Publishing Repos Operational Semantics Contract v0.11.0

1. Purpose

This contract defines the operational semantics for the Probabilistic Systems Engineering publishing repository.

Its purpose is to transform committed publishing inputs into a deployed public research/library site while preserving explicit authority over:

- publish-unit discovery
- HTML export normalization
- structural classification
- authority collection projection
- output topology
- homepage, listing, and document navigation/discovery behavior
- structured metadata emission
- recommendation artifact emission
- published-state observability
- deployment and reconciliation behavior
- refusal conditions

This contract governs the publishing and deployment repository only.

Canonical prose-authoring authority remains outside this repository.

2. Scope

This contract applies to:

- authority collections and derived authority essays
- papers
- contracts
- replication and verification materials

This contract covers:

- committed incoming publish inputs
- grouped incoming folder traversal
- Google Docs HTML ZIP extraction and normalization
- rendered document generation
- authority collection landing-page generation
- derived authority essay-page generation where supported
- homepage generation
- latest and archive listing generation
- per-document navigation generation
- per-document structured metadata emission
- site-level metadata artifact generation
- recommendation artifact generation
- sitemap generation
- explicit reference discovery
- distinct recommendation/discovery surface generation
- static asset copying from committed asset roots

- deployment
- build manifest generation
- scheduled reconciliation and self-heal behavior
- refusal behavior when qualification or projection conditions are not satisfied

This contract does not yet cover:

- comments
- social features
- broad semantic search
- tags
- arbitrary author-curated taxonomy systems
- sidecar authoring metadata files
- arbitrary generalized multi-tab reconstruction across all document classes
- opaque embedding-only recommendation authority
- Google Drive ingestion
- repo-stored generated HTML trees as committed source state

3. Repository Roles

3.1 Canonical Authoring Source

Canonical document authority remains outside this repository.

The repository SHALL NOT be treated as the prose authoring source of truth.

3.2 Repository Role

The repository is the publishing and deployment system.

It stores:

- committed publish inputs
- build scripts
- templates
- workflow definitions
- minimal site scaffolding
- supported PDF-only contract entries under `contracts/`, where explicitly allowed by this contract

3.3 Generated Output

Generated site output is derived state only.

`dist/` SHALL be treated as build output.

Generated HTML, extracted assets, derived metadata artifacts, recommendation artifacts, and other `dist/` contents SHALL NOT be committed back into git as part of normal publishing behavior.

4. Source Artifact Model

4.1 Incoming Roots

Committed governed source roots SHALL be:

- `incoming/authority/`
- `incoming/papers/`
- `incoming/contracts/`
- `incoming/replication/`
- `incoming/assets/`

4.2 Publish-Unit Roots vs Asset Roots

`incoming/authority/`, `incoming/papers/`, `incoming/contracts/`, and `incoming/replication/` are publish-unit roots.

`incoming/assets/` is not a publish-unit root. It is a committed static-asset source copied into `dist/assets/`.

Files under `incoming/assets/` SHALL NOT be interpreted as candidate publish units.

4.3 Grouping Folders

Grouping folders MAY exist at arbitrary depth under a publish-unit root.

Grouping folders are organizational only.

A grouping folder SHALL NOT itself be treated as a publishable unit unless it independently satisfies §4.4.

4.4 Renderable Publish Unit

A renderable publish unit is a directory under a publish-unit root that contains:

- exactly one `*.pdf`
- exactly one `*.zip`

A candidate directory that does not satisfy both requirements SHALL NOT be published.

For a candidate publish-unit directory:

- if it contains exactly one PDF and exactly one ZIP, it SHALL qualify as a publish unit
- otherwise build SHALL fail for that directory
- the system SHALL NOT guess intended grouping or descend further in search of a fix

4.5 Authority Publish Unit

An authority publish unit remains one publish unit even when it emits:

- one collection landing page

- zero or more derived authority essay pages

Derived authority essay pages SHALL NOT be treated as separately discovered incoming publish units.

4.6 Publish-Unit Identity

The publish-unit identity SHALL be its relative path from the incoming type root.

Examples:

- `incoming/papers/foo/` → slug `foo`
- `incoming/papers/program-a/paper-1/` → slug `program-a/paper-1`
- `incoming/authority/failure-archaeology/authority-boundaries-v0.3/` → slug `failure-archaeology/authority-boundaries-v0.3`

The full relative slug path is authoritative for output routing and grouping.

4.7 Display Label

Default display label SHALL be the PDF stem unless a more specific governed rule overrides it for a derived authority essay page.

No sidecar authoring metadata file is required by this contract.

5. Discovery Semantics

5.1 Traversal

Discovery SHALL traverse each governed incoming publish-unit root recursively.

Traversal order SHALL NOT affect governed output semantics.

5.2 Candidate Directory Rule

If a directory under a publish-unit root contains any governed publish artifacts (`*.pdf` or `*.zip`), that directory MUST be treated as a candidate publish-unit directory.

5.3 Qualification and Refusal

Qualification and refusal SHALL follow §4.4.

The system SHALL fail explicitly rather than guess intended grouping, parentage, or primary content shape.

5.4 Traversal Termination

If a directory qualifies as a publish unit, traversal below that directory SHALL terminate.

Child directories beneath a qualified publish unit SHALL NOT be independently discovered or published.

6. HTML ZIP Handling

6.1 Accepted Export Shape

The system SHALL accept Google Docs HTML exports packaged as ZIP files.

6.2 Extraction Rule

For each qualified publish unit, the ZIP SHALL be extracted into a temporary working directory.

6.3 Main HTML Detection

After extraction:

- if exactly one `.html` file exists anywhere in the extracted tree, it SHALL be used as the main document
- if zero `.html` files exist, build SHALL fail
- if more than one `.html` file exists, build SHALL fail

The system SHALL NOT guess a primary HTML file.

6.4 Asset Preservation

Assets required for correct rendered output SHALL be preserved in published output.

6.5 Authority Collection Projection

For authority publish units:

- the normalized export MAY be segmented into derived authority essay outputs only when deterministic section boundaries are available from the normalized export
- failure to infer section boundaries SHALL NOT cause guess-based projection
- if deterministic section boundaries are unavailable, the system MAY still publish the authority collection landing page
- if deterministic section boundaries are unavailable, the system SHALL NOT emit guessed, inferred, or heuristically fabricated child essay pages

6.6 Projection Refusal Boundary

The system SHALL fail rather than guess when ambiguity affects:

- whether a candidate qualifies as a governed publish unit
- whether the main HTML file can be determined
- whether a proposed child page has a deterministic section boundary sufficient to justify separate rendered emission

The absence of child-essay projection alone does not require failure if the collection landing page remains deterministically renderable.

7. Normalization Semantics

7.1 Head and Metadata Normalization

Rendered pages SHALL normalize or inject:

- page title
- page description
- author metadata
- canonical URL
- per-page structured metadata

7.2 Exported Style Preservation

Exported Google Docs `<style>` blocks MAY be preserved where required for text fidelity.

7.3 Empty Paragraph Cleanup

Paragraphs containing no meaningful text and no structural content SHALL be removed.

Paragraphs containing structural content such as images, tables, SVG, or rules SHALL NOT be treated as empty.

7.4 Generic Structural Cleanup

The system MAY remove or normalize generic Google Docs export leakage where that cleanup does not alter governed reading semantics.

7.5 Canonical Rendered Title

Each rendered page SHALL expose one canonical rendered title.

7.6 Competing Title Cleanup

Where exported HTML contains competing title-like wrappers that clearly represent the same document title, the system MAY canonicalize them down to one rendered title.

The system SHALL prefer omission over duplicate title noise.

7.7 Authority Essay Title Authority

For derived authority essay pages:

- the canonical essay title SHALL be taken from the segmentation boundary metadata used to create the essay page
- cleaned body content SHALL NOT be treated as the primary title authority

7.8 Duplicate Wrapper Removal for Authority Essays

For derived authority essay pages only:

- leading title-wrapper paragraphs or duplicate title blocks that normalize to the canonical essay title MAY be removed from body content

- truncated wrapper titles MAY be removed only when they clearly normalize to the canonical essay title

- the system SHALL NOT generically remove the first heading unless it is a duplicate of the canonical essay title

7.9 Structural Classification

The system MAY classify normalized blocks into governed presentation classes such as lead-in, callout, compact paragraph, and similar presentational categories where implemented.

This contract governs the output behavior, not one mandatory internal classifier implementation.

8. Homepage Semantics

8.1 Homepage Role

The homepage is a governed public entry surface for the published site.

It SHALL reflect the current site role as a research/library surface rather than only an archive landing page.

8.2 Required Homepage Contents

The homepage SHALL include:

- site title

- author attribution

- concise public framing text

- primary CTA for Authority

- secondary CTA for proof/papers entry

- browse/library CTA

- a primary-path summary emphasizing Authority and Papers

- secondary/support exposure for Contracts and Replication & Verification

8.3 Homepage Role Hierarchy

Homepage ordering SHALL reflect role hierarchy, not only box order.

The primary path SHALL emphasize:

- Authority
- Papers

Supporting surfaces SHALL include:

- Contracts
- Replication & Verification

The browse/library path SHALL expose public browsing into the larger corpus.

8.4 Homepage Reachability

Every homepage entry MUST expose a valid primary navigation target.

8.5 Homepage Proof Entry

The homepage secondary proof/papers CTA SHALL resolve deterministically to a designated proof-entry family or designated proof-entry artifact as defined by implementation under this contract.

That choice SHALL NOT be left as silent hardcoded version-specific authority without explicit designation.

If no qualifying proof-entry target exists, the CTA SHALL be suppressed rather than dead-link.

9. Listing Semantics

9.1 Listing Surfaces

The system SHALL emit:

- `dist/latest/index.html`

- `dist/archive/index.html`

9.2 Listing Sections

Listing surfaces SHALL expose:

- Authority
- Papers
- Contracts
- Replication & Verification

9.3 Deterministic Listing Order

Section order, group order, family order, and item order SHALL be deterministic and SHALL respect current version/lineage rules.

9.4 Authority Collection Listing Integrity

Listing surfaces MAY expose derived authority essay links beneath their parent authority collection entry.

Authority derived essays SHALL NOT be flattened into independent peer top-level list items when they are already being represented via their parent collection entry.

9.5 Latest Visibility

`latest` SHALL:

- surface current/latest items only, subject to current family/version rules
- include authority collection entries with their essay children when available
- not flatten authority essays into peer top-level artifacts

9.6 Archive Visibility

`archive` SHALL:

- surface latest and older lineage, subject to current family/version rules
- preserve collection/child relationship for authority collections
- maintain stable reachability for versioned artifact URLs

10. Structured Metadata Semantics

10.1 Per-Rendered-Document Metadata

Every rendered HTML document SHALL emit required derived metadata and page-level structured metadata.

10.2 Site Metadata Index

`dist/metadata/documents.json` SHALL contain one metadata entry per rendered document.

10.3 Authority Collection Metadata

For authority collection landing pages, metadata MAY include:

- essay count
- child essay links
- collection relationship information
  when such data exists deterministically

10.4 Derived Authority Essay Metadata

For derived authority essay pages, metadata MAY include:

- parent collection slug
- essay order within collection
- collection-relative navigation fields

Such metadata is implementation-supporting derived metadata, not sidecar authoring metadata.

10.5 Metadata Completeness Boundary

Only rendered HTML documents are required members of rendered-document metadata completeness under this contract.

PDF-only contract entries are not rendered HTML documents and are excluded unless a future contract version explicitly changes that rule.

11. Discovery and Recommendation Semantics

11.1 Distinct Discovery Surfaces

When emitted, the following surfaces SHALL remain distinct and labeled:

- Referenced artifacts
- Read next
- Related
- Verification & replication
- Version relationship guidance

11.2 Recommendation Computation Timing

Recommendations SHALL be computed at build/publish time over the full rendered document corpus.

11.3 Candidate Generation Latitude

Candidate generation MAY use corpus-wide textual similarity over cleaned rendered document text.

Metadata MAY be used for reranking or policy shaping.

This contract does not freeze one specific scoring formula or threshold implementation beyond the governed outcomes below.

11.4 Hard Exclusions

Normal recommendation/discovery surfaces SHALL exclude, where applicable:

- the current document
- older same-family versions where disallowed
- current-family duplicates where disallowed
- archive-only artifacts when not appropriate
- low-confidence candidates

11.5 Reference Conservatism

Referenced-artifact detection MUST prefer omission over weak or ambiguous matches.

11.6 Related-Docs Conservatism

Heuristic related-document recommendations MUST be thresholded, suppressible, and conservative.

False positives are worse than no recommendation.

11.7 Authority Essay Read-Next Precedence

For derived authority essay pages:

- ordered collection navigation takes precedence for Read next
- Read next SHALL resolve to the next essay in collection order when one exists
- generic recommender behavior SHALL NOT override ordered collection progression

This clause governs Read next precedence only. It does not require blanket suppression of other discovery surfaces when they otherwise qualify.

12. Version and Lineage Semantics

12.1 Slug-Family Authority

Version-family derivation MAY only use the final slug segment and terminal version suffix pattern.

12.2 Stable Historical Reachability

Versioned artifact URLs SHALL remain stable even when newer versions exist.

12.3 Latest-Only Main Visibility

When multiple versions exist in the same slug family, only the latest version appears in latest-oriented primary listing contexts, subject to family rules.

12.4 Authority Child Lineage

Authority collection child essay pages are not independent version families derived from essay titles.

They inherit collection lineage from the parent publish unit.

12.5 No Cross-Collection Latest Collapse for Child Essays

Collection child essays SHALL NOT be independently latest-collapsed across unrelated parent collections.

13. Document Navigation Semantics

13.1 Depth-Correct Navigation

Per-document home navigation MUST resolve correctly for the document's relative slug depth.

13.2 Collection-Relative Navigation

Authority essay navigation SHALL be ordered and collection-relative.

13.3 Collection Back-Link

Derived authority essay pages SHALL expose navigation back to the parent collection landing page.

13.4 Previous/Next Essay Navigation

When projection succeeds and adjacent essays exist, derived authority essay pages SHALL preserve deterministic previous/next navigation according to collection order.

13.5 Collection Landing Essay List

Authority collection landing pages MAY render an ordered essay list when derived essay pages exist.

13.6 PDF Navigation for Authority Collections

Per-document PDF navigation for authority collection-derived pages SHALL remain rooted in the collection output directory as implemented under this contract.

14. PDF-Only Contract Entry Semantics

14.1 Supported PDF-Only Mode

PDF-only contract entries stored under `contracts/` are a supported contract publication form.

14.2 Scope Restriction

PDF-only contract entry mode is supported only for contracts.

This mode SHALL NOT be generalized by this contract to papers, authority collections, or replication materials.

14.3 PDF-Only Entry Behavior

A PDF-only contract entry:

- SHALL link directly to the PDF as its primary title target
- MAY be surfaced in contract listing/navigation contexts
- SHALL expose PDF action semantics only
- SHALL NOT imply the existence of a rendered HTML page

14.4 Precedence

If both a rendered incoming contract and a PDF-only contract entry exist for the same slug:

- the rendered incoming contract SHALL take precedence
- the PDF-only contract entry SHALL NOT be separately surfaced

14.5 Rendered-Surface Exclusion

PDF-only contract entries SHALL NOT be treated as rendered documents for:

- rendered-document metadata completeness
- HTML sitemap scope
- rendered-document recommendation corpus membership

unless a future contract version explicitly adds such support.

15. Static Asset Semantics

15.1 Static Asset Copy

Committed assets under `incoming/assets/` SHALL be copied to `dist/assets/`.

15.2 Non-Publish-Unit Rule

Static assets SHALL NOT be treated as publish units.

15.3 Preservation

Static asset copy semantics SHALL preserve relative asset reachability required by governed rendered output.

16. Sitemap Semantics

16.1 Sitemap Output

The system SHALL emit `dist/sitemap.xml`.

16.2 HTML Reachability Constraint

Sitemap entries MUST resolve to valid published HTML pages.

16.3 Completeness Boundary

This contract requires sitemap validity for included entries.

This contract does not require every non-listed reachable artifact class to appear in sitemap unless otherwise stated by a future version.

17. Build Manifest and Published State

17.1 Build Manifest

Successful builds MUST emit `dist/build.json`.

17.2 Manifest Sufficiency

The manifest MUST be sufficient for live published-state observability and drift comparison.

17.3 Recommendation Artifact

When discovery/recommendation emission is enabled, successful builds SHALL emit `dist/metadata/recommendations.json`.

18. Deployment and Reconciliation

18.1 Deployment Source

Deployment SHALL publish from `dist/`.

18.2 Reconciliation Role

Scheduled reconciliation MAY republish generated site output but MUST NOT mutate repository source artifacts.

18.3 Drift Detection

Scheduled reconciliation MUST determine drift from live manifest readability and manifest/source identity sufficient to compare current published state to the current source state for the run.

18.4 Self-Heal

When governed drift is detected, reconciliation MAY republish generated output to restore intended published state.

19. Invariants

- INV-001 Repository Role Separation
  Canonical prose authority remains outside the publishing/deployment repository.

- INV-002 Derived Output Non-Authority
  `dist/` and generated artifacts are derived state only.

- INV-003 Supported Content Types
  Supported content types are authority, papers, contracts, and replication/verification.

- INV-004 Publish Unit Qualification
  Renderable publish units require exactly one PDF and exactly one ZIP.

- INV-005 Traversal Termination
  Discovery terminates below a qualified publish unit.

- INV-006 Main HTML Uniqueness
  Exactly one extracted HTML file is required; the system does not guess a primary HTML file.

- INV-007 Explicit Failure over Guessing
  The system fails explicitly rather than guessing when qualification, primary HTML detection, or child-page justification is ambiguous.

- INV-008 Authority Collection Projection Boundary
  A qualified authority collection may emit one collection landing page and zero or more derived authority essay pages.

- INV-009 No Guess-Based Essay Projection
  Derived authority essay pages may be emitted only from deterministic section boundaries.

- INV-010 Authority Essay Navigation
  Derived authority essay pages preserve deterministic collection order and previous/next navigation when projection succeeds.

- INV-011 Homepage Reachability
  Every homepage entry exposes a valid primary navigation target.

- INV-012 Deterministic Listing Order
  Homepage, latest, and archive section/group/item ordering are deterministic and respect applicable lineage rules.

- INV-013 Authority Listing Integrity
  Authority collection listings may expose child essay links without flattening them into independent top-level listing entries.

- INV-014 Published State Observability
  Successful builds emit a manifest sufficient for live published-state observability.

- INV-015 Self-Heal Without Source Mutation
  Scheduled reconciliation may republish generated output but does not mutate source artifacts.

- INV-016 PDF-Only Contract Precedence
  Rendered incoming contracts take precedence over PDF-only contract entries with the same slug.

- INV-017 Sitemap Reachability
  Sitemap entries resolve to valid published HTML pages.

- INV-018 Per-Document Metadata Emission
  Every rendered document emits required derived metadata and structured metadata.

- INV-019 Site Metadata Index Completeness
  `dist/metadata/documents.json` contains one metadata entry per rendered HTML document.

- INV-020 Explicit Reference Conservatism
  Referenced-artifact detection prefers omission over weak or ambiguous matches.

- INV-021 Related-Docs Conservatism
  Related-document recommendations are thresholded, suppressible, and conservative.

- INV-022 Distinct Discovery Surfaces
  Referenced, read-next, related, verification/replication, and version-guidance surfaces remain distinct when emitted.

- INV-023 Slug-Family Authority
  Version-family derivation may use only final-slug terminal version patterns.

- INV-024 Stable Historical Reachability
  Versioned artifact URLs remain stable even when newer versions exist.

- INV-025 Latest Visibility Constraint
  Latest-oriented listing contexts expose only latest items subject to lineage rules.

- INV-026 Collection-Lineage Inheritance
  Authority child essays inherit lineage from the parent collection and are not independent essay-title version families.

- INV-027 Static Asset Copy
  Committed assets under `incoming/assets/` are copied to `dist/assets/` and are not treated as publish units.

- INV-028 Recommendation Artifact Emission
  Recommendation artifact emission is deterministic at publish time when enabled.

- INV-029 PDF-Only Rendered-Surface Exclusion
  PDF-only contract entries are not treated as rendered HTML documents for rendered-document-only surfaces.

20. Non-Goals

This contract does not require:

- comments or social features
- sidecar authoring metadata files
- arbitrary taxonomy systems
- full-text or embedding-only search as an authority surface
- arbitrary generalized multi-tab reconstruction across all document classes
- pixel-perfect fidelity to Google Docs layout
- repo-stored generated HTML trees
- historical relocation of older versions
- flattening authority child essays into top-level peer listing artifacts
- treating PDF-only contract entries as rendered HTML documents

21. Acceptance Criteria

This contract is satisfied when the system can:

1. consume incoming inputs under `incoming/authority/`, `incoming/papers/`, `incoming/contracts/`, and `incoming/replication/`
2. copy static assets from `incoming/assets/` to `dist/assets/`
3. qualify only candidate directories containing exactly one PDF and one ZIP
4. terminate traversal below qualified publish units
5. extract and normalize Google Docs HTML export content
6. fail rather than guess when no unique main HTML file exists
7. canonicalize duplicate title wrappers conservatively
8. generate collection landing pages for authority publish units
9. generate derived authority essay pages only when deterministic section boundaries are available
10. refrain from guessed child-essay projection when deterministic section boundaries are unavailable
11. generate rendered pages under `dist/authority/...`, `dist/papers/...`, `dist/contracts/...`, and `dist/replication/...`
12. generate homepage, latest, and archive surfaces consistent with the governed role hierarchy
13. keep authority child essays nested under their parent collection in listing contexts rather than flattening them into peer top-level items
14. emit per-document structured metadata and site-level metadata index artifacts
15. emit `dist/metadata/recommendations.json` when recommendation/discovery emission is enabled
16. emit `sitemap.xml`
17. emit `build.json` with required published-state fields

18. detect explicit references conservatively
19. emit distinct discovery/recommendation surfaces at publish time
20. preserve old versioned URLs while showing only latest versions in latest-oriented listing contexts subject to lineage rules
21. resolve collection and per-document navigation correctly
22. support PDF-only contract entries under `contracts/` without implying rendered HTML pages
23. enforce rendered incoming contract precedence over same-slug PDF-only contract entries
24. deploy from `dist/`
25. detect governed drift and republish on drift
26. avoid mutating source artifacts during reconciliation