

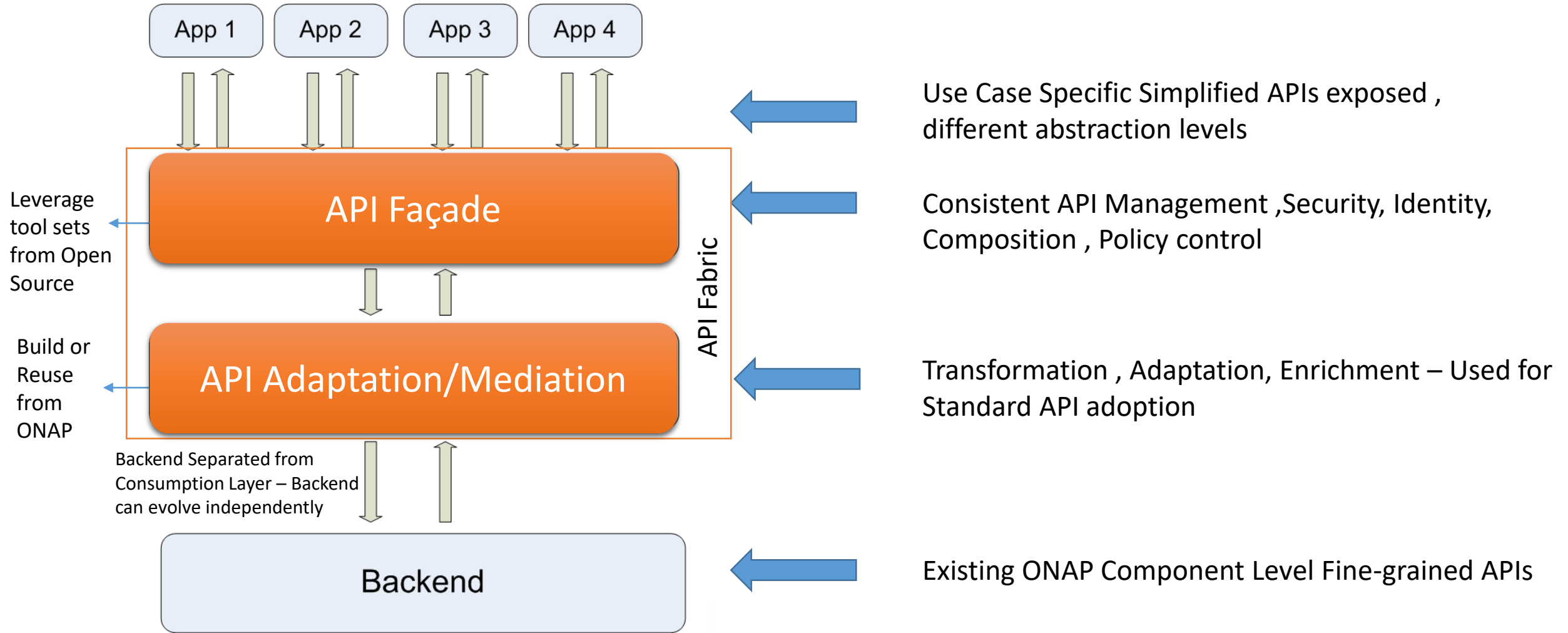


Operational Lifecycle Enabler

Use Case Based on API Fabric Proposal and other PoC Ideas

June, 2019

API Fabric : Re-Cap From F2F Event



Handle common API transactional requirements at Façade, Focus on integration of adaptation logic in Mediation/Adaptation

Agenda

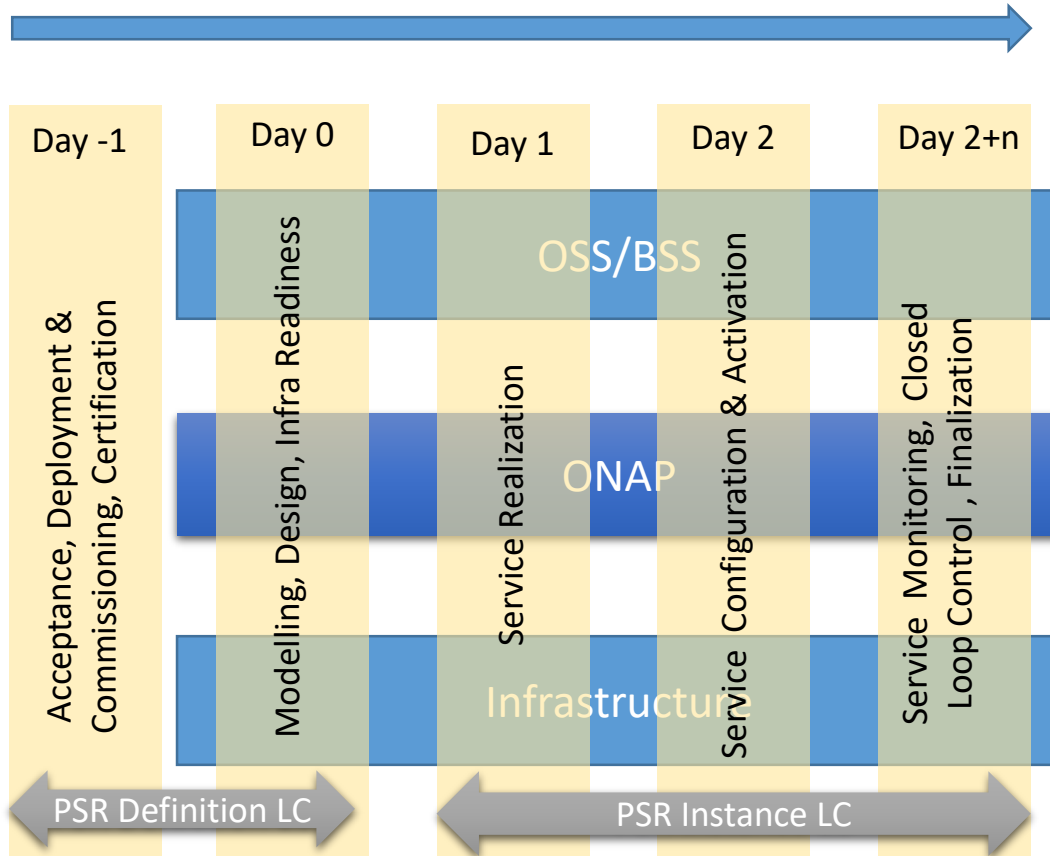
- Use Case Overview
- Proof of Concept Plan



Use Case Overview

Brief Context

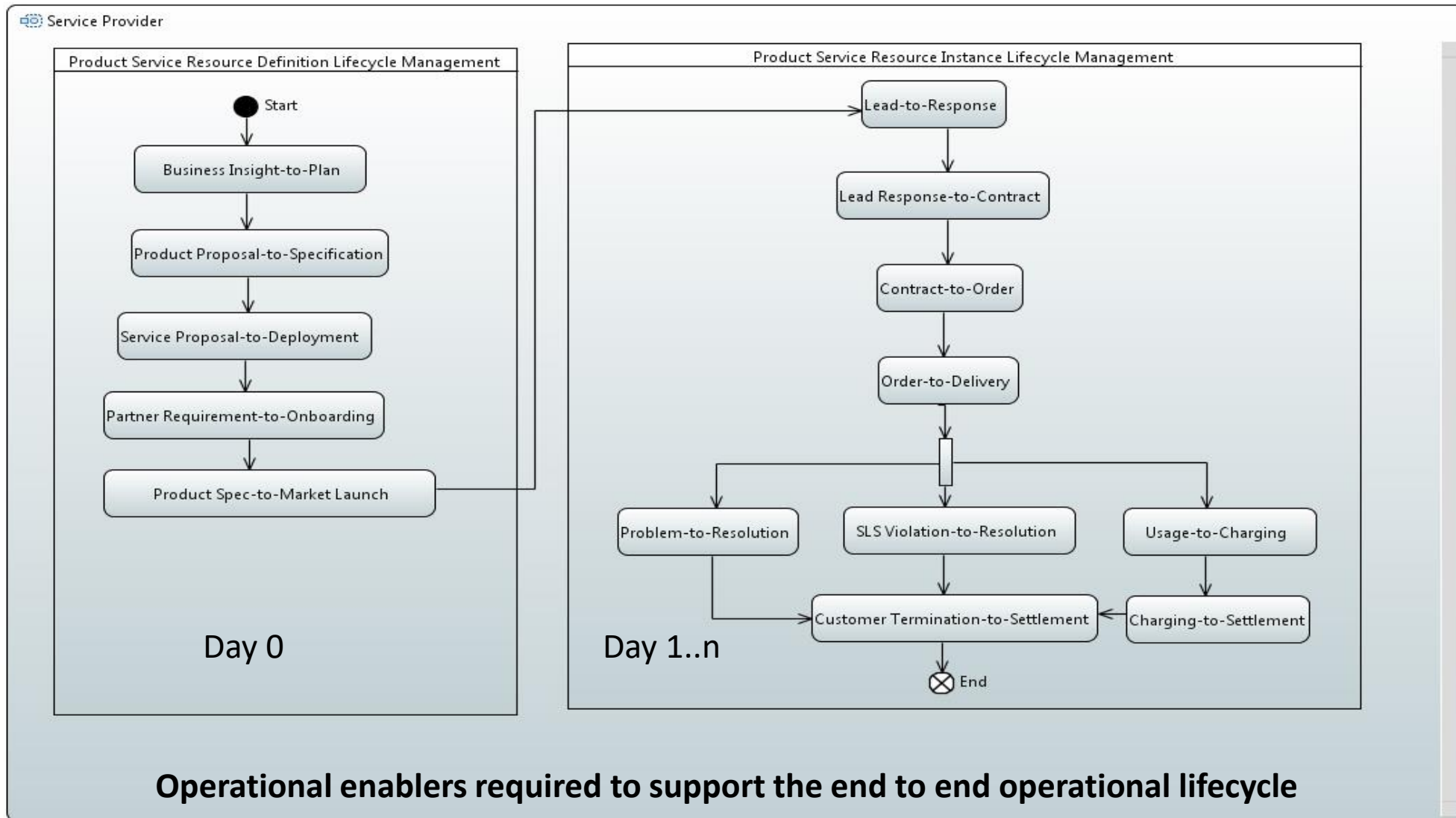
Operational Life cycle



Use Case Context

- In a typical Telecom Operator production environment the deployed software solutions/systems goes through different operational lifecycle stages , or processes based on a well defined business process flows
- This is typically classified in terms of Day 0,Day 1 and Day 2 Operations or PSR Definition/PSR Instance Lifecycle Operations
- The operational activities cut across different layers of solutions and systems.
- From ONAP perspective the key enablers available currently for supporting such operational activities are set of granular APIs exposed by ONAP components
- The proposed use case suggest a more generic approach for exposing ONAP operational capabilities through a function by name API Fabric which exposes a façade layer in Ext-API rather than deep integration with individual ONAP components.

MEF PSR Definition and PSRI Lifecycle – Ref MEF 50.1



MEF LSO Operational Threads

Operational Threads describe the high level Use Cases of LSO behavior as well as the series of interactions among LSO management entities, helping to express the vision of the LSO capabilities. The interactions de-scribed within each Operational Thread address the detailed involvement of the Interface Reference Points in the LSO Reference Architecture.

Reference : MEF 55.1

- Partners on-boarding
- Product Ordering and Service Activation Orchestration (Partially Supported in Ext-API)
- Controlling a Service
- Customer Viewing Service Performance and Fault Reports and Metrics
- Placing and Tracking Trouble Reports
- Assessing Service Quality Based on SLS
- Collection and Reporting of Billing and Usage
- Securing Management and Control Mechanisms
- Providing Connectivity Services for Cloud

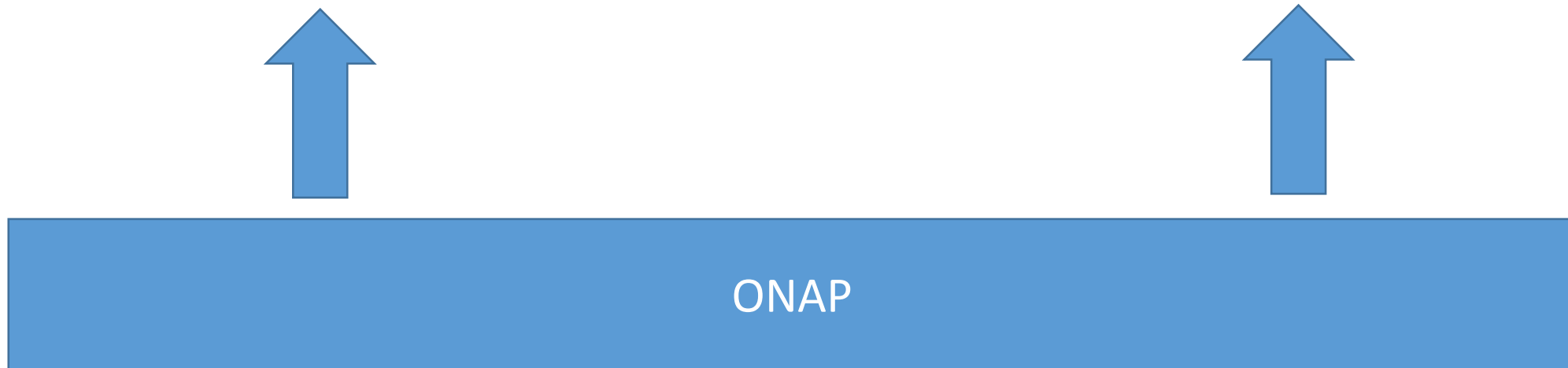
Operational Lifecycle : How ONAP Services are Consumed in Production

Consumption through Standard interface

- TMF, ETSI, MEF, ONF, IETF based interfaces depending on ONAP deployment context
- Each standard based on a specific set of managed object model and standard set of operations on managed objects
- Mostly used for integration

Consumption based on custom interface

- ONAP Proprietary i/f or Consumer Proprietary i/f
- No fixed pattern for interfaces or Managed objects
- Mostly used for consuming either ONAP internal Services or integration with custom operator solutions
- Expect to operate on a unified/normalized/abstracted set of managed objects



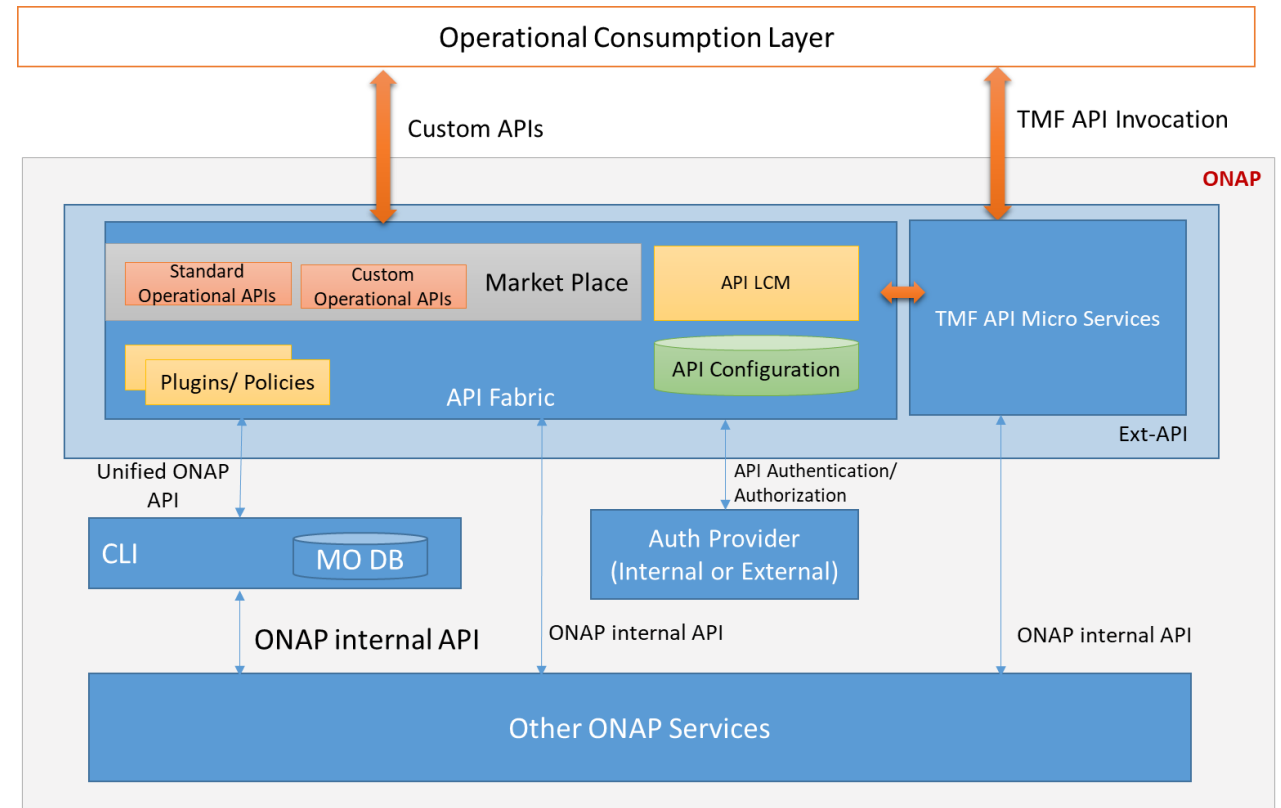
From a consumer point of view it is always beneficial to have a consistent/simple/uniform interface. Deep integration with internal modules are not sustainable

Use Case Overview Operational Lifecycle Enablement

Use Case Overview

- An approach by which operators can consume the operational capabilities of ONAP in a simple or standard based way
- An API Fabric that exposes Operational APIs through Market place for reference and subscription
- API Fabric resides in Ext-API function as an additional MS and integrated with CLI , also supports custom integration with other ONAP internal services to expose a Facade view
- API Fabric also supports establishing secure channel with ONAP internal APIs exposed by ONAP components
- What API Fabric provides
 - API Marketplace of supported APIs that can be subscribed on demand
 - API LCM – Interface to onboard new Operational APIs and integrate with backend
 - Development Toolkit – Ability to insert additional logic while processing APIs

Target : Operations Engineers, Devops Engineers, Operations Team, Product Design Team, Planning



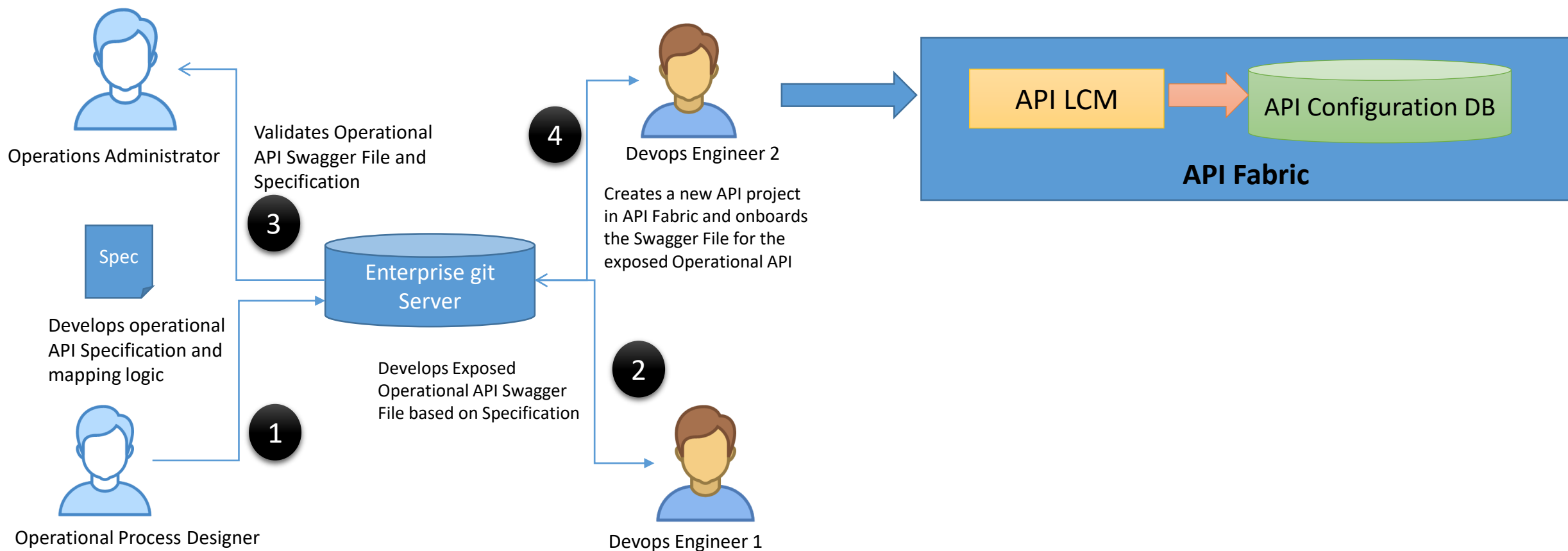
- API Fabric reside as a sub component of Ext-API and provide additional capabilities such as security, API management, marketplace etc.
- CLI project has built in normalized set of managed objects corresponding to the internal APIs in ONAP which can be operated upon via APIs exposed through API Fabric

Use Case Scenarios

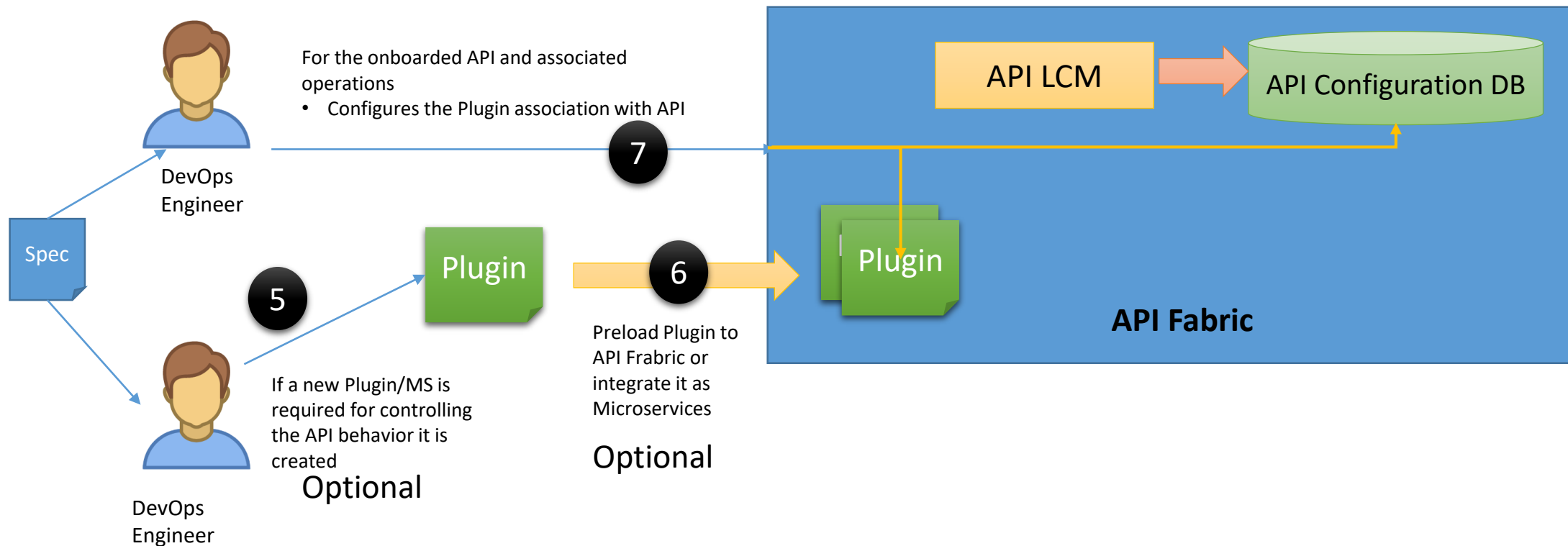
1. Operational API is on boarded for consumption
2. Operational APIs is instrumented for alignment with ONAP API
3. Operational API is secured through an Oauth 2.0 Provider
4. Operational API is activated to be subscribed by external consumers
5. Operational API is subscribed by external consumer
6. Operational API is consumed by external consumer

Use Case Scenario 1 : A new Operational API is on boarded for consumption

Actors : DevOps Engineer , Operations Administrator, Operational Process Designer



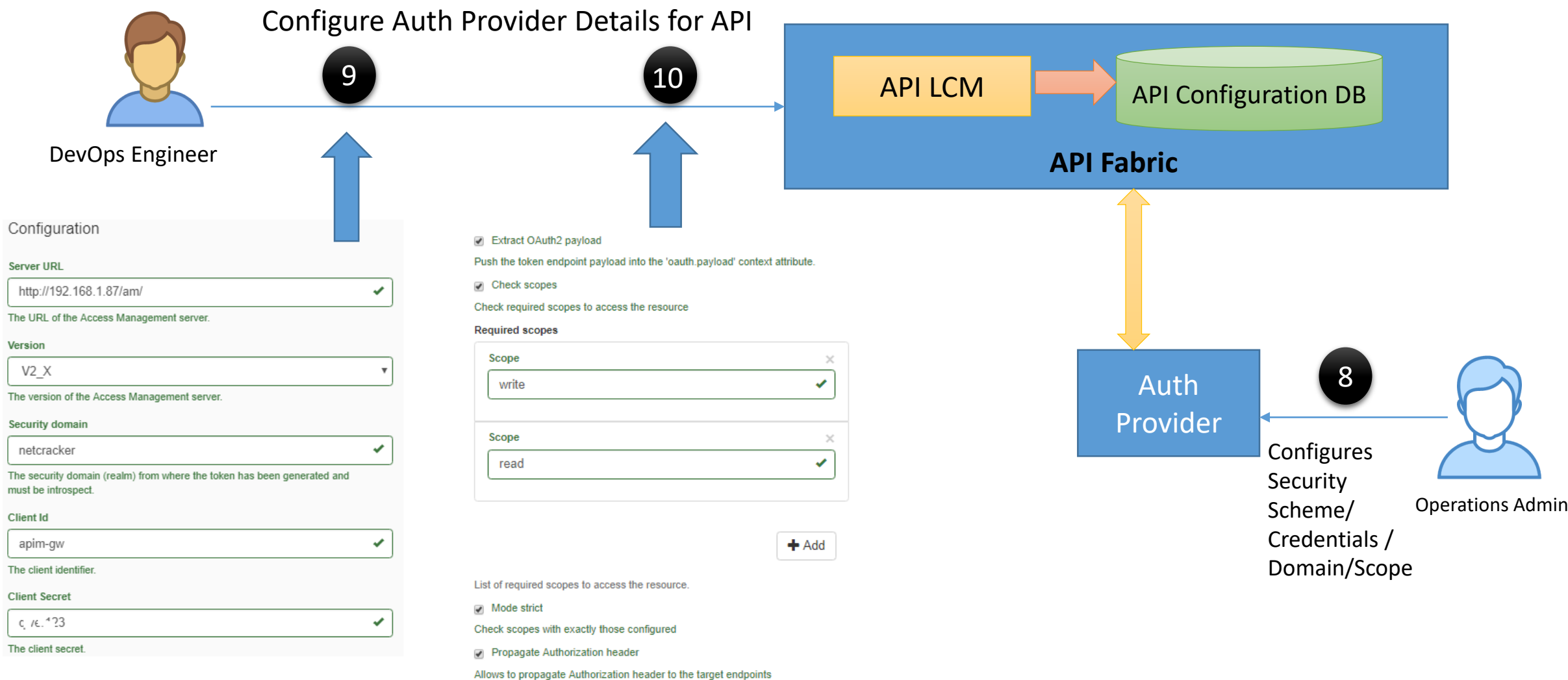
Use Case Scenario 2 : Operational APIs is instrumented for alignment with ONAP API



Plugin Example : API Transformation Logic, Enrichment Logic, Composition Logic

This step will be demonstrated with the built in Plugins in API Fabric

Use Case Scenario 3: Operational API is secured through an OAuth 2.0 Provider

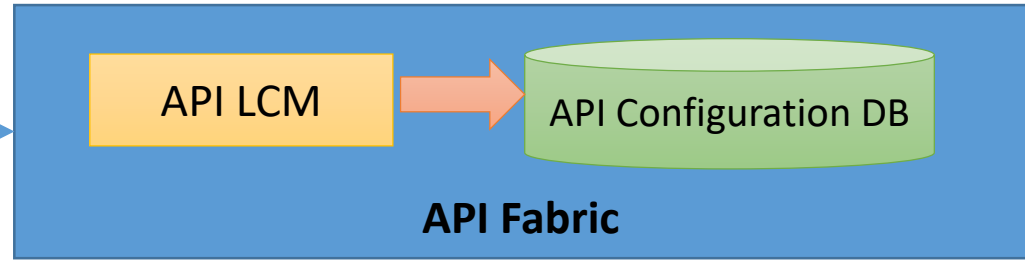


Use Case Scenario 4: Operational API is activated to be subscribed by external consumers



Publishes and Starts API for Subscription/Consumption

11



Published API is available in Marketplace for subscription

Start the API. It will be available on all gateways (depending on sharding tags).

START THE API

Publish the API. It will be available on portal (depending on visibility/rights).

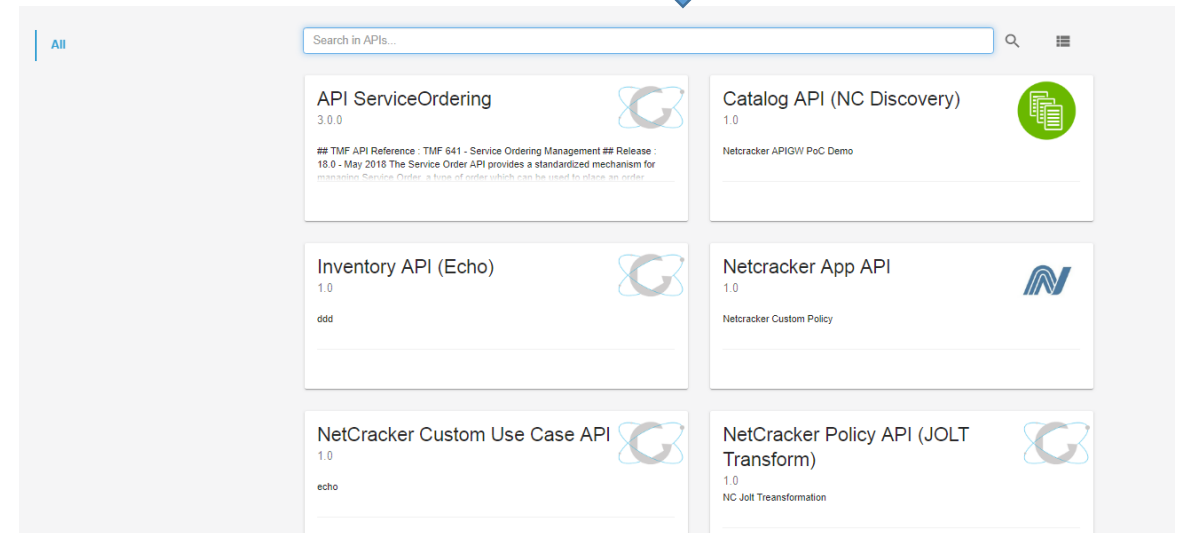
PUBLISH THE API

Make this API public. Everyone can see it.

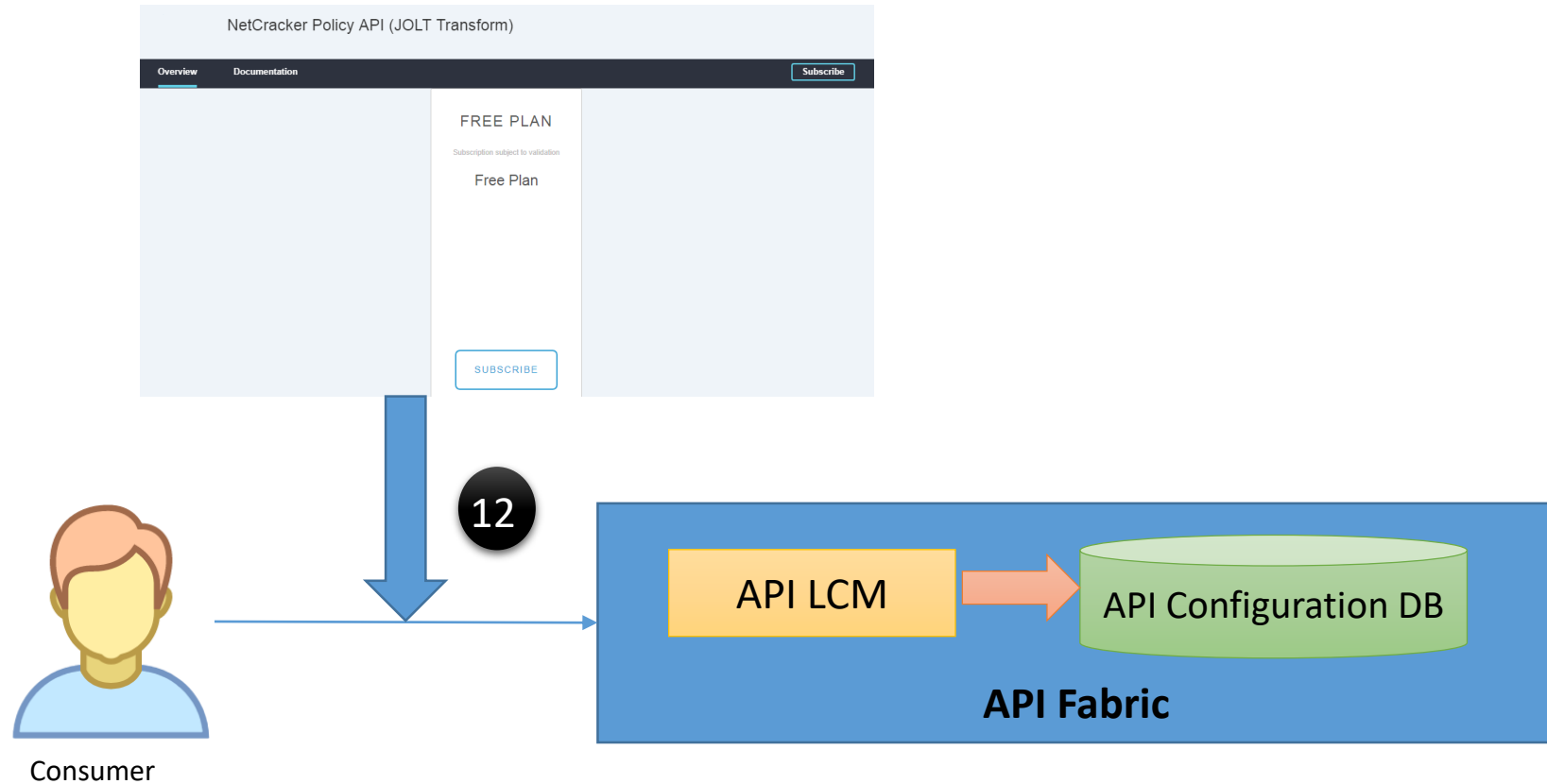
MAKE PUBLIC

Delete this API.

DELETE

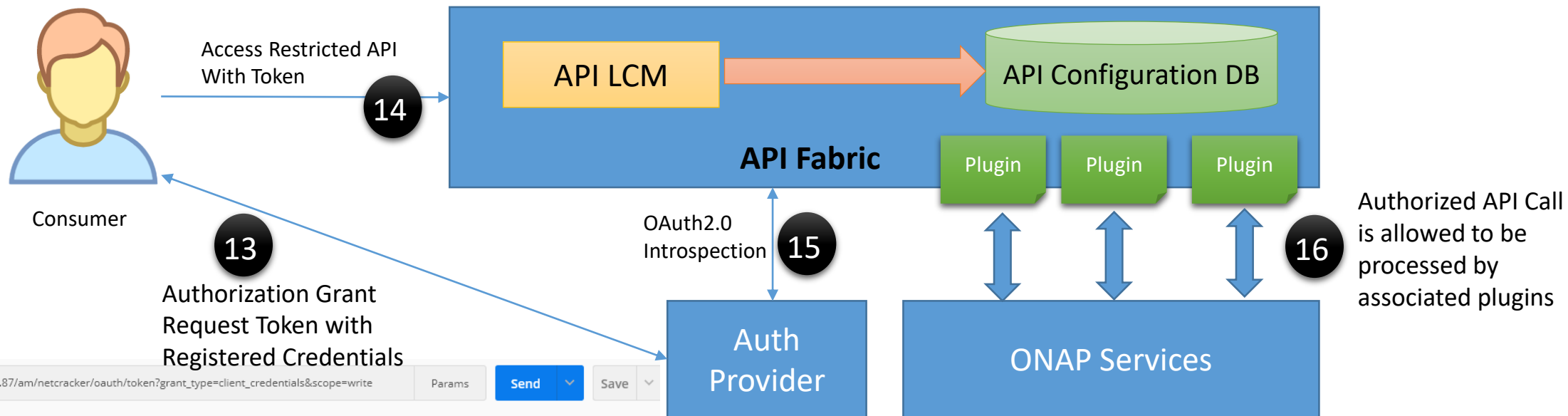


Use Case Scenario 5: Operational API is subscribed by external consumer



Consumer App Subscribes to the API from Market Place or uses the API Fabric Management API to Subscribe

Use Case Scenario 6: Operational API is consumed by external consumer



```

POST http://192.168.1.87/am/netcracker/oauth/token?grant_type=client_credentials&scope=write
Authorization: ...
Body:
{
  "access_token": "eyJraWQ1OjJkZWZhdWx0LWdyYXZpdGV1LUFuLWtleSIsImFsZyI6IkhTMjU2In0...
  "token_type": "bearer",
  "expires_in": 7199,
  "scope": "write"
}
    
```


API Fabric Scope – Wish List

API Façade

- Model Driven API Management – Swagger import, LCM management (version, canary, artifacts/plugin association)
- API Composition toolsets – HTTP Callout and aggregation
- Integration with ONAP specific or external Auth Provider
- API Marketplace , Subscription Management, Plan Management
- API Policy Management –, Tenancy, Rate Limit, Quota, Circuit Break
- Documentation Tools
- Input Validation

API Mediation/Adaptation

- Script insertion – Groovy, Python or Custom
- Business logic insertion – Plugin SDK
- Transformation templates – JOLT , Velocity
- Expression Language – Query strings , Regular expression support
- Alert Generation and Control loop support
- Runtime Mediation Control
- Flexibility to support API variance – SOAP, REST, GraphQL, gRPC, XML, JSON

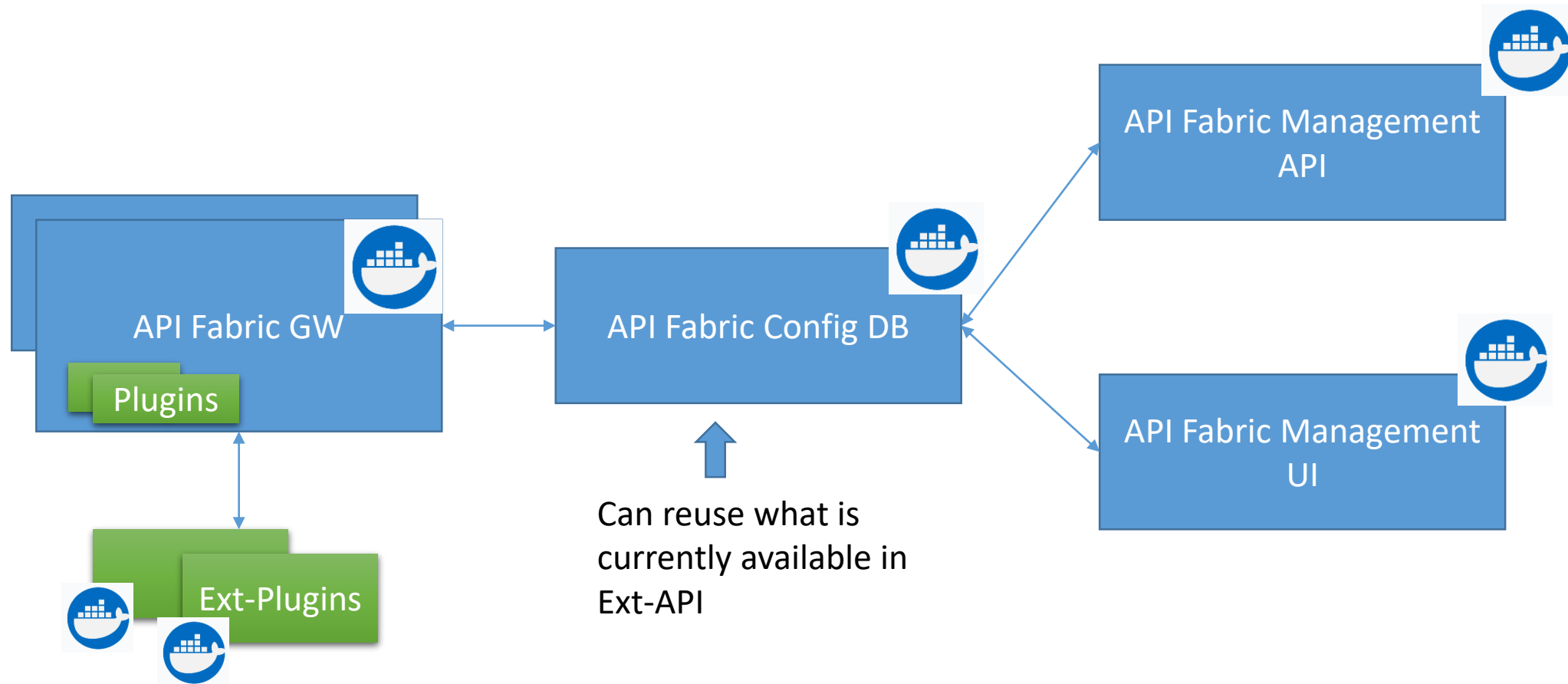
Common (including non-functional)

- API Monitoring, Metrics Collection, Analytics
- Cloud native friendly - Distributed, Microservice based, Scalable
- API Sharding , Canary Support
- Low Maintenance overhead
- Developer friendly toolsets , Low effort

Managed Objects Supported by CLI Project

Managed Objects - CLI Project
Cloud (A&AI)
Customer (A&AI)
Ems (A&AI)
EP (Entitlement Pool) (SDC)
License Group (SDC)
License Mode (SDC)
License-agreement (License Agreement – SDC)
Logic Link (A&AI)
Microservice (MSB)
PNF (manage in A&AI)
Policy-Operational (Policy)
Policy-Config (Policy)
SDNC (A&AI)
Service (SO)
Service-Instance (A&AI)
Service-Model (SDC)
Service Type (SDC)
Subscription (A&AI)
Tenant (A&AI)
VF (SDC)
VF-Model (SDC)
VF Module (SDC)
VIM (A&AI)
VNF (A&AI, APPC)
VNFM (A&AI)
VSP (A&AI)

API Fabric Deployment View



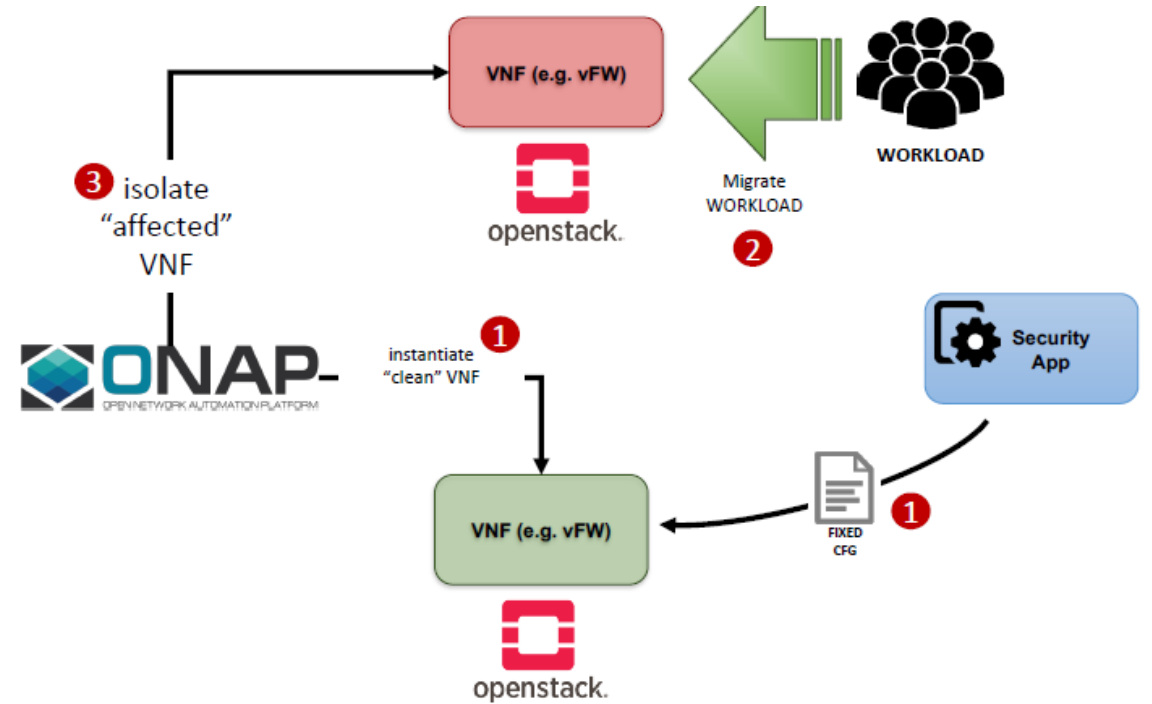
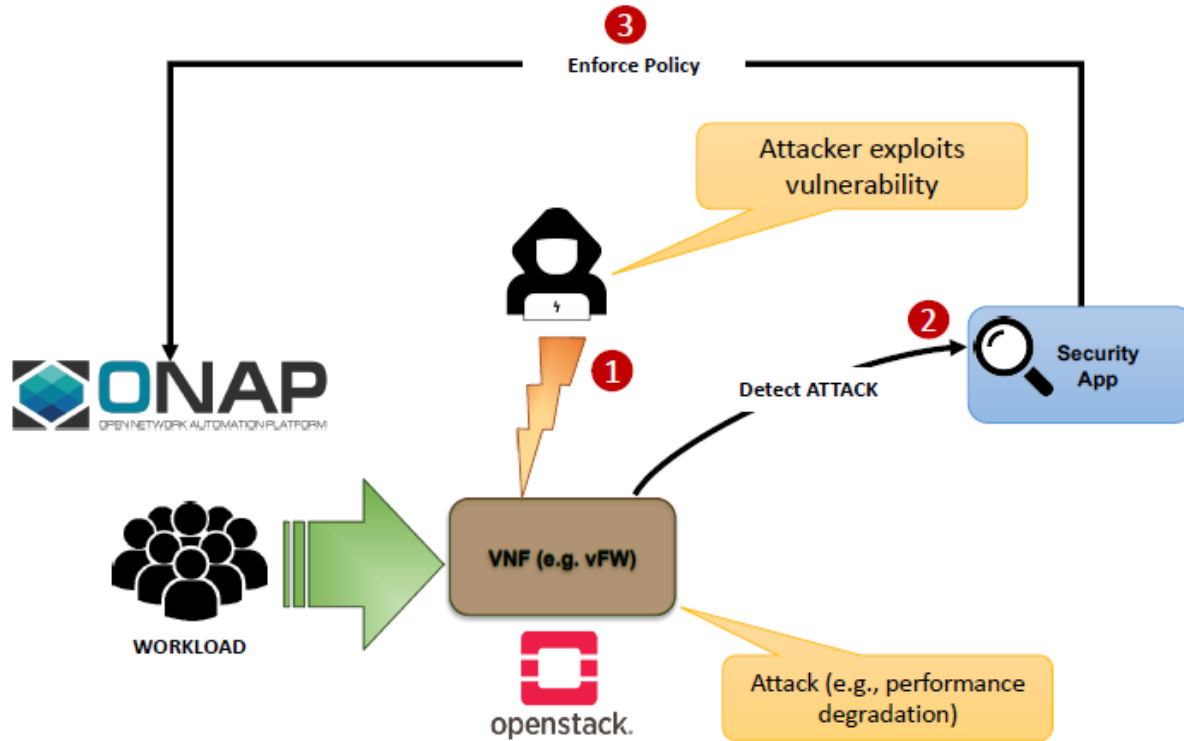


Proof Of Concept Plan

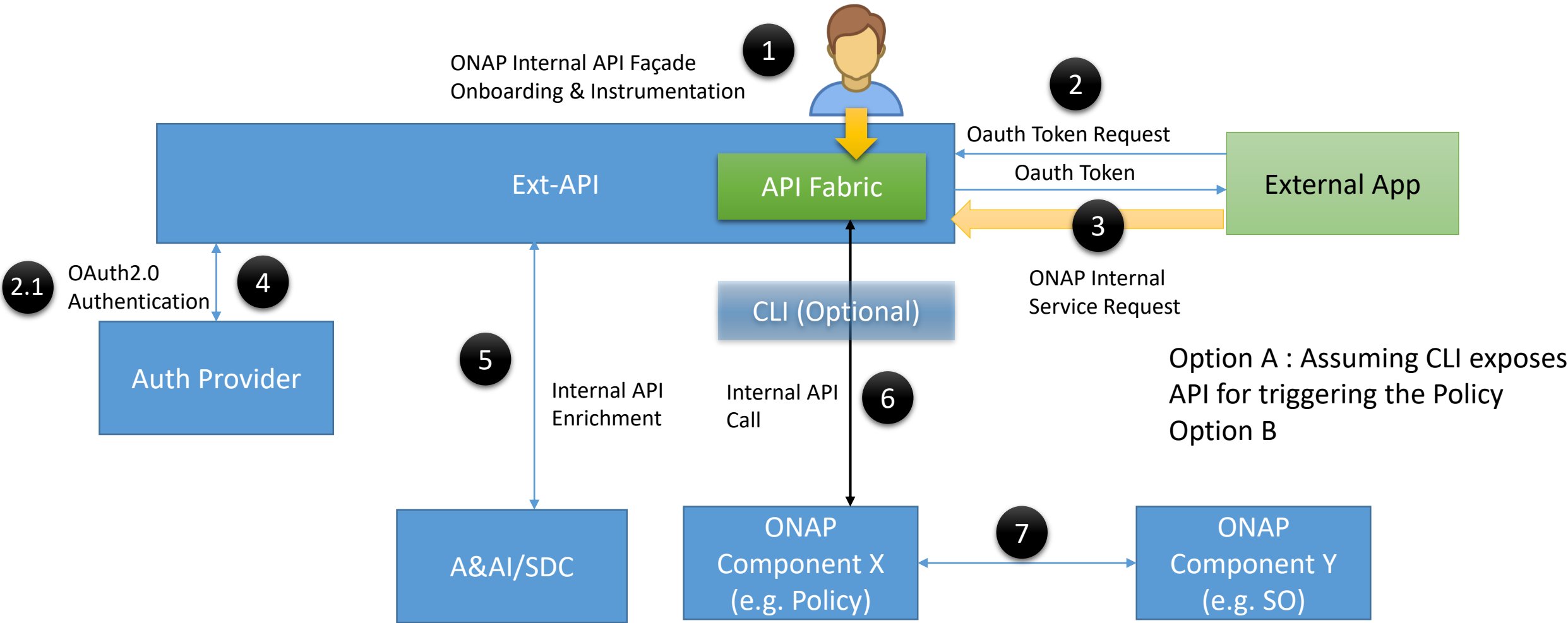
POC Option 1: VF Secure Invocation of ONAP API by External App

- An external Security Monitoring Application monitors VNF for threats (DDOS, Failed Logins)
- ONAP Policy FW is triggered for taking corrective action (isolate the VNF after instantiating a standby VNF)

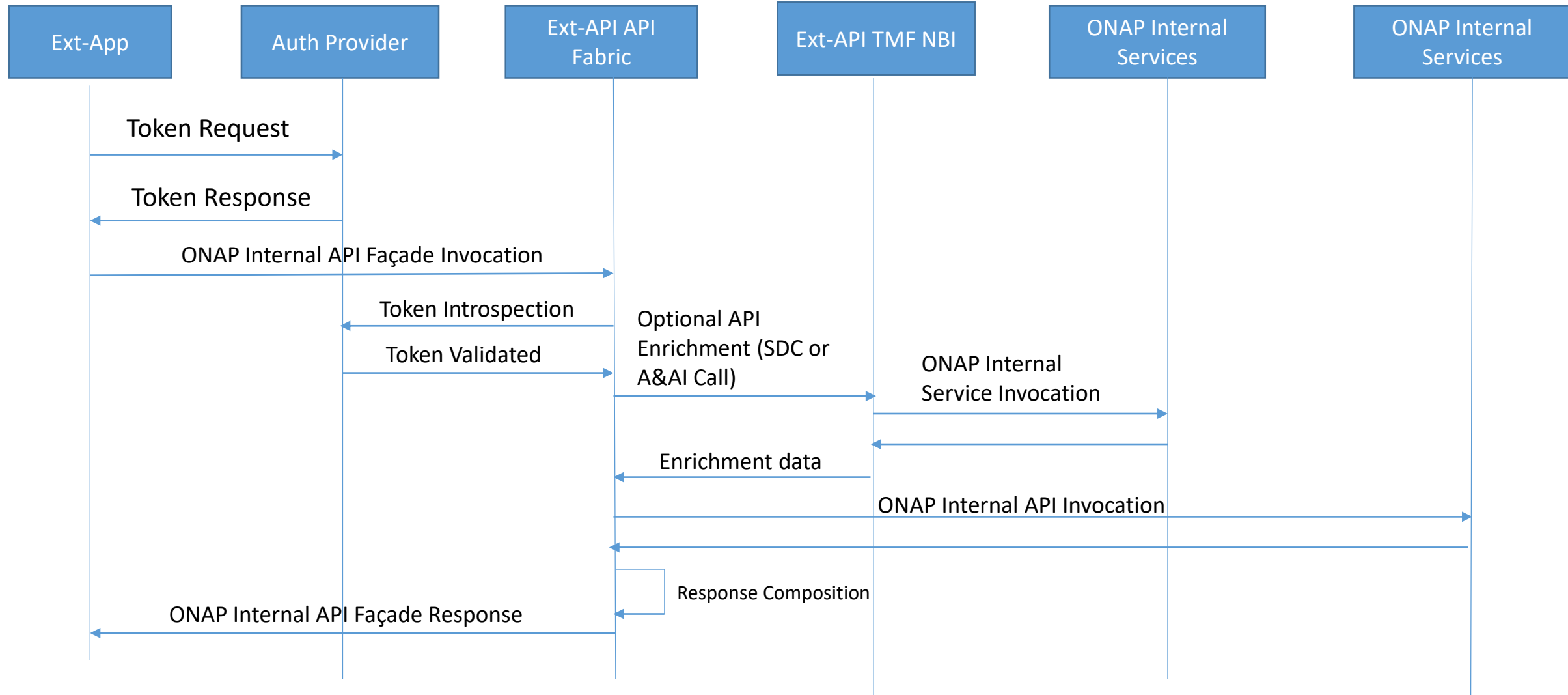
- Currently ONAP does not expose Policy APIs directly through a secure channel.
- VF Requirement is to expose ONAP internal APIs through a Façade interface with security – API GW is the right choice as per the feedback they received from community



POC Option 1: Realization



POC Option 1: Sequence of Operation

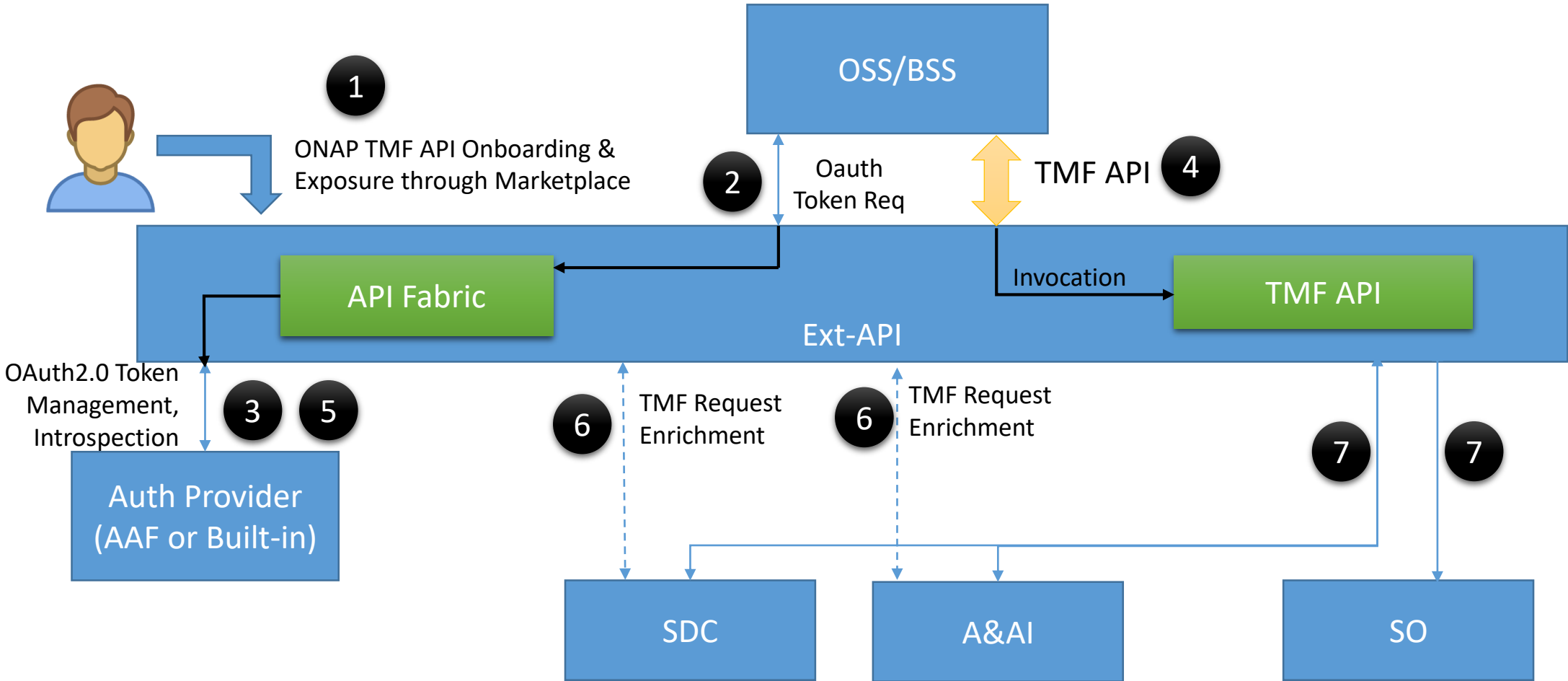


POC Option 2: Secure Invocation of External API

- **Co-locate API Fabric with External API Project and enable Secure interface for all Ext-API invocations**
- **For any Custom/Façade API invocations from outside ONAP, support them through the approach followed by POC Option 1**

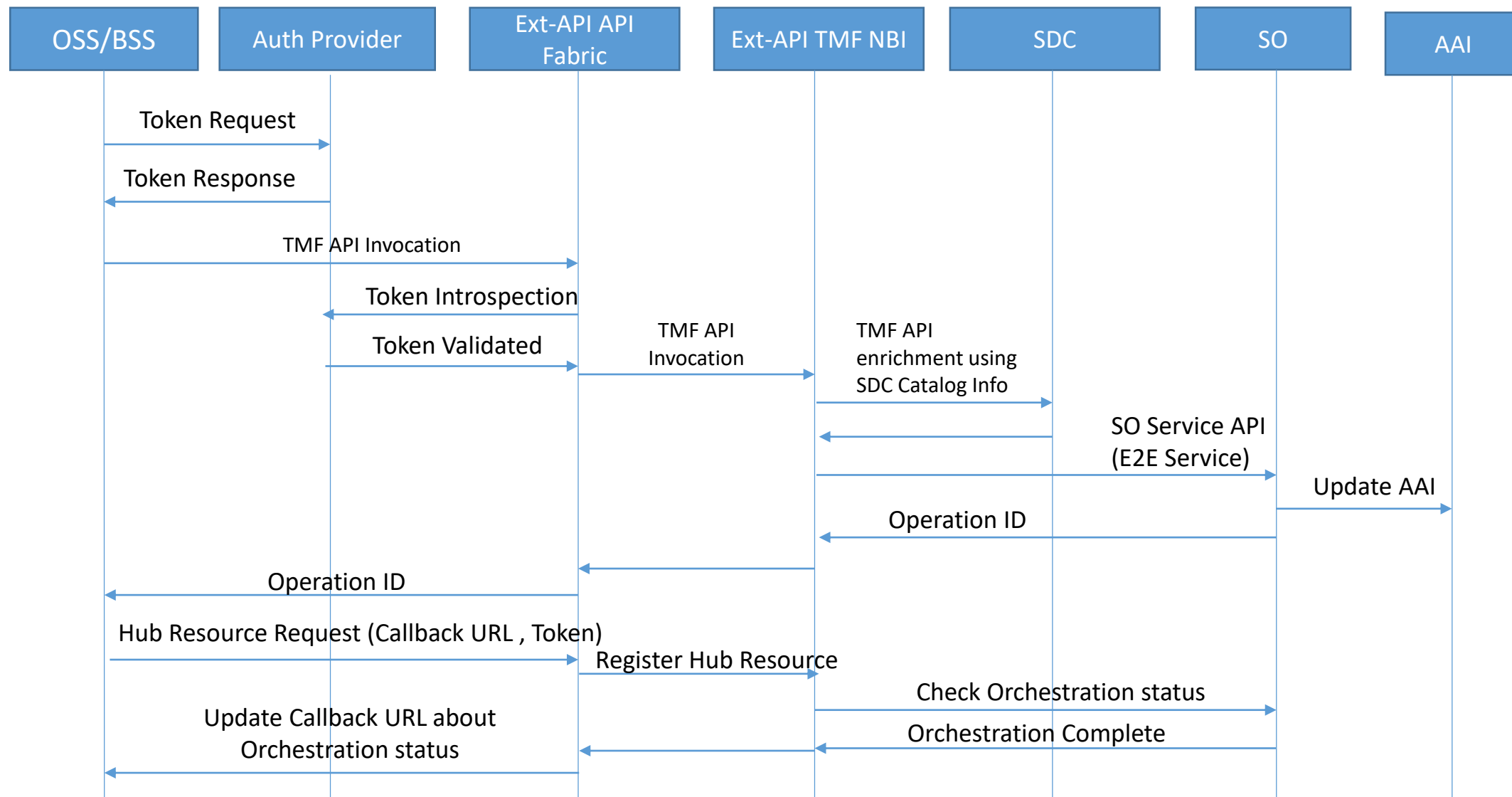
- API Fabric to expose the supported TMF APIs in the Marketplace
- Allow subscription of supported TMF APIs
- Support API Management and Security Management at API Fabric level
- Integrate API Fabric with ONAP AAF or an In-Built Auth Provider to enable Oauth 2.0 based authentication

POC Option 2: Secure Access of ONAP External API

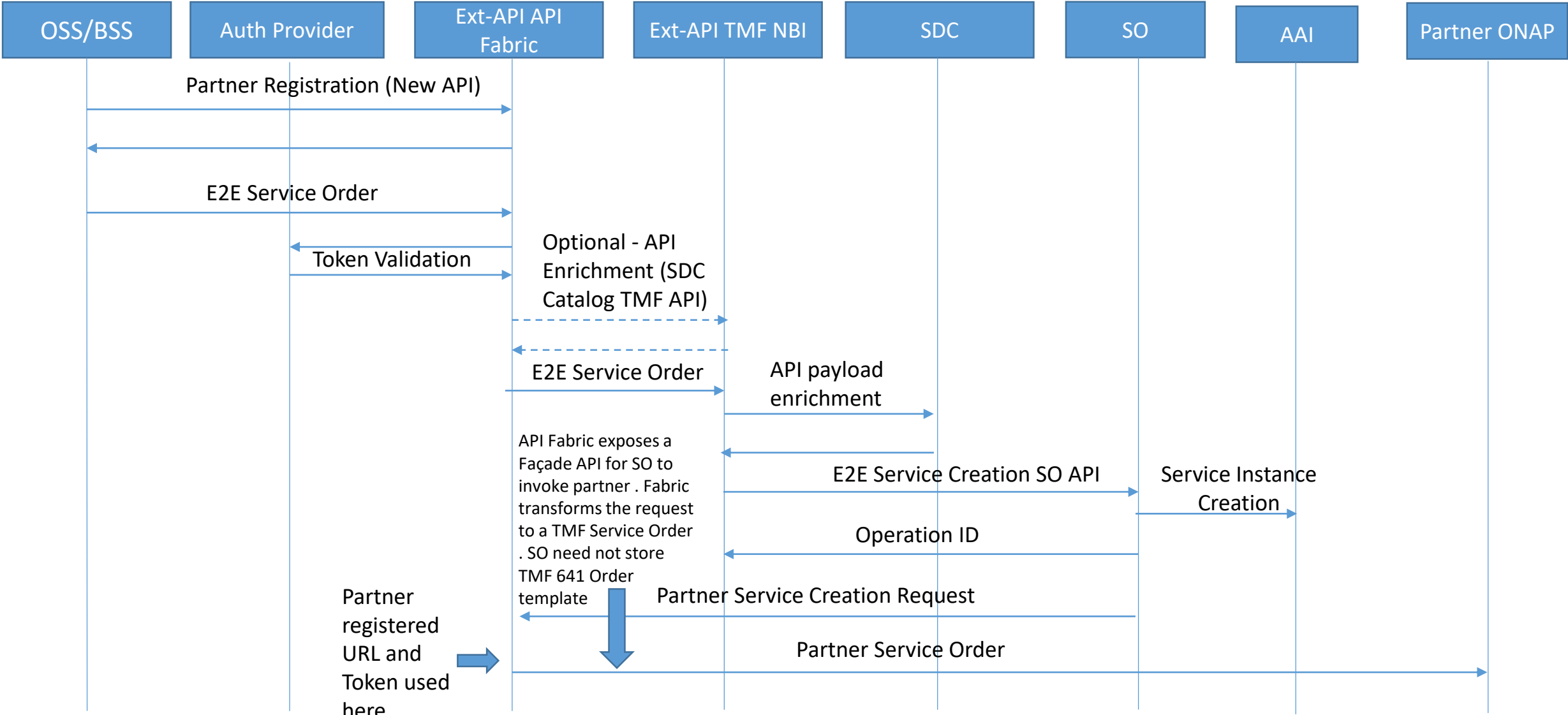


PoC 2: Sequence Diagram 1/2

Subscription of API through Marketplace is not shown



PoC 2: Sequence Diagram 2/2



Potential Security Mechanism

- MEF initiated a new work item “Security Mechanisms for Inter-carrier Interfaces” in May 2019. Currently in exploratory stage
- ETSI GS NFV-SEC 022 V0.1.0 (2019-04) suggests OAuth 2.0 Access Token based Security for APIs
- We can start supporting OAuth2.0 based Security Mechanism
- API Fabric can maintain keystore and truststore to expose https based interaction externally

Potential Challenges

- Hub Resource Security – Security Mechanism for Call Back
- Use of AAF as Auth Provider : Oauth 2.0 Protocol Interface availability to be checked
 - Need for Special Auth Client to interact with AAF



ONAP

OPEN NETWORK AUTOMATION PLATFORM

Q&A