WG Working Group
Internet-Draft
Intended status: Informational
Expires: 15 August 2026

Y. Cui
Tsinghua University

11 February 2026

Cross-Domain Interoperability Framework for AI Agent Collaboration
draft-cui-dmsc-agent-cdi-00

Abstract

This document defines a framework for enabling seamless cross-domain interoperability among AI agents operating across different networks, administrative domains, and heterogeneous platforms. The framework addresses the challenges of identity federation, trust establishment, policy harmonization, and secure communication that arise when AI agents from distinct administrative realms need to collaborate on shared tasks. It specifies mechanisms for agent discovery, capability negotiation, trust delegation, and federated policy enforcement, enabling scalable and secure multi-domain AI collaboration without requiring centralized control.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-cui-dmsc-agent-cdi/.

Discussion of this document takes place on the WG Working Group mailing list (mailto:WG@example.com), which is archived at https://example.com/WG.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 15 August 2026.

Table of Contents

1. Introduction 4
2. Conventions and Definitions 5
3. Terminology 5
4. Cross-Domain Interoperability Framework 6
4.1. Overview 6
4.2. *Architecture Components* 6
4.2.1. Interoperability Gateway 7
4.2.2. Agent Registry and Directory 7
4.2.3. Cross-Domain Agent Runtime 8
5. Trust Establishment Framework 8
5.1. Trust Model 8
5.2. Trust Establishment Protocols 9
5.2.1. Domain Federation Protocol 9
5.2.2. Agent Delegation Protocol 10
6. Identity and Access Management 10
6.1. Federated Identity Model 10
6.1.1. Agent Identifiers 10
6.1.2. Credential Types 11
6.2. Access Control Framework 11
6.2.1. Multi-Domain Policy Model 11
6.2.2. Policy Negotiation Protocol 11
6.2.3. Attribute-Based Access Control (ABAC) 12
7. Capability Discovery and Negotiation 12
7.1. Capability Description Language 12
7.2. Discovery Mechanisms 13
7.2.1. Directory-Based Discovery 13
7.2.2. Peer-to-Peer Discovery 14
7.3. Capability Negotiation Protocol 14
8. Communication Protocols 15
8.1. Cross-Domain Message Format 15

8.2. Communication Patterns 16
8.2.1. Direct Agent-to-Agent Communication 16
8.2.2. Gateway-Mediated Communication 17
8.2.3. Store-and-Forward Communication 17
8.3. Protocol Layering 17
9. Security Considerations 18
9.1. Threat Model 18
9.2. Security Controls 18
9.2.1. Authentication and Authorization 18
9.2.2. Data Protection 19
9.2.3. Monitoring and Detection 19
9.3. Security Assurance 19
9.3.1. Trust Verification 19
9.3.2. Audit and Accountability 20
10. Performance and Scalability 20
10.1. Performance Metrics 20
10.2. Scalability Considerations 20
10.2.1. Horizontal Scaling 20
10.2.2. Federation Scaling 21
10.3. Resource Management 21
10.3.1. Computational Resources 21
10.3.2. Network Resources 21
11. Deployment Considerations 22
11.1. Deployment Models 22
11.1.1. Centralized Gateway Model 22
11.1.2. Distributed Gateway Model 22
11.1.3. Hybrid Model 22
11.2. Integration with Existing Systems 23
11.2.1. Legacy Agent Systems 23
11.2.2. Cloud and Edge Deployments 23
11.3. Operational Considerations 23
11.3.1. Monitoring and Management 23
11.3.2. Update and Maintenance 24
12. Use Cases and Examples 24
12.1. Cross-Enterprise Supply Chain Optimization 24
12.2. Smart City Federation 25
12.3. Cross-Border Financial Services Collaboration 25
12.4. Cross-Domain Emergency Response System 26
12.5. Multi-Cloud AI Workload Orchestration 28
12.6. International Research Data Commons 29
12.7. Smart Manufacturing Supply Chain 30
12.8. Cross-Jurisdictional Law Enforcement Collaboration . . . 31
12.9. Global Energy Grid Coordination 32
13. IANA Considerations 33
14. Normative References 33
Acknowledgments 34
Author's Address 34

1. Introduction

The proliferation of AI agents across different organizational boundaries has created a need for mechanisms that enable these agents to collaborate effectively across domains. Traditional agent systems are typically designed to operate within a single administrative domain, where trust relationships, identity management, and policy enforcement are centrally controlled. However, real-world applications increasingly require AI agents to work together across organizational boundaries, such as in supply chain optimization, cross-enterprise workflow automation, smart city coordination, and distributed scientific research.

Cross-domain interoperability for AI agents presents several unique challenges:

1. *Trust Establishment*: How to establish and verify trust between agents from different administrative domains without a central authority.

2. *Identity Federation*: How to manage agent identities and credentials across domains while preserving privacy and sovereignty.

3. *Policy Harmonization*: How to reconcile potentially conflicting policies from different domains to enable collaborative operations.

4. *Capability Discovery*: How agents can discover and understand the capabilities of foreign agents while respecting domain boundaries.

5. *Secure Communication*: How to establish secure communication channels that cross domain boundaries with appropriate security guarantees.

6. *Accountability and Auditing*: How to maintain accountability for actions taken by federated agents while respecting domain autonomy.

This document proposes a framework that addresses these challenges through a combination of federated trust mechanisms, standardized capability descriptions, policy negotiation protocols, and secure communication patterns. The framework is designed to be incrementally deployable and compatible with existing single-domain agent architectures.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] .

3. Terminology

The following terms are defined in this draft

* *MCP (Model Context Protocol)*: A protocol that standardizes tool invocation and data interaction between AI agents and external hosts using JSON-RPC over stdio or HTTP.

* *Host*: The process or service that provides tools or resources to the agent via MCP, typically running as a backend process or plugin.

* *Agent*: An AI entity that uses MCP to invoke external tools to accomplish tasks.

* *Security Enhancement Layer (SEL)*: An external security layer that provides transparent security controls for MCP communication without protocol modification.

* *Client Shield*: The security component deployed on the MCP client (Host) side, responsible for protecting user interests.

* *Server Shield*: The security component deployed on the MCP server (tool provider) side, responsible for protecting service provider interests.

* *Data Backflow*: The transmission of sensitive data from the agent environment to external tools beyond authorized scope or without proper safeguards.

* *Dual Authorization*: An authorization model requiring both platform policy validation and explicit user consent for sensitive operations.

* *Trace ID*: A globally unique identifier used to correlate events across the MCP communication chain for auditing and troubleshooting.

The following abbreviations are used in this document:

* *CDI*: Cross-Domain Interoperability

* *CDA*: Cross-Domain Agent
* *FIM*: Federated Identity Management
* *ABAC*: Attribute-Based Access Control
* *REST*: Representational State Transfer
* *JWT*: JSON Web Token
* *OIDC*: OpenID Connect
* *SAML*: Security Assertion Markup Language
* *DID*: Decentralized Identifier
* *VC*: Verifiable Credential

4. Cross-Domain Interoperability Framework

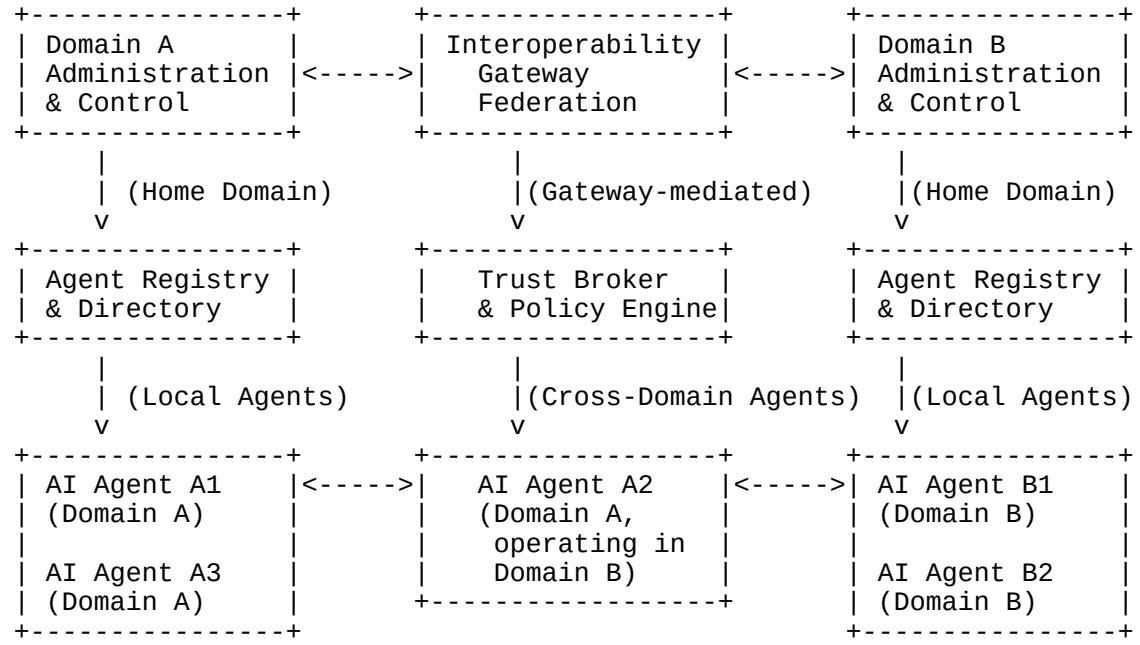
4.1. Overview

The Cross-Domain Interoperability Framework for AI Agents is designed as a federated architecture that enables secure and scalable collaboration between agents from different administrative domains. The framework operates on several key principles:

1. *Domain Autonomy*: Each administrative domain maintains control over its own agents, policies, and resources.
2. *Minimal Trust Assumptions*: Domains establish trust relationships only as needed for specific collaborations.
3. *Graduated Interoperability*: Domains can choose different levels of integration based on trust and requirements.
4. *Auditability*: All cross-domain interactions are logged and auditable by all participating domains.
5. *Fail-Safe Design*: System failures or malicious actions in one domain should not compromise other domains.

4.2. *Architecture Components*

The framework consists of the following core components:



4.2.1. Interoperability Gateway

The Interoperability Gateway serves as the primary interface between domains, providing:

* *Trust Brokerage*: Mediates trust establishment between domains using verifiable credentials and reputation systems.

* *Policy Mediation*: Translates and reconciles policies between domains to enable collaborative operations.

* *Agent Discovery*: Facilitates discovery of agents and their capabilities across domain boundaries.

* *Communication Routing*: Routes messages between agents in different domains with appropriate security.

* *Audit Collection*: Collects and correlates audit trails from cross-domain interactions.

4.2.2. Agent Registry and Directory

Each domain maintains an Agent Registry that provides:

* *Local Agent Directory*: Catalog of agents within the domain with their capabilities and status.

* *Foreign Agent Cache*: Cached information about agents from other domains that have been authorized to operate locally.

* *Capability Repository*: Structured descriptions of agent capabilities using standardized schemas.

* *Policy Repository*: Storage for domain-specific policies governing agent behavior.

4.2.3. Cross-Domain Agent Runtime

Agents operating in foreign domains require enhanced runtime components:

* *Policy Compliance Engine*: Ensures agent behavior complies with both home and foreign domain policies.

* *Context Adaptation Module*: Adapts agent behavior based on the operating context and domain-specific requirements.

* *Secure Communication Module*: Implements cross-domain communication protocols with end-to-end security.

* *Audit Logging Module*: Maintains detailed logs of all cross-domain operations for accountability.

5. Trust Establishment Framework

5.1. Trust Model

The framework employs a multi-layered trust model:

1. *Domain-to-Domain Trust*: Established through bilateral agreements, verifiable credentials, or reputation systems.

2. *Agent-to-Agent Trust*: Derived from domain trust but enhanced with agent-specific credentials and behavioral attestations.

3. *Task-Specific Trust*: Limited trust established for specific collaborative tasks with clearly defined boundaries.

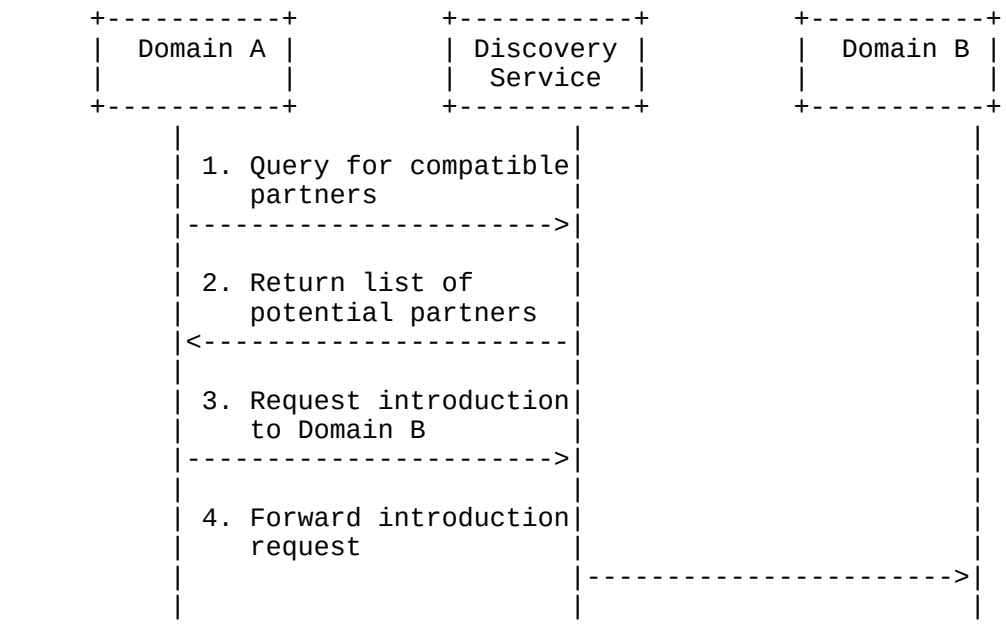
4. *Revocable Trust*: All trust relationships are time-bound and revocable by any participating domain.

5.2. Trust Establishment Protocols

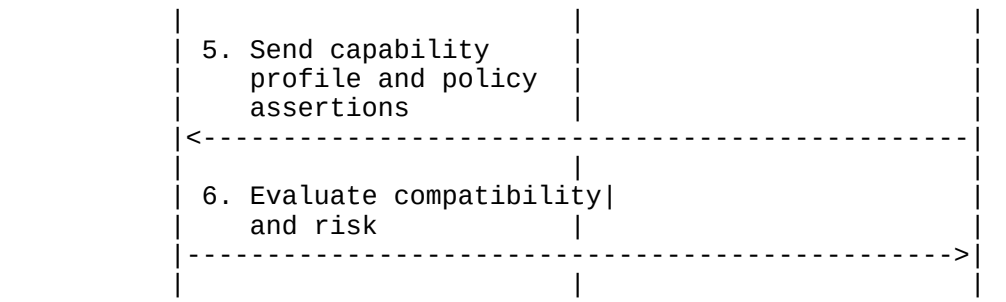
5.2.1. Domain Federation Protocol

Domains establish trust relationships through a multi-phase protocol:

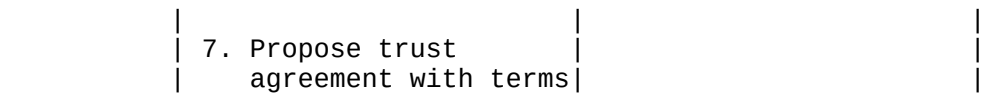
Phase 1: Discovery and Introduction

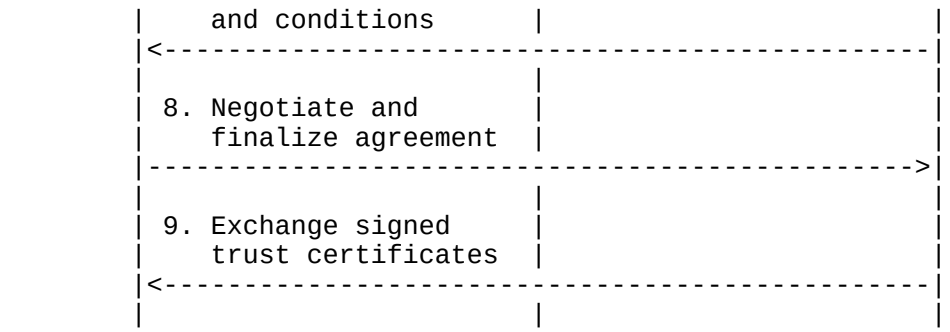


Phase 2: Capability and Policy Exchange



Phase 3: Trust Agreement





5.2.2. Agent Delegation Protocol

Once domains establish trust, agents can be delegated to operate in foreign domains:

1. *Delegation Request*: Home domain requests delegation for specific agents with defined capabilities and constraints.

2. *Policy Alignment*: Foreign domain evaluates the request against its policies and may negotiate constraints.

3. *Credential Issuance*: Home domain issues verifiable credentials to agents for use in the foreign domain.

4. *Local Registration*: Agents register with the foreign domain's Agent Registry using their credentials.

5. *Periodic Revalidation*: Delegated credentials are periodically revalidated and can be revoked at any time.

6. Identity and Access Management

6.1. Federated Identity Model

The framework employs a federated identity model with the following components:

6.1.1. Agent Identifiers

Each agent has multiple identifiers for different contexts:

* *Home Domain Identifier*: Unique identifier within the home domain (e.g., UUID, email-style address).

* *Federated Identifier*: Globally unique identifier for cross-domain recognition (e.g., DID, URN).

* *Session Identifiers*: Temporary identifiers for specific collaborative sessions.

6.1.2. Credential Types

* *Domain Identity Credentials*: Verifiable credentials issued by the home domain attesting to the agent's identity and basic attributes.

* *Capability Credentials*: Credentials attesting to specific agent capabilities and skill levels.

* *Delegation Credentials*: Credentials authorizing an agent to operate in a foreign domain with specific permissions.

* *Task Credentials*: Limited-scope credentials for specific collaborative tasks.

6.2. Access Control Framework

6.2.1. Multi-Domain Policy Model

Access control decisions consider policies from multiple domains:

Access Decision = f(Home Domain Policy, Foreign Domain Policy, Collaborative Task Policy, Dynamic Context Factors)

6.2.2. Policy Negotiation Protocol

When policies conflict, domains negotiate to find acceptable compromises:

1. *Policy Disclosure*: Domains share relevant policy fragments (not necessarily complete policies).

2. *Conflict Detection*: Identify conflicts between policies from different domains.

3. *Resolution Proposal*: Propose policy modifications or exceptions to resolve conflicts.

4. *Agreement Finalization*: Domains agree on a merged policy for the specific collaboration.

5. *Policy Enforcement*: The agreed policy is enforced by all participating domains.

6.2.3. Attribute-Based Access Control (ABAC)

The framework uses ABAC with attributes from multiple sources:

* *Agent Attributes*: Identity, capabilities, reputation score, home domain.
* *Resource Attributes*: Sensitivity, ownership, location, classification.
* *Environmental Attributes*: Time, network location, threat level.
* *Collaboration Attributes*: Task type, participating domains, trust level.

7. Capability Discovery and Negotiation

7.1. Capability Description Language

Agents describe their capabilities using a structured language based on JSON Schema:

```
{
  "agent_capability_profile": {
    "version": "1.0",
    "agent_id": "did:example:agent123",
    "home_domain": "example.com",
    "capabilities": [
      {
        "capability_id": "natural_language_processing",
        "category": "ai_processing",
        "subcategories": ["text_analysis", "sentiment_analysis"],
        "performance_metrics": {
          "accuracy": 0.92,
          "latency_ms": 150,
          "throughput_rps": 100
        },
        "input_formats": ["text/plain", "application/json"],
        "output_formats": ["application/json"],
        "privacy_attributes": {
          "data_retention": "ephemeral",
          "data_sharing": "none"
        },
        "resource_requirements": {
          "compute_units": 2,
          "memory_mb": 512,
          "network_bandwidth_mbps": 10
        }
      }
    ],
    "constraints": {
      "max_concurrent_tasks": 5,
      "supported_domains": ["example.com", "partner.org"],
      "geographic_restrictions": ["US", "EU"],
      "compliance_frameworks": ["GDPR", "HIPAA"]
    }
  }
}
```

7.2. Discovery Mechanisms

7.2.1. Directory-Based Discovery

Domains publish authorized agent capabilities in federated directories:

1. *Local Directory Registration*: Agents register capabilities with their home domain directory.

2. *Federated Directory Sync*: Domain directories synchronize authorized capability information.

3. *Query Interface*: Agents query directories to discover capabilities across domains.

4. *Privacy-Preserving Queries*: Support for capability discovery without revealing specific agent identities when not needed.

7.2.2. Peer-to-Peer Discovery

For dynamic or ad-hoc collaborations:

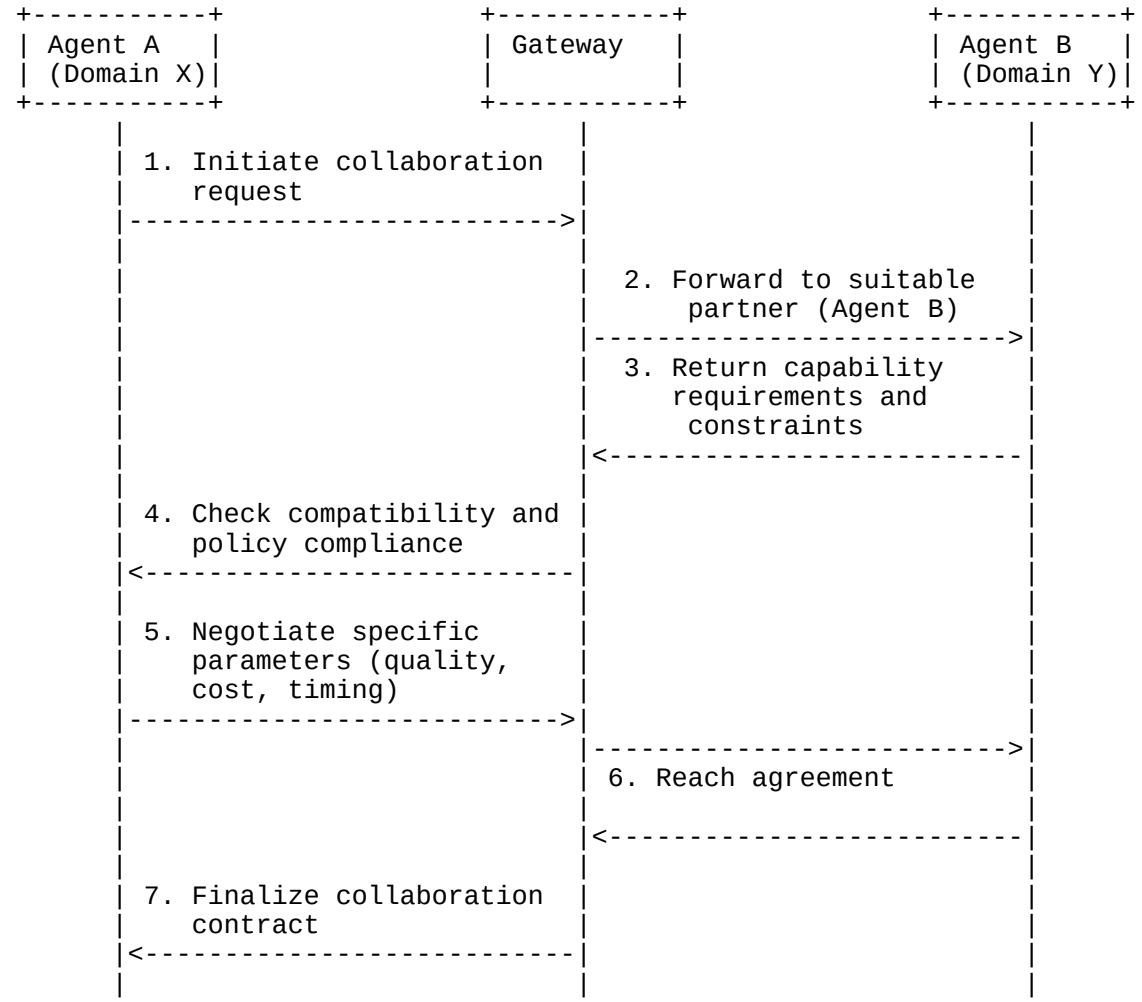
1. *Capability Advertisement*: Agents broadcast capability summaries within trusted networks.

2. *Matchmaking Services*: Third-party services match agents with complementary capabilities.

3. *Reputation-Based Discovery*: Discover agents based on reputation scores from previous collaborations.

7.3. Capability Negotiation Protocol

When agents with complementary capabilities are identified:



8. Communication Protocols

8.1. Cross-Domain Message Format

All cross-domain messages use a standardized envelope format:

```
{
  "message_envelope": {
    "version": "1.0",
    "message_id": "urn:uuid:550e8400-e29b-41d4-a716-446655440000",
    "timestamp": "2024-01-15T10:30:00Z",
    "sender": {
      "agent_id": "did:example:sender123",
      "domain": "example.com",
      "credentials": ["vc:example:credential456"]
    },
    "recipient": {
      "agent_id": "did:example:recipient789",
      "domain": "partner.org"
    },
    "routing": {
      "path": ["gateway.example.com", "gateway.partner.org"],
      "hops": 2,
      "ttl": 3600
    },
    "security": {
      "encryption": "A256GCM",
      "signature": "ES256",
      "integrity_protection": true
    },
    "payload": {
      "type": "task_request",
      "content": "Base64-encoded encrypted payload",
      "content_type": "application/json"
    },
    "metadata": {
      "priority": "normal",
      "response_required": true,
      "audit_trail_id": "audit:example:trail123"
    }
  }
}
```

8.2. Communication Patterns

8.2.1. Direct Agent-to-Agent Communication

For high-trust, high-performance scenarios:

1. *Tunnel Establishment*: Establish secure tunnels through domain gateways.

2. *Peer Authentication*: Mutual authentication using domain-issued credentials.

3. *End-to-End Encryption*: Application-layer encryption independent of transport security.

4. *Heartbeat Monitoring*: Continuous monitoring of connection health and security.

8.2.2. Gateway-Mediated Communication

For enhanced security and policy enforcement:

1. *Message Inspection*: Gateways inspect messages for policy compliance.

2. *Content Transformation*: Gateways may transform content to meet domain requirements.

3. *Quality of Service*: Gateways provide QoS guarantees for cross-domain communication.

4. *Load Balancing*: Gateways distribute traffic across multiple agent instances.

8.2.3. Store-and-Forward Communication

For disconnected or asynchronous operations:

1. *Message Queuing*: Gateways queue messages when recipients are unavailable.

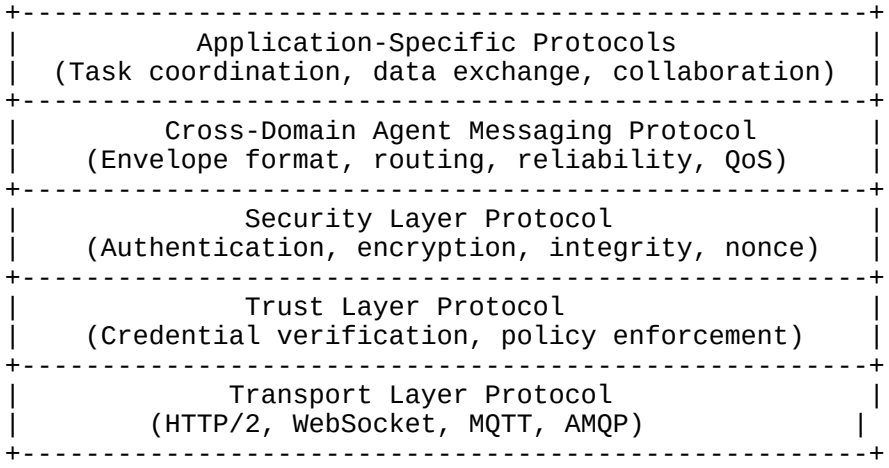
2. *Delivery Guarantees*: Configurable delivery guarantees (at-most-once, at-least-once, exactly-once).

3. *Priority Handling*: Priority-based message scheduling and delivery.

4. *Dead Letter Handling*: Processing of undeliverable messages.

8.3. Protocol Layering

The communication protocol stack for cross-domain agent interaction:



9. Security Considerations

9.1. Threat Model

The framework addresses the following threat vectors:

1. *Impersonation Attacks*: Malicious agents pretending to be legitimate agents from trusted domains.

2. *Policy Bypass Attacks*: Attempts to circumvent domain policies through technical or procedural means.

3. *Data Exfiltration*: Unauthorized transfer of sensitive data across domain boundaries.

4. *Denial of Service*: Attacks targeting the availability of cross-domain collaboration services.

5. *Trust Exploitation*: Abuse of trust relationships to gain unauthorized access or privileges.

6. *Audit Tampering*: Attempts to modify or delete audit logs to conceal malicious activities.

9.2. Security Controls

9.2.1. Authentication and Authorization

* *Multi-Factor Authentication*: Require multiple credentials for sensitive operations.

* *Continuous Authentication*: Re-authenticate agents periodically during long sessions.

* *Context-Aware Authorization*: Adjust permissions based on dynamic risk assessments.

* *Least Privilege Enforcement*: Grant minimal necessary permissions for each task.

9.2.2. Data Protection

* *End-to-End Encryption*: Encrypt sensitive data throughout its lifecycle.

* *Data Minimization*: Transfer only necessary data for each collaboration.

* *Privacy-Preserving Computation*: Use techniques like federated learning and secure multi-party computation.

* *Data Sovereignty*: Respect data sovereignty requirements of each domain.

9.2.3. Monitoring and Detection

* *Behavioral Analytics*: Monitor agent behavior for anomalies indicative of compromise.

* *Cross-Domain Correlation*: Correlate security events across multiple domains.

* *Real-Time Threat Intelligence*: Integrate threat intelligence feeds from all participating domains.

* *Automated Response*: Automated containment and mitigation of detected threats.

9.3. Security Assurance

9.3.1. Trust Verification

* *Credential Chain Validation*: Verify the complete chain of credentials from root authorities.

* *Revocation Checking*: Continuously check credential revocation status.

* *Reputation Scoring*: Use reputation systems to assess agent trustworthiness.

* *Behavioral Attestation*: Use remote attestation to verify agent integrity.

9.3.2. Audit and Accountability

* *Immutable Audit Logs*: Store audit logs in tamper-evident storage.

* *Cross-Domain Audit Correlation*: Enable correlation of audit trails across domains.

* *Non-Repudiation*: Ensure agents cannot deny actions they have taken.

* *Forensic Readiness*: Maintain evidence for potential legal or investigative proceedings.

10. Performance and Scalability

10.1. Performance Metrics

The framework defines the following performance metrics:

1. *Discovery Latency*: Time to discover suitable collaboration partners (target: < 5 seconds for 95% of queries).

2. *Trust Establishment Time*: Time to establish trust between previously unknown domains (target: < 30 seconds).

3. *Message Delivery Latency*: End-to-end latency for cross-domain messages (target: < 100ms for 95% of messages within same region).

4. *Policy Evaluation Time*: Time to evaluate access control policies (target: < 10ms per decision).

5. *Agent Registration Time*: Time for an agent to register in a foreign domain (target: < 2 seconds).

10.2. Scalability Considerations

10.2.1. Horizontal Scaling

* *Stateless Design*: Design components to be stateless where possible for easy scaling.

* *Load Distribution*: Distribute load across multiple instances of each component.

* *Geographic Distribution*: Deploy components in multiple geographic regions.

* *Caching Strategy*: Implement multi-level caching to reduce backend load.

10.2.2. Federation Scaling

* *Hierarchical Federation*: Organize domains into hierarchies to reduce pairwise trust relationships.

* *Transitive Trust*: Support transitive trust with appropriate safeguards.

* *Partial Mesh Networks*: Optimize trust relationships based on collaboration patterns.

* *Domain Clustering*: Group domains with frequent collaborations to reduce overhead.

10.3. Resource Management

10.3.1. Computational Resources

* *Elastic Scaling*: Automatically scale resources based on demand.

* *Resource Quotas*: Enforce resource quotas per domain and per agent.

* *Efficient Algorithms*: Use computationally efficient algorithms for cryptographic operations.

* *Hardware Acceleration*: Leverage hardware acceleration for performance-critical operations.

10.3.2. Network Resources

* *Bandwidth Management*: Prioritize cross-domain traffic based on importance.

* *Connection Pooling*: Reuse connections to reduce connection establishment overhead.

* *Compression*: Compress messages to reduce bandwidth usage.

* *Adaptive Protocols*: Use protocols that adapt to network conditions.

11. Deployment Considerations

11.1. Deployment Models

11.1.1. Centralized Gateway Model

* *Description*: All cross-domain traffic flows through centralized interoperability gateways.
* *Advantages*: Simplified management, consistent policy enforcement, comprehensive auditing.
* *Disadvantages*: Single point of failure, potential performance bottleneck, higher latency.
* *Use Cases*: Highly regulated environments, initial deployments, domains with limited technical resources.

11.1.2. Distributed Gateway Model

* *Description*: Each domain operates its own gateway, with gateways forming a peer-to-peer network.
* *Advantages*: No single point of failure, better performance, preserves domain autonomy.
* *Disadvantages*: More complex to manage, requires inter-gateway trust establishment.
* *Use Cases*: Large-scale deployments, domains with technical expertise, performance-sensitive applications.

11.1.3. Hybrid Model

* *Description*: Combination of centralized and distributed elements based on requirements.
* *Advantages*: Flexibility to optimize for different use cases, incremental deployment.
* *Disadvantages*: Increased complexity, potential consistency challenges.
* *Use Cases*: Evolving deployments, heterogeneous domain environments, transitional architectures.

11.2. Integration with Existing Systems

11.2.1. Legacy Agent Systems

* *Wrapper Components*: Develop wrappers that enable legacy agents to participate in cross-domain collaborations.
* *Protocol Translation*: Translate between legacy protocols and the cross-domain interoperability protocols.
* *Policy Mapping*: Map legacy policy mechanisms to the framework's policy model.
* *Gradual Migration*: Support gradual migration from legacy systems to native implementations.

11.2.2. Cloud and Edge Deployments

* *Cloud-Native Design*: Design components as cloud-native microservices.
* *Edge Optimization*: Optimize for edge deployments with limited resources.
* *Hybrid Cloud Support*: Support deployments spanning multiple cloud providers and on-premises infrastructure.
* *Containerization*: Package components as containers for easy deployment.

11.3. Operational Considerations

11.3.1. Monitoring and Management

* *Unified Dashboard*: Provide a unified dashboard for monitoring cross-domain collaborations.
* *Health Checks*: Implement comprehensive health checks for all components.
* *Alerting System*: Configurable alerting for security events and performance issues.
* *Capacity Planning*: Tools for capacity planning and resource allocation.

11.3.2. Update and Maintenance

* *Rolling Updates*: Support rolling updates without disrupting ongoing collaborations.
* *Version Compatibility*: Maintain compatibility between different versions of the framework.
* *Backward Compatibility*: Ensure new versions don’t break existing deployments.
* *Migration Tools*: Provide tools for migrating between framework versions.

12. Use Cases and Examples

12.1. Cross-Enterprise Supply Chain Optimization

Scenario: Multiple companies in a supply chain need to collaborate using AI agents to optimize inventory, logistics, and production scheduling.

Challenges:

* Each company has its own systems, policies, and data sovereignty requirements.
* Sensitive business information must not be exposed to competitors.
* Real-time coordination is needed across organizational boundaries.

Solution using CDI Framework:

1. Each company deploys an interoperability gateway.
2. Supply chain partners establish limited trust relationships for specific collaboration areas.
3. Agents are delegated with carefully constrained capabilities for supply chain optimization.
4. Privacy-preserving techniques (like federated learning) are used to train optimization models without sharing raw data.
5. Audit trails are maintained for compliance and dispute resolution.

12.2. Smart City Federation

Scenario: Multiple municipalities want to collaborate on smart city initiatives while maintaining control over their own infrastructure and data.

Challenges:

* Different municipalities have different regulations and policies.

* Citizens' privacy must be protected across jurisdictional boundaries.

* Emergency situations require rapid coordination.

Solution using CDI Framework:

1. Each municipality operates as an independent domain.

2. A federation agreement establishes baseline trust and policies.

3. Specialized agents for traffic management, emergency response, and utility optimization can operate across municipalities.

4. Data sharing follows strict privacy-preserving protocols.

5. During emergencies, trust levels can be temporarily elevated to enable faster coordination.

12.3. Cross-Border Financial Services Collaboration

Scenario: Financial institutions across different regulatory jurisdictions need AI agents to collaborate on fraud detection, risk assessment, and compliance monitoring while adhering to strict regulatory requirements.

Challenges:

* Different countries have different financial regulations and compliance requirements.

* Sensitive customer financial data cannot be freely shared across borders.

* Real-time fraud detection requires rapid information exchange while maintaining data privacy.

* Audit trails must be comprehensive and legally admissible in multiple jurisdictions.

Solution using CDI Framework:

1. Each financial institution operates within its regulatory domain with institution-specific policies.

2. Cross-border trust agreements are established with legal and regulatory compliance baked into the trust parameters.

3. Specialized agents for fraud pattern recognition can operate across institutions using privacy-preserving techniques like federated learning.

4. Regulatory compliance agents ensure all cross-border operations adhere to applicable laws (e.g., GDPR, CCPA, Basel III).

5. All transactions are logged with cryptographic proofs that can be verified by regulators in any jurisdiction.

Key Technical Features:

* *Regulatory Policy Engines*: Each domain includes policy engines that encode jurisdictional regulations.

* *Privacy-Preserving Analytics*: Use of homomorphic encryption for cross-institution fraud pattern analysis without exposing raw transaction data.

* *Multi-Jurisdictional Audit*: Audit logs structured to meet requirements of multiple regulatory bodies simultaneously.

* *Emergency Override Protocols*: Controlled mechanisms for temporary policy relaxation during financial crises or systemic risk events.

12.4. Cross-Domain Emergency Response System

Scenario: During natural disasters, public health emergencies, or cybersecurity incidents, AI agents from different organizations need to coordinate rapid response while maintaining security boundaries and respecting organizational autonomy.

Challenges:

* Time-critical operations requiring collaboration establishment within minutes.

* Different organizations have varying command hierarchies and decision-making processes.

* Sensitive information must be shared only within necessary scope.

* Resource constraints during emergencies may affect communication reliability.

* Post-emergency accountability and after-action analysis requirements.

Solution using CDI Framework:

1. *Pre-Established Trust Frameworks*: Organizations establish baseline trust relationships and policies during peacetime.

2. *Emergency Trust Elevation*: During declared emergencies, trust levels can be temporarily elevated with appropriate safeguards and post-event review requirements.

3. *Dynamic Policy Adaptation*: Emergency-specific policies override normal operating policies temporarily.

4. *Resource Pooling Coordination*: AI agents coordinate resource allocation across organizational boundaries while maintaining ownership tracking.

5. *Situational Awareness Sharing*: Secure sharing of situational data with granular access controls based on role and need-to-know.

Implementation Components:

* *Emergency Declaration Protocol*: Standardized mechanism for organizations to declare emergencies and activate enhanced collaboration modes.

* *Priority Communication Channels*: Dedicated, high-priority communication paths for emergency coordination.

* *Resource Discovery and Allocation*: Dynamic discovery of available resources (personnel, equipment, facilities) across participating organizations.

* *Common Operational Picture*: Federated data aggregation to create unified situational awareness without centralizing sensitive information.

Example Flow - Natural Disaster Response:

1. Earthquake occurs affecting multiple jurisdictions.

2. Emergency services from affected areas declare emergencies through their interoperability gateways.

3. Trust levels are automatically elevated based on pre-configured emergency protocols.

4. AI agents from different emergency services discover each other's capabilities through federated directories.

5. Resource coordination agents negotiate allocation of search-and-rescue teams, medical supplies, and evacuation routes.

6. Privacy-preserving techniques allow sharing of victim location data while protecting personally identifiable information.

7. Post-event, all actions are audited, trust levels revert to normal, and after-action reports are generated from the comprehensive audit logs.

12.5. Multi-Cloud AI Workload Orchestration

Scenario: Enterprises running AI workloads across multiple cloud providers need agents to manage resources, optimize costs, and ensure compliance while avoiding vendor lock-in.

Challenges:

* Different cloud providers have proprietary APIs and management interfaces.

* Data sovereignty requirements may restrict where workloads can run.

* Cost optimization requires dynamic workload migration across providers.

* Security policies must be consistently enforced across heterogeneous environments.

Solution using CDI Framework:

1. Each cloud environment operates as a separate domain with cloud-specific policies and capabilities.

2. Enterprise governance domain defines overarching policies that apply across all cloud domains.

3. Orchestration agents operate across cloud domains to optimize workload placement based on cost, performance, and compliance requirements.

4. Data movement agents manage secure data transfer between cloud environments with appropriate encryption and access controls.

5. Compliance monitoring agents ensure consistent policy enforcement regardless of where workloads execute.

Technical Implementation:

* *Cloud Abstraction Layer*: Standardized interface that abstracts cloud provider differences.

* *Cost Optimization Engine*: AI agents that analyze pricing models and performance metrics to recommend optimal workload placement.

* *Data Gravity Management*: Intelligent agents that minimize data movement costs by colocating computation with data.

* *Compliance Mapping*: Automated mapping of cloud provider compliance certifications to enterprise regulatory requirements.

12.6. International Research Data Commons

Scenario: Global research collaborations need to share and analyze sensitive research data (e.g., genomic data, climate data, epidemiological data) while protecting participant privacy and respecting national data sovereignty laws.

Challenges:

* Research data often contains sensitive personal information with strict privacy requirements.

* Different countries have different data protection laws and ethical review requirements.

* Researchers need to collaborate on analysis while maintaining control over their contributed data.

* Funding agencies require accountability for data usage and research outputs.

Solution using CDI Framework:

1. Each research institution operates as an independent domain with institutional review board (IRB) approved policies.

2. Research consortia establish collaboration domains with consortium-specific policies that respect all member institutions' requirements.

3. Data analysis agents operate on federated data without needing to centralize sensitive information.

4. Publication agents coordinate multi-institution authorship and ensure proper attribution.

5. Data usage tracking agents monitor compliance with data use agreements across all participating institutions.

Key Features:

* *Ethical Compliance Engine*: Automated checks for compliance with ethical research standards across jurisdictions.

* *Differential Privacy Integration*: Built-in differential privacy mechanisms for aggregate analysis.

* *Data Provenance Tracking*: Complete lineage tracking for research data from collection through analysis to publication.

* *Automated Institutional Review*: Streamlined process for cross-institutional research approval.

12.7. Smart Manufacturing Supply Chain

Scenario: Manufacturers, suppliers, and logistics providers in a smart manufacturing ecosystem need to coordinate production, inventory, and delivery using AI agents while protecting intellectual property and competitive information.

Challenges:

* Competitors may be part of the same supply chain, requiring information boundaries.

* Real-time coordination needed for just-in-time manufacturing and inventory management.

* Quality control data needs to flow upstream to suppliers without revealing proprietary manufacturing processes.

* Cybersecurity threats targeting interconnected industrial control systems.

Solution using CDI Framework:

1. Each company in the supply chain operates as an independent domain with competitive boundary policies.

2. Supply chain collaboration domains establish limited information sharing agreements for specific coordination tasks.

3. Production planning agents coordinate across organizations while preserving competitive intelligence.

4. Quality monitoring agents share defect information with suppliers without revealing manufacturing process details.

5. Cybersecurity monitoring agents collaborate on threat detection while protecting each company’s security posture information.

Implementation Approach:

* *Need-to-Know Information Sharing*: Granular controls on what information is shared with which partners.

* *Competitive Boundary Management*: Clear policies defining what constitutes competitive information that should not be shared.

* *Real-Time Coordination Protocols*: Low-latency communication for time-sensitive manufacturing coordination.

* *Industrial IoT Security*: Specialized security protocols for industrial control system integration.

12.8. Cross-Jurisdictional Law Enforcement Collaboration

Scenario: Law enforcement agencies from different jurisdictions need to collaborate on investigations while respecting legal boundaries, privacy laws, and chain of evidence requirements.

Challenges:

* Different legal systems have different requirements for evidence collection and sharing.

* Privacy laws restrict sharing of personal information across borders.

* Chain of custody must be maintained for digital evidence.

* Need for rapid information sharing in time-sensitive investigations (e.g., kidnapping, terrorism).

Solution using CDI Framework:

1. Each law enforcement agency operates within its jurisdictional domain with legally compliant policies.

2. Mutual legal assistance treaty (MLAT) frameworks are encoded into trust agreements between domains.

3. Investigation coordination agents facilitate information sharing within legal boundaries.

4. Evidence handling agents maintain chain of custody across jurisdictional transfers.

5. Legal compliance agents ensure all cross-jurisdictional operations adhere to applicable laws.

Technical Considerations:

* *Legal Authority Verification*: Automated verification of legal authority for specific investigative actions.

* *Privacy-Preserving Matching*: Techniques for matching persons of interest without sharing full identity information.

* *Digital Evidence Chain of Custody*: Cryptographic proof of evidence integrity and custody chain.

* *Automated Legal Documentation*: Generation of legally required documentation for cross-jurisdictional operations.

12.9. Global Energy Grid Coordination

Scenario: Energy providers, grid operators, and renewable energy producers across regions need to coordinate energy production, distribution, and consumption while maintaining grid stability and respecting commercial interests.

Challenges:

* Different grid operators have proprietary control systems and protocols.

* Renewable energy sources introduce variability requiring real-time coordination.

* Commercial energy trading requires secure transaction mechanisms.

* Grid cybersecurity requires collaborative threat detection without revealing vulnerability information.

Solution using CDI Framework:

1. Each energy provider and grid operator operates as an independent domain.

2. Regional coordination domains establish protocols for grid stability management.

3. Energy trading agents facilitate secure transactions across organizational boundaries.

4. Grid optimization agents coordinate production and consumption across regions.

5. Cybersecurity agents collaborate on threat detection while protecting proprietary system information.

Key Components:

* *Grid Stability Protocols*: Standardized protocols for maintaining grid stability across interconnected systems.

* *Renewable Integration Management*: Coordination of variable renewable energy sources across the grid.

* *Secure Energy Trading*: Privacy-preserving mechanisms for energy market transactions.

* *Grid Resilience Coordination*: Collaborative response to grid disturbances and failures.

13. IANA Considerations

This document has no IANA actions.

14. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

Acknowledgments

Author’s Address

Yong Cui
Tsinghua University
Beijing, 100084
China
Email: cuiyong@tsinghua.edu.cn
URI: http://www.cuiyong.net/