



中国移动
China Mobile

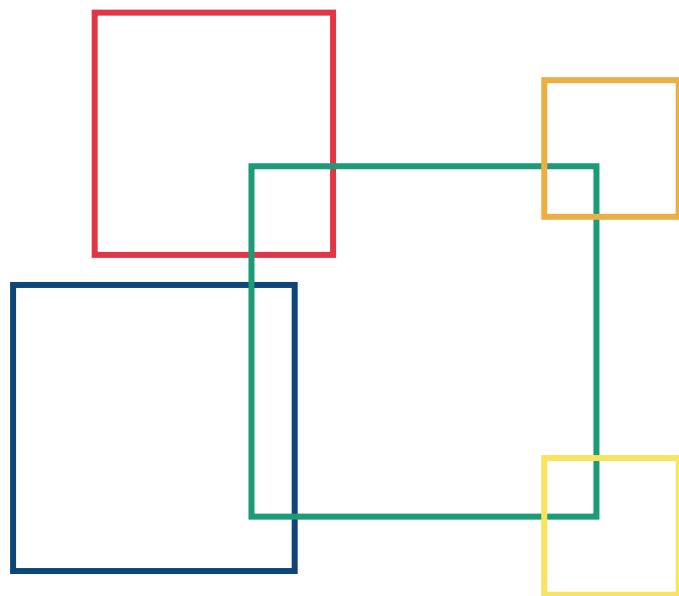
Open Networking Automation Platform (ONAP) Policy Management Framework & Usecase

Keguang He, Yan Yang (China Mobile)

ENI Technology Workshop on Policy Management

2020-09

www.10086.cn



- 01. Introduction to ONAP policy framework and APEX policy execution engine
- 02. Example of ONAP policy: VoLTE Closed Loop

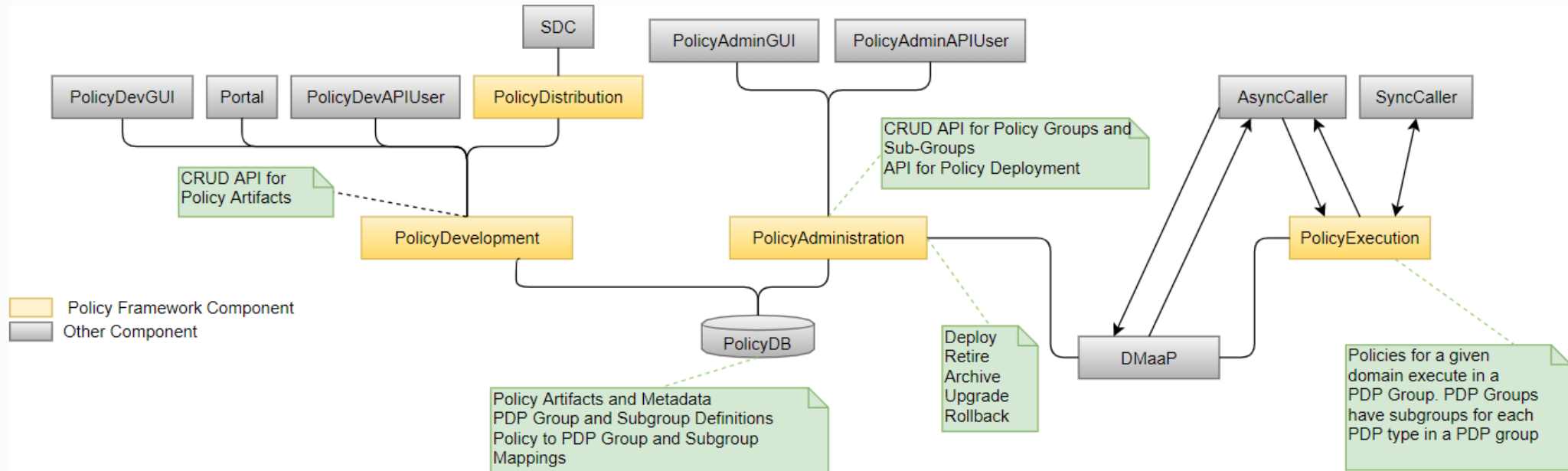
ONAP policy framework has the following five capabilities

- It can be capable of being triggered by an event or invoked, and making decisions at run time.
- It can be deployment agnostic; capable of managing policies for various Policy Decision Points (PDPs) or policy engines.
- It can be metadata driven, allowing policies to be deployed, modified, upgraded, and removed as the system executes.
- It can provide a flexible model driven policy design approach for policy type programming and specification of policies.
- It can be extensible, allowing straightforward integration of new PDPs, policy formats, and policy development environments.

Introduction to ONAP policy type

- A Policy Type describes the properties, targets, and triggers that the policy for a feature can have.
- Policy Types are hierarchical, A Policy Type can inherit from a parent Policy Type, inheriting the properties, targets, and triggers of its parent.
- A policy is defined using a Policy Type, including the values for each property of the policy type, the specific targets (network element, function, service, resource) on which this policy will act, and the specific triggers that trigger this policy.
- A Policy Type Implementation is the logic that implements the policy. When one of the triggers described in the Policy Type occurs. The corresponding logic will be executed and acts on the targets specified in the Policy Type.

ONAP policy framework architecture



PolicyDevelopment

- Provides a CRUD API for policy types and policies.
- Other applications can use the API to create, update, delete, and read policy types and policies.

PolicyAdministration

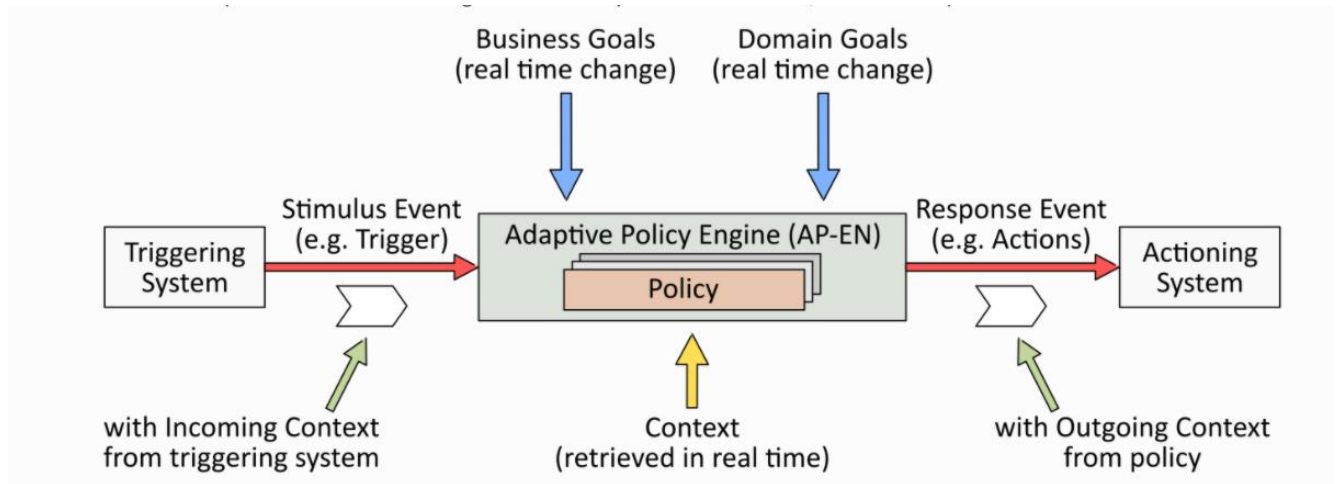
- Management of the life cycle of PDPs in an ONAP installation.
- Management of the deployment of policies to PDPs in an ONAP installation.

PolicyExecution

- The set of running PDPs that are executing policies, logically partitioned into PDP groups and subgroups.
- The PDP supported by ONAP includes XACML, DROOLS and APEX.

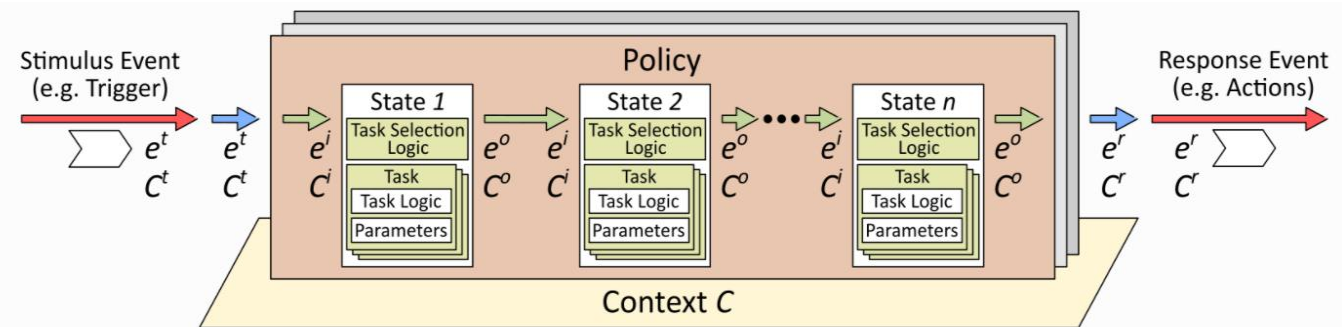
Simple APEX Overview

- Stand for Adaptive Policy EXecution. It is a lightweight engine for execution of policies.
- Specify logic as a policy that can adapt on the fly as your system executes.
- Policies are triggered by incoming events. The logic of the policies executes and produces a response event.
- You can design the policies that APEX executes and the trigger and action events that your policies accept and produce.



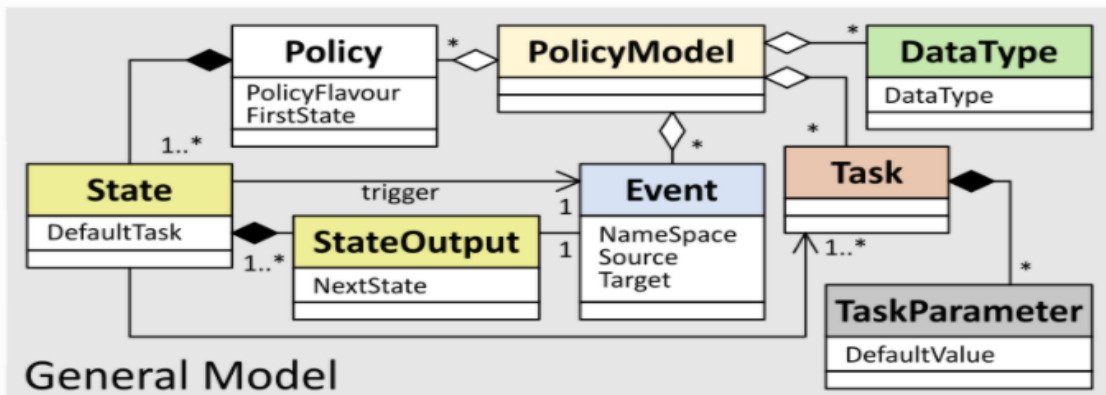
APEX States and Context

- Design your policy as a chain of states, with each state being fed by the state before.
- All States together form a directed acyclic graph.
- Context is simply the state information and data used by your policies.
- APEX takes care of distribution, locking, writing of context to persistent storage, and monitoring of context.

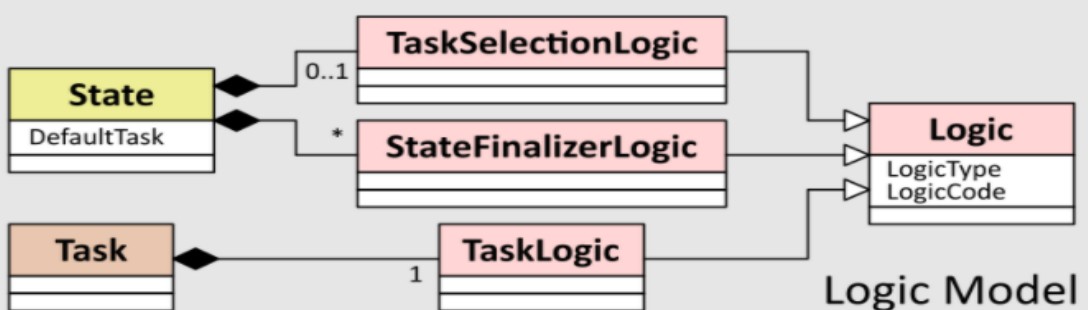
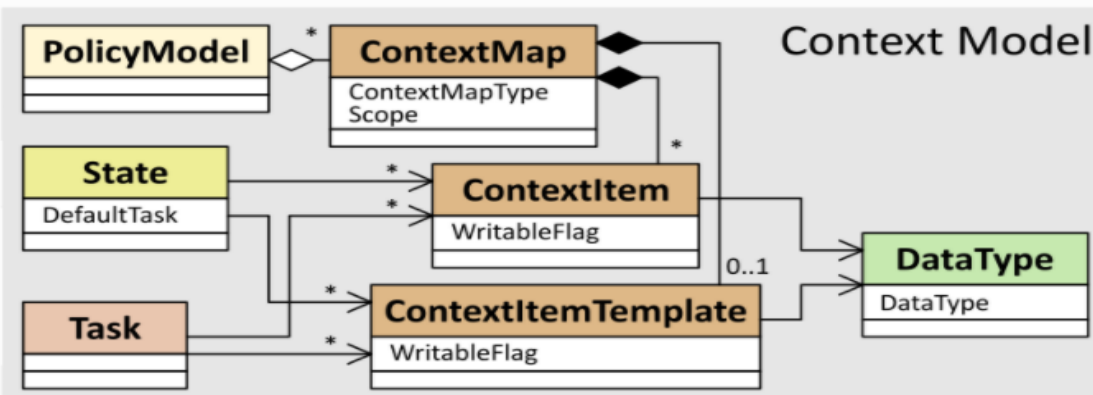


APEX Policy Model

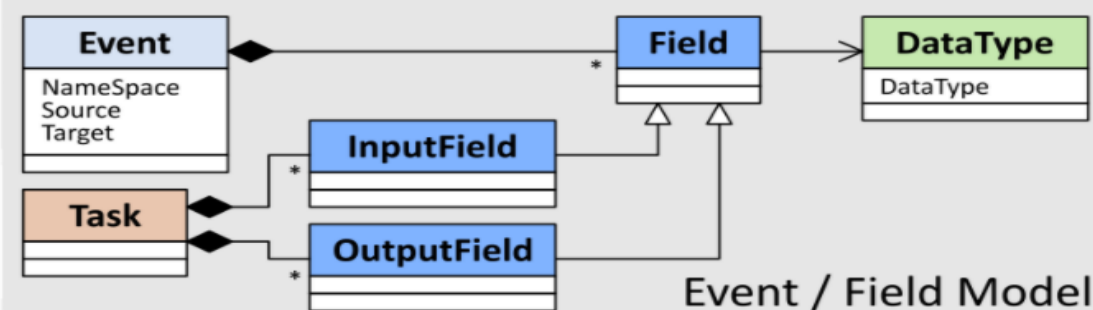
Shows the main parts of a policy: state, state output, event, and task.



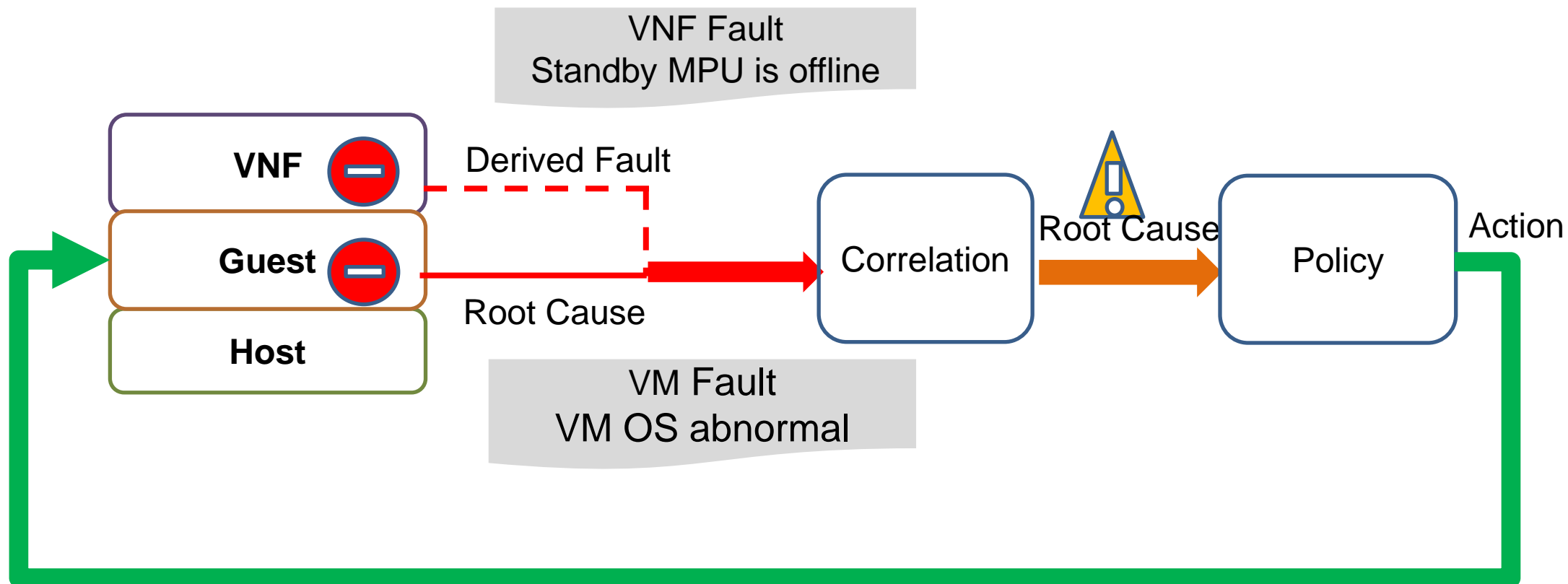
Shows how context is injected into a policy. States collect context from tasks. Task define what context requires for decision making. Context is a collection of items with data types.



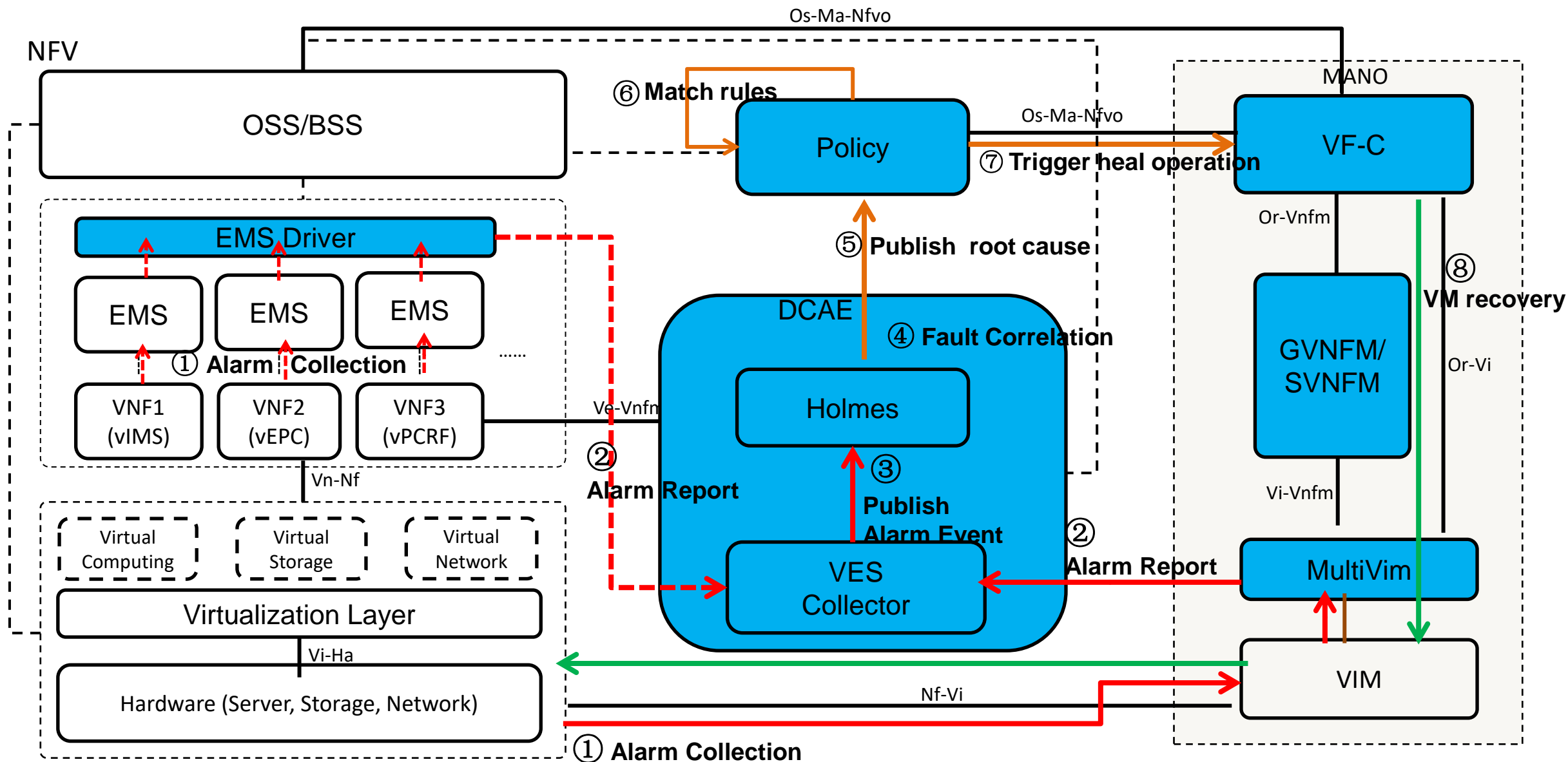
Emphasizes how decision-making logic is injected into a policy. Three types: task logic, task selection logic, state finalizer logic.



Shows the events in the policy model. Tasks define what information they consume (input) and produce (output). This information is modeled as fields, Events are collection of fields.



Part02— Interaction with ETSI NFV-MANO



Policy Engine : Drools

Policy Type: Operational Policy

```
{
  "PolicyType": "operational"
  "PolicyRule": {
    "Condition": "ControlLoop-VOLTE-2179b738-fd36-4843-a71a-
a8c24c70c55b",
    "Event": "ONSET",
    "Action": {
      actor: "VFC"
      recipe: "RestartVM"
    }
  }
}
```

Onset Event Info

VF-C Rest Invocation

```
{
  "target": "vserver.vserver-name",
  "target_type": "VM",
  "AAI": {
    "vserver.is-closed-loop-disabled": "false",
    "vserver.vserver-id": "example-vserver-id-val-64888",
    "vserver.vserver-name": "example-vserver-name-val-75044",
    "generic-vnf.vnf-id": "example-vnf-id-val-31366",
    "service-instance.service-instance-id": "example-service-id-val-
33989"
  },
  "closedLoopAlarmStart": 1413378172000000,
  "closedLoopEventStatus": "ONSET",
  "closedLoopControlName": "ControlLoop-VOLTE-2179b738-
fd36-4843-a71a-a8c24c70c55b",
  "target": "vserver.vserver-name",
  "requestID": "a0d549da-cd11-4425-af1f-fa40fd44ff",
  "from": "DCAE"
}
```

```
POST http(s)://[hostname][:port]/api/ns lcm/v1/ns/example-service-id-val-
33989/heal
Request JSON:
{
  " healVnfData": {
    "vnfInstanceId": "example-vnf-id-val-31366",
    "cause": "vm is down",
    "additionalParams": {
      "action": "restartvm",
      "actionvminfo": {
        "vmid": "example-vserver-id-val-64888",
        "vmname": "example-vserver-name-val-75044"
      }
    }
  }
}
```

THANK You

hekeguang@chinamobile.com