


FNAL ESCPS Efforts

Phil DeMar, Andrey Bobyshev, Kazim Hussain
October 26, 2010

Project Information:

ESCPS Documents:

https://plone3.fnal.gov/P0/ESCPS

ESCPS Software TRAC repository:

https://damsl.cis.udel.edu/cgi-bin/trac-escps.cgi

Outline

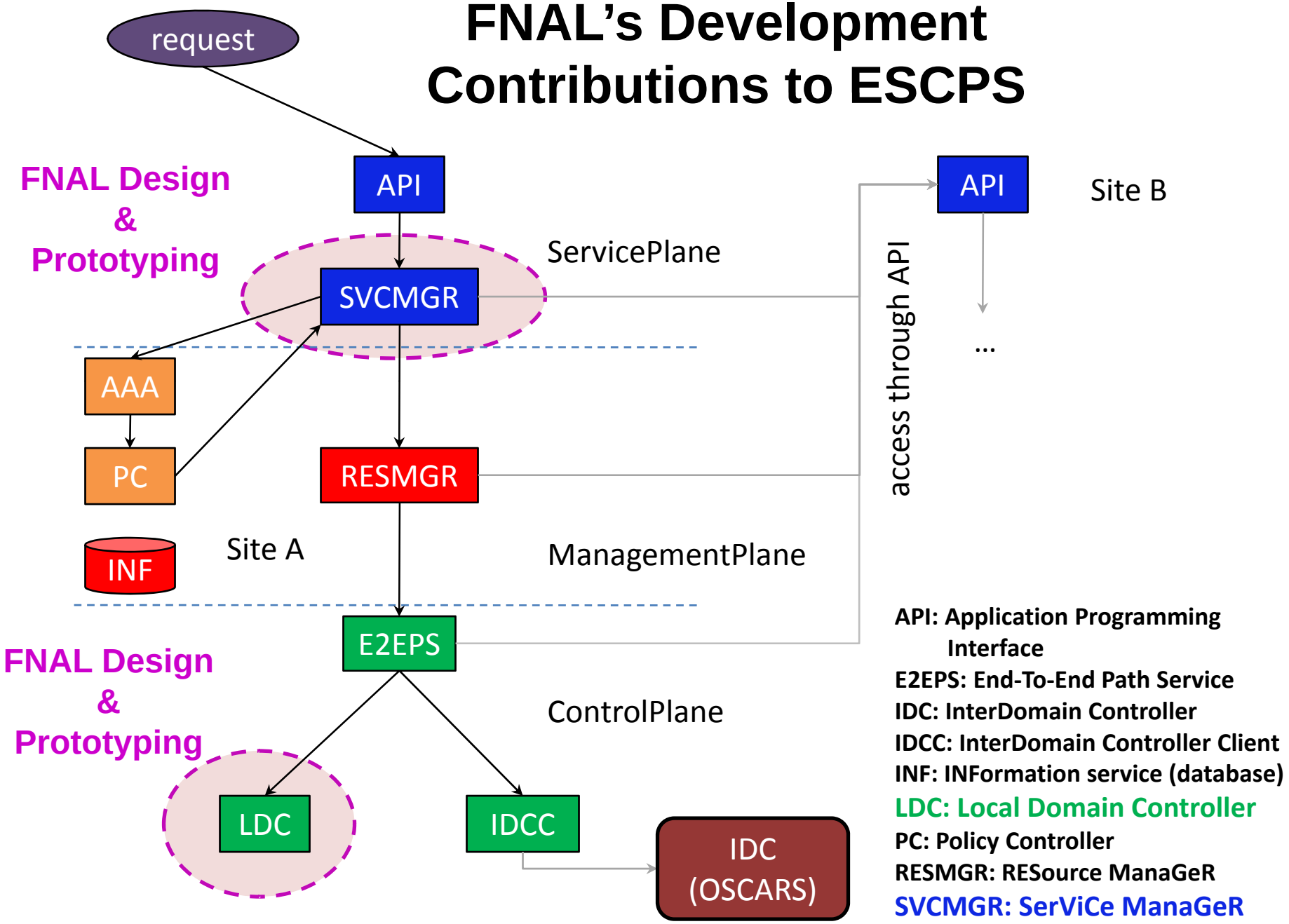
- Overview
- LDC
- SVCMGR

OVERVIEW

FNAL ESCPS Personnel & Expenditures

- Personnel:
  - Lead developer: Andrey Bobyshev
  - Contract software developer: Kazim Hussain
- Budget:
  - In FY10, spent at ~$150k of $180 year 1 budget
    - Almost all personnel & outside contractor costs
    - Very minimal travel (~$2k)
  - Expect to catch up to project spending profile in year 2

FNAL's Development Contributions to ESCPS



API: Application Programming Interface
E2EPS: End-To-End Path Service
IDC: InterDomain Controller
IDCC: InterDomain Controller Client
INF: INFormation service (database)
LDC: Local Domain Controller
PC: Policy Controller
RESMGR: RESource ManaGeR
SVCMGR: SerViCe ManaGeR

Service Manager (SVCMGR)

Accept client's requests, assign unique ID and provide status information

- SVC parameters:
  - createReservation (AFE@domainA, AFE2@domainB, startTime, endTime,BWout, BWin,QoS-Class,...)
    - createReservation(srcIP blocks,dstIPblocks,startTime, endTime,...)
  - listReservations
  - queryReservation
  - cancelReservation
  - joinReservation
  - modifyReservation
  - getAFEInfo - provide information about available AFEs at local and remote sites based on various search criterios

Service Manager (SVCMGR) Status

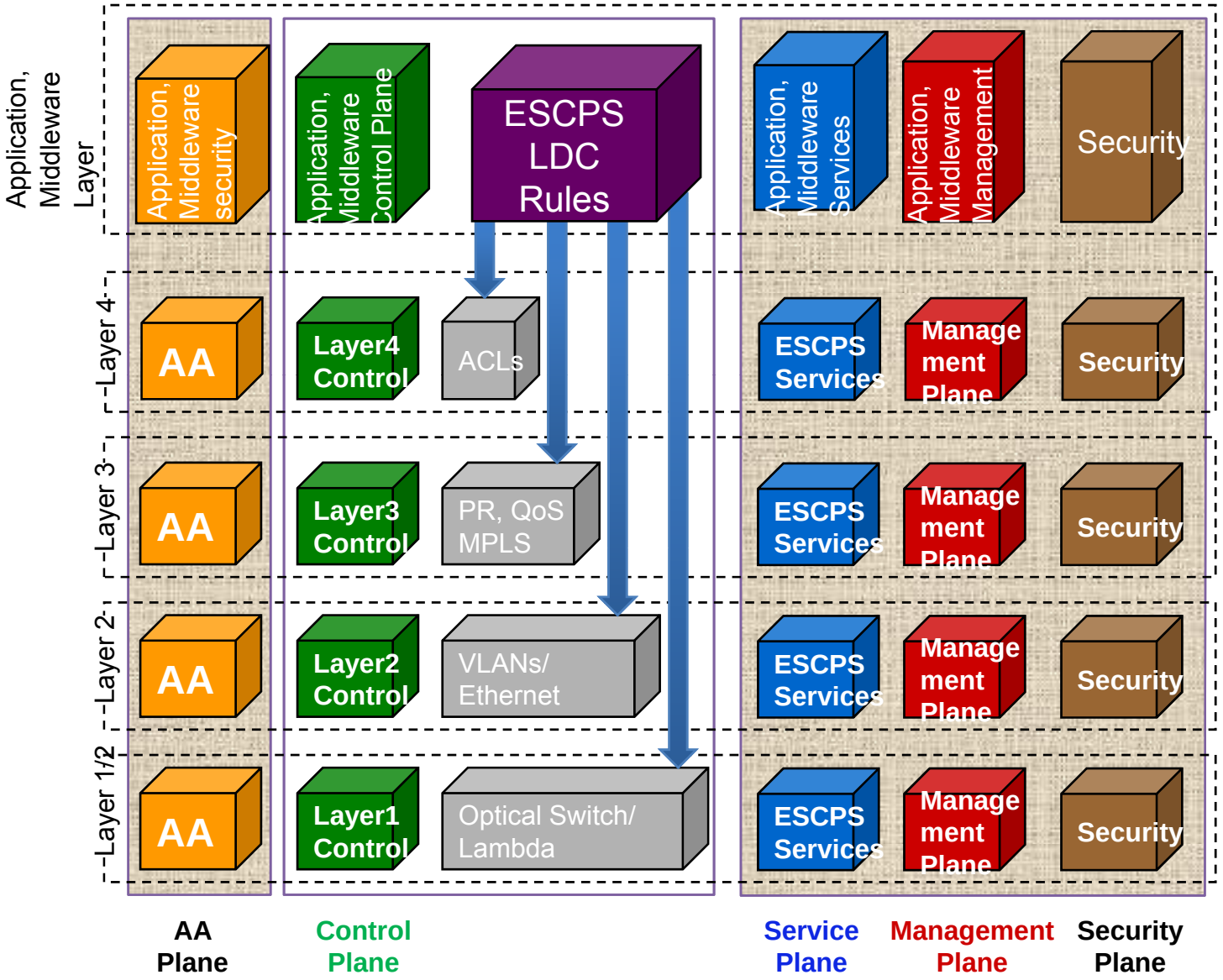
- Detailed design note completed
- Web Service interface is done
- Ticket status and progress visualization
- Software release of SVSMGR in alpha phase
- Issues:
  - Dependency on other ESCPS components is simulated
  - Authorization & security
  - Change of ticket's parameters in-fly
- Future directions & efforts:
  - Integration into general ESCPS framework (w/ BNL)
  - Potential inclusion of XPS support (w/ U-Del)

LDC – ESCPS Local Domain Controller

Modifying site network on-demand to accommodate requested services

- LDC parameters:
  - setSiteVirtualPath (srcAFE, dstAFE, cktID, flows, BW, QoS)
  - getVirtualPathStatus(Id)
  - GetVirtualPathInfo (ID)
  - updateSiteVirtualPath(Id, params)
  - addFlows (src, srcOptionalSrcAttr, dst, dstOptionalAttr, circuit, flows, BW, QoS) – multi form of input parameters will be supported
  - deleteFlows
  - getKnownAFE – return information about locally defined AFEs, lust or XML-Docs
  - getAFEInfo - return info about specific AFE
  - getKNownCircuit
  - getCircuitInfo
  - getQoSModel

LDC Multi-Layer Capability



Basic Definitions (LDC applicability)

Flows

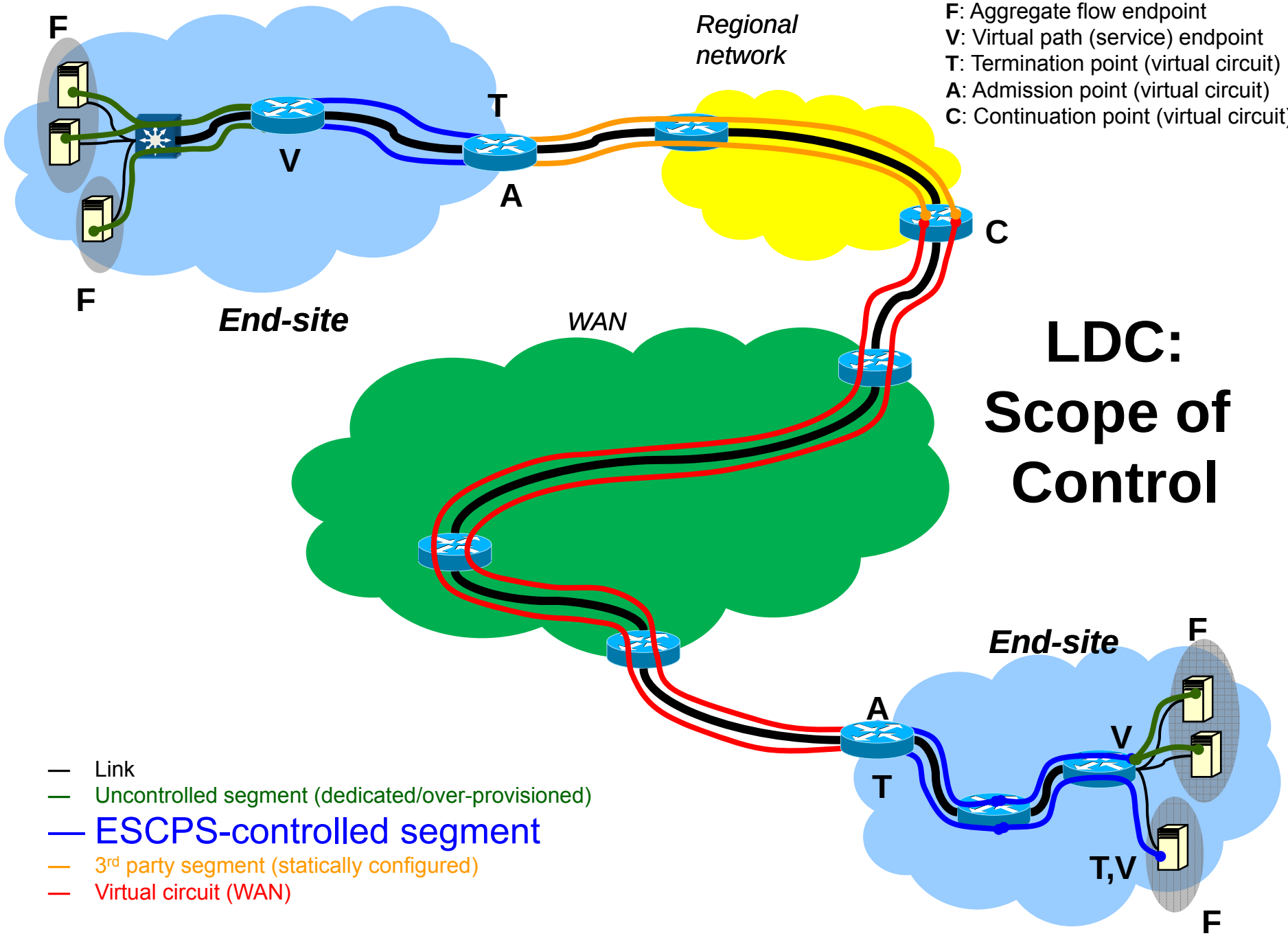
- ***Flow:*** Unidirectional stream of packets identified by common set of keys:
  - Source/dest. IP addr., source/dest. port #, protocol ID, others (ie. DSCP field)
- ***Application Flow:*** Bidirectional aggregation of flows from a common data movement
- ***Aggregate Flow End-Point Entity (AFEE):*** A physical or logical entity that sources or sinks application flows

Circuits / Paths

- ***Virtual circuit:*** Dedicated network path between layer 3 end points, with no intermediate layer 3 hops, that provides specific network services for designated application flows
- ***Virtual network service path:*** The portion of an end-to-end network path for which network services are provided
- ***End-to-end network path:*** Network path between source & destination AFEs

Devices

- ***Circuit Termination Device:*** Network device on which a circuit terminates
- ***Circuit Transit Device:*** Network device that supports a transiting circuit
- ***Circuit Admission Device:*** Network perimeter device interface where a circuit enters a network domain

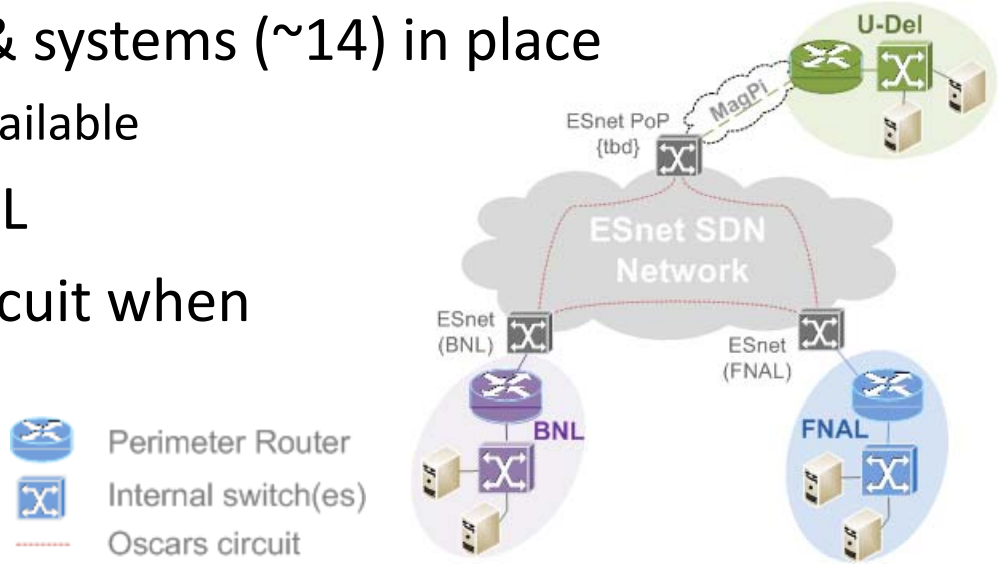

F
V
T
A
F
End-site
Regional network
C
WAN
End-site
F
A
V
T
T,V
F
F: Aggregate flow endpoint
V: Virtual path (service) endpoint
T: Termination point (virtual circuit)
A: Admission point (virtual circuit)
C: Continuation point (virtual circuit)
LDC: Scope of Control
Link
Uncontrolled segment (dedicated/over-provisioned)
ESCPS-controlled segment
3rd party segment (statically configured)
Virtual circuit (WAN)

Local Domain Control (LDC) Status

- Detailed design note (in writing phase)
- XML schemas are in alpha stage
- Software development in pre-alpha stage
- Issues:
  - The testbed is disassembled (see the next slide)
  - Single vendor (Cisco) testbed – when existed
- Future directions & efforts:
  - Integration into general ESCPS framework (w/ BNL)
  - Multi-vendor testbed/more diversity of equipment
  - More widely present ESCPS Network model with other sites

FNAL ESCPS TEST BED Status

- Disassembled as in spring as part of overall Comp. Div. response to computer room power crisis
  - Retained enough test systems for ESCPS development
  - But general WAN/LAN test infrastructure wasn't available
- Test bed infrastructure in process of restoration
  - Local switch (6509) & systems (~14) in place
    - Also have F10 S50 available
  - OSCARs circuit to BNL
  - Will deploy U-Del circuit when capability is there…



ESCPS/LDC Site Network Model

Andrey Bobyshev, Fermilab,
10/27/2010

Main intended audience of this talk is Network administrators and engineers.

Basic Abbreviations

- ESCPS – End Site Control Plane Subsystem
- LDC – Local domain controller
- AFEE – Aggregated flows End Entity (End Point)
- DCN – Dynamic Circuit Network
- PBR – Policy Based Routing

Outline:

- Questions: What/What/Why/How – WWW and how
- Basic definitions
- ESCPS View on DCN
- ESCPS/LDC Model of Site Network
- Elements of ESCPS/LDC Network Model
- Examples

Q&A before build anything

1. What do we have?:
 - A complex site network
 - Internet
 - DCN – special networks like an internet
 - “something” that is connected to network and

2. What do we want ?
we want to improve network services: more specifically improve network services by a special treatment of selected *flows*

3. Why the existing technologies, MPLS, VRF cannot be used ?.. - in fact, we can use it but we need it on-demand of applications and on per flow based

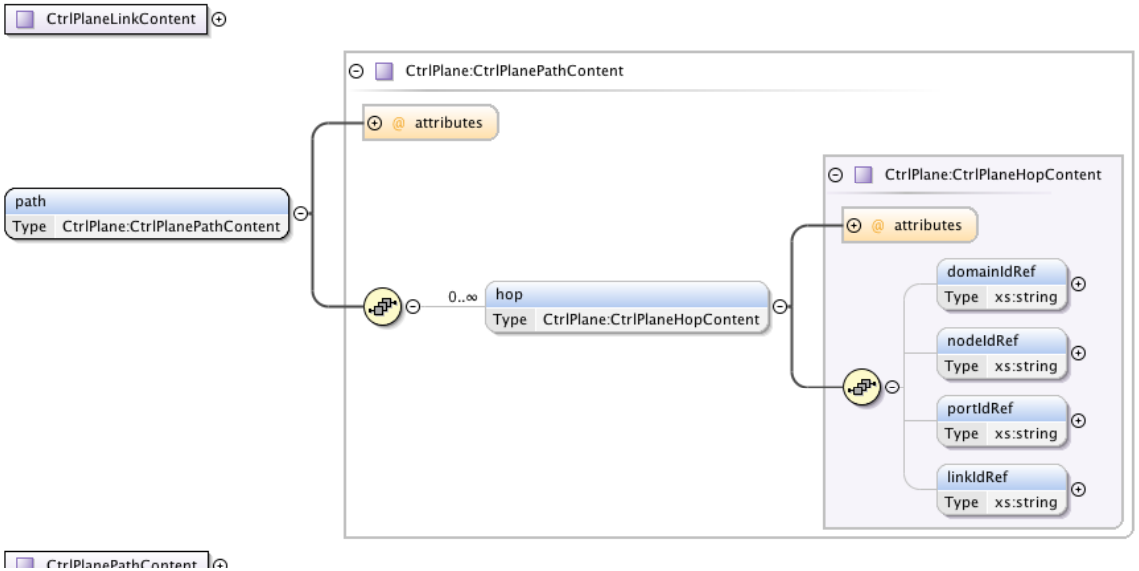
4. How can we do it ? Modify network configuration in-fly

Typical Network Model

- A Host
- A device, router, switch
- Physical links, ethernet, interface
- Many interconnected links, multiple virtual topologies on same physical infrastructure

OSCARS Network model

OSCARS assumes that you have described your topology in the Open Grid Forum (OGF) Network Measurement Working Group (NMWG) control plane topology schema. The NMWG topology schema consists of a hierarchy of **domains, nodes, ports,** and **links**. The table below describes each of these elements



Element	Child Element	Description
domain	node	Represents an administratively-similar set of devices
node	port	Represents a network device. For this installation, it represents a VLSR
port	link	Represent a physical or virtual port on the network. Corresponds to a physical port number and/or DRAGON local-id for this installation.
link		Represents a connection between two ports. For the purposes of this installation, you will likely have one port per link.

NetML
BGP configuration

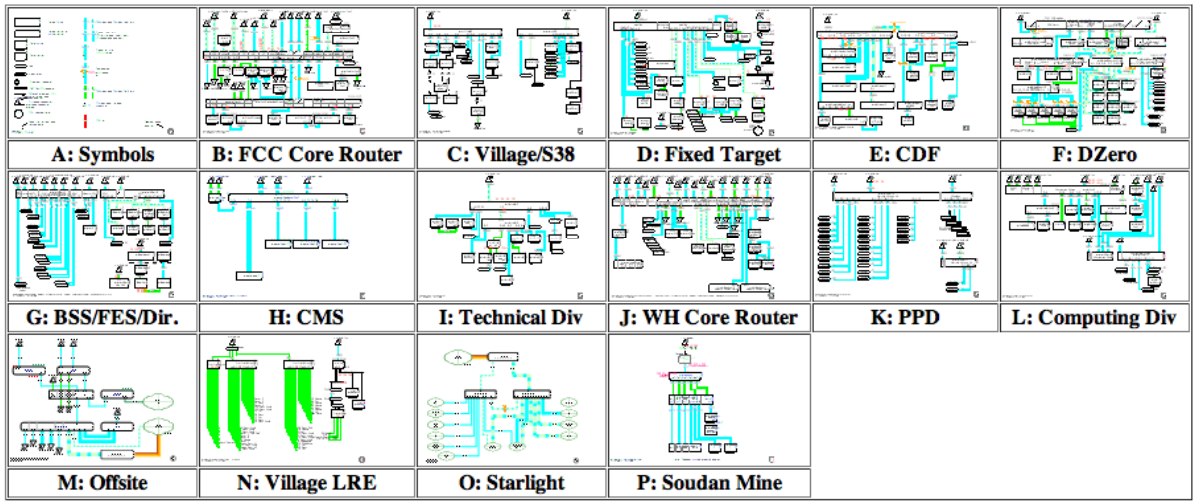
Ivan Santarelli
i.santarelli@tiscali.it
Alexandra Bellogini
a.bellogini@tiscali.it

```
<RouterConf id="r_20_1" Hostname="as20r1">
   <BGPConf as="20">
     <StaticRoutes>
     <s address="0.0.0.0/0"/>
     <s address="..."/>
   </StaticRoutes>
     <Policy type="export">RedistConnected</Policy>
     <PeerGroup name="EBGP">... </PeerGroup>
   </BGPConf>
</RouterConf>
```

Specify static routes

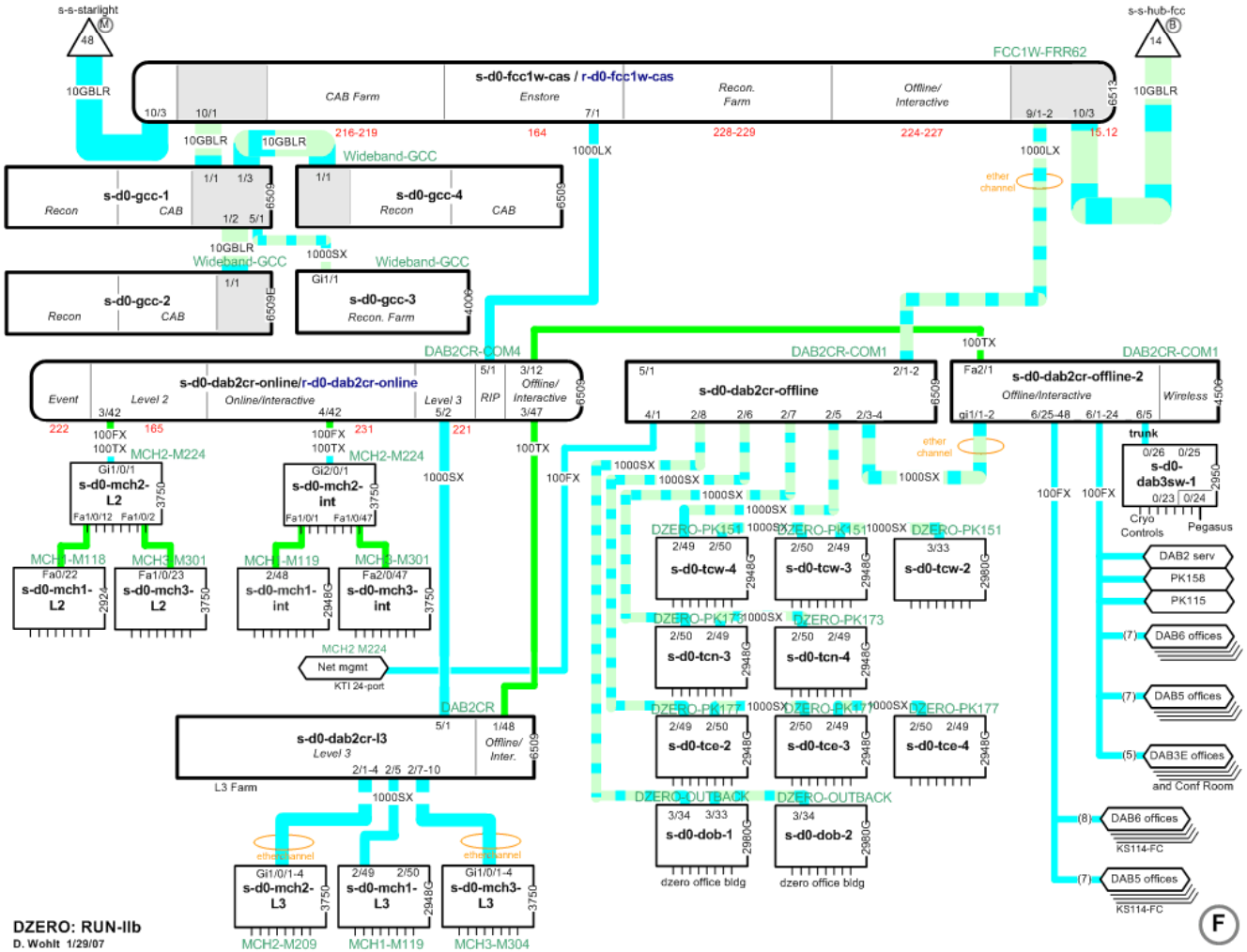
Summary Diagram of Fermilab Network

Fermilab Site Network



- ~317 devices
- ~2500 links

Dzero portion



- ~30 devices
- ~100 physical L1 links
- ~200 link's ports
- ~L2/L3 – connections ???

How: PBR, QoS,MPLS, VRF and others

- Back to our objective – special treatment of selected flows
- Multiple tasks to configure any particular technology
- Multiple implementations to accomplish the same task – a particular implementation may be good for one site and not applicable for another

To accomplish our goals we need to change configuration of network devices in-flight. That is something that network administrators don't like for many reasons

Create a base line configuration and allow to modify some particular fragments of the configurations on-demand

Basic flow definitions

***Flows*:** A unidirectional stream of IP packets identified by common set of keys such as source/destination IP addresses, source/destination port numbers or range of ports if applicable, protocol ID, TOS and DSCP fields. In context of ESCPS project, a list of keys selected to identify flows can be variable. A subset of listed above keys can be selected to identify flows for any specific tasks or applications

Application Flows (APPLFLOWS): Flows associated with a particular application and aggregated per each direction of application's traffic

Aggregated Flows (AGGFLOWS): multiple application flows aggregated for each direction of traffic (bidirectional flows) that associated with one or multiple applications or aggregated based on other schemes as defined by site's needs.

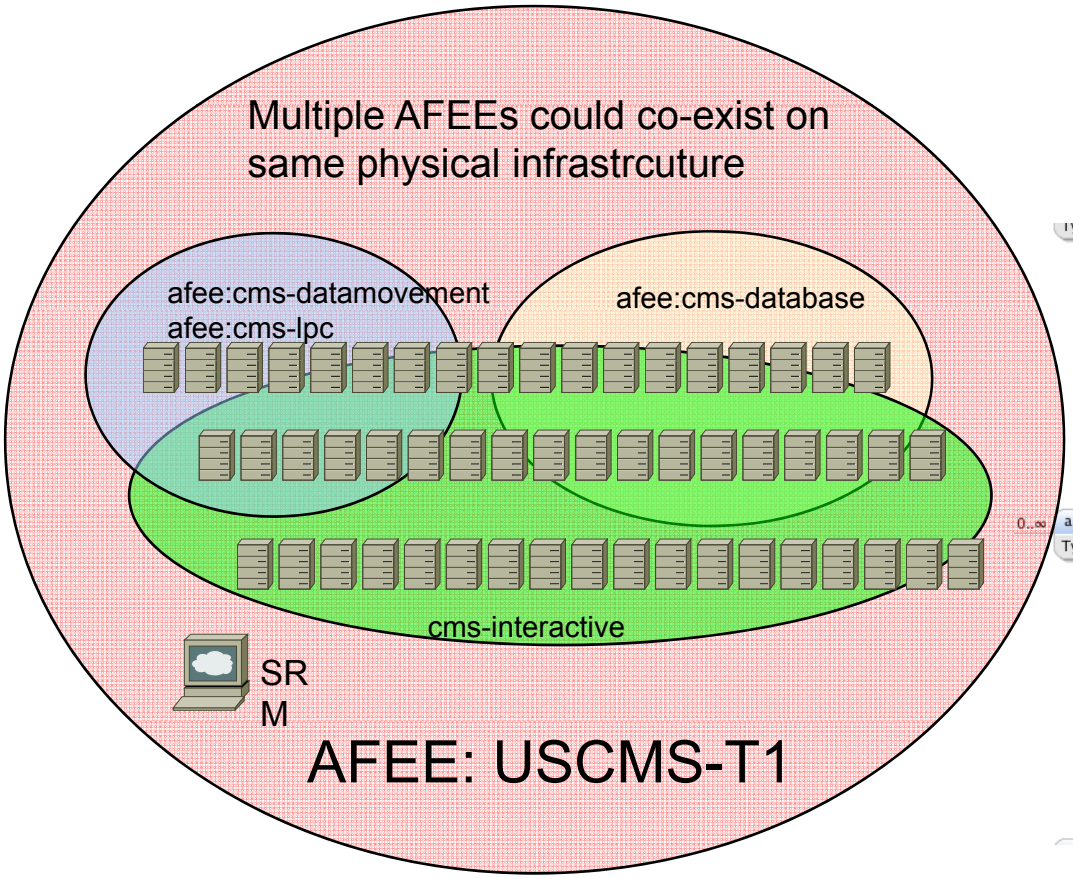
Aggregated Flows End Entity (AFEE): An entity of IP network that generate or sinks aggregated flows.

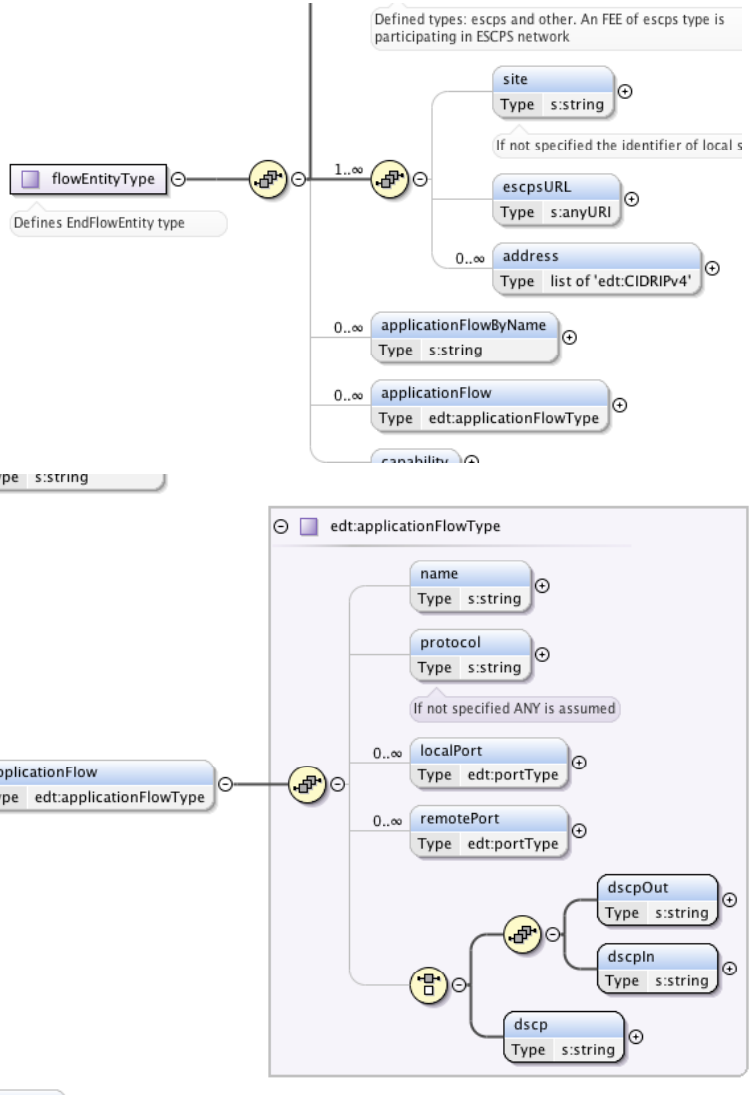
Application – a logically grouped software and/or hardware, directly or indirectly accomplishing some tasks or providing services

Aggregated Flow End Entity (AFEE) is an End Point in ESCPS Network

Aggregated Flows (AGGFLOWS): *multiple application flows aggregated for each direction of traffic (bidirectional flows) that associated with one or multiple applications or aggregated based on other schemes as defined by site's needs.*

Aggregated Flows End Entity (AFEE): *An entity of IP network that generate or sinks aggregated flows.*





AFEE Definition in BNF

<AFEE> ::= <list-of-VLSM>
<***list-of-applications***><***site-network-infrastructure***>

<list-of-VLSM> ::= <VLSM> | <VLSM><deliminator><list-of-VLSM>

<list-of-applications> ::= <application> | <application><delimator><list-of-applications>

<VLSM ::= <network-ip-address>"/"<network-mask>

<application> ::= <list-of-protocol-port>

<list-of-protocol-port> ::= <protocol-port> | <protocol-port><delimator>
<list-of-protocol-port>

<protocol-port> ::= <protocol>|<protocol><source-port>
| <***protocol***><***destination-port***>
| <***protocol***><***source-port***><***destination-port***>

<source-port> ::= <port> | <port><deliminator><source-port>

<destination-port> ::= <port> | <port><deliminator><destination-port>

<port> ::= <number> | <symbolic-name> | <number> "-" <number>

<deliminator> ::= <whitespace> | <comma>

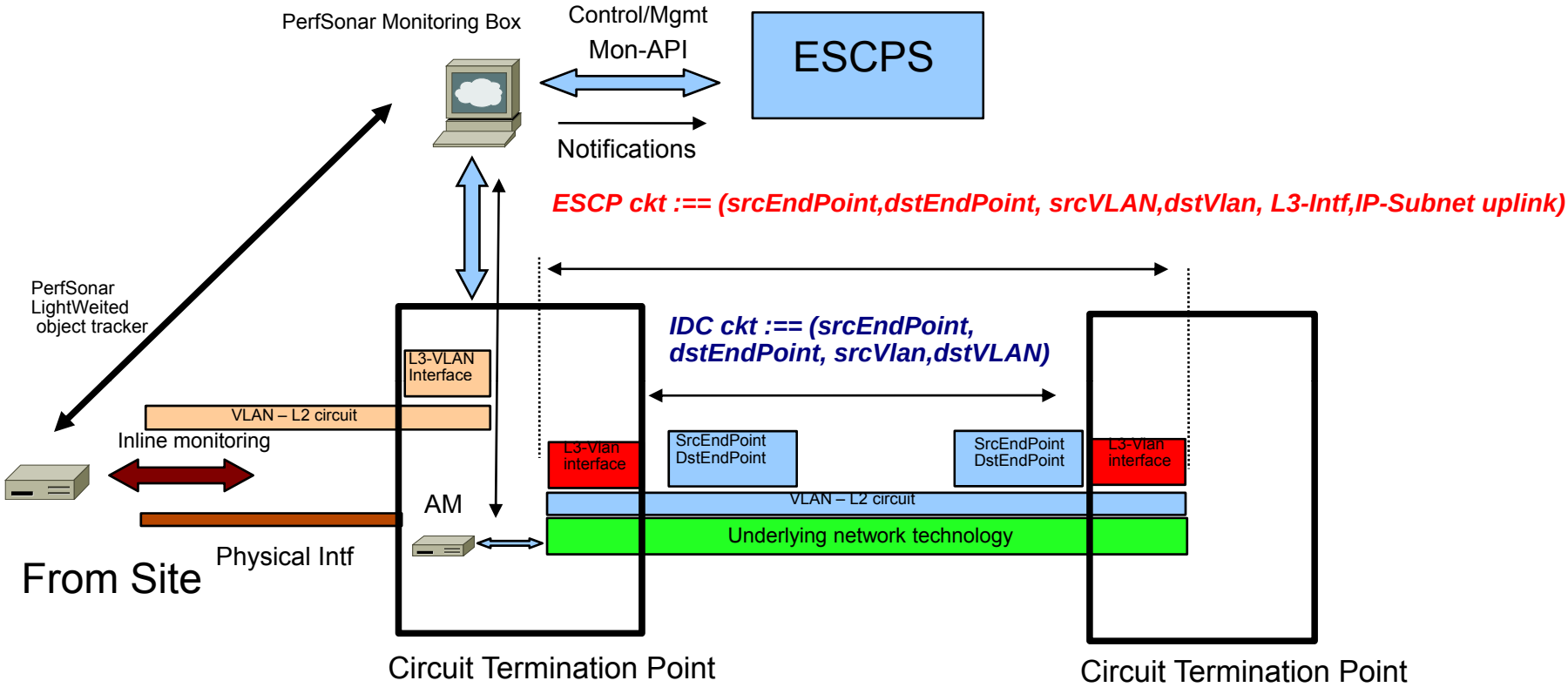
Example of AFEE definition

```
<Definition>
    <applicationFlow>
        <!--  Critical traffic is marked by DSCP AF21
-->

<applicationName>critical</applicationName>
        <dscp>af21</dscp>
                    <flowEntity>
<applicationClass>datamovement</applicationClas
s>                          <site>fnal.gov</site>
                            <address>131.225.204.0/22</address>
      </applicationFlow>    <address>131.225.188.0/22</address>
</Definition>               <address>131.225.184.0/22</address>

                             <applicationFlowByName>critical</applicationFlowByName>

                  <applicationFlow>
                        <applicationName>srm</applicationName>
                        <protocol>tcp</protocol>
                        <remotePort>1000 - 65000</remotePort>
                        <applicationClass>transactional</applicationClass>
                   </applicationFlow>

                  </flowEntity>
```

AFEE Definition Summary

- AFFE is an End Point/Entity in ESCPS network (similar as a host is an end point in IP network)
- AFEE defines applications that could belong to that AFEE and how to identify/match flows of these applications
- Points to elements of network infrastrcuture that need to be configured dynamically on-demand in order to provide desired characteristics for AFEE flows.

Two points circuit model



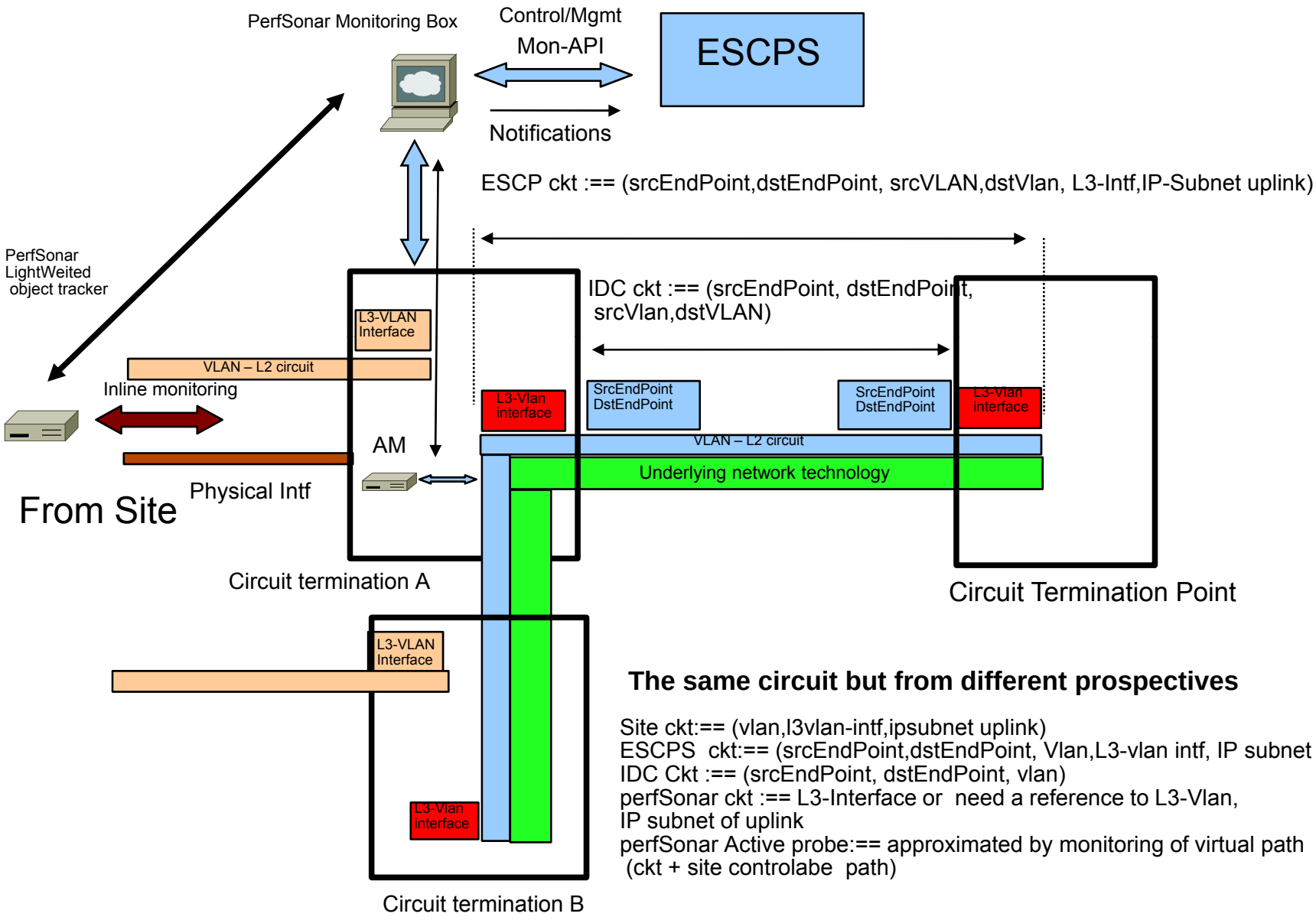
AM – Active measurements from the inside of Termination Point
As of now there are a number of proprietary implement ions, vendor specific (SLA monitor, object tracker, EventManager)

The same circuit but from different prospectives

Site ckt:== (vlan,l3vlan-intf,ipsubnet uplink)
ESCPS ckt:== (srcEndPoint,dstEndPoint, Vlan,L3-vlan intf, IP subnet uplink)
IDC Ckt :== (srcEndPoint, dstEndPoint, vlan)
perfSonar ckt :== L3-Interface or need a reference to L3-Vlan, IP subnet of uplink
perfSonar Active probe:== approximated by monitoring of virtual path (ckt + site controlabe path)

AM Main functions :
- is ckt UP ?
- Is performance OK ?

Multipoint circuit model - Highway



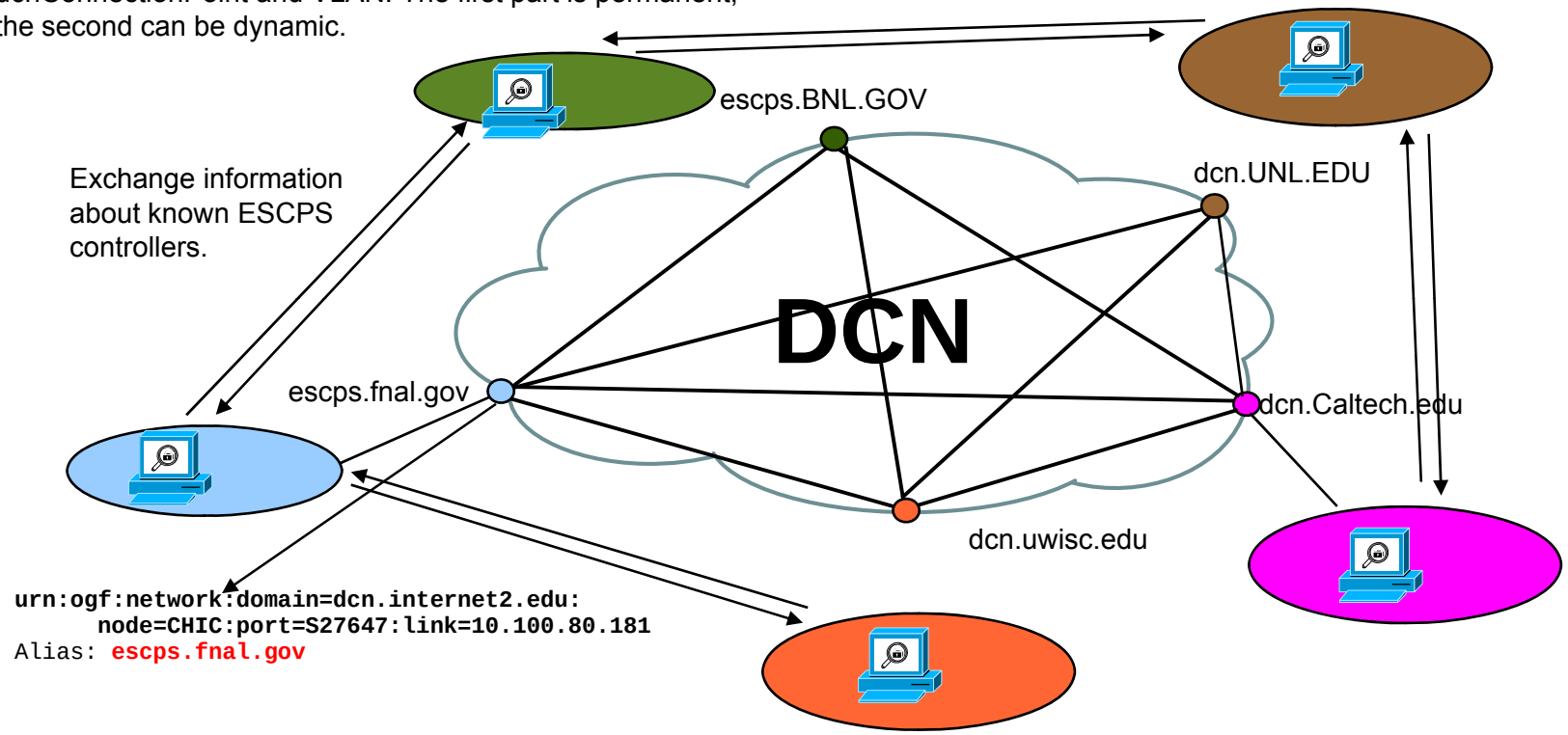
The same circuit but from different prospectives

Site ckt:== (vlan,l3vlan-intf,ipsubnet uplink)
ESCPS ckt:== (srcEndPoint,dstEndPoint, Vlan,L3-vlan intf, IP subnet uplink)
IDC Ckt :== (srcEndPoint, dstEndPoint, vlan)
perfSonar ckt :== L3-Interface or need a reference to L3-Vlan,
IP subnet of uplink
perfSonar Active probe:== approximated by monitoring of virtual path
(ckt + site controlabe path)

ESCPS pointview on DCN

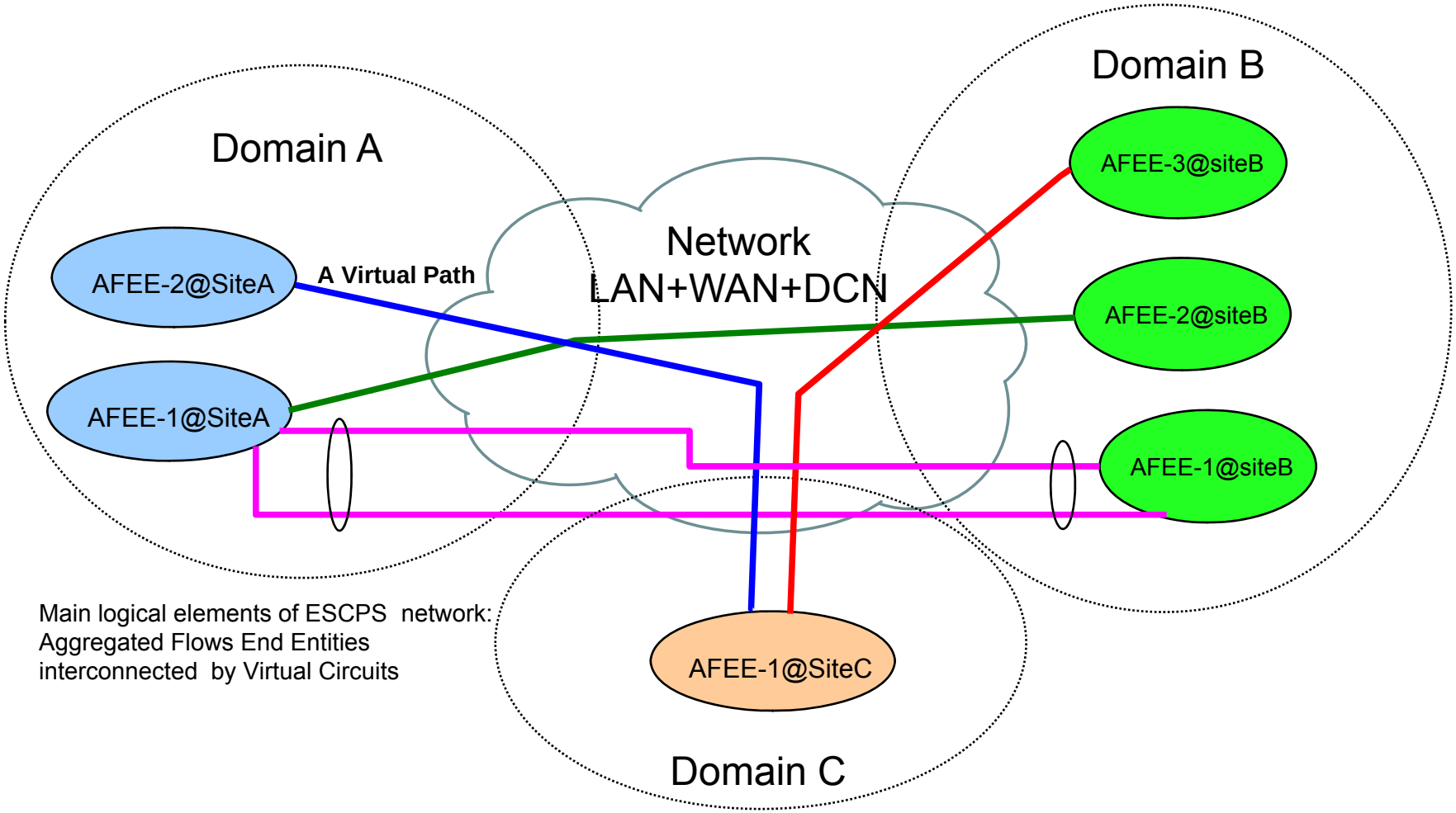
A Site Address in DCN is (dcnConnectionPointAddress, VLAN)

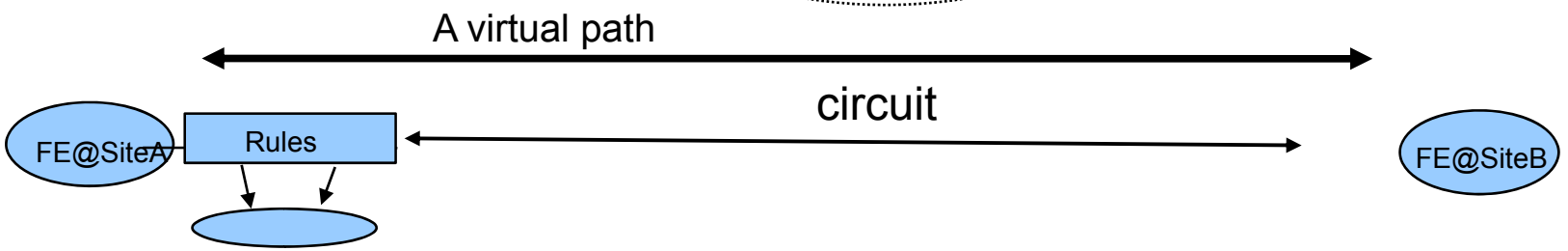
Site's Address in DCN network consists of two parts, an address of dcnConnectionPoint and VLAN. The first part is permanent, the second can be dynamic.



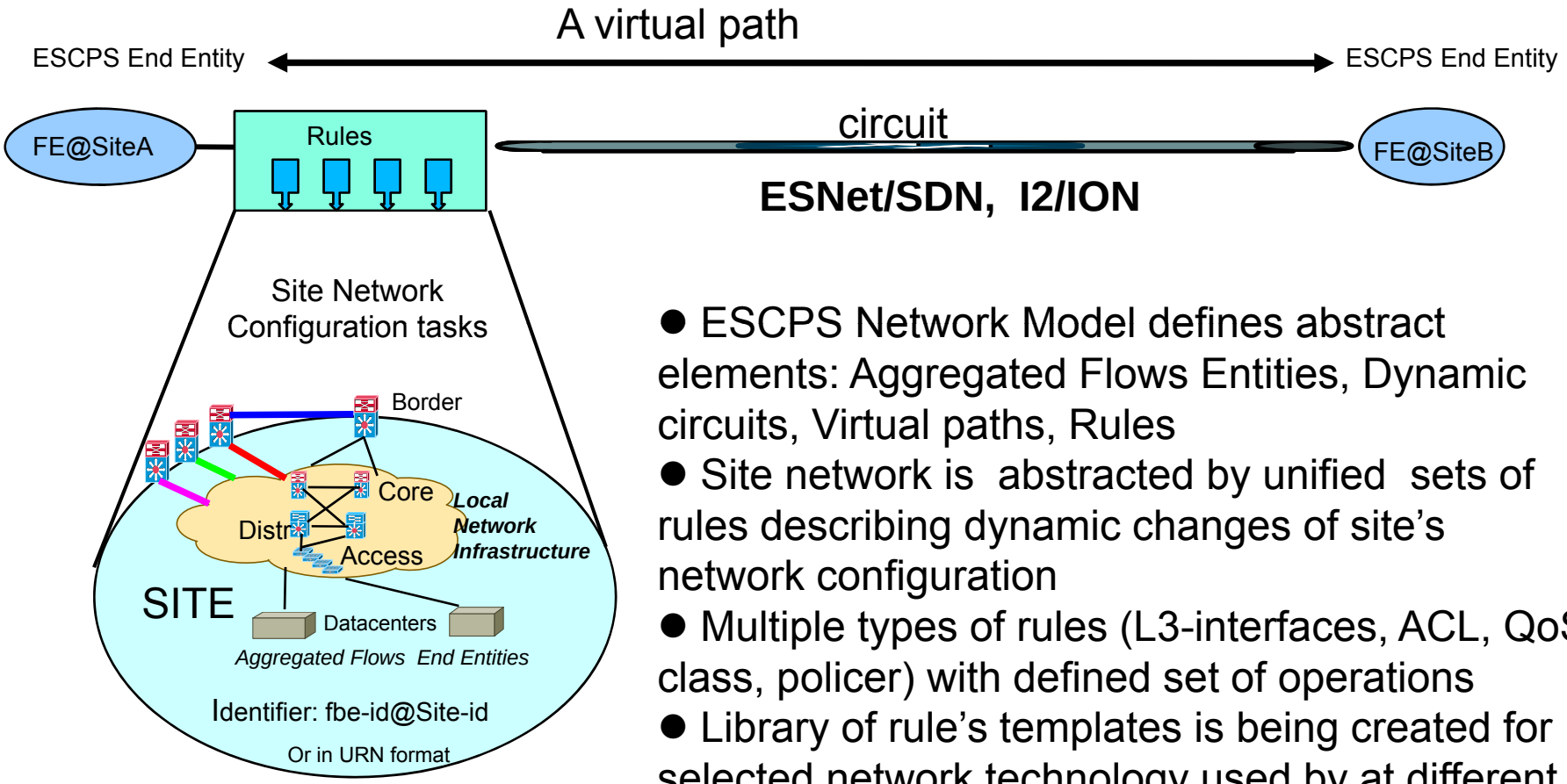
- exchange knownlege about known ESCPS controllers
- query site's ESCPS controller for detailed data and DCN address directly
- assumption that any site can establish a circuit to any other site if its address is known
- to find it out: submit request to DCN provider's IDC, analyzes response

ESCPS Conceptual View on Network



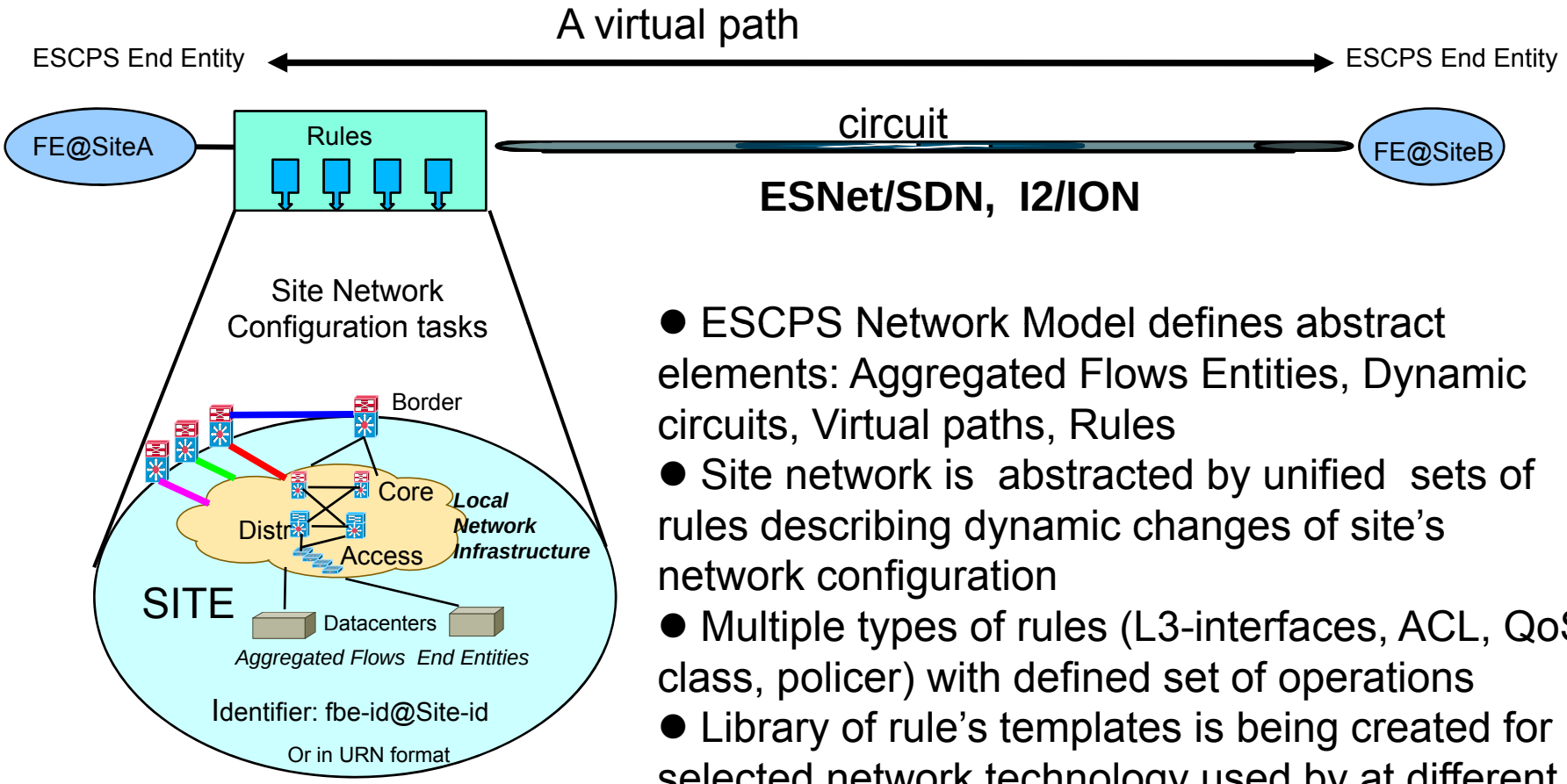


ESCPS Network Model



- ESCPS Network Model defines abstract elements: Aggregated Flows Entities, Dynamic circuits, Virtual paths, Rules
- Site network is abstracted by unified sets of rules describing dynamic changes of site's network configuration
- Multiple types of rules (L3-interfaces, ACL, QoS class, policer) with defined set of operations
- Library of rule's templates is being created for selected network technology used by at different sites

ESCPS Network Model



- ESCPS Network Model defines abstract elements: Aggregated Flows Entities, Dynamic circuits, Virtual paths, Rules
- Site network is abstracted by unified sets of rules describing dynamic changes of site's network configuration
- Multiple types of rules (L3-interfaces, ACL, QoS class, policer) with defined set of operations
- Library of rule's templates is being created for selected network technology used by at different sites

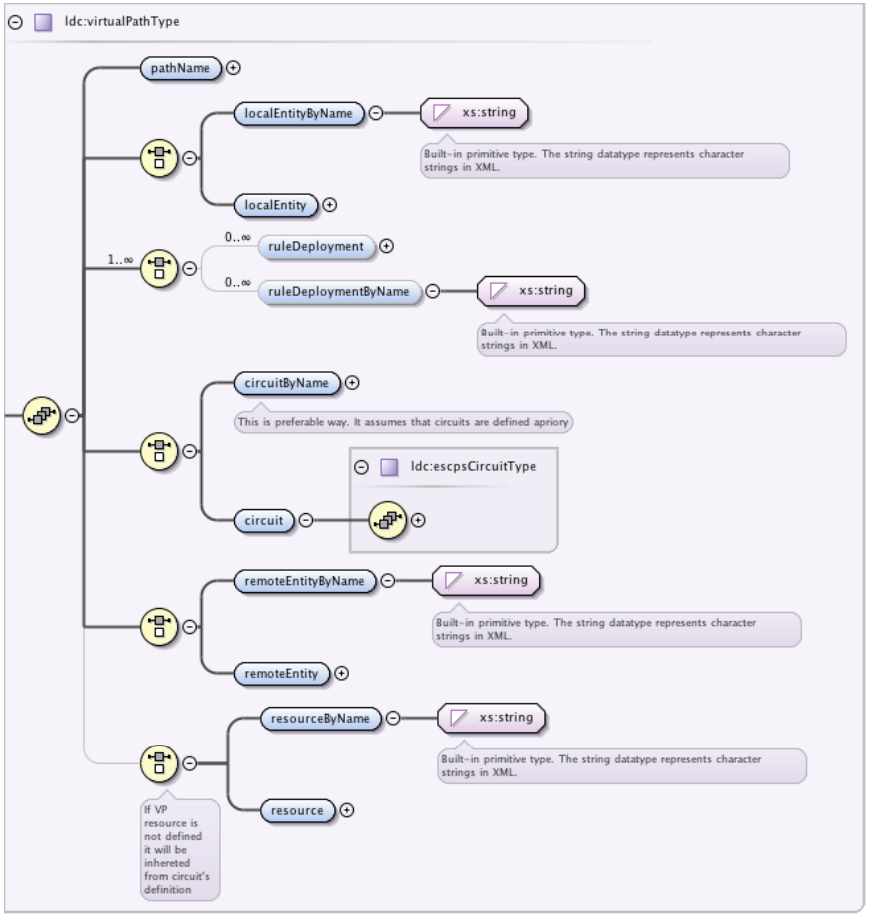
A Virtual Path

Application's API

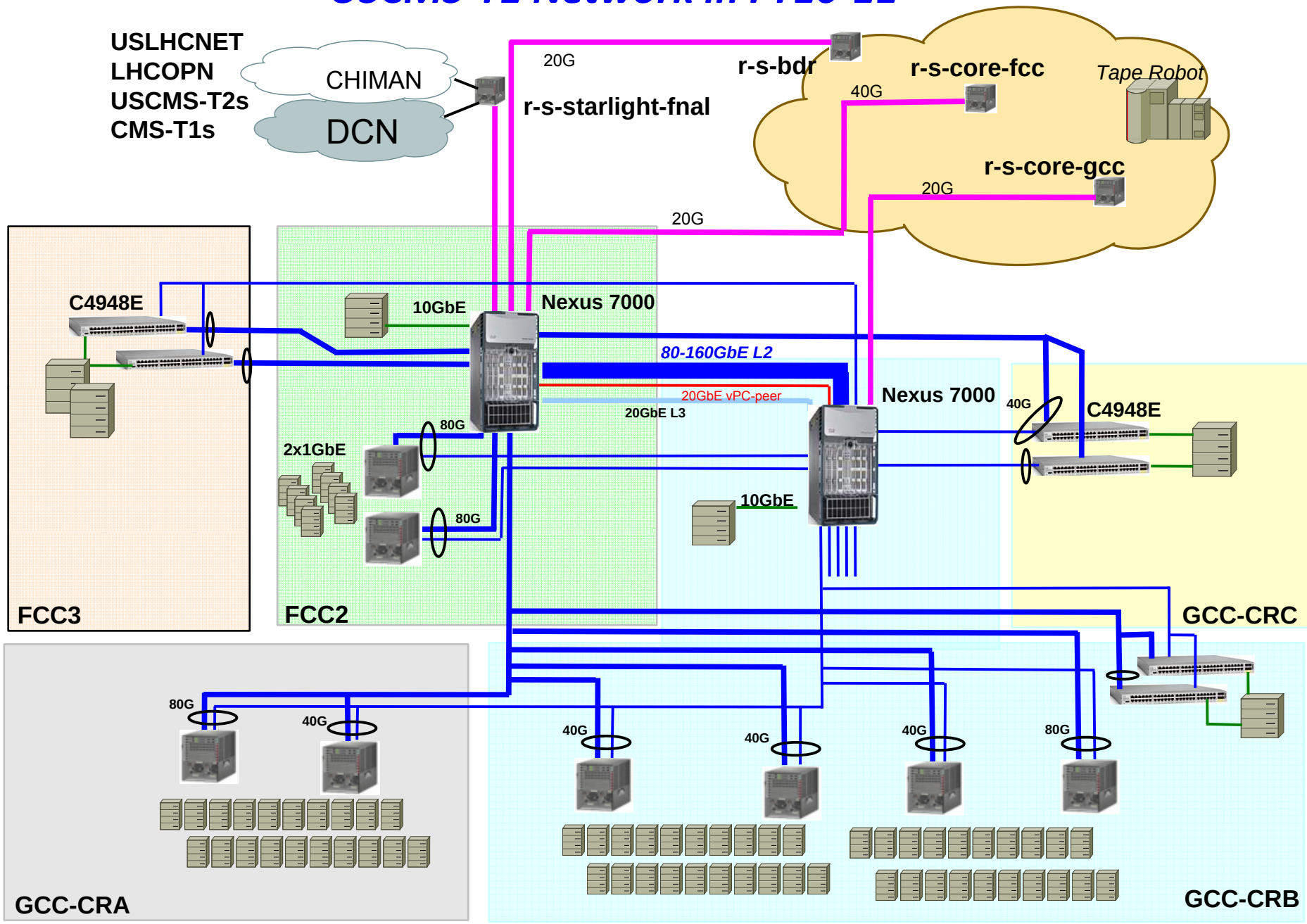
- addfFlows – add flows to all rules that have property 'flows'
- removefFlows – remove flows from all rules that have property 'flows'
- getStatus – a combined status (rules are loaded/not loaded,DCN circuit is up/down/reserve)
- getStatistics (capacity, bytesIn,bytesOut)

Internal commands (autodial):

- setup – call DCN to reserve/establish circuit , load rules
- teardown – drop DCN circuit, remove all operational flows, rules are not unloaded
- reset circuit – same as teardown but also unload all rules



USCMS-T1 Network in FY10-11

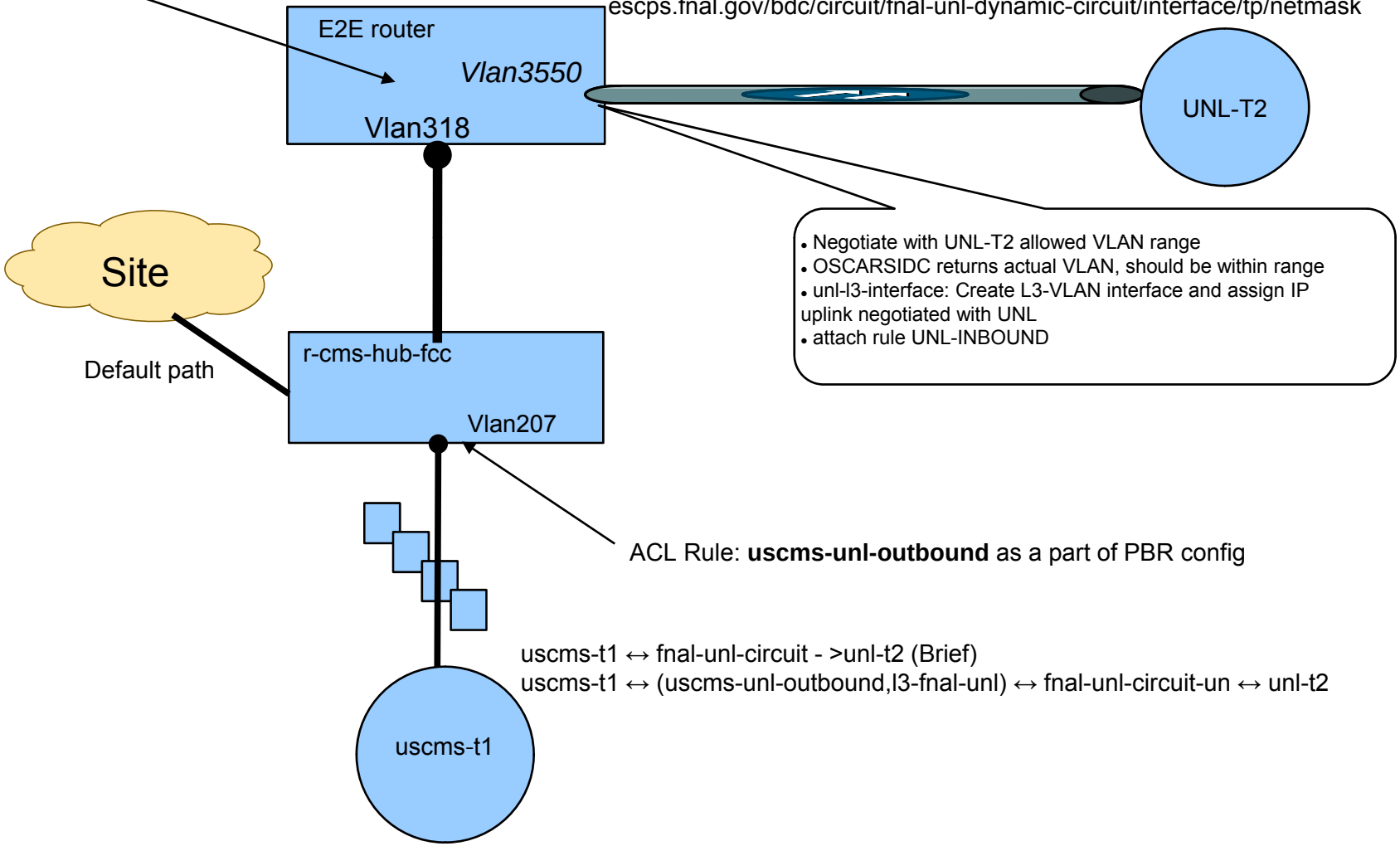


$AB, 2010-10-01

FNAL-UNL-DYNAMIC Circuit Example

Rules to configure:
L2-VLAN
L3-FNAL-UNL INTERFACE
ACL to “attach” selected flows to PBR configuration

escps.fnal.gov./bdc/circuit/fnal-unl-dynamic-circuit
escps.fnal.gov/bdc/circuit/fnal-unl-dynamic-circuit/interface
escps.fnal.gov/bdc/circuit/fnal-unl-dynamic-circuit/interface/vlan
escps.fnal.gov/bdc/circuit/fnal-unl-dynamic-circuit/interface/uplink
escps.fnal.gov/bdc/circuit/fnal-unl-dynamic-circuit/interface/tp/
escps.fnal.gov/bdc/circuit/fnal-unl-dynamic-circuit/interface/tp/ipaddress
escps.fnal.gov/bdc/circuit/fnal-unl-dynamic-circuit/interface/tp/netmask



- Negotiate with UNL-T2 allowed VLAN range
- OSCARSIDC returns actual VLAN, should be within range
- unl-l3-interface: Create L3-VLAN interface and assign IP uplink negotiated with UNL
- attach rule UNL-INBOUND

ACL Rule: **uscms-unl-outbound** as a part of PBR config

uscms-t1 ↔ fnal-unl-circuit - >unl-t2 (Brief)
uscms-t1 ↔ (uscms-unl-outbound,l3-fnal-unl) ↔ fnal-unl-circuit-un ↔ unl-t2

Workflow for the example on previous slide

LDC

1.LDC gets info for ticket USCMS-T1 → UNL-T2
2. From the model identifies the path and its status (Up,Down,inSetup,inRemove)
3. If status is Up → add flows
4. If vpath is new (loading rules): fnal-unl-l3-interface is a property of circuit but LDC is going to configure it, LDC asks IDC for for the corresponding resources VLAN, Ipaddress,netmask
5. Rule engine generate config by using received properties, and templates for specific CLI syntax,Netconf
6. upload rule
7. if ACL rules are UP, then add new flows if not Rule engine generates new ACL and upload it
8.upload attach rule...
9. LDC checks the status of circuit, if down remove all flows from rules

IDC/BorderDC

From ticket via URL + startTime,endTime get L3-interface properties (vlan,ipaddress,netmask)

Example of rules for configuring PBR

L3-FNAL-UNL Interface rule

r-cms-hub-fcc Router

```
interface Vlan207
 ip address 131.225.207.202 255.255.252.0
 ip helper-address 131.225.189.73
 ip flow ingress
 ip pim sparse-dense-mode
 ip policy route-map CMS-E2E-Circuits
.....
route-map CMS-E2E-Circuits permit 5
 match ip address USCMS-UNL-OUTBOUND
 set ip next-hop 198.151.133.238
!
.........
route-map CMS-E2E-Circuits permit 13
 match ip address LHCOPN
 set ip next-hop verify-availability 198.151.133.238 10 track 131
 set ip next-hop verify-availability 198.49.208.222 20 track 138
.......
```

```
ip access-list extended USCMS-UNL-OUTBOUND
10   permit ip 131.225.184.0 0.0.3.255 129.93.239.128 0.0.0.63
20   permit ip 131.225.188.0 0.0.3.255 129.93.239.128 0.0.0.63
....
80  permit ip host 131.225.207.253 host 129.93.6.22
90  permit ip host 131.225.191.72 host 129.93.6.22
110  permit ip host 131.225.191.253 host 129.93.6.22
2147483640 deny ip any any
```

E2E Router

```
interface Vlan3550
 description Circuit-to-UNL
 mtu 9216
 ip address 198.151.133.26 255.255.255.252
 ip policy route-map LHCOPN-CIRCUIT-INBOUND
```

```
route-map LHCOPN-CIRCUIT-INBOUND permit 5
 match ip address LHCOPN-CIRCUIT-INBOUND
 set ip next-hop 198.151.133.129
!
interface Vlan381
 description CMS-E2E-1-UPLINK
 ip address 198.151.133.238 255.255.255.252
 ip policy route-map LHCOPN-CIRCUIT-OUTBOUND
....
! fanout outbound traffic into corresponding circiuts
! there is a route-map for each circuit
route-map LHCOPN-CIRCUIT-OUTBOUND permit 100
 match ip address USCMS-UNL-OUTBOUND
 set ip next-hop 198.151.133.209
...

ip access-list exended LHCOPN-CIRCUIT-INBOUND
! All inbound traffic from circuits is treated same way
 10 permit ip any any
2147483640 deny ip any any

ip access-list extended USCMS-UNL-OUTBOUND
10   permit ip 131.225.184.0 0.0.3.255 129.93.239.128 0.0.0.63
20   permit ip 131.225.188.0 0.0.3.255 129.93.239.128 0.0.0.63
....
80  permit ip host 131.225.207.253 host 129.93.6.22
90  permit ip host 131.225.191.72 host 129.93.6.22
110  permit ip host 131.225.191.253 host 129.93.6.22
2147483640 deny ip any any
```

ACL Rule

A routing table example

Virtual path:

T1-USCMS-> (PBR@r-cms-fcc2-3/cms-unl-out, unl-in @r-s-starlight-fnal) → Circuit->T2-UNL

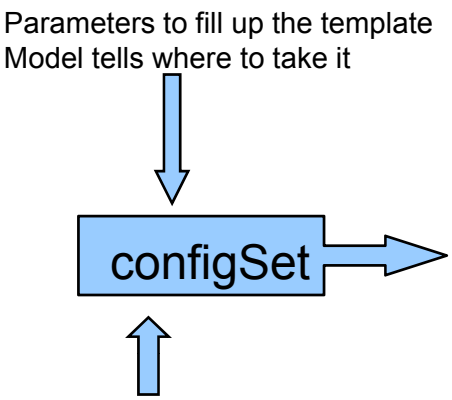
1) Application requests ost(uscms,unl)
2) Virtual path available:
   up- > rules loaded, Circuit is UP
1) Assembly rules, load rules, setup dc-circuit

Local Flows Entity	Rule infrastructure	DCN Circuit	Remote Flows Entity
T1-USCMS@FNAL.GOV			T2-UNL@UNL.EDU
	USCMS-UNL-OUTBOUND@r-cms-hub-fcc	UNL-FNAL-DYNAMIC	
	L3-FNAL-UNL @ E2E USCMS-UNL-OUTBOUND @ E2E		
T1-USCMS@fnal.gov	cms-lhcopn@r-cms-fcc2-3	UWISC-CKT	T2-UWISC@uwisc.edu
	cms-wisc-out@r-s-starlight-fnal		
	cms-wisc-in@r-s-starlight-fnal		

Definitions(for now)

- RULE is a symbolic name assigned to configuration fragment.
- Generic abstraction to represent typical building blocks used for configuring of networks. Actual representation is hidden in template.

Configuring FNAL-UNL Dynamic circuit: L3-Interface L3-FNAL-UNL



TEMPLATE

```
VLAN {INTERFACE_NAME}
#(Assumption that we are using a Cisco switch and
#create VLAN-based L3 Interface)
interface  VLAN {INTERFACE_NAME}
 no switchport
 description {DESCRIPTION}
 mtu {MTU Value}
 bandwidth 10000000
 ip address {IP ADDRESS}  {NETMASK}
 ip flow ingress
 ip policy route-map LHCOPN-CIRCUIT-INBOUND
 load-interval 30
 hold-queue 2000 in
 hold-queue 2000 out
```

```
vlan 3550
!
interface Vlan3550
 description UNL
no switchport
 mtu 9216
 bandwidth 10000000
 ip address 198.151.133.26 255.255.255.252
 ip flow ingress
 ip policy route-map LHCOPN-CIRCUIT-INBOUND
 load-interval 30
 hold-queue 2000 in
 hold-queue 2000 out
```

Permanent configuration fragment:

```
route-map UNL permit 13
 match ip address LHCOPN-CIRCUIT-INBOUND
 set ip next-hop 198.151.133.129

ip access-list exended LHCOPN-CIRCUIT-INBOUND
! All inbound traffic from circuits is treated same way
 10 permit ip any any
2147483640 deny ip any any
```

XML Definition of ACL

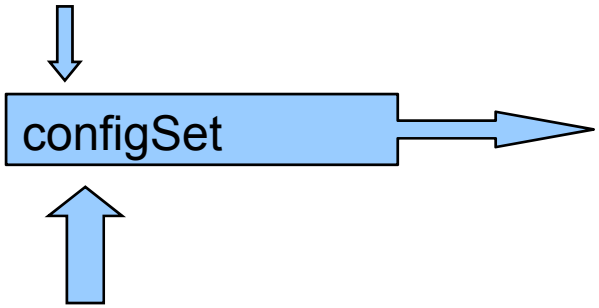
```
<ACLDEFs version="1.0"
    name="ACL_Extended_Default"
    xmlns="http://www.lambdastation.org"
    xmlns:ns="http://www.lambdastation.org">
<ACL_Extended_Default>
  <templateDir>
/home/netadmin/netconf/cfg/Template/ACL_Extended_Default
</templateDir>
  <type>  extended </type>
  <staticFrames> Init Top Bottom Resequence </staticFrames>
  <dynamicFrames> addACLEntries ,
   deleteACLEntries , initACLEntries
deleteACLEntriesByFlow</dynamicFrames>
  <maxSize> 2000 </maxSize>
  <acl> Top addACLEntries Bottom </acl>
 .......
</ACL_Extended_Default>
```

XML *Definition of ACL (continue...)*

```
.....
  <OPERATION>
    <configSet> Init addACLEntries Bottom</configSet>
   <configUnset> Init Bottom </configUnSet>
      <configure>
         <addFlows> addACLEntries </addFlows>
        <deleteFlows> deleteFlows <\deleteFlows>
      </configure>

      <hardReset> Init addACLEntries  Bottom</configReset>

      <softReset> Resequence ResequenceBottom </configResequence>

      <configure>

  </OPERATION>

</ACL_Extended_Default>
```

Configuring FNAL-UNL Dynamic circuit: ACL Rule USCMS-UNL-INBOUND

Flows data
Model tells where it will come from




```
no ip access-list extended UNL-IN
ip access-list extended UNL-IN
ip access-list extended UNL-IN
 10 permit ip 129.93.239.0 0.0.0.255 131.225.204.0
0.0.3.255
 20 permit ip 129.93.239.0 0.0.0.255 131.225.160.0
0.0.0.255
 30 permit ip 129.93.239.0 0.0.0.255 131.225.188.0
0.0.3.255
 40 permit ip 129.93.239.0 0.0.0.255 131.225.184.0
0.0.3.255
 55 permit ip host 198.151.133.25 131.225.204.0 0.0.3.255
 61 permit ip host 198.151.133.25 131.225.184.0 0.0.3.255
 70 permit ip host 198.151.133.25 131.225.188.0 0.0.3.255
 71 permit ip host 198.151.133.25 131.225.160.0 0.0.0.255
 80 permit ip host 129.93.6.22 131.225.204.0 0.0.3.255
 91 permit ip host 129.93.6.22 131.225.184.0 0.0.3.255
 100 permit ip host 129.93.6.22 131.225.188.0 0.0.3.255
 101 permit ip host 129.93.6.22 131.225.160.0 0.0.0.255
 2147483647 deny ip any any
```

INIT Template:

AddACLEntries Template:

```
no ip access-list extended {ACLNAME}
ip access-list exended {ACLNAME}
```

Bottom Template:

```
ip access-list exended {ACLNAME}
{index} permit ip {srcflows} {srcOps} {dstFlows} {dstOps} DSCP {DSCP}
```

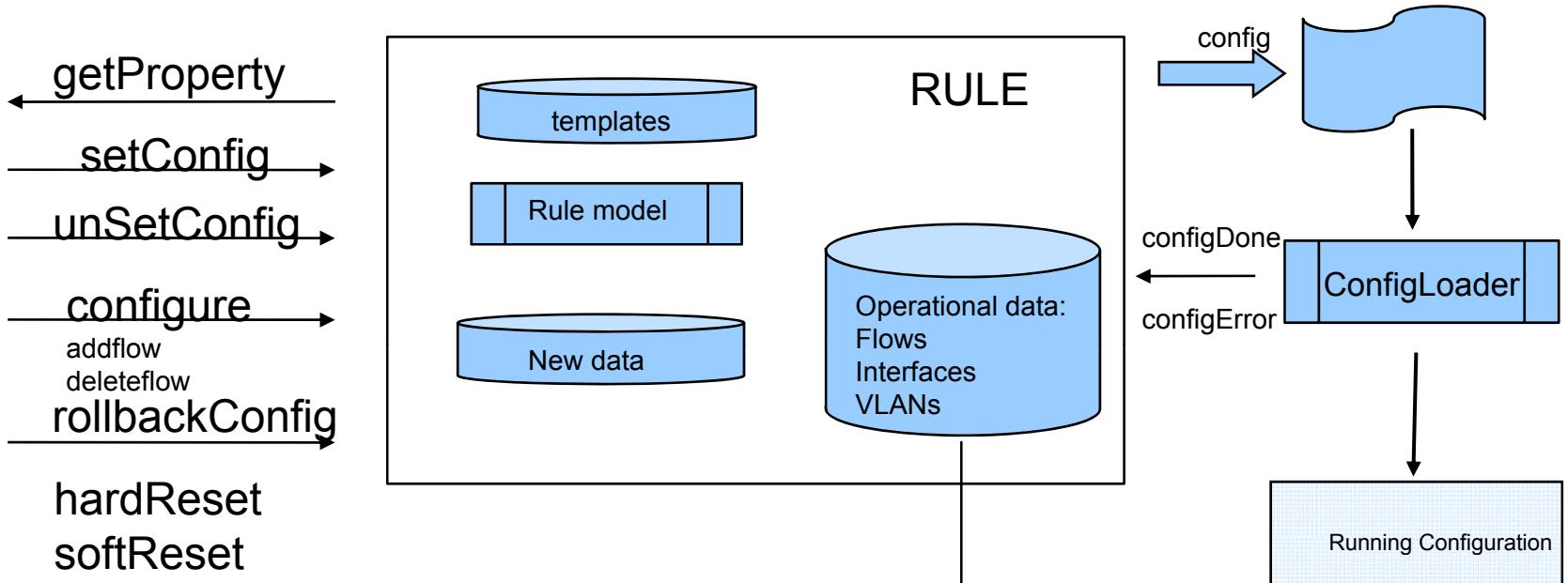
```
ip access-list exended {ACLNAME}
2147483647 deny ip any any
```

- Indexes are generated by rule enginee
- Duplicated strings will be removed by Netconfig loader

Example of Virtual Path Definition

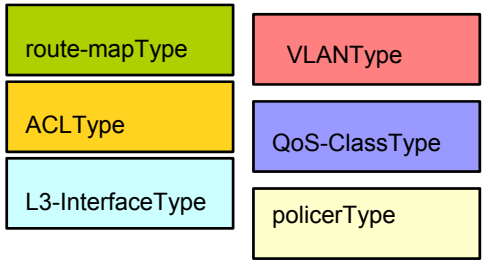
```
<virtualPath xmlns="">
    <pathName>fnal-unl-vpath</pathName>
    <localEntity>
      <name>USCMS-T1</name>
    </localEntity>
    <ruleByName>unl-l3-interface
      <ruleDeployment><device>r-s-starlight-fnal</device></ruleDeployment>
    </ruleByName>
    <ruleByName>fnal-unl-dynamic-in
      <ruleDeployment>
        <attach>

          <!-- attach means the rule should exists or deployed first -->
          unl-l3-interface
          <attachName>inboundPolicy</attachName>
        </attach>
        <device>r-s-starlight-fnal</device>
      </ruleDeployment>
    </ruleByName>

    <ruleByName>e2e-lhcopn-circuits-out</ruleByName>

    <circuitByName>fnal-unl-dynamic-circuit
    </circuitByName>
    <remoteEntity>UNL-T2</remoteEntity>
  </virtualPath>
```

Architecture of a Simple Type Rule



Rule's model describes configuration fragments created by this rule, their type, properties and operations.

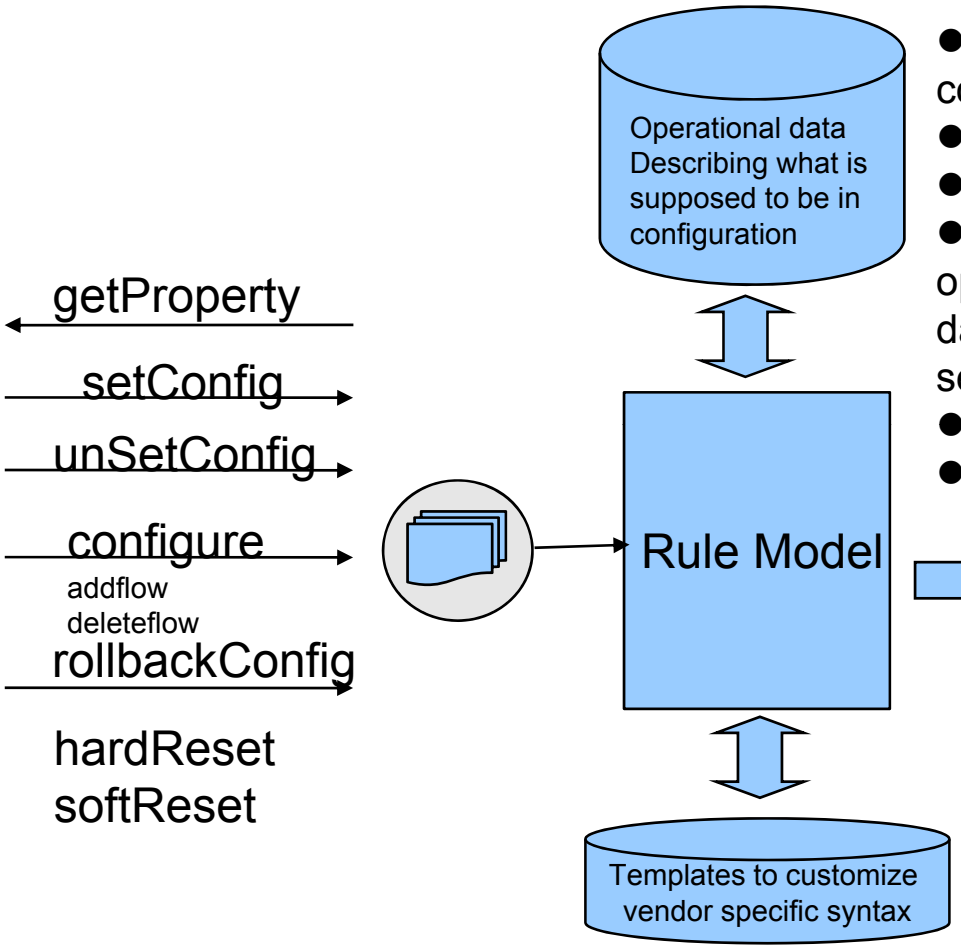
Examples of rules



Primitive configuration elements/rules

- ACL – most widely used in various techniques
- L2-VLAN
- L3-Interface
- QoS Class
- QoS Policer
- route-map

Unified Rule Interface

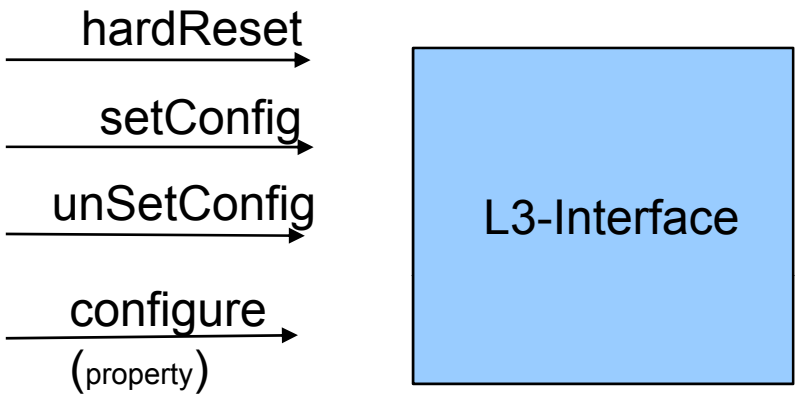


- setConfig – create fragment of new configuration
- unSetConfig – remove configuration fragmen
- configure – configure selected rule's property
- hardReset – recreate configuration with all operational
data removed and then do unSetConfig/ setConfig
- softReset – remove all operational data
- rollback – undo the last change

A fragment of configuration to upload in devices

Properties: (e.g. IP address, netmask,FlowsToAdd, FlowsToDelete, Status)

L3-Interface Rule



Properties:
- if-name
- IP Address
- netmask
- mtu
- status:
- up/down/synchronized

(in the running config)

- setConfig - create new interface
- unSetConfig – remove interface
- configure – (attach, detatch)
- rollback – undo recent change
- reset – no operation

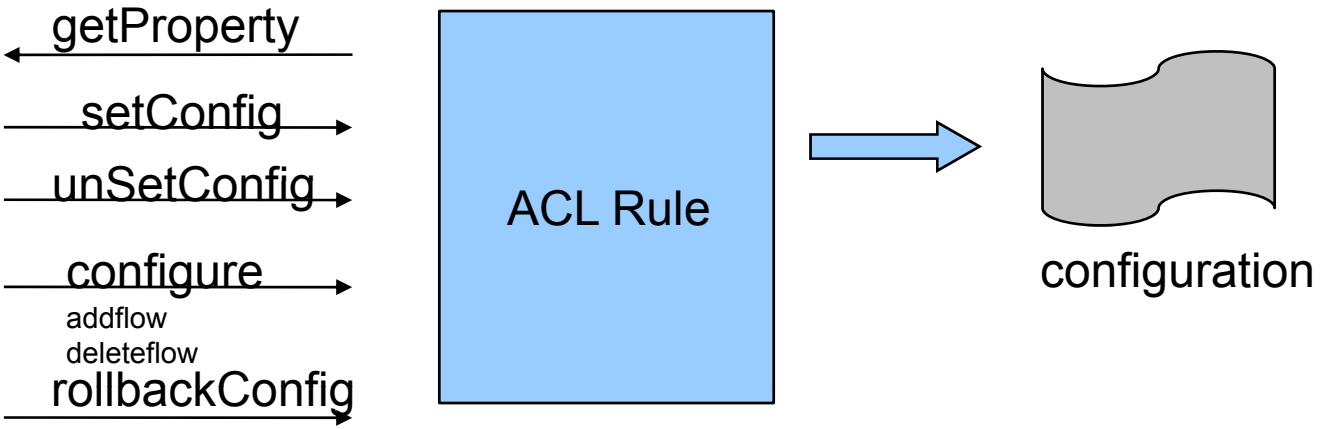
setConfig:
interface Vlan811
ip address 198.168.1.1 255.255.255.0
MTU 9216
no shutdown

unSetConfig:

interface Vlan811
shutdown
no interface Vlan811

configure:
interface Vlan811
MTU 9216

Sequencing ACL Rule



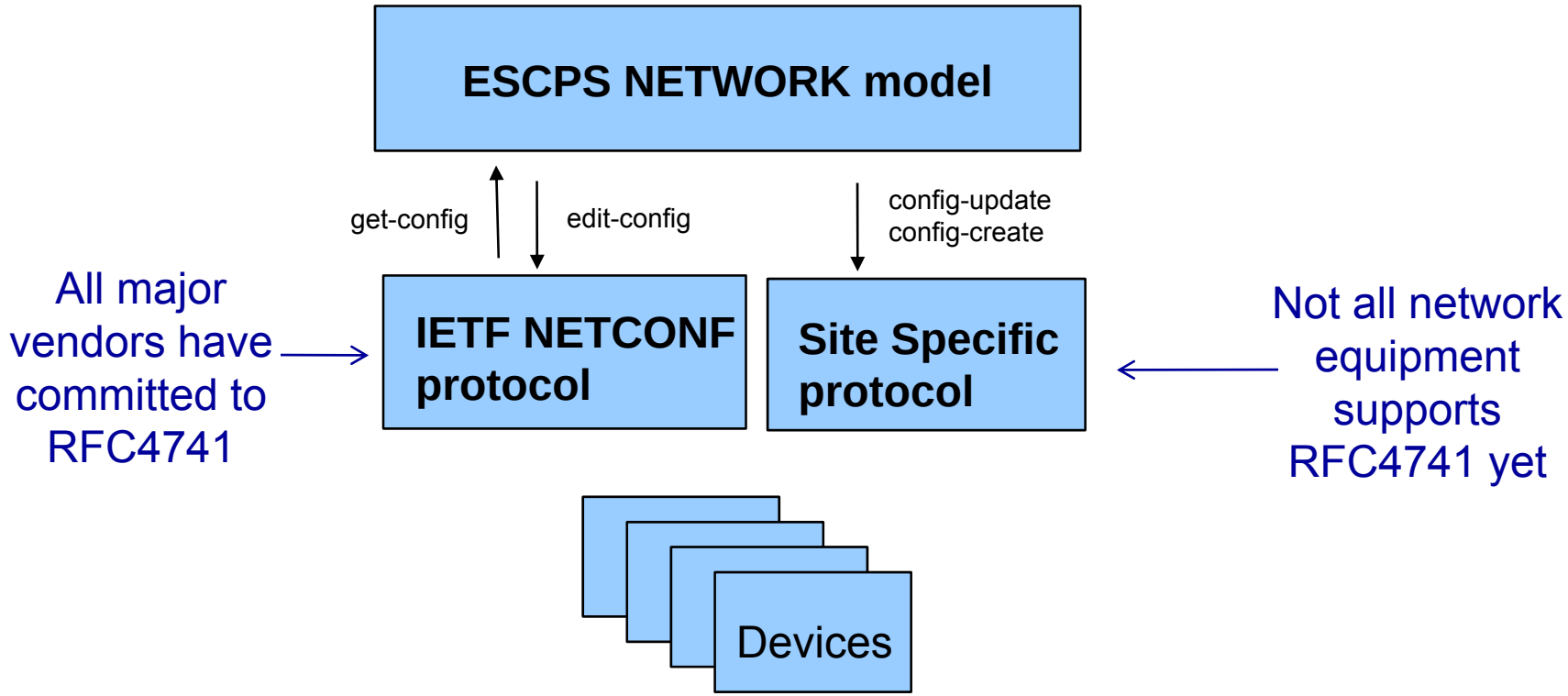
Property:
- Status (synchronize with running config)
- ACLName
- addedFlows
- deletedFlows

- setConfig – create a new ACL
- unSetConfig remote ACL
- configure – add,delete flows
- HardReset – delete all operational data,
- then unset/set
- softReset: delete all operational data,
- then resync
- rollback – undo recent change
  for addedflows – delete
  for deleted flows – do nothing,

Definitions

- A RULE-based network model is an abstract representation of site network in terms of dynamic configuration tasks to establish a virtual path with desired characteristics for selected aggregated flows. This model defines primitive building configuration blocks, defines unified interface and service to synchronize model and actual configurations

ESCPS NM & NETCONF Protocol (RFC4741)



ESCPS Resource Data Model

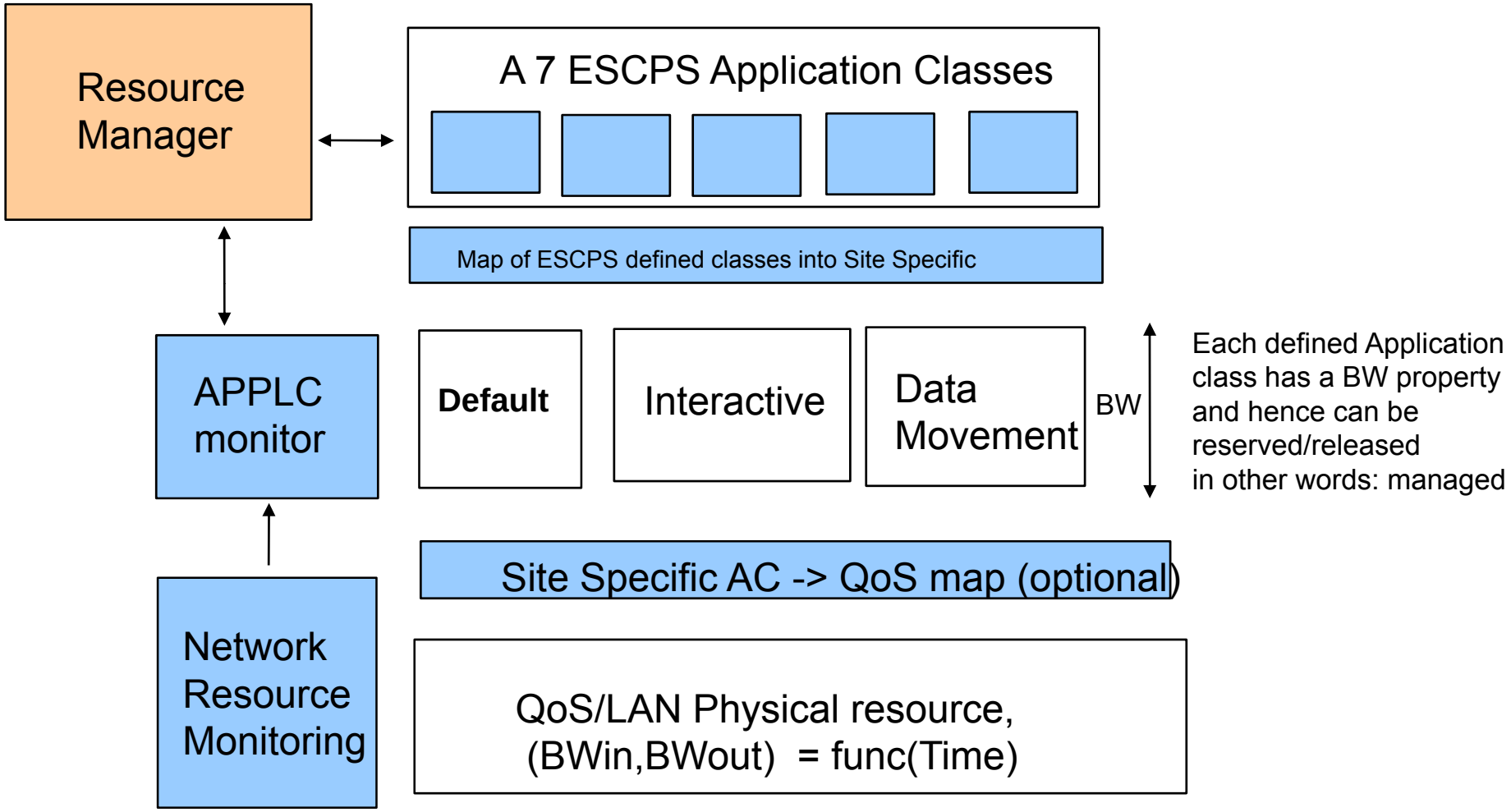
- ESCPS Resource is a bandwidth associated with physical LAN resources such as physical links or BW associated with specific QoS classes. Other characteristics, such as jitter, RTT and OWD, are taken into consideration when creating various internal maps but it cannot be sold. For example, I can say: “I will give you 2Gbps of BW but I cannot say: “I will give you 50ms jitter“. Or can we ?
- Let’s define 5 Application Classes (differentiated by different traffic characteristics and hence could be mapped into different QoS classes)

1. Interactive
2. Transcational
3. Real-Time
4. Streaming
5. Data Movement
6. Bulk Scanvenger
7. Default

Each Site is required:
- define at least one application class (e.g. Default)
- If more than one but less than 5 application classes are defined, create MAP for remaining classes not defined locally into defined ones. This map will be used to accept or deny requests for BW coming from remote sites.

Each Site will need to assign bandwidth to each application class and create mapping into physical resource. This map will be used to monitor actual resource utilization

ESCPS Resource Data Model (cont)



AddACLEntries Template

```
! addACLEntries Template for ACL_Extended_Default
!
{
$OUT .= "ip access-list extended $ACLNAME\n";
#print "indexList", join(',',$indexList);
  my $src = 'from';
  my $dst = 'to';
  if ($DIRECTION && ($DIRECTION eq 'in')) {
    $src = 'to';
    $dst = 'from';
   }

foreach my $index (@indexList) {
   next if (!$DATA{$index});
   my $proto           = ($DATA{$index}{protocol})?
$DATA{$index}{protocol} : 'ip';
   my $command         = ($DATA{$index}{command})?
$DATA{$index}{command}   : 'permit';
   my $srcBase         = $DATA{$index}{$src}{ipaddress};
   my $srcWild         = $DATA{$index}{$src}{hostmask};
   my $srcOperator     = $DATA{$index}{$src}{operator};
   my $srcPort1        = $DATA{$index}{$src}{port1};
   my $srcPort2        = $DATA{$index}{$src}{port2};

   my $srcOps          = ($srcOperator && $srcPort1)? "$srcOperator
$srcPort1" : undef;
      $srcOps         .= " $srcPort2"  if ($srcOperator && ($srcOperator eq
'range') && $srcPort2);

   my $dstBase         = $DATA{$index}{$dst}{ipaddress};
   my $dstWild         = $DATA{$index}{$dst}{hostmask};
   my $dstOperator     = $DATA{$index}{$dst}{operator};
   my $dstPort1        = $DATA{$index}{$dst}{port1};
   my $dstPort2        = $DATA{$index}{$dst}{port2};
!
 my $dstPort2        = $DATA{$index}{$dst}{port2};

   my $dstOps          = ($dstOperator && $dstPort1)?
"$dstOperator $dstPort1" : undef;
      $dstOps         .= " $dstPort2"  if ($dstOperator &&
($dstOperator eq 'range') && $dstPort2);

   my $established     = ($DATA{$index}{established})?
'established' : undef;
   my $precedence      = ($DATA{$index}{precedence})?
"precedence $DATA{$index}{precedence}" : undef;
   my $DSCP            = $DATA{$index}{dscp};
   my $time            = ($DATA{$index}{'time-range'})?
$DATA{$index}{'time-range'} : undef;
   my $fragments       = ($DATA{$index}{'fragments'})?
'fragments' : undef;
#
  $OUT  .= "$index $command $proto $srcBase $srcWild";
  $OUT  .= " $srcOps" if ($srcOps);
  $OUT  .= " $dstBase $dstWild";
  $OUT  .= " $dstOps" if ($dstOps);
  $OUT  .= " $established" if ($established);
  $OUT  .= " $precedence"  if ($precedence);
  $OUT  .= " dscp $DSCP"   if (defined $DSCP);
  $OUT  .=  " time-range $time" if ($time);
  $OUT  .=  ($fragments)? " fragments\n" : "\n";

 }
}
```

XML Model example FNAL-UNL-T2-DYNAMIC

```
<?xml version="1.0" encoding="UTF-8"?>
<NetworkModel xmlns="http://www.escps.org/net"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation=
"http://www.escps.org/net
file:/Users/bobyshev/mysvn/escps/xml/ESCPSNetworkModel.xsd">
   <!-- This is an example of dynamic circuit that needs the following
      A hypotetical example UNL-T2-DYNAMIC
      1. Demand l2-circit via ESNet/OSCARs
      2. Create vlan L3 interface dynamically at StarLight Termination point
      3. Apply circuit inbound policy on that interface
      4. Manage an ACL on inbound interface to the termination point
   -->
   <Definition xmlns="">
      <flowEntity>
         <name>T2-UNL</name>
         <type> ESCPS
         </type>
        <escpsURL>https://escps.unl/edu/escps/ </escpsURL>
         <!-- All required information will be obtained dynamically
         -->
      </flowEntity>
   </Definition>
   <Definition xmlns="">
      <circuit>
         <escpsCircuitName>fnal-unl-dynamic-circuit</escpsCircuitName>
         <resourceByName>fnal-unl-10G-resource</resourceByName>
         <vlan>300 - 400 </vlan>
         <escpsAddress>escps.fnal.gov</escpsAddress>
         <dcnProvider>
            <!-- Call defined URL of ESNET's IDCs -->
            ESNET
         </dcnProvider>
         <!-- Dynamic circuit -->
         <l3>https://escps.fnal.gov/escps/idc/interface/fnal-unl-dynamic-circuit/</l3>
      </circuit>
</Definition>
   <virtualPath xmlns="">
      <pathName>fnal-unl-vpath</pathName>
      <localEntity>
         <name>USCMS-T1</name>
      </localEntity>
      <ruleByName>unl-l3-interface
         <ruleDeployment><device>r-s-starlight-fnal</device></ruleDeployment>
      </ruleByName>
      <ruleByName>fnal-unl-dynamic-in
         <ruleDeployment>
            <attach>
               <!-- attach means the rule should exists or deployed first -->
               unl-l3-interface
               <attachName>inboundPolicy</attachName>
            </attach>
            <device>r-s-starlight-fnal</device>
         </ruleDeployment>
      </ruleByName>

         <ruleByName>e2e-lhcopn-circuits-out</ruleByName>
      <circuitByName>fnal-unl-dynamic-circuit
      </circuitByName>
      <remoteEntity>UNL-T2</remoteEntity>
   </virtualPath>

   <Definition xmlns="">
      <rule>
         <ruleName>unl-l3-interface</ruleName>
         <ruleType>l3-interface</ruleType>

         <!-- L3 Interface is associated with circuit
            If it is dynamic, "idc" service should provide it
            However, IDC only determines/negotiates it with remote site
            but does not configure.
            To configure it is a LDC job
         →
<L3-interface>https://escps.fnal.gov/idc/circuit/fnal-unl-dynamic-circuit/interface</L3-interface>
         <IPAddress>https://escps.fnal.gov/idc/circuit/fnal-unl-dynamic-circuit/interface/ipaddress</IPAddress>
         <Netmask>https://escps.fnal.gov/idc/circuit/fnal-unl-dynamic-circuit/interface/netmask</Netmask>
      </rule>

      </Definition>
</NetworkModel>
```

Summary

- A rule-based ESCPS network model is aimed to describe a site network in terms of configuration tasks and primitive configuration elements rather than describe network topology in terms of traditional network elements such as routers, switches, links, ports or interfaces. In rule-based model these elements could be parameters of configuration elements.

Summary: Outcome of ESCPS/LDC development

- Conceptual ESCPS/LDC model of site network (XML)
- software package to create ESCPS/LDC RESTFul Web Services
- Administrator's GUI software and tools to create a site model, manage and monitor whole system
- A library of primitive building blocks for several network technologies (PBR, QoS, VRF-Lite , MPLS (???)
- A library of templates to configure devices of most popular network vendors (focusing on ones are being used at developer's sites)

What Next....

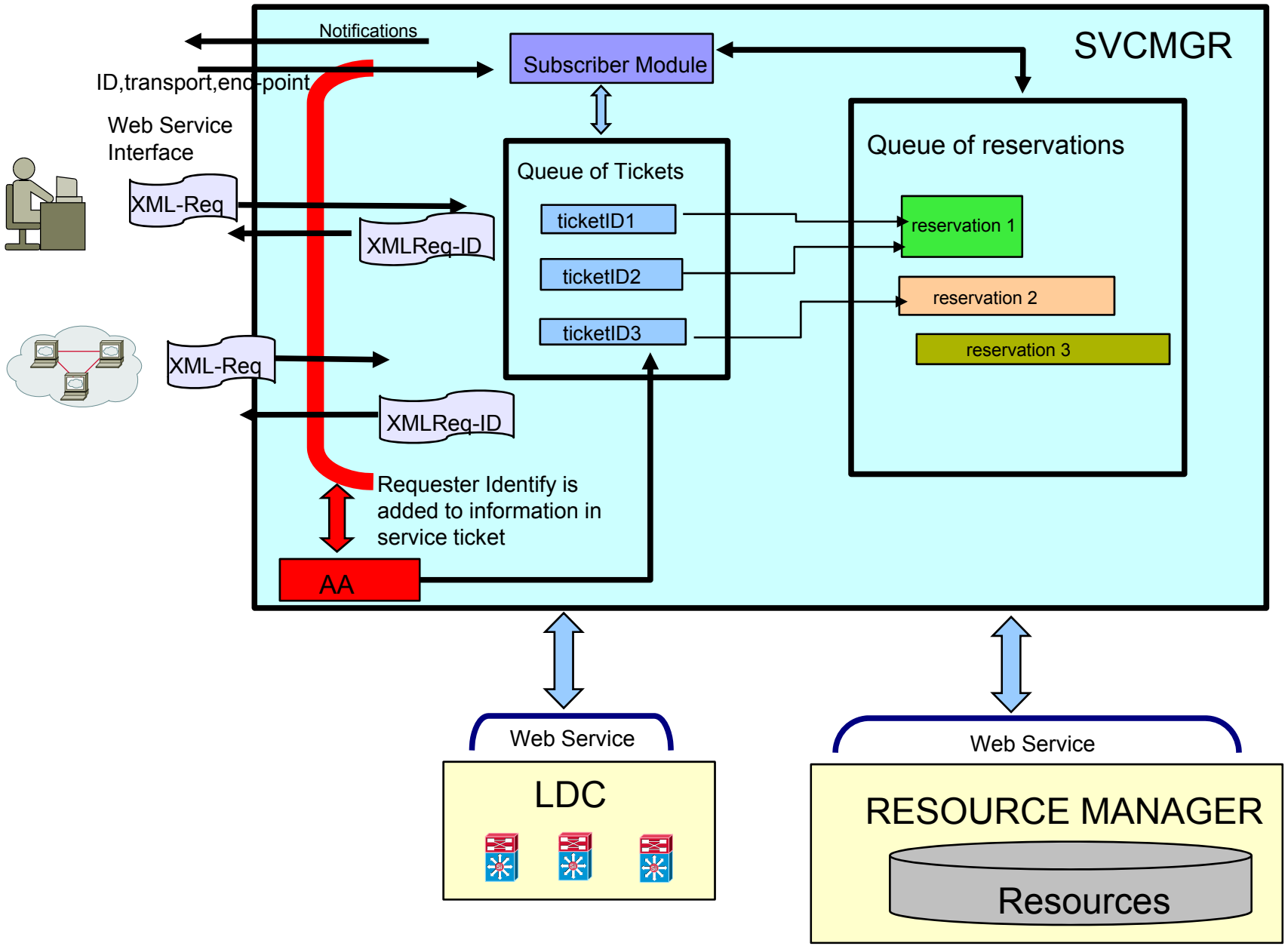
- Alpha/beta phase of code(s)
- More primitive configuration elements
  - MPLS building blocks
  - VRF-Lite building blocks and etc…
- Multi-vendor adaption
- Contact more other sites and describe their network in terms of this model
- Administrator's GUI software and tools to create a site model, manage and monitor whole system

ESCPS SVCMGR

Andrey Bobyshev
10/27/2010

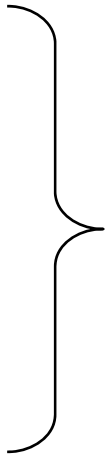
Outline

- *Architecture* of SVCMGR
- SVCMGR Functionality
- RESTFul Service


Notifications
ID,transport,end-point
Web Service
Interface
XML-Req
XMLReq-ID
XML-Req
XMLReq-ID
Requester Identify is
added to information in
service ticket
AA
SVCMGR
Subscriber Module
Queue of Tickets
ticketID1
ticketID2
ticketID3
Queue of reservations
reservation 1
reservation 2
reservation 3
Web Service
LDC
Web Service
RESOURCE MANAGER
Resources

Service Manager Functionality

- Create Service Ticket - request for a service
- Get Ticket Status (identifier) - return status of Service Ticket
- Cancel Ticket (identifier) – cancels the Ticket
- List Tickets (searchCriteria)
- listReservations(searchCriteria)
- Get Ticket Info (ticketID)
- getReservationInfo(reservationID)
- Modify Ticket
- ***getDCNSite - Get information about remote sites reachable via DC networks***
- ***getAFEEInfo - provide information about available AFEs at local and remote sites based on various search criterion***
- ***AFEELookup***
- ***listCircuits***
- ***getCircuitInfo***



Relay IDC services

RESTFul Web services

REST-style architectural consists of clients and servers in context of original HTTP specifications. It is a collection of resources with three defined aspects:

- base URI for Web services
- the MIME type of data supported
- the set of operations by the web servces using HTTP methods (POST,GET, PUT or DELETE)

OpenSvcTicket: Initiate setup of specified services for selected aggregated flows

URI: $BASE_URI/requestManager/serviceTicket
HTTP Method = POST

To get list of tickets: HTTP GET

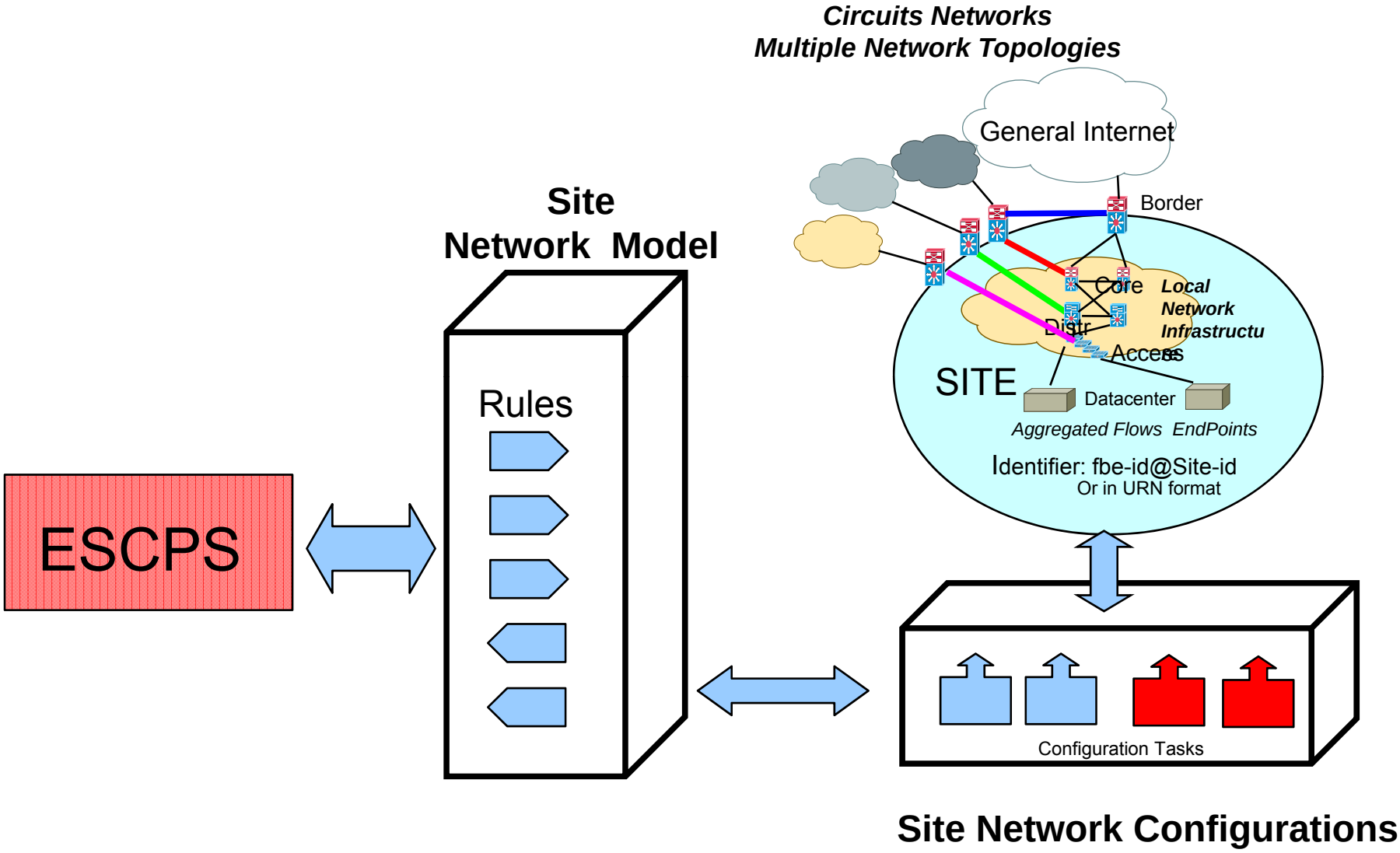
URI: $BASE_URI/requestManager/serviceTicket

SVCMGR summary

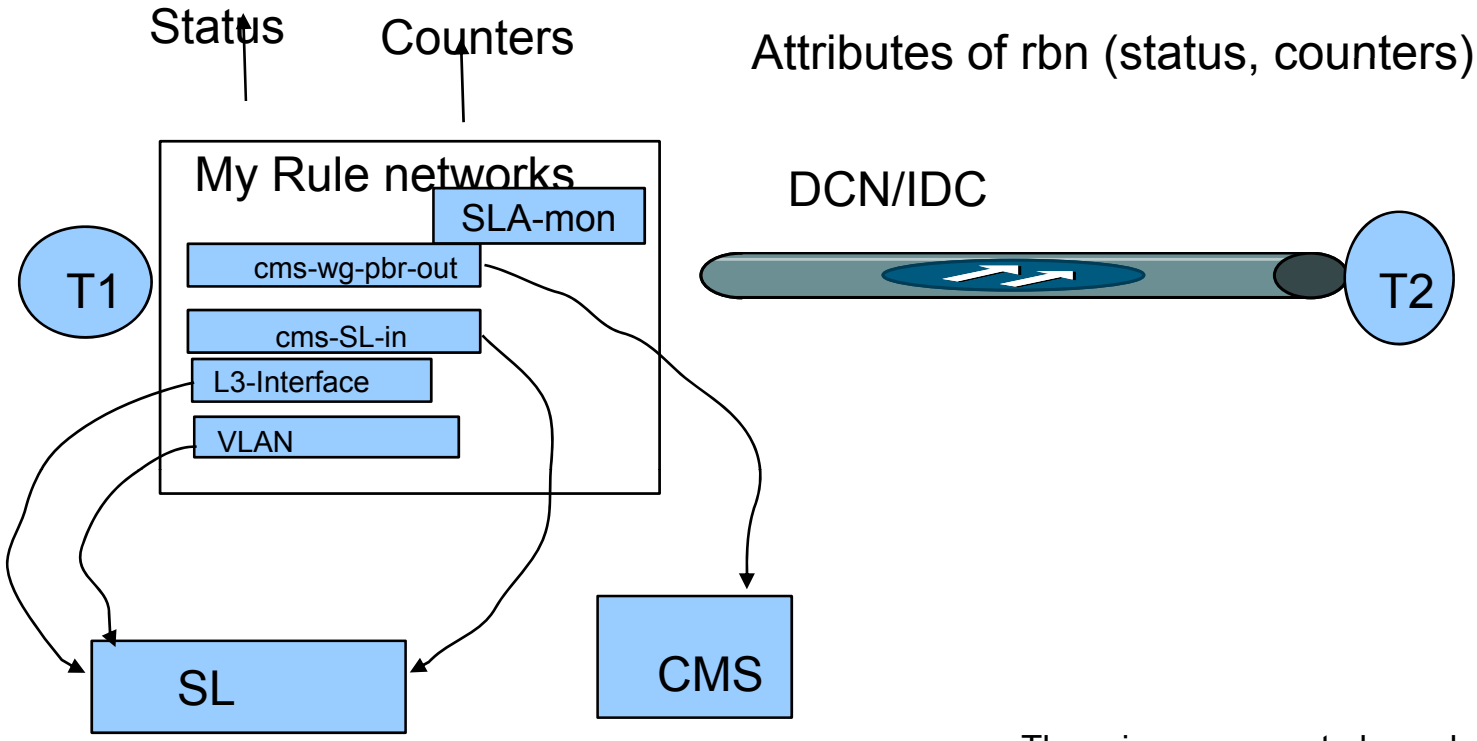
- Detailed design note completed
- Web Service interface is done
- Ticket status and progress visualization
- Software release of SVCMGR is in alpha phase
- Issues:
  - –Dependency on other components of ESCPS system is simulated
  - –Authorization and security
  - –Change of ticket's parameters in-fly
- Future directions & efforts:
  - –Integration into general ESCPS framework (w/ BNL)
  - –Potential inclusion of XPS support (w/ U-Del)

EXTRA SLIDES

A conceptual view on ESCPS Site Modeling



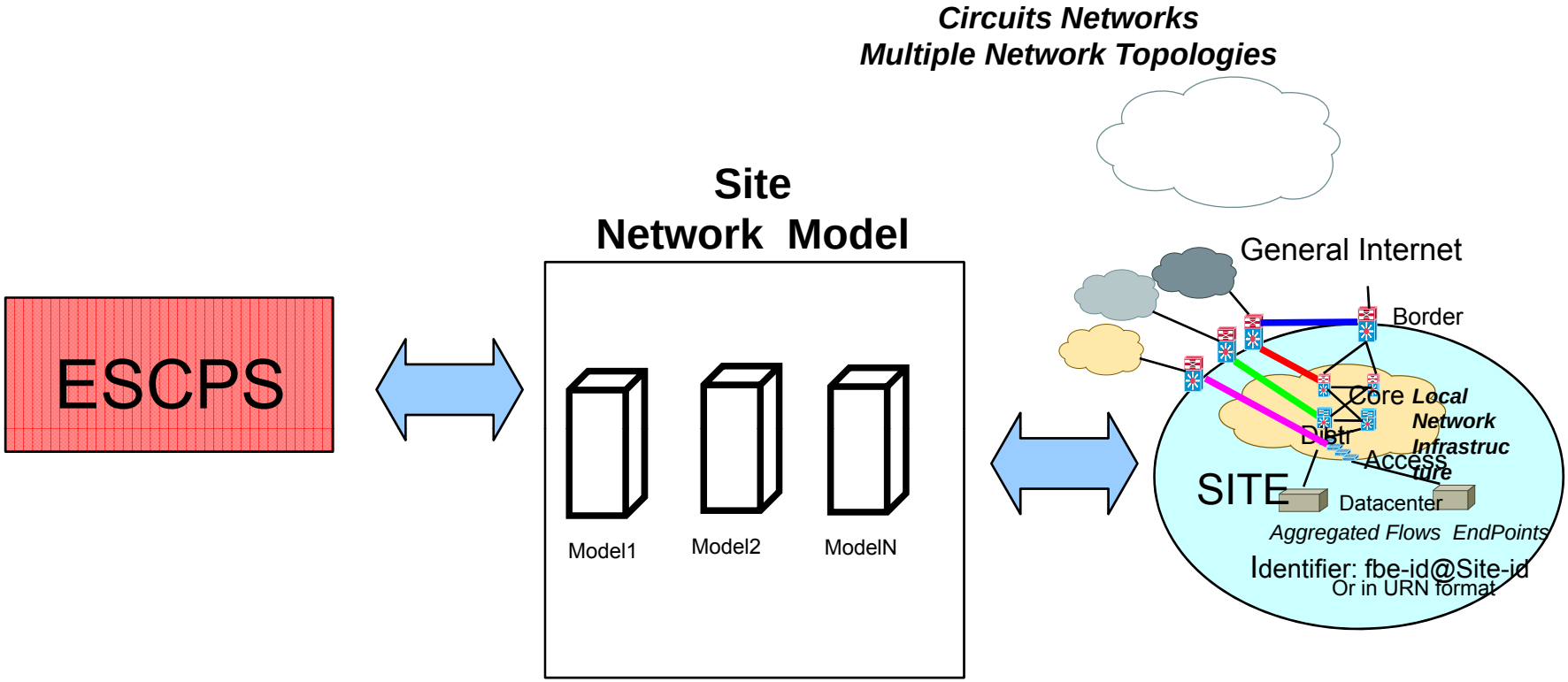
FNAL-UNL Dynamic circuit created dynamically



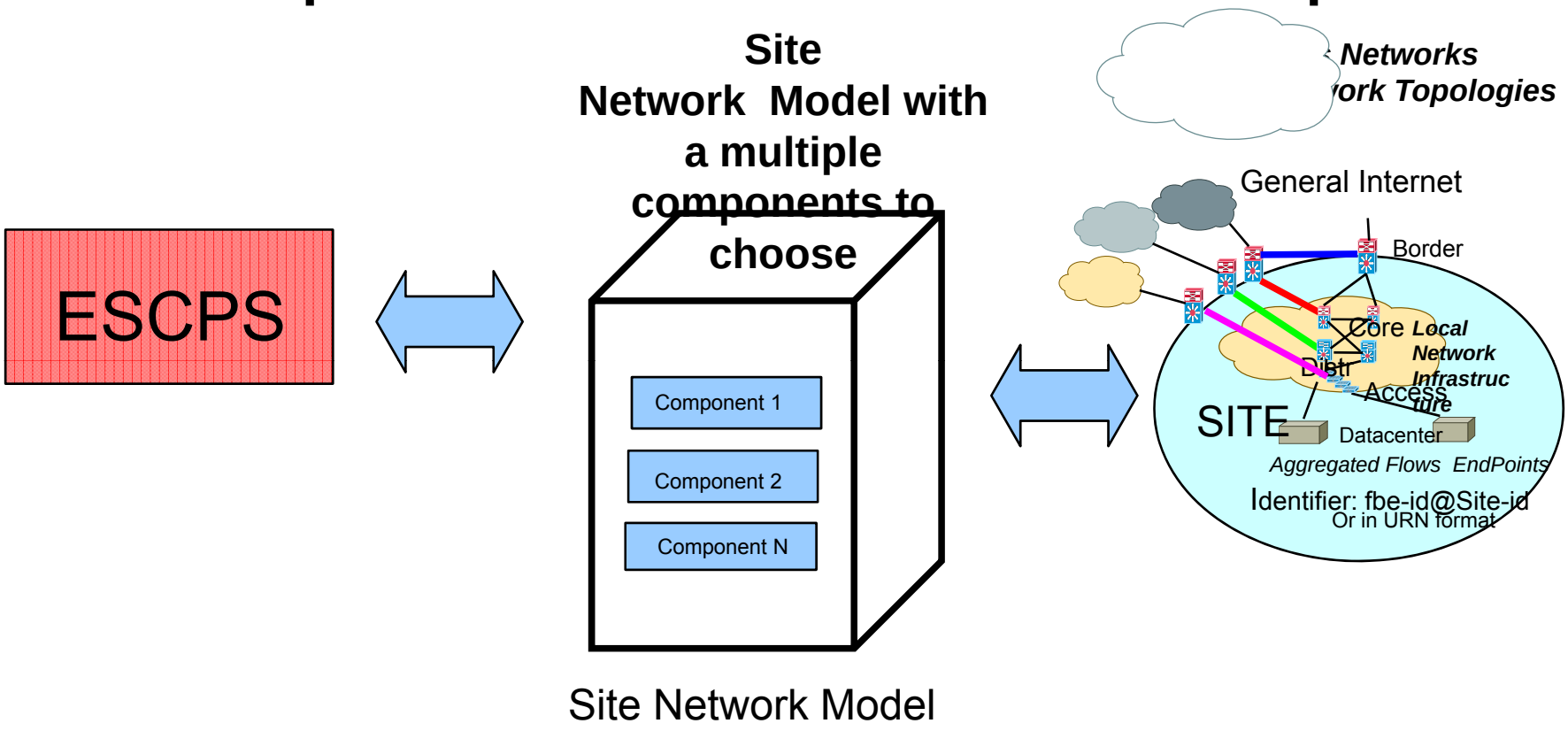
- There is no reason to know how these physical devices are connected
- No need to know what are rules on the other side

1. VLAN offer is received from IDC (accepted or denied)
2. Create L3-Interface (ESCPS/IDC job to negotiate all parameter)
3. LDC creates and applies rules

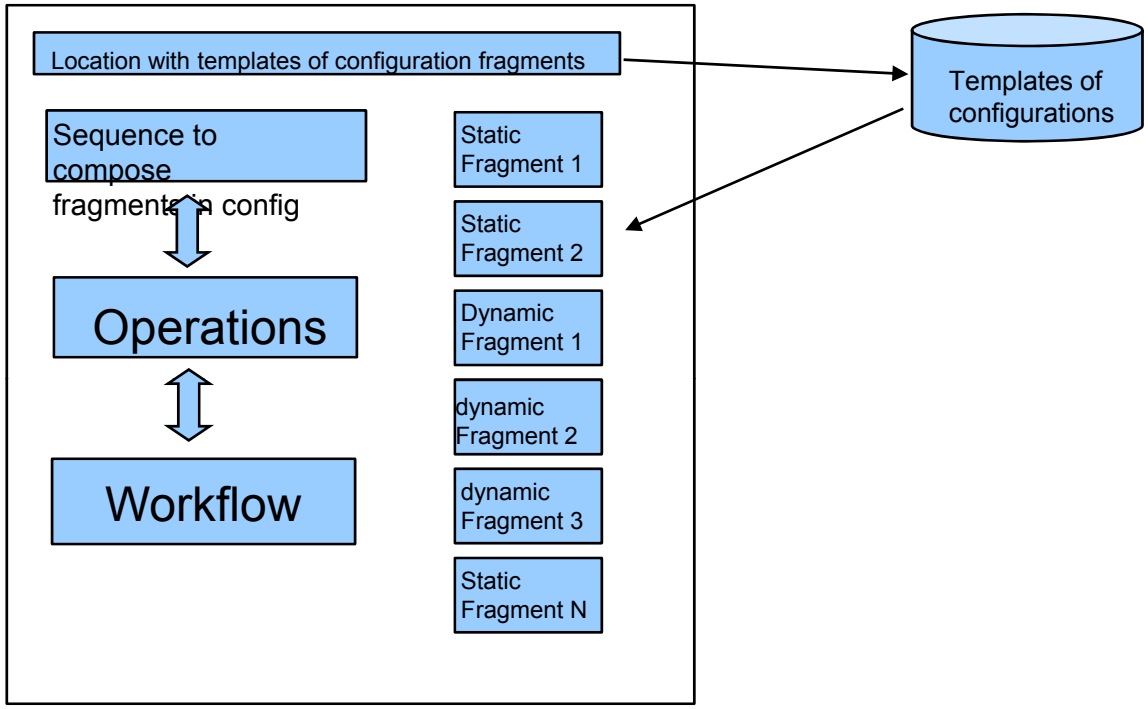
Multiple selection of network models

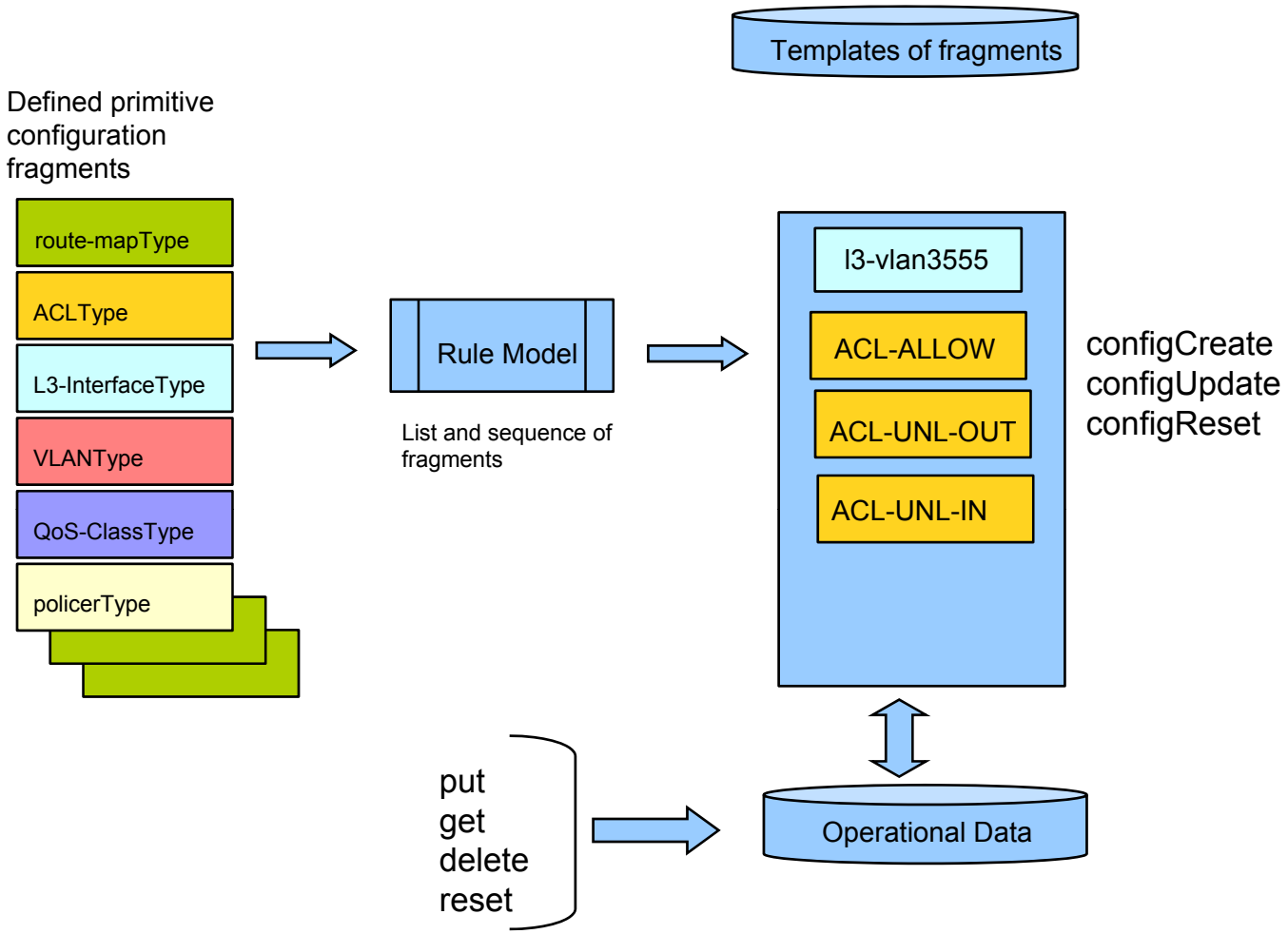


A single model with unified interaface but multiple selection of Network Model components



A rule of type simpleRuleType




Templates of fragments
Defined primitive configuration fragments
route-mapType
ACLType
L3-InterfaceType
VLANType
QoS-ClassType
policerType
Rule Model
List and sequence of fragments
l3-vlan3555
ACL-ALLOW
ACL-UNL-OUT
ACL-UNL-IN
configCreate
configUpdate
configReset
put
get
delete
reset
Operational Data

A single model with unified interaface but multiple selection of Network Model components