GMO INTERNET

DNSのマルチリージョン監視をやってみた

DNSOPS.JP BoF
2018/11/29
GMOインターネット株式会社
永井祐弥
藤井勇太

目次

- 自己紹介
- イントロダクション
- 構成
- 設計
- 今後の課題
- 質疑応答

自己紹介

■ 名前

永井　祐弥（ながい　ゆうや）

■ 所属

GMOインターネット株式会社
システム本部 インフラサービス開発部

■ 担当

2012年にGMOインターネット株式会社へ入社。
お名前.com、ConoHa、Z.comのDNSや、
GMOインターネットグループ会社でレジストリシステムの
DNSなど、DNS関連サービスの開発、運用を担当

自己紹介

- 名前: 藤井勇太
- 経歴: 2014年にGMOインターネット株式会社へ入社
  以降、社内WEB/メールシステムの構築から運用保守を担当
  2017年より、DNS周りのシステム担当として運用保守を担当

- DNSOPS.JPは3年前ほどから聴くほうで参加
  今日、初めて発表!!

イントロダクション

- DNSのサービス監視ってどうしていますか？
  - システム内監視
    - LBなど冗長構成の内側からのDNS応答を監視
    - 冗長構成の外側からのDNS応答を監視
  - 外形監視
    - バックボーンの外側からのDNS応答を監視
  - SNS監視
    - サービスに対するコメントなど、情報の監視

イントロダクション

- 外形監視で困ったこと
  - 弊社のDNSサービスは現在4ヶ国（日本、シンガポール、タイ、ベトナム）にDNSサーバーを設置している
  - 国際回線を経由する外形監視では途中経路に問題が発生すると影響を受けてしまうことがある
    - 特に一部東南アジア方面
  - 途中経路の回線品質に影響されないようにDNSサービスを監視する方法は無いものか？
    - そういえば過去の発表に、、、

イントロダクション

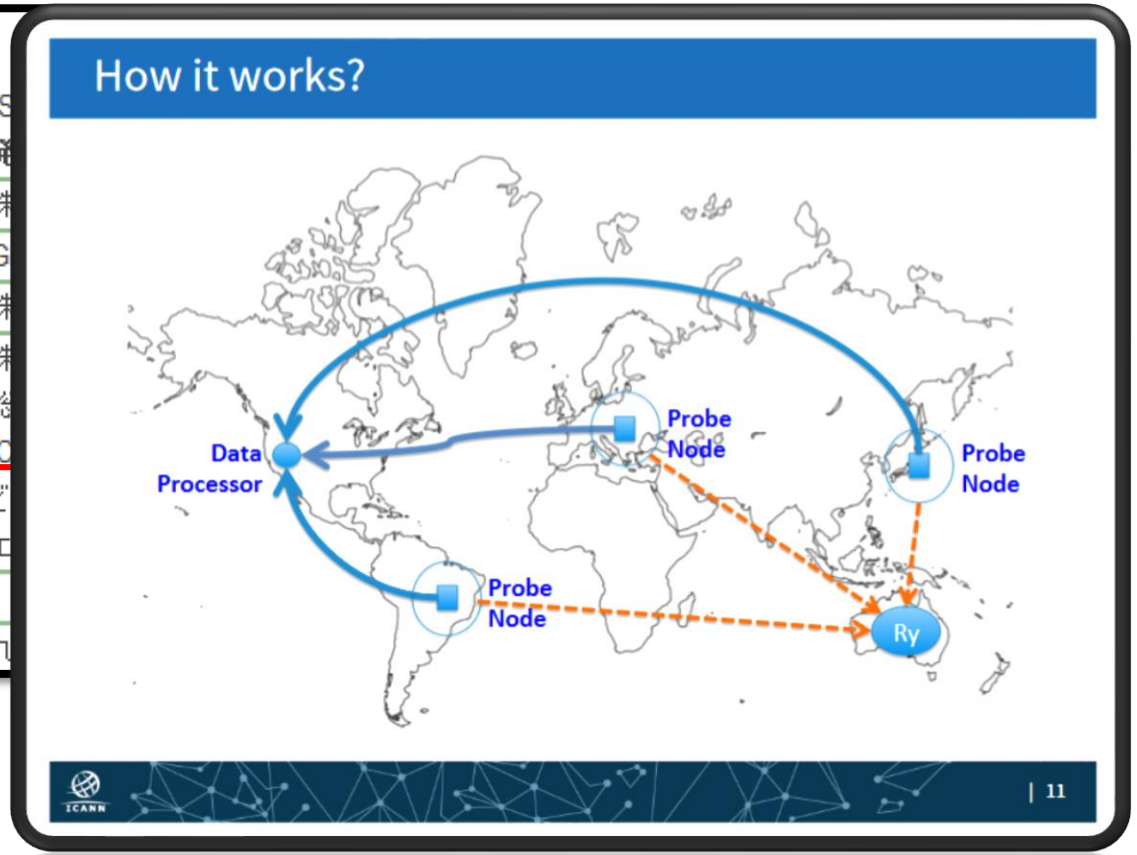
- 振り返ること DNS Summer Day 2015

午後(ワークショップ) 12:45-18:00

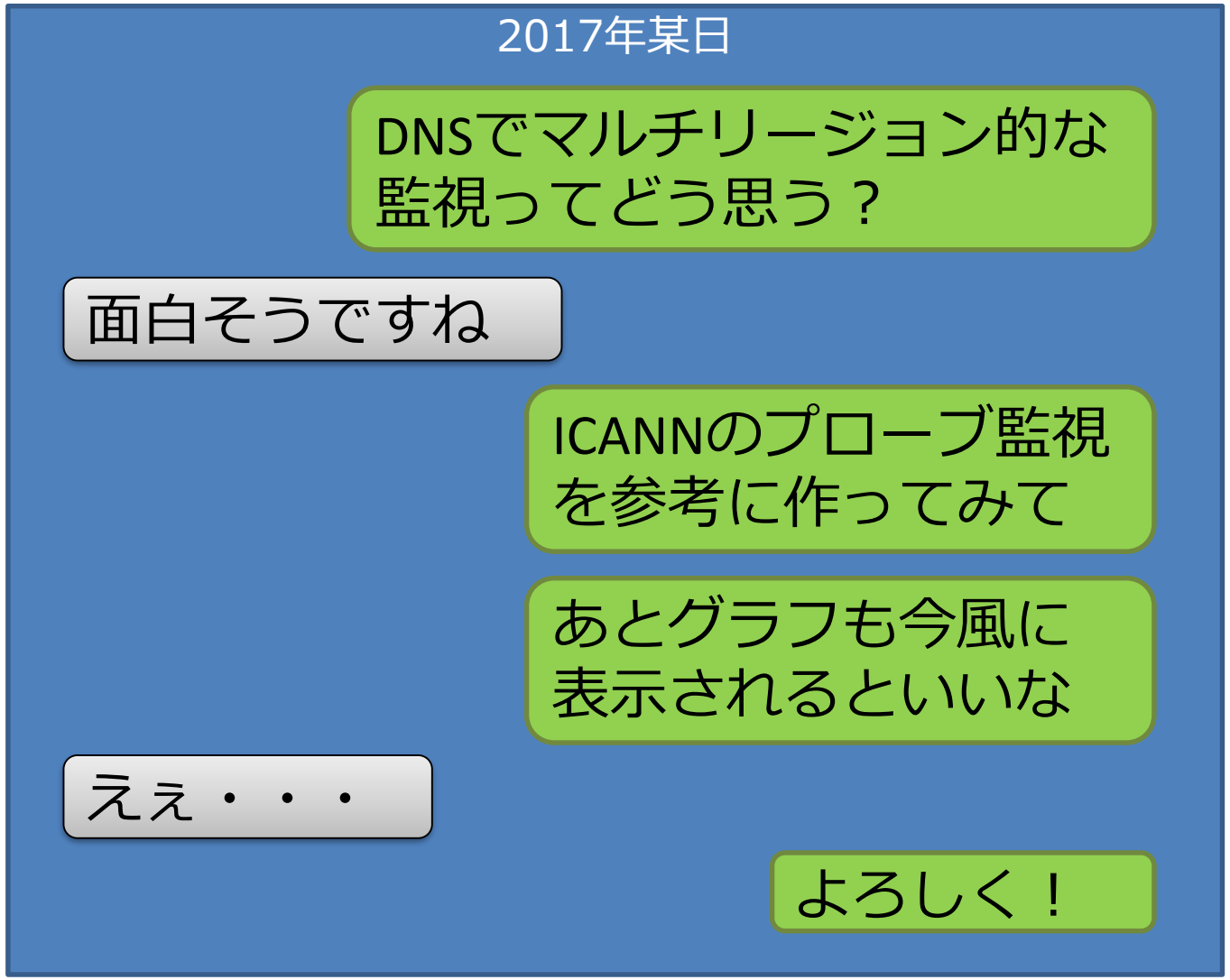
対象:DNSの運用に関わる技術者(各組織のDNSに関わるICT担当者、xS

時間	タイトル	発
12:45-LT:	WindowsDNSサーバと社内情シス担当者	株
13:00-LT:	OpenStack Designateで作るDNSaaS	G
13:10-LT:	権威DNSサーバーってどこにあるの？～日本が孤立したら～	株
		株
		総
13:20-	DNSのサービスレベル	IC
		ビ
		エ
	対策と動向 (仮)	九

これだ！



イントロダクション



GMO INTERNET

イントロダクション

あとよろ！

「あとよろ！」されてしまった

そもそも、
プローブ(Probe)とはなんだろう？

Probeを使用しての監視

- probeとは
  - (傷・穴などの深さを調べる)探り針、ゾンデ、厳密な調査、徹底的な調査、宇宙探査用装置
    ※ weblioから引用 https://ejje.weblio.jp/content/probe
- 様々なネットワーク上に設置されたProbeの他視点なデータを元に監視
  - ICANN probe
  - RIPE Atlas(DNSMON)
- 留意点として、IP Anycastの構成では意図したノードにDNSクエリが到達するとは限らないため収集するデータに偏りがでる

その他の監視

- プライベートネットワーク内の監視
  - ローカルサーバーや、監視サーバーからの監視
  - 外部からの監視とは別で、切り分けの為この監視も必要
- IP spoofingを利用した監視
  - RIPE72(2016年)で、当時Dyn社(現Oracle社)が IP AnycastのDNS測定方法として発表されたもの
  - 監視対象となるローカルサーバーで、ソースIPアドレスを監視サーバーのIPアドレスに偽装したDNSクエリを送信し、監視サーバー側でレスポンスを受け取り監視

A New Anycast DNS Measurement
https://ripe72.ripe.net/presentations/75-dknight-lightning-a_new_anycast_dns_measurement.pdf

構想

構想

- 今回は実験的な試み
  - 業務と業務の合間に開発

- 既存のサーバーリソース内で開発
  - 各リージョンにある管理用サーバーに設置

構想

・Probeを使用しての監視システムを
開発するにあたり必要な機能

①名前解決の結果を取得 → digコマンドの結果をパースして整形するだけなら自力で書けそう

②結果データを記録保存
③データをグラフ化
→ 自力では無理
OSSのちからを借りよう

構想

②結果データを記録保存 → influxdb

③データをグラフ化 → Grafana

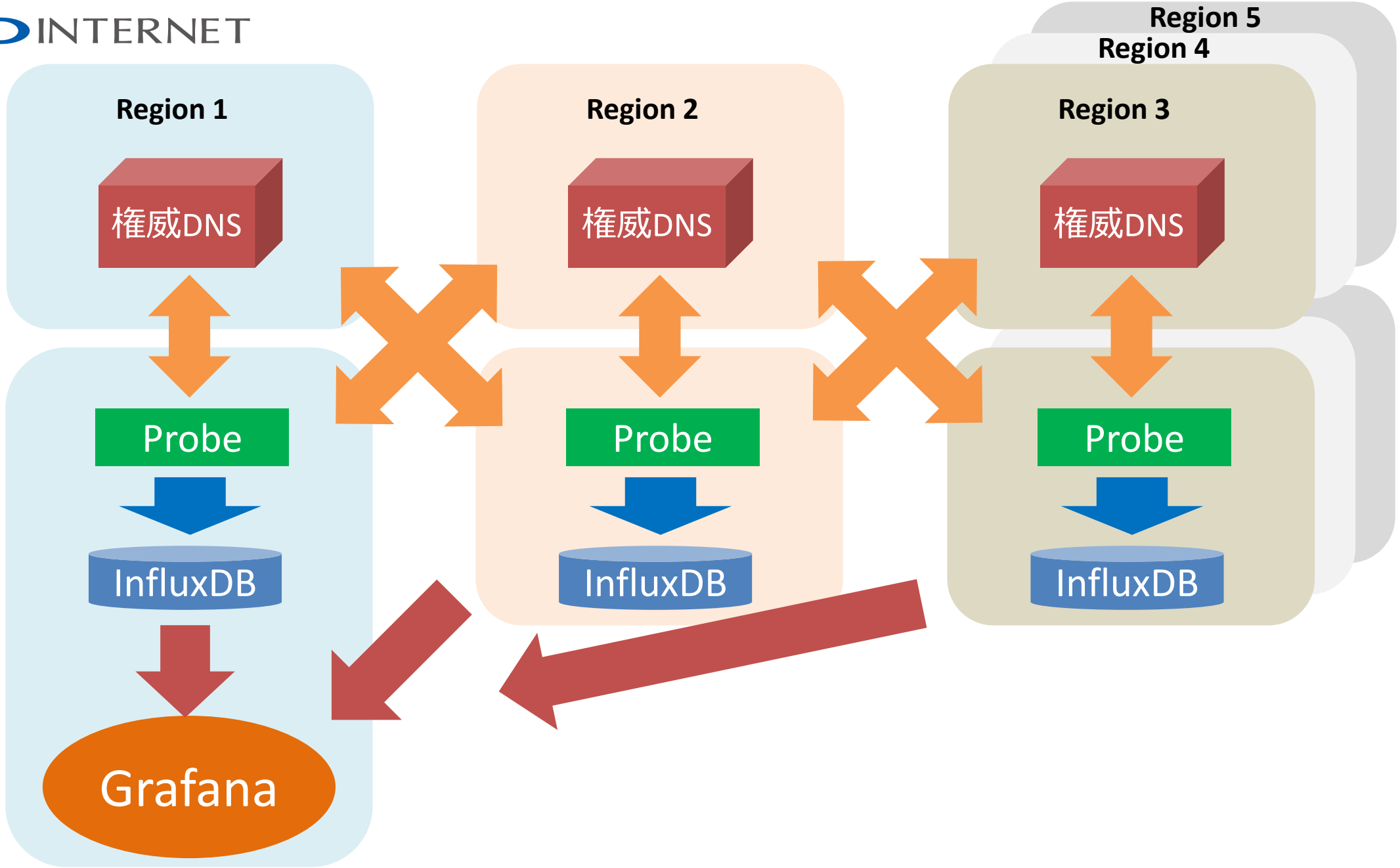
- 時系列DB(time series database)
- 開発言語: Go
- ライセンス: MIT License
- 特徴:
  - SQLライクなCLIツール
  - REST API
  - スキーマレス

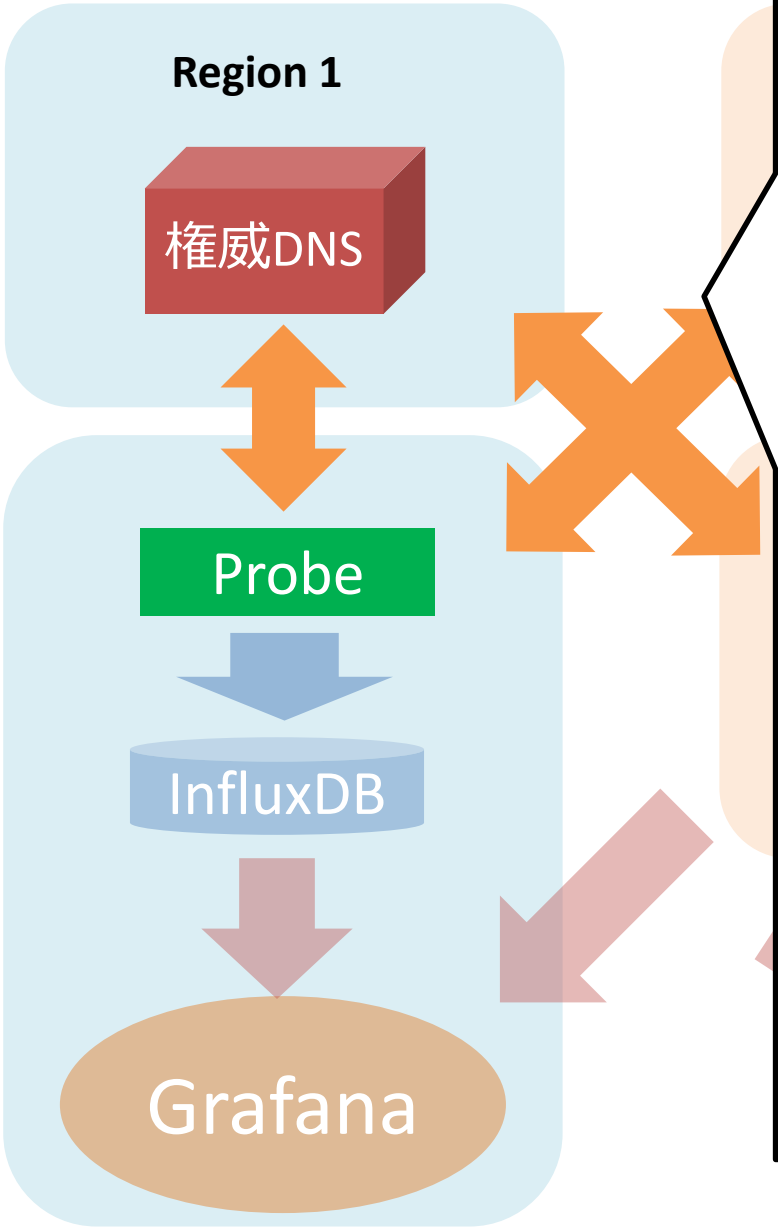
- 複数DB(Graphite, influxDB, Elasticsearchなど)に対応したメトリクス可視化、モニタリング機能を備えるOSS
- 開発言語: Go, NodeJS
- ライセンス: Apache License 2.0
- 特徴:
  - 用途に応じたPluginで様々な表示が可能
  - データに基づいたアラート(Email,slack,LINEなど)通知も可能

構想

- 利用ソフトウェア選定のポイント
  - CPU,Memory,Diskなどリソースが少なくても使えるもの
  - 事例やドキュメントが多いもの

設計

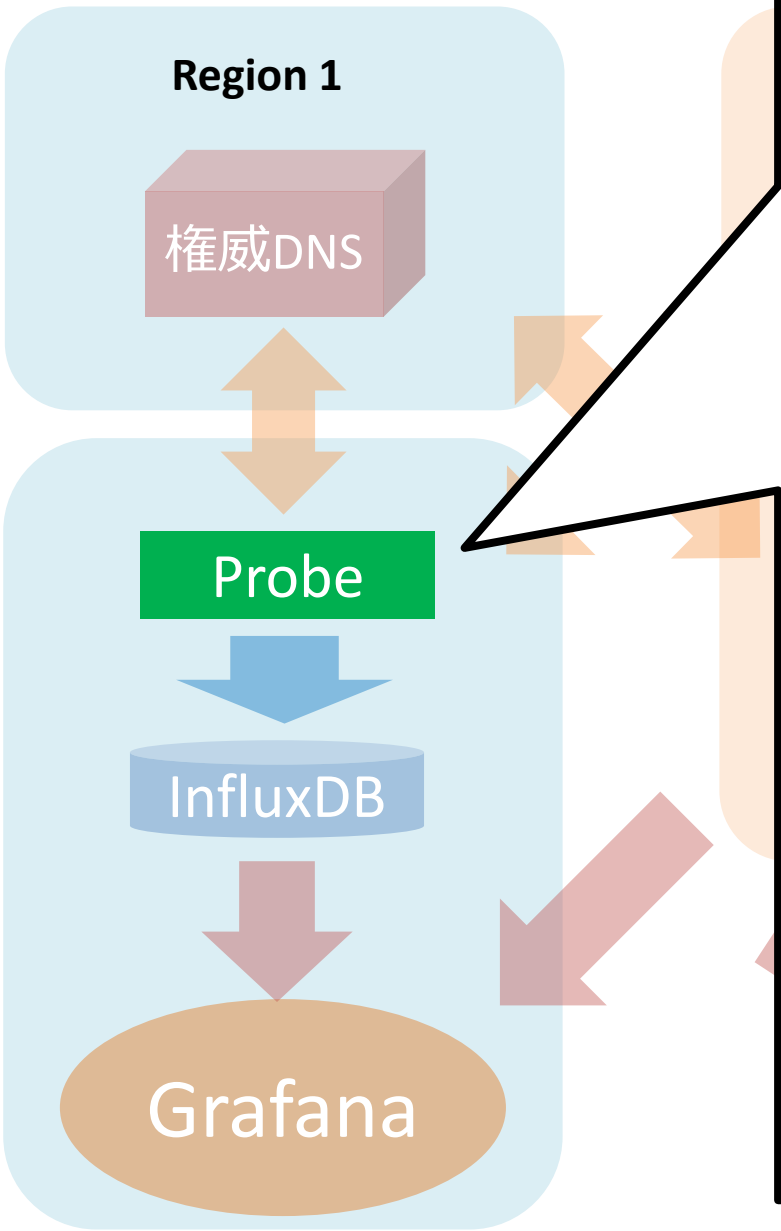




- 名前解決
  digコマンドで

(例) example.com/SOAレコードの名前解決

```
$ dig +norec -t soa example.com @ns1.example.com

; <<>> DiG 9.13.3 <<>> +norec -t soa example.com @ns1.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60222
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com.                   IN      SOA

;; ANSWER SECTION:
example.com.            3600    IN      SOA     sns.dns.icann.org. noc.dns

;; Query time 108 msec
```



- レスポンスの結果をパースして整形

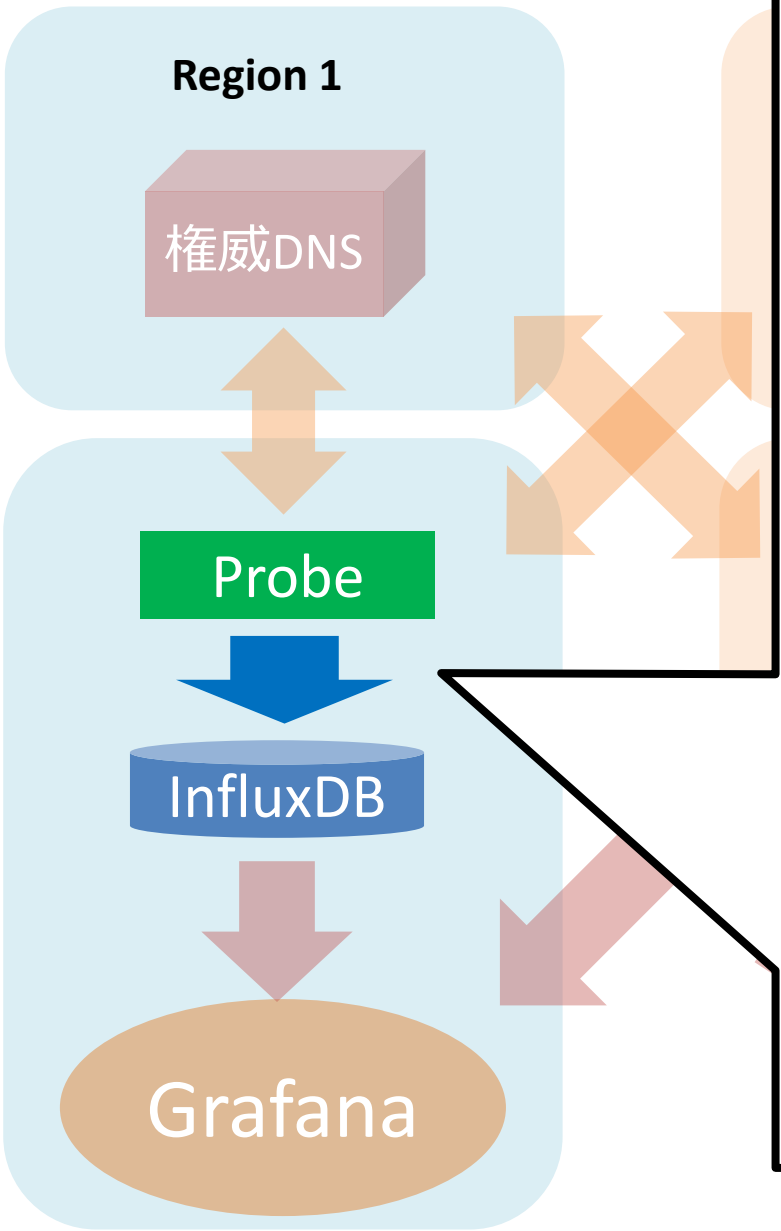
```
$ dig +norec -t soa example.com @ns1.example.com

; <<>> DiG 9.13.3 <<>> +norec -t soa example.com @ns1.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR,
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY:

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com.                   IN      SOA

;; ANSWER SECTION:
example.com.            3600    IN      SOA

;; Query time 108 msec
```

```
$ dig +norec -t soa example.com @ns1.example.com

; <<>> DiG 9.13.3 <<>> +norec -t soa example.com @ns1.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60222
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com.                   IN      SOA

;; ANSWER SECTION:
example.com.            3600    IN      SOA     sns.dns.icann.org. noc.dns

;; Query time 108 msec
```

(例)

curl -ks -X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-a,probe=hoge, target=**ns1.example.com**, ¥
transport=**4**, nsid=a rcode=**0**, querytime=**53**

curl -ks -X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-a,probe=hoge, target=**ns2.example.com**, ¥
transport=**6**, nsid=b rcode=**0**, querytime=**35**



- REST APIでInfluxDBにデータを登録

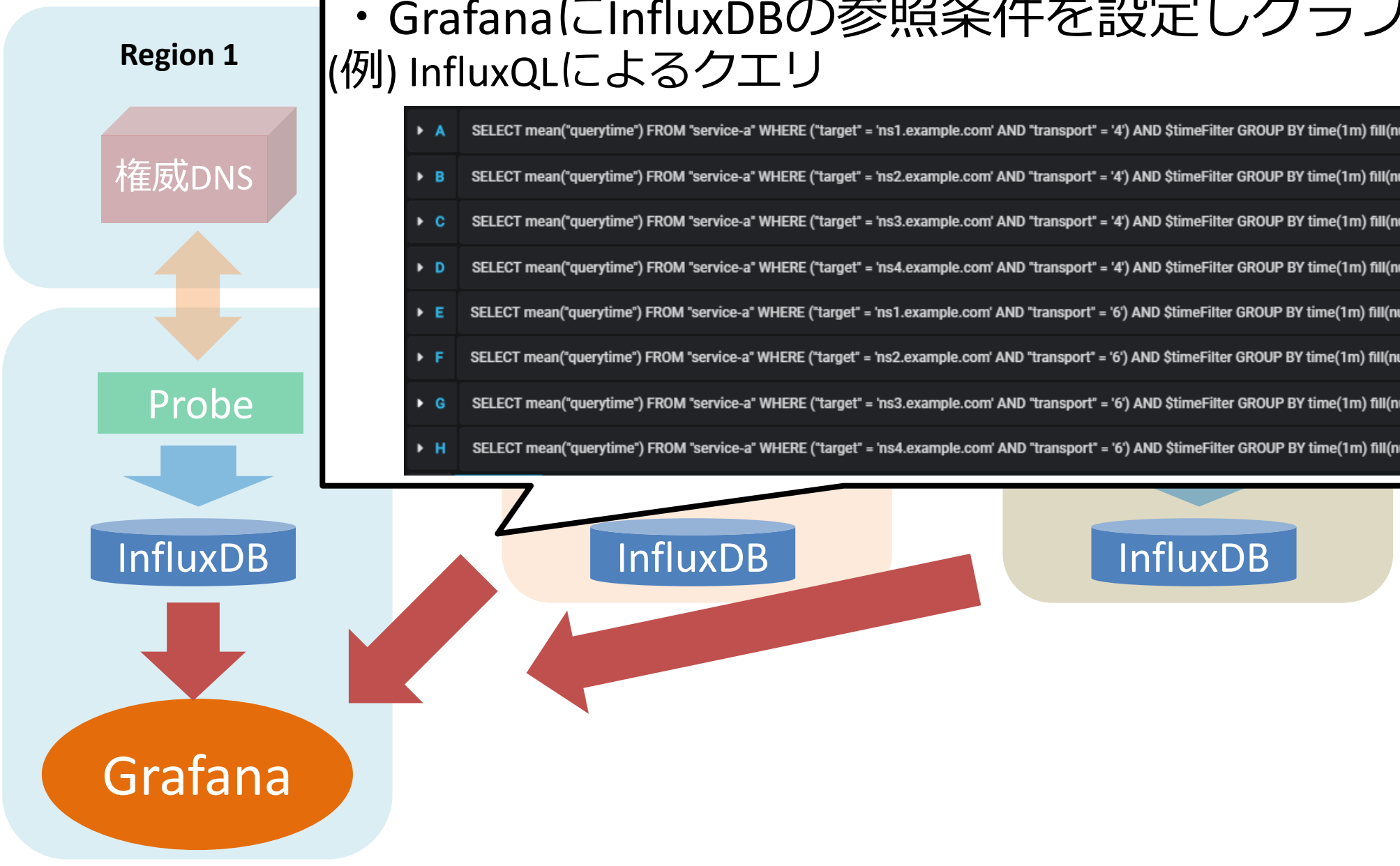
(例)

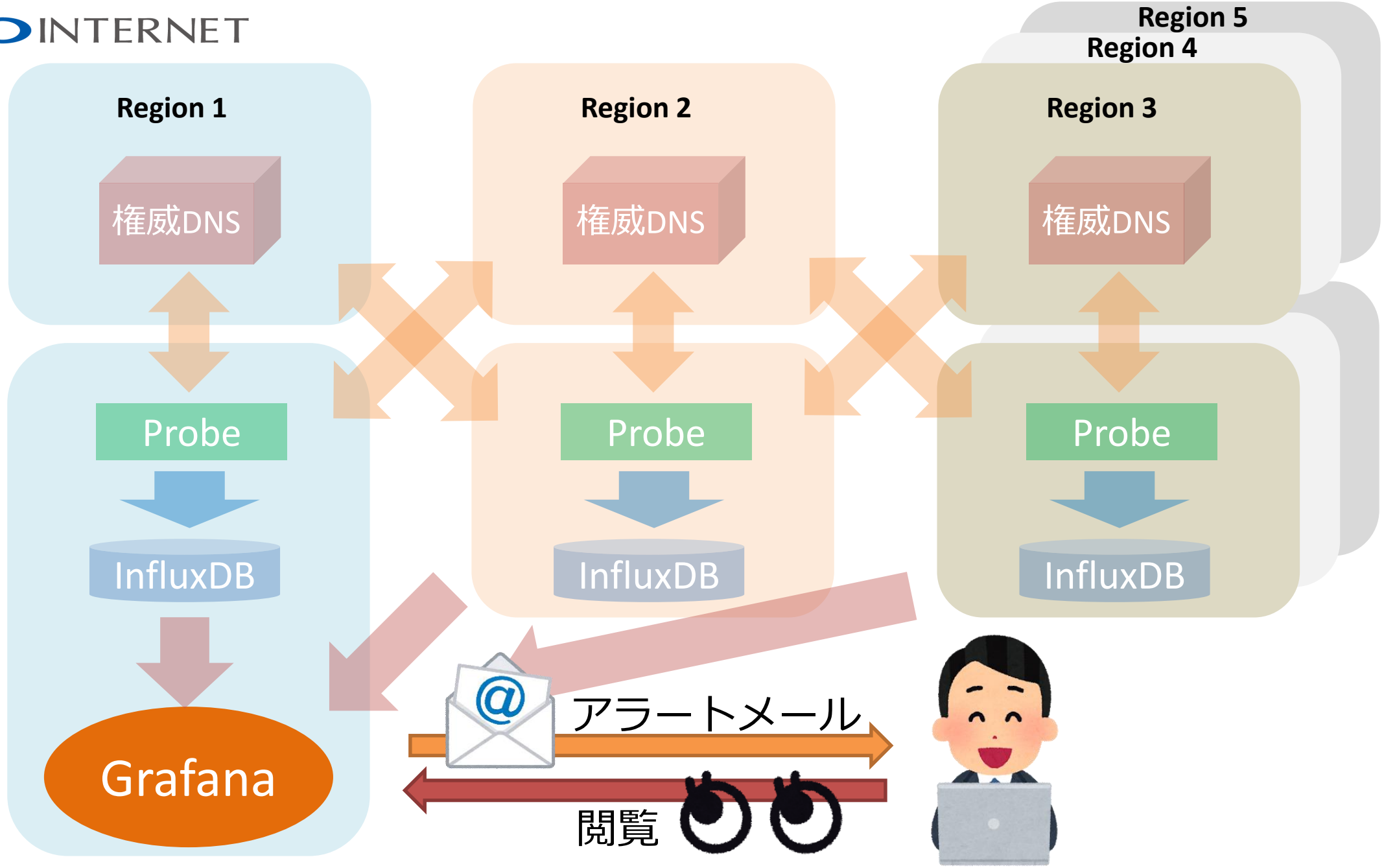
```
curl -ks -X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-a,probe=hoge,target=01.example.jp,transport=4,nsid=a rcode=0,querytime=4
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-a,probe=hoge,target=02.example.jp,transport=4,nsid=b rcode=0,querytime=1
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-a,probe=hoge,target=03.example.jp,transport=4,nsid=c rcode=0,querytime=10
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-a,probe=hoge,target=04.example.jp,transport=4,nsid=d rcode=0,querytime=8
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-a,probe=hoge,target=03.example.jp,transport=6,nsid=e rcode=0,querytime=2
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-a,probe=hoge,target=04.example.jp,transport=6,nsid=f rcode=0,querytime=2
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-b,probe=hoge,target=ns1.example.com,transport=4 rcode=0,querytime=5
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-b,probe=hoge,target=ns1.example.com,transport=6 rcode=0,querytime=21
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-b,probe=hoge,target=ns1.example.jp,transport=4 rcode=0,querytime=53
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-c,probe=hoge,target=ns1.example.jp,transport=4, rcode=0,querytime=105
curl –ks –X POST https://127.0.0.1:8086/write?db=local_dig ¥
--data-binary service-c,probe=hoge,target=ns2.example.jp,transport=4 rcode=0,querytime=10
…
```

・GrafanaにInfluxDBの参照条件を設定しグラフ化

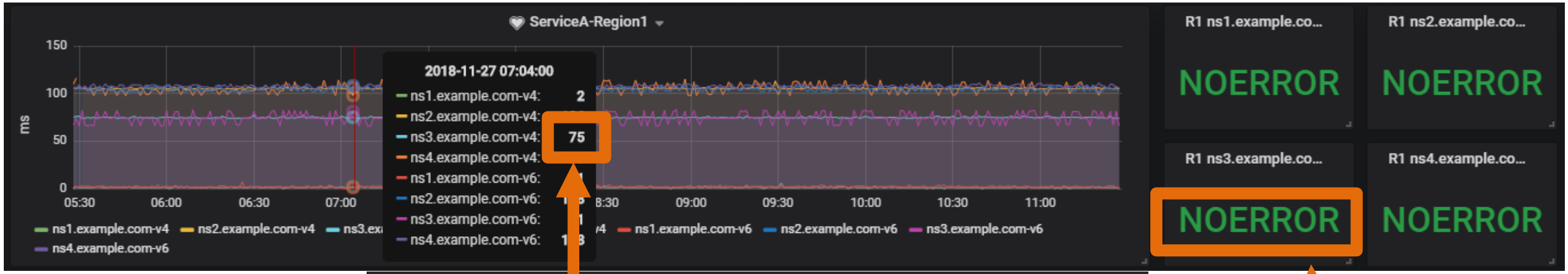
(例) InfluxQLによるクエリ

A	SELECT mean("querytime") FROM "service-a" WHERE ("target" = 'ns1.example.com' AND "transport" = '4') AND $timeFilter GROUP BY time(1m) fill(null)
B	SELECT mean("querytime") FROM "service-a" WHERE ("target" = 'ns2.example.com' AND "transport" = '4') AND $timeFilter GROUP BY time(1m) fill(null)
C	SELECT mean("querytime") FROM "service-a" WHERE ("target" = 'ns3.example.com' AND "transport" = '4') AND $timeFilter GROUP BY time(1m) fill(null)
D	SELECT mean("querytime") FROM "service-a" WHERE ("target" = 'ns4.example.com' AND "transport" = '4') AND $timeFilter GROUP BY time(1m) fill(null)
E	SELECT mean("querytime") FROM "service-a" WHERE ("target" = 'ns1.example.com' AND "transport" = '6') AND $timeFilter GROUP BY time(1m) fill(null)
F	SELECT mean("querytime") FROM "service-a" WHERE ("target" = 'ns2.example.com' AND "transport" = '6') AND $timeFilter GROUP BY time(1m) fill(null)
G	SELECT mean("querytime") FROM "service-a" WHERE ("target" = 'ns3.example.com' AND "transport" = '6') AND $timeFilter GROUP BY time(1m) fill(null)
H	SELECT mean("querytime") FROM "service-a" WHERE ("target" = 'ns4.example.com' AND "transport" = '6') AND $timeFilter GROUP BY time(1m) fill(null)





(例)



(例)

```
$ dig +norec -t soa example.com @ns3.example.com

; <<>> DiG 9.13.3 <<>> +norec -t soa example.com @ns3.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60222
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com.                   IN      SOA

;; ANSWER SECTION:
example.com.            3600    IN      SOA     sns.dns.icann.org.

;; Query time: 75 msec
```

QueryTime

RCODE

digコマンドの結果をまとめて見れることを重視した表示設定

今後の課題

- Grafanaの耐障害性向上
  - 現在はシングル構成のため障害に弱い
  - 将来的には冗長構成に変更したい

- アラートの精度向上
  - 1 Probeで異常を検知するとアラートが発生する
  - これを複数Probe（全体の51%以上）で異常検知している場合にアラートを発生させたい

質疑応答

- 発表のここの所、もう少し詳しく！
- 私の所だと、こんなことやっているよ！
- いやいや、ここはこんな風にやってみたら？
- など

疑問、質問、コメントなど是非！！

すべての人にインターネット

GMO