PRGI Detailed Solution Architecture Description (ESO-LT part)

Project Name	PRGI - Pakartotinai registruojamų gedimų identifikavimas IS
Project ID	**RFC-446** - Getting issue details... STATUS
Solution Architect	Alenas Jankaitis
Enterprise Architect	• Andrius Tiškevičius
Infrastructure Architect	N/A
IT security	Tautvydas Švėgžda
Reviewed by	Aleksandras Sorokinas Aleksandras Jerdiakovas Sergej Opara

Introduction

Purpose

Solution for possibility to identify a repeated request to register outage received from customer to ESO-Lt, and after identifying it, do not send it to UVIS and later to DMS, and inform Customer about outage status.

Target audience

#	Information System	Description
1	ESO.lt	Energy Distribution Operator (ESO for short), which is managed by the state capital Ignitis grupė, distributes electricity and gas and maintains distribution networks so that they are reliable and efficient, takes care of troubleshooting network failures and connecting new customers.

2	EIMS	Enterprise Integration Management System
3	IMPS	Information messages processing system. IMPS will protect DMS IS from an excessive and uncontrolled flow of messages about possible power outages.
4	UVIS	Request management information system
5	ElasticSearch	Elasticsearch is a distributed search and analytics engine built on Apache Lucene.
6	BS	BS IS DB data will be used to create ElasticSearch address and obj indexes

Glossary

#	Term	Description
1	ESO.lt	Energy Distribution Operator (ESO for short), which is managed by the state capital Ignitis grupė, distributes electricity and gas and maintains distribution networks so that they are reliable and efficient, takes care of troubleshooting network failures and connecting new customers.
2	EIMS	Enterprise Integration Management System
3	IMPS	Information messages processing system. IMPS will protect DMS IS from an excessive and uncontrolled flow of messages about possible power outages.
4	UVIS	Request management information system
5	Elastic Search	Elasticsearch is a distributed search and analytics engine built on Apache Lucene. It is popular search engine and is commonly used for log analytics, full-text search, security intelligence, business analytics, and operational intelligence use cases
6	BS	BS IS DB data will be used to create ElasticSearch address and obj indexes

References

#	Name, Version	Description	Location
1	Conceptual model	Prepared conceptual model	PRGI Conceptual Solution Architecture Description

Assumptions

<All the assumptions considered by the project, e.g. deliverables of the other projects>

We assume that this project is

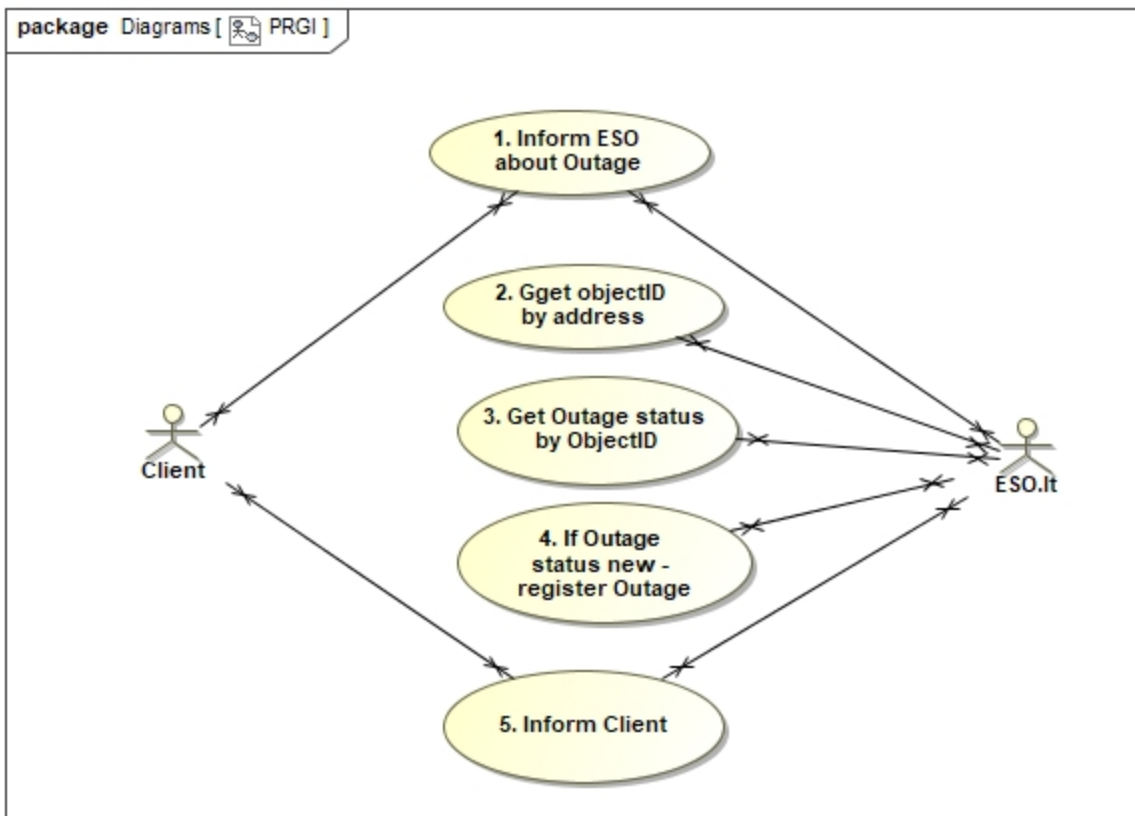| #	Assumptions
1	IMPS Outage status checker will be ready, accurate and fast enough.
2	ESO.LT will implement customer request processing changes
3	A new ElasticSearch obj index will be created

Policies and Standards

<All the standards fallowed by the project>

#	Standards, Policies
1	API aprašymo rekomendacijos - IS dokumentacija - Confluence (etic.lt)
2	Techninių reikalavimų gairės EIMS integracijoms (v3) - IS dokumentacija - Confluence (etic.lt)
3	Swager (OpenAPI) specifikacija - IS dokumentacija - Confluence (etic.lt)
4	AsyncAPI specifikacija - IS dokumentacija - Confluence (etic.lt)

Requirements View

Use Cases

package Diagrams [PRGI]

1. Inform ESO about Outage

2. Gget objectID by address

3. Get Outage status by ObjectID

4. If Outage status new - register Outage

5. Inform Client

Client

ESO.lt

All the main use cases that solution need to fulfill

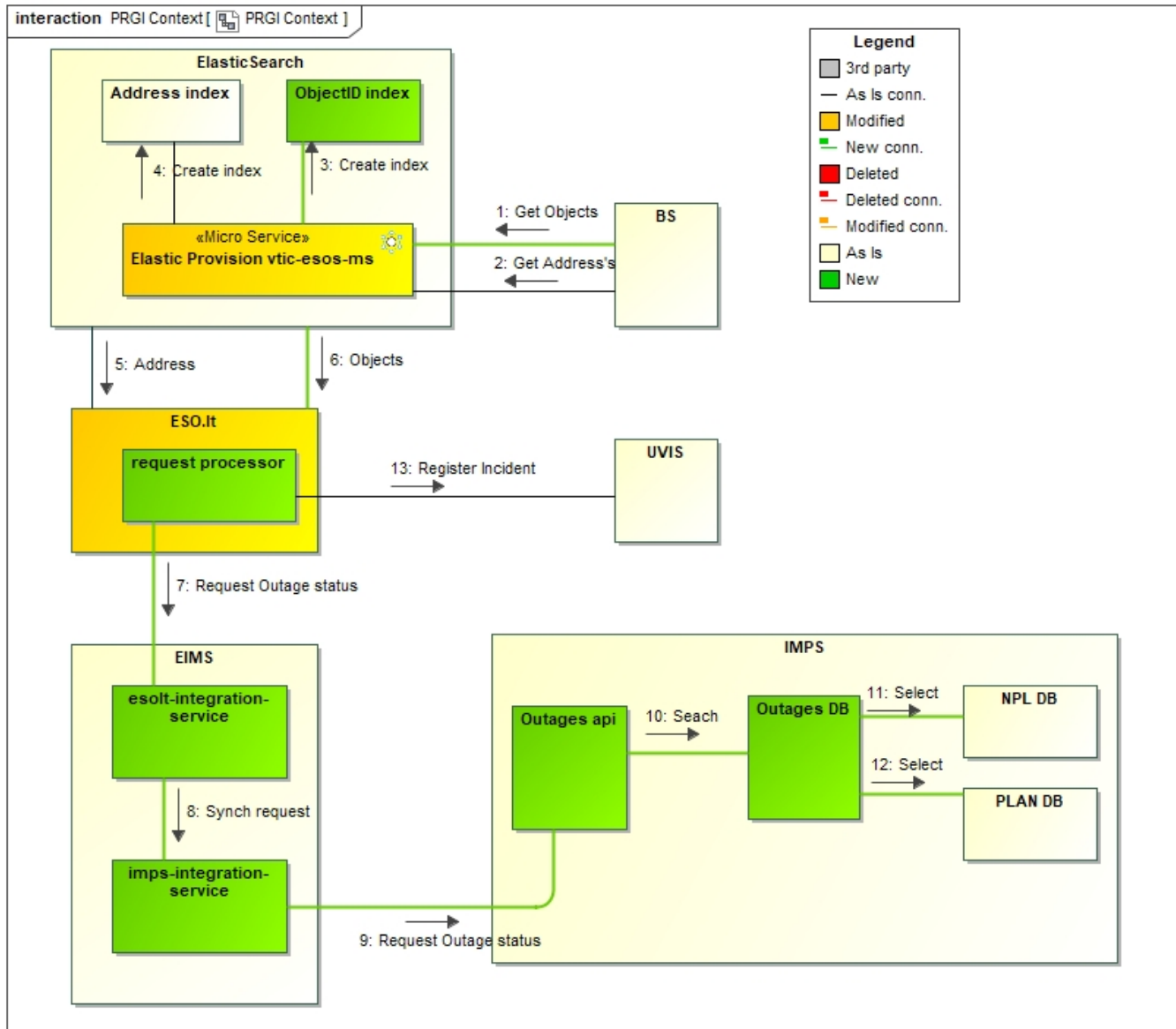| Use Case	Description
1.	Customer Inform ESO about outage
2.	Get ObjectID by address
3.	Get Outage status by ObjectID
4.	If outage status new - register incident to UVIS
5.	Inform customer

Architecture

Architecture decisions

This section briefly describes what decisions you made and why you made those decisions

#	Decision	Description (Pros, Cons, Risk)
1	Create new ElasticSearch **obj** index	BS IS DB data will be used to create ElasticSearch **address** and **obj** indexes. A copy of data from the BS_ADRESAI and BS_OBJ tables will be made daily to the ElasticSearch local DB. The local ElasticSearchDB will be used to create and update indexes. this whole process takes a lot of time and resources.

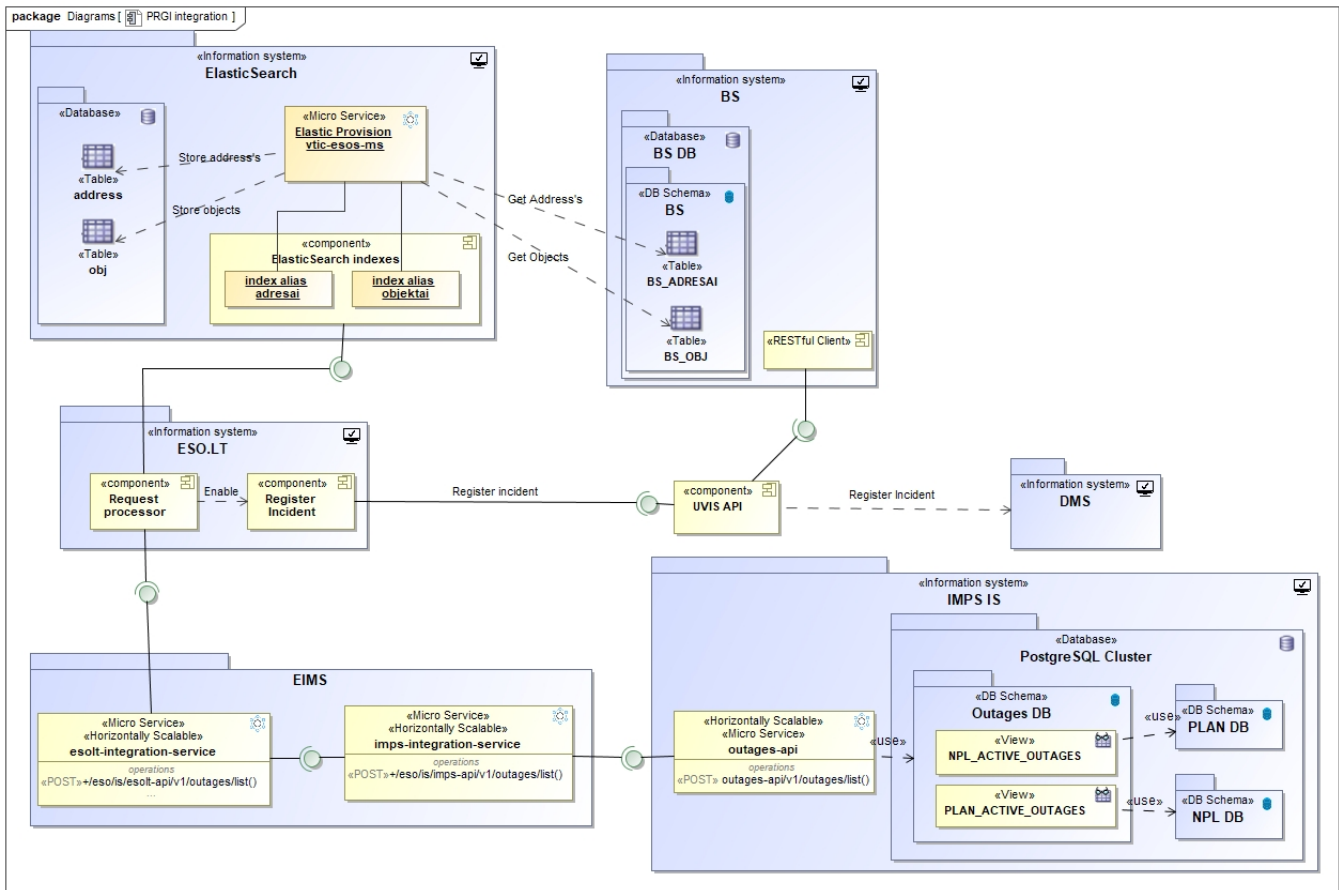| 2	ESO.lt have to implement new customer request handling logic	Similar logic should be implemented in other channels that process user requests if they want to use PRGI functionality.
		If later there are changes in PRGI, then all systems will have to be changed
3	EIMS will be used to pass synch. request from ESO.lt to IMPS	Synchronous requests will be made. in case of a large number of requests, there may be delays. A load test should be performed.
4	IMPS NPL and PLAN DB will be used to get requested outages statuses	The PLAN and NPL databases are designed to monitor and manage the process of messaging to customers affected by an electricity incident.
		Getting accurate information about an object that has lost electricity is complicated and not always accurate.

Context View

Rules, links, recommendations to be followed.

Summary of the changes should be provided in the table

#	System	Type of Change	Description
1	ElasticSearch	NEW	Get active electric object data from BS DB, BS_OBJ table

2	ElasticSearch	AS-IS	Get active address data from BS DB, BS_ADRESAI table
3	ElasticSearch	NEW	Create and maintain objects index
4	ElasticSearch	AS-IS	Create and maintain address's index
5	ESO.lt	AS-IS	Gets address's from Elastic Search
6	ESO.lt	NEW	Gets objects ID's from Elastic Search by address ID
7	ESO.lt	NEW	POST array of objects ID's to EIMS API and waits for synchronous response
8	EIMS	NEW	EIMS esolt-integration-service POST array of objects ID's to EIMS imps-integration-service API and waits for synchronous response
9	EIMS	NEW	EIMS imps-integration-service POST array of objects ID's to IMPS outages-api and waits for synchronous response
10	IMPS	NEW	Search for all objects in Outages DB to check is object outage is active and generate response and send it back
11	IMPS	NEW	Select to view, all known objects in NPL DB effected by electricity outage
12	IMPS	NEW	Select to view, all known objects in PLAN DB effected by electricity outage
13	ESO.lt	AS-IS	If object status is new - then allow to register incident to UVIS

Integration View

Rules, links, recommendations to be followed.

Additionally, rules and recommendations to be followed when designing component diagrams for EIMS.

All the changes of the integrations should be listed in the table below

#	High level flow description	Source system	Target system	Initiator (Active side)	Integration component						Protocol/Type								M
					EIMS	ESB	Oracle AQ	SFTP	App. Server	Other	HTTPS	Messaging	SFTP	SMTP	DB conn.	LDAP	Other	SOA	
1	ESO.lt POST request to EIMS (esolt-integration-service) to get Outages statuses	ESO.lt	EIMS (esolt-integration-service)	ESO.lt	☑	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☑
2	EIMS (esolt-integration-service) POST request to EIMS (imps-integration-service)	EIMS (esolt-integration-service)	EIMS (imps-integration-service)	EIMS (esolt-integration-service)	☑	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☑
3	EIMS (imps-integration-service) POST request to IMPS outages-api to get Outages statuses	EIMS (imps-integration-service)	IMPS outages-api	EIMS (imps-integration-service)	☑	☐	☐	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐	☐	☑
4	ElasticSearch Provision vtic-esos-ms queries BS DB	BS DB	ElasticSearch Provision vtic-esos-ms	ElasticSearch Provision vtic-esos-ms	☐	☐	☐	☐	☐	☑	☑	☐	☐	☐	☑	☐	☐	☐	☐
5	ESOlt queries ElasticSearch	ElasticSearch Provision vtic-esos-ms	ESOlt	ESOlt	☐	☐	☐	☐	☐	☑	☑	☐	☐	☐	☐	☐	☐	☐	☑

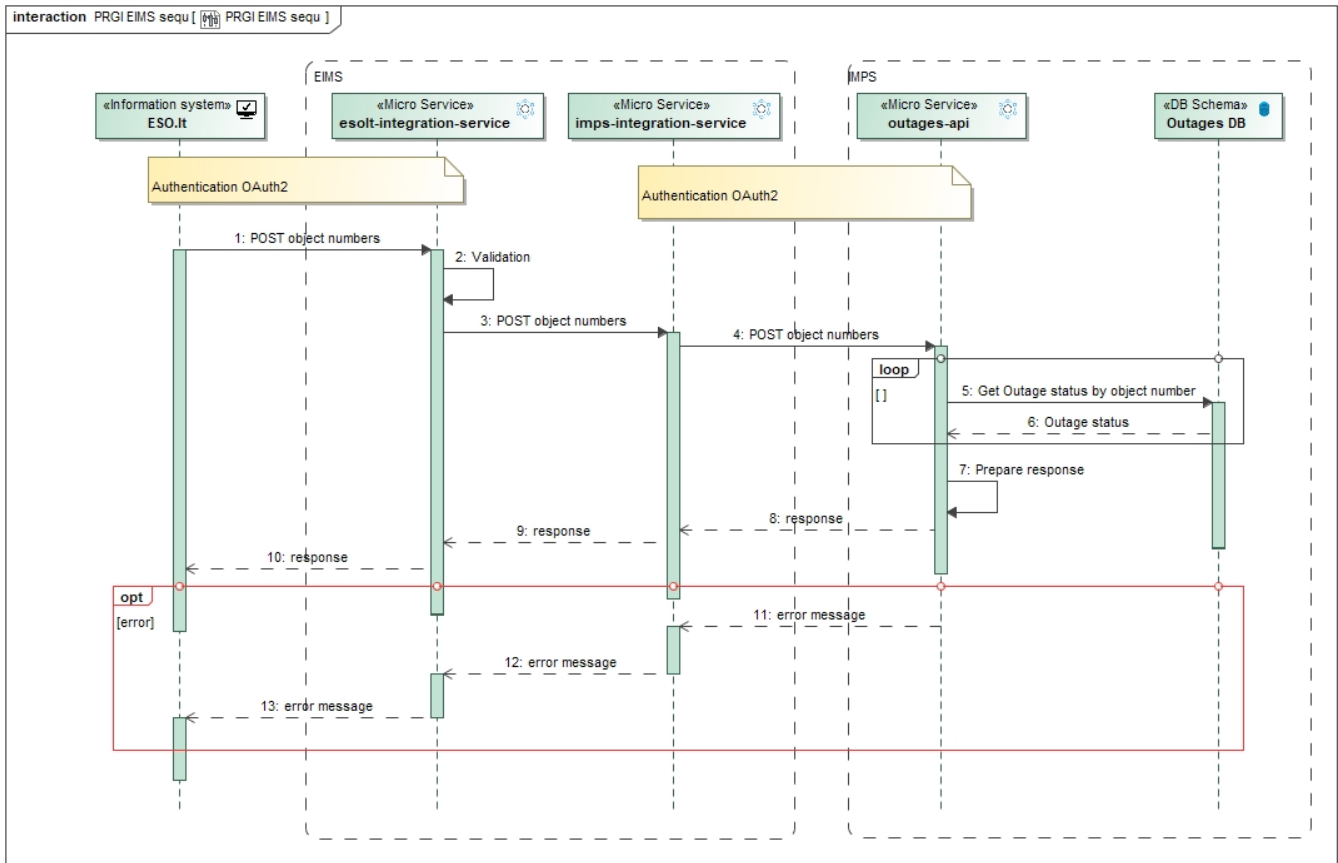## Security Definition (Authentication and Authorization)

This section describes security types, roles, and rules between systems depicted on the component diagram. This table must be filled out according to the component diagram (which is provided in Integration View section) and communication need between information systems.

In case Open API specification exists with a clear definition of Authentication and Authorization, a reference link to the Open API specification must be provided.

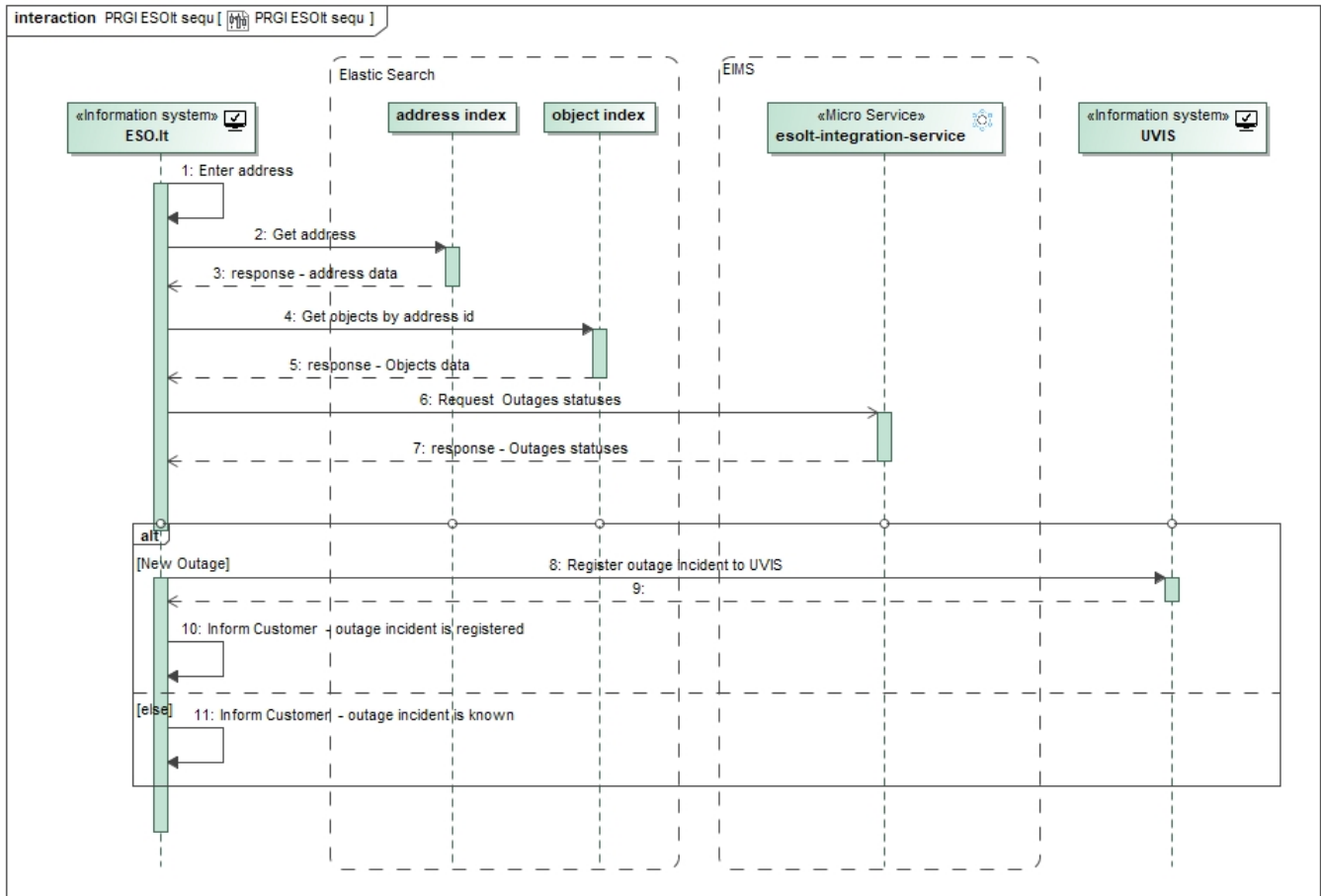| #	Information System (IS)	EndPoint's Type	EndPoint's List	Type of Authentication	Roles (Authorization)	Description
1	ElasticSearch Provision vtic-esos-ms	GET	https://tstelastic.eso.lt/adresai/_search	HTTPS, TLS1.2		ESOlt queries ElasticSearch
2	ESO.lt	POST	/eso/is/esolt-api/v1/outages	OAuth2.0		ESO.lt POST request to EIMS (esolt-integration-service) to get Outages statuses
3	EIMS (imps-integration-service)	POST	outages-api/v1/outages	OAuth2.0		EIMS (imps-integration-service) POST request to IMPS outages-api to get Outages statuses

Sequence steps:

ESO.lt - EIMS-IMPS request/response sequence diagram

#	Source	Destination	Description	Comment
1	ESO.lt	EIMS (esolt-integration-service)	POST array of ObjectID, to check Outages statuses	https://gitlab.ignitis.lt/apihub/eims-api/system-apis-layer/esolt-integration-service/-/blob/main/esolt-integration-service-openapi.yaml
2	EIMS (esolt-integration-service)	EIMS (esolt-integration-service)	validation	he following validations should take place: • Required Fields. • Format of **objectNumber** • array size In case of a validation errorthen flow should proceed to step #13 and EIMS IS (esolt-integration-service) should return a 400 HTTP error code back to ESO.lt
3	EIMS (esolt-integration-service)	EIMS (imps-integration-service)	POST array of ObjectID, to check Outages statuses	If an error occurs during this HTTP call (HTTP Status <> 200), then flow should proceed to step #12 and return the error.
4	EIMS (imps-integration-service)	IMPS (outages-api)	POST array of ObjectID, to check Outages statuses	
5	IMPS (outages-api)	Outages DB	Select data to get outages statuses from NPL_Outages & PLAN_Outages views in Outages DB	In step 4 we will get array of object numbers. OutagesDB has two views that contain object numbers with known active Outages. we need to create a SQL query that searches for records by object number. We have to query for each object number in array.

6	NPL/PLAN DB	IMPS (outages-api)	response message with contracts information data	If query returns a record - that means that outage for this object number is known and outage object property **isActive** must be set to **true**
				
				if the query doesn't return any records - then **isActive** must be set to **false**
				
				outage:
				
				shema:
				
				{
				  objectNumber*: string
				  isActive*: boolean
				  isPlanned*: boolean
				  incDmsRef: stringnull
				  startDateTime: date-timenull
				  endDateTime: date-timenull
				  estimatedRestorationDate: date-timenull
				}
				
				data sample:
				  {
				    "objectNumber": "AAAAAAAA",
				    "isActive": false,
				    "isPlanned": false,
				    "incDmsRef": "AAAAAAAA",
				    "startDateTime": "2024-05-29 08:34:18.000",
				    "endDateTime": "2024-05-29 08:34:18.000",
				    "estimatedRestorationDate": "2024-05-29 08:34:18.000"
				  }

7	IMPS (outages-api)	IMPS (outages-api)	Generate response	Server successfully executed the request and response was received
				
				We have to create object and send it as a response:
				
				{
				  "reqDateTime": "1970-01-01T00:00:00.000Z",
				  "outages": [
				    {
				      "objectNumber": "AAAAAAAA",
				      "isActive": true,
				      "isPlanned": false,
				      "incDmsRef": "AAAAAAAA",
				      "startDateTime": "2024-05-29 08:34:18.000",
				      "endDateTime": "2024-05-29 08:34:18.000",
				      "estimatedRestorationDate": "2024-05-29 08:34:18.000"
				    },
				
				    {
				      "objectNumber": "AAAAAAAA",
				      "isActive": false,
				      "isPlanned": false,
				      "incDmsRef": "AAAAAAAA",
				      "startDateTime": "2024-05-29 08:34:18.000",
				      "endDateTime": "2024-05-29 08:34:18.000",
				      "estimatedRestorationDate": "2024-05-29 08:34:18.000"
				    }
				  ]
				}
				
				reqDateTime - request receipt date - taken from the request header

8,9, 10	IMPS (outages-api), EIMS	ESO.lt	response message with Outages statuses information data	IMPS synchronously returns response to EIMS (imps-integration-service) in JSON format.
11, 12,13	IMPS (outages-api), EIMS	ESO.lt	providing response error message to is system	If an error occurs during this HTTP call ([HTTP Status <> 200]), then EIMS should return the error.

ESO.lt sequence diagram

#	Source	Destination	Description	Comment
1	ESO.lt	ESO.lt	Start registering incident, enter address	Designed and implemented by ESO.lt contractor
2	ESO.lt	ElasticSearch address index	GET address from ElasticSearch address index	Designed and implemented by ESO.lt contractor
3	ElasticSearch address index	ESO.lt	response with address data	Designed and implemented by ESO.lt contractor
4	ESO.lt	ElasticSearch object index	GET objects from ElasticSearch object index	will be designed and implemented by ESO.lt contractor
5	ElasticSearch object index	ESO.lt	response with object data	will be designed and implemented by ESO.lt contractor
6	ESO.lt	EIMS (esolt-integration-service)	POST array of ObjectID, to check Outages statuses	https://gitlab.ignitis.lt/apihub/eims-api/system-apis-layer/esolt-integration-service/-/blob/main/esolt-integration-service-openapi.yaml
7	EIMS (esolt-integration-service)	ESO.lt	response with Outages status data	https://gitlab.ignitis.lt/apihub/eims-api/system-apis-layer/esolt-integration-service/-/blob/main/esolt-integration-service-openapi.yaml
8	ESO.lt	UVIS	register incident to UVIS	Designed and implemented by ESO.lt contractor
9	UVIS	ESO.lt	UVIS response	Designed and implemented by ESO.lt contractor
10	ESO.lt	ESO.lt	Inform Customer - outage incident is registered	will be designed and implemented by ESO.lt contractor
11	ESO.lt	ESO.lt	Inform Customer - outage incident is known	will be designed and implemented by ESO.lt contractor

Implementation View

APIs Description

This section provides detailed APIs description.

	GIT
EIMS - esolt-integration-service	https://gitlab.ignitis.lt/apihub/eims-api/system-apis-layer/esolt-integration-service/-/blob/main/esolt-integration-service-openapi.yaml

Technology View

<Short description of the project>

Deployment View

Environments View

Information System Environments

This section describes information about existing information systems that are used for designed technical solutions.

Information System (IS)	Environment	URL	Comments
EIMS	TEST	vtst-eims-ws1	
EIMS	DEV	vdev-eims-ws1	
EIMS	PROD	vtic-eims-ws1	
IMPS	TEST	vtst-imps-api1	
IMPS	DEV	vdev-imps-api1	
IMPS	PROD	vtic-imps-api1	
ElasticSearch	TEST	vtst-esos-ms	
ElasticSearch	DEV	vdev-esos-ms	
ElasticSearch	PROD	vtic-esos-ms	

FW rules list