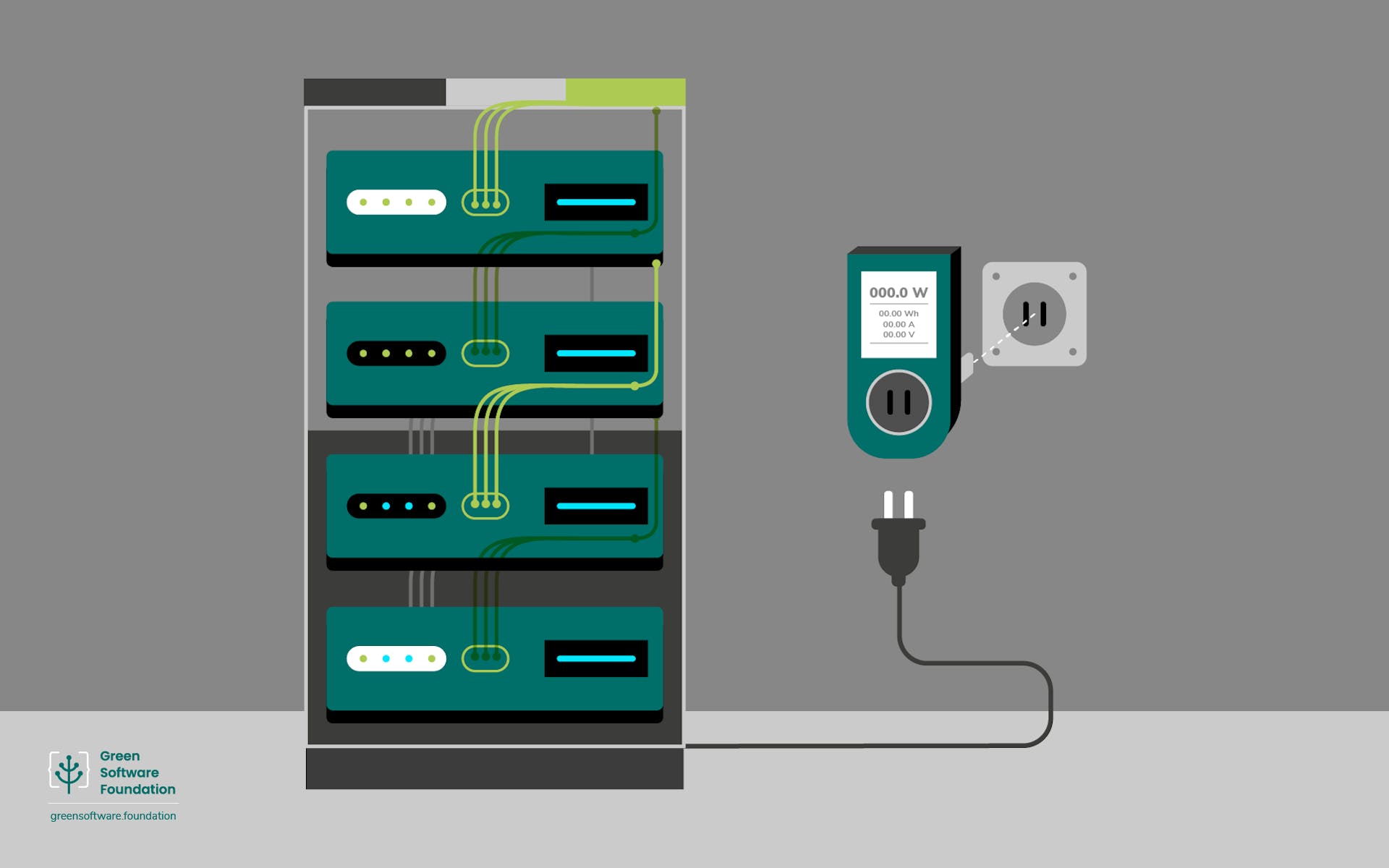
**Measure energy consumption:**

The second principle of [Sustainable Software Engineering](https://principles.green/) is to build energy efficient applications. The very first step in that direction is to measure the energy your application consumes, also known as its energy cost. Once you know your energy cost, you can start to work to decrease it. Understanding your energy cost when the software runs on a server you own is one thing, but how do you measure it when your software is quite literally in the hands of your customers?

### This blog post focuses on measuring energy cost for applications running client-side on a computer, such as web-applications and native applications. It does not cover native applications running on mobile devices such as phones and tablets. For more details on measuring the energy cost of server-side applications, see my post on [the energy consumption of backend services](https://greensoftware.foundation/articles/how-to-measure-the-energy-consumption-of-your-backend-service). This post is operating system agnostic and is primarily meant for front end applications.

**Measure energy consumption directly from the wall socket**

Measuring energy consumption directly from the wall socket is the same approach as for backend applications. Using a watt-hour meter, you can measure the total power consumption of your device, which helps you calculate energy cost. This will always be the best way to measure power consumption because it will measure exactly what your hardware consumes.

****

*Measuring power consumption directly from the wall socket*

Using a watt-hour meter gives you the energy usage of the entire device, but with some planning, you can also use it to determine the approximate energy consumption of an application running on your device. However, this will work best for applications that consume a lot of system resources on the host and/or are long-running.

For example, a program like Visual Studio is a good candidate, but a lightweight website will be trickier. Why is that? Well, there is a lot of noise when running a laptop or a cellphone – the operating system, background processes, the browser itself if you have a web page, among other things. All of these consume power as well. The noise will make it harder to determine the impact of your application, especially if your application consumes little system resources for a short amount of time.

**Considerations on measuring energy consumption**

If your application is a suitable candidate for this approach, here are a few considerations for using a watt-hour meter to measure the energy consumption of an application on your device:

* Any hardware will consume power when idle. Consider this a baseline energy consumption. If you are interested in absolute numbers for your application, then you will need to measure the baseline consumption and then subtract it from the overall consumption measured when running the application.
* Always start measuring from Steady State (SS). There will be a transient event, often an oscillation, observed as a sudden burst of energy when starting up any electrical component, so start measuring when you can observe SS.
* If your device has a battery, it needs to be fully charged when measuring. Otherwise, you will end up measuring the power consumption of charging your battery, which can be interesting, but not what we are going for here.

**Use tools to measure or estimate energy use**

What do we need to measure? On a device, the largest energy consumers are usually the screen, central processing unit (CPU), graphics processing unit (GPU), and memory.

Estimating and measuring the impact of the CPU, GPU, and memory is relatively straight-forward.

Measuring the energy consumption for a device’s screen is more challenging, because there are several types of screens and screen technologies available on the market. Varying screen types and technologies make it difficult to control when measuring energy consumption.