

AI/ML — progress & three-sprint next steps

Audience: AI / ML engineering

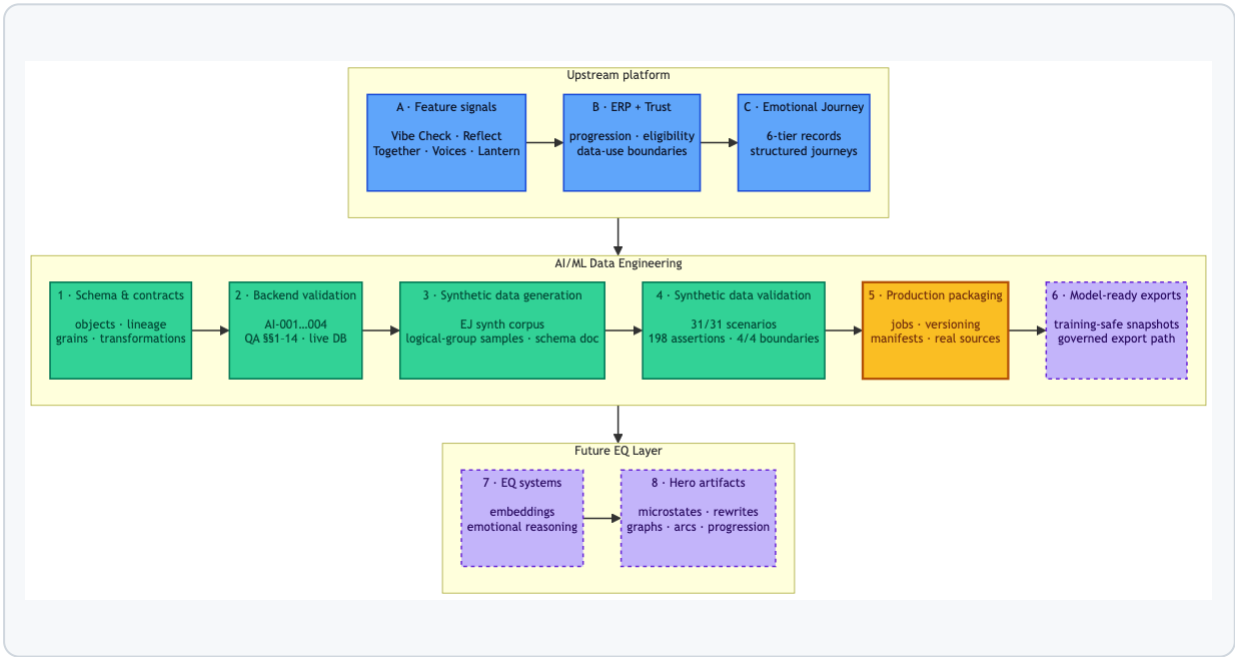
Date: 2026-08-07

Status: Schema + **synthetic generation** + **synthetic validation** complete (Aditya) · packaging still stub-only · ready for production packaging work

Source of truth for product intent: [AI-ML project overview – v1.2.pdf](#)

Epic map — overall AI/ML progress

Bright green = complete · bright amber = **current stage** · violet dashed = ahead · bright blue = upstream



Color	Meaning	Epics
Bright green	Complete	Schema & contracts · Backend validation · Synthetic generation (Aditya) · Synthetic validation (Aditya)
Bright amber	Current stage	Production packaging (Sprint 2)
Violet dashed	Ahead / deferred	Model-ready exports · EQ systems · Hero artifacts
Bright blue	Upstream context	Features · ERP/Trust · Emotional Journey

Bigger picture — where the AI team is right now

SAY!T’s ultimate objective is not “another mental health app.” It is to build an **EQ layer for AI**: structured emotional-intelligence data that helps future systems reason about emotion, context, triggers, change over time, and social-emotional dynamics — journeys model makers cannot get from generic text alone.

That EQ layer will eventually produce five **hero artifacts**:

- 1. Microstates + triggers + intensity + identity context
- 2. Peer-to-peer emotional rewrites
- 3. Emotional reasoning graphs
- 4. Longitudinal emotional arcs
- 5. Progression modeling (spark → glow → lift → let go)

Those artifacts are **downstream**. Before they can exist, the platform must run this pipeline:

Features (Vibe Check, Reflect*, Together, Voices, Lantern, Your Layers, Help Me, ...)

- ERP (progression) + Trust Layer (eligibility / data-use boundaries)
- Emotional Journey (structured emotional records / 6-tier taxonomy)
- AI/ML Data Engineering (governed transform · lineage · package · version · export)
- future EQ systems (embeddings, reasoning, hero artifacts)

*Product rename: overview PDF still says “Moments”; current platform name is **Reflect**.

What the purple (ahead) steps mean

These are the deferred stages after the current packaging work. They are intentionally not started until governed, training-safe datasets exist.

Ahead step	What it is	Why it matters
Model-ready exports	The governed handoff path that turns packaged snapshots into training-safe export batches — reproducible manifests, eligibility/sensitivity re-checked at export time, raw/private/crisis content kept out of model_ready_payload	Without this, EQ work would train on unversioned or unsafe data; packaging alone is not yet a consumable model feed
EQ systems	Downstream models and services that learn from those exports — embeddings, emotional-reasoning	This is where “EQ layer for AI” becomes runnable

Ahead step	What it is	Why it matters
	machinery, and related inference over structured journeys	capability, not just governed tables
Hero artifacts	The productized outputs of that EQ layer — the five deliverables listed above (microstates, peer rewrites, reasoning graphs, longitudinal arcs, progression modeling)	These are the visible outcomes for model makers and product; they depend on both model-ready exports and working EQ systems

In short: **model-ready exports** feed **EQ systems**, which produce **hero artifacts**. Everything green in the epic map exists to make that chain trustworthy.

What the AI/ML team owns (and does not)

Per the project overview, the AI/ML team builds the **governed Data Engineering layer** that turns structured platform records into trustworthy, reproducible datasets for future EQ systems. The team does **not** invent product behavior, own ERP progression logic, or run Trust Layer decisions.

Owner focus	Owns	Current state
Schema (Aditya)	Objects/events, direct vs derived separation, lineage, dataset contracts, transformation structure	Complete — live on Supabase; Sprint 1 QA 88/88
Synthetic data generation (Aditya — sole)	EJ synth corpus, representation guide, generators, logical-group sample exports, synthetic schema doc	Complete — <code>synth-emotional-journeys.csv</code> · 8 logical-group CSVs · synthetic-schema.md
Synthetic data validation (Aditya — sole)	31-scenario suite, validation runner, lineage/grain/eligibility/sensitivity proofs, platform-boundary checks, results docs	Complete — 31/31 scenarios · 198/198 assertions · 4/4 boundaries
Datasets / packaging (Clare + Aditya)	Dataset types, grain, packaging jobs, versioning, quality, export governance	Contracts complete · production packaging still open
Shared	Governed transforms, reproducibility, exports, model-ready datasets	Ready to implement packaging jobs on validated contracts + synthetic gates

Where we sit on the pipeline today

Pipeline step	Status for AI/ML
Features → ERP → Trust → Emotional Journey	Upstream / out of AI/ML ownership — consume only approved structured outputs
Schema + dataset contracts	Done (Aditya) — <code>AI-001</code> ... <code>AI-004</code> applied; Sprint 1 QA 88/88 on project DB
Synthetic data generation	Done (Aditya sole) — EJ corpus + generators + logical-group samples + synthetic schema doc
Synthetic data validation	Done (Aditya sole) — Sprint 3 31/31 · 198/198 · 4/4 boundaries; runner + results live
Governed packaging / versioning / lineage / reproducible export	In progress — proven by Aditya's synthetic stub + suite; production jobs not shipped
Model-ready datasets for EQ / hero artifacts	Blocked on packaging — contracts and training-safety gates exist; no production export path yet
Future EQ layer (embeddings, reasoning, hero artifacts)	Not started — correctly deferred until governed datasets exist

One-line position: Aditya has closed **schema**, **synthetic generation**, and **synthetic validation**. The team is entering the **packaging** phase — the step that turns validated contracts into reusable datasets the future EQ layer can actually learn from.

Current progress (what is already done)

The AI/ML Data Engineering layer has completed foundation and acceptance validation. Live project checks against Supabase passed after the Moments → **Reflect** rename.

Track	Owner	What it covers	Status
Sprint 1 — Backend validation	Aditya (schema)	SQL package + QA checklist §§1–14	Complete — 88/88 checks on project Supabase
Synthetic data generation	Aditya (sole)	EJ synth corpus, generators, logical-group samples, synthetic schema doc	Complete — corpus + 8 group CSVs + synthetic-schema.md

Track	Owner	What it covers	Status
Sprint 2 — Packaging implementation	Clare (+ Aditya contracts)	Production packaging jobs, versioning, lineage, manifests	Open — schema ready; only a validation stub exists
Sprint 3 — Synthetic data validation	Aditya (sole)	31-scenario suite, assertion harness, boundary proofs	Complete — 31/31 · 198/198 · 4/4 boundaries

Evidence

Artifact	Role
docs/ai-ml-implementation-progress-report.md	Stage/sprint progress detail
docs/emotional-journey-data.md	Synthetic EJ representation guide
scripts/generate_synth_emotional_journeys.py · scripts/build_synth_from_jsonl.py	Synthetic corpus generators
data/synth-emotional-journeys.csv · exports/synthetic_logical_groups/	Generated synthetic fixtures
docs/synthetic-schema.md	Synthetic logical-group schema doc
docs/ai-ml-synthetic-validation-results.md	Latest Stage 3 validation results
docs/ai-ml-synthetic-validation-plan.md	Scenario plan
sql/ai_ml/AI-001 ... AI-004	Canonical backend package
scripts/validate_ai_ml_schema.py	Sprint 1 QA harness
scripts/run_synthetic_validation.py	Sprint 3 runner + packaging stub

Product naming note: the feature formerly called **Moments** is now **Reflect**

(`source_system=reflect`, `reflect_saved`, `reflect_prompt_exposure`). Use Reflect in all new packaging and model contracts.

What this means for the AI team

1. **You can trust the governed schema.** Dataset grains, eligibility/sensitivity registries, lineage columns, export batch records, and governed views are live and constraint-tested.
2. **You cannot yet treat packaging as production.** Stage 3 proves invariants with a minimal `Packager` stub inside the synthetic runner. Real jobs that read Emotional Journey / feature sources and write model-ready snapshots are still the gap.
3. **Regression gate is ready.** Re-run Sprint 1 + Sprint 3 after every packaging change:

```
python3 scripts/validate_ai_ml_schema.py --psql-url "$DATABASE_URL"
python3 scripts/run_synthetic_validation.py --psql-url "$DATABASE_URL"
```

Three sprints as next steps

These are the same three engineering tracks, sequenced as the AI team’s forward plan. Sprint 1 and Sprint 3 are **closed as validation milestones** but remain **required regression gates**. Sprint 2 is the primary delivery sprint.

Sprint 1 — Backend validation (gate, not greenfield)

Goal: Keep the project database aligned with the AI/ML SQL package and QA checklist.

Already done

- `AI-001` ... `AI-004` applied and re-runnable
- Live Supabase QA: **88/88 pass**
- Seeds include Reflect, 31 scenarios, grain/source rules, transformation versions

Next steps for AI team

#	Action	Done when
1.1	Treat <code>sql/ai_ml/</code> as source of truth for any schema change	PR includes SQL + harness update
1.2	Run <code>validate_ai_ml_schema.py</code> against project DB before packaging PRs merge	CI or checklist shows §§1–14 green
1.3	Record pass/fail on each environment cut (staging vs prod)	Short note in PR or results doc

Exit criteria: No packaging change ships without a current Sprint 1 green run.

Sprint 2 — Packaging implementation (primary delivery)

Goal: Replace the Stage 3 stub with production packaging jobs that write governed, model-ready datasets.

Already done (contracts only)

- Tables for snapshots, dataset records, lineage, exclusions, export batches
- Transformation version registry
- Export-time re-evaluation model (`ai_ml_export_batch_records`)
- Stub `Packager` used only for synthetic acceptance

Next steps for AI team

#	Action	Done when
2.1	Promote stub into a real packaging job/service	Job reads approved sources and writes snapshots + records
2.2	Implement transformation version selection + approval checks	Only <code>approved_for_use=true</code> versions package for training
2.3	Generate full source lineage (separate EJ / Reflect / Trust / ERP ids)	Lineage rows present; no coalesced ids
2.4	Enforce grain/source rules at package time	Invalid Reflect/EJ grain pairs rejected
2.5	Emit export manifests + reproducible export batches	Matching hashes across re-exports of same snapshot
2.6	Keep raw / Help Me / crisis content out of <code>model_ready_payload</code>	Model-ready view stays training-safe only

Non-negotiable packaging rules (from validation refinements)

1. Export statuses: only `planned` / `building` / `ready` / `failed` / `archived`
2. Do **not** mutate packaging-time eligibility for later governance changes — write export-time decisions
3. Consult `ai_ml_dataset_grain_source_rules` in application code (no DB trigger today)
4. Force `is_training_safe=false` for non-training sensitivity classes

Exit criteria: Production packager can create a snapshot from real (or staging) sources, pass Sprint 3 regression, and produce a reproducible export manifest.

Sprint 3 — Synthetic data validation (Aditya sole · acceptance + regression)

Goal: Prove lineage, eligibility, sensitivity, immutability, export reproducibility, and platform boundaries against the generated synthetic corpus and seeded scenario suite.

Already done (Aditya)

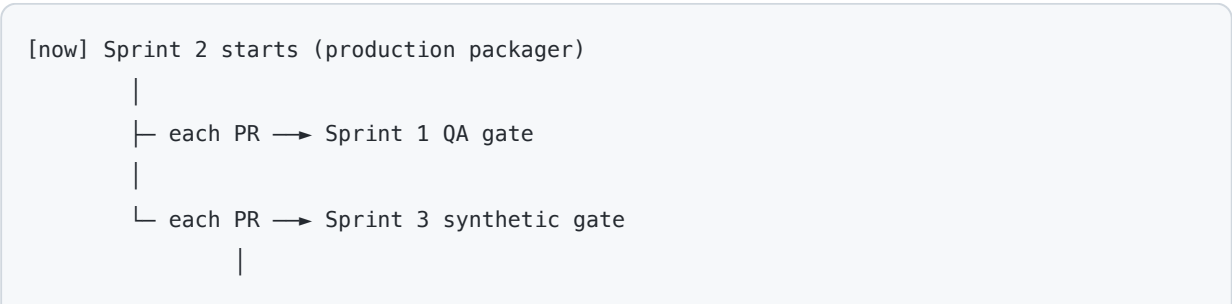
- Full plan executed: **31/31 scenarios, 198/198 assertions, 4/4** platform boundaries
- Scenario seeds in `AI-002`; runner `scripts/run_synthetic_validation.py`; results in [ai-ml-synthetic-validation-results.md](#)
- Runner persists `ai_ml_synthetic_runs` / `ai_ml_synthetic_run_records`
- Upstream generation assets already shipped: EJ generators, `data/synth-emotional-journeys.csv`, logical-group sample exports, [synthetic-schema.md](#)
- Harness hardened for PostgREST missing-column detection and secret redaction in failure notes

Next steps for AI team

#	Action	Done when
3.1	Re-run full synthetic suite after every Sprint 2 packaging change	<code>run_synthetic_validation.py</code> exits 0
3.2	Add scenarios only when a new packaging risk appears	New key seeded in <code>AI-002</code> + runner coverage
3.3	Keep platform boundaries intact (no Trust/ERP/EJ/Home-MDG logic under <code>ai_ml_*</code>)	Boundary checks stay 4/4
3.4	Update ai-ml-synthetic-validation-results.md on material runs	Dated evidence for handoff

Exit criteria for a packaging release: Sprint 3 green on the same commit as the packaging job.

Suggested sequence



▼
first training-safe export from governed packager

Priority	Sprint	Why now
P0	Sprint 2	Only incomplete delivery track
P0	Sprint 1 + 3 as gates	Prevent contract drift while packaging lands
P1	Reflect-aware packaging fixtures	Product rename must appear in real source adapters

Out of scope for these three sprints

- Rebuilding Trust Layer, ERP, Emotional Journey, or Home/MDG logic inside AI/ML
- Copying raw private Reflect / Help Me / crisis payloads into training tables
- Expanding beyond the approved AI-004 governed view set without an explicit compatibility review

Quick commands

```
# Sprint 1 – backend QA
python3 scripts/validate_ai_ml_schema.py --psql-url "$DATABASE_URL"

# Sprint 3 – synthetic acceptance / regression
python3 scripts/run_synthetic_validation.py --psql-url "$DATABASE_URL"
```

Requires `.env` with `NEXT_PUBLIC_SUPABASE_URL`, `NEXT_PUBLIC_SUPABASE_PUBLISHABLE_KEY`, and `DATABASE_URL` (prefer Session pooler on IPv4-only networks).