



# Explaining Sustain

## I. Analyzers

This folder of sustain analyzes the folder where the npm package is downloaded in.

It does the following:

- Analysing the project scope and idea → saves scope in a nice temp file to refer to later for other checks, running scope again will overwrite
- Analyzing the quality of code from function names
- Analysis the security aspects of code
- Analyses any spaghetti code

## II. Simulation

This automatically runs the analyzer.

If analyzer has been run → cache exists

It will read from cache the scope and project idea and it will estimate the "burdens" of this scope on the environment. Mostly to be calculated as KWh or Wh for the scope for the project.

## Estimations Per Score & Examples

## **Website**

Server & DNS:

A website needs a server to take all requests, a server strong enough to handle a DDoS attack a server new enough to load the website as fast as possible while keeping latency low.

Otherwise you can use a service that can provide both

Users:

Calculate energy used per user to load website for number of pages considering its the only page they will use and the buttons they push wont open a new page

We use a firefox estimate with okay-ish resources

AMD Ryzen U - processor

intel i5 U - processor

M2 apple processor

Scaling up users:

10 → 10k concurrent

distribution: 65% windows, 25% macos 10% linux

Macos can only have an M2

## **AI**

Inference costs

Training costs

Upkeep costs

embodied footprint of SoTA Technology used for application

## **Gaming**

# Automation

## **III. Collectors**

- composers: analyses the composer in downloaded folder root to see
- docker: goes to docker to see the current active ones and see how much resource is being used