Deliverable D 4.2-3-4

Proposal for inclusion of KPIs, Graphical User-interface, and Technical Design of the Integrated Decision Support System
DISCLAIMER

The opinion stated in this report reflects the opinion of the authors and not the opinion of the European Commission.

All intellectual property rights are owned by the ECODISTR-ICT consortium members and are protected by the applicable laws. Except where otherwise specified, all document contents are: “© ECODISTR-ICT project - All rights reserved”. Reproduction is not authorised without prior written agreement.

The commercial use of any information contained in this document may require a license from the owner of that information.

All ECODISTR-ICT consortium members are also committed to publish accurate and up to date information and take the greatest care to do so. However, the ECODISTR-ICT consortium members cannot accept liability for any inaccuracies or omissions nor do they accept liability for any direct, indirect, special, consequential or other losses or damages of any kind arising out of the use of this information.
ACKNOWLEDGEMENT

This document is a deliverable of the ECODISTR-ICT project, which has received funding from the European Union’s Seventh Programme for research, technological development and demonstration under grant agreement No 608913.
# PROJECT DATA

<table>
<thead>
<tr>
<th>Project Acronym</th>
<th>ECODISTR-ICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Full Title</td>
<td>Integrated decision support tool for retrofit and renewal towards sustainable districts</td>
</tr>
<tr>
<td>Call Identifier</td>
<td>FP7-2013-NMP-ENV-EeB</td>
</tr>
<tr>
<td>Grant Agreement</td>
<td>608913</td>
</tr>
<tr>
<td>Coordinator</td>
<td>VITO (BE) Vlaamse Instelling voor Technologisch Onderzoek nv</td>
</tr>
</tbody>
</table>
| Consortium partners | SP (SE) Sveriges Tekniska Forskningsinstitut ab  
TNO (NL) Nederlandse organisatie voor Toegepast Natuurwetenschappelijk Onderzoek  
VABI (NL) Vabi Software bv  
CSTB (FR) Centre Scientifique et Technique du Batiment  
TNO (NL) Nederlandse organisatie voor Toegepast Natuurwetenschappelijk Onderzoek  
OMGEVING(BE) Multiprofessionele Ontwerpers- en Architectenvennootschap cvba  
STRUSOFT (SE) Structural Design Software in Europe ab  
WHITE (SE) White Arkitekter Aktiebolag  
SIGMA (FR) Sigma Orionis sa  
BIPOLAIRE (ES) Bipolaire Arquitectos slp  
ARUP (NL) Arup bv |
| Funding Scheme | Collaborative Project |
| Project Duration | 36 months |
| Starting Date | 01/12/2013 |
| Website | http://ecodistr-ict.eu/ |
## DELIVERABLE DATA

<table>
<thead>
<tr>
<th>Title</th>
<th>Technical Design of the Integrated Decision Support System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Deliverable D 4.2, 4.3 and 4.4 from WP4 – Integrated Decision Tool</td>
</tr>
<tr>
<td>Lead Beneficiary</td>
<td>TNO</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Holy Andrianantenaina, Hugo Vivier (Sigma Orionis), Fredrik Ståhl (SP), Johnny Kronvall, Andreas Rudenå (Strusoft), Esra Bektas, Walter Lohman, Bart Luiten, Sander Schaminee, Eric Willems (TNO), Wim Plokker (Vabi)</td>
</tr>
<tr>
<td>Contributors</td>
<td>All beneficiaries</td>
</tr>
</tbody>
</table>
| Reviewers | Chapter 2: Blanca Pedrola (Bipolaire)  
Chapter 3, 4 and 5: Rasmus Einarsson (SP)  
Chapter 4 and 5: Olivier Tournaire (CSTB) |
| Deliverable Nature | R – Report |
| Dissemination level | PU - Public |
| Submission date | 31st May 2014 |
| Status | Final version for submission to EC |

## DOCUMENT HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author/Reviewer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>30th May, 2014</td>
<td>See above</td>
<td>Final version for submission to EC</td>
</tr>
</tbody>
</table>

## ABSTRACT

Deliverable D4.234 describes the Technical Design of the Integrated Decision Support System (IDSS) that will be developed and tested in ECODISTR-ICT. It elaborates further on D4.1, the Functional Design of the IDSS. It describes how ECODISTR-ICT includes Key Performance Indicators (KPIs) in the decision process, the Graphical User Interface for and the Technical Design of the Dashboard that controls the IDSS, and the Technical Design of the Framework that integrates the Dashboard with the Databases and Modules used in the decision process.
EXECUTIVE SUMMARY

The objective of the ECODISTR-ICT project is to develop an Integrated Decision Support System (IDSS) for sustainable retrofitting of urban districts with a focus on energy efficiency.

Deliverable D4.234 describes the Technical Design of the Integrated Decision Support System (IDSS) that will be developed and tested in ECODISTR-ICT. It elaborates further on D4.1, the Functional Design of the IDSS. It describes how ECODISTR-ICT includes Key Performance Indicators (KPIs) in the decision process, the Graphical User Interface to and the Technical Design of the Dashboard that controls the IDSS, and the Technical Design of the Framework that integrates the Dashboard with the databases and Modules used in the decision process.

The results described in this Deliverable will be the basis for the upcoming tasks in ECODISTR-ICT. Developing the IDSS and its components will be an iterative process, in which relatively small development steps are followed by tests with users. First, internal project partners and later also external stakeholders in the use cases will give feedback. This feedback will be taken into account for the next development steps.
# TABLE OF CONTENT

- EXECUTIVE SUMMARY
- TABLE OF CONTENT
- GLOSSARY / LIST OF ACRONYMS
- LIST OF FIGURES
- LIST OF TABLES

1 **INTRODUCTION**
   - 1.1 **PURPOSE, INTENDED AUDIENCE AND SCOPE**
   - 1.2 **APPLICABLE DOCUMENTS**
   - 1.3 **CONTEXT ECODISTR-ICT**
   - 1.4 **READING GUIDE**

2 **INCLUSION OF KEY PERFORMANCE INDICATORS**
   - 2.1 **INTRODUCTION**
   - 2.2 **DECISION PROCESS**
   - 2.3 **KPIs in defining to-be ambition**
   - 2.4 **KPIs in the selection of the best alternative**
   - 2.5 **Actions supported by the IDSS**
   - 2.6 **Conclusion**

3 **DESIGN OF THE GRAPHICAL USER-INTERFACE OF THE DASHBOARD**
   - 3.1 **GUI REQUIREMENTS**
   - 3.2 **CONCLUSION**

4 **TECHNICAL DESIGN OF THE IDSS DASHBOARD**
   - 4.1 **INTRODUCTION**
   - 4.2 **Dashboard server technical design**
   - 4.3 **Dashboard client application technical design**
   - 4.4 **Dashboard data model**
   - 4.5 **IMPLEMENTATION**

5 **TECHNICAL DESIGN OF THE IDSS FRAMEWORK**
   - 5.1 **INTRODUCTION**
   - 5.2 **TECHNICAL DESIGN OF THE IDSS FRAMEWORK**
   - 5.3 **IMPLEMENTATION**
   - 5.4 **PLANNING AND SCOPE OF WORK FOR IMPLEMENTING THE IDSS FRAMEWORK**

6 **CONCLUSIONS**

ANNEX A – MOCK-UP OF GRAPHICAL USER-INTERFACE FOR DASHBOARD
### GLOSSARY / LIST OF ACRONYMS

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative</td>
<td>A possible solution, represented by a set of related values for variables. Typically, a decision is based on a comparison of the KPI scores of alternatives in several contexts.</td>
</tr>
<tr>
<td>Ambition</td>
<td>A quantified vision that includes a target for each KPI (both the value and the time when to reach it)</td>
</tr>
<tr>
<td>Assessment</td>
<td>Quantification of the KPIs of an alternative within contexts.</td>
</tr>
<tr>
<td>BIM</td>
<td>Building Information Model</td>
</tr>
<tr>
<td>Calculation Module</td>
<td>Computer application that calculates the score of one or more KPIs</td>
</tr>
<tr>
<td>Component</td>
<td>A component is a smallest independent part of the IDSS-Framework that performs a specific task within a Module or in the IMB communication framework.</td>
</tr>
<tr>
<td>Context</td>
<td>A possible future development outside the span of control of the users, represented by a set of related values for aspects. Alternatively defined as a future situation which is determined by the external factors that people cannot influence from within the district (e.g. Number inhabitants will double by 2030; Industry outside district can deliver 2 MW heat; Energy price: annual increase of 4%)</td>
</tr>
<tr>
<td>Context factors</td>
<td>External factors you cannot influence from within the district (e.g. Number inhabitants will double by 2030, Industry outside district can deliver 2 MW heat, Energy price: annual increase of 4% etc.)</td>
</tr>
<tr>
<td>Decision Support Tool</td>
<td>Abbreviated as DST, the part of the whole IDSS system (see Module)</td>
</tr>
<tr>
<td>Design proposal</td>
<td>A strategically combined set of measures, specified in the district (i.e. 5m² PV on building X, 10m² on building Z, no thermal grid, all single pane glazing replaced)</td>
</tr>
<tr>
<td>District</td>
<td>A subdivision of a city or municipality. In this project we expect a district to encompass +- 1000 buildings.</td>
</tr>
<tr>
<td>DoW</td>
<td>Description of Work, our project description which is part of our contract with the EU</td>
</tr>
<tr>
<td>End-user</td>
<td>Someone using the IDSS, e.g. a trained city planning professional</td>
</tr>
<tr>
<td>ESCO</td>
<td>Energy service contracting company</td>
</tr>
<tr>
<td>Evaluation</td>
<td>The process of comparing the assessment score of a variant with the targets set in the ambition</td>
</tr>
<tr>
<td>Framework</td>
<td>Platform for integration of system components, e.g. based on Inter Model Broker</td>
</tr>
<tr>
<td>Functional components</td>
<td>Functional description of a decision support element that supports one (or more) elements of a decision process</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User-Interface</td>
</tr>
<tr>
<td><strong>HVAC</strong></td>
<td>Heating, ventilation and air-conditioning systems</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>ICT</strong></td>
<td>Information and communication technologies</td>
</tr>
<tr>
<td><strong>IDSS</strong></td>
<td>Integrated Decision Support System, a system that supports the decision making process in district renovation</td>
</tr>
<tr>
<td><strong>IDSS Framework</strong></td>
<td>Linking the components of the system, such as the Dashboard, data management Modules and calculation Modules</td>
</tr>
<tr>
<td><strong>IDSS Dashboard</strong></td>
<td>The interface of the end-users to the IDSS and enables them to use the IDSS in their decision processes</td>
</tr>
<tr>
<td><strong>IMB</strong></td>
<td>Inter Model Broker</td>
</tr>
</tbody>
</table>

**Interface**

In the context of computers, three common interface types are: Hardware, Software and User interface. In technical contexts, interface requirements are governed by standard conventions called protocols which could be so complex that the interface itself is considered a separate device or system.

**KPI**

Key performance indicator, a measurable indicator for one aspect of the users’ ambition

**KPI Value**

Required or predicted score of a KPI

**Measure**

A punctual intervention (i.e. window replacement; PV panels installation; building demolition)

**Mobile client**

Any software that can run on a mobile device

**Model**

A model performs calculations of KPIs as a simplified view of reality. It makes use of data input and creates data as output.

**Module**

An independent functional part of the IDSS. A Module can consist of several coherent components e.g. calculation model, IMB client, data storage, assessment of variants visualization (in order to gather, create, change, present, and/or analyse data)

**Protocol**

Set of agreed upon, and openly published and distributed, computer interaction standards that enables different firms to manufacture compatible devices to the same specifications

**Prototype**

Pre-production model of a product, engineered for full service test. Changes based on test results are incorporated into the prototype which undergoes the same tests again. On achieving the desired results, the product is approved for volume production.

**Required functionality**

User perspective: general - case specific decision process (WP1, 5) Functional integration perspective: Modules (WP3)

**Semantics**

A conceptual data model in which semantic information is included. This means that the model describes the meaning of its instances. Such a semantic data model is an abstraction that defines how the stored symbols (the instance data) relate to the real world.

**Stakeholder**

A person, group or organization that has interest or concern in an organization. Stakeholders can affect or be affected by the organization’s actions, objectives and policies.
<table>
<thead>
<tr>
<th>System architecture</th>
<th>Framework + components</th>
</tr>
</thead>
<tbody>
<tr>
<td>System component</td>
<td>Realized functional component, e.g. a calculation Module</td>
</tr>
<tr>
<td>Toolbox</td>
<td>It refers to the IDSS that is developed in ECODISTR-ICT seen from the perspective of the end-user</td>
</tr>
<tr>
<td>Variable</td>
<td>An aspect on which a decision can be made</td>
</tr>
<tr>
<td>Variant</td>
<td>A combination of a design alternative within a context; typically, a Module analyses one variant, resulting in a predicted score of KPIs</td>
</tr>
<tr>
<td>Vision</td>
<td>A leitbild or desired end situation</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1  Work package structure of ECODISTR-ICT (DoW); this deliverable is part of WP4 ........ 14
Figure 2  Process steps and Tasks of WP4 ................................................................................ 15
Figure 3  Phase A: Defining the Ambition ............................................................................... 17
Figure 4  Phase B: Selection of the best alternative ............................................................... 17
Figure 5  The inclusion of KPIs in analyse problem step (step A1) ........................................ 19
Figure 6  The inclusion of KPIs in data collection (step A2) .................................................. 20
Figure 7  The inclusion of KPIs in Assessing AS-IS (step A3) ................................................. 20
Figure 8  Prioritizing and define target values for KPIs (step A4) ......................................... 21
Figure 9  Developing alternatives (step B1) ......................................................................... 22
Figure 10 The alternatives are assessed against the AS-IS and TO-BE situation (step B2) ...... 22
Figure 11 The best alternative is selected (step B3) ............................................................. 23
Figure 12 The legend to read the IDSS supported actions in the decision process ................. 24
Figure 13 The IDSS in Phase A, decision steps A1~A4.......................................................... 24
Figure 14 The IDSS in Phase B, decision steps B1~B3 ......................................................... 25
Figure 15 Dashboard system architecture .......................................................................... 31
Figure 16 Server processes used to scale up and control the load of the Dashboard server ...... 32
Figure 17 Some of the processing needs and functionality is isolated on dedicated processes, such as parsing files during import or preparing data for visualisation in the Dashboard client ................................................................. 33
Figure 18 Dashboard application layers with distributed MVC ............................................. 34
Figure 19 Data model of the Dashboard ............................................................................... 36
Figure 20 Schematic overview of the IDSS and its main components .................................. 41
Figure 21 The IMB Hub as the integrating component in the IDSS ....................................... 42
Figure 22 Architecture overview of the IDSS Framework ...................................................... 43
Figure 23 An example of multiple Hubs and federations ..................................................... 44
Figure 24 Possible technical options for implementation of an IMB processor .................. 46
Figure 25 API inputs and outputs of the IMB ....................................................................... 48
Figure 26 Example of a possible hardware setup of the IDSS Framework ........................... 49
LIST OF TABLES

Table 1  General support of HTML5 and related techniques ....................................................... 37
Table 2  Support for WebGL in common browsers ................................................................. 38
1 INTRODUCTION

1.1 PURPOSE, INTENDED AUDIENCE AND SCOPE

The purpose of this deliverable is to define the technical design of the Integrated Decision Support System (IDSS) in ECODISTR-ICT. The participants in the ECODISTR-ICT project are the primary intended audience. This document helps us to agree upon a more detailed scope of the deliverables of the project, and to plan our activities. The deliverable is also intended for parties in the close vicinity of the project, such as the stakeholders in the case studies, the EC reviewing committee and other interested people, to manage their expectations.

The scope of this Deliverable is the Technical Design of an Integrated Decision Support System that can be used in complex, multi-stakeholder, multi-level decision making processes with a strong geographical focus, such as retrofitting urban districts with a focus on energy efficiency. The IDSS will be applied in the use cases of ECODISTR-ICT. Which software Modules will be applied in the use cases (e.g. for analysing energy efficiency of design alternatives, or sharing geographical data), will be determined later in the project.

1.2 APPLICABLE DOCUMENTS

- ECODISTR-ICT Description of Work (DoW, d.d. 3rd December 2012)
- ECODISTR-ICT Deliverable D1.1 List of specifications for the decision tool in function of stakeholder input (d.d. 28th February 2014)
- ECODISTR-ICT Deliverable D4.1 Functional Design of the IDSS (d.d. 31th March 2014)
- ECODISTR-ICT Task plan Task 4.234 (d.d. 14th April 2014)

The main documents can be found on http://ecodistr-ict.eu/.

1.3 CONTEXT ECODISTR-ICT

This deliverable is the second result of work package 4 (WP4), Integrated Decision Support System of the ECODISTR-ICT project. Figure 1 shows the work package structure of ECODISTR-ICT, in which WP4 integrates the results of WP1 Stakeholders and Scenarios, WP2 Methods for Data Collection and WP3 Support Modules.

---

1 The Description of Work (DoW) uses the term Integrated Decision Tool. The term Integrated Decision Support System describes the result of work package 4 (WP4) better. From now on, this term will be used in this Deliverable.
WP4, Integrated Decision Support System, contains four tasks:
1. Task 4.1 Translation of user requirements into desired functionality of the IDSS
2. Task 4.2 Development of a prototype of Integrated Decision System
3. Task 4.3 Application and implementation in case studies

WP4 is organised in tasks and steps, as shown in Figure 2. As can be seen in the Figure, WP4 will develop the IDSS Dashboard and the IDSS Framework. The IDSS Dashboard supports the end-user of the IDSS to follow the decision process. WP2 and WP3 will develop the data collection and support Modules. The IDSS Framework integrates the Dashboard, the data storages and the Modules.

This Deliverable is the result of the third process step in Task 4.1: Define Technical Design. It builds on Deliverable D4.1, Functional Design of the IDSS. It describes the Technical Design of the IDSS. It describes how ECODISTR-ICT includes Key Performance Indicators (KPIs) in the decision process, the Graphical User Interface to and the Technical Design of the Dashboard that controls the IDSS, and the Technical Design of the Framework that integrates the Dashboard with the databases and Modules used in the decision process.
The Description of Work mentions three separate Deliverables which are combined in this Deliverable D4.234: D4.2, Technical Design of the IDSS, D4.3, Design Graphical User Interface, and D4.4, Process for Inclusion of KPIs.

1.4 **READING GUIDE**

This document should be read completely by the ECODISTR-ICT participants. Others should start with the Executive Summary and this Introduction, and from there select those Sections that are relevant to them.

Chapter 1 introduces the document.

Chapter 2 describes how we include Key Performance Indicators (KPIs) in the decision making process and the IDSS.

Chapter 3 describes the design of the Graphical User Interface (GUI) of the Dashboard.

Chapter 4 describes the technical design of the IDSS Dashboard. The Dashboard is the interface of the end-users to the IDSS and enables them to use the IDSS in their decision processes.

Chapter 5 describes the technical design of the IDSS Framework.

Chapter 6 concludes this Deliverable D4.234, Technical Design of the IDSS.
2 INCLUSION OF KEY PERFORMANCE INDICATORS

2.1 INTRODUCTION

This Chapter corresponds to the Subtask 4.2.3: Inclusion of Key Performance Indicator set (KPI’s) for decision making. It focuses on how the decision makers use KPIs and include them in the decision process. Consequently, the Chapter also deals with the actions that need to be supported in the IDSS. It defines the process of KPI use by the users of the IDSS and sheds a light on the functionality and the technical design of the IDSS components. These components will be elaborated further in Chapters 3, 4 and 5.

The Chapter starts with highlighting the decision process defined in Deliverable D4.1. It then elaborates on the two phases of the decision making; A) defining the ambition, and B) selection of the best alternative. In each phase, the actions of the users in relation with the use of KPIs are elaborated. This chapter thirdly connects these actions to the IDSS components and attempts to illustrate which actions are supported by the IDSS. This connection will establish a bridge to the visual and technical design of the Graphical User Interface (GUI), and the IDSS Framework.

2.2 DECISION PROCESS

The decision process is structured in seven steps as described in Deliverable D4.1, Functional Design of the IDSS. The first four steps, A1-A4, describe the first phase [Phase A], where the ambition of the project is defined (See Figure 3). The last three steps, B1-B3, describe the second phase [Phase B], where the selection of best alternative is made (see Figure 4).

Phase A is initiated by the high-level ambitions/choices of the stakeholders. This high-level ambition is formulated into a ‘problem statement’. In order to analyse problem, the problem substructures (problem trees) are explored. KPIs are used for identifying and exploring these substructures. KPIs therefore represent different angles (or lenses of the stakeholders) to look at the problem. Data collection is needed to calculate KPIs by the selected calculation Modules. Through calculation, the stakeholders assess the AS-IS situation. And based on that assessment, the actors identify where the improvement should be and establish a TO-BE ambition, for each KPI. Defining your TO-BE ambition therefore becomes setting target values for each KPI.
After defining the TO-BE ambition, the next three steps (B1-B3) help decision makers in selecting the best alternative to reach their ambition (see Figure 4), and consequently solving the problem.

Phase B is initiated through the gap between TO-BE and AS-IS (quantified by KPI target values and their AS-IS scores). Design teams work towards different alternatives in which they incorporate selected measures for each KPI and try to meet the TO-BE ambition. Design Modules are used to develop these alternatives. As alternatives refer to long periods of time for the investments (see program and project level decision process in D4.1), the decision makers take different contexts (predicted yet uncontrolled future situations) into account when assessing the alternatives. In other words, the alternatives are positioned into different contexts. A context is a possible future development outside the span of control of the stakeholders. An alternative positioned in one context is called a variant. The scores of each variant are acquired through calculation Modules.
The decision makers then make a well-founded decision. Not only by selecting the alternative that scores best in the most likely future. But also by assessing the alternatives in other contexts and using this information to select the most robust alternative. This is also an iterative process, as the results in B2 phase can be used for re-formulation of the TO-BE ambition and the decision process needs to be re-done.

The next two sections elaborate these seven basic process steps further by focusing on the role of Key Performance Indicators (KPIs) in decision making.

## 2.3 KPIs in Defining TO-BE Ambition

In this section, KPIs in Phase A, Defining TO-BE ambition, are elaborated. This elaboration is done per decision step.

### 2.3.1 Analyse problem, step A1

As indicated in Section 2.2, analyse problem is derived from the high-level ambitions of decision makers and requires identification of the substructures the problem. KPIs in this step are used to identify and analyse the problem substructure. The decision makers select KPIs that become relevant to their process (i.e. individual buildings, block of buildings, the whole district). They identify different angles in order to assess AS-IS via KPIs as being quantitative (i.e. Operational Costs, Energy Consumption, Energy Bill, CO2 emission, Total Area of Use), and/or qualitative (i.e. End-user Satisfaction, Aesthetic, Safety, Comfort, and Health). In both cases, each KPI requires prioritization, scoring, and weighting. Regardless of being qualitative or quantitative, each KPI also requires assessment which either becomes calculation and/or expert judgement, depending on how automated the assessment can be performed.

The IDSS will support the following actions in this step (see Figure 5). These actions are not solely about content-related (KPIs inclusion) but also administrative that helps the user to capture the information and make it explicit for retrieving the information later.

The first action supported by the IDSS is administrative. It deals with 1) capturing the users of the IDSS that initiates the decision process, 2) pinpointing the geographical area, and 3) defining the problem that decision makers phrase at the beginning. The IDSS can make evident that who the user/facilitator of the IDSS is and which stakeholders are involved at that moment, and what the phrased/contextualized problem is.

The second action supported by the IDSS is content-related. It deals with enabling the user to define and select KPIs which becomes necessary to identify the extent of the problem. The IDSS will provide three choices to select KPIs:

1. Choose a certified full KPI set (i.e., from BREEAM, Open House, and Super Buildings);
2. Select KPIs from existing sets;
3. Adding new KPIs (when existing KPIs do not satisfy the specific needs of the stakeholders for the specific case).
As the decision steps are highly iterative, more KPIs can emerge in later steps (i.e. after the data collection starts, after assessing AS-IS). The IDSS will also support the user while KPIs are defined, weighted, and valued iteratively.

The third one is content-related and it deals with management of the content of KPIs and is to determine the scores of KPIs, to prioritize/weight KPIs and to select calculation Modules which will be registered in the IDSS prior to data collection. Depending on how the user forms its KPI set, the IDSS will display when/whether scoring of KPIs can to be defined or altered. When possible, the user will define parameters (through determining minimum and maximum values), weight and select calculation methodology in relation with each KPI (i.e. selecting a calculation Module in the IDSS or referring to an expert opinion).

Figure 5 The inclusion of KPIs in analyse problem step (step A1)

2.3.2 Collect data, step A2

KPIs in collect data step deal with understanding what data needs to be collected as well as acquiring relevant, sufficient and credible data so that calculation Modules for KPIs can be executed in the next step, Assessing AS-IS. The data needed for selected KPIs can differ due to the calculation methodology (formulas). And the methodology can vary depending on the geographical location which might impose using certain norms, reference regulations for calculation Modules used for KPIs. In this step, there is also the relevant contexts are created. These contexts are necessary to calculate KPI scores as contexts refer to sets of variables which do not have fixed and/or certain hard figures yet they change over time outside of the control of the stakeholders (i.e. energy prices, interest rate). Assessment of AS-IS (and also alternatives in the B2 step) requires these contexts in order to make a well-founded assessment and decision.

The IDSS will support the following actions in this step (see Figure 6). The first one is to inform the user about which data needs to be collected for the selected KPIs. The data can be collected either manually by the user or by semi/full-automatically by the IDSS depending on the capability of the IDSS. In either way, the user will be informed how detailed the data is needed for the calculation due to the selection of the calculation Modules. The second one is to create contexts and variables to be filled in definition of the contexts. The third one is to enable the user to upload the data that will be used for the calculation and check the consistency/sufficiency of the data uploaded.
2.3.3  Assess AS-IS, step A3

KPIs in assess AS-IS step deal with knowing the state of the art of the problem using the scores of KPIs in the AS-IS situation. This means that the KPI scores are calculated with that assumption that the current policy is continued through the assessment period. In city planning this is often called the Zero-alternative, indicating the most probable development if the project is not executed.

The IDSS will support the following actions in this step (see Figure 7). The first one is to run calculation Modules to assess AS-IS of each KPI. When calculation Modules are registered to the IDSS, the users can access to the system via the Dashboard. When Modules are not registered (i.e. in case the users created new KPI which is not supported by any Modules), the users can run a calculation Module separately and enter the values manually to the Dashboard. In case of a newly added qualitative KPI (i.e. aesthetic, uniqueness etc.), the users find alternative ways such as consulting experts, conducting workshops with expert panels etc. In other words, calculation Modules produce the score of the selected KPIs, which result in AS-IS situation. AS-IS situation becomes a basis for establishing TO-BE ambition which has more focus and boundaries comparing to the high-level ambition.

2.3.4  Define TO-BE ambition, step A4

KPIs in the define TO-BE ambition step deal with using the calculated KPI scores as a basis to elevate the AS-IS situation. After assessing the AS-IS situation, decision makers acquire better understanding on 1) where real problems are situated (i.e. high energy consumption due to thermal bridges in existing apartment blocks in the district) and 2) where the room for improvement should/can be (i.e. reduce energy consumption, yet keeping the green level as it is). Therefore in this step, the scores of KPIs become important reference points in order to define the
TO-BE ambition. The TO-BE ambition consists of KPI target values which are defined in relation with the priority level and the boundary conditions of the stakeholders.

The IDSS will support the following actions in this step (see Figure 8). The first one is to visualize AS-IS scores in each KPIs and provide a room to fill-in KPI target values for the TO-BE ambition. The second one is to enable the user to re-prioritize KPIs and to define target values for each KPI. The definition of target values can be filled manually or by using particular Modules depending on the technical functionality of IDSS. The user, if necessary, can re-select KPIs, calculation Modules, weight & priority of KPIs and get new results. Based on new results, the user can re-define target values. The IDSS allows the user for this iteration.

2.4 KPIs in The Selection of the Best Alternative

In this section, KPIs in Phase B, the selection of the best alternative, are elaborated. This elaboration is done per decision step.

2.4.1 Develop alternatives, step B1

KPIs in develop alternatives step deal with a process where the design team works towards several solutions which incorporate several measures in order to meet the KPI target values. The selection of which measures need to be taken into account is embedded into the practice of the design team. Towards the measures, the design team creates different alternatives which then will be assessed against KPIs. Regarding the use of KPIs, this step is relatively passive compared to the steps in Phase A explained by now.

The IDSS will support the following actions in this step (see Figure 9). The first is to inform the user and the design team about which measures can be taken in relation with the selected KPIs. For example, when the height of the energy bill is selected as a KPI, the relevant measures would be insulating the building envelope, changing the heating system, or installing PV panels. The IDSS can provide lists of possible measures the design team can take into account when developing alternatives. The second is to administer alternatives (design proposals) created by the design
team. These alternatives refer to sets of data which KPI calculation Modules will access to run the calculation. Similar to the registered calculation Modules, design Modules can be registered to the IDSS. In this case, the IDSS can become more active, as the users can directly access these design Modules via the IDSS. The possibility of the access to these licensed programs will be defined in technical design. The third action is to create variants by selecting and modifying the context in which the design alternative will be positioned. In other words, the preparation for the assessment of alternatives is done through administering, uploading the design alternatives and selecting the context. Each alternative will be assessed by positioning it in different contexts.

![Figure 9 Developing alternatives (step B1)](image)

### 2.4.2 Assess alternatives, step B2

KPIs in the assess alternatives step deal with calculation of the KPI scores of the created variants and gathering all the results to compare 1) variants among themselves and 2) variants against TO-BE ambition prior to the selection of the best alternative and context.

The IDSS will support the following actions in this step (see Figure 10). The first one is to run calculation Modules. The second one is to visualize the scores of KPIs of AS-IS, TO-BE ambition and Variants. The IDSS will enable different visualization. If needed, the set of KPIs and their target values can be revised. At this stage it is also possible to use qualitative methods (i.e. expert opinion, workshops etc.) to assess and rate the different alternatives. It is also possible to identify explicit trade-offs in the alternatives and adjust the assessment accordingly.

![Figure 10 The alternatives are assessed against the AS-IS and TO-BE situation (step B2)](image)
2.4.3 Select best alternative, step B3

KPIs in the select best alternative step deal with making the final choice for an alternative based on the KPI scores of the variants. This results in selecting the best and most robust alternative.

In this step, the IDSS will provide a ‘scoreboard’ (see Figure 11). This scoreboard will become a summary screen that will enable the user to make comparison between variants in relation with the AS-IS and the TO-BE ambition. For example, the scores of an alternative in several contexts can be compared, or, the scores of the alternatives in one context. This allows the end-users to create a solid foundation for their decision.

At this stage, implicit KPIs might emerge (i.e. aesthetic, personal values attached to a certain alternative or specific design character of a particular alternative). The user can decide whether or not these emerging KPIs should be included, and be recorded in the decision process. This means that the users might need to go back to the previous steps. After the alternatives are assessed and revised, the IDSS will visualize which of the alternative is best suited to fulfil the TO-BE ambition based on personal choices of the stakeholders that are administered up until this moment. However, the decision may also be to redo all or part of the decision making process or perhaps revise the ambition of the project.

![Figure 11 The best alternative is selected (step B3)](image)

1. Providing a ‘scoreboard’ as a summary that the final decision can be made.

2.5 Actions supported by the IDSS

The aim of the IDSS is to support the user in the decision making process. The IDSS consists of several components such as the IDSS Dashboard, which is the interface of the end-users to the IDSS, the IDSS Framework, which links the components of the system together, and the Modules. In the following Figures illustrate the actions where the inclusion of KPIs is supported by the IDSS.

Figure 12 provides the legend to read the IDSS supported actions in the decision process in Figure 13 and Figure 14. Different colours and symbols represent where actions in the decision process take place. Some of actions can be performed both outside of the IDSS and by the help of the components of the IDSS.
Figure 13 shows the actions of the first three steps (A1-A4) of the decision process but also how the basic steps are linked together in the process.
Figure 14 shows the last three steps (B1-B3) of the decision process.

![Diagram of the IDSS in Phase B, decision steps B1~B3]

2.6 CONCLUSION

This chapter focused on KPI inclusion in the decision making process. As shown, in each decision step KPIs play an important role. KPIs help decision makers to 1) understand the deepness of the problem, 2) accurately assess the state of the art situation of the problem (AS-IS), and 3) effectively identify the room for improvement. Furthermore, KPIs enable decision makers to steer their decision process towards more quantified and more objective choices, particularly in multi-stakeholder complex settings. This does not mean that only quantitative KPIs are to be used and supported in the IDSS. It becomes also important to identify qualitative KPIs and make them explicit in the decision process, even though there may not be an automated calculation Module (yet). Both quantitative and qualitative KPIs will be supported in the IDSS.

This chapter provides crucial input to the technical and user-interface design of the Dashboard and the Framework. The design of the Dashboard and the Framework need to meet the expectations from KPI inclusion in the decision process.
3 Design of the Graphical User-Interface of the Dashboard

The process to include KPIs in the decision making process, as described in the previous Chapter, will be supported with an Integrated Decision Support System (IDSS). The IDSS consist of a Dashboard, Modules and a Framework. The IDSS Dashboard supports the end-users of the IDSS to follow and control the decision making process. It is the facade (or front-end) for the end-user. Via the Dashboard the end-users will interact with the available Modules of the IDSS. For the end-users, the integration of the Dashboard with the Modules will happen out of sight. The Framework will facilitate this integration. This Chapter describes the Graphical User-Interface (GUI) of the Dashboard. The technical design of the Dashboard and the Framework are described in Chapters 4 and 5.

The GUI is described in a functional way in a PowerPoint mock-up. This Chapter starts with the GUI requirements from the DoW and D4.1. The GUI is built up using the 7 process steps as introduced in D4.1 and extended with KPI inclusion in Chapter 2 of this document. The mock-up is provided in Annex A.

3.1 GUI Requirements

3.1.1 Requirements from the DoW

The DoW states that the GUI should aim at a good balance between usability of the tool and a profound level of technical detail. The elements and functionality of this GUI are in line with Chapter 2 and contain, among others, functionality to:

- Quickly alter input to control and follow the decision making process. This will contribute to usability of the IDSS from the end-user perspective;
- Display, select, alter variables, start calculation and show results of KPIs;
- Input of geo-data by referring to existing data bases;
- Represent results graphically, both in graphs and maps;
- Display the evolution over time of costs and benefits;
- Compare the results of different variants;
- Report in the form of tables and maps.

The mock-up is partly based on already accepted dashboards from Urban Strategy and the Vabi Assets Software with all necessary ingredients already included, such as reporting in a graphical way (maps, diagrams, drawings) and in a tabular way (calculation results). Based on these, a mock-up GUI was made in PowerPoint and discussed in multiple rounds within the WP4 team.
3.1.2 Requirements from D4.1

In Deliverable D4.1 the requirements from the DoW were further specified in a Functional Design. Two main users of the Dashboard were identified: The facilitator and the stakeholder. In D4.1 the required functionality was structured according to the seven basic steps of decision making (see also Figure 3 Figure 4 of this Deliverable). Chapter 2 elaborated this by describing how KPIs are included in these seven steps.

3.2 CONCLUSION

In Annex A, the mock-up of the GUI is given for the seven process steps. The mock-up is self-explanatory. As can be seen, the KPIs play an important role in the decision making process.

The mock-up of the GUI is a starting point for further development. From this mock-up, prototypes will be developed and tested on the stakeholders in the pilot phases. With their feedback the GUI will be adjusted. Possibly, the layout of the GUI will be changed based on technical feasibility and multi-platform compatibility considerations.

We expect discussion between software developers and end-users on at least two subjects: aggregation and contexts.

When looking at KPIs in a multi-stakeholder process, such as urban renovation, it should be possible to zoom in or out to different aggregation levels, both for maps and tables. Possible aggregation levels are: 1) individual buildings, 2) block of buildings and 3) the whole district. Assessing different aggregation levels is important for purposes of analysing and informing both individual owners/renters and stakeholders at district level. Not all KPIs will be suitable to be displayed at these aggregation levels. Depending on the functionality of the Modules, the aggregation could be done in the Modules and/or in the Dashboard.

For most stakeholders, comparing alternatives (for the choices to be made by the stakeholders) by using scoring predefined KPIs is common practice. However, working with contexts (for the prediction of future variables the stakeholders cannot influence) is not. Working with alternatives and contexts gives stakeholders the opportunity to make robust decisions. It is, however, not common practice in today’s decision processes. When interacting with stakeholders we will define to what extend and how they want to work with contexts and alternatives.
4 TECHNICAL DESIGN OF THE IDSS DASHBOARD

This Chapter describes the technical design of the IDSS Dashboard. It builds on the functional design of the Dashboard, as described in Chapter 4 of Deliverable D41, and on design of the Graphical User-Interface of the Dashboard, as described in Chapter 3 and Annex A.

It starts with introducing the technical design of the Dashboard in a general fashion. It explains the technical approach and summarizes the Chapter for participants with less background in the web development field.

The next part of Chapter will go through the overall system architecture of the Dashboard, which is divided into a server environment and client applications. The server environment will be used to communicate with the IDSS Framework and the dedicated database for the Dashboard. The end-users will use Dashboard clients on multiple platforms, connected to the server via the Internet. Also the data model of the Dashboard is discussed.

The latter parts of the Chapter include the implementation choices relevant for the software implementers in the project.

4.1 INTRODUCTION

As explained before the IDSS consists of a Dashboard, Modules and a Framework. Most Modules and the Framework already exist and will be altered for use in this project. The IDSS Dashboard will be developed from scratch.

4.1.1 Web-based

Since decision support depends on understanding complex relations between different views in the decision process as well as handling a lot of data for different kinds of visualisation, require us to create a modern and powerful tool, accessible for use in communication and sharing via Internet. This will require the use of the state-of-the-art technologies. One of the major challenges for the Dashboard is to combine these modern technological possibilities with mobile devices that generally lack the same support and processing power that of a personal computer. Another challenge is to efficiently handle the vast amount of network traffic being pushed to and from the Dashboard client. This includes the data for calculations and processing that generally is done externally from the Dashboard client, and often handed over to a dedicated Module in the IDSS Framework.

To overcome these challenges, the IDSS Dashboard will be a web-based responsive application that will be accessible on devices using a web browser such as computers, tablets and mobile phones. One of the advantages of creating an application within a browser environment is that the application is platform independent, and is relying on the vendors of the browser to follow the
web standard specifications from W3C\textsuperscript{2}. The support for web standards is constantly increasing and today the web browsers are a great and mature choice when creating cross-platform applications. In recent years the market together with the developer community has driven forward the implementation of the next generation of web technology often referred to HTML5. This strongly increases the possibility to create powerful fast applications in the browser with techniques such as streaming, animations, local data storage, performance boost using both several cores on the CPU as well as hardware accelerated 3D, among others.

4.1.2 Technical demands of the Dashboard

The following list of functionality is connected to the technical requirements of the Dashboard. It is followed by a summary of the technical approach.

a) The Dashboard needs to respond to user interaction with the system creating new events for the IDSS Modules to react to. The system needs to notify the user on the progress so that the user knows what happens in the background. A typical example of this is the procedure of starting up the KPI calculations for a variant and to notify the user when these calculations are finished.

b) The result of the Modules should be allowed to be extensive when it comes to result data and visualization contexts. This puts a demand on the server and the client not to be unresponsive during processing or loading data. A typical example of this is when the user needs to get a deeper understanding of what input is given to a calculation or what the result score really means when it comes to output data.

c) The user needs to save and refer back to different variants during the decision process, which implicates a persistent storage connected to the Dashboard. Connected to this storage functionality comes a need to define a data model dedicated to the Dashboard that only handles the users and their instantiated decision processes.

d) User and session management is needed to handle secure authentication and authorization.

e) File upload and download functionality is required to save and later import project data. In the beginning of the development phase this will be the only way to persist work between working sessions. Imported data is also needed for Module input in some cases.

f) Caching is needed to avoid large datasets to be downloaded several times.

4.1.2.1 Dashboard communication (a, b)

The communication from the client to the server should mostly be done through the so-called WebSocket protocol. When communicating with the IDSS Framework a special connector (called the IMB Connector as explained in Chapter 5) will be used that is compliant with this protocol to handle and convert the traffic in an efficient way. This handles the problem of notifying the user on system progress and handling large amount of data in real time. This is described in Section 4.5.

\textsuperscript{2} See http://www.w3.org/
4.1.2.2 Dashboard storage and data model (c)

Project data specific to the user’s progress in the decision support process will be saved in the Dashboard database. This includes user specific settings for KPIs and Modules, projects, variants and result data. The data model for the Dashboard is described in Section 4.4.

4.1.2.3 Dashboard intermediate format specification (c)

Any kind of data returned from Modules could be stored in the Dashboard database for later retrieval. What kind of data depends on the performance and the functionality of the system and will be elaborated during the development. This data specification needs to be agreed upon between the components that are integrated in the system. The specification is discussed in Section 4.5.2.1.

4.1.2.4 Dashboard user management (d)

User profile settings and project data will be saved for a user. This includes storing name, password and other user related information that is needed. Authentication as well as authorization needs to be handled. The security as an important concern is described in Section 4.5.3.

4.1.2.5 Dashboard import/export functionality (e)

File uploads and export functionality requires a data format schema or specification for every type of file type supported by the Dashboard. In some cases the Modules handle the validation for file uploads. For Dashboard specific data the format will confirm to the Dashboard Data Model (also Section 4.4) and the Dashboard Data Interchange Specification (also Section 4.5.2.1).

4.1.2.6 Dashboard client caching functionality (f)

Extended caching could be necessary in the client that goes beyond the default caching mechanism in the browser. When working with large datasets such as GIS data, the data can stored using the local storage functionality of a HTML5 enabled browser.

4.1.3 Integration with the IDSS Framework

The Dashboard will communicate via the IDSS Framework with the Modules creating output for the decision process. As described in D4.1, the Dashboard server and the Modules will communicate to the Framework via messages serviced by a so-called Inter-Model Broker (IMB). The IMB acts as a message hub that connects those Modules which have integrating an IMB connector (see also Chapter 5). This means that when the user requests output from the Modules using the Dashboard client, a message is sent out from the Dashboard server via the IMB connector to the relevant Modules. The IMB connector then listens to any events from the Framework that implicates new results from the Modules ready to be presented for the user in the
Dashboard client. The data storage connected to the Dashboard server is able to persistently store the user progress creating variant states that the user will refer back to during the decision making process, and communication with stakeholders.

![Dashboard system architecture](image)

Figure 15  Dashboard system architecture

4.1.4 Open source development

An important part of creating open source software is to create a foundation for developers to easily adopt. It is of great importance to follow what is considered as praxis of the community in respective field, and follow the conventions that are being formalized through popular techniques, libraries and tools. The goal of the IDSS is to create a tool that will be maintained and improved after the end of the project, thus creating a solid foundation and good basis for creating professional open source software. The principles behind the use of these libraries and tools form the basis of the technical specification of the Dashboard together with the specific technical demands that the Dashboard conforms to.

4.2 Dashboard Server Technical Design

The Dashboard is a web application and is dependent on an application server, a server API, and a dedicated database. Furthermore the server API is divided into several clustered processes to be able to scale when the load from multiple clients increase and the demand of handling large amount of data is getting more critical. The ability to scale efficiently on a server machine makes it necessary to use high-performance storage between processes to keep track of user sessions and state.
The load on the server depends on several factors such as how many concurrent users the Dashboard has and what type of processing functionality is needed. Heavy processing responsibility should not be delegated to the Dashboard server in the first place but there is still a need to handle a lot of data. The major processing functionality is the one related to reading and writing files, parsing and preparing data for visualization, and querying the database.

Since the Dashboard should be flexible to handle future needs of processing and since the major bottlenecks of the server cannot be predicted in advance, a modular approach of the server architecture is needed. In that way the processing needs and functionality can be isolated and increased on demand. This would imply that a queue for the processing jobs would be implemented to control the load on the processes.

The distribution between processes as shown in Figure 13 and Figure 14 is to be elaborated and decided upon during the implementation phase depending on the actual processing needs of the Dashboard.
4.2.1 Technical choice of platform

To be able to manage modularity and control extensive data flow, real time streaming and lots of concurrent requests, a natural choice of server platform is Node.js. The convenient fact that the server programming language is the same as on the client makes the development more efficient. This is due to the fact that in some cases it is hard to determine in advance if processing functionality should be on the server or the client with the classic considerations on whether to make the client thin or thick. A good approach is to be very flexible and be able to move functionality between the client and the server, and if the same functionality is needed on both places, use the same code base. A common problem is that when developing on a computer with first class hardware and later test the application on a mobile device, the application may perform poorly. The major approach for the Dashboard development is to try as far as possible to do processing on the client, with the fall back solution of putting the processing on the server. This processing only includes preparation of data for use in the GUI and is often a matter of trying to save bandwidth and round-trips to the server. This creates a more reactive application, but the load on the client device is increased.

4.3 Dashboard Client Application Technical Design

The client application is a self-contained web application that is distributed from the server. The application is executed in a web browser after being downloaded completely. This is also referred to a Single Page Application and means that even though the Dashboard contains several pages...
during usage, the server does not necessarily return any other mark-up, style sheets or logic than during the initialisation phase. After initialising the communication with the server the client and server internally will have full-duplex communication using WebSocket making it possible to send and receive data continuously. The data to be sent includes:

- Request and response traffic to the IMB connector (see also Chapter 5)
- Managing versions or other project specific data that is saved in the Dashboard database
- File uploads and file export

### 4.3.1 Common design patterns and principles

The most common approach today in web development when it comes to application structure is to use a variant of the MVC-pattern (Model-View-Controller). It is a principle that divides the application in different layers making it easier to collaborate and separate concerns in the code base. The approach is different between different frameworks and the basic concept for the Dashboard is that the model layer should be close to the back-end and the Module data defined in WP3, and the view layer should be separate from complex logic enabling designers and front-end developers to focus on the interaction and the look-and-feel of the GUI. The controller layer should be the glue that binds the models and the different views together. Since the Dashboard is a dependent part of a bigger picture (a complete system together with the IDSS Framework and the connected components) the model layer should be defined at the back-end of the system enabling logic and processing on the server side. This together with an eventual mapping abstraction layer for the database (ORM) is shown in Figure 18 at the two topmost layers. The ORM is used to connect to an eventual relational database (also Section 4.4).

This interpretation of MVC contradicts a common principle for single page applications, to maintain the MVC on the client and to create a server API with CRUD functionality (create, read, update, delete). The conceptual design of the Dashboard is shown in Figure 18, and shows how the model layer resides at the server side close to the database.

---

**Figure 18** Dashboard application layers with distributed MVC
4.3.2 Model-driven approach

The purpose of implementing a model-driven approach in the client is to make it easier for the GUI of the application to respond to changes from the Modules sending data and for the Dashboard to respond to user input. This is done by implementing a two-way data binding in the client, and is basically an abstraction of handling events in the controller layer and the view layer so that when the model changes the both layers are notified. An advantage with this approach is the increased possibility to collaborate between different roles in the development process since the logic is separated from the document object model (DOM) of the browser. The user interface designer and the programmer responsible for the logic layer agree through the definition of the model and work independently in separate layers of the application structure.

Data binding is implemented in several modern frameworks such as WPF, JFace and Cocoa. When it comes to JavaScript frameworks and two-way data binding there are some existing solutions of which Angular.js by Google is a popular and well-maintained alternative.

4.3.3 Selecting framework and libraries

To select framework and libraries in open source web development is an extensive process that often differs from the way of traditional platforms such as .Net or Java. Picking the solution for your applications is a modular approach for selecting smaller parts that works well together. Different kinds of package managers, such as NPM and Bower, exist to simplify this administration. With these tools the third party code does not need to be checked in into version control except for the file referring to the name and version of the different packages. This also means that a part of the development cycles and maintenance is to update several frameworks and libraries. A part of the development is to swap things out that do not perform well or misses some important features, and another natural part is to contribute to open source libraries to implement important features that are needed. When releasing a stable version of the Dashboard it is important to make sure that the versions of third party libraries are stable and that their version number is locked.

4.3.4 Distribution of code, version control and issue tracking

A GitHub account, or a similar service, will be setup to handle the distribution and versioning of the source code for the Dashboard. This approach is by far the most common in open source web development, and something that is expected for an open sourced modern web application. This creates easy access to the source code and also enables measurement of the activities and issues with the application code available for all parties involved.
4.4 DASHBOARD DATA MODEL

The data model of the Dashboard should follow the formalized entities and terminology of the decision support as far as possible. This implicates a strongly object-oriented approach that will help developers and other involved sharing the common terminology and behaviour of the decision design process all the way into the code base of the Dashboard.

![Data model of the Dashboard](image)

The data model is a subject of change in the early development phase and the database scheme should be flexible to enable modifications. The flexible nature of the input and result data from Modules is aided by a NoSQL (Not Only SQL) solution. In that way the Modules and the Dashboard can be more flexible in what kind of data structures it uses in the Dashboard than if an SQL-approach is used. This also explains that the database model could contain data types not allowed in an SQL-environment such as complex objects or arrays. The database technology could be changed later if necessary.

4.4.1 KPI management and data model

The KPI entity should be seen as an object specification external from the Dashboard. The current approach is that the KPIs will be handled by a directory Module within the Framework available for the Dashboard and for connecting Modules to the system. This is necessary for Modules to share the same KPI. The Module vendor should also be able to create new KPIs if there is no KPI that
matches the functionality of the Module. The KPI specification is also used internally in the Dashboard for the users to create custom KPIs only accessible for the individual user.

To solve the need of a KPI directory in the initial implementation phase, the Dashboard will temporarily provide this as a service. The external directory Module will be created later if there is a need for that kind of service for handling KPIs.

4.5 IMPLEMENTATION

4.5.1 Browser Support

The Dashboard should be designed with the future in mind. Modern browser support for web standards is constantly increasing and is expected to make remarkable progress during this project. The Dashboard should be considered as a next generation of web application, leaving out the support for browser not capable to handle certain techniques. The compatibility Table below gives an overall support view.

![Compatibility Table](image)

<table>
<thead>
<tr>
<th>Calculation of support for currently selected criteria</th>
<th>IE</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>IOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Opera Mobile</th>
<th>BlackBerry Browser</th>
<th>Chrome for Android</th>
<th>Firefox for Android</th>
<th>IE Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.0: 11%</td>
<td>31.0: 89%</td>
<td>90%</td>
<td>2.3: 24%</td>
<td>2.0: 31%</td>
<td>2.2: 23%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.0: 28%</td>
<td>32.0: 90%</td>
<td>90%</td>
<td>4.0: 31%</td>
<td>3.0: 43%</td>
<td>10.0: 24%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.0: 62%</td>
<td>27.0: 79%</td>
<td>33.0: 91%</td>
<td>3.2: 24%</td>
<td>4.0: 31%</td>
<td>10.0: 24%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current: 11.0: 67%</td>
<td>28.0: 79%</td>
<td>34.0: 92%</td>
<td>7.0: 70%</td>
<td>20.0: 89%</td>
<td>7.0: 67%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Near future: 29.0: 84%</td>
<td>35.0: 92%</td>
<td>92%</td>
<td>4.4: 76%</td>
<td>7.0: 67%</td>
<td>5.0: 70%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Farther future: 30.0: 84%</td>
<td>36.0: 92%</td>
<td>92%</td>
<td>10.0: 80%</td>
<td>33.0: 88%</td>
<td>26.0: 78%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 versions ahead: 31.0: 84%</td>
<td>37.0: 92%</td>
<td>92%</td>
<td>22.0: 89%</td>
<td>26.0: 78%</td>
<td>10.0: 57%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1 General support of HTML5 and related techniques

Source: [http://caniuse.com/#cats=HTML5,JS_API,SVG&statuses=All&agents=All](http://caniuse.com/#cats=HTML5,JS_API,SVG&statuses=All)
Not all techniques that affect the compatibility results in this Table will be used in the Dashboard, but the overall picture is fair in the conclusion that some browsers with version number older than current (2014-04-24) will not fully support the Dashboard. With the exception from Opera Mini all current common browsers are expected to fully support the Dashboard at the end of the project. Special attention is needed when it comes to support for 3D-graphics that is still unclear and requires a separate picture of the current state:

<table>
<thead>
<tr>
<th># WebGL - 3D Canvas graphics - other</th>
<th>*Usage stats: Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method of generating dynamic 3D graphics using JavaScript, accelerated through hardware</td>
<td>Support: 34.28%</td>
</tr>
<tr>
<td></td>
<td>Partial support: 29.71%</td>
</tr>
<tr>
<td></td>
<td>Total: 63.99%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Show all versions</th>
<th>IE</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>IOS</th>
<th>Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Opera Mobile</th>
<th>Blackberry Browser</th>
<th>Chrome for Android</th>
<th>Firefox for Android</th>
<th>IE Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0-4.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.0</td>
<td></td>
<td>31.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.0</td>
<td></td>
<td>32.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.0</td>
<td></td>
<td>27.0</td>
<td>33.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current</td>
<td>11.0</td>
<td>28.0</td>
<td>34.0</td>
<td>7.0</td>
<td>20.0</td>
<td>7.0</td>
<td>5.0-7.0</td>
<td>4.4</td>
<td>16.0</td>
<td>10.0</td>
<td>33.0</td>
<td>26.0</td>
<td>10.0</td>
<td></td>
</tr>
<tr>
<td>Near future</td>
<td>29.0</td>
<td>35.0</td>
<td></td>
<td>21.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Farther future</td>
<td>30.0</td>
<td>36.0</td>
<td></td>
<td>22.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 versions ahead</td>
<td>31.0</td>
<td>37.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Support for WebGL in common browsers

The use of 3D graphics support in the Dashboard is not a part of the current functional or technical scope, but should be considered something to strive for.

4.5.2 Dashboard Communication Protocols

The Dashboard communicates with the message hub from the IDSS Framework using an IMB Connector (see Chapter 5). Internally the Dashboard will use JSON as communication format between the client and the server. This Section describes the rules for the Modules to communicate through the Dashboard.

---

4 Source: [http://caniuse.com/#search=webgl](http://caniuse.com/#search=webgl)
The input and output of the Modules should be handled in the Dashboard application without any specific knowledge from the client about the specific Modules profile. This means that the client and the Modules connected should agree on certain data structures for dynamic generation of mark-up for the input controls and the visualisation. This also means that the Dashboard client only conforms to how the layout and components of the user interface will look and behave, not the actual data that is input or output from the Modules.

4.5.2.1 Dashboard Data Interchange Specification

The Dashboard Data Interchange Specification (DDIS) is JSON data structures defining different types of input data and output data that the Modules and KPIs, should confirm to. The user interface will be able to adjust itself to the data defined in these data structures and generate the appropriate input controller or visualization.

The DDIS will be formalized during the development process and extended on demand from added Modules and KPIs, and the increasing need for visualization in the Dashboard.

DDIS Module meta data

Every Module should be defined with the same set of meta data. This will make it possible in the Dashboard to group Modules, connect them to certain KPIs and in other ways create an expressive user interface that is easy to use. The specification of meta data will include properties like Module name, what KPI it uses, description and vendor.

DDIS Module input data

The Module input data specification is a data structure that starts from the Module output and goes all of the way to auto generated mark-up in the GUI and back. By following this specification it is left to the Module to decide what kind of input it needs through the Dashboard. This could be primitive data types, file uploads or spatial data selected on a map. The foundation of these types of data is taken from the HTML standard form input, and is extended with suitable HTML5 inputs and a map input component.

DDIS Module output data

The Module output data specification is a data structure that indicates different kind of data visualizations that is supported by the Dashboard. The definition consists of a key and an appropriate data structure. The foundation is taken from typical chart types, such as line chart, pie chart and bar chart, and is extended with map visualization and other suitable options such as tabular data.

DDIS KPI data

The KPI data specification is a data structure that will enable the Dashboard to create interactive views of the KPIs in the system. When a KPI is selected, the user will be able to store relevant data
connected to this KPI such as as-is values, ambition and other data that is needed by KPIs. This approach is otherwise similar to the ones that is used for Module input and output data except that this data is stored internally in the Dashboard connected to the variants created by the user.

4.5.2.2 IMB connector payload specification

The internal traffic of the Dashboard is handled by JSON-object confirming to the Dashboard Data Interchange Specification (DDIS). The IMB connector will translate some of this traffic to the message payload that is being sent through the IMB Hub.

The IMB connection to the Dashboard is described further in Chapter 5.

4.5.3 Security

The communication between client and server needs to be encrypted by SSL and a certificate needs to be installed. Since the Dashboard is an open source tool a key-pair cryptography needs to be adopted by the Dashboard host using a private key as part of the setup configuration. The session management will use this encryption to ensure secure connections between the Dashboard server and any connected clients.

The user management in the Dashboard will handle levels of authorization to give the user different privileges. The different roles are discussed in D4.1.4 and include the facilitator role and the stakeholder role.

4.5.4 Testing

An integration test suite will be created to cover the different kinds of data being input to the Dashboard server. The data will be contained in physical files and during test execution the file input will be compared to expected output. Automated unit test functionality will be prepared in the client but the actual tests will not be created. The code will be well prepared for unit test by creating modular code as described in Section 4.3.1.

4.5.5 Alternative Dashboard Implementations

The open source approach of the Dashboard and a well-documented installation procedure should make it feasible for external parties to implement their own version of the IDSS Dashboard, thus also taking in consideration how to host and keep user and project data. In this way the IDSS Framework and the IDSS Dashboard separates the concerns of providing functionality versus keeping the decision support data of individual users. The Dashboard configuration should determine which IMB Hub to connect to. The Dashboard hosting should be considered as a possible separate business model connecting users, protecting data and creating custom-made views of decision support for targeted groups.
This Chapter provides the technical design of the IDSS Framework, used in the IDSS for starting and interconnecting Modules, transferring and synchronizing data and communicating with the Dashboard.

The desired functionality has been described in deliverable D4.1, Chapter 3. The technical design of the Framework is strongly related to the technical design of the Dashboard as described in Chapter 4. Some parts will be developed in close cooperation with the development of the Dashboard.

**5.1 INTRODUCTION**

The IDSS Framework has the aim to connect two main parts: the Dashboard and the Modules. As described in Deliverable D4.1, Functional Design of the IDSS, Chapter 3, the connection between these parts is provided by the Inter-Model Broker (IMB) technology, as schematically drawn in Figure 20. The Dashboard (see Chapter 4) is a common interface for the end-users to follow the decision making process and to control of the connected Modules.

As described in D4.1, the Functional Design of ECODISTR-ICT requires for a message service to enable the communication (signaling) between the Modules and the Dashboard within the IDSS, e.g., between the Dashboard server and the KPI calculating Modules. The technology that will be used to achieve this is TNO’s IMB (Inter-Model Broker), because of its scalability, flexibility, speed and TNO’s long-running expertise.

These requirements lead to the more detailed set-up as pictured in Figure 21 where the IMB-Hub as a central part provides the connection with the Dashboard and the Modules.
5.2 TECHNICAL DESIGN OF THE IDSS FRAMEWORK

Each component that is connected to the IMB Hub is called a client. Main pillar of IMB is that clients register events with the Framework. These events can have a signaling structure, but can also carry a data payload. Since the Framework is using a publish/subscribe method, published events with data payload will arrive at the clients that have been subscribe with these events.

With the ‘Publish’ mechanism, clients register the outputs that are to be published to the Framework. With the ‘Subscribe’ mechanism, clients inform the HUB what data should be transferred to the client. Each service can register the data and/or event it outputs to other services (Publish) and also register the data and/or events it needs as an input (Subscribe).

The IMB has three main components that are closely related. In Figure 22 a more detailed schematic overview of the IDSS Framework is presented. The left part of the figure is the set-up of the Dashboard system architecture taken from Figure 15 in Chapter 4. On the right hand side the IMB Framework is presented with its components (in blue) and with different types of Modules attached.
The IMB Framework is built around the IMB communication Hub, the IMB processors and IMB Connector. IMB consists of a publish/subscribe communication Framework and client plug-ins to interconnect computational models, databases, user interface and other clients. These plug-ins make communication possible between Modules that have different data formats. For the IDSS, the starting point in development is existing client plug-ins to connect Modules and data. For the Dashboard a special plug-in will be developed, the IMB Connector.

The IDSS Framework is a ‘push driven’ architecture. The clients are informed by the Hub when new data arrives (call-back structure). This saves the clients from periodically checking the Framework for new or updated data.

5.2.1 Technical demands of the IDSS Framework

The functionalities and demands of the Framework are listed below. The IMB model broker is chosen to meet these functionalities:

a) Interactivity between users on different locations across Europe
b) Interactivity between a user and one or more Modules (responding to a user action)
c) Interactivity between Modules (responding to another Module)
d) Data exchange in standardized formats between Modules, allowing more Modules to be hooked up to the Framework relatively easy
e) Data exchange between Modules and data stores in standardized formats, disabling the need for Modules to write and read to a data store themselves, making the Modules more manageable for developers
f) Flexibility in terms of where Modules, user interfaces and data stores can reside within the Framework (e.g. on a local machine or on different locations)
g) Long term expansibility: Framework capacity can be enlarged
5.2.2 IMB technical aspects

To ensure reliable transmission and correct sequence of the messages and data transported, the TCP protocol is used. TCP is, as part of the TCP/IP (internet) protocol, a connected protocol. By choosing TCP instead of UDP (Unconnected Datagram Protocol), the Framework design can be relatively simple. Clients can be built light weighted having normal hardware requirements and can be easily implemented on different operating platforms.

The target platform is Microsoft Windows (x86 and x64). However, the Java client is also successfully used on Linux and Android platforms before by TNO.

In order to balance load at the IMB Communication hub, the IMB design supports a setup with multiple HUBs. Clients connect to the HUB and become part of the Framework. Clients are part of a domain identified as its ‘Federation’. Clients can only communicate with clients belonging to the same federation. The federation segments the communication between clients. In Figure 23 an example of multiple Hubs and federations is presented. One or more Modules within a certain federation can communicate with other Modules in the same federation on another machine (that is accessible through the network). Furthermore, several federations can exist at the same time. Modules can be the same or different.

![Diagram](image)

**Figure 23** An example of multiple Hubs and federations

Presented is a situation where Modules can run on different machines connected by the IMB Framework, and at the same time belong to the same federation. And specific Modules can run in different federations.
5.2.3 IMB Communication Hub

The IMB Communication Hub is the core of the IMB Framework, handling all messages on a publish/subscribe basis. Modules connected to the Framework inform the Framework of the data they produce (publish) and the data they need (subscribe). The Framework matches the published data with the requested data and therefore effectively interconnects the Modules. For interaction between Modules, a common data format needs to be set for the data or objects to be exchanged.

Using IMB connect several existing Modules, currently each with their own storage of input (parameters) and output data (results). The parts that interconnect will embed an IMB client/IMB processor Module, which will enable communication between the Modules. Integration of the Modules and Dashboard are based on a data driven design as requested by the Dashboard design.

The ‘model-side data’ can be stored in various ways, to be determined from the results of WP2 and WP3, e.g. in:
- a corporate (relational) Data Base Management System (RDBMS);
- local files with model-specific data formats;
- Excel sheets;
- Other, depending on the inventory in WP3.

The Dashboard and the IMB Communication Hub will interact through the IMB Connector with 2 types of data:
- Spatial data: Real-world objects (e.g. buildings or roads) that need to be displayed on a web map and might be changed/edited by the user. In this case, it serves as model input, but it can also be used to visualize model output in the form of KPIs projected on a map.
- Non-spatial data: Tabular data that can serve as model input (e.g. parameters or a lookup table). It can also store model output in order to visualize KPIs in a graph for instance. This data will be stored in the Dashboard database.

The output of the various computational models will be transferred via the IMB Framework. This process will be controlled mainly by the embedded IMB client/processor of the Dashboard server, the IMB Connector. The IMB Connector is a part of the IDSS Framework development and will be incorporated in the software of the Dashboard. The Dashboard server will function as a webserver for central access by the user platforms (mobile, desktop, etc.).

5.2.4 IMB processors

The (existing) Modules need to be connected to the Framework in order to participate in the IDSS Framework. This can be achieved by either:
1. Adapting the model program to include a suitable IMB processor
2. Adding a separate connector (IMB processor) which is able to connect to the Framework and also to do the conversion of the input/output parameters of the model.
Depending on the flexibility of adapting existing models to the IDSS architecture, option 1 or 2 is selected per individual Module. This will enable the models to:

- Send IMB Events for communication to the Dashboard. Events are used to signal the model’s readiness, issue of calculations or perform data requests.
- Exchange Data. This can be any type of data that has either been retrieved from the model data store or has been received over IMB and needs to be stored on the model side. The processor helps translate data back and to an intermediary generic data format (e.g. JSON).

The Figure below gives possible technical options for implementation of IMB processors.

![Possible technical options for implementation of an IMB processor](image)

The graph shows that there are several types of IMB processors:

- IMB data manager for connection to a BIM server
- IMB data manager for connection to a database
- IMB client for a different kind of calculation models
- External data link for connection to WMS-services for ‘Open data’
- External data link for sensor applications

Suitable processors are developed to connect each service to the IMB hub. Currently adapters are available for the following languages:

- C++ (Windows and Linux)
- C#
- Java
- Delphi
- .NET (including micro framework for embedded devices)
• x86 and x64 DLL (Dynamic Linked Library)
• MatLab (using the DLL)
• LabView (using the DLL)

When an existing Module can be easily modified to include support of the IMB hub, this is to be preferred. Input/Output data can be received and transmitted directly by the Framework. Support for various computer languages is available from the IMB program suite.

When existing models are not easily modified to include IMB support a dedicated Module should be developed for the interchange of data. This dedicated Module will handle the transport of the data between model and the Framework. Conversion of the data into usable information for the Dashboard will be part of the Module.

5.2.5 Coupling of calculation Modules

From the preliminary analysis done in WP3, it is expected that the output of a calculation Module will be the input for another calculation Module. This functionality will be provided within the IMB Framework. Only the results of the final calculation model in a row will be sent to the Dashboard.

5.2.6 IMB Connector for the Dashboard

This paragraph is related to Section 4.5.2. It describes the IMB Connector. The IMB Connector will be incorporated in the Dashboard. It will be provided by the developers of the IMB hub.

The Dashboard will be a webserver application and function as the user interface. The Dashboard needs to be connected to the Framework. Through the Framework the Dashboard is able to:

• Visualize resulting data from models
• Visualize KPIs
• Able to send input data to the models
• Able to req. execution of models

The Dashboard will be designed using ‘Node.js’. Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.\(^5\)

Since IMB is an online framework, and has limited ways of dealing with transport to clients or models that are not online. To overcome this limitation, the Dashboard will implement a queuing mechanism in the Dashboard database. When data transport to offline clients has not been acknowledged, the Dashboard is able to resend the data, when the model comes online.

\(^5\) See [http://nodejs.org](http://nodejs.org)
5.2.7 Detailed description publish/subscribe principle

The events can have a signalling structure, and can also carry a data payload from the Dashboard and Modules. Published events with data payload will arrive at the clients that have been subscribed with these events. The design of this publish/subscribe mechanism is presented in figure 25. It describes the API inputs and outputs. The external trigger may be a signal from the Dashboard GUI initiated by a user.

![Figure 25 API inputs and outputs of the IMB](image)

5.3 IMPLEMENTATION

This paragraph deals with implementation of the IDSS Framework in practice. For practical operation a possible hardware configuration is described. For safe operation remarks are made on security of the data and licencing the IDSS Framework.

The possibilities for implementation on hardware set-up, security and licencing will be decided upon during development in close cooperation with the developers of the Dashboard.
5.3.1 Hardware configuration and hosting

In terms of machinery, a possible configuration for the IDSS Framework is shown below. For clarity this will match the logical design given in Section 5.2.

![Diagram of hardware setup](image)

Figure 26 Example of a possible hardware setup of the IDSS Framework

In order to save on hardware expenses and to simplify things in terms of IT management, the IMB Communication hub will run on the same machine as the Web Server during the start of the project.

The hosting can be arranged in several ways:
- At a ECODISTR-ICT-partner during the project
- At a commercial company e.g. Amazon

The costs for commercial hosting are not specified yet but they will be expected not too high because the server space is relatively small.

It is suggested that the solution for hosting during ECODISTR-ICT and after ECODISTR-ICT will be decided during the development of the project.

5.3.2 Security

Over the IMB Framework the messages and Module data, referred to as data payload are transmitted. No encryption, compression or other data handling process is performed on the data payload on the IMB Framework. When secured transport of payload data is necessary, IMB needs to be enhanced using SSL or other secure transport on top of its TCP/IP stack.
5.3.3 Licensing

IMB software is licensed for use in the ECODISTR-ICT project. When used outside ECODISTR-ICT a suitable license model should be agreed upon with the owners of the IPR.

5.4 Planning and scope of work for implementing the IDSS Framework

The implementation of the IDSS Framework will take place in several steps. In this process the developers of the Dashboard and the IDSS Framework will have to work closely together. This section highlights the steps along which the implementation of the IDSS Framework and integration of Modules will take place.

5.4.1 From manual to automatic operation

The development of the IDSS Framework is related to Figure 13 and Figure 14 in Chapter 2 showing the workflow of the IDSS. Steps can be operated manually (green boxes) or automatically (blue boxes). The development of the Framework will take place along the development of the blue boxes. This will result in a growth process of the IDSS from manual operation to fully automatic operation.

Manual operation
The manual operation will prove that the IDSS Framework can be used in decision making processes to acquire the necessary results. The Dashboard is used to guide the decision process and to administer the results. The results of the decision process, e.g., the scores of KPIs that result from using a calculation Module to assess a variant, are stored in the Dashboard database via the Dashboard. The operations on the Modules, the storage and transfer of data, the launch of visualization Modules, etc. are performed directly on the Modules. Or they can also be done without computers by stakeholders discussing the problems and alternative solutions face to face. The end-user copies these results manually to the Dashboard.

Automatic operation
This final stage will result in an IDSS that operates under control of the end user via the Dashboard. The actions in the blue boxes actions can be controlled directly through the IDSS Dashboard.

5.4.2 Template example implementation

In order to give the Dashboard design a jumpstart, a template example will be designed. This template will consist of a webserver example based on Node.js connected to the IMB hub using a modified Java IMB client code. The IMB code will used the call back convention found in the Node.js framework, to made further Dashboard design easier.

The example will be in the form of the Chat example demo as found in the IMB suite.
For integration of Modules at least the next steps need to be executed:

- Specification of the model input (could be (open) data from a data source or an output form another calculation Module)
- Data structure of the data input (model data, context data)
- Make an IMB client to match format data source and Module
- How to run the Module, what signal is required from the Dashboard?
- Data structure of the output
- Make an IMB client to make the output data suitable for storage in the Dashboard database

5.4.3 Development IMB processors

A first we will choose in cooperation with WP3 a relevant calculation Module to follow these steps. From that experience we will learn during development. After that, and with input from WP2 and the preliminary and final results of WP3, we will make an overview of the possibilities of coupling Modules and data. In cooperation with WP3 we will prioritize the development of the Framework related to the desired Modules and the amount of work of development.

An analysis on a per Module basis should be executed to determine which data and controls should be available to the Dashboard. When this information is available, a suitable IMB client will be configured to connect the model to the Framework.

5.4.4 Issues to be determined during development

In this stage of the technical design the issues to realize a Framework cannot be described in detail. Some technical issues of the Framework will be solved during development. These issues will be related to the following aspects and actions:

- Prioritize, with WP3 and WP2, the Modules and data structures for connection to the Framework to decide which Modules will be supported by the IDSS
- Detailed development of the IMB Connector and integration in the Dashboard in close cooperation with the development of the Dashboard
- Security of the data and licencing of the IDSS Framework during ECODISTR-ICT operation and afterwards
- Choices of the hardware configuration and hosting
6 CONCLUSIONS

The objective of the ECODISTR-ICT project is to develop an Integrated Decision Support System (IDSS) for sustainable retrofitting of urban districts with a focus on energy efficiency.

Deliverable D4.234 describes the Technical Design of the IDSS that will be developed and tested in ECODISTR-ICT. The aim of this IDSS is to support professional participants in their decision making processes in sustainable retrofitting projects in urban districts, both for decision making on district and on building level.

This Deliverable elaborated further on D4.1, the Functional Design of the IDSS, published March 1st, 2014. The Functional Design was based on the requirements from potential end-users (What is needed?) and the requirements from the developers of software Modules, data management and integration tools (What is feasible?). The user requirements are generalised into 7 basic steps of decision processes, to be supported by the IDSS. The IDSS consists of many components, such as data management Modules and calculation Modules. The IDSS Framework links the components of the system. The IDSS Dashboard is the interface of the end-users to the IDSS and enables them to use the IDSS in their decision processes.

This Deliverable describes how ECODISTR-ICT includes Key Performance Indicators (KPIs) in the decision process, as an elaboration of the 7 basic steps (in Chapter 2). The mock-up for the Graphical User Interface for the Dashboard shows how the end-users will interact with the IDSS (Chapter 3 and Annex A). The Technical Design of the Dashboard (Chapter 4) and the Framework (Chapter 5) describe the main technical choices made.

The results described in this Deliverable will be the basis for the upcoming tasks in ECODISTR-ICT. Developing the IDSS and its components will be an iterative process, in which relatively small development steps are followed by tests with users. First, internal project partners and later also external stakeholders in the use cases will give feedback. This feedback will be taken into account for the next development steps.
ANNEX A – Mock-up of Graphical User-Interface for Dashboard

The Graphical User-Interface (GUI) for the IDSS Dashboard has been developed as a PowerPoint Presentation. When viewing this PowerPoint presentation in the Slide Show mode, also animations become visible which guide the reader through the process.

The next pages contain the static view on the GUI for the Dashboard, by presenting one slide per page. This version is good for printing the document or viewing the document as a static pdf file on your computer.

For best viewing and understanding the GUI for the Dashboard, the file version should be viewed in the Slide Show mode of PowerPoint. Then the reader is guided through the process by animations. This file will be submitted as Deliverable D4.3 to the EC for review. It will also be made available on the ECODISTR-ICT website http://ecodistr-ict.eu/.
ECODISTR-ICT

Functional Mock-up of Graphical User-Interface for IDSS Dashboard

This mock-up is best viewed in the Slide Show mode of PowerPoint
IDSS Dashboard - Key features

- Web interface - Responsive design
- Targeted users of this mockup: Facilitators
- Functional mock-up – not graphical
- Authorization levels not represented in this mockup

This kind of comment is just a help through this mock up, it is not part of the GUI
Let’s start a new decision process...
Please give a title to this decision process (ex: your high level ambition):

*Reduce energy bills in the district*

Please define the area of your project:

Please name the stakeholders of this project:

*John Smith*

Please add a description text of your project (ex: aim, context...) (optional):

*This project is aimed at...*
Please create and configure your set of KPIs for this decision process.

To start, please drag & drop any KPI or KPI set into this box.

The title just entered will now be displayed.

We can’t go further for now…

This is all we have to do (however we can’t D&D on this mock up…)

Reduce energy bills in the district.
Please create and configure your set of KPIs for this decision process.
Please create and configure your set of KPIs for this decision process.

This “warning” icon appears as the KPI has not been configured yet.

Just click on the KPI to configure it.

You can already go to the next step but this will display a warning popup message.
Please create and configure your set of KPIs for this decision process.

**KPI 1 (e.g. Energy Bills) – AS IS settings**

- **Limits**
  - **Excellent**: [Input Field]
  - **Bad**: [Input Field]

- **Calculation module**
  - [Select Calculation Model/Module]

- **Data**
  - [Browse Data]

- **Priority**
  - Please select how important this KPI is for you (default: 3)

- **Current situation**
  - Average energy bill cost today: [Input Field]

- **Ambition**
  - Set your target value (expectation)
    - [Excellent: 1€]
    - [Bad: 20€]
    - [Acceptable: 8€]

Some areas are fixed, or cannot be determined for now.

The user has to define the score (here is just an example of interface for selecting extreme values).

The list of calculation modules is automatically selected depending on the KPI.

You can finally set the priority of this KPI.

Just save!
Please create and configure your set of KPIs for this decision process.

KPIs database

- KPI 1
- KPI 2
- KPI 3
- KPI 4
- KPI 5

Create a new KPI

KPI sets database

- BREEAM 1
- BREEAM 2
- CUSTOM SET 1

KPI set for this decision process

- KPI 1
- KPI 3

We're good.

The same can be done for our other KPI.

Save this KPI set

Reduce energy bills in the district
Discover all the other functionalities of this screen...

Please create and configure your set of KPIs for this decision process.

Here are all the KPIs you have (with filtering options). You can delete KPIs created manually by users.

This is the set you will use today.

You can create a KPI, with various options/fields (this screen is not represented in this mock up).

These buttons actually display help bubbles like this one!

This can save your set, and add it to the “KPI sets database” on the left. You’ll be able to reuse it or delete if later with the button.

This is a certified KPI set (e.g. BREEAM)

This is a set created by the user. It can be deleted.
Please create and configure your set of KPIs for this decision process.

We can now go safely to the next step. This will run calculation modules for assessing AS-IS KPI.
Collect data

Please give a title to this context:

My Context A: Green Society by 2020

Context data (Variable) database

- Interest rate
- Energy pricing
- Price of PV panels
- Annual increase rate by government

The user first defines different ‘contexts’, which might change the investment.

Selecting the relevant variables and filling in these figures may be needed for KPI calculation, as several modules use these variables (values) for their calculation.

To start, please drag & drop any Variable(s) into this box to define the CONTEXT

i.e. = 4%
Please collect data for the selected KPIs for this decision process.

This "warning" icon appears as data necessary to calculate this KPI has not been collected yet.

You can already go to the next step but this will display a warning popup message.

Just click on the KPI to collect data.

Reduce energy bills in the district.
Please create and configure your set of KPIs for this decision process.

KPI 1 (e.g. Energy Bills) – AS IS settings

**Limits**  
Please define scores

| Excellent | 1 |
| Bad       | 20 |

**Calculation module**  
Please select the calculation model/module you want to use for this KPI

**Data**  
Please upload the data (.xyz file)

**Priority**  
Please select how important this KPI is for you (default: 3)

1 2 3 4 5

**Current situation**  
Average energy bill cost today: ?

**Ambition**  
Set your target value (expectation)

Acceptable (8€)  
Bad (20€)  
Excellent (1€)

Just save!

Discard  
Save settings

You can upload data for this KPI.

NB: A popup message is displayed if any data is missing.
Reduce energy bills in the district

“As is” situation

After a first calculation, this is the AS-IS situation

There are different views

[Any options for table view INTERFACE]
“As is” situation

[Chart type selection (spider / bubbles / pie...) and any other options INTERFACE]

Reduce energy bills in the district
"As is" situation

This would be useful if we need to change calculation model, data, values, KPI priority, KPI list, or anything else we may have set before.

This is optional, and allows to save the data of the process so far, or to export it (PDF, or other).

This will bring us to the TO-BE.

"As is" situation

Layers: Layer 1
Layer 2
Layer 3
Layer 4

Comments/Reviews:
Click on the map to add geolocalised comments

[any other options for map view]

Save/Export these results

Reduce energy bills in the district
Please go through your KPIs again (to set your ambitions...):

KPI set for this decision process

If we want to change our list of KPIs, we have to go back the the previous steps. The same may apply if we want to modify “basic” settings...

The gears tells us we need to take a look into the settings. Let’s see...
We have our KPI settings screen again, with more options available this time. Some options available before can also be inactive at this step (example here: Limits & Calculation Module).

This screen may contain different available or unavailable options, this is just an example.

This has been calculated by the first calculation (as is).

We can now set the to-be (for instance on a slider).

NB: A popup message is displayed if any data is missing.
Please go through your KPIs again (to set your ambitions...):

**KPI set for this decision process**

- KPI 1: Reduce energy bills in the district
- KPI 3: No more wheel

The same can be done for our other KPI

“Next” will run another calculation to determine the TO-BE
"To be" situation

AS-IS can be (re-) calculated.

TO-BE has been set by the stakeholders.

"As is" situation

[Any options for table view INTERFACE]

"To be" situation

[Any options for table view INTERFACE]
“To be” situation

Reduce energy bills in the district

“As is” situation

We can also go back and change some settings before going further.

“To be” situation

This brings us to “Develop alternatives”.

Or export what we already have so far to exploit it.

Save/Export these results

Any options for table view INTERFACE

Analyse problem | Collect data | As is | To be | Develop Alternatives | Assess alternatives | Select Best Alternative
Develop alternatives

Please select one of the following Alternatives and Contexts

Available Alternatives

- My Alternative 1
- My Alternative 2
- My Alternative 3

Variant to assess

This list of available alternatives. Clicking on “Alternative 1” will display the details.
Develop alternatives

Please select one of the following scenarios:

Available scenarios

Scenario 1
Scenario 2
Scenario 3
Scenario 4

Alternative 1 - Description

Description

This alternative uses ...

Measures

Insulate building envelope
Change heating system
Install solar panels

Data
Please upload the data (.xyz file)

Here we have the description of the alternative, in words and in a list of predefined measures.

... and the data (in a file)

Let’s save this alternative

Don’t save this alternative

Save this alternative
In order to make an assessment, the alternatives have to be put into predefined contexts. Each combination of an Alternative and a Context becomes a variant.

Please drag & drop an alternative and a context into this box.

Clicking next will run the calculation and assess this variant.
Assess alternatives

Reduce energy bills in the district

"As is“ situation

"To be“ situation

My variant 1B

[Any options for table view INTERFACE]

[Any options for table view INTERFACE]

[Any options for table view INTERFACE]

The view can be different from this one, and the user can choose to compare various variants (e.g. only “As is” and “My variant1”)

This allows you to modify the alternative & context

This brings you back to the list of alternatives & contexts

This leads to the edition of a report

Develop new alternative

Export report

Your new variant appears next to your AS IS and TO BE

Edit

As is

To be

Develop Alternatives

Assess alternatives

Select Best Alternative

Analyze problem

Collect data

As is

To be

Develop Alternatives

Assess alternatives

Select Best Alternative

Reduce energy bills in the district
With this overview, decisions can be made.

<table>
<thead>
<tr>
<th>KPI</th>
<th>AS-IS</th>
<th>TO-BE</th>
<th>VRT 1A</th>
<th>VRT 1B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unit</td>
<td>Min.</td>
<td>Max.</td>
<td>Translation to Scores</td>
</tr>
<tr>
<td>Green percentage (%)</td>
<td>[%] of Total district</td>
<td>10%</td>
<td>20%</td>
<td>manual</td>
</tr>
<tr>
<td>Energy Bill</td>
<td></td>
<td>1</td>
<td>5</td>
<td>Incomplete</td>
</tr>
<tr>
<td>CO2 Emission</td>
<td>W/m²·K</td>
<td>0.30</td>
<td>0.40</td>
<td>automatic</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<<cells can be hidden later>>

<<can be filtered as only see, e.g., All Variants for Alternative 2 or All Variants in Context B>>

Select best alternative

Reduce energy bills in the district