**API Management and System Integration**

**Vendor solution and POC in Lightpath**

**Nov. 2016**

**Lightpath**

**200 Quadrangle**

**Jericho, NY 11753**

# Table of Contents

## [Introduction](#_Introduction)

[Evaluation Approach](#_Evaluation_Approach)

[**Business Goals**](#_Business_Golas)

[**Technical Requirements**](#_Technical_Requirements)

[**POC**](#_POC)

[POC Goals](#_POC_Goals)

[Objectives](#_Objectives)

[POC Architecture Choices](#_POC_Architecture_Choices)

[POC Environments](#_POC_Environments)

[Lightpath Lab](#_Lightpath_Lab)

[Altice USA Private Cloud](#_Altice_USA_Openstack)

[POC Process and Timeline](#_POC_Process_and)

[**Use Cases**](#_Use_Cases)

[**High Level Business Scenarios/Use Cases**](#_High_level_business)

[**Use Cases**](#_use_cases_1)

[**POC Environment**](#_POC_Environment)

[**Environment Parameters**](#_POC_Environment_Parameters)

[**Accessible Resources**](#_POC_Accessible_resources)

# Introduction

This document outlines the business need, technical drivers for introducing API management and system integration to Lightpath. It further details the POC process, use cases, timeline and vendor requirements for conducting the POC.

## Evaluation and POC Approach

Lightpath prefers on-premises POC to assess and validate the key functions and features of proposed solution that can benefit and bring in immediate values to the ongoing projects and integration projects in 2017. From a high-level, the following functions are high on the use cases list for POC.

1. Rapid prototype, development and deployment of APIs;
2. Full lifecycle management of API including design, develop/create, secure, publish, register, consume, monitor, block, depreciate and retire.
3. API orchestration and workflow creation for business process;
4. Security
5. Connect to legacy applications without REST API;
6. Workflow orchestration
7. 3rd party cloud applications integration such as salesforce;
8. Event-Trigger-Messaging based workflows
9. Data mapping and transform

# Business Goals

* Improve operational efficiency with automation, DevOps and achieve greater business agility;
* Improve developer productivity, shorten project lifecycle, and streamline internal integration development process;
* Link IT projects to revenue generating initiatives, simplify and speed onboarding process of new systems, services and products to Lightpath;
* Streamline network device configuration, network services configuration and provisioning process;
* Improve and standardize security, open up legacy backend and provide data services securely and consistently;
* Modernize IT systems to move to next generation programmable network;

# Technical Requirements

The function and feature level requirements are detailed in the RFP requirement documents, this section describes the high-level technical requirements and drivers for introducing API management and lightweight ESB solution to Lightpath.

The proposed integration solution is to use API-centric approach but also provide lightweight ESB support for web services, messaging, orchestration as well as connectors to legacy systems. The solution should include four logical components and bring four technical benefits to Lightpath IT and Network Engineering systems.

1. **API Management**

A solid API management system, include API gateway, API monitoring/analytics and developer portal, provides the key enabler for cloud computing, API-based integration, API-based microservices architecture, and new approach for interacting with customers and business partners. It is the effective approach for integrating new systems, new technologies with existing legacy systems.

1. **Lightweight ESB**

The lightweight enterprise service bus (ESB) allows for business workflow automation, orchestration without extensive developer involvement, as well as advanced system integration.

1. **Enterprise Messaging System**

Create asynchronous message-based enterprise systems and integrate with other BSS/OSS applications using ESB to better communicate among internal applications as well as with customers, for example, notify only those customers being impacted by DDos attack via emails or texting message, or automatically notify customers when certain tasks are performed or certain services are activated.

1. **Security**

The API management tool also has built-in standardized and sophisticated security modules such as SAML, oAuth, TLS, WS-Security, and SIEM. A common gateway is used for control security;

The proposed RFP addresses all those four technical function areas. The preferred solution is to use single system architecture infrastructure to support API management, B2B integration, Cloud-On-Premises integration, internal system integration, messaging-based integration, “southbound” integration such as direct network device access and SDN/NFV API-based integration, as well as solid security related all above mentioned integrations.

# POC

The purpose of POC is to systematically evaluate vendor’s solutions for API management and system integration in Lightpath development environment.

## POC Goals

1. Get experience with API management and lightweight ESB systems with top vendors;
2. Determine if vendor solutions could address Lightpath’s challenges and needs;
3. Identify a vendor solution that might best fits Lightpath’s immediate needs and long term technical strategy and roadmap;
4. Achieve these goals in less than two months;

## Objectives

1. Adopt API-centric, event-messaging driven approach for building service layers/microservices to integrate on-premises systems and cloud applications;
2. Replace existing solutions that Lightpath currently use to integrate systems
3. Reduce point to point integration and replace with API-based integration;
4. Make integrations between/among Lightpath on-premises systems and cloud applications more resilient, agile, easy to replace and reuse by reducing or eliminating non-standard based solution and 3rd party dependencies;
5. Single source of “truth” data entity services such as Account, Building, Customer, Circuit, Device etc.

## POC Architecture Choices

As the following diagram shows, the architecture is simplified to make it easy to conduct POC in a virtual environment. Two API gateway nodes are clustered or HA available, so are the two lightweight ESB nodes. The message queue is a standalone on a remote node, using Apache ActiveMQ. The applications are both on-premises applications and cloud based 3rd party applications.



However the POC architecture will be discussed with vendors right after the onsite presentation, vendor may change or suggest the architecture for best fit vendor’s solution but also consider and be able to conduct Lightpath POC use cases.

## POC Environments

Lightpath will prefer POC in Lightpath virtual environment. However this needs more discussion with vendors

### Lightpath Lab

Lightpath development lab has VMware ESXI v 5.5.0, and can provision virtual machine using CentOS 7 or Ubuntu 15.0 images (or other Linux images) with four network interfaces.

Developers can access this lab from desktop using vSphere Client V5.5.0 (free download). Vendor can decide prebuild Linux images in OVF format and import to Lightpath ESXI server or let Lightpath prepare a Linux images in the lab and then vendor can download all its software, install to the vm in the lab.

### Altice USA Openstack private cloud

Altice USA also has Openstack private cloud development. It is also possible to create a POC environment in Openstack cloud.

## POC Process and Timeline

1. Make sure that NDA is signed
2. Onsite presentation, two hours;
3. Plan for POC, timing, duration, architecture, environment etc;
4. Use cases for POC are in two group: Vendor defined and Lightpath defined
5. Vendor and Lightpath work together to finalize the POC architecture choice;
6. Both vendor and Lightpath will provide its own use cases to each other ahead of time for reviewing and preparing;
7. Lightpath builds POC lab environment for each vendor;
8. Lightpath provides the necessary backend access points and API details
9. Lightpath strongly suggests that vendor install and execute the POC in Lightpath dev environments;
10. Vendor’s presentation and POC may have different timeframes;
11. All vendors can plan to have **1 to 3 day** timeframe for POC;
12. Both vendor and Lightpath engineers will commit time to implement POC use cases;
13. Vendor provides/downloads all software required for conducting POC;
14. If necessary, Vendor can prepare the vm images with loaded necessary software to conduct POC, especially for those vendor defined use cases;
15. Engineers meeting to discuss and share the POC result, use case by use case;

# Use Cases

Lightapth engineers are still actively working on use cases. However, the following two sections show some examples of Lightpath defined use cases. Section 5.1 lists the high level functionalities that vendor may need to demo or conduct POC either with vendor defined use cases or Lightpath defined use cases. Section 5.2 lists some specific use cases.

## High level business scenario and use case

ESB / API Mgmt RFP Use Case: Communication Service

Business problem: Multiple processes (auto and manual) send communications to customers with no consistent format or audit trail.

Story: As an organization I can manage data driven outbound communications with configurable templates so that email messages are professional and consistent.

Demonstrate lifecycle from prototype to deployment and ongoing management with specific focus on:

1. Prototype

API driven/led pattern, signature first, schemasample responses

Automated test case/suite creation, auto test suite execution

Run time configuration

1. Development

Consume web services SOAP/Rest, compile a WSDL, send and receive xml/json message.

Custom code activity stitched into data flow

Consume select JMS (queue & topic) event

Consume / Create / Update Salesforce objects

Database - Oracle, PostgreSql etc

Transform (xml, json, xslt,etc), Merging (two xml nodes)

Scheduled job (minutes, hourly, daily, weekly, etc)

Send and Receive Emails.

Multiple developers work on same project (code repo, commits, merges).

Step through debug

1. Logging (vendor’s logger, soap endpoint, liferay soay portlet, )

Central logging.

Exception handling and retries.

JMS message handling in case of fault

1. Deployment

Versioning (multiple version of same application with different features)

Push to different environments.

Automated deployment from GIT/any versioning repo.

How HA configured.

1. Management

Monitor health of services

Performance monitoring

How HA fail over handled.

Security (Authentication, Authorization, SSO)

Auditing

Functional Requirements:

Handle Web Service request with unformatted data.

Acquire template for formatted message. Transform data into message.

Store all relevant message info.

Deliver message to recipients.

## use cases

send ip addresses

Monday after 4:00-5:30pm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Use Case Name** | **RFP**  **Code** | **Use Case Description** | **Use Case**  **Resources** | **Note** |
| **1** | Manage  Existing RESTful API |  | Given an existing raw RESTful API that provides data from Lightpath backend system, design a new API based on the best practice to manage the existing API, add manageable features such as security (oAuth, API key, client/secret id), caching, monitoring, logging etc., then publish and register the managed API to API catalog or repository where developers can access. | Swagger 2.0 |  |
| **2** | Crease new RESTful  API |  | Use API-First to design an API for prototype |  |  |
| **3** | Create a new REST API over external  REST API |  | Design a new Lightpath REST API and call an external 3rd party REST API, for example: <https://localhost/lightpath/api/ip/216.2.193.1/json>. | Example external API  <http://geo.groupkt.com/ip/216.2.193.1/json>, |  |
| **4** | Import a REST  API from a swagger file |  | Generate API from swagger file | Provide the swagger.json file to define an API |  |
| **5** | Management of existing web services |  | * Chaining of web service calls * Combining of results – e.g. phone/circuits REST call results into one * Providing for security | Lightpath will provide the the existing URLs | Send json data for circuit and phone |
| **6** | Event-driven notification |  | Design two workflows: WF-A and WF-B,  WF-A has frontend REST API that can accept POST request, and immediately writes it to a queue;  WF-B has a queue watcher and read from the queue, and then transform the message, prepare the mail body or texting body, and send it out via email and/or sms texting message; |  |  |
| 7 | Network Device autoconfig services |  | Chaining devide config services and expose a new services targeting different network devices with different data format (text file, json, xml etc.) |  |  |
| 8 | JMS task notification |  | Provisioning system completes a task, and the task pushes a JMS payload to a messaging queue (ActiveMQ or Weblogic queue). Design a workflow that pick up the message and call salesforce to update |  |  |
| 9 | File based integration |  | Design a workflow to allow for transfering files between endpoints using secure protocol (SFTP etc.) and also with standard-based encryption protection. |  |  |
| 10 | CI/CD, GIT integration |  | How to achieve continuous integration and continuous deployment, as well as source code control such as GIT |  |  |

# POC Environment

## POC Environment Parameters

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Node A | Node B | Node C | Node D | Node E | Note |
| 1 | VM Name |  |  |  |  |  |  |
| 2 | Dev user/pwd | lpdev/.lpdev | lpdev/.lpdev | lpdev/.lpdev | lpdev/.lpdev | lpdev/.lpdev |  |
| 3 | Root user/pwd | root/.root. | root/.root. | root/.root. | root/.root. | root/.root. |  |
| 4 | Cluster/HA | Apim cluster | Apim cluster | Esb cluster | Esb cluster | Queue/Web |  |
| 5 | Ens32 Green net | 10.5.18.159 | 10.5.18.84 | 10.5.18.61 | 10.5.18.9 | 10.5.18.210 |  |
| 6 | Ens33 Private net | 10.20.100.16 | 10.20.100.15 | 10.20.100.10 | 10.20.100.9 | 10.20.100.36 |  |
| 7 | DMZ (not used) | 10.20.200.9 | 10.20.200.8 | 10.20.200.7 | 10.20.200.6 | 10.20.200.9 |  |
| 8 | Cluster/HA | APIM cluster | APIM cluster | ESB cluster | ESB cluster |  |  |
| 9 | Oracle client |  |  | Instant 12.1 | Instant 12.1 |  |  |
| 10 | Java |  |  |  |  |  |  |
| 11 | Python |  |  |  |  |  |  |
| 12 | Git |  |  |  |  |  |  |
| 13 | Docker |  |  |  |  |  |  |
| 14 | OS | CentOS 7.2 or  Ubuntu 15.04 | CentOS 7.2 or  Ubuntu 15.04 | CentOS 7.2 or  Ubuntu 15.04 | CentOS 7.2 or Ubuntu 15.04 | CentOS 7.2 or  Ubuntu 15.04 |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

## POC Accessible resources

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Endpoint** | **Hostname** | **IP** | **TCP Port** | **User/pwd** | **Resources** | **Note** |
| 1 | **CCOL DB** | cvsdlpd1.cscdev.com | 172.16.14.34 | 1522 | WEBPUSR/WEBPUSR |  | Connected tested |
| 2 | **Portal**  **REST APIs** | lprestdev.cscdev.com | 172.16.24.237  10.144.125.149 | 80,8080, 443  8181 | none |  | Connection not ready yet |
| http://lprestdev.cscdev.com/circuits/v1/accounts/{ACCOUNT\_NUMBERS}/circuit/{CIRCUIT\_ID}.json  http://lprestdev.cscdev.com/circuits/v1/accounts/{ACCOUNT\_NUMBERS}/buildingid/{BUILDING\_ID}.json  http://lprestdev.cscdev.com/circuits/v1/accounts/{ACCOUNT\_NUMBERS}.json  http://lprestdev.cscdev.com/phone-numbers/v1/accounts/{ACCOUNT\_NUMBER}/phone/{PHONE\_NUMBER}.json  http://lprestdev.cscdev.com/phone-numbers/v1/accounts/{ACCOUNT\_NUMBERS}/buildingid/{BUILDING\_ID}.json  http://lprestdev.cscdev.com/buildings/v1/accounts/{ACCOUNT\_NUMBERS}/types/{BUILDING\_TYPES}.json | | | | |  |
| 3 | **ACS**  **REST APIs** | cvldlpacs1new.cscdev.com | 172.16.110.55 | 80, 8081  8082, 443 | none |  | See below |
| Device Config Rest ACS Micro Service: (Port 8081)  http://cvldlpacs1new.cscdev.com:8081/lightpath/acs/rest/?serial\_number=CFG1423924  Device Config Text ACS Micro Service:  Un-Secured: (Port 80)  http://cvldlpacs1new.cscdev.com:80/lightpath/acs/config/v1/download/CFG1423924  Secured: (Port 443)  https://cvldlpacs1new.cscdev.com:443/lightpath/acs/config/v1/download/CFG1423924  Transfer Phone config Files: (Port 8082)  http://cvldlpacs1new.cscdev.com:8082/lightpath/TransferPhoneConfigFiles/HCVLNYAHDS5/cvldlpacs1.cscdev.com | | | | | Connection tested,  port 8081 is good,  port 80, 8082, 443 need to further verify |
| 4 | **ACS DB** | cvldfdb1.cscdev.com | 172.16.106.226 | 1531 | ACSUSR / justacs |  | Connection tested |
| 5 | **Netcracker DB** |  | 172.16.18.73 | 1535 | rdbmgr /  rdbmgr\_dev | Connect string  oracle://rdbmgr:rdbmgr\_dev@172.16.18.73:1535/lpnetcd | Connection tested |
| 6 | **REST API Netcracker**  **ODB, RDB** |  | 10.5.18.208 | 8080 | none |  | Connection tested |
| http://10.5.18.208:8080/api/lightpath/netcracker/docs/ | | | | | tested |
| 7 | **salesforce** | https://test.salesforce.com/ |  |  | [apidemo@poc.dev](mailto:apidemo@poc.dev)  Gr8@gain | API ? |  |
| 8 | **Liferay**  **Soap**  **End**  **Point 1** | olpintranetdev2.cscdev.com | 172.16.24.172 | 80 | none | No need | Firewall not ready |
| http://olpintranetdev2.cscdev.com/LightpathCommunicationServices-portlet/api/axis/Plugin\_lp\_communication\_CommunicationService?wsdl | | | | |  |
| 9 | **Liferay soap endpoint 2** | cvldfap3.cscdev.com | 172.16.18.40 | 25053 | none | Use this one | Connection  tested |
| http://cvldfap3.cscdev.com:25053/LightpathCommunicationServices-portlet/api/axis/Plugin\_lp\_communication\_CommunicationService?wsdl | | | | | tested |
| 10 | **ActiveMQ** | http://10.5.18.31:8161/admin/ | 10.5.18.31 | 8161 | admin/admin |  | tested |
| 11 | **Opendaylight REST API examples** |  | 10.5.18.222 | 8181 | admin/admin | These are some examples of REST APIs exposed from network controllers such as Openday light | Tested, |
| <http://10.5.18.222:8181/apidoc/explorer/index.html>  http://10.5.18.222:8181/index.html#/yangui/index | | | | | tested |
| 12 |  |  | | | | |  |
|  |  |  | | | | |  |