Requirements Quality Study

This document contains all instructions for participation in the requirements quality experiment. **Please read the information carefully and in order.** This part of the study is expected to take 50 minutes, and the concluding survey (the link to which you will find at the bottom of this document) will take about 10 minutes.

Instructions

The Experiment

This experiment investigates the effect of requirements quality on the activity of domain modeling. To this end, **you will be presented with five requirements and tasked to create a domain model for each requirement**. The requirements are derived from real requirements specifications. You will find a description of its context before the requirements.

The Task: Domain Modeling

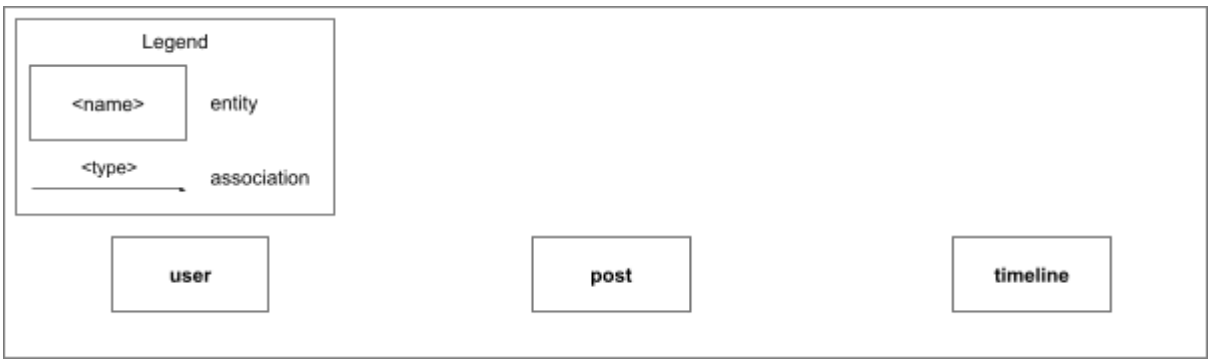
The task of domain modeling consists of two steps:

1. Identifying all **entities** in the requirement.
2. Identifying all **associations** in the requirement.

The resulting domain model consists of nodes (entities) and edges (associations) and represents the information of the natural language requirement. We demonstrate these steps with a simple example, using the requirement, “A user shall be able to create a post on a timeline.” of a hypothetical social media platform.

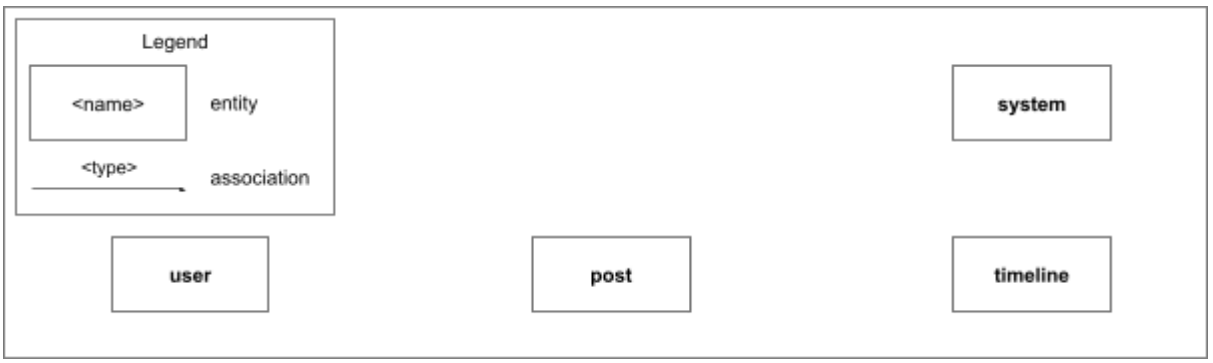
1. Identifying entities

Entities are all objects and actors (humans) mentioned in the requirement. Entities are usually characterized by nouns or noun phrases. To complete this task step, add one node for each entity you identify in the requirement.
The exemplary requirement contains three explicit entities (one human and two objects): “**A user** shall be able to create **a post** on **a timeline**.”



Note that not every requirement contains a human actor, but all requirements have to contain at least one object representing the part of the system responsible for the

requirement. Not all of these entities might be mentioned explicitly in the requirement. In this example, no entity represents the system responsible for the requirement. Hence, we add a fourth entity:



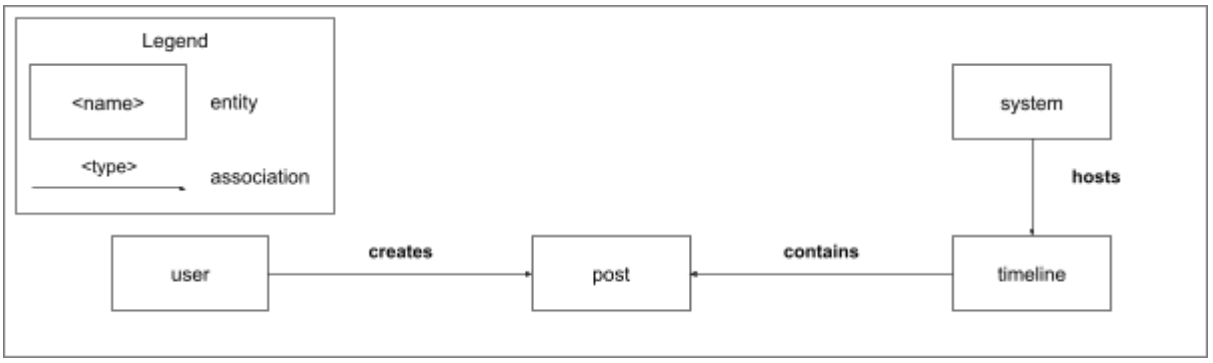
Because, in this example, the context (“a hypothetical social media platform”) does not provide us with more information about the parts of the system that could be responsible for this requirement, we add a generic “system”-entity to the model.

2. Identifying associations

Associations are relationships between entities. Associations are usually described by verbs that characterize the interaction between two actors or entities. To complete this task step,

1. connect the two related nodes (actors/entities) with a line,
2. annotate the line with the type of relationship (often the used verb), and
3. add an arrow to the relationship type to indicate the correct direction.

The exemplary requirement contains one explicit and two implicit associations: “A user shall be able **to create** a post **on** a timeline.”



You will notice that while verbs with a single argument (e.g., “X creates Y”) are easy to translate into an association, verbs with multiple arguments (e.g., “X creates Y in Z”) cannot be represented by one single association (since associations connect only 2 entities). The example above shows how a verb with multiple arguments translates into two associations. You can reword the requirement to make this relationship explicit (“A user shall be able to create a post **which is contained in** a timeline”). Determine associations with implicit entities (here: the system) based on the context.

Each association should become “readable” based on the following template: a <source> <type> a <target>. The two associations in the target are readable as follows: “a user creates a post,” “a timeline contains a post,” and “the system hosts the timeline.” Notice how the direction of the arrows influences how the associations are read.

Experimental Task

Here, the experimental task begins. The section “Context” explains the domain and system for which the requirements were written. The section “Requirements” contains the requirements you are supposed to process.

Context

The **ASPERA-3 Processing and Archiving Facility (APAF)** is a data system responsible for processing telemetry (i.e., sensor data) of the ASPERA-3 instrument package, which was flown to Mars on the Mars Express mission in 2003. ASPERA-3 contains several different sensors that measure the particles, neutral atoms, and fields in the near Martian environment. The APAF data system provides data processing algorithms to support the ASPERA-3 science team in data analysis. It consists of a data processing unit procuring and handling the data, interfaces to various databases, a web-based front-end, and a dedicated logger for maintenance purposes.

Begin

You can now begin the experimental task. Please note the time of beginning.

Starting time

15:10

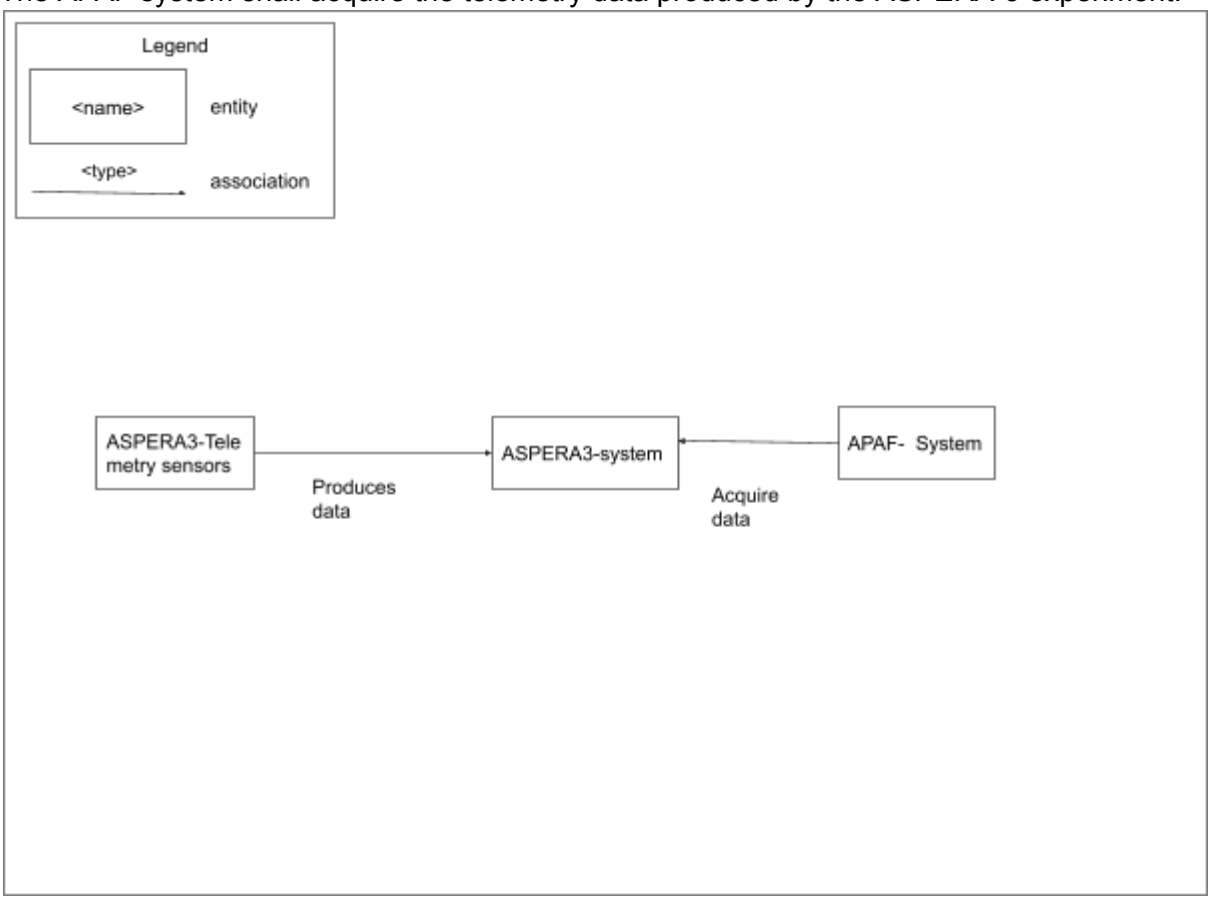
Requirements

Consider the following requirements and create a domain model according to the above instructions for each of them.

Use the template provided to you under each requirement. **Click on the bordered box and select “Edit” to create your model.**

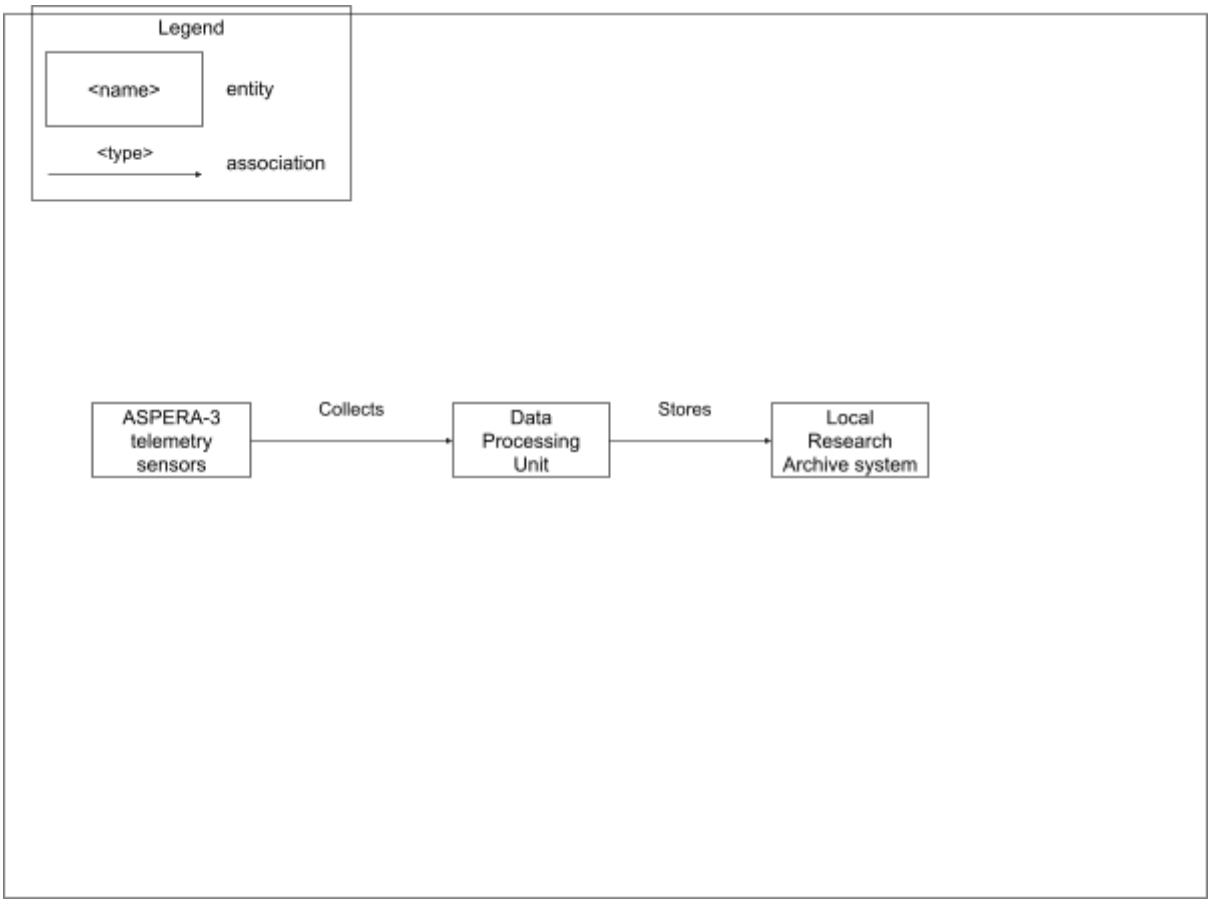
Requirement 1

The APAF system shall acquire the telemetry data produced by the ASPERA-3 experiment.



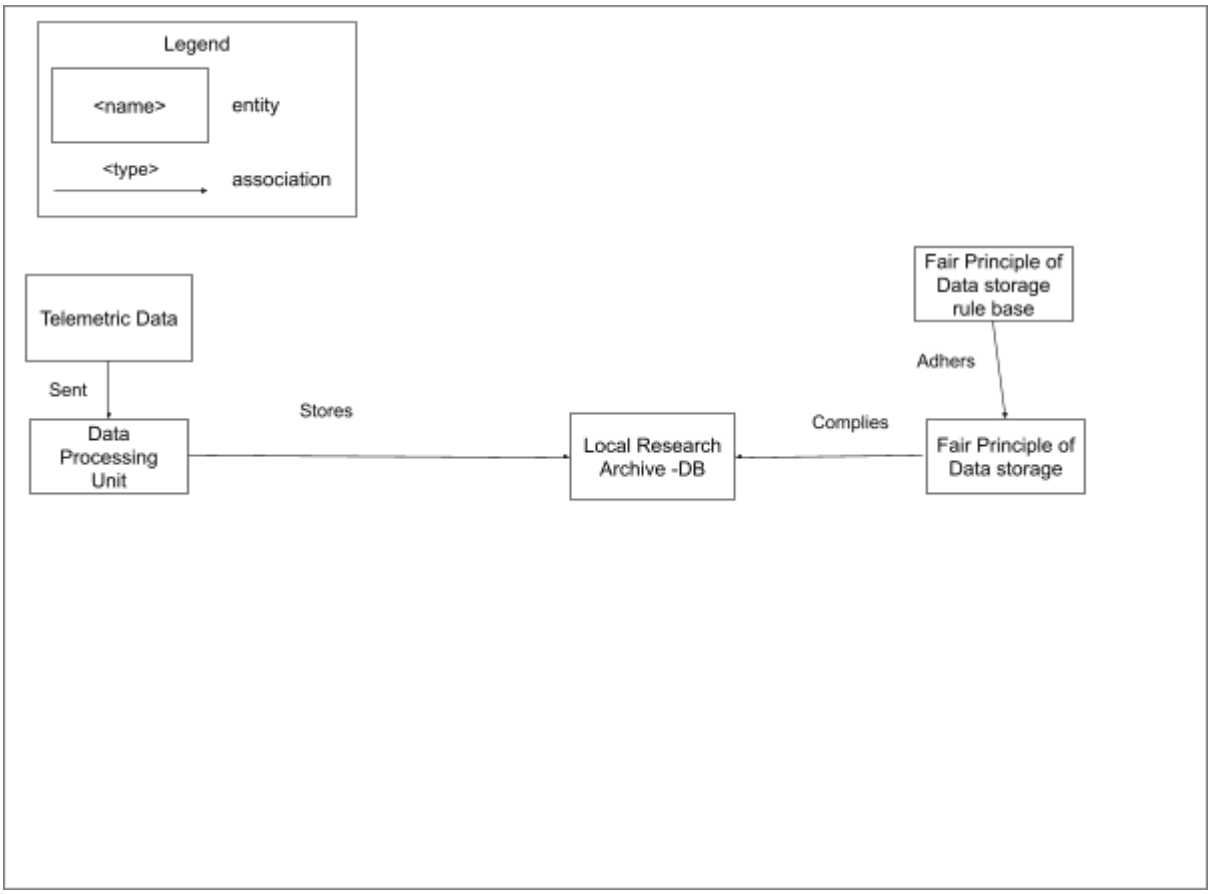
Requirement 2

The data processing unit shall store the ASPERA-3 and Mars orbit data sets on a local research institute archive.



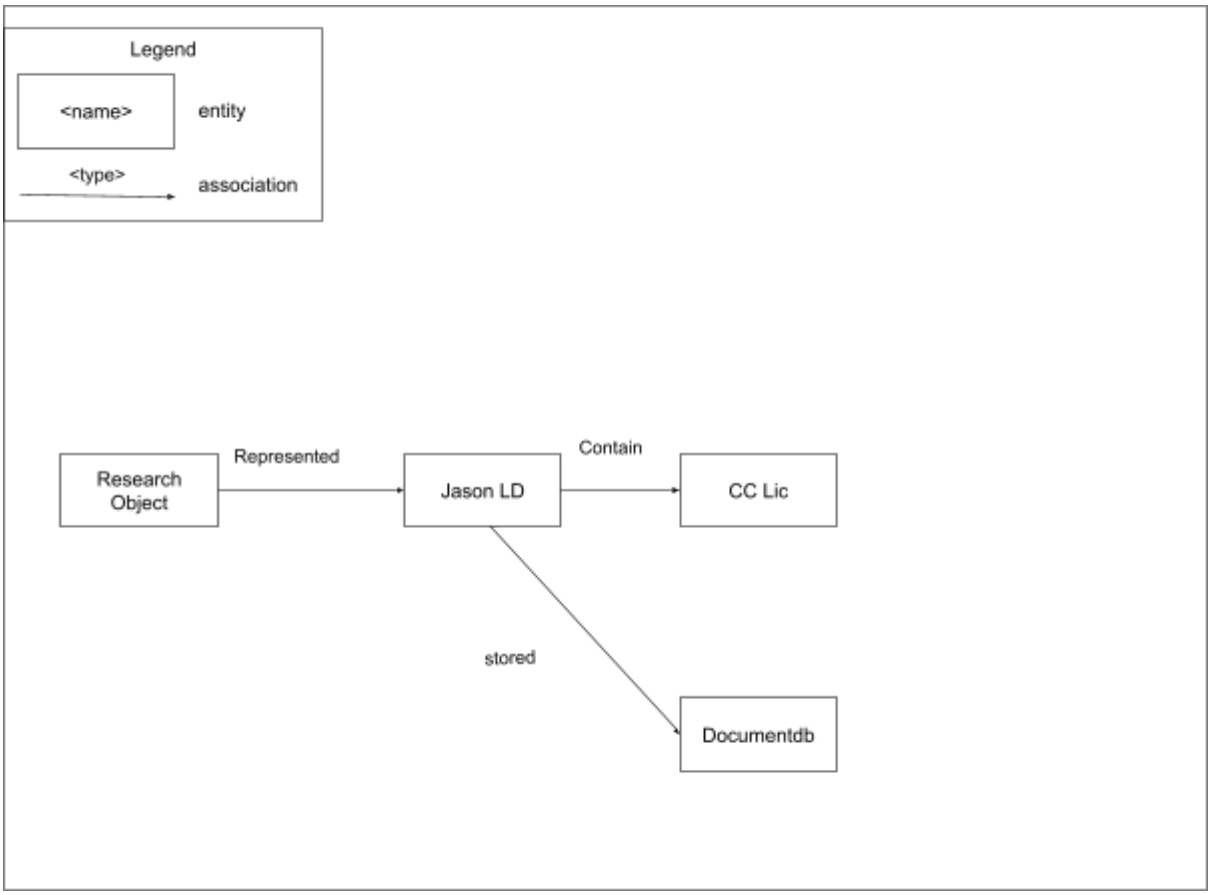
Requirement 3

The data processing unit stores telemetric data for scientific evaluation; therefore, it needs to comply with the FAIR principles of data storage.



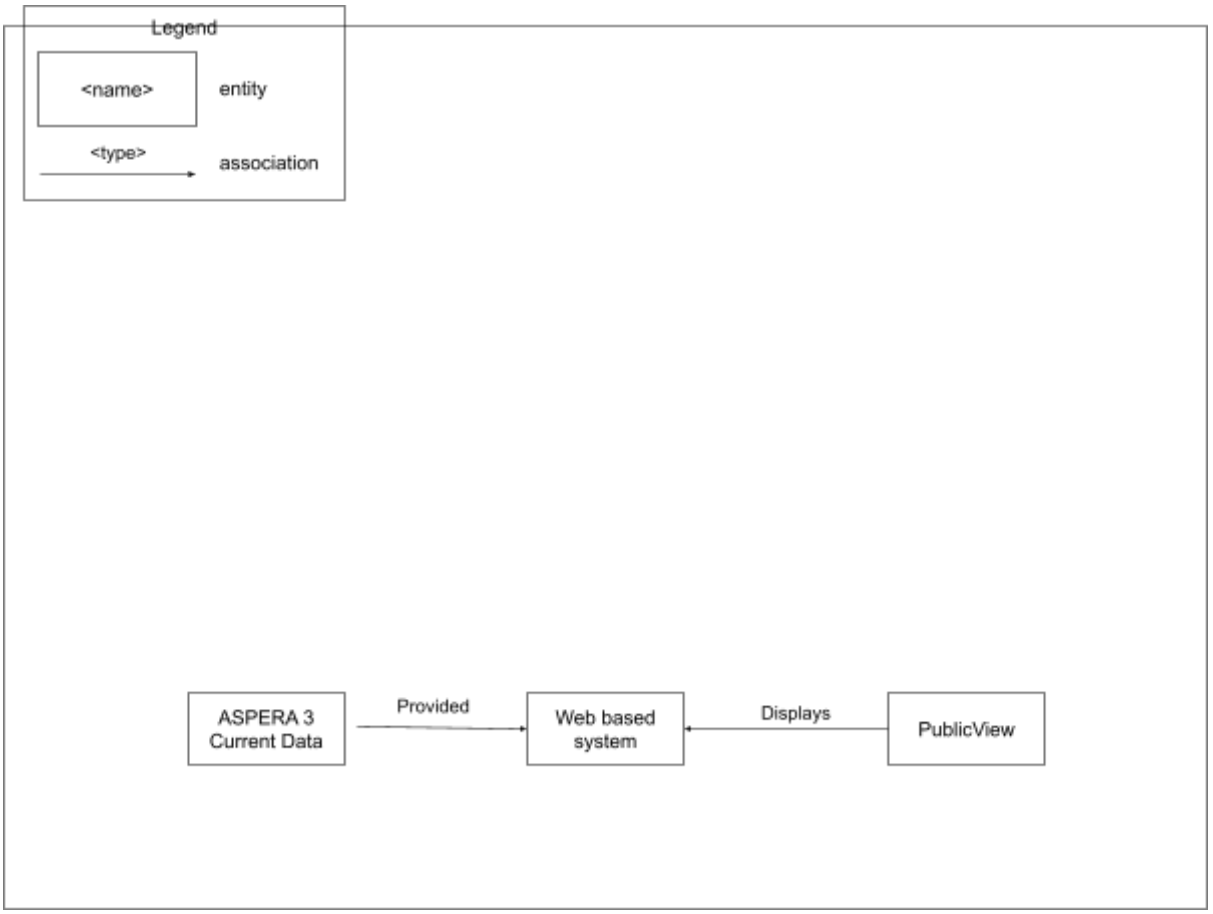
Requirement 4

Every research object is represented in a JSON-LD format and stored in a document database if it contains a CC license.



Requirement 5

Web-based displays of the most current ASPERA-3 data shall be provided for public view.



Ending Time

Once you have completed the experimental task, please **note the completion time**.

Ending time

16:08

Concluding survey

Before you conclude, please **fill in the following survey** to complete the study participation: https://docs.google.com/forms/d/e/1FAIpQLSdqvIJKv0Oq7Sq4IFF3SPsOgLFVp-hemkXDaDvjyvIT7NDZqw/viewform?usp=sf_link