SLA и индекс здоровья

SLA и индекс здоровья (ИЗ)

Краткий справочник: **чем отличаются** показатели, **где смотреть** в UI, **как считаются** и **какой код трогать**, если нужно переделать.

Два разных сигнала

	SLA (готовность)	**ИЗ (индекс здоровья)**
Смысл	Доля времени, когда объект **доступен** (reachability) за период	Мгновенный **score 0–100** по открытым проблемам правил шаблона здоровья
Где в списке Мониторинг	Чип **SLA** (у всех kind)	Чип **ИЗ** только у **узлов** (`POST /api/v1/health-index/bulk`)
Где на обзоре	Узел: блок SLA и полоска **Готовность скрыты** (решение не принято); лента **ИЗ** остаётся. Канал/доступность — SLA в шапке + лента.	Gauge ИЗ (узел) + контекст шаблона здоровья; история — строки `service_health_snapshots`
Источник	Ряды в ClickHouse (ICMP / agent / loss канала)	Открытые trigger-problems + effective health config
Можно ли восстановить «назад»	Да — из уже хранимых точек	Нет честно без снимков

Не путать: высокий ИЗ ≠ выполненный SLA за сутки, и наоборот.

Где в интерфейсе

1. **Список** `/services` — колонка **SLA / ИЗ**
   - узел: чип **ИЗ** из `POST /api/v1/health-index/bulk`
   - канал / доступность: один чип `SLA p%`
2. **Обзор** `/services/:id` — у **узла** блок SLA и полоска **Готовность скрыты**; лента **Индекс здоровья** остаётся; у канала/доступности — SLA в шапке + лента; ИЗ gauge справа от шапки
3. **Каталог** `/sla-sli/health-templates` — Шаблоны здоровья
4. **Над графиком** — лента **История статусов** (на узле сейчас только ИЗ; на канале/доступности — готовность). Маркеры смены шаблона/привязки/override — в блоке ИЗ

Подробнее для оператора: Сервисы и узлы, Шаблоны здоровья.

Как считается SLA

- **Узел:** ICMP (`icmpping`) и/или агент (`agent.ping`). Если нет ни ICMP, ни агента, но есть **ODBC ping** (`db.odbc.select[pgsql.ping...]` / mysql) — готовность считается по нему. **Готовность %** при ICMP+агент — OR. На ленте SLA: оба UP ⟶ ok; один UP ⟶ **warn**; оба DOWN ⟶ bad.
  Сервис: `NodeAvailabilitySlaService` + `segmentsFromNodeReachability`.
- **Канал (по loss + RTT):** пороги шаблона **warn / crit** ⟶ банды (худший из loss и RTT **avg/max**):
  - **ok / warn / bad** — quality; **готовность % = доля времени не в bad** (жёлтое не сжигает).
    Полоска и SLA берут `max(rtt.avg, rtt.max)` — один медленный пакет в пробе красит ленту, даже если AVG на KPI ещё «норма». Карточки KPI показывают **последний AVG** (и тон задержки учитывает max). График в режиме AVG — среднюю; MAX — пики.
    `ChannelAvailabilitySlaService` + `fromLossTimeWeighted` / `fromLossAndRttTimeWeighted`.
- **Доступность услуги:** готовность = успех проверки (`*.ok`, иначе loss < crit как UP). RTT **не** входит в %; на ленте может дать **warn** (медленно), **bad** только при провале проверки.
  `segmentsFromAvailabilityCheck` / `availabilityReadySamples`.
- API сводки: `GET /api/v1/services/{id}/sla/summary?hours=...` (+ `template_key`, `direction` для канала).

Тон блока SLA относительно цели (по умолчанию 99%): **ok** ≥ цели, **warn** чуть ниже, **bad** сильнее — это про **итог %**, отдельно от цвета сегментов на timeline.

Окна ППР и исключений SLA вычитаются из знаменателя (`maintenance_excluded_ms`, `exception_excluded_ms`); на ленте — отдельные тона. Исключение SLA **не** глушит Events.

Как считается ИЗ (узел)

Score строится **только из открытых проблем триггеров** и effective-конфига шаблона здоровья. Выражения триггеров и окна метрик в этом пути не вычисляются.

```
activeRulesInSeverity = count(enabled effective rules for severity)
ruleImpact = severityWeight / activeRulesInSeverity
penalty = min(100, sum(impact of firing enabled effective rules))
baseScore = 100 - penalty
finalScore = min(baseScore, lowest score ceiling among firing rules)
```

- Веса важности (Not classified, Information, Warning, Average, High, Disaster) в сумме 100%. Если у важности нет активных правил, её вес **перераспределяется** на остальные (в каталоге после seed такие оси хранятся как 0%). Если есть хотя бы одно активное правило, вес этой важности **не меньше 1%**.
- По умолчанию **info** и **not_classified** в шаблоне здоровья выключены. Каталог создаётся только если у шаблона мониторинга есть включённый триггер.
- Критичность `key_function` (потолок по умолчанию 60) и `service_outage` (0) не дают размыть отказ средним по многим правилам. В редакторе поле потолка видно, когда его можно править или когда значение уже задано.
- **Недоступность системы** — это отказ услуги, даже если в шаблоне важность Average/Warning/High. Иначе один AVERAGE среди ~8 правил даёт штраф ~3% (ИЗ

97%) или HIGH даёт потолок 60% на мёртвом объекте. Сюда входят агент («agent is not available», `agent.offline`), **SNMP** («No SNMP data collection», ICMP ping unavailable), **ODBC** («Service is down / unreachable / unavailable», «Failed to fetch info data», порт БД), JMX/HTTP без агента (нет данных JMX, status/stats page unavailable, connection unavailable). LLD-экземпляры (один backend, один интерфейс) не считаются отказом всей системы. Переопределение критичности **на узле** не трогается.
- LLD: одно правило на прототип; режимы `any_instance` / `affected_share` / `each_instance_capped`.
- Несколько шаблонов мониторинга на узле: правила **сливаются**; переопределения сервиса — разреженный overlay.
- Каталог: `/sla-sli/health-templates`. API: `GET /api/v1/services/{id}/health`, `GET /api/v1/services/{id}/health/breakdown` (разложение для Sankey/радара, только при открытии модалки; у правил есть `tags`; **без** записи снимка — persist только у worker после open/close события), `POST /api/v1/health-index/bulk`.
- На обзоре узла клик по блоку **индекса здоровья** и на списке **Мониторинг** клик по чипу **ИЗ** открывают модалку: справа от процента переключатель **Потоки** / **Радар**. Потоки — Sankey от узла **Здоровье** (цвет как у текущего ИЗ) ⟶ остаток по осям важности, штраф от открытых проблем, при необходимости срез потолка критичности. Спокойные правила одной оси сверх 12 сворачиваются в узел с числом («14 правил»); клик открывает полный список. Радар — вершины по тегам `scope:*` выбранных правил шаблона (локализованные имена), центр 0%, край 100% по категории. Bulk latest на списке по-прежнему без firing — разложение не префетчится.
- Снимки истории: партиционированная таблица Postgres `service_health_snapshots` — **только** смена тона (ok/warn/bad) или процента. Пересчёт после открытия/закрытия события на узле (система или вручную); смена шаблона здоровья / override тоже ставит узел в очередь. Нет периодического обхода флота.
- **BREAKING:** колонка `node_templates.health_index_keys` удалена (Flyway V141).

История статусов (timeline)

API

```
GET /api/v1/services/{id}/status-history?kind=sla|health|both&hours=24
```

Опционально: `template_key`, `direction` (канал).

Ответ: `tracks[]` с полями `kind`, `available`, `segments: [{ from, to, tone, value, issues? }]`.

`issues[]` на **warn/bad**: `{ key, tone, value }` — какие метрики сломали срез (ICMP/агент, loss/RTT, CPU/RAM/диски для ИЗ). Нужны для тултипа ленты и доп. линий на общем графике узла.

kind	Поведение
`sla`	Узел: ok/warn(partial)/bad; **канал** loss+RTT; **доступность** ok/warn(RTT)/bad(fail); пробелы ⟶ **unknown**; пересечение ППР ⟶ **maintenance**, исключение SLA ⟶ **exception**
`health`	Только **node**; снимки score ⟶ ok/warn/bad. Если текущий ИЗ критический (0 / tone `bad`) и открыта недоступность системы (`agent.offline`, ICMP, SNMP/ODBC/JMX/HTTP «нет сбора / сервис недоступен»), лента **пе-**

kind	Поведение
	рекрашивает этот интервал в красный — старые снимки ~97% не оставляют жёлтую «деградацию». Ручки выбора интервала на ленте ИЗ красятся по **этой** колбасе, не по SLA. Пока точек нет ⟶ `available: false`
`both`	Оба трека (обзор узла)

Канал / доступность на FE запрашивают `kind=sla`; узел — `both`.

Снимки ИЗ

- Пересчёт: `HealthIndexRecomputeWorker` (очередь, drain ~1.5 с) после **open/close** проблемы на сервисе, в том числе ручного закрытия; также после правки шаблона здоровья / override. При **удалении сервиса** ID снимается с очереди; persist не пишет summary/snapshot для несуществующей строки `services`.
- История: `service_health_snapshots` (Postgres, RANGE по месяцу). Строка пишется **только** если сменились `%` или тон. INSERT снимка — отдельная транзакция, чтобы FK на удалённый сервис не abort'ил основной persist и не давал `25P02` в логах.
- GET попапа / breakdown **не** пишет снимок
- Retention партиций: `wisla.health.snapshot-retention-months` (по умолчанию 12); DDL-only job `HealthSnapshotPartitionMaintainer`
- ClickHouse / `metric_last_values` для ИЗ **не** используются

Пока снимков нет, UI показывает: *«История индекса здоровья появится после накопления снимков»* (после первого события или первого заполнения списка).

Карта кода (чтобы переделать)

Backend

Что	Где
REST	`ServiceController` ⟶ `GET …/status-history`, `…/sla/summary`
Сборка истории	`ServiceStatusHistoryService`
Сегменты	`StatusHistoryMath` (+ тест `StatusHistoryMathTest`)
Причины warn/bad	`StatusHistoryIssueAnnotator`
Счёт ИЗ	`HealthScoringEngine` + `HealthIndexService`
Снимки истории	`ServiceHealthSnapshotStore` (`service_health_snapshots`)
Очередь пересчёта	`HealthIndexRecomputeWorker` (не флот-таймер)
Партиции	`HealthSnapshotPartitionMaintainer`
SLA узла	`NodeAvailabilitySlaService`
Кэш SLA	`SlaSummaryCache` (`wisla.sla.summary-cache-ms`)
SLA канала	`ChannelAvailabilitySlaService`, `SlaSummaryMath`
DTO	`ServiceDtos.StatusHistory*View`, `ChannelSlaSummaryView`
Док API	

Что	Где
	services-icmp.md

Frontend

Что	Где
Список: два чипа	`service-list/service-list-page.*` (`loadHealth`, `showsHealth` / `showsSla`)
Клиент API	`services/data/services-api.service.ts` → `slaSummary()`, `statusHistory()`
Timeline UI	`shared/components/status-history-timeline/` (тултип на warn/bad/unknown)
Линии инцидента	`host-dynamics-chart` + `status-history-issues.ts` (доп. серии на общем графике)
Обзор узла	`hosts/host-overview/` (+ `host-overview-data.service.ts`, период графика)
Обзор канала / доступности	`services/channel-overview/`, `services/availability-overview/`
i18n	`services.badge.*`, `services.statusHistory.*` в `core/i18n/messages.ts`
Док UI	services-list-page.md, host-overview-page.md

Типовые вопросы

В списке у узла «—» у SLA, а ИЗ есть.
Так и задумано на compact-стенде: список не сканирует ClickHouse за 24 ч по каждому узлу. SLA узла — на обзоре. ИЗ живёт от latest.

На timeline ИЗ пусто, SLA уже цветной.
Нормально, пока не было смены ИЗ (open/close события) и нет строки в `service_health_snapshots`. Первый проход списка может записать стартовый снимок для узлов без summary.

Почему на timeline у канала есть жёлтый сегмент, а у узла нет?
У узла жёлтый есть, если жив только один из сигналов (ICMP **или** агент). Канал — quality loss+RTT. Доступность — жёлтый от медленного RTT. Цвета SLA и ИЗ одинаковые (норма / деградация / недоступно / нет данных). Жёлтый **не** снижает % готовности.

Почему сегменты SLA только ok/bad, без warn?
Устарело: у канала/доступности/узла (при двух сигналах) — ok/warn/bad. Warn/bad у **цифры** SLA vs цель — отдельный слой.

Связанные документы

- Пользователь: Сервисы и узлы
- Админ / сайзинг: Системные требования
- Backend API: Services + ICMP
- Patch notes: changelog