ClientSynth Roadmap

> **Purpose:** A platform for generating realistic synthetic participant populations for thin market prototyping, demonstration, and simulation.
>
> **Date:** March 6, 2026 **Author:** Mustafa Uzumeri

---

1. What is ClientSynth?

ClientSynth is a **multi-tenant platform for generating realistic synthetic data** using AI. Users design participant schemas, run generation jobs, and export the results in multiple formats. The platform is designed to populate prototype marketplace instances with demographically plausible, economically coherent, culturally appropriate synthetic participants — without using real user data.

Synthetic participants are used **exclusively** for testing and demonstration. A real-world marketplace must be built with real participants recruited by the sponsor. Mixed use of synthetic and real users in the same marketplace instance is not permitted.

---

2. What's Built and Working

Core Data Generation Pipeline ✅

- **Visual Schema Designer** — Drag-and-drop schema creation with 17+ field types (name, email, phone, company, address, date, number, text, image, PDF, enum, boolean, URL, JSON, custom AI)
- **AI-Powered Generation** — Context-aware field value generation. Per-field prompts, system prompts, example data awareness, batch generation, and retry logic
- **Job Processor** — Batch processing with configurable batch sizes, exponential backoff retry, recovery states, real-time pause/resume/cancel, progress tracking, and de-duplication
- **Deterministic Fallbacks** — When AI generation fails, the system falls back to deterministic generators for common field types

Schema Intelligence ✅

- **Schema Discovery** — Automatic schema inference from uploaded data files
- **Schema Induction** — Field clustering, type detection, constraint detection, and LLM-assisted field description generation
- **Universal File Parser** — Parses CSV, JSON, XLSX, and other formats for schema discovery

Image Generation ✅

- **Multi-Provider Architecture** — Provider abstraction layer with fallbacks
- **Prompt Enhancement** — Context-aware image prompt building from record data
- **Batch Image Generation** — Parallel image generation for performance
- **Cloud Storage Integration** — Automatic upload, URL generation, and metadata tracking

PDF Generation ✅

- **Template-Based PDF Creation** — Full template management with versioning
- **AI-Powered Content Generation** — Generates entire documents from data context
- **PDF Parsing** — Extracts schema information from existing PDF documents for schema discovery

Data Quality & Variation ✅

- **Distribution Manager** — Target distributions for categorical fields with deviation analysis and rebalancing recommendations
- **Pattern Detector** — Detects repetitive patterns in generated data
- **Similarity Scorer** — Cross-record uniqueness scoring
- **Cooldown Tracker** — Prevents recent value re-use

Intelligence Layer (Partially Complete) 🟡

- Seed quality predictor, AI labeling engine, and generation metrics infrastructure is built
- Database tables and service code exist but are not fully connected to the user interface

Multi-Tenant Architecture ✅

- Full tenant isolation with role-based access (owner / admin / member)
- Organization selection with role display
- Shared tenant context management

Export System ✅

- Multi-format export: CSV, JSON, XLSX, SQL
- Export management dashboard

Additional Capabilities ✅

- Seeding infrastructure — pre-populate schemas with example data
- MCP server — external tool connectivity
- Job console — real-time job monitoring with bulk operations and advanced filtering
- Image gallery — AI-generated image management
- PDF template management

---

3. What's Not Yet Built

Capability	Notes
Cosolvent API integration	No shared format or import/export contract defined yet
Scenario-based generation	Currently generates individual records, not coherent populations
Inter-record relationships	Records are generated independently; no referential consistency
Behavioural scripting	Static data only; no time-series or action sequences
Population-level quality scoring	Quality scoring is per-record, not per-population
Headless API mode	UI-only; no programmatic generation without the interface

Capability	Notes
Real-time collaboration	Single-user schema editing
Schema version control	No schema history or diff
Domain-specific templates	No pre-built schema templates for common use cases

4. Roadmap

The roadmap is organized into three tracks:

- **Track S — Standalone Product:** Features that make ClientSynth more useful as an independent platform
- **Track C — Cosolvent Integration:** Integration points with the Cosolvent marketplace platform
- **Track D — Digital Twin:** Simulation and population generation for market modelling

Foundation Phase (Weeks 1–4)

Quick wins that improve the product regardless of integration:

Item	Description	Effort
F.1	Fix migration numbering conflict	< 1 day
F.2	Rename project in package configuration	< 1 hour
F.3	Add structured logging with request correlation	2–3 days
F.4	Complete Intelligence Layer UI — connect quality predictor and generation metrics to the dashboard	3–5 days
F.5	Add schema versioning — track changes over time, allow rollback	3–4 days
F.6	Build headless API mode — programmatic job creation and result retrieval without the UI	5–7 days
F.7	Add pre-built schema templates — common data shapes users can start from	3–5 days

Track S — Standalone Product Enhancements

S1: Generation Quality (Weeks 3–8)

Item	Description	Effort
S1.1	Inter-record relationships — support relationships between schemas (e.g., an "orders" schema references generated "customers")	5–7 days

Item	Description	Effort
S1.2	Conditional field generation — field values depend on other field values in the same record	3–5 days
S1.3	Computed/derived fields — fields calculated from other generated fields	2–3 days
S1.4	Multi-model generation strategy — different AI models for different field types within the same job	3–4 days
S1.5	Locale-aware generation — culturally appropriate names, addresses, and phone formats for specific countries and regions	3–5 days

S2: User Experience (Weeks 4–10)

Item	Description	Effort
S2.1	Schema import from database — connect to an existing database and infer schemas from table structures	5–7 days
S2.2	Real-time generation preview — show sample records as the user builds the schema, before running a full job	3–5 days
S2.3	Generation profiles — save and reuse generation configurations	2–3 days
S2.4	Improved export formats — add Parquet, Avro, and direct database insert	3–5 days
S2.5	Job scheduling — schedule recurring generation jobs	3–5 days

S3: Monetization & Platform (Weeks 6–12)

Item	Description	Effort
S3.1	Usage metering — track tokens, images, and records per tenant for billing	3–5 days
S3.2	Subscription billing with usage-based tiers	5–7 days
S3.3	Public schema marketplace — users can publish and share schema templates	5–7 days
S3.4	Team collaboration — real-time schema editing, shared job history, role-based permissions	7–10 days
S3.5	Webhook notifications — notify external systems when jobs complete	2–3 days
S3.6	Audit logging — track all user actions for compliance	3–5 days

Track C — Cosolvent Integration

Integration Strategy: Files First, API Later

Cosolvent and ClientSynth currently have no integration. The pragmatic approach is to not wait for a formal API contract. Cosolvent already uses flexible JSON storage for participant data, which means ClientSynth can produce Cosolvent-compatible export files immediately — without API changes needed on either side.

Integration maturity ladder:

Stage	Approach	When
C0 — File-based	ClientSynth exports JSON files in Cosolvent's participant format. A script loads them.	Now — no dependency on Cosolvent's timeline
C1 — Configuration awareness	ClientSynth imports a Cosolvent marketplace configuration and auto-generates conformant schemas	After Cosolvent Phase 1 stabilizes
C2 — API integration	Direct API calls between the two platforms	Future

C0: File-Based Integration (Weeks 2–4) — Do First

No changes to Cosolvent required. Results in: every Cosolvent feature becomes testable with realistic synthetic data immediately.

Item	Description	Effort
C0.1	Document Cosolvent's participant data format	< 1 day
C0.2	Add "Cosolvent Participant" as an export format	2–3 days
C0.3	Create a Cosolvent participant schema template in ClientSynth	1–2 days
C0.4	Write an import script for Cosolvent — reads a ClientSynth export and loads it into a Cosolvent instance	1–2 days
C0.5	End-to-end validation — generate 50 synthetic participants, export, load into Cosolvent, verify they appear in the gallery and matching pipeline	1–2 days

Total C0 effort: 5–9 days. Highest-leverage integration work — makes every other Cosolvent feature testable immediately.

C1: Configuration Awareness (Weeks 6–10)

Starts when Cosolvent's dynamic profile schema model stabilizes:

Item	Description	Effort
C1.1	Configuration import — accept a Cosolvent marketplace configuration and auto-generate corresponding ClientSynth schemas for each participant type	5–7 days

Item	Description	Effort
C1.2	Participant type awareness — understand the relationships between participant types and generate balanced ratios	3–5 days
C1.3	Field semantic awareness — ensure that matching fields across participant types (e.g., "certification" on a producer vs. "certification_required" on a buyer) are generated with compatible values	5–7 days

C2: Synthetic Population Engine (Weeks 8–14)

Item	Description	Effort
C2.1	Scenario definition — define population scenarios in structured format: participant types, counts, regional distributions, and behavioral parameters	5–7 days
C2.2	Geographic distribution controls — generate geographically coherent populations with culturally appropriate attributes per region	5–7 days
C2.3	Inter-participant consistency — ensure the synthetic population is internally coherent (regional producer counts match appropriate logistical capacity; buyer demand matches available supply)	7–10 days
C2.4	Document generation for participants — generate realistic supporting documents (certificates, invoices, compliance docs) attached to synthetic participants	5–7 days
C2.5	Population-level quality scoring — evaluate the population as a whole for market realism: buyer/seller ratios, facilitator coverage, geographic distribution, capacity matching	5–7 days

---

Track D — Digital Twin & Simulation

These capabilities enable simulation scenarios for market modelling and research:

D1: Behavioural Scripting (Weeks 12–18)

Item	Description	Effort
D1.1	Time-series data generation — generate data that changes over time (seasonal production volumes, price fluctuations, availability windows)	5–7 days
D1.2	Action/event generation — generate behavioural sequences for participants over time	7–10 days
D1.3	Market event simulation — generate exogenous events (supply shocks, new trade agreements, certification audits) that affect participant behaviour	5–7 days
D1.4	Cosolvent simulation API — feed generated behaviours into a running Cosolvent instance to simulate marketplace dynamics	7–10 days
D1.5	Simulation analytics — capture and visualize simulation results: matches formed, deals assembled, market clearing rates	5–7 days

5. Priority Sequence

For standalone product value — do first

1. **Foundation housekeeping** (F.1–F.3) — migration fix, project rename, structured logging. Less than 1 week.
2. **Headless API mode** (F.6) — unlocks programmatic access and is a prerequisite for integration with Cosolvent and for developer adoption.
3. **Schema templates** (F.7) — immediate usability improvement; new users can start generating data in minutes.
4. **Inter-record relationships** (S1.1) — the most significant feature gap for serious synthetic data users. Without it, only flat, unrelated records can be generated.

For Cosolvent integration — start immediately with C0

1. **File-based integration** (C0.1–C0.5) — do this in weeks 2–4, in parallel with Foundation work. No dependency on Cosolvent's development timeline. Every Cosolvent feature becomes testable from day one.
2. **Configuration import** (C1.1) — start when Cosolvent's Phase 1 profile schema work stabilizes.
3. **Scenario definitions** (C2.1) — makes ClientSynth qualitatively more powerful than a simple data generator.

For digital twin simulation — after Cosolvent matching engine is functional

1. **Time-series and behavioural scripting** (D1.1–D1.2) — can begin independently but only become useful when Cosolvent's deal entity and matching engine are functional.

---

6. Effort and Timeline Summary

Track	Items	Developer effort	With AI assistance
Foundation (F.1–F.7)	7	4–6 weeks	1–2 weeks
S: Standalone (S1–S3)	16	12–18 weeks	5–8 weeks
C0: File-based (C0.1–C0.5)	5	1–2 weeks	< 1 week
C1–C2: Cosolvent	8	9–13 weeks	4–6 weeks
D: Digital Twin (D1)	5	6–9 weeks	3–5 weeks
Total	41	32–48 weeks	**13–22 weeks**

Development Timeline

Phase	Calendar weeks	What becomes possible
Foundation	1–4	API mode, templates, quality metrics dashboard

Phase	Calendar weeks	What becomes possible
C0: File-based integration	2–4	Synthetic participants loadable into Cosolvent — all Cosolvent features become testable
Standalone MVP	3–10	Related records, conditional fields, database import, real-time preview
C1–C2: Cosolvent integration	6–14	Configuration import, population generation, semantic field matching
Digital Twin	12–18	Behavioural scripting, simulation, analytics

With one developer and AI assistance: approximately 4 months to complete Foundation + C0 + Standalone + Cosolvent integration. Digital Twin adds 4–6 weeks on top. C0 alone (file-based integration) can be done in under 2 weeks and delivers immediate value.

---

7. Architectural Strengths

1. **Multi-tenant with proper isolation.** Row-level security policies are production-grade.
2. **Schema-driven, not hardcoded.** The schema builder and AI generation pipeline are genuinely flexible — adding new field types is straightforward.
3. **Robust job processor.** Pause/resume/cancel, retry with exponential backoff, recovery states, and progress tracking are well-engineered for production.
4. **Provider abstraction.** Both text and image AI generation use provider interfaces, making new providers pluggable.
5. **Data quality infrastructure.** Distribution management, pattern detection, uniqueness scoring, and cooldown tracking provide a thoughtful variation layer.

This roadmap will be updated as implementation progresses and as integration with Cosolvent and the MarketForge workflow evolves.