Open Net BSS/OSS Functional Requirements Specification

N(N)LE OPEN NET

2024

Business Support Systems (BSS) Functional Requirements Specification

1. Business Support Systems (BSS) Functional Requirements Specification

1.1. High Level Functional Requirements

1.1.1. BSS solution shall include the following modules / applications:

- Customer Information Management module, that support functionality to manage customers, their accounts, and subscriptions
- Customer Order Management functionality that the support the capture, handling and fulfillment of customer orders
- Product and Service Catalog functionality that supports the management to define and manage all services, products and their offerings provided by Open Net
- Rating and charging functionality that supports all periodic and one-time fees applicable to Open Net wholesale service model
- Billing and invoicing functionality that supports Tax Invoices in accordance with the applicable Georgian legislation
- Functionality supporting the settlement of invoices, including handling incoming payments from customers to cover their outstanding bills

1.1.2. Solution shall support integration with the following modules/applications that are external to BSS:

- Network and Service Resource Inventory
- Service Provisioning functionality
- Problem Management functionality that supports creating service tickets and accompanying business process of handling and tracking of service tickets
- Integration to the Enterprise ERP system available at Open Net

1.2. Customer Information Model

1.2.1. Solution shall support a notion of Party, which is an entity that Open Net has a business relationship with

1.2.2. Solution must provide capabilities to define what is the Type of Relationship of the party with Open Net

1.2.3. In case of Open Net there are at least two Party Relationship Types (or Roles) to be supported: Customer and Interconnection Partner

1.2.4. System shall support the notion on Account, which is the representation of a business relationship with customer or party of any type

1.2.5. System shall support a notion of subscription. Subscription is a service instance the customer is subscribed to. One customer may have 1..n accounts and one account may have 1..n subscriptions

1.2.6. Parties that maintain relationships with Open Net are legal entities and solution shall support following attributes for Party entity:

- Name of the party
- Organization legal form (e.g., Limited Liability company, State Entity, Non-profit etc.)
- Organization contact persons (one or more)
    - Role of contact (e.g., Director, Account manager etc.)
    - Contact name
    - Contact information (e-mail, phone etc.)
- Tax identifier - a 9-digit Georgian tax identification number
- Banking account information
    - Bank code (SWIFT Code) and IBAN
- Organization legal address
- Organization factual address (if different from legal address)

1.2.7. Addressing information database shall be at least compatible, or in best case the same one used in the Network and Resource Inventory database of the OSS system

1.2.8. Solution shall maintain the system of identifiers for unambiguous identification of all entities within the Customer Information model. Therefore, each Customer or Party, Account and Subscription shall have its own ID that is either numeric or alphanumeric

1.2.9. Customer Information Model shall contain a free-format text field in Customer/Party, Account and Subscription tables

1.3. CRM – Customer Relationship Management

1.3.1. Solution shall provide a graphical user interface (GUI) to perform Customer/Party Information management with the following functionality as a minimum:

- Create, Modify and Delete Party information
- Provide functionality that ensures avoidance of duplication of party information
- Provide capabilities for searching party information by combination of important parameters, such as:
    - Party ID, Party Name
    - Relationship type (Role)
    - Tax Identification Number
    - Subscription information (Product, Service)
    - Subscription information shall contain service expiration date

1.3.2. Solution shall support storage and retrieval of copies of the documents (e.g., PDF files) related to the party and accounts associated with the party

1.3.3. Each document shall be marked with a tag describing type/purpose of the document. Examples of such tags are "Contract text", "Amendment to the contract", "ID of contact person", etc.

1.3.4. A party may have different roles (relationship types) with Open Net for each account (e.g. a telecommunication organization that leases 2 fiber cores from Open Net but at the same time Open Net leases fiber from this operator elsewhere)

1.3.5. System shall allow different accounts for services of the same type (e.g. 2 accounts with 4 dark fiber lease services each)

1.3.6. System shall allow different services for the same account (e.g. 3 data transmission services and 2 dark fiber leases on the same account)

1.3.7. BSS solution shall offer the capability to track customer premises equipment (CPE) that may be installed at customer premises for the purpose of fulfilling customer order and is the property of Open Net

1.3.8. BSS solution shall provide functionality to view the list of such equipment; replace already installed equipment with a new one and support returning the CPE to Open Net in the case the corresponding service is decommissioned

1.3.9. CPE movement functions shall be integrated with the Open Net ERP system

1.3.10. The solution shall display all current product subscriptions, as well as the history of past products subscriptions

1.3.11. The solution shall display all Product Subscriptions for this account, both with pending and fulfilled orders, as well as the offers under which these products were subscribed to

1.3.12. Solution shall support the display of Customer/Party's billing-related data, such as payment history and invoice history

1.3.13. Solution shall support display of all customer orders for given customer

1.3.14. Order display functionality shall provide general capability to filtering customer orders. Possible filtering parameters are: Order status (Fulfilled/Outstanding), Order date (creation date, fulfillment date etc.), Order type (New Connection, Change Connection etc.)

1.3.15. Solution shall support Problem Ticket display functionality that will be available through of a dedicated CRM screen

1.3.16. It shall be possible to filter the list of Problem Tickets by various filtering parameters, such as problem status, problem creation date, etc.

1.3.17. Solution shall support the definition of Responsible persons/Account managers at Customer/Party and Account level

1.3.18. Responsible persons/Account managers are Open Net employees. Solution shall allow standing-in user definition for the cases when primary account manager is unavailable

1.3.19. System shall allow sending some Customer dunning notifications to Account managers as well

1.4. Customer Order Management

1.4.1. Solution shall provide the capability to capture following Customer Orders:

- New Connection Order, that represents new or existing customers' willingness to obtain one or more Products described in Open Net Product Catalog
- Change Connection Order - willingness to change an existing service instance, such as increasing or decreasing capacity of existing data transmission service
- Suspend Connection Order - Customer's willingness to temporarily suspend, or pause, a service instance. This shall result in the pausing of all or some charges being applied, as described in Product definition
- Resume connection Order - this will resume previously suspended connection, basically reversing the previous effects of suspension on charging and provision of the services
- Cancel Connection Order - Customers willingness to decommission a service instance

1.4.2. Apart from product data, New Connection Order shall contain the information about locations at which the Product chosen from the Product Catalog is to be delivered

1.4.3. Solution shall support all services currently offered by Open Net. Currently these services are the following:

- Point-to-point Dark fiber rental from location A to location B
- Point to Point data transfer service from location A to location B
- Point to Multipoint data transfer service from location A to multiple locations

1.4.4. Customer Order Management function shall support the checking a feasibility of providing services described in ordered products at desired locations

1.4.5. Proposer shall provide the explanation and descriptions of the feasibility check algorithms the system will employ in the process of handling Customer Order

1.4.6. Feasibility check shall consult OSS, or more specifically, Network and Service Resource module, to establish whether there are sufficient resources available to support the creation of new service instance at the requested locations

1.4.7. In the case of the availability of the requires resources, OSS Service Order Management (SOM) function shall perform reservation of required resources in Network Resource Database and respond back to Customer Order Management that feasibility check is positive

1.4.8. Conversely, if resources for the desired Product at the requested location are not available, negative feasibility check result shall be returned and the Customer Order will be rejected

1.4.9. After positive result, Customer Order shall move to execution phase, which means Customer Order will be moved to OSS Service Order Management (SOM) function for fulfillment.

1.4.10. Solution shall support both immediate and scheduled execution of the Customer Order

1.4.11. OSS SOM module shall report back to BSS Customer Order Management the result of Service Order fulfillment

1.4.12. Customer Order Management function shall instruct Charging module to execute charging logic as described in Offer definition (see below for Offer definition), provided that the Customer Order is fulfilled

1.4.13. Solution shall support cancellation of pending Customer Order before the order is fully fulfilled. After the order fulfillment, the only way to cancel the subscription is to issue a separate cancellation order

1.4.14. Solution shall support changes to the ongoing Customer Order. This applies to New Connection and Change Connection orders before the orders are fulfilled

1.4.15. It is obvious that the Order Change functionality will have some limitations depending on the current stage of the original order. Proposer shall provide a clear explanation of these limitations

1.4.16. Solution shall support automatic customer notifications generation and sending when Customer Order reaches predefined stages:

- Customer Order is received and validated
- Customer Order Management finishes feasibility check. In this case, customer shall receive the result of the check
- Customer Order is submitted for fulfillment. In this case, solution shall provide capability to specify fulfillment date at least manually
- Proposer is compelled to offer functionality that automatically calculates order fulfillment date, in which case it shall supply description on the algorithm
- Customer Order is fulfilled, and charging started
- Solution shall support manual entry of notifications if Open Net shall deem necessary to send additional notifications during Customer Order Fulfillment

1.4.17. Notifications shall be sent to Customer contacts either by SMS or email, or both

1.4.18. Content of the notifications shall be configurable by Open Net users

1.4.19. Notification system shall be flexible enough to allow excluding some customers from some types of notifications

1.5. Product Catalog/Product specification

1.5.1. The system shall support Product Catalog functionality that contains the specifications of all products available to Open Net customers

1.5.2. Product specification shall include: At least one service, which is a Customer Facing Service (CFS) that is further mapped to one or more Resource Facing Service (RFS) in Service Inventory application of OSS system, and Types and quantity of Customer Premises Equipment (CPE) required to fulfill the Customer Order

1.5.3. Product specification shall allow to identify parameters of the Product that will serve as basis for charge calculation, that includes: CFS parameters, such as data transmission rate, distance between two points, and:

- Types of required CPE that will be taken into charge calculation
- Solution shall be able to distinguish between the use cases when CPE is rented or purchased by Customer

1.5.4. Solution shall provide a capability to support Product Offers. Product Offers define:

- How products are sold. Products can be sold as individual products or as part of product bundles. In this case, an offer is called a Bundled Offer
- How products are charged for and at what price. Charging specification describes all charges applicable to the products included in the Offer.

1.5.5. Solution shall support both one-time charges and recurring charges (e.g., monthly subscription fees)

1.5.6. Solution shall take Customer tax status into account when calculating charges

1.5.7. Solution shall provide flexible mechanism for calculating tax amounts that as a minimum, shall support the following taxation statuses:

- Customer with VAT
- Customer exempt from VAT obligation
- Customer with 0-VAT status

1.5.8. Tax calculation system shall be configurable via GUI and. Proposer shall explain its functionality in detail

1.5.9. The System should have easy-configurable (via respective GUI) functionality for taxation purposes, for example (including, but not limited to):

- Ability of the System to apply different taxes/exemptions to specific charge items.
- Ability of the System to apply different taxation rules to specific customers.

- Ability of the System to apply different taxation rules to specific charges / types of usage / services.
- Ability of the System to apply taxes calculated in different ways: absolute amount, relative percentage, and their logical mix

1.5.10. Tax rates should be kept in the form of hierarchy and assigned to clients depending on the tax category assigned. The System operator, who has the authority to make such changes, can change tax rates assigned through the category for individual clients.

1.5.11. Charging function shall support the flexible definition of discounts for all types of charges, such as one-time charges, monthly fees and other recurring charges

1.5.12. Supported discount schemes shall span multiple Accounts within a Customer/Account hierarchy and allow for flexible configuration of the discount eligibility and distribution logic

1.5.13. System shall support flexible schemes of applying discounts, such as:
- Flat amount/percentage discounts
- Tiered amount/percentage discounts

1.5.14. Solution shall support quoting of ordered services according to the configuration of the offers

1.5.15. Solution shall support manually changing automatically generated/quoted prices by Open Net staff

1.5.16. It shall be possible the export the quote to PDF file for sending to the ordering customer for evaluation and decision making

1.5.17. Content and layout of the quote PDF file shall be configurable

1.5.18. Solution shall provide the capability to create and modify charging and discounting plans using system GUI that is serviceable by Open Net users with sufficient qualification and training

1.5.19. Solution shall support Product Offer and/or Discount assignment in bulk mode, i.e. to a group of Customer/Accounts confirming to user-defined criteria

1.5.20. System should have possibility to present all discounts on the bill

1.6. Balance management/Payments

1.6.1. System should allow to manage a payment into several smaller payments (partial payment support) and split them across several invoices or accounts. For example: the customer has several monthly bills due, later it pays a large amount which covers most of these dues. The solution should have capability to split this amount to all pending invoices (on due date sequence order)

1.6.2. System should support configurable payment allocation rules

1.6.3. The system should have ability to prioritize the invoices and distribute the payment based on it

1.6.4. Each payment record should show the reason for the payment. For example, debt coverage, balance, service charge payment etc. The list of payment reasons shall be configurable by using system GUI

1.6.5. Open Net customers shall be able to cover their outstanding balances using bank payments

1.6.6. System shall support both manual entry of incoming bank payments and automatic import from an external source, which may be direct automatic interface with banking service provider

1.6.7. For Open Net, banking service provider is Treasury, which is an independent, direct participant of the existing Georgian interbank payment system. Solution shall implement interface with Treasury system to download and import Open Net incoming payments

1.6.8. The System shall provide a report that should enable reconciliation with the banking service provider system on daily and monthly basis.

1.6.9. Solution shall support refunds of overpayments.

1.7. Balance management/Adjustment

1.7.1. The solution shall support balance adjustments (both positive and negative). Adjustment subsystem should support the following possibilities:

- Adjustment with different tax combinations and tax rates
- Implementation of different adjustment type and its reflection in the invoices – adjustment of services and adjustment of balance (payments) should be stated separately
- Adjustment attributable to customer account in total, as well as to each individual subscriber
- Adjustments of the fixed one-off payments
- Adjustments of individual line items of generated invoice
- Adjustments without binding to the invoice

- Funds transfer (remaining of balance) from one customer to another (excluding taxes), i.e. balance clearance in favor of another customer.
- All mentioned adjustment should be possible to load from file with rollback option

1.7.2. Solution shall support making an adjustments to a specific invoice charge, or the whole invoice.

1.7.3. Solution shall allow the user to reverse an adjustment action that was already completed.

1.8. Billing and Invoicing

1.8.1. It must be possible to configure through GUI (without coding) the following dates for each bill cycle:

- The date of closure of a billing cycle
- The due date for the payment of the bill/invoice

1.8.2. The System should support intermediate billing for creating intermediate charges for any time period.

1.8.3. Solution shall support canceling an active billing procedure, for any reason, and at any time of the procedure should be possible. At that point the operator should be given the choice to either continue or rollback.

1.8.4. Solution shall generate alarms and/or notification in case of error during billing cycle or in case of billing failure

1.8.5. The System must have the ability to reverse ("undo") the Billing process and the recalculation of the bill once the problem has been rectified. This reversal option can either be used for the whole cycle or selectively for specific accounts.

1.8.6. System should support the ability of immediate invoice calculation (billing) at any time before the end of the current billing period, by customer request for the case when customer would like to cancel its contract wholly or partially and would like to pay his Invoice up to the date of cancelation.

1.8.7. The System should support invoices containing all charges made during a billing period, including the following items at a minimum (including, but not limited to):

- Customer and account details
- Organizations
- Billing period for which an invoice is issued
- Amount due
- Amount charged for each type of service
- Types of services provided
- Date and amount of the last payment

- Invoicing date
- Payment due date
- Penalties or fines
- Customer credit and debit adjustments

1.8.8. The System should enable different types of invoicing components to be distinguished, i.e., recurring charges, one-time charges, discounts, and credits.

1.8.9. The System should support creation custom rules for aggregation invoice items information on invoices by combining different parameters (charge types, services, zones, etc.)

1.8.10. System should support detailed charges invoicing

1.8.11. For all charges displayed in the invoice, calculated taxes that apply to invoice items shall also be displayed. At present, applicable tax is and it is calculated in accordance with the customers tax status

1.8.12. Multiple languages for invoices should be supported and configurable by Open Net. The default invoice language should be Georgian

1.8.13. The solution should provide a toolkit for creating and the editing of invoice template. This should be a tool with GUI, which allows a simple design (via drag & drop capability, for example)

- The solution should enable modification of invoice template and reflection of data entered by billing solution administrator for example:
- Modification/deletion/addition of static text, picture or graphical objects
- Modification of information format: font format, table format
- Modification/deletion/addition of data on charge types
- The solution should allow to create/delete/edit the invoices of new sections, establishment of information presentation rules for the sections, section presentation order in the invoice.

1.8.14. The System should provide a mechanism for charging late fees on overdue amounts.

1.8.15. If the customer misses the payment due date, the late fee could be applied to the next customer periodical invoice. The late fee should be calculated by the billing system as a fixed percentage of the invoice amount, for the number of days in delay and be limited by maximum penalty value.

1.8.16. The solution should be able to activate/deactivate late fees per customer and account

1.8.17. As in the most contracts accrual and collection of late fees is a right of the operator and not an obligation, it should be possible to exclude late fees, if opted to do so.

1.9. Billing and Invoicing/Collection and dunning

1.9.1. The System should be capable to automatically initiate dunning processes at the end of a configurable period of time following the invoice generation (e.g. 5th day of each month)

1.9.2. The System should support the definition of an of dunning plans, via (including, but not limited to) a user-friendly, GUI-based tool.

1.9.3. The System should support the definition of an unlimited number of steps within a dunning plan, such as customer reminders and barring.

1.9.4. The System should have the capability to manage and produce various reminder communications (i.e., emails, SMS messages, etc.) at configurable intervals in the event of non-payment.

1.9.5. Dunning messages shall be sent to the Customer Contacts determined by their role in relationship with Open Net

1.9.6. Dunning functionality shall support automatic issuing of Suspend Connection Orders as defined in dunning plan

1.9.7. Conversely, the solution shall allow to automatically issue Resume Connection Orders if the customer covers the obligation for which the suspension order was issued

1.9.8. Some dunning policies shall allow automatic cancelation of customers connections if the outstanding obligations are not covered for an extended amount of time

1.9.9. Dunning policies and plans shall be configurable at system level with the possibility of full or partial override at Customer or Account level. For example, a one customer in arrears may receive dunning notifications only whereas another customers' service might be suspended automatically

1.9.10. The System should be able to allow customer to negotiate a promise to repay a debt by a certain date, either in a single payment or a series of payments. Promised Payment function shall allow pausing of the dunning actions to be performed by the system until agreed date. If this date is reached and the debit is still outstanding, Promised Payment shall be deleted and the dunning process will resume

1.10. Interconnect Partner Billing

1.10.1. Open Net will rely on the network resources provided by other wholesale operators. Solution data model shall support definition of such relationships

1.10.2. Solution shall support separate Accounts for the maintenance of Interconnect Partner relationships

1.10.3. While the service provider will present invoices for such Accounts, the solution shall support calculating and accruing charges for Interconnect contracts for reconciliation and settlement purposes

1.10.4. Solution shall support entering payments into the system to support covering outstanding invoices for interconnect services

1.10.5. Solution shall export such payments and charges into the ERP

1.10.6. Solution shall support the balance adjustment operations described earlier in this document for interconnect contracts as well

1.11. Integration with ERP

1.11.1. Solution shall provide the capability to export transaction information to the General Ledger (GL) of ERP system of Open Net. It is expected that the ERP already has the corresponding GL accounting plan set up

1.11.2. The items/transactions to be exported from BSS solution for GL accounting purposes are as follows:

- Service Charges
- Tax amounts calculated
- Payments and payment reversals
- Charge adjustments and account balance adjustments
- Refund and write-off transactions

1.11.3. The System should support the ability to send revenues to GL and automatically split based on pre-defined criteria (for example, product, service etc.)

1.11.4. Solution shall support flexible mapping between transaction types and GL transaction types. This shall be configurable by a GUI function

2. Problem management Functional Requirements Specification

2.1. High-Level functional requirements

2.1.1. Objective of Problem Management Functionality is to manage the process of capturing, handling, tracking, and closing of tickets that result from:

- Customer inbound interactions with relevant Open Net customer service and support personnel via dedicated communication channels
- Actions initiated at Open Net (such as planned network outages)
- Emergency network outages

2.1.2. The solution shall offer the following high-level functions:

2.1.3. Ability to configure different types of tickets with categories within the same types of tickets and sub-categories within the same categories:

- Examples of ticket types can be Customer ticket, Emergency network outage ticket;
- Examples of ticket categories include (within Customer ticket): Information Request, Complaint
- Within Complaint, the ticket by be further sub-categorized as Service complaint, Billing complaint etc.
-

2.1.4. Capability to configure and then execute different workflows per ticket type and category, as well as depending on some additional parameters

2.1.5. Functionality to create ticket based on information captured

2.1.6. Functionality to modify ticket as more information becomes available

2.1.7. Tracking tickets from opening to closing

2.1.8. Capability to notify concerned/affected customers regarding the creation and progress of the outstanding tickets

2.1.9. Solution shall provide relevant user management capabilities. The users will be main actors executing the ticket handling workflows

2.1.10. Solution shall expose API interfaces necessary for interaction with the external applications/modules such as CRM in BSS and Service Inventory in OSS

2.2. Ticket attributes and Configuration

2.2.1. Solution shall support the following ticket types:

2.2.1.1. Customer ticket - this ticket type will be created when customer addresses Open Net to request information (Information ticket) or report a problem/complaint (Complaint)

- System shall support creating Customer Ticket as a result of outbound communication with a customer representative
- Information Ticket and Problem/complaint Ticket are Categories of tickets. The system shall allow GUI-based method of the management of ticket categories
- Problem/complaint category of customer tickets shall be sub-categorized as Service complaint, Billing complaint, complaints related to payments, invoices, quality of service etc. The system shall allow GUI-based method of the management of ticket sub-categories

2.2.1.2. Planned Network Outage ticket - this ticket type will be created by Open Net personnel before any planned works that may affect one or more services already used by Customers

2.2.1.3. Emergency Network Outage ticket - this ticket type will be created by Open Net personnel when there is a firm reason to believe that part of the network being used by customers is out of service

- Emergency network outage ticket shall have an attribute that describes impact of the outage, which can be: High, Medium or Low

2.2.1.4. Planned Network Outage ticket - this ticket type will be created by Open Net personnel when there is a plan to take part of the network out of service for maintenance and/or expansion purposes

2.3. Customer Ticket Capture

2.3.1. Customer tickets shall be opened when the information about the customer in question is captured. Customer Information Management function in BSS holds all valid customer contacts that are eligible to open Customer Ticket. Therefore, calling party shall produce at least Customer ID value

2.3.2. Solution shall support retrieval of the information from BSS using quoted Customer ID, including customer contact information that may be used by Open Net customer service staff to validate the identity of the caller

2.3.3. Solution shall support retrieval of the information regarding all service instances for this customer from BSS

2.3.4. Solution shall support capturing Account and Service Instance information that is affected by the complaint

2.3.5. It shall be possible to capture additional information about the complaint, such as specific payment, bill, invoice etc.

2.3.6. Solution supports entry of the free text for the user to be able to describe additional information about the customer problem or information request

2.3.7. System shall support captured data validation, such as if quoted Customer ID, Account ID or Service Instance ID refer to valid object. After validation is successful, the ticket shall be submitted to the selected workflow instance for the execution

2.4. Emergency Network Outage Ticket Capture

2.4.1. System shall support creating Emergency Network Outage Ticket, which will be created by Open Net personnel when a network element is out of service. This can be a network element of any type (Passive, Active, Active/Passive network element port, etc....)

2.4.2. Solution shall support the integration with Network and Service Resource Inventory and Service Inventory databases of OSS system. This is necessary to accurately capture ticket data

2.4.3. Emergency Network outage ticket data shall always include a reference of to a network element resource that is thought to be damaged/out of service

2.4.4. Service Inventory Database of the OSS contains all RFS instances that are dependent on the damaged network resources and these RFS instances in turn point to their corresponding CFS instances. Therefore, Emergency Network Outage ticket data capture process shall produce a list of CFS instances affected by the network outage

2.4.5. For each CFS instance, solution shall record date and time of the outage

2.4.6. If identified CFS instances are referenced by open Customer Tickets, solution shall associate Customer Tickets with the Emergency Network Outage ticket

2.4.7. Solution shall support automatic closure of Customer Tickets associated with Emergency Outage ticket based on the discretion of Open Net employee working on Emergency Ticket

2.4.8. Solution shall support the integration with the system monitoring solution Zabbix, that is used in Open Net

2.4.9. As Network and Service Resource Inventory and Service Inventory databases shall be integrated with Zabbix as per OSS system requirement specifications

2.4.10. The solution shall be configured such a way that outages detected by Zabbix may result in automatic creation of Emergency Network Outage tickets

2.5. Planned Network Outage ticket capture

2.5.1. System shall support creating Planned Network Outage Ticket, which will be created by Open Net personnel when a network element, or set of network elements, are to be taken down for network maintenance or expansion purpose. This can be a network element of any type (Passive, Active, Active/Passive network element port, etc.…)

2.5.2. Solution shall support capturing Planned Outage Start Date/Time and Expected Planned Finish Date/Time

2.5.3. Just as in the case of Emergency Network Outage Ticket, solution shall produce the list of affected CFS instances

2.5.4. It shall be possible to configure the solution the way that it will rely on Zabbix network monitoring solution to detect CFS outages when the actual network maintenance work starts

2.5.5. Alternatively, CFS instances affected by the outage shall be considered out-of-service when Planned Outage Start Date comes

2.6. Ticket Processing

2.6.1. Solution shall support defining flexible workflow specifications for handling tickets

2.6.2. Workflow specification shall allow Open Net to handle tickets depending on their basic attributes, such as type, category and sub-category

2.6.3. Workflow specification shall take into account other data, such as affected customer, type and location of the network element concerned etc.

2.6.4. System shall allow definition of groups of system users that participate in the workflows

2.6.5. System shall route tickets to the groups of system users as per defined previously defined workflow. For example, one user group will handle Customer Information Tickets; another - Customer Complaint tickets; other group - emergency network tickets concerning active network devices of specific type etc.

2.6.6. Solution shall create separate workflow instance for handling each ticket

2.6.7. All actions performed by the users within the scope of the corresponding workflow instance will be added to the ticket audit trail

2.6.8. Solution shall support sending reminders to users (at least in the form of email) when their tickets have not been acted upon for the specified amount of time

2.7. Ticket closing

2.7.1. Open Net users shall be able to close the tickets when the conditions that have led to the creation of the ticket no longer exists

- For Customer Tickets, this means that customer complaint and/or information request has been addressed
- For Emergency Network Outage Tickets, this means that the corresponding outage have been fixed
- For Planned Network Outage Tickets, this means that the corresponding works have been finished and service is restored

2.7.2. All affected CFS instances shall be brought back to Active state

2.7.3. Solution shall be capable of accomplishing this by integration with Zabbix network monitoring tool

2.7.4. All tickets shall be closed with a defined closing category that describes how the ticket was resolved

2.7.5. System shall provide the capability to maintain the list of closing category values per each type of ticket

2.7.6. System shall support opening a new ticket based on the closing category values

2.7.7. System shall support functionality linking different tickets based on the information that becomes available during ticket handling

2.8. Customer notifications

2.8.1. Solution shall provide the capability to send notifications to the relevant customer contacts regarding the progress of the ticket

2.8.2. Notifications shall be sent (as a minimum) under following conditions:

- Ticket has been created
- For customer complaint tickets - When the original ticket is associated with Emergency Network Outage ticket
- For Emergency Network Outage tickets - when system determines that the CFS is affected
- For Planned Network Outage tickets - when system determines that the CFS WILL BE affected.
- When a ticket is closed. In this case, the notification text shall be configurable per each closing category value

2.8.3. Planned Network Outage notifications shall be sent at least 72 hours before the actual work start. This value shall be configurable per Planned Network Outage ticket.

2.8.4. Solution shall allow sending extra notifications manually to the selected customer(s), for example to inform of the delays or other non-standard developments regarding their tickets

2.9. SLA management

2.9.1. System shall provide the capability to extract the information from the Problem Management solution that will be useful for SLA calculation purposes

2.9.2. The extracted information shall include (as a minimum):

- All registered CFS outages per customer for reporting period (e.g., given month) including outage start date/time, outage end date/time that were results of Emergency Network Outages
- All registered CFS outages per customer for reporting period (e.g., given month) including outage start date/time, outage end date/time that were results of Planned Network Outages

2.9.3. Report for planned network outages shall also identify the outages for which the actual start and end of the outage are different from the periods identified as start and end date in the original Planned Network Outage ticket

3. Operations Support Systems (OSS) Functional Requirements Specification

3.1. Network and Service Resource Inventory

3.1.1. Location management

3.1.1.1. Solution must be capable of storing and managing locations, that may be identified by location hierarchy information, that should include:

- Country region (e.g., Kartli, Imereti, Svaneti etc.)
- City/Town/Village (when applicable)
- District (when applicable)
- Address hierarchy (e.g., street, block, building number, when applicable)

3.1.1.2. System must support storage of geographic information, i.e. longitude and latitude for the location with the precision of degrees, minutes, seconds and fraction of seconds

3.1.2. Containment facilities management

3.1.2.1. System must support different facility types, which includes:

- Building
- Shelter

3.1.2.2. In case of a multi-room building, the system must have a notion of a room

3.1.2.3. System should support facility attributes, such as:

- Name of the facility
- Cadastral code (when applicable)
- Whether this location is either owned or rented

3.1.3.System should support the capability of attaching different files to the location/facility, which may include PDF files (such as rent contracts, statements from public register, photographs, drawings etc.)

3.1.4.System must support storage of non-technical linear facilities of the network:

- Tranches
  - Trenches should have an attribute that should indicate whether it contains duct pipe or cable is buried directly
  - Trenches should have an attribute that shows depth of the trench
- Poles
  - Poles shall have an attribute that shows height of the pole
- Manholes

3.1.5. Linear containment facilities should have their relevant geospatial information (e.g., for trenches, it should support start and end points)

3.1.6. Inside plant containment devices

- Racks
  - Height (mm), Width (mm), Depth (mm)
  - Heights (in units)

3.1.7. System should support containment facility attributes, such as :

- Identifier (either a name of numeric id)
- Whether this facility is either owned or rented

3.1.8. Solution must be capable of associating containment facilities with the network elements that are located in the facility

3.1.9. The above association shall be bi-directional in a sense that the network elements placed/located in a given containment element, have link to its immediate container

3.1.10. System must support the precise location of inside plant network devices, such as:

- For rack-mounted devices - precise location within the rack
- For free standing devices, or wall-mounted equipment, location should be at room level

3.1.11. Passive Network/Optical ducts

3.1.11.1. Solution shall support the definition of optical ducts that may be by OpenNet as extra protection for underground placement of optical cables. The non-exhaustive list of possible duct attributes are as follows

- Name/ID, Color
- Reference to the trench the cable duct is laid in
- If a duct is not buried in a trench, but follows a structure (e.g., roadside/bridge side conduit), the solution shall support such cases as well
- If a duct is placed within another duct, the reference to enclosing duct
- The depth at which the duct is placed in a trench
- All cables placed in a duct shall have references to their enclosing duct

3.1.12. Passive Network/Optical cables

3.1.12.1. Solution must be capable of maintaining information about deployed optical cable. This information includes:

- Cable type (ITU-T)
- Cable identifier (Name/ID)
- Information about tubes in the cable
    - Core quantity
    - Tube color
- Within each tube, each core should be defined separately
    - Color of the core
    - Identifier of the core (Name/ID)

3.1.13. Passive/Network Optical Splice Closure (OSC)

3.1.13.1. Solution shall support the definition of Optical Splice Closure with at least the following attributes

- Type of Optical Splice Closure (OSC) - Horizontal/Vertical
- Identifier (Name/ID)
- Number of input ports and output ports
- Type of fiber splice trays
- Number of fibers per splice tray
- Number of fiber splice trays
- Identifier of the fiber splice tray

3.1.13.2. Solution shall support definition of cable cross connections within Optical Splice Closure, as well as the modification of cross connections

3.1.13.3. Solution shall support definition and modification of additional splicing details, such as number of spliced/spare fibers, etc.

3.1.14. Passive/Network Optical Distribution Frame (ODF)

3.1.14.1. Solution shall support the definition of - Optical Distribution Frame (ODF) with at least the following attributes

- Identifier (Name/ID)
- ODF and patch panel information.
- Details of the ODF connection/patching/spare.
- ODF device structure (e.g., Device-Shelf-Module-Port, Device-Module-Port, Standalone etc.)
- Number of fibers per splice tray
- Number of fiber splice trays
- Identifier of the individual fiber splice tray
- Total number of ports
- Inventory state (Available/Unavailable)
- Service state (Occupied/Free/Suspended)

3.1.14.2. Solution shall support definition of cable cross connections within ODF, as well as the modification of cross connections

3.1.15. Active Network

3.1.15.1. Solution must support at least the following types of active devices:

- Switches (L2)
- Routers (L3)
- DWDM devices
- UPS
- Other equipment supporting SNMP

3.1.15.2. Solution shall support the maintenance of attributes for active devices, such as:

- Type of device
- Manufacturer
- Management IP address

3.1.15.3. Solution must support structure of active devices, which can be: Chassis-based/Standalone

3.1.15.4. For chassis-based devices, the system must support the definition of Card with the following attributes:

- Type of card - Management/Line card
- Slot number in the chassis that is occupied by this card
- Number of ports supported by the card

3.1.15.5. For each port within give card, or within devices with standalone structure, system should support following attributes:

- Port Identifier (Name/ID)
- Port rate (10G/40G/100G or dual rate 10/40G, 40/100G
- Inventory state (Available/Unavailable)
- Service state (Occupied/Free/Suspended/OutOfService)

3.1.15.6. Network Inventory System data model must support interconnection between the different active network elements. An example of such an interconnection includes, but is not limited to:

- An L2 switch utilizing a port of an L3 device for up-link connectivity
- An L2 and L3 devices utilizing a DWDM port for up-link connectivity

3.1.15.7. Network Inventory System should keep track of all optical transceivers installed in the ports of active devices

- The capacity of the port will depend on the actual type of installed transceiver
- Solution shall support definition of optical transceiver types with at least the following attributes:

- Type (SFP, SFP+, QSFP+, SFP28, QSFP28), Data transmission rate, Range of operation

3.1.16. Active Network/Network management systems

3.1.16.1. Network inventory system must provide support for NMS/EMS systems that will be used for managing active network devices. NMS/EMS capabilities may be used for:

- New device and its capabilities discovery functionality (if supported by the NMS/EMS)
- Support the protocols necessary for implementing device discovery, such as:
    - Device and NMS/EMS-specific northbound interface protocols
    - Support for SNMP MIB-based device and it's capabilities discovery

3.1.16.2. Network inventory system shall consider NMS/EMS systems as inventory items, regardless of whether they are provided as a physical device (e.g. server) or a virtual appliance

3.1.16.3. Network Inventory system should maintain the relationship between an NMS/EMS instance and the devices it manages

3.1.17. Logical resource management

3.1.17.1. The solution shall allow storage and management of IP addresses used as for providing L3 services to customers

3.1.17.2. The system shall provide complete management of IP addresses along with the association of these IP addresses to underlying resources.

3.1.17.3. The system shall support subnets of IP addresses.

3.1.17.4. The system shall allow IP addresses to be associated with network resources

3.1.17.5. The system shall allow for the specification of rules that specify is a given connection requires IP addresses or not.

3.1.17.6. The system shall provide IP address management functionalities such as assign, track and manage IP addresses, including the ability to:

- Associate an IP address with a port
- Enter a base IP address into the inventory
- Divide a base IP address into subnets
- Divide a subnet into more subnets
- View host IP addresses within a subnet
    - Track the status of an IP address
    - Query for existing IP addresses
    - Support IP address inventory states (Assigned/Free)

3.1.17.7. The solution shall allow storage and management of VLAN IDs used as for providing L2 and L3 services to customers

3.1.17.8. The system shall provide VLAN management functionalities such as assign, track and manage VLAN ID's, including the ability to:

- Entering and deleting VLAN IDs to and from the inventory
- Support VLAN ID inventory states (Assigned/Free)
- Solution shall provide description field for VLAN

3.1.18. User interface

3.1.18.1. User interface of the network and service inventory module shall support hierarchical view of the deployed network elements, including containment part, passive part and active part

3.1.18.2. User interface must provide searching/filtering capabilities applicable to all supported types of network elements, based on the following search keys:

- Name/ID (For all devices)
- Region/Area/Street/Building etc.
- Geographical coordinates
- Management IP address (for active devices)
- Managed by a given instance of NMS/EMS system
- Search by any attribute value

3.1.18.3. User interface must support generation of a schematic view of a given network segment

3.1.18.4. Schematic view shall contain all network elements (Active/Passive/Interconnection/Containment facilities. Examples of such schematic views include, but are not limited to:

- A layout of the network elements deployed withing a given containment facility (e.g., shelter)
- Network scheme of a set of connections/other network elements starting from a particular network device, such as:
    - From splice closure up the last ODF
    - Starting from a port of an active device up to customer demarcation point

3.1.18.5. Schematic view must be generated "on-the-fly" based on the most up-to-date information in network inventory database

- Cable/fiber core trace view
- Fiber core cross-connections view
- Schematic view must be exportable to a static file format (e.g. PDF)

3.1.19. Integration with GIS

3.1.19.1. Network and Resource Inventory shall be integrated into a GIS environment

3.1.19.2. Solution shall provide capability it's GIS database to be accessible from open source solutions such as QGIS and GeoServer

3.1.19.3. Proposer shall provide it's own GIS system which is integrated with it's Network and Service resource Inventory and supports the following functions:

- All containment network facilities (Trenches, manholes, ducts, poles, buildings, shelters etc.) must be mapped in the GIS system
- GIS system shall support mapping of passive and active network components located in the containment network facilities

3.1.19.4. Location and addressing information created in the Network Inventory database will have to be synchronized with the addressing database existing in the GIS application

3.1.19.5. The database of GIS system provided with the solution shall be accessible by the Open Net staff

3.1.19.6. Proposer shall make available the documentation describing GIS database structure

3.1.20. Integration with ERP

3.1.20.1. For asset tracking and inventory management purposes, the solution shall be integrated with an ERP system being used at Open Net at the time of solution implementation.

3.1.20.2. It is expected that the ERP system maintains a system of identifiers, called "nomenclature", which are identical for assets sharing the same purpose. Solution shall support the "nomenclature" attribute for network elements

3.1.20.3. It is expected that each network element in ERP has its own unique identifier (manufacturer serial number, in most cases). Solution data model shall support ERP unique identifier as attributes of network elements

3.1.20.4. ERP system is expected to support the notion asset location for tracking purposes. Storages operated by Open Net are examples of such locations. Solution shall support the functionality required to notify ERP system when network element is registered in Network and Service Resource inventory

3.1.20.5. Conversely, the relevant notifications will have to be sent when a network element is decommissioned from Network and Service resource inventory

3.1.20.6. Proposer shall document the existing protocols and procedures used for ERP integration and shall share it's own experience

3.1.21. Integration with network monitoring tools

3.1.21.1. The Network and Service Resource inventory shall be integrated with Zabbix, that is used at Open Net as main system monitoring tool

3.1.21.2. Solution must support the synchronization between the monitoring database and Network Resource database

3.1.21.3. Solution shall support the synchronization of service state of the network resources in its database with that of in the monitoring application

3.1.21.4. In order to support this, the solution must be capable of importing network element service state directly from monitoring database or catch and process the device state change events, such as SNMP notifications

3.1.22. Integration with other modules/functions

3.1.22.1. Solution shall expose the API that allows other modules to use the functionality available in Network and Service Resource Inventory. The following section contains examples of the functions to be implemented as part of this API:

- Network element status inquiry
- Network element capabilities inquiry (e.g. get number of ports for L2 switch)
- Network resource availability inquiry (e.g. get free port information for ODF)

3.1.22.2. As a minimum, the exposed API must be enough to integrate with the other modules of the solution

3.1.22.3. API protocol shall be an HTTP based commonly used protocol, such as REST API, SOAP or any other XML-based protocol

3.1.22.4. Proposer shall provide comprehensive documentation of the API, containing at least function signatures and usage diagrams as part of the description

3.1.23. General requirements

3.1.23.1. Apart from standard attributes described in previous paragraphs, for each type of network element, solution shall allow to define custom attributes

3.1.23.2. Custom attributes will have unique name, be of a defined type (date, string, number etc.) and contain a value

3.1.23.3. User interface shall support searching the inventory database by the values of customer attributes or the combination of attributes

3.1.23.4. Solution shall support free-format text field for every network element that may be used for adding additional information

3.1.23.5. All lists (attributes, fields, types, etc.) provided by this functional requirement description are non-exhaustive. Solution vendor is compelled to enhance the lists based on its experience

3.1.23.6. Solution shall support exporting the data about network components into a text-based format. Corresponding description shall be provided by the vendor

3.1.23.7. Solution shall support importing of data from a text base format. Corresponding description shall be provided by the vendor

3.1.23.8. Solution shall provide network element definitions that are already available in the Solution, but are not described in this functional specification

3.1.23.9. Solution shall provide capabilities for modeling other network elements. This shall be usable by Open Net staff with the appropriate training

3.2. Service Inventory

3.2.1. CFS Specification

3.2.1.1. Solution shall support Customer Facing Services (as defined by TM Forum) with the functionality outlined below:

3.2.1.2. Define customer facing services to support multiple products. A CFS can support multiple products if it is not defined for a specific technology.

3.2.1.3. Support CFS to include attributes of the service that are important to a customer, and hide technology details that are not relevant to a customer.

3.2.1.4. Define customer facing services to represent a single domain. For example, the Dark Fiber Rental CFS represents the Physical Level Services Domain, whereas the DWDM-based services belong to DWDM domain

3.2.1.5. Solution shall support capabilities to define different service domains

3.2.1.6. Solution shall support definition of customer facing services with no explicit dependencies to other customer facing services

3.2.1.7. Solution shall support a notion of Primary Relationship, that is used to associate a Product to CFS services and this type of relationship is used to instantiate a new Service Order. Example of such a relationship is when a customer orders the Data Transmission Product, the Data Transmission CFS fulfills thar order and connects the ordered product with the technical aspects required to fulfill the product services (Products are defined in BSS requirements document; Service Orders are defined in Service Order Management module requirements)

3.2.1.8. Solution shall support Auxiliary relationships between Products and CFS that do not create a new Service Order for Product, but merely enriches the existing Service Order with additional capabilities

3.2.1.9. Solution should support Exclusive relationships (e.g., a FO core cannot be used for two instances of Dark Fiber CFS)

3.2.1.10. Solution should support Shared relationship (e.g., a fiber cable connected to uplink port of a L2 switch can be shared between two Data Transmission CFS instances that utilize different ports

3.2.1.11. Solution shall provide the capabilities to manage CFS definitions/specifications by supporting the following functionality

- Creation new CFS specification manually using system GUI
- Support establishment of relationships (described above) between CFS and product(s) using System GUI

- Supports configuring relevant RFS(es) for the CFS specification using system GUI"

3.2.1.12. Solution shall support Creation, Modification/Edit and Deletion of CFS specifications

3.2.2. RFS Specification

3.2.2.1. Solution shall support Resource Facing Services (RFS, as defined by TM Forum). It basically describes how customer facing services are configured on a given Network Element

3.2.2.2. System shall provide the functionality that allows system users to determine and configure which resource facing service(s) are used to provide CFS services during service design stage

3.2.2.3. Solution shall support Resource Facing Services that are technology-specific but not necessarily tied to vendor-specific implementation

3.2.2.4. Solution shall support Creation, Modification/Edit and Deletion of RFS specifications

3.2.3. CFS/RFS instances

3.2.3.1. Solution shall provide the capability to store all CFS and RFS instances

3.2.3.2. Each CFS instance shall be associated with the instances of RFS that were allocated to fulfill the service delivery

3.2.3.3. Each RFS instance shall be associated with Network and Service resources that are used to fulfill the RFS

3.2.3.4. The system shall have ability to present complete Customer view of inventory, which means displaying all CRF and RFS instances belonging to a given Customer and/or Account (Customers and Accounts are defined in BSS Requirements document)

3.2.3.5. Solution shall support querying/searching capabilities for CFS instance database. Following are examples possible search criteria: Customer ID, Service Instance ID, Service Specification

3.2.3.6. Solution shall support exporting full or partial service inventory information into a text-based format

3.2.3.7. Proposer is compelled to provide Service Inventory functionality that will aid Open Net in the process of preventing unauthorized access and use of the Open Net network

3.3. Service Order Management (SOM) and Service Provisioning

3.3.1. SOM/Service Order Execution

3.3.1.1. The Service Order request execution require several operations that need to be executed in a consistent way and these operations shall be orchestrated by the solution

3.3.1.2. Processing step defined in the workflow represents an action to be performed on CFS and RFS instances, possibly involving sending provisioning commands to network devices involved **(Optional)**

3.3.1.3. Solution shall allow monitoring of workflow executions

3.3.1.4. Solution shall support suspension and/or resumption of workflow execution process

3.3.2. SOM/Service Order Management configuration

3.3.2.1. The Service Order execution workflow shall be described using a declarative notation, preferably BPMN

3.3.2.2. Workflow shall be customizable, and the customization should not require coding

3.3.2.3. With the appropriate training, the customization shall be manageable by Open Net staff, who should be able to assemble and configure all steps that will be executed in the workflow from the reception of a Service Order request until the provision of the result of the execution.

3.3.2.4. Workflow customization capabilities shall be flexible enough to skip one or more fixed step if they are not needed for specific Service Order types

3.3.3. SOM/Order decomposition

3.3.3.1. Solution shall support splitting a Service Order into several Resource Order requests (a process sometimes called Order Decomposition)

3.3.3.2. The process of breaking down Service Order into Resource Orders shall not be hard-coded and be configurable from the relevant management screens

3.3.3.3. Depending on the type of the Resource Order, or type of the resource itself, Service Order Management function shall support sending Resource Order to Service Provisioning function, or create a task to be handled manually by a member of Open Net staff

3.3.3.4. The rules for selecting the way particular Resource Order are handled, shall be configurable and shall not require extra coding.

3.3.4. Service Provisioning/Resource Order Execution

3.3.4.1. The Provisioning module shall execute Resource Orders generated from Service Order.

3.3.4.2. Upon encountering an error during the execution of a Service Order, processing shall stop and corresponding alarm message must be generated

3.3.4.3. Solution shall allow restarting the execution of the Service Order

3.3.4.4. Solution shall provide capability to undo (rollback) the changes made by successfully executed Resource Orders should one or more subsequent Resource Order execution fail

3.3.4.5. In the case of an error is detected when exciting Resource Order, appropriate alarm shall be raised in the system

3.3.4.6. Solution shall support re-routing of a failed Resource Order to be manually executed by the member of Open Net staff

3.3.4.7. In some cases, instead of direct access to network equipment, Proposer will have to support the EMS/NMS systems that manages the actual network elements

3.3.4.8. Proposer may opt to use information available in Network and Service Resource Inventory Database to identify the network elements and/or EMS/NMS systems to be accessed when executing Resource Orders

3.3.4.9. Service Provisioning solution shall be flexible enough to onboard a new type of network and EMS/NMS equipment

3.3.4.10. Proposer shall provide a list of network equipment and EMS/NMS systems it had already integrated

3.3.4.11. Transport protocols supported by Service Provisioning shall include commonly used ones, such as:

- Sending equipment-specific command over plain TCP socket connection
- MML command lines over Telnet, SSH
- API invocations using REST and/or SOAP-based web services

3.3.4.12. Solution shall be able to display the state of provisioning commons during the execution of a Service Order

3.3.4.13. Executed provisioning commands shall be logged for analysis and troubleshooting purposes

3.3.4.14. Service Provisioning solution shall support "previewing" Resource Order commands to be verified by network engineers at provisioning logic design stage before launching is into production environment

4. General technical requirements

4.1. User interface/Localization

4.1.1. The solution should support Unicode encoding both in database and user interface. This is required for Georgian language support

4.1.2. System default user interface shall be English, but defining additional languages shall be also possible

4.1.3. Solution vendor shall provide language-specific resource files in case if Open Net decides to translate whole or part of the UI into Georgian language

4.2. GUI

4.2.1. System GUI shall be browser based

4.2.2. Browser based GUI shall support modern concept of single page application and fully utilize HTML 5 and Javascript capabilities of modern browsers

4.2.3. Browser based GUI shall be optimized to be accessible from a tablet device

4.2.4. It is allowed to have some functions implemented as native GUI desktop-based application (e.g. mapping functions, advanced features of inventory management etc.)

4.2.5. If a system function GUI is implemented as desktop-based local application, the Proposer shall provide a complete list of hardware and 3rd party software environment specification for client workstation running the GUI-based desktop application

4.2.6. Functional and business requirements of the system state the list generation and filtering capabilities. In all such cases, the GUI function shall support exporting the list in the following formats: CSV, Excel file, Printable PDF file

4.2.7. The system must have a maximum response time of 5 sec for any CRM actions in any application screen.

4.3. Traceability/Audit

4.3.1. The solution should be able to keep audit trail of any actions performed by uses that resulted in a change of any object managed by the system, which includes:

- User performing the action
- Timestamp
- Comment field for explaining the action (optional)

4.3.2. System should support integrity and protection for audit/log files.

4.3.3. System should provide transaction logs to record executed functions to facilitate diagnosis and reconciliation of system errors

4.3.4. System should be able to provide full audit trail of every access to sensitive information, such as viewing customer information, customer contact information, together with each and every time a screen is accessed, by whom, when, and from which IP address

4.3.5. Vendor shall make the logs available for the access and analysis by the external SIEM (security incident and event monitoring) systems, such as Graylog

4.3.6. Solution should retain historical records of values of the objects in the database, along with the start and end timestamps

4.4. Security/Authentication and Authorization

4.4.1. Any access to the solution is provided only after personal identification and authentication.

4.4.2. System should provide the ability to track all successful and failed log-ins

4.4.3. It shall be possible to define different user profiles with associated access rights to the Solution and its tools.

4.4.4. It shall be possible to define at least different types of user profiles like:

- Administration,
- Operation,
- Workflow designer,
- Configuration manager,
- Error manager...

4.4.5. Proposer will describe in its technical offer all pre-existing users profile in the Solution and their associated rights (parameters read / creation / modification, errors retrieval / correction / resubmission, workflow view / creation / modification...)

4.4.6. Each user defined in the system can be associated with one or several user profiles.

4.4.7. It should be possible to overwrite the standard rights associated to a user profile for a given user (adding additional rights or removing some of the standard user's profile rights).

4.4.8. Application access controls must be used to limit visibility and access to highly sensitive information.

4.4.9. Users cannot access information and resources for which they are not authorized (description of profiles and rights)

4.5. Security/Password protection

4.5.1. Access to the Solution must require secure passwords.

4.5.2. Password must be encrypted (no password must be stored in readable format)

4.5.3. Password must expire after a pre-defined period

4.5.4. User must be locked after n unsuccessful attempts, n is configuration parameter (for

4.5.5. example 3 unsuccessful attempts)

4.5.6. System must verify the strength of passwords and refuse weak passwords.

4.5.7. On expiry of user password, change of password must be possible through the GUI by the user.

4.5.8. Session timeout – User disconnection after a configurable period of inactivity (for example 30 mins).

4.5.9. Inactivity interval value shall be configurable per user role

4.6. Security/Access control

4.6.1. Access to the Solution must be allowed only if it is made from known IP address (or IP address range).

4.6.2. System must support SSL encryption for all HTTP-based interfaces, including Web-based UI and programmatic API access to prevent illegal capture of sensitive information

4.6.3. Logging of users accessing the system and monitoring of online activity needs to be provided as part of the Solution.

4.6.4. Sufficient information should be logged to allow full traceability of user actions, for example (but not limited to):

- IP address
- Username
- Date and Time
- Type of transaction
- Data associated with transaction

4.7. Security/Credential management

4.7.1. It should be possible to integrate the new solution with an Active Directory system

4.7.2. No generic login should be used for one system to communicate with another system. Specific logins should be used.

4.7.3. The system should allow the regular change of components' passwords. Any change of password should not stop the proper functioning of the system due to the old password being hard-coded in a configuration file or in a database.

4.7.4. Password information must be encrypted or hashed

4.8. Hardware

4.8.1. System shall not require bespoke hardware - it shall run on commodity servers - machines with standard x86 architecture

4.8.2. Proposer is expected to provide hardware requirements for the system, which includes:

- Server specifications
- Storage specifications - Proposer shall indicate if solution requires external storage
- Network specifications

4.8.3. Above specifications shall be based on the assumption that the system must not have single point of failure

4.8.4. Server hardware shall consist of 2 servers of commodity x86 architecture. Each of these servers will be running production workloads either in Primary or Disaster Recovery locations.

4.8.5. Servers with production workloads will be distributed between primary and disaster recovery sites. At each site, one server will be shared for BSS and OSS parts.

4.8.6. For resiliency and scalability, the proposed hardware shall contain two pieces of external shared data storage equipment, one located at Primary and another – at Disaster Recovery site. Servers are directly attached to the storage located at the same site.

4.8.7. It is assumed that the solution contains a relational database management system (RDBMS) that will host most of the data and will generate a large part of I/O activity. It is also assumed that solution vendor shall set-up data replication between the Primary and Disaster recovery instances of RDBMS. Also, the RDBMS system shall be an open-source system such as MySQL and Postgres SQL.

4.8.8. The storage capacity shall consider the additional 4TB storage for Open Net needs.

4.9. Deployment

4.9.1. It is expected that the system will be deployed in a virtualized environment controlled by a unified virtual environment management software (such as VMWare, KVM etc.)

4.9.2. It shall be possible to deploy Modules of the system as virtual machines (dependent on the overall architecture of the system)

4.9.3. If the system is designed using microservice architecture, the alternative deployment option is a “contrainerized” deployment (such as Docker) managed by an orchestration solution (such as Kubernetes)

4.9.4. Docker/Kubernetes software may run both in "bare metal" and virtualized environment

4.9.5. The critical components of the solution shall be deployed in an on-premise infrastructure (provided by the Proposer). Non-critical components may be deployed in a well-known and established public cloud environment without the need of significant code change

4.10. Operation and Administration

4.10.1. Proposer should provide all necessary operational and monitoring tools.

4.10.2. Operational and monitoring tools should have user-friendly, interactive GUI.

4.10.3. Operational and monitoring tools should offer threshold based alerting mechanism

4.10.4. Operational and monitoring tools should have enhanced errors and warnings’ logging functionality with configurable logging levels.

4.10.5. Operational and monitoring tools should allow real-time dynamic view and monitoring of the running processes, also full or semi-automated (pre-configured per case) and manual intervention or reaction in case of necessity.

4.10.6. Operations management should support both automatic and manual error analysis and reaction

4.10.7. Monitoring tools shall monitor all components of the solution (hardware, operating systems, 3rd party software, core system software)

4.10.8. Open Net shall be able to attach an external monitoring application to the solution

4.11. 3rd party software

4.11.1. Proposer shall produce an exhaustive list of 3rd party software needed to operate the system:

- Operating systems
- Virtualization environment
- Containerization environment
- RDBMS and other (NoSQL) database systems

- Any type of middleware required (e.g. message queue brokers, enterprise service bus etc.)
- Reporting, dashboarding and data visualization systems
- Any other 3rd party software not listed above (ETL, CI/CD pipeline etc.)

4.11.2. Open Net strongly prefers open source and free software. However, commercial alternatives are also acceptable, provided that:

- Proposer transparently lists all costs associated with running, maintenance and upgrade of the software
- The above shall include both licensing and support costs

4.11.3. 3rd party software maintenance shall be performed by Proposer after the system is launched

4.11.4. It is Proposers responsibility to maintain compatibility between the solution and 3rd party components

4.12. Test and training environment

4.12.1. In order isolate live data and processes from being accidentally adversely affected, Proposer shall provide test and training environment

4.12.2. Proposer shall strive to minimize the costs of test and training environment by recommending a bare minimum of hardware and 3rd party software for test and training environment

4.12.3. All updates to the system, including the updates of the executable code of the solution, system configuration, updates to third-party software e.g. OS, virtualization/containerization software, database, shall be tested on test and training environment before being deployed in the production environment

4.12.4. Solution should support adequate infrastructure that allows easy transfer from the application development/test environment to the production environment or vice-versa

4.12.5. All non-production environment should be fully integrated with all the third parties like the production environment.

4.13. Database Structure

4.13.1. The solution shall provide full and unrestricted access to all relational database structures used for all components of the system along with the guidance on how to access it for data extraction purposes without negatively affecting performance of the system

4.13.2. Solution shall contain comprehensive database structure documentation

4.13.3. If a non-relational database is used, the Proposer shall provide the necessary tools, documentation, and software capability to extract the data from No SQL databases for further processing by Open Net

4.14. Configuration management

4.14.1. The System should support versioning of configurations, manual or automated scheduled application, rollback and activation of specific version via GUI.

4.14.2. All the history/versions of system's configuration must be stored.

4.14.3. All the history/versions of system's configuration should be accessible via by a source-code/configuration management system (GIT is an example of such a system)

4.14.4. Solution shall support allow export/import functionality for all the configurations present in system.

4.14.5. Required configuration can be exported and imported to production on-the-fly without any downtime.

4.15. Capacity

4.15.1. System must respond within 2 seconds to any API request

4.15.2. System must support sufficient number of concurrent connections to and from any Network element directly or indirectly through Provisioning in order to provide required performance.

4.15.3. System must have no software/license limitation for the number of concurrent batch executions/processes.

4.15.4. System must not have any software or License limitation on concurrent User login and handling actions or operations.

4.15.5. System must not have any software or License limitation on the number of Customer Contacts created in CRM

4.15.6. System must have no SW limitation or License limitation for inbound and outbound integration of all External Systems.

4.15.7. System must support sufficient number and volume of transactions on any system interface

4.15.8. Provisioning system capacity must match or exceed the provisioning capacity of all related Network elements.

4.15.9. System must provision all services in minimum possible time (depending on service type).

4.15.10. System must have no SW limitation for number of creating Product catalogue entities.

4.15.11. System must have no SW limitation creation of global product catalogue entities and assign to all the Subscribers.

4.15.12. System must have no SW limitation to define Product Catalogue items per subscriber, customer or billing account without performance degradation.

4.15.13. System must support unlimited Inventory in resource management. All the Resource (With all the status normal, used, Sold or deactivated) should be Queried at any time. Query condition can be single or multiple at a time

4.15.14. System must have no SW limitation on the number and size of upload documents from users per customer.

4.16. Resilience and fault tolerance

4.16.1. Solution shall be provided in a highly available configuration that eliminates any possible single points of failure

4.16.2. The high availability requirement is applicable to any proposed deployment model - on premise (fully or partially virtualized or containerized) and or cloud based

4.16.3. Proposed solution shall achieve 99% availability per year

4.16.4. Any maintenance work (software, hardware or networking component upgrades) shall be able to be finished in a night-time maintenance window (e.g. 02:00 - 07:00)

4.16.5. Proposed solution deployment model shall allow geographic location redundancy

4.16.6. Solution shall be integrated into the backup environment of Open Net, which uses Veeam a primary backup environment