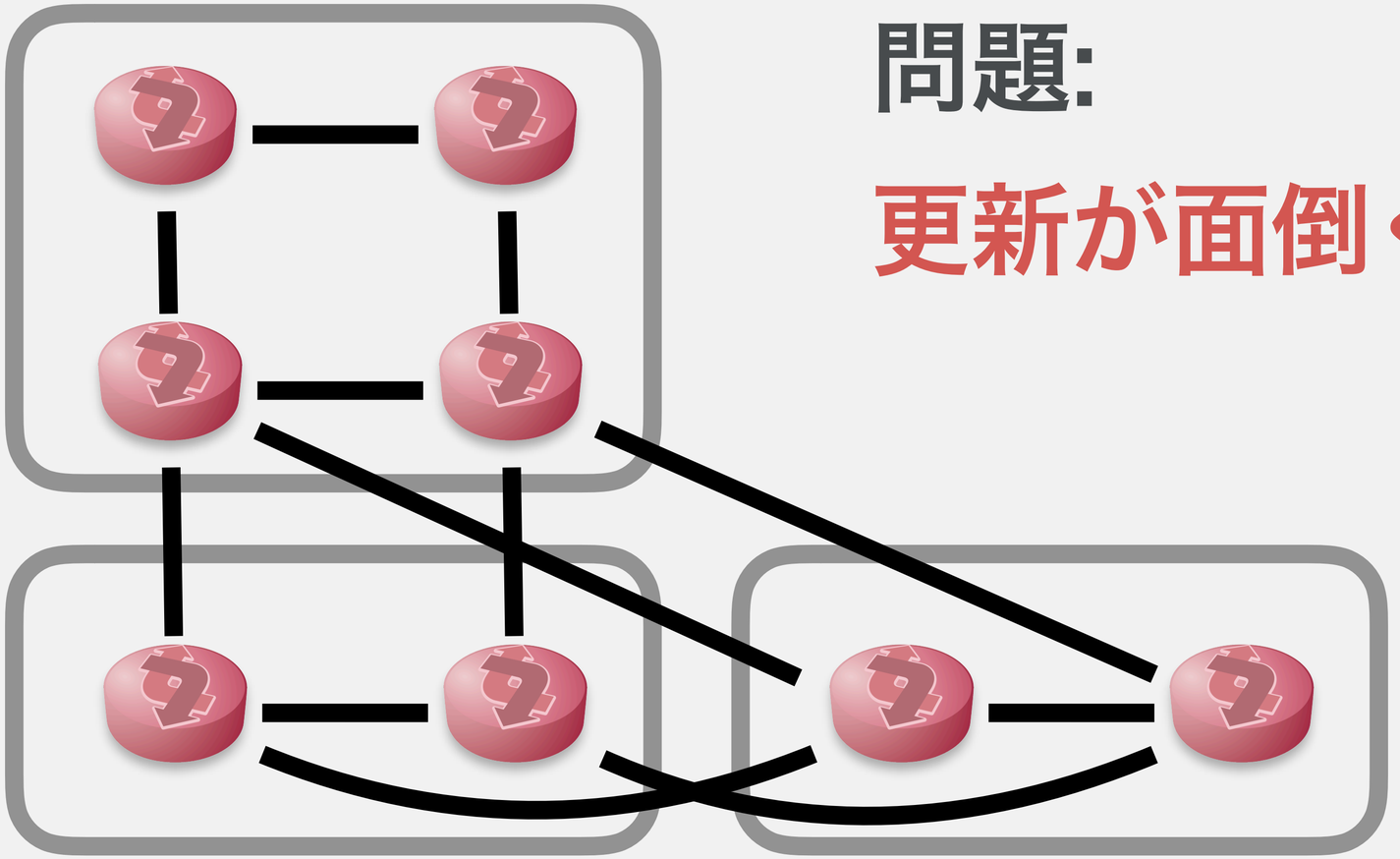
構成管理DB から ネットワーク図を 自動生成する

Shintaro Kojima
コーダンス / @codeout

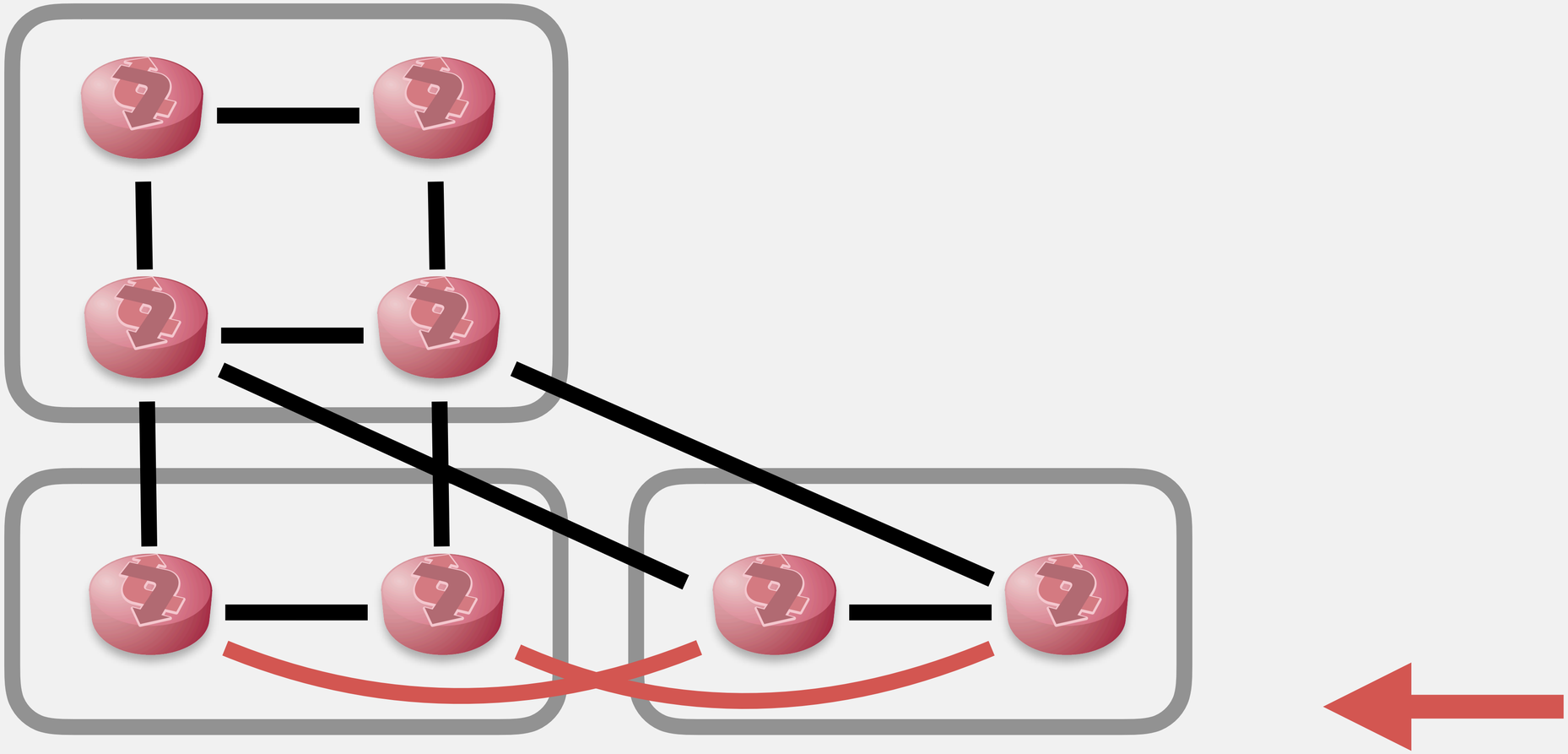
ネットワーク図の話をします



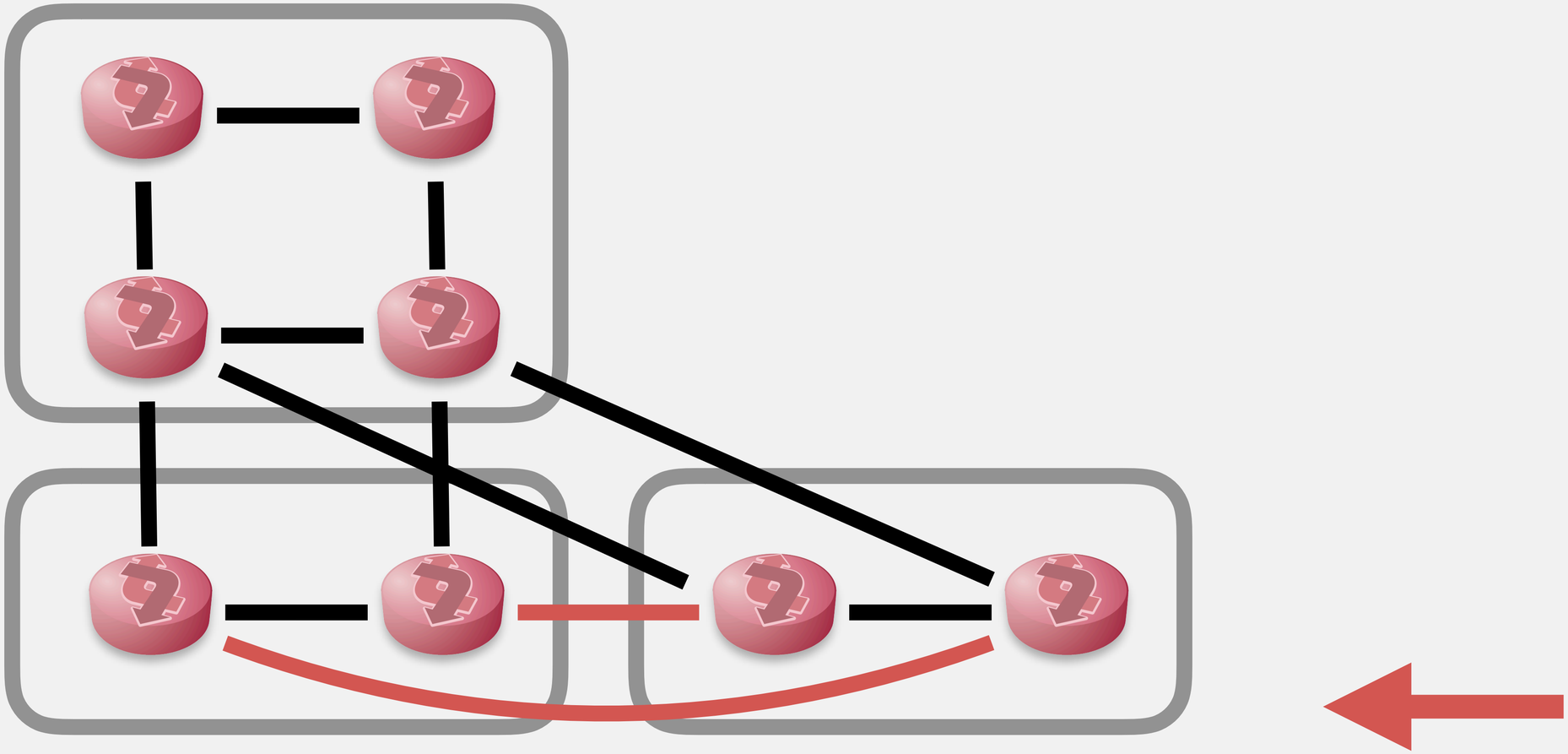
問題:

更新が面倒くさい 😢

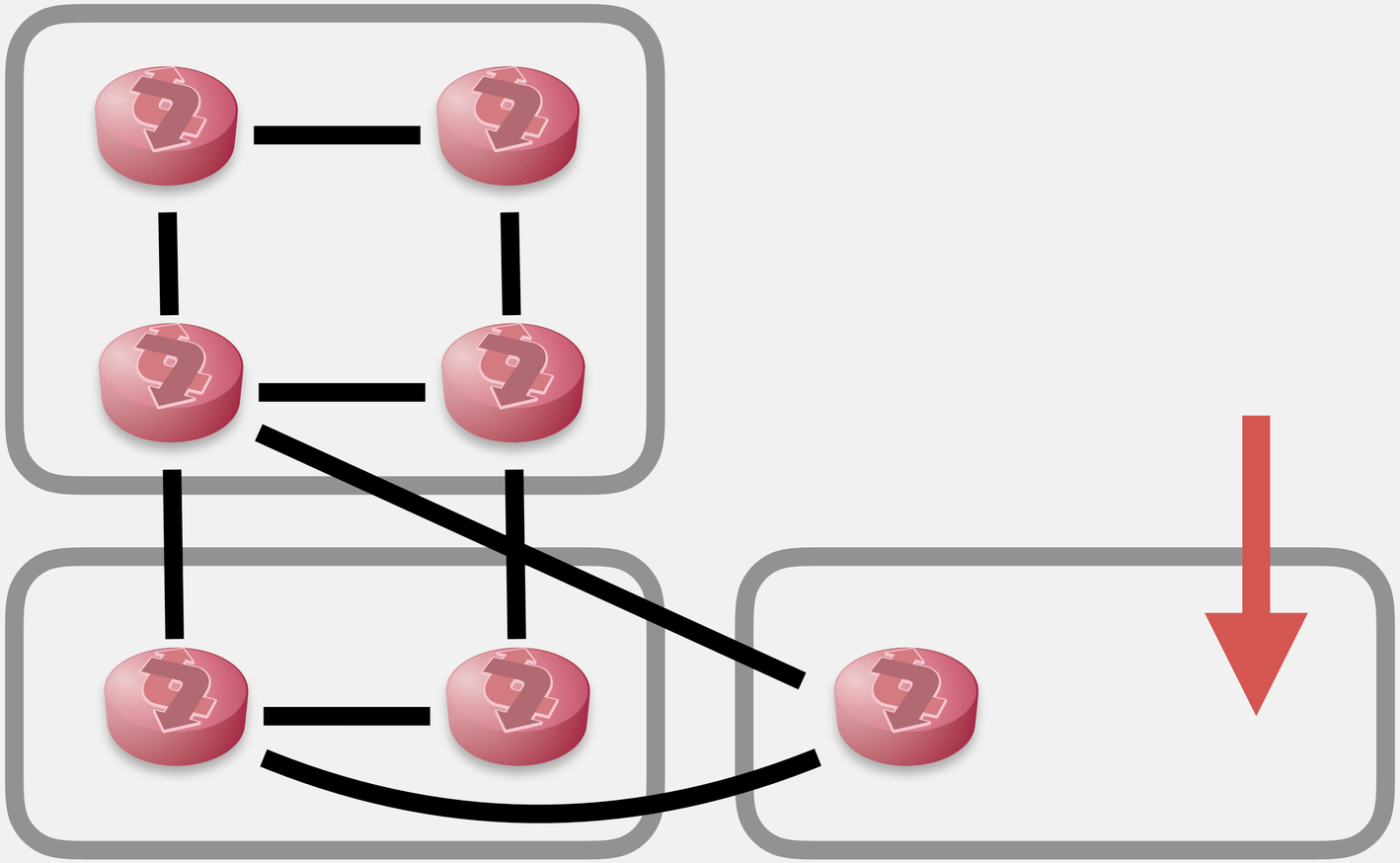
作図で悩む



間違う

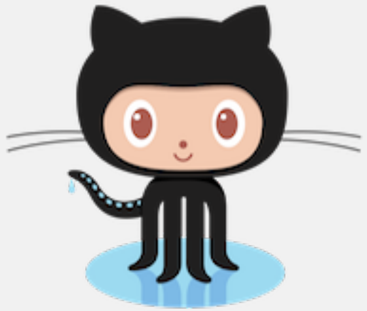


忘れる



ネットワーク図？ ないけど？

なんとかしたい

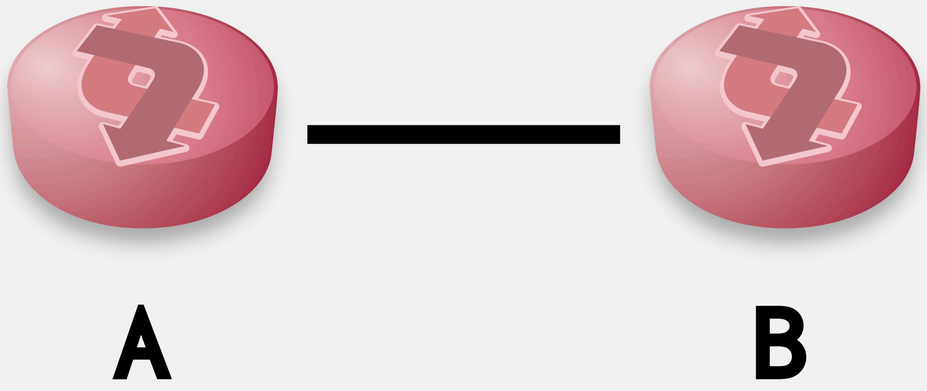


codeout/inet-henge

- 自動でネットワーク図を描くチャレンジ
- 入力 = 「デバイス1 と デバイス2 がつながっている」レベルの**接続情報**の集まり
- 「位置情報なし」縛り

入力する情報

A と B がつながっている




```
{
  "nodes": [
    { "name": "A" },
    { "name": "B" }
  ],

  "links": [
    { "source": "A", "target": "B" }
  ]
}
```

使いかた

1. 構成管理DB から JSON を書き出す
2. HTML + Javascript と一緒に、Web サーバーにのせる
3. ブラウザでアクセスする。以上！ 🎉

参考: https://github.com/codeout/inet-henge

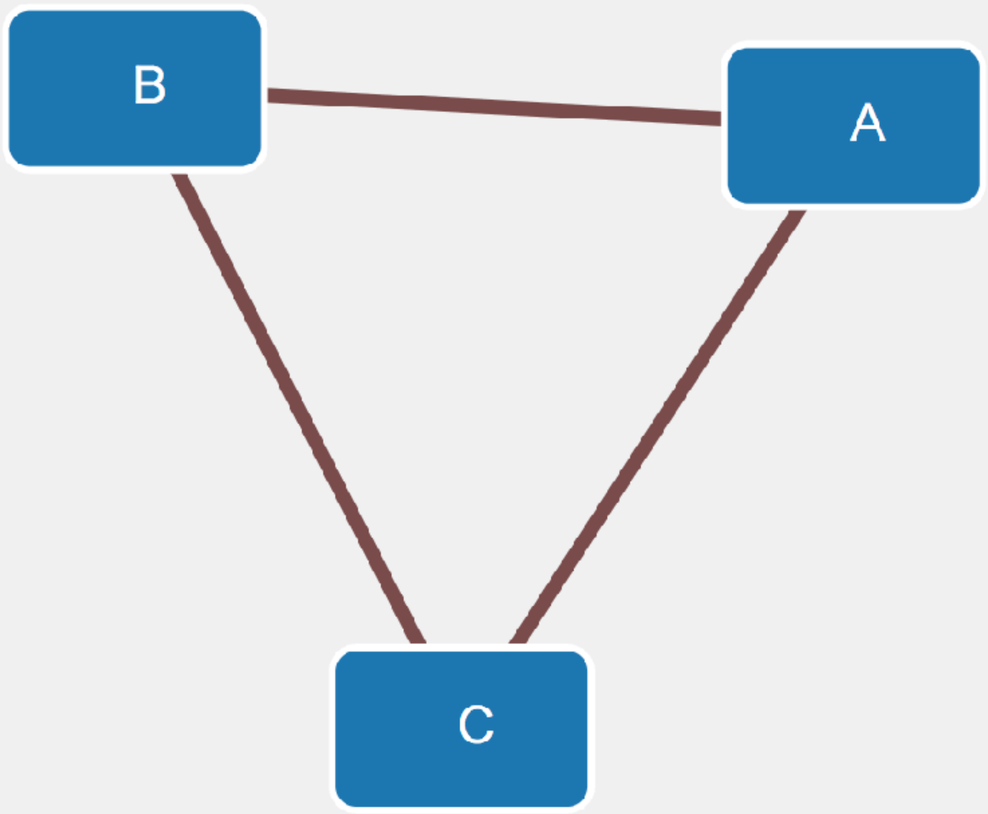
デモ: https://inet-henge.herokuapp.com/

先ほどの例

入力

```
{
  "nodes": [
    { "name": "A" },
    { "name": "B" }
  ],

  "links": [
    { "source": "A", "target": "B" }
  ]
}
```

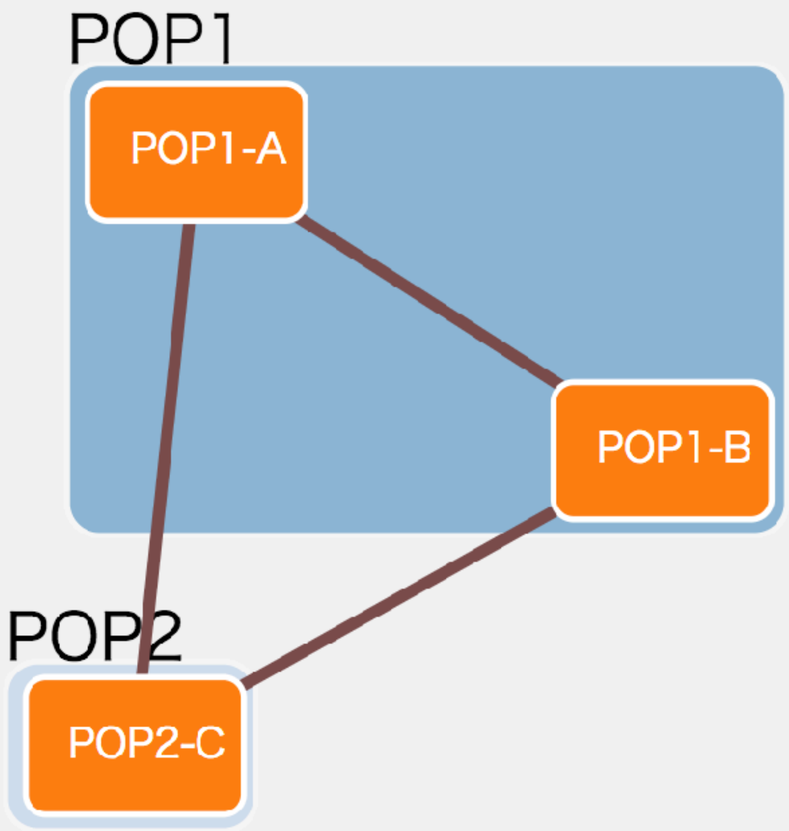
3ノード



入力

```
{
  "nodes": [
    { "name": "A" },
    { "name": "B" },
    { "name": "C" }
  ],

  "links": [
    { "source": "A", "target": "B" },
    { "source": "B", "target": "C" },
    { "source": "C", "target": "A" }
  ]
}
```

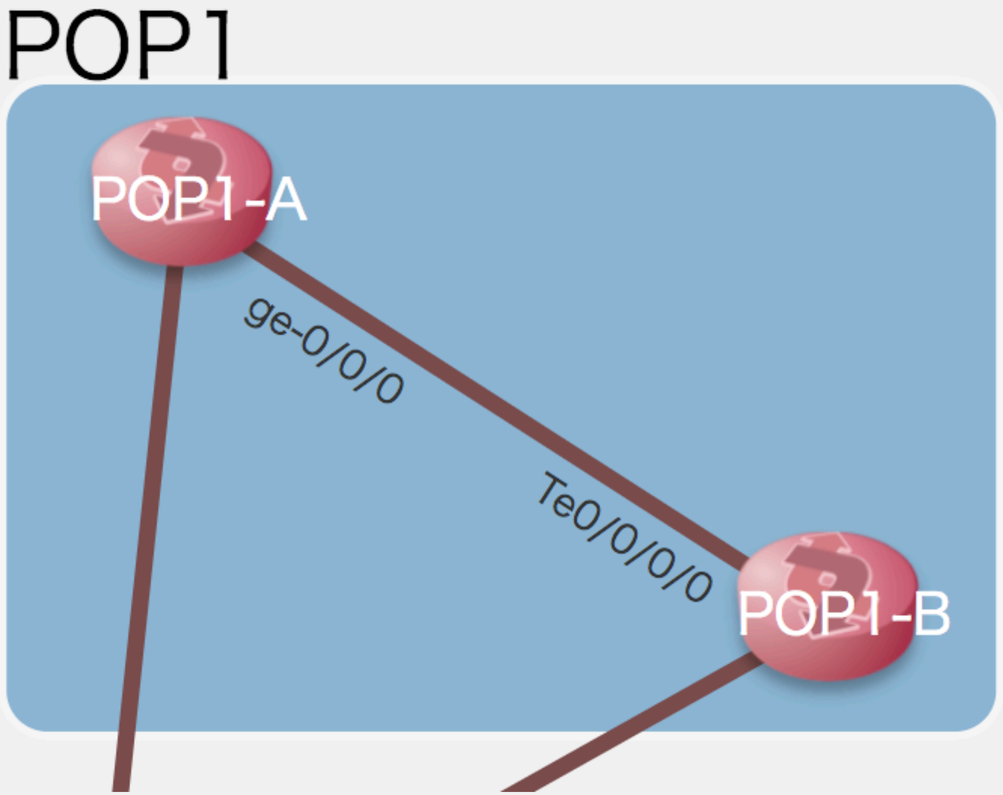
POP名による グルーピング



入力

```
# json
{
  "nodes": [
    { "name": "POP1-A" },
    { "name": "POP1-B" },
    { "name": "POP2-C" }
  ],

  "links": [
    { "source": "POP1-A", "target": "POP1-B" },
    { "source": "POP1-B", "target": "POP2-C" },
    { "source": "POP2-C", "target": "POP1-A" }
  ]
}

# html
<script>
  var diagram = new Diagram(
    '#diagram', 'index.json', {pop: /^([^\s-]+)-/}
  );
  diagram.init('loopback', 'interface');
</script>
```

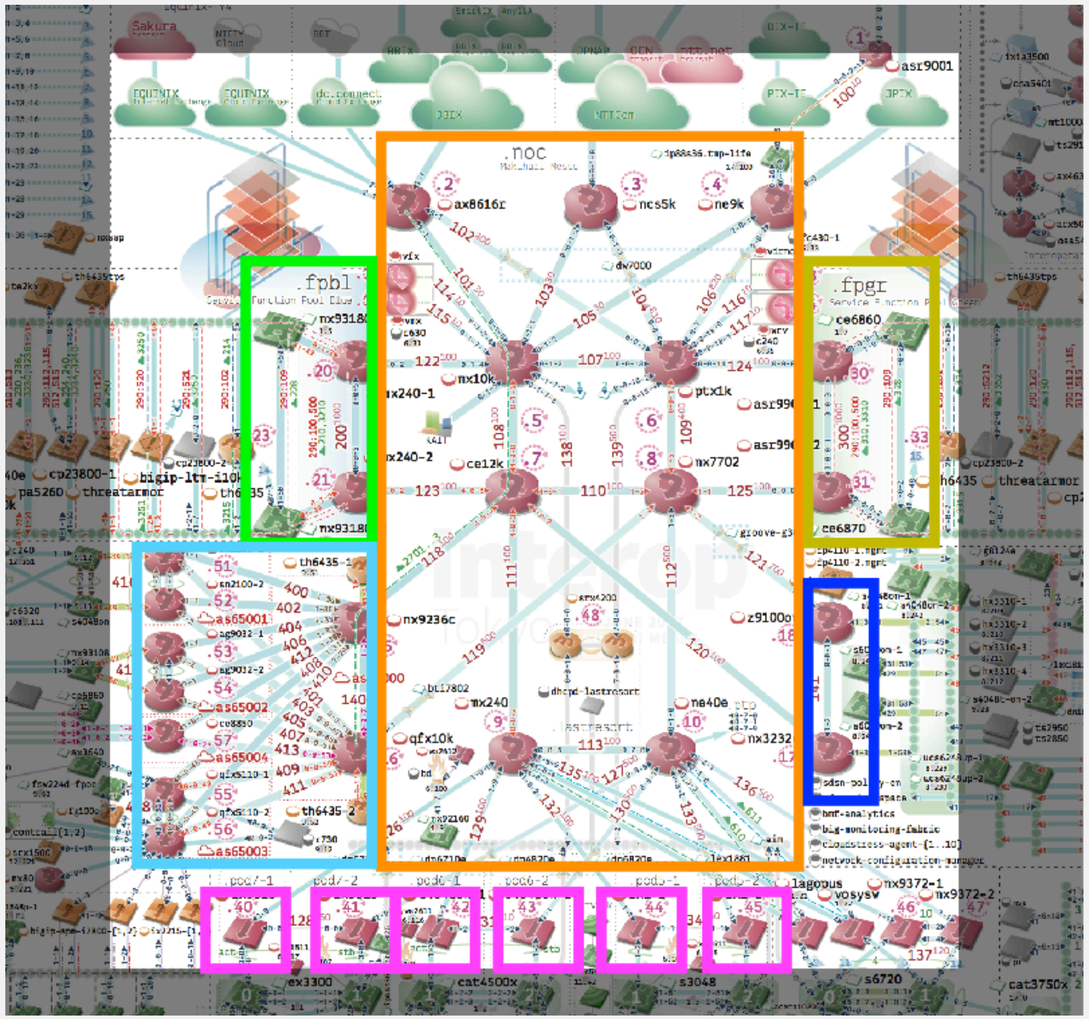
アイコンと
メタデータ



入力

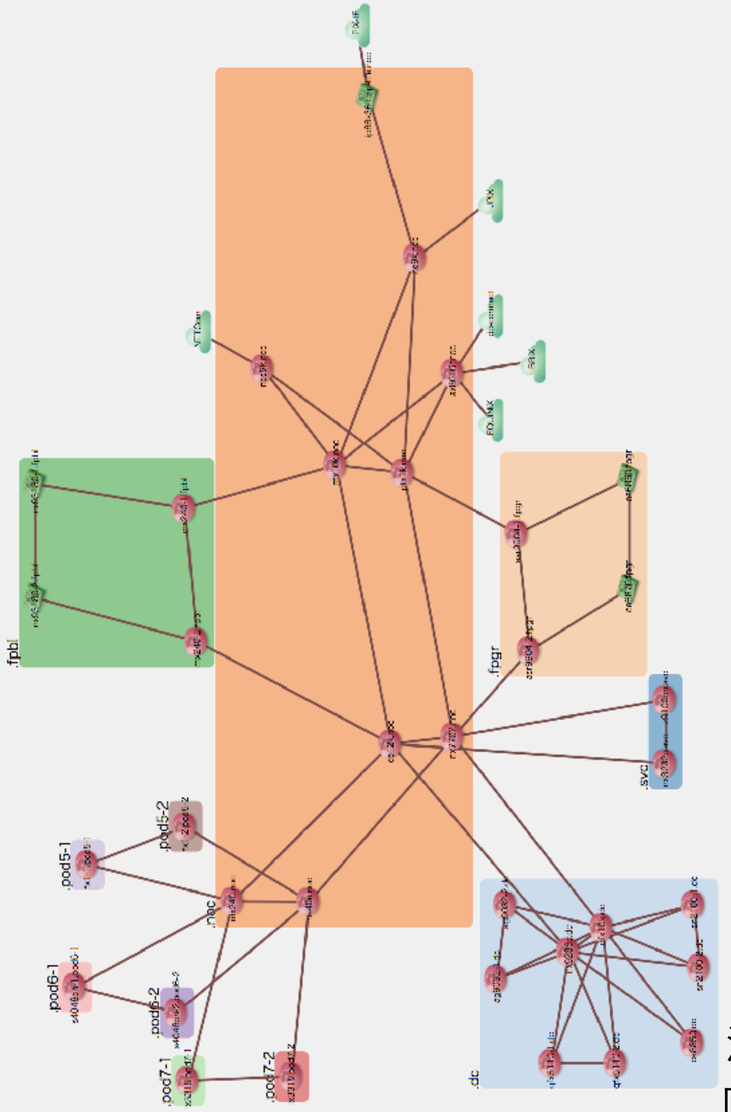
```
{
  "nodes": [
    { "name": "POP1-A",
      "icon": "./images/router.png" },
    { "name": "POP1-B",
      "icon": "./images/router.png" },
    ...
  ],

  "links": [
    { "source": "POP1-A", "target": "POP1-B",
      "meta": {
        "interface": {
          "source": "ge-0/0/0",
          "target": "Te0/0/0/0" }}
      },
    ...
  ]
}
```

ShowNet 2017 トポロジー



https://www.interop.jp/2017/images/shownet/shownet_topology.pdf

JSON



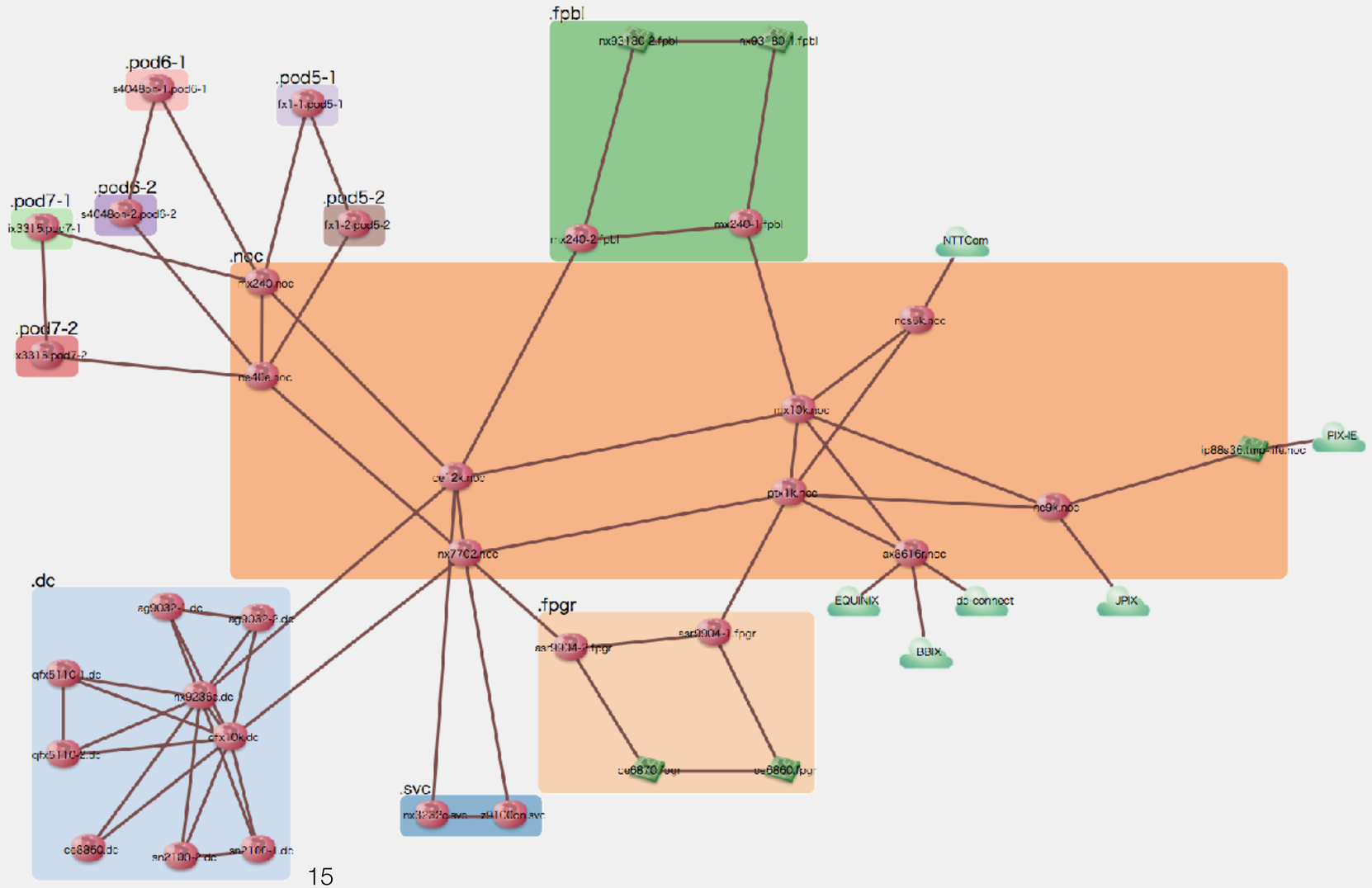
注) 比較のため
回転しています

いいのでは！ 🎉🎉🎉

美しさで

手書きに勝てないが、

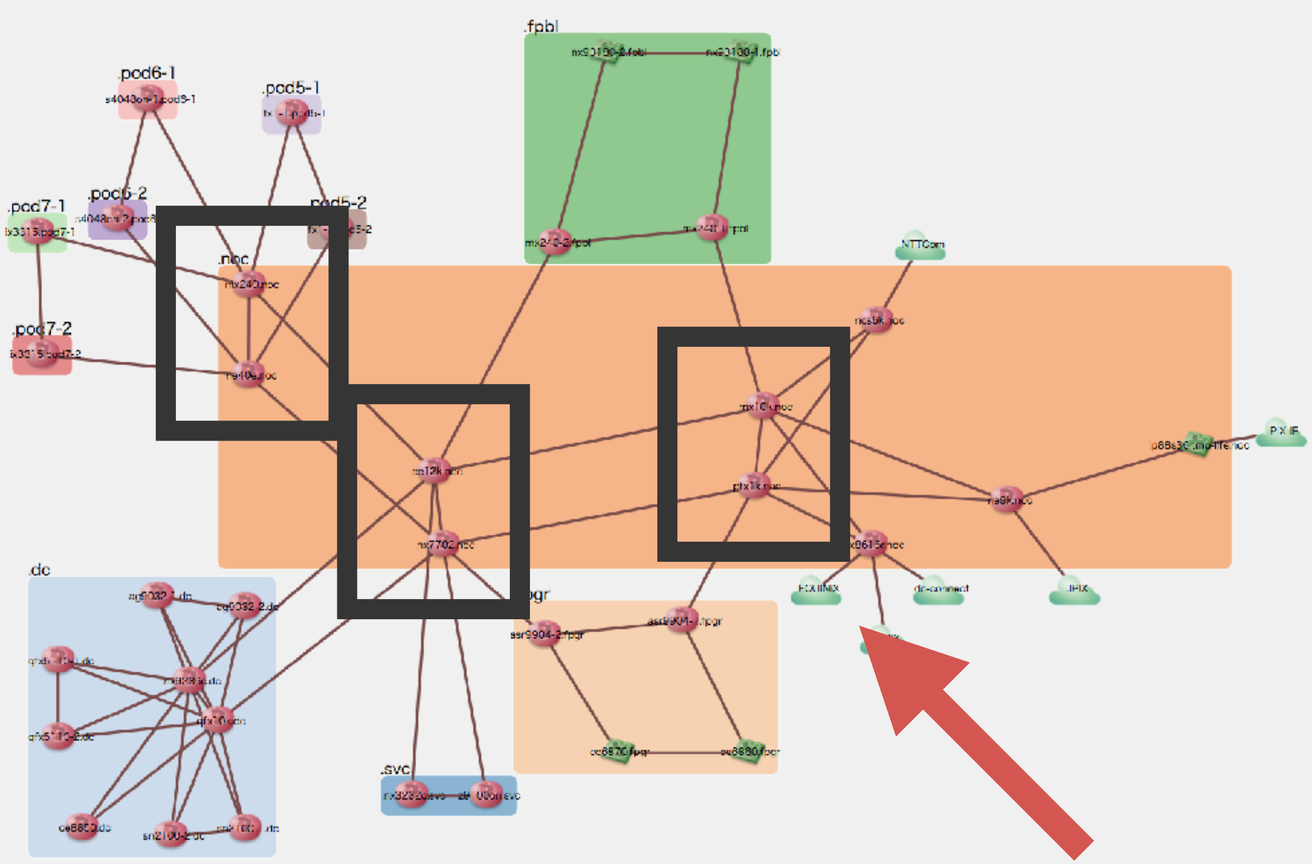
実用に耐える



オートレイアウトについて少し

アルゴリズムによる描画。
基本的なアイデアは

- D3.js の force layout に制約を加える
- 制約 = 似たノードは近いところに配置

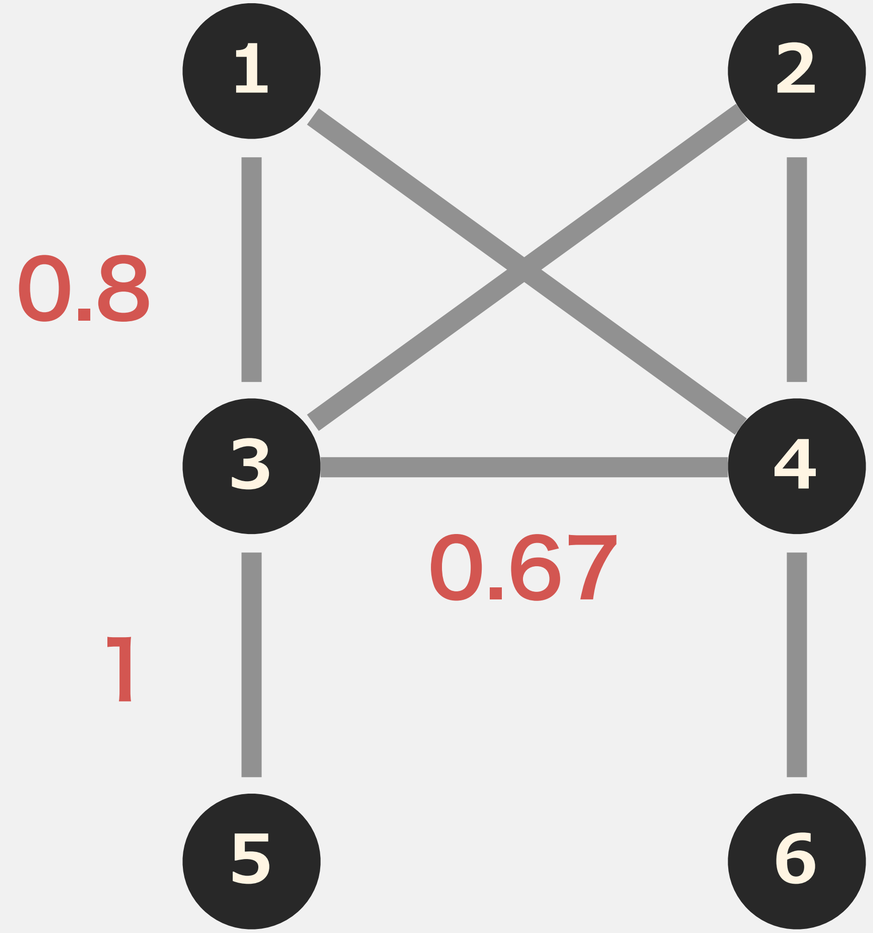


「似たノード」とは？

「似たノード」＝
「共通のノードにつながっている」

- 対向ノード集合のジャッカード距離を
  ノード間の距離に採用

- 例: ③ は [①, ④, ⑤] に、
  ④ は [②, ③, ⑥] につながっている。
  ③ ↔ ④ の距離は、[①, ④, ⑤] ↔ [②, ③, ⑥]
  のジャッカード距離



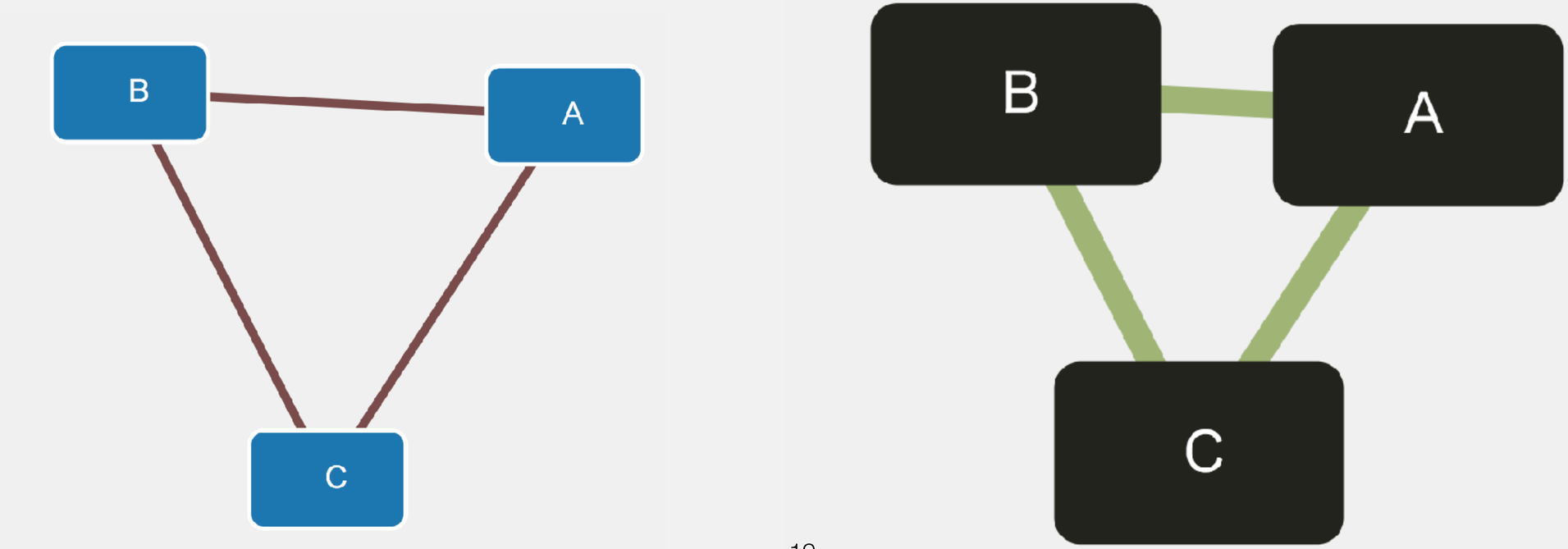
デモ

https://inet-henge.herokuapp.com/

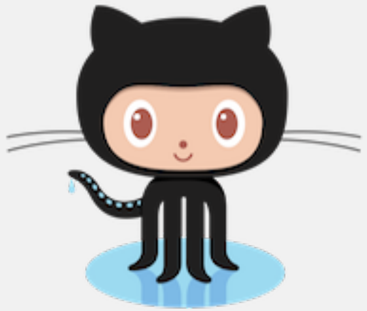
- 💡 注目ポイント
  - リロードで配置が変わらない
  - ズームすると細かい情報を表示

SVG のメリット

CSS によるスタイリングが可能



まとめ



codeout/inet-henge

- 自動でネットワーク図を描くチャレンジ
- 「位置情報なし」縛り
- D3.js ベースの SVG を出力
- そこそこいけるはず。ぜひ使ってみてください！

手伝ってくれる方を募集！！

- 使ってみてくれる方
- ご意見、不具合報告をしてくれる方
- パッチを送ってくれる方

みなさんのご協力をお待ちしています！