A-CORD
(Analytics for CORD)

CORD
Central Office Re-architected as a Datacenter

#OpenCORD

Goal

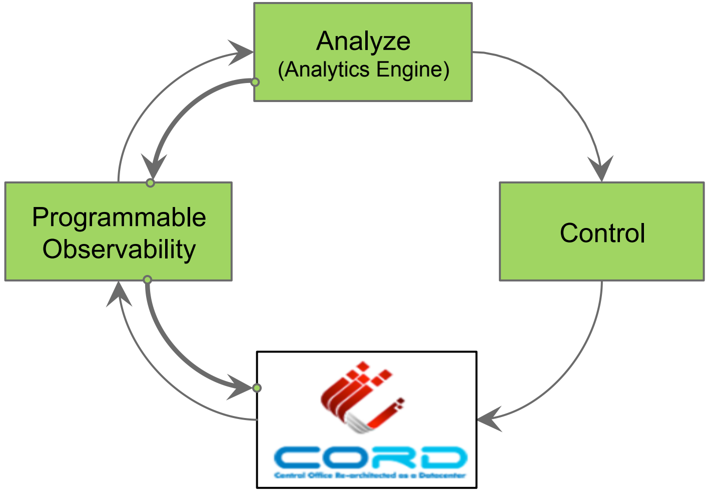
Enable programmable observability and closed loop control based on analytics

Examples:

- Fault correlation and Root cause analysis
- Fault prediction, Resource utilization prediction
- Anomaly Detection
- E2E SLA Violation Detection

Examples:

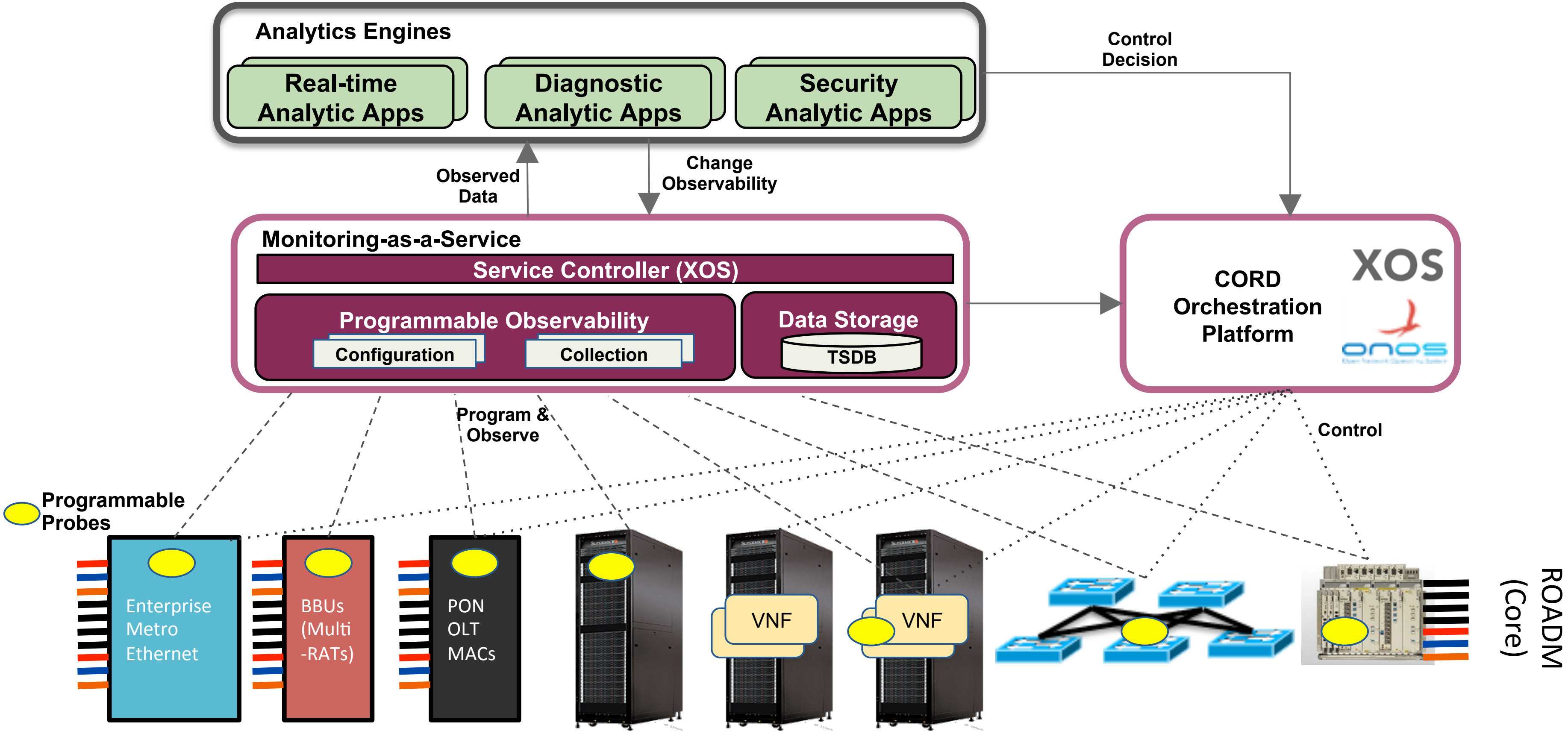
- Observe packet ‘inter-arrival-time’ of signaling traffic for a given time window and notify results
- Activate smart virtual probe and Observe S1-U traffic for a target subscribers for a given time and notify events


Analyze
(Analytics Engine)
Programmable Observability
Control
CORD
Central Office Re-architected as a Datacenter

Examples:

- Resource scaling & load balancing
- SDN Flow Rerouting (Traffic Engineering)
- Blacklisting Subscriber Traffic

A-CORD


Analytics Engines
Real-time Analytic Apps
Diagnostic Analytic Apps
Security Analytic Apps
Control Decision
Observed Data
Change Observability
Monitoring-as-a-Service
Service Controller (XOS)
Programmable Observability
Configuration
Collection
Data Storage
TSDB
CORD Orchestration Platform
XOS
onos
Program & Observe
Control
Programmable Probes
Enterprise Metro Ethernet
BBUs (Multi -RATs)
PON OLT MACs
VNF
VNF
ROADM (Core)

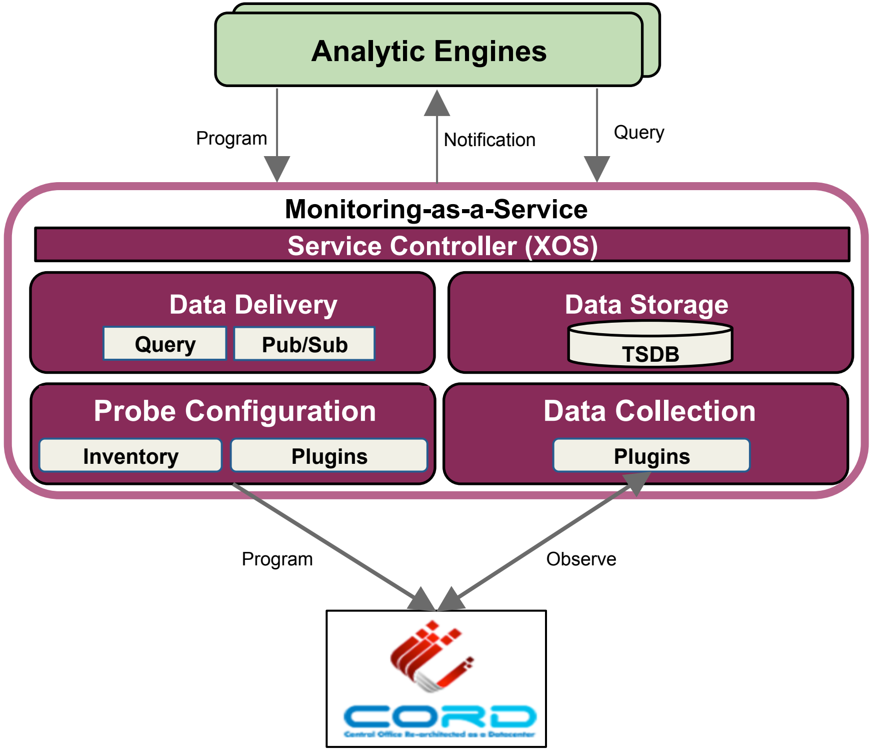
A-CORD

- Programmable probes: Infrastructure and Service level probes
  - Hardware building blocks: Access devices (OLT), Switches, Servers, ROADMs
  - Software building blocks: Service VNFs running in VMs/Containers
- Monitoring-as-a-Service
  - Decouples analytics engines from underlying targets
  - Unified interface to program and observe hardware and software probes
  - Scalable Time Series Data Store for traces of observed data and analysis results
- Analytics Engines
  - Performs real-time and batch analytic computing of collected data
  - Can be simple open source versions or sophisticated closed proprietary ones
  - *Examples*: Root cause analysis/Diagnostics, Resource optimization, Resource capacity predictions, Fault prediction
- Closed loop control
  - *Examples*: SDN enabled devices - Flow rerouting, Scaling and load balancing of services running in VMs/Containers

Monitoring-as-a-Service

- Scalable, Multi-tenant Service
- Leverages Micro-services Architecture

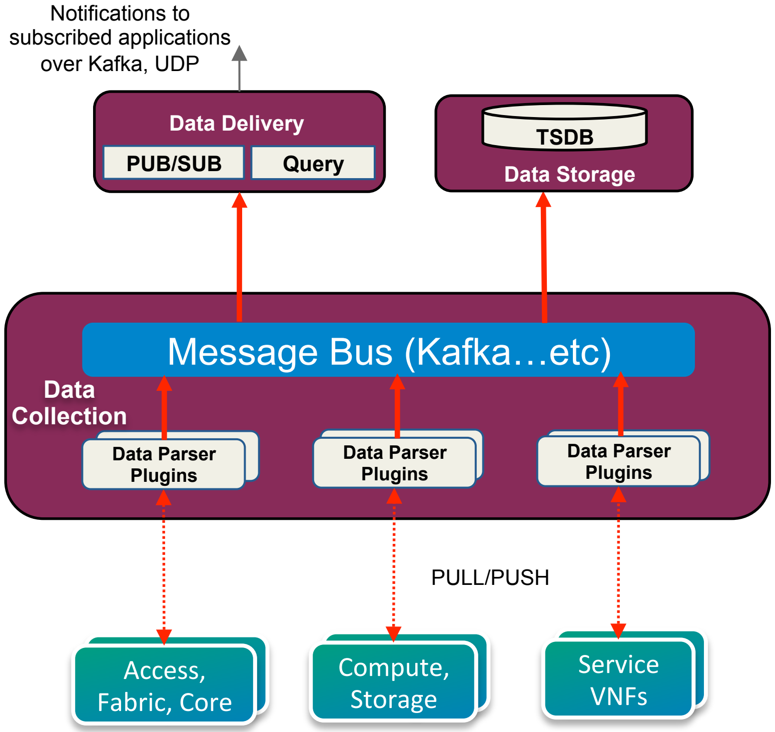
- Data Collection Service
- Probe Configuration Service
- Data Storage Service
- Data Delivery Services



*Data = Data Collected from Probes

Monitoring-as-a-Service: Data Collection Service

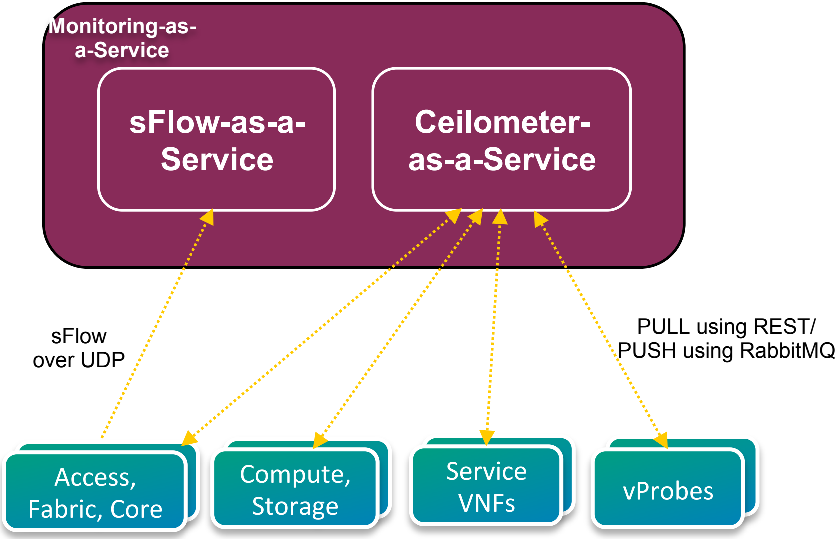
- A plugin based framework
- Event Parser Plugins
- Distributed Message Bus
- Both push and pull based data retrieval



Monitoring-as-a-Service: Proof-of-Concept

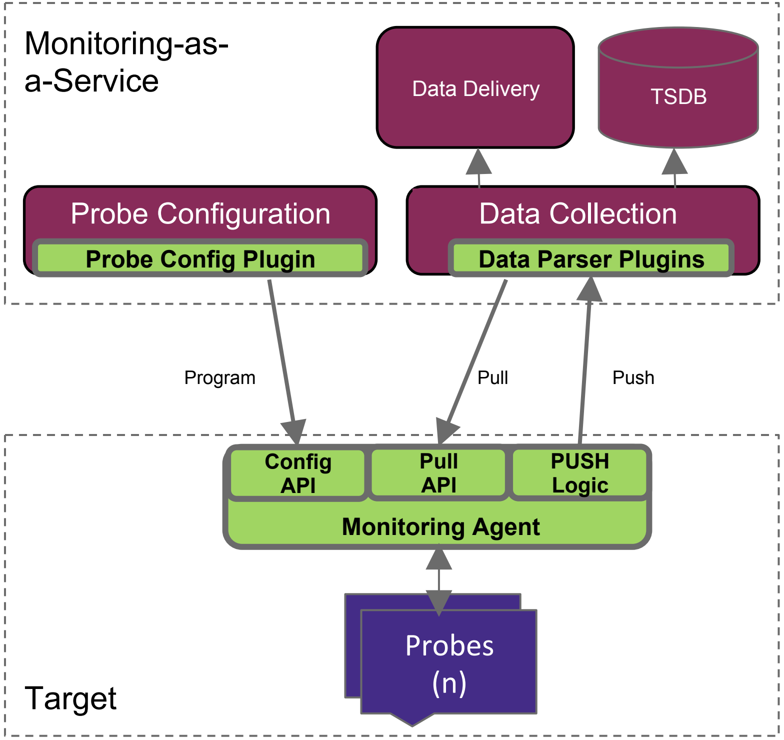
- To begin with, Monitoring-as-a-Service is implemented using:
  - Open Stack Ceilometer*
  - sFlow

*Leverages Ceilometer as a starting point, but we have decoupled Ceilometer from OpenStack and we have deconstructed Ceilometer into sub-systems that can be enhanced and replaced as necessary.



Programmable Probes

- Programmability of probes entails:
  - **Monitoring Agents** on the Target
  - **Collection Plugins** in Monitoring Service
  - **Probe Config Plugins** in Monitoring Service



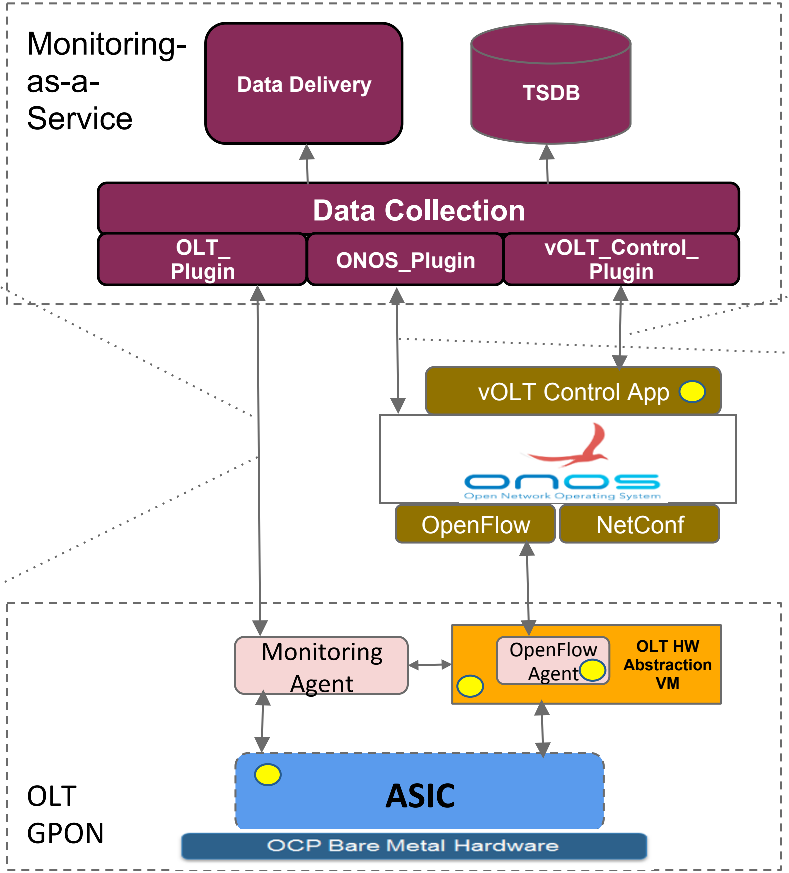
Programmable Probes: OLT Access Devices

Infrastructure Metrics

- instance.cpu
- instance.vcpu
- instance.memory.usage
- instance.network.incoming.bytes.rate
- instance.network.outgoing.bytes.rate
- instance.disk.usage
- instance.create
- instance.delete
- hardware.ipmi.node.cpu_util
- hardware.ipmi.node.mem_util
- hardware.ipmi.node.io_util

SoC Level Probes

- OLT.optical.interface.power.level
- **TBD**



vOLT Control App Events

- vOLT.active.subscribers
- vOLT.subscriber.connected
- vOLT.subscriber.disconnected
- **TBD**

OpenFlow Probes/Metrics

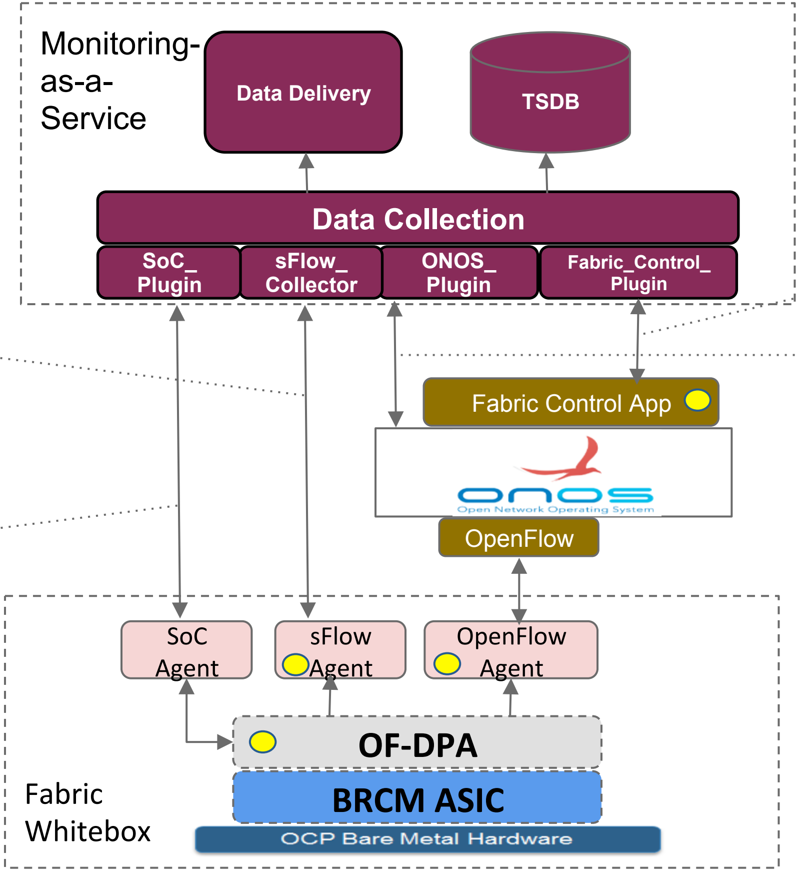
- switch.port.receive.packets
- switch.port.transmit.packets
- switch.port.receive.bytes
- switch.port.transmit.bytes
- switch.port.receive.drops
- switch.port.transmit.drops
- switch.port.receive.frame_error
- switch.port.receive.overrun_error
- switch.port.receive.crc_error
- switch.port.collision.count
- switch.table.active.entries
- switch.table.lookup.packets
- switch.table.matched.packets

● **Programmable Probes**

Programmable Probes: Fabric Switches

sFlow
- Packet Samples

SoC Level Probes
- **TBD**



Fabric Control App Events
- **TBD**

OpenFlow Probes/Metrics
- switch.port.receive.packets
- switch.port.transmit.packets
- switch.port.receive.bytes
- switch.port.transmit.bytes
- switch.port.receive.drops
- switch.port.transmit.drops
- switch.port.receive.frame_error
- switch.port.receive.overrun_error
- switch.port.receive.crc_error
- switch.port.collision.count
- switch.table.active.entries
- switch.table.lookup.packets
- switch.table.matched.packets

● **Programmable Probes**

Programmable Probes: Service VNFs

Service Level Events

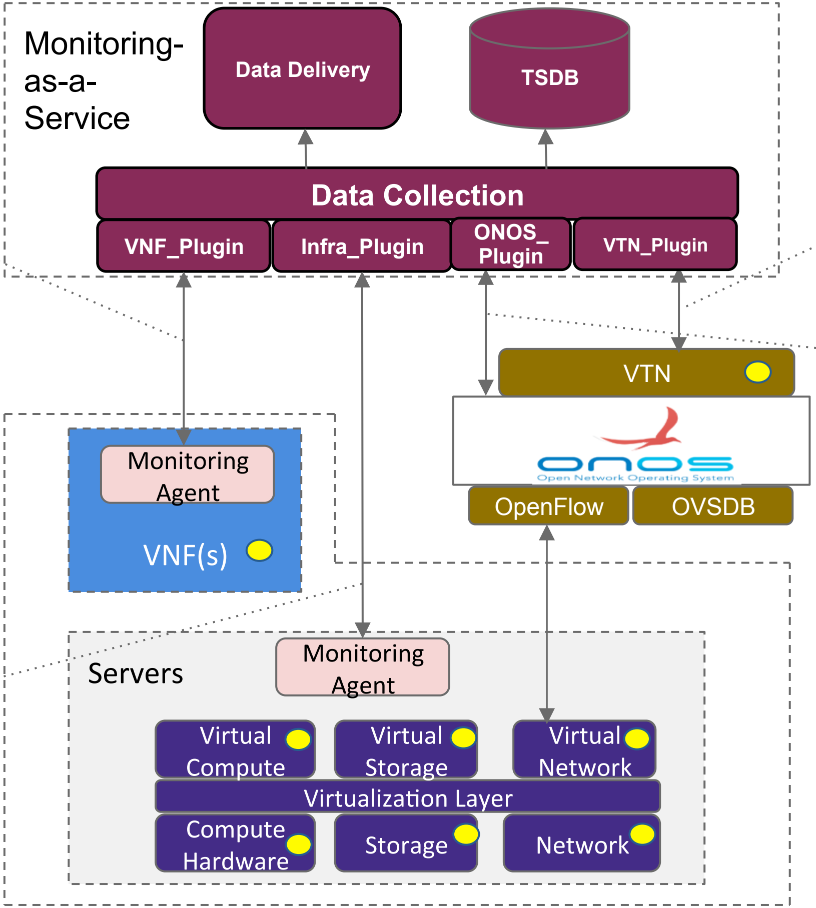
vSG
- vsg.dns.cache.size
- **TBD**

vMME
- **TBD**

vBBU
- vbbu.power.level
- **TBD**

Infrastructure Metrics

- instance.cpu
- instance.vcpu
- instance.memory.usage
- instance.network.incoming.bytes.rate
- instance.network.outgoing.bytes.rate
- instance.disk.usage
- instance.create
- instance.delete
- hardware.ipmi.node.cpu_util
- hardware.ipmi.node.mem_util
- hardware.ipmi.node.io_util



Virtual Tenant Networking App Events

- **TBD**

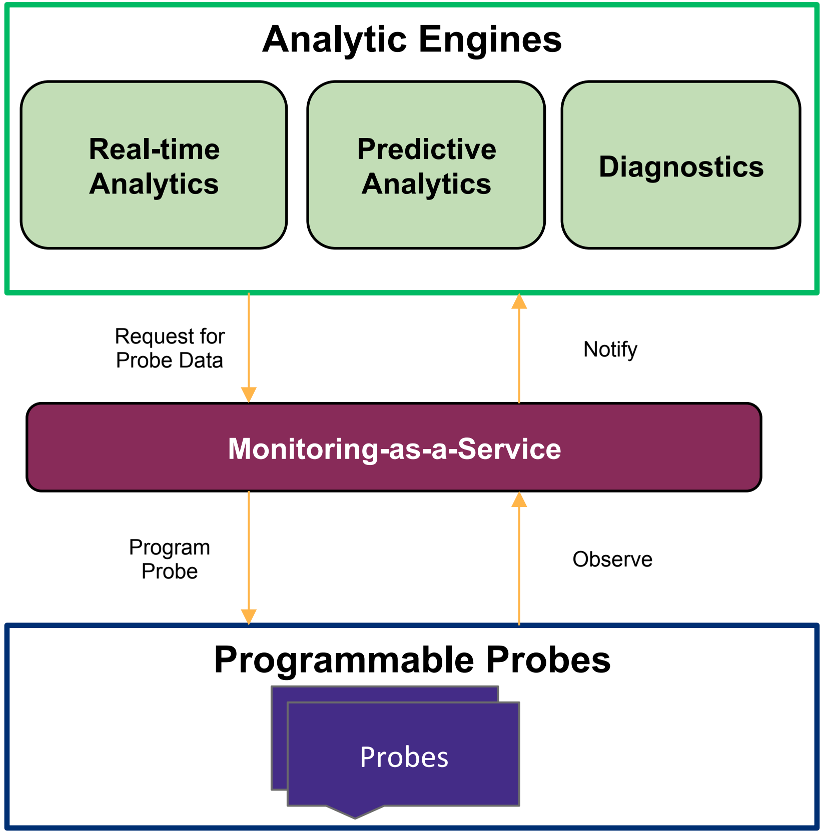
OpenFlow Probes/Metrics

- From Virtual Network Switches

● **Programmable Probes**

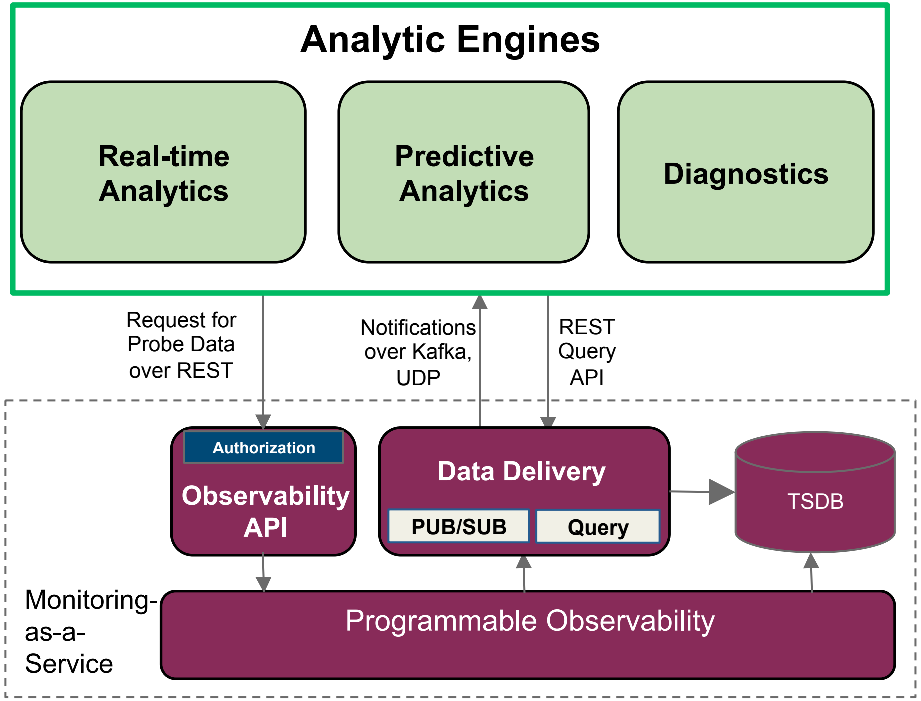
Analytic Engines

- A-CORD **decouples** analytics engines from underlying targets
- Diverse Analytics
  - Real-time Analytics
  - Diagnostics
  - Predictive Analytics
- Simple open source analytics and sophisticated closed proprietary analytics engines



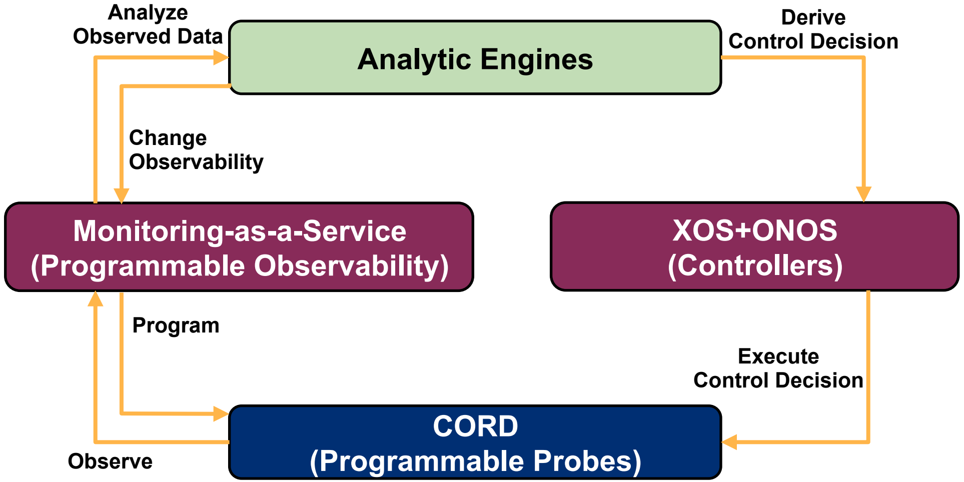
Analytic Engines - Interfaces

- Interfaces exposed by Monitoring-as-a-Service
  - Observability API
  - Query API
  - Data Notification
- Authentication to control access to probe data



Closed Loop Control

- “Closed Loop Control” is key in making networks Autonomic
- Derive control decisions
- Execution of control decision
- Dynamic change of observability



Smart Virtual Probe

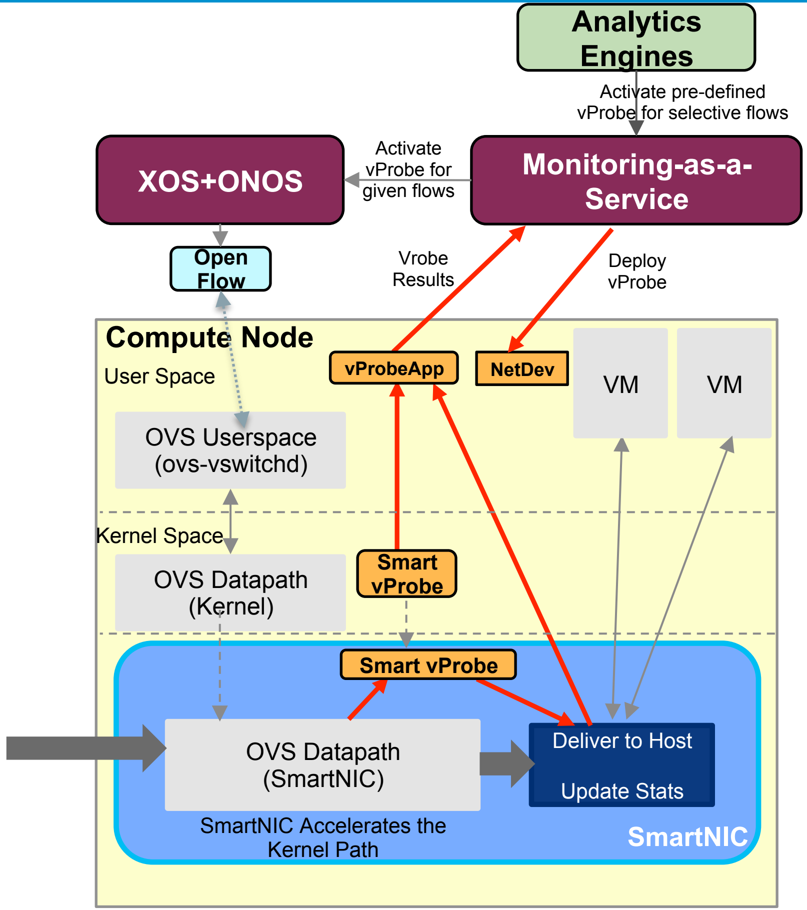
- Embedded programs that can perform data aggregation and simple analytics to reduce data volumes and enable low latency control loops.

- An application can dynamically download different programs depending on the state of the system

- Smart virtual probes can be deployed in physical devices (like SmartNICs, P4 enabled devices…etc.) or in virtual environment (inside VM/Container)

Smart Virtual Probes - SmartNIC

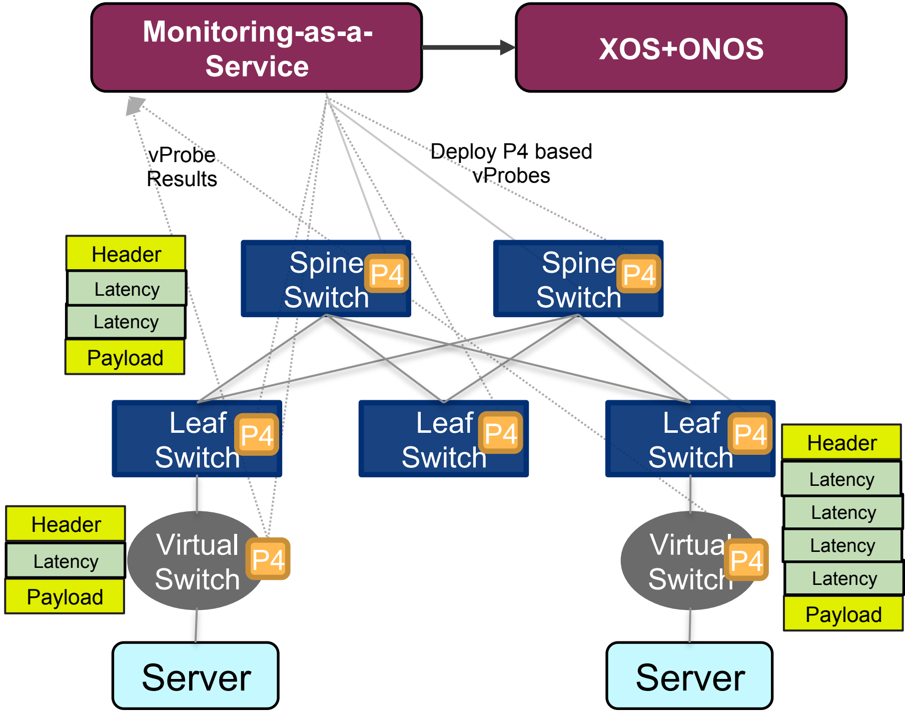
- Smart virtual probes can be computationally intensive
- Applications configure chain of smart virtual probes and download to the kernel using eBPF C programs or P4 programs
- These programs can then be pushed to the SmartNIC
- Openflow custom action to identify flows to send to virtual probes



Smart Virtual Probes – P4 Enabled Switches

- Custom P4 programs recording latencies at each hop in data path

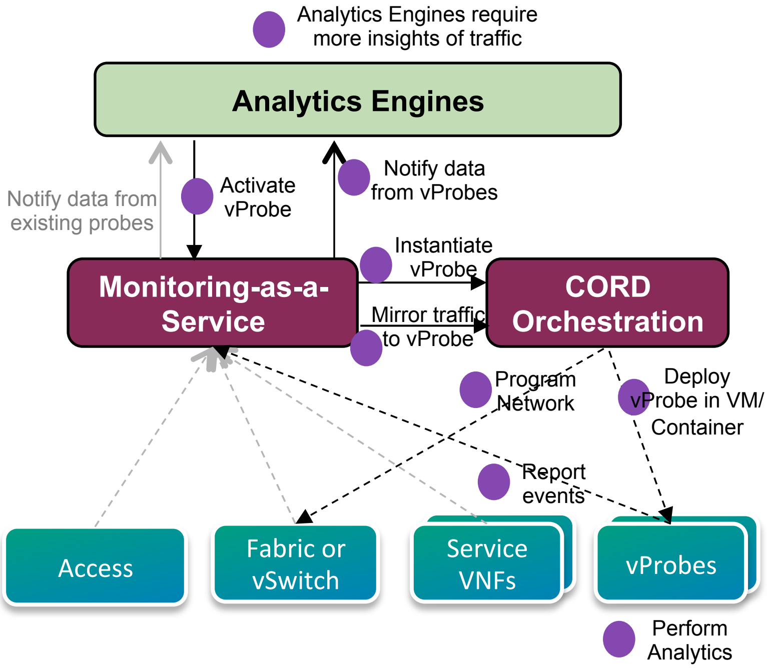
- This can then be used to identify a faulty device within the switch fabric



Smart Virtual Probes – Compute Intensive Programs

- Deploy compute intensive virtual probes such as DPI in the user space VM/Container

- Mirror the traffic from specified flows (match condition)



Active Testing

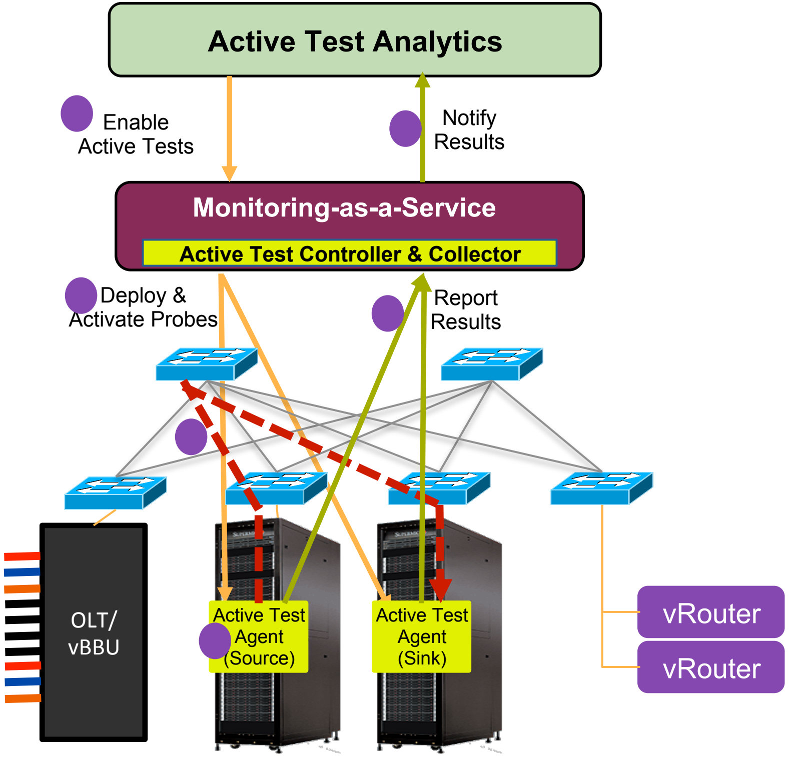
- Passive Monitoring vs Active Testing

- Example objectives of Active Testing
  - Reproduce & diagnose the customer care issues and E2E network connectivity issues
  - To analyze the round-trip performance, latency between devices or services
  - To verify E2E SLAs are being met

- A-CORD enables deployment of software-based Active Testing agents as programmable probes inside CORD.

Active Testing in A-CORD

- Active Test Controller
- Active Test Agents
- Active Test Collector



Active Testing: CORD Use Cases

- Fabric Connectivity, Throughput Validation & Congestion detection
- E2E Subscriber Internet Connectivity and SLA validation
- Certification of New CORD Services
- Monitor Video Quality of Experience

Current Implementation

- Proof-of-Concept
  - Implementation of Monitoring-as-a-Service using openstack ceilometer and sflow based data collection, storage and delivery mechanisms in CORD development (cloudlab & devstack) environment
  - Integration with existing probes in CORD (Compute & Hypervisor, ONOS SDN controller, vSG service VNF)
  - Implementation of open source Service aware Auto-scaling analytics application and Residential Subscriber diagnostics portal on top of A-CORD
  - Integration of A-CORD platform with 3rd party analytics applications

Roadmap (1/3)

- Programmable Probes
  - Hardware
    - OLT GPON
      - Implement OpenFlow metrics
      - Define and implement ASIC level Metrics (power_levels…etc)
      - Expose metrics of ‘OLT Hardware Abstraction’ VM
    - Fabric whitebox
      - Implement sFlow agent
    - ROADM
      - Implement OpenFlow Metrics
      - Implement sFlow agent

Ensure each hardware element implements minimum set of programmable probes before they are introduced in CORD

Roadmap (2/3)

- Programmable Probes
  - Software
    - vSG VNF
      - Define and implement subscriber traffic events
      - Define and implement subscriber service events
    - vOLT ONOS Application
      - Define and implement subscriber connection events
    - vRouter ONOS Applications
      - Define and implement subscriber internet connectivity events
    - VTN and Fabric controller ONOS Applications
      - Define and implement metrics
    - Active Testing
      - Active test agents for fabric connectivity
    - Virtual Probes
      - “Inter packet arrival” & “Jitter” virtual probes to be pushed to OVS kernel data path or SmartNICs

Ensure each service VNF implements minimum set of programmable probes before they are introduced in R/M/E CORD

Roadmap (3/3)

- Monitoring-as-a-Service
  - Deployment of Monitoring-as-a-Service using Ceilometer and sFlow in CORD POD
  - Replace Ceilometer components with OpenStack Monasca based API and data storage subsystems to improve query API performance
  - Explore Ceilometer Gnocchi framework
  - Load balancing framework for Ceilometer event parser plug-in subsystem
  - Containerize publish/subscribe delivery service
  - Framework for programmability of probes
  - Integration of Monitoring Service with new probes
  - Framework for smart vProbes and Active Testing
- Analytics Engines
  - Visualization (Dashboard)
  - Fault correlation and Root cause analysis for service VNF faults
  - Fabric network connectivity and throughput validation using Active Testing Probes
  - End-to-End SLA violation detection using Active Testing Probes
- Closed Control Loops
  - DDoS detection and mitigation
  - For more use cases of closed control loops, refer to R/M/E-CORD slides

Summary

- A-CORD enables programmable observability and closed loop control based on analytics
  - Fully exploits SDN & NFV
  - Fully exploits Micro-Services
  - Monitoring-as-a-Service provides programmable observability
  - Analytics Engines derive intelligent control decisions from the collected data
  - CORD Orchestration layer executes the control decisions