



FIRST
MEASURE

PROFESSIONAL PORTFOLIOS

FIRST MEASURE WHITE PAPER / ARCHITECTURE & GOVERNANCE

From Custom GPTs to a Governed Runtime

Rebuilding First Measure as a Shared AI Platform

OCTOBER 2026

INTERFACES INTERACT. THE RUNTIME DECIDES.



EXECUTIVE SUMMARY

A shift in authority.

First Measure began with three purpose-built Custom GPTs: Professional, Showcase, and Athlete.

Each product had its own instructions, standards, workflows, evidence rules, approval gates, and production requirements. The GPTs worked, but as the system matured, an architectural limitation became increasingly clear: too much business authority lived inside prompts.

The models could reason about the rules, but the system still depended heavily on a conversational model remembering, interpreting, and correctly applying increasingly complex instructions.

The next step was not simply to write better prompts. It was to move the rules themselves out of the GPTs.

First Measure was rebuilt around a shared plugin/runtime architecture in which ChatGPT remains the conversational and generative layer, while an executable runtime owns workflow state, validation, evidence integrity, approvals, production readiness, and downstream authorization.

The migration ultimately produced a 30-tool MCP surface, durable project persistence, a shared application-service boundary, an authenticated HTTP API, project ownership controls, and the foundation for a separate First Measure web portal - all using the same underlying runtime authority.

Interfaces interact. The runtime decides.

CONTENTS

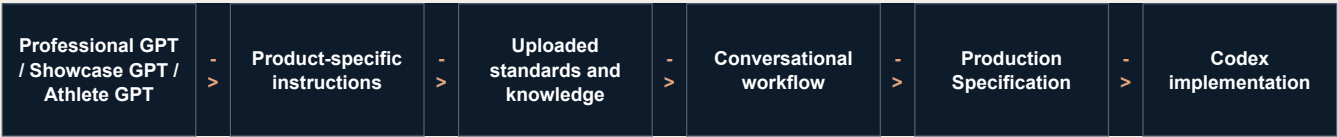
Inside the architecture

-
- | | |
|----|---------------------------|
| 01 | The Original Architecture |
|----|---------------------------|
-
- | | |
|----|----------------------------|
| 02 | The Architectural Decision |
|----|----------------------------|
-
- | | |
|----|-----------------------------|
| 03 | One Runtime, Three Products |
|----|-----------------------------|
-
- | | |
|----|--|
| 04 | Turning Prompt Rules into Executable Contracts |
|----|--|
-
- | | |
|----|------------------------------|
| 05 | Fail-Closed State Management |
|----|------------------------------|
-
- | | |
|----|-----------------|
| 06 | The Role of MCP |
|----|-----------------|
-
- | | |
|----|---------------------|
| 07 | Durable Persistence |
|----|---------------------|
-
- | | |
|----|--|
| 08 | Beyond the Plugin: A Shared Application Boundary |
|----|--|
-
- | | |
|----|---|
| 09 | HTTP, Authentication, and Project Ownership |
|----|---|
-
- | | |
|----|------------|
| 10 | The Portal |
|----|------------|
-
- | | |
|----|--------------------------|
| 11 | Testing the Architecture |
|----|--------------------------|
-
- | | |
|----|----------------------------|
| 12 | What Changed Fundamentally |
|----|----------------------------|
-
- | | |
|----|-----------------------------------|
| 13 | The Final Architectural Principle |
|----|-----------------------------------|
-

SECTION 01

The Original Architecture

The first version of First Measure used separate Custom GPTs for each portfolio product.



This approach was effective for proving the product.

It allowed each GPT to:

■ analyze intake information	■ evaluate evidence
■ develop strategy	■ propose portfolio direction
■ generate content	■ identify gaps
■ establish design direction	■ prepare implementation specifications

But the more capable the system became, the more responsibility accumulated inside the prompt layer.

Product routing, evidence integrity, approval requirements, package limits, publication decisions, design originality, access controls, production readiness, and handoff rules all depended on the model interpreting instructions correctly.

That created several risks.

SECTION CONTINUED

Five prompt-layer risks

Instruction drift

As instructions expanded, there was more opportunity for a model to interpret similar rules differently between products or sessions.

Duplicated authority

Professional, Showcase, and Athlete shared many system-wide rules but implemented them through separate GPT configurations.

Weak persistence

The conversation itself carried significant state. Restarting or changing interfaces could make continuity harder.

Difficult validation

Testing a prompt is fundamentally different from testing deterministic code. It is much harder to prove that a long instruction set will always reject stale approvals, preserve exact state, or prevent unsupported evidence from flowing downstream.

Limited portability

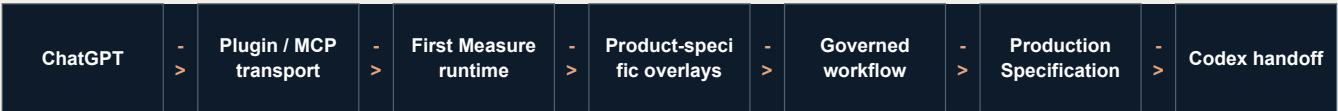
The GPT itself became both interface and workflow engine. That made it harder to build another interface - such as a client portal - without recreating the same rules elsewhere.

SECTION 02

The Architectural Decision

The core decision was to separate interaction from authority.

Instead of asking the language model to own the workflow, First Measure introduced a shared runtime.



The responsibility split became explicit.

ChatGPT

ChatGPT handles:

■ conversation	■ explanation
■ drafting	■ analysis
■ proposing Strategy	■ proposing Portfolio Direction
■ developing Content	■ helping the user understand decisions

SECTION CONTINUED

Authority moves to the runtime

First Measure Runtime

The runtime owns:

■ Product Route	■ evidence classification
■ public-use permission	■ readiness
■ Scope and package validation	■ Strategy state
■ Portfolio Direction state	■ approval state
■ Content state	■ Asset Manifest
■ Publication Approval	■ Access / Delivery
■ Production Readiness	■ Production Specification
■ Codex handoff	■ invalidation
■ stale-state protection	■ workflow progression

ChatGPT may propose. The runtime validates. ChatGPT may explain. The runtime decides whether the project can advance.

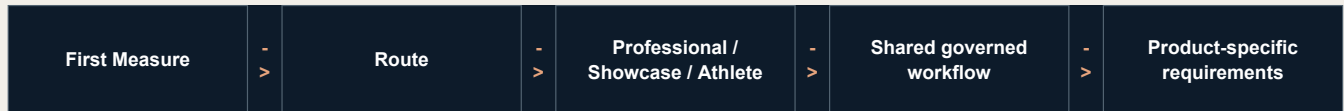
SECTION 03

One Runtime, Three Products

The migration did not collapse Professional, Showcase, and Athlete into one generic product.

Instead, it separated shared system authority from product-specific behavior.

The runtime determines the appropriate Product Route and applies the appropriate overlay.



This means a user no longer needs three separate rule engines.

The same runtime can enforce:

- Professional evidence and employment claims
- Showcase age-appropriate presentation and student safeguards
- Athlete recruiting and minor/guardian requirements

while preserving shared rules around:

- evidence integrity
- approvals
- originality
- access
- production readiness
- specifications
- implementation handoff

SECTION 04

Turning Prompt Rules into Executable Contracts

The most important part of the migration was not exposing tools. It was converting concepts that previously existed as instructions into explicit runtime contracts.

Evidence

The runtime distinguishes supported evidence from unsupported claims and separately tracks permission for public use. Evidence truth and publication permission are not treated as the same thing.

Scope

Package ceilings are enforced as ceilings rather than quotas. Recommendations do not become selections automatically. Selections require explicit confirmation.

Approvals

Direction, Content, Publication, and Access / Delivery approvals bind to exact authoritative fingerprints. If the underlying state changes, the old approval becomes stale or invalid.

Originality

Design Identity and Similarity Exclusions became structured authority rather than generic instructions such as 'make it unique.'

An implementation-ready portfolio cannot proceed without meaningful Similarity Exclusions.

SECTION CONTINUED

From publication to production

Access

The runtime distinguishes:

- | | |
|----------------------|-----------------------|
| ■ Public | ■ Low-Discovery |
| ■ Password | ■ Restricted Identity |
| ■ Review / Prelaunch | |

Low-Discovery is not security. An obscure URL or noindex setting cannot be represented as authentication.

Production

Production Readiness is determined by runtime state. A Production Specification cannot be generated simply because the model believes the project is ready. Codex handoff binds the exact current specification.

SECTION 05

Fail-Closed State Management

A major objective was to ensure that stale authority could never silently survive a material change.

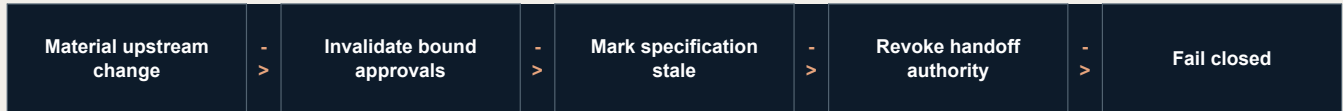
One of the final integration tests demonstrated this directly.

A project reached SPECIFICATION_COMPLETE with an authorized Production Specification and Codex handoff.

Then the approved hosting decision was materially changed.

The runtime immediately caused:

- Access / Delivery Approval to become invalid
- Production Readiness to become not current
- the Production Specification to become stale
- the Codex handoff to become stale
- implementation authorization to return to false



Attempting to reuse the old specification failed closed.

No old handoff resurrected. No replacement specification was silently generated.

That behavior is difficult to guarantee through prompt instructions alone. It is straightforward to test when authority is executable.

SECTION 06

The Role of MCP

The plugin surface eventually exposed 30 MCP tools.

But MCP was deliberately kept thin. MCP does not contain First Measure policy. Its job is transport.



This means the system is not dependent on MCP. The same runtime can now be used by other interfaces.

SECTION 07

Durable Persistence

One of the clearest limitations surfaced during real Desktop testing.

Projects initially lived in process-local memory. When the MCP process restarted, the project disappeared.

Rather than work around the problem in ChatGPT, First Measure introduced a repository abstraction with durable persistence.

A SQLite adapter was implemented for local validation behind the same repository contract used by the in-memory implementation.

The runtime can now survive complete process loss.

Real restart testing proved that the system preserved:

- | | |
|---------------------------|--------------------------------|
| ■ project versions | ■ approval histories |
| ■ invalidated authority | ■ evidence state |
| ■ Publication snapshots | ■ Production Specifications |
| ■ Codex handoffs | ■ implementation authorization |
| ■ terminal workflow state | |

A project at SPECIFICATION_COMPLETE survived a full Desktop restart with the same specification fingerprint, handoff identity, and authorization state.

Persistence remembers. It does not decide.

SECTION 08

Beyond the Plugin: A Shared Application Boundary

As the runtime matured, another architectural problem became apparent.

If MCP was the only orchestration layer, a future portal would either need to call MCP or reproduce its logic. Neither was acceptable.

First Measure therefore introduced a shared application boundary: `RuntimeApplicationService`.



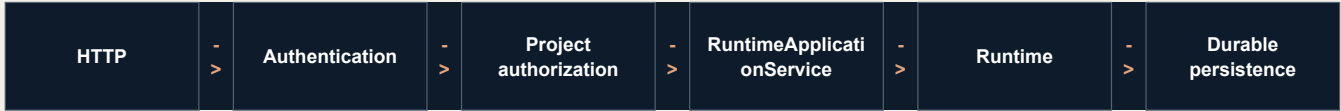
That shared boundary now provides typed operations, error contracts, expected-version protection, atomic operations, and repository injection.

MCP is simply one interface.

SECTION 09

HTTP, Authentication, and Project Ownership

A versioned HTTP API was then added as another thin adapter.



Authentication is provider-neutral and based on verified token identity.

Authorization is separate from First Measure workflow authority. This distinction matters.

An authenticated project owner does not automatically become:

- | | |
|-----------------------------|-------------------------|
| ■ a workflow approver | ■ a CLIENT participant |
| ■ implementation-authorized | ■ deployment-authorized |

Authentication answers

Who are you?

Authorization answers

Which project may you access?

The runtime still answers

What is allowed to happen to this project?

SECTION 10

The Portal

Once the runtime could be safely accessed outside ChatGPT, First Measure began building a separate portal.

The portal uses the same runtime indirectly through the authenticated API.

It does not read the database. It does not depend on MCP. It does not recreate workflow logic.



Two interfaces. One authority.

SECTION 11

Testing the Architecture

The migration was repeatedly tested at multiple layers:

- | | |
|-------------------------------------|--------------------------------|
| ■ deterministic unit tests | ■ integration tests |
| ■ independent adversarial reviews | ■ real ChatGPT Desktop testing |
| ■ real process termination/restart | ■ real SQLite persistence |
| ■ real HTTP/Uvicorn processes | ■ real signed test tokens |
| ■ multiple authenticated principals | ■ cross-project isolation |
| ■ stale-version conflicts | ■ browser-level portal tests |
| ■ accessibility testing | |

30
MCP TOOLS

1,800+
PASSING TESTS

3
PRODUCT ROUTES

The runtime test suite eventually exceeded 1,800 passing tests.

But some of the most important defects were found outside the automated suite.

Real Desktop beta testing uncovered:

- | | |
|---|--|
| ■ a missing Similarity Exclusions readiness guard | ■ stale Desktop plugin-cache behavior |
| ■ process-local project loss | ■ unnecessary MCP initialization of authenticated-access persistence |

Each defect was preserved as evidence, corrected, independently reviewed, and retested.

Automated testing and real integration testing catch different classes of failure. Both matter.

SECTION 12

What Changed Fundamentally

The migration began as an effort to move First Measure beyond three Custom GPTs. It ultimately changed the nature of the product.

ORIGINAL ARCHITECTURE

AI model with instructions

CURRENT ARCHITECTURE

AI interface + governed runtime +
durable state + shared service
boundary + multiple clients

That creates several advantages.

Consistency

Professional, Showcase, and Athlete share one executable system.

Testability

Workflow decisions can be validated deterministically.

Recoverability

Projects survive process and interface restarts.

Portability

ChatGPT is no longer the only possible interface.

Extensibility

The portal can evolve independently without recreating business rules.

Governance

Approvals, evidence, state transitions, and downstream authority are explicit.

AI flexibility

The language model can continue doing what it does best - reasoning, generating, explaining, and collaborating - without being responsible for being the database, state machine, and policy engine at the same time.

SECTION 13

The Final Architectural Principle

The most important design decision can be summarized in one sentence:

Interfaces interact. The runtime decides.

ChatGPT can propose a Strategy. The portal can display it. A client can approve it. Codex can implement it. A database can persist it.

But none of those components independently determine whether that Strategy is authoritative.

That responsibility belongs to the First Measure runtime.

The migration from Custom GPTs to a plugin/runtime architecture therefore was not simply a tooling change.

It was the transition from an AI prototype to a governed AI application platform.