


ESCPS Status

Phil DeMar, Dantong Yu, Martin Swany
January, 2012

End Site Control Plane System (ESCPS)

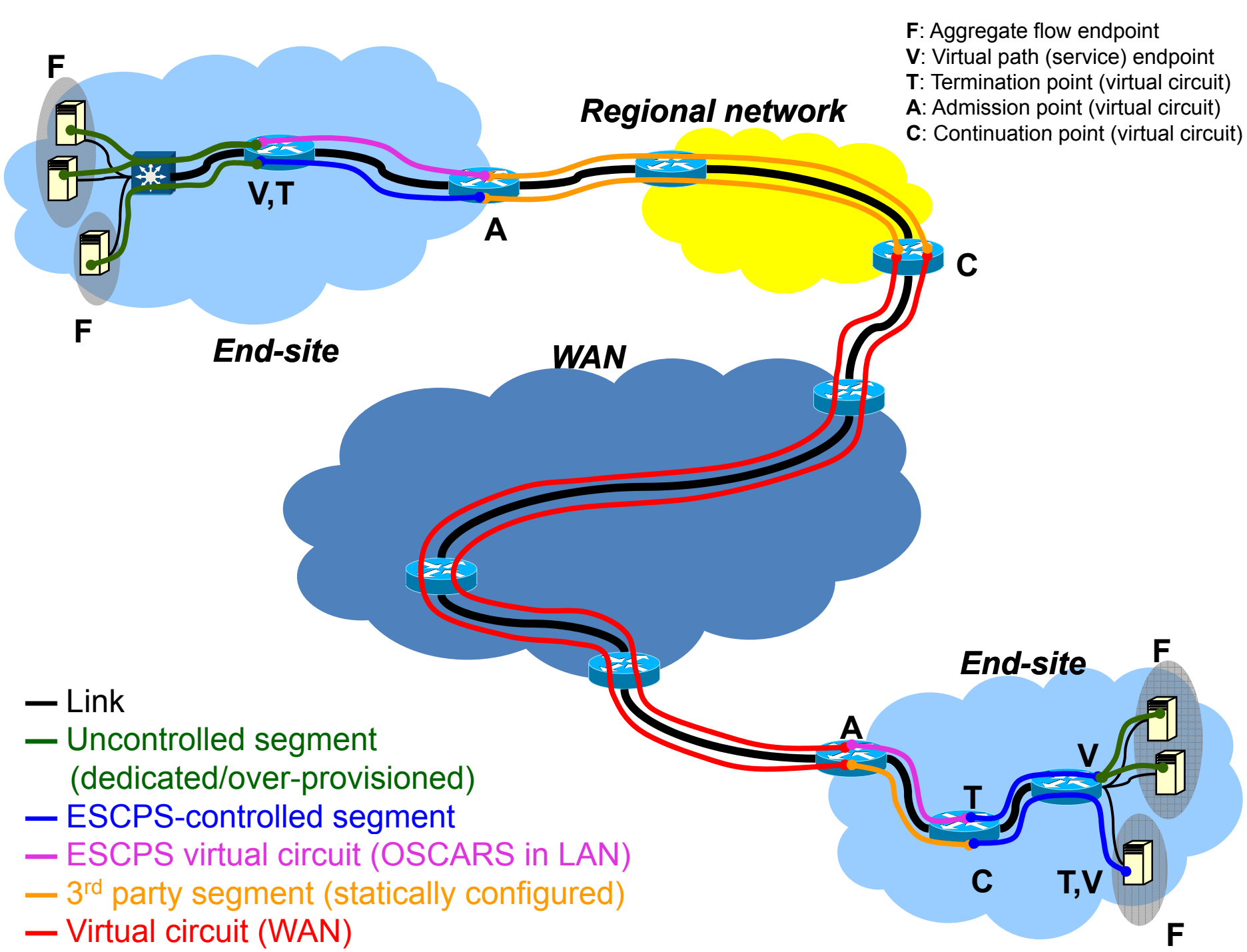
- Network service to facilitate site use of circuit services
- Accept and process user/app requests for circuit services
- Provide local interface to & coordination of WAN circuit services
- Configure local network infrastructure for use of circuits
- Monitor local network segments of end-to-end path
- Long term vision: End site component of federated control plane for circuit services

Original Project Outline Plan

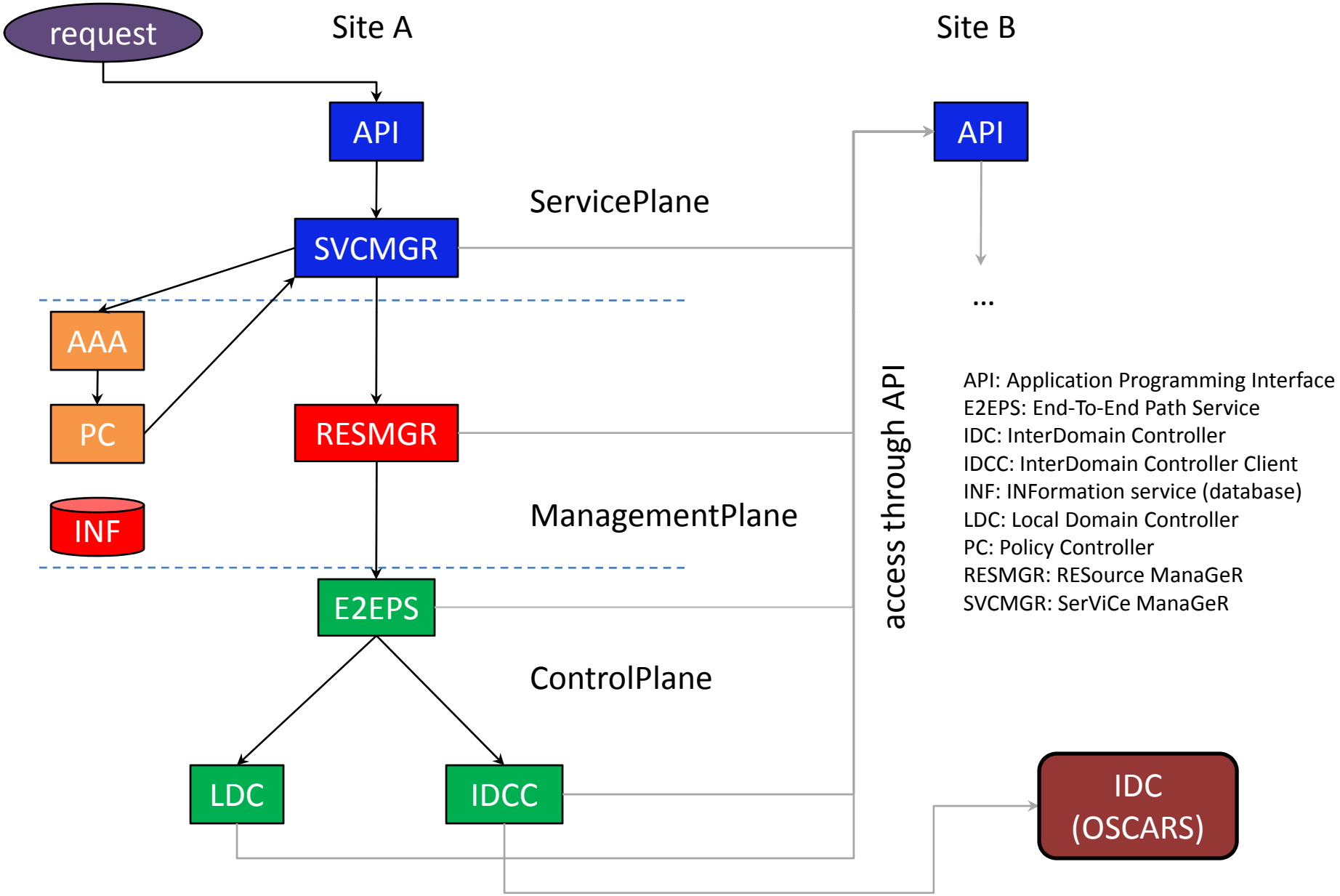
- Year 1:
  - Collaborative analysis of issues based on past experience
  - Collaborative design of system architecture and components
  - Year 1 prototype on existing testbed
- Year 2:
  - Finalization of component interfaces
  - Implementation of components as assigned to each institution
  - Deployment and component interoperability testing on expanded testbed
- Year 3:
  - Iteration and improvement on component design
  - Implementation of revisions
  - Extensive testing, debugging, and hardening of code

ESCPS Concepts

- Elements of ESCPS model
  - Aggregated Flow Endpoints (AFEs)
    - Sinks/sources for data flows; often clusters of systems
  - Circuits
    - OSCARS constructs with L2/L3 terminations
  - Virtual Paths
    - Complete end-to-end path between AFEs
  - Rules:
    - configuration units that need to be deployed to extend a circuit to become a desired virtual path


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

Initial Architecture & Components



Component Assignments

- BNL:
  - Resource manager
  - Inter-domain controller (IDC) interface
  - Coordination of integration effort
- FNAL:
  - Local domain controller (LDC)
  - Initial service manager (SVCMGR) design
- UDel:
  - Local network monitoring
  - Service manager (SVCMGR) adaptation for XSP

BNL Components of ESCPS

BNL Contributions

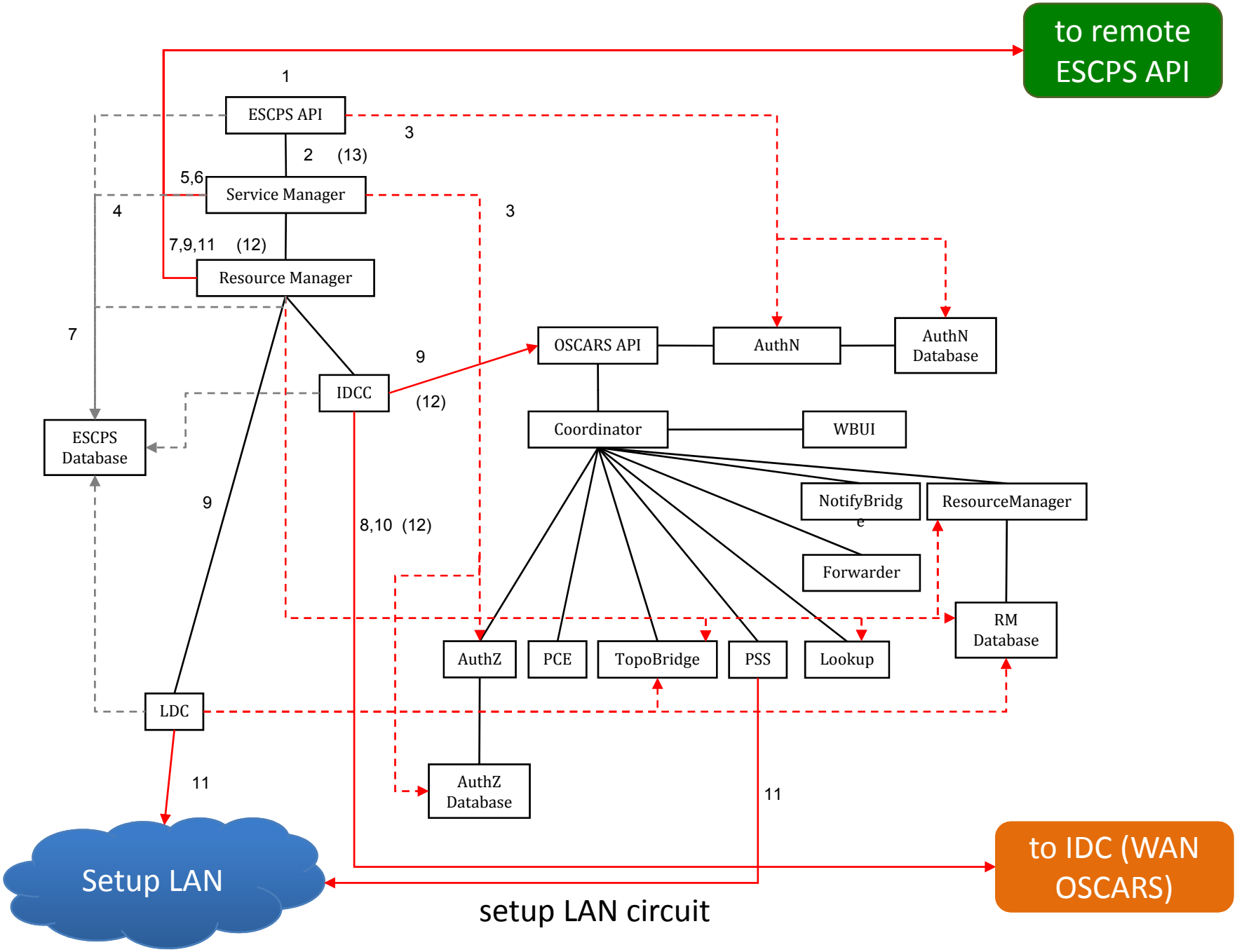
- Initial and revised design of RESMGR and IDCC
  - Components based on legacy TeraPaths code but modified to perform new RESMGR functions (in progress):
  - RESMGR coordinates multiple domain reservations to compose an end-to-end path
    - Fixed (legacy) and flexible reservation support
    - Support for trial-and-error and BAG-based negotiation (based on experience with StorNet)
    - Major database revision to support multi-segment/multi-reservation end-to-end paths

BNL Contributions

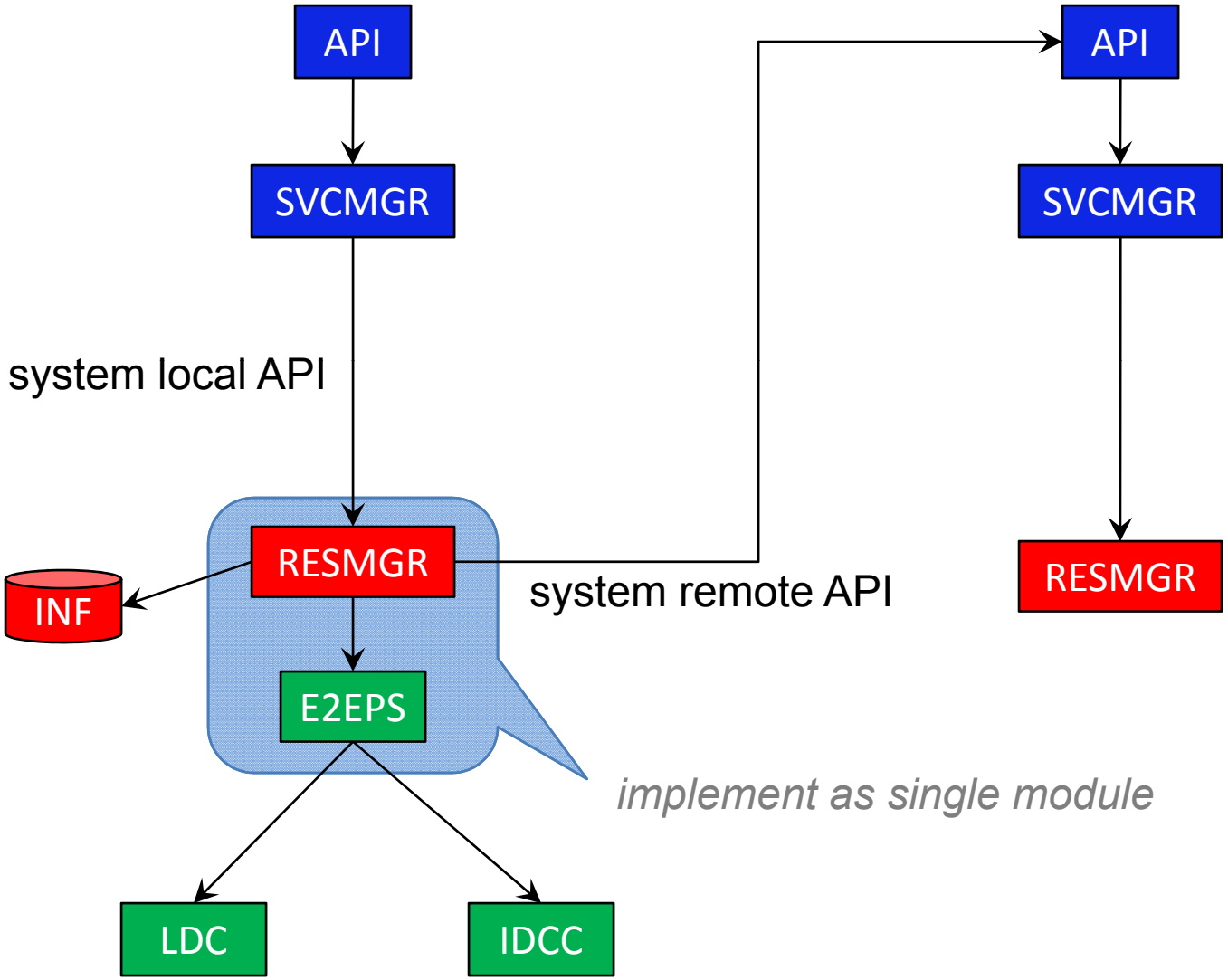
- Modifications to code base to allow reservation information display in ESCPScope
  - Pull (polling) mode through API call
  - Push mode through invocation of web-service
- Added support for circuit VLAN translation, including VLAN tag/private address allocation scheme
- Deployed ESCPScope at BNL, UMich, LBNL
- Analysis of OSCARS 0.6 architecture and code to revise ESCPS architecture to include OSCARS circuit functionality at end sites in manner consistent with LAN-specific requirements

BNL Contributions

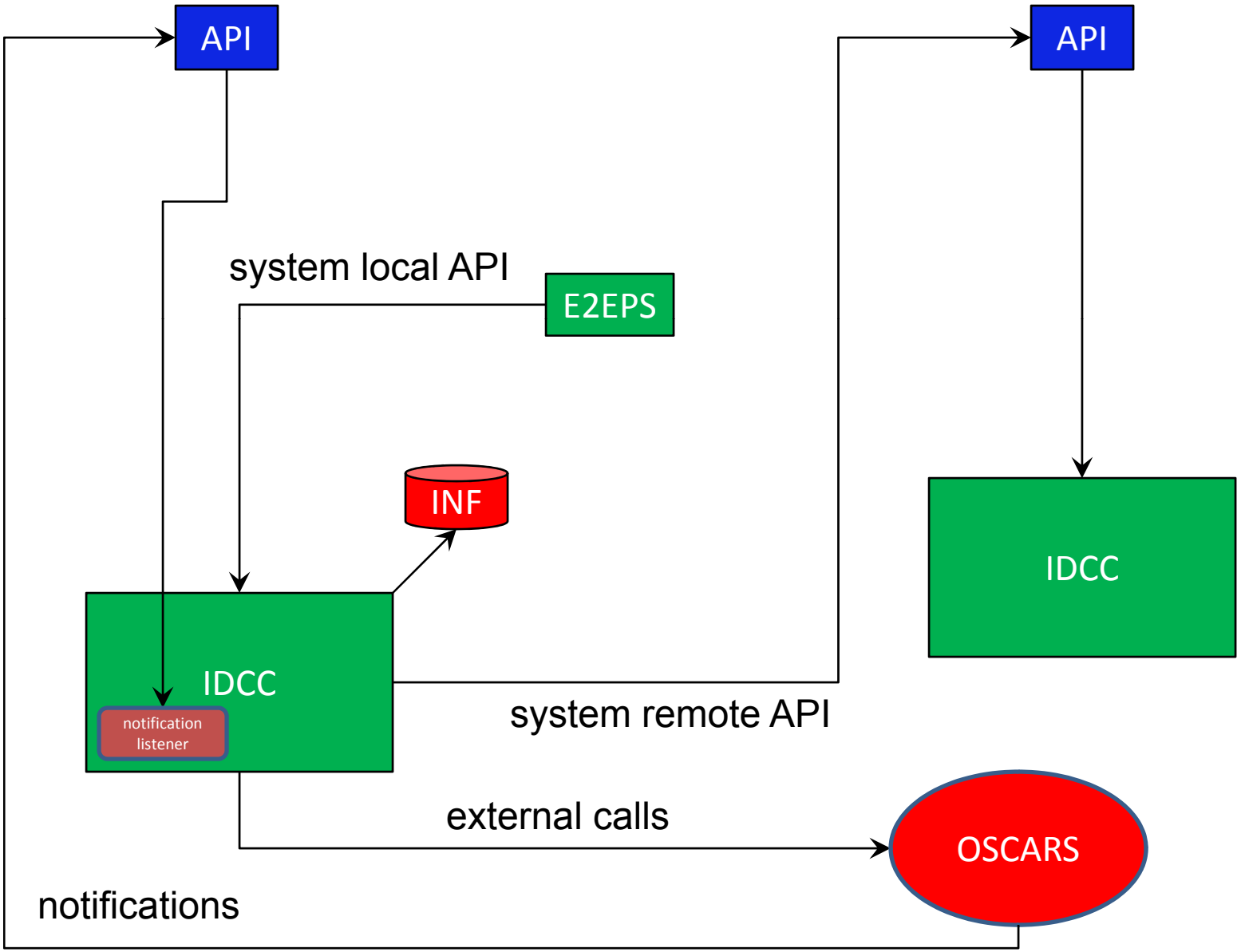
- Deployment of prototype code at UDel
- Devised short term plan to incorporate OSCARS functionality into ESCPS for LAN circuit segments


to remote ESCPS API
1
ESCPS API
3
2 (13)
5,6
Service Manager
4
3
7,9,11 (12)
Resource Manager
7
9
OSCARS API
AuthN
AuthN Database
IDCC
(12)
ESCPS Database
Coordinator
WBUI
9
8,10 (12)
NotifyBridge
ResourceManager
Forwarder
RM Database
AuthZ
PCE
TopoBridge
PSS
Lookup
LDC
11
AuthZ Database
11
Setup LAN
to IDC (WAN OSCARS)
setup LAN circuit

RESMGR Design



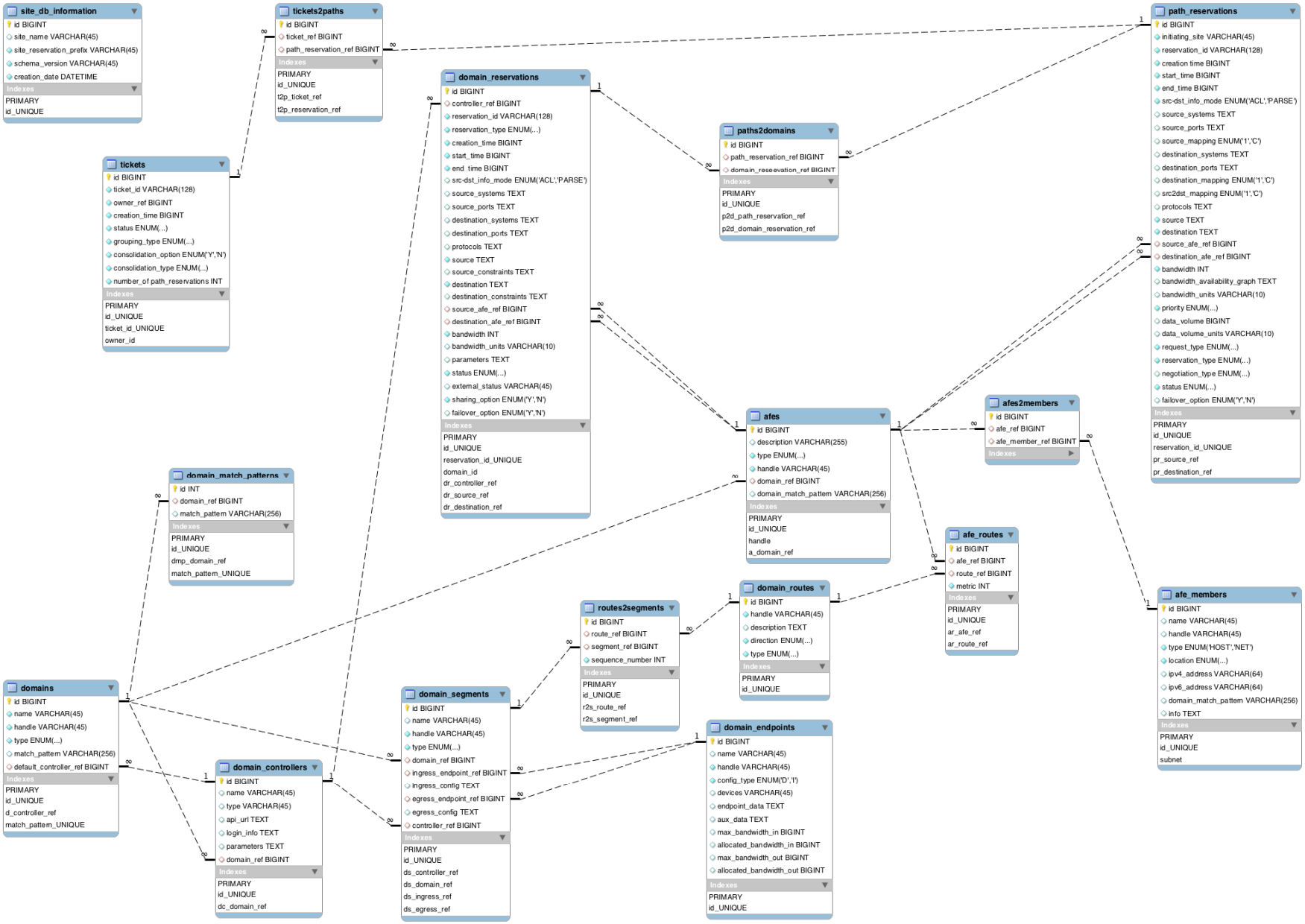
IDCC Design


API
API
system local API
E2EPS
INF
IDCC
IDCC
notification listener
system remote API
external calls
OSCARS
notifications

- Client submits request to SVCMGR via the API
  - User information
    - SVCMGR authenticates by looking up escpsdb or oscarsdb
  - Source and destination information
    - CIDR blocks, port ranges, protocols, src/dst combinations
    - SVCMGR maps onto src and dst AFEs – may need to dynamically generate AFEs (escpsdb)
  - Reservation information
    - Fixed:
      - Start time
      - End time or duration
      - Bandwidth
      - Priority class (optional, could simply use EF always)
    - Flexible:
      - Source and destination information
      - Bandwidth Allocation Graph (BAG)
        - » Earliest start time
        - » Deadline
        - » Maximum bandwidth (achievable)
      - Data volume

- RESMGR processes request
  - Looks up escpsdb with src/dst and/or invokes LDC and/or local OSCARS
    - Finds suitable routes within end-site LAN
      - DiffServ/PBR segment (endpoints)
      - Circuit segment (endpoints, full path)
  - Gathers bandwidth availability information
    - escpsdb for diffserv
    - oscarsdb for circuits of via IDCC to local OSCARS
      - Feasible with simple topologies (known routes)
      - ARCHSTONE will support availability requests
    - Primary RESMGR intersects local and remote BAGs
  - Negotiates transit circuit via IDCC to IDC
    - BAG from ARCHSTONE for alternative paths
    - Trial-and-error otherwise
  - End-to-end path consists of multiple reservations, e.g.:
    - DiffServ/PBR, L2, L2, L2, PBR/DiffServ
    - DiffServ, L3, L3, L3, DiffServ
    - L3, L3, L3 (end-to-end MPLS tunnel)
    - L2, L2, L2 (end-to-end circuit)

Database Design



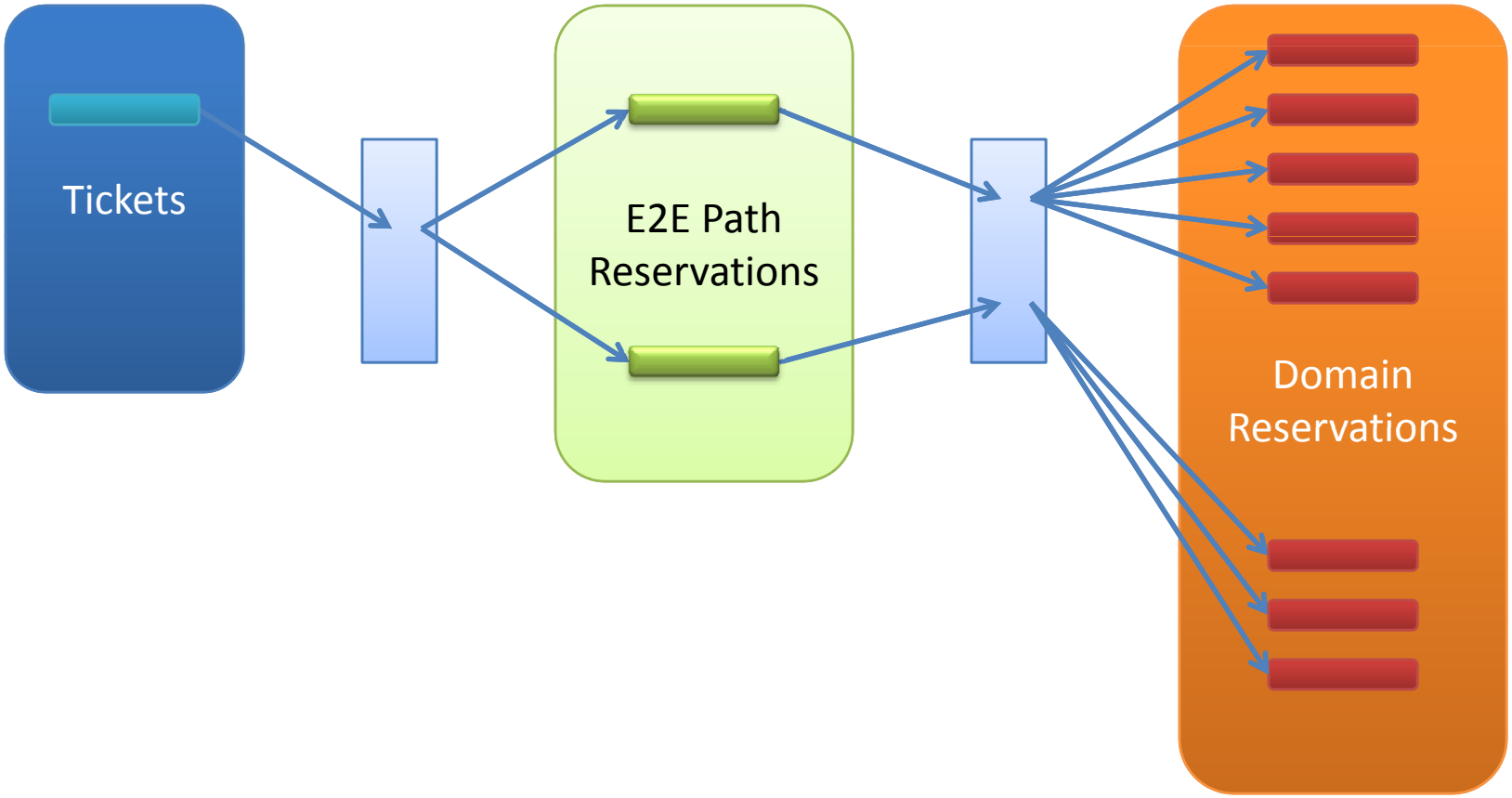
Database Design Highlights

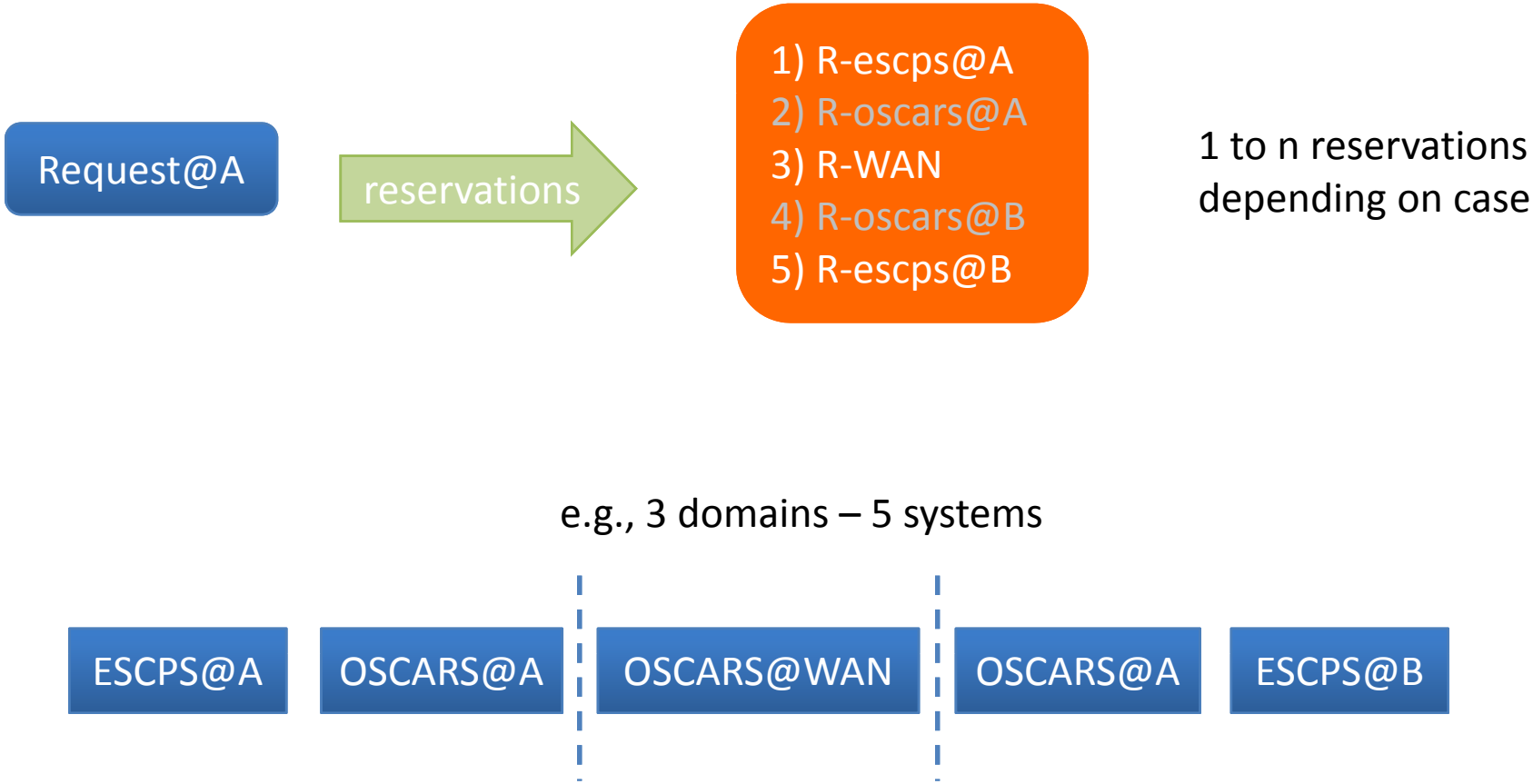
- Tickets (requests)
  - Processed by the SVCMGR
  - TicketID, src AFE, dest AFE, time window, bandwidth, etc.
- End-to-End Path Reservations
  - Link tickets to paths
- Domain Reservations
  - LAN or WAN reservation data
- Any number of domains can be included in an end-to-end path

Database Design Highlights (ii)

- Many-to-many relation between Tickets and Path Reservations
  - Standard case, one ticket is satisfied by one end-to-end path reservation
    - One ticket can include multiple end-to end path reservations with separate time frames
    - One end-to-end path reservation can serve multiple tickets (consolidation)
- Many-to-many relation between Path Reservations and Domain Reservations
  - An end-to-end path reservation consists of multiple LAN and WAN reservations
    - One transit domain reservation (OSCARS circuit) can serve multiple end-to-end reservations
- Use junction tables to convert many-to-many relations to one-to-many

Core DB schema



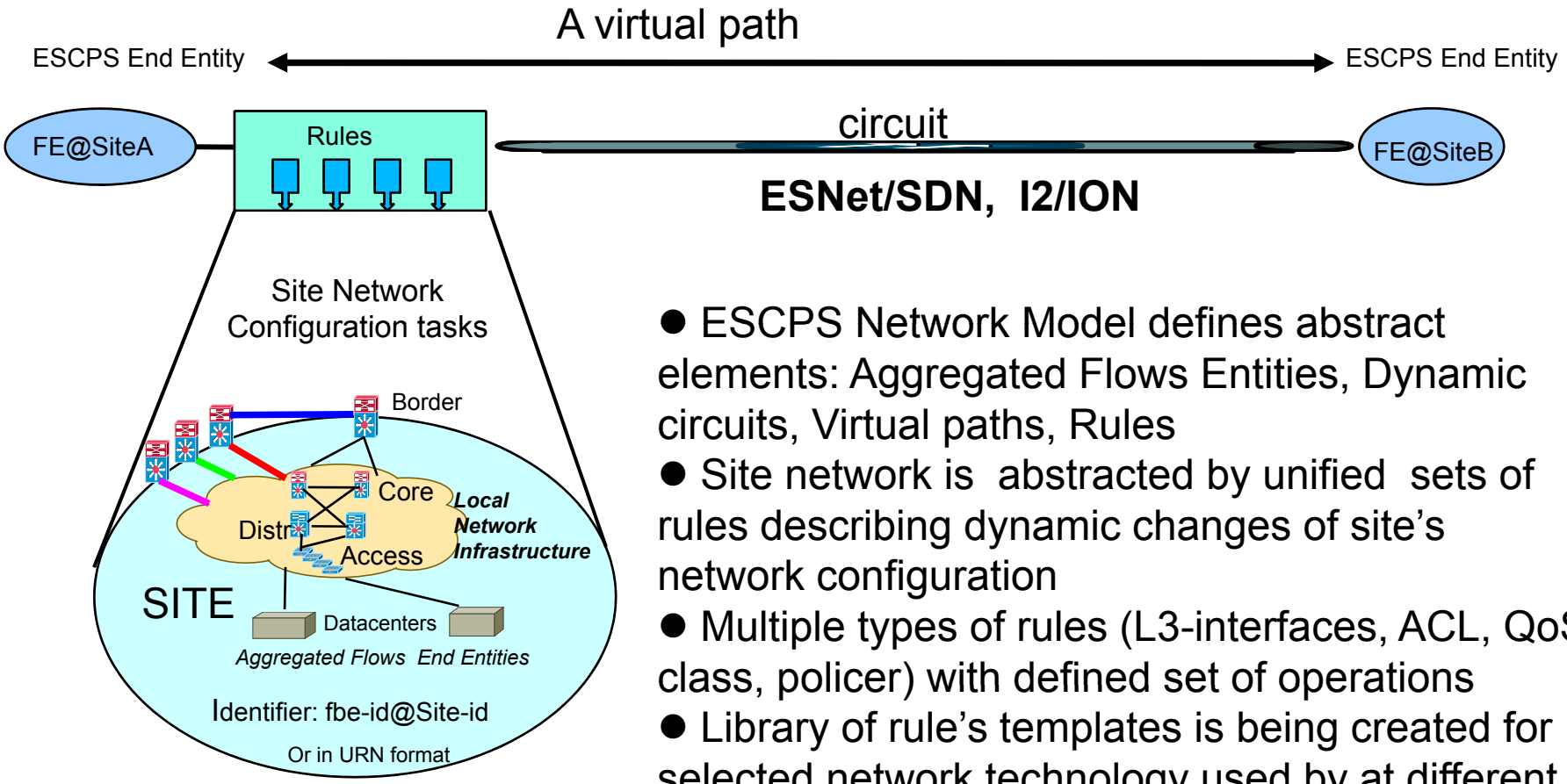

Request@A
reservations
1) R-escps@A
2) R-oscars@A
3) R-WAN
4) R-oscars@B
5) R-escps@B
1 to n reservations
depending on case
e.g., 3 domains – 5 systems
ESCPS@A
OSCARS@A
OSCARS@WAN
OSCARS@A
ESCPS@B

FNAL Component of ESCPS

Local Domain Controller (LDC)

- Configures LAN component of end-to-end path on-demand
- Technology agnostic, create a virtual path within LAN as specified by site and based on local network technology deployed
- Describes local network path in terms of a finite number of configuration rules, unified interface
- Interacts with RESMGR to obtain information on flows in submitted tickets
  - signals RESMGR when job is done or in case of failures
- RESTFul web interface to access its services

ESCPS Local Path 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

Basic LDC Rules

- L3-interface
- ACL
- Vlan
- QoS Class, QoS policer
- Route-map

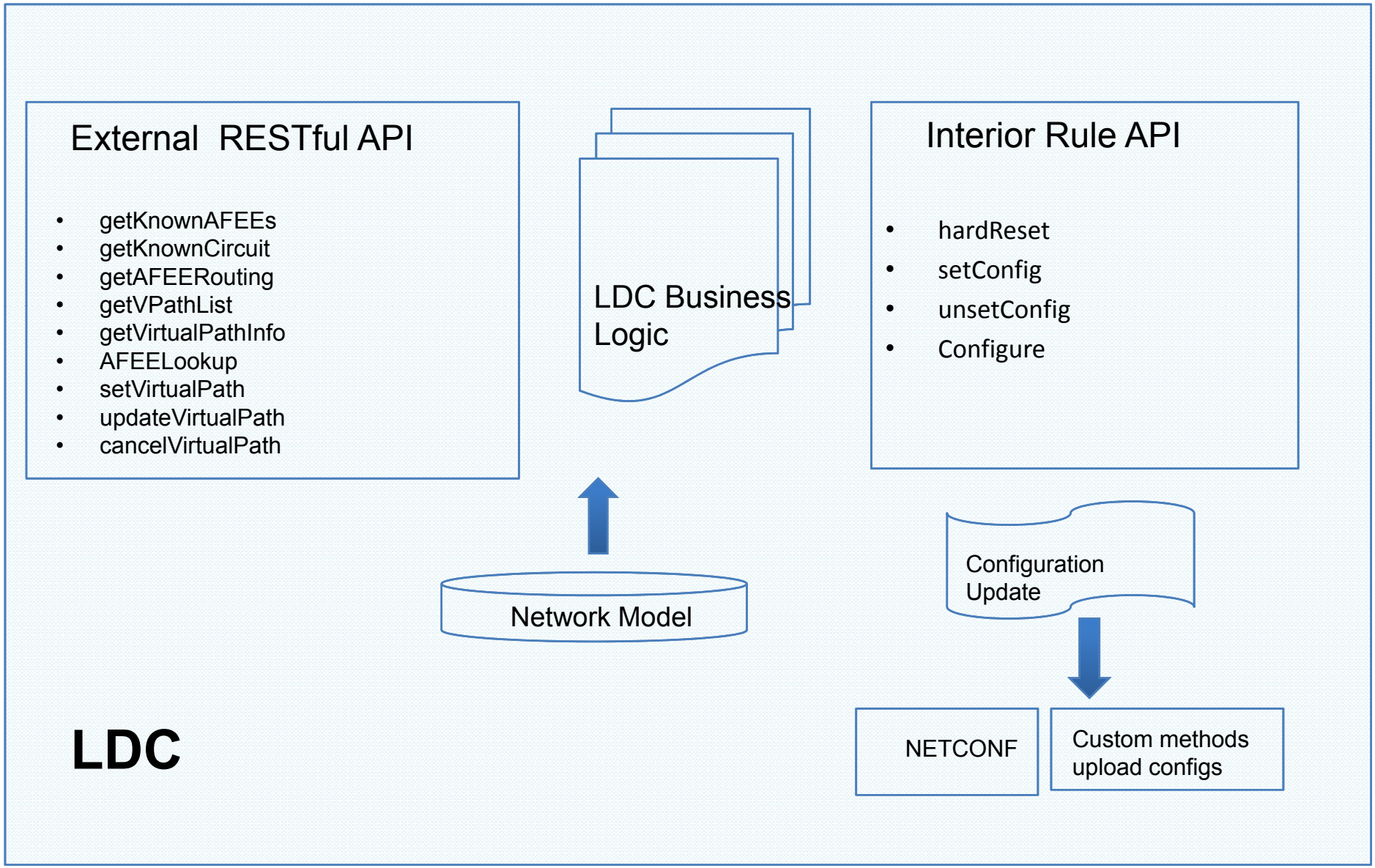
Vendor-specific syntax of these rules is hidden in templates that can be selected/modified when creating a site's specific description of the rules. It is also easy modifiable for site's specifics.

LDC Rules Operations

Unified operations to assemble rules in a virtual path across LAN:

- hardReset – re-create rule from scratch
- softReset – bring a rule into initial state, similar to above but does it less disruptive (depend on a rule), i.e.hardReset ACL – remove ACL completely and then recreate it, softReset: remove all entries but keep ACL in configuration)
- setConfig – create a configuration fragment for rule ready to be uploaded in device(devices)
- unsetConfig – create a configuration fragment to undo setConfig
- Configure – configure selected properties of rule (manipulate by rule's representation in server's memory and rule's operation data)

Glue all together

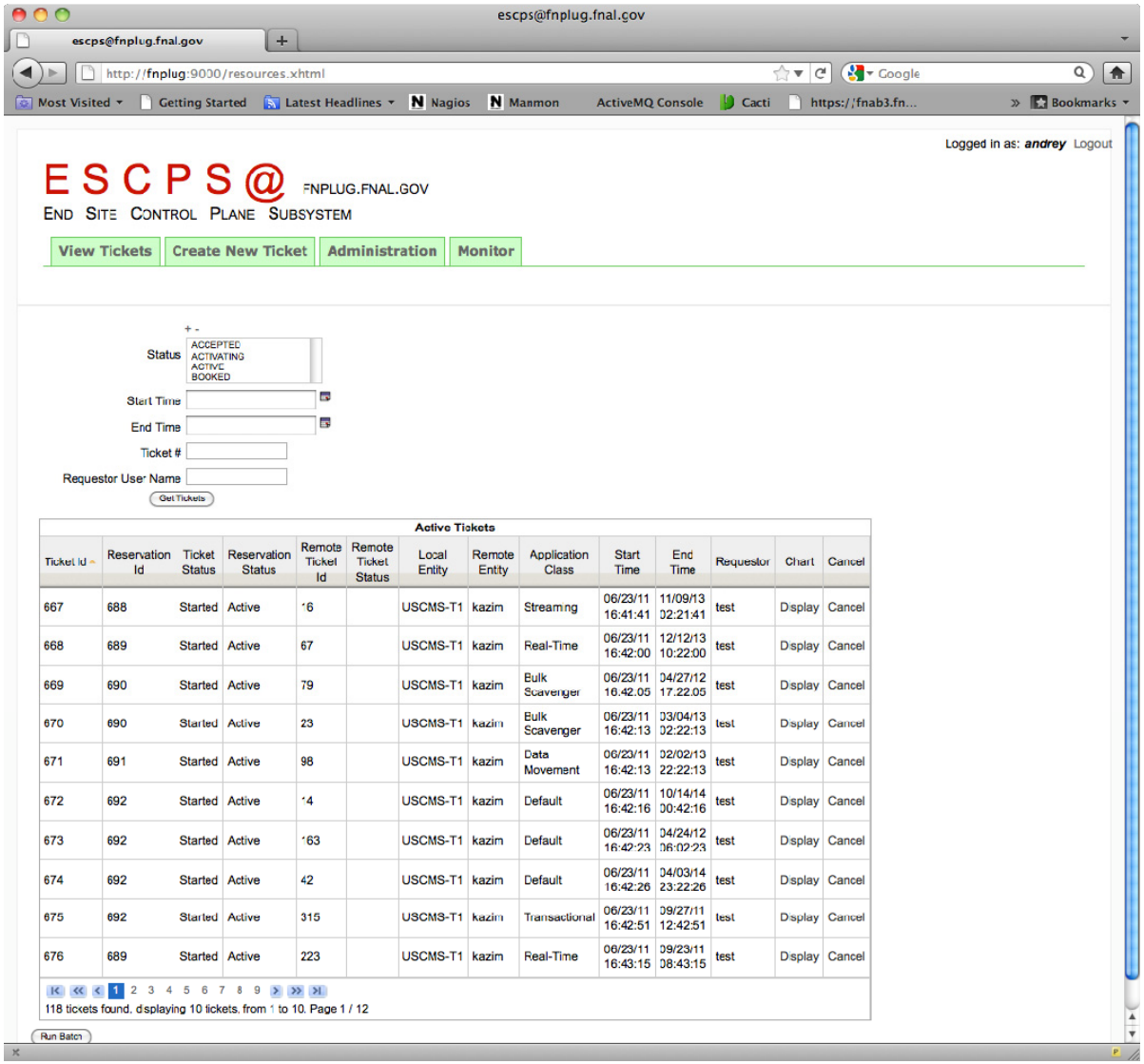


LDC Status and Plans

- Design is completed (XML schemes, Rule templates, Network model examples in XML)
- Current software development is in Perl
- Currently in prototyping stage
- Finished LDC rules for Catalyst 6509/IOS
- Developing LDC rules for NX-OS, Cisco Nexus 7000/5000

LDC Service Manager

- Workflow manager for LDC:
  - provides information about available services, sites and virtual paths
  - Functionally a front-end for LDC services
- Accessible via RESTful Web Services
- Interfaces with other ESCPS components via a simple event-based mechanism.
  - Service requests processed based on the defined workflow.
  - Each state in workflow can be configured with a checklist of awaiting sevents from external processes.
- A java package in beta stage.
  - Installation package includes all components needed in Linux environment

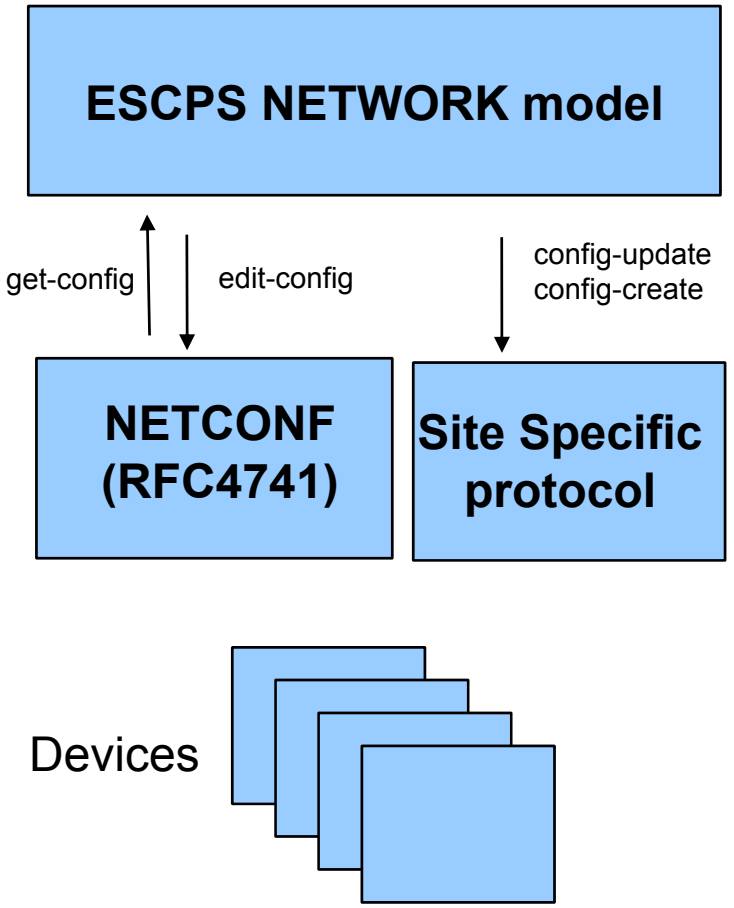


LDC Service Manager

Interactive control tool allows manual service requests, monitor status of requests and reservations, and facilitates testing of LDC rules configurations

LDC Evaluations with NetConf

- NetConf:
  - IETF Network Management Protocol (RFC4741)
  - RPC-based device configuration
  - Uses XML
- Limited vendor support
- Completed evaluation(s) on Cisco Nexus:
  - Complicated, complex, & somewhat confusing
  - Proprietary configuration methods still much easier



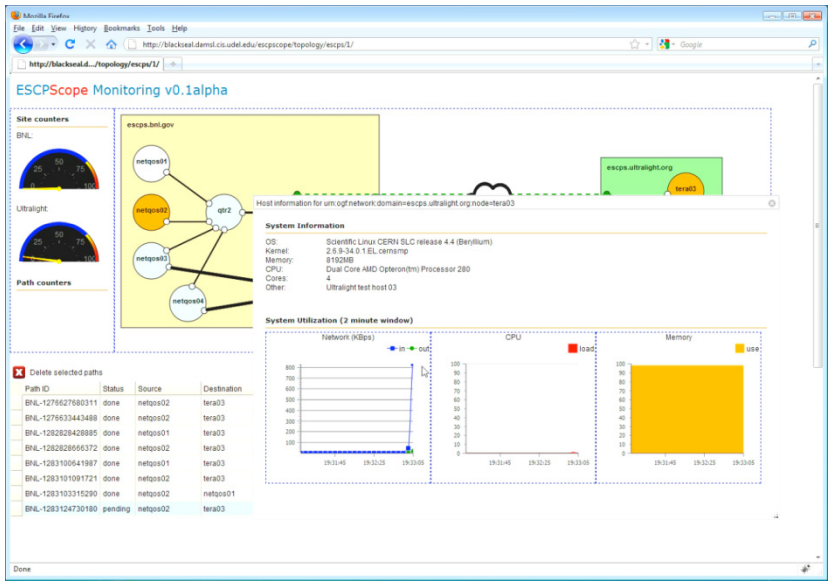
OpenFlow

- OpenFlow support not part of ESCPS project plan
  - However, emergence of OpenFlow shouldn't be ignored
- Current thinking is LDC OpenFlow support would entail just another rules set
  - May entail some modifications to API calls/handling as well
- Will be evaluate OpenFlow over next several months
  - Independent of ESCPS project (and effort…)
  - But will be keeping OpenFlow capabilities in mind for ESCPS

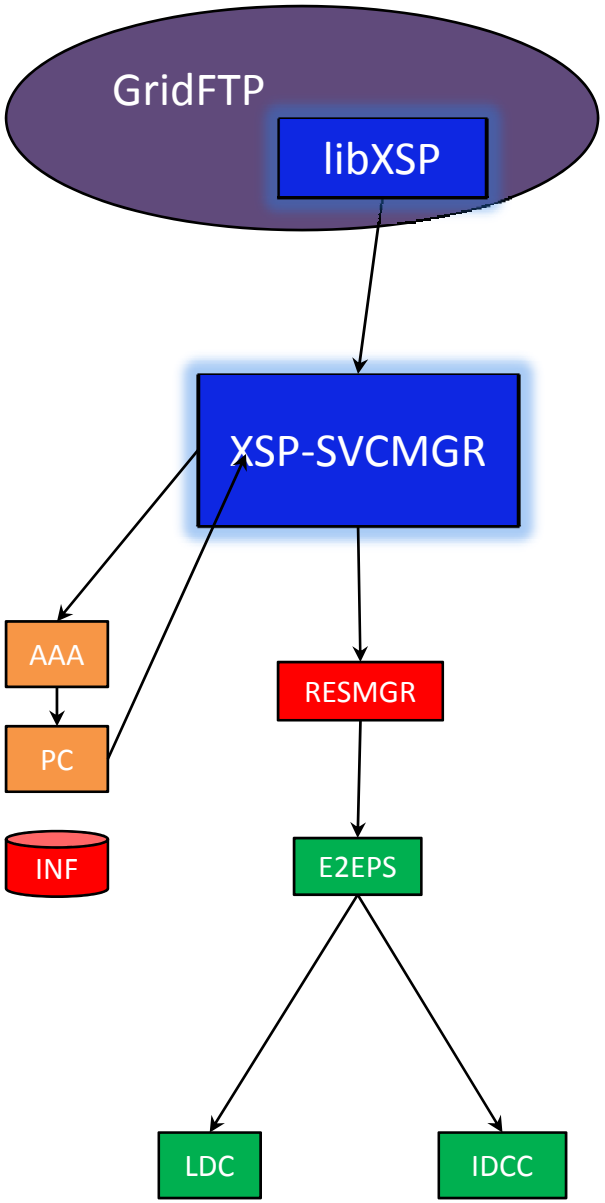
U-DEL Component(s) of ESCPS

Monitoring

- Prototype ESCPScope (Periscope) visualizes topology with integrated data
- Monitors:
  - SNMP counters
  - Looking glass (counters for policy-based routing and filter-based forwarding rules)
  - Host variables (network, cpu, memory)
  - Application performance with summarized logging information to show application-visible performance



XSP SVRMGR



- Talks to prototype RESMGR (or OSCARS) for provisioning
- Talks to the ESCPScope monitoring agent to initiate monitoring
- Socket-compatible library, libXSP can replace network calls in Linux
- libXSP controls host monitoring
- GridFTP XIO driver can be loaded on demand

ESCPS Test Bed Status

- Initial pre-prototype testing between BNL, UMich, and UDel on ESCPS (expanded TeraPaths) test bed
- Project plan calls for integrated test bed
- Not quite in place yet, but…
  - OSCARS circuit in place between BNL & FNAL now

Short Term Objectives

- Complete evaluation of synergy w/ OSCARS platform:
  - Make use of OSCARS code where easy & beneficial
  - Determine longer term convergence path
- Integrate component work into functional prototype
  - Looking at ~6 month development effort
  - Means most ESCPS functions will remain outside of OSCARS
  - Seek out opportunities for collaboration (DYNES?)
- Get testbed fully deployed

Long Term Objectives (Time Permitting)

- Incremental Steps toward the goal of Integration of OSCARS capabilities into ESCPS. These items are categorized as "Research and Development" along with OSCARS.
  - Topology System Management: Use the PerfSONAR to establish local topology, i.e. Topology Schema and Annotation.
  - Add AFE into Topology Schema and Store
    - It has the commonality with OSCARS's new paradigm of Multi-Point to Multi-Point
  - LDC Functionalities can be added into OSCARS framework (Andrey needs to confirm)
  - Authentication in SVCMGR

Long Term Vision beyond Current Project

- Integrate OSCARS capabilities into ESCPS
  - OSCARS is designed for WANs with well defined set of network technologies
  - Significant modifications are necessary to handle heterogeneous, hybrid end-site technologies, including L2, L3, DiffServ, IntServ
  - Extend the OSCARS code base by modifying and/or replacing components to add end-site functionality, including rich API for application use
- ESCPS becomes the end-site version of OSCARS
- Utilization of the OpenFlow framework for network device configurations
- Large scale deployment into DOE sites for eScience applications
- Network Resource Management and Co-scheduling along with CPU and Data Management

Project Information:

ESCPS Documents:

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

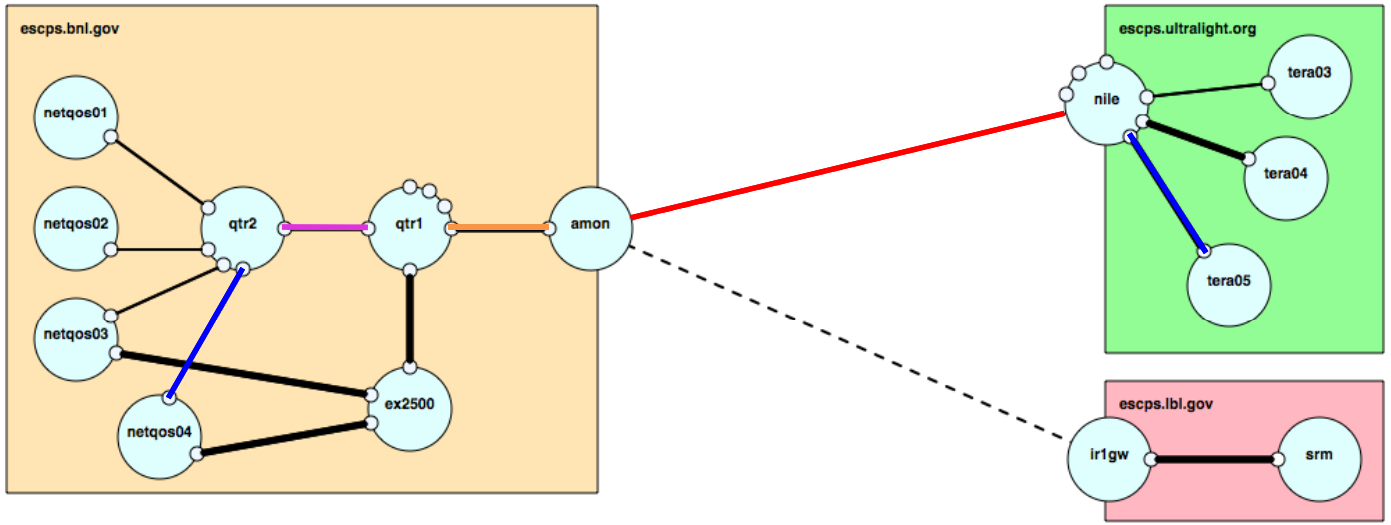
ESCPS Software TRAC repository:

https://damsl.cis.udel.edu/escps

Extra Slides - BNL

ESCPS workflow

1) User sends advance reservation request with source, destination, minimum bandwidth requirements and time period
2) Local service ticket gets created with unique ID
3) Authentication, authorization and policy validation
4) Local validation of resources
   - Find remote ESCPS
5) Initiate service request with remote ESCPS
6) Synchronization of service tickets
   - Identify which ESCPS will have primary role
7) Primary resource manager develops local and remote bandwidth availability graph (BAG)
8) Request transit circuit (parameters resulting from negotiation through BAG intersection or by trial and fail, iterations)
9) Scheduling of E2E path
   - Coordination with remote ESCPS
   - Collect local configuration
10) Wait for circuit setup
    - Verify transit circuit is in place
11) Local domain path activation
    - Synchronize with remote ESCPS
12) Graceful circuit shutdown
    a) Local domain path deactivation
    b) Synchronize with remote ESCPS
    c) Transit circuit teardown
13) Ticket closeout and acknowledgement



- (netqos04.bnl.gov, 10000-10050) ⇔ (tera05.ultralight.org, 15000), protocol ip, 5/17/11@11:00am, 1hr, EF, 500 Mbps
- BNL: #1 {(qtr1, diffserv/pbr)}, #2 {(qtr2,diffserv), (qtr1, pbr)}, #3 {(qtr2, diffserv/pbr), (qtr1, eompls, e1 i/f X, e2 i/f Y)}
- UMich: #1 {nile, diffserv/pbr}
- Pick BNL #3, UMich #1
- Intersect availability from BNL ESCPS, BNL OSCARS, UMich ESCPS
- Try solution with ESnet IDC
- BNL: reservation for LDC to setup blue segment (also perform additional configuration anywhere necessary (?))
- BNL: reservation for local OSCARS to setup magenta segment
- BNL -> UMich: reservation for LDC to setup blue segment
- BNL -> ESnet IDC: reservation for red segment (ESnet + I2)

ESCPS + OSCARS

- Why use OSCARS independently in end-sites? Because…
  - Linking an end-site OSCARS instance to WAN instances, i.e., using it as an IDC, violates end-site security requirements
  - Implementation becomes difficult:
    - End-site “breaks” into two pieces
    - Additional configuration “glue” necessary in the end-site (to be added by the LDC) will be hard to synchronize
    - Switching the end-sites partially to a daisy chain model breaks the negotiation scheme

Ticket Status Monitoring

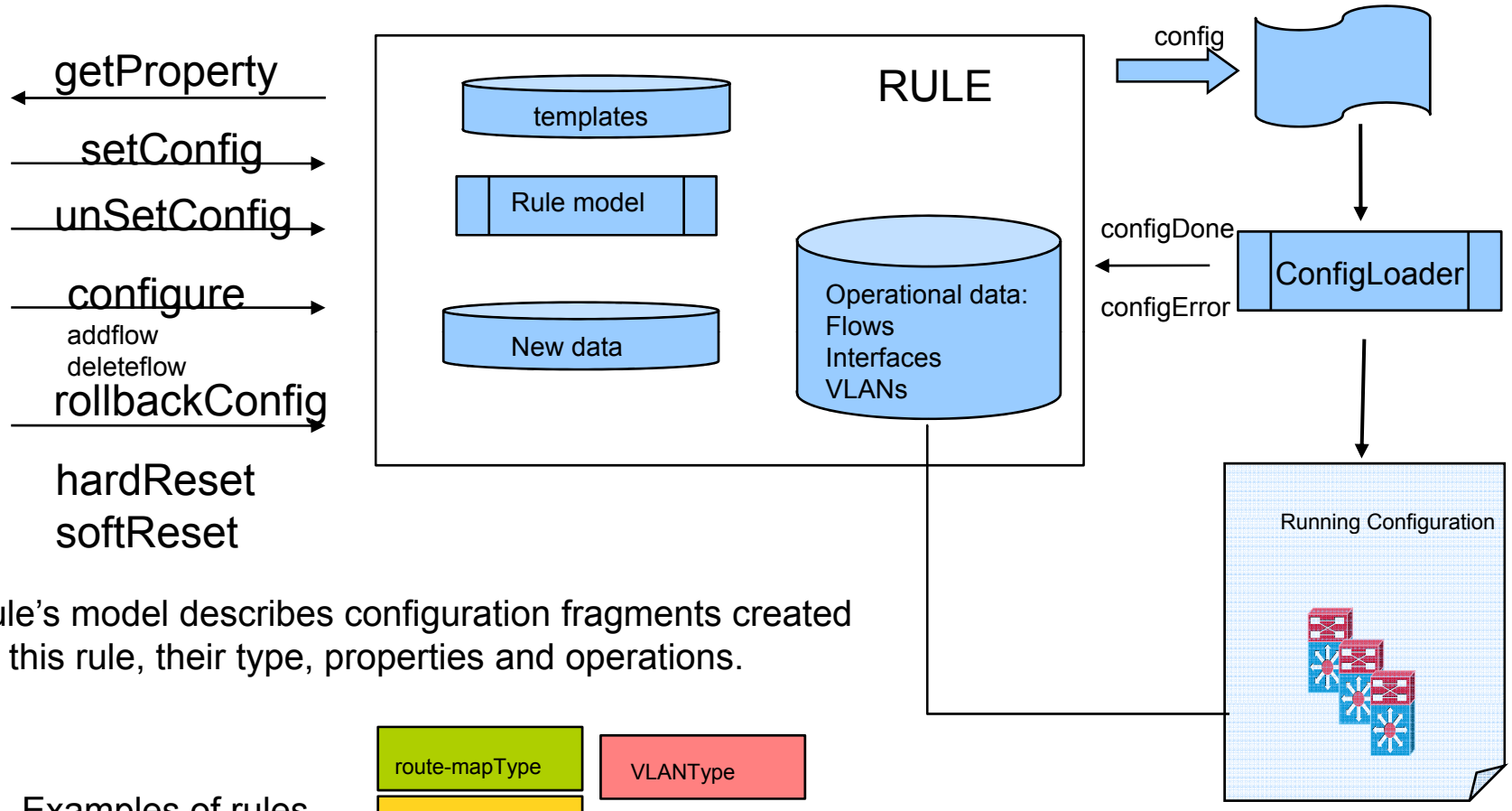
Ticket ID	Owner	Creation Time	Request Type	Status	…				
Reservation #XYZ1									
Source: AFE x = …					Destination: AFE y = …				
Domains	Source	Destination	Creation Time	Start Time	End Time	Bandwidth	Status		
#1									
#2									
#3									
#4									
#5									
Reservation #XYZ2									
Source: AFE x = …					Destination: AFE y = …				
Domains	Source	Destination	Creation Time	Start Time	End Time	Bandwidth	Status		
#1									
#2									
#3									
#4									
#5									

Extra Slides - FNAL

RESTFul LDC API

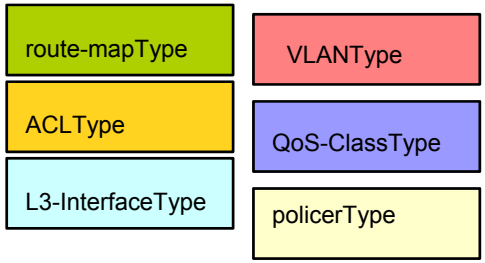
- getKnownAFEEs
- getKnownCircuit
- getAFEERouting
- getVPathList
- getVirtualPathInfo
- AFEELookup
- setVirtualPath
- updateVirtualPath
- cancelVirtualPath

Architecture of a Simple Type Rule



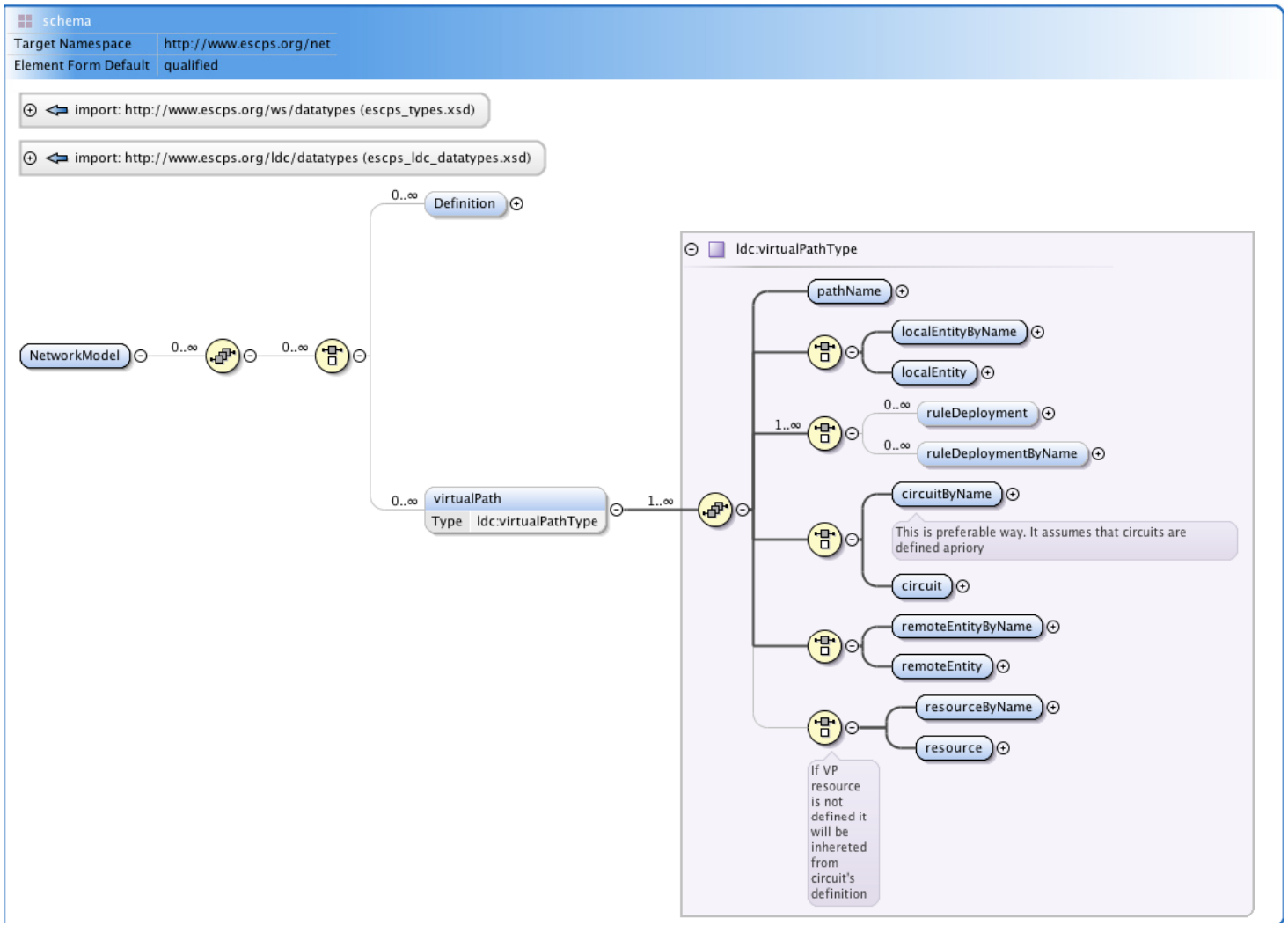
Rule's model describes configuration fragments created by this rule, their type, properties and operations.

Examples of rules



Network Model Schema

Main abstract elements: Virtual Path, Flow Entity (AFEE), circuit, Rule, Resource)



Network Model (Cont)

Defines a virtual path (paths) across LAN in terms of flow entities (AFEE), local and remote, and configuration rules:

….. A fragment of network model:

```
   <virtualPath>
      <pathName xmlns="">UNL-DYNAMIC-VP-2</pathName>
      <localEntity xmlns="">
            <name>USCMS-T1-SRM</name>
            <address>131.225.204.1/32</address>
      </localEntity>
      <ruleDeploymentByName xmlns="">cms-wg-pbr-out</ruleDeploymentByName>
      <ruleDeploymentByName xmlns="">unl-circuit-
dynamic</ruleDeploymentByName>
      <ruleDeploymentByName xmlns="">e2e-pbr-in</ruleDeploymentByName>
      <circuitByName xmlns="">unl-fnal-circuit-dynamic</circuitByName>
      <remoteEntity xmlns="">
         <name>UNL</name>
         <address>129.128.1.1/32</address>
         </remoteEntity>
   </virtualPath>
</NetworkModel>
```

Monitoring status an LDC service ticket…

Display Resource Details

Id:	675	Remote Ticket Id:	315
Status:	Started	Remote Status:	
Application Class:	Transactional	Requestor:	test
Bandwith In:		Bandwith Out:	
Total Bandwith:			
Start Time:	06/23/2011 04:42 PM	End Time:	
Duration:	8280000		

Ticket Workflow

Status	Time	Message	Event	Process	Event Result
Remote Booking	06/23/2011 04:45:12 PM	Request from service	Remote Ticket Booking		Recieved
Local Booking	06/23/2011 04:45:41 PM	Process Flow Started	Resource Bandwidth Reservation	resourceManager	Recieved
			Resource Bandwidth Reservation	secondProcess	-
Booked	06/23/2011 04:46:03 PM	Resource Manager booked Ticket			Recieved
Reserved	06/23/2011 04:46:10 PM	Ticket is consolidated			Recieved
Started	06/23/2011 04:46:10 PM	Ticket Start Time reached			Recieved

Local Flow Entity	
Name:	USCMS-T1
Addresses	
Applications	

Remote Flow Entity	
Name:	kazim
Addresses	
Applications	

Close

Monitoring reservation…

Display Resource Details

Id:	692	Remote Ticket Id:	
Status:	Active	Remote Status:	
Application Class:	Default	Requestor:	
Bandwith In:		Bandwith Out:	
Total Bandwith:			
Start Time:	06/23/2011 04:42 PM	End Time:	10/14/2014 12:42 AM
Duration:			

Reservation Workflow

Status	Time	Message	Event	Process	Event Result
Accepted	06/23/2011 04:45:40 PM	New reservation created			Recieved
Activating	06/23/2011 04:45:41 PM	Process Flow Started	Network Config Setup:activating	ldc	Recieved
			Configuring DCN:DCNConfigs	ldc	Received
Active	06/23/2011 04:46:21 PM	LDC activated reservation			Recieved

Local Flow Entity	
Name:	USCMS-T1
Addresses	
Applications	

Remote Flow Entity	
Name:	kazim
Addresses	
Applications	

Close

Monitoring ticket's status ….

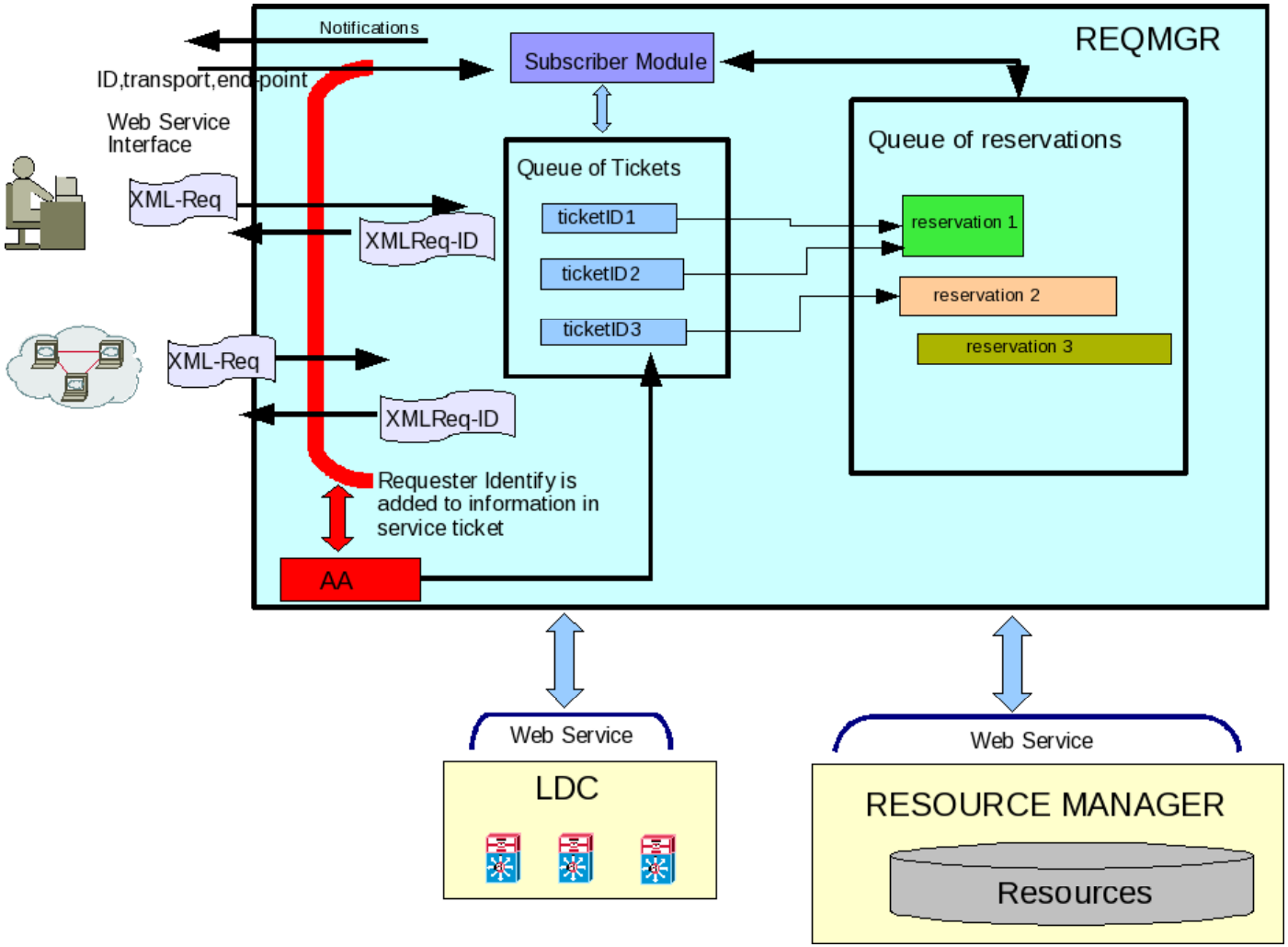
Active Tickets

Ticket Id	Reservation Id	Ticket Status	Reservation Status	Remote Ticket Id	Remote Ticket Status	Local Entity	Remote Entity	Application Class	Start Time	End Time	Requestor	Chart	Cancel
667	688	Started	Active	16		USCMS-T1	kazim	Streaming	06/23/11 16:41:41	11/09/13 02:21:41	test	Display	Cancel
668	689	Started	Active	67		USCMS-T1	kazim	Real-Time	06/23/11 16:42:00	12/12/13 10:22:00	test	Display	Cancel
669	690	Started	Active	79		USCMS-T1	kazim	Bulk Scavenger	06/23/11 16:42:05	04/27/12 17:22:05	test	Display	Cancel
670	690	Started	Active	23		USCMS-T1	kazim	Bulk Scavenger	06/23/11 16:42:13	03/04/13 02:22:13	test	Display	Cancel
671	691	Started	Active	98		USCMS-T1	kazim	Data Movement	06/23/11 16:42:13	02/02/13 22:22:13	test	Display	Cancel
672	692	Started	Active	14		USCMS-T1	kazim	Default	06/23/11 16:42:16	10/14/14 00:42:16	test	Display	Cancel
673	692	Started	Active	163		USCMS-T1	kazim	Default	06/23/11 16:42:23	04/24/12 06:02:23	test	Display	Cancel
674	692	Started	Active	42		USCMS-T1	kazim	Default	06/23/11 16:42:26	04/03/14 23:22:26	test	Display	Cancel
675	692	Started	Active	315		USCMS-T1	kazim	Transactional	06/23/11 16:42:51	09/27/11 12:42:51	test	Display	Cancel
676	689	Started	Active	223		USCMS-T1	kazim	Real-Time	06/23/11 16:43:15	09/23/11 08:43:15	test	Display	Cancel

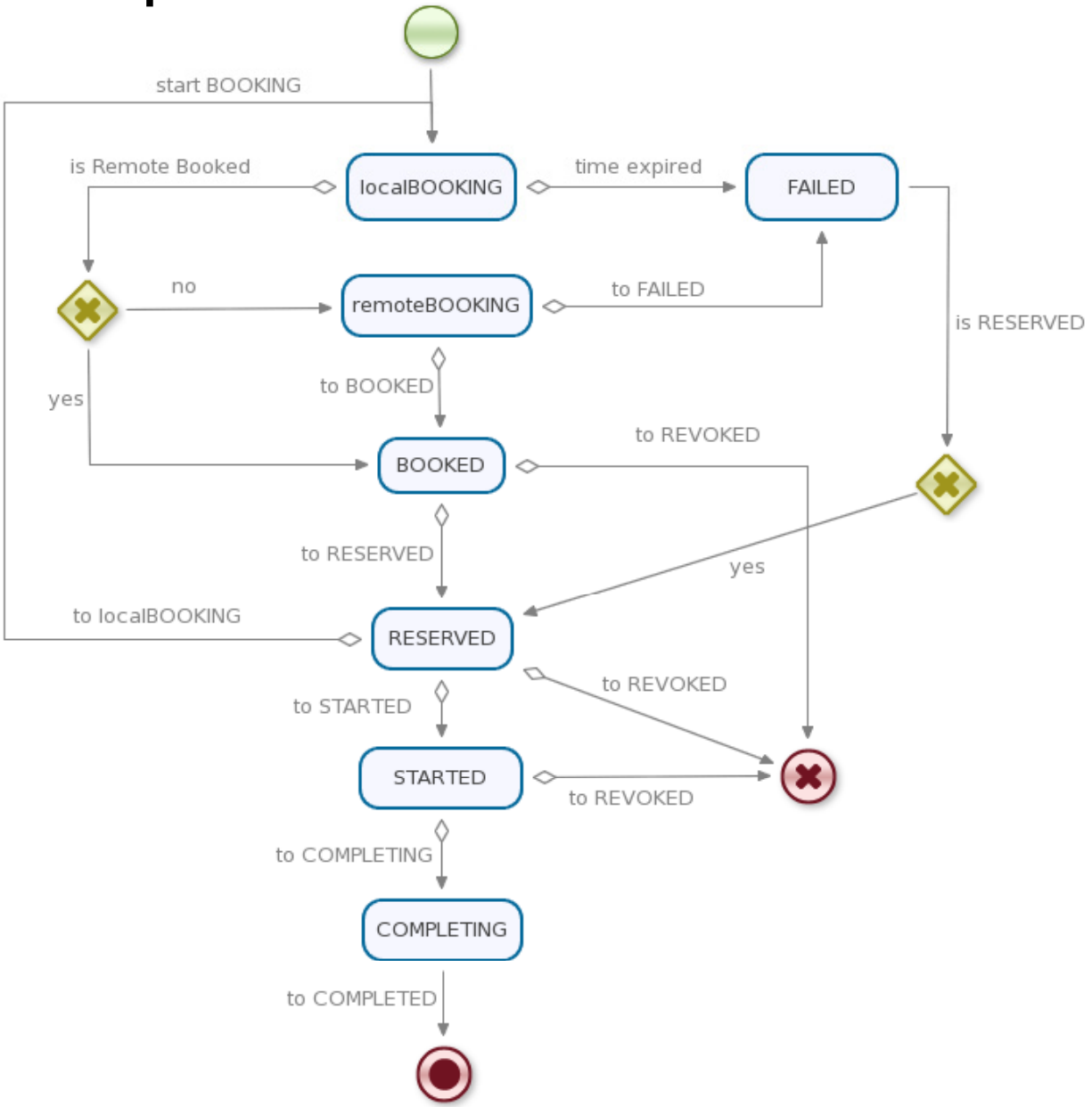
|< << < 1 2 3 4 5 6 7 8 9 > >> >|

118 tickets found, displaying 10 tickets, from 1 to 10. Page 1 / 12

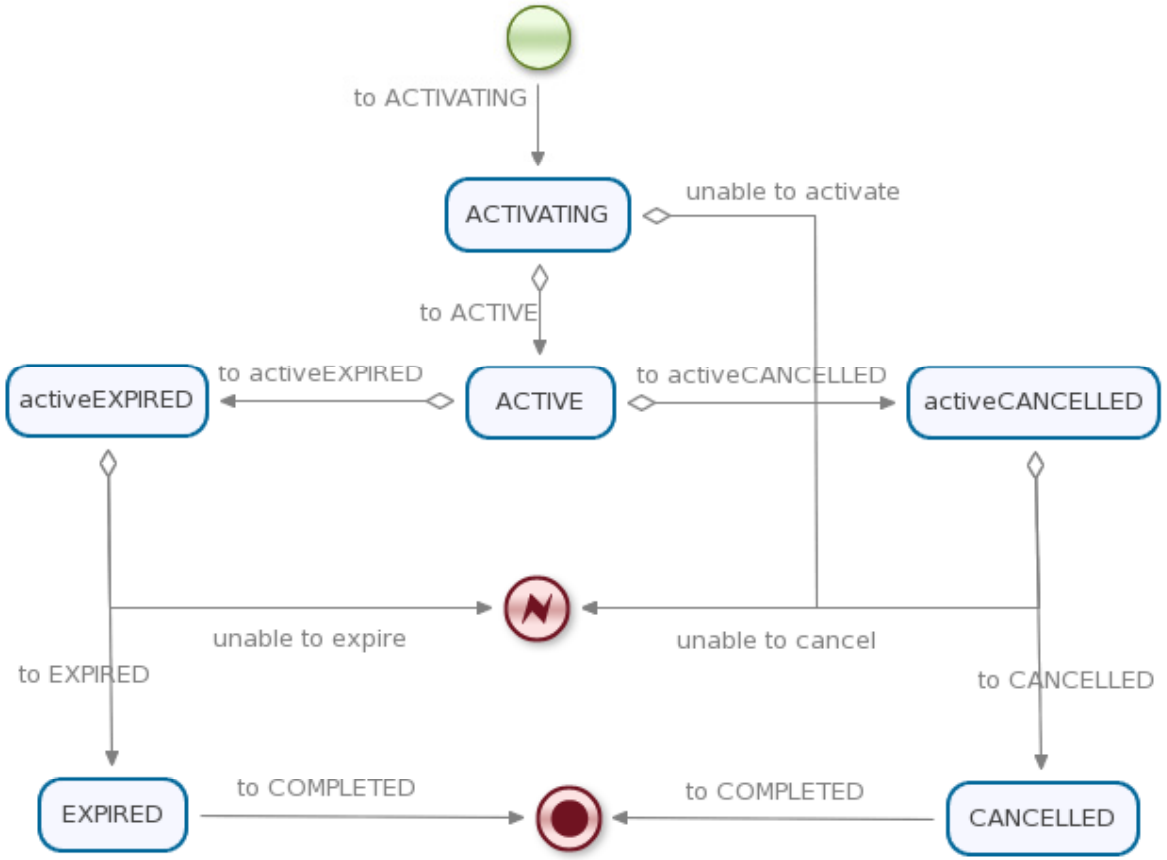
LDC Service Manager Architecture



Simplified LDC Ticket's workflow



Simplified LDC Reservation's workflow



LDC Service Manager RESTFul API

- getServiceTicket
- searchServiceTicket
- addServiceTicket
- updateServiceTicket
- getTicketStatus
- updateTicketStatus
- cancelTicket
- listReservation
- getReservationInfo

Inherit LDC API calls to provide Required informative service to users applications

- getCircuits
- getCircuitInfo
- getListAFEEs
- afeeLookup

Signaling from additional external processes can be to Ticket's workflow via XML configuration file (needs server's restart)

```
<events xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="file:events.xsd">
                <global>
                                <processId
name="resourceManager">
                <userid>resID</userid>
                <password>asasa</password>
                                </processId>
                                <processId
name="secondProcess">
                <userid>secondProcessID</userid>
                <password>asasa</password>
                                </processId>
                                <processId name="ldc">
                <userid>ldcUser</userid>
                <password>asasa</password>
                                </processId>
                                <event>
                <timeout>300000</timeout>
                                </event>
                </global>
……..
```

```
                <state name="ACTIVATING"
check="ALL">
                                <event id="activating">
                <name>Network Config
Setup:activating</name>
                <timeout>250000</timeout>
                                                <process
id="ldc">
                <defaultMessage>LDC activated
reservation</defaultMessage>
                <defaultErrorMessage>Resource Manager
unable to book Ticket</defaultErrorMessage>
                                                </process>
                                </event>
                                <event id="DCNConfig">
                <name>Configuring
DCN:DCNConfigs</name>
                                                <process
id="ldc">
                <defaultMessage>LDC activated
reservation</defaultMessage>
                <defaultErrorMessage>Resource Manager
unable to book Ticket</defaultErrorMessage>
                                                </process>
                                </event>

                </state>
```