RIPE NCC
RIPE NETWORK COORDINATION CENTRE

RIPE NCC Routing Information Service (RIS)

2019 Update

Iñigo Ortiz de Urbina on behalf of Chris Amin, Oleg Muravskiy | ESNOG23 | May ‘19

Introduction

- $(whoami)
  - Iñigo Ortiz de Urbina Cazenave
  - Linux system engineer @ ~~RIPE NCC~~${PSP}
- **Not** my slides
  - Consent from {camin,oleg}@ripe.net

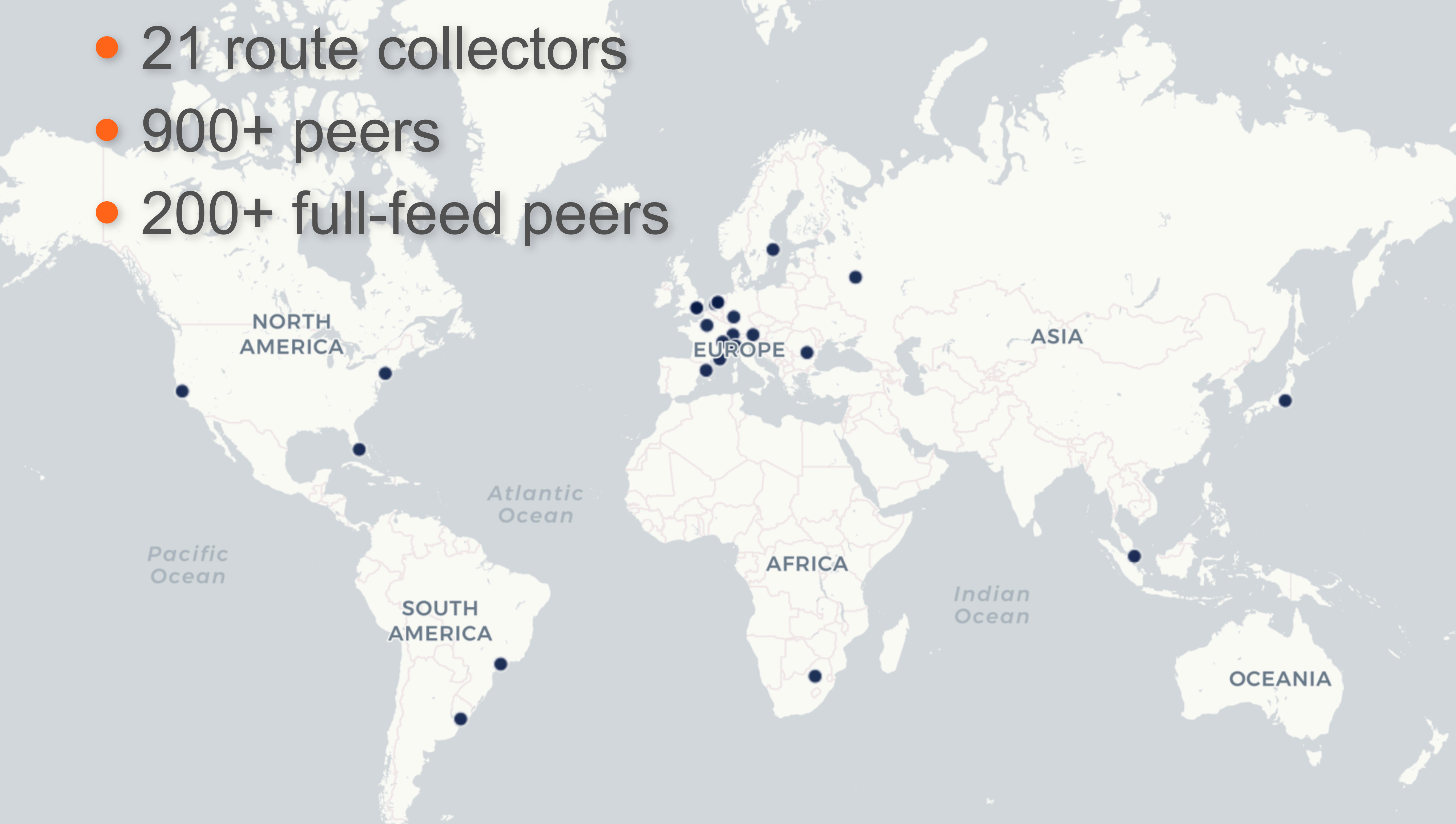
What is RIS?

What is RIS?

- Routing Information Service
- Worldwide network of BGP collectors
- Deployed at Internet Exchange Points
- Collects raw BGP data from peers
- Stores BGP messages and routing table dumps
- 19 years of history
- Used by network operators and researchers every day

Collector locations

- 21 route collectors
- 900+ peers
- 200+ full-feed peers



Why RIS?

Why are we doing this?
A bit of history

Why RIS?

- Original project was defined in RIPE-200 in 1999:

  “In other words, it can be regarded as one integrated Looking-Glass for the entire Internet that includes history information”

- Looking glasses are instantaneous
- Routing problems are also instantaneous
- BGP history is recorded to track what is happening and what has happened
- Also to provide statistics and reporting on routing table metrics

Why the RIPE NCC RIS?

- RIPE NCC is a neutral body
- Experience running measurement platforms
  - Test Traffic Measurement project
  - RIPE Atlas
- Supporting our own members
  - who are mainly network operators
- Supporting the community
  - researchers
  - operators

RIS interfaces

Raw data!

- 19 years of raw data (8.7 TB) available to download and analyse yourself :)
  - https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data
- Data stored in MRT (RFC6396) format
- Readable using BGPdump utility
  - open source, maintained by RIPE NCC
  - https://bitbucket.org/ripencc/bgpdump
- …and by other tools

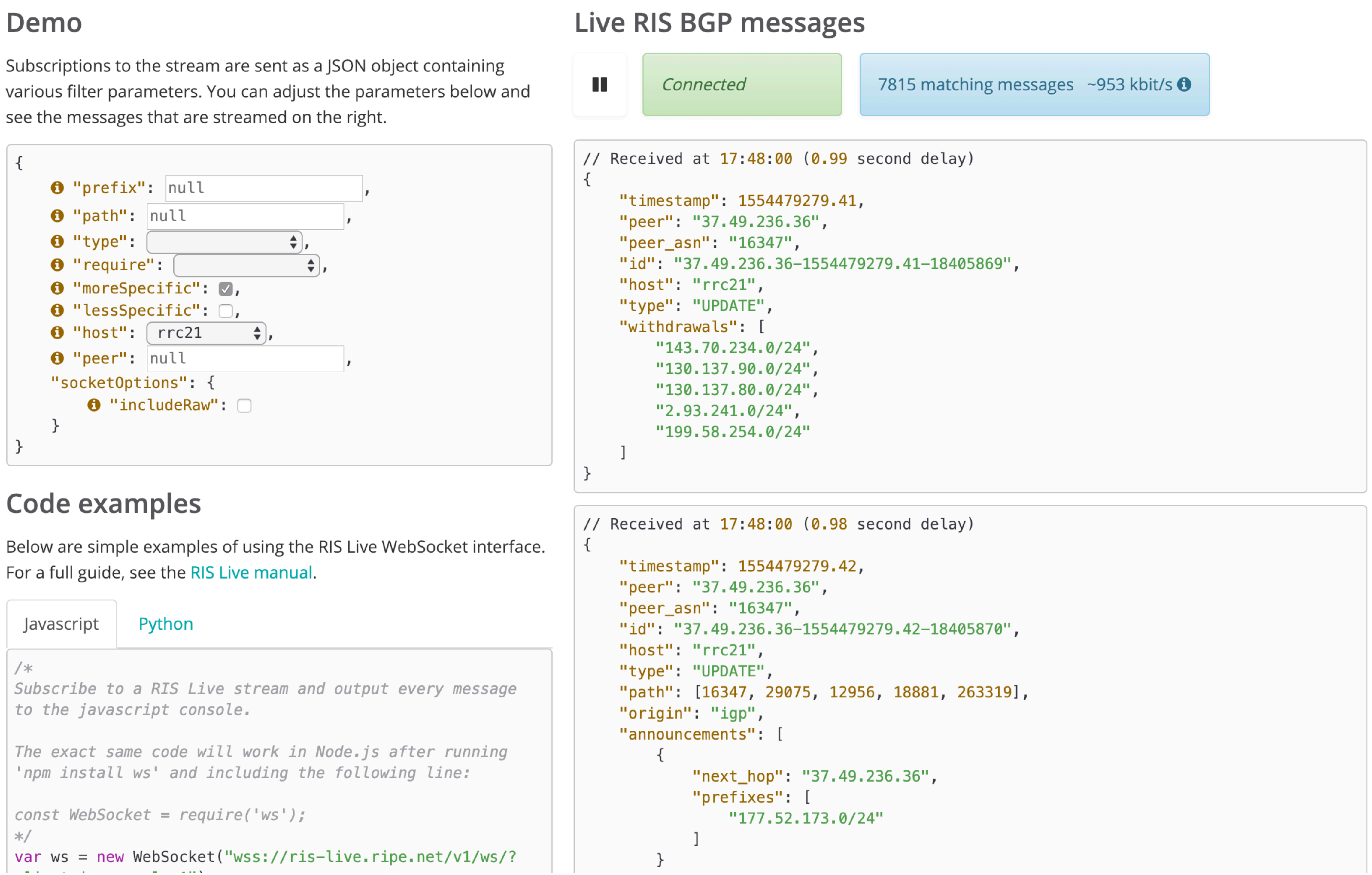
Web interfaces and APIs

- Of course, if all we did was store the raw data, we'd just need a bunch of hard disks and an FTP server
  - But you want to query all our lovely datasets!
- RIPEstat widgets – https://stat.ripe.net
  - our portal for everything you ever wanted to know!
- RIPEstat API – https://stat.ripe.net/data
  - all the data for widgets and much more
- RIS Live – https://ris-live.ripe.net
  - near-real-time BGP data stream

Web interfaces and APIs

- Of course, if all we did was store the raw data, we'd just need a bunch of hard disks and an FTP server
  - But you want to query all our lovely datasets!
- RIPEstat widgets – https://stat.ripe.net
  - our portal for everything you ever wanted to know!
- RIPEstat API – https://stat.ripe.net/data
  - all the data for widgets and much more
- **RIS Live – https://ris-live.ripe.net**
  - **near-real-time BGP data stream**

RIS Live – https://ris-live.ripe.net



Demo

Subscriptions to the stream are sent as a JSON object containing various filter parameters. You can adjust the parameters below and see the messages that are streamed on the right.

```
{
    ℹ "prefix": null,
    ℹ "path": null,
    ℹ "type": ,
    ℹ "require": ,
    ℹ "moreSpecific": ☑,
    ℹ "lessSpecific": □,
    ℹ "host": rrc21,
    ℹ "peer": null,
    "socketOptions": {
        ℹ "includeRaw": □
    }
}
```

Code examples

Below are simple examples of using the RIS Live WebSocket interface. For a full guide, see the RIS Live manual.

```
/*
Subscribe to a RIS Live stream and output every message
to the javascript console.

The exact same code will work in Node.js after running
'npm install ws' and including the following line:

const WebSocket = require('ws');
*/
var ws = new WebSocket("wss://ris-live.ripe.net/v1/ws/?
```

Live RIS BGP messages

```
// Received at 17:48:00 (0.99 second delay)
{
    "timestamp": 1554479279.41,
    "peer": "37.49.236.36",
    "peer_asn": "16347",
    "id": "37.49.236.36-1554479279.41-18405869",
    "host": "rrc21",
    "type": "UPDATE",
    "withdrawals": [
        "143.70.234.0/24",
        "130.137.90.0/24",
        "130.137.80.0/24",
        "2.93.241.0/24",
        "199.58.254.0/24"
    ]
}
```

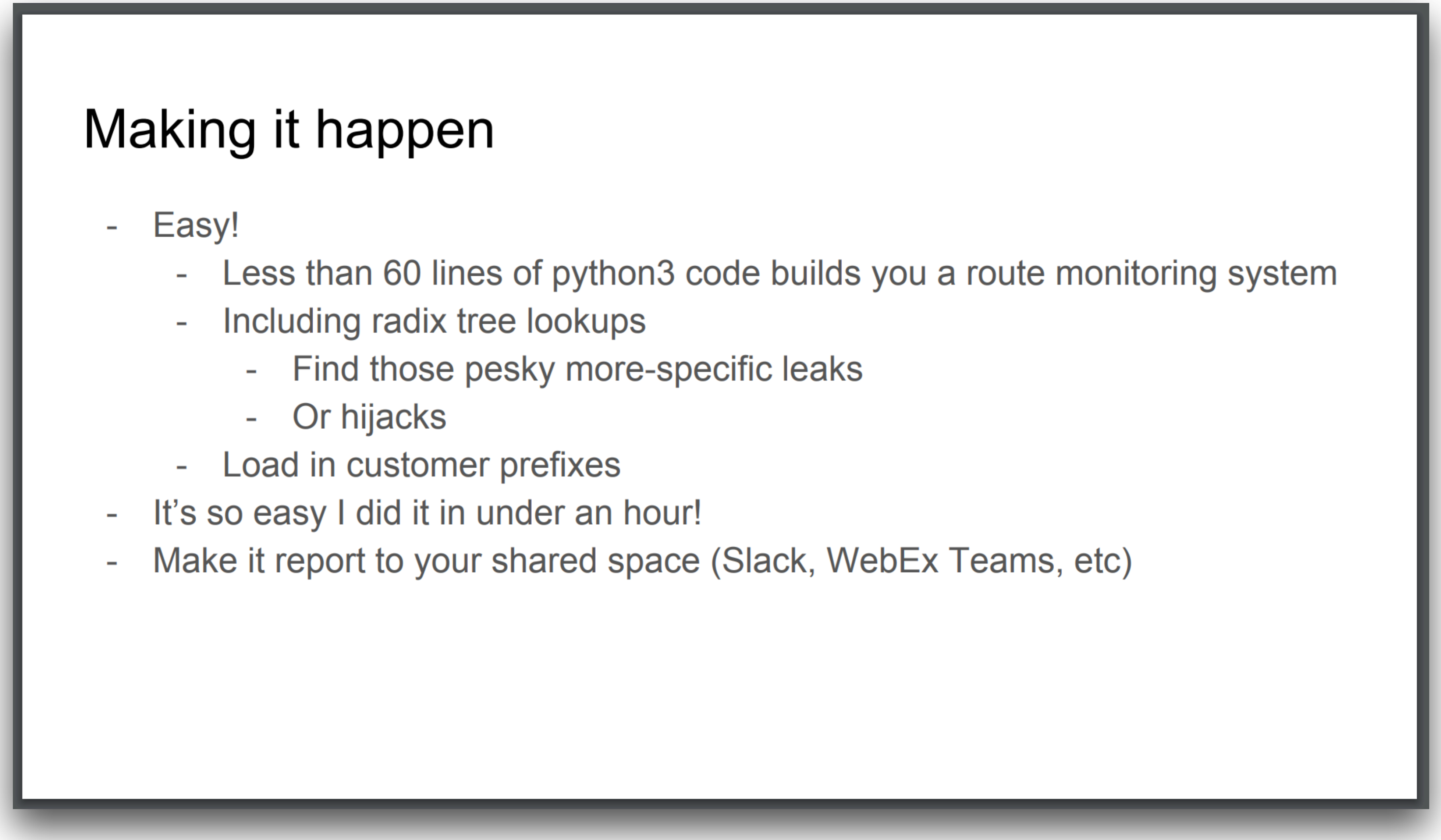
```
// Received at 17:48:00 (0.98 second delay)
{
    "timestamp": 1554479279.42,
    "peer": "37.49.236.36",
    "peer_asn": "16347",
    "id": "37.49.236.36-1554479279.42-18405870",
    "host": "rrc21",
    "type": "UPDATE",
    "path": [16347, 29075, 12956, 18881, 263319],
    "origin": "igp",
    "announcements": [
        {
            "next_hop": "37.49.236.36",
            "prefixes": [
                "177.52.173.0/24"
            ]
        }
```

RIS Live – https://ris-live.ripe.net

- Access to BGP messages from all RRCs
  - in near-real-time, sub-second latency
  - in JSON format (includes raw, base64 message)
  - using WebSockets API
  - with configurable filtering
- Formerly known as "RIS stream" *since 2014*
- Has a "production" status
- But in a "prototype" phase until August 2019
- **Please visit http://bit.ly/esnog23-ris-feedback if you want this service to remain**

RIS Live

- From Jared Mauch’s talk at NANOG 75:



Making it happen

- Easy!
  - Less than 60 lines of python3 code builds you a route monitoring system
  - Including radix tree lookups
    - Find those pesky more-specific leaks
    - Or hijacks
  - Load in customer prefixes
- It’s so easy I did it in under an hour!
- Make it report to your shared space (Slack, WebEx Teams, etc)

RIS Live – status so far

- Since 15 February 2019 we’ve had:
- 2,325 unique IPs
- 53,649 WebSockets sessions
- 14,111,267,775 messages streamed

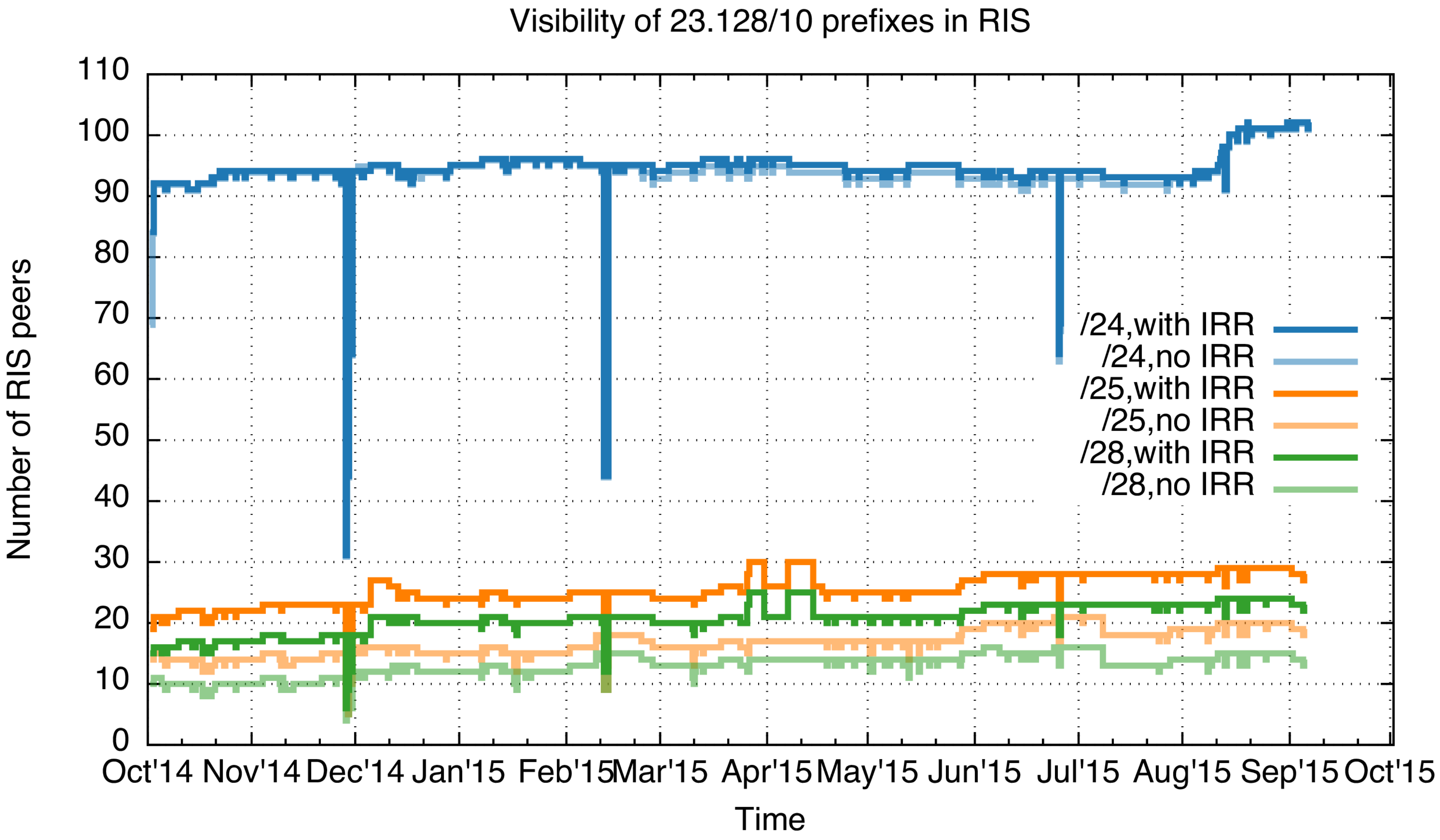
- **Final evaluation is due in September 2019**
- https://www.ripe.net/participate/forms/apply/ris-live-feedback-form/

What else can you do?

Lots of analysis that this data allows

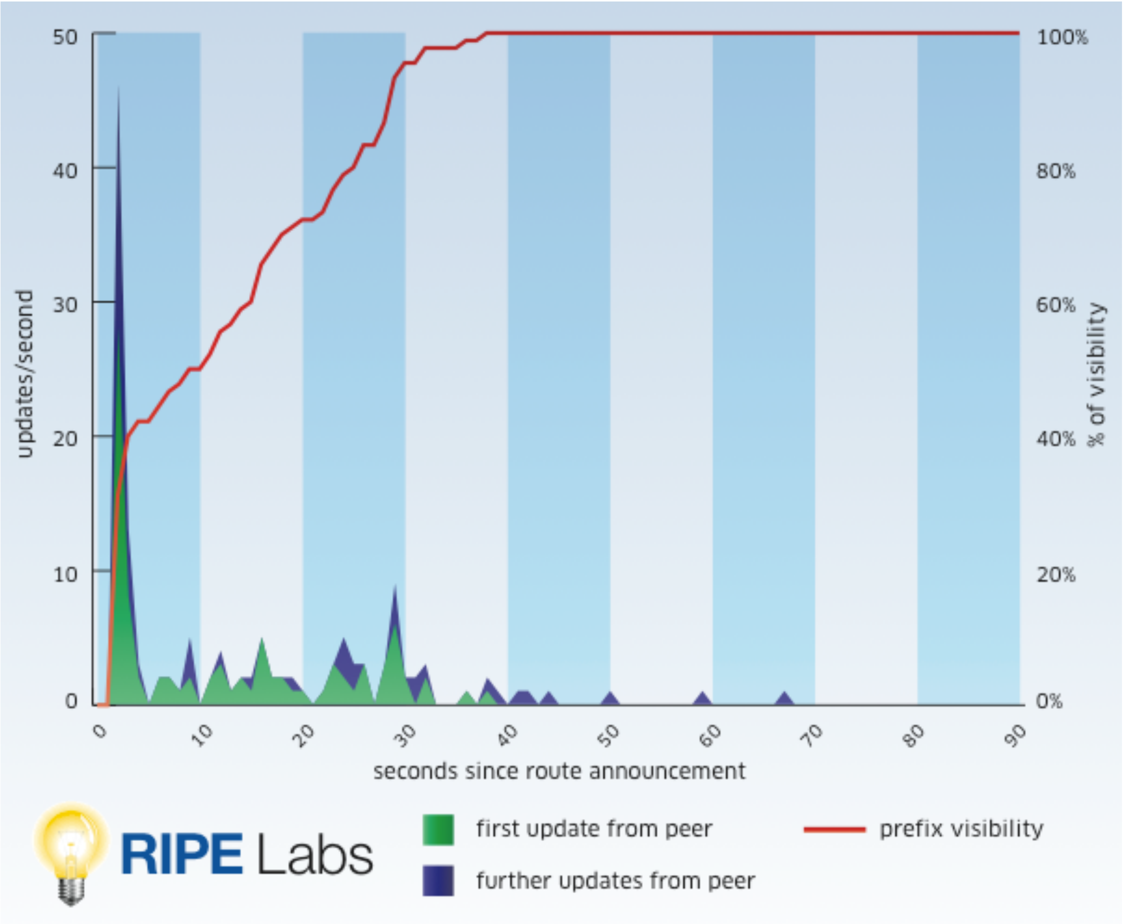
Prefix reachability studies

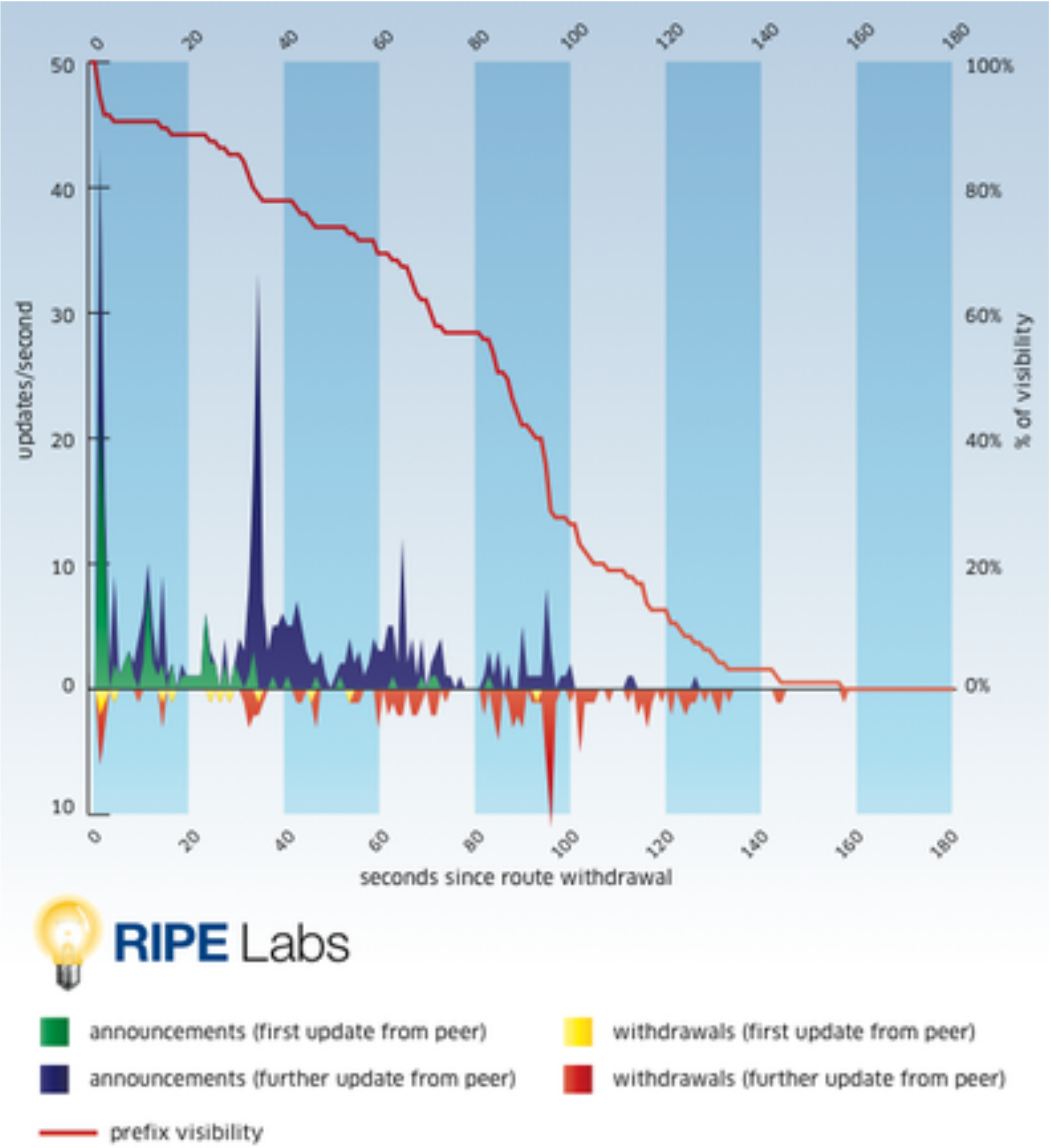
- https://labs.ripe.net/Members/emileaben/has-the-routability-of-longer-than-24-prefixes-changed



BGP update propagation

- https://labs.ripe.net/Members/vastur/the-shape-of-a-bgp-update





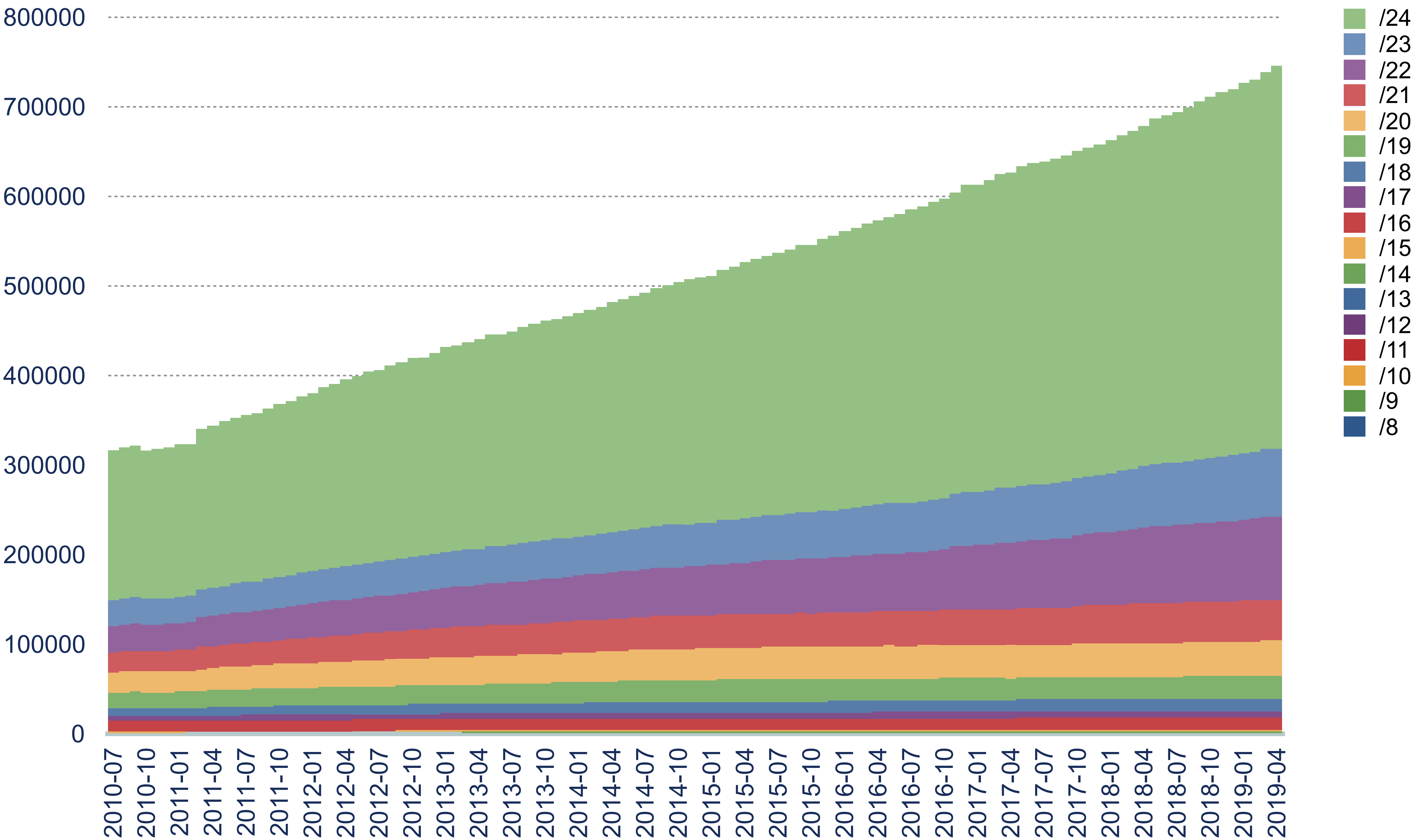
RIS growth

Because the internet keeps growing

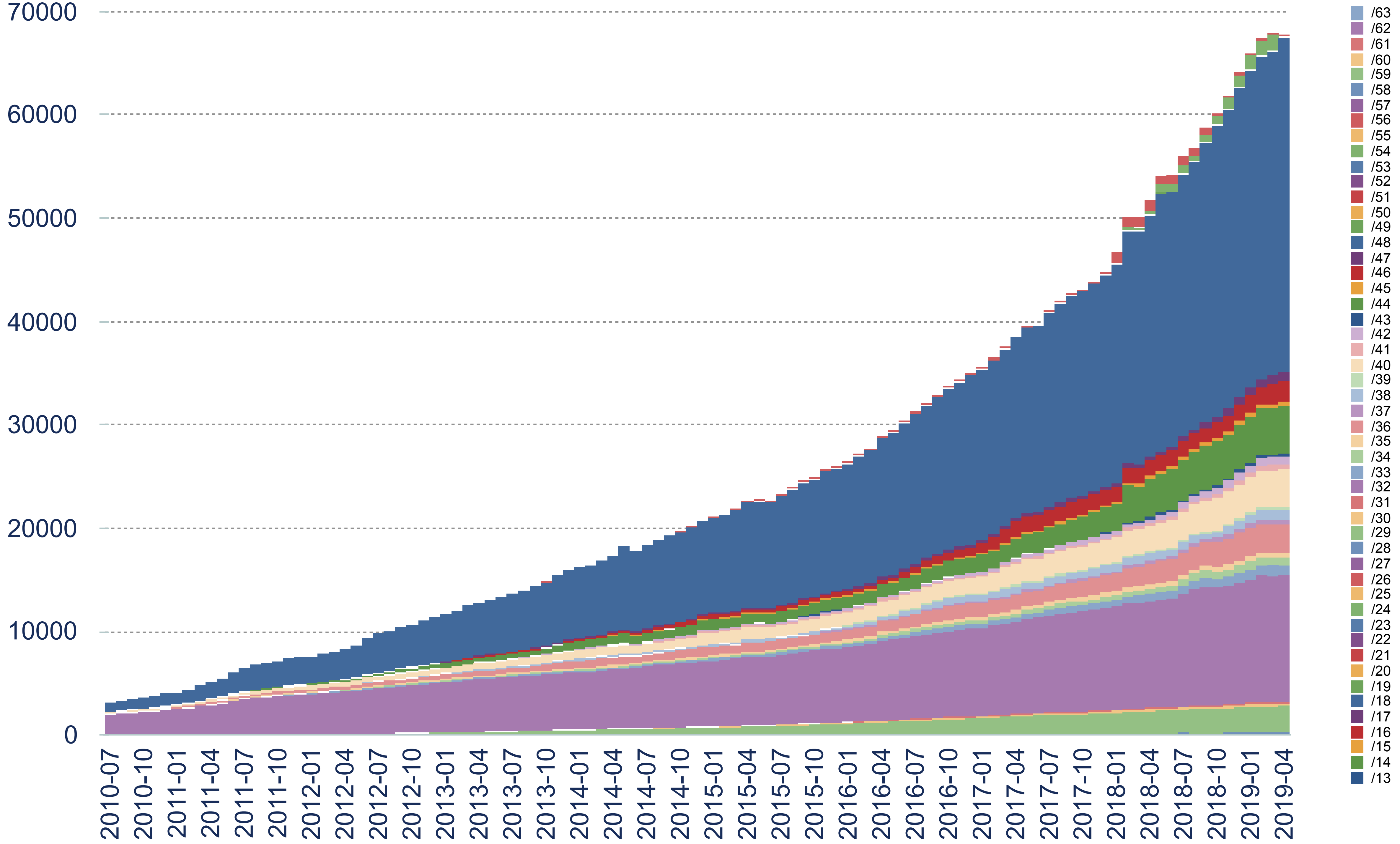
Collector history

Collector	Location	IXP	Deployed	Removed
RRC00	Amsterdam	Multi-hop	1999	
RRC01	London	LINX	2000	
RRC02	Paris	SFINX	2001	2008
RRC03	Amsterdam	AMS-IX	2001	
RRC04	Geneva	CIXP	2001	
RRC05	Vienna	VIX	2001	
RRC06	Tokyo	DIX-IE	2001	
RRC07	Stockholm	Netnod	2002	
RRC08	San Jose	MAE-West	2002	2004
RRC09	Zurich	TIX	2003	2004
RRC10	Milan	MIX	2003	
RRC11	New York	NYIIX	2004	
RRC12	Frankfurt	DE-CIX	2004	
RRC13	Moscow	MSK-IX	2005	
RRC14	Palo Alto	PAIX	2005	
RRC15	Sao Paulo	PTT-Metro SP	2006	
RRC16	Miami	NOTA	2008	
RRC18	Barcelona	CATNIX	2015	
RRC17				
RRC19	Johannesburg	NAPAfrica JB	2016	
RRC20	Zurich	SwissIX	2015	
RRC21	Paris	FranceIX	2015	
RRC22	Bucharest	InterLAN	2017	
RRC23	Singapore	Equinix SG	2017	
RRC24	Montevideo	LACNIC multi-hop	2019	

Number of IPv4 prefixes seen



Number of IPv6 prefixes seen

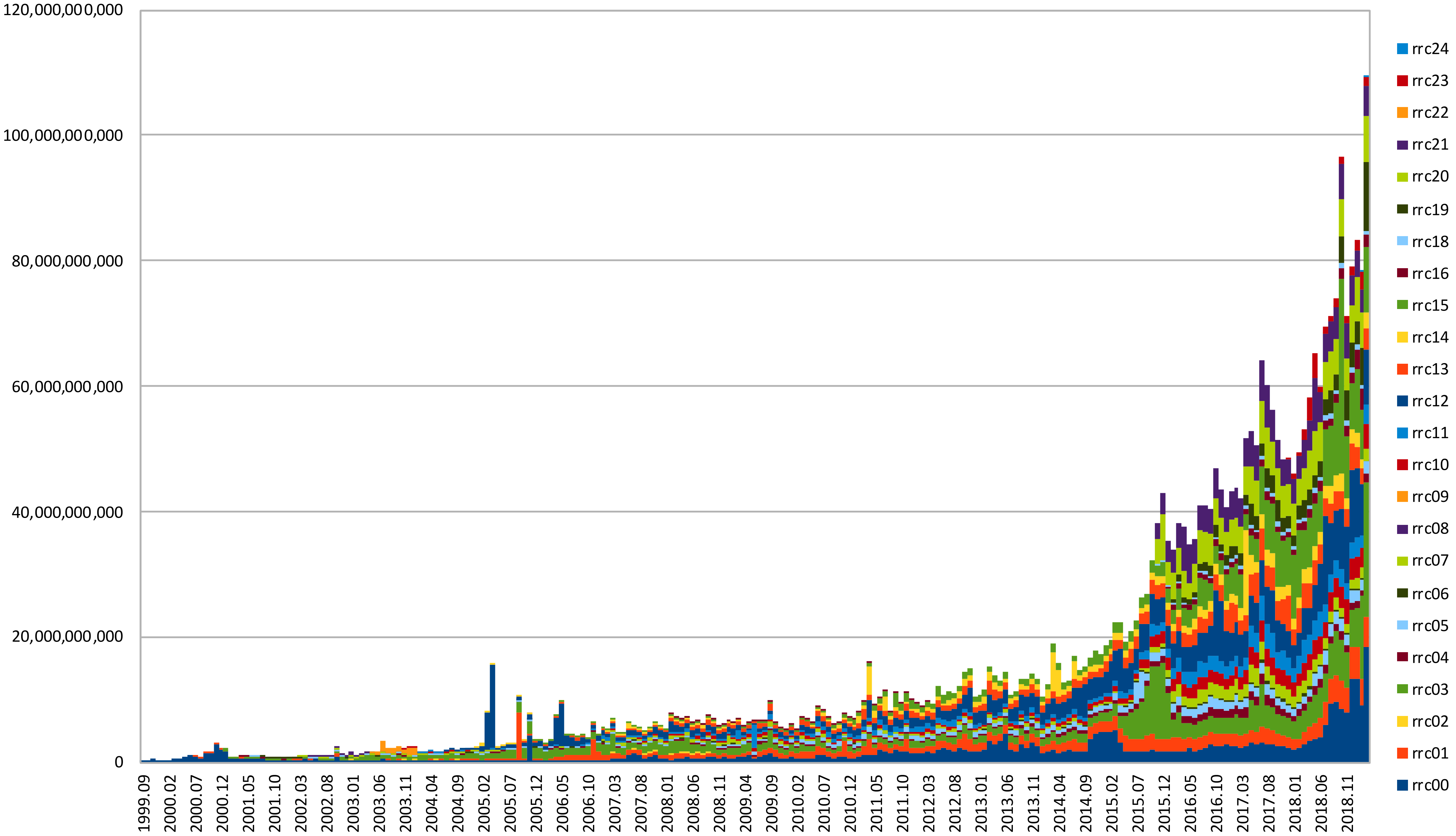


Data growth

- More BGP data
  - BGP table has grown from 60,000 to 800,000 routes
  - BGP updates happen more often
  - larger RIB (table) dumps
- More RIS collectors
- More peers at each collector

- Non-linear growth curve ;)

Compressed BGP updates per month



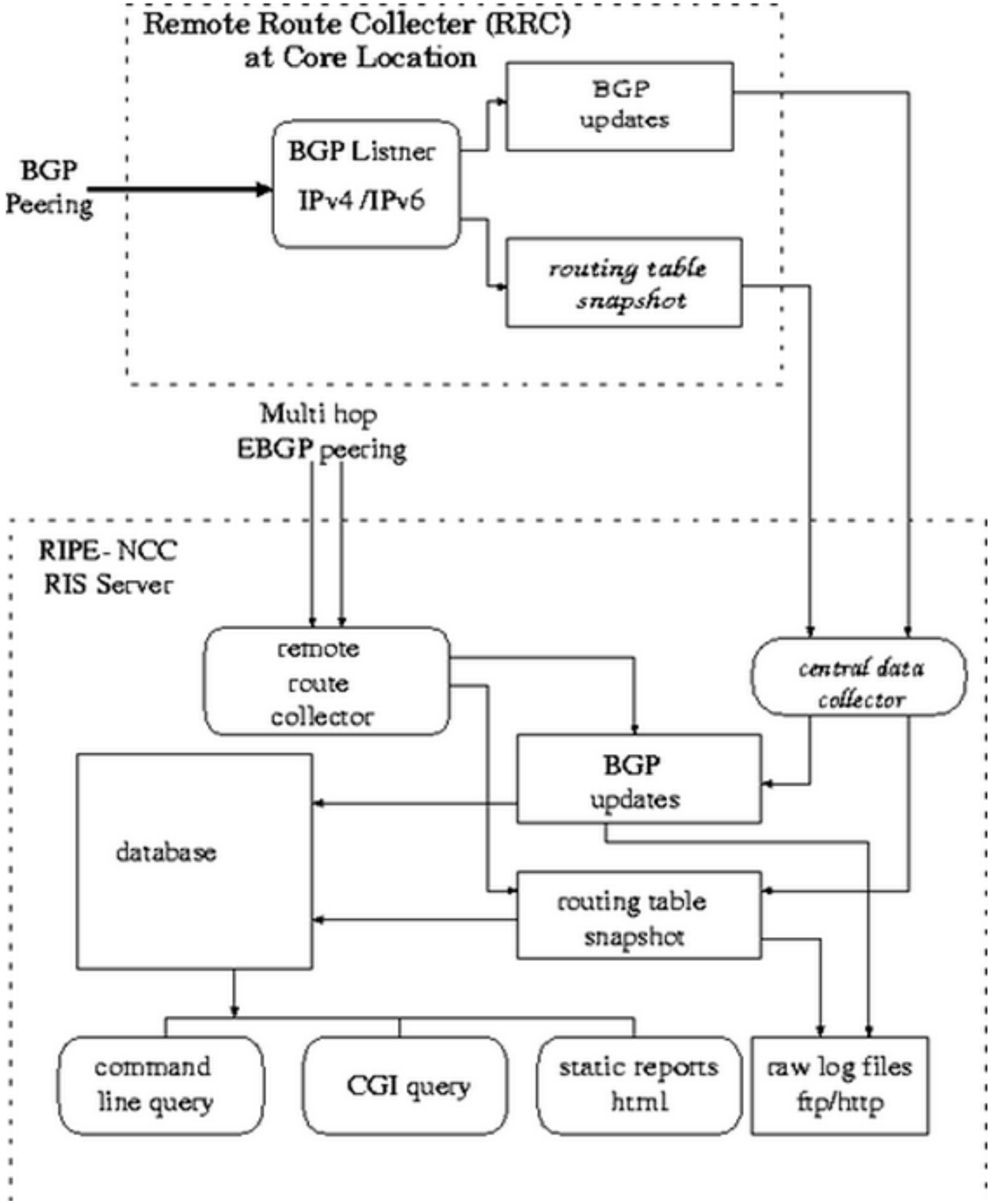
The future of RIS

- Increasing the number of peers on existing collectors does not bring much value
- Adding new collectors in Europe / North America does not bring much value either
- Adding new collectors in other regions proved difficult / economically unjustifiable
- Possible solutions:
  - add multi-hop collectors?
  - connect route servers?
  - collect data from different sources?
- Please provide your feedback! – ris@ripe.net

RIS Operations

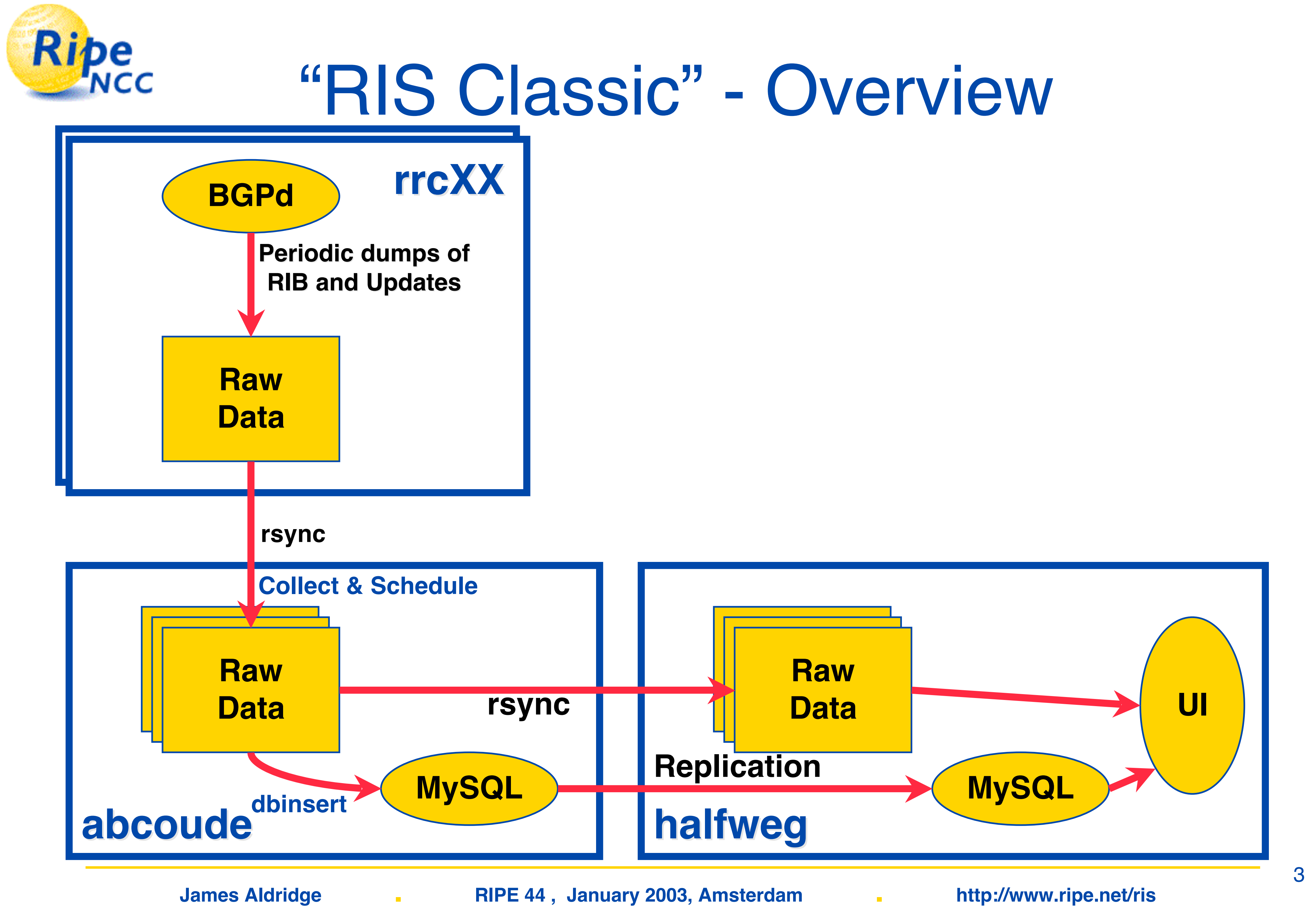
As the system has evolved

Original architecture (1999)



- Diagram from RIPE-200 (original concept)
- Note “*RIS Server*”
  - singular!
- Also, “*the database*”
  - this becomes the hardest part!!

“Classic” architecture (2003, 9 collectors)



“RIS Classic” - Overview

rrcXX

BGPd

Periodic dumps of RIB and Updates

Raw Data

rsync

Collect & Schedule

Raw Data

rsync

dbinsert

MySQL

abcoude

Raw Data

UI

Replication

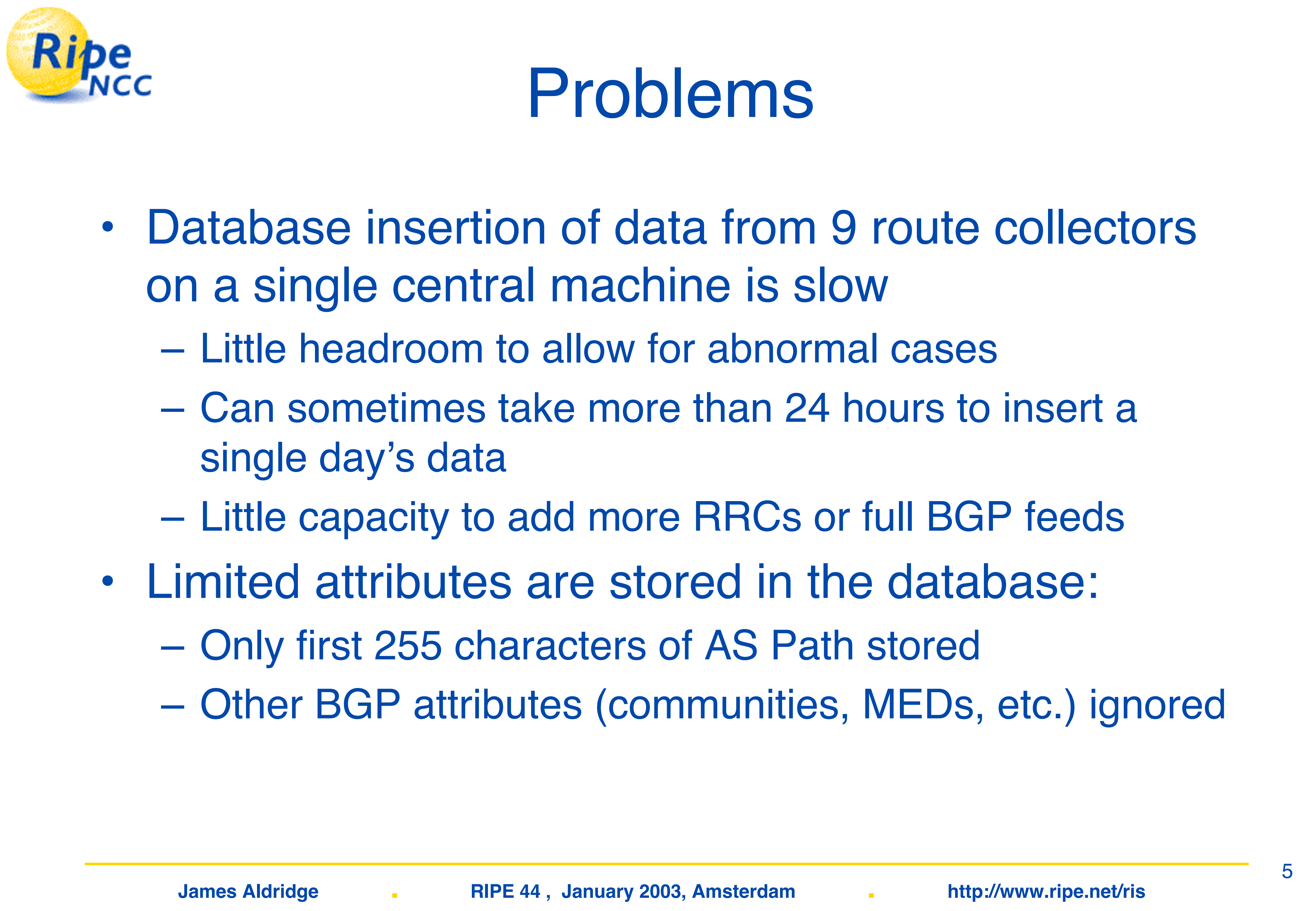
MySQL

halfweg

3

James Aldridge . RIPE 44 , January 2003, Amsterdam . http://www.ripe.net/ris

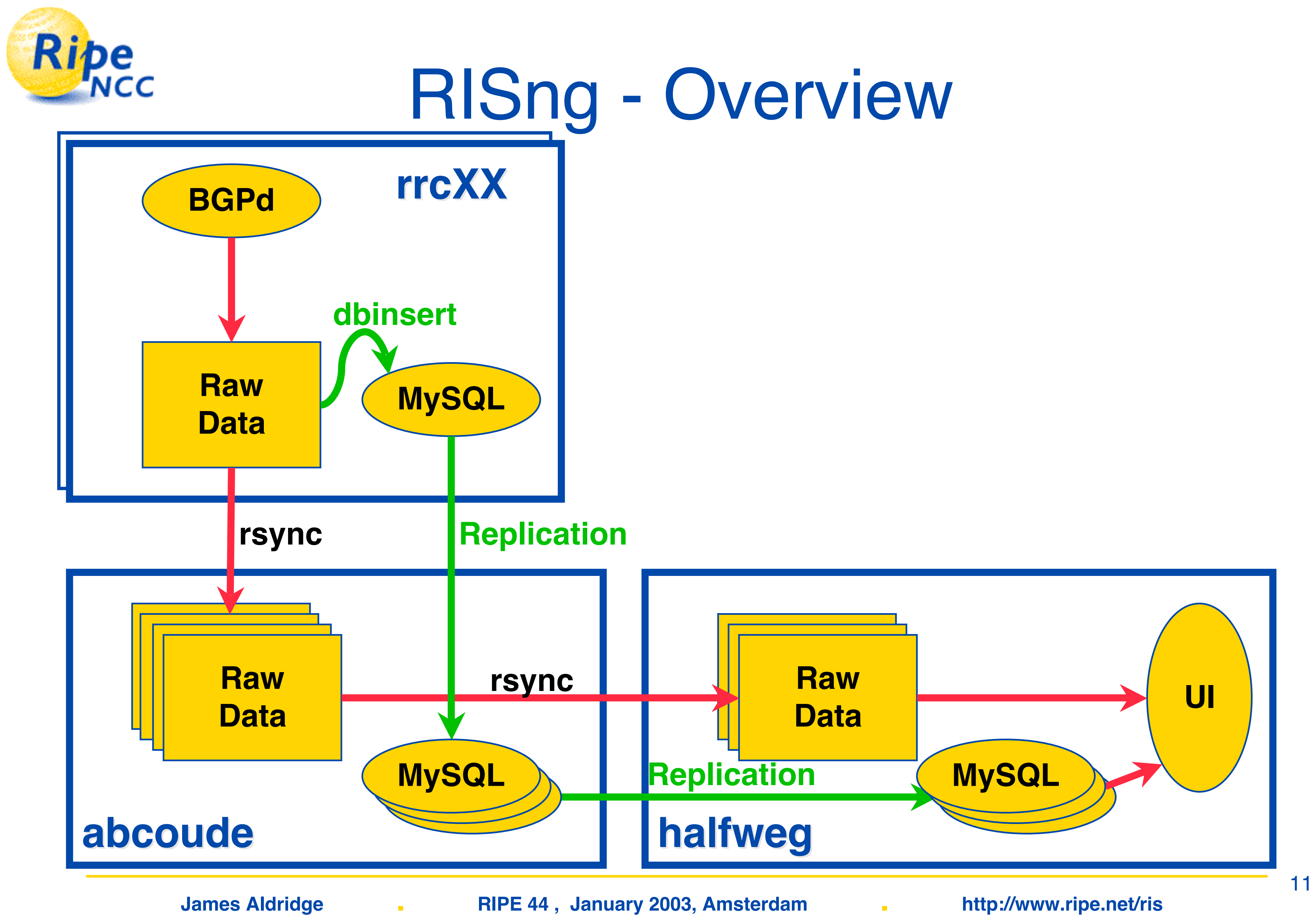
“Classic” architecture (2003, 9 collectors)



Problems

- Database insertion of data from 9 route collectors on a single central machine is slow
  - Little headroom to allow for abnormal cases
  - Can sometimes take more than 24 hours to insert a single day’s data
  - Little capacity to add more RRCs or full BGP feeds
- Limited attributes are stored in the database:
  - Only first 255 characters of AS Path stored
  - Other BGP attributes (communities, MEDs, etc.) ignored

“RISng” architecture (2003, 9 collectors)



RISng - Overview

James Aldridge . RIPE 44 , January 2003, Amsterdam . http://www.ripe.net/ris

11

Scaling the Database

- MySQL: splitting and sharding
  - 8 MySQL servers
  - some collectors were so big they needed their own MySQL server!
- Data retention
  - database was only query-able for 3 months worth of data
  - the references grew too large, that every 3 months we basically had to drop all the data, and let it start again
- Time for Big Data!

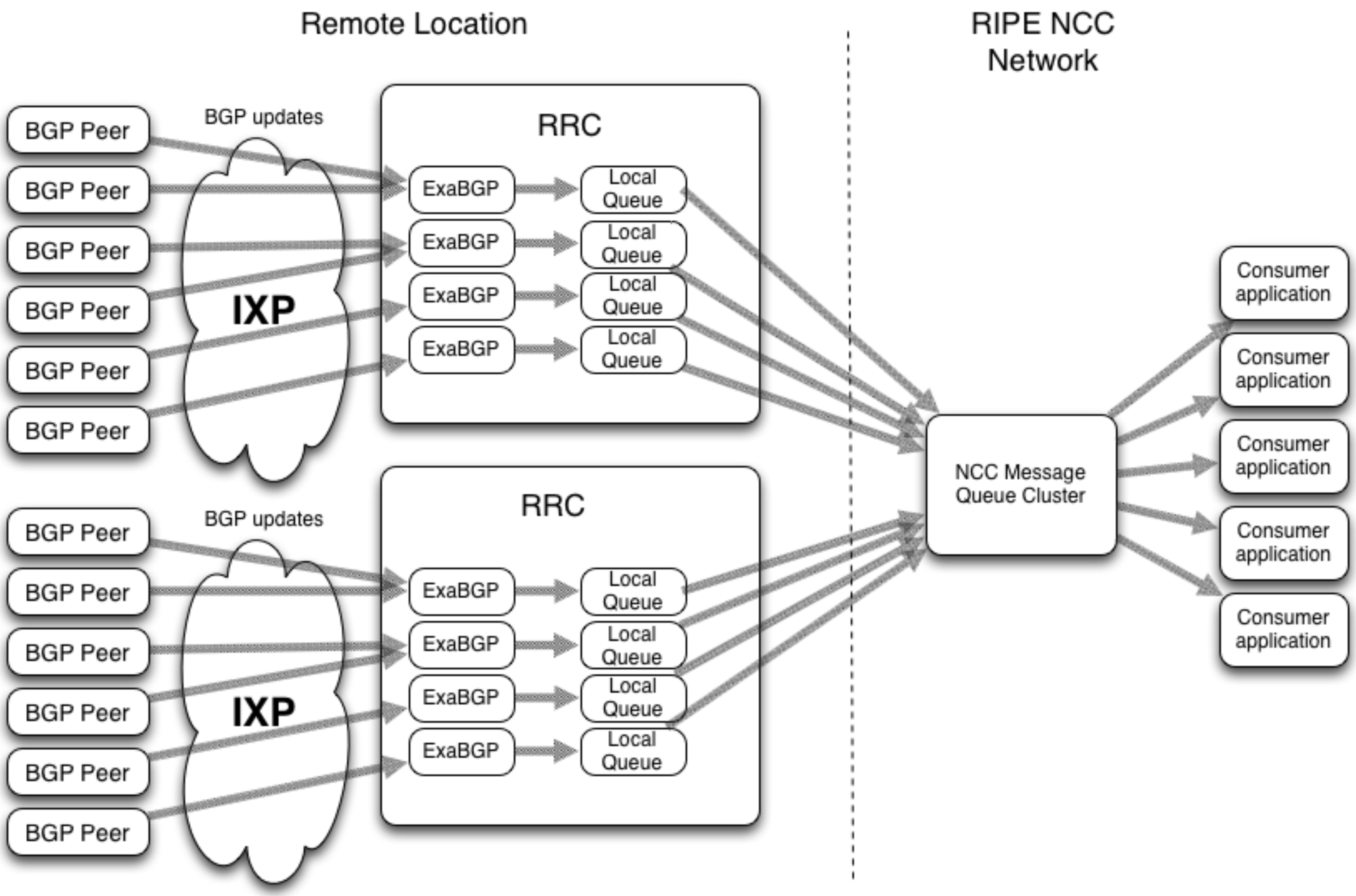
Scaling the collectors

- Quagga used as BGP collector
- Single-threaded
  - Not as scalable on modern multi-core CPUs
- Locks updates during table-dump process
  - Requires that dump completes before the hold timer expires, or BGP session will drop
- Some data consistency issues
  - Sometimes updates are missing from the update dumps at the time of a table dump
  - This makes it difficult to accurately rebuild BGP state at a intermediate time, if updates are not reliable in-between

RIS operations

Time for a redesign
(and this is the current design!)

Data collection



Big Data processing

- Apache Hadoop
  - An open-source software framework for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware
- Allows us to build a scalable storage and processing cluster
  - Attributes and aggregations for all historical data are available
- Currently over 150 servers in the cluster
  - Although the cluster is not only used for RIS
  - Also used by RIPE Atlas and other projects

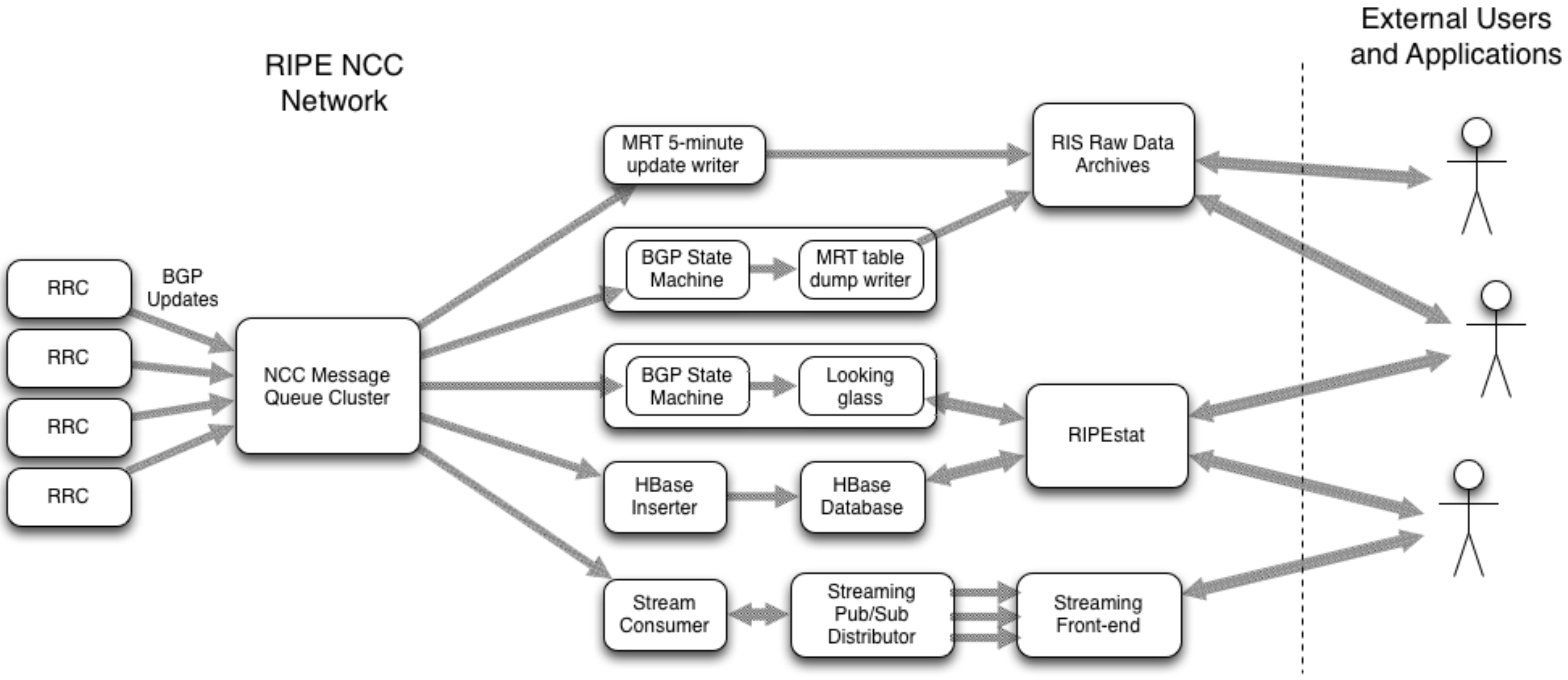
Big Data processing – components

- HDFS
  - distributed, replicated, cluster filesystem
- HBase
  - non-relational distributed database
  - large tables – billions of rows × millions of columns
- Map/Reduce
  - massive batch job processing
- Kafka
  - Message Queue and stream processing

New architecture

- Multiple BGP daemons (ExaBGP) – at least 1 per core
  - lightweight daemon
  - finally could saturate RRC server
- Message Queue
  - RabbitMQ ➠ Kafka
- Stream processing
  - Looking Glass
  - RIS Live
  - raw updates files
  - RISwhois – in progress
- Batch processing
  - aggregations

Back-end data distribution



Questions

http://bit.ly/esnog23-ris-feedback

https://ripe.net/ris

https://stat.ripe.net

ris@ripe.net