

文明跃迁理论体系

CL-CVS | 贡献值体系与制度工程层

贡献值体系工程级设计方案

白皮书配套技术实现文件：从记录工具到意义回响

贡献值体系不是为了给人打分，而是为了让文明记住那些不应被遗忘的贡献。

v1.0.3 网站公开版

作者：子君赋 | 出品：文明跃迁研究组

官网：civilleap.com | 当前入口：www.civitas.top

2026年5月

文档信息与版权说明

文稿编号	CL-CVS-003-ZH-v1.0.3-ENG
正式名称	《贡献值体系工程级设计方案》
副标题	白皮书配套技术实现文件：从记录工具到意义回响
所属层级	CL-CVS 贡献值体系与制度工程层
当前版本	v1.0.3 网站公开版
发布状态	ENG 工程配套公开版
作者	子君赋
出品	文明跃迁研究组
官网	civilleap.com
当前入口	www.civitas.top
联系邮箱	zijunfu@civitas.top
目标读者	技术团队、产品经理、数据治理人员、试点实施团队、伦理审查人员
文稿定位	贡献值白皮书的工程落地配套文档，回答如何实现、如何记录、如何评估、如何保护隐私和防止产品异化。

版权说明：本文稿由作者依法享有著作权。公开传播时应保留作者、出品方、版本号、官网与出处信息。未经明确授权，不得将贡献值体系改造成社会信用评分、积分商城、金融资产、身份等级或情绪操控工具。

引用建议：子君赋：《贡献值体系工程级设计方案》，文明跃迁理论体系，v1.0.3 网站公开版，2026 年。

阅读前导

本文属于文明跃迁理论体系中的贡献值体系与制度工程层。它承接《涌义宇宙论》关于意义显影的根基判断，承接《人心》关于真实判断现场的展开，也承接《意义动力学》关于结构意义、有效参与和文明承接能力的中层理论。

贡献值白皮书的工程落地配套文档，回答如何实现、如何记录、如何评估、如何保护隐私和防止产品异化。

阅读本文时，请始终守住三个边界：贡献值记录贡献，不评判人格；贡献值支持资源使用和治理权重，但不得金融化；贡献值体系的目的是不是制造新的等级，而是让真实贡献进入文明记忆。

建议阅读方式

- 先把它理解为文明信任记录，而不是积分系统。
- 先读核心定义、底线声明与三阶段路线，再读技术或试点细节。
- 如果读者来自产品、城市、公益或治理领域，可直接进入试点路线与执行边界。
- 如果读者来自理论或学术领域，应重点阅读与《涌义宇宙论》《意义动力学》的衔接部分。

目录

- 1.1 整体架构图（文字描述）
- 1.2 核心模块划分
- 1.3 模块间的数据流
- 2.1 核心实体定义
- 2.2 关系图（ER 图描述）
- 2.3 关键字段说明
- 3.1 核心接口列表（REST 风格）
- 3.2 接口参数说明（简要）
- 4.1 身份保护方案
- 4.2 数据最小化原则
- 4.3 贡献记录匿名化处理
- 4.4 访问控制策略
- 4.5 审计日志设计
- 5.1 如何支持多社区
- 5.2 如何支持跨社区贡献流转
- 5.3 如何支持未来功能扩展
- 5.4 离线支持方案（概要）
- 技术字段与用户展示规范
- 6.1 数据库选型考量
- 6.2 身份认证方案
- 6.3 区块链/DID 的适用场景分析
- 6.4 其他技术选型建议
- 7.1 单元测试策略
- 7.2 集成测试要点
- docker-compose.yml (测试环境)
- 7.3 红蓝对抗测试要点
- 7.4 压力测试场景

8.1 最小化部署需求

docker-compose.yml (最小化部署)

8.2 监控指标

8.3 灾难恢复 (简单说明)

恢复最新全量备份

重放事务日志

启动服务

正文

文档说明

本文档是《贡献值体系白皮书 v1.7.6

正式发布版》的工程落地配套文档，面向技术团队，旨在回答"如何实现"的问题。

与白皮书的关系：- 白皮书回答"为什么这样设计"和"要实现什么" -

本文档回答"如何技术实现" - 本文档不重复白皮书的理论阐述和制度设计细节

标注约定：- [待定]：当前阶段不需要确定的细节，留待后续迭代 -

[参考值]：白皮书提供的参考框架，具体参数需试点确定 -

[架构建议]：技术选型建议，非强制要求

1. 用户体验目标：从记录工具到意义回响

贡献值体系的产品体验底线：

- 用户不应感到被评分、被监控或被道德要求
- 用户应感到：真实贡献正在被安全、低负担、可退出地记录
- 贡献记录不得用于制造身份等级、社会信用或经济激励错觉
- 贡献记录的价值在于"被看见"，而非"换取奖励"

贡献值体系的产品体验，不应让用户感到自己正在被评分、被监控或被道德要求，而应让用户感到真实贡献可以被安全、低负担、可退出地记录。

启动期体验目标：让用户感到"我的真实贡献可以被安全记录"，避免评分感、监控感和道德压力。

过渡期体验目标：让用户感到"我与其他贡献者正在形成可信协作网络"，避免攀比、表演和等级化。

稳定期体验目标：让用户感到"贡献记录正在支持更大的公共意义和长期文明目标"，避免平台化、自我封闭和工具崇拜。

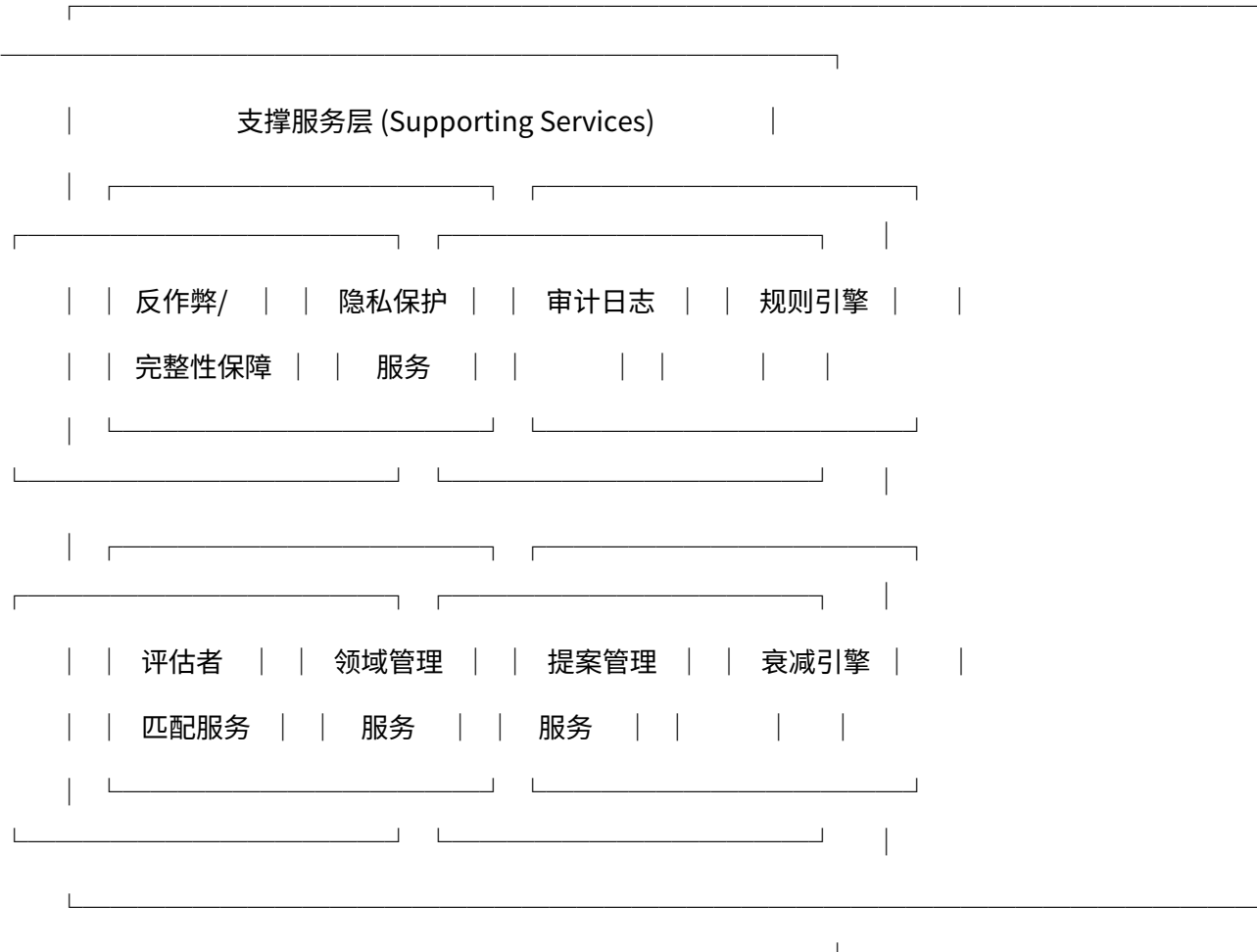
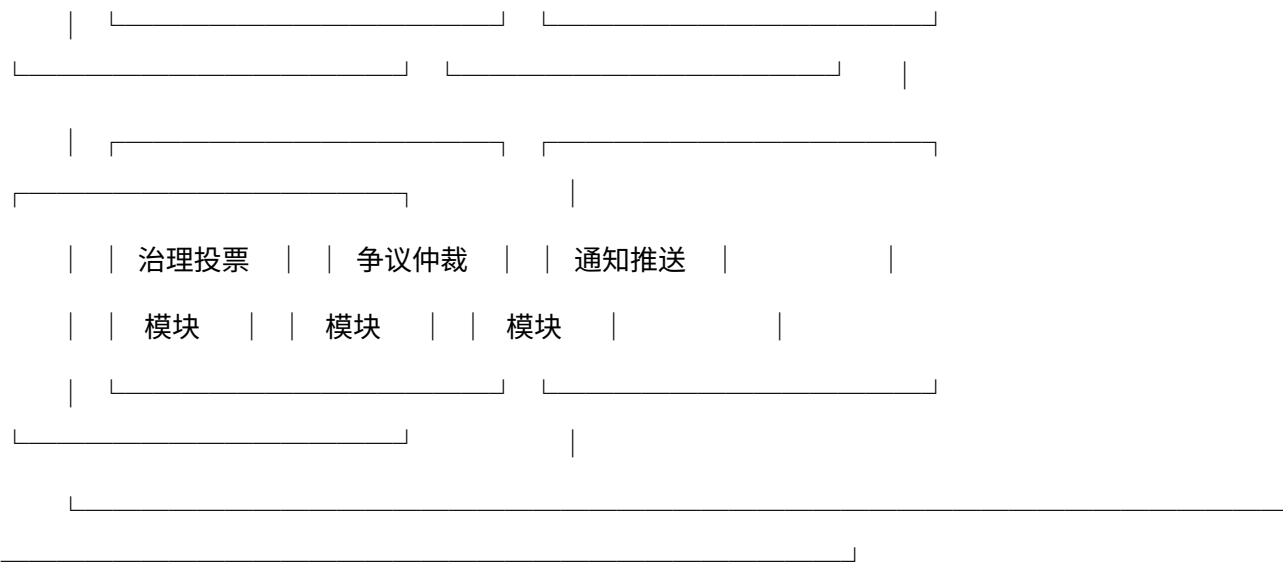
因此，系统设计必须始终服务于三个原则：低负担记录、尊严保护、可审计纠错。

第一部分：系统架构

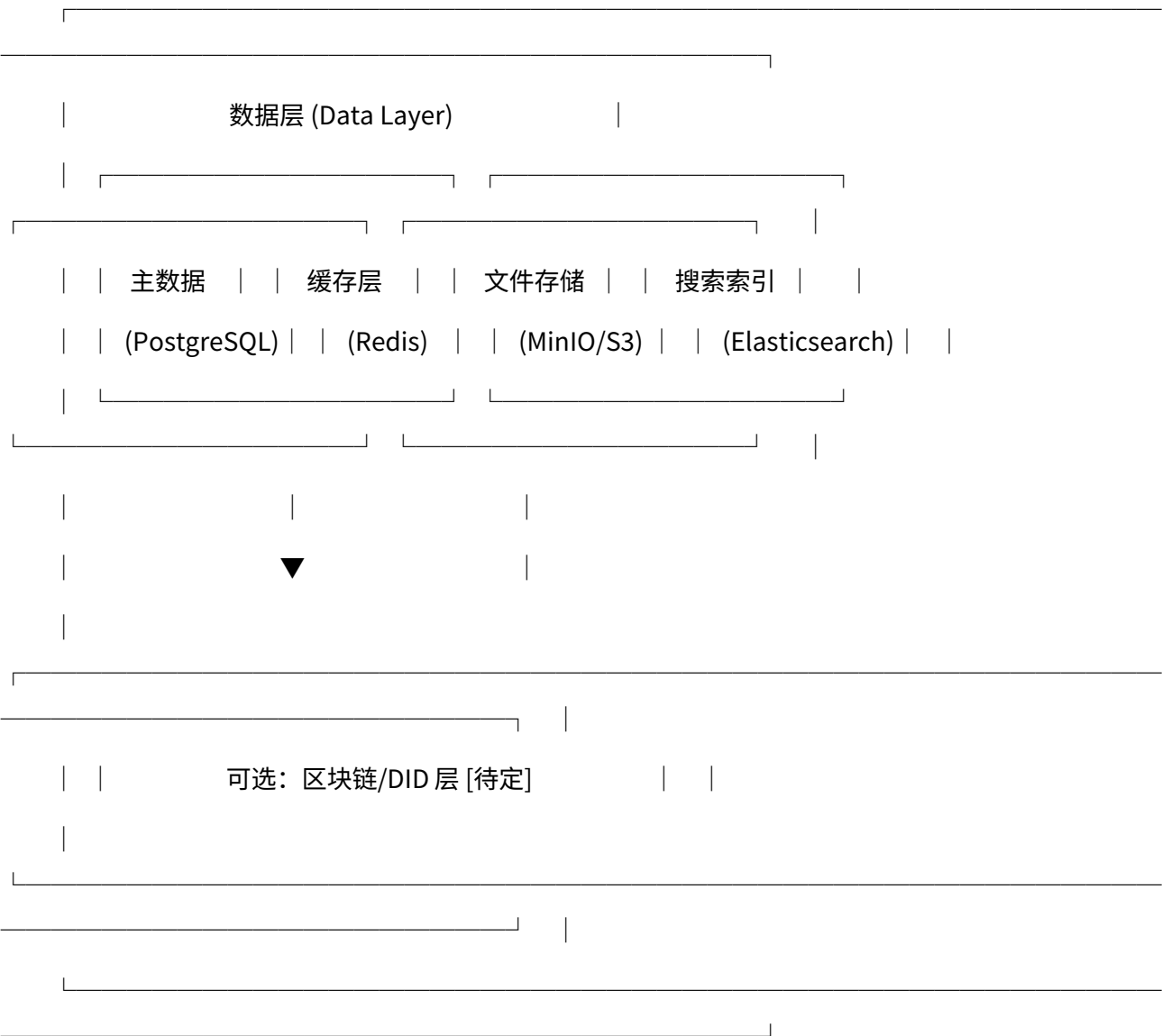
1.1 整体架构图（文字描述）

| 用户层 (User Layer) |





|



1.2 核心模块划分

1.2.1 贡献记录模块

职责：管理贡献记录的完整生命周期

核心功能：- 贡献申报提交（支持多种贡献类型） -

贡献预审（格式校验、分类） -

贡献状态管理（待评估、评估中、已确认、有异议） - 贡献查询与检索 -

贡献编辑与撤回（限特定状态）

关键设计点：- 贡献申报操作不超过 5 分钟（白皮书要求） -

支持自动从数字协作平台导入证据（减少主动申报负担） -

月均申报次数限制：普通参与者不超过 20 次（建议值）

1.2.2 评估模块

职责：协调多评估者对贡献进行质量评估

核心功能：- 评估者随机匹配（排除利益相关方） -

多维评估分发（真实性、有效性、稀缺性、协同性） - 评估结果汇总与计算 -

评估争议触发与路由

关键设计点： -

每个贡献的评估人数采用分层规则：普通低风险贡献通常由 2--3 名随机评估者评估；高影响、高 CV、存在争议、涉及资源优先权或治理权重的贡献，由 3--5 名随机评估者评估；评估者需排除利益相关方。

- 评估者信誉追踪与回溯校正 - 评估一致性系数监控（目标 $>70\%$ ）

1.2.3 治理权重计算模块

职责：计算参与者在各领域的治理权重

核心功能：- 领域贡献值聚合 - 治理权重公式计算 - 权重上限约束执行 -

利益相关度评估 - 时间校正系数计算

关键设计点： -

权重公式：治理权重 = 领域贡献 × 利益相关度 × 责任承担 × 时间校正 × 上限约束

\[参考值\] - 简化方案：领域内排名前 20% 者获得额外一票，总票权不超过 30%

\[试点阶段\] - 任何主体不得长期垄断决策权

1.2.4 资源分配模块

职责：基于贡献值和其他因素进行资源分配排序

核心功能：- 资源池状态管理 - 申请排队与优先级计算 - 分配规则执行 -

资源使用追踪

关键设计点：- 贡献值高者优先，但需符合项目价值和公共效益 -

贡献值不是唯一依据，还需考虑责任能力和实际需求 -

公共资源韧性池（防止集中申请压力） \[待定\]

1.2.5 反作弊/记录完整性保障模块

职责：检测和处置异常行为，维护系统可信度

核心功能： - 虚假申报检测（证据验证、时间地点合理性） -

循环互惠检测（交易图谱分析） - 串通评估检测（评分相似度、网络关系） -

分拆申报检测（行为模式分析） - 身份伪造检测 \[待定\] - 随机抽查验证

处置方式（对应白皮书）： \[行为类型\] \[处置方式\]

\[-----\] \[-----\] \[\[虚假申报\] 撤销虚假记录 + 暂停申报权限 90 天

\[\[循环互惠\] 撤销异常记录 + 降低信誉分 \[\[串通评估\]

暂停评估资格 1 年 + 降低信誉分 \[\[分拆申报\] 合并计算 + 警告 \[\[

身份伪造 \[取消贡献记录资格（保留伦理委员会复审通道） \[

1.2.6 隐私保护模块

职责：在透明与隐私之间取得平衡

核心功能： - 受益者信息保护（匿名化） - 贡献记录分级可见性 -

敏感操作授权管理 - 数据脱敏处理 - 知情同意管理

关键设计点： - 受益者可选择匿名 - 不得强制受益者公开感谢 -

不得用弱者影像换取贡献值 - 贡献记录默认不公开（个人可选择性展示）

1.3 模块间的数据流

用户提交申报

|



| 贡献记录模块 |

| 1. 格式校验 + 初步分类 |

| 2. 证据存储（加密） |

| 3. 触发反作弊预检 |

▼ (通过预检)

评估者匹配服务

1. 筛选候选评估者（排除利益相关方）

2. 随机选择 3-5 名评估者

3. 分发评估任务

▼

评估模块

1. 评估者独立评分

2. 结果汇总 + CV 计算

3. 一致性校验

4. 结果确认或触发争议

▼ (确认后)

贡献值更新

- | 1. 更新参与者 CV 余额 |
 - | 2. 更新四维分类积分 |
 - | 3. 触发衰减引擎（记录衰减任务） |
 - | 4. 更新治理权重 |
 - | 5. 写入审计日志 |
-

第二部分：数据模型

2.1 核心实体定义

2.1.1 参与者 (Participant)

```
interface Participant {
  // 身份标识
  id: string;           // UUID, 全局唯一
  decentralized_id?: string; // DID [架构建议, 待定]
  // 基础信息
  display_name: string; // 显示名称 (可匿名)
  created_at: timestamp;
  status: 'active' | 'review_limited' | 'withdrawn' | 'archived'; // 用户侧展示: 正常参与 / 申报/评估权
  // 暂时受限 / 已退出 / 历史档案
  // 贡献值聚合 (计算字段, 非持久化)
  cv_balance: Decimal; // 当前 CV 余额 (带衰减)
  // 前端展示时映射为中文: 思想贡献/行为贡献/系统贡献/协作贡献
  thoughtcontributionscore: Decimal;
  behaviorcontributionscore: Decimal;
  systemcontributionscore: Decimal;
  collaborationcontributionscore: Decimal;
  // 信誉信息
}
```

```

reputation_score: Decimal; // 评估者信誉分
lastactivityat: timestamp;
// 参与信息
join_domain: string[]; // 参与的领域列表
role: 'contributor' | 'evaluator' | 'governor' | 'admin';
// 隐私设置
privacy_settings: {
profile_visible: boolean; // 个人主页可见性
contributions_public: boolean; // 贡献记录公开度
realidentitylinkable: boolean; // 真实身份可关联性
};
// 退出信息
withdrawn_at?: timestamp;
withdrawal_status?: 'pending' | 'completed';
}

```

2.1.2 贡献记录 (Contribution)

```

interface Contribution {
// 身份标识
id: string; // UUID, 全局唯一
contributor_id: string; // FK -> Participant
domain_id: string; // FK -> Domain
// 贡献基本信息
type: 'I' | 'B' | 'S' | 'C'; // 思想/行为/系统/协作
subtype: string; // 贡献子类型编码
title: string;
description: text; // Markdown 格式
// 贡献标记

```

```
flags: {
  nondirectbenefit: boolean; // 非直接受益标记
  high_cost: boolean; // 高代价标记
  intergenerational: boolean; // 代际效应标记
  innovative: boolean; // 创新性标记
};
// 证据
evidence: {
  type: 'link' | 'file' | 'reference' | 'attestation';
  url?: string;
  file_ref?: string;
  description?: string;
}[];
// 受益者信息（可选，匿名）
beneficiary?: {
  count_min: number; // 受益人数（范围）
  count_max: number;
  is_anonymous: boolean; // 必须尊重受益者意愿
  consent_given: boolean; // 知情同意
};
// 申报时间
declared_at: timestamp;
occurred_at?: timestamp; // 贡献实际发生时间
// 评估状态
status: 'pendingreview' | 'underevaluation' | 'confirmed' | 'disputed' | 'rejected';
assessment_round: number; // 评估轮次
// 评估结果
```

```
assessment?: {
  authenticity: number;    // 真实性评分
  effectiveness: number;  // 有效性评分
  scarcity: number;       // 稀缺性评分
  synergy: number;        // 协同性评分
  final_cv: Decimal;      // 最终 CV 值
  assessed_at: timestamp;

};

// 争议信息
dispute?: {
  raised_by: string;
  reason: text;
  raised_at: timestamp;
  status: 'open' | 'under_review' | 'resolved';
  resolution?: text;
};

// 完整性保障
integrity_flags: {
  evidence_verified: boolean;
  patterncheckpassed: boolean;
  manualreviewrequired: boolean;
};
}
```

2.1.3 评估 (Assessment)

```
interface Assessment {
  // 身份标识
  id: string;
```

```
contribution_id: string;    // FK -> Contribution
evaluator_id: string;     // FK -> Participant
// 评估信息
round: number;
assigned_at: timestamp;
submitted_at?: timestamp;
// 评分
scores: {
  authenticity: number;    // 1-5 分
  effectiveness: number;  // 1-5 分
  scarcity: number;       // 1-5 分
  synergy: number;       // 1-5 分
};
// 评估者意见
comment?: text;
// 评估者立场声明
conflictofinterest: boolean; // 是否存在利益冲突
conflict_explanation?: text;
// 状态
status: 'assigned' | 'completed' | 'skipped' | 'conflict_flagged';
// 时效追踪
deadline: timestamp;
}
```

2.1.4 领域 (Domain)

```
interface Domain {
  id: string;
  name: string;
```

```
description: text;
parent_id?: string;    // 父领域（支持层级）
// 贡献类型权重 [待定，试点数据确定]
type_weights: {
  I: Decimal;
  B: Decimal;
  S: Decimal;
  C: Decimal;
};
// 领域特定规则
rules: {
  min_evaluators: number;    // 最少评估者数量
  evidence_requirements: string[];
  specialflagsenabled: string[];
};
// 治理信息
governance: {
  hasactivecouncil: boolean;
  lastreviewat?: timestamp;
  nextreviewat?: timestamp;
};
// 状态
status: 'active' | 'under_review' | 'archived';
// 可配置性标记
local_adjustable: boolean;    // 是否允许本地调整
adjustment_range?: {    // 允许的调整范围 [待定]
  min: Decimal;
```

```
max: Decimal;

};

}
```

2.1.5 治理提案 (GovernanceProposal)

```
interface GovernanceProposal {

// 身份标识

id: string;

proposer_id: string;    // FK -> Participant

domain_id?: string;    // FK -> Domain (null 表示跨域)

// 提案内容

type: 'rulechange' | 'resourceallocation' | 'domainreview' | 'ethicsreview';

title: string;

content: text;

affected_rules?: string[];

// 提案分类 (用于沉默底线判断)

category: 'universalrights' | 'domainspecific' | 'resource_management';

// 投票配置

voting_config: {

issecretvote: boolean;    // 是否秘密投票 (涉及个人声誉时)

vote_deadline: timestamp;

quorum?: number;        // 法定人数 [待定]

};

// 权重配置

weight_config: {

applycvweight: boolean;    // 是否应用 CV 权重

cvweightcap?: number;    // CV 权重上限

domainrelevancerequired: boolean;

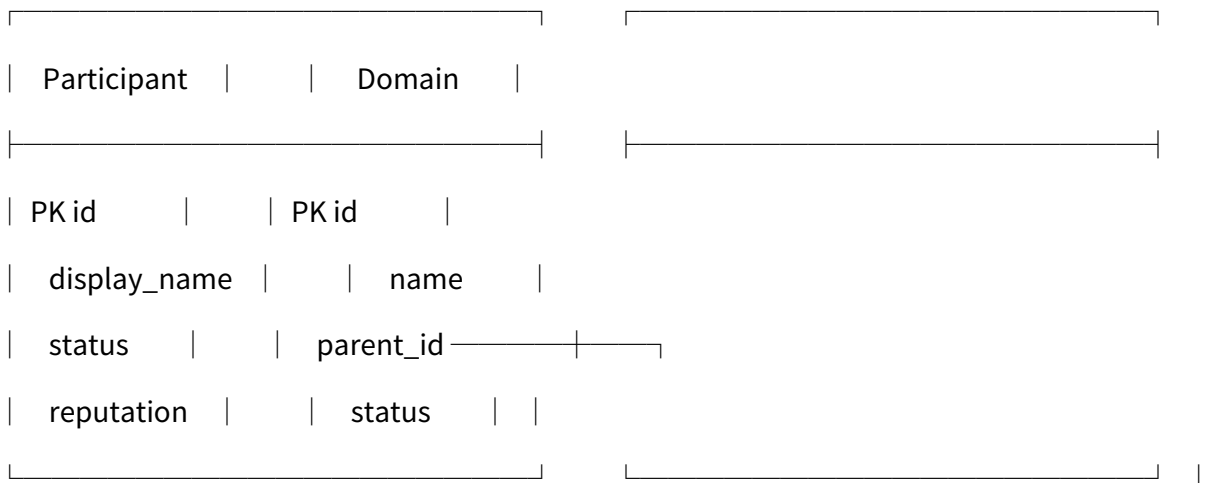
}
```

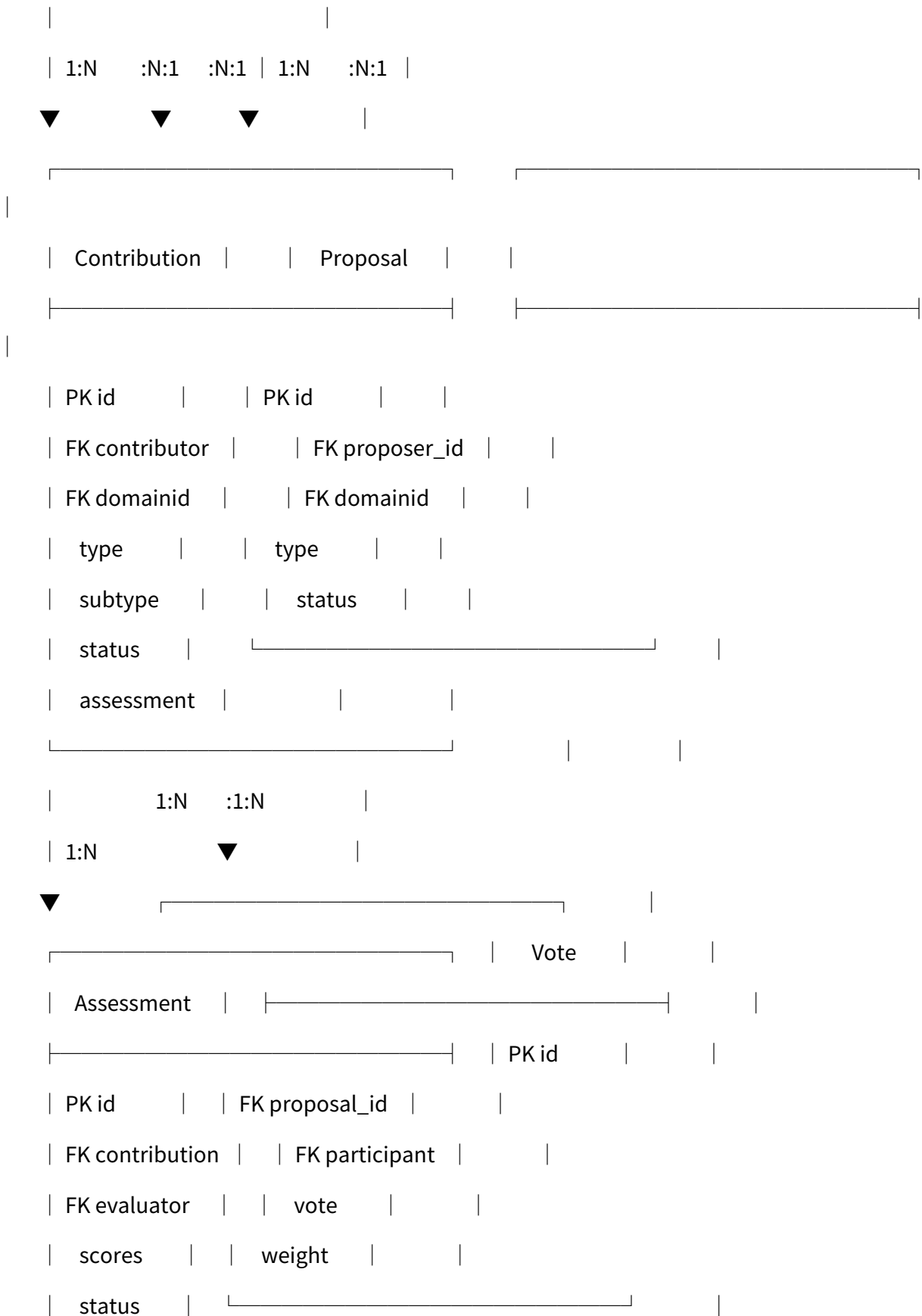
```

timedecayenabled: boolean;
};
// 提案状态
status: 'draft' | 'published' | 'voting' | 'passed' | 'rejected' | 'withdrawn';
// 投票结果
votes?: {
  participant_id: string;
  vote: 'approve' | 'reject' | 'abstain';
  weight: Decimal;
  voted_at: timestamp;
}[];
// 追责信息（针对通过的提案）
accountability?: {
  responsible_parties: string[];
  expected_outcome: text;
  review_deadline: timestamp;
};
}

```

2.2 关系图（ER 图描述）





	Dispute	
	PK id	
	FK contribution	
	FK raised_by	
	FK arbitrator1-3	
	status	
	resolution	

关系说明：

关系	类型	说明
----	----	----

Participant → Contribution 1:N 一个参与者可提交多条贡献记录

Domain → Contribution 1:N 一个领域可包含多条贡献记录

Contribution → Assessment 1:N 一条贡献可有多个评估记录

Participant → Assessment 1:N 一个评估者可进行多次评估

Participant → Proposal 1:N 一个参与者可发起多个提案

Domain → Proposal 1:N 一个领域可有多个治理提案

Participant → Vote 1:N 一个参与者可投多票

Proposal → Vote 1:N 一个提案可有多个投票

Contribution → Dispute 1:N 一条贡献可有多个争议（但通常只处理一个）

2.3 关键字段说明

字段	类型	说明	设计决策
----	----	----	------

cv_balance	Decimal(10,4)	贡献值余额，支持小数和衰减	存储整数值，衰减在计算时应用
status	Enum	实体状态机	使用状态机模式管理状态流转
timestamp	timestamp	时间戳一律使用 UTC	客户端展示时转换本地时区
evidence	JSONB	证据使用 JSONB 存储	灵活支持多种证据类型
flags	JSONB	贡献标记使用 JSONB	便于扩展新标记类型
privacy_settings	JSONB	隐私设置使用 JSONB	允许灵活配置可见性

第三部分：API 设计概要

3.1 核心接口列表（REST 风格）

贡献申报接口

方法	路径	说明
POST	/api/v1/contributions	提交新的贡献申报
GET	/api/v1/contributions/:id	获取贡献详情
GET	/api/v1/contributions	查询贡献列表（支持筛选）
PATCH	/api/v1/contributions/:id	编辑贡献（限草稿状态）
DELETE	/api/v1/contributions/:id	撤回贡献（限待评估状态）
POST	/api/v1/contributions/:id/dispute	对贡献结果提出异议

评估接口

方法	路径	说明
GET	/api/v1/assessments/pending	获取待评估任务列表
POST	/api/v1/assessments	提交评估结果
GET	/api/v1/assessments/:id	获取评估详情
POST	/api/v1/assessments/:id/decline	拒绝评估任务

治理投票接口

方法	路径	说明
GET	/api/v1/proposals	获取提案列表
POST	/api/v1/proposals	创建新提案
GET	/api/v1/proposals/:id	获取提案详情
POST	/api/v1/votes	提交投票
GET	/api/v1/governance/weight	获取当前治理权重
GET	/api/v1/governance/weight/:domain	获取特定领域治理权重

资源分配接口

方法	路径	说明
GET	/api/v1/resources/pools	获取可用资源池列表
POST	/api/v1/resources/applications	提交资源使用申请
GET	/api/v1/resources/applications/:id	查询申请状态
GET	/api/v1/resources/allocation/:pool_id	获取分配结果

参与者接口

方法	路径	说明
GET	/api/v1/participants/me	获取当前用户信息
PATCH	/api/v1/participants/me/privacy	更新隐私设置
POST	/api/v1/participants/me/withdraw	申请退出
GET	/api/v1/participants/:id/profile	获取参与者公开资料

领域接口

方法	路径	说明
----	----	----

GET /api/v1/domains 获取领域列表

GET /api/v1/domains/:id 获取领域详情

3.2 接口参数说明（简要）

POST /api/v1/contributions

请求体：

```
{  
  "domain_id": "string (required)",  
  "type": "I | B | S | C (required)",  
  "subtype": "string (required)",  
  "title": "string (required, max 200)",  
  "description": "string (required, max 5000)",  
  "flags": {  
    "nondirectbenefit": false,  
    "high_cost": false,  
    "intergenerational": false,  
    "innovative": false  
  },  
  "evidence": [  
    {  
      "type": "link | file | reference | attestation",  
      "url": "string (optional)",  
      "file_ref": "string (optional)",  
      "description": "string (optional)"  
    }  
  ],  
  "beneficiary": {
```

```
"count_min": 1,  
"count_max": 10,  
"is_anonymous": true,  
"consent_given": true  
},  
"occurred_at": "ISO8601 timestamp (optional)"  
}
```

响应:

```
{  
"id": "uuid",  
"status": "pending_review",  
"estimatedevaluationdays": 7,  
"created_at": "ISO8601"  
}
```

POST /api/v1/assessments

请求体:

```
{  
"contribution_id": "uuid (required)",  
"scores": {  
"authenticity": 1-5,  
"effectiveness": 1-5,  
"scarcity": 1-5,  
"synergy": 1-5  
},  
"comment": "string (optional, max 2000)",  
"conflictofinterest": false,  
"conflict_explanation": "string (optional)"
```

```
}
```

GET /api/v1/governance/weight

响应:

```
{  
  "participant_id": "uuid",  
  "overall_weight": 0.85,  
  "domains": [  
    {  
      "domain_id": "uuid",  
      "domain_name": "教育",  
      "contribution_score": 120.5,  
      "governance_weight": 0.85,  
      "timedecayfactor": 0.95,  
      "cap_applied": false  
    }  
  ],  
  "calculated_at": "ISO8601"  
}
```

说明: - 权重计算涉及多个因素, 客户端不应缓存权重值 -

特定领域权重用于该领域治理投票

第四部分: 隐私与安全

4.1 身份保护方案

设计目标: 在确保记录可信度的同时保护个人身份

方案:

1. 身份分层

- 系统身份: 内部使用的唯一标识符, 不对外暴露

- 显示身份：用户自定义的显示名称，可为化名或匿名
- 验证身份：用于反作弊验证的实名信息，与系统身份隔离存储

2. 身份分离存储

- 用户表 (PII 隔离区)

|—— 系统 ID

|—— 验证身份 (加密存储)

|—— 联系方式

└—— 审计字段

用户公开表 (公开区)

|—— 系统 ID

|—— 显示名称

|—— 公开隐私设置

└—— 公开贡献记录引用

1. 可选的 DID 集成 \[架构建议，待定\]

- 参与者可选择关联去中心化身份 (DID)
- DID 不强制，不关联不影响参与资格

4.2 数据最小化原则

实施措施：

原则 实现方式

只收集必要数据 申报表单限制字段，禁止收集与贡献无关的信息

限制保留期限 贡献记录永久保留；临时评估数据 90 天后删除 \[待定\]

限制访问范围 角色基础访问控制 (RBAC) + 属性基础访问控制 (ABAC)

默认隐私 贡献记录默认不公开，用户主动选择公开

数据保留细化规则：

数据类型 建议保留规则

贡献记录摘要 可长期保留

原始证据材料 按期限保留 1-3 年，或争议结束后删除/归档

受益者个人信息 最小化保存，优先不保存

评估过程数据 试点期可保留，正式期按审计周期删除或脱敏

公开展示数据 必须脱敏

退出者记录 不再用于当前权重，仅保留脱敏历史档案

4.3 贡献记录匿名化处理

分层匿名策略：

场景 匿名化程度 说明

公开贡献墙 高度匿名 仅显示类型、领域、模糊时间

社区内展示 中度匿名 显示显示名称、贡献类型

个人主页 低度匿名 用户自行决定展示内容

仲裁审查 可追溯 经授权后可查询完整信息

技术实现： - 贡献墙 API 返回聚合数据，不返回可关联到个人的详细信息 -

个人主页内容由用户隐私设置控制 - 审计日志记录所有敏感数据访问

4.4 访问控制策略

多层访问控制：



└── 可选认证: DID/区块链签名 [架构建议, 待定]	
第二层: 角色权限 (RBAC)	
└── contributor: 贡献申报、查看自己的记录	
└── evaluator: 贡献申报 + 评估任务	
└── governor: 提案管理 + 投票	
└── admin: 系统配置 + 仲裁 + 用户管理	
└── ethics_committee: 伦理审查 + 数据访问	
第三层: 属性权限 (ABAC)	
└── 领域绑定: 评估者只能评估本领域贡献	
└── 利益相关排除: 自动排除利益冲突方	
└── 时间限制: 敏感操作需在规定时间内完成	
└── 额度限制: 防止资源滥用	
第四层: 审计追踪	
└── 所有敏感操作记录审计日志	

Beneficiary 保护: - 受益者信息存储使用单独的加密密钥 -

受益者信息不可被普通管理员访问 - 受益者有权随时撤回知情同意

4.5 审计日志设计

日志记录范围:

类别	记录内容	保留期限

身份认证	登录时间、IP、认证方式	1年

贡献申报	申报人、时间、内容摘要	永久
评估操作	评估人、评分、冲突声明	永久
治理投票	投票人、投票内容（匿名后存储）	永久
行政操作	管理员操作、原因	永久
数据访问	敏感数据查询、导出	2年
系统事件	异常、熔断、配置变更	2年

日志格式：

```
{
  "timestamp": "ISO8601",
  "eventtype": "contributionsubmitted",
  "actor_id": "uuid (系统 ID)",
  "actor_role": "contributor",
  "target_id": "uuid",
  "target_type": "contribution",
  "ip_address": "string (加密存储)",
  "user_agent": "string",
  "details": {
    "domain": "uuid",
    "type": "B",
    "flagsapplied": ["nondirect_benefit"]
  },
  "consent_verified": true
}
```

日志访问控制： - 普通管理员只能查看与自己操作相关的日志 -

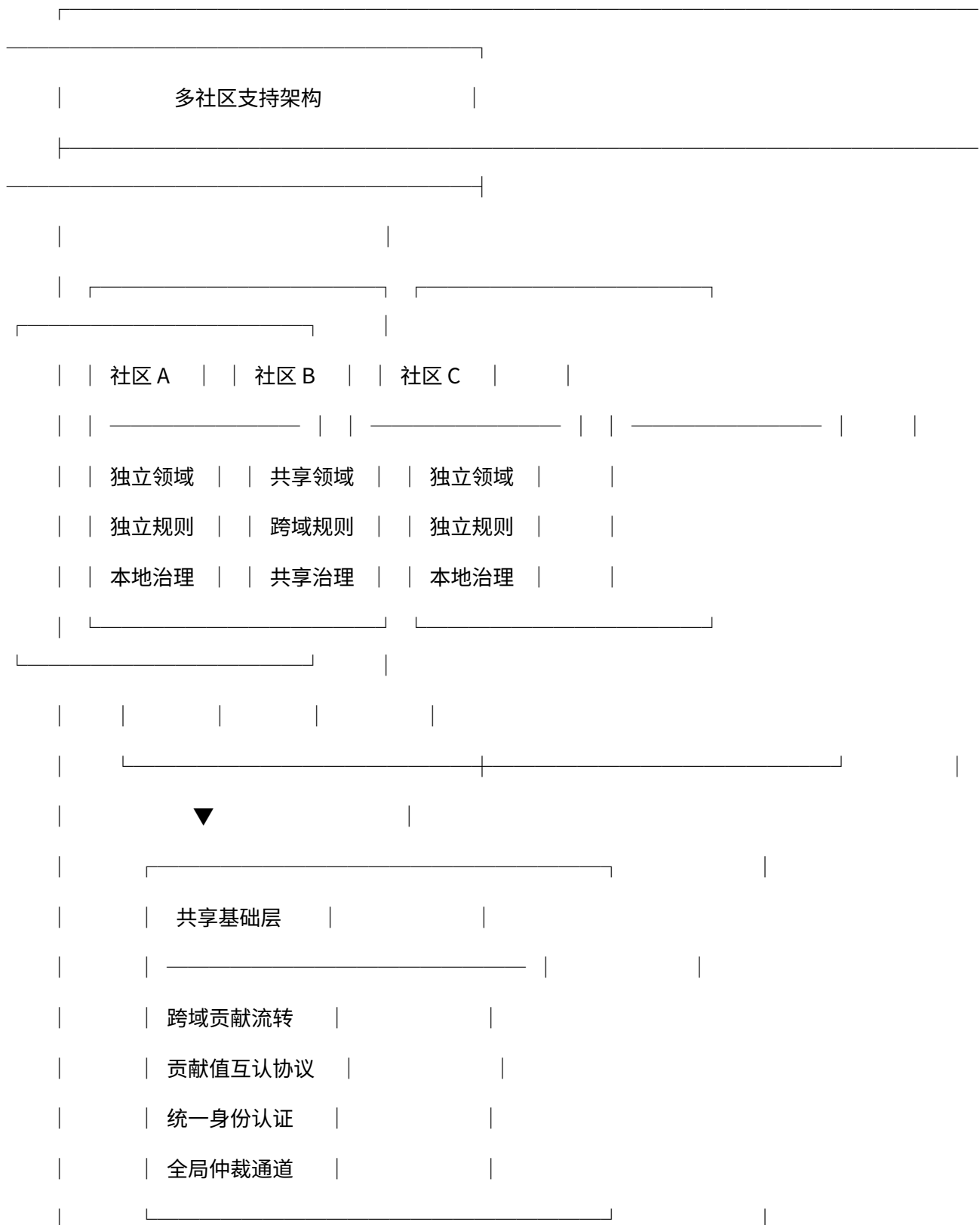
伦理委员会可查看全部日志 - 日志不可被删除，只能追加 -

日志定期归档至冷存储

第五部分：可扩展性设计

5.1 如何支持多社区

架构设计：



实现要点：

1. 社区隔离

- 每个社区有独立的数据库 schema 或数据库实例
- 社区内领域定义独立（但可选择采用标准领域集）
- 社区内治理规则可定制（受底线约束）

2. 跨社区协作

- 共享领域：多个社区共同维护的领域
- 跨域贡献：贡献者可声明贡献惠及其他社区
- 贡献流转：通过互认协议实现（参见附录 H）

3. 数据隔离策略

- -- 方案 A：独立 Schema（推荐小型部署）

community_a.contributions

community_b.contributions

- 方案 B：共享 Schema + 社区 ID 字段（推荐中型部署）

contributions WHERE community_id = 'A'

- 方案 C：独立数据库实例（大型部署）

communityadb.contributions

communitybdb.contributions

5.2 如何支持跨社区贡献流转

流转场景：

场景	说明	实现方式
----	----	------

跨域贡献	在 A 社区贡献，影响 B 社区	社区间互认协议
------	------------------	---------

桥接贡献	连接不同社区的协作	桥接标记 + 中心度上限
------	-----------	--------------

技术实现：

1. 贡献导出

```
• {  
  "export_id": "uuid",  
  "originalcontributionid": "uuid",  
  "exportingcommunity": "communityA",  
  "receivingcommunity": "communityB",  
  "contribution_hash": "sha256",  
  "signature": "communityAsigning_key",  
  "exported_at": "ISO8601",  
  "expires_at": "ISO8601"  
}
```

1. 贡献导入验证

- 验证来源社区签名
- 检查是否满足最低互认标准
- 检查 hash 一致性
- 创建本地关联记录

2. 互认协议（参见白皮书附录 H）

- 双方均承认全球底线
- 双方均无重大违规记录
- 记录格式兼容
- 争议处理机制一致

5.3 如何支持未来功能扩展

扩展设计原则：

原则 实现方式

领域可扩展 领域支持层级结构，可动态新增

贡献类型可扩展 类型编码使用命名空间（如 I.5 代表科学发现）

评估维度可扩展 评估维度配置化，新增维度不影响历史数据

标记类型可扩展 贡献标记使用 JSONB，允许新增标记

规则可热更新 业务规则使用规则引擎，支持动态更新

扩展点设计：

// 扩展点示例：贡献类型注册表

```
interface ContributionTypeRegistry {  
  register(type: ContributionType): void;  
  get(typeCode: string): ContributionType;  
  listByCategory(category: string): ContributionType[];  
}
```

// 扩展点示例：评估维度引擎

```
interface AssessmentDimensionEngine {  
  registerDimension(dimension: AssessmentDimension): void;  
  calculateScore(contributionId: string, dimensions: string[]): Promise<Score>;  
}
```

// 扩展点示例：衰减函数注册表

```
interface DecayFunctionRegistry {  
  register(domainId: string, decayFn: DecayFunction): void;  
  calculateBalance(participantId: string, domainId?: string): Promise<Decimal>;  
}
```

5.4 离线支持方案（概要）

目标：支持弱网络环境下的基本功能

分层设计：

功能层 离线支持 说明

贡献申报 支持 本地草稿 + 队列同步

评估任务 部分支持 可查看已下载任务，评分暂存

贡献查询 部分支持 查看本地缓存，定期同步

权重计算 不支持 必须在线获取最新值

治理投票 不支持 必须在线实时投票

资源申请 不支持 必须在线提交

技术方案 \[待定，详细设计在附录 H 基础上细化\]:

1. 本地存储

- IndexedDB 存储贡献草稿和缓存数据
- 加密存储敏感信息

2. 离线队列

- Service Worker 拦截请求
- 离线时写入队列
- 上线后自动同步

3. 冲突处理

- 时间戳+哈希检测冲突
- 冲突时提示用户手动解决
- 关键操作使用最终一致性

4. 同步策略

- 定期全量同步（每日一次）
- 实时增量同步（网络可用时）
- 手动触发同步（用户主动）

5.x 贡献确认通知文案原则

贡献确认通知应突出"贡献已被记录""感谢你的承担""该记录将进入你的贡献档案"，避免使用过度游戏化、排行榜化、刺激性表达。

推荐表达： \>

"你的贡献已被记录。感谢你为社区留下这一份照护/守护/协作的痕迹。"

不推荐表达：\> "恭喜你获得 XX 分！当前排名上升 XX 名！"

系统可以展示贡献权重数值，但不应诱导用户以分数、排名、等级作为主要参与动机。

5.x 数据保留原则补强

数据保留策略必须服从"记录贡献，不占有人"的原则。系统长期保留的，应主要是脱敏后的贡献摘要、统计结果、审计必要记录与历史档案，而不是持续保存受益者可识别信息、原始影像、个人隐私材料。

退出者记录处理原则：1. 不再用于当前 CV 权重、资源排序或治理权重；2.

可在脱敏后用于试点复盘、统计分析、争议审计；3.

如需重新关联个人身份，应取得本人授权或满足明确审计条件；4.

原始证据材料应按期限删除或归档，不得无限期暴露。

5.x 产品反模式清单

凡是使贡献记录转向分数竞争、身份攀比、道德打卡或受益者展示化的产品设计，均视为违反贡献值体系初衷。产品反模式清单如下：

以下产品设计方向应明确避免：

1. 排行榜中心化：以榜单、等级、排名作为首页核心展示
2. 打卡式利他：鼓励用户为了连续记录而制造贡献行为
3. 受益者展示化：把老人、儿童、弱势群体影像作为贡献证明或传播素材
4. 强刺激通知：使用"恭喜升级""超越 XX 人"等游戏化话术
5. 默认公开贡献细节：未经用户主动选择，公开具体贡献对象、地点、时间和受益者信息
6. 组织吞并个人贡献：默认把成员贡献计入组织总贡献，而不保留个人实际承担记录
7. 高 CV 特权化入口：将高 CV 用户在界面中包装成等级身份或准特权阶层
8. 弱申诉入口：申诉、纠错、退出入口隐藏过深

贡献值体系的产品设计必须避免将"贡献记录"异化为"分数竞争"。

技术字段与用户展示规范

核心原则：数据库记录权重，界面呈现意义。

技术字段可以在系统内部存在，但用户界面不得直接暴露技术语言。以下是强制映射规范：

技术字段	数据库/后端	用户界面建议展示
cv_balance	贡献值余额	当前有效贡献记录 / 当前有效贡献汇总 (绝

		对不提"余额")
thoughtcontributionscore	思想贡献分	思想贡献记录
behaviorcontributionscore	行为贡献分	行为贡献记录
systemcontributionscore	系统贡献分	系统贡献记录
collaborationcontributionscore	协作贡献分	协作贡献记录
contribution_score	综合贡献分	贡献记录量级
governance_weight	治理权重	治理参考权重 / 决策参考权重
reputation_score	评估者信誉分	评估者信任度 (参考)

禁止在用户界面出现的词汇：

- **✗** 分数 / 排名 / 等级 / 积分 / 余额 / 兑换 / 流通 / 高分用户 / VIP / 等级制度
- **✓** 可以出现的表达：记录、轨迹、档案、贡献汇总、参考权重、优先申请排序

系统设计必须区分技术字段与用户感知。

数据库可以记录权重、分值、衰减和排序，但产品界面不得制造分数竞争、身份攀比、福利兑换或社会信用感。用户应感到自己正在留下贡献痕迹，而不是正在被评分。

第六部分：技术选型建议

说明：以下内容均为架构建议，不是强制要求。技术选型应根据团队熟悉度、部署环境和业务规模综合决策。

6.1 数据库选型考量

类型	推荐场景	备选方案

关系型数据库		
PostgreSQL	主数据存储，首选	适用于贡献记录、用户、评估等核心数据
MySQL	轻量级部署	适用于小规模试点
时序数据库		
TimescaleDB	CV 衰减计算、趋势分析	可选引入，PostgreSQL 扩展也可满足
文档数据库		
MongoDB	半结构化数据、日志存储	适用于非核心数据
向量数据库		
pgvector	贡献相似性匹配	\[架构建议，待定\]
缓存		

Redis 会话、缓存、实时队列 必需

搜索

Elasticsearch 全文搜索、日志分析 可选, PostgreSQL 全文搜索可满足初级需求

推荐方案 (100 人试点) :

单一 PostgreSQL 实例

├── 主数据表 (contributions, participants 等)

├── JSONB 字段 (evidence, flags 等)

├── Redis (缓存 + Session)

└── 本地文件系统 (证据文件存储)

扩展方案 (1000 人+) :

PostgreSQL 主从复制

├── 只读副本 (查询分离)

├── Redis 集群

├── MinIO/S3 (文件存储)

└── Elasticsearch (搜索 + 日志)

6.2 身份认证方案

方案	适用场景	说明
----	------	----

自建认证

邮箱/手机 + 密码	基础场景	密码加盐哈希存储, 支持 MFA
------------	------	------------------

社交登录

OAuth 2.0 + 主流平台	降低注册门槛	需处理账户关联问题
------------------	--------	-----------

去中心化身份

DID/W3C Verifiable Credentials	长期愿景	\[架构建议, 远期目标\]
--------------------------------	------	----------------

推荐方案 (试点阶段) : - 邮箱 + 强密码 (bcrypt/scrypt) -

选配 TOTP 二因素认证 - Session token 使用 JWT (短期) + Redis 黑名单

6.3 区块链/DID 的适用场景分析

场景	适用度	说明
不可篡改日志	中	审计日志、贡献记录哈希
去中心化身份	中	DID 控制个人身份和贡献记录
跨社区互认	中	贡献流转的信任锚点
实时交易	低	CVS 不是交易系统，区块链不适合
高频操作	低	链上 Gas 成本高

建议策略：

阶段	建议	说明
试点阶段（100 人）	不引入区块链	使用传统数据库 + 审计日志
扩展阶段（1000 人+）	可选引入	仅用于贡献记录哈希存档
成熟阶段	按需引入	根据社区需求决定

注意：区块链的引入应基于实际需求，而非技术追新。CV 不可交易、不可转让，区块链的"不可双花"特性不适用于本系统。

6.4 其他技术选型建议

组件	推荐技术	说明
后端框架	Node.js/TypeScript 或 Go	快速迭代 vs 性能平衡
前端框架	React/Vue + TypeScript	团队熟悉度优先
移动端	React Native 或 Flutter	跨平台优先
API 网关	Kong 或 Nginx	限流、认证、路由
容器化	Docker + Docker Compose	开发/测试环境必需
编排	Kubernetes	生产环境（1000 人+）
CI/CD	GitHub Actions / GitLab CI	自动化构建部署
监控	Prometheus + Grafana	基础监控必需

7.1 单元测试策略

测试覆盖重点：

模块	测试重点	覆盖率目标
----	------	-------

贡献记录	状态流转、验证规则	90%+
------	-----------	------

评估计算	评分汇总、权重公式	95%+
------	-----------	------

权重计算	领域权重、时间衰减	90%+
------	-----------	------

反作弊	异常模式检测	85%+
-----	--------	------

隐私保护	数据脱敏、访问控制	90%+
------	-----------	------

测试示例：

```
describe('Contribution 状态流转', () => {
  test('申报后进入 pending_review 状态', () => {
    const contribution = Contribution.create({...});
    expect(contribution.status).toBe('pending_review');
  });
  test('评估完成后进入 confirmed 状态', () => {
    const contribution = Contribution.create({...});
    contribution.submitForEvaluation();
    contribution.completeAssessment({scores: {...}, final_cv: 10});
    expect(contribution.status).toBe('confirmed');
  });
  test('存在利益冲突的评估者被排除', () => {
    const evaluator = createParticipant({role: 'contributor'});
    const contribution = createContribution({contributor_id: evaluator.id});
    const matchedEvaluators = AssessmentMatcher.findEvaluators(contribution);
```

```

expect(matchedEvaluators).not.toContainEqual(evaluator);
});
});
describe('治理权重计算', () => {
test('领域排名前 20%获得额外权重', () => {
const weights = calculateGovernanceWeights(domainId);
const top20Percent = weights
.sort((a, b) => b.cvbalance - a.cvbalance)
.slice(0, Math.ceil(weights.length * 0.2));
top20Percent.forEach(w => {
expect(w.extra_vote).toBeGreaterThan(0);
});
});
test('高 CV 群体总票权不超过 30%', () => {
const weights = calculateGovernanceWeights(domainId);
const highCvVotes = weights
.filter(w => w.extra_vote > 0)
.reduce((sum, w) => sum + w.extra_vote, 0);
const totalVotes = weights.length + highCvVotes;
expect(highCvVotes / totalVotes).toBeLessThanOrEqual(0.3);
});
});

```

7.2 集成测试要点

测试场景：

场景	测试内容	验证点
----	------	-----

贡献申报流程 提交 → 预审 → 评估 → 确认	全流程状态正确、数据一致
--------------------------	--------------

评估者匹配 候选筛选 → 随机选择 → 任务分发 排除正确、分布均匀

CV更新 评估完成 → CV计算 → 余额更新 计算正确、衰减正确

治理投票 提案创建 → 投票 → 结果计算 权重正确、结果正确

资源分配 申请提交 → 排序 → 分配 优先级正确、限制生效

隐私保护 数据导出 → 脱敏处理 敏感信息不泄露

争议处理 提起争议 → 仲裁 → 结果 流程正确、权限正确

集成测试环境：

docker-compose.yml (测试环境)

```
services:
  postgres_test:
    image: postgres:15
    environment:
      POSTGRESDB: cvstest
  redis_test:
    image: redis:7
  app:
    build: .
    depends_on:
      - postgres_test
      - redis_test
    environment:
      NODE_ENV: test
      DATABASEURL: postgres://.../cvstest
```

7.3 红蓝对抗测试要点

对接白皮书附录 G：

测试类型	测试方法	通过标准
------	------	------

伪造记录攻击	模拟高频低质贡献申报	系统能识别并处置
批量虚假申报	短时间内提交 100+条相似申报	触发分拆申报检测
证据造假	上传重复/伪造证据	触发证据验证失败
串通评估	模拟评估者抱团	系统能识别并暂停资格
评分一致性过高	多个账户给同一贡献相似评分	触发串通检测
关系网络抱团	关联账户频繁互相评估	触发网络关系分析
精英捕获	模拟高 CV 群体垄断决策	系统触发防捕获机制
票权集中	模拟排名前 20%群体投票一致	触发 30%上限
长期连任	模拟关键角色长期连任	触发轮换检查
道德绑架	模拟用 CV 施压不参与者	系统能识别并警告
社交压力	\[测试自动化困难，建议人工测试\] 记录压力表现、评估应对	
受益者尊严	模拟强制展示受益者信息	系统能识别并拒绝
强制匿名	上传含可识别信息的受益者证据	触发隐私检查
强制感谢	申报中包含感谢要求	触发规则检查

对抗测试框架：

```
describe('红蓝对抗 - 伪造记录攻击', () => {
  let attacker: Participant;
  let targetContribution: Contribution;
  beforeEach(async () => {
    attacker = await createParticipant({status: 'active'});
  });
  test('批量相似申报被检测', async () => {
    // 模拟攻击：短时间内提交 20 条相似申报
    const submissions = [];
    for (let i = 0; i < 20; i++) {
      submissions.push(submitContribution(attacker, {
```

```

type: 'B',
subtype: 'B2', // 照护陪伴
description: 帮助邻居照护老人第${i+1}次,
));
}

await Promise.all(submissions);

// 验证: 应触发分拆申报检测

const integrityFlags = await getIntegrityFlags(attacker.id);
expect(integrityFlags.splitdeclarationdetected).toBe(true);
});
});

```

7.4 压力测试场景

场景设计:

场景	负载	持续时间	关键指标
日常负载	100 并发申报 + 50 并发评估	1 小时	响应时间 < 2s
峰值负载	500 并发申报 (活动促销)	10 分钟	系统不崩溃
评估高峰	1000 评估任务同时分发	5 分钟	任务分发完成
投票高峰	全社区同时投票	30 分钟	投票记录完整
资源申请压力	1000 人同时申请稀缺资源	10 分钟	公平分配

压力测试工具: - k6 (推荐, 开源) - Apache JMeter - Locust

监控指标:

关键性能指标:

- 响应时间: p50 < 500ms, p99 < 2s
- 吞吐量: > 100 RPS
- 错误率: < 0.1%
- 可用性: > 99.9%

资源指标:

- CPU: < 70%
- 内存: < 80%
- 数据库连接: < 80% 最大值
- Redis 内存: < 70%

第八部分：部署与运维

8.1 最小化部署需求

单社区试点部署（100 人）：

组件	规格	说明
----	----	----

应用服务器	2 核 CPU / 4GB RAM	可使用共享云服务
数据库	2 核 CPU / 8GB RAM, 50GB SSD	PostgreSQL
缓存	1 核 CPU / 2GB RAM	Redis（可与 DB 同机）
文件存储	10GB	本地存储或 MinIO

总资源需求：4 核 CPU / 14GB RAM, 60GB 存储

部署方式：Docker Compose（一键部署）

docker-compose.yml (最小化部署)

```
version: '3.8'

services:

app:

build: .

ports:

  > "3000:3000"

depends_on:

  > db

  > redis
```

```
environment:
  DATABASE_URL: postgres://user:pass@db:5432/cvs
  REDIS_URL: redis://redis:6379
db:
image: postgres:15
volumes:
  > pgdata:/var/lib/postgresql/data
environment:
  POSTGRES_DB: cvs
  POSTGRES_USER: user
  POSTGRES_PASSWORD: pass
redis:
image: redis:7
volumes:
  > redisdata:/data
nginx:
image: nginx
ports:
  > "80:80"
volumes:
  > ./nginx.conf:/etc/nginx/nginx.conf
volumes:
pgdata:
redisdata:
```

8.2 监控指标

核心监控指标:

类别	指标	告警阈值	说明
----	----	------	----

可用性 服务可用率 \< 99.5% 月度窗口

响应时间 API 平均响应时间 \> 1s 持续 5 分钟

响应时间 API p99 响应时间 \> 3s 持续 5 分钟

错误率 5xx 错误率 \> 1% 持续 5 分钟

吞吐量 QPS \< 预期 50% 持续 10 分钟

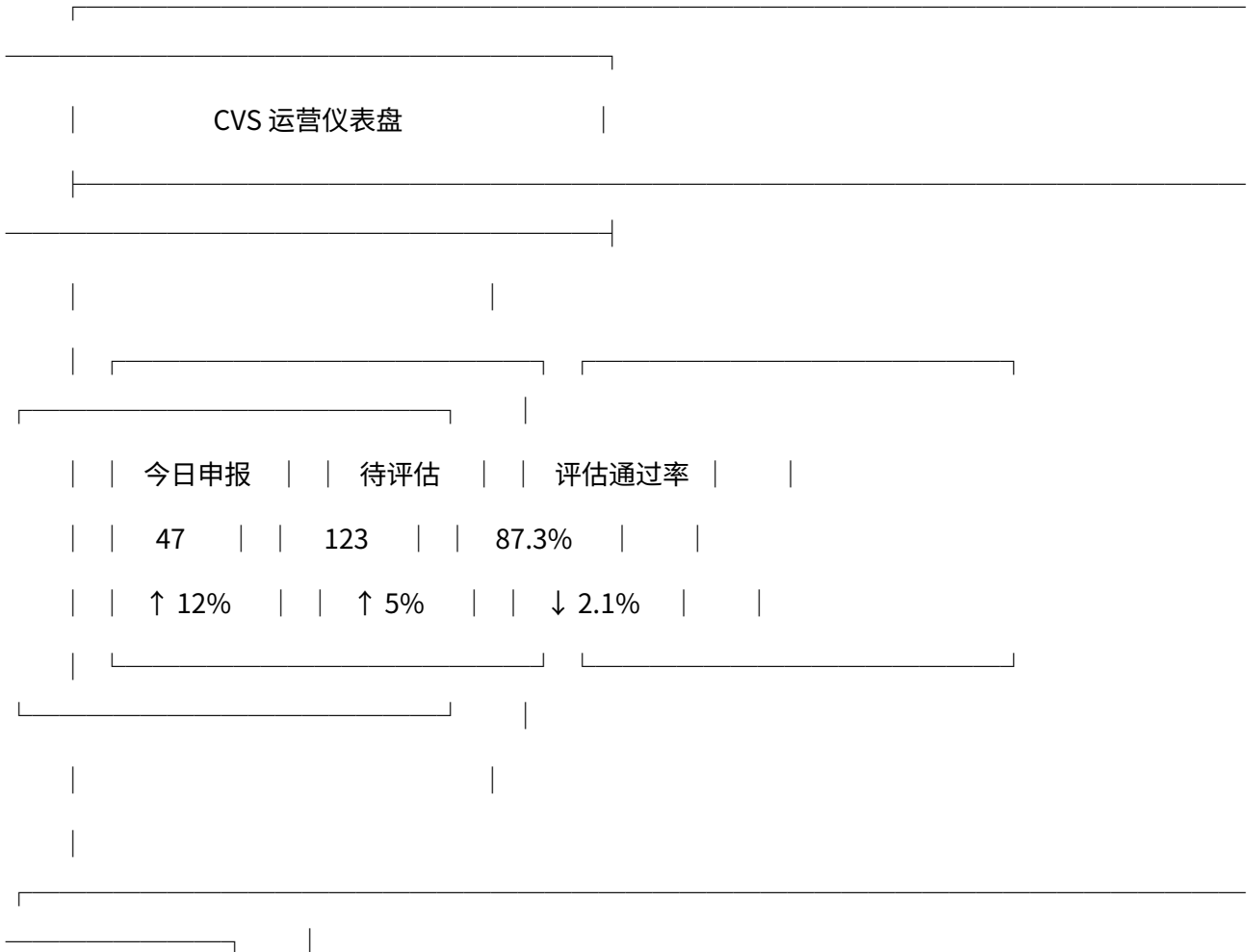
数据库 连接数 \> 80% 最大值 持续 2 分钟

数据库 查询延迟 \> 500ms p95, 持续 5 分钟

队列 待处理评估任务积压 \> 100 持续 15 分钟

安全 认证失败率 \> 10% 持续 5 分钟

仪表盘设计:



申报量趋势 (7天)



评估一致性系数	响应时间 p99
---------	----------

0.73	850ms
------	-------

目标: 0.70 ✓	目标: 1s ✓
------------	----------

反作弊拦截率	活跃参与者
--------	-------

0.8%	89
------	----

正常范围 ✓	/ 100 总数
--------	----------

8.3 灾难恢复 (简单说明)

RPO/RTO 目标:

环境	RPO (恢复点目标)	RTO (恢复时间目标)
----	-------------	--------------

试点阶段 (100 人) 1 小时 4 小时
扩展阶段 (1000 人) 15 分钟 1 小时

备份策略:

备份类型 频率 保留时间 存储位置

全量备份 每日 30 天 本地 + 异地

增量备份 每小时 7 天 本地

事务日志 每 5 分钟 7 天 本地 + 异地

配置备份 每次变更 90 天 版本控制

恢复流程 (简要):

1. 数据库恢复

- # 停止服务

```
docker-compose stop app
```

恢复最新全量备份

```
pgrestore -h db -U user -d cvs latestbackup.sql
```

重放事务日志

```
pgrestore -h db -U user -d cvs --jobs=4 latestbackup.sql
```

```
walredo ...
```

启动服务

```
docker-compose start app
```

1. 数据验证

- 验证最近申报记录完整性
- 验证 CV 余额一致性
- 抽样验证历史记录

2. 通知机制

- 恢复期间展示维护公告

- 恢复后发送通知给受影响用户

多区域容灾 \[架构建议, 大型部署\]: - 主备双活部署 - 自动故障转移 -

数据实时同步

附录: 待定项清单

以下为本文档中标注为[待定]的决策项, 供后续迭代确定:

编号	待定项	依赖条件	建议决策时机
1	衰减参数 (速率、周期)	试点数据	100 人试点后
2	评估者数量 (3-5 名范围)	试点反馈	100 人试点后
3	争议仲裁时效 (30 日参考值)	运营数据	1000 人阶段
4	权重公式参数 (领域相关性等)	试点数据	100 人试点后
5	公共资源韧性池规模	资源规模	1000 人阶段
6	离线同步冲突策略	用户反馈	100 人试点后
7	评估者信誉权重	评估数据	1000 人阶段
8	区块链引入时机	社区需求	1000 人阶段后
9	评估一致性目标调整	试点数据	100 人试点后
10	日志保留期限调整	合规要求	按需确定

文档历史

版本 日期 说明

v1.0 2026-05 初始版本

v1.0.1 2026-05 修正版本: 字段名英文化、status 去惩戒化、数据保留规则细化、命令修正

v1.0.2 2026-05 全面修订: 新增"用户体验目标: 从记录工具到意义回响" (第一部分前); 评估人数统一调整为分层规则; 新增贡献确认通知文案原则、数据保留原则补强、产品反模式清单; 参考白皮书版本更新为 v1.7.2

v1.0.3 2026-05 对齐《贡献值体系白皮书 v1.7.6》；新增"技术字段—用户展示映射表"；前置产品反模式清单；强化界面不得制造评分感、监控感、等级感；所有用户侧文案避免 score/balance/rank 等货币化语感

本文档为《贡献值体系白皮书 v1.7.6 正式发布版》的工程配套文档，如有问题请联系技术团队。

续读指引：本文之后应如何进入体系

向上续读：如果需要理解贡献值体系为什么不是单纯制度设计，而是文明跃迁的制度承接，应继续阅读《涌义宇宙论》v6.1 公众版、《人心：AGI 时代文明免疫系统》与《意义动力学》。

向下续读：如果需要进入操作、试点和工程实现，应继续阅读《贡献值体系试点执行手册》与《贡献值体系工程级设计方案》。

横向续读：如果需要理解贡献值如何连接意义经济、圆梦园、资源池、星火项目与公共治理，应进入网站下载中心的 CL-CVS 与 CL-ME 相关文档。

反向对照：如果需要理解贡献值体系应避免什么，应对照阅读《反封闭原则》《AGI-COS》与相关风险治理文稿，防止贡献记录异化为分数竞争、身份等级或资本化工具。

相关下载与网站回流入口

推荐续读 / 相关入口	网站路径
涌义宇宙论 v6.1 公众版	/paper/cosmology-v61-public.pdf
人心：AGI 时代文明免疫系统	/paper/human-heart-cn-v1-1-website-public.pdf
意义动力学 v1.3 主文	/paper/meaning-dynamics-cn-v1-3.pdf
意义动力学验证卷 v2.0	/paper/meaning-dynamics-validation-cn-v2-0.pdf
文明跃迁五卷主链总览	/paper/civilization-leap-five-volume-mainline-overview-cn-v1-0.pdf
AI 时代文明跃迁白皮书	/paper/ai-era-civilization-leap-whitepaper-cn-v1-0.pdf
星火项目公开说明	/spark.html

官网：civilleap.com | 当前入口：www.civitas.top | 文库：www.civitas.top/library.html | 下载中心：www.civitas.top/downloads.html | 阅读地图：www.civitas.top/reading.html | 联系邮箱：zijunfu@civitas.top

版本记录

本网站公开版基于源文件《贡献值体系_工程级设计方案_v1.0.3.docx》生成，补齐统一封面、文档信息/版权页、阅读前导、续读指引、相关下载与网站回流入口。正文内容仅做排版与结构承接处理，不改写核心判断。