

# gRPC Services under One Roof

## Introduction and Practical use cases

Reda Laichi  
Principal Consulting Engineer, NOKIA

Saju Salahudeen  
Principal Consulting Engineer, NOKIA

# Agenda

- Introduction to gRPC
- Introduction and use cases for:
  - gNMI
  - gNOI
  - gNSI
  - gRIBI

# GitHub Repo for this talk

<https://github.com/srlinuxamericas/n93-grpc>

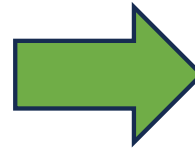


# Containerlab Workshop at N93

Time: Wednesday 2PM

Room: A706 + A707

**Registration  
Required**



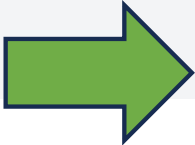

♥ 14:00 - 15:30 (EST) A706 + A707

🚫 Workshop: Containerlab: a Modern way to Design, Deploy, and Test Network Labs - Part I (additional signup required)

Saju Salahudeen - NOKIA  
Reda Laichi - NOKIA

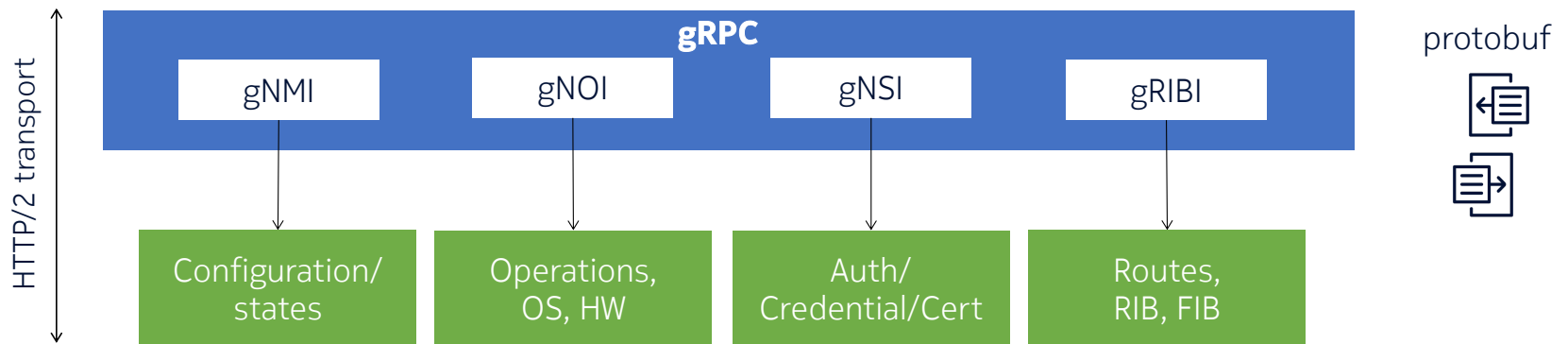
**Full Abstract**

Sign up for a seat in this workshop at <https://forms.gle/AnnCG8p2JFCwxMcE8>



# gRPC-based services

- RPC framework using HTTP 2.0 as the underlying transport
- gRPC does not expose HTTP 2.0 to the user (unlike REST)
- All services leverage Proto Buffer
- Network management and programmability
- Different data models and focus

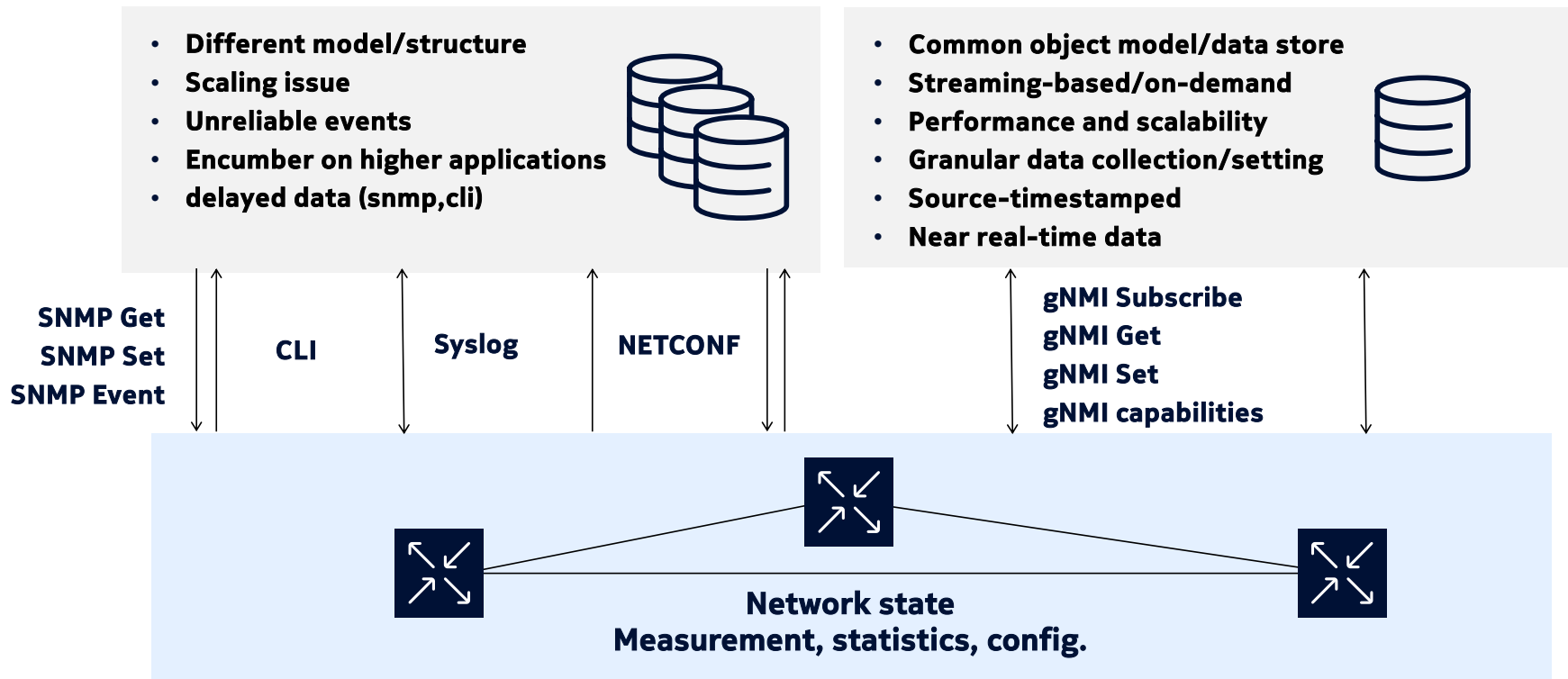


# gNMI

- gRPC Network Management Interface
- gRPC based service for configuration, state and telemetry
- Standards defined by OpenConfig - <https://github.com/openconfig/gnoi>



# gNMI Overview



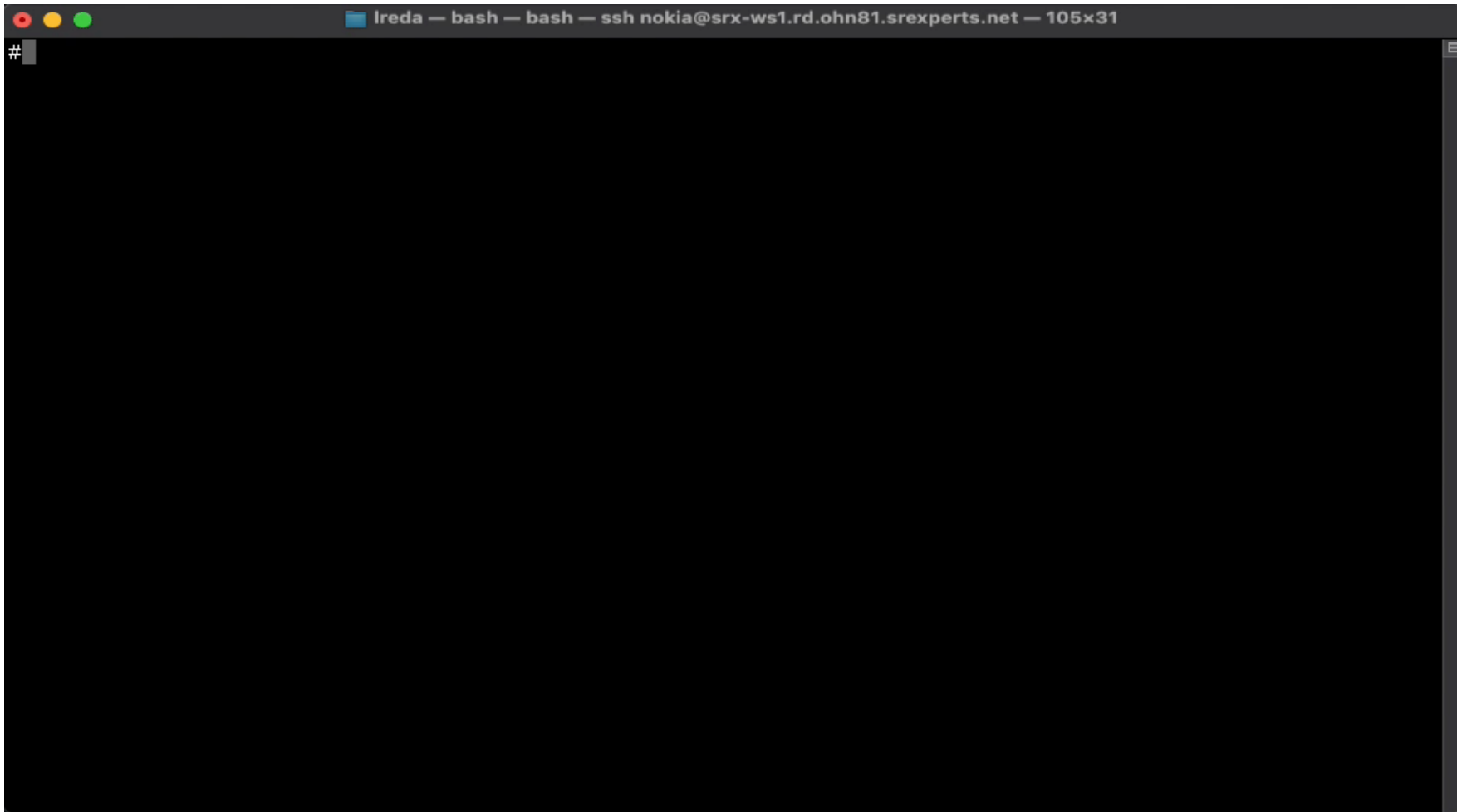
# gNMI Use Cases

- Configuration: Initial deployment, underlay/overlay configuration, update configuration
- State retrieval: Troubleshooting, pre/post-configuration checks, closed-loop automation
- Streaming Telemetry: Statistics collection, performance monitoring, Alarm notification

# gNMI Client

- gNMIC – gNMI CLI client that provides support for gNMI services
- Developed by Nokia and donated to Openconfig
- <https://gnmic.openconfig.net/>

# gNMI for Configuration



# gNMI for state retrieval

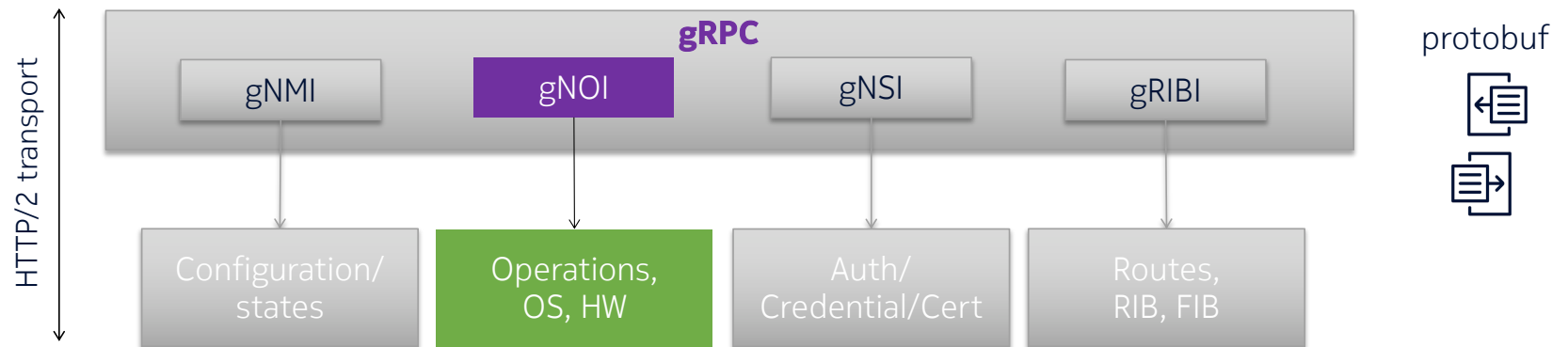
```
nokia@rd-srx-ws1-155afda:~/n93-grpc$  
nokia@rd-srx-ws1-155afda:~/n93-grpc$ gnmic -a leaf1 -u admin -p admin --skip-verify get --path /interface[name=ethernet-1/10]/oper-state/ --encoding json_ietf
```

# gNMI for Streaming Telemetry

```
nokia@rd-srx-ws1-155afda:~/n93-grpc$  
nokia@rd-srx-ws1-155afda:~/n93-grpc$  
nokia@rd-srx-ws1-155afda:~/n93-grpc$ gnmic -a leaf1 -u admin -p admin --skip-verify sub --path /interf  
ace[name=ethernet-1/10]/statistics/in-packets --mode stream --stream-mode on_change
```

# gNOI

- gRPC Network Operations Interface
- gRPC based service for executing operational commands
- Standards defined by OpenConfig - <https://github.com/openconfig/gnoi>



# List of current gNOI Services

## SYSTEM

Ping  
Traceroute  
Time  
SetPackage  
SwitchControlProcessor  
Reboot  
RebootStatus  
CancelReboot  
KillProcess

## FILE

Get  
TransferToRemote  
Put  
Stat  
Remove

## FACTORYRESET

Start

## HEALTHZ

Get  
List  
Acknowledge  
Artifact  
Check

## LINKQUALIFICATION

Create            Delete  
Get                List  
Capabilities

## OS

Install  
Activate  
Verify

# gNOI Use Cases



```
/Interface  
<bgp>  
<vrf>  
<oam>
```


Config  
backup/  
restore

R24 → R25


Software  
upgrade




Hardware  
health  
checks



Factory  
reset



Node  
reboot



Control  
processor  
switch

# gNOI Services - File

- gNOI File service – allows client to transfer files to and from the target node.
- gNOI File RPCs:
  - GET
  - PUT
  - REMOVE
  - STAT

# gNOI Client

- gNOIc – gNOI CLI client that provides support for select gNOI services
- <https://gnoic.kmrd.dev/>
- <https://gnxi.srlinux.dev/>

The screenshot shows the 'gNOI Protobuf Documentation' page for the 'file' service. The page title is 'File v0.1.0'. A search bar is present at the top. The left sidebar contains a navigation menu with 'SERVICES' selected, showing a list of services including 'File', 'GetRequest', 'GetResponse', 'PutRequest', 'PutRequest.Details', 'PutResponse', 'RemoveRequest', 'RemoveResponse', 'StatInfo', 'StatRequest', 'StatResponse', 'TransferToRemoteRequest', and 'TransferToRemoteResponse'. The main content area displays the 'File' service details, including a table of methods and their descriptions.

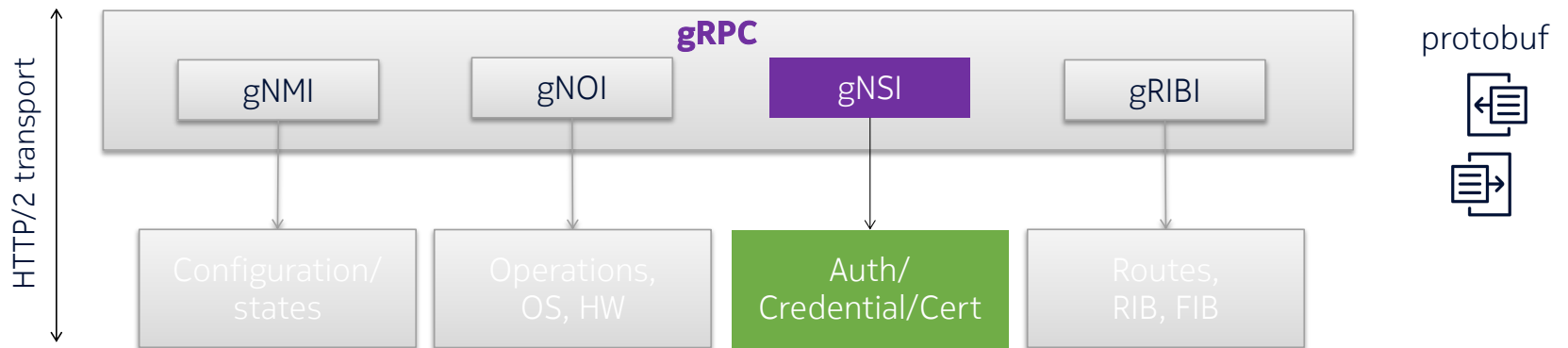
METHOD NAME	REQUEST / RESPONSE TYPE	DESCRIPTION
<b>Get</b>	<a href="#">GetRequest</a> <a href="#">GetResponse</a>	Get reads and streams the contents of a file from the target. The file is streamed by sequential messages, each containing up to 64KB of data. A final message is sent prior to closing the stream that contains the hash of the data sent. An error is returned if the file does not exist or there was an error reading the file.
<b>TransferToRemote</b>	<a href="#">TransferToRemoteRequest</a> <a href="#">TransferToRemoteResponse</a>	TransferToRemote transfers the contents of a file from the target to a specified remote location. The response contains the hash of the data transferred. An error is returned if the file does not exist, the file transfer fails, or if there was an error reading the file. This is a blocking call until the file transfer is complete.
<b>Put</b>	<a href="#">PutRequest</a> stream <a href="#">PutResponse</a> stream	Put streams data into a file on the target. The file is sent in sequential messages, each message containing up to 64KB of data. A final message must be sent that includes the hash of the data sent. An error is returned if the location does not exist or there is an error writing the data. If no checksum is received, the target must assume the operation is incomplete and remove the partially transmitted file. The target should initially write the file to a temporary location so a failure does not destroy the original file.
<b>Stat</b>	<a href="#">StatRequest</a> <a href="#">StatResponse</a>	Stat returns metadata about a file on the target. An error is returned if the file does not exist or there is an error in accessing the metadata.
<b>Remove</b>	<a href="#">RemoveRequest</a> <a href="#">RemoveResponse</a>	Remove removes the specified file from the target. An error is returned if the file does not exist, is a directory, or the remove operation encounters an error (e.g., permission denied).

# CONFIG BACKUP

```
nokia@rd-srx-ws1-155afda:~/n93-grpc$  
nokia@rd-srx-ws1-155afda:~/n93-grpc$ gnoic -a leaf1 -u admin -p admin --skip-verify file stat --path /etc/opt/  
srlinux/config.json
```

# gNSI

- gRPC Network Security Interface
- gRPC based service for security configuration
- Standards defined by OpenConfig - <https://github.com/openconfig/gnsi>

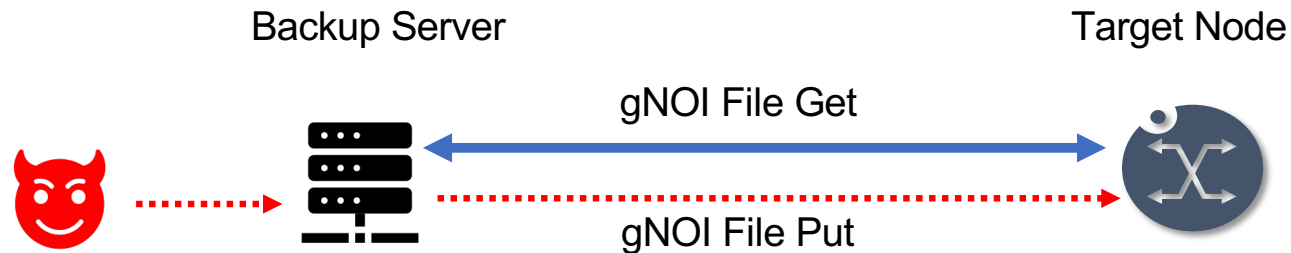


# gNSI Services

- Certz – allows a client to replace a certificate, trust bundle, CRL or a combination of these on a target node.
- Acctz – allows clients to stream accounting records from a target node.
- Pathz – allows access control for specific gNMI paths

# gNSI Services

- Authz - gRPC-level authorization policy that defines which users can access which gRPC services/RPCs on the target node



# gNSI Usecases



gRPC  
Access  
Control

The icon features a black devil face on the left, a white arrow pointing right in the center, and a red prohibition sign on the right.




TLS  
Certificate  
Mgmt

The icon shows a black document with a gear and a checkmark inside a square frame.



Accounting  
records

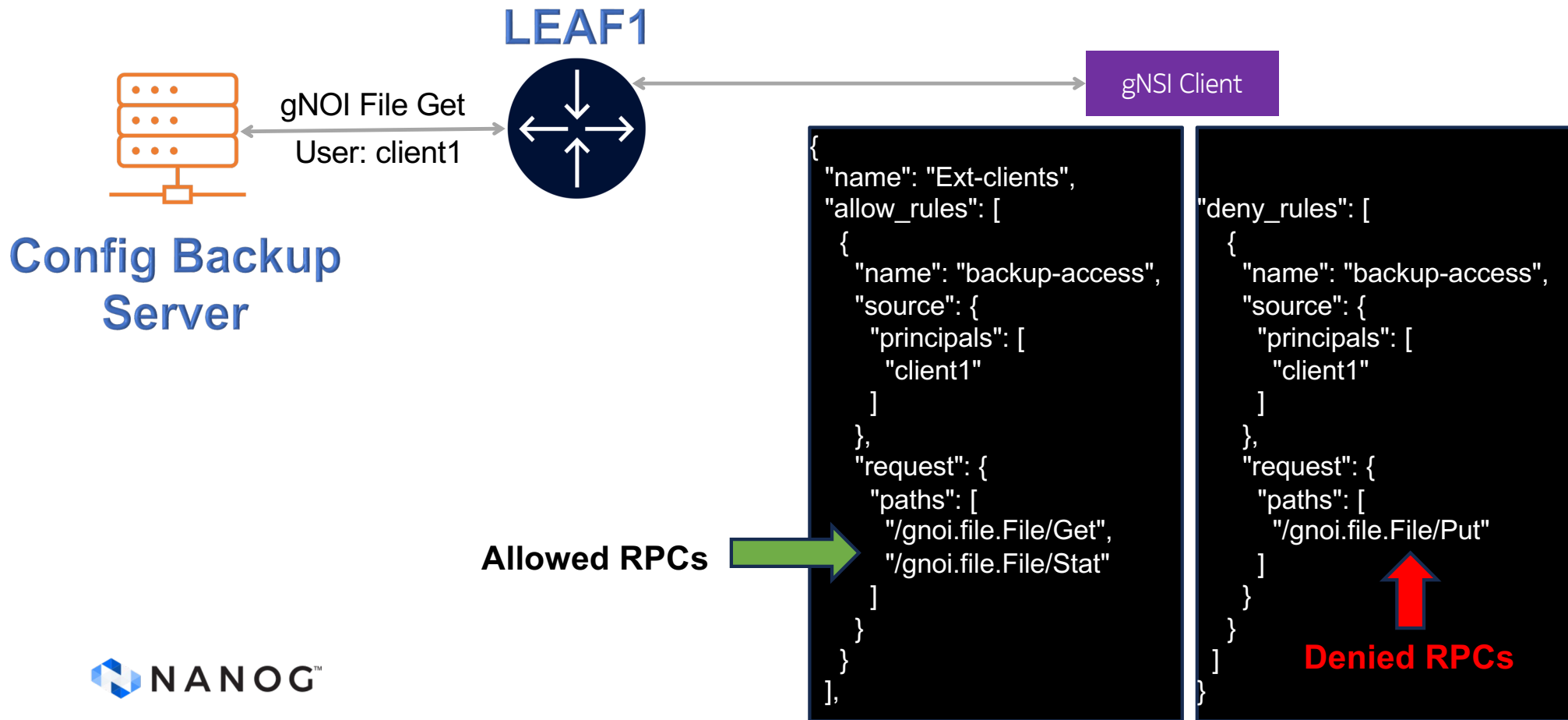
The icon depicts a green smiley face wearing sunglasses on the left, a white arrow pointing right in the center, and an orange-bordered list with checkmarks on the right.



Secure  
boot

The icon includes a thumbs-up gesture, a server rack, and a document with a gear icon.

# gNSI Authz Demo Setup

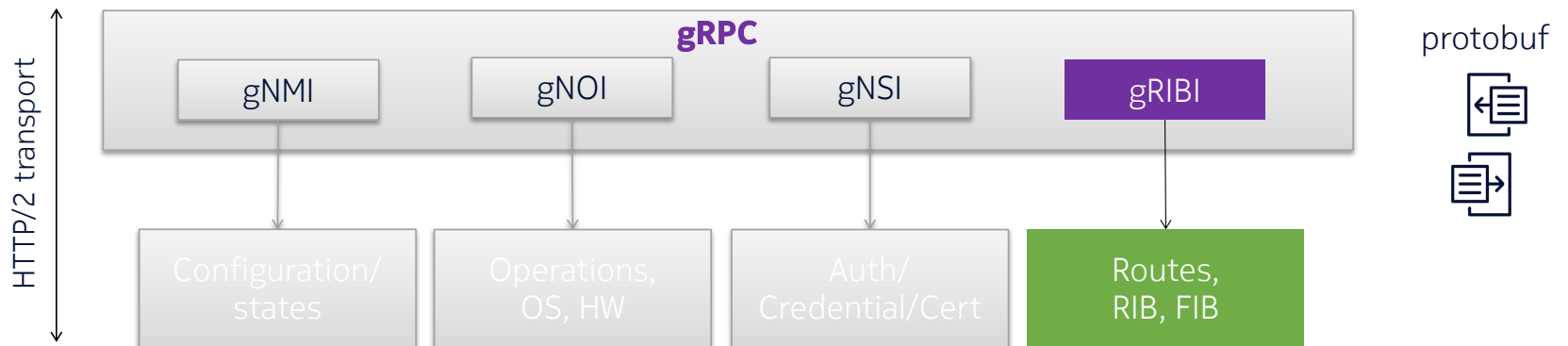


# gNSI Authz Demo

```
nokia@rd-srx-ws1-155afda:~/n93-grpc$ gnoic -a leaf1 -u client1 -p client1 --skip-verify file stat --path /etc/opt/sr/linux/config.json
```

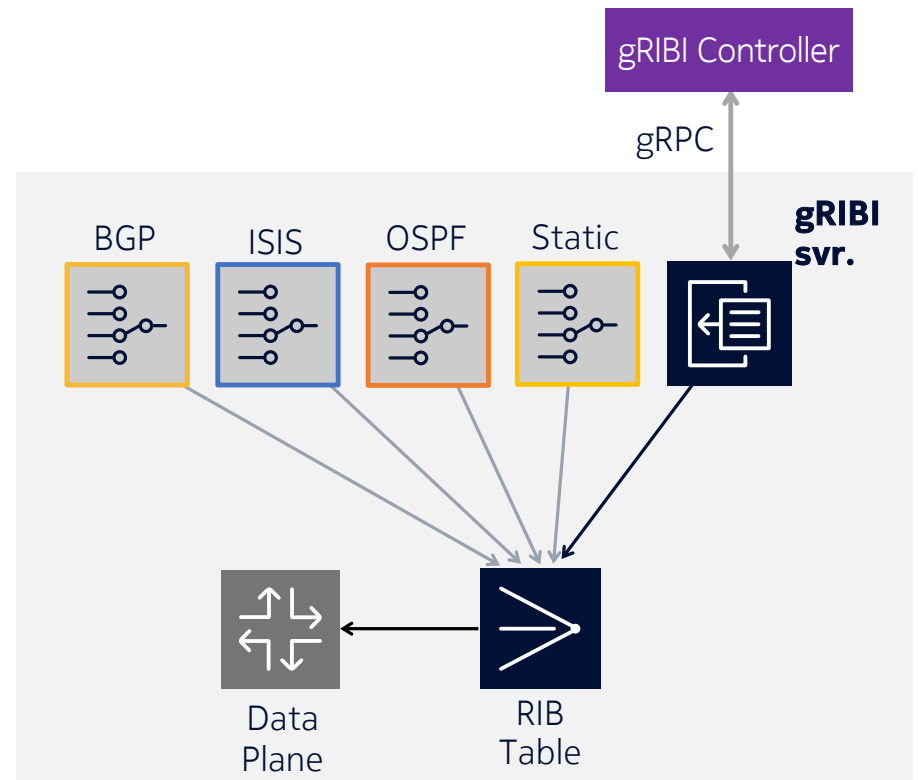
# gRIBI

- gRPC **R**outing **I**nformation **B**ase **I**nterface
- gRPC based service for route injection
- Standards defined by OpenConfig - <https://github.com/openconfig/gribi>



# gRIBI Overview

- Entries are created as though they are learned through a dynamic routing protocol
- Multiple gRIBI clients can connect to the network device (from all clients or a primary client only)
- Supported RPC
  - **Modify** RPC (ADD, DELETE, and REPLACE AFT Operations)
  - **Get** RPC
  - **Flush** RPC



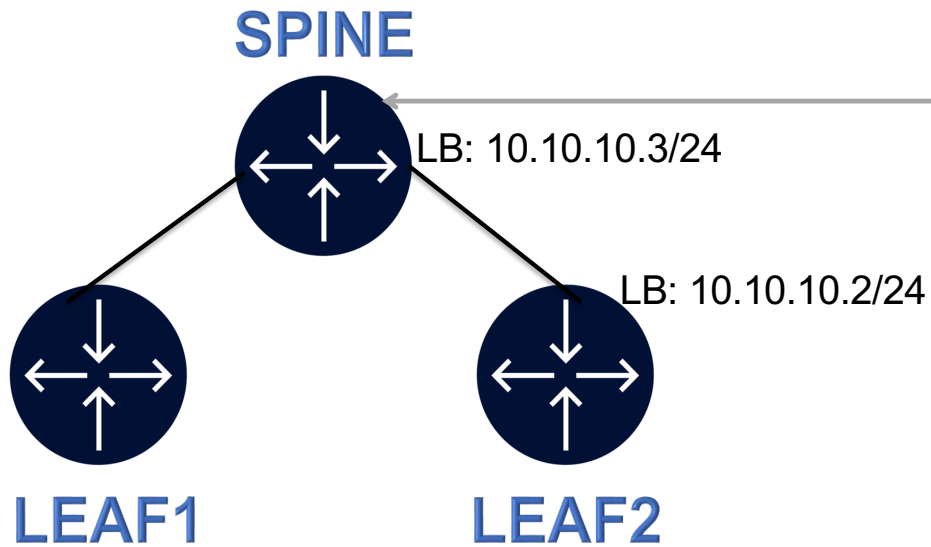
# gRIBI Usecases

- SDN-based network control environment
- Emergency route update
- Peering/Traffic Engineering
- On-demand path creation

# gRIBI Client

- gribic – gRIBI CLI client that provides support for gRIBI services
- <https://gribic.kmrd.dev/>

# gRIBI Demo Topology



gRIBI Controller

```
operations:  
- op: add  
  nh:  
    index: 1  
    ip-address: 192.168.20.2  
  
- op: add  
  nhg:  
    id: 1  
    next-hop:  
      - index: 1  
  
- op: add  
  ipv4:  
    prefix: 10.10.10.2/32  
    nhg: 1
```

# gRIBI Demo

```
✓ spine x ✓ bash
--{ + running }--[ ]--
A:spine#
--{ + running }--[ ]--
A:spine# ping 10.10.10.2 network-instance default -c 1 -w 1

Current mode: + running admin (68) Mon 10:48PM
```

# Leveraging what we learned today

gNOI System Ping

1. Use `gRPC` to ping remote loopback IP (10.10.10.2)
2. Use `gRPC` to check Leaf2 route table entries → `gNMI get`
3. Use `gRPC` to configure gribic user, AAA rules
4. Use `gribic` to inject route on Spine → `gNSI Authz Rotate`
5. Use `gRPC` to check Leaf2 route table entries
6. Use `gRPC` to ping remote loopback IP (10.10.10.2)

# gRIBI Demo

```
nokia@rd-srx-ws1-155afda:~/n93-grpc$  
nokia@rd-srx-ws1-155afda:~/n93-grpc$ gnoic -a spine -u admin -p admin --skip-verify system ping --dest  
ination 10.10.10.2 --ns default --count 1 --wait 1s
```

# Future gRPC Services

- gNPSI – gRPC Network Packet Streaming Interface
- gSII – gRPC State Injection Interface

Source: <https://github.com/openconfig>

# Summary

- Modern tools for network management
- Openconfig standards-based interfaces
- Vendor neutral RPCs
- Secure communication using TLS



**Thank you**