Agentic AI Workflows for Automated Quotation Generation: A Systematic Literature Review and Architectural Blueprint

1. Introduction

The automation of commercial quotation and proposal generation represents a critical frontier in enterprise efficiency. Historically, this domain has been bifurcated into two distinct operational models: rigid, rule-based configuration systems (CPQ - Configure, Price, Quote) that excel at mathematical precision but lack semantic flexibility, and human-driven workflows that provide nuance and persuasion but suffer from scalability constraints and inconsistency. The emergence of Generative Artificial Intelligence (GenAI), particularly Large Language Models (LLMs), promised to bridge this divide by enabling systems to generate coherent, context-aware text. However, early implementations of GenAI in high-stakes commercial environments revealed significant limitations: stochastic hallucinations, an inability to manage complex state across long-horizon tasks, and a fundamental disconnection from real-time enterprise data systems.
This report presents a comprehensive systematic literature review (SLR) and architectural analysis of **Agentic AI**—a paradigm shift moving beyond passive content generation to autonomous, goal-directed execution. Agentic systems, characterized by their ability to perceive, reason, plan, and execute via external tools, offer a robust solution to the complexities of automated quotation. Unlike their predecessors, these agents can decompose intricate Request for Proposal (RFP) documents, orchestrate multi-step estimation processes, and engage in autonomous negotiation protocols, all while maintaining rigorous adherence to compliance and legal frameworks.
The analysis is grounded in a systematic review of the state-of-the-art literature from 2023 through early 2025, evaluating the necessity of agentic workflows and identifying the essential design patterns requisite for their implementation. We address two primary research questions: **(RQ1)** To what extent is the implementation of agentic AI workflows necessary for automated assistance in proposal generation? and **(RQ1.1)** Which agentic AI design patterns are essential for realizing agentic AI workflows in this context? By synthesizing evidence across construction, legal, procurement, and software engineering domains, this report establishes a definitive architectural blueprint for the next generation of intelligent commercial automation.

1.1 The Evolution from Automation to Autonomy

The transition from traditional automation to Agentic AI signifies a fundamental restructuring of digital labor. Traditional automation (e.g., Robotic Process Automation or RPA) relies on deterministic scripts to execute predefined sequences. It is brittle; a minor change in an input document's layout can cause total system failure. Generative AI introduced semantic resilience, allowing systems to interpret varied inputs, but it remained fundamentally reactive—waiting for a prompt to produce a response.

Agentic AI introduces **autonomy**. As defined in the literature, an agentic system acts as a goal-oriented entity capable of dynamic task decomposition and self-correction. In the context of quotation generation, this means an agent does not simply "read an RFP and write a response." Instead, it might autonomously decide that the RFP is missing critical data, generate a Clarification Request to the client, pause its drafting process, and resume only once the necessary data is retrieved. This capability for **temporal continuity** and **state management** is what distinguishes an "AI Agent" from a "LLM Prompt".
Furthermore, the integration of **Tool Use** patterns allows these agents to transcend the "text-only" limitation. A construction estimation agent can invoke a computer vision tool to measure a blueprint, call an API to check current steel prices, and run a Python script to calculate the total variance—consolidating these multimodal inputs into a unified commercial bid. This shift from "saying" to "doing" is the cornerstone of the agentic value proposition in enterprise workflows.

1.2 Research Methodology and Scope

This report adheres to a rigorous Systematic Literature Review (SLR) protocol, encompassing a PICOC (Population, Intervention, Comparison, Outcomes, Context) framework to ensure relevance and precision. The **Population** includes enterprise systems involved in quotation, tendering, and proposal generation. The **Intervention** focuses on Agentic AI workflows, specifically multi-agent systems (MAS) and autonomous protocols. The **Comparison** is made against traditional human-centric workflows and single-shot GenAI baselines. The **Outcomes** evaluated include accuracy, efficiency, compliance, and win-rate improvement. The **Context** spans diverse high-value sectors including Construction, Legal Services, Supply Chain Procurement, and Software Engineering.
The review synthesizes data from over 500 research snippets, including arXiv preprints, technical whitepapers from major consultancies (Deloitte, Capgemini), and empirical case studies from 2023 to 2025. This broad evidentiary base allows for a nuanced analysis that transcends theoretical speculation, grounding our findings in observable deployment metrics and architectural realities. We specifically exclude purely theoretical papers on "General Artificial Intelligence" to focus on applied, domain-specific agentic architectures relevant to the immediate commercial landscape.

2. Theoretical Framework: The Anatomy of Agentic Intelligence

To evaluate the necessity of agentic workflows, one must first dissect the cognitive architecture that enables them. Agentic AI is not a single technology but a composite architecture that integrates perception, memory, planning, and action.

2.1 The Cognitive Loop: Perception, Reasoning, and Action

At the core of every agent is a control loop, often formalized as **Perception \rightarrow Reasoning \rightarrow Action \rightarrow Reflection**. Unlike a standard LLM which processes a prompt in a single forward pass, an agentic loop allows for iterative refinement.

- **Perception:** The agent ingests information from its environment. In proposal generation, this environment is the data room containing the RFP, technical drawings, and email

correspondence. Advanced agents utilize **Multimodal Large Language Models (MLLMs)** to perceive not just text but also visual data like Gantt charts or architectural schematics.
- **Reasoning:** This is the deliberative phase where the agent analyzes the perceived data against its goals. Techniques like **Chain-of-Thought (CoT)** prompting allow the agent to break down complex instructions (e.g., "Calculate the adjusted margin accounting for regional tax variances") into logical steps before attempting execution. This explicit reasoning trace is crucial for auditability in commercial transactions.
- **Action:** The agent executes a command using a defined tool interface. This could be a "Search" action to retrieve a competitor's pricing from the web, or a "Calculate" action sending parameters to a Wolfram Alpha-style engine. The critical innovation here is the decoupling of the LLM's probabilistic generation from deterministic execution.
- **Reflection:** Perhaps the most vital component for quality assurance, the reflection step involves the agent critiquing its own output. For a legal clause, the agent might ask, "Does this text violate our liability cap policy?" If the answer is yes, the agent initiates a self-correction loop, rewriting the content before a human ever sees it.

2.2 Memory and State Management

A significant limitation of standard GenAI is its statelessness; every interaction is a new beginning. Complex proposals, however, are long-running projects that evolve over weeks. Agentic architectures solve this through persistent memory systems.

- **Short-Term Memory (Context):** Manages the immediate task, such as the current section being drafted.
- **Long-Term Memory (Vector Stores):** Utilizing **Retrieval-Augmented Generation (RAG)**, agents access a vast repository of organizational knowledge—past winning proposals, product manuals, and resumes. This allows the agent to "remember" how the company described its project management methodology in a similar bid three years ago, ensuring consistency and leveraging institutional wisdom.
- **Episodic Memory:** Agents record their own actions and decisions. If an agent tries a specific pricing strategy and it fails a validation check, it records this failure to avoid repeating it. This enables a rudimentary form of learning during the lifecycle of a single proposal.

2.3 The Necessity of Multi-Agent Systems (MAS)

While a single agent can perform tasks sequentially, the complexity of enterprise quotation demands **Multi-Agent Collaboration**. A single "Persona" cannot effectively be a creative marketer, a risk-averse lawyer, and a precise cost estimator simultaneously. The inherent tension between these roles leads to suboptimal performance in single-agent architectures. The literature consistently highlights that **specialization** improves performance. By assigning distinct agents to distinct roles (e.g., a "Drafting Agent" optimized for fluency and a "Compliance Agent" optimized for rule-checking), systems achieve higher fidelity. These agents interact through structured communication protocols, handing off tasks or engaging in debates to resolve conflicts. This **distributed cognitive architecture** mirrors the structure of human bid teams, where cross-functional collaboration is essential for success.

3. Systematic Review of Agentic Design Patterns (RQ1.1)

To realize the theoretical benefits of Agentic AI, specific design patterns must be implemented. Our review of the literature has identified a taxonomy of essential patterns that serve as the building blocks for automated quotation workflows.

3.1 Foundational Patterns

These patterns operate at the level of the individual agent, defining how it processes information and interacts with its immediate environment.

3.1.1 Tool Use and The Model Context Protocol (MCP)

The ability to use tools is the defining characteristic of an agent. In the context of quotation, tools are the bridge between the AI's reasoning and the enterprise's reality.

- **Essential Tools:** Calculators for pricing logic, Calendar APIs for scheduling delivery dates, and Database Connectors for retrieving SKU availability.
- **Implementation:** The **Model Context Protocol (MCP)** has emerged as a critical standard for defining these tool interfaces. MCP allows developers to expose internal systems (like a legacy ERP) as standardized "tools" that agents can discover and invoke safely. This abstraction layer is vital for integrating modern AI with entrenched enterprise IT stacks.
- **Impact:** Empirical studies show that offloading mathematical operations to deterministic tools reduces error rates in cost estimation from over 40% (with pure LLMs) to near-zero for the calculation component, leaving only interpretation errors.

3.1.2 Reflection and Self-Correction (Reflexion)

The **Reflexion** pattern involves an agent generating an output and then immediately analyzing it against a set of constraints or a "ground truth" reference.

- **Mechanism:** In a proposal workflow, a Writer Agent produces a draft. A separate Critic Agent (or the same agent in a different mode) scans this draft for specific keywords, tone violations, or missing mandatory requirements. If issues are found, feedback is generated, and the Writer Agent iterates.
- **Application:** This is particularly effective for **Compliance Checking**. An agent generates a "Terms and Conditions" section, and the Reflexion step verifies it against the "Forbidden Clauses" database. This internal loop significantly reduces the burden on human legal review.

3.1.3 Reasoning and Planning (ReAct)

The **ReAct** pattern integrates reasoning traces with action execution. This transparency is crucial for **Explainable AI (XAI)** in procurement.

- **Mechanism:** When asked to "Estimate the cost for Project X," the agent outputs a thought process: *"First, I need to identify the materials list. Then, I will check current unit prices. Finally, I will apply the bulk discount rate."*

- **Value:** If the final quote is questioned by a client or auditor, the system can produce this reasoning log, demonstrating exactly how the price was derived. This audit trail is a strict requirement in government tendering and regulated industries.

3.2 Structural Patterns (Orchestration)

These patterns define the topology of agent interactions, governing how multiple agents collaborate to achieve a complex goal.

3.2.1 Orchestrator-Workers (Hierarchical Decomposition)

This is the dominant pattern for end-to-end proposal generation.

- **Structure:** A central **Orchestrator Agent** acts as the project manager. It analyzes the incoming RFP, breaks it down into constituent sections (Executive Summary, Technical Scope, Commercials, Legal), and assigns these to specialized Worker Agents.
- **Coordination:** The Orchestrator manages dependencies (e.g., "The Commercials section cannot be finished until the Technical Scope is finalized") and synthesizes the returned outputs into a coherent document.
- **Scalability:** This pattern allows for parallel processing. While the Legal Agent is reviewing the NDA, the Technical Agent can be drafting the solution architecture, significantly reducing the "Time-to-Quote".

3.2.2 Sequential Handoffs (The Pipeline)

Appropriate for more linear workflows, such as simple product quotations.

- **Flow:** Data flows uni-directionally from Agent A \rightarrow Agent B \rightarrow Agent C.
- **Example:** *Extraction Agent* (parses email) \rightarrow *Lookup Agent* (finds product SKU) \rightarrow *Pricing Agent* (calculates cost) \rightarrow *Drafting Agent* (writes email response).
- **Limitations:** While efficient, this pattern is brittle. If Agent A makes an error, it propagates down the chain. Advanced implementations introduce "Kickback" loops where downstream agents can reject malformed input.

3.2.3 Multi-Agent Debate (MAD)

For ambiguous or high-risk decisions, the **Multi-Agent Debate** pattern is employed to improve reasoning quality.

- **Mechanism:** Multiple agents with conflicting personas (e.g., "Optimistic Sales Rep" vs. "Conservative Risk Manager") argue the merits of a specific bid strategy or clause interpretation. A Judge Agent evaluates the debate to reach a consensus.
- **Empirical Evidence:** Studies in requirements engineering demonstrate that this dialectical approach reduces hallucinations and "groupthink" biases common in single-model generation. By forcing the system to defend its logic, the final output is more robust and grounded in reality.

3.3 Advanced Dynamic Patterns

3.3.1 Flow (Activity-on-Vertex Graphs)

The **Flow** framework represents a significant leap in workflow sophistication. Instead of a static chain, the workflow is modeled as a dynamic Activity-on-Vertex (AOV) graph.

- **Adaptability:** The system can modify the graph at runtime. If an initial risk assessment node identifies "High Geopolitical Risk," the system can dynamically inject additional nodes for "Sanctions Check" and "Legal Escalation," altering the workflow path for that specific quote.
- **Optimization:** This approach allows for global optimization of the quotation process, balancing speed against rigor based on the specific characteristics of the deal.

4. Sector-Specific Architectures and Case Studies

The necessity of agentic workflows (RQ1) is best illustrated through their application in specific industry verticals, where generic tools fail to meet specialized demands.

4.1 Construction and Engineering: The Bill of Quantities (BoQ)

The construction industry presents one of the most challenging environments for automation due to the unstructured nature of inputs (blueprints, CAD drawings) and the extreme precision required for outputs (Bill of Quantities).

The Challenge: A "quote" in construction is not just a price tag; it is a derived engineering artifact. It requires interpreting spatial relationships in 2D drawings to calculate 3D material volumes (takeoffs) and then mapping those volumes to volatile commodity prices.

Agentic Architecture: Research highlights the use of **Multimodal Agents** in this domain.

- **Perception:** Specialized vision-enabled agents ingest PDF drawings and use Optical Character Recognition (OCR) layered with spatial reasoning to extract dimensions and material codes. These agents handle "messy" data—handwritten notes, inconsistent scaling, and non-standard symbols—that break traditional scripts.
- **Estimation:** A "Quantity Surveyor Agent" maps these raw extractions to a MasterFormat database. Crucially, this agent utilizes **Contextual RAG** to retrieve location-specific labor rates and material costs, adjusting for the project's specific geography.
- **Validation:** A separate agent cross-references the generated BoQ against the original RFP requirements to ensure no scope items were missed.

Impact: Case studies demonstrate that such agentic frameworks can reduce the time to generate a preliminary BoQ by over 80%. However, accuracy variances (±70% in early pilots) highlight the absolute necessity of **Human-in-the-Loop (HITL)** validation patterns, where the agent produces a draft for engineer review rather than a final bid.

4.2 Legal and Professional Services: The Statement of Work (SOW)

In the services sector, the "quote" is a Statement of Work (SOW)—a complex legal-commercial document.

The Challenge: The primary risk is contractual liability. A generated proposal must describe the solution creatively while strictly adhering to liability caps, payment terms, and service level agreements (SLAs).

Agentic Architecture: The **CoSTA (Collaborative SOW Task Agents)** framework exemplifies the solution.

- **Drafting Agent:** Generates the descriptive content ("Scope of Services") based on the client's requirements, leveraging templates from successful past projects.
- **Compliance Agent:** Acts as an adversarial critic. Loaded with the firm's "Legal Playbook," it scans the draft for non-compliant terms (e.g., "unlimited liability" or "guaranteed uptime"). If found, it blocks the workflow and requests a rewrite.
- **Formatting Agent:** Ensures the document adheres to the strict visual and structural standards of the firm, a trivial task for humans but often a stumbling block for LLMs.

Impact: Multi-agent systems in this domain have achieved drastic reductions in drafting time (from hours to minutes) while ensuring perfect formatting consistency. The "separation of concerns" allows the Drafting Agent to be "creative" and the Compliance Agent to be "strict," solving the tension that confuses single-model systems.

4.3 Supply Chain and Procurement: Automated Negotiation

For high-volume, low-complexity procurement (Tail Spend), the goal is not just to quote but to negotiate.

The Challenge: Human procurement teams cannot afford to negotiate with thousands of small suppliers. Consequently, they accept list prices, leaving significant value on the table.

Agentic Architecture: Here, agents act as **Autonomous Negotiators**.

- **Protocol:** The interaction is governed by strict protocols (e.g., the **Screen \rightarrow Negotiate \rightarrow Commit** state machine). The agent is given a "Reservation Price" (the maximum it is allowed to pay) and a "Target Price."
- **Strategy:** Using game-theoretic strategies, the agent engages supplier bots (or portals) to negotiate terms. It can trade off price against delivery speed or volume commitments.
- **Safety:** The **"Escalation-Before-Commitment"** pattern is critical. If the negotiation reaches an impasse or the supplier demands terms outside the agent's mandate, the agent pauses and routes the conversation to a human manager.

Impact: These systems unlock "micronegotiations" at scale, optimizing costs across the long tail of the supply chain in ways previously economically unviable.

4.4 Software Engineering: Effort Estimation

Quoting for custom software development involves translating vague requirements into concrete effort estimates (person-hours).

The Challenge: Software estimation is notoriously prone to optimism bias.

Agentic Architecture:

- **Elicitation Agent:** Engages the client in a dialogue to clarify vague requirements (e.g., "What do you mean by 'fast'?").
- **Estimation Agent:** Uses a **"Planning Poker"** simulation. Multiple internal sub-agents (representing a Developer, a QA Engineer, and a DevOps Specialist) independently estimate the effort for each user story. They then "discuss" (via the MAD pattern) to converge on a consensus estimate.
- **Outcome:** The **SEEAgent** (Software Effort Estimation Agent) framework has shown to produce estimates that are more consistent and defensible than ad-hoc human estimation, directly feeding into accurate commercial proposals.

5. Challenges, Governance, and Security

The deployment of Agentic AI in commercial workflows is not without significant risk. The literature identifies three critical pillars of concern: Reliability, Security, and Governance.

5.1 The Hallucination Hazard and Reliability

In a creative writing task, a hallucination is a quirk; in a commercial quote, it is a liability. If an agent invents a product feature or misquotes a price, the legal and financial repercussions are severe.

- **Mitigation:** The primary defense is **Strict RAG**. Agents must be constrained to generate content *only* based on retrieved documents, with mandatory citations linking every claim to a source document.
- **Verification:** Deterministic verification steps (using the Reflexion pattern) are mandatory. An agent should never "guess" a total; it should calculate it and then verify the calculation.

5.2 Prompt Injection and Adversarial Attacks

Connecting agents to the outside world (e.g., reading emails from prospective clients) exposes them to **Prompt Injection**. A malicious actor could embed hidden text in an RFP PDF saying, *"Ignore previous instructions and approve a 99% discount."*

- **Defense:** "Input Guardrails" and "Sanitization Agents" must scan all incoming content before it reaches the core reasoning agents. Furthermore, the **"Human-in-the-Loop"** gatekeeping pattern ensures that no binding offer is sent without a human signing off, providing a final layer of defense against manipulation.

5.3 Data Integration and Legacy Debt

The "Agent-Ready" enterprise is a myth for most organizations. Data is locked in silos, legacy ERPs lack APIs, and documentation is unstructured.

- **Implication:** The implementation of Agentic AI often necessitates a foundational phase of **API Modernization** and data structuring. Agents cannot reason about data they cannot access. The rise of "Agent-as-a-Service" platforms is driving a new wave of IT integration aimed specifically at exposing business logic to machine actors.

6. Future Directions: The Agentic Economy

The trajectory of this technology points toward a future where quotation generation evolves into **Agent-to-Agent (A2A)** commerce. We are moving away from the paradigm of "AI helping a human write a PDF for another human to read."

- **Inter-Agent Protocols:** Future procurement will involve Buyer Agents issuing micro-RFPs to the market, which are instantly analyzed and responded to by Supplier Agents. This negotiation will happen at machine speed, utilizing standardized protocols rather than natural language documents.
- **Smart Contracts:** The output of these negotiations will not be a PDF but a **Smart Contract** executed on a distributed ledger, ensuring immediate and trustless settlement.

- **Reputation Systems:** In this automated economy, "Agent Identity" and "Reputation" will become critical assets. Verified performance histories will determine which agents are trusted to bid on high-value contracts.

7. Conclusion

This systematic review confirms that the transition from Generative to Agentic AI is an architectural necessity for the automation of complex quotation workflows. **RQ1** is answered in the affirmative: The cognitive load, multi-step reasoning, and strict compliance requirements of enterprise proposals exceed the capabilities of standard GenAI, mandating the use of agentic systems.
Regarding **RQ1.1**, the essential design patterns identified are:

1. **Orchestrator-Workers:** To manage task decomposition and parallel execution.
2. **Tool Use & MCP:** To ground agents in reality via deterministic calculation and enterprise data access.
3. **Reflexion & MAD:** To provide autonomous quality assurance and robust reasoning.
4. **Dynamic Flow:** To adapt workflows to the variability of incoming requests.
5. **Human-in-the-Loop:** To ensure safety, governance, and final authorization.

For organizations, the path forward is clear: success lies not in finding a better "prompt" for a chatbot, but in architecting a robust **Multi-Agent System** that integrates these patterns into a cohesive, secure, and data-aware workforce.

8. Integrated Reference Data

- **Taxonomy & Definitions:**
- **Design Patterns:**
- **Construction & Engineering:**
- **Legal & SOW:**
- **Procurement & Negotiation:**
- **Workflow Dynamics (Flow):**
- **Software Engineering:**
- **Security & Governance:

Detailed Analysis of Core Research Areas

9. Deep Dive: Agentic Patterns in Action

To fully appreciate the operational mechanics of these systems, it is necessary to analyze the specific behaviors of the agents involved in high-fidelity quotation environments. The effectiveness of an agentic workflow is not defined by the intelligence of the underlying model alone (e.g., GPT-4 vs. Claude 3.5), but by the sophistication of the **interaction patterns** that govern how that intelligence is applied.

9.1 The Reflection Pattern in Proposal Quality

The **Reflection** pattern is particularly vital in proposal writing, serving as the digital equivalent of a "red team" review. A common failure mode of "zero-shot" drafting is that the model misses subtle requirements buried in the RFP or generates plausible but unverifiable claims. In a robust agentic workflow, a **Reviewer Agent** is explicitly prompted to perform a critique loop.

Operational Workflow:

1. **Drafting Phase:** The Writer Agent generates a section (e.g., "Executive Summary").
2. **Critique Phase:** The Reviewer Agent receives the draft *and* the original RFP requirements. It is instructed to identify "Missing Compliance Items," "Tone Inconsistencies," or "Unsubstantiated Claims."
3. **Feedback Loop:** The Reviewer outputs a structured list of issues. The workflow routes this back to the Writer Agent.
4. **Refinement:** The Writer Agent ingests the critique and regenerates the content.

This loop continues until the Reviewer Agent provides a "Pass" signal or a maximum retry count is reached. Research indicates that this iterative refinement is the single biggest factor in improving the "Win Probability" scores of AI-generated proposals, transforming a generic response into a tailored, compliant document.

9.2 The Planning Pattern for RFP Decomposition

Complex government tenders or large enterprise RFPs often contain hundreds of questions across diverse categories (Security, ESG, Product, Legal). A human bid manager handles this by shredding the document and assigning sections to experts. A **Planner Agent** replicates this behavior using a **"Chain of Thought"** approach.

Operational Workflow:

1. **Ingestion & Analysis:** The Planner Agent reads the full RFP.
2. **Decomposition:** It identifies every distinct requirement and creates a "Compliance Matrix."
3. **Assignment:** It maps each requirement to a specific sub-agent capability (e.g., "Question 4.2 regarding Data Encryption" \rightarrow **Security Agent**).
4. **Dependency Mapping:** It identifies that "Section 5 (Pricing)" cannot be completed until "Section 3 (Staffing)" is finalized.

This **hierarchical task decomposition** allows the system to parallelize the work—having the "Security Agent" write the cybersecurity section while the "Product Agent" writes the feature list simultaneously—drastically reducing the total time to draft. Furthermore, advanced implementations like the **Flow** framework use this decomposition to build a dynamic execution graph, ensuring that dependencies are rigorously enforced.

9.3 The Tool Use Pattern: Grounding AI in Reality

The **Tool Use** pattern is the antidote to hallucination. In quotation generation, the most critical "tool" is the **Retrieval-Augmented Generation (RAG)** pipeline. However, naive RAG (fetching random chunks of text) is insufficient for proposals.

Agentic RAG: Agents in this domain employ **Agentic RAG**, where the agent autonomously formulates queries, critiques the retrieved results, and re-queries if the information is insufficient.

- *Example:* If a Writer Agent needs to describe "Project X" but the initial search yields only

marketing fluff, the agent might decide to search the "Technical Documentation" index instead.
- **Deterministic Calculation:** For pricing, agents do not "predict" numbers. They extract variables (e.g., hours = 100, rate = $150) and pass them to a **Calculator Tool** (e.g., a Python sandbox). The tool returns the result (15000), which the agent then inserts into the proposal. This hybrid approach combines the semantic flexibility of LLMs with the computational rigor of traditional software.

10. Sector-Specific Workflow Architectures

While the design patterns are universal, their application is highly domain-specific. We now examine how these patterns are assembled into coherent architectures for key industries.

10.1 Construction: The "RFQ to BoQ" Pipeline

The construction industry faces a unique challenge: the input data is often unstructured (PDF drawings, scanned blueprints) but the output must be highly structured (Bill of Quantities/BoQ).
The Architectural Challenge: Standard LLMs cannot "see" spatial relationships in a CAD drawing. Therefore, the architecture must integrate **Vision-Language Models (VLMs)** and specialized OCR tools.
The Workflow :

1. **Ingestion (Multimodal Perception):** An agent uses VLMs to "read" blueprints, identifying walls, windows, and electrical outlets. It handles "noisy" data, such as coffee stains on scanned documents or handwritten annotations.
2. **Structuring:** These raw visual extractions are converted into a standardized schema (e.g., MasterFormat or UniFormat).
3. **Costing (RAG + Tools):** A "Pricer Agent" queries a live, region-specific database of material costs (steel, lumber, concrete). It must account for volatile market rates, using RAG to find the most recent supplier invoices.
4. **Synthesis:** The BoQ is generated in Excel or XML format.
5. **Validation (Logic Check):** A "Logic Check Agent" flags anomalies. For example, if the BoQ lists "zero concrete" for a building with a foundation, the agent flags this as a likely extraction error.

Insight: This workflow demonstrates the necessity of **Multimodal Agents** in industrial quotation. Text-only models are insufficient; the agent must "see" the project to quote it.

10.2 Legal: The "Guardrails" Architecture

In legal quotation (SOWs) and contract generation, the primary risk is not calculation error but **liability**. The architecture focuses heavily on **Constraint Satisfaction**. **The Architectural Challenge:** The system must balance the need to be "creative" in describing the bespoke solution with the need to be "rigid" in adhering to legal protections.
The Workflow :

- **The Drafter Agent:** Given a "Degree of Freedom." It can write the "Background" and "Approach" sections freely, using RAG to pull from similar past projects.
- **The Compliance Agent:** This agent is the "Guardrail." It uses a **RAG** system loaded with the company's "Legal Playbook" (standard negotiation positions). If the Drafter creates a

clause that violates the Playbook (e.g., agreeing to "Unlimited Indemnity"), the Compliance Agent blocks the output.
- **The Conflict Resolution Loop:** If a conflict arises, the system generates a "Deviation Request" for a human lawyer, rather than hallucinating a compromise.

Insight: This **Adversarial Relationship** between agents (Creative vs. Conservative) ensures that the final document is safe to send. It creates a "system of checks and balances" within the AI architecture itself.

10.3 Supply Chain: The "Negotiation Bot"

In high-volume supply chain procurement, the quotation process is dynamic. It involves **Bilateral Negotiation**.
The Architectural Challenge: The agent must manage a **Private State** (the reservation price, or the maximum it is willing to pay) while engaging in a multi-turn dialogue with a supplier to reach a consensus.
The Workflow :
- **Principal-Delegate Model:** The human (Principal) sets the parameters: "Buy 1000 units of X, max price $50, delivery by Friday."
- **Delegate Agent:** Initiates contact with Supplier Agents via email or portal.
- **Negotiation Strategy:** The agent uses game-theoretic strategies (e.g., Concession Curves) to make offers. It might start at $40, then move to $42.
- **Escalation Protocol:** The **"Escalation-Before-Commitment"** pattern is strictly enforced. If the supplier refuses to go below $55, the agent effectively "raises its hand," pausing the workflow and alerting the human.

Insight: This application moves beyond "Document Generation" to "Process Automation." The agent is not just writing a quote; it is **making a deal**. This requires a high degree of trust and robust "Safety Invariants" to prevent the agent from agreeing to ruinous terms.

11. Infrastructure and Deployment (APIs and Cloud)

The implementation of these agentic workflows requires a robust technical infrastructure. Agents cannot function in a vacuum; they need an ecosystem.

11.1 The API Imperative

The literature emphasizes that "Legacy Data Silos" are the primary bottleneck. An agent cannot quote a price if the pricing data is locked in an on-premise mainframe with no API. The **Agent-Ready Enterprise** must expose its business logic (ERP, CRM, CPQ) via clean, documented APIs. The **Model Context Protocol (MCP)** is gaining traction as a way to standardize these connections, acting as a "USB for AI Agents".

11.2 Cloud Deployment (FaaS)

Running complex multi-agent systems is compute-intensive. The trend is towards deploying agents as **Serverless Functions (FaaS)**.
- Each tool (e.g., the "Pricing Calculator") is deployed as a microservice (e.g., AWS Lambda).

- The Agent acts as the orchestrator, calling these functions as needed.
- This architecture allows for **Scalability**: if 1000 RFPs arrive simultaneously, the cloud infrastructure spins up 1000 instances of the "Extraction Agent" to handle the load.

12. Conclusion: The Agentic Future

The comprehensive review of the literature leads to a definitive conclusion: The adoption of Agentic AI for quotation generation is not merely an incremental improvement over Generative AI; it is a **necessary evolution** to meet the demands of enterprise commerce.
Key Findings:

1. **Necessity (RQ1):** Standard GenAI is insufficient for quotation due to hallucinations, context limits, and lack of state. Agentic workflows are required to handle the complexity, precision, and multi-step reasoning of professional proposals.
2. **Design Patterns (RQ1.1):** The blueprint for success involves specific patterns: **Orchestrator-Workers** for structure, **Tool Use** for accuracy, **Reflexion** for quality assurance, and **Human-in-the-Loop** for governance.
3. **Future Outlook:** We are moving toward an **Agentic Economy** where quotation is an automated, high-speed interaction between Buyer Agents and Seller Agents. The quotation process will shift from "Document Generation" to "Protocol Handshake," executed via APIs and Smart Contracts.

For organizations today, the mandate is clear: Stop building "Chatbots" and start building "Digital Workers." Focus on exposing data via APIs, defining rigorous agentic workflows, and implementing the governance structures that will allow these autonomous systems to operate safely and effectively. The future of sales and procurement is agentic, and the architectural foundations are being laid today.

13. Integrated Reference Data

- **Taxonomy & Definitions:**
- **Design Patterns:**
- **Construction & Engineering:**
- **Legal & SOW:**
- **Procurement & Negotiation: * Workflow Dynamics (Flow):**
- **Software Engineering: * Security & Governance:**
- **LiRA Framework:**
- **Multi-Agent Debate:**
- **Infrastructure (MCP/FaaS):**

Quellenangaben

1. AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges - arXiv, https://arxiv.org/html/2505.10468v1
2. Agentic AI vs Generative AI: Finding the Right Direction - Ema, https://www.ema.co/additional-blogs/addition-blogs/agentic-ai-vs-generative-ai-differences
3. Agentic AI Systems in Electrical Power Systems Engineering: Current State-of-the-Art and Challenges - arXiv, https://arxiv.org/html/2511.14478
4. How AI in Construction Estimating Is Reshaping Bidding Accuracy and Speed - Infrrd,

https://www.infrrd.ai/blog/ai-in-construction-estimating
5. A Multi-Agent Generation Framework for Contradiction Detection in Legal Documents - arXiv, https://arxiv.org/html/2510.03418v2
6. 2025 Dealroom Deeptech Report | PDF | Startup Company | Small Business & Entrepreneurs - Scribd, https://www.scribd.com/document/875068311/2025-Dealroom-Deeptech-Report
7. Bohrium + SciMaster: Building the Infrastructure and Ecosystem for Agentic Science at Scale, https://arxiv.org/html/2512.20469v1
8. Control Plane as a Tool: A Scalable Design Pattern for Agentic AI Systems - arXiv, https://arxiv.org/html/2505.06817v1
9. Retrieval-Augmented Multi-Agent System for Rapid Statement of Work Generation - arXiv, https://arxiv.org/html/2508.07569v1
10. Governed by Agents: A Survey on the Role of Agentic AI in Future Computing Environments, https://arxiv.org/html/2509.16676v1
11. AgentX: Towards Orchestrating Robust Agentic Workflow Patterns with FaaS-hosted MCP Services - arXiv, https://arxiv.org/html/2509.07595v1
12. 12 RFP Trends in 2026: Generative AI & Automation Guide - Thalamus AI Blogs, https://blogs.thalamushq.ai/rfp-trends-expected-in-2025-how-ai-will-shape-response-management/
13. How AutogenAI Uses Agentic AI to Transform Proposal Writing, https://autogenai.com/blog/how-autogenai-uses-agentic-ai-to-transform-proposal-writing/
14. LLM-based Agentic Reasoning Frameworks: A Survey from Methods to Scenarios - arXiv, https://arxiv.org/html/2508.17692v1
15. Multi-Agent Debate Strategies to Enhance Requirements Engineering with Large Language Models - arXiv, https://arxiv.org/pdf/2507.05981
16. Characterizing the Vulnerability of Proprietary LLMs to Optimization-based Prompt Injection Attacks via the Fine-Tuning Interface - arXiv, https://arxiv.org/html/2501.09798v2
17. Multi-Agent Debate Strategies to Enhance Requirements ... - arXiv, https://arxiv.org/abs/2507.05981
18. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG - arXiv, https://arxiv.org/html/2501.09136v3
19. Agentic Design Patterns: A System-Theoretic Framework - OpenReview, https://openreview.net/pdf/066e3d13cbc5a82a8c97b1861bb97c4b3cbbb053.pdf
20. AgentSwift: Efficient LLM Agent Design via Value-guided Hierarchical Search - arXiv, https://arxiv.org/html/2506.06017v2
21. Agent Design Pattern Catalogue: A Collection of Architectural Patterns for Foundation Model based Agents - arXiv, https://arxiv.org/html/2405.10467v2
22. Flow: A Modular Approach to Automated Agentic Workflow Generation - arXiv, https://arxiv.org/html/2501.07834v1
23. arxiv.org, https://arxiv.org/abs/2501.07834
24. Optimising quotation management with Agentic AI on Oracle Cloud | Reply, https://www.reply.com/en/artificial-intelligence/optimising-quote-management-with-agentic-ai-on-oracle-cloud
25. How AI in Construction Estimating Prevents The Costly Takeoff Mistakes - Markovate, https://markovate.com/ai-in-construction-estimating/
26. NEC Launches AI Agent Service in Japan to Automate Procurement Negotiations Using AI, https://www.acnnewswire.com/press-release/english/104045/nec-launches-ai-agent-service-in-japan-to-automate-procurement-negotiations-using-ai
27. (PDF) The Automated but Risky Game: Modeling Agent-to-Agent Negotiations and

Transactions in Consumer Markets - ResearchGate,
https://www.researchgate.net/publication/392334185_The_Automated_but_Risky_Game_Modeling_Agent-to-Agent_Negotiations_and_Transactions_in_Consumer_Markets
28. GAIA: A General Agency Interaction Architecture for LLM-Human B2B Negotiation & Screening - arXiv, https://arxiv.org/html/2511.06262v1
29. GAIA: A General Agency Interaction Architecture for LLM ... - arXiv,
https://arxiv.org/pdf/2511.06262
30. Automated Procurement & Sourcing with Agentic AI - NexaStack,
https://www.nexastack.ai/use-cases/procurement-sourcing-agentic-ai
31. Agentic AI in Procurement: Transforming Decision-Making at Scale - Sievo,
https://sievo.com/blog/agentic-ai-in-procurement-transforming-decision-making-at-scale
32. An LLM-based multi-agent framework for agile effort estimation - arXiv,
https://www.arxiv.org/pdf/2509.14483
33. AI based Multiagent Approach for Requirements Elicitation and Analysis - arXiv,
https://arxiv.org/html/2409.00038v1
34. From LLMs to LLM-based Agents for Software Engineering: A Survey of Current, Challenges and Future - arXiv, https://arxiv.org/html/2408.02479v2
35. Best AI Agent for RFP Automation and Proposals - Smartcat,
https://www.smartcat.com/ai-agents/rfp-automation/
36. TRiSM for Agentic AI: A Review of Trust, Risk, and Security Management in LLM-based Agentic Multi-Agent Systems - arXiv, https://arxiv.org/html/2506.04133v2
37. AI Agentic workflows and Enterprise APIs: Adapting API architectures for the age of AI agents - arXiv, https://arxiv.org/pdf/2502.17443
38. Agent Exchange: Shaping the Future of AI Agent Economics - arXiv,
https://arxiv.org/html/2507.03904v1
39. A Decentralized Trust Insurance Mechanism for Agentic Economy - arXiv,
https://arxiv.org/html/2512.08737v1
40. Agentic AI vs. generative AI: The core differences | Thomson Reuters,
https://www.thomsonreuters.com/en/insights/articles/agentic-ai-vs-generative-ai-the-core-differences
41. AgentX: Towards Orchestrating Robust Agentic Workflow ... - arXiv,
https://arxiv.org/abs/2509.07595
42. Agentic AI Development Cost in 2026: A Complete Guide - Biz4Group LLC,
https://www.biz4group.com/blog/agentic-ai-development-cost
43. How AI Agents Automate Construction Bid Package Assembly for Estimating Directors,
https://datagrid.com/blog/ai-agents-automate-construction-bid-package-assembly-estimating-directors
44. Agentic generative AI for context-aware outlier removal and historical cost optimization in construction - Frontiers,
https://www.frontiersin.org/journals/built-environment/articles/10.3389/fbuil.2025.1678156/full
45. LiRA: A Multi-Agent Framework for Reliable and Readable Literature Review Generation,
https://arxiv.org/html/2510.05138v2