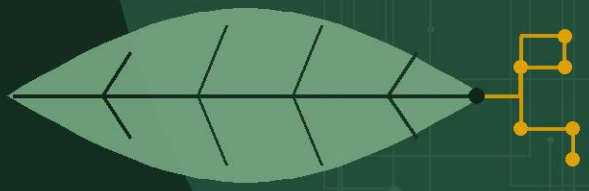

OPEN ELECTIVE · THIRD YEAR AI & DATA SCIENCE

Sustainable Development

for Computing, AI and Data Science

A Coursebook in Four Chapters

Foundations · Lifecycle & Circular Economy
Green Software & AI · Data Centers & Cloud

Dr. Sneha Thombre
MKSSS's Cummins College of Engineering for Women, Pune

Foreword

The global computing infrastructure today accounts for roughly a low single-digit percentage of anthropogenic greenhouse gas emissions, a figure that continues to climb as artificial intelligence workloads scale exponentially. Yet the engineers designing these systems are rarely taught to see energy consumption, embodied carbon, or lifecycle impact as first-order design constraints. This book addresses that gap with remarkable precision.

Dr. Sneha Thombre has produced a work of considerable depth and practical rigour. What distinguishes this coursebook is its refusal to treat sustainability as an appendage to computer science; instead, it weaves GHG Protocol scopes, Life Cycle Assessment under ISO 14040/14044, PUE/CUE/WUE metrics, circular economy principles, and ESG reporting frameworks directly into the fabric of how students learn to write code, train models, and architect infrastructure. The progression from foundational concepts through green software engineering to carbon management and MRV is both pedagogically sound and industry aligned.

I must commend Dr. Thombre for her consistent effort and persistence in bringing this book to fruition. In my experience as an IPCC Expert Reviewer and climate technology practitioner, I have seen how difficult it is to bridge the language of climate science with the vocabulary of engineering education. Dr. Thombre has done so with clarity, intellectual honesty, and an evident commitment to equipping the next generation of engineers with the instinct to measure what they build against the planet's capacity to absorb it.

This coursebook is not merely timely; it is necessary. I commend it to every student, educator, and practitioner who believes that the future of computing must also be the future of accountability.

Vinayak Satpute
Founder & CEO, Switch Climate Tech Private Limited
IPCC Expert Reviewer

Preface

Most textbooks on sustainability were written for environmental science students. This one wasn't. It was written for you: someone who's going to spend the next few decades writing code, training models, and making infrastructure decisions, and who needs to understand exactly where the environmental cost of that work actually shows up.

You'll notice this book doesn't treat sustainability as a separate subject bolted onto computer science. It treats it as a lens you apply to computing itself, the same way you'd apply a lens for correctness or a lens for security. By the end of four chapters, a reasonable goal is that you stop seeing "sustainable computing" and "computing" as two different things at all.

A few notes on how this book is organised. Each chapter opens with a short scenario meant to be concrete rather than abstract, moves through numbered sections that build on each other, and closes with a chapter summary, a key terms table, and review questions you can genuinely use to check whether the material stuck. Along the way you'll find three kinds of boxes: case studies, drawn from real companies and real published numbers wherever possible; short activities you can actually try, alone or with classmates; and quick "did you know" notes, for the facts that are worth remembering even if you forget everything else on the page.

One honest disclosure before you start: some of the statistics in this book, particularly around AI energy use, are recent enough that they may already be slightly out of date by the time you're reading this. That's not a flaw in the book. It's a feature of the field. Sustainable computing is one of the few areas of engineering where the numbers you learn in your third year genuinely might look different by the time you graduate. Get comfortable checking current sources rather than trusting any single textbook, including this one, as a permanent record.

Contents

Chapter 1 — Understanding Sustainable Development and Green Computing

Pick up your phone and check how many apps refreshed themselves in the background while you slept. A weather app pulled fresh data. A messaging app synced your chats. Somewhere, a photo you took last week finished uploading to a server you've never seen and never will. None of this felt like work to you. It was, however, work for a machine somewhere, and that machine needed electricity to do it.

This chapter is about that gap between what technology feels like to use and what it actually costs to run. It's a gap most people, including most engineers, rarely think about, mostly because the whole point of good design is to make the cost invisible. Your job, over the next four units, is to make it visible again — at least in your own head — so that the systems you eventually build account for it.

1.1 A Laptop's Backstory

Ask a room full of engineering students where their laptop came from, and most will say something like "the store" or "Flipkart." That's not wrong, exactly. It's just the very last chapter of a much longer story.

Before your laptop reached a shelf, someone mined lithium and cobalt out of the ground, often in countries thousands of kilometres from where the laptop was assembled. A semiconductor plant turned raw silicon into a processor through a process that uses staggering amounts of ultra-pure water and precisely controlled energy. Parts travelled by ship, by truck, sometimes by air, crossing borders multiple times before they ever met each other inside a single casing. By the time the laptop reached you, it had already accumulated an environmental cost that had nothing to do with how you'd use it.

Then the actual using begins. Every charge draws electricity from a grid that is, in India as of the mid-2020s, still roughly seventy percent powered by coal. Every video call, every model you train for a college project, every file you back up to the cloud triggers computation somewhere else — usually inside a data centre that runs continuously, day and night, cooling itself as hard as it computes.

And eventually, the laptop dies, or more often, gets replaced long before it actually dies. What happens next depends entirely on whether it goes to a formal recycler who safely recovers its materials, or to an informal scrapyard where copper gets recovered by burning cable insulation in the open air. Both outcomes are common in India today. Only one of them is safe.

Figure 1.1 — The Hidden Journey of a Digital Device

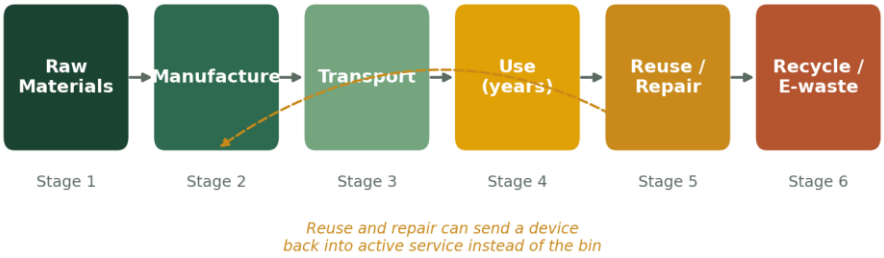

Figure 1.1 — A device's real story starts long before you switch it on, and continues long after you stop using it.

None of this is meant to make you feel guilty about owning a laptop. It's meant to make a simple point: every digital device is a physical object with a physical history, and pretending otherwise is exactly the mistake this whole course exists to correct.

1.2 Why This Generation of Engineers Can't Look Away

Every generation of engineers inherits a different set of constraints. Engineers in the 1960s worried about structural loads and material strength. Engineers in the 1990s worried about scaling networks fast enough to keep up with the internet's growth. Your generation inherits a harder problem: how do you keep building more capable, more widely used technology, at a moment when the planet's tolerance for that technology's side effects is visibly running out?

The signs aren't subtle anymore. Heatwaves arrive earlier each year and last longer. Cities that never used to worry about water now ration it seasonally. None of that is caused by your laptop specifically, but it is caused, in aggregate, by exactly the kind of industrial and digital growth your laptop is part of.

Here's the harder part, and it's worth sitting with rather than rushing past: technology is simultaneously one of the best tools we have for solving this problem and one of the things making it worse. An AI system can optimise a power grid, predict a drought three months in advance, or route delivery trucks so they burn less fuel. The same AI system, if it's a large language model, might have taken months of GPU time and hundreds of tonnes of CO2 to train in the first place. Both of those facts are true about the same technology, sometimes about the exact same model.

> **Did you know?**
> Training a single large AI model can require thousands of GPUs running for weeks. All of that hardware generates heat, which is precisely why data-centre cooling design has become its own specialised engineering discipline, and why we'll spend an entire unit later in this course on it.

So the question this course keeps returning to, in different forms, is not "should we build more technology?" It's "how do we build it so that its benefits keep outpacing its costs?" That's a design question, not a protest slogan, and design questions are exactly what engineers are trained to answer.

1.3 What Sustainable Development Actually Means

The phrase "sustainable development" gets used so often, in so many contexts, that it risks becoming background noise — the kind of phrase you nod along to without really parsing. So let's actually parse it.

The most widely cited definition comes from a 1987 United Nations report, usually called the Brundtland Report after the Norwegian politician who chaired the commission that wrote it. Its definition: development that meets the needs of the present without compromising the ability of future generations to meet their own needs.

Read that twice. It's doing something quietly radical. Before this report, the mainstream assumption in policy circles was that environmental protection and economic growth were opposites — you could have one or the other, and poorer countries in particular were often told to choose growth first, environment later. The Brundtland definition refused that trade-off outright. It said, in effect: if your growth today wrecks the resources your children will need, you haven't actually developed anything. You've just moved the cost forward in time.

That reframing matters enormously for computing specifically. A data centre that delivers cheap, fast cloud services today by running on the dirtiest, cheapest electricity available isn't developing sustainably, even if its quarterly numbers look excellent. It's borrowing against a future that will have to deal with the emissions.

1.4 From Stockholm to Paris: How the Idea Grew Up

Sustainable development didn't arrive fully formed in 1987. It grew out of decades of argument, and knowing the rough timeline helps you understand why today's rules look the way they do.

The first major global environmental conference happened in Stockholm in 1972, and it was the first time governments collectively admitted, on record, that industrial development had environmental costs worth discussing internationally. It took another fifteen years to get from that admission to the Brundtland Report's actual definition. Five years after that, in 1992, the Rio Earth Summit turned the idea into the first real international framework, including an early climate change treaty. The Kyoto Protocol in 1997 was the first attempt to put binding emissions targets on paper, though it excluded developing economies including India, which was a major point of friction for years. The Paris Agreement in 2015 finally brought nearly every country, rich and poor, into a single framework with nationally set targets. That same year, the UN adopted the seventeen Sustainable Development Goals we'll cover in the next section.

Notice the shape of that timeline: roughly one major shift per decade, each one broader and more binding than the last. There's no reason to expect that trend to stop with Paris. If anything, the speed of AI adoption over the last few years suggests the next shift, whatever it turns out to be, might arrive faster than a decade.

1.5 The Three Pillars: People, Planet, Profit

If sustainable development is the destination, the Triple Bottom Line is the compass most organisations actually use to navigate toward it. The term comes from the writer John Elkington, and it says an organisation, a product, or really any decision should be judged against three things at once: its environmental impact, its economic impact, and its social impact.

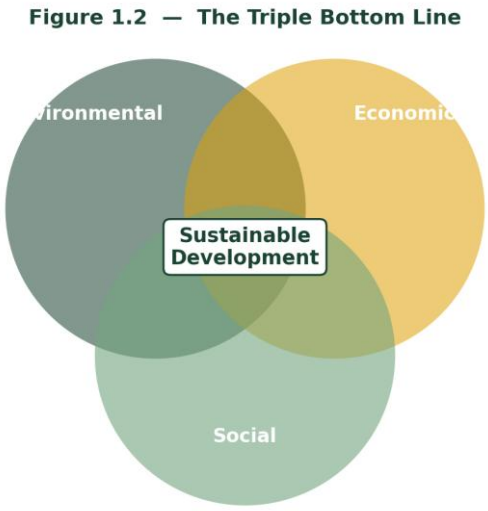

Figure 1.2 — Sustainable development sits at the overlap of all three pillars, not inside any single one.

Here's why the three-way framing matters more than it might first appear. A technology that's environmentally spotless but economically unviable will never get built at scale — it stays a prototype. A technology that's wildly profitable but socially harmful, say a recommendation algorithm that quietly worsens mental health outcomes to maximise engagement, eventually faces regulation, boycotts, or reputational collapse. And a technology that ignores the environmental pillar entirely, however profitable and socially popular, is simply borrowing against a future bill, exactly as Section 1.3 described.

A useful habit to build now, in your third year, rather than after you've already shipped your first product: whenever you evaluate a system, run it through all three pillars deliberately, not just the one that's easiest to measure. Cost is easy to measure. Carbon footprint takes more effort. Social impact takes the most effort of all, and is exactly the pillar most engineering teams skip.

1.6 The Seventeen Goals

In 2015, the UN turned the Triple Bottom Line's fairly abstract three-way balancing act into something governments, companies, and even student projects could actually target: seventeen Sustainable Development Goals, each with a 2030 deadline. Some are obviously environmental — SDG 13 is literally titled Climate Action. Others aren't, at first glance: SDG 4 covers education, SDG 5 covers gender equality, SDG 9 covers industry and infrastructure.

For a course aimed at AI and Data Science students, three goals deserve special attention, because your future work will touch them more directly than most engineering disciplines ever will.

Goal	Title	Where AI and Data Science fit in
SDG 7	Affordable and Clean Energy	Load forecasting, grid optimisation, and renewable-output prediction all rely on data models.
SDG 9	Industry, Innovation, Infrastructure	Green computing itself, from efficient chips to sustainable data centres, sits inside this goal.
SDG 13	Climate Action	Every carbon-accounting and carbon-aware-computing technique in this course serves this goal directly.

It's worth being honest about the tension here rather than glossing over it. The same AI capability that helps optimise a solar grid (advancing SDG 7) also requires GPU clusters that draw heavily from that same grid, often before it's fully renewable. This isn't a contradiction to be embarrassed about. It's the actual, unresolved engineering challenge this entire field is built around solving.

1.7 Green IT: Two Very Different Eras

The phrase "Green IT" has meant genuinely different things at different points in the last twenty years, and it's worth separating those meanings clearly, because textbooks sometimes blur them together.

The first era, which researchers now often call Green IT 1.0, was narrow and largely hardware-focused. It asked one question: how do we make computers themselves — their processors, their power supplies, their cooling fans — consume less electricity? Energy Star ratings, sleep modes, and efficient power supplies all belong to this era. It was useful work, but it treated sustainability as a hardware efficiency problem, full stop.

Green IT 2.0 asks a bigger question. Instead of only making IT itself efficient, it asks how IT can make everything else more efficient too. A smart building management system that cuts a campus's air-conditioning load by twenty percent is Green IT 2.0 in action, even though the system itself runs on servers that consume electricity. A logistics algorithm that shaves ten percent off a delivery fleet's total distance travelled is doing the same thing. The environmental win happens somewhere else in the economy, made possible by computing.

Both eras still matter today, and most real organisations practice a mix of the two without necessarily labelling it that way. But if you remember only one distinction from this section, make it this: Green IT is no longer just about efficient machines. It's about using machines to make everything around them more efficient, while trying not to let the machines themselves become the new problem.

1.8 The Digital Carbon Footprint

Put Sections 1.6 and 1.7 together and you arrive at a concept this entire course keeps circling back to: the digital carbon footprint, meaning the total greenhouse gas emissions caused, directly and indirectly, by digital devices, networks, and data infrastructure across their whole working life.

Every stage of a typical AI or data science workflow carries its own slice of this footprint. Collecting and storing training data draws electricity in a data centre somewhere. Training a model draws far

more, often for days or weeks at a stretch on specialised hardware. Running the trained model afterward, called inference, happens every single time someone actually uses it, and because a popular model gets used millions of times, this repeated cost can quietly outweigh the one-time training cost over the model's lifetime.

> **A number worth remembering**
>
> In August 2025, Google published measured figures for a typical Gemini text response: about 0.24 watt-hours of electricity, 0.03 grams of CO2 equivalent, and roughly a quarter of a millilitre of water, mostly for cooling. Google's own comparison: that's about the same as watching nine seconds of television. Individually, that's a tiny number. Multiply it by a few billion queries a day across the industry, and the picture changes completely — which is exactly why this course exists.

If you take away one myth-busting fact from this entire chapter, make it this: software has a physical footprint. It always has. The only thing that changed recently is that the footprint got large enough, and the tools to measure it got good enough, that pretending otherwise stopped being a reasonable position.

1.9 Counting What You Can't See: The GHG Protocol

Once you accept that computing has a measurable footprint, the obvious next question is: measured how, exactly, and by whom? The answer almost every serious organisation converges on is the Greenhouse Gas Protocol, usually just called the GHG Protocol, and its central move is splitting emissions into three categories called scopes.

Figure 1.3 — The GHG Protocol's Three Scopes

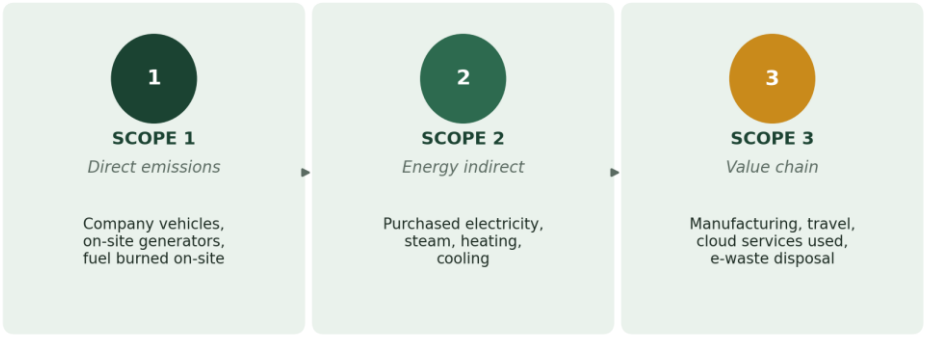

Figure 1.3 — Scope 1 is what you burn. Scope 2 is what you plug in. Scope 3 is everything else you're still responsible for.

Scope 1 covers direct emissions — anything an organisation burns itself, like fuel in a company vehicle or diesel in a backup generator. For most technology companies this is a small slice of their total footprint, because they don't burn much fuel directly.

Scope 2 is usually the big one for anything computing-related: emissions from electricity purchased from the grid. A data centre's servers, cooling systems, and networking equipment all draw grid power, and Scope 2 is where that shows up in a company's carbon accounting.

Scope 3 is the sprawling, hardest-to-measure category: everything else in the value chain. Manufacturing the servers before they were even purchased. Employees commuting to the office. Cloud services the company uses but doesn't own. E-waste when hardware finally gets retired. Most companies find Scope 3 both the largest and the most uncomfortable number to publish, precisely because so much of it is outside their direct control.

> **Case Study 1.1 — The Same Model, Two Very Different Footprints**
>
> In 2020, OpenAI trained GPT-3, a 175-billion-parameter language model. Estimates put its training energy at roughly 1,287 megawatt-hours, and its emissions at somewhere between 500 and 550 tonnes of CO2 equivalent — comparable to driving a petrol car from New York to San Francisco more than four hundred times.
>
> Around the same time, the BigScience research collaboration trained BLOOM, a model with almost exactly the same parameter count: 176 billion. BLOOM was trained on a French supercomputer running mostly on nuclear electricity, which has a much lower carbon intensity than many national grids. Its training run emitted only about 25 to 50 tonnes of CO2 equivalent — ten to twenty times less than GPT-3, for essentially the same size of model.
>
> The lesson isn't that GPT-3's creators did anything wrong. It's that a model's carbon footprint depends as much on where and how it's trained as on how large it is. That single comparison is arguably the most important idea in this entire chapter, because it means engineers actually have meaningful control over a number that might otherwise seem fixed by the model's architecture alone.

1.10 Who Actually Has to Act on This?

It would be convenient if sustainability in computing were purely a technical problem, solvable entirely by better code and smarter chips. It isn't. Four different groups need to move, more or less together, for any of this to work at scale.

- Governments set the rules of the game: emissions targets, mandatory disclosure requirements like India's BRSR framework, and incentives for renewable energy adoption.
- Industry has to actually adopt efficient practices and disclose honest numbers, even when the honest numbers are unflattering.
- Academia, meaning institutions exactly like the one you're studying at, has to research better techniques and train engineers who take the problem seriously from day one rather than treating it as an elective afterthought.
- Citizens and consumers, through what they buy and what they tolerate, ultimately decide whether sustainable products succeed commercially or get undercut by cheaper, dirtier alternatives.

Notice that you personally belong to at least two of these groups already, and you're about to belong to a third the day you take your first job. That's not a guilt trip. It's just an accurate description of where you're standing.

Try This — Audit Your Own Digital Day

For one full day, keep a simple running list of every digital service you use: video calls, streaming, cloud storage syncing, AI assistant queries, online gaming, anything.

At the end of the day, pick your three heaviest activities and estimate, even roughly, which ones likely ran on your own device versus which ones sent work to a remote data centre.

Ask yourself honestly: which single change to your own habits would cut your footprint the most, without meaningfully hurting your productivity? You don't need to actually make the change. Just work out what it would be.

1.11 Chapter Summary

Sustainable development means meeting today's needs without stealing from tomorrow's, a definition that dates back to the 1987 Brundtland Report and has since expanded into the UN's seventeen Sustainable Development Goals. Any technology, including the AI systems you'll build in your own career, gets judged against three pillars at once: environmental, economic, and social impact. Green IT has evolved from a narrow focus on efficient hardware into a much broader project of using technology to make the rest of the economy more efficient, while trying to keep its own footprint in check. That footprint is real and measurable, thanks to frameworks like the GHG Protocol's three scopes, and as the GPT-3 versus BLOOM comparison shows, where and how you compute often matters as much as how much you compute.

1.12 Key Terms

Term	What it means
Sustainable development	Meeting the needs of the present without compromising the ability of future generations to meet their own needs (Brundtland Report, 1987).
Triple Bottom Line	Judging a decision, product, or organisation on three fronts at once: environmental, economic, and social impact.
SDGs	The UN's seventeen Sustainable Development Goals, adopted in 2015, each targeted for 2030.
Green IT 1.0 / 2.0	1.0: narrow focus on efficient hardware. 2.0: using IT as a tool to improve sustainability elsewhere in the economy.
Digital carbon footprint	Total greenhouse gas emissions caused, directly and indirectly, by digital devices, networks, and data infrastructure across their lifetime.
GHG Protocol	The globally used framework for reporting an organisation's greenhouse gas emissions, split into Scope 1, 2, and 3.
CO_2e	Carbon dioxide equivalent — a common unit for expressing the combined warming effect of different greenhouse gases.

1.13 Review Questions

1. State the Brundtland definition of sustainable development in your own words, without quoting it directly.
2. Explain the difference between Green IT 1.0 and Green IT 2.0, using an example of each that wasn't used in this chapter.
3. Classify the following as Scope 1, 2, or 3: (a) diesel burned in a data centre's backup generator, (b) grid electricity purchased to run servers, (c) emissions from manufacturing the servers before purchase.
4. Using the GPT-3 and BLOOM comparison, explain why a model's parameter count alone doesn't determine its carbon footprint.
5. Pick any one of the seventeen SDGs not discussed in Section 1.6, and explain one way AI or data science could help — and one way it could hinder — progress on it.
6. A data centre uses 60,000 kWh of grid electricity in a month. Using an emission factor of 0.75 kg CO2e per kWh (a reasonable approximation for India's grid), estimate its Scope 2 emissions for that month in tonnes of CO2e.

Chapter 2 — Lifecycle Sustainability and the Circular Economy

There's a question that sounds almost too simple to ask in an engineering classroom: what actually happens to a laptop after you stop using it? Most students can answer for the first six months of a device's life in detail, right down to specific benchmark scores. Ask about year five, and the room usually goes quiet.

That gap in knowledge isn't an accident. Manufacturers spend enormous effort making the buying experience visible and the disposal experience invisible. This chapter tries to close that gap, by walking through a device's entire life, not just the part you actually see.

2.1 A Life With More Chapters Than You'd Think

Every computing device, from a smartwatch to a rack of servers, moves through the same rough sequence of stages, even though the details vary wildly by product. Raw materials get pulled out of the ground. Factories turn those materials into components, then assemble the components into a finished product. Distribution networks move that product to wherever it's going to be used. Then comes the part everyone focuses on: actual use, which might last two years or might last ten. And finally, a decision point arrives, whether anyone consciously makes it or not: repair and keep using it, sell it on for someone else to use, or throw it away.

Figure 1.1 — The Hidden Journey of a Digital Device

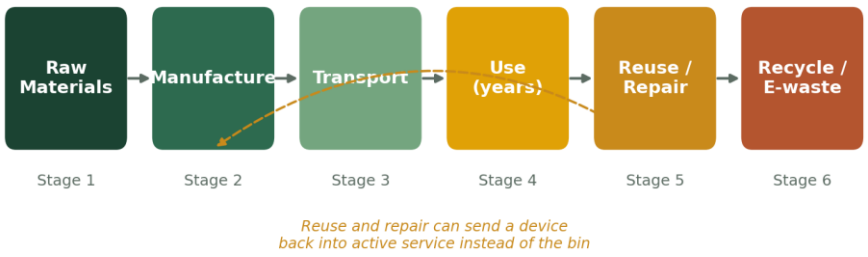

Figure 2.1 — The same lifecycle from Chapter 1, worth revisiting now that we're about to dig into each stage.

Here's the genuinely surprising part, and it's worth sitting with because it overturns a lot of common-sense intuition. For many electronic products, manufacturing causes more total environmental damage than years of actual use. A laptop's chip fabrication process alone, which happens entirely before you ever touch the device, can account for a larger share of its lifetime carbon footprint than two or three years of daily charging. If that's true, then the popular advice "buy the more energy-efficient model" is often less useful than the far less exciting advice "keep the one you already own for longer."

2.2 Embodied Energy: The Bill You Never See

Engineers have a specific term for the environmental cost baked into a product before it ever gets used: embodied energy. It covers everything from the electricity used to smelt the metals, to the water used to rinse silicon wafers in a chip fab, to the diesel burned shipping components across oceans.

Chip fabrication deserves special attention here, because it's genuinely unusual among manufacturing processes. Building a modern processor requires cleanrooms kept thousands of times cleaner than a hospital operating theatre, ultra-pure water used and discarded in enormous volumes, and multiple stages of etching and deposition, each of which consumes electricity and specialty chemicals. None of this shows up on the box your laptop came in. All of it happened before the box existed.

> **Did you know?**
>
> A single semiconductor fabrication plant can use several million litres of ultra-pure water a day, comparable to a small city's daily water demand. That number rarely appears in marketing material for the chips it eventually produces.

Once you accept that embodied energy is real and often large, a design implication follows immediately: extending a device's useful life doesn't just delay a replacement purchase. It defers, and sometimes entirely avoids, all of that upfront cost from ever needing to happen again.

2.3 The Circular Economy: Breaking the Straight Line

For most of industrial history, products have followed a straight line: extract materials, manufacture, use, throw away. Engineers now call this the linear economy, and increasingly, they treat it as a design flaw rather than a neutral fact of life.

The alternative is called the circular economy, and its goal is to bend that straight line back into a loop, keeping materials in use for as long as possible instead of letting them become waste. Three moves make this happen, and the order they're listed in isn't arbitrary.

Figure 2.1 — The Circular Economy Loop

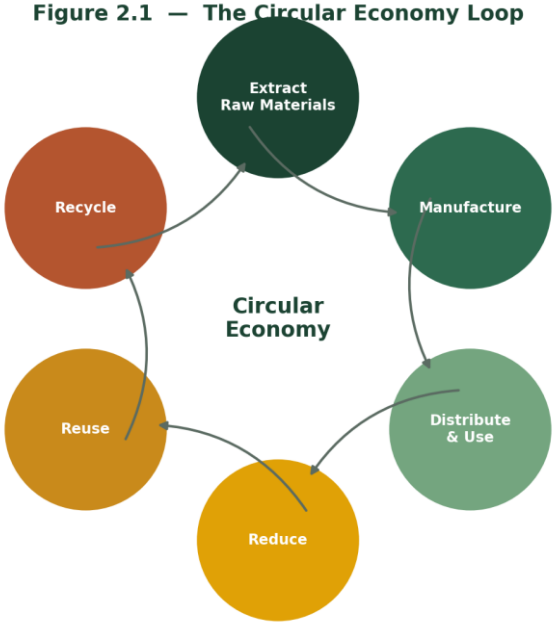

Figure 2.2 — Reduce, reuse, and recycle aren't three equal options. They're a hierarchy.

Reduce comes first because it's the only move that prevents the impact from happening at all. It happens at the design stage: using fewer materials, choosing components that last longer, building devices where a cracked screen doesn't force you to throw away a perfectly good motherboard. Reuse comes second, through refurbishment and resale markets that extend a device's productive life and delay the need for a fresh manufacturing run. Recycle comes last, not because it's unimportant, but because by the time a device reaches this stage, the reduce and reuse opportunities have already been missed.

Two real companies illustrate this hierarchy in action, at a scale well beyond a classroom example. Apple operates a disassembly robot it calls Daisy, purpose-built to take apart returned iPhones and recover materials like cobalt and rare earth elements for reuse in new products, rather than sending them to a shredder where recovery rates are far lower. Dell runs a comparable closed-loop programme that reclaims plastics from old computers and feeds them back into new product casings. Neither of these is a small pilot scheme sitting in a corporate sustainability report for show. Both are stitched directly into each company's actual manufacturing supply chain.

2.4 Urban Mining: Treasure Hiding in a Drawer

Here's a statistic that tends to surprise people the first time they hear it, so it's worth reading slowly. Japan's Ministry of the Environment calculated that one tonne of discarded mobile phones, roughly ten thousand handsets, yields around 280 grams of recoverable gold. A tonne of freshly mined gold ore, by contrast, typically yields far less. The exact comparison usually quoted is that discarded phones are around fifty-six times richer in gold than typical ore.

This concept has a name: urban mining, meaning the recovery of valuable materials from the things we've already made and discarded, rather than digging up new ore. It reframes recycling from a purely environmental obligation into something closer to a genuine economic opportunity. The old phones sitting in your family's drawer right now are, quite literally, a higher-grade deposit than most active gold mines.

If the economics are this favourable, a reasonable question follows: why does so much e-waste still go unrecycled? The honest answer has less to do with the value of the material and more to do with collection logistics, consumer inertia, and competition from informal recyclers who can undercut formal ones on price, even though they handle the material far less safely.

2.5 India's E-Waste Story, In Numbers

India generated roughly one million tonnes of e-waste in 2019-20. By 2023-24, that figure had climbed to somewhere between 1.6 and 1.8 million tonnes a year, making India the third-largest e-waste generator in the world, behind only China and the United States.

Year	E-waste generated	Formally recycled
2019-20	~1.01 million tonnes	~22%

Year	E-waste generated	Formally recycled
2023-24	~1.75 million tonnes	~43%

Read those two rows together and you get a genuinely mixed picture. The good news: formal recycling nearly doubled its share in just a few years. The uncomfortable news: even at 43 percent, well over half of all e-waste generated in India still isn't handled through safe, formal channels, which almost certainly means a great deal of it ends up processed informally, sometimes by burning cables in the open to recover copper, or using acid baths to strip gold from circuit boards.

India's regulatory response has a specific name and a specific date worth remembering: the E-Waste (Management) Rules, 2022, notified in November 2022 and in force since April 2023, replacing the older 2016 rules. Its central innovation is a digital marketplace for Extended Producer Responsibility, where manufacturers meet their legal recycling obligations by buying and trading EPR certificates from registered recyclers, rather than handling collection informally themselves. Whether a certificate-trading system actually increases the amount of material safely recycled, or just creates a cleaner-looking paper trail, is a genuinely open question, and a fair one to debate rather than settle.

2.6 Life Cycle Assessment: Measuring the Whole Picture

Everything discussed so far in this chapter has been mostly qualitative: manufacturing matters, reuse helps, recycling has trade-offs. Eventually, though, engineers need to move from "probably matters" to an actual number they can compare against another number. That's the job of Life Cycle Assessment, almost always shortened to LCA, governed internationally by two standards, ISO 14040 and ISO 14044.

Figure 2.2 — The Four Phases of Life Cycle Assessment (ISO 14040)

LCA is iterative: interpretation often sends you back to refine the scope

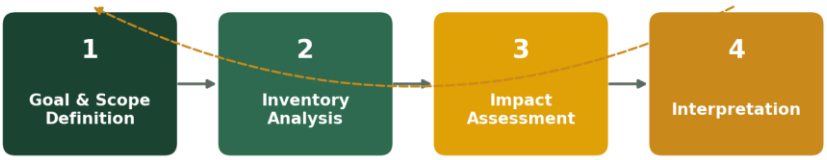

Figure 2.3 — LCA runs through four phases, and often loops back to refine earlier ones as new data comes in.

The first phase, goal and scope definition, forces you to decide exactly what you're comparing and how widely to draw the boundary. Are you counting only manufacturing, or manufacturing through disposal? This phase also introduces a concept called the functional unit — the basis of comparison, phrased as a service delivered rather than a product itself. Comparing "one laptop" against "another

laptop" is close to meaningless. Comparing "computing services for one student, over five years" against the same service delivered a different way is meaningful, because it forces both options to be judged on delivering the exact same thing.

The second phase, inventory analysis, is where the actual data collection happens: every material input, every unit of energy consumed, every emission released, at every stage of the life you defined in phase one. In practice, this is the slowest and most painstaking part of any real LCA study, often taking far longer than the analysis that follows it.

The third phase, impact assessment, translates that raw inventory data into categories anyone can actually interpret: carbon footprint, water consumption, resource depletion, toxicity. The fourth phase, interpretation, is where conclusions get drawn, hotspots get identified, and recommendations get made, whether that means redesigning a product or simply keeping an existing one in service for longer.

Case Study 2.1 — A Laptop's Second Life

Consider an ordinary four-year-old laptop: still functional, a little slow, overdue for a new battery. One option is to recycle it and buy a replacement. Another is to refurbish it, swap the battery, clean it out, reinstall the operating system, and sell it on.

Given how much of a laptop's total footprint sits in manufacturing rather than years of use, as Section 2.1 explained, refurbishing usually saves more emissions than people initially expect, even though buying something new feels like the more exciting choice.

Now imagine the alternative path: this same laptop gets thrown away informally instead. Over half of India's e-waste, as Section 2.5 showed, is still processed this way. Informal handling often means burning cable insulation to recover copper, or dissolving circuit boards in acid to extract gold, both of which release lead, mercury, and other toxic substances into local soil and water.

Run the four LCA phases on this laptop and the pattern holds up: battery and chip manufacturing, not years of daily charging, is almost always the biggest hotspot. That single finding is why refurbishment, not just recycling, deserves a place near the top of any sustainable-computing strategy.

2.7 A Quick Worked Example

Numbers stick better than principles, so here's a small calculation worth working through by hand. Suppose a university has 200 desktop computers, each rated for 5 years of expected service life before replacement. If the university extends that lifespan to 7 years through better maintenance and timely repairs, how many fewer computers get manufactured over a 21-year planning horizon?

At a 5-year cycle, 21 years requires roughly 4.2 replacement cycles, meaning about 840 total computers manufactured and disposed of. At a 7-year cycle, the same 21 years requires exactly 3 cycles, meaning 600 computers. That's 240 fewer computers manufactured, each one carrying its own share of embodied energy from mining, fabrication, and transport that never has to happen. No

change in usage pattern, no change in software, nothing except keeping hardware in service two years longer than the default replacement schedule.

> **Try This — Design a Take-Back Scheme**
>
> Imagine your college wants to launch a scheme where students can trade in old phones or laptops before graduating, instead of taking them home to gather dust or throwing them away.
>
> Sketch, in a few bullet points, what the scheme would need: where devices get collected, who decides whether something gets refurbished versus recycled, and what happens to the data on each device before it changes hands.
>
> Then ask the harder question: what would actually make students participate, rather than just keep their old devices out of habit or emotional attachment?

2.8 Chapter Summary

A device's environmental story begins long before it's switched on and continues long after it stops being useful. Embodied energy, especially from chip fabrication, often outweighs years of operational electricity use, which is why extending a device's life through reuse and repair frequently beats replacing it with something more efficient. The circular economy's reduce-reuse-recycle hierarchy gives engineers a concrete way to act on that insight, and real companies like Apple and Dell already build parts of it into their supply chains. India's e-waste numbers, over 1.75 million tonnes a year and still growing, make this a locally urgent problem, not just an abstract one. Life Cycle Assessment, standardised under ISO 14040 and 14044, gives engineers a rigorous four-phase method for turning "probably matters" into an actual, defensible number.

2.9 Key Terms

Term	What it means
Embodied energy	The total energy used to extract, produce, and transport a product before it's ever switched on or used.
Linear economy	The traditional take-make-dispose model of production and consumption.
Circular economy	An economic model that keeps materials in use for as long as possible, through reduce, reuse, and recycle.
Urban mining	Recovering valuable materials, such as gold and copper, from discarded electronics rather than newly mined ore.
Extended Producer Responsibility (EPR)	A regulatory approach making producers responsible for a product's end-of-life management, including recycling targets.
Life Cycle Assessment (LCA)	A standardised, four-phase method for measuring a product's total environmental impact across its entire life.
Functional unit	The basis of comparison used in an LCA, phrased as a service delivered rather than a product, so that two options can be compared fairly.

2.10 Review Questions

1. Explain, in your own words, why manufacturing can cause more environmental impact than years of actual product use.
2. List the three moves of the circular economy in order of preference, and explain why the order matters.
3. A tonne of discarded mobile phones yields about 280 grams of gold. Look up the approximate current market price of gold per gram, and estimate the value of the gold alone in a tonne of discarded phones.
4. What is a functional unit in LCA, and why is comparing "one laptop" to "another laptop" usually not meaningful without one?
5. India's E-Waste (Management) Rules, 2022 introduced a certificate-trading mechanism for Extended Producer Responsibility. Argue for or against this approach, compared to requiring producers to handle collection directly themselves.
6. A college replaces 150 laptops every 4 years. If it extends the replacement cycle to 6 years, how many fewer laptops get manufactured over a 24-year planning horizon?

Chapter 3 — Green Software Engineering and Sustainable Artificial Intelligence

Ask most programmers what makes code good, and you'll hear about correctness, readability, and maybe how fast it runs. Almost nobody mentions how much electricity it draws while running, even though that number is real, measurable, and directly tied to how the code was written. This chapter asks you to add one more question to that list, right alongside correctness and speed: how much energy does this actually cost?

If you're heading into an AI or Data Science career specifically, this chapter probably matters more to your daily work than anything else in this book. The models you'll train, the pipelines you'll build, and the cloud resources you'll spend without a second thought are exactly what this chapter is about.

3.1 Slow Code Isn't Just Slow

Here's an idea that rarely gets stated directly in a data structures course, even though it follows immediately from what's already taught there: an algorithm with worse time complexity doesn't just take longer to run, it also draws more electricity to do the exact same job. Choosing an O(n log n) sorting algorithm over an O(n^2) one isn't only a performance decision anymore. It's an energy decision too, whether the programmer thinks of it that way or not.

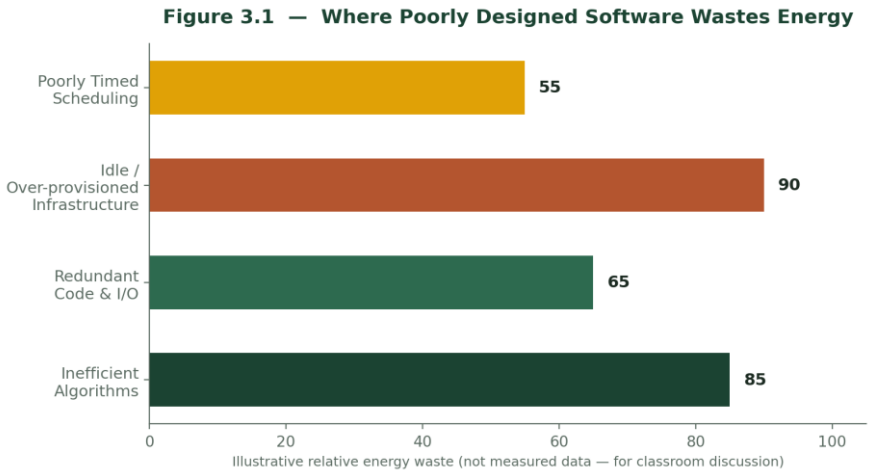

Figure 3.1 — Four places where ordinary software habits quietly waste energy.

The same logic holds below the algorithm level. Redundant computation, unnecessary disk reads and writes, and memory allocated but never really needed all add up to real energy cost per request, per training step, per user interaction. None of this requires exotic new tools to fix. It mostly requires programmers who've been taught to notice.

At the system level, a different kind of waste shows up: infrastructure that's simply oversized for the job it's doing. A server provisioned for peak load but running at ten or fifteen percent utilisation most of the day still draws power close to its maximum, regardless of how little useful work it's actually performing. Caching, batching similar requests together, and right-sizing infrastructure to match real demand all address this same underlying problem from different angles.

And then there's a lever that has nothing to do with code quality at all: timing. Carbon-aware scheduling means running a job, especially a flexible one like overnight batch processing or non-urgent model training, at whatever moment the electricity grid's mix happens to be cleanest, drawing more from solar or wind and less from coal. The job itself doesn't change one bit. Its emissions do, purely because of when it ran.

Case Study 3.1 — Teaching a Data Centre to Cool Itself

In 2016, Google gave DeepMind's engineers access to years of sensor data from its own data centres: temperatures, pump speeds, power draw, and more. DeepMind trained a machine learning system to predict and continuously optimise the cooling systems in real time, adjusting settings far more precisely and far more often than any human operator reasonably could.

The result, independently reported and still cited today: a 40 percent reduction in the energy used specifically for cooling, which translated into a 15 percent drop in that facility's overall Power Usage Effectiveness, the lowest the site had ever recorded.

It's worth noticing what's actually happening in this example: an efficient algorithm (the neural network) improved a piece of sustainable architecture (the cooling system), and the two together produced a result that neither could have achieved alone. We'll return to Power Usage Effectiveness properly in Chapter 4, but it's worth remembering this story once that metric formally arrives.

3.2 Green AI: Treating Efficiency as a Real Metric

A widely cited paper by Schwartz and colleagues introduced a term that's now common across the AI research community: Green AI. Its central argument is straightforward, though it cuts against how most machine learning research has historically been judged. Efficiency, meaning the compute, data, and energy a model consumes, should be measured and optimised for right alongside accuracy, not buried in a footnote or ignored entirely.

Three techniques come up again and again in this space, and none of them are exotic research curiosities anymore. Most mainstream deep learning frameworks support at least one out of the box.

Technique	What it does	Typical trade-off
Model compression	Shrinks a trained model's size while preserving most of its accuracy.	Small accuracy loss for a large reduction in storage and inference cost.
Pruning	Removes weights or neurons that contribute little to the model's output.	Requires careful tuning to avoid removing something that mattered.
Quantization	Uses lower-precision number formats, such as 8-bit instead of 32-bit, for a model's weights.	Cuts memory and compute cost substantially, usually for marginal accuracy loss.

It's worth correcting a common assumption before moving on, because it changes how you should think about all three techniques above. Most public conversation about AI's energy use focuses on training, because that's where the headline numbers come from. But Google has estimated that around sixty percent of AI energy use actually goes toward inference, meaning the model being used repeatedly after it's already trained, not the training run itself. A model trained once and then queried millions of times a day can easily spend more total energy on inference over its life than it ever spent on training.

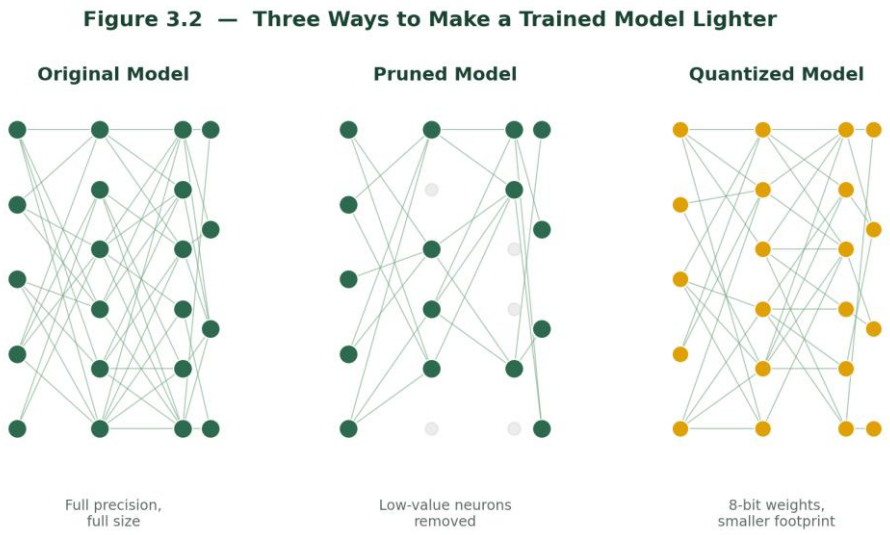

Figure 3.2 — Pruning removes the connections that barely matter. Quantization keeps the connections but stores them more cheaply.

3.3 A Live Comparison Worth Knowing

In August 2025, Google published measured figures for its Gemini model: a typical text response uses about 0.24 watt-hours of electricity. Around the same time, OpenAI's Sam Altman gave a comparable figure for ChatGPT, about 0.34 watt-hours for an average query. Those two numbers sit reassuringly close together, and both are genuinely small.

But independent researchers measuring OpenAI's newer, larger reasoning-style models found individual responses running to 18 to 40 watt-hours, roughly fifty to over a hundred times higher than the average-query figure quoted above. Why would a model that reasons step by step before answering cost so much more per response than one that answers immediately? Because it's doing more computation per query, sometimes generating and discarding several draft answers internally before settling on a final one. That gap is a direct, practical illustration of the training-versus-inference point from the previous section: a larger, slower-thinking model multiplies its inference cost every single time someone uses it, which is exactly why the Green AI techniques from Section 3.2 matter more as models get more capable, not less.

Here's a small calculation worth working through yourself. If a research group runs 50,000 inference queries a month against a deployed model at the Gemini-level figure of 0.24 watt-hours each, total energy comes to 12 kilowatt-hours a month. At India's grid emission factor of roughly 0.75 kilograms of CO2 equivalent per kilowatt-hour, that's about 9 kilograms of CO2e a month for inference alone.

Now suppose quantization cuts per-query energy by 40 percent, a figure genuinely achievable with many 8-bit conversions. The new monthly footprint works out to roughly 5.4 kilograms. The arithmetic isn't the point. The point is that one engineering decision, made once at deployment time, keeps paying off every single month for as long as the model stays in production.

3.4 Carbon Accounting for Software Teams

The Scope 1, 2, 3 framework from Chapter 1 applies just as directly to a software or AI team's day-to-day operations as it does to an entire company. Scope 2, the electricity purchased to run training jobs and serve deployed models, is usually the dominant category by a wide margin. Increasingly, organisations use ESG analytics platforms and sustainability dashboards to track these numbers continuously, rather than calculating them once a year for a report nobody reads until the deadline.

Generative AI complicates this picture in a way worth flagging honestly. A single trained model might be queried by millions of people doing wildly different things with it: drafting an email, generating an image, debugging code. Attributing a fair share of emissions to any one specific use case is genuinely difficult, and how honestly the industry currently reports these numbers is a live, unresolved debate rather than a settled question with a tidy answer.

> **Try This — Code Golf, But for Energy**
>
> Take a piece of deliberately inefficient code, for example a nested loop searching through a list where a dictionary or set lookup would do the same job far faster.
>
> Rewrite it to be efficient, and if you have access to a tool like CodeCarbon, or even just a stopwatch and a rough power-draw estimate for your machine, measure the before-and-after energy difference, not just the runtime difference.
>
> Compare your result with a classmate's. The runtime improvement is usually the more dramatic-looking number. The energy improvement is the one this chapter actually cares about.

3.5 Chapter Summary

Software design choices, from algorithm selection to infrastructure sizing to job scheduling, are also energy choices, whether or not the programmer making them thinks of them that way. Green AI treats efficiency, meaning compute, data, and energy, as seriously as accuracy, rather than as an afterthought. Model compression, pruning, and quantization are practical, deployable techniques, not research curiosities reserved for papers. Inference, not just training, is a major and frequently underestimated source of a deployed model's total energy use over its lifetime. And carbon-aware scheduling shows that emissions can sometimes be cut purely through timing, without touching a single line of the model or the code around it.

3.6 Key Terms

Term	What it means
Green software engineering	Writing and architecting software with energy efficiency treated as a genuine design goal, not an afterthought.
Carbon-aware scheduling	Timing computing jobs to run when the electricity grid's mix is cleanest, lowering emissions without changing the code.
Green AI	An approach that reports and optimises for compute, data, and energy efficiency alongside model accuracy.
Model compression	Reducing a trained model's size while preserving most of its original accuracy.
Pruning	Removing weights or neurons that contribute little to a model's output, making it smaller and faster.
Quantization	Using lower-precision number formats for a model's weights, cutting memory and compute cost.
Training vs. inference energy	Training is the one-time cost of building a model; inference is the repeated cost of using it afterward.

3.7 Review Questions

1. Explain, with an example of your own, why choosing an algorithm with better time complexity is also an environmental decision.
2. Describe the DeepMind cooling case study in your own words, and identify which specific green-computing lever from Section 3.1 it demonstrates.
3. Define pruning and quantization, and explain how each one reduces a model's energy footprint differently.
4. Why might a widely deployed model spend more total energy on inference than it ever spent on training?
5. A team's model handles 200,000 queries a month at 0.3 watt-hours each. Calculate its monthly energy use in kilowatt-hours, then estimate its Scope 2 emissions using an emission factor of 0.75 kg CO2e per kWh.
6. Argue for or against this statement: "Leaderboards that rank AI models should be legally required to publish a training-emissions figure alongside accuracy scores."

Chapter 4 — Sustainable Data Centers, Cloud Infrastructure and Carbon Management

Every chapter so far has zoomed in: from the whole idea of sustainable development, down to a single laptop's lifecycle, down further to a single line of code. This final chapter zooms all the way back out, to the buildings full of servers that make almost everything discussed earlier in this book physically possible, and then further still, to the policy and financial systems that sit above the engineering.

It's also, frankly, the chapter most likely to connect directly to your future career. Job titles like carbon analyst, ESG analyst, and sustainability consultant live almost entirely inside the material covered here.

4.1 What's Actually Inside a Data Centre

Picture a data centre not as a single box but as a chain of linked systems, each one dependent on the others. IT load comes first: the servers, storage arrays, and GPU clusters actually doing useful computation. Right beside it sits cooling, which is often the single largest non-IT energy draw in the entire facility, sometimes rivalling the IT load itself, because dense racks of processors generate serious heat that has to go somewhere.

Figure 4.1 — Where a Data Centre's Energy Actually Goes

Figure 4.1 — A data centre's energy doesn't just power computation. It powers everything computation needs to survive.

Power distribution comes next in the chain: UPS systems, transformers, and backup diesel generators, each one adding its own small losses before electricity ever reaches a server. On top of all this sits renewable integration, meaning on-site solar, long-term power purchase agreements, and the broader decarbonisation of whatever grid the facility happens to be connected to. And finally, monitoring and audit, the ongoing process of checking whether everything described so far is actually working as intended, using a small set of standard metrics.

4.2 Three Numbers Every Data Centre Reports

Those metrics are Power Usage Effectiveness, Carbon Usage Effectiveness, and Water Usage Effectiveness, almost always abbreviated PUE, CUE, and WUE. All three follow exactly the same formula: total resource consumed, divided by the resource consumed by IT equipment alone. Lower is always better.

PUE's theoretical best case is 1.0, meaning zero overhead beyond the compute itself, a number no real facility actually hits but every facility is measured against. In practice, the industry average sits around 1.5 to 1.6. Google's own recent sustainability disclosures put its fleet-wide PUE at around 1.09, achieved mainly through better cooling design, predictive control systems like the DeepMind project from Chapter 3, and aggressive server utilisation.

Figure 4.2 — What PUE Actually Shows You

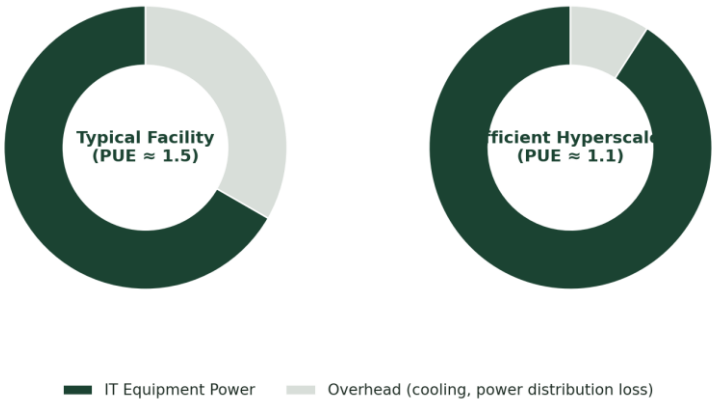

Figure 4.2 — At a PUE of 1.5, a third of a facility's power never reaches the actual computers. At 1.1, almost all of it does.

A quick worked problem: a facility draws 2,200 kilowatts of total power, of which 2,000 kilowatts reaches IT equipment. Work out the PUE before reading on. It comes out to 1.1, right in hyperscaler territory, and tells you the facility's cooling and power distribution are running about as efficiently as anything currently operating at scale.

CUE applies the identical logic to carbon: total CO2 emissions from the facility, divided by IT equipment energy. WUE does the same for water, a metric that matters more every year as data centres cluster in regions already under water stress, a real concern in parts of India.

4.3 Making the Cloud Itself Sustainable

A single data centre being efficient is only half the story. The way workloads get distributed across many data centres, potentially across many countries, matters just as much, and this is where cloud computing's specific tricks come in.

Virtualisation lets many logical servers share one physical machine, raising utilisation and eliminating the idle capacity that comes from a strict one-application-per-server setup. Containers push this further: where a virtual machine bundles an entire operating system alongside each application, a container shares the host operating system and carries only what the application itself actually needs, making it smaller, faster to start, and more energy-efficient to run at scale. Tools like Docker and Kubernetes have made containerisation close to the default choice for new cloud deployments over the last decade.

Auto-scaling addresses a different problem: demand that isn't constant. An online retailer might normally serve 10,000 users, then spike to 500,000 during a festival sale. Without auto-scaling, a company would need to permanently provision for the peak, leaving most of that capacity idle, and still drawing power, for the other 360 days of the year. With auto-scaling, the cloud adds servers automatically as demand rises and releases them just as automatically once it falls.

Workload consolidation attacks a related, quieter problem: three servers each running at fifteen to twenty percent utilisation can often be combined onto one server running at a healthy sixty to seventy percent, with the other two switched off or placed in low-power standby. And carbon-aware scheduling, introduced in Chapter 3 at the level of a single training job, applies at cloud scale too: identical workloads can run in whichever available data centre, India, Norway, Canada, currently has the cleanest grid mix, shifting emissions down without changing a single line of the application.

Case Study 4.1 — Two Companies, Same Workload, Very Different Bills

Company A runs 500 physical servers at an average utilisation of just 15 percent, a common pattern for organisations that provisioned hardware for occasional peak demand and never revisited the decision. Electricity consumption stays high around the clock, regardless of how little useful work is actually happening most of the time.

Company B migrates the same workload to the cloud, adopting virtualisation, containers, auto-scaling, and carbon-aware scheduling together. Average utilisation climbs to around 70 percent. Electricity consumption drops sharply, not because any individual server became more efficient, but because far fewer servers are needed to do the same total amount of work.

A simple numerical version of this story: one physical server consuming 600 watts, multiplied across 100 servers, totals 60 kilowatts. After virtualisation consolidates the same workload onto 40 active servers, total consumption drops to 24 kilowatts, a saving of 36 kilowatts. And because fewer active servers also generate less heat, cooling energy drops too. The saving effectively happens twice: once in computing energy, and again in cooling energy needed to remove the heat that computing would otherwise have produced.

4.4 Sustainable AI Infrastructure Specifically

Everything in Section 4.3 applies to general-purpose cloud computing. GPU clusters built specifically for AI training and inference need a few additional considerations layered on top. Efficient training pipelines avoid wasted GPU-hours from failed runs or poorly tuned hyperparameter searches. Smarter job scheduling packs training jobs onto clusters more tightly, reducing the idle GPU time that's common when jobs are scheduled without regard for how well they actually fit together. And continuous hardware utilisation monitoring catches the same fifteen-percent-utilisation problem described in Section 4.3, except now applied to some of the most expensive and most power-hungry hardware in the entire data centre.

Two real facilities illustrate how location and climate themselves become part of the efficiency strategy. Google operates a data centre in Hamina, Finland, that draws cold seawater from the Bay of

Finland for cooling, a design choice made possible entirely by geography. Microsoft, more experimentally, ran Project Natick, sinking a sealed data centre capsule onto the seabed off Scotland's Orkney Islands for several years, testing whether ambient ocean cold and a sealed, oxygen-free environment could cut both cooling energy and hardware failure rates simultaneously. Neither approach is available to a facility being built in, say, an inland city with a hot, dry climate, which raises a fair question: what does the equivalent "use what's locally available" strategy look like somewhere landlocked and hot rather than coastal and cold?

4.5 From Engineering to Policy: Who Checks the Numbers?

Everything described so far in this chapter is an engineering story. This section is a policy story, and it matters just as much, because engineering improvements only count for something once they're measured, reported, and trusted.

A green audit is the starting point: a systematic assessment of a facility's energy, water, and emissions performance against established benchmarks. You can't improve what you haven't measured, and a green audit is simply the act of measuring honestly before making any claims about improvement.

ESG reporting, short for Environmental, Social, and Governance reporting, is how organisations disclose their sustainability performance to investors, regulators, and the public. In India, this takes a specific mandatory form called BRSR, Business Responsibility and Sustainability Reporting, enforced by the securities regulator SEBI for the country's top listed companies. Data centre and IT metrics, PUE among them, feed directly into these disclosures, meaning a number an engineering team calculates in a server room can end up in a document read by institutional investors.

Carbon credits and carbon markets add a financial incentive on top of regulatory obligation: an organisation that cuts its emissions below an agreed baseline can earn tradeable credits, which other organisations can then purchase to offset their own emissions. Whether this is a genuine driver of real efficiency investment, or mostly a way to purchase reputational cover without changing much operationally, is a serious and ongoing debate, not a settled question with an obvious right answer.

What keeps carbon markets honest, at least in principle, is MRV: Measurement, Reporting, and Verification. It's the auditing layer that checks whether a claimed emissions reduction actually happened, rather than simply being asserted. Without credible MRV, a carbon credit is only as trustworthy as the word of whoever issued it.

> **Try This — Design Your Own Data Centre**
>
> You've been asked to recommend a location for a new data centre in India. Your shortlist includes a coastal city, a hot inland city, and a town in the Himalayan foothills.
>
> Using PUE, WUE, and grid carbon intensity as your main pieces of evidence, argue for one location over the other two. Draw on the Hamina and Project Natick examples from Section 4.4 for

inspiration, but don't just copy their approach; explain what would actually work for the location you chose.

Then consider the opposite question: what would make your chosen location a poor choice five or ten years from now? Climate change, water stress, and grid changes are all fair things to weigh in.

4.6 Chapter Summary

A data centre's total energy use extends well beyond its servers, encompassing cooling, power distribution, and increasingly, renewable integration, all tracked through three standard metrics: PUE, CUE, and WUE. Cloud computing's core techniques, virtualisation, containerisation, auto-scaling, workload consolidation, and carbon-aware scheduling, turn chronically underutilised infrastructure into something closer to fully utilised, cutting both computing and cooling energy simultaneously. Sustainable AI infrastructure applies these same principles specifically to GPU clusters, where the stakes and the costs are both higher. And above all of this sits a policy layer, green audits, ESG and BRSR reporting, carbon markets, and MRV, that determines whether engineering improvements translate into numbers anyone outside the engineering team can actually trust.

4.7 Key Terms

Term	What it means
PUE (Power Usage Effectiveness)	Total facility power divided by IT equipment power. Theoretical ideal is 1.0; industry average is roughly 1.5 to 1.6.
CUE (Carbon Usage Effectiveness)	Total CO2 emissions from a facility divided by its IT equipment energy use.
WUE (Water Usage Effectiveness)	Total water used, mostly for cooling, divided by IT equipment energy.
Virtualisation	Running several logical servers on one physical machine, raising utilisation and cutting idle capacity.
Container	A lightweight application package that shares the host operating system, unlike a full virtual machine.
Auto-scaling	Automatically increasing or decreasing computing resources to match real-time demand.
Green audit	A systematic assessment of a facility's energy, water, and emissions performance against benchmarks.
BRSR	Business Responsibility and Sustainability Reporting, India's mandatory ESG disclosure framework enforced by SEBI.
MRV	Measurement, Reporting, and Verification — the process that checks whether claimed emission reductions are genuine.

4.8 Review Questions

1. A data centre draws 3,000 kW of total power, of which 2,400 kW reaches IT equipment. Calculate its PUE and compare it to the industry average.
2. Explain the difference between a virtual machine and a container, and why containers are generally more energy-efficient.
3. Describe auto-scaling using a real-world example other than the online retailer example used in this chapter.
4. Name the regulator responsible for enforcing BRSR in India, and explain what kind of organisations it applies to.
5. Why does a facility's geographic location affect its Scope 2 emissions even when two facilities have identical equipment and identical PUE?
6. Explain what MRV is for, and describe what could go wrong with a carbon credit market that lacked it.

Glossary

Every key term from all four chapters, gathered here and sorted alphabetically. Useful for last-minute revision, or for double-checking a definition mid-chapter without flipping backward.

Term	What it means
Auto-scaling	Automatically increasing or decreasing computing resources to match real-time demand.
BRSR	Business Responsibility and Sustainability Reporting, India's mandatory ESG disclosure framework enforced by SEBI.
Carbon-aware scheduling	Timing computing jobs to run when the electricity grid's mix is cleanest.
Circular economy	An economic model that keeps materials in use for as long as possible, through reduce, reuse, and recycle.
CO_2e	Carbon dioxide equivalent, a common unit for expressing the combined warming effect of different greenhouse gases.
Container	A lightweight application package that shares the host operating system, unlike a full virtual machine.
CUE	Carbon Usage Effectiveness — a facility's total CO2 emissions divided by its IT equipment energy use.
Digital carbon footprint	Total greenhouse gas emissions caused by digital devices, networks, and data infrastructure across their life.
Embodied energy	The total energy used to extract, produce, and transport a product before it's ever used.
Extended Producer Responsibility (EPR)	A regulatory approach making producers responsible for their product's end-of-life management.

Term	What it means
Functional unit	The basis of comparison used in an LCA, phrased as a service delivered rather than a product.
GHG Protocol	The global framework for reporting greenhouse gas emissions, split into Scope 1, 2, and 3.
Green AI	An approach that reports and optimises for compute, data, and energy efficiency alongside model accuracy.
Green audit	A systematic assessment of a facility's energy, water, and emissions performance against benchmarks.
Green IT 1.0 / 2.0	1.0: narrow focus on efficient hardware. 2.0: IT used as a sustainability tool elsewhere in the economy.
Life Cycle Assessment (LCA)	A standardised, four-phase method for measuring a product's total environmental impact.
Linear economy	The traditional take-make-dispose model of production and consumption.
Model compression	Reducing a trained model's size while preserving most of its original accuracy.
MRV	Measurement, Reporting, and Verification — checks whether claimed emission reductions are genuine.
Pruning	Removing weights or neurons that contribute little to a model's output.
PUE	Power Usage Effectiveness — total facility power divided by IT equipment power. Ideal is 1.0.
Quantization	Using lower-precision number formats for a model's weights, cutting memory and compute cost.
Scope 1, 2, 3 emissions	The GHG Protocol's three categories: direct, purchased-energy, and value-chain emissions.
SDGs	The UN's seventeen Sustainable Development Goals, adopted in 2015 with a 2030 target date.
Sustainable development	Meeting today's needs without compromising future generations' ability to meet their own (Brundtland, 1987).
Triple Bottom Line	Judging an activity on three fronts at once: environmental, economic, and social impact.
Urban mining	Recovering valuable materials from discarded electronics rather than newly mined ore.
Virtualisation	Running several logical servers on a single physical machine, raising utilisation and cutting idle capacity.
WUE	Water Usage Effectiveness — total water used, mostly for cooling, divided by IT equipment energy.

A Closing Note

By the end of this book, you can calculate a PUE, name the three GHG scopes without looking them up, and explain why BLOOM emitted a fraction of what GPT-3 did despite being almost exactly the same size. Those are specific, useful, examinable skills, and you should hold onto them.

But the harder, less testable shift this book was actually aiming for is this: noticing, by instinct now rather than by deliberate effort, that every technical decision you make carries a cost that used to be invisible to you and now isn't. An algorithm choice. A training run. A choice of which cloud region to deploy to. None of these felt like environmental decisions before you started this course. All of them are.

That instinct doesn't expire after your final exam. If anything, it's the part of this book most likely to still matter fifteen years into your career, long after the exact PUE formula has faded from memory and Google's next Gemini efficiency numbers have made this edition's statistics look quaint.

Good luck with the rest of the semester, and with whatever you build after it.

References

Sources drawn upon across all four chapters, listed alphabetically by author or organisation.

[1] Altman, S. (2025). Public statement on average ChatGPT query energy use (approximately 0.34 Wh per query).

[2] Apple Inc. (2024). Environmental Progress Report — disassembly robotics (Daisy) and materials recovery.

[3] Dell Technologies. (2024). Sustainability Report — closed-loop plastics recycling programme.

[4] Elkington, J. (1997). Cannibals with Forks: The Triple Bottom Line of 21st Century Business. Capstone Publishing.

[5] Evans, R., & Gao, J. (2016). DeepMind AI Reduces Google Data Centre Cooling Bill by 40%. DeepMind Blog, Google.

[6] Government of India, Ministry of Environment, Forest and Climate Change. (2022). E-Waste (Management) Rules, 2022.

[7] Google. (2025). Environmental Report and Gemini App efficiency disclosures — per-prompt energy, carbon, and water use.

[8] Green Software Foundation. (2024). Green Software Practitioner curriculum and patterns catalogue.

[9] International Energy Agency. (2025). Energy and AI. IEA, Paris.

[10] International Organization for Standardization. (2006). ISO 14040:2006 — Environmental management — Life cycle assessment — Principles and framework.

[11] International Organization for Standardization. (2006). ISO 14044:2006 — Environmental management — Life cycle assessment — Requirements and guidelines.

[12] Luccioni, A. S., Viguier, S., & Ligozat, A.-L. (2022). Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. arXiv preprint.

[13] Ministry of the Environment, Japan. (2017). Study on Metal Recovery from Small Electronic Devices (Urban Mining estimates).

[14] Murugesan, S., & Gangadharan, G. R. (2012). Harnessing Green IT: Principles and Practices. Wiley-IEEE Press.

[15] Patterson, D., Gonzalez, J., Le, Q., et al. (2021). Carbon Emissions and Large Neural Network Training. arXiv preprint.

[16] Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. Communications of the ACM, 63(12), 54–63.

[17] Securities and Exchange Board of India. (2021). Business Responsibility and Sustainability Reporting (BRSR) framework.

[18] United Nations. (1987). Our Common Future (The Brundtland Report). World Commission on Environment and Development.

[19] United Nations. (2015). Transforming Our World: The 2030 Agenda for Sustainable Development.

[20] United Nations Institute for Training and Research (UNITAR). (2024). Global E-waste Monitor 2024.

[21] World Resources Institute & World Business Council for Sustainable Development. (2004). The Greenhouse Gas Protocol: A Corporate Accounting and Reporting Standard.