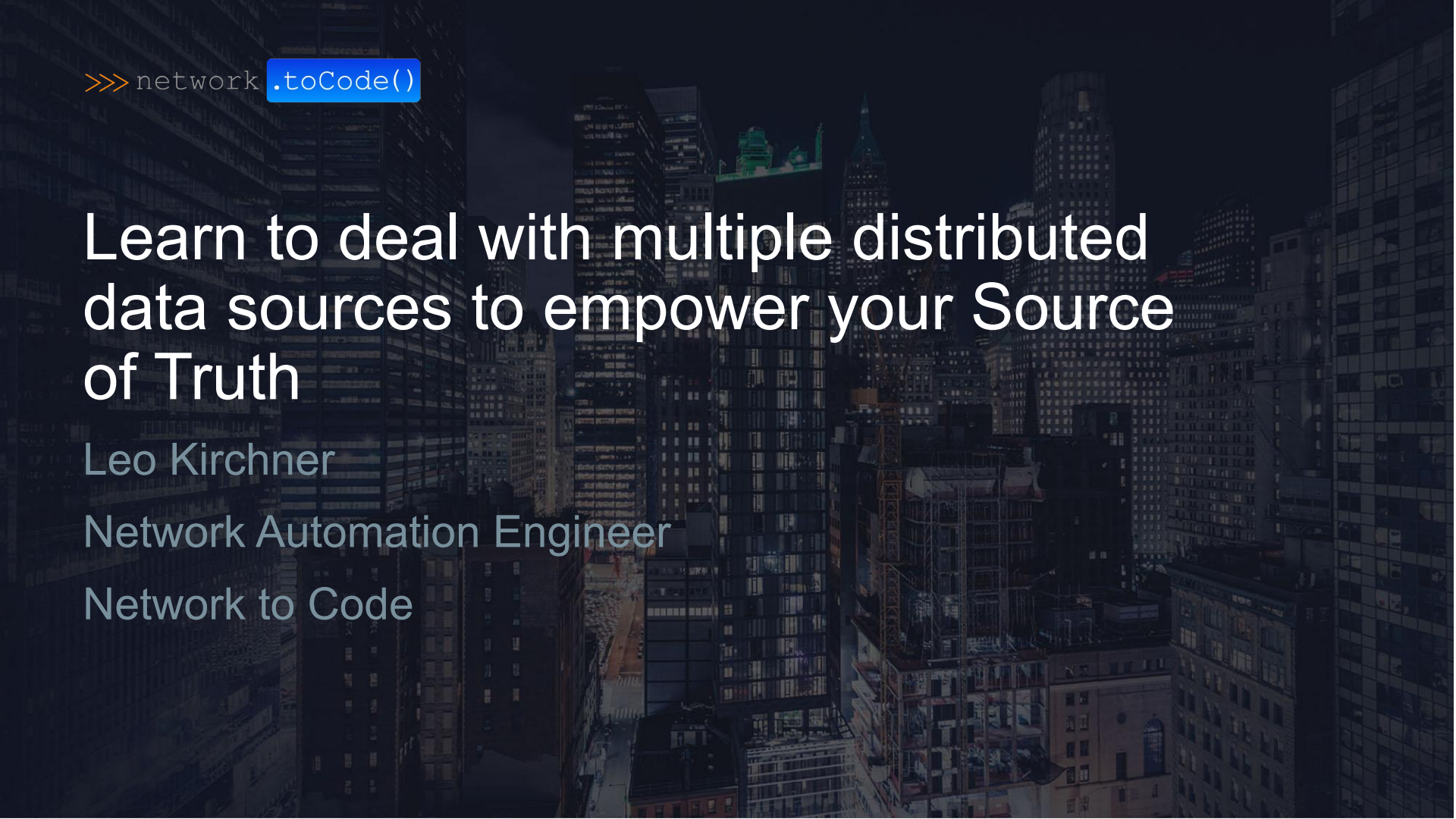


Learn to deal with multiple distributed data sources to empower your Source of Truth

Leo Kirchner

Network Automation Engineer

Network to Code

Who is Network to Code

Network Automation Solutions Provider

We are laser-focused on helping companies transform the way their networks are deployed, managed, and consumed using network automation and DevOps technologies.

A Diverse Team, with Deep Expertise

Engineers and developers in network automation, software and security, with leadership from vendors, integrators, and top tier consulting firms - all drive value to our clients.

Driven by Community & Industry Collaboration

Rooted in Community, NTC believes industry-wide collaboration is the catalyst needed for true innovation. Host 23,000+ members and 300+ channels at slack.networktocode.com

Industry-Recognized Thought Leaders

Working with clients across all industries and geographies, we promote a vendor- and tool-agnostic approach, making automation a reality for any network.


Source of Truth

Understanding Source of Truth

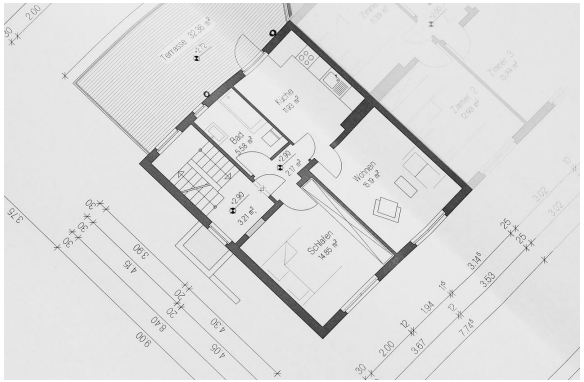
Actual State

All images are from Pixabay

>>> Understanding Source of Truth

Actual State

Intended State

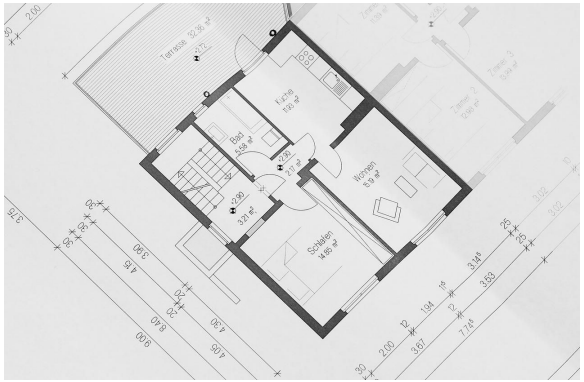


Source of Truth is all about **intentions** and planning, the expected state

All images are from Pixabay

Understanding Source of Truth

Intended State



Source of Truth is all about **intentions** and planning, the expected state

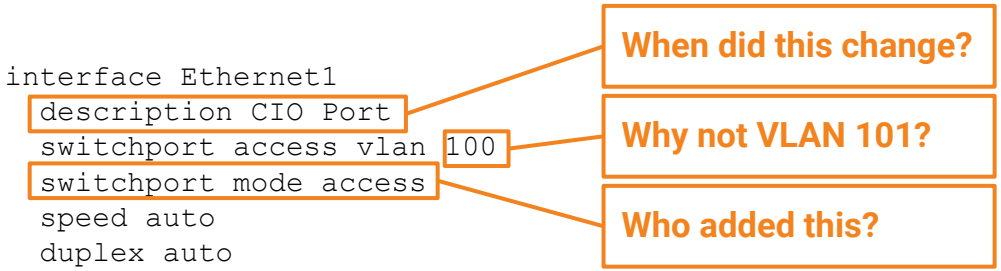
Actual State

All images are from Pixabay

>>> Intended State 101

```
interface Ethernet1
  description CIO Port
  switchport access vlan 100
  switchport mode access
  speed auto
  duplex auto
```

>>> Intended State 101



>>> Intended State 101

```
interface Ethernet1
  description CIO Port
  switchport access vlan 100
  switchport mode access
  speed auto
  duplex auto
```

When did this change?

Why not VLAN 101?

Who added this?

Programmatic access

```
interface {{ int['name'] }}
  description {{ int['desc'] }}
  switchport access vlan {{ int['vlan'] }}
  switchport mode access
  speed auto
  duplex auto
```

+

```
interfaces:
  - name: 'Ethernet1'
    desc: 'CIO Port'
    vlan: 100
```

=

```
interface Ethernet1
  description CIO Port
  switchport access vlan 100
  switchport mode access
  speed auto
  duplex auto
```

>>> Intended State, stored in a **Source of Truth**

Inventory

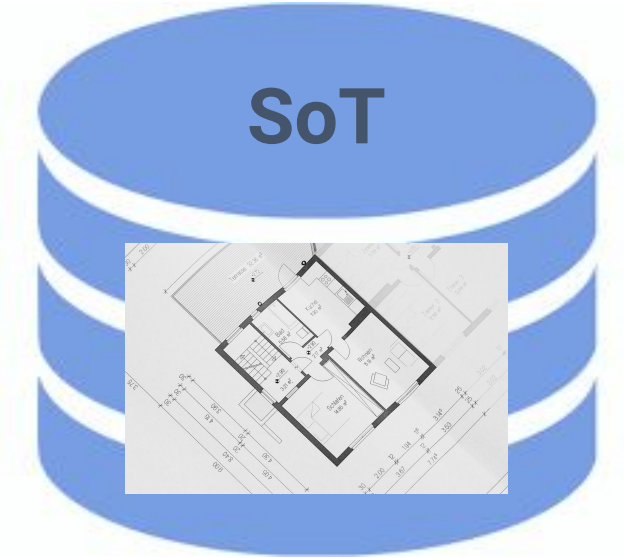
- Device List
- Software Version
- Hardware
- License

DCIM

- Location
- Racks
- Cables
- Power

IPAM

- IP Addresses
- IP Prefixes
- VLANs
- VRFs



Config Mgmt

- Templates
- Programmatic Interfaces

Circuits

- Circuits
- Providers
- Maintenances

Cloud Infrastructure

- Network Services

Network Properties

- NTP
- AAA
- DNS
- Routing
- ACLs


Single Source of Truth

>>> Systems of Record

Each Data Construct has an Authoritative Source

Inventory

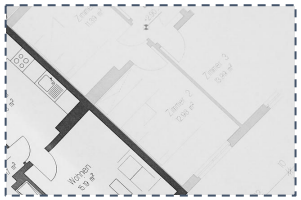
- Device List
- Software Version
- Hardware
- License

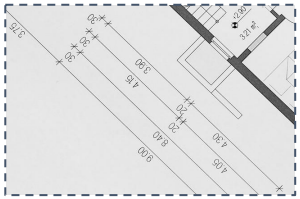
DCIM

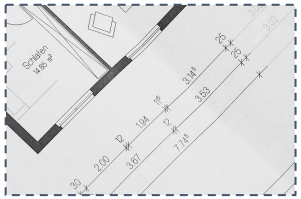
- Location
- Racks
- Cables
- Power

IPAM

- IP Addresses
- IP Prefixes
- VLANs
- VRFs







Config Mgmt

- Templates
- Programmatic Interfaces

Circuits

- Circuits
- Providers
- Maintenances

Cloud Infrastructure

- Network Services

Network Properties

- NTP
- AAA
- DNS
- Routing
- ACLs

>>> Systems of Record

Each Data Construct has an Authoritative Source

Inventory

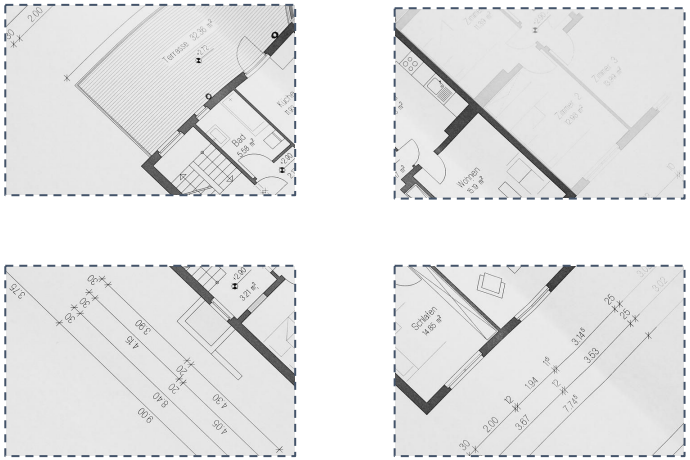
- Device List
- Software Version
- Hardware
- License

DCIM

- Location
- Racks
- Cables
- Power

IPAM

- IP Addresses
- IP Prefixes
- VLANs
- VRFs



Multiple sources of information

Incompatible data models

Inconsistent data

Distributed authority

Config Mgmt

- Templates
- Programmatic Interfaces

Circuits

- Circuits
- Providers
- Maintenances

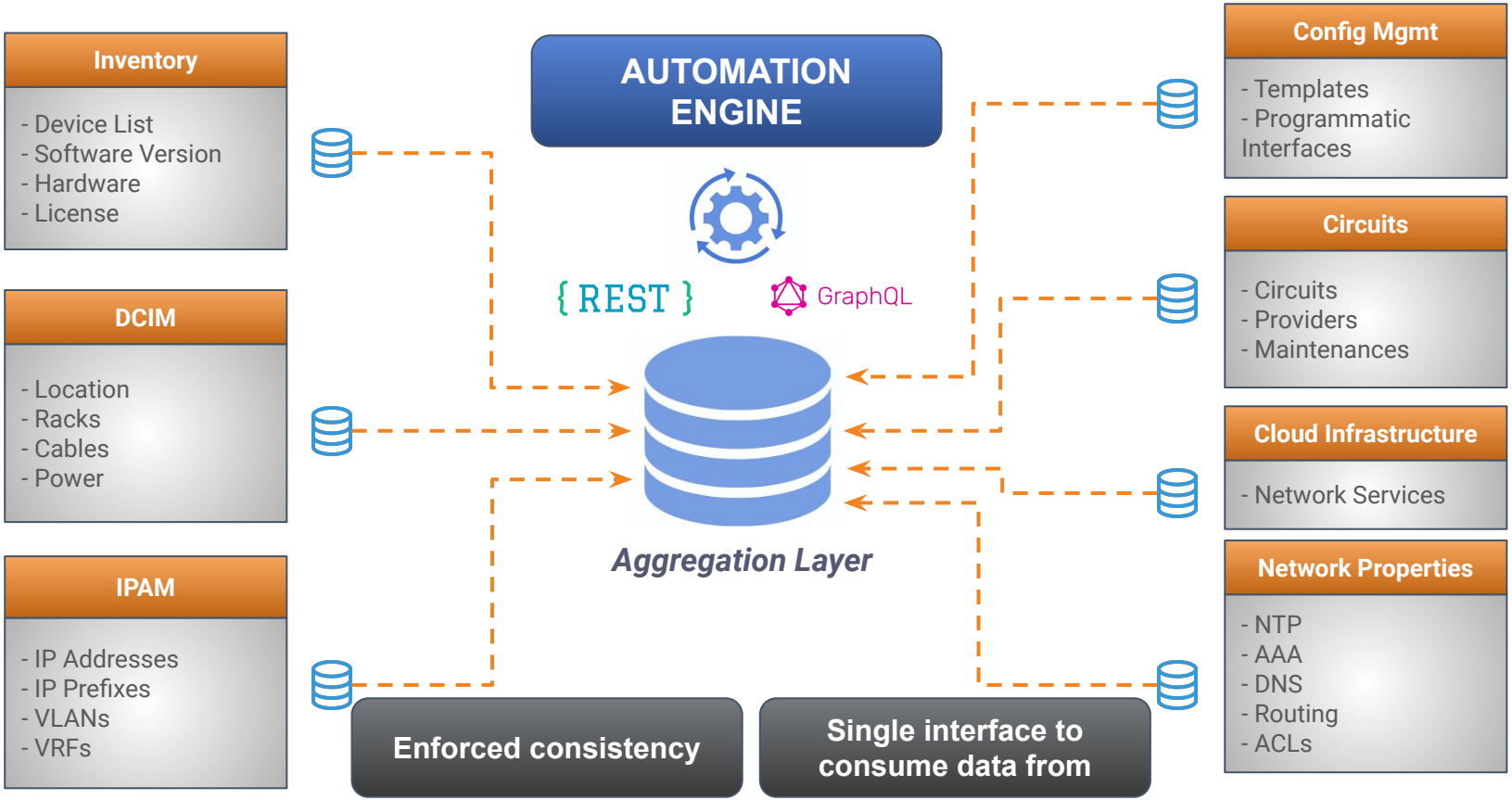
Cloud Infrastructure

- Network Services

Network Properties

- NTP
- AAA
- DNS
- Routing
- ACLs

>>> Why a Single Source of Truth?



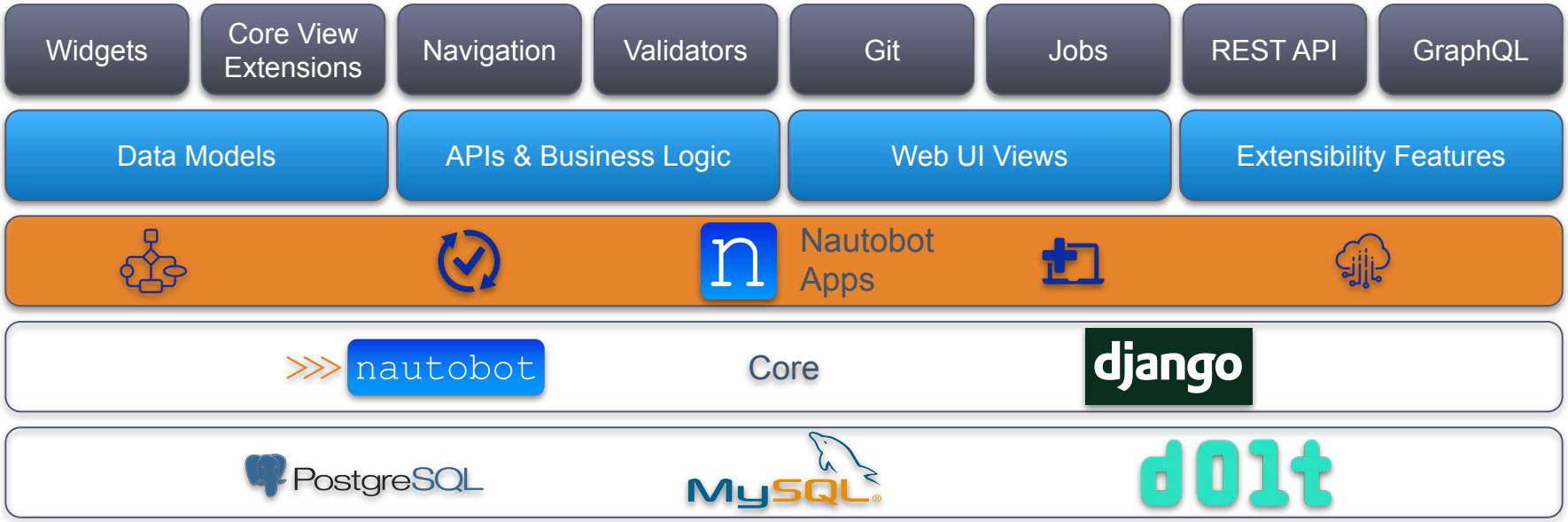

Nautobot SSoT

Nautobot

Source of Truth and Network Automation Platform

https://demo.nautobot.com

- Open source community project launched in 2021
- Sponsored by Network to Code
- 25+ open source Nautobot Apps



>>> A quick demo!

DB API Marketplace

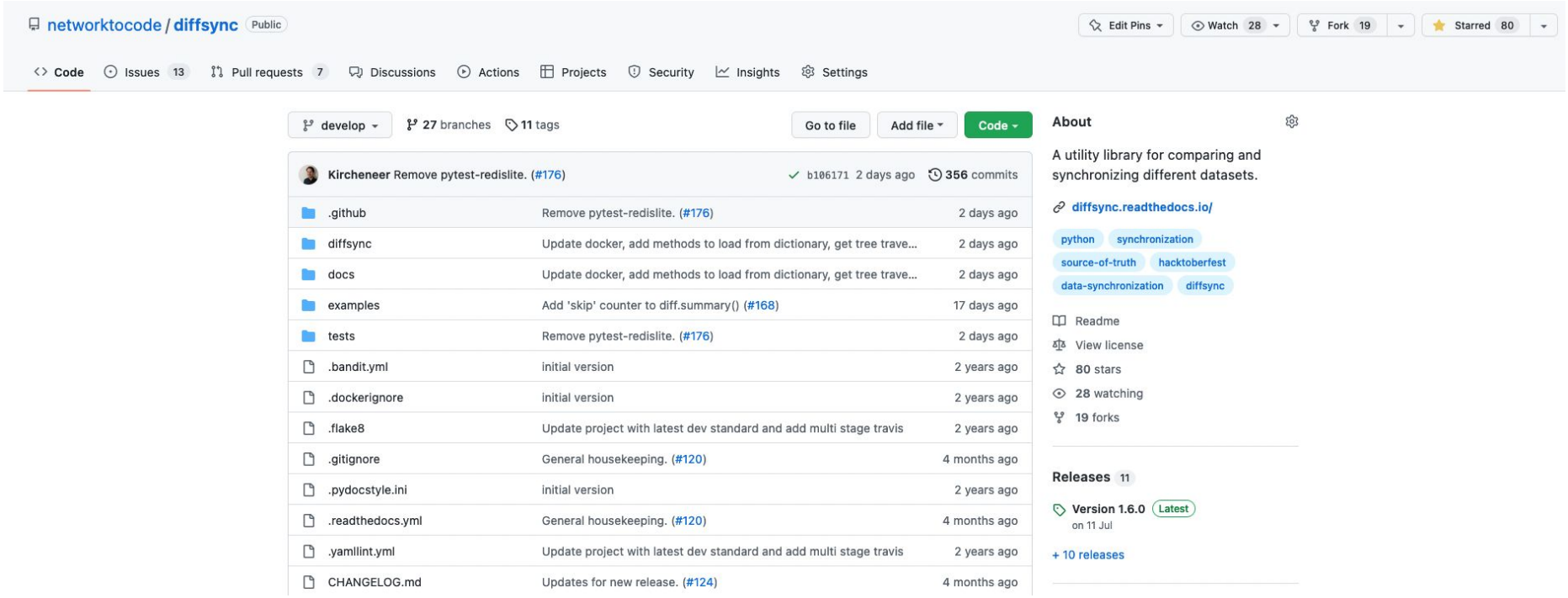
Nautobot Single Source of Truth

>>> nautobot

>>> A quick demo!


DiffSync library
$ pip install diffsync

>>> DiffSync



networktocode / diffsync Public

Edit Pins | Watch 28 | Fork 19 | Starred 80

Code | Issues 13 | Pull requests 7 | Discussions | Actions | Projects | Security | Insights | Settings

develop | 27 branches | 11 tags | Go to file | Add file | Code

Kircheneer Remove pytest-redislite. (#176) — b106171 2 days ago — 356 commits

.github	Remove pytest-redislite. (#176)	2 days ago
diffsync	Update docker, add methods to load from dictionary, get tree trave...	2 days ago
docs	Update docker, add methods to load from dictionary, get tree trave...	2 days ago
examples	Add 'skip' counter to diff.summary() (#168)	17 days ago
tests	Remove pytest-redislite. (#176)	2 days ago
.bandit.yml	initial version	2 years ago
.dockerignore	initial version	2 years ago
.flake8	Update project with latest dev standard and add multi stage travis	2 years ago
.gitignore	General housekeeping. (#120)	4 months ago
.pydocstyle.ini	initial version	2 years ago
.readthedocs.yml	General housekeeping. (#120)	4 months ago
.yamllint.yml	Update project with latest dev standard and add multi stage travis	2 years ago
CHANGELOG.md	Updates for new release. (#124)	4 months ago

About

A utility library for comparing and synchronizing different datasets.

diffsync.readthedocs.io/

python | synchronization | source-of-truth | hacktoberfest | data-synchronization | diffsync

Readme | View license | 80 stars | 28 watching | 19 forks

Releases 11

Version 1.6.0 Latest on 11 Jul

+ 10 releases

>>> How can Diffsync help you?

SUBNET
IPAM A

cidr	192.0.2.0/24
family	4
vrf	vrf-blue
vlan	VLAN123
customer_id	abc

PREFIX
IPAM B

network	192.0.2.0
prefix_length	24
vrf	vrf-blue
vlan_id	123
tenant	abc

>>> How can Diffsync help you?

SUBNET
IPAM A

cidr 192.0.2.0/24
family 4
vrf vrf-blue
vlan VLAN123
customer_id abc

How can we load the data?

What is the difference?

How could we compare vlan name and vlan id?

How can we synchronize the data?

PREFIX
IPAM B

network 192.0.2.0
prefix_length 24
vrf vrf-blue
vlan_id 123
tenant abc

>>> DiffSync, by example

PREFIX DIFFSYNC

prefix	192.0.2.0/24
vrf	vrf-blue
vlan_id	123
tenant	abc

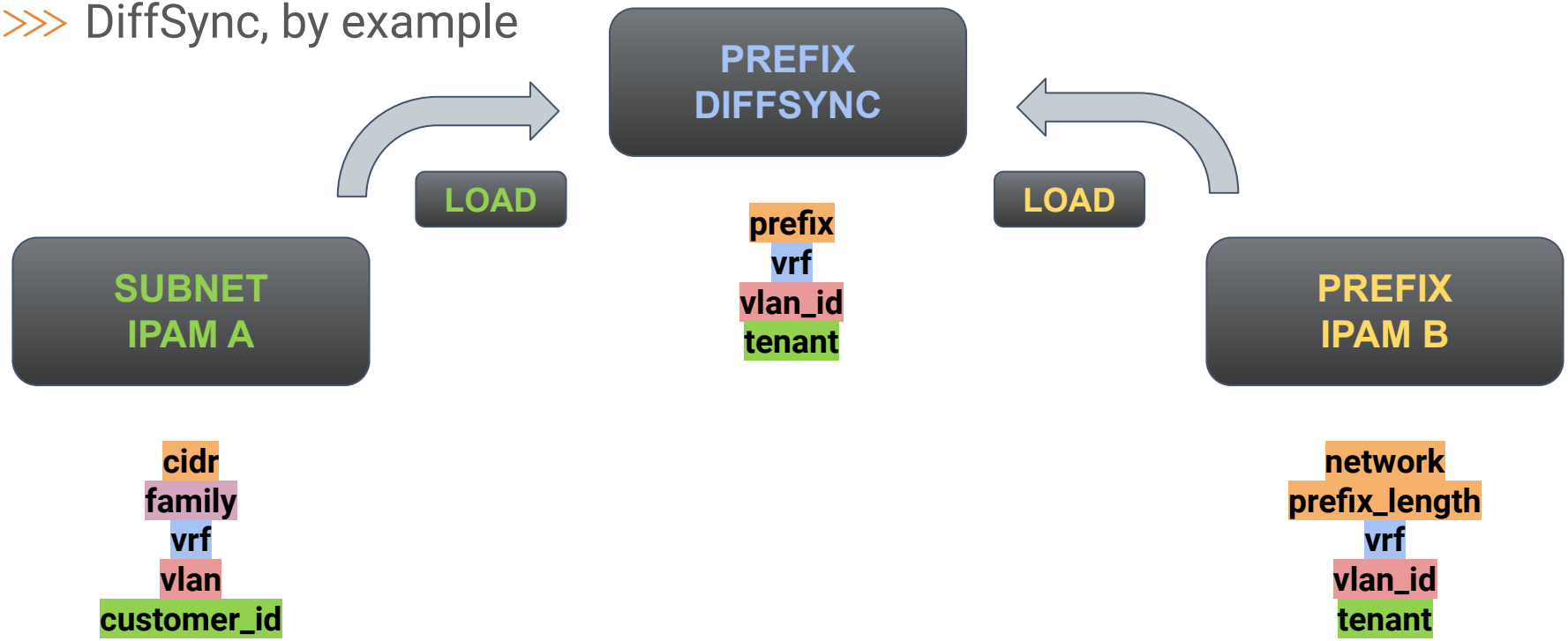
SUBNET IPAM A

cidr	192.0.2.0/24
family	4
vrf	vrf-blue
vlan	VLAN123
customer_id	abc

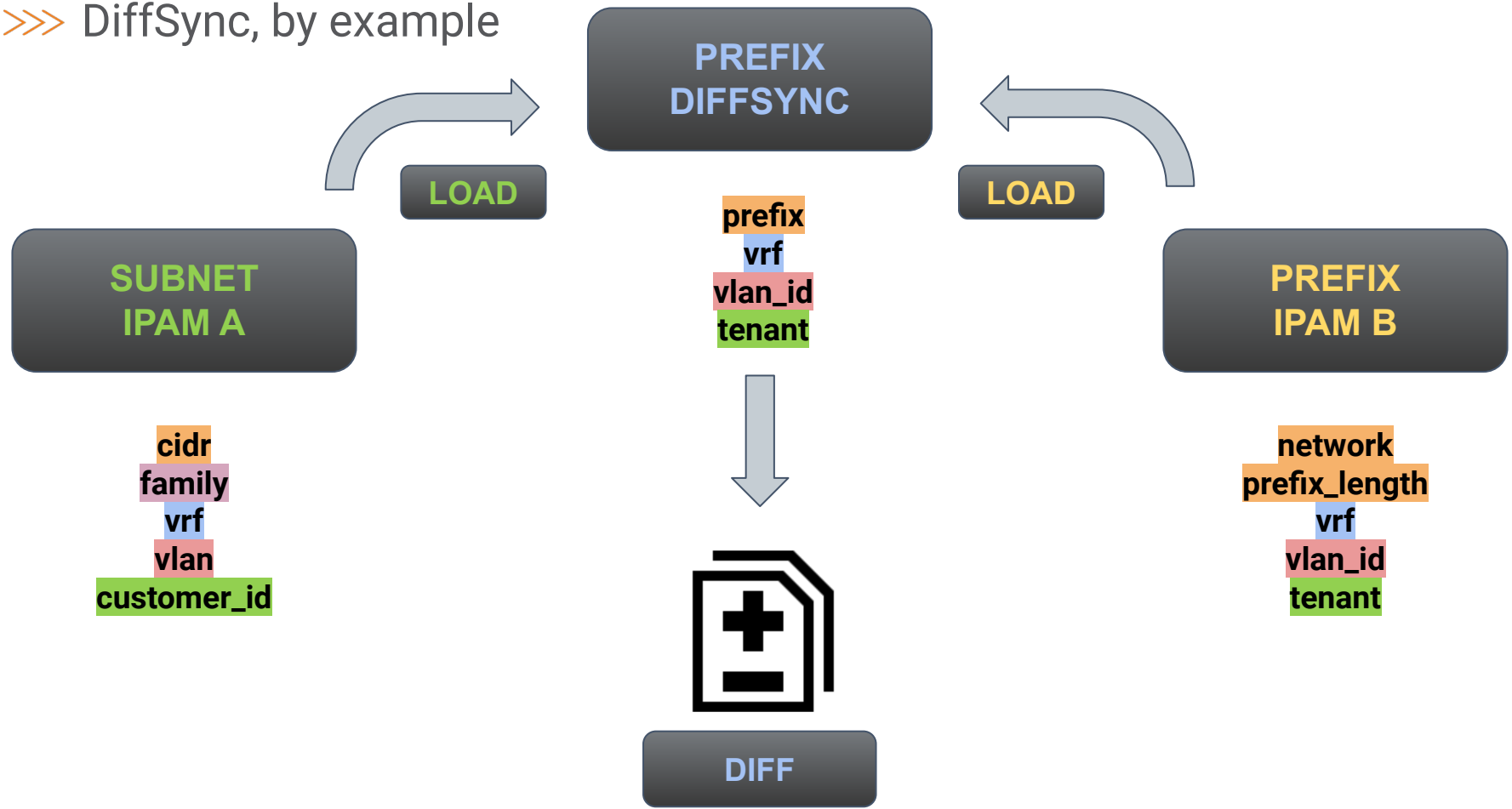
PREFIX IPAM B

network	192.0.2.0
prefix_length	24
vrf	vrf-blue
vlan_id	123
tenant	abc

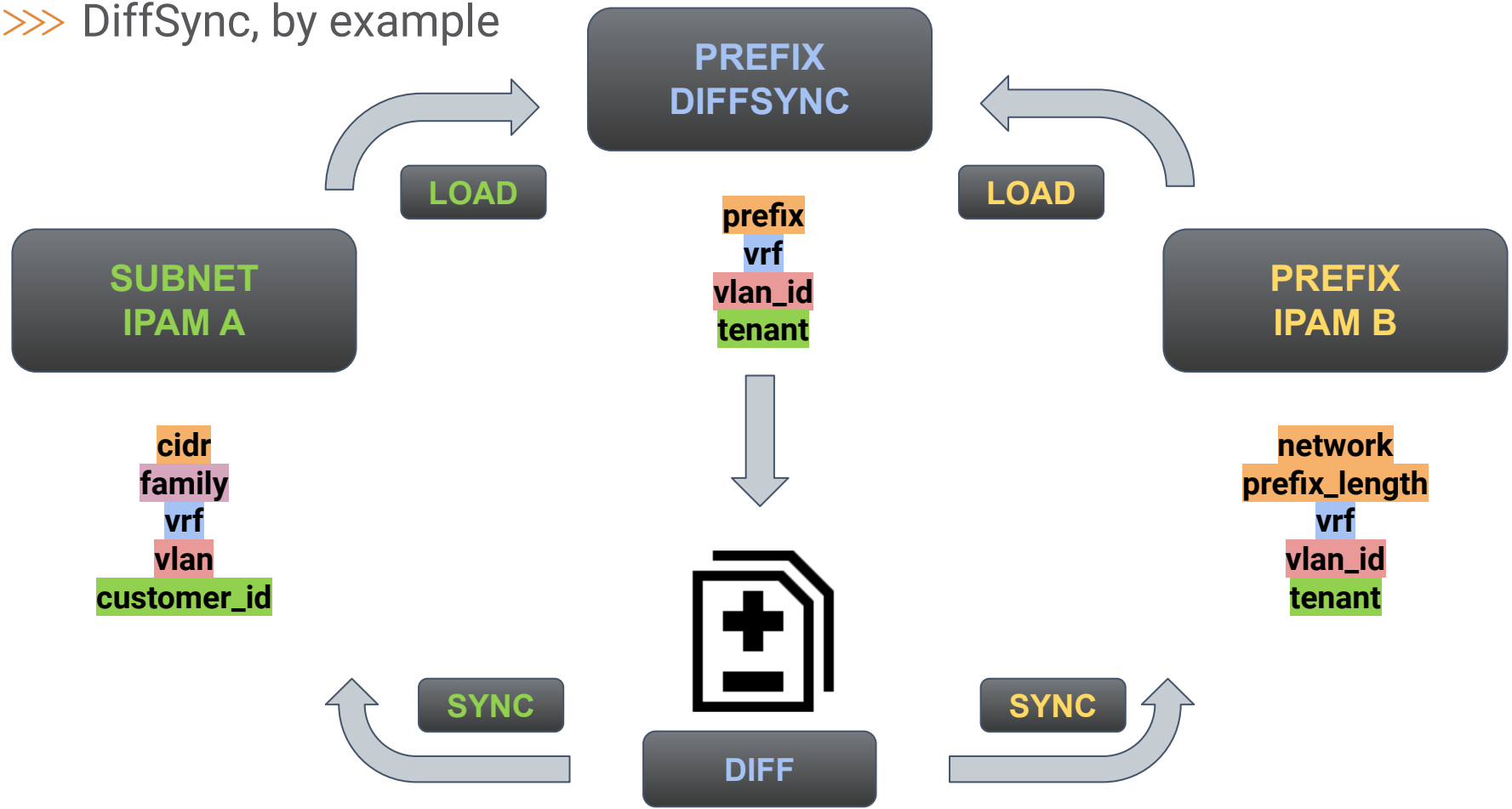
>>> DiffSync, by example



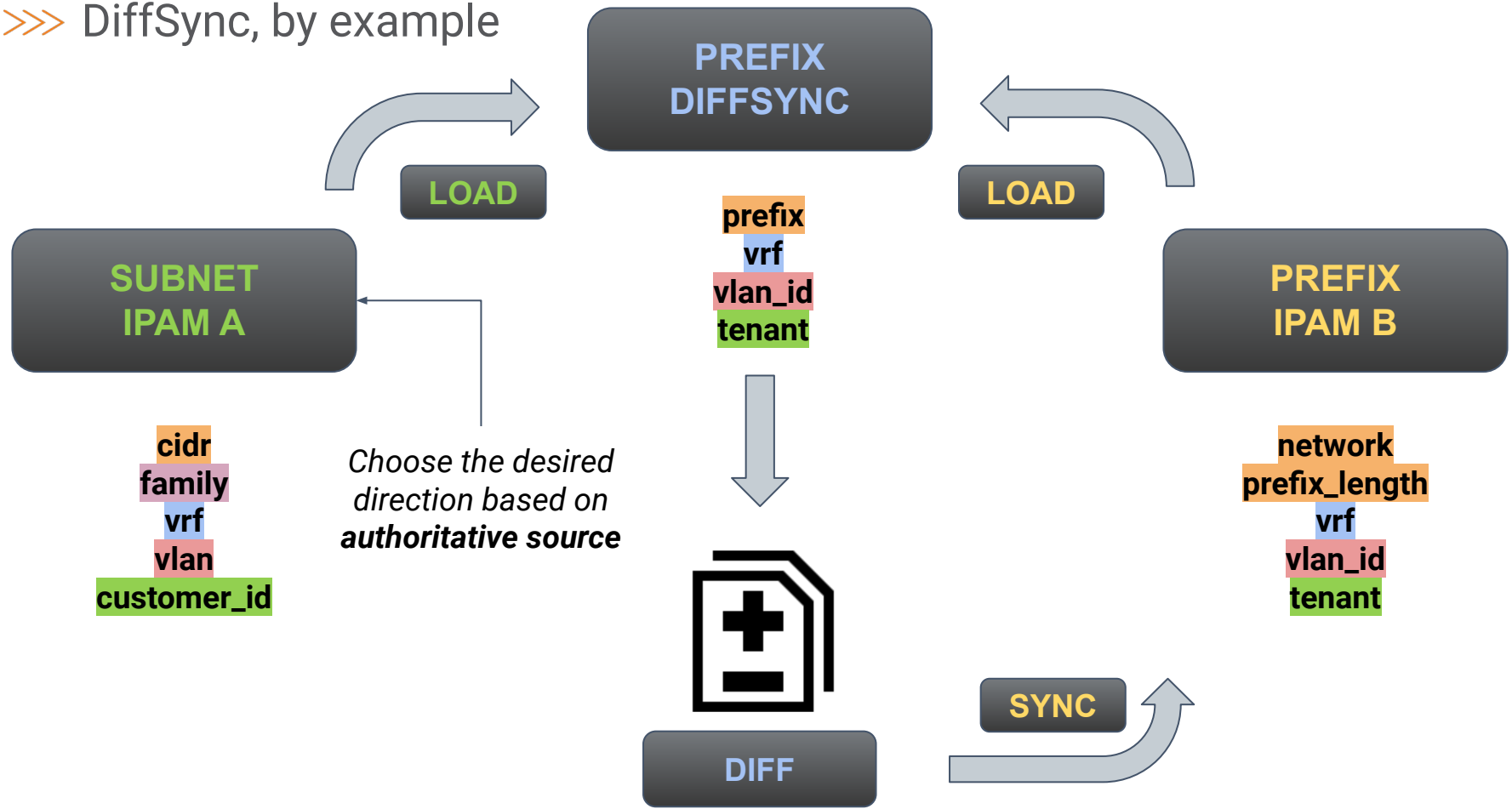
>>> DiffSync, by example



>>> DiffSync, by example



>>> DiffSync, by example



>>> A quick tour of the code!

DB API Marketplace

Nautobot Single Source of Truth

>>> nautobot

>>> The diffsync model

```
class BaseSite(DiffSyncModel):
  _modelname = "site"
  _identifiers = ("name", "region_name")
  _attributes = (
    "description", "shipping_address", "latitude", "longitude"," contact_name", "contact_phone"
  )
  _children = {}

  name: str
  region_name: str
  description: Optional[str]
  shipping_address: Optional[str]
  latitude: Optional[float]
  longitude: Optional[float]
  contact_name: Optional[str]
  contact_phone: Optional[str]
```

>>> The diffsync model

```
class NautobotSite(BaseSite):
  @classmethod
  def create(cls, diffsync, ids, attrs): pass
  def update(self, attrs): pass
  def delete(self): pass
```

>>> The diffsync model - create

```python
from nautobot.dcim.models import Site, Region
from nautobot.extras.models import Status

@classmethod
def create(cls, diffsync, ids, attrs):
  site = Site(
    name=ids["name"],
    region=Region.objects.get(name=ids["region_name"]),
    description=attrs["description"],
    shipping_address=attrs["shipping_address"],
    latitude=attrs["latitude"],
    longitude=attrs["longitude"],
    contact_name=attrs["contact_name"],
    contact_phone=attrs["contact_phone"],
    status=Status.objects.get(slug="active")
  )
  site.validated_save()
  return super().create(diffsync=diffsync, ids=ids, attrs=attrs)
```

>>> The diffsync model - update

```python
from nautobot.dcim.models import Site

def update(self, attrs):
  site = Site.objects.get(name=self.name)
  for field, value in attrs.items():
    setattr(site, field, value)
  site.validated_save()
  return super().update(attrs)
```

>>> The diffsync model - delete

```python
from nautobot.dcim.models import Site

def delete(self):
  site = Site.objects.get(name=self.name)
  super().delete()
  site.delete()
  return self
```

>>> The diffsync adapter

```
from nautobot.dcim.models import Site, Region

class NautobotAdapter(DiffSync):
  [...]
  def load(self):
    for region in Region.objects.all():
      diffsync_region = NautobotRegion(name=region.name)
      self.add(diffsync_region)
      for site in Site.objects.filter(region=region):
        diffsync_site = NautobotSite(
          name=site.name,
          region_name=diffsync_region.name,
          description=site.description,
          shipping_address=site.shipping_address,
          latitude=site.latitude,
          longitude=site.longitude,
          contact_name=site.contact_name,
          contact_phone=site.contact_phone
        )
        self.add(diffsync_site)
```

>>> The diffsync job / data source

```python
from nautobot.extras.jobs import BooleanVar, Job
from nautobot_ssot.jobs.base import DataSource, DataMapping

class DBStadaDataSource(DataSource, Job):
  [...]
  class Meta:
    name = "DB StaDa to Nautobot"
    data_source = "DB StaDa"
    data_target = "Nautobot"
    description = "Sync information from DB StaDa to Nautobot"

  def load_source_adapter(self):
    self.source_adapter = db_stada.DBStadaAdapter(job=self, sync=self.sync)
    self.source_adapter.load()

  def load_target_adapter(self):
    self.target_adapter = nautobot.NautobotAdapter(job=self, sync=self.sync)
    self.target_adapter.load()

jobs = [DBStadaDataSource]
```

>>> References

- Network to Code **Slack** channel
  - https://networktocode.slack.com/
- Network to Code **Blog** #ssot
  - https://blog.networktocode.com/post/advanced-ssot-app/
- Nautobot
  - https://github.com/nautobot/nautobot/
- Nautobot SSoT
  - https://github.com/nautobot/nautobot-plugin-ssot
- Nautobot DB StaDa SSOT
  - https://github.com/Kircheneer/nautobot-plugin-ssot-db-stada
- Diffsync
  - https://github.com/networktocode/diffsync


>>> network .toCode()
Thank you!
Questions?