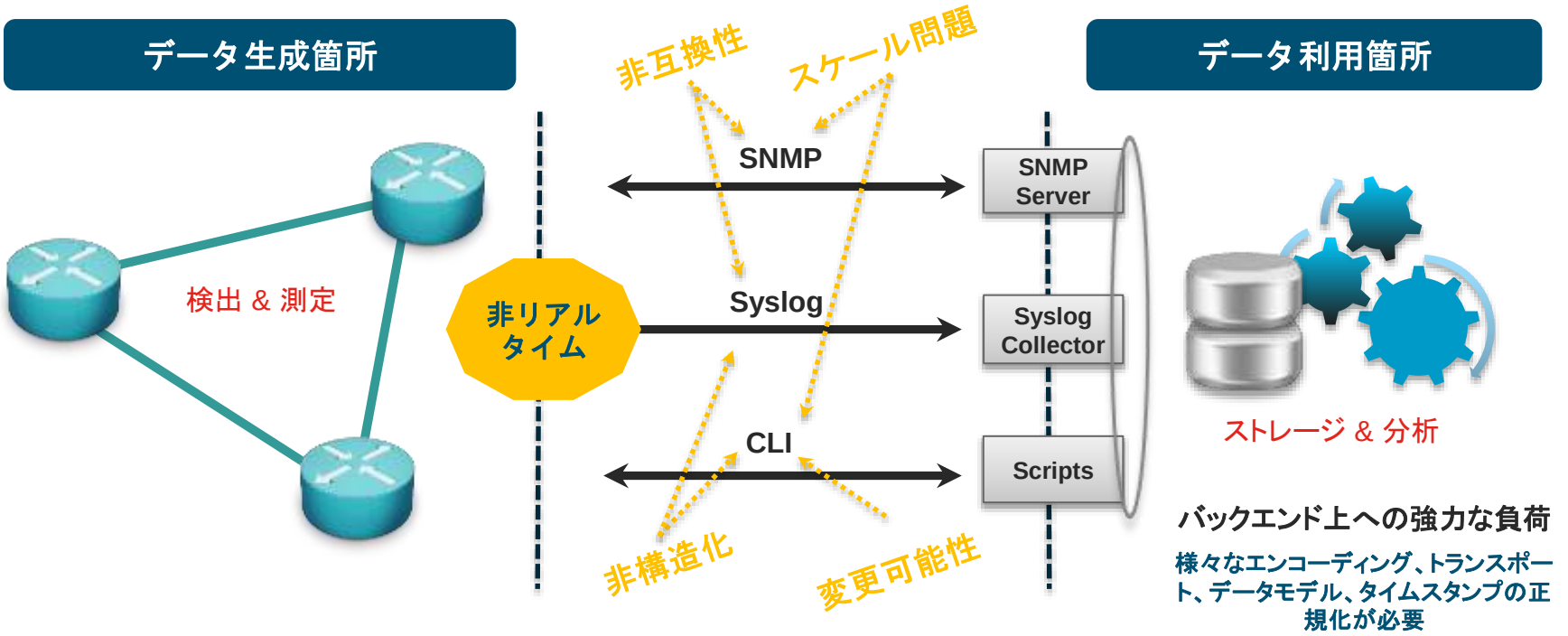
# Telemetryについて

シスコシステムズ合同会社
佐藤 哲大

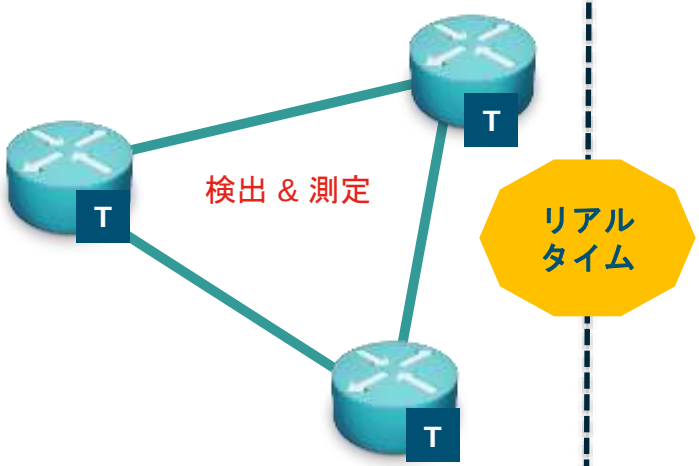
# 従来のモニタリング手法の問題点



# 新しいパラダイムへ

データ生成箇所

データ利用箇所




リアル
タイム

可能な限り
より多くのデータ
より速く
より役に立つ
より容易に

ストレージ & 分析

# Telemetry新しいアプローチ

プル型ではなくプッシュ型


解析可能なデータ

データモデルドリブン

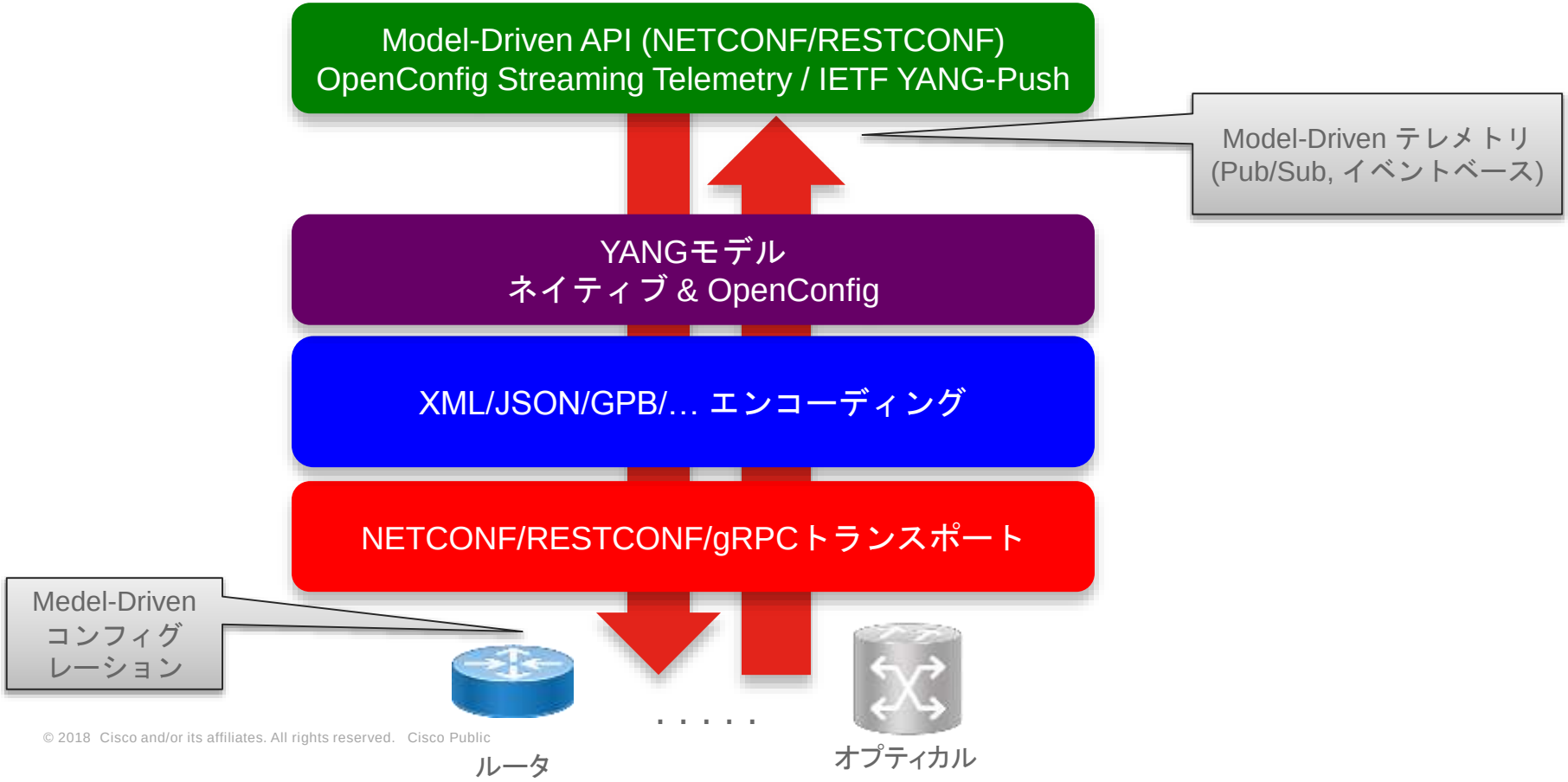
パフォーマンス

ツールチェーン

自動化

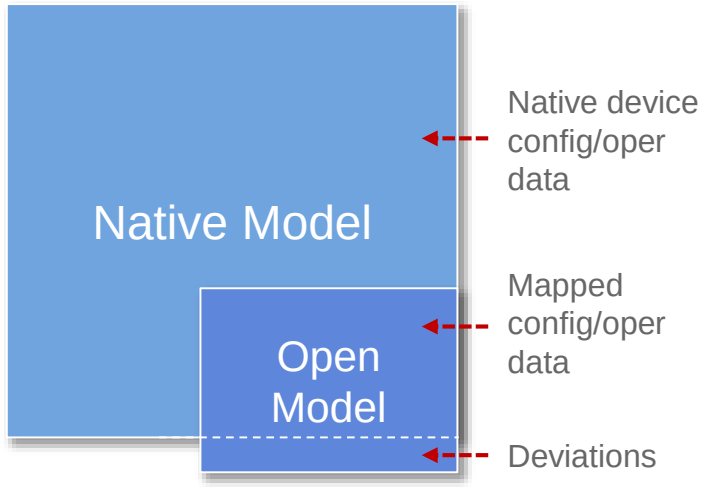
# Model Driven Manageability

# Model Driven Manageability



# Native vs Open Data Models

- ネイティブデータモデルはほとんどのコンフィグレーションとオペレーション範囲を提供
- オープンモデルはネイティブデータモデルにマッピングされる
- オープンモデルからの逸脱はdeviationモジュールで定義



# チョイス@Telemetry


- ネットワーク機器以外のコンポーネント（コレクタ、データベース、ダッシュボード等）をどうするかには、大きく2つ選択肢がある・・・

**ベンダ謹製**

- （一般論として）高機能、高品質
- 汎用的な用途に限定
- インテグレーションが難しい
- $$$

**OSS活用**

- 色々とインテグレーションできる
- ネットワーク機器側の実装方式と情報開示具合による
- 楽しい

**今日はこっちにフォーカスして、**
**機器実装とツールチェーンをご紹介**

# （XR中心に）機器実装

# Model-Driven Telemetry 設定例

```
telemetry model-driven
 destination-group DGroup1
  address family ipv4 192.0.2.1 port 5432
   encoding self-describing-gpb
   protocol tcp

!
 sensor-group SGroup1
  sensor-path Cisco-IOS-XR-infra-statsd-oper:infra-
   statistics/interfaces/interface/latest/generic-counters

!
 subscription Sub1
  sensor-group-id SGroup1 sample-interval 10000
  destination-id DGroup1
```

どういった形式でどこに送る？

どういったデータ？

どういった頻度で？

# sensor-group: 取得するデータの指定

どういったデータ？

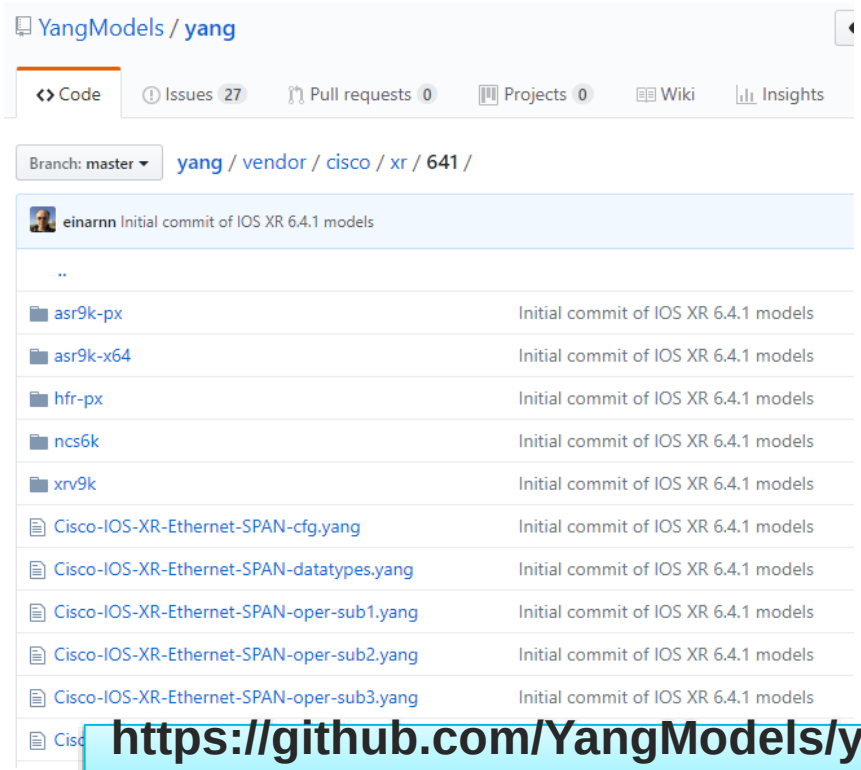
```
telemetry model-driven
 sensor-group SGROUP1
  sensor-path Cisco-IOS-XR-infra-statsd-oper:infra-
   statistics/interfaces/interface/latest/generic-counters
```

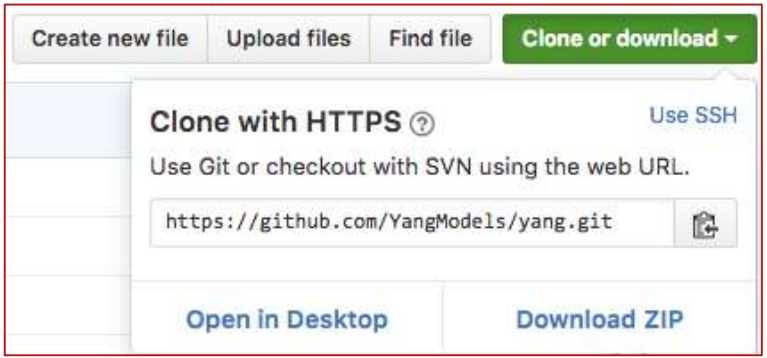
YANG Model

subtree path

# YANGモデルはGithubに公開

どういったデータ？




- 500近いオペレーションYANG モデルが存在する（IOS XR6.4.2現在）

**https://github.com/YangModels/yang/tree/master/vendor/cisco/xr**

# 該当するサブツリーの探し方

どういったデータ？

- pyangはYANGモデルを扱うためのツール
  - モデルのバリデーション
  - 他のモデル（Yin, XMLスキーマ等）へ変換
  - などなど
- ツリー形式で表現することもできる

```
$ pyang -f tree Cisco-IOS-XR-infra-statsd-oper.yang \
   --tree-path infra-statistics/interfaces/interface/latest/generic-counters

module: Cisco-IOS-XR-infra-statsd-oper
  +--ro infra-statistics
    +--ro interfaces
      +--ro interface* [interface-name]
        +--ro latest
          +--ro generic-counters
            +--ro packets-received?                uint64
            +--ro bytes-received?                  uint64
            +--ro packets-sent?                    uint64
            +--ro bytes-sent?                      uint64
            +--ro multicast-packets-received?      uint64
             …
```

**https://github.com/mbj4668/pyang**

# 主要なSensor Path: システム

どういったデータ？

| Data | Model |
|---|---|
| Interface Oper State | Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-xr/interface |
| Interface Data Rate | Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/data-rate |
| Interfaces Stats | Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters |
| Optics Ports Info | Cisco-IOS-XR-controller-optics-oper:optics-oper/optics-ports/optics-port/optics-Info |
| Uptime Info | Cisco-IOS-XR-shellutil-oper:system-time/uptime |
| CPU State | Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization |
| Memory Info | Cisco-IOS-XR-nto-misc-oper:memory-summary/nodes/node/summary |
| Processes Memory | Cisco-IOS-XR-procmem-oper:processes-memory/nodes |
| NCS5500 NPU Stats | Cisco-IOS-XR-fretta-bcm-dpa-npu-stats-oper:dpa/stats/nodes/node |
| NCS5500 NPU Resources | Cisco-IOS-XR-fretta-bcm-dpa-hw-resources-oper:dpa/stats/nodes/node/hw-resources-datas/hw-resources-data |

# 主要なSensor Path: プロトコル

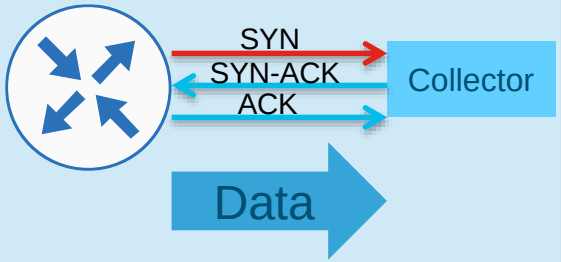
どういったデータ？

| Data | Model |
|---|---|
| LLDP Info | Cisco-IOS-XR-ethernet-lldp-oper:lldp/nodes/node/neighbors/summaries/summary |
| IPv4 RIB Info | Cisco-IOS-XR-ip-rib-ipv4-oper:rib/vrfs/vrf/afs/af/safs/saf/ip-rib-route-table-names/ip-rib-route-table-name/routes/route |
| IPv6 RIB Info | Cisco-IOS-XR-ip-rib-ipv6-oper:ipv6-rib/vrfs/vrf/afs/af/safs/saf/ip-rib-route-table-names/ip-rib-route-table-name/routes/route |
| BGP IPv4 Routes Info | Cisco-IOS-XR-ip-rib-ipv4-oper:rib/vrfs/vrf/afs/af/safs/saf/ip-rib-route-table-names/ip-rib-route-table-name/protocol/bgp/as/information |
| BGP IPv6 Routes Info | Cisco-IOS-XR-ip-rib-ipv6-oper:ipv6-rib/vrfs/vrf/afs/af/safs/saf/ip-rib-route-table-names/ip-rib-route-table-name/protocol/bgp/as/information |
| BGP IPv4 Neighbor | Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/neighbors/neighbor |
| MPLS-TE Tunnels | Cisco-IOS-XR-mpls-te-oper:mpls-te/tunnels/summary |
| RSVP Interface Info | Cisco-IOS-XR-ip-rsvp-oper:rsvp/interface-briefs/interface-brief |

# トランスポートのタイプ

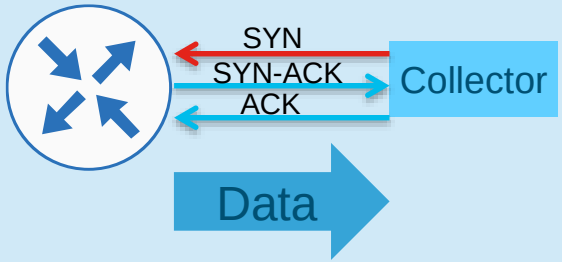
どこに送る？

## Dial-Out




- TCP & gRPC (from 6.1.1)
- UDP (from 6.2.1)

## Dial-In




- gRPC only (from 6.1.1)

# destination-group: 宛先

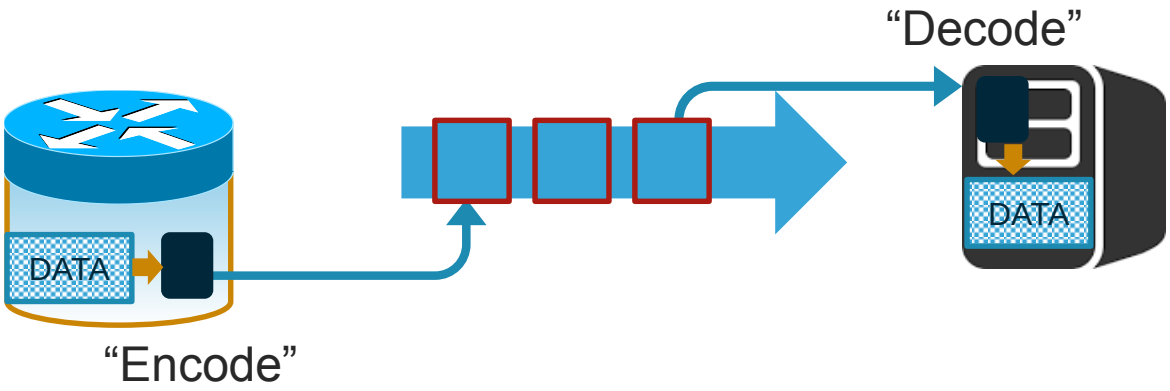
どこに送る？

```
telemetry model-driven

 destination-group DGROUP

  address family ipv4 192.168.1.1 port 2104

                  ---- and/or ----

  address family ipv6 2001:db8::1 port 2104

   encoding self-describing-gpb

   protocol tcp
```

＊Dial-In方式のときは ***address family ...*** 行を省略

# エンコーディング

どういった形式で？




## エンコーディング

- Compact GPB
- Self-describing GPB
- JSON (XR 6.3.1)

エンコーディングはデータをネットワーク経由で送信できる形式に変換
レシーバは元のデータと意味的に同一のコピーを作成するためにデコードを実施

# GPB compact vs. GPB self-describing

## GPB – “compact”

```
1: GigabitEthernet0/0/0/0
50: 449825
51: 41624083
52: 360333
53: 29699362
54: 91299
    <snip>
```

## GPB – “self-describing”

```
{InterfaceName: GigabitEthernet0/0/0/0
    GenericCounters {
      PacketsSent: 449825
      BytesSent: 41624083
      PacketsReceived: 360333
      BytesReceived: 29699362
      MulticastPacketsReceived: 91299
      <snip>
```

2倍速い
Operationを表すYANGモデル毎にメッセージ定義（.proto）ファイルが必要

3倍大きい
Telemetryヘッダのメッセージ定義（.proto）ファイルのみ

# （おまけ）ちょ、そこkwsk！

```
syntax = "proto3";
option go_package = "telemetry_bis";

/* Common Telemetry message */   // this is common for both
message Telemetry {
  oneof node_id {
    string node_id_str = 1;
    bytes node_id_uuid = 2;              // not used
  }
  oneof subscription {
    string   subscription_id_str     = 3;
    uint32   subscription_id         = 4; // not used
  }
  string   sensor_path               = 5; // not used
  string   encoding_path             = 6;
  string   model_version             = 7; // not used
  uint64   collection_id             = 8;
  uint64   collection_start_time     = 9;
  uint64   msg_timestamp             = 10;
  repeated TelemetryField data_gpbkv = 11;
  TelemetryGPBTable data_gpb         = 12;
  uint64   collection_end_time       = 13;
  uint64   heartbeat_sequence_number = 14; // not used
}
```

```
/* KV GPB specific payload definition */
message TelemetryField {
  uint64         timestamp = 1;
  string         name      = 2;
  oneof value_by_type {
    bytes          bytes_value   = 4;
    string         string_value  = 5;
    bool           bool_value    = 6;
    uint32         uint32_value  = 7;
    uint64         uint64_value  = 8;
    sint32         sint32_value  = 9;
    sint64         sint64_value  = 10;
    double         double_value  = 11;
    float          float_value   = 12;
  }
  repeated TelemetryField fields = 15;
}
/* (Compact) GPB specific payload definition */
message TelemetryGPBTable {
  repeated TelemetryRowGPB row = 1;
}
message TelemetryRowGPB {
   uint64 timestamp = 1;
   bytes  keys      = 10;
   bytes  content   = 11;
}
```

**https://github.com/cisco/bigmuddy-network-telemetry-proto/blob/master/staging/telemetry.proto**

# Subscription: 全てをまとめる

どういった頻度で？

```
telemetry model-driven
 subscription SUB1
  sensor-group-id SGROUP1 sample-interval 30000
  destination-id DGROUP1
```

# 実装状況まとめ

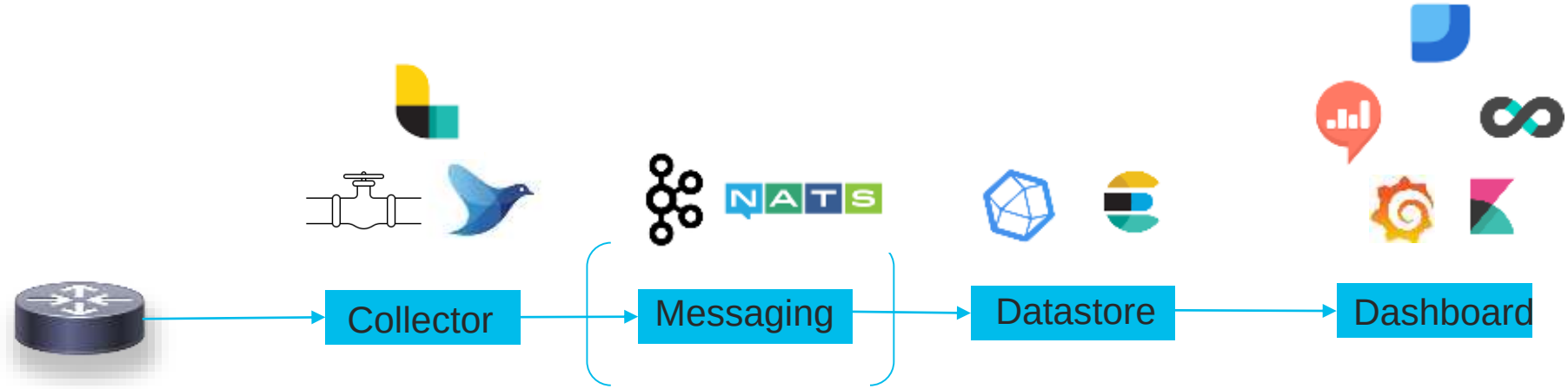
| | サポート | データモデル | トランスポート | エンコーディング | Data Plane Telemetry |
|---|---|---|---|---|---|
| **IOS XR** | 6.1.1 | Native(YANG)<br>OpenConfig(YANG) | TCP<br>UDP<br>gRPC | JSON<br>GPB | ✓ * |
| **NXOS** | 7.3(0)I5(1) | Native(not YANG)<br>Native(YANG) * | HTTP<br>gRPC<br>UDP * | JSON<br>GPB | ✓ * |
| **IOS XE** | 16.6.1 | Native(YANG)<br>OpenConfig(YANG) | Netconf | XML | |



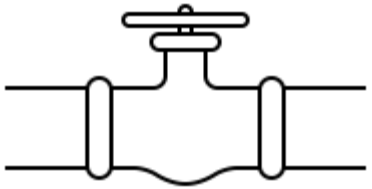
*** Roadmap**

# ツールチェーン

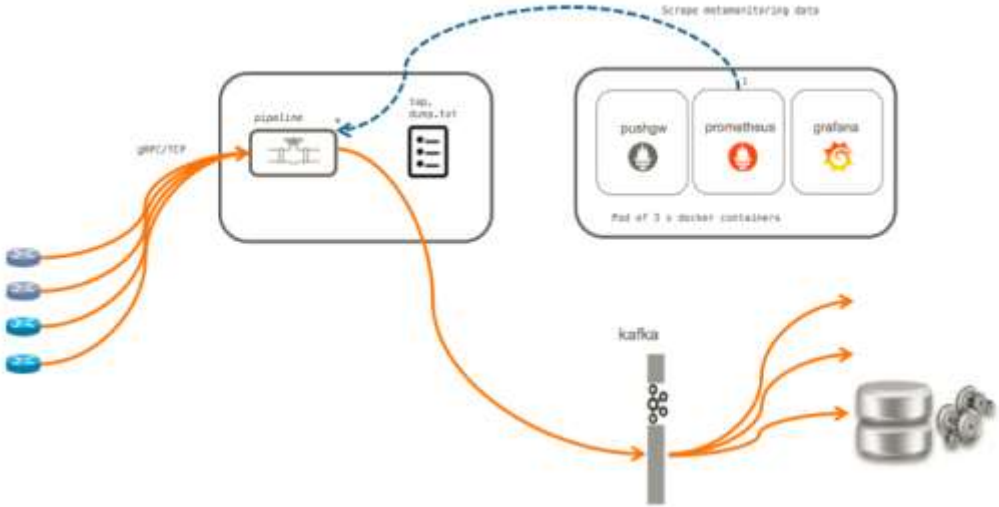
# OSS活用の全体像



# Pipeline



- オープンソース軽量テレメトリレシーバ
  - Go実装
  - 入力
    - プロトコル
      - UDP, TCP, gRPC
    - エンコーディング
      - GPB, GPB key-value(self-describing)
  - 出力
    - InfluxDB, Apache Kafka, Prometheus pushgateway
  - IOS-XR、NXOSに対応




https://github.com/cisco/bigmuddy-network-telemetry-pipeline

# Logstashプラグイン

- LogstashのCodecプラグイン
- 豊富なLogstash Outputプラグイン(Elasticksearch, Kafka, etc)と組み合わせて使える

  https://github.com/cisco/logstash-codec-bigmuddy-network-telemetry (JSON encodingに対応)

  https://github.com/cisco/logstash-codec-bigmuddy-network-telemetry-gpb

- PrometheusやSignal FXのためにHTTPでメトリック情報を提供するOutputプラグイン

  https://github.com/cisco/logstash-output-bigmuddy-network-telemetry-metrics

- ELK,Prometheus,Signal FX,Apache出力にそれぞれに対応した設定済みのDockerベースのスタックが存在（あくまでデモ用途、サンプル設定として参考になる）

  https://github.com/cisco/bigmuddy-network-telemetry-stacks

# Fluentdプラグイン(Cisco非公式)

- **fluent-plugin-telemetry-iosxr**

https://github.com/tetsusat/fluent-plugin-telemetry-iosxr

https://rubygems.org/gems/fluent-plugin-telemetry-iosxr

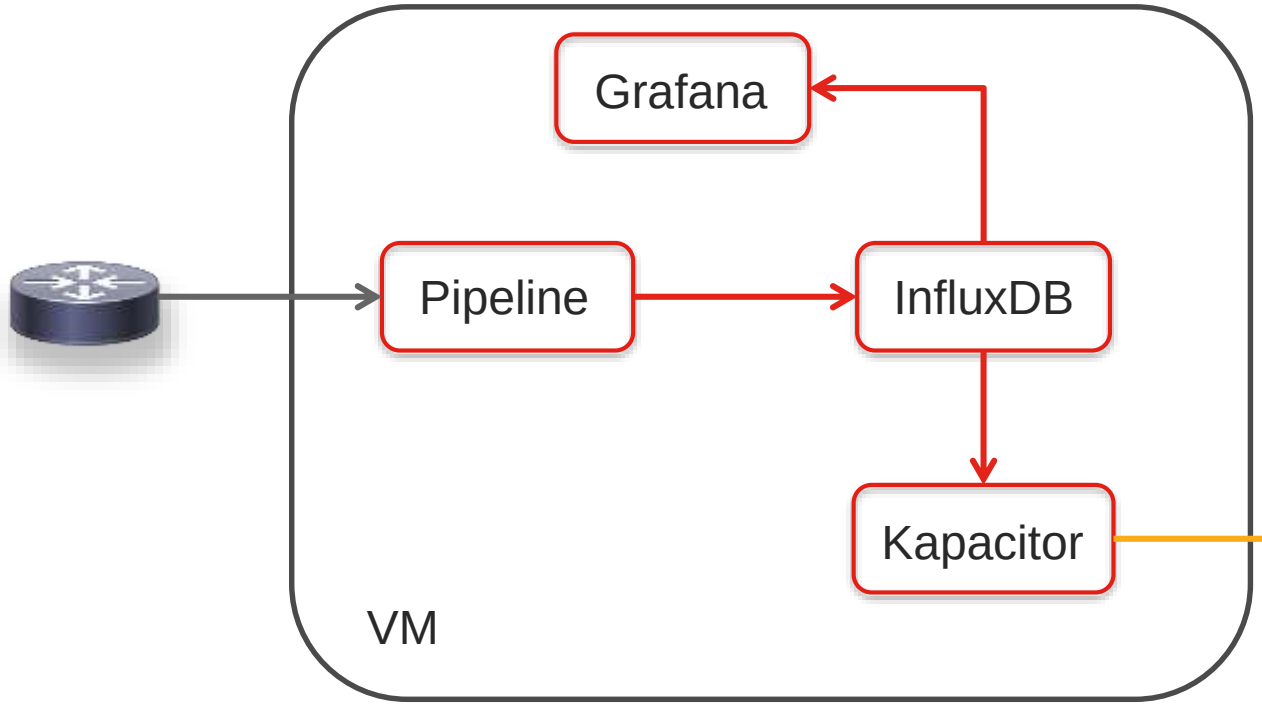
- **fluent-plugin-telemetry-iosxe**

https://github.com/tetsusat/fluent-plugin-telemetry-iosxe

https://rubygems.org/gems/fluent-plugin-telemetry-iosxe

# デモ

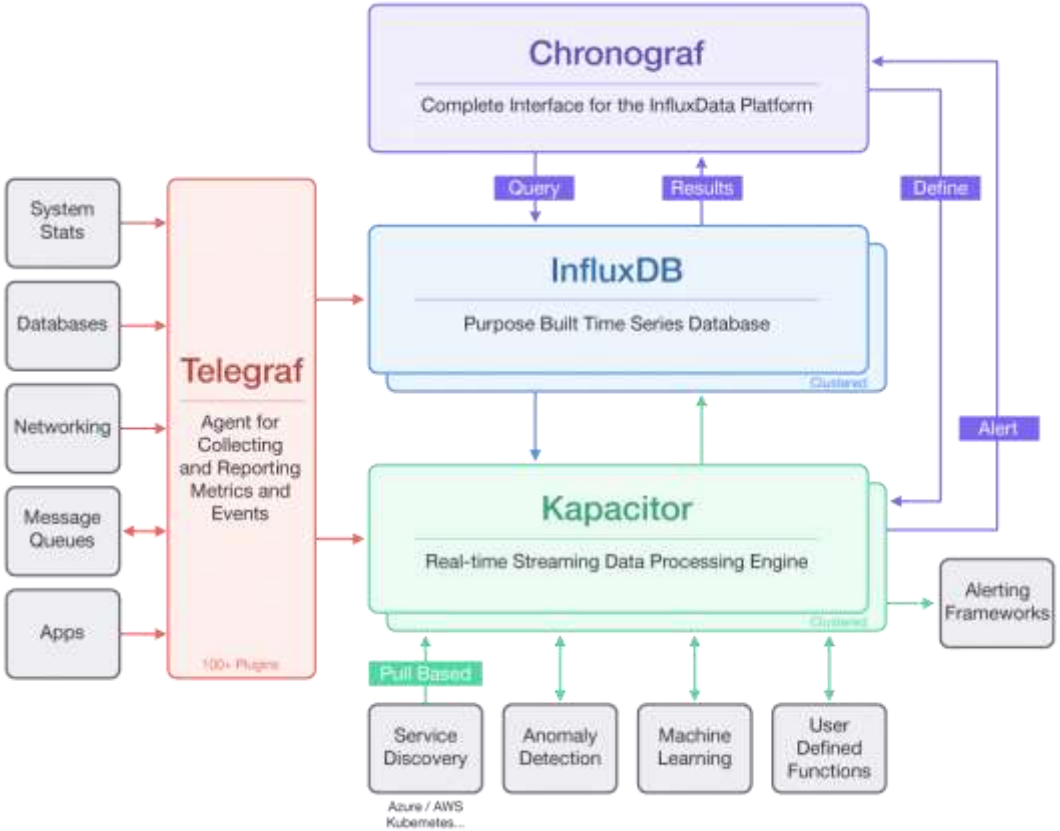
# デモ


slack

# Kapacitor

- オープンソースの時系列データ向けのデータ処理エンジン
- 独自DSLのTICKScriptで簡単にアラートを作成
- HipChat、Slackなどと統合



# 参考

**Kapacitorアラート設定(cpu_alert.tick)**

```
dbrp "telemetry"."autogen"

stream
    | from()
        .measurement('Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization')
        .where(lambda: "node-name" == '0/RP0/CPU0')
    | alert()
        .info(lambda: "total-cpu-one-minute" > 5)
        .warn(lambda: "total-cpu-one-minute" > 10)
        .crit(lambda: "total-cpu-one-minute" > 15)
        .stateChangesOnly()
        .message('{{ .Level }}: CPU utilization is {{ index .Fields "total-cpu-one-minute" }}%')
        .log('/tmp/alerts.log')
        .slack()
```

# 参考

## Kapacitorサービス設定

```
...
[slack]
  enabled = true
  url = "https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXX"
  channel = "#demo"
  username = ""
  icon-emoji = ""
  global = false
  state-changes-only = false
  ssl-ca = ""
  ssl-cert = ""
  ssl-key = ""
  insecure-skip-verify = false
...
```

# 参考

- Everything you need to know about Pipeline
  https://xrdocs.github.io/telemetry/tutorials/2018-03-01-everything-you-need-to-know-about-pipeline/


CISCO