ADLIB

Making Claims Ready for AI Agents

Combine insurance-native, agentic intake with an Accuracy & Trust Layer to make AI-based claims automation fast, compliant, and defensible.

Who this is for:

- Claims Operations leaders, Claims Transformation / COE owners, IT platforms leaders
- Anyone responsible for **cycle time, leakage, compliance, and audit defensibility** across document-heavy claims workflows.

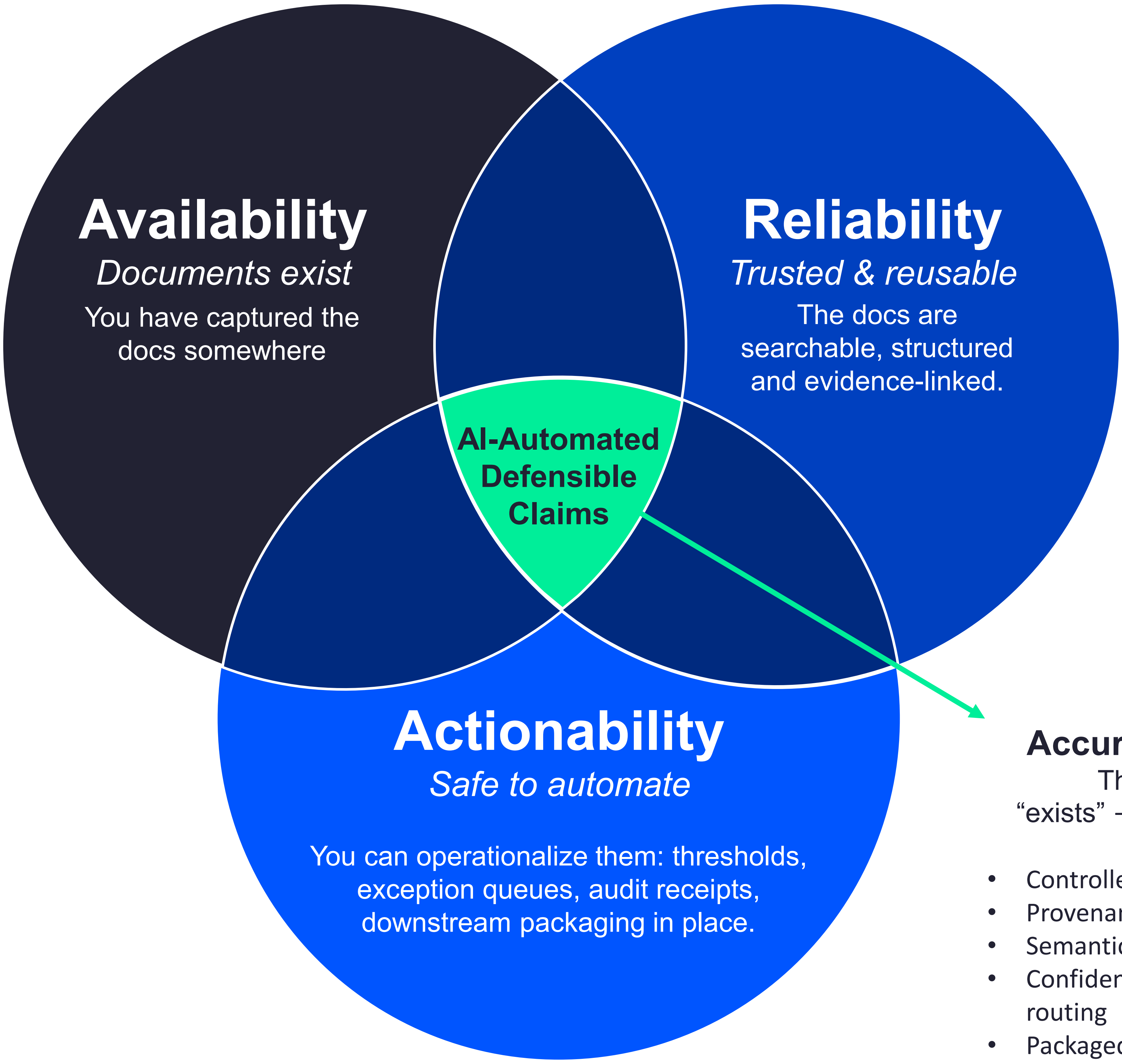
ADLIB

The Claims Truth Gap: Availability vs Reliability vs Actionability

In claims, “the truth” arrives messy.

A single claim can include a rotating cast of inbound content: email threads, scanned forms, police reports, medical records, repair estimates, photos, invoices, and adjuster notes, often in mixed quality and mixed formats.

Documents can exist without being trustworthy. AI-based automation only works in the overlap.



Availability

Documents exist

You have captured the docs somewhere

Reliability

Trusted & reusable

The docs are searchable, structured and evidence-linked.

AI-Automated Defensible Claims

Actionability

Safe to automate

You can operationalize them: thresholds, exception queues, audit receipts, downstream packaging in place.

Accuracy & Trust Layer

The bridge between “exists” → “usable” → “defensible”

- Controlled conversion + OCR
- Provenance + version control
- Semantic structure
- Confidence thresholds + exception routing
- Packaged as a Claims Data Product

ADLIB

The Missing Layer: Accuracy & Trust

Upstream of claims platforms and AI

Most teams try to modernize claims by picking a new claims platform workflow, bolting on OCR/IDP, or layering in GenAI. Then they discover the hard truth:

Every downstream system inherits upstream uncertainty.

An upstream Accuracy & Trust Layer turns inbound claims chaos into trusted, machine-navigable inputs for downstream claims systems, fraud workflows, analytics, and AI copilots.

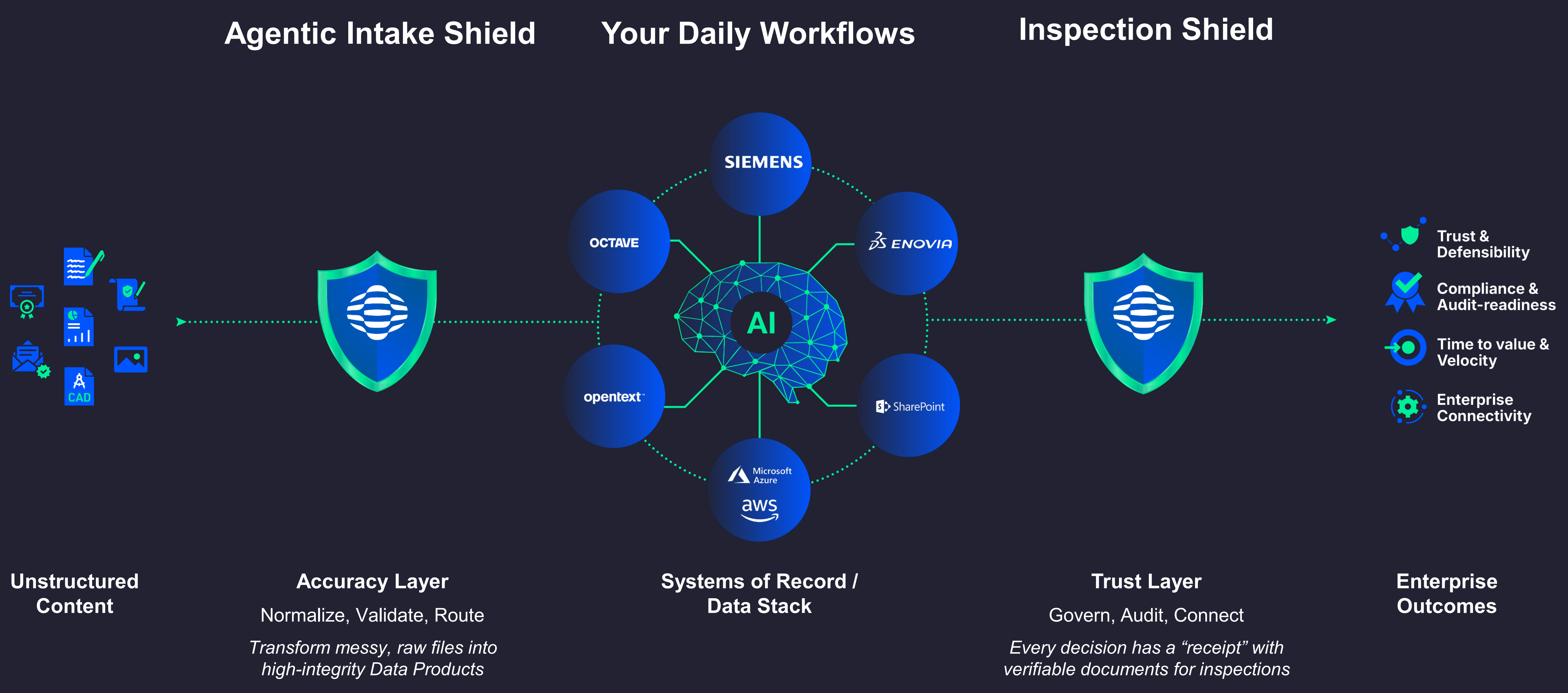
How This Architecture Works in Insurance

Think of the modern claims stack as two protective shields:

1) Agentic Intake Shield (the "front door" or the Accuracy Layer)
Insurance-native, agentic intake that captures inbound communications (especially email + attachments), preconditions claim documents into contextualized and AI-ready claim packets, structures them into tasks/cases inside claims systems, and reduces manual triage and routing friction.

2) Inspection Shield (the Trust Layer)
Validates, governs, and routes work based on measurable confidence so decisions remain defensible and audit-ready.

In short: intake protects what goes in; trust protects what goes out.



Agentic Intake Shield

Your Daily Workflows

Inspection Shield

Trust & Defensibility

Compliance & Audit-readiness

Time to value & Velocity

Enterprise Connectivity

Unstructured Content

Accuracy Layer
Normalize, Validate, Route
Transform messy, raw files into high-integrity Data Products

Systems of Record / Data Stack

Trust Layer
Govern, Audit, Connect
Every decision has a "receipt" with verifiable documents for inspections

Enterprise Outcomes

What “Trust Controls” Look Like in Claims

Claims need quality gates for document + data.
Examples of practical trust controls you can run every day:

1 Completeness gates

Required docs present for claim type (e.g., PoL + estimate + photos) before downstream steps proceed

2 Identity & key-field validation

Policy #, claim #, insured name/address consistency across sources

3 Confidence thresholds per document class

Different pass/fail thresholds for claim form vs. medical record vs. estimate (not one global setting)

4 Exception routing

Low-confidence items auto-route to human review; high-confidence flows straight-through

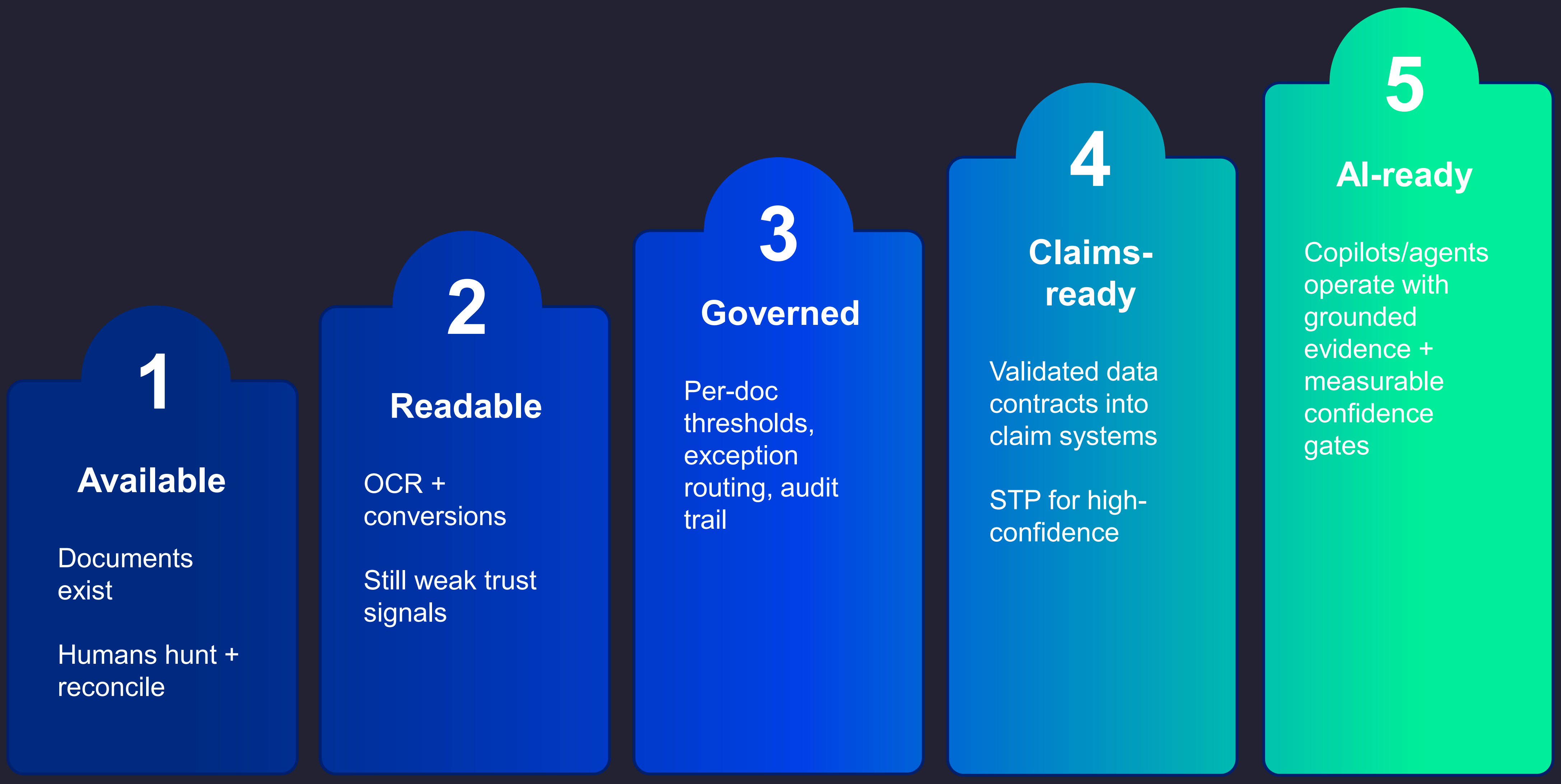
5 Document-of-record production

Pixel-perfect, searchable, audit-ready outputs (e.g., PDF/A where required) with traceability

6 Decision receipts

Every automated action logs “what happened, why, with what confidence” so audits don’t become archaeology

A Simple Claims Trust Maturity Model



1 **Available**

Documents exist

Humans hunt + reconcile

2 **Readable**

OCR + conversions

Still weak trust signals

3 **Governed**

Per-doc thresholds, exception routing, audit trail

4 **Claims-ready**

Validated data contracts into claim systems

STP for high-confidence

5 **AI-ready**

Copilots/agents operate with grounded evidence + measurable confidence gates

Claims Modernization-Readiness Checklist

Reality check: If any of the items below are being handled by spreadsheets, shared drives, manual PDF clean-up, or “tribal knowledge approvals,” you’re building in error, low throughput, and bottlenecks. It will not scale across lines of business, geographies, or volume spikes.

Target state: an automated pipeline that standardizes → makes machine-readable → validates with trust controls → packages outputs for repeatable ingestion into claims systems and downstream AI.

1. Intake Channel & Case Structuring Readiness

What breaks when manual: Claims arrive fragmented across inboxes, portals, and forwarded threads; attachments get missed or separated from context; the same claim gets worked twice (or not at all) because there’s no consistent case creation and routing.

Automated readiness requirements:

☐ Capture inbound communications end-to-end (email + attachments) with lineage preserved
☐ Detect missing attachments and incomplete submissions at intake
☐ Associate every inbound artifact to the correct claim/policy context
☐ Classify intake intent and document types
☐ Convert inbound messages into structured claim tasks/cases with consistent routing rules
☐ De-duplicate repeated submissions and near-duplicates
☐ Preserve a complete intake timeline

Ready looks like: Every inbound claim packet becomes a complete, structured case with correct attachments, clear routing, and a defensible intake timeline, ready for downstream automation.

2. Format & Fidelity Readiness

What breaks when manual: Adjusters save/print/scan documents inconsistently; pages get lost, rotated, cropped, or compressed; attachments detach from the claim record.

Automated readiness requirements (what the pipeline must do):

☐ Convert inbound files into controlled, fidelity-preserving outputs
☐ Detect and prevent missing pages/attachments (packet completeness checks)
☐ Normalize geometry (rotation/deskew/crop) so key regions don’t “drift”
☐ Apply consistent image cleanup (contrast/denoise) without altering evidentiary content
☐ Preserve original files + produce a controlled “gold” copy for processing and audit

Ready looks like: Every claim packet renders deterministically and completely.

3. Revision & Provenance

What breaks when manual: Teams argue over “latest vs. authoritative” versions; duplicates flood queues; decision timelines are reconstructed by hand during disputes/audits; provenance gets lost when documents move between inboxes and folders.

Automated readiness requirements (what the pipeline must do):
☐ Capture source provenance (sender, timestamps, system of origin, attachment lineage)
☐ Link every document to claim + policy context (and retain chain-of-custody)
☐ Detect duplicates and near-duplicates and apply dedupe rules (retain best/authoritative)
☐ Apply versioning rules by document type (authoritative vs. supplemental vs. replaced)
☐ Maintain a traceable document timeline (what arrived when, what changed, and why)

Ready looks like: One defensible claim record with clean version control, no more “which estimate is the right one?” debates.

4. Searchability & Machine-Readable Text Readiness

What breaks when manual: Scanned PDFs aren’t reliably searchable; people skim and miss details; OCR errors slip through as “truth”; AI search/RAG underperforms because the text layer is incomplete or garbled.

Automated readiness requirements (what the pipeline must do):
☐ Create a high-quality searchable text layer for scans and image-based PDFs
☐ Detect low-text-quality outputs (garbled OCR, missing text, language/encoding issues)
☐ Preserve page coordinates for evidence anchoring (quote/cite where a value came from)
☐ Normalize text for retrieval (consistent encoding, whitespace, and section boundaries)
☐ Route low-confidence text to remediation instead of silently passing downstream

Ready looks like: Every document is reliably searchable and citation-ready. Humans and AI can find the same facts fast.

5. Policy & Claim Key Identifiers Readiness

What breaks when manual: A single mis-keyed policy or claim number routes documents to the wrong case; insured vs. claimant confusion causes delays/disputes; mismatches are discovered late (after downstream steps fail).

Automated readiness requirements (what the pipeline must do):
☐ Extract and normalize critical identifiers (policy #, claim #, insured, claimant, loss date, loss location; plus VIN/property IDs as applicable)
☐ Validate identifier formats (length/pattern checks; carrier-specific rules where available)
☐ Reconcile identifiers across documents in the packet (cross-doc consistency checks)
☐ Detect conflicts (two claim numbers, mismatched insured names) and flag as exceptions
☐ Prevent downstream progression when identity is unresolved (gated routing)

Ready looks like: Identifiers are captured once, validated early, and reused everywhere—no downstream rework from mismatched policy/claim data.

6. Semantic Structure Readiness

What breaks when manual: Triage relies on subject lines and memory; document types get misclassified; estimates/invoices are summarized inconsistently; line items are re-entered or ignored, increasing leakage and slowing settlements.

Automated readiness requirements (what the pipeline must do):
☐ Classify each document by type (FNOL, estimate, invoice, police report, medical bill, etc.)
☐ Extract structured fields per doc type (dates, parties, totals, coverage-relevant facts)
☐ Preserve tables and line items (repair estimate lines, invoice/bill lines) as usable data
☐ Standardize a “claim facts” model (consistent field definitions by claim type)
☐ Attach evidence anchors for key fields (page/region references)

Ready looks like: Every document class produces consistent, structured claim facts, line items included, ready for rules, analytics, and automation.

7. Validation, Trust Controls & Exception Routing

This is the difference between “automation” and “safe automation.”

What breaks when manual: Either everything gets reviewed (slow) or nothing does (risky); exceptions are found late; QA varies by person and region; decisions lack a consistent “why” trail for audits, disputes, and governance.

Automated readiness requirements:
☐ Assign confidence scores by document type and field criticality
☐ Apply validation gates (completeness, consistency, reasonableness checks)
☐ Configure thresholds per doc type (not one global confidence setting)
☐ Route exceptions to the right queue (adjuster, supervisor, SIU, medical review) with reasons
☐ Log decision receipts (what passed/failed, why, evidence anchors, and who resolved)

Ready looks like: High-confidence flows straight-through; only true exceptions reach humans, with clear reasons and defensible audit trails.

8. Packaging for Ingestion (reusable “Claims Data Product”)

What breaks when manual: Downstream integrations are brittle and one-off; reprocessing creates duplicates; different teams define “done” differently; analytics and automation can’t scale because outputs aren’t standardized.

Automated readiness requirements (what the pipeline must do):
☐ Produce a standardized output bundle per claim packet (“claims data product”)
☐ Include controlled documents of record + structured fields/tables + provenance metadata
☐ Include confidence scores, validation results, and exception status/resolution notes
☐ Enforce a stable schema/contract for ingestion into Claims systems and downstream tools
☐ Support safe reprocessing (idempotency, versioning) + monitoring metrics

Ready looks like: Any downstream system can ingest the same reusable, versioned claims data product, without custom rework per team or claim type.

“What You Get” Artifacts

✓ Controlled documents of record

✓ Structured claim facts + line items

✓ Confidence + validation results

✓ Exception log + resolution

✓ Evidence anchors (“decision receipts”)

✓ Claims Data Product schema/contract

ADLIB

Why Not Just OCR/IDP + GenAI?

OCR/IDP can produce text and extracted fields, but claims automation needs trust: measurable confidence gates, provenance, exception routing, and audit-ready outputs. Without that upstream layer, every downstream system inherits upstream uncertainty.

Approach	What it's good for	Where it fails in claims (what breaks)	Why it fails (root cause)	Why the Accuracy & Trust Layer is key
OCR / Document Conversion	Turning scans/images into text; basic searchability; converting file formats	Garbled text, missed characters, unreadable zones; inconsistent quality across scans; downstream automation runs on "bad text"	OCR produces *text*, not *trust*; quality varies widely by input fidelity; weak detection of when text is unreliable	**Readiness gates** detect low-quality text early; controlled conversion creates consistent, machine-readable "gold" versions; low-quality routed to remediation instead of poisoning downstream
IDP (classification + field extraction)	Classifying doc types; extracting common fields from known layouts/templates	Wrong doc classification; field extraction errors treated as truth; critical identifiers mismatch across docs; exceptions discovered late	One-size thresholds across doc types; limited cross-document reconciliation; lack of evidence anchoring and provenance	**Per-doc and per-field confidence thresholds**, cross-doc validation (consistency/reasonableness), and evidence anchors turn extraction into defensible inputs
Rules engines / workflow automation (in Claims Systems)	Routing and workflow steps once data is clean; operationalizing known processes	Workflows stall on missing/incorrect upstream info; high exception volume; rework and manual triage persist	Claims platforms assume inputs are reliable; they don't solve untrusted upstream documents	The layer **standardizes + validates upstream** so workflows receive clean, consistent, versioned input, — reducing exceptions before they hit Claims Systems
GenAI summarization (notes, packet summaries)	Faster reading; initial summaries; drafting correspondence; extracting "themes"	Hallucinations; missed caveats; inconsistent outputs; poor grounding when documents aren't searchable/clean	GenAI inherits uncertainty; without strong text + evidence anchors, it can't reliably cite facts	The layer provides **machine-readable text + evidence anchors + trust signals**, enabling grounded summaries and defensible outputs
GenAI/RAG search across claim docs	Semantic search; Q&A across the claim file; faster retrieval	Wrong answers due to weak OCR; missing attachments; conflicting versions; poor relevance due to messy structure	RAG can't retrieve what isn't indexed correctly; conflicts aren't resolved; no governance over "authoritative" docs	The layer enforces **completeness, version/provenance control, and retrieval-ready indexing**, so AI searches the *right* content and can cite it
Agentic automation without Trust Layer	Automating routine tasks when inputs are reliable; orchestrating steps across systems	Risky actions on uncertain data; silent errors; hard-to-audit decisions; escalations when confidence is low	Agents act on imperfect inputs; lack of measurable confidence gates + exception routing	The layer adds **trust controls, thresholds, and exception routing** so agents only act when safe and produce decision receipts when they do
Point solutions (fraud flags, vendor portals, intake tools)	Improving a single step (e.g., intake capture, anomaly signals)	Siloed outputs; inconsistent formats; brittle integrations; doesn't create reusable, system-wide data contract	Optimizes one step but doesn't standardize end-to-end; no shared "claims data product"	The layer packages outputs as a **reusable Claims Data Product** (docs + structured data + provenance + confidence + exceptions) for any downstream consumer

ADLIB

Key Claim Documents, How the Accuracy & Trust Layer Delivers Value, Expected Outcomes

Key Claims Document Types	Where claims agents get tripped up today	How Accuracy & Trust Layer Delivers Value	Expected Outcomes
Claim forms / FNOL	Missing fields, conflicting versions from broker/insured, attachments not linked to claim, re-keying identifiers from emails	Standardize intake; extract/normalize key identifiers; completeness gates for required fields/docs	Faster claim setup, fewer NIGO claims, less rework
Proof of Loss	Wrong version, unsigned/partially completed forms, scanned quality too poor to read, duplicate submissions	Controlled conversion + searchable text; version/provenance tracking; field validation + exception routing	Fewer re-requests, fewer delays, stronger defensibility
Police / incident reports	Inconsistent formats by jurisdiction, hard-to-find incident details, poor scans, multiple parties/vehicles confusion	OCR + classification; extract incident date/location/parties; confidence gating + targeted review	Faster liability assessment, fewer missed details
Medical records & bills	Unstructured packets, buried dates/codes/amounts, duplicates, poor OCR, inconsistent patient/claimant identifiers	Normalize + index; extract key fields; reconcile identity; route low-confidence to specialist review	Reduced leakage, faster review, improved audit posture
Repair estimates	Multiple revisions from shops, line items not captured, totals don't reconcile, supplements arrive separately	Version control + dedupe; preserve tables/line items; validate totals vs. line items; exception routing	Faster settlement, fewer overpayments, fewer disputes
Invoices / receipts	Missing vendor info, ambiguous service dates, unreadable images, mismatched totals, duplicates	Controlled conversion; extract vendor/date/amount; reasonableness checks; dedupe	Fewer payment errors, faster approvals
Photos / supporting evidence	Attachments separated from context, unclear subject/location/time, duplicates, "which photo matters?" overload	Preserve provenance; link to claim + doc set; dedupe/cluster; ensure completeness of evidence set	Less back-and-forth, better adjuster productivity
Adjuster reports & notes	Notes scattered across systems, inconsistent summaries, hard to prove "why" later, copy/paste errors	Convert + index; structure key facts; evidence anchoring + decision receipts	Faster supervisor review, stronger auditability
Email threads & attachments	Key facts buried in long threads, missing attachments, mismatched claim/policy numbers, inconsistent routing	Agentic intake structuring; attachment completeness checks; identifier validation + reconciliation	Fewer triage delays, fewer misrouted claims
Coverage letters / correspondence	Templates diverge, supporting evidence not traceable, versions stored in multiple places	Controlled document-of-record outputs; provenance + versioning; traceability to evidence	Reduced dispute risk, faster compliance response

ADLIB

Book a workshop

“Complex Claim to Data Product”

Bring one painful, real-world claim packet set. We’ll map what can be automated, where trust thresholds belong, and what your packaged outputs should look like.

You’ll walk away with:

- a reference workflow: ingest → convert → extract → validate → assemble → deliver
- defined trust controls (thresholds + exception routing)
- a metrics plan (pass rate, exception rate, latency, accuracy uplift)
- a “ready for Claims/AI systems” output definition (documents of record + structured data contract)

SCAN ME

About Adlib

Adlib is the accuracy authority in AI-driven document automation for high-stakes, regulated industries. As the critical layer in front of agentic systems, LLMs, and RAG pipelines, Adlib ingests messy, multi-format content and transforms it into AI-ready, machine-navigable pipelines. The platform normalizes file types, applies fidelity-preserving rendering and advanced OCR, classifies and chunks content with citation anchors, enriches it with metadata, extracts information into structured data contracts, and validates outputs against each organization’s business and compliance rules. This results in compliant, searchable outputs and high-quality structured data that downstream systems can trust.

adlibsoftware.com