SliceNet architecture
Cognition Sub-plane, and application use-cases

PRESENTER: KENNETH NAGIN, IBM HAIFA RESEARCH LAB

VENUE: TB EWORKSHOP

DATE: 27 MAY 2020

slicenet.eu

Introduction

Terminology

- Cognition (Artificial Intelligence, Machine Learning, Big Data)
- Quality of Service (QoS)
- Quality of Experience (QoE)
- Vertical (network slice user)
- Network Service Provider (NSP)
- Digital Service Provider (DSP)
- Plug & Play (P&P) Plugin

Goals

- Cognitive Driven Problem Determination (Predict problem before QoE degrades)
- Cognitive Driven Remedial Actuation (Automate network optimization)
- Vertical in the loop

Webinar Agenda

- Agenda
  - Purpose/Objectives (Why is Cognition required for Slice QoE Management?)
  - Requirements and challenges (Why is it hard?)
  - Technical approaches for design and prototyping (What are the basic building blocks?)
  - Technical achievements and Use Cases (What did we actually do?)
  - Summary of innovations (rap-up and time for more questions)

Why use cognition for slice QoE management?

- Many workloads, dynamic traffic patterns
  - Must constantly **adapt, anticipate**
- Multiple data sources, multiple owners, multiple semantics, multi-layering, multi-domain
  - Must **combine sources, interpret, predict outcomes**
- E2E Quality of Experience (QoE) per slice
  - Must derive QoE from Quality of Service (QoS)
- Explosion of possible per slice states and possible configuration
  - Must **scale**

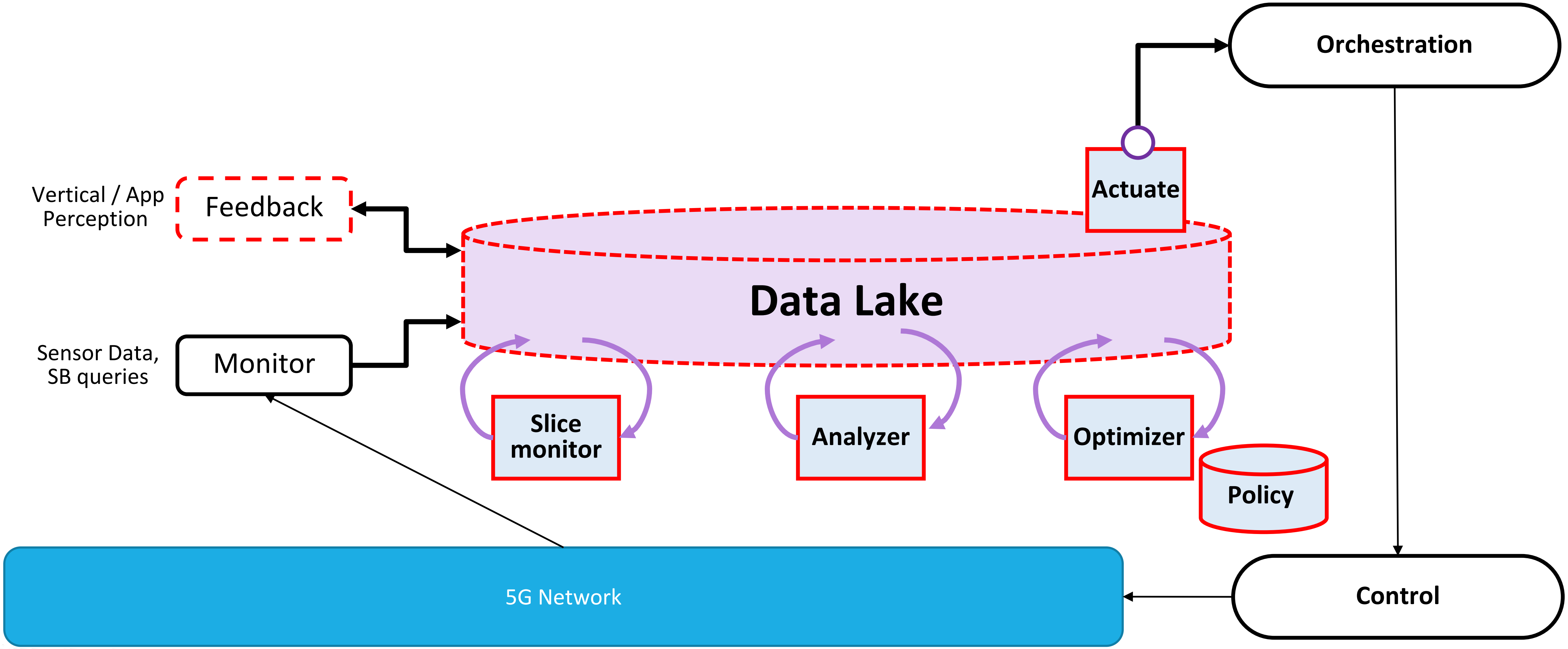
Traditional problem determination, e.g. thresholding, not adequate.

Cognition Required

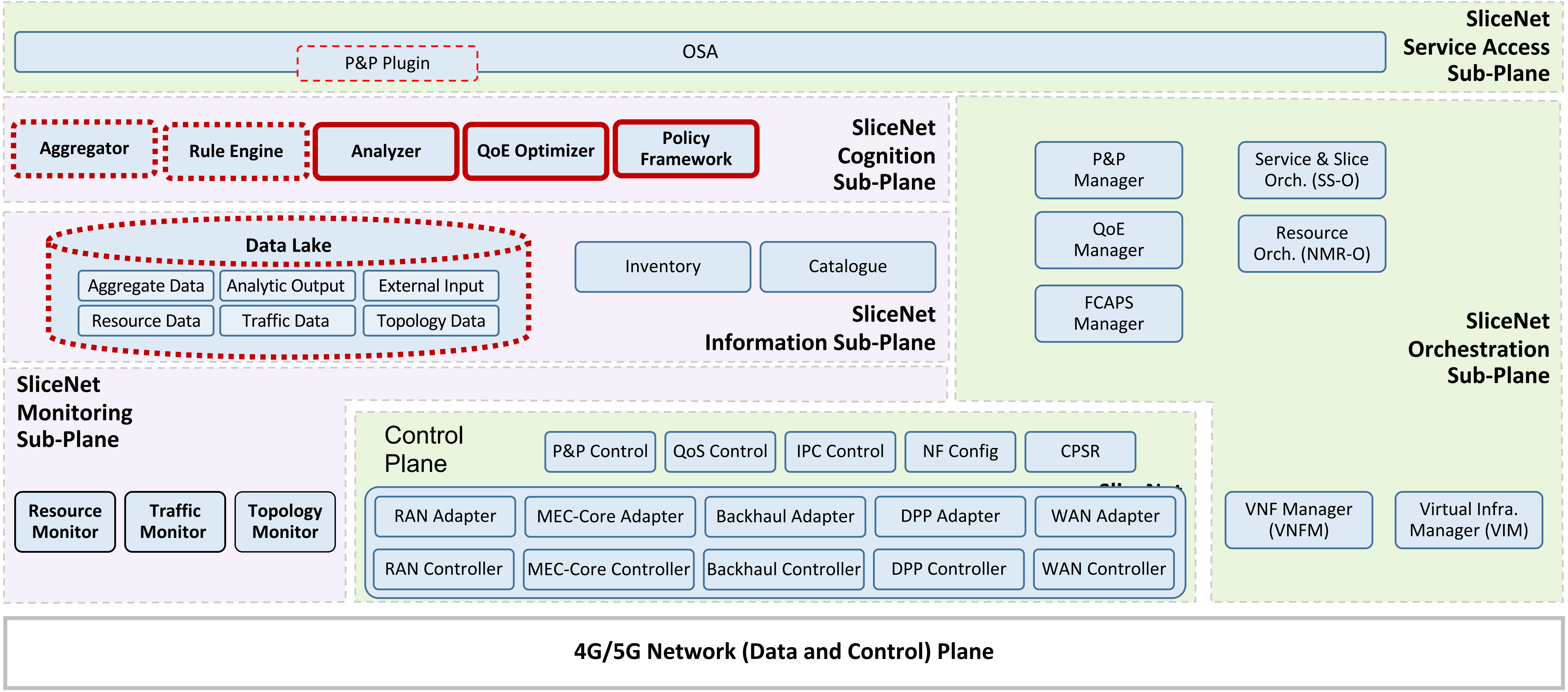
Challenges

- Combine Cognition with “traditional” network operations management
  - Event-action, policies
- Many machine learning methods
  - Allow easy integration of new analytics
- Big Data management
  - Many sources and Many components using data
- Harmonize under single architecture
  - Allow mix-and-match of different tools, orchestrate cognition across layers and domains
  - One paradigm for both NSP and DSP
- Quality of Service (QoS) vs Quality of Experience (QoE)
  - Network level QoS KPIs do not reflect E2E QoE
  - Must **estimate** and **predict** actual QoE

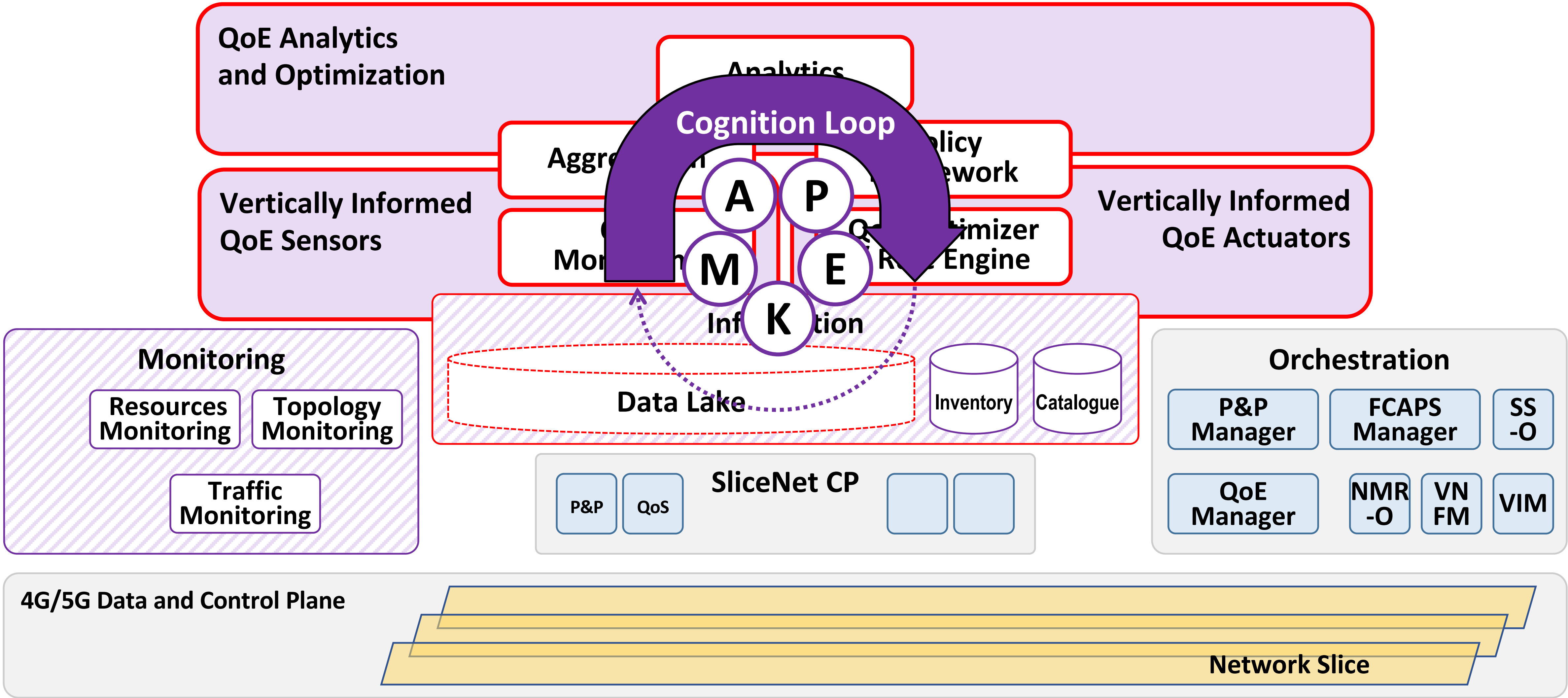
Cognitive driven
problem determination, prediction and remedial actuation



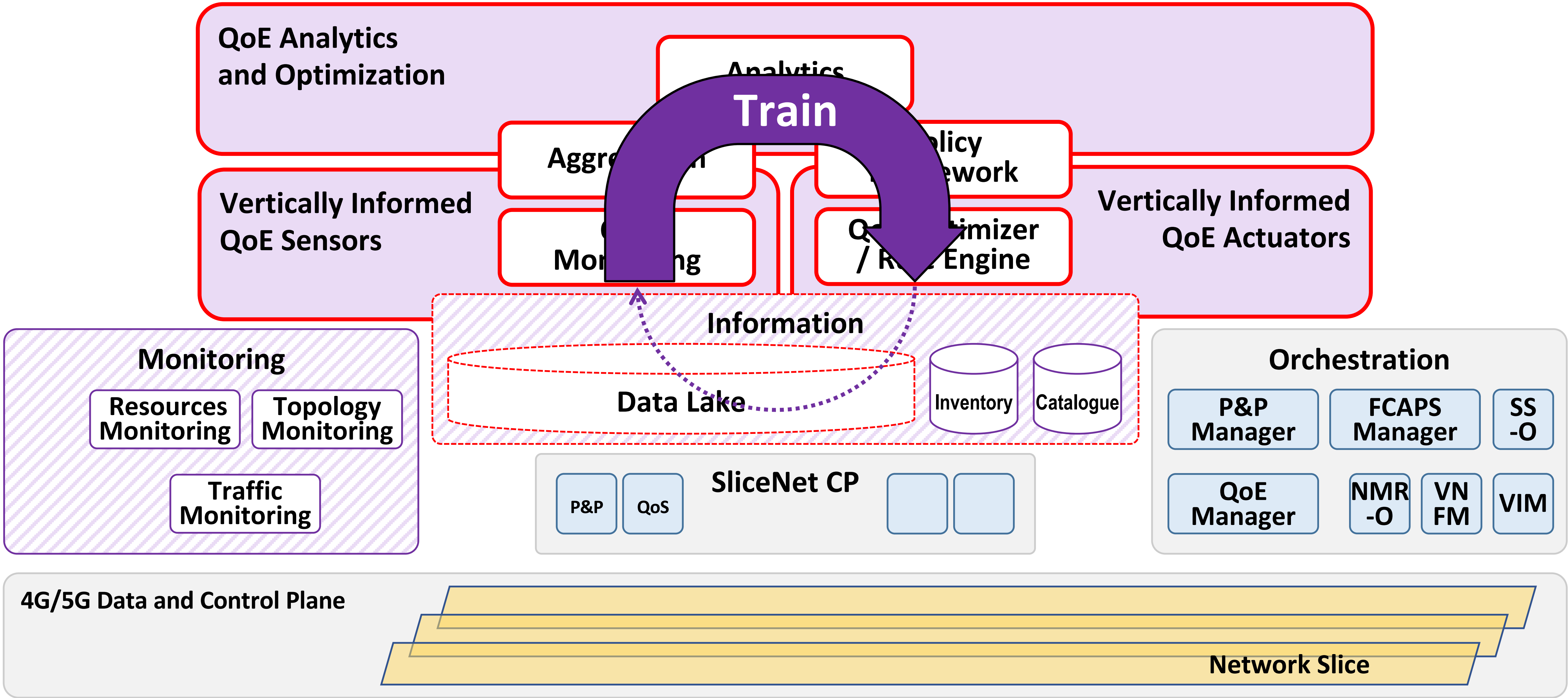
SliceNet Architecture



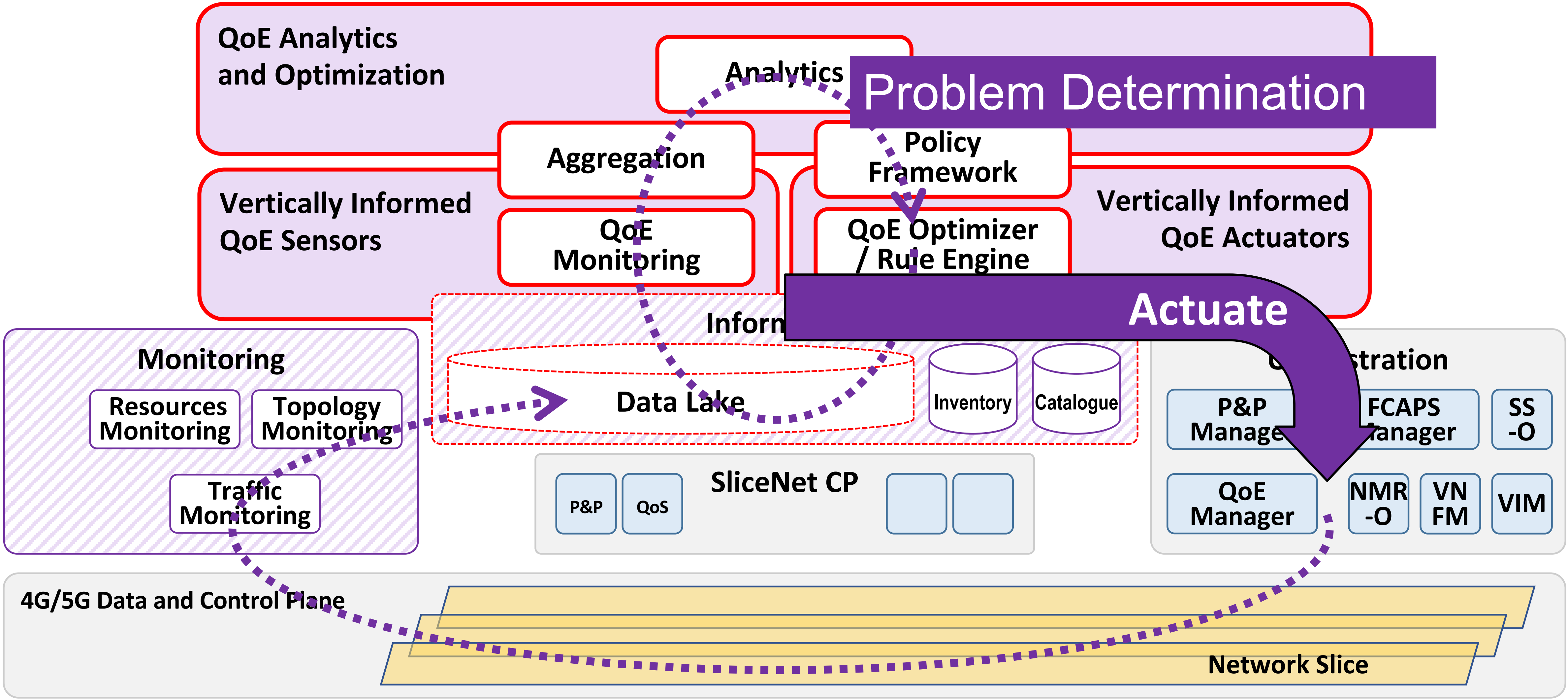
MAPE-K cognitive management loop



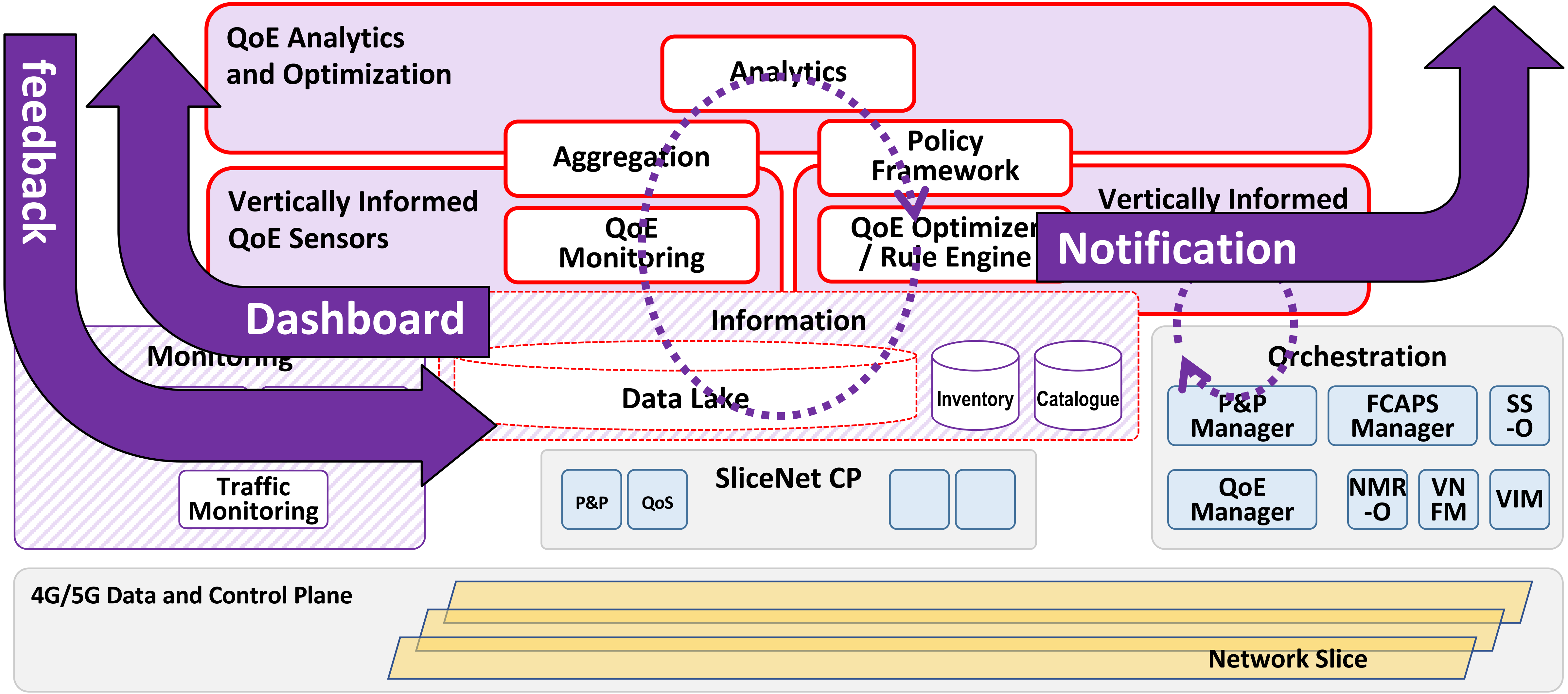
Learn/train: generate knowledge (as policy)



Cognitive Driven Remedial Actuation



Vertical In the Loop (Plug & Play Plugin)



Three Use Cases

Use Case	ML Model	Model Type	Remedial Actuation	Quality of Experience (QoE)
Smart Grid	Predict RAN degradation and RAN failures from alarm data	Neural Network	• Modify slice network parameters (bandwidth), • Failover to new RAN	Power grid under constant observation and control.
Smart City	**Detect performance degradation due to Noisy Neighbours**	**Random Forest**	• **Bandwidth** • **VNF scaling (VM Scaling),** • **VNF migration (VM Migration)**	**All signals from light sensors received as usual.** **No lose of control of lights.**
eHealth	**Anomaly Detection:** • **Data from ambulance mobile plug-in** • **Observe network behavior for the last 5 minutes in order to forecast the signal strength degradation within the future 5 minutes.**	**Random Forest**	• **Traffic Re-direction within same NSP** • **Hand-Over to another NSP**	**No degradation in video stream noticed by health workers.**

Prototyping

- ❖ Delivered SW components prototypes and interfaces available at SliceNet Git:
  - ✓ QoE REST Client: https://gitlab.com/slicenet/qoe-rest-client
  - ✓ QoE Plugin: https://gitlab.com/slicenet/qoe-plugin
  - ✓ QoE Optimizer: https://gitlab.com/slicenet/qoe-optimizer
  - ✓ Policy Manager: https://github.com/onap/policy-engine , Docker: nexus3.onap.org:10001/onap/policy-pe
  - ✓ Smart Grid RAN NS Prediction Model: https://gitlab.slicenet.oteresearch.gr/jose-nuno-sousa/cog-demo
  - ✓ Smart City Noisy Neighbour Model: https://gitlab.com/slicenet/noisy-neighbor
  - ✓ eHealth Anomaly Detection Model: https://gitlab.com/slicenet/anomaly_detection

Innovations

- Cognitive-driven state analysis and problem determination
  - Multiple ML Model Support
  - One paradigm for both NSP and DSP
- Cognitive-driven remedial actuation
  - Cognitive-driven triggers
  - Cognitive-driven policy framework
  - Actuators de-coupled from triggers (reusable)
- Cognitive-Driven & Traditional Network Management Integration
- Slice aware, vertical in the loop
  - Plug & Play Plugins
  - Vertically-informed Quality of Experience (QoE) sensors
- Data Lake
  - Data Sharing
    - Between monitors and Cognitive Sub-Plain
    - Between NSP and DSP
  - Component Decoupling

Further Information

Website: https://slicenet.eu/

Email: contact@slicenet.eu

Further information: https://slicenet.eu/publications/

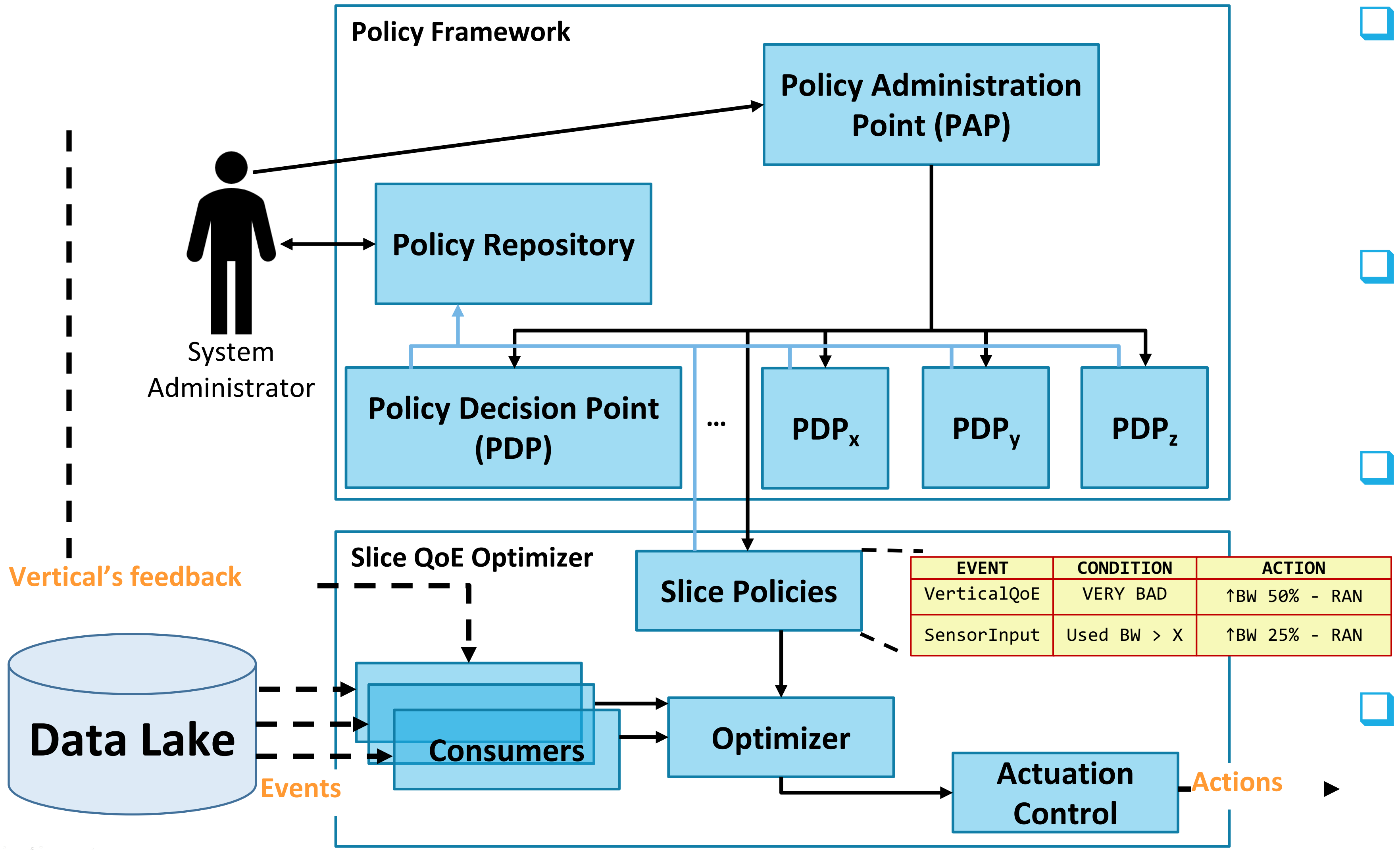
SliceNet Open source contributions:
https://slicenet.eu/software-contributions/

Questions ?

Thank you!

Backup Slides

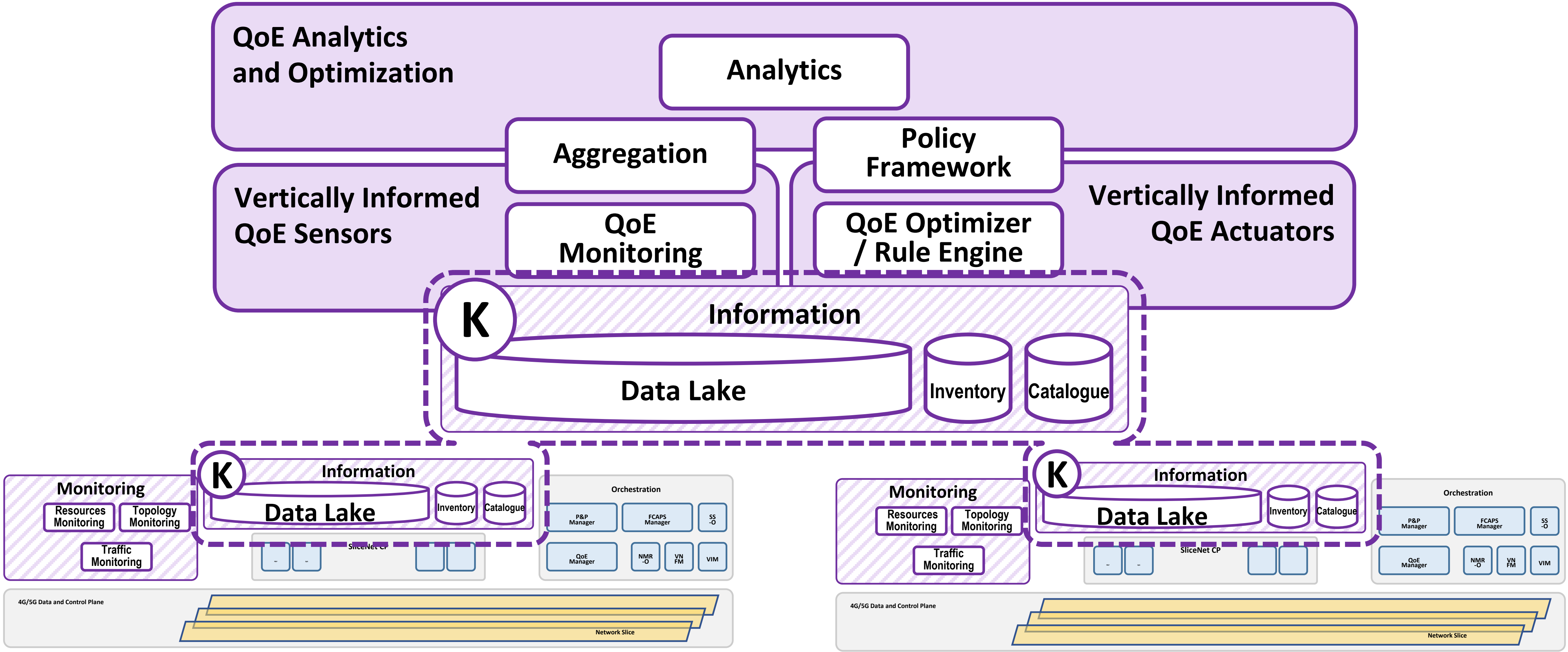
Policy-driven actuation approach



EVENT	CONDITION	ACTION
VerticalQoE	VERY BAD	↑BW 50% - RAN
SensorInput	Used BW > X	↑BW 25% - RAN

- A **policy-driven** actuation framework has been designed and implemented for the **Planning and Execution** phases.
- The Policy Framework defines rules in the form of **Event, Condition and Action** (ECA).
- The QoE optimizer acts as a PDP, enforcing actions to multiple points of the SliceNet architecture to **keep slice QoE levels**.
- Actuators can be **enriched with further analysis**

Knowledge sharing, hierarchical QoE mangmnt.



Workflows: QoE sensors, Vertical-informed actuators

- ❑ Reliable RAN slicing using NSP alarm data → (Smart Grid Use Case)
- ❑ Anomaly detection → (eHealth Use Case)
- ❑ Noisy neighbour detection → (Smart City Use Case)
- ❑ QoE classification from QoS metrics
- ❑ RAN optimization

Workflows: Vertical-informed actuators

- QoS modification (Increase Bandwidth) → (Smart City, Smart Grid)
- NSP sequence modification (Change RAN) → (Smart Grid)
- NSP sequence modification (Hand-Over, Traffic Re-direction) → (eHealth)
- VNF scaling (VM Scaling) → (Smart City)
- VNF migration (VM Migration) → (Smart City)
- OVS-based traffic classification

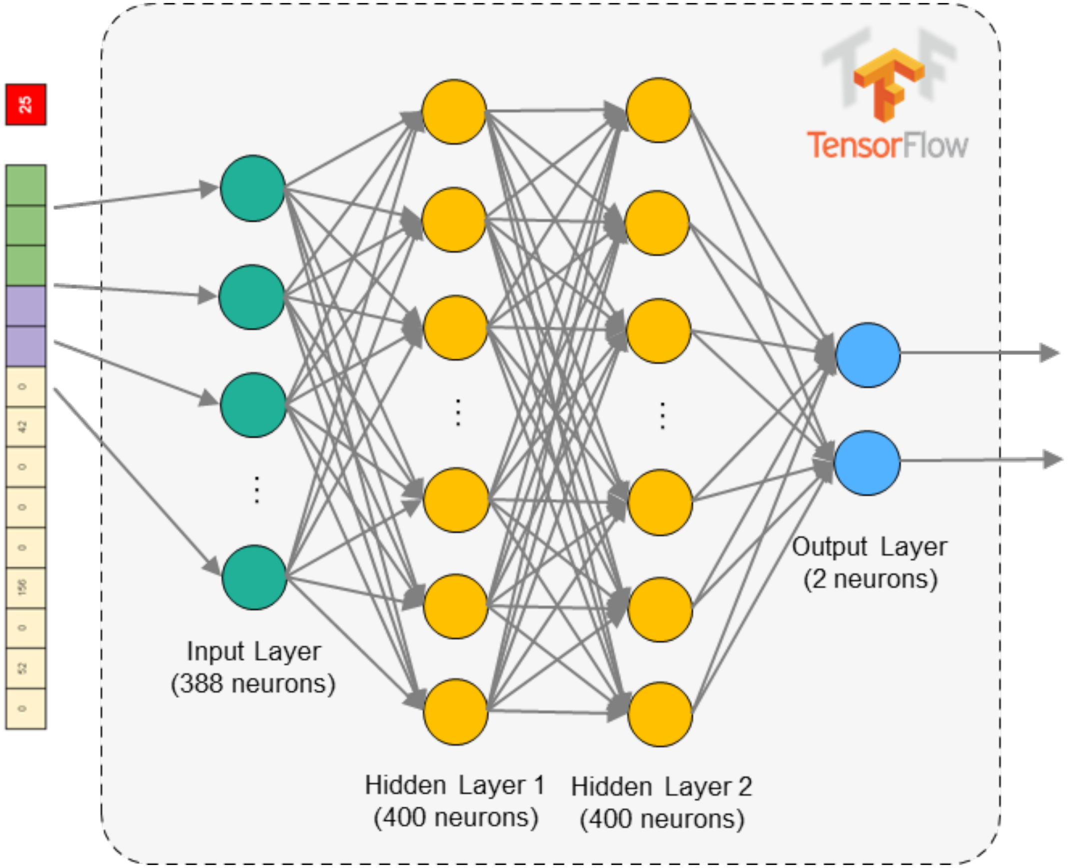
Reliable RAN Slicing using NSP Alarm Data (SG UC)

- Goal: Predict RAN degradation and RAN failures from alarm data
  - Estimate likelihood of imminent failure based on active alarms associated with the RAN resources (network equipment), as captured and generated by an **external system**
    - Data Source: Alarm Manager Platform from MEO (real)
  - Create a **reliability sensor**
- Solution: ML prediction model
  - Transform raw alarms to snapshots of active alarms per location
  - Label data using time to critical alarm
  - Deep learning (TensorFlow/DNN) model
- Usage: Trigger early actuation if failure predicted (before QoE is affected)
  - E.g., modify slice network parameters
  - E.g., failover to a different NSP

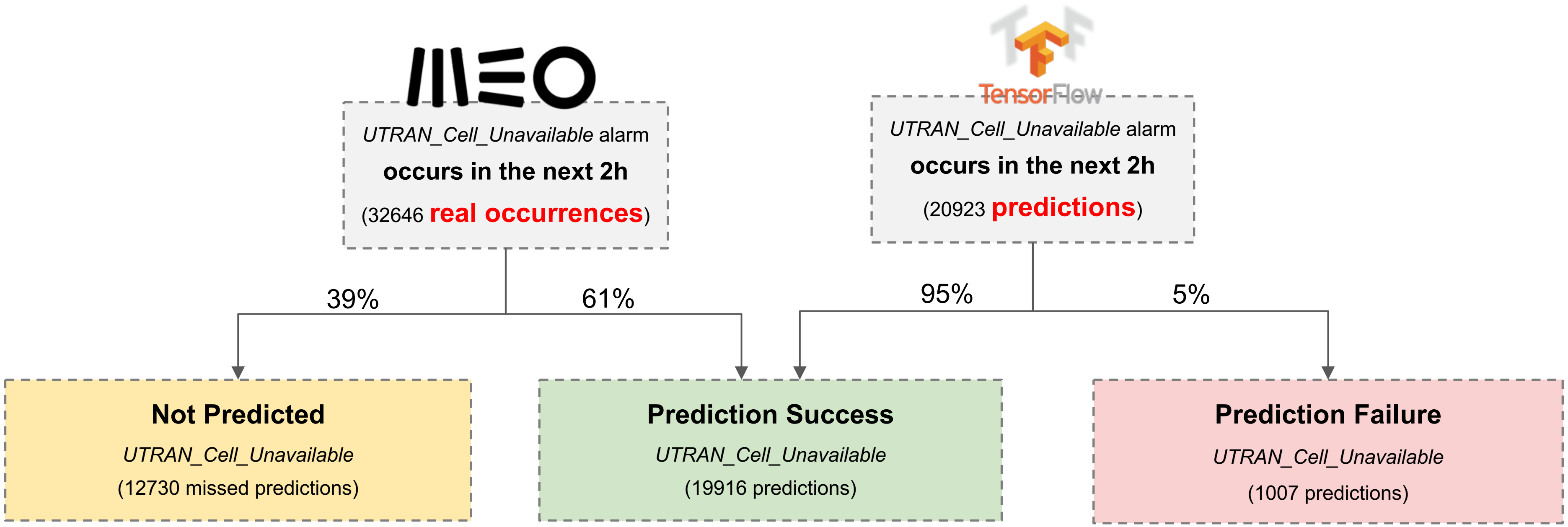
Reliable RAN Slicing using NSP Alarm Data

❑ Neutral Network Characterization

- Input
  - ❑ Label pre-processed into Time-to-Event and event occurrence flag
- DNN Configuration
- Deep Neural Network with 2 hidden layers of 400 nodes each
  - ❑ Custom loss function that gives the Time-to-event and the occurrence flag a proper interpretation in the model
- Output
  - ❑ Outputs into 2 nodes for the Time-to-event and probability of occurrence of the event



Reliable RAN Slicing using NSP Alarm Data



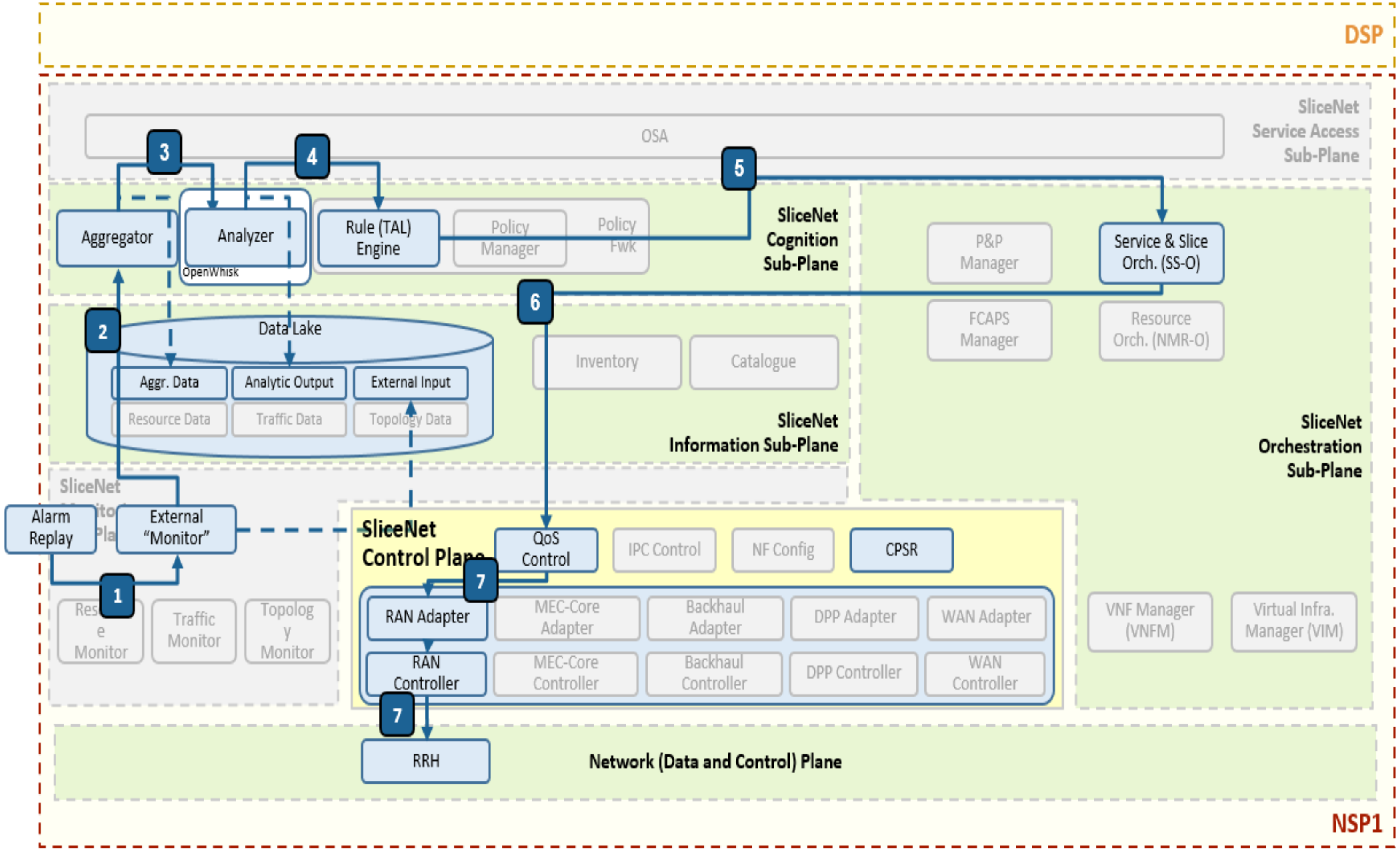
Summary:
- *61% of all real UTRAN_Cell_Unavailable alarms were predicted with success*
- *39% of all real UTRAN_Cell_Unavailable alarms were not predicted (not prejudicial to business as it is actual reality)*
- *prediction accuracy: 95% (of all predictions made, 95% were correct)*
- *predictions failed: 5% (the system predicted a UTRAN_Cell_Unavailable alarm but it never occurred)*

Industry Vertical Applications, Prototyping

Smart Grid

NSP

Network Slice optimization



Anomaly detection (eHealth UC)

- **Goal**: To observe the behavior of the network for the last 5 minutes in order to forecast the signal strength degradation within the future 5 minutes
- **Solution**: A machine learning approach using functional data analysis
  - A window-based approach
  - The model observes the curves of a set of KPIs for the last 5 minutes and labels the signal quality for the future 5 minutes

- **Usage**
  - To avoid the interruption of communication between the paramedic team in eHealth UC
  - To guarantee a top QoS in the slices offered to the vertical
  - To maintain the respect of SLA

Anomaly detection

Evaluation metrics

Correct classification rate:

$$CCR = \frac{\text{Number of correctly classified instances}}{\text{Total number of instances}}$$

Recall: $R = \frac{\text{True Positive}}{\text{True positive} + False\ Negative}$

To which point the malfunctions are detected

Precision: $P = \frac{\text{True Positive}}{\text{True Positive+False Positive}}$

To which point the detected malfunctions are pertinent

F-measure: $F = 2\,\frac{\text{precision} * \text{recall}}{\text{precision+recall}}$

weighted average of the precision and recall, (harmonic mean) where an F_1 score reaches its best value at 1 and worst at 0

Test on the training data with cross-validation

Learning with **Random Forest**

Validation with **cross-validation using 5 folds**

Evaluation metric	Value
Precision	**91%**
Recall	**93%**
F-measure	**92%**
CCR	**96%**

Test on the test data

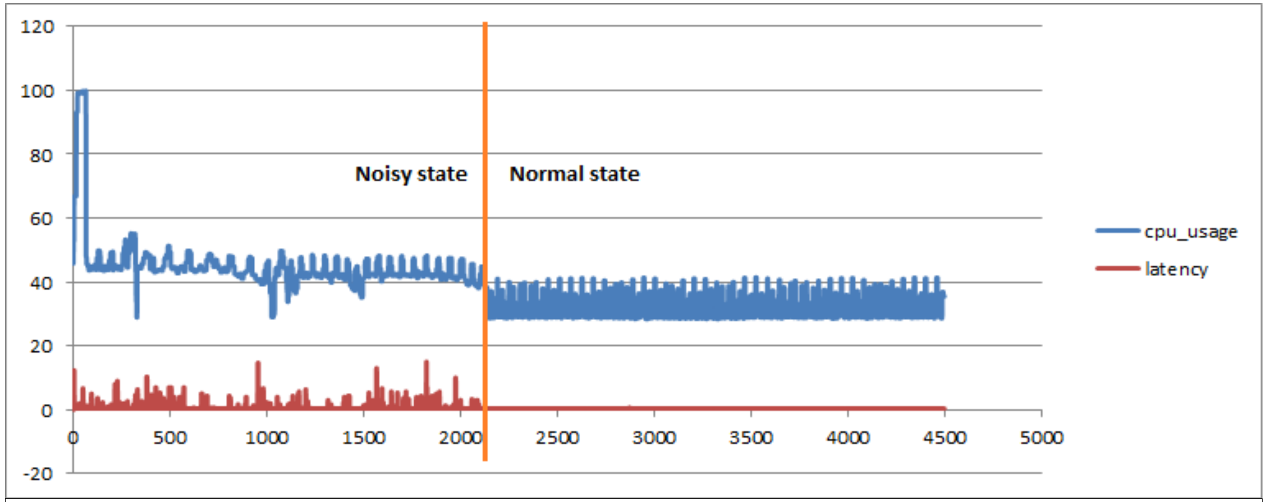
- 2h 21 minutes
- 960 instances

Evaluation metric	Value
Precision	**99%**
Recall	**99%**
F-measure	**99%**
CCR	**99%**

Noisy neighbour detection (Smart City UC)

- Goal: Find root cause of performance degradation as **noticed by Slice**
  - **Either** a "Noisy Neighbour" – provider is not meeting SLA
    Or not enough resources – need to scale out slice resources
- Solution: ML classification model
  - Use labeled data to train model
  - Simulate both cases to generate training data
- Usage: Trigger correct actuation (using KPI available to slice)
  - Notify provider (complain) OR adjust resource (scale out)
- Goal2: Find optimal VM placement to minimize "Noisy Neighbours"
  - Allow **provider** to fix the problem through VM migration
  - ILP model to minimized migration while eliminating noisy neighbour conflicts

Noisy neighbour detection



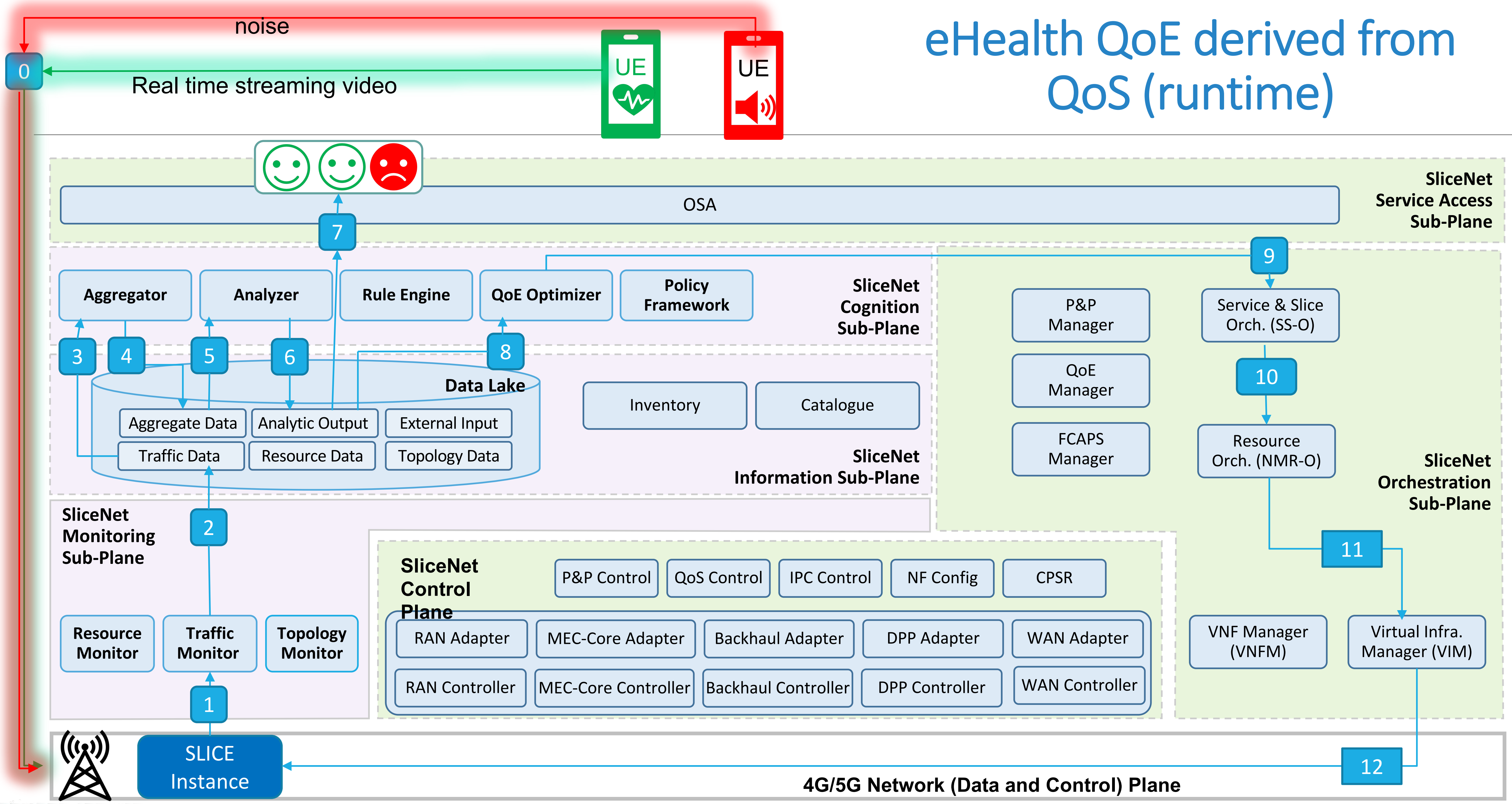
Impact of the noisy neighbour on the perceived QoE by the Vertical

Status	Percentage
Normal	30,7%
Noise	47,7%
Overload	21,6%

Algorithms Evaluation metrics	Decision tree	Random forest	KNN
Accuracy	77,9	99,9	99,8
Classification Error	22,1	0,02	0,15
Recall	77,9	99,9	99,8
Specificity	65,9	99,9	99,8
False positive rate	22,3	0,02	0,2
Precision	77,7	99,9	99,7
F_1 Score	77,79	99,9	99,74

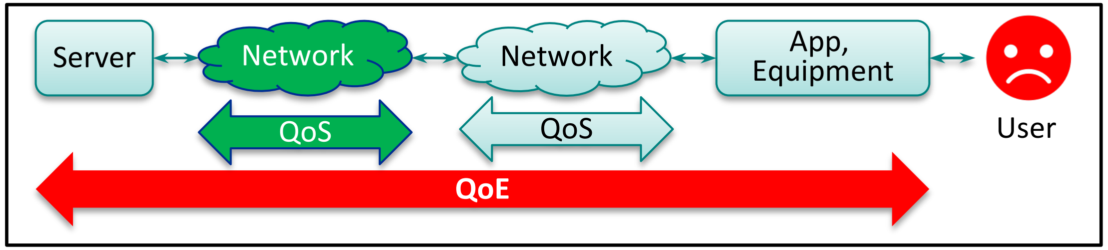
Random Forest and K nearest neighbor outperform the DT algorithm

eHealth QoE derived from QoS (runtime)



QoE classification from QoS metrics

- Goal: Estimate the QoE from measured network QoS metrics
  - Find relation between network metrics we can measure and the end-to-end QoE, as perceived / experienced by the vertical
  - Create an **end-to-end QoE sensor**
- Solution: ML classification model
  - Use labeled data to train model
  - Labels based on vertical feedback (actual QoE)
  - Align feedback with historical QoS KPIs



- Usage: Trigger actuation if estimated QoE is poor
  - E.g., notify vertical to allow graceful service degradation (e.g., reduce resolution)
  - E.g., adjust network parameters

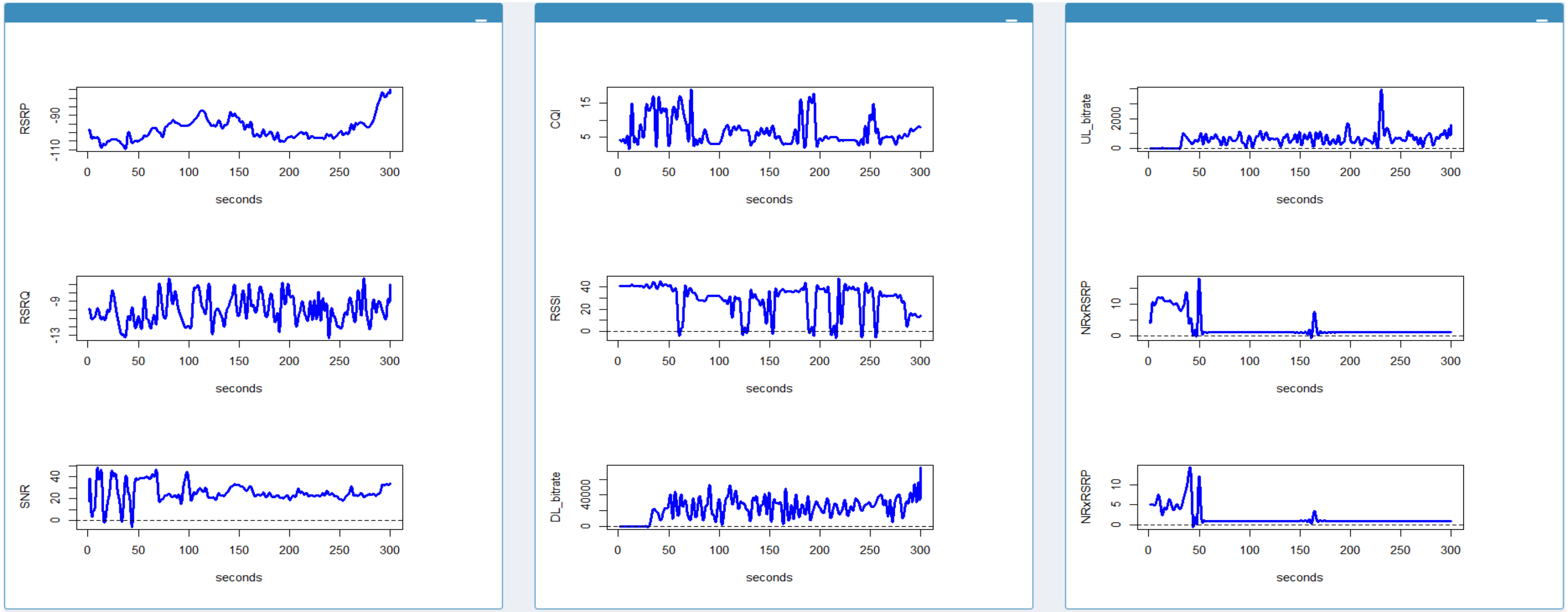
RAN Optimization

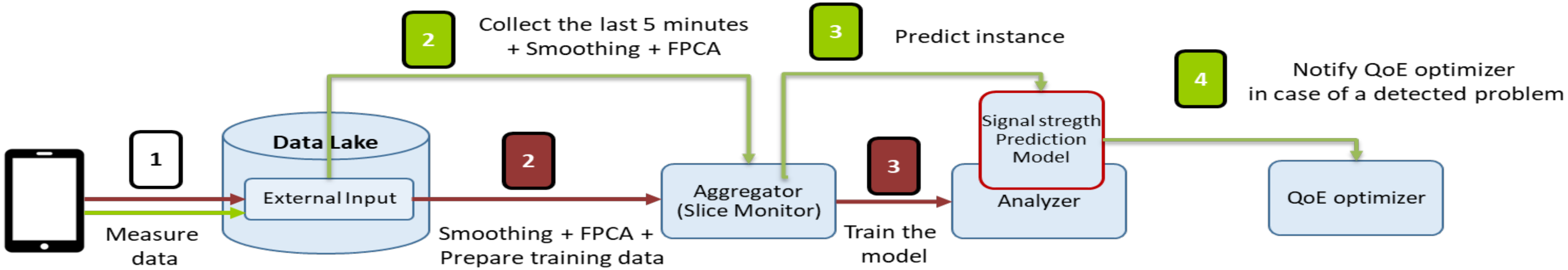
- Goal: actuate cognitive control and management over SliceNet's RAN
  - Semantic-based RAN abstraction
  - Cross-domain decision making for control Apps (e.g., video-optimization)
    - Objective: To maintain service continuity and SLA (downlink stream of at least 10 Mb/s)
- Solution: synergy of semantic reasoning and analytics
  - Monitor real-time link quality parameters (Spectrum Management Application data: transmission power, operating frequency, bandwidth; Radio Resource Management data: downlink throughput)
  - Create the domain specific semantic knowledge base
  - Fuzzy reasoning
- Usage:
  - System-wise optimizations which need complex control decisions
  - E.g., joint multi-operator spectrum management

Anomaly detection

The observed KPIs

- A Phone has been put into an ambulance
- A mobile application allows to compute the network metrics perceived by the UE
- A caption of 8 real traces having between 2 and 4 hours of length
- A measurement each **1s**





SLICENET

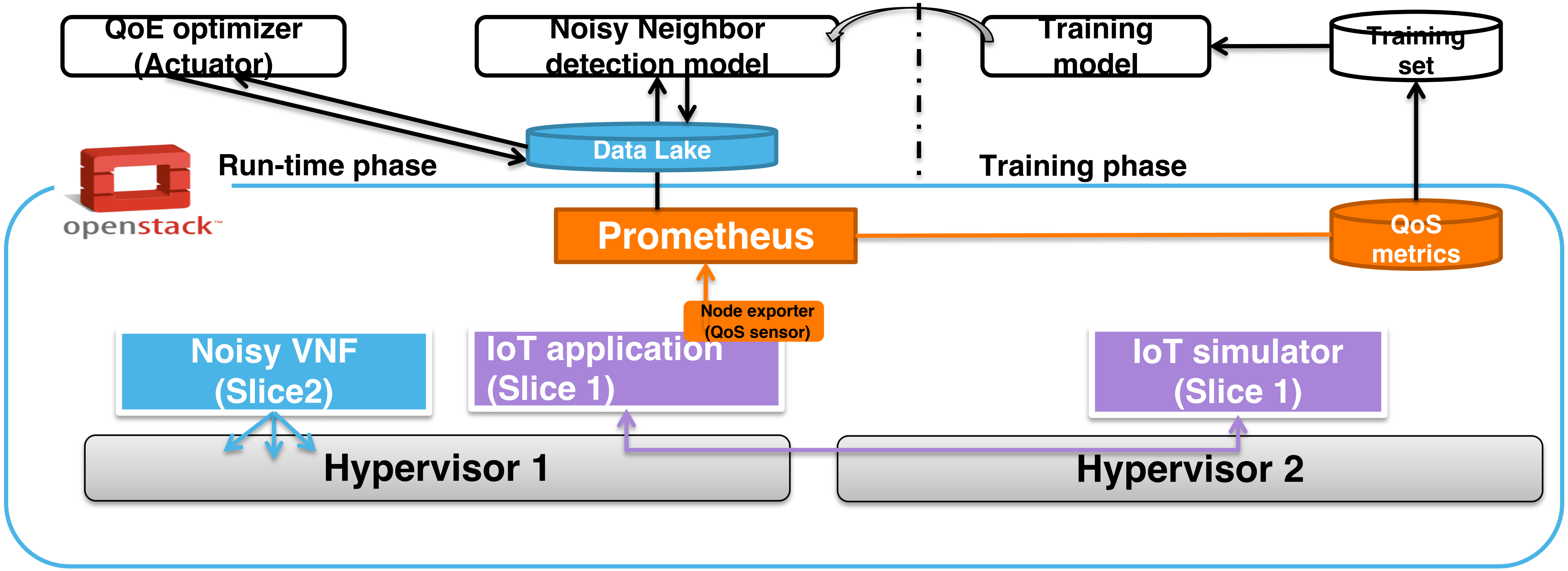
Noisy neighbour detection

Experimental Setup:

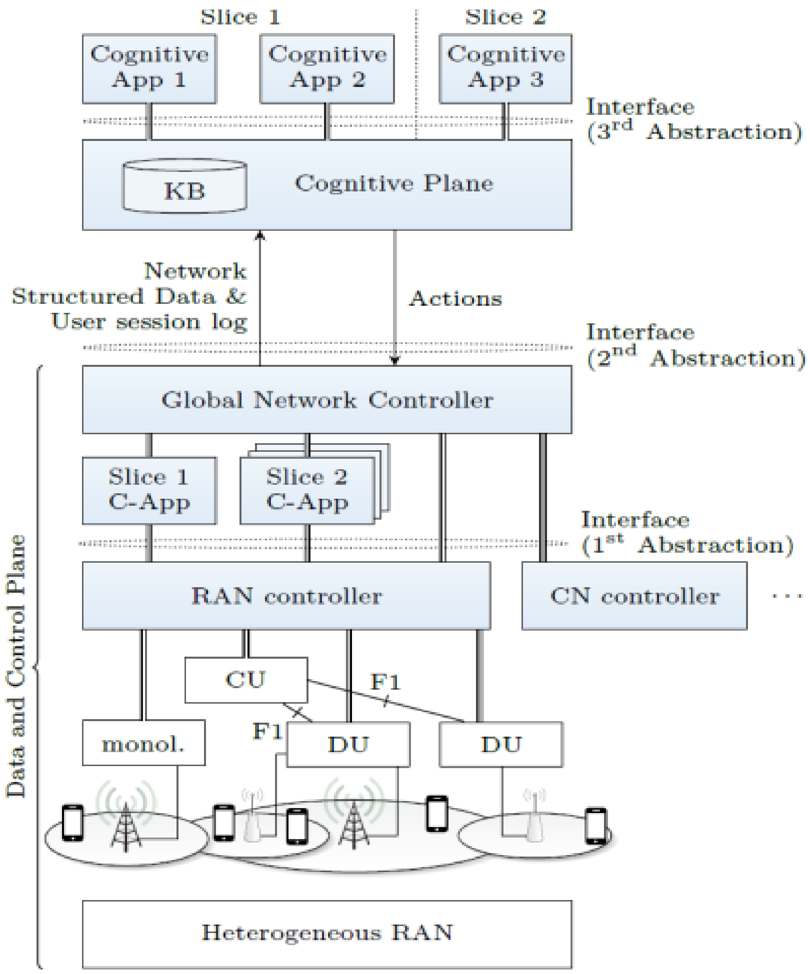
- IoT application: supervised VNF
- IoT simultaor: load generator to simulate a large number of IoT devices (lighting poles) connected to the IoT application
- Noisy VNF:acting as a noise generator to stress and to generate noise to other VNFs

Data Details:

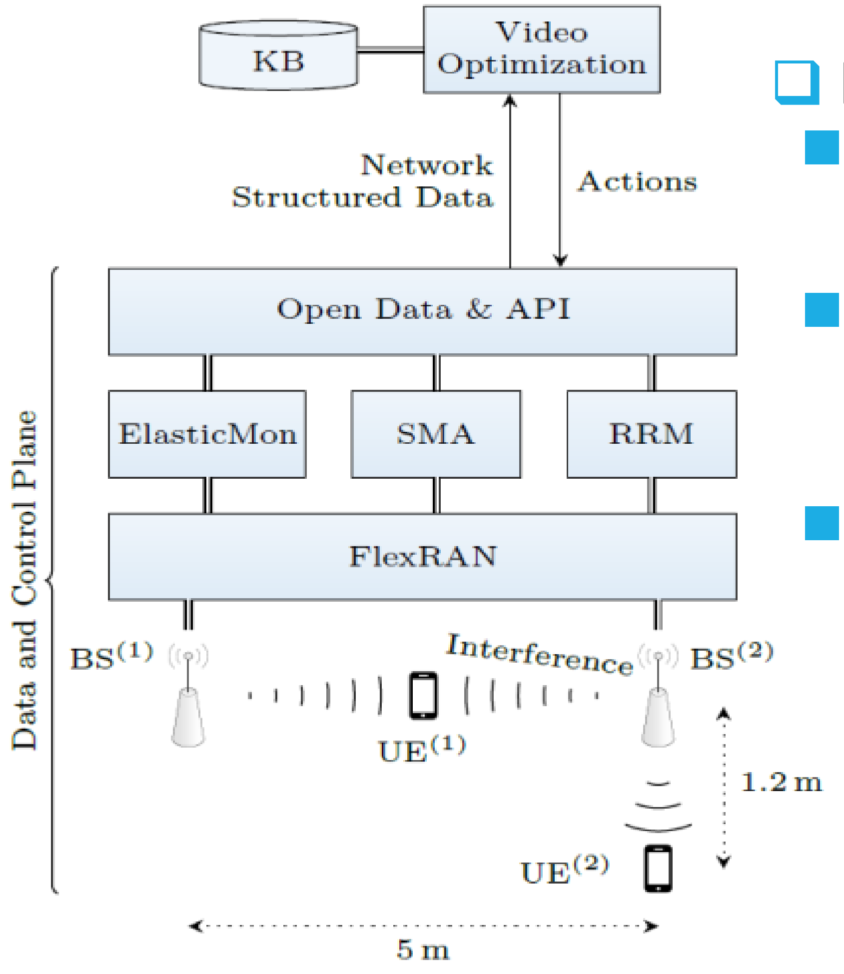
- Monitoring system: Prometheus server and node exporter as QoS sensor
- Three levels of stress, namely: noisy, overload, and normal
- Real data collected from ORO testbed: 48112 records and 4 features (cpu_usage, memory_usage, bandwidth_in and bandwidth_out).



RAN Optimization



Architecture of cognitive RAN control



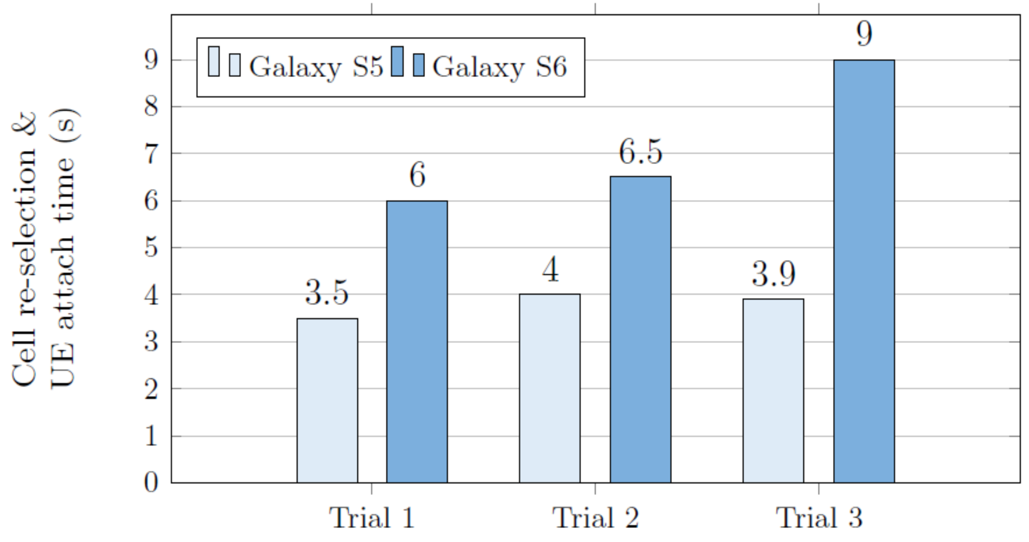
Video–optimization decision-making Element

- Experimental Setup:
  - Operation Band: 7, Frequency: 2.6GHz, Bandwidth: 5MHz
  - eNBs operate in the same band 7 on close to equal frequencies therefore create interference.
  - BS $^{(1)}$ streams video to its user UE $^{(1)}$

RAN Optimization

- Control decisions:
  - Adapt the video bit rate through video optimizer
  - Add/provision a new BS through SMA+ARCH
  - Increase the BW of the current BS (SMA)
  - Interference coordination through RRM
  - Update frequency and power through SMA
- Different objectives fed to the video optimizer influence the outcome:
  - Avoid interference to high priority users and/or their slices,
  - Small base station energy consumption,
  - Maximize joint system throughput,
  - High frequency reuse.

No.	P_{TX}(1)	BW_o (%)	Thr(1) (Mbps)	Score(1)	Thr(2) (Mbps)	Score(2)
1	L	0	12.1	0.72	16.8	1.00
2	L	50	5.5	0.33	16.8	1.00
3	L	100	5.5	0.33	16.7	0.99
4	M	0	16.8	1.00	16.8	1.00
5	M	50	16.6	0.99	16.8	1.00
6	M	100	14.2	0.85	15.8	0.94



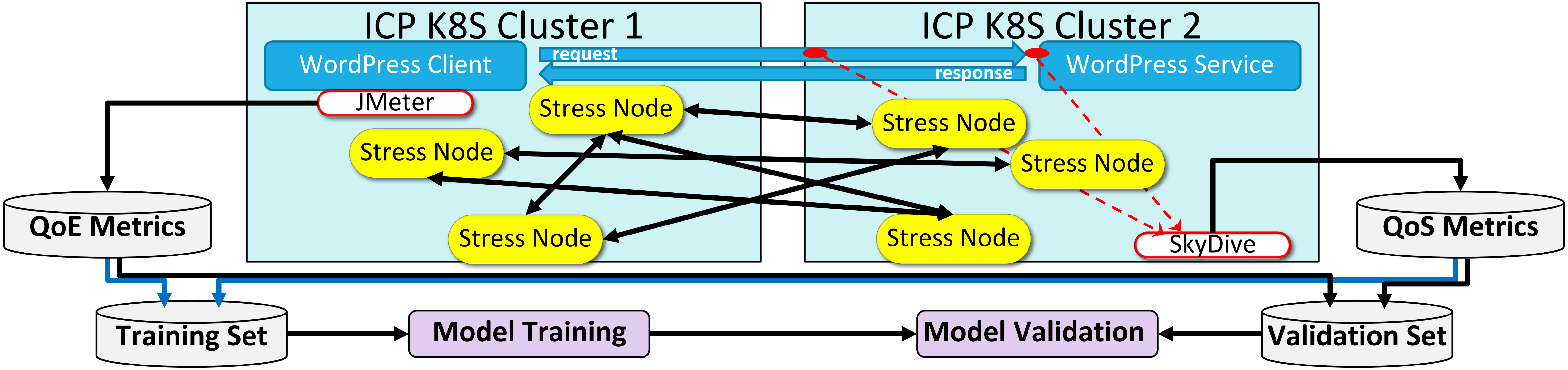
QoE classification from QoS metrics

❑ Experimental Setup:

- Two ICPs (K8S clusters)
- ICP1: WordPress client (Jmeter), Stress Generators
- ICP2: WordPress service, Stress Receivers, Skydive Monitor

❑ Data Details:

- Data curation: errors, outliers. TCP only, time, etc.
- Aligning QoS and QoE traces: 10 minutes / sample
- 108 Training / 28 Validation
- Use QoE service completion time (Jmeter)
- Calculate QoS Flow Duration (Skydive)
- Aggregations: mean, max, min, median, std, skew



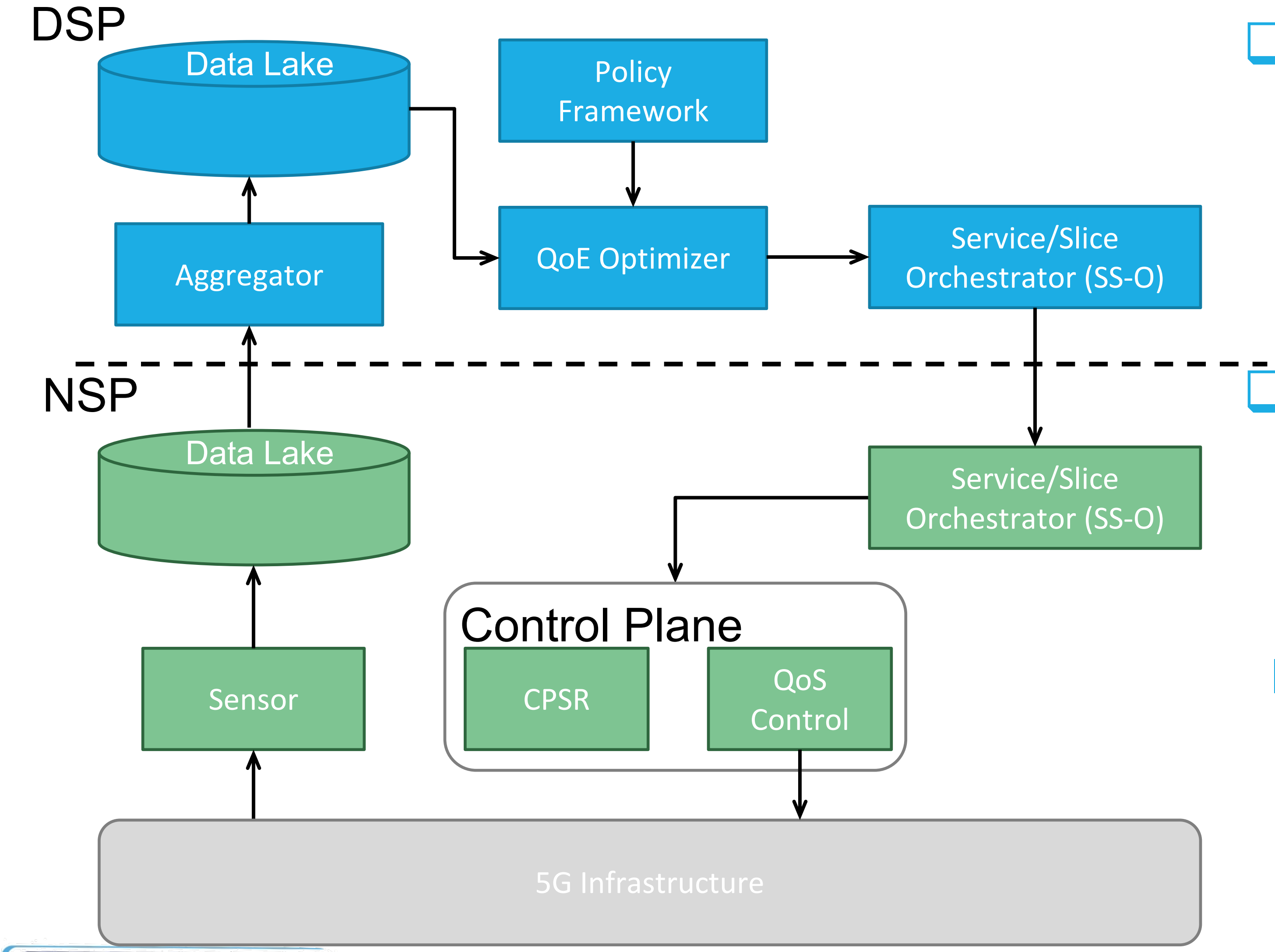
QoE classification from QoS metrics

Classifier	F Score	Correct Predictions	Under Estimations	Over Estimations
DecisionTreeClassifier	.93	25	1	1
RandomForestClassifier	.89	24	1	2
LinearDiscriminantAnalysis	.81	22	2	3
LogisticRegression	.63	17	0	10
GaussianNB	.63	17	0	10
SVC	.63	17	0	10
KNeighborsClassifier	.56	15	5	7
MLPClassifier	.11	3	17	7

Label	Mean Avg. Response Time Range Per Sample (ms)	QoE Description
1	1-1000	Good
2	1001-4000	Acceptable
3	4001-infinity	Bad

Actuation Slides

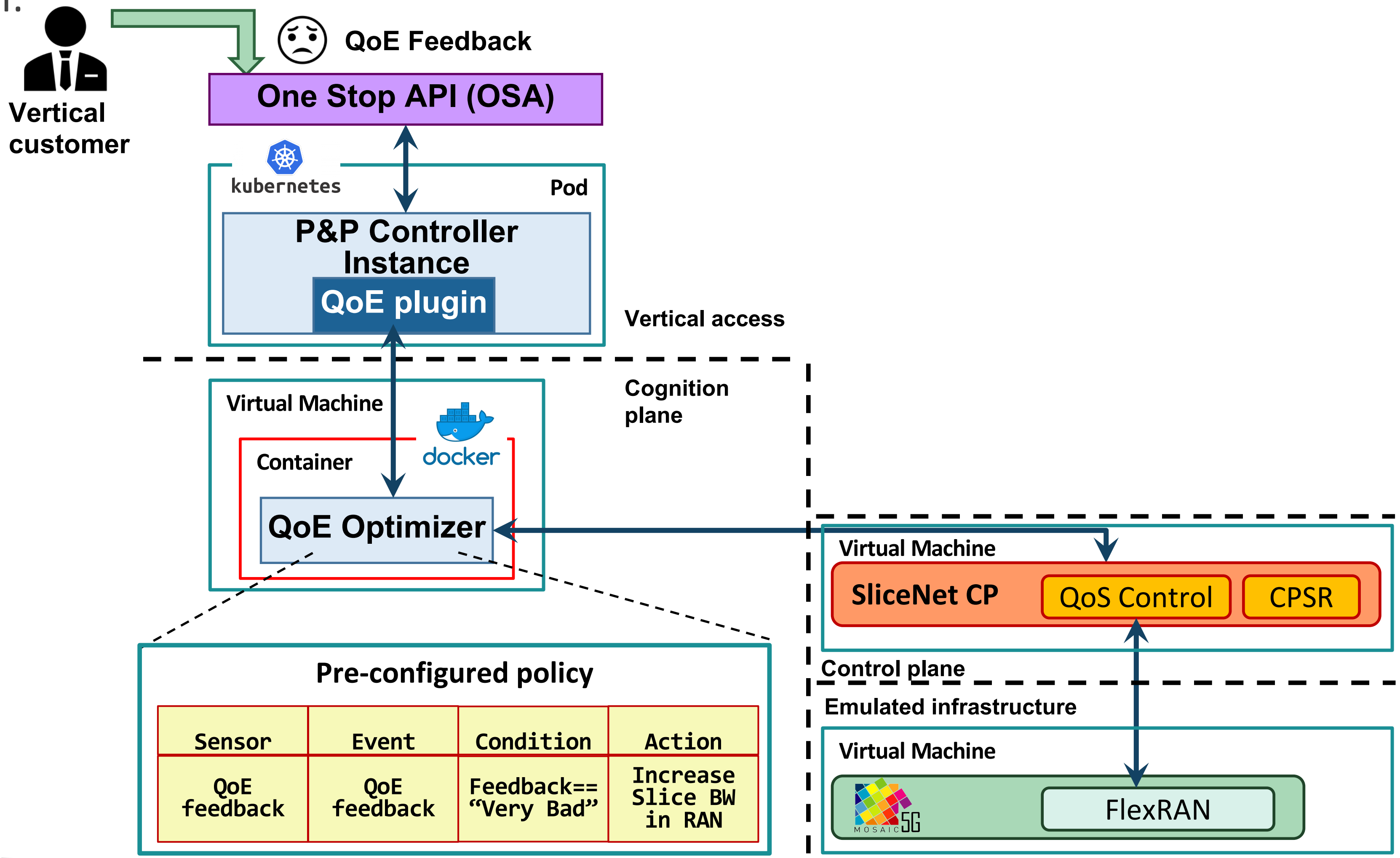
QoS modification



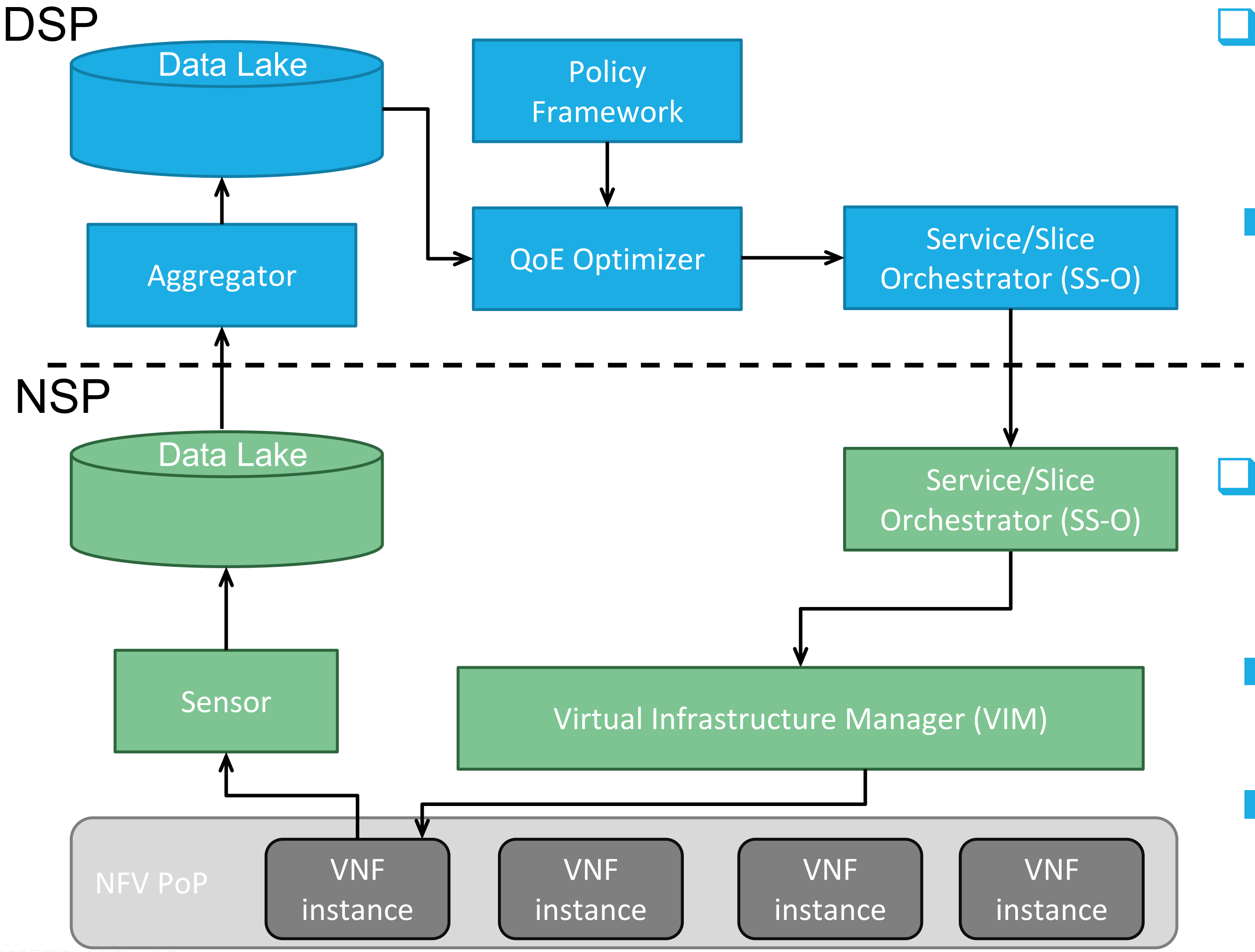
- Goal: **Modify slice/sub-slice QoS parameters** (bandwidth and priority) to maintain QoE levels
- Usage: to **remedy bad QoS situations** that impact on the perceived QoE by the vertical user/service
  - e.g. bad video quality in the eHealth UC may be attributed due to insufficient bandwidth of the slice

QoS modification

- Experimental set-up for PoC. A QoS modification is enforced to the RAN due to bad QoE feedback from the vertical:



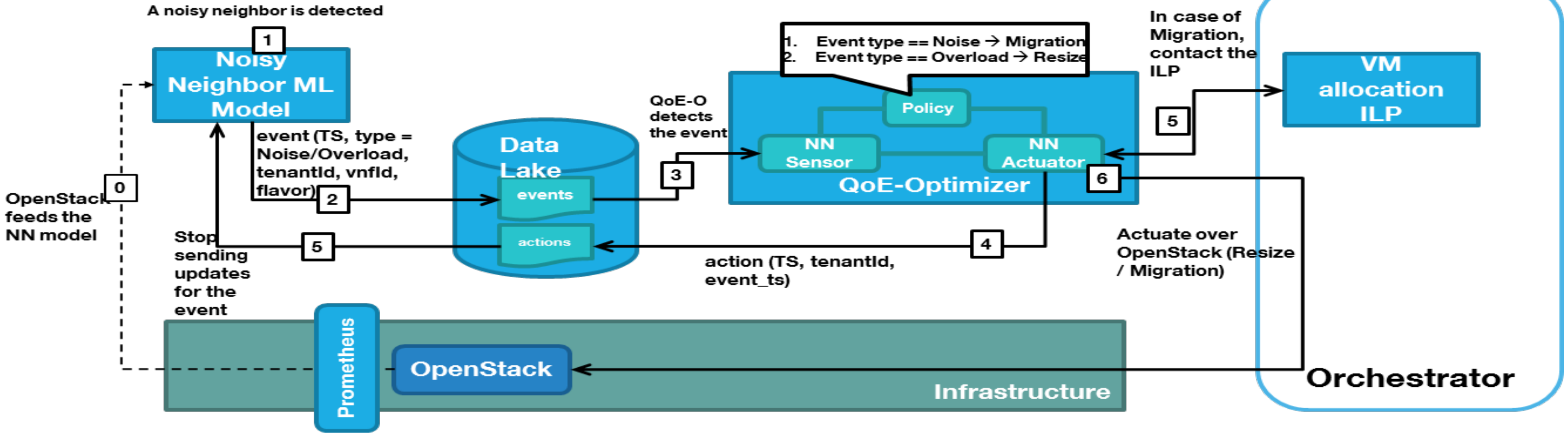
VNF scaling and migration



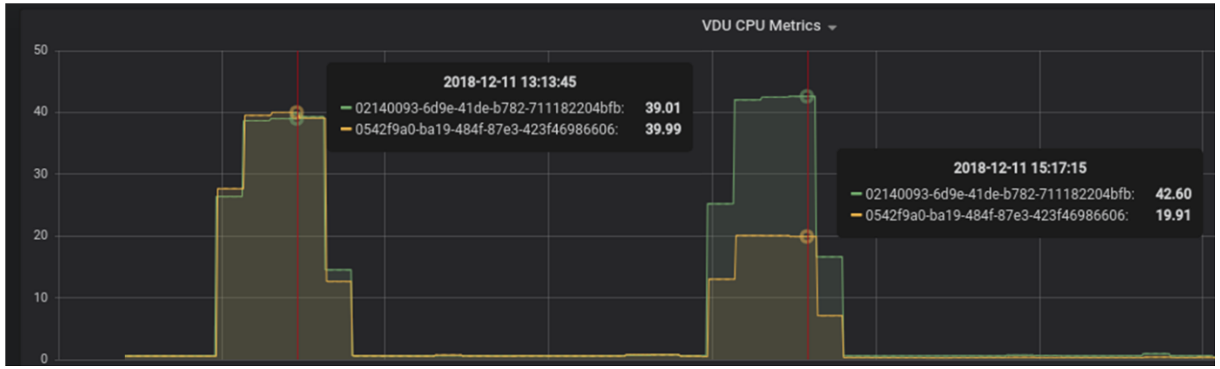
- Goals: **Modify the resources** assigned to a VNF instance; **migrate a VNF instance** from its original host to another host
  - A **policy** has been defined for the **Noisy Neighbor** cognition UC. The policy ties the state "overloaded" and "noisy" to the VNF scaling and migration actuators, respectively
- Usage: **enhance the performance** of infrastructure or application VNFs; **keep** the vertical service **KPIs** that are **affected by VNF placement**
  - e.g. increase the performance of the vEPC VNF to allow for high packet processing capabilities.
  - e.g. reduce CPU noise from VNFs collocated to the same host as the target VNF

VNF scaling/migration

- Experimental set-up for PoC. It is enclosed in the NN UC, with VNF scaling and migration performed directly to OpenStack:



An ILP has been designed and implemented to determine the servers in which the noisy VNF needs to be migrated (VNF migration actuation)



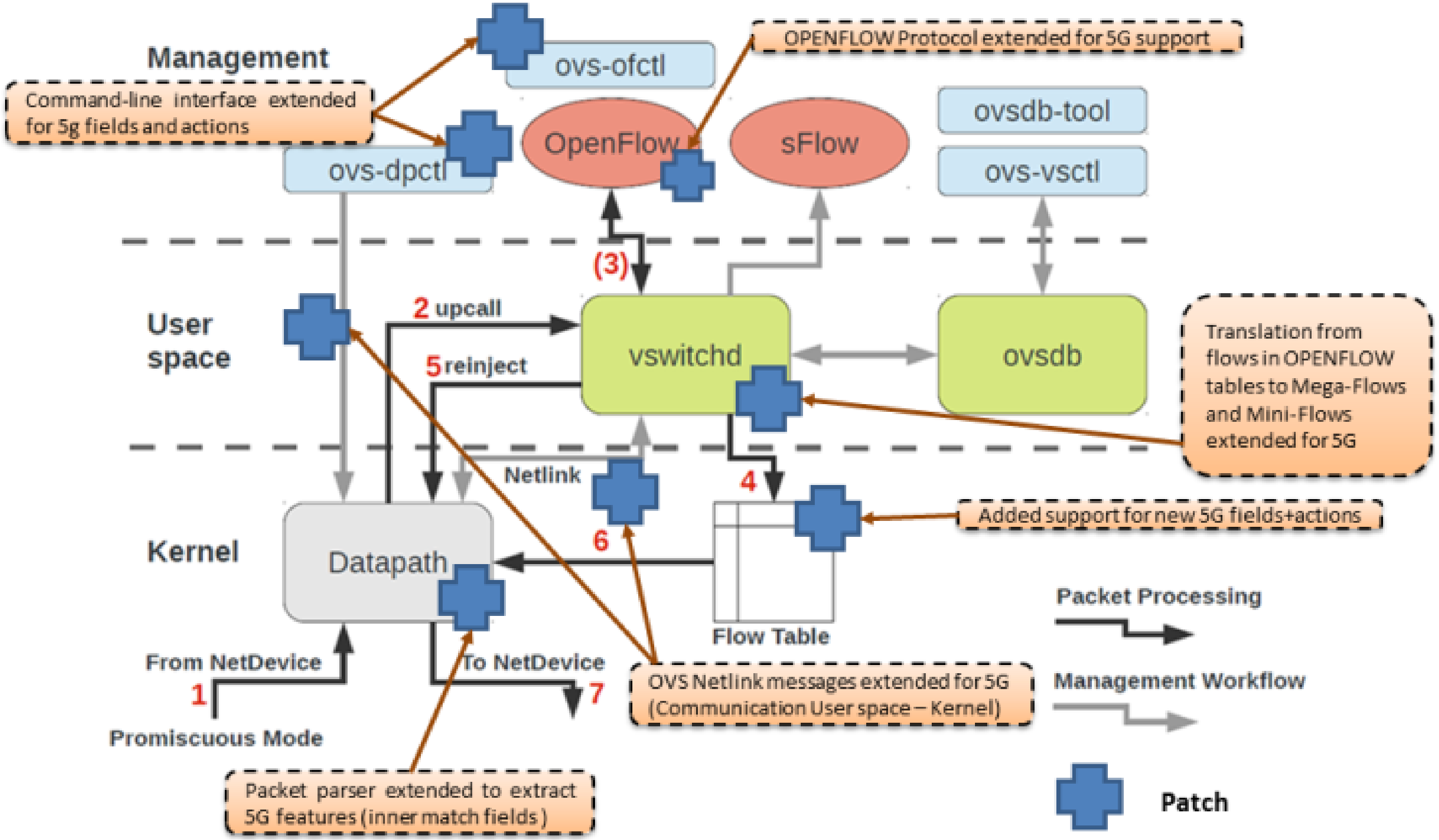
Gathered CPU utilization from Prometheus before and after the VNF scaling actuation. VNF 1 (orange) has been scaled by doubling the CPU cores assigned to it

OVS-based traffic classification

- Goal: Classify 5G traffic at the flow level for deployed slices at the NSPs
  - Traffic control and isolation is key for delivering slices with QoS guarantees at the NSP level. As such, flows must be properly identified and colored (classified)
  - Classified traffic can then be subject to specific control rules by other control/management functions.
  - The main enabler of this classification is an OVS-based classifier. The standard OVS architecture has been extended to incorporate the functionalities that allow for flow monitoring and classification.
- Usage: to allow for fine grain QoS control at the flows of slices
  - e.g. different rules, for instance, priorities, can be applied to classified traffic depending on its “tag”.
  - e.g. traffic re-direction or dropping.

OVS-based traffic classification

- Architecture of an enhanced OVS node with the patches for flow classification:



Why use cognition for slice QoE management?

- Scale
  - Must be **automated, autonomous**
- Flexibility, heterogeneity
  - Too many possible states, too many possible configuration
  - Must **generalize, comprehend**
- Dynamicity
  - Many workloads, dynamic traffic patterns
  - Need to constantly **adapt, anticipate**
- Abstraction, multi-layering, multi-domain
  - Multiple information sources, multiple owners, multiple semantics, partial data sharing
  - Must **combine sources, interpret, predict outcomes**
- E2E QoE per slice
  - Must derive QoE from QoS

Cognition Required (Traditional problem determination, e.g. thresholding, not adequate)

Innovations

- Multi-layer, multi-level
  - Physical, virtual net, sub-slice, domain, slice, P&P
  - Unified architecture at NSP and DSP levels
- Integrated monitoring
  - Telemetry and resource metrics
  - Traffic and flow-level
  - Topology
  - P&P capable (application specific)
- Slice aware, vertical in the loop
  - Vertically-informed QoE sensors
  - Cross-layer vertical context
- Multiple cognitions loops, knowledge sharing
  - NSP multi-slice, DSP multi-NSS, DSP multi-domain
- Cognitive-driven actuation
  - Cognitive-driven triggers
  - Cognitive-driven policy framework
  - Actuators de-coupled from triggers (reusable)
- AIOPs ready

Use cases, PD (ML Models), Remedial Actuation

Use Case	ML Model	Model Type	Remedial Actuation
Smart Grid	Predict RAN degradation and RAN failures from alarm data	Neural Network	Modify slice network parameters (bandwidth), Failover to new RAN
Smart City	**Detect performance degradation due to Noisy Neighbour**	**Random Forest**	**Bandwidth** **VNF scaling (VM Scaling),** **VNF migration (VM Migration)**
eHealth	**Anomaly Detection:** **observe network behavior for the last 5 minutes in order to forecast the signal strength degradation within the future 5 minutes.** **Data from UE P&P plug-in**	**Random Forest**	**Hand-Over,** **Traffic Re-direction**