

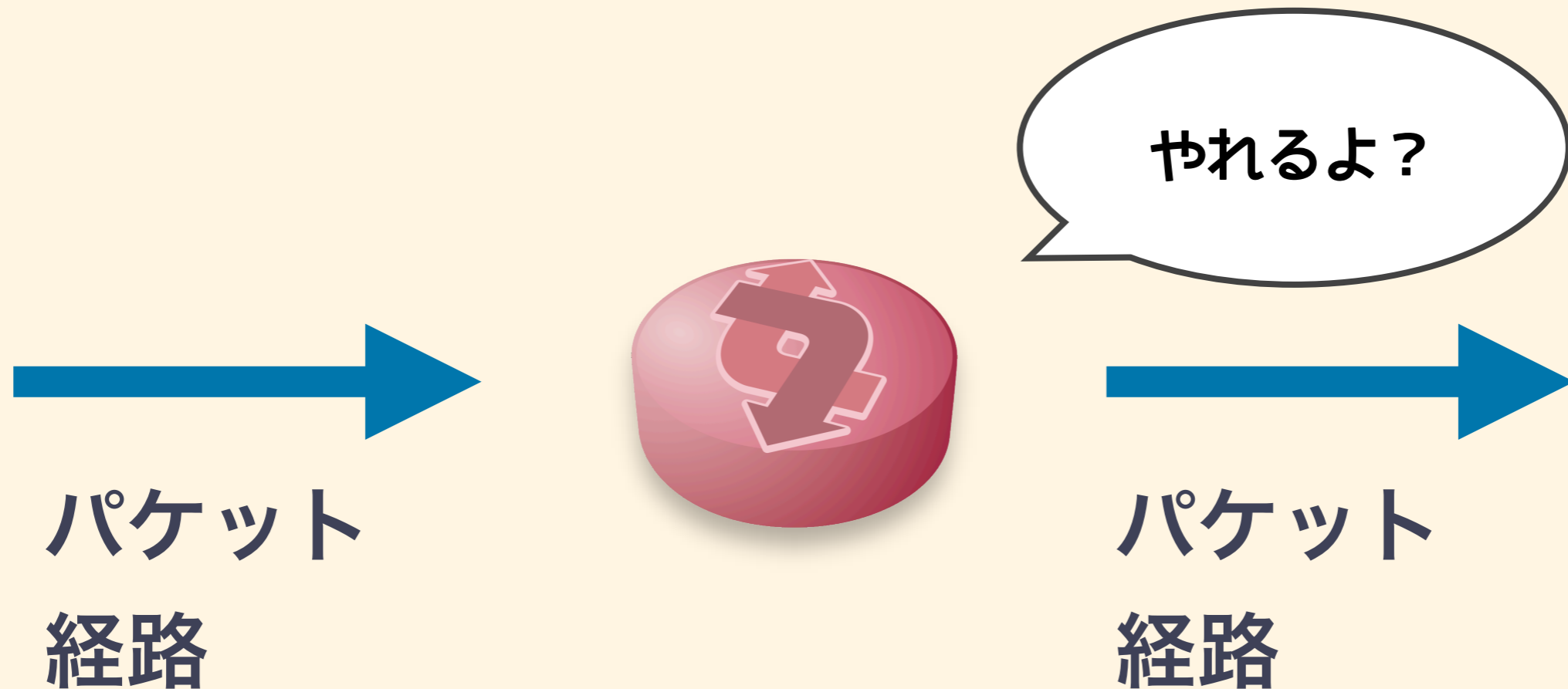
Event Driven BGPD

BGP によるコナミコマンド

Shintaro Kojima
コーダンス/@codeout

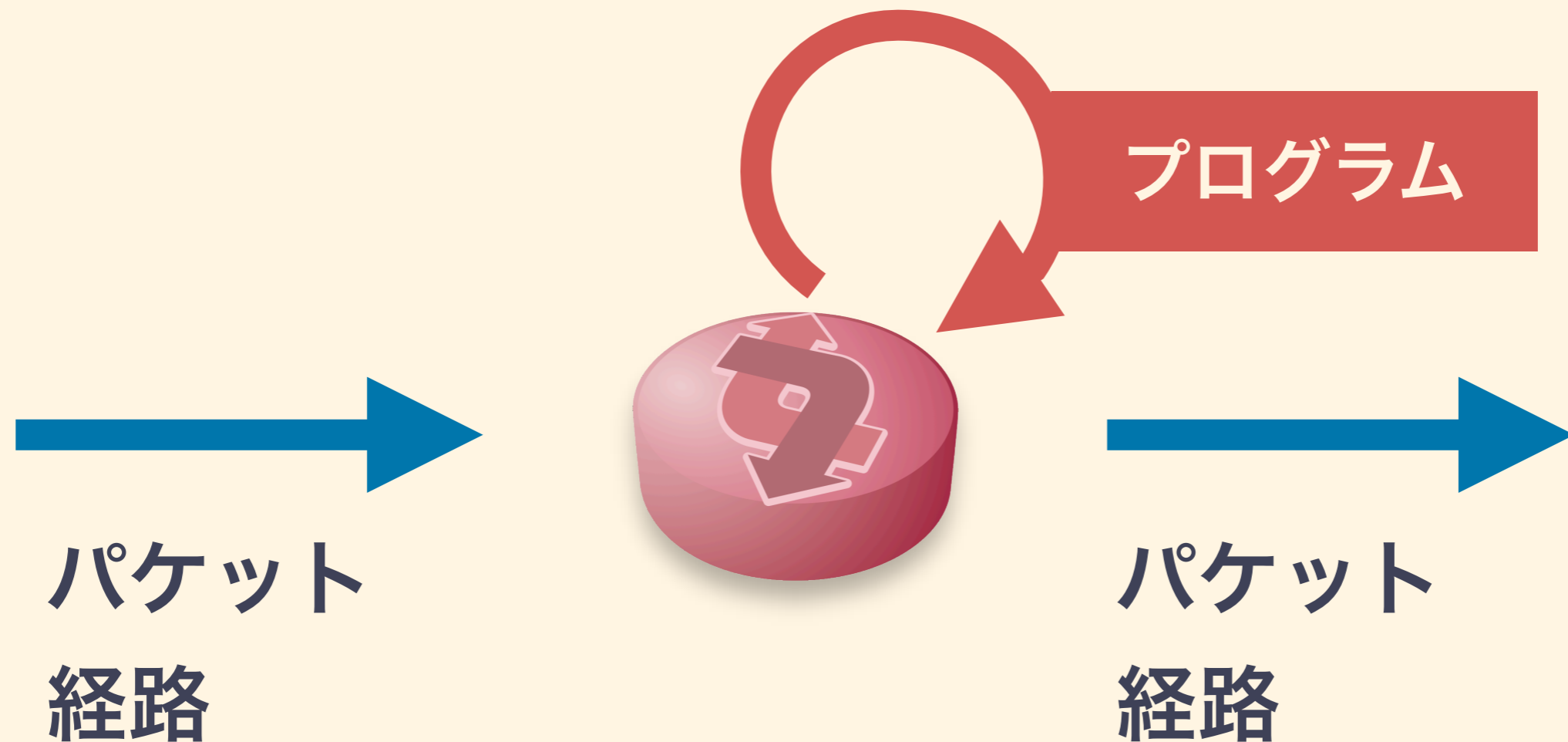


Event Driven ?



「入力に対して何かする」という意味では
もともとルーターは Event Driven

Event Driven ?



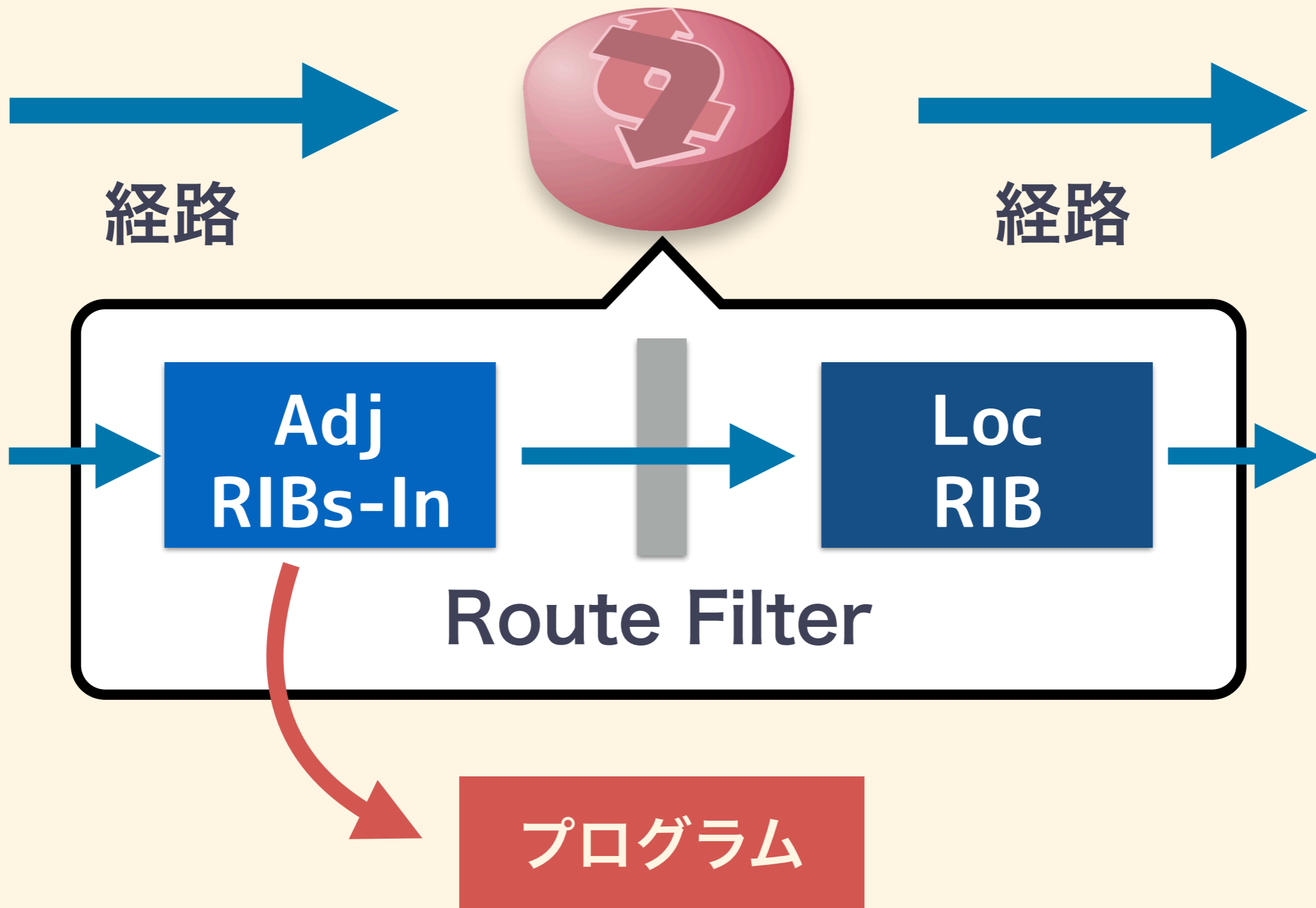
それでは不十分で、入力に対して
任意のプログラム実行 をはさみたい

Event Driven ?

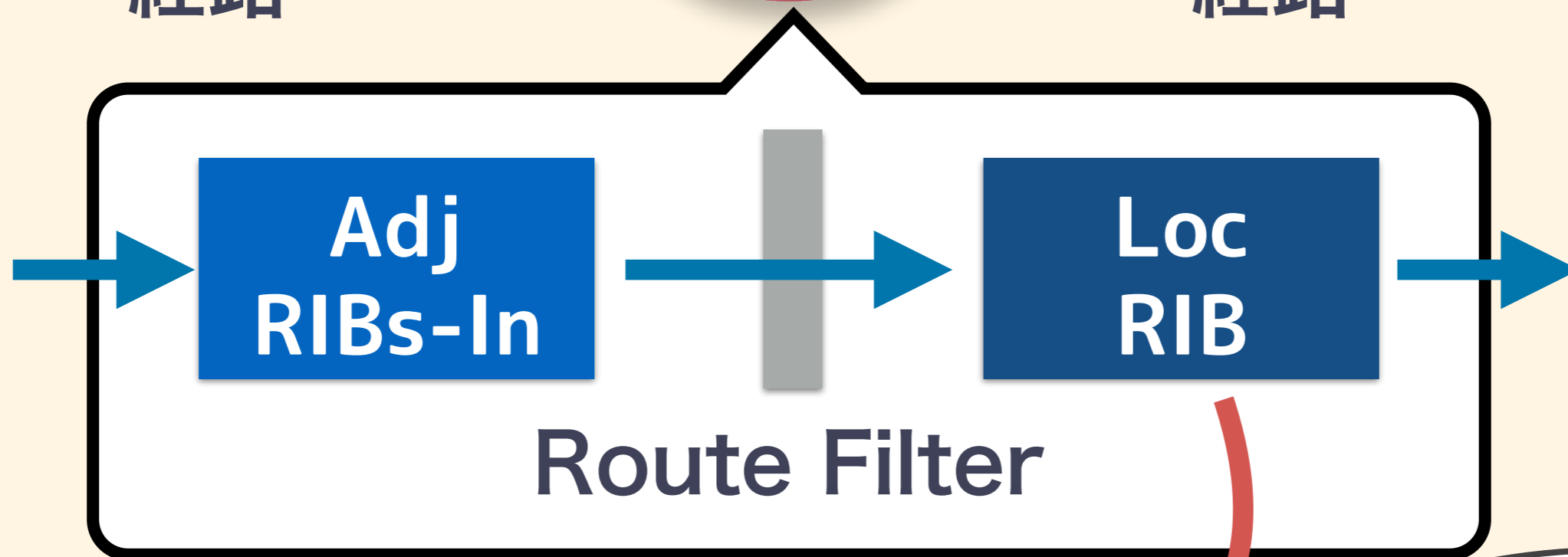
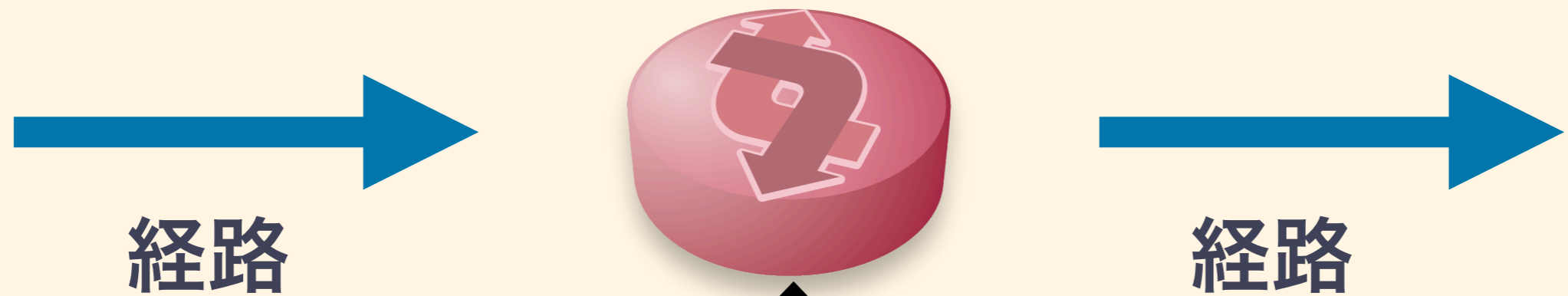
パケット転送に
プログラム実行をはさむのは
スループットが足りないが、
経路転送ならいけるのでは！



BMP があつた 😊



興味があるのはLoc-RIB 🙄



Route Filter

プログラム

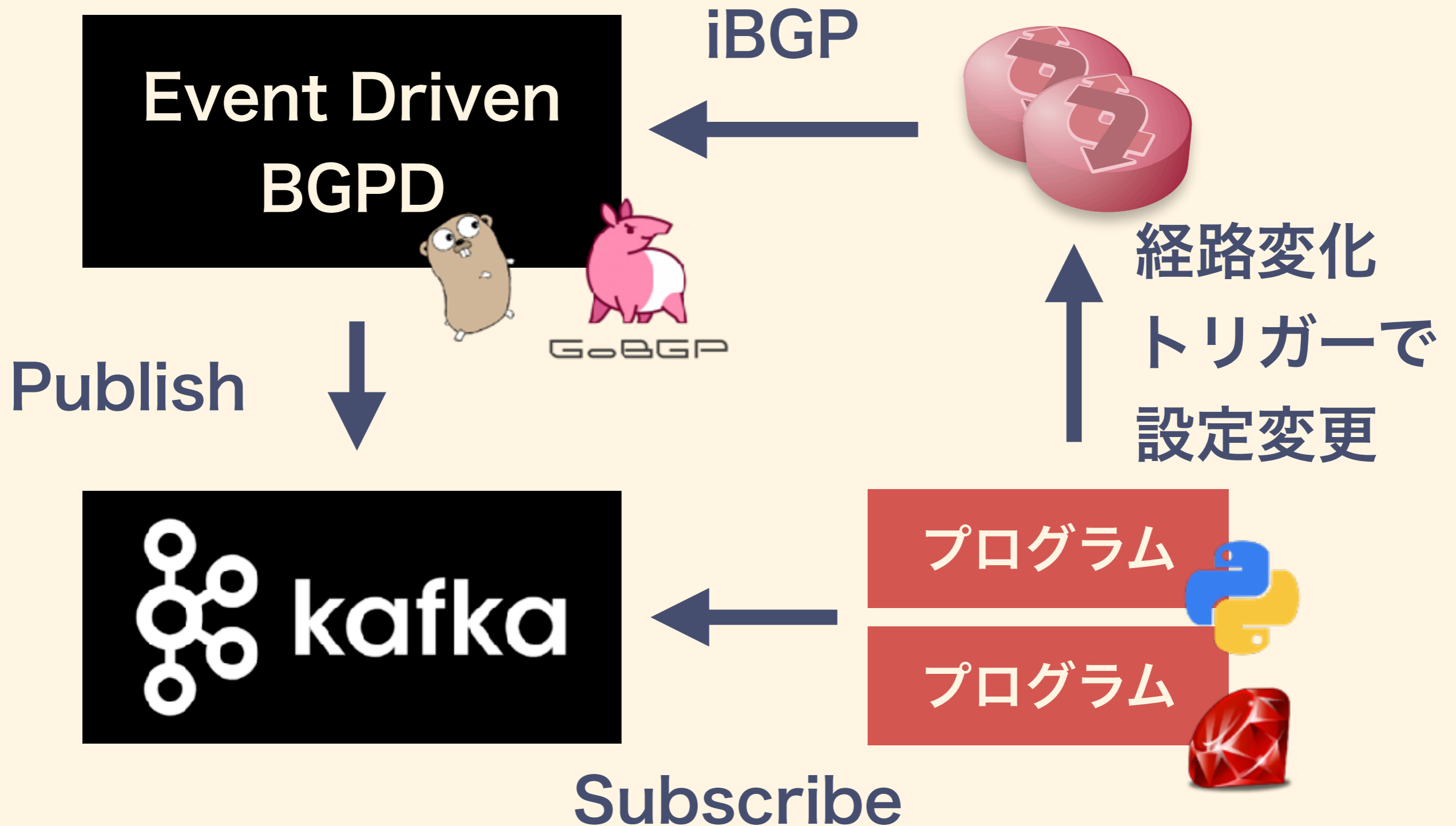
もっと言うと
Best Path

やりたいこと

全ルーターのBest Path 変化にフックして

- ・ 他社の経路変化をつかまえる
 - ・ 重要な変化を通知
 - ・ 変化したらルーターの設定を変える
 - ・ BGP Community に応じて送信経路を…
 - ・ Nexthop が変わったらそれをコピー…
- ・ メンテナンス時の経路確認をラクに
 - ・ 制御対象がどう変わったか通知

実装した



サンプル実装



`codeout/event-driven-bgpd`

- <https://github.com/codeout/event-driven-bgpd>
- 通常はデフォルトルートのみ広告
- 特定の経路を、決まった順番に受信したらフルルートを広告しはじめる

サンプル実装

```
Every 1.0s: gobgp neighbor                               kojimae: Mon Jan 16 08:19:17 2017

Peer          AS   Up/Down State      |#Received  Accepted
192.168.0.64 65001 00:00:53 Establ    |           1           1

event-driven-bgpd $ ./konami_client/send_command.sh
```

<https://youtu.be/4uQqg7uXqwY>

コナミコマンド

なにそれ? と思ったら
ググって 🤖

```
#!/bin/sh
```

```
gobgp global rib add 10.0.0.0/8 community 65000:38 # ↑
gobgp global rib del 10.0.0.0/8
gobgp global rib add 10.0.0.0/8 community 65000:38 # ↑
gobgp global rib add 10.0.0.0/8 community 65000:40 # ↓
gobgp global rib del 10.0.0.0/8
gobgp global rib add 10.0.0.0/8 community 65000:40 # ↓
gobgp global rib add 10.0.0.0/8 community 65000:37 # ←
gobgp global rib add 10.0.0.0/8 community 65000:39 # →
gobgp global rib add 10.0.0.0/8 community 65000:37 # ←
gobgp global rib add 10.0.0.0/8 community 65000:39 # →
gobgp global rib add 10.0.0.0/8 community 65000:66 # B
gobgp global rib add 10.0.0.0/8 community 65000:65 # A
gobgp global rib del 10.0.0.0/8
```

```
gobgp neighbor 192.168.0.64 reset # Start
```

サンプル実装

実装は難しくない



BGPD

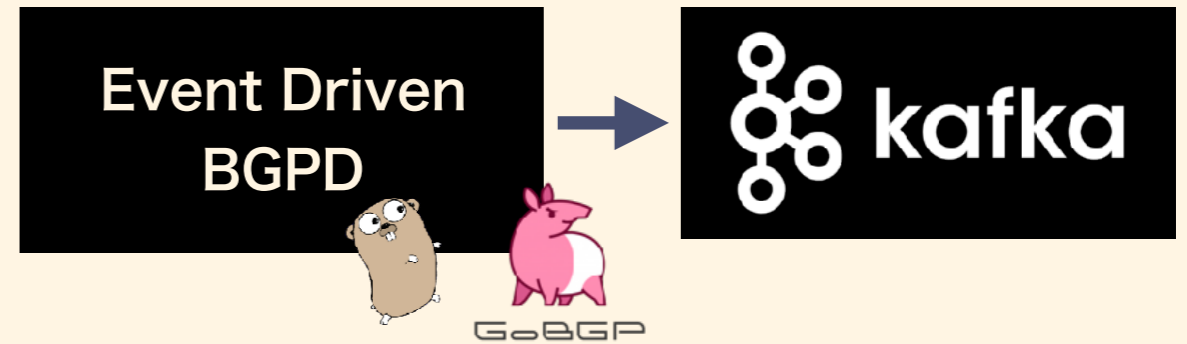
- Event をQueue に投げる部分だけ書いて、BGPD としてのふるまいは gobgpd に頼る
- API を叩くだけ



プログラム

- Event をQueue から取ってきて何かする
- 目的ごとに小さく書く。言語はなんでもOK

Queue に イベントを投げる



```
// setup kafka
client, err := kafka.NewProducer()
if err != nil {
    log.Fatal(err)
}

// monitor new routes and peer state
w := s.Watch(gobgp.WatchBestPath(), gobgp.WatchPeerState(false))
for {
    select {
    case ev := <-w.Event():
        switch msg := ev.(type) {
        case *gobgp.WatchEventBestPath:
            client.SendPathListMessage(msg)
        case *gobgp.WatchEventPeerState:
            client.SendPeerStateMessage(msg)
        }
    }
}
```

kafka
ドライバー初期化

A mole character is positioned on the right side of the slide, pointing towards the text in the speech bubble.

Best Path が
変化したら処理する。
変化があるまでブロック

A mole character is positioned on the right side of the slide, pointing towards the text in the speech bubble.

Queue から イベントを取ってくる



プログラム

kafka、イベントハンドラー
ドライバー初期化

```
kafka = Kafka.new(seed_brokers: 'localhost:9092')  
handler = EventHandler.new('10.0.0.0/8')  
  
kafka.each_message(topic: 'edb', start_from_beginning: false) do |message|  
  handler.process message.value  
end
```

イベントが取れたら
処理する。
取れるまでブロック

イベントハンドラー

```
def configure_peer
  if @sequence==KONAMI
    @gobgp.update_policy default_accept: true
  else
    @gobgp.update_policy default_accept: false
  end
end

def process(event)
  event = JSON.parse(event)

  case event['type']
  when 'peer_state'
    process_peer_state event['value']
  when 'best_path'
    process_best_path event['value'] unless event['value']['withdrawal']
  end
end
```

コナミの並びなら
広告ポリシーを変える

ピア状態イベントの処理
落ちたらコナミ判定

Best Path 変化イベントの処理
Update の履歴を記録

そんなに難しくくない



```
event-driven-bgpd $ ls bgpd/**/*.go konami/**/*.rb~*_pb.rb
bgpd/kafka/client.go      konami/event_handler.rb  konami/subscriber.rb
bgpd/main.go              konami/gobgp_client.rb

event-driven-bgpd $ cat bgpd/**/*.go konami/**/*.rb~*_pb.rb | wc -l
209
```

- 全部で 5ファイル、210行

- えっ！ たったそれだけ？ 簡単



まとめ

- Event Driven BGPD をつくるのはそんなに難しくない
 - 既存ネットワークに付加しやすいよう iBGP で
- 運用コストが下がったり、API としての BGP を提供できる可能性がある
 - “show route” を減らす
 - BGP でリクエストを受けてサービス設定を変える

べんりですよ！ ぜひ試してみてください

