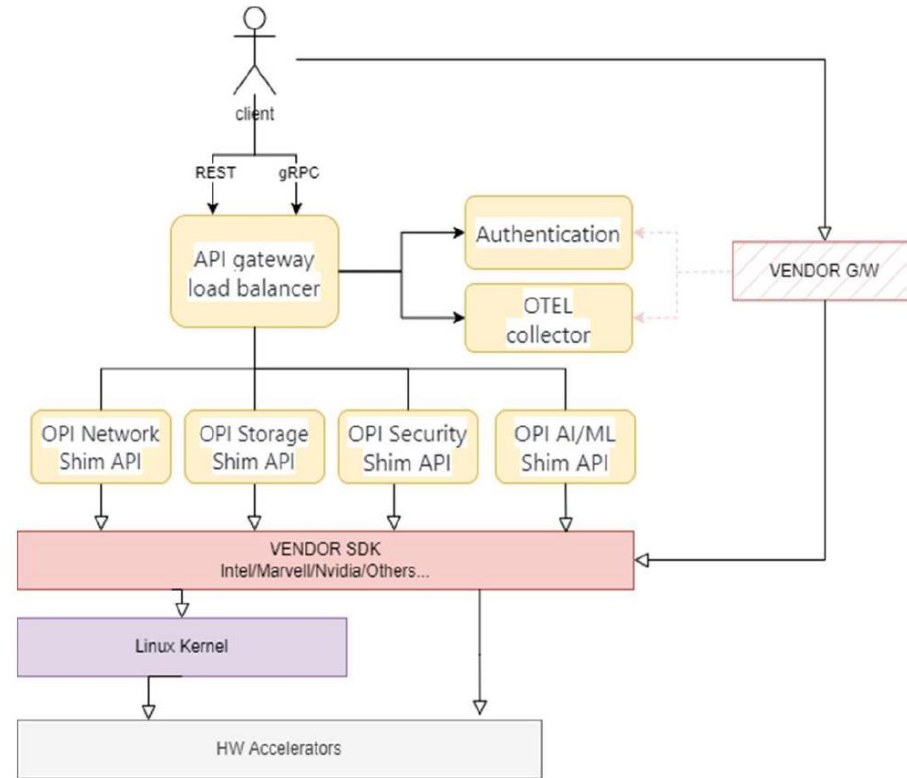
<https://opiproject.org/presentations/IntelON-OPI-IPDK.pdf>



Physical server

# API Mechanism

- Utilize gRPC for the config/control interface
- Use API Gateway
  - Direct delivery of gRPC messages to appropriate shim layer
  - gRPC to REST translation
  - Support gNMI and gNOI
- Expose VF/PF for the data consumption interface



# コミュニティの状況

- 現在は必要とされる機能が議論され実装されている状態
- 成果を待つのではなく、必要な機能、使いやすいデザインをインプットし一緒に実装していくことが重要なフェーズ



# P4 + IPDK 勉強会 & ハンズオン by 日本P4ユーザ会

IPDK Networking recipeをビルドした仮想マシンを使った  
P4プログラムを実際に動作させるハンズオンを実施

<https://github.com/ipdk-io/networking-recipe/blob/v23.01/docs/ipdk-dpdk.md>

<https://connpass.com/event/284037/>



6月 27 P4 + IPDK 勉強会 & ハンズオン  
主催: 日本 P4 ユーザ会

**P4 users Japan**  
日本P4ユーザ会

6月27日  
参加者18名  
+講師

## 【勉強会 パート】

- データプレンプログラミングとは
- P4言語の基礎
- P4言語でプログラミング可能なデバイスの紹介
- IPDKとは
- IPDKの各Recipeの紹介
- FPGA NICご紹介 (会場スポンサーより)

## 【ハンズオン パート】

- IPDK Networking Recipeのデモの実行
- 上記デモの内部動作の説明
- IPDK Networking Recipeに含まれるP4プログラムの改造 + 動作試験

日本P4ユーザ会 Slack : <https://p4users.org/slack-channel/>



# まとめ & 議論

# サーバーサイド・アクセラレーションのユースケース

## CPUのオフロード&セキュリティ向上 (主にデータセンター)

- サーバーサイドで必要なネットワーク機能
- クラウド基盤の高速化 (オフロード)
- セキュリティ向上 (ユーザと管理ドメインの分離)
- ストレージアクセスの高速化
- 高速暗号化 (IPsec)
- 分散ファイヤーウォール

## ネットワーク製品 (機能) の高速化・効率化 (主に通信事業者)

- ネットワーク仮想アプライアンス (NVA)
- VPN/NAT/Cloud GW
- ORAN, UPF (モバイル)
- 高速化&電力効率向上 (MEC対応)

どちらを想定するか? で必要な技術やコミュニティが異なる  
(もちろん重複もある)

# 誰が使っているのか？使えるのか？

## CPUのオフロード&セキュリティ向上 (主にデータセンター)

- サーバーサイドで必要なネットワーク機能
- クラウド基盤の高速化 (オフロード)
- セキュリティ向上 (ユーザと管理ドメインの分離)
- ストレージアクセスの高速化
- 高速暗号化 (IPsec)
- 分散ファイヤーウォール

## ネットワーク製品 (機能) の高速化・効率化 (主に通信事業者)

- ネットワーク仮想アプライアンス (NVA)
- VPN/NAT/Cloud GW
- ORAN, UPF (モバイル)
- 高速化&電力効率向上 (MEC対応)

### 既存&想定ユーザ

クラウド事業者・ホスティング事業者  
サービス事業者 (Hyper Scaler を含む)

モバイル・固定通信事業者 (MNO・CSP)  
ネットワーク機器ベンダ

# エコシステム拡大に向け、解決すべき課題（例）

- エコシステムの消費者として利用可能なライブラリやフレームワーク
  - 中小エンタープライズ企業は（Hyper Scalers と異なり）ライブラリやフレームワークの研究開発をSmartNICやチップベンダと推進するのが難しい
  - 提供を期待されるのは、SmartNICベンダ、チップベンダ、NFVベンダ、等
  - オープンコミュニティ等を通じた、ユーザニーズのインプットが必要
- オブザーバビリティの向上
  - トラフィック状況の可視化（フロー、テレメトリ、統計情報）
  - デバッグに有用な情報やフレームワーク



# 議論ポイント（例）

- どのように活用可能か？活用したいか？（ユースケース）
- どのような課題の解決が必要なのか？
- コミュニティとの関わり
  - コミュニティ参加へのハードルは？
  - 参加する人がいたら、フィードバックして欲しい事はあるか？

# Appendix

参考資料

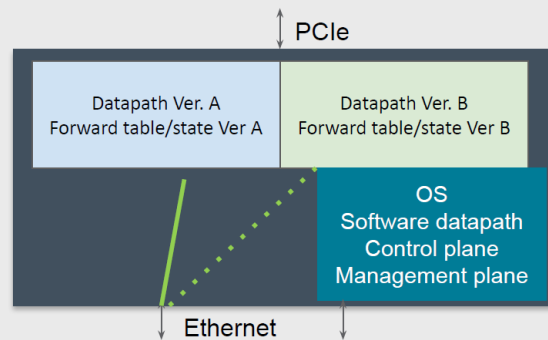
# その他 SmartNIC 関連で押さえておくべき話題

- CXL等、次世代インターコネクト
- OCP Open Domain-Specific Architecture (ODSA)
  - how to accelerate the design and production of new domain-specific hardware
  - IP blocks (chiplets)
- VMware Project Monterey
  - ESXi ハイパーバイザーをSmartNIC上で動かす取組
  - SmartNIC: Intel, Pensando, Nvidia
  - サーバ: Dell, HPE, Lenovo
  - The goals behind this project include:
    - Improving security and manageability by using the SmartNIC hypervisor to act as an air gap and secure supervisor
    - Enhancing storage and network I/O performance by offloading these to the SmartNIC
    - Providing bare metal OS support, allowing the SmartNIC hypervisor to manage and deploy the OS on the host

# (参考) SmartNIC 内部データパス 無停止アップデート

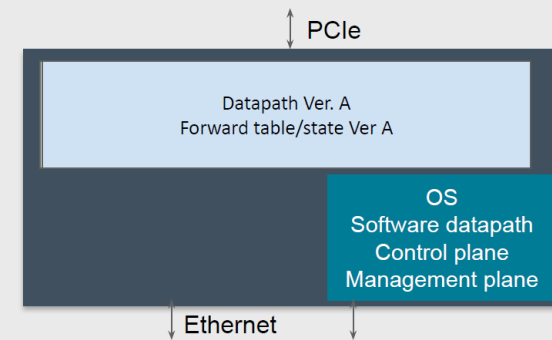
## Seamless Upgrades in Production without Host Reboot

### Hitless Software Upgrade



- All links remain up
- Minimum to no traffic impact
- Build new P4 program in partition B
- Build table in partition B and sync flow state from A
- Continue data forwarding with A until B is fully built
- Need extra memory for Partition B

### Graceful Software Upgrade



- PCIe link remains up during upgrade
- Upgrade datapath program
- Rebuild forwarding table and relearn session state (if needed)
- No restriction on kernel upgrade
- No extra memory reserved for upgrade