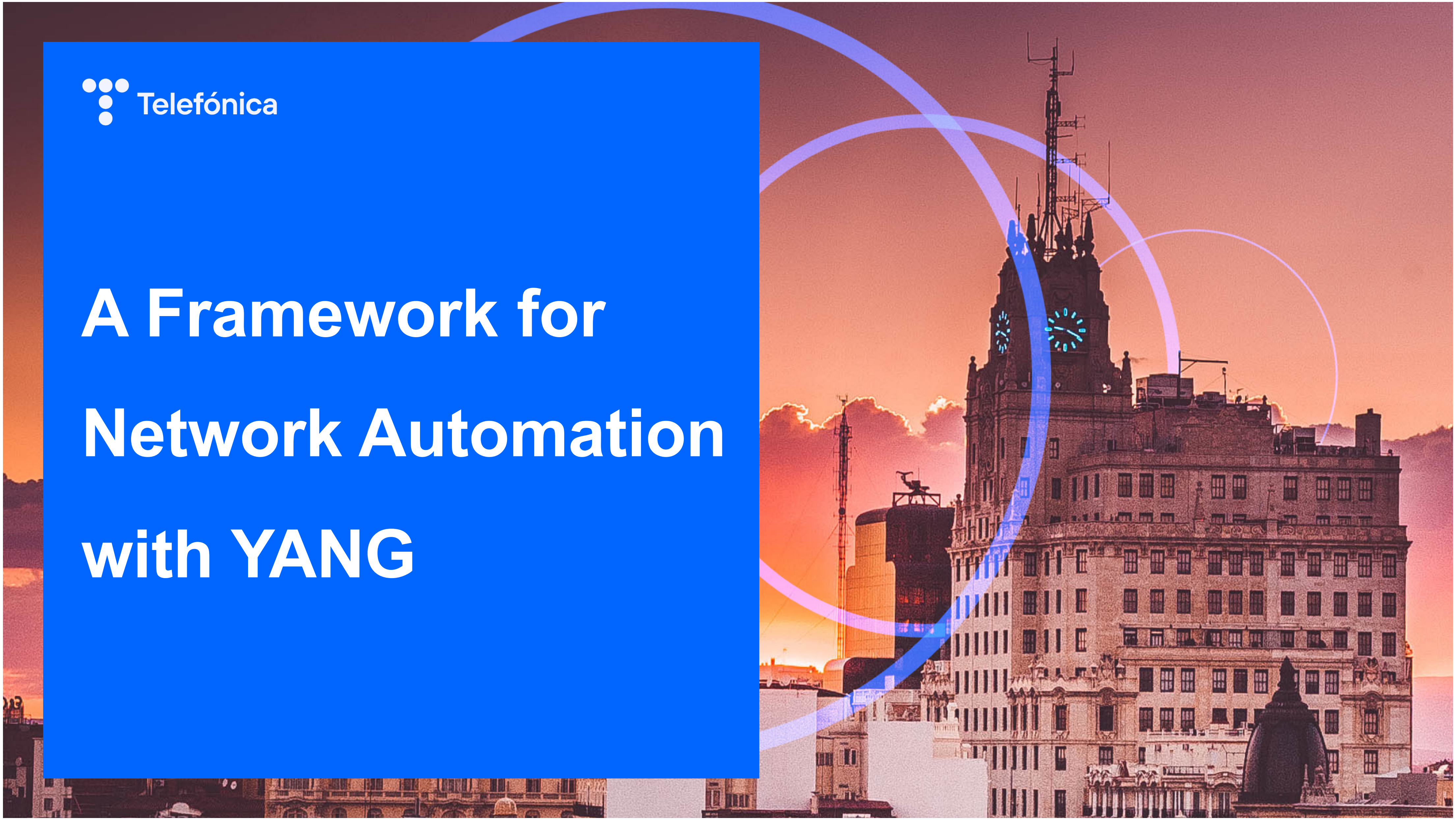
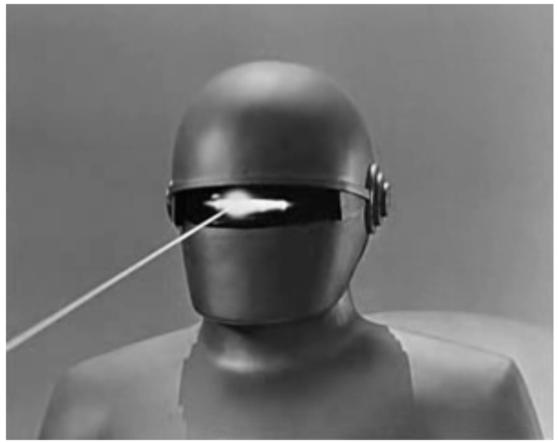
Telefónica

A Framework for Network Automation with YANG



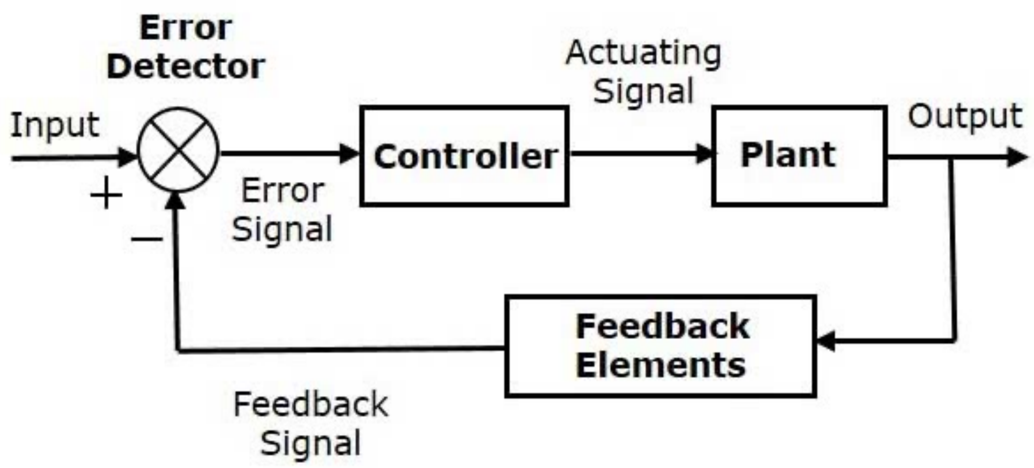
The Automation Aspiration

- Fully autonomous networks
    - After the automatic and autonomic stages
- Based on a definition of goals and policies
    - Network intents
    - Policy enforcement and propagation
- Detection and action
    - Dynamic identification
    - Adaptable response
- Well-known closed-loop approaches
    - Completed with AI and software network techniques
    - Requiring proper modeling



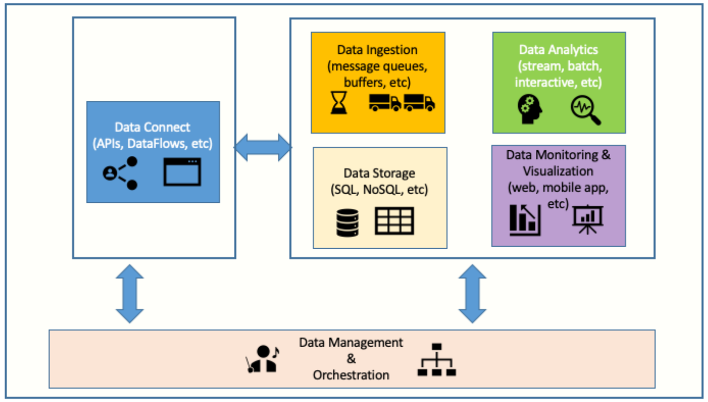
Automatics 101

- Automatics have been around for a long time
  - AI as a tool to derive further insights from data and improve policies
  - Extended capabilities, but we should not expect Skynet
- Beware the network differential facts
  - Topology (and geometry!) awareness
  - The conservation principle
  - Openness
  - Integrity and auditability
  - Isolation

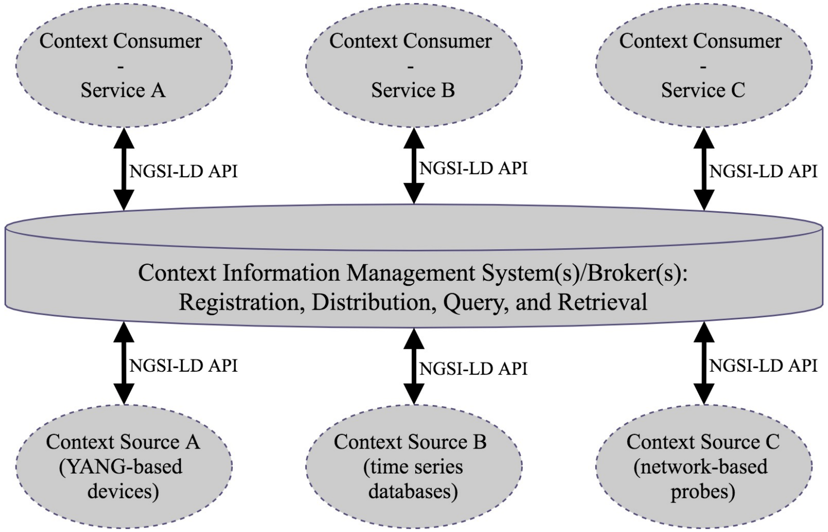


- Model obliviousness
- Flows and architectures
- Software network abstractions as essential enablers

The Data Flow

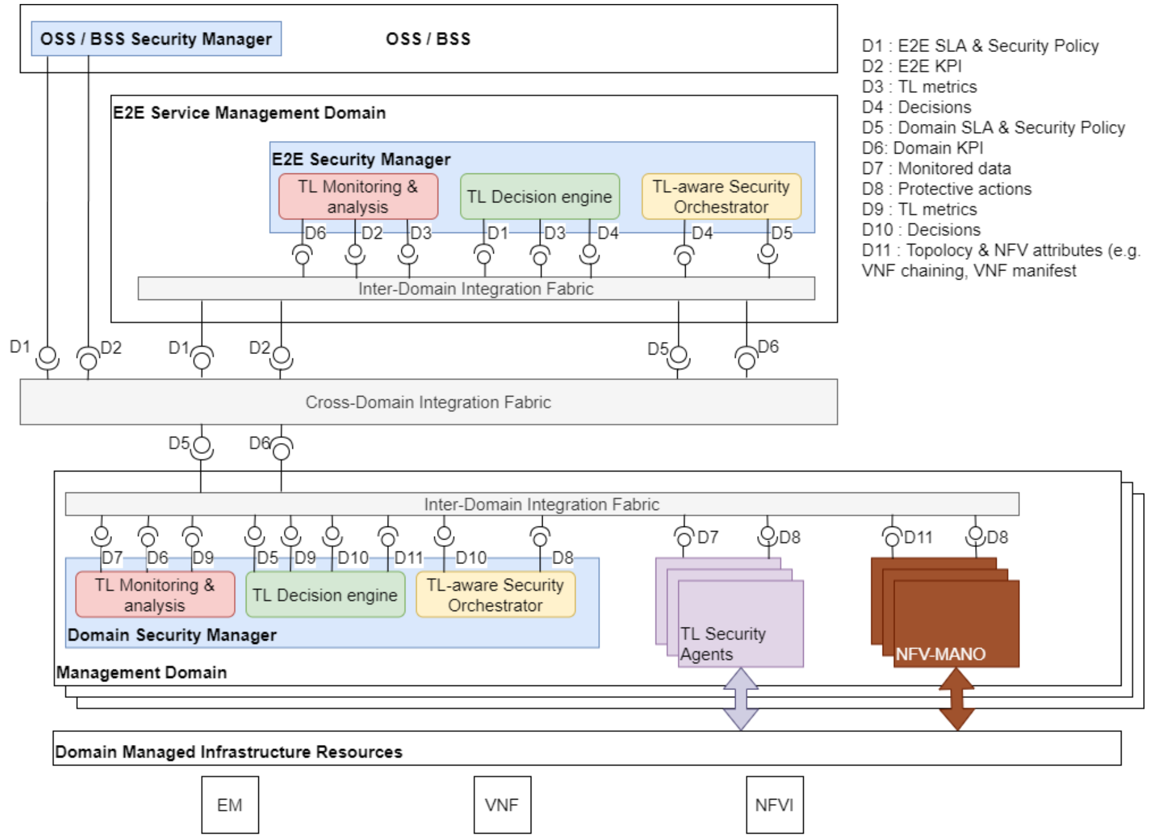


- No matter how intelligent: Crap in means crap out
  - Usable: Adaptation (formats, scales…)
  - Sufficient: Topology (sources, aggregators…)
  - Safe: Provenance (origin, timestamps…)
  - Steady: Continuity (pace, availability…)
- Deal with heterogeneity
  - At all levels: sources, consumers, models, deployment styles, supporting infrastructures
- An enhanced data fabric seems the logical approach
  - Metadata becomes essential
  - Including semantic mappings
  - A data stream ontology, not that far away



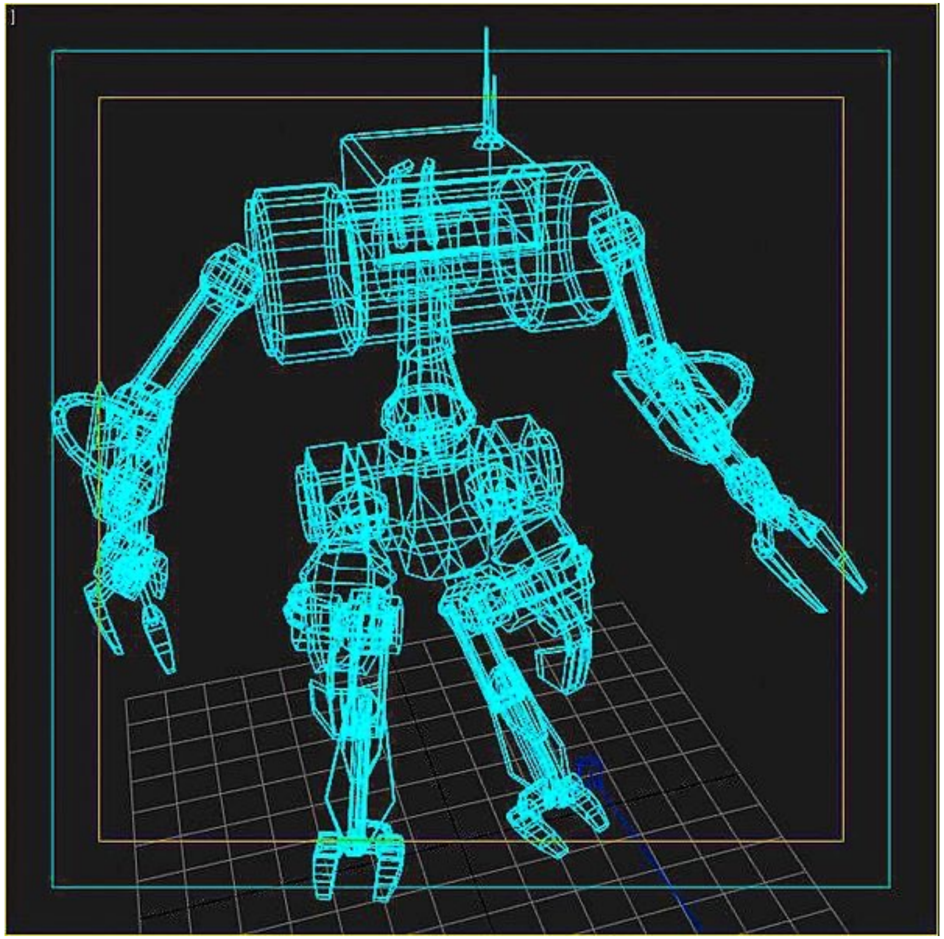
The Action Flow

- Heterogeneity at its best
    - OAM actions at a wide variety of different domains
- Initial strategies in network management
    - Domain specific
    - Recommendation systems
    - Autonomic protocols
- SBA approaches and capability models
    - Abstractions of element functionalities
    - Usable as building blocks
    - Compositional models
    - Inter-domain collaboration for E2E
    - Registration and discovery mechanisms

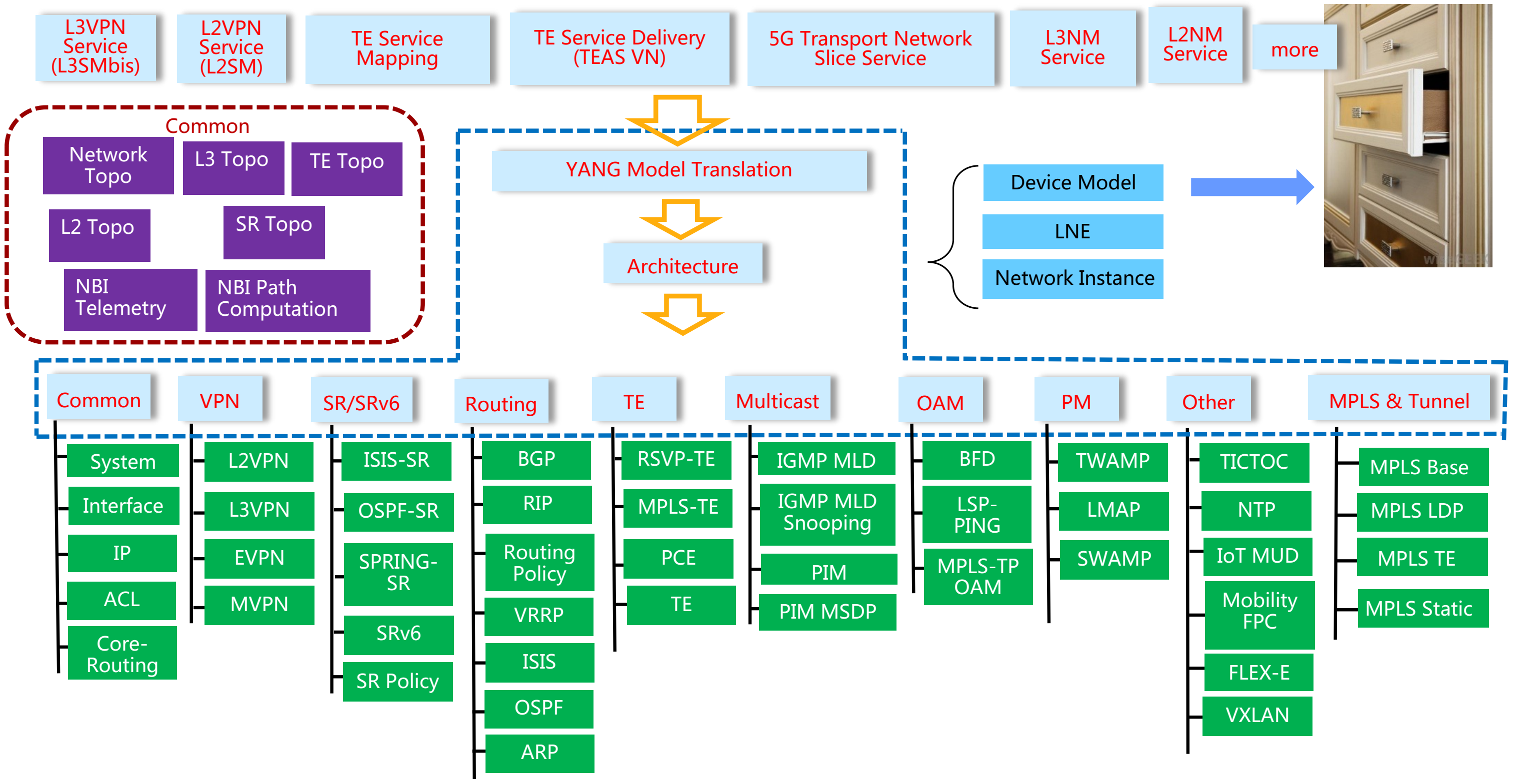


Modeling at the Roots

- Use the model to describe data flows
  - Sources
  - Consumers
  - Elements in the flow
- In both directions
  - Despite the current asymmetry
- And including
  - The identification of the relationships to the flow data model
  - Provenance metadata
  - Security
  - Support for federation mechanisms
- Multidimensional multidomain



A Landscape of Models (just in IETF)



- Integration
  - Management targets
  - Abstractions
  - Layers
  - Protocols
- Composition
  - Sources
  - Consumers
  - Control loops
  - Models themselves
- The need for *metamodels*

A Layered Metamodel for Automation – RFC 8969

```
+-----------------------------------------------------------------+
|                                    Hierarchy Abstraction        |
|                                                                 |
| +-----------------------+              Service Model            |
| |     Orchestrator      |           (Customer Oriented)         |
| |+---------------------+|         Scope: "1:1" Pipe model       |
| ||  Service Modeling   ||                                       |
| |+---------------------+|                                       |
| |                       |              Bidirectional            |
| |+---------------------+|         +-+  Capacity, OWD +-+        |
| ||Service Orchestration||         | +----------------+ |        |
| |+---------------------+|         +-+                +-+        |
| +-----------------------+       Ingress              Egress     |
|                                                                 |
|                                                                 |
| +-----------------------+            Network Model              |
| |   Controller          |         (Operator Oriented)           |
| |+---------------------+|      +-+    +--+    +---+   +-+       |
| || Network Modeling    ||      | |    |  |    |   |   | |       |
| |+---------------------+|      | o----o--o----o---o---o |       |
| |                       |      +-+    +--+    +---+   +-+       |
| |+---------------------+|      src                    dst       |
| ||Network Orchestration||           L3VPN over TE               |
| |+---------------------+|    Instance Name/Access Interface     |
| +-----------------------+  Protocol Type/Capacity/RD/RT/...     |
|                                                                 |
|                                                                 |
| +-----------------------+             Device Model              |
| |     Device            |                                       |
| |+-------------------+  |                                       |
| || Device Modeling   |  |      Interface add, BGP Peer,         |
| |+-------------------+  |        Tunnel ID, QoS/TE, ...         |
| +-----------------------+                                       |
+-----------------------------------------------------------------+
```

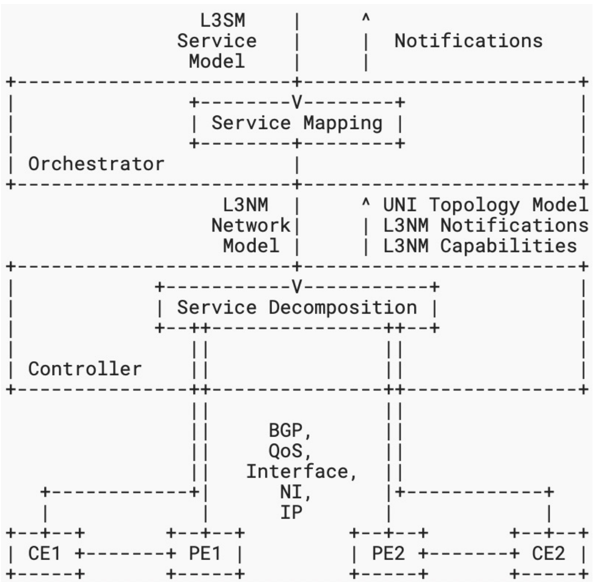
- Service Layer
  - Customer-facing model, SLAs
  - Connectivity scope and guarantees
    - Flow identification
    - Traffic isolation
  - Notice on
    - Statistics on aggregate traffic
    - Failures
    - Planned maintenance operations

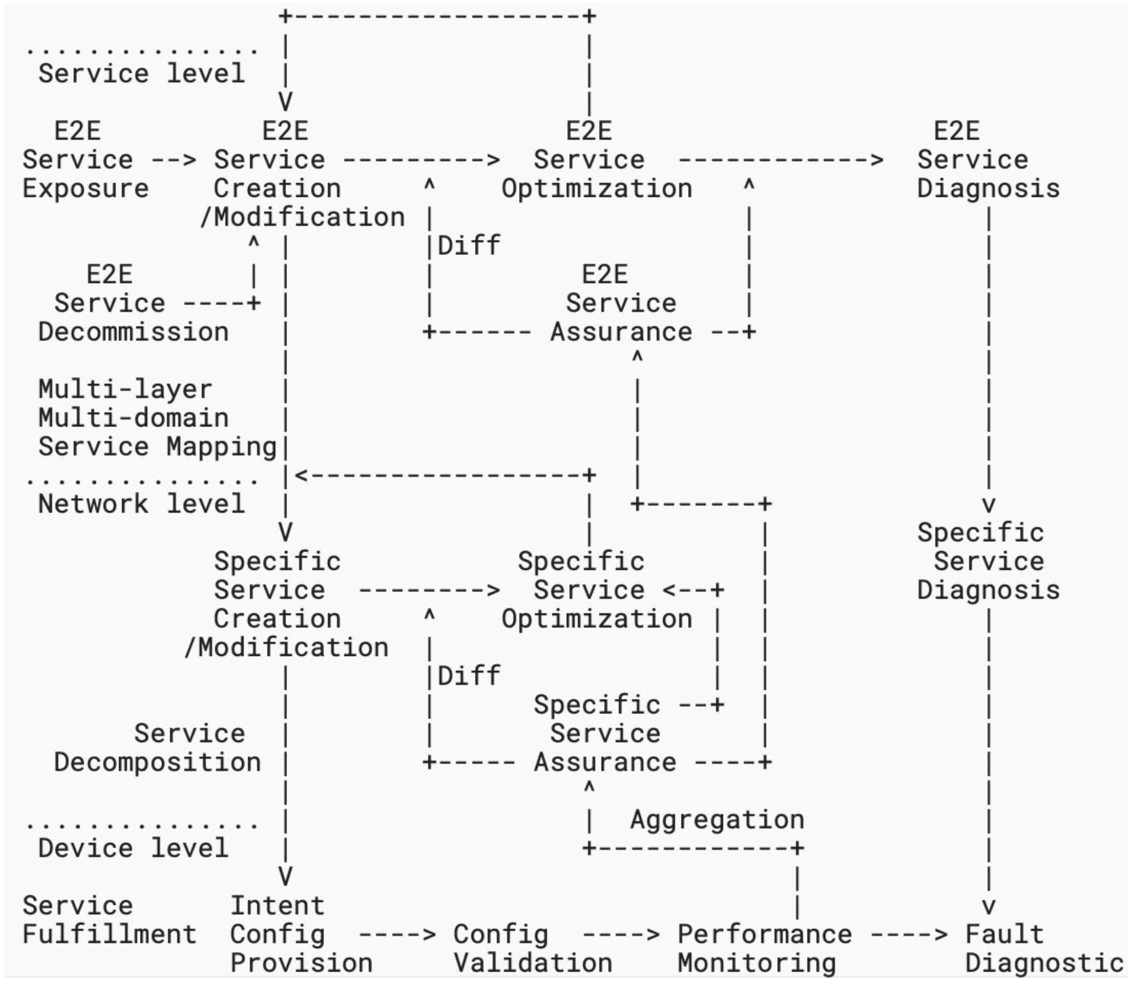
- Network Layer
  - Resource-facing model
  - Network resource management
    - Topologies
    - Inventories
    - Protocol parameters

- Device Layer
  - Device configuration parameters

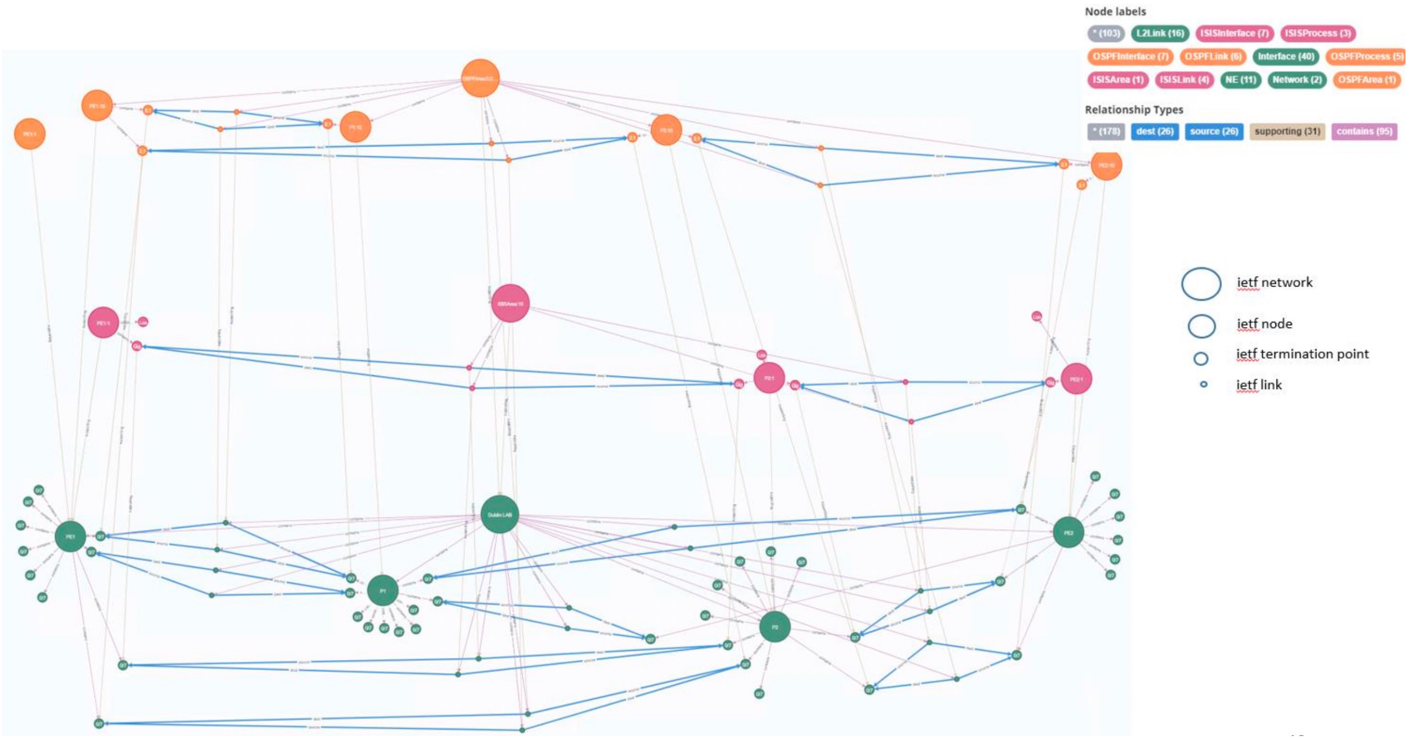
The RFC 8969 Lifecycle

- Connecting the evolution of the different layers
  - Correlations among the models used at the different layers
  - Most relevant examples provided
  - Need for general approaches





The Digital Map Proposal

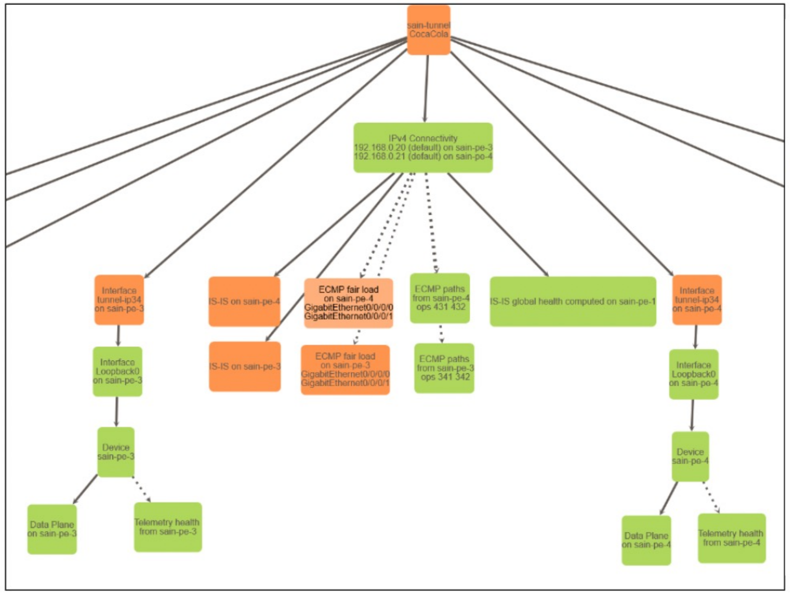


- A Digital Map provides a core multi-layer topology model defining:
  - The core topological entities
  - Their role in the network
  - Core properties that identify entities
  - Relationships between the entities
- Correlates all topological entities at different layers
- Currently applied to *digital twinning*
  - The DM represents the mapping between the real and the twin

The SAIN Proposal

```
+-----------------+
| Service         |
| Configuration   |<--------------------+
| Orchestrator    |                     |
+-----------------+                     |
   |            |                       |
   |            | Network               |
   |            | Service               | Feedback
   |            | Instance              | Loop
   |            | Configuration         |
   |            |                       |
   |            V                       |
   |        +-----------------+    +-------------------+
   |        | SAIN            |    | SAIN              |
   |        | Orchestrator    |    | Collector         |
   |        +-----------------+    +-------------------+
   |            |                     ^
   |            | Configuration       | Health Status
   |            | (assurance graph)   | (Score + Symptoms)
   |            V                     | Streamed
   |     +-------------------+        | via Telemetry
   |     |+-------------------+       |
   |     ||+-------------------+      |
   |     +|| SAIN              |------+
   |      +| agent             |
   |       +-------------------+
   |                ^ ^ ^
   |                | | |
   |                | | |  Metric Collection
   V                V V V
+-------------------------------------------------------------+
| Monitored Entities                                          |
|                                                             |
+-------------------------------------------------------------+
```

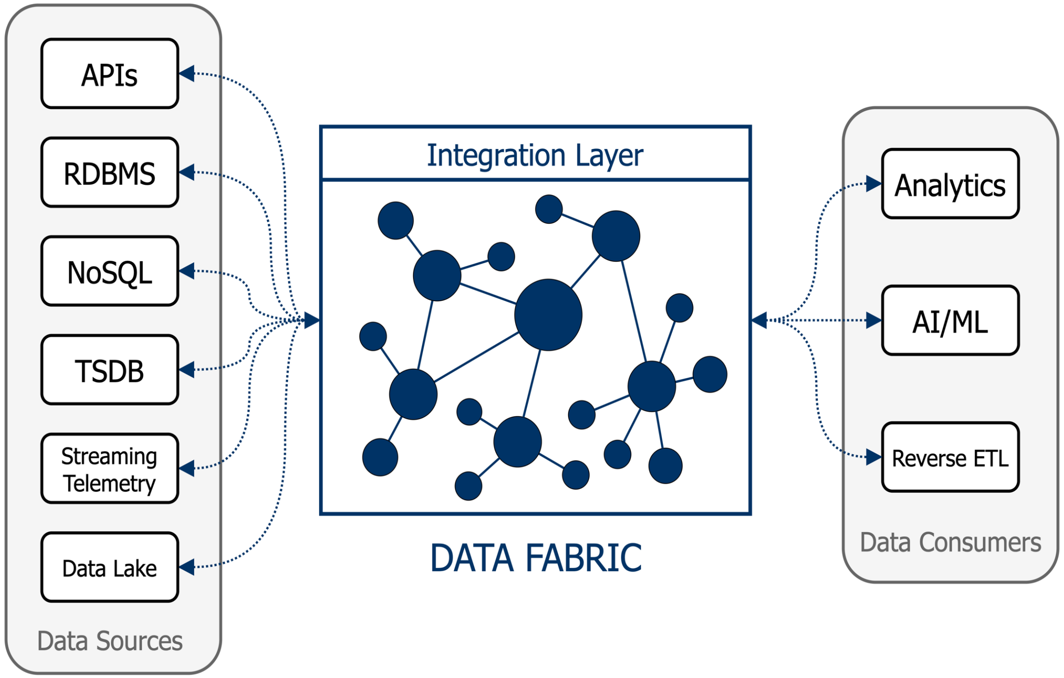
Telefónica



- Address service assurance
  - Following the layered metamodel
  - Map symptoms to causes in service degradation
  - Analyze impact of component failures
- Decompose the problem into smaller components
  - *Subservices*
  - An assurance graph links subservices to the service
  - Subservices are assured independently
  - Infer a service health score
- Assurance graph based on model mappings

YANG and the Data Fabric Paradigm

- Combining YANG data for advanced management
  - Scattered YANG data sources, both at device and service levels of abstraction
  - Multi-vendor networks, with different YANG data models and telemetry protocols
  - Need for data infrastructure to collect, integrate, and expose all these data from the network
  - Other data sources (think cloud-edge and NFV and…)
- Data Fabric architecture provides a unified view of integrated data
  - Graphs to abstract consumers from the underlying complexities of the data source
  - Standard, secure interfaces for interacting with data



The goal → Implement a data fabric solution integrating YANG monitoring data

Standards for Data Fabric: Enter NGSI-LD

ETSI CIM ISG defines the NGSI-LD protocol, which is composed of two parts:

NGSI-LD Information Model

- Based on the labelled property graph (LPG) model
- Semantic annotations based on Semantic Web standards (RDF, OWL)
- Serialized using JSON-LD

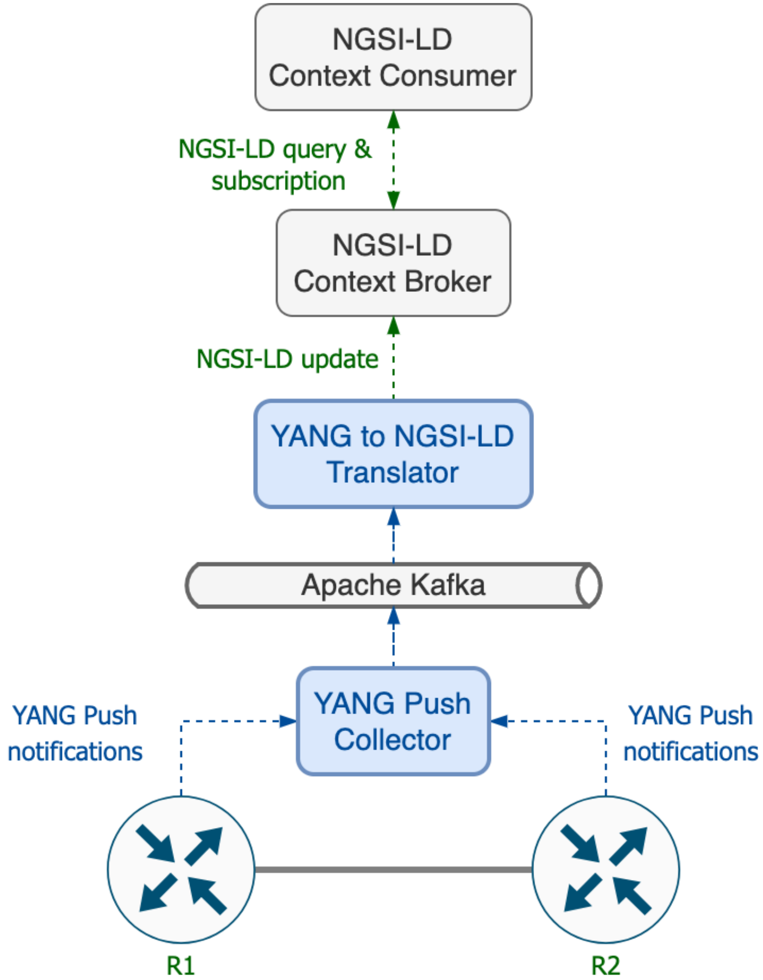
NGSI-LD API

- REST-based API
- Context information management
- Queries & subscriptions
- Temporal evolution
- Distributed & federated architectures

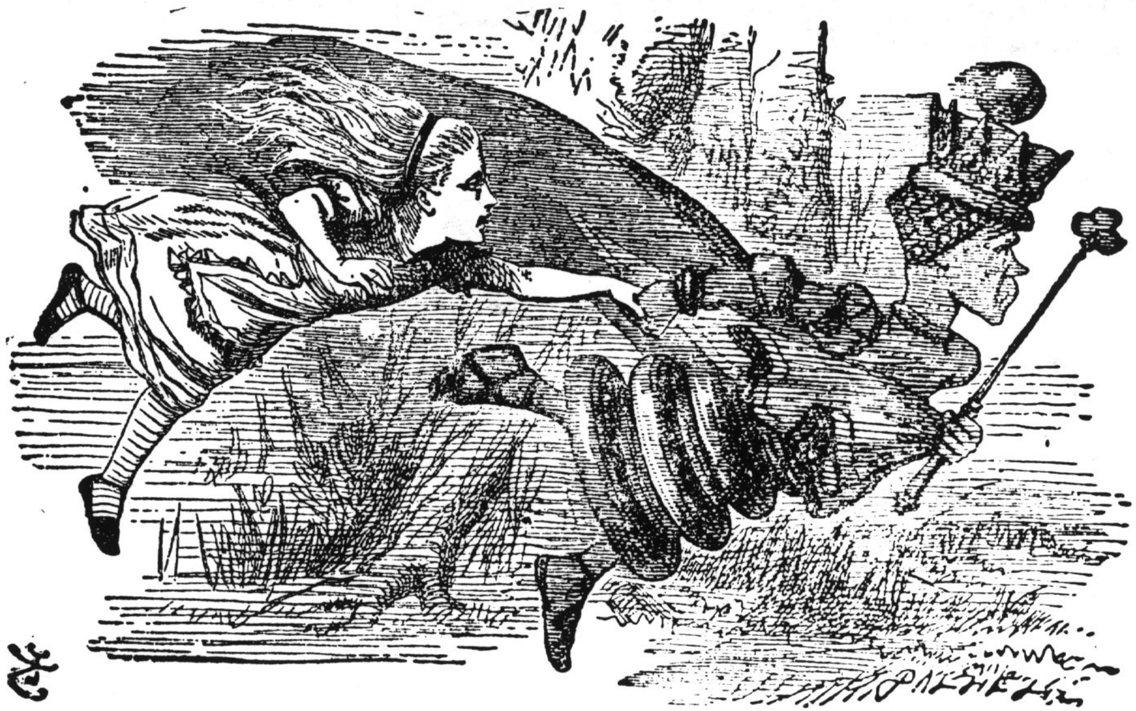
- Address the whole YANG scope
  - Connect YANG data from device and service levels → Support mapping approaches
- Mapping rules focused on NGSI-LD information model
  - Basis for mappings to LPG (e.g., Neo4j) and RDF (OWL ontologies from YANG data models)

A Data Fabric Prototype

- *Containerlab* for network virtualization
  - Based on Docker containers
- Testing devices that implement YANG Push and IETF modules (NMDA ideally)
  - Cisco CSR1000v
  - Looking to include more vendors
- Two custom components
  - YANG Push collector, leveraging the *ncclient* library
  - YANG to NGSI-LD translator
- Align prototype development with YANG native onboarding
  - https://github.com/network-analytics/draft-daisy-kafka-yang-integration/blob/main/draft-daisy-kafka-yang-integration-03.md



Come Running Twice as Fast



Now, here, you see, it takes all the running you can do, to keep in the same place.
If you want to get somewhere else, you must run at least twice as fast as that!