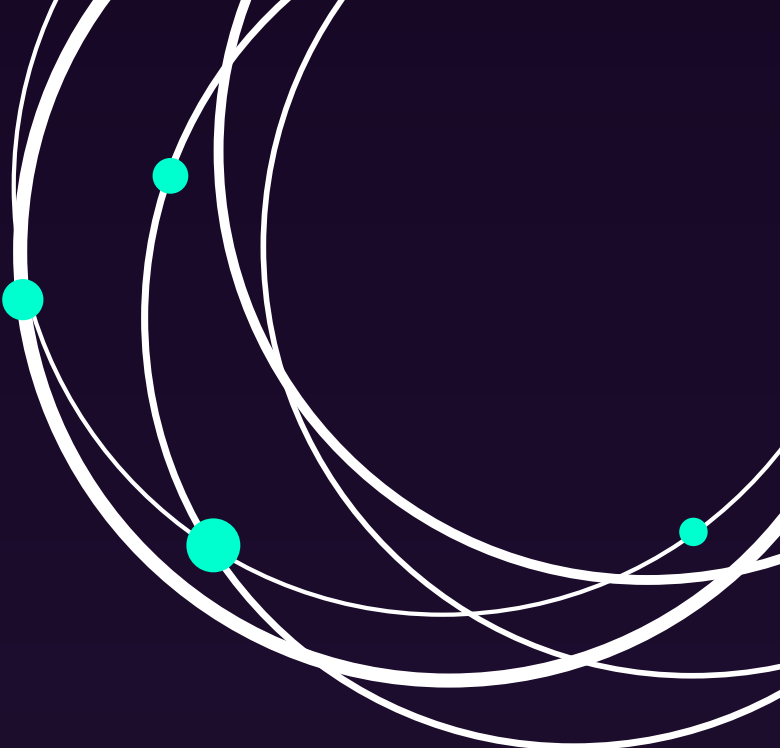




RIPE DB handling with NetBox

UKNOF50 London



What's the challenge?

Status quo



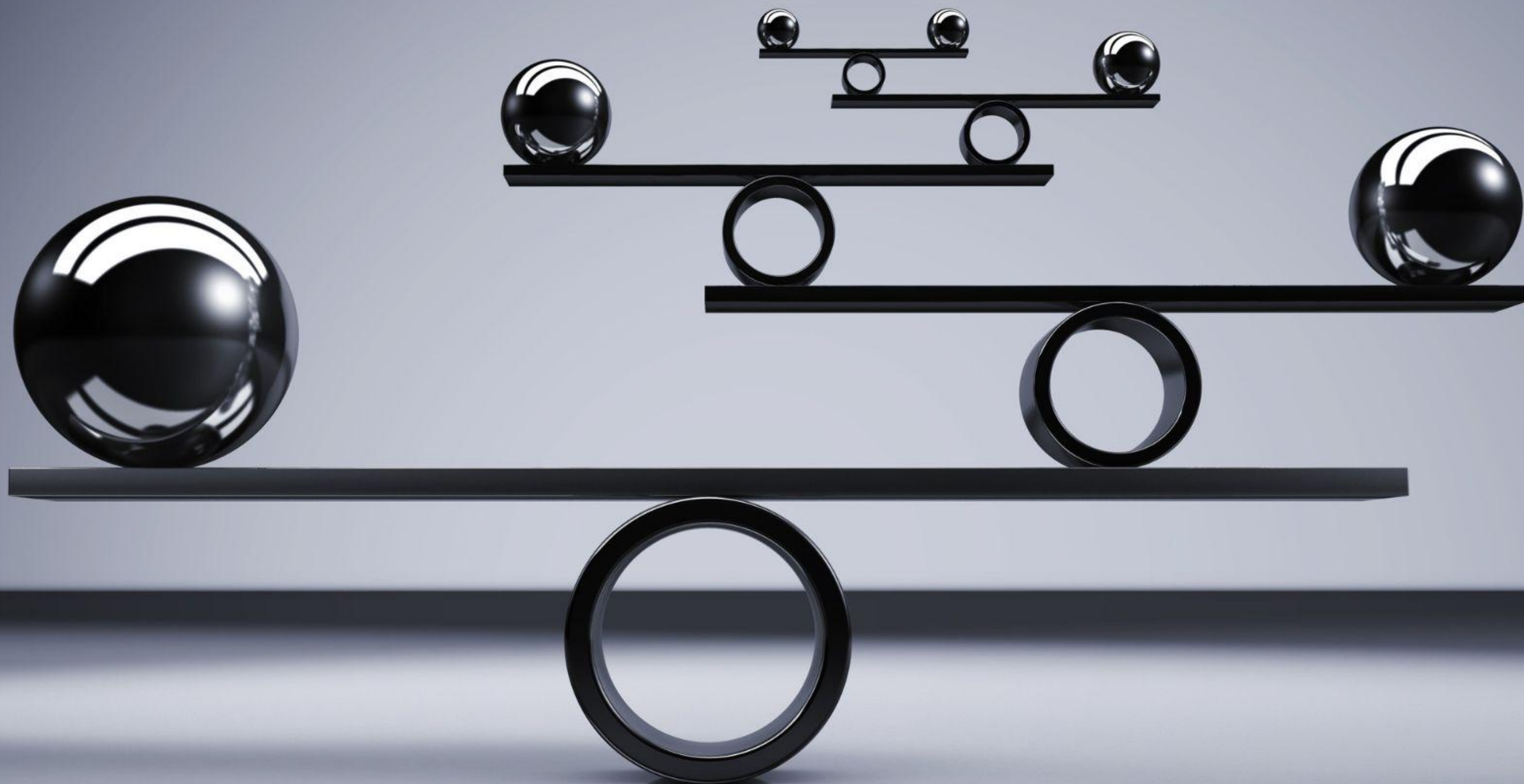
RIPE Policies



Short Name	Title
ripe-738	IPv6 Address Allocation and Assignment Policy
ripe-733	IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region
ripe-731	RIPE NCC IRR Database Non-Authoritative Route Object Clean-up
ripe-705	Abuse Contact Management in the RIPE Database
ripe-682	RIPE Resource Transfer Policies
ripe-679	Autonomous System (AS) Number Assignment Policies
ripe-639	RIPE NCC Services to Legacy Internet Resource Holders
ripe-637	Contractual Requirements for Provider Independent Resource Holders in the RIPE NCC Service Region
ripe-636	IPv6 Addresses for Internet Root Servers in the RIPE Region

Many prefixes, you are responsible for

RIPE policies to comply with





Super exiting work

- > manually...
- > updating...
- > inetnum...
- > objects...



But wait, there is an api for that...

NetBox Documentation

netbox-community/netbox

Webhooks

Table of contents

Webhooks

Configuration

Jinja2 Template Support

Available Context

Default Request Body

Webhook Processing

Troubleshooting

Webhooks

Warning

Configuration

• Name - A unique name for the webhook. The name is not included with outbound messages.

• Object type(s) - The type or types of NetBox object that will trigger the webhook.

• Enabled - If unchecked, the webhook will be inactive.

• Events - A webhook may trigger on any combination of create, update, and delete events. At least one event type must be selected.

WHOIS REST API

Ed Shryane edited this page on 16 Apr · 73 revisions

Introduction

Welcome to the RIPE Database REST API documentation.

For more information about the REST paradigm, see https://en.wikipedia.org/wiki/Representational_state_transfer.

If you used the old (beta) API, consider reading the [migration guide for old API users](WHOIS REST API Migration Guide).

All the services are accessible via HTTPS.

Use of the Whois REST API is governed by the RIPE Database [terms and conditions](#).

RESTful URI format

Each object in the RIPE Database has a unique locator URI, in the following format:

`https://rest.db.ripe.net/{source}/{objecttype}/{key}`

Where:

- Source is the database source (e.g. RIPE).
- Objecttype is the object type (e.g. inetnum)
- Key is the primary key(s)
 - Normally key is a single attribute value.
 - Use the `nic-hdl` attribute value for person or role object types.
 - Combine the `route6` attribute value and `origin` attribute value for route or route6 object types (e.g. route 193.0.22.0/23AS3333).

Supported methods:

- POST: create
- GET: lookup
- PUT: update
- DELETE: delete

Additional services

- search

But they don't match

NetBox

prefix: 198.51.100.0/25

tenant: cust1

aggregate: 198.51.100.0/24

...



RIPE DB

inetnum: 198.51.100.0 - 198.51.100.127

netname: **X?**

desc: **X?**

admin-c: **X?**

tech-c: **X?**

...



Looking for a solution



ripe-updater

ripe-updater

Features

ripe-updater

Features

- › Automate INET(6)NUM objects

ripe-updater

Features

- › Automate INET(6)NUM objects
- › Validation

ripe-updater

Features

- › Automate INET(6)NUM objects
- › Validation
- › Templates

ripe-updater

Features

- › Automate INET(6)NUM objects
- › Validation
- › Templates
- › Backups

ripe-updater

Features

- › Automate INET(6)NUM objects
- › Validation
- › Templates
- › Backups
- › Reporting

Requirements



NetBox

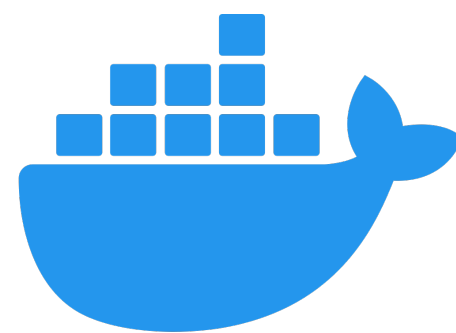
Version 2.4 or later

Add some custom fields



RIPE Database

Access with maintainer password



Platform

Able to run python code and a reverse proxy
docker-image also available

Add a new prefix



Prefix	
Prefix	198.51.100.0/25 IPv4 or IPv6 network with mask
Status	Active ✕ ▾ Operational status of this prefix
Description	cust1 ☐ Is a pool All IP addresses within this prefix are considered usable

Custom Fields	
RIPE Report	True ✕ ▾
RIPE Template	----- ▴ CUST-TRANSFER-NET INFRA-MGMT-NET INFRA-ROUTING-NET INFRA-SERVICE-APP-NET INFRA-TRANSFER-NET

Tags
Tags

Create

Create and Add Another

Cancel

Workflow

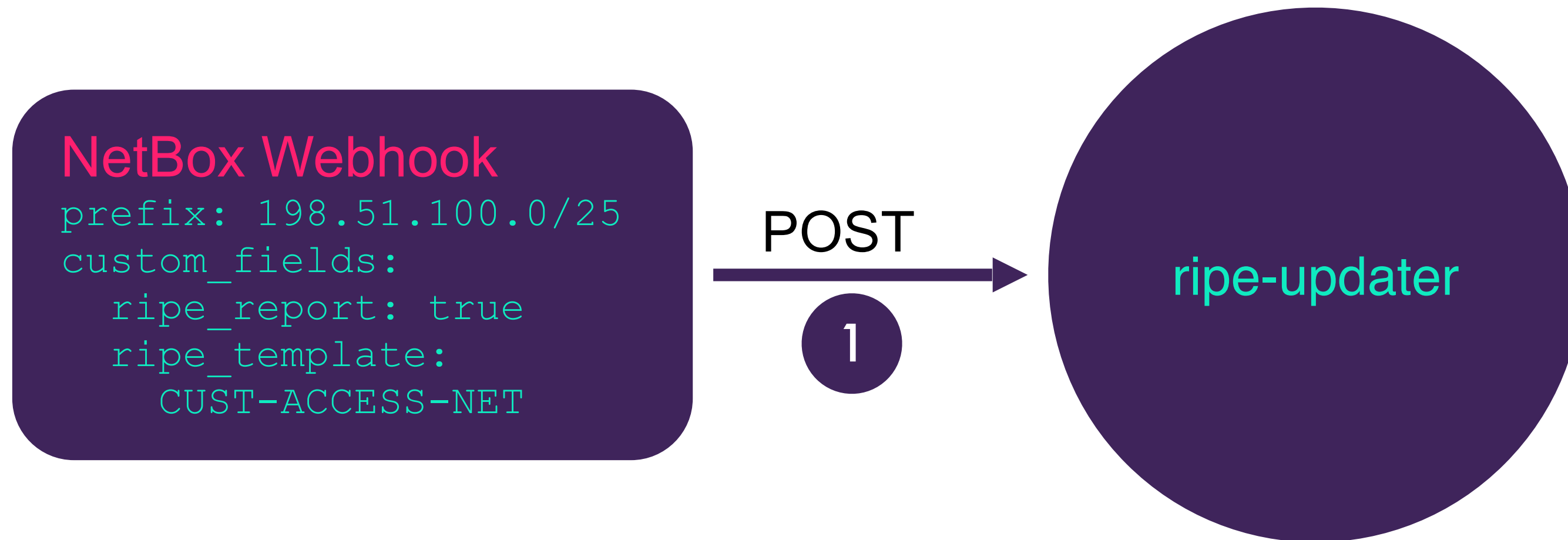
NetBox Webhook

```
prefix: 198.51.100.0/25
custom_fields:
  ripe_report: true
  ripe_template:
    CUST-ACCESS-NET
```

POST

1

ripe-updater



Workflow

NetBox Webhook

```
prefix: 198.51.100.0/25
custom_fields:
  ripe_report: true
  ripe_template:
    CUST-ACCESS-NET
```

POST

1

ripe-updater

read

2

Template("CUST-ACCESS-NET")

```
{ "attributes": [
  {"remarks": "Managed by ripe-updater"},
  {"admin-c": "AA1-TEST"},
  {"tech-c": "AA1-TEST"},
  {"mnt-by": "TEST-DBM-MNT"},
  {"source": "TEST"},
  ...
]}
```

Workflow

NetBox Webhook

```
prefix: 198.51.100.0/25
custom_fields:
  ripe_report: true
  ripe_template:
    CUST-ACCESS-NET
```

POST

1

ripe-updater

read

2

Template("CUST-ACCESS-NET")

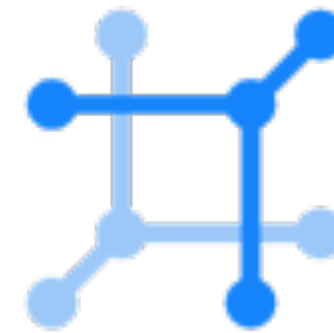
```
{ "attributes": [
  {"remarks": "Managed by ripe-updater"},
  {"admin-c": "AA1-TEST"},
  {"tech-c": "AA1-TEST"},
  {"mnt-by": "TEST-DBM-MNT"},
  {"source": "TEST"},
  ...
]}
```

search
overlap

3

validate
overlap

4



Workflow

NetBox Webhook

```
prefix: 198.51.100.0/25
custom_fields:
  ripe_report: true
  ripe_template:
    CUST-ACCESS-NET
```

POST

1

ripe-updater

read

2

Template("CUST-ACCESS-NET")

```
{ "attributes": [
  {"remarks": "Managed by ripe-updater"},
  {"admin-c": "AA1-TEST"},
  {"tech-c": "AA1-TEST"},
  {"mnt-by": "TEST-DBM-MNT"},
  {"source": "TEST"},
  ...
]}
```

search
overlap

3

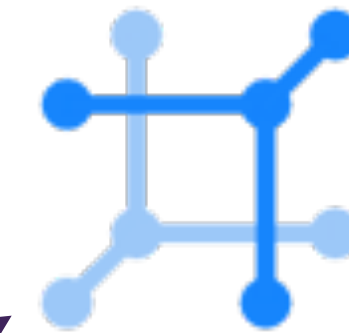
validate
overlap

4

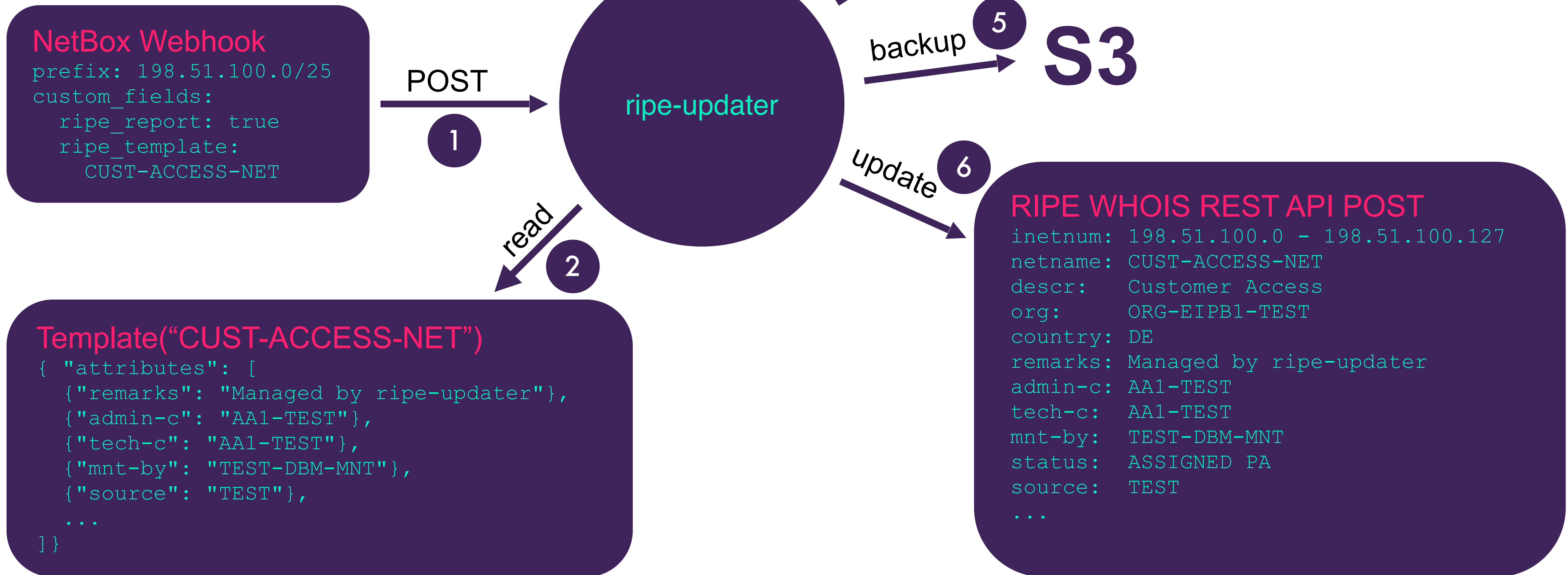
backup

5

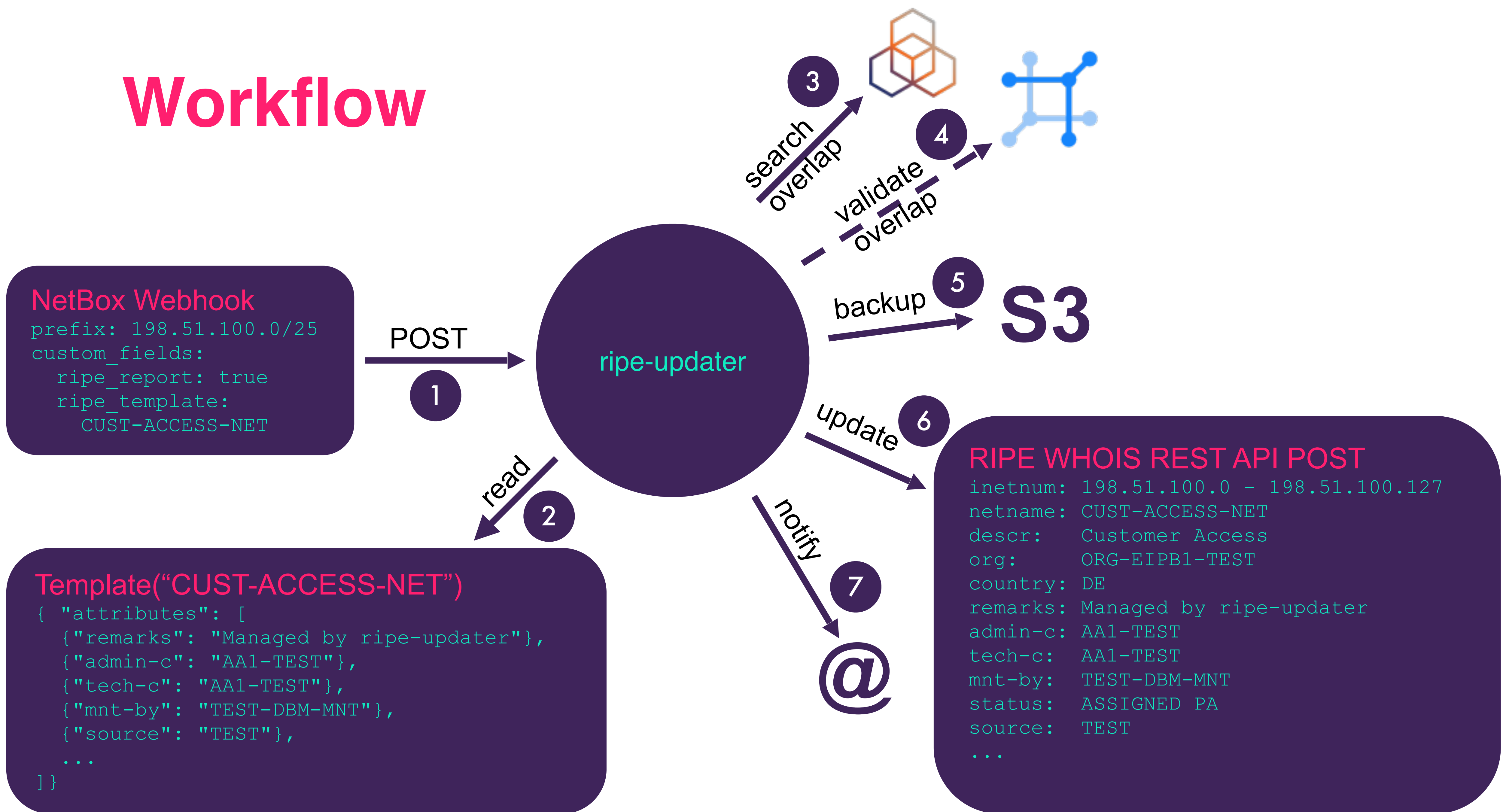
S3



Workflow



Workflow



With ripe-updater

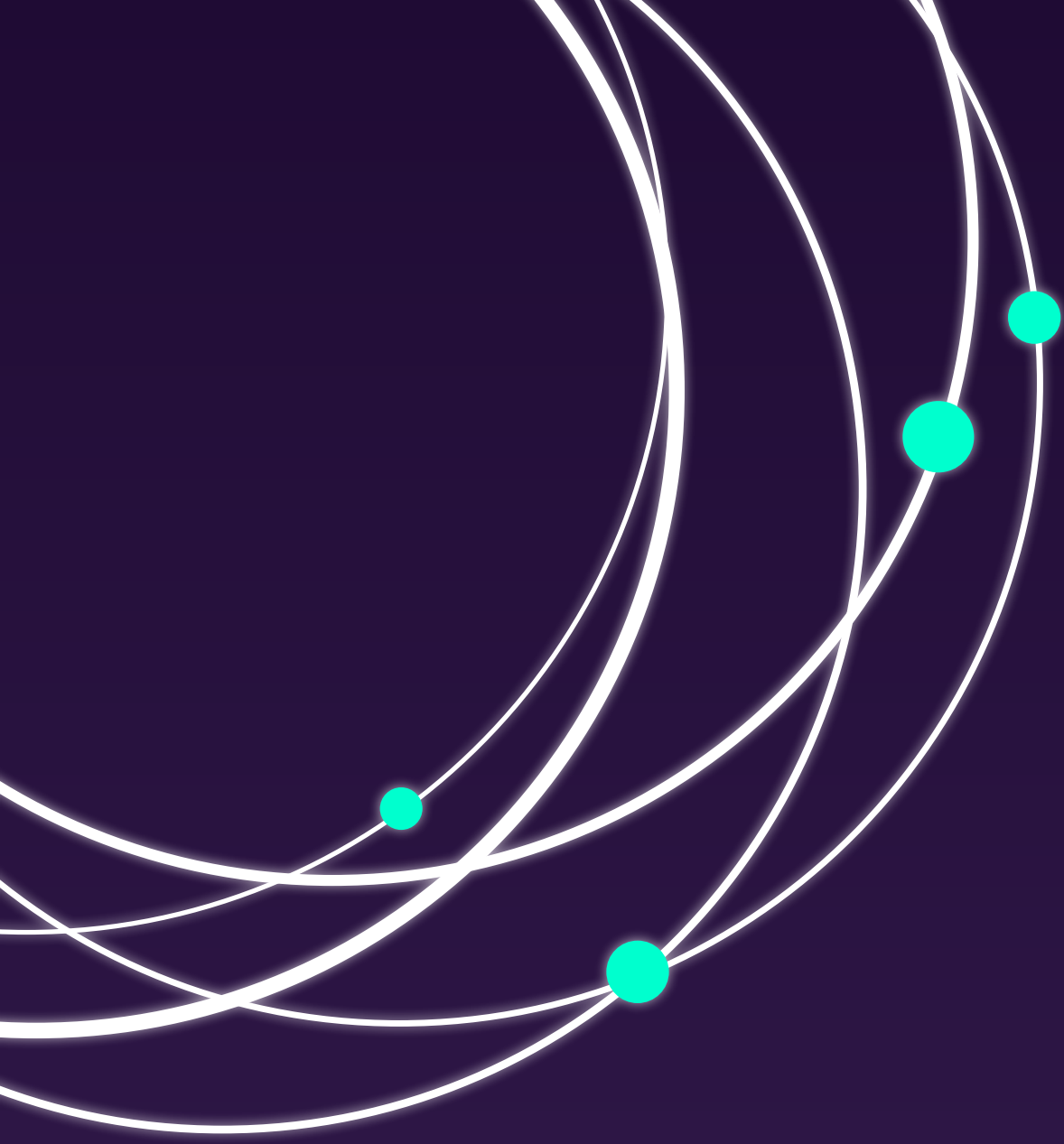
- ✓ Database is always updated
- ✓ Consistent objects, be relaxed on your ARC
- ✓ Batch processing of all your prefixes is no problem



Now on GitHub

<https://github.com/interdotlink/ripe-updater>

17.05.2022



<https://github.com/interdotlink/ripe-updater>

Thank you. Questions?

Get in touch: hello@inter.link / vincentz@inter.link

inter.link