Dockerized WebTool for the Impact Framework

Product Vision and Architecture Document

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Version*** | ***Date*** | ***Authors*** | ***Comments*** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1. Purpose and scope of the document 2](#_Toc56416036)

[2. Product Vision and Requirements 2](#_Toc56416037)

[2.1 Business Goals 2](#_Toc56416038)

[2.2 Stakeholders 2](#_Toc56416039)

[2.3 The System Context 2](#_Toc56416040)

[2.4 Overview of Requirements 2](#_Toc56416041)

[3. Product Architecture 3](#_Toc56416042)

[4. Development and Evaluation Plan 3](#_Toc56416043)

# Purpose and scope of the document

*describes the purpose and scope of the document, its intended audience, the approach followed to elaborate this document (e.g. what method was used, who was consulted) and the status of the document.*

# Product Vision and Requirements

## Background Information

Software's environmental impact is multifaceted, including greenhouse gas emissions, water usage, and energy consumption. To address these issues, the Green Software Foundation, a non-profit under the Linux Foundation, is dedicated to reducing software's ecological footprint through various projects.

A key project of this foundation is the Impact Engine Framework (IEF), which has evolved throughout the Green Software Foundation's existence. A notable development is the creation of the Software Carbon Intensity (SCI) Specification, which defines a methodology for calculating a software system's total carbon emission. Initially, the challenge was data acquisition, leading to the launch of a project focused on generating necessary data sets. The team later realised that various existing data sources were available. They anticipated that the challenge would shift from sourcing data to selecting the appropriate data set for each use case. This initiative eventually evolved into the SCI Guide, which provides comprehensive documentation on existing data sets and their application.

The project's current phase aims to establish formal standards and tooling, transforming software measurement into a disciplined and widespread practice. The SCI Specification is positioned as the fundamental standard, while the Impact Framework serves as the essential tooling component.

## Business Goals

*Describes the main business goals, business drivers, and constraints for the product.*

*The goals are sorted in the descending order of importance and priority.*

|  |  |  |
| --- | --- | --- |
| Goal | Details | Explanation/Justification |
| Optimisation | * Provide users optimisation suggestions / solutions for their input configuration. * Demonstrate trade-offs between (financial) cost, performance and environmental impact to users. | Giving users with an idea of improvement should be one main purpose of this framework. Users should not only know about the environmental impact for their software, but also have an idea of how to make it better. |
| Comparison | * Enable users to compare their input configurations in parallel, explicitly highlight those significant differences. * Compared data should not only come from the input parameters, but also include the simulation results’ output, and output that might produce from middle-process. | Comparison between different options should be a frequently demanded functionality. Especially for software developers / companies. In reality, environmental issues are not the only thing considered by stakeholders, but also cost and performance. Giving users a comprehensive understanding of the difference and trade-offs between their technical options means something more to them. |
| Automation | * Automate the simulation procedure as more as possible, minimize unnecessary manual operations. * Only crucial operations should require decisions from users. * Automation of deployment should also be taken into account. | * The entire IEF and optimisation process should ideally require as little manual operations as possible * So we can promote IEF to a wider range of target audience. |
| Versatility/Flexibility | * There should be inputs for multiple parameters such as location, time, etc * Allow more flexibility when it comes to generating statistics and ideas for reducing emission | * It does not exist an universal solution for all softwares * So users can choose what parameters they wish to optimise * Give more flexibility and can also help the simulation to generate better suggestions |

|  |  |  |
| --- | --- | --- |
| Goals | Details | Explanations/Justifications |
| User-friendliness / Readability | * Allows users with non-technical background been able to understand everything in the app. * No technical details should be exposed to users. Manipulation on technical operations (e.g. manifest file editing) should be abstracted and wrapped. | * Clients have emphasized this point since the first meeting with them. Although their main points seems shifted now, I believe this is still one goal they’d want to achieve. * User friendliess is one essential point of attracting and keeping the users |
| Expand the User Base | * Collaborate with external organisations such as educational institution, government organisations, or corporations * i18n: Multi-language support | * Clients have mentioned this aspect many times as well. * User base is the foundation of a software system * There’s a lot to dig here, a very broad goal |
| Enhance the Accuracy | Improve the accuracy of the existing IEF framework by providing more reliable models and information, | It is crucial to improve the accuracy of the existing IEF framework in order to better reduce the carbon emission |
| Expand the Scope | * One of our primary goals is to expand the scope, including the scope of user adoption, the scope of the sources of data used in IEF. * Currently the system only support Azure as cloud provider, but we would aim to expand to AWS, GCloud, and other cloud providers.   This would improve the utility of IEF | This goal links to the requirements client proposed last week. Making this system adapt to a wider range of cloud provider could wider their business scope. This is very important to clients. |
| Improve user stickiness | Convincing to users to stay, persuade them to continue using the framework to reduce emissions | If the users dropped using IEF after their first tries then the effects of emission reduction would be limited |
| Deployability | Deployment procedure should be as simple as possible.  Minimize the difficulty of introducing new models into IEF. | * Better deployablility makes users more willing to deploy their product * And receive less query for technical assistance |
| Data security | Assure the users that IEF would not exploit their data privacy.  Data confidentiality and integrity should be carefully protected. | Since IF would require to scan user’s device, privacy and data security could be the concern of the user, especially for larger organisations such as private company or government agencies |
| More Functionalities | To further reduce emissions and to cater the needs of the users | To attract more users and to better reduce emission |
| [Optional]  Automation | Automate the adjustment of key parameters based on environmental impact goals so that the users don’t have to manually tweak settings, making the process more efficient and less error-prone | * Easier for users without programming background to use IEF * Reduce human errors |

## Stakeholders

*Describes the main stakeholders and their interests.*

The IF web tool has the following stakeholders.

| **Stakeholder** | **Concerns, wishes and expectations related to the project** |
| --- | --- |
| Anyone involved in software develop & deploy process | * They will use the framework to assess and optimize the environmental impacts of their applications. * IF will impact the configurations of their app, like their design and code. It will drive them to use more efficient algorithms. * They hope that the framework is user-friendly and intuitive. |
| Green Software Foundation | * GSF may concerns the development and operation costs, reputation and strategic objectives. * The results from the IF can provide GFS with crucial insights on how to adapt this technique to other projects related to the environment. * GSF may expect that the tool can be adopted widely and make a significant impact on reducing the software emissions. |
| End users | * End users may indirectly benefit from more environmentally conscious software development practices. * They hope that their data will be used by framework responsibly, ethically, and with respect for privacy. |
| NGO | * IF may impact their promotional strategies. * They hope that the framework would have a meaningful impact on the environment and could raise public awareness of the environment. |
| Environmental Engineers / Researchers | * Their works or studies will be affected by the data based on the result generated by the framework. * They expect the more precise accuracy of the simulations and calculations of IF to reflect the environmental impact of software truthfully. |
| National and local government politicians | * The data and insights gained from the framework will affect their policy and regulatory standards related to environmental sustainability. * They hope the framework be ready before next election. |
| UCL Team | * Develop and implement user-friendly solutions to help users simulate, measure, and monitor the impact of the software on the environment. Allows users with different backgrounds to be able to understand everything in the framework. * They hope that the IF can convince users to continue using it. * They care about the scalability, resilience and efficiency for updating the existing IEF. |
| Large corporations | * They might use the environmental impact of their software as a selling point. * IF can help them find ways to optimise the energy efficiency of their software, potentially saving money on energy consumption. * They care the availability of IF since they expect being able to access the framework on all environments and platforms with ease. |
| Open-source community | * They care about whether all contributions from the project comply with open-source licenses to foster transparency and community involvement. * They hope that all developers in the open-source community can be able to contribute and benefit from it. |
| Law enforcement agency | * IF can influence the awareness of them and help them develop the culture of environmental responsibility. * Since IF would require scanning a user’s device, privacy, and data, law enforcement agency, especially for larger organisations such as private companies or government agencies, should concern the security of IF. |
| Energy supplier | * The success of this framework may cause negative impact on their profit. * They may expect IF to provide insights into the future demand for energy based on the result of the framework. This can assist them in planning and optimizing energy production and distribution. |
| Cloud Service Providers | * They will provide essential information such as the energy consumption, server utilization of various computing environments for the framework. * IF may affect the infrastructure or resource provisioning of cloud service providers. * They may expect to integrate the IF functionalities into their platforms to support their customers in measuring environmental impacts. |
| Tech Hardware Companies (like Intel, AMD, GPU makers) | * They will provide detailed information such as the power consumption metrics of various kinds of hardware for the framework. * IF may affect innovation in new hardware design, and manufacturers may focus more on developing products that reduce environmental impact. * They may expect IF to provide valuable insights into how their products impact software systems in term of environmental footprint. |
| Competing framework or tools | * The IF framework may affect their users’ stickiness. They don't want IF to be the first choice for users who need environmental impact assessment framework. * They expect that this competitive relationship can foster motivation for improvement, driving each other to enhance their projects. |
| Educational Institutions | * This project may update the content of their teaching and change the teaching tools they use. * They hope this framework can be a useful tool for their teaching in the areas of environmental science, sustainability or software engineering. |
| General Public | * This framework may stimulate the production of more sustainable software, reducing the impact on the environment. * IF will increase their demand for green technology solutions, which can drive innovation and development of other environmental technologies. |
| Future Generations | * IF aims to create a healthier planet for future generations, so IF will leave a more sustainable environmental legacy. * They want to live in a better, safer place that doesn't waste resources. |

## The Product Context

## Core Features of the System

**1. Measurement Model**

* Extend the IEF by developing a flexible measurement model for software environmental impact
* Extend the capabilities of the existing Software Carbon Intensity (SCI) specification to offer a flexible measurement model for software environmental impacts
* Easily calculate software impact metrics, such as carbon, water, or energy without the need for writing any code

**2. Simulation Tool**

* Create a simulation tool to project and analyse the environmental impact during the design phase, enabling users to make informed choices
* This feature gives users the ability to optimize their software design decisions by evaluating different scenarios, based on their budget, context, and other business constraints
* Allow users to easily change the model to represent “what if” scenarios, reducing the cost by validating different software architectures without making any changes

**3. Modular Design**

* Modular design will allows for the integration of custom models or selection from a range of pre-existing models, knows as plugins
* Follow the IF design philosophy that all models should be plugins, so that the configuration is as modular as possible
* Provide clear separation between the engine and models

**4. Integration Capability**

* Seamlessly integrate with various environments, such as cloud, bare-metal, and virtualized systems, to assess software impacts comprehensively

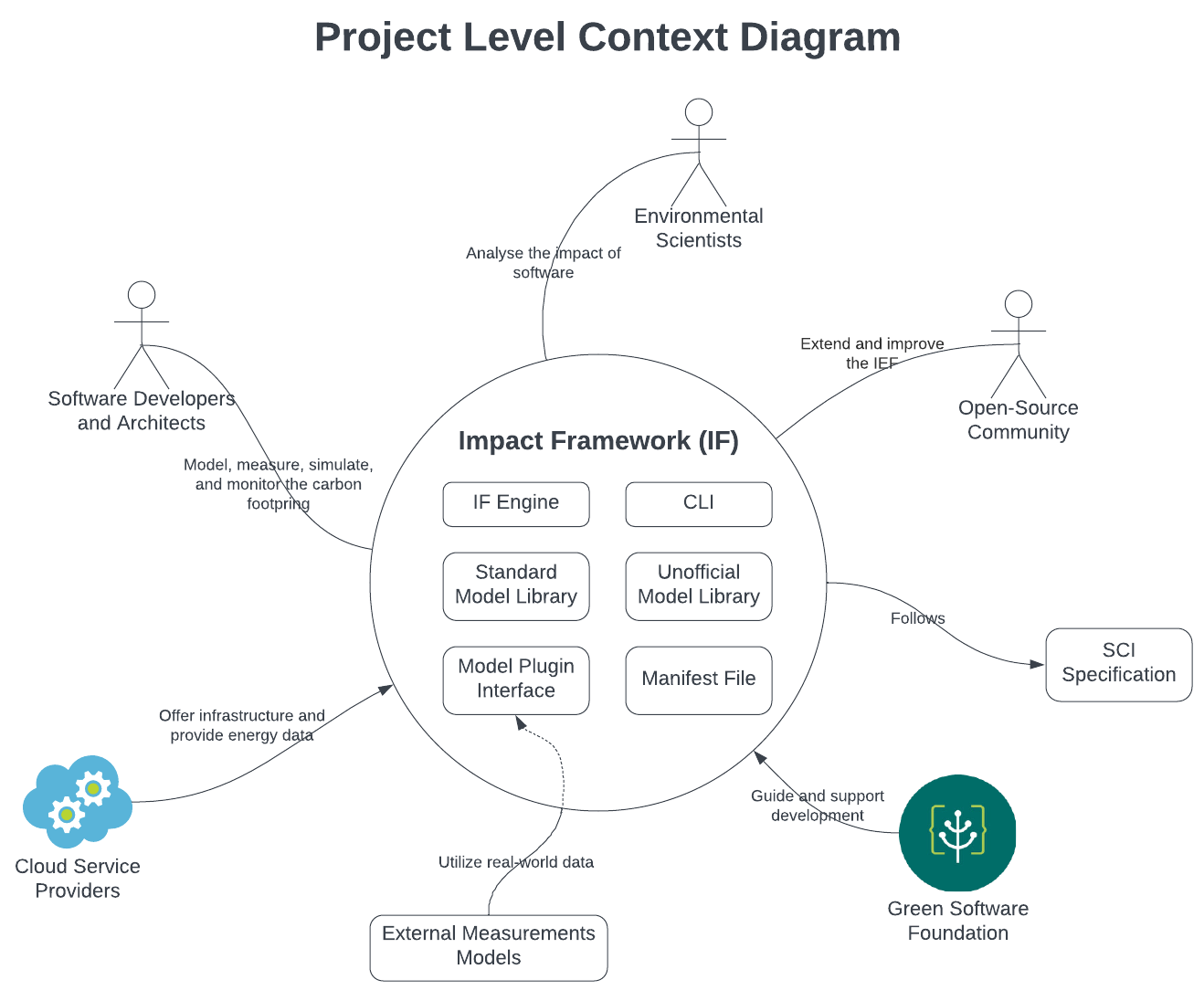
**5. User-friendly Interface**

* The design is easy-to-use, intuitive and accessible for software engineers and environmental scientists, encouraging widespread adoption
* Simple installation via *yarn* or *npm* package managers
* All that is needed, is to define the scope and parameters in a simple YAML manifest file, known as impl, the IF handles the rest

**6. Compliance**

* Adhere to the guidelines and objectives of the Green Software Foundation, promoting sustainability in software

Context diagram of the system as-is:



*Context diagram describing the world in which the system operates and the interactions of the system with users and other systems.*

## Overview of Requirements

*A summary of the main architecturally significant functional and quality requirements for the system.*

*Do not include detailed requirements - if you have detailed requirements such as gherkin scenarios you can present them in an appendix, or better in an online repository with a link.*

*Section 2 can include additional subsections relevant to the business analysis of the system under study, for example a domain conceptual model, description of workflows, domain scenarios, a risk analysis, reference to standards, analysis of competitors, technology opportunities, etc.*

# Possible projects

## Web UI – Simulation tool

The original idea was to create a web UI for the IEF that would also serve as a simulation tool. By providing an accessible and user-friendly interface for the Impact Framework, which currently operates through a command line, the team aimed to help users model, measure, simulate, and monitor the environmental effects of their software without the need for high level programming knowledge. The IF tries to tackle a very difficult and complex issue: the environmental impacts of modern software, which often runs on diverse environments and involves numerous components .In turn the web UI would be able to simplify the use of the IF even more by adding another tool, an abstraction level above it so that users don’t have to interact with the IF at all. By simplifying the input process, and enhancing the output visualization, it would be possible for users to try different inputs and scenarios to find the one with the smallest environmental impact.

Unfortunately, this option is not viable due to the maturity level of the project. The IEF has just come into alpha and everything a UI will rely on (types and number of inputs) will change in the next 6-12 months.

## AWS model

There is currently an unofficial model for the IEF called Azure-importer that allows you to provide some basic details about an Azure virtual machine and automatically populate your IMPL input file with usage metrics that can then be passed along a model pipeline to calculate energy and carbon impacts.

A proposed project idea was to create a similar model for the AWS cloud platform to measure the environmental impact of VMs or other cloud services provided by them. Such a model would help expand the IEF userbase with AWS users which would be extremely significant since AWS is the current market leader for cloud infrastructure with a 33% market share. After discussion with the IEF development team though it was decided that this was not a viable project since AWS does not expose similar APIs to Azure to get the required metrics. Thus, it would not be possible to provide a working model within the available timeframe.

## Carbon Aware Computing Simulation tool

A model which if added to a pipeline would shift a workload to the least carbon moment

<More information is awaited from the Green Software Foundation Developers>

## Right Sizing Simulation tool

Most users pick servers that are far too powerful for their needs, so they run at low utilizations.

Right sizing is the task of picking a more appropriate server.

A model which helps figure out the right sized server to use given the workload utilization.

<More information is awaited from the Green Software Foundation Developers>

## Parameter Optimization Model using Search-based Software engineering

The team is tasked with developing a new model that seamlessly integrates into the current model pipelines. This model will be incorporated following the data import models, like azure-importer, but preceding the models responsible for converting energy measurements into various impact metrics, such as carbon emissions.

* This option requires the implementation through a model and not through an external tool that invokes the impact engine for two reasons:  
  The creation of models for such functionality aligns with the IEF team’s way of working and best practices and it was strongly advised by them to pursue the model approach
* The data that will need to be manipulated rest in the middle of the calculation pipeline so a tool before or after the impact engine would not have access to that.

Utilizing established search-based techniques and similar heuristics, the model aims to conduct efficient explorations and identify parameter combinations within acceptable limits. The ultimate goal is to minimize the impact metrics as desired.

The first and most obvious parameters to be optimized would be location and time but further investigation might reveal more parameters that could be optimized using this method.

# Product Architecture

## System components

System as-is components:

* **IEF CLI:**

The only interface currently available for interacting with the IEF to utilize its functionalities.

* **IEF Impact Engine:**

Performs impact computations based on the input IMPL file and generates a Manifest file containing the computed impact.

* **IMPL Manifest File:**

A formal report in YAML format that details all assumptions, inputs, and models used for calculating the impact. It describes the system to be measured and serves as an input.

* **Model:**

A model converts an input into a specific output impact metric.

* **Model Pipeline:**

A chain of models. Calculating a component's impacts often requires using multiple models in sequence. Each model takes as input the outputs of the previous model in the chain, all working together to calculate impacts from inputs.

* **Standard Library of Models:**

A standard library of models built and maintained by the IF core team.

* **Unofficial Library of Models:**

A separate repository for models that community members maintain.

* **Model Plugin Interface:**

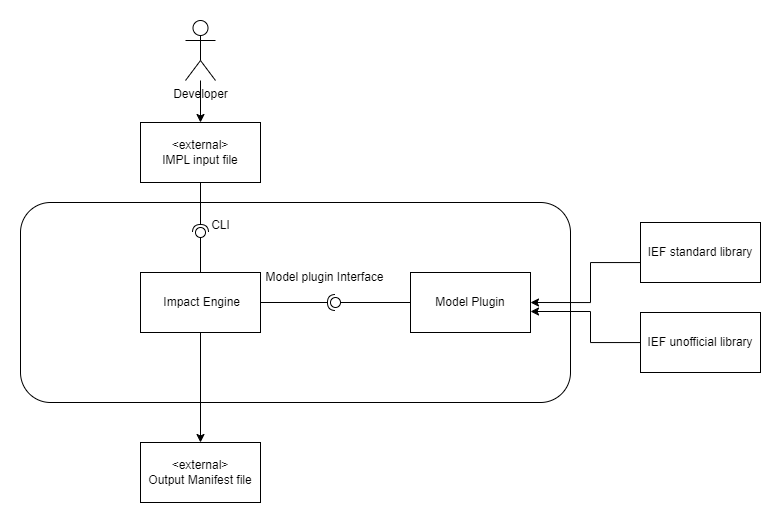
A common class interface that every Model Plugin needs to extend and implement.

* **Output Manifest File:**

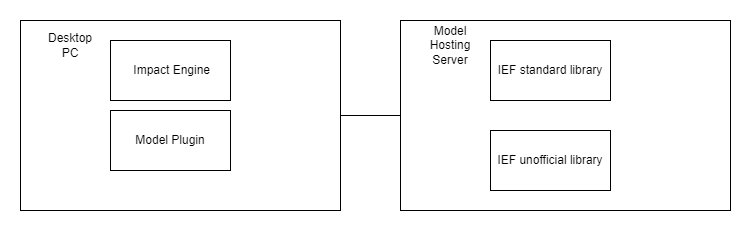
A formal report in YAML format detailing not just the end impact but also all the assumptions, inputs, and models used in calculating the impact. (Can be in a CSV format as well.)

## Functional Views

Functional view of the system as-is:



The deployment viewpoint of the system as-is:



*The structure of this section may vary from group to group.*

*We expect all reports to include:*

* *a functional view of the system with clear description of functional elements, their responsibilities and interfaces;*
* *a deployment view if the system is composed of multiple nodes*
* *a discussion of how the architecture supports all important system qualities outlined in Section 2.4.*
* *a discussion of trade-offs and key architectural decisions*

*The section can include other architectural views that are relevant to describe the system and rationale for important architectural decisions.*

*You can select sections from Nick Rozanski and Eoin Woods' architecture description template, available on Moodle, that are appropriate for the analysis of your systems. You can also take sections from the structure of design documents at Google, also available from the course Moodle page.*

*See the marking sheet for information to include in this section.*

# Development and Evaluation Plan

*This section will define your development and evaluation plans for the term 2 project. How will you split the term 2 project into small increments? How will you test and evaluate the success of your project?*

*What you intended to develop in term 2 may be a subset of the requirements and architecture components defined in the previous section. This subset must be a self-contained product. Make sure your description of your development plan is consistent with and make clear references to requirements and architecture elements in the previous sections.*