

>>> network .toCode()

# How to handle your Circuit Maintenances proactively

RIPE82

Christian Adell Querol  
Network to Code

## >>> About me

- **Network Automation** Consultant at **Network to Code**
- Organizer of **NetBCN.cat** community

>>> network .toCode()  
*slack.networktocode.com*



@christianadell



@chadell0



@chadell



# >>> Agenda

The problem

Parser

SoT

Demo

Use Cases

Wrap-up

>>> The problem

## >>> Handling Circuit Maintenances

- Networks are build on top of a **lot of circuits**
- Every circuit will require **periodic maintenances**
- A circuit maintenance not handled properly will:
  - **Impact your operations** -> your **business**
  - **Add noise** to your alerting system
- Every provider uses his own **custom format**

>>>

~ 3

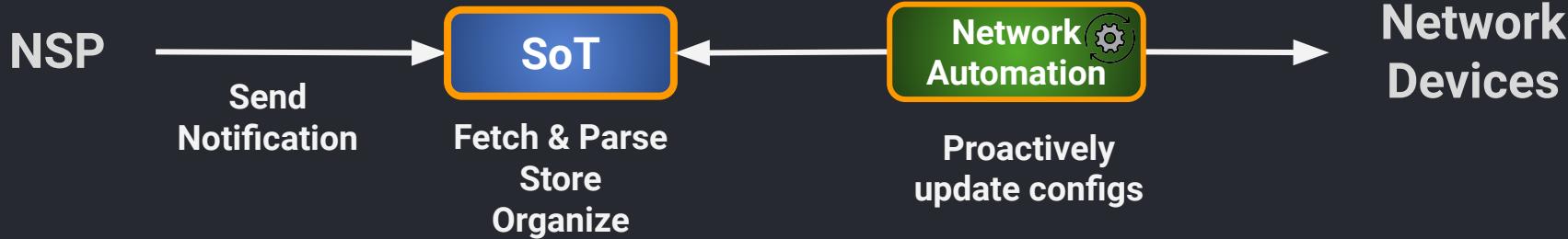
maintenances  
per circuit  
per year

# >>> Multiples problems to Solve

## Current Situation



## Potential Solution



# >>> Previous Work

## Draft NANOG BCOP

Shepherd: Erik Klavon ([erik.klavon@gmail.com](mailto:erik.klavon@gmail.com))

Subject Matter Expert(s) (SME): Francisco Hidalgo, Tylar Keese, Tj Trask, Sean Stuart, Randy Neals, Peter Hoose, Dave McGaugh, Paul Schultz, Joel Wride

Status: Draft 0.1

BCOP Subject: A machine parseable standard for formatting maintenance notifications

### 1. BCOP Summary (Appeal)

The format of maintenance notification varies from sender to sender, making it difficult to automate processing of these messages. This BCOP defines conventions for machine parseable formatting of information within common forms of maintenance notifications.

Versions: [00](#)

Calendaring Extensions  
Internet-Draft  
Intended status: Experimental  
Expires: January 4, 2020

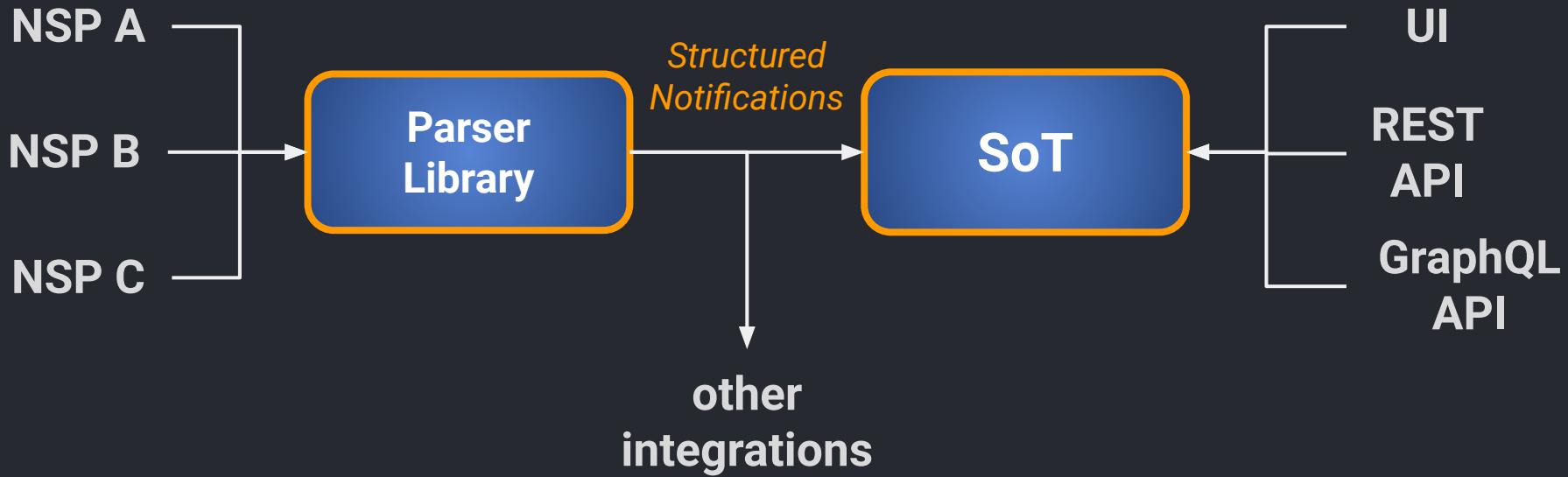
R. Gunter, Ed.  
Twitch  
July 3, 2019

Maintenance Notification Improvements Using iCalendar  
[draft-gunter-calexit-maintenance-notifications-00](#)

#### Abstract

This document proposes a maintenance notification convention that requires the use of an iCalendar file augmented with standardized and machine parseable metadata. The metadata is constructed by using the x-name property in the iCalendar file in compliance with [[RFC 5545](#)] [[RFC5545](#)]. This specification substantially reduces automation efforts, and still provides easy calendaring for manual processing.

# >>> Proposed Architecture



# >>> Circuit Maintenance Parser

*pip install circuit-maintenance-parser*

# >>> Just parse it

iCal

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Maint Note//https://github.com/maint-notification//BEGIN:VEVENT
SUMMARY:Maint Note Example
DTSTART;VALUE=DATE-TIME:20151010T080000Z
DTEND;VALUE=DATE-TIME:20151010T100000Z
DTSTAMP;VALUE=DATE-TIME:20151010T001000Z
UID:42
SEQUENCE:1
X-MAINTNOTE-PROVIDER:example.com
X-MAINTNOTE-ACCOUNT:137.035999173
X-MAINTNOTE-MAINTENANCE-ID:WorkOrder-31415
X-MAINTNOTE-OBJECT-ID:acme-widgets-as-a-service
X-MAINTNOTE-IMPACT:NO-IMPACT
X-MAINTNOTE-STATUS:TENTATIVE
ORGANIZER;CN="Example NOC":mailto:noone@example.com
END:VEVENT
END:VCALENDAR
```

HTML

```
)>Reason for Maintenance:=C2=A0</b><span style=3D"color:rgb(0,0,0)">Zayo w=ill implement planned maintenance to troubleshoot and restore degraded span= =C2=A0</span><br style=3D"color:rgb(0,0,0)"><br style=3D"color:rgb(0,0,0)">= <b style=3D"color:rgb(0,0,0)">Expected Impact:=C2=A0</b><span style=3D"color:rgb(0,0,0)">Service Affecting Activity: Any Maintenance Activity directly= impacting the service(s) of customers. Service(s) are expected to go down = as a result of these activities.=C2=A0<span><br style=3D"color:rgb(0,0,0)">= <br style=3D"color:rgb(0,0,0)"><b><br style=3D"color:rgb(0,0,0)"><table border=3D"3Dquot;"& quot;"><tr><td><tr><td><th>Circuit Id</th><th>A= fected:=C2=A0</td><td><tr><td><th>Expected Impact</th><th>A= fected:=C2=A0</td><td><tr><td><th>Location CLI</th><th>Legacy Circuit Id</th></tr><tr><td>OGYX/000000/Z YO /</td><td>Hard Down - up to 2 hours</td><td>DLLST= X37</td><td>SN5CAJN=</td><td></td></tr></tbody></table><br style=3D"color:rgb(0,0,0)">Please contact the Zayo Maintenance Team with any questions regarding this = maintenance event. Please reference the Maintenance Ticket number when call= ing.=C2=A0</span><br style=3D"color:rgb(0,0,0)"><br style=3D"color:rgb(0,0,0)">= <b style=3D"color:rgb(0,0,0)">Maintenance Team Contacts:=C2=A0</b><br s= tyle=3D"color:rgb(0,0,0)"><br style=3D"color:rgb(0,0,0)"><div style=3D"color:rgb(0,0,0)"><div style=3D"margin:0in 0in 0.0001pt;background-color:white">= <b><span style=3D"font-family:&quot;Trebuchet MS&quot;,sans-serif;color:rg= b(0,89,89)">Zayo</span></b><b><span style=3D"font-family:&quot;Trebuchet MS&quot;,sans-serif;color:rgb(255,128,0)">oc</span></b><b><span style=3D"font= family:&quot;Trebuchet MS&quot;,sans-serif;color:rgb(0,89,89)">C2=A0Globa= l Change Management Team/<i>=C3=89quipe</i><i>=C2=A0de=C2=A0gestion=C2=A0du= l
```

Circuit  
Maintenance  
Parser

```
{
  "account": "137.035999173",
  "end": 1444644000,
  "maintenance_id": "WorkOrder-31415",
  "circuits": [
    {
      "impact": "NO-IMPACT",
      "circuit_id": "acme-widgets-as-a-service"
    }
  ],
  "organizer": "mailto:noone@example.com",
  "provider": "example.com",
  "sequence": 2,
  "stamp": 1444608600,
  "start": 1444636800,
  "status": "CONFIRMED",
  "summary": "Maint Note Example",
  "uid": "42"
}
```

# >>> How to use it

```
from circuit_maintenance_parser import init_parser

raw_text = """BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Maint Note//https://github.com/maint-notification// 
BEGIN:VEVENT
SUMMARY:Maint Note Example
DTSTART;VALUE=DATE-TIME:20151010T080000Z
DTEND;VALUE=DATE-TIME:20151010T100000Z
DTSTAMP;VALUE=DATE-TIME:20151010T001000Z
UID:42
SEQUENCE:1
X-MAINTNOTE-PROVIDER@example.com
X-MAINTNOTE-ACCOUNT:137.035999173
X-MAINTNOTE-MAINTENANCE-ID:WorkOrder-31415
X-MAINTNOTE-IMPACT:OUTAGE
X-MAINTNOTE-OBJECT-ID;X-MAINTNOTE-OBJECT-IMPACT=NO-IMPACT:acme-widgets-as-a-service
X-MAINTNOTE-OBJECT-ID;X-MAINTNOTE-OBJECT-IMPACT=OUTAGE:acme-widgets-as-a-service-2
X-MAINTNOTE-STATUS:TENTATIVE
ORGANIZER;CN="Example NOC":mailto:noone@example.com
END:VEVENT
END:VCALENDAR
"""

data = {
    "subject": "this is a circuit maintenance from some NSP",
    "sender": "support@networkserviceprovider.com",
    "source": "gmail",
    "raw": raw_text,
}
```

```
parser = init_parser(**data)

parsed_notifications = parser.process()

print(parsed_notifications[0].to_json())
{
    "account": "137.035999173",
    "circuits": [
        {
            "circuit_id": "acme-widgets-as-a-service",
            "impact": "NO-IMPACT"
        },
        {
            "circuit_id": "acme-widgets-as-a-service-2",
            "impact": "OUTAGE"
        }
    ],
    "end": 1444471200,
    "maintenance_id": "WorkOrder-31415",
    "organizer": "mailto:noone@example.com",
    "provider": "example.com",
    "sequence": 1,
    "stamp": 1444435800,
    "start": 1444464000,
    "status": "TENTATIVE",
    "summary": "Maint Note Example",
    "uid": "42"
}
```



# >>> Circuit Maintenance Nautobot Plugin

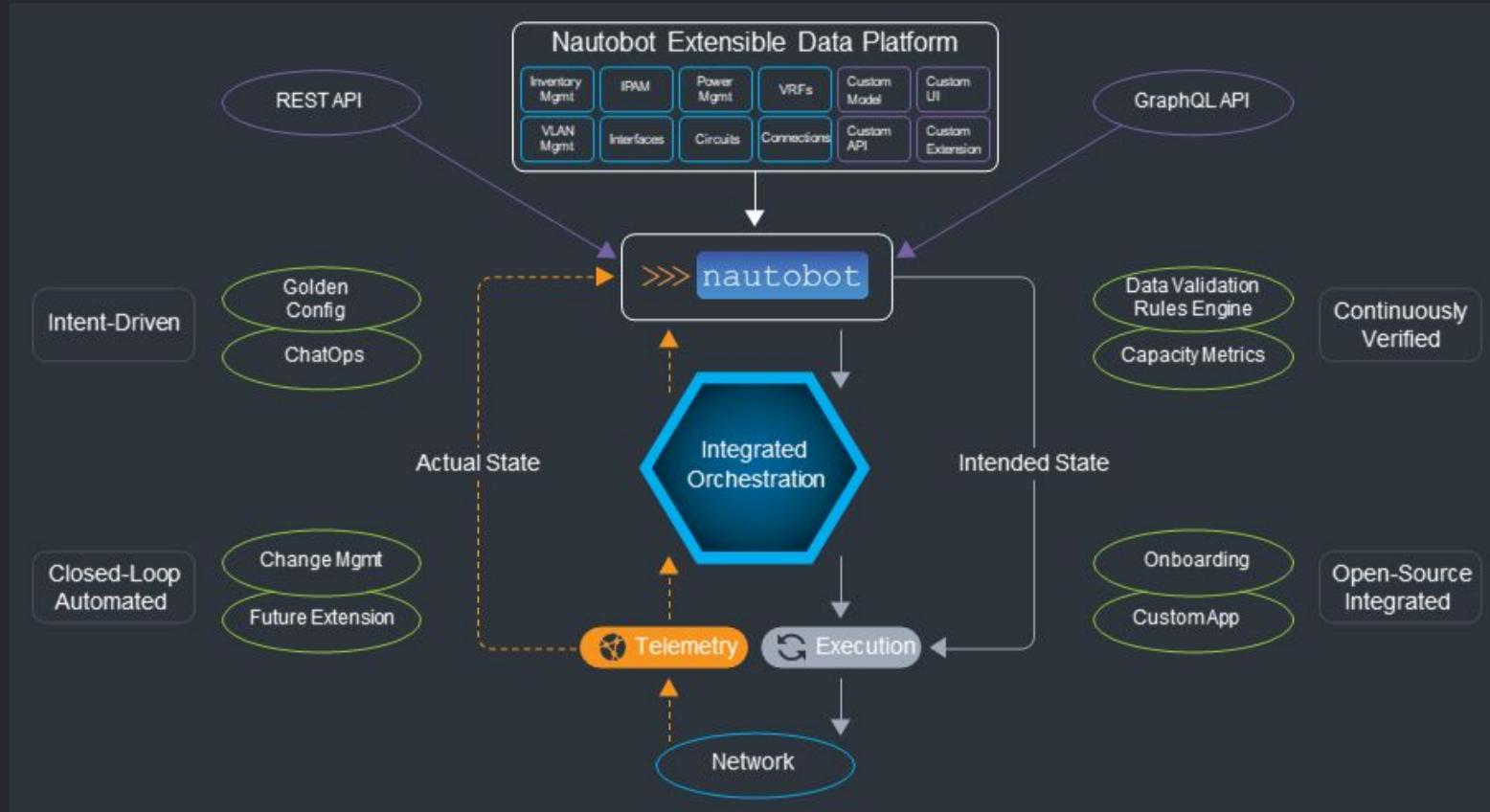
*pip install plugin-circuit-maintenance*

## Source of Truth and Network Automation Platform

- Open source community project created in 2021
  - Apache2 license
- Sponsored by Network to Code
- Forked from existing **NetBox** project
  - Python / Django



# >>> Network Automation with Nautobot



# >>> Key Features



## Data Validation

Codify business rules to ensure there is nothing but high-quality data in Nautobot.



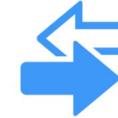
## User-Defined Relationships

Create custom relationships between existing data models that replicate your network design.



## Custom Fields

Augment existing data models through custom fields on any object including interfaces.



## Data Source (Git) Integration

Seamlessly integrate YAML-based structured data files directly into Nautobot.



## Jobs

Using Python scripts to dynamically create self-service forms and reports that are easily executable in the UI.



## GraphQL

Easily fetch the exact data you desire across data models with a single API call.



## Webhooks

Have Nautobot make an outbound HTTP API call based on create, update, and delete operations.



## Plugin System

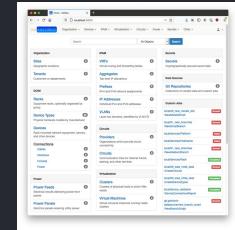
Add custom extensions and apps catering to your specific SoT and network automation requirements.

# >>> Nautobot Use Cases

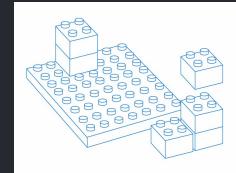
## 1 Flexible Source of Truth for Networking



- Devices
  - IP Addresses
  - VLANs
  - ASN
  - ...
  - Custom
- User-Defined Relationships
  - Custom Fields
  - Data Validation



## 2 Extensible Data Platform for Network Automation



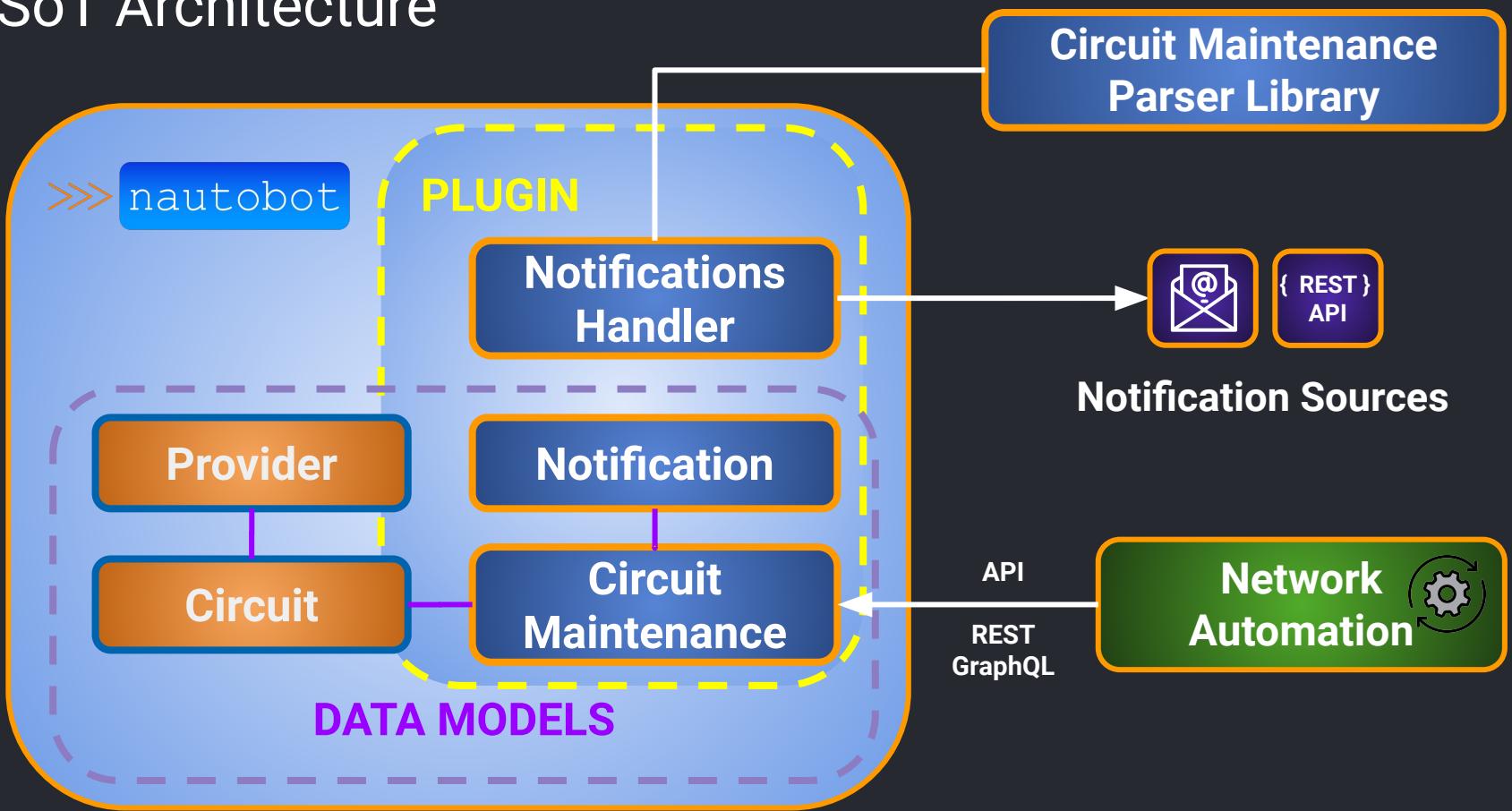
Extensible  
Plugin  
System

## 3 Platform for Network Automation Apps



- Use Open Source Apps
- Build Custom Apps
- Save 70% development time using the platform

# >>> SoT Architecture

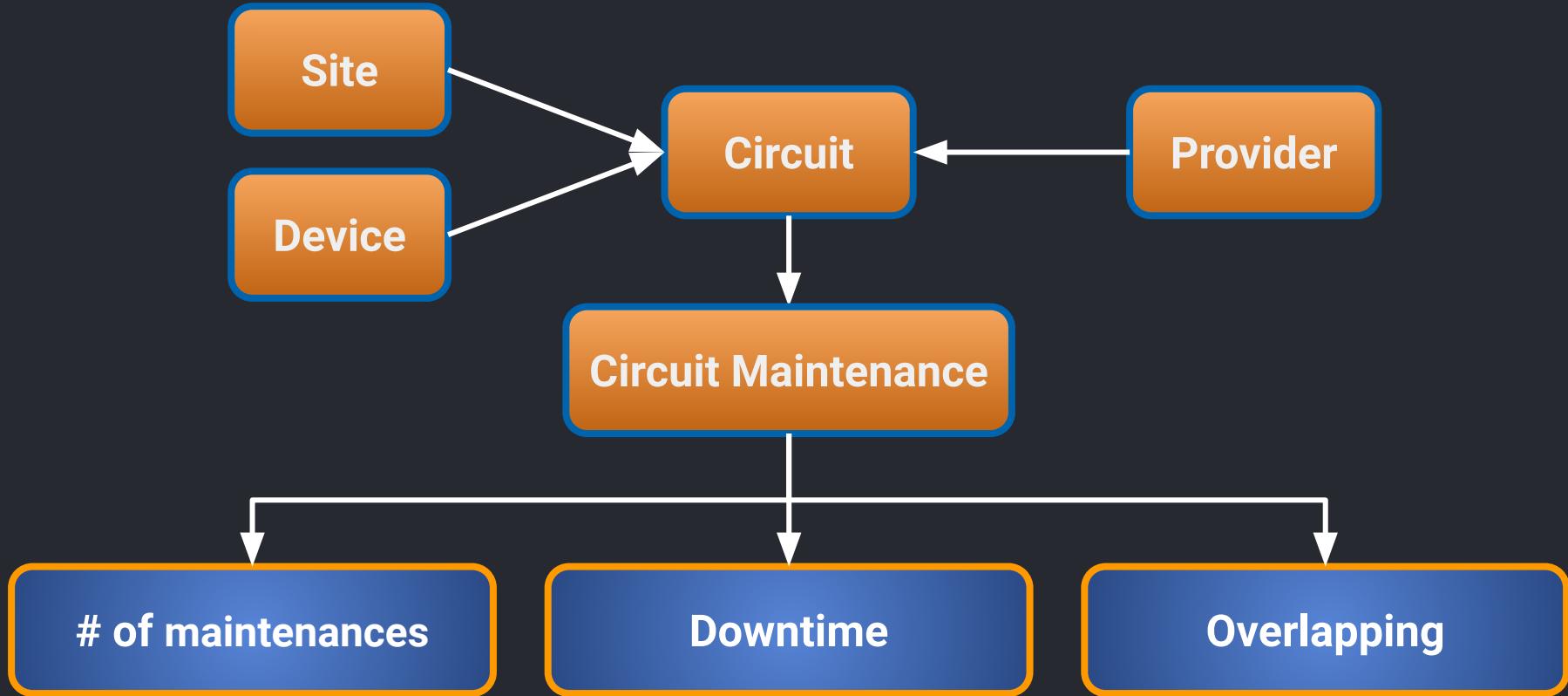


>>> Demo time

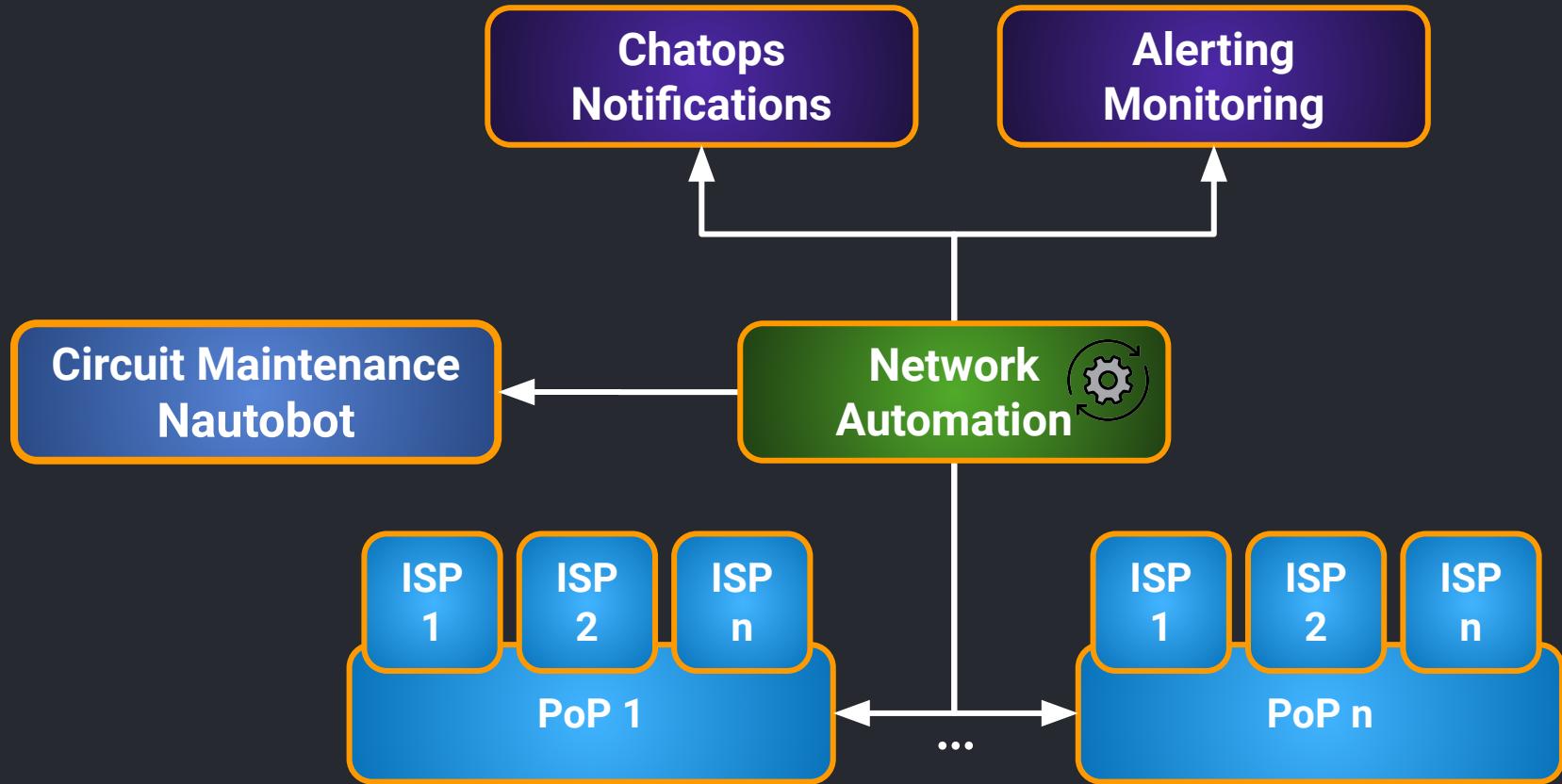
The background of the slide features a nighttime cityscape with a high density of skyscrapers. The buildings are mostly dark blue and black, with their windows glowing with white and yellow light, creating a sense of depth and urban density. In the foreground, there is a large, semi-transparent blue rectangular area that serves as a backdrop for the title text.

# >>> Use Cases

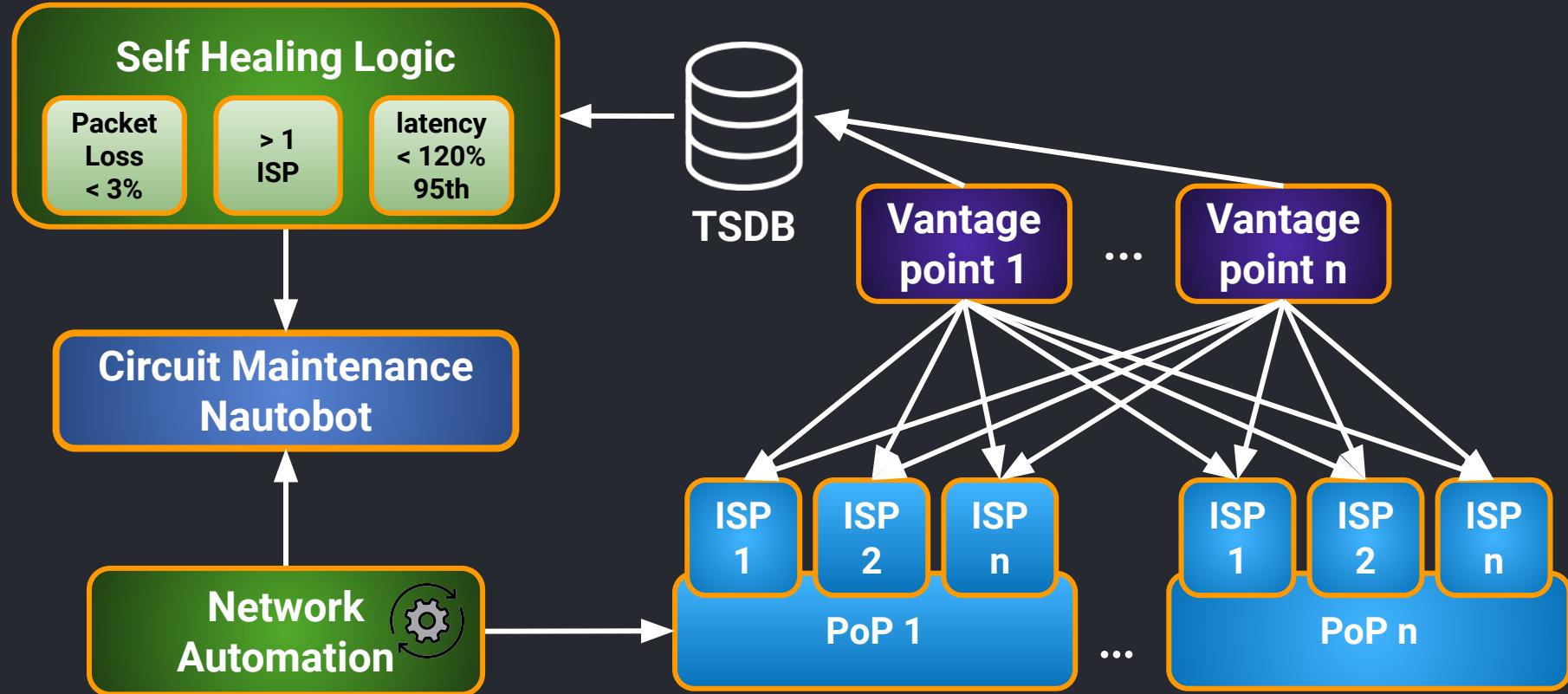
# >>> Reporting and Alerting



# >>> Automated Circuit Operations



# >>> Self-healing Networks



>>> Wrap-up

## >>> What's next?

1. Try it :)
2. More supported **parsers**
3. Add **reporting view** and **alerting integrations**
4. **Scheduler** built-in
5. Extend **source integrations**
6. ... and soon a **stable release!**

# >>> References

- IETF Draft, Maintenance Notification Improvements Using iCalendar
  - <https://tools.ietf.org/html/draft-gunter-calext-maintenance-notifications-00>
- Draft NANOG BCOP
  - <https://github.com/jda/maintnote-std/blob/master/standard.md>
- Nautobot
  - <https://github.com/nautobot>
- Nautobot Circuit Maintenance Plugin
  - <https://github.com/nautobot/nautobot-plugin-circuit-maintenance>
- Circuit Maintenance Parser
  - <https://github.com/networktocode/circuit-maintenance-parser>

>>> network .toCode()

# Thank You



@christianadell



@chadell0



@chadell