BalticLSC Platform Technical Recommendations

Summary of BalticLSC Platform Technical Feedback Workshops

Version 1.0

Interreg Baltic Sea Region
EUROPEAN UNION
EUROPEAN REGIONAL DEVELOPMENT FUND

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 Platform Technical Recommendations

Work package	WP4
Task id	4.5
Document number	O4.5
Document type	Output
Title	BalticLSC Platform Technical Recommendations
Subtitle	Summary of BalticLSC Platform Technical Feedback Workshops
Author(s)	Henri Hanson(TSP), Madara Siliņa (IMCS)
Reviewer(s)	Agris Šostaks (IMCS), Krzysztof Marek (WUT)
Accepting	Michał Śmiałek
Version	1.0
Status	Final version

Interreg
Baltic Sea Region
EUROPEAN UNION
EUROPEAN REGIONAL DEVELOPMENT FUND

History of changes

Date	Ver.	Author(s)	Change description
05.01.2020	0.1	Madara Siliņa (IMCS)	Feedback regarding the Kuberneted & Rancher cluster setup using the first Platform version
06.12.2021	0.2	Henri Hanson (TSP)	Workshop feedback description
20.12.2021	0.3	Henri Hanson (TSP)	Document formatting and remaining content
21.12.2021	0.4	Agris Šostaks (IMCS)	Review
27.12.2021	0.5	Krzysztof Marek (WUT)	Review
28.12.2021	1.0	Henri Hanson (TSP)	Final version

Executive summary

The overall aim for the Baltic LSC activities is developing and providing a platform for truly affordable and easy to access LSC Environment for end-users to significantly increase capacities to create new innovative data-intense and computation-intense products and services by a vast array of smaller actors in the region.

Current report contains various best practices in designing BalticLSC Environment. These best practices are based on comments, issues and questions formulated by partners and external consultants. This provides additional, practical guidance for potential further developers of the LSC Platform. The recommendations will pertain to descriptions of both the structure and dynamics of the tools.

The output document gathers and organizes the workshop results according to the structure of design documentation (see Output 4.3-4.4). It contains guidance on applying the technologies (platforms, OS) that should be used to develop the prototype. It also addresses issues of assuring compliance with the end-user requirements (see Output 3.2). Special care will be taken to provide guidance on applying the design models trans-nationally by organizations from the Baltic Sea region.

BalticLSC Platform Feedback Notes document is based on work done within activities of the BalticLSC Work Package 4 (WP4) - Activity 4.3 (A4.3): Design of the BalticLSC Platform, A4.4: Design of the BalticLSC Operating System, and A4.5: BalticLCS Platform technical feedback workshops.

As Covid19 situation impacted the workshop organization, most input was gathered based on online workshops and individual meetings instead of originally planned physical meetings.

Table of contents

1. Introduction

1.1 Objectives and scope

The objective of the document is to describe gathered feedback regarding the BalticLSC Platform.

1.2 Relations to Other Documents

The BalticLSC Platform Technical Recommendations document provides the description of experience using the BalticLSC Platform. It is based on major outputs of the project (mainly outputs from Activities 3-4) and feedback from end-users' perspective. The output document gathers and organizes the workshop results according to the structure of design documentation (Output 4.3-4.4). It contains guidance on applying the technologies (platforms, OS) that should be used to develop the prototype. It also addresses issues of assuring compliance with the end-user requirements (see Output 3.2).

1.3 Intended Audience and Usage Guidelines

This document is one of the project outputs for future BalticLSC Platform and Software development activities foreseen for improving the BalticLSC Platform.

2. BalticLSC Environment

The BalticLSC environment consists of an online platform developed by the project technical partners which is designed to affect two major groups across the Baltic Sea region, namely:

1. Large **LSC service providers** who can offer their computational power to perform calculations by joining the network. Even start-ups and SMEs can take up BalticLSC specifications and emerge as LSC service providers by creating their own local computing centers. Furthermore, several of such centers can combine to form a network and increase their computing capability which can be distributed across Baltic Sea region.
2. **End-users** such as SMEs, start-ups, research and development centers, design centers and others can utilize the platform to develop and test their own applications and carry out complex computations and calculations thus reducing the time to market in the delivery of innovative products and services. With the help of this environment, the end-users can now make use of LSC services easily which were previously difficult to access.

The project strives to develop a solution that is replicable, transferable, sustainable, and easy to implement. Building the environment includes decisions on hardware, platform components and software to utilize the potential and provide access to both partners as well as potential end-users of the Baltic LSC service.

Overall, more than 100 companies for potential use cases (see output 3.5 for detailed use cases descriptions) have been approached during the duration of the project, including participants at project events, individual meetings with potential service providers/end-users, targeted online marketing campaigns etc.

From these, more than 30 companies have had in-depth discussions on using the Baltic LSC services and 8 successfully run cases, where there is a chance for continuous future services. The small turnaround from companies approached to successfully run cases stems from several factors, including from one side the low level of knowledge in the companies (e.g. no data scientists available or the low quality of data available), but also on the platform risk, as Baltic LSC is run as a cross-border service by publicly funded partners.

Additionally, the Baltic LSC platform itself is not yet flexible enough for some specific calculations, needing further development. As a result, although the business plan was developed, additional investments into not only building the platform itself, but also train the wider client base in the possibilities and tools available on the market for the gain of their business. Building Baltic LSC service as a long-term sustainable business in the future will continuously be the challenge in this field.

3. Key components of the BalticLSC Platform

3.1 Platform Architecture

The BalticLSC Platform consists of several different components which are described in detail in Baltic LSC Output 4.3, including general description, management, installation, configuration and setup of the Baltic LSC Platform. Additionally, it provides an overview of billing based on utilisation and API usage specifics. The main platform was set up during the Baltic LSC project based on equipment procured during the project and available at project partner premises. In the future, external resource providers can join the platform with their own hardware.

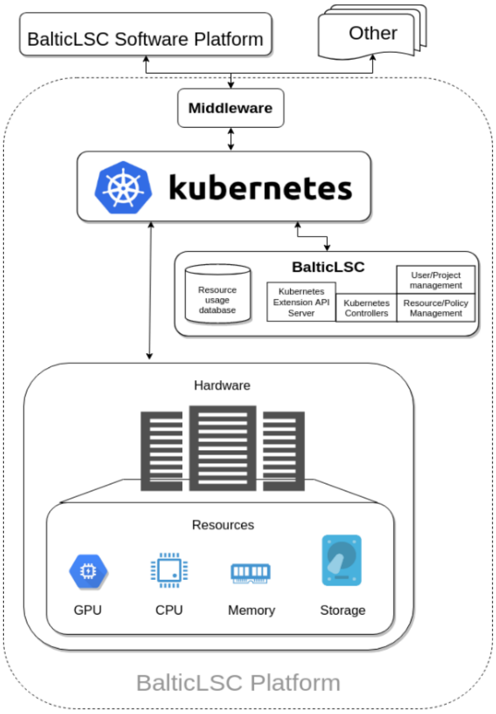

Figure 1 BalticLSC Platform Architecture

Key technologies used:

- **Docker containers –** industry standard for containers, portable anywhere. Lightweight, secure, standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

- **Kubernetes** (K8s) for container orchestration - an open-source system for automating deployment, scaling, and management of containerized applications.

- **Rancher** for cluster management –100% free and open source software with native upstream Kubernetes. Baltic LSC Platform manager uses Rancher for cluster management and monitoring. It is a complete container management platform with a Graphic User Interface (GUI) for user and project management, authentication and policing.

General Platform description:

- distributed computing and networking platform
- applications are executed in containers
- REST API – application programming interface is available for end-users
- computing resources can be shared in a single cluster and resource usage is being tracked
- user policing and accounting available for platform manager

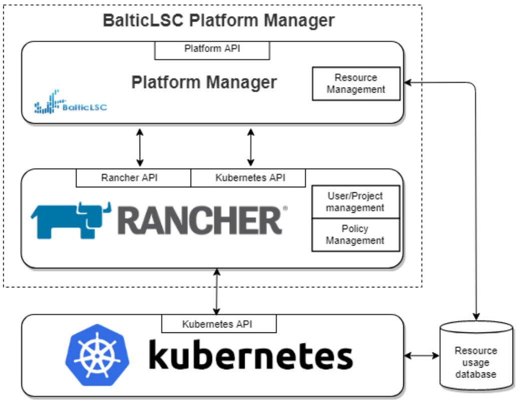

Figure 2 BalticLSC Platform Manager Architecture for Configuration with Kubernetes

3.2 Platform Software

The BalticLSC Software development plans and results are described in detail in Baltic LSC outputs 5.2-5.4, including design of the admin tool, the computation application language (CAL) and design of the computational tool available for the end-users. The current report provides general overview without going into too much detail - structural model of the designed software tools, namely, the component diagrams describe the main components reflecting high-level componentization of software. Industry standard - UML Component and UML Class diagrams have been used to provide comprehensible and repeatable solutions. The software and hardware components form the Baltic LSC platform.

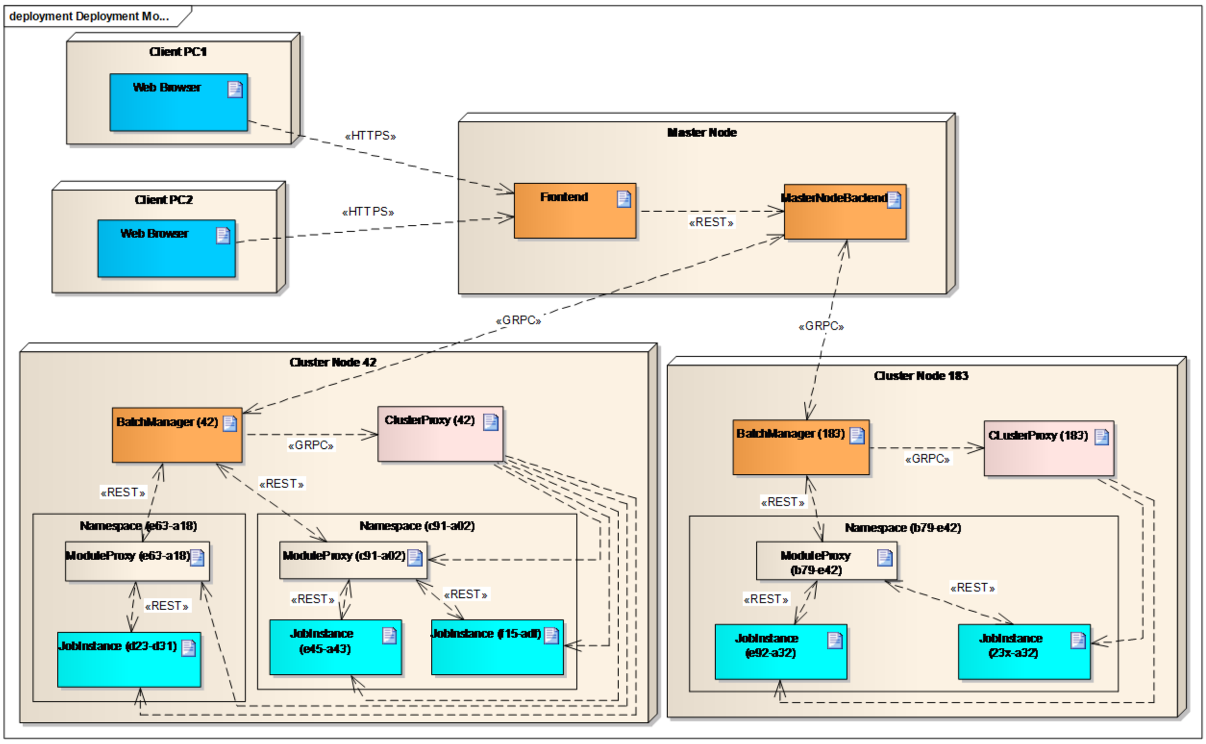

Figure 3 Overview of the BalticLSC Software

Master node is a central access point of the BalticLSC Network, providing users interface (Frontend) and functionality (Backend). **Master Node** is a software component which provides BalticLSC Software user interface via **FrontEnd**: CAL Editor, App Store, Computation Cockpit, Data Shelf, Resource Shelf, Development Shelf, and implements the main functionality of the BalticLSC Software via **Master Node Backend**: computation application compilation and distribution to computation resources, computation application management, user management and security, etc.

Master node is securely accessed using web browser from the client device. Communication between Frontend and Master Node Backend components is done using REST protocols. Master Node Backend distributes the particular computations to the available computation resources.

The main components of the Backend are divided into two categories (see **Błąd! Nie można odnaleźć źródła odwołania.**4) – grey boxes are components which implement the BalticLSC Network functionality, but blue boxes are components for the infrastructure – data registries and utilities.

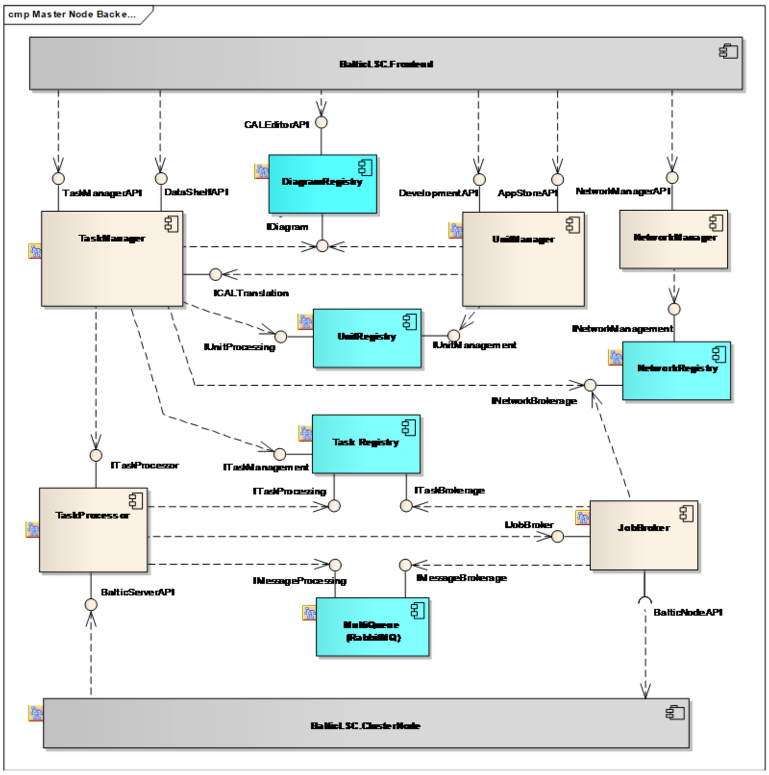

Figure 4 Components of Backend

Frontend provides user interface to the various actors of the BalticLSC Network. Frontend is split in several components visible on Fig 9.

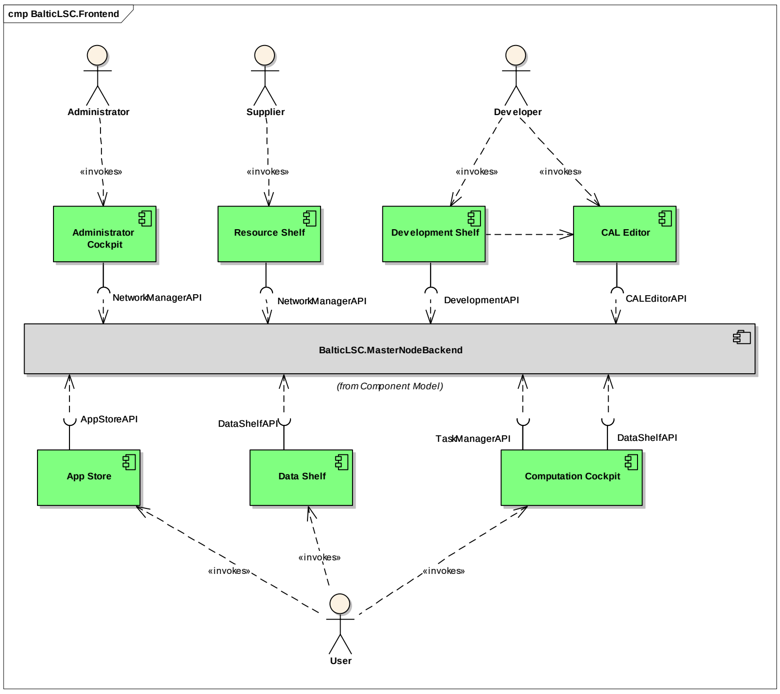

Figure 5 Component Diagram for the FrontEnd

End-users have functionality available to create computation applications out of readymade modules, and to create modules itself.

4. End-user workshop feedback

This section contains the end-user feedback and notes on using the technologies intended for the use within the BalticLSC Platform – Kubernetes and Rancher. The experience of installing the technologies and running Docker containers is also described here.

4.1 Baltic LSC Platform feedback from workshops and individual meetings

Initial idea in the Baltic LSC project was to organise a lot of end-user workshops, but as Covid hit hard all over the world, for increasing risks, mainly the focus was on individual contacts with the end-users and only partly the planned workshops were carried out.

As the service was new to the market, it was not easy to find enough users. However, more than 100 companies have been approached during the project, with more than 30 companies having in-depth discussions on using the Baltic LSC services and 8 successfully run cases, where there is a chance for continuous future services.

Feedback from the discussions with end-users::

- low level of knowledge in the companies – most potential end-users know, there is a lot of operational data available running their daily activities and processes, but there is low usage of this data for improving the potential of the company or improve or develop new services, etc.
- no data scientists available – although, IT staff or even departments are available, these usually do not utilize data scientists for working with the data.
- low quality of data available – the data is usually not available in an easy-to-use, straightforward, operational format, but needs heavy adjustments and cleaning before using for LSC calculations.
- platform risk – BalticLSC is new to the market and without a clear private sector perspective, but rather open in its future developments, thus there is a high risk for the end-users, that the service might not be available 100%.
- no clear pricing model – the users of the system do not get a full price in advance, as the service might be very time-consuming. It is very hard to calculate the price for end-user before using the service.

4.2 Installing Kubernetes and Rancher – novice users' perspective

For users, that have not used **Kubernetes** and **Rancher** before, tutorials from *Katacoda* (https://www.katacoda.com/courses/kubernetes) were quite helpful for getting started with *Kubernetes*. Most of the functionality is already accessible through *Rancher*.

After that, tutorials were used for installing *Rancher* and *Kubernetes*. At first, 4 computers with *Ubuntu Server 18.04*. installed on them were used. One of them as a master node with *Rancher* installed on it (*SSH* was used*)*. It is suggested to use Virtual machines for deploying control plane and *etcd* nodes, e.g. *Virtualbox* for virtualization.

After installing *Docker* on all of the nodes and *Rancher* on the master node, the *Rancher* GUI was accessible. When trying to add other nodes to the created cluster, there was a problem with ports, but after making a new cluster, it did not appear again. After deleting the cluster and before adding nodes to the new cluster, the nodes were "cleaned". Removed all docker containers, images, volumes, etc. This link was helpful: https://rancher.com/docs/rancher/v2.x/en/cluster-admin/cleaning-cluster-nodes/

After "cleaning" the nodes, they also ran with no problems in the new cluster. There was no problem with making new users and creating a new security policy.

Installing *kubectl* on the master node was successful as well on a computer that is not in the cluster. After providing *kubeconfig* file, copied from *Rancher* GUI, created necessary directories, and after that cluster details were available, deployments could be created etc.

After enabling monitoring, *Grafana* was available. It is necessary to install *Operator Metering* on each cluster to make reports later.

4.3 Running Docker Containers - novice users' perspective

There are 3 options from where to get the *Docker* image for deployment:

- official *DockerHub* images,
- private *DockerHub* images, and
- private *Docker* repository images.

There was no problem at all with running official images in the cluster. For running *Docker* images from the private *DockerHub* repository, it was not clear where to put *Docker* credentials (Rancher GUI). Providing only *Docker* username, repository name and tag was not enough. The linked tutorial suggested setting *imagePullSecrets* as "*regcred*" in YAML file, and it worked: https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/

For accessing provided API actions, proxy was created in *kubectl* and actions were run in *Postman*.

Further research is needed on a project called "*Kubeflow*" to make machine learning projects easier to deploy in a *Kubernetes* cluster.

Advice for testing the *Kubernetes* cluster - use *Chaoskube*, which tests how a system behaves under arbitrary pod failures. *Chaoskube* periodically kills random pods in *Kubernetes* cluster, to check if the cluster is maintaining the number of replicas correctly.

5. Summary

The overall aim for the Baltic LSC activities is developing and providing a platform for truly affordable and easy to access LSC Environment for end-users to significantly increase capacities to create new innovative data-intense and computation-intense products and services by a vast array of smaller actors in the region.

As Covid19 situation impacted the workshop organization, most input was gathered based on online workshops and individual meetings instead of originally planned physical meetings.

During the first versions of the BalticLSC Platform joining the cluster as an external provider turned out to be not the easiest task. Staff working with the setting up of Kubernetes, Rancher and Docker containers, to approach the Baltic LSC Platform either as end-users or resource providers, needed to understand how basic cluster management works. Whenever getting stuck with things that are a bit more specific, tutorials are available to understand and set up the *Kubernetes* cluster and *Rancher* for the first time. They have provided to be useful and understandable. This problem with difficult configuration for less experienced cluster administrators has been also later addressed by developing an alternative technological stack for BalticLSC Platform using Docker Swarm technology. It is much simpler solution, ideal for smaller, less experienced resource providers.

The Platform documentation and design documents have proven to be understandable for specialists and experienced administrators, which was the main goal of them. The main concerns regarding the platform were regarding the safety of the data (expressed especially by the inexperienced end-user, not the resource providers) and lack of already established and working pricing model. The concerns regarding the Platform security has been addressed by better explanation of security features BalticLSC Platform has including creating a separate video covering the topic. This has resulted in decrease in such feedback (unexperienced end users were unaware how the data is protected by the technology solutions used by thePlatform). The main result regarding the future improvements and further development of BalticLSC Platform should focus on the financial model implementation. The outside computation power providers would be much more eager to join the network it the financial model would be fully working (not only theoretical). Such concerns are natural for private organizations and couldn’t be fully addressed in this project.

Annex I - Workshops questions

The following set of questions was prepared for the joined workshops (Platform and Software workshops held at the same time with common introduction and later discussions in separate Platform and Software paths):

A) Introduction to the BalticLSC – information we gather every feedback we get

Does performing computation through the platform make sense to you?

What are your expectations regarding this workshop?

B) Introduction to the BalticLSC Environment

Would you apply/use the BalticLSC in your work? And WHY?

What is lacking in the system for you to use it in your work?

In what areas would you apply this? Or in what areas you see the potential to use BalticLSC?

What are your expectations from computation tools (in the future)?

Are the security measures sufficient for your work specifics?

C) Platform Session

Would it be possible to share your resources in this network?

How would you use it in your work?

What kind of benefits do you expect?

Do you have unused resources?

D) Software Session

How would you use it in your work?

Do you see yourself as a client? (Is it something you would pay for?)

What kind of benefits do you expect?

Is it user-friendly?

How do you find data management in the system (Data Pins, Data Shelf etc.)?

Would you develop an application in BalticLSC for yourself/someone else/marketplace?

E) Questions for workshop evaluation:

1) On the scale from 1 to 5 did the workshop content live up to your expectations? (very disappointed) 1 – 2 – 3 – 4 – 5 (exceeded my expectations)

Comments…………………………………………….

2) On the scale from 1 to 5 how did you like the workshop format?

(very disappointed) 1 – 2 – 3 – 4 – 5 (exceeded my expectations)

Comments…………………………………………….

3) Do you have any recommendations to improve the training session?

a. If yes, which?

4) Will you recommend the BalticLSC workshop to others?

a. Yes / No, and why?

EUROPEAN UNION
EUROPEAN REGIONAL DEVELOPMENT FUND