

ネットワークの構成図を 継続的にアップデートし続ける方法

田島 照久

NTTコミュニケーションズ イノベーションセンター

teruhisa.tajima@ntt.com, slack janog WSにも参加しています

JANOG47 LT (HAKATA; online) 2021/01/28

自己紹介

■ 田島 照久 (たじま てるひさ)

■ 略歴

- DCNW構築やNWのテスト自動化の開発@SIer
- 2019/1~ NTTCom検証網の企画・設計・構築・運用

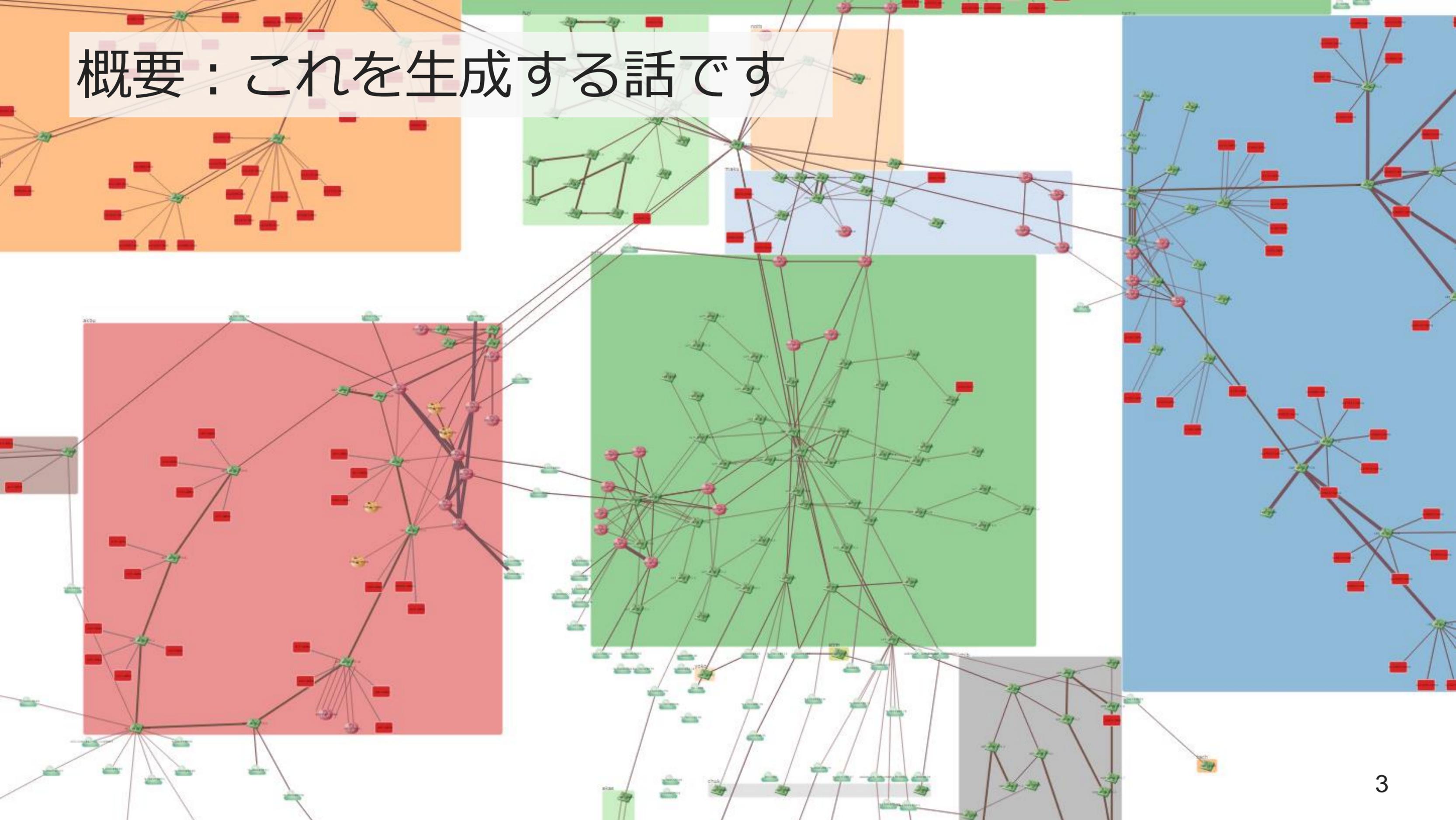
■ 興味

- Automation, Programmability

■ JANOGとのかかわり

- JANOG41(広島) ネットワーク運用自動化BoF

概要：これを生成する話です



構成図のメンテコストが高い問題

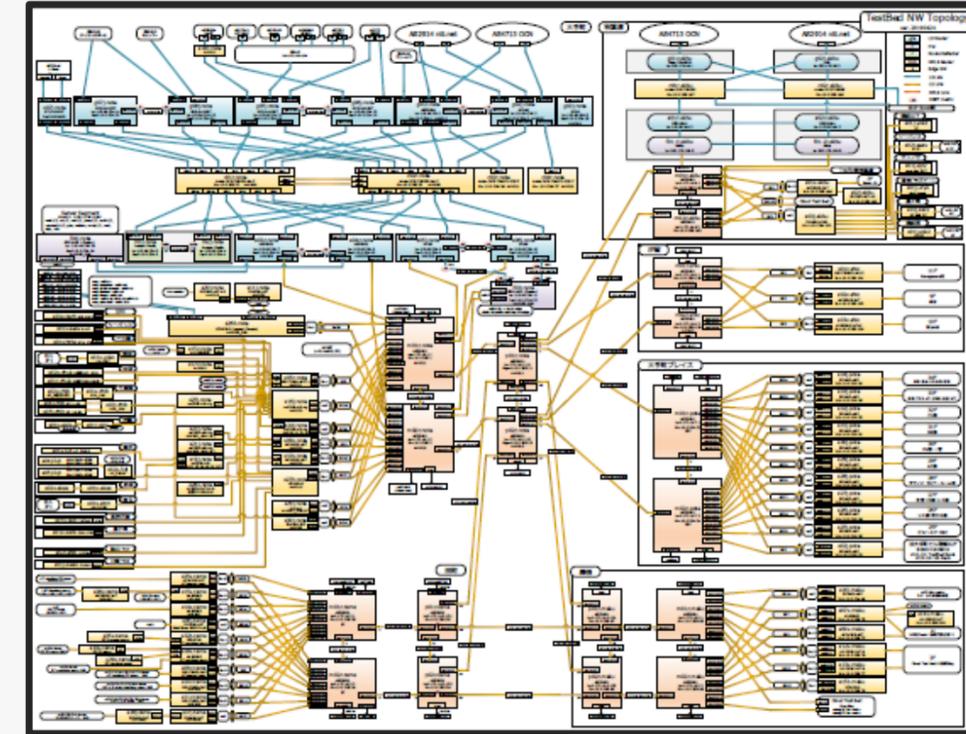
■ 図は重要

- 構成変更時はもちろん障害時にも正確なものが必要
- 全体を「だいたい」把握するには手書き図が最適

■ ラボ環境の構成図事情

- 検証構成のバリエーションが多くメンテコストが高い
- 頻繁に構成が変更され、さらにコストがかさむ
- 市販NMSで一括生成しようにもマルチベンダ対応が難しい

→ 何とか自動生成したい → **データソース** + **表示部分** が課題



※ ラック/伝送装置/underlay/overlayなどいくつか階層がありますが、今回は簡単化のために単純なL2のトポロジとします。

データソース: NetBoxからトポロジの生成

■ DCIM/IPAMとしてNetBoxを使用していた

- 拠点、ラック、デバイスなどの情報が登録済み
- 回線情報やケーブルの情報を追加



■ デバイスをノード、ケーブルや回線をリンク(エッジ)としてグラフ化

- 閉路が多数あるので探索済みリストを使いつつ深さ優先探索などで実施
- cronで定期実行し720ノード 986リンクのグラフを生成中

表示部分: inet-hengeでの可視化

- jsonで記述したトポロジをブラウザで表示するD3.jsモジュール

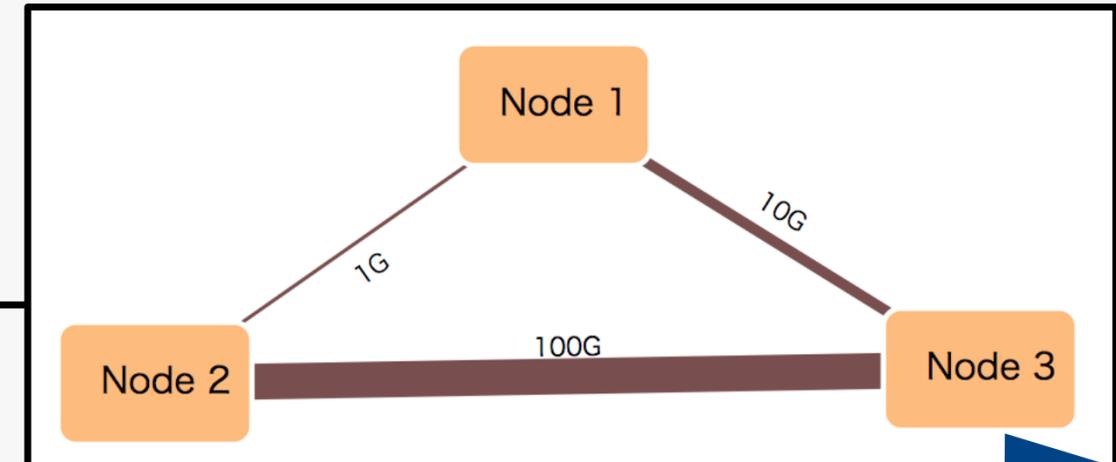
<https://github.com/codeout/inet-henge>

- svg生成の利点

- 検索できて見つけやすい
- NetBoxにリンクを貼れる

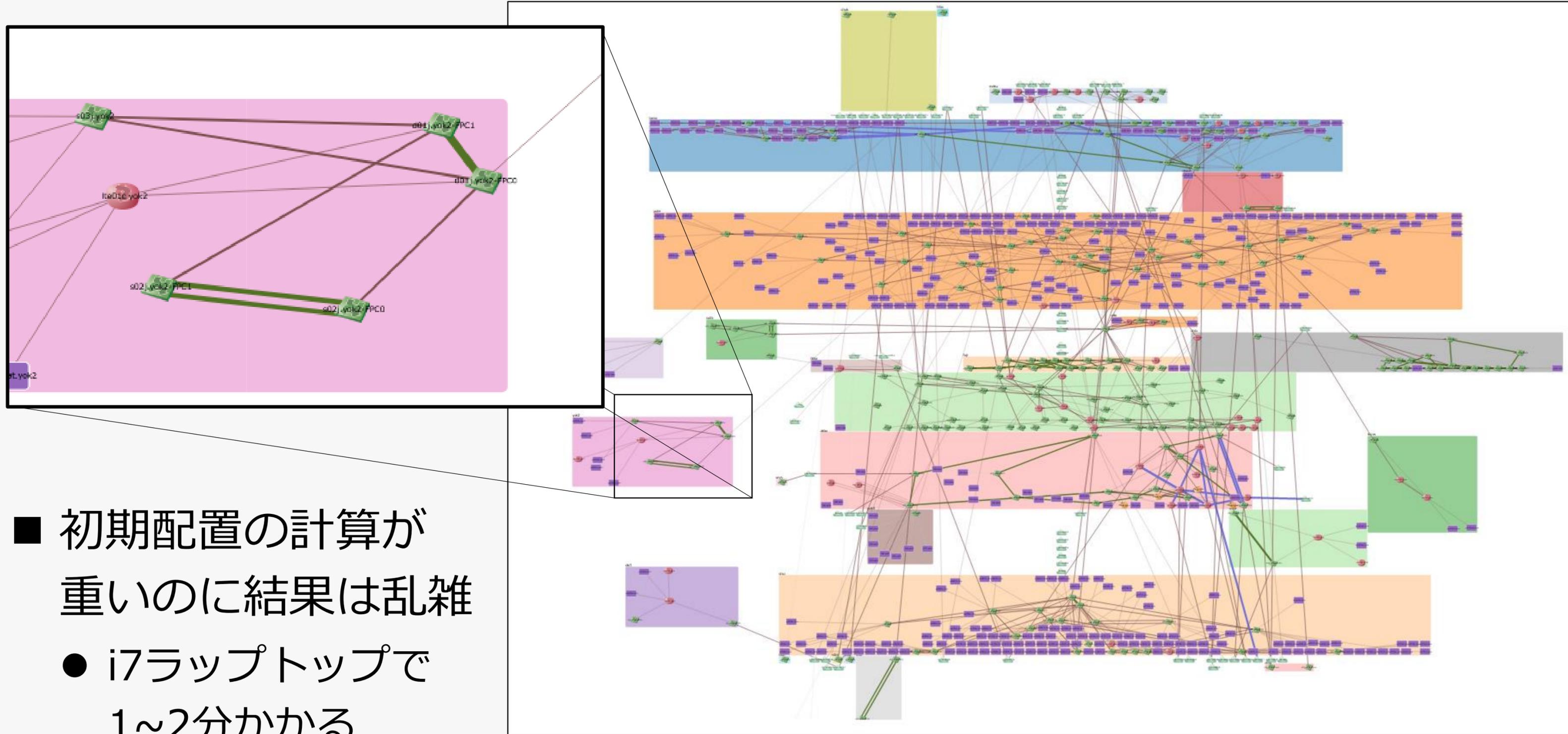
- 見た目のカスタマイズ

- グルーピング・ノード画像・リンクの幅などNW図っぽい図が作れる



```
{  
  "nodes": [  
    { "name": "Node 1" },  
    { "name": "Node 2" },  
    { "name": "Node 3" }  
  ],  
  "links": [  
    { "source": "Node 1", "target": "Node 2",  
      "meta": { "bandwidth": "1G" } },  
    { "source": "Node 2", "target": "Node 3",  
      "meta": { "bandwidth": "100G" } },  
    .....  
  ]  
}
```

結果：細かい表現ができたが、少し見づらい



- 初期配置の計算が重いのに結果は乱雑
 - i7ラップトップで1~2分かかる

改良へのきっかけ

■ janog slack 雑談ch



codeout 3:49 PM

株式会社コーダンスの小島と申します。

ネットワーク図を自動レイアウトで描くという取り組みをしています。アルゴリズム評価をしたくて、100ノード以上あるネットワーク図を提供していただける方を探しています。もしご興味ある方いらっしゃいましたら、ぜひご連絡ください！

どんなことをやっているかはコチラ→<https://github.com/codeout/inet-henge>



GitHub

[codeout/inet-henge](https://github.com/codeout/inet-henge)

Generate d3.js based Network Diagram from JSON data. - codeout/inet-henge



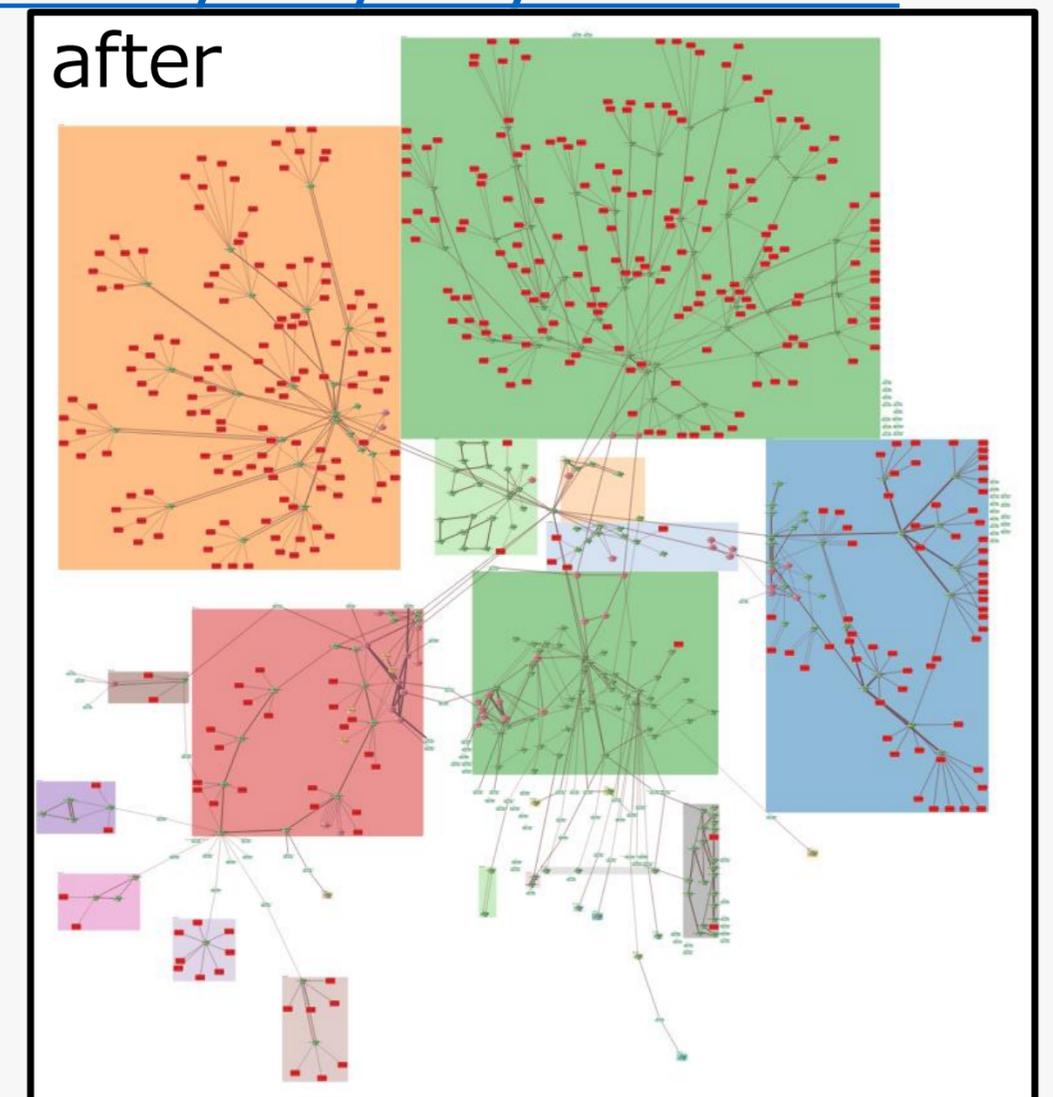
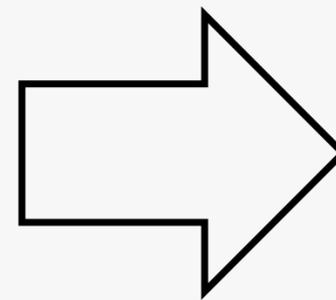
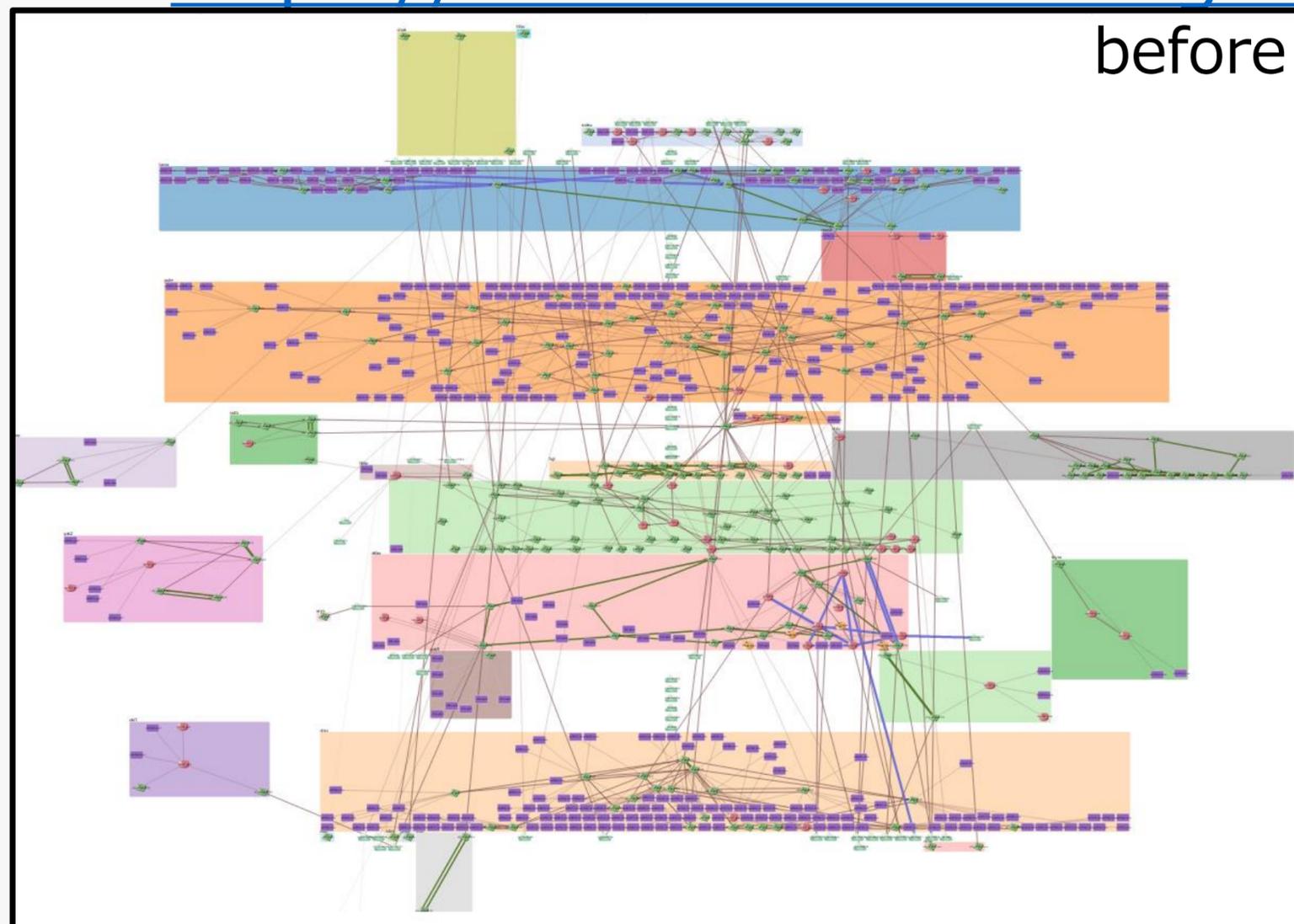
7



これに協力できればWin-Win！

inet-hengeの改良 と 改良後の表示

- リンクやグループの重なり防止が目的関数の降下を阻害
- 干渉を無視して似たノードを近くに配置する初期化用フェーズを導入
- <https://codeout.hatenablog.com/entry/2020/12/25/121305>



引き続きのチャレンジ

■ 当初の目的であった自動生成ができ、手書き図との併用はできた

■ NetBoxの登録情報のチェック

- 結局手間がNetBoxの編集に残ったため、補助するツールが必要
- LLDP情報やBatfishの解析結果など複数のソースと組み合わせて登録情報のチェック

これは「今」も正しいのか？



■ トポロジデータの再利用

- グラフデータとしてNeo4jなどに保存して再利用性を高める目的
- フロントエンド側などでオンデマンドに範囲を絞り込んだ可視化