Executive Summary

This technical assessment document describes a system designed to produce defensible, traceable analytical findings about construction project disputes under adversarial conditions. Rather than assuming complete, consistent, or neutral data, the system is explicitly engineered for environments in which records are fragmented, selectively disclosed, retrospective, or contradictory. Its purpose is not to replace legal, contractual, or expert judgment, but to establish what can and cannot be concluded from the available record and under what evidentiary constraints.

The assessment is grounded in a realistic case study involving a Guaranteed Maximum Price (GMP) public construction project affected by schedule delays, subcontractor default, owner-directed design changes, and contested claims of responsibility and cost. This scenario reflects a broader class of disputes in which multiple parties advance competing narratives, key decisions occur outside shared systems of record, and critical artifacts are missing, disputed, or revealed only after the fact. The system is designed to operate credibly in precisely these conditions.

At its core, the proposed architecture treats evidence, not narrative, as the primary analytical substrate. All project artifacts are ingested with immutable custody, provenance, and time-of-disclosure metadata. Assertions are extracted and attributed to their sources without resolving conflicts or inferring intent. Events and chronologies are reconstructed as candidate structures that preserve alternative timelines and competing interpretations. Missing or expected-but-undisclosed artifacts are surfaced as analytically relevant gaps rather than silently ignored.

Causality is handled as an explicit analytical question rather than a storytelling exercise. The system represents multiple plausible causal hypotheses for disputed outcomes, evaluates them against temporal relationships, dependencies, corroboration, and reliability signals, and preserves uncertainty where the record does not support a single defensible explanation. Findings are produced as conditioned statements that make their assumptions, limitations, conflicts, and evidentiary support explicit and inspectable.

Large language models are deliberately constrained to a non-inferential interface role: they render pre-existing analytical outputs intelligible and navigable without ingesting raw evidence or deriving new facts, interpretations, or conclusions. All analytical reasoning is performed by deterministic and bounded processes, and this separation is central to preventing hallucination, preserving auditability, and ensuring that all outputs remain traceable to underlying artifacts.

The result is a system designed not to “decide who is right” but to withstand scrutiny in adversarial settings by making the evidentiary record legible, the analytical process auditable, and the limits of knowledge explicit. In doing so, it enables more disciplined dispute resolution, more constructive challenge of stakeholder expectations, and more credible analytical outcomes in environments where truth is contested and data is imperfect.

What should the system be able to do?

Produce traceable, citable analytical findings, each explicitly linked to immutable source artifacts, about construction project chronology and contested causal claims affecting schedule, cost, and contractual outcomes, under conditions of incomplete, inconsistent, and adversarial records.

What characteristics should the system have to fulfill its mission?

1. The system ingests heterogeneous project artifacts in a manner that preserves their original form, provenance, and time of disclosure. Ingestion explicitly records who produced each artifact, when it was produced, and under what circumstances it was made available, and it treats missing, delayed, or selectively disclosed artifacts as analytically relevant signals rather than simple data gaps.
2. Rather than producing a single authoritative narrative, the system reconstructs project chronologies from heterogeneous records and evaluates competing factual assertions and causal claims about events and their impacts. It explicitly represents disagreement, uncertainty, and gaps in the available records, and it allows multiple interpretations to coexist while assessing their relative support based on provenance, timing, source reliability, and corroboration across independent sources. The system can

incorporate ongoing events as reflected in currently available artifacts while preserving the distinction between contemporaneous records and retrospective assertions.

3. The system treats causality as an analytical question rather than a narrative conclusion. For a given outcome, it represents multiple plausible causal explanations and evaluates them against temporal relationships, documented dependencies, and contemporaneous evidence, including the presence of potentially concurrent causes. Where the available records do not support a defensible single-cause attribution, the system preserves that uncertainty rather than forcing a definitive causal determination. In doing so, the system explicitly identifies conflicts between records, variations in reliability across sources, and expected artifacts that are absent, and incorporates those conditions into its analytical evaluation rather than attempting to resolve them prematurely.
4. The system preserves an auditable record of its analytical process, including the intermediate representations, explicitly stated assumptions, and evidence considered when evaluating competing assertions and causal explanations. This allows findings to be inspected, challenged, and reproduced based on the same underlying artifacts and analytical criteria.
5. The system enforces strict access controls and tenant isolation to support adversarial use cases, ensuring that artifacts, analyses, and findings are visible only to authorized parties. All access to source artifacts and analytical outputs is logged as part of the evidentiary record, preserving chain-of-custody and enabling later verification of who accessed or modified analytical materials and when.

What are the core system components and their responsibilities?

1. **Evidence Intake and Custody** is responsible for recording and preserving project submission events and the artifacts disclosed in those events. It captures non-interpretive provenance and disclosure metadata associated with each submission, including the submitting party, recipients, submission channel, disclosure time, and declared identifiers, titles, and labels provided at the time of submission. It preserves all submitted artifacts in their original form

with immutable storage guarantees and maintains a complete, auditable custody ledger that records all submissions and all subsequent access to the preserved materials.

The component produces an authoritative record of submission events, an immutable repository of submitted artifacts, and an auditable chain-of-custody ledger linking each artifact to the submission event through which it was disclosed. It does not read or interpret artifact contents, infer intent or expectations, assess completeness, resolve conflicts, or evaluate reliability.

2. **Evidence Structuring and Claim Extraction** is responsible for transforming preserved submission events and raw artifacts into structured, machine-readable representations that support downstream analytical evaluation. It parses artifact contents to extract participants, dates, locations, and other salient fields, and it identifies verbatim assertions, requests, directives, denials, and other stated positions as attributed claims linked to their source artifacts and specific text spans. It also extracts explicit identifiers and references among artifacts, such as references to RFIs, submittals, change directives, drawings, meetings, schedules, and cost items, and it normalizes these references into link candidates without reconciling conflicts or determining equivalence. The component may apply explicit, project-provided alias mappings for names or identifiers without performing identity resolution or equivalence inference.

   The component produces a set of graph-ready nodes and edges suitable for populating a project knowledge graph, including extracted entities, attributed claims with source pointers, and candidate cross-references between artifacts, all traceable back to immutable source artifacts and submission events. It does not assess truth, determine completeness, evaluate reliability, resolve contradictions, or perform causal analysis.

3. **Event and Chronology Construction** is responsible for organizing extracted claims and references into structured event candidates and temporal relationships. It identifies asserted events described in artifacts, associates

them with claimed times, locations, participants, and scopes, and links them to the submission events through which they were disclosed. The component distinguishes between event times asserted by sources and the times at which those assertions were made or disclosed, and it represents alternative or conflicting temporal claims without resolving them.

The component produces a time-indexed set of event candidates and temporal relationships, suitable for downstream analysis of sequencing, overlap, and dependency, all traceable to their originating claims and source artifacts. It does not determine which events occurred, resolve conflicting timelines, assess credibility, or perform causal or contractual evaluation.

4. **Conflict, Completeness, and Reliability Analysis** is responsible for evaluating the internal consistency and evidentiary quality of the structured record produced by upstream components. It identifies conflicting claims, incompatible timelines, and unresolved reference links, and it infers expected but undisclosed artifacts or assertions based on observed production patterns, cross-references, and submission behavior. It characterizes the evidentiary context surrounding claims and events by recording observable factors such as when assertions were made relative to the events they describe, whether they are supported by multiple independent sources, and whether disclosure around an issue is balanced or one-sided across parties. These observations are recorded as contextual signals only and do not resolve disputes or assign responsibility.

   The component produces an annotated evidence graph that flags conflicts, gaps, and reliability signals associated with claims, events, and relationships, providing downstream analytical components with explicit visibility into the strengths and limitations of the available record. It does not determine truth, interpret contracts, assign responsibility, or perform causal attribution.

5. **Causal Analysis and Dependency Evaluation** is responsible for evaluating plausible causal relationships among events, claims, and outcomes using the structured record and contextual signals produced by upstream components. It analyzes temporal sequencing, dependency relationships, and the presence

of potentially concurrent or alternative causes, and it evaluates how evidentiary gaps and conflicts constrain causal explanations. The component represents multiple competing causal hypotheses and assesses their consistency with the available record without selecting a single definitive cause.

The component produces a set of structured causal hypotheses and dependency relationships, each explicitly linked to supporting and contradicting evidence and conditioned on identified conflicts, gaps, and reliability indicators. It does not interpret contracts, assign responsibility, calculate damages, or render determinations of entitlement.

6. **Analytical Synthesis and Findings Generation** is responsible for assembling structured analytical outputs that articulate what can be concluded from the available record, and under what conditions. It synthesizes causal hypotheses, temporal analysis, and evidentiary signals into explicit findings that are scoped, qualified, and traceable to underlying evidence. Each finding explicitly states the assumptions, uncertainties, conflicts, and gaps that condition its validity, and identifies the evidence that supports or constrains it.

   The component produces citable analytical findings expressed as structured statements linked to supporting and contradicting evidence, associated causal hypotheses, and identified limitations of the record. It does not interpret contracts, apply legal standards, calculate damages, or issue determinations of responsibility or entitlement.

7. **LLM Interaction and Explanation Interface** is responsible for enabling users to explore, query, and interrogate the analytical outputs produced by upstream components through natural language interaction. It uses large language models solely to explain, summarize, reframe, and navigate existing findings, causal hypotheses, conflicts, and supporting evidence.

   All responses generated by this component are explicitly grounded in previously produced analytical artifacts and must reference the evidence, assumptions, uncertainties, and limitations already recorded by the system.

The component does not generate new analytical conclusions, introduce new causal hypotheses, resolve conflicts, interpret contracts, or modify the underlying evidentiary record. Its role is limited to making existing analysis accessible and understandable while preserving traceability, uncertainty, and the analytical boundaries established upstream.

How does the system detect and handle missing, conflicting, or unreliable data?

1. **Missing data** - the system detects missing data by inferring absence from within the evidentiary record rather than from external requirements or assumptions about completeness. As evidence is ingested, a time aware custody boundary is established that defines which artifacts are available at each point in the analysis. During structuring, the system extracts internal references, dependencies, and recurring production patterns such as cited attachments, referenced presentations, prior schedules, or expected periodic reports. When expected artifacts cannot be found within the custody boundary, their absence is explicitly recorded as a gap in the record.

   In analysis, missing data is interpreted as an analytical constraint rather than as an error or an implication of intent. Claims, timelines, and causal hypotheses that depend on absent artifacts are marked as indeterminate on the current record, and competing narratives remain unresolved when resolution would require access to missing information. The system actively surfaces expected but missing artifacts and associates them with the findings or relationships they constrain, allowing analysts to understand where the evidentiary record is incomplete and to decide through external processes whether obtaining additional material would meaningfully strengthen the analysis.

   When previously missing data later becomes available, it is treated as a new disclosure event rather than as a retroactive correction. The new material is ingested with its own custody, provenance, and timestamp, and previously recorded gaps are marked as resolved from that point forward. Analytical processes are reevaluated using the expanded record, and updated findings are produced while preserving earlier constrained results and the timing of disclosure. This approach allows the record to strengthen over time while maintaining transparency, auditability, and defensibility in adversarial settings.

2. **Conflicting data** - The system handles conflicting data by preserving competing assertions rather than attempting to normalize them into a single version of events. As evidence is ingested and structured, statements are captured as attributed claims with explicit sources, timing, and context. Event construction allows multiple candidate timelines or interpretations to coexist when the record supports incompatible accounts. This ensures that conflicts are identified only when the record contains logically incompatible assertions, timelines, or dependencies, rather than being resolved through implicit assumptions or source prioritization.

   During analysis, conflicting data is interpreted as a condition that constrains certainty. When claims, timelines, or causal explanations are incompatible, the system records the conflict explicitly and carries it forward into causal evaluation. Competing hypotheses are maintained side by side when resolution would require additional evidence or judgment beyond the record. No attempt is made to reconcile conflicts through narrative smoothing, inferred intent, or implied credibility. Instead, conflicts remain visible and analytically operative, preventing any party from gaining advantage through selective or self-consistent disclosure.

   In synthesized findings and explanations, conflicts are surfaced as limits on defensible conclusions. Outputs identify where multiple interpretations remain plausible, which evidence supports or contradicts each, and why the conflict cannot be resolved on the current record. If additional data later becomes available, the conflict can be reevaluated without erasing the historical existence of disagreement. This approach allows the system to operate credibly in adversarial settings by representing disagreement faithfully, preserving uncertainty, and ensuring that unresolved conflict remains explicit rather than silently collapsed.

3. **Unreliable data** - The system handles unreliable data by characterizing the conditions under which information was created and disclosed rather than by labeling evidence as true or false. As evidence enters custody, provenance, timing, authorship, and disclosure context are preserved. During structuring and analysis, these attributes allow the system to identify signals that affect reliability, such as whether a claim was created contemporaneously with the events it describes, whether it is retrospective, whether it is corroborated by independent sources, and whether it appears in isolation or alongside contradictory records.

   In analysis, unreliable data is interpreted as a qualifier on confidence rather than as a basis for exclusion. Claims or assertions that are weakly supported, poorly corroborated, or temporally distant from the events they describe are not discarded. Instead, they are carried forward with explicit reliability conditions that constrain how they can be used. Causal hypotheses and timelines that depend heavily on such data are marked as conditional or weakly supported, while hypotheses supported by contemporaneous and cross corroborated evidence are treated as more robust. This prevents unreliable inputs from silently dominating analysis while still preserving the full evidentiary record.

   In synthesized findings and explanations, reliability conditions are made explicit. Outputs identify where conclusions depend on evidence with limited support and where confidence is reduced due to lack of corroboration or delayed disclosure. The system does not infer motive, intent, or credibility of parties. It only reflects how the quality and context of the data affect what can be defensibly concluded. By treating unreliability as a measurable analytical condition rather than a judgment, the system remains neutral, transparent, and defensible in adversarial environments where self serving, incomplete, or retrospective records are expected.

How AI workflows (LLMs, embeddings, classification, summarization) support synthesis?

Upstream analysis relies first on representation and organization models that structure the evidentiary record without interpreting it. Embedding models are used to encode documents, communications, and extracted statements into a semantic space so that related material can be grouped even when different terminology is used. Clustering and similarity search algorithms operate on these embeddings to surface related topics, recurring themes, and latent connections across large volumes of data. These models support organization and retrieval and do not make judgments about correctness or credibility.

Information extraction and categorization are handled by bounded natural language processing models and rule-guided classifiers. These models identify entities, dates, assertions, references, and dependencies and assign them to predefined schemas such as event types, claim types, or artifact roles. Because the output space is constrained and schema-driven; these models are auditable and repeatable. Their function is to transform unstructured inputs into structured analytical primitives rather than to interpret meaning or resolve ambiguity.

Analytical reasoning is performed using explicit algorithmic and graph-based methods rather than generative models. Event graphs, dependency graphs, and rule-based consistency checks are used to construct timelines, detect conflicts, infer missing dependencies, and represent alternative causal hypotheses. Where uncertainty exists, it is preserved through conditional relationships and competing explanations rather than being statistically resolved. This stage produces inspectable analytical outputs that can be reevaluated as the record evolves.

Large language models are used only at the final synthesis stage as explanatory and navigational tools. They operate on the structured outputs produced upstream to summarize findings, explain assumptions, surface conflicts, and describe limitations in natural language. These models do not perform upstream analysis, generate new conclusions, or resolve disputes. Their role is to make the system’s analytical results accessible and understandable while preserving traceability, uncertainty, and evidentiary rigor.

How will data be modeled?

A high-level data model for this system is a property graph where the primary design goal is to represent the evidentiary record, extracted assertions, and analytical outputs as linked objects with explicit provenance. The graph is event-sourced and time-aware. It distinguishes between an underlying artifact, the claims extracted from it, and the analytical interpretations produced downstream. This allows competing narratives to coexist, allows gaps to be represented explicitly, and allows every conclusion to be traced back to specific evidence items.

<u>Core node types</u>

Evidence and provenance

- Artifact: file, email, attachment, image, drawing, schedule export, report
- Extract: a quoted span, table row, image region, or structured snippet taken from an Artifact
- ProvenanceRecord: custody metadata such as submitter, recipient, submission time, hash, and source system
- Version: alternate or later disclosed copies of an Artifact

Actors and commercial structure

- Person
- Organization
- RoleAssignment: person in role within organization, project role, contractual role
- Contract: prime, subcontract, change order, amendment
- CommercialTerm: notice requirements, milestones, payment terms, deliverables, scope clauses

Work structure and time

- ProjectItem: activity, work package, deliverable, RFI, submittal, drawing set, inspection
- Schedule: baseline schedule, update, lookahead
- CalendarEvent: meeting instance, site walk, milestone date

Assertions, events, and analysis

- Assertion: a source-attributed statement extracted from evidence
- Event: an event candidate in the project chronology
- Hypothesis: a causal or explanatory candidate that links events and conditions
- Finding: a synthesized statement grounded in evidence and conditioned on uncertainty
- Signal: non-decisional indicators such as reliability signals, completeness flags, and contradiction markers

Key relationship types

Provenance and grounding

- `Artifact HAS_PROVENANCE ProvenanceRecord`
- `Extract DERIVED_FROM Artifact`
- `Assertion SUPPORTED_BY Extract`
- `Finding GROUNDED_IN Assertion` and `Finding GROUNDED_IN Extract`

Actors and participation

- `Person AFFILIATED_WITH Organization`
- `Person PARTICIPATED_IN Communication` and `Person ATTENDED CalendarEvent`
- `Organization PARTY_TO Contract` and `Contract GOVERNS ProjectItem`

Project structure and time

- `ProjectItem DEPENDS_ON ProjectItem`
- `Schedule INCLUDES ProjectItem` and `ProjectItem OCCURS_AT SiteLocation`
- `Event RELATED_TO ProjectItem` and `Event ON_TIMELINE Timeline`

Conflict, completeness, and reliability

- `Assertion CONFLICTS_WITH Assertion`
- `Signal APPLIES_TO Assertion` and `Signal APPLIES_TO Event`
- `ExpectedArtifact EXPECTS ArtifactType` and `ExpectedArtifact IMPLIED_BY Assertion`
- `Artifact SATISFIES ExpectedArtifact`

Causal reasoning and synthesis

- `Hypothesis EXPLAINS Event`
- `Hypothesis REQUIRES Assertion` and `Hypothesis CONSTRAINED_BY Signal`
- `Finding DERIVED_FROM Hypothesis` and `Finding QUALIFIED_BY Signal`

<u>Properties that make the graph defensible</u>

Every node and edge carries properties that preserve auditability and adversarial integrity, such as:

- Time fields that separate event time, document time, and disclosure time
- Attribution fields that preserve who asserted what and in what context
- Confidence and status fields for assertions, events, hypotheses, and findings
- Traceability fields that link analytical outputs to exact extracts and source artifacts
- Immutability conventions where raw artifacts are not overwritten and later disclosures become new versions linked to prior gaps

This model makes the graph itself the analytical substrate. A connected record of evidence, claims, competing interpretations, and the constraints that shape what can be concluded.

How would the system prevent or mitigate AI hallucinations in a legal/claims context?

The system prevents and mitigates AI hallucinations by structurally separating analysis from explanation and by ensuring that generative models never operate on raw evidence or incomplete context. Large language models are not used to infer facts, resolve disputes, or generate conclusions. All analytical outputs are produced upstream using bounded models and explicit logic over structured data. By the time an LLM is invoked, the analytical state is already fixed, constrained, and traceable.

Hallucination risk is further reduced by forcing grounding and traceability at the interface level. Every LLM response is generated only from pre existing analytical artifacts such as findings, hypotheses, conflicts, signals, and linked evidence extracts. The model is not allowed to introduce new entities, events, timelines, or causal relationships. Outputs must reference the underlying graph objects they describe, and responses that cannot be grounded in existing artifacts are either refused or explicitly framed as indeterminate. This prevents the model from filling gaps with plausible-sounding but unsupported content.

The system also mitigates hallucinations by preserving uncertainty and refusing narrative completion. Missing data, conflicting assertions, and weak reliability signals are surfaced explicitly and carried through to the explanation layer. The LLM is required to reflect these constraints verbatim rather than smoothing them into a coherent story. Where the record does not support a conclusion, the system responds by explaining why resolution is not possible on the current evidence. This ensures that the model never substitutes linguistic fluency for evidentiary support.

Finally, the system maintains auditability and guardrails at runtime. Prompts, grounding references, and generated outputs are logged and inspectable. Pattern based checks can flag responses that exceed the allowed scope, such as introducing new conclusions or asserting certainty where none exists. In a legal or claims setting, this combination of architectural separation, grounding enforcement, uncertainty preservation, and auditability ensures that AI assistance enhances clarity without introducing hallucinated facts or unsupported judgments.

What strategy does the system provide to respectfully challenge stakeholder expectations when analytical findings differ?

The system provides a strategy for respectfully challenging stakeholder expectations by reframing disagreement as an examination of evidence, assumptions, and analytical constraints rather than as a dispute over conclusions. Findings are presented as conditioned statements that make explicit the supporting and contradicting evidence, unresolved conflicts, and gaps that limit confidence, emphasizing that outputs reflect what can be defensibly concluded from the current analytical state. When expectations differ, the system surfaces the specific assumptions on which those expectations rely, such as interpretations not supported by contemporaneous artifacts, reliance on referenced but undisclosed information, or the need to resolve conflicts the system has deliberately preserved. By making these dependencies visible, the challenge becomes an analytical inquiry into which assumptions should be revisited or what additional evidence would strengthen the record, enabling constructive engagement without personalizing disagreement or undermining stakeholder credibility.

Given limited time and resources, what would be an MVP, and what would be its 30/60/90 day roadmap?

The MVP should focus on making the evidentiary record inspectable and analytically constrained, not on full automation or exhaustive intelligence. The minimum viable system proves three things:

1. Evidence can be ingested with immutable custody and provenance.
2. Claims, events, and gaps can be represented explicitly in a graph, and
3. Synthesis can surface what can and cannot be concluded on the current record.

At the MVP stage, the system does not need full scale causality modeling, advanced productivity analytics, or broad system integrations. Its value comes from turning unstructured, adversarial material into a navigable, traceable analytical graph that preserves missingness, conflict, and uncertainty rather than resolving them.

<u>First 30 days: establish the evidentiary spine</u>

Primary goal: create a trustworthy record of what exists and what is asserted.

- Implement evidence intake with immutable storage, hashing, timestamps, and basic provenance metadata.
- Stand up the core property graph with a small set of node types: Artifact, Extract, Assertion, Person, Organization, and Event.
- Use bounded NLP models to extract entities, dates, and attributed assertions from emails, meeting minutes, and reports.
- Create basic internal reference detection to surface expected but missing artifacts.
- Deliver a simple graph exploration interface and a minimal LLM-based explanation layer that can answer questions like “What evidence supports this claim?”

30-day outcome: users can see who said what, when, based on which documents, and where the record is incomplete.

<u>60 days: add analytical constraints and conflict visibility</u>

Primary goal: make disagreement and uncertainty explicit and operational.

- Extend the graph to represent competing events, alternative timelines, and explicit conflict relationships between assertions.
- Add rule-based detection for internal inconsistencies in timing, sequencing, and dependencies.
- Introduce non-decisional reliability signals based on provenance, timing, and corroboration.
- Allow analysts to associate claims and events with provisional hypotheses without resolving them.
- Enhance synthesis to surface conflicts, gaps, and conditional findings clearly.

60-day outcome: users can see why disputes exist, what evidence supports each side, and why resolution is constrained on the current record.

<u>90 days: strengthen synthesis and iteration</u>

Primary goal: support iterative analysis as the record evolves.

- Support versioning and late disclosure so previously missing artifacts can be ingested and linked to prior gaps.
- Enable reevaluation of findings when new data becomes available while preserving earlier analytical states.
- Improve summarization and explanation so users can ask scenario questions such as “What would change if this document were added?”
- Add basic reporting and export suitable for review, audit, or dispute preparation.

90-day outcome: a defensible analytical system that can evolve with disclosure, withstand adversarial scrutiny, and demonstrate clear value without overreach.

What would be the minimal team required to fulfill the 30/60/90 day roadmap?

Technical lead or principal engineer
Owns the system architecture, data model, and analytical boundaries. Ensures the graph-based model, custody rules, and separation between analysis and explanation are enforced consistently. Makes tradeoffs to keep the MVP disciplined and defensible.

Backend engineer
Implements evidence intake, immutable storage, provenance tracking, and the core property graph. Builds APIs for ingestion, querying, and versioning. Focuses on correctness, traceability, and performance over polish.

Data engineer or applied ML engineer
Implements embedding generation, extraction pipelines, and bounded classification models. Responsible for entity extraction, assertion extraction, reference detection, and basic similarity search. Ensures models are schema-driven and auditable.

Frontend engineer
Builds a minimal but usable interface for graph navigation, evidence inspection, and analytical outputs. Focuses on making relationships, conflicts, and missing data visible rather than on rich visualization.

Forensic analyst or domain expert
Guides how claims, events, conflicts, and missing data should be represented to align with real adversarial analysis. Validates that system outputs make sense to practitioners and do not cross into interpretation or judgment.

DevOps or platform engineer

Part-time support for secure storage, access controls, and deployment. Especially important if handling sensitive or litigation-grade material.

What delivery and execution risks does the roadmap present, and how could they be mitigated?

Risk 1: Under-resourcing against evidentiary complexity

Description:
With a tight 90-day roadmap and a lean team, there is a risk that variability and messiness in real-world evidence ingestion will consume disproportionate effort. Emails, attachments, schedules, and reports often contain edge cases that are difficult to normalize. If ingestion and structuring are owned by too few engineers, progress may slow or quality may degrade in areas critical to credibility.

Mitigation:
Introduce early redundancy in backend and data engineering responsibilities. Allocate at least two engineers to evidence ingestion, graph modeling, and reprocessing workflows. Prioritize correctness, provenance, and traceability over breadth of supported formats. Defer non-essential integrations to avoid overloading the ingestion pipeline during the MVP phase.

Risk 2: Loss of analytical credibility due to insufficient domain validation

Description:
Adversarial and legal analysis requires careful alignment with real-world practice. If forensic or domain expertise is treated as a late-stage review rather than a continuous input, the system may encode assumptions or abstractions that appear technically sound but fail under practitioner scrutiny.

Mitigation:
Ensure sustained involvement of a forensic analyst or domain expert throughout the roadmap, beginning in the first 30 days. Use this role to validate data models, assertion handling, conflict representation, and missing data surfacing on an ongoing basis. Treat domain feedback as a design constraint rather than post hoc validation.

Risk 3: Scope creep toward advanced analytics or AI-driven conclusions

Description:
Stakeholder pressure may push the team to demonstrate advanced AI capabilities prematurely, such as automated causality or decision-making. This risks diluting effort and undermining the system’s defensibility, especially given limited staffing.

Mitigation:
Explicitly constrain MVP scope to evidentiary structuring, conflict visibility, and defensible synthesis. Document which analytical capabilities are intentionally out of scope for the first 90 days. Reinforce architectural separation between upstream analysis and downstream explanation to prevent generative models from expanding beyond their role.

Risk 4: Team burnout and single point of failure

Description:

A small team operating at high intensity for 90 days risks burnout and knowledge concentration. Loss of a single contributor or sustained overload can materially impact delivery.

Mitigation:
Distribute ownership across critical subsystems and avoid single-owner designs. Favor simple, inspectable implementations over highly optimized or clever solutions. Maintain shared documentation and regular cross-review to reduce individual dependency.