Solution Description Document

**Automated Lab-as-a-Service**

# Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Key changes** | **Approver** |
| **0.1** | 24-Oct-2019 | Ashima Sharma | Initial Draft | Marin Komadina/Andreas Watermann |
| **0.2** | 28-Oct-2019 | Ashima Sharma | Updates after first architecture meeting | Marin Komadina/Andreas Watermann |
| **0.3** | 30-Oct-2019 | Ashima Sharma | Updates after Lab Managers Meeting | Marin Komadina/Andreas Watermann |
| **0.4** | 22-Nov-2019 | Ashima Sharma | Review comments from Marin incorporated | Marin Komadina/Andreas Watermann |
|  |  |  |  |  |

[Document History 2](#_Toc25360807)

[1 Introduction 5](#_Toc25360808)

[1.1 Document Purpose 5](#_Toc25360809)

[1.2 Intended Audience 5](#_Toc25360810)

[1.3 Glossary 5](#_Toc25360811)

[1.4 Document Organization 6](#_Toc25360812)

[2 Solution Overview 7](#_Toc25360813)

[2.1 Solution objective 7](#_Toc25360814)

[2.2 Labs under consideration 7](#_Toc25360815)

[2.3 User Personas 8](#_Toc25360816)

[2.4 Solution scope 9](#_Toc25360817)

[2.5 Assumptions 9](#_Toc25360818)

[2.6 Constraints 9](#_Toc25360819)

[3 Functional Architecure 10](#_Toc25360820)

[3.1 Lab Portal 10](#_Toc25360821)

[3.2 Inventory Management 11](#_Toc25360822)

[3.3 User Management 12](#_Toc25360823)

[3.4 Service Management 12](#_Toc25360824)

[3.5 Capacity Management 13](#_Toc25360825)

[3.6 Central Execution Engine 13](#_Toc25360826)

[3.7 Monitoring Management 14](#_Toc25360827)

[3.8 Analytics Engine 15](#_Toc25360838)

[3.9 Workflow Engine 16](#_Toc25360839)

[3.10 Automation Engine 16](#_Toc25360840)

[3.11 Content Management System 16](#_Toc25360841)

[3.12 Search Engine 16](#_Toc25360842)

[3.13 Test Automation 17](#_Toc25360843)

[3.14 API Gateway 17](#_Toc25360844)

[3.15 Internal Communication Channels 17](#_Toc25360845)

[3.16 Logging and Auditing 18](#_Toc25360846)

[3.17 Notification and alerts 18](#_Toc25360847)

[4 Interfaces 19](#_Toc25360848)

[4.1 Internal 19](#_Toc25360849)

[4.2 External 19](#_Toc25360850)

[5 Message and Function Workflow 20](#_Toc25360851)

[5.1 User Registration 20](#_Toc25360852)

[5.2 Connectivity Management 20](#_Toc25360853)

[5.3 Network Monitoring 20](#_Toc25360854)

[6 Data design 22](#_Toc25360855)

[7 Non Functional Design Considerations 23](#_Toc25360856)

[7.1 Scalability 23](#_Toc25360857)

[7.2 Reliability 23](#_Toc25360858)

[7.3 Modularity 23](#_Toc25360859)

[7.4 Security 23](#_Toc25360860)

[7.5 Flexibility 23](#_Toc25360861)

[8 Detailed Software Design 24](#_Toc25360862)

[8.1 Portal 24](#_Toc25360863)

[8.2 LaaS 24](#_Toc25360864)

[8.3 Monitoring 24](#_Toc25360865)

[8.4 Networks and Operations 24](#_Toc25360866)

[9 Development Environment 25](#_Toc25360867)

[A. Appendix 26](#_Toc25360868)

[9.1 A.1 Open Issues 26](#_Toc25360869)

# Introduction

## Document Purpose

The purpose of this document is to present the overall Solution Architecture of the Automated LaaS solution by DT. It describes the subsystems and components of the solution, presenting a view of how these subsystems interact for servicing different user stories.

It provides a conceptual view of the major building blocks then detailing how these blocks can be achieved and integrated in the existing lab.

## Intended Audience

This document is intended as a reference document for the following set of people:

* Automated LaaS Team
  + Product Owner
  + Architects
  + Developers
  + QA/Test Team
* Lab Manager
* Lab customer

## Glossary

This section covers the various acronyms used in the document.

|  |  |
| --- | --- |
| Acronym | Description |
| DT | Deutsche Telekom |
| LaaS | Lab as a Service |
| GTF | Group Test Facility |
| CMDB | Configuration Management Database |
| E2E | End to end |
| IMS | IP Multimedia Subsystem |
| KPI | Key Performance Indicators |
| L2/L3 | Layer 2 / 3 |
| IOT | Internet of Things |
| SSC | Shared Service Centers |
| NE | Network Element |
| ePC | Evolved Packet Core |
| API | Application Programming Interface |
| ELK | Elastic Search |

Table ‑1: Acronyms

## Document Organization

|  |  |
| --- | --- |
| Section | Description |
| Section 1 | Introduction |
| Section 2 | Solution Overview – describes the solution objective, lab under consideration, high level scope, constraints, assumptions etc. |
| Section 3 | Functional Architecture – describes the functional view of the solution explaining the features of the different functional blocks |
| Section 4 | Interfaces – Captures the internal and external interfaces of the solution |
| Section 5 | Message and Function Workflow – Captures the flow of different user stories and procedures |
| Section 6 | Data Design – covers the detailed data view of the solution |
| Section 7 | Non Functional Design Considerations – captures the non-functional requirements |
| Section 8 | Detailed Software Design – Links to the SAD for different squads |
| Section 9 | Development Environment – captures the development environment and guidelines for development |
| Appendix | Open Issues |

# Solution Overview

## Solution objective

Modern development processes require continuous integration/deployment for accelerating the time-to-market. This has made process automation an imperative. Mostly, the sophisticated and expensive tools/equipment’s required for testing (product under development, test tools, infrastructure, software, and others) are located in silos and are not optimally utilized. Another typical challenge in a conventional lab is of automating the access, management and maintenance of test setups and equipment.

The answer is to create a self-service, on-demand, automated and flexible environment so that manual intervention for different tasks can be reduced.

A typical next-generation lab is modelled across the lab data, processes and the lab users. The diverse data sources are collated in a structured database through the Configuration Management Database (CMDB). Based on the CMDB data, lab users and admins can access different services, generate tickets, collaborate effectively amongst each other using the different applications like Incident Management, Asset Tracking and Inventory Management, Change Management, etc.

The solution architecture document shall detail out the various services DT aims to offer as part of LaaS solution. The overarching objective is to integrate the existing systems that are either working in silos/sub-system in a seamless manner so-as-to provide an enhanced and intuitive experience to lab users and create a more efficient and effective mechanism

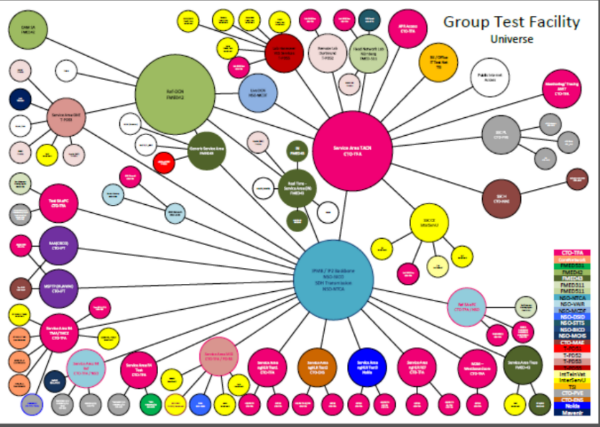
## Labs under consideration

The lab under question is a varied telecom lab spread over different locations catering to different telecom technologies like Fixed Access , 4G, 5G, IMS, IoT etc.

The vision is to present a “One Lab” unified solution to the lab users to address all their lab needs. The following 3 labs are currently under consideration.

### Bonn Lab

The below picture captures the GTF universe for Bonn Lab where the different bubbles present the different service areas supported.



Some salient facts/features are captured beow:

* Bonn Lab is directly responsible for nodes marked in MAGENTA (CTO-TFA)
* IPMB is the backbone network in Germany and Bonn Lab uses the services provided by them
* BK/Office IT Test Net responsible for providing the PC (Desktop) connectivity for employees
* Monitoring/Tracing Service Area hosting the monitoring systems
* SSC- are shared service centres in different locations like Hungary, Czech Republic having different technology equipments like ePC etc.
* APR Access Managed by Bonn Lab for Virtualized N/W Access
* Fixed N/W Nbg. Lab through which the CRD Portal (for external vendors) access the lab services, also the LAB RAN sites in Nbg connect with Bonn MSAN
* Lab Hannover VES services responsible for different voice systems
* Ref-DCN FMED42 through which the OAM SA connects with SA TACN
* Test SA ePC - EPC equipment managed by Bonn Lab
* RAA for RAN Transport - CISCO, HUAWEI
* Service Area R4 for 3G voice (Main,reference and Test)
* Service Area Mobile Internet
* Service Area HLR - Test 1, Test2, Test 3 (/Nokia), Ref
* M2M Westbound core

### Nuremberg Lab

The Nuremberg lab is focussed on the testing of the following areas:

* Fixed Net Products

Retail – xDSL, FTTx, Wholesale

Business- DCIP, EVPL, Wholesale

* Additional Products

Hybrid Access – Connecting Mobile and xDSL

* Interworking with Tel-IT OSS

Common Testing – OSS, FF and ASR

Following NEs are managed and tested:

* CPE
* Access Nodes – MSAN DSLAMs
* BNG
* IMS
* NG-TV Platform
* Plattformsteuerung
* Backbone/Transport

### Other Labs

<To be updated >

## User Personas

As part of overall LaaS solution, the following key user personas have been identified, details of which is provided below

|  |  |  |
| --- | --- | --- |
| Name of the Persona | Brief description | What benefits user intends to accomplish from LaaS |
| Lab Guest | Any person thas has interest in exploring, investigating or subsequrntly using central lab for testing. Lab guest becomes lab customer post successful registration | Lab guest has found some of services being offered fulfilling the business requirements |
| Internal Lab customer | Any employee of DT Group having DT account and pass DT account validation | User want to get online, selfconfigure services from lab product to make tasks efficient and easy |
| External Lab customer | Any external person not holding DT account, but supporting DT project or activity as external partner, external workforce or vendor | User want to execute test activities on hybrid environment, using proprietary set of tools, check integration with rest of environment and get acceptance in shortest period of time, following DT quality and security standards |
| Lab Portal | Custom made portal, supported by backend systems, helping any lab user to execute complex task on a guided manner | Make routine tasks easy and understandable, guided by solution, verification built in to help users access all information without any delay |
| Lab Admin | Suporting person behind central tools, that will handle customer requests | Any request coming from end customer, internal or external will be managed using central platform, to reduce delays and current complexity |
| Lab Change manager | Suporting person behind planned changes in the lab and handle alligment with customer projects | Any request coming from end customer, internal or external, will be managed using central platform as planned change , to reduce delays and current complexity |
| Lab Incident manager | Role is managing incidents in lab environment | Support automatic detection, resolution and reporting for incidents using agreed incident management process flow |
| Lab Announcer | Role is announcing changes in lab environment | To have all activities in sync , announcements are delivered on time, with good quality (informative and understandable) to lab customers |
| Gard Expert | Gard expert is configuring GARD system | GARD is selected as central ticketing tools for customers |
| DevOps Engineer | Role is executing automated or manual test cases on lab equipment | Lab services will support CI/CD pipeline, test cases automatic detection and log collection |

## Solution scope

### In scope

### Out of scope

## Assumptions

1. The solution shall support the following web browsers – Chrome and Firefox
2. As part of the solution, only existing tools being used by DT and/or any new open source tools shall be considered (no investments in tools/license cost)
3. As part of MVP, approx. 120 user stories (out of 300) across Portal, Monitoring, LaaS functionalities shall be considered

## Constraints

<To be updated>

# Functional Architecure

The below section captures the high level functional decomposition of the Automated Lab-as-a-service solution. These building blocks interact with each other and the existing DT systems to provide an overall lab as a service to the end user. The following sections detail out these blocks along with their features.



Figure 3.1 Functional Blocks of Automated LaaS

## Lab Portal

The Lab Portal is intended to serve as a single unified medium for the various lab users accessing the lab resources and services in different ways. It is the interface through which the users of the lab interact with the Automated LaaS for fulfilling their connectivity requests, capacity demands, requests for allocation of different test resources, raising or reporting any issues, extending the allocation request etc.

It is also a means to insights into all important aspects of the entire solution via reports and dashboards. It can be also be used to access historical data regarding any new configuration applied to a managed resource and thus act as an important aid in resolving critical lab issues quicker and more efficiently.

It also includes an appropriate medium for collaboration amongst the different lab users facilitating information sharing, broadcast/multicast, etc. All these channels help in streamlining the communication process and promotes the agile way of working.

Some of the portal features are as follows:

* Registration of users
* Search for information
  + Based on keywords and other advanced filters
  + list of resources along with their location and other relevant metadata
  + list of documents including HLD, LLD, User documents for different Lab resources
  + search for connectivity related information including the preferred and availability connectivity options
  + Download of all the above information for reference
* Login on to the portal with authorized credentials
  + Check the status of any raised request
  + View any notifications subscribed for
  + Logging of the activities
* Raise service requests to reserve resources (connectivity request, capacity demands, blueprint activation)
* Raise change requests to demand a change in the existing lab infrastructure
* Raise requests to report a problem and track its closure
* Lab admin to support handling of service, incident and problem requests raised by the lab users
* Monitoring the health of the lab NE, network
* Check the status of any raised request
* Collaboration
  + Sharing announcemens and important updates about the lab

## Inventory Management

Inventory management provides for managing the resources in the lab. The resources include all hardware, software and network assets in the lab. It brings in all the lab assets in the Central Inventory Database which can be further accessed by the portal for servicing any user requests.

The major functions of inventory management are as follows:

* Asset discovery: Scan, discover and update assets
* Asset lifecycle management: Contracts and license agreements
* Asset reporting and alerting: Triggering the procurement and reconciliation

Based on the above functions it can be logically divided into the below submodules:

### Discovery

To keep a tab of resources at any point of time is a very time-consuming task. Services exist today to automatically track and add resources to the database.

If a new asset is added in the lab the discovery module should be able to export the information from that system and update the Central Inventory Database. Similar changes can happen at the time of deletion of assets.

Additionally it shall also provide ways of importing and exporting Data from an existing inventory system.

### License Management

This module will be typically responsible checking the license validity and filtering the end of life resources and perform any database reconciliation.

### Health Check

All the resources added in the central database need to be monitored continuously so that the inventory database can be kept up to date and only the trackable resources are available for blueprint modelling to the end user.

## User Management

The user management provides a controlled access to the users based on their rights and profiles. It is responsible for provisioning, authenticating and authorizing different types of lab users to access the lab services

### Authentication

Authentication can be managed via a variety of credential management systems. The internal users can be at the first level identified via the registered/known domains and then authenticated via their configured credentials in AD.

### Authorization

Authorization is implemented through different user roles defined in the user database.

This module can be integrated with an already existing database where user roles can be defined or mapped for each user persona. The portal view presented to the user is decided based on the user role.

The user can also subscribe for notifications to be informed of any important and changes of interest based on Network Elements, Service, Service Area etc.

Additionally, it can also be integrated with the user management of any existing monitoring or other tools used in the lab to provide a unified Single-Sign-ON experience to the end user.

## Service Management

This is where the end-to-end delivery of lab services are managed and is responsible for Incident, Problem and Change Management.

### Incident Management

In a telecom lab the incidents/tickets can be primarily around the following areas:

* Enabling access or reserving some test equipment for use
* Creating and Provisioning specific test-setups that range from standardized setups to more project-based and specialized configurations

These services can be bundled together as part of a Service Catalog. This can be further referred by the lab users to request for any specific test or group of equipment.

Service Catalog provides the users with the ability to login and order items from the lab. These catalogue items can include devices, services, configurations.

After booking a test or other service, its progress can be tracked by the requester. Fulfilment groups complete a predefined series of tasks in a fulfilment process that includes required approvals. Lab Admins can review, record history and run reports to improve the service catalogue process.

To handle these requests typically a service desk is deployed. It is intended to be a primary point of engagement between the lab users and the lab admin. It manages incidents (service disruptions) and service requests (routine service-related tasks) along with handling user communications for things like outages and planned changes to services.

The process begins when a lab user places a test booking or similar service request from the defined service catalogue. Once a service is requested, it may automatically be approved or alternately routed for approval, since occasionally services are offered that still require management or financial approval, or even approval from a compliance perspective.

After a requested service is approved, or when no approval is required, the request must be assigned to the appropriate lab admin for review, and ultimately, fulfilment.

For a typical lab test service like running a VoIP test, it may interact with the Resource Manager to allocate the desired resources for this test followed by triggering the automation layer to provision the allocated resources needed to set the pre-requisites for the service.

### Problem Management

Problem management is the set of processes and activities responsible for managing the lifecycle of all problems that could happen in the lab.

Resolving a problem means fixing the error that will stop these incidents from occurring in the future. While Incident Management deals with fighting symptoms to incidents, Problem Management seeks to remove the causes of incidents permanently from the lab.

This module should support the recording of problems, create knowledge from problems, request changes, assign to appropriate groups, escalate, and manage through to resolution and reporting.

### Change Management

Change management application provides a systematic approach to control the lifecycle of all changes, facilitating beneficial changes to be made with minimum disruption to the lab services.

Typically change Management will be responsible for any scheduled upgrade required on any test device.

## Capacity Management

This block is used to support the capacity demand requests from the user. This shall be used to access if the capacity demand request can be serviced by the current infrastructure or the capacity extension needs to be triggered. Once the capacity demands has been accepted the user will be notified by email or other suitable channels about the same.

## Central Execution Engine

This is the central block which is responsible for instantiating the connectivity, blueprints and the E2E environment including all the key configurations.

### Reservation Manager

This module is responsible for allocation of resources as selected in the blueprint to a particular request/user.

### Blueprint Orchestration

This module is responsible for orchestration any blueprint modelled by the lab user.

It is an act of mapping the blueprint to actual concrete resources and setting up the connectivity between them. Once the set of resources to be used is found and reserved, necessary configurations can also be triggered towards them by invoking the automation engine.

These policies may invoke various workflows, including tasks that need to be done manually like physical cable management. Under control of the workflow system, emails can be sent to a configured set of users to perform required tasks and react to when that task is complete to continue the automated setup of the environment.

This layer can use drivers to control layer 1 and layer 2 switching and routing infrastructure to

setup the desired connectivity.

### Virtual Resource Manager

The DT lab is a hybrid lab and resources can either be physical or virtual over the cloud. This module is responsible for orchestrating any virtual resources included in the blueprint. It can also control different hypervisors and orchestrators, including OpenStack to instantiate virtual resources and their associated networking. It is also responsible for managing hybrid topologies, setting up the connectivity between physical resources in a lab and virtual resources in a cloud.

### Connection Manager

The connection manager is an important sub-module intended to provide connectivity solutions between multiple location labs. It is responsible for managing the connectivity journey for the end user. It may include following tasks:

* IP address assignment and management
* VPN alignments of connection
* Security alignment – Firewall configuration and security design
* Routing request- L3 routers configurations
* Switching request –L2 switches configuration
* Cabling

As mentioned in the blueprint orchestration this can also invoke defined workflows to schedule sequential manual and automated tasks for delivering the connectivity to the end user. It can also issue notification and alerts to keep the user up to date about the request status.

## Monitoring Management

This module is responsible for monitoring of any network infrastructure, services and connections in the lab. In the southbound it can connect with different monitoring tools via their exposed APIs to gather the monitoring related information for different activated setups, connections. This information can be stored in the internal monitoring and reporting database, which can be further fed to the analytics engine for KPI reporting.

As soon as the blueprint is activated the monitoring management module is responsible for tracking the health of the different assets/resources used in the blueprint.

At a broader level the monitoring can be further classified into the following categories:

### Connectivity Monitoring

The lab consists of multiple network segments (Service Areas) and are connected via critical routers, switches. When a connectivity between two systems is requested, it is very much desired to monitor the health of the connection. The focus of network connectivity is health check of the Routers/Switches/Firewall constituting the e2e connection using existing systems (Like Solar winds, checkMk).

### Network Element Monitoring

This module is responsible for monitoring the health of individual network elements. Typically these devices generate some PM/FM counters and traps which can be analysed using an analytics framework for KPI calculation and monitoring insights.

### Service Monitoring

Service Monitoring is typically needed for tracking the availability of an e2e service like VoIP, IoT etc. It encompasses a group of connected NEs each serving its intended functionality. Failure or non availability of any one of these NEs in the chain may be impact the overall service.

One possible way of monitoring the service is continuous tests running over the e2e service and then analysing the percentage of failures. This can be treated as an indicator of the service availability.



## Analytics Engine

This module is responsible for ingesting data generated by the different systems (internal and external) and processing the collected data for generating meaningful insights for management users regarding the lab utilisation, availability etc.

### Data Ingestion

The different network elements and tools generate a huge volume of data either through syslogs, SNMP traps, PM, FM counters, configuration changes etc. These data sources are diverse and so is the format of data generated by each of them. These data can be bought to a centralized storage using standard plugins like filebeat, packetbeat, wireshark probes etc. This data can further undergo a data cleansing process via opensource tools like logstash and be fed to a scalable, fast data storage like Hadoop, ELK etc. The reporting database and can be visualized using an opensource interface like Kibana, Graffana etc.

### Data Processing

The data processing is responsible for co-relating the diverse datasets and generating useful data insights. These insights can be derived using BI analytics tools. Further machine learning models can also be deployed for predictive fault detection and Root Cause Analysis. These models are tuned and enriched over the time with data generated by the NEs and sytems.

## Workflow Engine

The workflow engine is responsible for implementing the business rules and processes required for servicing a user request. It is responsible for controlling the lifecycle of a service request received and invoking the different sub-systems for processing the request. It is also used to co-ordinate the tasks allocated to different executors based on the workflow defined.

## Automation Engine

Automation is required for the efficient functioning of the lab with minimal human intervention. This layer takes care of such tasks like resource/asset provisioning; and adding analytical decision-making capabilities to the lab.

### Backup and Recovery

This module is responsible for backup and recovery of test configurations and setups. Through the portal automatic Chef, Ansible scripts can be triggered which takes care of these configuration management tasks.

### Op-Task Automation

This takes care of the automatic provisioning of network resources required for services. It shall be very tightly coupled with the services defined and will be interacting mostly with the domain specifics. This, in turn, enables to improve the productivity and efficiency of lab usage by isolating the lab complexities from the end user.

## Content Management System

Content Management System is responsible for managing the lab related information. It is extremely important and relevant for collaboration amongst different lab users. As part of this documents like HLD, LLD, User documents can be maintained for different service areas. Based on the lab users roles these documents can be referenced and accessed for easy retrieval of information.

## Search Engine

### Query Parsing

This is responsible for query refinement and processing. Most queries are simple keyword queries but complex natural language based queries may also be supported through a NLP module plugged inside the search engine.

This may also include query transformation techniques like tokenizing etc.

### Indexing

Intutive indexing allows ease of search and retrieval functions for the data managed by content management systems. Indexing process collect, parse and store data to facilitate fast and accurate information retrieval

Documents inside the CMS are indexed based on keywords, titles and content inside

### Cache

Search cache is typically used to preserve the user specific searches and presenting similar view in case of next login.

## Test Automation

### Test Automation Framework

Test Automation Framework is through which users can create, automate and scale their tests for different services. It provides a platform to create business level test cases/scripts which are further translated into domain specific methods and procedures for the System Under Test (SUT).

It also provides a framework to execute tests on demand or scheduled periodically suitable for regression and load testing.

### Automatic Verification

This is responsible for automatic verification of executed tests based on logs generated by both the SUT and simulators. The results can be presented in dashboards indicating the quality and stability of the SUT

### Test Data Management

This is where the test data like test scripts, input data, logs etc. are stored.

## API Gateway

The API Gateway is a single unified interface for the backend modules to connect to the portal. It provides an API tailored for each backend sub-system. These subsystems implemented as micro services get an individual view of the APIs from the Portal

## Internal Communication Channels

The internal communical channels/bus are used for messages exchanged between the different sub modules/microservices. There can be multiple communication patterns needed for different kind of functionalities :

Synchronous – Request/Response

Asynchronous – Notifications

Each service may use a combination of one or more of these interaction styles. APIs are defined by all the service through which this interaction takes place. Messaging systems like Rabbit MQ, Apache Kafka, Active MQ can be chosen for Async communication whereas REST is most preferred for Sync mode.

## Logging and Auditing

This is a common utility module for logging and auditing purposes. Based on the technology stack for each sub-module appropriate logging framework can be supported. The general principle for logging to be followed is concise and clear logs controlled by log levels .

## Notification and alerts

Each sub-system is capable of generating alerts which needs to be passed on to the portal so that the same can be viewed by the lab user. These notifications can also be delivered to lab users email by integrating the portal with an email server.

# Interfaces

<Captures the interfaces internal and external for the solution>

## Internal

## External

# Message and Function Workflow

This section captures the flow of events, messages exchanged between different subsystems to service a particular user story/service.

<To be updated >

## User Registration

## Connectivity Journey Management

## Connectivity Monitoring

# Data design

<Captures the data layer of the overall solution with a high level view of the existing and new data sources needs >

# Non Functional Design Considerations

## Scalability

## Reliability

## Modularity

## Security

## Flexibility

# Detailed Software Design

## Portal

## LaaS

## Monitoring

## Networks and Operations

# Development Environment

<To be updated>

# Appendix

## A.1 Open Issues