Automation and programmability of network services at the edge

SDN Meetup
Barcelona, 25th June 2019

Josep Lluis Ferrer <jl@voltanet.io>

Agenda

- Motivation for network automation
- State of the art of network device programmability
- Adding value to networks: Do we have the right tools?

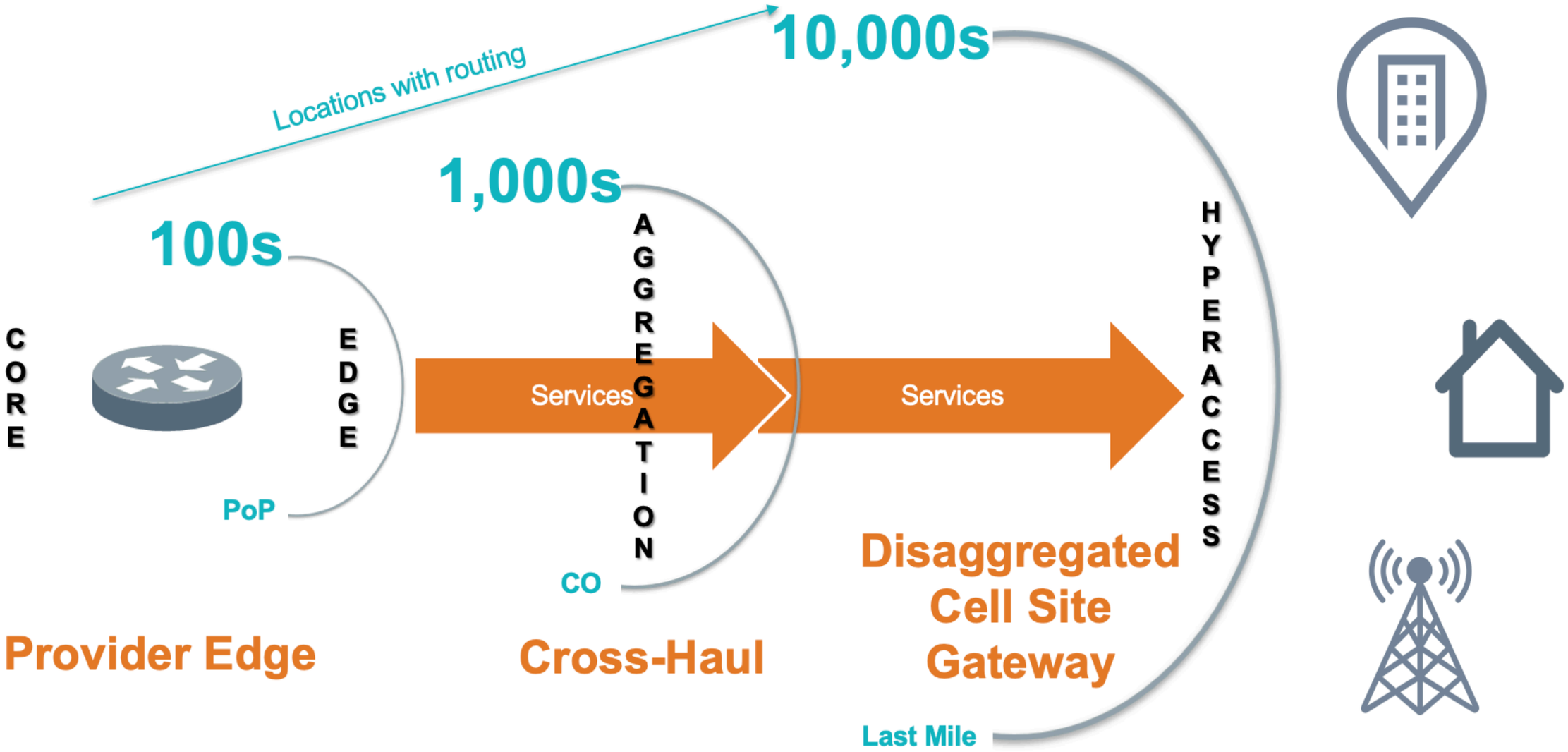
Motivation for network automation

Traditional network management

Custom | Complex | Expensive | Closed (non extensible)

- Networks are simply considered as collection of switches and routers.
- This practice leads to more complexity, due to increased number of systems that must be managed directly. Non-scalable.
- Lack of programmability, forced to think in terms of device configuration.
- Lack of agility on delivering new features. Locked to equipment provider release cycles.
- Tools and protocols: CLI, SSH, SNMP, custom tools.

The Edge: New service edge changes routing



VOLTA Networks

Network automation is a methodology in which software automatically configures, provisions, manages and tests network devices

Network automation is a methodology in which software automatically configures, provisions, manages and tests network devices *(with minimal or no human intervention).*

Network automation is a methodology in which software automatically configures, provisions, manages and tests network devices *(with minimal or no human intervention).*

It can be achieved by means of **programmatic** capabilities, i.e., via **APIs** which define the supported remote calls.

API based management approach

Programmable | Data-driven | Automated | Open

- All network elements could be configured using the same tools and abstractions
- Software based workflow to increase agility
- Network operator could focus on build new services
- Network engineers could build custom services with common set of tools
- Tools and protocols: YANG, NETCONF, Openconfig, gRPC, vendor SDKs, open source libraries

State of the art of network device automation

API Definition Requirements

- **Data model**: Define the data consumed by the methods (e.g. YANG)

- **Operations**: Define the operations that can be performed via API.

- **Serialization**: The encoding, how data is sent over the wire (examples JSON, XML, protobuf)

- **Transport**: The underlaying protocol to consume the API calls. E.g., HTTP, HTTP/2, QUIC, SSH…

Data model

- YANG (IETF) adopted as main data-model for networking devices, providing both configuration and operational state (including statistics)

- Defines data hierarchy as tree structure

- Specifies data types, restrictions (read, read+write), valid values, defaults, …

- Can be converted to any encoding format: JSON, XML

- **Open** models (vendor neutral): IETF, Openconfig

- Vendor models

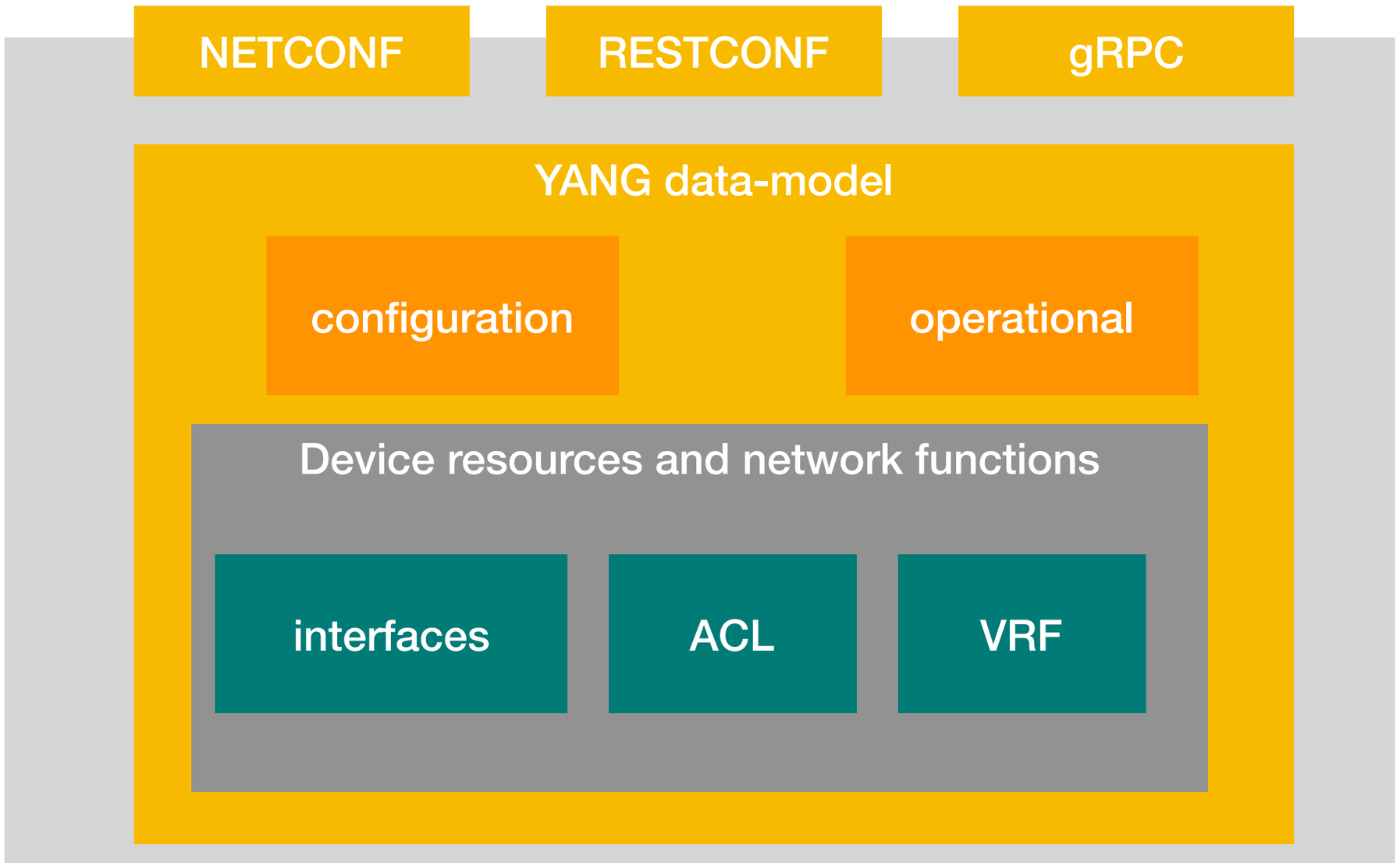
Transport protocols

- **NETCONF (SSH)**
  - RPCs (XML/JSON): GET-CONFIG, EDIT-CONFIG, COMMIT,…
- **RESTCONF (HTTP/S)**
  - RPCs (XML/JSON): GET, POST, DELETE, PUT
- **gRPC (HTTP/2)**
  - RPC: Req/Rsp, streaming, bidirectional, …
  - —> De-facto standard for telemetry

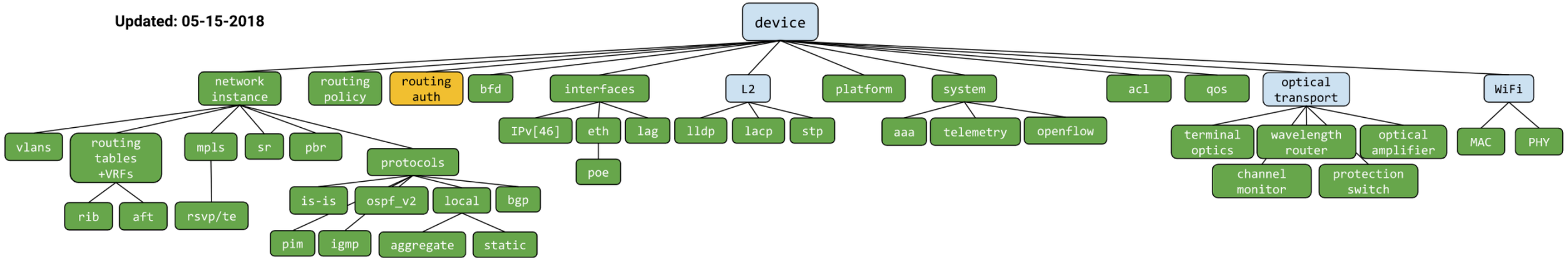
HTTP/2

- **Request/Response multiplexing**
- **Bidirectional streams**
- **Binary framing**
- **Streams priorization**

Network device configuration and management APIs



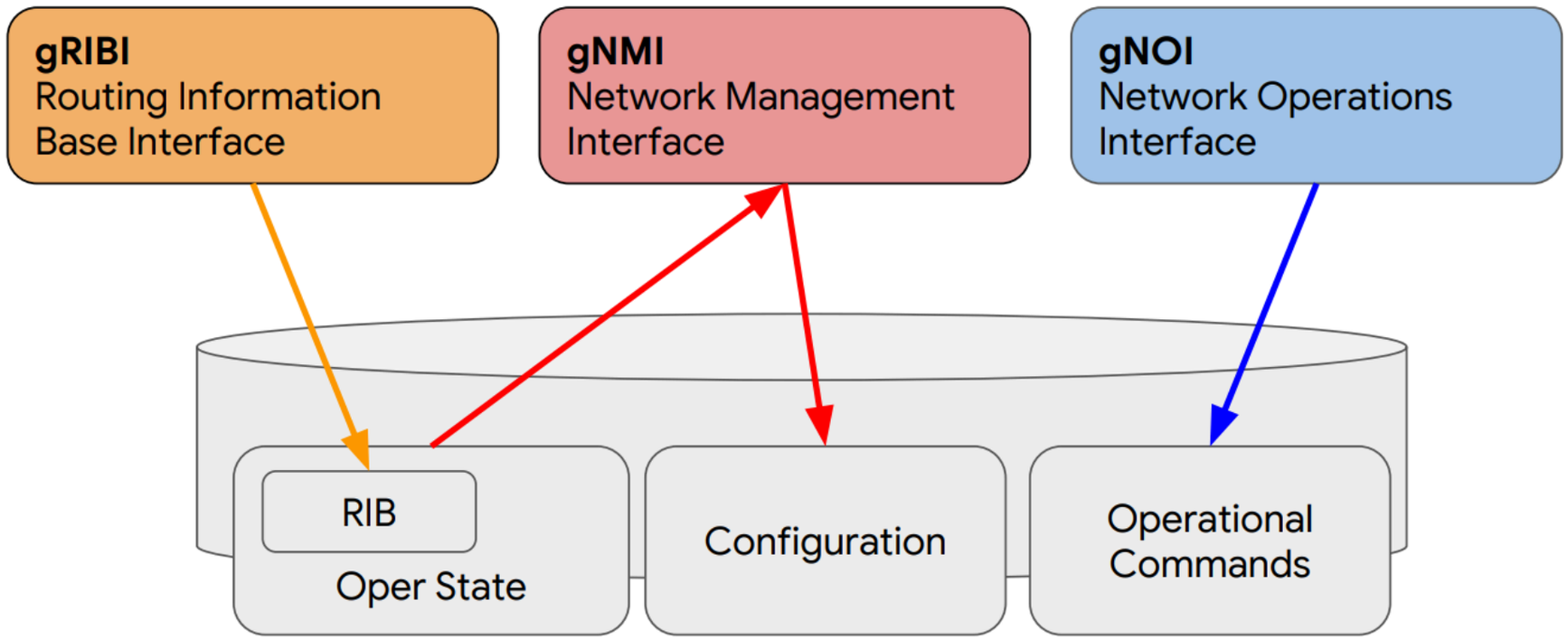
OpenConfig: Device Models*



*http://www.openconfig.net/projects/models/

VOLTA Networks

OpenConfig: gRPC interfaces



gNMI Figure source: https://datatracker.ietf.org/meeting/101/materials/slides-101-rtgwg-sessa-grpc-services-on-network-devices-00

gNMI

```
service gNMI {

  rpc Capabilities(CapabilityRequest) returns
(CapabilityResponse);

  rpc Get(GetRequest) returns (GetResponse);

  rpc Set(SetRequest) returns (SetResponse);

  rpc Subscribe(stream SubscribeRequest) returns
(stream SubscribeResponse);

}
```

gNMI Telemetry

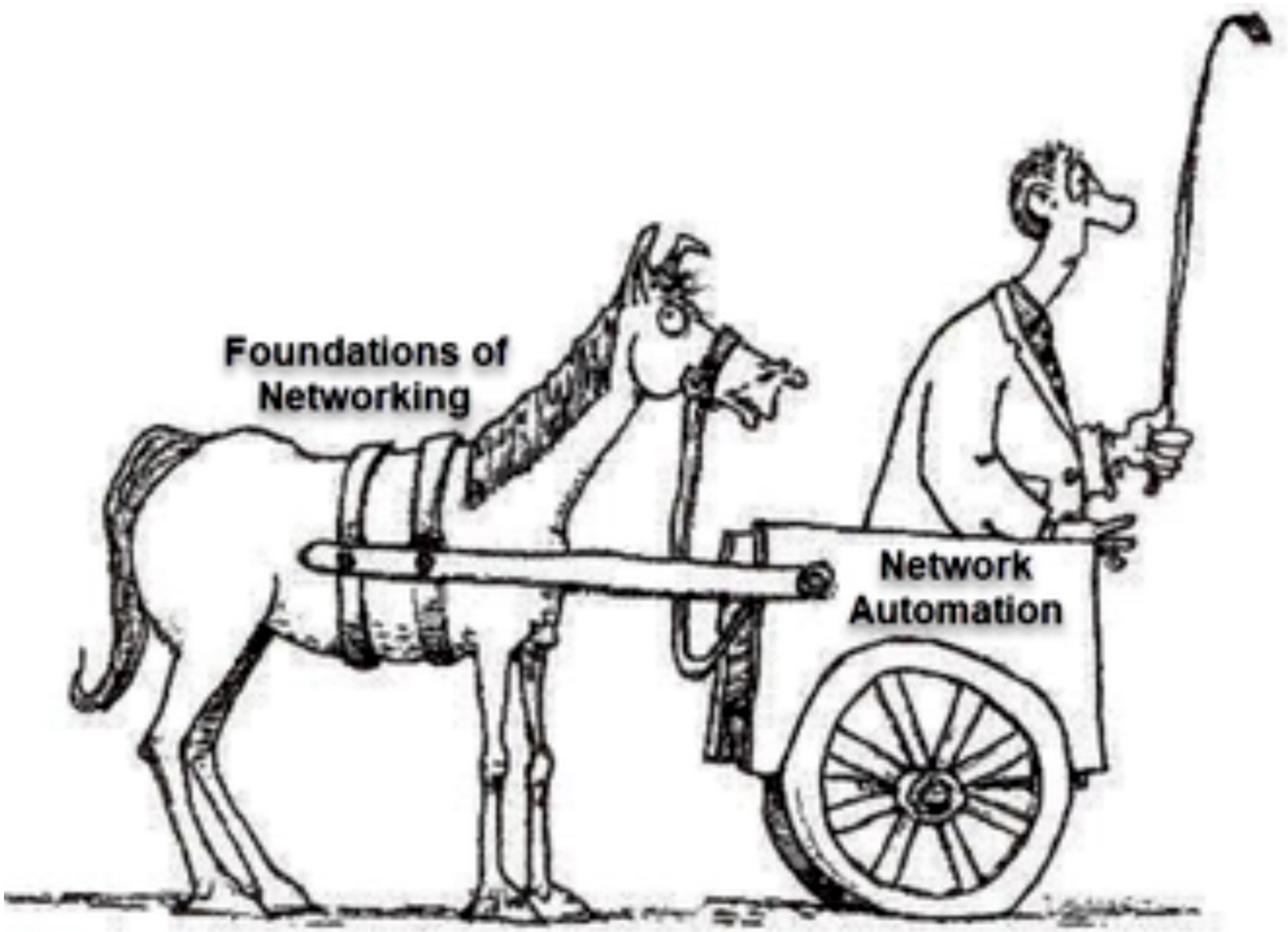
```
message Subscription {
  Path path = 1;
  SubscriptionMode mode = 2;
  uint64 sample_interval = 3;
  bool suppress_redundant = 4;
  uint64 heartbeat_interval = 5;
}

message Path {
  repeated string element = 1
[deprecated=true];
  string origin = 2;
  repeated PathElem elem = 3;
  string target = 4;
}


message PathElem {
  string name = 1;
  map<string, string> key = 2;
}
```

- SubscribeRequest message allows multipe subscriptions via SubscriptionList message.
- Each SubscriptionList includes multiple Subscription messages
- Modes
  - STREAM: Sends value on change
  - ONCE: Only sends 1 update
  - POLL: Actively poll for the value
- Path and PathElem represent serialization of XPATHs telemetry clients can be subscribed —> XPATH is text based
- gNMI encoding (TypedValue):
  - JSON
  - BYTES
  - PROTO
  - ASCII
  - JSON_IETF
  - Native (int, bool,…)

Adding value to networks: Do we have the right tools?



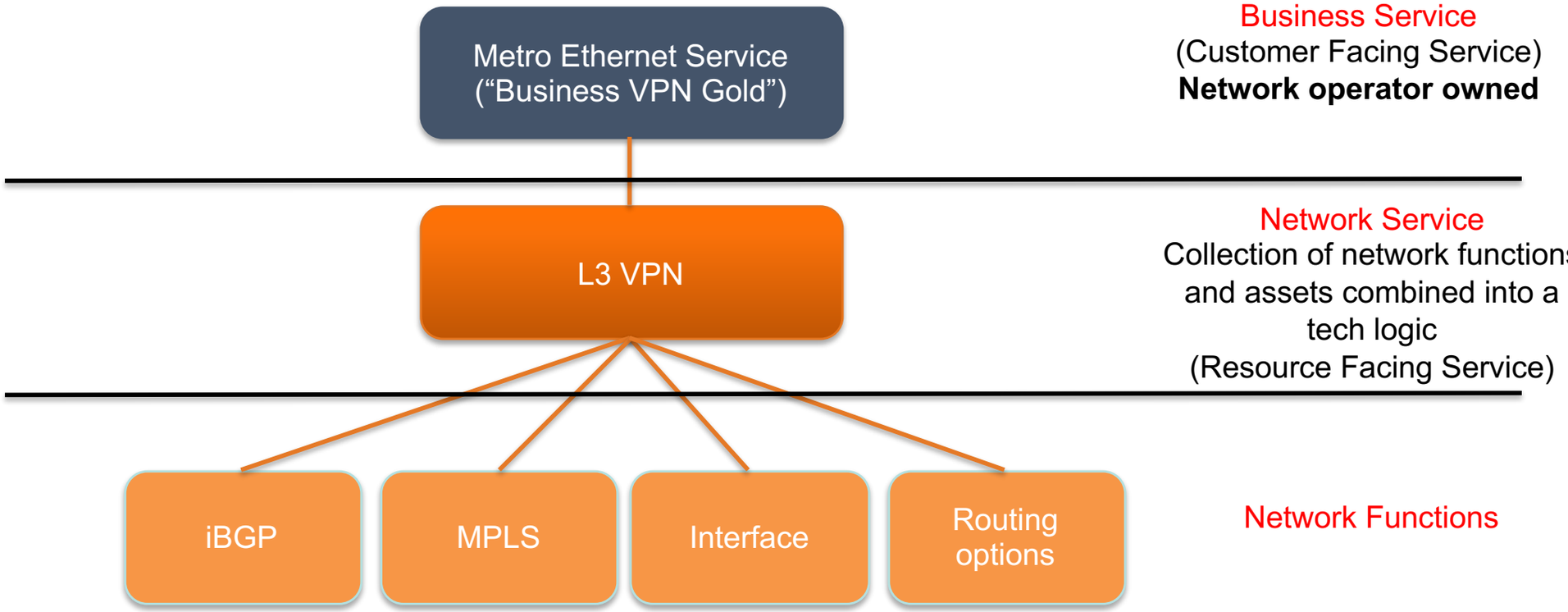
Source: Twitter

VOLTA Networks

Network Service is a collection of network functions and device resources combined into a business and/or technology logic distributed among different network elements.

Network function describes the configuration parameters of a specific device technology or feature and exposes via API (ACLs, routing protocols, policies,…)

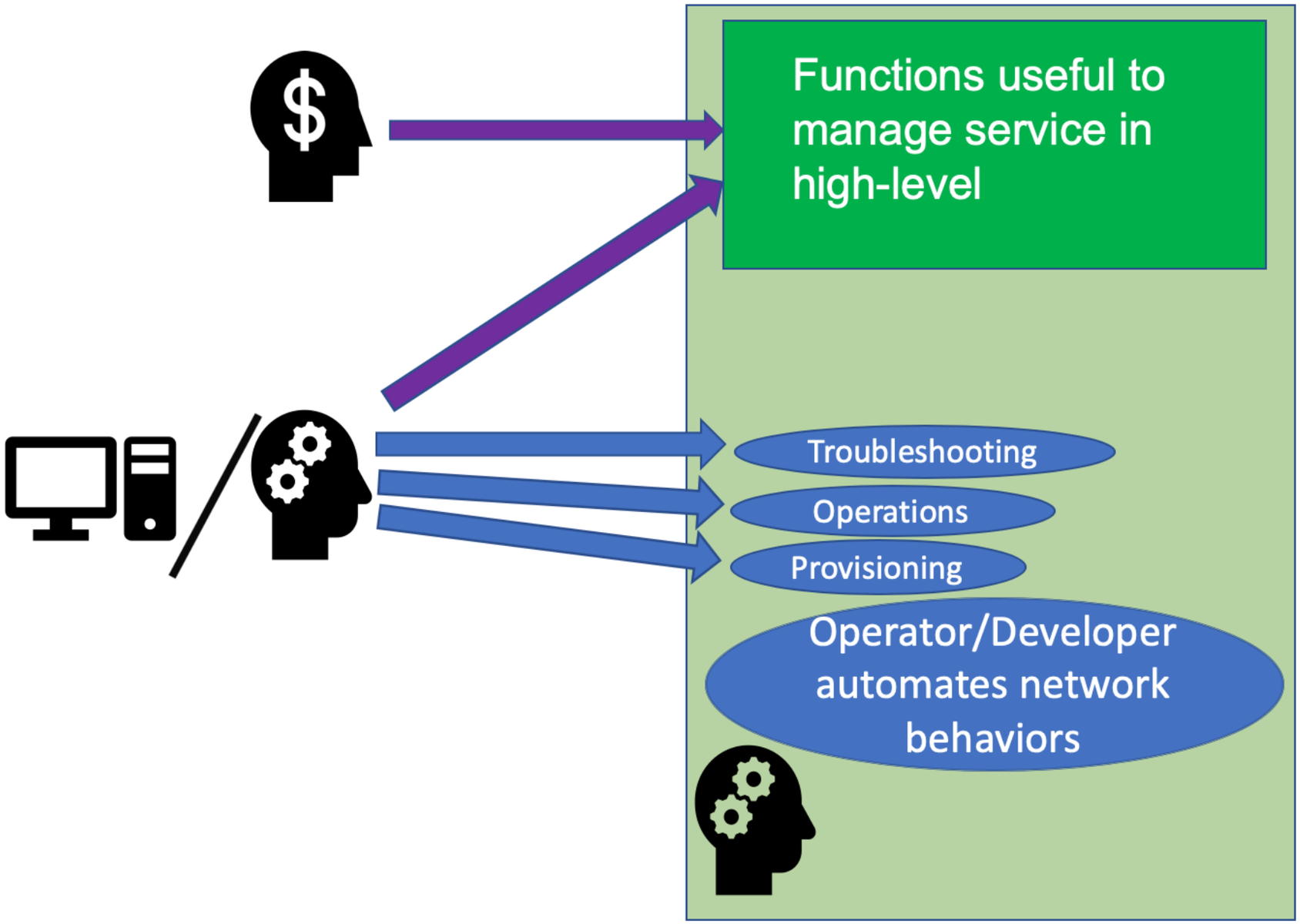
Network Service Example



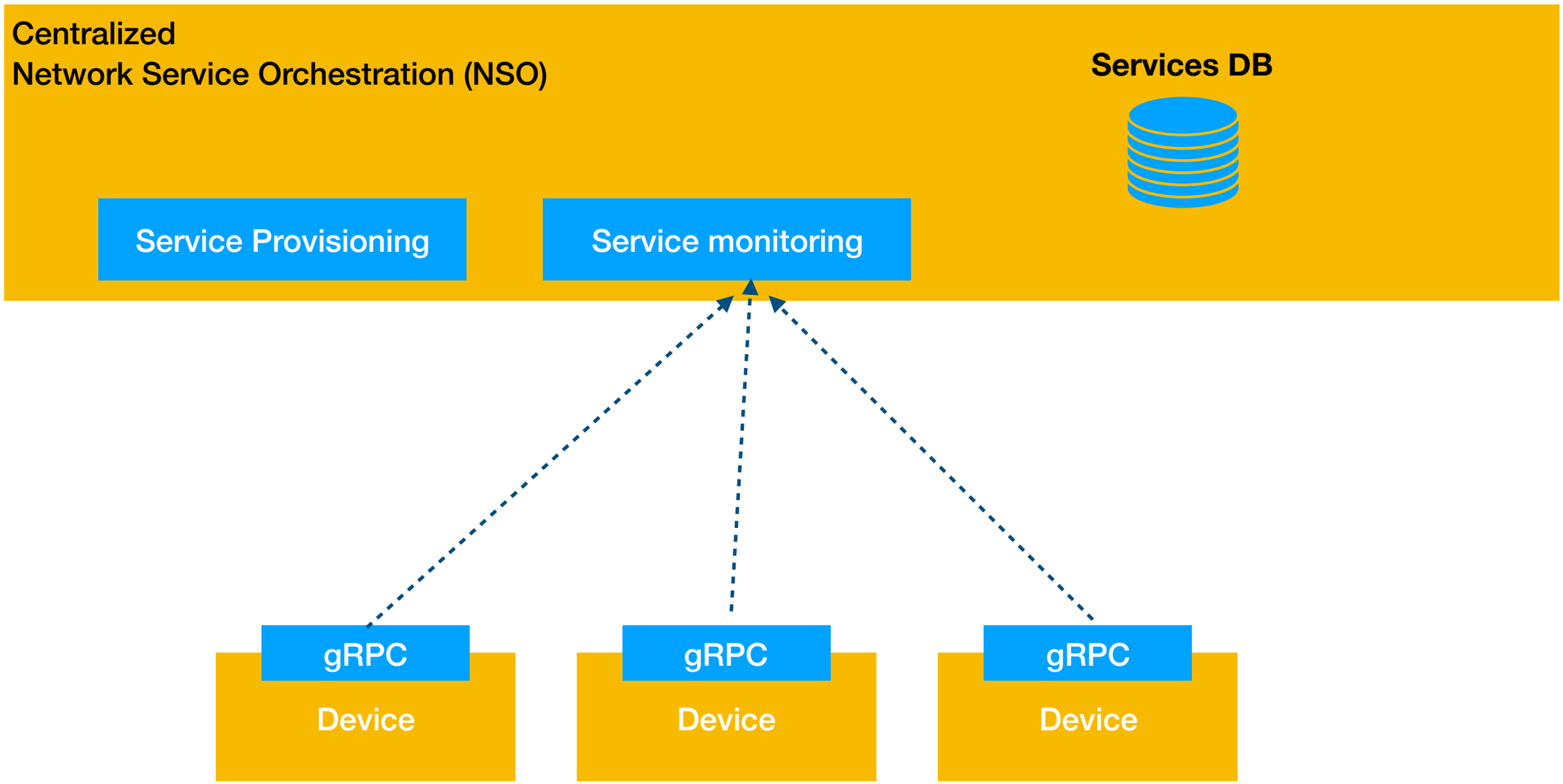
Services models

- IETF also defines service level YANG models
  - L2VPN (RFC8466) and L3VPN (RFC8299)
- Openconfig only defines models at device level
- Network operators and architects still have to create their own tools to create and manage services (and create value!!)
- **Could we use any re-usable pattern to design and automate networking services?**

Nework service automation

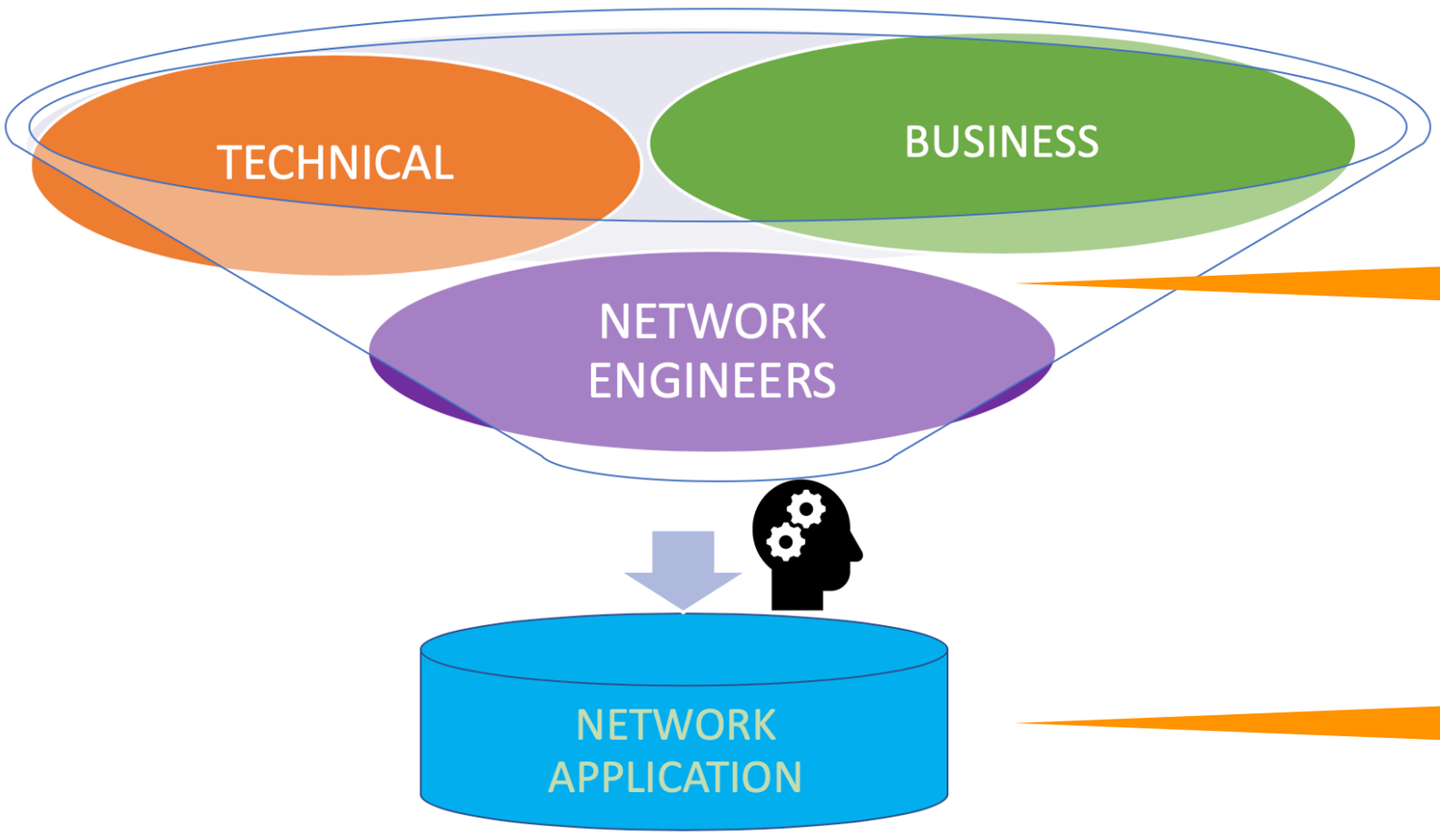


Traditional approach: Network Service Orchestration (NSO)



VOLTA Networks

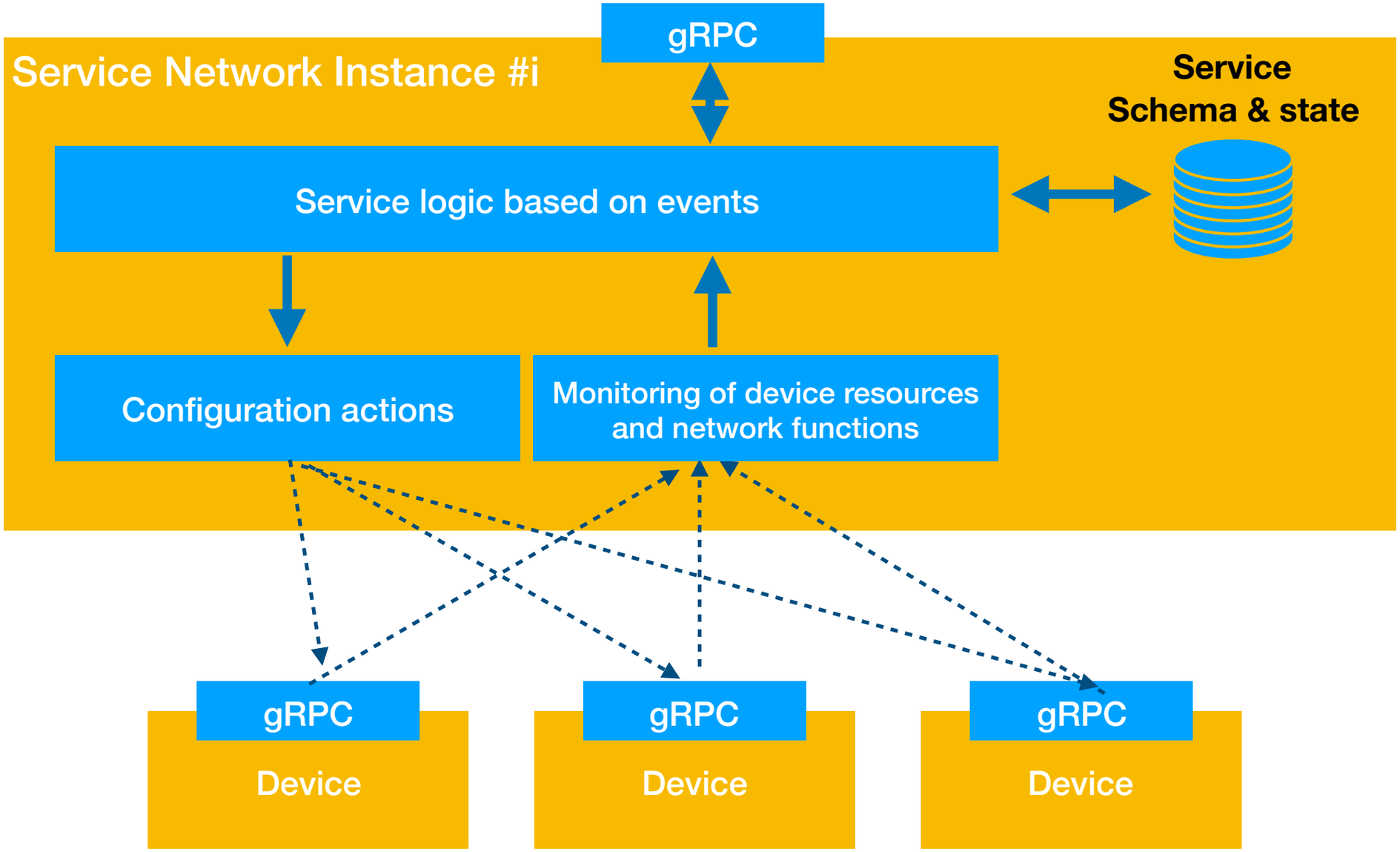
Network Service as SDN network Application



Foundations of networking

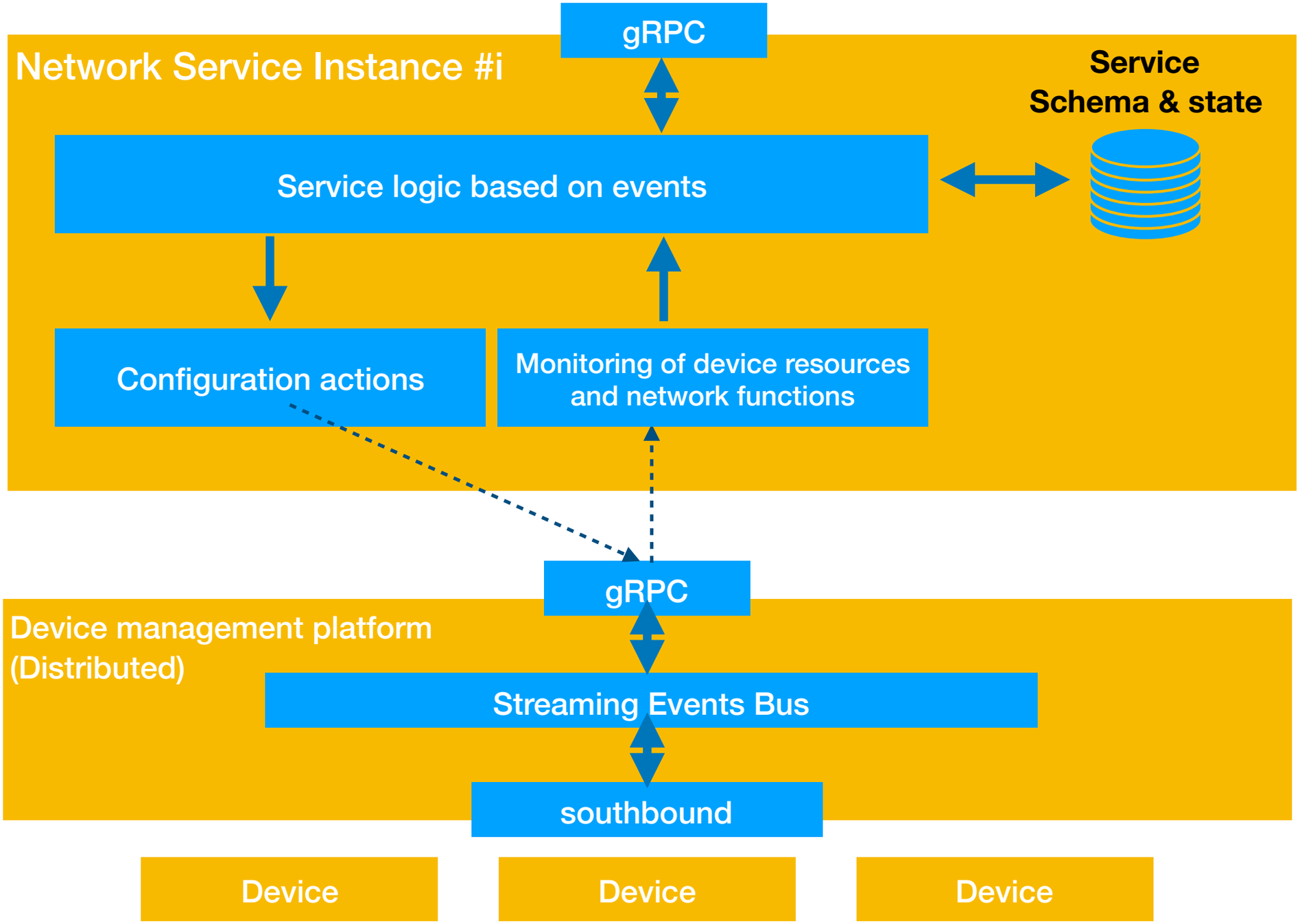
Network automation

Network Service as SDN network Application (II)



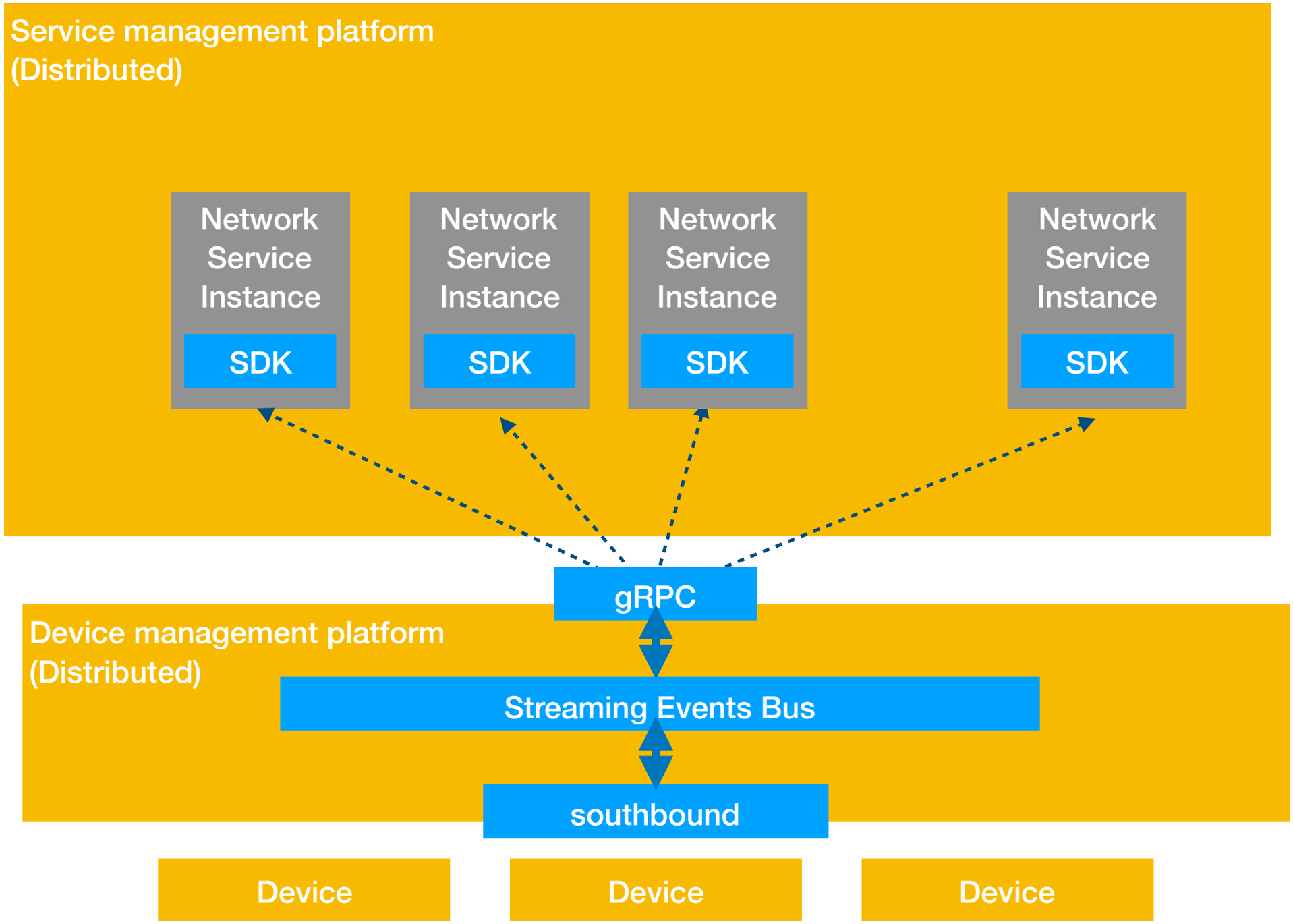
VOLTA Networks

Network Service as SDN network Application (III)



VOLTA Networks

Network Service as SDN network Application (IV)



VOLTA Networks

Questions ?

Do you like networking, automation and programmability?

We are hiring:

https://voltanet.io/careers/

careers@voltanet.io