



# NPM(Network Performance Monitoring) New Relic で実現するNetwork Observability

Akihiro Ito

Senior Technical Support Engineer

# Agenda

- Network Observability とは
  - NetworkモニタリングからNetworkオブザーバビリティへ
  - SNMP Monitoring
  - NetFlow Monitoring
  - NPMの費用

**なぜか社内システムにつながらない！！**

**特に9時と13時、、、**



VPN



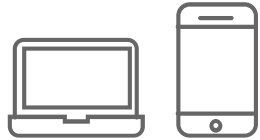


きっとアイツ(VPN)のせいだ、、、

# NetworkモニタリングからNetwork オブザーバビリティへ

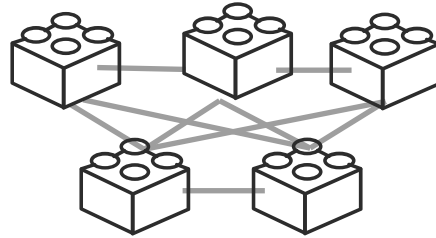
# これまでのNew Relic

RUM



New Relic.  
**MOBILE**<sup>™</sup>  
New Relic.  
**BROWSER**<sup>™</sup>

アプリ



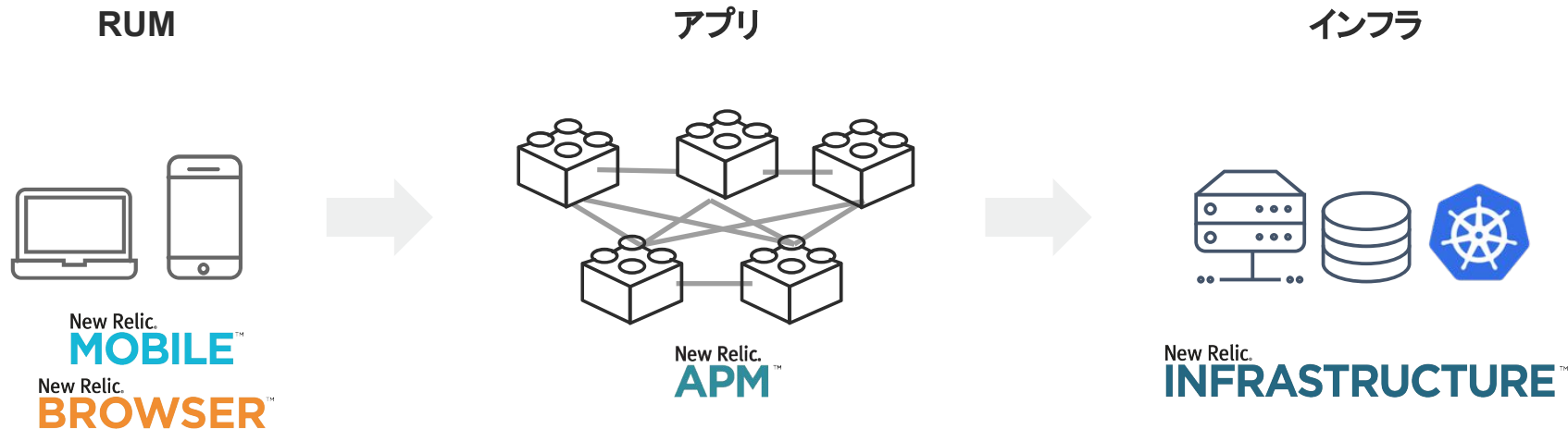
New Relic.  
**APM**<sup>™</sup>

インフラ



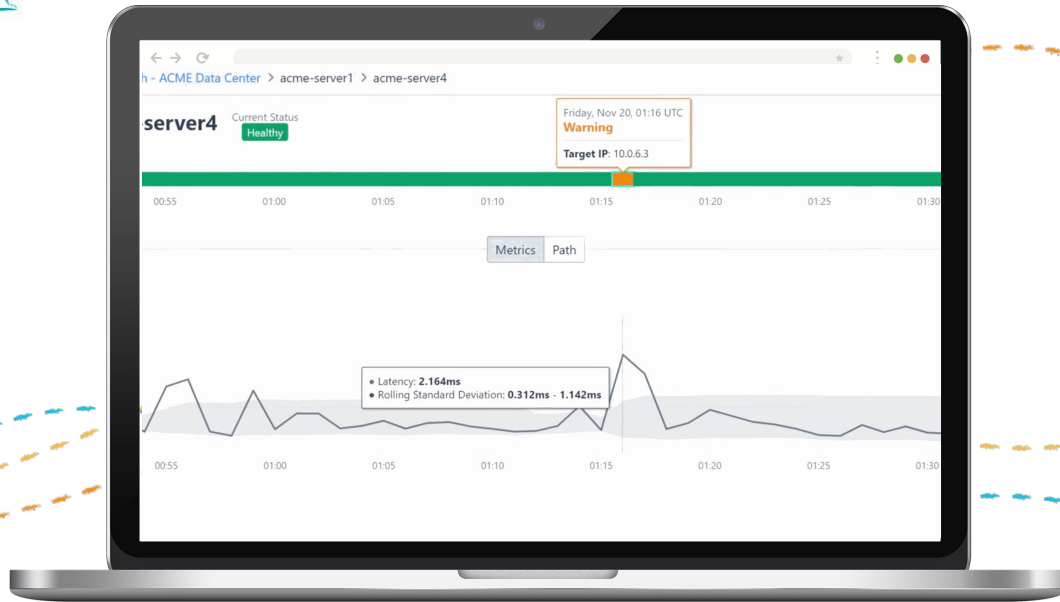
New Relic.  
**INFRASTRUCTURE**<sup>™</sup>

# これまでのNew Relic



ネットワーク監視はOn-host Integrationを利用したSNMP監視のみ

# New!! Kentikとの協業によるNetwork Observability





# Network Observabilityで実現できること

- **SNMPによるNetwork機器監視**

- CPU使用率、Memory使用率、Port探知のPacket数監視
- CIDR を指定する事で、ネットワーク機器を自動でディスカバリ

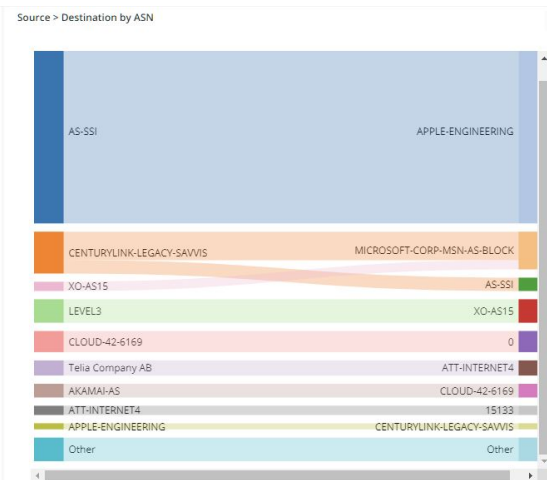
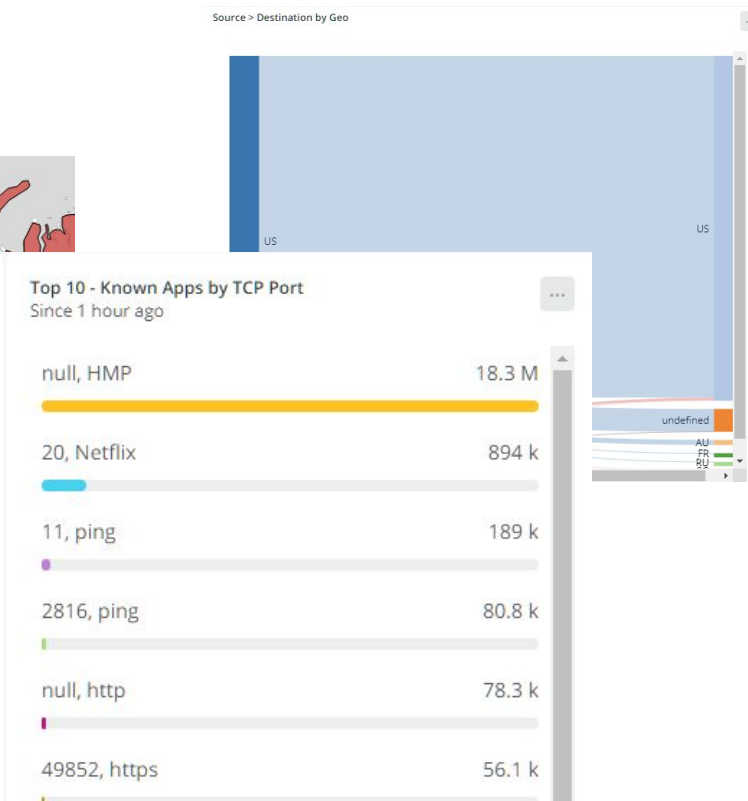
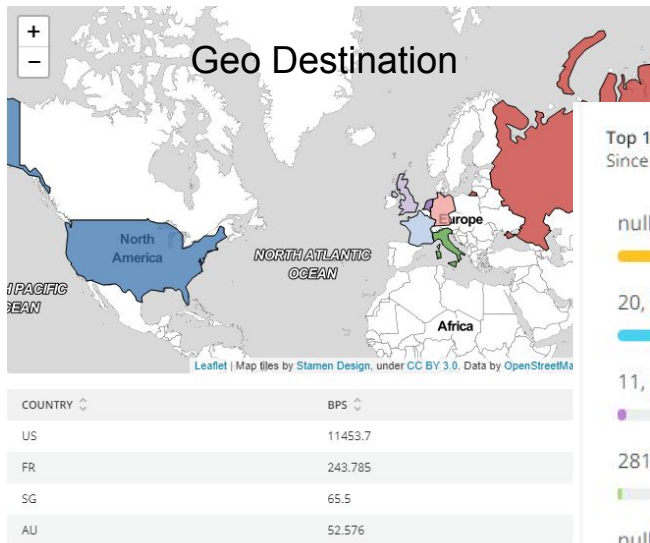
- **FLOW技術によるNetworkトラフィックの可視化**

- プロトコル単位、宛先単位でのトラフィック収集、ボトルネックを分析
- 機器のキャパシティーに加え、アプリケーションの通信品質、QoSの効果確認
- 経路の可視化
- AWS、GCP、Azureのネットワークトラフィックの可視化

※New Relic版Kentikでは一部未実装の機能あり

# Network Observability

## Flowデータの可視化



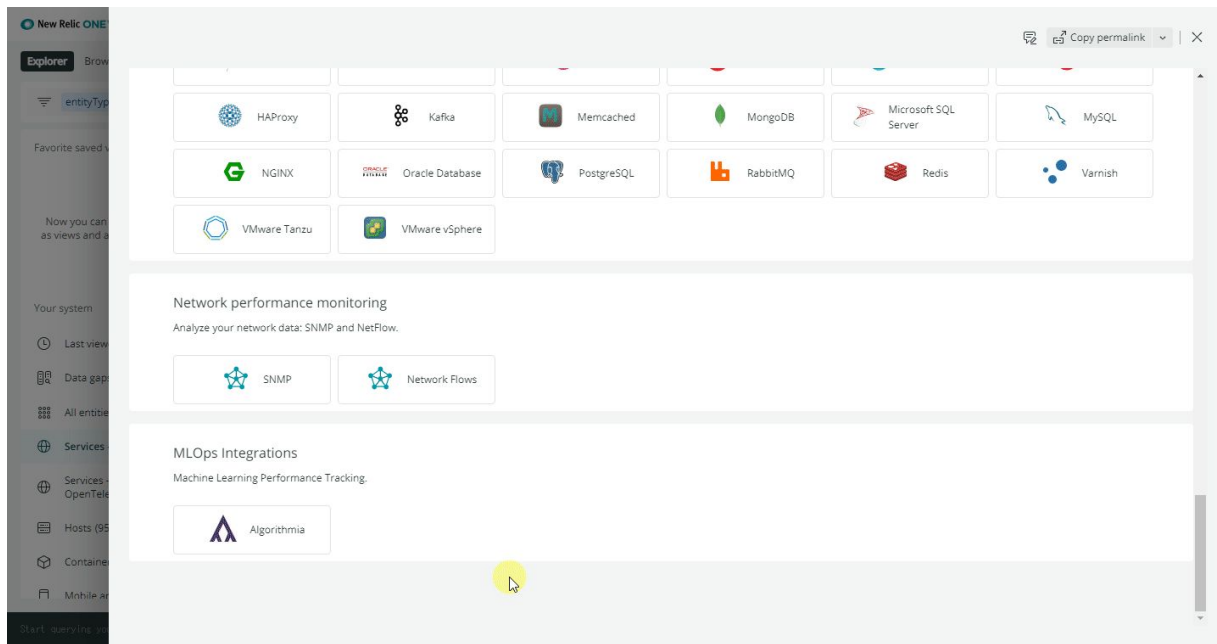
## Source-Destination グラフ

## プロトコル 通信量

# SNMP Monitoring

# SNMP Monitoring

CIDR を指定する事で、ネットワーク機器をディスカリーし、自動的に値を収集します。  
YAMLにOIDを追加する事で標準MIB以外の値を収集することができます。



# NetworkObservability

## SNMP データの可視化

| Interface Inventory<br>Since 1 hour ago |                  |                    | Unique OIDs per MIB<br>Since 1 hour ago |                      |
|---|------------------|--------------------|---|----------------------|
| DEVICE NAME ↕                           | INTERFACE NAME ↕ | INTERFACE STATUS ↕ | MIB ↕                                   | OBJECT IDENTIFIERS ↕ |
| techops_branch01_edge                   | Et0/0            | up                 | BGP4-MIB                                | 16                   |
| techops_branch01_edge                   | Se0/0            | up                 | IF-MIB                                  | 14                   |
| techops_branch01_edge                   | Se0/0.1          | up                 | TCP-MIB                                 | 8                    |
| -----                                   | -----            | -----              | -----                                   | -----                |

# 設定項目

- **SNMP version**

- モニタリング対象Network機器が対応しているSNMPバージョンを指定します。

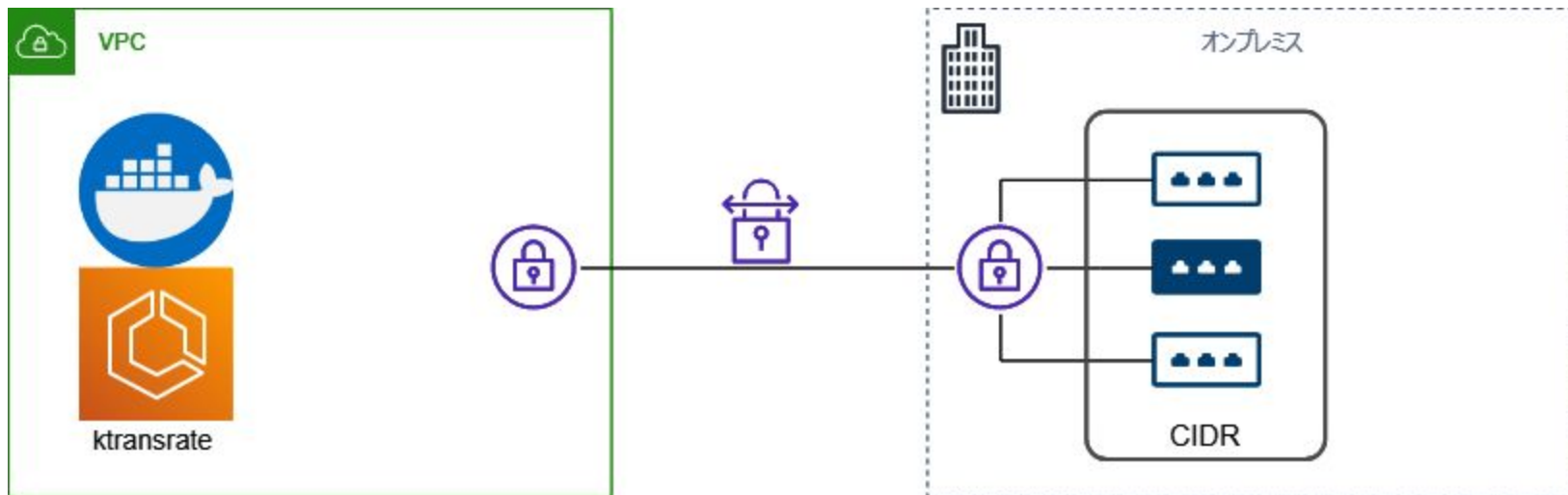
- **CIDR(Classless Inter-Domain Routing)**

- Network範囲のこと。Ktransrateがネットワーク機器を探すIPアドレス範囲を指定します。

- **Community string(コミュニティー名)**

- SNMPにおけるパスフレーズです。これによりNetwork機器はSNMPモニタリングツールを認証しています。

# 設定イメージ



KtranslateにはDockerを配置するCIDRではなく、収集対象のNetwork機器が置かれているCIDRを設定します。

# NetFlow Monitoring



# NetFlow Monitoring

NewFlow の送信元となるデバイス情報をSNMPから連携する、あるいは手動でデバイス名を登録することができます。

NetFlow V5,V9  
IPFIX,sFlow,jFlowに  
対応しています。

The screenshot shows the New Relic ONE console interface for configuring Network Flows collection. The page is titled "Network Flows collection setup" and contains the following steps:

- Select your flow type**  
Flow type: `netflow9`
- Configure your Network Flow collection**  
Select a method below. Did you already set up SNMP collection?  
 Yes, use SNMP yaml file to map device names  
 No, manually add details about devices sending network flows  
[Validate and continue](#)
- Listen for Network Flows**  
We will use your default Network Flow insert key. Optionally you can [generate and use a new key](#).  
Then run the following command:  
`Load line...` [Copy](#)
- Next steps**  
If you'd like to send data with a different flow type, repeat steps 2 and 3.  
Use this tool to [Visualize your network data](#)

The left sidebar shows the "Explorer" menu with options like "entityType", "Favorite saved", "Now you can as views and a", "Your system", "Last view", "Data gap", "All entity", "Services", "Services - OpenTe", "Hosts (95)", "Contain", and "Mn". The right sidebar contains "monitoring" links and "More help" links.

# NetFlow Monitoring

|                        |                |
|------------------------|----------------|
| Network flow data type | ktranslate 設定値 |
| IPFIX                  | ipfix          |
| NetFlow v5             | netflow5       |
| NetFlow v9             | netflow9       |
| sFlow                  | sflow          |
| AppFlow                | netflow5       |
| Argus                  | netflow5       |
| cflowd                 | netflow5       |
| J-Flow                 | netflow5       |
| NetStream              | netflow5       |
| RFlow                  | netflow5       |
| Cisco NSEL             | netflow9       |

# 設定項目

- **Network Flows collection setup**

- Ktranslate コンテナは1つのNetFlow形式にしか対応できません。
- 複数のNetFlow形式の機材が混在する場合には形式毎にコンテナを建てます

- **Configure your Network Flow collection**

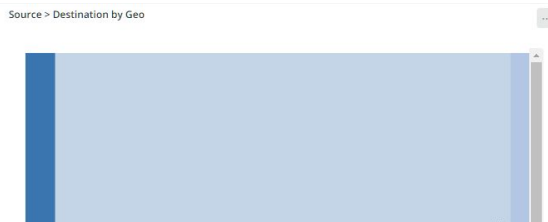
- NetFlowを送ってくる機器名称をSNMPデータからマッピングするか、手動で設定するかを選択します。
- 手動設定の場合はNetwork機器のIPアドレスと機器名称を入力します。

# Network Observability

## Flowデータの可視化



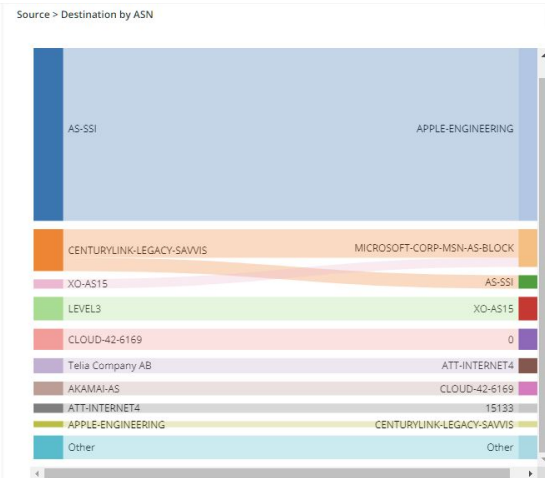
| COUNTRY | BPS     |
|---------|---------|
| US      | 11453.7 |
| FR      | 243.785 |
| SG      | 65.5    |
| AU      | 52.576  |



### Top 10 - Known Apps by TCP Port Since 1 hour ago

|              |        |
|--------------|--------|
| null, HMP    | 18.3 M |
| 20, Netflix  | 894 k  |
| 11, ping     | 189 k  |
| 2816, ping   | 80.8 k |
| null, http   | 78.3 k |
| 49852, https | 56.1 k |

プロトコル 通信  
量

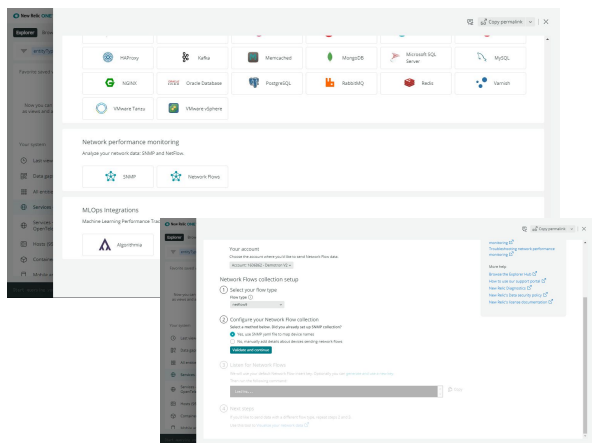


## Source-Destination グラフ

# NPMの費用

# 利用までたったの3STEP、追加契約なしで利用可能！

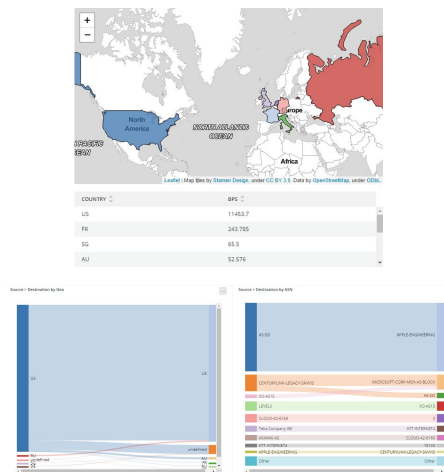
## 1 初期設定 (config作成)



## 2 監視用コンテナのデプロイ

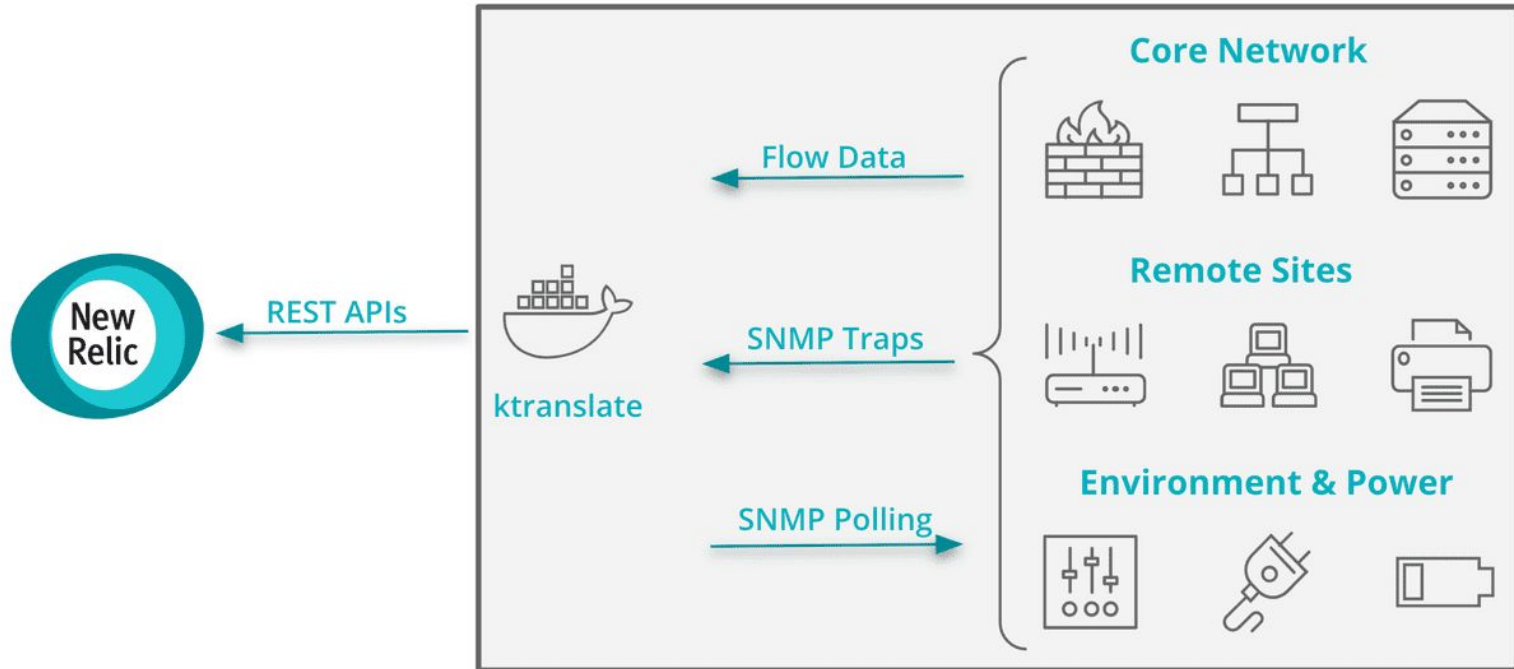


## 3 可視化分析



※データの保管にはTDP、GeoやSource-Destinationを利用したダッシュボードはFSO Fullユーザーが必要になります

# Network Observabilityの構成



# NPMの費用

- Kentikとの契約は不要です。
- ktranslateコンテナを利用します。
- SNMP情報、NetFlow情報分のTDP利用量が発生します。
- Geo Map, Source-Destination グラフ を利用する場合はFSOUserが必要です。



# Demo

*from Network Monitoring to **Network Observability!**  
and  
Happy “Full” Observability Life!*



**Thank You**