

## Deliverable Report



### Extending Design Thinking with Emerging Digital Technologies

#### Grant Agreement Number 101060231

HORIZON-CL2-2021-TRANSFORMATIONS-01-05

(Integration of emerging new technologies into education and training)

#### Deliverable D4.1

#### Technical Specifications for DT Platform, LA, AR and 3D Printing Technologies

Due date of deliverable: Month 6 - 28 February 2023

Actual submission date: Month 6 - 27 February 2023

Exten(DT) <sup>2</sup> identifier	D4.1 Technical Specifications for DT Platform, LA, AR and 3D Printing Technologies
Name of authors and Affiliations	Alisa Lincke (LNU), Ahmed Taiye Mohammed (LNU), Marianthi Grizioti (NKUA), Sokratis Karkalas (SIMPLE), Mavrikis Manolis (UCL), Marcelo Milrad (LNU), Johanna Velander (LNU), Fisnik Dalipi (LNU)
Work Package/Task no.	WP4/T4.1
Work Package lead	LNU
Reviewers	Boban Vesin (NTNU), Jake Byrne (TCD)
Final editorial review	Shamim Patel (LNU)
Document status	Final version
Confidentiality	Public

## Table of Contents

ABBREVIATIONS	4
LIST OF FIGURES	4
LIST OF TABLES	4
<b>1. SUMMARY</b>	<b>5</b>
<b>2. INTRODUCTION</b>	<b>5</b>
2.1 OBJECTIVES	5
2.2 CORRELATION TO OTHER WPS	5
<b>3. EXTEN.(D.T.)2 PLATFORM SPECIFICATION</b>	<b>6</b>
3.1 DESCRIPTION	6
<b>3.1.1 EXTEN.(D.T.)2 PLATFORM</b>	<b>6</b>
<b>3.1.2 EDUCATIONAL TOOLS AND TECHNOLOGIES</b>	<b>6</b>
<b>3.1.2.1 MaLT2</b>	<b>6</b>
<b>3.1.2.2 SorBET</b>	<b>7</b>
<b>3.1.2.3 ChoiCo</b>	<b>7</b>
<b>3.1.2.4 Virtual Robotics</b>	<b>7</b>
<b>3.1.2.5 nQuire</b>	<b>7</b>
<b>3.1.3 AUTHORABLE LEARNING ANALYTICS</b>	<b>8</b>
<b>3.1.3.1 MaLT2 Extension with LA</b>	<b>8</b>
<b>3.1.3.2 ChoiCo Extension with LA</b>	<b>9</b>
<b>3.1.3.3 SorBET Extension with LA</b>	<b>9</b>
<b>3.1.4 AUTHORABLE LA VISUALIZATION DASHBOARD</b>	<b>10</b>
<b>3.1.4.1 ALA Dashboard for Teachers</b>	<b>11</b>
<b>3.1.4.2 ALA Dashboard for Students</b>	<b>11</b>
<b>3.1.4.3 ALA Dashboard for Scientist/ Researchers</b>	<b>11</b>
3.2 TECHNICAL REQUIREMENTS	11
<b>3.2.1 EXTEN.(D.T.)2 PLATFORM</b>	<b>11</b>
<b>3.2.2 EDUCATIONAL TOOLS</b>	<b>12</b>
<b>3.2.2.1 MaLT2 Requirements</b>	<b>12</b>
<b>3.2.2.1 SorBET Requirements</b>	<b>12</b>
<b>3.2.2.3 ChoiCo Requirements</b>	<b>13</b>
<b>3.2.3 AUTHORABLE LEARNING ANALYTICS</b>	<b>14</b>
<b>3.2.4 AUTHORABLE LA VISUALIZATION DASHBOARD</b>	<b>14</b>
<b>3.2.5 NQUIRE REQUIREMENTS</b>	<b>16</b>
3.3 EXTEN.(D.T.)2 DESIGN	17
<b>3.3.1 ARCHITECTURE OVERVIEW</b>	<b>17</b>
<b>3.3.2 SYSTEM COMPONENTS</b>	<b>18</b>
<b>3.3.3 DATA MODEL DESIGN</b>	<b>20</b>
<b>3.3.3.1 Data Model for Educational Tools</b>	<b>21</b>
<b>3.3.3.2 Sensitive Data</b>	<b>22</b>

<b>3.3.4 INTEROPERABILITY OF LEARNING TOOLS</b>	<b>23</b>
<b>3.3.5 DEPLOYMENT STRATEGY</b>	<b>23</b>
<b>3.3.6 SERVER HARDWARE SPECIFICATION</b>	<b>24</b>
3.4 POLICY, ETHICS, AND SECURITY	25
<b>3.4.1 POLICY</b>	<b>25</b>
<b>3.4.2 ETHICS</b>	<b>25</b>
<b>3.4.3 SECURITY</b>	<b>26</b>
<b>4. CONCLUSIONS</b>	<b>28</b>
<b>REFERENCES</b>	<b>29</b>
<b>APPENDIX A: EDUCATIONAL TOOLS DATA DESCRIPTION</b>	<b>30</b>
<b>APPENDIX B: SERVER CONFIGURATION SPECIFICATION</b>	<b>49</b>

## Abbreviations

Exten.(D.T.) <sup>2</sup>	Extended squared
LNU	Linnaeus University
NKUA	National and Kapodistrian University of Athens
OU	The Open University
UGent	Ghent University
NTNU	Norwegian University of Science and Technology
TCD	Trinity College Dublin
SIMPLE	SIMPLE - SME
UCL	University College London
WP	Work Package
LA	Learning Analytics
AR	Augmented Reality
ALA	Authorable Learning Analytics
AVD	Authorable Visualization Dashboard
DT	Design Thinking
SMILE	Smart Interactive Learning Ecosystem
ML	Machine Learning
VRobotics	Virtual Robotics
IOT	Internet Of Things
DBMS	DataBase Management System
SOA	Service Oriented Architecture
JWT	JSON Web Tokens
IP	Internet Protocol
VPC	Virtual Private Cloud
VPN	Virtual Private Network
API	Application Programming Interface
LTI	Learning Tool Interoperability

## List of Figures

Figure 1. Connection of WP4 to other WPs

Figure 2: Node-RED interface for development

Figure 3: The platform architecture at a high level

Figure 4: Relational data model diagram for Exten D.T.2 platform

## List of Tables

Table 1. Malt2 Requirements

Table 2. SorBET Requirements

Table 3. ChoiCo Requirements

Table 4. General Technical Requirements for ALA dashboard

## 1. SUMMARY

Deliverable 4.1 presents the technical documentation about Exten.(D.T.)2 platform. Exten.(D.T.)2 platform integrates educational tools (such as ChoiCo, SorBET, MaL2, nQuire). The educational tools are extended by some of the emerging technologies such as Augmented Reality (AR), 3D printing/ scanning and Virtual Robotics (VRobotics). In addition, the Exten.(D.T.)2 platform has the Authorable Learning Analytics (ALA) component, and Authorable Visualization Dashboard (AVD) component to leverage digital implementation, monitoring and assessment of digital thinking (DT) projects by teachers in schools. The first section provides the main objectives of this deliverable and the connection to other WPs in the project. The second section provides overall description of the Exten.(D.T.)2 platform, and integrated educational tools, Authorable Learning Analytics (ALA) and dashboard. The third section presents the initial requirements identification, framing, and architectural design of Exten.(D.T.)2 platform and its components (ALA, AVD). In the fourth section, the Exten.(D.T.)2 platform architecture is described, including the identified systems components and conceptual data model. The last section provides the description of the security, privacy, and ethics consideration for the development of Exten.(D.T.)2 platform.

## 2. INTRODUCTION

### 2.1 Objectives

The objectives of this Deliverable are to:

- (a) identify initial requirements for Exten.(D.T.)2 platform that will guide the development and implementation process,
- (b) present the initial Exten.(D.T.)2 platform architecture design and overall system components,
- (c) present initial design of ALA tool and AVD components,
- (d) describe the system specifications of the Exten.(D.T.)2 platform infrastructure, the privacy and security, and
- (e) describe ethics consideration for ALA, and dashboard.

### 2.2 Correlation to Other WPs

The output of WP2 (initial requirements, challenges) provides the initial foundations for the design and development of WP4 enhanced learning technologies (learning tools, nQuire, ALA and LA). The key findings identified by WP2 are: (a) little evidence found in the literature for the use of Virtual Robotics (VRobotics), Augmented Reality (AR) and Learning Analytics (LA) in DT projects (Roumelioti et al., 2022; Leinonen et al., 2020); (b) There is no a single technology/tool which covers all the stages and activities of DT project; (c) most of the existing technologies for DT activities focused on students (as primary users), with limited support for teachers and pedagogical aspects. These challenges and limitations are addressed during the design of WP4-enhanced learning technologies.

The output of WP4 (the developed Exten.(D.T.)2 platform) will be used by WP5 to design and conduct school interventions, as it is shown in Figure 1.

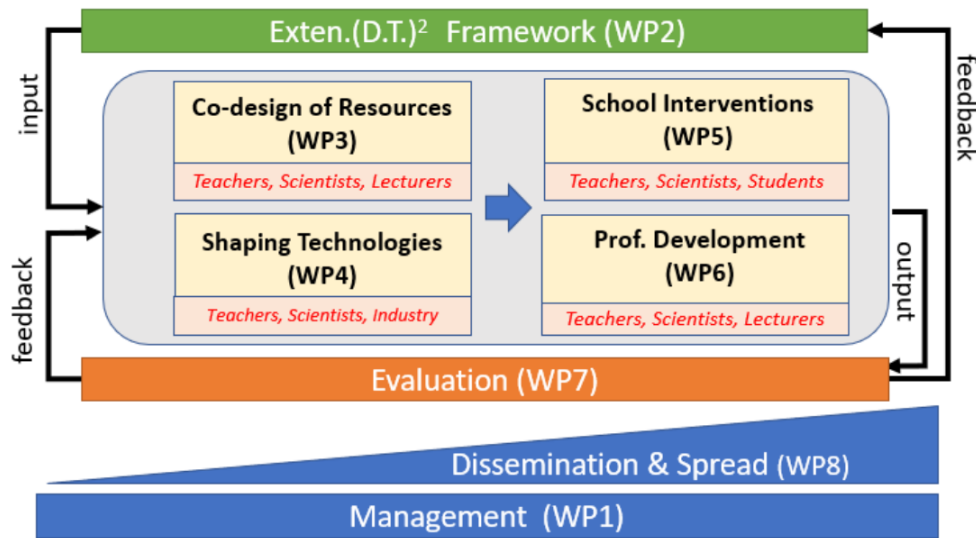


Figure 1. Connection of WP4 to other WPs

### 3. EXTEN.(D.T.)2 PLATFORM SPECIFICATION

#### 3.1 Description

##### 3.1.1 Exten.(D.T.)2 Platform

The Exten.(D.T.)2 platform is hosting an ecosystem of diverse web-based learning environments that will enable the seamless and interactive execution of digital design thinking activities for schools and other educators. The platform should provide all the necessary infrastructure to facilitate seamless integration and interoperability of existing and new components, as well as the ability to intercept and log user interactions, allow processing, analysis, and visualization of data, and enhance learning activities with automated support. The platform to be developed will be based on SMILE (Smart Interactive Learning Ecosystem). SMILE is a platform designed and developed jointly by UCL Knowledge Lab and Simple.

##### 3.1.2 Educational Tools and Technologies

###### 3.1.2.1 MaLT2

MaLT2 (MachineLab Turtleworlds2) is an open-source online tool of symbolic expression in mathematical activity by means of programming for the creation and tinkering of 3D dynamic graphical models. MaLT2 is implemented using JavaScript, HTML5, jQuery, and Bootstrap 5 technologies. Two third-party libraries are used: Three.js (v 50) that is a JavaScript-based library of WebGL for creating the 3D graphics, and codemirror.js (v. 4.0) for the programming editor component.

### 3.1.2.2 SorBET

“SorBET” (Sorting Based on Educational Technology) affords the play and design of Tetris-like sorting games in which the player scores by ‘pushing’ elements falling off the top of the screen to drop into the right container according to the right category. ‘Pushing’ elements can be done by picking and dragging on a screen and will be extended to also include body movements interaction. The gameplay builds on quick decision-making, pattern recognition, and abstraction of the characteristics of falling objects. The tool is web-based and written with Java Script, HTML5, jQuery, and Bootstrap 5. In addition, one third-party library is used: *ML media pipe*<sup>1</sup> javascript library which uses machine learning (ML) for body tracking.

### 3.1.2.3 ChoiCo

“ChoiCo” (Choices with Consequences) is an open-source, online authoring tool that provides an opportunity to play, design and modify choice-driven simulation games related to complex real-life issues. In these games, the player navigates through different map-based areas making choices that affect the game parameters (e.g. Money, Health, Fun etc). In the design mode, the user can modify a game or design a new one by using 3 interconnected computational affordances: a GIS map designer, an editable database, and block-based programming (Kynigos & Grizioti, 2018). It is a web-based tool written with Java Script, HTML5, jQuery, and Bootstrap 5 web technologies. In addition, two third-party libraries are used: (a) *leaflet.js*<sup>2</sup> for creating the GIS-based map editor of the games, and (b) *blockly.js*<sup>3</sup> is a Google library for the block-based programming workspaces.

### 3.1.2.4 Virtual Robotics

Several studies have shown that virtual robotics can provide a successful learning environment for most students learning kinematics, as well as improve their critical thinking and problem-solving abilities (Alsoliman, 2022; González-García et al. 2020, Mistretta, 2022 ). The overview of available open source virtual robotics platforms is in the initial and exploratory stage. One robotics platform such Cyberbotics<sup>4</sup> have been considered. Hence, we have faced several technical integration problems issues with Exten.(D.T.)2 system. Therefore, we continue to search another candidate of VRobotic technology which can be integrated with Exten.(D.T.)2 system.

### 3.1.2.5 nQuire

nQuire (nquire.org.uk) is a web-based community and citizen science platform designed and maintained by the Open University UK.

It enables the general public (any stakeholder) to: a) design, pilot, manage and share research studies, and b) take part in studies designed by others in innovative ways that promote learning and participation. It supports two types of studies (called “missions”): i) confidential missions in which all data are private and only accessed by the author of the mission, and ii) social missions in which all data are open and public for others to read, comment and like.

---

<sup>1</sup> <https://mediapipe.dev/>

<sup>2</sup> <https://leafletjs.com/>

<sup>3</sup> <https://developers.google.com/blockly>

<sup>4</sup> <https://cyberbotics.com/>

In Exten.(D.T.)2, nQuire will be used by students and teachers to support specific stages of the design thinking processes. For example, students could use nQuire to design studies with the aim to understand the needs of the target group for which they are designing a solution for (this is the Empathise stage of design thinking). Students can work collaboratively or individually to design a study on nQuire. The teacher can give feedback on the pilot version of their study helping students improve their questions. Similarly, in the ‘deliver’ phase of design thinking, students can share their final prototypes with their end-users and ask for their feedback and ways they could improve them further.

### 3.1.3 Authorable Learning Analytics

The Authorable Learning Analytics (ALA) component gathers data of student activity generated from the project’s educational tools (described in 3.1.2). The component integrates high-level authoring tools such as drag and drop UI tools, that enable different types of users (teacher, researchers) to author:

- i) which data to be captured for a learning activity, and
- ii) when to provide feedback to students and what feedback for each activity, aiming to enhance students’ engagement with the Design Thinking stages e.g. feedback on how they may increase their empathy or improve the reusability of their prototype as they use the relevant technologies.

#### 3.1.3.1 MaLT2 Extension with LA

MaLT2 will be extended to capture anonymous user interaction data that are meaningful and useful for learning analytics and export them to the Exten.(D.T.)2 platform in real-time. The student ID and group ID is generated by the Exten.(D.T.)2 system and the teacher will document which student belongs to which ID in his/her own document (paper based or stored on her/his own computer). The data are captured based on events that are triggered by user interaction with the environment. While it is possible in principle to record all student actions, because of the programming nature of the user interaction with Malt2 it would be overwhelming both for the server collecting the data and an author or teacher eventually having access to them. As such, we decided to capture certain indicators that can be exploited for further analysis depending on the activity scenario and expected educational outcomes. For more details about the data model see Section 3.3.3.1.

For example, when a user uses the MaLT2 periscopic camera the camera event is triggered by the application and the logging message to be sent to LA looks like this:

```
Logging_message = {
  'id': cameraID,
  'type': "camera",
  'state': { 'view_type': "3dview", 'track_count': 3, 'event_count': 25, 'event_last_count': 3 },
  'event': "camera",
  'timestamp': 1667736467
}
```

Where the ‘state’ object consists from the view type (2D or 3D *type\_view*), the number of models on the scene created by different Logo procedures (*track\_count*), the total number of times the event occurred since activity started (*event\_count*), the times the event occurred



since last Logo code execution (*event\_last\_cout*). The *'timestamp'* variable contains the timestamp in milliseconds when the event occurred (triggered) in Unix format.

So far, a first list of events and relevant data are being captured by MaLT2 and will be sent to the Exten.(D.T.)2 database when their integration is completed. The full list is described in detail in Appendix A: Educational Tools Data Description. The list will be refined after the first-year pilot evaluation of MaLT2 in a school context in which, amongst others, we will survey teachers about the value of capturing these events and data for monitoring and evaluating the learning process.

### 3.1.3.2 ChoiCo Extension with LA

ChoiCo will be extended to capture anonymous user interaction data that are meaningful and useful for learning analytics and send them to the Exten.(D.T.)2 platform in real-time. The data will be captured based on events that are triggered by user interaction with the environment. Similarly to MaLT2, while it is possible in principle to record all student actions in ChoiCo environment, we have decided to capture certain indicators that can be exploited for further analysis depending on the activity scenario and expected educational outcomes.

The method of capturing the events and data sending is the same as described in 3.1.3.1 for MaLT2. For more details about the data model see Section 3.3.3.1.

Below is an example of a logging message that is being sent to the Exten.(D.T.)2 platform when the event *"select point"* is triggered, i.e. when the user selects to visit a game point in play mode:

```
Logging_message = {
  'id': this.gameID,
  'type': "playmode",
  'event': "point_selection",
  'state': {'game_progress': {'fields': {Money: 10, Joy: 50, Health: 30}, 'score': 10} 'value':
    {name: 'Banana', times: 5},event_count: 4},
  'timestamp': 1667736467
}
```

The user data that are being sent with this message are the following. With the variable *'state'*: the current values of the game fields (*'fields'*), the current score (*'score'*), the name of the selected point (*'name'*), the times the player has selected this point from the beginning of that game round (*'times'*), the times the event occurred from the beginning of the activity (*'event\_count'*). The variable *'timestamp'* is the real-time when the event has occurred in Unix format.

So far, a first list of events and relevant data are being captured by ChoiCo and will be sent to the Exten.(D.T.)2 database when their integration will be completed. The full list is described in detail in Appendix A: Educational Tools Data Description.

### 3.1.3.3 SorBET Extension with LA

SorBET will be extended to capture anonymous user interaction data that are meaningful and useful for learning analytics and send them to the Exten.(D.T.)2 platform in real-time. The data will be captured based on events that are triggered by user interaction with the environment.

Similarly to MaLT2, while it is possible in principle to record all student actions in SorBET environment, we have decided to capture certain indicators that can be exploited for further analysis depending on the activity scenario and expected educational outcomes. For more details about the data model see Section 3.3.3.1.

The method for capturing the events and data collection is the same as described in 3.1.3.1 for MaLT2. Below is an example of a logging message that is being sent to the Exten.(D.T.)2 platform when the event “*modify category*” is triggered. For example, when the user modifies the name of a game category in the game database. The user data that are being sent with this message are the following:

```
Logging_message = {
  'id': this.editdbID,
  'type': " database",
  'event': "modify_category",
  'state': {'info': {entriycounter: 6}, 'name': {old: 'Verb' , new: 'Noun', event_count: 6},
  'timestamp': 166773646
}
```

The variable ‘*state*’ capture: the number of existing game categories (‘*entriycounter*’), the old category name (‘*old*’), the new category name (‘*new*’), the times this event occurred from the beginning of the activity (‘*event\_count*’). The variable ‘*timestamp*’ stores the real-time (in milliseconds) when the event has occurred in Unix format.

So far, a first list of events and relevant data are being captured by SorBET and will be sent to the Exten.(D.T.)2 database when their integration will be completed. The full list is described in detail in Appendix A: Educational Tools Data Description. The list will be refined after the first-year pilot evaluation of SorBET in the school context in which, amongst others, we will survey teachers about the value of capturing these events and data for monitoring and evaluating the learning process.

### 3.1.4 Authorable LA Visualization Dashboard

The authorable LA visualization dashboard will effectively analyze and display collected data from the Exten.(D.T.)2 platforms. The dashboard will also allow flexible and customizable data presentation of stored data, making it easier for stakeholders (Students, Teachers, and Scientists/Researchers) to draw useful insights from data. This will allow for a better understanding of the impact and effectiveness of the Exten.(D.T.)2 platform learning experience by making informed decisions about possible future implementations and developments using Design Thinking concepts to develop students' 21st skill sets. In addition, the interface will make emphasis when students are doing well while completing certain tasks. The Exten.(D.T.)2 ALA will try to consider standard and uniform dataset (unification), if data is not unified it poses challenges in many other ALA dashboards. The dashboard will be designed to operate on an open-ended environment for both real-time and non real-time visualization of datasets. Real-time datasets are especially useful for large numbers of students (easy monitoring of students' performance). While the non real-time visualization is good for retrospective analysis by both teachers and researchers/scientists.

#### 3.1.4.1 ALA Dashboard for Teachers

ALA dashboard for teachers will aim at providing them with improved teaching instructions, monitoring students' learning progress to inform and adjust teaching strategies thereby ensuring that teachers provide the most effective instruction for their students. Teachers can also share data from Exten.(D.T.)<sup>2</sup> ALA dashboard with administrators, parents, and other stakeholders, for demonstrating student progress and engaging them with collaborative efforts to improve learning outcomes. The ALA dashboard will emphasize when students performed well in any given task. This feature will enable ease of use and reduce orchestration and cognitive load on the teachers.

#### 3.1.4.2 ALA Dashboard for Students

The ALA student dashboard will empower students with an active role in their own learning by providing them with all the necessary information they need to succeed. By giving students access to data and feedback, the dashboard will support student-driven learning (like learning how to develop their 21st century competencies and giving them the knowledge of Design Thinking processes). This approach will allow students to monitor their own progress and set goals. This is aimed at them reflecting on their learning experiences and understanding their strengths and weaknesses, to make adjustments and improve on their learning performances.

#### 3.1.4.3 ALA Dashboard for Scientist/ Researchers

A comprehensive view of scientific research and performance will be provided to help scientists make data-driven decisions for the Exten.(D.T.)<sup>2</sup> research goals. More importantly, research progress will be tracked over time to monitor performance progress in different Exten.(D.T.)<sup>2</sup> projects and experiments (either from students or teachers). Insights gathered will be shared as research data among peers to collaborate on the Exten.(D.T.)<sup>2</sup> projects.

### 3.2 Technical Requirements

#### 3.2.1 Exten.(D.T.)<sup>2</sup> Platform

The Exten.(D.T.)<sup>2</sup> system will be a platform able to host an ecosystem of diverse web-based learning environments that will enable the seamless and dynamic deployment of digital design thinking activities for schools and educators. The platform will provide all the necessary infrastructure to facilitate integration and interoperability of existing and new components as well as the ability to intercept and log user interactions, allow processing, analysis and visualisation of data and enhance learning activities with automated support. The platform to be developed will be based on SMILE (Smart Interactive Learning Ecosystem). SMILE is a platform designed and developed jointly by UCL Knowledge Lab and Simple.

One of the fundamental assumptions in SMILE is that diverse and heterogeneous software components can be utilised to create an ecosystem of services that will allow the development of learning activities as well as the seamless deployment of them. The various software components that will be used to synthesise the new system are diverse in nature (different architectures, technologies, data formats, communication protocols etc). Although we do have access to the internal implementations of those components our approach is to be as less

invasive as possible and therefore a tight Enterprise Application Integration (EAI) paradigm should not be used.

### 3.2.2 Educational Tools

#### 3.2.2.1 MaLT2 Requirements

MaLT2 will be extended to support 3D printing of the 3D models the user programs. After requirements analysis and review of the state-of-the art technologies and tools the following functional requirements identified and presented in Table 1.

**Table 1. MaLT2 Requirements**

N	Requirement Description
R1.1	Convert the 3D models produced by WebGL to downloadable .obj or .stl files, which are the file types supported by most 3D printers.
R1.2	User interface for downloading the file
R1.3	User interface allowing the user to set some basic parameters for the 3D printing (e.g. texture), facilitating the 3D printing preparation process. The type of parameters will depend on the 3D printer software since they need to be compatible.
R1.4	Capture and export anonymous user interaction data to the Exten.(D.T.)2 platform that are meaningful and useful for learning analytics.
Moreover, we recognized the following non-functional required features for the 3D printer device:	
R1.5	Minimum Build Volume: 250mm x 250mm x 300mm.
R1.6	Support of PLA filaments for safety usage by kids; Preferably support also WOOD and FLEX filaments.
R1.7	Support of eco-friendly/recyclable filaments.
R1.8	User friendly and flexible software.

The exact features & requirements will be finalized after the pilot evaluation of existing technologies with teachers and students and exploration of their needs for technology extension.

#### **Technical Requirements for MaLT2 current version:**

- Supported browsers: Internet Explorer (min v.11), Mozilla Firefox: (min v.39), Safari (min v.8), Google Chrome (min v.31) on desktop and, browsers like Safari, Android, Chrome, Firefox for mobiles.
- Hardware requirements: minimum of 2 GB system memory, at least 1 GB of RAM that supports OpenGL 3.0 (Shader Model 4.0).

#### 3.2.2.1 SorBET Requirements

SorBET will be extended to support AR game play and game design, in the form of body interaction with the falling objects and voice control of the game settings (e.g. speed). The extension will enable students to engage with the game physically, in larger or smaller visual settings and will support multiple persons to simultaneously engage with the same game, supporting embodied collaborative learning experiences. After a first requirements analysis

and review of the state-of-the art technologies and tools the following features will be developed and presented in Table 2.

**Table 2. SorBET Requirements**

N	Requirement Description
R2.1	Gesture recognition by any recent computer camera, not requiring any extra devices.
R2.2	Recognition of at least two players at the same time, to support collaborative play
R2.3	Speech recognition by computer microphone, english language. Preferable to support other languages as well.
R2.4	Support programming of game rules by the end-user in a visual programming environment as block-based programming.
R2.5	Capture and export anonymous user interaction data to the Exten.(D.T.)2 platform that are meaningful and useful for learning analytics.

The exact features & requirements will be finalized after the pilot evaluation of existing technologies with teachers and students and exploration of their needs for technology extension.

Third-party libraries used:

- blockly.js* is a google library that will be used for the block-based programming of the game
- ML media pipe* JavaScript library for body and gestures tracking
- Speech and voice recognition JavaScript library such as *Artyom.js*, *voice-commands.js* or *Pocketsphinx.js*

Technical Requirements for SorBET current version:

- Supported browsers: Chrome, Firefox, Safari 5+, Opera 12+, Internet Explorer 11+ iOS 12+, Safari 12+

**3.2.2.3 ChoiCo Requirements**

ChoiCo will be extended to support AR game play and design, in the form of geolocation data affecting the game content and flow. For instance, game parameters could be affected by the temperature or the traffic in an area in real time, or the game designer could automatically insert different kind of maps and points of interest from their current location.

After requirements analysis and review of the state-of-the art technologies and tools the following features need to be developed presented in Table 3.

**Table 3. ChoiCo Requirements**

N	Requirement Description
R1	Include interactive map tiles that support different map views, e.g. city, satellite, ground, 360.
R2	Use a geolocation API to gather real-time information, e.g. traffic, weather.
R3	Enable user as game designer to display an interactive map of their current location along with some information e.g. points of interest, monuments.

The exact requirements will be finalized after the pilot evaluation of existing technologies with teachers and students and exploration of their needs for technology extension.

Third-party libraries used:

A geolocation API compatible with leaflet. Options: Google maps API (pros: widely used app, has many features that other APIs don't, e.g. routes, places, traffic, supports maps around the world, easy to learn and develop, cons: charges after 28,500 maploads per month), leaflet geolocation plugins (pros: free, cons: not many map types supported, issues or lack of information in some regions, not many data available e.g. routes, traffic)

**Technical Requirments for ChoiCo current version:**

- Supported browsers: Chrome, Firefox, Safari 5+, Opera 12+, Internet Explorer 11+, iOS 12+, Safari 12+

**3.2.3 Authorable Learning Analytics**

The Exten.(D.T.)2 platform will be equipped with a specialised authoring tool (AuthELO) to enable the development of automated support for highly interactive web-based user interfaces. This operation enhances tools like Malt2 and Choico with the ability to provide real-time automated and adaptive support to students working on the learning activities. The purpose of using the tool is to increase the level of engagement and the overall user experience while reducing the cognitive load of learning designers producing the support. For that part this tool will be used for the production of authorable feedback.

Part of the authoring process involves the acquisition of data being generated as the students work on the learning activities. This data has the form of indicators that depict user interactions with the learning tools. AuthELO is equipped with a special editor that dynamically adapts to the nature of the learning tool it is attached to and gives the details of the data it generates. The same editor can be used by the designer to set up rules about what data to log and the level of granularity of that data. This is the data that will eventually be stored in the database and ultimately be used for analysis and visualisations. This is the authorable learning analytics facet of AuthELO.

**3.2.4 Authorable LA Visualization Dashboard**

The initial requirements for LA visualization dashboard were gathered from Grant Agreement document and an input from WP2 literature review outcome and presented in Table 4.

**Table 4. General Technical Requirements for ALA Dashboard**

N	Requirement Description	End Users
R3.1	Configuration: the dashboard should allow users to configure it by selecting the most suitable visualization components/widgets for learning activity (or pedagogy approach) and visualization representations for user’s purpose (e.g., tracking student’s individual progress, evaluation learning activity).	Teachers, Students, Researchers
R3.2	The users should be able to save, load the configured visualization dashboards.	Teachers, Researchers, Students
R3.3	Annotations: the users should be able to make annotations on visualization components/widgets in order to add extra information to complement the visualization.	Teachers, Researchers
R3.4	Annotations: the teachers should be able to make annotations on visualization components/widgets in order to add extra information to complement the visualization and notify this to a particular student or group of students. Students may be able to see the teacher’s annotations to better understand the visualization.	Teachers, Students
R3.5	Notes: the users should be able to make notes in dashboard for yourself use/purpose.	Teachers, Researchers
R3.6	<p>The dashboard for teacher should:</p> <ul style="list-style-type: none"> <li>(a) provide visualization on different levels: (1) student’s individual level; (2) group level (3-5 students in group); (3) class level (all student in a class 20-25 students),</li> <li>(b) provide visualization which allows to compare the learning activity outcomes between students or groups,</li> <li>(c) provide visualization components/widgets for: <ul style="list-style-type: none"> <li>(i) student’s learning outcome or performance</li> <li>(ii) student’s individual progress</li> <li>(iii) student’s engagement</li> <li>(iv) student’s developed 21st century skills,</li> </ul> </li> <li>(d) provide explanatory visualizations rather/or just exploratory,</li> <li>(e) real-time visualization: during the DT project to monitor groups activities for teachers and highlight groups of students which needs extra help from teacher; the interface would also emphasize when students are doing well while completing certain tasks, and</li> <li>(f) post DT project activity visualization as summary.</li> </ul>	Teachers

R3.7	The dashboard for the student should provide: (a) a widget/component for visualizing student’s developed 21st century skills, and (b) a widget/component for visualizing student’s performance in all phases of DT process and learning activity.	Students
R3.8	The dashboard for researcher should provide: (a) a widget/component for assessment of DT project or learning activity, and (b) a widget/component for evaluation of DT project or learning activity.	Researchers
R3.9	Visualisations should be more explanatory rather/or then just exploratory.	Teachers, Researchers, Students
R3.10	The visualization should also emphasize when students are progressing well or not well.	Teachers
R3.11	The visualization should support some accessibility features of the dashboard to accommodate disadvantaged (disabled) users or groups.	Students

These requirements and features will be refined and updated after performing several workshops with teachers using participatory design method.

### 3.2.5 nQuire Requirements

The extension of nQuire for Exten.(D.T.)2 will include the design of a version of nQuire for use in formal education. It will be a password protected (safe) environment that will be accessible by only students and teachers. It will enable participation by any student no matter their age (currently, nQuire is used by those 16+ years old). This means that students will be able to create their projects and share them with their peers as a way to collect data for their design thinking projects. The teacher will be able to view the pilot studies of students and provide them with feedback before s/he approves their study to be shared with other students/schools of the project. In addition, nQuire will enable sharing of outputs from other project technologies such as ChoiCo games for students to share, compare and discuss with others, while it will inform the design of the AI-authorable dashboard by capturing and providing data indicators from nQuire such as students’ participation in studies, number of studies created and piloted, number of comments left.



### 3.3 Exten.(D.T.)2 Design

The Exten.(D.T.)2 system is designed to accommodate different deployment scenarios. Although the system is intended to be used as a service, we decided to follow an architectural design that is flexible and generic enough to accommodate any possible deployment scenario (cloud/on premise).

A Service Oriented Architecture (SOA) design was selected as a loose-coupling integration and interoperability paradigm. All the software components in the system (own and third-party) are being considered as self-contained black boxes that encapsulate their data and business logic and communicate with each other through standardised APIs and communication protocols. This offers a sense of a federation of independent applications that interoperate in a disciplined and organised manner. In such a system the chances of something breaking because of changes implemented in some components are very slim. This provides adequate flexibility along with stability and robustness. The REST architectural pattern will be used to develop the component services. REST is protocol neutral (can be coupled with HTTP or any other protocol that supports URIs) and it allows the components to extend/modify themselves without breaking the services.

#### 3.3.1 Architecture Overview

The core of the system is the application server. This part will be developed in Node-RED<sup>5</sup>. Node-RED is an open source flow-based low-code development platform designed for web based Internet Of Things (IOT) applications. It offers a visual programming environment and a plethora of operators that can significantly simplify and speed up the development of any type of component. It can be used both as an IDE and as a deployment server. The development environment is accessible through a web interface (see Figure 2). It is a mature platform that has been heavily used in the industry for many years and is actively being supported by a large community of IT professionals.

The various series of services are categorised in flows. There will be a proxy component for every own or third party service in the application server. In this layer we can perform various operations on the data like aggregation, analysis, transformation, semantic enhancement etc. Handling all of this at the server side eliminates CORS-related problems too. These services are then exposed through the API gateway which will also be implemented in Node-RED. The idea behind this is to homogenise all the services and offer a common api through Exten.(D.T.)2. Even components that can have direct access to the actual services will always go through the api gateway.

---

<sup>5</sup> <https://nodered.org/>

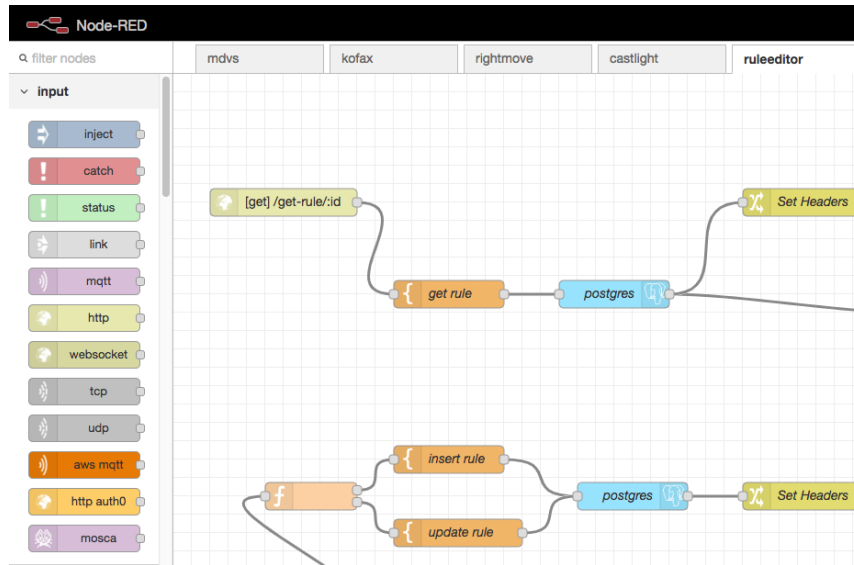


Figure 2: Node-RED interface for development

The purpose of the application server is to encapsulate own, third party and data services and homogenise them. These services are then complemented by components that provide Exten.(D.T.)2 “business logic” and can orchestrate the workflow.

The Exten.(D.T.)2 UI will be implemented as a front-end HTML5/Bootstrap-Vue application without any dependencies on back-end server-based technologies and will be deployed on the NRG web server (integral part of Node-RED).

The back-end data storage will be implemented with the object-relational DBMS PostgreSQL. PostgreSQL is free, open-source, feature-rich, ACID-compliant, transactional DBMS that can handle large workloads and multiple concurrent users. It is available on all major platforms, very extensible and scalable.

In general, the intention is to transfer as much of the processing as possible to the DBMS so that we don’t have too many roundtrips between the application server and the back end. Stored functions implemented in PL/pgSQL will be used for this purpose. The following diagram illustrates the architecture at a high level.

### 3.3.2 System Components

As depicted in the previous diagram the system we envisage to develop will comprise the following components (see Figure 3):

- Web-based GUI intended to be used for administration, development of learning / DT tasks and overview of learner usage. The technologies that will be used for that part will not have any dependencies on external engines or runtime systems. This is to ensure compliance / compatibility with future version of web browsers and the functionalities they inherently support.

- Tools to support learning through DT. These are the specialised tools Malt2, ChoiCo and Sorbet. In this category there is also a specialised tool design to enhance other tools with automated support (AuthELO).

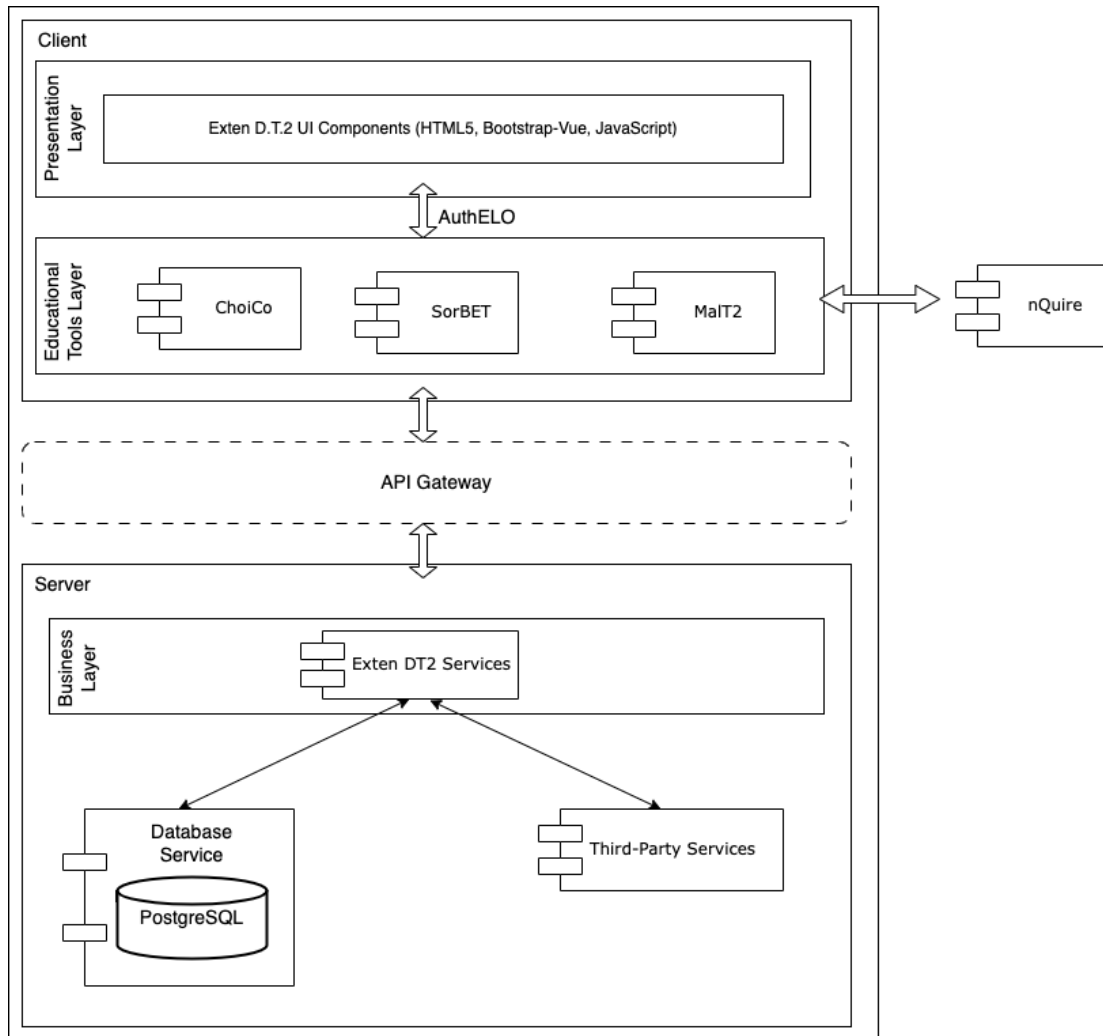


Figure 3: The platform architecture at a high level

- API Gateway. This is a common unified set of managed APIs to enable the interoperating software components in the system to combine with each other following a SOA (Software Oriented Architecture).
- Exten.(D.T.)2 Services. These are software components that implement the “business” logic of the system. That is any process or decision not related to data operations with the back end DBMS.
- Data Services. These are software components responsible for the interfacing with the back end DBMS. All the data-related operations will have to go through those components.
- DMBS. This is the system managing the database for the entire platform. The DBMS that will be used is Postgresql. The database that will be implemented will follow the relational model.

- Third party services. These are software components that will interface and thus represent services external to the system that will be used for additional functionality where deemed necessary.
- nQuire. This is an external platform that will gradually become an integral part of the system. Initially this platform will operate in a loosely coupled manner with the main Exten.(D.T.)2 platform.

### 3.3.3 Data Model Design

The conceptual model of the Exten.(D.T.)2 system (shown on Figure 4) consists from 8 entities (tables):

- *Schools table* store information about the school identifier (generated by the system), name, and additional attributes for secure authorization to the system: token, password, and verified flag. The system will be used by many schools in different countries, and all data will be stored in one database. For research, we need to access only the specific school data.
- *Learners table* store information about the learners’ identifier (generated by the system), alias (e.g., real name or pseudo-name), and school identifier to which a learner belongs.
- *Tutors (or Teachers) table* store information about the tutor’s identifier, first name, last name, school identifier to which a tutor is belong, and other three attributes are used for secure connection to the system are token, password, and verified flag.
- *Groups table* store information about the learners combined into groups. It records the group identifier (generated by the system), group name, description, the timestamp when the group was created, and the tutor’s identifier who was tutoring this group.

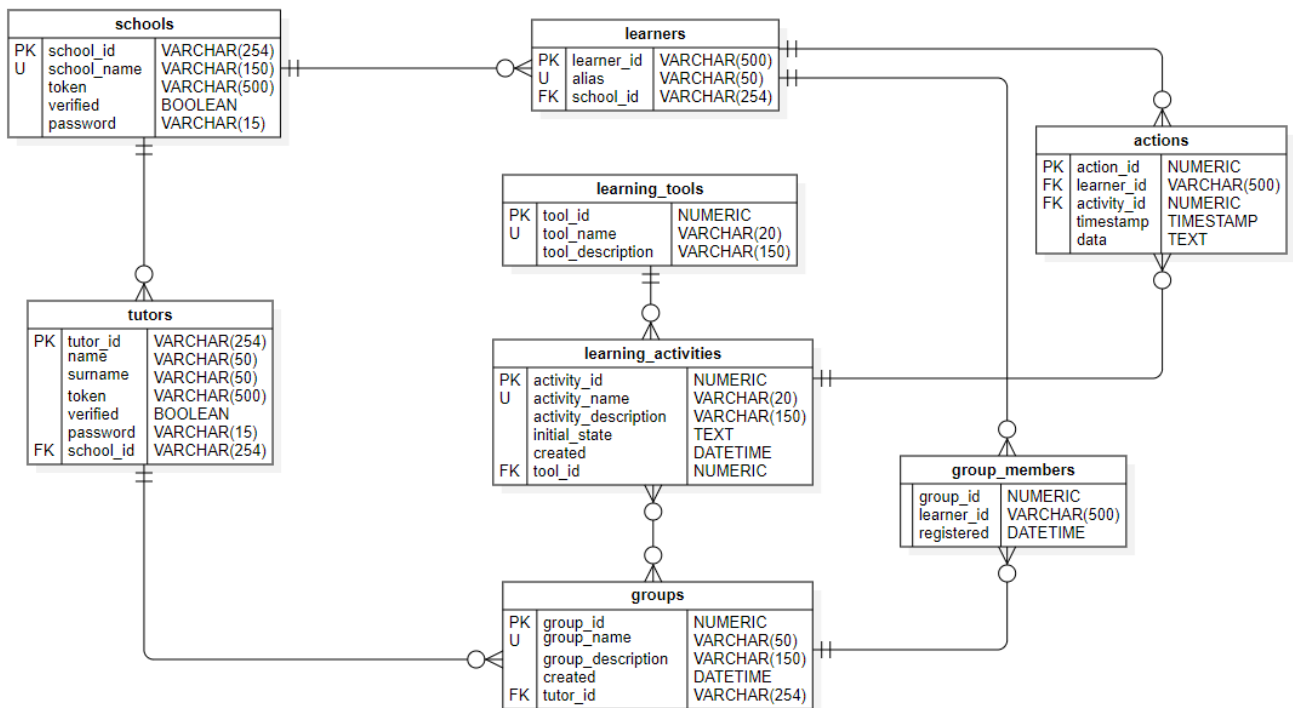


Figure 4: Relational data model diagram for Exten.(D.T.)2 platform

- *Group\_Members* table store information about the members (or learners) of the group, and records the learner’s identifier (*learner\_id*) and group identifier (*group\_d*) to which the learner belongs, and the timestamp when the learner is joined the group.
- *Learning\_activities* table store information about the learning activity designed by a teacher, and it records the activity identifier (generated by the system), activity name, description, timestamp when the activity is created, the learning tool identifier (*tool\_id*), and initial state of the activity is recorded as text.
- *Learning\_tools* table store information about the learning tool identifier (*tool\_id*, generated by the system), name (*tool\_name*, e.g., ChoiCo), and tool description.
- *Actions* table stores information for LA and it records every action made by learner in learning activity. It records the action identifier (generated by the system), learner’s identifier (*learner\_id*), activity identifier (*activity\_id*), the timestamp when the action occurred, and data is JSON object (see examples in Section 3.1.4.1, 3.1.4.2, and 3.1.4.3).

### 3.3.3.1 Data Model for Educational Tools

The extended version of the three learning tools (MaLT2, ChoiCo, SorBET) will track user actions through a list of pre-defined events (e.g. the user moves a slider to manipulate the 3D model in MaLT2) and in the case of an event, will create and send ‘logging messages’ with information about the context of these actions (e.g. in case of slider move event, the message will contain information about the slider start and end values, the sliding duration, the code of the 3D model). As users interact with the learning tool, the data that gets generated will be sent directly to the analytics servers in the form of xAPI messages. xAPI is powerful because it is simple, semantically rich (expressive) and easily extendable (versatile). The communication between the different system learning tools and the platform, relies on a special programming layer that ensures the interoperability of current and future application integration.

#### Events and logging messages

An ‘event’ can be: a) a user action performed on a certain element of the learning tool. For instance, the user moved the periscopic camera by clicking the arrows on MaLT2 scene. b) a condition that was reached after certain user actions. For instance, the user has clicked 30% of the available map pinpoints in a ChoiCo game. During user engagement with an activity a number of events can emerged.

For each tool, we will decide on a list of events that will be tracked, based on their added value for providing the end users valuable information on student learning processes with the tool. A first list of the events for the three existing tools is presented in the Appendix A.

When a loggable event takes place during user interaction, the learning tool publishes the data in the form of a JSON message. The message holds information about a specific user action in the tool during activity. The messages generated by all learning tools follow a common structure, namely:

- Example: { ‘id’: ‘’, ‘type’: ‘’, ‘event’: ‘’, ‘state’: {}, ‘timestamp’: real\_time, }, where
- ‘id’: The event id (integer)
- ‘timestamp’: The real time when the event occurred (string)

- **‘type’**: The learning tool’s element that user action was performed on. E.g. in case of MaLT2 this can have the following values *camera*, *code*, *slider* (string)
- **‘event’**: The name of the event that is related to the element (e.g. ‘changed’, ‘execution’). (string)
- **‘state’**: A JSON object with capturing information relevant to the user’s action. e.g. for the camera event logging message state would be `{view_type: "3dview", 'track_count': "3", event_count="25", event_last_count="3"}`

### 3.3.3.2 Sensitive Data

The data stored for each account type follows (see 3.4.3 for more details):

- Educational institutions (i.e schools)
  - Email, password (generated by the platform), name
- Designers / tutors
  - Email, password (generated by the platform), name
  - Learning activities
- Learners
  - Id (generated by the platform), real name or alias
  - Interaction indicators with learning tools
- System administrators / developers
  - No data

The student ID and group ID is generated by the Exten.(D.T.)2 system and the teacher will document which student belong to which ID in his/her own document (paper based or stored on her/his own computer).

According to GDPR regulations (Art. 9 GDPR) there is a clear distinction between personal and sensitive data. Personal data is any piece of information that can be used to identify someone. In that respect the data given above that could be thought of as personal are **emails** and **names**. Sensitive data is a special category of data. According to GDPR any piece of data that may reveal a subject’s information can be considered sensitive. The list below shows some examples:

- Racial or ethnic origin
- Political beliefs
- Religious beliefs
- Genetic or biometric data
- Mental health or sexual health
- Sexual orientation
- Trade union membership

According to this definition there is no sensitive data stored in the system for any of the participating entities.

### 3.3.4 Interoperability of Learning Tools

Learning Tools Interoperability (LTI) is an education technology specification developed by the IMS Global Learning Consortium (1EdTech Consortium) for e-Learning software products to communicate with external systems. The Exten.(D.T.)2 platform uses OAuth2 to login to the system once and access different learning tools such as SorBET, ChoiCo, MaLT2, without a need to login to each learning tool. All of the integrated educational system follow the same standard for data communication with Exten.(D.T)2 ALA componet. Data send from education tools follows the same structure, for example:

```
Logging_message = {
    'id': this.editdbID,
    'type': " database",
    'event': "modify_category",
    'state': {'info': {entryscounter: 6}, 'name': {old: 'Verb' , new: 'Noun',
        event_count: 6},
    'timestamp': 166773646
}
```

### 3.3.5 Deployment Strategy

As mentioned in 3.3 the platform is designed to be flexible and generic enough to accommodate any possible deployment scenario (cloud/on premise). It is difficult to anticipate how a project will evolve over time. That gives us the freedom to accommodate different needs regardless of the orientation the project will take.

The initial deployment will be made on Amazon Web Services<sup>6</sup>. The decision of a cloud based deployment scenario is based on the following:

#### Low cost of ownership

Cost of ownership includes both the cost purchasing something and the cost of maintaining over its useful life. In a cloud based deployment there is no need to purchase hardware or software. You pay only for the service you use. In addition to that the cost of maintenance is negligible. In most of the cases there is the option to deploy software as a fully managed service. That means that the service provider undertakes the maintenance.

#### Scalability

New resources can be allocated in very little time if it is deemed necessary. Most cloud-based providers promise a near-unlimited scalability making available on-demand resources to meet project needs at any level. The process to employ new resources may also be an automated task. Of course, it is equally easy to release allocated resources that are being underused to minimise expenses.

#### Development & Deployment

A worldwide network of resources that get replicated automatically and become easily accessible from anywhere gives great flexibility when people from different places need to

---

<sup>6</sup> <https://aws.amazon.com/>

develop software collaboratively. Easy access to the same resources increases maintainability, testability and deployment efficiency as it makes the outcomes of the project directly available for user testing.

The decision to use AWS instead of other cloud providers is based on the following:

- Amazon is by far the biggest cloud based service provider in the global market. It holds about 34% of the total market. Microsoft Azure comes next holding a 21% share and GCP Google Cloud Platform holds about 10%. Judging solely by numbers Amazon seems to be the safe way to go.
- UCL has an ongoing collaboration - partnership with Amazon and after negotiations we secured an initial funding from Amazon that should be enough to run the infrastructure for the first year.
- Members of the project teams have previous experience with Amazon Services and that makes the process of setting up the infrastructure and using this particular environment an even more cost-effective option.

### 3.3.6 Server Hardware Specification

It is expected that there will be a Virtual Private Cloud (VPC) with a single subnet that initially will be isolated from external networks for inbound connections. Outbound connections should be permissible so that we can take advantage of third-party services and have some administrative flexibility if new software is required. Initially access should be given through VPN and once the solution is internally tested and ready to be released to schools, we can allow inbound access to selected networks. This solution is adequately secure as well as easy to implement and maintain.

#### VM spec for the application server:

vCPU: 8

Primary Storage (RAM): 16 (preferably 32) GB

Secondary Storage (Disk1 - OS): 512 GB SSD R≈560MB/s, W≈520MB/s

Secondary Storage (Disk2 - Data): 2 TB M.2 NVMe 1.4 R≈7000MB/s, W≈5500MB/s

OS: Windows server 2019 or 2022

#### VM spec for the DBMS:

If this is provisioned as a service the underlying instance should be equivalent to t2.large AWS EC2 spec or higher.

More detailed hardware specification can be found in Appendix B Server Configuration Specification.



## 3.4 Policy, Ethics, and Security

### 3.4.1 Policy

The System and Network Administrator and Head of IT assure the following:

1. Clear and defined joiner, mover and leaver processes, and controls for them, will be in place:
  - Give access to AWS Console (test, staging, production, development) only to persons who are involved in the development of Exten.(D.T.)<sup>2</sup> system.
  - The access to AWS Console will be performed using secure VPN connection.
  - Remove access to AWS Console (test, staging, production, development) when it is not needed anymore.
2. All access to data will be based on the ‘least privilege’ principle. This means that users and processes are only able to access information that they need to have access to.
3. Multi-Factor Authentication (MFA) must be applied as a default to cloud services or other services made available over the Internet. MFA must be applied to high privileged accounts used in critical infrastructures.
4. Internal systems, applications and databases must apply strong password in line with best practice.
5. Vulnerability scan assessments must run across all internal and external systems (or clusters of these based on similarity) on a bi-monthly basis, or with every major change.
6. All data (including but not limited to authentication information such as passwords or tokens) must be encrypted in transit and at rest.
7. Only secure encryption measures that are vendor supported shall be used. Protocols, key strengths etc. must be documented.

### 3.4.2 Ethics

Although data driven practices in educational settings have been increasingly adopted, recent research has mainly been conducted in higher education settings (Slade & Tate., 2019). Ethical aspects related to LA has recently started to gain attention and ethical and privacy concerns as well as stakeholders data literacy and feedback literacy are some aspects that require attention (Viber et al., 2021; Ifenthaler., 2020). This project seeks to involve stakeholders in the design of the Learning Analytics from an early stage of the process in order for stakeholders’ values and ethical considerations to be reflected in the design. Value Sensitive Design (VSD) is a “theoretically grounded approach to the design of technology that accounts for human values in a principled and comprehensive manner throughout the design process” (Friedman at al., 2013).

As such this project will elicit stakeholder values and ethical reflections following the VSD tripartite methodology approach including conceptual, empirical and technical investigations through involving stakeholders in activities such as workshops, interviews and surveys. In the conceptual investigations both direct and indirect stakeholders will be identified and appropriate methods for eliciting these stakeholders’ values will take place. Previously identified values such as transparency, data ownership and control, accessibility of data, validity and reliability of data, institutional responsibility and obligation to act, consent and trust will also be evaluated according to compliance with the global guidelines of ethics in LA

(Slade & Tait., 2019). During workshops stakeholders will also be engaged in empirical investigations to examine and evaluate their experiences of the technology concerning the values they consider important and the technical investigations allow stakeholders to reflect on how values they deem important are hindered or supported by the design. This process is iterative in nature and will as such engage stakeholders throughout the design and development of the LA platform.

### 3.4.3 Security

#### Authentication and Authorisation

There are four entities associated with the system:

- Educational institutions (i.e., schools)
- Designers / tutors
- Learners
- System administrators / developers

Authentication of accounts and authorisation to access the system services will be managed internally by the Exten.(D.T.)2 platform. We chose not to use an external cloud based SSO service provider in order to avoid external dependencies. We also wanted to allow users the freedom to choose any email address they want to uniquely identify their accounts. The method used for authentication and authorisation is based on JSON Web Tokens (JWT). It is a simple, efficient and robust mechanism that offers adequate security for the purposes of the project.

The data stored for each account type follows:

- Educational institutions (i.e., schools)
  - Email, password (generated by the platform), name
- Designers / tutors
  - Email, password (generated by the platform), name
- Learners
  - Id (generated by the platform), real name or alias
- System administrators / developers
  - No data

The process of using the system for each account type follows (high level overview):

- **School administrators** register their schools with the platform. After confirming the school email they receive their password. Then, they log in the platform and they register tutors and learners. School admins are allowed to see tutor names and emails but not their passwords. They are also allowed to see the student names and their unique identifiers. No IP information and timestamps are stored for the sessions.
- **Tutors** receive an invitation to confirm their email and once that is done, they receive their passwords. Then, they log in and start developing learning materials. Once,

learning materials are available, they can create groups of students and assign learning tasks to them. Tutors are not allowed to see information about other tutors teaching in the same school. They are allowed to see student names or aliases though so that they can make decisions about how to allocate them to groups. No IP information and timestamps are stored for the sessions.

- **Learners** are given secret unique passcodes that can be used to get access to their activities. Learners will be able to access their learning tasks using their unique passcodes combined with the class ID they belong to. This is going to allow easy access as well as anonymisation of data and security. There is no personal or sensitive information about them being exchanged throughout the sessions. The user interface given to the students does not display any personal information about them or other members of the groups they participate. As learners are doing the given activities anonymised and timestamped interaction indicators are generated and stored in the database. No IP information is stored for the sessions.
- **System administrators and developers** have direct and unlimited access to user data. Of course, sensitive information stored in the database will be encrypted and therefore they will not be able to identify individuals, associate them with usage data or reuse their passwords.

The backend infrastructure of the solution will be built entirely from the following services:

- **API Gateway**  
Used to create and publish API's used for all internal communications within the solution. It will contain a function acting as a Custom Authorizer, ensuring that all incoming connections from front-end instances have been authenticated.
- **Services**  
Services will be implemented as components that run small code modules that perform the various activities required of the solution. Services are stateless, and do not preserve data between instances when they are called.
- **DBMS**  
This is a Relational Database Service, in which the user data is stored.

All components will be contained within a Virtual Private Cloud (VPC).

### Protective Monitoring

Depending on the platform used the system may be integrated with a Security Information and Event Management (SIEM) tooling.

### Operating System Hardening

The solution will comprise virtual server instances in the context of Amazon Web Services. Prefabricated images of Operating systems with embedded security will be employed to ensure a certain degree of protection from day 0. These images are made by specialised IT Support teams at UCL. There will of course be ongoing updates as the project progresses to ensure that the required degree of security is maintained.

### **Patching**

As the solution will comprise virtual server instances continuous monitoring of security updates and patching will be necessary.

### **Protection of Data-In-Transit**

All traffic flows from the front-end to the API Gateway, and all internal API calls within the solution will be encrypted using HTTPS.

### **Protection of Data-At-Rest**

The only data stored in the system is within the RDS database. This database will be provisioned encryption enabled. The keys will be managed using the Node-RED Key Management Service. Data stored at rest on the volume, database snapshots, automated backups, and read replicas will all be encrypted. The username and password required to make authorised API calls by the services will be stored as encrypted Node-RED environment variables.

## **4. CONCLUSIONS**

This deliverable describes the extension of education tools (SorBET, ChoiCo, MaLT2), general architecture overview of Exten.(D.T.)2 system, user requirements for LA dashbard, and extended educational tools. The requirements provided in the document lead to further design work to determine the specifics for each of the educational tools within the Exten.(D.T.)2 system. The identified requirements will be used for starting the first development circle of Exten.(D.T.)2 system.

## References

- Alsoliman, B. S. H. (2022). Virtual robotics in education: The experience of eighth grade students in STEM. In *Frontiers in Education* (p. 490). Frontiers.
- Friedman, B., Kahn, P. H., Borning, A., & Huldtgren, A. (2013). Value sensitive design and information systems. *Early engagement and new technologies: Opening up the laboratory*, 55-95.
- González-García, S., Rodríguez-Arce, J., Loreto-Gómez, G., and Montaña-Serrano, V. M. (2020). Teaching forward kinematics in a robotics course using simulations: transfer to a real-world context using LEGO mindstorms™. *Int. J. Interact. Design Manufact.* 14, 773–787. doi: 10.1007/s12008-020-00670-z
- Ifenthaler, D., Gibson, D., Prasse, D., Shimada, A., & Yamada, M. (2021). Putting learning back into learning analytics: Actions for policy makers, researchers, and practitioners. *Educational Technology Research and Development*, 69, 2131-2150.
- Leinonen, T., Virnes, M., Hietala, I., & Brinck, J. (2020). 3D Printing in the Wild: Adopting Digital Fabrication in Elementary School Education. *International Journal of Art and Design Education*, 39(3), 600–615. Scopus. <https://doi.org/10.1111/jade.12310>
- Mistretta, S. (2022). “Virtual robotics in hybrid teaching and learning,” in *New Updates in E-Learning*, ed. E. Babulak (London: IntechOpen), doi: 10.5772/intechopen.102038
- Roumelioti, E., Pellegrino, M. A., Rizvi, M., D’Angelo, M., & Gennari, R. (2022). Smart-Thing Design by Children at a Distance: How to Engage Them and Make Them Learn. *International Journal of Child-Computer Interaction*, 33, 100482. <https://doi.org/10.1016/j.ijcci.2022.100482>
- Slade, S., & Tait, A. (2019). *Global guidelines: Ethics in learning analytics*.
- Viberg, O., & Grönlund, Å. (2021). Desperately seeking the impact of learning analytics in education at scale: Marrying data analysis with teaching and learning. In *Online learning analytics* (pp. 19-31). Auerbach Publications.
- Art. 9 GDPR. Processing of special categories of personal data. Web URL: <https://gdpr-info.eu/art-9-gdpr/> (Retrieved, February, 2023)
- IMS Global Learning Consortium has since become 1EdTech Consortium.

## Appendix A: Educational Tools Data Description

Data	Description	Data Format	Connection to Design Thinking Theory / Practice	Connection to 21st-century skills
<b>MaLT2 - Event-driven data</b>				
<p><b>Event:</b> New Logo Procedure Defined</p> <p><b>Data:</b></p> <ul style="list-style-type: none"> <li>-Procedure code</li> <li>-Procedure name &amp; parameters - Start &amp; end position of the text in the editor</li> <li>-Timestamp</li> <li>-Event count since the activity started</li> </ul>	<p>When a Logo Procedure (to...end) is defined or redefined by the student in the Logo editor.</p>	<pre> Json object {   'id':   this.editorId ,   'type': "editor",   'state' :   {"definition" : {}},   "position":{     start:{row: 1, col: 2},     end: {row: 3, col: 10}},   "value":{proc_name: "ladder",     "parameter_names": ['x','y','z']},   "event_count": 1},   "event":   "proc_definition",   "timestamp" : UTC time } </pre>	-Prototype, - Develop solution	-Computational Thinking-&gtAbstraction, Automation, Modeling
<p><b>Event:</b> Logo Procedure Executed</p> <p><b>Data:</b></p> <ul style="list-style-type: none"> <li>-Procedure code</li> <li>-Procedure name &amp; parameters names</li> <li>-Parameters execution values</li> <li>-Start &amp; end position of the procedure in the</li> </ul>	<p>Triggered when a Logo Procedure is executed in the editor e.g. when student executes <i>ladder 100</i> (not as sub-procedure)</p>	<pre> Json object {   'id':   this.editorId ,   'type': "editor",   'state' :   {"definition" : {}},   position:{start:{row: codeRange.start.line, col: codeRange.start.ch},     end: {row: codeRange.end.line, col: codeRange.end } </pre>	-Testing	-Computational Thinking -> Automation, Abstraction, Debugging

<p>editor</p> <p>-Event count since the activity started</p> <p>-Timestamp</p>		<pre>d.ch}}, value:{proc_ name: "ladder", parameter_na mes: ['x', 'y', 'z'] ,parameter_v alues:[10, 100, 30] }, event_count: 2}, "event": "proc_definition", "timestamp" : UTC time }</pre>		
<p><b>Event:</b> Procedures Count</p> <p><b>Data:</b> -The number of defined procedures</p> <p>-The names of defined procedures</p> <p>-Start &amp; end position of the procedures code in the editor</p> <p>-Event count since the activity started</p> <p>-Timestamp</p>	<p>It is triggered when the student selects defines a number of procedures in the logo editor</p>	<p>Json object</p> <pre>{ 'id:this.editorId ', 'type': "editor", 'state' : {"definition " : {"number": 3, "procedures" : ['ladder', 'test', 'square']}, "position":{ start:{row: 2, col: 20}, end: {row: 15, col:5}}, "event_count ": 1}, "event": "procedure_count_ch ecked", "timestamp" : UTC time }</pre>		
<p><b>Event:</b> Trace Clicked</p> <p><b>Data:</b> -Name of the related Logo procedure -Names of procedure's parameters</p>	<p>Triggered when the student clicks on the graphic created by the execution of a Logo procedure e.g. on a square</p>	<p>Json object</p> <pre>{ 'id': object_id , 'type': "trace", 'state' : {definition: {proc_name: "ladder", parameter_na mes: ['x','y','z' ]}, parameter_va lues: [10,</pre>	<p>-Exploring - Testing -Understanding solution</p>	<p>-Computational Thinking -&gt; Debugging -Spatial Thinking ++</p>

<ul style="list-style-type: none"> <li>- Execution values of procedure's parameters</li> <li>- Event count since the activity started</li> <li>- Mouse position on the 3D scene</li> <li>-Timestamp</li> </ul>		<pre>100, 30]], position: {x:450, y: 300}, event_count: 5, event_last_c ount: 1} ,  "event": "click", "timestamp" : UTC time }</pre>		
<p><b>Event:</b> Slider Use <b>Data:</b></p> <ul style="list-style-type: none"> <li>-Name of related Logo procedure</li> <li>-Slider value when user clicked the slider (e.g. 10)</li> <li>-Duration in msec that user helded the slider</li> <li>-Slider's parameter name (e.g. 'a')</li> <li>-Timestamp</li> </ul>	<p>Triggered when the user student a slider of any Logo procedure</p>	<pre>Json object { 'Id': sliderID , 'type': "trace", 'state' : { "value": 13, "definition" :{proc_name: "ladder" , min: 1, max:100, step:1, initialvalue :200, experduratio n:13000, variable:par ameter_name} , "event": "click", "timestamp" : UTC time }</pre>	<p>Exploring, Testing prototype, Presenting,</p>	<ul style="list-style-type: none"> <li>-Computational Thinking -&gt; Debugging, Pattern Recognition, Visual abstraction, Modelling analysis</li> <li>-Spatial Thinking</li> </ul>
<p><b>Event:</b> Camera Use <b>Data:</b></p> <ul style="list-style-type: none"> <li>-View Type: 2D or 3D</li> <li>-Number of models on the scene created by different Logo procedures</li> <li>-Times the event occurred since last Logo code execution</li> <li>- Times the event occurred since activity started</li> </ul>	<p>Triggered when the student rotates/zooms the camera in any way</p>	<pre>Json object { 'id': cameraID, 'type': "camera", 'state':{ 'view_type': "3dview", 'track_count ':3, 'event_count ':25, 'event_last_ count': 3}, 'event': "camera", 'timestamp' : UTC time }</pre>	<p>Exploring, Testing prototype, Understanding +++</p>	<ul style="list-style-type: none"> <li>-Computational Thinking -&gt; Debugging, Visual abstraction, Modelling analysis</li> <li>-Spatial Thinking</li> </ul>



The following events are generated only if the designer (e.g. teacher) has activated them in the MaLT2 analytics options

<p><b>Event:</b> Correct solution</p> <p><b>Data:</b></p> <ul style="list-style-type: none"> <li>- Solution type i.e. absolute or relative</li> <li>- the values of the parameters that the procedure was executed</li> <li>-times the event occurred since activity started, including both relative and absolute solutions</li> <li>-Timestamp</li> </ul>	<p>Triggered when a specific Logo procedure is executed by the student with the correct numbers as inputs (or relative correct). The procedure name, the correct inputs and the 'absolute' or 'relative' solution are defined by the activity designer (e.g. the teacher) in the MaLT2 analytics options.</p>	<pre> Json object {   'id':   this.editorId,   'type': "editor",     'state' :     {'solution_kind':     "absolute ",     'value' :     [10,     20], 'event_count': 1},    'event':   "correctParameters",   'timestamp' : UTC   time }         </pre>	<p>Develop solution, Modeling ++</p>	<p>Computational Thinking -&gt; Automation, Debugging, Model Design ++</p>
<p><b>Event:</b> Parameters Count</p> <p><b>Data:</b></p> <ul style="list-style-type: none"> <li>-The number of parameters in relation to min and max number. I.e. "no parameters" (procedure's parameters = 0), "out of limits" (procedure's parameters &lt; min or procedure's parameters &gt; max), "between limits" (min &lt; procedure's parameters &lt; max)</li> <li>-Timestamp.</li> </ul>	<p>Triggered when a specific procedure is defined or redefined by the student and informs the system about whether the number of parameters are in or out of specific min and max limits of expected parameters. The Logo procedure name, and the expected min and max no of parameters are defined by the activity designer (e.g. the teacher) in the MaLT2 analytics options.</p>	<pre> Json object { 'id':   this.editorId ,   'type':"editor",   'state' : {count:     'between limits'},   'event' :   "parameter_count",   'timestamp' :UTC   time}         </pre>	<p>Develop solution, prototype Building a model ++</p>	<p>Computational Thinking -&gt; Abstraction, Automation, Debugging, Model Design</p>

<p><b>Event:</b> Code Modified by Characters <b>Data:</b> -The number of typed characters so far -Timestamp</p>	<p>It is sent only once, when the student defines or executes any procedure and by this time he/she has typed in the editor equal or more characters than “min characters”.</p> <p>The min characters number is defined by the activity designer (e.g. the teacher) in the MaLT2 analytics options.</p>	<p>Json Object {   'id': this.editorId,   'type':" editor" ,   'state':   {'mod_char_count': 60},   'event':"code_min_changed",   'timestamp' : time };</p>	<p>Develop solution Prototype ++</p>	<p>Computational Thinking -&gt; Automation, Model Design</p>
<p><b>Event:</b> Keyword found <b>Data:</b> -Keyword(s) found -Times the specific keyword was detected in the Logo procedure -Times the event has been triggered from the beginning of the activity (for all the keywords) -Timestamp</p>	<p>Triggered when one of the keywords is detected in a specific Logo procedure. It is sent when this procedure is defined (or redefined) by the student.</p> <p>The keywords and the Logo procedure name are defined by the activity designer (e.g. the teacher) in the MaLT2 analytics options.</p>	<p>Json object { 'id': this.editorId ,   'type':"editor",   'state' : {     definition:{     },     position:{},     value:{keyword: 'sin',     occurrences: 5},     event_count : 10},   'event' :   "keyword_found",   'timestamp':UTC time}</p>	<p>Develop solution Prototype ++</p>	<p>Computational Thinking -&gt; Automation, Model Design</p>

ChoiCo - Event-driven Data				
Game Play - The following events are triggered while the student plays a ChoiCo game in play mode				
<p><b>Event:</b> Select Point</p> <p><b>Data:</b></p> <ul style="list-style-type: none"> <li>- Values of the game fields at the time of the event</li> <li>- Number of choices made so far</li> <li>-Point name</li> <li>-Map layer that the point appears</li> <li>- Times player has selected this point so far in the current game</li> <li>-Times the event occurred since activity started</li> <li>-Timestamp</li> </ul>	<p>Triggered when the player selects a point during the game (Clicks the button "select point")</p>	<p>Json object</p> <pre>{   'id': gameId,   'type':     "playmode",   'event":     "point_selection",   'state':     {'game_progress': {'fields':       {Money: 10,       Joy: 50,       Health: 30},       'score': 10}     'value': {name:       'Banana',       layer: 'Main',       times: 5},     event_count:       4},   'timestamp' :     1667736467 }</pre>	<p>Understand Empathize Explore Analyze ++</p>	<p>++</p>

<p><b>Event:</b> layer changed</p> <p><b>Data:</b></p> <ul style="list-style-type: none"> <li>-Values of the game fields at the time of the event</li> <li>-number of choices made by the player at the time of the event</li> <li>- layer name</li> <li>-Times the player has visited this layer in the current game (either by selection or automatically)</li> <li>-Times the event occurred since activity started</li> <li>-Timestamp</li> </ul>	<p>Triggered when a layer (map area) is changed either by the player's choice (chooses to visit this layer) or automatically by the game</p>	<pre> Json Object {   'id': gameId,   'type':   "playmode",   'event":   "change_layer",   'state':   {'game_progress': {'fields':   {Money: 10,   Joy: 50,   Health: 30},   'score': 10}   'value': {name:   'Kitchen',   times: 5},   event_count:   4},   'timestamp' :   1667736467 } </pre>	<p>Understand Empathize Explore Analyze ++</p>	<p>++</p>
<p><b>Event:</b> Game over</p> <p><b>Data:</b></p> <ul style="list-style-type: none"> <li>- Values of the game fields at the time of the event</li> <li>- Number of choices made by the player at the time of the event</li> <li>- Name of the field that caused the game to end</li> <li>- Times the event has occurred since activity started</li> <li>-Timestamp</li> </ul>	<p>Triggered when the game ends</p>	<pre> Json Object {   'id':gameID,   'type':   "playmode",   'event":   "game_over",   'state':   {'game_progress': {'fields':   {Money: 10,   Joy: 50,   Health: 30},   'score': 10}   'value':   {'Money'},   event_count:   4},   'timestamp' :   1667736467 } </pre>	<p>Not applicable</p>	<p>Not applicable</p>
<p>The following play mode events are generated only if the designer (e.g. teacher) has activated and defined them in the ChoiCo analytics options</p>				

<p><b>Event:</b> Player Performance <b>Data:</b> -Values of the game fields at the time of the event -Number of choices made by the player at the time of the event -Type of player's performance i.e. low, medium, high -Timestamp in UTC</p>	<p>Triggered when the player has selected a percentage of the available choices that is considered a milestone for his/her activity (e.g. 30%). There are 3 types of performance: low, medium, high. The milestone percentages are indicated by the activity designer (e.g. the teacher) in the ChoiCo analytics options.</p>	<pre>Json Object {   'id': gameId,   'type':     "playmode",   'event':     "player_performance",   'state':     { 'game_progress': { 'fields':       { Money: 10,         Joy: 50,         Health: 30},       'score': 10}       'value':         { 'medium' }},   'timestamp':     1667736467 }</pre>	<p>Understand the problem Empathize Decision making Sustainability/ System balance ++</p>	<p>System/model analysis Systems thinking Decision Making Critical Thinking ++</p>
<p><b>Event:</b> Player Activity <b>Data:</b> -Values of the game fields at the time of the event  -Number of points clicked by the player at the time of the event  -Type of player's activity i.e. low, medium, high  -Timestamp UTC</p>	<p>Triggered when the player has explored (clicked on) a certain number of choices related to the total number of available choices that is considered a milestone for his/her activity (e.g. 80 %). There are 3 types of activity: low, medium, high. The milestone percentages are indicated by the activity designer (e.g. the teacher) in the ChoiCo analytics options.</p>	<pre>Json Object {   'id': this.gameID,   'type':     "playmode",   'event':     "player_activity",   'state':     { 'game_progress': { 'fields':       { Money: 10,         Joy: 50,         Health: 30},       'score': 10}       'value':         { 'medium' }},   'timestamp':     1667736467 }</pre>	<p>Understand Empathize Explore options Analyze data ++</p>	<p>Model analysis Systems thinking Data comparison/analysis  ++</p>
<p>Game Design - The following events are triggered while the student designs/modifies a ChoiCo game in design mode</p>				
<p><b>Event:</b> Add new point</p>	<p>Triggered when the student adds a new</p>	<pre>Json Object {   'id': mapID,</pre>	<p>Prototype Develop</p>	<p>Computational thinking</p>

<p><b>Data:</b>          -Number of points on the game map          -Number of map layers          -Coordinates of the new point (x,y or lat, lng)          -id of the new point          -Times the event occurred since activity started          -Timestamp UTC</p>	<p>point on the map interface</p>	<pre> '`type': " map_editor", 'event": " "new_point",   'state':     {'map_state':       {points: 15,        layers:3},      'value':       {coordinates:        {134, 50}, id:        12},      'event_count':        4}, 'timestamp': 1667736467 }         </pre>	<p>solution          Define options          ++</p>	<p>++</p>
<p><b>Event:</b> Modify point  <b>Data:</b>          -Number of points on the game map          -Number of layers of the game map          -Coordinates of the point (x,y or lat, lng)          -Point name          -Point id          -Times the event occurred since activity started          -Timestamp UTC</p>	<p>Triggered when the student modifies a point of the game map. There are 3 types of the event, depending on the modding action: name, position, delete</p>	<pre> Json Object { 'id':mapID, 'type':" map_editor", 'event": " "modify_point",   'state': {'map state':     {points: 15,      layers:3},     'value':     {coordinates:      {134, 50},      name: 'Banana',      id: 12}, 'type'     : 'move',     'event_count':       4}, 'timestamp': 1667736467 }         </pre>	<p>Prototype          Explore options          Test solutions          Define options          ++</p>	<p>Computational Thinking          ++</p>
<p><b>Event:</b> modify graphics  <b>Data:</b>          -Number of points on the game map          -Number of layers of the game map          -Layer's name          - New background URI</p>	<p>Triggered when the student changes the background image of a map layer</p>	<pre> Json Object { 'id': this.mapID, 'type':" map_editor", 'event": " "modify_graphics",   'state':     {'map state':       {points: 15,        layers:3},      'value':       {name:        'Kitchen',        'imgURi': "" } }         </pre>	<p>Prototype          Develop solution          Define options          ++</p>	<p>Computational Thinking          Communication          ++</p>

<p>-Times the event occurred since activity started -Timestamp UTC</p>		<pre>, 'times': 3}, 'event_count' : 4}, 'timestamp': 1667736467 }</pre>		
<p><b>Event:</b> Modify database cell for a certain point <b>Data:</b> - Name of the field (column) -Point Id (row) -Old cell value -New cell value -Times the value of the specific cell has been changed -Times the event occurred since activity started -Timestamp UTC</p>	<p>Triggered when the student changes the database value of a certain point (row) for a certain field (column) in the database i.e. a database cell</p>	<pre>Json Object { 'id': this.databaseID, 'type': " database", 'event": "change_value", 'state': {'info': {name: 'Money', pointID: 14}, 'value': {old: 50, new: 60, times: 2}, event_count: 6}, 'timestamp': 1667736467 }</pre>	<p>Prototype Explore options Define options ++</p>	<p>Computational Thinking -Data manipulation -Data Handling ++</p>
<p><b>Event:</b> Add field <b>Data:</b> - Number of database fields (columns) - Number database of points (rows) -Times the event occurred since activity started -Timestamp UTC</p>	<p>Triggered when the student adds a new field (column) in the database</p>	<pre>Json Object { 'id': this.databaseID, 'type': " database", 'event": "add_field", 'state': {'info': {fields: 5, points: 20}, 'value': "", event_count: 5}, 'timestamp': 1667736467 }</pre>	<p>Prototype Develop solutions Define options ++</p>	<p>Computational Thinking -Data manipulation -Data Handling ++</p>
<p><b>Event:</b> Debug Code <b>Data:</b> -Number of changes the user has made in the 3 codespaces -Number of errors appearing</p>	<p>Triggered each time the student presses "Debug" button in the design mode</p>	<pre>Json Object { 'id': gameID 'type': "codespace", 'event": "debug", 'state': {'definition': {code: "", number: 14},</pre>	<p>Testing Developing Rapid prototyping ++</p>	<p>Computational Thinking -&gt; Debugging Algorithmic Thinking</p>

<p>- Times the event has occurred since activity started</p> <p>-Timestamp UTC</p>		<pre> 'value': 10, event_count: 1}, 'timestamp': 1667736467 }         </pre>		
<p><b>Event:</b> Design-Play Switch</p> <p><b>Data:</b></p> <p>-The mode the player changed to. E.g. if changed from play to design the name would be "design"</p> <p>- Times the event has occurred since activity started</p> <p>-Timestamp UTC</p>	<p>Triggered each time the user changes from play to design and vice versa</p>	<pre> Json Object { 'id': gameID 'type': " playmode ", 'event": " switch mode ", 'state': {'value': 'design', event_cou nt: 1}, 'timestamp': 1667736467 }         </pre>	<p>Testing solution</p> <p>++</p>	<p>Computational Thinking</p> <p>++</p>
<p>The following game design events are generated only if the designer (e.g. teacher) has activated and defined them in the ChoiCo analytics options</p>				
<p><b>Event:</b> Map design activity</p> <p><b>Data:</b></p> <p>-Number of points on game map</p> <p>-Number of map layers</p> <p>-Type of user activity i.e. low, medium, high</p> <p>-Timestamp UTC</p>	<p>Triggered when the student has made a number of changes to the available game choices (add new points, delete points, move points) that is considered a milestone for his/her design activity (e.g. 50 changes). There are 3 types of activity: low, medium, high. The 3 thresholds are indicated by the activity designer (e.g. Teacher) in the ChoiCo analytics options</p>	<pre> Json Object { 'id': this.mapID, 'type': " map_editor", 'event": " design_activity", 'state': {'map state': {points: 15, layers:3}, 'value': "low"}, 'timestamp': 1667736467 }         </pre>	<p>Developing solution</p> <p>Prototyping</p> <p>Modelling</p> <p>++</p>	<p>++</p>



<p><b>Event:</b> Database design activity  <b>Data:</b>          - Number of database fields (columns)          -Number database of points (rows)          -Type of user activity i.e. low, medium, high          -Timestamp UTC</p>	<p>Triggered when student has made a number of changes to the database values, i.e. cells, that is considered a milestone for his/her design activity (e.g. 50 changes). There are 3 types of activity: low, medium, high. The 3 thresholds are indicated by the activity designer (e.g. Teacher) in the ChoiCo analytics options</p>	<pre>Json Activity {   'id':   this.databaseID,   'type': " database",   'event':   "design_values_act   ivity",   'state':   { 'info':   { fields: 5,   points: 20},   'value':   "Medium"},   'timestamp':   1667736467 }</pre>	<p>Developing solution Prototyping Modelling ++</p>	<p>Computational Thinking -&gt; Data editing/analysis Critical Thinking Model Analysis</p>
<p><b>Event:</b> Initial Settings activity  <b>Data:</b>          - Code in the "initial settings" codespace          -Number of initializing commands in the code (e.g. set x to 50)          -No of changes the student has made so far in this code          - Times the event has been triggered since activity started          -Timestamp UTC</p>	<p>Triggered each time the student makes a number of changes to the "initial settings" code, that is considered a milestone for his/her design activity (e.g. 5 changes). The event is triggered every X changes. The milestone number of changes is indicated by the activity designer (e.g. teacher) in the ChoiCo analytics options.</p>	<pre>Json Object {   'id':1,//1st codespace   'type':   "codespace",   'event':   "activity",   'state':   { 'definition':   { 'code':   "sometext",   'number': 4},   'value': 5,   'event_count':   1},   'timestamp':   1667736467 }</pre>	<p>Developing solution Prototyping ++</p>	<p>Computational Thinking -&gt; Data editing/analysis Critical Thinking Model Analysis</p>
<p><b>Event:</b> Gameplay rules activity  <b>Data:</b>          - Code in the "gameplay rules" codespace          -Number of</p>	<p>Triggered each time the student makes a number of changes to the "gameplay rules" code, that is considered a</p>	<pre>Json Object {   'id':2,//2nd codespace   'type':   "codespace",   'event':   "activity",   'state':   { 'definition':</pre>	<p>Developing solution Prototyping ++</p>	<p>Computational Thinking -&gt; Algorithm creation, Conditionals System design ++</p>

<p>active gameplay rules (i.e. conditional structures) -No of changes the student has made so far in this code - Times the event has been triggered since activity started -Timestamp UTC</p>	<p>milestone for his/her design activity (e.g. 5 changes). The event is triggered every X changes. The milestone number of changes is indicated by the activity designer (e.g. teacher) in the ChoiCo analytics options.</p>	<pre>{'code': "some code in text", 'number': 6}, 'value': 10, 'event_count': 2}, 'timestamp': 1667736467 }</pre>		
<p><b>Event:</b> End rules activity <b>Data:</b> - Code in the "End rules" codespace -Number of active end rules in the code (i.e. conditional structures with game over command) -No of changes the student has made so far in this code - Times the event has been triggered since activity started -Timestamp UTC</p>	<p>Triggered each time the student makes a number of changes to the "end rules" code, that is considered a milestone for his/her design activity (e.g. 5 changes). The event is triggered every X changes. The milestone number of changes is indicated by the activity designer (e.g. teacher) in the ChoiCo analytics options.</p>	<pre>Json Object { 'id':3, //3rd codespace 'type': "codespace", 'event": "activity", 'state': { 'definition': { 'code': "some code in text", 'number': 6}, 'value': 10, 'event_count': 2}, 'timestamp': 1667736467 }</pre>	<p>Developing solution Prototyping  ++</p>	<p>Computational Thinking -&gt; Algorithm creation, Conditionals System design ++</p>

SorBET

Design Game Mode

The following events are triggered while students use SORBET in Design Mode (log message: "Load Design Mode")

\*entriescounter = the number of the column that corresponds to the specific category

\*idcounter = the number of the row that corresponds to the specific object

<p><u>Event:</u> Modify a Category <u>Data:</u> -Name of the Category (old) -Name of the Category (new) -Timestamp</p>	<p><u>Triggered:</u> When the designer modifies an already existing Category (column)</p>	<p>Json Object {   'id': this.editdbID,   'type': " database",   'event': "modify_category",   'state':     {'info':       {entriescounter : 6}, 'name':       {old: 'Verb' ,       new: 'Noun',       event_count: 6},   'timestamp': 166773646</p>	<p>(re)Define, Explore, Define Options, System Balance, Improve</p>	<p>Computational Thinking Critical Thinking Self-direction Modelling Debugging Decision Making</p>
<p><u>Event:</u> Modify an Object <u>Data:</u> -Name of the Object (old) -Name of the Object (new) -Type of the Object (old) -Type of the Object (new) -Timestamp</p>	<p><u>Triggered:</u> When the designer modifies an already existing Object (row)</p>	<p>Json Object {   'id': this.editdbID,   'type': " database",   'event': "modify_object",   'state':     {'info':       {idcounter: 12, type:       'text', times: 5}, 'name':       {old: 'running' ,       new: 'swimming',       times: 2}   'type': {old: 'text' , new:   'text', times: 2},   event_count: 6},   'timestamp': 166773646</p>	<p>(re)Define, Explore, Define Options, System Balance, Improve</p>	<p>Computational Thinking Critical Thinking Self-direction Modelling Debugging Decision Making</p>

<p><u>Event:</u> Add new Category <u>Data:</u> -Number of existing categories (columns of the Array) -Name of new category -Timestamp</p>	<p><u>Triggered:</u> when the designer adds a new category in the Edit Game Database.</p>	<pre>Json Object {   'id':   this.editdbID,   'type': " database",   'event":   "add_category",   'state':   {'info':   {entryscounter   : '7', name:   'Adjective',   event_count:   6},   'timestamp':   166773646</pre>	<p>(re)Define, Explore, Define Options, System Balance, Improve</p>	<p>Critical Thinking Self-direction Modelling Debugging Decision Making</p>
<p><u>Event:</u> Add new Object <u>Data:</u> -Number of existing objects (rows of the Array) -Name of new object -Type of new object -Timestamp</p>	<p><u>Triggered:</u> When the designer adds a new object in the Edit Game Database.</p>	<pre>Json Object {   'id':   this.editdbID,   'type': " database",   'event":   "add_object",   'state':   {'info':   {idcounter:   '13', name:   'swimming',   type: 'text',   times: 6},   event_count:   6},   'timestamp':   166773646</pre>	<p>(re)Define, Explore, Define Options, System Balance, Improve</p>	<p>Critical Thinking Self-direction Modelling Debugging Decision Making</p>
<p><u>Event:</u> Delete Category <u>Data:</u> -Number of existing categories (columns of the Array) -Name of new category -Timestamp</p>	<p><u>Triggered:</u> When the designer deletes a specific category in the Edit Game Database.</p>	<pre>Json Object {   'id':   this.editdbID,   'type': " database",   'event":   "delete_category",   'state':   {'info':   {entryscounter   : '13', name:   'Adjective'},   event_count:   6},   'timestamp':   166773646</pre>	<p>(re)Define, Define Options, System Balance, Improve</p>	<p>Critical Thinking Self-direction Modelling Debugging Decision Making</p>

<p><b>Event:</b> Delete Object  <b>Data:</b>                  -Number of existing categories (rows of the Array)                  -Name of new category                  -Timestamp</p>	<p><b>Triggered:</b> When the designer deletes a specific object in the Edit Game Database</p>	<pre>                 Json Object                 {                 'id':                 this.editdbID,                 'type': " database",                 'event":                 "delete_object",                 'state':                 {                 'info':                 {                 idcounter:                 '13', name:                 'swimming',                 type: 'text',                 times: 6},                 event_count:                 6},                 'timestamp':                 166773646                 </pre>	<p>(re)Define, Define Options, System Balance, Improve</p>	<p>Critical Thinking                  Self-direction                  Modelling                  Debugging                  Decision Making</p>
<p><b>Event:</b> Select all objects in a specific category  <b>Data:</b>                  -Number of existing categories                  -Name of new category                  -Timestamp</p>	<p><b>Triggered:</b> When all available objects have been selected to be assigned to a specific category</p>		<p>Define options                  Explore</p>	<p>Automation                  Decision making</p>
<p><b>Event:</b> Have no object selected in a category  <b>Data:</b>                  -Number of existing categories                  -Name of new category                  -Timestamp</p>	<p><b>Triggered:</b> When none of the available objects has been selected to be assigned to a specific category</p>		<p>Define options                  Explore</p>	<p>Abstraction                  Decision Making</p>
<p>The following game design events are generated only if the designer (e.g. teacher) has activated and defined them in the SORBET analytics options</p>				
<p><b>Event:</b> Designer Activity  <b>Data:</b></p>	<p><b>Triggered:</b> When the designer has made a number of</p>	<pre>                 Json Activity                 {                 'id':                 this.editdbID,                 </pre>	<p>Test                  Prototype                  Explore</p>	<p>Modelling                  analysis                  Decision Making</p>

<p>-Number of columns (Categories) -Number of lines (Objects) -Type of user activity ("low", "medium", "high") -Timestamp</p>	<p>changes to the database values, rows and columns, that is considered a milestone for his/her design activity (e.g. 30 changes). There are 3 types of activity: low, medium, high. The 3 thresholds are indicated by the activity designer (e.g. Teacher) in the SORBET analytics options</p>	<pre> 'type': " database", 'event": "design_activity",   'state':     {'info':       {columns: 5,        rows: 20}},     'value':       "Medium"}, 'timestamp': 1667736467 }         </pre>	<p>Understanding Solution</p>	<p>Critical Thinking</p>
<p>Game Playing Mode</p>				
<p>Gameplay - The following events are triggered while students use SORBET in Play Mode (log message: "Entering Play Mode")</p>				
<p><u>Event:</u> Click and drag a falling object <u>Data:</u> -Times the mouse has been clicked and dragged so far -Message if after a while there is no movement from the player (Inaction) -Timestamp</p>	<p><u>Triggered:</u> when a player enters Play Mode and press the "start" button</p>	<pre> Json object {   'id': gameId,   'type':     "playmode",   'event":     "clickanddrag",     'state':       {'game_progress':         {'fallingObj':           {Classified: 5,            Unclassified:             2, 'score': 5}           'Object':             {name: 'verb',              type: 'text',              times: 5},             event_count:               4},         'timestamp' :           1667736467         } }         </pre>	<p>Exploring Understand</p>	<p>Computational Thinking Decision Making Critical Thinking Pattern recognition</p>

<p><u>Event:</u> Correct or Wrong Sorting Objects <u>Data:</u> -Objects Classification (Correct/Wrong) -Timestamp</p>	<p><u>Triggered:</u> when the falling object touches the box of the category and it turns red if wrong or green if correct.</p>	<pre>Json object {   'id': gameId,   'type':   "playmode",   'event":   "correct",   'state':   {'game_progress':   {'fallingObj':   {Classified: 5,   Unclassified: 2, 'score': 5}   'Object':   {name: 'verb',   type: 'text',   times: 5},   event_count: 4},   'timestamp' :   1667736467 }</pre>	<p>Exploring Understand</p>	<p>Modelling analysis Computational Thinking Decision Making Critical Thinking Pattern recognition</p>
<p><u>Event:</u> Game Over <u>Data:</u> -Values of the statistics and score at the time of the event -Timestamp</p>	<p><u>Triggered:</u> when the game finishes or stops by the player It needs to be a different event when it finishes because there are no objects left and when the player pressed the stop button?</p>	<pre>Json Object {   'id':gameID,   'type':   "playmode",   'event":   "game_over",   'state':   {'game_progress':   {'fallingObj':   {Classified: 5,   Unclassified: 2,   'score': 5}   'Object': {name:   'verb', type:   'text', times: 5},   event_count: 4},   'timestamp' :   1667736467 }</pre>	<p>Not applicable</p>	<p>Not applicable</p>
<p>The following play mode events are generated only if the designer (e.g. teacher) has activated and defined them in the SORBET analytics options:</p>				
<p><u>Event:</u> Player Activity <u>Data:</u></p>	<p><u>Triggered:</u> when the player has clicked and</p>	<pre>Json Object {   'id': this.gameID,</pre>	<p>Understand Empathize Explore options</p>	<p>Model analysis Systems thinking</p>

<p>-Values of the times the mouse clicked and moved the falling objects Type of players' performance "Low", "Medium", "High" -Timestamp</p>	<p>dragged a certain number of objects related to the total number of falling objects that is considered a milestone for his/her activity (e.g. 80 %). There are 3 types of activity: low, medium, high. The milestone percentages are indicated by the activity designer (e.g. the teacher) in the SORBET analytics options.</p>	<pre> 'type': "playmode", 'event": "player_activity", 'state': {'game_progress': {'fallingObj': {Classified: 5, Unclassified: 2, 'score': 5} 'value' : {'medium'}}}, 'timestamp': 1667736467 }                     </pre>	<p>Analyze data</p>	<p>Data comparison/analysis</p>
<p><u>Event:</u> Player Performance <u>Data:</u> -Values of the correct objects classifications -Timestamp</p>	<p><u>Triggered:</u> when the player has classified correctly a certain percentage of the falling objects that is considered a milestone for his/her activity (e.g. 30%). There are 3 types of performance: low, medium, high. The milestone percentages are indicated by the activity designer (e.g. the teacher) in the SORBET analytics options.</p>	<pre> Json Object { 'id': gameID, 'type': "playmode", 'event": "player_performance", 'state': {'game_progress': {'fallingObj': {Classified: 5, Unclassified: 2, 'score': 5} 'value' : {'medium'}}}, 'timestamp' : 1667736467 }                     </pre>	<p>Understand the problem Empathize Decision making System balance</p>	<p>System/model analysis Systems thinking Decision Making Critical Thinking</p>



## Appendix B: Server Configuration Specification

Server Description	Amazon Lightsail Cloud Ubuntu Server 22.04 4 vCPUs 16 GB RAM 320GB SSD 6TB Transfer
Networking	A single subnet that will initially be isolated from external networks for inbound connections. Outbound connections should be permissible so that we can take advantage of third-party services and have some administrative flexibility if new software is required.
Security	Initially access should be given through VPN, and once the solution is internally tested and ready to be released to schools, we can allow inbound access to selected networks.
SSH	Enable
Software	Apache Node.js Node-RED Mosquitto MQTT broker
Database	Amazon RDS (Managed Relational Database Service) PostgreSQL