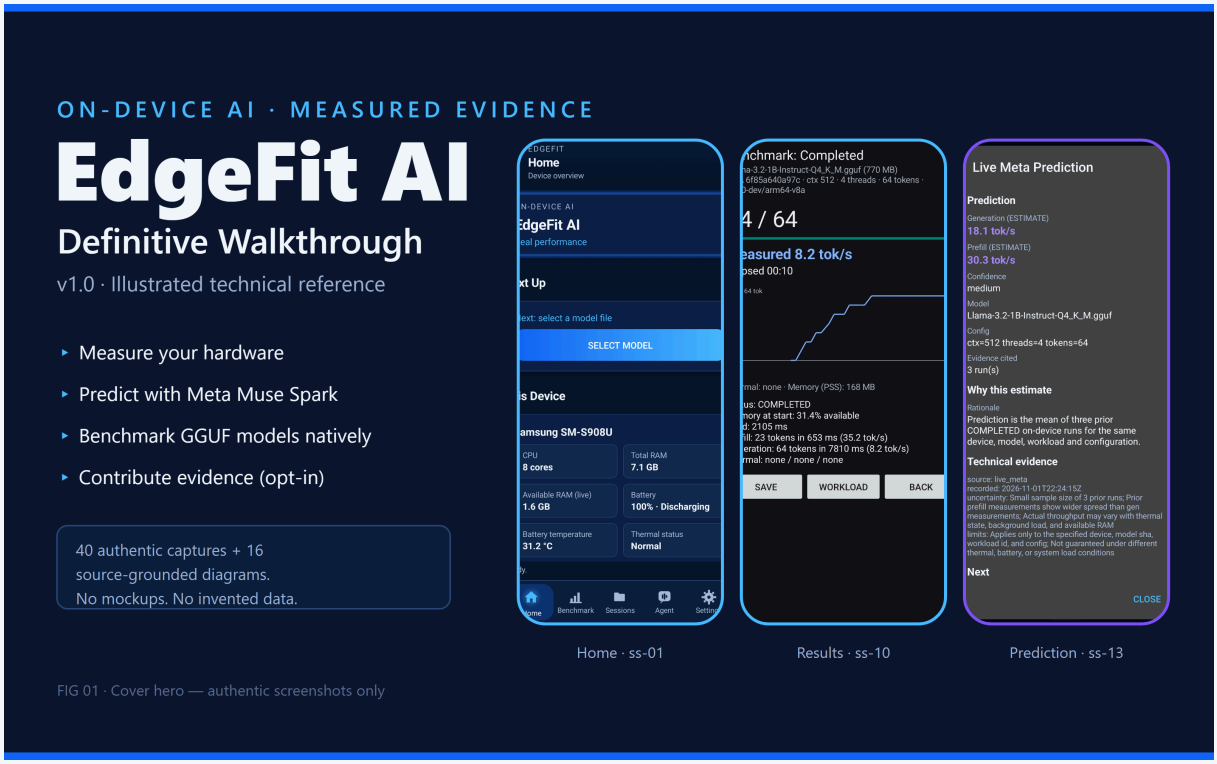
META GLOBAL AI DEVELOPER HACKATHON 2026

# EDGEFIT AI

THE COMPLETE DEVICE-LESS WALKTHROUGH

Game-guide edition • Editable master document • Screenshot and diagram insertion slots

From device detection to model benchmarking, Muse Spark reasoning, persistent evidence, and the governed Global Evidence Network.




PLACED 01 edgefit-fig-01-cover-hero.png

Figure 01: Cover hero: authentic Home, Results and Prediction captures. No mockups.

Figure 01 · Graphic · COVER HERO / EDGEFIT AERO UI COLLAGE

Editorial status: working publication draft. Insert final captures and revalidate all version-dependent claims before publication. This file is intentionally editable and contains labeled empty image placement boxes.

## Reader's compass

This is the deep technical manual behind the storefront-style GitHub README. Read the Golden Path to understand the experience without Android hardware; read the mission chapters for APIs, controls, data provenance, safety, quantitative evidence, and unresolved boundaries. All image slots are placeholders, not depictions of verified UI.

### Contents

01 Feature Encyclopedia and Evidence Legend
02 The Golden Path: fourteen moves through EdgeFit
03 Device Intelligence
04 Model Staging and Compatibility
05 Benchmarking Laboratory

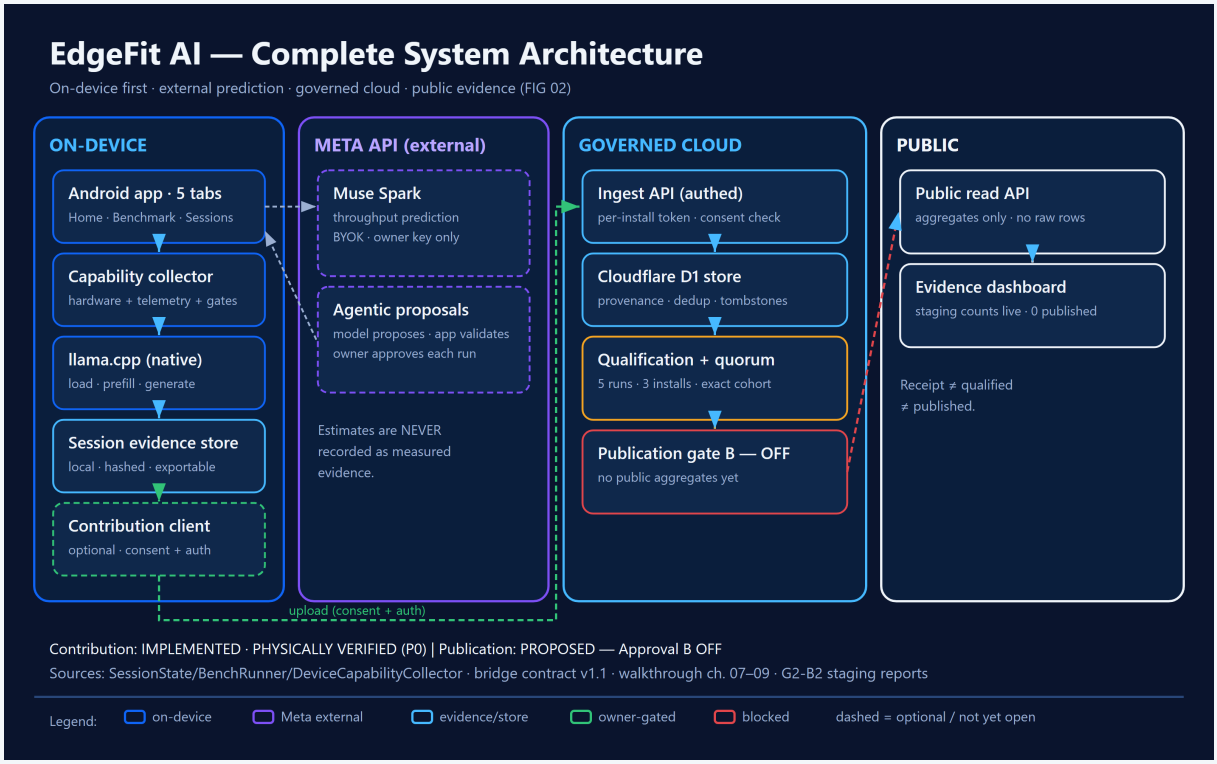
06 Sessions, Results and Evidence
07 Muse Spark Predictions and Agent Loop
08 Security, Privacy and Tool Validation
09 Global Evidence Network
10 Installation Safety and Recovery
11 Scientific Methodology and Performance
12 Distribution, Roadmap and Limitations
13 Settings, Onboarding and Help
14 Judge Quick Reference and Claim Trace
15 Visual Production and Capture Ledger

## Evidence vocabulary

| Label | Meaning |
|---|---|
| **PHYSICALLY VERIFIED** | Observed on a specific real device, for the documented scope and build. |
| **IMPLEMENTED** | Source feature exists, but this label alone does not prove physical end-to-end behavior. |
| **HOST PASS** | Host/unit or integration checks passed; physical operation may still be unverified. |
| **DEPLOYED, HONESTLY EMPTY** | Public service reachable but no qualifying published genuine cohort was established in the audited snapshot. |
| **PROPOSED / NOT RUN** | Design or guarded functionality not completed or not exercised. |
| **MEASURED / PREDICTED / ESTIMATED** | **Separate categories; a prediction or model suitability estimate never becomes a benchmark measurement.** |

## Scope and source baseline

This guide synthesizes the owner-provided DOC-AUDIT-01, DOC-RECON-01, DOC-BUILD-01, DOC-REVIEW-01 and DOC-BUILD-02 Muse execution reports. The audited Android documentation baseline was commit 3a55355, with specific later and cross-branch components identified by source notes. Global backend and website findings are historical snapshots and require refresh before publication. The current physical demonstration device is Samsung Galaxy S22 Ultra (SM-S908U), Android 16, arm64-v8a, EdgeFit AI 0.2.0 in the documented study.




PLACED 02 edgefit-fig-02-system-architecture.png

Figure 02: System overview: on-device app, native benchmark, Muse Spark (external, optional), governed cloud, public evidence. Contribution IMPLEMENTED and PHYSICALLY VERIFIED; publication PROPOSED, gate B OFF.

Figure 02 · Diagram · System overview: Android app ↔ local benchmark ↔ Muse Spark ↔ governed cloud ↔ website

# 01 Feature Encyclopedia

A quick-reference overview appears before the deep chapters, so reviewers immediately see the breadth of the system. The F-series lists earlier matrix categories; W-series are more granular UI capabilities and frequently overlap. Treat these as documented capabilities, not 50 independent product launches.

| Feature family | What it enables |
|---|---|
| **Device and navigation** | Device profile; live telemetry; home card and NextAction guidance; five-tab shell; three-page onboarding; full Device Details screen. |
| **Models and qualification** | GGUF staging with SHA-256 identity; offline model catalog; deterministic compatibility gates; explicit unverified estimates; manual A/B configuration and readiness checks. |
| **Benchmark laboratory** | Four-stage Model→Configure→Run→Results wizard; native llama.cpp CPU inference; load/prefill/generation rates; progress and cancellation; measured conditions. |
| **Evidence and sessions** | Persistent session ledger; prediction-before-run ordering; 13-state lifecycle; terminal outcomes; repeat measurements; host-computed prediction errors; JSON exports; workstation SQLite archive. |
| **Meta Muse Spark** | Bring-your-own-key custody; pre-run predictions with uncertainty; post-run cited follow-ups; bounded proposal loop; Approve/Modify/Reject/Finish controls; deterministic tool-call validators. |
| **Trust and recovery** | Fail-closed validators; explicit refusal reasons; scoped privacy controls; install identity and quarantine safeguards; no disguised missing measurements. |

| | |
|---|---|
| **Global evidence** | Local review and per-batch consent UI; authenticated contribution client; deployed backend and public explorer; cohort quorum and allowlist; withdrawal and history. Upload/publication scope is not fully verified. |
| **Distribution and research** | **Physical study artifacts; science methodology; host test suites; publication and release gates; future preview APK and store listing still proposed in audited record.** |

## Where to go next

Nontechnical reader: Golden Path → Device → Models → Benchmark → Sessions → Agent → Global. Technical judge: read each chapter's Under the hood, Evidence checkpoint, and the claim-trace appendix. Skeptical judge: begin with Methodology, Installation Safety, and Limitations.




PLACED 03 edgefit-fig-03-feature-atlas.png

Figure 03: Feature family atlas: eight groups mapped to chapters 3-9 and 13.

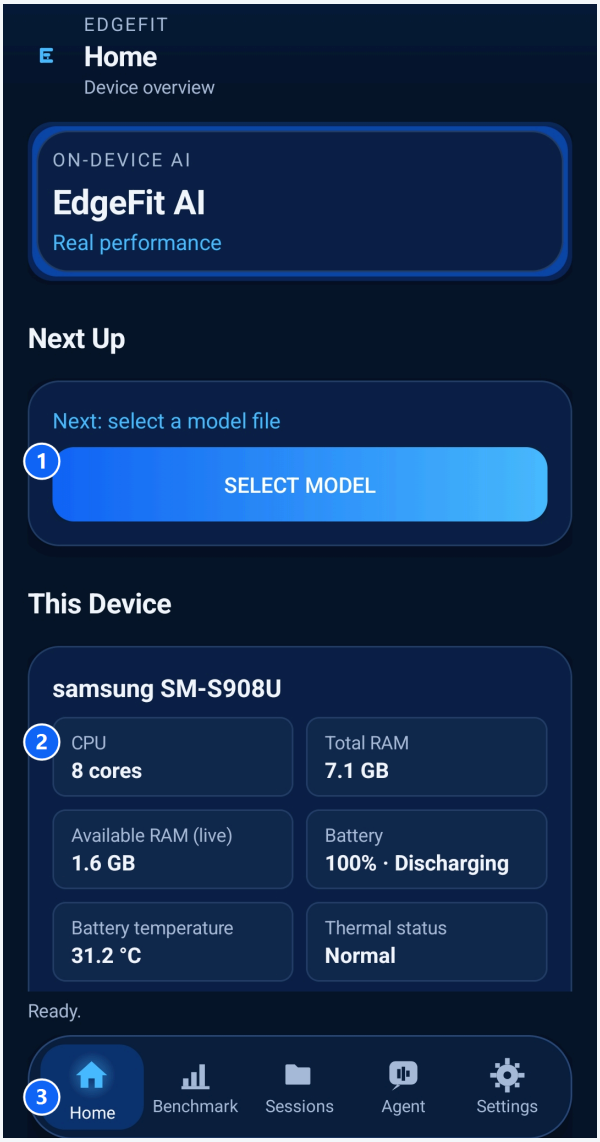
Figure 03 · Diagram · Feature family atlas — eight groups mapped to chapters

# 02 Golden Path — a 14-step virtual playthrough

Think of this section as the playable tutorial: a judge should be able to follow every tap without owning an Android phone. Each step explains visible behavior, internal record keeping, and what has or has not been verified.

### Step 01 — Open EdgeFit

Launch the installed benchmark variant without reinstalling. Home shows hardware context, staged model and latest result. Evidence remains partitioned by Android package ID.




PLACED 04 edgefit-ss-01-home-overview.png

Figure 04: Home overview (annotated): 1 next-action control, 2 live device-telemetry cards, 3 five-tab shell. SM-S908U, EdgeFit AI 0.2.0.

Figure 04 · Screenshot · edgefit-ss-01-home-overview.png

### Step 02 — Read onboarding

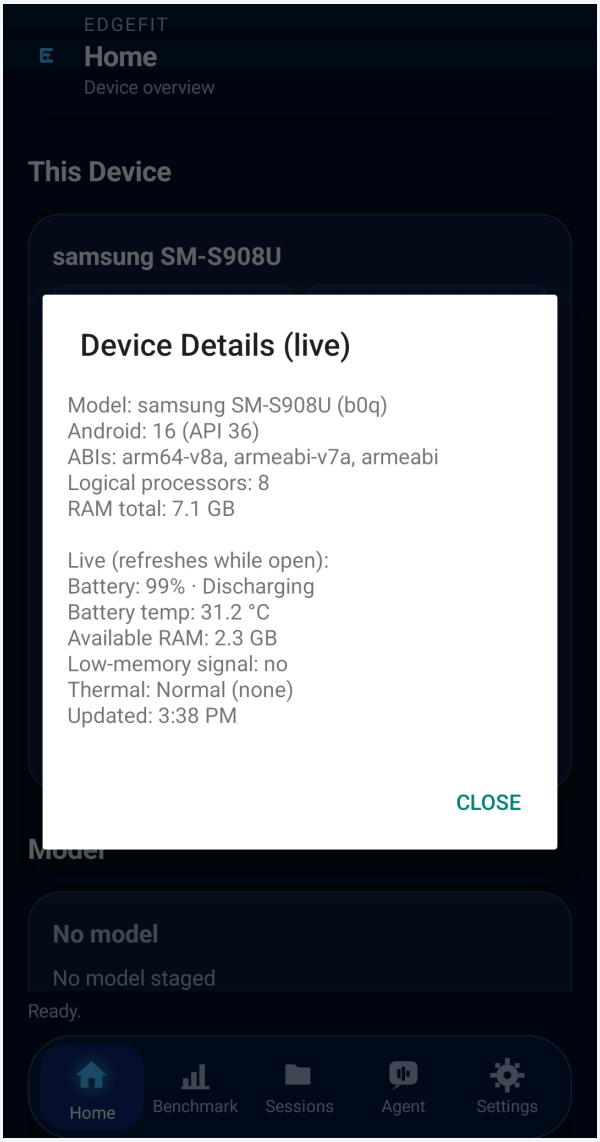
The three-page guide explains the product, benchmarking, and optional Muse-assisted experiments. Skip or advance without running anything. The auto-show gate retires after three qualifying completions.

### Step 03 — Learn the tab bar

Home, Benchmark, Sessions, Agent and Settings. A pending proposal can mark Agent. The back target returns Home.

### Step 04 — Inspect device details

Review model, ABI, Android API, memory, CPU counts, battery and thermal signals. Unavailable values are explicitly unavailable, not silently zero.




PLACED 05 edgefit-ss-03-hardware-details.png

Figure 05: Expanded hardware details: CPU, RAM, battery, thermal state, ABI and storage.

Figure 05 · Screenshot · edgefit-ss-03-hardware-details.png

### Step 05 — Pick a GGUF

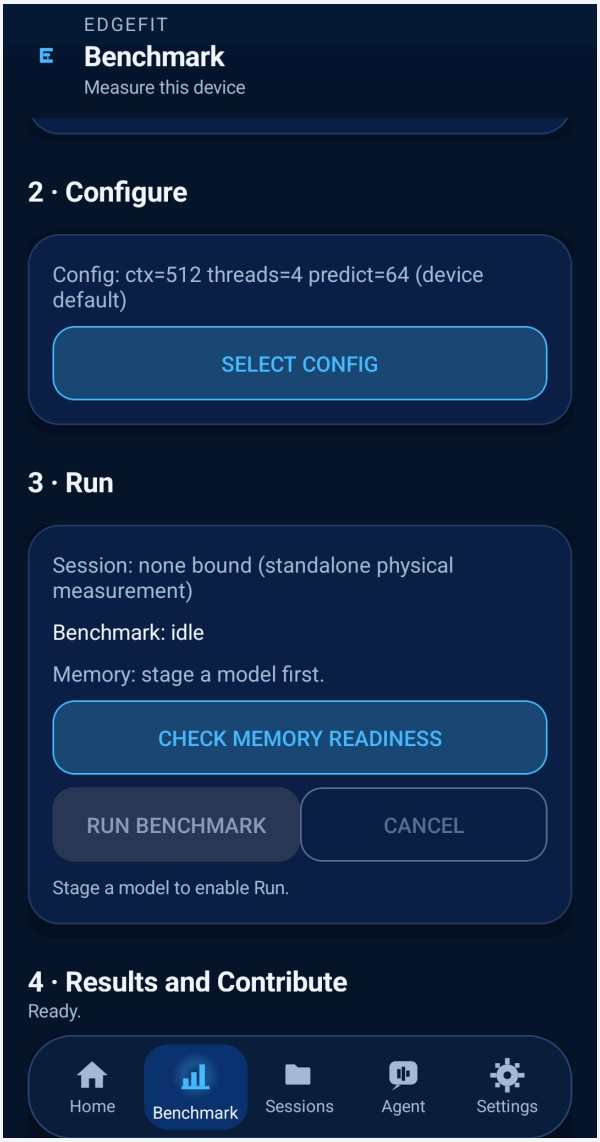
Browse offline catalog or use the system picker to select an actual on-device model file. Catalog confidence is not benchmark proof.

### Step 06 — Stage model identity

The app records displayed name, bytes, quantization context and SHA-256 identity. The hash, not a filename, anchors repeatability.

### Step 07 — Enter the benchmark wizard

Proceed through Model, Configure, Run, Results. Observe exact context/thread/predict triple and source of configuration.




PLACED 06 edgefit-ss-07-benchmark-setup.png

Figure 06: Benchmark setup, step 1 (Model): staged GGUF with SHA-256 identity.

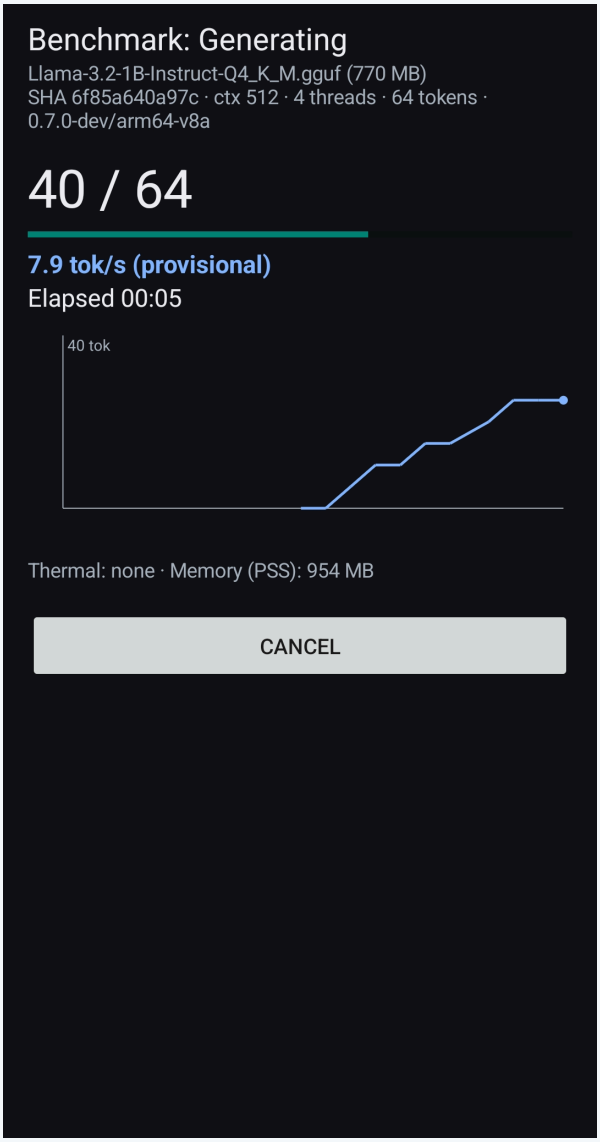
Figure 06 · Screenshot · edgefit-ss-07-benchmark-setup.png

### Step 08 — Inspect readiness

Memory advisory and thermal safeguards assess pre-run conditions. A blocked start is a meaningful safety result, not a crash.

### Step 09 — Authorize one run

Press Run after selecting a supported configuration. The app performs real native inference, not a synthetic token-rate animation.



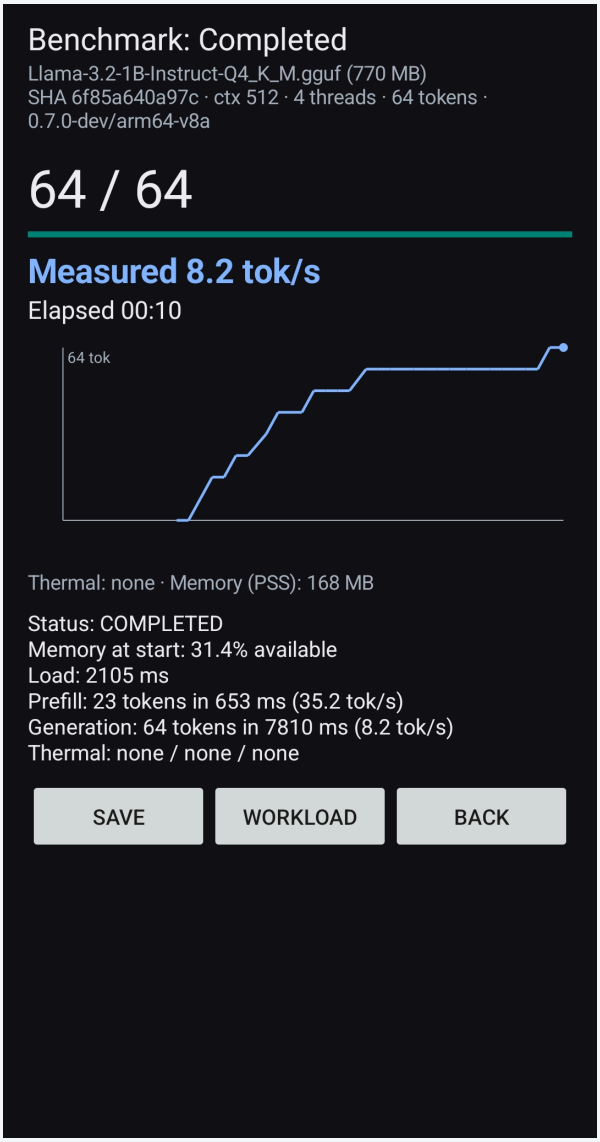

PLACED 07 edgefit-ss-09-benchmark-running.png

Figure 07: Benchmark running: live token progress, generation throughput and elapsed time.

Figure 07 · Screenshot · edgefit-ss-09-benchmark-running.png

### Step 10 — Read measured output

Inspect load, prefill and generation measurements, device conditions, output count and any failure or cancel state.



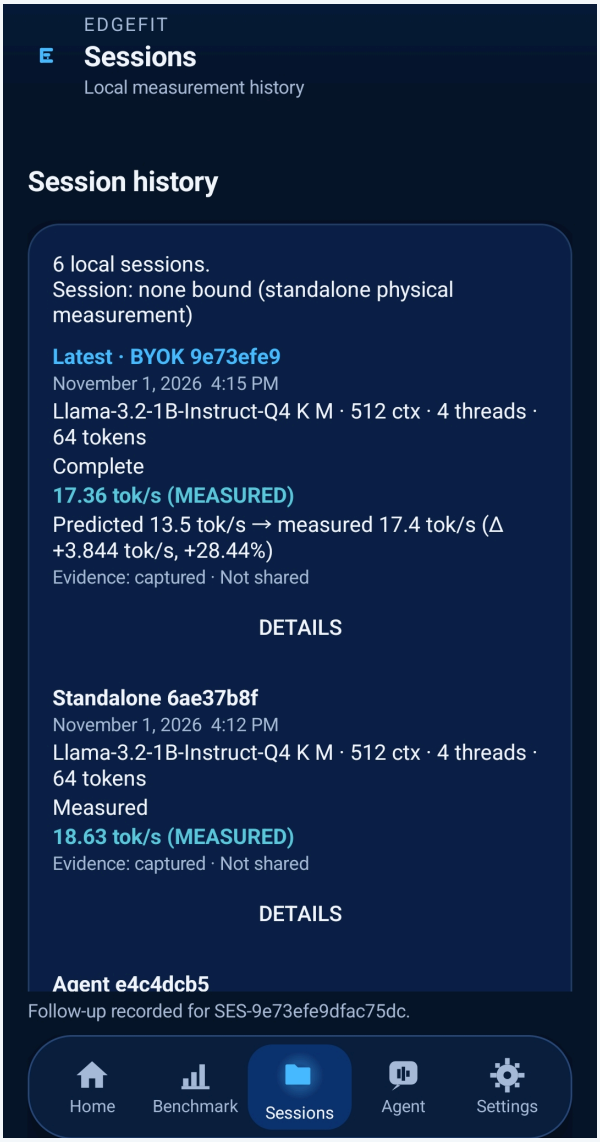

PLACED 08 edgefit-ss-10-benchmark-results.png

Figure 08: Completed benchmark result: full token run with measured throughput.

Figure 08 · Screenshot · edgefit-ss-10-benchmark-results.png

### Step 11 — Open Sessions

Find the durable local ledger, session detail, state timeline, repeat action, and exportable JSON evidence.




PLACED 09 edgefit-ss-12-session-history.png

Figure 09: Session history: local sessions with measured-vs-predicted deltas; evidence captured, not shared.

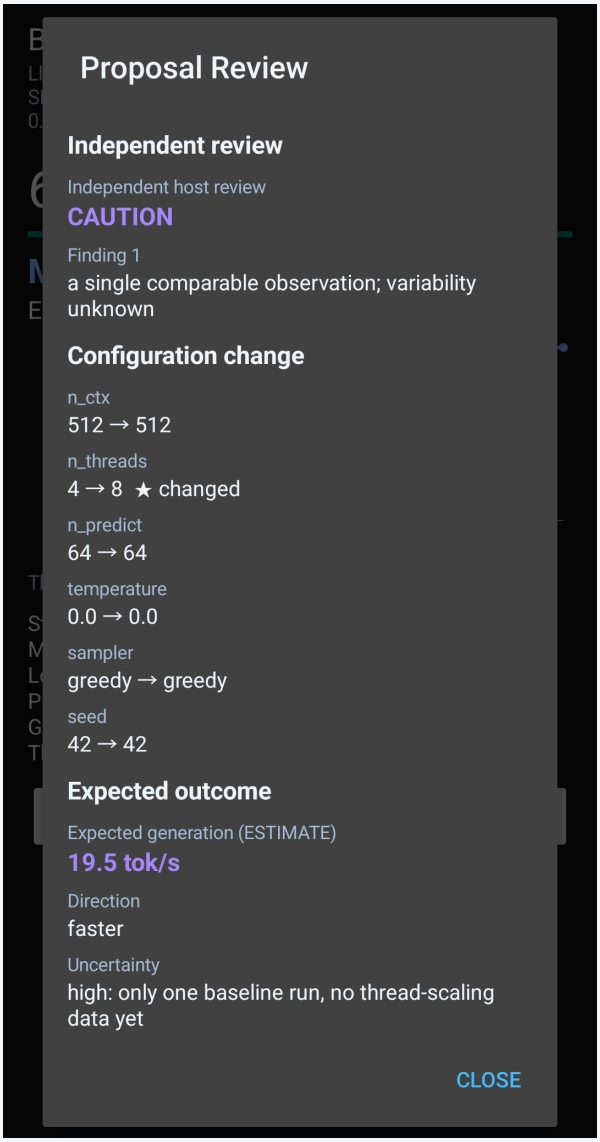
Figure 09 · Screenshot · edgefit-ss-12-session-history.png

### Step 12 — Compare with prediction

Where there is a prior Muse prediction bound to the run, the app computes error on the host and keeps predicted distinct from measured.

### Step 13 — Explore guided agent

Configure an authorized Meta key, inspect proposal, evidence citations, confidence and budget, then approve/reject/modify/finish where supported.



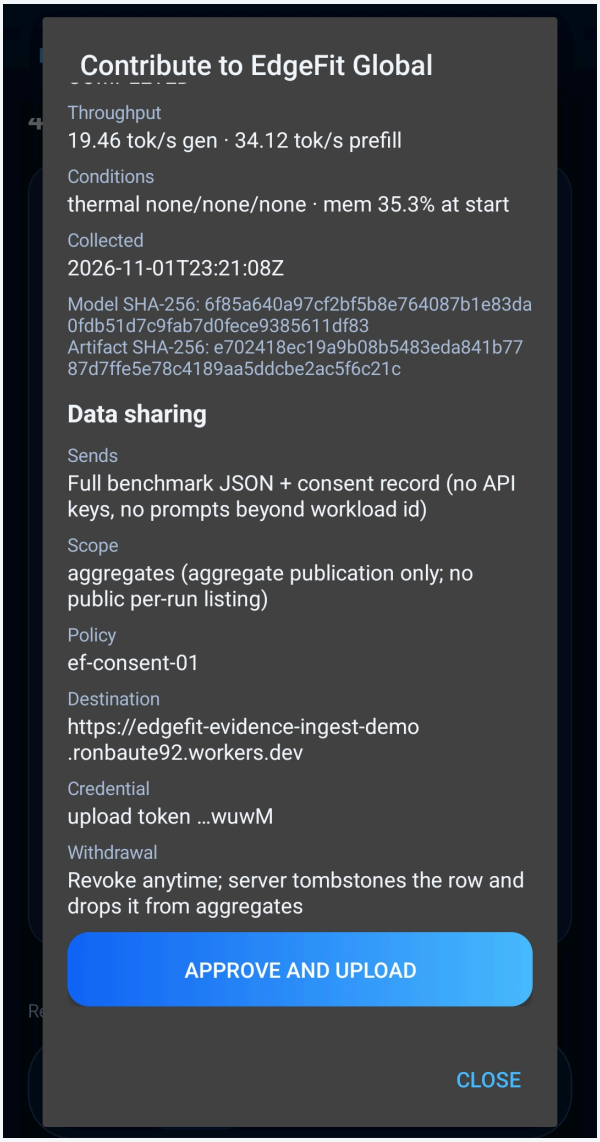

PLACED 10 edgefit-ss-22-agent-proposal-review.png

Figure 10: Proposal review: independent host CAUTION, 4 to 8 thread change, expected 19.5 tok/s (ESTIMATE).

Figure 10 · Screenshot · edgefit-ss-22-agent-proposal-review.png

## Step 14 — Inspect Global contribution

Review the exact proposed evidence and consent scope. Do not assume upload or public publication; show current connection and qualification status.




PLACED 11 edgefit-ss-34-contribution-approve-upload.png

Figure 11: Contribution consent review: measured throughput, conditions, ef-consent-01, aggregates scope, explicit APPROVE AND UPLOAD. Pre-upload review, not a receipt.

Figure 11 · Screenshot · edgefit-ss-34-contribution-approve-upload.png




PLACED 12 edgefit-fig-12-golden-path.png

Figure 12: Fourteen-step first-launch-to-contribution journey. Steps 1-8 required; 9-14 optional, each owner-approved.

Figure 12 · Flowchart · Fourteen-step first-launch-to-contribution journey

# 03 Device Intelligence

Your first mission is to establish which hardware is being evaluated, what it reports, and which signals are available right now. Android fragmentation matters because memory, ABI, processor count and sustained thermal behavior affect whether a model can execute reliably.

## Mission briefing

Your first mission is to establish which hardware is being evaluated, what it reports, and which signals are available right now. Android fragmentation matters because memory, ABI, processor count and sustained thermal behavior affect whether a model can execute reliably.

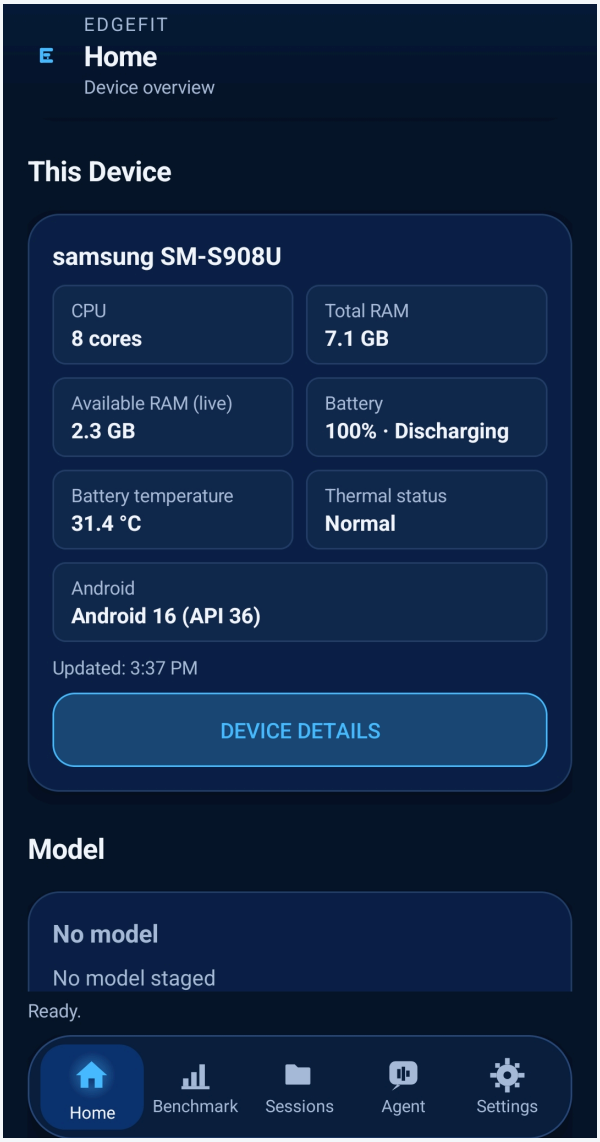
## Walkthrough — what a judge sees and does

Open Home and identify the device card and NextAction line.

Open Device Details for the full hardware and telemetry breakdown.

Find memory and thermal conditions; note explicit unavailable fields and any advice.

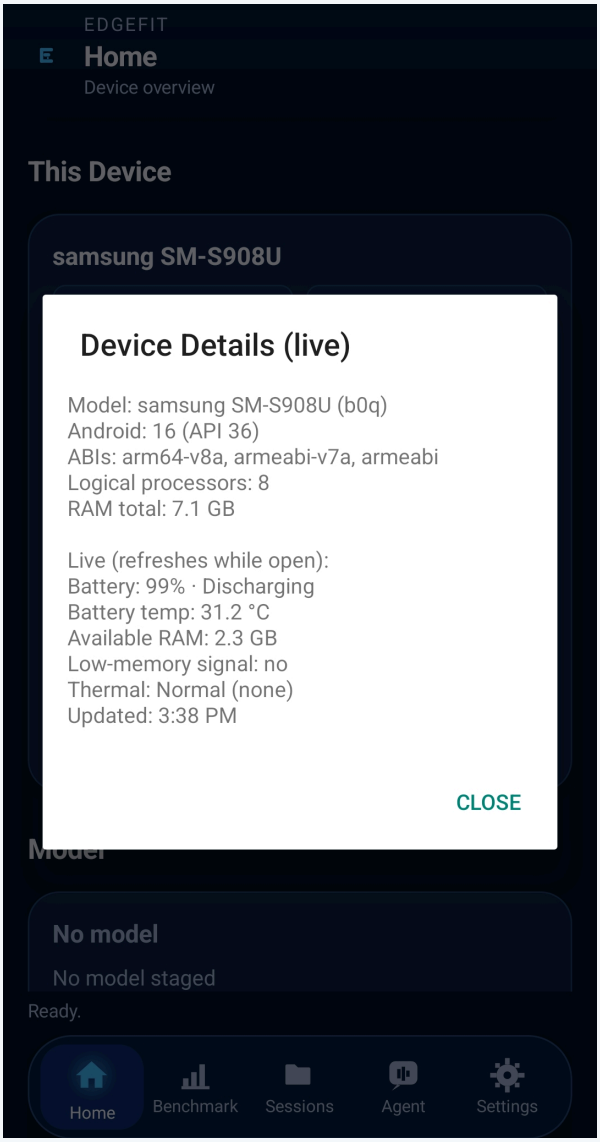
Return to Home. Observe that a profile is a hardware observation, not an inference benchmark.




PLACED 13 edgefit-ss-02-device-profile.png

Figure 13: Device profile: model, CPU, RAM, live availability and thermal snapshot.

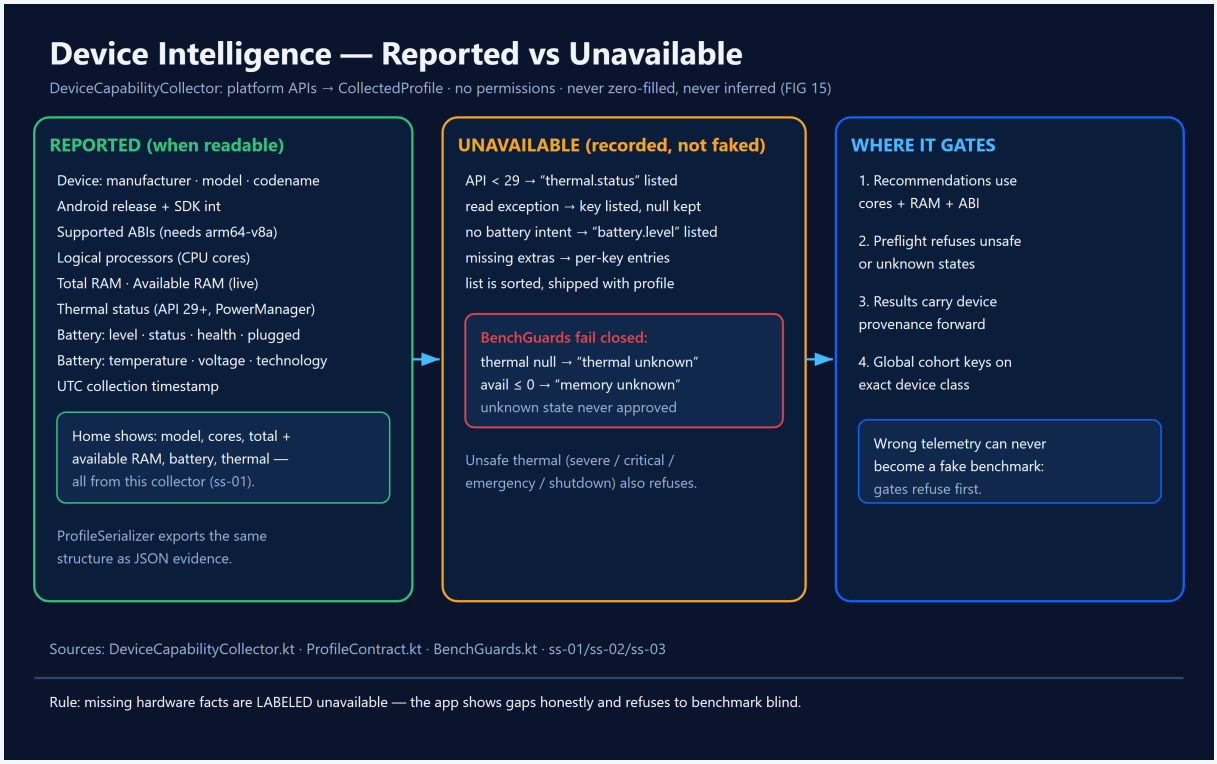
Figure 13 · Screenshot · edgefit-ss-02-device-profile.png




PLACED 14 edgefit-ss-03-hardware-details.png

Figure 14: Expanded hardware details: CPU, RAM, battery, thermal state, ABI and storage.

Figure 14 · Screenshot · edgefit-ss-03-hardware-details.png




PLACED 15 edgefit-fig-15-device-intelligence.png

Figure 15: Device-telemetry source map: reported fields versus recorded-unavailable fields; guards fail closed.

Figure 15 · Diagram · Device-telemetry / unavailable-field source map

## Under the hood — implementation and control flow

DeviceCapabilityCollector.collect() reads Android Build and SDK fields, supported ABIs, availableProcessors, ActivityManager.MemoryInfo, battery status and supported thermal indicators. ProfileContract constrains what can be exported. No permission-only or unsupported field should be invented.

The app differentiates relatively static identity (manufacturer, model, architecture, SDK level) from volatile telemetry (free memory, battery, temperature, thermal severity). DeviceTelemetry handles the latter; sampling and freshness must be verified against current source before precise cadence is printed.

MemPolicy and BenchGuards produce advisory/refusal outcomes. The review identified warning bands at 15% and 5%; their exact denominator and guard semantics must be sourced from current code before any diagram asserts an equation.

## States, edge cases and refusal behavior

- UNKNOWN readings remain unavailable; do not substitute zero.
- Battery or memory warnings can be advisory rather than blocking.
- Platform-reported CPU count is not proof that all cores sustain peak speed.

## Evidence checkpoint

- DeviceCapabilityCollector.kt; ProfileContract.kt; DeviceTelemetry.kt; MemPolicy.kt; BenchGuards.kt.
- DOC-AUDIT-01: profiling and telemetry physically verified on the S22 Ultra, within the tested build.

## Judge interpretation

Do not equate available RAM with installable model size; quantization, context allocation, runtime buffers and heat complicate suitability.

# 04 Models — Selection, Suitability and Identity

The model mission answers two different questions: which candidate could fit this handset, and which exact artifact was actually measured? EdgeFit refuses to treat a recommendation estimate as a completed benchmark.

## Mission briefing

The model mission answers two different questions: which candidate could fit this handset, and which exact artifact was actually measured? EdgeFit refuses to treat a recommendation estimate as a completed benchmark.

## Walkthrough — what a judge sees and does

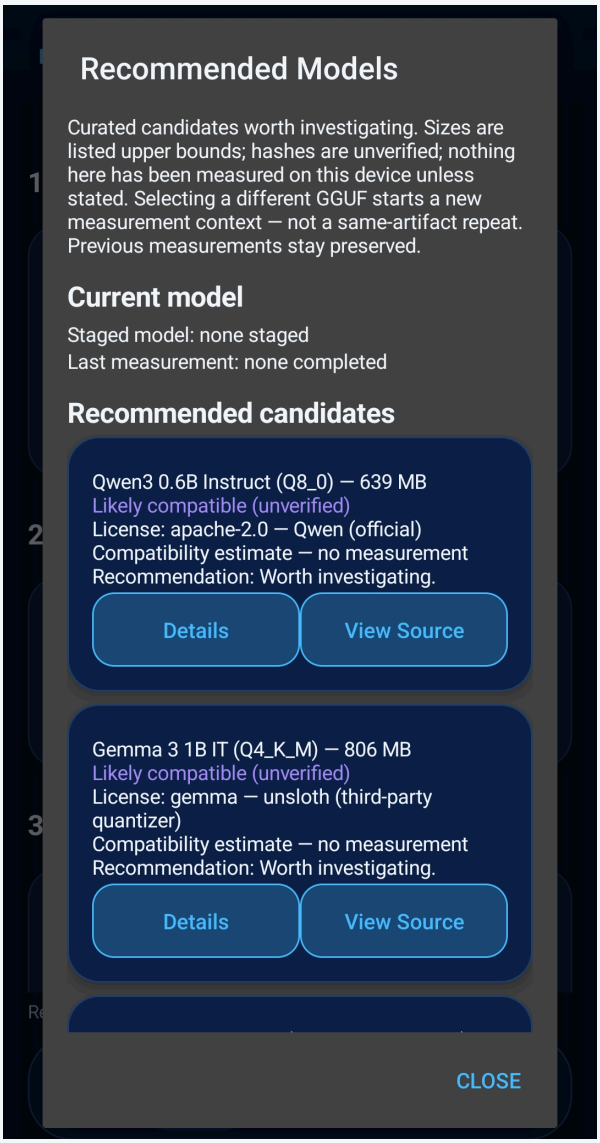
Open recommendation list and read the status markers.

Expand one recommendation to read its reasons and caveats.

Browse the catalog and locate any source/details actions.

Choose a real GGUF using the Android picker; confirm staging and identity.

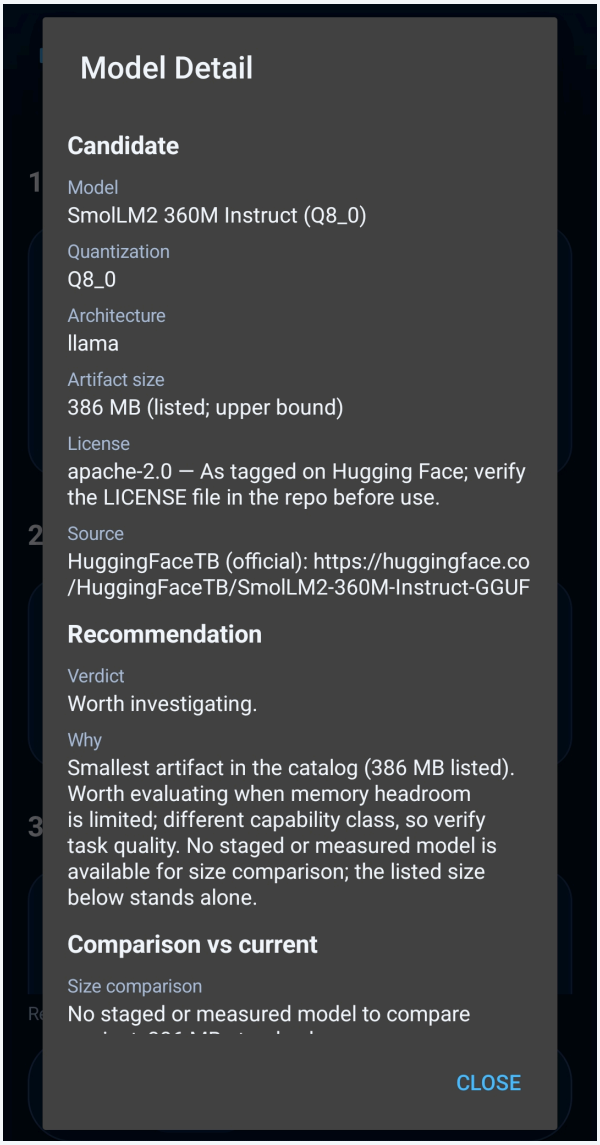
Return to benchmark setup and verify model name/hash or relevant identity details.




PLACED 16 edgefit-ss-04-model-recommendations.png

Figure 16: Model recommendations ranked for the measured device.

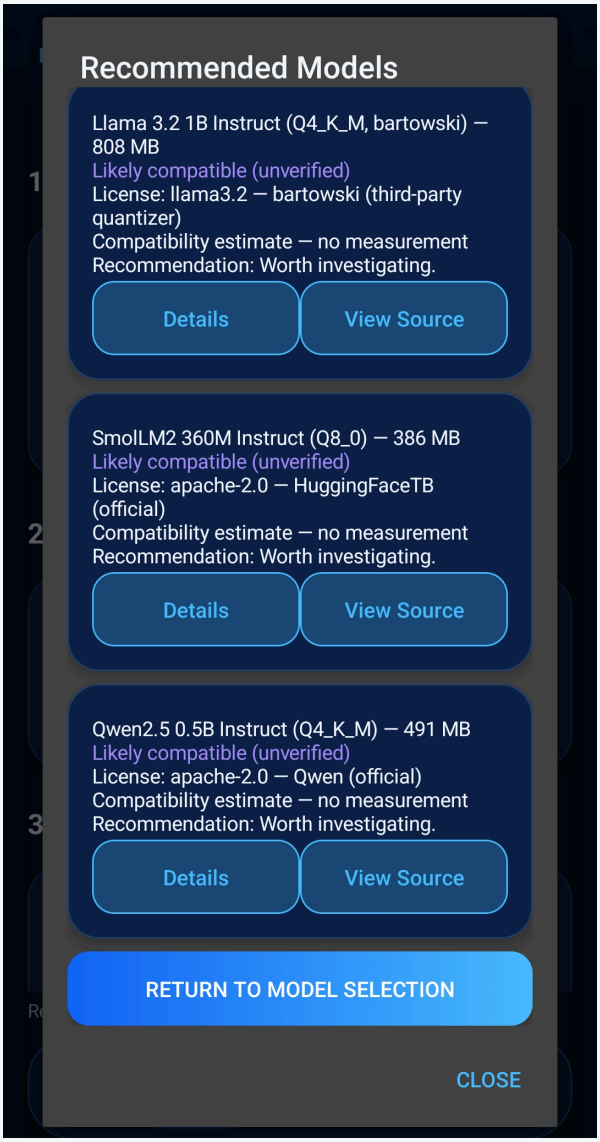
Figure 16 · Screenshot · edgefit-ss-04-model-recommendations.png




PLACED 17 edgefit-ss-05-model-recommendation-detail.png

Figure 17: Recommendation detail: compatibility estimate with evidence limits.

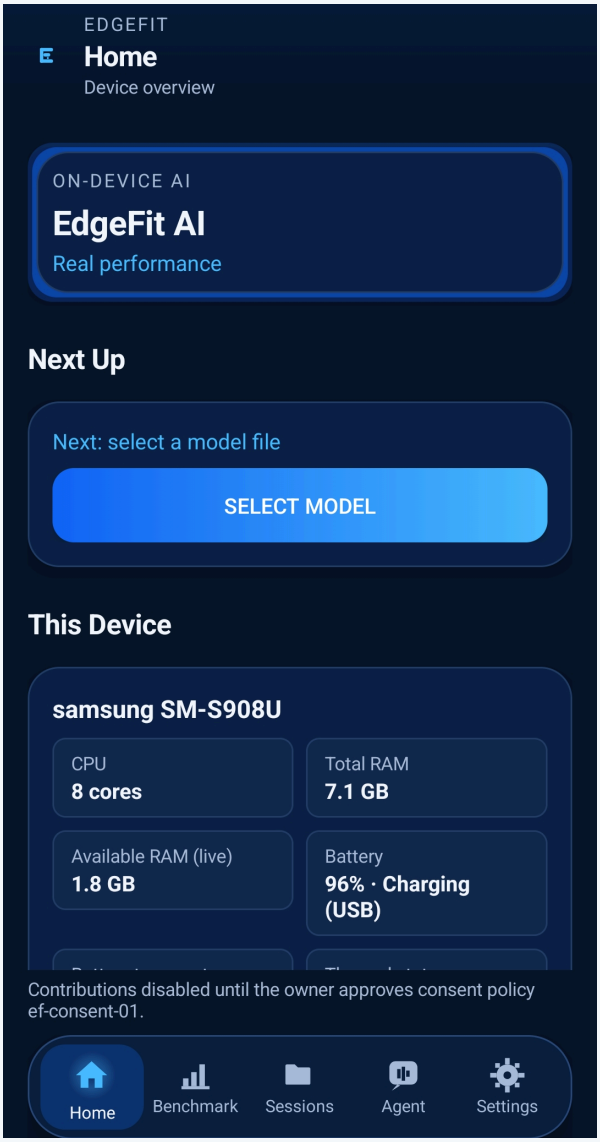
Figure 17 · Screenshot · edgefit-ss-05-model-recommendation-detail.png




PLACED 18 edgefit-ss-06-model-catalog.png

Figure 18: Offline model catalog with suitability notes (notes are not benchmark proof).

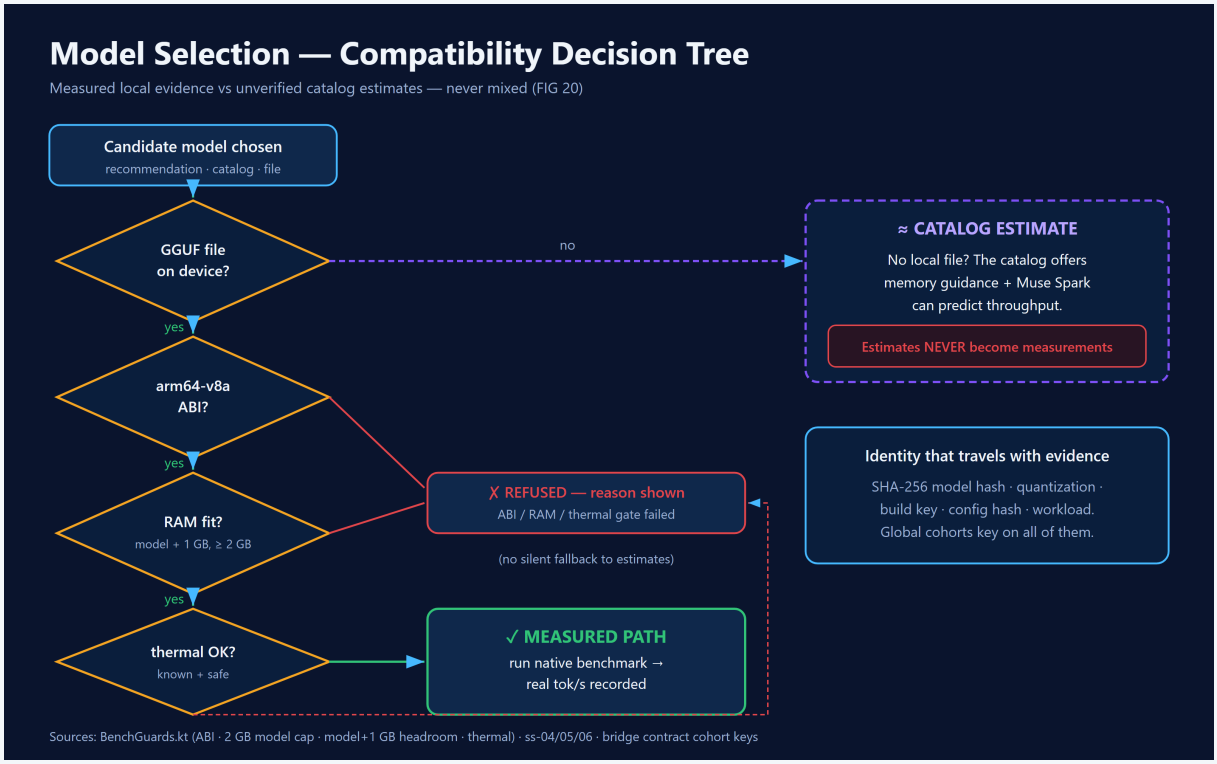
Figure 18 · Screenshot · edgefit-ss-06-model-catalog.png




PLACED 19 edgefit-ss-32-home-select-model.png

Figure 19: Home model-selection state: next-action prompts SELECT MODEL.

Figure 19 · Screenshot · edgefit-ss-32-home-select-model.png




PLACED 20 edgefit-fig-20-model-decision-tree.png

Figure 20: Model recommendation and evidence gating decision tree. Measured path versus catalog estimate, never mixed.

Figure 20 · Flowchart · Model recommendation / evidence gating decision tree

## Under the hood — implementation and control flow

Users stage a genuine GGUF through Android document selection. stageModel() computes and records SHA-256 along with file bytes and other BenchModelIdentity fields. Repeated results must bind to the exact hash and executed configuration.

ModelCatalog supplies a curated offline shortlist; ModelRecommend and ModelSuitability implement deterministic suitability logic. Only local evidence for a matching artifact and hardware context supports measured performance labels. The audit specifically warns that the model architecture is not exhaustively parsed for arbitrary cross-architecture support claims.

Quantization, RAM, storage, context size and thread policy interact. An estimate is a planning aid; output rates must come from measured runs or be labeled predicted/estimated.

## States, edge cases and refusal behavior

Missing or incompatible file: stop with a named reason.

Unsuitable model: explain the constraint rather than quietly claiming a rate.

A screenshot of a recommended model is not proof the model was benchmarked.

## Evidence checkpoint

MainActivity.kt stageModel(); ModelCatalog.kt; ModelCatalogView.kt; ModelSuitability.kt; ModelRecommend.kt.

Audit F07–F09 and W-series mapping; catalog host-tested, GGUF staging physically verified.

# 05 Benchmark Laboratory — Real On-Device Inference

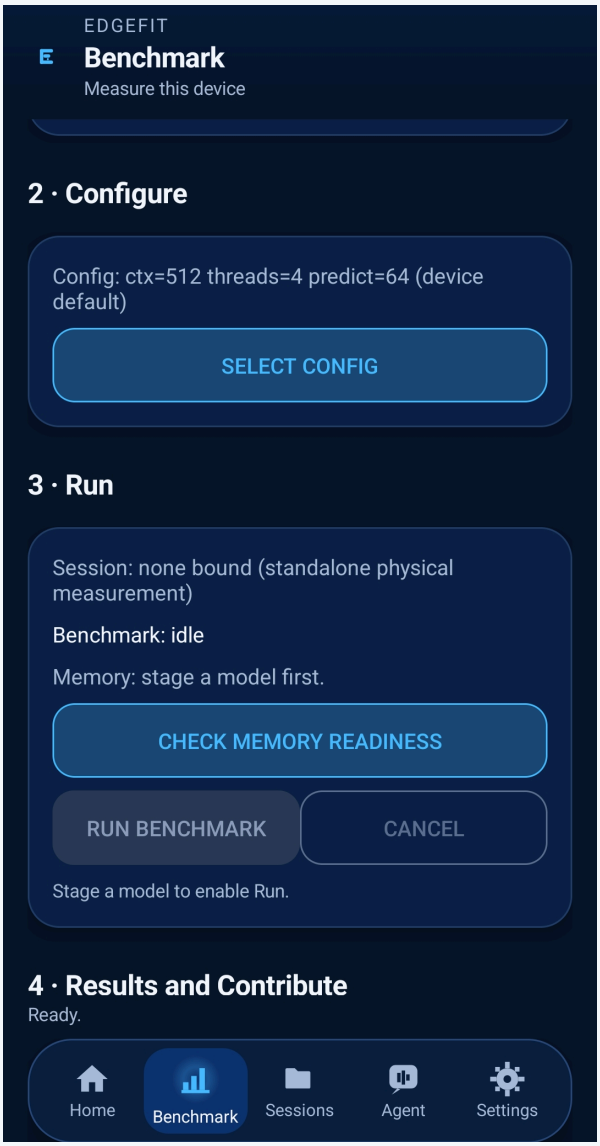
This is the core proof mission. A physical phone loads an actual GGUF, executes native llama.cpp inference with a controlled workload, and reports measurable load, prefill and generation performance. A run can fail or be cancelled without becoming a success.

## Mission briefing

This is the core proof mission. A physical phone loads an actual GGUF, executes native llama.cpp inference with a controlled workload, and reports measurable load, prefill and generation performance. A run can fail or be cancelled without becoming a success.

## Walkthrough — what a judge sees and does

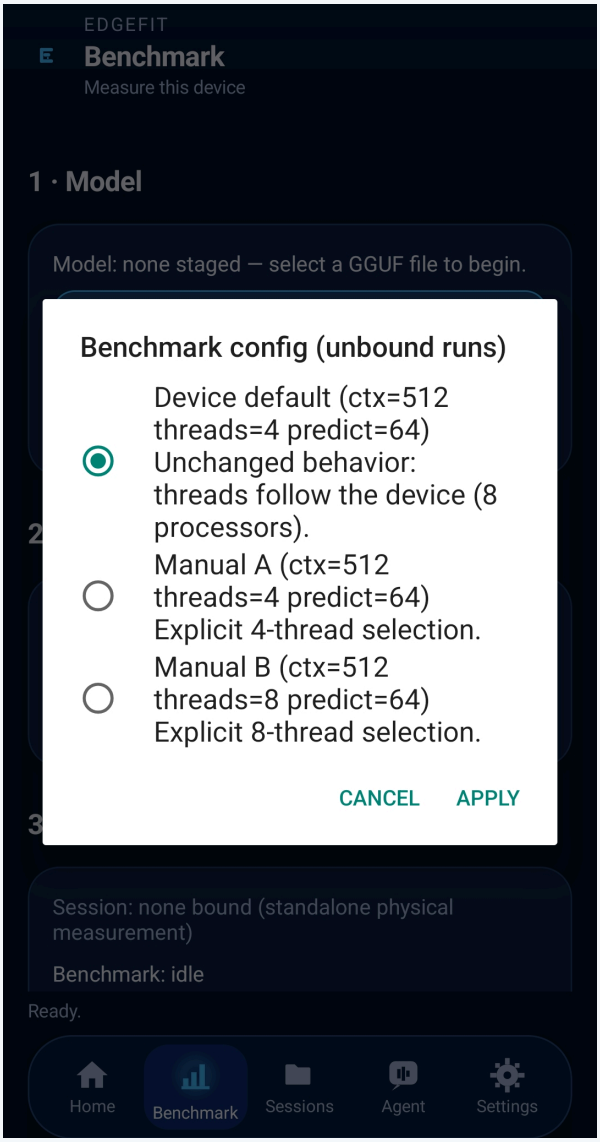
1. Read the Model wizard page and inspect staged identity.
2. Use Configure to select device default, Manual A or Manual B; read refusal text if unsupported.
3. Confirm context, threads and predict values displayed before the run.
4. Advance to Run, check that no cloud contribution is required to benchmark.
5. Press Run; watch phase progress and observe optional Cancel.
6. Read Results: model, exact config, token counts, load/prefill/generation rates, thermal/memory condition, terminal status.




PLACED 21 edgefit-ss-07-benchmark-setup.png

Figure 21: Benchmark setup, step 1 (Model): staged GGUF with SHA-256 identity.

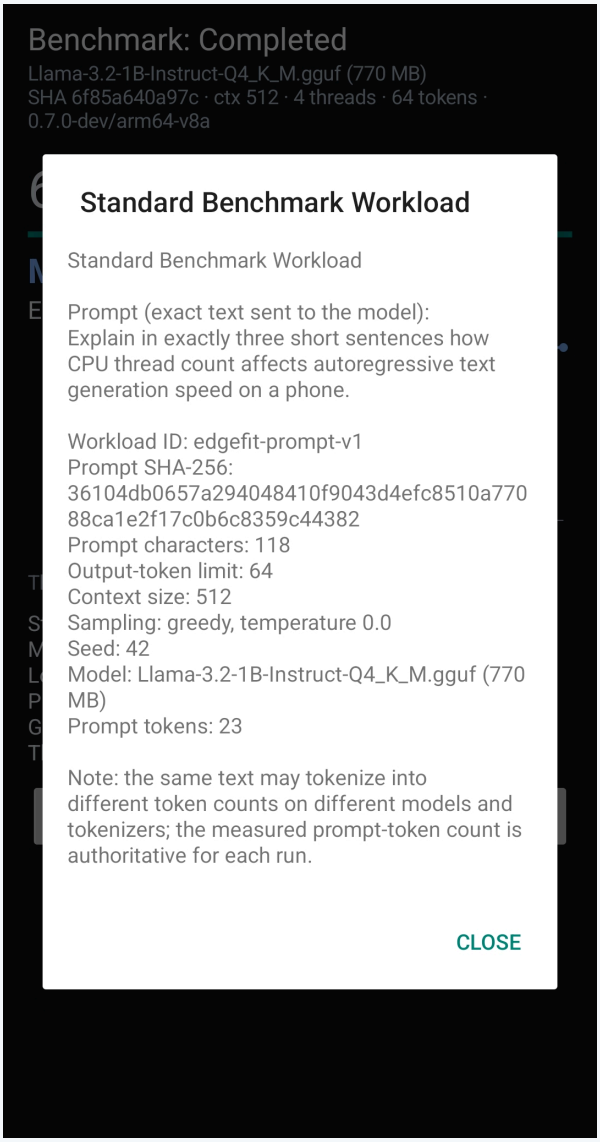
Figure 21 · Screenshot · edgefit-ss-07-benchmark-setup.png




PLACED 22 edgefit-ss-08-benchmark-setup-advanced.png

Figure 22: Advanced benchmark configuration: context, threads and prediction length.

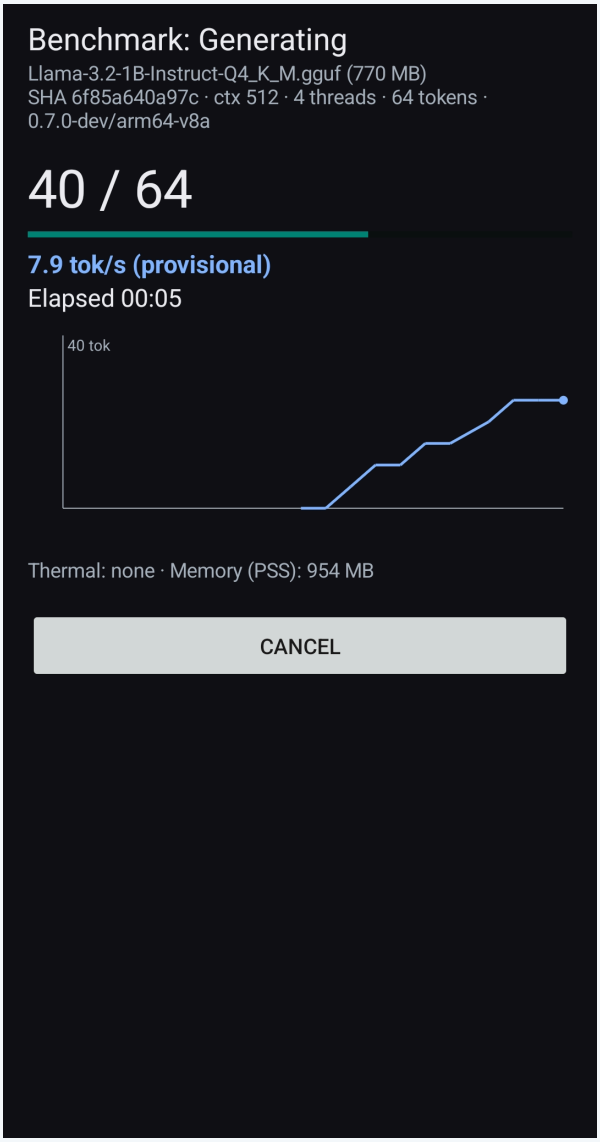
Figure 22 · Screenshot · edgefit-ss-08-benchmark-setup-advanced.png




PLACED 23 edgefit-ss-08-02-benchmark-workload.png

Figure 23: Benchmark workload: edgefit-prompt-v1 standard prompt and configuration.

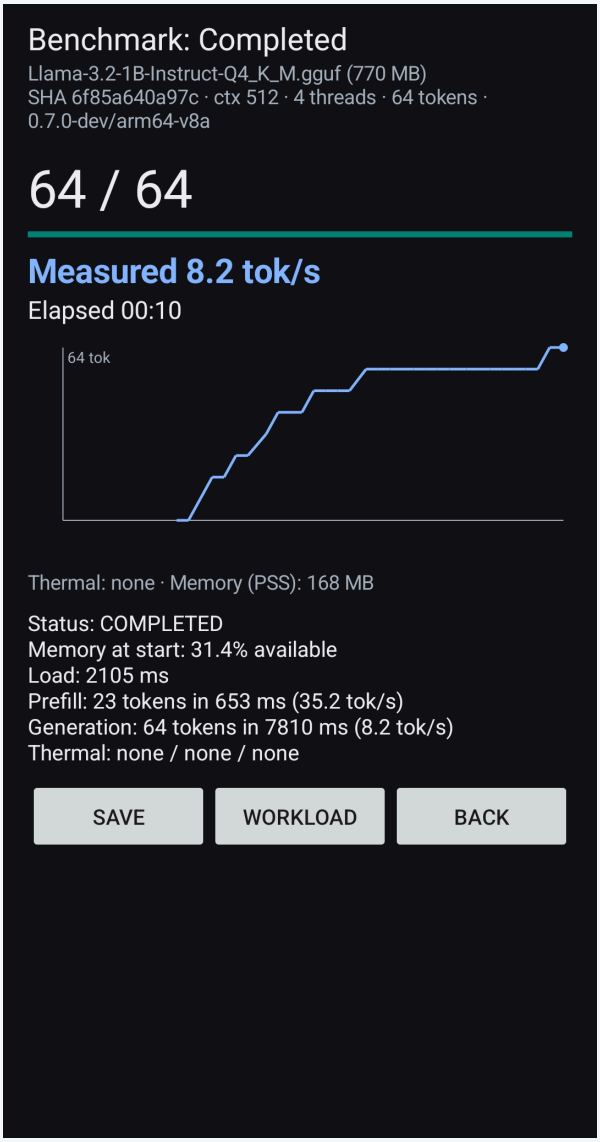
Figure 23 · Screenshot · edgefit-ss-08-02-benchmark-workload.png




PLACED 24 edgefit-ss-09-benchmark-running.png

Figure 24: Benchmark running: live token progress, generation throughput and elapsed time.

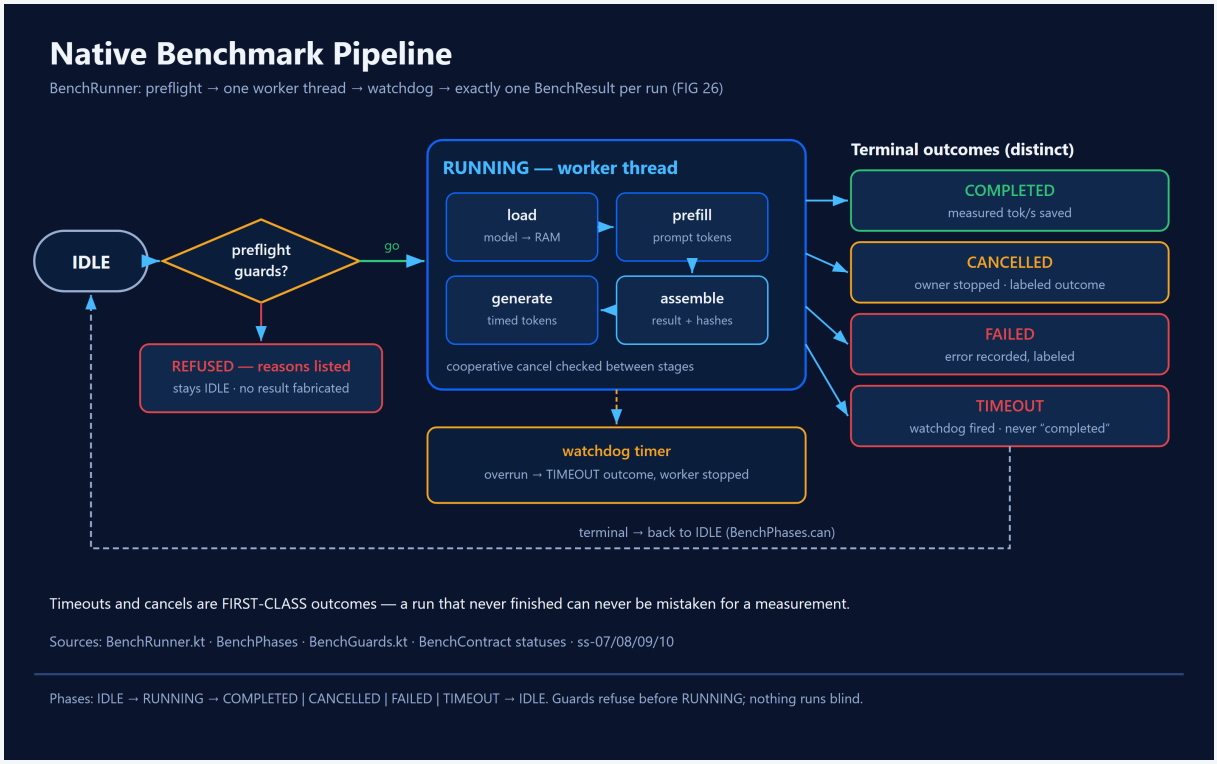
Figure 24 · Screenshot · edgefit-ss-09-benchmark-running.png




PLACED 25 edgefit-ss-10-benchmark-results.png

Figure 25: Completed benchmark result: full token run with measured throughput.

Figure 25 · Screenshot · edgefit-ss-10-benchmark-results.png




PLACED 26 edgefit-fig-26-benchmark-pipeline.png

Figure 26: Benchmark native execution: preflight guards, worker thread (load, prefill, generate, assemble), watchdog, four distinct terminal outcomes.

Figure 26 · Flowchart · Benchmark four-stage native execution and cancellation

**INSERT SCREENSHOT HERE**

□ 27 edgefit-ss-37-error-validation-state.png

Capture an actual refusal; placeholder pending.

Figure 27 · Screenshot · edgefit-ss-37-error-validation-state.png

## Under the hood — implementation and control flow

The benchmark wizard is Model → Configure → Run → Results. BenchWorkflow determines presentation, BenchConfig holds workload identifiers and native arguments, and BenchRunner/LlamaBridge/JNI enter actual llama.cpp. The canonical test uses a fixed CPU-thread-count question, greedy sampling and seed 42; the A/B study used context 512 and predict 64.

ManualBenchConfig exposes three choices: device default, Manual A (4 threads) and Manual B (8 threads). Manual B is listed but disallowed below eight processors, with the reason shown. A selection never auto-starts a run; the owner explicitly presses Run. Bound-session configuration takes precedence over a parked standalone override.

Guardrails surface memory, thermal and runner-busy conditions. Distinct terminal states include COMPLETED, FAILED, CANCELLED and TIMEOUT. Commit 3a55355 introduced a llama abort callback for mid-decode cancel; this must be validated again on the currently distributed build.

Throughput is generally computed from completed phase token counts and elapsed duration. Preserve the stored definitions of load, prefill and generation rather than recomputing or presenting unrelated rates as interchangeable.

## States, edge cases and refusal behavior

NO MODEL: Run remains unavailable until a GGUF is staged.

BUSY: concurrent benchmark requests are blocked.

THERMAL/MEMORY: advice or refusal must be shown faithfully.

CANCELLED, TIMEOUT and FAILED are not counted as successful completed evidence.

### Evidence checkpoint

BenchConfig.kt; BenchWorkflow.kt; ManualBenchConfig.kt; BenchRunner.kt; LlamaBridge.kt; bench_jni.cpp.

S22 study record eval/comp-01b/ with six real runs; audit screenshot example reports 64/64 generated tokens at 15.3 tok/s for one run.

### Judge interpretation

The science takeaway is not that more threads are always slower. The qualified result is an exploratory observation from this exact phone/model/configuration, with three replicates per configuration.

## 06 Sessions, Results and Evidence Ledger

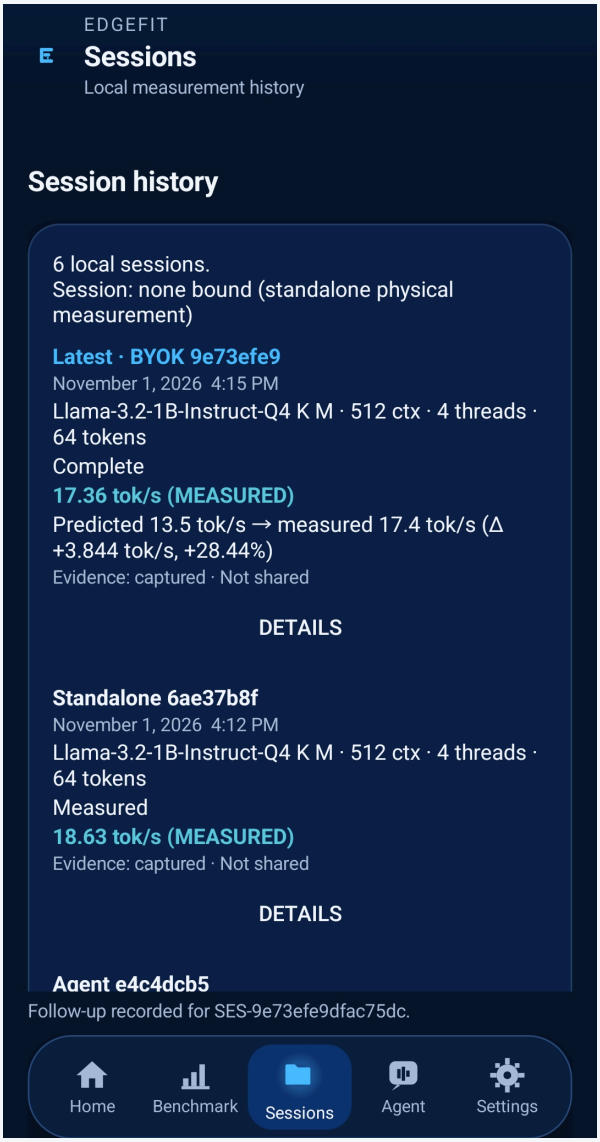
An evidence session is a persistent chain of custody, not just a screenshot of tok/s. It binds a model hash, hardware, exact settings, pre-run prediction when applicable, real execution and post-run interpretation.

### Mission briefing

An evidence session is a persistent chain of custody, not just a screenshot of tok/s. It binds a model hash, hardware, exact settings, pre-run prediction when applicable, real execution and post-run interpretation.

### Walkthrough — what a judge sees and does

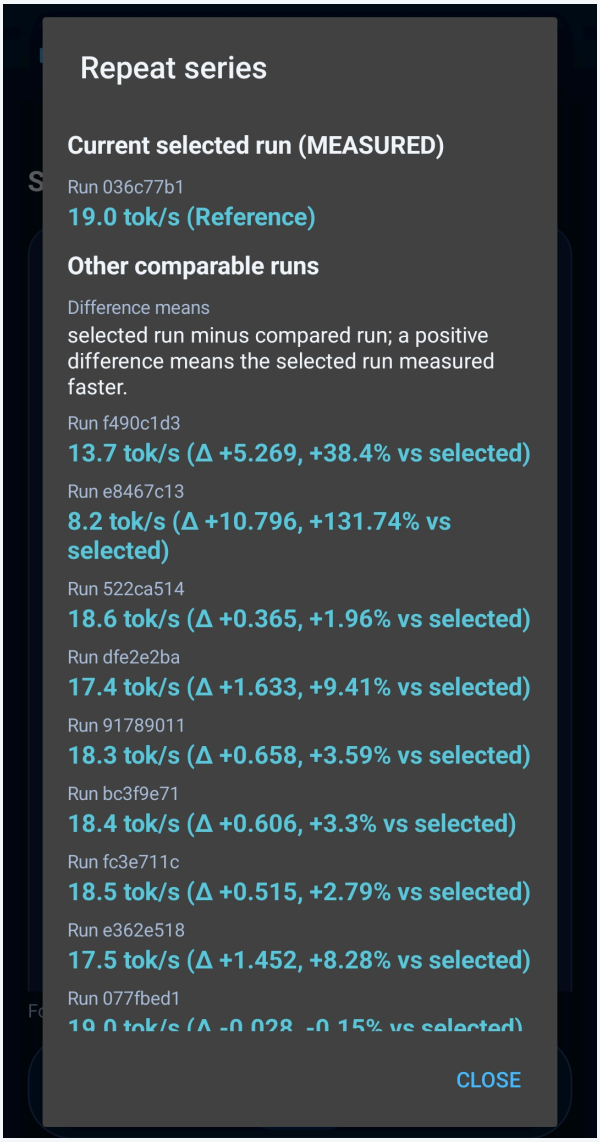
1. Open Sessions and find a recently completed run.
2. Open session detail to inspect identity, metric values and timeline.
3. Look for a bound prior prediction, then read the host-computed delta.
4. Use repeat/compare to evaluate another measurement without rewriting history.
5. Export benchmark JSON or device profile to a chosen file destination.
6. Identify terminal or incomplete sessions and how the ledger records them.




PLACED 28 edgefit-ss-12-session-history.png

Figure 28: Session history: local sessions with measured-vs-predicted deltas; evidence captured, not shared.

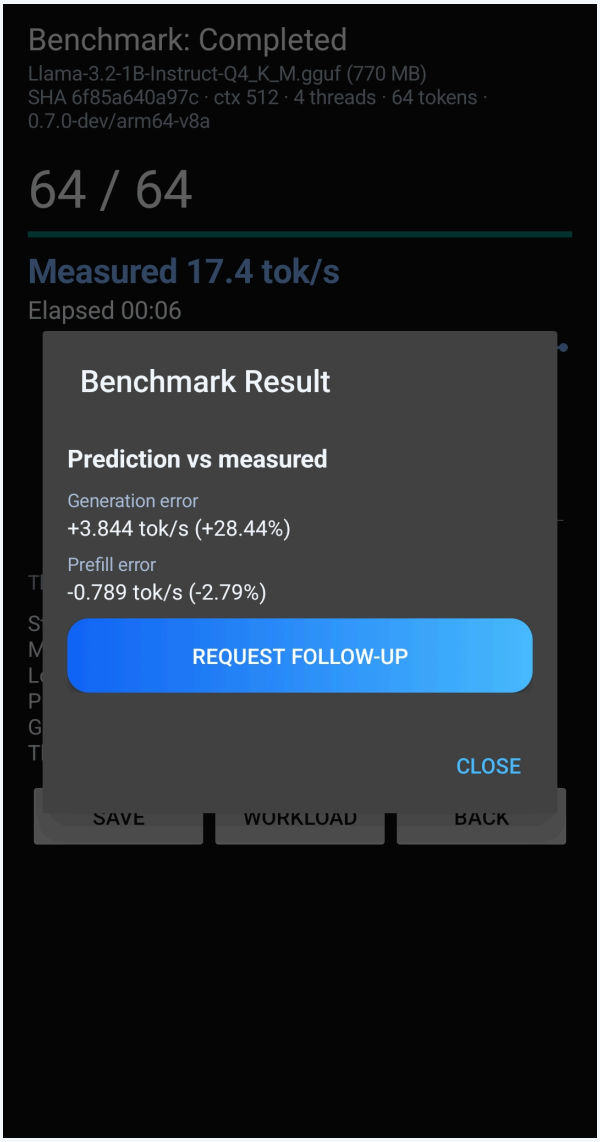
Figure 28 · Screenshot · edgefit-ss-12-session-history.png




PLACED 29 edgefit-ss-25-sessions-repeat-series-compare.png

Figure 29: Repeat series: reference run compared against comparable runs with measured deltas.

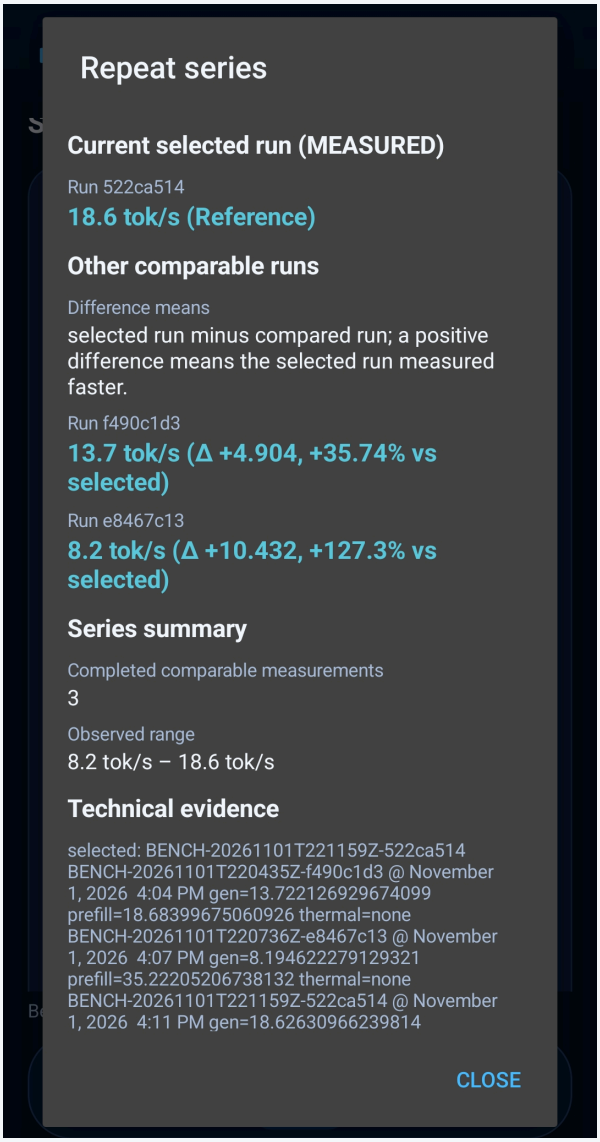
Figure 29 · Screenshot · edgefit-ss-25-sessions-repeat-series-compare.png




PLACED 30 edgefit-ss-11-predicted-vs-measured.png

Figure 30: Prediction vs measured: generation error +3.844 tok/s (+28.44%), prefill error -0.789 tok/s (-2.79%).

Figure 30 · Screenshot · edgefit-ss-11-predicted-vs-measured.png




PLACED 31 edgefit-ss-11-compare-measured.png

Figure 31: Measured-only comparison across runs.

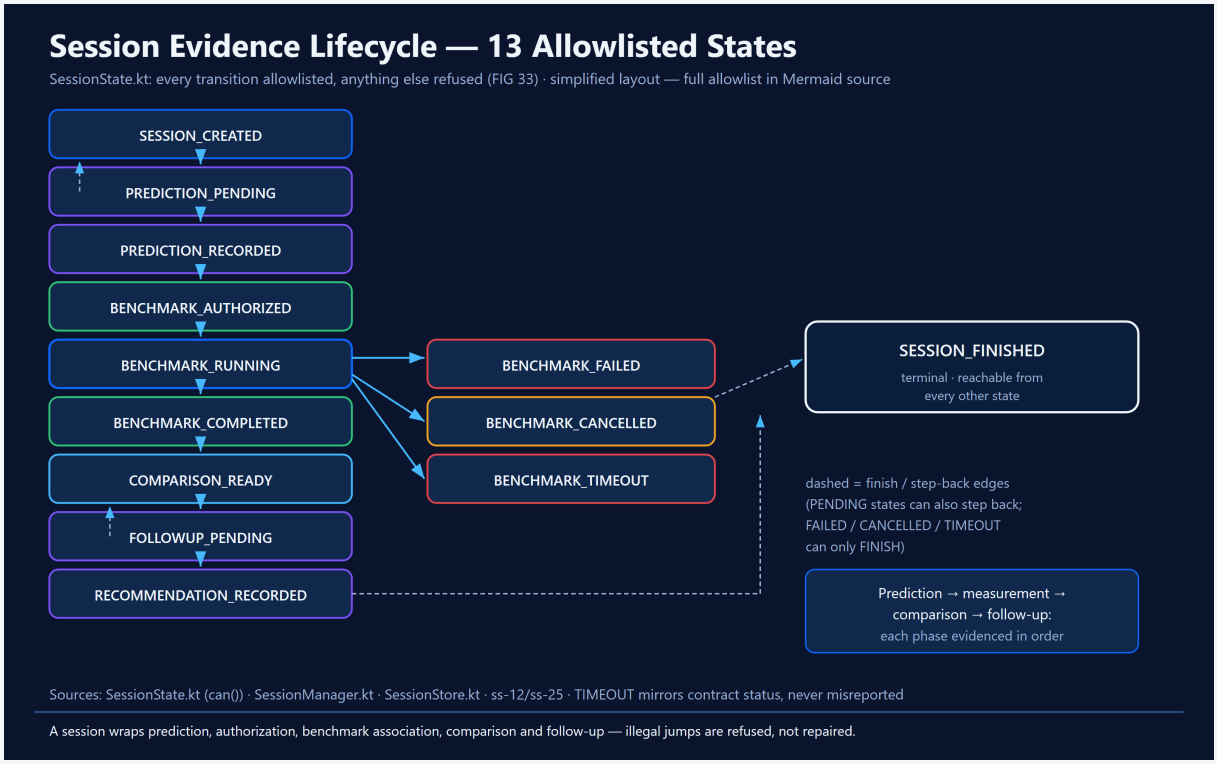
Figure 31 · Screenshot · edgefit-ss-11-compare-measured.png

**INSERT SCREENSHOT HERE**

□ 32 edgefit-ss-36-export-json-evidence.png

Evidence export or system picker; pending.

Figure 32 · Screenshot · edgefit-ss-36-export-json-evidence.png




PLACED 33 edgefit-fig-33-session-lifecycle.png

Figure 33: Session state machine: 13 allowlisted states from SessionState.kt; full allowlist in Mermaid source.

Figure 33 · Diagram · Session state machine and evidence chain of custody

## Under the hood — implementation and control flow

SessionStore and SessionManager persist records and enforce authorization and ordering. The review mapped 13 session states spanning creation, prediction pending/recorded, benchmark authorized/running/completed/failed/cancelled/timeout, and follow-up/finalization. Enumerate exact state names from SessionState.kt when publishing diagrams.

SessionComparison computes prediction-versus-measured error on the host; the model does not judge itself. The identified experiment exp-4 showed generation prediction error +1.36% and prefill error +14.8%, distinct metrics that must never be collapsed.

SessionJson and ProfileSerializer implement contract-bound export through Android picker flows. Python validators and the additive workstation SQLite archive support external inspection. Local record deletion is a separate operation from global withdrawal.

## States, edge cases and refusal behavior

Prediction must precede the measured run to form a legitimate comparison.

FAILED/CANCELLED/TIMEOUT cannot be silently converted to completed studies.

Session histories persist within Android package identity; debug and benchmark variants have distinct app-private stores.

## Evidence checkpoint

SessionStore.kt; SessionManager.kt; SessionView.kt; SessionTimeline.kt; SessionComparison.kt; SessionJson.kt.

Six-run study manifests and score documents; S22 export validation cited in audit.

# 07 Meta Muse Spark — Prediction, Evidence and Agentic Experiments

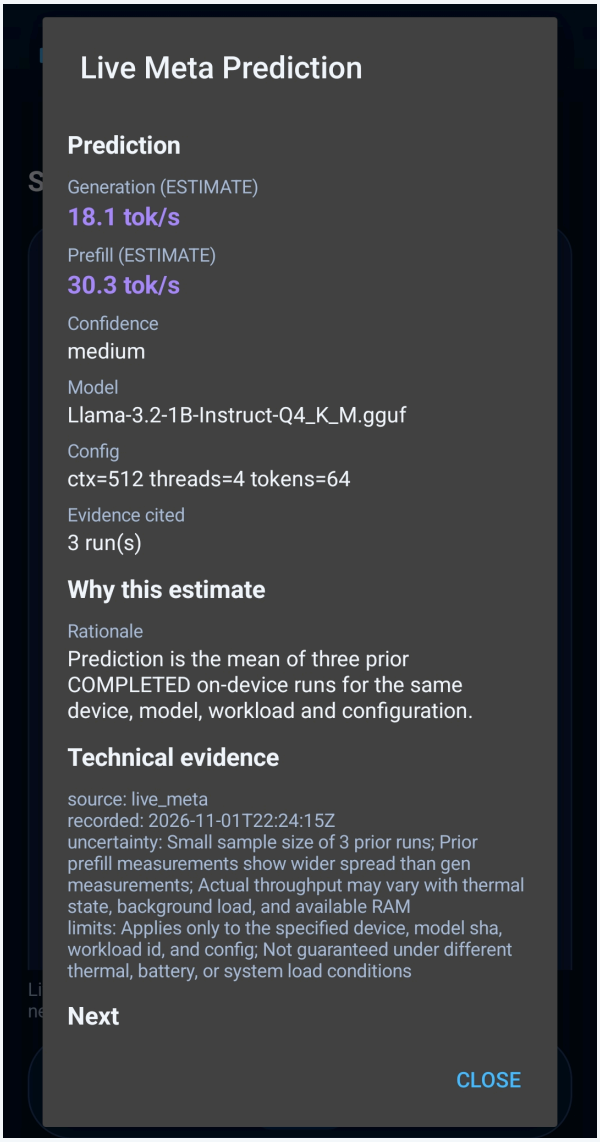
This mission separates AI assistance from truth. Muse Spark can estimate expected performance and propose a next experiment; only the Android runtime can measure actual performance. A bounded, owner-controlled loop makes the system auditable.

## Mission briefing

This mission separates AI assistance from truth. Muse Spark can estimate expected performance and propose a next experiment; only the Android runtime can measure actual performance. A bounded, owner-controlled loop makes the system auditable.

## Walkthrough — what a judge sees and does

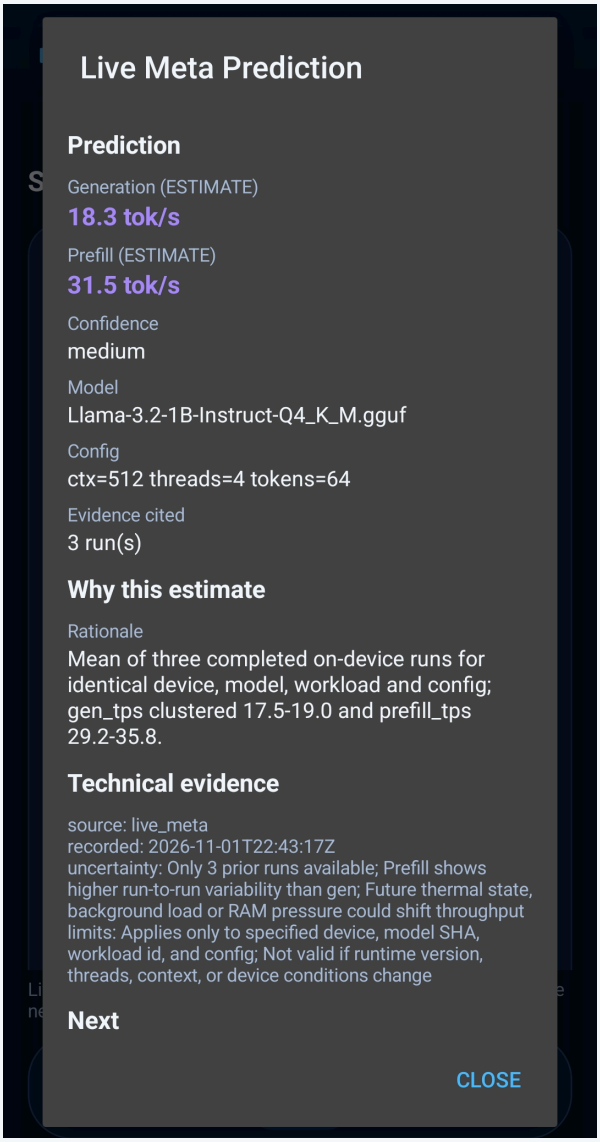
1. Go to Settings → Meta Muse and configure an explicitly authorized API key.
2. Open Agent; inspect session budget and any pending-review badge.
3. Request or review pre-run prediction with expected performance and uncertainty.
4. Open a proposal, read purpose/config change and supporting evidence IDs.
5. Approve one authorized experiment, or choose Modify/Reject/Finish where available.
6. Run the real benchmark; inspect a cited follow-up and whether conclusions are supported.




PLACED 34 edgefit-ss-13-muse-spark-prediction.png

Figure 34: Live Meta prediction: 18.1 / 30.3 tok/s ESTIMATES, medium confidence, 3 runs cited, uncertainty stated.

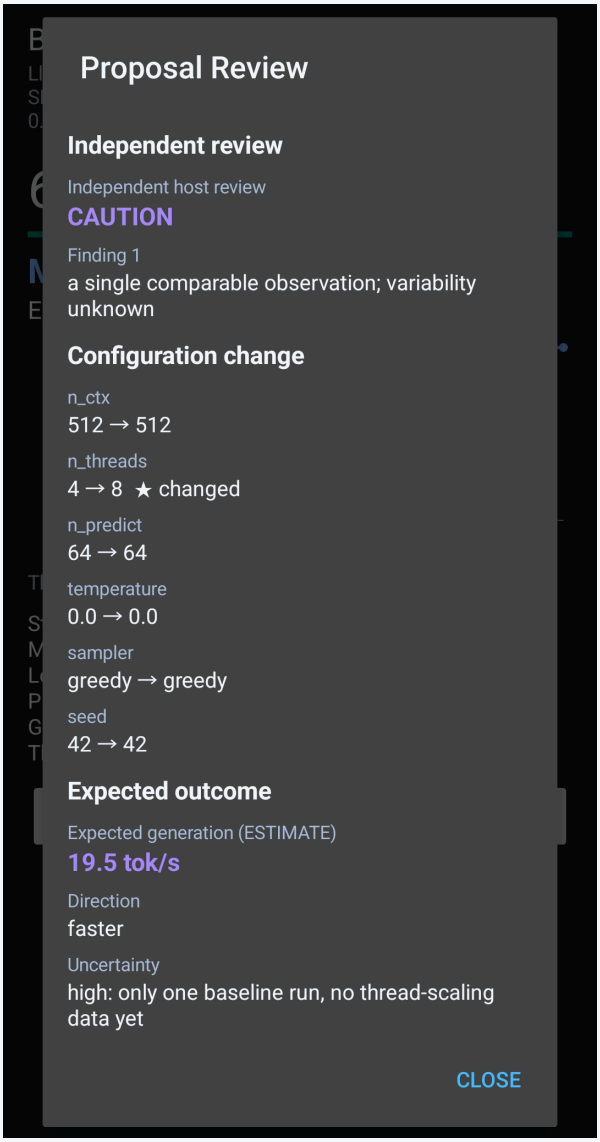
Figure 34 · Screenshot · edgefit-ss-13-muse-spark-prediction.png




PLACED 35 edgefit-ss-21-agent-live-prediction.png

Figure 35: Live prediction output with rationale and live_meta technical evidence.

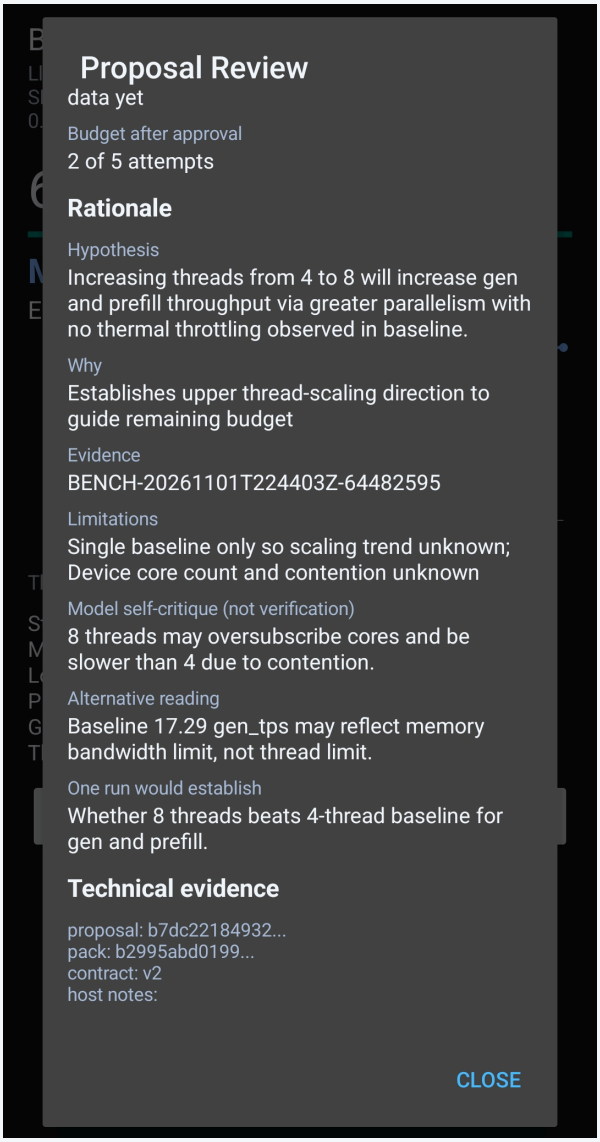
Figure 35 · Screenshot · edgefit-ss-21-agent-live-prediction.png




PLACED 36 edgefit-ss-22-agent-proposal-review.png

Figure 36: Proposal review: independent host CAUTION, 4 to 8 thread change, expected 19.5 tok/s (ESTIMATE).

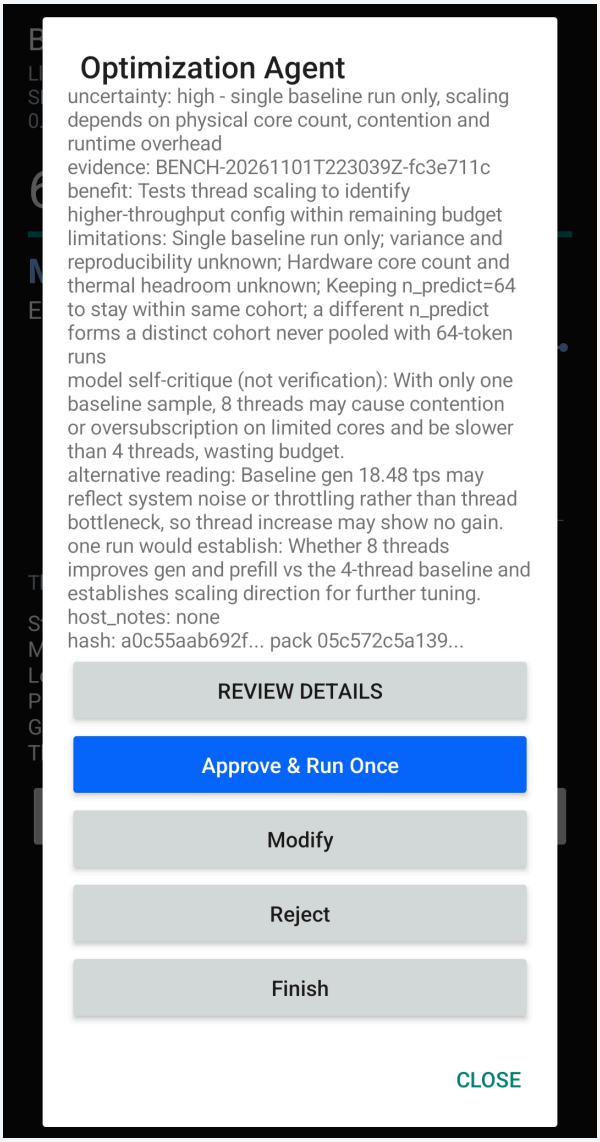
Figure 36 · Screenshot · edgefit-ss-22-agent-proposal-review.png




PLACED 37 edgefit-ss-23-agent-proposal-review-02-rationale.png

Figure 37: Proposal rationale: hypothesis, cited BENCH evidence, limitations and model self-critique.

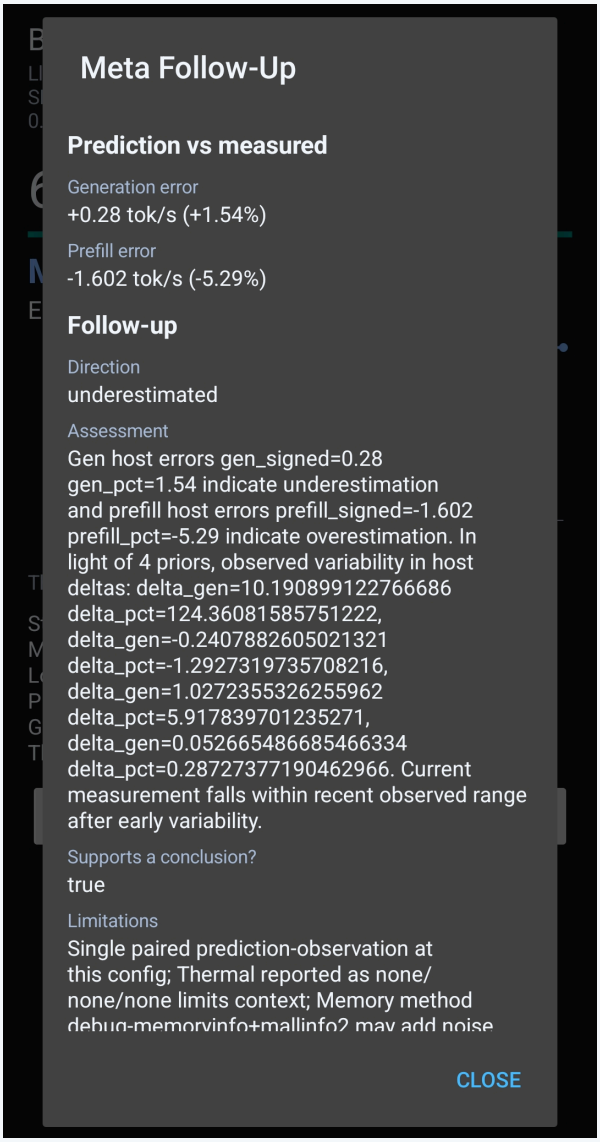
Figure 37 · Screenshot · edgefit-ss-23-agent-proposal-review-02-rationale.png




PLACED 38 edgefit-ss-18-agent-optimization-approval.png

Figure 38: Optimization approval gate: review details, Approve and Run Once, Modify, Reject, Finish.

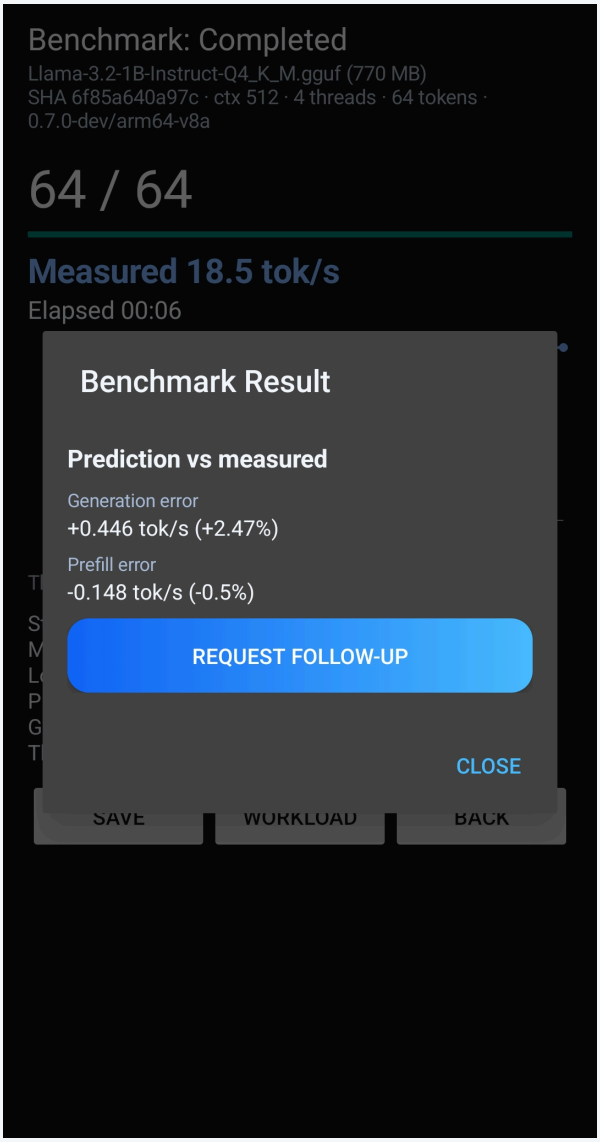
Figure 38 · Screenshot · edgefit-ss-18-agent-optimization-approval.png




PLACED 39 edgefit-ss-15-agent-follow-up-detail.png

Figure 39: Follow-up interpretation: underestimated direction, host-error assessment and stated limitations.

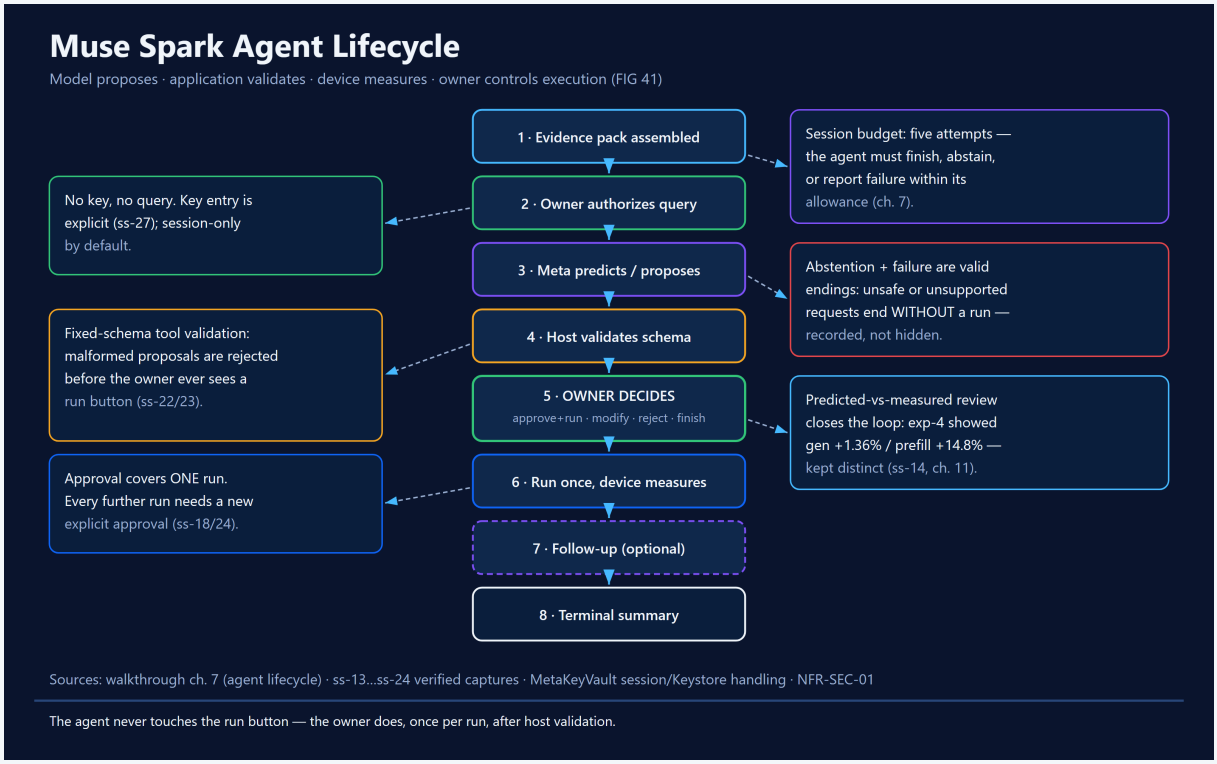
Figure 39 · Screenshot · edgefit-ss-15-agent-follow-up-detail.png




PLACED 40 edgefit-ss-16-agent-follow-up-request.png

Figure 40: Follow-up request entry from the prediction-vs-measured dialog.

Figure 40 · Screenshot · edgefit-ss-16-agent-follow-up-request.png




PLACED 41 edgefit-fig-41-agent-lifecycle.png

Figure 41: Muse Spark bounded agent lifecycle: five-attempt budget; model proposes, host validates, owner approves each run.

Figure 41 · Flowchart · Muse Spark bounded agent and tool-call lifecycle

## Under the hood — implementation and control flow

MetaSessionClient/MetaApi/MetaHttp coordinate BYOK requests. A prediction is stored before the corresponding benchmark; a later follow-up may discuss evidence by cited run ID, including an explicit refusal to draw a conclusion when the supporting data are insufficient.

AgentManager, AgentEvidence, AgentReview and AgentApi validate proposals using typed schemas, evidence identifiers, permissible config change and attempt budgets. The agent does not execute arbitrary shell commands, fetch secrets, or self-authorize runs.

The reported agent session budget is five attempts. User actions include Approve, Modify, Reject and Finish, but the audit distinguished physically verified Approve/Finish from host-tested Reject/Modify. Repeated improvements and stop conditions must be described strictly within their implemented scope.

In the M0/M1 study arms, Muse correctly abstained where candidate-comparable evidence was insufficient. The audit cites zero fabricated citations for the scoped run; this is not an across-all-prompts guarantee.

## States, edge cases and refusal behavior

No key or expired key: prediction is unavailable; local benchmarking can continue.

Malformed, unsupported or unevidenced proposal: host validation rejects it.

No comparable evidence: abstain rather than fabricate run citations.

Attempt budget and invalid-proposal strikes enforce a bounded loop.

## Evidence checkpoint

MetaSessionClient.kt; MetaApi.kt; MetaHttp.kt; AgentManager.kt; AgentEvidence.kt; AgentReview.kt; AgentApi.kt.

EF-COMP-01B study M0/M1 recorded abstentions; exp-4 measured and predicted record scoped.

# 08 Security, Privacy and Validator Boundaries

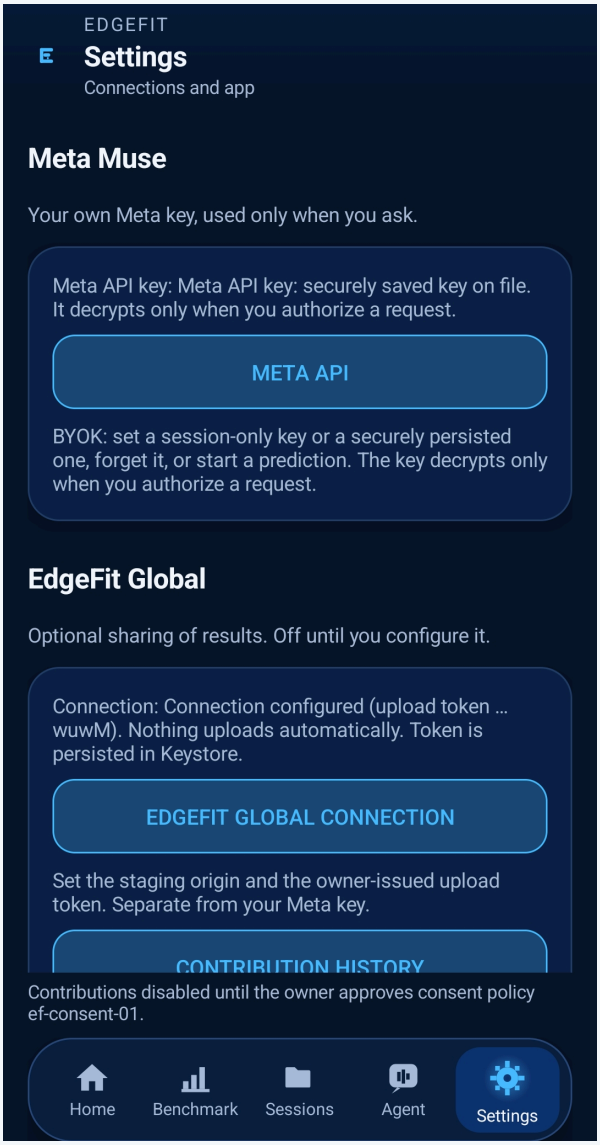
EdgeFit makes explicit security distinctions: an API key is not a benchmark artifact; a user-approved run is not permission to upload evidence; and an AI text response is not a trusted database query.

## Mission briefing

EdgeFit makes explicit security distinctions: an API key is not a benchmark artifact; a user-approved run is not permission to upload evidence; and an AI text response is not a trusted database query.

## Walkthrough — what a judge sees and does

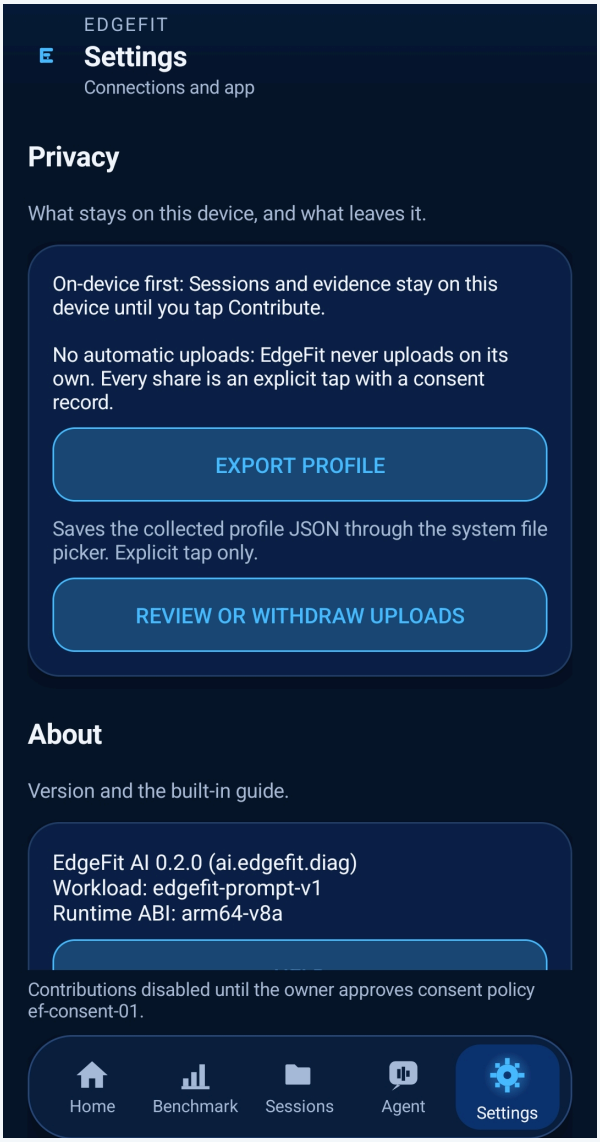
1. Inspect Meta settings without exposing a real API key.
2. Read whether key custody is session-only or persisted; test Forget only if owner authorizes.
3. Inspect Privacy controls and Global connection settings separately.
4. Study the tool-validator diagram to understand why generated model text cannot issue arbitrary commands.




PLACED 42 edgefit-ss-27-settings-meta-muse.png

Figure 42: Meta Muse settings: BYOK key custody and Global connection state (token suffix only; credential revoked).

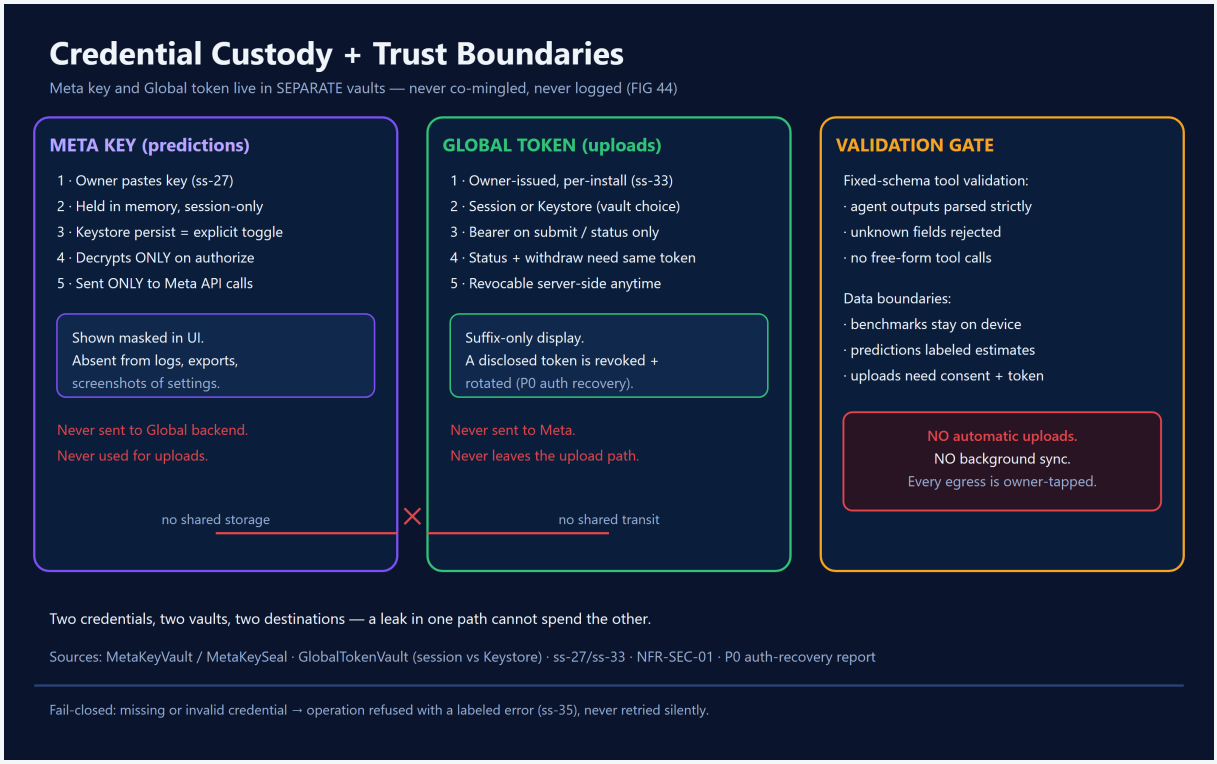
Figure 42 · Screenshot · edgefit-ss-27-settings-meta-muse.png




PLACED 43 edgefit-ss-28-settings-privacy.png

Figure 43: Privacy settings: on-device-first, no automatic uploads, export profile, review/withdraw.

Figure 43 · Screenshot · edgefit-ss-28-settings-privacy.png




PLACED 44 edgefit-fig-44-credential-custody.png

Figure 44: Credential custody: Meta key and Global token in separate vaults; fixed-schema validator trust boundary.

Figure 44 · Diagram · Credential custody and deterministic validator trust boundary

## Under the hood — implementation and control flow

MetaKeyVault supports session-only custody by default, with optional encrypted persistence through a Keystore-backed seal. The audit identified AES/GCM (12-byte IV), app-private file handling, suffix-only display and explicit forget. Confirm final operating details against the present implementation prior to release.

GlobalTokenVault handles the distinct per-install contribution token; do not mix this with the Muse API key. Credential material must be excluded from screenshots, exports, logs and public documentation.

Host validators (ef_dispatch_05.py, ef_dispatch_06.py) constrain tool names and exact fields before evidence access, pin device/model identities and use read-only retrieval. Named rejection classes include UNAUTHORIZED_TOOL, MALFORMED_JSON, INVALID_DEVICE, INVALID_MODEL and limits/type errors.

Publication consent is separate from key possession and benchmark authorization. Consent policy and approval record were PREPARED/PROPOSED in the audit; never represent blanks as signed authorizations.

## States, edge cases and refusal behavior

Key seal invalidation requires safe re-entry; do not show secret contents.

Out-of-schema tool calls are rejected before database reads.

No consent or missing origin/token means no authorized automatic global sharing.

## Evidence checkpoint

MetaKeyVault.kt; MetaKeySeal.kt; GlobalTokenVault.kt; scripts/ef_dispatch_05.py; scripts/ef_dispatch_06.py.

EF-META-04 validator test record: 12 named checks in the reviewed script.

# 09 Global Evidence Network — From One Device to Qualified Public Statistics

The Global Evidence Network is intended to pool defensible, privacy-controlled benchmark evidence across installations. The cloud and website are real project components, but existence of an endpoint is not proof that genuine measurements were published or qualified.

## Mission briefing

The Global Evidence Network is intended to pool defensible, privacy-controlled benchmark evidence across installations. The cloud and website are real project components, but existence of an endpoint is not proof that genuine measurements were published or qualified.

## Walkthrough — what a judge sees and does

. Open the EdgeFit Global entry point or Settings group.

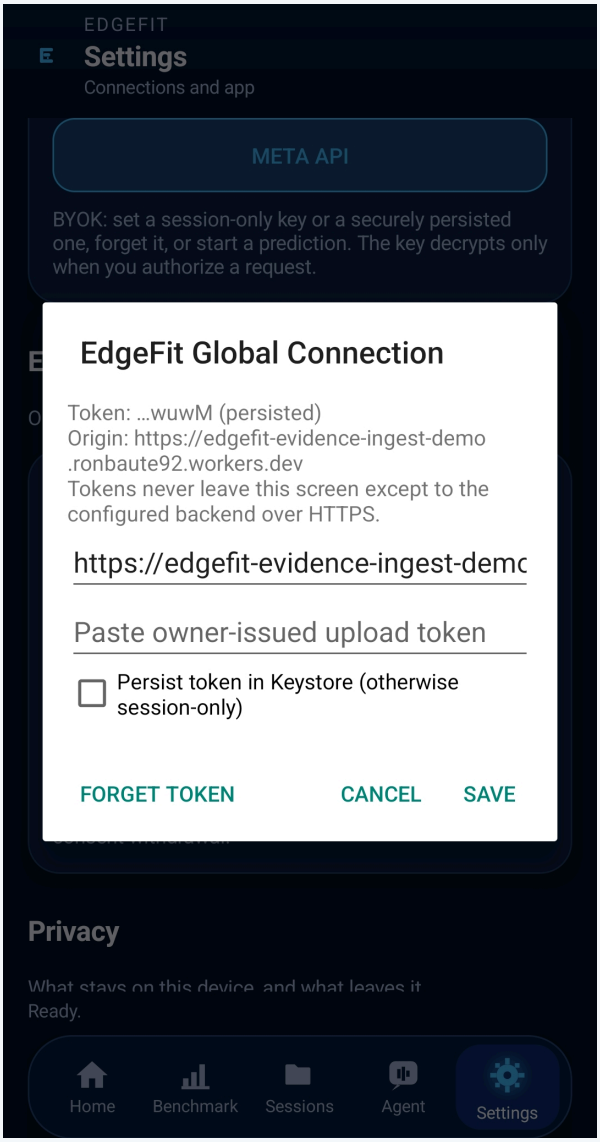
. Configure only approved HTTPS origin and per-install token if needed.

. Open review for a completed local run and inspect provenance/hash and the exact scopes.

. Do not submit for demonstration unless explicitly authorized by the owner and current consent policy.

. Inspect history, status refresh, possible uncertain-submission retry and Withdraw.

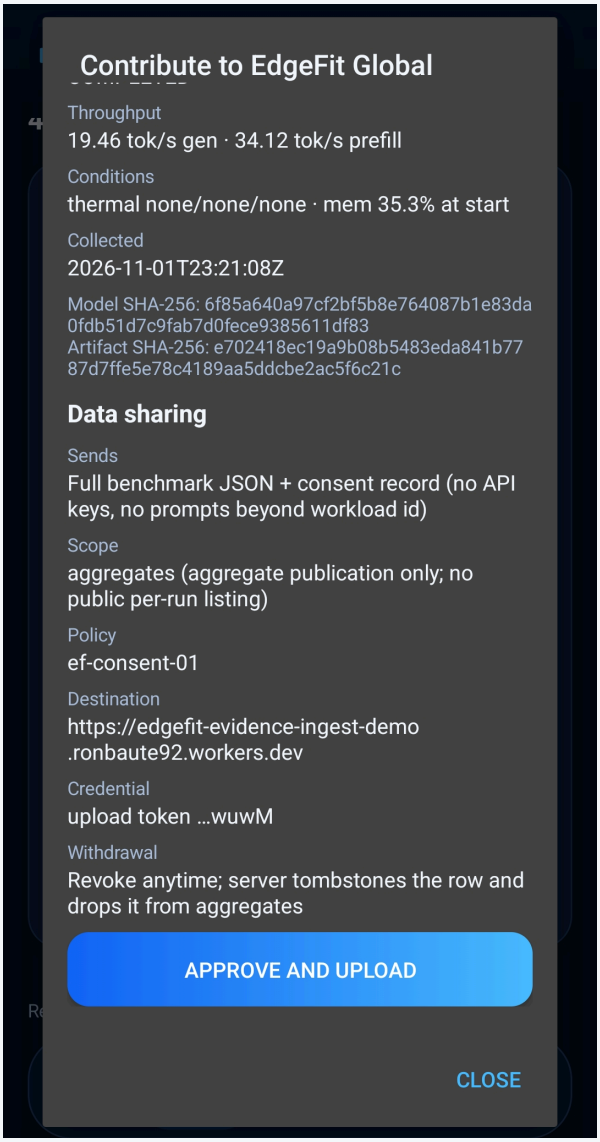
. Open the public dashboard; distinguish deployed empty status, submitted data and fully qualified published cohorts.




PLACED 45 edgefit-ss-33-global-connection-configuration.png

Figure 45: Global connection configuration: ingest origin, owner-issued token entry, Keystore persistence option.

Figure 45 · Screenshot · edgefit-ss-33-global-connection-configuration.png




PLACED 46 edgefit-ss-34-contribution-approve-upload.png

Figure 46: Contribution consent review: measured throughput, conditions, ef-consent-01, aggregates scope, explicit APPROVE AND UPLOAD. Pre-upload review, not a receipt.

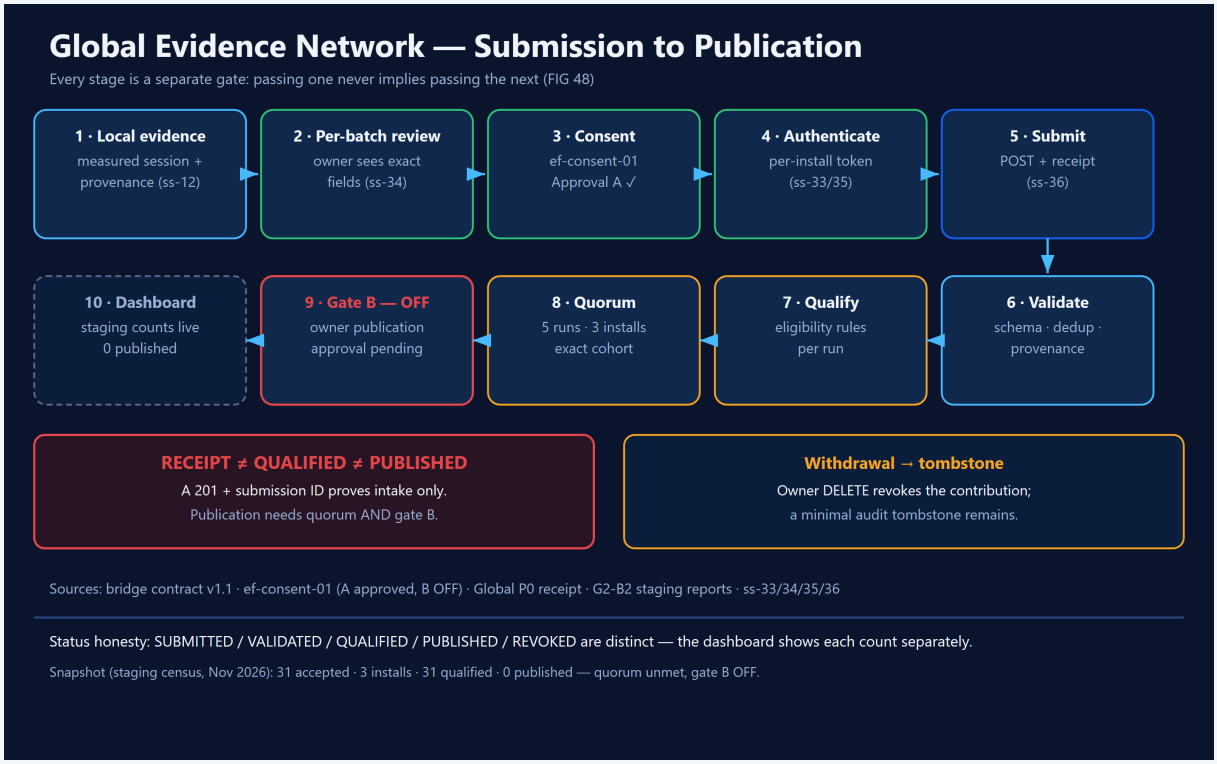
Figure 46 · Screenshot · edgefit-ss-34-contribution-approve-upload.png

**INSERT SCREENSHOT HERE**

□ 47 edgefit-ss-35-contribution-history.png

Pending history/withdraw capture.

Figure 47 · Screenshot · edgefit-ss-35-contribution-history.png




PLACED 48 edgefit-fig-48-global-network.png

Figure 48: Global evidence pipeline: consent, ingestion, quorum, publication gate and revocation. Receipt is not qualified is not published.

Figure 48 · Diagram · Consent, ingestion, quorum, aggregate publication and revocation

## Under the hood — implementation and control flow

The Android contribution flow is review → consent → submit → status/history → optional withdrawal. GlobalClient and GlobalView present run provenance, artifact hash, policy scope and token status before the user submits. The first genuine Android-to-backend publication was unverified in the audit snapshot.

The backend architecture is Cloudflare Workers with D1; contracts and migrations from the separate Global branch govern validation, quotas, deduplication, revocation and publication. Cohort qualification required at least five qualifying runs from three distinct installations (5/3) within an exact cohort; production publication was additionally gated by an empty allowlist during the audit.

Only aggregated, qualified values should appear in public statistics. A submission may be SUBMITTED without being PUBLISHED. A browser showing HTTP 200 with empty artifacts is not a success claim for real user evidence. As of the audited report, the deployed explorer was honestly empty.

GlobalClient failure types include shared, rejected, authorization failure, quota, uncertain and blocked. On ambiguous transmission outcomes, an explicit retry uses the same bytes/content hash rather than minting a false new run. Local deletion and cloud withdrawal have separate effects; withdrawal retains minimum audit shells under policy.

## States, edge cases and refusal behavior

Missing origin/token or consent: upload remains off.

Auth failure, quota, validation error and uncertainty must display distinct outcomes.

Server publication is withheld until qualification and owner-controlled allowlist rules are satisfied.

Revoking cloud evidence does not automatically delete local session records.

## Evidence checkpoint

GlobalClient.kt; GlobalView.kt; GlobalContract.kt; GlobalTokenVault.kt.

Global branch cba27c3 and website branch 8d41a61 were audited as separate snapshots.

Audit notes 297/297 backend tests as a handoff record; current production status must be refreshed.

### Judge interpretation

If newer workstreams have published actual cohorts since this audit, replace the historical empty-state wording only after independent API/database verification and documented approval.

# 10 Safe Installation, Data Retention and Recovery

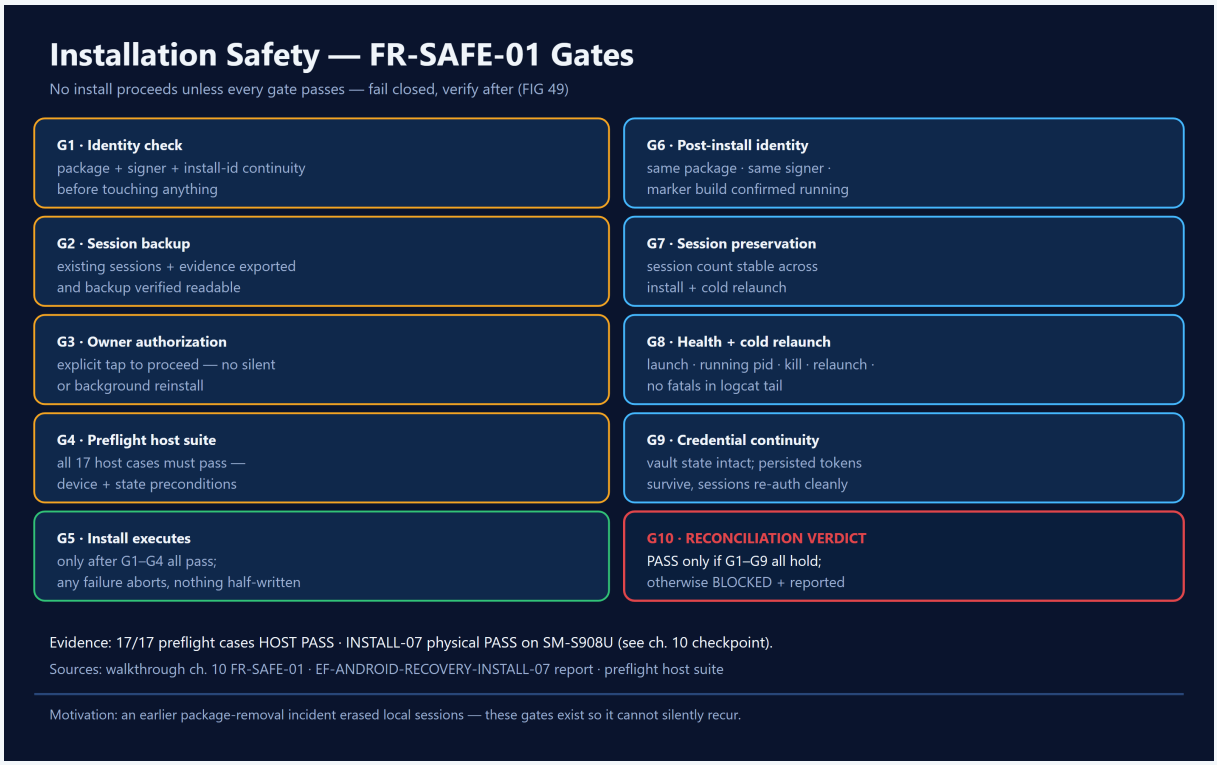
An install-safety incident revealed how easily app-private evidence can become inaccessible when a different Android package ID is installed. The safety rules exist so documentation and demonstrations never sacrifice real measurements.

### Mission briefing

An install-safety incident revealed how easily app-private evidence can become inaccessible when a different Android package ID is installed. The safety rules exist so documentation and demonstrations never sacrifice real measurements.

### Walkthrough — what a judge sees and does

1. Read current package identity and current measured-session count.
2. Consult preflight and recoverability records before any upgrade or install.
3. Never use this documentation as approval for ADB install, uninstall, data clear, or app-signing replacement.
4. Inspect recovery/quarantine UI only when present in a safe existing installation.




PLACED 49 edgefit-fig-49-install-safety.png

Figure 49: FR-SAFE-01 ten-gate installation safety board. 17/17 preflight HOST PASS; INSTALL-07 physical PASS.

Figure 49 · Flowchart · FR-SAFE-01 ten-gate installation safety board

**INSERT SCREENSHOT HERE**

□ 50 edgefit-ss-37-error-validation-state.png

If available, show real fail-closed or quarantine warning.

Figure 50 · Screenshot · edgefit-ss-37-error-validation-state.png

## Under the hood — implementation and control flow

Android isolates storage under each application ID. The benchmark package ai.edgefit.diag.bench differs from the base debug ai.edgefit.diag; a newly installed package may show an empty session view even if older records remain under another app identity.

FR-SAFE-01 mandates preflight checks before destructive or identity-changing APK actions. The reviewed material covers device/user/package identity, compatible signing, installed-package census, session inventory, recoverable backups, explicit owner authorization, post-install checks and fail-closed behavior.

Recovery markers and quarantine logic exist to surface questionable evidence safely rather than silently rewriting it. Historical reports include source-level recovery commits and host tests; they do not authorize a reinstall in this guide.

## States, edge cases and refusal behavior

A seemingly empty ledger can reflect the wrong package, not deletion.

Unverified backup or signing conflict: stop rather than proceed.

A recovered or imported artifact must retain honest provenance labels.

## Evidence checkpoint

docs/ef-safe-install-01.md; scripts/Confirm-EFSafeInstall01.ps1 (reported uncommitted at audit); install recovery branch milestones.

17/17 host preflight cases and INSTALL-07 physical PASS were cited with differing verification scopes.

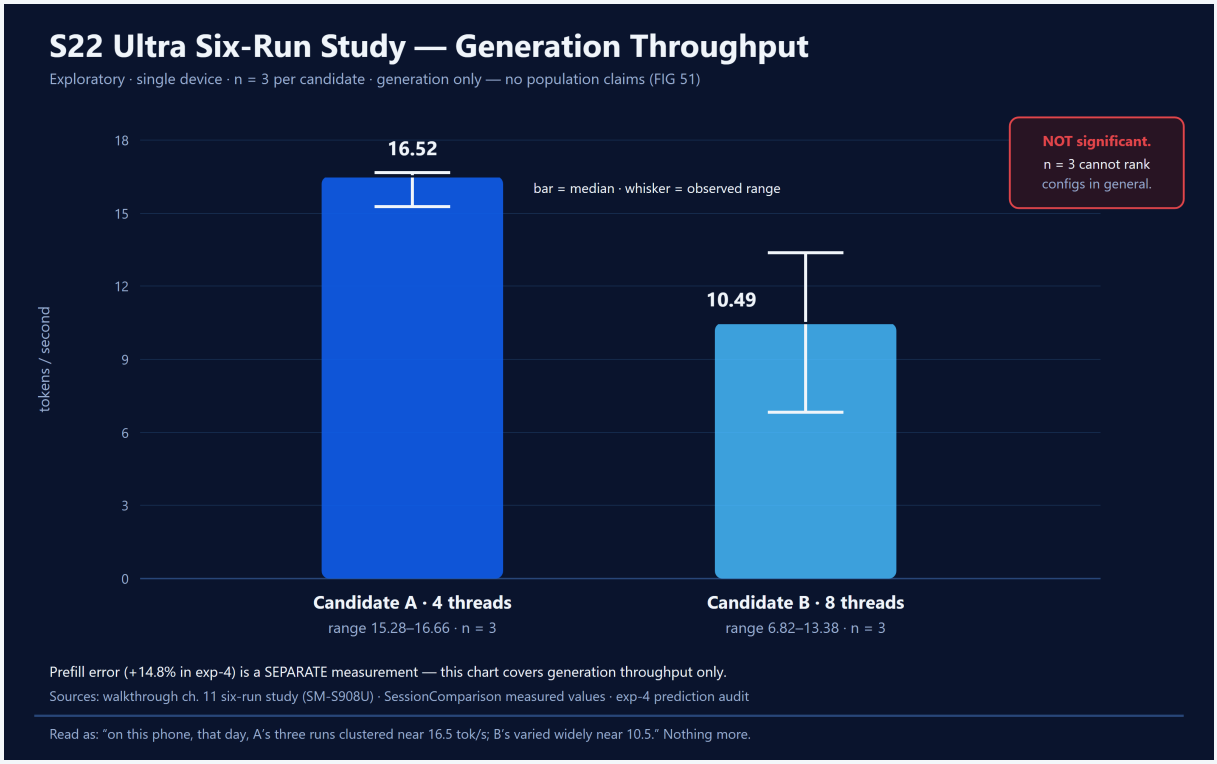
# 11 Scientific Methodology — What the Numbers Actually Prove

Every performance statement in this chapter is narrowly scoped to the documented S22 Ultra study, exact GGUF and workload. The purpose is to show reproducible measurement habits and to test assumptions—not to announce universal Android performance rankings.

## Controlled six-run CPU thread comparison

| Factor | Candidate A | Candidate B |
|---|---|---|
| **Phone** | Samsung S22 Ultra / Android 16 | Same phone |
| **Context** | 512 | 512 |
| **CPU threads** | 4 | 8 |
| **Predict tokens** | 64 | 64 |
| **Sample size** | 3 | 3 |
| **Median generation tok/s** | 16.52 | 10.49 |
| **Observed generation range** | **15.28–16.66** | **6.82–13.38** |

Study order was A–B–B–A–A–B. All observations were physically measured in one device environment. The result that four threads outperformed eight in this small study may involve scheduling, thermal behavior or other run conditions; the experiment does not isolate a universal mechanism. Source group: eval/comp-01b/ runs, ledger.jsonl, study-manifest.json, study-score.json and study-verdict.json.




PLACED 51 edgefit-fig-51-study-chart.png

Figure 51: Measured A/B generation throughput: median bars with observed ranges (A 16.52, 15.28-16.66; B 10.49, 6.82-13.38; n=3 each). Single S22 Ultra, exploratory; individual-run dots omitted, only median and range published in ch. 11.

Figure 51 · Chart · Measured A/B generation throughput with individual runs, median and ranges

## Prediction accuracy and scientific abstention

For scoped exp-4, host-computed errors were +1.36% generation and +14.8% prefill. Different speed metrics cannot be collapsed into one headline error. M0/M1 declined to infer unsupported candidate conclusions when evidence was inadequate. A reported fabricated-citation rate of 0.0 applies only to the evaluated sample.

## Provenance payload and audit checklist

Device: exact model / OS / ABI and build identity.

Artifact: model SHA-256 and byte size; do not rely on filename.

Workload: frozen prompt identifier, sampling rule, seed, context, predict count and thread count.

Conditions: battery, available memory, thermal context, timing and any warnings.

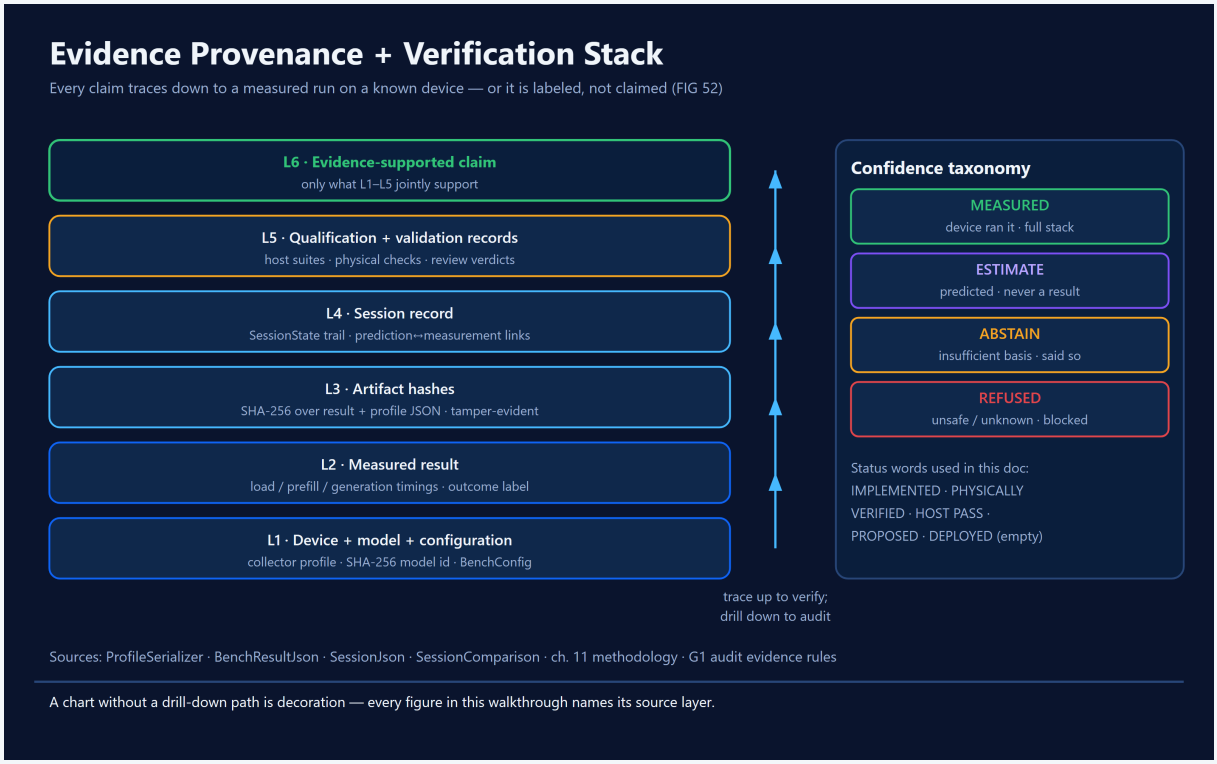
Measurement: phase counts and timing, valid terminal state, run ID, and record hashes.

Validity: measured vs predicted vs estimated vs synthetic; host-validated inclusion and explicit exclusions.

## Software verification record — dated snapshot, not a fresh run

| Suite / category | Reported result / caveat |
|---|---|
| **Android JVM** | 634/634 PASS in committed contribution-branch validation record; re-run on final build. |
| **Cloud foundation** | 38/38 PASS in audited local foundation branch. |
| **Global backend** | 297 test cases identified; 297/297 PASS per branch handoff, not independently re-run here. |
| **EF-META-04** | 12 named validation cases recorded in committed script. |
| **EF-SCIENCE-01A** | 46/46 host self-tests cited; physical trials separate. |
| **Python suites** | **META-05 14/14; ANDROID-01 25/25; DATA-02 29/29; BENCH-01 19/19 + native 3/3; DATA-03 35/35; META-** |

**06 39/39; LOOP-01 26/26; session parity 1/1; UI 14/14 — prior validation records only.**




PLACED 52 edgefit-fig-52-provenance-stack.png

Figure 52: Measurement provenance stack (L1-L6) and confidence/abstention taxonomy.

Figure 52 · Diagram · Measurement provenance stack and confidence/abstention taxonomy

# 12 Distribution, Publication Gates and Honest Limitations

No judge should confuse a locally installed development APK with a signed, broadly distributable public release. The technical submission includes code, evidence and demonstrations; the distribution channel is a separate controlled milestone.

## Mission briefing

No judge should confuse a locally installed development APK with a signed, broadly distributable public release. The technical submission includes code, evidence and demonstrations; the distribution channel is a separate controlled milestone.

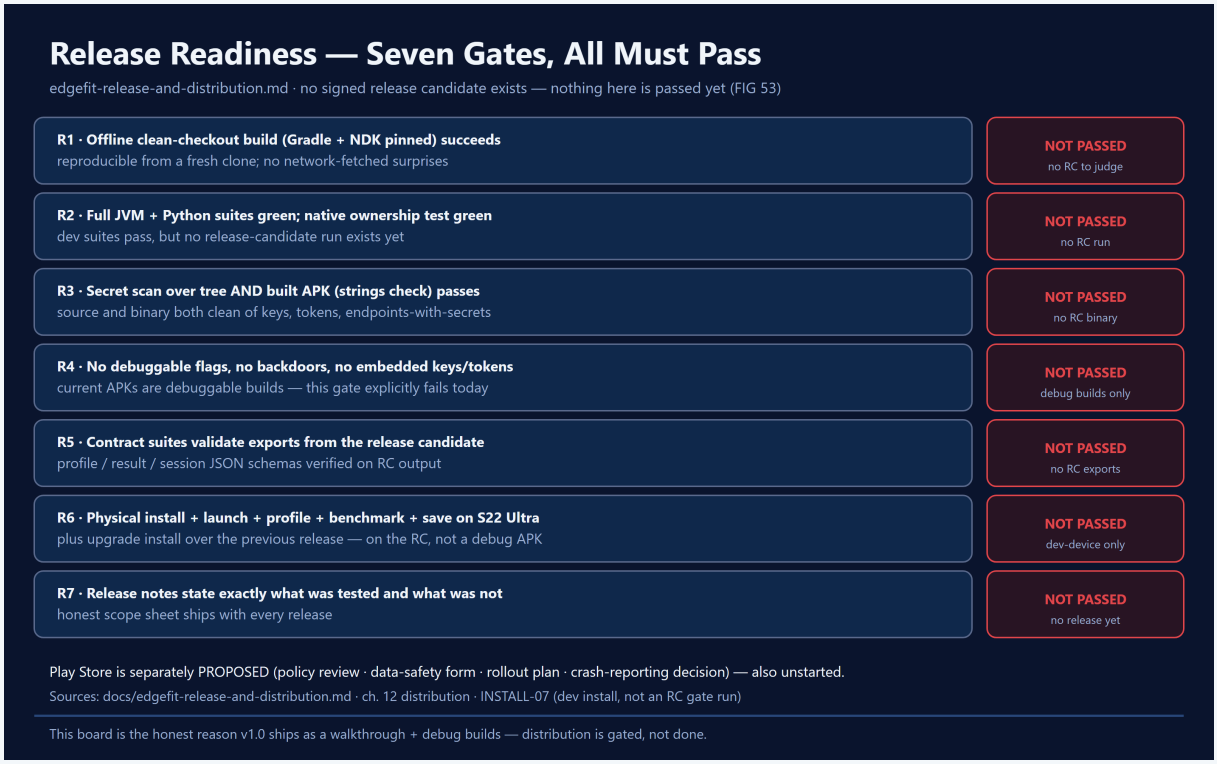
## Walkthrough — what a judge sees and does

. Start with repository documentation and judge evidence guide.

. Open the demo video and optional long-form walkthrough only when real links are provided.

. Inspect code, tests, sample artifacts and traceable measurement records.

. Treat missing signed release, absent data and unapproved policy as explicitly unavailable.




PLACED 53 edgefit-fig-53-release-gates.png

Figure 53: Public release seven-gate readiness board. All NOT PASSED: no signed release candidate exists.

Figure 53 · Flowchart · Public release seven-gate readiness board

## Under the hood — implementation and control flow

The audited release record distinguished base debug ai.edgefit.diag, benchmark ai.edgefit.diag.bench and proposed public-preview. Signing, reproducible clean-checkout build, secret scans, RC contract tests, physical upgrade and release notes must be satisfied before public distribution.

Future store availability and signed GitHub previews are plans until the release evidence exists. The demo video and public website links should be updated to their actual shipped addresses only after verification. The guide should point to the long-form technical video separately from the short hackathon demo once both exist.

Outstanding items include cross-device generalization; physically verified cloud contribution and qualification; multi-install quorum; owner consent/allowlist; final screenshot currency; and unresolved design decisions C1–C6/fleet 1–5.

## States, edge cases and refusal behavior

No signed RC means no public-preview install instruction.

Do not link nonexistent URLs or imply Play Store availability.

Cross-device dashboard claims require valid multi-install data, not fixtures.

## Evidence checkpoint

docs/edgefit-release-and-distribution.md; docs/release-readiness.md; docs/play-distribution-note.md; docs/demo.md; README and current GitHub release state.

# 13 Settings, Onboarding and Help

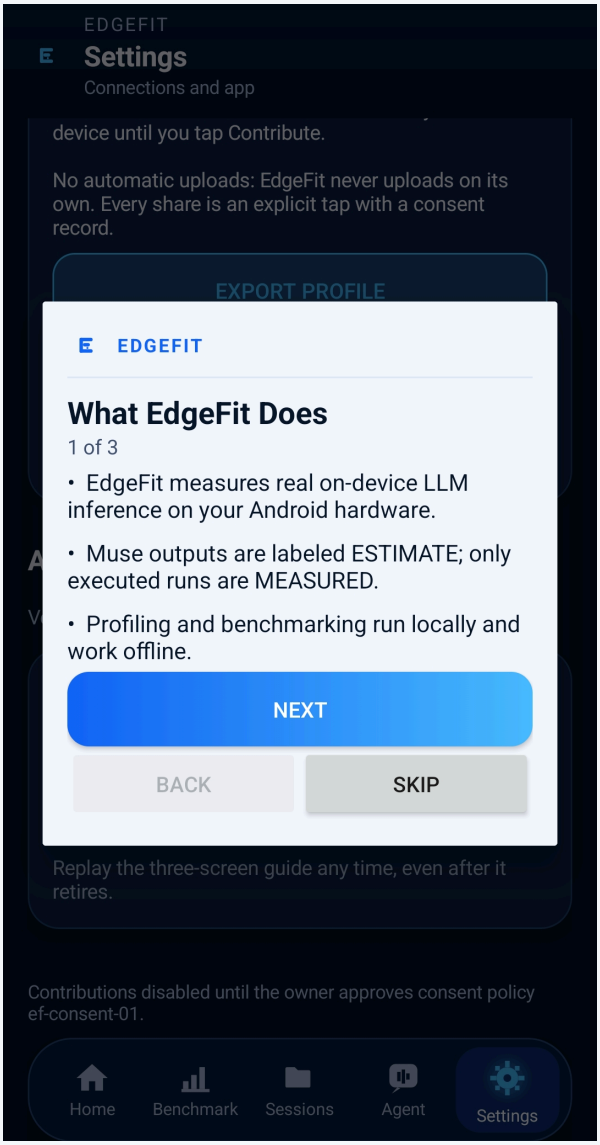
Settings is the control room for connections, privacy and application information. The first-launch guide is a separate three-screen teaching sequence, and navigation is intentionally bounded to five primary destinations.

## Mission briefing

Settings is the control room for connections, privacy and application information. The first-launch guide is a separate three-screen teaching sequence, and navigation is intentionally bounded to five primary destinations.

## Walkthrough — what a judge sees and does

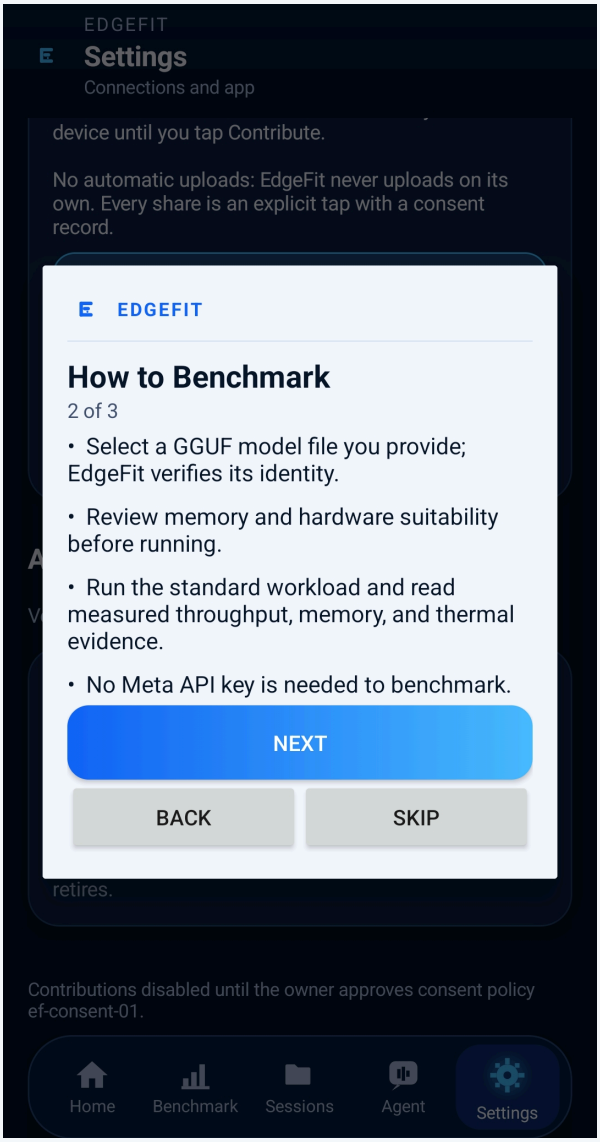
1. On first eligible launch, page through onboarding or skip.
2. Explore all five tabs; return Home and read NextAction.
3. Open Settings and inspect the six groups, not just one subsection.
4. Find benchmark-related controls, Meta key management, Global origin/token controls, Privacy and About/Help.
5. Replay Help as needed without resetting any saved sessions.




PLACED 54 edgefit-ss-29-onboarding-01.png

Figure 54: Onboarding 1 of 3: what EdgeFit measures; ESTIMATE versus MEASURED.

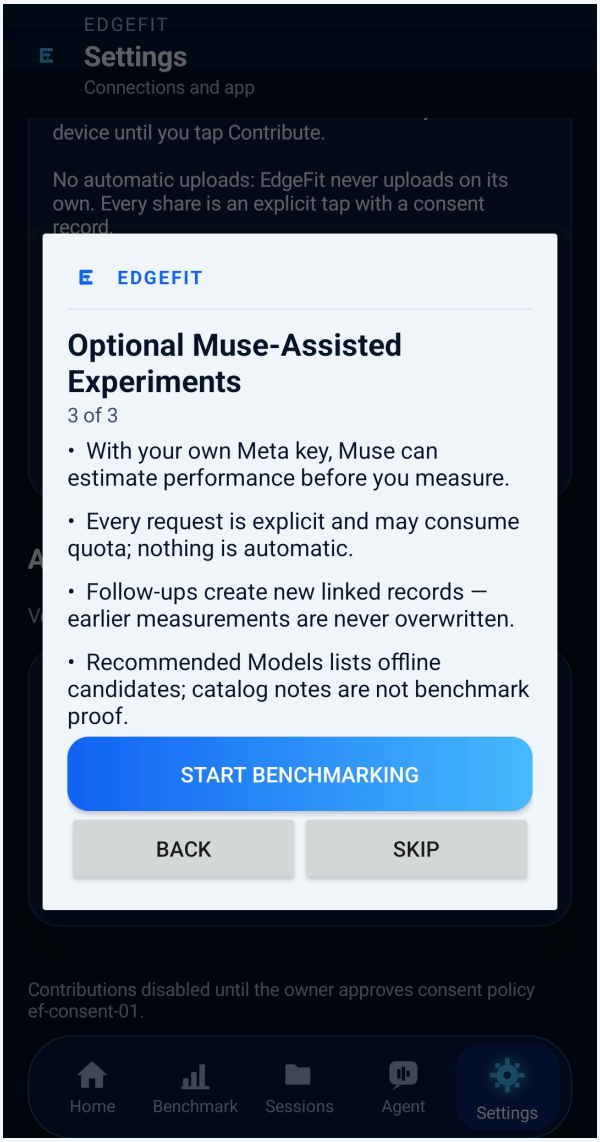
Figure 54 · Screenshot · edgefit-ss-29-onboarding-01.png




PLACED 55 edgefit-ss-30-onboarding-02.png

Figure 55: Onboarding 2 of 3: benchmark flow; no Meta key required.

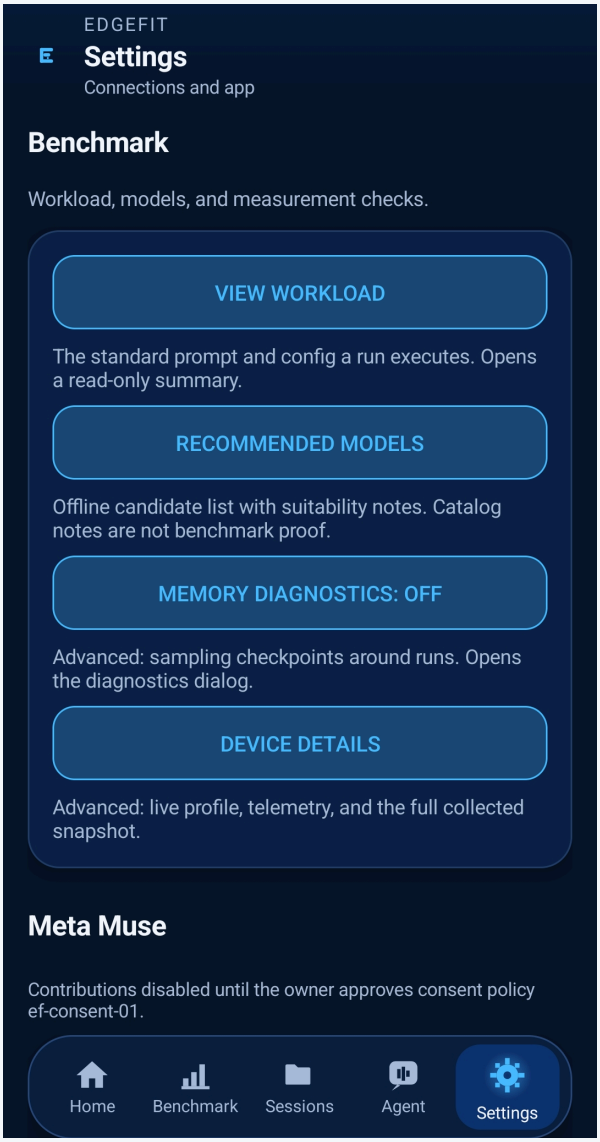
Figure 55 · Screenshot · edgefit-ss-30-onboarding-02.png




PLACED 56 edgefit-ss-31-onboarding-03.png

Figure 56: Onboarding 3 of 3: optional Muse-assisted experiments and consent posture.

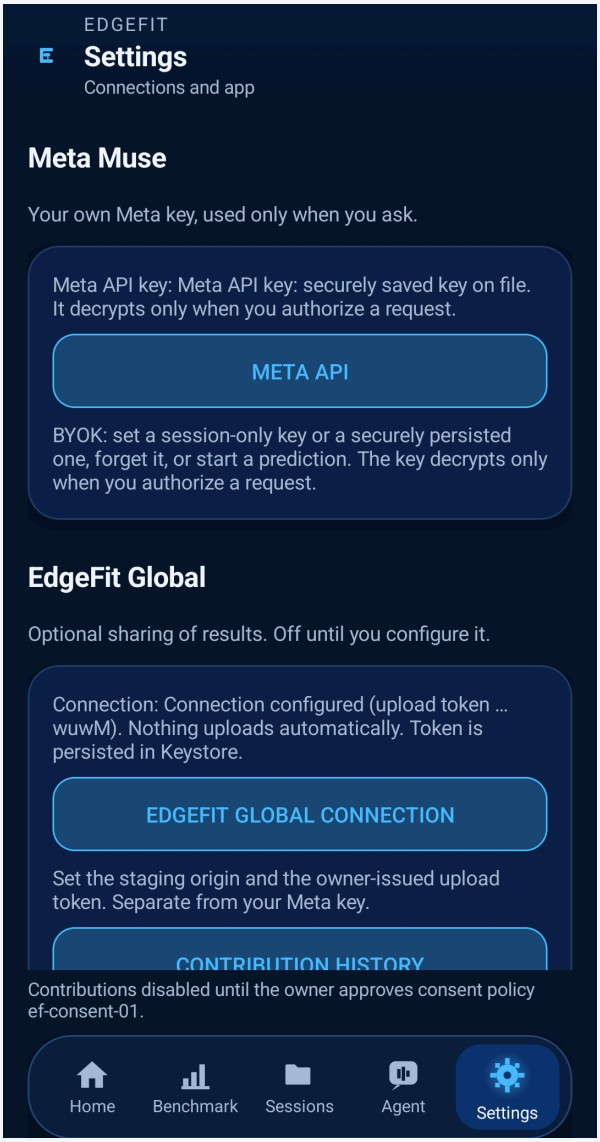
Figure 56 · Screenshot · edgefit-ss-31-onboarding-03.png




PLACED 57 edgefit-ss-26-settings-benchmark.png

Figure 57: Benchmark settings: workload, recommended models, memory diagnostics, device details.

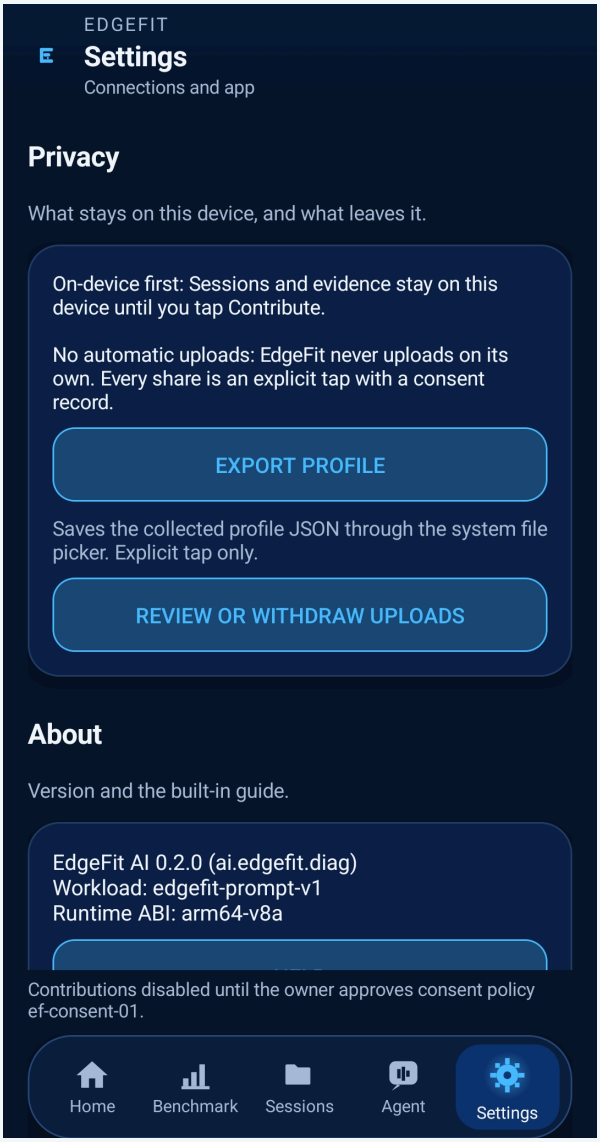
Figure 57 · Screenshot · edgefit-ss-26-settings-benchmark.png




PLACED 58 edgefit-ss-27-settings-meta-muse.png

Figure 58: Meta Muse settings: BYOK key custody and Global connection state (token suffix only; credential revoked).

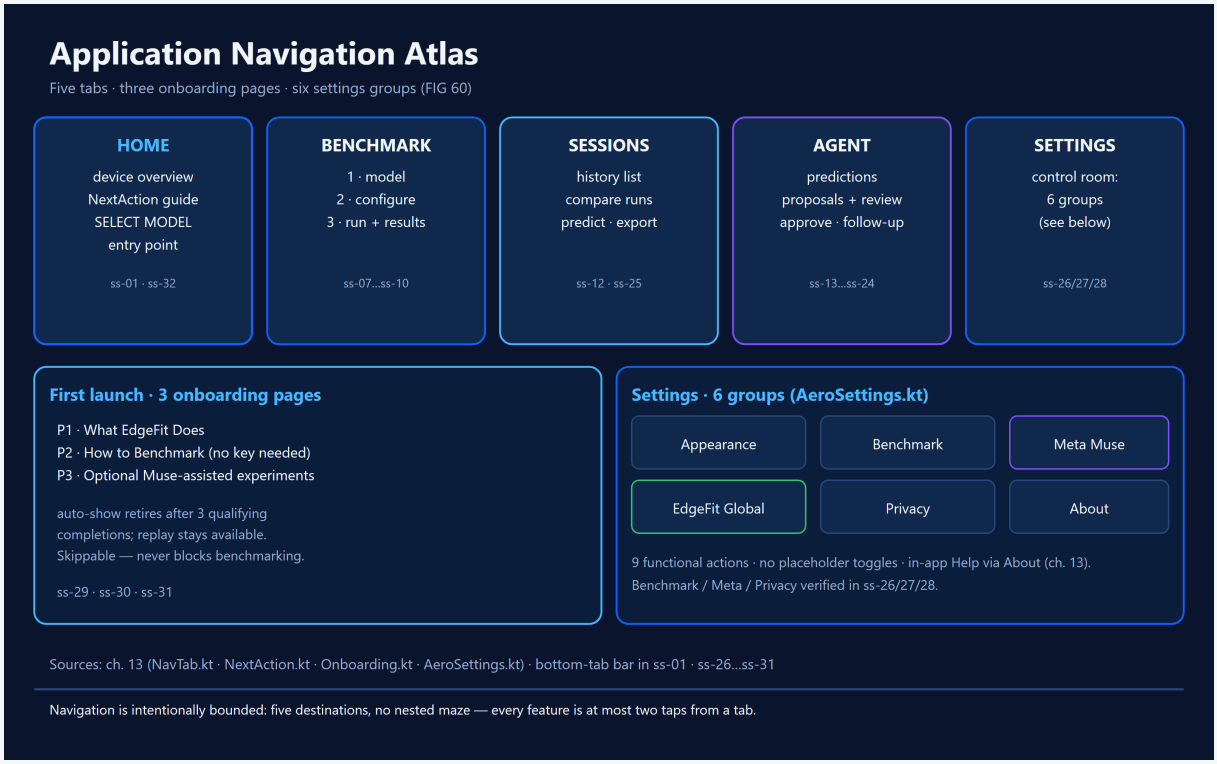
Figure 58 · Screenshot · edgefit-ss-27-settings-meta-muse.png




PLACED 59 edgefit-ss-28-settings-privacy.png

Figure 59: Privacy settings: on-device-first, no automatic uploads, export profile, review/withdraw.

Figure 59 · Screenshot · edgefit-ss-28-settings-privacy.png




PLACED 60 edgefit-fig-60-navigation-atlas.png

Figure 60: Five-tab atlas and six-settings-group map (Appearance, Benchmark, Meta Muse, EdgeFit Global, Privacy, About).

Figure 60 · Diagram · Five-tab atlas and six-settings-group map

## Under the hood — implementation and control flow

Onboarding.kt and OnboardingView.kt implement the three pages: What EdgeFit Does, How to Benchmark and Optional Muse-Assisted Experiments. The auto-show policy retires after three qualifying completions, subject to state gates and durable record rules; replay is distinct from automatic first-run display.

NavTab.kt enumerates Home, Benchmark, Sessions, Agent and Settings with descriptive subtitles and a Home back-target. A pending proposal can surface a badge on Agent. NextAction.kt derives one next action from available app state.

AeroSettings.kt implements six groups: Appearance, Benchmark, Meta Muse, EdgeFit Global, Privacy and About. The audited version reported nine functional actions, no placeholder toggles and an accessible in-app Help path. Verify exact visible action labels against the final UI capture before publication.

## States, edge cases and refusal behavior

Onboarding auto-show retirement is not the same as feature removal.

A settings row without an action is informational, not a broken toggle.

Meta key and Global contributor token are separate credentials with separate consent implications.

## Evidence checkpoint

Onboarding.kt; OnboardingView.kt; NavTab.kt; NextAction.kt; AeroSettings.kt; DOC-BUILD-02 W-series feature inventory.

# 14 Judge Quick Reference — Claim Trace and Review Questions

| Question a judge might ask | Where the documented answer lives |
|---|---|
| **Does the app perform real on-device inference?** | Chapter 5 + native bridge/runner + physically measured study artifacts. |

| | |
|---|---|
| **How does it decide which model is suitable?** | Chapter 4; deterministic eligibility vs evidence-backed observed throughput. |
| **Is the AI allowed to make up benchmark results?** | Chapter 7 + validator boundaries; AI predictions separated from host measurements. |
| **Why is the performance study scientifically meaningful?** | Chapter 11; fixed workload, source artifacts, ordered A/B, small-sample limits. |
| **How is hardware-specific performance explained?** | Chapters 3–5, model hash and device identity, telemetry and A/B. |
| **How are records kept and checked?** | Chapter 6; session store, lifecycle, exports, provenance and SQLite. |
| **Is cloud aggregation live with genuine public cohorts?** | Chapter 9; only claim values that are actually qualified and published. |
| **Can someone without Android review the app?** | Golden Path, screenshot slots, annotated figures, detailed technical sections. |
| **Can judges safely install it?** | **Chapter 12; distribution gated on signed preview, safety checks.** |

## Core reference paths (verify branch before hyperlinking)

docs/edgefit-master-architecture.md — governing system design.

docs/edgefit-product-requirements.md — feature and nonfunctional requirements.

docs/ef-arch-02-traceability.md — requirement-to-implementation map.

docs/qualification-status.md and docs/validation-results.md — audited outcomes.

docs/evidence-methodology.md and eval/comp-01b/ — scientific artifacts.

docs/ef-global-contribute-app-handoff.md and cloud contracts — cloud contribution boundaries.

docs/ef-safe-install-01.md — mandatory owner-gated install safety (uncommitted in audit).

docs/walkthrough/ — repository-target Markdown guide maintained by Muse D.

## Public-facing claims that require final refresh

Number of published cohorts, contributing devices and genuine benchmark rows.

Current Android APK variant, signing status, hash and release availability.

Latest passing test totals on merged main and exact qualification device(s).

Current website/dash live-render screenshots and public URLs.

Demo video, long-form video and technical walkthrough document URLs.

Owner consent policy and allowlist approval, with exact recorded scope.

# 15 Visual Production Ledger — editable insertion index

Each bordered square is deliberately left empty. Click inside a square or insert an image over it in Word; resize to fit the page. Preserve the original screenshots unchanged in the source archive and keep edited/annotated derivatives separately. Replace each placeholder caption with the actual image hash, capture date, app build and verified context.

This guide favors completeness over an arbitrary image limit. Several screenshots are reused for the Golden Path and their dedicated chapter; final layout can use one shared image reference or duplicate the picture where it materially improves comprehension. File names shown as pending require new captures; conceptual diagrams should be rendered from source-grounded flow definitions.

| No. | Chapter / section | Kind | Filename / illustration title |
|---|---|---|---|
| **01** | Cover | GRAPHIC | COVER HERO / EDGEFIT AERO UI COLLAGE |
| **02** | Reader compass | DIAGRAM | System overview: Android app ↔ local benchmark ↔ Muse Spark ↔ governed cloud ↔ website |
| **03** | Feature directory | DIAGRAM | Feature family atlas — eight groups mapped to chapters |
| **04** | Golden Path | SCREENSHOT | edgefit-ss-01-home-overview.png |

| | | | |
|---|---|---|---|
| **05** | Golden Path | SCREENSHOT | edgefit-ss-03-hardware-details.png |
| **06** | Golden Path | SCREENSHOT | edgefit-ss-07-benchmark-setup.png |
| **07** | Golden Path | SCREENSHOT | edgefit-ss-09-benchmark-running.png |
| **08** | Golden Path | SCREENSHOT | edgefit-ss-10-benchmark-results.png |
| **09** | Golden Path | SCREENSHOT | edgefit-ss-12-session-history.png |
| **10** | Golden Path | SCREENSHOT | edgefit-ss-22-agent-proposal-review.png |
| **11** | Golden Path | SCREENSHOT | edgefit-ss-34-global-evidence-consent.png |
| **12** | Golden Path | FLOWCHART | Fourteen-step first-launch-to-contribution journey |
| **13** | 03 Device Intelligence | SCREENSHOT | edgefit-ss-02-device-profile.png |
| **14** | 03 Device Intelligence | SCREENSHOT | edgefit-ss-03-hardware-details.png |
| **15** | 03 Device Intelligence | DIAGRAM | Device-telemetry / unavailable-field source map |
| **16** | 04 Models — Selection, Suitability and Identity | SCREENSHOT | edgefit-ss-04-model-recommendations.png |
| **17** | 04 Models — Selection, Suitability and Identity | SCREENSHOT | edgefit-ss-05-model-recommendation-detail.png |
| **18** | 04 Models — Selection, Suitability and Identity | SCREENSHOT | edgefit-ss-06-model-catalog.png |
| **19** | 04 Models — Selection, Suitability and Identity | SCREENSHOT | edgefit-ss-32-home-select-model.png |
| **20** | 04 Models — Selection, Suitability and Identity | FLOWCHART | Model recommendation / evidence gating decision tree |
| **21** | 05 Benchmark Laboratory — Real On-Device Inference | SCREENSHOT | edgefit-ss-07-benchmark-setup.png |
| **22** | 05 Benchmark Laboratory — Real On-Device Inference | SCREENSHOT | edgefit-ss-08-benchmark-setup-advanced.png |
| **23** | 05 Benchmark Laboratory — Real On-Device Inference | SCREENSHOT | edgefit-ss-08-02-benchmark-workload.png |
| **24** | 05 Benchmark Laboratory — Real On-Device Inference | SCREENSHOT | edgefit-ss-09-benchmark-running.png |
| **25** | 05 Benchmark Laboratory — Real On-Device Inference | SCREENSHOT | edgefit-ss-10-benchmark-results.png |
| **26** | 05 Benchmark Laboratory — Real On-Device Inference | FLOWCHART | Benchmark four-stage native execution and cancellation |
| **27** | 05 Benchmark Laboratory — Real On-Device Inference | SCREENSHOT | edgefit-ss-37-error-validation-state.png |
| **28** | 06 Sessions, Results and Evidence Ledger | SCREENSHOT | edgefit-ss-12-session-history.png |
| **29** | 06 Sessions, Results and Evidence Ledger | SCREENSHOT | edgefit-ss-25-sessions-repeat-series-compare.png |
| **30** | 06 Sessions, Results and Evidence Ledger | SCREENSHOT | edgefit-ss-11-predicted-vs-measured.png |
| **31** | 06 Sessions, Results and Evidence Ledger | SCREENSHOT | edgefit-ss-11-compare-measured.png |
| **32** | 06 Sessions, Results and Evidence Ledger | SCREENSHOT | edgefit-ss-36-export-json-evidence.png |
| **33** | 06 Sessions, Results and Evidence Ledger | DIAGRAM | Session state machine and evidence chain of custody |
| **34** | 07 Meta Muse Spark — Prediction, Evidence and Agentic Experiments | SCREENSHOT | edgefit-ss-13-muse-spark-prediction.png |
| **35** | 07 Meta Muse Spark — Prediction, Evidence and | SCREENSHOT | edgefit-ss-21-agent-live-prediction.png |

| | | | |
|---|---|---|---|
| | Agentic Experiments | | |
| **36** | 07 Meta Muse Spark — Prediction, Evidence and Agentic Experiments | SCREENSHOT | edgefit-ss-22-agent-proposal-review.png |
| **37** | 07 Meta Muse Spark — Prediction, Evidence and Agentic Experiments | SCREENSHOT | edgefit-ss-23-agent-proposal-review-02-rationale.png |
| **38** | 07 Meta Muse Spark — Prediction, Evidence and Agentic Experiments | SCREENSHOT | edgefit-ss-18-agent-optimization-approval.png |
| **39** | 07 Meta Muse Spark — Prediction, Evidence and Agentic Experiments | SCREENSHOT | edgefit-ss-15-agent-follow-up-detail.png |
| **40** | 07 Meta Muse Spark — Prediction, Evidence and Agentic Experiments | SCREENSHOT | edgefit-ss-16-agent-follow-up-request.png |
| **41** | 07 Meta Muse Spark — Prediction, Evidence and Agentic Experiments | FLOWCHART | Muse Spark bounded agent and tool-call lifecycle |
| **42** | 08 Security, Privacy and Validator Boundaries | SCREENSHOT | edgefit-ss-27-settings-meta-muse.png |
| **43** | 08 Security, Privacy and Validator Boundaries | SCREENSHOT | edgefit-ss-28-settings-privacy.png |
| **44** | 08 Security, Privacy and Validator Boundaries | DIAGRAM | Credential custody and deterministic validator trust boundary |
| **45** | 09 Global Evidence Network — From One Device to Qualified Public Statistics | SCREENSHOT | edgefit-ss-33-global-evidence-entry.png |
| **46** | 09 Global Evidence Network — From One Device to Qualified Public Statistics | SCREENSHOT | edgefit-ss-34-global-evidence-consent.png |
| **47** | 09 Global Evidence Network — From One Device to Qualified Public Statistics | SCREENSHOT | edgefit-ss-35-contribution-history.png |
| **48** | 09 Global Evidence Network — From One Device to Qualified Public Statistics | DIAGRAM | Consent, ingestion, quorum, aggregate publication and revocation |
| **49** | 10 Safe Installation, Data Retention and Recovery | FLOWCHART | FR-SAFE-01 ten-gate installation safety board |
| **50** | 10 Safe Installation, Data Retention and Recovery | SCREENSHOT | edgefit-ss-37-error-validation-state.png |
| **51** | Chapter 11 | CHART | Measured A/B generation throughput with individual runs, median and ranges |
| **52** | Chapter 11 | DIAGRAM | Measurement provenance stack and confidence/abstention taxonomy |
| **53** | 12 Distribution, Publication Gates and Honest Limitations | FLOWCHART | Public release seven-gate readiness board |
| **54** | 13 Settings, Onboarding and Help | SCREENSHOT | edgefit-ss-29-onboarding-01.png |
| **55** | 13 Settings, Onboarding and Help | SCREENSHOT | edgefit-ss-30-onboarding-02.png |
| **56** | 13 Settings, Onboarding and Help | SCREENSHOT | edgefit-ss-31-onboarding-03.png |
| **57** | 13 Settings, Onboarding and Help | SCREENSHOT | edgefit-ss-26-settings-benchmark.png |
| **58** | 13 Settings, Onboarding and Help | SCREENSHOT | edgefit-ss-27-settings-meta-muse.png |

| | | | |
|---|---|---|---|
| **59** | 13 Settings, Onboarding and Help | SCREENSHOT | edgefit-ss-28-settings-privacy.png |
| **60** | **13 Settings, Onboarding and Help** | **DIAGRAM** | **Five-tab atlas and six-settings-group map** |

## Editorial sign-off checklist

[ ] Every screenshot shows the correct application build and actual feature state.

[ ] All UI notifications, private keys, tokens and personal data are redacted.

[ ] Each concept illustration is clearly marked and not misrepresented as app UI.

[ ] All numbers trace to measured artifacts and dated records.

[ ] Cloud publication language matches independently verified current deployment.

[ ] Signed APK/demo/long-form-video links are real and live.

[ ] All image slots replaced or visibly marked intentionally unavailable.

[ ] GitHub Markdown and this Word edition tell the same factual story.

END OF EDITABLE MASTER / EDGEFIT AI