



ADROSONIC®



Agentic SOV Cleansing & Intelligence System

Automating Statement of Values standardisation through collaborative AI agents.

24-Hour Hackathon Challenge · AI Agents · Insurance and Enterprise Automation

24 hrs CHALLENGE DURATION	2–4 TEAM SIZE	4 agents REQUIRED ARCHITECTURE	3 samples TEST SOV FILES	≥ 74% MAPPING ACCURACY TARGET
-------------------------------------	-------------------------	---	------------------------------------	--

01 PROBLEM TITLE

What you are building

Agentic SOV Cleansing & Intelligence System – automating Statement of Values standardisation through collaborative AI agents.

02 DETAILED PROBLEM STATEMENT

Background, context, and the core gap

Business Context

A Statement of Values (SOV) is the foundational data file in Property and Casualty insurance. It catalogues every insured asset for a client, including location, building characteristics, construction type, and monetary values, and feeds directly into exposure models that determine maximum financial risk and underwriting strategy.

SOVs arrive from clients in wildly inconsistent formats: different sheet structures, non-standard column names, absent headers, merged cells, and missing or malformed data. Currently the Exposure Management Team cleanses these files manually, a slow, error-prone, and unscalable process with no standardised procedure.

FINANCIAL STAKES

Poor SOV data quality directly causes incorrect risk assessment, mispriced policies, and underwriting losses. The cleaned SOV is the single source of truth for all downstream exposure models.

03 CHALLENGE

What makes this hard

The challenge is not merely parsing an Excel file. Every client submission is effectively a unique schema. Key obstacles your system must overcome:

- **Structural chaos:** SOV files arrive with different sheet counts, header rows at varying positions, merged cells, and no naming convention shared across clients or submissions.
- **Semantic gap:** Column headers are rarely identical to the target schema. Resolving “Fire Prot.” to “Fire Sprinklers (Y/N)” or “Bldg Repl Cost New” to “Building Value” requires contextual reasoning, not simple pattern matching.
- **LLM reliability:** AI models can hallucinate mappings or invent data. Every AI-generated recommendation must be validated before it touches the data.
- **Schema strictness:** Downstream exposure models are strictly typed. A wrong column name, wrong data type, or missed required field will corrupt downstream risk calculations.
- **Scale and performance:** Enterprise SOV files can span thousands of rows and dozens of sheets. Processing must remain performant within the 24-hour hackathon timeline.

04 OBJECTIVES AND SCOPE

What your solution must achieve in 24 hours

Primary Objectives (Must-Have in 24 Hours)

These are the core deliverables your solution will be evaluated on. All five must be addressed to qualify for full scoring.

1. **Ingest and detect structure.** Accept an Excel SOV file (.xlsx or .csv), identify the correct data sheet, and detect the header row regardless of its position or format.
2. **Map columns to target schema.** Map all raw column headers to the 17 required target fields using a combination of exact matching, fuzzy matching, and semantic similarity. Output a confidence-scored JSON mapping.
3. **Detect data quality issues.** Identify missing values, incorrect data types, logical errors (negative TIVs, future Year Built), and format inconsistencies across all mapped columns.
4. **Generate explainable recommendations.** For every mapping and every data issue detected, produce a plain-English explanation of why the recommendation was made and what the agent is uncertain about.
5. **Human approval and transformation.** Present recommendations to a human reviewer through a UI. Apply transformations only after explicit approval. Export the final cleaned SOV as an .xlsx file conforming to the target schema.

Bonus Objectives (Stretch Goals)

These are not required for evaluation but will earn additional points and demonstrate system maturity.

- **Vector database integration:** store past successful mappings for cross-submission learning.
- **Agent workflow visualisation:** a real-time diagram of agent states and handoffs.
- **Iterative refinement (reinforcement learning):** the agent re-reasons when a human rejects a recommendation with feedback.
- **Confidence calibration:** show an aggregate data quality score per SOV at intake.

24-HOUR SCOPE NOTE

Focus on file ingestion, schema detection, quality analysis, human approval, and cleaned output. A working end-to-end pipeline on 3 test SOV samples with at least the 4 required agents is the minimum bar. Do not over-engineer the UI; a functional approval interface matters more than aesthetics.

05 FUNCTIONAL REQUIREMENTS

Specific capabilities the system must implement

FR-1 File Ingestion

- Accept .xlsx and .csv file uploads through a web or desktop UI.
- Support multi-sheet workbooks; the system must identify the relevant Excel tabs and headers automatically.
- Handle header rows at any row position, not necessarily row 1.
- Read files up to 5,000 rows without timeout or memory failure.
- Produce a structured internal representation (for example, a pandas DataFrame) before passing to agents.

FR-2 Schema Detection Agent

- Identify and rank candidate data sheets by header density and data completeness.
- Map each raw column header to a target field using a two-pass strategy.
- Pass 1: exact string match, then normalised fuzzy match (RapidFuzz or equivalent, threshold ≥ 0.75).
- Pass 2: semantic similarity using sentence embeddings (Sentence-BERT or OpenAI Embeddings) for unresolved headers.
- Assign a confidence score (0.0 to 1.0) and resolution method to every mapping.
- Flag unresolved columns with confidence below 0.50 for human review.
- Output a structured JSON mapping object (format shown in Section 09).

FR-3 Data Quality and Anomaly Detection Agent

- Calculate the per-field completeness rate (percentage of non-null values) for all mapped columns.
- Validate data types against the target schema for every row: Building Value, Contents, BI, and Other must be numeric; Zip, Storeys, Number of Buildings, and Year Built must be integer; Fire Sprinklers must be one of Y, N, Y13, or Y(13R).
- Detect logical errors such as negative monetary values, Year Built after the current year, or Storeys below 1.
- Detect format inconsistencies such as currency symbols in numeric fields, mixed date formats, and inconsistent state abbreviations.
- Produce a structured data quality report with per-row and per-field anomaly flags.

FR-4 Recommendation and Reasoning Agent

- Consume outputs from FR-2 and FR-3 and generate a recommendation list.
- Each recommendation must include an action type (column_mapping, data_correction, standardisation, or flag_for_review), a proposed transformation with a before and after example, a plain-English rationale, and a confidence score with an explicit uncertainty statement where confidence is low.
- Powered by OpenAI GPT-4o (Assistant API or direct completion) or an equivalent LLM.
- Must not auto-apply any recommendation. All suggestions are queued for human review.
- Support re-reasoning: if a human rejects a recommendation with a note, the agent should re-evaluate.

FR-5 Human Approval Interface (Orchestration Layer)

- Display, for each recommendation, the raw column, proposed mapping, confidence score, and rationale.
- Allow the reviewer to accept, reject, or edit each recommendation individually.
- Show a data preview with before and after cleaning side by side for at least a sample of rows.
- Provide an “Approve All” action for high-confidence recommendations (≥ 0.90).
- Prevent file export until all flagged, low-confidence items have been reviewed.

FR-6 Transformation and Output Agent

- Apply only approved transformations. No data mutation without human sign-off.
- Rename columns to exact target schema field names (case-sensitive).
- Cast data types according to the data dictionary (Section 09).
- Handle null values by preserving them as empty cells; do not substitute zeros or placeholders.
- Export cleaned output as a single-sheet .xlsx file named “Cleaned_SOV.xlsx”, with the sheet named “Cleaned_SOV”, headers in row 1, data from row 2, and no merged cells.
- Generate a full audit log as a second sheet (“Audit_Log”) or a companion .json file. Each entry should record source_column, target_column, transformation_applied, before_value, after_value, confidence, approved_by, and timestamp.

06 NON-FUNCTIONAL REQUIREMENTS

Performance, quality, and operational standards

ID	Requirement	Acceptance criterion
NFR-1	Mapping accuracy	Header mapping accuracy $\geq 74\%$ on the provided test SOV samples (baseline is 53%).
NFR-2	Anomaly recall	The data quality agent must detect $\geq 90\%$ of known anomalies in the test files.
NFR-3	Audit completeness	100% of applied transformations must appear in the audit log with source, target, confidence, and timestamp.
NFR-4	Error handling	The system must not crash on malformed input and must return a user-readable error message for any unsupported file.
NFR-5	Output schema conformance	Cleaned output must pass schema validation: all 17 required columns present, correct data types, no merged cells.
NFR-6	Explainability	Every agent decision must have a human-readable rationale string. No silent transformations.
NFR-7	Agent state sharing	Agents must share context through a structured state object. No agent may operate on inputs the previous agent has not validated.

07 SOLUTION CAPABILITIES

The four-agent architecture you must implement

Your solution must implement four collaborating AI agents. Each agent owns a discrete stage of the pipeline, passes structured outputs to the next agent, and contributes to a shared state object that is visible to the human reviewer.

Agent 1: Sheet Intelligence and Discovery Agent

Objective. Identify where the meaningful business data resides within the workbook before any mapping or transformation occurs.

Responsibilities

- Scan all worksheets in the uploaded workbook.
- Analyse structural characteristics, including header quality and density, data consistency and regularity, column richness, null-value distribution, and row volume and continuity.
- Score and rank each worksheet based on data relevance.
- Categorise sheets as Primary (likely source of core business data), Secondary (supporting or reference data), or Reject (metadata, notes, summaries, or low-value sheets).
- Generate a ranked sheet manifest with confidence scores and reasoning.

Example output

```
{
  "sheet_name": "SOV_Data",
  "classification": "Primary",
  "confidence": 0.94,
  "reasoning": [
    "Consistent tabular structure",
    "High header confidence",
    "Low null ratio"
  ]
}
```

Success metric. Accurately identify the workbook's authoritative data source while minimising false-positive sheet selections.

Agent 2: Schema Mapping Agent

Objective. Translate source columns into the standardised 17-field target schema with explainable confidence.

Responsibilities

- Receive only the Primary sheet(s) identified by Agent 1.

- Execute a two-stage mapping process: Pass 1, fuzzy header matching; Pass 2, semantic understanding using LLM reasoning.
- Resolve variations in naming conventions and business terminology.
- Assign confidence scores to each mapped field and identify ambiguous or unmapped columns.
- Produce a structured schema-mapping output ready for validation.

Example output

```
{  
  "target_field": "PolicyNumber",  
  "source_column": "Pol No",  
  "confidence": 0.97,  
  "mapping_method": "Semantic"  
}
```

Success metric. Maximise mapping coverage and confidence while maintaining traceability and explainability.

Agent 3: Data Quality and Reasoning Agent

Objective. Assess dataset readiness by detecting data quality issues and generating actionable remediation recommendations.

Responsibilities

- Validate mapped fields against schema expectations and profile dataset completeness and consistency.
- Detect missing values, invalid data types, duplicate records, format inconsistencies, business-rule violations, and cross-field logical anomalies.
- Use LLM reasoning to explain identified issues, assess business impact, recommend corrective actions, and prioritise remediation opportunities.
- Generate a human-review recommendation queue.

Example output

```
{
  "issue": "Policy Start Date occurs after Policy End Date",
  "severity": "High",
  "affected_rows": 47,
  "recommendation": "Swap dates where source system formatting is
                    reversed and flag exceptions for review."
}
```

Success metric. Provide high-quality, explainable recommendations that reduce manual data cleansing effort.

Agent 4: Controlled Transformation Agent

Objective. Execute approved transformations safely, transparently, and auditably.

Responsibilities

- Activate only after explicit human approval.
- Apply approved corrections and transformations while preserving original source data integrity.
- Generate a complete audit trail for every change and produce the final standardised workbook.
- Record transformation metadata, approvals, and execution details.

Output. Cleaned_SOV.xlsx, a transformation audit log, and a processing summary report.

Success metric. Deliver a production-ready dataset with full traceability and zero unapproved modifications.

08 TARGET OUTPUT FORMAT: DATA DICTIONARY

The exact schema your output must conform to

Every cleaned SOV exported by your system must conform exactly to the following schema. Field names are case-sensitive. Downstream exposure models perform strict schema validation and will reject files that deviate from this specification.

OUTPUT FORMAT

File: Cleaned_SOV.xlsx. Sheet name: "Cleaned_SOV". Headers: row 1. Data: row 2 onward. No merged cells. No colour formatting. UTF-8 encoding. Blank cells for missing values, not zeros or "N/A".

Required field	Data type	Definition
Reference	String	Unique identifier for the property used in insurance documentation.
Address	String	Full street address of the insured property.
City	String	City where the insured property is located.
State	String	State or region (2-letter abbreviation preferred).
Zip	Integer	Postal / ZIP code for geographic risk identification.
County	String	County for regional risk analysis and policy assignment.
Country	String	Country; defines jurisdiction for policy coverage.
Building Value	Float	Insured value of the building structure (replacement cost / market value).
Contents	Float	Value of personal property or contents within the building.
BI	Float	Business Income coverage; potential income loss during interruption.
Occupancy	String	Use or purpose of the building (residential, commercial, industrial).
Construction	String	Construction type or material (wood, steel, concrete, masonry).
Storeys	Integer	Number of floors in the building.
Number of Buildings	Integer	Total number of buildings at the location covered under the policy.
Year Built	Integer	Year the building was constructed.
Fire Sprinklers (Y/N)	String	Whether fire sprinkler systems are installed: Y, N, Y13, or Y(13R).
Other	Float	Additional insured values not classified under Building Value or Contents.

Example JSON mapping format (Schema Detection Agent output)

```
{
  "sheet_identified": "Sheet2 - Location Schedule",
  "header_row": 3,
  "mappings": {
    "Loc #": { "target": "Reference", "confidence": 0.97, "method": "fuzzy" },
    "Street Address": { "target": "Address", "confidence": 0.99, "method": "exact" },
    "Bldg Repl Cost": { "target": "Building Value", "confidence": 0.91, "method": "semantic" },
    "Fire Prot.": { "target": "Fire Sprinklers", "confidence": 0.84, "method": "semantic" },
    "Unresolved_Col_7": { "target": null, "confidence": 0.31, "method": "none",
      "flag": "human_review_required" }
  },
  "unresolved_count": 1,
  "overall_confidence": 0.87
}
```

09 CONSTRAINTS AND RESTRICTIONS

Hard rules your solution must respect

Mandatory Constraints

ID	Constraint	Detail
C-01	No auto-transformation	Transformations must never be applied without explicit human approval. Silent data mutation is a disqualifying failure.
C-02	No hallucinated data	The system must not invent, interpolate, or fill in missing values with fabricated data. Missing fields must remain blank in the output.
C-03	Schema conformance required	Output must contain exactly the 17 required fields. No additional columns, no renamed fields, no reordered columns.
C-04	Multi-agent architecture required	Single-agent or monolithic pipelines will not score on the Agentic Design criterion. State must be shared across at least 4 distinct agents.
C-05	Explanation required for every decision	Every mapping and every data transformation must carry a human-readable rationale string. Unexplained changes will receive zero Explainability score.
C-06	Test on provided samples	Evaluation will use the 3 anonymised sample SOV files provided. Solutions must process all 3 without crashing.
C-07	Output file naming	Cleaned output must be named exactly "Cleaned_SOV.xlsx". Audit log must be named "Audit_Log.xlsx" or "Audit_Log.json".

Data Privacy Constraints

- The 3 test SOV files provided have been anonymised. All named insureds, street addresses, and contact details have been replaced with synthetic equivalents.
- Your solution should be designed to handle sensitive commercial data securely; evaluators will note data handling practices.
- Secure data handling practices are preferred over open source solutions where data security is a concern.
- Do not hardcode API keys in your submission. Use environment variables or a secrets manager.

10 SUCCESS METRICS

How your solution will be scored

Criterion	Weight	How it is measured
Mapping accuracy	20%	Percentage of headers correctly mapped to target schema on all 3 test SOV samples (target $\geq 74\%$, baseline 53%).
Data quality detection	15%	Recall on planted anomalies in test files: nulls, type errors, logical violations (target $\geq 90\%$).
Transformation correctness	15%	Percentage of output rows in the cleaned file that match ground truth for type, value, and field name (target $\geq 95\%$).
Agentic design quality	20%	Judges assess agent separation, state sharing, orchestration logic, and whether the 4 required agents are clearly distinct.
Explainability	15%	Percentage of recommendations that include a clear, accurate natural-language rationale (target 100%).
Human-in-the-loop UX	10%	Approval interface completeness: can a non-technical reviewer understand and act on every recommendation?
Innovation and bonus	5%	Points for vector memory, workflow visualisation, multi-format support, or iterative re-reasoning.

11 DEMO REQUIREMENTS

What your live demonstration must show

Your demo must walk evaluators through the complete end-to-end pipeline using at least one of the provided test SOV files. The following 7 stages must be clearly demonstrated. Demos will be time-boxed to 10 minutes.

1. **File upload and sheet detection.** Upload a test SOV file. Show the system correctly identifying the data sheet from a multi-sheet workbook and detecting the header row.
2. **Schema detection in action.** Show the JSON mapping output with confidence scores. Highlight at least one semantic match, not just exact or fuzzy, for example “Bldg Repl Cost” to “Building Value”.
3. **Data quality report.** Display the anomaly detection output: per-field completeness scores and at least 3 flagged issues with their types.
4. **Reasoning agent recommendations.** Show the recommendation queue with natural-language rationale for at least 3 items: one column mapping, one data correction, one standardisation rule.
5. **Human approval workflow.** Demonstrate accepting a recommendation, rejecting one with feedback, and the agent re-reasoning on the rejected item or escalating to a human.
6. **Transformation and audit log.** Apply approved transformations and show the cleaned output alongside the audit log entries.
7. **Cleaned output download.** Download Cleaned_SOV.xlsx and briefly show schema conformance: correct headers, data types, and no merged cells.

12 SAMPLE DATA: REDACTION GUIDE

How the test SOV files were sanitised and what has been preserved

The 3 test SOV files distributed for this hackathon have been derived from real insurance submissions. The table below describes exactly what was redacted and what was preserved to maintain testing realism.

Preserved in test files	Redacted or replaced
All column headers, including ambiguous or abbreviated ones; TIV and monetary values; state, county, zip; construction type, occupancy, year built; fire sprinkler codes; flood, EQ, and named storm coverage flags; deductible amounts; multi-sheet structure and merged cells; intentional inconsistencies in header naming.	Named insured / entity names, replaced with synthetic values (for example, Redwood Housing LLC); street addresses, replaced with plausible fake addresses in the same state or county; property manager names, replaced with [PROPERTY MANAGER]; broker and adjuster names, replaced with [REDACTED]; policy numbers, replaced with SAMPLE-UMR-XXXX; email or phone fields, replaced with sample@example.com.

SAMPLE SELECTION RATIONALE

The 3 samples were chosen to maximise structural diversity. Sample 1 is clean, well-structured headers (baseline test). Sample 2 has ambiguous or abbreviated headers (semantic matching test). Sample 3 has a multi-sheet layout with missing columns (sheet detection and null handling test).

13 IMPACT CREATED

What a successful solution achieves

53% HEADER MAPPING ACCURACY, BASELINE (MANUAL)	Hours MANUAL ANALYST TIME PER SOV
≥ 74% HEADER MAPPING ACCURACY, AGENTIC TARGET	< 60 sec AGENTIC PIPELINE PLUS HUMAN REVIEW

- **Speed:** hours of analyst time per SOV reduced to a sub-minute agentic pipeline plus a brief human review.
- **Scalability:** handles any SOV format without manual rule updates or template maintenance.
- **Consistency:** eliminates human error variance across analysts, time zones, and experience levels.
- **Auditability:** every transformation is logged and narrated, supporting full compliance readiness.
- **Downstream reliability:** schema-conformant output means exposure models receive high-integrity data.
- **Compounding accuracy:** vector memory stores successful mappings, so the system improves with every SOV processed.

This challenge goes beyond automation. It explores how AI agents can collaborate, reason, and assist humans in critical enterprise workflows.

Property & Casualty Insurance AI · Confidential · For Hackathon Use Only



ADROSONIC®

THANK YOU

www.adrosonic.com