



JUNIPER[®]
NETWORKS

JUNIPER DAY

16 октября 2018 | Москва

AppFormix HealthBot РЕШЕНИЕ СЕТЕВЫХ ПРОБЛЕМ

Автоматизация диагностики:
задачи и решения

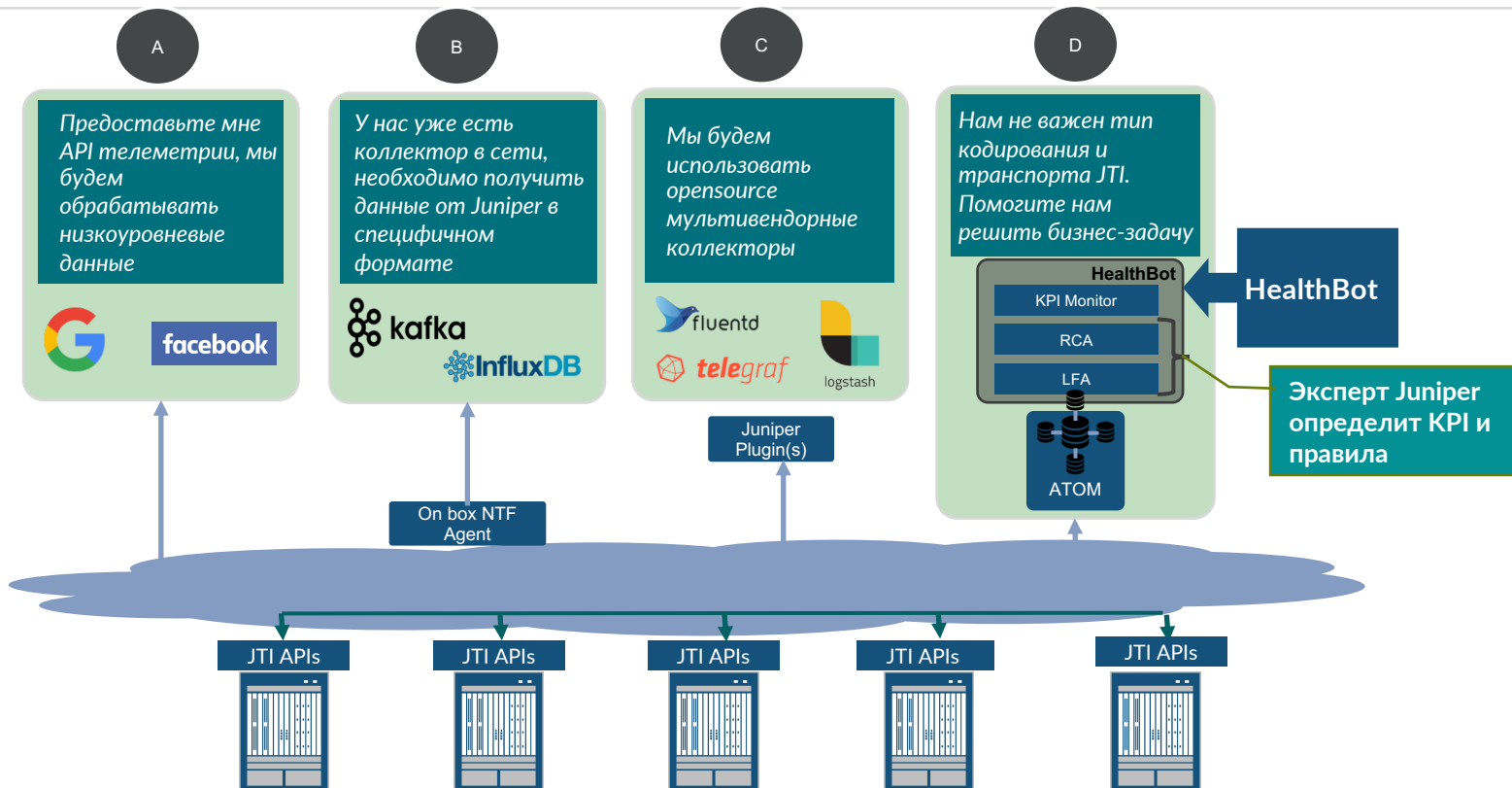
ДМИТРИЙ КАРЯКИН
Старший системный инженер
dkaryakin@juniper.net

LEGAL STATEMENT

This statement of direction sets forth Juniper Networks' current intention and is subject to change at any time without notice.

No purchases are contingent upon Juniper Networks delivering any feature or functionality depicted in this presentation.

Классификация потребности в телеметрии и аналитики



Некоторые задачи по диагностике

1 Детектирование Traffic black hole
Детектирование аномальных потерь трафика на PFE или фабрике

2 Мониторинг состояния BNG
Вычисление нескольких ключевых показателей KPI для оценки общего состояния системы BNG

3 Детектирование microburst трафика
Идентификация существования всплесков трафика, которые могут привести к его потерям

4 Подвисание PFE (wedge conditions)
Определение наличия ошибок на PFE, которые приводят или уже привели к зависанию PFE

...

Что в итоге?

- Широкий набор возможных задач по диагностике
- Многие задачи требуют сильной экспертизы по специфичному предмету знаний (BNG, ..)
- Зависят от контекста и конкретной реализации
- Нужна автоматизация диагностики
- Требуется возможность открытой программируемости

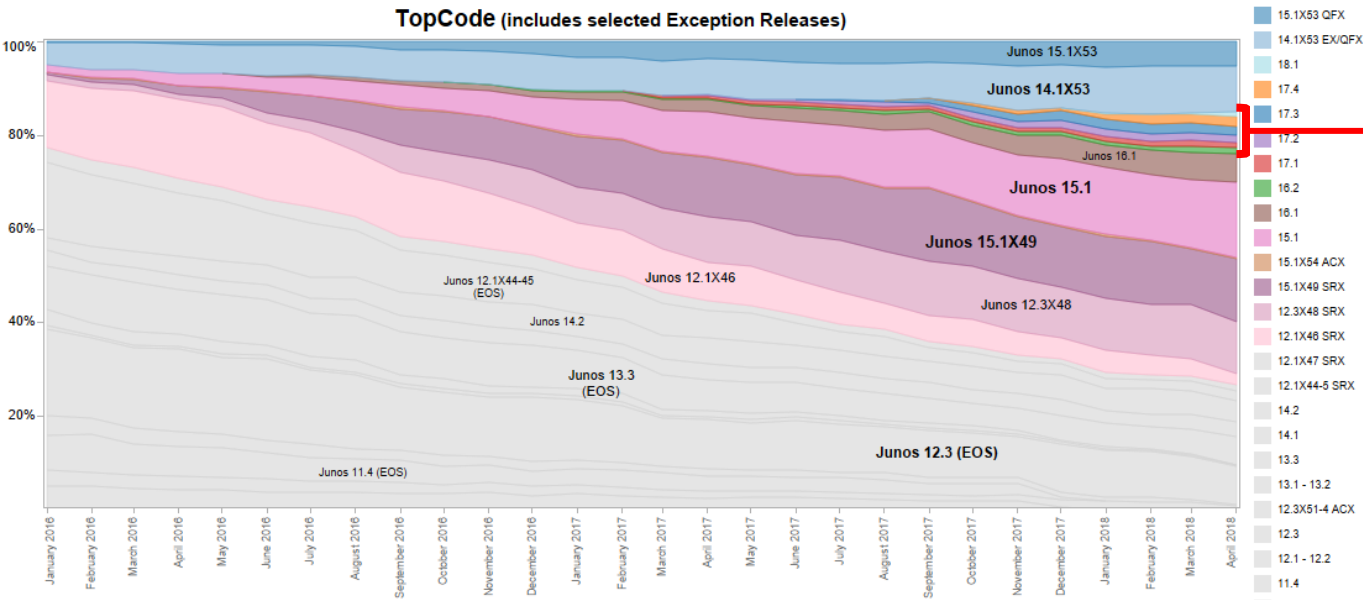
5 маршрутизации
выявление аномалий

6 Внедрение правил планирования ёмкости
Применение правил планирования расширения ёмкости и детектирование аномалий

7 Мониторинг состояния сервисов
Сквозная оценка состояния сервисов (pseudowires, EVPN, etc.)

...

Реальность...



Junos релизы с поддержкой телеметрии

Платформа	Поддержка телеметрии
MX	
PTX	
QFX	
ACX	X
SRX	X
EX	X

~35% заказчиков находятся на EOE или EOS релизах

Поддержка телеметрии



Что в итоге?

- GRPC/JTI телеметрия – это будущее. Давайте ее продвигать :)
- Для решения задач эксплуатации будем использовать все доступные механизмы
- Не нужно ничего ждать
- Много информации доступно только в SNMP/Netconf/CLI.
- Нет особой необходимости получать все данных каждые 2 секунды

Три ключевые задачи эксплуатации

1

Мониторинг

Какие данные
нужны?

2

Понимание

Как
интерпретировать и
анализировать эти
данные?

3

Действие

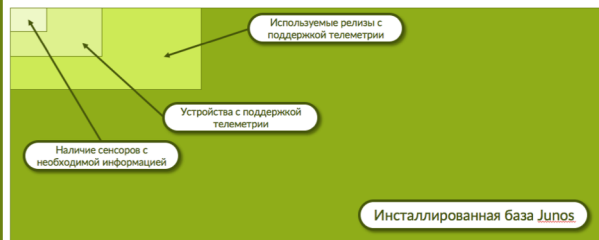
Какие действия
должны быть по
результатам
анализа?

Картина в целом...

Различные задачи диагностики и поиска неисправностей

1	Детектирование Traffic black hole	Детектирование аномальных потерь трафика на PFE или фабрике
2	Мониторинг состояния BNG	Вычисление нескольких ключевых показателей KPI для оценки общего состояния системы BNG
3	Детектирование микровсплесков трафика	Идентификация осуществления всплесков трафика, которые могут привести к его потерям
4	Подвисание PFE (wedge conditions)	Определение наличия ошибок на PFE, которые приводят или уже привели к зависанию PFE
5	Диагностика протоколов маршрутизации	Мониторинг и оценка поведения протоколов маршрутизации и выявление первичных аномалий
6	Анализ состояния таблиц маршрутизации	Оценка состояния маршрутизации и состояния форвардинга, выявление аномалий
7	Внедрение правил планирования видимости	Применение правил планирования расширения видимости и детектирование аномалий
8	Мониторинг состояния сервисов	Связанная оценка состояния сервисов (pseudowires, EVPN, etc.)

Необходимость поддержки нескольких источников телеметрии и операционных состояний



Автоматизация эксплуатации сети и диагностики



Appformix Healthbot
Программируемый инструмент для эксплуатации и диагностики



AppFormix HealthBot

Основная суть инструмента HealthBot

1

Постановка задачи

- Контроль состояния сложных сервисов и функций, таких как BNG, PE, etc.

Без HealthBot

- Контроль и визуализация десятков и сотен индикаторов KPI.
- Сильная зависимость от контекста
- Сложно для понимания
- Предмет различной интерпретации в зависимости о каждого конкретного специалиста
- Отсутствие действий

С использованием HealthBot

- Контроль сотен KPI
- Абстрактная функция оценки состояния с агрегированным показателем «здоровья»
- Гибкое программирование дерева состояния с помощью Playbook
- Playbook обеспечивает унифицированный анализ состояния
- Простой для понимания и принятия дальнейших действий

Основная суть инструмента HealthBot

2

Постановка задачи

- Диагностика проблемы устройства или сети

Без HealthBot

- Инженеры последовательно запускают десятки операционных команд для выявления первопричины
- Различные инженеры могут следовать различным процессам
- Медленно, реактивно и противоречиво
- Ресурсоёмко

С использованием HealthBot

- Определение процесса диагностики в Playbook
- Процесс всегда выполняется последовательно и непротиворечиво
- В режиме реального времени, реактивно и проактивно
- Автоматизировано

Основная суть инструмента HealthBot

3

Постановка задачи

- Превратить логи в набор полезных элементов и действий

Без HealthBot

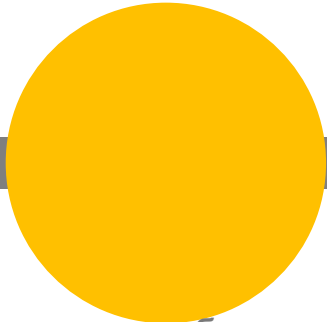
- Тысячи лог-сообщений
- Сложно расшифровать
- Высокая степень шума
- Инструменты логирования сфокусированы только логах
- Не очевидные дальнейшие действия

С использованием HealthBot

- Программируемый набор правил для обработки и визуализации логов
- Уменьшение шума
- Задействование кросс-корреляции логов и данных телеметрии
- Превращение полученных логов в набор действий

Мониторинг состояния сети и системы Junos

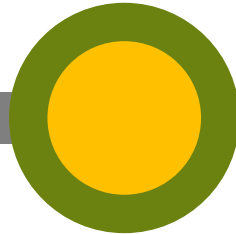
Другие коммерческие продукты для мониторинга сети



- Отсутствие гибкости
- Очень дорого
- Привязка к производителю (vendor lock in)
- Производитель определяет способы применения

«Из коробки»

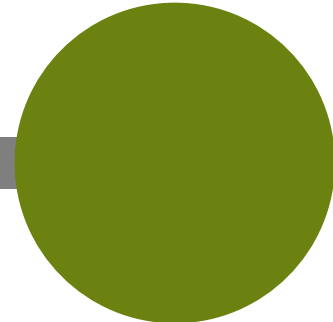
AppFormix Health Bot Программируемый инструмент сетевой эксплуатации



- Открытые и настраиваемые Playbooks
- Использование компонентов сообщества opensource
- Нет зависимости от производителя
- Поддерживаемое ядро и движок продукта
- Легко расширяемый
- Низкая стоимость владения (ТСО)

Адаптированный

Специально разработанное программное обеспечение для мониторинга



- Гибко
- Очень дорого
- Сложно для развития и поддержки

Ключевые атрибуты HealthBot

**Прежде всего –
программируемость:**
на основе Playbook

Автоматизация эксплуатации сети:
анализ первопричин и анализ лог-файлов;
детектирование аномалий на основе машинного
обучения

Несколько источников:
телеметрия, netconf, cli, события

Сообщество:
источником
playbook
является
широкая
аудитория

Масштабируемость:
архитектура на основе
микросервисов

Автоматизация:
уведомления,
интеграция (kafka, slack,
webhooks)

Playbooks: ключевые компоненты

1

Сбор

Программируемые
Playbook для
мониторинга, какие
данные являются
интересными (Sensors,
NETCONF, Events)

2

Анализ

Автоматизированные
программируемые
рабочие процессы
диагностики

3

Действие

Программируемая
автоматизация с
замкнутым контуром

Примеры использования – PLAYBOOKS

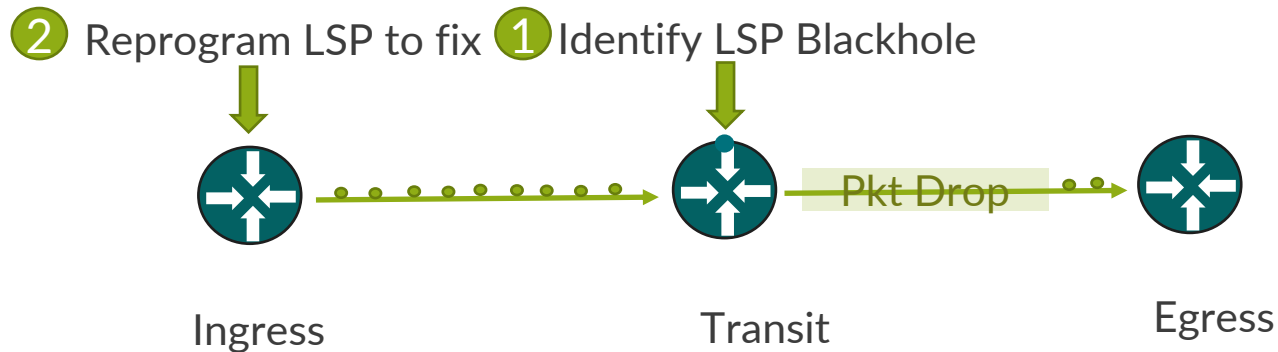
Protocols Health – OSPF, ISIS, BGP, MPLS, RSVP, etc.	Blackhole detection	L2 Circuit health
Microburst	System KPIs	BBE subscriber health
Interface threshold & health	Wedge detection	L3 VPN health

LSP BLACKHOLE

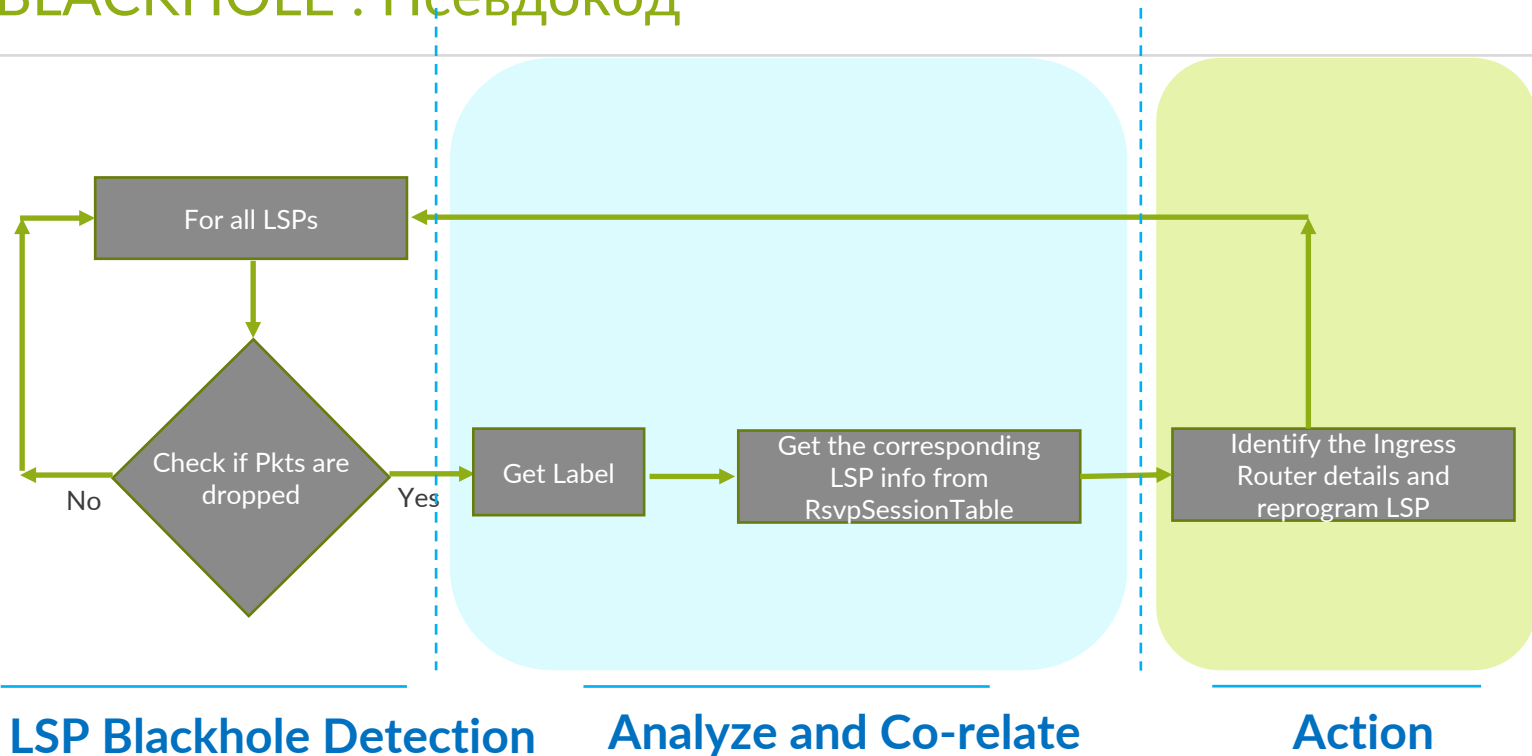
Детектирование, анализ и коррекция

LSP BLACKHOLE : Детектирование

- Идентифицировать состояния LSP Blackhole в сети и перепрограммировать те LSP, в которых теряются пакеты



LSP BLACKHOLE : Псевдокод

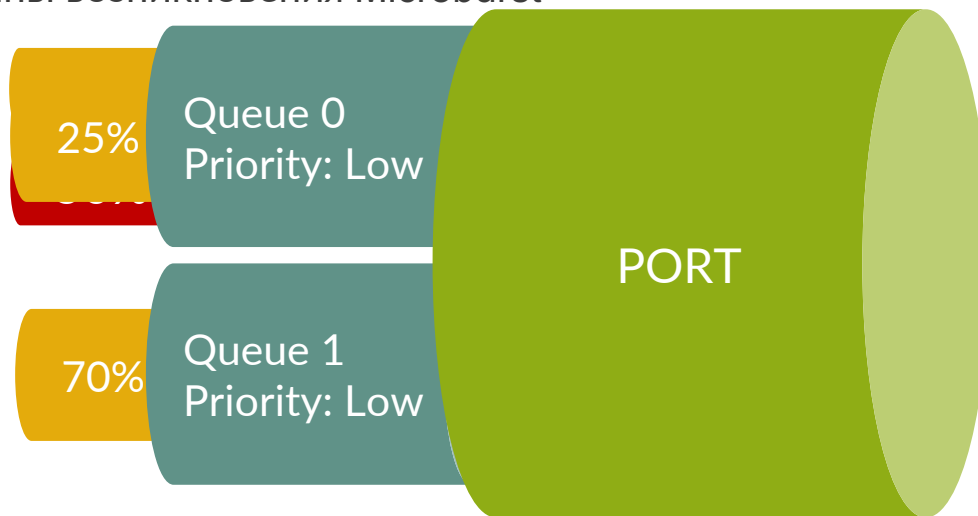


MICROBURST

Детектирование, анализ первопричины

MICROBURST : Детектирование

- Идентификация состояния Microburst
- Использование данных телеметрии – сенсор QMON
- Идентификация интерфейса и очереди, где случился Microburst
- Поиск первопричины возникновения Microburst



Сценарий 1: Microburst в одной очереди



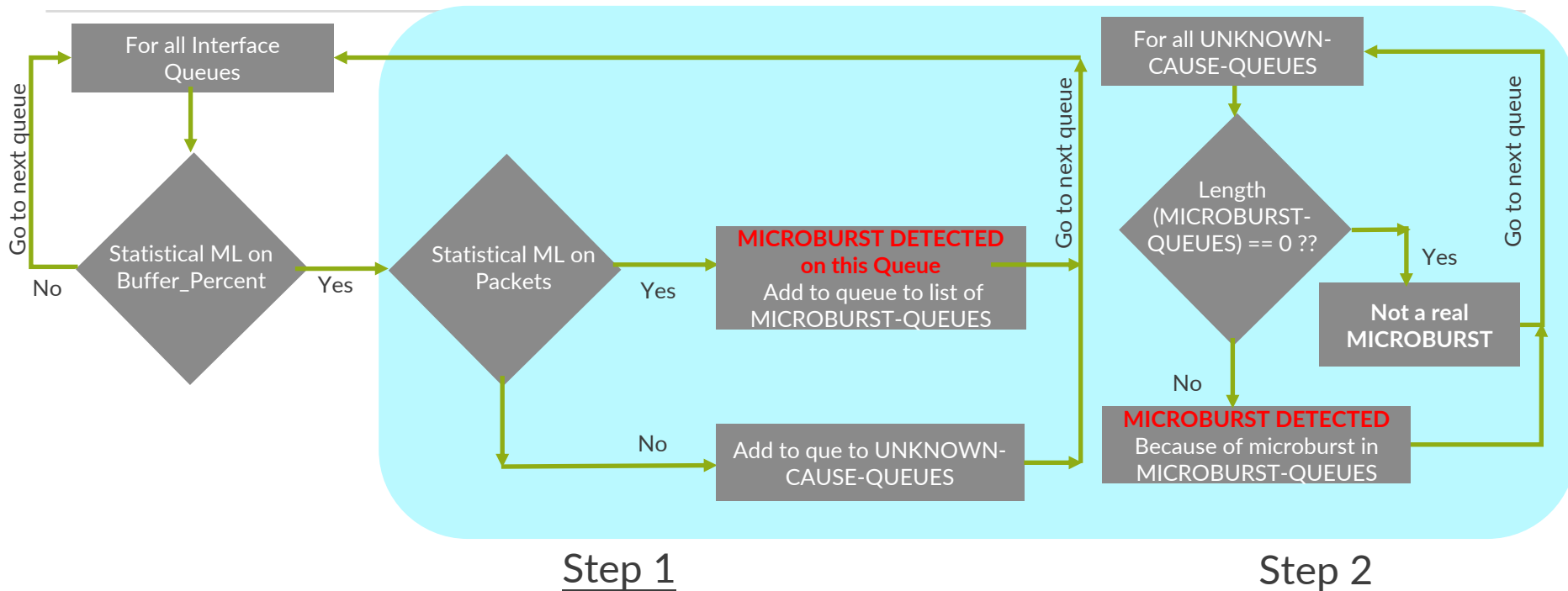
Сценарий 2: Microburst из-за всплеска в другой очереди



Сценарий 3: Увеличение скорости в более приоритетной очереди



MICROBURST : Псевдокод



Microburst Detection

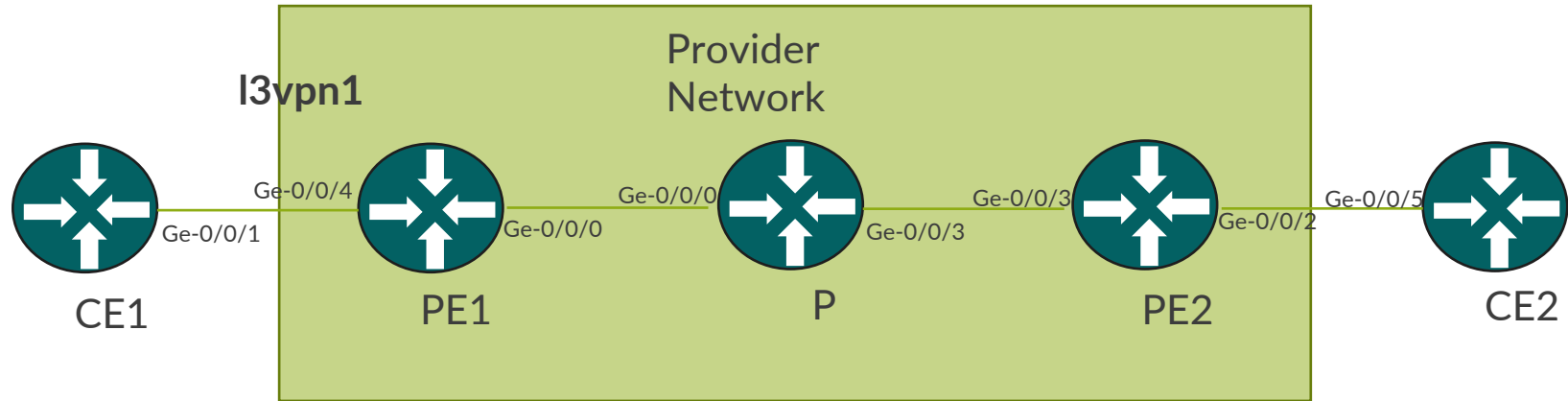
Root Cause Analysis for Microburst

СКВОЗНАЯ ПРОВЕРКА L3VPN

Детектирование, анализ первопричины

Сетевой сенсор

- Сбор данных с нескольких устройств
- Анализ и корреляция данных, поступающих с разных устройств
- Например, проверка разницы скорости передачи/приема пакетов на различных точках включения L3 VPN





Playbooks

+ Create Playbook

⬇ Upload Playbook

>... Playbook Builder CLI

Name	Apply	Live	Delete	Synopsis
interface-playbook				This playbook is to monitor the interface state check for anomaly
isp-blackhole				To check the isp drop condition
microburst-playbook				To check the MicroBurst condition on interface/queues
ospf-diagnostic				To check the health of ospf neighborship
system-kpis-playbook				System key performance indicators

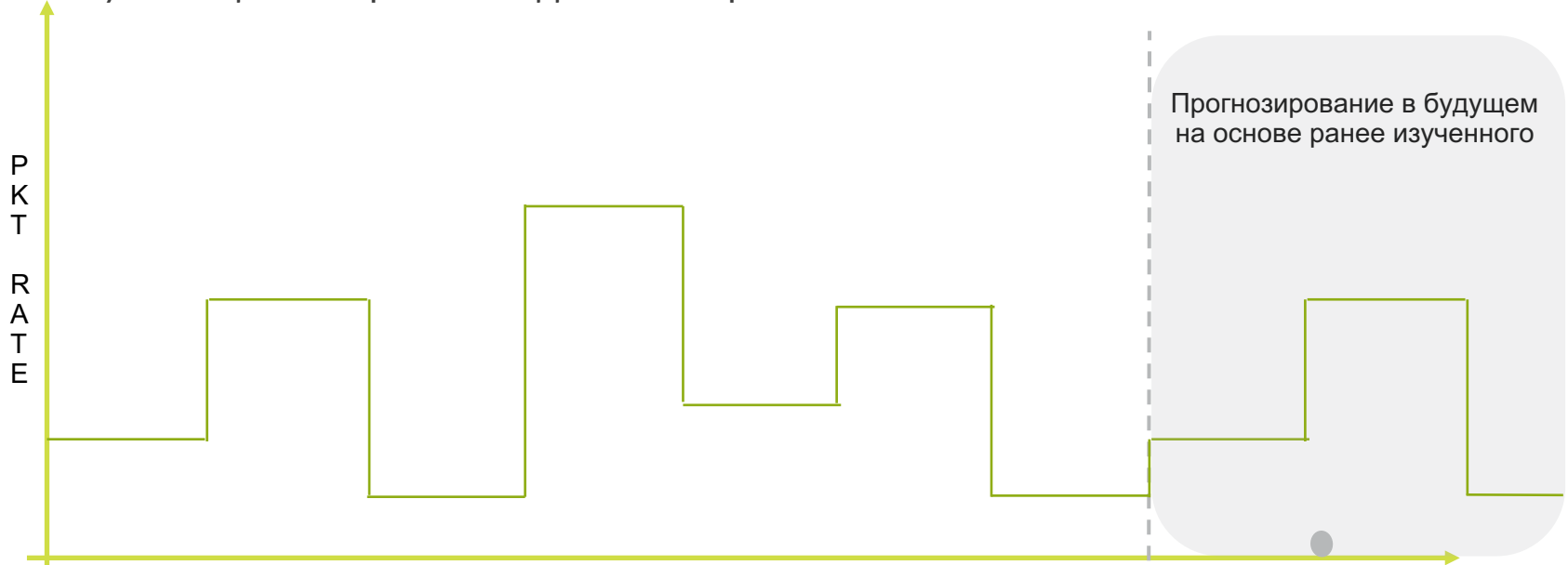


ПРОГНОЗИРОВАНИЕ

на основе прошлого обучения

Прогнозирование

- Изучение исторических данных
- Создание блоков данных на основе периодичности шаблонов
- Визуализация исторических данных и прогноза





Bng-lab-floor-4 ▾

r2edge ▾

▶ Sensor tables

▶ **Field tables**

▶ Prediction tables

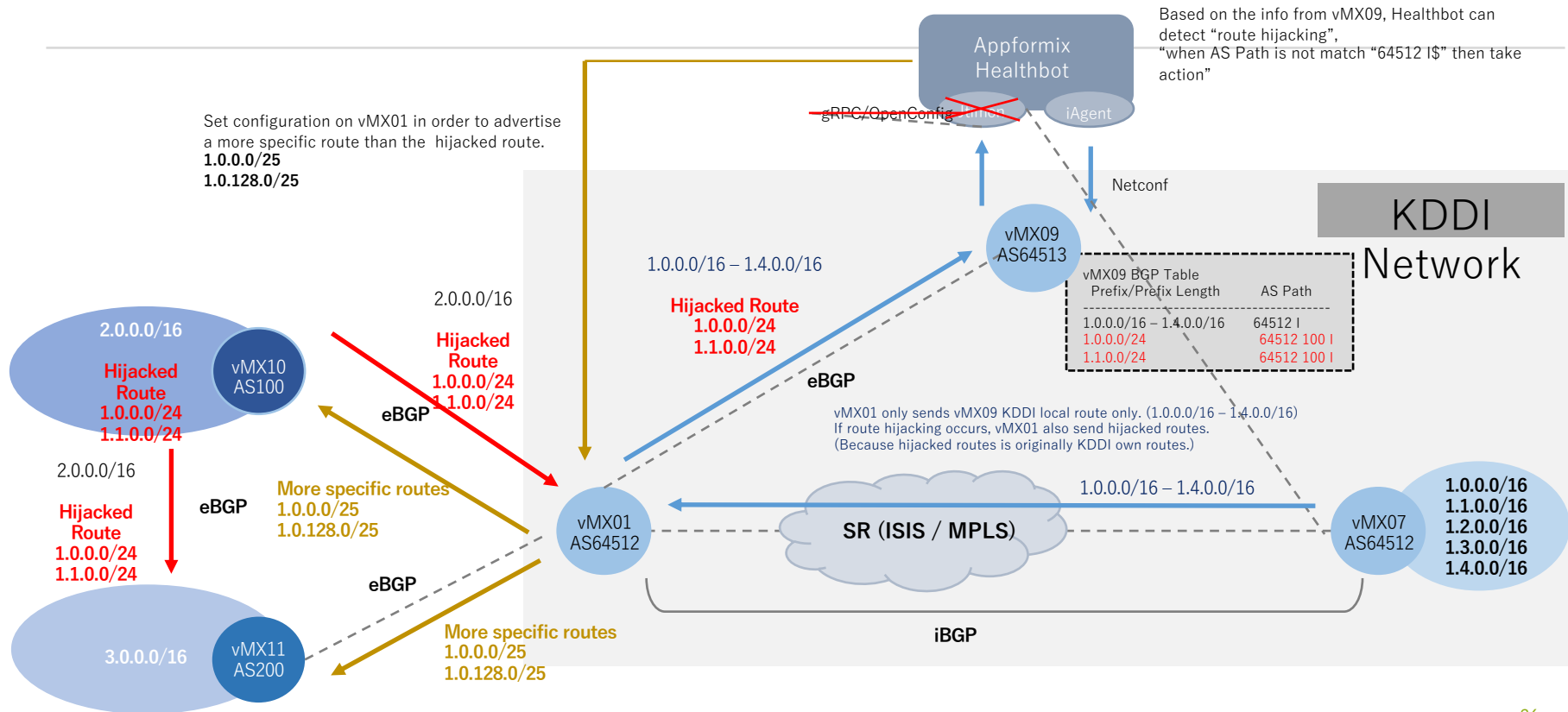
▶ Rule evaluation tables





ДРУГИЕ ЗАДАЧИ И РЕШЕНИЯ

Детектирование и реакция на Route HIJACK



Отладка протоколов маршрутизации

- Идентификация проблем, связанных с протоколами, например флапы OSPF, BGP
- Поиск первопричины по мере детализации стадий распространения маршрутной информации и обработки потоков трафика

```
show ospf stat
```

```
show ospf nei detail
```

```
.....
```

```
Kernel:
```

```
show system process exten
```

```
show system virtual-memory
```

```
.....
```

```
Chassis:
```

```
show chassis routing-engine
```

```
show chassis fpc
```

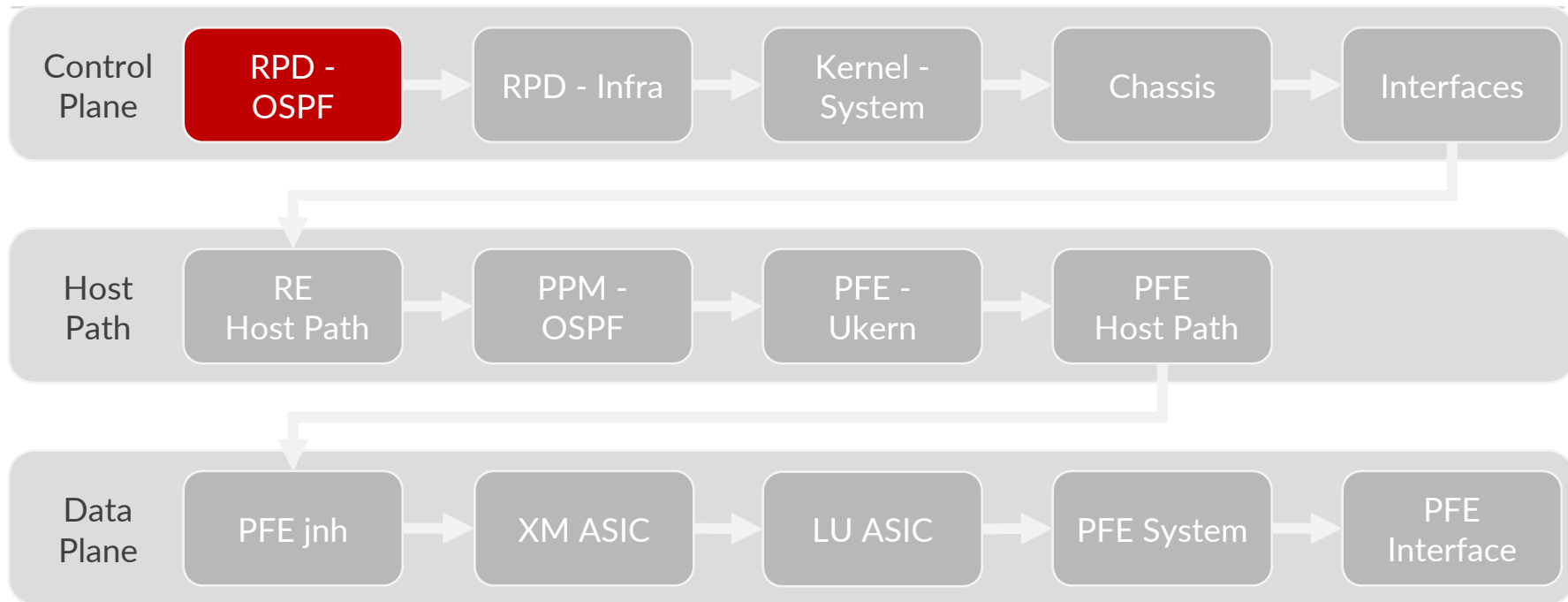
```
.....
```

```
Host path:
```

```
.....
```

Детализация

Процесс отладки/диагностики OSPF



HealthBot – основные компоненты алгоритма

Оценка состояний

Одномерная корреляция KPI

```
if ($temperature > 80) ||  
if ($mem > dynamic threshold)
```

Корреляция между полями

```
if ($in-packet-rate <  
$out-packet-rate)
```

Корреляция между устройствами

```
if (device-1.$in-packet-rate <  
device-2.$out-packet-rate)
```

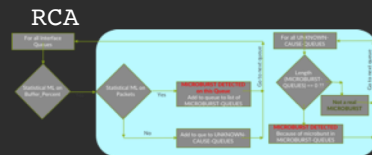
Правила оценки

Простые правила оценки

```
if (condition)  
then:  
flag RED  
flag YELLOW  
flag GREEN
```

Сложное дерево принятия решений

```
if (condition)  
then:
```



Детализация

```
if (condition)  
then:
```

ZOOM-IN

ZOOM-OUT

Действия

Простые действия

```
if (condition)  
then:  
<simple/complex eval>  
send-notification
```

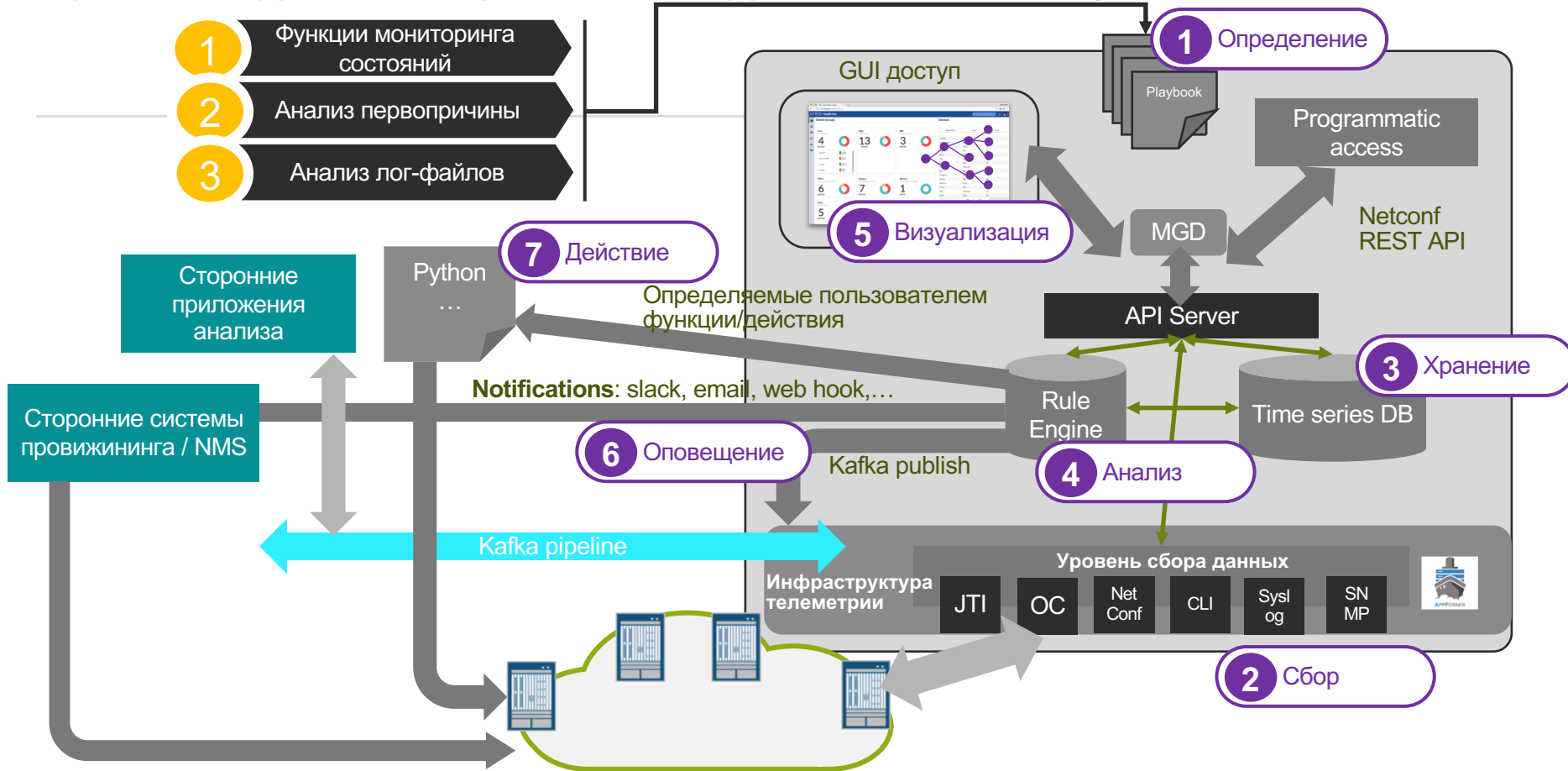
Действия на устройстве

```
if (condition)  
then:  
<simple/complex eval>  
configure-on-device
```

Замкнутый цикл автоматизации

```
if (condition)  
then:  
<simple/complex eval>  
web-hook  
write-to-Kafka  
RUN-PYTHON-SCRIPT
```

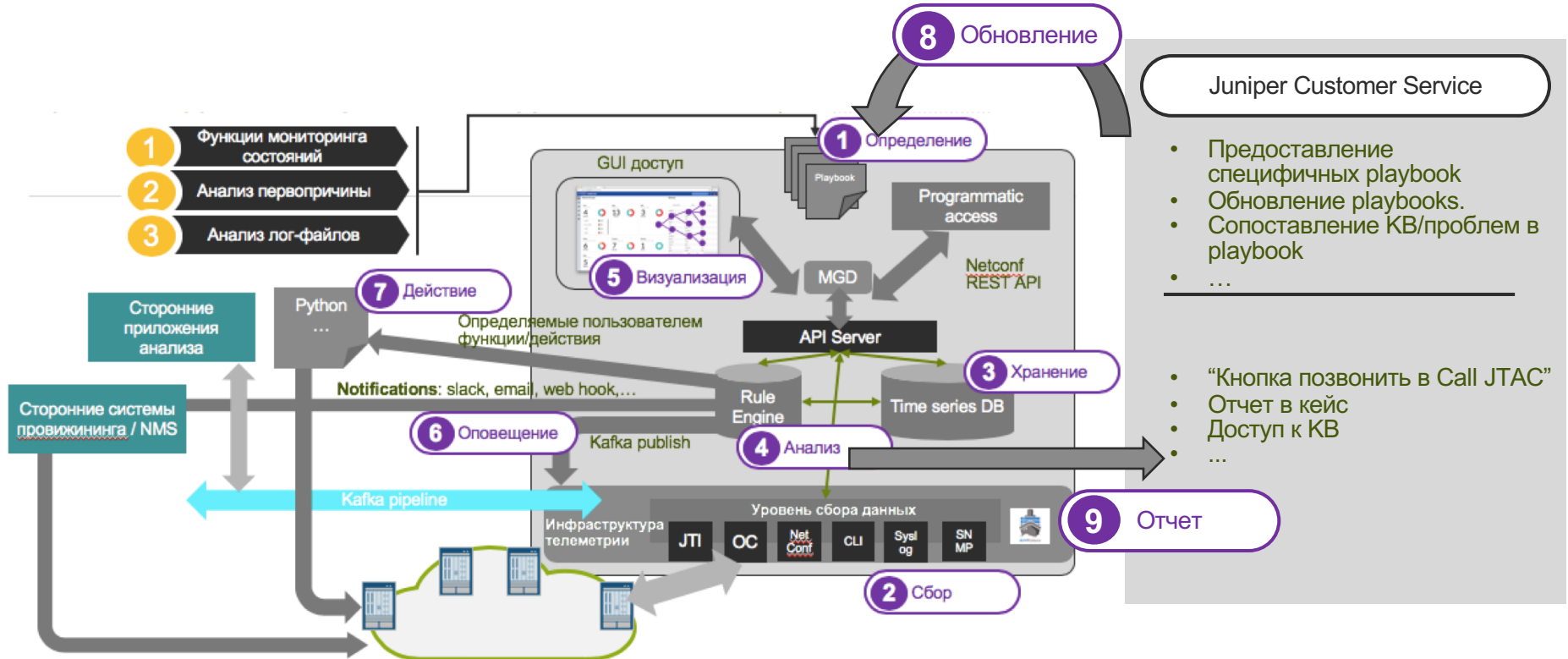
Архитектура замкнутого контура автоматизации HealthBot



HealthBot и служба поддержки пользователей

* планируется

Дополнительный инструмент проактивного сервиса



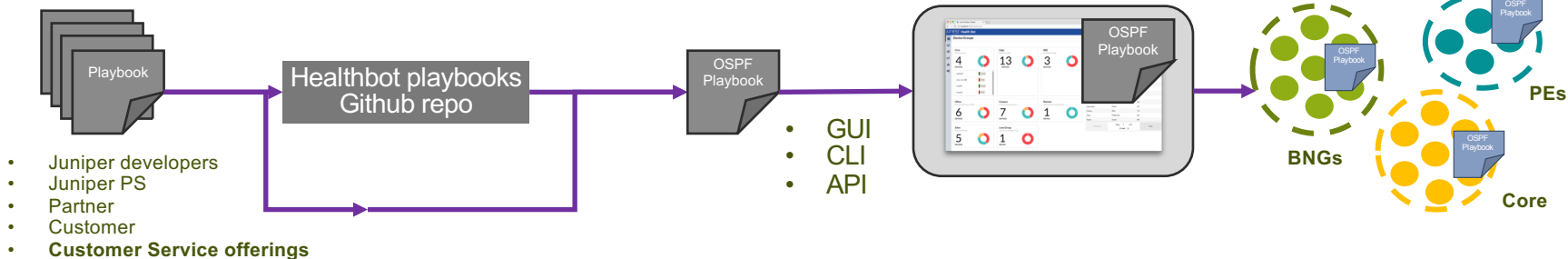
Использование Playbook

1 Разработка

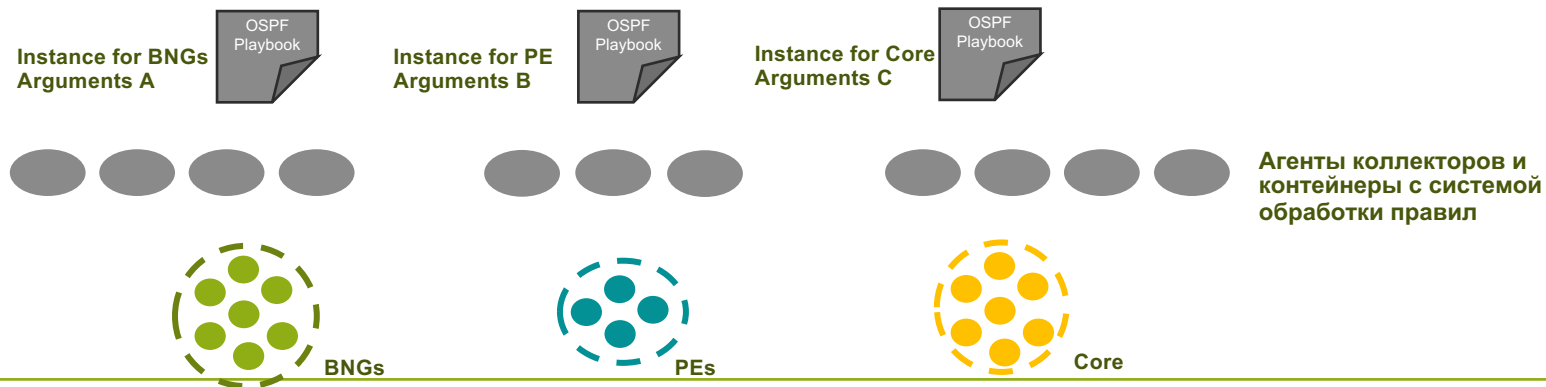
2 Передача в сообщество

3 Загрузка в HealthBot приложение

4 Применение на устройствах



5 Масштабирование: создание нескольких экземпляров и контейнеров



Проект HelathBot Rules в Github

GitHub, Inc. [US] | https://github.com/Juniper/healthbot-rules

Search or jump to... Pull requests Issues Marketplace Explore

Juniper / healthbot-rules Private Watch 31 Star 2 Fork 14

<> Code Issues 0 Pull requests 8 Projects 0 Wiki Insights

No description, website, or topics provided.

50 commits 3 branches 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

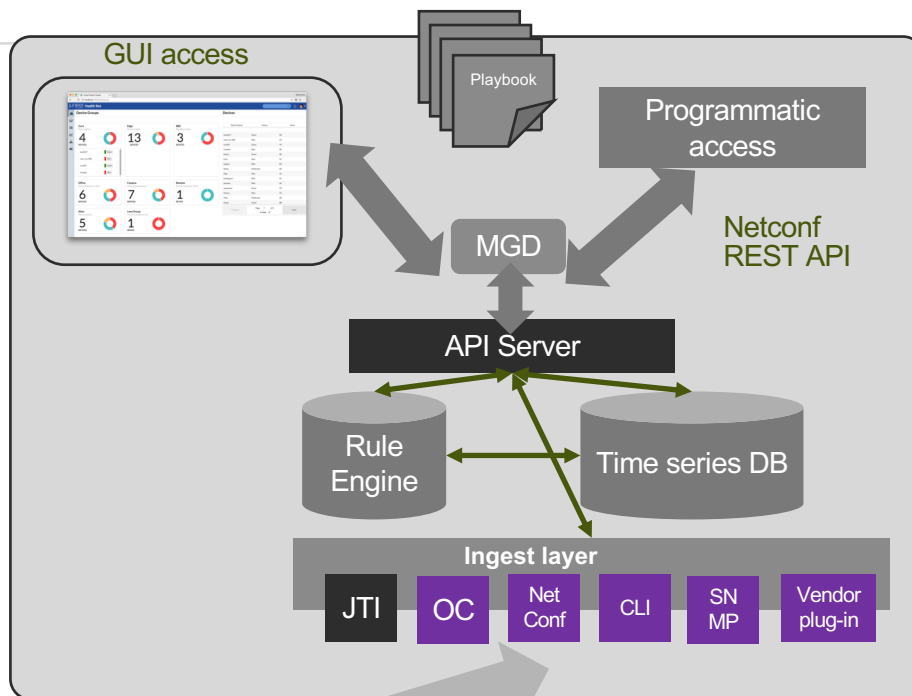
ayushgithub	development to master release september 21th 2018	Latest commit fe224f5 23 days ago
community_supplied	Rename Readme.md to README.md	6 months ago
juniper_official	development to master release september 21th 2018	23 days ago
.gitignore	fixing .gitignore	6 months ago
README.md	Update README.md	5 months ago

README.md

How to write rules

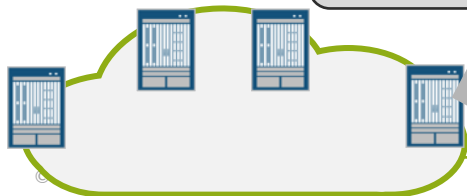
Please follow the DATA model user guide present [here](#). Submit only the .conf files

Поддержка сторонних вендоров в HealthBot



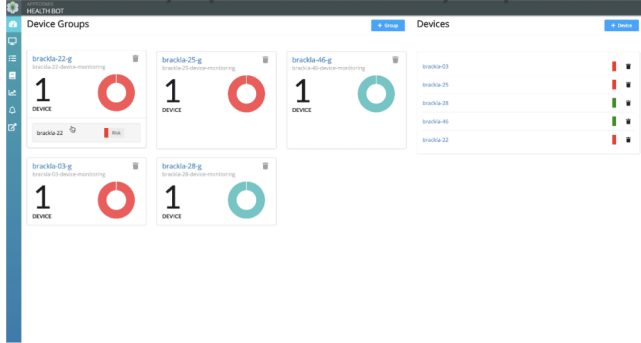
Опции поддержки сторонних вендоров:

- Стандартная телеметрия Open Config
- Интеграция NAPALM для Netconf & CLI.*
- Сбор данных SNMP *
- Возможность подключения специфичных плагинов

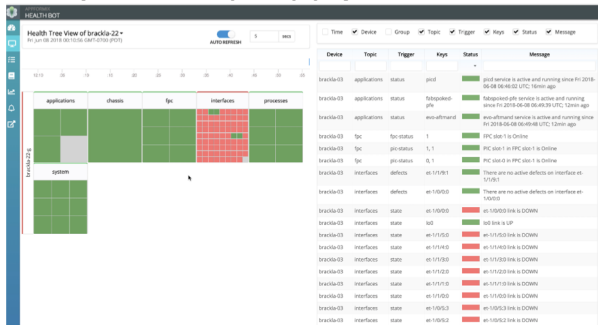


Healthbot – интерфейс управления

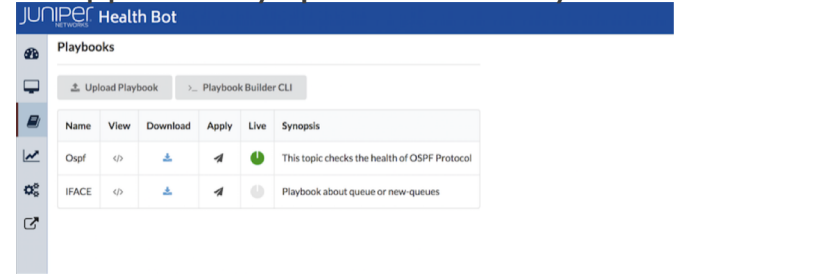
Панель управления – устройства и группы



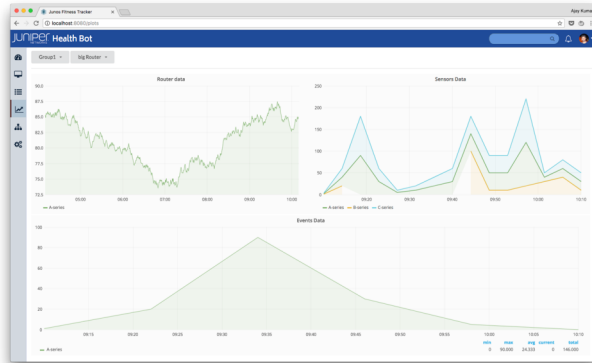
Визуализация дерева состояний



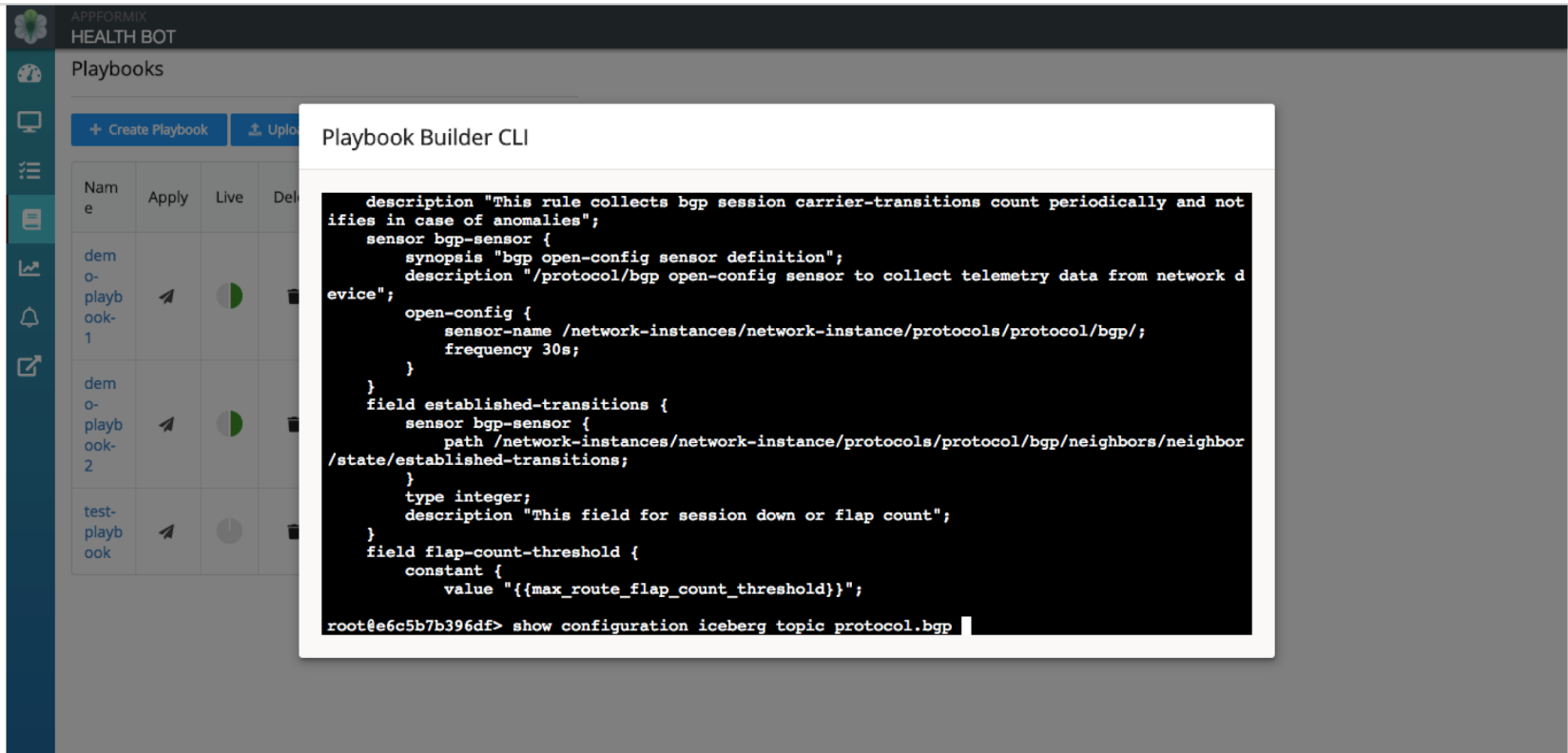
Создание и управления Playbook и Rule



Общая инспекция



Встроенная консоль управления HealthBot



The screenshot shows the Juniper HealthBot interface. On the left is a navigation sidebar with icons for Home, Playbooks, Alerts, and Reports. The main area is titled 'HEALTH BOT Playbooks' and contains a table of playbooks. A 'Playbook Builder CLI' window is overlaid on top, displaying a configuration script for a BGP sensor.

Name	Apply	Live	Delete
demo-playbook-1	🔍	🟢	🗑️
demo-playbook-2	🔍	🟢	🗑️
test-playbook	🔍	🟡	🗑️

```
description "This rule collects bgp session carrier-transitions count periodically and notifies in case of anomalies";
sensor bgp-sensor {
  synopsis "bgp open-config sensor definition";
  description "/protocol/bgp open-config sensor to collect telemetry data from network device";
  open-config {
    sensor-name /network-instances/network-instance/protocols/protocol/bgp/;
    frequency 30s;
  }
  field established-transitions {
    sensor bgp-sensor {
      path /network-instances/network-instance/protocols/protocol/bgp/neighbors/neighbor/
/state/established-transitions;
    }
    type integer;
    description "This field for session down or flap count";
  }
  field flap-count-threshold {
    constant {
      value "{{max_route_flap_count_threshold}}";
    }
  }
}
root@e6c5b7b396df> show configuration iceberg topic protocol.bgp
```



JUNIPER NETWORKS | JUNIPER DAY
16 октября 2018

ВОПРОСЫ?

dkaryakin@juniper.net