

# Содержание

Мониторинг ePDG .....	3
<b>Комплексная система мониторинга шлюза VoWiFi (ePDG) .....</b>	<b>3</b>
<b>1. Обзор решения .....</b>	<b>3</b>
Ключевые преимущества .....	3
Четырёхуровневая архитектура мониторинга .....	3
<b>3. Компоненты и метрики .....</b>	<b>4</b>
Покрытие мониторингом .....	4
Количественный обзор по категориям .....	4
Принципы именования .....	4
<b>4. Перечень метрик .....</b>	<b>5</b>
4.1 Config (2) .....	5
4.2 Network (1) .....	5
4.3 IKEv2 SWu (3) .....	5
4.4 GTPv2-C S2b (4) .....	5
4.5 GTP-U data plane (3) .....	6
4.6 Diameter SWm/SWx/S6b (5) .....	6
4.7 Service KPI (4) .....	6
4.8 Session State (4) .....	6
4.9 Application (3) .....	6
4.10 System (4) .....	7
Типы метрик (напоминание) .....	7
<b>5. Интерфейсы интеграции .....</b>	<b>7</b>
5.1 Prometheus (CNCF Standard) .....	7
5.2 SNMP v2c — EPDG-MIB .....	7
5.3 Grafana .....	8
5.4 Alertmanager Webhooks .....	8
<b>6. Система алармов .....</b>	<b>9</b>
Категории алармов .....	9
Полный перечень алармов (20+ правил) .....	9
Процесс обработки алармов .....	9
Особенности .....	10
<b>7. Визуализация и операционные дашборды .....</b>	<b>10</b>
Состав дашбордов .....	10
Дизайн для Центра управления сетью (NOC) .....	10
<b>8. Интеграция в единый стек EPC Monitoring .....</b>	<b>11</b>
<b>9. Покрытие метрик по уровням OSI .....</b>	<b>11</b>
Детализация метрик по уровням .....	11
Уровень 9: Качество восприятия сервиса VoWiFi .....	12
<b>10. Стандарты и совместимость .....</b>	<b>12</b>
<b>11. Модель развёртывания .....</b>	<b>13</b>
Характеристики развёртывания .....	13
Варианты размещения .....	13
<b>12. Конфигурация экспортёра метрик .....</b>	<b>13</b>



# Мониторинг ePDG

## Комплексная система мониторинга шлюза VoWiFi (ePDG)

### 1. Обзор решения

Система мониторинга VAS Experts ePDG Monitoring обеспечивает полный операционный контроль компонента **fast-epdg** — шлюза VoWiFi (Voice over WiFi), работающего согласно 3GPP TS 29.273 и TS 24.302. Шлюз обеспечивает защищённую передачу голосового и пакетного трафика через недоверенные каналы Wi-Fi с IPSec/IKEv2 туннелированием и интеграцию с EPC-ядром через интерфейсы SWu, SWm, SWx, S2b, S6b.

Решение предоставляет единую платформу мониторинга для оперативных служб мобильного оператора — от уровня IPSec SA (L3 security) до KPI абонентского опыта VoWiFi.

### Ключевые преимущества

- **Мониторинг в реальном времени** — обновление метрик каждые 10-15 секунд, непосредственное отображение состояния IKE SA / Child SA и GTP-туннелей в NOC-дашбордах без отложенной агрегации (здесь и далее NOC — Network Operation Center, центр управления сетью).
- **Проактивное обнаружение аномалий** — 20+ алармов с автоматической эскалацией по важности. Недоступность PGW/AAA, рост задержек IKEv2, рост ошибок EAP-AKA' — детектируются до того, как абоненты заметят проблемы со звонками.
- **Открытые интерфейсы интеграции** — Prometheus, SNMP v2c, Alertmanager webhooks, поддержка Grafana. Интеграция в существующую NMS/OSS инфраструктуру без привязки к вендору.
- **Минимальные внешние зависимости на уровне плагинов** — встроенный /metrics endpoint в fast-epdg, без Java, без JMX, без внешних агентов.
- **Покрытие всего стека SWu → S2b** — IKEv2 (SWu), Diameter SWm/SWx/S6b, GTPv2-C (S2b) и GTP-U data plane — в одном месте. 33 метрики суммарно, покрывают control plane и data plane.

### Четырёхуровневая архитектура мониторинга

Уровень	Компонент	Технология
<b>Сбор</b> (Collection)	Встроенный /metrics endpoint fast-epdg	Текстовый формат Prometheus поверх HTTP
<b>Хранение</b> (Storage)	Prometheus TSDB	Локальное хранение, 15-дневное хранение по умолчанию
<b>Визуализация</b> (Visualization)	Grafana + поддержка JSON	Автозагрузка 4 дашбордов
<b>Сигнализация</b> (Alerting)	Alertmanager + SNMP Trap Sender	PromQL rules → webhook → SNMP v2c trap

## 3. Компоненты и метрики

### Покрытие мониторингом

flowchart LR  
EXP["fast-epdg /metrics :9817"] --> CFG["Config 2 метрики"]  
EXP --> NET["Network 1 метрика"]  
EXP --> PROTO["Protocols L5-L7 15 метрик"]  
EXP --> SVC["Service KPI 4 метрики"]  
EXP --> SESS["Session State 4 метрики"]  
EXP --> APP["Application 3 метрики"]  
EXP --> SYS["System 4 метрики"]  
PROTO --> IKEV2["IKEv2 SWu — 3"]  
PROTO --> GTPC["GTPv2-C S2b — 4"]  
PROTO --> GTPU["GTP-U S2b data — 3"]  
PROTO --> DIA["Diameter SWm/SWx/S6b — 5"]

### Количественный обзор по категориям

Категория	Кол-во метрик	Интервал опроса	Ключевые показатели
<b>Config</b>	2	10 с	Статус конфигурации, счётчик reload
<b>Network</b>	1	10 с	Статус соединений с узлами (PGW/AAA/HSS)
<b>IKEv2 (SWu)</b>	3	10 с	Сообщения по типам (IKE_SA_INIT, IKE_AUTH, CREATE_CHILD_SA), диаграмма задержки, ошибки
<b>GTPv2-C (S2b)</b>	4	10 с	Сообщения (Create/Modify/Delete Session), задержки, ошибки, ретрансляции
<b>GTP-U data plane</b>	3	10 с	Packets/bytes, ошибки туннелирования
<b>Diameter (SWm/SWx/S6b)</b>	5	10 с	Сообщения по command code (DER/DEA, MAR/MAA, AAR/AAA), задержки, ошибки, watchdog, статус соединения
<b>Service KPI</b>	4	10 с	Процент успешных попыток, гистограмма продолжительности, доступность сервиса, время безотказной работы
<b>Session State</b>	4	10 с	IKE SA, Child SA, GTP-сессии, всего пользователей
<b>Application</b>	3	10 с	Количество потоков, память, лог-сообщения по уровням
<b>System</b>	4	10 с	Утилизация CPU, память, утилизация памяти, открытые FD
<b>Итого</b>	<b>33 метрики</b>		

### Принципы именования

Все метрики имеют префикс epdg\_ и организованы по иерархии:

```

epdg_
├── config_*           # Конфигурация
├── network_*         # Сетевой уровень
├── ikev2_*           # SWu (IKEv2/IPSec)
├── gtp_*             # S2b control-plane GTPv2-C
├── gtpu_*            # S2b data-plane GTP-U
├── diameter_*        # SWm/SWx/S6b
├── service_*         # KPI сервисов (attach, availability, uptime)
├── session_*         # Состояние сессий (IKE SA, Child SA, GTP,
subscribers)
├── app_*             # Метрики приложения (memory, threads, logs)
└── system_*         # Системные метрики (CPU, disk, network)

```

## 4. Перечень метрик

Все метрики экспортируются через единый `/metrics` endpoint в текстовом формате Prometheus. Именованье следует правилам Prometheus: `epdg_<группа>_<имя>[_unit]`, тип Counter имеет суффикс `_total`, Histogram — суффикс `_seconds` / `_bytes`.

### 4.1 Config (2)

Имя	Тип	Назначение
<code>epdg_config_status</code>	Gauge	Статус конфигурации компонента (0=error, 1=ok)
<code>epdg_config_reload_total</code>	Counter	Счётчик конфигурационных загрузок (success/failure)

### 4.2 Network (1)

Имя	Тип	Назначение
<code>epdg_network_connection_status</code>	Gauge	Статус TCP/UDP-соединения к узлу (0=down, 1=up) — применяется к PGW (S2b), AAA (SWm), HSS (SWx)

### 4.3 IKEv2 SWu (3)

Имя	Тип	Назначение
<code>epdg_ikev2_messages_total</code>	Counter	Счётчик IKEv2 сообщений (IKE_SA_INIT / IKE_AUTH / CREATE_CHILD_SA / INFORMATIONAL)
<code>epdg_ikev2_request_duration_seconds</code>	Histogram	Распределение времени ответа на запросы IKEv2
<code>epdg_ikev2_errors_total</code>	Counter	IKEv2 ошибки (NO_PROPOSAL_CHOSEN, AUTHENTICATION_FAILED, INVALID_SYNTAX и т.д.)

### 4.4 GTPv2-C S2b (4)

Имя	Тип	Назначение
<code>epdg_gtp_messages_total</code>	Counter	Сообщения GTPv2-C (Create/Modify/Delete Session, Echo)
<code>epdg_gtp_request_duration_seconds</code>	Histogram	Время ожидания запрос → ответ

Имя	Тип	Назначение
epdg_gtp_errors_total	Counter	GTP-C ошибки по Cause Code
epdg_gtp_retransmissions_total	Counter	Перенаправление GTP-C запросов

#### 4.5 GTP-U data plane (3)

Имя	Тип	Назначение
epdg_gtpu_packets_total	Counter	Пакеты через GTP-U туннель (uplink/downlink)
epdg_gtpu_bytes_total	Counter	Байты через GTP-U туннель
epdg_gtpu_errors_total	Counter	Ошибки туннелирования (TEID mismatch, decap fail)

#### 4.6 Diameter SWm/SWx/S6b (5)

Имя	Тип	Назначение
epdg_diameter_messages_total	Counter	DER/DEA (SWm), MAR/MAA (SWx), AAR/AAA (S6b), STR/STA
epdg_diameter_request_duration_seconds	Histogram	Время ожидания запрос → ответ по Diameter
epdg_diameter_errors_total	Counter	Ошибки по Experimental-Result-Code
epdg_diameter_watchdog_status	Gauge	Статус DWR/DWA watchdog до узла (0=timeout, 1=ok)
epdg_diameter_connection_status	Gauge	Статус соединения по Diameter до узла (0=disconnected, 1=connected)

#### 4.7 Service KPI (4)

Имя	Тип	Назначение
epdg_service_attach_total	Counter	Попытки соединения (success/failure) по APN
epdg_service_attach_duration_seconds	Histogram	Длительность соединения (IKE_SA_INIT → session ready)
epdg_service_availability	Gauge	Флаг доступности (0=down, 1=up)
epdg_service_uptime_seconds	Gauge	Время доступности сервиса

#### 4.8 Session State (4)

Имя	Тип	Назначение
epdg_session_ike_sa_total	Gauge	Активные IKE SA
epdg_session_child_sa_total	Gauge	Активные Child SA (IPSec tunnels)
epdg_session_gtp_sessions_total	Gauge	Активные GTP-C сессии на S2b
epdg_session_subscribers_total	Gauge	Уникальные абоненты (подключённые UE)

#### 4.9 Application (3)

Имя	Тип	Назначение
epdg_app_threads_total	Gauge	Общее число рабочих потоков
epdg_app_memory_bytes	Gauge	Память процесса по типам
epdg_app_log_messages_total	Counter	Лог-сообщения по уровням (debug/info/warn/error/fatal)

## 4.10 System (4)

Имя	Тип	Назначение
epdg_system_cpu_usage_percent	Gauge	Загрузка CPU
epdg_system_memory_bytes	Gauge	Системная память
epdg_system_disk_bytes	Gauge	Дисковое пространство
epdg_system_open_fds	Gauge	Открытые описания файлов

### Типы метрик (напоминание)

Тип	Назначение
<b>Counter</b>	Моноotonно растущий счётчик (сообщения, ошибки, перезагрузки)
<b>Gauge</b>	Текущее значение (активные сессии, память, статус)
<b>Histogram</b>	Распределение значений с автоматическими срезами по интервалам (длительность, время жизни)

## 5. Интерфейсы интеграции

flowchart LR  
CORE["VAS Experts  
ePDG Monitoring"] CORE --> P["Prometheus  
CNCF / OpenMetrics"]  
CORE --> S["SNMP v2c  
EPDG-MIB"]  
CORE --> G["Grafana  
JSON Provisioning"]  
CORE --> W["Webhooks  
ChatOps"]  
CORE --> AM["Alertmanager  
Routing"]  
P --> P1["Cloud-native NMS  
Thanos / Cortex / Mimir"]  
S --> S1["Legacy NMS  
HP OpenView, NetAct  
IBM Tivoli"]  
G --> G1["NOC Wall Displays  
Drill-down Analytics"]  
W --> W1["Telegram / Slack  
PagerDuty / OpsGenie"]  
AM --> AM1["Smart routing  
Severity-based"]

### 5.1 Prometheus (CNCF Standard)

Нативный `/metrics` endpoint на порту **9817** встроен в fast-epdg. Формат — стандартный текстовый формат Prometheus v0.0.4 (совместим с OpenMetrics). Поддерживается объединение для агрегации с центральным Prometheus оператора; поддержка команды `remote_write` для долгосрочного хранения в Thanos, Cortex, Grafana Mimir.

### 5.2 SNMP v2c — EPDG-MIB

**47 OID** покрывают SMI-аналог Prometheus-метрик + **14 trap notifications** (с парами raise/clear согласно RFC 3877 ALARM-MIB). Совместимость с HP OpenView, IBM Tivoli NetCool, Nokia NetAct, Huawei U2000.

flowchart TB  
IANA["IANA PEN"]

enterprises  
.1.3.6.1.4.1"] VAS["VAS Experts  
.1.3.6.1.4.1.43823  
(vas.expert") EPDG["EPDG-MIB  
.43823.1"] EPC["EPC Monitoring  
.43823.100"] IANA --> VAS VAS --> EPDG VAS --> EPC EPDG --> OBJ["epdgObjects  
.43823.1.1"] EPDG --> NOTIF["epdgNotifications  
.43823.1.2  
14 trap types"] EPDG --> CONF["epdgConformance  
.43823.1.3"] OBJ --> SERVICE["service .1.1.1  
4 OID"] OBJ --> IKE["ikev2 .1.1.2  
6 OID"] OBJ --> GTP["gtp .1.1.3  
8 OID"] OBJ --> DIAM["diameter .1.1.4  
7 OID"] OBJ --> SESS["sessions .1.1.5  
8 OID"] OBJ --> SYS["system .1.1.6  
8 OID"] OBJ --> NET["network .1.1.7  
6 OID"] NOTIF --> TRAPAGR["7 raise / 7 clear  
pairs"]

Примеры SNMP-запросов:

```
# Все дерево ePDG
snmpwalk -v2c -c public <host> .1.3.6.1.4.1.43823.1

# Service availability (Gauge 0..1)
snmpget -v2c -c public <host> .1.3.6.1.4.1.43823.1.1.1.1.0
```

## 5.3 Grafana

**4 поддерживающих JSON дашборда** (35+ панелей суммарно):

- **ePDG Overview** — доступность, KPI соединений, сессии, состояние интерфейсов
- **IKEv2 Details** — сообщения, производительность, ошибки, жизненный цикл IKE SA
- **GTP Details** — GTPv2-C + GTP-U данные по PGW узлам
- **Diameter Details** — сообщения по приложениям, задержки, watchdog

Автоматическая установка через API, поддерживающее Grafana. Адаптивный дизайн для мониторов состояния центра управления сети (NOC) с автообновлением каждые 15 секунд.

## 5.4 Alertmanager Webhooks

Webhook-интерфейс для интеграции с любой системой оповещений: Telegram Bot, Slack, PagerDuty Events API v2, OpsGenie, Microsoft Teams. Отдельный **SNMP Trap Sender** service конвертирует Alertmanager webhooks в SNMP v2c traps с Enterprise OID.

## 6. Система алармов

### Категории алармов

Критичность	Алармы	Описание	Реакция
<b>Critical</b>	ePDG_Service_Down, ePDG_High_Attach_Failure_Rate, ePDG_PGW_Unreachable, ePDG_AAA_Unreachable, ePDG_Diameter_Watchdog_Timeout	Компонент недоступен, массовый отказ соединений, узлы недоступны	Немедленная эскалация: Email + SNMP Trap + Webhook. Повтор раз в 1 час
<b>Warning</b>	ePDG_High_IKEv2_Latency, ePDG_High_GTP_Latency, ePDG_High_IKEv2_Error_Rate, ePDG_High_GTP_Error_Rate, ePDG_High_Memory_Usage, ePDG_High_CPU_Usage, ePDG_Low_Disk_Space, ePDG_High_Error_Log_Rate	Деградация производительности, аномалии ресурсов	Email. Повтор раз в 4 часа. Подавляется при наличии Critical на том же компоненте

### Полный перечень алармов (20+ правил)

```

flowchart LR
  AL["ePDG Alert Rules  
20+"] --> CR["Critical  
5 rules"]
  AL --> WR["Warning  
8 rules"]
  AL --> INFO["Recording  
34 rules"]
  CR --> C1["Service_Down  
availability == 0"]
  CR --> C2["Attach_Failure_Rate  
> 10%"]
  CR --> C3["PGW_Unreachable  
connection_status{s2b} == 0"]
  CR --> C4["AAA_Unreachable  
connection_status{swm} == 0"]
  CR --> C5["Diameter_Watchdog_Timeout  
watchdog_status == 0"]
  WR --> W1["High_IKEv2_Latency  
p95 > 1.0 s"]
  WR --> W2["High_GTP_Latency  
p95 > 0.5 s"]
  WR --> W3["High_IKEv2_Error_Rate  
> 5%"]
  WR --> W4["High_GTP_Error_Rate  
> 5%"]
  WR --> W5["High_Memory_Usage  
> 80%"]
  WR --> W6["High_CPU_Usage  
> 80%"]
  WR --> W7["Low_Disk_Space  
< 10%"]
  WR --> W8["High_Error_Log_Rate  
> 10/s"]
  INFO --> I1["attach_success_rate  
preaggregated"]
  INFO --> I2["p95_p99_latency  
preaggregated"]
  INFO --> I3["throughput  
preaggregated"]
  
```

### Процесс обработки алармов

```

sequenceDiagram
  participant M as Метрика (Prometheus)
  participant R as Alert Rule (PromQL)
  
```

participant AM as Alertmanager participant E as Email (SMTP) participant SG as SNMP Trap Gateway participant NMS as Внешняя NMS participant W as Webhook (ChatOps) M->>R: Значение превышает порог R->>R: Ожидание (for: 1-10 мин) R->>AM: Alert FIRING AM->>AM: Group by [alertname, component] AM->>AM: Inhibition check (critical подавляет warning) alt severity = critical AM->>E: Email [CRITICAL] AM->>SG: Webhook → SNMP Trap SG->>NMS: SNMP v2c Trap (OID .1.3.6.1.4.1.43823.1.2.X) AM->>W: Webhook (Telegram / PagerDuty) else severity = warning AM->>E: Email [WARNING] end Note over M,R: Метрика возвращается в норму R->>AM: Alert RESOLVED R->>SG: clear-trap (paired notification) AM->>E: Email [RESOLVED]

## Особенности

- **Inhibition:** Critical-алармы автоматически подавляют Warning для того же компонента
- **Grouping:** Алармы группируются по alertname + component с 30-секундным окном
- **Dead time / Hysteresis:** параметр for от 1 до 10 минут предотвращает ложные срабатывания
- **Trap pairing:** raise/clear одновременных событий для соответствия RFC 3877 ALARM-MIB

## 7. Визуализация и операционные дашборды

### Состав дашбордов

Дашборд	Панели	Назначение
<b>ePDG Overview</b>	10	Доступность сервиса, коэффициент успешного подключения, количество активных сеансов, состояние SWu/SWm/S2b, пропускная способность по интерфейсам
<b>IKEv2 Details</b>	10	Сообщения в секунду по типам, гистограмма продолжительности запросов, задержка в 95-м процентиле, ошибки по типам, жизненный цикл IKE SA
<b>GTP Details</b>	8	Сообщения GTPv2-C по PGW, повторные передачи, ошибки по коду причины, GTP-U (восходящий/нисходящий канал), несущие
<b>Diameter Details</b>	7	Количество сообщений по приложениям (SWm/SWx/S6b), продолжительность запросов, состояние сторожевого таймера, распределение кодов результатов, хронология состояний соединений

### Дизайн для Центра управления сетью (NOC)

flowchart TB
 NOC["NOC Dashboard Layer"] --> OVER["ePDG Overview KPI Summary"]
 NOC --> IKE["IKEv2 Details Drill-down"]
 NOC --> GTP["GTP Details Drill-down"]
 NOC --> DIA["Diameter Details Drill-down"]
 OVER -->|Click attach KPI| IKE
 OVER -->|Click session count| GTP
 OVER -->|Click peer status| DIA

- **Автообновление:** 15-секундный период обновления
- **Адаптивная цветовая схема:** зелёный → жёлтый → красный по пороговым значениям
- **Drill-down:** от Overview к детализации до компонента
- **Time-range selector:** от 5 минут до 30 дней истории
- **JSON provisioning:** дашборды разворачиваются автоматически

## 8. Интеграция в единый стек EPC Monitoring

ePDG мониторинг полностью интегрирован в общий мониторинг пакетного ядра:

```
flowchart TB
    subgraph Common["Единый Monitoring Stack"]
        PROM["Prometheus"]
        GRAF["Grafana"]
        AM["Alertmanager"]
    end
    subgraph Sources["Источники метрик EPC"]
        DPI["FastDPI:9110"]
        SMF["SMF /metrics:9090"]
        PCEF["fast-pcef /metrics:9090"]
        PCRF["FastPCRF"]
        EPDG["fast-epdg:9817"]
    end
    DPI --> PROM
    SMF --> PROM
    PCEF --> PROM
    PCRF --> PROM
    EPDG --> PROM
    PROM --> GRAF
    PROM --> AM
```

Оператор NOC видит **все компоненты EPC** (DPI, SMF, PCEF, FastPCRF, ePDG) в одном интерфейсе Grafana, с единой системой алармов и маршрутизации уведомлений через один Alertmanager.

## 9. Покрытие метрик по уровням OSI

```
graph LR
    L1["L1 Physical  
NIC counters via system"] --> L2["L2 Data Link  
MAC, VLAN"]
    L2 --> L3["L3 Network  
IP, IPsec ESP, GTP-U"]
    L3 --> L4["L4 Transport  
TCP/UDP/SCTP"]
    L4 --> L5["L5 Session  
GTPv2-C, IKEv2"]
    L5 --> L6["L6 Presentation  
IKEv2/IPsec encryption, EAP-AKA"]
    L6 --> L7["L7 Application  
Diameter, service bearer ops"]
    L7 --> Operations["Operations  
KPI, SLA, Capacity"]
    Operations --> CX["CX Level  
Subscriber Experience"]
```

style L1 fill:#e74c3c,color:#fff style L2 fill:#e67e22,color:#fff style L3 fill:#f39c12,color:#fff style L4 fill:#2ecc71,color:#fff style L5 fill:#1abc9c,color:#fff style L6 fill:#3498db,color:#fff style L7 fill:#9b59b6,color:#fff style Operations fill:#34495e,color:#fff style CX fill:#2c3e50,color:#fff

### Детализация метрик по уровням

Модель OSI:

Уровень	Метрики	Примеры
<b>L1/L2 Physical / Data Link</b>	—	Покрывается отдельным node_exporter / аналогом на уровне ОС (не входит в перечень метрик ePDG)
<b>L3 Network / IPsec tunnels</b>	3	epdg_gtpu_packets_total, epdg_gtpu_bytes_total, epdg_gtpu_errors_total — GTP-U data plane
<b>L4 Transport</b>	1	epdg_network_connection_status — TCP к узлам (PGW/AAA/HSS)
<b>L5 Session</b>	3	epdg_session_ike_sa_total, epdg_session_child_sa_total, epdg_session_gtp_sessions_total

Уровень	Метрики	Примеры
<b>L6 Presentation/Security</b>	3	epdg_ikev2_messages_total, epdg_ikev2_request_duration_seconds, epdg_ikev2_errors_total — IKEv2/IPSec шифрование и EAP-AKA' аутентификация
<b>L7 Application</b>	9	epdg_diameter_* (SWm/SWx/S6b, 5 метрик), epdg_gtp_* (GTPv2-C, 4 метрики)

Операторский уровень:

Уровень	Метрики	Примеры
<b>Operations</b>	11	epdg_service_availability, epdg_service_uptime_seconds, epdg_app_* (3), epdg_system_* (4), epdg_config_* (2)
<b>Customer Experience</b>	3	epdg_service_attach_duration_seconds p95, epdg_service_attach_total (success rate), epdg_ikev2_request_duration_seconds p99

## Уровень 9: Качество восприятия сервиса VoWiFi

QoE-индикатор	Метрики-источники	Интерпретация
<b>Время подключения VoWiFi</b>	epdg_service_attach_duration_seconds p95	> 3 сек — абонент замечает задержку при переключении на WiFi
<b>Непрерывность сервиса</b>	epdg_session_ike_sa_total delta	Массовый сброс > 50 IKE SA = проблема доступности
<b>Успешность аутентификации</b>	ePDG_High_Attach_Failure_Rate alert rate	> 5% = проблема с HSS/AAA узлами
<b>Задержка назначения bearer</b>	epdg_gtp_request_duration_seconds{msg=create-session} p99	> 500 мс — задержка готовности голосового канала
<b>Ошибки GTP-U tunnel</b>	epdg_gtpu_errors_total rate / epdg_gtpu_packets_total	> 0.1% = деградация качества голоса
<b>IKEv2-надёжность</b>	epdg_ikev2_errors_total по типам	NO_PROPOSAL_CHOSEN / AUTHENTICATION_FAILED — проблемы с certs / UE

## 10. Стандарты и совместимость

Стандарт	Область	Применение
<b>3GPP TS 29.273</b>	SWx/S6b/SWm	Методология учёта Diameter-сообщений и результирующих кодов
<b>3GPP TS 24.302</b>	SWu (IKEv2)	Определение IKEv2 типов сообщений и кодов ошибок
<b>3GPP TS 33.402</b>	3GPP security for non-3GPP access	Параметры безопасности EAP-AKA' / IKEv2
<b>3GPP TS 23.402</b>	Non-3GPP access architecture	Структура интерфейсов (SWu/SWm/SWx/S6b/S2b)
<b>3GPP TS 32.421</b>	Performance Measurement	Методология сбора KPI
<b>3GPP TS 32.409</b>	Performance measurement charging	Структура счётчиков
<b>IETF RFC 7296</b>	IKEv2	Типы сообщений, уведомления об ошибках, состояние SA

Стандарт	Область	Применение
<b>IETF RFC 6733</b>	Diameter	Command codes, Result-Codes
<b>IETF RFC 4187</b>	EAP-AKA	Аутентификация через SIM
<b>IETF RFC 3877</b>	ALARM MIB	Структура Enterprise MIB для алармов
<b>IETF RFC 3418</b>	SNMPv2 MIB	Совместимость SNMP v2c
<b>Prometheus Exposition Format</b>	Metrics (v0.0.4)	Формат экспорта метрик
<b>OpenMetrics</b>	CNCF Standard	Перспективная совместимость

## 11. Модель развёртывания

```

flowchart TB
    subgraph Host1 ["Сервер ePDG"]
        EPDG1["fast-epdg (VoWiFi gateway)"]
        PLUG1["/metrics endpoint :9817"]
        EPDG1 --> PLUG1
    end
    subgraph Host2 ["Сервер мониторинга"]
        PROM["Prometheus"]
        GRAF["Grafana"]
        AM["Alertmanager"]
        SNMPTRAP["SNMP Trap Sender (webhook gateway)"]
        PROM --> GRAF
        PROM --> AM
        AM --> SNMPTRAP
    end
    subgraph Host3 ["Внешние системы"]
        NMS["Операторская NMS (HP OpenView / NetAct / Tivoli)"]
        CHAT["ChatOps (Telegram / PagerDuty)"]
    end
    PLUG1 --> |HTTP :9817/metrics| PROM
    SNMPTRAP --> |UDP 162| NMS
    AM --> |Webhook| CHAT

```

### Характеристики развёртывания

Параметр	Значение
<b>Metrics footprint</b>	Интегрированные (~2 MB memory overhead)
<b>Внешние зависимости</b>	Самодостаточный пакет fast-epdg (rpm)
<b>Управление</b>	fast-epdg.service systemd
<b>Конфигурация</b>	Секция monitoring в fast-epdg.conf
<b>Обновление</b>	Обновление конфигурации без прерывания работы
<b>ОС</b>	Linux (RHEL/CentOS 8+, Ubuntu 22.04+)
<b>Порт</b>	9817 TCP (listen 0.0.0.0, настраивается)
<b>Время развёртывания</b>	< 5 минут (enable plugin в config + restart)

### Варианты размещения

- **On-premise** — плагин работает в адресном пространстве fast-epdg, нулевое потребление ресурсов
- **Co-located Prometheus** — Prometheus собирает метрики с приложения, работающего на том же хосте
- **Централизованный** — единый Prometheus собирает со всех ePDG узлов

## 12. Конфигурация экспортёра метрик

Секция monitoring в fast-epdg.conf:

```
monitoring {
    enabled = yes
    listen_port = 9817
    listen_address = 0.0.0.0
    update_interval = 10
    metrics {
        ikev2 = yes
        gtp = yes
        diameter = yes
        service = yes
        session = yes
        app = yes
        system = yes
    }
}
```

Каждая группа метрик может быть независимо включена/выключена без перекомпиляции.