

Service Assurance for Intent-based Networking Architecture & YANG Modules for Service Assurance

[draft-claise-opsawg-service-assurance-architecture-03](#)

B. Claise (Cisco), J. Quilbeuf (Cisco), Y. El Fathi (Orange), D. Lopez (Telefonica), D. Voyer (Bell Canada)

[draft-claise-opsawg-service-assurance-yang-05](#)

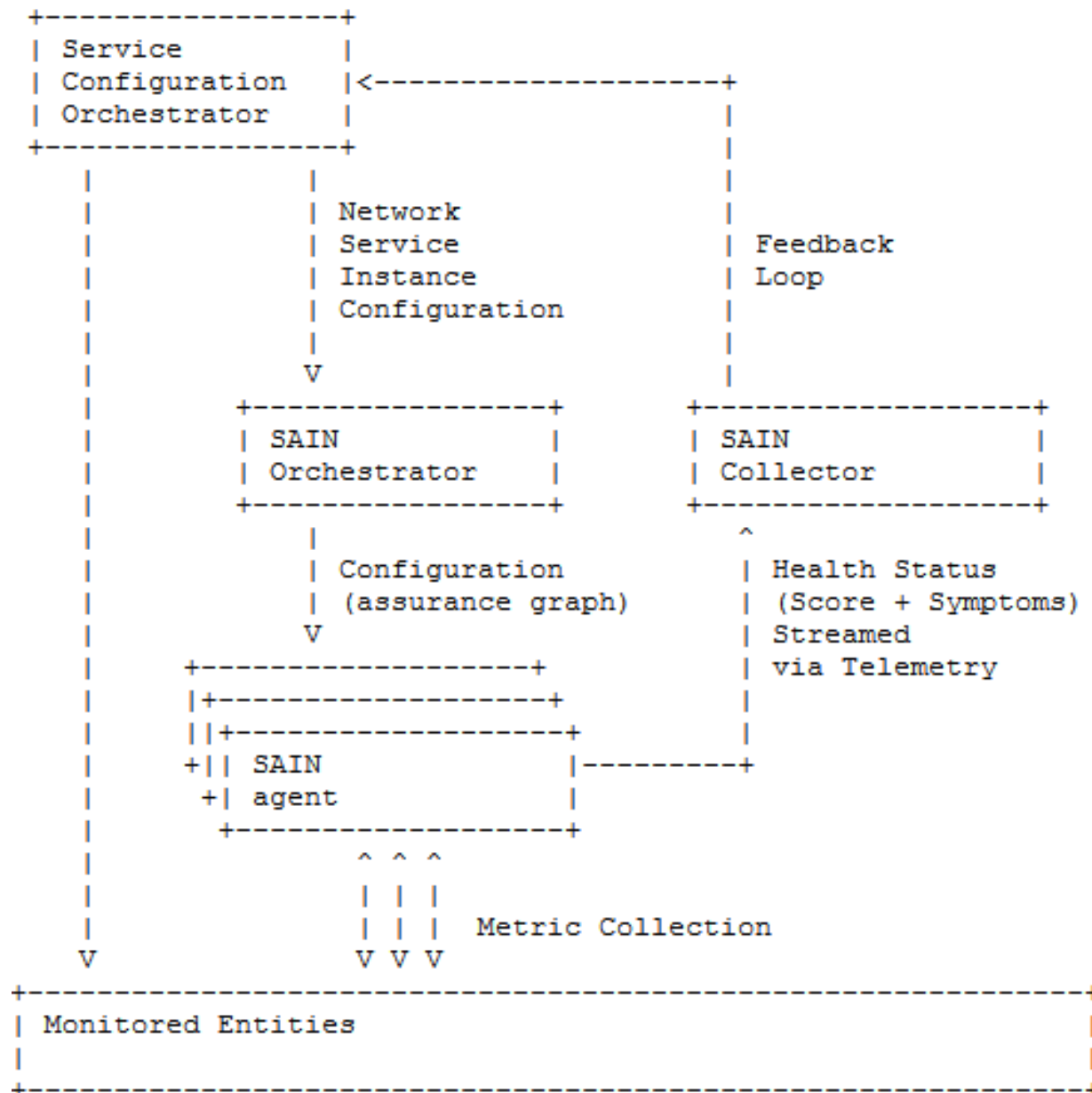
B. Claise (Cisco), J. Quilbeuf (Cisco), P. Lucente (NTT), P. Fasano (Telecom Italia Mobile)

IETF 108, Virtual

Issue & Proposal

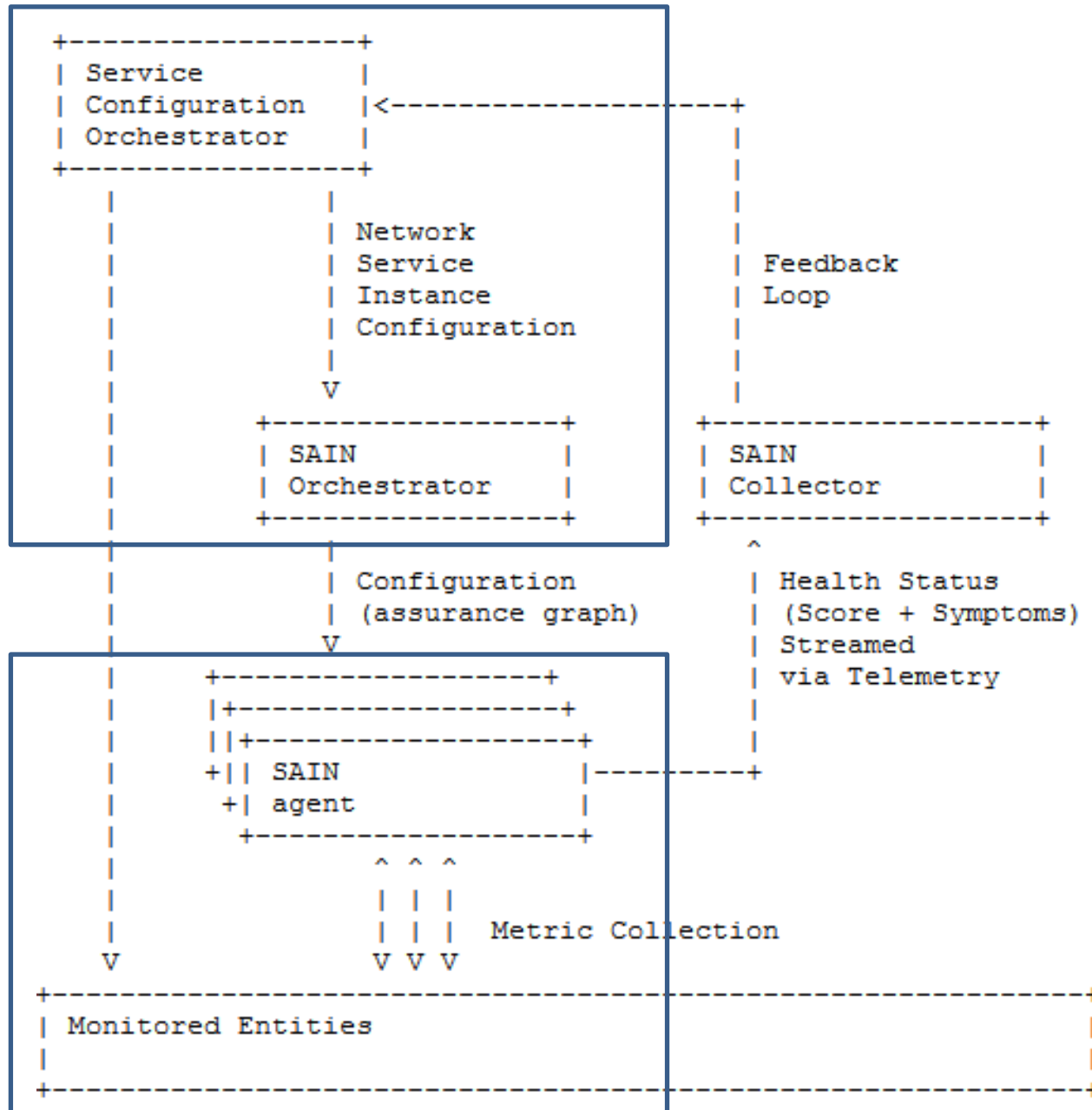
- Issues:
 - When a service degrades, where is the fault? what are the symptoms? what is the root cause?
 - When a network component fails, which services are impacted?
- Service Assurance for Intent-based Networking Architecture proposal:
 - Decompose the problem into smaller components (=subservices)
 - The assurance graph links those subservices to map the service “intent”
 - The subservices are assured independently
 - Infer a service health score
- This complements the end-to-end synthetic testing

Architecture



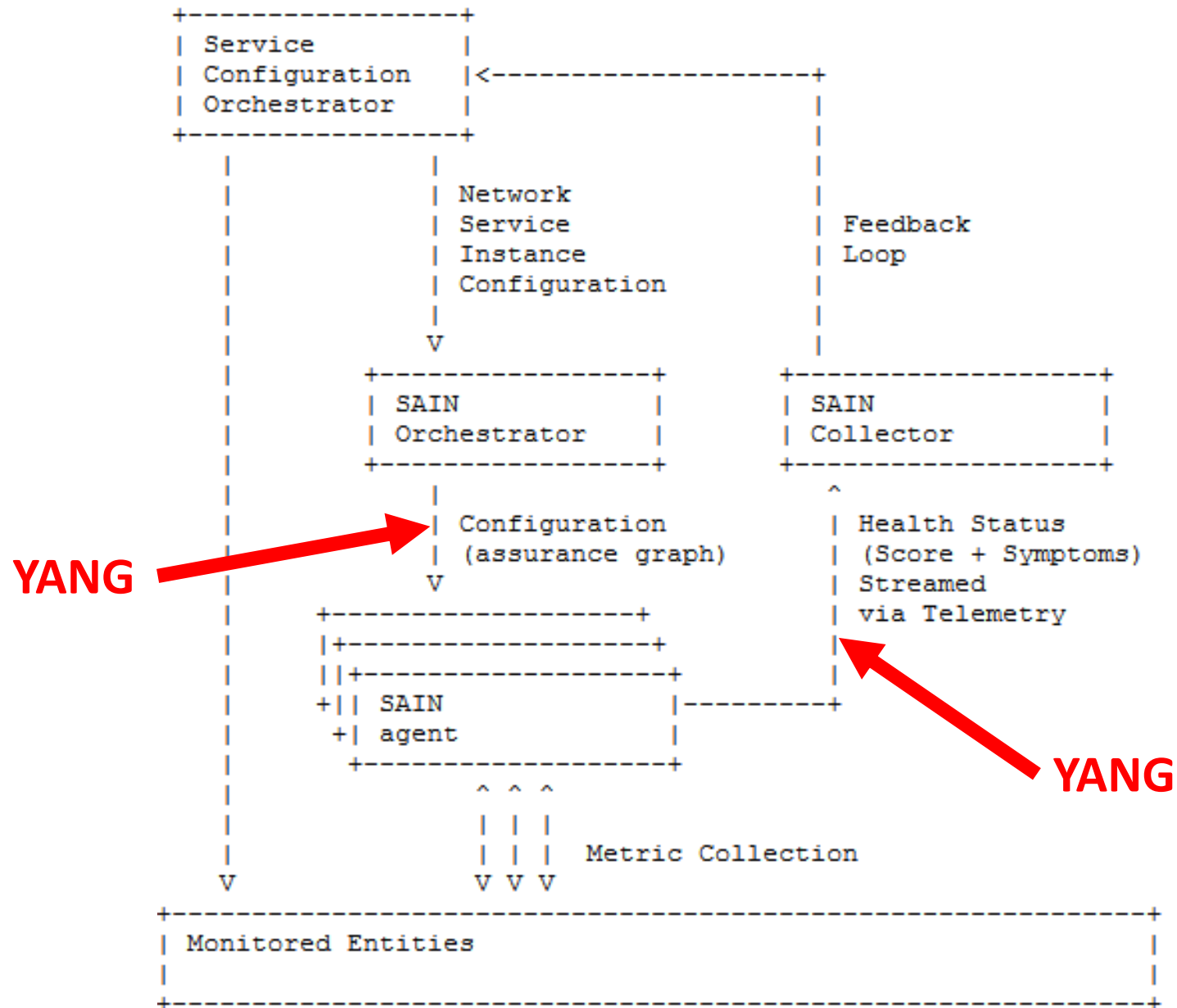
Flexible Architecture

Could be a single box

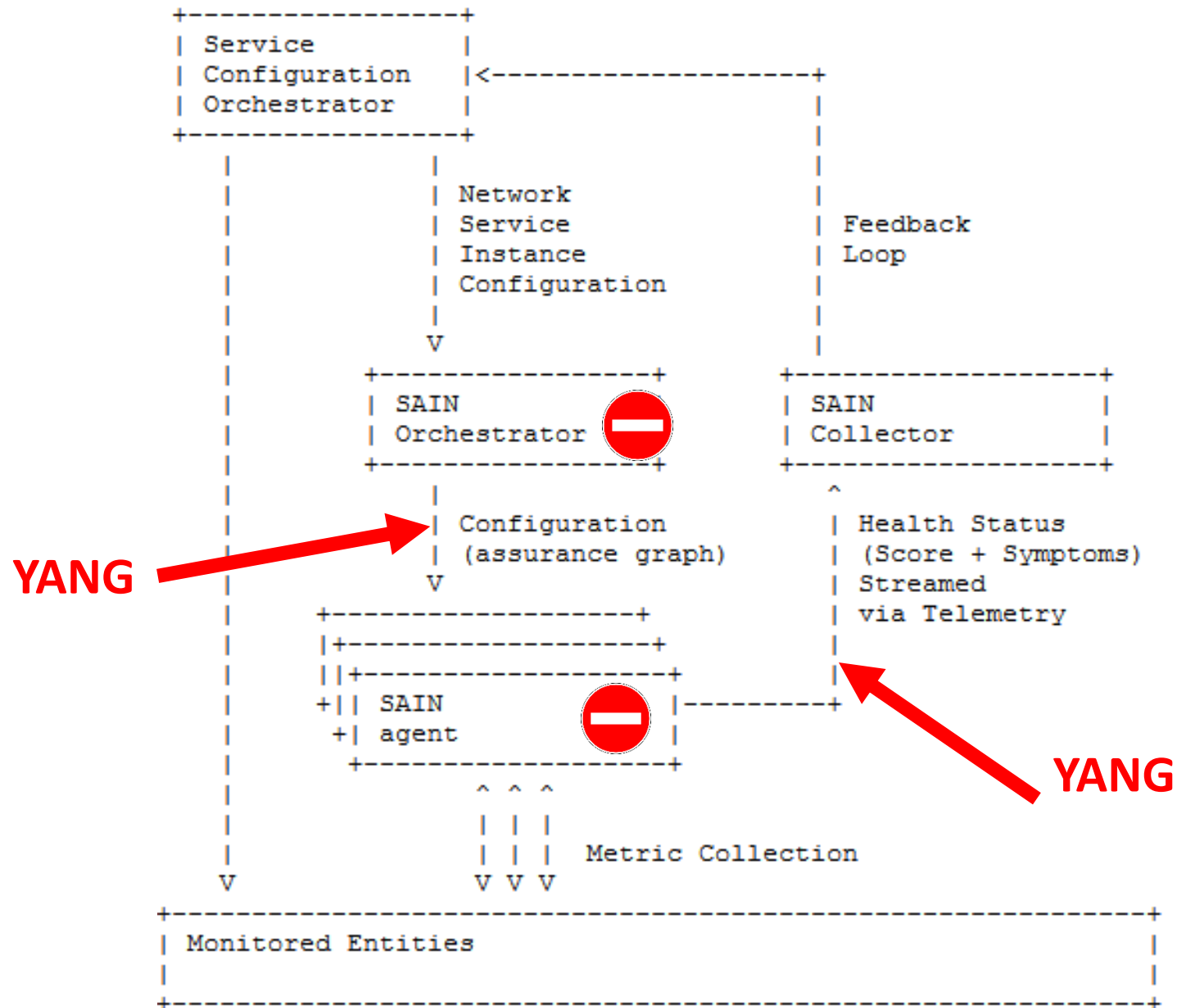


Agent could be in or off routers

Open Architecture with YANG Models



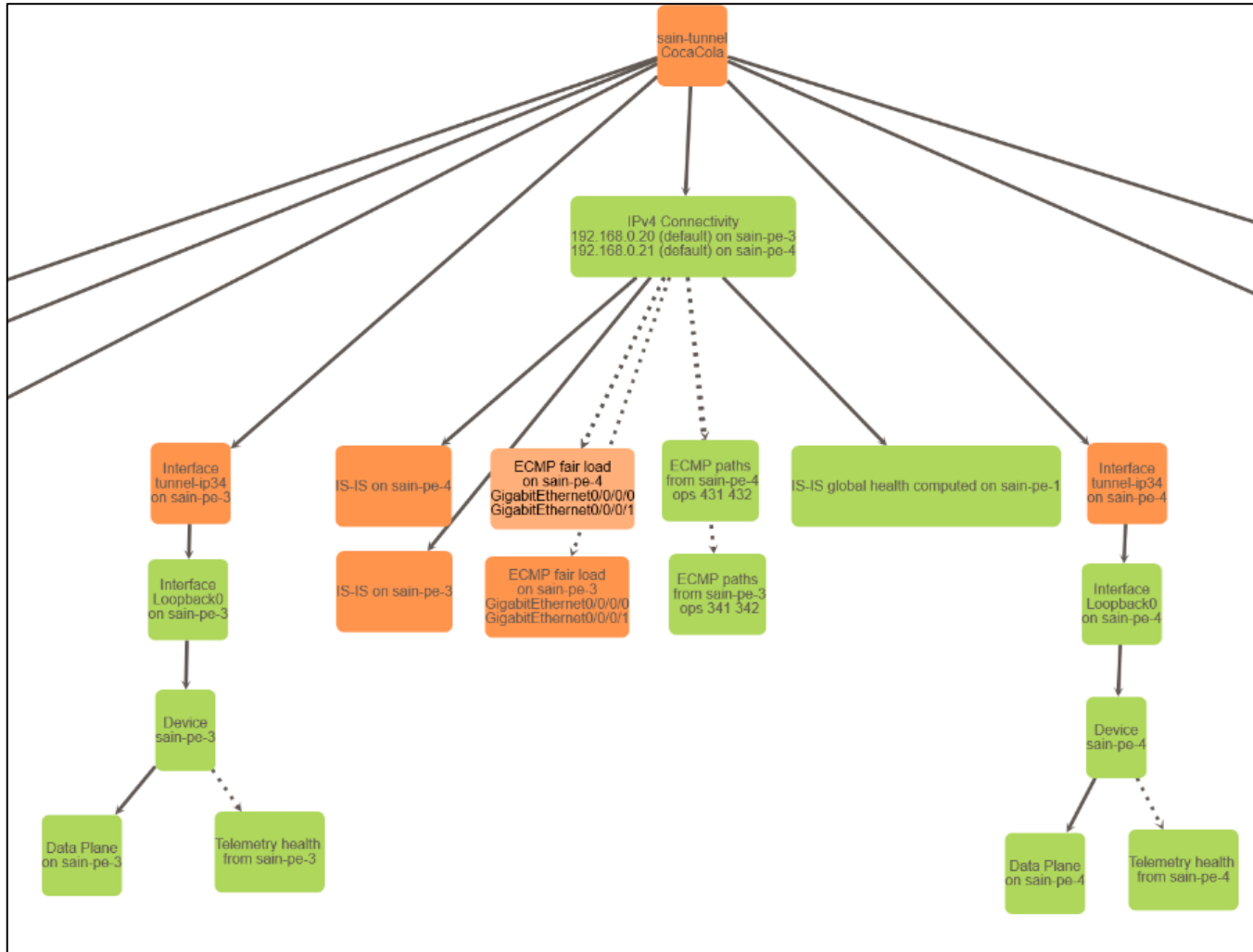
Open Architecture with YANG Models



Changes since Last IETF

- Stressed the generic architecture:
applicable to wireline, wireless, 5G, VIM, etc.
- Timing: NTP is required all over the place
- Introduced the symptoms history start
"Date and time at which the symptoms history starts for this subservice instance, either because the subservice instance started at that date and time or because the symptoms before that were removed due to a garbage collection process."

Assurance Graph PoC



ECMP fair load on sain-pe-3
GigabitEthernet0/0/0/0
GigabitEthernet0/0/0/1
Value: 0.5
Expression tree

List of impacted services:

- sain-tunnel CocaCola
- I2vpn-p2p HSBC
- sain-tunnel-ipv6 RedBull

Symptoms/Root causes:

- Output traffic on interface GigabitEthernet0/0/0/0 in ECMP bundle with(GigabitEthernet0/0/0/1) is not fairly balanced
- Output traffic on interface GigabitEthernet0/0/0/1 in ECMP bundle with(GigabitEthernet0/0/0/0) is not fairly balanced

Changes since Last IETF Hackathon: Lessons Learned

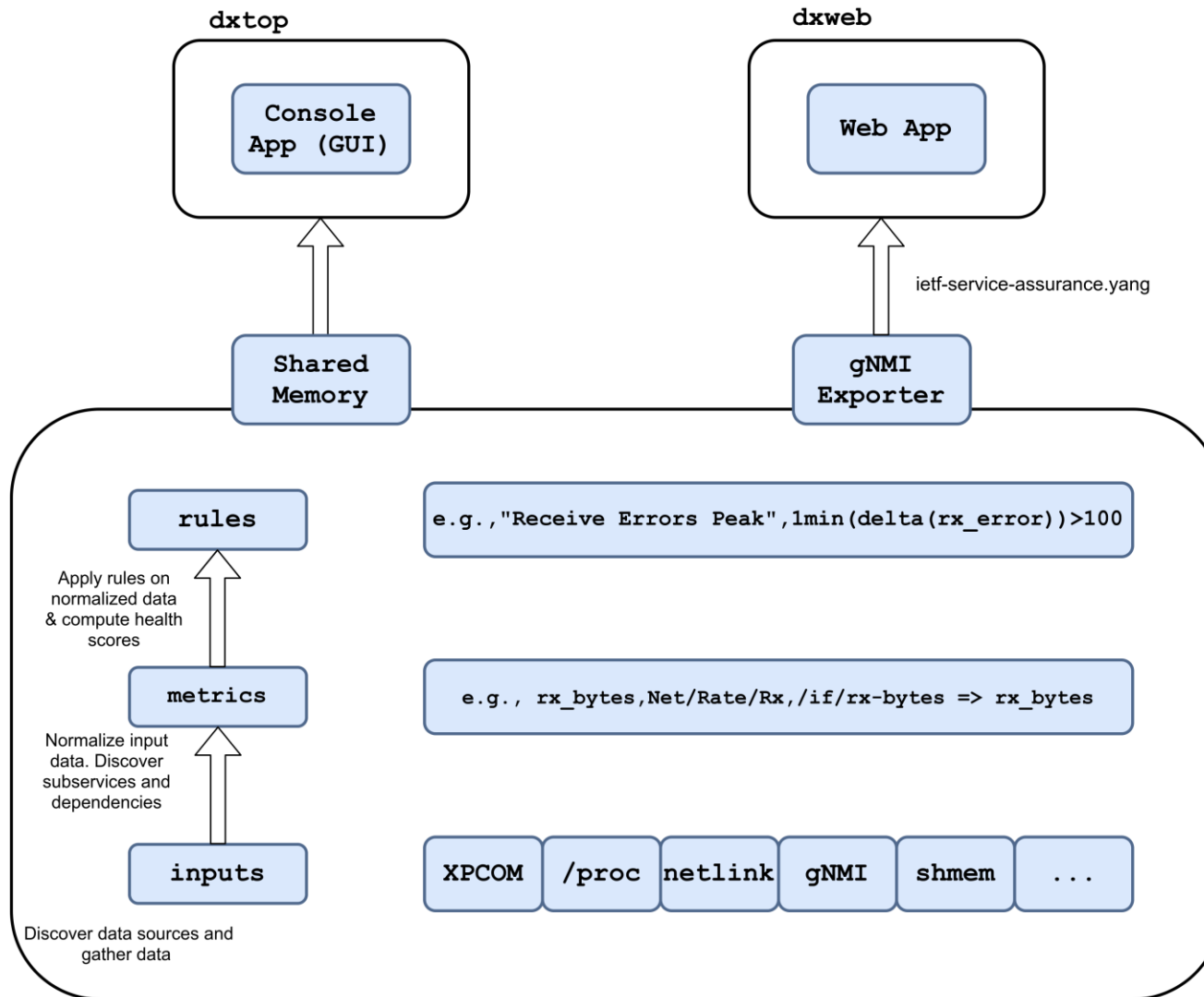
- Assurance graph version & last change now compulsory
 - while per sub-service last change remains optional
- Explain what a change means
- Rename symptoms “label” to “description”
 - Not to be confused with the subservice label

Service Assurance for Intent-based Networking Architecture & YANG Modules for Service Assurance

HACKATHON
Korian Edeline (Liège University)

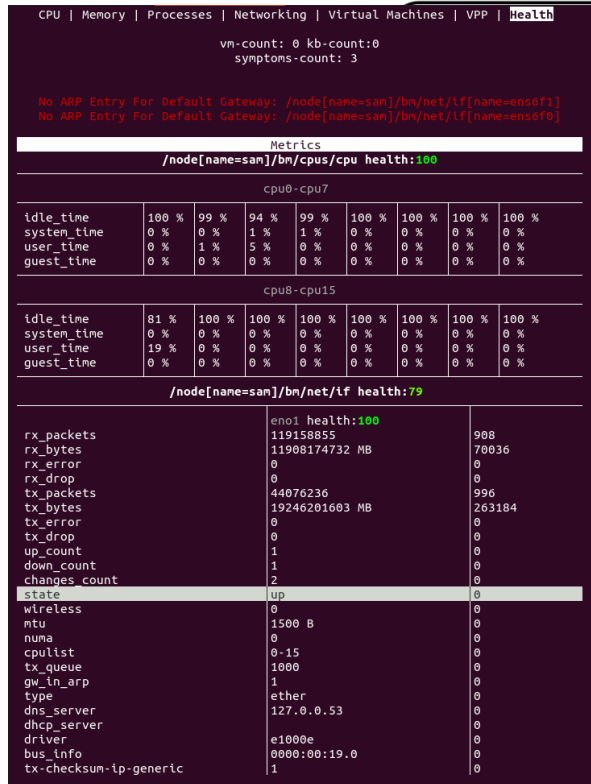
IETF 108, Virtual

An open-source SAIN agent

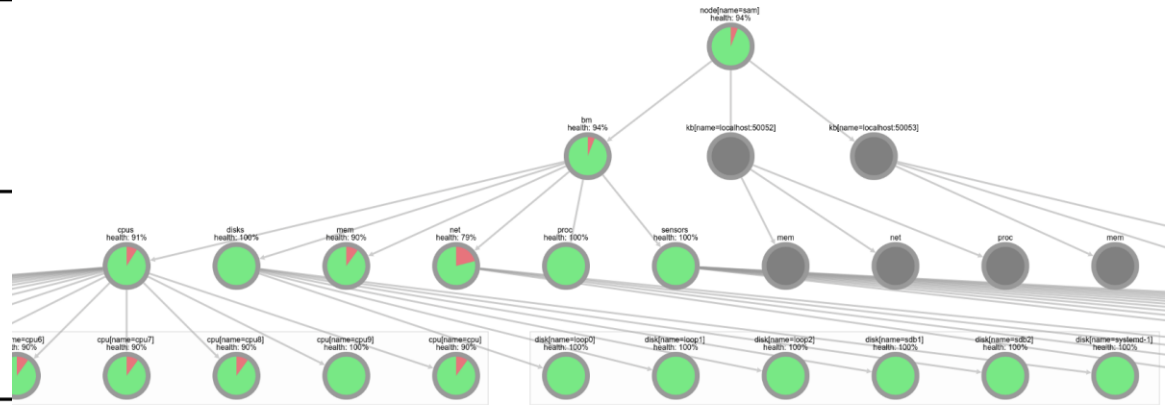


An open-source SAIN agent

dxtop



dxweb



e.g., rx_bytes, Net/Rate/Rx, /if/rx-bytes => rx_bytes

inputs

XPCOM

/proc

netlink

gNMI

shmem

...

Discover data sources and gather data

Rule Engine : Highlighting symptoms

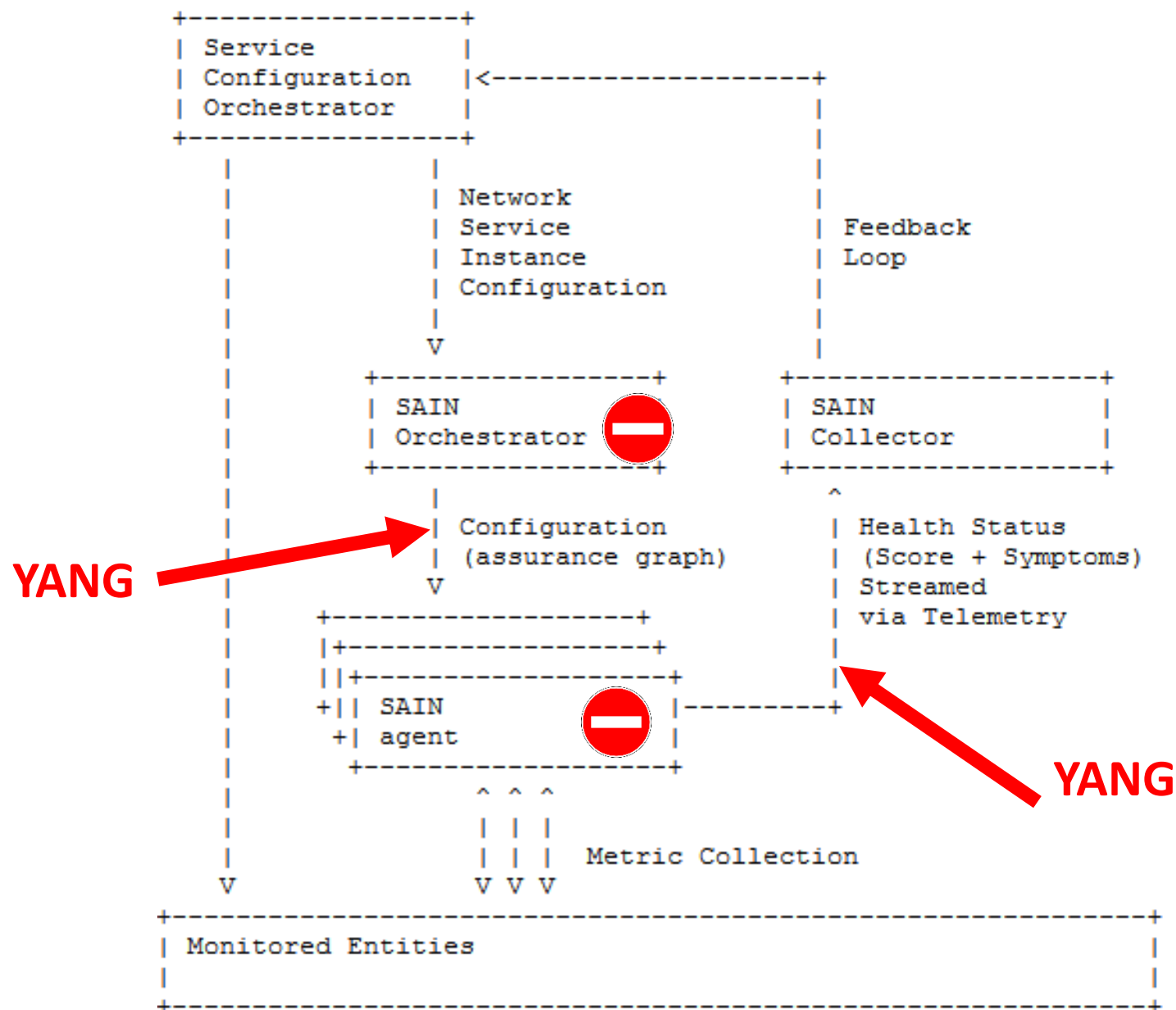
- Subservice expertise for anomaly-highlighting rules
- Variables (metrics), basic operators and more (temporality, selection, has_changed, ...)

```
"Interface Flapping",           /node/bm/net/if,      Red,      1min(dynamicity(changes_count))>=6
"Low Buffer Availability",       /node/kb/mem,         Orange,   (buffer_free/buffer_total)<0.1
"DPDK Buffer Alloc Errors",     /node/kb/net/if,      Orange,   dynamicity(dpdk_alloc_errors)>0
"Sensor reached critical temperature", /node/bm/sensors/sensor, Red,      input_temp>=critical_temp
"Non-standard Ethernet MTU",    /node/bm/net/if,      Red,      (mtu!=1500) and (type=="ether")
```

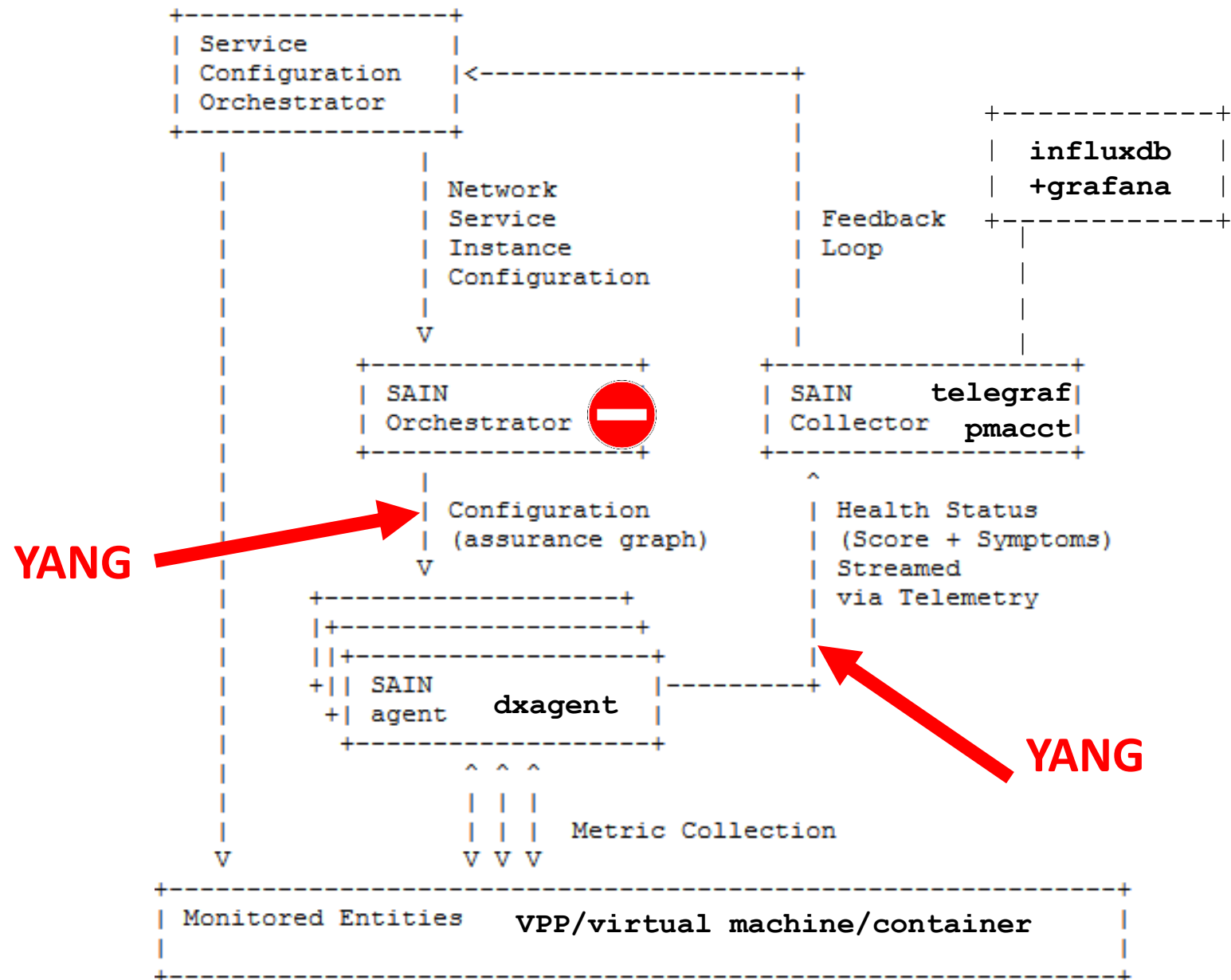
An open-source SAIN agent

- Client service monitoring by concatenating assurance trees
- From failing component, find impacted subservices
- Monitor multiple subservices or components

SAIN Hackaton : Open Architecture with YANG Models



SAIN Hackaton : Open Architecture with YANG Models

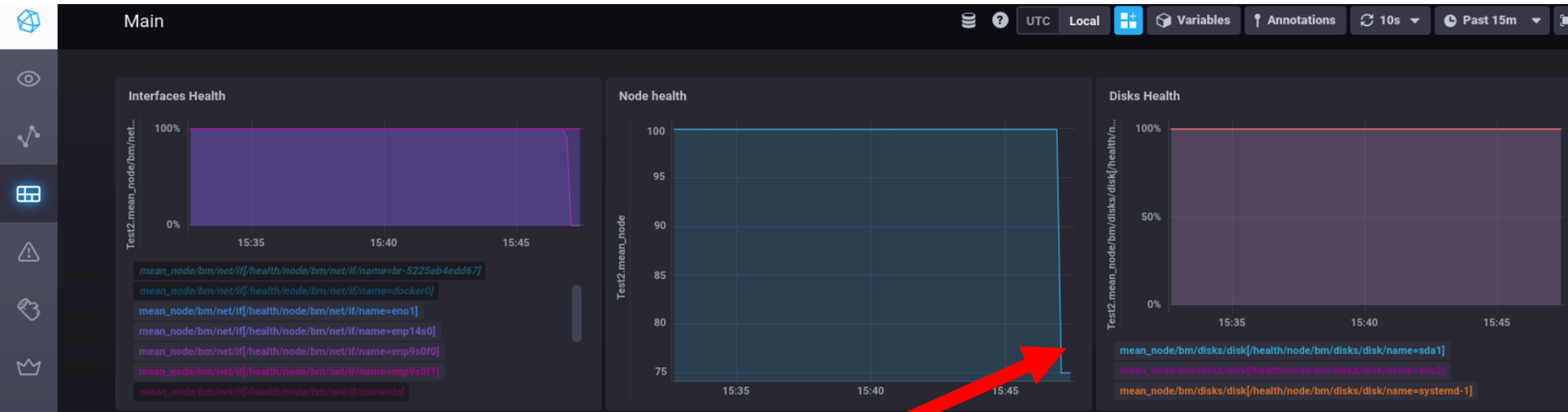


SAIN Hackaton : Open Architecture with YANG Model (Example)



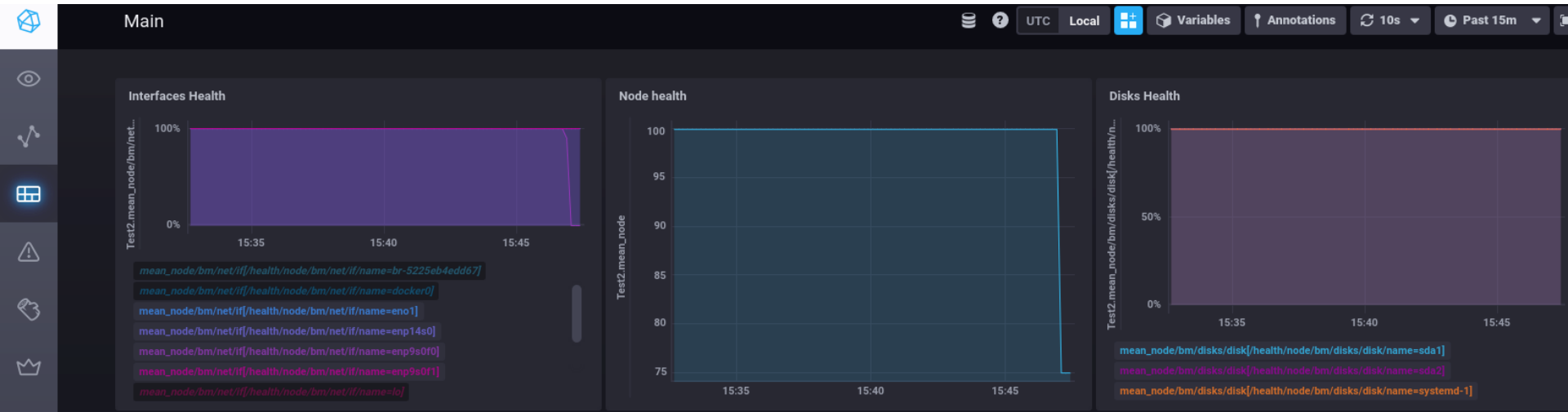
- Monitor a VPP-in-VM instance

SAIN Hackaton : Open Architecture with YANG Model (Example)

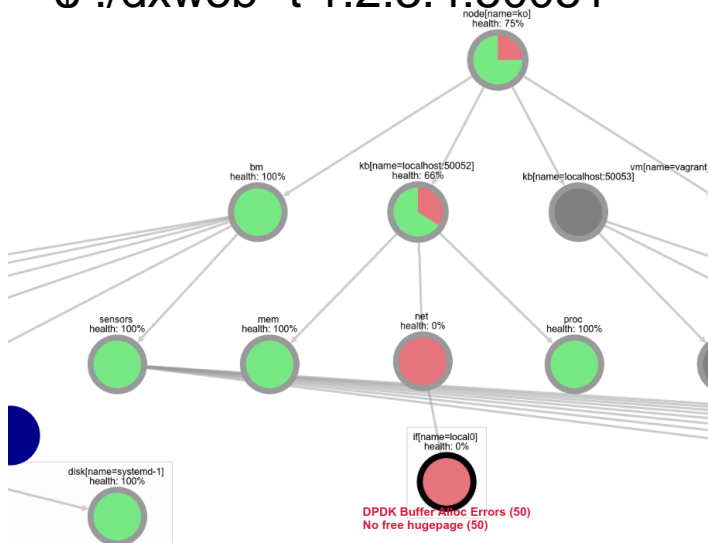


- Health score decreases

SAIN Hackaton : Open Architecture with YANG Model (Example)



\$./dxweb -t 1.2.3.4:50051



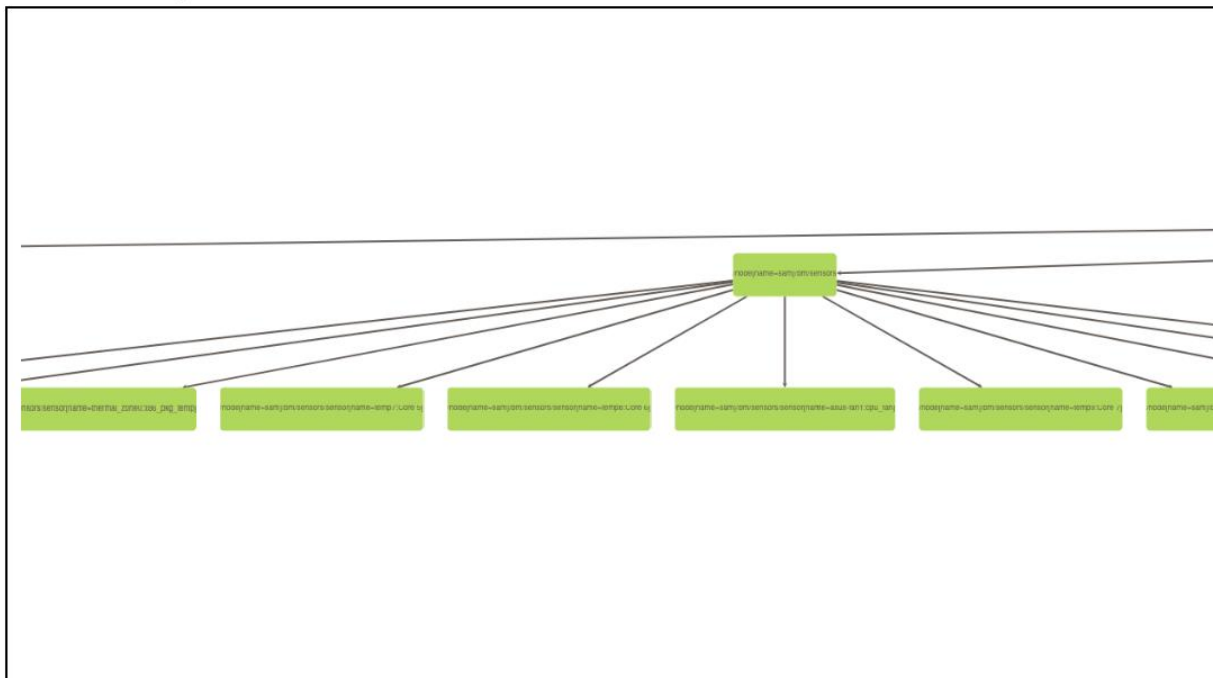
\$ ssh "dxtop"

CPU Memory Processes Networking Virtual Machines VPP Health								
vm-count: 1 kb-count:1 symptoms-count: 2								
Symptoms								
DPDK Buffer Alloc Errors: /node[name=ko]/kb[name=localhost:50052] No free hugepage: /node[name=ko]/kb[name=localhost:50052]								
Metrics								
/node[name=ko]/bm/cpus/cpu health:100								
cpu0-cpu7								
idle_time	92 %	94 %	93 %	95 %	91 %	90 %	95 %	66 %
system_time	2 %	2 %	2 %	2 %	2 %	2 %	1 %	1 %
user_time	6 %	3 %	5 %	3 %	8 %	8 %	3 %	33 %
guest_time	0 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %
/node[name=ko]/bm/net/1f health:100								
rx_packets								
vboxnet1 health:100							0	

Interoperability

- Dxagent exporting to the Cisco SAIN collector

Service dependencies



Next Steps

- More complete rule engine
- Add end-to-end probing as input
- Multi-node architecture (path assurance)
- gNMI support in pmacct (Kannan Jayaraman)
- White paper on specific use case
- More input, more rules

<https://github.com/ekorian/dxagent>

Questions & Feedback on both Presentations

- Going in the right direction?
- Time to provide more feedback?
- If yes, please consider as WG adoption.