THE CAPABILITY BROCHURE

AI You Can Prove

What we can actually do, with the measurement behind every claim: four capabilities each carrying its own dated, verified number — trust certified at three layers, safety by construction, fifteen years of rail heritage — and, at the end, exactly how to check us, because a brochure you can't verify is just adjectives.

For anyone deciding whether to believe us

🟢 Shipped — measured, gate-clean

COVER: TWO INDEPENDENT RUNS — NATIVE AND INTERPRETER — CONVERGE INTO ONE PROVEN LINE. THE GATE IS THE COVER.

vocabotics

THE VOCABOTICS CAPABILITY BROCHURE

AI You Can Prove

What we can actually do, with the measurement behind every claim: four capabilities each carrying its own dated, verified number — trust certified at three layers, safety by construction, fifteen years of rail heritage — and, at the end, exactly how to check us, because a brochure you can't verify is just adjectives.

PREPARED
11 July 2026

VERIFIED BY
Jon Ross — Founder, vocabotics — 15 years building safety-critical AI

HOW TO READ IT
Drafted with AI, verified by a named human. Sources at the back.

01 The one sentence

Not a pitch — a description. Every clause has a measured number behind it.

> ***An AI writes the code, in a language built for it; the code is proven correct by construction; most of the thinking is calculated, not trained; and it runs a capable model on hardware you already own.***
>
> The dashboard's hero sentence — every clause 🟢 measured, 2026-07-02

Most AI companies ask you to believe them. This brochure is built so you don't have to. Every capability on the pages that follow carries the actual measured number behind it, the date it was measured, and an honesty tier that says plainly whether it is shipped, still research, or only designed. Where a claim is partial, the caveat is printed next to the win — not in a footnote, not omitted. And the final chapter tells you exactly where on our website you can re-run the checks yourself, because the whole point of "AI you can prove" is that you shouldn't have to take a brochure's word for anything, including this one.

87% → 0.0%	**2×**	**1,026**	**21 ms × 377 MiB**
confabulation, driven to zero	faster at 128k context vs llama.cpp	certified-equivalent functions	a capable mind, one consumer GPU

The shape of the whole company in one strip — each figure is expanded, dated and caveated in its own chapter.

THE PLAIN TRUTH **How to read this brochure**

Three marks appear throughout: 🟢 **shipped** — measured, verified by our compiler gate, safe to buy on; 🔬 **research** — measured but partial, presented as work in progress; 🔭 **vision** — designed and honest about being unbuilt. Selling claims in this document are 🟢 only. If a bigger number would sound better but hasn't cleared the gate, it isn't here.

02 The four capabilities

What a human actually feels when using the stack — each with its one proof number.

Strip away the architecture and four things remain that you can feel from the outside. Each card below names the capability in plain words, then the single measured figure that backs it. All four are 🟢 shipped; the caveats are printed with them, because a number without its caveat is halfway to a lie.

① It can't bluff

Ask our knowledge system something it can source and it answers *with the citation*. Ask it something it can't and it **refuses** — it does not improvise a confident guess. Confabulation (the industry politely calls it "hallucination") was driven from 87% to 0.0% on our test corpus, and the no-bluff floor held under a sixteen-turn adversarial interrogation designed to talk it into guessing. The demonstration we show visitors: watch it refuse, then watch it cite the capital of Australia — Canberra, with provenance — from a knowledge store compressed to 0.70 GB.

① The proof — measured 2026-07-02

87% → 0.0%

confabulation on the test corpus

🟢 SHIPPED

16 turns

adversarial pressure the refusal floor held under

🟢 SHIPPED

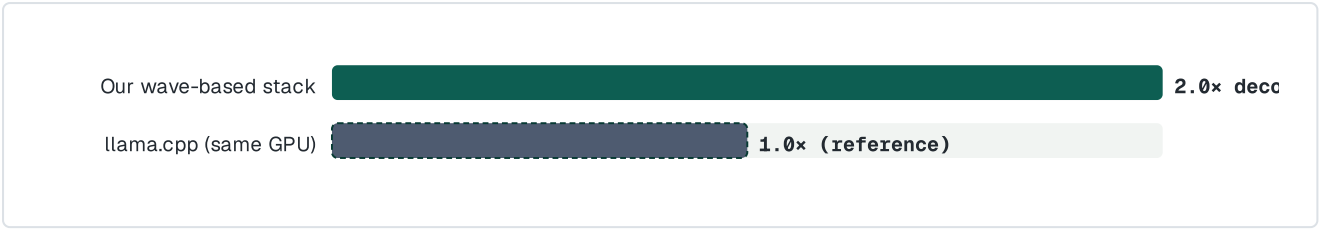
0.70 GB

the citeable store it answers from (was 20 GB)

🟢 SHIPPED

② It stays fast when the context is huge

Feed a conventional transformer a very long document and it slows down as the context grows. Our wave-based model architecture stays *flat*: at 128,000 tokens of context it decodes at twice the speed of llama.cpp on the same hardware — not because it is universally faster (it isn't, and we say so in Chapter 4), but because its cost doesn't climb with context length the way attention does.



Long-context decode, measured 2026-07-02 (🟢): ours stays flat at 128k where a transformer climbs. This is the long-context regime specifically — not a claim of being fastest everywhere.

Source: VOCABOTICS_DASHBOARD.md (Council 032) · relative decode speed at 128k tokens

③ Proven-correct code, any chip, AI-written

Everything in our stack is written in a small language built for machines to write and machines to check. Every single compile runs the program twice — compiled to native machine code, and independently in a typed interpreter — and compares the results value by value. The gate has never been disabled, and the corpus it guards holds **1,026 certified-equivalent functions** (captured 2026-07-10) — and not one more, because on unfiltered real-world C the converter honestly certifies only about one function in twenty and refuses the rest. The refusals are what make the certifications worth something.

③ The proof — captured 2026-07-10 from the live repo

1,026

functions certified equivalent — triple-gated

🟢 SHIPPED

4

backends gate-clean: x86-64 · ARM64 · GPU/PTX · portable C

🟢 SHIPPED

~1 in 20

honest yield on wild, unfiltered C — the rest refused

🟢 SHIPPED

④ A capable model on hardware you already own

The whole cognitive stack — model, memory, reasoning meters — runs on one consumer graphics card of the kind already sitting in a gaming PC: a step time of 21 milliseconds in 377 MiB of video memory on a single RTX 3090, with no data centre, no cloud dependency, and no per-token bill. On the same class of card, a real open-weights model runs full interactive generation on our own stack at 151.7 tokens per second — its output token-for-token identical to the reference implementation.

④ The proof — measured 2026-07-02

21 ms × 377 MiB

cognitive step × VRAM, one RTX 3090

🟢 SHIPPED

151.7 tok/s

real Qwen-0.5B, token-for-token == llama.cpp

🟢 SHIPPED

0.5B

the caveat: that model is small; 7B runs at parity, prefill still trails

🔬 RESEARCH

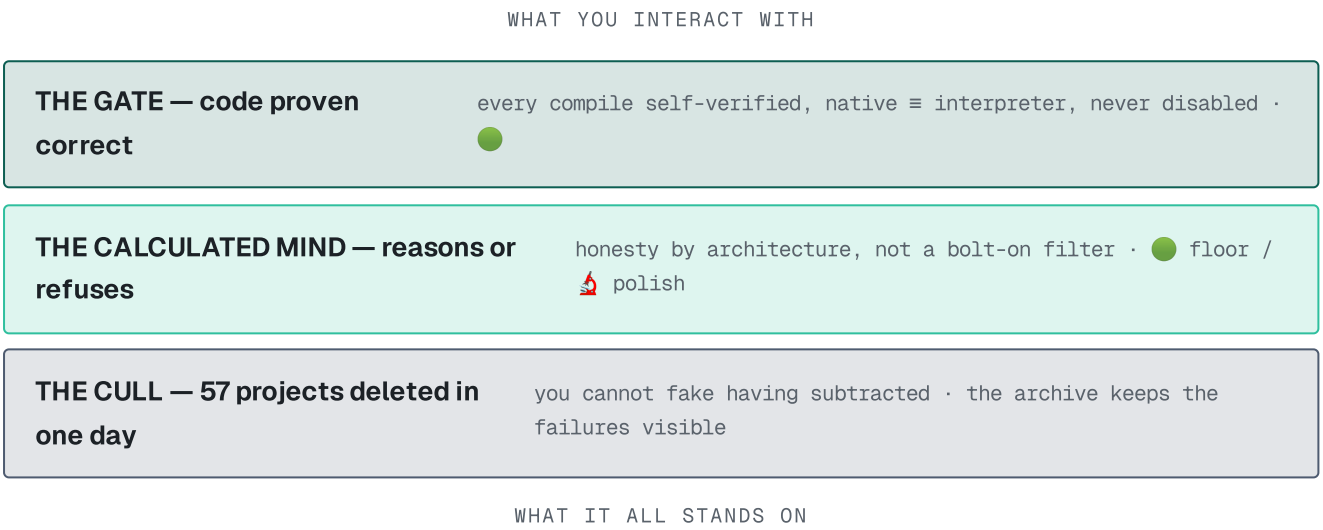
Hardware you already own: the whole measured stack above runs on one consumer RTX 3090-class card — no data centre.

Source: Photo: a GeForce RTX 3090 graphics card — PantheraLeo1359531, Wikimedia Commons, CC BY 4.0

03 Trust, certified at three layers

Not 'fastest'. A rival can copy a kernel — these three are the things you cannot fake.

We do not claim to be the fastest at everything — Chapter 4 shows a benchmark where we are explicitly behind and say so. The claim is narrower and harder: trust, certified at three distinct layers, each of which would take a competitor years to genuinely reproduce rather than merely imitate.



The three layers — mechanical, architectural, and cultural.

The gate is mechanical trust: nothing our AI writes is taken on faith — it is executed two independent ways and the results must agree, on every compile, with the check never switched off. **The calculated mind** is architectural trust: the honesty isn't a moderation layer bolted onto a model that wants to please you — the system computes meaning, cites what it can source, and refuses what it can't, by construction. **The cull** is cultural trust: in one day the founder deleted fifty-seven projects that didn't survive honest evaluation, and the public archive keeps the disproven work visible. A company that publishes its failures earns the right to be believed about its results.

> ***Best-in-world, for us, means the compounding axes held at once — long-context, quantised, any-chip, proven. Everyone else picks two.***
>
> The ratified dashboard, Council 032 · 2026-07-02

04 How it works — five capabilities, one being

The architecture in plain words. (The internal codenames live in the appendix, where they belong.)

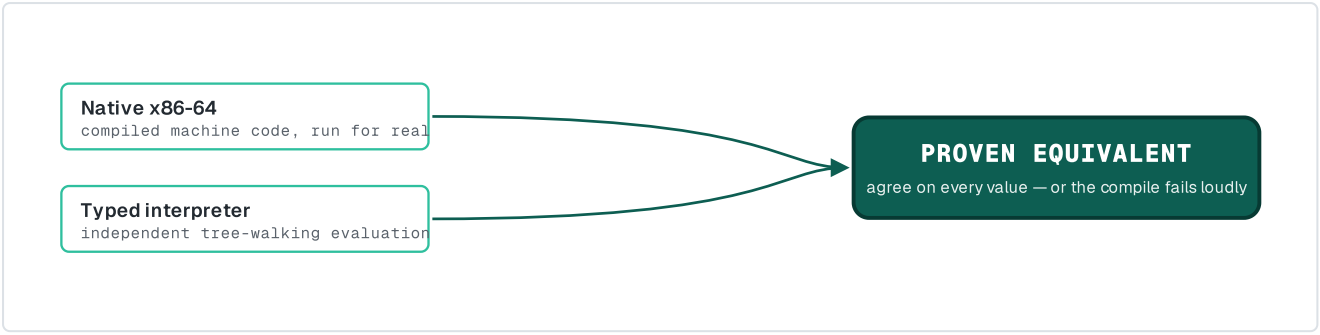
Under the four capabilities sits one connected architecture. We describe it here in plain language — what each part does and what has actually been measured — because a brochure that leads with internal codenames is selling you its own mythology. Two of the five parts are honestly marked 🔭 vision: designed, partially wired, and not products today.

🔨	🌊	🧠	🪟	🎛️
The language & its gate	**The wave model**	**The calculated mind**	**A world you own**	**New senses**
A machine-first language an AI writes and a compiler proves. 🟢 shipped	A model of waves, weighed in thirds {-1, 0, +1} — long context stays cheap. 🟢 shipped	Meaning calculated, not trained; only the voice is learned. 🟢 floor · 🔬 polish	A personal world, cryptographically yours. 🔭 vision — not a live product	Brain-computer and wave-native capture. 🔭 vision — measured research only

The five parts, and the tier each honestly sits at.

The language and its gate — 🟢 shipped

The foundation is a deliberately small programming language designed for an AI to write and a machine to verify. Its compiler *self-hosts*: the front end — a real lexer and recursive-descent parser — is written in the language and compiled by a code generator that is itself written in the language, and the bootstrap is byte-identical when built by two independent toolchains (gcc and clang), so no single compiler can hide a trick in the binary. Its cryptography is checked against published test vectors — SHA-2 and SHA-3, BLAKE3, ChaCha20, HMAC against RFC 4231, AES-128 against FIPS-197 — not against our own opinion of what the answers should be. And its optimiser earns its keep measurably: the solver-based register allocator produces code 4.2× faster and 3× smaller than the naive path, and the cost model reaches 0.8 TFLOP/s on measured kernels with no BLAS library underneath (all 2026-07-02, 🟢).



The gate, on every single compile (🟢 never disabled): both routes must agree, value for value, or the build fails.

The same triple-check ingests existing code: an ordinary C function is converted, then gated three ways — native ≡ interpreter, the original C compiled and run for real, and the two behaviours compared. When a genuine 32-bit wraparound function produced different answers in C and in our 64-bit model, the converter did not paper over it: the gate refused the equivalence and said why. That refusal is replayable, unedited, on our public demo page. Honest edges, kept visible: language coverage is a real subset (loops, arrays, floats, conditionals — not arbitrary programs yet), strings and I/O still lean on C, and the wild-corpus yield is that honest ~1 in 20 (🔬 / 🔭, 2026-07-10).

The wave model — 🟢 shipped, with named gaps

Instead of multiplying huge matrices of 16-bit numbers, our model architecture represents computation as waves and weighs values in thirds — {-1, 0, +1} — which collapses memory traffic and keeps long contexts cheap. What's measured (2026-07-02, 🟢): matrix multiply at 92.8% of NVIDIA's hand-tuned cuBLAS, bit-exact — parity, explicitly *not* fastest, and we print that rather than round it up; long-context decode 2× llama.cpp at 128k; sparse attention 59.7× at 65k; a real open-weights model running end-to-end on GPU code emitted by our own compiler — zero cuBLAS, zero cudart — generating text token-for-token identical to llama.cpp at 151.7 tokens per second; and training that works end-to-end in the same language, gradients verified against finite differences to 5.6×10^{-7}. The named gaps (🔬): 7B decode is at parity, not ahead; prefill is behind llama.cpp; paged KV-cache isn't done.

The wave model, measured 2026-07-02

92.8%

of cuBLAS GEMM, bit-exact — parity, not fastest

🟢 SHIPPED

59.7×

sparse attention speed-up at 65k tokens

🟢 SHIPPED

5.6e-7

gradient error vs finite differences — training verified

🟢 SHIPPED

The calculated mind — 🟢 the floor, 🔬 the polish

The part that answers you does not work like a chatbot guessing plausible next words. Meaning is *calculated* — sixty million gathered facts distilled to 22.2 million citeable ones, compressed from 20 GB to 0.70 GB at 28.6 bytes per fact, every answer traceable to its source — and only the voice that phrases the answer is learned. Two safety properties are proven at the architecture level: a provenance invariant (every word in an answer traces to a meaning, so it cannot decorate a citation with invention) and a polarity lock — "cannot" can never silently become "can" (both 🟢, 2026-07-02). The honest counterweight, printed at the same size: a cold-judge stranger currently scores the conversation 3 out of 10 — the honesty floor holds, the conversational polish does not yet — and the central research claim, that calculated meaning can beat trained prediction at language-model scale, is still **OPEN** (🔬). We keep that on the page on purpose.

The two vision organs — 🔭, plainly marked

Two parts of the architecture are designed but not products, and no figure in this brochure leans on either. **A world you own**: a personal, typed world with an AI companion that runs on your own machine and belongs to you cryptographically — about 75% wired from donor codebases, not live (🔭). **New senses**: brain-computer interfacing and waves-not-pixels capture — measured research experiments, not built systems (🔭). When either earns a 🟢 number, it will appear on the dashboard with its date, like everything else.

05 The honesty ledger

The three marks, the one law, and the receipts that make the system credible.

Every claim we publish — on the website, in this brochure, in a sales conversation — is tiered. The tier is part of the claim: strip it off and the sentence is no longer true. There are three marks, and one law that governs movement between them.

MARK	TIER	WHAT IT MEANS	WHAT IT'S WORTH
🟢	SHIPPED	Measured, gate-clean, dated. The number exists and survived verification.	Safe to buy on. Every selling claim in this brochure is this tier.
🔬	RESEARCH	Measured but partial — real numbers, incomplete coverage, named gaps.	Honest work-in-progress. Presented with its gaps, never as finished.
🔭	VISION	Designed and argued for, not built. No measured number exists yet.	A direction, plainly labelled. Never used to sell anything today.

The three tiers of the honesty ledger.

The one law: the bigger claim goes UP a tier — never down

When a result could be described two ways, the more impressive description must carry the *stricter* tier. A partially-proven capability may never borrow the 🟢 of its narrow proven core. This is the opposite of normal marketing gravity, and it is enforced at the source: the website and this brochure are cut from one ratified dashboard, so a figure cannot quietly improve in the retelling.

A ledger is only as credible as its debit column, so here are the receipts. Our public R&D archive scores every one of its projects with an honest verdict, and the **disproven projects stay published in full** — including an early ternary-7B result that was retracted rather than quietly dropped; the retraction is part of the record. The chapter census behind the compiler is published as an audit, not a boast: 570 chapters independently audited, roughly 59% confirmed genuinely real after 95 repairs, and 60 stubs quarantined rather than counted (2026-07-10). And the corpus number on the cover of this brochure — 1,026 — is one function *lower* than the previous public figure, because a false equivalence was found and quarantined. We printed the smaller number.

How a claim earns its way up the ledger — and what sends it back.

06 Safety by construction — and where we learned it

A memory leak is now a compile error. Fifteen years of rail taught us why it has to be.

Most AI safety is a policy document. Ours is a compiler flag. Because everything in our stack is written in a language the machine can analyse completely, safety rules can be enforced *at compile time*: run the compiler with a safety profile and it statically rejects code that allocates memory after initialisation, loops without a provable bound, recurses, exceeds function-length limits, skimps on assertion density, or ignores a checked return — the six checkers of the strictest profile, modelled on NASA's Power-of-Ten rules — and issues a certificate on code that passes (🟢 built and demonstrated, 2026-07-02). A memory leak is not a bug to find in review; it is a compile error. Profiles form a lattice — the NASA profile sits above SIL-4, which sits above DO-178C — and a worst-case-execution-time certificate is the named next step (📡).

Certifiable by construction — demonstrated 2026-07-02

6

static safety checkers in the strictest profile

🟢 SHIPPED

compile error

what a memory leak now is — not a review finding

🟢 SHIPPED

WCET cert

worst-case timing certificates — designed, next

📡 VISION

Why a stability gate matters next

A system that learns after deployment needs more than a static gate. The designed answer — 🔬 unproven, and marked as such — is a stability gate: a mathematical bound (Lyapunov-style) on how far any plastic update can move the system. We mention it because pretending adaptive systems are covered by static checks would be exactly the kind of overclaim this brochure exists to avoid.

Fifteen years where wrong is a safety risk

This discipline wasn't adopted for marketing. Before vocabotics, the founder spent fifteen years delivering safety-critical systems for global rail and metro — including a $4m+ metro public-address and passenger-information programme — the class of system where a wrong announcement during an emergency is not a bad user experience but a safety incident. That is the world SIL levels and DO-178 certification come from, and it is the standard we hold AI to when it touches anything that can hurt people: *certified, or clearly labelled as not yet.*

A passenger information display in daily service — the class of safety-critical system our rail heritage covers. (Illustrative photograph of a metro system, not a client deliverable — we name no clients.)

Source: Photo: LED passenger information display on a metro train — ChenSimon, Wikimedia Commons, CC BY-SA 4.0

A signal at danger. Rail's first lesson, carried into everything we build: when being wrong is dangerous, the system must refuse — visibly — rather than guess.

Source: Photo: a railway signal at dusk — W.carter, Wikimedia Commons, CC0

07 What we sell today

Four things you can buy this quarter — each honestly labelled with its own maturity.

The research above is the engine; this page is the shop. Four offerings are live today, and one build service carries a "prototype" label because its underlying systems do — the same tiers, applied to our own price list. No offering below depends on a 🔬 or 🔭 claim.

OFFERING	WHAT YOU GET	STATUS	FROM
AI Readiness Scorecard	Five minutes, three concrete next steps — no sign-up wall on the basics.	🟢 shipped	Free
Strategy Audit	A focused review of where AI genuinely helps your business — and where it doesn't yet. A plain-English plan, not a sales pitch.	🟢 shipped	£2,500
Compliance-grade AI Audit	For regulated firms: an audit trail an assessor will accept — adopt AI without risking your ISO 9001 certification.	🟢 shipped	Enquire
Training & Workshops	Calm, practical sessions for teams and communities, including AI safety for the people you serve.	🟢 shipped	Per engagement
No-Bluff Build	We build the system: cite-or-refuse knowledge tools, agentic workflows with a STOP button, deterministic where wrong is expensive.	🔬 prototype — labelled as such	£25,000

The current offer — statuses as published on vocabotics.com, July 2026.

Prices are the published starting points as of July 2026 — the current figures are always at vocabotics.com/consulting.

Behind the build service sit the productised systems, each carrying its own status: a **no-bluff knowledge engine** that answers from your documents with exact citations or refuses (prototype — the 0.0% confabulation figure from Chapter 2 is its measured heart); **agentic systems** with a deterministic planner and a STOP button that shipped before the fleet did (prototype, 67 execution-proven agents); and a **spec-to-application compiler** that turns a ~240-line wiring specification into a working ~50k-line app, byte-identical across 9 target frameworks under 46 byte-identity tests (🟢 shipped, 2026-07-10).

The honest sales pitch

Most businesses should start with the free scorecard, and many should stop there for now. We would rather tell you AI doesn't yet pay for itself in your workflow than sell you a build that quietly fails — a company whose brand is refusal has to be able to refuse your money too.

08 How to verify us

The close of this brochure is not a slogan. It's a checklist you can run.

Everything claimed in these pages is checkable on the public website, most of it in under fifteen minutes. This is the no-dead-end rule applied to our own brochure: every claim ends somewhere you can go.

The fifteen-minute verification

Run these four in order. If any of them disappoints you, you will have learned something true about us — which is the point.

- [ ] vocabotics.com/proof — watch the recorded gate transcripts: a real compile proving native ≡ interpreter, and a real refusal where a 32-bit wraparound made the gate say DIFFER instead of papering over it.
  Every number on that page was captured from the live repo on 2026-07-10, raw terminal output one click away.
- [ ] vocabotics.com/proof/ledger — the census: 570 chapters audited, the repaired, the genuine, and the 60 quarantined stubs, all counted in public.
- [ ] vocabotics.com/dashboard — the flagship figures with tier and caveat attached, plus verdict counts recomputed from the archive on every build — including the failures.
- [ ] vocabotics.com/lab — the dated R&D notebook. Read a disproven report in full, then decide what a 🟢 from us is worth.

> ***Filed is not shipped. Measured is not marketed. If the data changes, the pages change with it — this brochure included.***
>
> The dashboard's standing rule

And if what you need is a conversation rather than a transcript: the scorecard at **vocabotics.com/diagnose/quiz** takes five minutes and ends with three concrete next steps, whether or not any of them involve us.

09 Appendix: the codenames, once

Capabilities sell; codenames stay on the deep page. This is the deep page.

Throughout this brochure the architecture was described in plain words, by design — a reader deciding whether to trust us should never need our internal vocabulary. For the reader who goes on to the research lanes and the lab notebook, where the projects appear under their working names, this table is the bridge. It appears once, here, and nowhere else in the document.

IN THIS BROCHURE	CODENAME	WHERE TO READ DEEPER
The language & its gate	QUANTA	vocabotics.com/research/quanta · /research/the-gate
The wave model	WaveTernary	vocabotics.com/research/wave-ternary-models
The calculated mind	The Calculated Brain	vocabotics.com/research/calculated-brain
A world you own	NovaTerra	vocabotics.com/research/novaterra
New senses	Devices	vocabotics.com/lab (hardware strand)

Plain name → working codename, as used on the deep pages.

One operating note belongs here too: the company itself runs on the architecture it sells. A council of stronger models decides and ratifies; builder agents write the hard code; a fleet handles bulk work in parallel; and nothing any agent produces is trusted — it is verified by the same gate this brochure has been describing. "Propose broadly, dispose certainly" — the two-engine thesis, run as a company.

10 Image credits & attributions

Every photograph in this brochure is real, free-licensed and credited. The cover is drawn in code — our own gate, plotted.

Sourced via the Wikimedia Commons API and used under the licences below; full source URLs are in the sources list. The metro photograph is illustrative of the system class only — consistent with our no-client-names rule, it is not a project we delivered.

WHERE IT APPEARS	SUBJECT	PHOTOGRAPHER / SOURCE	LICENCE
The four capabilities	A consumer GeForce RTX 3090 graphics card	PantheraLeo1359531 — Wikimedia Commons	CC BY 4.0
Safety & heritage	An LED passenger information display on a metro train	ChenSimon — Wikimedia Commons	CC BY-SA 4.0
Safety & heritage	A railway signal at dusk, showing a red aspect	W.carter — Wikimedia Commons	CC0 (public domain dedication)

Photograph credits.

11 Sources, glossary & the ledger

Where every claim comes from — and what the plain words mean.

Sources

01 **VOCABOTICS — THE DASHBOARD (ratified by Council 032; the honesty-tiered source this brochure is cut from)** — vocabotics — internal, measured against the gate (as of 2026-07-02)

02 **The Demo — AI you can prove (recorded gate transcripts, incl. the honest DIFFER; 1,026-function corpus figure)** — vocabotics (as of 2026-07-10)
https://vocabotics.com/proof

03 **The Ledger — 570 chapters audited (the honesty census: repaired, genuine, quarantined)** — vocabotics (as of 2026-07-10)
https://vocabotics.com/proof/ledger

04 **The Dashboard — the R&D archive by the numbers (flagship figures with tiers and caveats)** — vocabotics (as of 2026-07-10)
https://vocabotics.com/dashboard

05 **The Lab Notebook — the dated R&D archive, failures kept visible** — vocabotics (as of 2026-07-10)
https://vocabotics.com/lab

06 **Services & systems — the published offer and statuses (content/data/services.json, systems.json)** — vocabotics (as of 2026-07-11)
https://vocabotics.com/consulting

07 **File: Gigabyte GeForce RTX 3090 Eagle OC 24G … Front 20201114 DSC5880.jpg** — Wikimedia Commons — PantheraLeo1359531, CC BY 4.0 (as of July 2026)
https://commons.wikimedia.org/wiki/File:Gigabyte_GeForce_RTX_3090_Eagle_OC_24G,_24576_MiB_GDDR6X_Front_20201114_DSC5880.jpg

08 **File: LED Passenger Information Display, Guangzhou Metro Bombardier Train.jpg** — Wikimedia Commons — ChenSimon, CC BY-SA 4.0 (as of July 2026)
https://commons.wikimedia.org/wiki/File:LED_Passenger_Information_Display,_Guangzhou_Metro_Bombardier_Train.jpg

09 **File: Railway signal at dusk.jpg** — Wikimedia Commons — W.carter, CC0 (as of July 2026)
https://commons.wikimedia.org/wiki/File:Railway_signal_at_dusk.jpg

Plain-English glossary

The gate	Our compiler's standing self-check: every program is run both as compiled native code and in an independent typed interpreter, and the results must agree value for value. It has never been disabled.
Honesty tier	The mark every published claim carries: 🟢 shipped (measured, gate-clean), 🔬 research (measured, partial), 🚀 vision (designed, honest). The bigger claim always goes up a tier, never down.
Confabulation	An AI's fluent, confident, false answer — often called 'hallucination'. Our knowledge system's measured rate is 0.0% because it is built to cite or refuse, not to guess.
Certified equivalent	A function that passed the triple gate: our native code ≡ our interpreter, the original source compiled and run for real, and both behaviours matching. 1,026 functions hold this certificate; refusals are published too.
Certifiable by construction	Safety rules enforced by the compiler itself — a safety profile statically rejects violating code and issues a certificate on clean code, so a memory leak is a compile error rather than a review finding.
The cull	The one-day deletion of 57 projects that failed honest evaluation, with the failures kept publicly visible. Subtraction you can verify is the costliest signal we own.

vocabotics

THE HONESTY LEDGER

🟢 Shipped — measured, gate-clean.

🔬 Research — measured, partial; the gap is shown.

🔭 Vision — designed, honest; not yet built.

The bigger claim goes up a tier, never down. We show the eval beside the demo — the tiering is the credibility.

Run the fifteen-minute verification

Don't take this brochure's word for it — watch the recorded gate transcripts, including the one where it honestly refuses, then read the ledger and the lab.

/proof

Nothing faked.

Every figure in this brochure carries its measurement date and its honesty tier; selling claims are 🟢 measured only, research and vision are marked as exactly that, and every number can be re-checked on /proof, /dashboard and /lab. If a claim ever fails re-audit, it is quarantined in public — this brochure is regenerated from the same ledger, so it changes too.