

# クラウドネイティブな新しいOSSのNWコントローラー 『Kuesta(ケスタ)』の概要と実用化への取り組み



2023年11月9日  
NTTコミュニケーションズ株式会社  
イノベーションセンター  
坂井 立晟

# 自己紹介スライド

- NTTコミュニケーションズ
- 開発エンジニア
  
- 坂井 立晟 (さかい たつき)
- 略歴
  - 2021年 NTTコミュニケーションズ入社
  - 伝送NWオーケストレーター開発
  - 社内検証NWの設定自動化ツール開発
  - 現在: Kuesta(ケスタ)開発
- メールアドレス: [tatsuki.sakai@ntt.com](mailto:tatsuki.sakai@ntt.com)





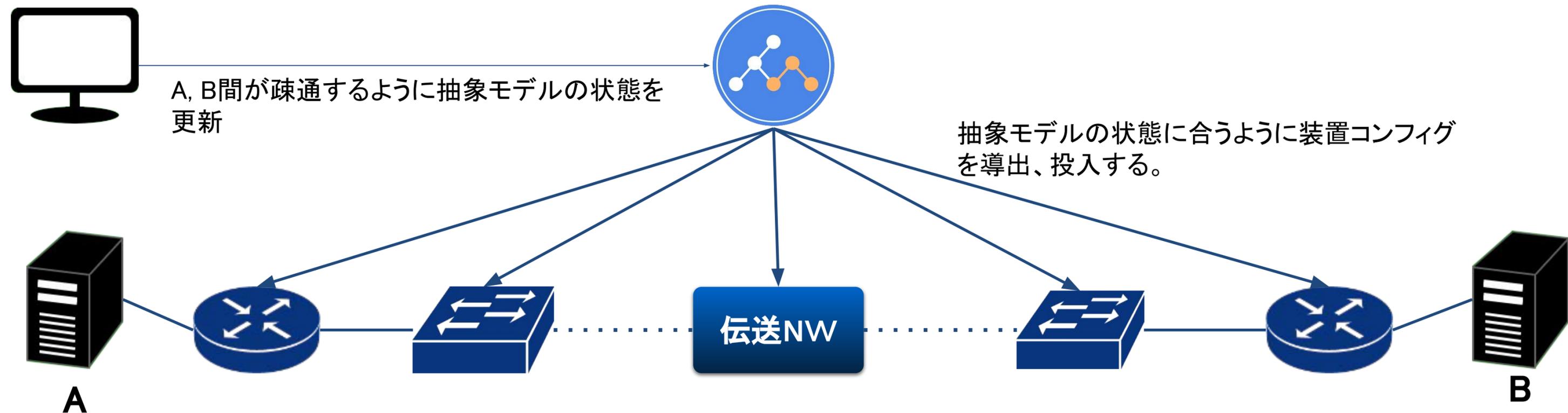
# Kuesta

Github: <https://github.com/nttcom/kuesta>

Community site: <https://nttcom.github.io/kuesta-website/community>

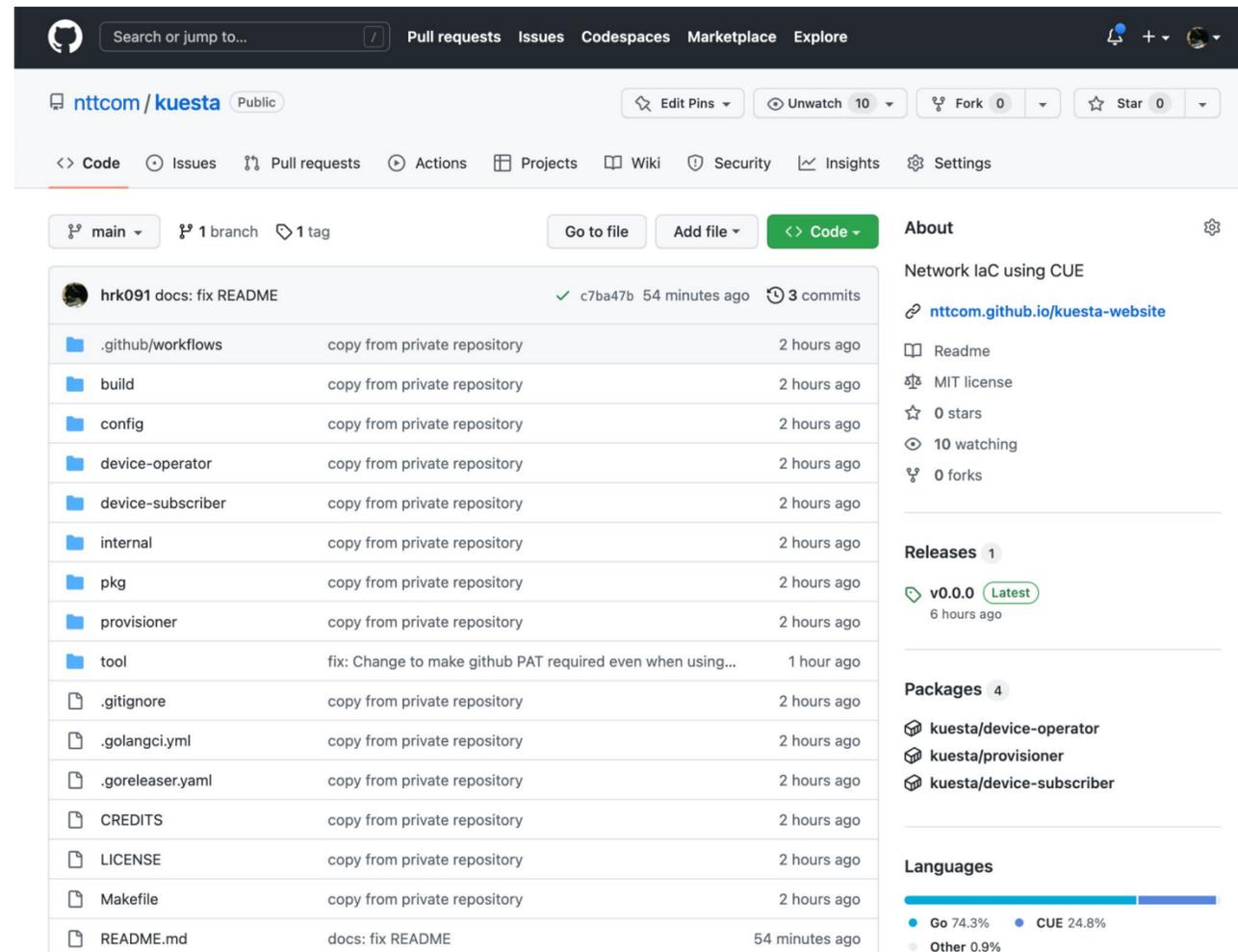
# Kuesta(ケスタ)とは

- K8s、GitOps等のクラウドネイティブ技術を活用した宣言的NWコントローラー
  - ルーター/スイッチ/伝送装置のコンフィグを抽象化してIaCで管理する
  - マルチベンダに対応(現リリースではgNMI/Openconfigに対応)
  - 機器のコンフィグ状態をGithub上で管理することで、設定変更の遷移確認・設定の戻しが容易

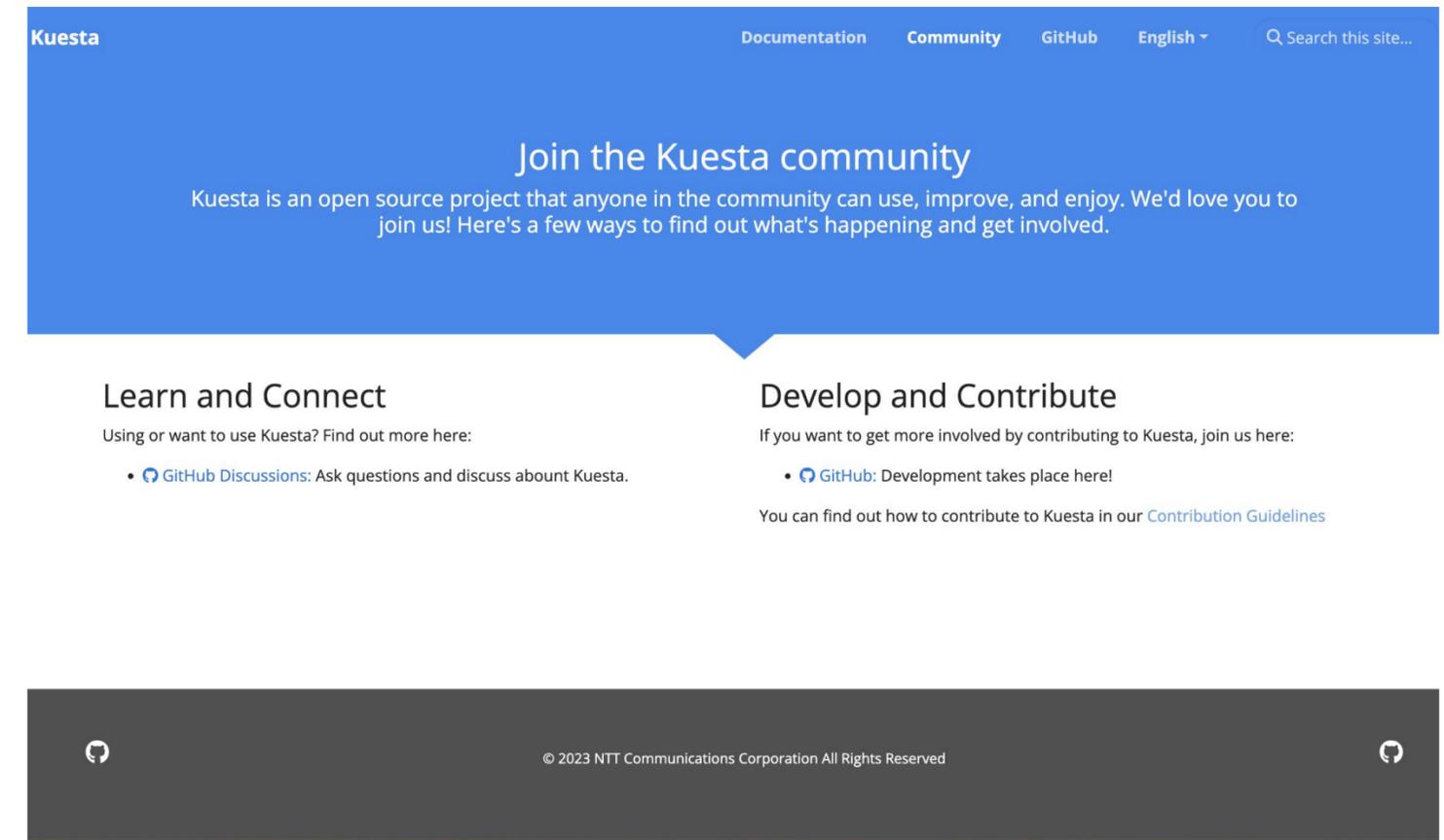


# 昨年度: ONIC2022, JANOG51で発表後、OSS化

- [ONIC2022発表](#)
- [JANOG51発表](#)
- [Github](https://github.com/nttcom/kuesta): <https://github.com/nttcom/kuesta>
- [コミュニティサイト](https://nttcom.github.io/kuesta-website/community): <https://nttcom.github.io/kuesta-website/community>



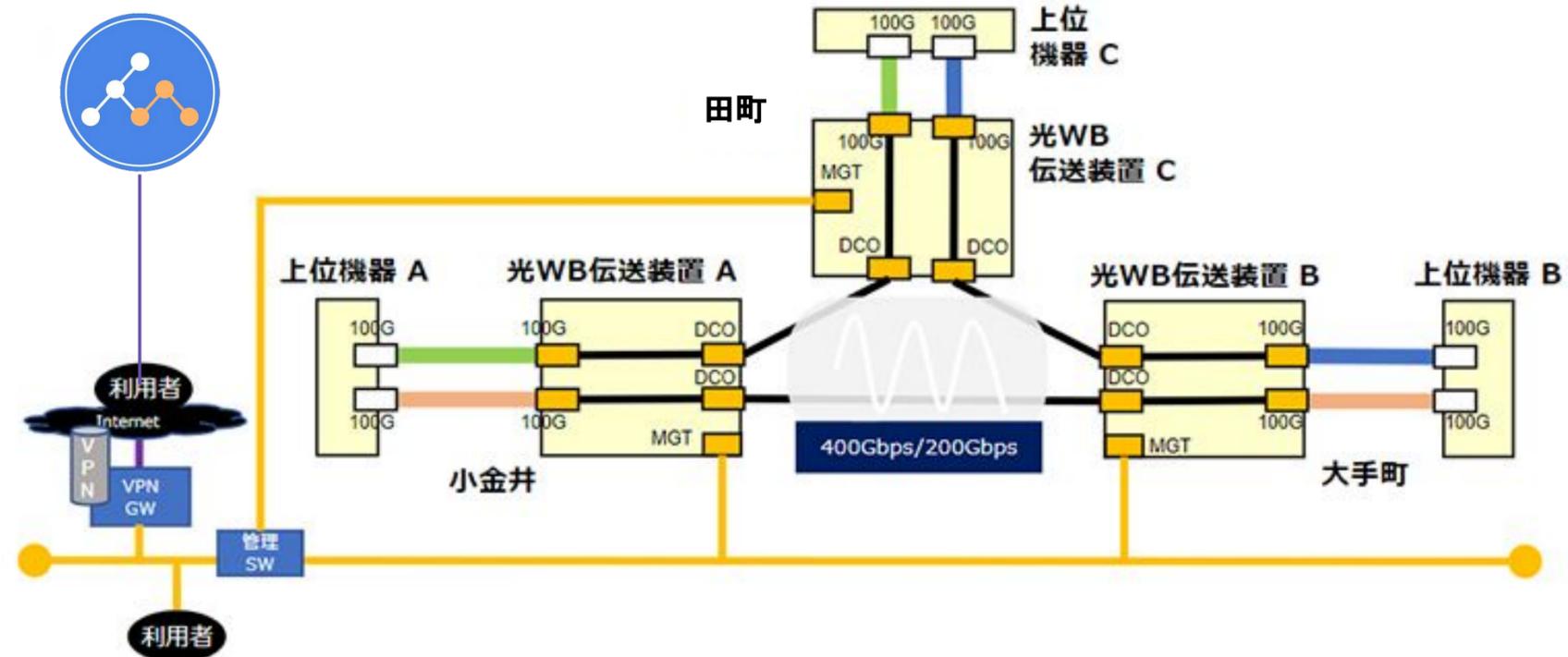
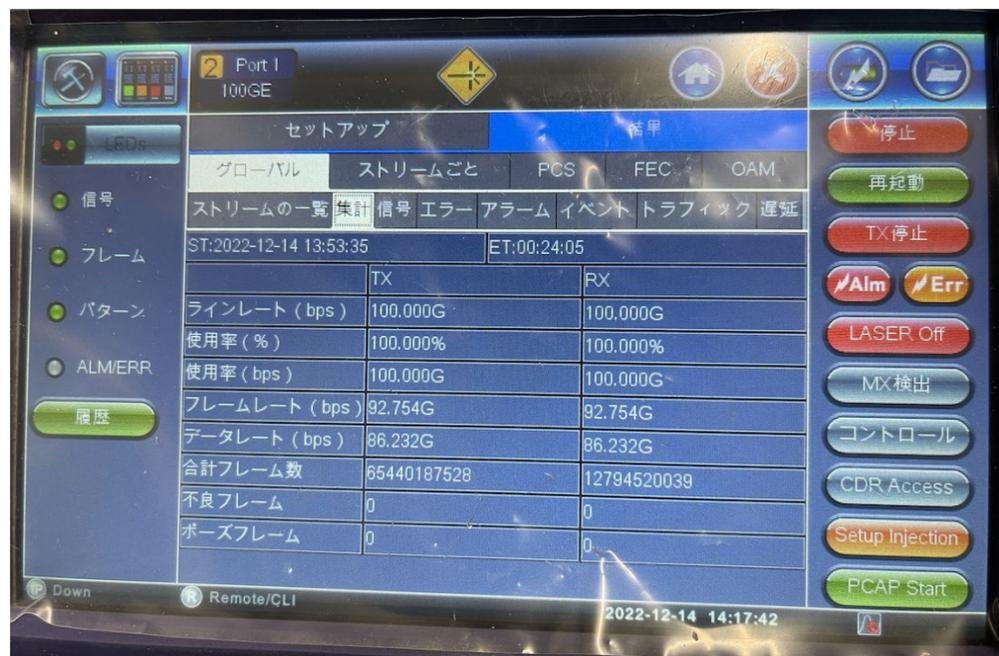
The screenshot shows the GitHub repository page for `nttcom/kuesta`. The repository is public and has 10 unwatchers, 0 forks, and 0 stars. The main branch is `main` with 1 branch and 1 tag. The repository contains several files and folders, including `.github/workflows`, `build`, `config`, `device-operator`, `device-subscriber`, `internal`, `pkg`, `provisioner`, `tool`, `.gitignore`, `.golangci.yml`, `.goreleaser.yml`, `CREDITS`, `LICENSE`, `Makefile`, and `README.md`. The repository is described as "Network IaC using CUE" and has a MIT license. The latest release is `v0.0.0`, published 6 hours ago. The repository also has 4 packages: `kuesta/device-operator`, `kuesta/provisioner`, and `kuesta/device-subscriber`. The languages used in the repository are Go (74.3%), CUE (24.8%), and Other (0.9%).



The screenshot shows the Kuesta website. The header includes the Kuesta logo, navigation links for Documentation, Community, GitHub, and English, and a search bar. The main content area features a blue banner with the text "Join the Kuesta community" and a sub-headline "Kuesta is an open source project that anyone in the community can use, improve, and enjoy. We'd love you to join us! Here's a few ways to find out what's happening and get involved." Below the banner, there are two columns of content: "Learn and Connect" and "Develop and Contribute". The "Learn and Connect" section includes a link to "GitHub Discussions" for asking questions and discussing about Kuesta. The "Develop and Contribute" section includes a link to "GitHub" for development and a link to "Contribution Guidelines" for finding out how to contribute to Kuesta. The footer of the website includes the GitHub logo and the copyright notice "© 2023 NTT Communications Corporation All Rights Reserved".

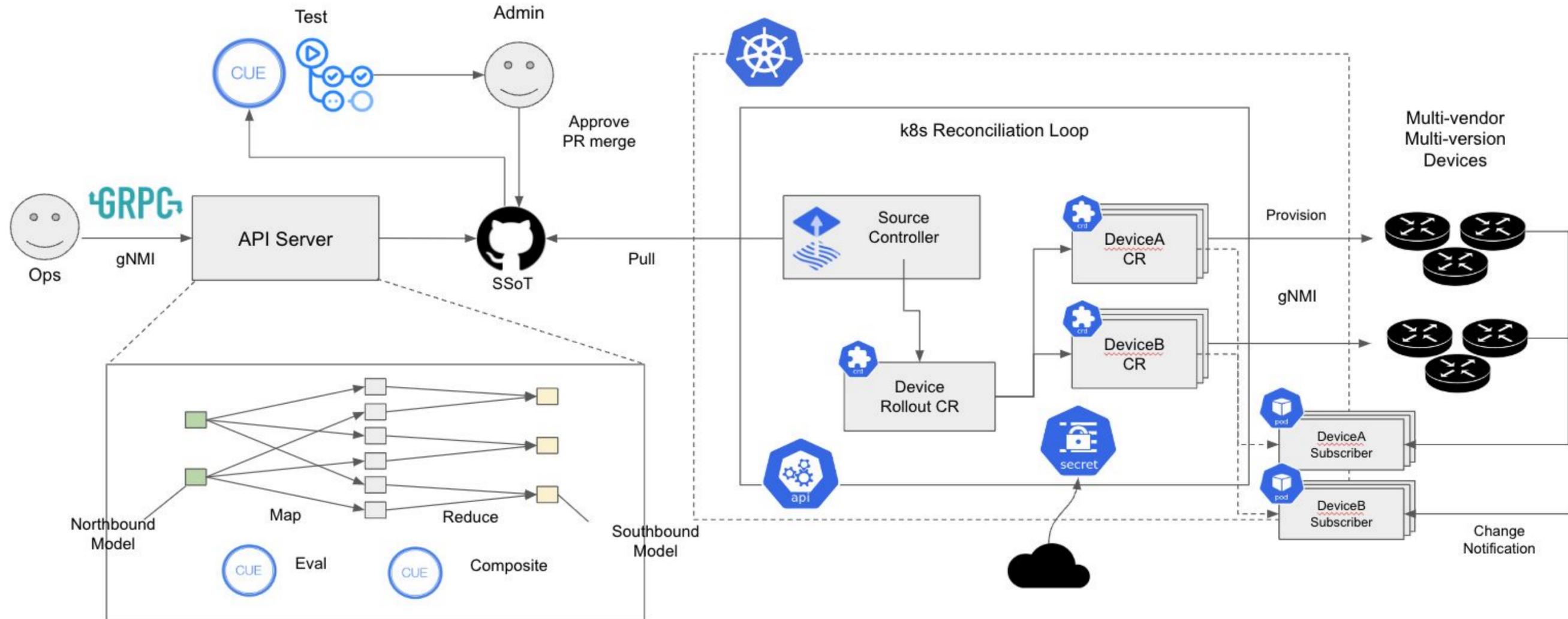
# 昨年度: フィールド環境での検証(伝送装置)

- [NICT様環境でのフィールド検証](#)
- Kuestaを用いてライン側を導通し、クライアント間を疎通させることに成功\*1



\*1: 本研究は、国立研究開発法人 情報通信研究機構(NICT)が運用する NICT 総合テストベッド「B5G高信頼仮想化環境」を用いて行われました。

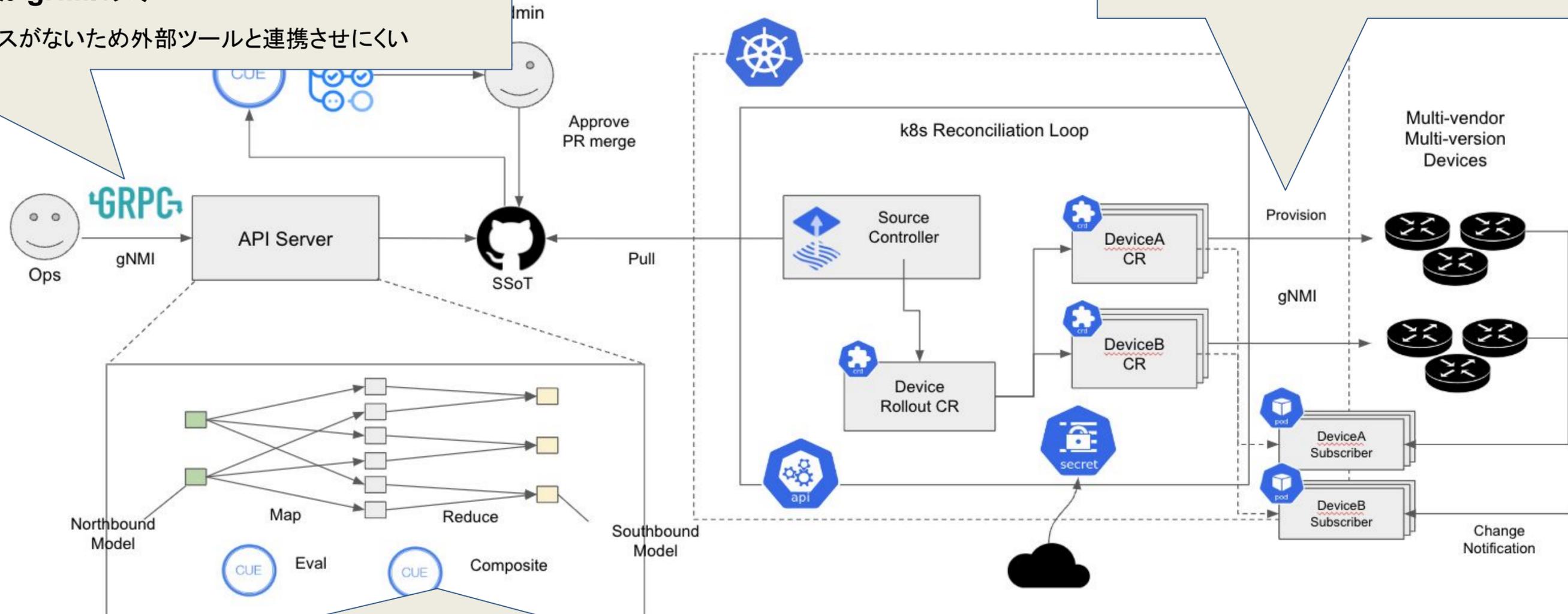
# 現在のKuestaアーキテクチャ



# 発表のフィードバック等から得た主な課題

**3. NorthboundがgNMIのみ**  
・Webインターフェースがないため外部ツールと連携させにくい

**1. SouthboundがgNMIのみ**  
・現状gNMI/openconfigに対応している機器は少ない。

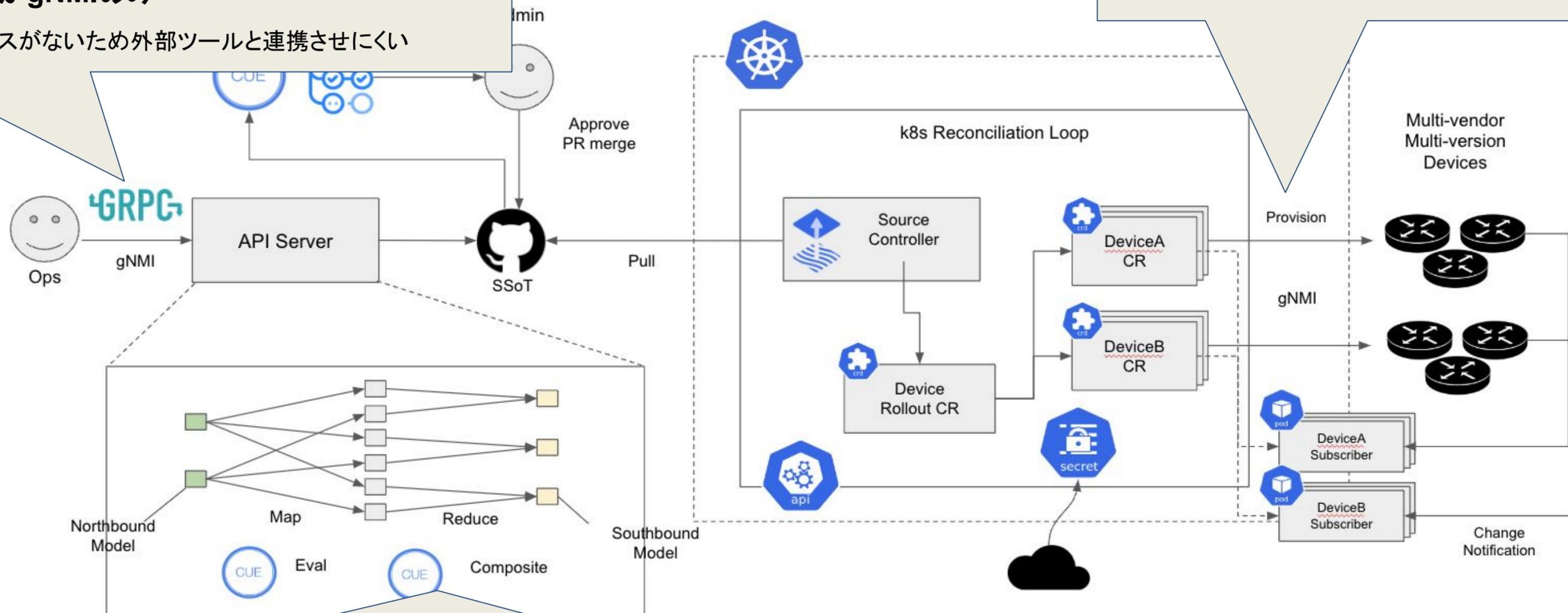


**2. 装置コンフィグ導出の複雑さ・パフォーマンス**  
・Cueは補完ツールやエラーログがまだ未整備な部分が多く習熟が大変  
・ファイル数や要素によってはパフォーマンスが問題

# 発表のフィードバック等から得た主な課題

**3. NorthboundがgNMIのみ**  
・Webインターフェースがないため外部ツールと連携させにくい

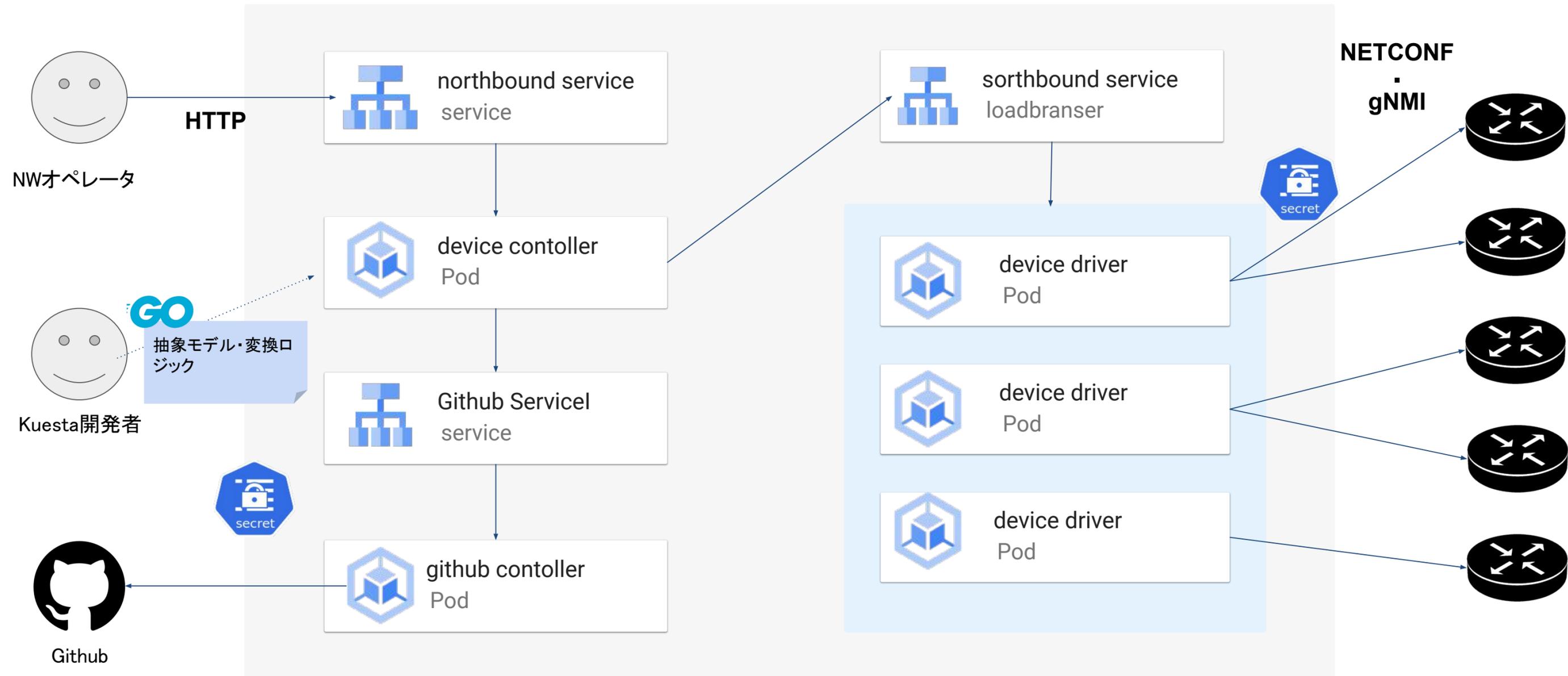
**1. SouthboundがgNMIのみ**  
・現状gNMI/openconfigに対応している機器は少ない。



**2. 装置コンフィグ導出の複雑さ・パフォーマンス**  
・Cueは補完ツールやエラーログがまだ未整備な部分が多く習熟が大変  
・ファイル数や要素によってはパフォーマンスが問題

1 ~ 3を解決するため、大幅なアーキテクチャ見直し

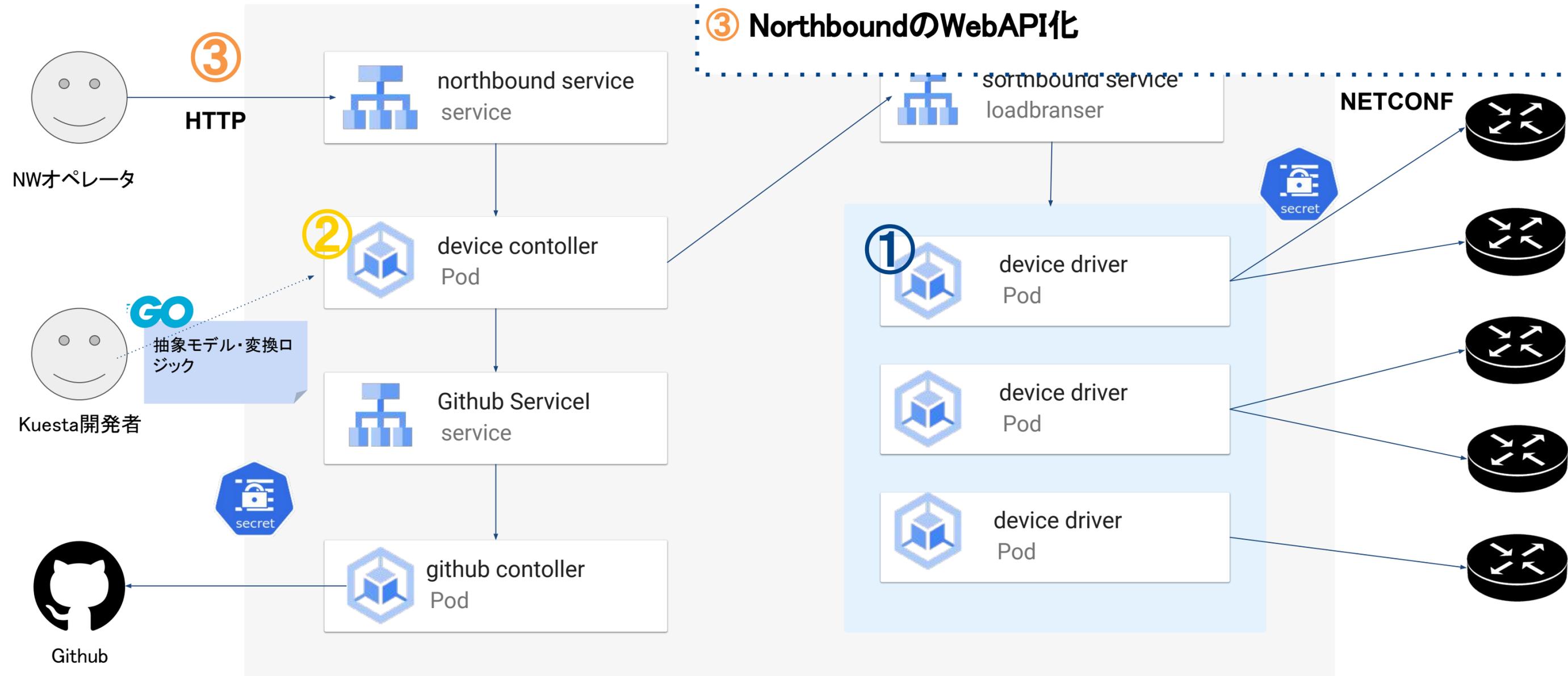
# 新バージョンのKuestaアーキテクチャ



# 新バージョンのKuestaアーキ

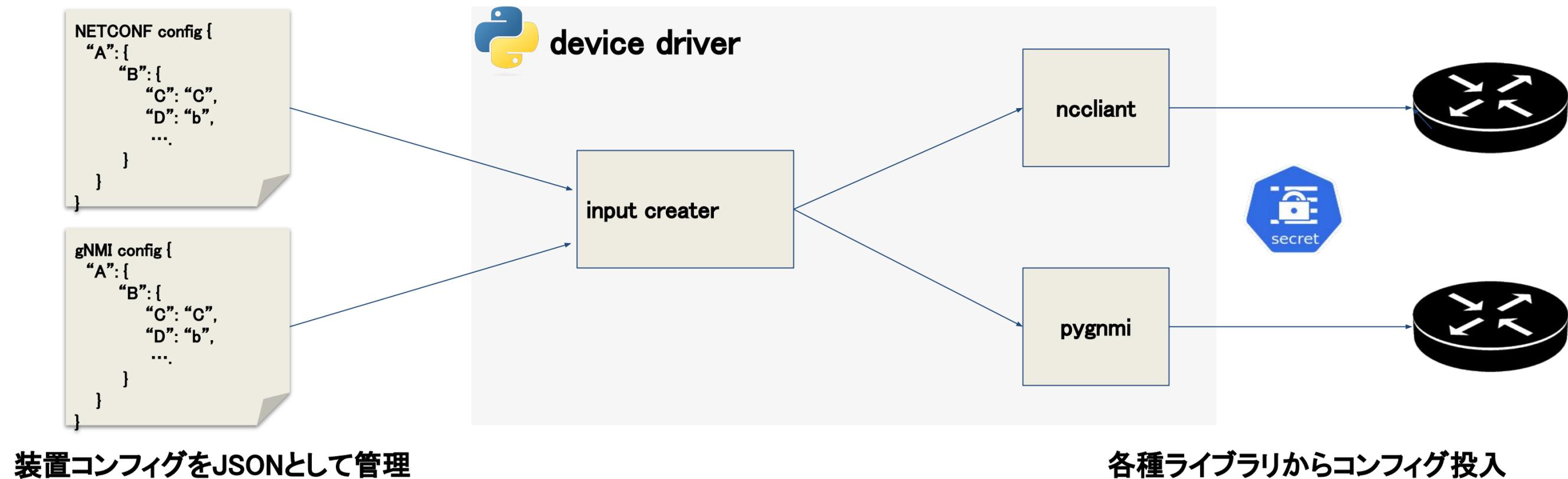
どのように現バージョンの課題を解決したか、3つにわけて説明する

- ① 汎用的・拡張性が高いdriverの開発
- ② Xpathによる装置コンフィグ管理
- ③ NorthboundのWebAPI化



# 汎用的・拡張性の高いdriverの開発

- 前実装では装置の種類毎にyangからdriverを作成しgNMIのライブラリで更新、投入を実施していた
- 新仕様では装置から取得したJSONをモデルとして管理するように変更、driver内部でライブラリへの入力を生成するコンポーネントとライブラリ利用部分に分けて実装した。
  - 装置の種類毎にdriverを用意する必要がなくなり、Kuesta開発者の負担が減った。
  - 装置コンフィグがJSONで取得できる & 対応プロトコルのPythonライブラリがあれば追加開発でNETCONF, gNMI以外の装置にも対応できるため、拡張性が向上

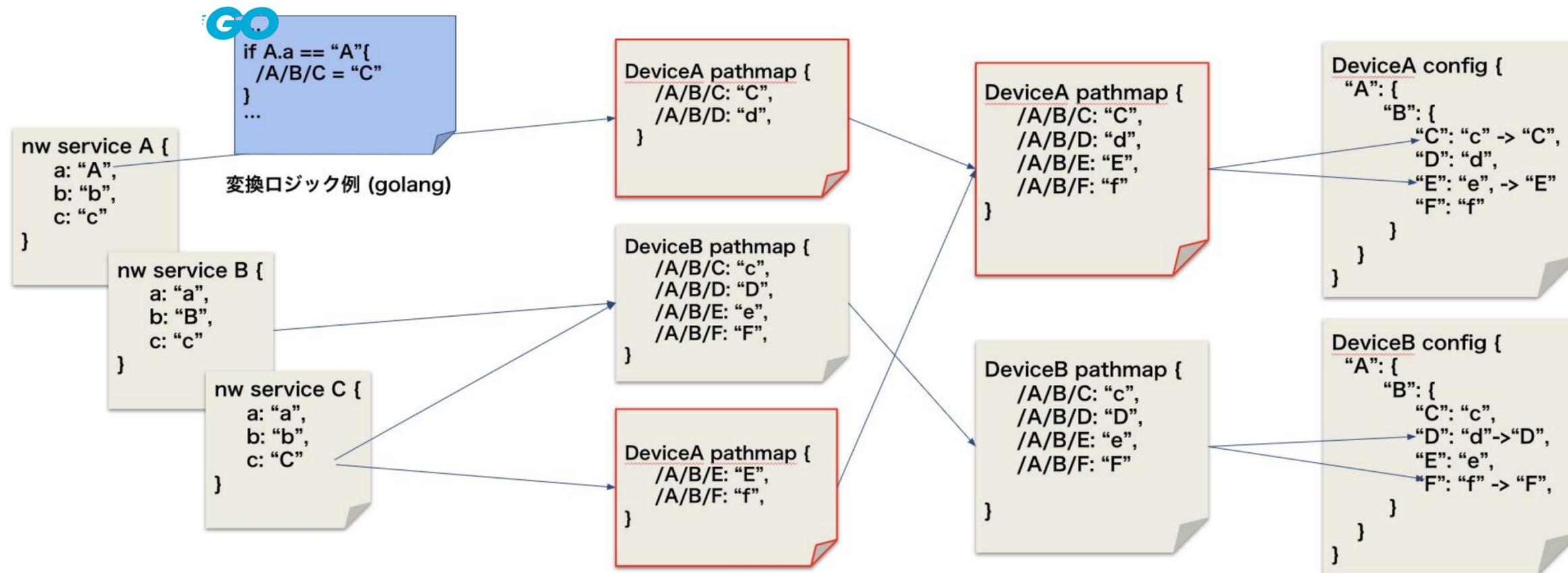


装置コンフィグをJSONとして管理

各種ライブラリからコンフィグ投入

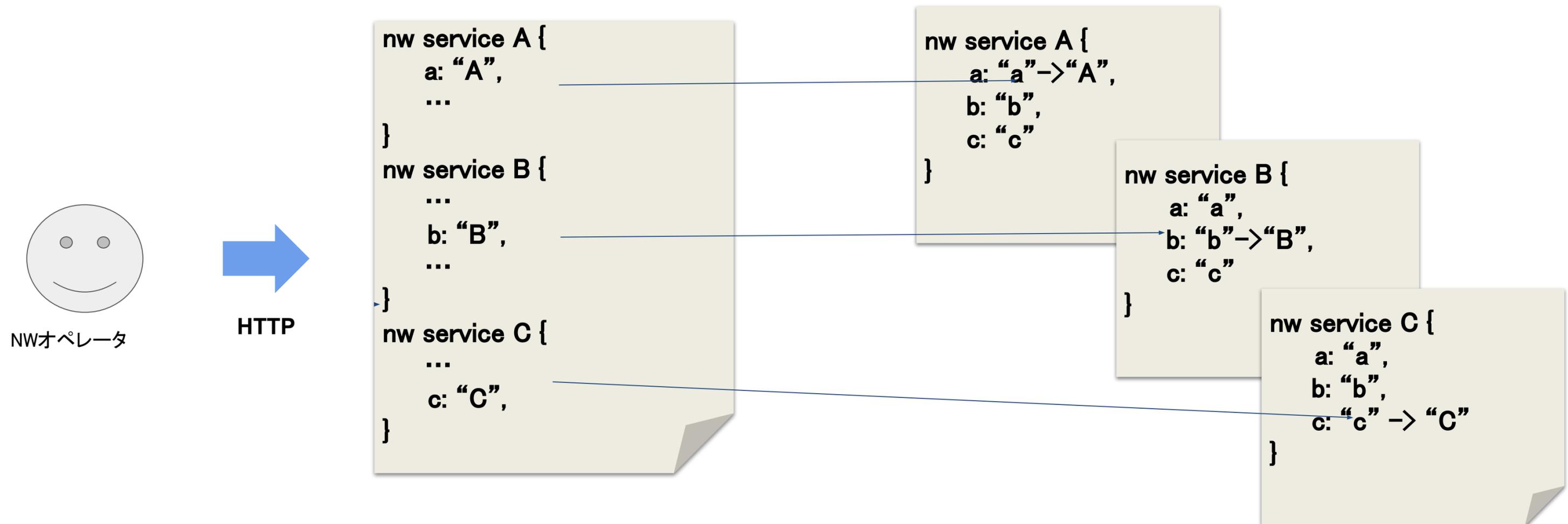
# Xpathによる装置コンフィグ導出

- 前仕様ではCueを利用して、投入するコンフィグを導出していた。
- 新仕様ではJsonからgolangによってpathmap(Xpathと値のセット)に変換し、装置コンフィグのjsonを更新することで導出するように変更
  - 今までCueで実装する必要があった変換ロジックをgolangで実装できるようになり、パフォーマンスが向上。
  - CueよりもGolangの方が広く普及しているため、技術者を集めやすくなった。

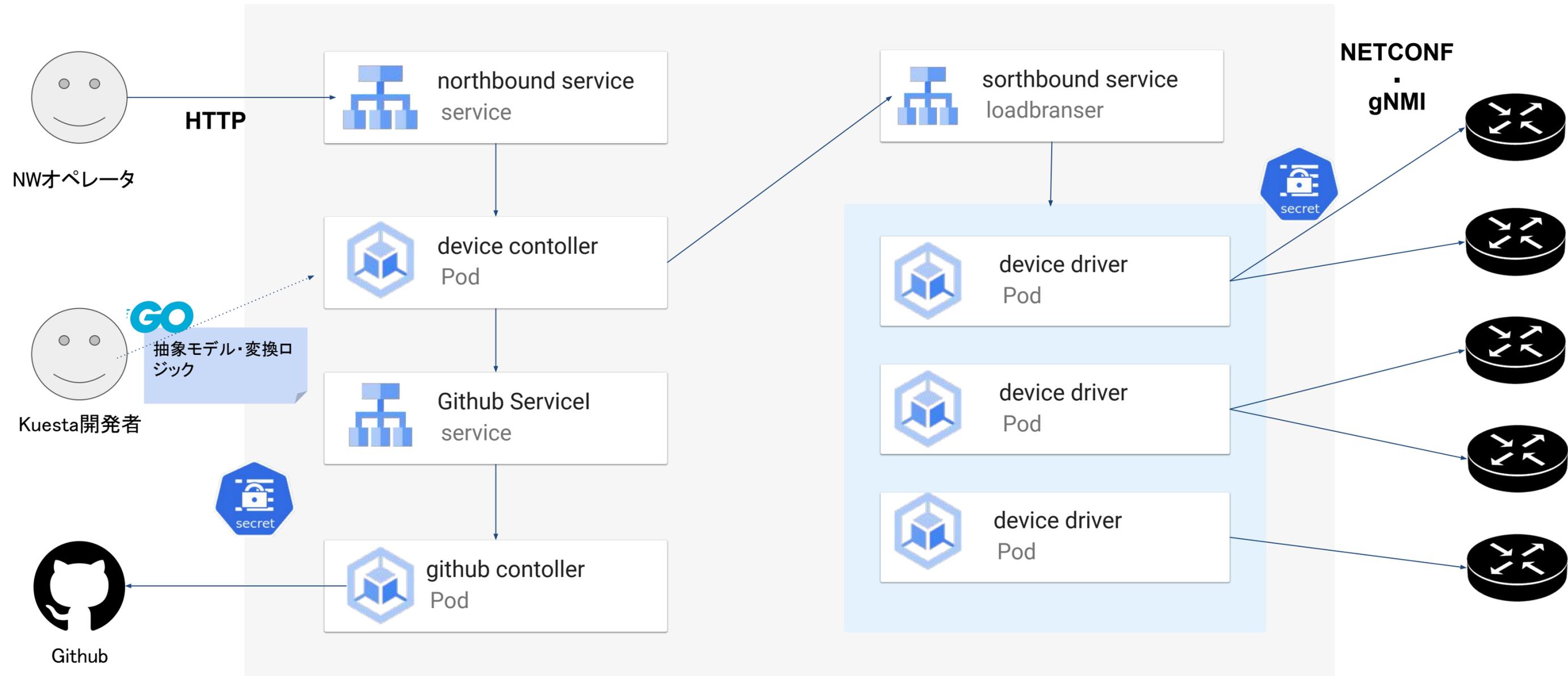


# NorthboundのWebAPI化

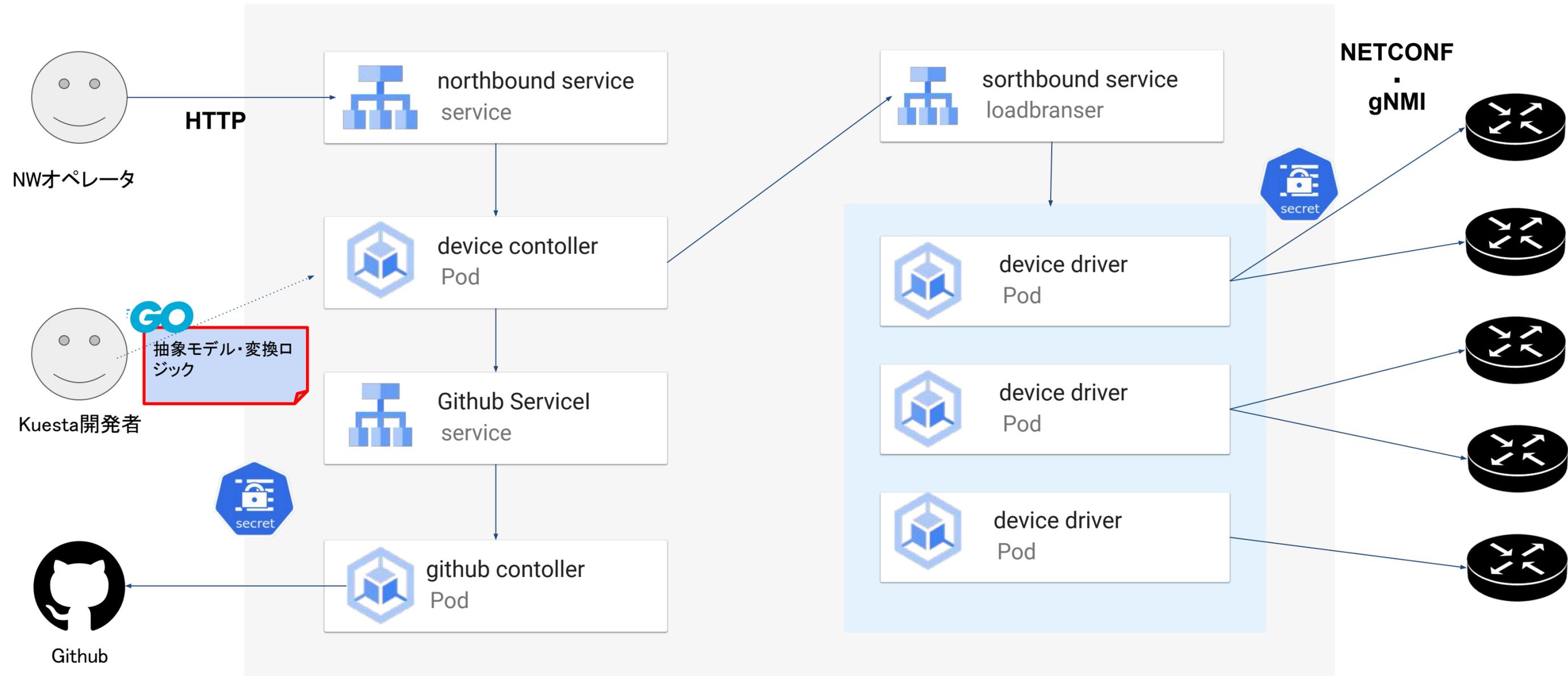
- 前仕様の実装ではgNMIを前提とした実装で、NorthboundはgNMIしか用意されていなかった
- 新仕様ではWebインターフェースを追加し抽象モデルをJSONでHTTP送信できるように実装を変更、また一度に複数の抽象モデルを変更できるようにした。
  - WebAPI化することで、以前と比較して外部ツールと連携しやすくなった



# 再掲: 新バージョンのKuestaアーキテクチャ



# 参考: Kuesta開発者で追加実装が必要な部分



# どんなNWコントローラーになるか？

- ネットワーク機器のコンフィグを抽象化したモデルによって操作できる
- マルチベンダ・複数デバイスの制御ができる(NETCONF or gNMI/openconfig対応)
- Git(+Github)によるトレーサビリティの高い装置コンフィグ情報の管理ができる
- K8sによるloadblancer、Secretによるセキュアなデバイス制御ができる



現時点である程度使えそう

# 今後の予定 & まとめ

- NWコントローラーへのクラウドネイティブ技術流用は結構いけそう。。。とはいえまだまだ課題はある

(1) 手動によるコンフィグ投入で管理データと実機のコンフィグが不整合になった場合の同期処理が未実装

(2) CLIインターフェースの対応

など

- **新バージョンは今年度末にリリース予定**

- 上の(1), (2)については現在開発チームで議論中(新バージョンに間に合うかも)

- 社内検証網でフィールドトライアルを実施し機能・品質をブラッシュアップ予定

- コミュニティと一緒に開発・検証してくださる方募集中です。是非Star、コメントよろしくお願ひ致します！！ 🙏

- OSSレポジトリ: <https://github.com/nttcom/kuesta>



**ご清聴ありがとうございました！！**



※背景画像使用イメージ