Standardization and flexibility in a shared component platform

Design principles for developing a shared component platform as a boundary resource

Simeon Andersen Tverdal

Thesis submitted for the degree of
Master of Science in Informatics
60 credits

Institute for Informatics
Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

June 2021

Standardization and flexibility in a shared component platform

Design principles for a shared component platform as a boundary resource

Simeon Andersen Tverdal

2021

2021

Tittel

Simeon Andersen Tverdal

http://www.duo.uio.no/

Trykk: Reprosentralen, Universitetet i Oslo

Abstract

For generic software such as a health information system to be effective it must be properly adapted to its local environment. This localization often requires that applications be remade from scratch, something that puts an increased burden on the development team responsible for localization. Despite the often-strict time constraints, developing new applications can be slow and tedious. One solution could be for local developers to reuse application components made for similar purposes by other localization teams to avoid redundancy. For this purpose, a platform for sharing components between different members and groups within the ecosystem could be imagined needed, but there is little knowledge on how such a platform should be designed and positioned. There is lacking research on component sharing platforms that allow for users to contribute to the platforms content, and this unique feature of the platform calls for knowledge on how it needs to be managed. To attempt to uncover principles on how to position such a platform in the larger ecosystem this thesis asks the question: "*How to balance standardization and flexibility for a shared component platform as a boundary resource in a diverse ecosystem".* The thesis attempts to answer this question by means of design science research, and a platform prototype is created as part of the process as a tool to gather information. The result of the project is five design principles that apply to creating a component reuse platform in such a context.

Keywords: *software reuse, generic software, DHIS2, platforms*

Acknowledgements

Special thanks to my supervisor Johan Ivar Sæbø and my co-supervisor Magnus Li for their help and guidance through the project. I must also give a warm thanks to the Design Lab and its members, especially my group members Håkon Heskja and Anastasia Bengtsson. Finally, I want to thank HISP Tanzania, HISP Mozambique and the DHIS2 core development team for participating in this project.

Simeon Andersen Tverdal

University of Oslo

2021

Table of Contents

Table of Figures

1 Introduction

1.1 Motivation

A generic system such as a health system is tasked with providing health care services to the target population, and to do so in an efficient manner. The scope in which these systems operate is large, and for the systems to function as intended and provide sufficient healthcare to the target population, as well as being able to react to changes such as the outbreak of a new disease or a pandemic, a substantial amount of information must be gathered, stored and analysed for use by the appropriate parties.

The requirements to quality contrasted with the methods chosen to gather this data present many challenges to the health management information system, many of which persist in even the most sophisticated systems in the world. A large issue is fragmentation in storage and use of health information, where information is not shared between systems and sectors that require the same information, leading to inefficient use and analysis of data(Lippeveld et al., 2000) The open-source information system DHIS2 seeks to remedy this problem by presenting a platform for data storage and analysis that contains a diverse set of applications for data gathering, thus removing the need for multiple systems. DHIS2 is currently seeing widespread use across the world, with several groups of local developers complementing the existing application base with their own self-made tools. These local developer groups are called HISP groups and are responsible for localizing the generic DHIS2 platform to fit their local communities. Together they make up the brunt of what can be referred to as the ‘global DHIS2 community’.

While the applications made by these groups are tailored to their specific case, due to their nature as part of a health software system, the applications are often made up of similar parts. These parts, for instance a search bar, an algorithm relevant to sorting a patient registry or an organization unit tree can often be found to exist in a similar form in multiple applications spanning several local DHIS2 instantiation. We can call these parts of application for components, and for the purposes of this thesis we define components as simply the building block of an application, which may vary in size. Examples can be a search form, or a map widget. In the terms of atomic design, we can consider components to be molecules and organisms, as opposed to the smaller atoms(“Atomic Design | Brad Frost,” n.d.). Instead of developing these components from scratch every time or basing the new application on a previously developed generic app, a scenario where these modular components are shared across different HISP groups can be imagined. This increase in cooperation could reduce the duplicate work caused by a lack of component and application sharing. Several of the groups have also already created their own local repositories where they store components for future use(“HISP Tanzania,” n.d.). This thesis is based on the hypothesis that there is a lot to be gained by sharing these components across local developer groups and will attempt to aid this global cooperation by creating a component sharing platform where the different developer groups can upload components for others to use. By allowing developers to search for already existing components, the intended benefit is a decrease in the difficulty of developing an application, increasing the speed and efficiency with which these apps are developed, and potentially serve as a driver for a more standardized ecosystem by means of streamlining the use of encouraged technologies.

1.2 Research question

Due to the diversity inherent in the global dhis2 community(“HISP Network,” n.d.) and as such also the HISP nodes that represent the intended target for such a platform, there are a plethora of challenges that are included in the general goal of creating a usable and efficient way of sharing components. As it will be used by developers with varying degrees of experience and diverse backgrounds some standardization must be implemented to make the platform usable for all relevant parties, as well as to make sure the components hosted on the platform are possible to utilize by all actors. Similarly, some degree of flexibility must be inherent to ensure that these same parties are willing to make use of it, as too much standardization might cause a scenario where the platform is virtually unusable for certain HISP groups.

The research question posed is then ‘*How to balance standardization and flexibility for a shared component platform as a boundary resource in a diverse ecosystem*.’

To address this research question, this thesis attempts to explore the design of a shared component platform, with a focus on how it can contribute to its ecosystem by means of standardization or the lack thereof. The goal was to discover prescriptive knowledge that can be used when designing similar platforms or repositories in the future, and I attempt to accomplish this by means of Design Science Research (DSR). DSR is a methodology that is centred around the design and evaluation of an artifact, and over the course of the thesis, I and a group of two other master students designed and developed a shared component platform and used it as a tool to gather knowledge about this design process.

1.3 Thesis structure

Chapter 2: Context

In this chapter the context to the project is introduced. The relevant parties that we cooperated with during the project as well as the motivation for it is covered.

Chapter 3: Related literature

Here the literature that provides a basis for the thesis is presented, and theories and models that will be used to frame the research are introduced.

Chapter 4: Methodology

The epistemology, methodology and methods utilized during the project will be introduced alongside the process of working on the project. The focus will be on the data gathering activities performed.

Chapter 5: Artifact presentation

Here the final artifact will be presented and explained. The process of designing and developing it will also be presented here instead of in the methods chapter.

Chapter 6: Findings

Here the findings from the evaluation will be presented. Some supplementary information from various other data gathering activities performed at other stages of the project will also make their appearance here.

Chapter 7: Discussion

The findings will be discussed against the research question and the related literature. Reflections on the project and the future development of the tool will also be included in this chapter.

Chapter 8: Conclusion

The thesis is summarized in this section.

2 Context

In this chapter the context surrounding the project will be presented. In addition to introducing the general context of Health Management Information System situated in a low-resource environment, a general understanding of the HISP network and DHIS2 is given, and some of the problems introduced in the introduction is elaborated on or further supported.

2.1 HMIS Background

While affluent countries have to a large degree digitized their health information systems, though often opting for suboptimal fragmented versions, this is not the case for the countries in which most of the earth's population find themselves in. Development of these health systems are ongoing, and in particular, software for data collection and data analysis in the health sector has the potential to greatly improve the quality and amount of data available (Manya et al., 2012). Data quality in particular is a necessity for decision makers to make evidence based decisions, as it concerns itself with the completeness and correctness of data, something sorely lacking under most third world health systems due to the inherent difficulties with a physical paper based system(Mate et al., 2009). A good health information system is a required step to provide all necessary parties with the information needed to provide the best health care services with the limited resources available to them.

Though the reliance upon non-digital tools for information gathering is a detriment to the regions where it is prevalent, a simple upgrade to digital replacements will not necessarily solve the challenges inherent in the system. A recurring problem is that initiatives that aim at improving the health information systems in these regions are separately funded, and do not coexist with one another(Sæbø et al., 2011; Stansfield et al., 2008) This further increases the workload on the health worker responsible for recording data, as they often must collect duplicate data and send it separately to these initiatives(Garrib et al., 2008). There is also the existence of legal issues when attempting to use data from several independent systems at once, reducing or removing the possibility for cross referencing data between these data collection systems(Anderson, 2007).

2.2 DHIS2

Simply digitizing each of the individual systems does not necessarily improve the functioning of the information system. The result of this would still be a fragmented health information system, something that makes it difficult to use the collected data in an optimal way, both for analysis and accessibility reasons (Lippeveld, 2001). A potentially improved solution is to include all data collection and analysis tools as part of a single initiative(Braa and Sahay, 2012). Experiments with data warehouses that aggregates data from several sources for use by decision-makers have seen success in the past, given that the socio-political climate allows for it(Sæbø et al., 2011). The open source software dhis2, as the world's largest health information system platform, is widely used by well over 60 countries and presents a powerful opportunity to replace a fragmented paper based health information system with a solution that integrates current separate reporting systems into one platform("DHIS2 Application Platform," n.d., p. 2). This allows information to be easily shared between projects for more comprehensive analytics(Sæbø et al., 2011).

2.2.1 Development of DHIS2

DHIS2 development is managed by the Health Information Systems Program at the University of Oslo, with developers located in various countries across the world. In 2012, as a response to an increase in demand, the core software development organization was professionalized by hiring full time developers, architects and project managers. This core team is responsible for the development of the core functionality, like the API, and for gathering user requirements as well as maintaining a regular release cycle("About DHIS2," n.d.). In addition to developing the platform infrastructure, the core team have developed several generic applications for the platform, focusing on including functionality that is widely used across different geographical regions. The development of the DHIS2 core is funded by a consortium of donors, including NORAD and UNICEF, to further promote the evolution of the platform as a Global Public Good, a model that promises non-exclusivity and non-rivalry(Koskinen et al., 2019; Nicholson et al., 2019).Also, due to the widespread adoption of the platform, the WHO has collaborated with the core team on creating standard health apps with recommended indicators and best practice for data use at country level("WHO Packages," n.d.). The flexibility of the platform architecture then allows for more idiosyncratic development by local groups of implementation level designers.

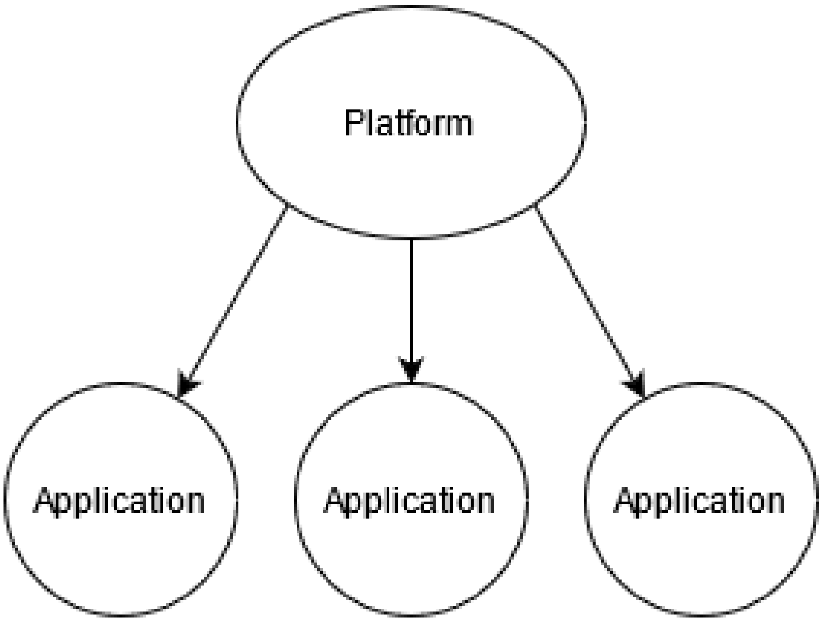

Figure 1: Basic platform structure

Though DHIS2 comes bundled with applications out-of-the-box, the diverging requirements and needs of local users contrast with the generic nature of the software, something which may make it harder to use as a result. Cases of neglect where the forms are filled with falsified or otherwise unusable information can occur when the information software is deemed difficult to use, often due to lack of consistent data gathering, the belief that data gathered isn't used by decision makers or as a result of misunderstanding the digital system(Garrib et al., 2008). In other words, If the system is not tailored to the user, they are often not able utilize the system properly, and the problems with a paper-based health information system will persist, despite the technical advances.

2.3 HISP groups

To combat this critical issue, local developer groups adapt the open-source health management information system to the needs of their local customers by modifying existing applications or developing new ones from scratch upon request. The groups we refer to here are those designated as HISP groups or nodes, that are in active cooperation with the core team("HISP Network," n.d.; Phone and fax, n.d.). In addition to creating tailored applications when required, these groups of local developers may contribute to the platform ecosystem by uploading their creations to the DHIS2 app store, making them available for download by other users of the software.

Still, the need for new specialized functionality exceeds what some of these groups can currently manage, and the quality of the apps is thus reduced with reference to maintainability and usability as the groups are forced to prioritize development speed at the cost of quality. The core developer team behind the health management information platform is aware of this and are constantly working on creating boundary resources that makes the creation of new applications on the platform both faster and easier. The focus of these resources is currently on creating visual components and style guides that will lower the barrier of entry with regards to making new apps from scratch, as well as serving as a method to further standardize the style of the apps hosted on the platform.

The existing resources referenced above, as well as the ones planned or currently in development are not subject to improvement by the community on the platform. This is necessary due to the nature of these boundary resources, as they among other things require a strict adherence to the design principles that govern the development of dhis2. There is however lot of potential in including the community in the work process. To complement these existing resources, we therefore propose a way for the different groups of implementation level developers to share their own self-made components with one another. There currently exists an unfortunate lack of communication between the HISP nodes, something which in part leads to a wasted time and resources. The nodes often share the need for similar or even identical applications, and due to the similarity of many of the requested applications there is a large potential for reuse of old components or parts of old applications. This lack of communication and thus lack of component and application sharing, cause duplicate work to be a somewhat common occurrence. Several of the groups have also already created their own local repositories where they store components for future use(“HISP Tanzania,” n.d.), and might thus interested in taking an active part in the development and testing of a common way to store and share these components

3 Related Literature

In this chapter the relevant fields of literature will be presented, and the theories and concepts required to understand the rest of the thesis will also be introduced.

3.1 Platform Theory

Digital platforms have come to play a vital role in the world economy, with massive companies like Apple, Facebook and Google benefiting greatly from the platform model (Evans and Gawer, 2016). Koskinen et.al (Koskinen et al., 2019) argue that there exist two main types of platform, namely transaction and innovation. While transaction platforms, like the platforms employed by the companies already mentioned, crudely put, focus on facilitating transactions by connecting sellers and consumers, innovation platforms, like DHIS2 and Android, are more concerned with presenting third party users with a basis for developing new applications and features(Koskinen et al., 2019; Nicholson et al., 2019). For the purpose of explaining DHIS2, we will focus on innovation platforms, and on exploring the structure of the relationship between the core platform and the third party developers this model facilitates for (Baldwin and Woodard, 2009; Koskinen et al., 2019).

We can view the more general concept of a platform from two different perspectives, both relevant for the development of the component platform. The more technical view, as defined by Baldwin and Woodard, is as “a set of stable components that supports variety and evolvability in a system by constraining the linkages among the other components” (Baldwin and Woodard, 2009). We view this core as part of what Tiwana et.al calls the ‘platform ecosystem’, which they consider to consist of two major elements, namely the mentioned platform, and its complementary apps (Tiwana, 2013). This can be defined differently as a set of core components with low variety; the platform; and a complementary set of peripheral components with high variety; the applications (Baldwin and Woodard, 2009).

Platforms can also be viewed through a more sociotechnical lens, a perspective that includes the individuals and organizations creating and interacting with the technical components. It has been defined by Tilson et al. as a “socio technical assemblage encompassing the technical elements (of software and hardware) and associated organizational processes and standards" (Tilson et al., 2012).

Innovation platforms focus on the structure of the relationship between the core and the periphery, and the resources being provided(Baldwin and Woodard, 2009; Koskinen et al., 2019). Henfridsson et. al argues that to understand digital platform dynamics, one must focus on the boundary resources rather than the core (Henfridsson and Bygstad, 2013), and we thus view the development of the component platform through the lens of boundary resource theory. Boundary resources can be explained as resources existing at the interface between the platform owner and third-party developers which facilitate the use of core platform functionality to build applications (Ghazawneh and Henfridsson, 2013). More substantially it has been described as "the software tools and regulations that serve as the interface for the arms-length relationship between the platform owner and the application developer (Ghazawneh and Henfridsson, 2013). Ghazawneh & Henfridsson state that to facilitate the growth of the platform ecosystem, the platform owner must create boundary resources instead of applications (Ghazawneh and Henfridsson, 2013). The proper design of boundary resources then, is of critical importance, as the goal is both the maintenance of platform control as well as the transfer of design capability to users (Ghazawneh and Henfridsson, 2013). Proper design of boundary resources will then function as a guide for developers, while also supporting application developers to build and sustain platform innovation (Ghazawneh and Henfridsson, 2013).

3.2 Generic Software

As mentioned in the section above, the core developer team should focus on creating boundary resources that facilitate innovation by third party developers, rather than creating applications themselves. To express this difference, we differentiate between two levels of design, namely generic-level design and implementation-level design (Dittrich et al., 2009; Li, 2019; Li and Nielsen, 2019). We can view this in the context established as generic-level design means creating the boundary resources intended for use by implementation-level design. M. Li introduces the concept of a design infrastructure to compliment this, by focusing on the boundary resources that facilitate design. The design infrastructure is defined in short as "The resources supporting the 'shaping' of the software", and is regarded as a sociotechnical concept (Li, 2019)". This concept can further help explain the difference between generic and implementation level design, by stating, in short, that generic-level design means designing resources that are made part of the design infrastructure, while implementation level design involves utilization of the design infrastructure to build systems according to specific organizational requirements.

Viewing the component platform through this lens, we can define it as a part of the design infrastructure and is mainly concerned with generic-level design. The goal is for the component platform to strengthen the implementation-level design capacity (Li, 2019), which is defined as how well the design infrastructure supports processes of implementation level design. In addition, the components which might be specifically made for an organization, will also feed back into the design infrastructure when becoming part of the component platform. This distinction is thus not absolute.

3.3 ICT4D

The HISP initiative is deeply rooted in third world countries, and the resulting platform will be tested by the developer groups localized within these developing countries. Richard Heeks writes that health information initiatives in developing countries often fail due to what he calls "design-reality gaps" (Heeks, 2006). Any planned change to an organizational structure will include assumptions and requirements in its design. The mismatch between those design expectations and the reality introduced by the context of the actual situation is what constitutes this design-reality gap (Bass and Heeks, 2011). Bass and Heeks further write in their article on curriculum change in Ethiopia that these design reality gaps extend to most aspects of organizational change, but that the dimensions of technology and Staffing and Skills are most affected (Bass and Heeks, 2011). Both dimensions are highly relevant for the development and integration of the platform, and we as developers must therefore be pointedly aware of our own assumptions when designing the tool, or the cultural distance between us and the users will likely present a significant design-reality gap.

3.4 Summary

The position of the shared component platform as a generic platform designed as a boundary resource for DHIS2 and it also allowing for its users to contribute components while still being part of an enterprise ecosystem means that there is lacking information on how to balance the different actors and interests. This thesis will attempt to gather some information on how to position such a platform in such a complex context.

4 Methodology

In this chapter the methodology used during the project will be introduced, and the project process presented. The focus will be on the data gathering activities, and the development process will be covered in a subsection of the chapter presenting the artifact.

4.1 Ontology and Epistemology

The research performed as part of this thesis is done with certain assumptions in mind that could be useful to understand and place the findings made and the data gathered. To use the established framework of 'research paradigms', the thesis could be considered to be a part of the increasing list of IS research considered a part of *Interpretivism.* Interpretivism can be discussed on many levels, but originates from the ontological perspective that reality is a social construct (Abdul Rehman and Alharthi, 2016). Interpretivism rejects the notion that a single, verifiable reality exists independent of our senses, in contrast to the more traditional view of *Positivism* which states that an objective reality that can be discovered exists.

Another reason for placing the thesis within the interpretive paradigm is the research question chosen. The thesis attempts to resolve the research question by means of the subjective opinion given by various HISP members. This fits the interpretive epistemology, which is subjective in nature, and acknowledges that all information is coloured by the prejudice of the observer. By placing the thesis within the interpretive paradigm, we are thus not looking for objective verifiable reality, but rather the view of the research subjects which is deemed as knowledge for the purposes of this thesis.

The reason for adopting this view for this particular project is the fact that we as researchers are entrenched in the subject we are researching (Grix, 2004). As part of the University of Oslo we must be aware of the fact that the HISP groups are likely to view us in a different light than if we were researchers completely separate from the ecosystem. Our view of the ecosystem and the individuals within are also inevitably tinted by the experiences of others as well as our own position as members of the Design lab and attempting to discover objective truth in such a muddled environment would prove difficult. In addition, a consideration to be made is that a component sharing platform cannot be separated from its users. Therefore, a platform is only good and 'usable' if its actual users deem it so. Any objective criteria in terms of design makes little sense in such an environment and gives further encouragement in choosing an interpretive paradigm.

4.2 Design Science Research

Design Science research as a methodology is focused on designing an artifact with the intention to evaluate and extract knowledge from it (Hevner et al., 2004). It is quite similar to what one could describe as a routine design process in software development but differentiates itself from regular development by virtue of extracting knowledge. The methodology is based on the assumption that as Hevner et al. writes "…doing innovative design that results in clear contributions to the knowledge base constitutes research" (Hevner and Chatterjee, 2010, p. 15). Hevner et. Al considers contributions from design science research to exist as part of one of three categories, these being *design artifact*, *design construction knowledge* and *design evaluation knowledge* (Hevner et al., 2004). Contributions in the form of *design artifact* is simply the artifact itself, that should solve the problem it was created for. *Design construction knowledge* constitute knowledge about creating an artifact whereas *design evaluation knowledge* is knowledge about the evaluation of artifacts such as evaluation metrics or methods.

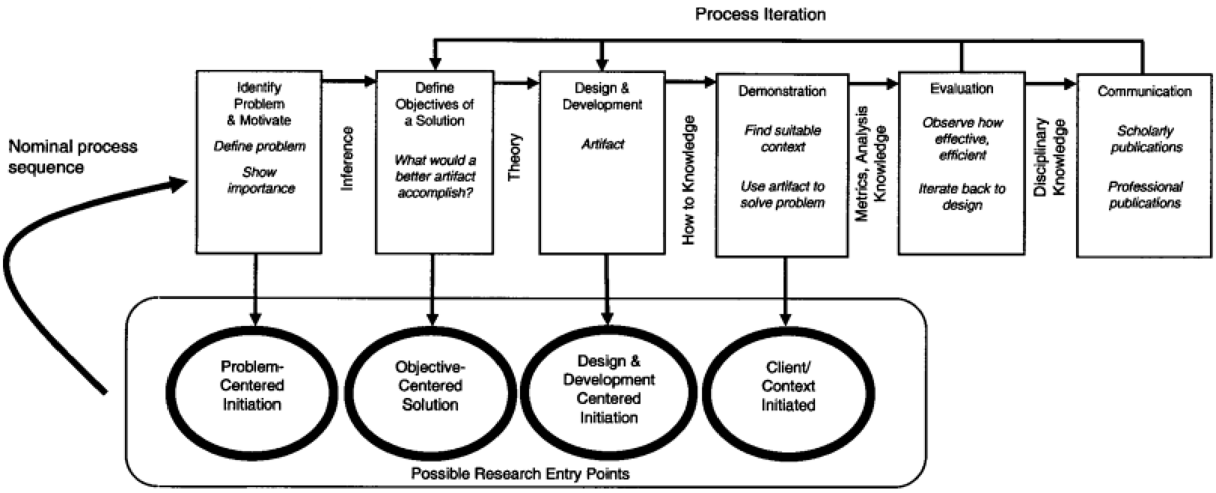

Figure 2: Design Science Research process (Peffers et al., 2007)

One classic interpretation of the general structure of a design science research project is shown in Figure 2. It describes the process from start to finish, including the iteration required, and in addition presents the four different research *entry points*. These entry points describe how a project is initiated, and where in the process it corresponds to. For the purposes of this thesis, I consider the project to be *Problem centred*. I understand it as such because while the goal of the project was to create a shared component platform, the exact objectives of such a platform was not given. We knew from the problem description that it was likely a good way of addressing the problem, but the objectives of the platform, what it should achieve exactly was up to us to define.

While during the start of this project I was not aware that I would be using Design Science Research much of the work can still be placed within this framework, perhaps due to the similarities to common software development. Our process was centred around the development of the artifact, and when examined can be mapped roughly to the model.

First was what we at the time deemed the planning process. This maps to the first two stages, namely *Identify problem and motivate,* and *Define Objectives of the Solution.* The problem, or at least *problem area* was to a degree identified from the start. Several formulations of similar problems could be placed as the reason for creating a shared component library, e.g., slow development, development of redundant applications, low quality applications, lack of standardization in the ecosystem, just to name a few. While the objective of making a shared component platform did not emerge from these problems from our perspective; we were given it after all; being aware of which problems the platform should address was important for further development. The second part of motivation was to motivate the platform itself, or in other words why a shared component platform would address these problems. This stage took the form of literature reviews and internal conversations the group and with supervisors.

The objectives of the solution was defined in a less formal manner than presented here, and over a larger stretch of time as a result. Many internal meetings as well as the first interview with the core team and two HISP groups were used to identify what the platform should be able to do. This processed overlapped with the *Design and development* step, which fits the model as well. Here we followed an agile process of software development, using a Kanban board to track work. When we then had something to show we would show it during a *Demonstration.* I consider the only real demonstration to be the final one that I performed as part of the final *Evaluation,* as this was the only time it was used in front of the evaluators. The final step is *Communication,* which this thesis is responsible for.

The design science research methodology seeks to create ‘what is effective’ in contrast to the more common behavioural-science methodologies which seeks to find ‘what is true’, and this fits our given problem quite well. Another choice would be to perform a case study centred around the completed artifact and consider the development as a separate endeavour, but as I am to be a part of the team creating the artifact, I find design science research to be a more compelling option.

4.3 Process

In this section the overarching process will be presented and explained, with the focus being on the interviews, internal meetings, evaluations and the decisions made as a result. More technical detail about the artifact as well as some more information about the technical development of the platform will be presented in the chapter dedicated to the artifact.

From the very start of the project the problem description was given, in other words we were entering into the project already aware of the practical problem we would be addressing, and to some degree how we would have to address it. To recap, referring to the model created by Peffers et al. shown in Figure 2, we could consider this be a *Problem centred initiation*, since while some direction was given, the exact objective of the project was not given at the start (Peffers et al., 2007).

With the main direction of the project already decided, we were saved the sizable effort of deducing what aspect of the ecosystem we wanted to improve upon, as well as how we want to do it. It is instead the particularities of how this given problem should best be addressed that lies at the crux of the project. To this extent we need to cooperate with the relevant parties of the platform and deciding who these stakeholders would be and what their roles should be was the first real step in the project process.

The IS research group at the University of Oslo has for a long time been concerned with creating software that is designed for the local user while still being generic enough as to be scalable. The HISP groups are one important part of the current solution to this problem, and

they have proven quite effective at adapting generic software to local user needs. The main goal of the artifact can be deemed as allowing the different HISP groups to achieve just this in a more efficient manner, and one of the important aspects of our project was to create a platform that is intuitive and user friendly from their own perspective. It is clear from this that one of the parties that should be included in the design process, and perhaps the partner most relevant during this process, would be the HISP groups, of which we needed to find willing representatives. Another critical party that we needed to cooperate with during the process of designing this artifact is the DHIS2 development team, from here often referred to as the 'core team', that is located at UIO. These developers would play a critical role in both guiding our platform in a direction that suits the DHIS2 vision and were important for our ability to get the respective HISP groups interested in participating in our project. With regards to the potential future of the project they are also critical, as their continued cooperation is not only beneficial for the proper integration of the platform into the DHIS2 ecosystem, but also for the adoption rate, as the HISP groups have shown interest in following along with what the core team is supporting. The core team's support could also play a large role in possibly maintaining the platform after our involvement with creating it was concluded.
The project was centred around the development of an artifact, the shared component platform, and the use of this artifact to gather data that can be analysed. To better develop this artifact under the timeframe given, a decision was made to gather several students to cooperate on the development of the artifact. This group consisted of three master students that are working on individual projects derived from the original problem description mentioned in the introduction. Most of the planning was as most of the initial data gathering done as a group, since most of this data was of import to all members of the group. Only later, during the final stages of the evaluation of the artifact, would the data gathering be more focused around everyone's core research question and thus the group would retreat from the centre stage and take a more indirect supportive role.

To start the process of developing the artifact we first decided upon the stakeholders that should be able to influence the artifact, as well as the degree to which they should be involved, in other words the role they would take in the development of the artifact. Perhaps the most obvious obligatory participant was the main DHIS2 development team, which was deemed critical to our ability to decide on the technical features of the platform, as well as for our ability to create an artifact that would be relevant for the future DHIS2 ecosystem. Their participation would also be critical for our ability to engage with the various HISP groups, providing our work with some more legitimacy that could entice the groups to participate more actively. The core team was also deemed to be important for the potential future of the project, as they have the ability to possibly maintain the platform after our involvement concludes. The core team are matched or perhaps even exceeded in terms of importance to the project by the HISP groups that will be the users of the artifact. It is important to make sure that the artifact is usable for the wide variety of HISP groups that might make use of it, and a close cooperation with some selected representatives is important to make sure that this is the case. The selected groups will play a vital role in providing feedback during several stages of the project, perhaps most notably at the project's end.

Choosing these two main parties, we utilized the network already established by the research group based on previous collaborations. This provided us with opportunities we would likely not have gotten otherwise and while this collaboration was hampered by the restrictions imposed due to covid19 it still represented the main data gathering opportunity for the project. From the core team we contacted several developers with varying degrees of experience working with DHIS2, and while we contacted three HISP groups, we ultimately managed to create a collaborative relationship with two of them, namely HISP Mozambique and HISP Tanzania.

While contacting the core team and the chosen HISP groups we had several internal meetings where the main goal was to adequately prepare for the upcoming meetings with the relevant parties, as well as preliminary discussion about what the platform should look like. During these discussions it became clear that creating a solution to host code snippets and components from scratch would be both too complicated for our level of experience, as well as too time consuming to properly fit into our planned timeline. During this time, we floated several different potential solutions to this problem, though ultimately no final decision would be made without consulting the more technically experienced core team. Considerations about which of the stakeholders should maintain ownership of the platform was also introduced in this stage, something that would prove more important later when important decisions would need to be made.

Though early communication was performed via email, and in the case of the HISP groups, utilizing the design lab as a proxy, as soon as we felt it relevant we tried setting up meetings with all parties. The parties were of importance for different reasons, and as such were relevant at slightly different stages of the process. The core team for instance were viewed as highly relevant especially during the early stages of development as we felt they had to approve of the concept behind the shared component platform, and thus be on board with the project itself. As such we contacted them before we tried setting up meetings with various groups in the DHIS2 ecosystem.

Very roughly, the cooperation with the different groups could be divided into a couple of stages, with some difference between the core team and the HISP groups respectively. First was the communication over mail, where we tried to gauge interest in the project. In this stage the resources of the design lab proved critical, with core members of the lab having invaluable contacts within the HISP ecosystem, as well as with the core team. Once contact was established, we attempted to arrange meetings. As noted above, we needed to know whether our conceptual platform was feasible to implement within the ecosystem, so a meeting with the core team was the first real item on the agenda. This first meeting took place in spring 2020, and included several members of the core team, as well as the lead developer from HISP Mozambique who provided a different perspective to supplement. During this first meeting we presented our existing thoughts on what a shared component platform would entail, in addition to listening to what the stakeholders wanted from this project. As preparation for this meeting we had held several internal meetings beforehand, making sure to cover most of our areas of interests.

4.3.1 Interviews

The main interviews were conducted as semi structured interviews, where we had created an interview guide with questions that we followed as a basis. At any point we could also follow up on a topic with follow up questions or latch on to an anecdote that we found interesting. We also encouraged the interview subjects to do the same, as that would expose what they found interesting about the different topics. For main interviews we held with the core team and the two HISP groups that we collaborated with, we also had a presentation of the shared component platform to introduce or remind them of what the intended goal of the project was.

These interviews were held during the summer and autumn of 2020. The topic was more on their general app development processes, with additional focus on topics of relevance to the development of the platform. The meetings were conducted separately, with the interview with Mozambique occurring first, followed by Tanzania sometime later. From the side of Tanzania, we had two participants including one lead developer and from Mozambique we interviewed just the lead developer. Worthy of special note are the additional visitors to each of the meetings that were not related to either the HISP groups or our group. During the interview with HISP Mozambique we were supported by Magnus Li who listened in to the meeting, and during the interview with Tanzania we had a core team member participating in the call as well as a fellow master student that was working on an adjacent project. Findings from these meetings are covered in the *Findings* chapter alongside the findings from the final evaluation.

In addition to what could be called the ‘main’ interviews, we had several minor meetings. Some of these were simple introductions, informal conversations where we discussed their interest in collaborating with us on the project. One was held with both Mozambique and Tanzania, though neither took the form of an interview but rather as a conversation. During these meetings we briefly introduced ourselves and the project, as well as what they would be expected to do as part of the collaboration. Both HISP Mozambique and HISP Tanzania expressed interest in further collaborating with our group during these meetings, and future meetings were planned. As a result of these meetings, we also set up additional means for communication with the groups, setting up a shared slack channel that we could use as an alternative to email. This channel was unfortunately rarely utilized, and email remained the main form of communication.

Of note is that one additional meeting was held with another HISP group, whereas with the other HISP groups we pitched our project and attempted to gauge their interest in it. During the meeting they seemed positive to continue collaborating with us, but when we attempted to follow up on email they never responded. As such our collaboration with us ended there.

4.3.2 Evaluation

At the end of the project and with the prototype in its final state the final piece of the puzzle was the final evaluation. This was done separately in contrast to all the previous interviews and meetings, due to different areas of interest. The evaluation was performed as a demonstration performed for HISP Tanzania, and as an outcome five design principles were formed. The evaluation and the detailed findings are covered in a separate chapter, together with more information on this style of evaluation.

4.4 Reflections

As mentioned, the chosen methodology for the project is DSR, though this was not always the case, especially not in the beginning. The problem description as well as preliminary discussion with other members of the team made AR the initial methodology of the project, following in the footsteps of several other similar studies performed by other members residing within the research group, included among being some supervisors and teachers.

The AR methodology is well known within the research group, as well as within the faculty in general and this familiarity combines well with the already existing infrastructure that is well suited for such projects. Action research centres itself around performing an action within an organization, and then observing organizational change as a result of this action (Baskerville and Wood-Harper, 2021; Checkland and Holwell, 1998). This fits well with the original project plan, though admittedly challenging with regards to the perspective of time as an organizational change is the goal of the shared component platform. The platform seeks to impact the user’s development practices by allowing them to easily cooperate and utilize code components created by other users, thus shaving critical time off their development process. Observing this change in the target organizations would be the optimal outcome of the thesis but would require both the development of the artifact as well as the successful integration of said artifact into the organization. According to this original plan, one of the main parts of the project would be the participation within the organizations that represent the target demographic for our planned artifact; the HISP groups that the research group already have a well-established relationship with.

Due to unforeseen circumstances, namely covid19, this direct participation became unfeasible due to travel restrictions as well as disease transmission concerns, and as such a large part of what constitutes the ‘action’ in AR was no longer within the realm of possibilities. This lack of local participation also hinders us as researchers in forming a close relationship of cooperation with the local community, further impeding collaboration efforts. These issues compound an already particularly challenging aspect of performing AR within the limits of a master thesis, namely time. Performing an action within a local organization is done with the intention of creating and observing some degree of change within the organization as a result of our action. In our case, that of creating an artifact to resolve a problem; this would imply creating an artifact and then introducing said artifact into the organization to observe how its use would affect their development process. Not only would this require close local cooperation, but even more importantly, it would require a significant amount of time to observe the desired results; or the absence of these results should the introduction of the artifact be deemed a failure. This is already a challenging prospect considering the time limit of a master project, and due to the exasperating effect of covid19 as mentioned above, Action Research was thus deemed unfeasible.

Thus, for a significant part of the project, including large parts of the development process as well as the data gathering process, I had no actual methodology to properly follow, something shared by my development group. As such, there was also no real framework to guide my work, and this was reflected in the earlier parts of the project with the planning being somewhat more erratic than what would be optimal. Instead of following a framework at that time, the artifact development team, myself included, continued to develop the project in a fashion that felt natural according to the resources we had available to us. It was only later in the project when we were more thoroughly introduced to Design Science Research as a valid option due to their similarities that we realized that our existing work could quite easily be placed within the frameworks existing in DSR, though loosely.

4.5 Research Ethics

The participants signed the NSD consent forms presented in Appendix 3, but care was still taken to avoid any personal information to be revealed as part of the thesis. No names of the participants are mentioned, and care is taken to avoid the mosaic effect. Some interviews were transcribed and stored on google drive but no information that could be considered sensitive was ever written down. I also replaced all names during the transcriptions with codes.

5 Artifact

In this chapter the artifact is presented and explained, in addition to embellishing on the process of designing it elaborated on in the methods chapter. Some early versions and thoughts are presented, showing how the result came to be, aiding the reader in understanding the finalized artifact. Finally, some feedback to the artifacts visual design is presented and contrasted to the existing version of the tool.

5.1 Artifact presentation

The artifact; from here often referred to as the 'shared component platform' , was created in order to address the practical problem presented in the introduction, namely the fact that 'new applications take too much time to develop', and 'duplicate applications are being created in the HISP community'. One method of speeding up the development process and reducing redundant work in this context; that of the DHIS2 ecosystem; is to create a method for the different groups participating in the ecosystem to make use of resources and application components created by others. The artifact seeks to facilitate this cooperation by presenting the users with a digital component sharing platform where the different groups as well as other interested individuals can present their components as well as view and download the components of others.

5.1.1 Overview

The shared component platform consists of three main modules and some standards, all critical for it to effectively address the practical problem at hand. The most visible of which is the *website*, responsible for presenting the components to the user, the *CLI* responsible for streamlining and facilitating the upload of one's own components to the platform, as well as the verification system that is made up of an external git repository hosting a *list of verified components* and utilizing a verification pipeline. In addition, there is the set of standards that to some degree ensures the usefulness of the components presented by the platform.

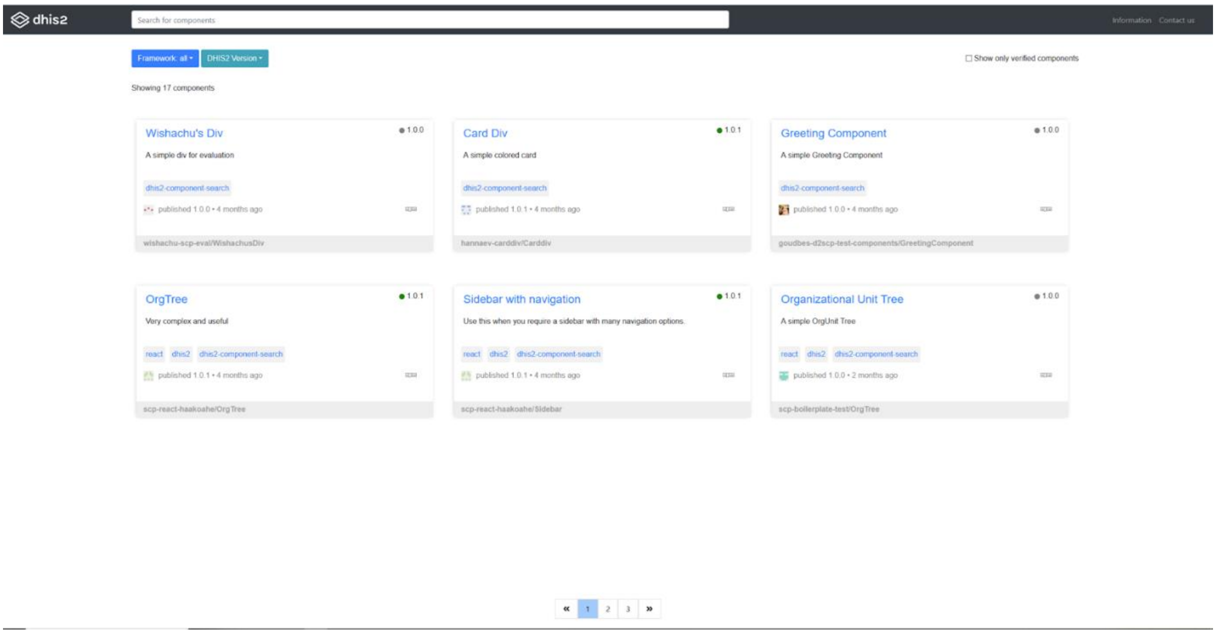

Figure 3: SCP front page (“DHIS2 Component Search,” n.d.)

The interface of the website is presented in the image above and is what most new users would be presented with. The focus of the shared component platform are the components created by users, and as such that is what the visitor is presented with when opening the website. Components are listed in the centre of the page and clicking on them leads to their respective npm page from where they can be further examined and downloaded. To find a specific component the user can search for it in the search tab, and if the user wishes to do so they can browse the components using several tags, allowing them to exclude some irrelevant components. These tags are simple, allowing the user to search for specific frameworks, DHIS2 version, as well as only searching for ‘verified’ packages.

5.2 Development Process

The development started with the overarching goal of the project already given. The project was about “*designing, developing and evaluating a platform that facilitates sharing of reusable web-based components*” [Addendum 2]. This meant that we did not have to start with identifying how we wanted to contribute to the ecosystem and could start planning how to best create this platform. The start of the project was therefore focused on internal meetings within our group of three students. During these early meetings we discussed what functionality we wanted from the platform, as well as what technology we wanted to create it with. It became clear quite quickly that we wanted to avoid hosting components in our own database, as this would add severe amounts of backend programming not relevant for all of our research questions, and due to the fact that solid solutions already exist for component hosting, some of them being NPM and GitHub, NPM later being chosen as the solution we went with for component hosting ("npm," n.d.). NPM is popular among developers all across the world and is in use by several of the developers in some HISP groups ("@iapps - npm search," n.d.).

Another topic of interest during these early meetings was the nature of what we wanted to allow on the platform. Components are a quite loose term and can be interpreted differently by different people. No conclusion was reached at this stage of the process, but we wanted it to be true to its name of a 'component sharing platform' and not an application store as such a solution already exists within the ecosystem. As such one of the early decisions made was to exclude fully developed applications from being presented on the platform. Referencing Atomic design, we wanted to focus on '*atoms*' but also allow *'molecules*' on our platform, in contrast to full 'organisms'.('Atomic Design by Brad Frost' n.d.). This entails that small component, such as buttons, organization unit trees, search bar would be some of the intended components for the platform.

The final focus of these early talks was about who should take on the role of *Product Owner*. We discussed options ranging from the platform being owned by the core team, to the HISP groups, to us being the owners of the platform. Some shared ownership models were also discussed. The plan was to know this for certain before we started to develop in earnest, so as to avoid clashing between different parties, and to allow for streamlined communication. No decision was reached at this stage, but the consensus was closer towards the platform being owned by the core team, something that affected our view on the standardization of the platform, with us expecting to focus on React based components. Later, as we realized how impacted we became by delays in communication, we decided on ourselves as product owners to avoid 'congestion' caused by slow communication.

After the first meeting with the core team in spring 2020, further detailed in the methodology chapter, these thoughts and our conceptual version of the platform were presented to the core team and a HISP lead developer. During this meeting we made sure that both parties were interested in cooperating with us during this project, and we gained more clarity of what the platform would look like. At this point the overall image of what the platform should look like began to form, with several details already being considered. Some of these details being the creation of a namespace or scope where all packages would be gathered (“About scopes | npm Docs,” n.d.). We also wanted to integrate the platform with the DHIS2 app platform to further streamline the creation of applications (“DHIS2 Application Platform,” n.d.). We knew we wanted to provide the user with the necessary tools, framework and standards for users to publish their components, but the details of this were not yet determined. Some thoughts that were brought up during the meeting was creating a bootstrapper or boilerplate that included among other things a basic CI/CD for publishing and updating, semantic versioning and release that would automate versioning, and visual guidelines and conventions that would standardize the components on the platform.

After the meeting, the plan was to create an NPM namespace where all components could be listed, as well as a website that would at first list components that had gained a “seal of approval” that could be recommended to other developers; this evolving later to the concept of verification in the later and final versions. The plan further was then to improve the website to the gallery like version of itself that would show examples and guides for each component.

Development of the artifact started in earnest during fall 2020, with the group adopting an agile model of development with no formal leader chosen. All members of the group took on some tasks and we had meetings once a week to update each other on what was done and what should be done next. To keep track of the work done we utilized a Kanban board using Trello. This allowed us to keep better track of which tasks that were planned, which ones were currently in progress, which ones that were finished.

The development from here went in a similar fashion, with weekly meetings to delegate tasks and discuss changes. The Kanban board was used more strictly as we developed something that helped with progress since everyone had something to do. During this process we were still discussing objectives of the artifact and changed as we developed. This iterative process allowed us to respond to requests discovered during interviews.

5.3 Modules of the Artifact

To quickly recap; the shared component platform can be considered to consist of three modules and a set of standards, all important to achieve the overall goal of the platform. The three components are the *Website* that presents the components to the user, the *CLI* that checks a package of components to ensure it follows the standards set, and the *Whitelist* that composes the backbone of the verification system by listing all verified packages. Of these three the website can be considered to be at the ‘centre’ of the artifact, and the component most of the users will interact with the most.

5.3.1 Website

The website is designed to be as simple as possible while still providing the functionality required to be usable by a diverse range of users. It is designed to look like a gallery in an attempt to present the components to the user as quickly as possible. To minimize the possibilities for confusion it presents the user with a relatively small amount of information to reduce the number of steps required to find the component desired by the user. A conscious decision was made to avoid using the DHIS2 style guide as the platform is not a part of the core of the DHIS2 ecosystem, but rather a community created boundary resource. It was decided that copying the look of official DHIS2 sites and applications might foster misunderstandings about the platform's role in the ecosystem, as well as to how well supported and vetted the site is.

Structure

Upon visiting the website, the user is instantly presented with the component page, the main purpose of which is to show a list of component cards, each representing a component. This page is for most intents and purposes the page where the user will spend all their time, excepting the *Information* and *Contact us* page shown in the upper right-hand corner. The body of the page consists of the component cards and two toggles, one for DHIS2 version, and one for framework, in addition to the *Verified* switch in the upper right-hand corner. Above the components lies the header of the page, consisting of the DHIS2 logo in the upper left-hand corner, the search bar directly to its right and finally the links to the *Information* and *Contact us* pages to the far left. The last component of the page is the footer, which only contains the paginator.

The omission of a front page or *landing page* was done as it was deemed an extra unnecessary step for the user to get to the components. The benefit of such a landing page was also estimated to be minimal, due to the small size of the platform as well as the low complexity of it. In early versions of the platform a landing page was created, but during internal testing we realized that it did not add anything meaningful to the user experience. This means that a central 'hub' from where one could navigate the platform was deemed as pointless, and the potential increase in visual design quality was considered to be too subjective to be of use to the platform.

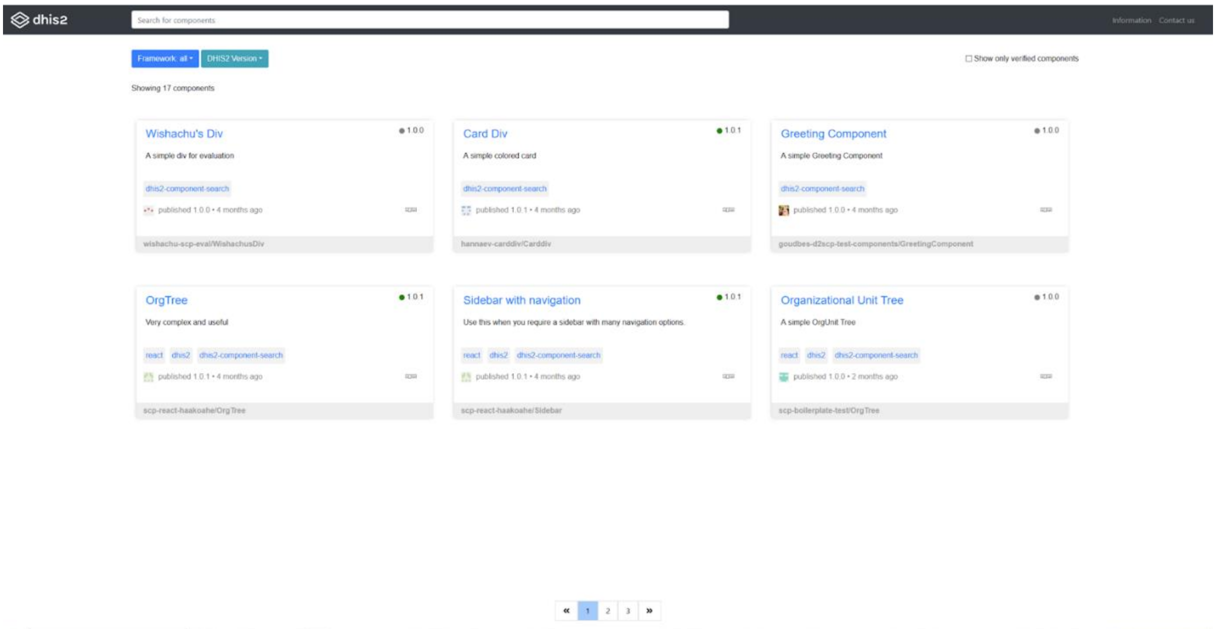

Figure 4: SCP Front page ("DHIS2 Component Search," n.d.)

Component cards

The components are presented using component cards, as seen in Figure 5. The goal of these cards is to concisely represent the different components in a format that is both easily understandable and visually pleasing; being able to take up the majority of the screen without it feeling cluttered. Contained within is the name and description of the component, the version, the package it was extracted from as well as the relevant information from NPM, such as the keywords used, the user that uploaded it and when it was uploaded. Finally, the link to NPM is given for the user to be able to download the component. Much of this information is extracted from the package.json of the package, most of it from a custom attribute that has to be manually added by the developer. This attribute contains among other things the name and description of the component.

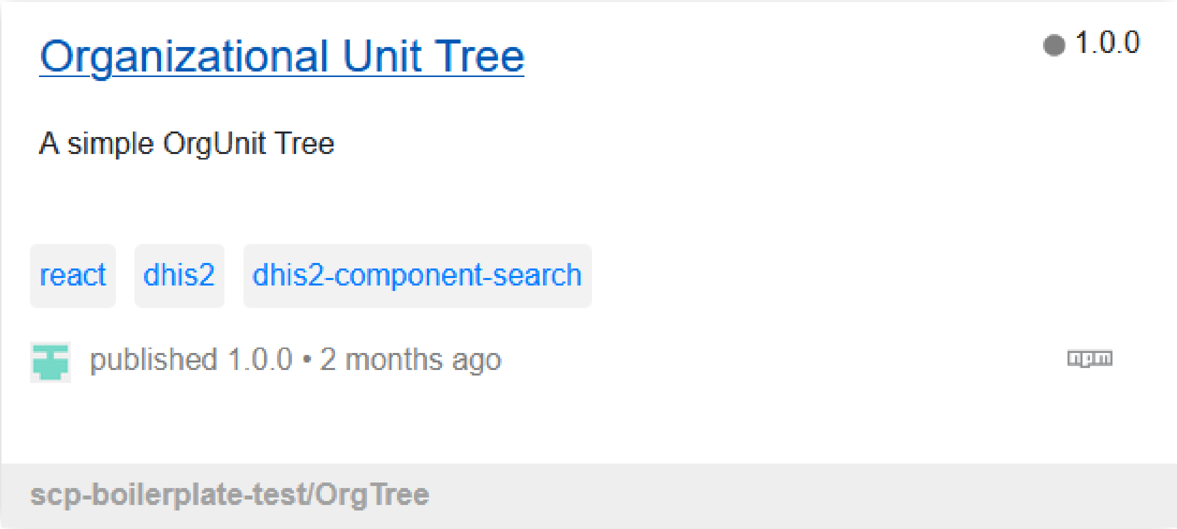

Figure 5: Component card

5.3.2 CLI

The CLI checks that the package is constructed in such a way as to allow the website to display it appropriately, in addition to checking for the presence of supported frameworks (*dhis2designlab/scp-cli*, 2021). While passing the CLI checks is not necessary for uploading a package to the website, it is a requirement for getting the package verified on the platform. This however is not done by the component developer but rather by a maintainer as a part of the verification pipeline. As such, for the component developer the CLI serves as a tool to help ensure that the package is in accordance with the standards imposed by the platform; whether the developer wishes to get it verified or not. In addition, the CLI runs the commands 'eslint' and 'npm audit' automatically, using any config files configured by the user ("ESLint - Pluggable JavaScript linter," n.d.; "npm-audit | npm Docs," n.d.). Neither 'eslint' nor 'npm audit' affects the actual outcome of the verification process in any way, and simply serves as a way of communicating more information to the developer. The commands do not have any built-in configuration files, as while that was included in an earlier version proved nigh impossible to scale up. This means that for these functions to serve a purpose the user should create a configuration file of their own.

The CLI provides a command line interface *dhis2-scp-cli* with two commands, *dhis2-scp-cli verify* and *dhis2-scp-cli pr-verify*. The former is the regular command that checks the package, and the latter is ran during the whitelist pipeline. As mentioned above, this checks the package in addition to running ‘eslint’ and ‘npm audit’. To be more precise it checks the package.json, and makes sure it contains the *dhis2-component-search* keyword, and that it contains the custom attribute *dhis2ComponentSearch* and that this attribute is correct. See the section on standards for more details on the package.json

5.3.3 Whitelist

With a simple verification system that is based around the CLI, the decision was made to create a fairly simple method for authenticating packages as verified. The main component of this system is a GitHub repository containing a list of packages that has been verified (*dhis2designlab/scp-whitelist*, 2021). To request for verification a developer must modify this list and create pull requests. This list is shown in Figure 6, with each item on the list containing two fields, namely the *package_identifier* and the *package_version*. These fields should be on the same line, separated with a comma. The names here are more or less self-explanatory, with the identifier being required for the website to recognize the NPM package in question, and with the version tag added so that the platform allows for the verification of individual versions of a package. This aids in making the verification system a little more useful as well as slightly more secure, by avoiding the scenario where a package is updated and that update automatically being verified. In other words, the tag system attempts to avoid the verification of unchecked code.

package_identifer	package_version
scp-component-test-library	1.0.1
scp-react-haakoahe	1.0.1
scp-component-library	1.0.0
@amcgee/scp-test-package	1.0.0
hannaev-carddiv	1.0.1

Figure 6: Component Whitelist

5.3.4 Standards

While there is a possibility to allow for any type of NPM package to be presented on the site, there is little purpose in allowing most of them on the platform. Some standards that the developers of the components must adhere to should be set to ensure that the platform maintains any relevance for its user base. There is of course the possibility of manually selecting components for display, or otherwise having a manual selection process, but this adds both a lot of labour and uncertainty, in addition to complicating the process further. Thus, while automation is not necessarily a goal in and of itself, the platform benefits from minimizing the number of manual steps involved with adding components, with full automation of a robust selection process being the ideal.

To get a package verified, the package.json must be properly edited. In the future this might be done automatically, but at the current stage it is a manual process. The package must contain the keyword “dhis2-component-search” as shown in Figure 7. This makes the website able to pick it up for presentation.

```
{
  "keywords": [
      "dhis2-component-search"
  ]
}
```

Figure 7: Keyword

The package.json must also contain a link to the github repository in a https format as shown in Figure 8.

```
{
   "repository": {
        "type" : "git",
        "url" : "https://github.com/npm/cli.git"
    }
}
```

Figure 8: Repository

The main addition to the package.json is the custom property “dhis2ComponentSearch”. This field contains the information about the package that the website needs to show the package properly on the website. It must include key/value pairs for framework, and component, with the component field containing key/value pairs for the *name* of the component that will be displayed on the website, the *export* field that must match the name of the exported component in code, and finally the *description* that contains the authors description of the component. Look to the cards in Figure 10 or Figure 11 to see how this is presented on the website. An example 'dhis2ComponentSearch' property can be seen in Figure 9.

```
{
   "dhis2ComponentSearch": {
    "language": "react",
    "components": [
      {
        "name": "Organizational Unit Tree",
        "export": "OrgUnitTree",
        "description": "A simple OrgUnit Tree",
        "dhis2Version": [
          "32.0.0",
          "32.1.0",
          "33.0.0"
        ]
      }
    ]
   }
}
```

Figure 9: 'dhis2ComponentSearch' property

Note the additional property "dhis2Version". This is an optional field that allows the developer to specify which DHIS2 versions the component should work for. In the future this field could be moved one step up; from the component level to the package level, as feedback from the core team has made it clear that a package should not contain components with varying compatibility.

The penultimate item on the list is that the package must be either commonsJS or ES modules to be verified. This means that it should work with 'npm install', and is compatible with the @dhis2/cli-app-script which is the CLI from the core team ("DHIS2 Application Platform," n.d., p. 2).

Finally, we return to the GitHub repository that was included in the package.json. Every package must have a public GitHub repository, and since we verify versions of each package and not just the package itself, a version number should be included. Tags must be in format 'vX.Y.Z' e.g 'v1.0.0'.

5.3.5 Verification

The concept of verification is one that has been the topic of many discussions, both internally within the team as well as with both the core team and the various HISP groups. This originates from the fact that the concept of verification is vague and can mean many different things, depending on context. Does it mean that we guarantee the quality, relevance and security of a component? Such a guarantee would be particularly useful to the user base, but would require a thorough system for examining components, no doubt including significant manual manpower for a reliable result. As the creation of such a system is not within the scope of our project, nor is the addition of manual labour desirable, we decided on a much simpler verification system.

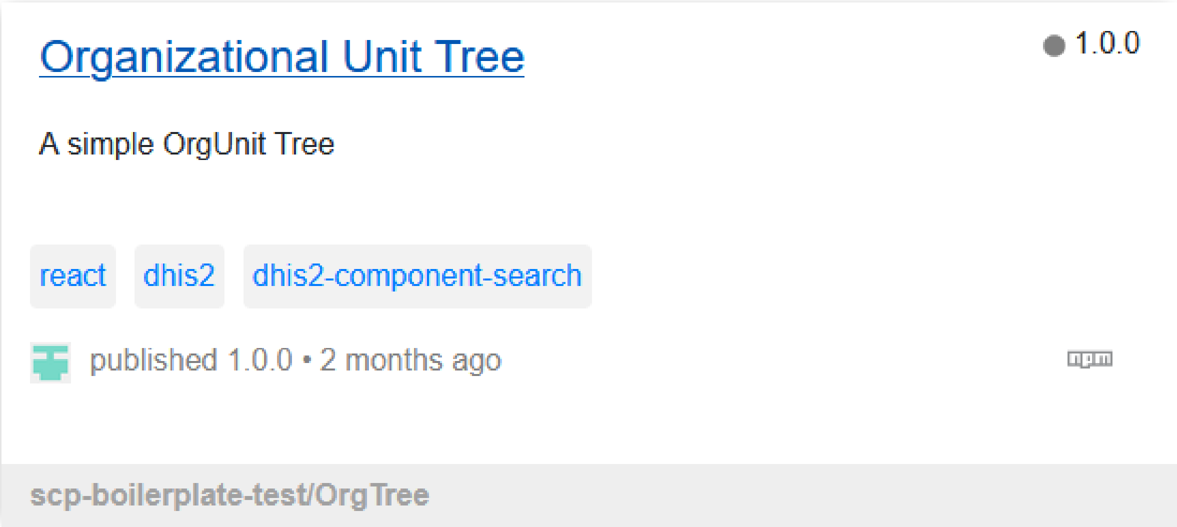

Figure 10: Unverified component

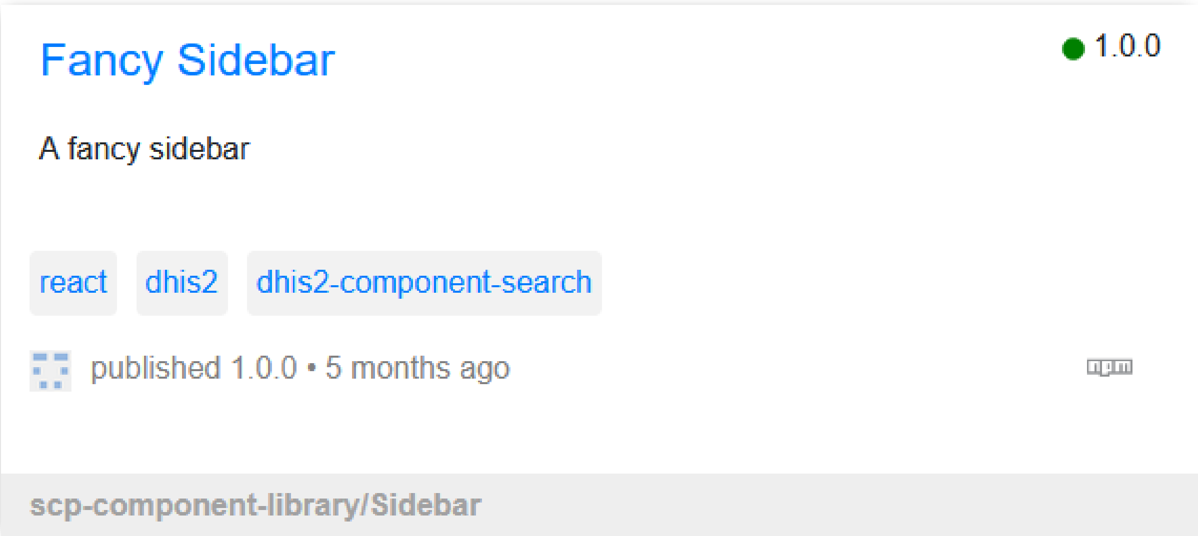

Figure 11: Verified component

The current version of verification simply guarantees that a component has successfully been made according to the standards listed in the section above, and that it has successfully passed CLI verification check in the automated pipeline. This means that any user of the platform will know that the framework used is either react or angular, but more importantly it ensures that the developer has taken the time and effort to create or adapt a component for the platform, increasing the likelihood that the component is of decent quality. What may be viewed as even more important for this loose verification system is what it allows the website to do. For the website to be able to present the components, its name and its description, the developer needs to insert this information in the package.json. Setting these standards in other words facilitates our communication of this critical information to the users and allows us to present the components in an understandable manner.

Reflections on verification

While the verification system chosen is binary, i.e., a component is either verified or not; several other models could be chosen. The possibility of a ternary or multary system was discussed at several points, in an effort to explore a softer verification system in an attempt to avoid the potential of a rift forming in between the creators of verified and unverified packages. Some worries were aired about the possibility of animosity forming within the ecosystem should verified components and their creators be given too many advantages over others. As the platform only allows for verified components to be created with react and angular, this could leave creators with different technological backgrounds feeling left out of the ecosystem. In the end this was not explored further for two reasons. One, it was deemed as adding more complexity to the platform and could prove challenging to effectively communicate to the users, and two, it was deemed unnecessary as the verification system does not provide advantages to the point where a rift forming was deemed a realistic worry. Information gathered from interviews support this decision, as the HISP groups did not seem concerned about this particular possibility.

6 Findings

In this chapter the relevant findings from the project will be presented, the most important of which are the design principles; prescriptive knowledge derived from the evaluations. Most of the focus will be on the final evaluation, though some contributions from earlier interviews will be mentioned as well.

6.1 Evaluation Method

As design science research is fixated on the design and development of an artifact, the natural conclusion to the process is the evaluation of said artifact. Evaluation knowledge has also been defined as one of the possible intended outcomes of a design science research project. As such, several articles have been written on the topic of listing and designing methods that are able to properly evaluate various types of artifacts (Peffers et al., 2012; Sonnenberg and vom Brocke, 2012; Venable et al., 2012).The diversity inherent in both the artifact types as well as the range of circumstances that will affect the project increase the number of methods that can be considered, and proper consideration of especially the artifact type as well as the stage of completeness is critical. Several authors list evaluation methods and their selection criteria, and among them Peffers et al. mentions several methods of design science research evaluation methods (Peffers et al., 2012). Included

DSR Evaluation Strategy Selection Framework		Ex Ante	Ex Post
		•Formative •Lower build cost •Faster •Evaluate design, partial prototype, or full prototype •Less risk to participants (during evaluation) •Higher risk of false positive	•Summative •Higher build cost •Slower •Evaluate instantiation •Higher risk to participants (during evaluation) •Lower risk of false positive
Naturalistic	•Many diverse stakeholders •Substantial conflict •Socio-technical artifacts •Higher cost •Longer time - slower •Organizational access needed •Artifact effectiveness evaluation •Desired Rigor: "Proof of the Pudding" •Higher risk to participants •Lower risk of false positive – safety critical systems	•Real users, real problem, and somewhat unreal system •Low-medium cost •Medium speed •Low risk to participants •Higher risk of false positive	•Real users, real problem, and real system •Highest Cost •Highest risk to participants •Best evaluation of effectiveness •Identification of side effects •Lowest risk of false positive – safety critical systems
Artificial	•Few similar stakeholders •Little or no conflict •Purely technical artifacts •Lower cost •Less time - faster •Desired Rigor: Control of Variables •Artifact efficacy evaluation •Less risk during evaluation •Higher risk of false positive	•Unreal Users, Problem, and/or System •Lowest Cost •Fastest •Lowest risk to participants •Highest risk of false positive re. effectiveness	•Real system, unreal problem and possibly unreal users •Medium-high cost •Medium speed •Low-medium risk to participants

Figure 12: DSR evaluation strategy. (Venable et al., 2012)

According to Venable et al. evaluations can be separated into four categories, divided on two axes as presented in the figure above. One axis being *Naturalistic* or *Artificial*, the other being *Ex Ante* or *Ex Post*. As one might expect from the name, the former axis concerns itself with the setting of the evaluation and how close to its real *natural* environment the artifact is evaluated. The closer it is, the more *real* and applicable the data gathered can be said to be. In addition, some information is only possible to acquire via a naturalistic evaluation, and some artifacts cannot be evaluated properly outside of it. To contrast, an artificial evaluation is an evaluation that occurs in a more constructed and controlled context, often useful to gather a smaller, more specific set of data. They are typically easier to plan and execute, and place fewer requirements on the setting, but this also means side effects are less likely to be identified. By virtue of being centred around artifacts, DSR is more suited to artificial evaluation than many other research methodologies, like for instance action research which would require the artifact to be evaluated within an organizational context which would be naturalistic by definition (Venable et al., 2012).

The main activity in the evaluation is the comparison of the artifact to its stated goals. This is not an empirical evaluation, and the intention is not to get an objective evaluation of the artifact. The purpose is to gather information on how the users in the HISP groups view the final artifact, and with a focus on how they respond to the standards it imposes on them. For more information on what I consider to be valid knowledge for the purposes of this evaluation, refer to the *Ontology and epistemology* section in the methodology chapter. As such a technical evaluation is not going to provide much relevant information, and similarly, an evaluation performed by either an external expert or the core team is also not necessary.

With the artifact not being actively used by the HISP groups, nor existing in the field in a properly functional state, it is clear that the evaluation would be an *artificial* one. However, when it comes to *ex ante* versus *ex post* the categorization becomes a little blurrier. Due to the special position of the artifact; that of being added to the design lab database; there exists a significant possibility that it will be modified and improved in the future. The goal of the evaluation should therefore not only be to summarize the artifact and the project, but also to gather data on how to further improve it. This place the evaluation in both the formative and summative bracket and makes it more challenging to put it into the framework of *ex ante* and *ex post* but given the fact that most knowledge gained is about features that should be improved, I place it in the *ex-ante* category despite it occurring as an end to the project.

Referring to the evaluation types identified as suitable for design science research by Venable et al. and Peffers et al. the choice was between a *demonstration* and an *illustrative scenario*. The main difference can be said to be the active party; in an illustrative scenario the subject is responsible for using the artifact and vice versa with regards to a demonstration. Of these two I chose to utilize the demonstration for two main reasons. One, a demonstration is easier to set up and coordinate with the HISP groups, something valuable when efficient communication has proven challenging. Two, the documentation is critical to the utilization of any artifact and was not an area of focus during the development of the platform. In addition, it was also not an aspect of the platform that I wanted to test. To avoid the results of the evaluation being affected by poor or lacking documentation I therefore opted for a demonstration for the final evaluation.

Emails were sent out to both groups, but ultimately only one of them; HISP Tanzania, responded and agreed to set up a meeting. Due to a miscommunication, only one of the members of the HISP group arrived. I had assumed the HISP member would act as an intermediary for the rest of the group, and they assumed I had sent the same email to all members. Still, the demonstration was performed, and we both ultimately agreed to set up a meeting; this time with all members invited.

6.2 Results

In this subsection the results and takeaways from the various interviews and meetings will be presented. This can be considered an extension to the methodology chapter and will have a connection to the process section of the artifact chapter which details some of the more technical findings and how they impacted the development process of the shared component platform.

6.2.1 Initial meeting with core team

The meeting was structured as a semi-structured interview, preceded by an introduction of all parties that included our areas of interests, and what we were planning on creating. The following interview was as mentioned loosely structured, allowing for interjections from all parties and encouraging discussion among different parties. The addition of the lead developer from Mozambique was beneficial to this, as we got a slightly deeper look at how the core team and the HISP group communicated.

As a result of this meeting, we managed to get an understanding of what the core team wanted from our project, in addition to some insight into how the HISP groups work to create applications. The initial form of our solution took form based on feedback received in this meeting, though in rough format. The gallery-like website for presenting components, a verification system, standards for uploading components are all examples of what were settled on in the meeting, though many of the technical details were either not discussed or changed in the later stages of the process. While the exact form of the solution was not decided at this point, we also planned for the platform to be in a usable state by around the middle of next semester to be able to utilize a course on digital platforms to test our solution. This course was held by a member of the design lab, and the initial plan was to get the students to use our solution in some way, not only as a means of stress testing it as the course regularly gains over one hundred students, but also as a way of kick starting it. One thought that was aired during this meeting was the idea of having the students create and upload components to the website, thus gaining several components presented on the website before the HISP groups got involved. Due to time constraints this idea was later scrapped, as the platform was not at a stage where this was deemed as feasible to execute in practice.

As the meeting was centred around the core team, it focused on what they would want from our solution as well as what they would be able to support in the future. At this stage we were still not certain as to which role we would have in the design of the platform, whether we would be the product owner or not. This would impact whether we could make decisions on the technology used and what standards to set by ourselves, or if we had to get a green light from the core team before we implemented anything. A final decision was not made on this during this meeting, but a tentative decision was made to consult the core team about the direction and big changes and make small decisions ourselves.

Another outcome of this meeting was the realization that the core team was wanting to nudge the ecosystem towards using the framework React. This is in part due to the fact that creating resources in a single framework is a lot easier than attempting to create them in several at once. As a result of this push, they also wanted us to contribute to this by encouraging the users of our solution to make use of react. A suggestion from the core team at this stage was to focus on React components only, trading flexibility for standardization. This would make the solution easier to integrate with the rest of the resources made by the core team and would also give them more incentive to help us create and later maintain our solution.

The main outcome of this initial meeting was direction for the future design of the solution, combined with assurance that what we created was within the scope of interests for both the core team and the HISP groups.

6.2.2 HISP interviews

The first interviews, while somewhat dispersed in time, were structured very similarly for both HISP groups we interviewed, in part due to a desire to contrast the groups and see how their processes compared to one another. At this point of the project what we were interested in was mainly how they worked in the groups and how they went about developing applications for the DHIS2 platform. That includes how they developed their applications from start to finish, what their design process looked like, which tools they used and how they interacted with the customer, among other things. While the early form of the platform was presented to the groups here, the platform itself was not the focus of these particular interviews.

6.2.3 HISP Mozambique

The structure of the interview was as presented above, a quick introduction of ourselves and the platform followed by a semi structured interview. For the purposes of this thesis, I will omit findings I deem irrelevant to the final version of the shared component platform, in other words findings that did not play a substantial role in shaping the final result. This for the most part includes some of the minutia on their development process; information that may have been interesting, though its contribution to my project being minor at best.

Collaboration efforts

One of the pieces of information that was rather critical to our project was their existing communication with other HISP groups, with a focus on collaboration and code reuse. All information we had acquired before, from for example informal conversation with other students, implied that this communication was limited or almost non-existent. When questioning HISP Mozambique in this however, they informed us that they did indeed have something like this as a part of their process. Before developing they check if the application has been made by other groups. First, they check public spaces, then they might contact other groups or colleagues. As an example, an instance of collaboration with HISP Tanzania was mentioned where they reused a mobile app for analytics after some localization; the addition of the language Portuguese and some other minor changes, were performed.

Verification

Another important topic that was touched upon in some form during all interviews was verification. HISP Mozambique have already implemented a verification step as part of their process. They work as a group divided into three teams: the research team, the implementer team and the development team. The implementation team is responsible for testing and implementing the application created by the development team, yet they realized they were receiving solutions and apps with a lot of bugs. This prompted the creation of a verification process performed by the development team on the application before sending it to the implementation team. For us this means that the team is more likely to accept the addition of work that our intended verification system would add to their development process, and that they perhaps could implement it as a step in an existing process.

Component development

When it comes to their process of developing these applications, we gathered some information that either reaffirmed our decisions or affected the path we took when developing the shared component platform. The first thing of note is the fact that they try to make their components as generic as possible, so that organizations similar to their client can use the application without issue. The phrase one of the developers used was “...think global, but work local”, and demonstrates a frame of mind that suits a shared platform such as the one at the centre of our project.

HISP Mozambique does not operate with a private repository of components, but they instead have an app skeleton with basic structure. They originally had a plan to build a kind of internal library where they could reuse components as dependencies, but instead they are now reusing source code of other applications. The reason for not creating this library is because they do not think they really need it as an organization, because they can quickly reuse the components from using the source code. They feel like they can easily copy the class, insert it into a new project and keep working on that. Since they think this is easy enough, creating a library is not the main issue at the time of the interview, though they still have some plans to revisit it in the future.

Use of the DHIS2 design infrastructure

As we can consider the shared component platform in a final working state as a boundary resource in the DHIS2 ecosystem there is value in knowing if and how they make use of other, existing boundary resources. When asked about this they responded that they initially used D2 UI libraries with mixed results; the mixed results stemming in large part from a varying level of documentation. Some components that were well documented were viewed positively, in contrast to the ones that were lacking in this aspect and thus provided a much poorer experience. As an anecdote, this coincides very well with my own experience with developing applications for the DHIS2 platform.

As mentioned, the main reason for stopping the use of most core components was the lack of documentation. They thus prefer to write the components they need to solve their issues from scratch, or using material-ui ("Material-UI," n.d.). They also try to not only go online and look at the documentation, but also check with the core development team, as they are part of the HISP groups. Originally, they used d2 (the app development platform), but when they realized that they might not get an updated version of the library, or updated documentation, this uncertainty made them change their mind about using these components.

Development framework

As mentioned in earlier sections, there is a desire from the core team to push the different HISP groups towards using React to develop applications. This is done by encouraging developers to utilize the framework via the creation of various boundary resources that are written for it. HISP Mozambique is one of the groups that has started to develop applications in the framework, in part due to efforts made by the core team.

One reason why they switched to React was that they had a lot of environment bugs occurring when working with Angular. They claim that this was mainly because of the quick version change of Angular, meaning that they had a lot of problems with sticking to a stable version that they wanted to use in all their applications. As an example, they state that "...sometimes we implement something using one version, but two months later we want to use the same component and start having some dependency problems because of the version change...".

In addition to existing frustrations with Angular, they also attended the DHIS2 app development academy in 2018 where they realized that the future approach to DHIS2 app development was going to be React. They found out that the core development team would be encouraging React for the next few years, so they made a deliberate choice to switch away from Angular towards React. They still have applications in Angular, but they have made an attempt to recreate applications using React in attempt to ease future maintenance, having already rewritten some of their simpler applications.

Component maintenance

HISP Mozambique also maintains the same apps that they develop. The lead developer stated that "...mainly this is about the API versions cause the customers don't want to know about that." For this reason, they try to keep a control of what DHIS2 versions being used by the different applications are being used. This contributed to our reasoning for including the DHIS2 version tag on the platform as mentioned in the artifact chapter. They also note that so far, they have not had many problems with maintaining the same component in different apps. This further implies that a component sharing platform which encourages this type of component reuse is feasible.

6.2.4 HISP Tanzania

We held a similar interview with Tanzania later during early fall of 2020, where we wanted to gather the same information as from Mozambique and contrast the findings. The groups present two different philosophies to component reuse that is quite likely not unique to HISP or DHIS2. Whereas Mozambique have shifted from Angular to React, Tanzania has not. This affected our development process, as their feedback changed our mind regarding the standards on frameworks, allowing not only React but also Angular in this initial version. During this interview we also had a representative of the core team sitting in and asking the occasional question, something we found very useful as a way of gaining insight into what they find interesting.

Package publishing

Of specific note is the decision on scope, an NPM feature that allows packages to be grouped under one namespace. As an example, HISP Tanzania has their own application published under their own namespace, namely @iapps, something that allows us to more easily find their applications (“@iapps - npm search,” n.d.). This seemed at first as the logical choice for us, but during this meeting the core team developer stated that “…don't think we should have a specific scope for verified packages, I think that complicates NPM publishing and updates … I would actually say that it is probably going to be a big barrier for package versioning.” This is the main reasoning as to why the platform does not introduce its own scope, as our intention is for this platform to be easily maintainable.

Development

While there is a large difference between Tanzania and Mozambique when it comes to component reuse, the starting point is actually somewhat similar, atleast in description. Tanzania starts development with what they call a ‘seed app’, or a ‘skeleton architecture’ that is written in angular. This seed app includes the basic components that they feel they need in most applications. This allows them to jumpstart development quickly, and then to use their own components to build on top of that.

Component library

Their own component library contains many of the components that they themselves have created and used in their applications. Most of these components are UI components, and they are written in Angular, and can be found under their scope @iapps (“@iapps - npm search,” n.d.). They package components by themselves which allows for easy reuse. This is very similar to what the shared component platform tries to do, and though it is less expansive verifies that a component platform is something that groups are already interested in. Tanzania confirmed this when we asked them, and they expressed their interest in contributing to the new platform.

Collaboration

When asked, Tanzania expressed interest in collaborating with other groups, but in contrast to Mozambique they claim to not have used components from other groups, something they believe to be because they might use technology that not many are using. They admit to looking for some specific componetnts, but have not found them. They also say that they believe they exist, and if they could find these components they would use them. This affirms the importance of a global component sharing platform.

When asked about what they would improve about the component reuse process their answers confirms this: "…the discoverability of the components, in a sense that it is easier to search for the components". In addition to this they also note that while a component might be discovered, there is also a need for proper documentation for the component to be useful in many cases. This is something our platform does not address, choosing instead to place less restrictions on its users.

They also express a desire for more collaboration, and believes component reuse could strengthen the community aspect of the ecosystem. "another aspect which I think we could improve is maybe the involvement of other developers … [the developers] can easily get acquainted or easily [use] components that we have, and also contribute … [that could help with] having a community around I'd say"

Lastly they invite to more collaboration. They state that they are open to collaboration, but have not received many requests for it. One developer states: "in principle we are open to any collaboration, unfortunately we haven't really received any [requests] … most of our components we have put in our GitHub repositories, and they are public … we are actually considering the approaches of creating issues, or requests, or etc to invite people to collaborate".

6.2.5 Other meetings

Other meetings were mainly oriented around garnering technical feedback about the platform and were less structured as a result. We attempted to set up a meeting with the HISP groups, but due to issues on their end that was canceled. Due to this and also the fact that this was so far into the project's lifespan, there is not much information that should be covered in a separate section, and any data gathered during this period will be presented as part of the next section.

6.2.6 Evaluation

As mentioned above, the final evaluation was directed at the HISP groups and took the form of a demonstration where the process of using the artifact was shown to the group and followed by an open discussion and some questions. Five members of HISP Tanzania showed up, including the lead developer of the group.

Choices

While the group seemed to agree that the site looked usable and intuitive, the most immediate comment to the presentation of the website was questioning how a user would go about choosing a good component when presented with several options. A member of the team commented "*how do i choose the best one*" when referring to the situation where a user would be presented with multiple similar components; an inevitable situation should the platform see a lot of use. This sparked some discussion around the topic of component selection and presentation, with several possible avenues of improvement identified for further research. An improvement that was brought up was the possibility for a community consensus, in other words a way for the users of the platform to give their opinion or otherwise rate the components in a way that would be helpful for the decision making of other users. As stated by a group member, "*it might be confusing if i'm searching for the org unit and then i get like five of them… i might be wondering which one is the best for us*". The lead developer of the group elaborated and said that, "*...there could be a mechanism [for] people to provide stars, to show how many people it has helped… the component with more stars might show that maybe that it is the component they could easily work with...* ". This feeds the common consensus that the verification system existing on the platform is somewhat lacking in what information it communicates to the user.

Verification

One of the major points of contention during the design and development process was the concept of *Verification.* As such this has been a topic that has been aired often during interviews, and the final evaluation was no exception. After the presentation of the website, some comments on the visual representation of the verification came forth. As the verified components are differentiated from the unverified by means of changing the color of a dot from grey to green, some concerns on whether this was clear were aired. The lead developer stated that: "*the green dots you put might not be so informative ... I think the thumbs up thing might be a better way*".

In a similar vein to the desire for more clear communication of information, the team also suggested that the verified components were presented to the user before other components. They suggested that a user be presented with only verified components when opening the website, and instead search for components that are not verified should this be desired. A team member said: " *I think maybe we should put it at least whenever somebody opens it [they] find components that have been verified...*".

Following up on the concept of a community consensus, when asked how they would weigh it compared to verification, they agreed that a more used component or a component that had gotten positive feedback from the other users would be chosen before a verified one. " *I would consider a more used component worthwhile compared to whether it has been verified or compared to other criteria i would say*". In other words, a verification system as presented would not be as important as a community ranking system. They placed a lot of importance on this concept, urging us to consider it. One developer stated: "*I would say that is huge, very important information I would say to help somebody decide*".

Standards

With the shift towards react from the core developer team, it has been made clear that they wish for the component sharing platform to aid in this regard by encouraging users to utilize react over other frameworks. As mentioned in the artifact chapter, a simple way of doing this would be to simply not support other frameworks, either during the verification process or on the site itself. During both earlier interviews and during the final evaluation itself some pushback from HISP Tanzania was discovered, a somewhat expected reaction due to their sunken cost in the framework Angular. They are not alone in the usage of this framework, and as such this response could be expected to be repeated across several nodes in the DHIS2 ecosystem as well. Despite a sizable amount of resources invested in Angular, the general sentiment towards the core developers push towards React seemed to be more positive than expected, with the consensus within the group leaning more towards disagreement rather than discontent. The lead developer for instance stated that: “we are not against react… we might at some point be shifting into using react… The only problem is the cost.”

Though it might be expected that the group was predisposed towards disliking the core teams push towards React, they presented several reasons for keeping the ecosystem more open and welcoming towards a multitude of technologies. One developer in particular seemed particularly engaged with the concept, stating in several interviews that “I really like the idea of diversity” when referring to participants in the ecosystem. The developer in question highlighted two main reasons as to why the platform could benefit from an increase in flexibility. The first of them being the risk of a new and better framework coming along in the future as technology develops, and therefore restricting the contributors to a single framework or otherwise designing the platform around the utilization of just one framework would likely result in more maintenance in the future. The developer states that: “*[I] wouldn't really want to choose one framework over the other as I know many will come and probably with even more advantages than react or angular*”. The second stated reason was for the ecosystem to be more attractive to a wider range of developers. The developer states that allowing for contributions to come in a wide range of formats using a wide range of technologies the platform could gain users with existing expertise in various technologies that otherwise would not consider creating for or uploading to the platform. He states: “*I think it will actually bring more developers, because you will find developers with different technological backgrounds rather than react, ...developers will come in, ... it brings them around dhis2 ecosystem*. ”

When asked about their opinion on what role the platform should play in the DHIS2 ecosystem when it comes to standardization, the team seemed to agree that a more flexible approach is the correct one. There was a consensus that some standards were beneficial to make sure the components hosted were relevant for the platform, but they did not want us to incentivize the use of some technologies if possible. One of them stated that standards should “...be universal or [we should]
have multiple for different technologies.”

Visual component presentation

The group at several times pointed out the lacking presentation of the components. The fact that there is only a name, description and verification tag presented on the website means that for the user to get additional information on the component they would need to visit the npm page that is linked in the card. This represents an additional step for the user, one that with benefit could be removed if possible. During early talks some functionality for presenting either pictures or snippets of the code were discussed, but due to time constraints this was not implemented. This was commented on during the presentation of the website, with a member explicitly requesting this. The group pointed to another project by the DHIS2 core team, namely the storybook, as an example to imitate. The storybook is a site that presents visual components and shows the code that composes them, and as such allows for easy duplication of a specific look. This way the users could "...get the sense [of the component]" and know with more assurance if a component was right for them.

Process

The process of preparing a component for the website was presented after the website, and initial reactions were positive. The manual editing of the package.json was deemed as feasible, and simple to implement in the workflow. The lead developer stated, "...I think it's doable, looking at the process I think it's not that complicated to work with". The manual aspect of the process was not framed by the group as one of the primary hindrances to start using the platform.

6.3 Design Principles

The outcome of the final evaluation can be distilled into five design principles that can be applied to component sharing platforms, in particular in similar developing countries, but due to their generality they should to some degree also be applicable to more general component sharing platforms where the community can contribute to the content of the platform.

Function over form

Communicate in a clear and concise manner with both language and iconography. Many websites opt for a minimalistic look and feel, us included. This works for the overall design, but when we need to communicate something to the user, they would rather we use more specific iconography. Similarly, presenting information is more important to them than a sleek layout.

Minimize work

Present information as soon as possible. The number of clicks a visitor must use to find what they want should be as low as possible. When a user must navigate several windows to find what they need they may not bother at all.

Decisions are hard - make them easy

When faced with several choices, some way of ranking them must be presented. This is especially important when it comes to a platform like ours, where there are likely multiple versions of similar components.

Community matters

A community consensus builds trust like nothing else. A package that is used by many will be considered above others, to the point of outranking other forms of verification.

Allow the platform to represent its community

An ecosystem like DHIS2 inevitable contains individuals with very different forms of expertise. By allowing for diverse contributions the site will be more attractive to experts with knowledge in a wider range of fields and ability with a wider range of tools, and will also avoid the possibility of ostracizing them by excluding them from the platform.

7 Discussion

7.1 Contributions

The first contribution that should be mentioned is what Hevner et. al., would call *design artifact* (Hevner et al., 2004). I do not consider this to be the main contribution, but it should be mentioned. The reason for this is the fact that it is not in use, and therefore does not contribute to the problem area at the moment of this thesis. That is not to say it does not contribute anything at all, but rather that the contribution is to the Design lab and its repository of software. From there it can be picked up by future students and completed to a stage where its intended purpose can be achieved.

What I consider the main contribution is the *design construction knowledge,* the design principles that distill the learnings from the project. I would place them in the field of generic software and platform theory due to the goal of creating a generic platform that still could be used by local developers in the global south, and the focus some of them have on the community aspect of the platform ecosystem. This connection means that they are of particular relevance to development in the global south, but they are still usable outside of such a context. Evaluation their value is difficult, as they are quite general, but that does not mean that reaching these conclusions is easy and that it can be done fast. They present what is important to the users of this type of platform and will allow developers to more easily decide what features to prioritize when developing. I believe that they especially could help to remove some of the misconceptions that could occur due to a design-reality gap when a developer not from the global south attempts to develop for that region.

The RQ was "*How to balance standardization and flexibility for a shared component platform as a boundary resource in a diverse ecosystem.*". It should be noted that this is mainly addressed by the final two principles concerning the community. The platform benefits from a wider range of users, and in its state where the community engagement is somewhat lacking, there is a potential discussion to be had about whether it is best to focus on growing the community or if that time is better spent on building the technological base of the platform. Ultimately this thesis does not provide an answer to this question, but rather presents a perspective from the community that should be considered, and maybe tested. While DHIS2 is not in any danger of losing its userbase, other platforms that could utilize these principles might be in that stage, where enough community engagement could be critical to its future, and such a decision could be made. The rest of the principles are prescriptive knowledge that could save time and effort for others creating similar platforms in similar situations. Their generality means that they are useful for a wider range of contexts, while also perhaps less important for this specific one.

Of note is the fact that the evaluation was not thorough enough to state this with absolute surety, and a study of the platform in the field, e.g., an Action research project, will be necessary to either confirm or deny these statements.

7.2 The role of the SCP in the DHIS2 ecosystem

From our evaluation we have gotten the impression that a boundary resources like this shared component platform is better suited not as a method for standardization, but rather as a vessel for collaboration. These are not mutually exclusive but maintaining both are difficult. Everyone we consulted from the HISP community expressed a desire for more collaboration, and Tanzania mentioned specifically that they believed sharing and indirectly collaborating on application via a shared platform could facilitate this. The principle of allowing the platform to represent its community attempts to show the value of allowing for some resources in the platform to be more flexible than perhaps the core needs to be.

7.3 The next step

In addition to the more general design principles that of course can and should be applied to the artifact when and if possible, some findings more specific to the artifact have been identified as well. Due to the formative aspects of the final evaluation some recommendations can be made, not only with regards to specific changes, but also with regards to some potential paths forward that might be of interest to interested parties in the future. As the artifact was presented to the different parties over the duration of the project, there were a plethora of aspects with the platform that could be vastly improved, though such endeavors were deemed to be outside the possible scope of this thesis, due to both time and skill constraints. Some avenues of particular interest that should if possible be explored further are the technical aspects of the platform, the possible future integration of the platform to the

DHIS2 ecosystem, the security aspects of an open component sharing platform, as well as the exploration of the platform in the field.

7.3.1 Technical advances

That the platform could benefit greatly from a technical upgrade became clear not only when the platform was presented to the other parties, but also when the platform was discussed internally within the development group. Of the identified technological improvements, the ones that are of specific note include *Automatic component extraction* and the *Display of components on the website.*

7.3.2 Potential core integration

The core development team is actively developing the DHIS2 platform, and this includes expanding on the tools available to the users in the ecosystem, as well as the communication of these tools. As a step in this process the core team has expressed interest in the inclusion of some part of the component sharing platform into the DHIS2 platform as part of the core. Special interest has been shown with regards to including the functions of the CLI as part of the general DHIS2 CLI. This would go a long way towards ensuring the platform is used by the users within the DHIS2 ecosystem. If the component sharing platform is included as part of the core of DHIS2 this would open up more possibilities for development and research, in part due to the professional support it would get, and the legibility it would receive as a result.

7.3.3 Field research

Regardless of the quality of the finished artifact, if it does not lead to a change within the organizations for which it is intended, it has failed to properly address the practical problem for which it was created. To test how it is received and used by real developers within the HISP ecosystem, it is necessary to perform a naturalistic evaluation in the field. This would require a more mature artifact and can be easily imagined as either an action research project performed by the future developers of the platform, or as a longitudinal case study performed by someone after the artifact’s ‘completion’.

7.3.4 Security

At the moment there are no security checks in place on the platform, and anyone can upload malicious code, should they wish to do so. As the current HISP community is fairly small and insular this is as of yet noy that big of a problem; security by means of obscurity might be somewhat of an applicable term here, as there is little that would attract malicious coders to the DHIS2 ecosystem. Still, the sensitive nature of health data as well as the possible harm to human life that could follow as a result of a cyberattack directed at the health sector means that the lacking security features of the platform is a problem that should be addressed. As the structure of the component sharing platform is quite unique, there is plenty of room for much interesting research with regards to security.

7.4 Notes on the pandemic

While mentioned elsewhere, the effect of covid on this project was large. Shutting down all travel possibilities meant that the planned form of research was impossible and forced a shift in thinking when the project had already begun. In addition, a high level of restrictions was in place for most of the master thesis. This had a substantial impact on the work by means of reducing face to face meetings and communication, in addition to impacting mental health by means of a pandemic induced isolation.

8 Conclusion

We developed a shared component platform prototype and evaluated it against its intended userbase with the intention of extracting design principles as part of a design science research project. As a result of this project, five principles were extracted as the primary contribution. While the platform is not in a state where it can be utilized by the DHIS2 ecosystem, it provides a robust technical basis for future development, complemented well by the design principles presented in this thesis. By considering these principles, future researchers can improve on the shared component platform and take it into the field for a naturalistic evaluation. There is still a need for a shared component platform in the ecosystem, in addition to the other, innovative solutions to the problems that lies as a reason for its existence. While the core teams momentum is currently good, and they are consistently improving the DHIS2 platform, it is important that they do not lose the community, but rather that they strengthen it. This thesis attempts to show that while standardization certainly is necessary, it must be tempered with flexibility to retain a large and diverse ecosystem.

References

Abdul Rehman, A., Alharthi, K., 2016. An introduction to research paradigms 3.
About DHIS2 [WWW Document], n.d. . DHIS2. URL https://dhis2.org/about/ (accessed 6.1.21).
About scopes | npm Docs [WWW Document], n.d. URL https://docs.npmjs.com/about-scopes (accessed 5.20.21).
Anderson, J.G., 2007. Social, ethical and legal barriers to E-health. Virtual Biomed. Univ. E-Learn. Secure EHealth Manag. Risk Patient Data 76, 480–483. https://doi.org/10.1016/j.ijmedinf.2006.09.016
Atomic Design | Brad Frost [WWW Document], n.d. URL https://bradfrost.com/blog/post/atomic-web-design/ (accessed 5.19.21).
Baldwin, C.Y., Woodard, C.J., 2009. The Architecture of Platforms: A Unified View, in: Platforms, Markets and Innovation. Edward Elgar Publishing.
Baskerville, R.L., Wood-Harper, A.T., 2021. Qualitative Research in Information Systems. SAGE Publications, Ltd, London. https://doi.org/10.4135/9781849209687
Bass, J., Heeks, R., 2011. Changing Computing Curricula in African Universities: Evaluating Progress and Challenges via Design-Reality Gap Analysis. Electron. J. Inf. Syst. Dev. Ctries. 48. https://doi.org/10.1002/j.1681-4835.2011.tb00341.x
Braa, J., Sahay, S., 2012. Integrated Health Information Architecture Power to the Users Design , Development and Use.
Checkland, P., Holwell, S., 1998. Action Research: Its Nature and Validity. Syst. Pract. Action Res. 11, 9–21.
DHIS2 Application Platform [WWW Document], n.d. URL https://platform.dhis2.nu/#/ (accessed 5.20.21).
DHIS2 Component Search [WWW Document], n.d. URL https://dhis2designlab.github.io/scp-website/ (accessed 6.1.21).
dhis2designlab/scp-cli, 2021. . DHIS2 design lab.
dhis2designlab/scp-whitelist, 2021. . DHIS2 design lab.
Dittrich, Y., Vaucouleur, S., Giff, S., 2009. ERP Customization as Software Engineering: Knowledge Sharing and Cooperation. IEEE Softw. 26, 41–47.
ESLint - Pluggable JavaScript linter [WWW Document], n.d. URL https://eslint.org/ (accessed 6.1.21).
Evans, P., Gawer, A., 2016. The Rise of the Platform Enterprise: A Global Survey.
Garrib, A., Stoops, N., McKenzie, A., Dlamini, L., Govender, T., Rohde, J., Herbst, K., 2008. An evaluation of the District Health Information System in rural South Africa. South Afr. Med. J. Suid-Afr. Tydskr. Vir Geneeskd. 98, 549–552.
Ghazawneh, A., Henfridsson, O., 2013. Balancing platform control and external contribution in third-party development: The boundary resources model. Inf. Syst. J. 23. https://doi.org/10.1111/j.1365-2575.2012.00406.x
Grix, J., 2004. The Foundations of Research, Palgrave Study Skills. Palgrave Macmillan.
Heeks, R., 2006. Health information systems: Failure, success and improvisation. Int. J. Med. Inf. 75 2, 125–37.
Henfridsson, O., Bygstad, B., 2013. The Generative Mechanisms of Digital Infrastructure Evolution. MIS Q. Manag. Inf. Syst. 37, 907–931. https://doi.org/10.25300/MISQ/2013/37.3.11
Hevner, A., Chatterjee, S., 2010. Design Science Research in Information Systems, in: Management Information Systems Quarterly - MISQ. pp. 9–22. https://doi.org/10.1007/978-1-4419-5653-8_2

Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design Science in Information Systems Research. MIS Q. 28, 75–105. https://doi.org/10.2307/25148625
HISP Network [WWW Document], n.d. . DHIS2. URL https://dhis2.org/hisp-network/ (accessed 6.1.21).
HISP Tanzania [WWW Document], n.d. . GitHub. URL https://github.com/hisptz (accessed 5.31.21).
@iapps - npm search [WWW Document], n.d. URL https://www.npmjs.com/search?q=%40iapps (accessed 6.1.21).
Koskinen, K., Bonina, C., Eaton, B., 2019. Digital Platforms in the Global South: Foundations and Research Agenda, in: Nielsen, P., Kimaro, H.C. (Eds.), Information and Communication Technologies for Development. Strengthening Southern-Driven Cooperation as a Catalyst for ICT4D. Springer International Publishing, Cham, pp. 319–330.
Li, M., 2019. Making Usable Generic Software - The Platform Appliances Approach. https://doi.org/10.13140/RG.2.2.11381.83687
Li, M., Nielsen, P., 2019. Making Usable Generic Software. A Matter of Global or Local Design? https://doi.org/10.13140/RG.2.2.31514.49603
Lippeveld, T., 2001. Routine Health Information Systems: The Glue of a Unified Health System.
Lippeveld, T., Sauerborn, R., Bodart, C., 2000. Design and Implementation of Health Information Systems.
Manya, A., Braa, J., Øverland, L., Titlestad, O., Mumo, J., Nzioka, C., 2012. National Roll out of District Health Information Software (DHIS 2) in Kenya, 2011 – Central Server and Cloud based Infrastructure.
Mate, K.S., Bennett, B., Mphatswe, W., Barker, P., Rollins, N., 2009. Challenges for routine health system data management in a large public programme to prevent mother-to-child HIV transmission in South Africa. PloS One 4, e5483. https://doi.org/10.1371/journal.pone.0005483
Material-UI: A popular React UI framework [WWW Document], n.d. URL https://material-ui.com/ (accessed 5.31.21).
Nicholson, B., Nielsen, P., Sæbø, J., Sahay, S., 2019. Exploring Tensions of Global Public Good Platforms for Development: The Case of DHIS2. pp. 207–217. https://doi.org/10.1007/978-3-030-18400-1_17
npm [WWW Document], n.d. URL https://www.npmjs.com/ (accessed 5.19.21).
npm-audit | npm Docs [WWW Document], n.d. URL https://docs.npmjs.com/cli/v7/commands/npm-audit (accessed 6.1.21).
Peffers, K., Rothenberger, M., Tuunanen, T., Vaezi, R., 2012. Design Science Research Evaluation, in: Peffers, K., Rothenberger, M., Kuechler, B. (Eds.), Design Science Research in Information Systems. Advances in Theory and Practice. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 398–410.
Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S., 2007. A Design Science Research Methodology for Information Systems Research. J. Manag. Inf. Syst. 24, 45–77.
Phone, V. address O.-J.D. husGaustadalléen 23 B.N.-0373 O.N.M. address P.O. box 1080 B. 0316 O.N., fax, n.d. Health Information Systems Programme (HISP) - Department of Informatics [WWW Document]. URL https://www.mn.uio.no/ifi/english/research/networks/hisp/index.html (accessed 6.1.21).
Sæbø, J.I., Kossi, E.K., Titlestad, O.H., Tohouri, R.R., Braa, J., 2011. Comparing strategies to integrate health information systems following a data warehouse approach in four

countries. Inf. Technol. Dev. 17, 42–60. https://doi.org/10.1080/02681102.2010.511702
Sonnenberg, C., vom Brocke, J., 2012. Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research, in: Peffers, K., Rothenberger, M., Kuechler, B. (Eds.), Design Science Research in Information Systems. Advances in Theory and Practice. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 381–397.
Stansfield, S., Orobaton, N., Lubinski, D., Uggowitzer, S., Mwanyika, H., 2008. The Case for a National Health Information System Architecture ; a Missing Link to Guiding National Development and Implementation.
Tilson, D., Sørensen, C., Lyytinen, K., 2012. Change and Control Paradoxes in Mobile Infrastructure Innovation: The Android and iOS Mobile Operating Systems Cases. 2012 45th Hawaii Int. Conf. Syst. Sci. 1324–1333.
Tiwana, A., 2013. Platform Ecosystems: Aligning Architecture, Governance, and Strategy. Platf. Ecosyst. Aligning Archit. Gov. Strategy 1–302.
Venable, J., Pries-Heje, J., Baskerville, R., 2012. A Comprehensive Framework for Evaluation in Design Science Research, in: Peffers, K., Rothenberger, M., Kuechler, B. (Eds.), Design Science Research in Information Systems. Advances in Theory and Practice. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 423–438.
WHO Packages [WWW Document], n.d. . DHIS2. URL https://dhis2.org/who/ (accessed 6.1.21).

Appendix

Appendix 1: Interview guide

1. The setting and context of person and organization. (*Introduction*)
   a. Their role, and what they work with?
   b. Shortly about the structure of the org/group?
      i. Developer backgrounds
      ii. Academic influence
      iii. Hierarchy
      iv. Size
      v. Focus. E.g. are they working solely within the scope of dhis2.
2. The process of app development (
   . From starting the project and until delivery.
   a. What languages they use for development.
   .React, Angular, plain js, something else?
   b. Do they use the Application Platform and the App Shell? (
   .The bootstrapper
   i.Headerbar
   ii.Build scripts
   iii.Runtime
   iv.Etc
   c. What design principles, if any, do you adhere to? (
   .Adherence to dhis2 visual guidelines. Colors/buttons
   i.Have you used any of the UI components developed by the core team? Why/why not (use Angular, so maybe not)
   d. Which coding conventions, if any, do you comply with?
   .Naming convention
   i.Peer reviews
   ii.Code formatting - linting etc.
   iii.Formal preparation of components for reuse
3. Design infrastructure (
   . How they experience the design infrastructure, meaning all the tools and resources they have available to use within the ecosystem. APIs, the Application Platform, community of practice, academies, etc.
   .Useful, not useful?
   i.What do they use?
4. How do you onboard new developers? Do they receive any training or do they start developing right away?
   . What experience do new developers typically have when it comes to programming?
5. The component library / component reuse
   . How the component library came to be? Why? When?
   a. How do you use it during app development?
   b. Is there further development work being done on it, adding components and such?
   c. Is everything written from scratch or do you reuse components?
   d. With regard to component reuse, what works well and what does not?
   e. What could be improved in the component reuse process?
   f. Where do you store and share components? *If not using gitlab, github, npmjs - why not? What are the reasons behind it?*
6. Collaboration
   . Do you collaborate with the other HISP groups and the core team?

.[If collaborate] what works well and what does not?

7. Tooling (

 .	Tools for development and collaboration

a.	Tools for internal communication or with other HISP groups

8.	Project setup and framework (

a.	Project management framework - Agile, different teams,

b.	Project members pr. Project?

c.	The number of ongoing projects at the same time?

d.	Any collaboration between teams on different projects?

9.	General work practices

 .	Can you tell us a bit about your typical workday?

a.	When do you start and end your workday? Breaks? Social events?

10.	Future plans

a.	Major plans moving forward?

b.	New practices, languages or frameworks, libraries?

11.	Their collaboration in the project

 .	Why do you want to participate in this project?

a.	Are there any comments or thoughts about this interview or the platform we are making?

b.	Do you have anything else you want to share that you feel relevant to our study?

Appendix 2: Project description

About the project

Increasingly, software implemented in complex organizational settings is based on generic software products that are developed to serve multiple organizations. To be usable within specific organizations, the software often undergoes some level of ‘localization’ during a process of what we refer to as implementation-level design. Here, generic attributes of the software such as user interfaces are adapted to fit with local work practices. In this process, designers and developers have to understand local requirements and try to reflect these in the design of the generic software. One approach to localization is to develop custom web-based ‘apps’. This approach provides the implementation-level designers with significant design-flexibility to make locally usable and relevant user interfaces and functionality. On the other hand, the process is often time-consuming as much has to be built from scratch. Responding to this challenge, the DHIS2 design lab wants to explore the possibility of creating a shared component library, which can be leveraged upon during local custom web-app development. When creating new custom apps, the components (i.e., React components or similar) may provide a starting point which makes the process more efficient. Further, the component store should be open so that different groups of implementation-level designers can contribute with components that may benefit others in the community.

Practically, this master project is about designing, developing and evaluating a platform that facilitates this sharing of reusable web-based components. Features that may be relevant are typical aspects from open source community projects such as the ability to upload components, facilitate automatic peer-reviews, and motivational mechanisms such as badges, rewards and rankings based on contributions. Theoretically, knowledge about the process and result of the practical work can contribute with interesting findings to research on generic software implementation, open source communities, and software platform design.

The project may include field-work in countries such as India or Tanzania to work together with real implementation-level designers as potential contributors to the platform.

Appendix 3: NSD Consent form

Are you interested in taking part in the research project *"Shared Component Platform"*?

This is an inquiry about participation in a research project where the main purpose is designing, developing and evaluating a platform that facilitates the sharing of reusable web-components for use in web-based app development for DHIS2 software platform. In this letter we will give you information about the purpose of the project and what your participation will involve.

Purpose of the project
Practically, this research project is about designing, developing and evaluating a platform that facilitates this sharing of reusable web-based components. Features that may be relevant are typical aspects from open source community projects such as the ability to upload components, facilitate automatic peer-reviews, and motivational mechanisms such as badges, rewards and rankings based on contributions. Theoretically, knowledge about the process and result of the practical work can contribute with interesting findings to research on generic software implementation, open source communities, and software platform design.
This research project will serve as a basis for the master's thesis of master's students at the University of Oslo.

Who is responsible for the research project?

The Department of Informatics at the University of Oslo is responsible for the project. The project is undertaken by three master's students (also referred to as the Component Team), [master student], [master student], and [master student]. These master's students are supervised by associate professor [supervisor], associate professor [supervisor], and doctoral research fellow [supervisor].

Why are you being asked to participate?
You are selected for participation in the meeting on the basis of your role and position, and activities related to development of DHIS2 applications.

What does participation involve for you?
If you choose to take part in the project, this will involve participation in an online meeting on the topic of the Shared Component Platform. The purpose of this meeting is to discuss and get feedback on the current state of the platform, and plan further development. Relevant topics will include the procedures of package verification, interface and visual design as well as core functionality. The discussion will take the form of a video-conference meeting, conducted over approximately 1 hour. The meeting will be recorded electronically.

Participation is voluntary
Participation in the project is voluntary. If you choose to participate, you can withdraw your consent at any time without giving a reason. All information about you will then be made anonymous. There will be no negative consequences for you if you choose not to participate or later decide to withdraw.

Your personal privacy – how we will store and use your personal data
We will only use your personal data for the purpose(s) specified in this information letter. We will process your personal data confidentially and in accordance with data protection legislation (the General Data Protection Regulation and Personal Data Act).
The recorded meeting will be transcribed and the transcription will be limited to the Component Team and their supervisors.
The data that includes PII will be stored on Google drive and access to it will be restricted using Google Drive authentication and authorization system. The transcript of the meeting will be anonymized by replacement of the names of the participants with pseudonyms and care will be taken to ensure that other information in the meeting that could identify the participant is not revealed. The anonymized meeting transcript may be used in academic publications which includes the Component Team's master's theses.

What will happen to your personal data at the end of the research project?
The project is scheduled to end before 1st of January 2022. Any data containing PII will only be stored until 1st of January 2022.

Your rights
So long as you can be identified in the collected data, you have the right to:

- access the personal data that is being processed about you

- request that your personal data is deleted
- request that incorrect personal data about you is corrected/rectified
- receive a copy of your personal data (data portability), and
- send a complaint to the Data Protection Officer or The Norwegian Data Protection Authority regarding the processing of your personal data

What gives us the right to process your personal data?
We will process your personal data based on your consent.

Based on an agreement with the University of Oslo, NSD – The Norwegian Centre for Research Data AS has assessed that the processing of personal data in this project is in accordance with data protection legislation.

Where can I find out more?
If you have questions about the project, or want to exercise your rights, contact:

- [insert contact information]

Yours sincerely,

The Component Team and their supervisors

Consent form

I have received and understood information about the project *Shared Component Platform* and have been given the opportunity to ask questions. I give consent:

- to participate in a meeting
- for my personal data to be processed outside the EU

I give consent for my personal data to be processed until the end date of the project, approx. *1st of January, 2022.*

(Signed by participant, date)