BalticLSC Software Prototype

Overview of the BalticLSC Software Prototype Code

Version 1.00

Priority 1: Innovation

Warsaw University of Technology, Poland
RISE Research Institutes of Sweden AB, Sweden
Institute of Mathematics and Computer Science, University of Latvia, Latvia
EurA AG, Germany
Municipality of Vejle, Denmark
Lithuanian Innovation Center, Lithuania
Machine Technology Center Turku Ltd., Finland
Tartu Science Park Foundation, Estonia

BalticLSC Software Prototype

Overview of the BalticLSC Software Prototype Code

Work package	WP6
Task id	6.2
Document number	O6.2
Document type	Main Output
Title	BalticLSC Software Prototype
Subtitle	Overview of the BalticLSC Software Prototype Code
Author(s)	Krzysztof Marek, Marek Wdowiak
Reviewer(s)	Radosław Roszczyk, Agris Sostaks, Kamil Rybiński
Accepting	Michał Śmiałek
Version	1.00
Status	Final version

Interreg Baltic Sea Region
EUROPEAN UNION
EUROPEAN REGIONAL DEVELOPMENT FUND

History of changes

Date	Ver.	Author(s)	Change description
13.09.2021	0.1	Krzysztof Marek (WUT)	Document creation
06.10.2021	0.2	Krzysztof Marek (WUT)	Chapters 1 and 2
09.12.2021	0.3	Marek Wdowiak (WUT)	Code structure and statistics
10.12.2021	0.4	Krzysztof Marek (WUT)	Rest of chapters 3 and 4
13.12.2021	0.5	Radosław Roszczyk and Kamil Rybiński (WUT)	First review
14.12.2021	0.6	Krzysztof Marek (WUT)	Updates after review and missing content
17.12.2021	0.7	Agris Sostaks (IMCS)	Final review
20.12.2021	1.0	Krzysztof Marek (WUT)	Final version

Executive summary

This document describes the main output of the activity 6.2 in form of the BalticLSC Software Code. The code is available in an open-source form on a public GitHub repository (https://github.com/balticlsc). This documents contains brief descriptions of the repository to help with understanding, use, and further development of code in it.

The BalticLSC Software uses the BalticLSC Platform to perform complex distributed computations and consists of three main components:

- The BalticLSC Computation Tool;
- The Computation Application Language Editor;
- The BalticLSC Admin Tool.

From the technical point of view the BalticLSC Software has been divided into parts stored in separate repositories:

- The BalticLSC Software Backend
- The BalticLSC Software Frontend

The code has been documented in a form of UML models precisely describing the complex architecture of the BalticLSC Software. The model is available at https://www.balticlsc.eu/model/

Table of contents

List of figures

1. Introduction

1.1 Objectives and scope

The objective of this document is to describe the main output of the activity in form of the BalticLSC Software Code. The code is available in an open-source form on a public GitHub repository. This documents contains brief descriptions of the repository to help with understanding, use, and further development of code in it.

The BalticLSC Software uses the BalticLSC Platform to perform complex distributed computations and consists of three main components:

- The BalticLSC Computation Tool;
- The Computation Application Language Editor;
- The BalticLSC Admin Tool.

From the technical point of view the BalticLSC Software has been divided into parts stored in separate repositories:

- The BalticLSC Software Backend
- The BalticLSC Software Frontend

1.2 Relations to Other Documents

This document remains in a very close relation with the joined outputs O5.2 – BalticLSC Admin Tool Technical Documentation, O5.3 – Computation Application Language Manual, and O5.4 – BalticLSC Computation Tool Technical Documentation which are the design artifacts of different components of the BalticLSC Software.

Another important document in O6.1 – The BalticLSC Platform Implementation Report. This document describes the BalticLSC Platform responsible for performing computations commissioned by the BalticLSC Software.

The last crucial document is the O6.3 – BalticLSC Handbook containing the instruction materials, tutorials and documentation on how to use the BalticLSC Software together with the BalticLSC Platform to perform advanced computation with the use of CAL Language and how to implement new Computation Modules that can be used to perform computations.

1.3 Intended Audience and Usage Guidelines

This document is intended for everyone interested in using the BalticLSC Software Code either to just understand how it works from technical side and learn from it or to continue the development by enhancing existing functionalities and adding new ones. If you are interested in learning how to use the BalticLSC to perform computations you should use O6.3 – BalticLSC Handbook which contains practical examples and instructions instead of pure software code.

This document should be used together with the BalticLSC Software Code repository available at: Baltic Large Scale Computing · GitHub (https://github.com/balticlsc)

2. The Architecture of the BalticLSC Software

The BalticLSC consists of the BalticLSC Platform and Software. The BalticLSC Software has been divided into to main parts: Frontend and Backend. The interactions between different components of the BalticLSC have been shown in figure 1.

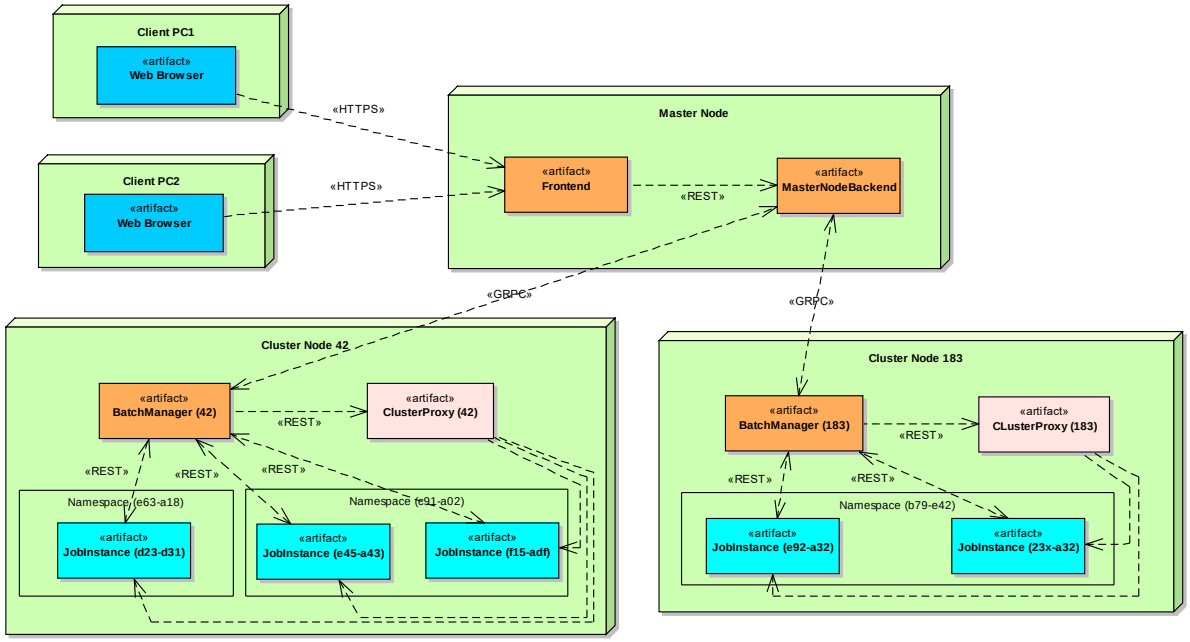

Figure 1 BalticLSC Architecture

The BalticLSC Software has been designed in UML (Unified Modeling Language). During the implementation the design has been frequently updated to include any changes made to the BalticLSC Software during the development phase. The most current version of the BalticLSC Software is available at https://www.balticlsc.eu/model/.

2.1 Model of the BalticLSC Software Frontend

The component model of the BalticLSC Software Frontend has been presented on Figure 2. The BalticLSC Software Frontend is used by 4 Actors: the Administrator (responsible for BalticLSC maintenance), Developer (end-user creating new Computation Applications in CAL), Supplier (of Computation power) and App User (end-user performing computations). Such separations allow for easier extension of the BalticLSC Software in the future. It is also an additional security measure, to ensure the security of data and computations. The User Interface (UI) us modern and user friendly. It's elements have been described in the next subsection.

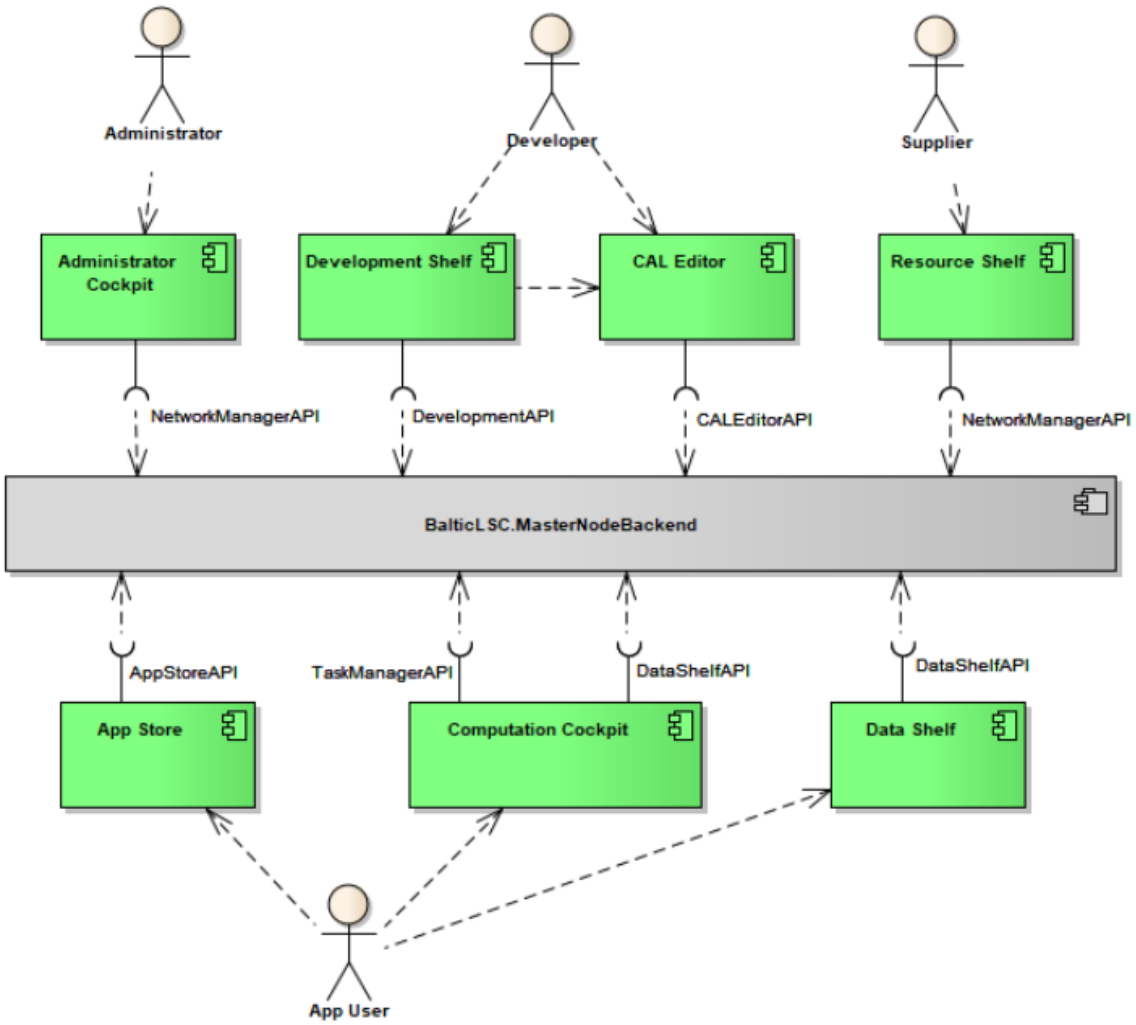

Figure 2 Model of the BalticLSC Software Frontend

2.2 Elements of the BalticLSC Software Frontend

The BalticLSC Software Frontend consists of multiple webpages (each with additional subpages).

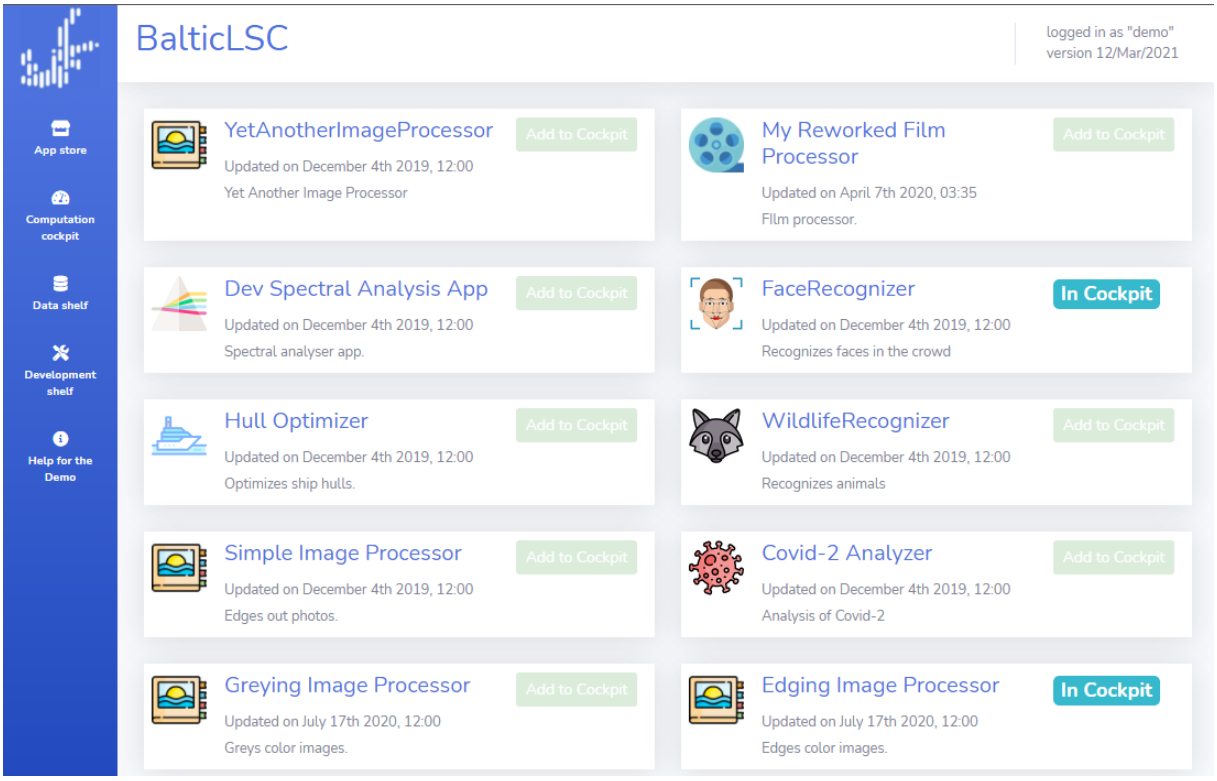

Figure 3 BalticLSC Software Frontend - Application Store

The Application Store is a place where end users can find Computation Application and Computation Modules that they can later use in the Computation Cockpit and CAL Editor.

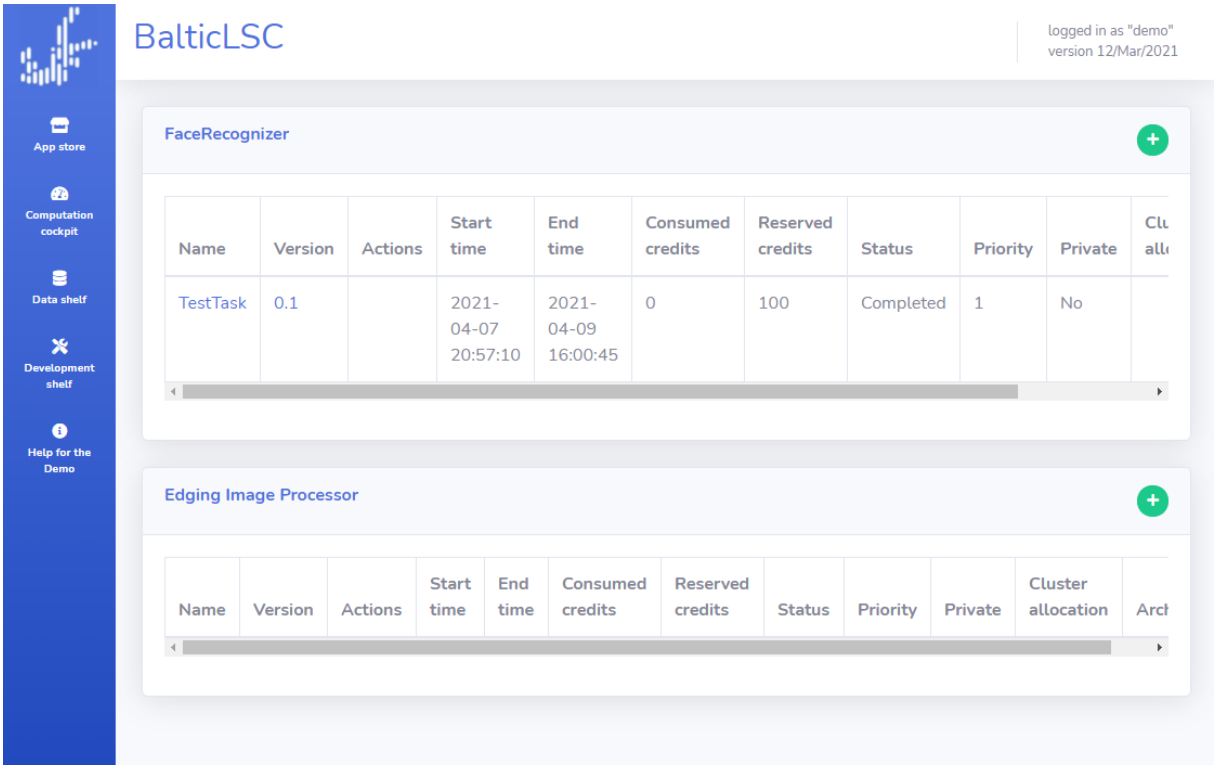

Figure 4 BalticLSC Software Frontend - Computation Cockpit

The Computation Cockpit is the centerpiece of the BalticLSC Software. It allows users to start and monitor their computations. For every Computation Task the end user can check its details, monitor statuses of each instance of Computation Modules used by the Computation Application, and download logs from the BalticLSC Platfrom.

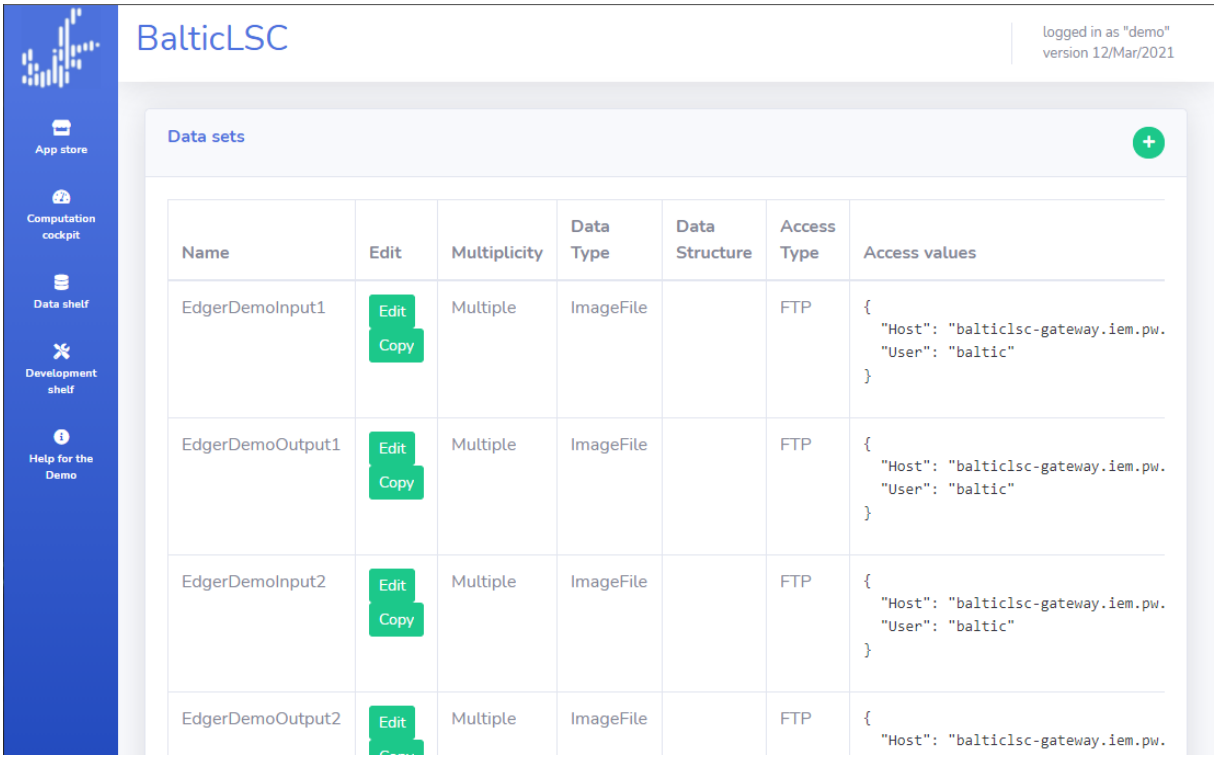

Figure 5 BalticLSC Software Frontend - Data Shelf

The Data Shelf allows the end user to define data sets used by the Computation Applications started in the Computation Cockpit. Every data set can be used as input source and output destination.

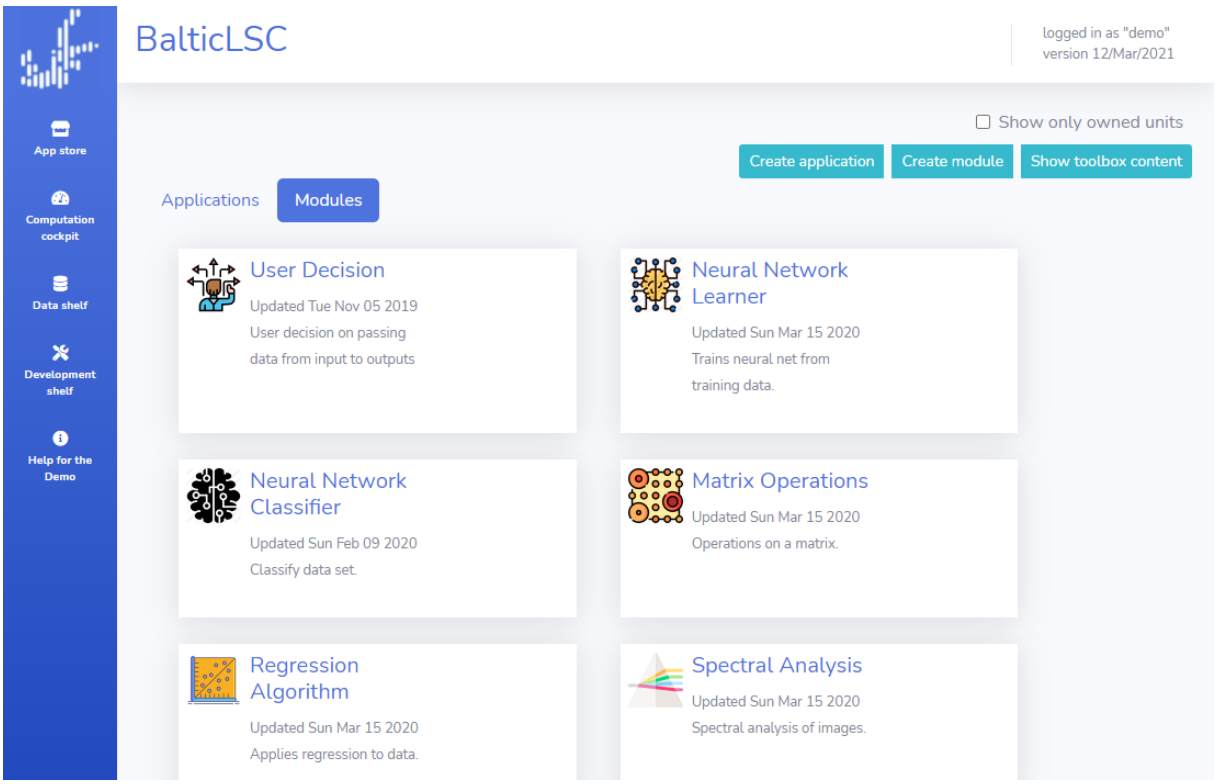

Figure 6 BalticLSC Software Frontend - Development Shelf

The Development Shelf allow the end-users to manage their own Computation Applications and Computation Modules. The end-user can create releases of Computation Applications and Modules which allow them to be used to perform computations. In addition they can manage their toolbox used while developing Computation Applications.

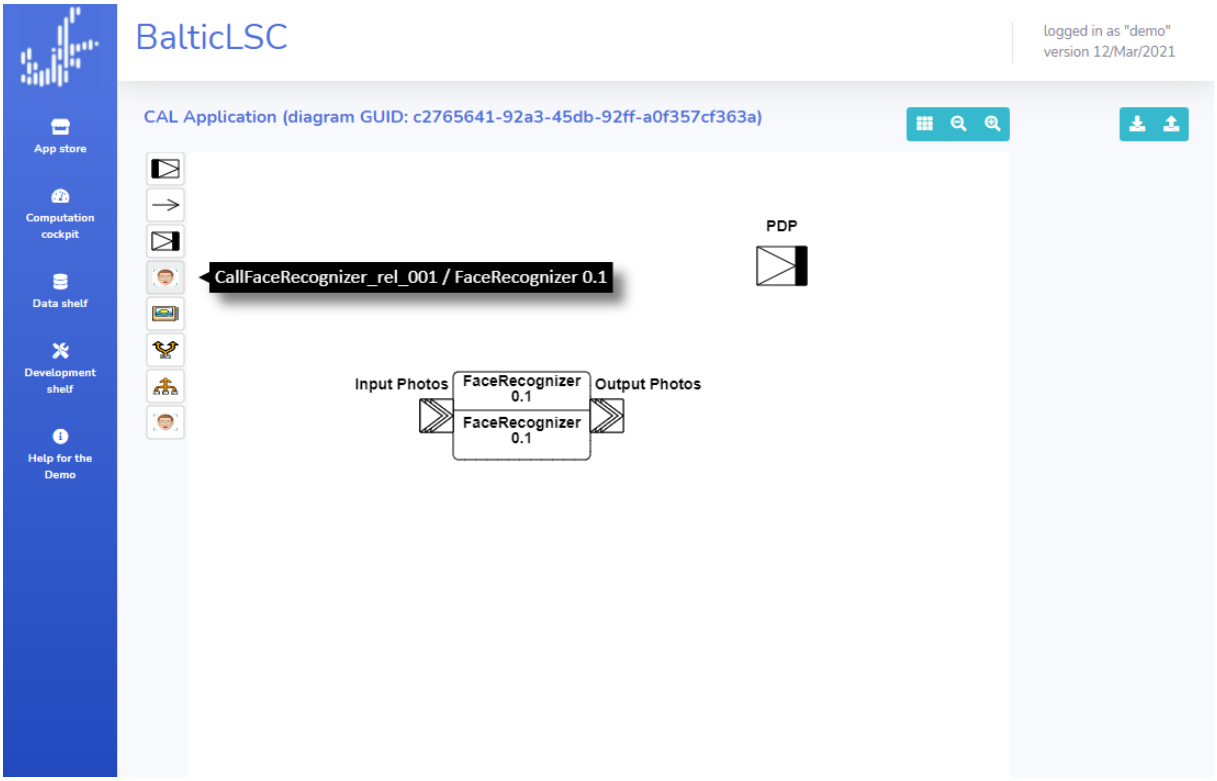

Figure 7 BalticLSC Software Frontend - CAL Editor

The CAL Editor allows for easy development of new Computation Applications and improvements to already existing ones. All the changes are saved in real time and the editor doesn't require any installations on the end user side.

2.3 Model of the BalticLSC Software Backend

The component model of the BalticLSC Software Backend has been presented on Figure 8. The microservices architecture allows for easier improvements and extensions of the BalticLSC Software in the future. The main components are: Task Manager, Unit Manager, Network Manager, Task Processor and Job Broker. The BalticLSC Software Backends processes actions performed by the end-users and communicates with every Computation Cluster Node in the BalticLSC Network to ensure the computations are correctly performed.

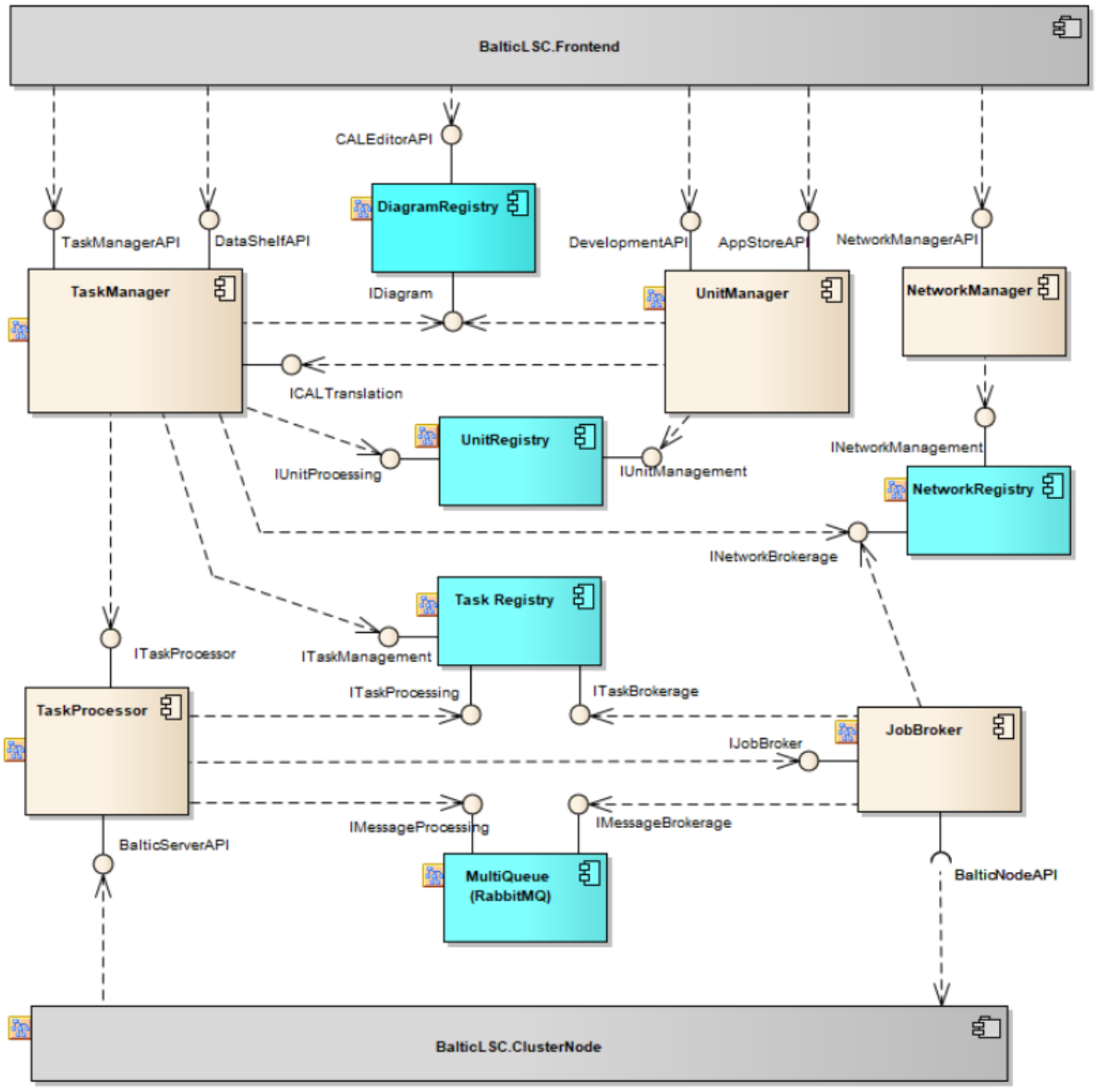

Figure 8 Model of the BalticLSC Software Backend

3. The code of the BalticLSC Software Backend

This section goes through the code structure and contents of the BalticLSC Backend. The Backend is responsible for the "logic" of the BalticLSC Software. It ensures that actions made by the user via the visual interface (BalticLSC Software Frontend) will translate into real activities from, for example, saving changes in user's account setting, storing and validating CAL diagrams, to commissioning specific parts of the entire computations to the selected members of the BalticLSC Network via the BalticLSC Platform.

The source code produced has been committed to the common project source code repository located at: Baltic Large Scale Computing · GitHub(https://github.com/balticlsc)

3.1 The Code in numbers

The BalticLSC Software Backend has been mostly written in C# programming language as shown in Figure 1. PostgreSQL been used as the database to store data for the BalticLSC Software. It is important to note that this database is not used to store any data used by the BalticLSC users to perform computations. It stores data about the users, required to use the BalticLSC and data regarding the CAL Applications defined by the users.

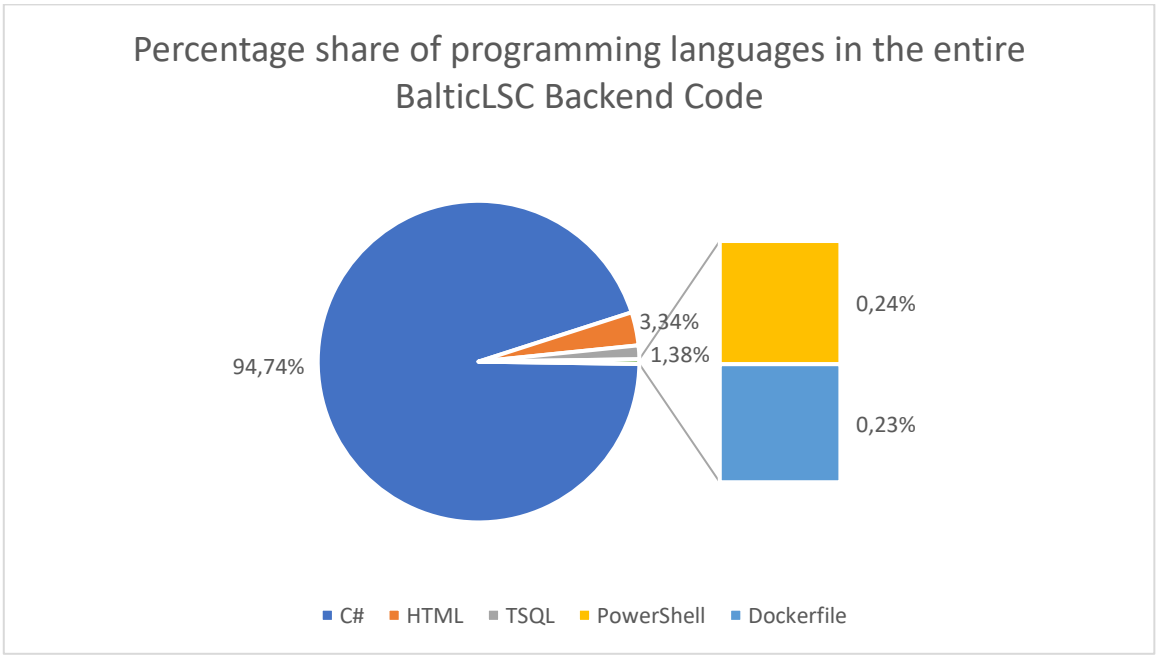

Figure 9 Percentage share of programming languages in the entire BalticLSC Backend Code

```
http://cloc.sourceforge.net v 1.64  T=81.03 s (19.5 files/s, 5979.3 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
JSON                           208              5              0         386801
C#                            1227          10502           7918          66393
XML                             30            282              0           5813
ASP.Net                          3             32             20           1912
MSBuild script                  51            188              0           1306
HTML                             1             82              2           1099
SQL                             15            145              2            927
YAML                             8             30             63            454
Protocol Buffers                 6             62             14            315
PowerShell                      19             23              0             99
Bourne Shell                     6              1              2             25
DOS Batch                        6              0              0             14
-------------------------------------------------------------------------------
SUM:                          1580          11352           8021         465158
-------------------------------------------------------------------------------
```

Figure 10 Detailed BalticLSC Software Backend code statistics

3.2 The structure of the BalticLSC Software Backend repository

```
D:\BALTICLSC\SOURCE\BALTICLSC-BACKEND
├───Attic
│   ├───Baltic
│   ├───Baltic.Core
│   │   ├───Extensions
│   │   └───Utils
│   ├───Baltic.Cryptography
│   ├───Baltic.Database
│   ├───Baltic.Mail
│   │   ├───messagesCreation
│   │   └───resources
│   ├───Baltic.Node
│   │   └───Properties
│   ├───Baltic.Node.Database
│   ├───Baltic.Node.Services
│   ├───Baltic.ProtocolBuffers
│   │   └───Protos
│   ├───Baltic.Server
│   │   ├───Common
│   │   ├───Controllers
│   │   │   └───Models
│   │   ├───Extensions
│   │   ├───Hubs
│   │   ├───Middleware
│   │   ├───Models
│   │   │   ├───App
│   │   │   ├───Billing
│   │   │   ├───Computation
│   │   │   ├───Jobs
│   │   │   ├───Module
│   │   │   └───Resources
│   │   └───Properties
│   ├───docs
```

```
│   ├──Generated
│   │   ├──Baltic.Database
│   │   │   ├──bin
│   │   │   │   ├──Debug
│   │   │   │   └──Release
│   │   │   ├──Entities
│   │   │   ├──NetworkRegistry
│   │   │   ├──Properties
│   │   │   ├──TaskRegistry
│   │   │   └──UnitRegistry
│   │   ├──Baltic.DataModel
│   │   │   ├──Accounts
│   │   │   ├──CAL
│   │   │   ├──CALExecutable
│   │   │   ├──Diagram
│   │   │   ├──Execution
│   │   │   ├──Properties
│   │   │   └──Resources
│   │   ├──Baltic.Engine
│   │   │   ├──bin
│   │   │   │   ├──Debug
│   │   │   │   └──Release
│   │   │   ├──Cluster
│   │   │   │   ├──BatchManager
│   │   │   │   ├──Cluster
│   │   │   │   ├──Common
│   │   │   │   ├──Job
│   │   │   │   └──JobManager
│   │   │   ├──JobBroker
│   │   │   ├──MultiQueue
│   │   │   ├──Properties
│   │   │   ├──TaskManager
│   │   │   ├──TaskProcessor
│   │   │   ├──UnitManager
│   │   │   └──Utilities
│   │   └──Baltic.Service
│   │       ├──Controllers
│   │       ├──Models
│   │       └──Properties
│   ├──SignalRTestingApplication
│   │   └──CommunicationDefiners
│   └──Tests
├──RegisterToNode
├──Test.Cryptography
└──Test.Mail
├──Baltic.CalEditorApi
│   ├──ApiTest
│   ├──Controllers
│   ├──DTO
│   ├──Entities
│   ├──Model
│   ├──Properties
│   ├──Scripts
│   └──Tables
│   ├──CompartmentStyleTables
│   ├──CompartmentTables
```

```
│   ├──ElementStyleTables
│   ├──LineStyleTables
│   └──LocationTables
├──Baltic.CommonServices
│   └──Properties
├──Baltic.Consul
│   ├──Exceptions
│   └──Properties
├──Baltic.Core
│   ├──Extensions
│   └──Utils
├──Baltic.Cron
├──Baltic.Database
│   ├──Migration
├──Baltic.DataModel
│   ├──Accounts
│   ├──CAL
│   ├──CALExecutable
│   ├──CALMessages
│   ├──Diagram
│   ├──Execution
│   ├──Properties
│   ├──Resources
│   ├──Types
│   └──UtilExtensions
├──Baltic.DiagramRegistry
│   ├──DataAccess
│   └──Properties
├──Baltic.EmptyModule
│   └──Properties
├──Baltic.Engine
│   ├──JobBroker
│   ├──Properties
│   ├──TaskManager
│   ├──TaskProcessor
│   └──UnitManager
├──Baltic.Guardian
│   └──Properties
├──Baltic.Language
│   └──Properties
├──Baltic.Mail.Service
│   └──Properties
├──Baltic.NetworkManager
│   ├──ApiTests
│   ├──Controllers
│   └──Models
├──Baltic.NetworkRegistry
│   ├──DataAccess
│   └──Properties
├──Baltic.Node
│   ├──logs
│   └──Properties
├──Baltic.Node.BatchManager
│   ├──ApiTests
│   ├──Controllers
│   ├──Models
```

```
│   ├───Properties
│   └───Proxies
├───Baltic.Node.ClusterProxy.Swarm
│   ├───Controllers
│   ├───logs
│   ├───Properties
│   └───Services
├───Baltic.Node.Database
├───Baltic.Node.Engine
│   ├───BatchManager
│   ├───DataAccess
│   ├───DataModel
│   ├───Job
│   ├───Properties
│   └───ServerAccess
├───Baltic.Node.ModuleManager
│   ├───Controllers
│   ├───logs
│   └───Properties
├───Baltic.ProtocolBuffers
├───Baltic.Queue
│   ├───MultiQueue
│   └───Properties
├───Baltic.Security
│   ├───ApiTest
│   ├───Auth
│   ├───Controllers
│   │   └───Models
│   ├───Entities
│   ├───Scripts
│   ├───Tables
│   └───Utils
├───Baltic.Server
│   ├───Controllers
│   ├───logs
│   ├───Properties
│   └───Swagger
├───Baltic.TaskManager
│   ├───ApiTests
│   ├───Controllers
│   ├───Models
│   ├───Properties
│   └───Proxies
├───Baltic.TaskRegistry
│   ├───DataAccess
│   ├───Entities
│   ├───Properties
│   ├───Scripts
│   └───Tables
├───Baltic.Types
│   ├───DataAccess
│   ├───Entities
│   ├───Models
│   ├───Protos
│   └───QueueAccess
├───Baltic.UnitManager
```

```
│   ├──ApiTests
│   ├──Controllers
│   └──Models
├──Baltic.UnitRegistry
│   ├──DataAccess
│   ├──Entities
│   │   └──ComputationAccounts
│   ├──Models
│   │   ├──Inne
│   │   ├──Types
│   │   └──UserAccount
│   ├──Properties
│   ├──Scripts
│   └──Tables
│   └──Obsolete
│   ├──Compuation Accounts
│   └──Sql
│   └──Computation Accounts
├──Baltic.UserRegistry
│   └──Properties
├──Baltic.Web
│   ├──Common
│   ├──Extensions
│   ├──Interfaces
│   ├──Middleware
│   ├──Module
│   └──Properties
├──Compose.Linux
│   └──conf
│   ├──consul
│   ├──rabbitmq
│   └──vault
├──Compose.Windows
│   ├──baltic-yamls-localhost
│   │   ├──configs
│   │   │   └──consul
│   │   └──logs
│   │   ├──balticlsc-node
│   │   ├──balticlsc-server
│   │   └──cluster-proxy-swarm
│   ├──baltic-yamls-server
│   ├──fluentd-docker-build
│   └──ftp_data
│   └──images
│   ├──in
│   └──out
├──dbTransactions
├──Diagrams
├──docs
├──Libraries
│   └──Docker.DotNet
│   ├──Endpoints
│   ├──Microsoft.Net.Http.Client
│   ├──Models
│   └──Properties
├──Patches
```

```
├──Test.Engine
│  └──Properties
└──Test.HttpClientTest
└──Properties
```

4. The code of the BalticLSC Software Frontend

The BalticLSC Software Frontend is responsible for the visual side of the BalticLSC Software (seen directly by the end-user). It consists of interactive pages such as: App Store, Data Shelf, Computation Cockpit, Development Shelf and CAL Editor. All with necessary subpages. BalticLSC Software presents information from BalticLSC Software Backend and allows end users to perform actions by interacting with the User Interface.

The source code produced has been committed to the common project source code repository located at: Baltic Large Scale Computing · GitHub(https://github.com/balticlsc)

4.1 The Code in numbers

The BalticLSC Software Frontend has been mostly written in JavaScript and Vue programming languages as shown in Figure 2. Used solutions allow for very interactive User Interface including the advanced CAL Editor fully working in the BalticLSC Software Fronted which allows for creating and editing of CAL diagrams of Computation Applications without the requirement of installing anything on the end-users computer.

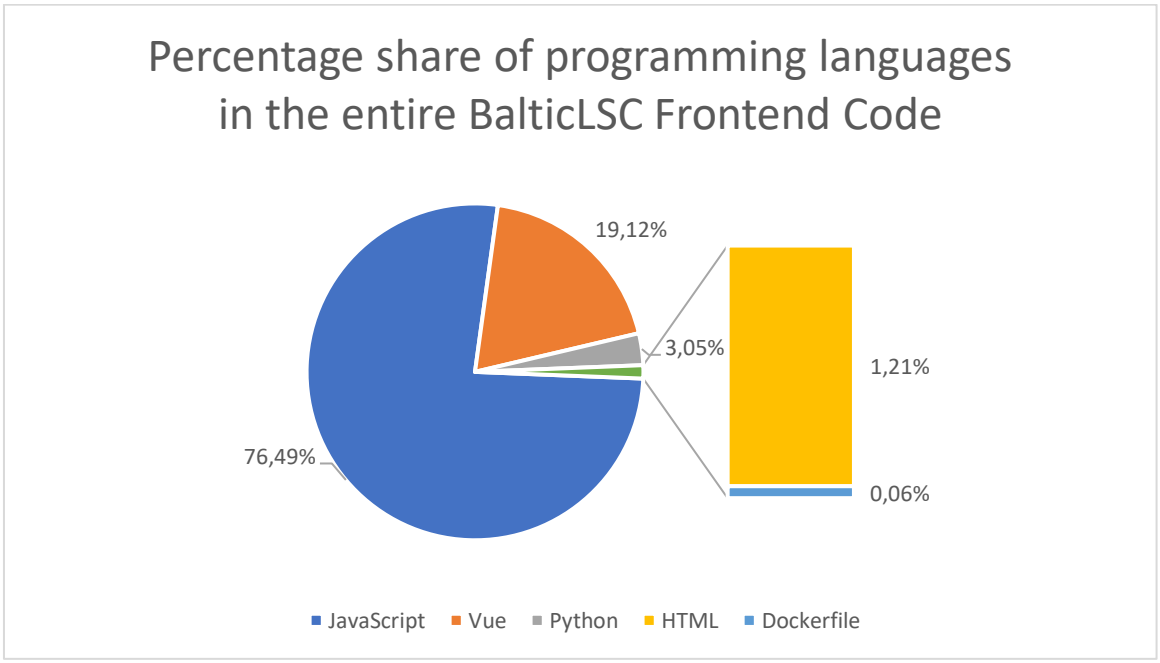

Figure 11 Percentage share of programming languages in the entire BalticLSC Frontend Code

```
http://cloc.sourceforge.net v 1.64  T=25.22 s (10.8 files/s, 3974.2 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Javascript                     122          11869           8687          48597
SASS                           107           1804           1149           9989
CSS                             17           3515             69           7789
LESS                            18            502             55           4585
Python                           3             74             12            876
HTML                             2            181            128            172
JSON                             2              0              0            167
Bourne Shell                     1              0              0              8
YAML                             1              0              0              4
-------------------------------------------------------------------------------
SUM:                           273          17945          10100          72187
-------------------------------------------------------------------------------
```

Figure 12 Detailed BalticLSC Software Frontend code statistics

4.2 The structure of the BalticLSC Software Frontend repository

```
D:\BALTICLSC\SOURCE\BALTICLSC-FRONTEND
├───backend
│   └───CALEditorBackendTests
└───sources
    ├───public
    │   ├───js
    │   ├───static_mine
    │   │   ├───css
    │   │   ├───js
    │   │   └───vendor
    │   │       ├───bootstrap
    │   │       │   └───js
    │   │       ├───jquery
    │   │       └───jquery-easing
    │   └───vendor
    │       ├───bootstrap
    │       │   ├───js
    │       │   └───scss
    │       │       ├───mixins
    │       │       ├───utilities
    │       │       └───vendor
    │       ├───fontawesome-free
    │       │   ├───css
    │       │   ├───js
    │       │   ├───less
    │       │   ├───scss
    │       │   ├───sprites
    │       │   ├───svgs
    │       │   │   ├───brands
    │       │   │   ├───regular
    │       │   │   └───solid
    │       │   └───webfonts
    │       ├───jquery
    │       └───jquery-easing
    └───src
```

```
├──ajoo
│  ├──Editor
│  ├──Elements
│  │  ├──Boxes
│  │  │  └──DrawingShapes
│  │  ├──Lines
│  │  │  └──routing
│  │  └──Swimlane
│  └──Selection
├──assets
│  └──smashing-freebies-smallicons-icon-set
│  ├──32
│  │  └──PNG
│  │  ├──32
│  │  └──@2x
│  ├──64
│  │  └──PNG
│  │  ├──64
│  │  └──@2x
│  └──SVG
├──layouts
├──mycomponents
│  ├──AdminPanel
│  ├──AppDetails
│  │  ├──ComputationApplicationRelease_Edit
│  │  └──ComputationApplication_Edit
│  ├──AppShelf
│  ├──AppStore
│  ├──BenchmarkDetails
│  ├──Benchmarks
│  ├──ClusterMachines
│  ├──ComputationApplication
│  ├──ComputationApplicationReadOnly
│  ├──ComputationApplications
│  │  └──AddModuleRelease
│  ├──ComputationModule
│  │  ├──ComputationModuleRelease_Edit
│  │  ├──ComputationModule_Edit
│  │  └──PinTable
│  ├──ComputationModules
│  ├──DataShelf
│  ├──JobDetails
│  ├──Jobs
│  ├──Login
│  ├──mixins
│  ├──Notifications
│  ├──Person
│  ├──Profile
│  ├──RegisterUser
│  ├──ResourcesCredits
│  ├──ResourcesShelf
│  ├──TaskDetails
│  └──TopBar
├──plugins
├──utils
└──views
```