

ONAP Orchestrated CCVPN Usecase

OSN 2018

Chaker Al-Hakim

Futurewei

Two thin, light blue wavy lines that span the width of the slide, positioned near the bottom. They overlap and create a subtle, flowing pattern.

ONAP-Powered Connectivity Service

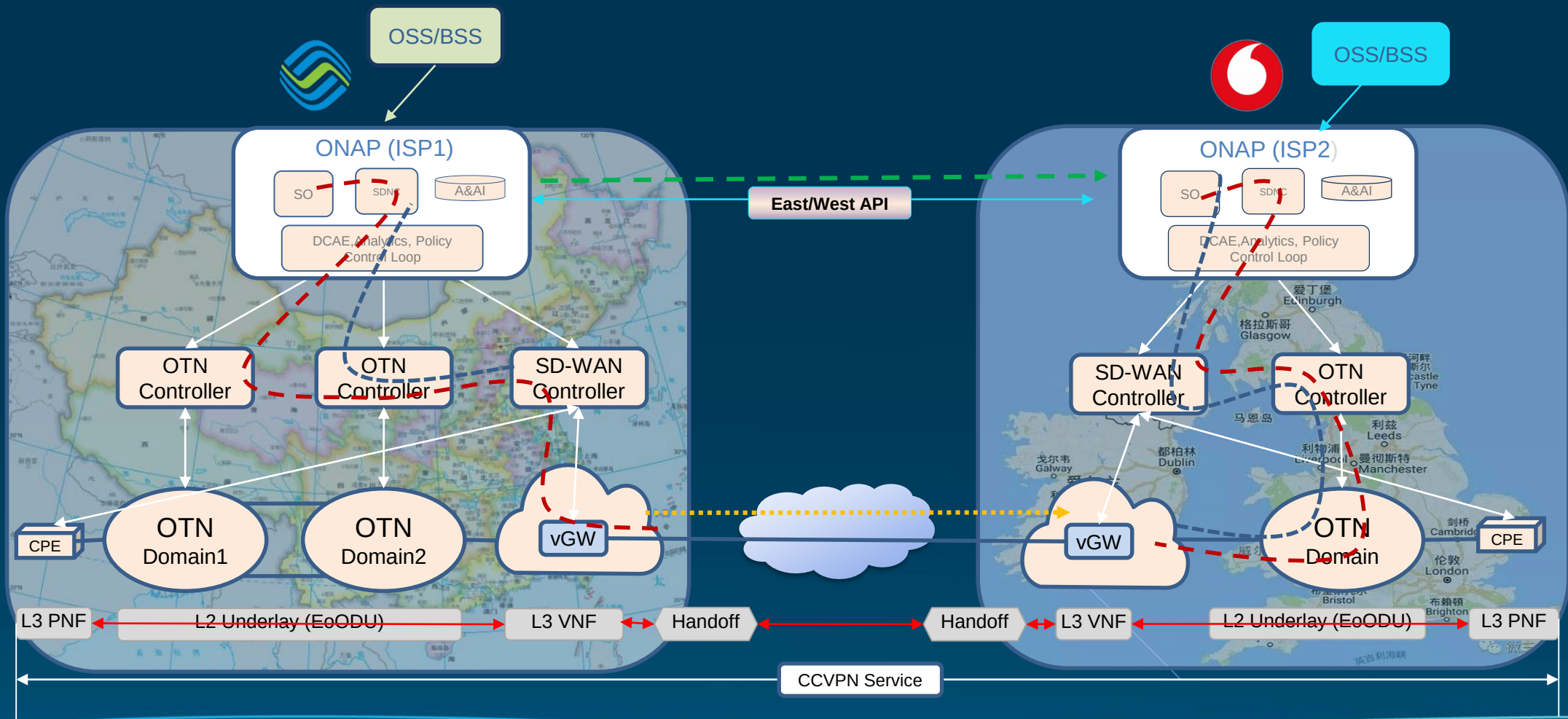
Supporting the CCVPN Usecase using the ONAP Casablanca Release

What is the CCVPN Usecase?

- **CCVPN:**
 - Cross Operator, Cross Domain, Cross Layer VPN Service
- **Cross Operator, Cross Domain, Cross Layer**
 - 2 Different Operators
 - ONAP is Deployed at 2 Different Geo Sites,
 - Different Networking Domains (Different Underlying Networking infrastructure)
- **VPN Service**
 - Modeled after the E-Line Service

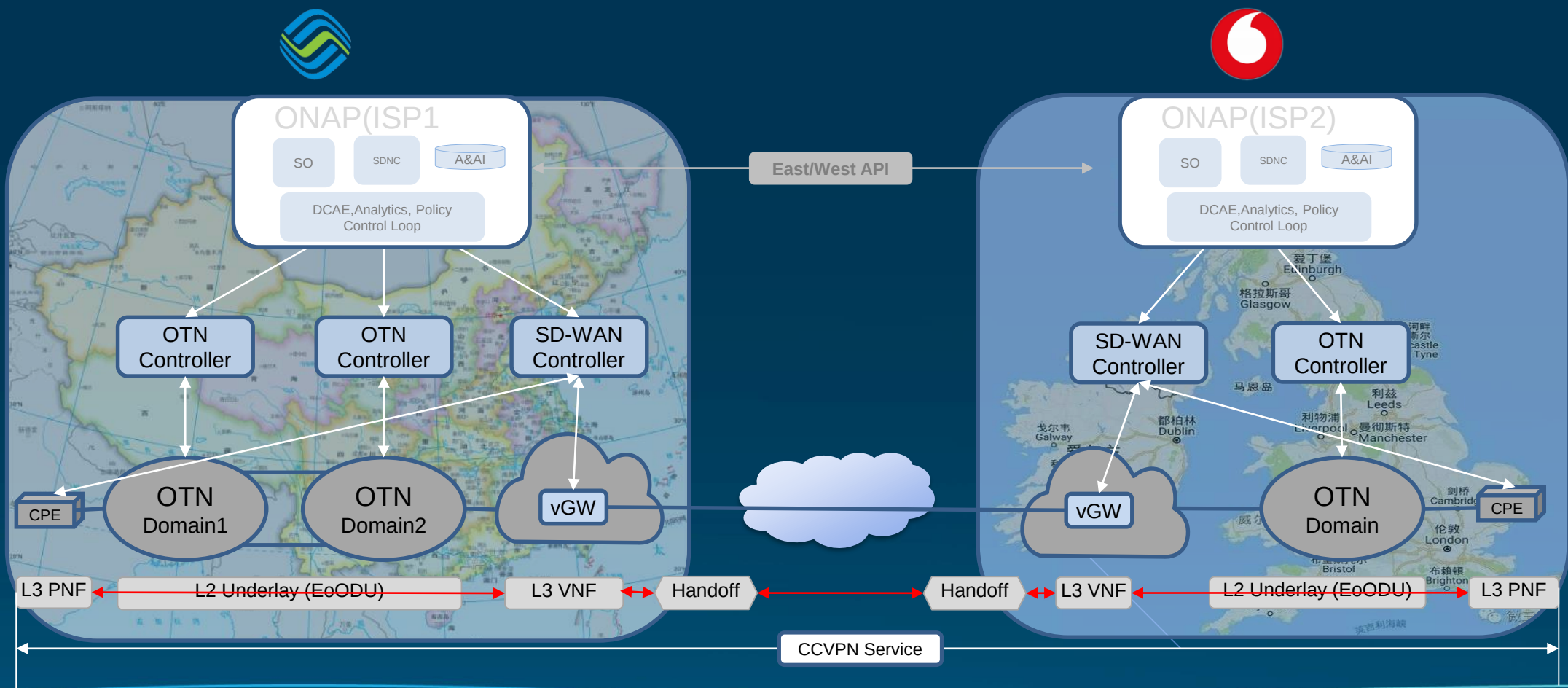
ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service – End State



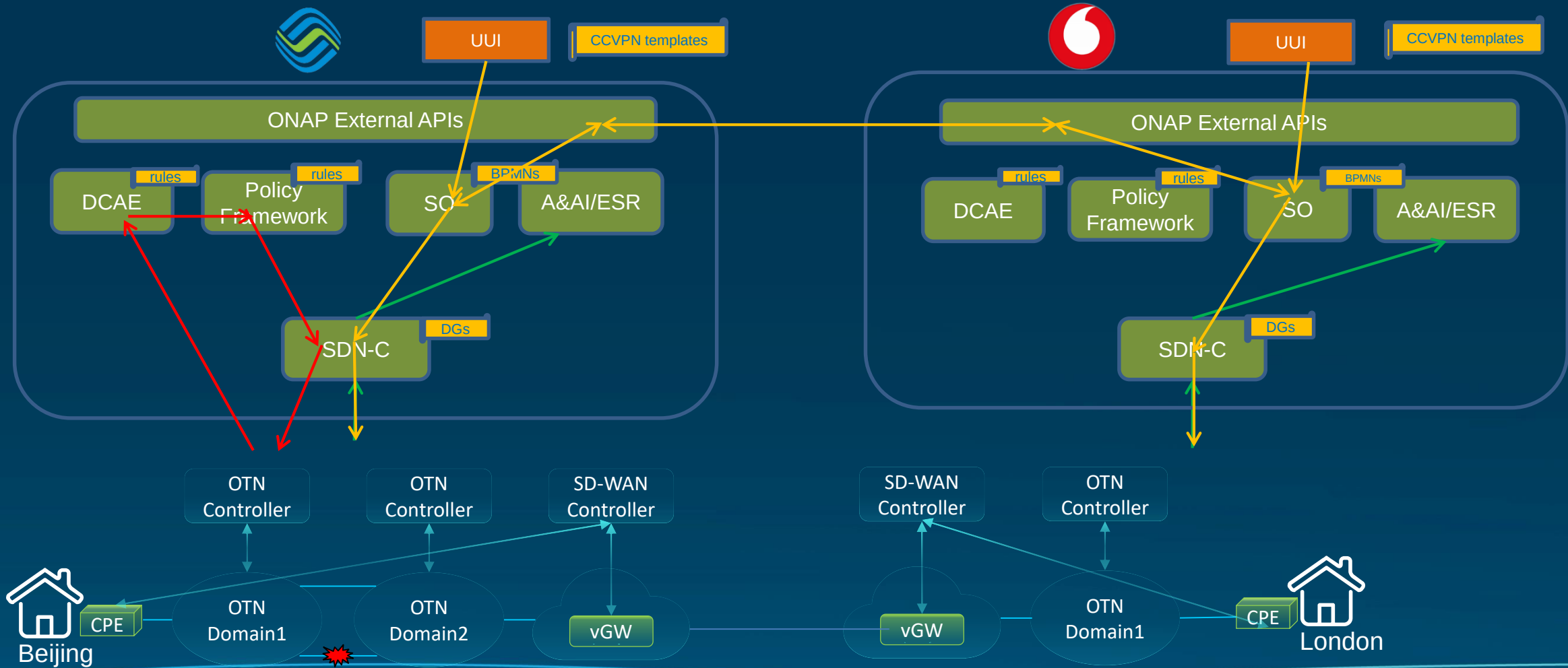
ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service



How ONAP is used to Orchestrated CCVPN

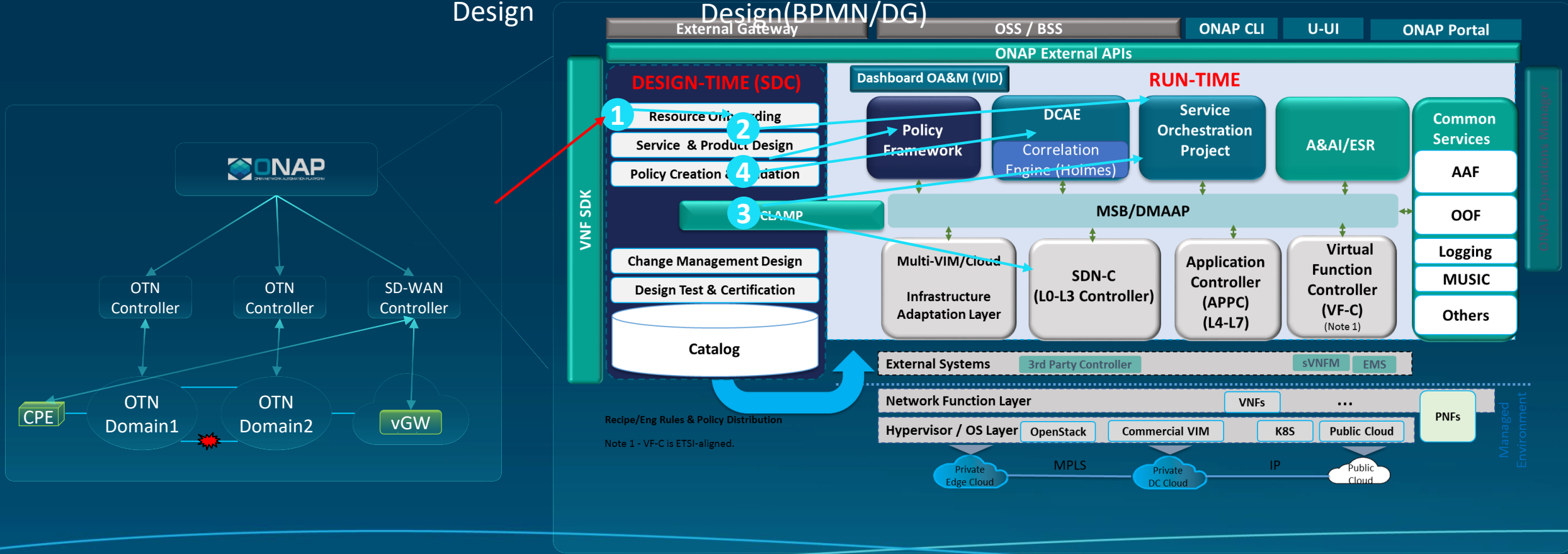
- Topology Discovery
- Service Provisioning
- Closed Loop



How the CCVPN Service is Orchestrated in ONAP

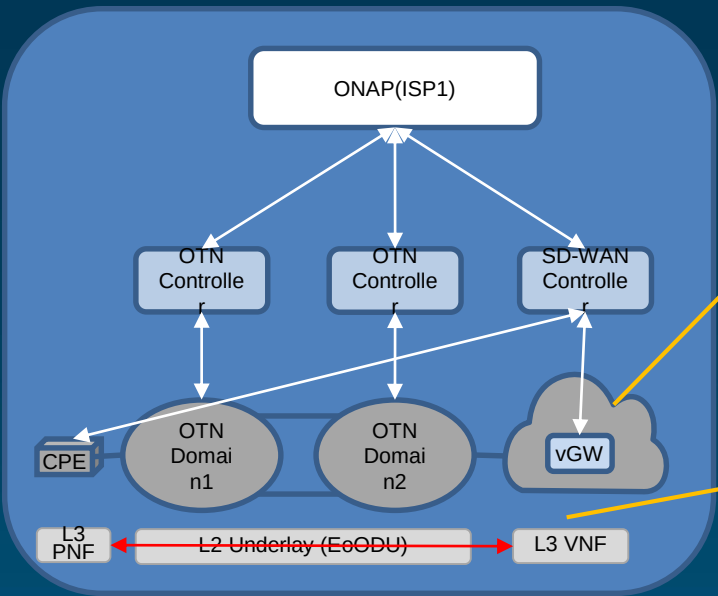
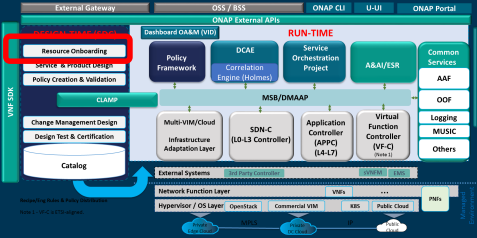
4 steps to develop CCVPN on ONAP:

- 1 Model-based VF Onboarding
- 2 Service Design
- 3 Control Logic Design(BPMN/DG)
- 4 Analysis/Policy Rule Design



Step 1: Model-based Resource Onboarding

- 1. CCVPN analysis and decompose scenario to resource definitions
- 2. Onboard VNF/Configuration as VF resource
- 3. Test VF

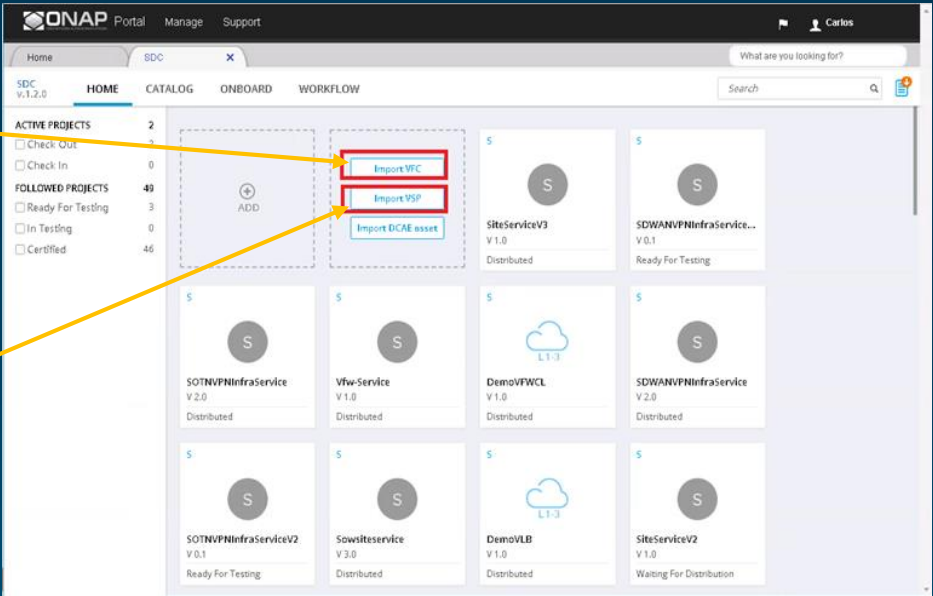


VNFs

ar1000V.csar

Configurations

- Site.yml
- Device.yml
- Sitewan.yml
- Sitelan.yml
- Sppartner.yml
- Sotn connectivity.yml
- SDWAN connectivity.yml



Step 2: Service Design

- 1. Design service, Add Specific Service Artifacts
- 2. Test service Flow
- 3. Onboard Service

ONAP Portal

Manage

Support

Home

SDC

What are you looking for?

SDC

V1.2.0

HOME

SERVICE: SiteServiceV3

Composition

Elements63

Search...

NETWORK CONNECTIVITY

GENERIC

CONFIGURATION

Configuration

deviceV2VF

deviceVF

deviceVFV3

Port Mirroring C...

Port Mirroring C...

siteV2VF

SiteVF

V1.0

DISTRIBUTED

Service Paths: Hide all

Check Out

Diagram showing service flow: Internet -> sdw/vpn... -> mpls -> siteV2VF 0 -> deviceV2... -> SPPartne... and sotr/vpnatt... -> mpls

SiteServiceV3

GENERAL INFO

Type: SERVICE

Version: 1.0

Category: E2E Service

Creation Date: 08/27/2018

Author: Joni Mitchell

Project Code: 33334444

Service Type:

Service Role:

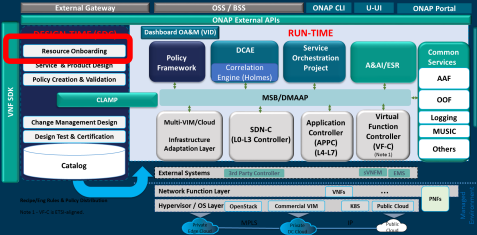
Contact ID: cs0008

Description: SiteServiceV3

ADDITIONAL INFORMATION

TAGS

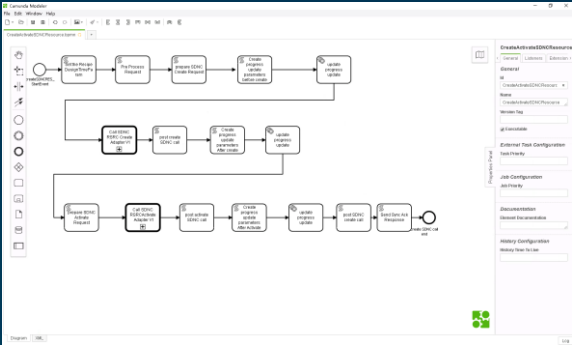
SiteServiceV3



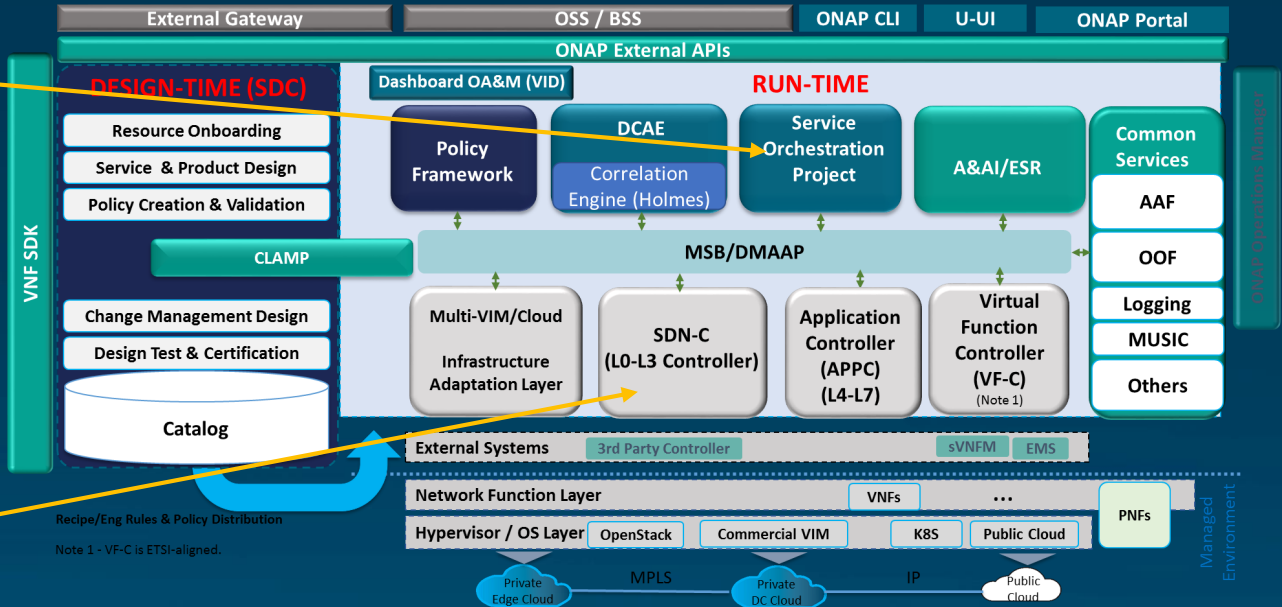
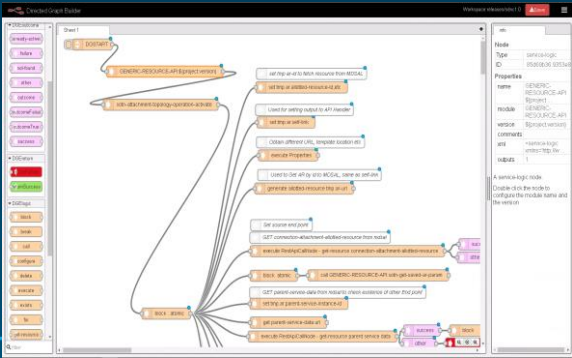
Step 3: Service/Resource Flows Design (BPMN/DG)

- 1. Service/Resource BPMN workflow design (Vendor Agnostic)
- 2. Resource instance DG flow design (Vendor Specific)
- 3. Upload BPMN to Service Orchestrator (SO)
- 4. Upload Directed Graph (DG) to SDN-C

Service/Resource workflow design(BPMN)



Resource flow design(DG)



Recipe/Eng Rules & Policy Distribution
Note 1 - VF-C is ETSI-aligned.

Step4: Analysis/Policy Rule Design

- 1. Define data analysis rule
- 2. Define policy rule
- 3. Upload data analysis rule to DCAE/Holmes
- 4. Upload policy rule to Policy

Data analysis rule design

- Step 1. Collect alarms
- Step2: Analyze the alarms
- Step3: trigger a cross-link down event

```
rule "Set Up Correlation"
no-loop true
salience 200
when
  fa: VesAlarm(sid: eventId,
  start: startTimestamp,
  eventName.indexOf("Fault_Route_Status") != -1)
  sb: VesAlarm(eventId != Sid,
  eventName.indexOf("Fault_Route_Status") != -1,
  watch.when(startTimestamp - Start) < 60000)
then
  String status = "Down";
  if (status.equalsIgnoreCase(getAdditionalField(fa, "opac-status"))) {
    // If any of the alarms have been marked as root, a policy message has ever been created and sent. Do NOT send it again.
    if (fa.getRootFlag() != 1 || sb.getRootFlag() != 1) {
      PolicyMsg msg = createPolicyMsg(fa);
      DmaapService dmaapService = ServiceLocatorHolder.getService(DmaapService.class);
      dmaapService.publishPolicyMsg(msg, "unauthenticated.DCAE_CS_OUTPUP");
      updateAdditionalField(fa, "status", status);
    }
  }
  fa.setRootFlag(1);
  sb.setRootFlag(1);
  update(fa);
  update(sb);
end

rule "Clear Alarms"
no-loop true
salience 100
when
  fa: VesAlarm(eventName.indexOf("Fault_Route_Status") != -1)
then
  if ("up".equalsIgnoreCase(getAdditionalField(fa, "opac-status"))) {

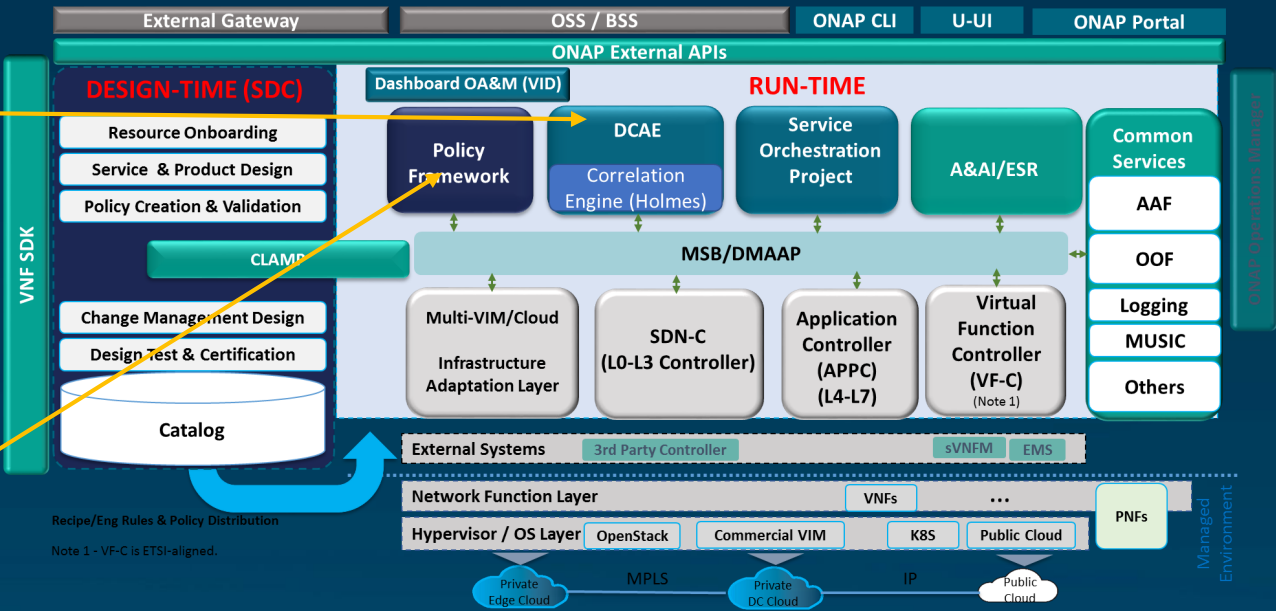
```

Policy rule design

- 1.Subscribe with the cross-link down event.
- 2.Take action to reroute

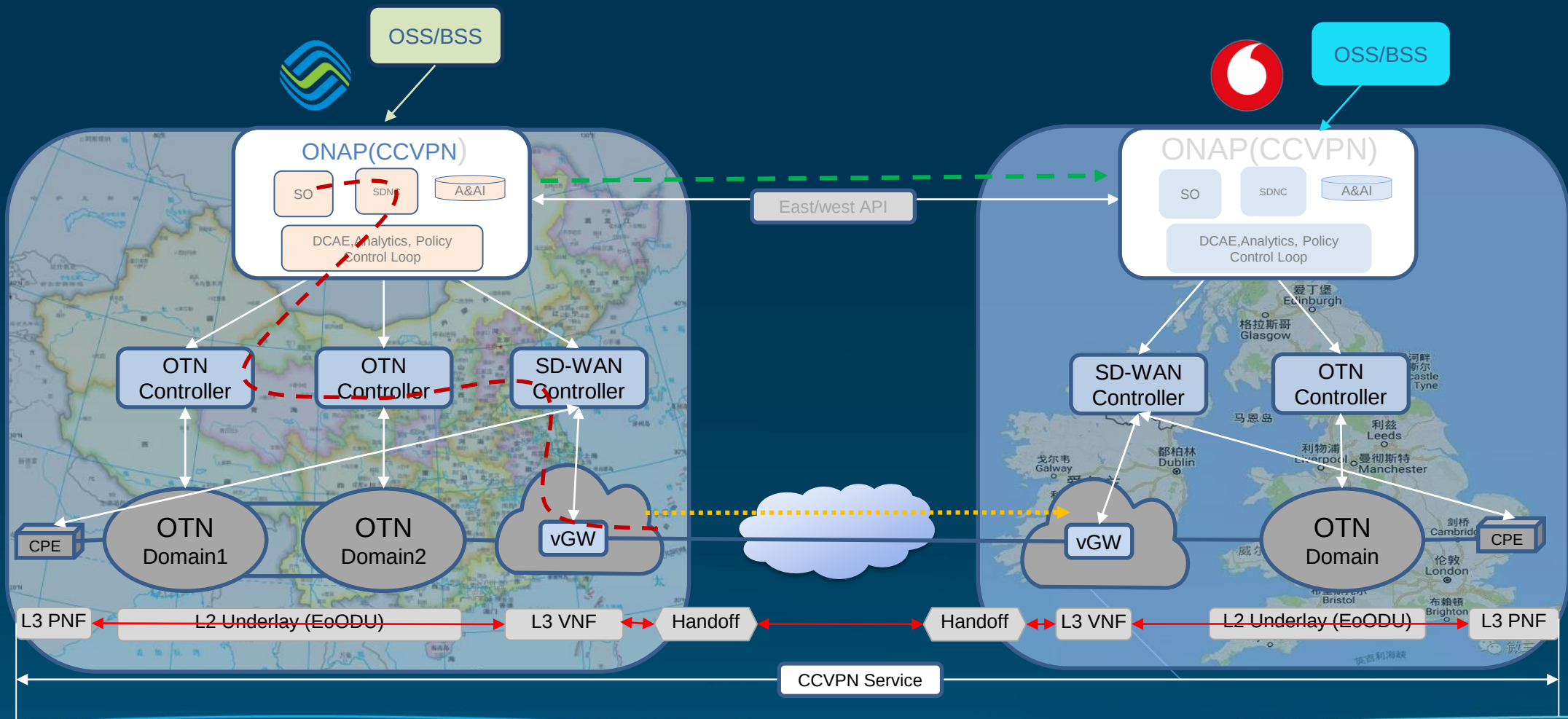
```
controlLoop:
version: 2.0.0
controlLoopName: ControlLoop-CCVFN-2179b738-fd36-4843-a71a-a8c24c70c55b
trigger_policy: unique-policy-id-16-Reroute
timeout: 3600
abatement: false

policies:
- id: unique-policy-id-16-Reroute
  name: Connectivity Reroute
  description:
  actor: SDNC
  recipe: Reroute
  target:
    type: VM
  retry: 3
  timeout: 1200
  success: final_success
  failure: final_failure
  failure_timeout: final_failure_timeout
  failure_retries: final_failure_retries
  failure_exception: final_failure_exception
  failure_guard: final_failure_guard
```



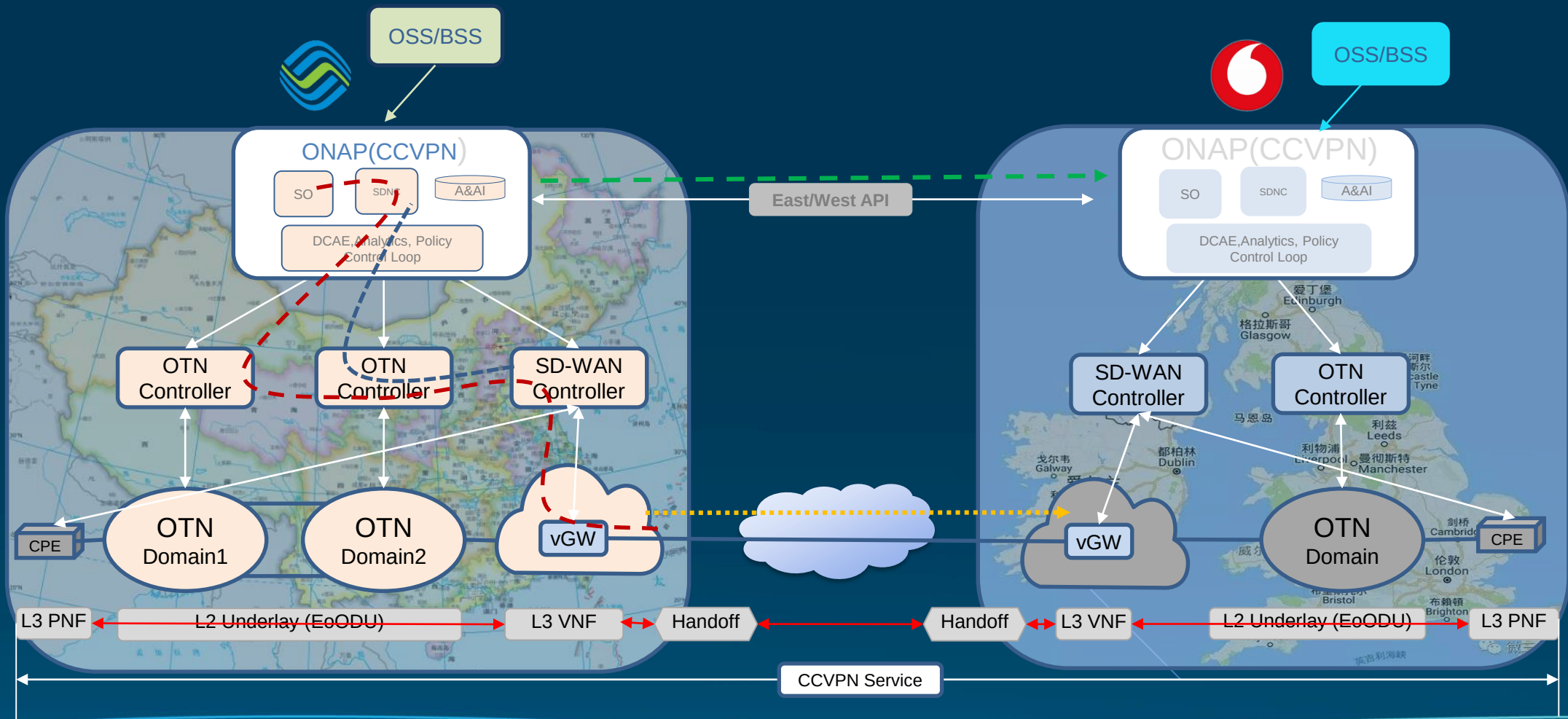
ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service



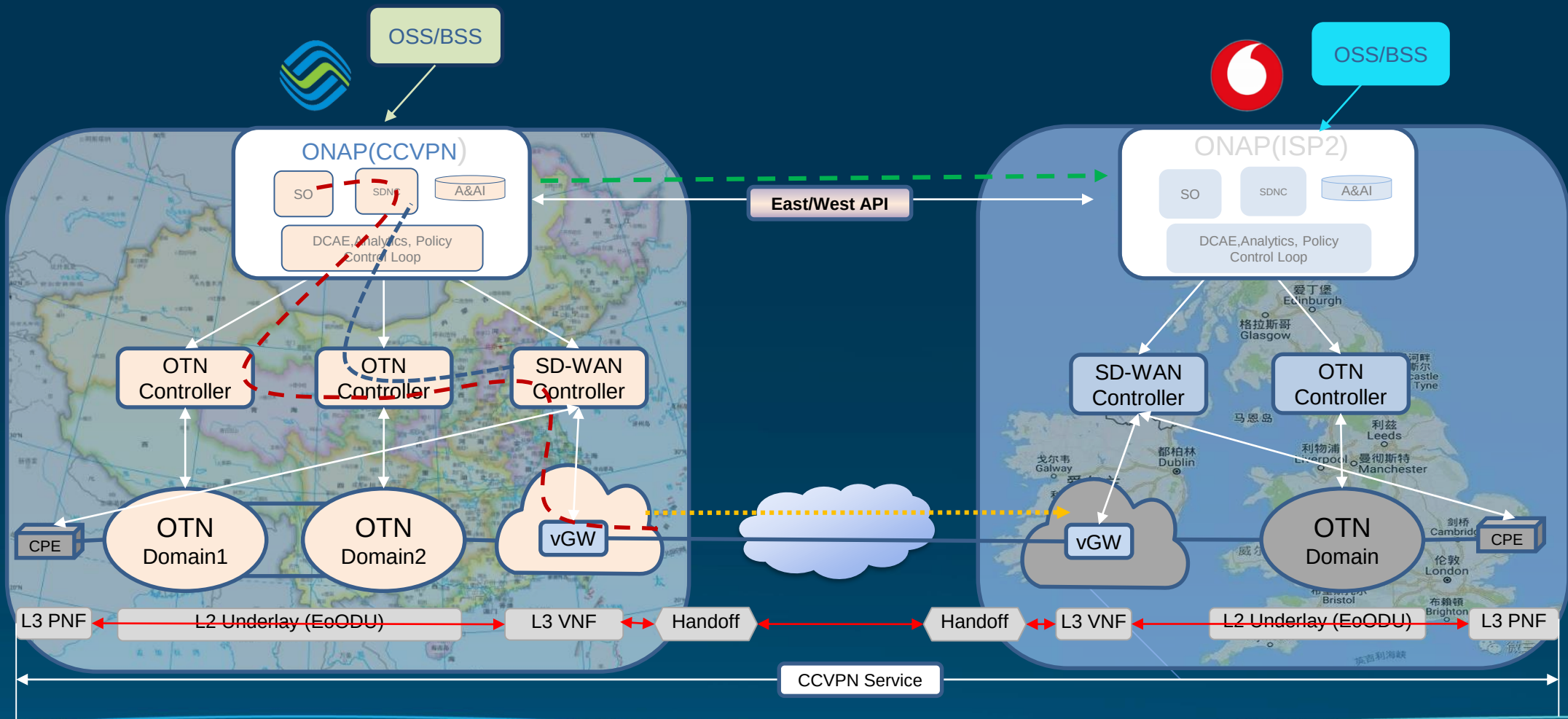
ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service



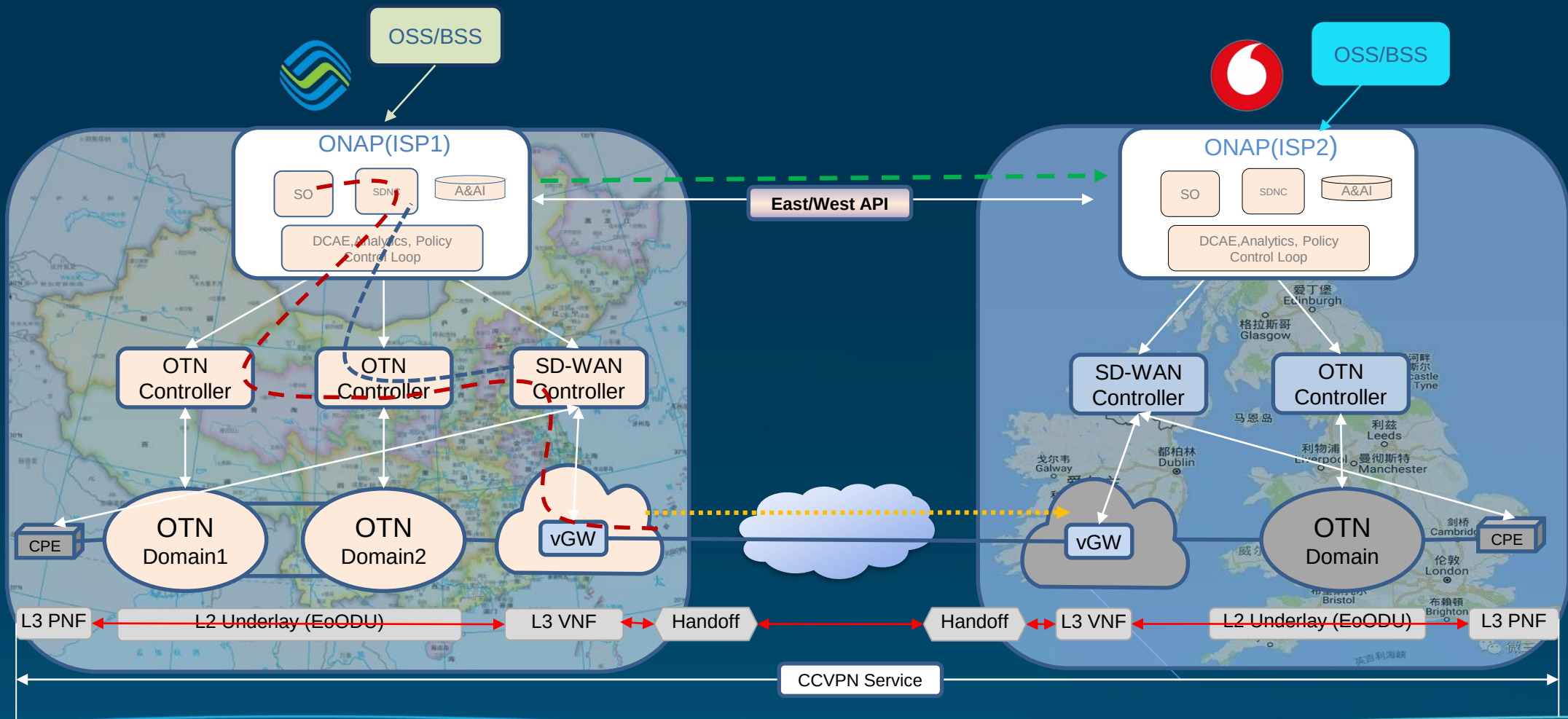
ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service



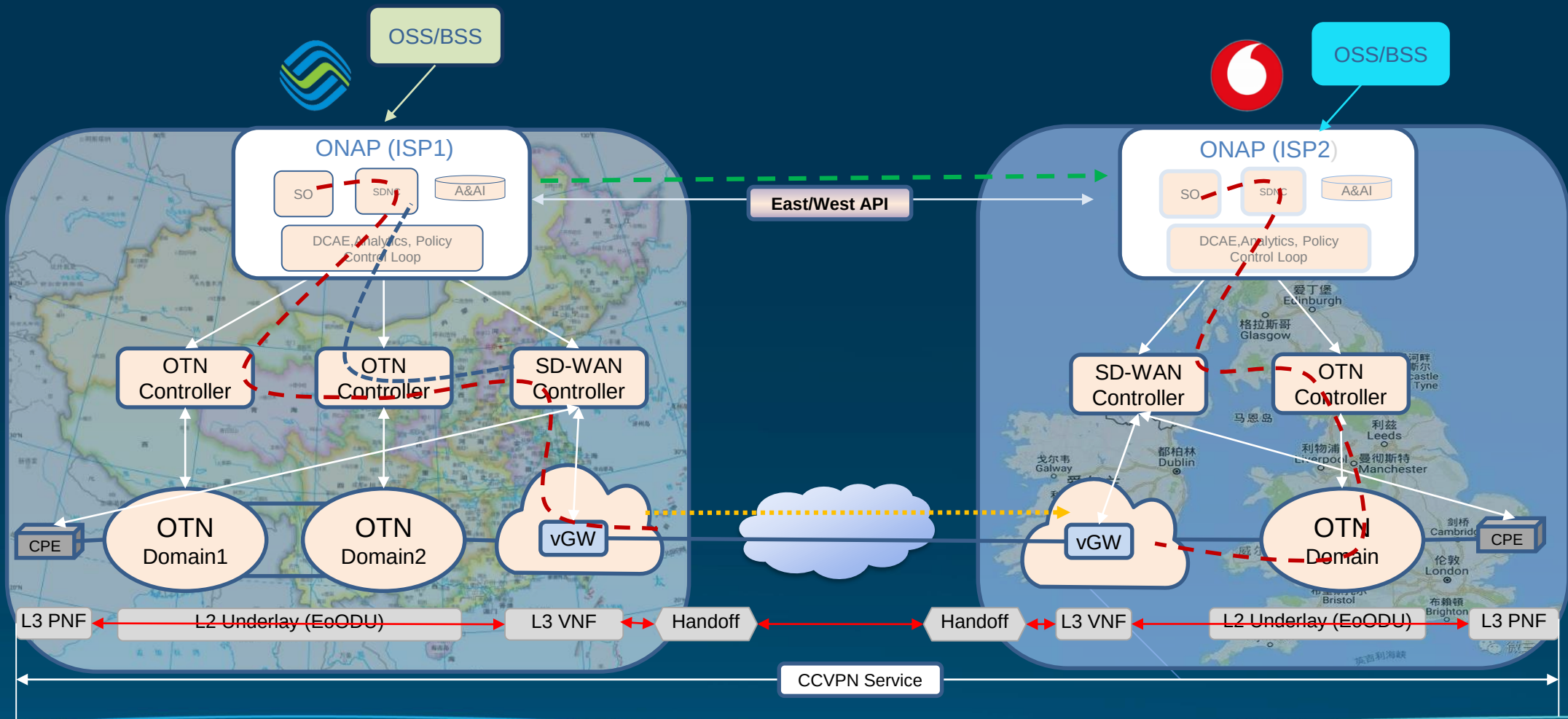
ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service



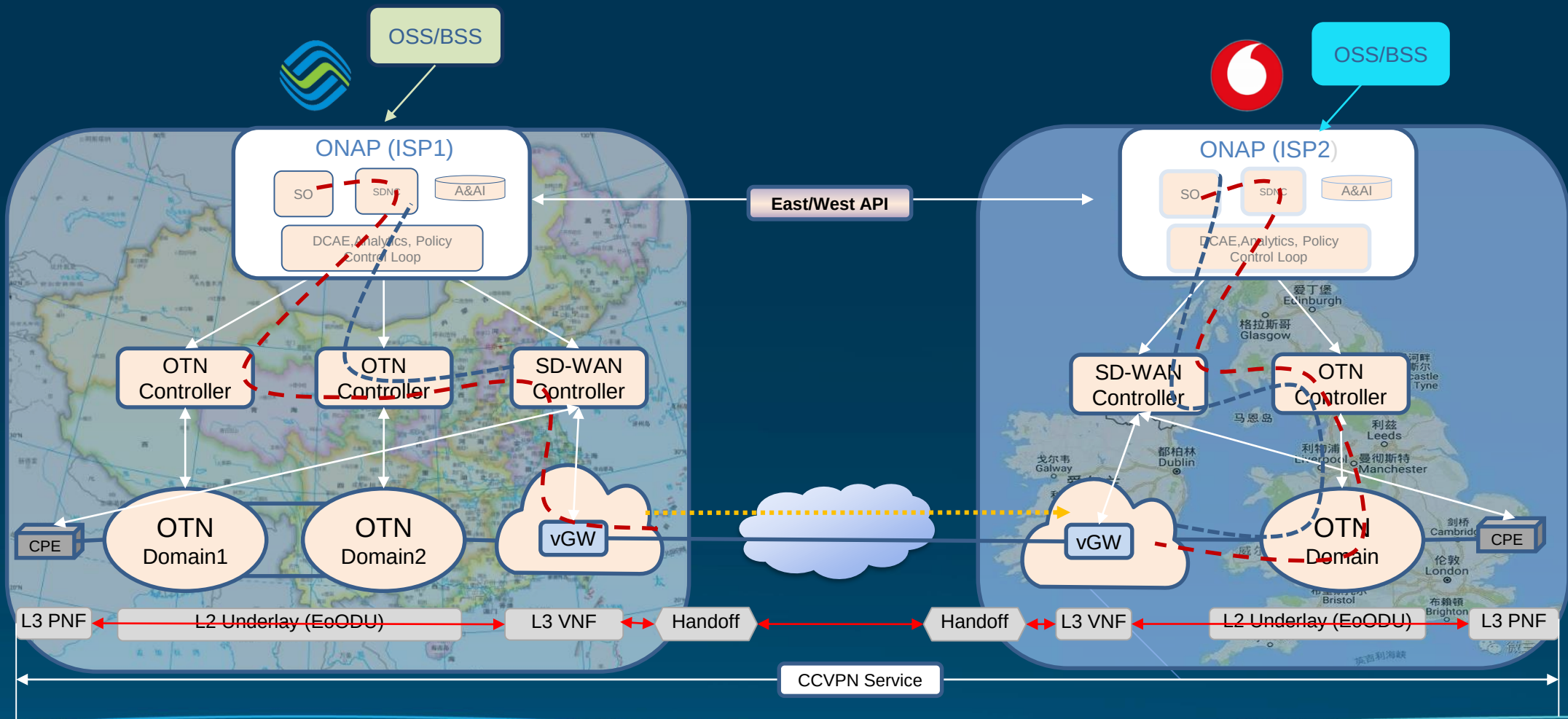
ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service



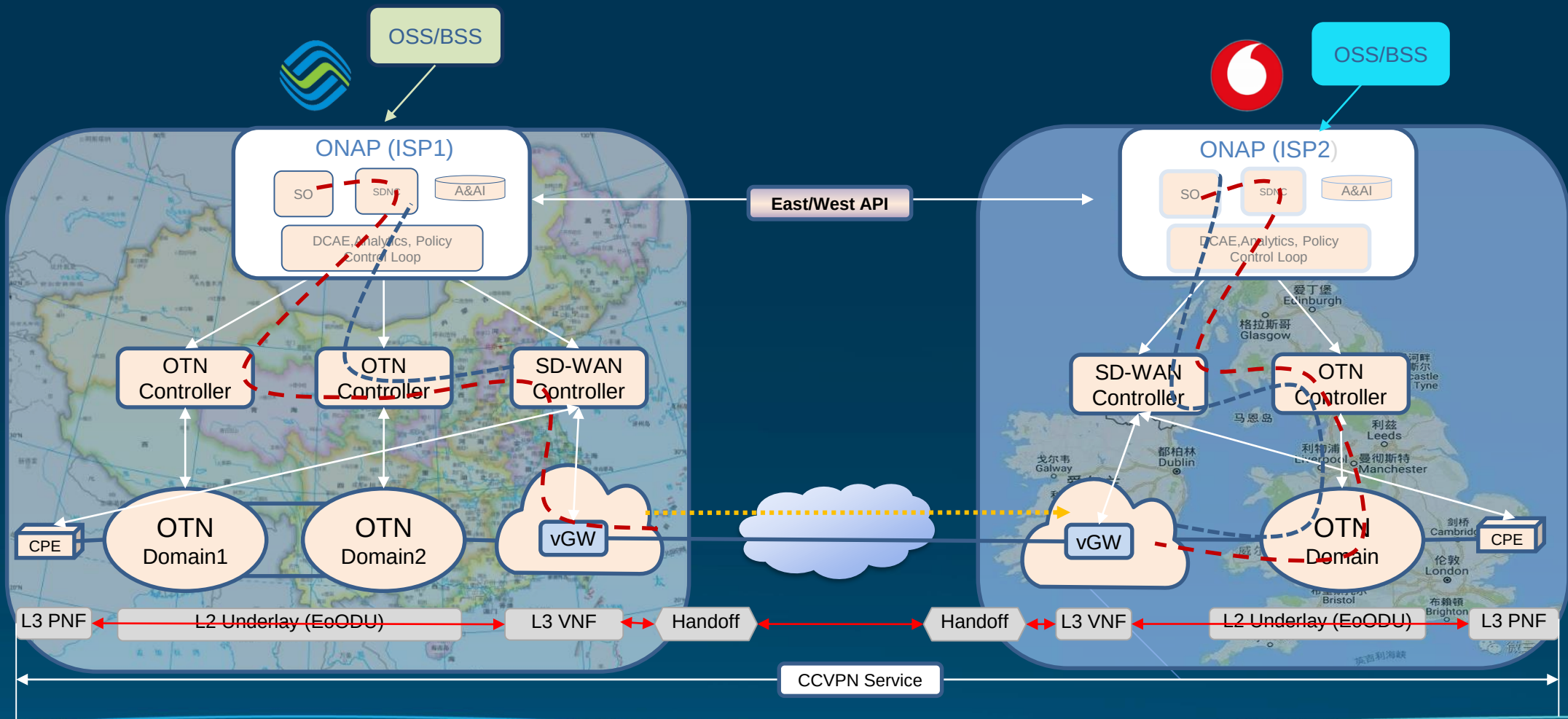
ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service



ONAP-Powered Connectivity Service: CCVPN Usercase

Cross Operator, Cross Domain, Cross Layer VPN Service



Thank you