The Layman's Guide to Marketplace Creation

Technical Briefing | DeeperPoint Outreach & Strategic Materials * Author & Copyright: *Mustafa Uzumeri | © 2026 Mustafa Uzumeri. All rights reserved. * **Publication Date:** May 2026 * **Document Status:** Preliminary & Provisional * **Disclaimer:** Designs may evolve differently than currently anticipated as we gain experience.*

This guide walks you through the step-by-step process of taking a target market vertical, setting up the infrastructure, curating the knowledge, stress-testing it with simulated users, customizing the interface, and preparing for a successful live launch.

1. The DeeperPoint Blueprint at a Glance

Before opening a terminal, it helps to understand how the five core components of the toolkit fit together to build a marketplace:

- **Cosolvent:** The core bilateral matching engine. It securely stores participant profiles, runs vector similarity matches, and hosts the private participant-to-participant negotiation workspace.
- **CommonContext:** The sponsor-curated industry reference library. It encodes the sector's regulatory rules, standard contract terms, and—crucially—alternative compliance "escape hatches" that bypass transaction-blocking standards.
- **ClientSynth:** The specialized synthetic cohort generation pipeline. Rather than a generic open-source utility, ClientSynth is a methodology and custom tooling suite that requires careful design and domain-specific interpretation to synthesize a highly diverse, realistic population of mock buyers, sellers, and facilitators for validation.
- **Digital Twin:** The flexible, self-contained testing sandbox. It integrates the core matching engines, standard contracts, and custom-designed synthetic cohorts within a local developer

environment, allowing sponsors to stress-test their specific matching physics and clear cold-start barriers before launching to real users.

- **Generative Match Story (GMS) Engine:** The translator that takes a high-scoring match and crafts a concrete, plain-English narrative of how a deal would actually unfold, showing both parties a step-by-step path to trust.

---

2. Core Platform vs. Vertical Customization: The Line of Responsibility

Building a successful vertical marketplace relies on a clean architectural split between the foundational engines and the regional business operations:

1. **The Core Platform Layer (DeeperPoint Open Source Project):** Owns and maintains the generic underlying engines (Cosolvent core matching engine, CommonContext reference framework, and the GMS narrative builder), base database models (PostgreSQL with `pgvector`), schema compilation pipelines, and feedback loops. These components are generic, open-source, and reusable across any industrial sector. Note that ClientSynth, because it requires highly specialized design to generate realistic cohorts, is currently maintained as a proprietary or high-touch methodology and custom-configured service rather than a generic open-source release.
2. **The Domain Customization Layer (Vertical Market Founder):** Owns the industry-specific configurations (`marketplace.yaml`), regional/cloud deployment, local knowledge curation (standards, bulletins, and legal contracts), frontend visual layout, GMS prompt styling, and active curator loops.

---

3. Marketplace Customization Responsibility Matrix

This matrix details the exact division of labor across the 12 steps of the creation guide:

Phase & Step	Task Summary	Primary Owner	Delivered by Core Project (Reusable Engine)	Delivered by Founder (Domain Configuration)
Phase 1: Infrastructure				
Step 1: Deployment	Clone repos & launch Docker	**Vertical Founder**	Container images & compose configurations	Regional cloud hosting (e.g. AWS) & local terminal execution
Step 2: Scoping	Custom roles, schemas & keys	**Vertical Founder**	Flexible `marketplace.yaml` schema compiler	Custom vertical YAML, participant roles, & field schemas
Phase 2: CommonContext				
Step 3: Document Ingest	Gather & convert reference papers	**Vertical Founder**	Text conversion services (`convert_pdf`, `convert_url`)	Sourcing local contracts, technical codes, & standards
Step 4: Meta-Schema	Extract & calibrate matching tags	**Joint Effort**	AI Schema Analyzer extraction pipeline	Reviewing, correcting, & calibrating industry vocabulary
Step 5: Escape Hatches	Ingest regulatory exemptions	**Vertical Founder**	ACR Database Schemas & Extraction Prompts	Crawling auxiliary bulletins, saving ACR workarounds
Step 6: Schema Lock	Update schemas with ACR triggers	**Vertical Founder**	Metadata-filtered vector search engines	Injecting physical & operational triggers in YAML schema
Phase 3: Stress-Test				
Step 7: ClientSynth	Define triggers & design cohorts	**Vertical Founder**	Cohort design framework & simulation specs	Custom persona synthesis logic & generation config

Preliminary & Provisional: Designs may evolve differently than currently anticipated as we gain experience.

Phase & Step	Task Summary	Primary Owner	Delivered by Core Project (Reusable Engine)	Delivered by Founder (Domain Configuration)
Step 8: Digital Twin	Deploy backend engine + code body	**Vertical Founder**	Core matching services & DB bindings	Seeding reference libraries, loading synthetic cohorts
Phase 4: UI & Stories				
Step 9: Front-end UI	Vibe-code layout & Mock Auth	**Vertical Founder**	REST API endpoints for profiles & matches	Custom HTML/CSS/JS frontend & Simulation Switcher
Step 10: Story Tuning	Calibrate GMS deal narratives	**Joint Effort**	Generative Match Story (GMS) core logic	Localized prompt styling, privacy gates & consent checks
Phase 5: Launch				
Step 11: Marketing	Write onboarding & user guides	**Vertical Founder**	Baseline templates & global whitepapers	Branded local onboarding guides & value propositions
Step 12: Maintenance	Curator loops & future roadmaps	**Vertical Founder**	Curator Gap Signal UI & auto-inference loops	Monitoring alerts, researching new ACRs, growth roadmap

4. Explanatory Notes & Technical Walkthrough

The following sections provide the detailed technical explanations and execution steps for each of the 12 process blocks outlined in the responsibility matrix above.

Preliminary & Provisional: Designs may evolve differently than currently anticipated as we gain experience.

Phase 1: Establish the Infrastructure

In this initial phase, you will set up a local developer sandbox on your machine to begin custom configuration.

Step 1: Clone the Repositories and Launch Docker Instances

> ***Responsibility Allocation:*** * ***Core Platform Role:*** *Reusable pre-packaged Docker container images and standard Compose configurations.* * ***Vertical Founder Role:*** *Local file cloning, terminal execution, and regional cloud hosting deployment.*

You do not need to compile complex source code from scratch. The DeeperPoint components are pre-packaged as lightweight **Docker containers** that run identically on any operating system.
1. **Clone the repositories:** Download the core open-source repositories (such as `Cosolvent` and `CommonContext`) and establish your vertical configuration directory on your local machine.
2. **Launch the sandbox:** Navigate to the project directory in your terminal and run `docker compose up --build`. This automatically pulls the pre-built databases (with `pgvector` enabled for AI search) and spins up the localized microservices.

Step 2: Configure Roles, Custom Schemas, and Authentication Parameters

> ***Responsibility Allocation:*** * ***Core Platform Role:*** *Flexible `marketplace.yaml` validation engine and base database profile structures.* * ***Vertical Founder Role:*** *Custom domain definition, role scoping, database credentials, and security configurations.*

Once the local containers are running, you will customize the authentication models and participant schemas to fit your specific vertical.

1. **Define Participant Roles & Subtypes:** Define who will participate in the marketplace. The standard Cosolvent framework operates on a highly generic, extensible role model. While these are customized at launch, the underlying schema compiles them into three core generic categories, accommodating a variety of distinct subtypes when transactions are complex:
    - **Suppliers / Providers / Caregivers / Sellers (`Supply`):** The side offering specialized capabilities, physical assets, technical products, services, or professional care.

- **Consumers / Recipients / Buyers / Patients / Clients (`Demand`):** The side requiring, purchasing, or receiving those specialized capabilities, products, services, or care.
- **Intermediaries / Brokers / Logistics / Service Providers / Inspectors (`Facilitators`):** Third-party entities that facilitate the transaction. When transactions are complicated, the system makes provision for a variety of distinct service provider roles—such as quality inspectors who certify standards, logistics providers who handle fulfillment, brokers who arrange trade finance, or local delegates who witness physical execution.

2. **Design Custom Onboarding & Profile Schemas:** Every participant role requires a custom data schema inside your vertical code body. In `marketplace.yaml`, you must declare the attributes that define a valid user. For example:
   - *For a specialty grain supplier profile (agricultural trade example):* collect certified protein contents, organic credentials, and storage facility metrics.
   - *For a consulting buyer profile (professional services example):* collect conflict-of-interest requirements, target timelines, and specialized case briefs.
3. **Configure Sponsor Authentication Parameters:** Establish administrator passwords, secure API keys for translation/LLM services (OpenRouter, Whisper), and database credentials.

Phase 2: Curate the CommonContext (The Knowledge Base)

A standard search engine fails in thin markets because it lacks domain expertise. In this phase, you will feed your industry's standards and rules into the system to ground the AI matching engine in commercial reality.

Step 3: Populate Starter Documents and Standards

> ***Responsibility Allocation:*** * ***Core Platform Role:*** *Parsing engines and text conversion services (`convert_pdf`, `convert_url`).* * ***Vertical Founder Role:*** *Identifying, collecting, and feeding industry reference documents, contracts, and standards.*

Identify the core legal, technical, and regulatory reference documents that govern your industry. 1. **Gather the files:** Collect standard contract templates (e.g., GAFTA contracts in grain trading, CWB welding codes, ISO metrology standards, or local zoning guidelines). 2. **Convert to clean**

text: Upload these PDFs or webpage URLs directly into the CommonContext ingestion tool. The system automatically converts them into structure-aware Markdown, preserving clause numbers, tables, and section hierarchies.

Step 4: Extract and Edit the Metadata Schema

> ***Responsibility Allocation:*** * ***Core Platform Role:*** *Automated AI Schema Analyzer and extraction pipeline.* * ***Vertical Founder Role (Joint Effort):*** *Reviewing, calibrating, and refining the generated terminology to match operator vocabulary.*

To make these documents searchable by the AI, they must be tagged using a standard taxonomy. 1. **Generate the schema:** Run the AI Schema Analyzer on your starter documents. The AI will propose a unified `marketplace.yaml` metadata schema (e.g., lists of countries, specific grain grades, material tolerances, or required inspector certifications). 2. **Refine the vocabulary:** Review the generated schema and edit it to ensure that the terms match the exact professional language used by local operators.

Step 5: Search for Alternative Compliance Channels (Escape Hatches)

> ***Responsibility Allocation:*** * ***Core Platform Role:*** *Alternative Compliance Rule (ACR) database models and Extraction Prompts.* * ***Vertical Founder Role:*** *Researching regional regulatory bulletins and executing the escape hatch extraction pipeline.*

This is where you transform the system from a simple catalog into a transaction-generating powerhouse. Most regional transactions are blocked by high-cost, rigid regulatory standards (e.g., requiring a specific costly certification or a centralized physical audit). However, regulators almost always write **“escape hatches” or alternative pathways** (delegated witnessing, risk-tiered exemptions, historical performance calibration) into supplementary bulletins. 1. **Ingest auxiliary guidelines:** Search for and upload regional regulatory bulletins, appeal committee minutes, or small business compliance guides. 2. **Extract the bypass rules:** Run the Alternative Compliance Extraction pipeline. The system will detect these escape hatches and model them as structured **Alternative Compliance Rules (ACRs)** with specific trigger conditions (e.g., in engineering services, *“testing requires a certified inspector signature in lieu of centralized facility accreditation”*, or in professional services, *“senior partner oversight witnesses and signs the review in lieu of a third-party institutional audit”*).

Step 6: Update the Vertical Schema

> ***Responsibility Allocation:*** ** **Core Platform Role:** Underlying metadata-filtered vector search queries. * **Vertical Founder Role:** Writing the concrete operational triggers back into the vertical's core `marketplace.yaml` configuration.*

Integrate the newly discovered alternative triggers directly back into your core metadata schema. This ensures the matching engine explicitly tracks the physical parameters (e.g., specialty storage certifications) or operational credentials (e.g., licensed inspector signatures) necessary to activate these bypass routes.

The CommonContext Knowledge Feedback Engine: Continuous Performance Improvement

The DeeperPoint toolkit does not treat industry knowledge as a static, pre-loaded database. Instead, **CommonContext operates as a learning engine that continuously improves site performance** through closed-loop information feedback.

As a marketplace sponsor, you do not need to map every possible regulatory exception before launching. The platform is designed to progressively absorb, structure, and scale industry workarounds through three distinct feedback loops.

Loop 1: The Initial Review & Editing Loop (Setup Phase)

- **Responsibility Split:** Core provides the analyzer extraction pipelines; Founder reviews and maps parameters.
- **Vocabulary Calibration:** You align LLM-proposed terms with the exact phrasing used by local operators.
- **Workaround Parameter Mapping:** When you ingest regulatory bulletins and discover alternative compliance "escape hatches" (Step 5), the extracted trigger conditions (such as physical bounds or personnel certifications) are fed directly back into your metadata schema (Step 6).
- **Result:** This loop optimizes the matching engine's accuracy, ensuring that it compares profiles using commercially precise metrics.

Loop 2: The Inbound Profile Enrichment Loop (Real-Time Participant Phase)

- **Responsibility Split:** Core automatically generates notifications when gaps block high-scoring matches; Founder customizes channel prompts.
- **Channel Prompts:** When the engine identifies a high-scoring match that is blocked only because a specific trigger condition is unknown (e.g., *"Does the provider carry the required local certification?"*), the platform sends a targeted prompt via WhatsApp or email.
- **Dynamic Profile Updates:** The participant's reply (text or voice) is parsed by the LLM, volunteers the missing attribute, and updates their profile in the database.
- **Result:** The matching engine immediately re-scores the match, instantly unlocking the transaction. The participant's profile is permanently enriched for all future matches.

Loop 3: The Curator Pull Loop (Ongoing Marketplace Maintenance Phase)

- **Responsibility Split:** Core surfaces Gap Signals on the sponsor dashboard; Founder curator researches and approves new ACRs.
- **Gap Signal Generation:** When a transaction is blocked by a standard that has no matching Alternative Compliance Rule (ACR) inside the CommonContext vector store, the engine generates a **Curator Pull Signal** on the sponsor dashboard.
- **Targeted Curation:** Rather than researching the entire web, the sponsor curator is directed *exactly* to the regulatory gap (e.g., *"We need an alternative pathway for a rigid institutional audit standard"*).
- **AI-Assisted Ingest:** The sponsor runs a targeted crawl, extracts the new exception (e.g., a delegated witnessing policy or risk-tiered exemption), and approves the new ACR.
- **Result:** Once ingested, **this newly discovered compliance escape hatch immediately applies to all future matches in that sector.** The platform becomes progressively smarter with every blocked match, continuously reducing transaction friction and improving marketplace liquidity.

Phase 3: Stress-Test via Custom Cohort Simulation (The Digital Twin)

Before inviting real, high-value clients to your marketplace, you must prove that the matching physics actually work and do not generate false matches.

Step 7: Apply ClientSynth Methodology to Design and Generate Synthetic Cohorts

> ***Responsibility Allocation:*** ** **Core Platform Role:** Base simulator schema configurations and cohort verification tests. * **Vertical Founder Role:** Custom-designing the synthetic profile parameters, interpreting trade logic, and executing the generation pipeline to synthesize diverse, realistic cohorts.*

Instead of writing fake profiles by hand, you will apply **ClientSynth** cohort design principles to synthesize a realistic population of simulated users. Because generating realistic participant behavior is highly complex, this requires careful domain interpretation rather than an off-the-shelf automated script: 1. **Feed the schema:** Input your custom `marketplace.yaml` metadata schema and discovered compliance workarounds into the synthesis tool. 2. **Partition the population:** Carefully design three distinct behavioral groups of synthetic users to stress-test your matching equations: * *The Standard Cohort:* Profiles possessing standard, high-cost credentials (e.g., a fully certified, high-overhead provider). * *The Treatment Cohort (using a typical test persona like "Dave"):* Non-certified profiles that explicitly carry the precise workaround triggers (e.g., a low-overhead local provider who possesses the specific delegated inspector signature). * *The Control Cohort:* Standard profiles lacking credentials *and* lacking triggers.

Step 8: Construct and Run the Digital Twin (Wiring the Engine to the Vertical Code Body)

> ***Responsibility Allocation:*** ** **Core Platform Role:** Reusable Cosolvent database structures and pgvector similarity search microservices. * **Vertical Founder Role:** Directory volume mounting, database seeding scripts, and simulation loop execution.*

A raw back-end matching service (Cosolvent) is useless on its own. The **Digital Twin** is created by combining the core **Cosolvent Docker back-end** with your **custom market vertical code body**.

1. **Mount the Configuration:** Mount your custom `marketplace.yaml`, vertical schemas, and curated CommonContext reference documents into the Cosolvent Docker container environment (using volume mounts).
2. **Seed the Simulator:** Run the `seed_reference_library.py` script to load the CommonContext documents and Alternative Compliance Rules (ACRs) into the PostgreSQL/pgvector database.

3. **Inject Synthetic Users:** Load the custom-designed synthetic buyer, seller, and facilitator cohorts into the Cosolvent profile database.
4. **Launch the Unified Sandbox:** Once the back-end has the rules, documents, and synthetic users loaded, it acts as a fully integrated Digital Twin. This allows developers to run automated matching loops and stress-test the system:
   - Verify that the *Standard Cohort* matches immediately.
   - Verify that the *Treatment Cohort* (using our "Dave" test persona) successfully triggers the alternative compliance "escape hatches" and bypasses rigid blockers.
   - Verify that the *Control Cohort* remains blocked, correctly triggering the Loop 1 profile-enrichment notification to prompt for missing certification data.

Phase 4: Vibe Code the Frontend UI/UX and Fine-Tune Stories

With the backend Digital Twin successfully populated and simulated, the next step is custom "vibe coding" the front-end to bring the marketplace to life.

Step 9: Vibe Code a Specialized Frontend Client

> ***Responsibility Allocation:*** ** **Core Platform Role:** RESTful APIs for profile retrieval, match-making, and vector searches. * **Vertical Founder Role:** Vibe-coding the look & feel, custom role forms, and Simulation Switcher.*

You do not need a heavy, production-grade frontend engineering team to construct the initial interface. Using an LLM, you can "vibe code" a lightweight, high-fidelity HTML/CSS/JS frontend that communicates directly with the Cosolvent and CommonContext REST API endpoints.

1. **Vibe Code the Vertical Look & Feel:**
   - Create a premium visual environment tailored to your industry. For specialty manufacturing, design a technical, high-contrast grid-overlay layout in deep slates and emerald accents (denoting engineering precision). For agricultural trade, design clean dashboards in warm tones with visible cargo tracking grids.
   - Ensure all typography uses modern google fonts (e.g. Outfit, Inter) rather than browser defaults, and utilize smooth transitions to make the UI feel responsive and premium.
2. **Build Custom Role Onboarding Forms:**

- Generate form-filling pages dynamically mapped to the participant profile schemas defined in Step 2.
- Provide intuitive upload fields where users can submit PDFs (such as machine calibration records or welder certificates) directly, triggering the back-end extraction pipeline.

3. **Vibe Code a "Simulation Switcher" (Mock Authentication):**
   - Production-grade OAuth (e.g., Auth0, Firebase) is a blocker during prototyping. To make the Digital Twin useful for walkthroughs, implement a developer-friendly **Simulation Switcher** dropdown directly in the header of the frontend.
   - For instance, in our testing sandbox, this dropdown lets you instantly impersonate different synthetic users in the Digital Twin. You can select a service provider like "Dave" to view his matching queue and accept data-sharing consent prompts, then click the switcher to instantly hop to a buyer like "Northern Mine" to read the resulting step-by-step Generative Match Story and view Dave's credentials from their side.

Step 10: Fine-Tune the Generative Match Story (GMS)

> ***Responsibility Allocation:*** ** **Core Platform Role:** GMS narrative synthesis engine, base LLM system prompts, and search bindings. * **Vertical Founder Role (Joint Effort):** Adapting narrative templates to domain jargon and verifying data privacy gates.*

When a workaround match is found, the system presents it as a **Generative Match Story**. Rather than returning a dry regulatory clause, the GMS drafts a plain-English, step-by-step narrative showing exactly how a deal can legally and physically unfold between these two specific parties.

1. **Test GMS templates:** Run match stories for your synthetic cohorts and inspect the output. 2. **Refine readability:** Ensure the AI avoids dense economic jargon (like *"bilateral cold-start physics"*) and translates terms into direct business plans (for example: *"Dave conducts the inspection locally, signs the certified report, and submits it to the sponsor on the buyer's behalf"*).
3. **Validate privacy gates:** Verify that sensitive information (like specific pricing ranges or machinery metrics) is hidden behind Loop 3 privacy consent gates before being integrated into the deal story.

Phase 5: Launch and Strategic Growth

In this final phase, you transition from developer mode to operational launch.

Step 11: Write Vertical-Branded Onboarding and Marketing Assets

> ***Responsibility Allocation:*** ** **Core Platform Role:** Baseline framework documentation, core whitepapers, and guides. * **Vertical Founder Role:** Writing industry-focused marketing copy, value sheets, and regional user manuals.*

To drive adoption, translate the technical capabilities of the DeeperPoint toolkit into clear, localized value propositions. 1. **Draft user guides:** Write simple, jargon-free step-by-step guides showing buyers and sellers how to onboard, upload specifications, and navigate matching. 2. **Create marketing material:** Focus entirely on your vertical's specific solutions (for example: *"How local providers can secure compliant matching in 48 hours at a fraction of the standard cost"*). Do not mention databases, pgvector, or ClientSynth. Focus on the temporal, spatial, and financial metrics unlocked by your platform.

Step 12: Write a Vertical-Specific Roadmap

> ***Responsibility Allocation:*** ** **Core Platform Role:** Sponsor administration interfaces, Gap Signal UI, and future API extensions. * **Vertical Founder Role:** Sourcing future corridor expansions, training curators, and managing local updates.*

As real-world users join, they will uncover new regulatory barriers and operational desires. 1. **Establish the curator loop:** Train your system curators to monitor the Curator Pull Signals dashboard, picking up missing documents highlighted by blocked user matches. 2. **Map future feature expansions:** Draft a long-term development roadmap showing how the platform will expand to adjacent corridors, integrate regional trade finance modules, or incorporate native digital signing layers to close transactions directly inside the platform.

---

5. The Go-Live Checklist

Before deploying your marketplace to production, ensure you can check off every item in this operational checklist:

- **Infrastructure:** Docker containers are configured for high-availability cloud hosting (e.g., AWS Elastic Beanstalk).
- **CommonContext:** Authority reference library is populated with core industry contracts, standards, and regulatory escape hatches.
- **Simulated Validation:** ClientSynth successfully ran 100+ synthetic match scenarios with zero false-positive matches.
- **Generative Stories:** GMS output has been reviewed for professional voice, plain-English clarity, and active privacy consent gates.
- **UI/UX Branding:** Frontend interface uses the vertical's design guidelines, matches modern typography, and renders responsively on mobile viewports.
- **Stakeholder Assets:** Marketing copies, user guides, and regional launch roadmaps are ready to be distributed to early adopters.

By systematically following this guide, you will transition your target industry from a fragmented, high-friction space into a highly connected, fluid, and trust-driven digital ecosystem.
