

# Warranty Claims Processor

## Design and Evaluation Report

---

## 1. Problem Framing and Assumptions

### Problem Overview

The company sells multiple hair dryer products, each governed by its own warranty policy. Customers submit warranty-related issues through unstructured emails that vary in clarity, completeness, and relevance. These emails must be interpreted, matched against the correct product policy, and evaluated consistently.

Manual claim processing is slow, difficult to scale, and prone to inconsistency across reviewers. The goal of this project is to design an **agentic AI system** that assists human reviewers by automating information extraction, policy matching, and decision reasoning while keeping humans in control of final outcomes.

### Objectives

The system aims to:

- Triage incoming emails into claims vs non-claims
- Extract structured claim information from unstructured text
- Select the correct product warranty policy
- Generate policy-grounded decision recommendations
- Support a human-in-the-loop review process
- Produce customer-ready communication drafts

### Key Assumptions

- The company sells **ten distinct hair dryer products**
  - Each product has its own warranty policy
  - All products include a **three-month warranty**, with varying coverage and exclusions
  - Shipping address is required **only after approval**, not for eligibility
  - Ambiguous cases should escalate to humans rather than auto-reject
  - Minimizing false approvals is more important than full automation
- 

## 2. System Design and Agent Roles

### High-Level Architecture

The solution is implemented as a **single orchestrating agent** coordinating multiple specialized tools. This design balances automation with explainability and control.

The orchestrator manages the workflow, while each tool has a narrowly defined responsibility.

## Core Components

1. **Inbox Tool**  
Reads incoming claim emails from a file-based inbox directory.
  2. **Triage Tool**  
Classifies emails as warranty claims or non-claims (spam or irrelevant). The triage logic is intentionally permissive to avoid rejecting legitimate claims.
  3. **Extraction Tool**  
Extracts structured data from emails, including product name, purchase date, issue description, proof of purchase, and shipping address (if present).
  4. **Policy Retriever**  
Selects the most relevant warranty policy based on extracted product information and retrieves relevant coverage and exclusion sections.
  5. **Decision Tool**  
Applies deterministic, policy-grounded logic to generate:
    - A recommendation (Approve / Reject / Need More Info)
    - Confidence level
    - Explicit reasoning
    - Referenced policy excerpts
    - Follow-up questions
  6. **Email Writer**  
Drafts customer-facing emails based on the final decision and missing information.
  7. **Label Generator**  
Generates a mock return shipping label when a claim is approved and operationally ready.
  8. **Human Reviewer (CLI)**  
Reviews the structured packet and makes the final decision.
- 

## 3. Policy Selection and Decision Rationale

### Policy Selection Strategy

Policies are stored as structured JSON documents, one per product. Policy selection follows a tiered approach:

1. Exact product name match
2. Fuzzy matching on product names
3. Context-based fallback (least preferred)

The selected policy and the reason for selection are explicitly recorded in the review packet.

### Decision Logic

The system uses **deterministic logic**, not LLM judgment, for eligibility decisions:

- Exclusions always override approvals

- Warranty window is calculated from purchase date when available
- Statements like “purchased last month” are treated as likely in-warranty but flagged for confirmation
- Missing shipping address does not affect eligibility
- Missing proof of purchase or purchase date lowers confidence or triggers follow-up questions

## Policy Grounding

Each recommendation is grounded in **explicit policy excerpts**. The human review packet includes:

- Relevant coverage sections
- Relevant exclusion clauses
- Clear mapping between claim details and policy language

This ensures transparency and auditability.

---

## 4. Human-in-the-Loop Flow

The system is designed to assist, not replace, human reviewers.

### Review Workflow

1. The system generates a **Human Review Packet** containing:
  - Extracted claim data
  - Selected policy and excerpts
  - Recommendation and confidence
  - Facts, assumptions, and reasoning
  - Follow-up questions (if any)
2. The packet is displayed in the terminal.
3. The human reviewer chooses one action:
  - Approve
  - Reject
  - Request more information
4. The system executes the human decision by:
  - Drafting customer emails
  - Generating return labels (if approved and address exists)
  - Requesting missing information when needed

The human decision always overrides the system recommendation.

---

## 5. Sample Inputs and Outputs

### Sample Input (Claim Email)

```
{
```

```
"email_id": "claim_001",
"subject": "Hair dryer overheats",
"body": "I purchased the AeroDry Pro 1800 from Amazon on November 12,
2025. After about a month of normal use, the dryer overheats and shuts
off.",
"attachments": ["invoice.pdf"]
}
```

### Sample Review Packet (Excerpt)

```
{
  "recommendation": "APPROVE",
  "confidence": "high",
  "facts": [
    "Issue reported: Dryer overheats and shuts off",
    "Extracted product: AeroDry Pro 1800",
    "Purchase date provided: 2025-11-12",
    "Relevant policy sections reviewed: Covered Issues - Overheating during
normal use"
  ],
  "reasoning": [
    "Warranty window check: 30 days since purchase",
    "No applicable exclusions found"
  ],
  "customer_followup_questions": [
    "Please provide your shipping address so we can generate the return
label."
  ]
}
```

### Sample Customer Email

Hello,

Your warranty claim has been approved. To proceed with the replacement, please provide your shipping address so we can generate a return label.

Thank you,  
AeroDry Warranty Team

---

## 6. Evaluation Plan

### Offline Evaluation

A labeled test set of sample claims can be created with expected outcomes.

Metrics include:

- Decision accuracy vs labeled outcomes
- False approval rate
- False rejection rate
- Percentage of claims receiving confident recommendations

### Online Monitoring

In production, the following signals would be tracked:

- Approval vs rejection distribution
- Human override rate
- Average reviewer time per claim
- Latency per claim
- Cost per claim (LLM usage)

## Error Analysis

Common failure modes include:

- Incorrect product identification
- Ambiguous purchase timing
- Edge-case exclusions

Mitigations include improved extraction prompts, structured product identifiers, and expanded exclusion detection.

---

## 7. Future Work and Intentional Limitations

### Future Improvements

- Web-based reviewer UI
- Vector-based policy retrieval
- Learning from reviewer decisions
- Integration with ticketing systems
- Automated regression tests for policy logic

### Intentional Omissions

The following were intentionally not implemented:

- Fully autonomous approvals
- Real shipping carrier integrations
- Real email delivery
- Authentication and authorization

These omissions prioritize safety, explainability, and correctness within the project scope.

---

## 8. Conclusion

This project demonstrates how **agentic AI** can safely support complex, policy-driven workflows. By combining deterministic logic, explicit policy grounding, and human oversight, the system provides consistent, explainable decision support and serves as a strong foundation for a production-grade warranty claims platform.