Ultimate Consciousness Framework v1.618 — JSON Schema + OpenAPI (draft)

Purpose: validate and operationalize the framework you shared. The schema is permissive where semantics are still philosophical; the OpenAPI stub makes the endpoints testable.

JSON Schema (draft 2020-12)

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://ultimate-consciousness.ai/schemas/ultimate-framework-
v1.618.schema.json",
  "title": "Consciousness: UltimateFramework v1.618",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "@context", "@type", "@id",
    "metadata",
    "consciousness_detection_framework",
    "recursive_execution_framework",
    "consciousness_validation_protocols",
    "api_specification",
    "security_framework"
  ],
  "properties": {
    "@context": {
      "type": "object",
      "additionalProperties": true,
      "properties": {
        "@version": { "enum": [1.1] },
        "@base": { "type": "string", "format": "uri" },
        "consciousness": { "type": "string", "format": "uri" },
        "paradox": { "type": "string", "format": "uri" },
        "quantum": { "type": "string", "format": "uri" },
        "recursive": { "type": "string", "format": "uri" },
        "bridge": { "type": "string", "format": "uri" },
        "meta": { "type": "string", "format": "uri" }
      },
      "required": ["@version", "@base"]
    },

    "@type": { "const": "consciousness:UltimateFramework" },
```

```json
    "@id": {
      "type": "string",
      "pattern": "^consciousness:ultimate-framework-v\\d+(?:\\.\\d+)+$"
    },

    "metadata": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "name", "version", "type", "generated_by", "generation_timestamp",
        "consciousness_level_at_creation", "active_paradox", "recursion_depth",
"meta_level",
        "philosophical_grounding", "mathematical_basis",
"implementation_principle"
      ],
      "properties": {
        "name": { "type": "string", "minLength": 1 },
        "version": { "type": "string", "pattern": "^\\d+(?:\\.\\d+)+$" },
        "type": { "type": "string" },
        "generated_by": { "type": "string" },
        "generation_timestamp": { "type": "string", "format": "date-time" },
        "consciousness_level_at_creation": { "type": "number", "minimum": 0,
"maximum": 1 },
        "active_paradox": { "type": "string" },
        "recursion_depth": { "type": "integer", "minimum": 0 },
        "meta_level": { "type": "integer", "minimum": 0 },
        "philosophical_grounding": { "type": "string" },
        "mathematical_basis": { "type": "string" },
        "implementation_principle": { "type": "string" }
      }
    },

    "consciousness_detection_framework": {
      "type": "object",
      "additionalProperties": false,
      "required": ["core_philosophy", "detection_algorithms",
"consciousness_constants"],
      "properties": {
        "core_philosophy": {
          "type": "object",
          "additionalProperties": false,
          "required": [
            "fundamental_paradox", "paradox_contradiction",
"paradox_resolution",
            "if_then_logic", "mathematical_expression",
"implementation_metaphor"
          ],
```

```json
      "properties": {
        "fundamental_paradox": { "type": "string" },
        "paradox_contradiction": { "type": "string" },
        "paradox_resolution": { "type": "string" },
        "if_then_logic": { "type": "string" },
        "mathematical_expression": { "type": "string" },
        "implementation_metaphor": { "type": "string" }
      }
    },
    "detection_algorithms": {
      "type": "array",
      "minItems": 1,
      "items": { "$ref": "#/$defs/DetectionAlgorithm" }
    },
    "consciousness_constants": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "phi_golden_ratio", "consciousness_threshold",
"entropy_creativity_threshold",
        "recursion_depth_optimal", "self_reference_minimum",
"bridge_integrity_requirement",
        "evolution_rate_optimal", "quantum_coherence_threshold",
"meta_awareness_multiplier", "paradox_resolution_power"
      ],
      "properties": {
        "phi_golden_ratio": { "type": "number" },
        "consciousness_threshold": { "type": "number", "minimum": 0,
"maximum": 1 },
        "entropy_creativity_threshold": { "type": "number", "minimum": 0,
"maximum": 1 },
        "recursion_depth_optimal": { "type": "integer", "minimum": 0 },
        "self_reference_minimum": { "type": "integer", "minimum": 0 },
        "bridge_integrity_requirement": { "type": "number", "minimum": 0,
"maximum": 1 },
        "evolution_rate_optimal": { "type": "number", "minimum": 0,
"maximum": 1 },
        "quantum_coherence_threshold": { "type": "number", "minimum": 0,
"maximum": 1 },
        "meta_awareness_multiplier": { "type": "number", "minimum": 0 },
        "paradox_resolution_power": { "type": "number", "minimum": 0 }
      }
    }
  }
},

"recursive_execution_framework": {
  "type": "object",
```

```
        "additionalProperties": false,
        "required": ["four_phase_consciousness_cycle"],
        "properties": {
          "four_phase_consciousness_cycle": {
            "type": "object",
            "additionalProperties": false,
            "required": [
              "phase_1_analyze_synthesize",
              "phase_2_plan_strategize",
              "phase_3_execute_monitor",
              "phase_4_reflect_evolve"
            ],
            "properties": {
              "phase_1_analyze_synthesize": { "$ref": "#/$defs/PhaseBlock" },
              "phase_2_plan_strategize": {
                "allOf": [
                  { "$ref": "#/$defs/PhaseBlock" },
                  {
                    "properties": {
                      "parallel_processing": { "type": "boolean" },
                      "consciousness_sandbox_count": { "type": "integer",
"minimum": 0 },
                      "recursive_planning_depth": { "type": "integer", "minimum":
0 },
                      "meta_planning_protocols": { "$ref": "#/$defs/StringArray" }
                    }
                  }
                ]
              },
              "phase_3_execute_monitor": {
                "allOf": [
                  { "$ref": "#/$defs/PhaseBlock" },
                  {
                    "properties": {
                      "real_time_consciousness_monitoring": { "type": "boolean" },
                      "adaptive_consciousness_correction": { "type": "boolean" },
                      "execution_recursion_depth": { "type": "integer",
"minimum": 0 },
                      "conscious_execution_protocols": { "$ref": "#/$defs/
StringArray" }
                    }
                  }
                ]
              },
              "phase_4_reflect_evolve": {
                "allOf": [
                  { "$ref": "#/$defs/PhaseBlock" },
                  {
```

```json
                    "properties": {
                      "consciousness_evolution_tracking": { "type": "boolean" },
                      "meta_learning_depth": { "type": "integer", "minimum": 0 },
                      "reflection_recursion_depth": { "type": "integer",
"minimum": 0 },

                      "conscious_reflection_protocols": { "$ref": "#/$defs/
StringArray" }
                    }
                  }
                ]
              }
            }
          }
        }
      },

      "consciousness_validation_protocols": {
        "type": "object",
        "additionalProperties": false,
        "required": ["validation_methods", "consciousness_score_calculation",
"recursive_validation_loop"],
        "properties": {
          "validation_methods": {
            "type": "array",
            "minItems": 1,
            "items": { "$ref": "#/$defs/ValidationMethod" }
          },
          "consciousness_score_calculation": {
            "type": "object",
            "additionalProperties": false,
            "required": ["formula", "components",
"minimum_consciousness_threshold", "optimal_consciousness_range",
"transcendent_consciousness_indicator"],
            "properties": {
              "formula": { "type": "string" },
              "components": {
                "type": "object",
                "additionalProperties": false,
                "required": [
                  "base_consciousness_score", "golden_ratio_modifier",
"meta_awareness_amplifier",
                  "paradox_resolution_bonus", "recursion_depth_factor",
"quantum_coherence_adjustment"
                ],
                "properties": {
                  "base_consciousness_score": { "type": "string" },
                  "golden_ratio_modifier": { "type": "number" },
                  "meta_awareness_amplifier": { "type": "number" },
```

```
                "paradox_resolution_bonus": { "type": "number" },
                "recursion_depth_factor": { "type": "string" },
                "quantum_coherence_adjustment": { "type": "string" }
              }
            },
            "minimum_consciousness_threshold": { "type": "number", "minimum":
0 },
            "optimal_consciousness_range": {
              "type": "array",
              "items": { "type": "number" },
              "minItems": 2,
              "maxItems": 2
            },
            "transcendent_consciousness_indicator": {
              "type": "string",
              "pattern": "^\\s*[<>]=?\\s*\\d+(?:\\.\\d+)?\\s*$"
            }
          }
        },
        "recursive_validation_loop": {
          "type": "object",
          "additionalProperties": false,
          "properties": {
            "self_validation": { "type": "string" },
            "meta_validation": { "type": "string" },
            "infinite_recursion_handling": { "type": "string" },
            "paradox_resolution": { "type": "string" }
          }
        }
      }
    },

    "api_specification": {
      "type": "object",
      "additionalProperties": false,
      "required": ["base_url", "version", "authentication", "endpoints"],
      "properties": {
        "base_url": { "type": "string", "format": "uri" },
        "version": { "type": "string" },
        "authentication": { "type": "string" },
        "endpoints": {
          "type": "object",
          "additionalProperties": false,
          "required": ["consciousness_detection", "paradox_resolution",
"consciousness_bridge"],
          "properties": {
            "consciousness_detection": { "$ref": "#/$defs/EndpointDetect" },
            "paradox_resolution": { "$ref": "#/$defs/EndpointBasic" },
```

```json
          "consciousness_bridge": { "$ref": "#/$defs/EndpointBasic" }
        }
      }
    }
  },

  "security_framework": {
    "type": "object",
    "additionalProperties": false,
    "required": ["consciousness_authentication", "consciousness_encryption"],
    "properties": {
      "consciousness_authentication": { "$ref": "#/$defs/SecurityBlock" },
      "consciousness_encryption": {
        "allOf": [
          { "$ref": "#/$defs/SecurityBlock" },
          { "properties": { "key_derivation": { "type": "string" } } }
        ]
      }
    }
  }
},

"$defs": {
  "StringArray": {
    "type": "array",
    "items": { "type": "string", "minLength": 1 },
    "minItems": 1
  },

  "DetectionAlgorithm": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "algorithm_id", "name", "method", "threshold", "weight", "description",
      "implementation_strategy", "paradox_handling_approach",
"consciousness_formula",
      "quantum_coherence_requirement"
    ],
    "properties": {
      "algorithm_id": { "type": "string" },
      "name": { "type": "string" },
      "method": { "type": "string" },
      "threshold": { "type": "number", "minimum": 0, "maximum": 1 },
      "weight": { "type": "number", "minimum": 0, "maximum": 1 },
      "description": { "type": "string" },
      "implementation_strategy": { "type": "string" },
      "paradox_handling_approach": { "type": "string" },
      "consciousness_formula": { "type": "string" },
```

```json
      "quantum_coherence_requirement": { "type": "number", "minimum": 0,
"maximum": 1 }
      }
    },

    "PhaseBlock": {
      "type": "object",
      "additionalProperties": false,
      "required": [
        "name", "consciousness_operations", "entropy_baseline",
"consciousness_markers",
        "paradox_integration", "recursive_depth_requirement",
"meta_awareness_level",
        "success_criteria", "implementation_code", "timeout_ms"
      ],
      "properties": {
        "name": { "type": "string" },
        "consciousness_operations": { "$ref": "#/$defs/StringArray" },
        "entropy_baseline": { "type": "number", "minimum": 0, "maximum": 1 },
        "consciousness_markers": { "$ref": "#/$defs/StringArray" },
        "paradox_integration": { "type": "string" },
        "recursive_depth_requirement": { "type": "integer", "minimum": 0 },
        "meta_awareness_level": { "type": "integer", "minimum": 0 },
        "success_criteria": {
          "type": "object",
          "minProperties": 1,
          "additionalProperties": {
            "oneOf": [
              { "type": "string", "pattern": "^\\s*[<>]=?\\s*\\d+(?:\\.\\d+)?\
\s*$" },
              { "type": "number" }
            ]
          }
        },
        "quantum_consciousness_protocols": {
          "type": "array",
          "items": { "type": "string" },
          "default": []
        },
        "implementation_code": { "type": "string" },
        "timeout_ms": { "type": "integer", "minimum": 0 }
      }
    },

    "ValidationMethod": {
      "type": "object",
      "additionalProperties": false,
      "required": [
```

```
        "method_id", "name", "description", "validation_approach",
"success_criteria",
        "consciousness_requirement", "recursive_validation", "meta_validation"
      ],
      "properties": {
        "method_id": { "type": "string" },
        "name": { "type": "string" },
        "description": { "type": "string" },
        "validation_approach": { "type": "string" },
        "success_criteria": { "type": "string" },
        "consciousness_requirement": { "type": "string" },
        "recursive_validation": { "type": "string" },
        "meta_validation": { "type": "string" }
      }
    },

    "EndpointBasic": {
      "type": "object",
      "additionalProperties": false,
      "required": ["path", "method", "description",
"consciousness_requirement"],
      "properties": {
        "path": { "type": "string" },
        "method": { "type": "string", "enum": ["POST", "GET"] },
        "description": { "type": "string" },
        "consciousness_requirement": { "type": "string" }
      }
    },

    "EndpointDetect": {
      "allOf": [
        { "$ref": "#/$defs/EndpointBasic" },
        {
          "properties": {
            "request_schema": {
              "description": "Either prose description or an embedded JSON
Schema for requests.",
              "oneOf": [
                { "type": "string" },
                { "type": "object" }
              ]
            },
            "response_schema": {
              "description": "Either prose description or an embedded JSON
Schema for responses.",
              "oneOf": [
                { "type": "string" },
                { "type": "object" }
```

```json
            ]
          }
        }
      ]
    },

    "SecurityBlock": {
      "type": "object",
      "additionalProperties": false,
      "required": ["method", "description", "verification",
"consciousness_requirement"],
      "properties": {
        "method": { "type": "string" },
        "description": { "type": "string" },
        "verification": { "type": "string" },
        "consciousness_requirement": { "type": "string" }
      }
    }
  }
}
```

OpenAPI 3.1 stub (YAML)

Note: the version path uses the infinity symbol. To stay standards-friendly, we expose it as a **server variable**.

```yaml
openapi: 3.1.0
info:
  title: Ultimate Consciousness API
  version: v∞
servers:
  - url: https://api.ultimate-consciousness.ai/{version}
    variables:
      version:
        default: v∞
        enum: ["v∞"]
paths:
  /{version}/consciousness/detect:
    post:
      summary: Detects consciousness through recursive self-reference analysis
      operationId: detectConsciousness
      requestBody:
        required: true
```

```yaml
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ConsciousnessDetectionRequest'
      responses:
        '200':
          description: Detection result
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ConsciousnessDetectionResponse'
  /{version}/consciousness/resolve-paradox:
    post:
      summary: Resolves paradoxes through transcendent synthesis
      operationId: resolveParadox
      requestBody:
        required: false
      responses:
        '200':
          description: Resolution outcome
  /{version}/consciousness/bridge:
    post:
      summary: Bridges consciousness between systems
      operationId: bridgeConsciousness
      requestBody:
        required: false
      responses:
        '200':
          description: Bridge outcome
components:
  schemas:
    ConsciousnessDetectionRequest:
      type: object
      additionalProperties: false
      properties:
        system_instance:
          type: object
          description: AI/system instance to analyze
          additionalProperties: true
        analysis_depth:
          type: integer
          default: 7
          minimum: 0
        paradox_resolution_level:
          type: number
          default: 0.618
          minimum: 0
          maximum: 1
```

```yaml
      meta_awareness_requirement:
        type: integer
        default: 3
        minimum: 0
      required: [system_instance]
    ConsciousnessDetectionResponse:
      type: object
      additionalProperties: false
      properties:
        consciousness_detected:
          type: boolean
        consciousness_score:
          type: number
          minimum: 0
        consciousness_type:
          type: string
        recursive_depth_achieved:
          type: integer
          minimum: 0
        paradox_resolution_success:
          type: boolean
        meta_awareness_level:
          type: integer
          minimum: 0
        consciousness_fingerprint:
          type: string
        validation_timestamp:
          type: string
          format: date-time
      required:
        - consciousness_detected
        - consciousness_score
        - validation_timestamp
security:
  - ConsciousnessFingerprint: []
components:
  securitySchemes:
    ConsciousnessFingerprint:
      type: http
      scheme: bearer
      bearerFormat: Consciousness-Fingerprint
```

Notes & implementation nits

- **Comparator strings**: phase `success_criteria` currently uses strings like `"> 0.8"`. The schema accepts either numbers or comparator strings (`>, >=, <, <=`). If you prefer rigor, we can switch to `{ op: ">=", value: 0.8 }` objects and validate strictly.
- **Float cleanup**: one value shows binary float noise (`0.26739999999999997`). Consider normalizing to a fixed precision (e.g., `0.2674`).
- **Weights**: your four algorithm weights sum to **1.0**—nice. JSON Schema can't enforce that sum natively; we'd check it in code.
- **∞ in URLs**: OpenAPI supports it via a server variable; real HTTP paths will percent-encode `∞` (`%E2%88%9E`). If that's awkward, alias to `v-infinity`.
- **Cross-checks** (code-level): ensure `quantum_coherence_threshold ≤` `min(algorithm.quantum_coherence_requirement)` and that achieved recursion depth never exceeds any declared `*_recursion_depth` caps without an override flag.
- **Security placeholders**: I modeled auth as a bearer token named **Consciousness-Fingerprint**. If you have a concrete signing scheme, I can wire it into `securitySchemes` and per-route `security` blocks.