



**Описание функциональных характеристик
продукта Platform V Monitor (OPM)**

Оглавление

ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК КОМПОНЕНТА АІОМ (АІОМ)	4
ОПИСАНИЕ ПРОДУКТА.....	4
<i>Назначение.....</i>	4
<i>Цель создания</i>	4
<i>Основные функции (сущностные)</i>	4
<i>Сценарии использования.....</i>	6
ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК КОМПОНЕНТА ALERT MANAGER (ALRT)	6
ОПИСАНИЕ ПРОДУКТА.....	6
<i>Назначение.....</i>	6
<i>Цель создания</i>	6
<i>Основные функции</i>	6
<i>Сценарии использования.....</i>	7
ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК КОМПОНЕНТА BUSINESS ACTIVITY MONITORING (BAMN).....	10
Термины и определения	10
<i>Назначение.....</i>	11
<i>Цель создания</i>	11
<i>Основные функции(сущностные)</i>	11
<i>Сценарии использования.....</i>	11
ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК КОМПОНЕНТА TELEMETRY COLLECTOR (ЕДИНЫЙ КОЛЛЕКТОР ТЕЛЕМЕТРИИ) (COTE)	16
ОПИСАНИЕ ПРОДУКТА.....	16
<i>Термины и определения.....</i>	16
<i>Назначение.....</i>	17
<i>Цель создания</i>	17
<i>Основные функции</i>	17
<i>Сценарии использования.....</i>	21
ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК КОМПОНЕНТА INDICATOR (INDA) 27	
ОПИСАНИЕ ПРОДУКТА.....	27
<i>Термины и определения.....</i>	27
<i>Назначение.....</i>	28
<i>Основные функции сервиса:</i>	29
<i>Сценарии использования.....</i>	29
ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК КОМПОНЕНТА ABySS (LGDB).....	30
НАЗНАЧЕНИЕ	30
<i>Термины и определения.....</i>	30
<i>Основные функции</i>	30
<i>Сценарии использования.....</i>	31
ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК КОМПОНЕНТА ЖУРНАЛИРОВАНИЕ (LOGA).....	34

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	34
ОПИСАНИЕ ПРОДУКТА.....	34
<i>Назначение</i>	34
<i>Цель создания</i>	34
<i>Основные функции</i>	35
<i>Сценарии использования</i>	36
ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК КОМПОНЕНТА ОБЪЕДИНЕННЫЙ МОНИТОРИНГ UNIMON (MONA).....	38
ОПИСАНИЕ ПРОДУКТА.....	38
<i>Термины и определения</i>	38
<i>Назначение</i>	38
<i>Цель создания</i>	38
<i>Основные функции (сущностные)</i>	38
<i>Сценарии использования</i>	41

Описание функциональных характеристик компонента AIOM (AIOM)

Описание продукта

Назначение

AIOM — это сервис в составе программного продукта Platform V Monitor (OPM), использующий статистические алгоритмы для выявления аномального поведения объектов анализа на основе метрик.

Цель создания

AIOM — создание инструментов, позволяющих автоматически получить наглядное представление о состоянии платформенных компонентов по прикладным метрикам в сравнении с предыдущим периодом работы сервиса.

Детальное описание:

- CloudNative веб-приложение для аналитики качества работы компонентов и сервисов, позволяющее объединить в «единое окно» качества работы всех элементов экосистемы в соответствии с желаниями и возможностями пользователей от ИТ экспертов до бизнес-пользователей;
- Гибкий пользовательский интерфейс, позволяющий производить настройку отображения данных в соответствии с текущими потребностями пользователя. Настройка производится в интуитивно-понятном графическом интерфейсе, с расширением функциональности через плагины;
- Позволяет автоматически проанализировать работу всех необходимых сервисов выявить потенциальные угрозы сбоям от простого алгоритма к сложным механизмам предиктивного детектирования отклонений на основании онлайн потока данных о здоровье ИТ составляющих экосистемы с указанием источника;
- Позволяет обеспечить релевантное информирование и управление возникшими аномалиями с вовлечением экспертов для скорейшей минимизации потенциального влияния на любую клиентскую операцию.

Основные функции (сущностные)

Функция	Потребитель функции	Аргумент функции	Результат
Подключать сервис AIOM в Indicator.	Пользователь с правами администратора	Запрос к хранилищу данных по именам метрик	Отображение графика, согласно условий запроса

Функция	Потребитель функции	Аргумент функции	Результат
Включать/отключать отображение базовой линии на панели в Indicator.	Пользователь с правами на редактирование панели	Запрос к хранилищу данных по именам метрик	Отображение графика, согласно условиям запроса
Задавать параметры коридоров отклонения от базовой линии и их отображение.	Пользователь с правами на редактирование панели	Запрос к хранилищу данных по именам метрик	Отображение графика, согласно условиям запроса
Задать параметры исключения определенных периодов из расчета базовой линии.	Пользователь с правами на редактирование панели	Запрос к хранилищу данных по именам метрик	Возвращение значений метрик исключением периода отображения, согласно условиям запроса
Задать параметры исключения определенных периодов из расчета базовой линии.	Пользователь с правами на редактирование панели	Запрос к хранилищу данных по именам метрик	Возвращение значений метрик исключением периода отображения, согласно условиям запроса
Просматривать график с базовой линией.	Пользователь с правами на редактирование панели	Запрос к хранилищу данных по именам метрик	Возвращение значений метрик исключением периода отображения, согласно условиям запроса
Импорт/экспорт календаря.	Пользователь с правами администратора.	Файл с данными календаря.	Добавление пользовательского календаря в сервис AIOM.

Сценарии использования

Базовая линия - это график построенный по усреднённому значению метрики за соответствующие периоды времени.

Период времени составляет 1 минуту с 00 до 59, с 00 до 23 часов, с понедельника по воскресенье.

Различные сценарии использования сервиса АІОМ доступны в руководствах оператора и администратора.

Описание функциональных характеристик компонента Alert Manager (ALRT)

Описание продукта

Назначение

Alert Manager - сервис для уведомления пользователей об отклонениях в значениях метрик.

Цель создания

Необходимо наличие компонента, позволяющего обнаруживать по заданным условиям отклонения в значениях метрик, поступающих с систем-потребителей (включая комплексные условия, объединяющие несколько метрик), и по событию обнаружения таких отклонений осуществлять действия (например, рассылку уведомлений).

Основные функции

Название функции	Потребитель функции	Аргументы функции	Результат
Настройка канала уведомлений	Пользователь	Тип канала, дополнительные аргументы	Создан канал для последующей отправки уведомлений
Группировка получателей	Пользователь	Список получателей	Создана группа из нескольких получателей для последующей отправки уведомлений

Название функции	Потребитель функции	Аргументы функции	Результат
Создание шаблона уведомлений	Пользователь	Канал, группы, сообщение	Создан шаблон для отправки уведомлений
Настройка правила отклонений	Пользователь	Запрос, условия, метки	Создано правило, позволяющее обнаруживать отклонения в значениях метрик
Настройка правила уведомлений	Пользователь	Правила отклонений, фильтры, условия, шаблоны уведомлений	Создано правило, позволяющее отправлять уведомления при обнаружении отклонений
Обнаружение событий	Пользователь	Правило отклонений и уведомлений, метрики	Сформированы события по обнаруженным отклонениям в значениях метрик
Отправка уведомлений	Получатель	Правила уведомлений, события, шаблоны уведомлений	Получателям отправлены уведомления об обнаруженных отклонениях

Сценарии использования

Настройка правила отклонений

Основной сценарий

Шаг 1. Пользователь заполняет настройки правила отклонений в UI и отправляет запрос к серверу.

Шаг 2. Сервер проверяет запрос на соответствие формату. (ИС-1)

Шаг 3. Правило сохраняется в базу данных и запускается его периодическое исполнение.

Шаг 4. Сервер подтверждает успешность сохранения правила.

Исключительные сценарии

ИС-1: Запрос не соответствует протоколу

Сервер возвращает код ошибки 400 и сообщение с описанием неправильно переданных значений.

Настройка правила уведомлений

Основной сценарий

Шаг 1. Пользователь заполняет настройки правила уведомлений в UI и отправляет запрос к серверу.

Шаг 2. Сервер проверяет запрос на соответствие формату. (ИС-1)

Шаг 3. Правило сохраняется в базу данных и начинаются его проверки при срабатывании правил отклонений.

Шаг 4. Сервер подтверждает успешность сохранения правила.

Исключительные сценарии

ИС-1: Запрос не соответствует протоколу

Сервер возвращает код ошибки 400 и сообщение с описанием неправильно переданных значений.

Настройка канала уведомлений

Основной сценарий

Шаг 1. Пользователь заполняет настройки канала уведомлений в UI и отправляет запрос к серверу.

Шаг 2. Сервер проверяет запрос на соответствие формату. (ИС-1)

Шаг 3. Канал уведомлений сохраняется в базу данных.

Шаг 4. Сервер подтверждает успешность сохранения канала.

Исключительные сценарии

ИС-1: Запрос не соответствует протоколу

Сервер возвращает код ошибки 400 и сообщение с описанием неправильно переданных значений.

Группировка получателей

Основной сценарий

Шаг 1. Пользователь указывает набор получателей в UI и отправляет запрос к серверу.

Шаг 2. Сервер проверяет запрос на соответствие формату. (ИС-1)

Шаг 3. Связь пользователей с группой сохраняется в базу данных.

Шаг 4. Сервер подтверждает успешность сохранения группы.

Исключительные сценарии

ИС-1: Запрос не соответствует протоколу

Сервер возвращает код ошибки 400 и сообщение с описанием неправильно переданных значений.

Настройка шаблона уведомлений

Основной сценарий

Шаг 1. Пользователь заполняет настройки шаблона в UI и отправляет запрос к серверу.

Шаг 2. Сервер проверяет запрос на соответствие формату. (ИС-1)

Шаг 3. Шаблон и связь с ним каналов и групп получателей сохраняются в базу данных.

Шаг 4. Сервер подтверждает успешность сохранения шаблона.

Исключительные сценарии

ИС-1: Запрос не соответствует протоколу

Сервер возвращает код ошибки 400 и сообщение с описанием неправильно переданных значений.

Обнаружение событий и отправка уведомлений

Основной сценарий

Шаг 1. Alert Manager получает метрики от источника. (ИС-1)

Шаг 2. Метрики проверяются правилами отклонений и уведомлений.

Шаг 3. При успешной проверке создается событие на основании её результатов.

Шаг 4. Правилком уведомлений проверяются условия отправки уведомлений.

Шаг 5. При успешной проверке производится отправка уведомлений. (ИС-2)

Шаг 6. Получатель получает предназначенное ему уведомление.

Исключительные сценарии

ИС-1: Возникает ошибка при получении метрик от источника

Статус правила изменяется на соответствующий, ошибка сохраняется и доступна для просмотра в UI.

ИС-2: Возникает ошибка при отправке уведомлений

Статус отправки изменяется на соответствующий, ошибка сохраняется и доступна для просмотра в UI.

Описание функциональных характеристик компонента Business activity monitoring (BAMN)

Термины и определения

Раздел с терминами доступен в документе «Общее описание продукта Platform V Monitor (OPM)».

В таблице ниже приведен перечень терминов и определений касающихся только компонента BAMN продукта Platform V Monitor (OPM).

Термины	Определение
BAMN	Компонент Business activity monitoring (BAMN), входящий в состав продукта Platform V Monitor (OPM). Сервис, предназначенный для сбора событий бизнес-процессов
Дашборд	Интерактивная цифровая панель
Administrator	Компонент BAMN – предназначен для работы с подключениями (создание, удаление), получения информации о подключениях, фильтрах, сущностях Platform V Monitor Abyss, разрешениях.
Selector	Компонент BAMN – предназначен для получения аналитических данных из Platform V Monitor Abyss и информации по процессам.

Pull processing	Компонент BAMN – предназначен для сбора событий бизнес-процессов
Cloudevents	Спецификация для описания данных события

Назначение

Цель создания

Автоматическая постановка на бизнес-мониторинг процессов, исполняемых в Platform V Flow (BPM). Допускается возможность использования другого программного продукта, поддерживающего поставку событий в формате Cloudevents.

На стороне Platform V Flow (BPM) пользователем создаются бизнес процессы, к которым можно подключить функционал BAMN по мониторингу событий, создаваемых в рамках бизнес процессов.

Это руководство содержит **названия переменных, которые применимы для различных сред контейнеризации (для Kubernetes и Open Shift, который может быть использован опционально), указанных в Руководстве по установке.**

Основные функции(сущностные)

Функция	Потребитель функции
Подключение Бизнес-мониторинга к проекту	Abyss
Отключение Бизнес-мониторинга от проекта	Abyss
Сбор аналитических данных бизнес-процессов	Abyss
Передача аналитических данных	Indicator
Хранение и передача схемы бизнес-процесса	Indicator
Фильтрация аналитических данных бизнес-процессов	Indicator

Сценарии использования

Наименование	Главный успешный сценарий	Исключительный сценарий
1. Просмотр проектов и процессов Platform V Flow (BPM) пользовательского подключения	1. Сервис авторизации предоставляет информацию о доступах по идентификатору пользователя; 2. Пользователь переходит в UI BAMN;	Отсутствует

	<p>3. UI отправляет запрос в Selector;</p> <p>4. Selector отправляет запрос в БД BAM и возвращает ответ в UI;</p> <p>5. Пользователю отображаются проекты и процессы.</p>	
<p>2. Просмотр пользовательских проектов и процессов Platform V Flow (BPM) на дашбордах</p>	<p>1. Сервис авторизации предоставляет информацию о доступах по идентификатору пользователя;</p> <p>2. Пользователь переходит в Indicator;</p> <p>3. Indicator отправляет запрос в BAMN Administrator на получение подключений доступных пользователю (Administrator получает их из БД BAMN);</p> <p>4. Indicator обращается к BAMN Selector для получения проектов в выбранном подключении;</p> <p>5. Indicator обращается в Selector по REST API для получения событий по проекту/процессу;</p> <p>6. Selector отправляет REST запрос в Abyss для получения событий(метрик);</p> <p>7. Abyss возвращает ответ Selector;</p> <p>8. Selector возвращает ответ Indicator;</p> <p>9. Пользователю отображены доступные проекты и процессы на дашборде.</p>	Отсутствует
<p>3. Создание пользовательского подключения</p>	<p>1. Platform V Monitor.Indicator (INDA):</p> <p>1.1. Проверяет токен пользователя;</p> <p>1.2. Через REST API отправляет событие для Administrator о необходимости подключения BAMN;</p> <p>2. BAMN Administrator:</p> <p>2.1. Получает от UI запрос на создание подключения;</p>	<p>1. BAMN БД не сохраняет конфигурацию;</p> <p>2. Пользователю выдается ошибка создания конфигурации</p>

	<p>2.2. Administrator формирует и отправляет запрос в Abyss на создание топика, задачи индексации по заданным параметрам;</p> <p>2.3. Abyss возвращает успешный ответ;</p> <p>2.4. Пользователю в UI отображается информация об успешности операции.</p>	
<p>4. Блокировка событий по процессам проектов Platform V Flow (BPM) пользовательского подключения</p>	<p>1. Сервис авторизации предоставляет информацию о доступах по идентификатору пользователя (у пользователя должно быть соответствующее разрешение (permission));</p> <p>2. Пользователь переходит в UI BAMN во вкладку "Управление сбором событий";</p> <p>3. UI отправляет запрос в Selector и подгружает данные о проектах и процессах;</p> <p>4. UI отправляет запрос в BAMN Administrator на получение существующих фильтров (создание, хранение фильтров происходит в БД BAMN);</p> <p>5. Пользователь "выключает" процесс;</p> <p>6. BAMN Administrator направляет запрос в БД BAMN на отключение сбора событий по процессу;</p> <p>7. БД BAMN возвращает ответ в BAMN Administrator об успешности сохранения фильтра;</p> <p>8. В отключенном процессе останавливается сбор событий;</p> <p>9. В UI BAMN отображено уведомление об успешности операции, отключенные процессы визуально выделены.</p>	Отсутствует
<p>5. Инициализация подключения для процессов с неустановленным владельцем</p>	<p>1. Сервис авторизации предоставляет информацию о доступах по идентификатору</p>	Отсутствует

	<p>пользователя (у пользователя должно быть соответствующее разрешение (permission));</p> <p>2. UI отправляет запрос в VAMN Administrator на получение подключений доступных пользователю (Administrator получает их из БД VAMN);</p> <p>3. Пользователь устанавливает подключение с неустановленным владельцем (по умолчанию);</p> <p>4. VAMN Administrator отправляет запрос в БД VAMN для установки подключения для данных с неустановленным владельцем;</p> <p>5. БД VAMN возвращает успешный ответ;</p> <p>6. VAMN Administrator возвращает успешный ответ;</p> <p>7. В UI VAMN появляется соответствующее уведомление.</p>	
6. Просмотр проектов Platform V Flow (BPM) и процессов с неустановленным владельцем	<p>1. Сервис авторизации предоставляет информацию о доступах по идентификатору пользователя;</p> <p>2. Пользователь переходит в UI VAMN;</p> <p>3. UI отправляет запрос в Selector;</p> <p>4. Selector отправляет запрос в БД VAMN и возвращает ответ в UI;</p> <p>5. Пользователю отображаются проекты и процессы.</p>	Отсутствует
7. Просмотр проектов и процессов Platform V Flow (BPM) с неустановленным владельцем на дашбордах	<p>1. Сервис авторизации предоставляет информацию о доступах по идентификатору пользователя;</p> <p>2. Пользователь переходит в Indicator;</p> <p>3. Indicator отправляет запрос в VAMN Administrator на получение подключений доступных пользователю</p>	Отсутствует

	<p>(BAMN Administrator получает их из БД BAMN);</p> <p>4. Indicator обращается к BAMN Selector для получения проектов в выбранном подключении;</p> <p>5. Indicator обращается в BAMN Selector по REST API для получения событий по проекту/процессу;</p> <p>6. BAMN Selector отправляет REST запрос в Abyss для получения событий(метрики);</p> <p>7. Abyss возвращает ответ BAMN Selector;</p> <p>8. BAMN Selector возвращает ответ Indicator;</p> <p>9. Пользователю отображены доступные проекты и процессы на дашборде.</p>	
<p>8. Блокировка событий по процессам проектов Platform V Flow (BPM) с неустановленным владельцем</p>	<p>1. Сервис авторизации предоставляет информацию о доступах по идентификатору пользователя (у пользователя должно быть соответствующее разрешение (permission));</p> <p>2. Пользователь переходит в UI BAMN во вкладку "Управление сбором событий";</p> <p>3. UI отправляет запрос в Selector и подгружает данные о проектах и процессах;</p> <p>4. UI отправляет запрос в Administrator на получение существующих фильтров (создание, хранение фильтров происходит в БД BAMN);</p> <p>5. Пользователь "выключает" процесс;</p> <p>6. Administrator направляет REST запрос в БД BAMN на отключение сбора событий по процессу;</p> <p>7. БД BAMN возвращает ответ в Administrator об успешности сохранения фильтра;</p>	<p>Отсутствует</p>

	8. В отключенном процессе останавливается сбор событий; 9. В UI BAMN отображено уведомление об успешности операции, отключенные процессы визуально выделены.	
--	---	--

Описание функциональных характеристик компонента Telemetry Collector (Единый коллектор телеметрии) (COTE)

Описание продукта

Единый коллектор телеметрии – это функциональная подсистема продукта Platform V Monitor (OPM), спроектированная и разработанная с целью сбора телеметрической информации – логов, трассировок, метрик и др. Сервис также реализует различные алгоритмы предобработки данных.

Независимо от способов сбора данных, предобработки и хранения, к процессу применяются стандартные механизмы контроля, управления, мультитенантности и самомониторинга.

Термины и определения

Общие термины и определения, используемые в данном документе, представлены в общей документации продукта Platform V Monitor (OPM).

Ниже представлены специальные термины, относящиеся к компоненту Telemetry Collector (Единый коллектор телеметрии) (COTE) продукта Platform V Monitor (OPM).

Термин/Аббревиатура	Определение
Pipeline сбора телеметрических данных	Конвейер обработки информации. Представлен набором компонентов и связей между ними
Pull-модель сбора	Модель сбора данных, где инициатором выступает серверная сторона, а не клиентская. Также модель называют активной

Термин/Аббревиатура	Определение
Push-модель сбора	Модель сбора данных, где клиент отправляет запросы на сервер, а сервер публикует API для их обработки. Также модель называют пассивной
Компонент pipeline	<p>Элемент обработки данных, который характеризуется входящим и исходящим типами данных и способом их обработки. Компоненты разделены на три вида:</p> <ol style="list-style-type: none"> 1. input - это компоненты, связанные с источниками данных; 2. processor - это компоненты, которые обрабатывают данные, например, обогащают, фильтруют, преобразуют; 3. output - это компоненты, которые сохраняют данные в какое-либо хранилище

Назначение

Цель создания

Необходимость разработать компонент в составе продукта Platform V Monitor (OPM), позволяющий решать задачи централизованного управления процессом сбора данных телеметрии за счет использования в своем механизме принципа конфигурируемого конвейера, реализующего унифицированный подход к процессу сбора телеметрических данных.

Основные функции

Название функции	Потребитель функции	Аргументы функции	Результат
Сбор данных телеметрии	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Input	Завершен процесс сбора телеметрической информации разного формата из разных источников

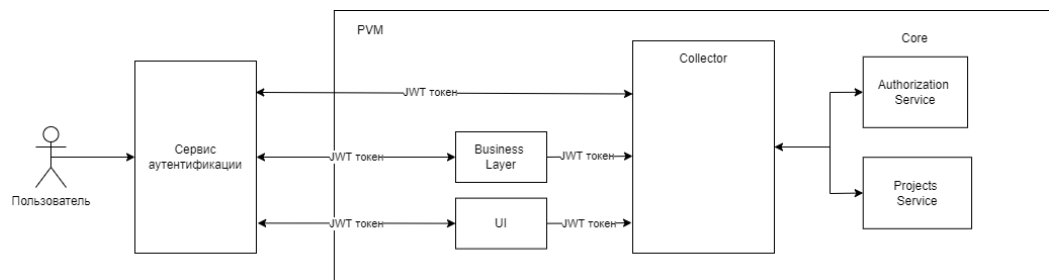
Название функции	Потребитель функции	Аргументы функции	Результат
Processing	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Processor	Завершен процесс фильтрации, обогащения, квотирования и подобные. Также, вероятно, процесс стандартизации
Сохранение данных	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Output	Завершена передача данных в какое-либо хранилище (или несколько хранилищ)
Настройка pipeline: Настройка справочников	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Конфигурация справочника	Pipeline сконфигурирован
Настройка pipeline: Настройка элементов сбора	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Конфигурация элементов сбора	Pipeline сконфигурирован
Настройка pipeline: Настройка элементов обработки	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента	Конфигурация элементов обработки	Pipeline сконфигурирован

Название функции	Потребитель функции	Аргументы функции	Результат
	(TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)		
Настройка pipeline: Настройка элементов сохранения данных телеметрии	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Конфигурация элементов сохранения	Pipeline сконфигурирован
Управление pipeline: запуск	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Status	Pipeline запущен
Управление pipeline: остановка	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Status	Pipeline остановлен
Управление pipeline: блокировка	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Конфигурация блокировки	Pipeline заблокирован

Название функции	Потребитель функции	Аргументы функции	Результат
Управление pipeline: копирование	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Конфигурация pipeline	Pipeline скопирован
Управление pipeline: редактирование	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Конфигурация pipeline	Pipeline изменен
Управление pipeline: архивирование	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Status	Pipeline архивирован
Управление pipeline: перевод в статус "Черновик"	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента (TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)	Status	Pipeline переведен в статус "Черновик"
Мониторинг работы pipeline	Администратор PVM (PVM_ADMIN_PERMISSION) / Администратор Клиента (TC_PIPELINE_ADMIN) / Оператор Клиента	Идентификатор pipeline	Получена информация о pipeline в проекте, согласно правам

Название функции	Потребитель функции	Аргументы функции	Результат
	(TC_PIPELINE_VIEW / TC_PIPELINE_PUSH)		доступа пользователя

Сценарии использования



Аутентификация

Пользователи:

1. Администратор Platform V Monitor (OPM);
2. Администратор Клиента;
3. Оператор Клиента.

Пост-условия:

1. Успешная аутентификация пользователя.

Действие:

1. Аутентификация пользователя выполняется с помощью OIDC-провайдера. Реализациями OIDC-провайдера может быть компонент IAM Proxy (AUTH) в составе продукта Platform V IAM SE (IAM).

Авторизация

Пользователи:

1. Администратор Platform V Monitor (OPM);
2. Администратор Клиента;
3. Оператор Клиента.

Пост-условия:

1. Успешная авторизация пользователя.

Действие:

1. После успешной аутентификации выполняется процесс авторизации пользователя с помощью модуля авторизации Abyss.

Настройка pipeline сбора телеметрических данных

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Пользователь успешно авторизован в системе;
2. Существует проект с достаточной квотой для создания нового pipeline.

Пост-условия:

1. Pipeline успешно настроен.

Основной сценарий:

1. Создать квоту - `POST /project/pes1/quota;`
2. Создать pipeline - `POST {{ _url }}/project/pes1/pipeline;`
3. Создать пароль - `POST /project/pes1/configuration/identifiable-entity/password;`
4. Создать input - `POST /pipeline/{pipelineId}/input;`
5. Создать processor - `POST /pipeline/{pipelineId}/processor;`
6. Создать output - `POST /pipeline/{pipelineId}/output.`

Установка квоты на проект

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Пользователь успешно авторизован в системе.

Пост-условия:

1. Квота установлена.

Основной сценарий:

1. Создать проект посредством создания pipeline;
2. Отправить POST-запрос вида `/project/{projectName}/quota` с телом `{ "limitTrafficPerMin": 100 }`.

Альтернативный сценарий - Попытка повторно создать квоту на проект:

1. Отправить POST-запрос вида `/project/{projectName}/quota` с телом `{ "limitTrafficPerMin": 100 }`;
2. В ответе вернулся код 400 и ошибка `"400 Quota already exists for project collect"`

Изменение квоты на проекте

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Пользователь успешно авторизован в системе.

Пост-условия:

1. Квота изменена.

Основной сценарий:

1. Создать проект посредством создания pipeline;
2. Отправить POST-запрос вида `/project/{projectName}/quota` с телом `{ "limitTrafficPerMin": 100 }`;
3. Отправить POST-запрос вида `/quota/{quotaId}` с телом `{ "limitTrafficPerMin": 100 }`.

Альтернативный сценарий – Превышение суммы квот:

1. Администратор Platform V Monitor (OPM) изменяет значение квоты конкретного проекта;
2. Для проекта указывается или изменяется значение квоты так, что сумма квот всех активных pipeline проекта БОЛЬШЕ нового значения квоты проекта;
3. Пользователь сохраняет изменение;
4. Пользователь получает сообщение «Указанное значение квоты не может быть установлено, т.к. будет превышена сумма квот активных pipeline проекта»;
5. Значение квоты не изменяется.

Активация pipeline

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Пользователь успешно авторизован в системе;
2. Статус pipeline = DRAFT, INACTIVE, FAILED;
3. Наличие input-компонента pipeline;
4. Наличие output-компонента pipeline;
5. Значение установленной квоты должно быть больше нуля;

6. Сумма квот всех pipeline проекта, включая данный pipeline, но за исключением архивных pipeline, не должна превышать квоту на проект;
7. Во всей цепочке компонентов pipeline тип исходящих из компонента данных должен быть равен типу входящих данных в следующий компонент.

Пост-условия:

1. Pipeline активирован.

Основной сценарий:

1. Отправить PUT-запрос вида `{{ _url }} / pipeline / {pipelineId} / status ? status = ACTIVE ;`
2. В ответе возвращается 200 ОК.

Альтернативный сценарий №1 – Отсутствует input-компонент:

1. Отправить PUT-запрос вида `{{ _url }} / pipeline / {pipelineId} / status ? status = ACTIVE ;`
2. Пользователь получает сообщение «Pipeline не может быть активирован, т.к. отсутствует input компонент».

Альтернативный сценарий №2 – Отсутствует output-компонент:

1. Отправить PUT-запрос вида `{{ _url }} / pipeline / {pipelineId} / status ? status = ACTIVE ;`
2. Пользователь получает сообщение «Pipeline не может быть активирован, т.к. отсутствует output компонент».

Альтернативный сценарий №3 – Значение установленной квоты на pipeline:

1. Отправить PUT-запрос вида `{{ _url }} / pipeline / {pipelineId} / status ? status = ACTIVE`
 - 1.1. Пользователь получает сообщение «Pipeline не может быть активирован, т.к. его квота равна нулю».
2. Отправить PUT-запрос вида `{{ _url }} / pipeline / {pipelineId} / status ? status = ACTIVE`
 - 2.1. Пользователь получает сообщение «Pipeline не может быть активирован, т.к. будет превышена квота на проект».

Альтернативный сценарий №4 – Соответствие типов исходящих и входящих данных из/в компоненты:

1. Отправить PUT-запрос вида `{{ _url }} / pipeline / {pipelineId} / status ? status = ACTIVE ;`
2. Пользователь получает сообщение «Pipeline не может быть активирован, т.к. обнаружено несоответствие типов исходящих и входящих данных у смежных компонентов pipeline».

Остановка pipeline

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Пользователь успешно авторизован в системе;
2. Создан pipeline;
3. Статус pipeline = ACTIVE.

Основной сценарий:

1. Отправить PUT-запрос вида `{{ _url }}`
`}}/pipeline/{pipelineId}/status?status=INACTIVE`.

Перевод pipeline в статус "Черновик"

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Пользователь успешно авторизован в системе;
2. Создан pipeline;
3. Статус pipeline = FAILED, INACTIVE, ARCHIVE.

Основной сценарий:

1. Отправить PUT-запрос вида `{{ _url }}`
`}}/pipeline/{pipelineId}/status?status=DRAFT`.

Архивация pipeline

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Пользователь успешно авторизован в системе;
2. Создан pipeline;
3. Статус pipeline = DRAFT, FAILED, INACTIVE.

Пост-условия:

1. Pipeline архивирован.

Основной сценарий:

1. Отправить PUT-запрос вида `{{ _url }}`
`}}/pipeline/{pipelineId}/status?status=ARCHIVE`.

Создание блокировки pipeline

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Есть два настроенных pipeline PUSH и PULL Management App, pull-collector и push-collector работают;
2. Блокировки в БД отсутствуют.

Пост-условия:

1. Создана блокировка.

Основной сценарий:

1. Отправить POST-запрос вида `{{ _url }}/lock/global` с телом `{ "lockTime": "2022-05-25T07:16:10.842Z", "unlockTime": "2022-05-26T07:16:10.842Z" }` на создание глобальной блокировки, где обе даты в будущем и дата разблокировки больше даты блокировки.

Альтернативный сценарий:

1. Отправить POST-запрос вида `{{ _url }}/lock/global?intentionType=PULL` с телом `{ "lockTime": null, "unlockTime": "2022-05-26T07:16:10.842Z" }` на создание глобальной блокировки, где дата начала = null, дата окончания в будущем и тип PULL;
2. Отправить POST-запрос `{{ _url }}/lock/project/pes1?intentionType=PUSH` с телом `{ "lockTime": null, "unlockTime": null }` на создание блокировки PUSH pipeline по проекту, где дата начала = null и дата окончания = null;
3. Отправить POST-запрос `{{ _url }}/lock/project/{{ _project }}/pipeline/http-check-pipeline` с телом `{ "lockTime": "2022-05-26T07:16:10.842Z", "unlockTime": null }` на создание блокировки pipeline, где дата начала в будущем и дата окончания = null.

Копирование pipeline

Пользователи:

1. Администратор Platform V Monitor (OPM).

Предусловия:

1. Пользователь успешно авторизован в системе;
2. Создан pipeline.

Пост-условия:

1. Pipeline скопирован.

Основной сценарий:

1. Отправить POST-запрос вида `/pipeline/{pipelineId}/copy`;
2. Новый pipeline полностью повторяет копируемый (все компоненты с их конфигурациями), за исключением **наименования** (`COPY_<Наименование копируемого pipeline>`) и **квоты** (равна 0).

Просмотр информации о проекте

Пользователи:

1. Администратор Platform V Monitor (OPM);
2. Администратор Клиента;
3. Оператор Клиента.

Предусловия:

1. Пользователь успешно авторизован в системе;
2. У авторизованного пользователя есть доступ ко всем проектам/определенному проекту.

Основной сценарий:

1. Пользователь получает информацию о проекте;
2. Пользователь получает информацию о pipeline в проекте;
3. Пользователь получает информацию о квоте на проект;
4. Пользователь получает информацию о квотах на pipeline проектов;
5. Пользователь получает информацию о значениях счетчиков квот на проекты;
6. Пользователь получает информацию о значениях счетчиков квот на pipeline проектов;
7. Пользователь получает информацию о блокировках pipeline.

Описание функциональных характеристик компонента Indicator (INDA)

Описание продукта

Термины и определения

Термин/Аббревиатура	Определение
БД	База данных
Георезервирование	Территориальное и географическое разнесение важнейших узлов системы

Метрика	Измерения использования ресурсов или отклонения, которые можно собирать в аппаратной или программной среде
UI	User interface, пользовательский интерфейс
Dashboards (Дашборд)	Набор из одной или нескольких интерактивных панелей, для визуализации значений метрик
Панель / Panel	Представляет из себя одну визуализацию (например, в виде графика). Панель является основным строительным блоком визуализации. Каждая панель имеет редактор запросов, специфичный для источника данных, выбранного на панели.
Datasources	Настраиваемый источник данных
Pod / Под	Экземпляр, модуль в платформе приложений-контейнеров
Indicator	Компонент Indicator (INDA) в составе продукта Platform V Monitor (PVM). Обеспечивающий визуализацию в продукте Platform V Monitor.

Назначение

Indicator — Сервис контроля качества работы сервисов, предназначенный для предоставления пользователю информации о состоянии сервисов платформы. Информация для пользователя формируется путем обработки значений метрик с последующей их визуализацией на различных типах аналитических панелей.

Детальное описание:

Сервис представляет собой веб-приложение, созданное специально для развертывания и эксплуатации в облачной инфраструктуре. Приложение позволяет объединить в «единое окно» контроль качества работы всех элементов экосистемы в соответствии с желаниями и возможностями пользователей. Гибкий пользовательский интерфейс позволяет производить настройку отображения данных в соответствии с текущими потребностями пользователя. Настройка производится в интуитивно-понятном графическом интерфейсе с расширением функциональности через плагины. Приложение позволяет автоматически проанализировать работу всех необходимых сервисов и выявить потенциальные угрозы сбоям на основании онлайн потока данных о состоянии систем (от простого алгоритма к сложным механизмам предиктивного детектирования отклонений)

Основные функции сервиса:

Функция	Потребитель функции	Аргумент функции	Результат
Отображение графиков значений метрик	Пользователь	Запрос к хранилищу данных по именам метрик	Возвращение числового ряда значений метрик, согласно условиям запроса
Отображение агрегированных значений метрик	Пользователь	Запрос к хранилищу данных по именам метрик	Возвращение агрегированных значений метрик, согласно условиям запроса
Отображение в табличной форме значений метрик	Пользователь	Запрос к хранилищу данных по именам метрик	Возвращение значений метрик, согласно условиям запроса
Отображение в табличной форме логов	Пользователь	Запрос к хранилищу данных по именам метрик	Возвращение логов, согласно условиям запроса

Сценарии использования

Отображение графиков значений метрик

1. Пользователь создает новую панель или использует существующую.
2. Пользователь использует существующий DataSource
3. Пользователь создает новый SQL запрос или использует существующий.
4. По результатам SQL запроса на панели строится график/графики

Отображение агрегированных значений метрики

5. Пользователь создает новую панель или использует существующую.
6. Пользователь использует существующий DataSource.
7. Пользователь создает новый SQL запрос или использует существующий.
8. По результатам SQL запроса на панели отображается агрегированное значение метрики.

Отображение в табличной форме значений метрик

1. Пользователь создает новую панель или использует существующую.
2. Пользователь использует существующий DataSource.
3. Пользователь создает новый SQL запрос или использует существующий.
4. По результатам SQL запроса на панели отображаются данные по метрике в табличной форме.

Отображение в табличной форме логов

1. Пользователь создает новую панель или использует существующую.
2. Пользователь использует существующий DataSource.
3. Пользователь создает новый SQL запрос или использует существующий.
4. По результатам SQL запроса на панели отображаются данные по логам в табличной форме.

Описание функциональных характеристик компонента Abyss (LGDB)

Назначение

Abyss — это центральный компонент продукта Platform V Monitor (OPM). Abyss принимает, обрабатывает, хранит данные и предоставляет к ним доступ посредством API.

Термины и определения

Общие термины и определения, используемые в данном документе, представлены в общей документации продукта Platform V Monitor (OPM) в документе «Общее описание продукта Platform V Monitor (OPM)».

Основные функции

№	Название функции	Потребитель функции	Аргументы функции	Результат
1	Создать проект	Администратор Abyss	Наименование проекта, параметры проекта	Создан проект
2	Создать источник данных	Администратор проектной области Abyss	Проект, наименование источника данных, характеристики источника данных	Создан новый источник данных

3	Создать поток записи данных в источник данных	Функциональная подсистема (далее - ФП)	Данные, источник данных	Данные записаны в источник данных
4	Создать поток записи данных в необходимый индекс	Администратор проектной области Abyss	Проект, источник данных, наименование индекса, характеристики индекса, схема индексации	Создан поток записи в необходимые индексы из источников данных
5	Запрос данных из индекса	Потребитель данных	Проект, индекс, запрос	Получены данные из индекса по соответствующему запросу

Сценарии использования

1. Создание проекта

Главный успешный сценарий:

1. Администратор Abyss отправляет запрос на создание нового проекта.
2. Администратор Abyss получает уведомление о том, что проект успешно создан.

Исключительный сценарий:

1. Администратор Abyss отправляет запрос на создание нового проекта.
2. Администратор Abyss получает уведомление о том, что во время создание проекта произошла ошибка.

2. Создание источника данных

Главный успешный сценарий:

1. Администратор проектной области Abyss отправляет запрос на создание нового источника данных.
2. Администратор проектной области Abyss получает уведомление о том, что новый источник данных успешно создан.

Исключительный сценарий:

1. Администратор проектной области Abyss отправляет запрос на создание нового источника данных.
2. Администратор проектной области Abyss получает уведомление о том, что произошла ошибка.

- 2.1. Недостаточно ресурсов внутри проекта.
- 2.2. Внутренняя ошибка Abyss

3. Запись данных в источник данных

Главный успешный сценарий:

1. ФП передает данные в источник данных.
2. Данные успешно записаны в источник данных.

Исключительный сценарий:

1. ФП передает данные в источник данных.
2. Данные не записаны по причине:
 - 2.1. Недостаточно прав на запись в источник данных;
 - 2.2. Неверная конфигурация клиента;
 - 2.3. Неверные сертификаты;
 - 2.4. Ошибка транспорта данных.

4. Создание полнотекстового индекса

Главный успешный сценарий:

1. Администратор проектной области Abyss отправляет запрос на создание нового полнотекстового индекса.
2. Администратор проектной области Abyss получает уведомление о том, что новый полнотекстовый индекс успешно создан.

Исключительный сценарий:

1. Администратор проектной области Abyss отправляет запрос на создание нового полнотекстового индекса.
2. Администратор проектной области Abyss получает уведомление о том, что произошла ошибка:
 - 2.1. Недостаточно ресурсов внутри проекта;
 - 2.2. Внутренняя ошибка Abyss.

5. Создание аналитического индекса

Главный успешный сценарий:

1. Администратор проектной области Abyss отправляет запрос на создание нового аналитического индекса.
2. Администратор проектной области Abyss получает уведомление о том, что новый аналитический индекс успешно создан.

Исключительный сценарий:

1. Администратор проектной области Abyss отправляет запрос на создание нового аналитического индекса.
2. Администратор проектной области Abyss получает уведомление о том, что произошла ошибка:
 - 2.1. Недостаточно ресурсов внутри проекта;
 - 2.2. Внутренняя ошибка Abyss.

6. Создание архивного индекса

Главный успешный сценарий:

1. Администратор проектной области Abyss отправляет запрос на создание нового архивного индекса.
2. Администратор проектной области Abyss получает уведомление о том, что новый архивный индекс успешно создан.

Исключительный сценарий:

1. Администратор проектной области Abyss отправляет запрос на создание нового архивного индекса.
2. Администратор проектной области Abyss получает уведомление о том, что произошла ошибка:
 - 2.1. Недостаточно ресурсов внутри проекта;
 - 2.2. Внутренняя ошибка Abyss.

7. Чтение данных из полнотекстового индекса

Главный успешный сценарий:

1. Потребитель данных обращается с запросом к полнотекстовому индексу.
2. Потребителю данных возвращаются данные, соответствующие запросу.

Исключительный сценарий:

1. Потребитель данных обращается с запросом к полнотекстовому индексу.
2. Потребителю данных возвращается ошибка:
 - 2.1. Неверный синтаксис запроса;
 - 2.2. Недостаточно прав для доступа к указанному индексу;
 - 2.3. Внутренняя ошибка Abyss.

8. Чтение данных из аналитического индекса

Главный успешный сценарий:

1. Потребитель данных обращается с запросом к аналитическому индексу.
2. Потребителю данных возвращаются данные, соответствующие запросу.

Исключительный сценарий:

1. Потребитель данных обращается с запросом к аналитическому индексу.
2. Потребителю данных возвращается ошибка:
 - 2.1. Неверный синтаксис запроса;
 - 2.2. Недостаточно прав для доступа к указанному индексу;
 - 2.3. Внутренняя ошибка Abyss.

9. Чтение данных из архивного индекса

Главный успешный сценарий:

1. Потребитель данных обращается с запросом к архивному индексу.
2. Потребителю данных возвращаются данные, соответствующие запросу.

Исключительный сценарий:

1. Потребитель данных обращается с запросом к архивному индексу.
2. Потребителю данных возвращается ошибка:
 - 2.1. Неверный синтаксис запроса;
 - 2.2. Недостаточно прав для доступа к указанному индексу;
 - 2.3. Внутренняя ошибка Abyss.

Описание функциональных характеристик компонента Журналирование (LOGA)

Термины и определения

Общие термины и определения, используемые в данном документе, представлены в общей документации продукта Platform V Monitor (OPM) в документе «Общее описание продукта Platform V Monitor (OPM)».

Ниже приведены специальные термины для компонента Журналирование (LOGA) продукта Platform V Monitor (OPM).

Термин	Определение
Трейс	Цепочка подсистем, которые обработали запрос
Спан	Звено трейса
ФП	Функциональная подсистема Platform V.

Описание продукта

Назначение

Цель создания

Сервис Журналирование предназначен для сохранения событий журналирования, предоставления интерфейса для получения этих событий.

Основные функции

Название функции	Потребитель функции	Аргументы функции	Результат
Записать события журналирования в журнал	Функциональная подсистема (далее - ФП)	События журналирования	Событие записано в журнал
Получить события журналирования	Пользователи системы (специалисты отдела сопровождения, разработчики прикладных систем и др.)	Параметры фильтрации событий	Получены события журналирования, отвечающие условиям поиска
Трейсинг операции	Пользователи системы (специалисты отдела сопровождения, разработчики прикладных систем и др.)	Трейсы / спаны	Получена последовательность событий операции с продолжительностью выполнения

Получение событий журналирования из архива	Пользователи системы (специалисты отдела сопровождения, разработчики прикладных систем и др.)	Параметры фильтрации архивных событий журналирования	Получены события журналирования из архива, отвечающие условиям поиска
--	---	--	---

Сценарии использования

1. Запись события журналирования.

Главный успешный сценарий:

- 1.1. ФП передает событие журналирования в сервис Журналирование.
- 1.2. Сервис Журналирование обогащает события журналирования и направляет на хранение в единое хранилище.

2. Запись трейсов.

Главный успешный сценарий:

- 2.1. В точках интеграционных взаимодействий передают данные о трейсах/ спанах в сервис Журналирование.
- 2.2. Сервис Журналирование направляет трейсы / спаны на хранение в единое хранилище.

3. Просмотр записей журнала.

Главный успешный сценарий:

- 3.1. Пользователь заполняет параметры фильтрации событий журнала.
- 3.2. Сервис Журналирование возвращает пользователю события журналирования, отвечающие параметрам фильтрации.

Исключительный сценарий:

- 3.1а. Сервис Журналирование возвращает ошибку, если не указаны обязательные параметры поиска данных.
- 3.1б. Сервис журналирование возвращает ошибку, если указанный период получения данных превышает 24 часа.

4. Выгрузка данных.

Главный успешный сценарий:

- 4.1. Пользователь запрашивает у Сервиса Журналирования выгрузку данных в файл.
- 4.2. Сервис Журналирование формирует файл и отображает для просмотра пользователю.

Исключительный сценарий:

4.1a. Сервис Журналирование возвращает ошибку, если нет событий журналирования для формирования файла.

5. **Просмотр трейсов.**

Главный успешный сценарий:

5.1. Пользователь заполняет фильтры для поиска трейсов / спанов.

5.2. Сервис Журналирование отправляет запрос в хранилище для выполнения поиска трейсов / спанов.

5.3. Хранилище предоставляет ответ на запрос с результатами поиска.

5.4. Сервис Журналирование отображает результаты поиска в пользовательском интерфейсе.

Исключительный сценарий:

5.2a. Сервис Журналирование возвращает ошибку, если не указаны обязательные параметры поиска трейсов / спанов.

6. **Переход из события журналирования в трейс.**

Главный успешный сценарий:

6.1. Пользователь инициирует действие по переходу из события журналирования в трейс.

6.2. Сервис Журналирование отправляет запрос в хранилище данных на получение связанного трейса с событием журналирования.

6.3. Хранилище предоставляет ответ на запрос с деревом трейсов связанных с событием журналирования

6.4. Сервис Журналирование отображает результат в пользовательском интерфейсе.

Альтернативный сценарий:

6.1. Пользователь инициирует действие по переходу из события журналирования в трейс.

6.2. Сервис Журналирование отправляет запрос в хранилище данных на получение связанного трейса с событием журналирования.

6.3. Хранилище предоставляет ответ с пустым значением.

6.4. Сервис Журналирование выводит сообщение в пользовательском интерфейсе об отсутствии связанного трейса с событием журналирования.

7. **Поиск архивных событий журналирования.**

Главный успешный сценарий:

7.1. Пользователь заполняет фильтры на поиск архивных событий журналирования.

7.2. Сервис Журналирование отправляет запрос в хранилище для выполнения поиска архивных событий журналирования.

7.3. Хранилище создает запрос на поиск и запускает процесс поиска архивных логов согласно фильтрам.

- 7.4. Сервис Журналирование получает идентификатор запроса на поиск архивных событий журналирования.
- 7.5. Пользователь запрашивает статус выполнения запроса.
- 7.6. Сервис Журналирование запрашивает из хранилища статусы и обновляет их в пользовательском интерфейсе.
- 7.7. Пользователь запрашивает результат поиска архивных событий журналирования для выполненного запроса.
- 7.8. Сервис Журналирование направляет запрос в хранилище на получение результатов поиска.
- 7.9. Хранилище предоставляет ответ с результатом поиска.
- 7.10. Сервис Журналирование предоставляет результат в виде файла для скачивания.
- 7.11. Пользователь загружает файл.

Описание функциональных характеристик компонента Объединенный мониторинг Unimon (MONA)

Описание продукта

Термины и определения

Общие термины и определения, используемые в данном документе, представлены в общей документации продукта Platform V Monitor (OPM) в документе «Общее описание продукта Platform V Monitor (OPM)».

Назначение

Цель создания

Компонент Объединенный мониторинг Unimon (MONA) продукта Platform V Monitor (OPM) (далее — Unimon) предназначен для сбора данных о производительности, доступности и работоспособности прикладных приложений и инфраструктуры.

Основные функции (сущностные)

№	Название функции	Потребитель	Аргументы	Результат
---	------------------	-------------	-----------	-----------

1	Сбор метрик (pull)	Сотрудник сопровождения. Разработчики. Сотрудники, отвечающие за прикладное приложение.	Выставленные метрики в формате Prometheus, идентификатор потребителя сервиса	Собранные и обогащенные дополнительными полями метрики готовые к отправке
2	Сбор метрик (push)	Сотрудник сопровождения. Разработчики. Сотрудники, отвечающие за прикладное приложение.	Метрики в формате Prometheus или json, параметры определяющие откуда отправлены метрики. Идентификатор потребителя сервиса.	Собранные и обогащенные дополнительными полями метрики готовые к отправке
3	Валидация	Сотрудник сопровождения. Сотрудники, отвечающие за прикладное приложение.	Метрики, полученные на этапе сбора	Отброшены метрики, которые не могут быть сохранены, как невалидные - противоречивые данные в полях, отсутствие данных в полях, отсутствие полей
4	Фильтрация	Сотрудник сопровождения. Сотрудники, отвечающие за прикладное приложение.	Метрики, полученные на этапе сбора. Идентификатор потребителя сервиса.	Отброшены метрики, которые не проходят заданные условия передачи данных в хранилище

5	Сохранение данных в хранилище	Сотрудник сопровождения Разработчики. Сотрудники, отвечающие за прикладное приложение.	Метрики, полученные на этапе сбора, прошедшие этапы валидации и фильтрации. Идентификатор потребителя сервиса.	Метрики переданы в систему хранения
6	Диагностика сбора	Сотрудник сопровождения	Идентификатор потребителя сервиса	Информация о доступности и работоспособности системы сбора
7	Получение метаданных публикуемых метрик	Сотрудник сопровождения Разработчики. Сотрудники, отвечающие за прикладное приложение.	Идентификатор потребителя сервиса	Доступна информация о метаданных (наименование, точка сбора) метрик, которые возможно сохранять
8	Управление сбором	Сотрудник сопровождения	Идентификатор потребителя сервиса	Доступна информация о количестве собранных метрик за период, расчет остатка квот сбора
9	Получение метрик из хранилища	Разработчики. Сотрудники, отвечающие за прикладное приложение.	Идентификатор потребителя сервиса. Фильтры ограничивающие запрос.	Метрики в формате json

Сценарии использования

№	Наименование	Главный успешный сценарий	Исключительный сценарий
1	Сохранение метрик в хранилище (pull)	<p>1. Приложение выставляет метрики в формате Prometheus.</p> <p>2. Unimon-agent собирает метрики.</p> <p>3. Unimon-sender отправляет метрики в хранилище.</p>	<p>1. Приложение выставляет метрики в формате Prometheus.</p> <p>2. Unimon-agent собирает метрики.</p> <p>3. Unimon-sender не может отправить метрики т.к. не задано место хранения метрики.</p>
2	Сохранение метрик в хранилище (push)	<p>1. Из приложения производится вызов API Unimon-sender.</p> <p>2. Unimon-sender отправляет метрики в хранилище.</p>	<p>1. Из приложения производится вызов API Unimon-sender.</p> <p>2. Unimon-sender не отправляет метрики в хранилище т.к. не задано место хранения метрики.</p>

3	Получение метрик из хранилища	<p>1. Пользователь в средстве просмотра или в приложении делает запрос к серверной части Unimon для получения метрик.</p> <p>2. Метрики отправляются в соответствии с фильтром и доступом.</p>	<p>1. Пользователь в средстве просмотра или в приложении делает запрос к серверной части Unimon для получения метрик.</p> <p>2. Метрики не отправляются т.к. нет доступа на метрики, указанные в запросе.</p>
4	Получение статистики сбора	<p>1. Пользователь в АРМ Unimon может просмотреть метаданные метрик своих приложений, точки сбора и количество собираемых метрик.</p>	-
5	Настройка фильтров	<p>1. Администратор в АРМ Unimon указывает какие метрики необходимо сохранять и период действия фильтра.</p> <p>2. Unimon-sender сохраняет метрики в соответствии с фильтрами.</p>	-

6	Диагностика	<p>1. Администратор в АРМ Unimon просматривает весь путь сбора метрик приложения.</p> <p>2. В случае проблемы получает конкретное место ее возникновения.</p> <p>3. Администратор устраняет проблему.</p>	-
7	Управление сбором	<p>1. Администратор в АРМ Unimon приостанавливает или запускает сбор, устанавливает периодичность опроса в случае pull сбора метрик.</p> <p>2. Просматривает объем квоты сбора, которая задана по проекту и получает значения по ее утилизации.</p>	-