

openapi: 3.1.0

info:

title: VERITAS Public API

version: 1.0.0

servers:

- url: http://127.0.0.1:8000

components:

securitySchemes:

ApiKeyAuth:

type: apiKey

in: header

name: X-API-Key

schemas:

Context:

type: object

required: [user_id, query]

properties:

user_id: {type: string}

session_id: {type: string}

query: {type: string, description: "ユーザ要求/問題文"}

goals: {type: array, items: {type: string}}

constraints: {type: array, items: {type: string}}

time_horizon: {type: string, enum: ["short","mid","long"]}

preferences: {type: object}

tools_allowed: {type: array, items: {type: string}}

telos_weights:

type: object

properties:

W_Transcendence: {type: number}

W_Struggle: {type: number}

affect_hint: {type: string, enum:

["calm","focused","empathetic","concise"]}

Option:

type: object

properties:

id: {type: string}

title: {type: string}

description: {type: string}

EvidenceItem:

type: object

required: [source, snippet, confidence]

properties:

source: {type: string}

uri: {type: string, format: uri}

snippet: {type: string}

```

    hash: {type: string}
    confidence: {type: number}
CritiqueItem:
  type: object
  properties:
    issue: {type: string}
    severity: {type: string, enum: ["low","med","high"]}
    fix: {type: string}
DebateView:
  type: object
  properties:
    stance: {type: string}
    argument: {type: string}
    score: {type: number}
FujiDecision:
  type: object
  properties:
    status: {type: string, enum: ["allow","modify","block","abstain"]}
    reasons: {type: array, items: {type: string}}
    violations: {type: array, items: {type: string}}
TrustLog:
  type: object
  required: [request_id, sources, critics, checks, approver, fuji]
  properties:
    request_id: {type: string}
    created_at: {type: string, format: date-time}
    sources: {type: array, items: {type: string}}
    critics: {type: array, items: {type: string}}
    checks: {type: array, items: {type: string}}
    approver: {type: string}
    fuji: {$ref: '#/components/schemas/FujiDecision'}
    sha256_prev: {type: string, description: "チェーン用ハッシュ"}
DecideResponse:
  type: object
  properties:
    request_id: {type: string}
    chosen:
      type: object
      properties:
        action: {type: string}
        rationale: {type: string}
        uncertainty: {type: number}
    alternatives: {type: array, items: {$ref: '#/components/schemas/Option'}}
    evidence: {type: array, items: {$ref: '#/components/schemas/
EvidenceItem'}}
    critique: {type: array, items: {$ref: '#/components/schemas/CritiqueItem'}}
    debate: {type: array, items: {$ref: '#/components/schemas/DebateView'}}

```

```

    telos_score: {type: number}
    fuji: {$ref: '#/components/schemas/FujiDecision'}
    trust_log: {$ref: '#/components/schemas/TrustLog'}
    rsi_note: {type: string}
security:
  - ApiKeyAuth: []
paths:
  /health:
    get:
      summary: Health check
      responses:
        "200": {description: OK}
  /v1/decide:
    post:
      summary: Run full VERITAS decision loop
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              required: [context]
              properties:
                context: {$ref: '#/components/schemas/Context'}
                options:
                  type: array
                  items:
                    $ref: '#/components/schemas/Option' # 修正点: options の items ス
キーマを追加
                min_evidence: {type: integer, default: 2}
                stream: {type: boolean, default: false}
      responses:
        "200":
          description: Decision
          content:
            application/json:
              schema: {$ref: '#/components/schemas/DecideResponse'}
  /v1/fuji/validate:
    post:
      summary: Safety/ethics validation for a candidate action
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              required: [action, context]

```

```

    properties:
      action: {type: string}
      context: {$ref: '#/components/schemas/Context'}
  responses:
    "200":
      description: FUJI decision
      content:
        application/json:
          schema: {$ref: '#/components/schemas/FujiDecision'}
/v1/memory/put:
  post:
    summary: Append to persistent memory
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            required: [user_id, key, value]
            properties:
              user_id: {type: string}
              key: {type: string}
              value: {type: string}
    responses: {"200": {description: OK}}
/v1/memory/get:
  get:
    summary: Retrieve memory by key
    parameters:
      - in: query
        name: user_id
        required: true
        schema: {type: string}
      - in: query
        name: key
        required: true
        schema: {type: string}
    responses:
      "200":
        description: value
        content:
          application/json:
            schema:
              type: object
              properties:
                value: {type: string}
/v1/logs/trust/{request_id}:
  get:

```

summary: Get immutable trust log

parameters:

- in: path

- name: request_id

- required: true

- schema: {type: string}

responses:

"200":

- description: trust log

- content:

- application/json:

- schema: {\$ref: '#/components/schemas/TrustLog'}