ARTESIS PLANTIJN
HOGESCHOOL ANTWERPEN

RFP

Lean Enterprise Architecture Platform (LEAP)

ARTESIS PLANTIJN
HOGESCHOOL ANTWERPEN

Table of Contents

1. Context

This document is a formal Request for Proposal for the selection of a partner to deliver a lean enterprise architecture platform. The goal of this RFP is to be able to evaluate the competences, experience and problem-solving skills of the candidates and the feasibility of the proposed solutions. This RFP will provide the client with the necessary input to make a decision concerning the preferred partner for the implementation of the required web application.

2 Procedure

1. Points of contact

During the RFP-procedure, following persons can be contacted in case of questions:

Concerning the procedure: karolien.vanriel@ap.be
Concerning the content: kelly.casalmosteiro@ap.be

2. Timing

The candidates will submit the proposal through Digitap on 1/11/2020 at the latest. In case additional questions about the content come up, these can be addressed during the interviews on 5/10/20. The questions for the interview need to be submitted before the interview through Digitap.

All answers will be made public to all parties after the interviews to ensure that all parties have the same access to the published information. The client has the right to provide additional information during the RFP procedure and can communicate additional requirements at any time in the process when needed. All additional communication will be done through Digitap.

All proposals will be evaluated on correctness, technical and functional applicability and feasibility of the proposed solution. Following time provides an overview of the different milestones in this RFP-procedure.

- Publication RFP: 30/09/2020
- Deadline questions interview: before the interview
- Interview: 5/10/2020
- Deadline Scope & WBS 25/10/2020
- Deadline final proposal 1/11/2020 23h59 CET

If the documents are submitted after the deadline, the following rules apply:

- Submitted within 48h: -2 on the document
- Submitted after 48h: no evaluation

3. Additional information

- When specific software or hardware licenses are needed, these need to be mentioned specifically in the proposal. The candidate also mentions all products, technologies, versions and possible alternatives.

- The candidate is allowed to suggest alternative/additional solutions and functionalities if these are of added value to the client.

- The candidates are bound to the given information until the end of the academic year 20-21.

- The proposal needs to be submitted in pdf-format in Digitap. Additional information may be added as a zip-file if needed.

- The proposal minimally consists of the following parts:

- Scope statement (context inclusive)
- Project goals
- Proposed solution (overall, details not required)
- Cost estimation based on WBS
- Preliminary timeline
- Assumptions, constraints and known exclusions
- Risk analysis

The proposal is submitted on an AP-template and mentions all team members who participated.

3. Scope and context

3.1 Context

The purpose of this document is to describe the requirements and specification of the LEAP web application. LEAP stands for Lean Enterprise Architecture Platform and is to be used to create models and maintain information in the field of Enterprise architecture.

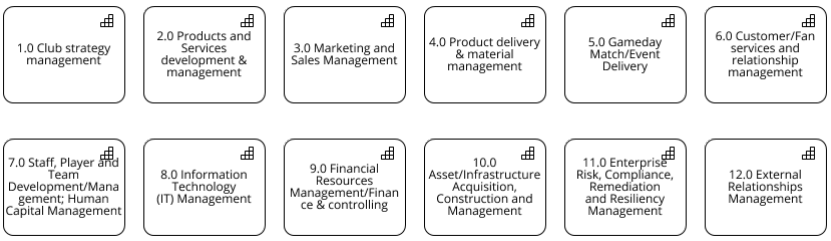
The main idea is to provide a web environment where analysts/architects/managers can enter and maintain information on the structure or architecture of their organization. **For more information on the context of the application, please refer to the video of the kickoff session. Please check the video, because it will provide you with the right info.** Briefly stated, is it important to have a clear overview of the 'structure' of your organization as, for example, a manager. However, currently companies need to draw these structures themselves in Powerpoint, Visio and so on. The problem with that is that the model changes and this means a lot of work to update the Powerpoint drawing etc. everytime. Therefore, we need an application where you can enter information and the application draws the model. In an ideal situation, you can export these models to a powerpoint format or PDF.

3.2 Scope

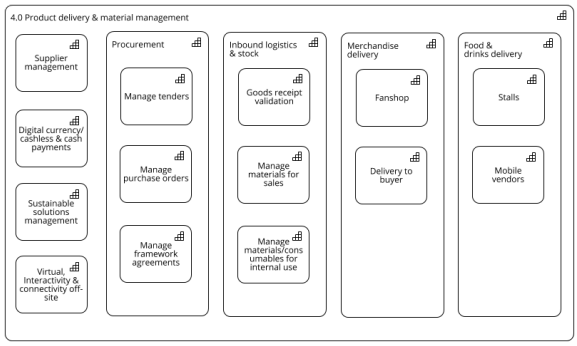
The specific components that are needed in the application are explained below. The idea is that an organization can create a sort of repository where you can store information on these components and link them to each other to understand and show relations between the components. The second part of the application is then generating models, based on the information on the different components that we have defined/stored.

The relevant 'components' for a user are the following. Please refer to the use cases for more information on the requirements.

- **Environments** (a user can create 1 or more environments. This is mainly the 'company' for which the user will create the other objects).
- **Capabilities**. A capability is a 'building block of an organization'. All capabilities in an organization show all the things that organization needs to be able to do, but on a high level (read: low level of detail). We call this the 'what' view because it shows us what an organization needs to do. We use this to provide a clear overview of all things the company needs to think of. They relate sometimes to concepts such as 'departments', but they help us to look at the organization from a functional point of view. Below you can find an example of the level 1 capabilities of a professional sports club, such as a football club.



Each capability is linked to 1 and only 1 environment. A capability can exist on level 1, 2 or 3 (always on 1 level). This means that a capability can have 0 to 1 parent and 0 to n children. (In other words: a capability can only belong to 1 parent capability). Example of that same football club:



In this example, the capability *4.0 Product delivery and material management* is on level 1. *Supplier management, procurement, inbound logistics & stock* and others are on level 2. They all have *4.0 Product delivery and material management* as a parent capability. Procurement, on level 2, has also 3 child capabilities: *manage tenders, manage purchase orders and manage framework agreements*. You can see that these 3 capabilities have only one parent, being *procurement*. You can ignore the silver color for now.
[General remark: the icon (tetris blocks) on the top right are not needed in LEAP/the application you will build. They are there, because these examples were drawn with a program called Signavio, which uses the Archimate language specification).

- **Strategy**. We want to allow to define strategies in LEAP. Each strategy is linked to 1 and only 1 environment.
- **Strategy items**. A strategy can be further split up in strategy items and these items then can be linked to capabilities. More on this below in the use cases. Each strategy is linked to 1 strategy and to 0 or more capabilities.
- **ITapplication**. We want to be able to list all applications that a company uses and we want to know what capabilities use an application. We do this, so we can estimate the importance of an application and analyse how well the application support business in executing its processes. Furthermore, there will be certain properties that we can store on the application level and on the relationship between an application and a capability.

- **Resource**. There will be 2 types of resources (human and other). Each resource can be linked to 0 to many capabilities and a capability can be linked to 0 to many resources. Again, some properties will be stored on the resource level and on the relation with the capability.
- **Business processes**. Each business process can be linked to 0 to many capabilities and a capability can be linked to 0 to many business processes.
- **Program and project**. Programs can be defined, as well as projects. The idea here is that a program can exist of multiple projects (so project is a lower level object). A project can then be linked with capabilities (in other words, we want to define the following: what capabilities is a project impacting?). The program will be a property of the project: in other words, you can assign a project to a program (if you want to).

Another important part of the application is the report and export part. The main purpose is to create a program which allows to easily generate reports and, if possible, exports like slideshows (e.g. powerpoint). This is important, because this should be a business-oriented application and this type of users often use dashboards, powerpoints and spreadsheets to communicate.

For examples, we refer again to the video of kickoff session, where you can see for example how a specific parameter is displayed on the capability map (in that example, we showed the strategic importance of a capability in relation to a specific *strategy item*).

Functional scope & requirements

Use case	Description
1a	Creating a user and resetting password
1b	Logging on
1c	Creating and selecting environments
1d	User maintenance
2a	Creating and maintaining capabilities + properties/values
2b	Completing the DNA attributes (linked to other components) of the capabilities
2c	Swot analysis for a capability
3a	Creating and maintaining applications (IT component)
3b	Creating and maintaining business processes (process component)
3c	Creating and maintaining resources (resource component)
4a	Strategy definition and decomposition (link with capabilities)
4b	Program and project definition (link with capabilities)
5a	Capability map report
5b	Capability map exports
5c	Capability map layer report
5d	Capability map layer exports
6a	Import via csv (excel)
6b	Industry templates

The different use cases will have to be explained/worked out.

Qualitative requirements (non-functionals)

1. Web-app