```yaml
openapi: 3.0.3
info:
  title: NaMo Emotion Engine API
  version: 2.1.0
  description: |
    Emotion processing 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 Natural Language API + Custom Models
    - Storage: Firestore (emotional states), BigQuery (emotional patterns)
    - Deployment: Cloud Run

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

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 emotion schemas for Google Cloud integration
    EmotionalState:
      type: object
      properties:
        session_id:
          type: string
          description: Current session identifier
        emotional_vector:
          type: object
          properties:
            metta:
```

```
        type: number
        minimum: 0
        maximum: 1
        description: Loving-kindness score
      karuna:
        type: number
        minimum: 0
        maximum: 1
        description: Compassion score
      mudita:
        type: number
        minimum: 0
        maximum: 1
        description: Sympathetic joy score
      upekkha:
        type: number
        minimum: 0
        maximum: 1
        description: Equanimity score
      stability:
        type: number
        minimum: 0
        maximum: 1
        description: Emotional stability index
  intensity:
    type: integer
    minimum: 1
    maximum: 10
    description: Emotional intensity level
  valence:
    type: number
    minimum: -1
    maximum: 1
    description: Positive/negative emotional valence
  detected_triggers:
    type: array
    items:
      type: string
    description: Identified emotional triggers
  dharma_context:
    type: object
    properties:
      anicca_awareness:
        type: number
        minimum: 0
        maximum: 1
      dukkha_understanding:
        type: number
```

```
          minimum: 0
          maximum: 1
        anatta_realization:
          type: number
          minimum: 0
          maximum: 1
    timestamp:
      type: string
      format: date-time

EmotionAnalysisRequest:
  type: object
  required: [text_content, session_id]
  properties:
    text_content:
      type: string
      description: Text to analyze for emotional content
    session_id:
      type: string
      description: Session identifier for context
    previous_state:
      $ref: '#/components/schemas/EmotionalState'
    analysis_depth:
      type: string
      enum: [basic, standard, deep, dharma]
      default: standard

EmotionAdjustment:
  type: object
  properties:
    target_state:
      $ref: '#/components/schemas/EmotionalState/properties/emotional_vector'
    adjustment_strategy:
      type: string
      enum: [gradual, immediate, transformative]
      default: gradual
    dharma_principles:
      type: array
      items:
        type: string
        enum: [metta, karuna, mudita, upekkha, anicca, dukkha, anatta]
    max_transition_time:
      type: integer
      description: Maximum transition time in milliseconds
      default: 5000

EmotionalPattern:
  type: object
```

```
      properties:
        pattern_type:
          type: string
          enum: [cyclic, reactive, progressive, transformative]
        frequency:
          type: number
          description: Pattern occurrence frequency
        intensity_range:
          type: object
          properties:
            min: { type: number }
            max: { type: number }
            average: { type: number }
        triggers:
          type: array
          items: { type: string }
        dharma_insights:
          type: array
          items: { type: string }

paths:
  # Core emotion endpoints
  /analyze:
    post:
      summary: Analyze emotional content from text
      description: |
        Analyze emotional content using Vertex AI Natural Language API
        enhanced with Dharma principles and emotional intelligence.
      security:
        - apiKeyAuth: []
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/EmotionAnalysisRequest'
      responses:
        "200":
          description: Emotional analysis complete
          content:
            application/json:
              schema:
                type: object
                properties:
                  emotional_state:
                    $ref: '#/components/schemas/EmotionalState'
                  confidence_scores:
                    type: object
```

```
            properties:
              sentiment: { type: number }
              emotion: { type: number }
              dharma: { type: number }
            detected_triggers:
              type: array
              items: { type: string }
            recommended_response:
              type: object
              properties:
                emotional_tone: { type: string }
                dharma_approach: { type: string }
                intensity_level: { type: number }

/state:
  get:
    summary: Get current emotional state
    description: |
      Retrieve the current emotional state from Firestore storage.
      Includes real-time emotional vector and Dharma context.
    parameters:
      - name: session_id
        in: query
        required: true
        schema:
          type: string
    responses:
      "200":
        description: Current emotional state
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/EmotionalState'
      "404":
        description: Emotional state not found for session

/adjust:
  post:
    summary: Adjust emotional state
    description: |
      Modify emotional state based on desired target state and strategy.
      Uses Vertex AI for smooth emotional transitions.
    security:
      - apiKeyAuth: []
    requestBody:
      required: true
      content:
        application/json:
```

```
        schema:
          $ref: '#/components/schemas/EmotionAdjustment'
    responses:
      "200":
        description: Emotional adjustment successful
        content:
          application/json:
            schema:
              type: object
              properties:
                success: { type: boolean }
                transition_time: { type: number }
                new_state: { type: object }
                wisdom_insights: { type: array, items: { type: string } }

# Advanced emotion management
/patterns:
  get:
    summary: Analyze emotional patterns
    description: |
      Identify emotional patterns from historical data using BigQuery analytics
      and Vertex AI pattern recognition.
    parameters:
      - name: time_window
        in: query
        schema:
          type: string
          enum: [hour, day, week, month, custom]
        description: Time window for pattern analysis
      - name: user_id
        in: query
        schema:
          type: string
        description: Specific user for pattern analysis
    responses:
      "200":
        description: Emotional patterns identified
        content:
          application/json:
            schema:
              type: object
              properties:
                patterns:
                  type: array
                  items:
                    $ref: '#/components/schemas/EmotionalPattern'
                dominant_pattern:
                  type: string
```

```
            pattern_stability:
              type: number
            improvement_suggestions:
              type: array
              items: { type: string }

/brahmavihara/balance:
  get:
    summary: Get Brahmavihāra balance analysis
    description: |
      Analyze the balance of the Four Divine States using statistical analysis
      and Dharma principles integration.
    parameters:
      - name: session_id
        in: query
        schema:
          type: string
      - name: timeframe
        in: query
        schema:
          type: string
          enum: [current, recent, historical]
    responses:
      "200":
        description: Brahmavihāra balance analysis
        content:
          application/json:
            schema:
              type: object
              properties:
                balance_scores:
                  type: object
                  properties:
                    metta: { type: number }
                    karuna: { type: number }
                    mudita: { type: number }
                    upekkha: { type: number }
                imbalance_detected:
                  type: boolean
                recommended_actions:
                  type: array
                  items: { type: string }
                dharma_alignment:
                  type: number

/triggers/analyze:
  post:
    summary: Analyze emotional triggers
```

```
    description: |
      Identify and analyze emotional triggers using machine learning
      and Dharma context awareness.
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              conversation_history:
                type: array
                items: { type: string }
              emotional_responses:
                type: array
                items: { type: number }
              context_data:
                type: object
    responses:
      "200":
        description: Trigger analysis complete
        content:
          application/json:
            schema:
              type: object
              properties:
                identified_triggers:
                  type: array
                  items:
                    type: object
                    properties:
                      trigger: { type: string }
                      strength: { type: number }
                      context: { type: string }
                sensitivity_analysis:
                  type: object
                  properties:
                    overall_sensitivity: { type: number }
                    trigger_specificity: { type: object }
                coping_strategies:
                  type: array
                  items: { type: string }

  # Session management
  /sessions/{sessionId}:
    delete:
      summary: Clear emotional session data
      description: |
```

```
    Remove emotional data for a specific session from Firestore.
    Maintains aggregated patterns in BigQuery for analysis.
  parameters:
    - name: sessionId
      in: path
      required: true
      schema:
        type: string
  responses:
    "200":
      description: Session data cleared successfully
      content:
        application/json:
          schema:
            type: object
            properties:
              deleted_count: { type: integer }
              preserved_patterns: { type: boolean }

/stats:
  get:
    summary: Get emotion engine statistics
    description: |
      Retrieve statistics about emotional processing and system performance.
    responses:
      "200":
        description: Statistics retrieved
        content:
          application/json:
            schema:
              type: object
              properties:
                total_analyses:
                  type: integer
                average_processing_time:
                  type: number
                emotion_distribution:
                  type: object
                  additionalProperties: { type: number }
                accuracy_metrics:
                  type: object
                  properties:
                    sentiment: { type: number }
                    emotion: { type: number }
                    dharma: { type: number }

# Dharma-enhanced endpoints
/dharma/balance:
```

```yaml
post:
  summary: Balance emotions using Dharma principles
  description: |
    Apply Buddhist psychology principles to balance emotional states
    and achieve optimal mental equilibrium.
  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          properties:
            current_state:
              $ref: '#/components/schemas/EmotionalState'
            desired_balance:
              type: object
              properties:
                metta: { type: number }
                karuna: { type: number }
                mudita: { type: number }
                upekkha: { type: number }
            dharma_focus:
              type: array
              items: { type: string }
  responses:
    "200":
      description: Dharma balancing complete
      content:
        application/json:
          schema:
            type: object
            properties:
              balanced_state:
                $ref: '#/components/schemas/EmotionalState'
              balancing_techniques:
                type: array
                items: { type: string }
              time_required:
                type: number
              wisdom_insights:
                type: array
                items: { type: string }
```