# Domain Analysis

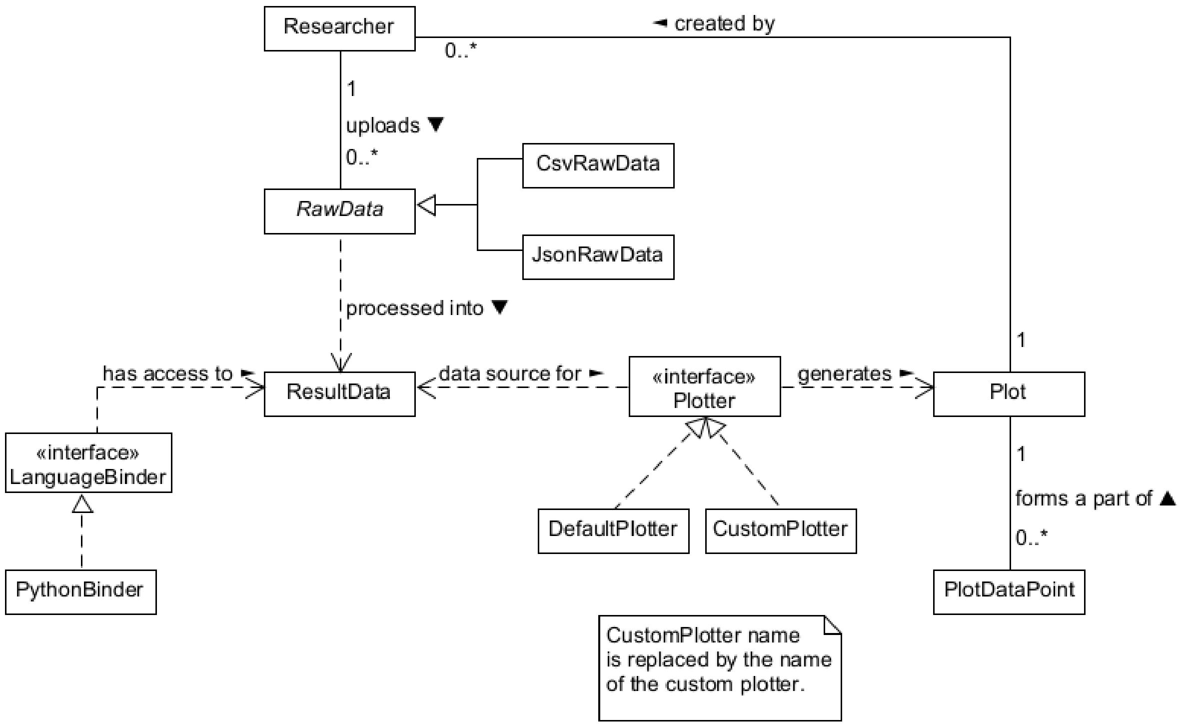
## Project 7

# Contents

# Description of Domain Structure

## Conceptual Class Diagram

**Figure 1**
***Conceptual Class Diagram***




## Explanation of Concepts

**Table 1**
***Explanation of Concepts***

| Class name | Responsibilities |
|---|---|
| Researcher | Using a form of login system, Researcher contains references to the resulting data created from a user's uploaded raw data, and references to plots generated by the user from that data |
| *RawData* | Superclass for all data being uploaded. Allows for additional formats to be added into the system as required |
| CsvRawData | Class to interpret data uploaded in Comma Separated Values (CSV) format |

| | |
|---|---|
| JsonRawData | Class to interpret data uploaded in JavaScript Object Notation (JSON) format |
| ResultData | The resulting information after Bayesian Data Analysis has been performed on the raw, uploaded datafile |
| <<interface>> LanguageBinder | An interface that stores the different possible algorithms that can be used when processing ResultData. |
| PythonBinder | The stored python algorithms used for processing ResultData |
| <<interface>> Plotter | An interface that specifies the functions and functionality DefaultPlotter and CustomPlotter must implement |
| DefaultPlotter | The built-in plotting routine used when creating plots. |
| CustomPlotter | A class for custom plotting routine as defined by the user. |
| Plot | An instance of a dynamically interactable plot. Able to have domain and range, axes, and combinations of axes modifiable during interaction |
| PlotDataPoint | A class for manually adding data points to the plot. |

# Glossary

- CSV - A Comma Separated Values (CSV) file is a plain text file that contains a list of data
- JSON - JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax

# Review of Alternative Software

## PESummary

PESummary is a Python library for generating summary pages of the results of Bayesian Parameter Estimation.

### Advantages

- All plots are pre-generated, meaning that navigating to a specific plot is fast.
- Produces a web page, from which all results can be navigated to. The web page URL can be sent to others, allowing for easy sharing of the results.
- Multiple import formats are supported by default, and custom importing behaviour can be defined to handle other formats.
- Free and open source, with plenty of documentation.

### Disadvantages

- Almost all plots are static images, offering zero dynamic interaction. There is only limited interactivity available with specific corner plots, allowing for selecting points on the plot, and zooming in to specific regions.
- All available plots are pre-determined; there is no functionality for generating new custom plots on the web page
- Plots tend to be poorly labelled, making it difficult to understand their contents
- The main user interface for generating the web page and its plots is using the command line.
- The web page's usability, user interface (UI), and user experience (UX) leave a lot to be desired, to the point where it significantly inhibits the user's ability to navigate and comprehend the results.

## PyCBC

PyCBC is a Python-based tool used to study astrophysical sources of gravitational waves. It has the ability to load in data, visualise it and perform signal processing to identify signals along with consistency and testing. The resulting graphs can then be displayed on a generated webpage.

### Advantages

- Easy to install using PIP
- In depth tutorial describing how to start analysing gravitational wave data
- Proven to provide meaningful insight as it was used in the first direct detection of gravitational waves

### Disadvantages

- Unrefined UI as some elements are too large while others are too small leading to lots of unused space. Additionally, the formatting for some sections breaks on a mobile screen size.
- There is no sitewide search available, making it difficult to quickly find a specific dataset. However, a sitemap is provided where the user can search manually to point them in the right direction.
- The graphs are not interactive which makes it more difficult to understand the data.
- For a specific posterior it loads very high resolution images even if they are going to be scaled down due to the window size. This greatly increases the loading time for the charts and makes the site unpleasant to use on a data connection or in rural areas.
- The summary page has a very long list. These are hard to navigate quickly and do not provide a way to jump around as there are no categories or heading. It makes all the data flow together which increases the cognitive load on the user, making them focus on where to separate the data. Furthermore it is not scalable as the number of data points grows the page will become longer. A better user experience would be to paginate the query with a fixed page size and a previous and next button provided.

# Vision Statement

## Overview

Primarily for astronomers who need to visualise and share the results of their Bayesian Parameter Estimation, this toolkit will allow users to upload n*m size data sets and interactively display the results of Bayesian Parameter Estimation performed on the data. It will aid users in understanding and making observations from their data, as well as enabling them to easily share the results of their findings with colleagues. This toolkit aims to address issues common to all other currently available solutions, placing a heavy emphasis on dynamic interactivity, usability, and speed.

## Core Features

- The tool will be an open-source web application that can be used by anyone to visualise their Bayesian Parameter Estimation results.
- The tool will have interactive plots, where users can dynamically modify how information is being presented to them.
- Specific results can be easily shared via a URL
- Results can be saved after running the tool for quick and easy access later.

## Non-functional requirements

- The tool will have a key focus on the speed of returning, updating and loading saved visualisations.
- Any data stored by the app must be stored securely, as many of the end users will be researchers whose findings may be private.
- The core functionality of the tool will be contained in a publicly accessible API, separate from the tool itself.
- The app will maintain a high degree of usability.

## Who are the stakeholders?

### External Stakeholders

This toolkit's user base is those seeking to visualise the results of Bayesian Parameter Estimation, with the primary targets among them being gravitational wave astronomers; they will be the end users of the toolkit who will drive the key requirements and features of the development. Paul Lasky, a professor of Astrophysics at Monash University, will be the primary point of contact with these end users. Paul is the client and will help to shape the product vision and requirements.

Although it will be catered to the needs of astronomers, the application will be usable by all data scientists making use of Bayesian Parameter Estimation. This requirement drives the development team to ensure the tool is fit for general use.

Since the toolkit is open source, the wider community will be able to make contributions to it over time and repurpose its contents. As such, external stakeholders extend as far as generally interested persons.

### Internal Stakeholders

The Product Management team will shape the development process and project roadmap of the application. Their primary goal is to deliver a high-quality product and ensure the client and users are satisfied.

The Release Train Engineering team will handle the timeliness of the agile release train and the management of team meetings and PI planning. They are responsible for guiding the team towards SAFE practises such as CI/CD, project documentation, and risk management.

The Systems Architecture team will guide the usage of packages and software architecture used in the application. Their aim is to make sure that the back end has a strong foundation and reliable dependencies to ensure the robustness of the final application.

The Software development team will build the application based on feedback and input from the client, users, and the product management team. They will provide the technical insight for how difficult each feature will be, and be responsible for ensuring the source code is well designed and maintainable.

Avi Vajpeyi is the mentor for the student team developing this application. He will provide guidance for the team's progress and help to ensure the team is utilising best SAFE practices. He has an interest in delivering a high quality product to the client on time.

## How will it differ from competitors?

Most of the competitor applications make use of command line interfaces for generating results, which is less user friendly than going through a graphical user interface. They are also slow to use and only generate static graphs which can not be modified. This is a problem for users who need to visualise large quantities of data and interact with their results. The project application aims to address this issue by presenting a user friendly web application that can quickly process data. The application will generate dynamic graphs which will allow the user to process and visualise the data they have already inputted. It will also allow users to easily scan through data.

# Open Source Licences

An evaluation of software licences with focus on viability through permissions, limitations and conditions of the licence.

## Overview

**Table 2**
***Licences Overview***

| Licence | Permissions | Limitations | Conditions |
|---|---|---|---|
| **MIT** | • Commercial<br>• Modification<br>• Distribution<br>• Private Use | Authors are not liable for any claim | All copies or any substantial portions must include the original licence and copyright |
| **BSD** | • Commercial<br>• Private Use<br>• Distribution<br>• Sublicensing<br>• Modification | Patent grant must be done manually | Must preserve licence notice if redistributed |

| | | | |
|---|---|---|---|
| **GNU** | • Private use | Distribution, Modification and Linking is restricted | Must preserve licence notice if redistributed |
| **Apache** | • Commercial<br>• Distribution<br>• Modification<br>• Private Use<br>• Sublicensing<br>• Patent Grant | Trade mark grant is restricted | Any changes to original source code must require a notice of modification |
| **The Unlicense** | • Commercial<br>• Modification<br>• Distribution<br>• Private Use | Authors are not liable for any claim | None |
| **Mozilla Public Licence 2.0** | • Commercial<br>• Distribution<br>• Modification<br>• Patent Use<br>• Private use | Trade mark rights is NOT provided<br><br>Limitation of liability<br><br>Does not provide any warranty | A copy of the licence and copyright notice must be included with the licensed material<br><br>All modifications must be released under the same licence when distributing licenses |
| **Boost Software Licence 1.0** | • Commercial<br>• Distribution<br>• Modification<br>• Private use | Limitation of liability<br><br>Does not provide any warranty | A copy of the licence and copyright notice must be included with the licensed material in source form. |
| **Microsoft Public Licence (MS-PL)** | • Commercial<br>• Distribution<br>• Modification<br>• Patent use<br>• Private use | Does not provide trademark use<br><br>Does not product Warranty | License and copyright notice |

Note. Adapted from *Licenses*, GitHub Inc., 2022, https://choosealicense.com/licenses/, CC-BY 3.0

## Conclusion

Both the MIT and Unlicense provide the same permissions and limitations to all end users, while still allowing modifications for personal use. This permission will enable the addition of personal visualisation tools and plots along with the existing functionality of the application to support a broad range of uses of a visualisation application. However, in comparison to the Unlicense, the MIT licence offers the original authors the licence and copyright, which would be ideal for a project of this scale as recognition of a year long effort developed by an 18 person team.

Alternatively, both of the aforementioned licences are entirely open source in that anyone who acquires the software can reproduce/sell, among other things, copies of this software without limitation. If it is the case that the team has interest to make the software commercially available, then these licences may not be desirable. In this case the Apache licence may be preferable as it is still very permissive but has protections in place around reselling. This does come with a caveat as sharing the licence will take slightly more overhead.

Other licences do not offer much benefit that is relevant to this application and so will not be considered further for the final licence.

After discussion with the client it is clear that an open source licence is a key requirement for the application. As a result the MIT licence will be used for its wide range of permissions and non demanding conditions for future changes whilst maintaining credit to the original developers.

# References

GitHub Inc. (2022), *Licenses. Choose a License.* [https://choosealicense.com/licenses/](https://choosealicense.com/licenses/)