Appendix 2

for the European procurement procedure

for

Enterprise Architecture Tooling

according to the open procedure

Final version

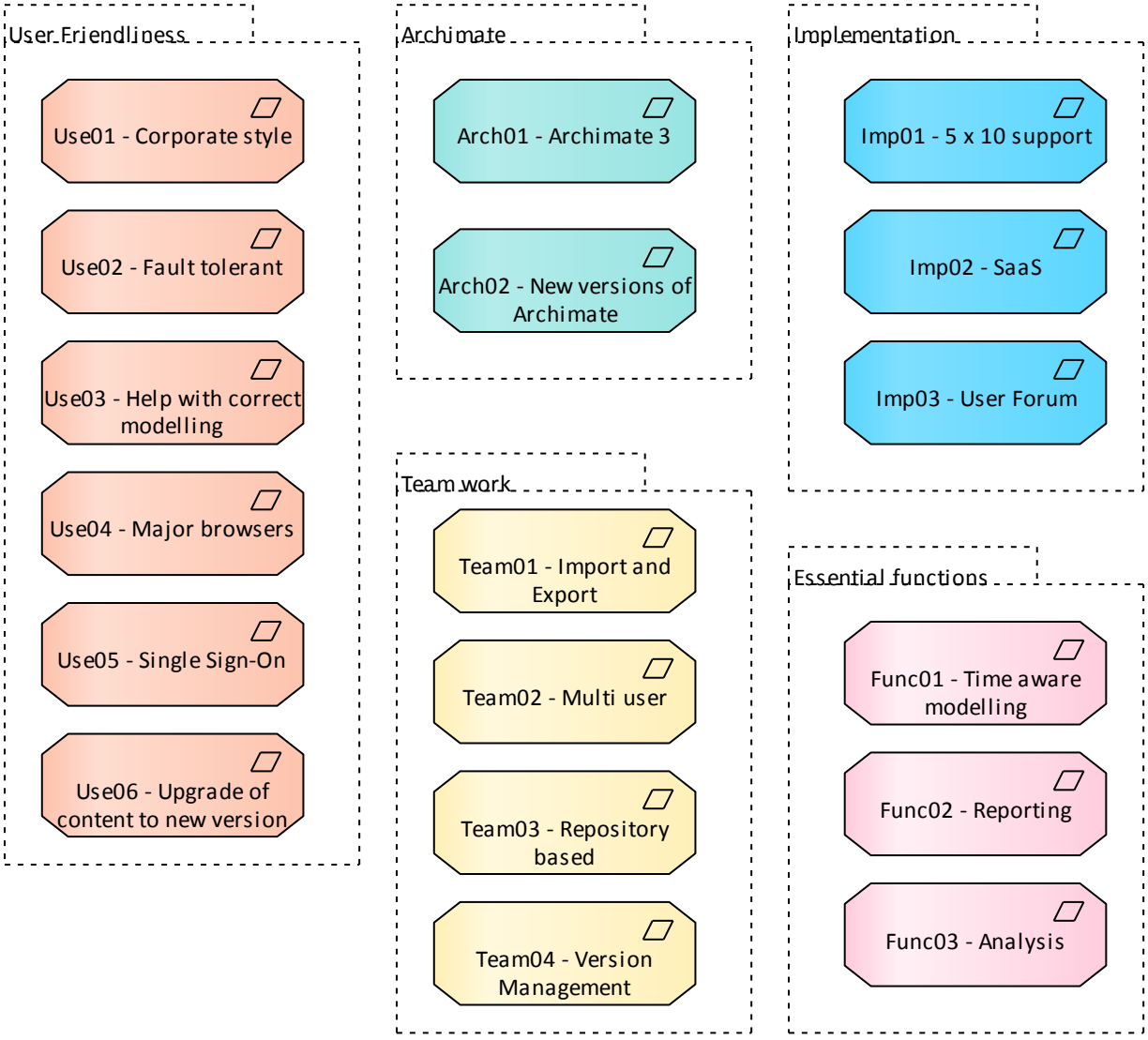
Date: 01-02-2019

Table of Contents

1. Requirements EA Tooling 3
Archimate 4
Essential Functionality 4
Implementation 5
Team work 5
User Friendliness 6
2. Use cases EA Tooling 7
The IoT reference model 8
Use Cases 10
Use Case: Usability 11
Use Case: Time aware modelling 11
Use Case: Version Management 12
Use Case: Analysis 12
Use Case: Reporting 13

1. Requirements EA Tooling

Below the major Schiphol requirements for an Enterprise Architecture modelling tool are stated. The respondents are requested to describe in short statements if and how their product fulfils each of the requirements.



Archimate

The tool must provide complete support for Archimate 3 and next. This is a mandatory requirement. Products that do not implement this requirement are not considered.

Requirement
Arch01 - Archimate 3 The tool must fully support the Archimate 3.0 specification
Arch02 - New versions of Archimate When a new version of Archimate is published by its supplier The Open Group, the new tool should be updated and supported with this new version as soon as possible. Please describe your release and update policy and the activities that are required to fulfill this requirement. Please provide your release notes of the previous Archimate update(s).

Essential Functionality

The tool should support functionality considered essential by Schiphol, this includes Life cycle support, Reporting and Analysis.

Requirement
Func01 - Time aware modelling The tool must support 'time aware modelling'. This means stepping from an AS-IS to a TO-BE architecture, meanwhile showing the difference between steps along different migration paths. Justification: This will allow “what-if” experimentation with different migration paths and it provides Schiphol insight in changes between life cycle phases. Both changes in the objects (applications etc.) and changes in the connectors between the objects must be considered as a step in a migration path. The same set of objects can be connected in different ways in different steps during a migration.
Func02 - Reporting The tool must be able to provide functionalities to create good looking complex reports based on selections or aspects of models and custom built templates. This allows Schiphol architects to create different reports for different audiences diverged from the same model.
Func03 - Analysis The tool should allow various types of analysis possibilities of a model. This, amongst others, allows Schiphol architects to: - Gain insight in direct and indirect dependencies between components - Gain insight in data flows: what are the sources and targets of certain data elements, etc. - Gain insight in complexity and cost, based on the properties of components and their interrelationships Please describe how analysis of direct and indirect dependencies is possible. Please describe other possible types of analysis made possible by your tool.

Implementation

The tool should be easy to be implemented in the Schiphol context, this entails user support, technical support, documentation, training, etc.

Requirement
Imp01 - 5 x 10 support Support should be available on workdays (Monday-Friday) from 08:00 until 18:00 (CEST) via (at least) email and/or phone. Schiphol prefers a resolution time of 2 work days at maximum. Please elaborate upon the kind of support that comes with your service.
Imp02 - SaaS The tool should be made available to Schiphol Group as a service via the internet (SaaS). No local software, apart from a browser, should be required.
Imp03 - User Forum The tool should have a sizable user community with a forum where questions are answered and possible best practises are shared. Please elaborate on the way this user community is structured, the amount of active users and how Schiphol can access the user forum. Schiphol expects the access to the (public or private) to be part of this tender.

Team work

The tool must be fit for a modern team environment, this includes Jira, Slack, version management, workflow, roles and access management.

Requirement
Team01 - Import and Export The tool should be able to import and export content in common formats such as CSV, XMI, etc. Diagrams should be exportable in PNG, SVG, etc.
Team02 - Multi user The tool must be able to be used by multiple users working on the same model simultaneously. This will allow Schiphol architects to co-operate in projects. The tool must thereby prevent conflicts and always keep the model consistent.
Team03 - Repository based The tool should be based on a common repository that contains the models, views, elements and connectors. Diagrams are views on the models in the repository. The repository is updated through changes of diagrams and text by users. Please indicate if there is any type of additional storage needed besides the repository. Please describe the underlying technology of the repository. Please elaborate on this common repository and its functions.
Team04 - Version Management The tool must support version management of artefacts and models at all granularity levels. This allows Schiphol architects to co-operate effectively and integrate each others work into a consistent model. This means: - Baselines can be defined as a reference for changes and as a restore point.

- Changes to the models and views are tracked and can be undone. This will work at all granularity levels where changes can be made. - Changes made by different users can be merged, where conflicts are detected and users will be able to accept and reject changes made by others or themselves

User Friendliness

The tool must be easy to use, also for occasional users, while preventing incorrect and inconsistent models. This includes the possibility to define a “corporate style” as a subset of Archimate.

Requirement
Use01 – Corporate modeling style The tool should have the capability to define a corporate style for modeling. This means that it should be possible to define a subset of (Archimate) element- and connector types to be used. The tool will support users to stick to this subset and will signal deviations. This allows for Schiphol architects: - to create models that are easy to understand and maintain - to enroll new architects into the Schiphol way of working easily - to hand over their work to other architects without the risk of misunderstanding
Use02 - Fault tolerant The tool should recover from technical failures or user errors in a short time without losing data.
Use03 - Help with correct modelling The tool should be able to help the modelling process and prevent incorrect models by e.g.: - Suggest correct element-types and relationship-types where possible - Prevent creating redundant copies of elements - Disallow incorrect or missing relationships This allows Schiphol architects to create models that are correct and consistent.
Use04 - Major browsers The tool should support at least the following internet browsers: Internet Explorer 11 and up, Firefox 60 and up, Chrome 70 and up, Safari v12 and up. Please indicate the browsers that are currently supported by your tool and share your release policy when it comes to (browser) updates.
Use05 - Single Sign-On The tool should support single sign on (SAML + SCIM 2) so that users who are logged on to the network do not have to authenticate again.
Use06 - Migration of content to new version The tool should make it easy to migrate existing content to a new version of the tool, including an upgrade to a new version of Archimate. Please indicate on the way(s) how you support the migration of existing content. Please provide insight in previous (Archimate, repository) upgrades and the migration of existing content to the newer version.

2. Use cases EA Tooling

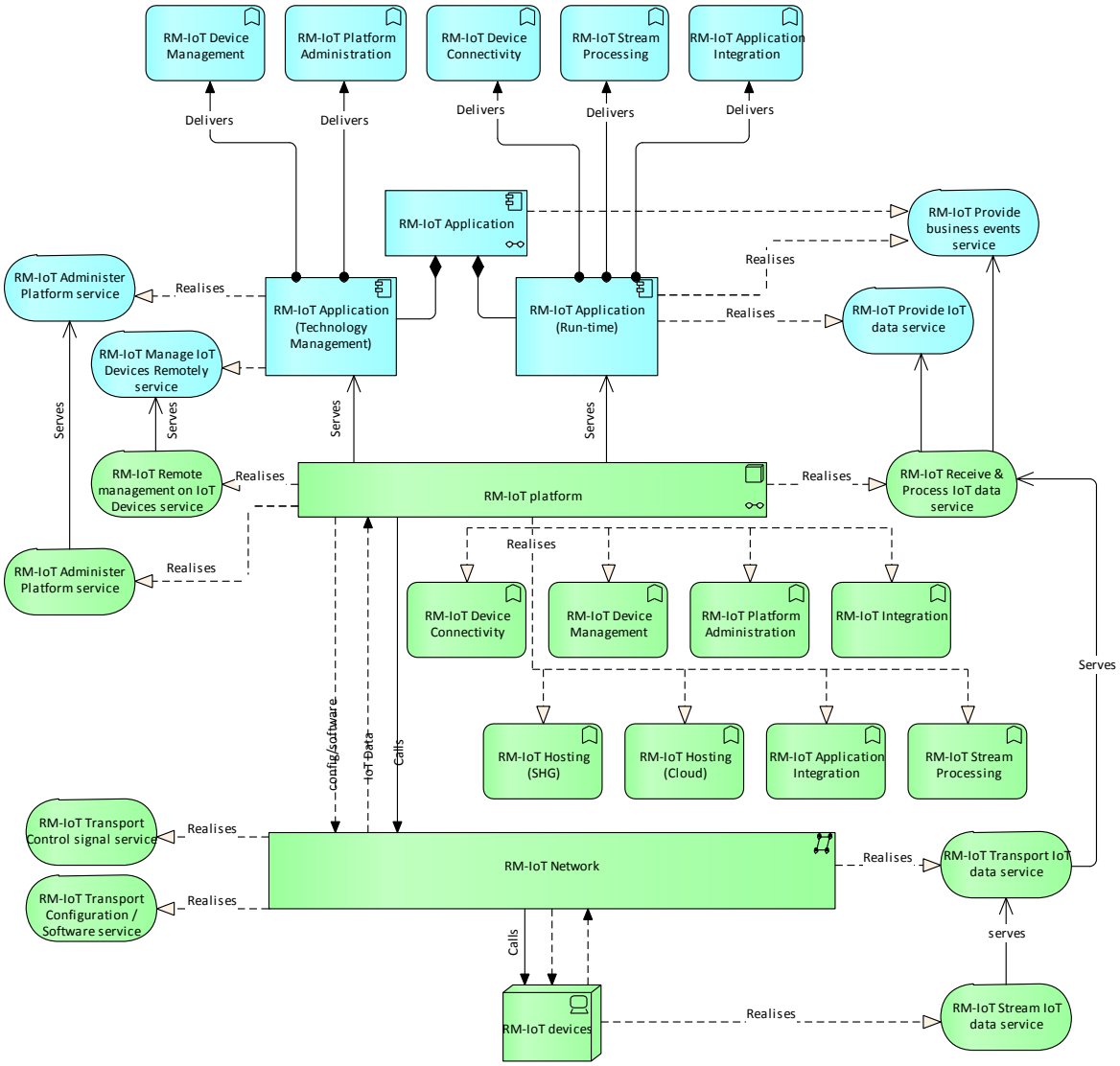
This section of the document is concerned with demonstrating the capabilities of the EA tooling.
In order to be able to compare the demonstrations of the various vendors, Schiphol has selected a specific domain for the demonstrations, namely the Internet of Things. Schiphol has also selected a number of typical practices of the EA tooling in the form of use cases.

Below you will find:

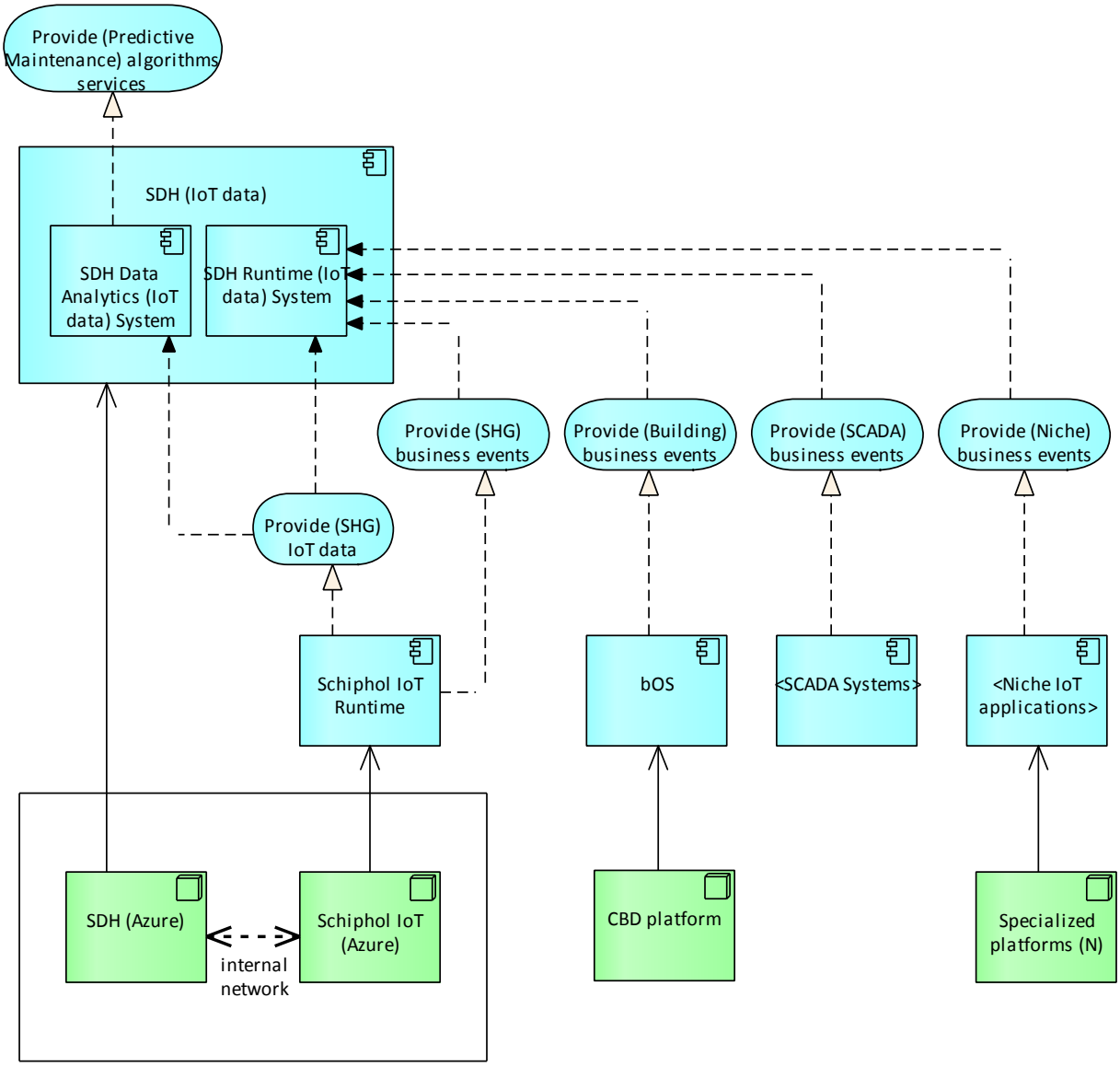
- The Schiphol IoT reference model (still under development)
- An hypothetical example of an IoT system based on the reference model
- A diagram showing the selected use cases
- A short description of each use case and room for annotation and scoring.

Please note that the goal of the demonstrations is to show the capabilities of the tool. Discussion of the content used for the demonstrations is out of scope.

The IoT reference model



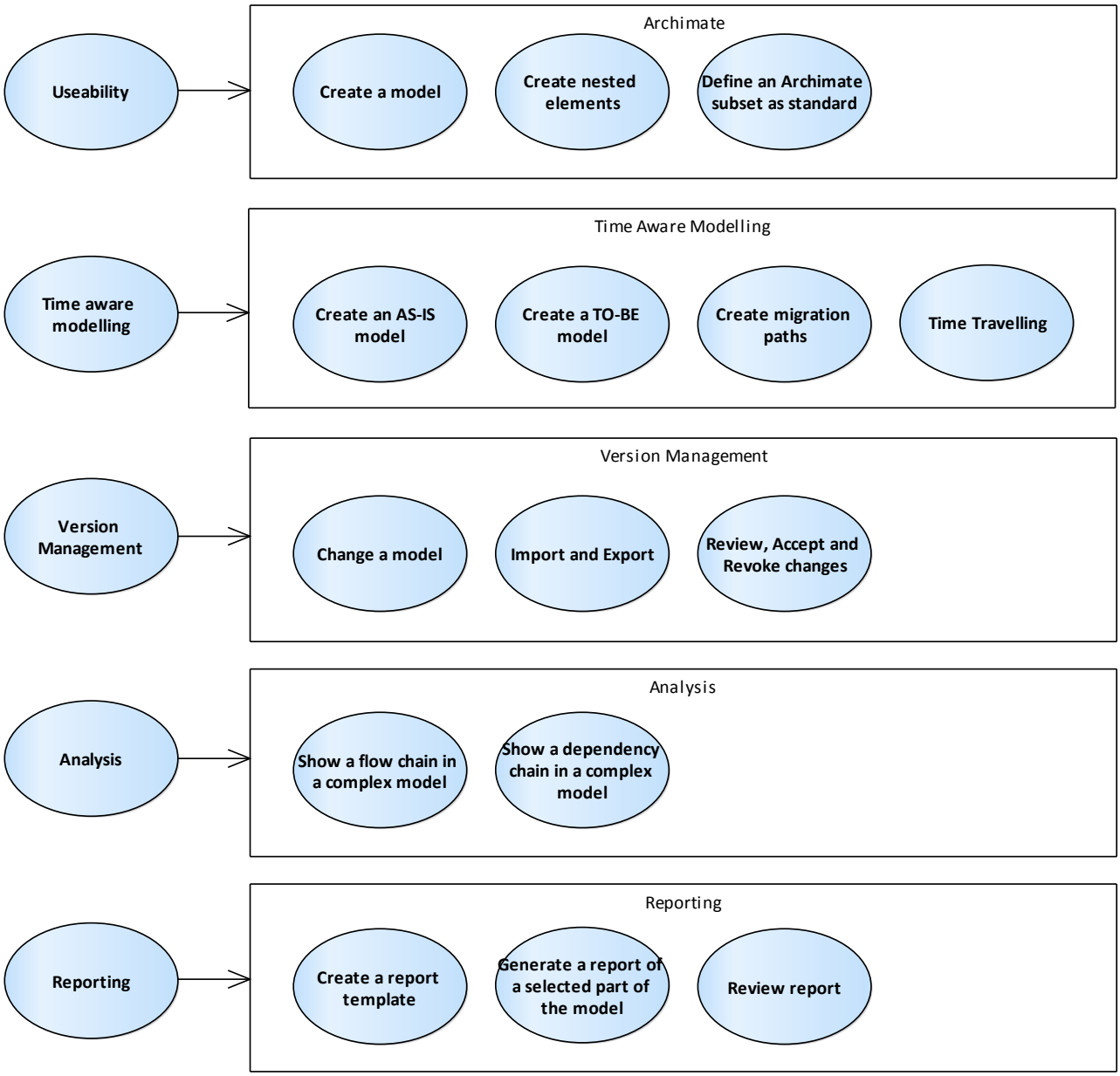
This is the SNBV Internet of Things reference model as an Archimate diagram. The core of the model is the IoT-Platform. This is realized on Azure as the Schiphol Data Hub (SDH). The IoT platform realizes a number of functions. It receives events from devices via the IoT Network. It contains applications for analysis as well as for run-time operations such as data exchange and management.



This diagram is an example of an IoT system based on the IoT reference model. It shows components generating events which are collected and used for data analytics purposes. The Schiphol Data Hub is hosted on an Azure cloud platform.

Use Cases

The following use cases are some of the generic uses of an EA modelling platform, to be used in the demonstrations of the tooling.



Use case	Comment	score
Use Case: Usability The Usability use case is intended to demonstrate aspects of usability of the tool. This includes amongst others: • the ease with which functions and features are found and used; • how the tool aids the user in producing good looking and syntactically correct diagrams; • how the tool can be tuned to the requirements of the organization.		
Create a model	Create an Archimate model with various element types and connector types.	
Create nested elements	Nested elements are visually attractive but nesting has no meaning in Archimate. The tool should help to create Archimate relationships between nested elements.	
Define an Archimate subset as standard	Creating an Archimate subset helps to simplify the models and make them better as communication tools. This might entail: Excluding certain element types, excluding certain connector types and showing how the subset is used.	

Use Case: Time aware modelling The use case on Time Aware Modelling is intended to demonstrate how the tool can be used to obtain insight in the life cycle of a system. An example of this is an investigation of the consequences of different migration paths from the current situation to a target situation.		
Create an AS-IS model	Create a model representing the running situation.	
Create a TO-BE model	Create a model representing a projected future situation.	
Create migration paths	Create a couple of intermediate models representing points in the lifecycle between AS-IS and TO-BE.	
Time Travelling	Show how migration path alternatives from AS-IS to TO-BE can be visualized and discussed.	

Use Case: Version Management The use case Version Management demonstrates how the tool enables different people to work on the same model while keeping the model consistent, making sure that changes are merged intelligently and conflicts are handled appropriately.		
Change a model	Take an existing model (diagram) and change elements and connectors.	
Import and Export	Show how parts of the model can be exported and imported and how this interacts with the version management.	
Review, Accept and Revoke changes	See changes made since a selected baseline, accept some changes, reject some changes. Show how the authors get informed, etc.	

Use Case: Analysis The Analysis use case shows how the EAT is useful to analyze interesting properties of the model and parts thereof. Things that are worth analyzing are dependencies, information flows, complexity, etc.		
Show a dependency chain in a complex model	Show how the tool can be used to give insight in dependency relationships, e.g. for impact analysis.	
Show a flow chain in a complex model	Demonstrate how the tool can be used to show how data flows through the system that is modeled, e.g. where does the data originate which a particular object uses.	

Use Case: Reporting The use case Reporting shows how human readable documents and possibly other types of files can be generated from the content of the tool repository. Reports are based on a layout template and a specific selection of content.		
Create a report template	Show how a template for a report can be created and changed. The report is intended to document a particular view and/or subset of the model.	
Generate a report of a selected part of the model	Show how a report is generated, using a particular template and/or with an ad hoc filtering.	
Review report	Show how a report can be generated that is easily understood by a non-technical audience.	