

Santa: “Making a (prefix) list and checking it twice”

Marzetti, Marco
Pavlidis, Adam

Presentation Outline

- Introduction
- Problem statement
- Generic Overview
- Internal Details
- Summary

Introduction

This presentation

- **PCCW Global, a.k.a. AS3491**

- Global tier-1 network
- Services include but not limited to:
IP Transit, MPLS, Mobility/Voice, Satellite and Security

- **Console Connect: Software-defined Interconnection Provider**

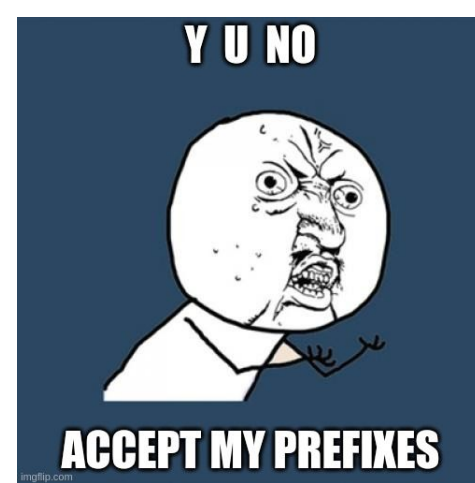
- L2/L3 VPN, IP Transit and IoT
- Automated, self-provisioning
- Demarcation: On-premise ↔ DC ↔ Cloud (SaaS / IaaS providers) ↔ Mobile clients

Problem Statement

Reliable lifecycle management: **inbound prefix-lists** (IP Transit)

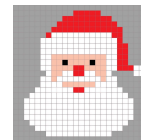
- Manual ops = Mundane & Error-prone
- Large scale => inherent-complexity
- Faster mgmt loops: **minutes/hours** not **days/weeks**
- Source-of-Truth (SoT) ??? => IRR records
 - Multiple regions => Multiple RIRs => more complexity
 - Unnecessary?: **#route objects > # advertised routes**
 - Flat-out wrong: **Accidental / On purpose**
 - Stale/Outdated

SUM(ALL_THE_ABOVE) = **Absolute need for automation**



Generic Overview

Santa

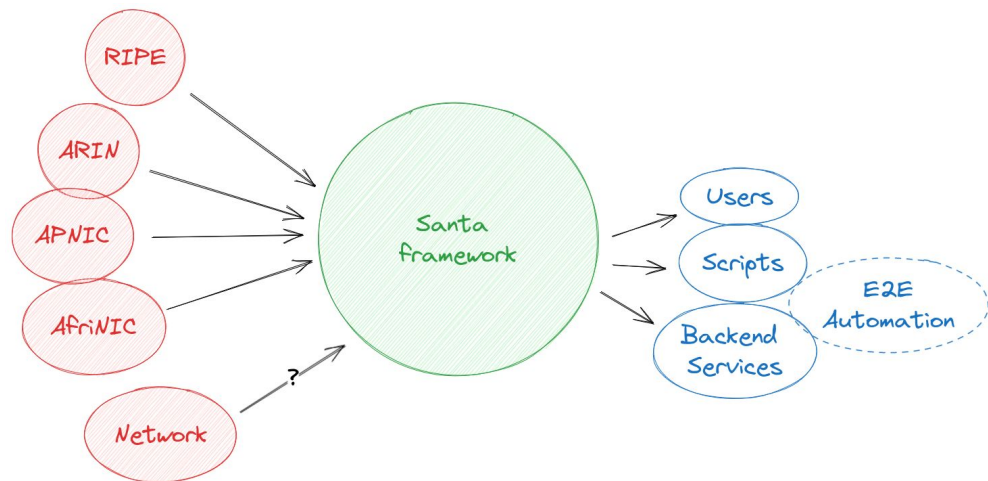


Santa: “Making a (prefix) list and checking it twice”

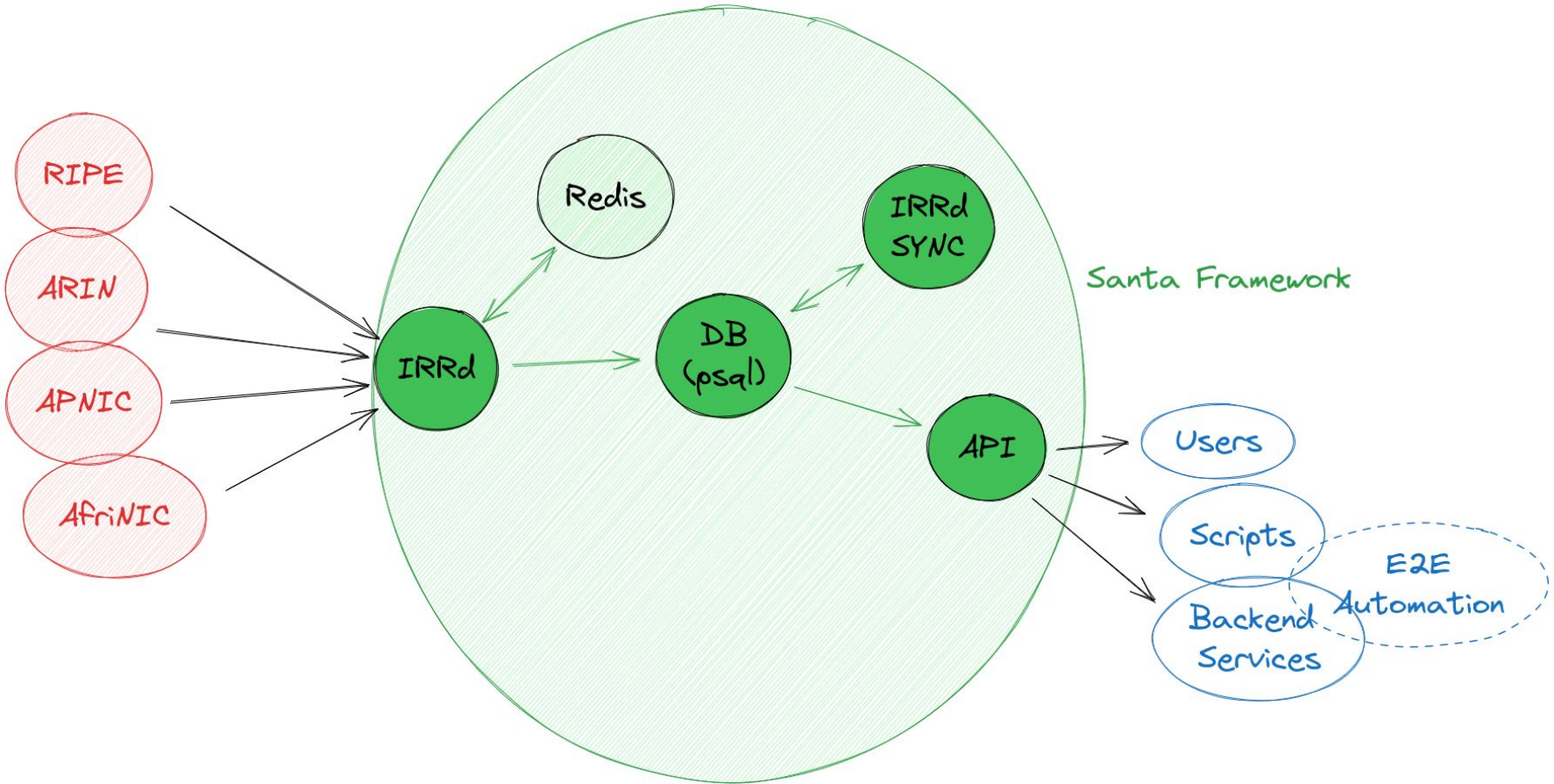
- Process IRR data (RIRs, RADB etc)
 - Local mirror: avoid remote query rate limiting
 - Sanitize data / apply preference: e.g. source RIPE >> source RADB
 - Need for performance

- **AS-SETs rule the waves:**

- **Customer** provided AS-SET (*mk1*)
- **Dynamic** - WYSIWYG (*mk2*)
Process data streams
- API: expose data to users/services/tools
 - AS-SET => Origin ASNs & Prefixes
 - ASNs => Prefixes



Detailed Architecture 1/4



Detailed Architecture 2/4

Santa mk1:

- Is already a production service (multiple years)
- Enables automated/self-provisioned IP transit

IRRd: local mirror of IRR data (route, route6, as-set)

IRRd-SYNC:

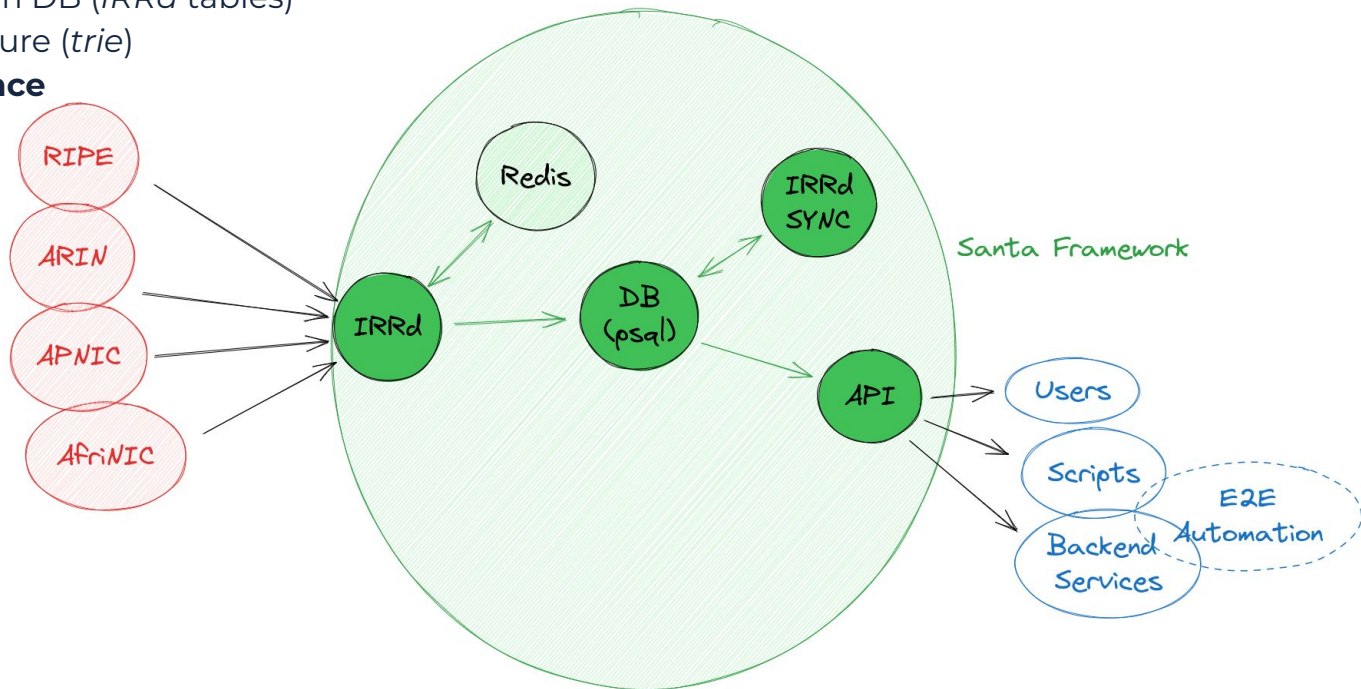
1. Imports & Parses **data** from DB (IRRd tables)
2. Maintains “RIB-like” structure (trie)
3. Validates/applies **preference**
e.g. RIPE > RADB
4. Saves data to DB (Santa tables)

API: Expose data:

1. GET ORIGINS per dstIP
2. GET PREFIXES per ASN (+ORIGINS)
3. GET ASNs per AS-SET (+PREFIXES, +ORIGINS)

Important primitive: **ORIGIN**

triple(t) (PREFIX, ASN, srcIRR)

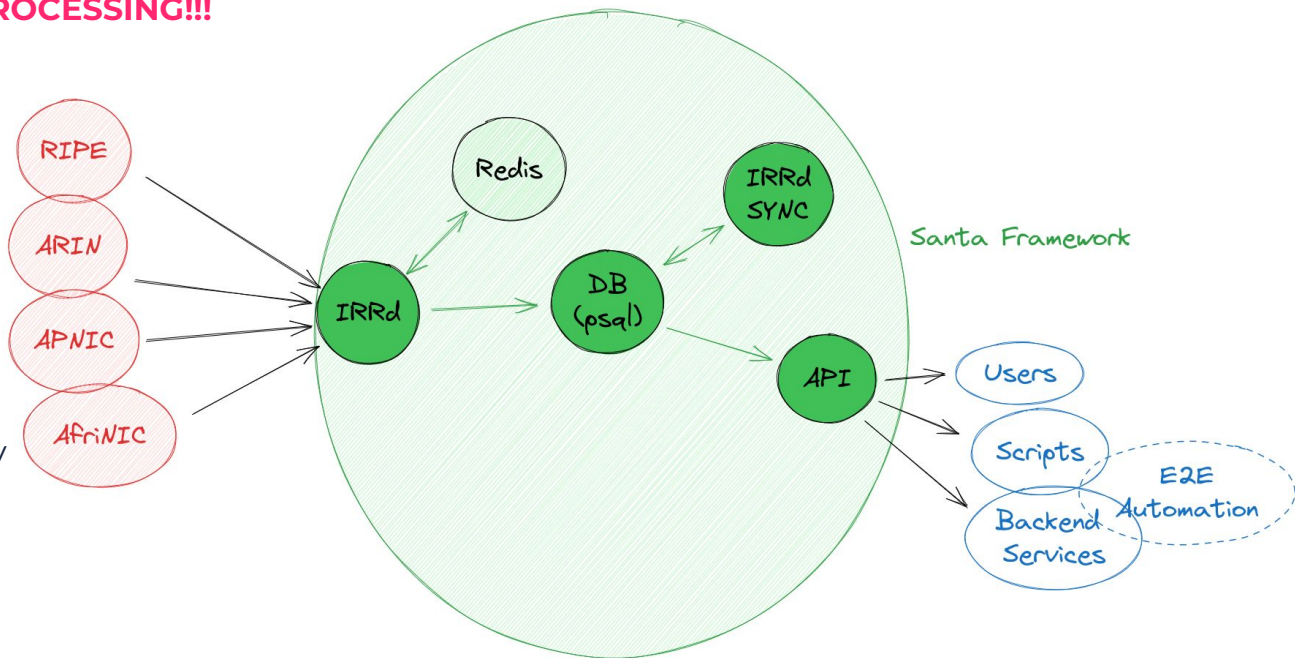


Detailed Architecture 3/4

YOU PROMISED US STREAM-like PROCESSING!!!

Why?

1. DYNAMIC & Customer AS-SETS
2. IRR data vs reality:
Concise and Exact prefix lists ?
3. (near)? Real-time observability
 - a. Highlight: problematic advertisements
 - b. Protect/Identify: errors and/or malicious acts
 - c. Facilitate: troubleshooting



Detailed Architecture 4/4

Some processing examples:

- Mark & Reject
 - VIP ASNs
 - VIP Prefixes
 - Problems (Loops)
- “Minified” pfx list
Needs real-time update

Santa mk2 (in testing):

How:

1. **BMP (per session)**
2. **BGP (per RIB/per router)**

=> **Data pipeline:**

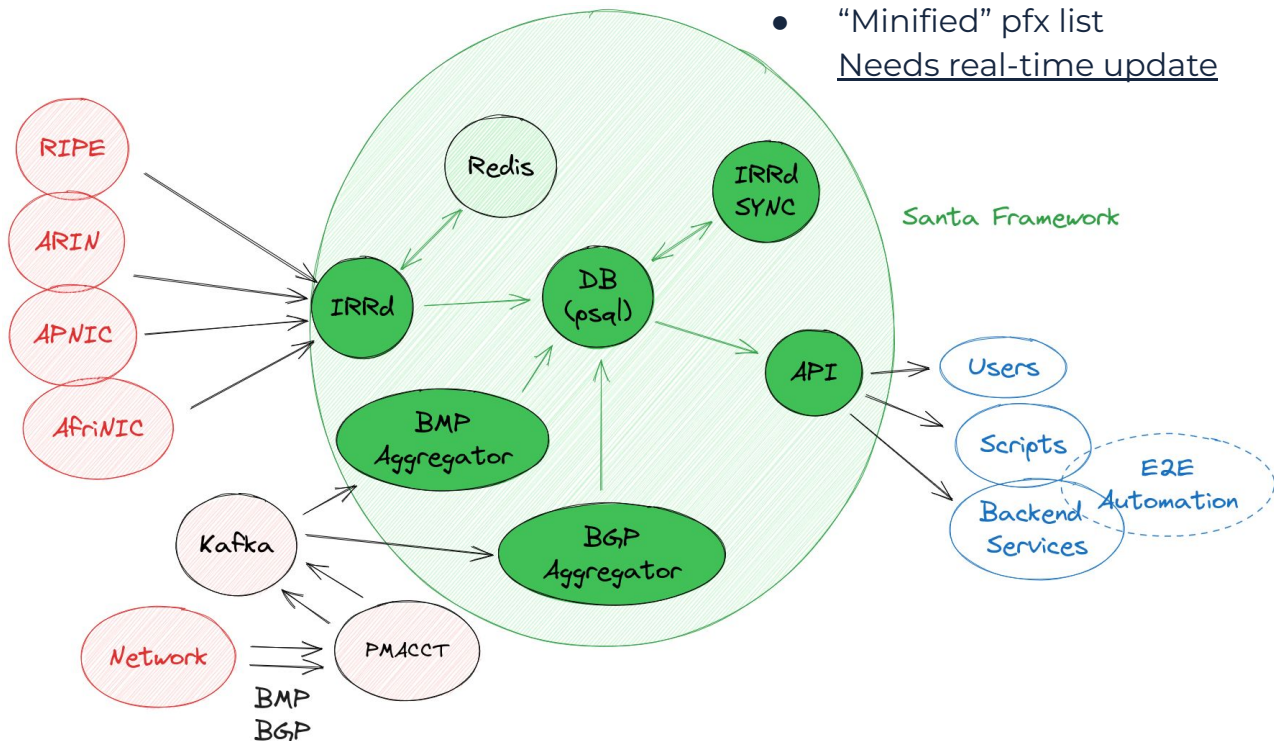
1. Network devices => PMACCT
2. PMACCT => Kafka (JSON/AVRO)

=> **Aggregators:**

3. Consume from Kafka
4. Process & Validate data
5. Aggregate & Insert to DB

=> **API:**








6. Additional endpoints



Summary / Interesting Facts

- *Santa* under the hood:
 - Python: heavy use of async libraries & constructs
 - Pluggable (micro)services: Docker & Kubernetes ready
 - Parallel processing (Aggregators): Deploy replicas
 - TEXT (JSON) vs BINARY (AVRO): versatility vs performance
- Invested heavily on CI/CD pipelines: unit tests, on-commit/merge, package images
- BMP vs BGP: why both?
- Data demographics:
 - AS3491: 180+ Border IP routers
 - BMP data - Customer sessions only (not peers)
 - *log* (steady-state): ~100s+ messages/sec
 - *dump* (burst): ~10000s+++ messages/sec

Summary / Challenges

- Fault-tolerant & distributed applications = awesome. RIGHT ?
Equally “awesome” & “interesting” problems / bugs :-)
- Cross-disciplinary skill-set required    
- Testing & Debugging   
 - #L unit tests >>> #L code
 - **“But_It_Worked_On_My_Laptop”** vs **Reality**
 - Abstractions => hide the *complexity* (still there)
- Log handling (Processing, Analysis, Alerting): must-have



THANK YOU!

TAKE CONTROL | CUT COMPLEXITY | MAKE INTERCONNECTIONS EFFORTLESS

AUSTRALIA

Level 3 | 200 Mary Street | Brisbane QLD 4000 | Australia

UNITED KINGDOM

6/F Exchequer Court | 33 St. Mary Ax | London EC3A 8AA | UK

FRANCE

5/F 223 Rue Saint-Honoré | 75001 Paris | France

GREECE

340 Kifisias Avenue/340 Olimpionikon | Neo Psychiko 154 51 | Athens | Greece

UNITED STATES – EAST COAST

Unit 3120 | 31/F Empire State Building | New York | NY 10118 | USA

UNITED STATES – WEST COAST

201 Redwood Shores Pkwy | Suite 175 | Redwood City | CA 94065 | USA

SINGAPORE

6 Temasek Boulevard | #41-04A/05 | Suntec Tower Four | 038986 | Singapore

HONG KONG

20/F, Telecom House | 3 Gloucester Road | Wan Chai | Hong Kong

JAPAN

3rd Floor | Marunouchi Trust Tower Main | 1-8-3 Marunouchi | Tokyo 100-0005 | Japan

SOUTH AFRICA

Building 12 | 1 Woodmead Drive | Woodmead | Johannesburg 2191 | South Africa

UAE, DUBAI

Office 401 & 408 | Level 4 | Arjaan Business Tower | Dubai Media City | Dubai



TALK TO US:

sales@consoleconnect.com