Trusted Service Function Chaining for Mission Critical Services

Tutorial - ISCC 2021

7/September/2021

Bruno Sousa (bmsousa@dei.uc.pt)
Nuno Antunes (nmsa@dei.uc.pt)

University of Coimbra, Portugal

UNIVERSIDADE D COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Contents

How we will run the tutorial

- With passion for the covered topics
- Open to comments, suggestions during tutorial
- Several breaks

- We will share the materials
  - https://github.com/bmsousa/tutorial-ISCC2021

Introduction

- “Public Safety and Critical Infrastructure industries need to be provided with the efficient and adequate digital tools”, as per Ericson

- “Business Critical Services need to be resilient, adaptive, and transformative (innovative)”, as per Cisco

- “Communication is vital. Staying up to date is critical”, as per Motorola

- “Secure, reliable, scalable, and flexible communications infrastructure to serve your vital operations and business services”, as per Nokia

Tutorial Objectives

- Mission Critical Services and their composition into functions
- Insights for the composition of functions for Service Function Chaining
- Comprehensive description of mechanisms to enable trust on service functions and nodes providing resources for Service Function Chaining
- Hands-on on simulation scenarios with SFC policies

Mission Critical Services

- Services in scenarios with some level of criticality:
  - Emergency services for safety of people (police, firemen)
  - Services in critical infrastructures like Smart Grids
  - Health services
  - Services to enable vehicles communication
  - Others like augmented reality or virtual reality

Modelling Critical Services

- Public Safety requirements
  - Fast communications
  - Efficient levels of communication
  - Require faster and more reliable cellular/mobile communication
  - Different levels of priorities, (as per hierarchy of police, firemen, …)
  - End-to-end security

Modelling Critical Services

- Public Safety communication systems, like Public Mobile Radio (PMR)
  - Terrestrial trunk radio (TETRA)
  - TETRAPOL (France)
  - Project P25 (North America)

- Relevant features:
  - Support for reliable communications
  - Secure mission critical voice-centric services
  - Support for group and priority calls (Push to Talk - PTT)
  - Support for Device to Device (D2D) communications or direct mode of operation (DMO)
  - High security and reliability levels

Public Safety Networks

- International First Responders Forum (IFAFRI) Capability Gaps:
  - Know location
  - Identify hazards,
  - Monitor physiological signs
  - …
- Public Safety Networks (PSN) need to:
  - Real-time video
  - Sensing applications with temperature and heart-rate
  - Relying on commercial networks and available smartphones

Advances in Long Term Evolution (LTE) and beyond

- Group call system enablers (GCSE-LTE)
- Proximity services (ProSE)
- Mission Critical Push to Talk (MCPTT)
- Mission Critical Video Services (MCVideo)
- Mission Critical Data Services (MCData)

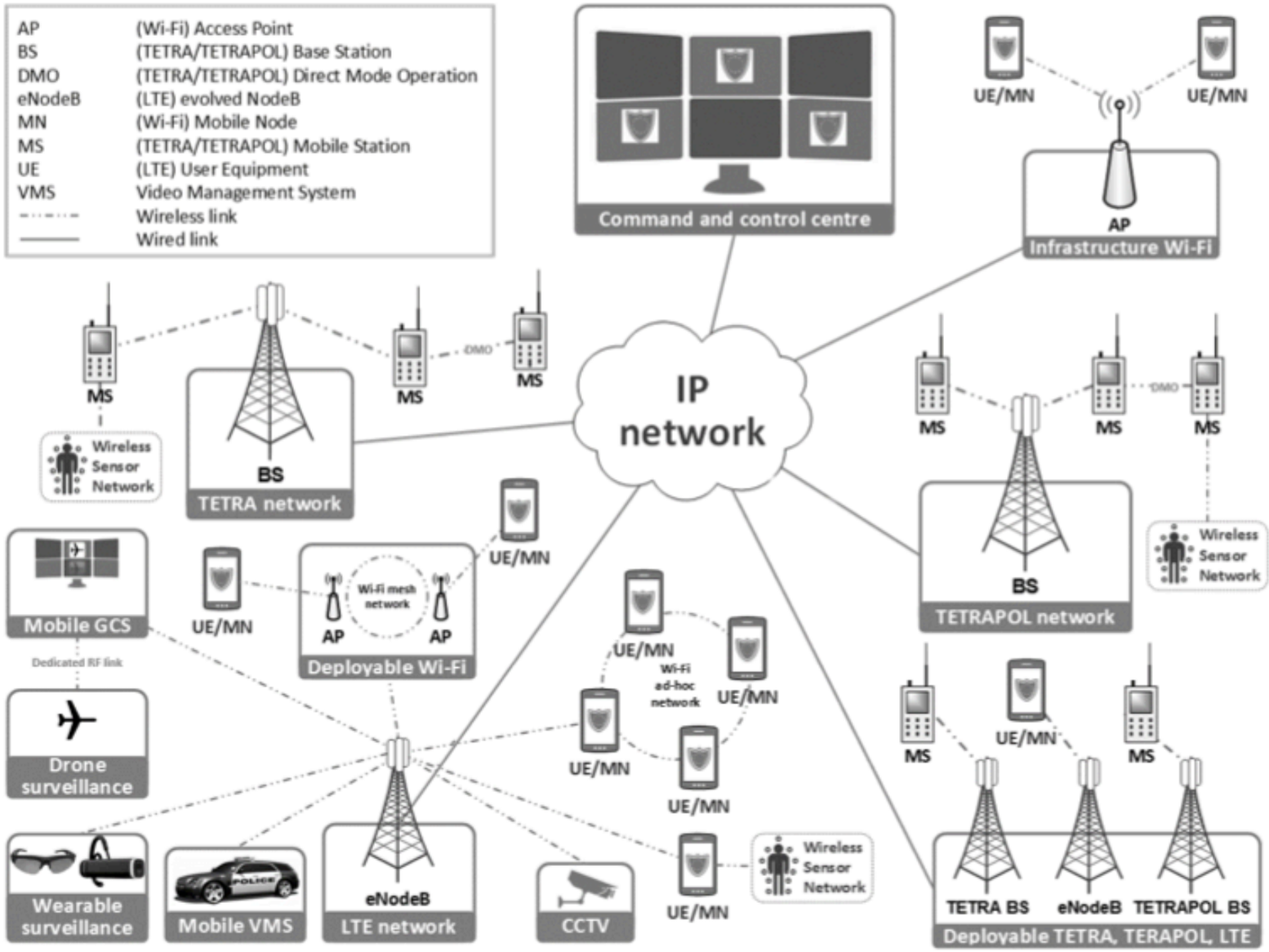
Comments, Questions

SALUS 1/2

- Interoperability between PMR and LTE
- SALUS project with 3 use cases:
  - City security
  - Temporary protection
  - Disaster Recovery
- Enterprise Architecture, with different services:
  - Situation Awareness (Location, monitoring, sensors)
  - Security services (IDS, Forensics)
  - Voice Communication and Interworking Services
  - Others

Hugo Marques et al, "**Next Generation Communication Systems for PPDR: the SALUS Perspective**", 2015

SALUS 2/2

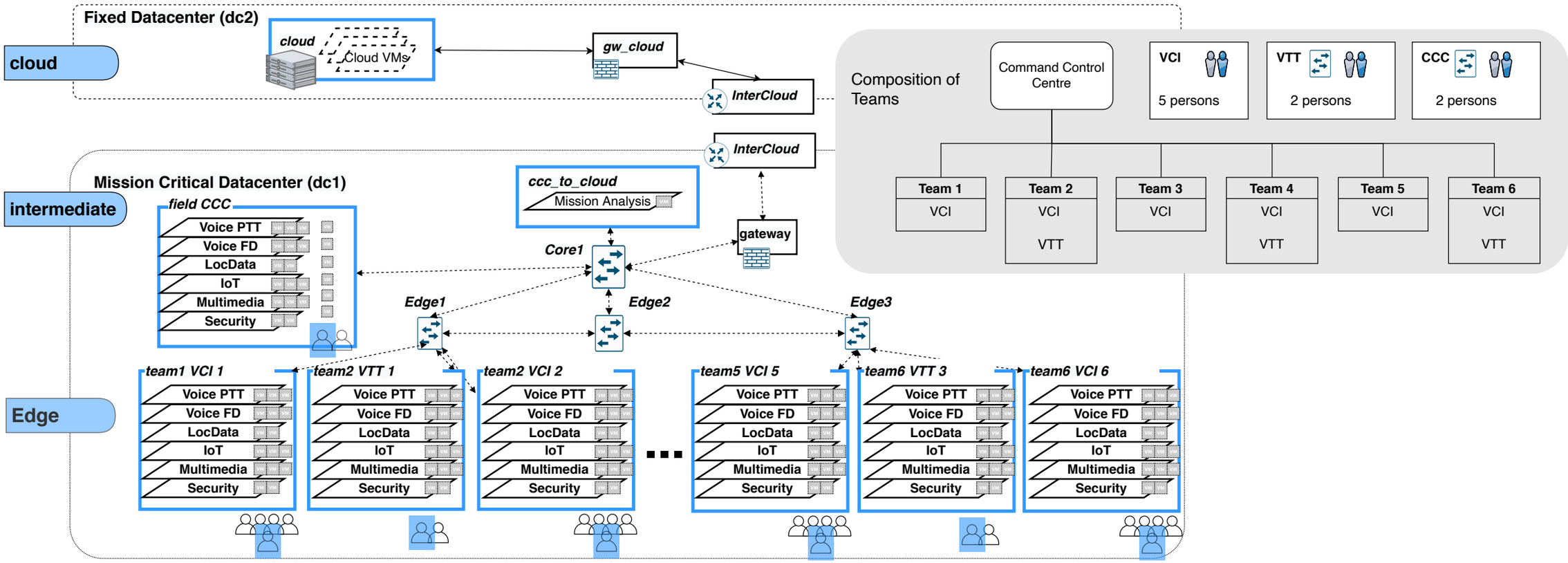


- SALUS Architecture with diverse technologies: TETRA, TETRAPOL, LTE and WiFi

Modelling Critical Services

- Exchange of information
  - Device location
  - Dispatch services
  - Alert messages
  - Late entry (mission is already ongoing)
  - Direct communications D2D (without infrastructure)
  - Data communications (e.g., technical information of vehicles, detailed maps to assist mission operations)

Mission Critical Services 1/2



- Considering the Directives for the Operations in wildfire combat

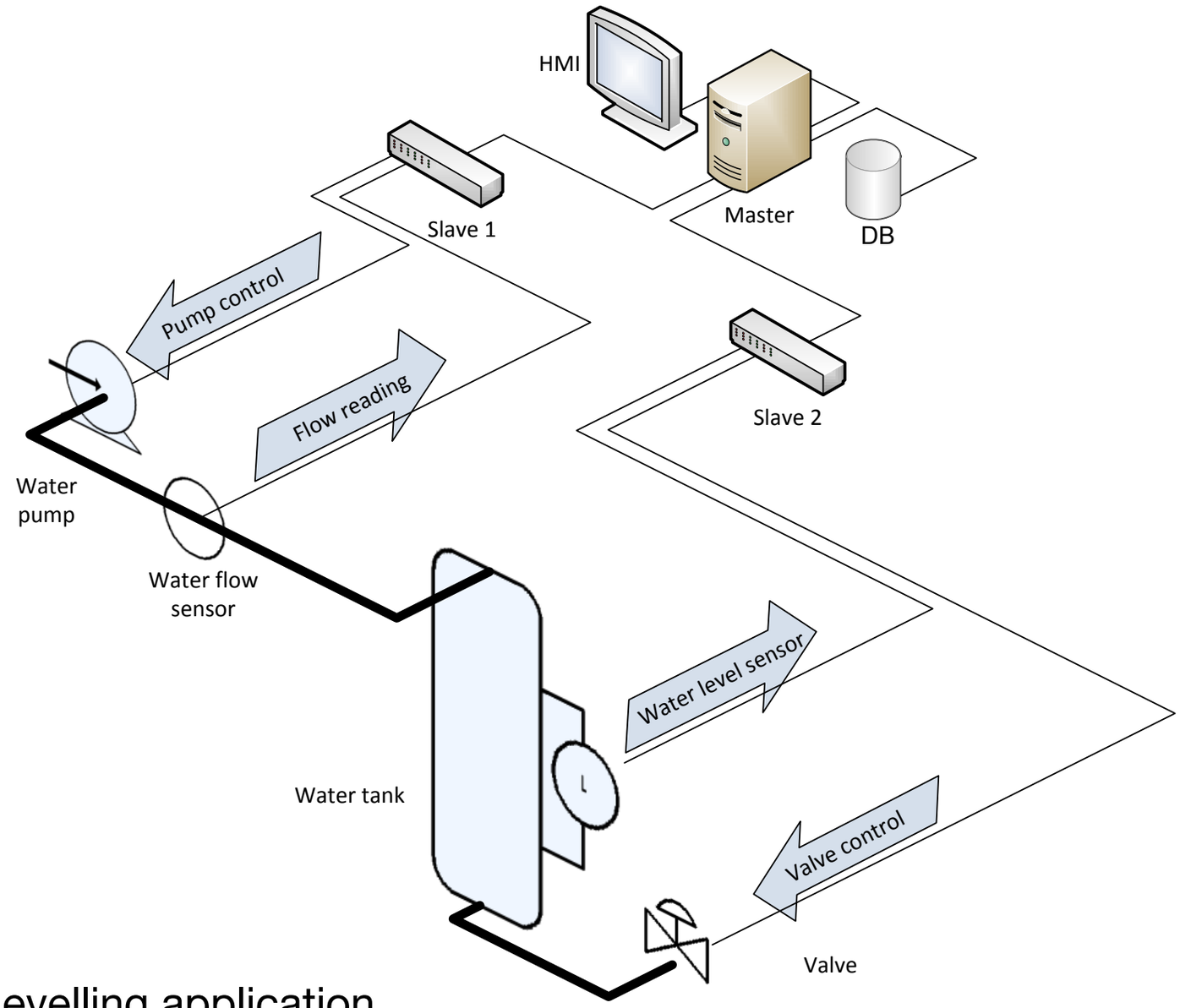
- Compliant with Fog/Edge and SDN paradigms

Bruno Sousa et al, "**Service Function Chaining in Wildfire Scenarios**", SoftIoT, NetSOFT 2021

Mission Critical Services 2/2

- Several services:
  - Voice PTT and Voice FD
  - Location Data
  - IoT
  - Multimedia
  - Security
  - Mission Analysis
- Different policies per service

Critical Infrastructures



- Water filling/levelling application
- Using Modbus/TCP to control slave nodes (controlling water pumps, valves)
- Human Machine Interface (HMI) to assess behaviour of system

Bruno Sousa et al, "**ELEGANT: Security of Critical Infrastructures with Digital Twins**", 2021

Comments, Questions

Service Function Chaining

- Key terminology (not exhaustive)
- Motivation for SFC
- SFC standardisation (IETF, ETSI, ONF)
- SFC in the literature
- Virtualisation platforms
- Edge and cloud continuum & Software Defined Networks (SDN)

What is Service Function Chaining ?

What is SFC

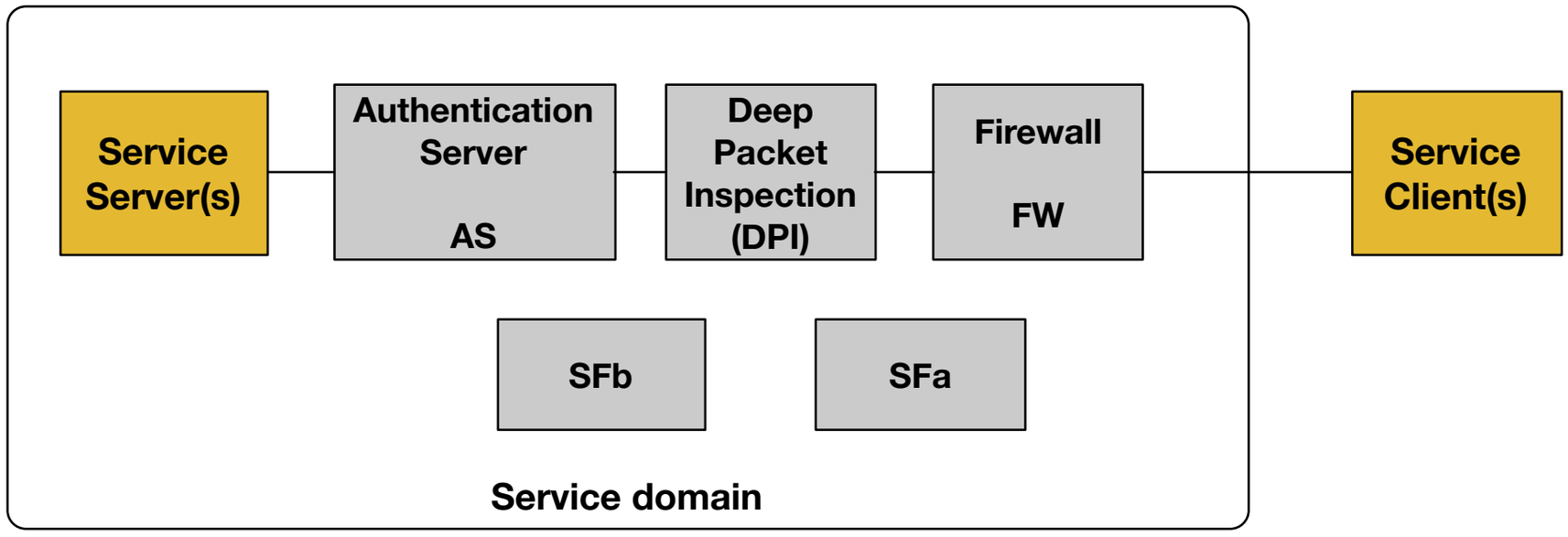
“Definition and instantiation of an ordered list of instances of service functions and subsequent application of traffic flows through the service functions”, by IETF, RFC 7498, 2015

Terminology:

Service Function <=> ETSI ISG NFV **Network Function**

ETSI GS NFV-EVE 005, “**Network Functions Virtualization (NFV) Ecosystem”**, 2015

What is SFC (example)



- Service: Database Service
- SFC purpose: secure access to information

H. Hantouti, N. Benamar, T. Taleb, and A. Laghrissi, “**Traffic steering for service function chaining**,” IEEE Commun. Surv. Tutorials, vol. 21, no. 1, pp. 487–507, 2019

SFC - key concepts 1/3

- **Service Function** (SF) — a function that is responsible for a specific processing
  - can be placed in any layer of the OSI model
  - can be associated with virtual or physical elements

(Brief) Examples of Service Functions:

Firewall	Deep Packet Inspection (DPI)
WAN and application accelerators	Load Balancers
NAT	TCP Optimizers
Lawful Interception (LI)	Host-ID injection

SFC - key concepts 2/3

- **Service Function Chain** (SFC) - Set of service functions and ordered constraints that must be applied to traffic flows, which result from the service classification
- **Classification** — match traffic flows against policy rules, select traffic that enters a service overlay. Determines which SFC processes the traffic
- **Network overlay** — logical network for mapping SFs
- **Network Service** — offering provided by an operator that is delivered through one or more service/network functions (composed service)
- **Service overlay** — network overlay created for the purpose of forwarding data to the service functions. Provides SF connectivity in the existing network topology (i.e., path between SFs)
- **Service Function Chain Symmetry** - can be:
  - unidirectional - traffic is forwarded through the SFs in one direction
  - bidirectional - traffic is forwarded in both directions

SFC - key concepts 3/3

- **SFC Encapsulation** — allows the creation of the SF chaining and conveys information regarding chain (data-plane metadata) —> Network Service Header (NSH)
- **Service Function Path** (SFP) - path that packets assigned to a service must traverse, that is which SFs are visited by a packet. Mechanism to express the result of applying more granular requirements of SFC (e.g., specify which endpoints will be visited by a packet acting as a SF).
- **Service Function Forwarder** (SFF) - entity responsible to forward traffic to one or more SFs belonging to a SFC. Processes the information carried in the SFC Encapsulation to determine the appropriate SF for the traffic.
- **Service Function Chain Proxy** (SFC Proxy) - entity that does the interface between SFC aware and SFC unaware domains. Logical entity to allow the communication with SFs that do not understand the SFC encapsulation.

Motivation for Service Function Chaining

Motivation for SFC

- Deployment of Services Functions (SFs) are static depend on the network topology —> limit the introduction of new services or modifications of services

- Complex network changes and device configurations (i.e., cannot be static)

- Elastic service environments require rapid creation, destruction, placement of functions or network elements (i.e., avoid slow provisioning or static deployments)

- High availability requirements (i.e., redundant service functions…)

- Policies for Service Functions ordering (i.e., a packet can only be routed if it has traversed a firewall)

Motivation for SFC

- Transport independence (i.e, service functions can operate in network overlays with different technologies, GRE, VLANs, MPLS)
- Elastic Service Delivery (i.e., increase/decrease functions in services)
- Traffic selection (i.e., not all traffic requires the same enforcement) and must consider the type of flow (i.e. bidirectional)
- Limited end-to-end service visibility (i.e. if service crosses multiple administrative domains, how to troubleshoot it)

Motivation for SFC

- Classification synchronisation between service functions (i.e., different functions can classify traffic differently).
- Complexity in maintaining symmetric traffic flows (e.g., firewalls need to keep state of flows)
- Interoperability between vendors

Comments, Questions

SFC *à la* IETF

- Service Function Chaining (SFC) works on 4 topics:
  - Metadata - type-length-values of data exchanged
  - Security and privacy - mechanisms to secure metadata (i.e. authentication, integrity protection, confidentiality, others)
  - Operations, Administration, and Maintenance (OAM) and operations & Management - Management mechanisms, YANG models, OAM frameworks
  - Transport considerations - transport considerations like congestion indication and response

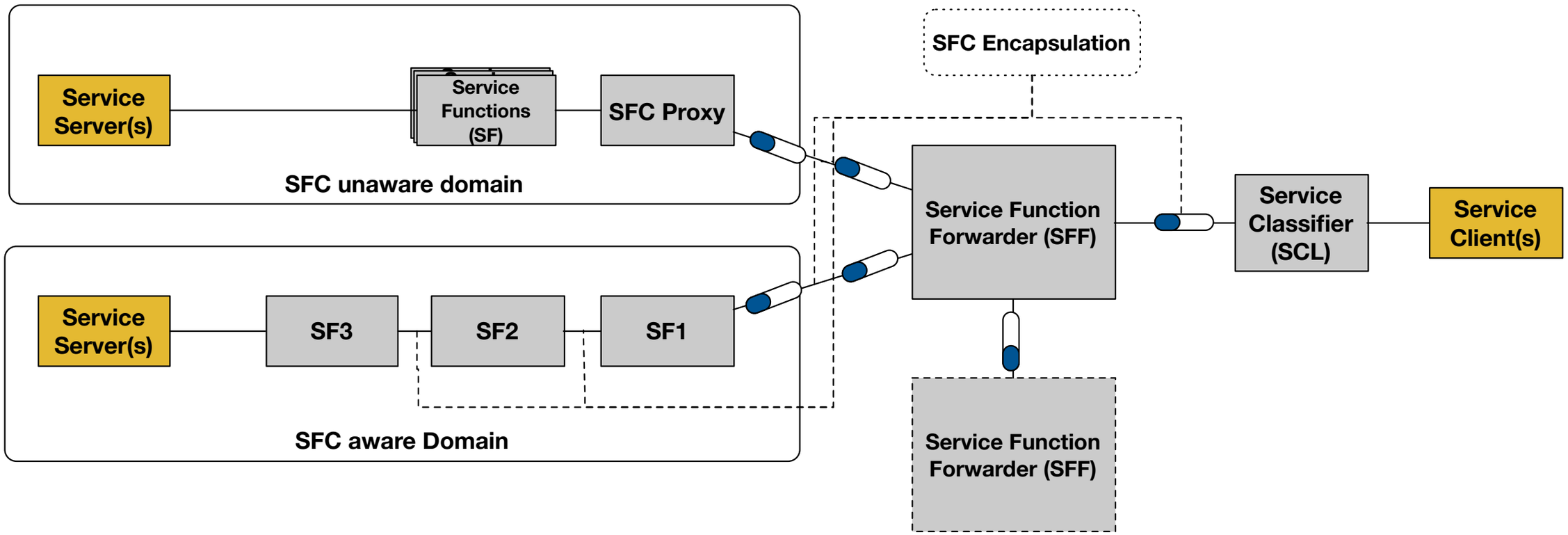
SFC Problem Statement 1/2

- Standardised in RFC 7498
  - Topological dependencies — Service deployments are coupled with network topology
  - Configuration complexity - Changing the order of services is not simple
  - Constrained availability - The dependency on the topological info, leads to constrained availability
  - Consistent ordering of SFs - Changing the order of SFs is complex
  - Application of Service Policy - Policies are complex and they are applied in an overload fashion (repeated through SFs)
  - Transport Dependence - Dependence on network transport technologies (i.e. Ethernet, GRE, MPLS, VLAN…)

SFC Problem Statement 2/2

- Standardised in RFC 7498
  - Elastic Service Delivery - Complexity in adding/removing SFs in existing configurations, issues with routing and risky
  - Traffic Selection Criteria - Inflexible and coarse traffic selection, data may traverse all the SFs without being required to
- Limited End-to-End Service Visibility - Complexity in troubleshooting service-related issues, multiple domains
- Classification/Reclassification per SF - SFs do not leverage from the classification performed by others
- Symmetric Traffic Flows - Existing deployments are static regarding the forwarding and reverse associations of SFs
- Multivendor SFs - interoperability issues

SFC base architecture



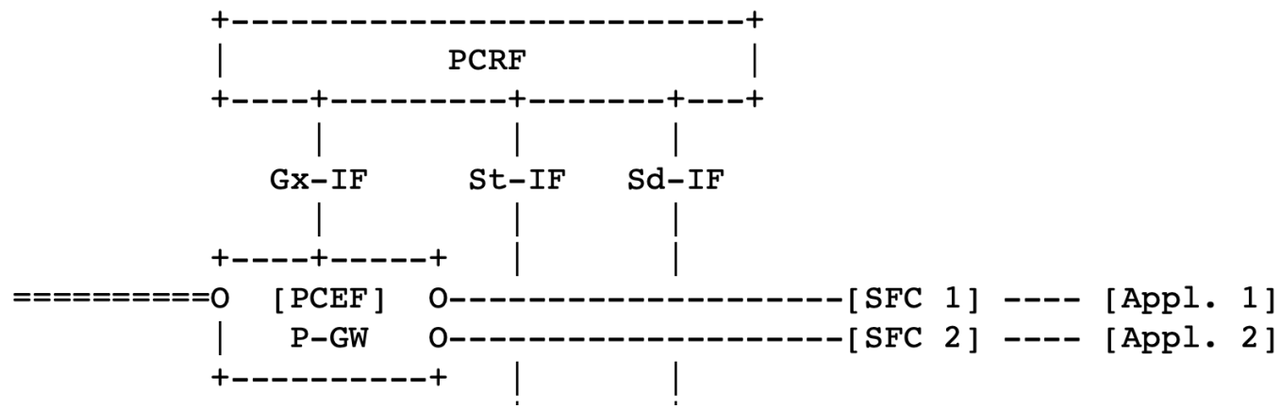
- Base architecture defined in RFC 7665
- SFC Encapsulation defined in RFC 8300, the Network Service Header (NSH)
- Network Service Header with Next Protocol RFC 8393
- Hierarchical architecture defined in RFC 8459
- SFC Operations, Administration and Maintenance (OAM) Framework defined in RFC 8924
- Subscriber and Performance Policy Identifier Context in NSH defined in RFC 8979

SFC Classifier

- Enforces the policies for services, applications
- Classifies traffic to check how it must be treated
- Establishes the Service Function Path (SFP) as per the configured policies
- Adds SFC encapsulation to packets (the Network Service Header).
- Can act as ingress, egress of SFC aware domain

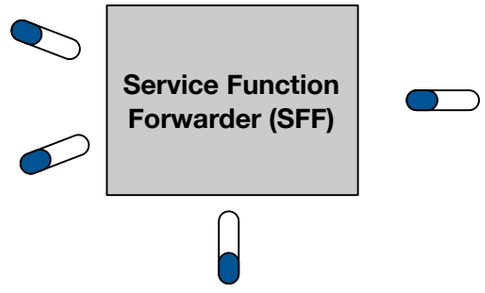
SFC Classifier
(example in mobile networks)

- Classification through the **Access Point Name** (APN)
  - APN web.meo.pt —> for Meo operator to classify web traffic (HTTP), which can be associated with a specific SF chain.
- Classification through **metadata** (i.e. ports, IP address, user subscriber information) that can be mapped into policies by control elements like the PCRF Policy and Charging Rules Function)



IETF Internet Draft, “Service Function Chaining Use Cases in Mobile Networks”, 2019

SF Forwarder



- SFP Forwarding - traffic is forwarded to the appropriate SF upon arrival to SFF.
  - SFF uses the information in the SFC Encapsulation (Service Path Identifier - SPI, Service Index - SI)
- Terminate SFPs
- Maintain flow state
- Role performed by SDN routers/switches in the NFVI

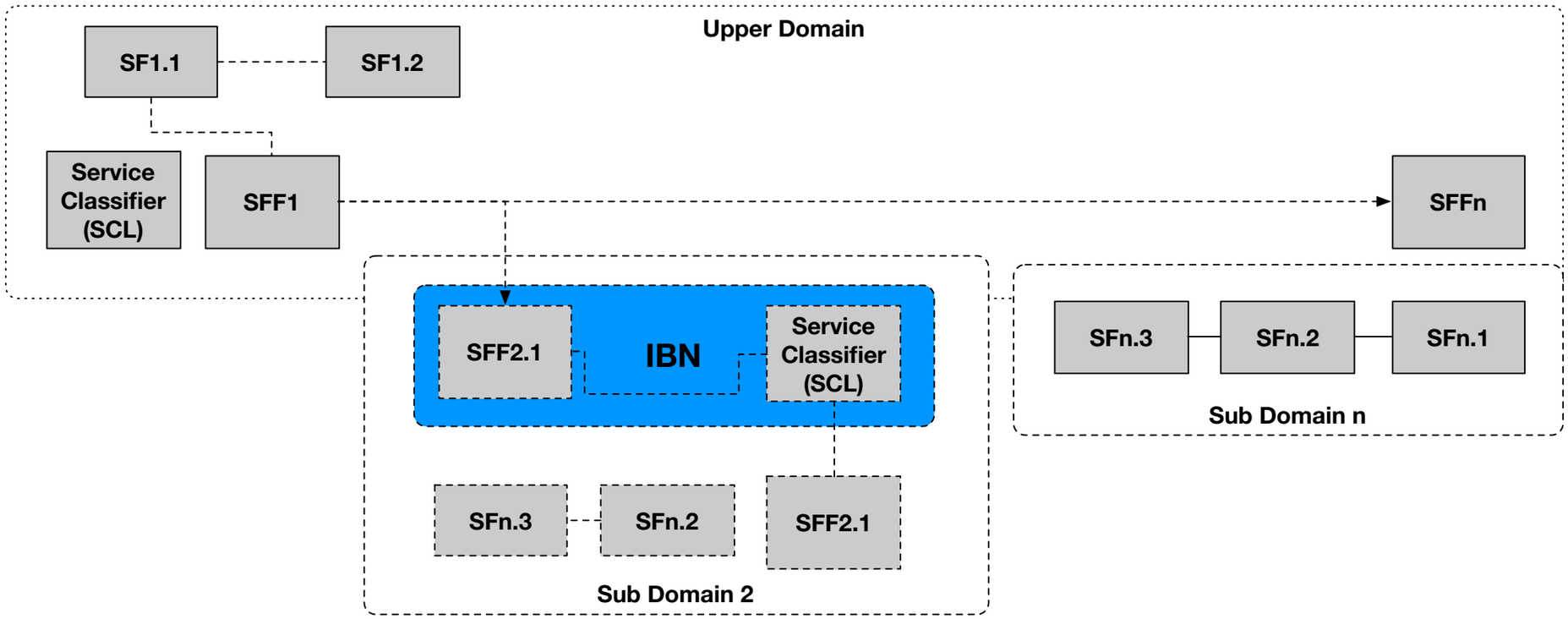
SF Forwarder
(example of mapping)

Service Path Identifier (SPI)	Service Index (SI)	Next Hop	Transport Encapsulation
10	255	192.0.2.1	VXLAN-gre
10	254	198.51.100.1	GRE
40	251	198.51.100.2	GRE
50	200	01:23:45:67:89:ab	Ethernet

- SFF can choose the next hop based on preferences or policies

RFC 8300, “**Network Service Header (NSH)”**, 2018

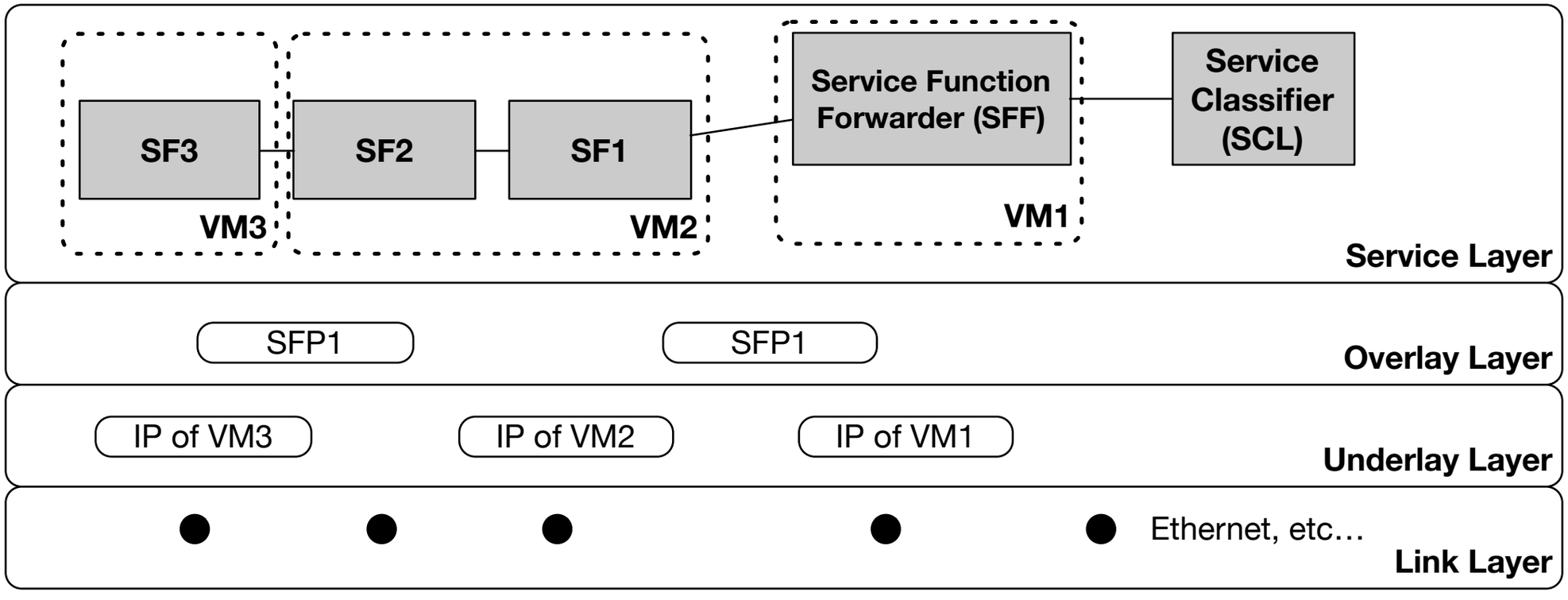
Hierarchical SFC



- Allows to decompose large-scale network into multiple (and more functional) domains
- Includes the **Internal Boundary Node (IBN)** that bridges packets between upper and lower level domains. IBN has classification rules for sub domain

SFC OAM

- Specified in RFC 8924
- The SFC OAM establishes a SFC layering model



SFC OAM components

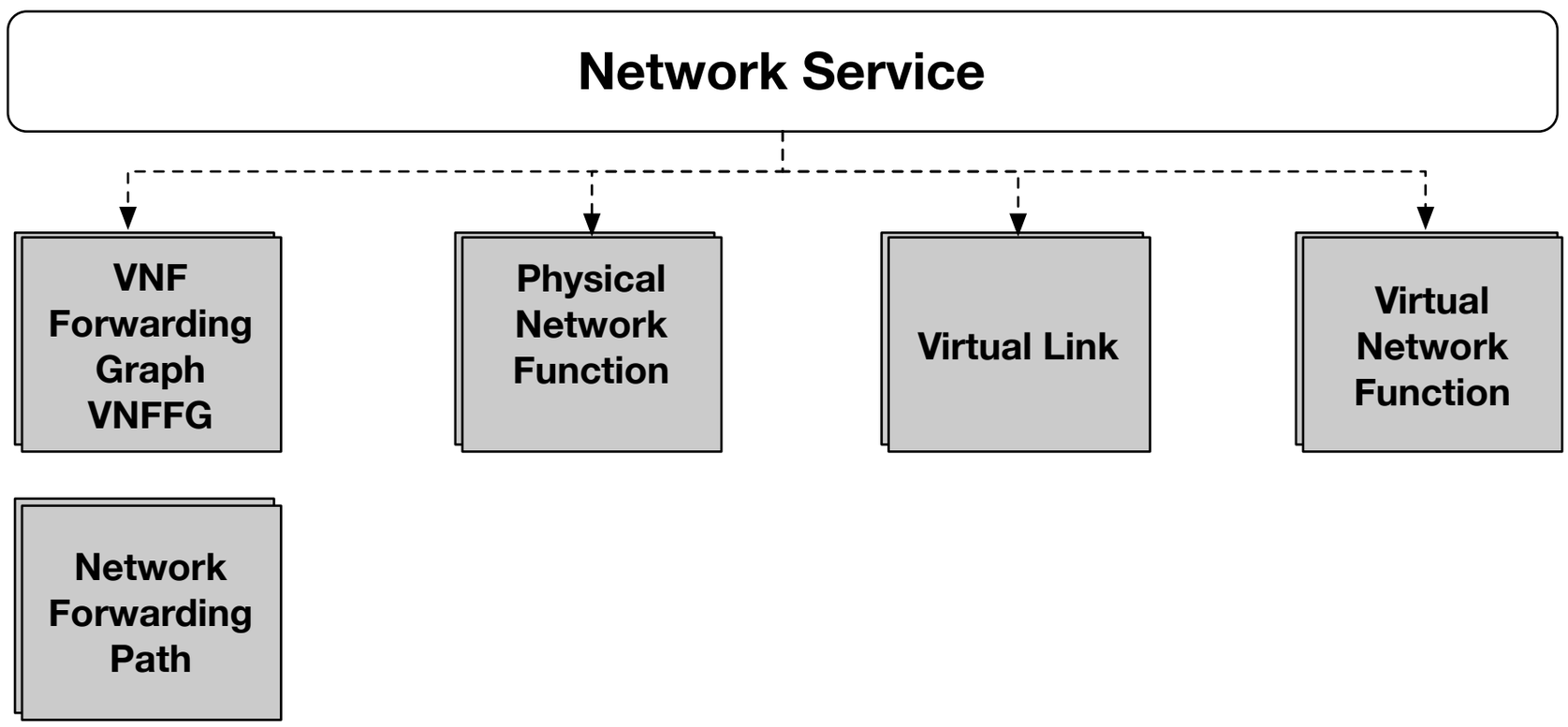
- SF Component
  - SF availability
  - SF performance measurement
- SFC Component
  - SFC Availability (validate SFC and actual forwarding path e.g. SFP)
  - SFC performance measurement (end-to-end SFC)
- Classifier Component
  - Updated configuration (e.g. policies)
  - Functioning as intended

SFC OAM functions

- **Connectivity functions**
  - Verify the connectivity and availability of SFs
    - Verification of Path MTU
    - Verify packet re-ordering and corruption
    - Verify policy of SFC, SF
    - Verification and validation of forwarding paths
    - Test alternate paths for reliability
- **Continuity functions**
  - Sent periodically to verify reachability of SF of a SFC (e.g. detect link failures, SF outages)
    - Notify detected failures
- **Trace functions**
  - Trigger actions, for instance to gather information, capabilities of SFs
- **Performance measurement functions**
  - Packet loss, delay, delay variance in SFC. Can be measured pro-actively or on-demand

Comments, Questions

SFC *à la* ETSI



- Specified in ETSI GS NFV-MAN 001
- Network Service & VNF Forwarding Graphs

Key concepts 1/2

- ETSI GR NFV 003
  - **Virtual Network Function** (VNF) - Network Function that can be deployed in a NFV Infrastructure
  - **Physical Network Function** (PNF) - Network Function that is associated with software or hardware systems
  - **Virtual Link** (VL) - Connection points with the connectivity relationship of two or more entities (VNFs, PNFs)
  - **Network function** (NF) - functional block that has well-defined interfaces and well-defined behaviour
  - **VNF Forwarding Graph** (VNFFG) - Forwarding graph of network functions, with at least one VNF
  - **Network Forwarding Path** (NFP) - Ordered list of connection points forming a chain of network functions

Key concepts 2/2

- ETSI GR NFV 003
  - **Virtual Infrastructure Manager** (VIM) - functional block responsible to control and manage compute, storage and network resources of the infrastructure
  - **Network Functions Virtualisation Orchestrator** (NFVO) - functional block responsible to manage the lifecycle of Network Services (NS) and VNF lifecycle. It is supported by VIM
- **Network Service Descriptor** (NSD) - deployment template (catalogue) for a network service
- **VNF Descriptor** (VNFD) - deployment template of a VNF with information for deployment and operational behaviour requirements (e.g., number of virtual CPUs, RAM)
- **VNFFG Descriptor** (VNFFGD) - deployment template describing the topology of the network service, contains references to VNFs/PNFs and Virtual Links
- **Virtual Link Descriptor** (VLD) - deployment template describing the resource requirements for a link between VNFs and PNFs (e.g. dual path for resilience)

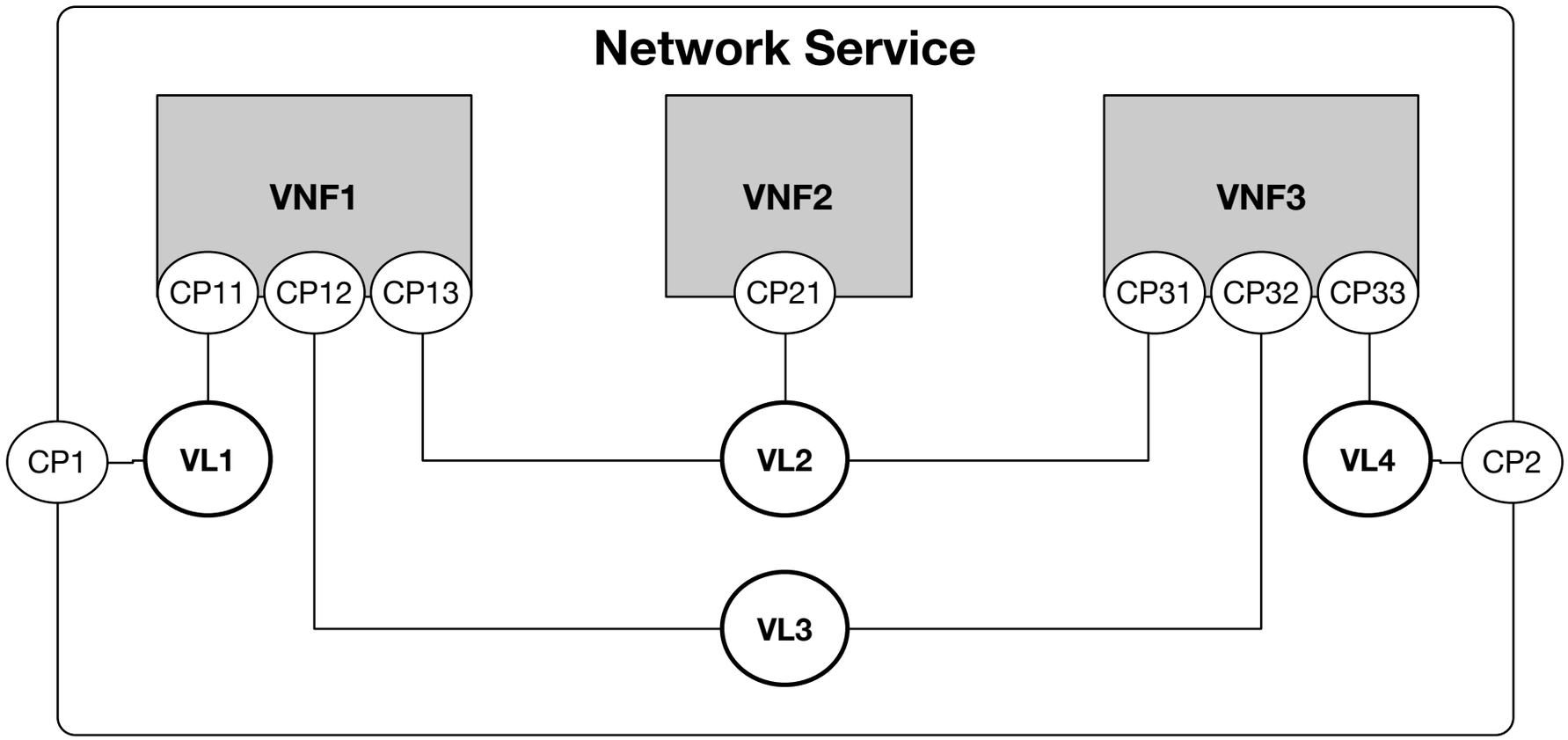
Network Service

- Described in the Network Service Descriptor (NSD):
  - References the VNFD
  - References one or more the VNFFGD (e.g. for control-plane traffic, for user-plane traffic with QoS)
  - References virtual link descriptions
  - Information of VNF dependencies (sequence of network nodes and links that need to be instantiated)
  - Scaling policies (calls per second and action to scale to other flavour)
  - Monitoring parameters to indicate levels of network service availability or others, for instance for scaling policies (e.g., calls-per-second)

Virtual Network Function

- Described in the VNF Descriptor (VNFD):
  - Virtual links
  - Connection points (external interfaces)
  - Lifecycle events
  - Dependencies
  - Monitoring parameters (CPU utilisation, memory, …)
  - Scaling policies
  - Information for deployment, Virtual Deployment Unit (VDU) with info of VM image, requirements in terms of computing, storage and network

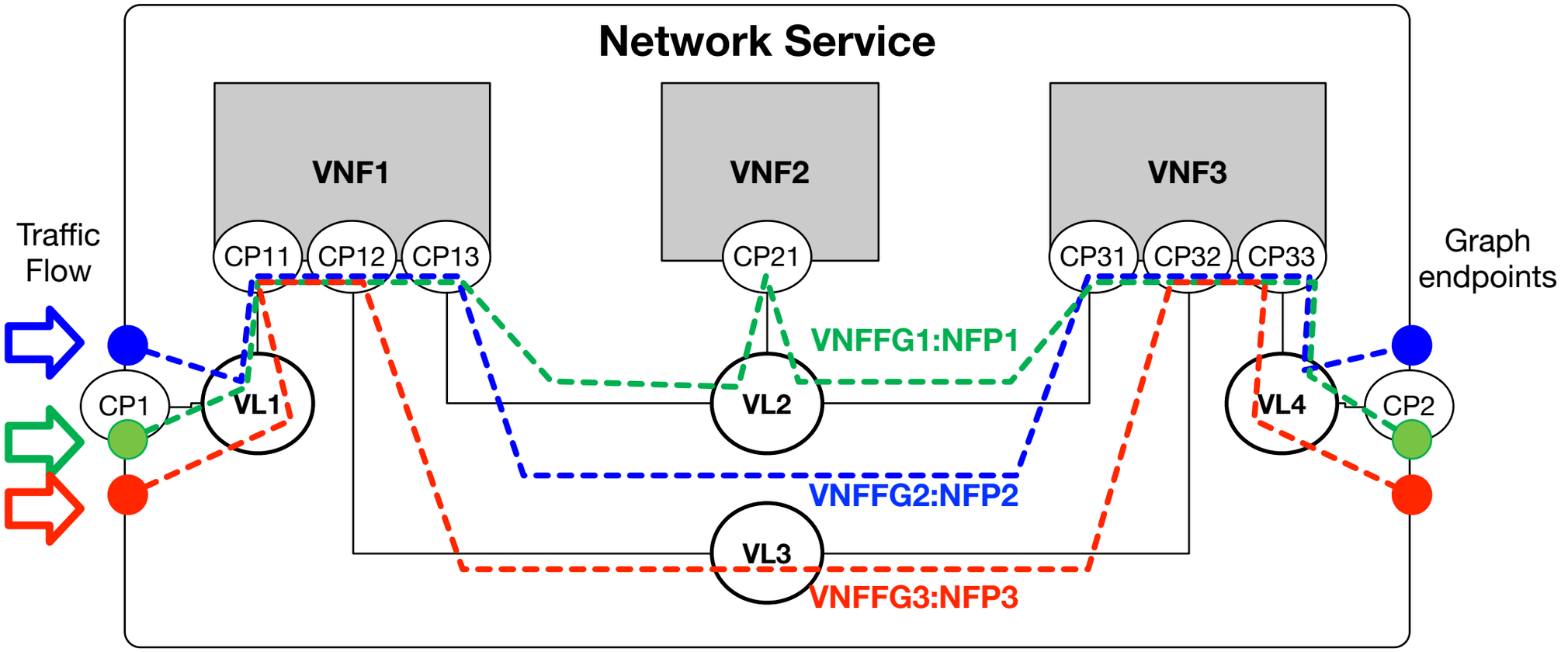
Virtual Link & Connections



ETSI GS NFV-MAN 001

- Described in the Virtual Link Descriptor (VLD)
  - QoS (latency, jitter, …)
  - Connectivity type (E-LAN, E-Tree for load balancing, …)

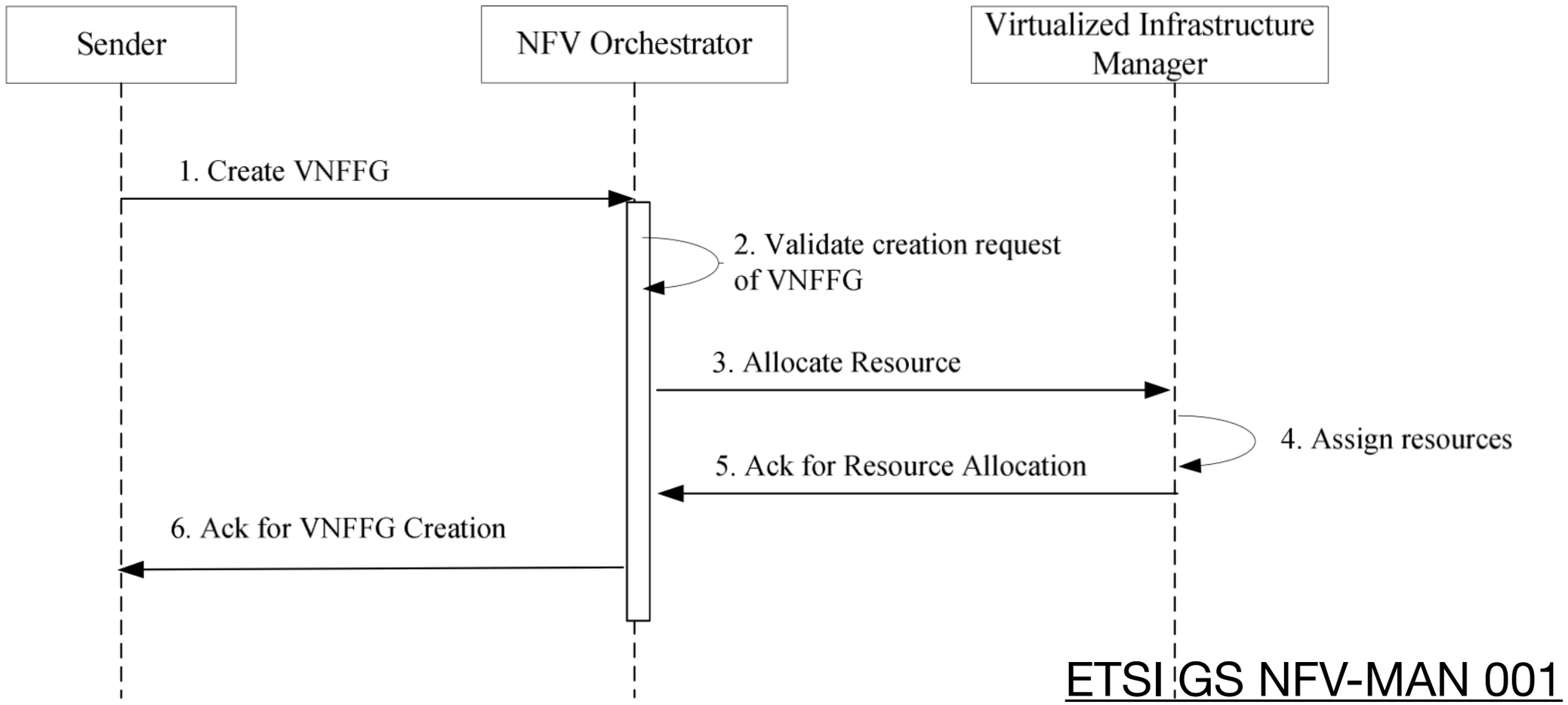
VNF Forwarding Graph 1/2



ETSI GS NFV-MAN 001

- Described in the VNFFG Descriptor (VNFFGD)
  - Network Forwarding Paths
  - Connection Points
  - Virtual Links

VNF Forwarding Graph 2/2

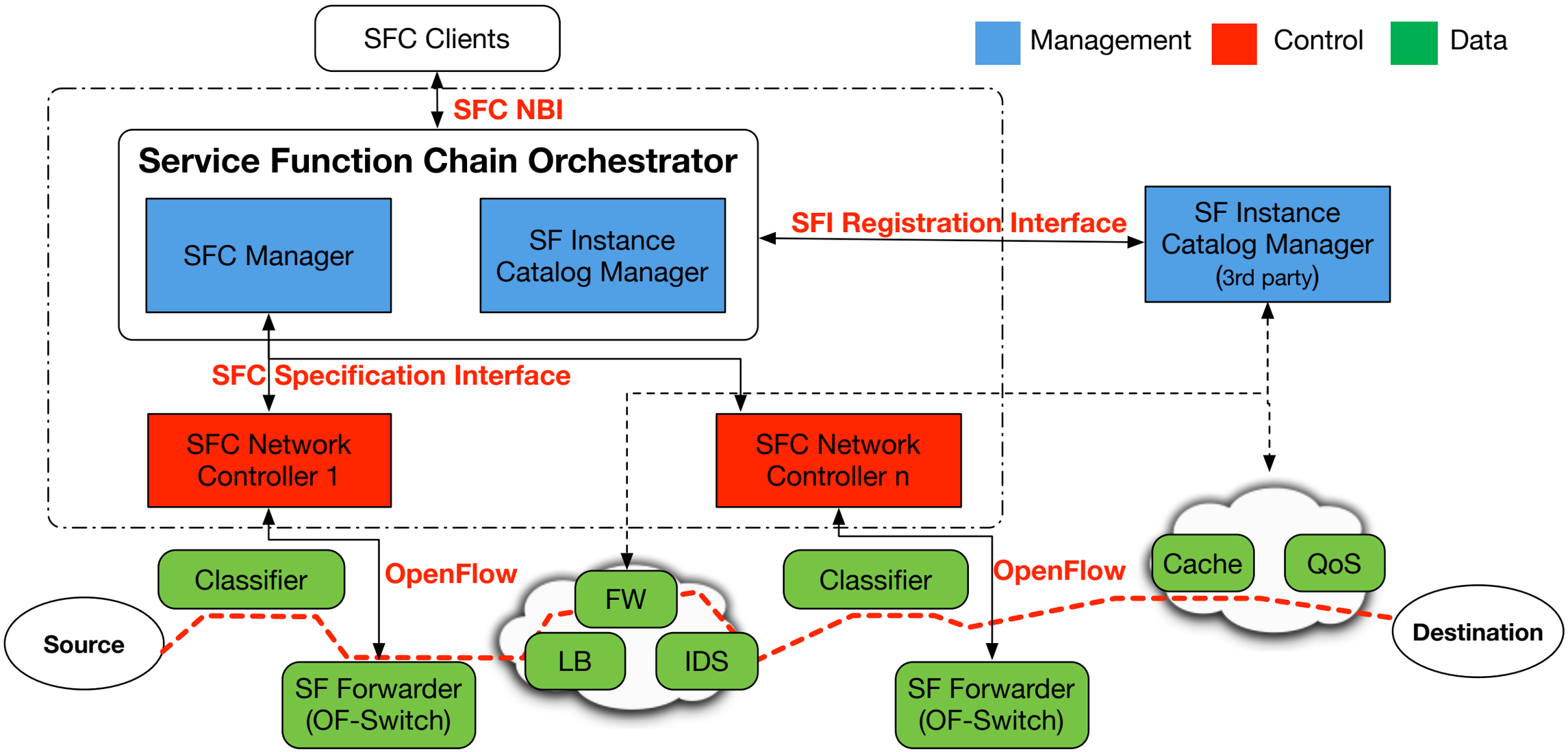


ETSI GS NFV-MAN 001

- Creation of a VNFFG
- VNFFG lifecycle management also includes update, delete and query operations

Comments, Questions

SFC *à la* Open Network Foundation (ONF)



- Specified in ONF TS-027

SDN SFC Architecture 1/3

- Includes management, control and data plane elements
- Management and control elements (SFC Orchestrator, SDN controllers) convert policies into network topology paths
- The SF Instance Manager manages the lifecycle of SFs
- The SFC Network controller manages the steering and chaining of paths for SFs
- The SFC Classifier performs classification and inserts a NSH header

SDN SFC Architecture 2/3

- The SFF forwards packets to SFs, considering the information in the SFC header.
- SFF can rely on OpenFlow, which enables the forwarding:
  - L2-L3 info such as MAC, IP addresses
  - L4 info such as TCP, UDP ports (only for OF versions above 1.3)
- Also specifies an information model for the diverse interfaces:
  - SFC-NBI — between SFC Orchestration and SFC Clients
  - SFI-Registration — between SFC Orchestration and SF Instance Manager
  - SFC-Specification — between SFC Manager and Network Controller

SDN SFC Architecture 3/3

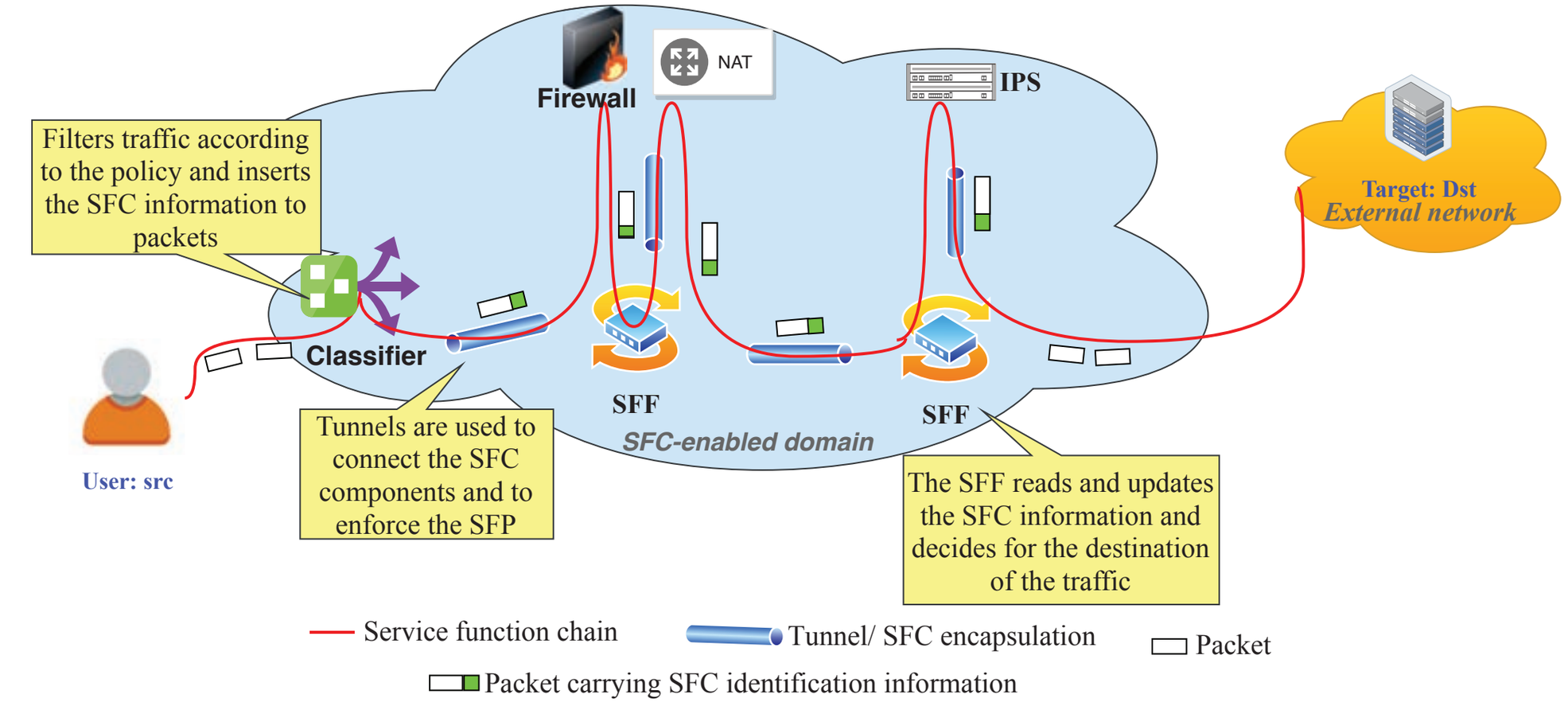
- Compliant with IETF SFC architecture
- No further updates since 2015
- Preliminary support was introduced in the ONOS SDN controller (but without further updates in current versions)

Comments, Questions

SFC in the literature

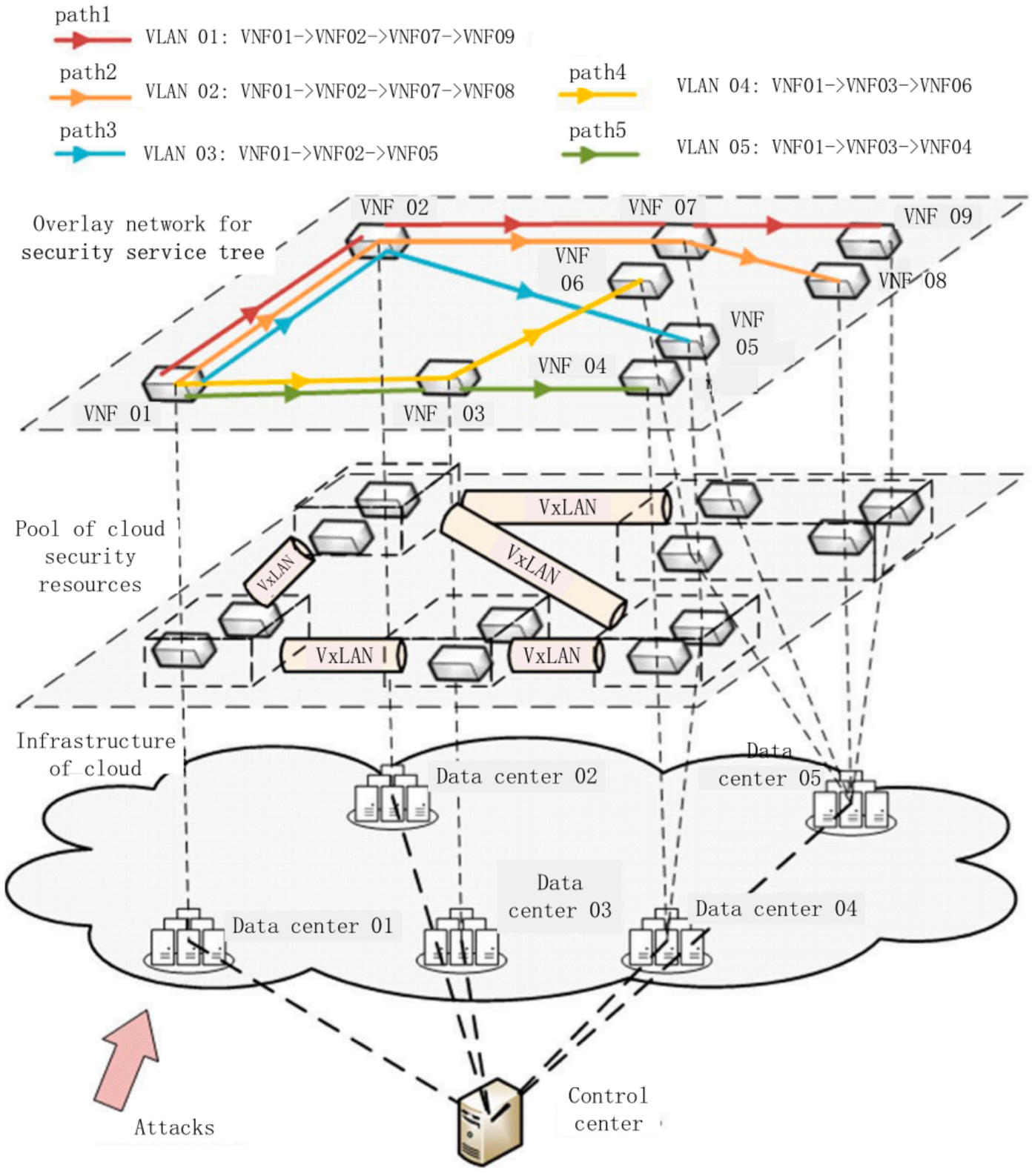
- SFC has been employed in multiple areas:
  - 5G networks with multiple tenants
  - In heterogeneous domains
  - In scenarios with edge computing
  - In scenarios with high requirements of security

SFC architecture
(deployment example)



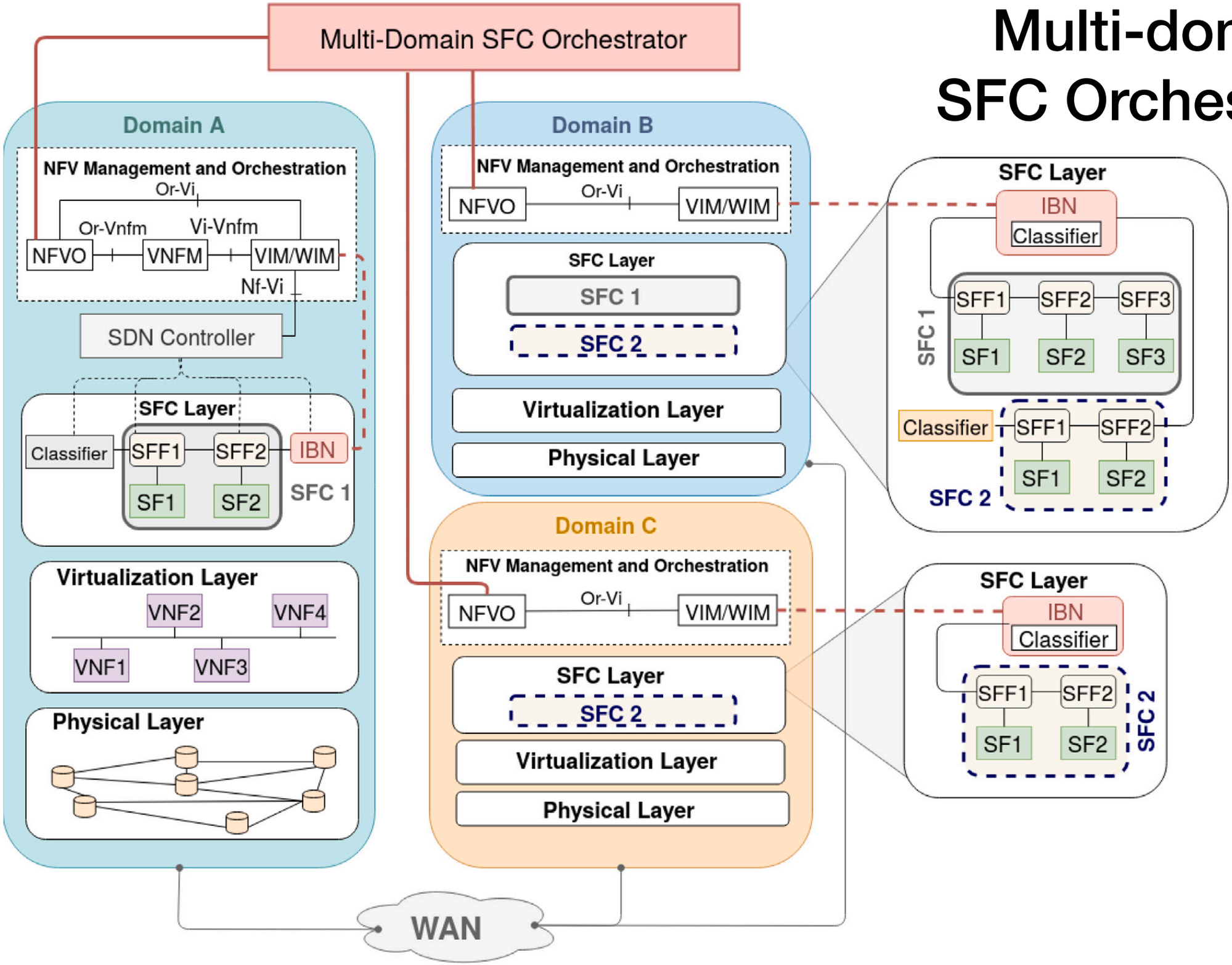
H. Hantouti, N. Benamar, and T. Taleb, "**Service Function Chaining in 5G and Beyond Networks: Challenges and Open Research Issues**," IEEE Netw., pp. 1–8, 2020.

SecSFT



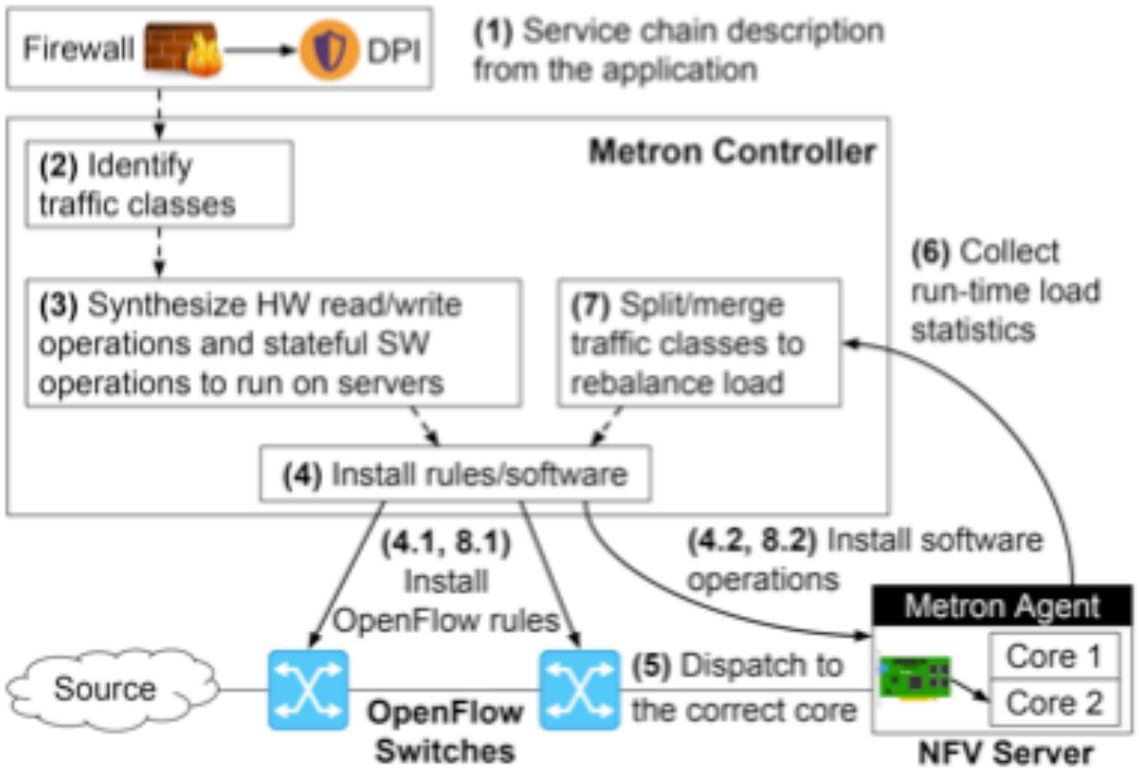
J.-L. Luo, S.-Z. Yu, and S.-J. Peng, "**SDN/NFV-Based Security Service Function Tree for Cloud**," IEEE Access, vol. 8, pp. 38538–38545, 2020.

Multi-domain SFC Orchestrator



N. Toumi, O. Bernier, D.-E. Meddour, and A. Ksentini, "**On cross-domain Service Function Chain orchestration: An architectural framework**," Comput. Networks, vol. 187, no. August 2020, Mar. 2021.

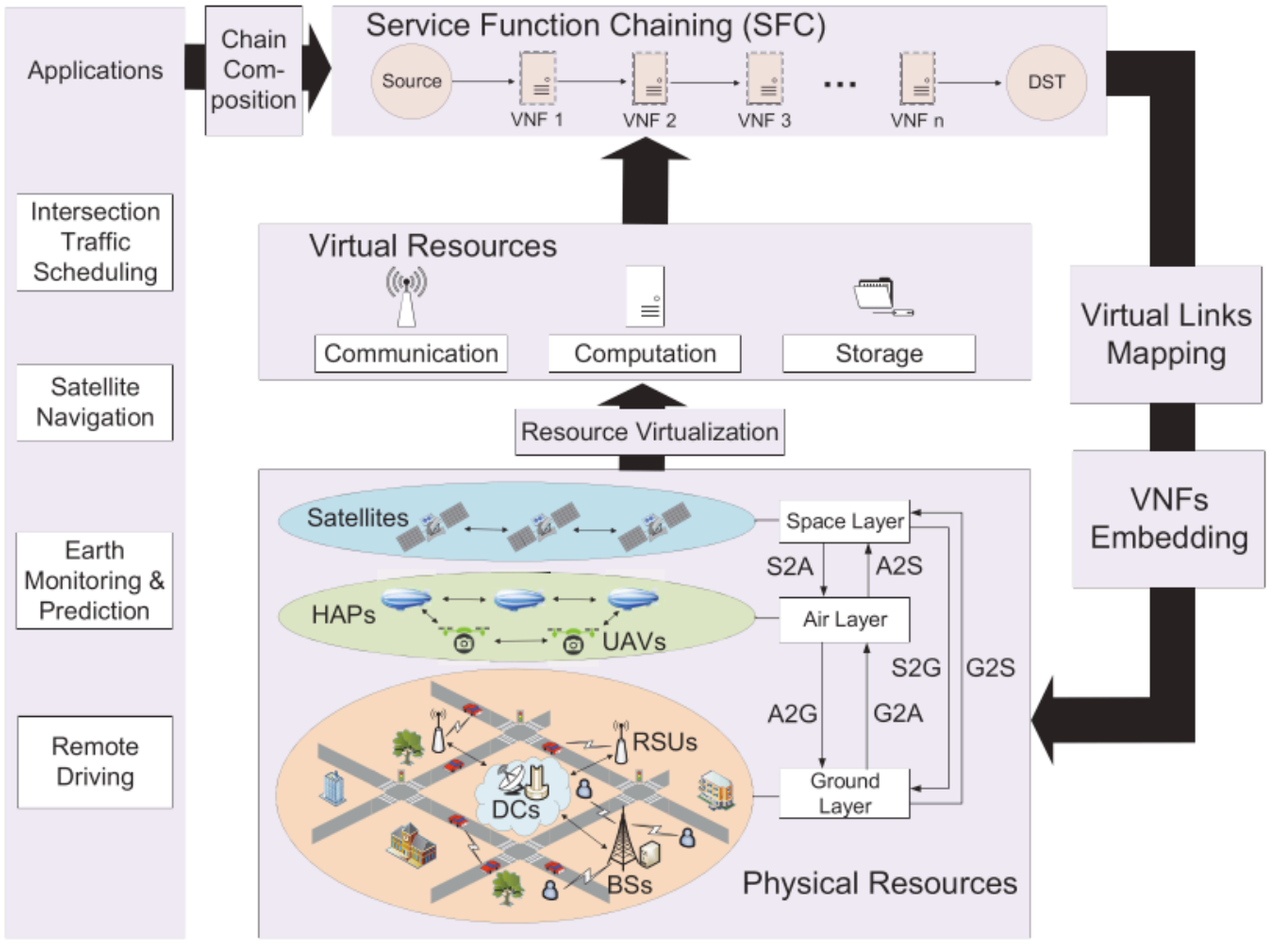
Metron NFV platform for service chains



- Achieves high performance by leveraging multiple cores in servers
- Includes placement techniques with minimal overhead

G. P. Katsikas, T. Barbette, D. Kostić, G. Q. Maguire, and R. Steinert, "**Metron: High-performance NFV Service Chaining even in the Presence of Blackboxes**," ACM Trans. Comput. Syst., 2021.

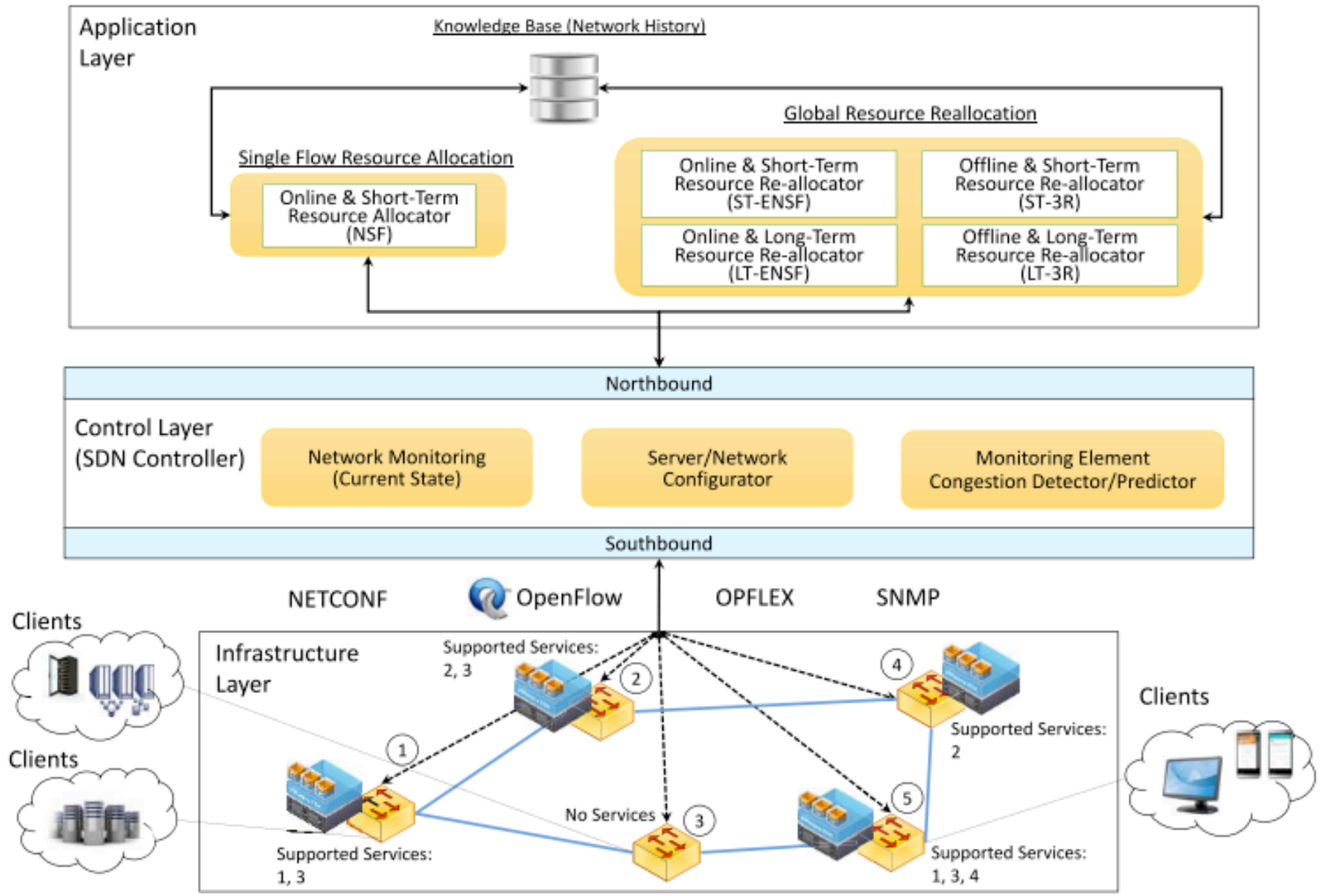
SFC in space-air-ground networks (SAGIN)



- Considers the tradeoff between computation and communication resources (NFV hosting multiple SFCs)
- Consideres heterogeneous networks

G. Wang, S. Zhou, S. Zhang, Z. Niu, and X. Shen, "**SFC-Based Service Provisioning for Reconfigurable Space-Air-Ground Integrated Networks**," IEEE J. Sel. Areas Commun., Jul. 2020.

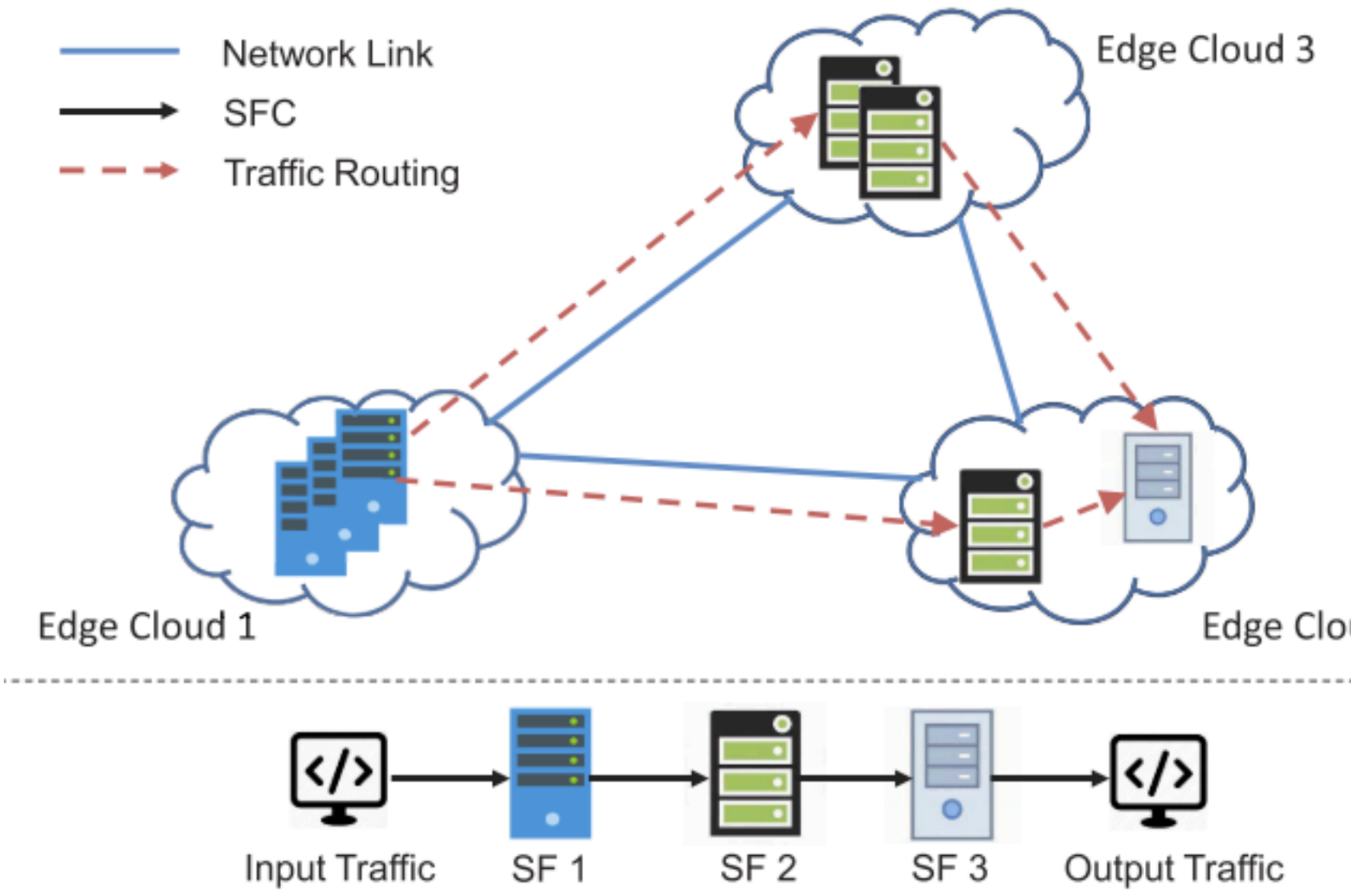
Energy-Aware SFC for SDN



- Considers the tradeoff between energy consumption and Quality of Service

M. M. Tajiki, S. Salsano, L. Chiaraviglio, M. Shojafar, and B. Akbari, **"Joint Energy Efficient and QoS-Aware Path Allocation and VNF Placement for Service Function Chaining**," IEEE Trans. Netw. Serv. Manag., vol. 16, no. 1, pp. 374–388, 2019.

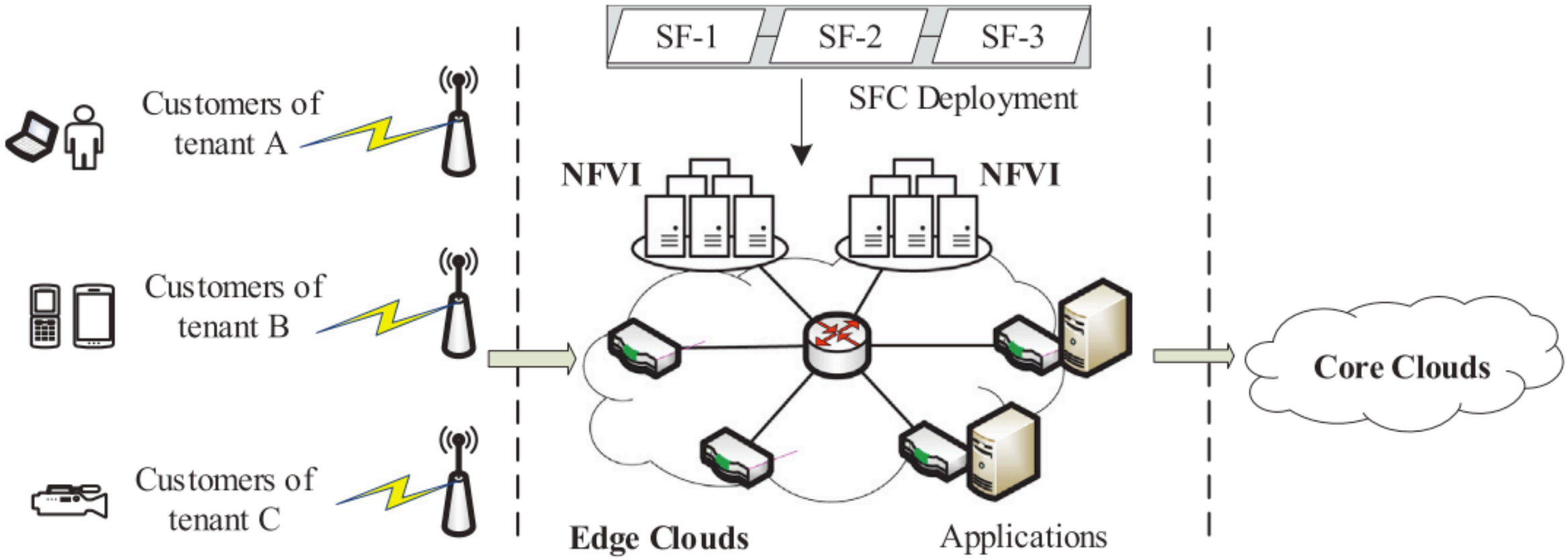
Edge SFC



- Orchestration for service function chaining to maximize cost-efficiency (optimised resource provision and traffic routing)

Z. Zhou, Q. Wu, and X. Chen, "**Online Orchestration of Cross-Edge Service Function Chaining for Cost-Efficient Edge Computing**," IEEE J. Sel. Areas Commun., vol. 37, no. 8, pp. 1866–1880, 2019.

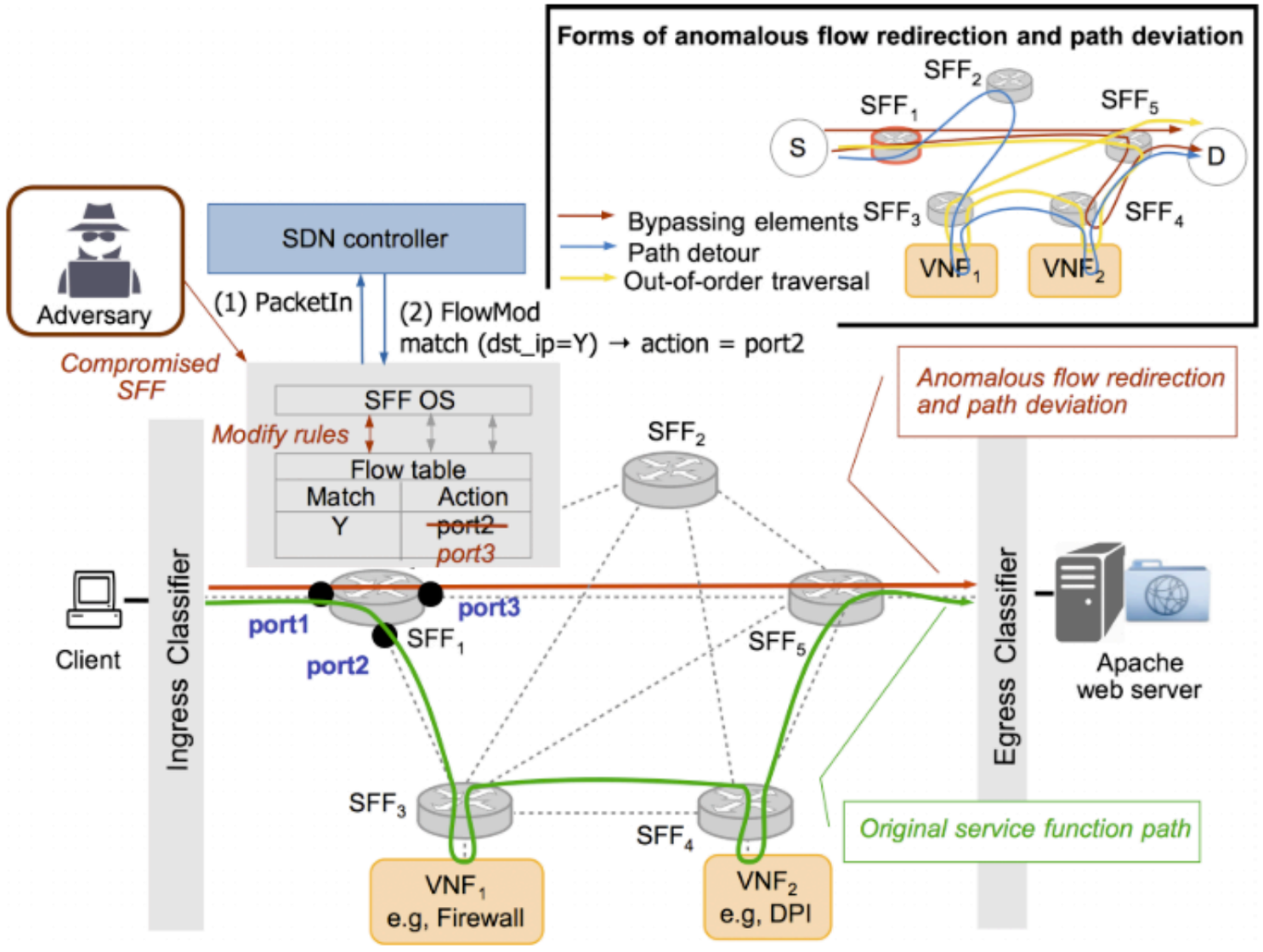
Adaptive SFC mapping in 5G



- Proposes an adaptive mechanism to map SFCs, considering multiple tenants and resource efficiency
- Relies on Deep Reinforcement learning models for the SFC mapping

G. Li, B. Feng, H. Zhou, Y. Zhang, K. Sood, and S. Yu, "**Adaptive service function chaining mappings in 5G using deep Q-learning**," Comput. Commun., vol. 152, no. 2019, pp. 305–315, Feb. 2020.

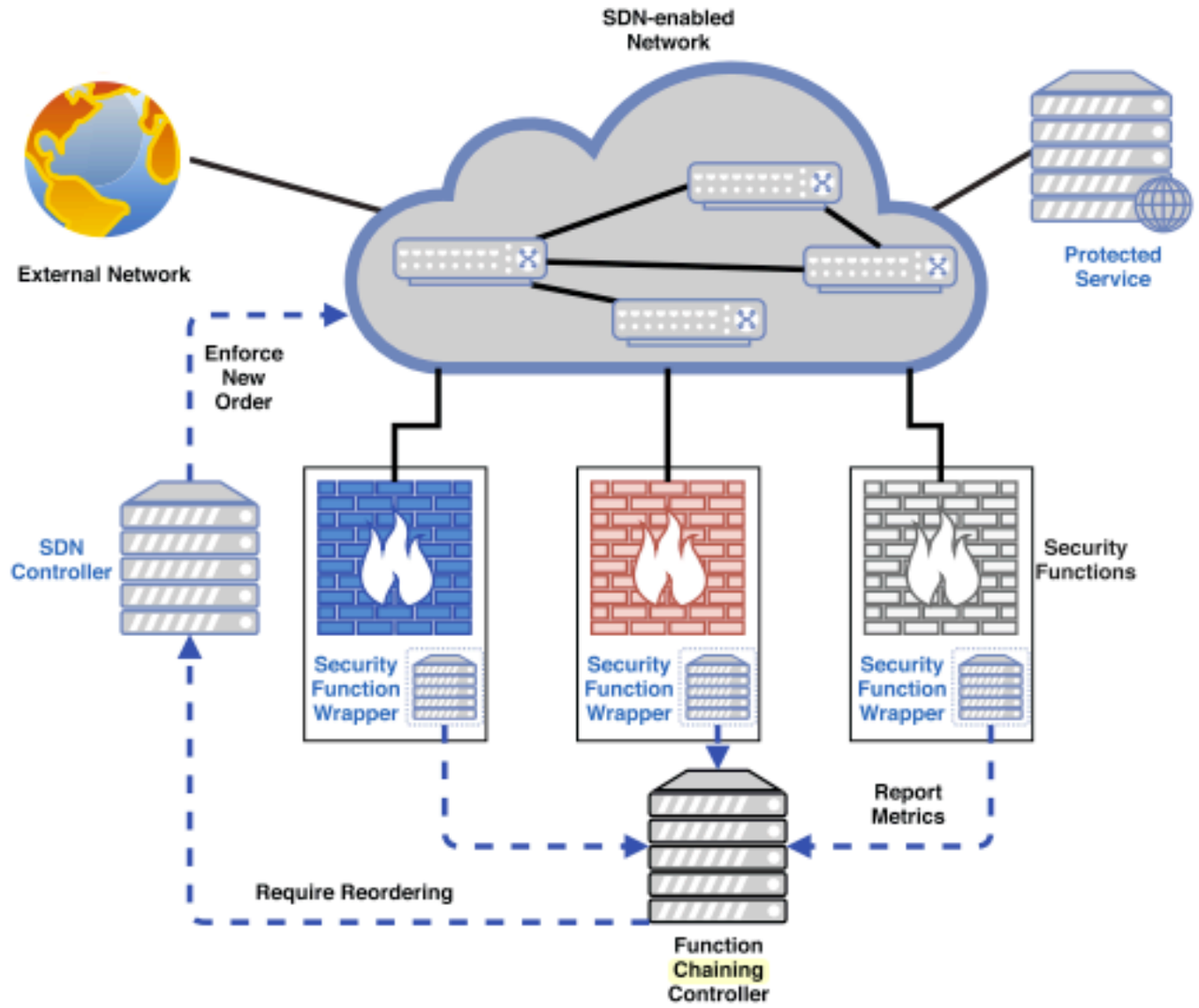
Footprints



- Introduces a signature per each SF (with low impact), in the NSH which is then checked by the final node in the SFC
- Mechanism to assure that SFs are implemented in securely and efficiently

M. Pattaranantakul, Q. Song, Y. Tian, L. Wang, Z. Zhang, and A. Meddahi, "**Footprints: Ensuring Trusted Service Function Chaining in the World of SDN and NFV**," in LNICST, vol. 305, 2019, pp. 287–301.

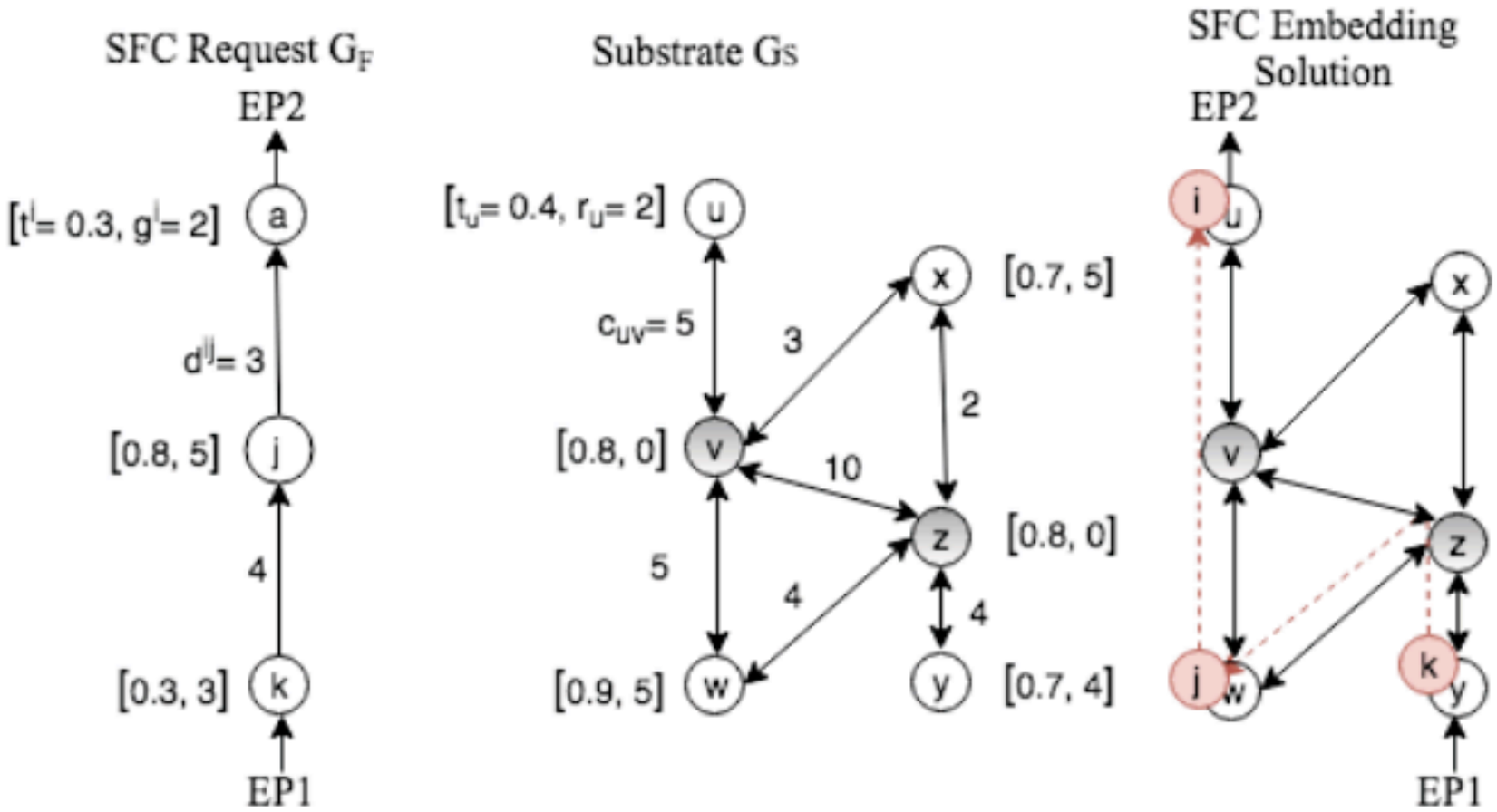
Attack-aware SFC reordering



- Mechanism to adapt dynamically the order of security functions
- Compliant with SDN

L. Ifflander, L. Beierlieb, N. Fella, S. Kounev, N. Rawtani, and K.-D. Lange, "**Implementing Attack-aware Security Function Chain Reordering**," in 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C), 2020, pp. 194–199.

Trust-aware SFC



- Each SF expresses its performance (computing, bandwidth) and security requirements
- The approach maps SFC chain to resources that match performance and security requirements
- The placement is formulated with MILP considering diverse constraints like trust, flow, capacity

N. Torkzaban, C. Papagianni, and J. S. Baras, "**Trust-Aware Service Chain Embedding**," in 2019 Sixth International Conference on Software Defined Systems (SDS), 2019, pp. 242–247.

Comments, Questions

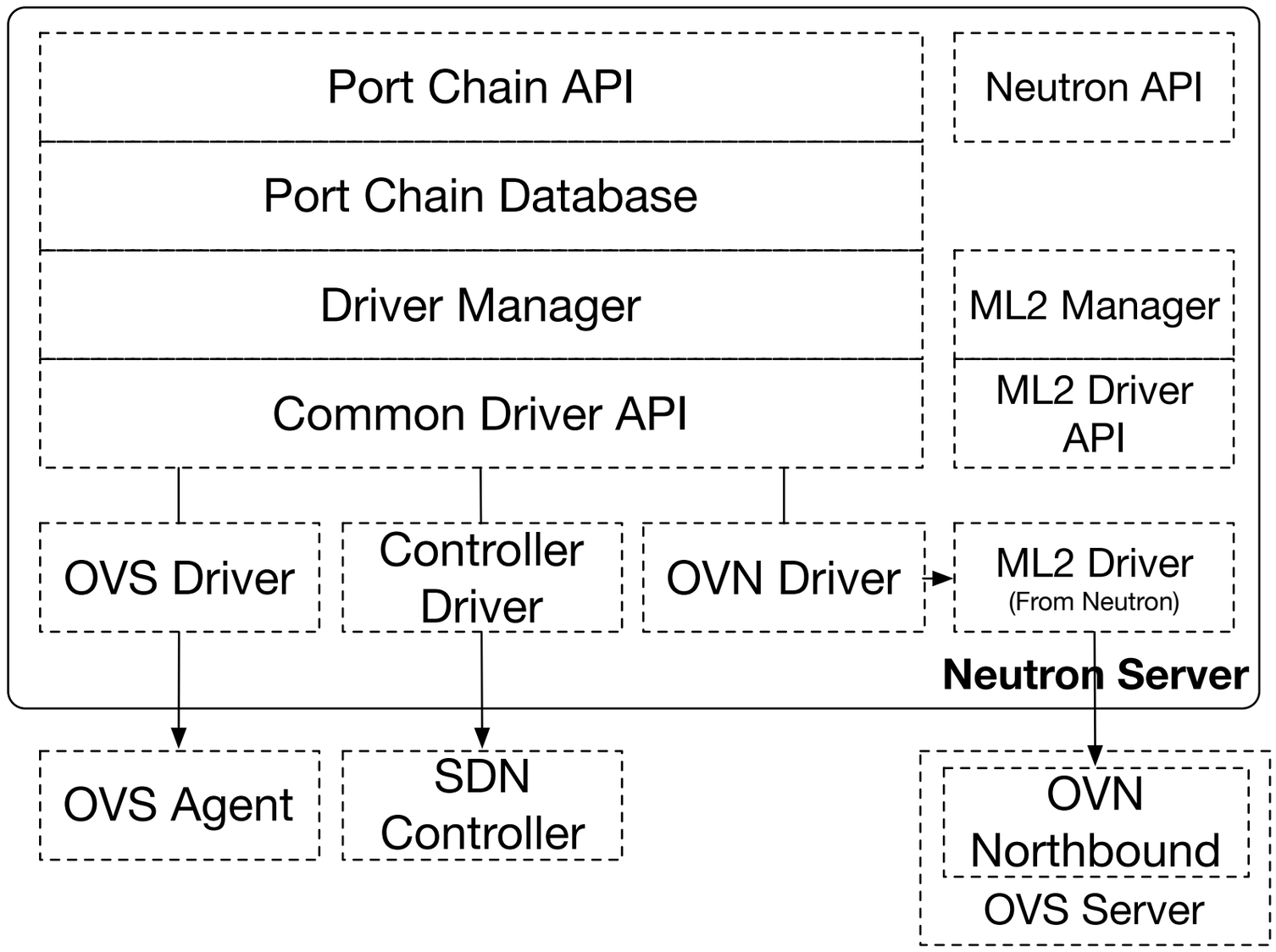
UNIVERSIDADE D COIMBRA

FACULDADE DE CIÊNCIAS E TECNOLOGIA

Virtualization paradigms & SFC

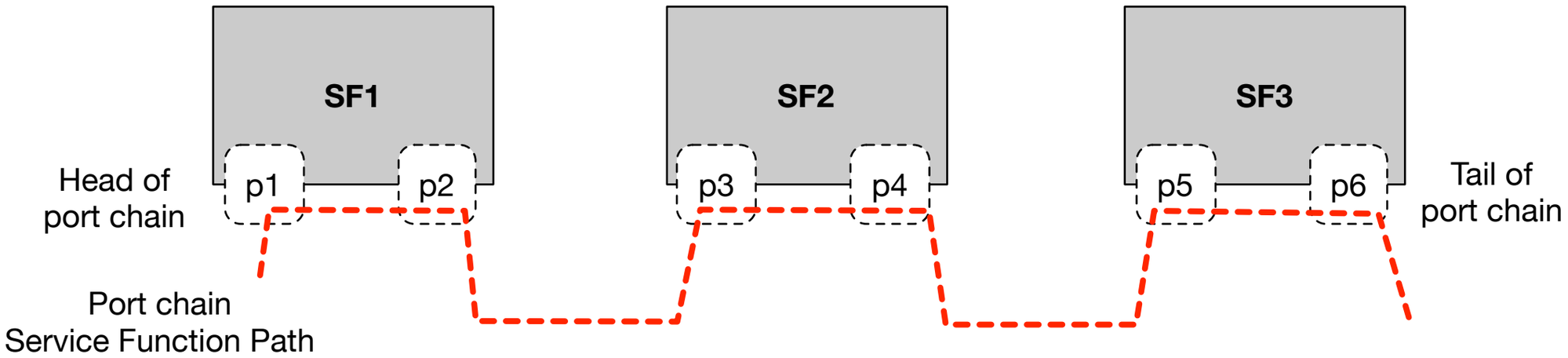
- OpenStack
- SDN controllers (OpenDayLight and its support for SFC)
- Cloud-Native Network Functions (CNFs)
- CNF and Kubernetes
- Service Mesh

OpenStack 1/2



- SFC supported through an extension to Neutron
- Relies on Open vSwitch

OpenStack 2/2

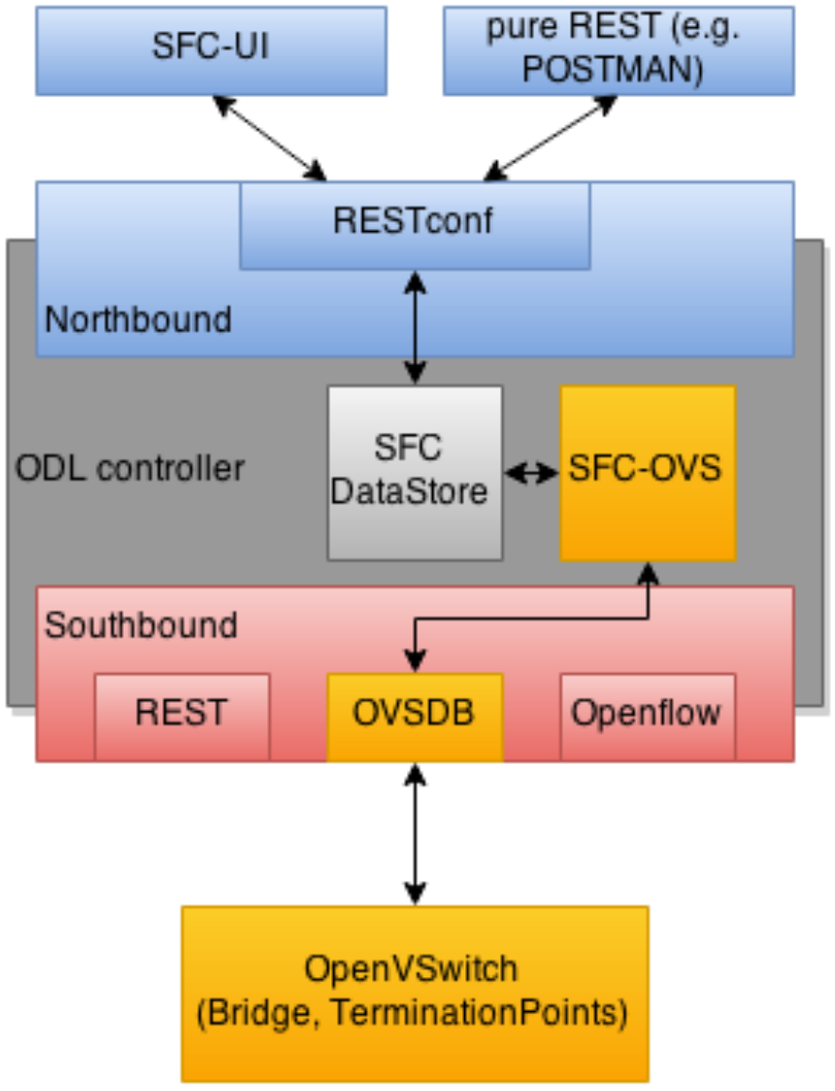


- Port chain is made of Neutron ports, with the sequence of SFs

- Flow classifiers to specific the flows that enter a port chain:
  - Can rely on L7 info (e.g. URLs)
  - Supports IPv4 and IPv6
  - Source and destination nodes info
  - Protocol

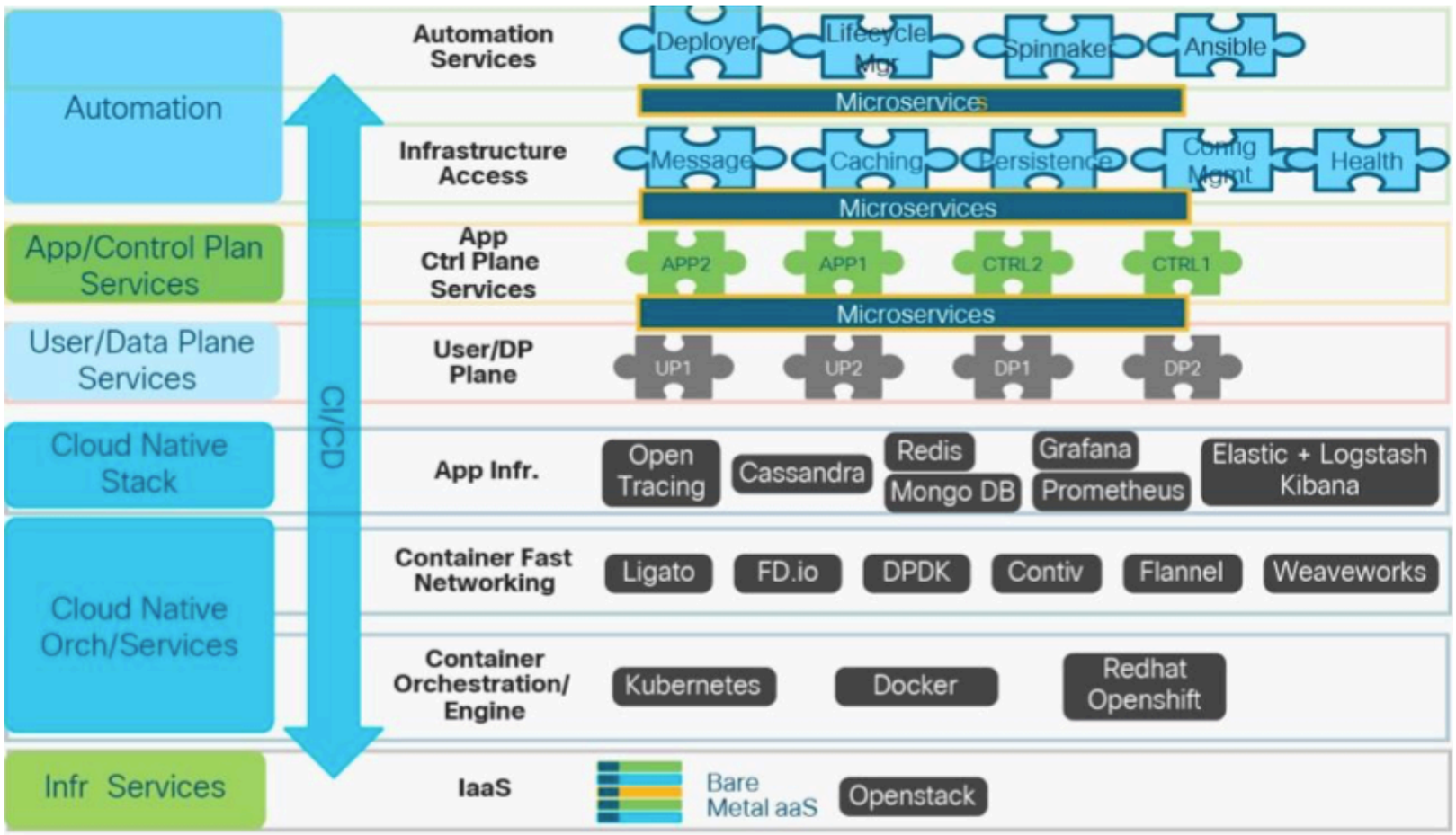
OpenDaylight & SFC

- ODL supports SFC to enable ordered list of network services
- ODL includes SFC-UI to configure the chaining process
- ODL also includes a REST API to configure the chaining
- The SFC can also be configured via Open vSwitch
- Classification of flows via OpenFlow or iptables
- More info here



Source ODL

Cloud-Native Network Functions



Source: Cisco

- CNFs are a software implementation of network functions, deployed in a cloud-native fashion (e.g. microservices)

Cloud-Native Network Functions

- The networking in CNFs considers:
  - **Data plane**, using solutions like FD.io, Vector Packet Processing (VPP), Data Plane Development Kit (DPDK), to support high-speed data-plane apps
  - **Management plane**, using solutions like Ligato.io, which allows the management of CNFs in a unified fashion (e.g., using gRPC)

CNFs and Kubernetes

- Multiple solutions are available for networking in K8s:
  - **Container Network Interface** (CNI) standardises the network interfaces in k8s (interoperability between multiple plugins)
  - Calico is the open-source reference for networking
  - Plugins like ovn4nfv-k8s enable support for SFC in K8s
    - includes a SFC manager in the master node of k8s
    - Includes SFC configuration in the minion nodes

Service Mesh & SFC

	NFV/SDN	Service Mesh
Orchestration	ONAP, OP-NFV, OpenNetVM	Kubernetes + Istio
Protocol Layer	L2/L3/L4 + NSH	L7
Transport	GRE/MPLS/VxLAN + vSwitch	HTTP, gRPC, Websocket + Envoy
Communication Pattern	Long-lived connections, proxy mode (passthrough)	Short-lived request-response traffic (server mode)
Service Routing/ Chaining	Network Service Header (NSH) + ToR/vswitch routing	HTTP Headers/Cookies + Istio HTTP routing
QoS	Per-flow	Per-request
Language/Platform	dataplane: C/C++ + DPDK, ctrl plane: Java	dataplane: Node.js + OS network stack, ctrl plane: Go
Performance	Rate: 10–100 mpps, Latency: < 5 millisec/packet	Rate: 100+ Kreq/sec, Latency: 10+ millisec/req

G. Antichi and G. Rétvári, “**Full-stack SDN: The next big challenge?**,” SOSR 2020 - Proc. 2020 Symp. SDN Res., pp. 48–54, 2020.

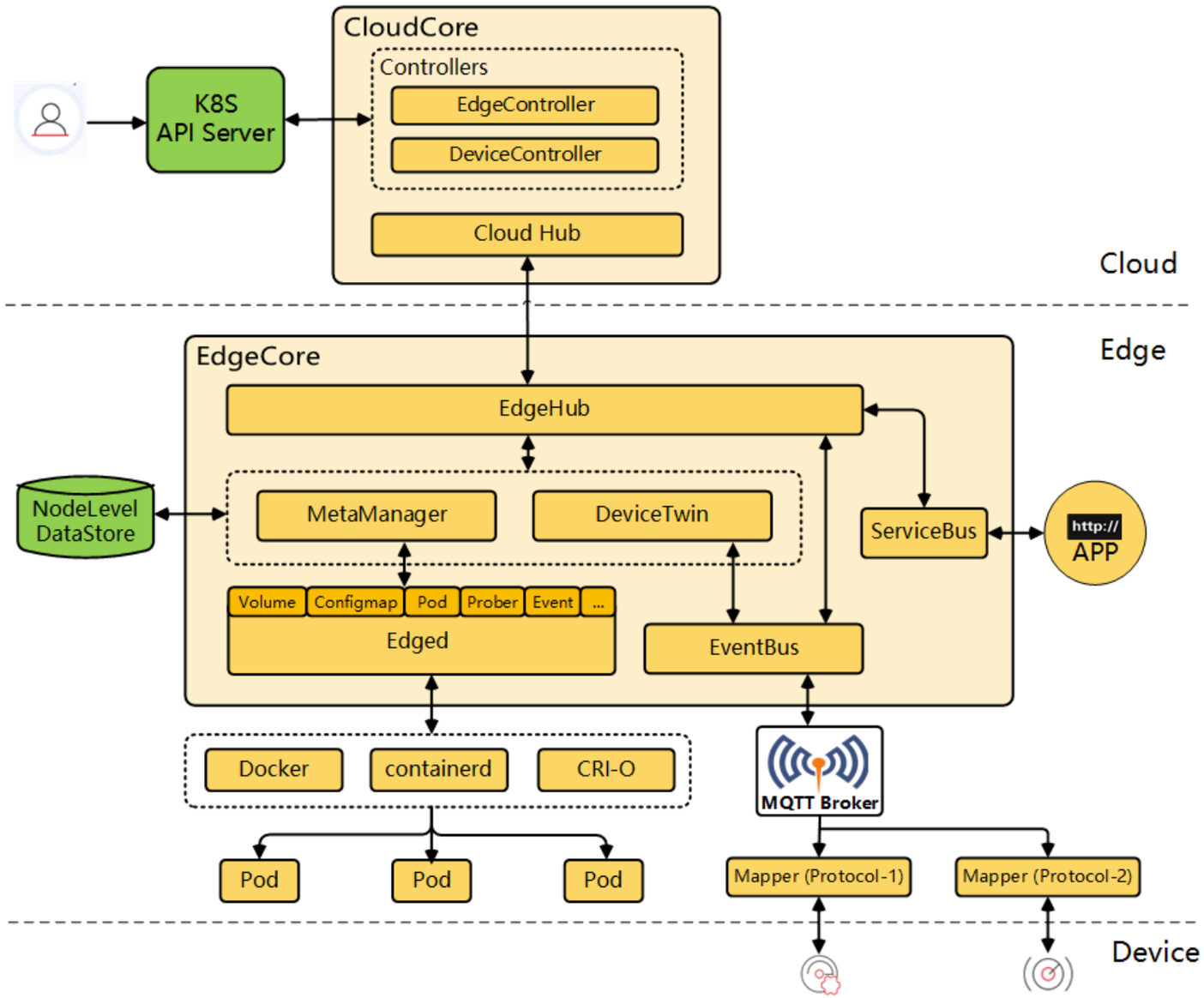
Comments, Questions

Edge and cloud continuum

- KubeEdge
- MicroK8s
- K3S
- Open Network Automation Platform (ONAP)

Z. Tao et al., “**A Survey of Virtual Machine Management in Edge Computing**,” Proc. IEEE, vol. 107, no. 8, pp. 1482–1499, 2019.

KubeEdge



Source KubeEdge

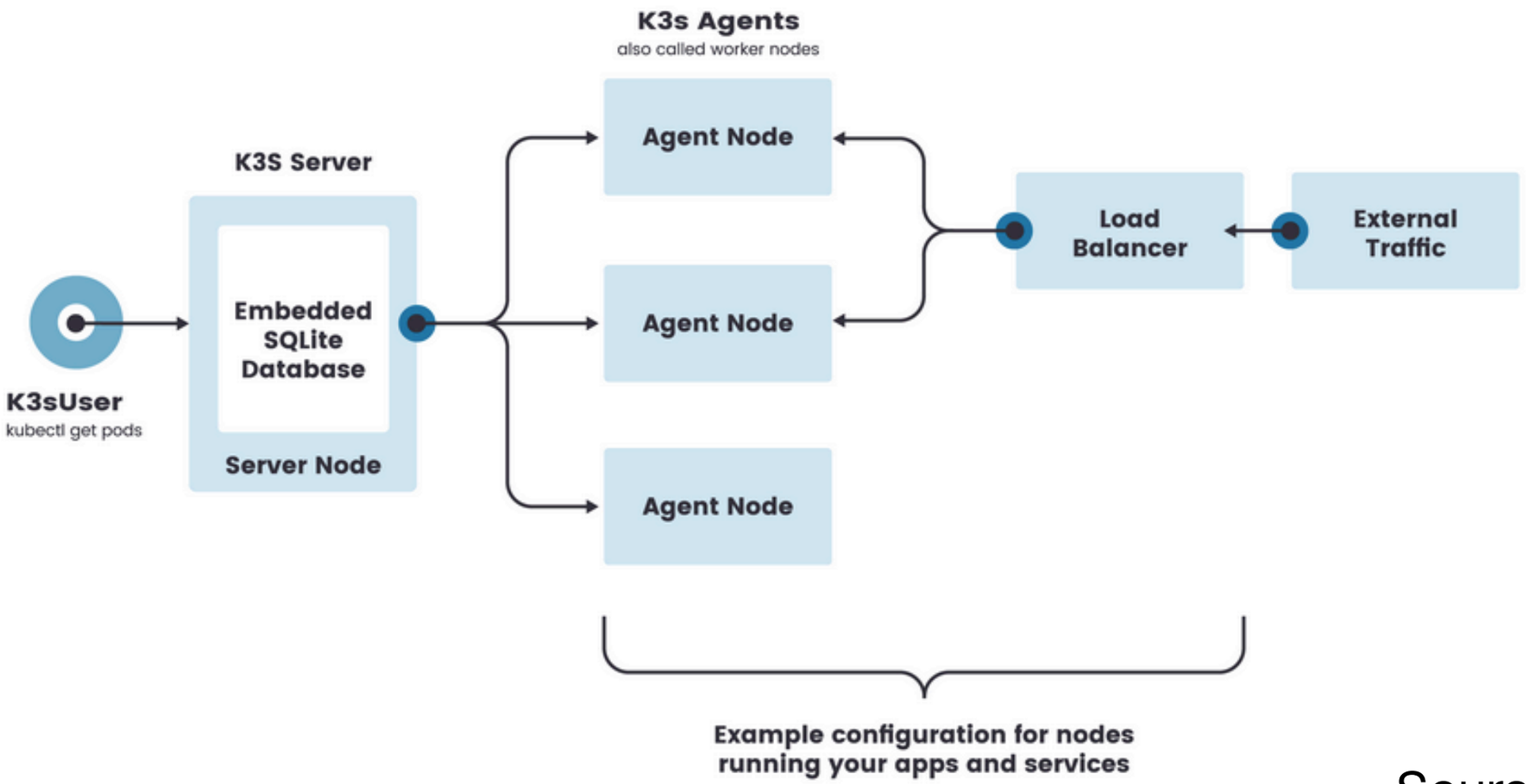
- Extends containerised application orchestration capabilities to hosts at the Edge
- Requires a “master node” running at the cloud side to manage multiple edge nodes
- Compliant with K8S

MicroK8S

https://microk8s.io/

- Lightweight Kubernetes deployment (can run on Raspberry PI)
- Suitable for Edge and IoT deployments
- With high availability mechanisms
- Runs on multiple platforms (and is supported by Canonical)
- Supports cluster mode with low maintenance (zero-ops)

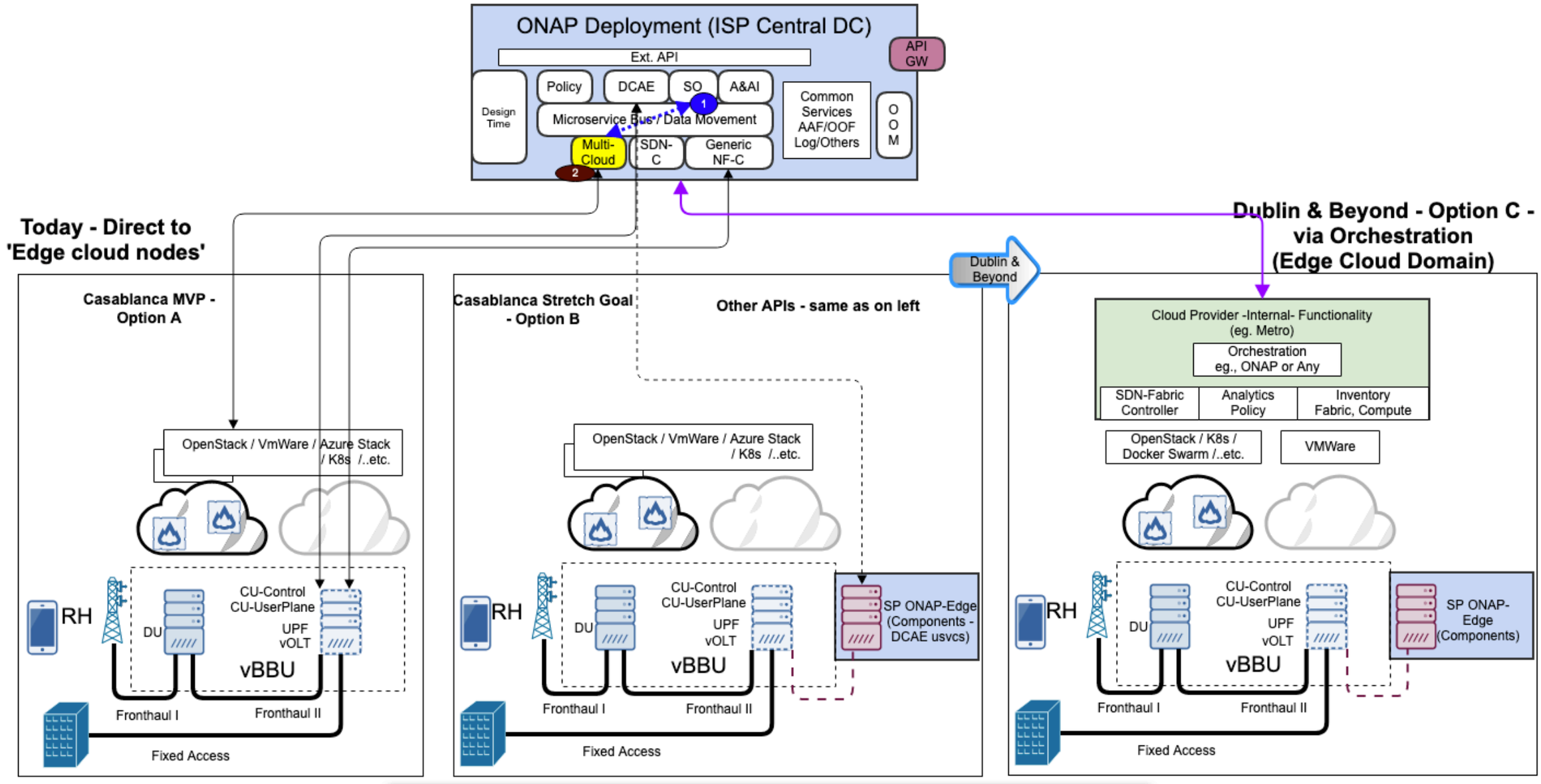
K3S



Source k3s.io

- Lightweight Kubernetes deployment but is flexible to support High-availability mechanisms (e.g., Postgres SQL)
- Suitable for Edge and IoT deployments
- Promoted as a CNCF project, and supported by Rancher

ONAP



Source ONAP

- Integrates several platforms (OpenStack, Docker)
- Aligned with the ETSI Mobile Edge Computing (MEC) platform
- Enables the orchestration of resources at the edge through the SP ONAP-Edge components

Comments, Questions

Trust in SFC

- Trust can be modelled and trustworthiness can be measured

- Attestation allows to build reputation and the level of trustworthiness in a component

- IETF in RATs is specifying an architecture to allow interoperable attestation between different solutions

- OpenID Connect is a promising solution for federated identification management of services across several domains

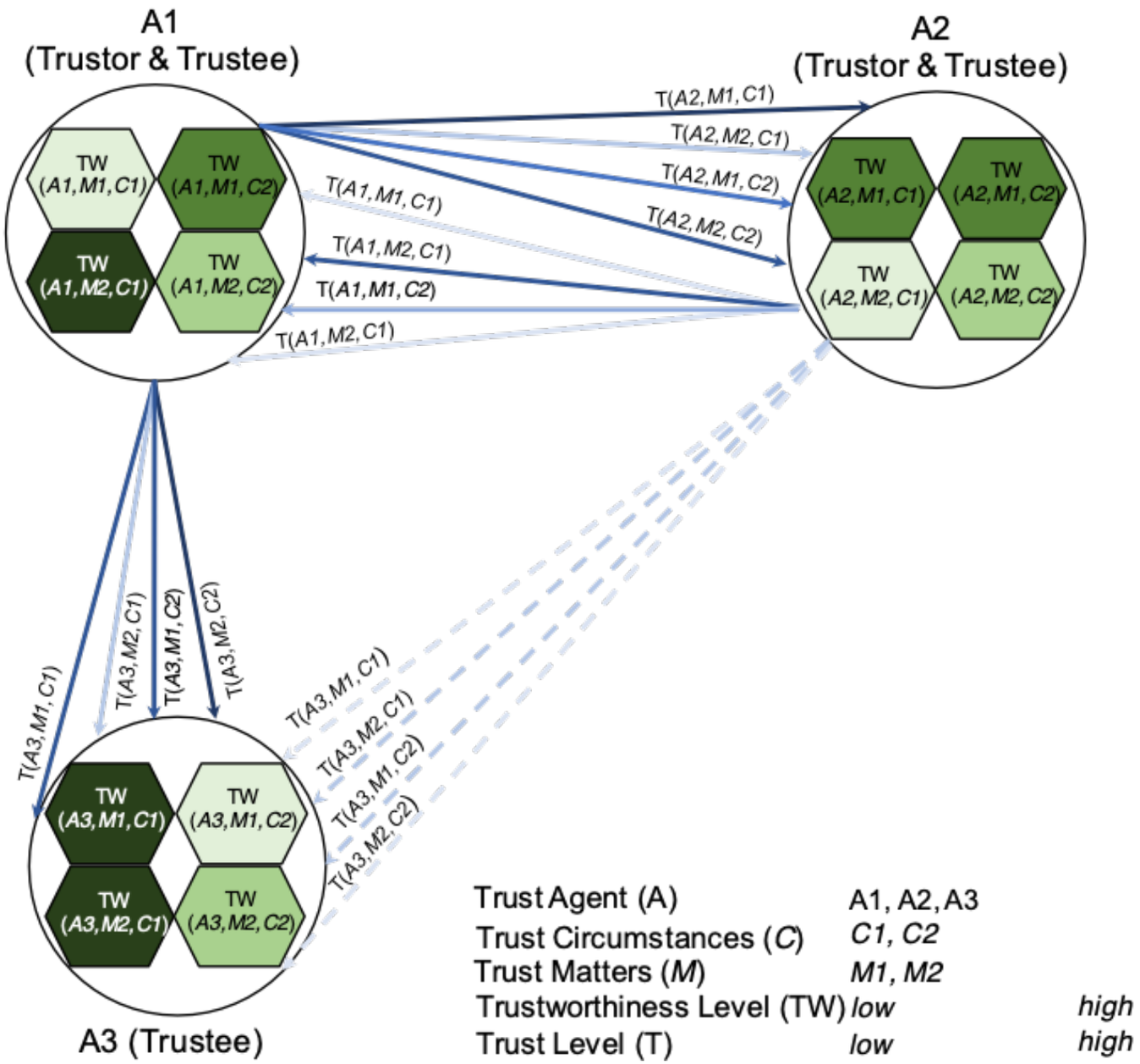
Trust & Trustworthiness

- Concepts broadly studied in many different areas:
  - Sociology, economics, psychology…
- Trust:
  - Reliance on a system or service that it will exhibit the expected behaviour (including many perspectives)
  - **Trust Level:** estimated probability of this reliance
- Trustworthiness
  - Worthiness of a system or service for being trusted
  - Assessed based on evidences
  - Complex and potentially subjective!

Human
Trust & Trustworthiness

- Changes over time and can be highly subjective…

Modelling trust

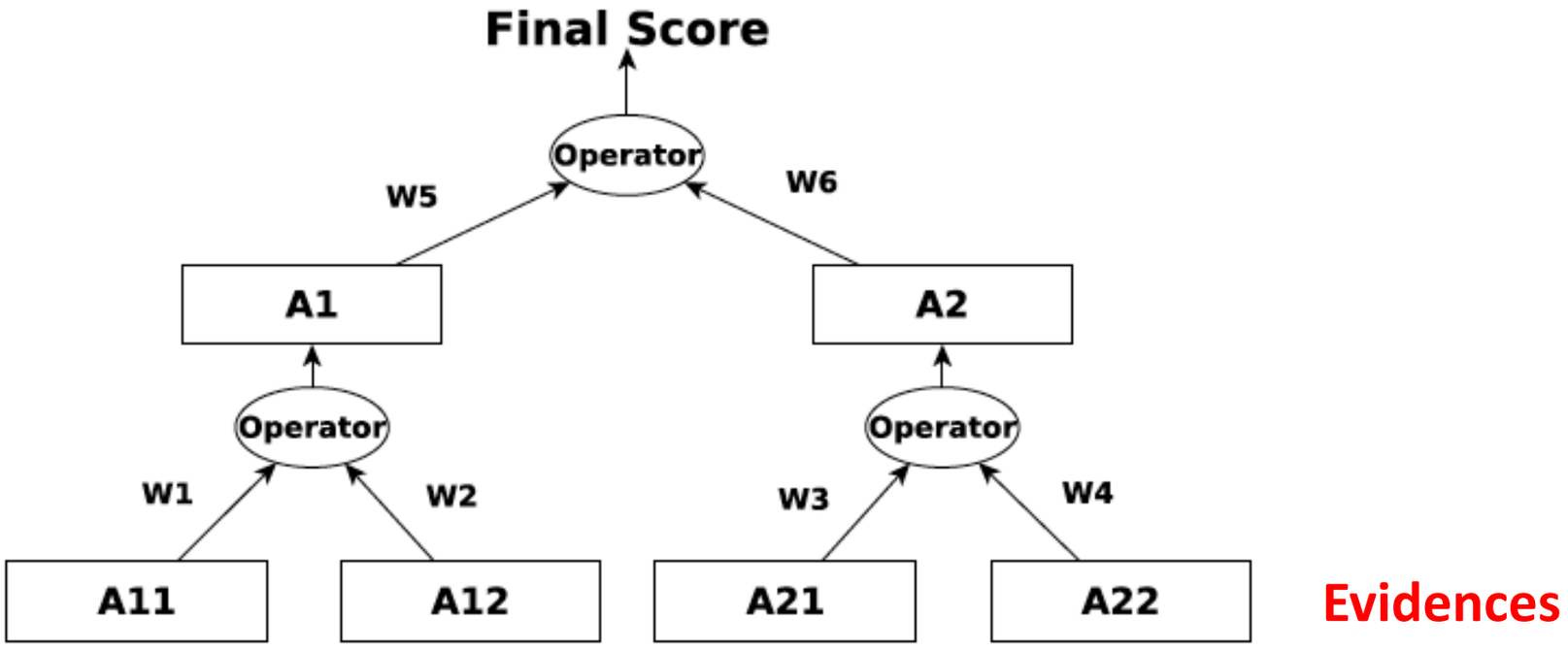


Trustworthiness Properties

- Trustworthiness is frequently seen as a security aspect
  - It is trustworthy if it is secure!?
- We consider it a more general notion
  - Even broader than dependability…
- Requires identifying and evaluating all relevant measurable characteristics that may influence reliance
  - Functional and non-functional
- Security, privacy, dependability, performance, fairness, transparency, stability…

From Properties to Scores

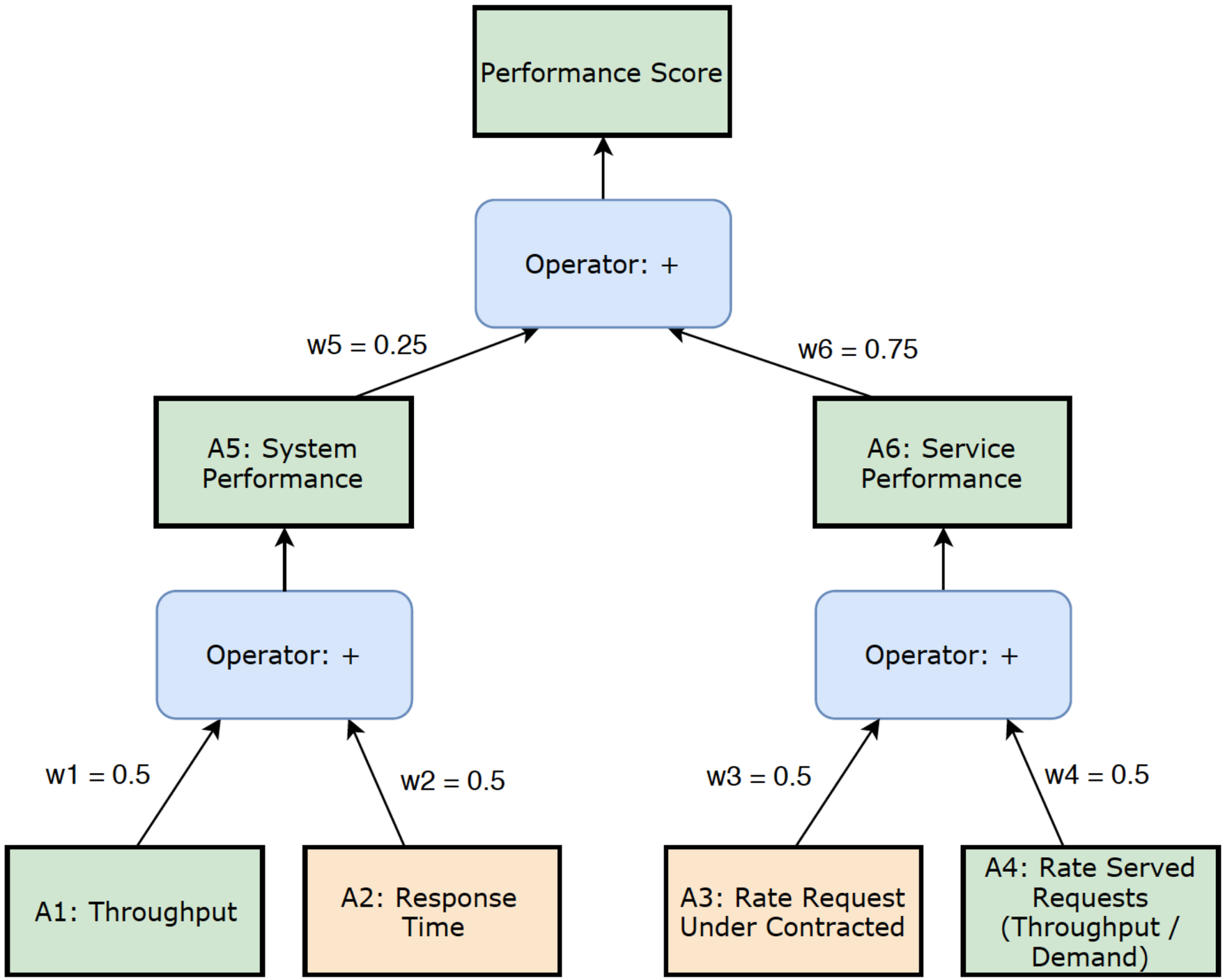
- A property may require several different scores
  - We need a model to map properties to scores



- Calculating a score requires collecting raw data
  - Trust evolves over time…

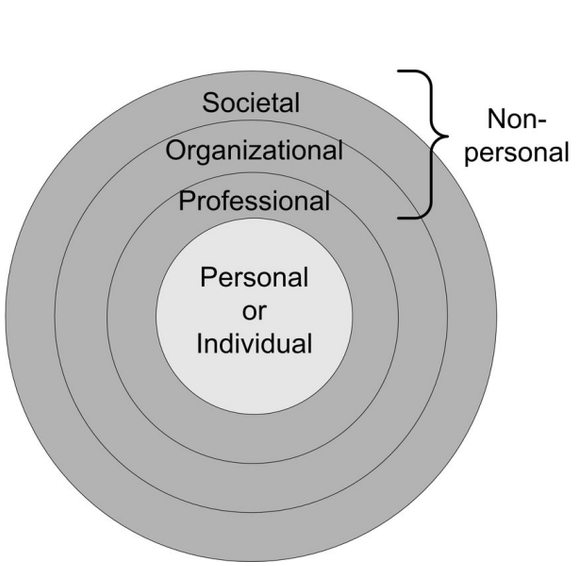
- Coherence *vs* Stability *vs* Fairness *vs* …

Example: Performance

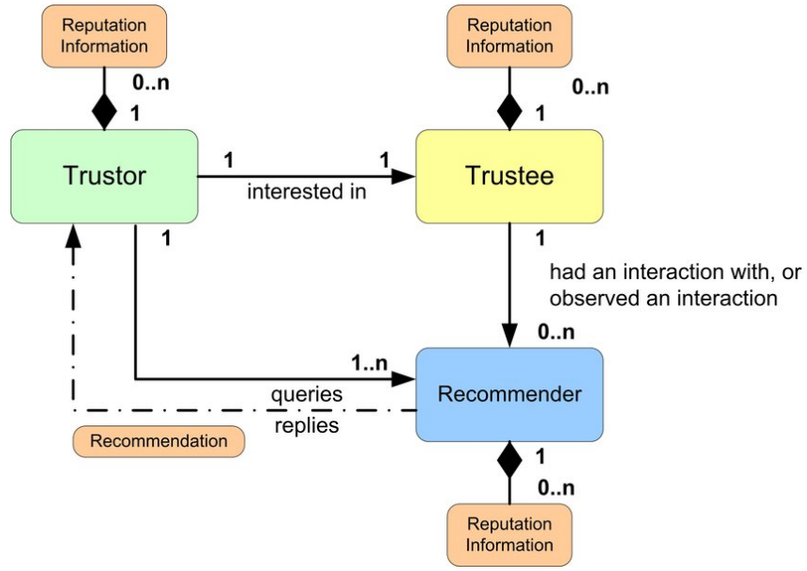


Reputation Systems

- Key Concepts
  - Trust
  - Risk
- Reference Models
  - Reputation Contexts
  - Reputation Systems
- Aggregation Methods
  - Counting
  - Discrete
  - Probabilistic
  - Fuzzy
  - Flow



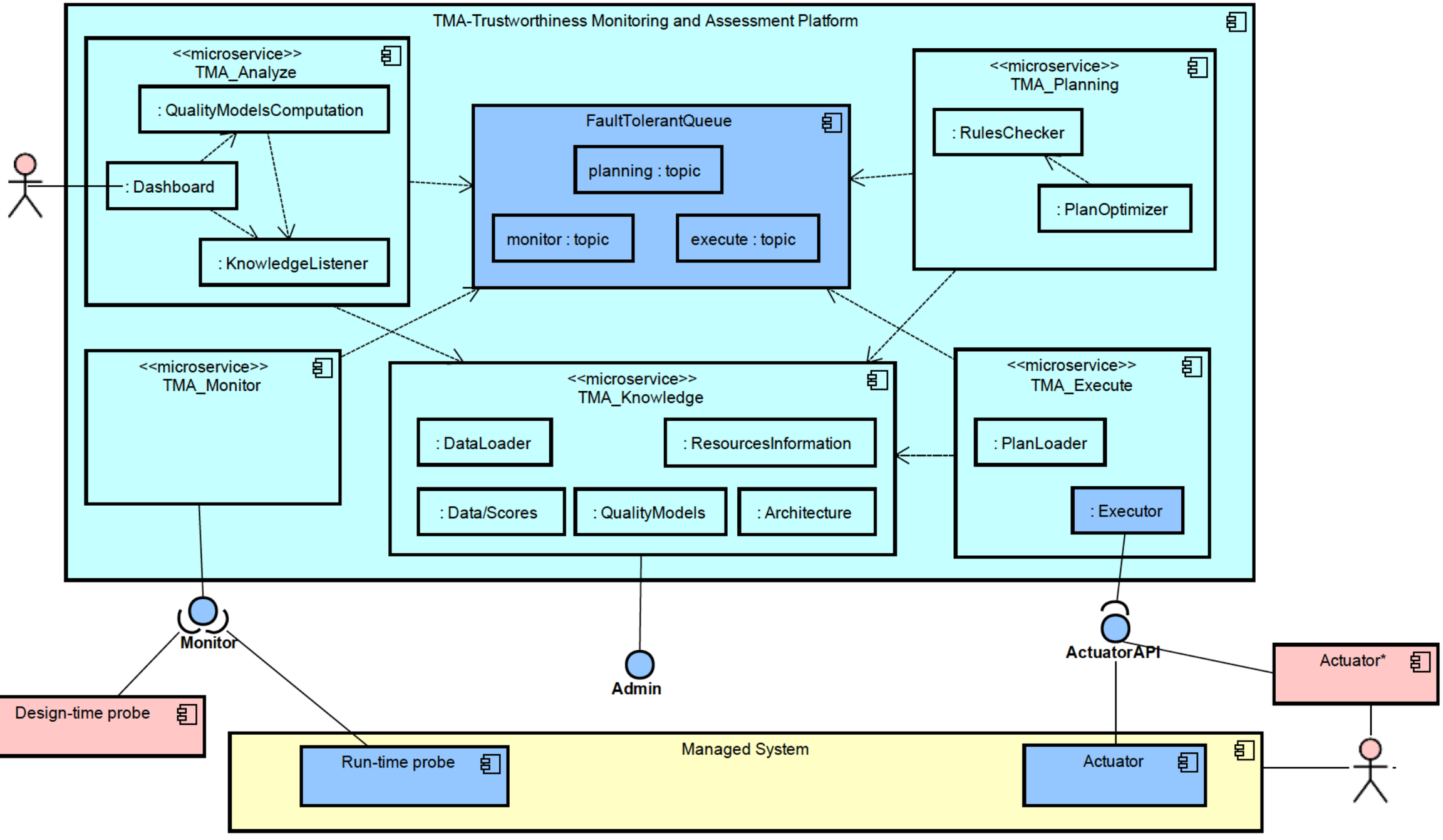
Reference model for reputation contexts (from [1])



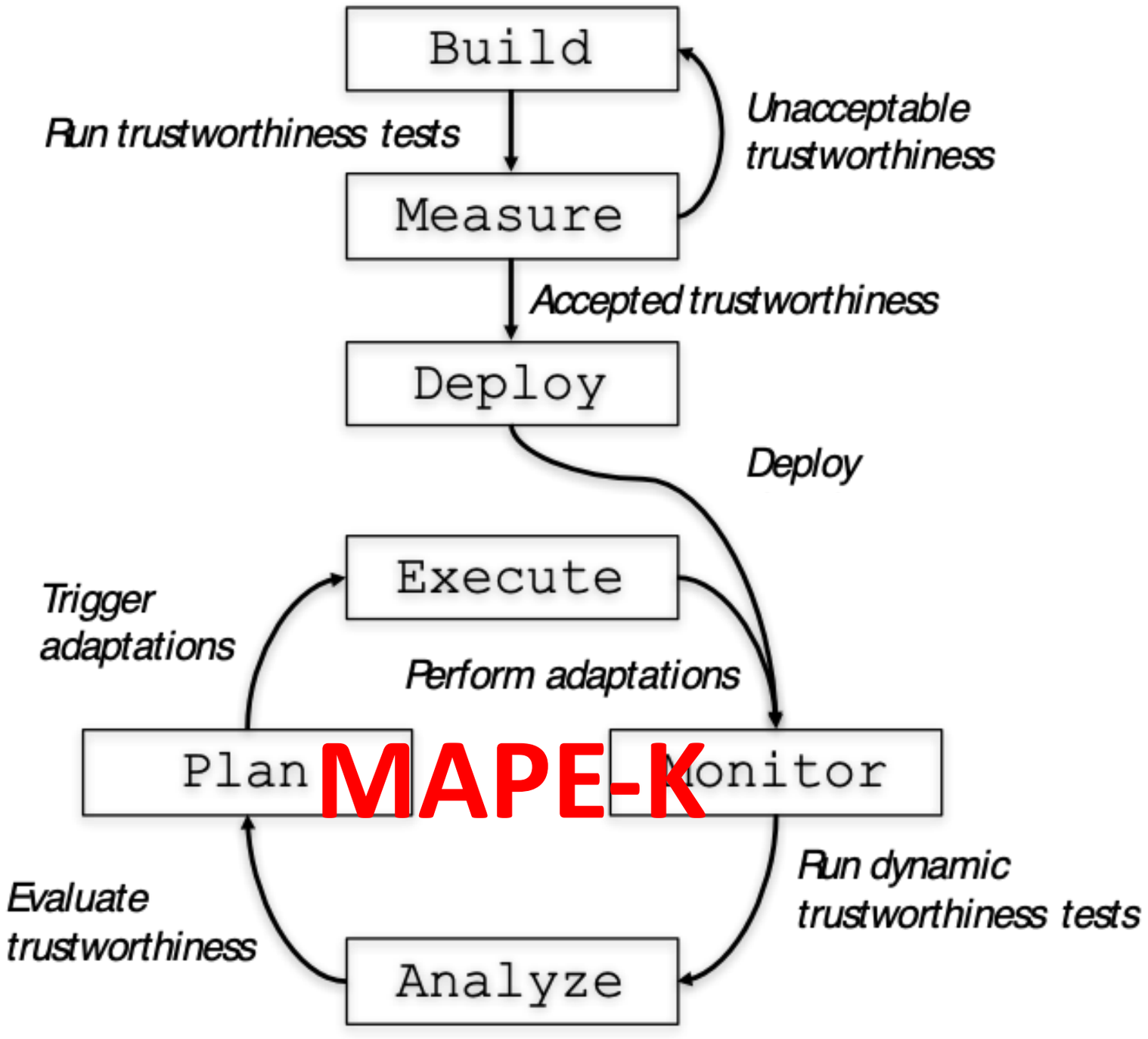
Reference model for reputation systems (from [1])

[1] Vavilis, Sokratis & Petković, Milan & Zannone, Nicola. (2014). A Reference Model for Reputation Systems.

ATMOSPHERE Platform

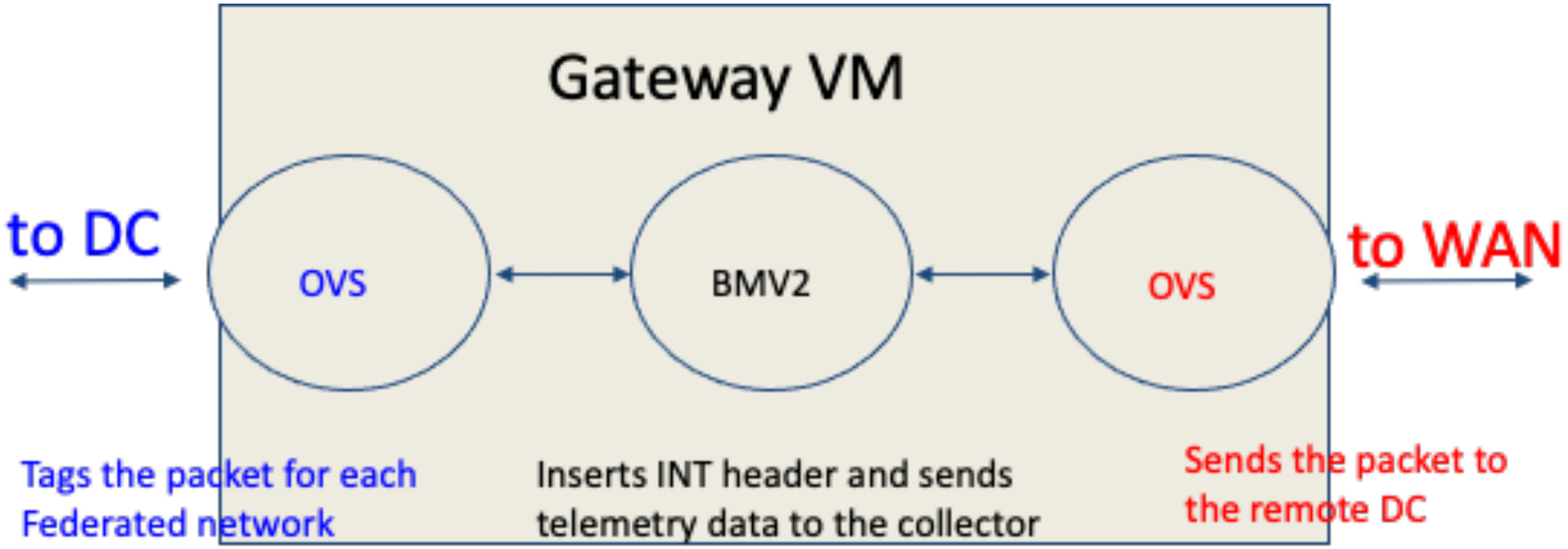


Trustworthiness Life-Cycle



Federated Infrastructure

- Distributed Network Federation Architecture
- Integration with Fogbow [http://www.fogbowcloud.org/]
- Kubernetes over Distributed Network Federation

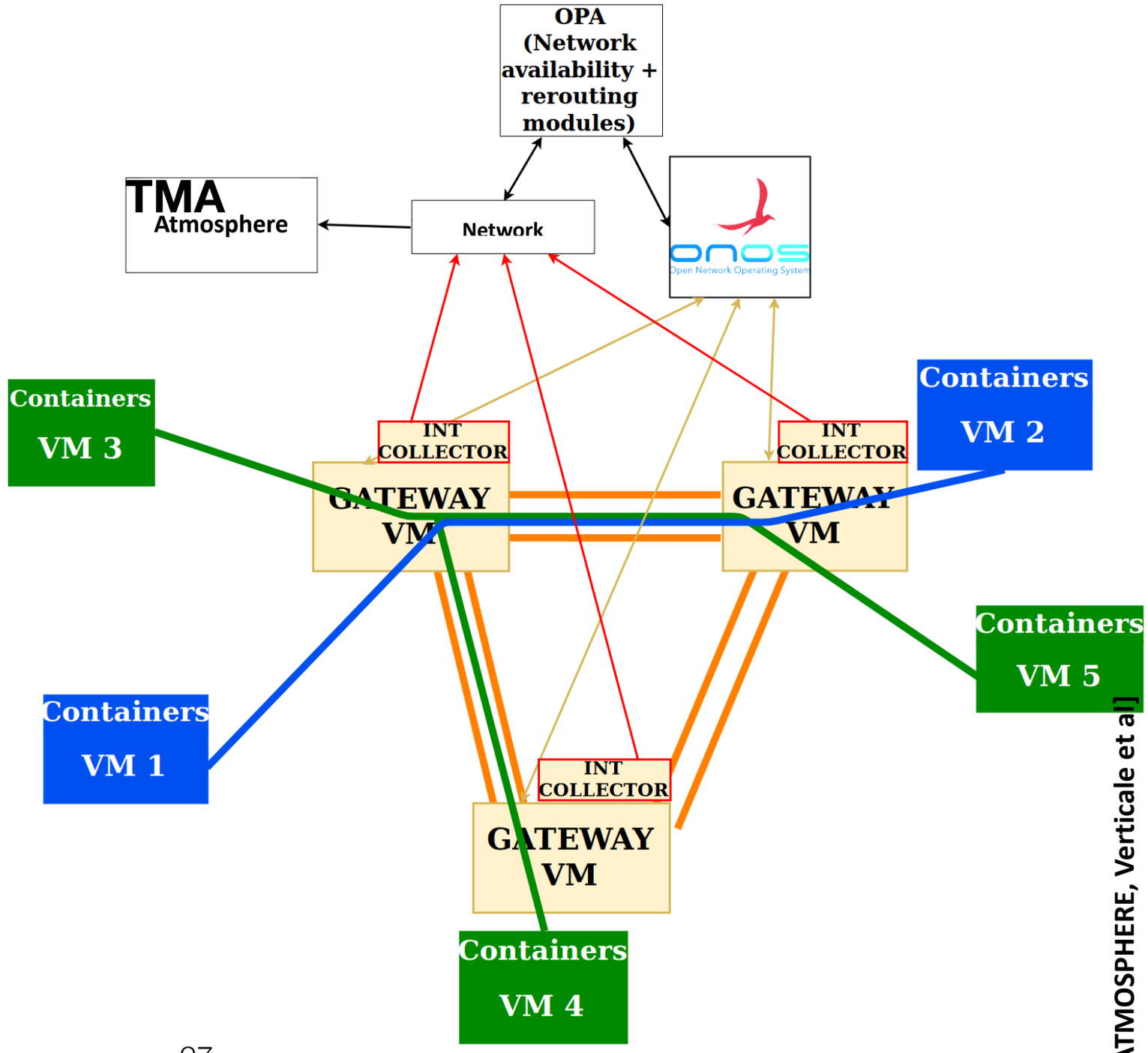


- P4 software switch (BMV2) enables the collection of the link latency metric.
- Written a P4 program that is executed by BMV2 (or SmartNIC)

[ATMOSPHERE, Verticale et al]

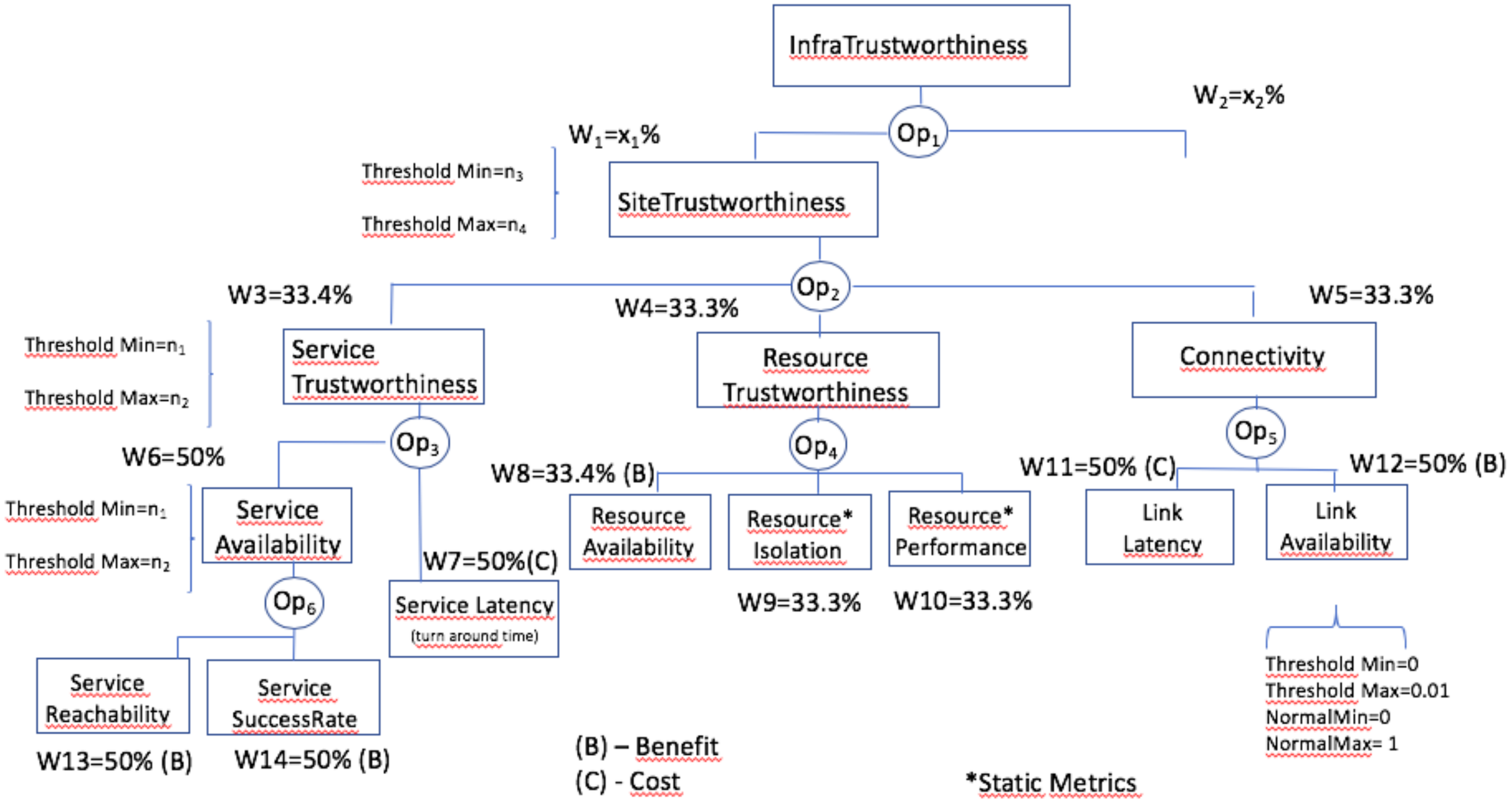
Architecture of the monitoring and (Re)Routing Module

- Availability metric based on ONOS link probing
- Link latency metric is collected if P4 hardware is available at the sites
- OPA calculates the best route based on metrics and requirements and pushes it to ONOS



https://www.atmosphere-eubrazil.eu/

Infrastructure Quality Model



[ATMOSPHERE, Verticale et al]

https://www.atmosphere-eubrazil.eu/

Comments, Questions

Attestation

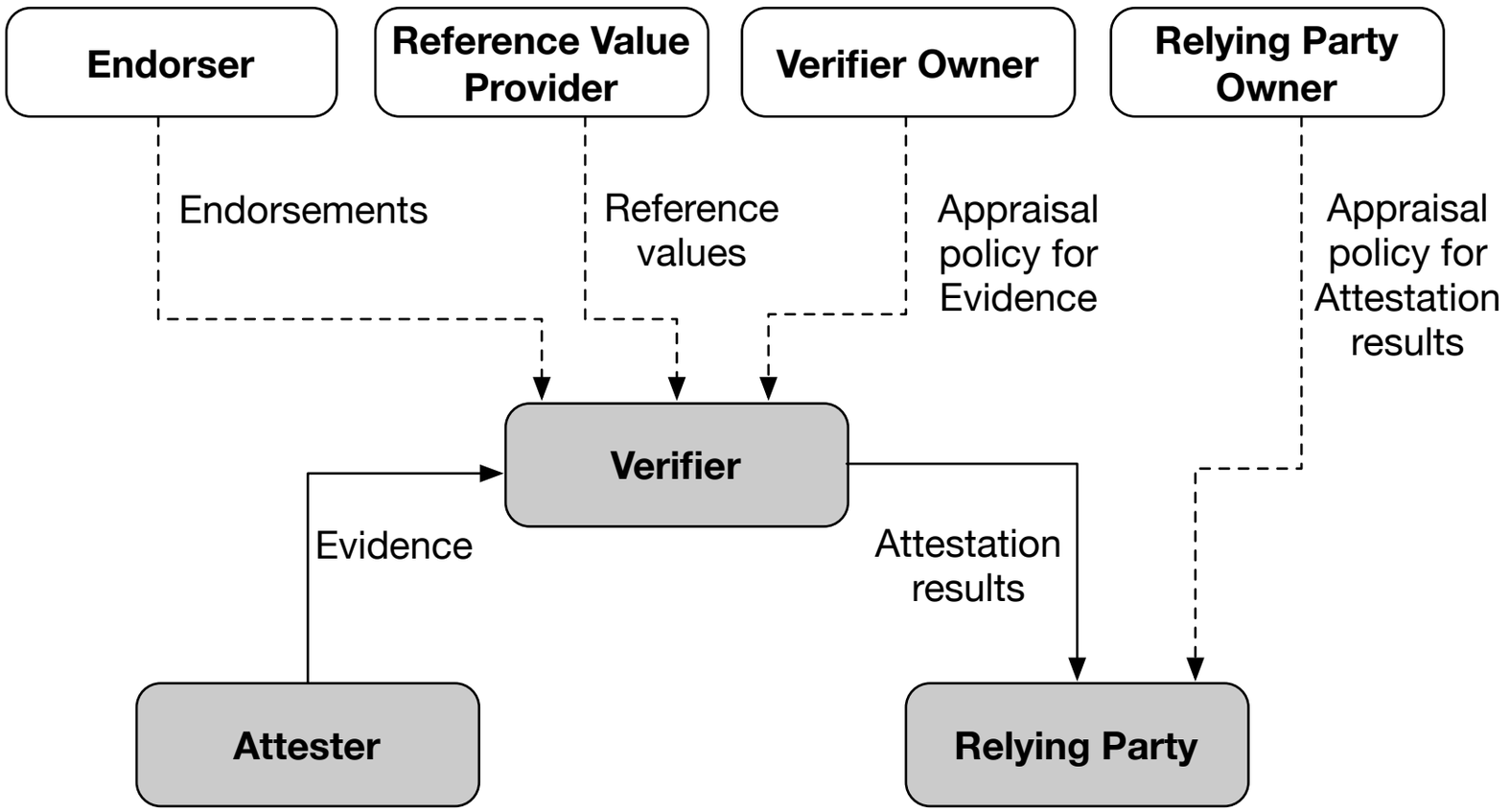
- “Process on which is provided evidence that a given entity is trustworthy“, as per IETF

- Being standardised at IETS in the Remote Attestation Procedures (RATS) WG:

  - Interoperability between system components from different manufactures and origins

  - Leverage from domain specific attestation mechanisms like Trusted Computing Group (TCG), Trusted Platform Module (TPM), Fast Identity Online (FIDO) Alliance

Remote Attestation per IETF 1/3



Data flow model IETF/RATS

- Attester creates evidence and conveys it to the Verifier
- Verifier uses the evidence, and info from others like the endorsements, reference values, according the policy to assess trustworthiness
- The attestation results are sent to the relying party, which relies on its appraisal policy to make decisions.

Remote Attestation per IETF 2/3

- Evidence is related with the environment on which is performed the attestation and can be performed in:
  - Trusted Execution Environment (TEE)
  - Embedded Secure Elements (eSEs)
  - Trusted Platform Modules (TPM)
  - BIOS firmware or others
- Composite devices need to attest the different devices and aggregate it in order to convey evidence to verifier
- Different models can be followed to convey Evidence:
  - Passport — Attester sends Attestation result to the Relying party, after receiving from Verifier
  - Background-check model - Evidence is sent to the Relying party that conveys it in background to the verifier

Remote Attestation per IETF 3/3

- The relying party trust a verifier through a trust model that can rely on Public Key, Certificates of the Verified, or implicitly

- Attester high need to trust other entities to which the Evidence is conveyed if sensible info is provided (e.g. Personal Identifiable Info)

- Evidence must be conveyed through a protocol that assures authentication and integrity protection

- Remote attestation can be performed through different models:
  - Challenge/Response Remote Attestation (CHARRA implementation)
  - Uni-Directional Remote Attestation
  - Streaming Remote Attestation (kind of public subscribe approach)

Federated Authentication 1/2

- OpenID Connect is built on top of OAuth 2.0 and works as an identity layer
- OAuth 2.0 is an authorisation framework for access delegation, and is considered the standard to secure the access to APIs
- OpenID Connect, along with SAML 2.0 Web SSO, Central Authentication Service (CAS) enable identify federation
- OpenID Connect enables the implementation of Open Policy Agent (OPA) for attribute-based access control

Prabath Siriwardena, "**OpenID Connection in Action**," version 4, Manning 2021.

Federated Authentication 2/2

- OpenID Connect relies on JSON Web Tokens to carry authentication and authorisation assertions (i.e. claims)

- Specifies standard claims and provides flexibility to use others specified by users

  - The standard claim includes info about name of user, picture, surname, email, gender, birthdate, phone number, address and other items

- Solutions like DEX promote the OpenID connect adoption, to enable the identity management in a federated fashion

Case Studies

- Based on projects we are working on or have ended recently

Smart Cities

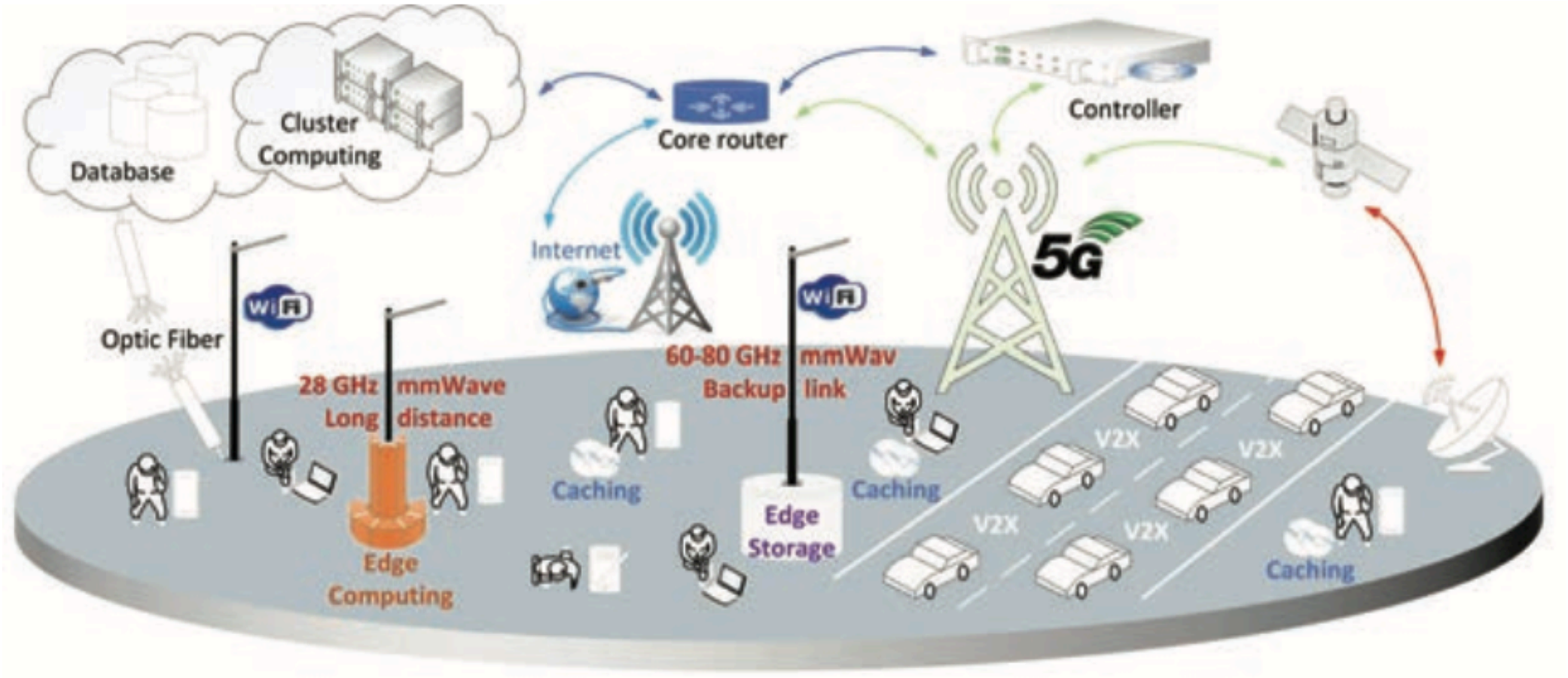
- Enhance content delivery
- To enhance safety in urban environments (pedestrian safety)
- To support deployment with high density

SNOB5G 1/3

- Scalable and Self-Optimized Wireless Network Backhauling for 5G (https://snob-5g.com/)

- Includes an architecture based on multiple technologies:
  - Network coding to enhance security and resilience
  - mmWave for backhaul connections
  - 5G in an urban scenario
  - SDN to manage network resources
  - OpenStack and K8S to manage computation and storage resources
  - Content Delivery Networks (CDN) to optimize content delivery

SNOB5G 2/3



A. Cohen et al., “**Bringing Network Coding into SDN: Architectural Study for Meshed Heterogeneous Communications**,” IEEE Commun. Mag., vol. 59, no. 4, pp. 37–43, Apr. 2021.

SNOB5G 3/3

- SNOB5G employs service chains in various scenarios (edge, cloud)

- A mechanism was specified to enhance resilience of Service chains, through replication:
  - Relying on a formal grammar to describe Service chains
  - Identification of replicas for different Virtual Functions
  - Employs Integer Linear Programming to prioritise Service Chain in nodes with higher availability
  - Heuristics are devised for complex scenarios

D. P. Abreu, K. Velasquez, L. Paquete, M. Curado, and E. Monteiro, "**Resilient service chains through smart replication**," IEEE Access, vol. 8, pp. 187021–187036, 2020.

OREOS 1/2

- Orchestration & Resource optimisation for reliable and low-latency services (OREOS) https://oreos.pt/

- OREOS architecture aims to enable orchestration and resource management for services with low-latency requirements:
  - Relies on ONAP framework for orchestration (edge, cloud)
  - Relies on the ONAP Optimisation Service Design Framework (OSDF) to enable models and policy driver optimisations
  - Relies on Open-RAN
  - Relies on SDN

OREOS 2/2

- The use cases of OREOS include:
  - City security in particular for pedestrian safety within vehicle mobility
  - End-to-end network slicing to enable different service providers within efficient and secure levels
  - Self-Organised networks (SON) for increased resilience
  - Intent Based networking to facilitate the management of rules and policies in SDN networks

Critical Infrastructures

- Smart Grids (Energy sector)

- Water management sector

Smart5G Grid

- Enable a 5G network in non Public network mode for smart Grids

- Enhance monitoring quality and resilience of 5G networks for critical systems, leveraging from ultra reliable low latency communications (URRLC) and massive machine type of communication (mMTC)

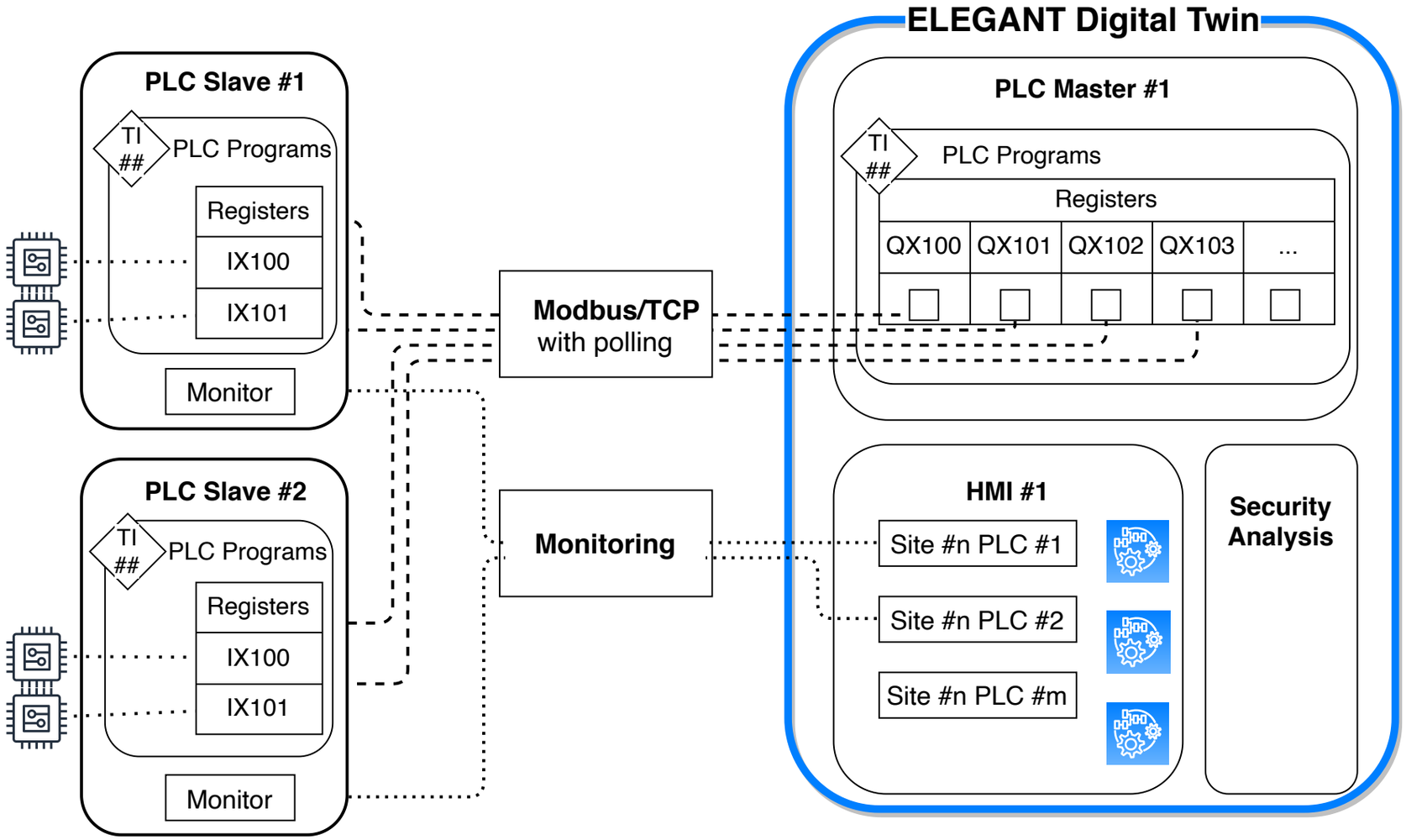
- Enhance security in SmartGrids, through the concept of Digital Twins

ELEGANT 1/3

- Enabling Security with Digital Twins (ELEGANT)
- Funded through Fed4Fire Open call
- The ELEGANT goals were:
  - Enhance Shadow Security Units (SSU)
  - Enhance SSU towards Digital Twins
  - Enable efficient and scalable data collection mechanisms for security analysis
  - Models for Denial of Service (DoS) attacks

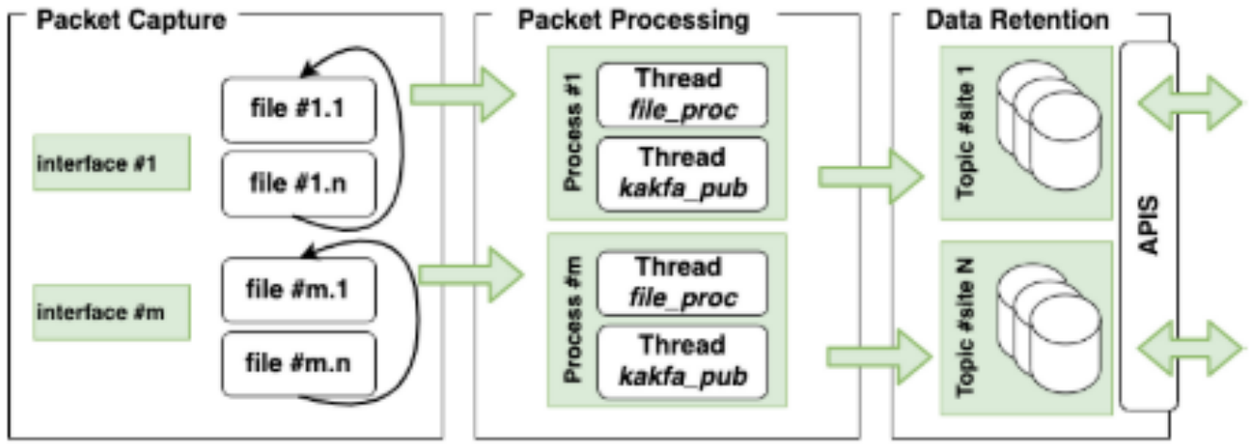
Bruno Sousa et al, "**ELEGANT: Security of Critical Infrastructures with Digital Twins**", 2021

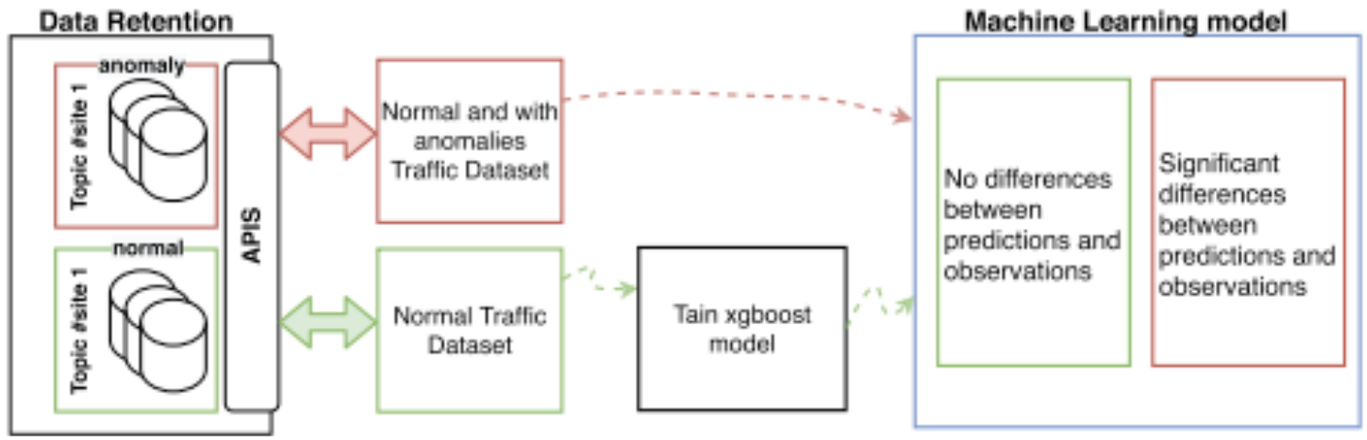
ELEGANT 2/3



- The ELEGANT Digital Twin:
  - Supports the Programmable Logic Controller (PLC)
  - Employs Human Machine Interfaces (dashboards)
  - Security components
  - “Orchestrations” several critical components (PLC Slave nodes connected to real sensors)

ELEGANT 3/3





Traffic	Precision	Recall	F1-Score
Normal	1.00	0.96	0.98
Attack	0.98	1.00	0.99

- Includes mechanisms to detect Denial of Service (DoS) attacks, relying on the Extreme Gradient (xboost)

Comments, Questions

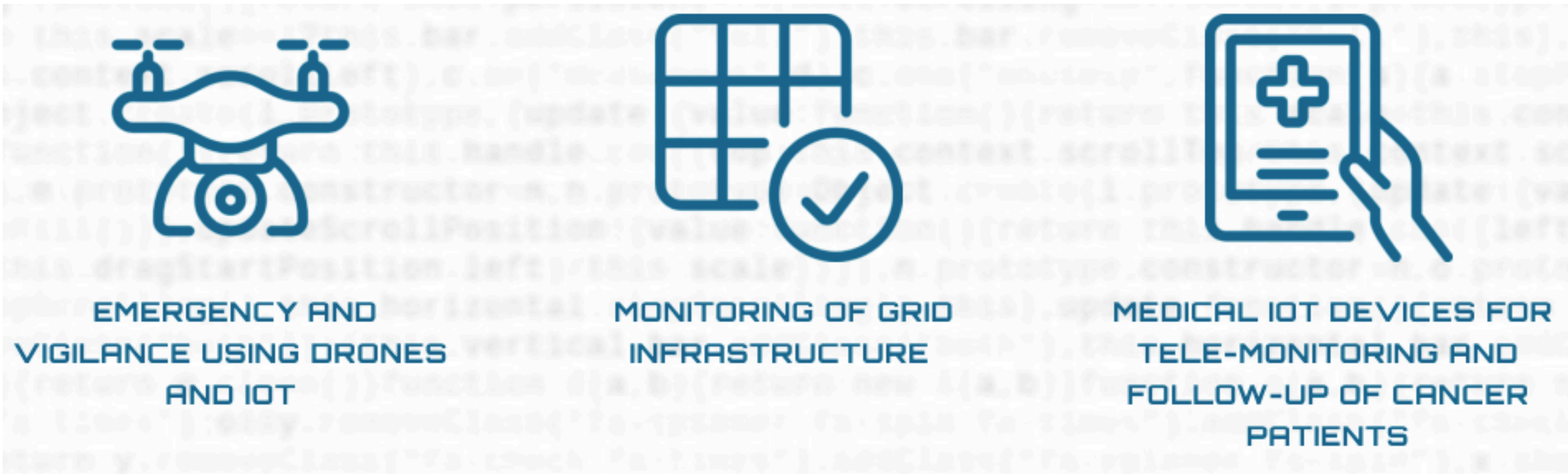
Reputation Systems & Security

- Reputation is built based on the honesty of the information source and on the accuracy of the information provided
- Scalable and efficient security analysis mechanisms in 5G networks to prevent attacks like botnets

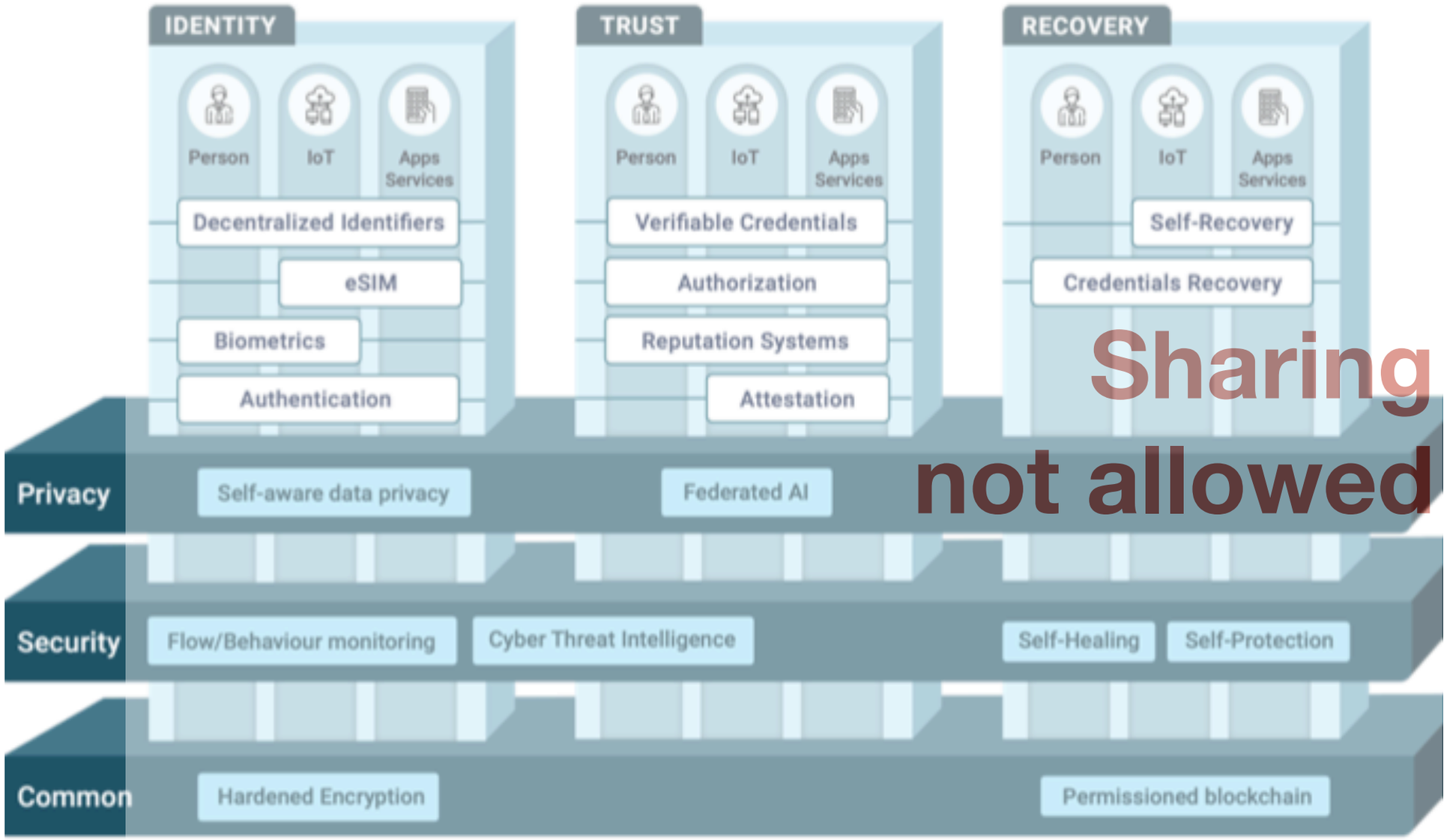
G. Fortino, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, “**Trust and Reputation in the Internet of Things: State-of-the-Art and Research Challenges**,” IEEE Access, vol. 8, pp. 60117–60125, 2020.

ARCADIAN-IoT 1/3

- ARCADIAN-IoT aims to develop a framework for trust, security and privacy management in IoT systems, including all the entities that interact with such systems
- It includes three use cases:



ARCADIAN-IoT 2/3



- The architecture of ARCADIAN-IoT includes several vertical and horizontal layers

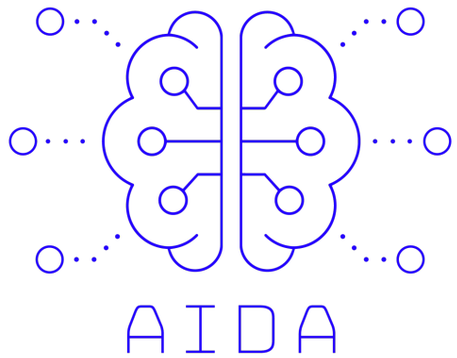
- Considers three types of entities: Persons, IoT objects, Applications and Services

ARCADIAN-IoT 3/3

- Attestation in ARCADIAN-IoT is compliant with IETF Remote Attestation Architecture
  - Using as Root of Trust (RoT) the embedded Subscriber Identity Module (eSIM)
  - Employing encryption chips (designed for military scenarios)

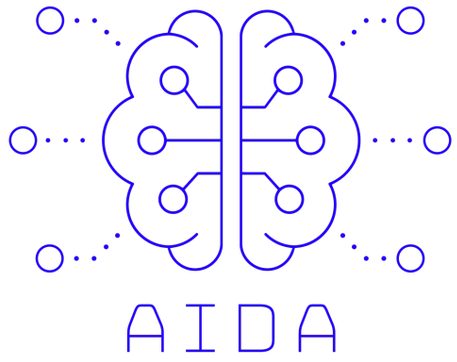
- Reputation is built:
  - On the interaction between the diverse entities (persons - IoT - apps) and between each type of entity
  - Stored using distributed systems such as blockchains

AIDA 1/3



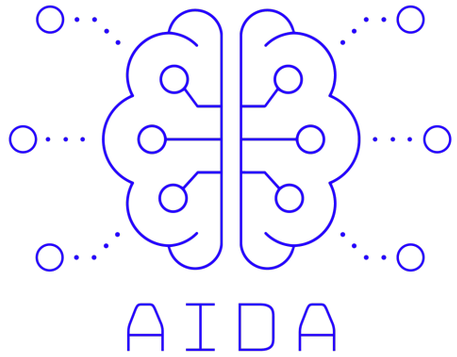
- Adaptive, Intelligent and Distributed Assurance Platform (AIDA)
- AIDA goals include:
  - Data Privacy and security
  - Support for edge computing and 5G
  - Device mechanisms for intrusion tolerance
  - Support Federated AI

AIDA 2/3



- The orchestration of resources and services relies on:
  - Service Mesh support (e.g. Istio)
  - Kubernetes clusters
  - KubeEdge for services in the edge
- Security of services is enhanced with:
  - Efficient and secure protocols (e.g. HTTP/3)
  - Federated authentication mechanisms (e.g. OpenID Connect)

AIDA 3/3



- AIDA Context Information (ACI) is being designed and evaluated to further enhance the security of services and devices:
  - Relies on OpenID Connect
  - Includes information to enable a trust relation between a person, a device and applications interacting with services
  - Specified in a way to enable trust relation between functions of service (e.g., employing ID tokens suited for components of a service)

Comments, Questions

Hands-on

- Introduce CloudSimSDN as a platform to simulate SDN and SFC
- Perform a simple simulation
- Files available at the GitHub repository:
  - https://github.com/bmsousa/tutorial-ISCC2021)

Initial (reproducible) steps

1. Assure you have Java installed

2. Download CloudSimSDN (https://github.com/Cloudslab/cloudsimsdn)

3. Integrate CloudSimSDN with the base code of CloudSim (https://github.com/Cloudslab/cloudsim)

   - Two approaches are suggested (the first one is advisable to access all the source code files)

4. Edit the pom.xml file

Initial (reproducible) steps

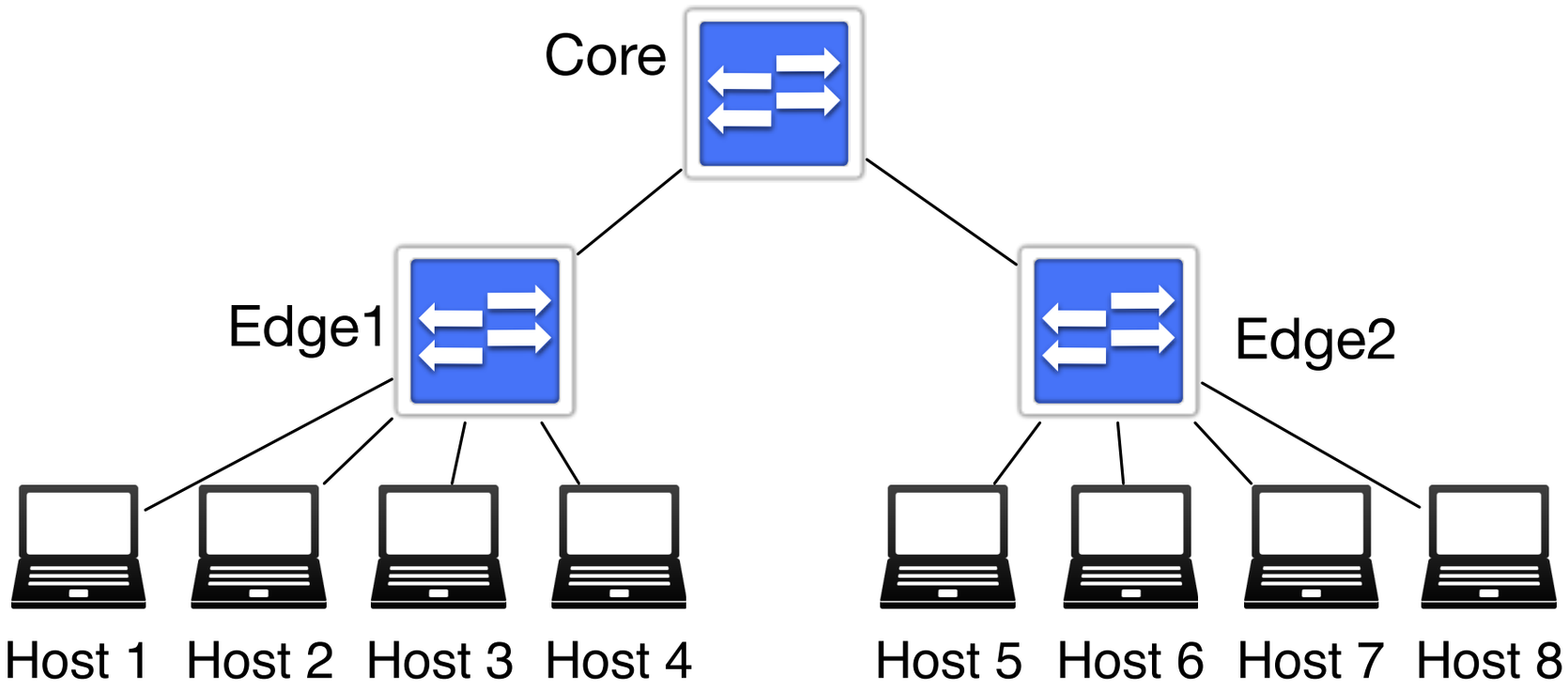
1. Download files from (GitHub of tutorial repo) and place them in a folder at the root folder (cloudsimsdn-master)

2. Place ***run.sh*** at the root folder

3. Compile project (maven tools)

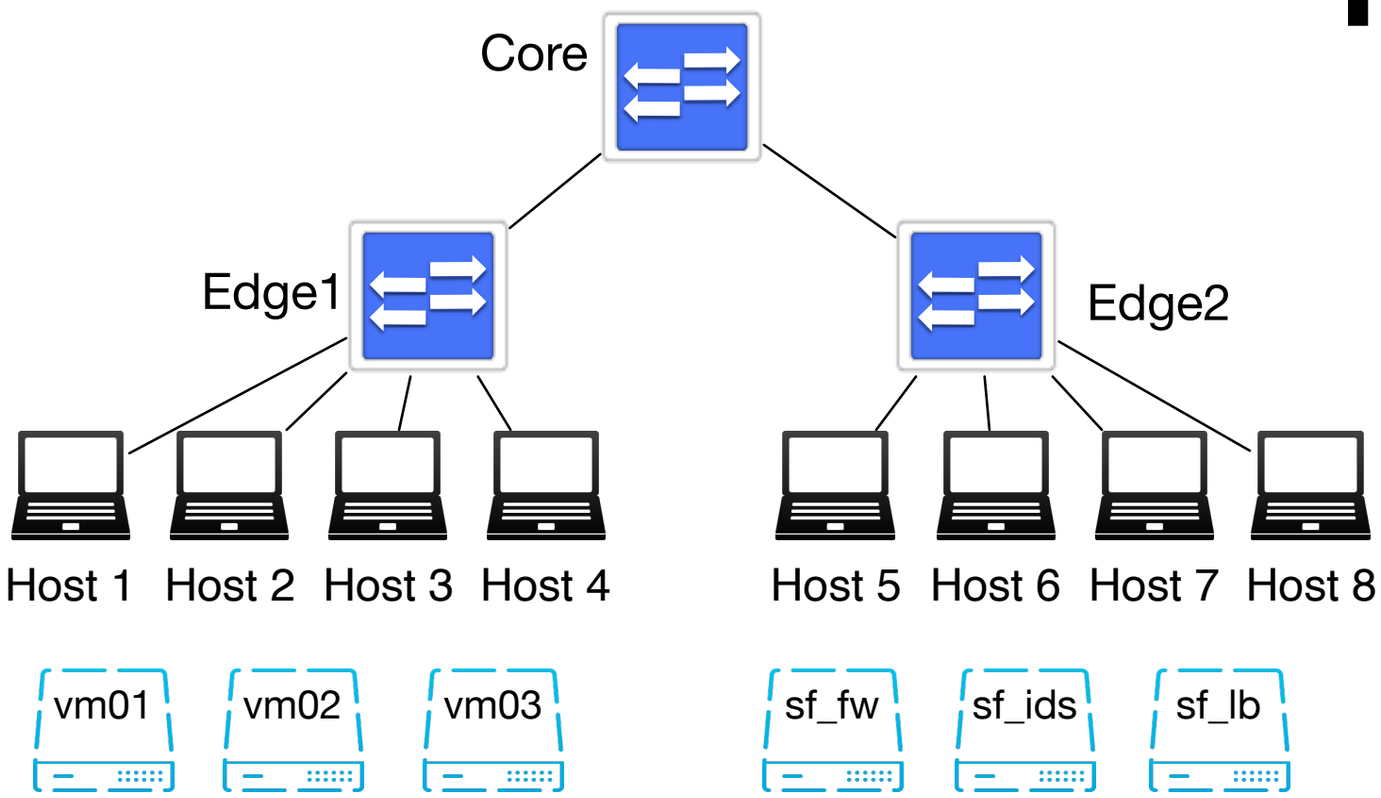
4. Execute the commands at the ***run.sh*** script

SFC scenario physical topology



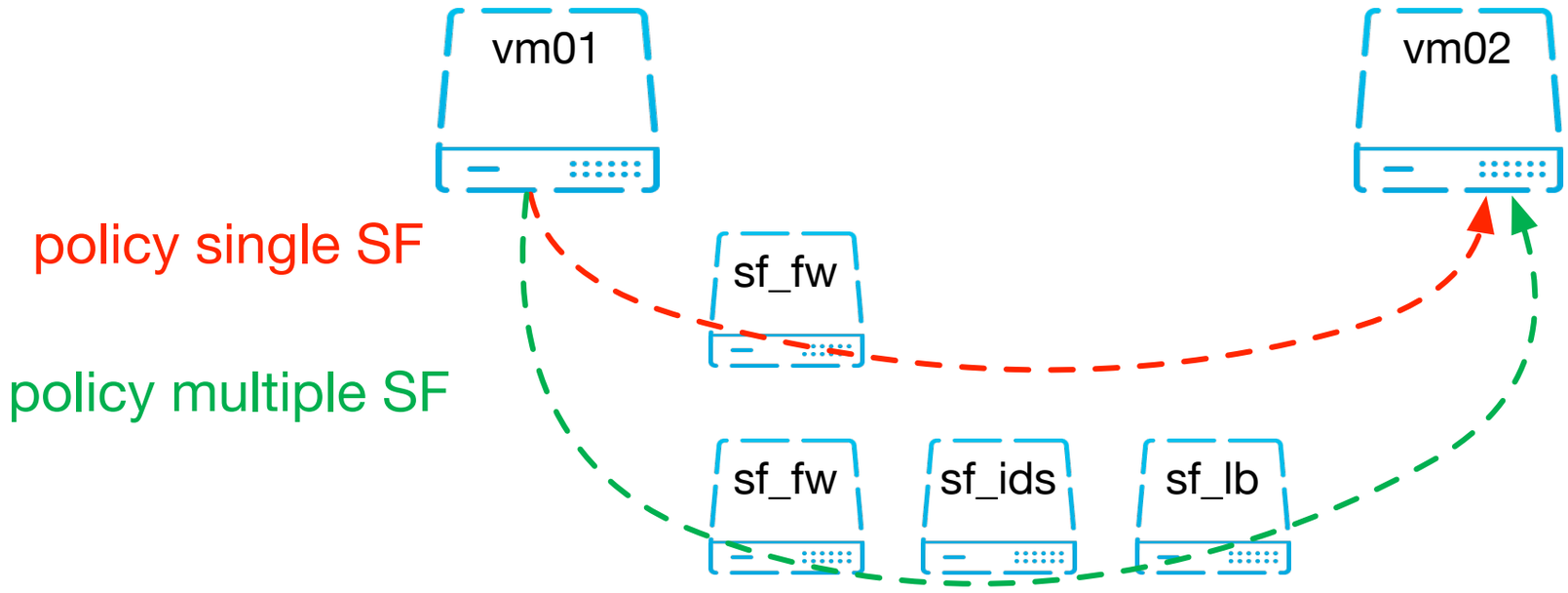
- File ***sfc-topology-physical.json***
- Single datacenter

SFC scenario virtual topology



- File ***sfc-virtual-topo.json***
- The following VNFs are considered: vm01, vm02, vm03
- The following SFs are considered: sf_fw, sf_ids, sf_lb

SFC policies



- File ***sfc-virtual-topo.json***
- With SLA values (expected time)

Expected results (simplified)

- Without SFC:
  - No SLA violations without SFC
  - No usage regarding the VNFs associated with the SFs
- With SFC:
  - security is enforced, traffic traverses the firewall, IDS and load balancer
  - VNFs associated with SFs are used

Comments, Questions

Coming to the end…

- Let’s summarise the tutorial…

Wrap up 1/3

- Services (critical or not) are:
  - composed by different functions
  - can be deployed in different fashions (physical nodes, VMs, containers)
  - have associated different policies
  - need updates, or means to provide it in a efficient fashion
  - need to complete with current legislation regarding privacy and security (e.g. GDPR)
  - …

Wrap up 2/3

- Trust in services (not only critical):
  - Is ephemeral and subjective…
  - Measuring trustworthiness requires well defined properties and monitoring
  - requires the assessment of services (attestation is a good fit here)
  - require historic and reliable information of past interactions, behaviours (distributed reputation systems are a good fit here)

Wrap up 2/3

- Service Function Chaining:
  - enables the application/enforcement of policies
  - is decoupled from technology, network topologies
  - provides the ‘means’ to prepare next generation services
  - But still require further enhancements in virtual platforms like k8s.

End

- Thank you for being with us !
- Comments, questions, suggestions (or beer) feel free to contact:
  - Bruno Sousa (bmsousa@dei.uc.pt)
  - Nuno Antunes (nmsa@dei.uc.pt)

UNIVERSIDADE D COIMBRA

FACULDADE DE CIÊNCIAS E TECNOLOGIA