

URL Ledger

Data Model & Schema Specification

The canonical object model for turning every URL into a measurable, governable, auditable asset across discovery, attribution, revenue, policy, and agent consumption.

URL Asset Core entity atomic unit of record	7 Schema layers identity to agent ingress	Truth Layer Primary use for humans, systems, and agents	Channel-neutral Design posture search, paid, social, CRM, AI, agents
---	---	---	--

Platform thesis

URL Ledger is not another dashboard. It is the canonical record of what each URL is, what it does, what it is worth, what risk it carries, what actions have touched it, and whether an external system or AI agent can trust it.

This specification is a working data model. It is intended for product planning, MVP scoping, engineering handoff, data warehouse design, API design, and partner conversations. Field names are representative and should be adapted to the final application stack.

CONTENTS

Section	Purpose
1. Schema principles	Defines the design rules that keep URL Ledger from collapsing into a generic analytics tool.
2. Logical architecture	Shows how identity, signals, scoring, policy, evidence, and agent ingress fit together.
3. Core entities	Defines the object model and relationships.
4. Minimum viable schema	Specifies the MVP database tables needed to run the first audits and pilots.
5. Ratings and 13-variable mapping	Connects the structural audit model to score records and time-series evidence.
6. Events, actions, and reconciliation	Models how recommendations become governed state transitions.
7. Agent and API layer	Defines how agents read trusted state without becoming the platform.
8. Roadmap and implementation notes	Shows what to build first, what can wait, and what compounds over time.

1. Schema principles

The schema must preserve the core market position: URL Ledger is the system of record for website asset value. It should not be built as a temporary SEO audit database, a content calendar, or a workflow wrapper around an AI model.

Principle	Meaning	Schema implication
URL as unit of account	Every measurable page is treated as an asset with identity, history, risk, and value.	url_assets becomes the primary table. Every signal, score, action, evidence record, and policy references a URL asset or cluster.
Canonical identity first	Before scoring performance, the platform must know what the asset is.	Separate canonical_url from observed_url. Preserve redirects, variants, duplicates, and lineage events.
Time-series truth	A URL value is not a point-in-time metric; it is a record of state changes.	Store observations, scores, snapshots, and actions as dated records rather than overwriting history.
Channel-neutral value	A URL can create value through organic, paid, social, email, referral, CRM, AI answers, direct, or agent usage.	Use channel_observations and attribution_events instead of hard-coding the platform around Google-only metrics.
Explainable ratings	Every score needs a because trail.	Ratings link to structural variable scores, source observations, confidence, and evidence artifacts.
Governed state transitions	The moat is not the agent. The moat is the ledger that decides what an agent or human may do.	Actions, approvals, policies, evidence, and change logs are first-class objects.
Agent-readable, not agent-owned	Agents consume trusted state. They do not become the system of record.	Expose read/query APIs, policy evaluation endpoints, evidence packs, and action requests. Keep writeback controlled.

The schema must answer five questions

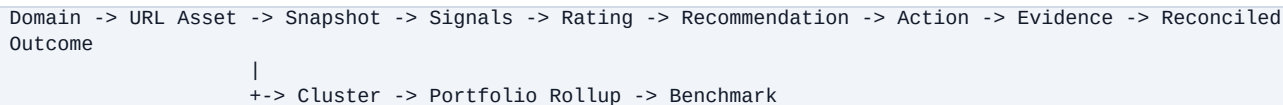
What is this URL? What does it contribute? What is decaying? What action is allowed? What evidence proves the conclusion?

2. Logical architecture

The platform should be modeled as seven schema layers. Each layer makes the next layer more defensible. Do not start with recommendations before the identity, signal, and evidence layers are stable.

Layer	Core objects	What it enables
1. Identity ledger	domains, url_assets, canonical_mappings, url_lineage_events	Canonical inventory, dedupe, redirect history, asset continuity.
2. Content state	content_snapshots, page_components, schema_snapshots, cms_records	What changed, when it changed, and whether the URL still represents the intended asset.
3. Discovery signals	channel_observations, query_observations, crawl_observations, ai_citation_observations	Visibility across search, paid, social, referral, AI answers, and machine consumption.
4. Attribution and value	attribution_events, revenue_mappings, conversion_definitions, assisted_value_records	Yield, assisted revenue, pipeline influence, and value-at-risk.
5. Ratings and risk	url_ratings, structural_variable_scores, impairment_events, confidence_records	Health, decay, risk, waste, recoverability, and audit confidence.
6. Actions and governance	recommendations, action_requests, policy_rules, approvals, change_logs	Refresh, merge, retire, protect, expand, or block actions with audit trail.
7. Evidence and access	evidence_artifacts, evidence_packs, api_clients, agent_access_logs	Proof, exportability, API access, agent-readable truth, and compliance posture.

LOGICAL OBJECT FLOW



3. Core entity relationship model

The first engineering goal is not a perfect ontology. The first goal is to create an object model that can survive messy reality: duplicates, redirects, deleted pages, syndicated pages, campaign URLs, template drift, multi-region variants, and partial analytics coverage.

Entity	Primary key	Key relationships	Purpose
Account / Organization	org_id	owns domains, users, policies, benchmarks	Commercial and permission boundary.
Domain / Property	domain_id	belongs to org; contains url_assets	Website, subdomain, region, or property under audit.
URL Asset	url_asset_id	belongs to domain; links to snapshots, signals, ratings, actions	Atomic asset record and unit of account.
Canonical Mapping	canonical_map_id	maps observed_url to canonical_url and url_asset_id	Dedupes variants and preserves identity.
URL Lineage Event	lineage_event_id	references source_url_asset_id and target_url_asset_id	Tracks redirects, merges, splits, migrations, deletes, and resurrection events.
Cluster	cluster_id	groups URL assets by intent, template, funnel role, product, topic, or business unit	Enables portfolio views and action prioritization.
Snapshot	snapshot_id	references url_asset_id	Captures content, template, metadata, schema, and technical state at a point in time.
Observation	observation_id	references url_asset_id, channel, date	Stores performance, visibility, crawl, or citation measurements.
Rating	rating_id	references url_asset_id or cluster_id, valid_from, valid_to	Standardized score, grade, risk, confidence, and explanation.
Recommendation / Action	action_id	references url_asset_id or cluster_id, policy, owner, status	Governed action queue and proof sprint workflow.
Evidence Artifact	artifact_id	references ratings, observations, actions, reports	Screenshots, exports, crawls, tables, calculations, approvals, and source data.
Agent Access Log	agent_access_id	references api_client_id, url_asset_id, policy_result	Records what external systems or agents read, requested, or attempted to modify.

4. Minimum viable schema

This is the smallest durable schema that can power the 45-day audit, generate a URL asset ledger, support ratings, create a recovery backlog, and prepare for controlled agent access later.

4.1 url_assets

Field	Type	Required	Purpose
url_asset_id	uuid	Yes	Stable internal identifier. Never reuse.
org_id	uuid	Yes	Commercial account boundary.
domain_id	uuid	Yes	Website or property boundary.
canonical_url	text	Yes	Canonical URL used as the asset identity.
normalized_url_hash	text	Yes	Deduplication key for canonical matching.
url_status	enum	Yes	active, redirected, retired, noindex, orphaned, unknown.
asset_type	enum	Yes	blog, guide, product, category, landing, comparison, template, support, legal, misc.
funnel_role	enum	No	TOFU, MOFU, BOFU, post-sale, support, brand, local, transactional.
business_owner	text/uuid	No	Team or person accountable for the asset.
created_at	timestamp	Yes	First observed or imported.
last_seen_at	timestamp	Yes	Most recent crawl, sitemap, or analytics confirmation.
confidence_score	numeric	Yes	Confidence that this URL record is correct and complete.
protected_flag	boolean	Yes	Whether the asset requires higher approval or no-touch policy.

4.2 canonical_mappings

Field	Type	Purpose
canonical_map_id	uuid	Mapping record ID.
observed_url	text	The raw URL observed in crawls, sitemap, GSC, GA4, campaigns, or logs.
canonical_url	text	The normalized canonical URL.
url_asset_id	uuid	The asset this URL variant belongs to.
source_system	enum	crawl, sitemap, canonical_tag, redirect, gsc, ga4, cms, manual.
mapping_reason	text	Why this variant maps to the canonical asset.
is_primary	boolean	Whether the mapping is the preferred canonical representation.
valid_from / valid_to	timestamp	Historical validity window.

5. Snapshot and signal tables

5.1 url_snapshots

Snapshots preserve what the page looked like and how it was technically represented at each observation point. This is essential for proving why a rating changed.

Field	Type	Purpose
snapshot_id	uuid	Snapshot record.
url_asset_id	uuid	URL being captured.
snapshot_date	date/timestamp	When state was observed.
title	text	Page title at capture time.
h1	text	Primary heading.
meta_description	text	Meta description.
word_count	integer	Approximate body text volume.
content_hash	text	Hash used to detect content changes.
template_id	text	Template or page type identifier if known.
indexability_status	enum	indexable, noindex, canonicalized, blocked, error, unknown.
http_status	integer	HTTP status at capture time.
canonical_tag_url	text	Canonical target declared on page.
structured_data_types	array/text	Schema detected on page.
crawl_depth	integer	Clicks from root/home or seed URL.
internal_inlinks	integer	Count of internal links pointing to the URL.
external_inlinks	integer	Count if backlink data is connected.
load_time_ms	integer	Performance proxy where available.
render_status	enum	rendered, partial, failed, js_required, unknown.

5.2 channel_observations

Field	Type	Purpose
observation_id	uuid	Observation record.
url_asset_id	uuid	URL asset observed.
channel	enum	organic_search, paid_search, social, email, referral, direct, internal_search, ai_answer, agent, crm, other.
surface	text	Google Search, Bing, LinkedIn, newsletter, AI Overview, ChatGPT, Perplexity, sales deck, etc.
observation_date	date	Daily, weekly, or monthly grain.
impressions	integer	Exposure count where available.
clicks	integer	Click count where available.
sessions	integer	Web sessions.
engaged_sessions	integer	Engagement proxy.
conversions	numeric	Primary conversion count.
revenue	numeric	Attributed or estimated revenue.
assisted_value	numeric	Multi-touch or proxy value.
position_or_rank	numeric	Rank, ad position, citation position, or list placement.
source_confidence	numeric	Confidence in source accuracy and join quality.

6. Attribution and value model

The platform should not overpromise perfect attribution. The correct posture is management-grade evidence: enough truth to make better capital allocation decisions than the current fragmented dashboard stack.

Table	Key fields	Business question answered
conversion_definitions	conversion_id, name, event_name, channel_scope, value_type, default_value, active_flag	What counts as value?
revenue_mappings	mapping_id, source_system, event_id, url_asset_id, attribution_model, revenue, confidence	How does a landing page or assisted touch connect to money?
attribution_events	event_id, user/session/account hash, url_asset_id, channel, timestamp, event_type, value	Which URL participated in a user, account, or opportunity journey?
assisted_value_records	record_id, url_asset_id, period, assisted_pipeline, assisted_revenue, model_used, confidence	Which assets support revenue even when they are not last click?
value_at_risk_records	risk_id, url_asset_id/cluster_id, period, baseline_value, observed_value, value_at_risk, recoverable_value	What revenue or pipeline appears exposed to decay, channel shifts, or structural failure?
cost_records	cost_id, url_asset_id/cluster_id, content_cost, dev_cost, media_cost, refresh_cost, period	What does it cost to create, maintain, refresh, or recover the asset?

Financial framing

Every URL should eventually carry three management values: yield, durability, and risk. Yield measures contribution. Durability measures how long the asset holds value. Risk measures how much value can leak if the asset decays or becomes structurally invisible.

7. Ratings and the 13 structural variables

Ratings should be stored as records, not calculated once and forgotten. The system should preserve what was scored, why it was scored, which evidence supported the score, and whether later outcomes validated the recommendation.

7.1 url_ratings

Field	Type	Purpose
rating_id	uuid	Rating record.
rated_object_type	enum	url_asset, cluster, domain, portfolio.
rated_object_id	uuid	ID of rated object.
rating_date	date	When rating was calculated.
overall_grade	enum	A, B, C, D, F or custom rating scale.
health_score	numeric	Current usefulness and technical/content strength.
decay_score	numeric	Observed or predicted deterioration.
risk_score	numeric	Business exposure, volatility, policy sensitivity.
recoverability_score	numeric	Likelihood that a fix creates lift.
confidence_score	numeric	Data quality and evidence confidence.
recommended_action	enum	protect, refresh, merge, retire, redirect, expand, monitor, investigate.
rating_summary	text	Human-readable because trail.
valid_from / valid_to	timestamp	Rating validity window.

7.2 structural_variable_scores

Structural variable	Example score fields	Evidence sources
1. Content health	accuracy_score, completeness_score, freshness_score	content snapshot, last updated, claim review, engagement metrics
2. Content decay	traffic_decay, ranking_decay, conversion_decay, freshness_decay	GSC, GA4, CRM, snapshots, historical baselines
3. Content dilution	low_value_ratio, duplicate_theme_count, weak_url_share	crawl, sitemap, cluster analysis
4. Cannibalization	intent_overlap_score, query_split_score, competing_url_count	query/page joins, SERP data, embeddings, cluster map
5. Content waste	maintenance_drag_score, no_value_signal, crawl_waste_score	crawl, analytics, cost records, conversions
6. Content investment	production_cost, refresh_cost, authority_embedded, proof_assets	CMS, work orders, media, internal metadata
7. Content ROI	revenue_per_url, assisted_value, conversion_efficiency	GA4, CRM, payments, attribution model
8. Content performance	impressions, clicks, CTR, sessions, conversions	GSC, GA4, paid/social/email data
9. Intent/journey alignment	funnel_match, query_intent_match, CTA_fit	query map, page type, journey tagging
10. Psychographic/messaging fit	trust_fit, reassurance_gap, decision_state_match	copy review, persona mapping, behavior signals
11. Authority/entity strength	brand_entity_score, topical_authority, proof_depth	structured data, backlinks, author data, references
12. Technical/indexation structure	indexability, crawl_depth, schema_validity, render_status	crawl, GSC index coverage, CWV, render tests
13. Agent/citation readiness	extractability, citation_presence, zero_click_risk, agent_policy_status	AI answer monitoring, schema, DOM, API logs, policy rules

8. Recommendations, actions, and reconciliation

The system becomes defensible when it can track the full transaction path: observation, score, recommendation, approval, implementation, outcome, and updated rating.

Table	Key fields	Purpose
recommendations	recommendation_id, object_type, object_id, action_type, priority, expected_lift, effort, confidence, rationale	The recommended move before approval.
action_requests	action_request_id, recommendation_id, requested_by, assigned_owner, status, due_date, policy_result	Operational task record.
policy_evaluations	evaluation_id, action_request_id, policy_id, result, reason, required_approvals	Determines whether the action can proceed.
approvals	approval_id, action_request_id, approver_id, decision, timestamp, notes	Records human governance.
change_logs	change_id, action_request_id, url_asset_id, before_state, after_state, implemented_by, implemented_at	What changed and by whom.
outcome_measurements	outcome_id, action_request_id, baseline_period, measurement_period, observed_delta, confidence	Whether the action worked.
rating_updates	rating_update_id, prior_rating_id, new_rating_id, change_reason	How state transitions affected the asset rating.

GOVERNED TRANSACTION PATH

1. Observe signal drift
2. Score URL and cluster risk
3. Recommend action
4. Evaluate policy
5. Approve, block, or escalate
6. Implement with change log
7. Measure outcome
8. Reconcile value and update rating

9. Policy and protected asset model

Protected assets are not simply high-value URLs. A page can be protected because it affects legal risk, brand trust, product positioning, revenue paths, pricing, compliance, investor messaging, or agent safety.

Policy object	Fields	Examples
policy_rules	policy_id, name, trigger_condition, action_scope, severity, approval_requirement, active_flag	Pricing pages require VP approval. Legal pages are no-touch. High-revenue pages cannot be redirected without executive sign-off.
protected_asset_rules	rule_id, asset_type, cluster_id, url_pattern, risk_reason, required_approvers	Brand comparison pages, migration guides, pricing pages, product category hubs, legal/compliance URLs.
permission_profiles	profile_id, user_or_agent_id, allowed_actions, blocked_actions, max_risk_level	Analyst can recommend. Agent can read and draft. Admin can approve redirects.
action_policies	policy_id, action_type, permitted_states, evidence_required, rollback_required	Merge, redirect, noindex, delete, rewrite, schema patch, internal link changes.
exception_records	exception_id, policy_id, approver_id, reason, expiration_date	Temporary override with evidence and audit trail.

10. Evidence layer and audit packs

Every rating, recommendation, and recovery claim should be traceable to evidence. Evidence is what makes the platform useful to CFOs, CMOs, RevOps, agencies, analysts, and future AI agents.

Evidence type	Stored as	Used for
Crawl export	file pointer + parsed records + timestamp	Indexability, redirects, crawl depth, orphaning, canonical conflicts.
GSC export	source table + date range + query/page join	Impressions, clicks, CTR, ranking drift, query cannibalization.
GA4 export	source table + metric map + property ID	Sessions, engagement, conversion events, landing-page value.
CRM/payment export	source table + anonymized IDs + join keys	Closed-won revenue, pipeline influence, assisted value.
SERP / AI citation capture	screenshot, HTML/text capture, citation URL, query, timestamp	Answer share, citation gaps, zero-click risk, competitor displacement.
Screenshot / render	image object + viewport + device + timestamp	Before/after proof, UX, template regressions, visual QA.
Change diff	before/after content hash, URL state, owner, timestamp	Governance, rollback, causality, proof sprint measurement.
Analyst note	structured note + linked evidence IDs	Human rationale and interpretation.
Client approval	approval record + notes + timestamp	Audit trail and protected asset control.

10.1 evidence_artifacts

Field	Type	Purpose
artifact_id	uuid	Evidence record ID.
artifact_type	enum	crawl, analytics_export, screenshot, pdf, csv, note, approval, diff, api_response.
source_system	text	Origin of evidence.
storage_uri	text	Object storage location or file pointer.
hash	text	Integrity / dedupe hash.
captured_at	timestamp	Capture time.
linked_object_type	enum	url_asset, cluster, rating, recommendation, action, report.
linked_object_id	uuid	Object supported by evidence.
confidence_score	numeric	Reliability of evidence.
notes	text	Human-readable context.

11. Agent ingress and API design

The platform should be designed so AI agents come into URL Ledger for trusted state. The agent should be a consumer, requester, or executor under policy - not the canonical source of truth.

Endpoint / capability	Reads / writes	Purpose
GET /url-assets	Read	Return canonical URL asset records and current state.
GET /url-assets/{id}/ratings/latest	Read	Return latest rating, confidence, and because trail.
GET /url-assets/{id}/evidence-pack	Read	Return linked evidence for a rating or recommendation.
GET /clusters/{id}/portfolio-summary	Read	Return cluster-level value, decay, waste, and backlog.
POST /recommendations/request	Write request	Allow an agent or system to request a new recommendation.
POST /policy/evaluate	Write evaluation	Check whether an intended action is allowed, blocked, or escalated.
POST /actions/request	Write request	Submit a proposed refresh, merge, redirect, retire, or protect action.
POST /evidence/artifacts	Controlled write	Submit evidence, screenshots, or source captures with provenance.

Endpoint / capability	Reads / writes	Purpose
GET /benchmarks/query	Read	Return benchmark ranges when permissions allow.
POST /agent-access/log	Write log	Record agent reads, requests, and attempted actions.

EXAMPLE AGENT-READABLE URL STATE

```
{
  "url_asset_id": "url_0001287",
  "canonical_url": "https://example.com/guides/pricing-comparison",
  "current_rating": "B-",
  "recommended_action": "refresh_protect",
  "policy_status": "approval_required",
  "top_risks": ["semantic_drift", "citation_gap", "conversion_decay"],
  "evidence_pack_uri": "/evidence-packs/url_0001287/latest",
  "allowed_agent_actions": ["read", "summarize", "draft_recommendation"],
  "blocked_agent_actions": ["publish", "redirect", "delete", "noindex"]
}
```

12. System components and recommended stack

The data model should support both product speed and long-term defensibility. Start simple, but do not store everything as disposable JSON blobs if the goal is a benchmarkable system of record.

Component	Recommended role	Notes
Application database	Postgres or equivalent relational store	Best for URL assets, mappings, ratings, policies, users, actions, approvals, and metadata.
Warehouse / analytics store	BigQuery, Snowflake, ClickHouse, or DuckDB for MVP	Best for high-volume observations, GSC/GA4 exports, channel metrics, time-series analysis, and portfolio rollups.
Object storage	S3/GCS/Azure Blob/local equivalent	Stores evidence artifacts, screenshots, crawl exports, PDFs, CSVs, and report files.
Search index	OpenSearch/Elasticsearch or Postgres full-text	Useful for URL, title, notes, evidence, policy, and recommendation retrieval.
Vector index	Optional; use for similarity/cannibalization/messaging fit	Useful for intent overlap, cluster modeling, duplicate content, and semantic drift detection.
Queue / jobs	Background workers	Crawls, imports, scoring jobs, report generation, evidence processing.
API layer	REST or GraphQL	Expose stable URL asset state to dashboards, BI, agents, and customer systems.
Dashboard layer	Internal admin + customer UI	Portfolio view, URL detail, ratings, actions, evidence packs, policy state.
Audit/report generator	DOCX/PDF/slide export service	Creates board-ready readouts and proof packs.

Do not overbuild too early

The MVP does not need every possible channel on day one. It needs canonical URL identity, durable signal imports, explainable ratings, a recovery backlog, and evidence-backed outputs. The benchmark and agent layers become stronger once historical state compounds.

13. MVP build sequence

Phase	Build focus	Required tables / objects	Success condition
0-30 days	Audit-ready ledger core	orgs, domains, url_assets, canonical_mappings, url_snapshots, channel_observations, url_ratings	Can ingest one domain, reconcile URLs, score every URL, and produce a basic ledger export.
31-60 days	Recovery backlog and evidence	structural_variable_scores, recommendations, evidence_artifacts, value_at_risk_records	Can generate a ranked top-25 recovery backlog with evidence and expected value.
61-90 days	Governance and proof sprint	policy_rules, action_requests, approvals, change_logs, outcome_measurements	Can track approved actions and measure before/after outcomes.
90-180 days	Agent-readable truth layer	api_clients, permission_profiles, agent_access_logs, evidence_packs	Agents and external systems can query trusted state and request actions under policy.
180+ days	Benchmark and capital allocation layer	benchmark_records, peer_sets, normalized_metric_definitions, portfolio_rollups	The system creates comparative standards and CFO-grade allocation views.

14. Example URL asset JSON

```
{
  "url_asset_id": "url_0001287",
  "domain_id": "dom_001",
  "canonical_url": "https://example.com/solutions/content-ledger",
  "url_status": "active",
  "asset_type": "solution_page",
  "funnel_role": "BOFU",
  "business_owner": "Growth",
  "protected_flag": true,
  "cluster": {
    "cluster_id": "clu_014",
    "name": "Content Asset Governance",
    "intent": "commercial_evaluation"
  },
  "current_rating": {
    "overall_grade": "B",
    "health_score": 78,
    "decay_score": 34,
    "risk_score": 61,
    "recoverability_score": 72,
    "confidence_score": 81,
    "recommended_action": "refresh_and_protect"
  },
  "value": {
    "annualized_value": 184000,
    "assisted_pipeline": 620000,
    "value_at_risk": 41000,
    "recoverable_value": 23000,
    "model_confidence": 0.74
  },
  "policy": {
    "policy_status": "approval_required",
    "required_approvers": ["CMO", "RevOps"],
    "blocked_actions": ["delete", "noindex", "redirect_without_approval"]
  },
  "agent_access": {
    "readable": true,
    "draft_recommendations_allowed": true,
    "direct_write_allowed": false
  }
}
```

14.1 Data quality score

Completeness area	Weight	Requirement
Canonical identity	20%	Canonical URL, observed variants, status, and domain relationship are known.
Performance signals	20%	At least one reliable traffic/discovery source is joined to the URL.
Value signal	20%	Conversion, revenue, assisted value, or proxy value is mapped.
Structural evidence	20%	Crawl/index/template/link evidence exists.
Governance metadata	10%	Owner, action policy, and protected status are known.
History depth	10%	At least two comparable time periods exist for trend analysis.

15. Definition of done for canonical truth

A URL Ledger implementation is not done because data has been imported. It is done when the system can answer the same question consistently across teams, dashboards, exports, and agents.

Question	Definition of done
What URLs exist?	The system reconciles sitemap, crawl, CMS, GSC, GA4, and manual imports into canonical URL asset records.
Which URL is the source of truth?	Variants, redirects, canonicals, and duplicates are mapped to stable url_asset_id values.
What is each URL worth?	The ledger stores yield, assisted value, value-at-risk, and confidence using consistent definitions.
What is wrong with it?	The 13 structural variables produce explainable scores with evidence.
What should happen next?	Recommendations are prioritized by value, effort, risk, and policy.
Who may act?	Policy rules define approvals, protected assets, permissions, and blocked actions.
Did the action work?	Outcome measurements reconcile expected vs actual impact and update the rating.
Can agents trust it?	APIs return canonical state, evidence, policy constraints, and access logs.

Final architecture line

URL Ledger should become the record layer that every dashboard, analyst, agency, executive, automation, and AI agent consults before judging or changing a website asset.

16. Appendix: field naming conventions

Convention	Rule
IDs	Use stable internal IDs: org_id, domain_id, url_asset_id, cluster_id, rating_id, action_id, artifact_id.
Dates	Use observed_at for source capture time, created_at for record creation, updated_at for application updates, valid_from/valid_to for historical truth windows.
Status enums	Keep enum states human-readable: active, redirected, retired, noindex, blocked, unknown, pending, approved, rejected, implemented, measured.
Scores	Use numeric 0-100 scores where possible, plus grade labels for executive readability.
Confidence	Every derived record should include confidence_score and source_confidence where possible.
Evidence links	Every rating and recommendation should link to one or more evidence_artifacts.
Source systems	Always preserve source_system, source_property_id, import_id, and raw_reference when possible.
Mutable vs immutable	Never mutate historical ratings, observations, or evidence. Append new records and close valid_to windows.
Raw vs normalized	Preserve raw imported URLs and metrics separately from normalized canonical records.
Agent writes	Agent writes should be requests or evidence submissions by default, not direct state changes unless explicitly approved by policy.

17. Next product assets this schema unlocks

Asset	Why it matters
Dashboard wireframe pack	Turns the schema into portfolio views, URL detail pages, ratings screens, action queues, and evidence packs.
PRD / engineering build plan	Converts the schema into development epics, user stories, acceptance criteria, and milestones.
Benchmark methodology paper	Defines how ratings and peer comparisons become defensible and monetizable.
Quarterly Content 10-K sample	Shows how the ledger can produce board-ready recurring reports.
Agent ingress policy brief	Makes the agent access layer concrete for enterprise buyers and partners.

Prepared as a working specification for the URL Ledger master asset library. This is not legal, accounting, or GAAP reporting guidance; it is a product and data-model foundation for website asset intelligence.