

How to Build a Parametric-Insurance or Data-Rights Marketplace Without a Central Arbiter

You want a marketplace where a payout, a data license, or a capacity slot is released only to a party that can prove eligibility, and where no single participant can drain the pool or replay a claim. This guide walks through an architectural approach that ties every allocation to an eligibility proof and bounds it against abuse, without routing trust through one cloud arbiter. It describes an architecture disclosed in U.S. Provisional Application No. 64/049,409, not a shipping library, and its home inventive step is the Governed Marketplace inventive step.

What You Are Building

You are building a marketplace where the thing being allocated is valuable and abusable at the same time: an insurance payout, a license to someone's personal data, a reserved unit of capacity, an attestation. In each case the marketplace has to answer two hard questions on every transaction. Is this party actually eligible for what they are asking to receive? And can any party, by replaying, flooding, or colluding, extract more than the rules allow?

Two concrete instances motivate the pattern. A parametric-insurance marketplace pays out automatically when a defined trigger occurs (a wind speed crossed, a flight delayed, a shipment's cold chain broke) rather than after a manual claims adjuster reviews a loss. A data-rights marketplace lets individuals offer their own observations under a license they control, and pays them when a consumer acquires those observations under terms they consented to.

The architecture below is drawn from the governed marketplace primitive disclosed in U.S. Provisional Application No. 64/049,409. It is a design you implement yourself, not a package you install. What it gives you is a shape: allocation is bound to an eligibility proof, settled bilaterally between the two transacting parties, and recorded as a reconstructable lineage instead of a private ledger held by a platform operator.

Why the Obvious Approaches Fall Short

The default is a central platform. One operator runs the accounts, holds the funds, decides who is eligible, settles the transaction, and keeps the transaction log. This works, and most real marketplaces are built this way. Its structural cost is that trust, identity, settlement, and dispute resolution all collapse into a single operator whose internal records are the only account of what happened. Contributing and consuming parties cannot independently reconstruct a transaction; they see what the operator shows them. Retire or replace the operator and the history is at risk.

Public blockchains remove the single operator but change the tradeoff rather than eliminate it. Settlement goes through distributed consensus, which gives block-granularity finality on the order of minutes and imposes per-transaction consensus overhead. Accounts are addressed by public-key pairs, so a determined actor can spin up many addresses. These are real properties, not flaws to be embarrassed about, but they are a poor fit for a marketplace that wants per-transaction finality and resistance to identity multiplication.

The structural gap in both is the same: eligibility and the allocation it authorizes are not welded together in a way any party can verify after the fact. A central operator asserts eligibility; a chain records a transfer but does not, by itself, know whether the recipient was entitled to it. The approach here closes that gap by making the eligibility proof part of the settlement artifact.

The Architecture

The disclosed approach rests on a few composed primitives. Each is described in the filing; none is a product you can download.

Authority-credentialed identity and an authority taxonomy. Every party (an insurer, an individual offering their data, a consumer, a regulator, an oracle) acts under an authority credential, and credentials carry taxonomy semantics so the system can differentiate behavior by the source's governance authority rather than treating all authenticated messages homogeneously. Eligibility is not a boolean the operator flips; it is expressed as which credential a party holds and what that credential is authorized to do.

Composite admissibility, not a binary allow/deny. A proposed transaction is run through a composite admissibility evaluator that produces one of several graded outcomes: accepted, gated, deferred, or rejected. This is the core of "cryptographically tied to eligibility proofs." An allocation does not proceed because a central service said yes; it proceeds because the matched observations, credentials, and conditions pass admissibility, and the evaluation is recorded.

Matched-pair settlement as the exchange substrate. The disclosed settlement primitive settles a bilateral exchange through two governance-credentialed observations: a first observation representing an offer, tender, claim, or commitment, and a second observation representing acceptance or fulfillment. The pair must fall within a governance-policy-defined spatial window and temporal window, each party's

authority credential is checked, the pair is admitted through composite admissibility, and a cryptographically bound settlement artifact is produced that supports non-repudiation. Settlement is direct between the two parties, without a third-party intermediary and without distributed consensus, at mesh-propagation latency. Consent is explicit and per-transaction, carried by the paired observations, rather than implied once at account setup.

Escrow and conditional release for triggers. For anything that should pay only when a condition holds (the parametric case), the disclosed escrow and chained-settlement mechanism places settlement content under credentialed escrow with an explicit release condition, and a release evaluator applies the same composite admissibility to release requests. The filing enumerates release patterns including event-triggered escrow bound to credentialed events, oracle-attested escrow where credentialed oracles attest external conditions, time-locked escrow, and multi-party escrow. A parametric payout is exactly this: escrow released on a credentialed trigger observation.

The marketplace primitive itself. Above settlement sits the governed marketplace primitive: a commodity schema registrar, a participant admission mechanism for credentialed contributors and consumers, discovery and matching of offers to interests, a pricing mechanism that explicitly includes a parametric form alongside fixed-price, auction, negotiated, and dynamic, and a licensing framework specifying permitted use, retention, aggregation, and redistribution rights per commodity. It integrates matched-pair settlement, extends dispute resolution to marketplace disputes, and supports a marketplace-operator-service interface as an *optional* participant rather than a required trust mediator.

Instance parameterization. The filing is explicit that these are the same primitive under different parameters. It names, among others, an insurance-risk marketplace instance where insurers offer risk coverage with parametric payout settlement on credentialed trigger observations, and a personal-data marketplace instance where

individuals offer their own observations under privacy governance through direct matched-pair settlement, plus an observation marketplace where observations are licensed individually or aggregated. Parametric insurance and a data-rights marketplace are two configurations of one architecture, not two systems.

Bounding against abuse. Two mechanisms in the filing do the bounding. The capability envelope constrains what a party or a place can be allocated: a capacity-exchange instance offers capability-envelope allocations (berths, charging stations, slots), and participation is gated by the envelope rather than unbounded. Separately, the disclosed preemption-budget enforcer limits the rate at which a credentialed actor can exercise a high-authority action, so a valid credential does not become an unlimited lever. Every rejection, timeout, and failure is recorded in lineage, which is what makes adversarial-failure patterns detectable after the fact.

Governance-chain lineage. Every step (offer, match, binding, escrow placement and release, negotiation round, failure, dispute, downstream consumption) is recorded in a governance-chain lineage field. The transacting parties and credentialed regulators can reconstruct the transaction independently. This is the substitute for the central operator's private log.

How to Approach the Build

The following is an ordered path for implementing the architecture yourself. The interface sketches are illustrative and faithful to the filing; they are not a supplied SDK.

1. **Define your authority taxonomy first.** Enumerate the party classes for your instance (insurer, insured, oracle, regulator for parametric; data subject, consumer, aggregator, regulator for data-rights) and what each credential authorizes. Eligibility rules live here, not in application code scattered downstream.

- 2. Fix the commodity schema and license.** Decide exactly what one unit of the commodity is (a coverage contract keyed to a trigger; a bundle of observations with a retention and redistribution license). The license terms travel with the commodity, so encode permitted use, retention, aggregation, and redistribution explicitly.
- 3. Model the two observations of a settlement.** For parametric insurance, the first observation is the coverage offer or the bound policy; the second is the fulfillment on trigger. For data-rights, the first is the individual's offer under their license; the second is the consumer's acceptance. Sketch of the recognition step:

```
// Illustrative only, faithful to the disclosed primitive
pair = recognize(firstObs, secondObs)
  require sameWindow(firstObs, secondObs, policy.spatialWindow, policy.temporalWindow)
  require credentialAuthorizes(firstObs.credential, action)
  require credentialAuthorizes(secondObs.credential, action)
  require admissible(pair) // accepted | gated | deferred | rejected
  bind(pair) // cryptographic, non-repudiable artifact
```

- 4. Choose your matching and window rules per transaction type.** The filing lists recognition rule forms (content-matching, cryptographic handshake, spatial or temporal coincidence, authority-pair, derivation-chain). A parametric payout leans on authority-pair plus the trigger derivation; a data license leans on content-matching plus handshake.
- 5. Wire escrow for anything conditional.** Put the payout under credentialed escrow with the trigger as the release condition, and require the release path to pass the same admissibility evaluation. For parametric triggers, use the oracle-attested or event-triggered escrow pattern so the external condition arrives as a credentialed observation, not an unauthenticated feed.

6. **Add the bounds before you go live.** Attach capability-envelope limits so no consumer can acquire beyond an allocated envelope, and attach a preemption-budget so no credential (including a compromised one) can fire high-authority releases without rate limits. Decide your failure responses (silent timeout, notify-and-retry, escalate-to-authority, rollback-with-lineage) per the filing's enumerated set.
7. **Make lineage the source of truth.** Record every step in the governance chain and give parties and regulators a reconstruction path. Treat dispute resolution as reading assembled lineage as evidence, per the disclosed dispute mechanism, rather than as re-litigating from a private log.

What This Does Not Give You

This is an architecture, not a drop-in library. There is no package to install and nothing here "just works" out of the box; you implement the primitives, the credential system, the escrow logic, and the lineage store yourself, against your own substrate and regulatory context. The approach is disclosed in a patent filing. It has not been presented here as a benchmarked or production-proven system, and this guide states no performance numbers because the filing is a disclosure of method, not a measured product.

The approach also does not remove the hard external dependencies. A parametric marketplace is only as trustworthy as its trigger oracles; the architecture credentials and attests them, but it does not manufacture a correct oracle for you. Credentialing is an authority problem: someone still has to issue and govern credentials, and the taxonomy you design is a policy decision the architecture executes, not one it makes for you. Where you genuinely need a regulated intermediary (custody of fiat funds, for instance), that intermediary participates as an optional credentialed party, but its presence reintroduces the trust you were trying to distribute. And bilateral matched-

pair settlement fits exchanges that admit paired-observation structure; a market whose transactions cannot be expressed as an eligibility-checked pair is outside where this pattern applies cleanly.

Disclosure Scope

The architecture described in this guide is disclosed in U.S. Provisional Application No. 64/049,409, whose governed marketplace primitive, matched-pair settlement primitive, escrow and conditional-release mechanism, capability envelope, and preemption-budget enforcer are the basis for every mechanism claimed above. This guide is educational. It teaches how to approach building such a marketplace and is not a warranty, a specification of a shipping product, or an offer of software. Any implementation, and its compliance with applicable insurance, data-protection, and financial regulation, is the responsibility of the party that builds it.

Governed Marketplace (</governed-marketplace>) [All 40 steps → \(/inventive-steps\)](/inventive-steps)

No platform operator. Governance-chain lineage replaces the trusted intermediary.

Provisional application

PRIMARY TECHNICAL DISCLOSURE

- [Governed Marketplace: Platform-Less Trust Through Governance-Chain Lineage \(/articles/governed-marketplace-platform-less-trust-through-governance-chain-lineage\)](/articles/governed-marketplace-platform-less-trust-through-governance-chain-lineage)

SECONDARY TECHNICAL

- [No-Platform-Operator Marketplace \(/articles/governed-marketplace/no-platform-operator\)](/articles/governed-marketplace/no-platform-operator)
- [Governance Chain as Trust Substrate \(/articles/governed-marketplace/governance-chain-trust-substrate\)](/articles/governed-marketplace/governance-chain-trust-substrate)
- [Commodity Class Plurality \(/articles/governed-marketplace/commodity-class-plurality\)](/articles/governed-marketplace/commodity-class-plurality)
- [Regulatory-Audit-Native Marketplace \(/articles/governed-marketplace/regulatory-audit-native\)](/articles/governed-marketplace/regulatory-audit-native)

- [Cross-Marketplace Composition \(/articles/governed-marketplace/cross-marketplace-composition\)](/articles/governed-marketplace/cross-marketplace-composition)
- [Spectrum Exchange Marketplace Embodiment \(/articles/governed-marketplace/spectrum-exchange\)](/articles/governed-marketplace/spectrum-exchange)
- [Capacity Allocation Marketplace \(/articles/governed-marketplace/capacity-allocation\)](/articles/governed-marketplace/capacity-allocation)
- [Port Berth Allocation Marketplace \(/articles/governed-marketplace/port-berth-marketplace\)](/articles/governed-marketplace/port-berth-marketplace)
- [Charging Station Capacity Marketplace \(/articles/governed-marketplace/charging-station-marketplace\)](/articles/governed-marketplace/charging-station-marketplace)
- [Runway Slot Allocation Marketplace \(/articles/governed-marketplace/runway-slot-marketplace\)](/articles/governed-marketplace/runway-slot-marketplace)

APPLICATIONS · GENERAL

- [Parametric Insurance Settlement on Credentialed Trigger Observations: A Governed Risk-Coverage Marketplace \(/articles/governed-marketplace/parametric-insurance\)](/articles/governed-marketplace/parametric-insurance)
- [Infrastructure Capacity Marketplace: Trading Grid, Runway, Rail, and Berth Capacity Without a Central Operator \(/articles/governed-marketplace/infrastructure-capacity-marketplace\)](/articles/governed-marketplace/infrastructure-capacity-marketplace)
- [Slot Allocation Markets for Runways, Berths, and Track Time \(/articles/governed-marketplace/infrastructure-slot-allocation\)](/articles/governed-marketplace/infrastructure-slot-allocation)
- [A Credentialed Spectrum Marketplace for Primary Auctions, Secondary Trading, and Dynamic Sharing \(/articles/governed-marketplace/spectrum-marketplace-ecosystem\)](/articles/governed-marketplace/spectrum-marketplace-ecosystem)
- [Distributed Compute Marketplace: Bilateral GPU Settlement Without a Platform Operator \(/articles/governed-marketplace/compute-marketplace-distributed\)](/articles/governed-marketplace/compute-marketplace-distributed)
- [Data Rights Marketplace for the EU Data Act and Data Governance Act: Neutral, Pair-Settled Personal-Data Exchange \(/articles/governed-marketplace/data-rights-marketplace\)](/articles/governed-marketplace/data-rights-marketplace)
- [IoT Data Monetization Without Platform Capture: A Governed Data Marketplace for the EU Data Act \(/articles/governed-marketplace/iot-data-monetization\)](/articles/governed-marketplace/iot-data-monetization)
- [Cross-Registry REC and Guarantee of Origin Settlement Without a Central Platform Operator \(/articles/governed-marketplace/renewable-energy-attribute-credits\)](/articles/governed-marketplace/renewable-energy-attribute-credits)
- [Water Rights Trading Platform Without a Central Operator: A Governed Marketplace for Western and Murray-Darling Basins \(/articles/governed-marketplace/water-rights-marketplace\)](/articles/governed-marketplace/water-rights-marketplace)
- [CBRS SAS Coordination and PAL Leasing Without a Central Operator: A Governed-Marketplace Architecture \(/articles/governed-marketplace/fcc-cbrs-saas\)](/articles/governed-marketplace/fcc-cbrs-saas)

APPLICATIONS · SPECIFIC

- [CME Globex vs a Governed Marketplace With No Privileged Operator \(/articles/governed-marketplace/cme-globex-marketplace\)](/articles/governed-marketplace/cme-globex-marketplace)
- [Federated Wireless CBRS vs a Governed Spectrum Marketplace \(/articles/governed-marketplace/federated-wireless-cbrs\)](/articles/governed-marketplace/federated-wireless-cbrs)

- [ICE Alternative: Governed Marketplace Beyond Per-Venue Exchange Integration \(/articles/governed-marketplace/ice-exchange-platform\)](/articles/governed-marketplace/ice-exchange-platform).
- [AWS Data Exchange vs a Governed Marketplace Substrate \(/articles/governed-marketplace/aws-data-exchange-marketplace\)](/articles/governed-marketplace/aws-data-exchange-marketplace).
- [Blockchain Marketplace Alternative: Governed Marketplace Beyond DEX and NFT Platforms \(/articles/governed-marketplace/blockchain-platforms-marketplace\)](/articles/governed-marketplace/blockchain-platforms-marketplace).
- [GridX Alternative: Governed Energy Marketplace Beyond Operator-Mediated DER Orchestration \(/articles/governed-marketplace/gridx-energy\)](/articles/governed-marketplace/gridx-energy).
- [CME Globex vs a Governed Marketplace: Composing Cleared and Bilateral Trades \(/articles/governed-marketplace/openexchange-cme\)](/articles/governed-marketplace/openexchange-cme).
- [Snowflake Marketplace Alternative: Cross-Platform Governed Exchange \(/articles/governed-marketplace/snowflake-marketplace\)](/articles/governed-marketplace/snowflake-marketplace).

[Governed Marketplace overview → \(/governed-marketplace\)](/governed-marketplace).