2019/08/02
OSC京都

ここまでできる!
設定ファイルからの
ネットワーク構成可視化

TIS株式会社
萩原 学

自己紹介

- 萩原 学 (HAGIWARA Manabu)
  - ネットワークの話が好きです

- 今日の資料はこのへんにあります

twitter.com/
github.com/
qiita.com/
speakerdeck.com/

corestate55

今日話したいこと

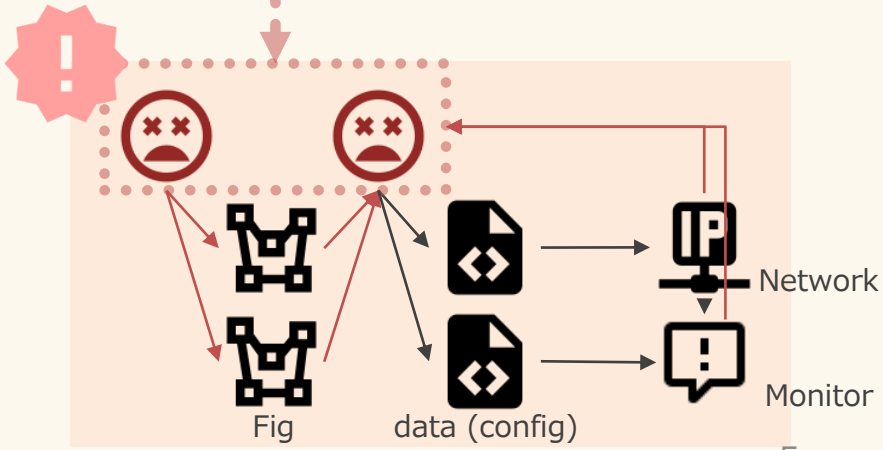
- 「**構成を可視化する**」話中心です
  - データソース=batfish の話はあまりしません
- どう?
  - イイ! / いまいち…
  - ウチで問題になってるこれにはどうだろう…
  - もっとこういうことはできないか…
- 何かやってみようかな

背景

課題感

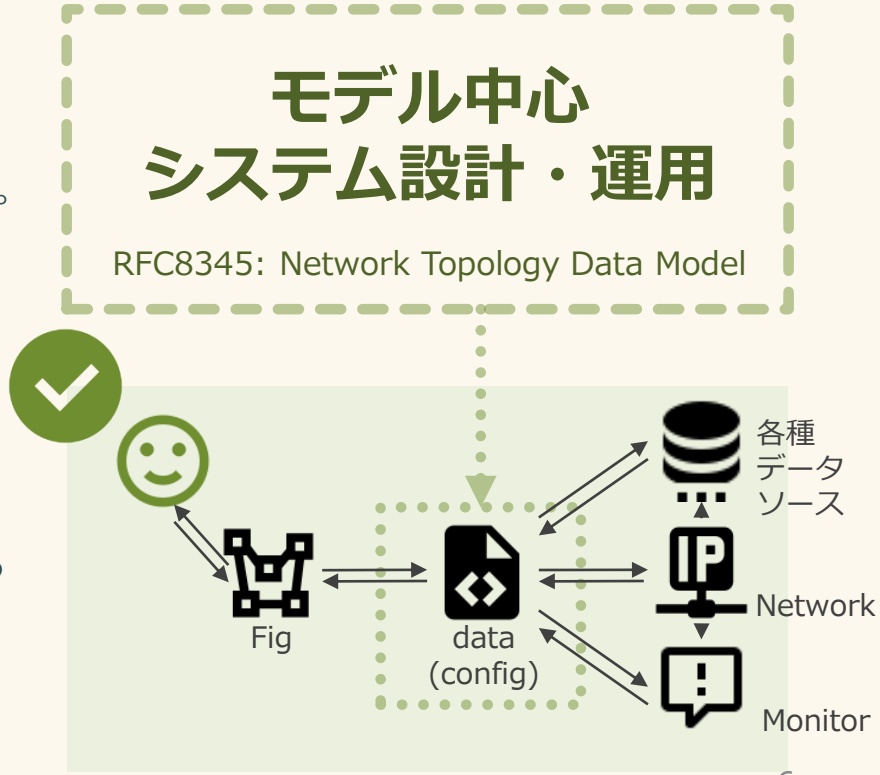
- システムの複雑化
  - **全体像をとらえるのが難しい**
  - どこで何をすべきか?
  - どこで何が起きているか?
- 結果…
  - 属人化
  - 多重レビュー
  - 初動対応の遅れ
- 構成要素単体ではなく、
  相互の関係性をとらえたい

図から情報を読み取る
図を基に判断する
ことがボトルネックに



目指す世界

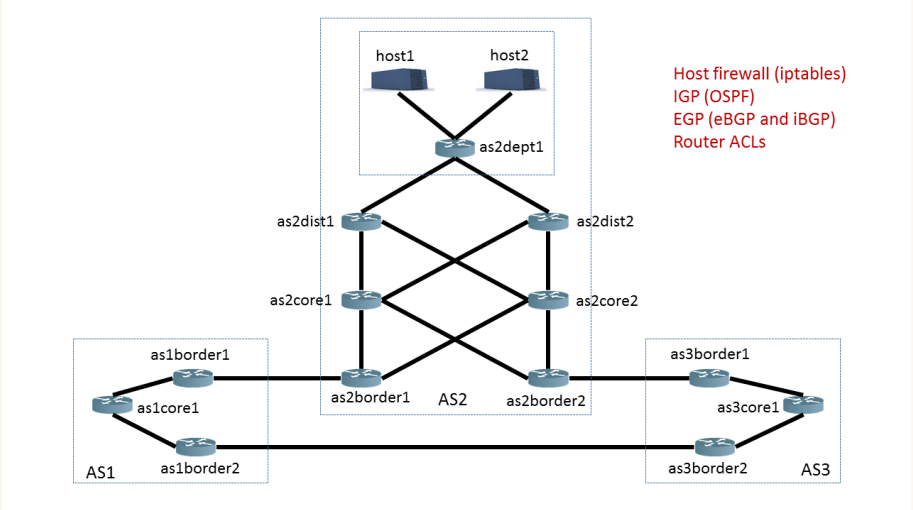
- **構成図 “職人芸” からの脱却**
  - 人による「図」読み書きの機械化
- **Read:** 全体像(構成)のキャッチアップ
  - 実際の環境情報のマッピング
  - モデル(設計情報)との照合…影響範囲調査
  - 状況に応じた情報量のコントロール
  - 変更差分の可視化
- **Write:** モデルベース設計
  - 書いた図(モデル)がそのままデプロイされる
  - 図(モデル)レベルでのテスト自動化
    (静的解析, verify, simulation)



DEMO

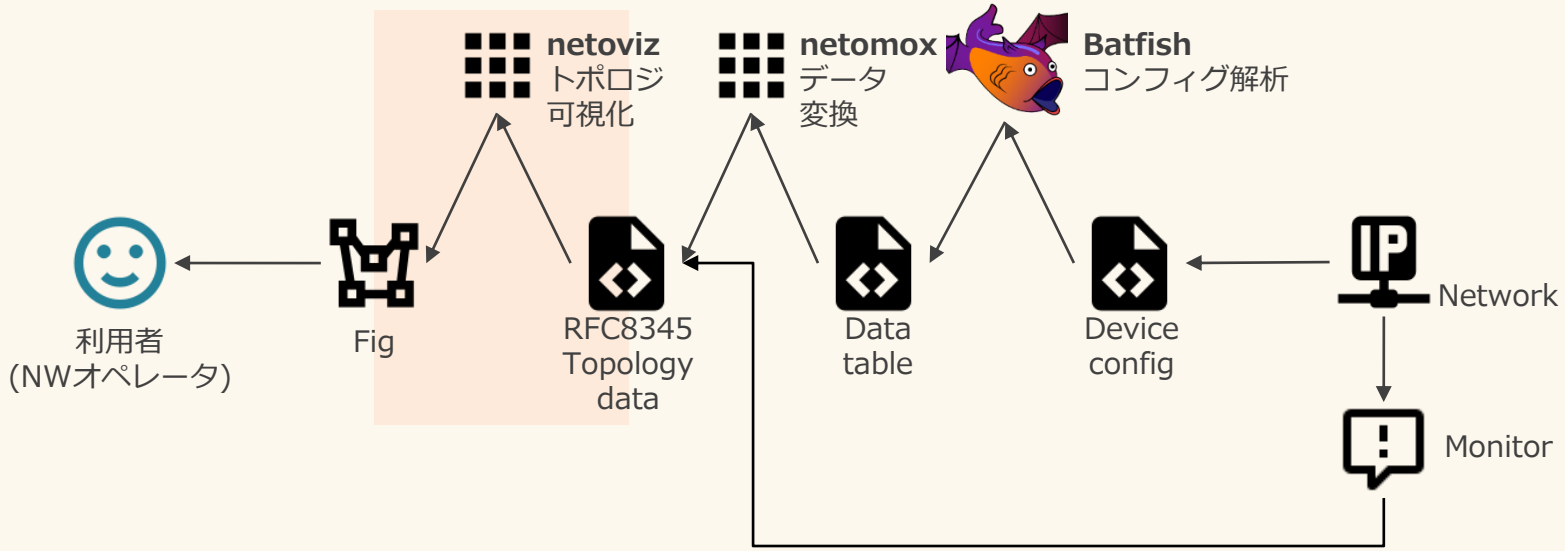
デモシナリオ

- batfish のチュートリアル用データを使用
  - pybatfish/jupyter_notebooks/networks/example at master · batfish/pybatfish https://github.com/batfish/pybatfish/tree/master/jupyter_notebooks/networks/example
- 元のNW構成図との対比をとりながら見てください

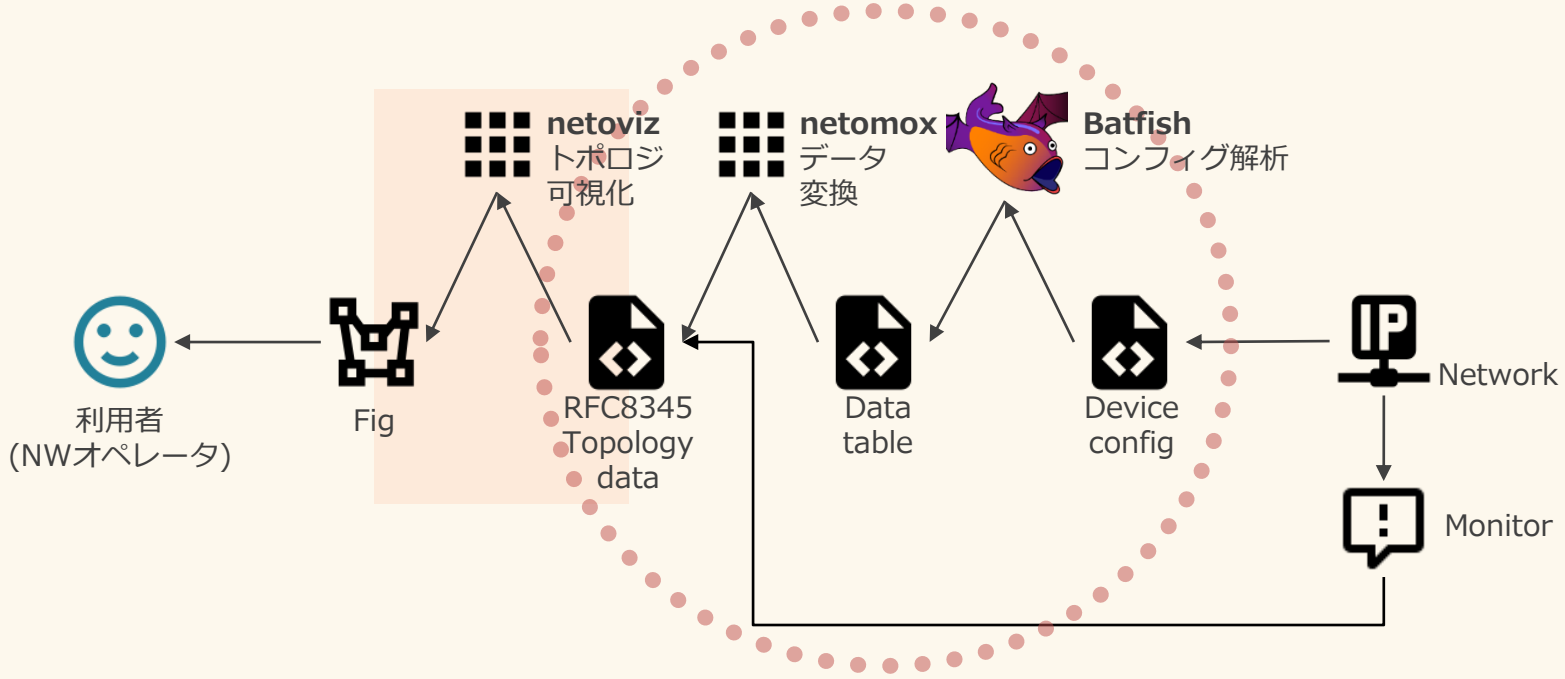


元図
https://github.com/batfish/pybatfish/blob/master/jupyter_notebooks/networks/example/example-network.png

データ処理フロー

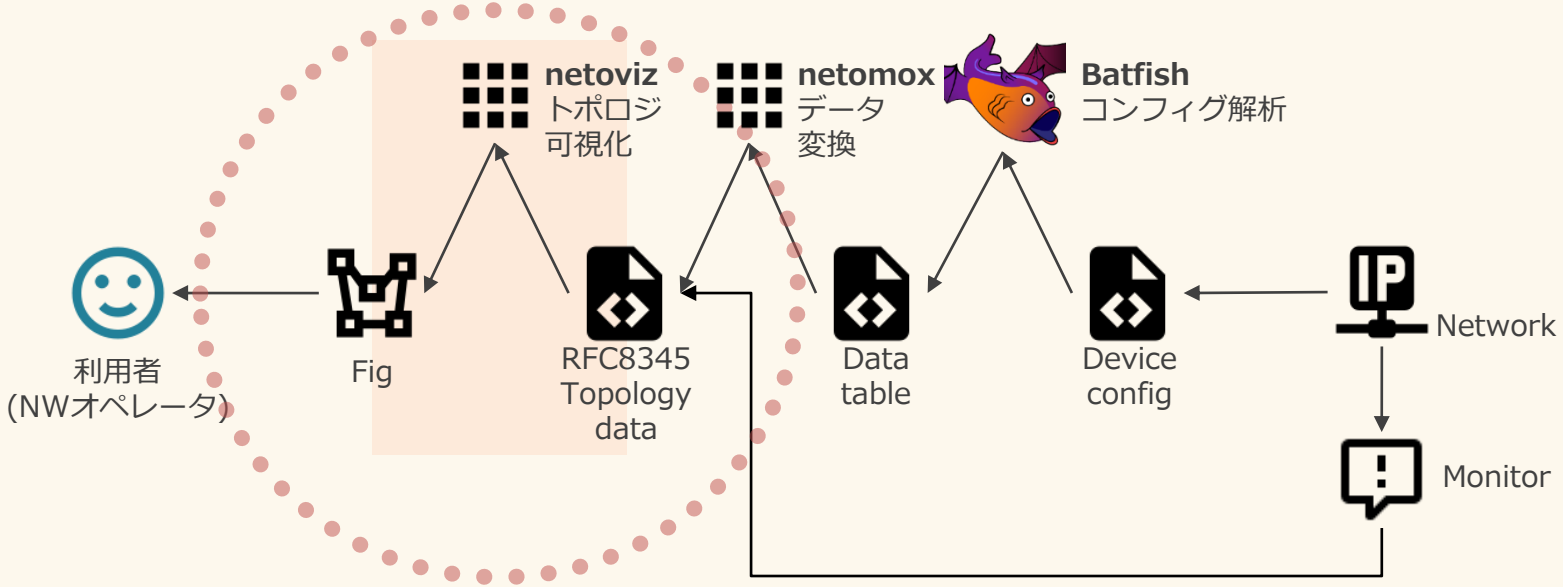

netoviz
トポロジ
可視化
netomox
データ
変換
Batfish
コンフィグ解析
利用者
(NWオペレータ)
Fig
RFC8345
Topology
data
Data
table
Device
config
Network
Monitor

データ処理フロー



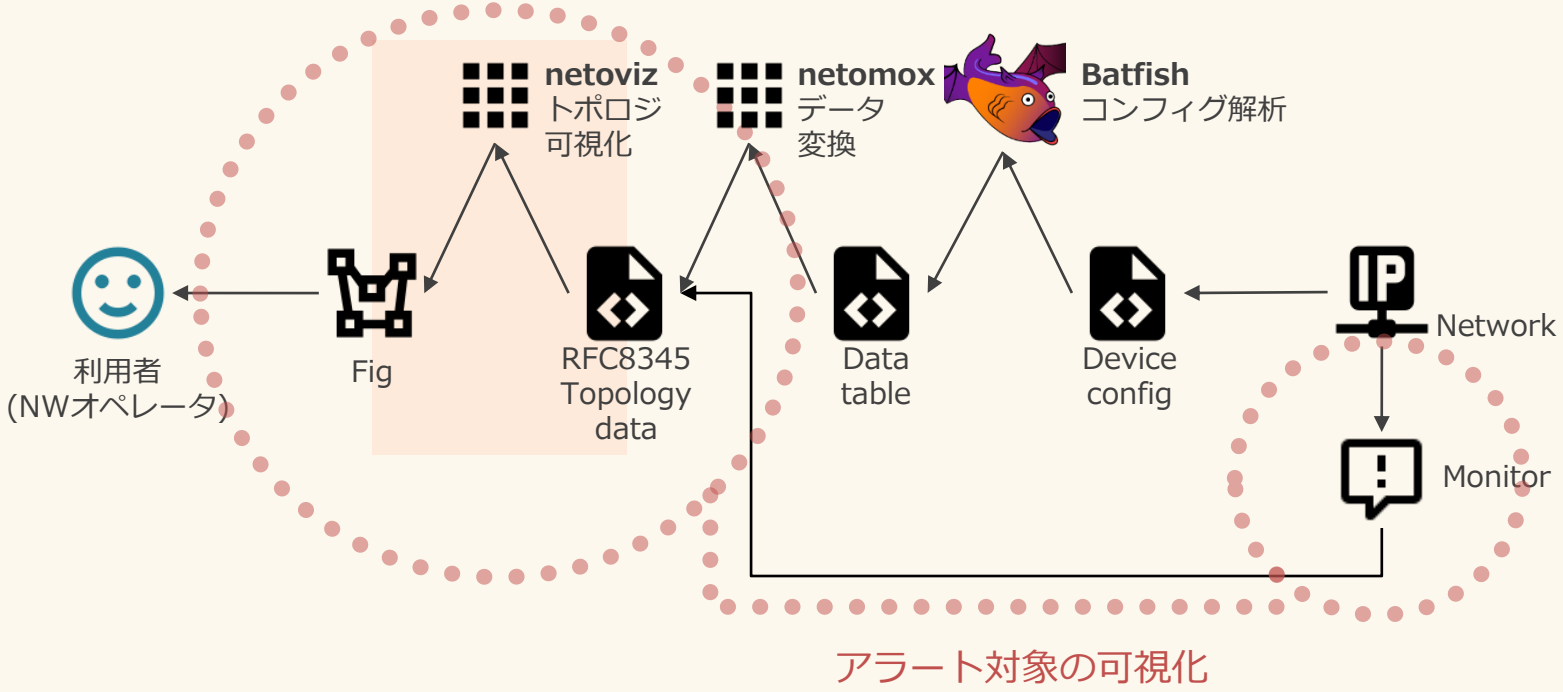
NW機器コンフィグをデータソースにした
トポロジデータの作成

データ処理フロー

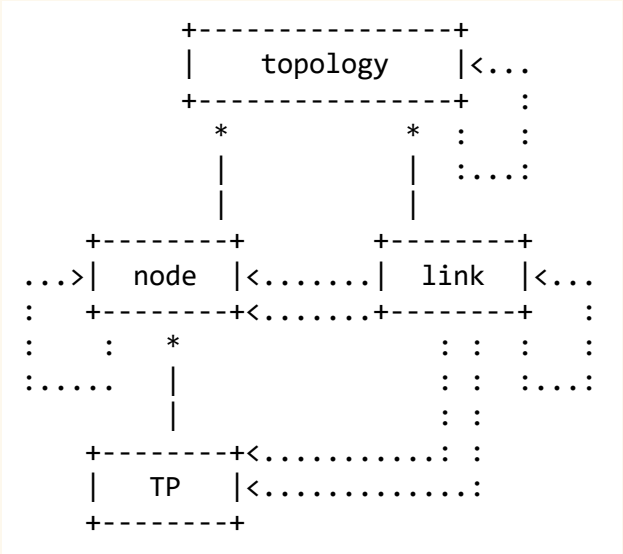


階層のあるトポロジデータの可視化

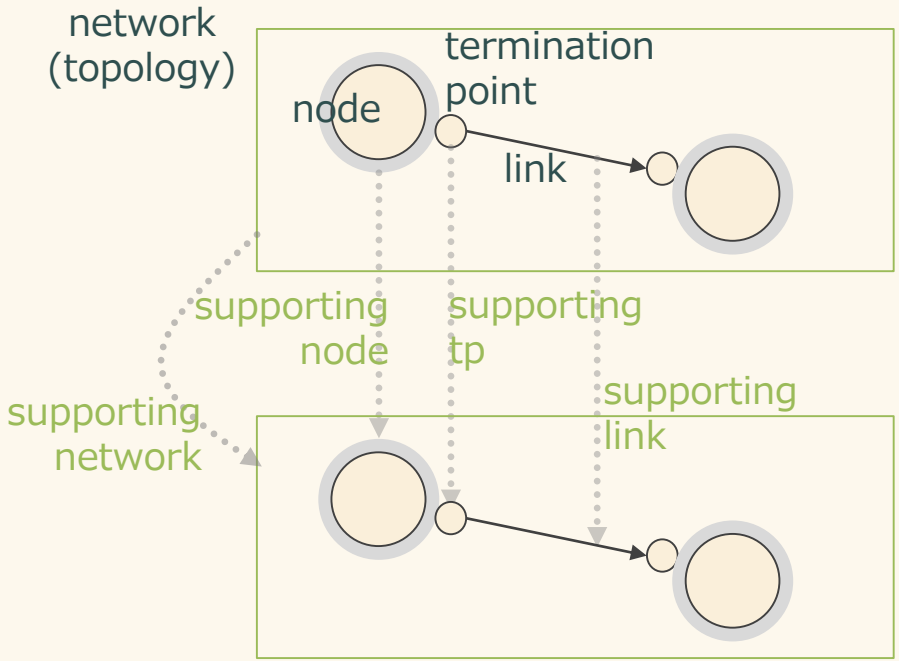
データ処理フロー


netoviz
トポロジ
可視化
netomox
データ
変換
Batfish
コンフィグ解析
利用者
(NWオペレータ)
Fig
RFC8345
Topology
data
Data
table
Device
config
Network
Monitor
アラート対象の可視化

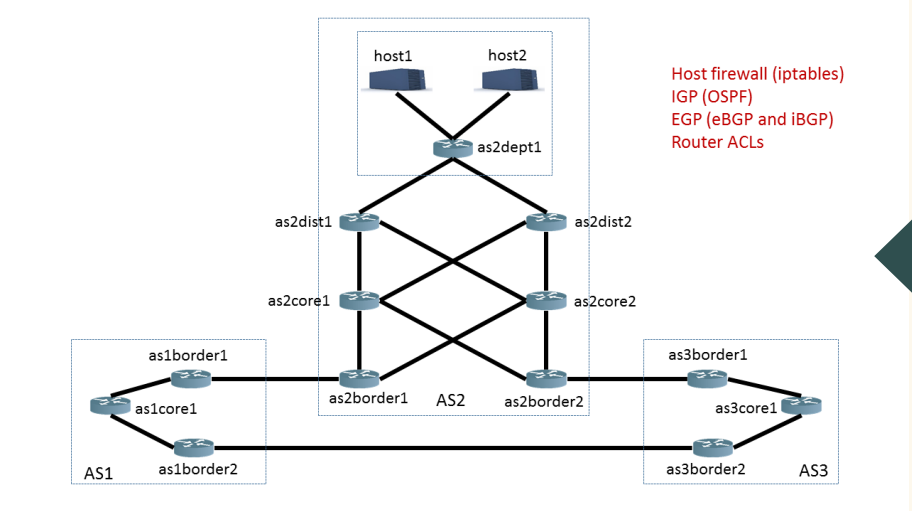
データモデルとグラフの対応



draft-medved-i2rs-topology-im-01 - An Information Model for Network Topologies
https://datatracker.ietf.org/doc/draft-medved-i2rs-topology-im/

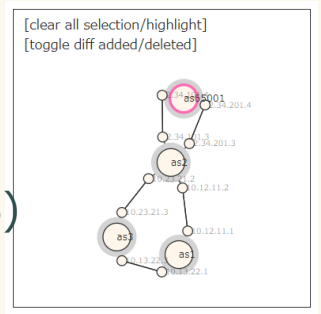


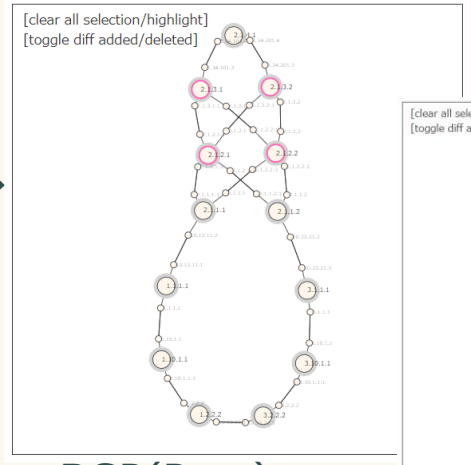
隣接関係 (bgp/layer3)



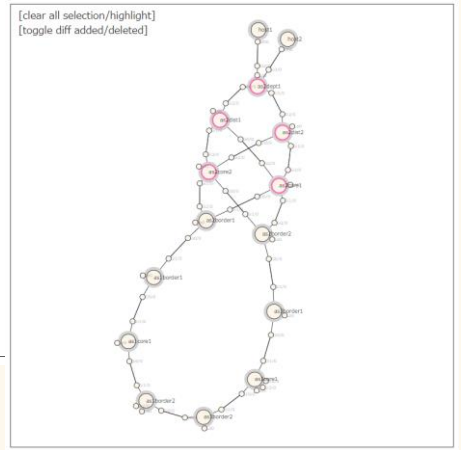
元図
https://github.com/batfish/pybatfish/blob/master/jupyter_notebooks/networks/example/example-network.png

BGP(AS)



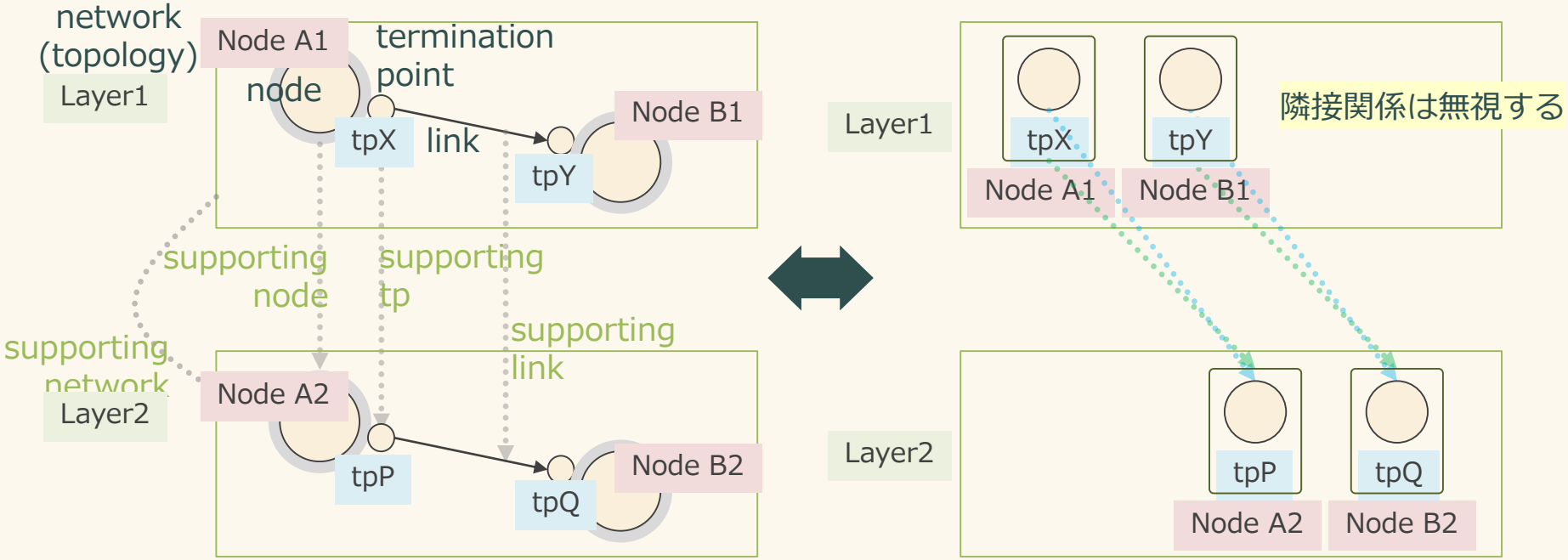


BGP(Proc)

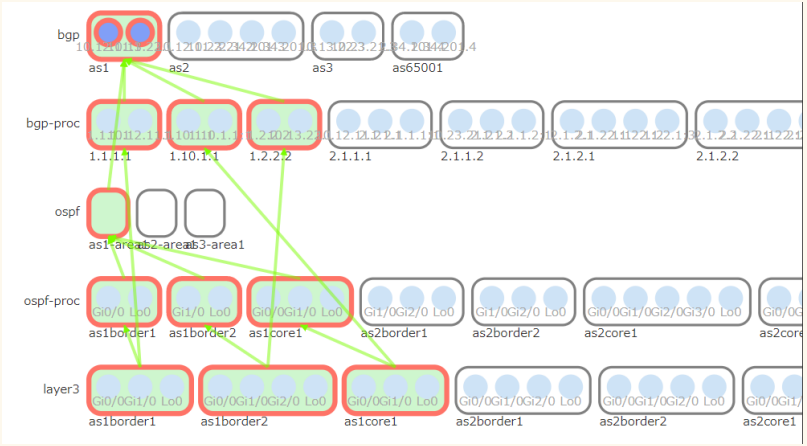


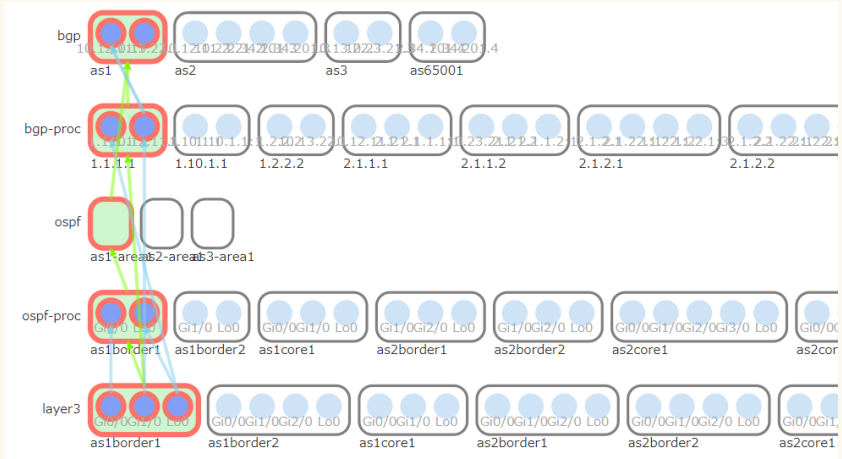
Layer3

見せ方 (dependency)



階層間の関係性

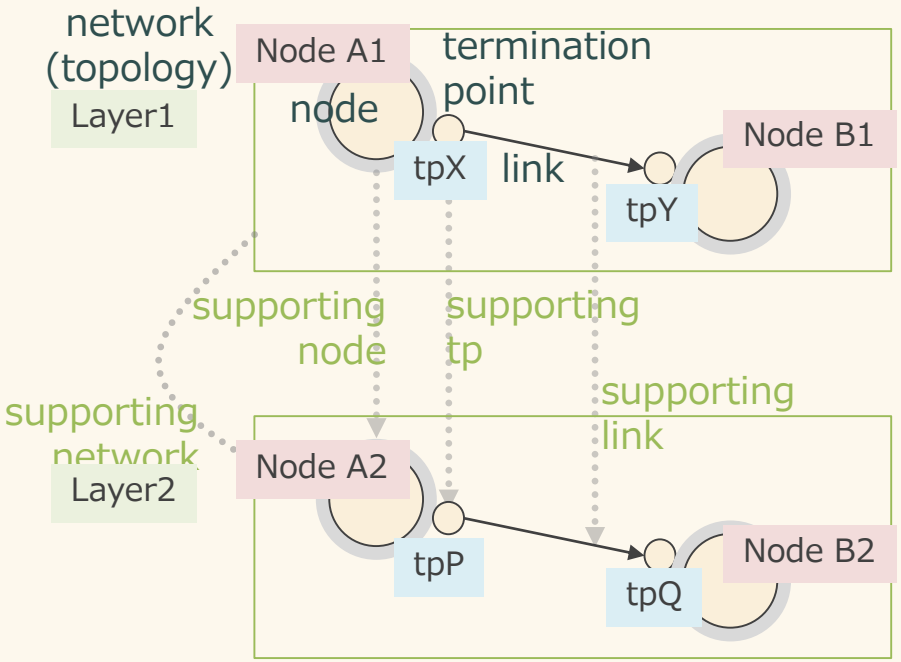




大きな構造(AS)が
何から構成されているか?

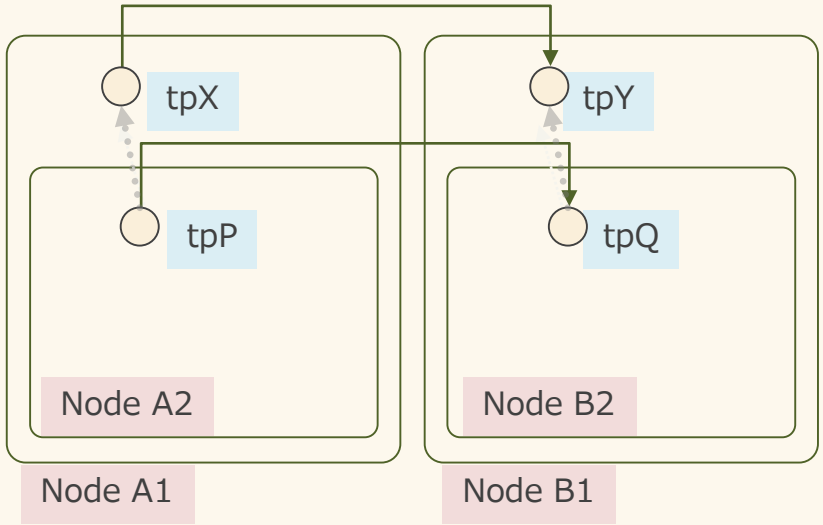
小さな構成要素(L3機器)が
何に影響を及ぼすか?

見せ方 (nested)

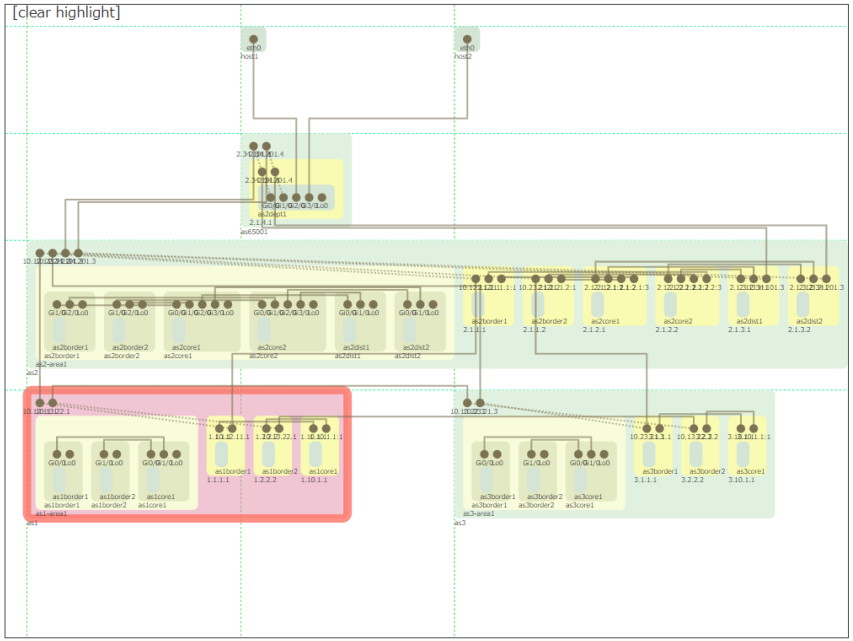


隣接関係と依存関係を合成する

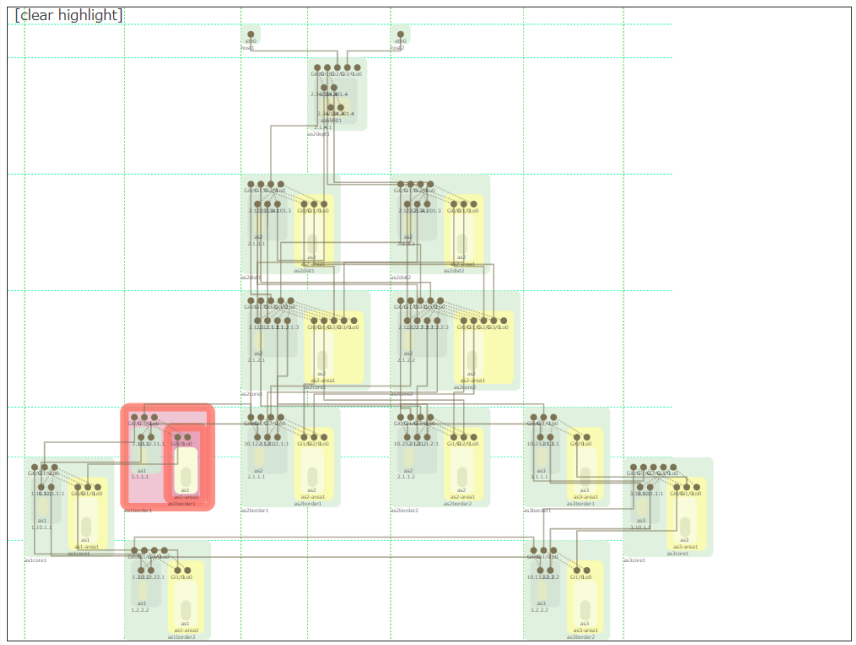
- ノードの依存関係→入れ子
- ポートの依存関係→矢印



隣接関係と階層間の関係性の合成

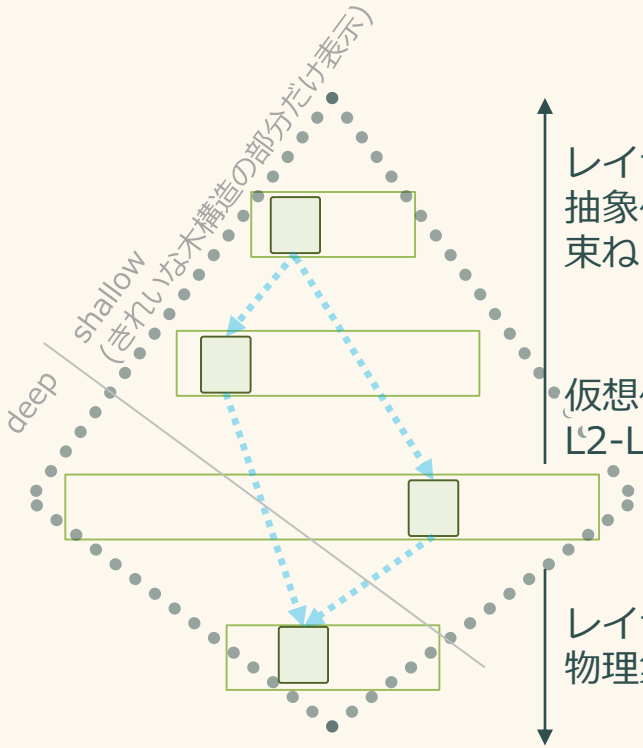


大きな構造(AS)が
何から構成されているか?



小さな構成要素(L3機器)が
何に影響を及ぼすか?

階層間の関係性はきれいな木構造にならない

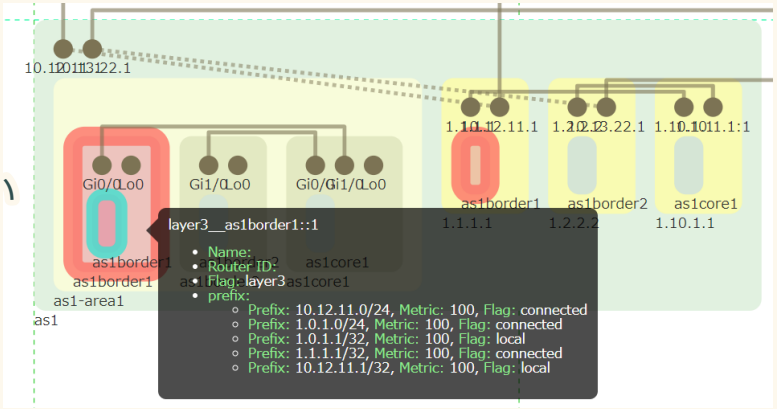


レイヤが高いところは
抽象化(グループ化)されて
束ねられる

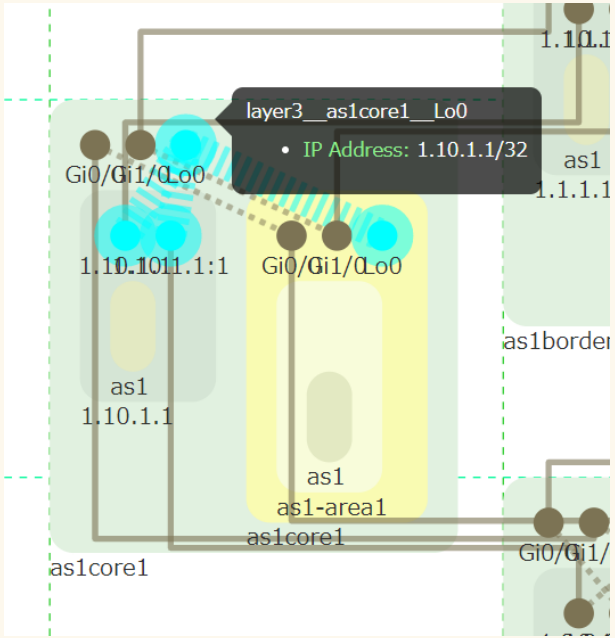
仮想化されるところ
L2-L3のオブジェクトが多い

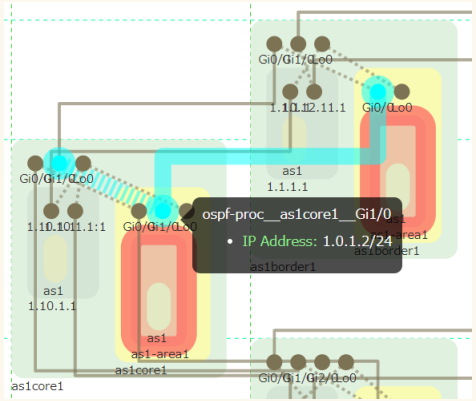
レイヤが低いところは
物理集約される

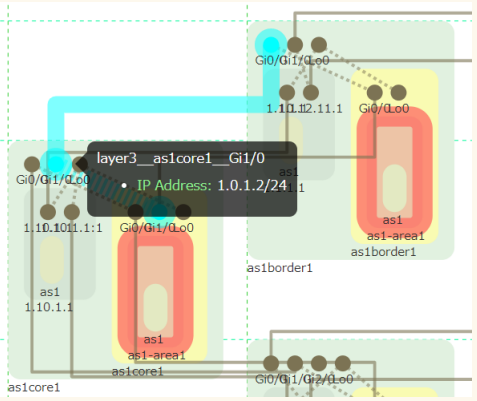
複数のプロセスが同一ノードにある
→ どう表現する?

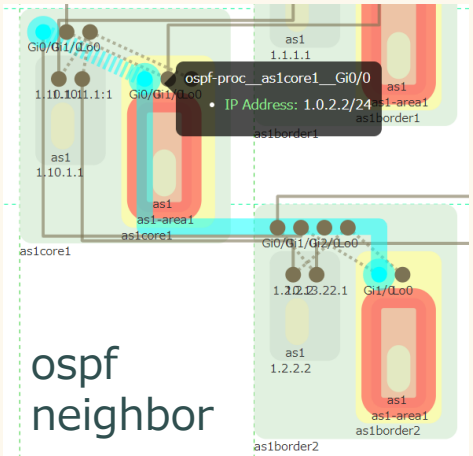


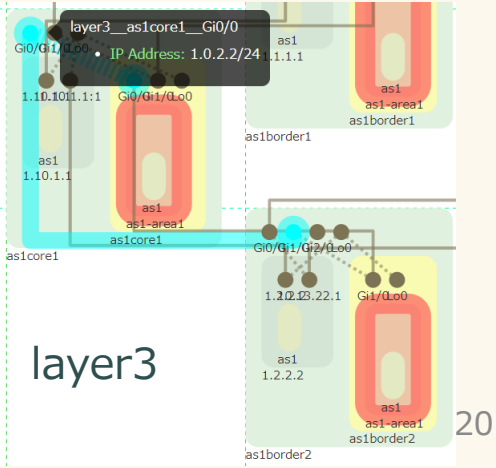
loopback & routing process (ospf)



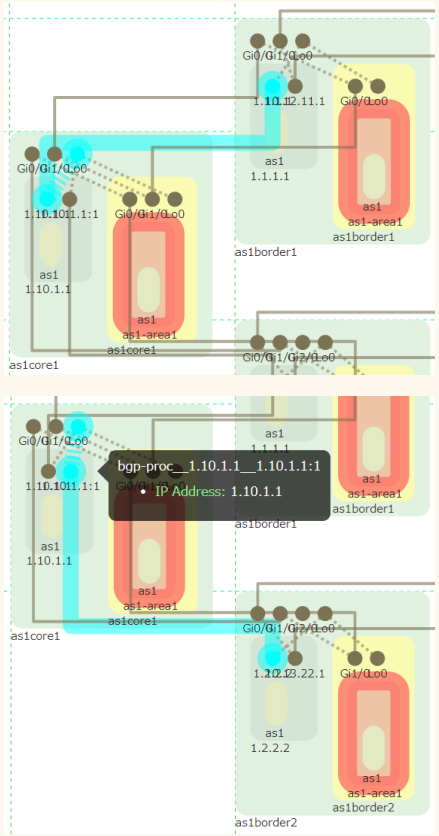








loopback & routing process (bgp)



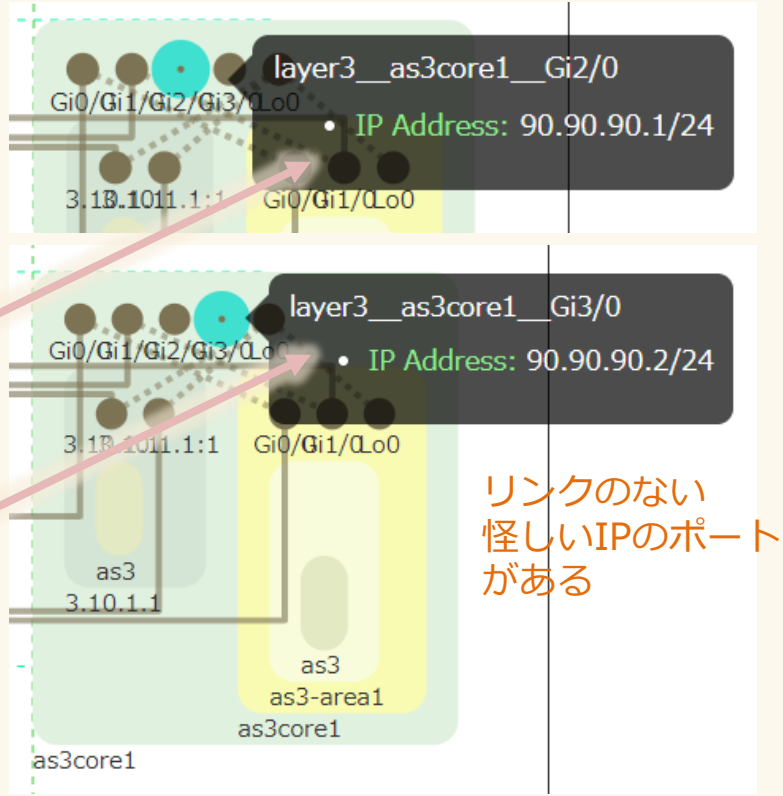

```
router bgp 1
 bgp router-id 1.10.1.1
 bgp log-neighbor-changes
 neighbor as1 peer-group
 neighbor as1 remote-as 1
 neighbor 1.1.1.1 peer-group as1
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 1.2.2.2 peer-group as1
 neighbor 1.2.2.2 update-source Loopback0
 !
 address-family ipv4
  bgp dampening
  bgp additional-paths select all
  bgp additional-paths send receive
  neighbor as1 send-community
  neighbor as1 route-reflector-client
  neighbor as1 advertise additional-pa
  neighbor 1.1.1.1 activate
  neighbor 1.2.2.2 activate
  maximum-paths 5
 exit-address-family
!
```

IGP(OSPF)の経路情報を加味しないと、BGP Proc の neighbor (loopback = router id) への到達性はわからない

https://github.com/batfish/pybatfish/blob/master/jupyter_notebooks/networks/example/configs/as1core1.cfg

モデル検査

```
hagiwara@dev01:~/nwmodel/netomox-examples$ bundle
exec netomox check public/model/bf_trial.json | jq
'.[] | select(.checkup == "link reference count of
terminal-point").messages[] |
select(.path|test("__Lo0")|not)'
{
  "severity": "warn",
  "path": "layer3__as1border2__Gi2/0",
  "message": "irregular ref_count:0"
}
{
  "severity": "warn",
  "path": "layer3__as3core1__Gi2/0",
  "message": "irregular ref_count:0"
}
{
  "severity": "warn",
  "path": "layer3__as3core1__Gi3/0",
  "message": "irregular ref_count:0"
}
hagiwara@dev01:~/nwmodel/netomox-examples$
```



まとめ

できたこと

- batfishをつかうと config から
  いろんなプロトコルのトポロジ情報が取れる

- RFC8345データモデルを使うと複数のトポロジ
  (レイヤ)の関係性を定義できる

- 関係性の定義された複数のトポロジの「見せ方」を
  いくつか試してみた

課題

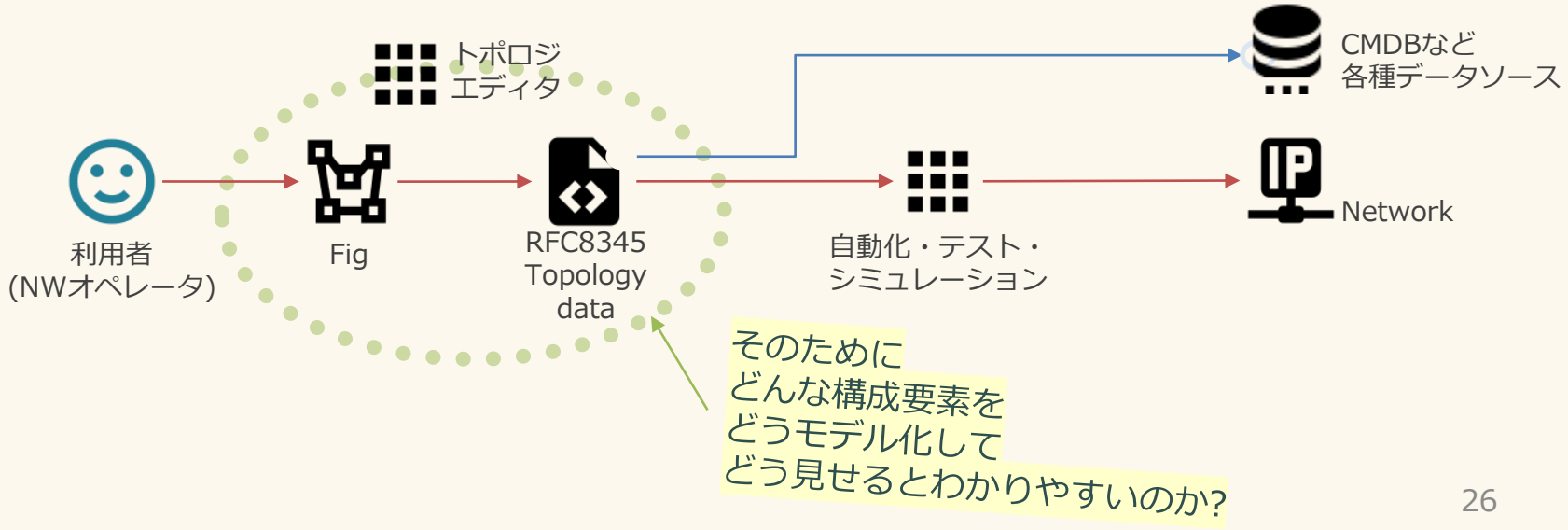
- 今回使った batfish tutorial のトポロジは
  シンプルでわかりやすい
  - All L3 (VLAN/VRF他 仮想化技術系がない)
  - 複数の要素を束ねるような冗長化機能を使っていない

- 運用上、把握が難しい構成要素をどう扱う(表現する)か?
  - 仮想化, 冗長化, Overlay, 動的に変わる状態など
  - 設計情報など “L8” 方向の関係性?
  - それらのモデル化 + 可視化(見せ方)
    - 脳内マッピングやめたい

考えたいこと

Write: モデル中心の設計・設定・構成変更
構築・本番作業前の事前チェック・テスト・シミュレーションへの応用

「図を書いたらその通りのシステムができる」へ



参照

- TISとフィックスポイント、「標準トポロジモデルを応用したネットワーク構成の可視化に関する研究」を共同で開始
  https://www.tis.co.jp/news/2018/tis_news/20181017_1.html
- 「ネットワーク図」のモデル化とモデルを起点にした自動化の可能性 / onic2018
  https://speakerdeck.com/corestate55/onic2018

今回発表した内容の解説ブログあります

- Batfish を使ってネットワーク構成を可視化してみよう (1) - Qiita
  https://qiita.com/corestate55/items/8a39af553785fd77c20a
- (1)-(3)まであります

ツール

- **netomox**: Network Topology Modeling Toolbox
  https://github.com/corestate55/netomox
  - データ定義DSL
  - Topology Data (JSON) の CLI diff

- **netoviz**: Network Topology Visualizer
  https://github.com/corestate55/netoviz
  - https://netoviz.herokuapp.com/ (demo)
  - Topology Data (JSON) Visualizer