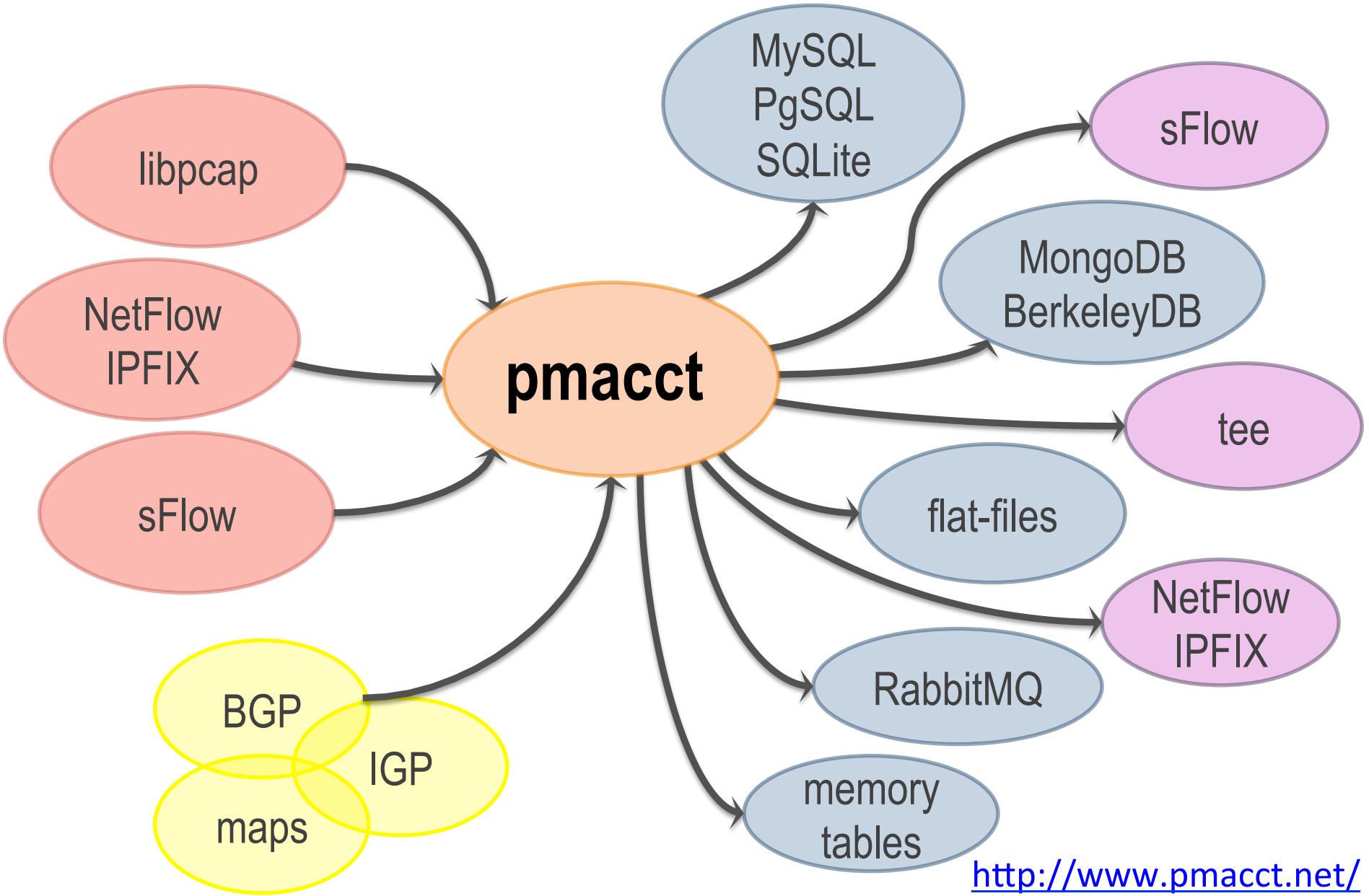
オープンソースのネットフローツールの運用

Paolo’s part, draft slides

v0.1

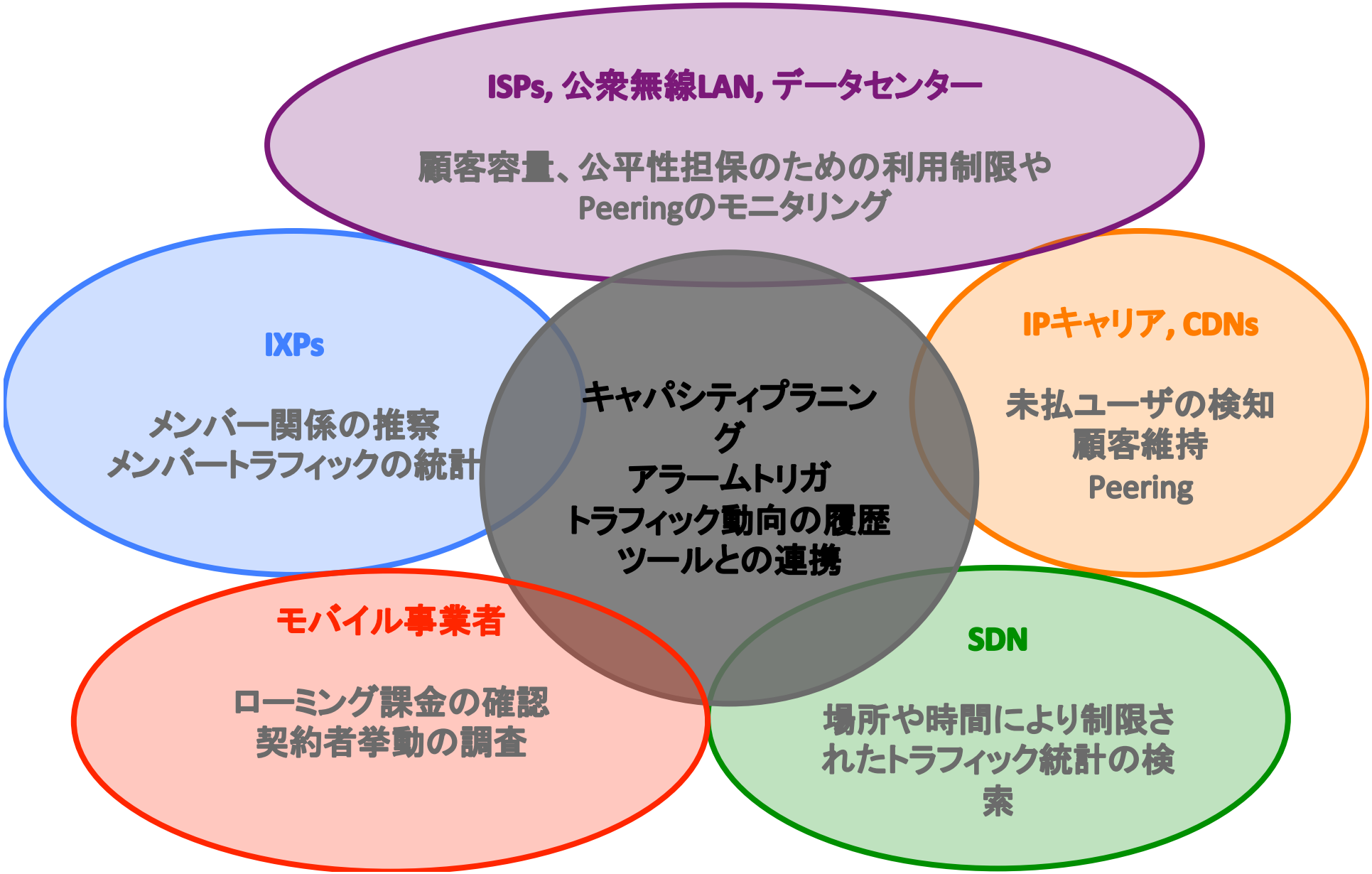
pmacctの概要

pmacctはオープンソース、無料、GPLソフトである



http://www.pmacct.net/

幅広く利用できる


ISPs, 公衆無線LAN, データセンター
顧客容量、公平性担保のための利用制限や
Peeringのモニタリング
IXPs
メンバー関係の推察
メンバートラフィックの統計
キャパシティプラニング
グ
アラームトリガ
トラフィック動向の履歴
ツールとの連携
IPキャリア, CDNs
未払ユーザの検知
顧客維持
Peering
モバイル事業者
ローミング課金の確認
契約者挙動の調査
SDN
場所や時間により制限さ
れたトラフィック統計の検
索

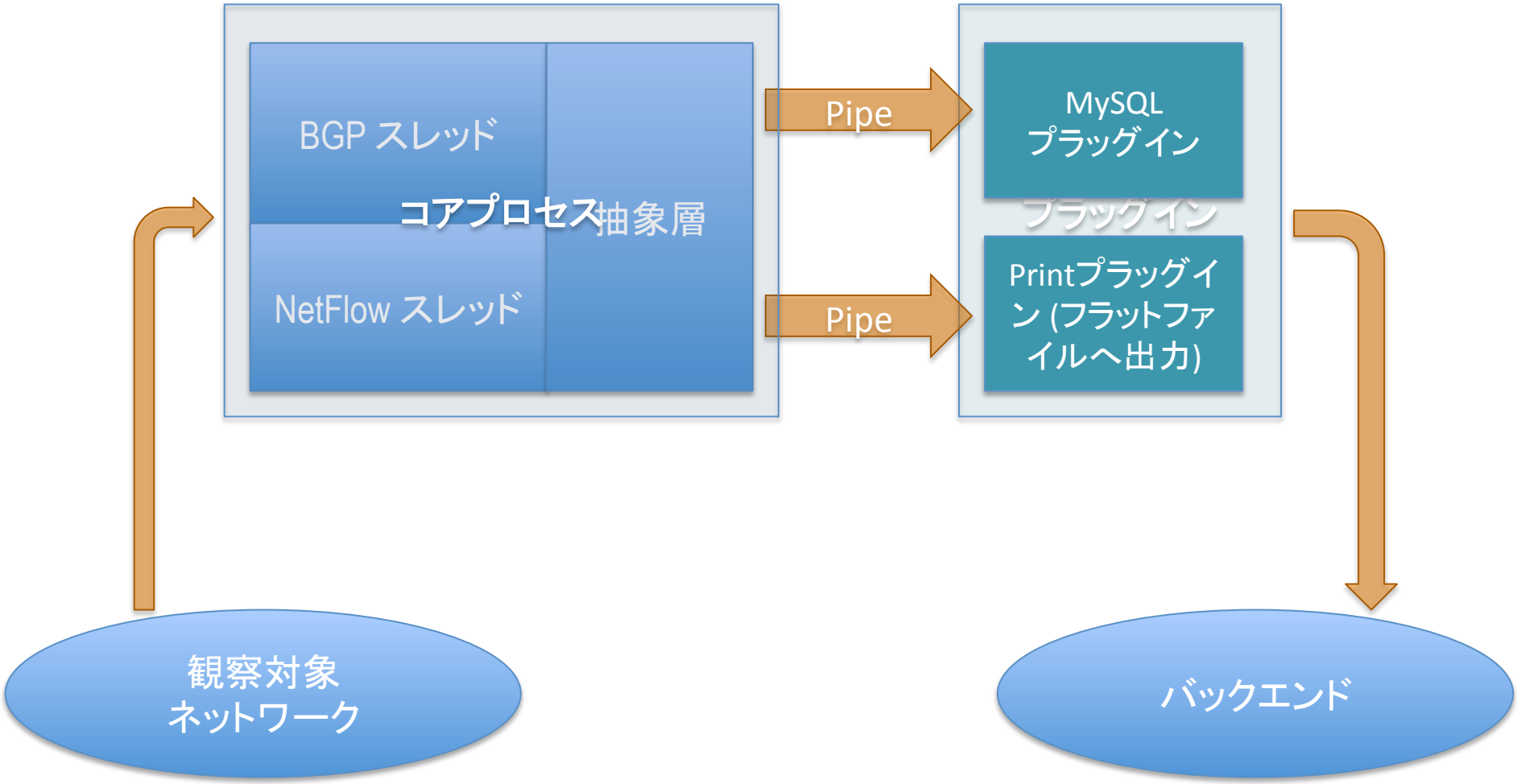
pmacct：技術面以外のポイント

- 10年以上歴史を持つプロジェクト
- 二杯目以降その名前のつづりを言うのは無理
- 無料、オープンソース、独立
- 積極的に開発中
- イノベーションが導入されている
- 大手SPにも、よく実装されてる
- オペレーターコニュニティーニーズに近い
  トラフィック集計ツールを目指している

技術の特徴 (1/3)

- 機能追加しやすいアーキテクチャ
  - 新しい集計方法やバックエンドの直接追加ができる
- どんな集計メソッドでもどんなバックエンドと連携を許可する抽象レイヤ
- マルチプロセスとマルチスレッディング（まだ粗いけど。。）にも対応
  - 複数のプラグインがランタイムでインスタンスを生成し、それぞれ別々で設定可能

技術の特徴 (2/3)


BGP スレッド
コアプロセス
抽象層
NetFlow スレッド
Pipe
Pipe
MySQL
プラッグイン
プラッグイン
Printプラッグイ
ン (フラットファ
イルへ出力)
観察対象
ネットワーク
バックエンド

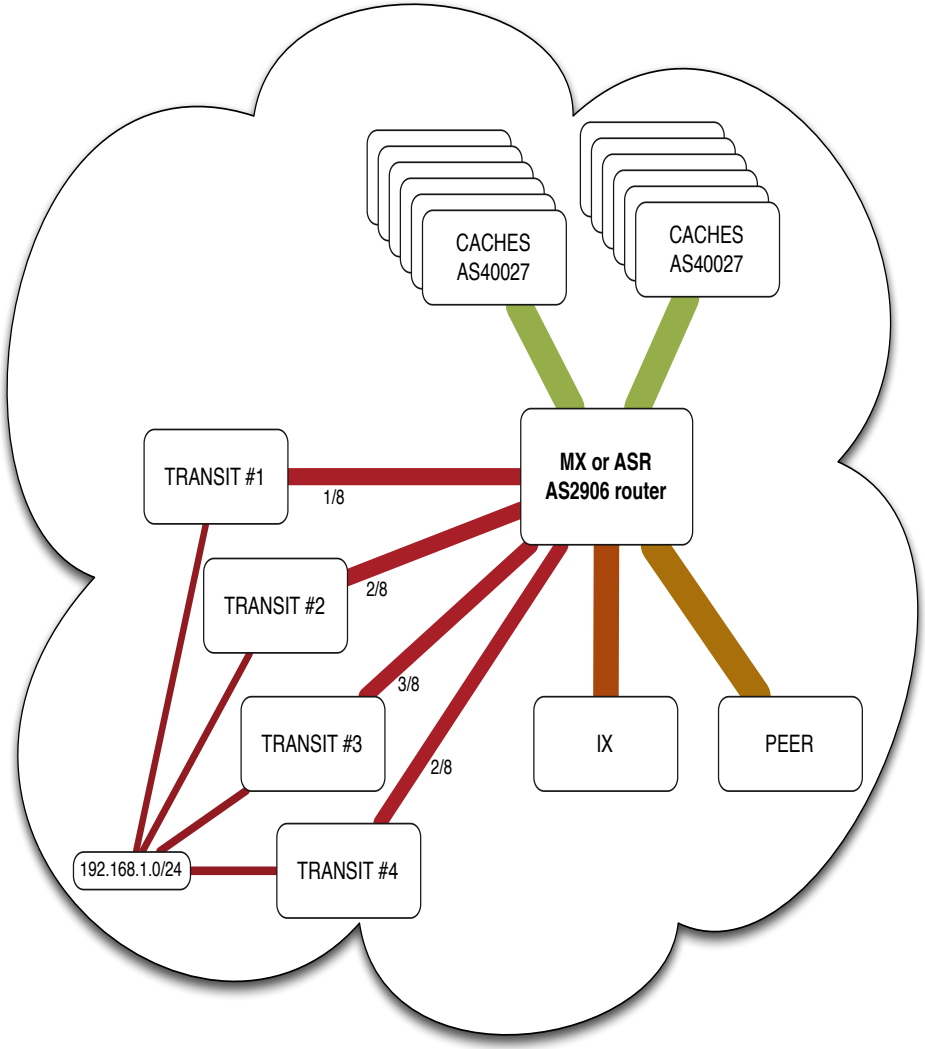
技術の特徴 (3/3)

- 一般的なデータ削減方法 例えば:
  - データアグリゲーション
  - タグ付けとフィルタリング
  - サンプリング
- 集計しているトラフィックデータセットから複数のビューを抽出が可能　例えば:
  - セキュリティや証拠捜査のため、アグリゲーションせずフラットファイル（生データ）で出力
  - キャパシティプランニングのため、[<ingress router>, <ingress interface>, <BGP next-hop>, <peer destination ASN>]のようなトラフィックマトリックスを作成

Netflixユーズケース
（ピアリング分析、トラフィック可視化）

Egress BGP hacks

- 4接続程度の対外接続でも対応できないほど多いトラフィックが発生している
- 異なるASNを経由するBGPマルチパスを運用している



BGP add-pathについて

- BGPの拡張の一つ。既存のパスを新しいものと置き換えないようにして複数パスの経路広報を許可します。
- Draft at IETF: draft-ietf-idr-add-paths-09

課題

- BGPマルチパスで、唯一のベストパスのみトラフィックを打つことではない
- pmacctはBGPフィードからベストパスしか受けてないことだった

BGP Multi-path

```
192.168.1.0/24      [BGP/170] 3w0d 01:19:58, MED 100, localpref 200
                      AS path: 789 I, validation-state: unverified
                    > to 10.0.0.1 via ae12.0
                    [BGP/170] 3w0d 01:15:44, MED 100, localpref 100
                      AS path: 123 456 789 I, validation-state: unverified
                    > to 10.0.0.2 via ae8.0
                    [BGP/170] 3w0d 01:13:48, MED 100, localpref 100
                      AS path: 321 654 789 I, validation-state: unverified
                    > to 10.0.0.3 via ae10.0
                    [BGP/170] 3w0d 01:18:24, MED 100, localpref 100
                      AS path: 213 546 789 I, validation-state: unverified
                    > to 10.0.0.4 via ae1.0
```

Traditional BGP to pmacct

```
* 192.168.1.0/24             10.0.0.1       100 200     789 I
```

作動中のBGP add-path

- BGP add-pathは複数のBGP multi-pathベストパスが見える

BGP Multi-path

```
192.168.1.0/24      [BGP/170] 3w0d 01:19:58, MED 100, localpref 200
                      AS path: 789 I, validation-state: unverified
                    > to 10.0.0.1 via ae12.0
                    [BGP/170] 3w0d 01:15:44, MED 100, localpref 100
                      AS path: 123 456 789 I, validation-state: unverified
                    > to 10.0.0.2 via ae8.0
                    [BGP/170] 3w0d 01:13:48, MED 100, localpref 100
                      AS path: 321 654 789 I, validation-state: unverified
                    > to 10.0.0.3 via ae10.0
                    [BGP/170] 3w0d 01:18:24, MED 100, localpref 100
                      AS path: 213 546 789 I, validation-state: unverified
                    > to 10.0.0.4 via ae1.0
```

BGP ADD-PATH to pmacct

```
* 192.168.1.0/24          10.0.0.1       100 200     789 I
                          10.0.0.2       100 100     123 456 789 I
                          10.0.0.3       100 100     321 654 789 I
                          10.0.0.4       100 100     213 546 789 I
```

NetFlow/IPFIXとBGP add-path (1/2)

- OK、N個のベストパスが見えるようになった…
- …でも、Netflowトラフィックをどうやったらそのパスにマッピングできるの?
  - トラフィックを複数のパスに分散する作業をやるなんて面倒くさい
  - NetFlowが教えてくれるさ！NetFlowのBGP next-hopは、BGPデータとトラフィックデータを結びつける選別子として使える
  - NetFlowのBGP Nexthopが実際のパス決定に使えるかどうか最初は心配したが、、、
    - 複数ベンダーの製品でも正確で合理的だと分かった

NetFlow/IPFIXとBGP add-path (2/2)

NetFlow

```
SrcAddr:         10.0.1.71
DstAddr:         192.168.1.148
NextHop:         10.0.0.3
InputInt:        662
OutputInt:       953
Packets:         2
Octets:          2908
Duration:        5.112000000 sec
SrcPort:         80
DstPort:         33738
TCP Flags:       0x10
Protocol:        6
IP ToS:          0x00
SrcAS:           2906
DstAS:           789
SrcMask:         26 (prefix: 10.0.1.64/26)
DstMask:         24 (prefix: 192.168.1.0/24)
```

BGP ADD-PATH to pmacct

```
* 192.168.1.0/24          10.0.0.1       100 200     789 I
                          10.0.0.2       100 100     123 456 789 I
                          10.0.0.3       100 100     321 654 789 I
                          10.0.0.4       100 100     213 546 789 I
```

実装について

- 複数のpmacctサーバを様々な場所で配置
- BGP ADD-PATHSはルータとpmacct serversの間で設定
  - セッションはiBGP, RR-clientとして設定
  - Juniper ADD-7 (maximum)
  - Cisco ADD-ALL
- NetFlowをpmacctサーバにエクスポートさせる
  - NetFlow v5, v9 と IPFIXが混在

Spotifyユーズケース（SDN）

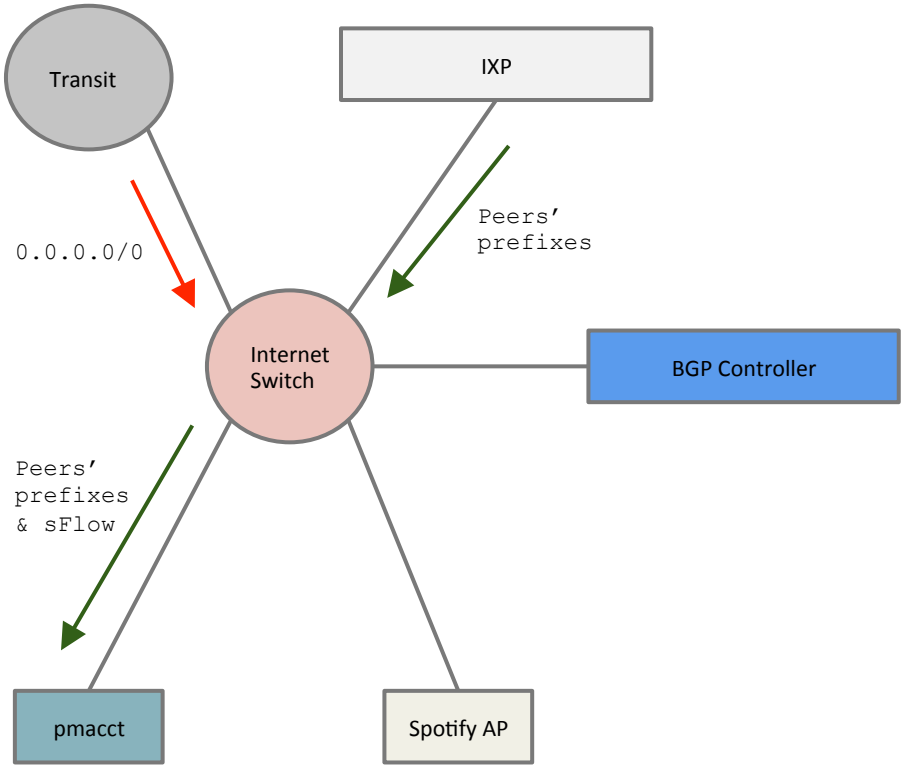
旅行中Spotifyを利用する際…

- 例：Spotifyデータセンター@ストックホルム
  - プレフィックス合計：~519k
  - ピアから受け取るプレフィックス数：~150k
  - 一日あたりの平均Activeプレフィックス数：**~16k**
- 例の説明：
  - Spotifyはユーザに音楽を配信している
  - だいたい一番近いデータセンターから配信される
  - なんでサンノゼのSpotifyデータセンターが$EU_COUNTRYのユーザへどうやって届けるか知らなければならないのだろうか？

私たちのゴール

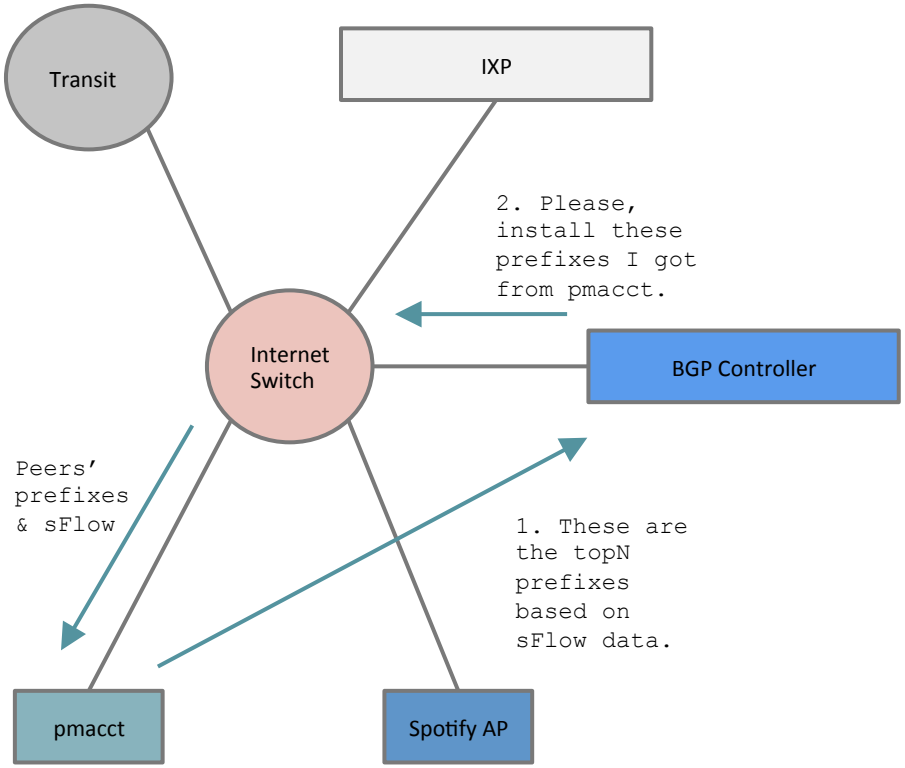
- 「必要な」ルートをRIBから選ばせ、汎用ASICを持つスイッチのFIBにも適用できるようにする
- その結果、汎用ASICに実装できるルートの数まで経路数を削減できる

アーキテクチャ概要



- TransitがdefaultルートをInternet Switchに広報し、そのルートがFIBにそのままインストールされる
- IXPやPeerからいくつかの経路を受信する。Internet SwitchにはインストールされないがpmacctとBGPコントロラには転送される
- pmacctはさらにsFlowデータも受信

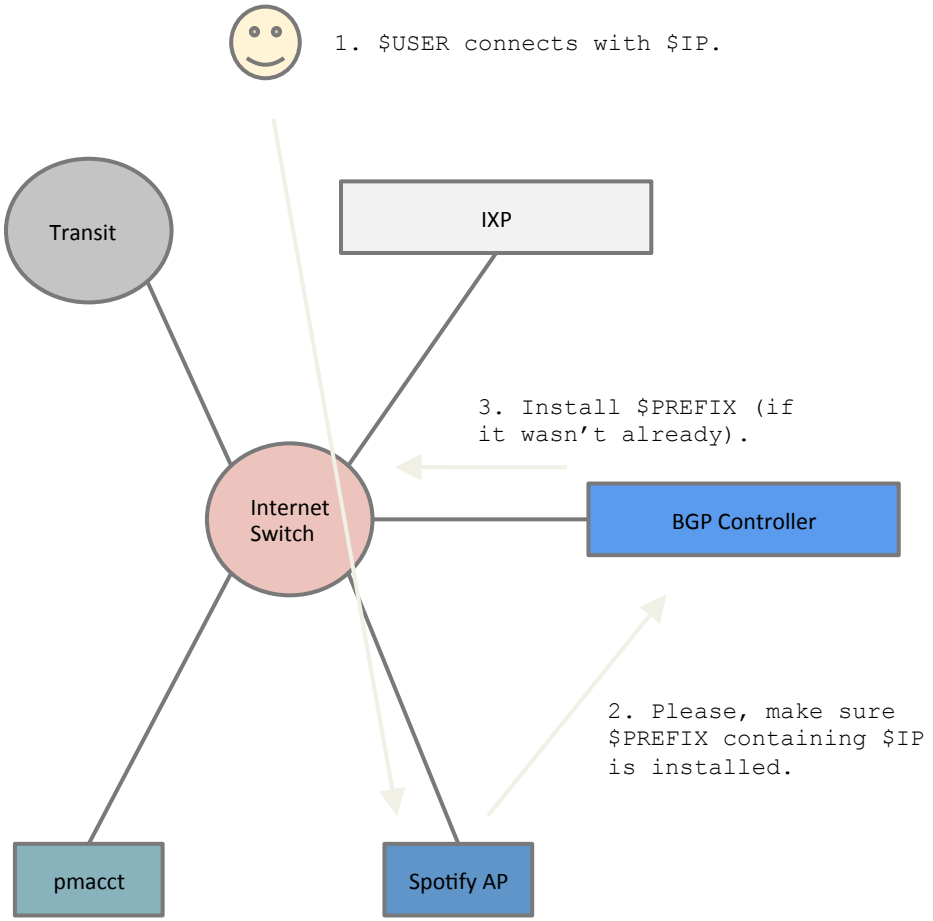
pmacct



- pmacctは先ほどインターネットスイッチから送信してきたBGP情報を利用し、sFlowデータをまとめる
- pmacctがフローデータをBGPコントロラに報告
- BGPコントロラはインターネットスイッチがそのTopN*プレフィックスをインストールするように指示

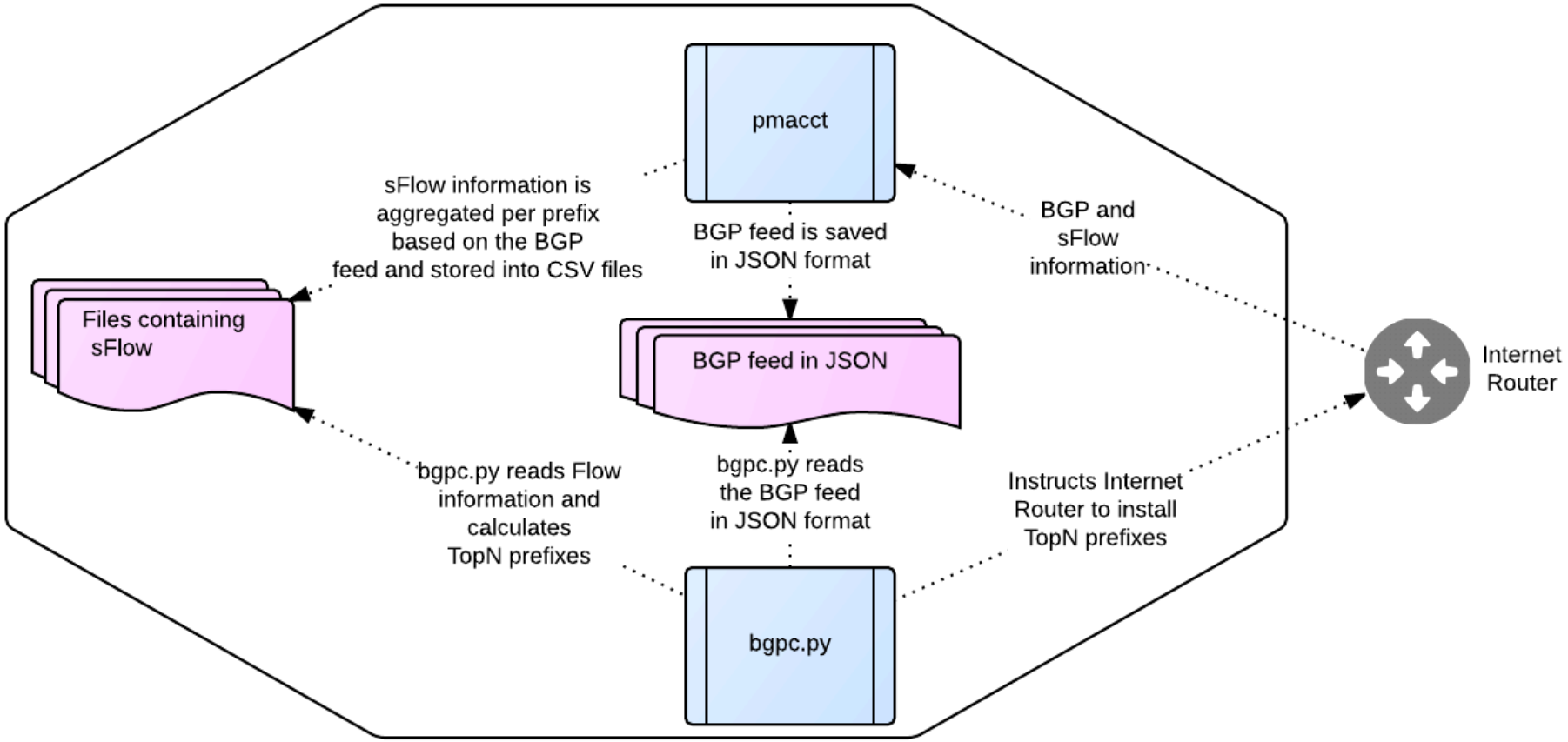
* N is a number close to the maximum number of entries that the FIB of the Internet Switch can support

Spotify AP

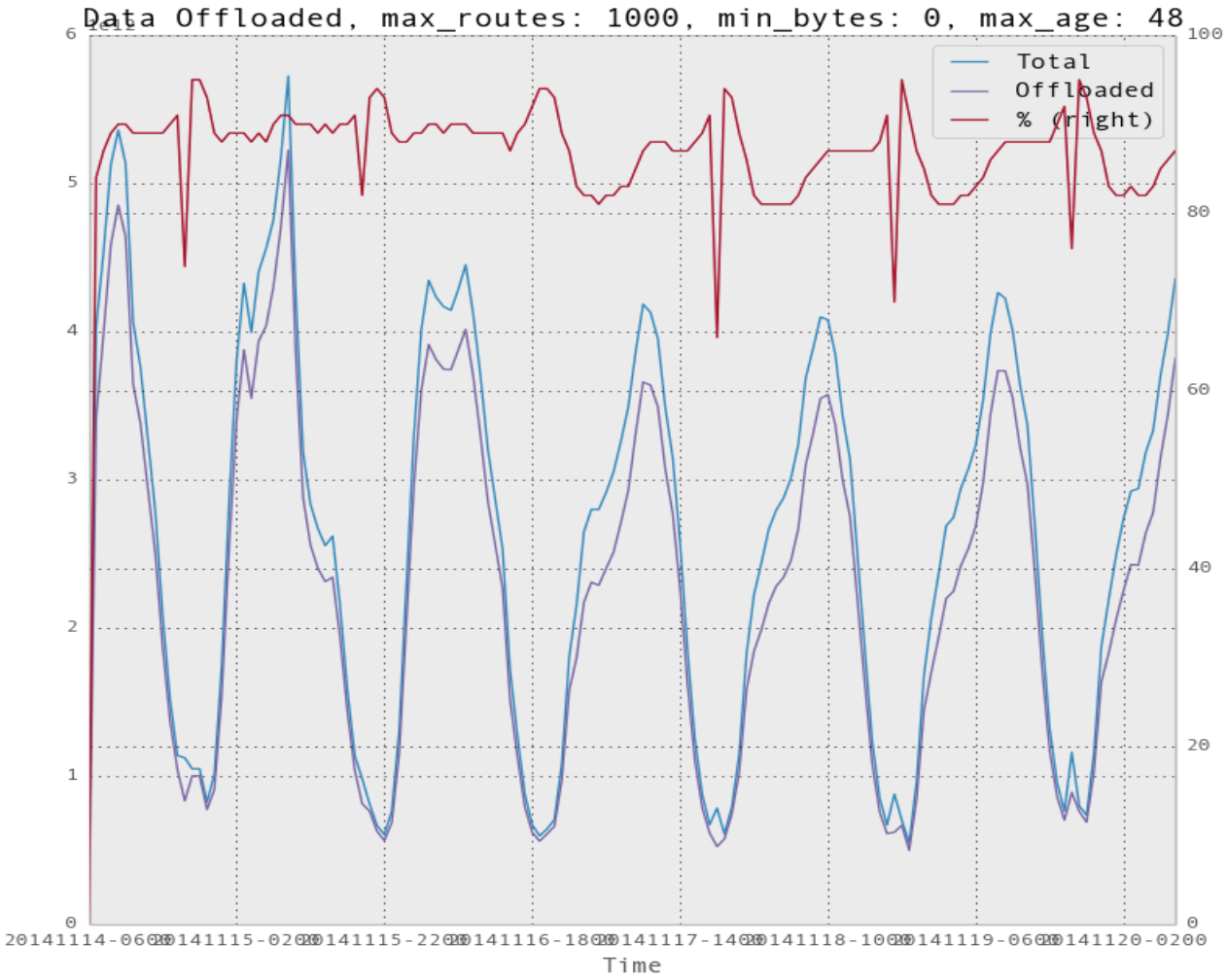


- $USERはサービスにアクセス
- アプリケーションはアクセスポイントに該当 $USER がすでに接続していることを通告し、その$IPを含めている$PREFIXをFIBにインストールするよう要求する
  - 他の同じレンジのユーザが接続したこと、もしくはpmacctが該当プレフィックスをTopNの一つとして報告したことがあれば、該当プレフィックスはすでにインストールされているかもしれない
- 必要であれば、BGPコントロラがインターネットスイッチに該当プレフィックスをインストールするように指示する

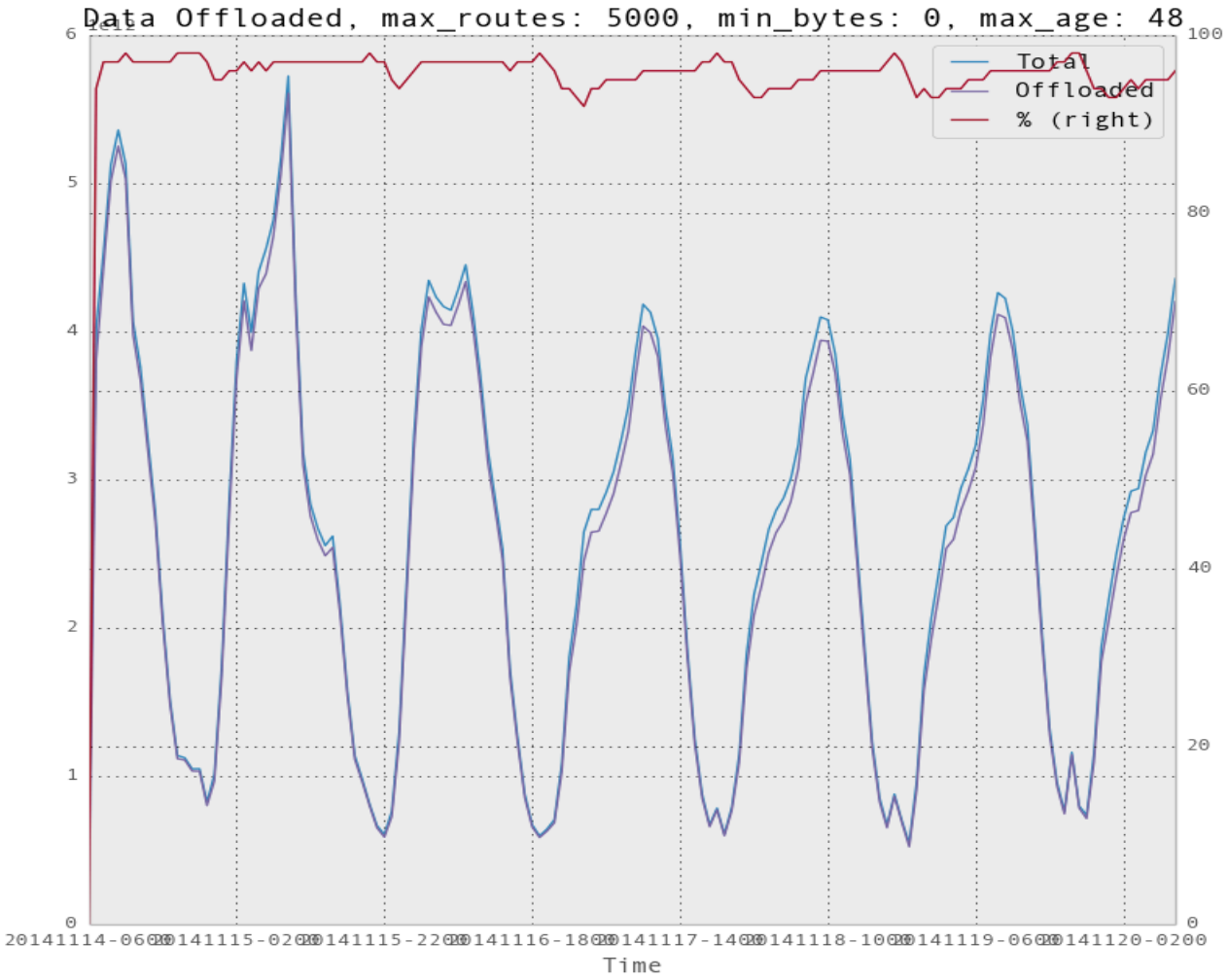
内部構造


pmacct
sFlow information is aggregated per prefix based on the BGP feed and stored into CSV files
BGP feed is saved in JSON format
BGP and sFlow information
Files containing sFlow
BGP feed in JSON
Internet Router
bgpc.py reads Flow information and calculates TopN prefixes
bgpc.py reads the BGP feed in JSON format
Instructs Internet Router to install TopN prefixes
bgpc.py

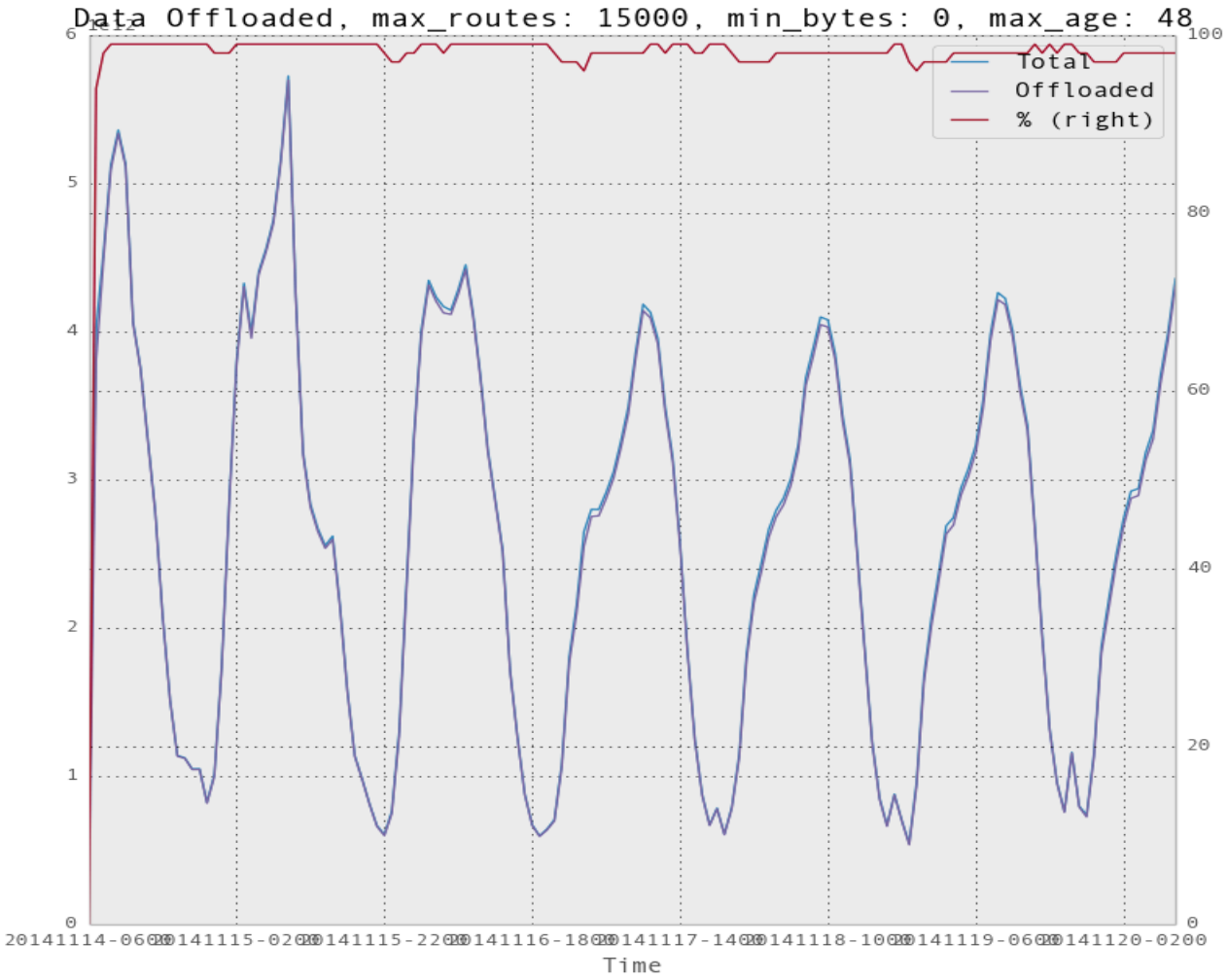
結果：Top 1kルート (1/4)



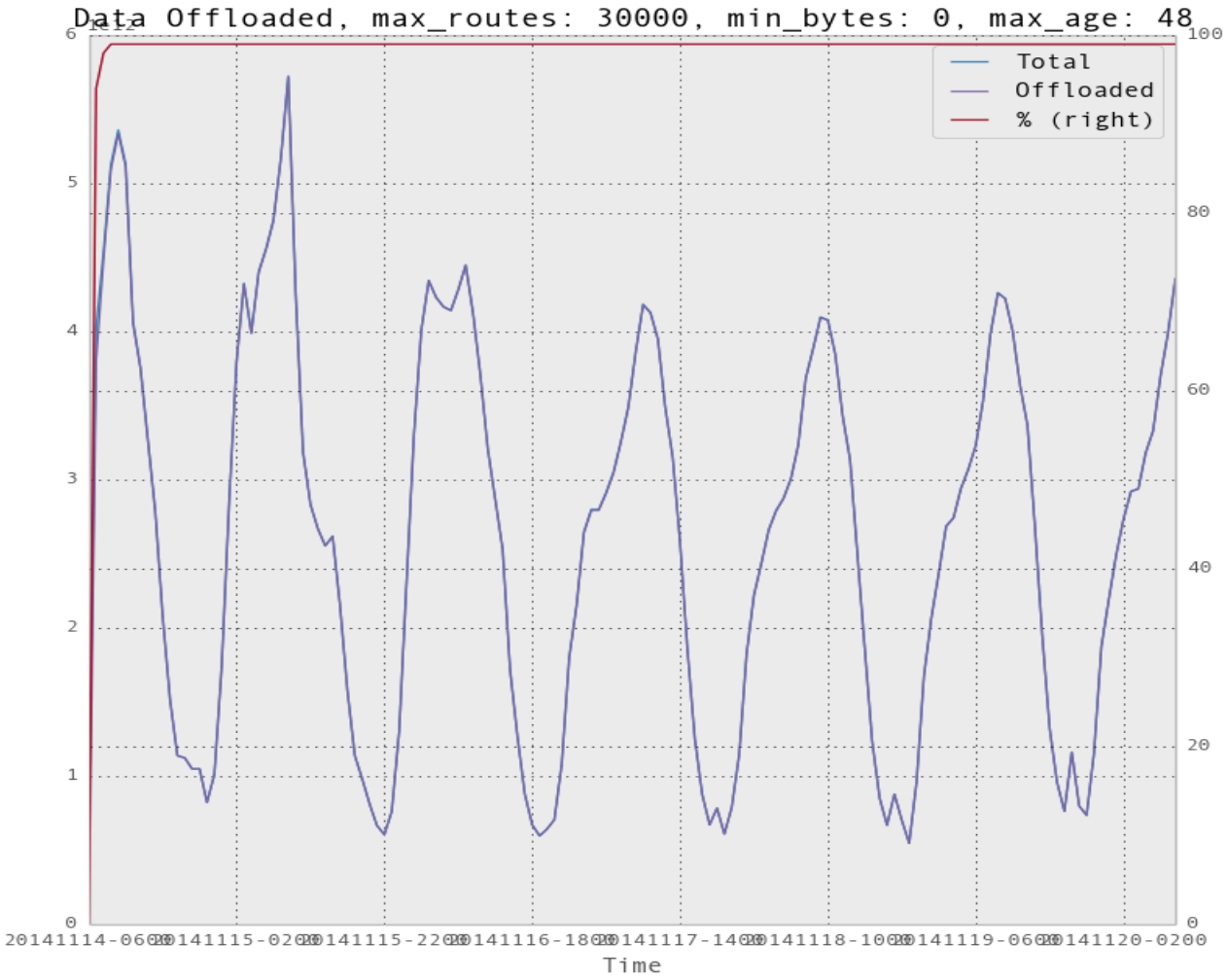
結果：Top 5kルート (2/4)



結果：Top 15kルート (3/4)



結果：Top 30kルート (4/4)



実装について

- デモはSpotifyストックホルムデータセンターで稼働中。Netnodに接続。
  - そこでルート情報を収集しているが、インターネットルータに変更することはまだ行っていない
- Spotifyは近いうちにヨーロッパの重要IXPと一緒に、パイロット試験を実施する予定

感謝

- Elisa Jasinska
  - elisa@bigwaveit.org
- David Barroso
  - dbarroso@spotify.com

まとめ

さらなる情報 (1/2)

- http://www.pmacct.net/dbarroso_plucente_waltzing_v0.5.pdf
  - Spotifyユーズケース詳細
- http://www.pmacct.net/nanog61-pmacct-add-path.pdf
  - Netflixユーズケース詳細
- http://www.pmacct.net/Lucente_collecting_netflow_with_pmacct_v1.2.pdf
  - pmacctのチュトーリアル

さらなる情報 (2/2)

- http://www.pmacct.net/lucente_pmacct_uknof14.pdf
  - 遠隔測定とBGPについて
- http://ripe61.ripe.net/presentations/156-ripe61-bcp-planning-and-te.pdf
  - 遠隔測定、トラフィックマットリックス、キャパシティプランニング、トラフィックエンジニアリング
- http://wiki.pmacct.net/OfficialExamples
  - pmacctのコンパイル及びクイックスタート案内
- http://wiki.pmacct.net/ImplementationNotes
  - pmacctの開発について (RDBMS、メンテナンスなど)

オープンソースのネットフローツールの運用

JANOG36 BoF

maoke@bbix.net
paolo@pmacct.net

JANOG36 meeting, Kitakyushu – Jul 2015