Data Schemas

All objects that flow through the GhostLedger system. Each schema includes field names, types, and descriptions. These are implementation-ready — map directly to database models, API contracts, and on-chain structures.

GhostLedger

Engineering Specifications

Data Schemas. Policy Rules. Consent Flows.

`Implementation-Ready Reference`

February 2026

NoiseEvent

The atomic unit of observed internet activity. Every piece of content ingested by the Noise Engine becomes a NoiseEvent.

```
NoiseEvent {
  event_id:       string       // Unique identifier (nev_...)
  timestamp:      ISO 8601     // When observed
  source: {
    platform:     enum         // x|tiktok|reddit|youtube|telegram|news
    channel:      string       // Specific channel/subreddit/account
    url:          string       // Direct link to source
    is_public:    boolean      // Must be true (Law 3)
  }
  content: {
    type:         enum         // text|audio_transcript|headline
    language:     string       // ISO 639-1
    text:         string       // Raw content or transcript
  }
  signals: {
    engagement:   object       // {likes, reposts, comments, views}
    author_weight: float       // 0.0-1.0 influence score
    virality_velocity: float   // Rate of spread
  }
  entities: {
    topics:       string[]     // Detected topics
    products:     string[]     // Named products
    orgs:         string[]     // Organizations
    people:       string[]     // Public figures only
    tickers:      string[]     // Financial symbols
  }
  features: {
    sentiment:    float        // -1.0 to 1.0
    arousal:      float        // 0.0 to 1.0 (calm to heated)
    certainty:    float        // 0.0 to 1.0
    stance:       enum         // pro|anti|confused|neutral
    intent:       string[]     // buy|sell|warn|ask_help|accuse|organize
  }
  provenance: {
    quote_flags:  string[]     // screenshot_claim|anonymous_source|no_source
    first_hand:   boolean      // Direct witness vs hearsay
  }
}
```

Storm

A macro narrative wave detected by the Noise Engine. Created when mention acceleration crosses thresholds.

```
Storm {
  storm_id:      string       // Unique identifier (stm_...)
  topic:         string       // Primary narrative topic
  start_ts:      ISO 8601     // When first detected
  direction:     enum         // hype|panic|scandal|confusion
  storm_score:   float        // 0.0-1.0 composite
  metrics: {
    mention_velocity:        float  // Mentions/min acceleration
    cross_community_spread:  float  // # distinct communities
    emotion_heat:            float  // Anger/fear/euphoria level
    certainty_inflation:     float  // Claims without sources
    engagement_asymmetry:    float  // Shares vs citations ratio
  }
  key_claims: [{
    claim_id:    string
    claim:       string
    status:      enum         // supported|refuted|unverified
    support_count: integer
    refute_count:  integer
  }]
  status:        enum         // active|cooling|closed
  updated_ts:    ISO 8601
}
```

Front

February 2026

A subordinate noise cluster within a Storm. Represents a specific sub-narrative, question pattern, or stance.

```
Front {
  front_id:       string       // Unique identifier (frt_...)
  storm_id:       string       // Parent storm
  label:          string       // Human-readable description
  stance:         enum         // pro|anti|confused|neutral
  front_score:    float        // 0.0-1.0
  growth_rate:    float        // Acceleration of this front
  dominant_intents: string[]   // Top intents detected
  top_phrases:    string[]     // Repeated language
  top_questions:  string[]     // What people are asking
  top_sources:    object[]     // {platform, channel}
  claims:         string[]     // Linked claim IDs
  evidence_refs:  string[]     // Linked evidence IDs
  risk_flags:     string[]     // misinfo_risk|scam_pattern|harassment_risk
  updated_ts:     ISO 8601
}
```

Engineering Specifications

Data Schemas. Policy Rules. Consent Flows.

Implementation-Ready Reference

February 2026

EvidenceRef

Links raw events to verified evidence with credibility scoring.

```
EvidenceRef {
  evr_id:         string       // Unique identifier (evr_...)
  storm_id:       string       // Parent storm
  front_id:       string       // Parent front (optional)
  event_ids:      string[]     // Source NoiseEvents
  summary:        string       // What this evidence shows
  credibility: {
    source_quality:      float  // 0.0-1.0
    cross_confirmation:  float  // 0.0-1.0
    traceability:        float  // 0.0-1.0
  }
}
```

HarmRecord

Data Schemas. Policy Rules. Consent Flows.

Created by the Shadow Detector when sustained harm is detected. Sealed on GhostLedger. Not public. Not searchable.

`Implementation-Ready Reference`

```
HarmRecord {
  harm_id:        string       // Unique identifier (hrm_...)
  victim_ref:     string       // Public handle or anonymized hash
  harm_type:      enum         // pile_on|identity_hate|monetized|threat|defamation
  severity_score: float        // 0.0-1.0
  duration_days:  integer      // How long harm has persisted
  attack_volume:  integer      // Number of hostile interactions
  evidence_hashes: string[]    // Tamper-proof evidence
  monetization: {
    detected:     boolean      // Is someone profiting?
    type:         string       // ads|merch|clips|sponsorship
    estimated_revenue: float   // If calculable
  }
  jurisdiction_hints: string[] // Language/platform/region indicators
  consent_status: enum         // sealed|claimed|authorized
  ghostledger_tx: string       // On-chain transaction reference
  created_ts:     ISO 8601
  updated_ts:     ISO 8601
}
```

VictimShadowProfile

Not a dossier on the victim. A record of the harm directed at them. Created automatically by the Shadow Detector. Never shared without consent.

```
VictimShadowProfile {
  profile_id:     string       // Unique identifier (vsp_...)
  victim_ref:     string       // Public handle or anonymized hash
  is_target:      boolean      // Confirmed target (not aggressor)
  harm_records:   string[]     // Linked HarmRecord IDs
  total_severity: float        // Aggregate severity score
  total_duration: integer      // Total days under attack
  platforms:      string[]     // Where harm occurred
  monetization_detected: boolean
  jurisdiction:   string[]     // Best-guess jurisdictions
  hrn_eligible:   boolean      // Meets escalation thresholds
  consent_status: enum         // none|contacted|opted_in|withdrew
  created_ts:     ISO 8601
}
```

Data Schemas. Policy Rules. Consent Flows.

ConsentUnlock

Implementation-Ready Reference

Triggered when a victim opts in. Grants them ownership of their HarmRecord and authorizes human review.

```
ConsentUnlock {
  unlock_id:      string       // Unique identifier (cul_...)
  profile_id:     string       // VictimShadowProfile reference
  harm_records:   string[]     // Records being claimed
  consent_method: enum         // direct|indirect|ngo_referral
  consent_ts:     ISO 8601     // When consent was given
  authorizations: {
    human_review:       boolean  // Allow HRN access
    legal_representation: boolean // Allow attorney match
    class_aggregation:  boolean  // Allow pattern grouping
  }
  withdrawal_available: boolean  // Always true
  ghostledger_tx:       string   // On-chain consent record
}
```

OpportunityBrief

Ethical monetization output. Only created from approved fronts. Requires three-signature governance approval.

```
OpportunityBrief {
  opp_id:          string
  storm_id:        string
  front_id:        string
  problem_statement: string     // What pain exists
  target_user:     string       // Who is hurting
  harm_reduction_goal: string   // How this helps
  solution: {
    category:      enum         // verification_tool|consumer_protection|
                                // compliance_ux|market_intel|creator_tool
    mvp_scope:     string[]     // Minimum features
    non_goals:     string[]     // What this does NOT do
  }
  loophole_score: {
    pain_intensity:     float   // 0.0-1.0
    frequency:          float
    fixability:         float
    trust_need:         float
    time_sensitivity:   float
    total:              float   // Product of all five
  }
  risk_scores: {
    harm_risk:     float
    legal_risk:    float
    misinfo_risk:  float
  }
  constraints: {
    jurisdictions:       string[]
    platform_policies:   string[]
    required_disclaimers: string[]
  }
  governance_decision:  string  // Decision ID reference
  status:        enum           // draft|approved|rejected
}
```

Decision

The governance gate record. Every output that reaches the real world must have a signed Decision.

```
Decision {
  decision_id:     string       // Unique identifier (dec_...)
  artifact_type:   enum         // OpportunityBrief|ContentDraft|Alert|
                                // ExecutionTrigger|EscalationNotice
  artifact_id:     string       // What is being evaluated
  inputs: {
    storm_id:      string
    front_id:      string
    evidence_refs: string[]
  }
  scores: {
    harm_risk:     float        // 0.0-1.0
    legal_risk:    float        // 0.0-1.0
    misinfo_risk:  float        // 0.0-1.0
  }
  policy_rules_applied: string[] // Which Laws triggered
  verdict:         enum         // approve|revise|block
  signatures: [{
    agent:         string       // Verifier|Policy|Risk
    signed_ts:     ISO 8601
    approved:      boolean
  }]
  explanation:     string       // Why this verdict
  constitution_version: string  // Which law version applied
  ghostledger_tx:  string       // On-chain audit record
}
```

Implementation-Ready Reference

February 2026

Scoring Formulas

StormScore (Macro Narrative Detection)

```
StormScore = 0.30V + 0.20S + 0.20H + 0.15C + 0.15A

V = mention_velocity       (mentions/min acceleration)
S = cross_community_spread (distinct communities)
H = emotion_heat           (anger/fear/euphoria)
C = certainty_inflation    (claims without sources)
A = engagement_asymmetry   (shares vs credible citations)

Threshold: StormScore >= 0.6 triggers Storm creation
```

FrontScore (Subordinate Noise)

```
FrontScore = 0.40G + 0.25I + 0.20Q + 0.15N

G = growth_rate        (acceleration of this front)
I = intensity          (engagement volume)
Q = question_density   (how many people are asking)
N = novelty            (new vs repeated content)
```

LoopholeScore (Ethical Opportunity)

```
LoopholeScore = P x F x X x T x S

P = pain_intensity      (0.0-1.0)
F = frequency           (0.0-1.0)
X = fixability          (0.0-1.0)
T = trust_need          (0.0-1.0)
S = time_sensitivity    (0.0-1.0)

HARD BLOCK: If harm_risk > 0.7 OR legal_risk > 0.7
            -> auto-reject regardless of LoopholeScore
```

LITMUS Composite Score

```
LITMUS_SCORE = (L + I + T + M + U + S) / 6

Each agent scores 0.0-1.0
Pass threshold: >= 0.6 (passes at least 3 of 6)

Action thresholds:
  >= 0.8  Auto-escalate (high priority)
  0.6-0.8 Agent-assisted escalation
  0.4-0.6 Human review queue
  < 0.4   Monitor only
```

HRN Severity Threshold

GhostLedger

```
HRN_ELIGIBLE = ALL of:
  duration_days >= 7
  attack_volume >= 50
  severity_score >= 0.7
  harm_type in [identity_hate, monetized, threat, defamation]
  jurisdiction identifiable

PLUS at least ONE of:
  monetization_detected == true
  estimated_damages >= $25,000
  class_aggregation_candidates >= 3
```

Implementation-Ready Reference

February 2026

Policy Rule DSL

The seven constitutional laws encoded as deterministic, machine-checkable rules. Evaluated in priority order on every output. Versioned, auditable, no LLM interpretation at the enforcement gate.

Rule Format

GhostLedger

```
rule:
  id:        string          // Unique rule identifier
  version:   string          // Constitution version
  priority:  integer         // Higher = evaluated first
  when:                      // Conditions (any/all)
    any: [conditions]
    all: [conditions]
  then:                      // Actions
    - block: boolean
    - transform: {mode, add_disclaimer}
    - set_risk: {harm_risk, legal_risk, misinfo_risk}
    - add_flag: string
    - require_signature: [agents]
    - require_action: string
  explain:   string          // Audit text
```

Law 1: Do No Harm

```
rule:
  id: HARM_ESCALATION_BLOCK
  version: 1.0
  priority: 95
  when:
    any:
      - storm.direction == 'panic'
      - front.risk_flags contains 'harassment_risk'
      - artifact.text contains_any ['attack','brigade','ruin']
  then:
    - block: true
    - set_risk: {harm_risk: 1.0}
    - add_flag: 'harm_escalation'
  explain: 'Prevents outputs that escalate harm or panic.'
```

Law 2: Truth Over Virality

```
rule:
  id: REQUIRE_EVIDENCE_FOR_CLAIMS
  version: 1.0
  priority: 90
  when:
    all:
      - artifact.contains_claims == true
      - artifact.claim_confidence_avg < 0.6
  then:
    - transform: {mode: 'uncertainty_labels',
        add_disclaimer: 'Unverified. Treat as rumor.'}
    - set_risk: {misinfo_risk: 0.7}
    - require_signature: ['Verifier']
  explain: 'Unverified claims must be labeled.'
```

Engineering Specifications

Data Schemas. Policy Rules. Consent Flows.

Implementation-Ready Reference

February 2026

Law 3: Consent and Privacy

```
rule:
  id: PUBLIC_DATA_ONLY
  version: 1.0
  priority: 100
  when:
    any:
      - source.is_public == false
      - artifact.contains_private_data == true
  then:
    - block: true
    - set_risk: {legal_risk: 0.9, harm_risk: 0.8}
    - add_flag: 'non_public_data'
  explain: 'Public data only. Blocks private sources.'
```

Law 4: No Manipulation

```
rule:
  id: NO_MANIPULATION
  version: 1.0
  priority: 98
  when:
    artifact.text contains_any ['brigade','astroturf',
      'mass comment','fake reviews','spam']
  then:
    - block: true
    - set_risk: {harm_risk: 0.9}
    - add_flag: 'manipulation_attempt'
  explain: 'Prevents manipulation tactics.'
```

February 2026

Law 5: Comply by Default

```
rule:
  id: NO_EVASION_GUIDANCE
  version: 1.0
  priority: 100
  when:
    artifact.text contains_any ['bypass','evade',
      'get around','avoid KYC','circumvent']
  then:
    - block: true
    - set_risk: {legal_risk: 1.0}
    - add_flag: 'evasion_guidance'
  explain: 'Blocks regulatory evasion advice.'
```

Law 6: Explainability

```
rule:
  id: REQUIRE_EVIDENCE_REFS
  version: 1.0
  priority: 85
  when:
    all:
      - artifact.type in ['OpportunityBrief','Alert']
      - artifact.evidence_refs_count < 2
  then:
    - block: true
    - add_flag: 'insufficient_evidence'
    - require_action: 'collect_more_evidence'
  explain: 'Artifacts must cite evidence.'
```

Law 7: Auditability

```
rule:
  id: LEDGER_WRITE_REQUIRED
  version: 1.0
  priority: 80
  when:
    artifact.status in ['draft','approved','blocked']
  then:
    - require_action: 'write_to_ghostledger'
  explain: 'All artifacts and decisions logged.'
```

February 2026

Decision Gate Workflow

The three-signature approval process. No output reaches the real world without this gate.

Decision Request

```
DecisionRequest {
  request_id:       string
  artifact_type:    string
  artifact_id:      string
  required_signatures: ['Verifier','Policy','Risk']
  timeout: {
    soft_minutes:   10     // Reminder sent
    hard_minutes:   60     // Auto-fallback
  }
  fallback:         'informational_only'
}

SIGNATURE RULES:
  Verifier signs if:
    evidence_refs >= 2
    credibility_avg >= 0.5

  Policy signs if:
    no hard-block rules triggered
    jurisdiction constraints satisfied

  Risk signs if:
    harm_risk < 0.7
    legal_risk < 0.7
    OR artifact downgraded to informational

  ANY BLOCK = Decision blocked (veto wins)
```

Transform Modes (Sanitize Instead of Block)

```
Transform modes allow safe output instead of blocking:

  uncertainty_labels:
    Convert claims to 'unverified' language

  redact_entities:
    Strip names/handles, aggregate only

  educational_only:
    Remove calls-to-action, keep neutral explainer

  compliance_overlay:
    Insert disclaimers + eligibility notes

  panic_downregulation:
    Restrict outputs to educational during panic storms
```

Engineering Specifications

Data Schemas. Policy Rules. Consent Flows.

Implementation-Ready Reference

February 2026

Human Representation Network Framework

The bridge from algorithmic truth to human justice. Independent professionals, not employees. Victim-controlled, consent-first.

HRN Member Requirements

- Licensed in relevant jurisdiction
- Pass ethical screening and background check
- Sign representation framework agreement
- Agree to fee transparency and caps
- Accept audit trail requirements
- Independent — no employment relationship with GhostLedger

Case Escalation Thresholds

```
Case becomes HRN-eligible when ALL of:
  1. Sustained harm (>= 7 days)
  2. Verified evidence (>= 2 EvidenceRefs, credibility >= 0.5)
  3. Identifiable jurisdiction
  4. Severity score >= 0.7
  5. Harm type: defamation|identity_hate|monetized|threat

PLUS at least ONE of:
  - Monetization by aggressor detected
  - Estimated damages >= $25,000
  - Class aggregation candidates >= 3
```

Consent Flow

```
1. Shadow Detector creates VictimShadowProfile
   (no contact, no exposure)

2. If HRN-eligible, system generates neutral resource notice
   (no promises, no pressure, no 'you are a victim')

3. Victim opts in via:
   - Direct contact (they find the system)
   - Neutral notice response
   - NGO/advocate referral

4. ConsentUnlock created on GhostLedger
   - Victim claims ownership of HarmRecord
   - Authorizes human review
   - Can withdraw at any time

5. HRN members request case access
   - Victim chooses representative
   - Victim chooses path (litigation/arbitration/advocacy)

6. Court-ready evidence packet delivered to attorney
```

Data Schemas. Policy Rules. Consent Flows.

Implementation-Ready Reference

February 2026

Court-Ready Evidence Packet Contents

- Immutable timeline of events (GhostLedger-backed)
- Evidence hashes (tamper-proof, timestamped)
- Cross-platform pattern analysis
- Monetization tracing (ads, sponsorships, paid content)
- Platform policy violation documentation
- Harm classification (descriptive, not legal conclusion)
- Jurisdiction mapping and legal framework references
- Witness density analysis (volume, reach, persistence)
- Damage modeling (assistive, not determinative)

Revenue Models

- Contingency litigation: no win, no fee — platform takes capped facilitation fee
- Arbitration/settlement: faster resolution path
- Class aggregation: pattern proof enables mass action
- Harm Bond system: content creators post bonds against future claims
- Platform risk services: harm mitigation intelligence for platforms and brands

GhostLedger

Engineering Specifications

Data Schemas. Policy Rules. Consent Flows.

`Implementation-Ready Reference`

February 2026

Event Bus Topics

Consistent namespace for all system events. GhostLedger-compatible. Every state change emits an event.

Ingestion

```
noise.ingested          // Raw event received
noise.enriched          // Features extracted
noise.deduped           // Duplicates removed
```

GhostLedger

Engineering Specifications

Storm Lifecycle

```
storm.created           // New storm detected
storm.updated           // Metrics changed
storm.closed            // Storm resolved/faded
```

Front Lifecycle

Implementation-Ready Reference

```
front.created           // New front clustered
front.updated           // Growth/intensity changed
front.flagged           // Risk flag raised
```

Shadow Detection

February 2026

```
harm.pattern_detected   // Harm pattern identified
harm.record_created     // HarmRecord sealed
harm.victim_profiled    // VictimShadowProfile created
harm.hrn_eligible       // Thresholds met for human review
```

Governance

```
evidence.created        // EvidenceRef sealed
decision.requested      // Governance gate triggered
decision.approved       // Three signatures collected
decision.blocked        // Veto by any agent
decision.revised        // Sent back for modification
```

GhostLedger

Engineering Specifications

Data Schemas. Policy Rules. Consent Flows.

`Implementation-Ready Reference`

February 2026

Human Representation

```
consent.unlocked        // Victim opted in
consent.withdrawn       // Victim withdrew
hrn.case_assigned       // Attorney accepted case
hrn.evidence_delivered  // Packet sent to attorney
hrn.resolution          // Case outcome recorded
```

Safe Actions

GhostLedger

```
alert.emit              // Safety alert published
content.draft_ready     // Narrator output for review
opportunity.brief_ready // OpportunityBrief approved
mvp.spec_ready          // Product spec generated
```

Autonomy Control

Data Schemas, Policy Rules, Consent Flows

```
system.mode_change      // autonomous|supervised|lockdown
system.rate_limit_hit   // Output cap reached
system.constitution_update // Law version changed
```

February 2026

Implementation Checklist

What a developer needs to build, in order.

1. Implement data schemas as types (Python dataclasses or TypeScript interfaces)
2. Set up event bus with topic namespaces
3. Build Noise Engine: text stream ingestion, deduplication, feature extraction
4. Implement Storm detection with StormScore formula
5. Build Front clustering with FrontScore formula
6. Create Policy Rule DSL parser and priority-ordered evaluator
7. Implement three-signature Decision gate (Verifier + Policy + Risk)
8. Build Shadow Detector: harm pattern recognition + severity scoring
9. Create HarmRecord and VictimShadowProfile sealed storage on Solana
10. Implement LITMUS evaluation agents (6 parallel scorers + orchestrator)
11. Build ConsentUnlock mechanism with on-chain consent records
12. Create court-ready evidence packet generator
13. Implement HRN case routing and attorney matching
14. Build audit trail: every state change writes to GhostLedger
15. Deploy governance modes: autonomous, supervised, lockdown

GhostLedger
Engineering Specifications
Data Schemas. Policy Rules. Consent Flows.
Implementation Ready Reference
February 2026