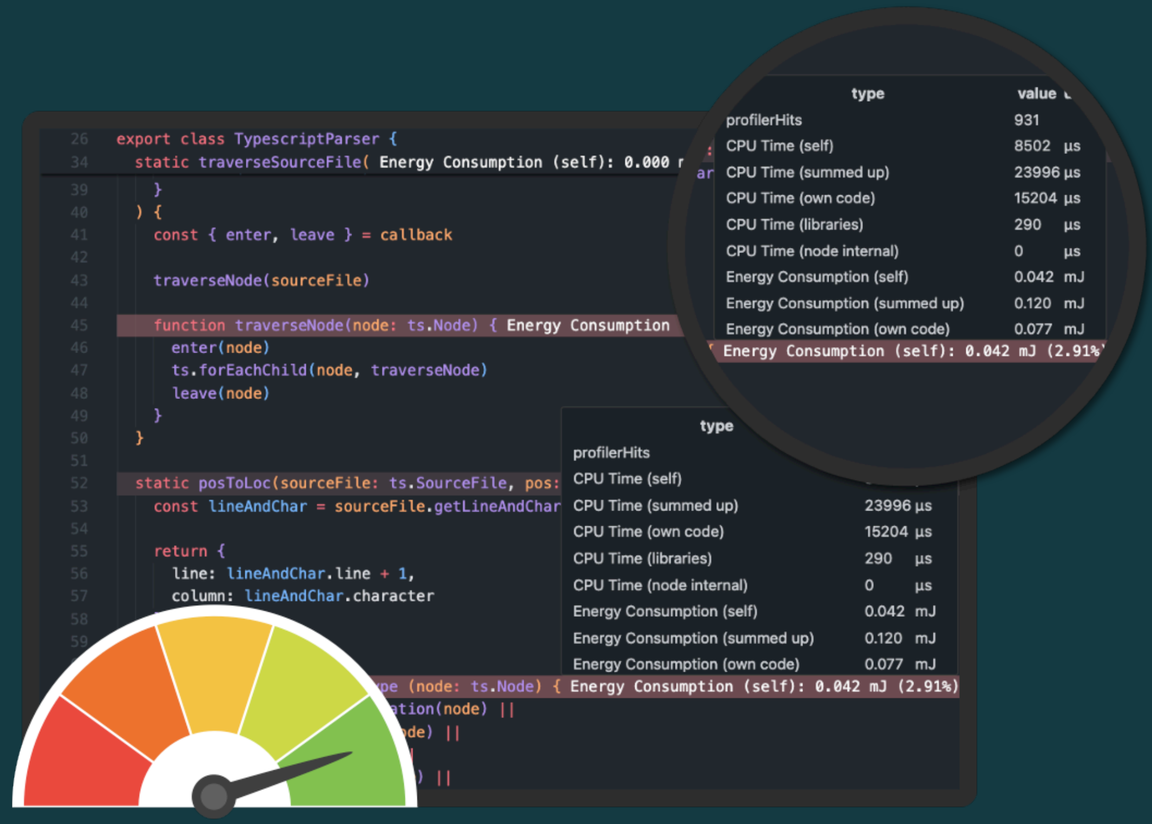
oaklean

Green your Code

Energy profiling for JavaScript & TypeScript



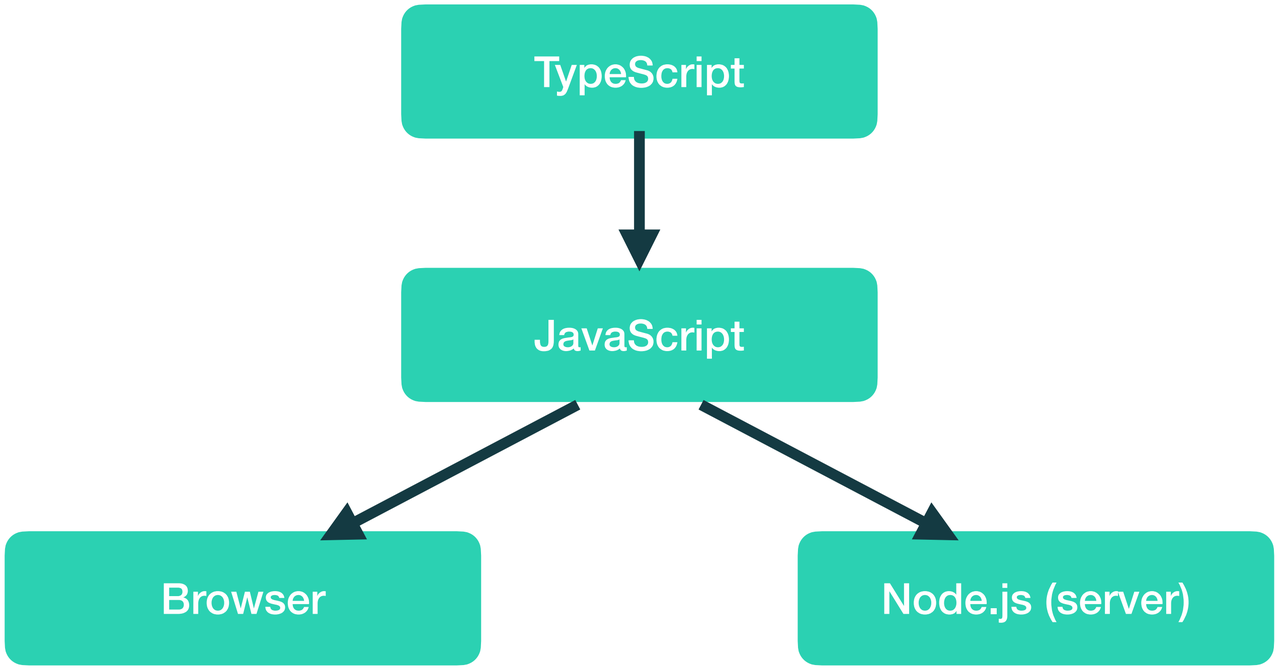
www.oaklean.io

Open Source MIT License

The Hidden Cost Of Every Line Of Code

- Developers usually focus on performance, readability, maintainability, but rarely on energy efficiency
- Every code choice (libraries, loops, APIs) affects energy footprint
- Existing tools don’t measure energy in Node.js/JS/TS apps
- Oaklean: makes JS/TS energy usage visible and optimizable

JavaScript/TypeScript/Node.js



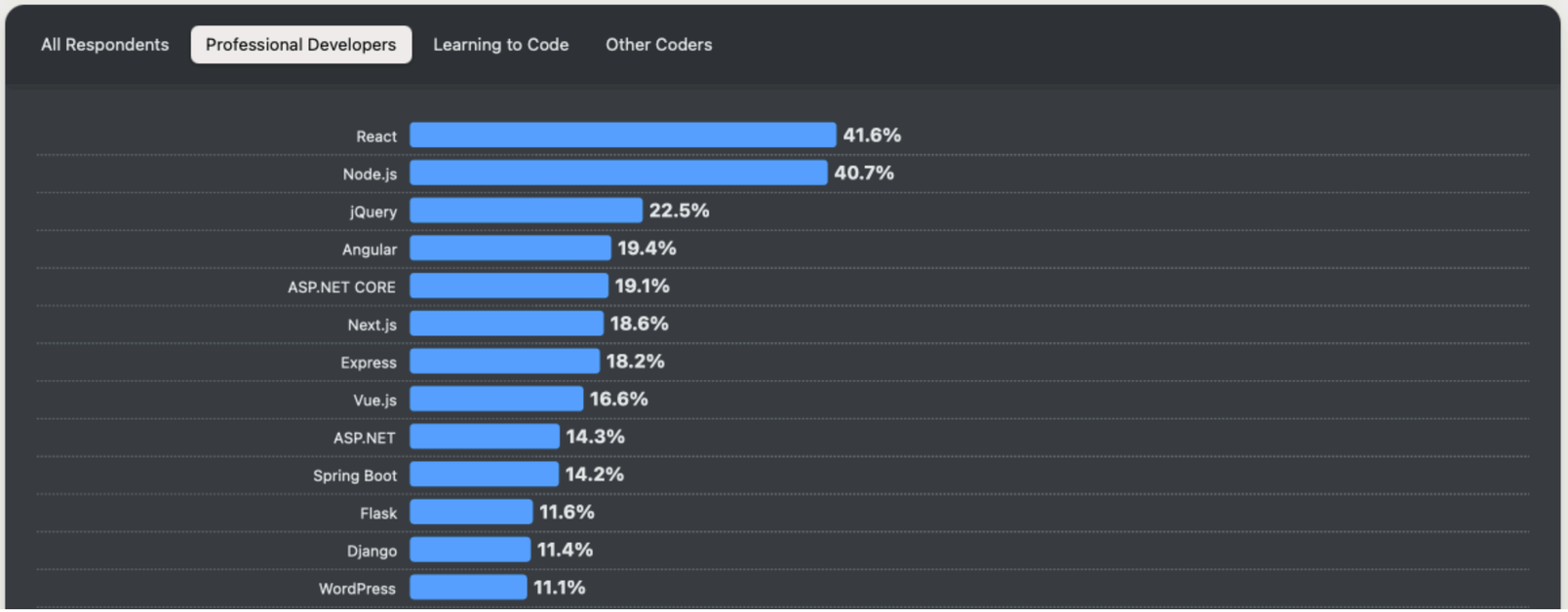
Node.js At Scale = Energy At Scale

Why JavaScript/TypeScript Ecosystems Especially Need Sustainability Awareness

Web frameworks and technologies

Node.js peaked in 2020 with its highest recorded usage score of 51%. While not as popular, it's still the most used web technology in the survey this year and has increased popularity among those learning to code from last year.

Which **web frameworks and web technologies** have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.)

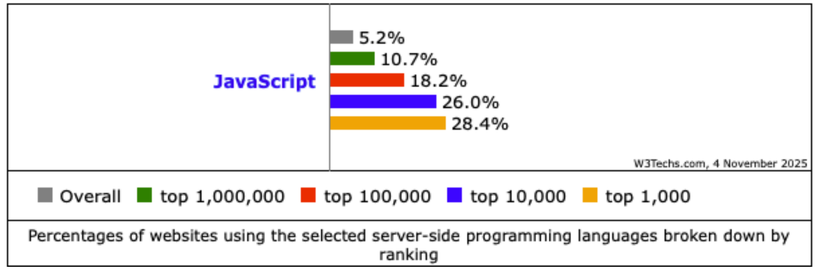


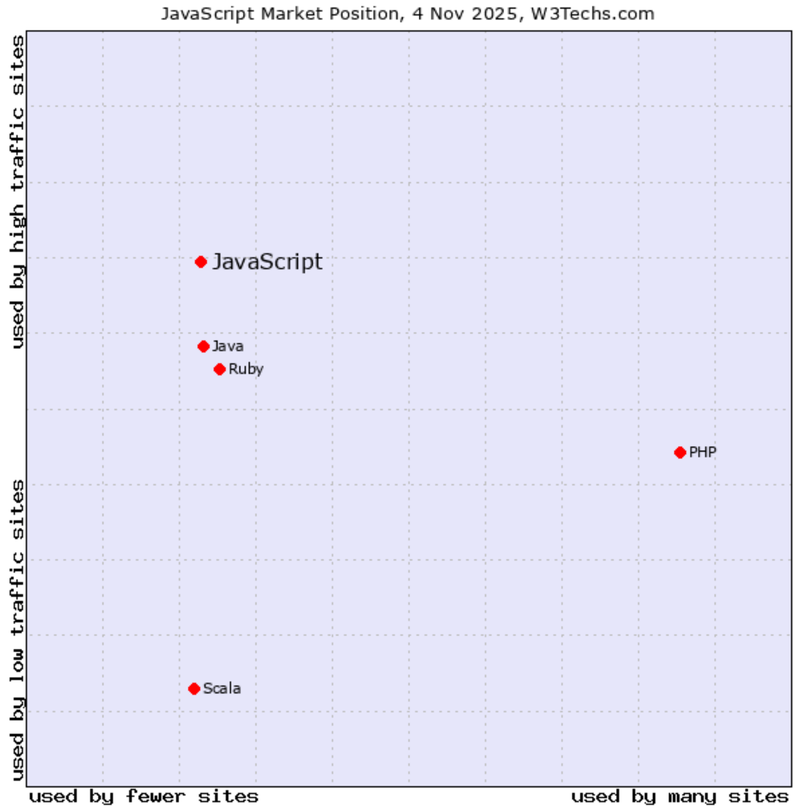
https://survey.stackoverflow.co/2024/technology

Node.js At Scale = Energy At Scale

Why JavaScript/TypeScript Ecosystems Especially Need Sustainability Awareness

- Node.js used as website backends:
  - 5.2% of all websites
  - 10.7% by the top 1,000,000 websites
  - 28.4% by the top 1000 websites





https://w3techs.com/technologies/comparison/pl-js

Node.js At Scale = Energy At Scale

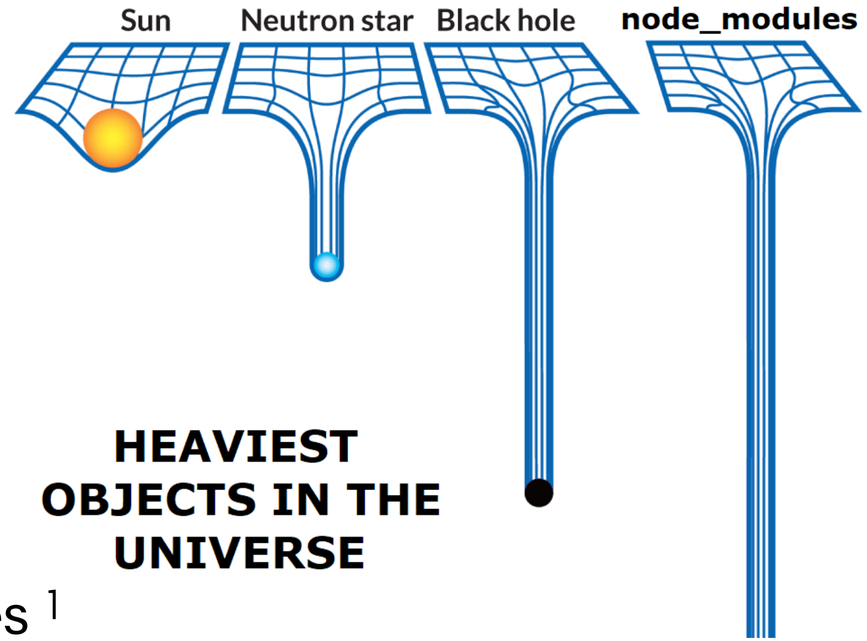
Why JavaScript/TypeScript Ecosystems Especially Need Sustainability Awareness

- **Paypal**: Adopted a unified JavaScript stack for frontend and backend
- **LinkedIn**: Migrated from Ruby on Rails to reduce overhead and server load
- **Netflix**: Server-side rendering and backend for global streaming
- **Uber**: Needed a high-concurrency backend to handle live GPS, surge pricing, and updates
- **Walmart**: Required event-driven architecture to handle traffic bursts
- **Trello**: Real-time UI updates for collaborative task management
- **eBay**: uses Node.js for parts of their large-scale marketplace backend

Node.js At Scale = Energy At Scale

Why JavaScript/TypeScript Ecosystems Especially Need Sustainability Awareness

- Dependencies = node_modules
- 1.3 millions in January 2021
- 2017-2019: 700 new packages per day
- Many redundant packages:
  - 61% of the dependency functions were duplicates [1]
  - 10.4% were clones of other packages [2]
  - 17.92% are trivial (low/no functionality/only data) [3]



[1] https://habr.com/en/articles/554334/

[2] What the Fork? Finding Hidden Code Clones in npm

[3] https://arxiv.org/abs/2510.04495

Meet Oaklean

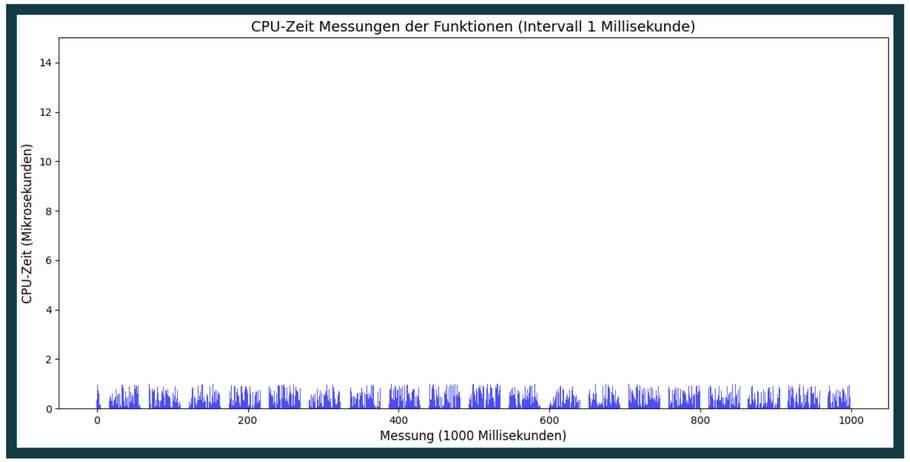
Measuring Energy In Your Codebase

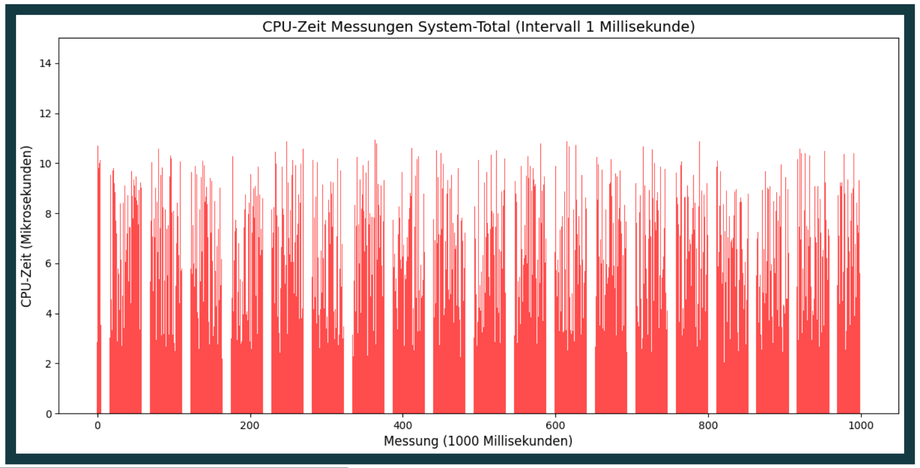
- Open-source tool by Hitabis forJavaScript/ TypeScript - Node.js
- Profiles CPU and memory energy consumption **per function or component**
- Reports energy cost directly in your workflow
- Empowers developers to code with environmental awareness

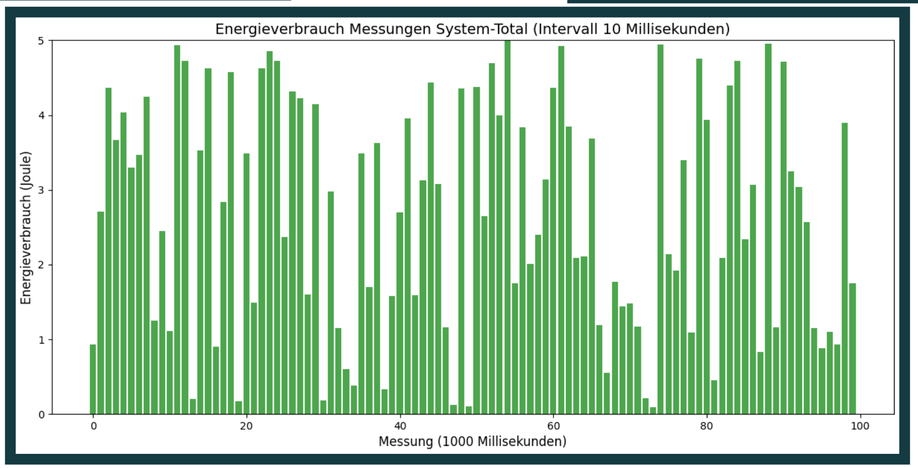
Demo

How Oaklean Works Under The Hood

Measuring Of CPU Time And Energy Consumption (CPU + RAM)



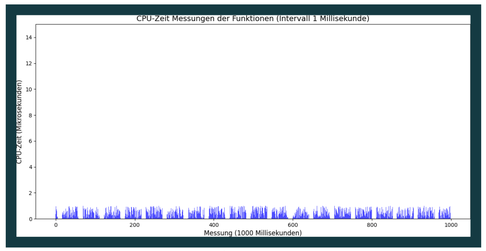


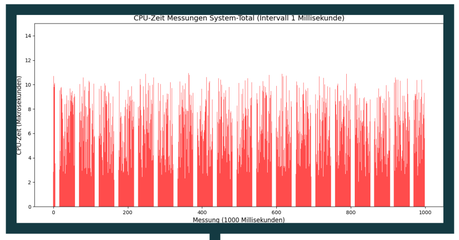


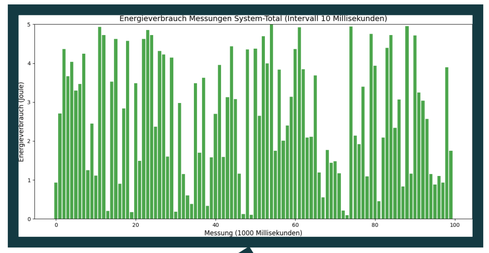
Collects low-level CPU and RAM energy data

How Oaklean Works Under The Hood

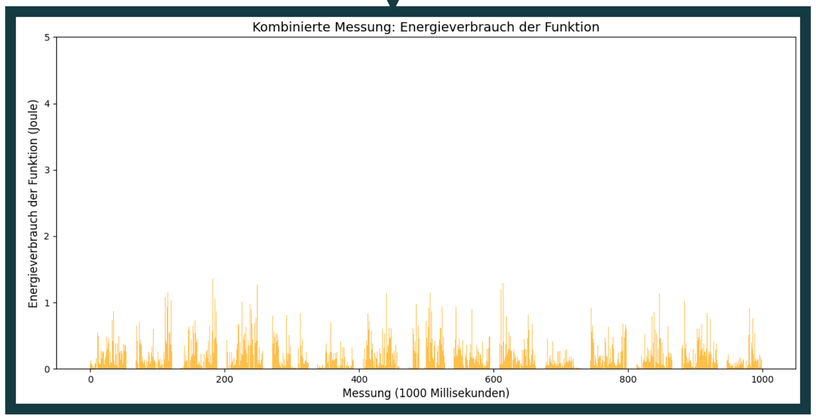
Correlate Measurements





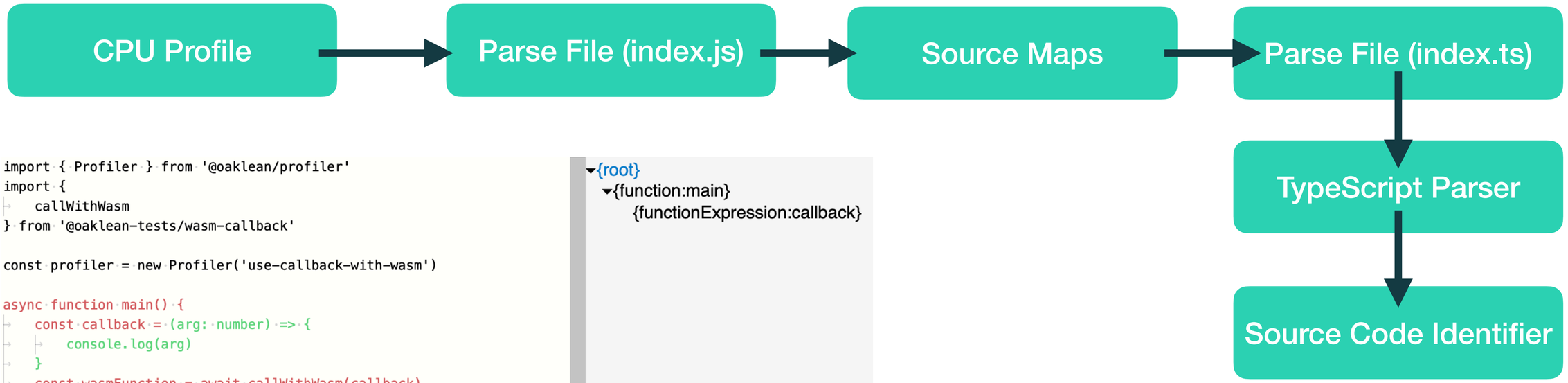


$$\text{Energy Consumption of Function} = \text{Energy Consumption of System} \times \frac{\text{CPU-Time of Function}}{\text{CPU-Time of System}}$$



How Oaklean Works Under The Hood

Parsing The Source Code And Cluster Into Components



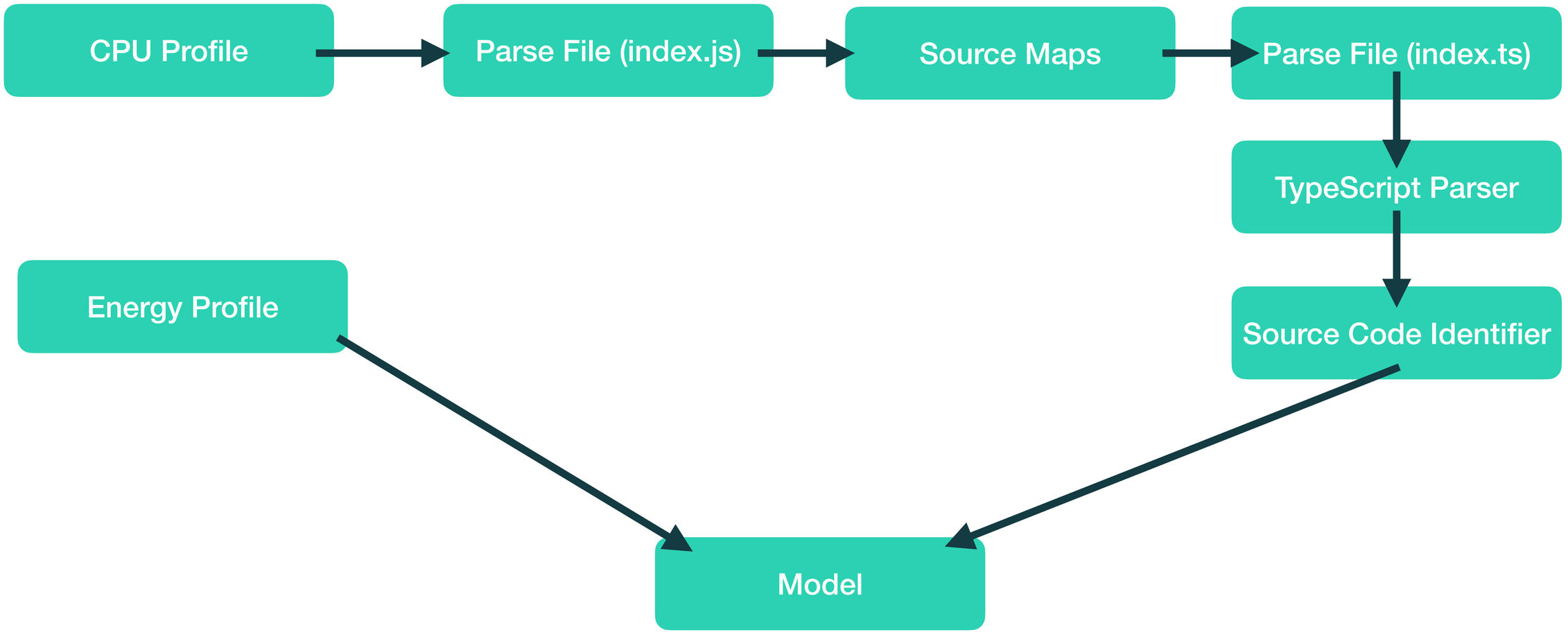

```
import { Profiler } from '@oaklean/profiler'
import {
    callWithWasm
} from '@oaklean-tests/wasm-callback'

const profiler = new Profiler('use-callback-with-wasm')

async function main() {
    const callback = (arg: number) => {
        console.log(arg)
    }
    const wasmFunction = await callWithWasm(callback)
    const title = 'latest'
    await profiler.start(title)
    for (let i = 0; i < 100; i++) {
        wasmFunction(i)
    }
    await profiler.finish(title)
}

main()
```

```
{root}
  {function:main}
    {functionExpression:callback}
```

{root}.{function:main}

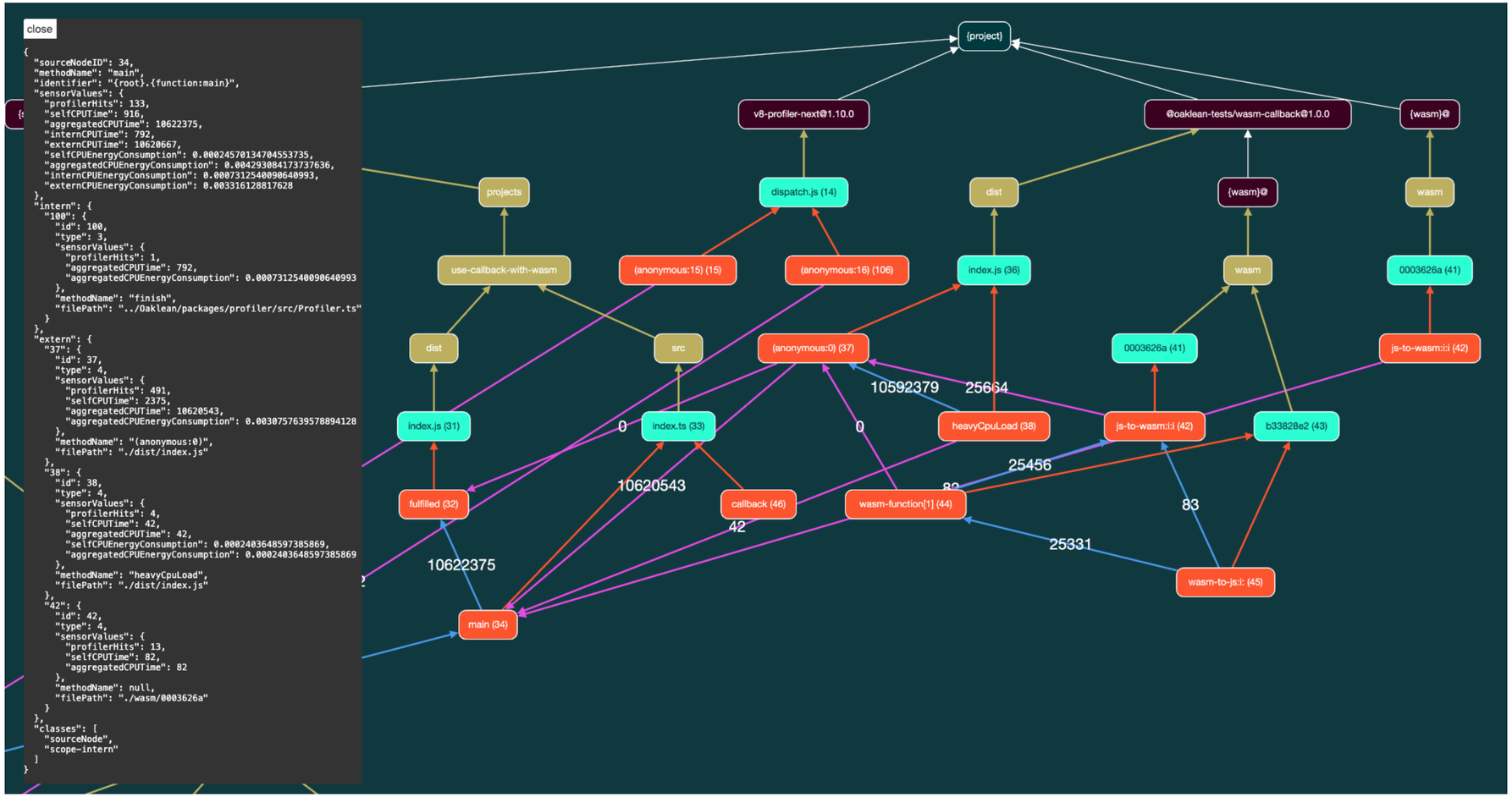
How Oaklean Works Under The Hood

Annotates Source Code With Cost Metrics



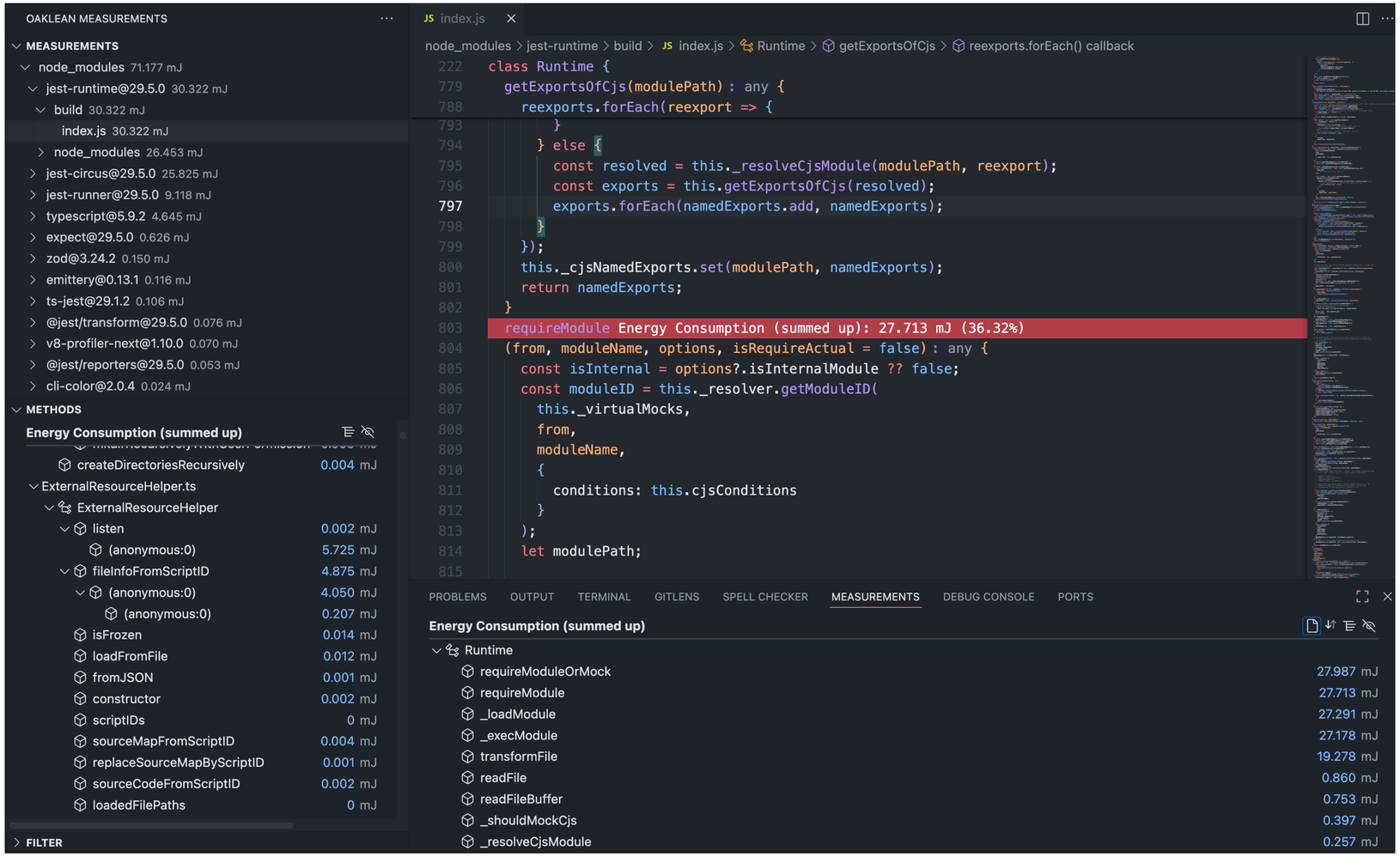
How Oaklean Works Under The Hood

Oaklean's Internal Model Representation



How Oaklean Works Under The Hood

Visualization In VS Code

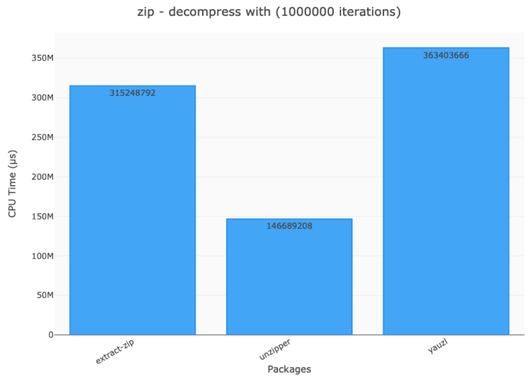


Case Study

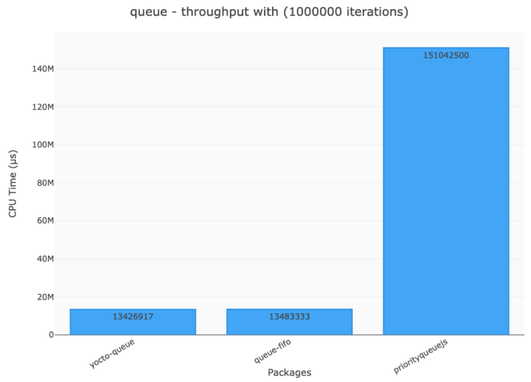
Node Module Comparison

cpu time

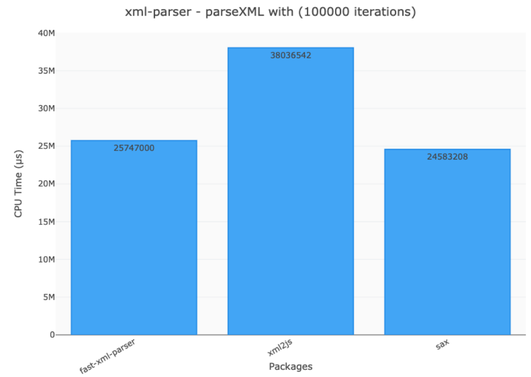
zip-decompress



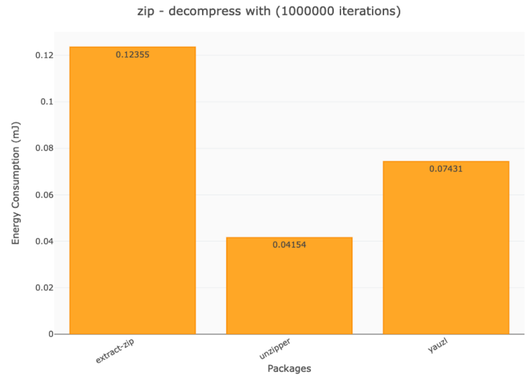
queue-throughput

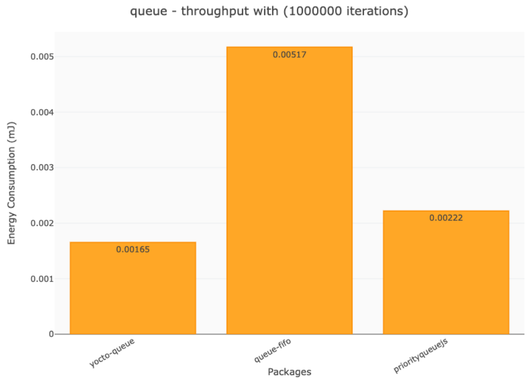


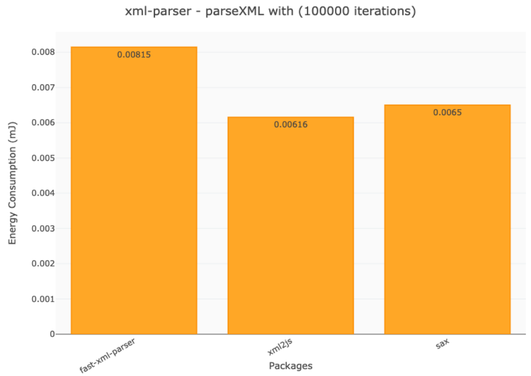
xml-parser



energy consumption







A Community Vision

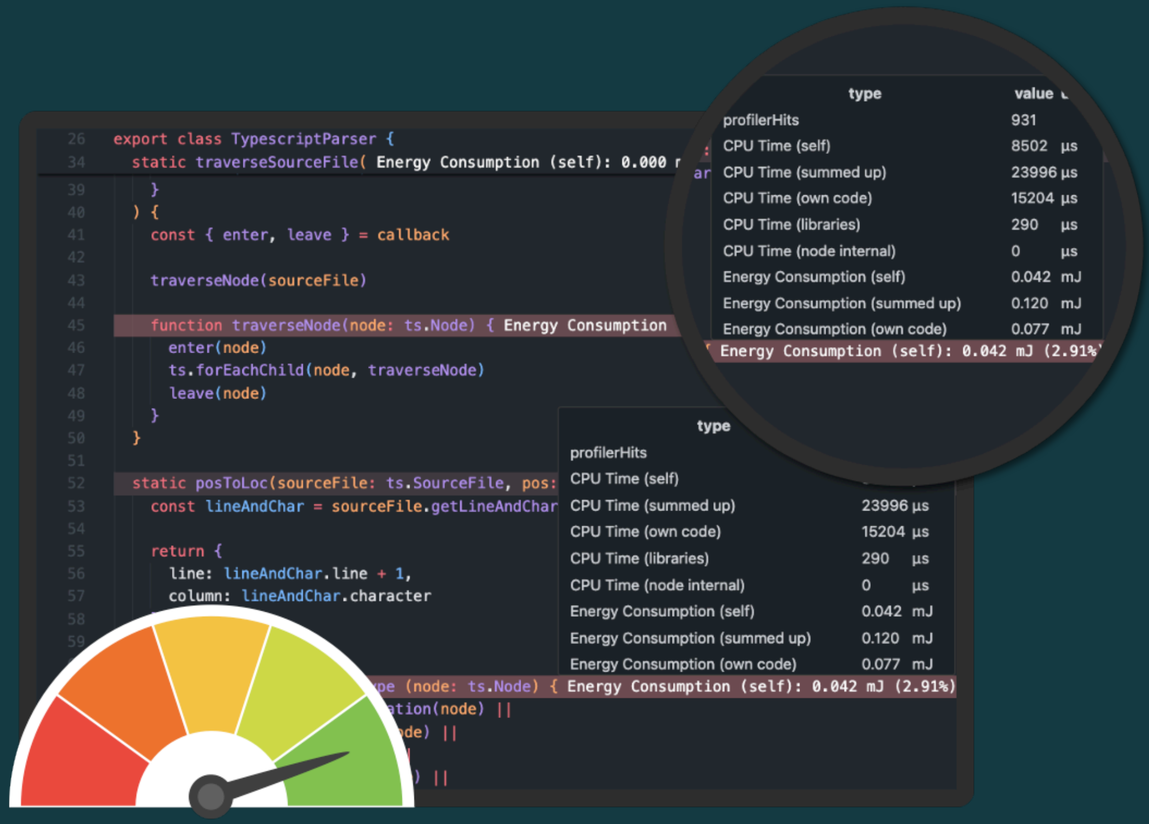
The Green Library Database

- Next step: build an open database of library footprints
- Share benchmarks: “How green is your dependency?”
- Encourage community contributions and comparisons
- In editor suggestions for „greener“ dependencies
- Contribute on GitHub: tools, metrics, libraries

oaklean

Green your Code

www.oaklean.io



Supports: Linux/Mac/Windows

Open Source MIT License

Questions & Reflections

Sources

- Stack overflow developer survey
  https://survey.stackoverflow.co/2024/technology#most-popular-technologies-webframe-prof
- Comparison of the usage statistics of JavaScript for websites
  https://w3techs.com/technologies/comparison/pl-js
- Why Top Companies Are Using Nodejs As A Backend
  https://enstacked.com/why-top-companies-are-using-nodejs-for-backend
- Top 10 Large Companies Using Node.js for Backend
  https://medium.com/devmap/top-10-large-companies-using-node-js-for-backend-f32aa3e55cdd
- „What the Fork? Finding Hidden Code Clones in npm“
  https://ldklab.github.io/assets/papers/icse22-shrinkwrap.pdf
- Only 39% of the functions in node_modules are unique in the default Angular project
  https://habr.com/en/articles/554334/
- Detecting and Characterizing Low and No Functionality Packages in the NPM Ecosystem
  https://arxiv.org/abs/2510.04495