SOFTFIRE

SoftFIRE: Constructing a Federated and Orchestrated Multi-Testbed Virtualisation Infrastructure

March 31, 2017

Why do we need FIRE APIs (T1.2)

- Resource discovery
- Experiment definition
- Resource reservation
- Resource provisioning
- Experiment control during execution
- Experiment Monitoring
- User Authentication
- User Authorization

Current status

- SoftFIRE enabled access to the platform adopting the Slice Federation Architecture (SFA) APIs
  - SFA main objective is to provide access to federated testbeds exposing heterogeneous resources
  - RSpec Schemas used for describing resources

- The T1.2 dealing with those activities was planned up to M6
  - Effort allocated to this task allowed extensions to the FITeagle tool for supporting provisioning mainly of network services deployed via Open Baton

Limitations encountered

- SFA/RSpec is not NFV/SDN oriented:
  - Resources could be of any type, however it is most of the time required an adapter for translating from SFA to a specific information model
- Limited programmability of the actual resources:
  - Perfect for basic virtual resources (i.e. VMs)
  - Not the best choice for NFV resources (VNF, NSD)
- Any modifications to the underneath infrastructure requires changes to multiple layers of the SoftFIRE middleware

Feedbacks from experimenters (call 1)

- Jfed does not provide enough information in case of ERRORS happening at the infrastructure level while deploying. Troubleshooting costs were very high for both experimenters and SoftFIRE team.

- Some experimenters preferred the MANO APIs directly since they were providing more powerful capabilities:
  - The MANO APIs already federate an indefinite number of testbeds plus they provide to the experimenter the flexibility and the freedom to upload their (shared/private) resources

Experiences from past FIRE projects

- Some FIRE projects are providing direct access to the API of an individual testbed (limited potentiality)
- SFA/RSPec used mainly in very heterogeneous testbeds (different technologies provided in each individual testbeds)
- Based on our knowledge there are very limited number of commercial SFA-enabled federated testbeds
- BonFIRE → NO SFA, OCCI-based → Very successful also in sustainability
- OPNFV Pharos-labs example of industry-oriented NFV federated testbeds → https://wiki.opnfv.org/display/pharos/Pharos+Home

Our plan

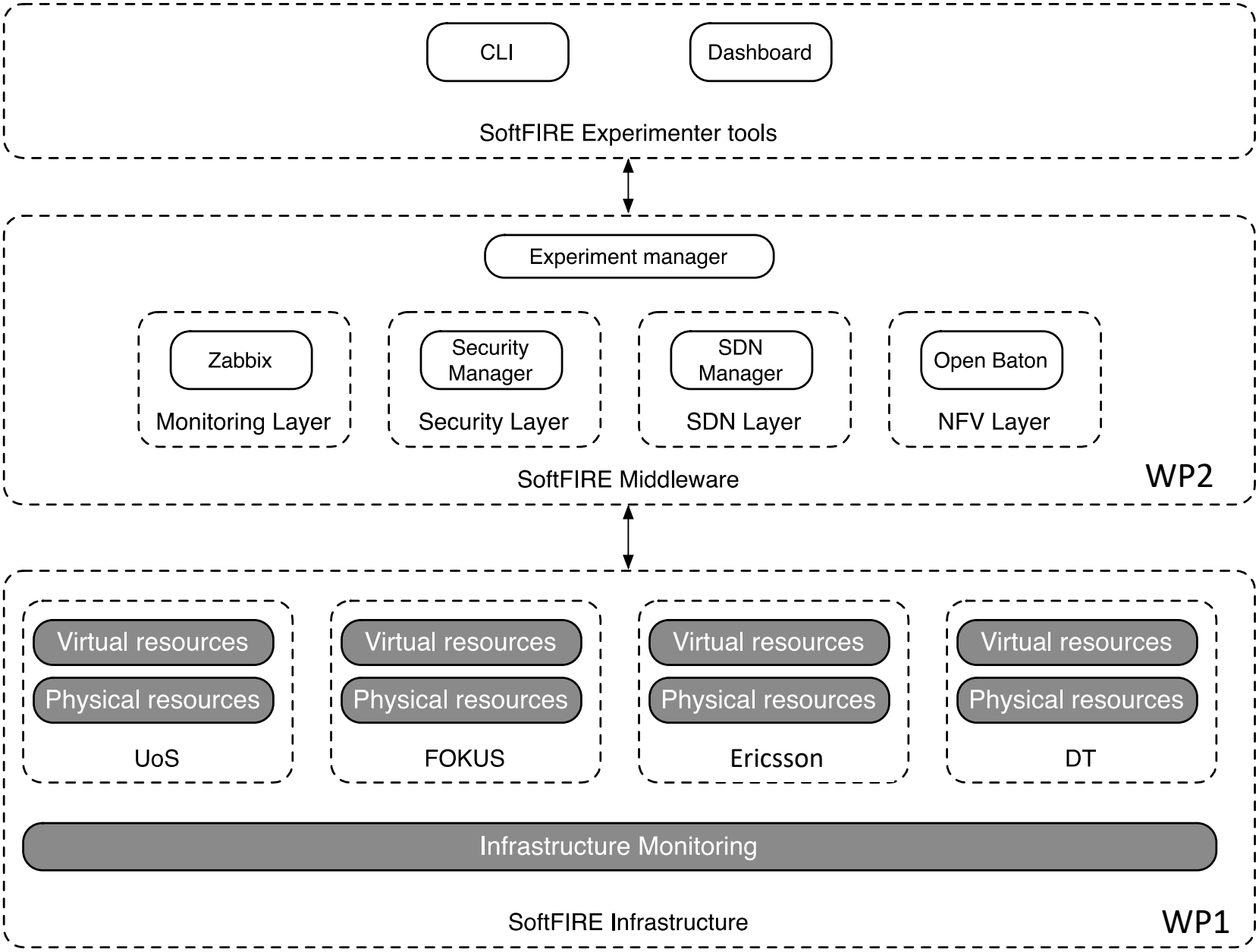
- Deprecate the SFA/RSPec APIs providing access to SoftFIRE via RESTful TOSCA APIs exposed by a new layer called Experiment Manager (part of the SoftFIRE middleware):
  - TOSCA is widely used in the NFV community
  - Federation and NFV resource provisioning achieved using Open Baton and interconnectivity between testbeds
  - Provisioning of different kind of resources using specific managers that are under the control of the Experimenter manager
- Access provided to experimenters mainly via a single portal and RESTful APIs

SFA VS Tosca

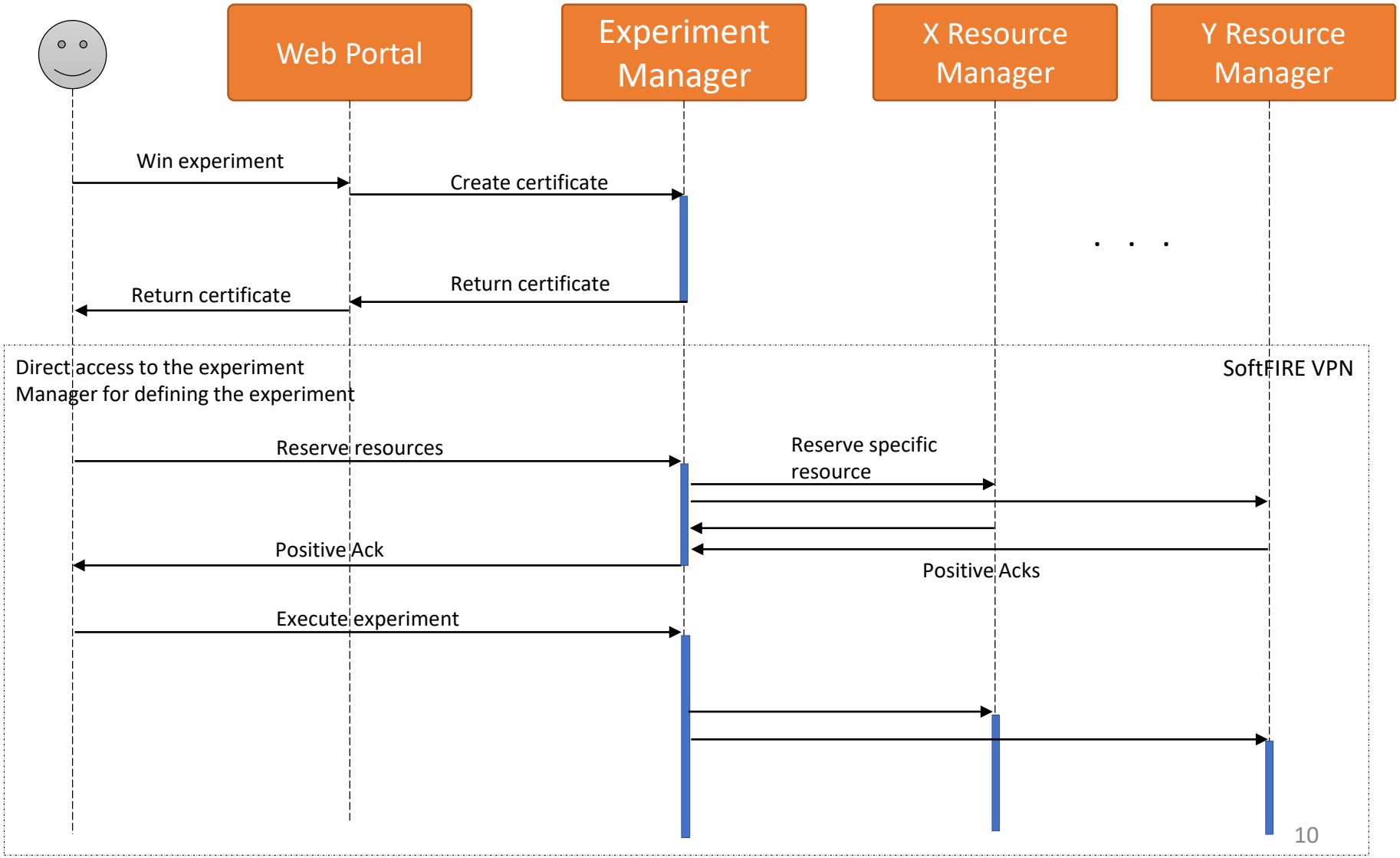
```
<node client_id=”dummy-server" exclusive="false"
component_manager_id="urn:publicid:IDN+localhos
t+authority+cm"
component_id="urn:publicid:IDN+localhost+node+ht
tp%3A%2F%2Flocalhost%2Fresource%2FOpenBaton-
Server-1+Gateway">

  <sliver_type name="http://open-
multinet.info/ontology/resource/openbaton#dummy
-server" />

<interface client_id=”dummy-server:if0">
</node>

…

<link client_id="link1">
  <component_manager
name="urn:publicid:IDN+localhost+authority+cm" />
  <interface_ref client_id="Gateway:if0" />
  <interface_ref client_id="ENodeB:if1" />
  <link_type name="lan" />
</link>
```

```
dummy-server:
        type: openbaton.type.VNF
        properties:
          vendor: Fokus
          version: 0.1
          endpoint: dummy
          type: server
          configurations:
            name: config_name
            configurationParameters:
              - key: value
              - key2: value2
          vnfPackageLocation: https://github.com/openbaton/
          deploymentFlavour:
            - flavour_key: m1.small
        requirements:
          - virtualLink: private
          - vdu: VDU2
        interfaces:
          lifecycle: # lifecycle
            instantiate:
              - install.sh
              - start-srv.sh
```

Proposed SoftFIRE v2 architecture



Proposed Scenario (v2 architecture)



Benefits

- Flexible middleware for managing NFV/SDN technologies based on industry oriented open APIs (TOSCA)
- Integration of security and monitoring capabilities
- Possibility of integrating also physical resources
- Provide the means to the experimenters that choose only particular kind of resources in a particular location for a dedicated amount of time