

# Функциональные характеристики программного обеспечения «Скоринг активности пользователя»

## 1. Описание

Настоящий документ содержит описание функциональных характеристик программного обеспечения «Скоринг активности пользователя» (далее «Сервис»). Сервис реализует JSON-RPC 2.0 API для взаимодействия с клиентскими системами.

## 2. Среда функционирования продукта

Сервис функционирует в любой среде, контейнерной виртуализации. Предпочтительной средой являются системы оркестрации Docker Swarm или Kubernetes.

## 3. Функциональные требования:

Система предназначена для вычисления интегрального показателя активности игрока (скоринга) на основании переданных заказчиком событий и статистики. Цель — предоставить заказчику единый внешний инструмент оценки уровня вовлечённости, вероятности возврата, качества текущей игровой активности и динамики изменений по каждому пользователю. Скоринг используется системой рекомендаций, CRM, маркетингом, антифрод-решениями и аналитикой.

Сервис полностью автономен: он не имеет прямого доступа к игровой платформе и обрабатывает только те данные, которые заказчик сам отправляет.

Сервис представляет следующую функциональность:

### 3.1. Приём событий активности пользователя через JSON-RPC

- Приём событий пользовательской активности в формате JSON-RPC 2.0
- Поддержка типов событий: вход/выход пользователя, запуск/завершение игры, игровые сессии, ставки, выигрыши, события конверсии
- Каждое событие содержит: обезличенный идентификатор пользователя, тип события, временную метку, дополнительные параметры
- Batch-обработка событий через JSON-RPC batch requests

### 3.2. Расчёт скорингового балла

- Расчёт скоринга на основе нескольких групп метрик: поведенческих, финансовых, динамических и глобальных
- Поддержка нескольких типов скорингов: EngagementScore, RetentionScore, ActivityScore, VolatilityScore, MonetaryScore
- Диапазон значений: 0–100, 0–1 или percentiles (1–99) с возможностью конфигурации

### **3.3. История и динамика скоринга**

- Хранение дневных/недельных агрегатов активности пользователя
- Сохранение истории скорингов
- Отслеживание изменений скоринга в сравнении с предыдущими периодами
- Поддержка временных диапазонов при запросе истории

### **3.4. JSON-RPC API для взаимодействия**

- Методы API:
  - `scoring.trackEvent` — приём события активности
  - `scoring.getScore` — получение актуального скоринга пользователя
  - `scoring.getScoreHistory` — получение истории скоринга за период
  - `scoring.recalculateScore` — принудительный пересчёт скоринга
  - `scoring.getTopUsers` — получение списка самых активных пользователей
  - `system.getConfig` — получение конфигурации сервиса
  - `system.updateConfig` — обновление конфигурации

### **3.5. Механизм конфигурации через JSON-RPC**

- Настройка весовых коэффициентов метрик через методы API
- Конфигурация периода анализа (сутки/неделя/месяц)
- Настройка нормализации скорингов
- Конфигурация чувствительности к всплескам активности
- Определение поведения при отсутствии данных

### **3.6. Логгирование и мониторинг**

- Логгирование всех JSON-RPC запросов и ответов
- Трейсинг цепочки выполнения методов
- Встроенная подсистема мониторинга для отслеживания ошибок
- Метрики производительности JSON-RPC обработчиков
- Возможность интеграции с системами мониторинга

### **3.7. Масштабируемость**

- Поддержка автоматического масштабирования средствами систем оркестрации
- Распределённая обработка JSON-RPC запросов

## **4. Системные требования к ПО**

### **Минимальные аппаратные требования:**

- Операционная система, способная запускать контейнеры. Предпочтительно Linux
- Система управления контейнерной виртуализацией. Предпочтительно Docker Swarm или Kubernetes
- Количество логических ядер процессора: 4
- Семейство процессоров: x86
- Частота процессора: 2.5 ГГц

- Объем установленной памяти: 8 Гб
- Объем дискового пространства: 50 Гб

#### **4.1. Минимальные требования к сторонним компонентам и/или системам, необходимым для установки и работы ПО**

- Debian 11 или Ubuntu 20.04+ (Открытая лицензия GNU)
- Docker 24.0+ (open-source community edition)
- PostgreSQL 16+ (Открытая лицензия PostgreSQL license)
- Prometheus 2.45+ (Открытая лицензия Apache 2.0) - для мониторинга
- Grafana 10.0+ (Открытая лицензия GNU) - для визуализации метрик

#### **4.2. Языки программирования**

При разработке Сервиса используется язык программирования GoLang 1.21+ (Открытая лицензия BSD)

### **5. Ограничения**

- Сервис не хранит персональных данных пользователей
- Сервис получает только обезличенные идентификаторы
- Качество скоринга напрямую зависит от объёма переданной заказчиком телеметрии
- Сервис не может проверять корректность присылаемых данных