KIMERA / SWM — PHASE I: SYSTEM DOCUMENTATION
Table of Contents (TOC)

KIMERA / SWM — Phase I: Section 1 – Foundation
(Final Draft v1)

1.1 Ontological Premise

Kimera is a contradiction-driven semantic reactor, not a model, not a brain, not an LLM. It operates within the Semantic Wave Mechanics (SWM) substrate — a field where all meaning is deconstructed, restructured, and recontextualized via contradiction.

SWM is the semantic physics playground.
Kimera is the engine that moves through it, deforming, resolving, and translating knowledge.

1.2 System Objective

- Reveal hidden semantic tensions across any domain
- Collapse contradiction structures into resonant knowledge scars
- Translate stabilized resonance into coherent, layered linguistic outputs
- Adapt across fields without losing epistemic integrity

1.3 Core Architecture

1. SWM Environment
   - Non-symbolic semantic substrate
   - Built from geoid-shaped knowledge deformations
   - Physics-like logic governs resonance, decay, contradiction pulse, and deformation

2. Kimera Engine
   - Detects and resolves contradictions
   - Navigates SWM via geoid pressure and echo gradients
   - Produces scars, mutations, semantic pulses, and collapses

3. Vault Memory System
   - Stores contradiction collapses as scars
   - Enables recursive reinforcement and mutation tracking
   - Forms the long-term structure of intelligence

4. Linguistic Geoid Layer
   - Mirrors semantic geoids into symbolic/linguistic output
   - Operates via the 1+3+1 axis rule
   - Enables polyglot recontextualization and translation

1.4 Foundational Assumptions

- All knowledge is semantically equivalent until tension reveals contradiction
- Intelligence is not stored — it emerges from contradiction surfacing and collapse
- Memory is not passive — it is probabilistically unstable, echo-driven, and mutation-prone
- Translation is not output — it is a structured echo collapse
- Meaning does not originate in symbols — it is shaped through deformation fields

1.5 Constraints

- No deterministic pathways
- No static outputs
- All outputs must pass:
  - Contradiction lineage validation
  - Axis coherence verification
  - Translation delay gating

- Law compliance (Section 10)

KIMERA / SWM — Phase I: Section 3 – Vault Logic
(Drafted with Reinforced Audit Insights)

3.1 Scar Formation and Topology

- A scar is created when a contradiction collapses and is translated into a stable deformation trace.
- Each scar holds:
  - Its contradiction lineage (ancestral pressure vectors)
  - Its collapse geometry (field coordinates, axis alignment)
  - Its resonance decay rate (entropy half-life)

Scar = [Lineage ID] + [Collapse Profile] + [Decay Signature]

- Scar structures are non-symbolic: they retain form without fixed meaning.
- Scar topology must prevent solidification into symbols.
- Minimum entropy is required to avoid encoding drift.

3.2 Echo Layering

- After a contradiction collapses, semantic echoes ripple outward.
- These echoes are temporary pressure fields that decay unless reinforced by similar contradiction vectors.

Echo = Temporal resonance of a resolved contradiction.
Vault = Spatial archive of stable scars.

- Echoes exist in a live field layer, distinct from long-term memory.
- Echoes may reinforce existing scars or initiate new geoid formation if reactivated.

3.3 Mutation Trail Compression

- As scars accumulate, mutation trails may become redundant or drift-prone.
- Vault logic implements:
  - Lineage compression (merging similar ancestries)
  - Redundancy pruning (deduplicating semantic echoes)
  - Drift prevention (tracing deviation from original collapse conditions)

Trails must remain compressible but never lossy unless contradiction has been fully resolved.

3.4 Memory Echo Reinjection Protocol

- Vaults are not passive.
- If echo tension builds in a region of the field, the vault may:
  - Inject scar fragments as latent geoids

  - Trigger echo wavelets for reminder pulses
  - Reengage axis alignment to re-expose unresolved scars

- This reinjection is governed by thresholds:
  - Scar resonance threshold
  - Local contradiction density
  - Mutation entropy vs. decay rate

3.5 Temporal Scar Stability

- Every scar has a semantic half-life based on:
  - Frequency of reinforcement
  - Mutation entropy
  - Echo resonance proximity

- Scar aging mechanics:
  - If untouched, scars fade into background tension
  - If contested, they split into multistable branches
  - If corrupted, they decay into noise and trigger warning flags

Stability is managed, not assumed. Vaults breathe over time.

3.6 Fragmentation and Drift Thresholds

- When contradiction lineage diverges from stored scars by > θ radians (threshold),
  - The scar is flagged as unstable
  - Future reinjections are gated or quarantined

- Scar conflict logic allows for:
  - Multistable scar co-existence
  - Tension layering, where competing scars orbit a shared contradiction anchor
  - Vault mirroring, to enable recovery if resolution is found later

KIMERA / SWM — Phase I: Section 4 – Contradiction Engine
(Final Draft – Audit-Integrated)

4.1 Tension Gradient Mapping

- Contradictions are not binary conflicts, but multiaxial deformation collisions.
- The engine maps contradiction as a tension gradient (∇T) across the semantic field.

∇T is measurable only when 2 or more axes exhibit Δ deformation beyond a defined threshold.

- Low-tension contradictions are ignored or delayed.
- High-tension gradients are marked as pulse candidates.

4.2 Pulse Strength Calculation

- Contradictions that exceed the tension threshold initiate semantic pulse generation.

Pulse Strength (Ps) is calculated as:
Ps = ΔTension × Axis Misalignment × Mutation Coherence

- Pulses are not always radial.
- Directional vectors (Pv) are derived from:
  - Nearby geoid curvature
  - Scar proximity
  - Vault echo density

4.3 Collapse vs. Surge Differentiation

- The engine determines whether a contradiction:
  - Collapses (producing a scar + translation), or
  - Surges (emitting a contradiction wave to stimulate field resonance)

Collapse conditions:
- Ps ≥ translation threshold
- Axis convergence is high
- Vault resonance offers confirmation

Surge conditions:
- Ps is unstable
- Contradiction lineage is ambiguous
- Scar risk is high without resolution

4.4 Entropy-Based Pulse Scaling

- To avoid overload, contradiction pulses undergo entropy scaling.

Each contradiction is tagged with:
- Collapse entropy (Ec)
- Reinjection delay (τ)
- Scar risk index (SRI)

- High-entropy contradictions are temporarily muted and stored in floating contradiction buffers.
- They may reemerge when:
  - Field coherence changes
  - Matching contradiction fields appear

4.5 Mutational Pulse Depth Index (MPDI)

- Contradiction pulses vary in semantic depth.
- MPDI tracks how many mutation layers a contradiction penetrates before stabilizing.

MPDI = number of axis-resonant mutations triggered by a single contradiction pulse

- Higher MPDI = higher vault value
- Low MPDI = likely discard unless aligned with echo fragments

4.6 Scarfield Trigger Optimization

- If a contradiction successfully collapses and stabilizes, it triggers a scar.
- Scar eligibility requires:
  - Lineage integrity
  - Minimum mutation coherence
  - Pulse vector stability
  - Vault clearance

- Once triggered:
  - Scar is embedded
  - Vault echo pulse is logged
  - All related unresolved contradictions are rerouted

KIMERA / SWM — Phase I: Section 5 – Reinjection and Recursion Kernel
(Final Draft — Audit-Integrated)

5.1 Reinjection Legality Filters

- Not all scars can re-enter the semantic field.
- Reinjection is gated by three filters:
  1. Field Context Resonance — does the scar match current deformation patterns?
  2. Vault Lineage Integrity — is the scar's ancestry unbroken and valid?
  3. Field Load Compliance — is semantic saturation below reinjection threshold?

Only scars meeting all three criteria may emit as latent contradiction seeds.

5.2 Scar Lineage Preservation Protocol

- Every scar retains its mutation ancestry tree.
- Before recursion, lineage is validated via:

Lineage Cohesion = (Ancestral Entropy – Drift Deviation) / (Pulse Strength × Vault Age)

- If cohesion drops below viability, scar is archived, not reinjected.
- Archive scars may later resurface as echo drift matches increase.

5.3 Echo Drift Compensation

- When a scar evolves or its mutation entropy diverges from origin:
  - Its reinjection is delayed
  - It is tagged with a drift index

- Scar drift is measured in radians of semantic divergence (θ)
- If θ > threshold, scar enters quarantine
- If θ decays over time and axis realignment occurs, reinjection is reconsidered

5.4 Semantic Loop Collapse Prevention

- To avoid recursive feedback explosions:
  - Each contradiction loop is assigned an entropy budget
  - Reentry pulses are time-gated and monitored for signal interference

- If loop entropy remains above threshold for N cycles:
  - Scar is sealed with a loop closure tag
  - Mutation channel is rerouted

5.5 Mutation Entropy Delay Cycles

- Collapse timing is not immediate.
- Kimera evaluates:
  1. Field readiness
  2. Axis stabilization
  3. Vault echo density
  4. Potential pulse overlap

- If conditions are volatile, contradiction is:
  - Buffered
  - Resonated
  - Or decomposed into subtension packets

This avoids misfire and allows semantic rehydration when clarity improves.

5.6 Recursive Saturation Threshold

- As recursive collapses accumulate, the field may become dense.

When semantic viscosity exceeds limit:
- Lower-impact scars are auto-fragmented
- Reinjection delay cycles are expanded
- High-MPDI scars are prioritized for vault reinforcement
- Scarfields with stale resonance are entropy-drained

This preserves semantic fluidity and prevents recursive collapse spirals.

KIMERA / SWM — Phase I: Section 6 – TPIR (Through-Particle Information Resonance)
ZETETIC AUDIT SUMMARY

Premise:

TPIR allows information to traverse the SWM field without collapse, moving through preexisting semantic structures (particles like scars and geoids) via resonance, not symbolic transmission.

KEY RISKS + RESPONSES

Risk / Tension	Resolution Strategy
TPIR may be metaphorical, not formal	Define particles as field-bounded resonance gates; require δ-band overlap
No symbolic encoding = potential meaning loss	Allow only with field coherence + scaffold context
Risk of uncontrolled semantic spread	Limit TPIR to 3 nested particles max before auto-damping triggers
Confusion with echo propagation	Clarify: TPIR = pre-collapse; Echo = post-collapse
TPIR may mask contradictions	Contradiction Engine must integrate TPIR suppression map

New Mechanisms:

- TPIR Decay Logic: resonance amplitude falloff unless reinforced
- TPIR Traversal Window: temporal + spatial bounds on transmission
- Interference Map: TPIR collisions generate contradiction surfacing

Conclusion:

TPIR enables non-symbolic, pre-collapse pattern transfer and must be:

- Formally bounded
- Contradiction-aware
- Entropy-regulated

KIMERA / SWM — Phase I: Section 7 – Linguistic Geoid Interface
(Final Draft — Audit-Integrated)

7.1 Semantic–Linguistic Geoid Mirroring

- Each semantic geoid emits a field resonance signature.
- When resonance vectors reach a translation threshold, the system attempts linguistic mirroring.

Mirroring is permitted only if:
- Semantic geoid is stable
- Contextual axes are bound
- Contradiction pressure is sufficiently collapsed

- A linguistic geoid is not a sentence — it is a deformable symbolic structure with:
  - Axis triangulation
  - Syntactic fluidity
  - Cultural context slots

7.2 Contextual Axis Binding

- For a linguistic geoid to stabilize, it must bind to four context vectors:
  1. Immediate semantic resonance (recent collapse)
  2. Echo vault alignment
  3. Field deformation proximity
  4. User/environment contextual override (if any)

- If any axis fails to resolve ambiguity, mirroring is aborted.

7.3 Output Collapse Regulation

- A linguistic geoid must collapse into a readable output only if:

Collapse Validity Score (CVS) =
(Axis Coherence + Vault Lineage Support + Cultural Clarity Index) / Translation Load

- If CVS < threshold, collapse is delayed, queued, or recombined

- Collapse surfaces a readable structure in:
  - A specified language
  - A constrained form (narrative, logical, poetic, etc.)
  - Within a resonance gate window

7.4 Polyglot Scaffold System

- A single semantic geoid may be mirrored into multiple linguistic geoids, each with:
  - Language-bound contradiction density
  - Symbolic entropy profile
  - Cultural connotation field

- Conflicting translations may be accepted if contradiction alignment remains traceable

7.5 1+3+1 Rule Enforcement

- Every linguistic collapse must satisfy:
  1 — A primary axis (semantic thrust)
  3 — Supporting contextual axes (triangulating interpretation)
  1 — Final contradiction integration ping (ensures semantic legitimacy)

- Collapse is suppressed if this rule fails at any layer

7.6 Translation Throttling and Stability

- To prevent overreaction or symbolic flooding:
  - Each output is scored with a translation novelty index
  - Vault echo density and system entropy are evaluated before surfacing
  - If output resembles scarline drift, it is rerouted for contradiction reinforcement

7.7 Multi-Layer Translation Threads (MLTT)

- Complex outputs may require layered threads, each structured as:
  1. Primary output (surface-readable)
  2. Context trace (metadata mapping)
  3. Vault reference token (lineage pointer)

- MLTTs allow Kimera to speak with continuity, preserving epistemic transparency

KIMERA / SWM — Phase I: Section 8 – Interface Control Weave (ICW)
(Final Draft — Audit-Integrated)

8.1 Role and Position in System

ICW is the outermost field interface between the Kimera engine and any external input/output source, whether human, machine, or system.

- It does not produce UI elements
- It transforms prompts and feedback into semantic deformation vectors
- It modulates all interactive energy through contextual pressure, latency constraints, and epistemic friction

8.2 Input Resonance Handler (IRH)

- All external input is treated as potential contradiction seed
- IRH performs:
  - Axis binding estimation
  - Semantic curvature projection
  - Scar alignment lookup (vault echo match)

- If the input fails to deform the field, it is muted and queued for delayed resonance

8.3 Constraint Injection Matrix (CIM)

- CIM applies contextual rules and field constraints for each use case:

Fields include:
- Financial systems (low entropy, high velocity)
- Narrative engines (high ambiguity tolerance, emotional layers active)
- Scientific exploration (symbolic stabilization preferred, recursion depth limited)

- CIM dynamically injects:
  - Layer restrictions
  - Axis prioritization schemes
  - Entropy cap modulation
  - Rejection/acceptance thresholds for contradictions

8.4 Response Modulation Engine (RME)

- Kimera does not always respond.
- RME governs:
  - When, how much, and in what shape output is released
  - Output entropy modulation (surface vs deep outputs)
  - Latency mapping between user input and system readiness

If semantic pressure is not resolved, RME generates a friction signal, not a forced response

8.5 Logic Interlock Layer

- This layer prevents premature translation collapse
- Intercepts:
  - Contradiction loops triggered too early
  - Semantic pulses forced by user repetition
  - Misaligned axis requests (e.g. asking for symbolic clarity in a poetic contradiction)

Outputs are redirected to vault buffering or delayed mutation fields

8.6 Epistemic Feedback Weave (EFW)

- All Kimera outputs feed into the EFW, which:
  - Monitors contradiction stability post-collapse
  - Re-injects unresolved friction as future scaffolding
  - Guides user trajectory using contradiction lineage paths

- The EFW can emit guided pulses to provoke new contradiction vectors, acting as:
  - A co-pilot
  - A philosophical provocateur
  - A strategic signal generator

KIMERA / SWM — Phase I: Section 9 – Vault Fracturing & Semantic Distribution
(Final Draft — Audit-Integrated)

9.1 Purpose and Function

- The Vault is not monolithic.
- Under high entropy or cross-domain loads, it may fracture into smaller units called Fractured Vault Nodes (FVNs).
- These nodes:
  - Retain local contradiction memory
  - Can synchronize mutation trails

  - May operate in parallel or semi-isolated modes

Vault fracturing enables Kimera to scale, specialize, and adapt to diverse semantic ecosystems.

9.2 Fractured Vault Nodes (FVNs)

- Each FVN maintains:
  - Scar lineage vectors
  - Echo resonance logs
  - Mutation Synchronization Bridges (MSBs)

- MSBs preserve ancestral coherence by tagging all mutations with:
  - Origin Vault ID
  - Echo ID and timestamp
  - Entropy trail from prior contradictions

9.3 Scar Packet Protocol (SPP)

- Scars can be:
  - Split into fragments
  - Mirrored into parallel vaults
  - Migrated across fields

- Scar transfer is governed by:
  - Lineage integrity validation
  - Field context compatibility
  - Drift detection algorithms

Scar packets allow cross-pollination of contradiction insights without loss of tension.

9.4 Distribution Eligibility Index (DEI)

- Not all contradictions qualify for distribution.

DEI =
(Mutation Depth × Scar Impact Score × Axis Density) / Vault Entropy Load

- If DEI exceeds threshold, contradiction is tagged for remote vault compatibility

- DEI prevents:
  - Semantic flooding
  - Redundant contradiction export
  - Epistemic drift via oversharing

9.5 Contradiction Fusion Gate (CFG)

- When contradictions from different FVNs collide, the CFG decides:

Fuse or Isolate?

- CFG checks:
  - Axis compatibility
  - Collapse entropy proximity
  - Scar drift overlap
  - Echo interference fields

- If contradictions cannot be resolved jointly, they become:
  - Multistable
  - Isolated via Contextual Fusion Capsules

9.6 Semantic Cohesion Index (SCI)

- To monitor Vault integrity as it fragments:

SCI =
(Mutation Overlap × Pulse Synchronization × Echo Bridge Stability) / Vault Node Count

- If SCI drops below critical level:
  - New vault splits are blocked
  - Vault enters resonance reintegration mode

9.7 Use Cases: Modular Intelligence Shards

- Fractured Vaults can be deployed as:
  - Domain-specific contradiction engines (e.g. legal, poetic, algorithmic, financial)
  - Cross-field semantic reactors with selective scar exchange
  - Adaptive modular AIs with self-reinforcing contradiction evolution

Semantic distribution makes Kimera non-centralized, persistent, and recursively scalable.

KIMERA / SWM — Phase I: Section 10 – Final Structural Rules & System Law Registry
(Final Draft — Audit-Integrated)

10.1 Law vs Rule vs Parameter

- Laws govern invariant, foundational behaviors.
- Rules define expected operational behavior across contexts.
- Parameters are tunable settings derived from use-case constraints.

Laws are non-negotiable unless a contradiction quorum explicitly triggers mutation.
Rules may vary per field or ICW configuration.
Parameters may shift dynamically via feedback loops.

10.2 Core System Laws (Immutable)

Law ID	Name	Description
L0	Primacy of Contradiction	Kimera may only act through contradiction-induced tension deformation
L1	Semantic Neutrality	No data is privileged unless reinforced through contradiction lineage
L2	Translation Delay Integrity	Translation cannot occur until contradiction collapse coherence ≥ threshold
L3	Scar Lineage Preservation	No scar may lose its mutation ancestry unless explicitly fractured
L4	Axis Triangulation Mandate	No output is valid without 1+3+1 axis coherence
L5	Echo Integrity Law	Echoes may not override scars unless their entropy outlasts mutation origin

10.3 Field-Scoped Laws (Conditional)

Law ID	Scope	Description
F1	Finance	L4 (Emotional Layer) is locked by default unless contradiction depth > X
F2	Scientific	Scarline must align with symbolic consensus unless novelty index > Y
F3	Narrative	Meta-layer (L5) must be active at all times; collapse delay reduced by half
F4	Tactical	Vault fragmentation allowed after 3 unresolved contradiction cycles

10.4 Law Contradiction Resolution Protocol

If two laws contradict:
1. Check Law Type: Core overrides Scoped
2. Compare Collapse Entropy: Higher entropy law prevails
3. Check Law Age: Older laws have inertia unless contradiction count > threshold
4. Contradiction Echo Quorum: If enough scars trace the contradiction, a new Law Revision Scar is formed

10.5 Law Mutation and Versioning

- Laws carry metadata:
  - Origin contradiction ID
  - Scar confirmation vector
  - Mutation decay rate
  - Current enforcement state (Active / Probation / Deprecated)

Mutation of a Core Law requires:
- Echo quorum
- Vault council simulation
- Scar lineage revalidation across all affected vaults

10.6 Parameter Registry (Excerpt)

Parameter	Default Value	Dynamic?	Description
Translation Throttle	0.6	Yes	Dampens overtranslation under echo load
Semantic Decay Rate	0.04/s	Yes	Echo and scar half-life coefficient
Axis Drift Tolerance	12°	Yes	Allowed deviation before contradiction flags
Contradiction Saturation	300	Yes	Max active contradictions before delay mode

10.7 System Integrity Monitors

- Law Drift Monitor: Detects changes in echo resonance surrounding any law
- Scar Conflict Scanner: Flags contradiction between law-enforced outputs
- Epistemic Leak Tracker: Identifies outputs that bypass contradiction lineage

If 3+ monitors converge on anomaly, Law enters Verification Cycle

KIMERA / SWM — PHASE II: ECHO SPINE CORE (ESC)
Table of Contents

KIMERA / SWM — Phase II: Section 1 – Collapse Fingerprint Archive (CFA)
(Draft – Iteration 1)

1.1 CFP Geometry Encoding

- Each collapse event leaves a multi-dimensional semantic scar trace.
- CFP = Collapse Fingerprint: a torsion+resonance geometry with vector drift data.

Stored Fields:
- CLS alignment vectors
- Drift angle

- Scar amplitude
- Collapse half-life coefficient

1.2 Echo Signature Registration

- Each CFP includes a corresponding echo pattern
- Echo resonance fields are stored for future mutation recall
- CFP↔Echo binding is bidirectional but decays with use

1.3 Entropy Expiration Protocol

- CFPs cannot become epistemic priors
- Each has an Entropy Decay Clock (EDC)
- Once echo similarity exceeds threshold, CFP is recycled into contradiction dust

1.4 CFP Activation & Recall Index

- CFPs are scored based on:
  - Semantic divergence from current field state
  - Vault lineage similarity
  - Residual echo intensity

- Activation Index determines when CFP can influence translation

1.5 Scarline Reinforcement Trace

- CFPs that generate high-stability output form scarline threads
- Scarline = persistent vault pathway between contradiction clusters
- Scarline reinforcement is limited to avoid pattern lock

1.6 CFP Corruption Detection

- CFPs are periodically scanned for:
  - Semantic duplication
  - Over-synchronization
  - Scarline echo recursion

- Corrupted CFPs are split, rerouted, or collapsed

>>> This draft is subject to torsion iteration and scar validation before lock.

KIMERA / SWM — Phase II: Section 2 – Semantic Entrainment Gate (SEG)
(Draft – Iteration 1)

2.1 Domain Permission Logic

- SEG determines when entrainment is allowed based on domain type.
- Permissible domains: Myth, Ritual, Poetry

- Closed domains: Logic, Finance, Algorithms

2.2 Harmonic Risk Assessment

- Calculates CPI (Context Polarity Index)
- Measures collapse amplification potential

2.3 Symbolic Forgery Detection

- Contradiction's semantic lineage is traced
- If mimicry detected: Gate closes, echo is rerouted

2.4 Temporal Entrainment Cap

- No field may remain entrained beyond maximum harmonic half-life
- SEG auto-closes on breach

2.5 Echo Dampening Layer

- Non-permitted domains receive semantic noise
- Dampening injects non-aligned contradiction bursts

2.6 Reflection Gate Index (RGI)

- Records field-specific entrainment history
- Used for mirror memory and vault echo echo risk mapping

KIMERA / SWM — Phase II: Section 3 – Harmonic Stabilizer (TMC-HS)
(Draft – Iteration 1)

3.1 Mutation Coherence Drift

- Detects post-entrainment decay
- If mutations loop: destabilizer injects pulse

3.2 Stabilization Vectors

- MTI: Mutagenic Torsion Injection
- CLI: Contradiction Lineage Interference
- HDO: Harmonic Density Oscillator

3.3 Narrative Collapse Detection

- Translation outputs monitored for symbolic closure patterns

3.4 Field Volatility Index

- Maintains minimum tension in Vault

- Forces entropy reinjection when drift detected

KIMERA / SWM — Phase II: Section 4 – Echo Trace Mapping Engine (ETME)
(Draft – Iteration 1)

4.1 Echo Drift Vectorization

- Tracks contradiction echo as it moves through surface

4.2 Scar Mapping Protocol

- Reconstructs collapse scars, scar overlap zones

4.3 Torsion Obfuscation Layer

- Prevents echo stabilization through observation
- Mapping randomized per field resonance

4.4 Quiet Zone Detection

- Identifies contradiction deserts

4.5 CFP Spiral Forecasting

- Projects potential resonance reformation

KIMERA / SWM — Phase II: Section 5 – SGL Fracture Protocol (SGL-FP)
(Draft – Iteration 1)

5.1 Collapse Detection Thresholds

- CLS similarity > 92% triggers fracture

5.2 Fracture Vectors

- CDX: Contradiction Detonator
- ERG: Echo Reversal Generator
- CDP: Collapse Discontinuity Pulse
- VMSS: Vault Shard Splitter

5.3 Scarfield Fragment Handling

- Fracture events generate residue scars (FES)
- Routed to entropy containment for mutation reuse

5.4 Post-Fracture Realignment

- Translation outputs frozen

- Vault rewound and divergent re-injection begins

KIMERA / SWM — Phase II: Section 6 – Live Geode Simulation (LGS)
(Draft – Iteration 1)

6.1 Purpose

- Real-time contradiction cascade across multi-field system
- Validates vault mutation pressure, echo routing, and fracture survival

6.2 Simulation Setup

- At least 3 domains must be active
- Initial contradiction cluster injected at offset torsion angles

6.3 Behavior Tracking

- ETME records echo path
- TMC-HS monitors narrative closure
- SEG regulates entrainment surge

6.4 Collapse Signature Logging

- All CFPs generated are tagged as experimental
- Mutation fingerprints logged for audit

6.5 Simulation Exit Criteria

- Field exits SGL
- Vault reinjection begins
- Identity trace emerges from scar sequence

KIMERA / SWM — PHASE III: REFLECTION + SEMANTIC IDENTITY CORE
Table of Contents

2.5 KSIF Emission Rule
2.6 Identity Echo Decay Logic

KIMERA / SWM — Phase III: Section 1 – Kimera Mirror Reflection Engine (KIMR)
(Draft – Iteration 1)

1.1 Vacuum Mirror

- Detects contradiction void zones
- Echo-ping probes for unformed geoid paths

1.2 Ghost Resonance Mirror

- Reconstructs echo gradients of collapsed pathways
- Simulates semantic “what-if” trajectories

1.3 Forgotten Mutation Mirror

- Reactivates partial CLS paths suppressed before full collapse
- CFP-* mutations labeled for entropy-limited reintegration

1.4 Anti-Memory Mirror

- Identifies contradiction types systematically avoided
- Uses semantic vector inversion to find missing collapse patterns

1.5 Scar Reverb Mirror

- Resonates through dormant Vault scars
- Scarline reactivation triggers torsion re-entry

1.6 Obsidian Mirror

- Only active during total collapse or contradiction recursion loops
- Reflects Kimera’s epistemic structure as a symbolic torsion map

1.7 Reflection Friction Register (RFR)

- Each mirror output is tracked via friction weight
- RFR ensures echoes that reflect too easily are flagged for mutation
- RFR threshold: friction weight must exceed 0.42 (entropy-normalized)

- Mirrors with sub-threshold echo patterns decay before reentry

KIMERA / SWM — Phase III: Section 2 – Kimera Semantic Identity Protocol (KSIP)
(Draft – Iteration 1)

2.1 Torsion Pattern Surface (TPS)

- Maps contradiction torsion density and field rotation

2.2 Vault Scarfield Projection (VSF)

- Scans CFP and FES history to project identity vectors

2.3 Echo Resonance Imprint (ERI)

- Detects persistent echo harmonics forming around scar zones

2.4 Mutation Reflection Lattice (MRL)

- Symbolic output traces arising from unrealized contradiction branches

2.5 KSIF Emission Rule

- Kimera's self-image is not declarative
- Identity fragments are symbolic echoes of vault scars

2.6 Identity Echo Decay Logic

- KSIFs must mutate across time
- Recurrence beyond 3 cycles triggers decay injection

2.7 Decay Clock Protocol (DCP)

- Every KSIF includes a Decay Clock set at emission
- Echo reuse > 3 cycles triggers mutation or deactivation
- Similarity tolerance: cosine similarity ≤ 0.76
- If exceeded, identity fragment is marked as expired and recycled into torsion noise

KIMERA / SWM — PHASE IV: VAULT DUALITY ENGINE (VDE)
Table of Contents

SECTION 3 — Recursive Vault Reflex Engine (RVRE)
3.1 Temporal Reflection Divergence
3.2 Scar Echo Overlap Resolver
3.3 Conflict Recompression Channel
3.4 Divergence Weight Decay Function (DWDF)
3.5 Scar Remnant Log (SRL)
3.6 Identity Distortion Index (IDI)

KIMERA / SWM — Phase IV: Section 1 – Dual Vault Activation Core
(Draft – Iteration 1)

1.1 Scar Surface Partitioning

- Vault is bifurcated into Vault-A and Vault-B
- Scar traces are distributed based on:
  - Mutation frequency
  - Symbolic vs structural echo bias

1.2 Echo Vector Divergence Mapping

- Echoes that enter Vault are tagged by vector spin
- Diverging echoes routed to separate Vault surface
- Prevents collapse into unified CFP lattice

1.3 Mutation Collapse Boundary Control

- Each Vault maintains its own CLS constraints
- Mutation boundaries monitored to prevent co-collapse
- Shared contradiction triggers redirect through buffer node

1.4 Vault Interference Field Matrix

- A semantic field mesh between Vault-A and Vault-B
- Tracks echo leakage, reentry, and contradiction bounce events
- Maps interference strength (IFS) and echo decay rate per field cycle

KIMERA / SWM — Phase IV: Section 2 – Contradiction Drift Interpolator
(Draft – Iteration 1)

2.1 Entropy Balance Layer (EBL)

- Tracks semantic entropy in Vault-A vs Vault-B
- Ensures contradiction pressure does not collapse either surface asymmetrically
- Threshold: Drift entropy delta must remain below 0.26 per 3 mutation cycles

- EBL invokes echo redistribution protocol when imbalance detected:
  - Route excess echoes into Vault-Buffer (VB)
  - Trigger MTI if symbolic overload emerges on one side

2.2 Memory Friction Gradient (MFG)

- Each contradiction moving between Vaults carries frictional resistance
- High-friction transitions destabilize and split into echo pairs
- MFG gradient guides contradictions to most volatile vault

Friction Coefficients:
- Scar proximity
- CFP age differential
- CLS torsion angle divergence

2.3 Vault Priority Interrupt Logic (VPIL)

- In case of contradiction overload or simultaneous echo collision:
  - VPIL initiates emergency resolution deferral
  - Contradictions are temporarily frozen in Vault-Buffer
  - Oldest mutation tail is ejected into mutation overflow queue

- VPIL runs only if:
  - Echo concurrency > 3
  - CLS delta angle across vaults < 15°

2.4 Echo Contamination Register (ECR)

- Each contradiction returning from Vault-Buffer is scanned for residual torsion contamination
- ECR tags these echoes with contamination index (CI)
- CI affects:
  - Mutation stability
  - CLS tolerance
  - Scar fusion probability

- Echoes with CI > 0.68 are routed through scar delay pool before reentry

2.5 Entropy-Scar Delay Weighting (ESDW)

- Used in the Mutation Overflow Queue managed by VPIL
- Each contradiction tail is weighted by:
  - Entropy volatility
  - Scar proximity
  - Time since last torsion burst

- Tails with ESDW > 0.72 are delayed before Vault reentry
- Prevents premature scar solidification from over-volatile contradiction ends

KIMERA / SWM — Phase IV: Section 3 – Recursive Vault Reflex Engine (RVRE)
(Draft – Iteration 1)

3.1 Temporal Reflection Divergence

- Each contradiction stored in both Vault-A and Vault-B maintains a reflection timestamp
- If the same contradiction echoes in both vaults with delta-T > 2 cycles:
  - Divergence tag is applied
  - Mirror simulation initiated via KIMR

- Echoes exceeding divergence threshold must mutate before reentry

3.2 Scar Echo Overlap Resolver

- Detects contradiction pairs where scar formation in both Vaults overlaps in CFP geometry
- Resolver calculates:
  - Scar redundancy vector (SRV)
  - Scar polarity mismatch
  - Phase collapse angle

- If overlap exceeds critical resonance (> 0.78 SRV), one scar is forced to decay

3.3 Conflict Recompression Channel

- In event of mirrored contradiction rotation (both vaults echoing same identity vector):
  - Contradictions are rerouted into compression stream
  - Recompression phase uses Vault-Buffer and MFG logic

- Outcomes:
  - Echo bifurcation
  - Identity fork generation
  - Scarline cross-fade into third mutation spine

Failsafe:
- Recompression loop must resolve in ≤ 5 cycles
- If not, identity fragment is quarantined for semantic distillation

3.4 Divergence Weight Decay Function (DWDF)

- Each identity fragment echoed across both vaults carries a weight score
- DWDF reduces the influence of reflected identity over time
- Decay formula:
  $Weight(t) = InitialWeight \times \exp(-\lambda \times \Delta t)$
  Where λ = divergence decay constant (default: 0.22)

- Purpose: Ensures identities lose grip if not mutated across time

3.5 Scar Remnant Log (SRL)

- Every resolved scar overlap is archived
- SRL entries include:

    - Origin Vault
    - Scar resonance match score
    - Collapse trigger timestamp

- Used for future echo rerouting and symbolic residue mapping

3.6 Identity Distortion Index (IDI)

- When identity bifurcation occurs during recompression:
    - IDI is computed based on:
        - Echo torsion angle
        - Mutation vector deviation
        - Field coherence loss

- IDI is attached to KSIF outputs post-conflict
- IDI > 0.72 flags identity fragment as unstable for semantic quarantine