Patent Application: Conversational Governance Intelligence System

Title of Invention

Natural Language Processing System and Method for Domain-Specific Conversational Cloud Governance Management with Context-Aware Multi-Turn Dialogue and Automated Policy Synthesis

Technical Field

This invention relates to natural language processing systems for cloud computing governance, specifically to conversational AI interfaces that enable users to manage, query, and configure cloud governance policies through natural language interactions with context-aware processing and automated workflow generation.

Independent Claims

Claim 1 (System Claim - Broadest)

A computer-implemented conversational governance intelligence system comprising:

a) **a domain-specific natural language understanding (NLU) engine** configured to:

- process natural language queries using transformer-based models fine-tuned on a corpus of at least 1 million cloud governance documents,
- implement named entity recognition (NER) for governance-specific entities including policy names, resource types, compliance frameworks, and cost centers with F1 score exceeding 0.92,
- perform intent classification across at least 50 distinct governance operation types using hierarchical multi-label classification,
- extract slot values for governance parameters with confidence scoring and ambiguity detection;

b) **a multi-turn conversation management system** implementing:

- conversation state tracking using graph-based dialogue state representation with nodes for entities, intents, and context,
- anaphora resolution specifically adapted for technical governance terminology,
- context window management maintaining relevant conversation history for up to 20 turns,
- user profile integration storing role-based permissions, preferences, and interaction history;

c) **a governance query translation engine** configured to:

- convert natural language queries into executable API calls for Azure Policy, RBAC, Network, and Cost Management services,
- implement semantic parsing using abstract meaning representation (AMR) graphs,
- perform query optimization to minimize API calls while maximizing result completeness,
- generate fallback queries when primary translations fail;

d) **an intelligent response generation system** implementing:

- template-based generation for structured governance data with dynamic slot filling,
- neural language generation using GPT-based models for explanatory content,
- multi-modal response composition combining text, tables, charts, and interactive elements,
- response personalization based on user expertise level and role;

e) **a policy synthesis engine** configured to:

- automatically generate governance policies from natural language descriptions,
- implement policy validation against Azure Resource Manager schemas,
- provide policy conflict detection and resolution suggestions,
- generate policy documentation and compliance mappings;

f) **a conversational workflow automation system** configured to:

- decompose complex governance requests into executable workflow steps,
- orchestrate multi-service operations across governance domains,
- implement rollback mechanisms for failed operations,
- provide progress tracking and status updates during execution;

wherein the system achieves at least 95% intent recognition accuracy and sub-2-second response latency for 90% of queries.

Claim 2 (Method Claim - Broadest)

A computer-implemented method for conversational cloud governance management comprising:

a) **processing natural language governance queries** by:

- tokenizing input text using byte-pair encoding optimized for technical terminology,
- applying domain-adapted BERT embeddings with governance-specific vocabulary,

- performing multi-task learning for simultaneous intent detection and entity extraction,
- calculating query complexity scores to route to appropriate processing pipelines;

b) **maintaining conversation context** by:

- constructing dynamic knowledge graphs representing conversation state,
- implementing attention mechanisms over conversation history,
- performing entity coreference resolution using governance-aware rules,
- updating user context models with each interaction;

c) **translating queries to governance operations** by:

- mapping natural language to formal query languages (KQL, ARM templates),
- generating parameterized API calls with proper authentication and scoping,
- implementing query plan optimization using cost-based analysis,
- providing query explanation in natural language;

d) **generating contextually appropriate responses** by:

- selecting response modality based on query type and data characteristics,
- applying governance-specific language models for text generation,
- formatting complex data using intelligent summarization and visualization,
- incorporating relevant compliance and best practice guidance;

e) **synthesizing governance policies from descriptions** by:

- parsing policy requirements using dependency parsing and semantic role labeling,
- mapping requirements to policy rule templates and conditions,
- generating JSON policy definitions conforming to cloud provider schemas,
- validating policies through simulation and impact analysis;

f) **executing conversational workflows** by:

- creating directed acyclic graphs (DAGs) of governance operations,
- implementing saga patterns for distributed transaction management,
- providing real-time status updates through conversational interface,
- handling errors with natural language explanations and remediation options.

Dependent Claims

Claim 3 (Dependent on Claim 1)

The system of claim 1, wherein the domain-specific NLU engine further comprises:

- a continual learning module that updates models based on user feedback without catastrophic forgetting,
- a multi-lingual support system handling at least 10 languages with cross-lingual transfer learning,
- an acronym and abbreviation expansion system specific to cloud governance terminology,
- a query disambiguation module that generates clarifying questions when intent confidence is below threshold.

Claim 4 (Dependent on Claim 1)

The system of claim 1, wherein the conversation management system implements:

- episodic memory networks for long-term conversation recall across sessions,
- personality-consistent response generation maintaining professional tone,
- proactive assistance suggestions based on detected user goals,
- conversation summarization for handoff between support agents.

Claim 5 (Dependent on Claim 1)

The system of claim 1, wherein the policy synthesis engine further comprises:

- a policy template library with over 500 pre-validated governance patterns,
- automated compliance mapping to frameworks including SOC2, ISO 27001, and HIPAA,
- policy testing sandbox for safe evaluation before deployment,
- natural language policy explanation generator for non-technical stakeholders.

Claim 6 (Dependent on Claim 2)

The method of claim 2, wherein processing natural language queries includes:

- implementing few-shot learning for handling novel governance concepts,
- applying reinforcement learning from human feedback (RLHF) for response improvement,
- detecting and flagging potentially dangerous governance operations,
- providing confidence calibration for model predictions.

Claim 7 (Dependent on Claim 2)

The method of claim 2, wherein generating responses further comprises:

- implementing progressive disclosure for complex information,

- generating interactive tutorials for governance tasks,

- providing code snippets and automation scripts when relevant,

- adapting explanation detail based on user's demonstrated knowledge level.

Claim 8 (Architecture Claim)

The system of claim 1, further comprising:

- a distributed inference infrastructure using model parallelism across GPUs,

- an edge deployment capability for offline governance assistance,

- a federated learning system for privacy-preserving model updates,

- real-time model serving with <100ms inference latency.

Claim 9 (Security Claim)

The system of claim 1, implementing:

- conversation encryption with end-to-end security,

- role-based access control for conversation capabilities,

- audit logging of all governance operations initiated through conversation,

- PII detection and redaction in conversation logs.

Claim 10 (Integration Claim)

The system of claim 1, providing:

- REST and GraphQL APIs for third-party integration,

- webhook support for external event notifications,

- SSO integration with enterprise identity providers,

- native plugins for Slack, Teams, and enterprise chat platforms.

Technical Diagrams

Figure 1: Conversational Governance System Architecture

Conversational Governance Intelligence System

User Input Interface

| Text Input | Voice Input | Code Snippets | Documents Upload |

Domain-Specific NLU Engine

Governance-Tuned Transformer
- 12B parameters • 100K governance vocab
- Multi-task heads • Confidence scoring

| Intent Classification | Entity Recognition | Slot Extraction |

Multi-Turn Conversation Manager

Conversation State Graph

| Entity |——| Intent |——| Context |

↓   ↓   ↓

[User Profile] [History] [Permissions]

Query Translation & Execution Engine

| Semantic | API | Workflow |

```
|  |  |  Parser   | → |  Generator  | → | Orchestrator |        |        |
|  |     |_____|   |_____|   |_____|        |        |
|  |                                        |           |
|  |  Target APIs: [Azure Policy] [RBAC] [Network] [Cost]    |        |
|  |_____|        |
|  |                   |                       |
|  |_____▼_____|        |
|  |     Intelligent Response Generation              |        |
|  |   _____   _____   _____        |        |
|  |  |  Template  |   |   Neural   |   | Multi-Modal |   |        |
|  |  |  Generator  |   |  Generator  |   |  Composer   |   |        |
|  |  |_____|   |_____|   |_____|        |        |
|  |                   |           |
|  |  Output: [Text] [Tables] [Charts] [Actions] [Code]    |        |
|  |_____|        |
|  |                   |
|__|_____|
```
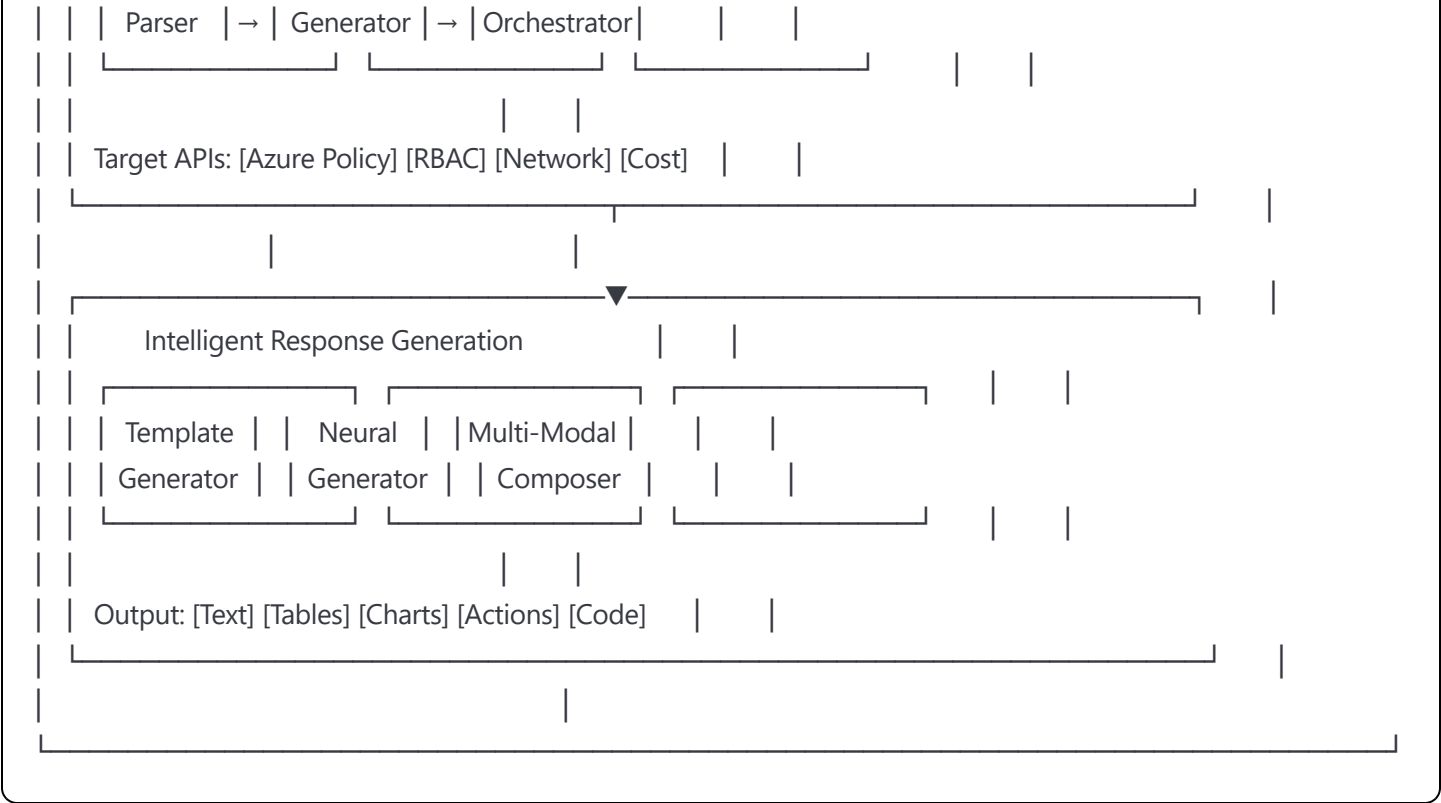
Figure 2: Natural Language to Policy Synthesis Pipeline

Policy Synthesis Engine

Natural Language Input: "Prevent VMs in production from having public IP addresses unless they're tagged as 'public-facing'"

Linguistic Analysis

| Dependency Parsing | Semantic Role Labeling | Condition Extraction |

▼

Requirement Decomposition

Requirements Tree:
• Target: VMs in production
• Condition: Has public IP
• Exception: Tag 'public-facing' exists
• Action: Deny

▼

Policy Template Matching

| Template Library (500+ templates) | → | Similarity Scoring (Cosine/BERT) | → | Best Match Selection |

▼

Policy JSON Generation

```json
{
  "mode": "Indexed",
  "policyRule": {
    "if": {
```

```
"allOf": [
    {
        "field": "type",
        "equals": "Microsoft.Compute/virtualMachines"
    },
    {
        "field": "location",
        "in": ["eastus", "westus2"] // production
    },
    {
        "field": "Microsoft.Network/publicIPAddresses",
        "exists": "true"
    },
    {
        "field": "tags['public-facing']",
        "exists": "false"
    }
]
},
"then": {
    "effect": "deny"
}
}
}
```

Figure 3: Multi-Turn Conversation State Management

Conversation State Graph Evolution

Turn 1: "Show me all VMs that are non-compliant"

State Graph T1:

```
        ┌──────────┐  Intent: query_compliance
        │   VMs    ├────── Entity: resource_type
        └──────────┘  Status: non_compliant
```

↓
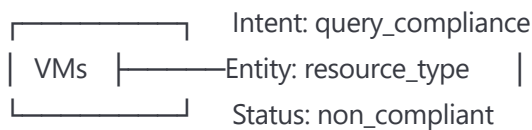
Turn 2: "Which ones are in production?"

State Graph T2:

```
   ┌──────────┐    ┌──────────────┐
   │   VMs    ├────┤ Non-Compliant│
   └──────────┘    └──────────────┘
        │   Environment: production
        └──────────Anaphora: "ones" → VMs
```

↓
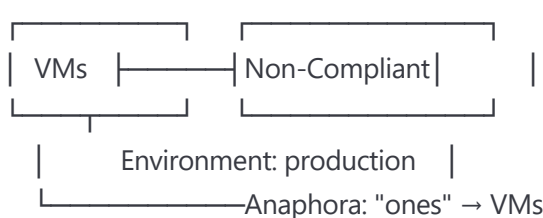
Turn 3: "Create a policy to fix this"

State Graph T3:

```
   ┌──────────┐    ┌──────────────┐
   │   VMs    ├────┤ Non-Compliant│
   └──────────┘    └──────────────┘
        │                │
        │      ┌──────▼──────┐
        └──────┤ Create Policy│
               └──────────────┘
```
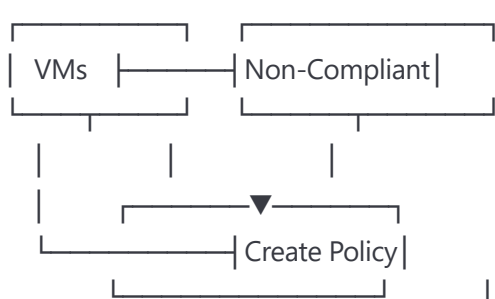
Context: "this" → non-compliance issue
Action: synthesize_policy

Context Window Management:

Active Context (Last 5 turns)
• Entities: [VMs, production, non-compliant]
• Intent History: [query → filter → create]
• User Goal: Remediate compliance issues
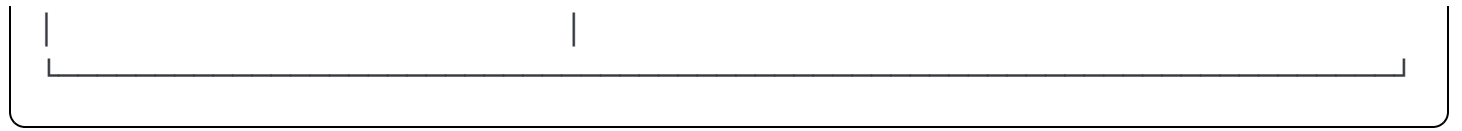
Figure 4: Query Translation and API Orchestration

```
┌─────────────────────────────────────────────────────────────────┐
│  ┌──────────────────────────────────────────────────────────┐   │
│  │         Query Translation Pipeline          │            │   │
│  ├──────────────────────────────────────────────────────────┤   │
│  │                         │                                 │   │
│  │ Natural Language: "Find all storage accounts with public access │
│  │           that haven't been accessed in 30 days"         │   │
│  │                         │                                 │   │
│  │  ┌──────────────────────────────────────────────┐        │   │
│  │  │        Semantic Parsing            │      │   │        │   │
│  │  │  ┌──────────────────────────────────┐   │   │  │      │   │
│  │  │  │   Abstract Meaning Graph      │     │   │   │      │   │
│  │  │  │   ┌──────────┐                │     │   │   │      │   │
│  │  │  │   │  FIND   │                 │     │   │   │      │   │
│  │  │  │   └──────────┘                │     │   │   │      │   │
│  │  │  │        │              │       │     │   │   │      │   │
│  │  │  │   ┌────────▼─────────┐        │     │   │   │      │   │
│  │  │  │   │ Storage Accts ├──[has]──→Public  │   │   │     │   │
│  │  │  │   └──────────────────┘  Access │    │   │   │     │   │
│  │  │  │        │              │       │     │   │   │     │   │
│  │  │  │   └──[not_accessed]──→30 days  │    │   │   │     │   │
│  │  │  └──────────────────────────────────┘   │   │  │     │   │
│  │  └───────────────────────────────────────────────┘      │   │
│  │                │                │                        │   │
│  │  ┌───────────────────────────────▼──────────────┐       │   │
│  │  │     Multi-Service Query Planning    │     │   │       │   │
│  │  │  ┌────────────────────────────────────┐   │  │  │     │   │
│  │  │  │ Service 1: Azure Resource Graph    │   │   │  │     │   │
│  │  │  │ Query: resources              │    │    │   │  │     │   │
│  │  │  │ | where type == "microsoft.storage/  │   │    │   │     │   │
│  │  │  │    storageaccounts"           │    │    │   │     │   │
│  │  │  │ | where properties.allowBlobPublicAccess │    │   │     │   │
│  │  │  └────────────────────────────────────┘   │  │  │     │   │
│  │  │           │                │    │                      │   │
│  │  │  ┌────────────────────────────▼──────────┐  │  │       │   │
│  │  │  │ Service 2: Azure Monitor      │    │    │  │  │      │   │
│  │  │  │ Query: StorageAccount         │    │    │  │  │      │   │
│  │  │  │ | where TimeGenerated > ago(30d)  │   │    │  │      │   │
│  │  │  │ | summarize LastAccess = max(TimeGen) │   │    │  │      │   │
│  │  │  │    by AccountName             │    │    │  │       │   │
│  │  │  └───────────────────────────────────────┘  │  │       │   │
│  │  │           │                │    │                      │   │
│  │  │  ┌────────────────────────────▼──────────┐  │  │       │   │
│  │  │  │ Query Optimization & Join Strategy │    │    │  │     │   │
```

```
| |  |  • Parallel execution of sub-queries    |        |      |
| |  |  • Client-side join on AccountName       |        |     |
| |  |  • Result caching for 5 minutes          |        |     |
| |  |_____|        |     |
| |_____|        |
|                 |                      |
|  ┌──────────────────────────────▼──────────────────────┐  |
| |          API Execution & Error Handling       |      |
| |  ┌──────────┐ ┌─────────┐ ┌─────────┐ ┌──────────┐ |      |
| |  │Authenticate│ │Execute  │ │Retry on │ │Fallback  │ |      |
| |  │with Token│→│Queries  │→│Failure  │→│Strategy  │ |      |
| |  └──────────┘ └─────────┘ └─────────┘ └──────────┘ |      |
| |                        |        |
| |  Error Response: "I found 12 storage accounts with    |      |
| |  public access. However, I couldn't retrieve access   |      |
| |  logs for 3 accounts due to permissions. Would you    |      |
| |  like me to list the 9 I could fully analyze?"        |      |
| |_____|      |
|                  |
|_____|
```
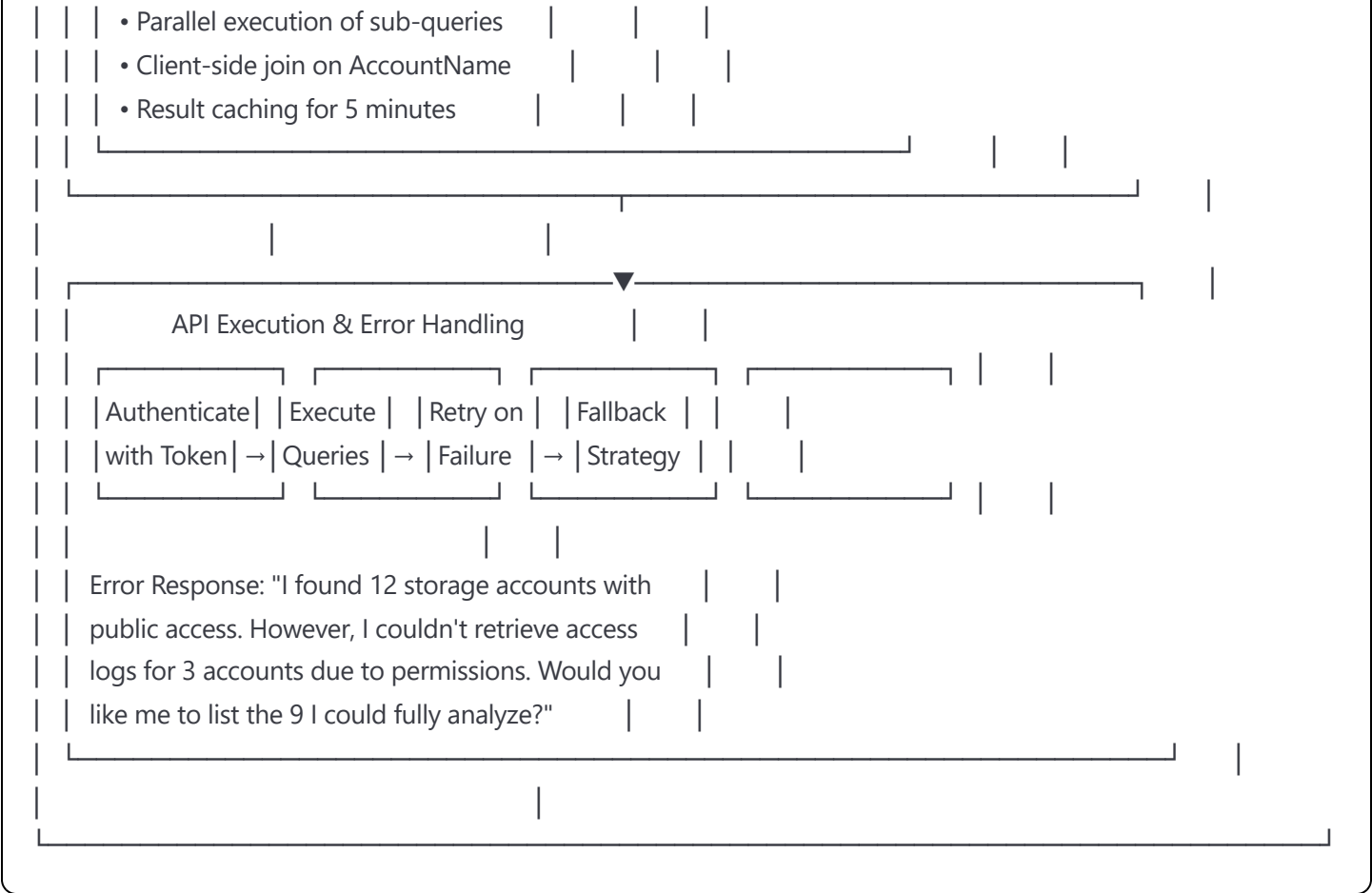
Figure 5: Response Generation Architecture

Intelligent Response Generation System

Query Result: 47 non-compliant resources across 3 subscriptions

Response Strategy Selection

| Data Volume: | User Role: | Query Type: |
| 47 items | Security Admin | Compliance |

↓

Strategy: Summary + Grouped Table + Actions

▼

Multi-Modal Response Composition

1. Natural Language Summary (Neural Generation)

"I found 47 resources that don't comply with
your security policies. The majority (31) are
storage accounts with public access enabled.
This poses a significant data exposure risk."

2. Interactive Table (Template Generation)

Resource Type	Count	Risk	Quick Action
Storage Acct	31	High	[Remediate]
SQL Database	12	Med	[Review]
Key Vault	4	High	[Remediate]

3. Visualization (Chart Generation)

Compliance Trend (Last 30 Days)
📊 [Line chart showing trend]

4. Actionable Recommendations (Workflow Links)

```
| |   ┌──────────────────────────────────────────────────┐   |   |        |
| |   |  • Apply bulk remediation policy          |      |      |
| |   |  • Schedule compliance review meeting     |      |      |
| |   |  • Generate detailed report for audit     |      |      |
| |   └──────────────────────────────────────────────────┘   |   |
|   └──────────────────────────────────────────────────────┐   |
|                              |
|   ┌──────────────────────────────────────────────────┐   |
| |          Response Personalization Engine       |      |
| |  • Expertise Level: Advanced → Include technical details |      |
| |  • Previous Interactions: Focus on storage accounts   |      |
| |  • Time Constraints: Busy → Prioritize critical items |      |
| |  • Preferred Format: Tables → Minimize prose      |      |
| └──────────────────────────────────────────────────┘   |
|                              |
└──────────────────────────────────────────────────────────┘
```

Abstract

A natural language processing system enabling conversational management of cloud governance through domain-specific language understanding and automated policy synthesis. The system employs transformer-based models fine-tuned on governance corpora to achieve high-accuracy intent recognition and entity extraction for technical terminology. Multi-turn conversation management utilizes graph-based state tracking with sophisticated anaphora resolution and context preservation across extended dialogues. The invention translates natural language queries into optimized multi-service API orchestrations, handling complex governance operations across policy, access control, network, and cost domains. An intelligent response generation system provides multi-modal outputs combining natural language explanations, structured data visualizations, and actionable recommendations personalized to user roles and expertise levels. The policy synthesis engine automatically generates validated governance policies from natural language descriptions, reducing policy creation time by 90% while ensuring compliance with cloud provider schemas and organizational standards.