TROMPA

TROMPA: Towards Richer Online Music Public-domain Archives

Deliverable 6.4

Working Prototype for Orchestras v2

Grant Agreement nr	770376
Project runtime	May 2018 - April 2021
Document Reference	TR-D6.4-Working Prototype for Orchestras v2
Work Package	WP6 - End User Pilots
Deliverable Type	Demonstrator
Dissemination Level	PU- Public
Document due date	February 28, 2021
Date of submission	February 28, 2021
Leader	RCO
Contact Person	Marcel van Tilburg (RCO)
Authors	David Linssen (VD), Cynthia Liem, Ioannis Petros Samiotis (TUD), Gonneke de Jong (RCO)
Reviewers	David Baker (GOLD) Aggelos Gkiokas (UPF)

Executive Summary

This deliverable is the second version of the demonstrator deliverable for TROMPA's orchestra use case. The main purpose of this document is to present the prototype(s) developed for the given use case, linking to the implementation, discussing main functionalities, and explaining how the work connects to the overall TROMPA project. The current prototypes evolved from an earlier release in M24 of the project, that went through intermediate usability and functionality evaluation with relevant audiences (**Deliverable 6.8 - Mid-term evaluation**). For the orchestras use case, these audiences consist of semi-professional and amateur youth orchestra players.

The deliverable contains three main sections. Section 2, the first main section, considers the main functionality of the prototype, and its alignment to the set requirements. We first discuss the background of the prototype. Outcomes of mock-up evaluation workshops, conducted with members of the Concertgebouworkest (RCO), showed that our initial focus on score annotations for professional orchestra players was not likely to lead to an outcome that the players of the orchestra would be ready to adopt. This, together with the notion that we generally lack digitally encoded scores that are not mere visual scans, led to a pivot towards collective crowd-assisted optical music recognition, to be supported by semi-professional and amateur orchestra players.

Following this pivot, an alternative use case scenario was adopted, in which student orchestras would launch collective score transcription campaigns through our prototype. From the perspective of a user, only a computer, tablet or smartphone and recent browser are needed. The user who needs for a campaign to be launched, can do this through the Campaign Manager system, by selecting a work and score in the CE, uploading an own score, or referring to a score in a SOLID pod. For users participating as collaborators in a campaign, the Campaign Manager also will be the place where tasks will be served. The final version of the Campaign Manager is mobile-friendly, such that tasks may be performed while on the go. Users have the option to share campaign links with other users, to further broaden the audience reach. Collaborators will work under a self-chosen nickname, and consent to the final outcome of their joint work being Creative Commons (CC BY 4.0) licensed.

The tasks that collaborators will be presented with, are generated by the Crowd Task Manager. The Crowd Task Manager processes PDF input data (image processing and segmentation), generates crowdsourcing tasks for the non-automated parts, and finally aggregates results to build an MEI version of the original music score. The human computation aspects of the Crowd Task Manager follow well-established studies on crowdsourcing, breaking down the problem of OMR into steps and tasks that can be performed by people on specific stages of the OMR process. The crowdsourcing tasks of the Crowd Task Manager have specific inputs (segments of the given score) and outputs (annotations), designed to be performed easily and efficiently by the users. These crowdsourcing tasks co-exist with automated methods such as measure detection, image segmentation and XML-tree aggregation, creating a hybrid system where human-machine collaboration achieves the shared goal of generating MEI orchestral pieces from PDF input.

Where the Crowd Task Manager initially served transcription and verification tasks, the current release accommodates much more user-friendly tasks, which involve clef detection, time signature detection, key signature detection, rhythm transcription, and pitch correction. These tasks are served through Scriptoria, a dedicated front-end server. Task prioritisation is done page by page; this way, at any point in time, collaborative work will yield an outcome that is as coherent as possible, counting from the start of the score. Communication between the Campaign Manager, Crowd Task Manager and Scriptoria goes through TROMPA's contributor environment. Links to relevant GitHub

repositories and demo videos are given in the document. The various systems supporting the prototype are built in a modular way, and can be easily adapted (e.g. for supporting other types of tasks) without raising too many interdependence issues.

The prototype integrates the measure detection methods from **Deliverable 3.4 - Visual analysis of scanned scores** to segment a PDF score into many small images, for which the tasks can be performed. Incentivisation mechanisms from WP4 are included (e.g. explicitly giving feedback to users on progress), and the Crowd Task Manager implements the vision of hybrid annotation workflows that was outlined as part of TROMPA WP4, following campaign designs that are understandable to broad audiences, even those who may not be extremely skilled at reading music.

From TROMPA's WP5, the prototype generally communicates through the TROMPA Contributor Environment specified by **Deliverable 5.1 - Data Infrastructure**, and incorporates TROMPA's **Deliverable 5.2 - Score Edition Component**, while the modular architecture would allow for future easy integration of components such as the **Deliverable 5.5 - Annotation Tools**.

With the pivot from professional orchestras to semi-professional and amateur (student/youth) orchestras, the intention had been to focus on younger audiences that would more naturally be open to technological innovation, and more likely to engage actively in collaborative campaign setups. However, the COVID-19 crisis deeply affected these target groups, who have been unable to rehearse, nor perform. With the lack of concrete projects for these orchestras, the initial plan to collaboratively work with on repertoire that the orchestras would rehearse and perform turned out infeasible. Generally, the crisis also affected motivation, and made it considerably harder to recruit participants for user studies. To maximize the amount of potential participants, while acknowledging the insecurity and distraction of the COVID-19 crisis, we decided to write all student orchestras in The Netherlands, as well as the Nationaal Jeugd Orkest (NJO), an orchestra for young professionals at conservatoires. With the help of the academy coordinator of the RCO, we contacted them, invited them for several usability studies, and offered concrete rewards to participants associated with the RCO brand. Several dozens of orchestra members informed the **Deliverable 6.8 - Mid-term evaluation** outcomes, which were important for understanding feasibility of campaigns, and the understandability of tasks. For the upcoming final evaluations, orchestra members will be invited to try out the various tasks in a test campaign, with evaluations being hosted in online sessions during multiple evenings in March.

Version Log		
#	Date	Description
v0.7	February 17 2021	Version for internal commenting
v0.8	February 22 2021	Version submitted for internal review
v1.0	February 28 2021	Final version

Table of Contents

1. Introduction

This deliverable is the second version of the demonstrator deliverable for TROMPA's orchestra use case. The main purpose of this document is to present the prototype(s) developed for the given use case, linking to the implementation, discussing main functionalities, and explaining how it connects to the overall TROMPA project.

The structure of the deliverables is shared amongst all deliverables D6.3 - D6.7 and contains three main sections. Section 2 presents the main functionalities of the pilot by providing screenshots, a URL where we can access the pilot software, and a demo video of the pilot along with instructions on how to use the pilot. This section also will discuss how the campaign manager and crowd manager interact, and highlight changes between the first prototype release (M24) and the current one. Section 3 is related to the integration into the pilot of technologies from WP3, components and methodologies from WP4, and the integration of WP5 components in the pilot. Section 4 is dedicated to the user evaluation of the pilot, discussing what audiences were targeted, how they were reached, considering the impact of the COVID-19 crisis, and what questions to keep in mind, going forward towards the final evaluation of this prototype. Finally, a conclusion is presented in Section 5.

2. Main functionalities of the prototype

2.1. Background

The main interest in the orchestras use case, as set out in **Deliverable 6.1 - Final Mock-ups Testing**[1] is to have TROMPA technology enable the establishment of a repository of public domain scores and orchestral parts in true digital form, including an application to access these.

Today, orchestras have to obtain quality, performable scores from commercial publishers, even though the musical works themselves are often in the public domain. This gives publishers power over orchestras, since publishers can exercise intellectual property rights on score editions, leading to limitations in distributing the performances further via audiovisual recordings: hiring performance materials (scores and parts) is expensive and each use other than live performance is charged extra. Secondary, current practice of publishers is based around paper distribution of scores, which creates administrative overhead for orchestra librarians and makes annotation of scores for performance cumbersome and preservation of annotations for musicology challenging.

Based on these challenges, Videodock (VD) designed a mock-up application for collaborative score annotations within the orchestra. The mock-up focused mainly on the score annotation interaction, based on data from interviews with members of the Royal Concertgebouw Orchestra (RCO). The mock-up application then was tested during a workshop with RCO members. This resulted in three key outcomes:

- First, a set of recommendations to improve annotation interaction.
- Second, the insight that RCO members weren't likely to use such applications themselves, simply because the day-to-day practice of a top-tier orchestra wouldn't allow for it. Being a musician at RCO is a very intense job and adding a new application to the musicians' workflow would disrupt it.
- Third, all participants agreed on the practicality of a shared annotation tool for rehearsal and research. They highlighted the usefulness of such a tool for both amateur and professional musicians on the condition that there would be **a significant amount of high-quality, performable scores available** to make this digital workflow sustainable.

Currently this is not the case: most scores are available as either scanned representations of their paper originals or exist in MusicXML, a digital representation of the engraved score. Most public domain scores are now available as PDF (via, for instance IMSLP). Creating a digital score annotation workflow for orchestras requires scores to be available as a digital object that could be opened and edited in score editing applications like Dorico or Sibelius, or in new applications with similar capabilities. Current MusicXML, a common interchange format, is unsuitable, because MusicXML is a representation of the music engraving, not of the musical work itself. MusicXML reflects, for instance, decisions made by the music engraver to aid the performer, which omit historically and musicological information. Within TROMPA, we have therefore chosen to adopt the Music Encoding Initiative file format[2] (MEI) as a suitable file format. With MEI-based scores, a number of opportunities for orchestras present themselves, such as new ways of annotating them and keeping track of these markings.

[1] This deliverable is confidential to the consortium only

[2] https://music-encoding.org/

Switching to digital (annotation) workflows depends on sufficient high quality scores being available for orchestras. And without an installed base of digital score consumers, there is no incentive for publishers to deliver their library of scores in this new format.

This is a typical chicken-and-egg-type problem: the lack of high-quality, performable digital scores, because of paper-based workflows and software, blocks the adoption of digital workflows in orchestras. And without demand from performers, there is little incentive for current market players to invent and evolve their music libraries and software towards digital workflows.

2.2. General presentation of the prototype

With the above in mind, VD in collaboration with RCO and TUD, developed a prototype scenario where an amateur orchestra, in need of a performable score, collaborates in the conversion of a PDF-based score from the IMSLP library into a high quality MEI rendering of the same score. Such a prototype would allow us to gather insights on distributed collaboration within orchestras, add more MEI based content to the TROMPA CE library and provide an opportunity to road test the hybrid OMR functionality that TUD has been developing under WP4. A conceptual user-journey was drafted for the design of the prototype, as illustrated in Figure 2.1.

Figure 2.1. TROMPA orchestras conceptual user journey.

2.2.1. Use case scenario

- The Delft student orchestra Krashna Musika wants to perform Gustav Mahler's 1st Symphony. A public domain version is available on IMSLP, however the quality of the score is insufficient for performance. The orchestra decides to use the TROMPA prototype to improve the score.

- The orchestra's leader delivers the IMSLP URL to the Crowd Task Manager. This system will retrieve the score from IMSLP and generate crowdsourcing tasks per segment of the music score.
- After completion of all tasks, the resulting MEI file will be made available via GitHub and will be made discoverable through the CE-API. The orchestra can then use this digitised version for their performance.

2.2.2. Access information

The **Pilot prototype** is accessible through the TROMPA subdomain[3]. The pilot prototype is currently available in English. The following **Demo videos are available:**

- An introductory video presenting the prototype scenario and the main features of the first version[4];
- A video presenting additional or updated user-facing functionality in the second version[5];
- A video presenting the different task types and how these can be generated from the Crowd Task Manager[6].

2.2.3. System requirements

The prototype only has very general system requirements, such that it is accessible to broad audiences:

1. A computer, tablet or smartphone.
2. A recent web browser.

2.2.4. System outline

The prototype is facilitated by several systems, with the Contributor Environment as intermediate exchange layer:

- The **Campaign Manager**, in which users can initiate a new campaign, and in which campaign tasks will be served to people joining the campaign;
- The **Crowd Task Manager**, which, based on a submitted score PDF, will break the PDF information down into tasks that can be performed by the crowd, and manages the scheduling of these tasks and aggregation of their outcomes;
- **Scriptoria**, a dedicated front-end server for the currently designed crowd tasks.

The Crowd Task Manager and Scriptoria jointly implement TUD's WP4 work on hybrid pipelines for crowd-assisted Optical Music Recognition (OMR). Architecturally, they are deliberately detached from the campaign manager, such that in the future, other types of campaigns and crowd tasks could be facilitated under the same architecture.

A global system outline, illustrating how these systems interact, is illustrated in Figure 2.2. In Section 2.3, we will discuss the main functionalities of the Campaign Manager, in Section 2.4, that of the Crowd Task Manager and Scriptoria systems, and in Section 2.5, more technical details will be given on how the systems communicate through the Contributor Environment.

[3] https://campaigns.trompamusic.eu
[4] https://storage.trompamusic.eu/video/orchestra_prototype_introduction.mp4
[5] https://storage.trompamusic.eu/video/orchestra_prototype_second_iteration_features.mov
[6] https://storage.trompamusic.eu/video/orchestra_prototype_crowd_task_manager.mp4

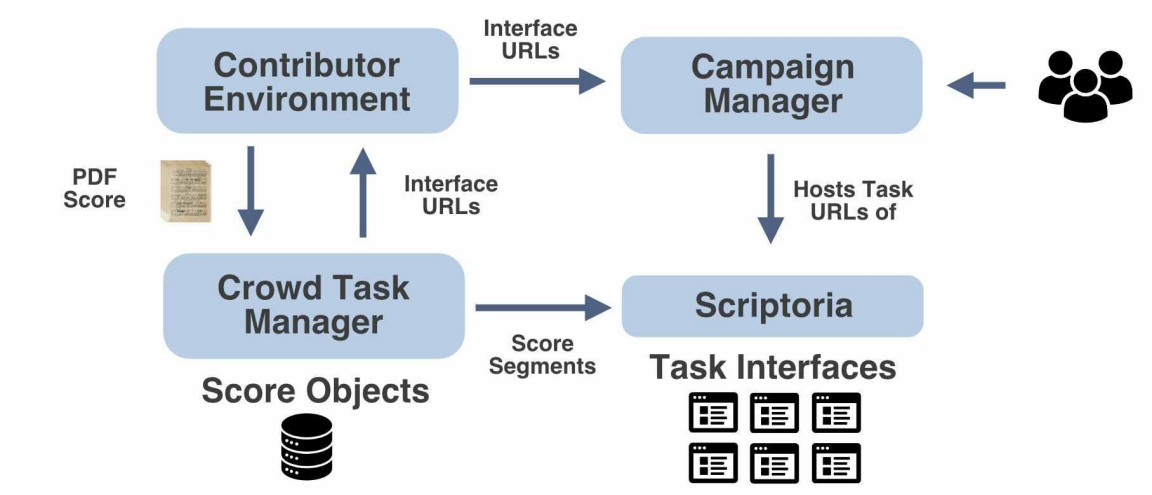

Figure 2.2. Global outline of the systems involved in the prototype.

2.3. Campaign manager

In this section, we will focus on the Campaign Manager system. First, we give an overview of the functionalities as realised over the two version releases (in M24 and M34 of the project); then, we give a walk-through of the main functionalities from a user perspective, including screenshots. The Campaign Manager is published in the TROMPA GitHub repository.[7]

2.3.1. Main functionalities per version

In the first prototype release (in M24 of the project), the majority of functionalities for the Campaign Manager was implemented, as listed below.

For the general public:

- Create awareness of TROMPA and specific campaign objectives:
  - View TROMPA objectives;
  - Start your own campaign (by submitting a request to the administrator);
  - Contribute to a running campaign;
  - Provide a global overview of the campaign process.

For campaign initiators:

- Setup a campaign for the MEI conversion of a PDF score:
  - Add an IMSLP URL to retrieve score from IMSLP (via backend, not visible);
  - Add a campaign name, campaign objective and campaign deadline date (via backend, not visible);
  - encourage participants to collaborate on the campaign via progress emails.

[7] https://github.com/trompamusic/trompa-campaign-manager

For campaign collaborators:

- Participate in crowd-assisted OMR campaigns:
  - View;
  - Provide a username (for named contribution purposes);
  - Execute tasks -> perform given tasks (user choices translate to MEI transcriptions);
  - View the progress of the campaign.
- Provide consent:
  - Collaborators must actively consent with CC licensing of the created MEI file.
- Enlarge the amount of contributors to the campaign:
  - Share the campaign landing page via social media share buttons or via a share link.
- Configure settings:
  - Forget stored username.

After the v1 release, further work was performed to strengthen communication through the Contributor Environment (CE), grant more autonomy to users to initiate an own campaign, and to allow for mobile-friendly, responsive task display. The latter was done, as the user studies in **D6.8-Mid-term Evaluation**[8] indicated that orchestra members may be willing to contribute to tasks to kill time while being 'on the go'. As a consequence, for version 2 of the prototype, the following functionalities were added to the campaign manager:

- Users can now initiate their own campaigns by selecting a composition & score that is referenced in the TROMPA CE (through the integrated TROMPA Multi Modal Component[9], part of the **Deliverable 5.3 - TROMPA Processing Library**[10]) and set up a title, description and deadline for the campaign.
- Users can now upload their own scores to the CE for initiating campaigns (PDF, MEI, MusicXML).
- Users can now submit files stored in their SOLID pods for initiating campaigns.
- Users can view the MEI outcome of a campaign from the Campaign landing page thanks to integration of the TROMPA Score Edition Component (see **Deliverable 5.2 - Score Edition Component**[11]).
- For contributors to a campaign, the app has been updated for use on mobile devices.
- Users can now view and navigate to all active campaigns from the home page.
- We have added the ability to pass the users' nickname to the CE and connected applications.
- To improve the experience for collaborators, we have created visual design guidelines for responsive collaboration tasks and agreed with TUD on implementation in the task engine system. The guidelines can be viewed online[12].
- The mechanism to communicate between the Campaign manager, the Task Engine and the TROMPA CE has been improved to provide users with better insight on task and campaign progress. This mechanism will provide users with a better experience moving from task to task.
- The shareability of the campaign page has been improved.

[8] https://trompamusic.eu/deliverables/TR-D6.8-Mid_Term_Evaluation.pdf
[9] https://github.com/trompamusic/trompa-multimodal-component
[10] https://trompamusic.eu/deliverables/TR-D5.3-TROMPA_Processing_Library_v2.pdf
[11] https://trompamusic.eu/deliverables/TR-D5.2-Score_Edition_Component_v2.pdf
[12] https://storage.trompamusic.eu/design/Responsive_collaboration_tasks.pdf

2.4.1. Collaborate on a campaign

The pilot core functionality is the collaboration on crowd-assisted OMR campaigns, aimed at converting PDF scores to MEI files. A campaign is the collective effort of a group of collaborators to digitise or digitally enhance a score, during a specific timeframe. From the Home page (see Figure 2.3), clicking on Join Campaign in the top menu, the user is redirected to the campaign main page.

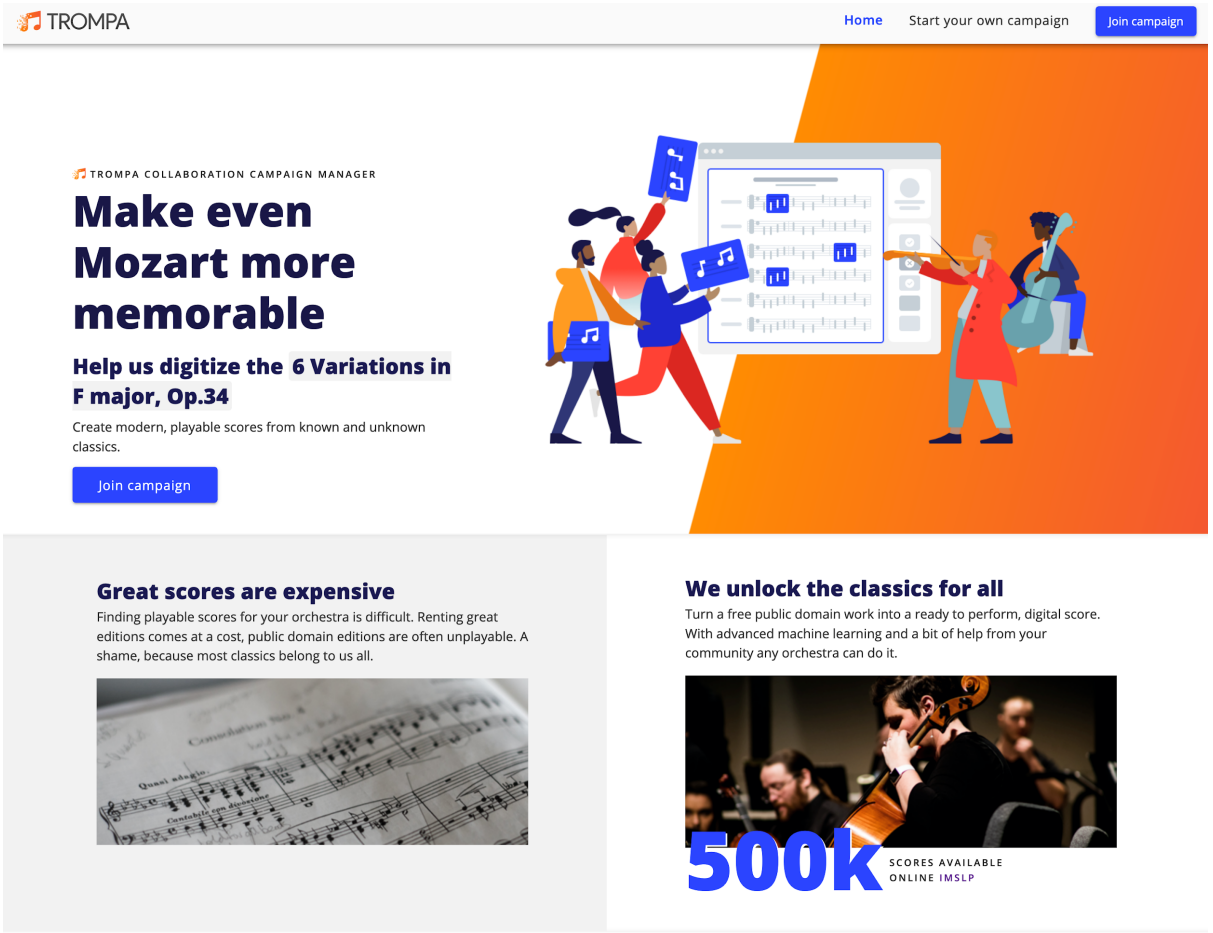

Figure 2.3. *Home* page. Users can contribute to a campaign via the 'Join campaign' button.

2.4.2. The Campaign page

For each campaign, users can visit a specific campaign landing page (Figure 2.4). On this page, they can view the campaign objectives, its progress towards completion and some general information about the work in progress. There are three main tasks on this page:

- *Joining a campaign*: This will bring up the tasks pages and will present tasks to the user for completion.
- *Subscribe for updates*: This will prompt the user to enter name and email address and put the user on the subscription list for a specific campaign in the email newsletter tool. From this tool we can send automated updates on the campaigns progress.
- *Share this campaign*: This allows the user to share this campaign with other people via buttons for Facebook, Twitter and email or by copying the link and sending it via any other platform.

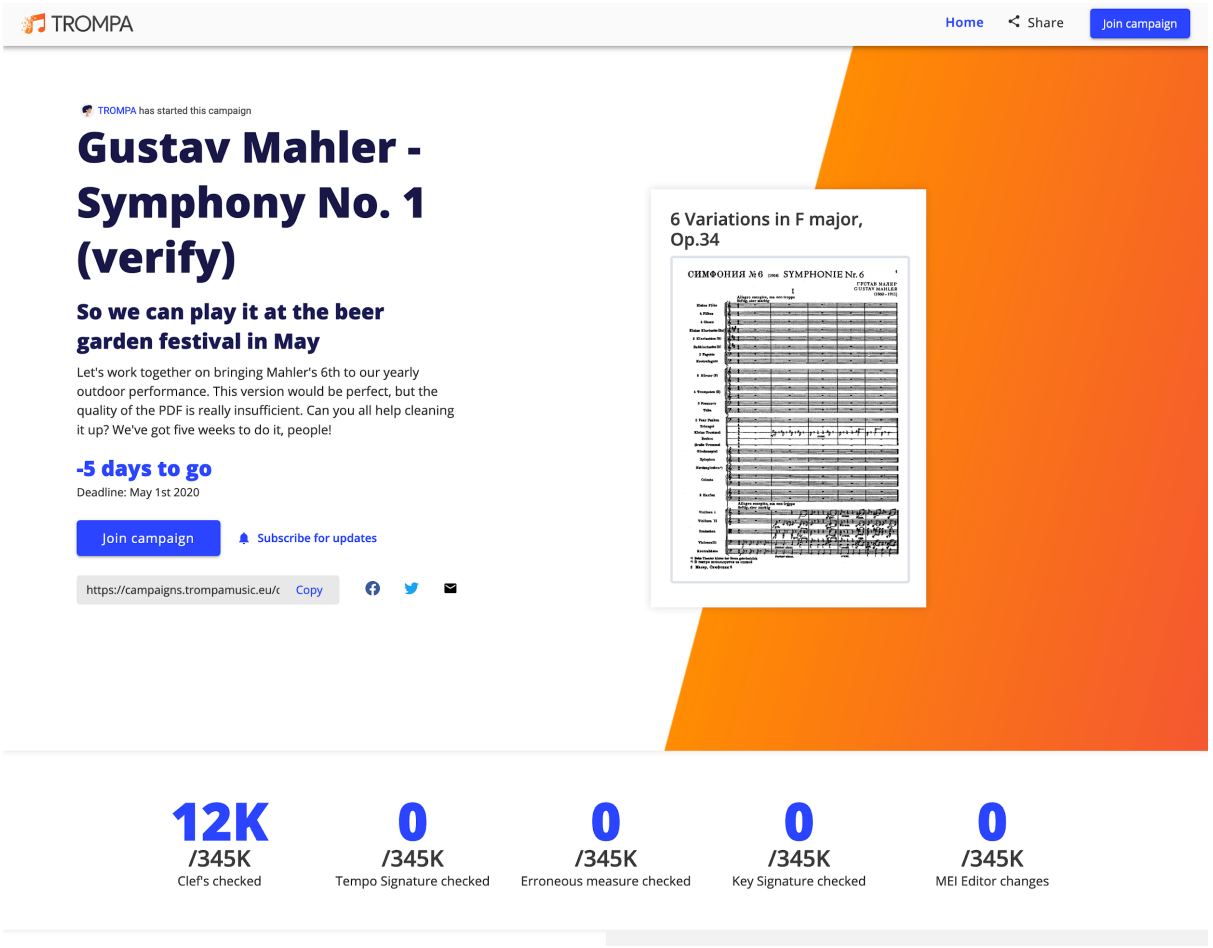

Figure 2.4. Campaign landing page. Users can view campaign objectives, subscribe for updates and invite others to the campaign via social media. The *Join campaign* button starts the distribution of tasks.

2.4.3. Attribution for CC licensing

Within TROMPA, considering our project focus on public-domain resources, all created MEI files will be published under a Creative Commons licence (CC BY 4.0). The licence requires contributors of a work to be attributed. This is why we provide the ability to contribute under a specific nickname. Before any contribution is made to a work, users will be asked for consent to the CC licensing (Figure 2.5) and will be given the opportunity to set a nickname. If users do not choose a name, their work will be published under a randomised nickname.

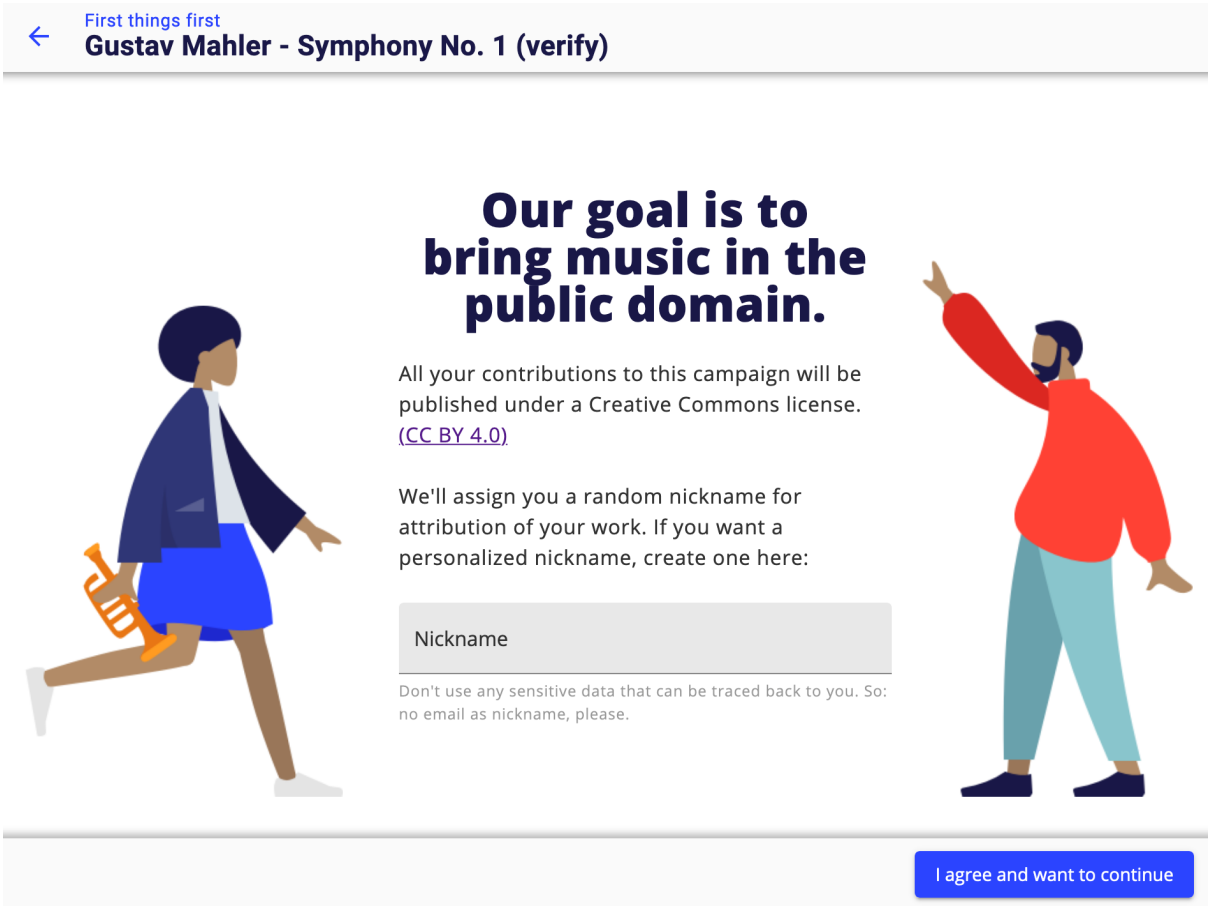

Figure 2.5. Creative Commons Consent Before contributing, users will have to agree to the Creative Commons licensing, either under a chosen nickname or a randomly assigned one.

2.4.4. The Tasks page and the Crowd Task Manager system

After joining a campaign, and giving consent, the user will be served a series of tasks to complete. The nature and recognition of these tasks are generated by the Crowd Task Manager, a crowd-assisted Optical Music Detection (OMR) system, developed by TU Delft (TUD), which has originally been documented as part of **Deliverable 4.4 - Hybrid Annotation Workflows**[13], and of which the main functionalities are summarised in the following section.

2.5. Crowd Task Manager and Scriptoria

In this section, we discuss the Crowd Task Manager and Scriptoria systems, which together power the crowd-assisted OMR tasks. After a general introduction of the Crowd Task Manager, we discuss the functionality development timeline, and illustrate how task designs evolved over the version releases.

2.5.1. General introduction of the Crowd Task Manager

The Crowd Task Manager is a system TUD built to accommodate a crowd-assisted OMR pipeline. As such it processes PDF input data (image processing and segmentation), generates crowdsourcing

[13] Visible to TROMPA consortium members only

tasks for the non-automated parts, and finally aggregates results to build an MEI version of the original music score. Each task generated by the Crowd Task Manager, utilises a dedicated front-end which can be distributed to any crowdsourcing platform of choice (currently: the Campaign Manager). A diagram showing all components of the Crowd Task Manager is given in Figure 2.6.

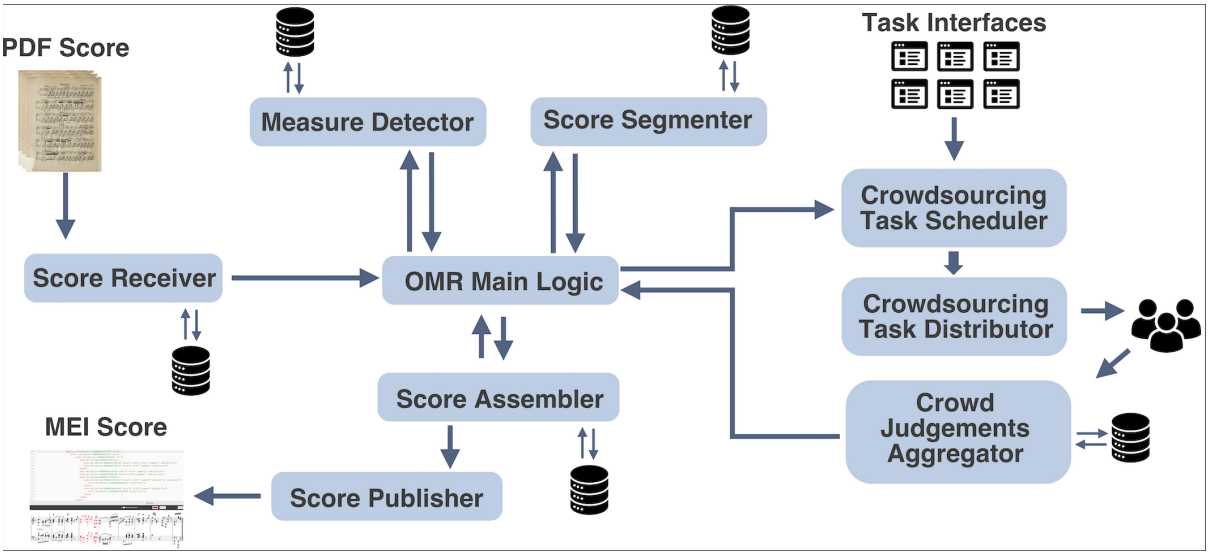

Figure 2.6 Schema of Crowd Task Manager's modules.

The human computation aspects of the Crowd Task Manager follow well-established studies on crowdsourcing, breaking down the problem of OMR into steps and tasks that can be performed by people on specific stages of the OMR process. The crowdsourcing tasks of the Crowd Task Manager have specific inputs (segments of the given score) and outputs (annotations), designed to be performed easily and efficiently by the users. These crowdsourcing tasks co-exist with automated methods such as measure detection, image segmentation and XML-tree aggregation, creating a hybrid system where human-machine collaboration achieves the shared goal of generating MEI orchestral pieces from PDF input.

Some final system requirements for our prototype were to: 1) design the system in a modular and distributed fashion and 2) store in the system all the data resulted from processes throughout each of the steps of our OMR pipeline, to make them easily accessible by all the system's modules. We set the first requirement to enable scalability and support easier maintainability. Each of the modules in the prototype represents a step on the OMR pipeline and serves a specific functionality. This helps to easily replace parts of the pipeline with more sophisticated ones, without breaking the overall operability of the system. We implemented a central module which holds the logic steps of the OMR pipeline which sends messages that dictate which of the modules should be activated and when. Each module inside the pipeline, imports data from our local database and stores data to it to make them available to the other modules.

2.3.1. Main functionalities per version

Based on the prototype's requirements and architecture that we described above, the first release of the Crowd Task Manager had the following functionalities:

- Segmentation of an input PDF file of a music score, on a measure level;
- Creation of an MEI file which contains information of regions of segments per page, alongside their corresponding measure headers;

- ❖ Creation of crowdsourcing tasks to transcribe the segments of the music score into MEI format;
- ❖ Separation of crowdsourcing tasks into **Transcription** and **Validation** tasks;
- ❖ Aggregation of results from the crowdsourcing tasks, which will dictate the content under each measure header in the MEI.

This first release especially focused on arranging the inner workings of the Crowd Task Manager, and serving tasks that could be hosted by the Campaign Manager in iframes. As for task design, for this first release, two basic task types were made available: transcription tasks (illustrated in Figure 2.7) and verification tasks (illustrated in Figure 2.8). The transcription tasks required participants to understand both music notation and writing XML, and therefore would need to be completed first. Then, verification tasks could be performed by any member of the orchestra, or even any member of the broader public.

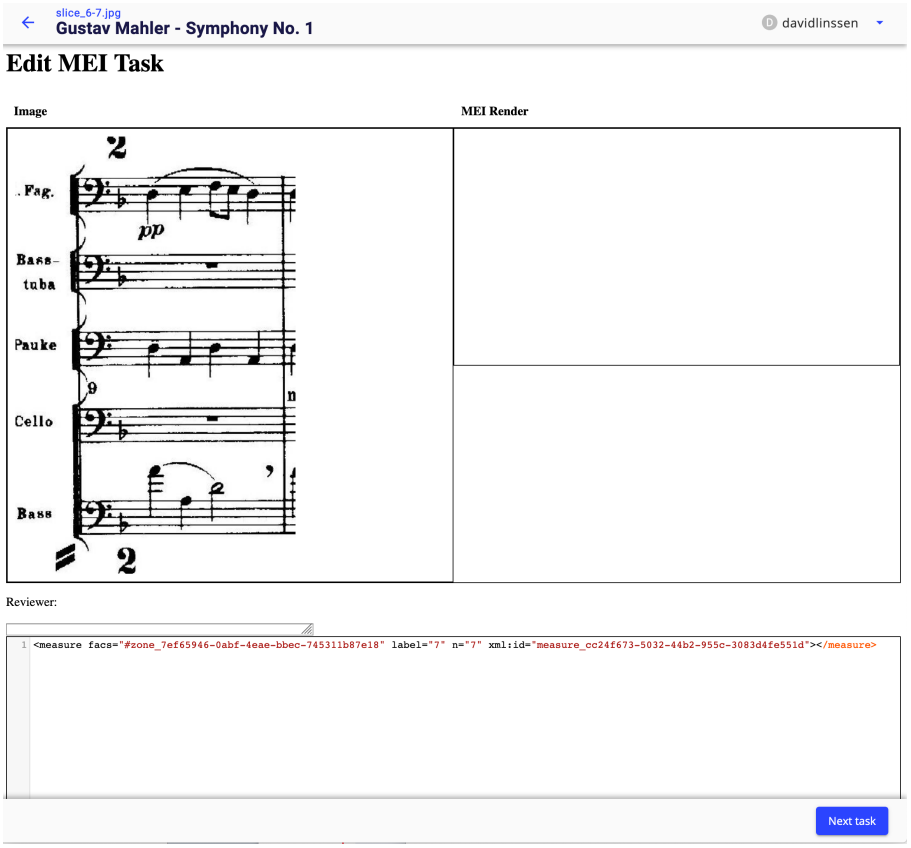

Figure 2.7. An example of an MEI transcription task within a campaign.

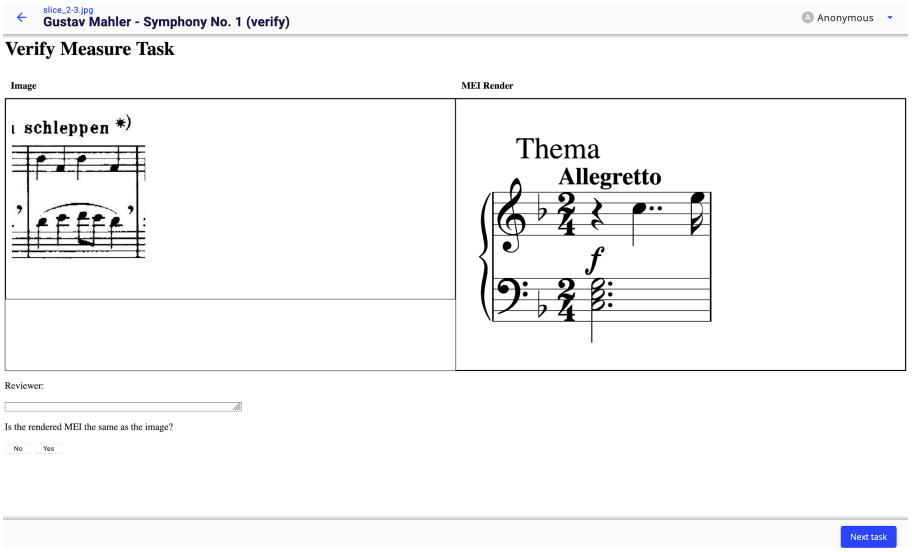

Figure 2.8. An example of an MEI verification task within a campaign.

For the second release of the Crowd Task Manager, several updates have been performed. The initial MEI transcription task was not user-friendly, and required contributors to have a basic understanding of XML, which would increase both the difficulty of the task and the time needed to perform it. Furthermore, since these crowdsourcing tasks would be targeted at semi-experts (youth/student members of orchestras), a certain high level of input was expected. Their expertise, combined with majority voting aggregation and tree aligning algorithms, would ensure high quality of output, therefore rendering possible verification tasks inessential. Verification tasks are supported in the Crowd Task Manager for the cases of inconsistent expertise in the crowd, but for the current use case, they were omitted, to ensure a more concise campaign, considering always the high expertise of the expected crowd.

When a PDF score is sent to the Crowd Task Manager, it first detects measure blocks per page in the PDF, and subsequently individual staffs, following the measure detection methods described in **Deliverable 3.4 - Visual Analysis of Scanned Scores**[14]. As the extraction process may not be 100% accurate, this procedure requires a human post-check before proceeding.

To allow for coherent completion, we then implemented a revised scheduling algorithm for the Crowd Task Manager, which follows a hierarchy of importance for MEI elements. For each measure, the clef, key and time elements are essential, as they could alter all subsequent music elements (notes/rests), which depend on them. Even when a campaign would not fully manage to conclude, it also is important to have as coherent and as comprehensive output as possible, rather than having contributions at random places in the large score.

Therefore, following the **order of the pages** of a given score, **for each staff in the page**, and **for each (part) segment within the staff**, the task scheduler takes the following prioritisation:

1. Detect if a clef exists in the given segment and if yes, select the correct one (Figure 2.9).
2. Detect if a time signature exists in the given segment and if yes, indicate the correct signature (Figure 2.10).
3. Detect if a key signature exists in the given segment and if yes, select the correct annotation (Figure 2.11).

[14] https://trompamusic.eu/deliverables/TR-D3.4-Visual_Analysis_of_Scanned_Scores.pdf

4. Rhythm transcription: transcribe the correct sequence of notes and rests, alongside their duration on a given segment of the original score (Figure 2.12).
5. Adjust to the correct pitches (Figure 2.13).

The current tasks are user-friendlier, and adhering more to the principles of microtask crowdworking design. The clef, key and time signature recognition tasks were evaluated as part of **Deliverable 6.8 - Mid-term Evaluation**; the note/rest transcription task is the result of usability evaluation that took place halfway February 2021 and followed the same structure as previous evaluation sessions (the report of which will be part of **Deliverable 6.9 - Final Evaluation**[15]).

Following the aggregation of user inputs, these are combined in the Crowd Task Manager to build up a MEI representation of the original score. Each increment is committed and then published on a dedicated git repository on GitHub[16]

For the front-end of these tasks, several changes were also made for the v2 release. Next to the Crowd Task Manager, a dedicated front-end server called Scriptoria was developed, to allow dynamic rendering of UI elements and dynamic route matching for the different types of tasks. Scriptoria is a NodeJS[17] server which hosts all the necessary front-end components such as interfaces, UI elements and dedicated task type components, while handling communications with the Crowd Task Manager through Axios[18]. When the input score is segmented by the Crowd Task Manager, each segment is stored in a MongoDB[19] database, alongside their identifiers and they become available in the Crowd Task Manager's API. Scriptoria can access these segments and render their images dynamically on the browser, while the user input is translated to MEI headers and communicated back to the Crowd Task Manager through the API. Both the front-end (Scriptoria) and back-end (Crowd Task Manager) of this crowdsourcing system, are hosted on the Dutch national e-infrastructure with the support of SURF Cooperative[20]. Their source code is published in the TROMPA repository on GitHub[21,22].

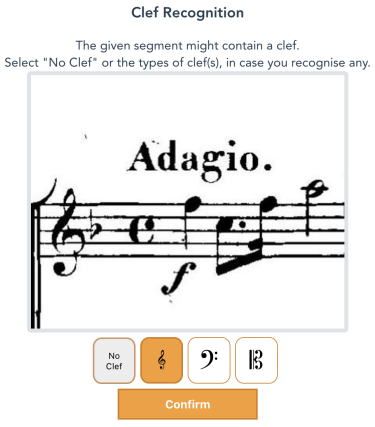

Figure 2.9. Clef Recognition task, as rendered by Scriptoria.

[15] https://trompamusic.eu/deliverables/TR-D6.9-Final_Evaluation.pdf
[16] https://github.com/Crowd-Transcribed-MEI-Repositories
[17] https://nodejs.org/en/
[18] https://github.com/axios/axios
[19] https://www.mongodb.com
[20] https://www.surf.nl/en/research-ict
[21] https://github.com/trompamusic/crowd_task_manager
[22] https://github.com/trompamusic/scriptoria

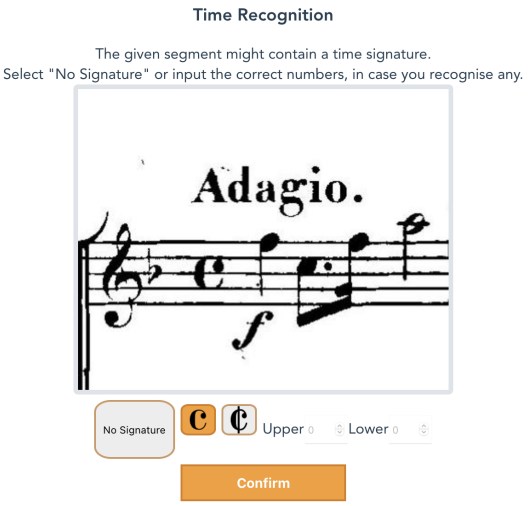

Figure 2.10. Time Signature Recognition task, as rendered by Scriptoria

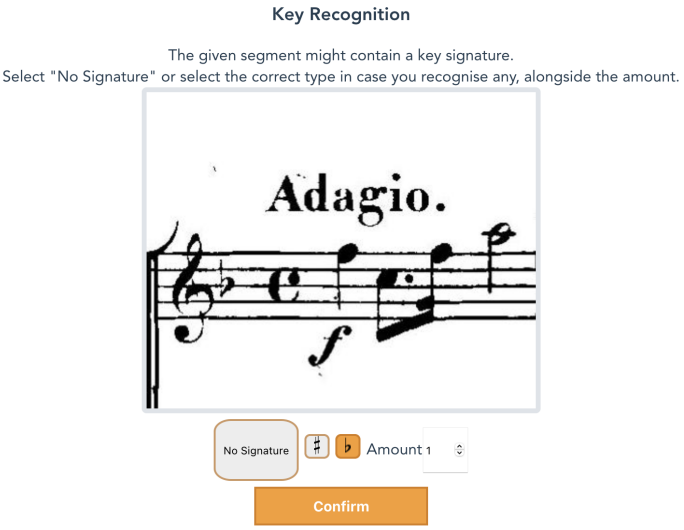

Figure 2.11. Key Signature Recognition task, as rendered by Scriptoria.

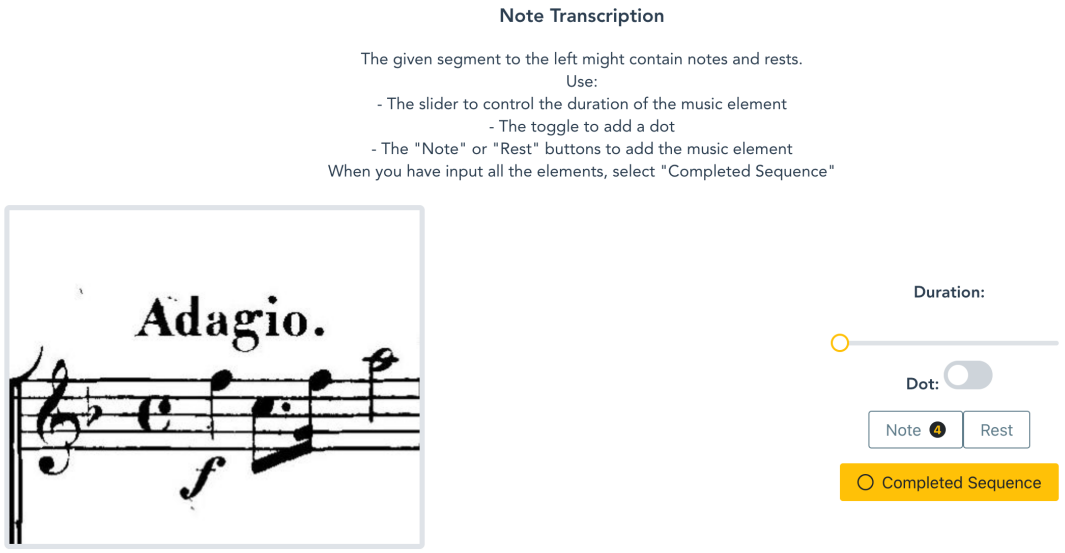

Figure 2.12. Rhythm Transcription task, as rendered by Scriptoria.

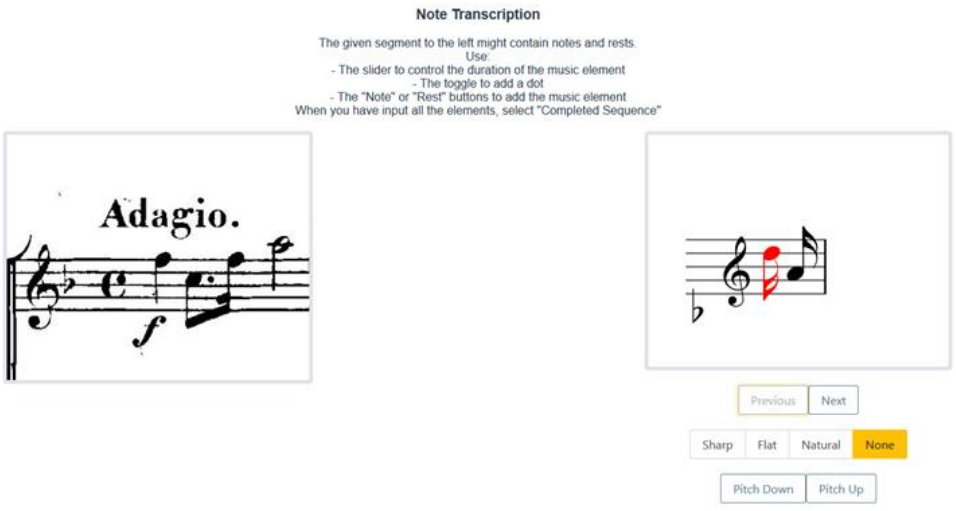

Figure 2.13. Pitch Correction task, as rendered by Scriptoria.

2.5 Interaction between systems

The Campaign Manager, the Crowd Task Manager and Scriptoria communicate through the Contributor Environment, through the following integration scheme:

- A potential user who wants to start a crowdsourcing OMR campaign uploads a PDF file of a music score through the Campaign Manager to the Contributor Environment. This PDF file may come from a public-domain repository, or be hosted in a Solid Pod.
- The file is fetched by Crowd Task Manager and the system generates crowdsourcing tasks per segment of the music score.

- ❖ The crowdsourcing tasks are distributed to the Campaign Manager which hosts the microtasks and users can enter the platform to perform them.
- ❖ The results of the crowdsourcing tasks are directly stored in the Crowd Task Manager and the status of each microtask is communicated with the Contributor Environment.

The Crowd Task Manager communicates with the Contributor Environment in the following fashion:

- ❖ The CE uses ControlActions[23] (also see **Deliverable 5.1 - Data Infrastructure**[24]) in their GraphQL API to model campaigns and tasks in the CE. The Crowd Task Manager gets alerted once there is a new ControlAction that relates to a campaign.
- ❖ Once the pipeline has retrieved and processed the music score for crowdsourcing tasks, ControlActions are created on the CE using the RequestControlAction mutation. The resulting ControlAction is related to the campaign in the CE and contains a URL to the crowdsourcing task, created by the Crowd Task Manager.
- ❖ When the Crowd Tasks Manager receives the first result from a task, the status of the related ControlAction is set to ActiveActionStatus.
- ❖ Once the Crowd Tasks Manager gets results from a task that forms a consensus, the ControlAction status is updated to CompletedActionStatus.
- ❖ For a more in-debt description of how crowdsourcing tasks are generated and the communication schemes between the above platforms, please read the 2nd version of the **Deliverable 4.4-Hybrid Annotation Workflows**[25].

2.6. Incentivisation strategies

In order to incentivise the contributors, campaigns include incentivisation mechanisms as suggested in **Deliverable 4.3 - Crowd Incentivisation Mechanisms**[26]. More specifically:

- ❖ On the landing pages we stress the value of the campaigns.
- ❖ We make sharing easy in order to let participants use their social networks to bring in more contributors.
- ❖ We present dynamic information regarding the progress of the crowdsourcing tasks.
- ❖ We keep contributors engaged through messages and announcements after completing tasks and through the campaign update emails.

[23] https://schema.org/ControlAction
[24] https://trompamusic.eu/deliverables/TR-D5.1-Data_Infrastructure_v2.pdf
[25] This deliverable is confidential to the consortium only
[26] This deliverable is confidential to the consortium only

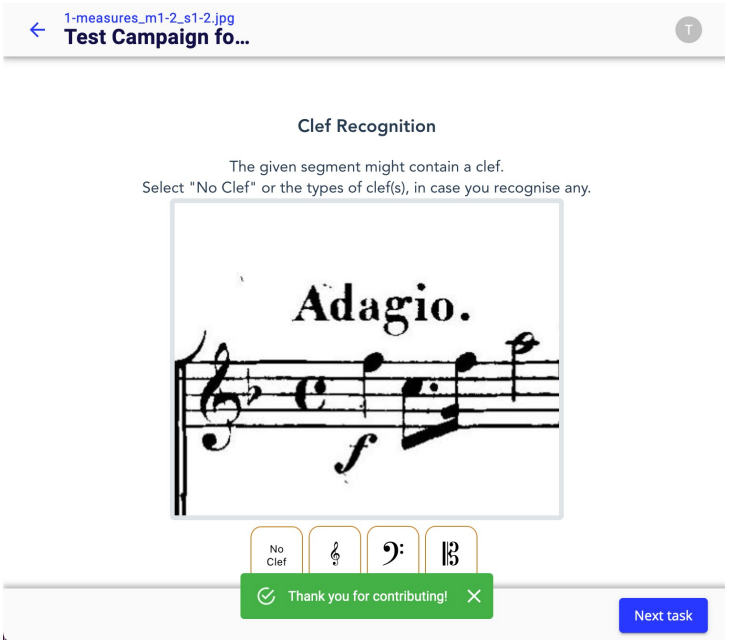

Figure 2.8. In-app 'Thank you' messages keep Contributors engaged

2.7. Connection to requirements

The functional requirements for the pilot were described in **Deliverable 2.2 - Complete Requirements**[27]. However, we pivoted on this plan, both following the feedback of the RCO participants, as expressed in **Deliverable 6.1 - Final Mock-ups Testing**[28] (underlining their extremely high output standards and declaring the use of such a system as too much overhead for musicians), and the lack of performable scores in MEI format. As mentioned in **Deliverable 6.2 - Planning for the execution of pilots in real life settings**[29], contacts were established with amateur student symphony orchestras in The Netherlands. This allows for the establishment of a larger engaged orchestra community, that is more dependent on the availability of playable digital scores, and that is digitally minded.

The insights from RCO have led us to investigate the needs of other (amateur) orchestras. The realities of an amateur orchestra are totally different from the needs of members of a top-10 global orchestra. Both groups indicated that - given the choice - they would prefer to annotate their parts the old-fashioned way, with pencil, which is in their view faster and more practical. At the same time: both groups underlined their need for good quality, performable scores in the public domain.

In the first delivery of this prototype in M24, the only feasible way to make more scores available in MEI format was by encoding them by hand. This is a very laborious process which can only be performed by a skilled musicologist. Therefore, while the Campaign Manager saw several functional and usability improvements, we also reworked task designs for the Crowd Task Manager, both being explained in Section 2.4.

[27] This deliverable is confidential to the consortium only
[28] This deliverable is confidential to the consortium only
[29] https://trompamusic.eu/deliverables/TR-D6.2-Planning_for_the_Execution_of_Pilots_in_Real_Life_Settings.pdf

To make the Campaign Manager as reusable as possible, it is built as a separate component and can handle task types of any kind. Similarly, the Crowd Task Manager and Scriptoria are separate systems, communicating through an API. As a consequence, for the accommodation of potential other tasks and campaigns in the future, all these systems can be easily adapted without raising too many interdependence issues.

2.8 Design vision for future improvements

Based on feedback within the consortium, and following the user studies performed so far, (**Deliverable 6.8 - Mid-term Evaluation**), we have updated the design vision for future development of the Campaign manager. We include the design vision here for context, and to give an impression of what a full-featured version might look like:

- Home page now with active campaigns[30]
- Create campaign[31]
- Campaign page[32]
- Mobile screens for future app[33]
- Responsive tasks[34]

3. Integration with other TROMPA WPs

This section summarizes the integration of the prototype with the components and technologies developed in other WPs, including WP3 (automatic description), WP4 (crowd-sourcing and human computation technologies) and WP5 (contributor environment and core components).

3.1. Relation with TROMPA WP3

WP3 tasks	Integrated in prototype v1	Integrated in prototype v2
Music description	n/a	n/a
Audio processing	n/a	n/a
Visual analysis of scores	n/a	X
Alignment of musical resources	n/a	n/a
Multimodal cross-linking	n/a	n/a

[30] https://storage.trompamusic.eu/design/Home_page_now_with_active_campaigns.pdf
[31] https://storage.trompamusic.eu/design/Create_campaign.pdf
[32] https://storage.trompamusic.eu/design/Campaign_page.pdf
[33] https://storage.trompamusic.eu/design/Mobile_screens_for_future_app.pdf
[34] https://storage.trompamusic.eu/design/Responsive_collaboration_tasks.pdf

3.2.1 Visual analysis of scores

Upon receiving a PDF, the crowd task manager will run the measure detection methods described in **Deliverable 3.4 - Visual Analysis of Scanned Scores**. With this information, we both can establish a MEI skeleton, and have the individual image crops that should be used as references in the crowd tasks.

3.2. Relation with TROMPA WP4

WP4 tasks	Integrated in prototype v1	Integrated in prototype v2
Crowd-powered improvement	X	X
Annotators	n/a	X
Incentivisation of TROMPA crowds	X	X
Campaign design	X	X

3.2.1. Crowd-powered improvement

The entire objective of this Prototype is to gain a better understanding of Crowd-powered improvements. That is why the prototype was built with the ability to address a large group of diverse people in mind. Currently served tasks do not require extensive musical expertise, and are simple enough to be conducted by laypeople.

3.2.2. Annotators

This prototype is built making use of and conforming to **Deliverable 4.2 - Annotator Properties and Metrics**[35]. In this first version of the prototype, we targeted users whose skills are known (e.g. orchestra members) thus their Competence Model is assumed. However, for data protection considerations, it was decided to not register user-related information on the Contributor Environment. Therefore, at present, it is not possible to classify an unknown crowd according to the Competence Model, and therefore we only serve simple tasks.

3.2.3. Incentivisation of TROMPA crowds

We have implemented several recommendations made in the 2nd version of **Deliverable 4.3 - Crowd Incentivisation Mechanisms**[36]. As mentioned under Section 2.6, these mostly are targeting intermediate updates on made progress. Generally, in prioritising tasks, we strive to always have an as-coherent-and-complete-as-possible MEI file. At the meta-level, by having campaigns being initiated by orchestras, the members and audiences of these orchestras should form crowds that intrinsically would be motivated to help the orchestra.

[35] This deliverable is confidential to the consortium only
[36] This deliverable is confidential to the consortium only

3.2.4. Campaign design

We have experimented with several campaign designs, ultimately going for designs that are easily understandable, and that uphold the microtask crowdsourcing paradigm (tasks that are small, and can quickly be performed in batches) as much as possible.

3.3. Relation with TROMPA WP5

WP5 components	Integrated in prototype v1	Integrated in prototype v2
Score edition component	n/a	X
Processing library	n/a	n/a
Multimodal integration	X	X
Performance assessment	n/a	n/a
Annotation tools	n/a	X

Generally, as illustrated in Figure 2.2 and summarised in Section 2.5, the prototype relies on the ControlActions mechanism of the TROMPA Contributor Environment for the communication between the Campaign Manager, the Crowd Task Manager and Scriptoria.

3.3.1. Score edition component

The score edition component (see **Deliverable 5.2 - Score Edition Component**) is used to render the current MEI file as a preview in the campaign overview.

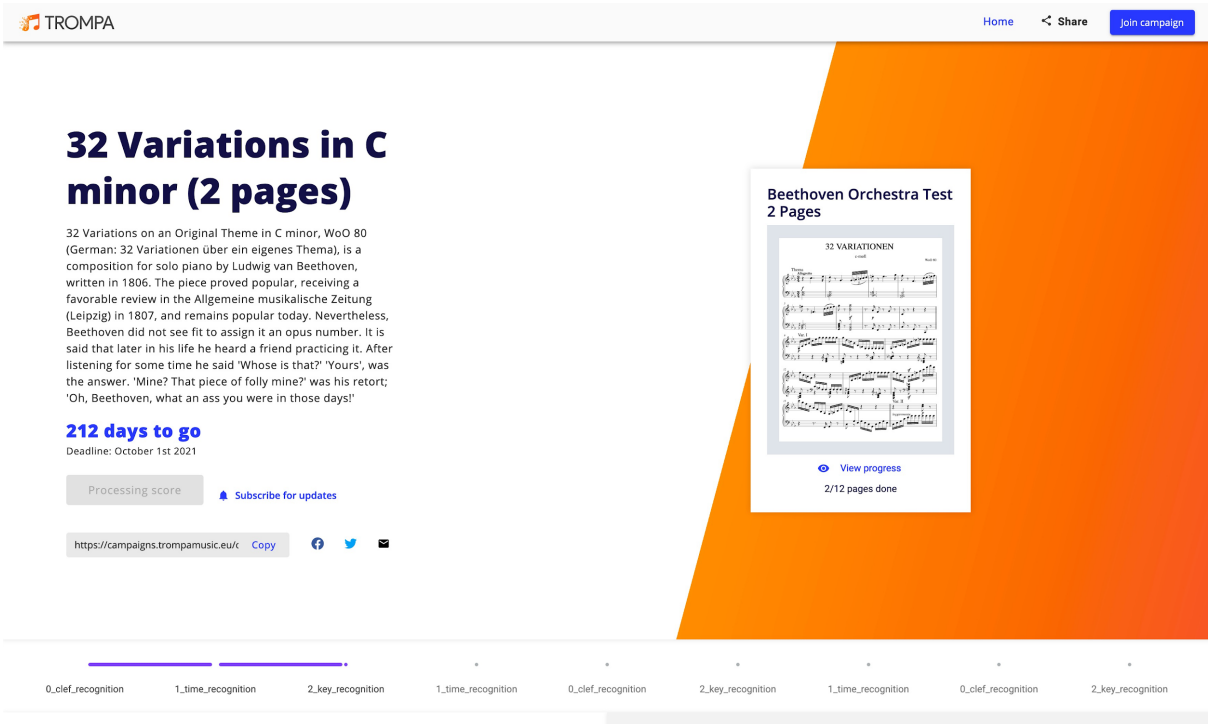

Figure 3.1. The Score Edition Component is used to render a preview of the campaign

3.3.2. Multimodal integration

When a user initiates a campaign, the multimodal component (see **Deliverable 5.1 - Data Infrastructure**) is used to make sure the campaign PDF score file will be associated with available score metadata in the Contributor Environment.

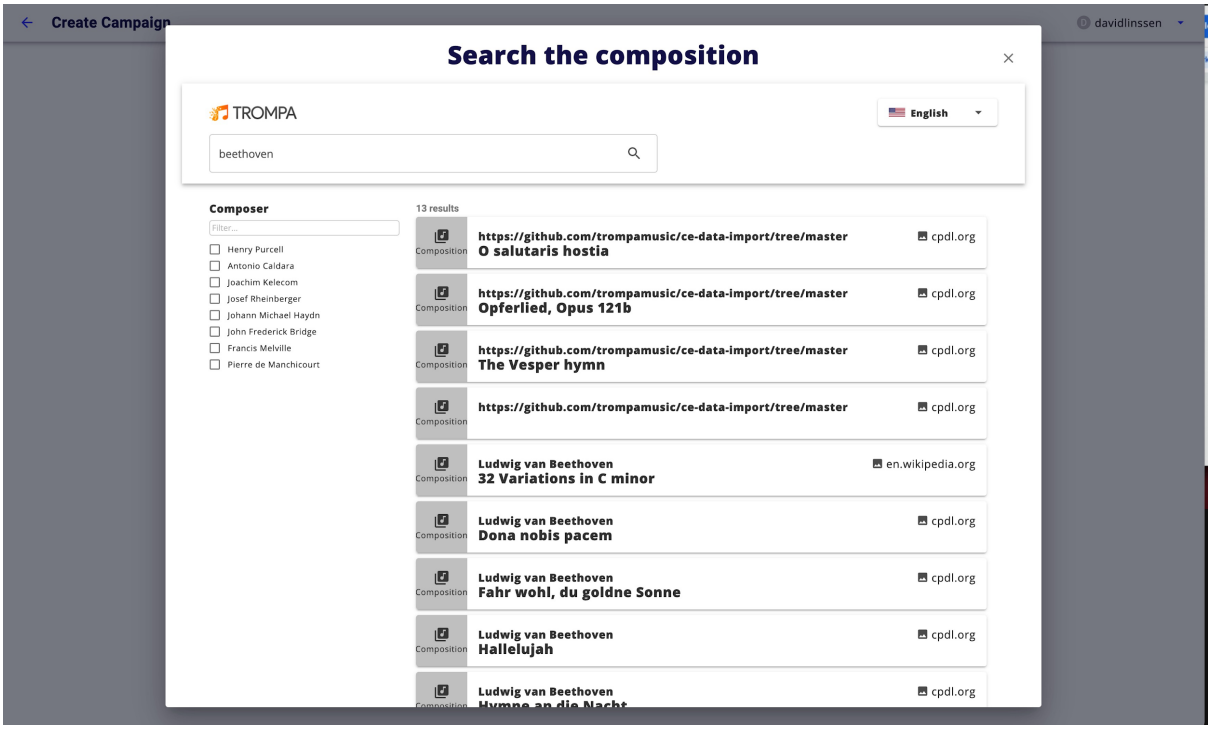

Figure 3.2. Associating available metadata in the CE through the Multi-modal component

3.3.3. Annotation tools

The Annotation Tools (see **Deliverable 5.5 - Annotation Tools**[37]) are currently not integrated, as the current crowd-assisted OMR tasks do not involve annotation. However, the system architecture allows for the tools to easily and logically be incorporated when needed.

[37] https://trompamusic.eu/deliverables/TR-D5.5-Annotation_Tools_v2.pdf

4. User evaluations

4.1. Initial target audience & recruitment strategies

Upon pivoting to student orchestras, the initial plan had been to heavily collaborate with student orchestra Krashna Musika, starting from Mid-April 2020. More specifically, we initially had intended to work with Krashna on moving towards MEI digitisation of Mahler's first symphony, which the orchestra was planning to perform on a concert tour.

4.2. Impact of COVID-19 crisis

Due to COVID-19 crisis, the Krashna Musika has not been able to rehearse, nor to perform. Some members have returned to their home countries; for others, continuing their studies during the quarantine period has proven challenging. While several orchestra members still indicated willingness to participate, there was no project to collaboratively push for. With no large physical rehearsals, which normally would be the place for announcing projects, the orchestra also had far less means to enthuse fellow players. Finally, we noticed that many people generally seemed less motivated throughout the COVID-19 crisis, making response times of possible participants slower, and also causing several no-shows or no-replies.

4.3. Adjusted target audience & recruitment strategies

To maximize the amount of potential participants, while acknowledging the insecurity and distraction of the COVID-19 crisis, we decided to write all student orchestras in The Netherlands, as well as the Nationaal Jeugd Orkest (NJO), an orchestra for young professionals at conservatoires. With the help of the academy coordinator of the RCO, we contacted them, invited them for several usability studies, and offered concrete rewards to participants associated with the RCO brand (the choice between a membership to Entrée (the RCO's youth audience association), or an RCO CD). More details on this have been given in **Deliverable 6.8 - Mid-term evaluation**. Usability studies then were planned as follows:

- First usability study with members of Krashna Musika in August 2020, informing the change from transcription & validation tasks to simpler crowd tasks and gauging support for MEI writing. Results of this study are reported as part of **Deliverable 6.8 - Mid-term evaluation**.
- Second usability study with 30 orchestra members of various orchestras, testing simpler crowd tasks and gauging support for campaigns. Results of this study are reported as part of **Deliverable 6.8 - Mid-term evaluation**.
- Third usability study with 5 orchestra members of various orchestras in February 2021, testing a live campaign, and assessing usability of transcription-related crowd tasks. Results of this study will be reported as part of **Deliverable 6.9 - Final Evaluation**.
- Final evaluation studies in March 2021 of campaigns, their usability, and support for their adoption in the practice of orchestras; target is several dozens of orchestra members. Results of this study will be reported as part of **Deliverable 6.9 - Final Evaluation**.

4.4. Insights taken along for prototype iterations

User feedback has been indispensable in the development of this prototype. Following the feedback in **Deliverable 6.1 - Final Mock-ups Testing**[38] from RCO members, we pivoted the focus of the orchestra use case to campaigns aimed at semi-professional youth and student orchestras. **Deliverable 6.8 - Mid-term evaluation** gives extensive reports of the usability studies conducted in Fall 2020; these triggered many of the changes reported under Section 2.7, most notably mobile-friendliness, the reduction of tasks to simple crowd tasks (no MEI writing), the removal of the validation task (had understandability issues, while at the same time not being necessary when quality assurance is performed by checking for similar input by multiple contributors), and paginated prioritisation, such that crowd input is gathered in a way that is as coherent as possible.

4.5. Points of attention for final evaluation

One major point of attention for the final evaluation will be to assess the realism of running campaigns in times of COVID-19. As no large projects are currently conducted, and our audiences seemed not too motivated to spend a lot of extra online time, it appears hard to have a real orchestra initiating a campaign, to incentivise a large audience to contribute, and to have this audience conducting tasks for a sustained period while being 'on the go'. For our third user study, despite several mails and pre-prepared messages for distribution over chatting apps, we also noticed a surprisingly low response of participants in the end (while we wrote multiple orchestras and offered flexibility in scheduling, only 5 participants from 2 orchestras ultimately showed up), which may have to do with hardship associated with a prolonged lockdown.

As a consequence, for the final evaluation, we will run a single campaign, and will offer multiple options (10 evenings in March 2021), in which we will host online sessions during which interested orchestra players can join and engage with the campaign.

[38] This deliverable is confidential to the consortium only

5. Conclusion

We presented the final version of the Orchestra Prototype, geared towards the collective creation and improvement of public-domain musical scores. It combines a collaborative campaign mechanism with the OMR capabilities of the Hybrid Annotation Workflows (WP4), partially powered by content analysis techniques from WP3. To realize the campaign mechanism, an architecture was chosen that can flexibly be adapted and that respects the Contributor Environment. Because of this, in the back, multiple systems are actually involved (Campaign Manager, Crowd Task Manager, Scriptoria).

As the other use cases, the orchestra use case has been facing unexpected challenges due to the COVID-19 crisis. Following the audience pivot after the first project year, from professional orchestras to semi-professional and amateur (youth) orchestras, this target audience has been heavily affected by the crisis. The social structure and community motivations that normally drive their music-making activities (which would intrinsically make them more open to crowd campaigns) has fallen away in times of crisis. We have adjusted our outreach and evaluation strategies accordingly.

With this document, we have illustrated the project progress at Milestone 4 (Evaluation results of Working prototypes v1.0, Working prototypes v2.0 ready). Final evaluation outcomes, including attention to usability and practical support in crisis times, will be reported in **Deliverable 6.9 - Final Evaluation**.

6. References

6.1. List of abbreviations

Abbreviation	Description
CE	Contributor Environment
MEI	Music Encoding Initiative (file format)
OMR	Optical Music Recognition

6.2. List of partner acronyms

Abbreviation	Description
TUD	Delft University of Technology
RCO	Concertgebouworkest
VD	Videodock