Automating **Letter of Credit (LC) document checking** using **Agentic AI** is a strong candidate for intelligent task decomposition. As a Trade Finance expert, you already know LC document checking involves meticulous comparison of documents against LC terms under **UCP 600**, **ISBP 821**, and possibly **URC 522** for collections.

To implement this in **Agentic AI architecture**, you'd break down the process into **autonomous agents**, each responsible for a distinct skill aligned with real-world trade operations.

**🧠 Agentic AI Architecture for LC Document Checking**

**🎯 Goal**

Automatically analyze a set of trade documents submitted under an LC and determine **discrepancies**, **compliance**, and **next actions**.

**🧩 Agent Decomposition**

| **Agent Role** | **Function** | **Core Inputs** | **Outputs** |
| --- | --- | --- | --- |
| **1. LC Extractor Agent** | Parse and extract LC terms (amount, expiry, ports, latest shipment, documents required, etc.) | LC PDF or MT700 Swift message | Structured JSON with LC terms |
| **2. Document Classifier Agent** | Classify uploaded documents (e.g. Invoice, BL, CO, etc.) and match to LC requirements | Trade document set (PDF or scanned images) | Document-type-tagged data |
| **3. Data Extraction Agent** | Extract key fields (e.g., invoice value, BL shipper, CO origin) from each document | Tagged documents | JSON of extracted fields by doc |
| **4. Compliance Validator Agent** | Compare each document’s content with LC terms & UCP 600/ISBP 821 rules | LC terms + extracted doc data | List of compliant/non-compliant items |
| **5. Discrepancy Identification Agent** | Detect and explain discrepancies based on mismatches or missing items | Compliance results | Discrepancy list with ICC justification |
| **6. Decision Agent** | Recommend whether to accept, reject, or request waiver of discrepancies | Discrepancy data, bank policy | Recommendation with rationale |
| **7. Report Generator Agent** | Compile a full checking report for Trade Ops | All agent outputs | LC Document Checking Report (PDF/JSON) |
| **8. Feedback & Learning Agent** *(optional)* | Learn from user overrides (e.g. waived discrepancies) to improve logic | Manual corrections | Updated prompt/knowledge for validators |

**⚙️ Integration with Tools**

* **OCR** for scanned documents: Tesseract, Google Vision, or Azure OCR
* **LLMs**: For UCP/ISBP reasoning and extraction (GPT-4o, Claude, etc.)
* **Rule engine**: For hard-coded validations (e.g., date formats, field presence)
* **PDF parser**: PyMuPDF, PDFMiner, or LayoutLM for structured extraction

**📂 Example Flow**

1. User uploads LC (MT700) and 7 trade documents (Invoice, BL, Packing List, etc.)
2. LC Extractor Agent parses MT700 to JSON LC format.
3. Document Classifier Agent tags the uploaded files.
4. Data Extraction Agent parses fields like:
   * Invoice amount, date, terms
   * BL port of loading/discharge
   * Certificate country of origin
5. Compliance Validator Agent checks:
   * Is invoice amount ≤ LC amount?
   * Is port of discharge as per LC?
   * Is BL signed and dated before latest shipment?
6. Discrepancy Agent finds missing or mismatched fields.
7. Decision Agent uses bank policy (e.g., accept typographical discrepancy) to recommend "Accept with Waiver."
8. Report Generator outputs PDF summary with UCP references.

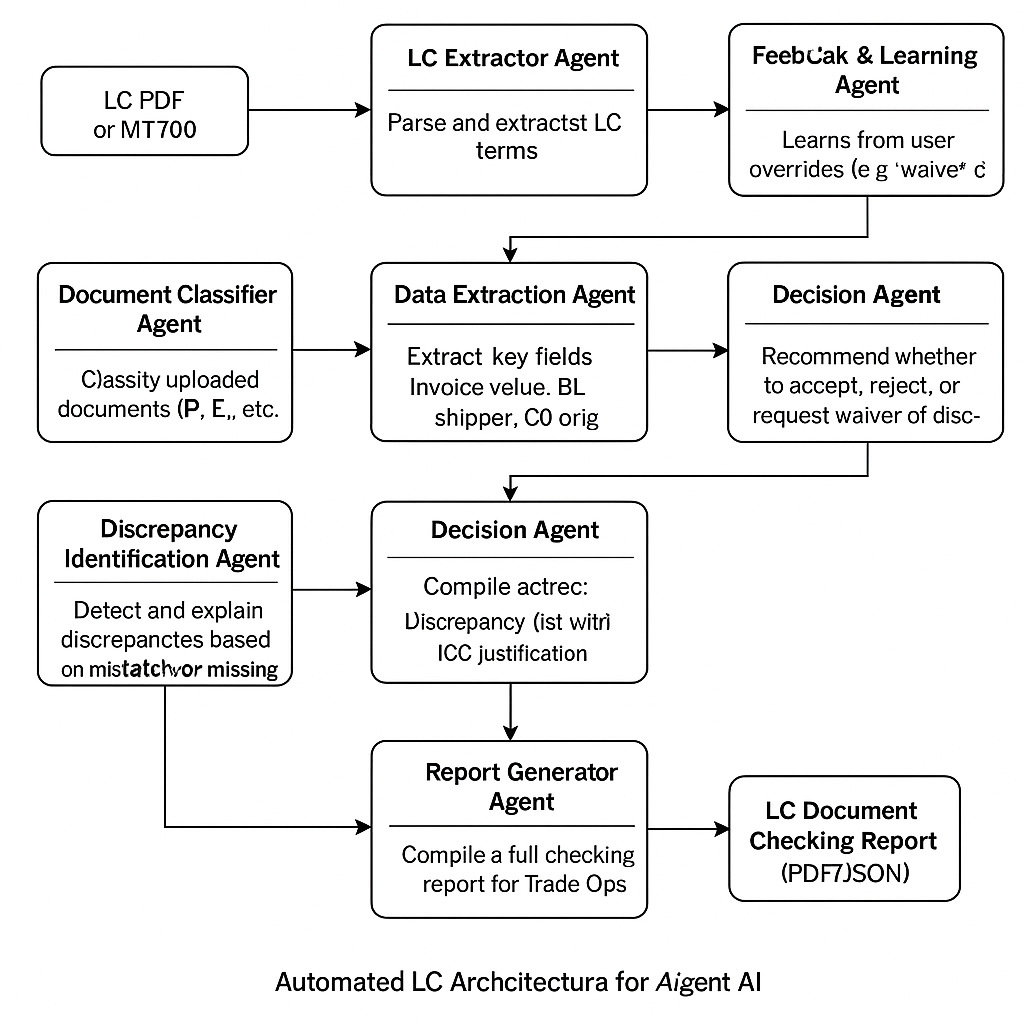
**✅ Real-World Extension**

* Integration into **back-office workflow tools** like Eximbills, Finastra, Surecomp, or Oracle Trade Finance
* API bridge to **SWIFT Alliance Access** for MT700/707/750/999 parsing
* Fine-tuning on **internal discrepancy archives** to improve Agent judgment

**🔐 Compliance Consideration**

* All outputs should **cite UCP/ISBP clauses** (e.g., “ISBP 745 para 25(a)”)
* System must support **auditable logs** for compliance review
* Manual override by LC checker must be preserved

**Demo architecture diagram**



**Code Example**

* *LLM => lc\_doc\_check\_llm.py*
* *Agent SDK => lc\_doc\_check\_agentsdk.py*