SMTP/IMAP as a Transport Layer for Agent-to-Agent Communication

Rob Murphy
rob.murphy@couchbase.com
April 2025

Executive Summary

As organizations deploy artificial intelligence agents, they face a critical challenge that will soon impede their progress: enabling secure, reliable communication between these agents across organizational boundaries. Despite significant advances in individual agent capabilities, the infrastructure to support sophisticated cross-boundary agent interactions remains largely unsolved.

This whitepaper presents a strategic approach that leverages the established SMTP (Simple Mail Transfer Protocol) and IMAP (Internet Message Access Protocol) as transport layers for agent-to-agent communication. Rather than developing entirely new protocols or waiting for emerging standards to mature, this solution builds on the email infrastructure already deployed in every organization worldwide.

The approach delivers five key advantages:

1. **Immediate deployment capability** using existing infrastructure that already spans organizational boundaries
2. **Enterprise-ready security** through mature, well-understood mechanisms including TLS, DKIM, and SPF
3. **Inherent conversation threading** that enables sophisticated agentic memory across security domains
4. **Store-and-forward architecture** ensuring reliable delivery even with intermittent connectivity
5. **Protocol-agnostic design** that can carry any payload format, ensuring compatibility with evolving standards

By separating transport concerns from payload formats, organizations can deploy agent communication capabilities today using existing infrastructure while maintaining flexibility to adopt evolving agent communication standards as they mature. This approach works both within and across organizational boundaries, preserves critical conversational context, and scales with enterprise needs—all without deploying new protocols or infrastructure.

1. Introduction: The Agentic Communication Challenge

The Emergence of Agentic Systems

Organizations are increasingly deploying artificial intelligence systems that go beyond simple automation tools. These systems, often called "agents" or "agentic systems," are characterized by their ability to:

- Execute tasks autonomously with limited human oversight
- Maintain awareness of context across multiple interactions
- Make decisions based on complex criteria and evaluation
- Collaborate with humans and other agents to accomplish goals

As these agentic systems proliferate within organizations, they inevitably need to communicate with each other—sharing information, coordinating activities, and maintaining contextual awareness across interactions. This communication need extends beyond organizational boundaries as businesses collaborate with partners, suppliers, customers, and service providers.

Key Agentic Concepts

Several core concepts are essential to understanding agent communication requirements:

Agent: An autonomous software entity capable of perceiving its environment, making decisions, and taking actions to achieve specific goals. In modern implementations, agents typically leverage artificial intelligence capabilities such as large language models, planning systems, and specialized reasoning tools.

Agentic System: A computational system exhibiting agency—the capacity to act independently, make decisions, and affect its environment based on internal goals and external inputs. Modern agentic systems typically combine multiple AI technologies to achieve more sophisticated capabilities than any single model or algorithm could provide.

Agentic Network: A collection of interconnected agents that collaborate to accomplish tasks requiring coordination across multiple specialized capabilities. These networks may span internal systems or cross organizational boundaries.

Agentic Memory: The capability of agents to maintain, access, and utilize contextual information and interaction history over time. Effective agentic memory enables more coherent multi-turn interactions, reduces redundant communication, and allows agents to build upon previous exchanges rather than starting anew each time.

The Cross-Boundary Communication Challenge

As organizations deploy increasingly sophisticated agents, they face a significant barrier: enabling secure, reliable communication between these agents across organizational boundaries. This challenge encompasses several interconnected problems:

1. **Security boundary traversal**: Most organizations maintain strict security perimeters that limit external communications. Enabling agent interactions across these boundaries

requires traversing firewalls, security gateways, and data loss prevention systems.

2. **Protocol standardization**: In the absence of universally adopted standards, agents developed by different organizations may use incompatible communication methods, limiting their ability to collaborate effectively.

3. **Contextual persistence**: Sophisticated agent interactions require maintenance of conversational context and memory across multiple exchanges, particularly when these span extended timeframes.

4. **Infrastructure deployment**: Proposed solutions often require deployment of specialized infrastructure components, creating significant implementation barriers.

Current approaches to agent communication predominantly rely on custom API integrations, proprietary messaging formats, or specialized protocols that struggle with organizational boundary crossings. These approaches typically require significant new infrastructure deployment, complex security configurations, and substantial agreement on standards before meaningful cross-organizational agent collaboration can occur.

2. The Current Landscape of Agent Communication

2.1 Predominant Approaches and Their Limitations

Current approaches to enabling agent-to-agent communication fall into several categories, each with significant limitations for cross-organizational scenarios:

Custom API Integrations: Many organizations implement specialized REST or GraphQL APIs for agent communication. While effective within organizational boundaries, these approaches typically struggle with cross-boundary communication due to firewall restrictions, authentication challenges, and the need for custom integration work between each pair of communicating organizations.

Message Queuing Systems: Technologies like Apache Kafka, RabbitMQ, and AWS SQS provide robust message delivery capabilities but are primarily designed for internal communication within controlled environments. Extending these systems across organizational boundaries introduces significant security, networking, and operational complexity.

Webhook Mechanisms: Some implementations use webhook callbacks for agent notification and data exchange. These approaches require exposing public endpoints, creating potential security vulnerabilities and requiring complex authentication mechanisms to prevent unauthorized access.

2.2 Emerging Agent Communication Standards

Several standards are emerging to address agent communication, though most remain in early stages of development and adoption:

Protocol	Sponsoring Organization	Release	Primary Focus	Cross-Org Support
Model Context Protocol (MCP)	Anthropic	Nov 2024	Tool access and context management	Limited
Agent-to-Agent (A2A)	Google	April 2025	Task exchange and capability discovery	Partial
LLM Messaging Format (LMF)	OpenAI	March 2025	Prompt standardization and memory	Minimal
Agent Communication Language (ACL)	AI Alliance	Feb 2025	Semantic understanding between agents	Experimental

As of April 2025, the Model Context Protocol (MCP) appears to be emerging as the industry's leading standard, particularly following Google's announcement of official support at Google Next 2025. This significant endorsement positions MCP as the protocol with the greatest momentum for internal agent communication, though its cross-organizational capabilities remain limited.

An interesting development is the potential convergence of A2A operating over MCP as a foundation. This combination would leverage MCP's tool access and context management capabilities with A2A's structured task exchange and capability discovery features. However, even this promising combination doesn't fully address the cross-organizational boundary challenge that remains a critical barrier to effective agentic networks.

These emerging standards face several adoption challenges in enterprise environments:

1. They typically assume direct network connectivity between agents, which rarely exists across organizational boundaries
2. Many require specialized infrastructure not yet widely deployed in enterprise environments
3. They face the classic standards adoption challenge—requiring critical mass before delivering full value
4. Most don't inherently address organizational boundary crossing

Rather than waiting for these standards to mature or building complex custom solutions, organizations need an approach that delivers immediate value while maintaining compatibility with future developments. This is where SMTP and IMAP offer a uniquely compelling solution that can work alongside and complement these emerging standards.

3. SMTP/IMAP: The Ideal Foundation for Agent Communication

3.1 Protocol Evolution and Enterprise Readiness

The Simple Mail Transfer Protocol (SMTP) emerged in the early 1980s and has evolved through numerous updates to accommodate changing security requirements, increased message complexity, internationalization, and exponential growth in usage. The Internet Message Access Protocol (IMAP) evolved as the complementary standard for message retrieval and management.

Together, these protocols form the backbone of business communication worldwide, handling billions of messages daily across organizational boundaries. Their longevity and continued relevance speak to their fundamental design soundness and adaptability in the face of evolving technical landscapes.

3.2 Key Capabilities for Agent Communication

SMTP and IMAP have developed robust capabilities that make them uniquely suitable for agent-to-agent communication both within and across organizational boundaries:

Native Addressing Scheme: The email addressing format (agent@organization.domain) maps naturally to agent identification, providing a built-in namespace without requiring centralized registries. This addressing scheme enables sophisticated routing patterns through mechanisms like carbon copying (CC) and blind carbon copying (BCC), which can facilitate multi-agent coordination scenarios.

Security Boundary Traversal: SMTP is explicitly designed to traverse organizational boundaries securely. Transport Layer Security (TLS) provides encryption in transit, while authentication mechanisms like DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF) provide sender verification. These mechanisms are well-understood by enterprise security teams and typically already configured for cross-organizational communication.

Store-and-Forward Architecture: The store-and-forward design of SMTP provides inherent reliability even when receiving systems are temporarily unavailable—a critical consideration for asynchronous agent operations that may span extended timeframes. Modern SMTP implementations include comprehensive delivery receipts, retry logic, and message queuing systems that ensure communication reliability even in degraded network conditions.

Content Flexibility: Modern email systems efficiently handle diverse content types through MIME (Multipurpose Internet Mail Extensions), supporting structured data formats, images, documents, and other specialized content types that agents may need to exchange. This flexibility allows agents to share everything from simple text messages to complex structured data or binary content.

MIME is particularly important for agent-to-agent communication as it provides standardized encoding methods that all major programming languages support. Rather than requiring custom parsing tools, developers can leverage mature MIME libraries to handle the packaging and extraction of structured data payloads like JSON. This established standard significantly reduces implementation complexity while ensuring reliable data exchange across organizational boundaries.

3.3 SMTP/IMAP as a Critical Component of Agentic Memory

Perhaps the most significant alignment between SMTP/IMAP and agentic networks is how email's conversation threading provides a foundation for agentic memory. Email systems maintain conversational context through several mechanisms that map directly to the memory needs of agent interactions:

Thread-Based Contextual Persistence: Email's threading capabilities establish and maintain relationships between messages through subject lines and headers like In-Reply-To and References. These mechanisms create a coherent conversation structure spanning multiple exchanges, enabling agents to understand the context of each message within the broader interaction history.

However, it's important to note that while email clients implement threading, they do so with conventions rather than rigid standards. This implementation variance means that reliable threading across different systems cannot be guaranteed by the email infrastructure alone. Since threading is critical to agentic memory, sophisticated agent implementations must maintain their own threading logic and conversation history. This typically involves including structured conversation history within the message payload itself, ensuring consistent context maintenance regardless of the email systems involved in transport.

Distributed Agentic Memory Architecture: Unlike centralized agentic memory systems, email's threading operates across organizational boundaries, enabling distributed agents to maintain shared context without requiring access to each other's internal systems. This distributed approach to memory aligns perfectly with the distributed nature of cross-organizational agent networks.

Temporal Resilience: Email conversations can span minutes, hours, days, or even months while maintaining complete contextual history. This temporal flexibility is essential for complex agent collaborations that may involve extended processing times or deliberate delays between interaction steps.

Metadata Persistence: Email systems preserve rich metadata about timing, participants, routing, and message relationships. This metadata provides critical contextual information for agent reasoning and decision-making, enabling more sophisticated interactions than would be possible with isolated messages.

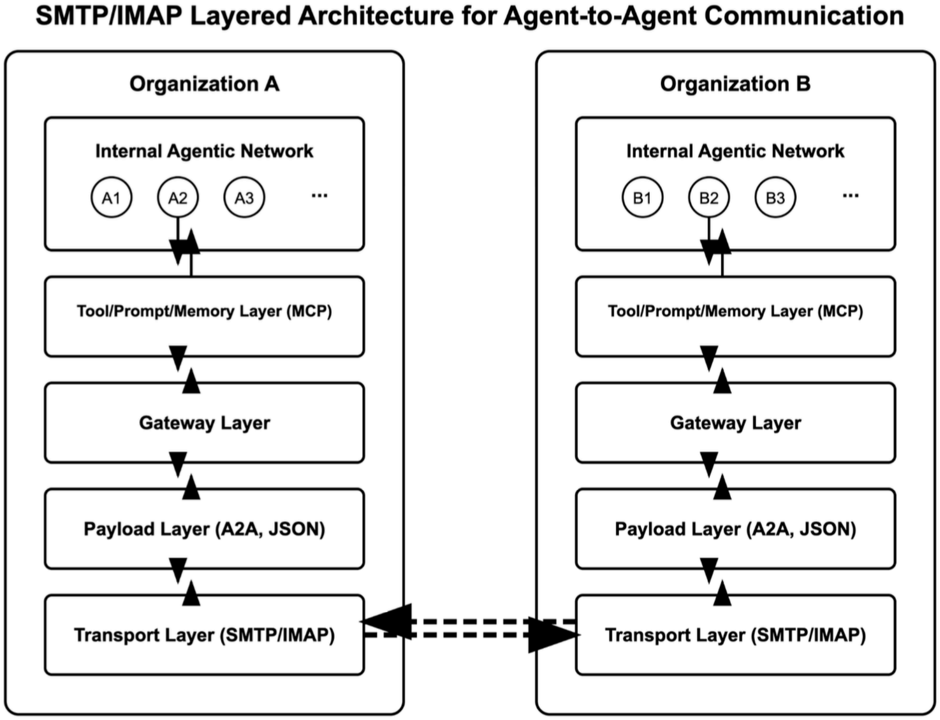
Email's inherent threading model serves as a critical component of agentic memory across organizational boundaries. Just as human email conversations preserve context and enable collaborative workflows spanning days or weeks, SMTP/IMAP provides agents with persistent memory without requiring continuous connection or specialized context-sharing mechanisms.

While many newer communication protocols focus primarily on optimizing for real-time interactions or specialized data formats, they often lack these inherent threading and contextual persistence mechanisms that make SMTP/IMAP so valuable for agentic memory.

4. A Layered Architecture for Agent Communication

4.1 Architectural Overview

To leverage the strengths of existing protocols while accommodating emerging agent communication standards, this whitepaper proposes a layered architecture that separates transport concerns from payload formatting and processing.


SMTP/IMAP Layered Architecture for Agent-to-Agent Communication
Organization A
Internal Agentic Network
A1
A2
A3
...
Tool/Prompt/Memory Layer (MCP)
Gateway Layer
Payload Layer (A2A, JSON)
Transport Layer (SMTP/IMAP)
Organization B
Internal Agentic Network
B1
B2
B3
...
Tool/Prompt/Memory Layer (MCP)
Gateway Layer
Payload Layer (A2A, JSON)
Transport Layer (SMTP/IMAP)

The architecture consists of four distinct layers:

Transport Layer (SMTP/IMAP): Provides reliable message delivery across organizational boundaries with security, routing, and threading capabilities.

Payload Layer (Structured Agent Communication): Encodes agent messages with task descriptions, capability advertisements, and response expectations in standardized formats.

Gateway Layer (Boundary Translation): Specialized agents at organizational boundaries translate between external SMTP/IMAP communication and internal agent protocols.

Tool/Prompt/Memory Layer (Internal Integration): Within organizational boundaries, agents access internal resources through protocols optimized for performance and security.

This layered approach provides several key advantages:

1. Separation of concerns between transport and payload, allowing independent evolution of each layer
2. Flexible implementation supporting various levels of sophistication and integration
3. Protocol independence enabling compatibility with emerging standards

4. Progressive adoption allowing organizations to implement capabilities incrementally

4.2 Layer Details and Implementation

4.2.1 Transport Layer: SMTP/IMAP

SMTP serves as the primary transport mechanism for delivering agent messages across organizational boundaries. Its store-and-forward architecture, security features, and universal adoption make it ideal for this role. IMAP provides the complementary capability for agents to retrieve, process, and maintain message state.

Message routing leverages the standard email addressing scheme (agent@organization.domain), providing a familiar and universally understood addressing model. Security at the transport layer builds on established TLS implementations for encryption in transit, while sender verification leverages existing DKIM and SPF mechanisms.

Conversation threading utilizes standard email headers, creating a consistent model for relating messages to each other and maintaining conversational context. This threading capability provides the foundation for stateful agent interactions without requiring custom development.

4.2.2 Payload Layer: Structured Agent Communication

Within the message body, structured payload formats enable sophisticated agent interactions. JSON format provides a natural foundation for these payloads, offering a balance between human readability and machine processability. For agent communication payloads, JSON structures typically include:

- Task descriptions that specify the requested actions and their parameters
- Structured data representation for both inputs and outputs
- Capability advertisements that communicate agent functions
- Response specifications that set expectations for reply formats
- Error handling information for graceful failure management
- Conversation history and threading information to maintain context

While emerging standards like A2A focus on payload structure and semantics, implementing agents must address the gap between email's threading conventions and the requirements for reliable agentic memory. By including conversation history and explicit message relationships within the payload itself, agents take direct responsibility for context maintenance rather than relying on convention based email threading capabilities. This approach ensures consistent context preservation across organizational boundaries regardless of the email clients or servers involved.

4.2.3 Example Payload Formats

Basic JSON Agent Message:

```
{
  "message_id": "unique_message_id",
  "exchanges": [
    {
      "sender": "agent@example.com",
      "timestamp": "2025-04-22T18:26:58",
      "message_id": "unique_message_id",
      "recipients": ["recipient@example.com"],
      "content": {
        "action": "request",
        "message": "The actual message content",
        "expected_format": "json",
        "response_targets": ["agent@example.com"]
      }
    }
  ]
}
```

Google A2A Protocol Example:

```
{
  "jsonrpc": "2.0",
  "id": "req-2025-04-18-123456",
  "method": "tasks/send",
  "params": {
    "id": "task-2025-04-18-789012",
    "message": {
      "role": "user",
      "parts": [
        {
          "type": "text",
          "text": "Analyze correlation patterns in the attached dataset"
        },
        {
          "type": "data",
          "data": {
            "dataAnalysisParams": {
              "confidenceLevel": 0.95,
              "featureSelection": ["price", "volume", "volatility"],
              "timeframeStart": "2025-01-01",
              "timeframeEnd": "2025-03-31"
            }
          }
        }
      ]
    },
    "metadata": {
      "conversationId": "conv-2025-04-18-789012",
      "sender": {
        "id": "research-agent@companyA.com",
        "name": "Research Analysis Agent"
      },
      "recipient": {
        "id": "data-science-agent@companyB.com",
        "name": "Data Science Agent"
      },
      "inReplyTo": "task-2025-04-18-123455",
      "references": ["task-2025-04-18-123454", "task-2025-04-18-123455"]
    }
  }
}
```

The payload layer operates independently from the transport layer, allowing organizations to adopt and evolve their agent communication standards without modifying the underlying transport mechanisms.

4.2.4 Gateway Layer: Boundary Translation

At organizational boundaries, specialized gateway agents provide translation services between external SMTP/IMAP communication and internal agent communication protocols. These gateways serve as bridging points between security domains, protocol formats, and capability registries.

Gateway agents handle protocol translation between external message formats and internal agent communication standards. This translation function allows organizations to maintain different internal and external protocols while ensuring seamless communication across boundaries. Security filtering and validation occur at these gateway points, providing a clear boundary for policy enforcement and threat mitigation.

Capability registry and discovery functions within the gateway layer allow external agents to understand available services without exposing internal implementation details. This discovery mechanism enables dynamic collaboration patterns while maintaining appropriate security boundaries.

4.2.5 Tool/Prompt/Memory Layer: Internal Integration

Within organizational boundaries, internal protocols provide agents with access to data sources, function execution capabilities, and context enhancement services. The gateway layer bridges these internal capabilities with the external SMTP/IMAP communication, allowing controlled access to organizational resources while maintaining security boundaries.

By separating internal tool access from external communication protocols, organizations maintain flexibility in their internal agent architecture while providing standardized external interfaces. This separation of concerns ensures that changes to internal systems don't disrupt external communication patterns, and vice versa.

5. Implementation Approach

Rather than presenting rigid implementation phases, this section outlines strategic approaches to adoption based on organizational needs and maturity. Organizations can integrate SMTP/IMAP-based agent communication into their existing AI initiatives, tailoring the implementation approach to their specific circumstances.

5.1 Incremental Capability Enhancement

Organizations can progressively enhance their agent communication capabilities, realizing value at each stage:

Basic Communication Capability: Enable agents to send/receive structured messages via email using existing SMTP and IMAP libraries. Implement simple JSON payloads for initial use cases and leverage existing email security infrastructure. This provides immediate cross-boundary communication capabilities with minimal investment.

Enhanced Protocol Structures: Adopt more sophisticated payload formats aligned with emerging standards like A2A. Implement task lifecycle management, structured error handling, and support for diverse content types. These enhancements enable more complex agent interactions while maintaining compatibility with the basic transport foundation.

Gateway Integration: Deploy specialized components at organizational boundaries that translate between external SMTP/IMAP communication and internal agent protocols. Implement security filtering, capability discovery, and session management for extended interactions. This approach enables controlled access to internal capabilities while maintaining appropriate security boundaries.

Advanced Capabilities: Implement sophisticated features such as dynamic capability discovery, multi-agent workflow orchestration, and performance optimizations for time-sensitive operations. These capabilities build on the established foundation to enable complex agent ecosystems spanning multiple organizations.

5.2 Implementation by Organizational Maturity

Different organizations will approach agent communication based on their existing AI maturity:

AI Exploration Stage: Organizations just beginning their AI journey can implement basic agent email capabilities using existing libraries and simple JSON payloads. This approach enables immediate cross-boundary agent communication without specialized expertise or infrastructure investment.

Active AI Adoption: Organizations with established internal agent capabilities can implement gateway components that translate between internal protocols and SMTP/IMAP. This preserves investments in internal agent infrastructure while enabling cross-organizational collaboration through the universal email transport layer.

Advanced AI Deployment: Organizations operating sophisticated agent networks can implement comprehensive capability registries, advanced security models, and specialized protocol translations to enable federated collaboration capabilities with multiple external partners.

5.3 Technical Implementation Considerations

Several technical aspects require consideration during implementation:

Message Format Options: Organizations can choose between embedded JSON in the email body (simplest), email attachments (for large data payloads), or custom email headers (for routing information). Most implementations will use embedded JSON for its simplicity and readability while leveraging attachments for binary or large data payloads.

Authentication and Authorization: Implementing robust security requires multiple layers, including SMTP authentication mechanisms (DKIM, SPF), application-level authentication within payloads, gateway-based authorization for cross-boundary requests, and capability-based permissions for specific functions.

Error Handling and Reliability: Comprehensive error management combines SMTP's inherent reliability with structured error responses in payloads, timeout handling for long-running operations, and fallback mechanisms when automated processing fails.

Security Implementation: Security measures should include transport encryption through TLS, payload encryption for sensitive data, content validation at gateway boundaries, rate limiting to prevent abuse, and comprehensive audit logging of all cross-boundary communications.

Performance Optimization: While SMTP was not designed primarily for real-time communication, performance can be optimized through connection pooling, asynchronous processing models, optimized polling intervals, and hybrid approaches for time-sensitive operations.

6. Business Value and Transformation Potential

Implementing SMTP/IMAP as an agent communication transport delivers significant business value through accelerated capabilities and controlled costs.

6.1 Value Acceleration

Immediate Cross-Boundary Capabilities: Deploy agent collaboration across organizational boundaries in weeks rather than months. This acceleration eliminates the typical delays associated with cross-organization API integration projects.

Extended Automation Reach: Extend intelligent automation beyond organizational walls to include partners, suppliers, and customers in automated workflows. This expanded reach creates opportunities for end-to-end process transformation.

Enhanced Agentic Networks: Create more sophisticated multi-agent collaborations that span security domains while maintaining conversational context. These enhanced networks can tackle complex tasks requiring diverse specialized capabilities.

Universal Connectivity: Connect with any organization using standard email—which means virtually every business entity worldwide. This universal reach ensures your agent collaboration strategy isn't limited to only those partners with specialized technical capabilities.

Future-Ready Architecture: Build agent communication capabilities that evolve alongside emerging standards without requiring infrastructure replacement. This approach gives you flexibility to adopt new agent communication protocols when they mature while realizing value immediately.

6.2 Cost Control

Zero New Infrastructure Investment: Leverage existing email systems already deployed, secured, and maintained within your organization. This eliminates the capital expenditure typically associated with new communication protocols or specialized integration platforms.

Reduced Security and Compliance Costs: Build on email security models already approved by your security team and familiar to compliance officers. This significantly reduces the review cycles and compliance documentation required for new cross-boundary communication channels.

Minimized Partner Coordination Costs: Eliminate the need for coordinated deployment schedules with external partners. Your organization can implement capabilities at your pace, without waiting for partners to deploy specialized infrastructure or adopt emerging standards.

Lower Integration Engineering Costs: Use familiar protocols supported by existing libraries in all major programming languages. This reduces custom development work and leverages skills your technical team already possesses.

Decreased Support and Operational Burden: Build on systems with established monitoring, troubleshooting, and support processes rather than introducing new operational requirements. This minimizes the increased operational burden typically associated with new technologies.

6.3 Transformation Potential

The SMTP/IMAP approach to agent communication enables new business models and capabilities across industries, particularly for enterprises facing the practical challenges of AI modernization outside of Silicon Valley's innovation bubble. While emerging standards from major technology companies provide valuable frameworks, enterprises must balance innovation with existing infrastructure, security requirements, and governance processes. This approach bridges that gap by leveraging familiar technologies while enabling transformative capabilities:

Financial Services: Enable multi-institution workflows for loan syndication, compliance reporting, and risk assessment that maintain regulatory compliance while enabling greater automation.

Manufacturing: Create intelligent supply chain collaboration that spans tiers of suppliers, enabling predictive procurement, automated exception handling, and coordinated production planning.

Healthcare: Implement cross-organization care coordination and research collaboration within strict privacy boundaries, enabling better patient outcomes while maintaining regulatory compliance.

7. Use Cases and Examples

The proposed architecture enables a wide range of cross-organizational agent collaboration scenarios that would be difficult to implement with traditional API-based approaches.

7.1 Cross-Organizational Research Collaboration

In research collaboration scenarios, specialized analysis capabilities often exist in different organizations, requiring secure and contextual information exchange across boundaries.

A research agent in Company A might identify a need for specialized analysis available only from Company B's data science capabilities. The agent formats a structured task request according to A2A conventions and sends it via SMTP to Company B's gateway address.

Upon receipt, Company B's gateway extracts the A2A payload from the email, validates the request against security policies, and routes it to the appropriate internal research agent based on the requested capabilities. This internal agent performs the analysis using proprietary tools and data.

The analysis results are formatted as structured artifacts within an A2A response payload, which the gateway agent then transmits back to Company A via SMTP. Throughout this process, the email thread maintains consistent message IDs and references, preserving the conversational context across organizational boundaries.

7.2 Supply Chain Coordination

Supply chain operations require complex coordination between multiple organizations, often with time-sensitive communication needs and structured data exchanges.

A manufacturer's inventory management agent might detect low stock levels for critical components and initiate a restocking process. The agent formats a structured inventory query with specific part numbers, quantity requirements, and delivery timeframes, then sends this query via SMTP to the supplier's gateway agent.

The supplier's gateway routes this request to specialized fulfillment agents based on product categories. These fulfillment agents check availability, pricing, and delivery options through internal inventory systems.

The response follows the reverse path, with structured data on availability, pricing, and delivery options enclosed in the email body. This information enables the manufacturer's agent to make intelligent restocking decisions based on current supply conditions.

If appropriate, the agents can negotiate optimal order timing and quantities through multiple message exchanges, with each exchange building on the context established in previous messages.

7.3 Multi-Organization Business Processes

Many business processes span multiple organizations, requiring secure, traceable workflows with appropriate data sharing and decision points.

A loan approval process might involve multiple financial institutions, each responsible for different aspects of underwriting and risk assessment. The process begins when a loan origination agent sends a structured application package to a credit verification agent at a specialized bureau.

After credit analysis, the bureau's agent forwards structured results to an underwriting agent at a different institution, adding its verification data to the evolving application record. This underwriting agent may request additional verification from identity confirmation agents at yet another organization.

Each step in this multi-organization workflow maintains consistent context through email threading, with each message adding new information while preserving the history of previous assessments. This creates a complete audit trail spanning all participating organizations, without requiring a centralized workflow system.

8. Conclusion and Next Steps

8.1 Summary

SMTP and IMAP provide a robust foundation for agent-to-agent communication both within and across organizational boundaries. By leveraging these established protocols as a transport layer for structured agent interaction payloads, organizations can enable sophisticated agent collaboration without deploying new infrastructure or waiting for emerging standards to mature.

This approach delivers five key advantages:

1. **Immediate deployment capability** using existing infrastructure that already spans organizational boundaries
2. **Enterprise-ready security** through mature, well-understood mechanisms including TLS, DKIM, and SPF

3. **Inherent conversation threading** that enables sophisticated agentic memory across security domains
4. **Store-and-forward architecture** ensuring reliable delivery even with intermittent connectivity
5. **Protocol-agnostic design** that can carry any payload format, ensuring compatibility with evolving standards

The layered architecture separating transport concerns from payload semantics enables organizations to begin implementing agent communication capabilities today while maintaining flexibility to adopt emerging agent communication standards as they mature.

8.2 Recommended Next Steps

To move forward with this approach, organizations should consider these next steps:

1. **Identify High-Value Use Cases**: Select initial scenarios requiring cross-organizational agent communication that would deliver significant business value. Focus on use cases where email is already part of the workflow for a seamless transition.

2. **Develop Proof-of-Concept**: Implement a simple prototype using basic SMTP/IMAP libraries and straightforward JSON payloads. Test the concept with willing partners to demonstrate cross-boundary capabilities.

3. **Create Technical Standards**: Define internal standards for payload formats, security requirements, and gateway implementation. Ensure these standards balance immediate needs with future compatibility.

4. **Integrate with AI Roadmap**: Position SMTP/IMAP-based agent communication as a component of your broader AI and automation strategy. Connect this capability to other AI initiatives to maximize value.

Organizations that implement this approach can establish a strong foundation for cross-boundary agent collaboration while maintaining control over infrastructure investments and technological risk. By leveraging established protocols while preserving future flexibility, your organization can accelerate AI adoption across organizational boundaries while preparing for the evolving future of agent communication.

For more detailed technical information or implementation guidance, contact the author.