Beyond Signature: Rethinking the Contract Lifecycle

Samuel Bourque
Amicus Docs (`https://www.amicusdocs.com`)

v1.26 — December 30, 2025

Terminology Note. Throughout this paper, certain terms are bolded to indicate that they are used in a precise, technical sense within the *Amicus Docs* model. Where a term is bolded, it should be understood as a defined concept rather than a loose description.

Executive Summary

Most contract software is optimized for getting a document signed. It performs well at execution, but is only weakly aligned with a contract's ultimate purpose: enabling enforcement and keeping parties accountable to their obligations. This white paper articulates a system that aligns the act of contracting with the practical need to preserve relevant evidence and context.

Amicus Docs captures the full contractual record in a single, intuitive structure while applying state-of-the-art encryption standards for authentication, integrity proofs, and timestamping. The result is a contract system designed not merely for execution, but for reliability, accountability, and enforceability throughout the life of the agreement.

1 Problem Statement

Contemporary contract software optimizes for execution, but execution alone is not enough for enforcement. These systems treat the signed document as the contract, even though courts and institutions decide disputes based on the full relationship between the parties. This section explains that mismatch and why it matters operationally.

1.1 Execution-Centric Contract Software

Most contract software treats an agreement as a static artifact. Once a document is signed, it is reduced to a finalized file—typically a PDF with embedded signatures—and effectively frozen in time. While this captures the moment of execution, it reflects only a narrow slice of the contract's legal and practical reality.

1.2 Enforceability Depends on the Whole Relationship

The ultimate purpose of a contract is enforceability. Courts do not interpret agreements based solely on the signed document, but on the totality of the relationship: negotiation history, subsequent conduct, amendments, performance, communications, and the sequence of events that give meaning to the written terms. These elements determine intent, compliance, breach, waiver, and remedy.

1.3 Signature-Only Tools Exclude Critical Evidence

Signature-only tools exclude this broader evidentiary context. By isolating the point of execution from everything that precedes and follows it, they fail to account for the information a court would later need to assess how the contract was understood, how it evolved, and how it was carried out. The result is a fragmented record in which legally relevant facts are dispersed across unrelated systems, personal inboxes, and informal memory rather than managed as part of a coherent contractual lifecycle.

1.4 Retroactive Reconstruction Under Pressure

In practice, this gap forces organizations to reconstruct contractual history retroactively—often under dispute—rather than manage it continuously. This disconnect between how contracts are enforced and how contract software operates represents a foundational limitation of signature-centric tools.

2 Thesis

To address these problems, *Amicus Docs* starts from a different premise: contracts should be modeled as living records rather than static artifacts. The core thesis is that a contract system must capture the ongoing conversation, structure, and evidence surrounding an agreement, not just its final signed document. This section outlines that model and the assumptions behind it.

2.1 Contracts as Ongoing Conversations

A contract is not merely a document; it is an ongoing conversation between parties. Through that conversation—before, during, and after execution—participants reveal the information necessary to evidence intent, rights, obligations, performance, and change.

If enforcement depends on the totality of this record, then contract systems must be designed to capture the conversation itself, not just its formal output. The signed document remains essential, but it is only one artifact within a broader evidentiary context.

2.2 The Conversational Interface as the Natural Frame

The most natural interface for this reality is a simple, structured conversation. A chat-based model is intuitive, familiar, and well-suited to recording chronological intent and action. Within this conversational framework, formal contract documents can be drafted, reviewed, attached, and signed, while remaining embedded in the broader context from which they arise.

2.3 A Unified Casefile for All Relevant Materials

In addition to signable documents, other relevant materials—images, audio, video, hyperlinks, and external references—can be attached directly to the same record. Together, these elements form a single integrated **Casefile**: a unified container that preserves the full contractual history as it unfolds.

2.4 From Static Artifacts to Living Records

This approach treats contracts as living records rather than static artifacts, aligning contract software with how agreements are formed, performed, and ultimately enforced in the real world.

3 Vision and Principles

The core purpose of *Amicus* is to make contractual relationships workable, not merely documented. Two operational principles guide the system design—clarity and integrity—resting on a deeper foundation: respect for human dignity and autonomy.

3.1 Principle 1: Clarity

Most avoidable disagreements in commerce are largely not about deliberate bad faith; they more often arise from unclear expectations. Parties believe they agreed on the same thing, only to discover in hindsight that they were working from different assumptions about scope, timing, authority, or acceptable outcomes.

Traditional contract tooling can contribute to this problem. Negotiation often happens in email threads, chat logs, and draft versions dispersed across systems. The signed document captures a moment in time, but not the context that gives its terms meaning. When something goes wrong, participants are frequently forced to reconstruct what was meant, what changed, and why.

Amicus treats clarity as a first-class requirement. Each agreement lives inside a single **Casefile** that preserves the conversation, the documents, the role assignments, and the sequence of decisions that led to signature and beyond. Terms, Specs, and amendments are expressed in structures that make responsibilities visible rather than implicit. The goal is simple: at any later point, a reasonable person should be able to see what the parties intended and how that intention evolved.

3.2 Principle 2: Integrity

Clarity is not enough if the record cannot be trusted. Contracts must withstand time, turnover, and pressure. When stakes are high, any ambiguity or weakness in the evidentiary trail becomes a point of leverage.

Amicus is designed so that the contractual record has integrity by construction. Actions are authenticated to the people who took them. Changes are recorded rather than overwritten, with explicit versioning and traceable authority. The system uses strong encryption and tamper-evident timelines so that later participants—including auditors, courts, or successors—can distinguish what actually happened from what is merely asserted.

This focus on integrity extends beyond cryptography. It is reflected in how the product constrains silent changes, requires explicit roles, and ties key decisions to concrete events in the **Casefile** history. The system is engineered so that it is easier to behave honorably than to manipulate the record.

3.3 Principle 3: Process Agnosticism and Factual Invariants

Amicus does not prescribe how parties negotiate, amend, or perform under an agreement. Multiple valid processes may exist for any contractual relationship, and Amicus remains agnostic among them.

Instead, Amicus enforces a limited set of factual invariants required for contractual facts to be well-defined, attributable, and inspectable. These invariants are not opinions about preferred workflows; they are conditions under which the system can reliably assert facts such as whether an agreement is in effect, who holds authority, or what has occurred in time. Interpretation and judgment are left entirely to people and institutions.

3.4 Principle 0: Dignity and Autonomy

Beneath clarity and integrity lies a prior commitment: contracts are expressions of human agency and autonomy. They record how people choose to bind themselves, allocate risk, and cooperate over time. Any system that manages contracts must therefore respect the dignity and decision-making authority of the people who use it.

Amicus is named for the Latin word for friend. A friend may advise, remind, and assist, but does not replace the person's will. In the same way, *Amicus* never supplants the parties' legitimate intention. It does not presume to speak as a principal, does not invent authority, and does not act on its own account. It serves only within the bounds of what the parties themselves establish.

AI is an important component of *Amicus*, but only in an assistant and tool capacity. It can help draft language, surface context, and highlight inconsistencies. It cannot and does not assume the role of a principal, and it does not exceed the authority granted to it by human users and by the agreements in which it participates. Accountability and final decision-making remain with people.

3.5 Principles in combination

When clarity, integrity, and dignity are applied together, they tend to change the incentives around contracting. Human intention remains primary. Clear, shared expectations reduce the surface area for misunderstanding. A reliable, transparent record makes it significantly more costly to misrepresent what was agreed or how it was performed, while the dignity constraint prevents the system from overreaching into decisions that belong to people.

In this environment, honorable conduct becomes the most stable strategy rather than a hopeful assumption. Parties can move faster because they trust that the system will keep their shared intention fit and healthy over time, without displacing their autonomy. Disagreements still arise, but they are more likely to be resolved against a stable, coherent record instead of vague recollections and fragmented evidence.

These principles shape the model that follows: a lifecycle-native system that treats contracts as living records rather than static artifacts.

4 Model

The *Amicus* model treats every contract as a lifecycle-native **Casefile**: a structured record that captures intent, negotiation, authority, execution, amendment, performance, and supporting evidence in one coherent frame. Instead of anchoring the system on static documents or isolated signatures, *Amicus* models the full arc of contractual activity as it actually unfolds between parties. This approach preserves the familiar elements of contract practice while making them explicitly machine-readable, auditable, and aligned with how agreements are enforced in the real world. The **Casefile** is the authoritative container, while individual records are artifacts created within it or derived from it.

4.1 Layers of Legal Reality and Records

4.1.1 Agreement and Contract

In this white paper, **Agreement** and **Contract** are used largely interchangeably to describe the **Parties**' legal relationship. Jurisdictions draw technical distinctions between these terms, but *Amicus* does not attempt to interpret or enforce those distinctions.

We use **Agreement** as a neutral label for the outcome of a completed signature process: the relationship that the **Parties** understand themselves to have entered into. Whether that **Agreement** satisfies all legal requirements to be treated as a **Contract** in a given jurisdiction is a question for the **Parties** and, where necessary, for courts or other decision-makers—not for the software.

4.1.2 Agreement and Document

Amicus distinguishes between the legal relationship and the artifacts that evidence it. An **Agreement** is the underlying relationship between the **Parties**; a **Document** is a text artifact that may record or support that relationship.

Within a **Casefile**, **Documents**—drafts, notes, instantiated **Templates**, and executed **Signable documents**—are treated as evidence of the **Agreement** over time. A single **Agreement** may be reflected across multiple **Documents** and other records (such as messages or performance events), and a given **Document** may contribute evidence to more than one stage in the same contractual relationship.

4.1.3 Template, Proposal, and Signable document

A **Template** in *Amicus* is a reusable contract pattern. It defines structure—**Roles**, **Terms**, and **Specs**—without yet binding specific **Parties** or belonging to any particular **Casefile**.

When a **Template** is instantiated as a **Proposal** into a **Casefile**—that is, when specific **Parties** are mapped to defined **Roles** and the **Template**'s structure is applied to a concrete

situation—*Amicus* treats that instance as a **Proposal**. The **Proposal** represents what the **Parties** are currently considering or negotiating for that relationship.

From a **Proposal**, *Amicus* produces a **Signable document**: the specific **Document** instance that enters the signature workflow. Once that **Signable document** is fully executed by the necessary **Parties**, *Amicus* records the resulting **Agreement** within the **Casefile** and its associated history, without asserting whether that **Agreement** qualifies as a **Contract** under any particular legal test.

4.2 Document-Centric vs Lifecycle-Centric Contracting

Legal Reality	**Signature-Only Software**	**Amicus Model**
Contract formation	Negotiation happens off-platform	Negotiation captured as conversation
Agreement	Signature marks completion	Signature marks a lifecycle milestone
Record	Static PDF	Continuous **Casefile** record
Context	Fragmented across tools	Unified within a single **Casefile**
Amendments	New documents, disconnected	In-context amendments
Evidence	Reconstructed after the fact	Preserved as it arises
Authority	Implicit, external	Role-based, explicit
Timeline	Implicit, unclear	Explicit, chronological
Enforcement	Manual reconstruction	Evidentiary record ready
System posture	Execution-centric	Enforcement-aligned

Table 1: Comparison between signature-only contract software and the Amicus lifecycle-native model, aligned with legal and evidentiary reality.

This mapping illustrates the core shift: from treating contracts as finalized documents to managing them as evolving legal records.

4.3 Cryptographic Integrity of Signed Records

A lifecycle-native contract system must do more than store documents. It must preserve a record that later participants can trust, even if they do not control the original system. In practice, this means capturing the text of the signable document, the signatures, and the key timestamps in a form that is both human-readable and independently verifiable.

4.3.1 Structured Signed Record

When a contract is fully signed in Amicus, the system constructs a single structured record describing the executed agreement at the document level. This record corresponds to the specific **Signable document** instance that entered the signature workflow and was completed. It is represented as a JSON document that includes the final contractual text, the

identity of the signing **Parties**, and the relevant signing and effectiveness timestamps. Together, these elements capture the state of that **Signable document** at the moment of execution.

4.3.2 Hashing and Tamper-Evidence

From this JSON record, Amicus computes a SHA256 hash. This hash functions as a cryptographic fingerprint: any change to the underlying JSON—however small—would produce a different hash. The hash therefore serves as a compact, tamper-evident representation of the complete signed record.

4.3.3 PDF with Audit Trail

Amicus then produces a human-readable PDF of the contract for ordinary use. This PDF reflects the same signable document instance and includes an audit trail page that summarizes the key signature events and displays the SHA256 hash of the JSON record. The JSON file and the corresponding PDF are distributed together to the parties, for example by attaching both artifacts to a confirmation email.

4.3.4 Independent Verification

Because every party receives both the structured record and the human-readable document, they are not dependent on Amicus to establish what was signed. Anyone in possession of the JSON file can compute its SHA256 hash using standard tools and confirm that it matches the hash printed in the PDF's audit trail. This linkage makes it substantially harder for a party to disavow the contents or timing of the agreement without contradicting the cryptographic evidence embedded in the record they already hold.

4.4 Casefile-Level Timeline Exports

In addition to document-level records for individual **Signable documents**, Amicus supports **Casefile**-level exports that capture the full contractual history as a single, ordered timeline. At any point, a user may trigger an export that renders the entire **Casefile**—messages, attachments, amendments, role changes, and key system events—into a structured representation suitable for archival and external verification.

These exports are timestamped and sealed using methods similar to those described above for signed documents. The resulting artifact includes a chronological log of **Casefile** activity and a corresponding cryptographic summary, so that later reviewers can verify both the contents of the export and the relative ordering of events.

Unlike the **Signable document** JSON record, which is generated automatically at the moment of execution and focuses on a single agreement instance, the **Casefile**-level export

is manual and encompasses the entire lifecycle of the relationship as recorded in *Amicus*. This allows parties to anchor not just what they signed, but how the surrounding context evolved over time.

5 Value Proposition

Amicus transforms contracting from a static, **Document**-centered task into a coherent, lifecycle-native process. By aligning software structure with legal reality, it delivers value across drafting, execution, performance, and enforcement—without increasing complexity for users.

5.1 Drafting, Negotiation, and Amendment

Amicus provides an integrated **Assistant** for drafting and negotiation within each **Casefile**, enabling **Parties** to move from reusable **Templates** to concrete **Proposals** and, where appropriate, to executed **Agreements**. Contracts are composed using structured **Terms**, **Role Assignments**, and **Specs**, allowing clarity without rigidity.

Within a **Casefile**, a **Proposal** is instantiated as a specific **Signable document** when it enters the signature workflow. Proposals, counterproposals, and amendments occur in context, preserving continuity and attribution for each **Signable document** instance. **Templates** can be created, shared, and searched, accelerating reuse while maintaining consistency across **Agreements**.

5.2 Unified Casefile and Evidentiary Record

All contractual activity is encapsulated within a single **Casefile**. This unified structure enables more than storage: it allows review, suggestion, and contextual reasoning by the **Assistant**, while preserving a complete chronological record.

Supporting materials—documents, attachments, links, voice recordings, images, and dialogue—can be added as they arise. The result is a living **Casefile** that reflects the true operational and evidentiary history of the **Agreement**.

5.3 Public Offers and Marketplace Readiness

Because contracts are structured and lifecycle-aware, the *Amicus* model is designed to enable the preparation of ready-to-execute **Public offers**. Each **Public offer** is based on an underlying **Template** and expressed as a **Proposal** that can be instantiated as individual **Signable documents** when counterparties accept. These ready-made **Proposals** can be posted, discovered, and entered into via a **Marketplace**, extending contracting beyond one-off private negotiations to repeatable, standardized offerings while preserving a clear chain from **Template** to **Proposal** to executed **Agreement**.

Together, these capabilities reduce friction in forming agreements, increase clarity during performance, and strengthen reliability when accountability or enforcement is required.

Taken alone, this lifecycle-native record already improves how parties form, manage, and enforce agreements. But real contracting also depends on who counterparties are, how they have behaved in prior relationships, and what is known about them beyond any single **Document**. To address this relational layer without turning it into a scoring system, *Amicus* introduces a constrained social surface.

6 Operational Intelligence

Because *Amicus* treats each contract as a lifecycle-native **Casefile**, it can support forms of automation and assistance that are contextually aware, auditable where appropriate, and explicitly bounded. These capabilities are designed to support clarity, continuity, and accountability without displacing human judgment or authority.

6.1 Contextual Assistants

Amicus supports AI assistance in multiple, deliberately distinct modes, each aligned with a different evidentiary posture.

6.1.1 On-the-Record Assistant

Within a **Casefile**, an on-the-record **Assistant** may participate in the shared conversational space. This **Assistant**:

- is contextually aware of the full **Casefile** history,
- answers questions or surfaces relevant context in the **Casefile** chat, and
- produces outputs that are visible to all **Parties** and preserved as part of the record.

This mode is appropriate where shared understanding, transparency, or mutual clarification is desired. Assistant contributions are attributable, timestamped, and retained alongside other **Casefile** events.

6.1.2 Off-the-Record Assistant

Amicus also provides an off-the-record **Assistant** for private consultation. Conversations in this mode:

- are contextually aware, drawing on the **Casefile** for reference,
- are visible only to the initiating user, and
- are not recorded, stored, or preserved in the **Casefile** or elsewhere.

Off-the-Record sessions are ephemeral and disappear when the browser session ends. This mode supports private reasoning, drafting, or exploration without creating evidentiary artifacts or affecting the shared contractual record.

6.1.3 Template Drafting Assistant

Separately, *Amicus* includes a drafting assistant dedicated to **Templates**. This assistant helps users author reusable contract structures using standard, interoperable components

such as **Roles**, **Terms**, and **Specs**.

By encouraging structured composition rather than free-form drafting, this assistant supports consistency, reuse, and downstream automation without constraining legal expression.

6.2 Contract Status Awareness

Each **Casefile** maintains an explicit lifecycle status, such as:

- negotiating,
- in effect,
- amended,
- suspended, or
- concluded.

Status is derived from concrete events—execution, amendment, termination—rather than inferred heuristics. This explicit state enables both human clarity and system-level support, including appropriate prompts, reminders, and reporting.

6.3 Event-Driven Notifications and Scheduled Reminders

Amicus supports notifications and reminders grounded in contractual structure and time.

Event-driven notifications may be triggered by actions such as **Proposals**, signatures, amendments, role changes, or status transitions.

Scheduled reminders may be generated based on **Terms** and timelines, including notice periods, renewal windows, milestones, or periodic obligations.

Because these signals are tied to explicit **Casefile** events and structures, they remain attributable, contextual, and auditable.

6.4 Casefile Reporting and Readiness

At any point, **Parties** may request a synthesized view of a **Casefile**. The **Assistant** can generate reports summarizing:

- current status,
- outstanding or completed obligations,
- recent activity,
- upcoming time-based events, and
- relevant amendments or decisions.

These reports support oversight, handover, and continuity without requiring manual reconstruction.

6.5 Inputs Beyond Text and Documents

The **Casefile** supports multiple forms of input beyond traditional **Documents**.

File attachments may be added directly to a **Casefile**. Supported formats such as images and PDFs can be viewed in place, preserving proximity to the surrounding context.

Recordings may be captured directly through the interface. Audio recordings can be reviewed alongside an automatically generated transcript, which may be corrected or revised before being retained as part of the **Casefile** history.

These inputs allow **Parties** to preserve relevant evidence and context in the form most natural to the situation, without forcing premature formalization.

6.6 Boundaries and Posture

Automation and assistance in *Amicus* are intentionally constrained. The system may notify, remind, summarize, draft, and surface context, but it does not:

- act as a **Party** or **Principal**,
- make binding decisions,
- initiate obligations on its own authority, or
- substitute automated judgment for human intent.

This posture preserves accountability and autonomy while enabling a higher standard of clarity, continuity, and evidentiary integrity throughout the contract lifecycle.

7 Social Layer and Relational Context

Contracting does not occur in isolation. **Agreements** are formed between people and organizations with prior relationships, shared history, and reputational context that exists outside any single **Document**. Traditional contract systems largely ignore this reality, treating counterparties as interchangeable identifiers rather than persistent actors with traceable interactions.

Amicus introduces a restrained social layer designed to preserve relational context without turning reputation into a quantified or system-imposed judgment.

7.1 Contact Book and Relationship Graph

Amicus provides each user with a persistent contact book: a structured record of people and organizations they have interacted with over time. Contacts may be added through direct requests or established implicitly through **Agreement**-level participation.

This contact book functions as a practical working network rather than a social feed. It exists to support continuity, recall, and accountability across repeated interactions, not to encourage visibility or engagement for its own sake.

7.2 Contact Requests and Controlled Visibility

Users may request to add one another as contacts. Acceptance is required before private contact information or expanded relational context is shared. This preserves agency and prevents unsolicited exposure while allowing networks to form organically through real interactions.

Contact status affects visibility and interaction boundaries but does not alter contractual rights or core system behavior.

7.3 Public Sharing of Templates and Structures

Amicus is designed to allow users to publish contract **Templates** and structural patterns for public reuse. These may be shared openly, independent of contact status, enabling practitioners and organizations to contribute useful starting points to the broader community.

Shared **Templates** are discoverable through free-text and tag-based search rather than editorial curation. This ensures that visibility is driven by relevance and user judgment rather than platform preference.

7.4 Free-Text, Tag-Based Discovery

Search across people, **Templates**, and publicly shared materials is based on simple free-text queries and user-defined tags. The system does not impose hierarchical taxonomies, rankings, or algorithmic promotion.

This design favors interpretability over optimization and allows users to form their own conclusions based on raw, inspectable data.

7.5 Verified Counterparty Reviews

Any counterparty to a completed or ongoing **Agreement** may leave a written review on the other party's profile. These reviews are "verified" in the narrow sense that they are cryptographically and contextually linked to an actual **Agreement** recorded in the system.

Amicus does not score, summarize, or evaluate these reviews. They are preserved as first-person statements, attributed to real counterparties, and anchored to real contractual relationships.

7.6 Character Letters from Contacts

In addition to counterparty reviews, contacts may submit written character letters to a user's profile. These letters are explicitly subjective and are distinguished from contract-linked reviews.

Character letters are voluntary, attributable, and unstructured. They exist to provide qualitative context rather than evaluation.

7.7 Qualitative Reputation Without Scoring

Taken together, counterparty reviews and character letters form a body of relational evidence. This evidence functions as a form of social credit without becoming a credit score.

Amicus deliberately avoids:

- numerical ratings,
- aggregate scores,
- sentiment analysis or summarization, and
- algorithmic judgments about trustworthiness.

The system records and verifies context but does not interpret it. Meaning is left to human readers, who may weigh statements differently depending on circumstance, relevance, and judgment.

7.8 System Neutrality

The social layer is designed to support informed decision-making without substituting platform opinion for human judgment. *Amicus* does not tell users whom to trust, whom to avoid, or how to interpret another party's history.

Instead, it preserves durable, attributable, and inspectable social context alongside contractual records. This aligns with the platform's broader posture: conservative, evidence-oriented, and enforcement-aware rather than promotional or performative.

Relational history and qualitative reputation, however, address only one side of trust; enforceable contracts also require confidence about who the **Parties** actually are.

8 Identity Verification and Trust Signaling

Enforcement presupposes identity. A **Contract** can only be meaningfully enforced if the **Parties** to it are identifiable with sufficient reliability for notice, attribution, and accountability. While *Amicus* preserves contractual evidence and context, it does not assume that all **Parties** will require the same level of identity assurance in every **Agreement**.

Accordingly, *Amicus* treats identity verification as contextual, optional, and tiered rather than universal or mandatory.

8.1 Verified Identity as a Contractual Condition

Some counterparties may choose to contract only with **Parties** whose identity has been verified to a particular standard. Others may accept lower levels of assurance for informal or low-risk **Agreements**. *Amicus* supports this variation by allowing identity verification status to be surfaced as part of a **Party**'s profile and considered during contracting.

Verification status may be used as:

- a prerequisite to entering certain **Agreements**,
- a condition specified within **Terms**, or
- a signaling mechanism for counterparties assessing risk.

The decision to require verification rests entirely with the contracting **Parties**.

8.2 Tiered Verification

Identity verification is not treated as binary. Different contexts require different levels of assurance. *Amicus* supports the concept of multiple verification tiers, ranging from basic identity confirmation to stronger forms of attestation.

Rather than performing verification itself, *Amicus* integrates with external identity and verification providers. This avoids duplicating regulated compliance functions while allowing verification results to be recorded, referenced, and relied upon within the contractual record.

8.3 Neutral Trust Signaling

Verified status is presented as a factual attribute, not an endorsement. A visual indicator—analogous to a verification mark—signals that a **Party** has completed a specified verification process through an external provider.

Amicus does not:

- assert character, reliability, or competence,

- rank verification levels as "better" or "worse", or
- substitute verification for contractual diligence.

Verification indicates identity assurance only. It does not replace judgment, negotiation, or professional review.

8.4 Integration with Social and Contractual Context

Identity verification complements, but does not replace, the social and relational context preserved elsewhere in the system. Verified identity, counterparty reviews, and character letters address different dimensions of trust:

- verification establishes who a **Party** is,
- contractual history shows how a **Party** has acted, and
- relational statements provide qualitative human perspective.

Together, these elements provide raw material for informed decision-making without imposing platform opinion.

8.5 Enforcement Alignment

By anchoring **Agreements** to identifiable **Parties** and preserving verification evidence alongside the contractual record, *Amicus* strengthens downstream enforceability without expanding its role into compliance or adjudication. The system enables **Parties** to signal, require, and rely upon identity assurance while remaining neutral as to how that assurance is used.

This approach reflects *Amicus*'s broader design philosophy: provide durable structure and verifiable facts, while leaving interpretation, judgment, and risk tolerance to the **Parties** themselves.

The same philosophy applies to how *Amicus* sustains itself as a business. Revenue models are designed to grow with the depth, durability, and institutional usefulness of contractual activity, rather than with superficial engagement or raw document volume.

9 Monetization

Amicus is designed to monetize in ways that align with its lifecycle-native model. Revenue grows with the depth and durability of contract activity—number and complexity of **Casefiles**, extent of AI assistance, and downstream institutional use—rather than with raw **Document** count alone. The structure of the system enables multiple pricing paths without locking in a single model.

9.1 Subscription Tiers

Amicus offers a default *Casual* tier, suitable for lightweight or infrequent use. A paid *Proprietor* tier provides increased capacity, higher limits, and expanded AI inference budgets for users who manage contracts as part of their ongoing operations. Because value is delivered at the **Casefile** level, subscription tiers scale naturally with the number and intensity of active contractual relationships.

9.2 Marketplace Fees

Contracts prepared as **Public offers** may be posted and discovered through a **Marketplace**. *Amicus* is structured to generate revenue through transaction-based fees associated with successful contract formation or **Marketplace** participation, aligning platform incentives with completed **Agreements**. The lifecycle-native record makes it straightforward to define when a transaction is complete and which **Parties** were involved.

9.3 White-Label Contract Management

Organizations may license *Amicus* as a white-label contract management system, embedding lifecycle-native contracting into their own products, services, or operational workflows. Because *Amicus* treats contracts as **Casefiles** with complete histories, white-label deployments can expose tailored slices of the record while preserving a consistent evidentiary core. This channel supports enterprise, platform, and institutional use cases without requiring end-user brand adoption.

9.4 Professional Referrals

By preserving structured, high-quality contractual records, *Amicus* is well positioned to facilitate referrals to legal and dispute-resolution professionals, including lawyers, notaries, mediators, and arbitrators. Referral arrangements create an additional revenue stream while improving access to downstream services when specialized expertise is required. The completeness of the record reduces onboarding friction for these professionals, making participation more attractive.

10 Community, Jurisdictional Stewardship, and Professional Access

Contract law is broadly recognizable across legal systems. Core principles such as intention, consideration, authority, performance, and breach are shared across jurisdictions, even where their doctrinal expression and procedural handling differ. These commonalities make it possible to design a unified contracting model. At the same time, jurisdictional distinctions remain legally material and cannot be abstracted away without loss of fidelity.

Amicus is designed to accommodate both realities: a shared structural foundation paired with jurisdiction-specific context.

10.1 A Shared Foundation with Local Context

At its core, *Amicus* provides a jurisdiction-agnostic framework for managing the contract lifecycle. The system captures intent, negotiation history, authority, amendments, performance, and supporting evidence in a consistent and chronologically coherent manner. This shared structure enables reuse, comparability, and interoperability across borders.

Jurisdiction is treated as contextual rather than decorative. It informs interpretation, defaults, expectations, and downstream handling without fragmenting the underlying model. This allows contracts to remain structurally consistent while being locally intelligible and defensible.

10.2 The Amicus Community Organization

To support this approach, *Amicus* contemplates an independent community organization that serves as a neutral platform for collaboration, stewardship, and shared understanding. The organization is not a law firm and does not provide legal advice. Its role is structural and educational rather than advisory.

The community organization exists to:

- steward shared contracting concepts, terminology, and lifecycle patterns,
- facilitate discussion around evidence-aware and enforcement-aligned contracting,
- support jurisdiction-specific interpretation without centralizing authority, and
- promote clarity, accountability, and defensible record-keeping practices.

Participation is open to practitioners, technologists, academics, and organizations with a common interest in improving how contracts are formed, managed, and evidenced over time.

10.3 Jurisdictional Chapters

Within the broader community organization, jurisdictional chapters provide localized stewardship. Each chapter corresponds to a specific jurisdiction or closely related legal system.

Chapters articulate how the shared *Amicus* model maps onto local legal reality, including:

- jurisdiction-specific defaults, assumptions, and constraints,
- common drafting conventions and professional practice norms,
- evidentiary expectations and procedural considerations, and
- interaction with local courts, institutions, and dispute-resolution mechanisms.

Chapters do not modify the core platform. Instead, they contextualize it, ensuring that contracts remain both structurally coherent and locally meaningful.

10.4 Access to Jurisdiction-Specific Professionals

While *Amicus* is designed to support clarity and accountability throughout the contract lifecycle, certain agreements require human professional judgment. Sophisticated transactions, novel structures, cross-border arrangements, or escalated disputes may exceed what software alone should handle.

To address this reality, the Amicus ecosystem includes a directory of jurisdiction-qualified professionals. This directory may include lawyers, notaries, mediators, arbitrators, and other regulated practitioners, organized by jurisdiction and area of specialization.

The directory is informational in nature. It does not provide legal advice, imply endorsement, or create professional relationships by default. Its purpose is to make qualified expertise discoverable when needed.

10.5 Contextual Awareness Without Endorsement

Because *Amicus* maintains a complete contractual record covering negotiation, amendment, authority, and performance, it can recognize moments where professional consultation is commonly sought. In such cases, the system may surface neutral prompts indicating that practitioners in the relevant jurisdiction and specialty are available for consultation.

These prompts are contextual, optional, and non-directive. *Amicus* does not recommend specific professionals, rank practitioners, or evaluate quality. The decision to consult, select, and engage a professional remains entirely with the user.

10.6 Alignment Without Centralization

This model avoids two common failures in legal technology:

- over-centralization, in which a single authority attempts to standardize legal interpretation across jurisdictions; and
- fragmentation, in which each jurisdiction develops incompatible tools and practices.

By combining a shared technical foundation with jurisdictional stewardship and optional access to human expertise, Amicus enables alignment without uniformity. Contracts remain coherent across systems while respecting local legal reality.

As the *Amicus* model extends into repeatable commercial relationships, master agreements, and collective governance, this community structure provides a durable bridge between individual contracting activity and institutional practice—grounded in real-world enforcement and professional accountability rather than abstraction.

11 Founder – Samuel Bourque

Samuel Bourque is the founder of *Amicus Docs* and the principal architect of its lifecycle-native approach to contracting. His work sits at the intersection of legal systems, governance design, and software engineering, with a consistent focus on aligning technical infrastructure with how institutions actually operate over time.

11.1 Education and Experience

Samuel holds degrees in Physics, Computer Science, and Law. His career highlights include serving as Deputy Director in technology infrastructure for a multinational insurance company, where he led large-scale automation efforts on its global technology infrastructure. In a separate role, he led specialized trading system infrastructure as a Vice President at a top-tier investment bank. He has also served as CTO of a digital asset exchange and, in a later role, as CTO for a derivatives trading platform. In addition, he has worked as a managing consultant and product manager for a smart contract platform.

11.2 Contract Automation and Smart Contracts

Samuel has been working on automating contracts since 2014. *Amicus* has been taking shape conceptually over the past decade, evolving through successive attempts to align software with the way legal obligations are actually formed, performed, and enforced. His early work on smart contracts exposed a core tension: many smart contract platforms operate largely outside the traditional legal system and adopt a "code is law" posture, in which the behavior of the program is treated as final, even when it diverges from the parties' actual intention. In that model, code can displace both human intention and the legal framework that exists to interpret and enforce it. *Amicus* represents a different posture—using technology to support contractual practice as it exists in law, rather than attempting to bypass or supplant it.

11.3 Governance and Design Philosophy

In parallel with this technical work, Samuel has been directly involved in trust, foundation, and organizational governance, both as a practitioner and as a system designer. This exposure informs *Amicus*'s core thesis: that contracts are not static documents, but living records whose meaning emerges through process, performance, and change. His approach favors conservative legal framing, intuitive user experience, and rigorous evidentiary integrity over novelty or abstraction.

Amicus Docs reflects this philosophy by treating contracting as durable infrastructure rather than transactional software, and by designing records, roles, and timelines so they can withstand institutional change. The goal is a stable layer of agreement infrastructure that can support real-world enforcement, accountability, and institutional memory.

12 Conclusion

12.1 People-first, Lifecycle-Native Contracting

Contracts are not static artifacts; they are lived relationships that unfold over time. Software that treats contracting as a single moment of execution fails to reflect how agreements are actually formed, performed, and enforced. *Amicus Docs* articulates a lifecycle-native alternative—one that aligns the act of contracting with the realities of accountability, evidence, and enforcement.

This white paper outlines a model for treating contracts as durable records rather than isolated documents, and for managing them within a single, coherent system from negotiation through conclusion. *Amicus Docs* is intended as foundational infrastructure for this purpose.

12.2 Get in Touch

We welcome conversations with investors, early users, partners, and advisors who share an interest in rethinking contract infrastructure.

For inquiries or to engage further, contact:

sam@amicusdocs.com

Glossary

Agreement (n.): a **Proposal** that has been fully executed by the necessary **Parties** and is now in effect. The **Agreement**, together with its **Casefile** history, represents the binding contract and its authoritative record. An **Agreement** may arise from a signed **Document**, but the **Agreement** is the legal relationship, not the file itself.

Assistant (n.): the AI-driven drafting and review helper embedded in the **Casefile**. The **Assistant** can suggest language, surface context, and help with structure, but it never becomes a **Party**, **Principal**, or decision-maker; final authority remains with humans.

Casefile (n.): a single, unified record for a contractual relationship. A **Casefile** holds the full lifecycle of an **Agreement**—drafting, negotiation, execution, amendments, performance events, and supporting materials—within one coherent timeline, and may contain one or more **Templates**, **Proposals**, and **Signable documents** over time.

Contract (n.): an **Agreement** that satisfies the legal requirements for contract formation in the relevant jurisdiction, such as offer and acceptance, sufficient certainty of **Terms**, consideration or an accepted functional equivalent, capacity, and any required formalities. *Amicus* does not itself determine whether an **Agreement** is a **Contract**. Instead, it preserves records so that this question can be evaluated by the **Parties** and, where necessary, by courts or other decision-makers.

Document (n.): a text artifact stored within a **Casefile**. **Documents** include drafts, notes, rendered **Templates**, **Signable documents**, and other human-readable materials. A **Document** can evidence an **Agreement**, but it is not itself the **Agreement**; it is an artifact that records some aspect of the **Parties**' intention or performance.

Marketplace (n.): the environment within *Amicus* where **Templates**, **Public offers**, and related contractual assets can be discovered, shared, and reused. The **Marketplace** supports reuse of well-structured **Agreements** while preserving **Parties**' autonomy and jurisdictional specifics.

Party / Parties (n.): the legal persons (individuals or entities) who hold rights and obligations under an **Agreement**. **Parties** are the subjects of the contract; they are bound by, and benefit from, the **Terms** and **Specs**.

Proposal (n.): a populated instance of a **Template** that has been drafted but not yet fully executed. A **Proposal** reflects the current state of negotiation—including **Terms**, **Specs**, **Roles**, and **Role Assignments**—before all required signatures are in place.

Public offer (n.): a prepared **Proposal** or pattern for a **Signable document** that counterparties can accept with minimal friction, often from a repeatable pattern. **Public offers**

are used for scenarios like standard onboarding, self-serve sign-ups, or **Marketplace** distribution.

Role / Roles (n.): named positions defined by a **Template** or **Agreement** (for example, Buyer, Seller, Landlord, Tenant, Administrator). **Roles** identify who is acting in the relationship and what authorities or permissions attach to that position.

Role Assignment / Role Assignments (n.): explicit records in the **Casefile** that map **Parties** to **Roles**. A **Role Assignment** answers "which **Party** is acting as Buyer?" or "who is the Tenant?" and is part of the authoritative history of the **Agreement**.

Specs (n.): structured details that describe what is being provided, exchanged, or done—for example, item lists, service descriptions, service levels, or delivery conditions. **Specs** make the subject matter concrete and machine-readable without deciding timing or conditions.

Signable document (n.): the concrete **Document** instance that enters the signature workflow. A **Signable document** is typically produced from a **Proposal** and, once fully executed, may give rise to an **Agreement**. The structured JSON record described in this white paper corresponds to this **Signable document** layer, not to the entire **Casefile**.

Template (n.): a reusable contract pattern that defines the structure of an **Agreement**. A **Template** specifies which **Roles** exist, what **Terms** and **Specs** it uses, and how those pieces fit together.

Terms (n.): structured parameters that describe how the relationship operates over time—such as price models, payment schedules, renewal behavior, termination conditions, milestones, and other operational rules. **Terms** govern the mechanics and evolution of the **Agreement** rather than its subject matter.

Coda: Extensions of the Lifecycle Model

This white paper focuses on the foundations of lifecycle-native contracting and Casefile-based records. The same model, however, naturally extends beyond individual agreements.

In subsequent work, the Casefile abstraction expands to cover repeatable commercial relationships—such as master agreements with catalogs, orders, invoices, and payments—where multiple agreements operate under shared terms and structure.

Beyond commerce, the model further generalizes to collective contexts, including organizational and governance scenarios, where authority, delegation, and decision-making must be recorded with the same clarity, integrity, and respect for autonomy described here.

These domains are not departures from the core thesis, but direct consequences of it. They will be addressed in future publications.

Copyright Notice