# **1. Executive Summary**

# **1.1 Objective & Scope**

**Objective:** To design, build, and deploy an institutional-grade, multi-chain trading system that unifies advanced routing, liquidity provisioning, flash lending, liquidity mining, mempool analytics, and privacy-preserving execution. The product will evolve in **36 structured rollout stages**, ensuring progressive validation, resilience against edge cases, and institutional readiness.

## **Scope of Rollout (36 Stages)**

### **A. Use Cases (27 Stages)**

Each of the 9 primary use cases is broken into **3 rollout stages**:

1. **MVP implementation** (basic functionality, narrow scope, single chain/venue)
2. **Scaling/expansion** (multi-chain integration, more venues, performance tuning)
3. **Institutionalization** (risk controls, compliance hooks, observability, governance)

**Use Cases**

1. **Single-Chain Swap Optimization**
2. **Cross-Chain Routing**
3. **Liquidity Provisioning (LPing, range management)**
4. **Flash Loan Arbitrage / Atomic Bundles**
5. **Liquidity Mining / Gauge Optimization**
6. **Long-Tail Token Routing & Price-Impact Awareness**
7. **Private/Intent-based Execution**
8. **Institutional Treasury Execution**
9. **Solana Orderflow Routing (Raydium/Jupiter/Jito)**

This yields **27 stages** (9 × 3).

### **B. Edge Cases (9 Stages)**

Each edge case is treated as a distinct rollout stage because it requires separate risk and technical handling. Suggested edge cases:

1. **Bridge Failures** — atomic cross-chain swaps failing due to bridge downtime or liquidity drain.
2. **Oracle Divergence** — discrepancy between on-chain pool pricing and external oracles.
3. **Extreme Gas Spikes** — sudden congestion (e.g., NFT mint storms, liquidations) impacting fill reliability.
4. **Mempool Manipulation** — sandwich or back-run threats, malicious ordering.
5. **Liquidity Rug / Pool Drain** — exit scams, permissioned pool upgrades, or whale withdrawals.
6. **Cross-DEX Desync** — stale subgraph/indexer data vs live pool state leading to mis-quotes.
7. **Stablecoin Depegs** — critical path dependence on USDC/USDT/DAI, including cascading effects.
8. **Chain Halts / Reorgs** — Solana liveness halts, Ethereum L2 reorgs, etc.
9. **Governance/Upgrade Risk** — protocol upgrade that breaks router assumptions (e.g., Balancer v2→v3).

1.2 Key outcomes (alpha sources, cross-chain reach, latency targets, uptime)  
 1.3 Non-goals (e.g., CEX connectivity, custodial user funds if out of scope)  
 1.4 Stakeholders & ownership (PM, Tech Lead, Security, Compliance, Ops)

# **2. Problem Statement & Use Cases**

2.1 Market problems (fragmented liquidity, MEV risk, cross-chain UX)  
 2.2 Primary use cases

* UC-1: Single-chain swap optimization
* UC-2: Cross-chain routing (bridge + DEX aggregation)
* UC-3: Market-making / LP management (rebalance, fee capture)
* UC-4: Flash-loan arbitrage/repay bundles
* UC-5: Liquidity mining / gauge optimization (ve(3,3), vote bribes)
* UC-6: Long-tail token routing with price-impact constraints
* UC-7: Private/intents execution (dark routing / RFQ)
* UC-8: Treasury trade execution (institutional risk controls)
* UC-9: Solana orderflow routing (Raydium/Jupiter/Jito)  
   2.3 Actors & roles (Trader, Strategy Dev, Risk Officer, Compliance Officer, SRE, Auditor)

# **3. Functional Requirements**

3.1 Protocol/Network Connectivity

* EVM chains (Ethereum, L2s, alt-L1s)
* Solana stack (Raydium/Jupiter/Jito)
* DEX protocols (Uniswap-style CPMM/CLAMM, Balancer, Curve, ve(3,3) forks)
* Aggregators (1inch, 0x, Odos, OpenOcean)
* Cross-chain middleware (LI.FI/Socket/Axelar)
* Lending/flash loans (Aave, Balancer Vault, DODO)  
   3.2 Order Routing & Execution
* SOR (Smart Order Routing) with multi-path, multi-venue
* Cross-chain pathfinding (bridge+swap atomicity where possible)
* Slippage, price-impact, and depth-aware sizing
* RFQ/intents and private relay submission (Flashbots/MEV-Share)
* Gas price strategy per chain; pre- and post-trade checks  
   3.3 Liquidity Provision (LP)
* Position sizing for v3/CLAMM ranges (volatility bands, fee APR targeting)
* Auto-rebalance, compound fees, impermanent loss monitors
* Multi-pool inventory management (token balances, health checks)  
   3.4 Flash Loan & Atomic Arbitrage
* Candidate detection (oracle/graph scans, on-chain price diffs)
* Route construction (borrow→swap(s)→repay)
* Failure fallbacks, simulation, revert-safe guards  
   3.5 Liquidity Mining / Gauge Strategies
* Gauge voting schedules, veNFT/veToken management
* Bribe marketplace integration (if in scope)
* APR/APY aggregation and auto-allocation policy  
   3.6 Mempool/Gas & Pricing Estimation
* Live mempool listeners (pending tx density, inclusion times)
* Gas estimation models (per-block basefee dynamics, EIP-1559 heuristics)
* Price-impact calculators (CPMM, CLAMM tick math, Balancer math)
* Sandwich risk estimator & protection toggles  
   3.7 Privacy & MEV Mitigation
* Private transaction submission (Flashbots, RPC privacy relays)
* Tx randomization, timing jitter, dummy hops (optional)
* Bundle simulation, back-run/sandwich risk scoring  
   3.8 Risk & Compliance Controls
* Pre-trade: deny lists, token risk, oracle sanity, max exposure limits
* Post-trade: P&L accounting, VaR/vol controls, reconciliation
* KYC/AML screening if interacting with custodial treasuries (toggle)  
   3.9 Observability & Ops
* Metrics: trade latency, path quality, fill ratio, slippage error, failed routes
* Structured logs, traces (per chain, per strategy)
* Alerts: RPC health, profit deviation, stuck nonce, failed bundles
* Runbooks & rollback procedures  
   3.10 Admin & Governance
* Role-based permissions (hot/cold actions)
* Feature flags (enable/disable chain, DEX, strategy)
* Safe upgrade paths, migration scripts

# **4. Non-Functional Requirements**

4.1 Performance & Latency (p95/p99 per chain; Solana vs EVM targets)  
 4.2 Reliability & Availability (SLOs, multi-region failover)  
 4.3 Security (key custody, HSM, MPC, secrets mgmt, code audits)  
 4.4 Compliance (jurisdictional notes, record-keeping, audit trails)  
 4.5 Scalability (horizontal workers, queueing, stateless services)  
 4.6 Cost Objectives (RPC plans, relays, infra, indexing)  
 4.7 Maintainability (modular adapters, version pinning, upgrade cadence)

# **5. System Architecture**

5.1 High-Level Diagram (control plane vs execution plane)  
 5.2 Services & Components

* API Gateway (internal gRPC/REST)
* Strategy Service (signals → intents)
* Router Service (venue discovery, scoring, path assembly)
* Execution Service (tx building, signing, submission)
* Simulation Service (node-level + mev-relay simulation)
* Price/Market Data Service (oracles, subgraphs, indexers)
* Risk Engine (limits, exposure, VaR, circuit breakers)
* Portfolio/Inventory Service (balances, P&L, fee compounding)
* Compliance Service (screening, logging, evidence retention)
* Scheduler/Job Orchestrator (rebalances, claims, votes, bribes)
* Telemetry/Alerting (metrics, logs, traces)  
   5.3 Data Flows (ingest → decision → execution → settlement → accounting)  
   5.4 Deployment Topology (multi-region, chain-proximate relays)  
   5.5 External Dependencies (RPC providers, indexers, relays, aggregators)

# **6. Connectors & Adapters (by category)**

6.1 DEX Adapters

* CPMM (Uniswap v2-like)
* CLAMM (Uniswap v3-like; tick math, liquidity ranges)
* Balancer (weighted pools, stable pools)
* Curve (stables/meta/factory pools)
* ve(3,3) forks (Velodrome/Camelot/Lynex patterns)  
   6.2 Aggregator Adapters
* 1inch, 0x, Odos, OpenOcean (quoting + tx building)
* Quote merging, dedupe, and rescoring  
   6.3 Cross-Chain Adapters
* LI.FI/Socket/Axelar (route discovery, bridging + swap)
* Safety: canonical routes, origin/destination validations  
   6.4 Lending/Flash-Loan Adapters
* Aave v3 (direct), Balancer Vault, DODO flash
* Flash-swap (Uni v2) pattern  
   6.5 Solana Adapters
* Raydium/Jupiter quote/execution
* Jito bundle submission (if used)  
   6.6 Privacy/MEV
* Flashbots/MEV-Share clients
* Private RPC relays (per chain)

# **7. Strategy Layer (Alpha Sources)**

7.1 Deterministic Routing Alpha

* Venue selection: depth, fees, gas, price impact, failure risk
* Cross-venue multi-path vs single-venue constraints  
   7.2 Market-Making/LP Alpha
* Volatility bands, inventory skew, dynamic fee capture
* Rebalance policy (thresholds vs time-based)  
   7.3 Funding/Perps (if included)
* Basis trades, cash-and-carry, spread capture  
   7.4 Flash-Arb Alpha
* Price diff scanner, path constructor, bundle simulation  
   7.5 Liquidity Mining Alpha
* Gauge vote calendar, ROI modeling, bribe ROI, lock/boost models  
   7.6 Risk-Adjusted Execution
* Slippage, exposure caps, max daily loss, cooldowns

# **8. Pricing, Math & Simulation**

8.1 CPMM (k-invariant), CLAMM tick liquidity, Balancer weighted math  
 8.2 Execution price vs mid-price, fee modeling, price-impact curves  
 8.3 Gas modeling (EIP-1559 dynamics, Solana fees)  
 8.4 Monte Carlo for LP range risk & IL distributions  
 8.5 Scenario sims for flash-loan routes & bridge-latency risk  
 8.6 Backtesting framework (block-accurate, forked node simulations)  
 8.7 Deterministic test vectors for core math functions

# **9. Data, Storage & Indexing**

9.1 Data catalogue (trades, quotes, pools, blocks, mempool stats, gauges)  
 9.2 Storage (OLTP vs OLAP): Postgres for state; columnar warehouse for analytics  
 9.3 Indexing strategy (subgraphs, bespoke indexers, archive nodes)  
 9.4 Retention & GDPR considerations; PII minimization  
 9.5 Data quality SLAs, schema evolution policy

# **10. Risk Management**

10.1 Limit framework (per-asset, per-venue, per-chain, per-strategy)  
 10.2 Circuit breakers (volatility halts, oracle divergence, bridge halts)  
 10.3 Counterparty/protocol risk scoring (oracle resilience, governance risk)  
 10.4 Liquidity risk & inventory concentration limits  
 10.5 Compliance risks (sanctions, illicit use, restricted assets)  
 10.6 Operational risk (RPC outage, nonce lock, replay, reorgs)

# **11. Security Architecture**

11.1 Key management (MPC/HSM, hot vs warm vs cold, withdrawal policies)  
 11.2 Secrets management (vaults, rotation, access policies)  
 11.3 Transaction-level controls (allowlists, policy engine)  
 11.4 Smart contract scope (custom vaults, upgradability safeguards)  
 11.5 Supply-chain security (pin deps, SBOM, SLSA levels)  
 11.6 Pen-testing & audits (internal, external; testnet attack sims)  
 11.7 Threat model (STRIDE/PASTA) and mitigations  
 11.8 Incident response (playbooks, forensics, comms)

# **12. Compliance & Legal (jurisdiction-aware)**

12.1 Entity structuring and nexus analysis  
 12.2 Licensing considerations (if custody or RFQ)  
 12.3 KYC/AML policy toggles (if institutional clients)  
 12.4 Record-keeping (trade logs, approvals, audits, retention)  
 12.5 Disclosures & user agreements (if external-facing)

# **13. DevEx, Tooling & SDLC**

13.1 Repos, branching model, codeowners  
 13.2 CI/CD (lint, unit, property-based, integration, fork-node e2e)  
 13.3 Environments (dev, staging, prod; testnets)  
 13.4 Feature flag framework, canary releases  
 13.5 Versioning & migrations (adapters, ABI changes)  
 13.6 Developer SDKs (internal Python/TS client libs)

# **14. Observability & SRE**

14.1 Metrics (cardinality plan, exemplars)

* Routing: quote latency, path score, success rate
* Execution: inclusion delay, revert rate, gas delta vs quote
* Finance: realized vs expected slippage, P&L attribution  
   14.2 Distributed tracing (critical paths)  
   14.3 Alerting (SLO burn rates, chain-specific anomalies)  
   14.4 Capacity planning, autoscaling, queue back-pressure  
   14.5 Disaster recovery (RPO/RTO, backups, region failover)

# **15. User Interface & Access**

15.1 Internal console (read-only dashboards, kill-switches)  
 15.2 Strategy config UI (guard-railed params, approvals)  
 15.3 Audit console (timeline, evidence export)  
 15.4 API/CLI (authN/Z, rate limits, idempotency keys)

# **16. Testing Strategy**

16.1 Unit tests (math, adapters, limits)  
 16.2 Property-based tests (invariants for pool math)  
 16.3 Integration tests (multi-venue quoting, bridge+swap)  
 16.4 Forked-node e2e (mainnet & Solana forks)  
 16.5 Chaos tests (RPC flaps, reorgs, nonce contention)  
 16.6 Security tests (fuzzing, permission checks, replay/simulations)

# **17. Deployment & Runbooks**

17.1 Release checklist (migrations, feature flags, rollback plan)  
 17.2 Chain onboarding runbook (new RPCs, tokens, DEX allowlist)  
 17.3 Emergency procedures (key rotation, bridge halts, relay changeover)  
 17.4 Maintenance windows & comms

# **18. Performance Targets & Benchmarks**

18.1 Latency budgets (quote, route, sign, submit, inclusion)  
 18.2 Throughput (tx/min per chain)  
 18.3 Quote freshness and staleness tolerances  
 18.4 Benchmark methodology & datasets

# **19. Economics & Cost Model**

19.1 Infra budget (RPC/WS, relays, storage, indexers)  
 19.2 Transaction cost projections (per chain, per strategy)  
 19.3 Expected alpha vs cost (unit economics)  
 19.4 Optimization levers (caching, private relays, bundle batching)

# **20. Roadmap & Milestones**

20.1 MVP (single chain + one aggregator + one DEX + private relay)  
 20.2 Phase 2 (cross-chain routes, flash-loan arb, LP v3)  
 20.3 Phase 3 (liquidity mining automation, bribe integrations)  
 20.4 Phase 4 (advanced privacy, intents marketplace, perps basis trades)  
 20.5 Acceptance criteria per milestone

# **21. KPIs & Success Metrics**

21.1 Execution quality (price improvement vs VWAP/mid)  
 21.2 Fill reliability (success rate, inclusion delay)  
 21.3 Risk metrics (max drawdown, VaR breaches)  
 21.4 Ops metrics (MTTR, incident count)  
 21.5 Economic metrics (net P&L after fees/gas)

# **22. Open Questions & Decision Log**

22.1 Bridge trust assumptions (canonical vs intent-based)  
 22.2 Oracle dependencies (TWAP, Chainlink, Uniswap oracles)  
 22.3 Key custody model (MPC provider vs self-hosted HSM)  
 22.4 Solana vs EVM symmetry trade-offs  
 22.5 DEX/aggregator allowlist policy

# **23. Appendices**

A. Glossary (CPMM, CLAMM, SOR, RFQ, MEV, IL, ve(3,3), TWAP)  
 B. Math derivations (price-impact, tick math, Balancer weights)  
 C. Threat model matrix (STRIDE)  
 D. Data schemas (core tables & events)  
 E. API specs (internal gRPC/REST, auth flows)  
 F. Runbook templates (incident, deploy, rollback)  
 G. Compliance checklists (per jurisdiction)  
 H. Example configs (per chain, per strategy)