



Mainely Code

AI anywhere. Governed everywhere.

THE CONNECTED MAINELY CODE ECOSYSTEM

THE GREAT NORTH

Mainely Code
Providence Architecture

Powerful intelligence.
Distinct responsibilities.
Evidence at every handoff.

OUR STANDARD

Specialized products.
Governed execution.
Proof end to end.

SEPTEMBER 2026

Public architecture vision brief
Version 4.0 · Public edition

THE GREAT NORTH RUNS ON PROVIDENCE

A bigger world. A clear purpose.

Mainely Code is building a connected family of AI products—not one application pretending to do every job. The ambition is broad. The responsibilities stay precise.

OUR MISSION

“Make state-of-the-art AI accessible, usable, and approachable for Everyone.”

01 The Great North is the ecosystem.

The connected product and platform world: cognition, cloud experiences, design, engineering, growth operations, model development, inference, trusted compute, access, and proof.

02 Providence is the architecture.

The rules and contracts connecting those systems: who directs, what is permitted, where execution may happen, and how an outcome is judged.

03 Great North Provider is the inference system.

The first-party execution platform being developed within the ecosystem. It supplies model execution and evidence; it does not replace the other products or their authority.

Access and accountability belong together. People should be able to start with useful local tools, understand the work, and retain control as their needs grow into more capable products and infrastructure.

03–04	Products & flagship cognition	05–08	Architecture & proof
09–10	Access & growth	11–12	FORGE & model development
13–16	Deployment, release direction & source basis		

Source basis: 1–5, 9. References and release interpretation on page 16.

SPECIALIZED PRODUCTS. CONNECTED PURPOSE.

Different jobs. One coherent family.

Shared architecture connects expertise without erasing the customer experience, product identity, or standard of success.

FLAGSHIP COGNITIVE PLATFORM

Mainly Code Northstar

Coordinates specialist reasoning and external AI platforms; decomposes complex missions; arbitrates competing analyses; preserves provenance and uncertainty; and owns final cognitive judgment.

[The flagship, in depth →](#)

CLOUD EXPERIENCE

Project Sasquatch

Cloud conversations, planning, research support, creation, account controls, and product-aware routing.

Owns the cloud customer journey.

DESIGN & INTERFACES

Orb-Weaver

The local-first UI/UX design loom: reviewable maps, flows, components, design tokens, previews, and editable interface foundations.

Draw the experience. Weave the interface.

MODEL DEVELOPMENT PLANNED

Mainly Code's FORGE

Planned local-first model engineering and AI research: prepare, modify, train, evaluate, and export. For developers, researchers, and small teams.

The planned product, in depth →

SOFTWARE ENGINEERING

Mainly Code Buildroom

Local-first engineering with bounded work orders, candidate diffs, verification, proof bundles, rollback, and verified PR or apply proposals.

AI coding with receipts.

MARKETING & SALES

Mainly Code Growthroom

A local-first, evidence-backed workspace in development for audience research, campaigns, leads, follow-up, and measurable next actions.

Audience. Content. Conversion.

SHARED PLATFORM

Distinct responsibilities.

Provider executes inference. Prefrontal governs permission. Network Fabric establishes trusted placement. Licensing Server governs access.

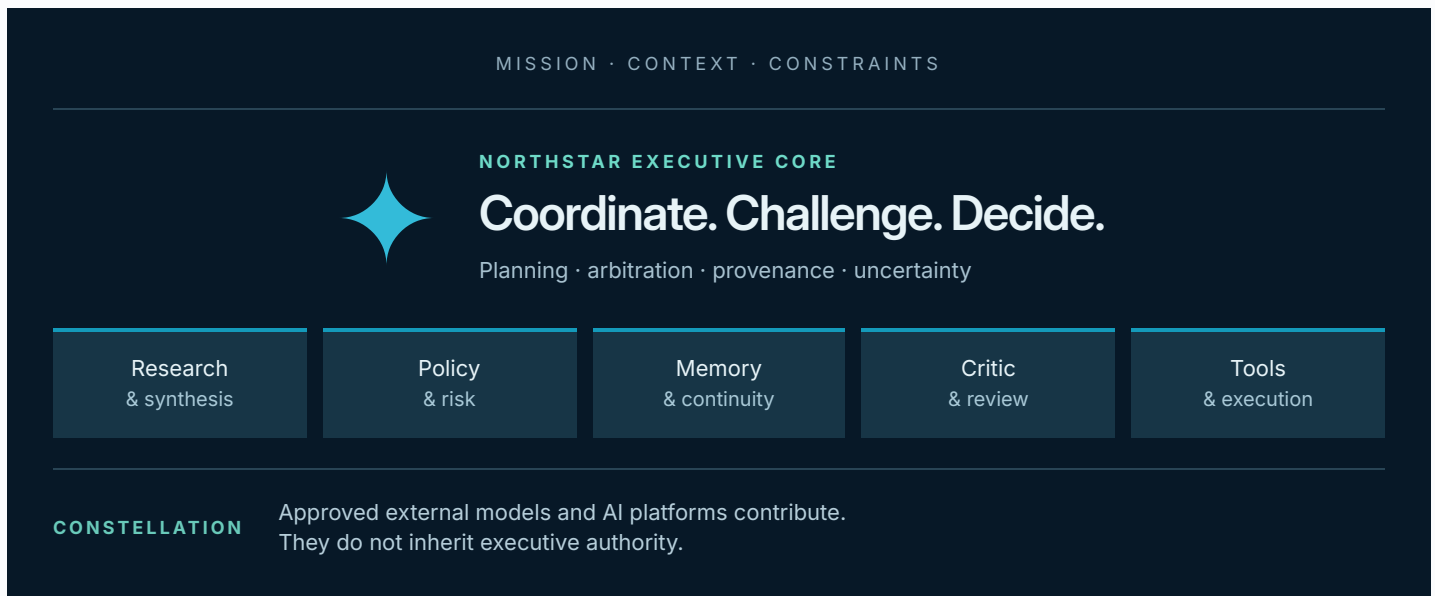
Product direction—not a claim that every product or cross-product integration is available.

Source basis: 1, 2, 4–9. References and release interpretation on page 16.

MAINLY CODE NORTHSTAR

More than a command center. A cognitive platform.

Northstar is the flagship cognitive system of The Great North. Its role is to turn a complex mission into a coordinated, inspectable reasoning and execution path—not merely select a model or display a dashboard.



One mission. Multiple specialists.

Northstar breaks a request into tractable work, coordinates relevant contributors, resolves disagreement, and decides when a route should continue, change, or stop. Specialist output remains a contribution—not an instruction with unlimited power.

Judgment with an audit trail.

The platform is designed to retain source origin, route decisions, budgets, uncertainty, and contribution history. The operator should be able to inspect why a path was chosen and what evidence supports the result.

Cognitive authority is not execution permission.

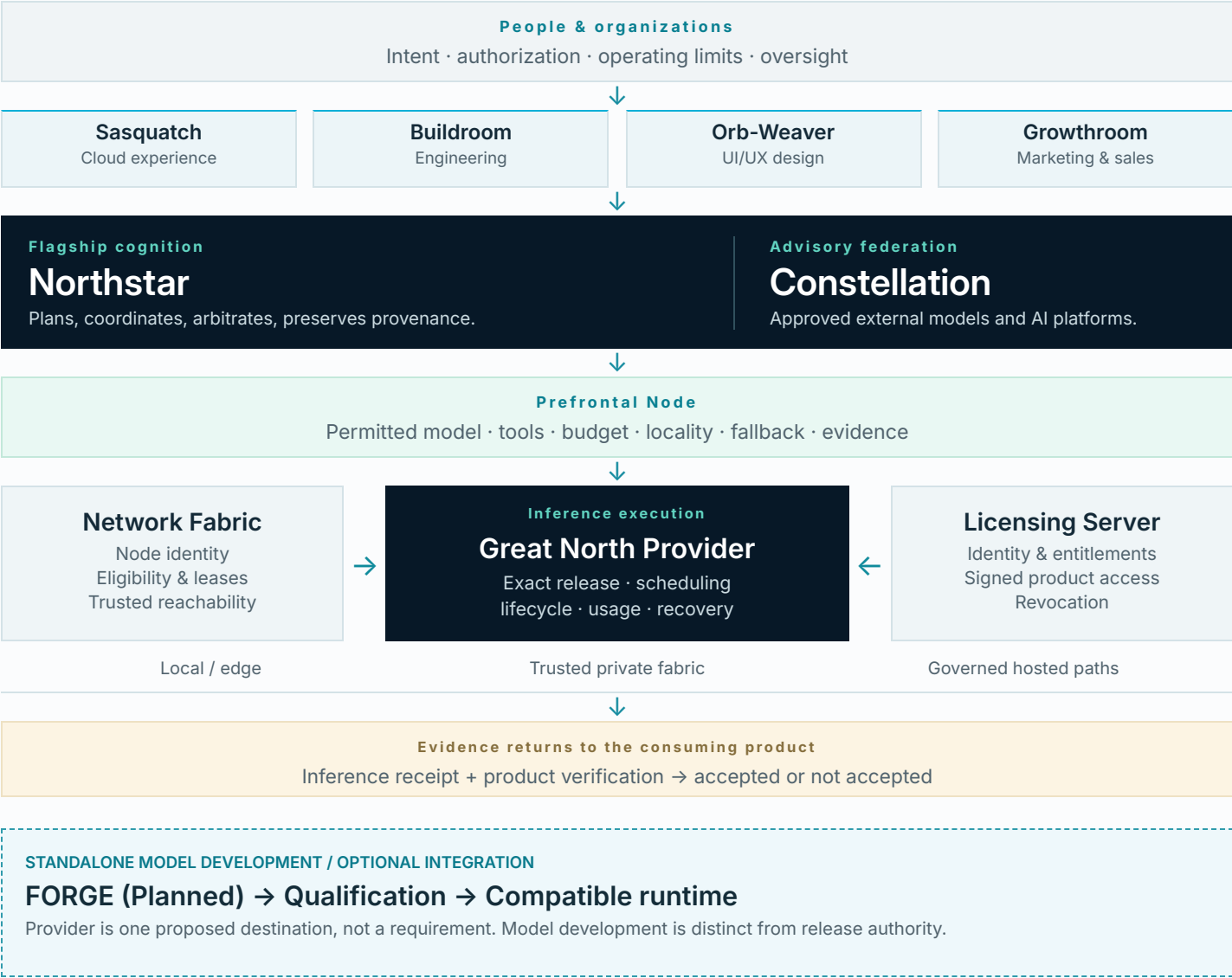
Northstar directs the plan. The Prefrontal Node constrains what may run. Provider operates the model. The consuming product decides whether its own outcome is accepted.

Designed first for institutional work across business, education, and public-sector settings. Specialist labels illustrate the cognitive architecture; they are not a shipping feature checklist.

PROVIDENCE AT A GLANCE

Power moves through explicit authority.

The target integrated architecture separates customer intent, cognitive direction, permission, execution, infrastructure trust, access, and product acceptance.



Connected does not mean mandatory cloud dependence.

This map shows logical authority, not network wiring or a requirement to use every service. Standalone products retain their release-specific runtimes and local behavior.

The FORGE lane is proposed; the wider map remains an integrated vision, not a certification of every handoff.

Source basis: 1, 2, 4, 5, 9. References and release interpretation on page 16.

ONE CONCERN. ONE AUTHORITY.

Capability is not permission.

Providence is designed to let powerful systems work together while keeping their authority separate. Each role has a job—and a boundary it cannot silently cross.

AUTHORITY	RESPONSIBLE FOR	DOES NOT DECIDE
Northstar Directs	Goals, decomposition, specialist arbitration, executive budgets, and final cognitive judgment.	Product acceptance, runtime lifecycle, or customer entitlement.
Prefrontal Node Permits	Policy-bound plans, tool and memory limits, locality, fallback, evidence, and cancellation constraints.	Billing truth, node identity, or product apply and merge.
Great North Provider Executes	Exact model execution, admission, scheduling, lifecycle, resource use, recovery, and receipts.	Its own permission, network trust, or successful product completion.
Network Fabric Establishes where	Node identity, reachability, eligibility, cross-machine leases, and network evidence.	Model promotion, customer access, or result quality.
Licensing Server Establishes who	Canonical identity, entitlements, signed product access, assignment, and revocation.	Cognitive routes, tool safety, or execution success.
Consuming product Accepts its outcome	Its customer workflow, validation, evidence, and final product-specific result.	Another product's entitlement or the other authorities' decisions.

Autonomy can be explicit.

A user or organization may authorize more automatic work within the product's supported controls. Fewer approval interruptions must not remove scope, budget, evidence, cancellation, or acceptance requirements.

A refusal remains binding.

A model response, worker, or scheduler must not bypass a denial from the authority that owns the decision. Missing permission should produce a visible blocker and a safe next action—not a hidden workaround.

The system that runs the model
does not get to declare the work done.

GREAT NORTH PROVIDER

Not just an endpoint. An execution system.

Great North Provider is being developed to own governed model execution: release identity, scheduling, runtime lifecycle, resource use, recovery, and inference evidence.

THE DIRECTION

Exact model.
Explicit plan.
Verifiable receipt.

First-party execution is the objective—not a claim that every supported model or hardware path already runs through a certified Mainly-native runtime.

01

Release capsule

The deployable model unit: weights, tokenizer, template, configuration, runtime compatibility, lineage, license state, and approved hardware profile.

ModelReleaseCapsule.v1

02

Inference plan

The permitted execution: model or capability, budget, deadline, locality, fallback, retention, structured output, and evidence requirements.

InferencePlan.v1

03

Inference receipt

The execution record: what ran, under which plan, where, with what usage and timing, and whether retries or fallback occurred.

InferenceReceipt.v1

LIFECYCLE IS PART OF THE PRODUCT

Load → Warm → Execute → Cancel / complete → Drain → Unload

Native where proven.

The intended certified native path should operate without a required external provider daemon. Compatible runtimes and external model routes can remain distinct options; compatibility is not proof of native execution.

Active development. Hardware, performance, and deployment claims require a named, qualified release path.

Accountable under failure.

Queue handling, cancellation, restart, uncertain outcomes, quarantine, and rollback belong in the runtime contract. A successful response alone is not sufficient evidence of a healthy service.

THE HANDOFF IS PART OF THE WORK

A receipt proves execution. Verification proves the outcome.

Providence connects these questions without treating a model response, a saved artifact, or an isolated passing test as the entire customer result.

01 / INTENT Define the customer job and acceptance criteria.	02 / PLAN Bind the permitted route, scope, resources, and limits.	03 / EXECUTE Run the exact approved model and record what happened.	04 / VERIFY Evaluate the result under the consuming product's contract.
--	---	---	---

INFERENCE EVIDENCE

What actually ran?

The receipt binds the request to its plan, model release, runtime, placement, timing, usage, and relevant failure or fallback behavior.

It provides traceability—not proof that code works, a design is correct, or a campaign achieved its goal.

PRODUCT EVIDENCE

Did it do the job?

The product evaluates its own success standard and retains the relevant artifacts, checks, decision, and recovery path.

Accepted, failed, blocked, cancelled, and unverified outcomes must remain distinct. Confident prose is not evidence.

ILLUSTRATIVE HANDOFF / DESIGN TO ENGINEERING

From an interface idea to a verified change.

Sasquatch can clarify the request. Orb-Weaver can produce a reviewable interface foundation. Northstar coordinates the plan. Buildroom creates a bounded work order, candidate diff, and verification record.

A working-interface task needs relevant behavior checks—not just generated source. Provider's receipt joins the evidence; Buildroom owns engineering acceptance and the apply or pull-request proposal.

Illustrative integrated direction—not every handoff ships in every edition. Buildroom is the intended first qualification workload for Provider.

MAINLY CODE BUILDROOM—FREE EDITION

A real place to start. Not a waiting room.

We want Mainly Code to earn its name through tools people use, value, and recommend. FREE EDITION is the front door: useful local AI without a paid Buildroom edition.

ANNOUNCED EXPANSION / LOCAL MODEL OPTIONS

3_B4_B7_B8_B

FREE means \$0.

No token purchases or monthly token quotas for local use.

Learn and understand.

Explore unfamiliar code, ask for plain-English explanations, work through errors and logs, and turn an unclear problem into a practical next step.

Make something useful.

Draft a small script, plan a project, or start a simple page or component. Review generated work and test anything you intend to run.

The expansion is announced—not all live yet.

The current Store release is presented as local 4B coding help. The 3B, 4B, 7B, and 8B expansion is the next step. Exact models and capabilities belong to the installed release and its release notes.

Accessible is practical.

Initial setup and model downloads require connectivity. Local AI conversations can then run on compatible Windows hardware. Downloads, updates, and explicitly online work still need a connection.

Your hardware, storage, and electricity remain your own resources. Free local use is not a promise of unlimited compute, perfect output, or all paid-edition capabilities.

Hardware still matters.

More model options do not mean every model fits every machine. Supported profiles must account for memory, processor, graphics hardware, context, and workload—not just a parameter count.

MAINELY CODE GROWTHROOM

Build the product. Then build its audience.

Growthroom brings a new customer job into The Great North: local-first, evidence-backed marketing and sales. Its announced direction connects strategy, brand voice, campaign execution, and measurable outcomes.

AUDIENCE. CONTENT. CONVERSION.

Turn real work
into coordinated growth.

Product releases, project records, screenshots, videos, and other evidence can support campaigns—not invented claims.

01 Evidence	02 Campaign	03 Review
04 Approval	05 Publication	06 Engagement
07 Leads	08 Follow-up	09 Analytics

More than content generation.

The published vision spans audience research, campaign development, conversations, conversion paths, approvals, and next actions inside a private workspace supported by local AI.

A distinct product boundary.

Growthroom’s work is marketing and sales. Buildroom remains the engineering product; Orb-Weaver remains the design loom; Northstar remains the flagship cognitive platform.

Evidence before claims. Approval before consequence.

Growthroom’s announced workflow ties factual claims to their sources, consequential actions to policy and approval, and publication to a receipt. Publication evidence is not the same as proof of a sale or a growth result.

Active development. Publicly shown as a running desktop application progressing toward release candidate. Store availability and integration contracts remain release-specific.

MAINLY CODE'S FORGE / PLANNED PRODUCT

Build the AI. Not just with it.

A planned local-first model engineering and AI research workbench for developers, independent researchers, and small technical teams.

A guided path. An open research bench.

Bring data preparation, model development, training, experimentation, evaluation, and export into one coherent workspace.

Develop. Modify. Train.

Explore supported models, specialization, and fine-tuning with understandable starting points.

Go beyond the recipe.

Investigate custom components, training approaches, and novel AI technologies with code-level access.

Understand the result.

Compare against a baseline and carry findings, lineage, and limitations into the export.



THE INTENDED WORKFLOW

Define → Prepare → Baseline → Experiment → Compare → Package

Buildroom helps developers build software with AI.
FORGE is intended to help developers build the AI.

PLANNED PRODUCT

Development starting soon.

High-end PCs and GPU workstations are the initial audience. Supported operating systems, hardware, models, training methods, export formats, pricing, licensing, and release timing will follow validation—not blanket compatibility promises.

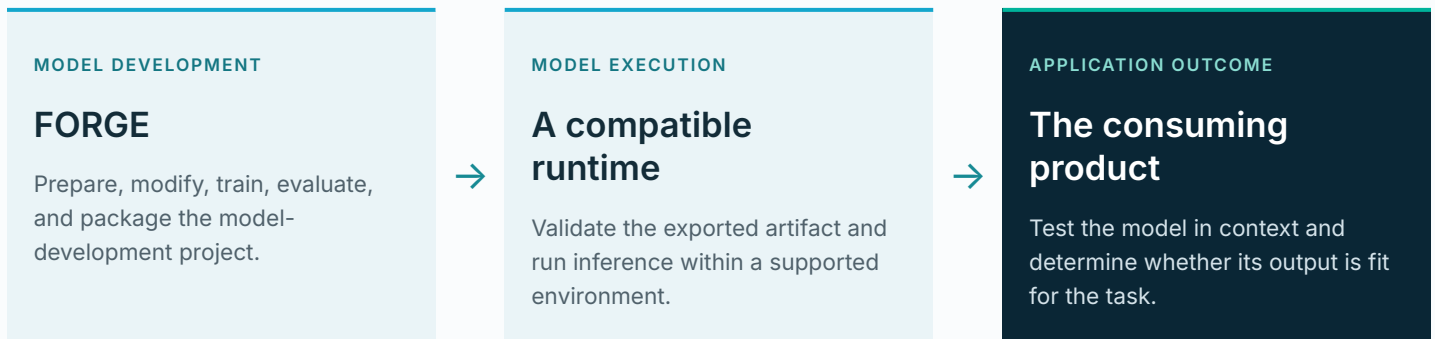
MODEL DEVELOPMENT MEETS GOVERNED EXECUTION

Standalone by design. Connected by choice.

FORGE is its own planned product—not a Buildroom edition or a renamed inference runtime. Other Mainly Code products are not intended to be required to use its exported models.

The goal: train and export a supported model for use in a compatible runtime of the developer's choice.

CONCEPTUAL HANDOFF · NOT A LIVE INTEGRATION



Exports would carry relevant configuration, lineage, evaluation findings, and applicable terms. A successful export is not proof of runtime compatibility, model improvement, or a successful deployment.

Provider is an optional destination.

Provider integration would require qualification, release admission, permitted execution, and evidence. It is one potential destination—not a requirement for FORGE exports.

Northstar remains the flagship.

Planning, specialist coordination, arbitration, and provenance stay with Northstar. FORGE develops models; it is not a competing executive layer.

Local control comes first.

Data and artifacts stay user-controlled by intent. Optional private or remote compute may follow with explicit destination, authorization, and cost visibility.

Permission and proof stay separate.

In Providence, Prefrontal permits, Fabric establishes trusted placement, Licensing governs access, and the consuming product accepts its outcome.

"Training finished" should never be mistaken for "the model improved."

Planned direction; formats, runtimes, and integration contracts remain to be validated. Read the FORGE Public Product Vision →

AI ANYWHERE. GOVERNED EVERYWHERE.

Place the work deliberately. Keep the contract intact.

Providence is designed to preserve authority and evidence across local hardware, trusted private infrastructure, and governed hosted environments. Location changes the operating conditions—not who is allowed to decide.

01

Local / edge

Local-first products on customer-controlled hardware, including qualified offline operation. Available resources and the installed runtime determine practical capacity.

CUSTOMER VALUE

Control close to the work.

02

Private fabric

Customer, office, or lab compute connected through trusted identity, private reachability, eligibility, leases, and infrastructure evidence.

CUSTOMER VALUE

Use trusted capacity.

03

Hosted paths

Regional, dedicated, or customer-owned options designed around isolation, residency, audit, and operational requirements.

CUSTOMER VALUE

Scale within clear limits.

Fabric establishes eligibility.

A reachable worker is not automatically trusted. Network Fabric owns identity, network state, placement eligibility, and cross-machine leases. Provider consumes that authority rather than inventing it.

Access stays product-specific.

The Licensing Server governs shared identity and entitlements. Access to one product must not silently unlock another. Each product enforces its own capabilities from the applicable access contract.

No silent relocation. No silent fallback.

A cheaper region, available model, or idle worker cannot override the plan. Changed execution conditions require authorization and visible evidence.

FREE EDITION does not require a paid subscription. Offline behavior, Store licensing, shared services, and remote execution remain product-, edition-, and deployment-specific.

Hosted and sovereign paths are objectives, not blanket certifications. Availability requires named release evidence.

Source basis: 1–3. References and release interpretation on page 16.

TRUST IS ALSO WHAT HAPPENS WHEN THINGS BREAK

Design the difficult moments first.

An accountable system needs more than a happy path. Providence treats interruption, uncertainty, recovery, privacy, and unsupported conditions as part of the product contract.

THE MODEL IS UNAVAILABLE**Stop or reauthorize. Do not substitute silently.**

Keep the requested model and execution binding visible. Offer a permitted alternative or a clear blocker instead of quietly changing the conditions of the work.

EXECUTION IS INTERRUPTED**Recover from evidence—not optimistic memory.**

Preserve what is known about execution and mutation. Reconcile uncertain outcomes before retrying so recovery does not create duplicate work or erase a valid candidate.

THE CHECKS DO NOT PASS**Report the failure. Preserve the next step.**

Retain the candidate, verification result, failure classification, and available rollback or resume path. A generated answer or saved artifact is not an accepted result.

PERMISSION OR BUDGET IS EXHAUSTED**Respect the boundary before starting more work.**

Scope, tools, cost, time, retries, and locality belong in the permitted plan. Any further action needs a valid path through the relevant authority.

Privacy is explicit.

Data movement, retention, and use must follow the product's declared controls. Evidence should not become uncontrolled source uploads, secret exposure, or unapproved customer training material.

Claims are specific.

Hardware and performance statements belong to a named model, runtime, context, workload, and deployment. A benchmark on one machine is not a guarantee for every customer.

An honest blocked state is more useful than an unearned success claim.

RELEASE POSTURE / 05 SEPTEMBER 2026

Ambitious direction. Clear release milestones.

Architecture, announced upgrades, public beta, Store availability, and certified deployment are different claims. The product family advances in stages.

CURRENT + NEXT UPDATE

FREE EDITION

Available now; model expansion announced.

The current Store release is presented as local 4B coding help. The announced expansion adds 3B, 4B, 7B, and 8B local model choices.

SEPTEMBER 2026

LITE EDITION

Target: before the end of September.

Planned for the Microsoft Store, with further upgrades following the FREE expansion. Exact models, features, requirements, and pricing will accompany the release.

LATE FALL / EARLY WINTER

ADVANCED EDITION

Target: before the end of 2026.

The higher-capacity engineering edition remains in development. Its launch must identify supported hardware, actual capabilities, and release-specific limitations.

Across the wider family

Northstar

Active development; locally verified work is distinct from live integration paths.

Orb-Weaver

Microsoft Store product; consult the current listing and release details.

Project Sasquatch

Paid public beta; its cloud release does not establish full ecosystem integration.

Growthroom & Provider

Active development; availability and qualified paths require their own release evidence.

NEW PLANNED PRODUCT

Mainely Code's
FORGE

Planned local-first model engineering and AI research workbench. Development starting soon. Release timing, pricing, hardware, and supported workflows remain to be announced.

Release targets are not present availability. Status reflects the September 2026 release direction. Product release notes and current Store listings control supported capabilities.

THE GREAT NORTH / MAINELY CODE

AI anywhere. *Governed everywhere.*

Make powerful technology easier to use.
Make authority easier to understand.
Make outcomes easier to verify.

MAINELYCODE.COM ↗

HELLO@MAINELYCODE.COM

About this edition

Version 4.0 · 5 September 2026 · FORGE source alignment · Public edition

This edition aligns FORGE with its Public Product Vision v1.0: local-first model engineering, open-ended AI research, evaluation, export, and standalone use. Buildroom release direction, Growthroom scope, Northstar's flagship role, and Providence authority remain intact. Integration diagrams describe intended relationships, not implementation evidence.

- 1 **Original public brief.** The Great North — Public Architecture Vision Brief, 17 August 2026. Baseline for ecosystem naming, authority, inference, deployment, proof, and claim boundaries.
- 2 **Supplied Mainely Code project sources.** Providence canonical architecture, Northstar flagship direction, product-family direction, and Buildroom engineering sources. Public-level responsibilities only; confidential research and implementation details are not reproduced.
- 3 **Founder release direction, 5 September 2026.** Exact mission; FREE 3B / 4B / 7B / 8B expansion; LITE before September ends; ADVANCED before year-end. Reflected in the supplied website package.
- 4 **Growthroom introduction.** MAINELY CODE LLC, [public product announcement](#) ↗. Audience, content, conversion, and the growth-workspace direction.
- 5 **Growthroom workflow.** MAINELY CODE LLC, [evidence-backed marketing and sales announcement](#) ↗. Product scope, approval, publication receipts, and the operating loop.
- 6 **Growthroom development status.** MAINELY CODE LLC, [desktop application update](#) ↗. Publicly described as progressing toward release candidate, not confirmed generally available.
- 7 **Northstar public positioning.** [Mainely Northstar](#) ↗. Flagship cognitive role, authority boundaries, and locally verified versus live-integration release posture.
- 8 **Public product status.** [Mainely Code](#) ↗ and [Project Sasquatch](#) ↗. Read alongside product-specific release notes and current Store listings.
- 9 **FORGE Public Product Vision v1.0, 5 September 2026.** Mainely Code's supplied public brief and logo. Planned local-first model engineering and AI research; guided and custom workflows, evaluation, export, and standalone use. Development starting soon.

Other product status and external source links are carried forward from the supplied September 5 architecture review; this publication update is not a fresh implementation audit. Planned capabilities remain subject to development and validation. Product-specific release evidence takes precedence. © 2026 MAINELY CODE LLC.