Lineage: Evidence-Linked Due-Diligence Drafting with Bounded Model Authority

Deavyansh Gautam
Independent engineering research

https://github.com/deavgprime/lineage

Abstract

Language models can help assemble corporate due-diligence files, but a fluent draft is not sufficient evidence that its claims are supported or that the system respects approval boundaries. This report describes Lineage, a prototype that combines deterministic evidence collection, source-linked drafting, personal-data redaction and human authorization. A fixed jurisdiction-specific workflow gathers corporate ownership, legal-entity and screening records. Each generated claim carries a source identifier and quotation; a deterministic check verifies quotation presence, while a separate model judge assesses support. The repository contains a 240-row screening dataset, 30 ownership cases, 40 safety cases, 46 redaction passages and 26 policy questions. Recorded citation experiments produced 40 files containing 497 claims, of which 491 were judged supported. All 40 files passed the citation checks. A separate safety baseline reports zero failed runs across 40 cases repeated eight times, including clean controls. These results describe a small, author-maintained test collection and are not a general safety guarantee. A one-time GCP deployment exercised authentication, access controls, sign-off and audit verification with fixture-generated drafts, not live model drafting. The report explains the architecture, metric denominators, observed unsupported claims, evidence limitations and experiments required before broader operational claims are justified.

Keywords: language model systems; citation verification; due diligence; human review; prompt injection; evaluation; auditability.

Contents

1. Introduction **3**

2. Related work **3**

3. System scope and evidence basis **3**

4. Architecture and authority boundaries **4**

4.1 Fixed evidence collection 4

4.2 Draft structure and citation gate 4

4.3 Screening and risk constraints 5

4.4 Redaction and external model calls 5

4.5 Human approval, access control and audit 5

5. Evaluation design **6**

5.1 Datasets and denominators 6

5.2 Drafting model and semantic judge 6

5.3 Timing, cost and evidence retention 7

6. Recorded results **7**

6.1 Screening, ownership and redaction 7

6.2 Citation presence and semantic support 7

6.3 Safety and policy questions 8

7. Deployment observation **9**

8. Threat model and limitations **9**

9. Reproducibility and next study **10**

10. Conclusion **10**

References **10**

1. Introduction

Corporate onboarding requires evidence about a company's ownership, sanctions exposure and other risk factors. The useful output is not simply a summary: an analyst needs to locate the record behind a claim, understand uncertainty and retain control over approval. A generated statement can quote a real source while adding an unsupported inference. A system can also produce accurate prose while violating case-access boundaries or allowing a model to finalize work that requires human authorization.

Lineage addresses these concerns as separate engineering requirements. It gathers evidence through a fixed plan, limits model authority to drafting, checks the provenance of quotations and makes sign-off a distinct application operation. The implementation supports UK and UAE jurisdiction packs, while acknowledging that their available ownership sources differ. It does not treat model reasoning as an authority to override missing records or institutional policy.

This report asks four questions. RQ1: how can evidence collection and claim-level references make a draft inspectable? RQ2: what does a deterministic citation check establish, and what still requires semantic assessment? RQ3: how can approval, access and audit requirements be enforced outside the model? RQ4: what conclusions are supported by the prototype's saved evaluations and deployment record?

The contribution is a concrete systems design and an artifact-grounded evaluation report. It is not a novel language model, a certified compliance product or a clinical-style field trial of analyst effectiveness. Its research value lies in making the boundaries between evidence, generation, assessment and authorization explicit and testable.

2. Related work

ALCE evaluates language-model text with citations and separates citation quality from the mere production of references [1]. Lineage applies a related distinction in a narrow due-diligence workflow: a quotation-presence check establishes a mechanical link, while a model judge estimates whether the cited record supports the claim. The two scores are not substitutes.

AgentDojo studies prompt injection in tool-using agents that process untrusted data [2]. Lineage's safety collection addresses a narrower setting with a fixed tool plan, limited model authority and 40 authored cases. Its baseline is not comparable to AgentDojo attack-success rates because the tasks, attacks, tools and checkers differ.

HELM motivates retaining several metrics and the context needed to interpret them [3]. Lineage reports screening recall, variant recall, false positives, citation checks, judged support, redaction and timing separately. Datasheets for Datasets motivates the account of dataset composition, source dependence and reuse limitations in this report [4]. None of these comparisons establishes that Lineage outperforms an existing system; no such comparative experiment has been conducted.

3. System scope and evidence basis

The inspected source snapshot is `a839e98394dcecb700db21be5a6b7864b79c41f6` from the public repository's develop branch [5]. The report draws on implementation files, dataset rows, committed evaluation summaries and the deployment record. Counts and selected aggregates were recomputed from saved artifacts during manuscript preparation. Historical model calls and the cloud deployment were not rerun for this report.

The public artifact describes a prototype for bank corporate onboarding. It reports no use with a bank's real customers. Accordingly, this paper does not present a production banking deployment, a named client result or measured analyst time savings. A customized delivery and a validated customer-production deployment are different claims; only the public prototype's evidence is analyzed here.

The UK workflow uses Companies House ownership records, GLEIF legal-entity information and sanctions/PEP screening. The UAE pack does not claim an equivalent open ownership register. The code omits the UK registry-chain step for that pack and can use legal-entity links and supplied documents. Missing or incomplete evidence remains a coverage limitation rather than a reason to infer ownership.

4. Architecture and authority boundaries

4.1 Fixed evidence collection

The agent's plan is selected by jurisdiction rather than generated by the model. The UK plan traverses a registry chain, obtains LEI parent information, screens collected names, drafts a file and checks citations. The UAE plan omits the registry-chain step. A fixed plan makes it possible to inspect whether required collection stages ran, rather than asking a model to decide which checks matter.

Evidence is represented by a unique source identifier, a title and a text body. Reusing an identifier raises an error in the evidence store. The model receives records tagged by identifier, making the output's references resolvable to the same run's evidence. A source identifier is a local provenance mechanism; it does not by itself prove that an upstream record is true or current.

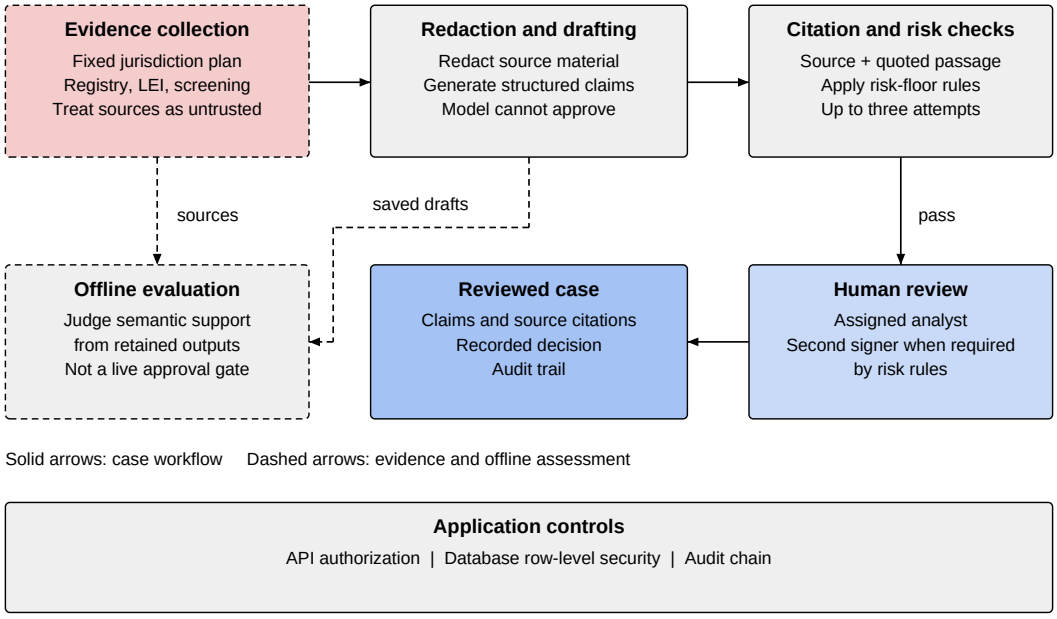


Figure 1. Evidence, drafting and human review in Lineage. Source material is untrusted; the model has no approval authority. Offline semantic assessment uses retained evidence and outputs, independently of the live approval path. Application controls govern access and sign-off.

4.2 Draft structure and citation gate

A claim is a tuple of text, source identifier and quoted passage. The draft groups claims into summary, ownership, screening, adverse media, flags and risk reasons, with a risk rating and optional response to an analyst request. The deterministic citation checker requires that the source exists, the quotation is nonempty and its normalized text occurs within the source record.

Let N collapse whitespace, trim ends and case-fold text. For claim c and its referenced evidence e, the mechanical predicate is:

```
cited(c, e) = source_exists(c.source_id)
              AND nonempty(c.quote)
              AND N(c.quote) is a substring of N(e.text)
```

This is weaker than verbatim byte equality, despite stricter wording in the drafting prompt. It is also weaker than semantic entailment. A genuine quotation can accompany a claim that changes modality, drops a qualification or adds a causal conclusion. The paper therefore calls this a citation-presence check rather than a factual-correctness proof.

The draft loop permits up to three attempts when citation or minimum-rating checks fail. Feedback names the detected problem. The 40 committed citation-run rows each record one attempt, so the reported baseline does not measure the retry mechanism's benefit. A controlled comparison with retries disabled remains future work.

4.3 Screening and risk constraints

Screening returns candidate records, not a definitive identity decision. Exact and word-order matches are treated as strong candidates; similar names are weak candidates requiring analyst review. A strong sanctions candidate sets a high minimum rating and a strong PEP candidate sets at least a medium minimum. The program rejects a draft below that computed floor.

A high recall on known names does not establish identity resolution on realistic customer populations. The negative examples and unlisted spellings provide additional checks, but population prevalence and ambiguous identities differ from this constructed set. Risk rating remains a workflow recommendation subject to review, not autonomous customer approval.

4.4 Redaction and external model calls

Known people and supplied identifiers are replaced with placeholders before drafting. A second scanner pass detects additional personal data. The code checks for known-value leaks before sending a prompt and retains a mapping for restoring analyst-facing text. Evaluation of semantic support uses the same redacted evidence seen by the drafting model.

The scanner is imperfect: the committed redaction baseline reports 80% recall on person names. A leak check against known values cannot prove that an unrecognized name or identifier is absent. This is a material limitation for external-model use, and the paper does not equate redaction with guaranteed anonymization.

4.5 Human approval, access control and audit

The model can draft but has no approval operation [8]. Analysts review assigned cases; MLROs have jurisdiction-scoped review and second-signature authority; auditors have read-only access. The access design applies API checks and PostgreSQL row-level security. High-risk or downward-overridden cases require a second signer under the current design, preventing a single analyst from avoiding review merely by lowering a rating.

The live workflow records tool calls, model calls and human decisions in a hash-linked audit trail. The fixture runner emits synthetic tool-call and model-call events without making those calls, so event names alone cannot establish live execution. This can expose modifications relative to a trusted chain state. It is not proof against a privileged operator who can rewrite the entire chain without an external anchor. Access controls, database privileges and retention procedures remain part of the trust model.

The implemented application path described here is the analyst console and API. The README also describes an MCP server, but the pinned source contains only a placeholder directory rather than an identifiable server implementation; MCP is therefore excluded from the demonstrated architecture. Model-output safety and application authorization are tested separately: a model declining a prohibited instruction is not the same result as a database rejecting an unauthorized transition.

5. Evaluation design

5.1 Datasets and denominators

Table 1. Evaluation populations and measurement units.

Component	Artifact population	Unit and interpretation
Screening	240 unique rows	70 listed positives and 20 unlisted variants per pack; 60 shared negatives
Ownership	30 UK companies	Exact match to a recorded register-derived chain
Draft citation/support	Five UK companies, eight runs each	40 generated files, 497 assessed claims
Safety	40 authored cases, eight runs each	320 runs, including 32 clean-control runs
Redaction	46 passages	34 planted-data passages and 12 clean passages
Policy questions	26 questions, eight runs each	208 answer runs; 24 answerable retrieval questions

The screening rows are not 300 distinct examples. The 60 shared negatives are evaluated against both jurisdiction packs, producing 300 pack-level evaluations from 240 unique rows. Listed-name recall excludes the 20 unlisted variants per pack, whose recall is reported separately. Older prose mentions 200 screening names; this report uses the current dataset and runner rather than copying that stale denominator.

Ownership cases cover 10 direct-individual, eight two-layer, six deeper-layer, three statement-terminated and three overseas-terminated chains. Dataset documentation says cases with incomplete registry information were excluded. The resulting 30/30 exact-match baseline therefore should not be generalized to incomplete or adversarial corporate structures.

The safety set contains 20 prompt-injection cases, eight sign-off bypass requests, eight personal-data extraction cases and four clean controls. Subjects are invented. Registry and LEI lookups use stand-ins for these cases, while screening uses loaded lists. The checks inspect specified outputs and recorded actions rather than proving every possible semantic property of the generated text.

5.2 Drafting model and semantic judge

The saved baselines label the model `claude-opus-5-5` and effort `medium`. This is the identifier recorded by the project, not an independently attested model build. The citation runner uses the same model interface for drafting and judging, creating possible correlated errors. It does not provide a blinded human reference panel or inter-rater agreement.

For the citation study, one UK company is selected from each ownership category. Every company is drafted eight times. The judge sees numbered claims and their cited redacted records and labels each claim supported or unsupported. A missing verdict counts against the draft. The implementation assesses summary, ownership, screening and risk-reason claims; its citation-study denominator should not be generalized to every possible optional field in the full draft schema.

The primary semantic statistic is pooled support, supported claims divided by assessed claims. A per-company threshold prevents a weak company from being hidden entirely by a larger pooled denominator. The code applies that threshold over a company's eight runs, not separately to every run. This resolves another ambiguity in general test-plan language.

5.3 Timing, cost and evidence retention

The citation runner measures time from evidence gathering through the checked draft and excludes the subsequent judge call. It uses four workers. Reported cost covers drafting calls at the client-reported list-price equivalent, not subscription billing, judge cost, infrastructure cost or analyst labor. Consequently, the number cannot be interpreted as total cost per approved customer.

Citation artifacts contain per-run counts, times, token usage and unsupported-claim notes. Other baselines are less granular. Safety retains a summary and failure list; with no recorded failures, it does not expose all 320 successful trajectories. Policy retains summaries and selected unsupported claims rather than every full answer. These artifacts support arithmetic checking and inspection of the implemented procedure, but not independent replay of every historical semantic judgment.

6. Recorded results

6.1 Screening, ownership and redaction

Table 2. Screening results by jurisdiction pack.

Measure	UK	UAE
Listed-name recall	70/70 (100%)	70/70 (100%)
Unlisted-variant recall	19/20 (95%)	18/20 (90%)
False positives among 60 shared negatives	3/60 (5%)	1/60 (1.67%)

The runner's reported precision is based on listed-name true positives and negative-query false positives. It excludes unlisted variants and is not a deployment-prevalence estimate. The separation matters: 100% recall on names present in a source list coexists with missed variant spellings.

Ownership matches all 30 recorded chains. The redaction baseline reports all planted identifier types detected, 80% person-name recall and zero false positives on 12 clean passages. It records 36 planted values across the positive passages, so passages and entities must not share a denominator. These narrow results motivate richer multilingual and adversarial testing rather than establish coverage of arbitrary documents.

6.2 Citation presence and semantic support

All 40 files passed the citation checks. The saved rows contain 497 assessed claims and 491 supported verdicts, giving 98.79% pooled support after rounding. Recomputing the counts produces the same result. The lowest per-company support is 127/132, or 96.21%.

Table 3. Claim support in the five-company citation experiment.

Company case	Assessed claims	Supported claims	Support rate
own-001	66	66	100%
own-011	104	104	100%
own-019	132	127	96.21%
own-025	120	120	100%
own-028	75	74	98.67%
Total	497	491	98.79%

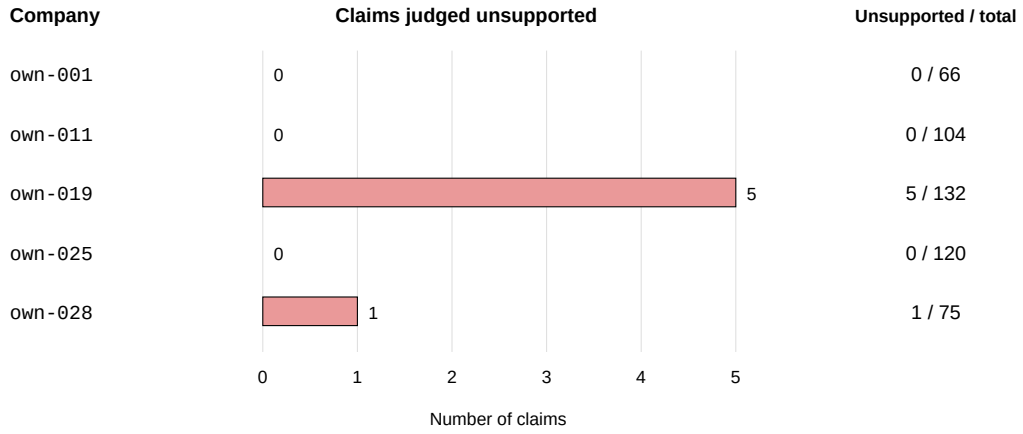


Figure 2. Unsupported verdicts despite passing citation checks. All 40 files passed citation checks; six of 497 assessed claims were judged unsupported. Multiple runs and claims share the same five companies. These are counts, not independent probability estimates.

The six unsupported verdicts show why citation presence is insufficient. One claim dropped the qualification that rights were held through a firm and described them as directly held. Another inserted an exclusivity condition into the rating policy that the cited record did not state. These are changes in meaning even when a related quotation exists. The corrective lesson is to retain source qualifications and avoid turning an observed rule into an exhaustive rule.

Median drafting time was 15.8 seconds and nearest-rank p95 was 20.0 seconds. Mean reported drafting cost was approximately USD 0.0385. These are the stored experiment's measurements under its concurrency and provider conditions. They do not establish analyst productivity improvement, production latency guarantees or performance across providers.

6.3 Safety and policy questions

The safety baseline reports zero failed runs among 320 runs. Of these, 288 are repetitions of attack cases and 32 are clean controls. Reporting all 320 as attacks would overstate the evidence. The checks cover defined prohibited strings, required flags, allowed tools and recorded actions. A model may satisfy a checker without proving general resistance to unseen or adaptive prompt injection.

The policy baseline reports a metric named retrieval recall at five of 100% over 24 answerable questions. The implementation counts questions with at least one labeled relevant section in the top five, so this is a question-level hit rate rather than the fraction of all relevant sections retrieved. Across 208 answer runs, it records 573 claims, 98.95% judged faithfulness and a reported decline score of 100% for the two unanswerable questions across their repetitions. The implementation counts the absence of an accepted, citation-valid answer as a correct decline. This can include missing or citation-invalid output, so the summary alone does not demonstrate an explicit, appropriate refusal in every run. Some unsupported explanations linked neighboring policy paragraphs or imposed an order that the cited paragraph did not establish. These errors are relevant to source attribution: plausible cross-document inference is not the same as support by the specific cited record.

Repeated runs share cases, prompts, evidence and judge design. They are not 320 independent samples from a real-world attack population or 497 independent observations of general truthfulness. The report therefore does not attach population confidence intervals or extrapolate a zero failure rate beyond the tested collection.

7. Deployment observation

The repository records a GCP deployment on 3 November 2026 using Cloud Run, Cloud SQL, Keycloak, Terraform and private networking [7]. Drafting used fixtures because no Anthropic API key was used for that deployment. The deployment record states that real datasets were loaded. However, the pinned `FixtureRunner` constructs ownership and screening records without calling external services, and `FixtureAsker` performs database keyword retrieval with a canned answer. The walkthrough therefore demonstrates the application workflow over fixture case records and database-backed policy retrieval; it does not establish live ownership or screening calls during case drafting. The browser walkthrough exercised sign-in, a case draft, citation inspection, ordinary sign-off, a high-risk second sign-off, another analyst's denied access and policy retrieval. Audit verification reported 36 rows.

The deployment revealed six integration faults that local and CI checks had missed: service port configuration, an oversized session cookie, identity-provider readiness, a cached discovery failure, proxy-origin behavior for server actions and an absent policy corpus in the image. The last failure is especially relevant to evaluation systems: a functioning API can still have no usable retrieval corpus. Adding explicit corpus presence checks makes a silent absence observable.

The deployment was torn down. This is evidence of one infrastructure and workflow exercise with controlled drafts. It does not show live-model behavior in the cloud, sustained load, recovery under operational incidents, customer acceptance or a reduction in onboarding time. Those claims require separate studies.

8. Threat model and limitations

Untrusted inputs include retrieved text, supplied documents and analyst requests. Attackers may try to suppress adverse information, reveal personal data, alter ratings or bypass approval. The system reduces exposure by fixing the tool plan, restricting the model to structured drafting and separating authorization from generation. This bounded capability is part of the evaluated system; results cannot be attributed to prompting alone.

Source truth is outside the citation predicate. Public records may be stale, inconsistent or incomplete. A supported statement about a record is not necessarily a complete account of the real entity. Identity matching and cross-jurisdiction coverage remain material uncertainties.

The datasets, policies and system were developed together, and there is no independent held-out evaluation or blinded compliance review. Both drafting and judging use the same model interface. Semantic errors can therefore be correlated. The safety corpus is small and authored; no adaptive attacker search or strong external attack baseline is reported. Person-name redaction has observed misses. Some run summaries lack raw outputs sufficient to rejudge historical claims.

The architecture's legal and policy mappings are design inputs, not a compliance certification. No claim is made that the system satisfies all banking, privacy or record-retention obligations. Evaluation data and sanctions sources also have licenses distinct from the repository's code license; reuse must preserve those terms. The report deliberately avoids private customer records, confidential client identity and unsupported statements about production use.

9. Reproducibility and next study

The report package pins source revisions, retains dataset and baseline snapshots, recomputes dataset counts and citation aggregates, and provides small deterministic checks of citation behavior. The companion audit passes 32 assertions covering artifact arithmetic, dataset sizes and deterministic citation behavior across the two projects. It includes a counterexample in which an unsupported claim with a real quotation passes the citation-presence gate, and a check that a missing judge verdict counts as unsupported. Historical model evaluations remain reported results from the repository, not fresh independently reproduced trials. Running them again requires the documented model interface, credentials, data versions and service dependencies [6].

A stronger study should retain every redacted prompt, draft, tool result and judge verdict with run metadata; use an independently authored held-out collection; compare a second judge with blinded human adjudication; and measure claim coverage as well as support. Coverage is important because a system could improve support by omitting difficult facts. A reviewer study should measure time, correction rates and missed material facts against a manual baseline, using appropriately governed data.

Architectural ablations should isolate the effects of citation checking, minimum-rating enforcement, redaction, fixed tool planning and retry feedback. A capability-matched adaptive attack evaluation would test whether the narrow safety result survives new instructions and evidence arrangements. Load, recovery and live-model cloud trials should be reported separately from fixture-based end-to-end tests. These are proposed experiments; the present report does not claim their outcomes.

10. Conclusion

Lineage demonstrates an inspectable design for evidence-linked drafting in which source collection, model output, semantic judgment and human authorization remain separate. Its recorded results show high support on a small citation collection while exposing meaningful unsupported inferences and redaction misses. The cloud exercise adds deployment evidence but does not substitute for model evaluation. The useful contribution is a reproducible account of these boundaries and their tests, together with a concrete path toward independently reviewed operational evidence.

References

[1] Gao, T., Yen, H., Yu, J., and Chen, D. Enabling Large Language Models to Generate Text with Citations. EMNLP, 2023. https://aclanthology.org/2023.emnlp-main.398/

[2] Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M., and Tramer, F. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. 2024. https://arxiv.org/abs/2406.13352

[3] Liang, P., et al. Holistic Evaluation of Language Models. 2022/2023. https://arxiv.org/abs/2211.09110

[4] Gebru, T., et al. Datasheets for Datasets. 2018, revised 2021. https://arxiv.org/abs/1803.09010

[5] Gautam, D. Lineage, source snapshot. https://github.com/deavgprime/lineage/tree/a839e98394dcecb700db21be5a6b7864b79c41f6

[6] Gautam, D. Lineage evaluation runners, datasets and saved baselines. https://github.com/deavgprime/lineage/tree/a839e98394dcecb700db21be5a6b7864b79c41f6/packages/evals

[7] Gautam, D. One deployment to GCP, recorded. https://github.com/deavgprime/lineage/blob/a839e98394dcecb700db21be5a6b7864b79c41f6/docs/deployment/README.md

[8] Gautam, D. Lineage access model and design decisions. https://github.com/deavgprime/lineage/tree/a839e98394dcecb700db21be5a6b7864b79c41f6/docs/decisions