



オープンNOS SONiCのコミュニティ動向

日本電信電話株式会社

NTTネットワークイノベーションセンタ

中野 寛二

- NTTではBeyond5G時代に向けたネットワークサービス基盤の構成要素として、オープン技術に立脚したパケット転送技術としてSONiCの機能拡張に関する研究開発を進めている。
- SONiCは、近年データセンタを中心に急速に利用が拡大しており、オープンNOSではデファクトスタンダードとなりつつある。
- 一方で従来のキャリアオペレーションなどを考慮すると機能改善すべき点もあり、SONiCにキャリア向け機能の追加や既存機能の改善を行っている。
- 本発表では、SONiCの動向を解説するとともに、次期リリースに向けたSONiCコミュニティへの機能拡張提案状況(次期リリースへの唯一の国内事業者からの提案)やアップストリーム作業を通じて得た経験を共有する。

- SONiCについて
- SONiCの動向
- SONiCの課題とNTT研究所における対応
 - SONiCの課題とアプローチ
 - SONiCコミュニティへのupstream方法とupstream状況
 - NETCONF機能拡張
 - Nex Hop Group機能対応
- まとめ

SONiCについて

- SONiCはオープンなネットワークOSである。
- MicrosoftがOCP(Open Compute Project)に2016年に寄贈し、2022まではOCP Networking傘下、2022以降はLinux Foundationで管理されている。
- 国内の導入企業としてKDDI、LINEがある。

出展: <https://www.zdnet.com/article/microsoft-submits-new-open-sourced-networking-components-to-open-compute-project/>

出展: <https://www.linuxfoundation.org/press/press-release/software-for-open-networking-in-the-cloud-sonic-moves-to-the-linux-foundation>

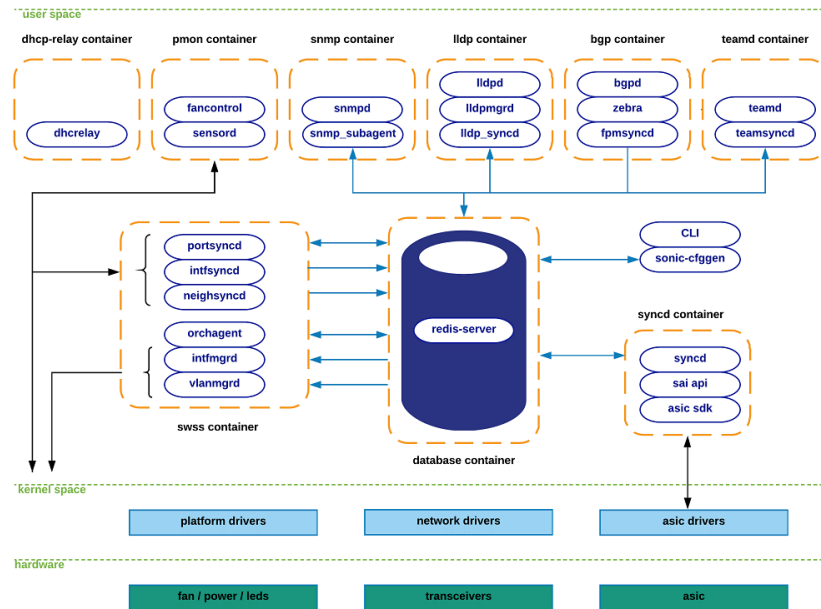
出展: <https://www.janog.gr.jp/meeting/janog51/sonic/>

出典: <https://www.janog.gr.jp/meeting/janog52/sonicztp/>

SONiCの技術概要



- コンテナベースアーキテクチャ
 - Docker (標準)
 - Kubernetes
- Redis DB集中管理システム
- Pub/Subメッセージング(Redis)
- SAIを用いたスイッチASIC制御



SONiCの動向

- SONiCは5月頃と11月頃の2回リリースしている。
- リリースまでの流れは次の通りである。

202311 Releaseの例

Date	Milestone
5/15/2023	Call for feature candidate in community - document subgroup
5/31/2023	202311 release kick-off
6/7/2023	Feature candidate review in community
6/30/2023	Feature submission deadline
9/16/2023	HLD Review deadline
9/30/2023	All code PRs are created
10/15/2023	Prepare SAI header and SAI SDK integration
10/31/2023	Fork release branch
11/30/2023	Ship

出典: https://github.com/sonic-net/SONiC/wiki/release_train

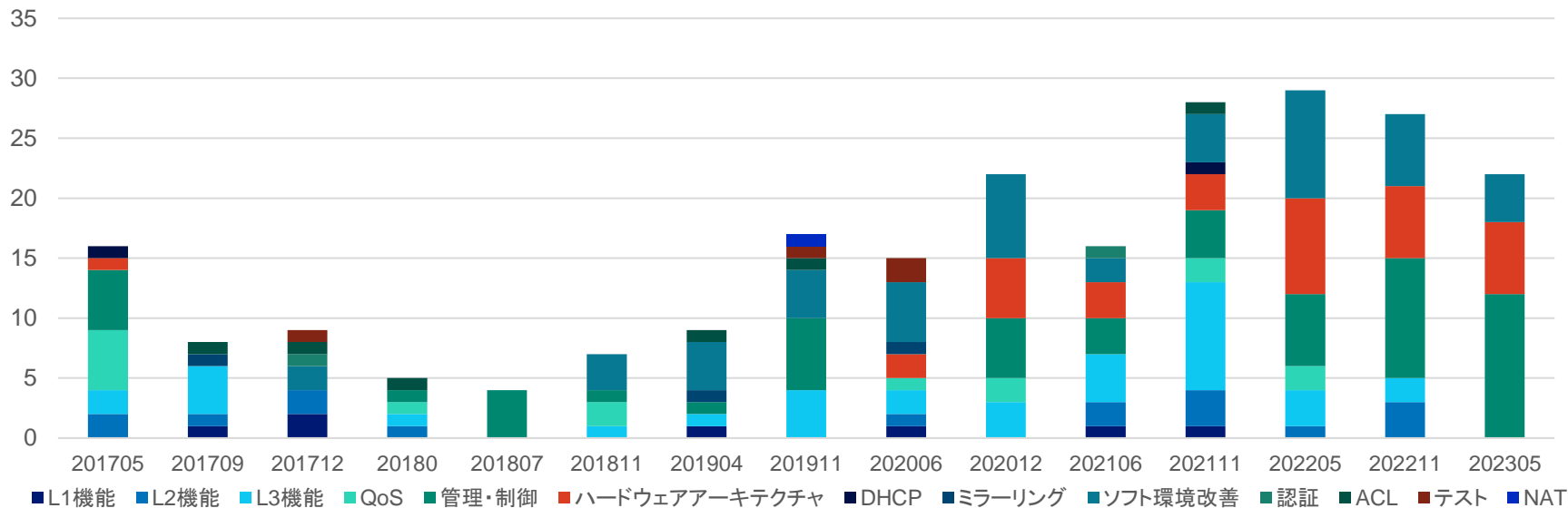
出典: <https://lists.sonicfoundation.dev/g/sonic-dev/message/317?p=%2C%2C%2C20%2C0%2C0%2C0%3A%3Acreated%2C0%2C202311%2C20%2C2%2C40%2C99214915>

dev/message/317?p=%2C%2C%2C20%2C0%2C0%2C0%3A%3Acreated%2C0%2C202311%2C20%2C2%2C40%2C99214915

SONiC 過去の動向



- SONiCは基本的にDCユースケース向けに開発が進み、機能数は最初のリリースを除くと2022年までは増加傾向だったがピークアウトしている傾向

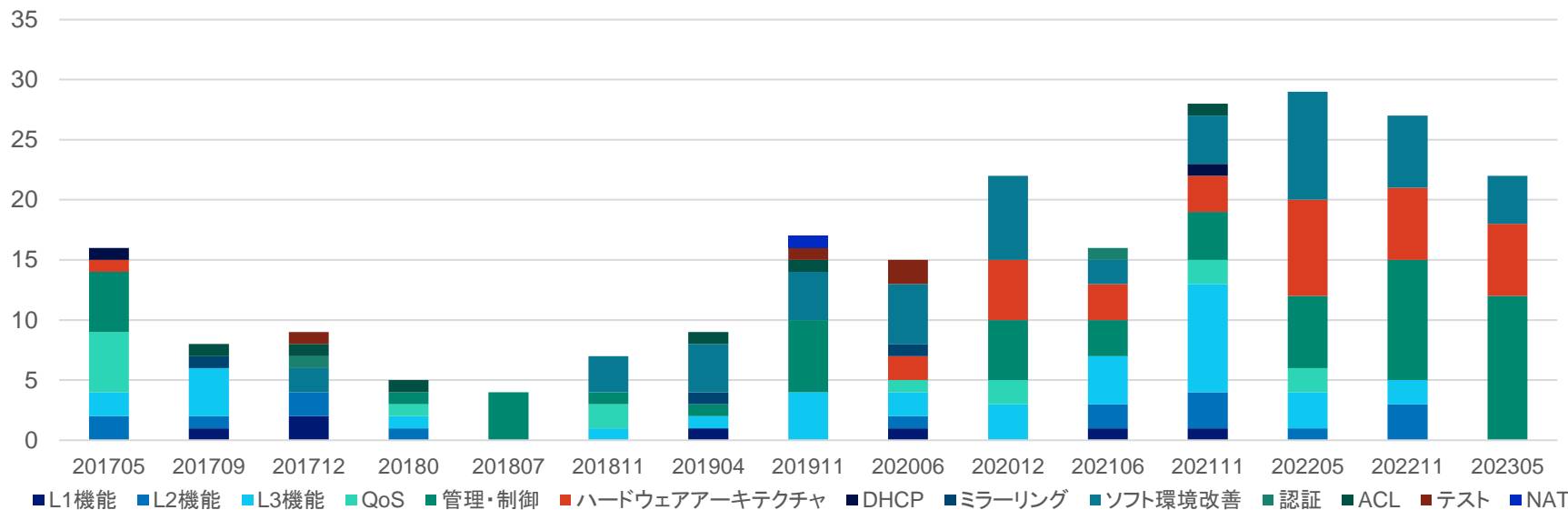


※機能分類はリリースノートより独自で分類

SONiC 過去の動向



- 2017年 : BGP、VLAN、LAG、ECMP、SNMPなど
- 2018年 : VRF、VXLAN、Warm Rebootなど
- 2019年 : BGP-EVPN、ZTP、NAT、Management Frameworkなど

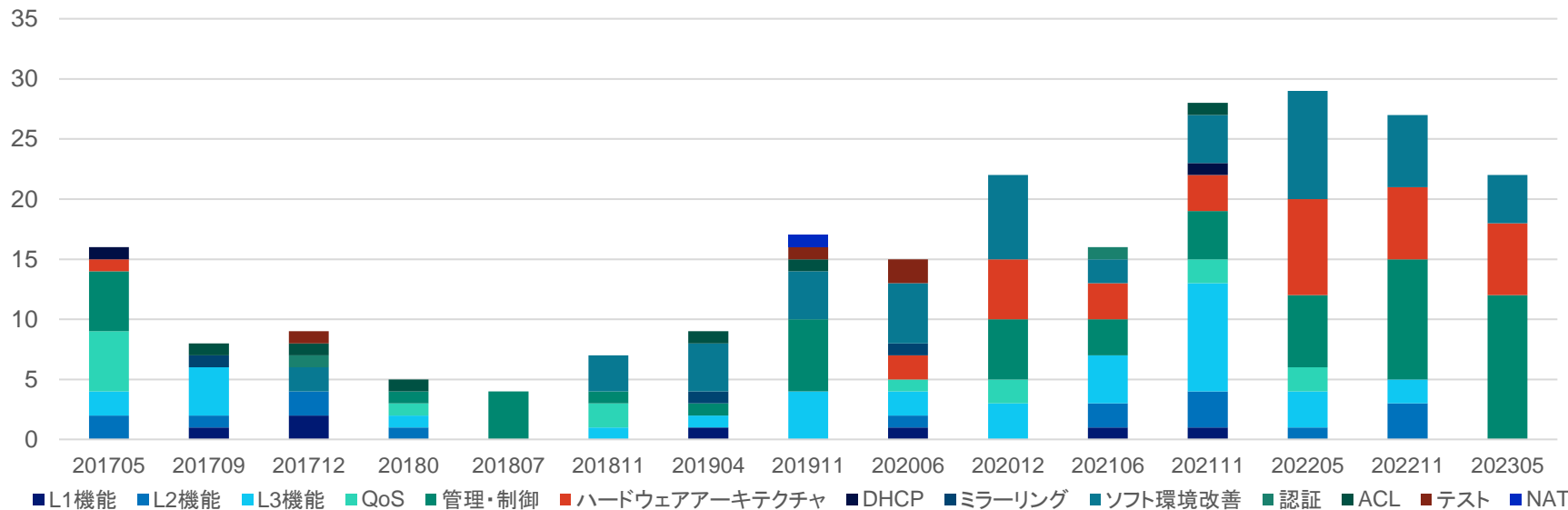


※機能分類はリリースノートより独自で分類

SONiC 過去の動向



- 2020年 : ECMP、Port Mirroring、Kubernetesなど
- 2021年 : BGP Unnumbered、SRv6、MC-LAG、P4(PINS)など
- 2022年 : SRv6 μ SID、Config Rollback、SmartNIC(DASH)、gNMIなど

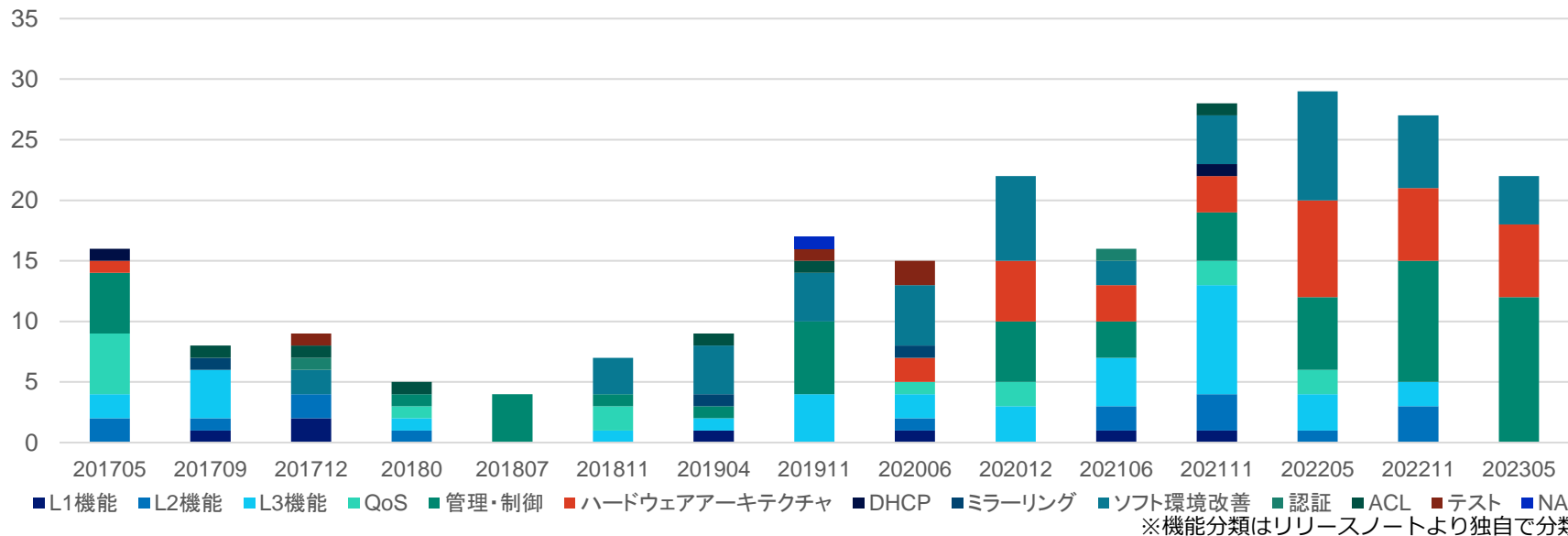


※機能分類はリリースノートより独自で分類

SONiC 過去の動向



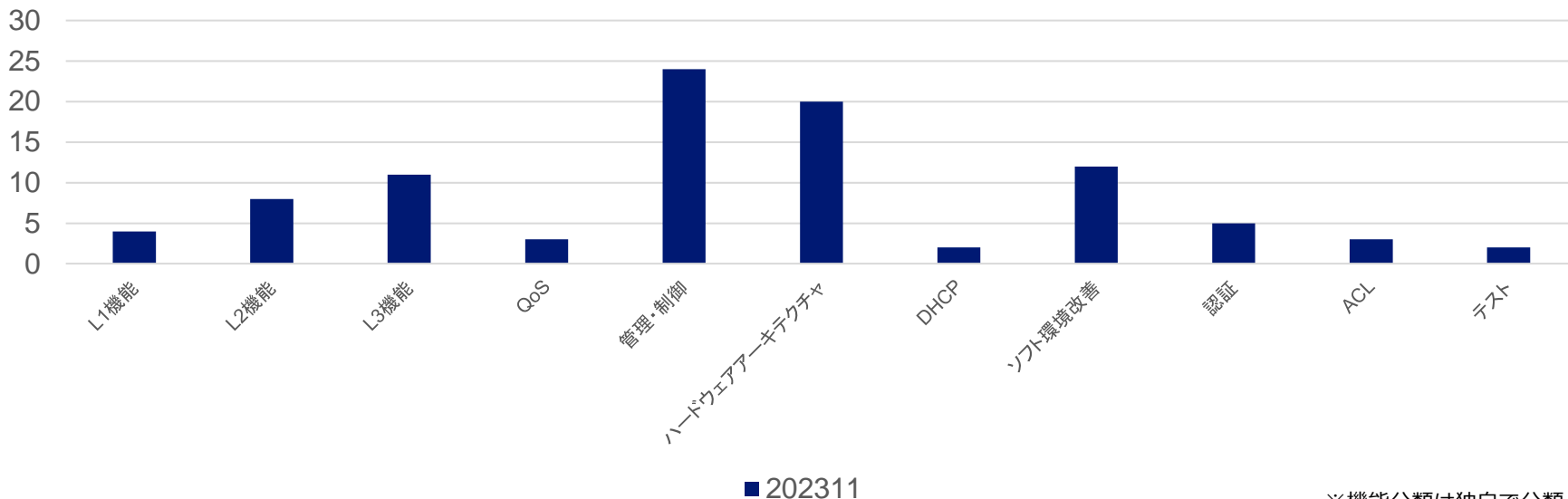
- 2023年: FPGAサポート、ASICのFIBに登録していない経路広報の抑制、YANGモデルの拡充などの機能リリース
- 近年はハードウェアアーキテクチャ(FPGA, SmartNIC)サポート、管理・制御機能(YANGサポート等)拡充、ソフトウェアインフラストラクチャの環境改善が多く行われている



SONiCの次期リリース



- 次期リリースの202311について94個の機能提案がされている。
- 2023/10/24時点で11個の機能がリリース予定となっている。



※機能分類は独自で分類

OCN Global Summit 2023におけるSONiC



- 今年のハイライトとして次の機能サポートについて説明
 - Security&Build, Management, Diagnostic, Reliability, System

SONiC 2023 Feature Highlight



- Container permission hardening
- UEFI key management
- REST Server DoS Attack fix
- TACACS NSS vul fix
- Build improvement w/ version caching
- 2 stages build w/ reduced time

Security & Build

- CMIS host management enhancements
- LDAP support
- gNMI subscription for OpenConfig YANG (UMF)
- gNMI save-on-set (UMF)
- NetConf server support
- gNOI API support
- NTP Server support

Management

- sFlow support
- Two Way Active Measurement Protocol Light support
- Auto tech support during SAI failure
- SRv6 path tracing
- Logs uploading to remote syslog server
- Static route BFD
- ECN & WRED statistics

Diagnostic

SONiC 2023 Feature Highlight - Cont



- FRR 8.5.1 support
- LACP PDU timeout optimization for warm-reboot with customized retry count
- Preserve CoPP table during fast-reboot to reduce dataplane downtime
- Config reload w/ faster system init
- FIB suppress announcements of routes not installed in HW
- Voltage and current sensor monitoring
- Link event damping

Reliability

- SRv6 L3 adjacency support
- SmartSwitch arch
- Multiple Spanning Tree Protocol support
- Debian 12 support with base OS
- Docker migration to Bullseye (Debian 11)
- Virtual SONiC Helper
- Local clock setting
- FPGA via I2C/PCIe in PDF
- Factory reset support

System



- SONiCコミュニティ update
 - 新しいSONiC ウェブサイトの公開
 - › 出典: <https://sonicfoundation.dev/>
 - Qualityの担保するためのドキュメント公開
 - › 出典: <https://github.com/sonic-net/SONiC/blob/master/doc/SONiC%20feature%20quality%20definition.md>
 - Working Groupの作成
 - › Routing WG, UMF WG, OTN WGなど
 - › 出典: <https://lists.sonicfoundation.dev/g/main/subgroups>

SONiCの課題とNTT研究所における対応

SONiCにおける現状見えている課題と検討したアプローチ



- マネジメントはキャリア観点で課題の抽出を実施
- 課題視した機能の中でNETCONFとfpm syncdについてアップストリームを実施

プレーン	分類	コミュニティSONiCの状況/課題	検討したアプローチ
マネジメント	設定変更コマンド体系	コマンド数が少ない。 L2/L3設定コマンド体系が異なる	KLISH CLI拡張 (Management Framework)
	外部制御プロトコル	設定項目が限定的 REST, gNMIのみ。旧来から利用するNETCONFに対応不可。相互接続性に難あり。	YANG拡張 (Management Framework) NETCONF拡張 (Management Framework)
コントロール	ルーティングプロトコル	標準ではBGPのみ利用可。OSPF等のIGPサポート不可。	COPP関連の設定ファイル変更
	ECMP経路処理	旧来の経路追加メッセージのみ利用可。Next hop Group (netlinkメッセージ)が利用不可	Fpmsyncd機能*拡張 *fpm形式のnetlinkメッセージの経路更新メッセージをRedis DBに同期する機能
データ	SAIのサポート	<ul style="list-style-type: none">• SAIサポートの対応状況がASIC毎に異なる。• BroadcomはSAIフルスペックをサポート不可	<ul style="list-style-type: none">• SAIの多くのAPIに準拠していたIntel Tofinoを検討 →Tofino開発中止のため再検討要。

SONiCへのUpstream方法



- SONiCに機能提案を行う際に次の開発サイクルを行う。

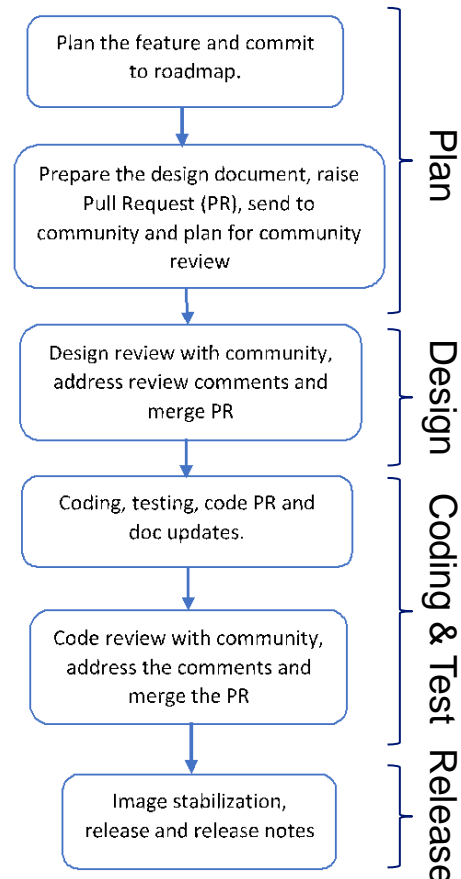
Plan : 次回リリースのロードマップに追加するように、1ヶ月前にコミュニティに機能提案する。また、HLDの概要を提案するのレビューの日付を決定する。

Design : HLDには詳細なデザインとSONiCアーキテクチャへの適合性を記載することが必要になる。HLDはコミュニティからレビューが行われる。

Coding & Test : HLDに従ってコーディングとテストを行う。HLDレビュー結果によってコード修正が必要となる。

Release : 機能を実装したブランチがリリース日に提供される。ブランチの動作検証/コード修正期間(1-2ヶ月)が設けてあり、その間に当該機能の検証/修正作業を行う。

- 当該ブランチのリリース日までに完成しなかった機能が次のリリース以降に実装される。



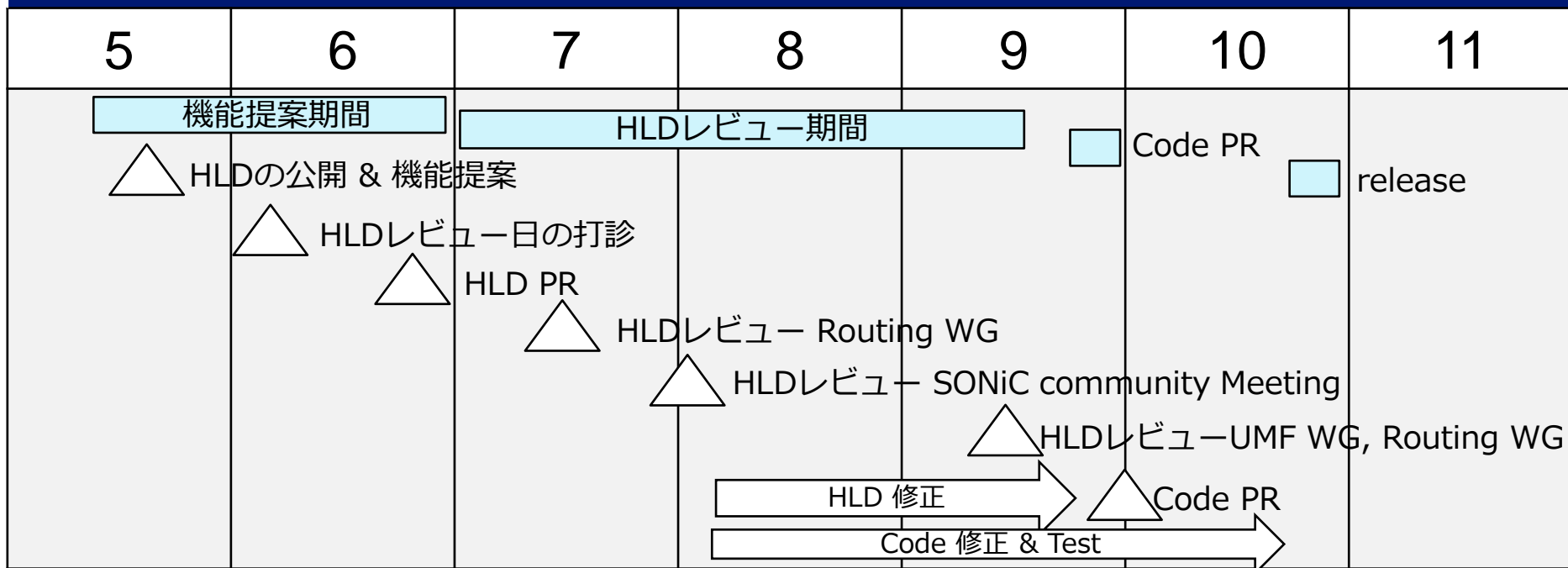
SONiCへのUpstream方法



- 202311リリースまでのスケジュール

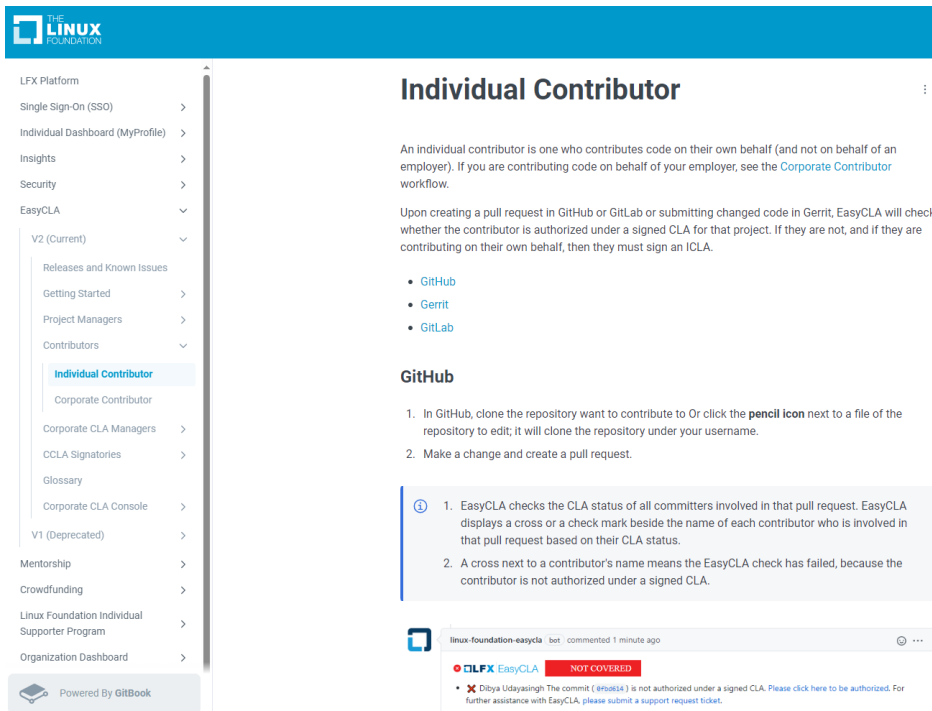
□ コミュニティのマイルストーン
△ NTTの状況

2023年



出典: <https://lists.sonicfoundation.dev/g/sonic-dev/message/317?p=%2C%2C%2C20%2C0%2C0%2C0%3A%3Acreated%2C0%2C202311%2C20%2C2%2C40%2C99214915>

- SONiCにコントリビューションする際はLinux FoundationのICLA(Individual CLA)にサインする必要がある。(企業名で活動する際は、属する企業のポリシーに従って上記に署名)
- 初めてPRを送る際にEasyCLAシステムでCLA署名を求められ、署名しないとCIが回らない。



The screenshot shows the 'Individual Contributor' page on the Linux Foundation EasyCLA platform. The page is divided into a left sidebar with navigation links and a main content area. The sidebar includes links for LFX Platform, Single Sign-On (SSO), Individual Dashboard (MyProfile), Insights, Security, EasyCLA, V2 (Current), Releases and Known Issues, Getting Started, Project Managers, Contributors, Individual Contributor (highlighted), Corporate Contributor, Corporate CLA Managers, CCLA Signatories, Glossary, Corporate CLA Console, V1 (Deprecated), Mentorship, Crowdfunding, Linux Foundation Individual Supporter Program, and Organization Dashboard. The main content area has a blue header with the Linux Foundation logo and the title 'Individual Contributor'. Below the header, there is a definition of an individual contributor and instructions on how to create a pull request. A list of links for GitHub, Gerrit, and GitLab is provided. A 'GitHub' section contains two steps: 1. In GitHub, clone the repository... 2. Make a change and create a pull request. A callout box with a blue 'i' icon contains two numbered points: 1. EasyCLA checks the CLA status... 2. A cross next to a contributor's name means the EasyCLA check has failed... At the bottom, there is a GitHub commit preview showing a 'NOT COVERED' status for a commit by Ditya Lidayasingh.

- HLDテンプレートに従ってデザインドキュメントの記載をする。

HLD Name [🔗](#)

Table of Content [🔗](#)

Revision [🔗](#)

Scope [🔗](#)

This section describes the scope of this high-level design document in SONiC.

Definitions/Abbreviations [🔗](#)

This section covers the abbreviation if any, used in this high-level design document and its definitions.

Overview [🔗](#)

The purpose of this section is to give an overview of high-level design document and its architecture implementation in SONiC.

Requirements [🔗](#)

This section list out all the requirements for the HLD coverage and exemptions (not supported) if any for this design.

Architecture Design [🔗](#)

This section covers the changes that are required in the SONiC architecture. In general, it is expected that the current architecture is not changed. This section should explain how the new feature/enhancement (module/sub-module) fits in the existing architecture.

If this feature is a SONiC Application Extension mention which changes (if any) needed in the Application Extension infrastructure to support new feature.

SAI API [🔗](#)

This section covers the changes made or new API added in SAI API for implementing this feature. If there is no change in SAI API for HLD feature, it should be explicitly mentioned in this section. This section should list the SAI APIs/objects used by the design so that silicon vendors can implement the required support in their SAI. Note that the SAI requirements should be discussed with SAI community during the design phase and ensure the required SAI support is implemented along with the feature/enhancement.

Configuration and management [🔗](#)

This section should have sub-sections for all types of configuration and management related design. Example sub-sections for "CLI" and "Config DB" are given below. Sub-sections related to data models (YANG, REST, gNMI, etc.) should be added as required. If there is breaking change which may impact existing platforms, please call out in the design and get platform vendors reviewed.

Manifest (if the feature is an Application Extension) [🔗](#)

Paste a preliminary manifest in a JSON format.

CLI/YANG model Enhancements [🔗](#)

This sub-section covers the addition/deletion/modification of CLI changes and YANG model changes needed for the feature in detail. If there is no change in CLI for HLD feature, it should be explicitly mentioned in this section. Note that the CLI changes should ensure downward compatibility with the previous/existing CLI. i.e. Users should be able to save and restore the CLI from previous release even after the new CLI is implemented. This should also explain the CLICK and/or KLISH related configuration/show in detail. <https://github.com/sonic-net/sonic-utilities/blob/master/doc/Command-Reference.md> needs to be updated with the corresponding CLI change.

オンラインでの議論



- SONiC community MeetingでHLDのレビューの実施
- HLDレビューの結果、WGで詳細議論を行う
 - NETCONFの提案: UMF WG
 - fpmsyncdの提案: Routing WG

SONiC Working Group on Routing

Working Group on Routing Area -- Routing Performance/Scale etc.

Created: Apr 3

178 Members, 79 Topics, Archive is visible to members only, Last Post: Oct 21

UMF Working Group

UMF Working Group Chair: Tomek Madejski(tmadejski@google.com) Charter: * Coordinate work on UMF to increase both development velocity and quality at the same time * For close cooperation between all contributors to the UMF code * Agree on coding and testing standards * Consider all important use cases * Discuss and decide on the direction of the evolution of UMF * Bring awareness and transparency across SONiC community...

Created: Aug 2

27 Members, 1 Topic, Archive is visible to members only, Last Post: Aug 23

既存機能に影響を与えないことが重要

- 既存機能に組み込む場合は初期動作は無効にする
 - 設定の有効化/無効化機能が必要
- 機能ごとにコンテナを作成するデザインが原則

コードPRの実施

- 実装したソースコードのPRを行う。
- このとき、PRの一覧をissueにまとめる必要がある。

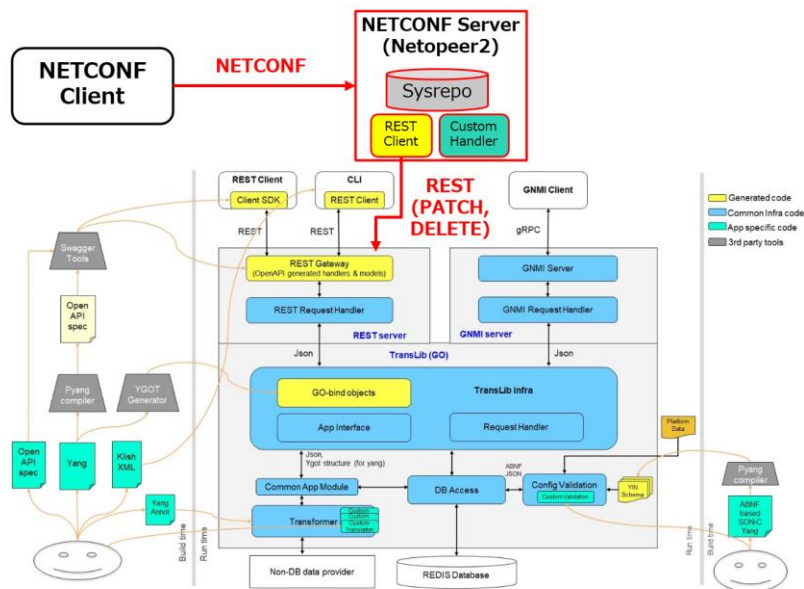
ntt-omw commented on Jul 13 • edited ▾

Fpmsyncd Next Hop Table Enhancement HLD

repo	PR title
sonic-buildimage	sonic-net/sonic-buildimage#16762
sonic-swss	sonic-net/sonic-swss#2919
sonic-mgmt-common	sonic-net/sonic-mgmt-common#107
sonic-mgmt-framework	sonic-net/sonic-mgmt-framework#122

提案したNETCONFの概要

- NETCONFを対応することで相互接続性の向上を行う。
- NETCONF ServerにNetopeer2を使用している。NETCONF ServerはNETCONFの設定値をRESTに変換してSONiC内部のRESTを操作する。



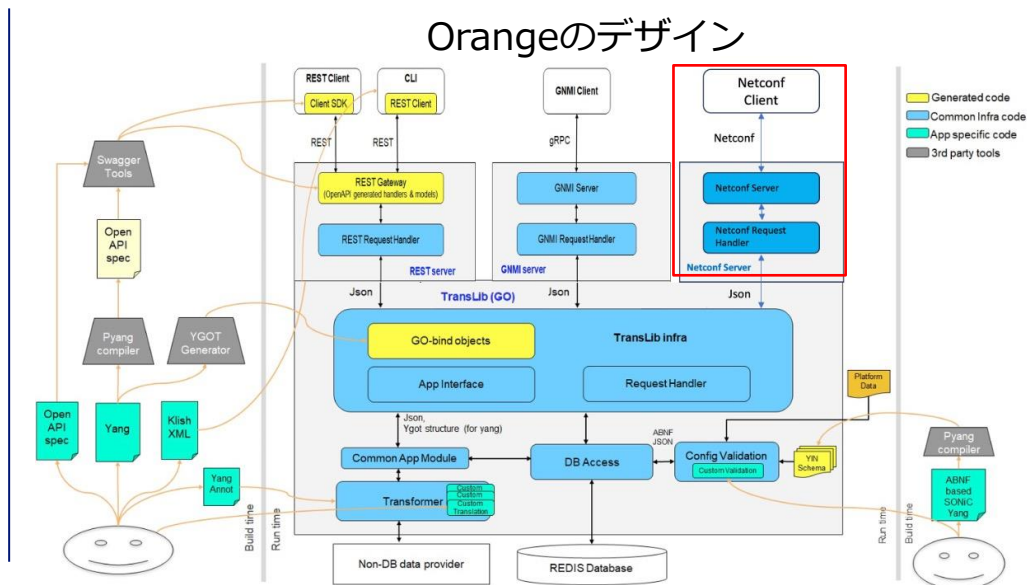
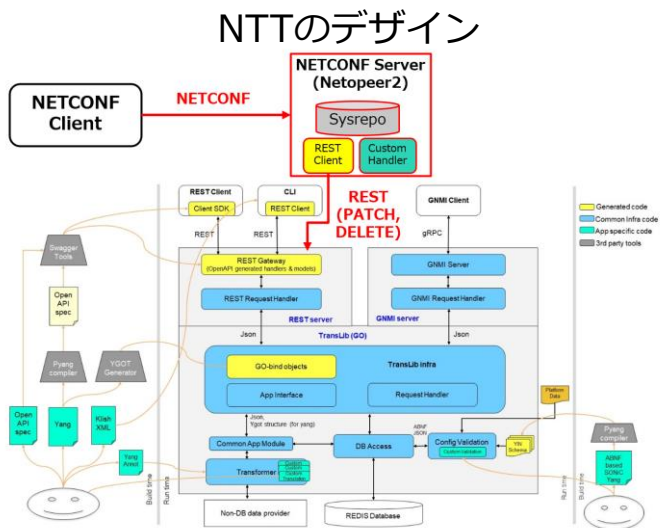
対応している機能

RFC Operation Name	Supported
edit-config	Partially
get-config	Partially
get	No
copy-config	No
delete-config	No
lock	No
unlock	No
close-session	Yes
kill-session	Yes

NETCONFのUpstream状況

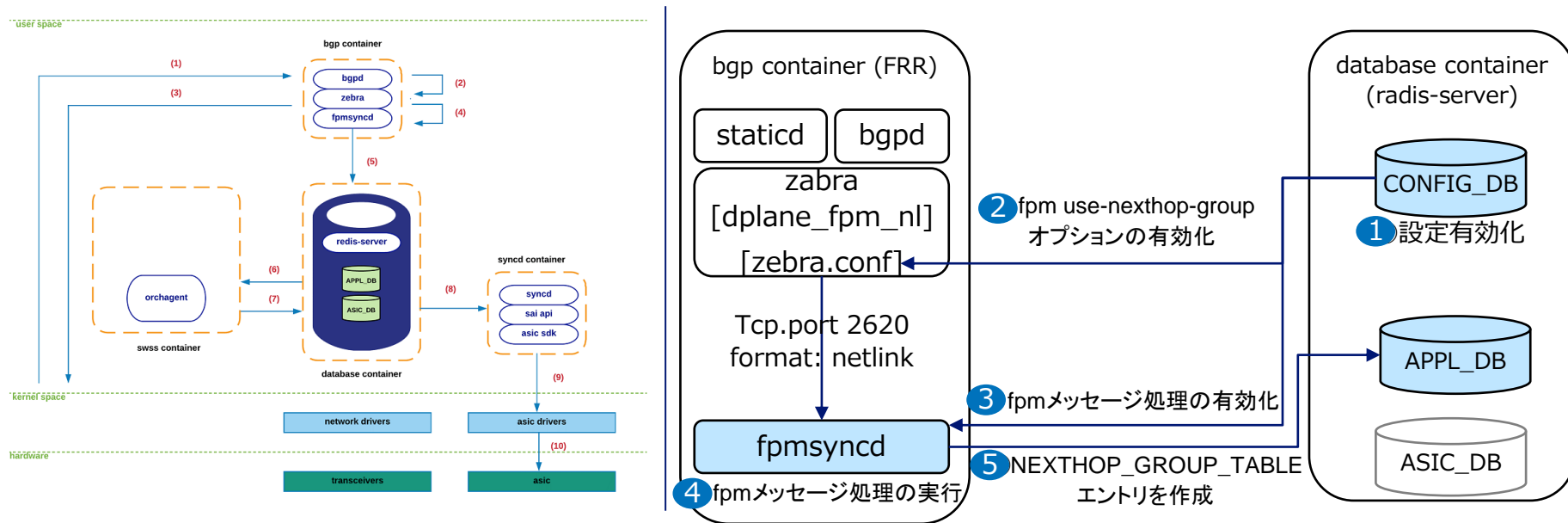


- DesignのHLDレビュー段階
- NETCONFのHLDをPRしたところフランスのテレコム会社のOrangeもNETCONFの実装をしていることが判明
- Orangeと議論しながらNETCONF拡張を目指している



提案したfpmsyncd拡張の概要

- 経路処理の効率化であるNext Hop GroupにDBからASICまでは対応している。
- FRRとAPPL_DB間のfpmsyncdはNext Hop Groupに非対応である。
- fpmsyncdをNext Hop Group対応を行い経路処理の効率化を目指す。

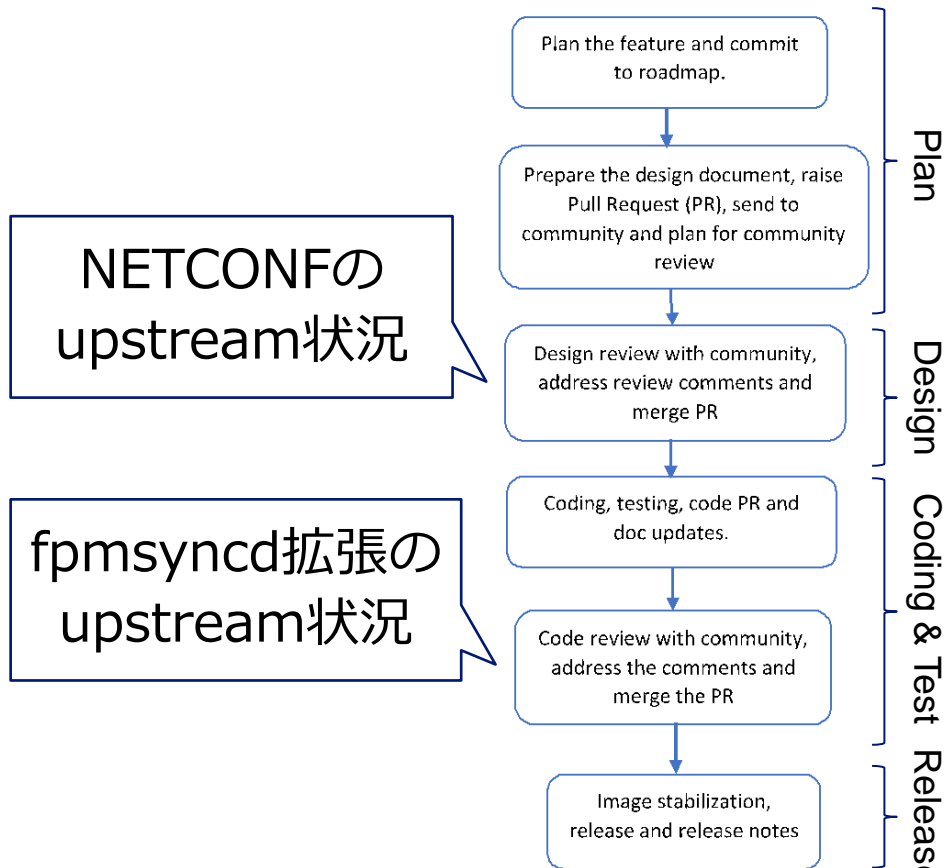


- Coding & Testのコード修正してテストが完了した段階
- PRしたfpmsyncd拡張のHLDはコミュニティミーティング及びRouting WGでのレビューにて、Next Hop Group機能の必要性が認められ機能追加が承認された
- HLDレビューでは、次の確認があり対応を行った
 - デフォルトはDisable動作が必要
 - Enable/Disableの追加するため修正
 - Warm boot時の動作確認

upstream状況のまとめ

NETCONFは、コミュニティから必要性の理解されたが、SONiCのデザイン原則に合わないため修正を要求

fpmsyncd拡張を行いレーティングの効率化はコミュニティから賛同を得られて順調に進んでいる状況



まとめ

- キャリアの考えているものとコミュニティのSONiCのギャップがあった。
- 一定のコントリビューション成果を得ることができた。
 - NETCONFに関してDC系事業者からは必要性の理解されたが、SONiCのデザイン原則に合わないため修正を要求された
 - ルーティングの効率化に関してはDC事業者からも賛同が得られた
- コントリビューションを行う中で得られた知見
 - 既存機能に影響を与えないことが重要
 - › 既存機能に組み込む場合は初期動作は無効にする
 - › 設定の有効化/無効化機能が必要
 - › 機能ごとにコンテナを作成するデザインが原則

謝辞



本研究成果は、国立研究開発法人情報通信研究機構（NICT）の委託研究（04801）により得られたものです。

Your Value Partner