

Elba

A quiet handbook for keeping what's yours

YOUR FILES. NOT THEIR BUSINESS.

Able was I ere I saw Elba.

01

A note before you begin

You picked this up because something about the arrangement had started to bother you.

Not a break-in. Nothing so dramatic. Just a slow, creeping sense that the machine on your desk — the one you paid for, the one that holds your half-written letters and your photographs and the document you don't like to think about — was never entirely yours. That somewhere in the quiet there was always a second reader. Patient. Tireless. Taking notes.

This is a manual for a small program that gives you one room the second reader can't enter.

It is not a long program. It is one file. But the idea underneath it is old, and it takes a little telling, so this handbook is longer than the program deserves and exactly as long as the idea needs. You do not have to read all of it. Keep it by the machine. Come back to the parts you need.

We wrote it the way you'd explain something to a friend at the kitchen table, late, after the others have gone to bed. Where the machinery matters, we've set it aside in its own boxes, so that those who want the wiring can have it and those who don't can walk straight past. Neither of you is the wrong kind of person.

Take your time. There's no rush. That's rather the point.

Contents

A note before you begin	2
Part One · The Island	6
The mainland	7
What Elba is	9
Why a fence and not a vault	11
The grey island	13
Part Two · Arriving	15
What you need	16
Claiming your ground	18
The one key	20
Raising the fence	22
Part Three · Living Inside	23
The first thing you keep	24
Bringing things in	26
Rooms	28
Opening and closing	30
Locking up	31
A day in the life	32
What people keep here	34
Part Four · The Machinery	36

What happens when you type	37
The one key, again	39
What the folder really holds	42
Nothing leaves	44
The sealed map	46
Hiding the shape	47
Part Five · Keeping It	49
Copies	50
Remembering	52
The long horizon	54
Passing it on	56
Part Six · Where the Fence Ends	58
The door with no second key	59
What a fence can't do	61
The room you're standing in	63
The weather outside	65
The edges of the map	67
Part Seven · Companionship	68
Small rituals	69
On not needing us	71
A soft landing	73
Appendix A · Quick Reference	74
At the door	75

Inside the fence	76
Things that happen on their own	77
Appendix B · A Short Glossary	78
Words as Elba uses them	79
Appendix C · For the Technically Minded	81
The whole stack, briefly	82
Appendix D · Questions People Ask	84
Plainly answered	85

PART ONE

The Island

The mainland

Somewhere right now, a machine is reading a sentence you thought was private, and finding it very slightly profitable.

Not a person. Don't picture a person in a room. Picture instead something more like weather — a vast, patient system that settles over everything you do, learns the shape of you the way water learns the shape of a stone, and asks for nothing but that you keep going, keep typing, keep leaving your small trail of self across the glass.

You have grown used to it. That's the strange part. Not that it happens, but that it stopped feeling like anything. There was a time, not long ago, when the word *private* meant something plain: a drawer that shut, a letter in a sealed envelope, a diary with a little brass lock a child could break but an adult would not. The lock wasn't the point. The *closing* was the point. It said: this much of me is mine.

Then the drawers went into the machine, and the machine went onto the network, and the network went to work. Now your files are grazed the way a field is grazed — gently, continuously, by things that never quite finish and never quite leave. Your operating system indexes them so it can find them faster, which is kind, and also means it has read them. Your backup service copies them to a building you have never seen, which is prudent, and also means someone else holds them. A model somewhere trains on whatever it can reach, which is the age we live in, and also means your turns of phrase may

end up, faintly, in the mouth of a machine that will one day sell them back to you.

None of this was done to you with malice. That is the thing that's hardest to hold. It was done with *convenience*, which is a softer word and a harder problem. Malice you can refuse. Convenience you have to choose against, over and over, while everything around you is built to make choosing tiresome.

So here is the small, unfashionable claim this whole handbook rests on. It is not that your thoughts are dangerous. It is not that you have something to hide. It is only this: that some things are yours, plainly and by default, the way your own breath is yours, and that being able to close a door is not the same as having something to be ashamed of behind it.

You know this already. You knew it when you were nine and you hid the notebook. What you may have forgotten is that you're allowed to still want it.

Elba is a way of wanting it again.

What Elba is

Elba is a fence you draw around one folder on your computer, and everything inside that fence becomes, at last, only yours.

That's the whole of it. You can stop reading now and you'd have the shape of the thing. But let's stay a moment, because the plainness hides how unusual it is.

Most tools that promise you privacy are, underneath, someone else's building you've been given a key to. They keep your things on their machines and swear not to look. The promise is only as good as the company, and companies change — they get bought, they get breached, they get a new set of lawyers with a new idea about what your data is for. You are asked, always, to *trust*.

Elba doesn't ask you to trust it, because it isn't holding anything. It has no account. It has no server. There is no building. It is one file that lives on your own machine, sets up camp inside a folder you choose, encrypts what you keep there, and — this is the part that matters — cannot send any of it anywhere, because it was never built with a way to. You'll see later exactly how we can promise that, and how you can check the promise yourself rather than take our word.

Think of your computer as a small country. Most of it is mainland — connected, mapped, patrolled by well-meaning services that mean you no harm and watch you constantly. Elba takes one patch of that country and makes it an island. What happens on the island stays on

the island. The boats that come to the shore — the sync clients, the indexers, the backup routines — can dock all they like. They just can't read the language anyone speaks there. To them, the whole island is written in a script only you can read, and you carry the only copy of the key in your head.

One folder. One password. One place. It will hold anything — the letters, the photographs, the spreadsheet of things you owe, the novel you don't show people, the scan of the passport, the file you don't like to think about — in as many rooms as you care to build. And when you close it, it closes. Not "logs out." Not "syncs and signs off." *Closes*, the way the drawer closed, the way the envelope sealed. The oldest gesture there is, done at last to the newest kind of paper.

04

Why a fence and not a vault

W e call it a fence, and the word was chosen carefully, so let's spend a page on why. It will tell you how the whole thing is meant to feel.

A vault is a box. You put your valuables *into* it, which means first you have to decide what counts as valuable. You stand there holding your life in your hands and sort it: this matters, lock it up; this doesn't, leave it out. A vault is for the few precious things. It assumes most of your life is ordinary and only some of it is secret.

But that isn't how privacy actually feels, is it? You don't want a safe-deposit box for three secrets. You want a whole patch of ordinary life — the dull and the tender and the half-finished, most of it not secret at all, just *yours* — to sit somewhere the world isn't looking over your shoulder. Not because any single thing in it would shock anyone. Because the looking itself is the thing you're tired of.

A fence is different from a vault the way a garden is different from a jewellery box. You don't fence a garden because each cabbage is a treasure. You fence it because it's *your ground*, and what grows there is your business, and a fence is simply how you say so. Some of it is prize roses. Most of it is potatoes. The fence doesn't care. It goes around the whole plot and declares the lot of it yours.

There's an old irony buried in the word, and if you like that sort of thing, here it is. To fence land — to enclose it — was once the act of

the powerful taking a shared commons *away* from ordinary people. But on your own computer the enclosure already happened, and it happened to you: your own files, fenced in by other people's software, grazed by other people's machines. Elba is you fencing your own ground *back*. Re-enclosing the commons in your favour, for once. Staking a post and saying: this stretch is mine again.

You don't sort. You don't decide what's precious. You draw a line and step inside it, and everything within is home.

05

The grey island

There is a small thing Elba does that we're quietly proud of, **H** because it turns the whole idea into something you can *see*.

Inside the fence, at rest, everything is grey. Your notes, your files, the rooms you've built — they sit there as plain grey cards, colourless, sealed. It can look, at first, a little austere. Give it a moment. There's an argument in it.

Out on the mainland, everything is bright. Every file has its thumbnail, its preview, its little burst of colour — because out there, everything is *visible*. The brightness of the ordinary computer is the brightness of a shop window: all of it lit, all of it on display, all of it legible to anything that passes. Colour, out there, is exposure.

On the island, we flipped it. Grey is the colour of being unseen — which is to say, grey is the privacy itself, made visible. A sealed thing is grey because it genuinely hasn't been opened; Elba isn't showing you a dimmed photograph of your file, it's showing you that it hasn't looked at your file at all. Nothing is lit because nothing is exposed. That's not a lack. That's the whole promise, drawn in the one language everyone reads without being taught.

And then you reach for something. You open one note, one file — and it wakes into colour in your hands. Just that one. It warms, it becomes legible, it's *yours in the clear* for as long as you hold it. Put it down and it greys again, sealed again, gone dark. The only thing that

can bring colour to the island is your own attention, and only to the single thing you're touching, and only for as long as you touch it.

Everyone else — every service, every crawler, every patient machine that meant you no harm — sees grey. Forever. Only the sovereign can colour the island, one patch at a time.

The mainland gives you forty bright things at once, all clamouring. The island gives you one, lit because you chose it. That is not a limitation we're apologising for. That is what the quiet was *for*.

PART TWO

Arriving

What you need

Very little, which is the nicest kind of requirement.

You need the one file — `Elba.html` — which you already have, or you wouldn't be holding its manual. You need a folder on your computer that you'd like to make yours; it can be empty, or it can be one you already use, it doesn't matter, Elba is happy to move in either way. And you need a particular kind of web browser, which is the one genuine catch, so let's be straight about it now rather than surprise you later.

Elba needs a *Chromium* browser to fence a real folder. That's Google Chrome, Microsoft Edge, or Brave — the common ones, likely already on your machine. The reason is dull and technical: the ability for a web page to safely write into a folder you've chosen is a fairly new power, and at the time of writing only those browsers have it. Firefox and Safari haven't built it yet.

If you open Elba in Firefox or Safari, it won't break — it'll simply run in a kind of demonstration mode, where the encryption all works but nothing is written to disk, so you can look around without committing anything. Useful for a first look. Not where you'd keep your life. For that, one of the Chromium three.

That's the whole list. No installation. No sign-up. No email address handed over, no box to tick, no "we've sent you a verification link." You open a file. That's the ceremony.

WHY CHROMIUM, PRECISELY

Elba writes to disk through the File System Access API (`showDirectoryPicker` and the `FileSystemDirectoryHandle` it returns). This API lets a page hold a durable, permissioned handle to a directory the user explicitly granted. As of this writing it ships in Chromium-based browsers and not in Gecko (Firefox) or WebKit (Safari). Everything else Elba uses – the Web Crypto API, IndexedDB, Blob handling – is available everywhere; only the disk-writing substrate is Chromium-bound. When that API is absent, Elba falls back to an in-memory store so the interface and the cryptography can still be exercised.

Claiming your ground

Open the file, and the first thing Elba asks is modest: which folder?

There's a button. It says *Claim a folder*. Press it, and your computer's ordinary "choose a folder" window appears — the same one you've used a thousand times. Find a folder, or make a new one, and choose it. That's you drawing the fence. You've told Elba: *this ground, here, is the ground I mean*.

Nothing dramatic happens on screen, and that's deliberate. Elba doesn't fill the folder with software or scatter its parts around your disk. It sets up camp — it writes a tiny, sealed record of what it needs, and otherwise leaves the folder looking much as it did. If you went and peeked at the folder in your file manager afterwards, you'd find a couple of files with names like nonsense and no way to open them. That nonsense is the point. We'll come to it.

A small thing worth saying plainly, because people worry about it: claiming a folder does *nothing* to the files already in it. Elba doesn't reach into your existing documents and transform them. It builds its own encrypted things alongside whatever's there. If you want to bring an existing file *into* the fence, you do that deliberately, later, by hand — and we'll show you how. Claiming is just choosing the ground. Nothing you didn't ask for gets touched.

If you'd rather not choose a real folder yet — if you just want to wander the island before you move in — there's a quieter link beneath the button: *explore a demo fence*. That gives you a pretend island that lives only in memory and vanishes when you close the tab. Everything works. Nothing persists. It's the show home, not the deed.

The one key

Now Elba asks for a password, and we need to talk about this properly, because it is the single most important decision you will make here, and the whole character of the thing lives in it.

Your password is not *a* key to Elba. It is *the* key. The only one. There is no master key held in reserve, no "forgot password" link, no support desk with a back way in, no company that can be subpoenaed into opening your fence because the company cannot open your fence either. What you choose here becomes, quite literally, the thing your files are made unreadable without. Lose it and they are not locked away from you. They are *gone* — encrypted rubble that no one, including us, including you, can turn back into words.

Read that twice, because your instinct will be to soften it, and we need you not to.

This sounds frightening. It is meant to. But turn it over, and the fear is the reassurance wearing a coat. The reason no one can recover your password *for* you is the very same reason no one can take your fence *from* you. A door with a second key is a door with a second keyholder. Elba has no second key. The total responsibility and the total sovereignty are the same fact, seen from two sides. You cannot have one without the other, and if a privacy tool ever offers you the sovereignty *without* the responsibility — "encrypted, and also we can help

you if you forget!" — check your pockets, because something has been taken.

So: choose well, and write it down somewhere the world can't reach but you can. A passphrase, not a password — four or five ordinary words strung together, copper thistle lantern ferry, is both far stronger and far easier to keep in your head than Xk9\$2q. Elba will nudge you toward strength as you type, and won't let you raise a fence on something flimsy, because the flimsy password is the one place the whole design can be undone, and we would rather be briefly annoying than quietly complicit.

WHAT YOUR PASSWORD ACTUALLY BECOMES

Elba never stores your password, and never sends it anywhere. When you type it, it's run through **PBKDF2-SHA-256** — a deliberately slow function — 310,000 times, over a random 16-byte salt generated for your fence, to derive a 256-bit key. That key exists only in your browser's memory, only while you're inside, and is discarded the moment you lock. The salt is stored in the clear (salts aren't secret; their job is only to make pre-computed attacks useless). Because deriving the key is slow by design, an attacker who somehow copied your whole folder would have to pay that cost on every single guess — which is exactly why the length of your passphrase matters more than any other choice you make. See *The one key*, again in Part Four for the full account, including the honest limitation: PBKDF2 is not the strongest such function available, and why a one-file program uses it anyway.

Raising the fence

You've named the ground and chosen the key. Press the last button — *Raise the fence* — and the island exists.

What happens in that instant is quiet and complete. Elba derives your key, writes a small sealed record so it can recognise your password next time without ever storing it, and drops a first note into your new island by way of welcome. Then it shows you the inside: a calm, near-black room with a single grey card in it, waiting.

That first note is yours to keep or throw away. It says, more or less, what we've been saying: that this ground is now fenced, that what you write here is sealed the moment you close it, that on the mainland these files are nonsense, that one thing opens at a time, and that there is no back door — so keep your key safe. A small ceremony. Read it once and you'll have the manners of the place.

And that's it. You've arrived. There was no installer's progress bar, no account confirmation, no tour with a cartoon guide pointing at buttons. You chose a folder, you chose a key, and a private place came into being on a machine you thought you'd already fully explored.

Sit in it a moment before you start filling it. It's rare, now, to be somewhere genuinely unwatched. You might feel the quiet as a slight pressure at first, the way silence rings after a loud room. That passes. What's left underneath it is the thing you came for.

PART THREE

Living Inside

10

The first thing you keep

The blank note is the most honest surface in the world. It doesn't flatter you and it doesn't watch you. Inside Elba, for perhaps the first time in years, it also doesn't *report* you.

To make one, you press *new note*. A clean writing space opens, and it is deliberately empty — a title waiting at the top, a body waiting below, an amber cursor blinking like a small patient light. There is no toolbar bristling with buttons, no formatting ribbon, no word count nagging in the corner unless you go looking. There is you, and the space, and the quiet.

Write. That's all. Elba's notes are plain text, which is the humblest and most durable format there is — the same words will be readable in fifty years when today's clever file formats are archaeology. Type a grocery list or the thing you've never told anyone; the space treats them the same, with the same indifference, which in a writing surface is a kind of grace.

When you're done, you close the note — the button says *Seal & close* — and in that moment the words are encrypted and set back down as a grey card among the others. Sealed. If you come back tomorrow and open it, it wakes into colour again, exactly as you left it. If you never come back, it waits, unreadable to everything but your key, for as long as the folder exists.

There's a particular feeling the first time you seal something true and watch it go grey. It's the feeling of the diary's little brass lock, forty years on. You'd forgotten a machine could give it to you.

11

Bringing things in

Notes are things you write. But a life is also made of things you receive — the scan of the passport, the photograph, the contract, the recording, the document from the solicitor with the long name. Elba holds those too.

You press *file*, and your ordinary "choose a file" window appears. Pick anything — an image, a PDF, a spreadsheet, a sound file, it doesn't much care — and Elba takes it in. It reads the file, encrypts it whole, and sets it down inside the fence as another sealed card. The original out on the mainland is still the original; what you've brought in is a private copy, sealed. If the thing is sensitive — and the things we scan and save usually are — you can now delete the mainland copy and keep only the fenced one.

Open a fenced file and Elba shows it to you in the clear: a picture displays, a document announces itself. When you're done looking, you seal it closed again. And if you ever need the real thing back out in the world — to email it, to print it, to hand it to someone — there's a button, *Take a copy out*, that hands you back the plain file to do with as you please. Bringing things in is not a trap. The fence has a gate, and the gate opens both ways, on your say-so.

A gentle word about large files. Elba will hold them, but everything inside the fence lives in your browser's care while it's open, so a folder stuffed with enormous videos will feel heavier than one of

notes and photographs. For the things that matter most — documents, images, the irreplaceable scans — it's perfectly at home. For your entire film collection, it's the wrong sort of shelf.

12

Rooms

Give a person a single drawer and within a week they'll want a divider. We're sorting creatures. So Elba lets you build rooms.

A room is just a named place inside the fence to gather related things — *Health*, *The House*, *Letters*, *Work I Don't Talk About*, whatever your life actually looks like. You make one by pressing *room* and naming it. You can put rooms inside rooms, as deep as you like, the way you'd nest folders, so a large and tangled life can be given the same shape it has in your head.

Here is the quietly remarkable part, and it matters more than it looks. The rooms are *inside the fence too*. Their names — *Divorce*, *Severance*, *The Thing With My Brother* — are sealed exactly like everything else. Out on the mainland, no one can see that you have a room at all, let alone what you called it. On an ordinary computer, your folder names are a confession written on the outside of the envelope; anyone glancing at the drive learns the headings of your life even if they can't read the letters. Inside Elba, even the headings are dark. The shape of your worry is nobody's business either.

You move between rooms by walking into them — press a room and you're inside it, with a little trail at the top showing where you are, *Elba / Health / Results*, each step of which you can press to walk back out. It's the plainest kind of navigation, and it's meant to be. You

already know how to do this. You've been doing it since your first computer. Elba just made the walls opaque.

13

Opening and closing

We've mentioned the ritual in pieces; let's name it whole, because it's the heartbeat of the place.

One thing opens at a time.

When you open a note or a file, it wakes into colour and everything else stays grey. When you open a second thing, the first one seals itself first — quietly, automatically, keeping your work — and *then* the second one opens. You are never holding two open things at once. The island lights one patch, the patch you're standing on, and no more.

This can feel, for about a day, like a constraint. Then it turns into the thing you didn't know you were missing. The mainland's great trick is to fracture you — forty tabs, a dozen half-read documents, everything open, everything lit, everything pulling at once, and you somehow busier than you are effective. Elba refuses that on your behalf. It gives you one thing, because one thing is what you can actually attend to, and because — this is the part that's also true underneath — one thing open means only one thing is unsealed in the machine's memory at any moment. The smallest possible exposure and the calmest possible desk turn out to be the same design, arrived at from two directions.

You came to the island to do one thing at a time. The interface simply won't let you forget it.

14

Locking up

Leaving is a single press. The button says *Lock*, and when you press it the key is dropped from the machine's memory, the open thing is sealed, the screen clears of anything readable, and you're back at the door asking for a password. The island goes fully dark. Nothing readable remains anywhere the machine can reach.

You don't have to remember to do it, though, which is the kindness. If you walk away — a phone call, a doorbell, the ordinary interruptions of a life — Elba notices the stillness, and after a while of no typing and no moving, it locks itself. It seals whatever you had open first, so nothing is lost, and then it drops the key and darkens, exactly as if you'd pressed the button yourself. You come back to a locked door and your own password, which is a small friction and a large protection: the thing you forgot to close closed itself.

There's a plainer way to say all of this. Elba is *shut by default*. Open is the exception, brief and deliberate and lit; shut is the resting state it always returns to, on its own, if you let your attention drift. Most of what watches us relies on our forgetting to close things. Elba closes them for you.

15

A day in the life

Let's put it in motion, because a list of features is a skeleton and you came for something that breathes.

Morning. You open Elba — the door, your passphrase, the four words you keep in your head. The island lights just enough to show you your rooms, grey and calm. You step into *Journal* and open last night's entry to add the thing you thought of in the shower; it wakes into warm light, you type two sentences, you seal it. It greys. You step back out.

You brought a form home from the doctor yesterday, a photograph of it on your phone, already copied to the laptop. You press *file*, bring it into *Health*, and then — this is the quiet luxury — you delete the copy sitting out in your ordinary Pictures folder, where the backup service and the photo app and heaven knows what else had already had a good look. Now the only copy that isn't nonsense-to-everyone lives behind the fence.

Lunch. Someone needs the insurance document. You open it in Elba, press *Take a copy out*, and the plain file drops into your downloads, ready to attach to an email. The fence has a gate. You use it, and close it behind you.

Evening. You're writing something you're not ready to be read — a letter, a chapter, a resignation you haven't decided to send. You open it, and for as long as you're in it, it's lit and legible and yours, and the

machine underneath is doing nothing with your sentences but holding them. You get up to make tea and don't come back for an hour. When you return, the door is closed and the island dark. It locked itself while the kettle went cold. You type your four words and it's all still there, exactly as you left it, waiting.

None of that is remarkable. That's what's remarkable about it. It's an ordinary day, conducted for once without an audience.

16

What people keep here

We can't tell you what belongs on your island, and we wouldn't. But people ask what it's *for*, as if privacy needed a permit, so here is the honest range of it — not to prescribe, only to give you permission you shouldn't need.

Some keep a **journal**, and mean it — the real one, the one with the sentences you'd never write if you thought a machine was reading over your shoulder, which of course it always was. Elba is, among other things, the first genuinely private place to write badly and truly, which is the only way anyone writes anything worth keeping.

Some keep the **paperwork of being alive** — the passport scan, the birth certificate, the deed, the policy, the tax years, the long documents with the official seals. The things that would be a nightmare in the wrong hands and are a mild ambient anxiety in the ordinary Documents folder. Fenced, they stop being a worry and become simply filed.

Some keep the **keys to everything else** — the list of passwords, the recovery codes, the seed phrase, the answers to the security questions that are really just more secrets. There's a certain sense in one strong passphrase, kept in your head, guarding the door to all the others.

Some keep **work in progress** — the novel, the invention, the case they're building, the resignation letter, the business that doesn't exist yet. Unfinished things are tender precisely because they're unfinished;

they need somewhere to be ugly and provisional without being seen. A fence gives the unformed thing room to stay unformed a while longer.

And some keep grief, and love, and the things that don't have a folder on the mainland because the mainland has no idea what to do with them. The last messages from someone gone. The photographs you can't look at yet but can't delete. The letters to people who'll never read them. These are not files, exactly. But they live as files, and they deserve somewhere the second reader never goes.

Notice what runs under all of it. Not one of these is *shame*. Not one of them is a thing to hide because it's wrong. They're things to keep because they're *yours* — ordinary, mostly, and tender, and none of anyone's business by the plain fact of being yours. That distinction is the whole of Elba. You are not building a place to hide guilty things. You are building a place to be, unwatched, with the ordinary contents of a life.

The mainland could never tell the difference between a secret and a private thing. Elba doesn't have to. It seals them both, and asks you nothing.

PART FOUR

The Machinery

What happens when you type

You don't need to read this part. That's worth saying up front. Elba works whether or not you understand how, the way a lock works whether or not you understand its pins. But some people can't quite trust a thing they can't see inside, and if you're one of them, this part is for you — the machinery, in plain language, with the precise version boxed off to the side.

So: what happens to a sentence when you seal it?

While the note is open, your words are just words, sitting in the machine's working memory the way any open document does. Nothing is protecting them in that moment except that you're the one looking. The protection happens at the *edge* — the instant you seal. In that instant, Elba takes your words and runs them through a piece of mathematics called a cipher, which scrambles them into a mess that has no pattern anyone can find and no way back to the original — *unless* you hold the key. With the key, the mess unscrambles in an eyeblink. Without it, the mess is permanent. Not "hard to reverse." As far as anyone knows how to compute, *not reversible at all*.

That scrambled mess is what gets written to your folder. Never the words. Only the mess. Every time you seal, a fresh scramble; open the same note twice, seal it twice, and the two scrambles on disk look completely unrelated, even though the words are identical. This isn't decoration. It's what keeps the pattern-finders from finding a pattern.

The key, remember, isn't stored anywhere. It's conjured from your passphrase each time you enter, used while you're inside, and thrown away when you leave. So the mess on disk and the key in your head are two halves of a thing that is only ever whole inside your own open session, on your own machine, at your own hand.

THE CIPHER, PRECISELY

Elba encrypts with **AES-256-GCM**, the Advanced Encryption Standard at a 256-bit key length, in Galois/Counter Mode. AES is the block cipher trusted for classified and financial data worldwide; GCM is an authenticated mode, meaning it doesn't only conceal the data, it also produces a tag that detects any tampering – a single altered byte in the ciphertext causes decryption to fail cleanly rather than returning garbage. Each encryption uses a fresh random 96-bit initialisation vector (the "nonce"), which is why identical plaintext yields unrelated ciphertext each time. All of it runs through the browser's built-in **Web Crypto API** (`crypto.subtle`), not a hand-rolled implementation – the same vetted cryptographic code the browser uses for HTTPS itself.

The one key, again

promised, back at the door, a fuller account of what your **W** passphrase becomes. Here it is, because this is the hinge the whole thing turns on, and you deserve to see it clearly — including the part we're not proud of.

A passphrase is not a key. It's too short, too guessable, too human to be a key directly. So Elba puts it through a second machine — a *key-derivation function* — whose only job is to turn a human passphrase into a proper cryptographic key, and to do it *slowly*. Slowness is the feature. It means that anyone trying to guess your passphrase by brute force has to pay a real cost — a slice of a second — on every single guess. A slice of a second is nothing to you, once, at the door. Multiplied across the billions of guesses an attacker would need, it becomes years, centuries, the heat-death of their patience.

But — and here is the honest part — the particular slow machine Elba uses is not the strongest one that exists. It's a good one, a standard one, but there are newer designs that are harder still for an attacker with specialised hardware. Elba doesn't use them, and the reason is a promise we made you elsewhere: that Elba is *one file*, beholden to nothing, importing no outside code. The stronger machines can't be had without pulling in a substantial extra component, and that would break the thing that lets you trust the file in the first place. So we chose the honest trade: a standard slow

machine, tuned as hard as we reasonably can, and a plain request to *you* to carry the rest of the weight with a long passphrase.

This is the shape of every real security decision — a trade, made in the open. Anyone who tells you their tool is simply, unqualifiedly secure is selling you the *feeling* of security, which is the one thing security cannot include.

DERIVATION, PRECISELY – AND THE LIMITATION

The key is derived using PBKDF2 with HMAC-SHA-256, at 310,000 iterations, over a random 16-byte salt unique to your fence and stored in the clear alongside the encrypted data (a salt is not a secret; it exists to defeat precomputed "rainbow table" attacks and to ensure two fences with the same passphrase derive different keys). The derived key is 256 bits, marked non-extractable, held only in memory, and discarded on lock.

The honest limitation: PBKDF2 is not memory-hard. Modern password-cracking rigs using GPUs or ASICs parallelise it efficiently, so it offers less resistance per iteration than Argon2 or scrypt, which are deliberately memory-hungry. Elba uses PBKDF2 because it is available natively in Web Crypto, whereas Argon2 would require bundling a WebAssembly module – extra code to audit and a departure from the single-file design. The practical consequence is real and simple: the strength of your fence is dominated by the entropy of your passphrase. A five-word random passphrase is comfortably beyond any foreseeable brute-force. A weak password is weak no matter what derives from it. This is why Elba refuses flimsy passphrases at the door rather than letting you undo the whole design with one lazy choice.

What the folder really holds

Go and look. Open your fenced folder in your ordinary file manager, the way you'd look in any folder. This is the most reassuring thing you can do, so do it.

You'll find a handful of files with meaningless names — a string of random characters, ending in `.elba` — and nothing else recognisable. No filenames that mean anything. No folders named after your rooms. No thumbnails, no previews, no readable text anywhere. If you open one of these files in a text editor out of curiosity, you'll get a wall of gibberish: the scrambled mess, and nothing but.

This is what the mainland sees. This is the whole of what your operating system, your backup service, your cloud sync, a snooping housemate, or a patient machine can learn from your fenced folder: that it contains some files, that the files are sealed, and *nothing else*. Not their contents. Not their names. Not the names of your rooms. Not how anything is arranged. The map of your island is itself drawn in the sealed language.

That last point deserves its own breath. On an ordinary encrypted tool, people often forget that the *structure* leaks — you can't read the files, but you can see there's a folder called `Divorce` with forty things in it, and the shape of a person's trouble is legible from the outside even when the details aren't. `Elba` doesn't do that. The names, the rooms, the nesting, the very count of how your life is sorted — all of

it lives inside a single sealed record, so from outside there is no shape to read at all. Just an even scatter of nonsense.

ON-DISK FORMAT, PRECISELY

Each item is a file with a random 16-hex-character name and an `.elba` extension, containing a small JSON envelope: an encrypted `meta` field (title, type, room path, timestamps) and an encrypted `payload` field (the note text or file bytes), each with its own IV. Filenames carry no information. There are no real subdirectories – the room hierarchy is virtual.

The structure lives in a single encrypted `index` file. Critically, each item also carries its own sealed self-description, so if the index is ever lost or corrupted, Elba rebuilds the entire structure by reading the items themselves – the map can be reconstructed from the territory. (Genuinely empty rooms, having no items, are the one thing a rebuild can't recover.) Every ciphertext is additionally bound, via GCM's additional authenticated data, to the filename it lives in and the role it plays, so an item cannot be silently swapped or replayed into another's place.

20

Nothing leaves

Here is the claim the whole product is named for, and here is why you don't have to take it on faith.

Elba makes no connection to the internet. None. Not to fetch a font, not to check for updates, not to send a single anonymous crumb of "usage data" of the kind that every app now helps itself to. There is no server it talks to because there is no server at all. The entire program runs on your machine, in front of you, complete.

Most tools would ask you to believe this. Elba lets you *enforce* it, using a feature built into your own browser. The program declares, in its own opening lines, a strict rule that forbids it from opening any network connection whatsoever — and the browser holds it to that rule. Even if the code wanted to phone home, even if some future meddler tried to make it, the browser would refuse on your behalf. The promise isn't guarded by our good intentions. It's guarded by something that doesn't work for us.

And you can check it yourself, in about ten seconds, with no technical skill. In your browser, view the page's source — right-click, *View Page Source*. Search it for the word `http`. Search it for `connect`. What you're looking for is the absence of anywhere to send anything, and the presence of the rule that forbids sending. It's all right there, in a file you can read. A privacy promise you can personally audit before lunch is a different kind of promise from one you're asked to trust.

THE ENFORCEMENT, PRECISELY

Elba ships a **Content-Security-Policy** meta tag including `default-src 'none'` and, explicitly, `connect-src 'none'`. This instructs the browser to block all network-connecting APIs – `fetch`, `XMLHttpRequest`, `WebSocket`, `sendBeacon`, everything – regardless of what the JavaScript attempts. Fonts are inlined as base64 (`font-src data:`), so even typography triggers no request. The policy is enforced by the browser's own engine, not by application code, which means it holds even against code injected or modified after the fact. The technical page ([how the fence works](#)) states the full policy for inspection.

21

The sealed map

etouched this above, but it earns its own moment, because **W** it's where privacy and *durability* meet — and durability is the promise people forget to ask about until the day it fails them.

An encrypted pile of files is only useful if something remembers how they fit together — which file is which note, what's called what, which room holds which things. Elba keeps that memory in a single sealed record, the index, encrypted like everything else so it leaks nothing. Fast to read, private by default.

But a single record is a single point of failure, and a privacy tool that loses your organisation the first time that record gets corrupted would be a cruel joke. So Elba does something quietly careful: every item carries, sealed inside itself, a small description of *what it is and where it belongs*. The index is the fast path; the items themselves are the safety net. If the index is ever damaged or lost — a bad sync, a half-finished write, an act of ordinary computer misfortune — Elba doesn't shrug and lose your world. It reads every item in turn, hears each one say where it goes, and rebuilds the whole map from scratch.

This is the same instinct as the fence itself: don't rely on trust where you can rely on structure. Don't rely on one fragile record where the truth can be recovered from the things themselves.

Hiding the shape

The last piece of machinery, and it's a subtle one, but it closes a gap most tools leave wide open.

We said the mainland can't read your files or their names. True. But there's a quieter leak in every encrypted store, and it's this: the *size* of a sealed file tells you something about what's in it. A one-line note makes a small file; a long letter makes a big one. Watch the sizes and you learn the rhythm of a life — which day someone wrote pages, which day a line. You can't read the diary, but you can see how much was felt.

Elba closes that gap by *padding*. Before it seals anything, it quietly pads the contents out to a standard size, so that a two-word note and a two-paragraph note come to rest as files of exactly the same size on disk. The length of your small things stops leaking. From outside, you can't tell the terse day from the overflowing one. The padding is generous where things are small — where the leak matters most and the cost is nothing — and gentle where things are large, so a big file isn't wastefully doubled.

It's a small mercy, and it's the kind of thing that separates a tool that's *mostly* private from one that's been thought all the way through. Privacy isn't only about the words. It's about the shadow the words cast — the sizes, the counts, the timing, the shape. Elba spends real

effort flattening the shadow, because the shadow is where the patient machines do their best reading.

PADDING, PRECISELY – AND WHAT STILL LEAKS

Before encryption, each plaintext is length-prefixed and padded to a bucket boundary: fixed coarse buckets (256 B, 1 KB, 4 KB, 16 KB, 64 KB) for small items, then rounding to the next 64 KB for larger ones. Two items in the same bucket produce byte-identical file sizes, so exact length – and thus title verbosity and small-content size – is concealed. The index is padded identically, hiding item count and title lengths within a bucket.

What still leaks, stated plainly: the number of `.elba` files reveals the approximate count of items (hiding it would require dummy files or a single-blob store, each with its own costs); a very large file still reveals its size to 64 KB granularity; and file modification timestamps, set by the operating system, reveal when items changed. These are inherent to storing real files in a real folder, and are documented rather than disguised.

PART FIVE

Keeping It

Copies

A fence protects what's inside it from being read. It does nothing whatever to protect it from being *lost* — and the two are different fears that people constantly confuse. Let's separate them cleanly, because the answer to the second is happy and simple.

Your fenced folder is just a folder. Sealed, yes, but an ordinary folder of ordinary files as far as your computer is concerned. Which means you back it up exactly like anything else, and here is the lovely part: *you can back it up anywhere at all, including the places you'd never dream of trusting with your secrets.*

Copy the folder to a USB stick. Drop it in your cloud storage — Dropbox, Google Drive, iCloud, the very services this whole manual has been side-eyeing. Email the folder to yourself. Leave a copy on a second machine. Do all of these. Because what those services receive is the sealed language, the scramble, the nonsense — and they can hold it forever and never read a word. You have inverted the usual bargain. Normally the cloud gives you durability and takes your privacy in exchange. Elba lets you take the durability and keep the privacy, because what you're handing over is already sealed before it leaves your hands.

Store your island in the enemy's warehouse. Let them keep it. They can't set foot on it. And now if your laptop is lost or stolen or simply dies the way laptops do, your fence — and your key, which lives in

your head — survive, and you carry your island to a new machine intact.

Back things up. It's the one homework this manual sets you. A fence with no copies is a sovereign territory of exactly one bad day's duration.

Remembering

By default, Elba is forgetful on purpose. Close the tab and it leaves **B** nothing of itself behind on your machine — no trace, no record, no crumb saying you were ever here. Next time, you point it at your folder afresh. This forgetfulness is a feature: the tool that leaves no footprint is the tool that can't betray a footprint.

But forgetfulness is a small daily tax — re-choosing the folder each time — and some people, on their own private machine, would rather Elba remembered where home was. So there's a choice, and it's yours to make, and it's off until you turn it on.

Inside the fence there's a *remember* switch. Turn it on, and Elba leaves a single small pointer on that machine — not your password, not your files, nothing readable, only a note to itself saying *the fence lives in that folder over there*. Next time you open Elba on that machine, it greets you with *Reopen*, and you skip the folder-choosing and go straight to your passphrase. Turn the switch off, or press *forget* at the door, and that pointer is wiped, and Elba goes back to leaving no trace.

We built it this way — off by default, a deliberate switch, forgettable from both sides — because leaving a trace on a machine should always be *your* decision, made knowingly, and never a convenience that quietly happens to you. On your own private laptop, remembering is a small kindness. On a shared or borrowed machine, you'll want

the forgetfulness. Elba lets you choose, every time, and defaults to the more private option, which is how these choices should be shaped.

Note the thing that *never* changes: remembering the folder never means remembering the *key*. Reopening still asks your passphrase, always. The pointer gets you to the door faster. It never opens it.

25

The long horizon

Most software is built to live forever and belong to its maker **M** forever — to keep you paying, keep you locked in, keep the source hidden so you can never quite see what it does or take it elsewhere. Elba is built on the opposite plan, and this is the strangest and most deliberate thing about it, so let us tell you plainly what will happen.

Elba is sold, for its first years, at a price that *falls* every year — because what you're really buying is the time between now and the day it becomes free, and that span shrinks. And on the first of January, 2030, Elba stops being ours. On that date, by the terms of its own licence, it becomes open source and passes to the Commons. Not disabled. Not discontinued. *Freed* — every line of it published for anyone to read, keep, run, change, and pass on, forever, at no cost, with no one's permission.

We call this *Mortalware*: software with a death date for its ownership. Our exclusive claim on it is the mortal thing. The program itself is what survives that death and becomes immortal by becoming everyone's.

Why do this? Two reasons, one principled and one practical. The principled one: a tool that asks you to trust it with your most private things has no business being a permanent black box. Its promise to you should be, eventually, *checkable by anyone* — and in 2030 it will

be, line by line. The practical one: a thing you keep for a lifetime shouldn't depend on a company surviving for a lifetime. Companies die. When Elba belongs to the Commons, it cannot be bought, shut down, discontinued, or ruined by a bad quarter. It will simply be there, the way a public library is there, for as long as anyone cares to keep a copy.

You are, in a small way, buying a thing that has agreed in advance to stop being owned. There aren't many of those.

Passing it on

Which brings us to the longest horizon of all, the one nobody likes to plan for and everybody should.

You are going to die. Not soon, we hope, and not the subject of this manual — but the files will outlive the writing of them, and some of them are meant for people who aren't you. The letter for your child to read later. The location of the important documents. The account nobody else knows exists. The things you'd want found, by the right person, at the right time, and by no one before.

A fence with no second key is a beautiful, terrible thing here. It means that if you tell *no one* your passphrase, your island dies with you — sealed forever, unreadable, a locked room no locksmith can open. For some of what you keep, that's exactly right; some things are meant to end when you do. But for the things meant to be *found*, you have to do the human part yourself, because Elba deliberately has no mechanism to do it for you — no "trusted contact," no dead-man's switch, no company holding a spare key against the day, because every one of those would be the back door we swore there wasn't.

So the inheritance plan is old and simple and entirely offline. Write your passphrase down. Put it somewhere a trusted person will find it when the time comes and not before — a sealed letter with a solicitor, a safe-deposit box, an envelope marked and lodged with someone who loves you. The passphrase, and a plain note saying *this opens the*

folder called such-and-such, on the machine in the study, or in the backup on the drive in the drawer. That's the whole ceremony. The fence, the folder, the backup, and one written key in trustworthy hands.

It's the same as it ever was, really. The diary with the brass lock also needed someone to be handed the little key. We've changed the paper and the lock beyond recognition. The act of trust at the end of a life is exactly as old as it always was.

PART SIX

Where the Fence Ends

27

The door with no second key

We have said this several times, gently, in passing. Now we say it once, directly, with no softening, because it is the most important sentence in the manual and the one you're most likely to wish away.

If you lose your passphrase, everything inside your fence is gone, and no one — not us, not you, not anyone — can get it back.

There is no reset link. There is no recovery code we quietly kept. There is no support desk that can verify your identity and let you in, because there is no "us" holding a spare key, and no identity to verify against, and no door but the one your passphrase opens. Your files aren't locked away in some vault we could crack open for you if you begged. They're transformed, mathematically, into noise that only your key turns back into words. Lose the key and you don't have locked files. You have noise.

We are not apologising for this. We're telling you because it is the exact price of the thing you wanted. You wanted a place no one else could open. This is what that costs. The very fact that makes your fence impregnable to them makes it impregnable to a forgetful you. There is no version of "no one can get in" that has an exception for you-having-lost-the-key, because "the person with the key gets in" is the *only* rule, and forgetting the key makes you, cryptographically, not that person.

So the limitation and the whole point are one thing. Write the passphrase down. Keep the copies. And make your peace, now, with the fact that the safety and the risk are the same wall, seen from your two sides of it.

What a fence can't do

A fence stops people reading your ground. It does not stop a vandal trampling it.

Be clear about the two, because they're different fears with different answers. Elba guarantees *confidentiality*: what's inside cannot be read without your key. It does not, and cannot, guarantee *integrity* against someone who can reach your folder and *write* to it. Somebody with access to your actual files could delete them. They could replace your latest sealed note with an older sealed copy they'd kept, rolling you back to a previous version without your knowing. They can't read any of it — the seal holds — but they can vandalise the plot even if they can't read the diary.

Why can't Elba stop this? Because stopping it would require some trusted place to record "this is the true, latest version" that the vandal couldn't also tamper with — and a single file on your own machine has nowhere to anchor such a record that an attacker with write access couldn't equally reach. It's an honest limit of the form. A lone program guarding a lone folder can make that folder unreadable; it cannot make it un-deletable by someone who already holds it.

The answer is the same homely one as always: backups, kept somewhere the vandal can't reach. Confidentiality Elba gives you outright. Integrity against an active attacker, you get the old-fashioned way, with copies in more than one place. Against the threat

Elba is actually *for* — the patient, watching, reading kind — the fence holds completely. Against a determined vandal with their hands on your disk, the fence holds the *secret* and you hold the *spare*.

The room you're standing in

There is a limit that no encryption tool can escape, and the honest ones say so.

While you have something open — a note in the clear, a file displayed — it is, in that moment, unsealed, sitting in your browser's memory so you can read it. That's not a flaw; it's what "open" means. But it means the protection has a shape: it guards your things *at rest*, sealed in the folder. It cannot guard the one thing you're actively looking at from something that has already gotten *inside your browser* — a browser that's been compromised, or a malicious extension with the right permissions, could read what's on your screen the same way you can, because at that instant it isn't sealed.

Elba does what a well-made tool can do about this: it keeps the open window as small as possible. One thing open at a time. Sealed again the moment you close it. The key dropped, the memory cleared, the screen wiped the moment you lock. It shrinks the exposure to the single thing in your hand for the brief moment it's in your hand. But it can't defend a room that something is already standing inside. No program running in your browser can.

The plain guidance, then: keep the browser you fence in clean. Be miserly about extensions, especially the ones that ask to "read and change all your data on the websites you visit." Elba is a good lock on

a good door. It cannot help you if you've invited someone into the house.

The weather outside

covered this in the machinery, but it belongs here too, among the limits, stated as a limit and not a triumph.

Elba hides your files, their names, your rooms, the structure of everything, and — through padding — the length of your small things. That's most of the shadow. But a little shadow remains, and we'd rather name it than let you discover it and feel misled.

Someone watching your fenced folder from outside can still tell *roughly how many things* are in it, because there are that many sealed files sitting there, and short of stuffing your folder with decoys there's no hiding a count of files. They can tell the *approximate size* of a genuinely large file, because a large thing is a large thing even sealed, and padding it entirely would waste real space. And they can see, from timestamps your operating system writes and Elba doesn't control, *when* you last changed something — the rhythm of your visits, if not their content.

None of this reads your diary. All of it is the faint outline of a life rather than its contents — the number of rooms lit, roughly, and when the light was last on. For the great majority of people and purposes, it's nothing. For a few, in a few situations, the outline itself matters, and those people should know that the outline exists. We'd rather tell you the fence casts a faint shadow than pretend it casts

none, because the tools that pretend are the ones that eventually let you down in the dark.

31

The edges of the map

A short, plain gathering of the smaller boundaries, so none of them surprises you later.

Elba fences real folders only in *Chromium browsers* — Chrome, Edge, Brave. Elsewhere it runs as a demonstration with no disk. This may change if other browsers add the necessary capability; today, it's a wall.

Elba is *only as available as your file and your browser*. There's no cloud copy that follows you around, which is the whole point, and also means that getting your fence onto a new machine is something *you* do — carry the folder, or restore it from your backup — rather than something that happens automatically. The absence of magic here is deliberate. Magic is another word for someone else's server.

And Elba is *deliberately spare*. It does one thing — hold your private things — and it does not try to be your calendar, your notebook-with-formatting, your photo album, your everything. Some of the things you might wish it did, it doesn't do, and won't, because every feature is a new surface to secure and a new promise to keep, and a tool that guards your secrets should stay small enough to be understood entirely. Its narrowness is a form of trustworthiness.

These aren't the dramatic limits. They're the small print, and we've set it in the same size as everything else, because small print set small is where trust goes to hide.

PART SEVEN

Companionship

32

Small rituals

A tool becomes part of a life through repetition, and the repetitions that stick are the ones that feel less like using software and more like a small habit of care. You'll find your own. Here are a few people report, not as instructions — you've had enough of those — but as company.

There's the *evening seal*: the last thing before the laptop closes, opening the day's note, setting down the one sentence that wouldn't leave you alone, and watching it go grey. A diary kept in single lines, some days, and that's plenty. The point was never volume. The point was that it went somewhere true.

There's the *incoming rite*: every sensitive thing that arrives — the letter, the scan, the form — brought straight in through the gate and its mainland copy deleted, so that within a day of anything reaching you, the only unsealed copy is the one behind the fence. It becomes reflexive. A tidiness of the anxious kind, which is the best kind, because it actually resolves the anxiety instead of just organising it.

And there's the *quiet check* — every so often, opening the folder in the ordinary file manager just to see the nonsense sitting there, unreadable, exactly as promised. It's a strange comfort, looking at the gibberish that is your own life to everyone but you. People do it more than they'll admit. It's the modern equivalent of trying the lock before

bed. Not because you doubt it. Because feeling it hold is its own small peace.

33

On not needing us

We'll say something now that no manual is supposed to say, because most manuals are written by people who want you to keep needing them.

We hope you forget about Elba.

Not the fence — the fence we hope becomes so ordinary, so woven into how you keep things, that you stop noticing it's there, the way you don't notice the lock on your front door until the rare moment you think about it and are glad. We mean we hope you forget about *us*. The makers. We're not meant to be a presence in your private life. We built a thing and we're stepping back from it — a few years from now, entirely, when it becomes everyone's. A tool for keeping your own counsel should not come with a proprietor peering in, and ours doesn't, and one day soon it won't even come with a proprietor.

This is the strange courtesy of the whole design. Elba is built so that its maker's continued existence, goodwill, and attention are things you will never have to depend on. No account to be managed. No relationship to maintain. No terms that change after the acquisition. You bought a fence, not a friendship, and the fence will keep holding whether or not we're around to be thanked.

That's not coldness. It's a specific kind of respect — the kind that knows the most private corner of your life is precisely the place a

company has no business lingering. We made you somewhere to be alone. It would be a poor gift if we kept standing in the doorway.

A soft landing

Somewhere in your house, probably, there's a drawer that doesn't fully open. A box at the back of a cupboard. A tin that once held biscuits and now holds the things without a category — a child's tooth, a ticket stub, a photograph of people whose names are going soft, the ring that doesn't fit anyone anymore.

Nobody audits that tin. No service indexes it. It isn't backed up to a building you've never seen. It is simply *yours*, in the old, plain, unremarkable way that so much of life used to be and quietly stopped being, one convenience at a time, while you were busy and grateful and not quite watching.

That's all this was ever trying to be. A tin at the back of the machine. A drawer that shuts. A patch of ground the weather can't read. Somewhere the second reader never goes, and the only light is the one you bring, and what you keep there is nobody's business for the plain and sufficient reason that it's yours.

The kettle's boiled. The house is quiet. Go and keep something.

APPENDIX A

Quick Reference

35

At the door

Claim a folder — choose the folder that will hold your fence. **Explore a demo fence** — look around a temporary, in-memory island that vanishes when you close the tab. **Reopen** — if you've turned on *remember* on this machine, go straight to your passphrase for the remembered fence. **Raise the fence (new fence)** — set your passphrase and create the island. **Enter (existing fence)** — type your passphrase to open your island.

36

Inside the fence

+ **note** — create a new plain-text note. **+ file** — bring a file in from your computer; it's encrypted on the way in. **+ room** — make a named place to gather things; rooms can nest. **Open** (press any card) — decrypt and view one item; anything already open seals itself first. **Seal & close** — encrypt the open item and set it back down as a sealed card. **Take a copy out** — export a plain, decrypted copy of a file back to your computer. **Remember / Remembered** — toggle whether this machine keeps a pointer to your fence. **Lock** — drop the key, seal everything, clear the screen, return to the door.

37

Things that happen on their own

Auto-lock — after about ten minutes of no activity, Elba seals what's open and locks itself. **Re-seal on close** — every item is encrypted again the instant you close it. **Index rebuild** — if the structure record is lost, Elba reconstructs it from the items themselves.

APPENDIX B

A Short Glossary

Words as Elba uses them

Fence — the encryption drawn around one chosen folder; the boundary that makes the folder's contents unreadable without your key. Elba's central metaphor and its central mechanism.

The mainland — everything on your computer and network *outside* the fence: your operating system, sync clients, backup services, indexers, and the wider internet. Not the enemy, exactly. Just not yours alone.

The island — the fenced folder as you experience it from inside: your rooms, notes, and files, decrypted and legible only to you, only while you're there.

Room — a named, nestable place inside the fence for organising items. Its name and structure are sealed like everything else.

Item — anything you keep: a note or a file, stored as a single sealed unit.

Sealed / to seal — encrypted / to encrypt. A sealed item is grey, at rest, unreadable. Sealing happens automatically the moment you close something.

In the clear — decrypted and readable; the temporary state of the one item you have open.

The key — the cryptographic key that unseals your fence, derived fresh from your passphrase each time and never stored. Colloquially, your passphrase.

Mortalware — Elba's model of ownership: sold now, its price falling yearly, and released to the Commons as open source on 1 January 2030. The ownership is mortal; the freed program is not.

APPENDIX C

For the Technically Minded

The whole stack, briefly

For readers who want the machinery gathered in one place rather than scattered through the boxes. Everything below is inspectable in the source of the single `Elba.html` file, and will be fully open on 1 January 2030.

Cipher. AES-256-GCM via the Web Crypto API (`crypto.subtle`). Fresh random 96-bit IV per encryption. GCM's authentication tag provides tamper-detection per item; altered ciphertext fails to decrypt rather than returning corrupt plaintext.

Key derivation. PBKDF2-HMAC-SHA-256, 310,000 iterations, random 16-byte per-fence salt (stored in the clear). 256-bit derived key, non-extractable, memory-only, discarded on lock. Limitation: PBKDF2 is not memory-hard; passphrase entropy dominates real-world strength. Argon2 is deliberately not used, to preserve the single-file, zero-dependency design.

Context binding. Every ciphertext is bound via GCM additional-authenticated-data to its filename, its role (meta vs payload), and a format version, preventing silent swap or replay of items.

On disk. Random-named `.elba` files, each a JSON envelope of encrypted meta and payload. No meaningful filenames; no real subdirectories. Structure held in a single encrypted index; every item additionally self-describes so the index can be rebuilt from the items if lost.

Size-padding. Length-prefixed plaintext padded to bucket boundaries (256 B / 1 KB / 4 KB / 16 KB / 64 KB, then 64 KB steps) before encryption, concealing exact lengths. Residual leaks, documented: item count, large-file size to 64 KB, OS modification timestamps.

Network. None. Content-Security-Policy `default-src 'none'`; `connect-src 'none'` enforced by the browser; fonts inlined as `data:;` zero requests of any kind, verifiable in source.

Persistence. The directory handle may optionally be stored in IndexedDB (opt-in "remember"), containing no key or content — only a re-permissionable pointer to the folder. Reopening always requires the passphrase.

Threat model, one line. Elba assures confidentiality of data at rest against anyone without the passphrase, including the party storing the folder. It does not assure integrity against an attacker with write access (deletion, rollback), nor protect data in memory while open, nor conceal item count or timing. Backups address loss and rollback; a clean browser addresses in-memory exposure.

APPENDIX D

Questions People Ask

40

Plainly answered

I forgot my passphrase. What are my options? None, honestly, and we're sorry. There's no recovery by design — the same design that keeps everyone else out. If you kept a written copy somewhere safe, this is its moment. If not, the fenced items can't be recovered by anyone. This is the one place the manual begs you, in advance, to keep a written copy.

Is my stuff safe in the cloud if I back the folder up there? Yes — that's the intended use. The cloud receives only sealed, unreadable files. It cannot read them; it can only store them. Back up freely.

Can you, the makers, see my files or help police/lawyers see them? No. We have no server, no copy, no key, and no capability. There is nothing for anyone to compel us to hand over, because we hold nothing. This isn't a policy we promise to keep; it's an architecture that makes the question moot.

What happens to my fence if your company disappears? Nothing. Your fence is files on your machine and a key in your head; neither depends on us. And on 1 January 2030 the whole program becomes open source and belongs to the Commons, so it outlives us by design.

Why can't I use Firefox or Safari? They haven't yet built the browser capability Elba needs to write to a folder you choose. In those browsers Elba runs as a no-disk demo. Use Chrome, Edge, or Brave to keep a real fence.

Is a single HTML file really secure enough for this? The single file is a *feature*, not a compromise: it's small enough to read entirely, depends on nothing it could be poisoned through, and uses the browser's own vetted cryptography rather than anything hand-rolled. Its limitations — chiefly PBKDF2 over Argon2 — are stated plainly in Appendix C, and its network silence is enforceable and checkable by you. "Small enough to understand" is itself a security property.

Can I keep more than one fence? Yes — point Elba at different folders for different fences, each with its own passphrase. But the design's spirit is *one place*: a single fenced ground that holds everything, organised into rooms. One strong passphrase you'll never forget beats five you might.