

repl-sentinel 产品设计白皮书

技术决策者 / 架构师视角的设计理念与架构说明。提炼自 docs/prd/ 文档组 (01 - 10) 、 docs/architecture.md、 docs/theme.md。 产品版本口径：VERSION = 0.721.3。

1. 产品定位与价值主张

一句话定位：轻量、零侵入的 MySQL 主从复制监控——只读采集 1 - 8 套主从集群 (type1/2/3) 全部节点状态，异常/恢复发邮件告警，Vue3 单页应用可视化，前后端分离双容器镜像交付。

核心价值：

- 零侵入：对被监控库只执行只读语句，不做任何写库、改配、DDL/DML，监控账号由用户在目标库侧手动创建，生产库零污染。
- 容器化、开箱即用：前后端分离双镜像，配置页面完成全部初始化，无需运维手工改配置文件。
- 闭环可观测：拓扑图 + 节点状态卡 + KPI 十项实时总览，异常/恢复邮件闭环，去抖去重避免告警风暴。
- 安全与可控：单管理员账号 + 强口令哈希、HttpOnly 会话、访问安全分层；颜色/动效/偏好本地持久化。

2. 设计原则

原则	含义
零侵入	仅 5 类 SHOW + 1 条 innodb_trx 只读 SELECT，不读 SQL 文本，严禁写库
容器化交付	双镜像 + 启动脚本，配置全在 Web 页面，无配置文件运维负担
前后端分离	双独立进程/镜像，禁 go:embed 内嵌，后端只出 JSON API
安全优先	强制登录、强口令哈希、会话 HttpOnly、访问安全分层、密码不回显
范围收敛	固定三架构模板、禁增删节点；依赖白名单，运行时零外部依赖
可审计可维护	日志走 stdout、配置集中、原子写入、scope 部分保存

3. 架构设计

3.1 技术选型

组件	选型	理由
语言	Go 1.22+	单二进制、跨平台、标准库完善
MySQL 驱动	github.com/go-sql-driver/mysql	官方推荐；DSN 自动补 timeout=3s&readTimeout=5s&writeTimeout=5s
SSH 隧道	golang.org/x/crypto/ssh	仅容器无端口映射场景；经 RegisterDialContext 注册 ssh-<id> 网络名，checker 零改动
邮件	net/smtp (标准库)	465 隐式 TLS / 587 STARTTLS
口令哈希	crypto/hmac+crypto/sha256 自实现 PBKDF2	不引第三方库

组件	选型	理由
Web 后端	net/http (标准库路由)	无框架；只输出 JSON API，不内嵌前端
Web 前端	Vue 3 + Vite (独立进程/镜像)	现代组件化；纯静态产物，运行时零外部依赖
前端动效	gsap (唯一第三方运行时依赖)	链式编排；prefers-reduced-motion 退化为静态终态
前端图标	Iconoir 内置 SVG (MIT)	随工程构建，运行时不访问 CDN
前端控件	reka-ui (无头件) + 自绘 shadcn-vue 模式	无成品 UI 库
容器	后端 alpine:3.20；前端 nginx:1.27-alpine	镜像小

明确不引入：Gin/Echo/GORM/Viper/Logrus；前端图表库、vue-router、任何成品 UI 库、CDN 资源、图标字体。

3.2 模块划分与依赖方向

```
graph TD
  cmd[cmd/repl-sentinel 入口] --> app[internal/app 引擎]
  app --> config[internal/config]
  app --> checker[internal/checker]
  app --> gtid[internal/gtid]
  app --> state[internal/state]
  app --> alerter[internal/alerter]
  web[internal/web HTTP层] --> auth[internal/auth]
  web --> config
  web --> checker
  web --> state
  web --> guard[internal/guard 零内部依赖]
  checker --> gtid
  tunnel[internal/tunnel SSH] -.扩展白名单.-> checker
  frontend[frontend Vue3] -->|HTTP /api| web
```

依赖方向铁律：cmd → internal/*；app → config/checker/gtid/state/alerter；web → auth/config/checker/state/guard；guard 零内部依赖（不 import net/http）。禁止反向依赖与包间循环。

包 / 目录	职责
backend/cmd/repl-sentinel/	入口：组装 auth.Store、引擎、API 服务；信号优雅退出
backend/internal/app/	运行时引擎：配置装载/保存、热重载、调度、跨节点 enrich、告警串联、测试邮件/连接
backend/internal/config/	配置模型（Topology + Cluster + Node）与校验、config.json 原子读写、运行时变量
backend/internal/auth/	单管理员账号、PBKDF2、会话与 Cookie、auth.json/sessions.json
backend/internal/checker/	从库 ReplStatus + 源节点 SourceProbe/SourceStatus + NodeStatus 聚合
backend/internal/gtid/	GTID 集合解析、区间合并、差集计数
backend/internal/state/	按节点名独立的告警去重状态机 + 快照聚合
backend/internal/alerter/	邮件构造（按角色区分）+ SMTP 发送
backend/internal/guard/	访问安全判定与记录（零内部依赖，不 import net/http）

包 / 目录	职责
backend/internal/tunnel/	SSH 隧道采集通道（扩展白名单项）
backend/internal/web/	HTTP 层（纯 API，无 go:embed）：server/auth/api/dto/json/gzip
frontend/	Vue3+Vite 独立工程：src/api、src/views、src/components、src/composables、src/assets
deploy/	Dockerfile + frontend.Dockerfile/nginx.conf + docker-compose.yml + .env(.example) + package/
scripts/	开发机构建/启停（dev-start.sh、build.sh、build-web.sh、test.sh 等）
tests/	e2e 套件（68 断言）、contrast-check.py、计划与报告

4. 架构决策记录（ADR-01~11）

编号	决策	理由	状态
ADR-01	前后端分离双镜像，严禁 go:embed 内嵌前端	独立升级部署、无构建期耦合；后端只出 JSON API	已定
ADR-02	零侵入只读红线（5 类 SHOW + 1 条 innodb_trx 只读 SELECT）	不对生产库造成任何污染；监控账号用户手动创建	已定
ADR-03	单管理员账号 + PBKDF2-HMAC-SHA256（20 万次）	安全可审计、不引第三方库；页面强制登录	已定
ADR-04	内置 OpenSMTPD 本地 MTA + 双邮件模式（builtin/custom）	默认零配置发信；外置作高级兜底；不新增运行时变量	已定
ADR-05	告警去抖状态机（连续 2 次异常才告警，恢复不去抖）	过滤单次抖动；避免告警风暴	已定
ADR-06	拓扑匹配三档（match/mismatch/unknown）+ 节点级忽略开关	容器/端口映射部署下避免误报 red；unknown 不标红	已定
ADR-07	固定三架构模板、禁增删节点	范围收敛、实现简单；1vN 任意拓扑不在本版	已定
ADR-08	配置 scope 部分保存（cluster/collect/mail/security）	各子页独立保存校验、不互相覆盖草稿	已定
ADR-09	访问安全分层（nginx limit_req + watcher 渲染 deny + 后端 guarded 兜底）	执行点前置、后端兜底、永不中断服务	已定
ADR-10	前端依赖白名单（仅 gsap 为第三方运行时；无图表库/UI 库/vue-router）	运行时零外部依赖、可控可审计	已定
ADR-11	config.json 磁盘格式 v5（v4→v5 仅迁移全局 relay 两阈值默认值）	旧版全局 0 视作未启用改写为推荐值；v5 对旧二进制不可读（回退前需备份）	已定

5. 监控模型与判定口径

5.1 拓扑与采集集合

- Topology $\in \{type1, 2, 3\}$ ；模板固定节点数量/角色/上下游。v2 及更早配置缺 topology/version $\neq 3$ 直接报错，不迁移。
- 每从库一个 Checker；每「有下游的节点」一个 SourceProbe（主库总有下游；type3 的 slave-1 两者兼做）。

5.2 跨节点 enrich

- topology_match 三档：相等 $\rightarrow$ match；解析同 IP $\rightarrow$ match；解析不出 $\rightarrow$ unknown；无交集 $\rightarrow$ mismatch；端口不参与比较。
- gtid_behind：以直接上游 Executed_Gtid_Set 为基准、只统计上游 UUID。
- gtid_holes：节点内自派生，区间缺号事务数。
- upstream_degraded；transfer_pending_bytes（同文件相减/跨文件按 SHOW BINARY LOGS 累加，滚出保留列表时 estimated）；binlog_rate_bytes：源侧推进速率。
- 位点规则：只给「位点后不退」的数字，否则标未知；relay 积压/占用阈值 v0.700.0 起默认启用（64MB / 1GB）。

5.3 异常判定（从库满足任一即异常）

IO $\neq$ Yes / SQL $\neq$ Yes / 延迟 > 阈值 / 最后错误非空 / relay 积压超阈 / relay 占用超阈 / GTID 空洞 / 其他复制通道异常 / 长事务超阈 / 双一偏离 / 半同步降级 / read_only=OFF / relay_log_purge=OFF / 实例 uuid 冲突；源节点：连接失败或 SHOW MASTER STATUS 0 行。

5.4 状态机与去抖

- internal/state 按节点名独立，debounceThreshold=2；三态 ok/watching/alerting；恢复不去抖；发送失败保持 pending 下周重试；Reset() 配置变更后清空。
- 判定边界：从库结果集 0 行 $\rightarrow$ 异常；Seconds_Behind_* NULL $\rightarrow$ LagValid=false；多源复制默认通道作主判定行；源节点连接失败或 SHOW MASTER STATUS 0 行 $\rightarrow$ 异常；辅助采集权限不足只降级留空。

5.5 检查项清单口径

- status $\in$ ok | warn | bad | unknown：warn=不进告警的提示项；bad=进健康判定的异常项；unknown=首轮无对比/采集降级/阈值未启用。
- alert：该项失败是否可能触发告警邮件。
- 项数：源节点组 11 项 / 从库组 24 项（含 GTID 空洞、其他复制通道、每秒事务等）；仅判定的展示翻译，不参与健康判定、不进 /api/history。

6. 安全设计

- 鉴权：单管理员账号；会话 = token + HttpOnly Cookie（12h 滑动；记住登录落盘 sessions.json 0600）；未登录 401 + code。除 auth 端点与 /healthz、/readyz 外全部需登录。
- 口令哈希：PBKDF2-HMAC-SHA256 自实现（20 万次迭代 + 随机 salt），auth.json 仅存 salt+hash（0600），无明文。
- 密码不回显：节点密码 / smtp_pass 返回空 + password_set/smtp_pass_set 标志；日志 DSN 用 SafeDSN() 打码。
- 传输层：外置 SMTP 支持 465 隐式 TLS / 587 STARTTLS；内置 MTA 127.0.0.1:25 免认证。
- 访问安全分层：仅未认证请求计数；高频未认证 IP 自动封禁（默认全关）；nginx limit_req + watcher 渲染 deny + 后端 guard 兜底；会话优先解封、白名单不封禁、DryRun、到期自动解封；清单落盘 security.json（0600），配置 scope=security。

- 热重载：Save() 先落盘再 apply (重建 checker/probe/alerter、store.Reset()、发 reload 信号)；旧连接池全部 Close()；不动 auth 会话。

7. 界面与体验设计

7.1 总体形态

- Vue 3 单页应用，独立进程/镜像（无 go:embed）；开发 Vite、部署 Nginx 反代 /api、/healthz、/readyz；运行时无 CDN/外链。
- 3 条 hash 路由（严禁 vue-router，自实现 composables/useRoute.js）：#/dashboard（监控总览）、#/settings/cluster（集群配置）、#/settings/mail（告警邮件）。
- 单页控制台 + 受控门禁：监控总览为受控入口，配置保存成功才解锁；未配置时登录默认落集群配置页；配置失效立即退回设置页。

7.2 三主题（颜色权威 = docs/theme.md）

- 经 html[data-theme] 切换：暖白色系（:root 默认，跟随系统：亮→暖白、暗→黑金）、黑色系（dark，夜间值守）、深蓝色系（blue，企业指挥中心）。
- 主题选择器三枚单色圆点（暖白 #FAF9F6/近黑 #09090B/深蓝 #0284C7），点选即换；偏好 localStorage['rs-theme']，默认跟随系统。
- 统一状态色跨主题语义一致：ok #22C55E / warn #F59E0B / bad #EF4444 / info #3B82F6 / unknown #94A3B8；浅色作文字时压深至 AA，深色系 bad 提亮 #F87171。
- 三种架构各占一色相族（--arch-1/2/3）：浅色蓝/青/紫（224°/175°/262°），独立于品牌辅色。
- 字重仅 300/400/600/700（禁用 500）；字号/间距/色值全部来自 main.css token，规则区不出现裸值；所有文字对比度经 tests/contrast-check.py 核算（三主题 119 对 ≥ 4.5:1）。

7.3 动效铁律

- 禁止以「不可见」作为起始态：gsap 靠 rAF 推进，标签页节流/无头截图时补间不推进，opacity:0 起会让内容永久不可见、数字恒显 0。淡入用 CSS 动画（默认态即终态）；数值直接渲染真实值，不做「从 0 滚动」；状态类动效以可见态为基准。
- 所有动效尊重 prefers-reduced-motion，并受用户级动效开关统一压停（html.motion-off，首访跟随系统，偏好 localStorage）。
- 背景动画密度滑杆：稀疏/标准/密集（默认标准），调背景动画出现频率；氛围背景为简单粒子星尘场，配色随主题翻色。

8. 非功能设计（阈值）

性能 NF-P

编号	指标	阈值
NF-P-01	单节点检查耗时	< 1s（逐节点串行，单节点 15s 兜底超时）
NF-P-02	常驻内存	< 50MB（后端进程含连接池）
NF-P-03	CPU 占用	空闲时接近 0
NF-P-04	二进制大小	< 20MB（strip 后，OpenSMTPD 在镜像内不计入）
NF-P-05	采集节奏	默认 30s 间隔

可靠性 NF-R

编号	要求
NF-R-01	断线自动重连（连接池自带）
NF-R-02	检查失败不影响进程（单节点异常不阻断其余）
NF-R-03	邮件失败不中断主流程（下周期重试至送达）
NF-R-04	未配置不致命（/healthz 仍 200，不 Fatal）

兼容性 NF-C

- MySQL 8.0.22+（兼容 5.7 / 8.0.22- 回退 SHOW SLAVE STATUS；GTID 需 5.6+）。
- 部署：容器化（Docker / Kubernetes），前后端双镜像。
- 操作系统：Linux（Alpine 基础镜像）；Go 1.22+。
- 浏览器：现代 evergreen，无 SSR、无 CDN 依赖。

安全 NF-S

- 页面强制登录；口令 PBKDF2 20 万次 + 随机 salt（auth.json 0600）；会话 HttpOnly 12h/记住 3 天；密码不回显；零侵入；访问安全分层；外置 SMTP TLS。

可维护性 NF-M

- 前后端独立（无 go:embed）；日志走 stdout；配置集中（运行时变量仅 LISTEN_ADDR/DATA_DIR/TZ）；依赖白名单（后端=Go 标准库 + mysql + ssh 仅隧道；前端=vue/vite/gsap/reka-ui 等，无图表库/UI 成品库 /vue-router）。

9. 已知限制与路线图

已知限制 L-*

- L-01 仅三种固定架构，不支持增删节点（1vN 任意拓扑不在本版）。
- L-02 状态不持久化：重启后告警去重状态与登录会话重置，可能重复告警一次并需重新登录。
- L-03 会话仅内存：多副本部署各副本会话独立，需会话粘滞（单实例无此问题）。
- L-04 延迟指标局限：Seconds_Behind_Source 在 SQL 线程空闲时可能为 NULL，并行复制下不够精确。
- L-05 发送失败重试无上限。
- L-06 去抖代价：单次异常不告警；故障持续不足两个检查周期不产生告警。
- L-07 配置变更即重置去重状态。
- L-08 单管理员账号：无多用户与角色分级。
- L-09 密码强度不校验（由使用者掌握）。
- L-10 前端独立进程，需同时运行前后端；只启后端则 GET / 返回 404（预期）。

风险 R-*（摘要）

- R-01 v5 对 v0.700.0 及更早二进制不可读（回退前先备份；改 "version":4 即可被旧版读）。
- R-02 邮件重试无上限（日志增长）。
- R-03 配置变更重置去重状态（设计取舍）。
- R-04 单管理员账号无权限分级。
- R-05 状态不持久化（重启可能重复告警一次）。

- R-06 会话仅内存（多副本需粘滞）。

后续路线图

优先级	优化项	说明
P2	状态持久化	告警去重状态写入文件（缓解 R-05/L-02）
P2	更多架构模板	支持自定义节点数量与拓扑（当前刻意固定三种，L-01）
P3	更多通知渠道	钉钉、飞书、企业微信
P3	主库 binlog 堆积告警	按 binlog 目录大小/文件数阈值告警

本白皮书为设计理念与架构说明，不与 PRD 冲突时以 docs/prd/ 文档组 + docs/architecture.md 现行章节为准；待决策事项（Q-* 共 14 项，均待定）拍板后回写对应上游文档。

以上是 EVE 的回答