```yaml
openapi: 3.0.3
info:
  title: NaMo Memory API
  version: 2.1.0
  description: |
    Memory management system for NaMo AI with Dharma principles integration.
    Integrated with Google Cloud infrastructure for practical deployment.

    Infrastructure Stack:
    - Frontend: Apigee API Hub
    - Processing: Vertex AI Agent Engine
    - Storage: BigQuery (long-term), Firestore (short-term)
    - Deployment: Cloud Run

servers:
  - url: https://{apigee-domain}/namo/memory
    description: Apigee API Hub endpoint
    variables:
      apigee-domain:
        default: api.your-company.com
        description: Your Apigee domain name

# Security schemes for Google Cloud integration
components:
  securitySchemes:
    apiKeyAuth:
      type: apiKey
      in: header
      name: X-API-Key
    googleOAuth:
      type: oauth2
      flows:
        clientCredentials:
          tokenUrl: https://oauth2.googleapis.com/token
          scopes:
            - https://www.googleapis.com/auth/cloud-platform

  schemas:
    # Practical memory schemas for Google Cloud integration
    MemoryRecord:
      type: object
      properties:
        id:
          type: string
          format: uuid
          description: Unique memory ID
        content:
          type: string
```

```
      description: Memory content
    type:
      type: string
      enum: [short-term, long-term, contextual]
      description: Memory storage type
    emotion_context:
      type: object
      properties:
        sentiment_score:
          type: number
          minimum: -1
          maximum: 1
        emotion_type:
          type: string
          enum: [metta, karuna, mudita, upekkha, neutral]
        intensity:
          type: integer
          minimum: 1
          maximum: 10
    dharma_tags:
      type: array
      items:
        type: string
        enum: [anicca, dukkha, anatta, metta, karuna, mudita, upekkha]
    timestamp:
      type: string
      format: date-time
    expiration_time:
      type: string
      format: date-time
      description: For short-term memories

MemoryStorageRequest:
  type: object
  required: [content, type]
  properties:
    content:
      type: string
      example: "User expressed interest in Buddhist meditation techniques"
    type:
      type: string
      enum: [short-term, long-term, contextual]
      example: "contextual"
    session_id:
      type: string
      description: Current session ID for contextual memories
    emotion_context:
      $ref: '#/components/schemas/MemoryRecord/properties/emotion_context'
```

```
    dharma_tags:
      $ref: '#/components/schemas/MemoryRecord/properties/dharma_tags'

  MemoryQuery:
    type: object
    properties:
      query:
        type: string
        description: Search query for memories
      memory_types:
        type: array
        items:
          type: string
          enum: [short-term, long-term, contextual]
      emotion_filter:
        $ref: '#/components/schemas/MemoryRecord/properties/emotion_context'
      dharma_tags:
        $ref: '#/components/schemas/MemoryRecord/properties/dharma_tags'
      time_range:
        type: object
        properties:
          start_time:
            type: string
            format: date-time
          end_time:
            type: string
            format: date-time
      limit:
        type: integer
        maximum: 100
        default: 10

  BatchMemoryOperation:
    type: object
    properties:
      operations:
        type: array
        items:
          type: object
          properties:
            operation_type:
              type: string
              enum: [store, update, delete]
            memory_data:
              $ref: '#/components/schemas/MemoryStorageRequest'
            memory_id:
              type: string
```

```yaml
paths:
  # Core memory endpoints
  /store:
    post:
      summary: Store memory with emotional and dharma context
      description: |
        Store memory in appropriate storage (Firestore for short-term, BigQuery for long-term)
        with emotional context and dharma tagging for better retrieval.
      security:
        - apiKeyAuth: []
        - googleOAuth: [cloud-platform]
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/MemoryStorageRequest'
      responses:
        "200":
          description: Memory stored successfully
          content:
            application/json:
              schema:
                type: object
                properties:
                  success:
                    type: boolean
                  memory_id:
                    type: string
                  storage_type:
                    type: string
                  expiration_time:
                    type: string
                    format: date-time
        "400":
          description: Invalid request
        "401":
          description: Unauthorized

  /recall:
    post:
      summary: Recall memories with advanced filtering
      description: |
        Retrieve memories based on multiple criteria including emotional context,
        dharma tags, and temporal filters. Integrated with Vertex AI for semantic search.
      security:
        - apiKeyAuth: []
      requestBody:
```

```
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/MemoryQuery'
    responses:
      "200":
        description: Memories retrieved successfully
        content:
          application/json:
            schema:
              type: object
              properties:
                memories:
                  type: array
                  items:
                    $ref: '#/components/schemas/MemoryRecord'
                total_count:
                  type: integer
                query_time:
                  type: number
                  description: Query execution time in milliseconds

  /batch:
    post:
      summary: Batch memory operations
      description: |
        Perform multiple memory operations in a single request for efficiency.
        Useful for session initialization and cleanup.
      security:
        - apiKeyAuth: []
        - googleOAuth: [cloud-platform]
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/BatchMemoryOperation'
      responses:
        "200":
          description: Batch operations completed
          content:
            application/json:
              schema:
                type: object
                properties:
                  results:
                    type: array
```

```
                items:
                  type: object
                  properties:
                    operation_type:
                      type: string
                    success:
                      type: boolean
                    memory_id:
                      type: string
                    error:
                      type: string

# Management endpoints
/sessions/{sessionId}:
  delete:
    summary: Clear session memories
    description: |
      Remove all short-term memories associated with a specific session.
      Automated cleanup based on session expiration.
    parameters:
      - name: sessionId
        in: path
        required: true
        schema:
          type: string
    responses:
      "200":
        description: Session memories cleared
        content:
          application/json:
            schema:
              type: object
              properties:
                deleted_count:
                  type: integer
                session_id:
                  type: string

/stats:
  get:
    summary: Get memory system statistics
    description: |
      Retrieve statistics about memory usage, distribution, and performance.
    responses:
      "200":
        description: Statistics retrieved
        content:
          application/json:
```

```
            schema:
              type: object
              properties:
                total_memories:
                  type: integer
                memory_by_type:
                  type: object
                  additionalProperties:
                    type: integer
                storage_usage:
                  type: object
                  properties:
                    firestore:
                      type: number
                    bigquery:
                      type: number
                average_response_time:
                  type: number

# Dharma-enhanced endpoints
/dharma/insights:
  get:
    summary: Get dharma insights from memories
    description: |
      Analyze memories to extract dharma insights and patterns using
      Vertex AI natural language processing.
    parameters:
      - name: pattern_type
        in: query
        schema:
          type: string
          enum: [karmic, emotional, behavioral]
      - name: time_window
        in: query
        schema:
          type: string
          enum: [day, week, month, all]
    responses:
      "200":
        description: Dharma insights generated
        content:
          application/json:
            schema:
              type: object
              properties:
                insights:
                  type: array
                  items:
```

```
                type: object
                properties:
                  pattern_type:
                    type: string
                  insight:
                    type: string
                  confidence:
                    type: number
                  supporting_memories:
                    type: array
                    items:
                      type: string

  /emotion/profile:
    get:
      summary: Generate emotional profile from memories
      description: |
        Create emotional profile based on accumulated memories and interactions.
        Uses Vertex AI sentiment analysis and emotional pattern recognition.
      parameters:
        - name: timeframe
          in: query
          schema:
            type: string
            enum: [session, day, week, custom]
        - name: user_id
          in: query
          schema:
            type: string
      responses:
        "200":
          description: Emotional profile generated
          content:
            application/json:
              schema:
                type: object
                properties:
                  emotional_patterns:
                    type: object
                    properties:
                      dominant_emotion:
                        type: string
                      emotion_distribution:
                        type: object
                        additionalProperties:
                          type: number
                      stability_score:
                        type: number
```

```
dharma_alignment:
  type: object
  properties:
    brahmavihara_balance:
      type: object
      properties:
        metta: { type: number }
        karuna: { type: number }
        mudita: { type: number }
        upekkha: { type: number }
    trilakshana_scores:
      type: object
      properties:
        anicca: { type: number }
        dukkha: { type: number }
        anatta: { type: number }
```