



**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**
(МИНЦИФРЫ РОССИИ)

ПРИКАЗ

01.11.2023

935

Москва

**Об утверждении требований к вычислительной мощности,
используемой провайдером хостинга,
для проведения уполномоченными государственными органами,
осуществляющими оперативно-разыскную деятельность или обеспечение
безопасности Российской Федерации, в случаях, установленных
федеральными законами, мероприятий в целях реализации возложенных
на них задач**

В соответствии с частью 3 статьи 10²⁻¹ Федерального закона от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации», пунктом 1 Положения о Министерстве цифрового развития, связи и массовых коммуникаций Российской Федерации, утвержденного постановлением Правительства Российской Федерации от 2 июня 2008 г. № 418,

ПРИКАЗЫВАЮ:

Утвердить прилагаемые требования к вычислительной мощности, используемой провайдером хостинга, для проведения уполномоченными государственными органами, осуществляющими оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации, в случаях, установленных федеральными законами, мероприятий в целях реализации возложенных на них задач.

Министр

М.И. Шадаев

УТВЕРЖДЕНЫ
приказом Министерства
цифрового развития, связи
и массовых коммуникаций
Российской Федерации
от 22.11.2018 № 235

ТРЕБОВАНИЯ
к вычислительной мощности, используемой провайдером хостинга,
для проведения уполномоченными государственными органами,
осуществляющими оперативно-разыскную деятельность или обеспечение
безопасности Российской Федерации, в случаях, установленных
федеральными законами, мероприятий в целях реализации возложенных
на них задач

I. Общие положения

1. Требования к вычислительной мощности, используемой провайдером хостинга, для обеспечения выполнения установленных действий при проведении оперативно-разыскных мероприятий (далее соответственно – требования, ОРМ), разработаны в целях обеспечения устойчивого, безопасного и целостного функционирования на территории Российской Федерации информационно-телекоммуникационной сети «Интернет», а также создания условий для выполнения уполномоченными государственными органами, осуществляющими оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации (далее – уполномоченные государственные органы), в случаях, установленных федеральными законами, возложенных на них задач с использованием технических средств ОРМ (далее – ТС ОРМ).

2. ТС ОРМ должны быть частью вычислительной мощности, используемой провайдером хостинга.

3. Состав и построение ТС ОРМ определяются с учетом особенностей построения информационных систем провайдеров хостинга (далее – ИС), перечнем предоставляемых информационных услуг и коммуникационных интернет-сервисов.

4. При построении ТС ОРМ провайдером хостинга должен быть использован один из следующих вариантов:

- 1) отдельный аппаратно-программный комплекс;
- 2) отдельный программный модуль в ИС;
- 3) комбинированный вариант, предусматривающий совместное использование элементов в соответствии с подпунктами 1 и 2 настоящего пункта.

5. Для каждого варианта построения ТС ОРМ провайдером хостинга должны реализовываться согласованные с уполномоченным государственным органом требования по информационной безопасности и защите от

несанкционированного доступа к информации, связанной с проведением ОРМ, в соответствии с подпунктами 8 и 9 пункта 18 настоящих требований.

II. Функциональные требования к ТС ОРМ

6. С использованием ТС ОРМ провайдера хостинга должны осуществляться поиск, обработка и передача на пульт управления уполномоченного государственного органа (далее – ПУ) информации, хранящейся в ИС и обрабатываемой с использованием вычислительной мощности провайдера хостинга, по запросу уполномоченного государственного органа или в автоматическом режиме.

7. С использованием ТС ОРМ провайдера хостинга должны обеспечиваться поиск согласованной с уполномоченным государственным органом, определяемым в порядке, установленном Правительством Российской Федерации в соответствии с частью 3 статьи 10²⁻¹ Федерального закона от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации», информации, хранящейся в ИС и обрабатываемой с использованием ТС ОРМ провайдера хостинга согласно приложению № 1 к настоящим требованиям, а также поддерживать согласованную с уполномоченным государственным органом схему ее предоставления.

8. ТС ОРМ должно осуществлять взаимодействие по единому каналу передачи данных с ПУ.

9. Суммарная пропускная способность канала передачи данных между ТС ОРМ и ПУ должна соответствовать данным, приведенным в таблице № 1.

Таблица № 1.

№ п/п	Среднесуточный объем новых данных, поступающих в ИС, Гбайт	Суммарная пропускная способность каналов передачи данных между ТС ОРМ и ПУ, не менее Мбит/с
1	<1	10
2	1 - 10	50
3	10 - 100	100
4	100 - 500	500
5	500 - 1000	800
6	1000 - 10000	1000
7	>10000	10000

10. Требования предъявляемые к ТС ОРМ в части обработки информации хранение, которой осуществляется в информационной системе провайдера хостинга приведены в приложении № 1 к настоящим требованиям.

11. Требования, предъявляемые к интерфейсу взаимодействия между ПУ и ТС ОРМ, приведены в приложении № 2 к настоящим требованиям.

12. Требования, предъявляемые к схеме представления информации, хранящейся в ИС (далее – схема данных), и выполнению запросов от ПУ приведены в приложении № 3 к настоящим требованиям.

13. Перечень типов данных являющихся стандартными для описания схемы данных (далее – базовые типы) приведен в приложении № 4 к настоящим требованиям.

14. Перечень типов данных являющихся служебными для описания схемы данных (далее – сервисные типы) приведен в приложении № 5 к настоящим требованиям.

15. Перечень входных объектов для задания параметров поиска приведен в приложении № 6 к настоящим требованиям.

16. Требования, предъявляемые к формату передачи данных на языке GraphQL по протоколу WebSocket, приведены в приложении № 7 к настоящим требованиям.

17. С использованием ТС ОРМ провайдера хостинга обеспечивается исполнение запросов о текущей конфигурации оборудования, системного и прикладного программного обеспечения (далее – ПО), а также их состояния согласно приложению № 8 к настоящим требованиям.

18. С использованием ТС ОРМ реализуются:

1) подключение к ПУ в соответствии техническими условиями, согласованными с уполномоченным государственным органом;

2) протокол взаимодействия ТС ОРМ с ПУ в соответствии с приложением № 2 к настоящим требованиям;

3) прием от ПУ поисковых запросов;

4) поиск информации, хранящейся в ИС, в соответствии с поступившими с ПУ поисковыми запросами. Для ТС ОРМ, осуществляющих поиск информации в ИС со среднесуточным объемом новых данных свыше 1 Гбайт, устанавливаются требования к поддерживаемым критериям поиска согласно пункту 21 приложения № 2 к настоящим требованиям;

5) передача от ТС ОРМ на ПУ данных в соответствии с поступившими с ПУ поисковыми запросами (в том числе постранично);

6) прием от ПУ критериев поиска и передачу на ПУ статистической, текстовой, мультимедийной, звуковой, графической и иной информации в исходном (декодированном) виде, хранящейся в ИС и отбираемой по критериям поиска, без дополнительной обработки (далее – неформатированные данные);

7) передача на ПУ схемы данных, в соответствии с поступившим с ПУ специальным запросом согласно приложению № 5 к настоящим требованиям;

8) защита от несанкционированного доступа со стороны производителей ТС ОРМ, неавторизованных пользователей, технического персонала, третьих лиц к хранящейся в ТС ОРМ информации и информации, непосредственно связанной с проведением ОРМ (информации, поступающей в ТС ОРМ с ПУ, и информации, подготовленной к передаче из ТС ОРМ в ПУ);

9) информирование ПУ о следующих попытках несанкционированного доступа к ТС ОРМ:

доступ к данным, связанным с проведением ОРМ, с использованием команд или сервисных программ;

резервное копирование данных, связанных с проведением ОРМ;

доступ к ТС ОРМ через интерфейсы, не предусмотренные для доступа к ним; вскрытие корпуса технических средств ОРМ (для варианта исполнения ТС ОРМ в виде отдельного аппаратно-программного комплекса и для комбинированного варианта исполнения ТС ОРМ);

10) контроль работоспособности и загруженности ТС ОРМ;

11) контроль за соблюдением предоставленных прав доступа к хранящейся в ТС ОРМ информации;

12) круглосуточный удаленный доступ со стороны операторов ПУ к хранящейся в ИС информации по протоколу взаимодействия ПУ и ТС ОРМ, описанному в приложении № 2 к настоящим требованиям;

13) ведение в автоматическом режиме системных журналов, содержащих информацию о работе ТС ОРМ, а также связанных с проведением ОРМ данных:

о сеансах связи с ПУ, а также о попытках установления таких сеансов;

о поисковых запросах;

об ответах на поисковые запросы;

о текущей конфигурации ТС ОРМ, системного и прикладного ПО;

об изменениях в конфигурации ТС ОРМ, системного и прикладного ПО;

о сбоях в ТС ОРМ, системном и прикладном ПО;

об изменениях схемы данных;

об обращении и доступе обслуживающего технического персонала к ТС ОРМ;

14) доступ уполномоченного технического персонала для обслуживания ТС ОРМ в соответствии с установленными правами доступа;

15) доступ технического персонала к системным файлам и ПО, в соответствии с правами, установленными парольной системой доступа;

16) сохранность и доступность для дальнейшего использования ранее накопленных данных при модернизации ТС ОРМ;

17) автоматическое определение способа выполнения поисковых запросов (в режиме реального времени или в отложенном режиме согласно пункту 20 приложения № 2 к настоящим требованиям), поступивших с ПУ, в соответствии с заданными временными параметрами;

18) временное хранение результатов выполнения поисковых запросов в отложенном режиме с учетом заданных временных параметров (согласно пункту 21 настоящих требований).

19. Функционирование ТС ОРМ не должно оказывать влияние на работоспособность ИС провайдера хостинга.

20. ТС ОРМ не должны иметь иных интерфейсов взаимодействия кроме интерфейсов взаимодействия с ИС и ПУ.

21. Максимальное время предварительной обработки информации с момента ее поступления в ТС ОРМ до момента, когда она становится доступной для выполнения поисковых запросов с ПУ, и максимальное время выполнения поисковых запросов определено в ходе внедрения ТС ОРМ в информационную систему провайдера хостинга. ТС ОРМ провайдеров хостинга должны соблюдать согласованные с уполномоченным государственным органом временные

параметры для определения способа выполнения поисковых запросов (в режиме реального времени или в отложенном режиме, пункт 20 приложения № 2 к настоящим требованиям) и максимального времени хранения результатов выполнения поисковых запросов в отложенном режиме должны.

22. Временные параметры, указанные в пункте 21 настоящих требований, могут корректироваться провайдером хостинга во время эксплуатации ТС ОРМ путем взаимодействия с уполномоченным государственным органом.

Приложение № 1
к требованиям к вычислительной мощности,
используемой провайдером хостинга, для
проведения уполномоченными
государственными органами, осуществляющими
оперативно-разыскную деятельность или
обеспечение безопасности Российской
Федерации, в случаях, установленных
федеральными законами, мероприятий в целях
реализации возложенных на них задач,
утвержденным приказом Министерства
цифрового развития, связи и массовых
коммуникаций Российской Федерации
от 1 ноября 2023 г. № 935

**Требования, предъявляемые к техническим средствам, обеспечивающим
выполнение установленных действий при проведении оперативно-разыскных
мероприятий, в части обработки информации, хранение которой
осуществляется в информационной системе провайдера хостинга**

1. С использованием ТС ОРМ провайдера хостинга должна обрабатываться информация, указанная в пунктах 2 – 9 настоящих требований.

2. Сведения о пользователях и регистрационных данных, в том числе регистрационных и иных данных, изменениях и дополнениях регистрационных данных, внесенных пользователем:

сведения о пользователе, включая идентификатор пользователя в ИС провайдера хостинга;

информация о сетевом адресе, с которого осуществлена регистрация пользователя, с указанием времени регистрации;

информация, которая должна вноситься в ИС провайдера хостинга при регистрации пользователя:

дата и время регистрации пользователя;

номер договора, дата и время заключения договора на обслуживание пользователя провайдером хостинга (при наличии такой информации);

информация, внесенная пользователем (при наличии):

фамилия, имя, отчество (при наличии);

псевдоним пользователя;

дата рождения;

адрес регистрации по месту жительства (пребывания);

реквизиты документа, удостоверяющего личность;

внесенные пользователем принадлежащие ему идентификаторы в других средствах электронного взаимодействия, в том числе наименование сервиса и идентификатор;

дата и время последнего обновления пользователем регистрационной информации;

дата и время прекращения регистрации пользователя в ИС провайдера хостинга;

дополнительная информация о пользователе, фиксируемая ИС провайдера хостинга.

3. Сведения о добавлении (исключении) зарегистрированным пользователем связанных с ним других зарегистрированных идентификаторов:

дата и время;

наименование сервиса и идентификатор;

4. Сведения о фактах авторизации и выхода из ИС провайдера хостинга, прекращениях регистрации:

дата и время;

идентификатор пользователя ИС провайдера хостинга;

технические идентификаторы пользователя ИС провайдера хостинга, переданные в ИС провайдера хостинга в силу используемых коммуникационных протоколов:

IP-адрес и порт;

номера телефонов пользователя, являющегося абонентом (пользователем) услуг подвижной радиотелефонной связи;

адрес электронной почты;

наименование программы клиента (текстовая строка в произвольном виде, содержащая сведения о наименовании программы, версии, дате обновления, а также иные регистрируемые сведения при передаче);

иная фиксируемая техническая информация;

вид события;

5. Информация в ИС провайдера хостинга, классифицирующая:

виды информационных сервисов, предоставляемых провайдером хостинга для пользователей;

виды событий, регистрируемых ИС провайдера хостинга при взаимодействии ИС провайдера хостинга с пользователем;

типы информационных ресурсов, создаваемых пользователями в ИС провайдера хостинга;

типы пользователей, обслуживаемых ИС провайдера хостинга;

виды платежных услуг (платежных сервисов), используемых провайдером хостинга;

6. Информация, фиксируемая в ИС провайдера хостинга об оказанных пользователям услугах:

идентификатор пользователя ИС провайдера хостинга;

дата и время;

описание оказанной услуги;

техническая информация, связанная с оказанной услугой;

иная фиксируемая информация о оказанной услуге;

7. Информация о фактах взаимодействия пользователей сети «Интернет» с информационными ресурсами, созданными в ИС провайдера хостинга пользователями провайдера хостинга;

8 Информация о фактах приема, передачи и (или) обработки голосовой информации, текстовых сообщений, изображений, звуков, видео- или иных электронных сообщений пользователей ИС провайдеров хостинга (при наличии), фиксируемая информация об организации приема, передачи, доставки и (или) обработки электронных сообщений, осуществляемых с использованием технологий электронных платежных систем, в том числе информация о произведенных денежных операциях (с указанием информации о корреспонденте - идентификаторе платежной системы, валюты, суммы, оплачиваемой услуги или товаров (при наличии), об иных данных, указанных при проведении денежной операции), осуществленных транзакциях (с указанием идентификатора платежной системы, суммы прихода либо расхода, иных данных, указанных при осуществлении транзакции):

дата и время события;

идентификатор пользователя ИС провайдера хостинга;

технические идентификаторы пользователя ИС провайдера хостинга, переданные в ИС в силу используемых коммуникационных протоколов:

IP-адрес и порт;

номер телефона (MSISDN) подвижной станции пользователя;

адрес электронной почты;

наименование программы клиента;

иная фиксируемая техническая информация;

вид события;

местоположение пользователя в случае его фиксации ИС провайдера хостинга (широта и долгота, иное описание местоположения);

текст сообщения и иная служебная коммуникационная информация;

техническая информация о платеже: вид платежной услуги (платежного сервиса), идентификатор платежа в сервисе, иная фиксируемая информация о платеже.

9. ТС ОРМ провайдера хостинга должны обеспечивать хранение информации о пользователях услуг провайдера хостинга в течении трех лет, о фактах взаимодействия пользователей услуг провайдера хостинга с пользователями информационно-телекоммуникационной сети «Интернет» в течение одного года со дня окончания осуществления действий, а также предоставление указанной информации уполномоченному государственному органу посредством ТС ОРМ.

Приложение № 2
к требованиям к вычислительной мощности,
используемой провайдером хостинга, для
проведения уполномоченными
государственными органами, осуществляющими
оперативно-разыскную деятельность или
обеспечение безопасности Российской
Федерации, в случаях, установленных
федеральными законами, мероприятий в целях
реализации возложенных на них задач,
утвержденным приказом Министерства
цифрового развития, связи и массовых
коммуникаций Российской Федерации
от 1 ноября 2023 г. № 935

**Требования, предъявляемые к интерфейсу взаимодействия
между пунктом управления и техническими средствами, обеспечивающими
выполнение установленных действий при проведении оперативно-разыскных
мероприятий**

1. Интерфейсы взаимодействия ТС ОРМ с ПУ должны быть организованы по выделенным каналам связи.

2. В качестве набора протоколов передачи данных следует использовать набор протоколов TCP/IP.

3. Для организации линий (каналов) связи, соединяющих ТС ОРМ и ПУ, должен быть использован сетевой интерфейс первого уровня.

4. ТС ОРМ должны предусматривать возможность создания виртуальной сети VPN (Virtual Private Network) для туннелирования всего рабочего TCP/IP трафика между ТС ОРМ и ПУ.

5. Взаимодействие ТС ОРМ с оборудованием ПУ на транспортном уровне должно осуществляться по схеме «клиент – сервер»:

1) в качестве «сервера» выступают ТС ОРМ;

2) в качестве «клиента» выступает оборудование ПУ.

6. При построении ТС ОРМ должны соблюдаться согласованные с уполномоченным государственным органом способы защиты линий (каналов) связи. ТС ОРМ должно соединяться с ПУ по протоколу TLS версии 1.2 или 1.3. При установлении соединения ТС ОРМ должны осуществлять взаимную аутентификацию с ПУ. В случае невозможности аутентифицировать одну из сторон TCP-соединение разрывается. Для аутентификации ТС ОРМ должны использовать сертификаты в формате X.509 согласованные с уполномоченным государственным органом.

7. Взаимодействие ТС ОРМ с ПУ должно осуществляться по единому каналу передачи данных, предназначенному для передачи:

- 1) от ПУ в ТС ОРМ поисковых запросов;
- 2) от ТС ОРМ на ПУ результатов выполнения поисковых запросов (в том числе постранично);
- 3) от ПУ в ТС ОРМ запросов на получение неформатированных данных (далее – запросы неформатированных данных);
- 4) от ТС ОРМ на ПУ ответов на запросы неформатированных данных;
- 5) от ПУ в ТС ОРМ специальных запросов согласно приложению № 5 к требованиям, утвержденным настоящим приказом, для получения схемы данных, статуса поисковых запросов в отложенном режиме согласно пункту 20 настоящего приложения и сигналов о функционировании ТС ОРМ согласно пункту 23 настоящего приложения;

6) от ТС ОРМ на ПУ текущей схемы данных, статуса поисковых запросов в отложенном режиме и сигналов о функционировании ТС ОРМ;

7) от ПУ в ТС ОРМ запросов о текущей конфигурации оборудования, системного и прикладного программного обеспечения ТС ОРМ, а также их состоянии (далее – запросы мониторинга);

8) от ТС ОРМ на ПУ ответов на запросы мониторинга.

8. Единый канал передачи данных создается для подключения к ТС ОРМ в виде ТСП-соединений. Посредством ТС ОРМ выполняется мониторинг порта подключения к ПУ на наличие ТСП-соединений.

9. Установление соединения ТС ОРМ с ПУ осуществляется по согласованному с уполномоченным государственным органом ТСП-порту с заданным временным интервалом.

10. Обмен данными в едином канале передачи данных на прикладном уровне должен осуществляться по протоколам HTTP 1.1 и WebSocket.

11. ТС ОРМ должны поддерживать четыре интерфейса (endpoints) взаимодействия с ПУ:

1) «/query» – для получения от ПУ поисковых запросов и специальных запросов (согласно приложению № 5 к настоящим требованиям) для получения схемы данных, управления поисковыми запросами в отложенном режиме, а также передачи на ПУ результатов выполнения данных запросов (в том числе постранично) в режиме реального времени согласно пункта 20 настоящего приложения.

2) «/subscription» – для получения от ПУ статуса выполнения поисковых запросов ТС ОРМ, которые выполняются в отложенном режиме (согласно пункта 20 настоящего приложения) и сигналов о функционировании ТС ОРМ согласно пункта 23 настоящего приложения.

3) «/download» – для получения от ПУ запросов неформатированных данных и передачи на ПУ результатов выполнения данных запросов.

4) «/metric» – для получения от ПУ запросов мониторинга и передачи на ПУ результатов выполнения данных запросов.

12. Обмен данными для конечных точек «/query», «/download» и «/metric» должен осуществляться по протоколу HTTP 1.1, для интерфейса «/subscription» –

по протоколу WebSocket. ПУ и ТС ОРМ для интерфейса «/subscription» должны поддерживать сетевое соединение с помощью механизмов протокола WebSocket.

13. Для запросов ПУ и ответов ТС ОРМ для конечных интерфейсов «/query» и «/subscription» должна использоваться версия языка описания данных и запросов GraphQL, актуальная на день опубликования настоящих требований и (или) совместимая с ней. Для кодирования содержимого запросов и ответов должен применяться формат JSON.

14. Требования, предъявляемые к формату передачи данных на языке GraphQL по протоколу WebSocket для интерфейса «/subscription», приведены в приложении № 7 к настоящим требованиям.

15. Требования, предъявляемые к запросам мониторинга и формату их передачи по интерфейсу взаимодействия между ПУ и ТС ОРМ для интерфейса «/metric», приведены в приложении № 8 к настоящим требованиям.

16. Для запросов ПУ и ответов ТС ОРМ для интерфейса «/download» должны использоваться стандартные сообщения протокола HTTP 1.1. ТС ОРМ для данной интерфейса должны поддерживать механизм передачи данных по частям Chunked Transfer Coding и механизм частичных запросов Range Requests.

17. Если результаты выполнения поисковых запросов для конечных точек «/query» и «/subscription» содержат неформатированные данные (файлы), то ответ ТС ОРМ должен содержать вместо непосредственно неформатированных данных (файлов) соответствующие ссылки (HTTP URI в полном формате согласно RFC7230 `http(s)://<IP-адрес или доменное имя ТС ОРМ>:<порт>/download/<уникальный путь к файлу>`) для каждого файла для интерфейса «/download». ТС ОРМ должны формировать уникальную ссылку для каждого файла.

18. Для получения неформатированных данных, ссылки на которые содержатся в ответе ТС ОРМ на поисковый запрос, ТС ОРМ для каждого файла должен обеспечивать возможность получения на интерфейс «/download» отдельных (от поисковых запросов) запросов получения неформатированных данных (HTTP GET-запросы) от ПУ, содержащих ссылку на запрашиваемые данные.

19. Для получения текущей схемы данных согласно приложению № 3 к требованиям, утвержденным настоящим приказом, ТС ОРМ необходимо предусмотреть возможность получения запроса «getSchema» согласно приложению № 5 к требованиям, утвержденным настоящим приказом, на интерфейс «/query». В ответ ТС ОРМ должны направить схему данных, описанную на языке SDL GraphQL.

20. Поисковые запросы должны выполняться в ТС ОРМ в двух режимах: режиме реального времени и отложенном режиме (далее – отложенный поисковый запрос). Режим выполнения поисковых запросов определяется ТС ОРМ в соответствии с установленными временными параметрами согласно пункту 21 настоящих требований, а также с учетом характера (количества запрашиваемых полей и критериев поиска) запроса, количества данных, хранящихся в ИС,

и производительностью ТС ОРМ. Все поисковые запросы от ПУ должны направляться на интерфейс «/query».

21. ТС ОРМ должны выполнять поисковые запросы в соответствии с критериями поиска, заданными на верхнем уровне запроса без вложенных объектов, согласно пункту 10 приложения № 3 к требованиям, утвержденным настоящим приказом, при следующих условиях:

по согласованию с уполномоченным государственным органом, если среднесуточный объем новых данных, поступающих в ИС, составляет от 1 до 10 Гбайт;

без согласования с уполномоченным государственным органом, если среднесуточный объем новых данных, поступающих в ИС, превышает 10 Гбайт.

22. В режиме реального времени ТС ОРМ должны поддерживать (не разрывать) сетевое соединение, по которому поступил запрос от ПУ, и результаты выполнения запроса должны передаваться от ТС ОРМ на ПУ по тому же сетевому соединению, не позднее установленных временных параметров согласно пункту 21 настоящих требований.

23. Отложенный поисковый запрос выполняется по следующему сценарию:

1) ПУ направляет поисковый запрос ТС ОРМ на интерфейс «/query»;

2) ТС ОРМ принимают решение о выполнении запроса в отложенном режиме;

3) ТС ОРМ назначают поисковому запросу уникальный в рамках данной ТС ОРМ идентификатор;

4) ТС ОРМ возвращают ПУ идентификатор отложенного поискового запроса и информацию о том, что запрос выполняется в отложенном режиме;

5) ПУ отправляет запрос «statusOfflineRequest» (запрос Subscription языка GraphQL), содержащий полученный от ТС ОРМ идентификатор отложенного поискового запроса, на интерфейс «/subscription» по протоколу WebSocket (согласно приложению № 5 к настоящим требованиям). ТС ОРМ в непрерывном режиме (без дополнительных запросов от ПУ) уведомляют ПУ об изменениях статуса выполнения отложенного поискового запроса.

ТС ОРМ должны поддерживать следующие статусы выполнения отложенного запроса:

NOTSTARTED – запрос получен, но не запущен на выполнение;

RUNNING – запрос получен и в настоящее время выполняется;

READY – выполнение запроса завершено, результирующие данные готовы;

ABORTED – выполнение запроса прервано (на стороне сервера – ТС ОРМ);

CANCELED – запрос отменен клиентом (ПУ);

6) при получении от ТС ОРМ на ПУ уведомления о завершении выполнения отложенного поискового запроса (статус «READY»), ПУ направляет запрос «getOfflineRequest» согласно приложению № 5 к требованиям, утвержденным настоящим приказом, на интерфейс «/query» с указанием идентификатора запроса. ТС ОРМ должны осуществлять передачу на ПУ результатов выполнения отложенных запросов и запросов в режиме реального времени постранично,

согласно подпункту 2 пункта 10 приложения № 3 к требованиям, утвержденным настоящим приказом;

7) в ответ на полученный запрос ТС ОРМ должны передавать на ПУ результаты выполнения отложенного поискового запроса в режиме реального времени в формате JSON. Структура результатов выполнения отложенного поискового запроса должна соответствовать структуре исходного поискового запроса, направленного ПУ на интерфейс «/query»;

8) в момент выполнения отложенного запроса (от момента получения идентификатора запроса от ТС ОРМ до получения статуса выполнения запроса «READY») ПУ может прервать данную операцию, направив запрос «_cancelOfflineRequest» согласно приложению № 5 к требованиям, утвержденным настоящим приказом) на интерфейс «/query» с указанием идентификатора запроса. В ответ ТС ОРМ должны направить результат выполнения данного запроса .

ТС ОРМ должны удалять промежуточные результаты прерванного отложенного запроса.

24. ТС ОРМ должны хранить результаты выполнения отложенных запросов согласно пункту 21 настоящих требований и незамедлительно их удалять при получении запроса от ПУ.

Для удаления на ТС ОРМ результатов выполнения отложенного поискового запроса ПУ направляет запрос «_delOfflineRequest» согласно приложению № 5 к требованиям, утвержденным настоящим приказом, на интерфейс «/query» с указанием идентификатора запроса. В ответ ТС ОРМ направляют результат выполнения данного запроса.

25. ТС ОРМ должны поддерживать два вида постраничной передачи результатов выполнения поисковых запросов: со смещением и курсорную. По согласованию с уполномоченным государственным органом может быть реализован только один из видов постраничной передачи результатов.

26. ТС ОРМ должны осуществлять контроль функционирования собственных параметров и передачу на ПУ следующих сигналов:

- 1) «Перезапуск ПО» (RESTARTDB);
- 2) «Попытка несанкционированного доступа» (UNAUTHORIZEDACCESS);
- 3) «Критическая ошибка ПО, потеря данных, дальнейшая работа невозможна» (CRITICALERROR);
- 4) «Серьезная ошибка ПО, потеря данных, но дальнейшая работа возможна» (MAJORERROR);
- 5) «Незначительная ошибка ПО, данные не потеряны, дальнейшая работа возможна» (MINORERROR);
- 6) «Изменение схемы данных» (SCHEMACHANGED);
- 7) «Предупреждение о проблеме с контролируруемыми параметрами (метриками)» (METRICALERTS). Для данного сигнала ТС ОРМ должны возвращать на ПУ дополнительную информацию о предупреждениях (согласно приложению № 5 к настоящим требованиям). ТС ОРМ должны

осуществлять обработку контролируемых метрик и предупреждений о проблемах согласованные с уполномоченным государственным органом.

Для получения сигналов ПУ необходимо отправить запрос «_trap» (запрос Subscription языка GraphQL) на интерфейс «/subscription» по протоколу WebSocket согласно приложению № 5 к требованиям, утвержденным настоящим приказом. ТС ОРМ в непрерывном режиме (без дополнительных запросов от ПУ) отправляет ПУ сигналы сразу после их возникновения на ТС ОРМ. Если в момент получения запроса «_trap» на ТС ОРМ есть сигналы, которые не были отправлены ранее ПУ (по причине разрыва соединения или по иным причинам), то ТС ОРМ должны незамедлительно передать на ПУ все данные сигналы.

27. Все запросы мониторинга и неформатированных данных должны выполняться в режиме реального времени.

28. ТС ОРМ должны использовать следующие коды ошибок для языка GraphQL:

Код ошибки	Описание ошибки
GRAPHQL_PARSE_FAILED	Запрос на языке GraphQL содержит синтаксическую ошибку
GRAPHQL_VALIDATION_FAILED	Запрос на языке GraphQL недействителен для текущей схемы данных
BAD_USER_INPUT	Запрос на языке GraphQL содержит некорректное значение для аргумента поля
OPERATION_RESOLUTION_FAILURE	Невозможно определить операцию для выполнения из запроса на языке GraphQL
BAD_REQUEST	Ошибка возникла до начала анализа запроса на языке GraphQL
REQUEST_NOT_FOUND	Отложенный запрос с указанным идентификатором не найден
NO_REQUEST_RESULT	Результаты выполнения отложенного запроса с указанным идентификатором не найдены
REQUEST_TOO_COMPLEX	Запрос на языке GraphQL содержит критерии поиска (аргументы поля) не только на верхнем уровне запроса согласно пункту 21 настоящего приложения, может применяться только в ТС ОРМ, осуществляющих поиск информации в ИС со среднесуточным объемом новых данных свыше 1 Гбайт
INTERNAL_SERVER_ERROR	Внутренняя ошибка. Возвращается в случае, если ни один из вышеуказанных кодов не подходит

Приложение № 3
к требованиям к вычислительной мощности,
используемой провайдером хостинга, для
проведения уполномоченными
государственными органами, осуществляющими
оперативно-разыскную деятельность или
обеспечение безопасности Российской
Федерации, в случаях, установленных
федеральными законами, мероприятий в целях
реализации возложенных на них задач,
утвержденным приказом Министерства
цифрового развития, связи и массовых
коммуникаций Российской Федерации
от 1 ноября 2023 г. № 935

**Требования, предъявляемые к схеме данных
и выполнению запросов от пункта управления**

1. Данные, хранящиеся в информационной системе провайдера хостинга, должны быть описаны с помощью схемы данных на языке SDL GraphQL.
2. Схема данных должна полностью описывать данные, хранящиеся в ИС.
3. Произвольный тип данных в схеме данных (далее – тип «А») имеет связь с произвольным типом данных (далее – тип «В»), если тип «А» содержит поле типа «В» или поле, являющееся списком типа «В».
4. Произвольный тип «А» в схеме данных имеет двухстороннюю связь с произвольным типом «В», если тип «А» связан с типом «В» и тип «В» связан с типом «А».
5. Для схемы данных определяется перечень базовых типов. Базовый тип данных на языке GraphQL должен содержать информацию, которая должна быть доступна для поиска в ТС ОРМ по запросам ПУ. Перечень базовых типов отражает перечень идентифицирующих признаков объектов физического мира (в том числе, номер паспорта, номер телефона, банковские реквизиты, государственный регистрационный номер транспортного средства). Перечень базовых типов и их определения на языке GraphQL приведены в приложении № 4 к требованиям, утвержденным настоящим приказом.
6. Для схемы данных определяется перечень сервисных типов. Сервисный тип данных на языке GraphQL должен отражать информацию о схеме данных, или описывать HTTP URI для неформатированных данных, или предназначаться для получения статуса отложенных поисковых запросов или сигналов о функционировании ТС ОРМ. Перечень сервисных типов, их определения и запросы для доступа к информации на языке GraphQL приведены в приложении № 4 к требованиям, утвержденным настоящим приказом.

7. Пользовательский тип данных на языке GraphQL должен описывать данные, содержащиеся в ИС и отличные от данных, описываемых базовыми типами.

8. Схема данных в части базовых, пользовательских и сервисных типов должна удовлетворять следующим требованиям:

1) каждое поле пользовательского типа, которое отражает идентифицирующий признак объекта физического мира и будет доступно в качестве критерия поиска в ТС ОРМ по запросам ПУ, должно представляться базовым типом (или его списком);

2) поля пользовательского типа, которые представляют собой ссылки (HTTP URI) на неформатированные данные (файлы), должны представляться сервисным типом «Media», определенным в приложении № 5 к требованиям, утвержденным настоящим приказом;

3) остальные поля пользовательских типов (то есть за исключением указанных в подпунктах 1 и 2 настоящего пункта) должны описываться встроенными типами языка GraphQL или любыми другими пользовательскими типами, определенными в схеме данных;

4) для каждого пользовательского типа, имеющего хотя бы одно поле базового типа, или связанного с другим пользовательским типом, имеющим хотя бы одно поле базового типа, должны быть определены следующие объекты языка GraphQL: тип для представления результатов выполнения поисковых запросов и входной объект для задания параметров поиска (далее – входной объект);

5) тип для представления результатов выполнения поисковых запросов должен иметь следующую структуру:

Имя поля	Тип поля	Назначение поля
cursor	String	значение курсора для последнего элемента списка (поле «result») с результатами выполнения поискового запроса для данного пользовательского типа
hasNextPage	Boolean	признак наличия следующей страницы с результатами выполнения поискового запроса для данного пользовательского типа (может использоваться с любым видом постраничного получения результатов выполнения запроса)
result	список соответствующего пользовательского типа	результат выполнения поискового запроса для данного

		пользовательского типа
--	--	------------------------

Формат типа для представления результатов выполнения поисковых запросов должен быть следующим:

```
type {{UserTypeResult}} {
  cursor: String
  hasNextPage: Boolean
  result: [{{UserType}}!]
}
```

в котором:

{{UserTypeResult}} – произвольное имя типа для представления результатов выполнения поисковых запросов,

{{UserType}} – имя пользовательского типа, для которого определен тип {{UserTypeResult}};

б) каждый входной объект для пользовательского типа должен иметь следующую структуру:

Имя поля	Тип поля	Назначение поля
and	список исходных входных объектов	для задания параметров поиска с логической функцией «И»
or	список исходных входных объектов	для задания параметров поиска с логической функцией «ИЛИ»
not	исходный входной объект	для задания параметров поиска с логической функцией «НЕ»
все имена полей пользовательского типа, представленные базовыми типами	входные объекты для базовых типов, определенные в приложении № 5 к требованиям, утвержденным настоящим приказом	для задания параметров поиска конкретных полей пользовательского типа, представленных базовыми типами

Формат входного объекта для пользовательского типа:

```
input {{UserTypeFilterInput}} {
  and: [{{UserTypeFilterInput}}]
  or: [{{UserTypeFilterInput}}]
  not: {{UserTypeFilterInput}}
  {{baseField1}}: {{BaseFilterInput1}}
  ...
  {{baseFieldN}}: {{BaseFilterInputN}}
```

| }

в котором:

`{{UserTypeFilterInput}}` – произвольное имя входного объекта для пользовательского типа,

`{{baseField1}}...{{baseFieldN}}` – поля пользовательского типа, представленные одним из базовых типов,

`{{baseFilterInput1}}...{{baseFilterInputN}}` – входные объекты для базовых типов, определенные в приложении № 6 к требованиям, утвержденным настоящим приказом;

7) все пользовательские типы, их поля и аргументы должны иметь описание на русском языке;

8) связь между двумя произвольными пользовательскими типами должна быть двухсторонней за исключением случаев, если один из пользовательских типов не имеет ни одного поля базового типа;

9) схема данных должна содержать все сервисные типы из перечня, определенного в приложении № 6 к настоящим требованиям.

9. Схема данных в части стандартных типов Query, Subscription (определяют доступные данные и параметры для поисковых запросов) языка GraphQL должна соответствовать следующим требованиям:

1) тип Query должен иметь поле «request» для задания поисковых запросов. Формат типа Query:

```
| request: QueryMessage!
```

2) тип QueryMessage должен иметь следующую структуру:

Имя поля	Тип поля	Назначение поля
requestID	String	идентификатор запроса
offline	Boolean	признак выполнения запроса в отложенном режиме
maxPageSize	Int	максимальное количество элементов в списке с результатами выполнения поискового запроса для одной страницы, которое поддерживает ТС ОРМ
все имена пользовательских типов, имеющие хотя бы одно поле базового типа	тип для представления результатов выполнения поисковых запросов, соответствующий пользовательскому типу	задание параметров поиска и постраничной передачи результатов поиска для соответствующего

		пользовательского типа
--	--	---------------------------

Каждое поле типа QueryMessage, заданное списком пользовательского типа, должно иметь следующие аргументы:

filter – задает параметры поиска и представляется соответствующим входным объектом для данного пользовательского типа;

offset – задает начальное смещение для постраничной передачи результатов;

limit – задает максимальное количество элементов на одной странице при постраничной передаче результатов, но не более значения maxPageSize;

cursor – значение курсора при использовании курсорной постраничной передачи результатов.

Аргументы «offset» и «limit» должны использоваться для постраничной передачи результатов со смещением, аргументы limit и cursor – для курсорной постраничной передачи результатов.

Формат типа QueryMessage:

```
type QueryMessage {
  requestID: String!
  offline: Boolean!
  maxPageSize: Int
  {{userType1}}(filter: {{UserTypeFilterInput1}}!, offset: Int, limit:
Int, cursor: String): {{UserTypeResult1}}!
  ...
  {{userTypeN}}(filter: {{UserTypeFilterInputN}}!, offset: Int, limit:
Int, cursor: String): {{UserTypeResultN}}!
}
```

в котором:

{{userType1}}...{{userTypeN}} – имена пользовательских типов, имеющих хотя бы одно поле базового типа;

{{UserTypeResult1}}...{{UserTypeResultN}} – имена типов для представления результатов выполнения поисковых запросов, соответствующие пользовательским типам;

{{UserTypeFilterInput1}}...{{UserTypeFilterInputN}} – входные объекты, соответствующие пользовательским типам;

3) типы Query и Subscription должны также содержать все запросы, соответствующие сервисным типам, которые приведены в приложении № 5 к требованиям, утвержденным настоящим приказом.

10. На ТС ОРМ должны использоваться актуальные перечни базовых, сервисных типов и входных объектов для поиска.

11. Использование базовых, сервисных типов и входных объектов для задания параметров поиска, не приведенных в приложениях №№ 4-6 к требованиям, утвержденным настоящим приказом, не допускается.

12. Названия всех пользовательских типов и их полей должны соответствовать следующим требованиям:

1) использование символа подчеркивания «_» в именах типов и их полей не допускается;

2) имена пользовательских типов пишутся с заглавной буквы и могут состоять из нескольких слов, написанных слитно без пробелов, при этом каждое слово с заглавной буквы;

3) имена полей пользовательских типов пишутся со строчной буквы и могут состоять из нескольких слов, написанных слитно без пробелов, при этом каждое слово кроме первого с заглавной буквы.

Приложение № 4
к требованиям к вычислительной мощности,
используемой провайдером хостинга, для
проведения уполномоченными
государственными органами, осуществляющими
оперативно-разыскную деятельность или
обеспечение безопасности Российской
Федерации, в случаях, установленных
федеральными законами, мероприятий в целях
реализации возложенных на них задач,
утвержденным приказом Министерства
цифрового развития, связи и массовых
коммуникаций Российской Федерации
от 1 ноября 2023 г. № 935

**Перечень базовых типов для описания схемы данных технических средств,
обеспечивающих выполнение установленных действий при проведении
оперативно-разыскных мероприятий**

№ п/п	Описание	Базовый тип
1.	Идентификатор мобильного абонента	<pre> """ Базовый тип идентификатора мобильного абонента """ type ImsiBase { """ Идентификатор мобильного абонента """ imsi: String! } </pre>
2.	Идентификатор мобильной станции	<pre> """ Базовый тип идентификатора мобильной станции """ type ImeiBase { """ Идентификатор мобильной станции """ imei: String! } </pre>
3.	Номер абонента сети сотовой связи	<pre> """ Базовый тип номера абонента сети сотовой связи """ type MsisdnBase { """ Номер абонента сети сотовой связи """ msisdn: String! } </pre>
4.	Код страны, в которой находится оператор мобильной связи	<pre> """ Базовый тип кода страны, в которой находится оператор связи """ </pre>

		<pre> type MccBase { """ Код страны """ mcc: Int! } </pre>
5.	Код оператора связи	<pre> """ Базовый тип кода оператора связи """ type MncBase { """ Код оператора связи """ mnc: Int! } </pre>
6.	Код географической зоны, которая обслуживается одним контроллером базовых станций	<pre> """ Базовый тип кода географической зоны, обслуживаемой одним контроллером базовых станций """ type LacBase { """ Код географической зоны """ lac: Int! } </pre>
7.	Идентификатор сектора базовой станции	<pre> """ Базовый тип идентификатора сектора базовой станции """ type CellBase { """ Идентификатор сектора базовой станции """ cell: Int! } </pre>
8.	Адрес электронной почты	<pre> """ Базовый тип адреса электронной почты """ type EmailBase { """ Адрес электронной почты """ email: String! } </pre>
9.	Адресная информация	<pre> """ Базовый тип адресных данных """ type AddressBase { """ Неструктурированный адрес """ address: String """ Почтовый индекс, zip-код """ zip: String """ Страна """ county: String } </pre>

		<pre> Область """ region: String """ Район, муниципальный округ """ zone: String """ Город, посёлок, деревня """ city: String """ Улица """ street: String """ Дом, строение """ building: String """ Корпус """ buildsect: String """ Квартира, офис """ apartment: String } </pre>
10.	Паспортные данные	<pre> """ Базовый тип паспортных данных """ type PassportBase { """ Серия паспорта """ series: String! """ Номер паспорта """ number: String! } </pre>
11.	Фамилия, имя, отчество (при наличии)	<pre> """ Базовый тип фамилия, имя, отчество """ type PersonBase { """ Неструктурированная информация о фамилии, имени, отчестве """ fullName: String! """ Имя """ name: String """ Отчество (при наличии) """ middleName: String """ Фамилия """ lastName: String } </pre>

		<pre> }</pre>
12.	Номер удостоверения водителя транспортного средства	<pre> """ Базовый тип номера удостоверения водителя транспортного средства """ type DrivingLicenseNumberBase { """ Номер удостоверения водителя транспортного средства """ drivingLicenseNumber: String! } </pre>
13.	Государственный регистрационный номер транспортного средства	<pre> """ Базовый тип государственного регистрационного номера транспортного средства """ type VehicleGosNumberBase { """ Государственный регистрационный номер транспортного средства """ vehicleGosNumber: String! } </pre>
14.	Идентификационный номер налогоплательщика	<pre> """ Базовый тип идентификационного номера налогоплательщика """ type InnBase { """ Идентификационный номер налогоплательщика """ inn: String! } </pre>
15.	Информация о дате и времени события	<pre> """ Базовый тип информации о дате и времени """ type DateTimeBase { """ Информация о дате и времени в расширенном формате местного времени с разницей со Всемирным координированным временем (UTC) YYYY-MM-DDThh:mm:ss±hh:mm """ utc: String! } </pre>
16.	Геометрические пространственно-временные данные: точка, массив точек, информация о месте и времени, полигон, мультиполигон, геотрек	<pre> """ Базовый тип пространственной информации: точка """ type PointBase { """ Долгота """ longitude: Float! """ Широта """ latitude: Float! } """ Базовый тип пространственной информации: последовательность точек (линия) </pre>

		<pre> """ type LineBase { """ Линия - последовательность точек в пространстве. """ points: [PointBase!]! } """ Базовый тип пространственно-временной информации: точка в пространстве с привязкой к дате и времени """ type PointTimeBase { """ Точка в пространстве """ point: PointBase! """ Информация о дате и времени """ time: DateTimeBase! } """ Базовый тип пространственной информации: полигон """ type PolygonBase { """ Последовательность линий """ lines: [LineBase!]! } """ Базовый тип пространственной информации: мультиполигон """ type MultiPolygonBase { """ Последовательность полигонов """ polygons: [PolygonBase!]! } """ Базовый тип пространственно-временной информации: трек """ type GeoTrackBase { """ Последовательность точек в пространстве с привязкой к дате и времени """ pointsInTime: [PointTimeBase!]! } </pre>
17.	Сведения о юридических лицах и индивидуальных предпринимателях	<pre> """ Базовый тип сведений о юридических лицах и индивидуальных предпринимателях """ type OrganizationInfoBase { """ Полное наименование """ nameFull: String """ Сокращенное наименование """ nameSmall: String } </pre>

```

"""
Государственный регистрационный номер ОГРН и ОГРНИП
"""
grn: GRNBase!
"""
Идентификационный номер налогоплательщика
"""
inn: InnBase
"""
Информация о банковских данных
"""
bankInfo: BankAccountInfoBase
"""
Адрес в пределах места нахождения
"""
address: AddressBase
"""
Информация о номере телефона
"""
msisdn: [MsisdnBase!]
"""
Web-сайт
"""
webSite: URLBase
"""
Электронная почта (при наличии)
"""
email: EmailBase
"""
Дата регистрации
"""
dateRegistration: DateTimeBase!
"""
Сведения о лице, имеющем право без доверенности
действовать от имени юридического лица
"""
representative: OrganizationRepresentativeBase!
"""
ОКВЭД
"""
okved: [String!]
"""
ЕГРЮЛ
"""
egrul: String
"""
ЕГРИП
"""
egrip: String
}

"""
Базовый тип государственного регистрационного номера
ОГРН и ОГРНИП
"""
type GRNBase {
    """
    Государственный регистрационный номер - ОГРН и ОГРНИП
    """
    grn: String!
}

"""

```

		<pre> Базовый тип сведений об учредителях (участниках) юридического лица, лицах, имеющих право без доверенности действовать от имени юридического лица """ type OrganizationRepresentativeBase { """ Фамилия, имя, отчество (при наличии) представителя """ name: PersonBase! """ Идентификационный номер налогоплательщика """ inn: InnBase! """ Должность """ position: String! } </pre>
18.	Банковские данные	<pre> """ Базовый тип банковских данных """ type BankAccountInfoBase { """ Наименование банка """ bankName: String """ Номер счёта """ account: String """ Корреспондентский счет """ corrAccount: String """ Номер карты """ cardNumber: String! """ Адрес банка """ bankAddress: AddressBase """ Банковский идентификационный код """ rcbic: String """ Код причины постановки на учет кредитной организации предназначен для идентификации юридического лица, которое является налогоплательщиком """ kpp: String } """ Базовый тип банковских переводов """ type BankTransferInfoBase { """ Сумма перевода """ amount: String! } </pre>

		<pre> """ Отправитель перевода """ from: BankAccountInfoBase """ Получатель перевода """ to: BankAccountInfoBase """ Дата и время перевода """ data: DateTimeBase! } </pre>
19.	IP-адрес	<pre> """ Базовый тип IP-адреса """ type IpAddressBase { """ IP-адрес """ ip: String! """ Маска подсети в виде префикса """ mask: Int """ Тип IP-адреса: ipv4, ipv6 """ type: IpType! } enum IpType { IPV4 IPV6 } </pre>
20.	Информация об участнике сетевого соединения	<pre> """ Базовый тип информации об участнике сетевого соединения """ type NetworkPeerBase { """ Информация о IP-адресе """ ip: IpAddressBase! """ Номер порта """ port: Int! """ Номер протокола согласно RFC3232 """ protocolNumber: Int } </pre>
21.	URL-адрес	<pre> """ Базовый тип URL-адреса """ type URLBase { """ URL-адрес """ url: String! } </pre>

22.	Доменное имя	<pre>""" Базовый тип доменного имени """ type DomainNameBase { """ Доменное имя """ domainName: String! }</pre>
23.	Имя и (или) идентификатор пользователя	<pre>""" Базовый тип имени/идентификатора пользователя """ type LoginBase { """ Имя/идентификатор пользователя """ login: String! }</pre>

Приложение № 5
к требованиям к вычислительной мощности,
используемой провайдером хостинга, для
проведения уполномоченными
государственными органами, осуществляющими
оперативно-разыскную деятельность или
обеспечение безопасности Российской
Федерации, в случаях, установленных
федеральными законами, мероприятий в целях
реализации возложенных на них задач,
утвержденным приказом Министерства
цифрового развития, связи и массовых
коммуникаций Российской Федерации
от 1 ноября 2023 г. № 935

**Перечень сервисных типов и соответствующих им запросов
для описания схемы данных технических средств, обеспечивающих
выполнение установленных действий при проведении оперативно-разыскных
мероприятий**

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
1	Схема данных, описанная на языке SDL	Отсутствует	<pre>type Query { """ Запрос схемы данных """ getSchema: String! }</pre>	Запрос схемы данных, описанной на языке SDL GraphQL
2	Идентификатор отложенного запроса	Отсутствует	<pre>type Query { """ Запрос результата выполнения отложенного запроса """ getOfflineRequest(requestID: String!, offset: Int, limit: Int, cursor: String): QueryMessage! }</pre>	Запрос результата выполнения отложенного запроса по его идентификатору либо постраничное получение результатов с использованием аргументов «offset», «limit» и «cursor» согласно подпункту 2 пункта 9 приложения № 2 к требованиям, утвержденным настоящим приказом. Ответ (результат)

				<p>выполнения запроса) предоставляется в виде данных в формате JSON в соответствии с изначальным GraphQL запросом.</p> <p>Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено сообщение об ошибке GraphQL с кодом REQUEST_NOT_FOUND.</p> <p>Если отложенный запрос еще не выполнен или результаты его выполнения уже удалены, то должно быть возвращено сообщение об ошибке GraphQL с кодом NO_REQUEST_RESULT.</p>
			<pre> type Query { """ Удаление результатов отложенного запроса """ delOfflineRequest(requestID: String!): Boolean! } </pre>	<p>Удаление результатов отложенного запроса по его идентификатору.</p> <p>Если удаление произошло успешно, то должен быть возвращен ответ True.</p> <p>Если удаление не удалось выполнить, то должен быть возвращен ответ False и описание</p>

				<p>ошибки GraphQL. Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено сообщение об ошибке GraphQL с кодом REQUEST_NOT_FOUND.</p>
			<pre> type Query { """ Отмена выполнения отложенного запроса """ cancelOfflineRequest(requestID: String!): Boolean! } </pre>	<p>Прерывание (отмена) выполнения отложенного запроса по его идентификатору. Если прерывание произошло успешно, то должен быть возвращен ответ True. Если прерывание не удалось выполнить, то должен быть возвращен ответ False и описание ошибки GraphQL. Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено сообщение об ошибке GraphQL с кодом REQUEST_NOT_FOUND.</p>
3	Информация о статусе выполнения отложенного запроса	<pre> """ Сервисный тип для информации о статусе выполнения отложенного запроса """ type RequestInfo { </pre>	<pre> type Subscription { """ Запрос статуса отложенного запроса """ statusOfflineRequest(requestID: String!): RequestInfo! } </pre>	<p>Запрос статуса отложенного запроса по его идентификатору. Ответ предоставляется в</p>

		<pre> Идентификатор отложенного запроса """ requestID: String! """ Статус отложенного запроса """ status: RequestStatus """ Дата и время начала выполнения отложенного запроса """ startsAt: DateTimeBase } """ Возможные статусы выполнения отложенного запроса """ enum RequestStatus { """ Запрос еще не выполняется """ NOTSTARTED """ Запрос выполняется """ RUNNING """ Запрос выполнен """ READY """ Выполнение запрос прервано TC OPM """ ABORTED """ Запрос отменен ПУ """ CANCELED } </pre>	}	<p>виде сообщения типа RequestInfo. Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено сообщение об ошибке GraphQL с кодом REQUEST_NOT_FOUND.</p>
4	<p>HTTP URI для неформатированных данных</p>	<pre> """ Сервисный тип для неформатированных данных """ type Media { """ HTTP URI для неформатированных данных в полном формате согласно RFC7230 http(s)://<IP-адрес или доменное имя TC OPM>:<порт>/download/<у никальный путь к файлу> """ </pre>	Отсутствует	Отсутствует

		<pre> link: String! } </pre>		
5	Сигналы о функционировании ТС ОРМ	<pre> """ Сервисный тип для сигналов о функционировании ТС ОРМ """ type Trap { """ Тип сигнала """ type: TrapType! """ Описание сигнала """ message: String """ Дата и время возникновения сигнала """ startsAt: DateTimeBase! """ Дополнительная информация о предупреждениях для контролируемых метрик. Только для сигнала типа METRICALERTS """ metricalerts: MetricAlerts } """ Возможные типы сигналов """ enum TrapType { """ Перезапуск ПО """ RESTARTDB """ Попытка несанкционированного доступа """ UNAUTHORIZEDACCESS """ Критическая ошибка ПО """ CRITICALERROR """ Серьезная ошибка ПО """ MAJORERROR """ Незначительная ошибка ПО """ MINORERROR </pre>	<pre> type Subscription { """ Запрос получения сигналов о функционировании ТС ОРМ """ trap: Trap! } </pre>	Запрос получения сигналов о функционировании ТС ОРМ. Ответ предоставляется в виде сообщения типа Trap. Если тип сигнала «Предупреждение о проблеме с контролируемыми параметрами (METRICALERTS)», то ТС ОРМ должны возвращать дополнительную информацию о предупреждениях в поле «metricalerts».

```

"""
Изменение схемы
данных
"""
SCHEMACHANGED
"""
Предупреждение о
проблеме с
контролируемой метрикой
"""
METRICALERTS
}

"""
Дополнительная
информация о
предупреждении
"""
type MetricAlerts {
    """
    Ключ, определяющий
    группу предупреждений
    """
    groupKey: String!
    """
    Статус предупреждения
    """
    status: AlertStatus!
    """
    Определяет получателя
    предупреждения
    """
    receiver: String!
    """
    Метки, по которым
    сгруппированы
    предупреждения
    """
    groupLabels:
    [KeyValueRef!]!
    """
    Общие метки для всех
    предупреждений
    """
    commonLabels:
    [KeyValueRef!]!
    """
    Общие описания для
    всех предупреждений
    """
    commonAnnotations:
    [KeyValueRef!]!
    """
    URL менеджера
    предупреждений
    """
    externalURL: String!
    """
    Список предупреждений
    """
    alerts: [Alert!]!
}

```

```

"""
Возможные статусы
предупреждений
"""
enum AlertStatus {
    """
    Проблема решена
    """
    RESOLVED
    """
    Проблема не решена
    """
    FIRING
}

"""
Предупреждение о
проблеме с
контролируемой метрикой
"""
type Alert {
    """
    Статус предупреждения
    """
    status: AlertStatus!
    """
    Метки для
    предупреждения
    """
    labels:
    [KeyValueRef!]!
    """
    Описания для
    предупреждения
    """
    annotations:
    [KeyValueRef!]!
    """
    Дата и время
    возникновения
    предупреждения
    """
    startsAt:
    DateTimeBase!
    """
    Дата и время решения
    проблемы, в результате
    которой возникло
    предупреждение
    """
    endsAt: DateTimeBase!
    """
    URL сущности, в
    которой возникла
    проблема
    """
    generatorURL: String!
    """
    Идентификатор
    предупреждения
    """
    fingerprint: String!
}

```

		<pre>""" Ключ-значение """ type KeyValueRef { """ Ключ """ key: String! """ Значение """ value: String! }</pre>		
--	--	---	--	--

Приложение № 6
к требованиям к вычислительной мощности,
используемой провайдером хостинга, для
проведения уполномоченными
государственными органами, осуществляющими
оперативно-разыскную деятельность или
обеспечение безопасности Российской
Федерации, в случаях, установленных
федеральными законами, мероприятий в целях
реализации возложенных на них задач,
утвержденным приказом Министерства
цифрового развития, связи и массовых
коммуникаций Российской Федерации
от 1 ноября 2023 г. № 935

Перечень входных объектов для задания параметров поиска

1. В схеме данных для задания параметров поиска должны быть описаны входные объекты Input Objects на языке GraphQL для каждого пользовательского типа, имеющего не менее одного поля базового типа, или связанного с другим пользовательским типом, имеющего не менее одного поля базового типа.

2. Для описания входных объектов, соответствующих пользовательским типам, должны использоваться входные объекты для базовых типов, приведенные в настоящем приложении.

3. Каждый входной объект для базового типа должен содержать поля «and», «or» и «not» (соответствуют логическим функциям «И», «ИЛИ», «НЕ»), а также может содержать поля для функций, приведенных в таблице № 1 настоящего приложения. В случае применения числовой функций к множеству строк используется лексикографический порядок.

Таблица № 1.

№ п/п	Функция	Использование	Описание
1	lt	Для целочисленных значений: lt: 999 Для дробных значений: lt: 999.99 Для строковых значений: lt: "999"	Меньше, чем
2	le	le: 999	Меньше либо равно
3	eq	eq: 999	Равно

4		in	in: [0, 999]	В [списке]
5		between	between: { min: 0, max: 999 }	Между значениями min и max
6		ge	ge: 999	Больше либо равно
7		gt	gt: 999	Больше
8	Регулярные выражения	regexp	regexp: "\\d\\d/\\d\\d/\\d{4}"	Регулярное выражение в формате POSIX Basic Regular Expression
9	Поиск по тексту	allofterms	allofterms: "Москва, Тверская"	Соответствие строкам, содержащим все указанные термы в произвольном порядке, без учета регистра
10		anyofterms	anyofterms: "Москва, Тверская"	Соответствие строкам, содержащим любой из указанных термов в произвольном порядке, без учета регистра
11		alloftext	alloftext: "Москва, Тверская"	Полнотекстовый поиск по всему полю
12		anyoftext	anyoftext: "Москва, Тверская"	Полнотекстовый поиск по любой части поля
13	Геопоиск	near	near: { distance: 1000.0, point: { longitude: 55.7506539, latitude: 37.6195138 } }	Соответствие объектам, в которых предикат находится не более, чем на расстоянии «distance» метров от точки «point»
14		within	within: { polygon: { lines: [{ points: [{ longitude: 55.0, latitude: 37.0 }, { longitude: 56.0, latitude: 37.0 },] }] } }	Соответствие объектам, предикат которых находится в пределах полигона, заданного в качестве аргумента

		<pre> { longitude: 56.0, latitude: 33.0 }, { longitude: 55.0, latitude: 38.0 }] }] } </pre>	
15	contains	<pre> contains: { polygon: { lines: [{ points: [{ longitude: 55.0, latitude: 37.0 }, { longitude: 56.0, latitude: 37.0 }, { longitude: 56.0, latitude: 33.0 }, { longitude: 55.0, latitude: 38.0 }] }] } } </pre>	Соответствует объектам, предикат которых содержит указанные точку или ПОЛИГОН
16	intersects	<pre> intersects: { polygon: { lines: [{ points: [{ longitude: 55.0, latitude: 37.0 }, { longitude: 56.0, latitude: 37.0 }, { longitude: 56.0, latitude: 33.0 }, { longitude: 55.0, latitude: 38.0 }] }] } } </pre>	Соответствует объектам, предикат которых имеет непустое пересечение с указанным ПОЛИГОНОМ или МУЛЬТИПОЛИГОНОМ

			<pre>] } } </pre>	
17	Геопоиск с учетом времени события, используется для геотрека.	nearTime	<pre> nearTime: { distance: 1000.0, point: { longitude: 55.7506539, latitude: 37.6195138 }, time: { utc: { between: { min: { utc: "2020-12-19T16:30:00+03:00" } max: { utc: "2020-12-19T16:40:00+03:00" } } } } } </pre>	Соответствие объектам, в которых предикат находится не более, чем на расстоянии «distance» от точки «point», при этом время, в которое предикат находился в данной области, ограничено соответствующим фильтром по времени
18		withinTime	<pre> withinTime: { polygon: { lines: [{ points: [{ longitude: 55.0, latitude: 37.0 }, { longitude: 56.0, latitude: 37.0 }, { longitude: 56.0, latitude: 33.0 }, { longitude: 55.0, latitude: 38.0 }] }] }, time: { utc: { between: { min: { utc: "2020-12-19T16:30:00+03:00" } max: { utc: "2020-12-19T16:40:00+03:00" } } } } } </pre>	Соответствие объектам, предикат которых находится в пределах полигона, заданного в качестве аргумента, при этом время, в которое предикат находился в данной области, ограничено соответствующим фильтром по времени

		<pre> } } } </pre>	
19	intersectsTime	<pre> intersectsTime: { polygon: { lines: [{ points: [{ longitude: 55.0, latitude: 37.0 }, { longitude: 56.0, latitude: 37.0 }, { longitude: 56.0, latitude: 33.0 }, { longitude: 55.0, latitude: 38.0 }] }] }, time: { utc: { between: { min: { utc: "2020-12-19T16:30:00+03:00" } max: { utc: "2020-12-19T16:40:00+03:00" } } } } } </pre>	Соответствует объектам, предикат которых имеет непустое пересечение с заданным полигоном или мультиполигоном, при этом время, в которое предикат находился в данной области, ограничено соответствующим фильтром по времени
20	intersectsTrack	<pre> intersectsTrack: { geotrack: { pointsInTime: [{ point: { longitude: 55.7506539, latitude: 37.6195138 }, time: { utc: "2020-12-19T16:50:00+03:00" } }, { point: { </pre>	Соответствует объектам, предикат которых имеет непустое пересечение с заданным геотреком с точностью по времени «deltatime» секунд и точностью по расстоянию «distance» метров

		<pre> longitude:55.8224554, latitude: 37.7201284 }, time: { utc: "2020-12- 19T16:40:00+03:00" } }], deltaTime: { second: 600 }, distance: 1000.0 } </pre>	
--	--	--	--

4. Перечень входных объектов для базовых типов и доступные для них функции приведены в таблице № 2 настоящего приложения.

Таблица № 2.

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
1	type ImsiBase	<pre> """" Входной объект для базового типа идентификатора мобильного абонента """" input ImsiBaseFilter { """" Фильтр для идентификатора мобильного абонента """" imsi: StringRegExpExactFilter and: [ImsiBaseFilter] or: [ImsiBaseFilter] not: ImsiBaseFilter } </pre>
2	type ImeiBase	<pre> """" Входной объект для базового типа идентификатора мобильной станции """" input ImeiBaseFilter { """" Фильтр для идентификатора мобильной станции """" imei: StringRegExpExactFilter and: [ImeiBaseFilter] or: [ImeiBaseFilter] not: ImeiBaseFilter } </pre>
3	type MsisdnBase	<pre> """" Входной объект для базового типа номера абонента сети сотовой связи """" </pre>

		<pre>input MsisdnBaseFilter { """" Фильтр для номера абонента сети сотовой связи """" msisdn: StringRegExpExactFilter and: [MsisdnBaseFilter] or: [MsisdnBaseFilter] not: MsisdnBaseFilter }</pre>
4	type MccBase	<pre>"""" Входной объект для базового типа кода страны, в которой находится оператор связи """" input MccBaseFilter { """" Фильтр для кода страны, в которой находится оператор связи """" mcc: IntFilter and: [MccBaseFilter] or: [MccBaseFilter] not: MccBaseFilter }</pre>
5	type MncBase	<pre>"""" Входной объект для базового типа кода оператора связи """" input MncBaseFilter { """" Фильтр для кода оператора связи """" mnc: IntFilter and: [MncBaseFilter] or: [MncBaseFilter] not: MncBaseFilter }</pre>
6	type LacBase	<pre>"""" Входной объект для базового типа кода географической зоны, обслуживаемой одним контроллером базовых станций """" input LacBaseFilter { """" Фильтр для кода географической зоны, обслуживаемой одним контроллером базовых станций """" lac: IntFilter and: [LacBaseFilter] or: [LacBaseFilter] not: LacBaseFilter }</pre>

		<pre> } </pre>
7	type CellBase	<pre> """" Входной объект для базового типа идентификатора сектора базовой станции """" input CellBaseFilter { """" Фильтр для идентификатора сектора базовой станции """" cell: IntFilter and: [CellBaseFilter] or: [CellBaseFilter] not: CellBaseFilter } </pre>
8	type EmailBase	<pre> """" Входной объект для базового типа адреса электронной почты """" input EmailBaseFilter { """" Фильтр для адреса электронной почты """" email: StringRegExpExactFilter and: [EmailBaseFilter] or: [EmailBaseFilter] not: EmailBaseFilter } </pre>
9	type AddressBase	<pre> """" Входной объект для базового типа структурированных адресных данных """" input AddressBaseFilter { """" Фильтр для неструктурированного адреса """" address: StringRegExpFullTextFilter """" Фильтр для почтового индекса, zip-кода """" zip: StringExactFilter """" Фильтр для страны """" county: StringExactFilter """" Фильтр для области """" region: StringExactFilter """" Фильтр для района, муниципального округа </pre>

		<pre> "" zone: StringExactFilter "" Фильтр для города, посёлка, деревни "" city: StringExactFilter "" Фильтр для улицы "" street: StringExactFilter "" Фильтр для дома, строения "" building: StringExactFilter "" Фильтр для корпуса "" buildsect: StringExactFilter "" Фильтр для квартиры, офиса "" apartment: StringExactFilter and: [AddressBaseFilter] or: [AddressBaseFilter] not: AddressBaseFilter } </pre>
10	type PassportBase	<pre> "" Входной объект для базового типа паспортных данных "" input PassportBaseFilter { "" Фильтр для номера паспорта "" number: StringRegExpExactFilter "" Фильтр для серии паспорта "" series: StringRegExpExactFilter and: [PassportBaseFilter] or: [PassportBaseFilter] not: PassportBaseFilter } </pre>
11	type PersonBase	<pre> "" Входной объект для базового типа фамилия, имя, отчество "" input PersonBaseFilter { "" Фильтр для неструктурированной информации о фамилия, имя, отчество </pre>

		<pre> """" fullName: StringTermFullTextFilter """" Фильтр для имени """" name: StringTermExactFilter """" Фильтр для отчества (при наличии) """" middleName: StringTermExactFilter """" Фильтр для фамилии """" lastName: StringTermExactFilter and: [PersonBaseFilter] or: [PersonBaseFilter] not: PersonBaseFilter } </pre>
12	type DrivingLicenseNumberBase	<pre> """" Входной объект для базового типа номера удостоверения водителя транспортного средства """" input DrivingLicenseNumberBaseFilter { """" Фильтр для номера удостоверения водителя транспортного средства """" drivingLicenseNumber: StringTermExactFilter and: [DrivingLicenseNumberBaseFilter] or: [DrivingLicenseNumberBaseFilter] not: DrivingLicenseNumberBaseFilter } </pre>
13	type VehicleGosNumberBase	<pre> """" Входной объект для базового типа государственного регистрационного номера транспортного средства """" input VehicleGosNumberBaseFilter { """" Фильтр для государственного регистрационного номера транспортного средства """" vehicleGosNumber: StringTermExactFilter and: [VehicleGosNumberBaseFilter] or: [VehicleGosNumberBaseFilter] not: VehicleGosNumberBaseFilter } </pre>
14	type InnBase	<pre> """" Входной объект для базового типа идентификационного номера налогоплательщика """" input InnBaseFilter { </pre>

		<pre> """ Фильтр для значения идентификационного номера налогоплательщика """ inn: StringTermExactFilter and: [InnBaseFilter] or: [InnBaseFilter] not: InnBaseFilter } </pre>
15	type DateTimeBase	<pre> """ Входной объект для базового типа информации о дате и времени """ input DateTimeBaseFilter { """ Фильтр для информации о дате и времени в расширенном формате местного времени с разницей со Всемирным координированным временем (UTC) YYYY-MM-DDThh:mm:ss±hh:mm """ utc: DateTimeStringFilter and: [DateTimeBaseFilter] or: [DateTimeBaseFilter] not: DateTimeBaseFilter } """ Входной объект: используемые операции сравнения для информации о дате и времени """ input DateTimeStringFilter { eq: String in: [String] le: String lt: String ge: String gt: String between: DateTimeRange } """ Входной объект: информация о временном промежутке для использования в операциях сравнения """ input DateTimeRange { min: String! max: String! } } </pre>
16	type PointBase	<pre> """ Входной объект для базового типа пространственной информации: точки """ </pre>

		<pre> input PointBaseFilter { """" Фильтр для точек в пространстве: нахождение около аргумента фильтра в заданных пределах """" near: NearFilter """" Фильтр для точек в пространстве: нахождение внутри заданного полигона """" within: WithinFilter and: [PointBaseFilter] or: [PointBaseFilter] not: PointBaseFilter } """" Входной объект: пространственный фильтр нахождения около заданной точки в указанных пределах """" input NearFilter { """" Дистанция в метрах до используемой точки в пространстве в фильтре """" distance: Float! """" Точка в пространстве, по дистанции до которой происходит фильтрация """" point: PointRef! } """" Входной объект: пространственный фильтр нахождения внутри заданного полигона """" input WithinFilter { """" Полигон в пространстве, по нахождению в котором происходит фильтрация """" polygon: PolygonRef! } """" Входной объект: точка в пространстве для использования в фильтрах. """" input PointRef { """" Долгота в градусах """" longitude: Float! </pre>
--	--	--

		<pre> """ Широта в градусах """ latitude: Float!} </pre>
17	type LineBase	<pre> """ Входной объект для базового типа пространственной информации: последовательности точек (линии) """ input LineBaseFilter { """ Фильтр для последовательности точек в пространстве: нахождение около аргумента фильтра в заданных пределах """ near: NearFilter """ Фильтр для последовательности точек в пространстве: нахождение внутри заданного полигона """ within: WithinFilter """ Фильтр для последовательности точек в пространстве: пересечение с заданным полигоном """ intersects: IntersectsFilter and: [LineBaseFilter] or: [LineBaseFilter] not: LineBaseFilter } """ Входной объект: пространственный фильтр пересечения с заданным полигоном """ input IntersectsFilter { """ Полигон в пространстве, по пересечению с которым происходит фильтрация """ polygon: PolygonRef } """ Входной объект: последовательность точек в пространстве для использования в фильтрах """ input LineRef { points: [PointRef!]! } </pre>
18	type PointTimeBase	<pre> """ Входной объект для базового типа пространственно- временной информации: точки в пространстве с </pre>

		<pre> привязкой к дате и времени """ input PointTimeBaseFilter { """ Фильтр для пространственной информации """ point: PointBaseFilter """ Фильтр для временной информации """ time: DateTimeBaseFilter and: [PointTimeBaseFilter] or: [PointTimeBaseFilter] not: PointTimeBaseFilter } """ Входной объект: точки в пространстве с привязкой к дате и времени для использования в фильтрах """ input PointTimeRef { point: PointRef! time: String! } </pre>
19	type PolygonBase	<pre> """ Входной объект для базового типа пространственной информации: полигона """ input PolygonBaseFilter { """ Фильтр для полигона: нахождение около аргумента фильтра в заданных пределах """ near: NearFilter """ Фильтр для полигона: нахождение внутри заданного полигона """ within: WithinFilter """ Фильтр для полигона: содержание заданных точки или полигона """ contains: ContainsFilter """ Фильтр для полигона: пересечение с заданным полигоном """ intersects: IntersectsFilter and: [PolygonBaseFilter] or: [PolygonBaseFilter] not: PolygonBaseFilter </pre>

		<pre> } """ Входной объект: пространственный фильтр включения заданной точки или полигона """ input ContainsFilter { """ Точка в пространстве, по включению которой в исходный полигон происходит фильтрация """ point: PointRef """ Полигон в пространстве, по включению которого в исходный полигон происходит фильтрация """ polygon: PolygonRef } """ Входной объект: полигон в пространстве для использования в фильтрах """ input PolygonRef { lines: [LineRef!]! } </pre>
20	type MultiPolygonBase	<pre> """ Входной объект для базового типа пространственной информации: мультиполигона """ input MultiPolygonBaseFilter { """ Фильтр для мультиполигона: нахождение около аргумента фильтра в заданных пределах """ near: NearFilter """ Фильтр для мультиполигона: нахождение внутри заданного полигона """ within: WithinFilter """ Фильтр для мультиполигона: содержание заданных точки или полигона """ contains: ContainsFilter """ Фильтр для мультиполигона: пересечение с заданным полигоном """ intersects: IntersectsFilter and: [MultiPolygonBaseFilter] </pre>

		<pre> or: [MultiPolygonBaseFilter] not: MultiPolygonBaseFilter } </pre>
21	type GeoTrackBase	<pre> """ Входной объект для базового типа пространственно- временной информации: трек """ input GeoTrackBaseFilter { """ Фильтр для временной информации: прохождение маршрута в указанное время """ time: DateTimeBaseFilter """ Фильтр для пространственной информации: нахождение около аргумента фильтра в заданных пределах """ near: NearFilter! """ Фильтр для пространственно-временной информации: нахождение около аргумента фильтра в заданных пределах в указанное время """ nearTime: NearTimeFilter! """ Фильтр для пространственной информации: нахождение внутри заданного полигона """ within: WithinFilter! """ Фильтр для пространственно-временной информации: нахождение внутри заданного полигона в указанное время """ withinTime: WithinTimeFilter! """ Фильтр для пространственной информации: пересечение с заданным полигоном """ intersects: IntersectsFilter! """ Фильтр для пространственно-временной информации: пересечение с заданным полигоном в указанное время """ intersectsTime: IntersectsTimeFilter! """ Фильтр для пространственно-временной информации: пересечение с заданным треком в </pre>

		<p>указанных промежутках времени и расстояния """"</p> <pre>intersectsTrack: IntersectsTrackFilter! and: [GeoTrackBaseFilter] or: [GeoTrackBaseFilter] not: GeoTrackBaseFilter }</pre> <p>""""</p> <p>Входной объект: пространственно-временной фильтр нахождения около заданной точки в указанных пределах в заданное время """"</p> <pre>input NearTimeFilter { distance: Float! point: PointRef! time: DateTimeBaseFilter! }</pre> <p>""""</p> <p>Входной объект: пространственно-временной фильтр нахождения внутри заданного полигона в указанное время """"</p> <pre>input WithinTimeFilter { polygon: PolygonRef! time: DateTimeBaseFilter! }</pre> <p>""""</p> <p>Входной объект: пространственно-временной фильтр пересечения с заданным полигоном в указанное время """"</p> <pre>input IntersectsTimeFilter { polygon: PolygonRef time: DateTimeBaseFilter! }</pre> <p>""""</p> <p>Входной объект: пространственно-временной фильтр пересечения с заданным треком в указанных интервалах расстояния и времени """"</p> <pre>input IntersectsTrackFilter { geotrack: GeoTrackRef! deltaTime: DeltaTimeRef! distance: Float! }</pre> <p>""""</p> <p>Входной объект: период времени в секундах """"</p> <pre>input DeltaTimeRef { second: Int! }</pre> <p>""""</p>
--	--	--

		<p>Входной объект: трек для использования в фильтрах """" input GeoTrackRef { pointsInTime: [PointTimeRef]! }</p>
22	type BankAccountInfoBase	<p>"""" Входной объект для базового типа банковских данных """" input BankAccountInfoBaseFilter { """" Фильтр для имени банка """" bankName: StringRegExpExactFilter """" Фильтр для номера счета """" account: StringRegExpExactFilter """" Фильтр для номера корреспондентского счета """" corrAccount: StringRegExpExactFilter """" Фильтр для номера карты """" cardNumber: StringRegExpExactFilter """" Фильтр для банковского идентификационного кода """" rbcic: StringRegExpExactFilter """" Фильтр для КПП """" kpp: StringRegExpExactFilter and: [BankAccountInfoBaseFilter] or: [BankAccountInfoBaseFilter] not: BankAccountInfoBaseFilter }</p>
23	type BankTransferInfoBase	<p>"""" Входной объект для базового типа банковских переводов """" input BankTransferInfoBaseFilter { """" Фильтр для отправителя """" from: BankAccountInfoBaseFilter """" Фильтр для получателя """" to: BankAccountInfoBaseFilter }</p>

		<pre> """ Фильтр для суммы перевода """ amount: StringRegExpExactFilter """ Фильтр для даты и времени перевода """ data: DateTimeBaseFilter and: [BankTransferInfoBaseFilter] or: [BankTransferInfoBaseFilter] not: BankTransferInfoBaseFilter } </pre>
24	type OrganizationInfoBase	<pre> """ Входной объект для базового типа сведений о юридических лицах и индивидуальных предпринимателях """ input OrganizationInfoBaseFilter { """ Фильтр для полного наименования организации """ nameFull: StringRegExpExactFilter """ Фильтр для сокращенного наименования организации """ nameSmall: StringRegExpExactFilter """ Фильтр для ОГРН организации """ grn: GRNBaseFilter """ Фильтр для ИНН организации """ inn: InnBaseFilter """ Фильтр для банковских данных организации """ bankInfo: BankAccountInfoBaseFilter """ Фильтр для номера телефона организации """ msisdn: MsisdnBaseFilter """ Фильтр для web-сайта организации """ webSite: URLBaseFilter """ Фильтр для электронной почты организации """ email: EmailBaseFilter </pre>

		<pre> """" Фильтр для даты регистрации организации """" dataRegistration: DateTimeBaseFilter """" Фильтр для неструктурированного адреса организации """" address: AddressBaseFilter """" Фильтр для представителя организации """" representative: OrganizationRepresentativeBaseFilter """" Фильтр для ОКВЭД организации """" okved: StringRegExpExactFilter """" Фильтр для ЕГРЮЛ """" egrul: StringRegExpExactFilter """" Фильтр для ЕГРИП """" egrip: StringRegExpExactFilter and: [OrganizationInfoBaseFilter] or: [OrganizationInfoBaseFilter] not: OrganizationInfoBaseFilter } """" Входной объект для базового типа государственного регистрационного номера ОГРН и ОГРНИП """" input GRNBaseFilter { """" Фильтр для государственного регистрационного номера – ОГРН и ОГРНИП """" grn: StringExactFilter and: [GRNBaseFilter] or: [GRNBaseFilter] not: GRNBaseFilter } </pre>
25	type OrganizationRepresentativeBase	<pre> """" Входной объект для базового типа сведений об учредителях (участниках) юридического лица, лицах, имеющих право без доверенности действовать от имени юридического лица """" input OrganizationRepresentativeBaseFilter { """" </pre>

		<p>Фильтр для должности " "" "" ""</p> <p>position: StringRegExpExactFilter " "" "" ""</p> <p>Фильтр для ИНН лица " "" "" ""</p> <p>inn: InnBaseFilter " "" "" ""</p> <p>Фильтр для ФИО " "" "" ""</p> <p>name: PersonBaseFilter and: [OrganizationRepresentativeBaseFilter] or: [OrganizationRepresentativeBaseFilter] not: OrganizationRepresentativeBaseFilter }</p>
26	type IPAddressBase	<p>"" "" "" ""</p> <p>Входной объект для базового типа IP-адреса " "" "" ""</p> <p>input IPAddressBaseFilter { " "" "" ""</p> <p>Фильтр для IP-адреса " "" "" ""</p> <p>ip: StringRegExpExactFilter " "" "" ""</p> <p>Фильтр для маски подсети " "" "" ""</p> <p>mask: IntFilter " "" "" ""</p> <p>Фильтра для типа IP-адреса: ipv4, ipv6 " "" "" ""</p> <p>type: IpType and: [IPAddressBaseFilter] or: [IPAddressBaseFilter] not: IPAddressBaseFilter }</p>
27	type NetworkPeerBase	<p>"" "" "" ""</p> <p>Входной объект для базового типа информации об участнике сетевого соединения " "" "" ""</p> <p>input NetworkPeerBaseFilter { " "" "" ""</p> <p>Фильтр для IP-адреса " "" "" ""</p> <p>ip: IPAddressBaseFilter " "" "" ""</p> <p>Фильтр для порта " "" "" ""</p> <p>port: IntFilter " "" "" ""</p> <p>Фильтр для номера протокола " "" "" ""</p>

		<pre> protocolNumber: IntFilter and: [NetworkPeerBaseFilter] or: [NetworkPeerBaseFilter] not: NetworkPeerBaseFilter } </pre>
28	type URLBase	<pre> """ Входной объект для базового типа URL-адреса """ input URLBaseFilter { """ Фильтр для URL-адреса """ url: StringRegExpExactFilter and: [URLBaseFilter] or: [URLBaseFilter] not: URLBaseFilter } </pre>
29	type DomainNameBase	<pre> """ Входной объект для базового типа доменного имени """ input DomainNameBaseFilter { """ Фильтр для доменного имени """ domainName: StringRegExpExactFilter and: [DomainNameBaseFilter] or: [DomainNameBaseFilter] not: DomainNameBaseFilter } </pre>
30	type LoginBase	<pre> """ Входной объект для базового типа имени/идентификатора пользователя """ input LoginBaseFilter { """ Фильтр для имени/идентификатора пользователя """ login: StringRegExpExactFilter and: [LoginBaseFilter] or: [LoginBaseFilter] not: LoginBaseFilter } </pre>
31	scalar Int	<pre> """ Входной объект: используемые функции для фильтрации значений типа Int """ input IntFilter { """ Равно """ eq: Int } </pre>

		<pre> """ В [списке] """ in: [Int] """ Меньше либо равно """ le: Int """ Меньше, чем """ lt: Int """ Больше либо равно """ ge: Int """ Больше, чем """ gt: Int """ В промежутке """ between: IntRange } """ Входной объект: промежуток типа Int для использования в фильтрах """ input IntRange { min: Int! max: Int! } </pre>
32	scalar Float	<pre> """ Входной объект: используемые функции для фильтрации значений типа Float """ input FloatFilter { """ Равно """ eq: Float """ В [списке] """ in: [Float] """ Меньше либо равно """ le: Float """ </pre>

		<pre> Меньше, чем """ lt: Float """ Больше либо равно """ ge: Float """ Больше, чем """ gt: Float """ В промежутке """ between: FloatRange } """ Входной объект: промежуток типа Float для использования в фильтрах """ input FloatRange { min: Float! max: Float! } </pre>
33	scalar String	<pre> """ Входной объект: комбинация используемых функций для фильтрации значений типа String """ input StringExactFilter { """ Равно """ eq: String """ В [списке] """ in: [String] } """ Входной объект: промежуток типа String для использования в фильтрах """ input StringRange { min: String! max: String! } """ Входной объект: комбинация используемых функций для фильтрации значений типа String """ input StringFullTextFilter { </pre>

		<pre> """ Полнотекстовый поиск, соответствующий всему заданному тексту """ alloftext: String """ Полнотекстовый поиск, соответствующий любому набору из заданного текста """ anyoftext: String } """ Входной объект: комбинация используемых функций для фильтрации значений типа String """ input StringRegExpFilter { """ Регулярное выражение в формате POSIX Basic Regular Expression """ regex: String } """ Входной объект: комбинация используемых функций для фильтрации значений типа String """ input StringTermFilter { """ Соответствие строкам, содержащим все указанные термы в любом порядке; без учета регистра """ allofterms: String """ Соответствие строкам, содержащим любой из указанных термов в произвольном порядке; без учета регистра """ anyofterms: String } """ Входной объект: комбинация используемых функций для фильтрации значений типа String """ input StringRegExpExactFilter { """ Регулярное выражение в формате POSIX Basic Regular Expression """ regex: String """ Равно """ </pre>
--	--	--

		<pre> eq: String """ В [списке] """ in: [String] } """ Входной объект: комбинация используемых функций для фильтрации значений типа String """ input StringRegExpFullTextFilter { """ Регулярное выражение в формате POSIX Basic Regular Expression """ regex: String """ Полнотекстовый поиск, соответствующий всему заданному тексту """ alloftext: String """ Полнотекстовый поиск, соответствующий любому набору из заданного текста """ anyoftext: String } """ Входной объект: комбинация используемых функции для фильтрации значений типа String """ input StringTermExactFilter { allofterms: String """ Соответствие строкам, содержащим любой из указанных термов в произвольном порядке; без учета регистра """ anyofterms: String """ Равно """ eq: String """ В [списке] """ in: [String] } """ Входной объект: комбинация используемых функций для фильтрации значений типа String </pre>
--	--	---

		<pre>"""" input StringTermFullTextFilter { """" Соответствие строкам, содержащим все указанные термы в любом порядке; без учета регистра """" allofterms: String """" Соответствие строкам, содержащим любой из указанных термов в произвольном порядке; без учета регистра """" anyofterms: String """" Полнотекстовый поиск, соответствующий всему заданному тексту """" alloftext: String """" Полнотекстовый поиск, соответствующий любому набору из заданного текста """" anyoftext: String }</pre>
--	--	--

Приложение № 7
к требованиям к вычислительной мощности,
используемой провайдером хостинга, для
проведения уполномоченными
государственными органами, осуществляющими
оперативно-разыскную деятельность или
обеспечение безопасности Российской
Федерации, в случаях, установленных
федеральными законами, мероприятий в целях
реализации возложенных на них задач,
утвержденным приказом Министерства
цифрового развития, связи и массовых
коммуникаций Российской Федерации
от 1 ноября 2023 г. № 935

Требования, предъявляемые к формату передачи данных на языке GraphQL по протоколу WebSocket

1. ТС ОРМ и ПУ для передачи данных на языке GraphQL по протоколу WebSocket должны использовать подпротокол graphql-transport-ws.
2. ТС ОРМ и ПУ взаимодействуют по протоколу WebSocket посредством сообщений. Для кодирования сообщений применяется формат JSON.
3. Все сообщения должны содержать поле «type», определяющее тип сообщения. Сообщение в зависимости от типа также может содержать дополнительные поля, в которых:
 - 1) «id» – идентификатор запроса, использующийся для последующего определения ответов ТС ОРМ и привязки их к запросам ПУ;
 - 2) «payload» – дополнительные данные для некоторых типов сообщений.
4. В любой момент времени ТС ОРМ должны позволять выполнять несколько запросов ПУ с уникальными идентификаторами. При этом передача сообщений, относящихся к различным запросам, может чередоваться.
5. ТС ОРМ должны закрывать соединение по протоколу WebSocket при возникновении критической ошибки (сбое в работе) и тем самым уведомлять ПУ о данной ситуации.
6. ПУ должен закрывать соединение по протоколу WebSocket с кодом 1000 и причиной закрытия «Normal Closure».
7. ПУ перед отправкой запросов на языке GraphQL должен открыть сессию (согласно подпункту 1 пункта 8 настоящего приложения) поверх протокола WebSocket.
8. ТС ОРМ и ПУ взаимодействуют по протоколу WebSocket с использованием следующих типов сообщений:
 - 1) сообщение «ConnectionInit». Данное сообщение направляется ПУ в адрес ТС ОРМ для открытия сессии. ПУ может передавать дополнительную

информацию о сессии в поле «payload». ТС ОРМ должны получить данное сообщение в течение заданного периода времени (определяется во время внедрения ТС ОРМ и согласовывается с уполномоченным государственным органом). В случае не поступления от ПУ в ТС ОРМ данного сообщения в течение заданного периода времени, ТС ОРМ должны закрыть соединение по протоколу WebSocket с кодом 4408 и причиной закрытия «Connection initialization timeout». В случае получения ТС ОРМ более одного сообщения «ConnectionInit», ТС ОРМ должны закрыть соединение по протоколу WebSocket с кодом 4429 и причиной закрытия «Too many initialization requests».

Формат сообщения «ConnectionInit» (поле «payload» опциональное):

```
{
  "type": "connection_init",
  "payload": "string"
}
```

;

2) сообщение «ConnectionAck». Данное сообщение подтверждает успешное создание сессии и направляется ТС ОРМ в адрес ПУ в ответ на сообщение «ConnectionInit». ТС ОРМ могут передавать дополнительную информацию о сессии в поле «payload». ПУ после получения данного сообщения может отправлять запросы на языке GraphQL с помощью сообщений «Subscribe» согласно подпункту 3 пункта 8 настоящего приложения.

Формат сообщения «ConnectionAck» (поле «payload» опциональное):

```
{
  "type": "connection_ack",
  "payload": "string"
}
```

;

3) сообщение «Subscribe». Данное сообщение направляется ПУ в адрес ТС ОРМ и должно содержать запрос на языке GraphQL в поле «query» поля «payload». Поле «id» должно содержать уникальный идентификатор запроса, формируемый ПУ. Если запрос с таким идентификатором уже выполняется, то ТС ОРМ должны закрыть соединение по протоколу WebSocket с кодом 4409 и причиной закрытия «Subscriber for <unique-operation-id> already exists». Разрешается повторное использование идентификатора после завершения выполнения запроса на стороне ТС ОРМ. В случае поступления от ПУ данного сообщения до получения сообщения «ConnectionAck», ТС ОРМ должны закрыть соединение по протоколу WebSocket с кодом 4401 и причиной закрытия «Unauthorized».

Формат сообщения «Subscribe» (поля «operationName», «variables», «extensions» опциональные):

```
{
  "id": "<unique-operation-id>",
  "type": "subscribe",
  "payload": {
    "operationName": "string",
    "query": "string",

```

```

    "variables": "string",
    "extensions": "string"
  }
}

```

4) сообщение «Next». Данное сообщение направляется ТС ОРМ в адрес ПУ и содержит полные или частичные результаты выполнения запроса, ранее направленного ПУ в сообщении «Subscribe». Поле «id» должно содержать уникальный идентификатор запроса, ранее сформированный ПУ. Поле «payload» должно содержать результаты выполнения запроса в формате аналогичном для интерфейса «/query» согласно спецификации языка GraphQL.

Формат сообщения «Next:

```

{
  "id": "<unique-operation-id>",
  "type": "next",
  "payload": "ExecutionResult"
}

```

5) сообщение «Error». Данное сообщение направляется ТС ОРМ в адрес ПУ и содержит информацию об ошибке в ходе выполнения запроса, ранее направленного ПУ в сообщении «Subscribe». Поле «id» должно содержать уникальный идентификатор запроса, ранее сформированный ПУ. Поле «payload» должно содержать информацию об ошибке в формате согласно спецификации языка GraphQL. В случае возникновения ошибки на начальном этапе выполнения запроса (в том числе, ошибки валидации запроса) либо в ходе выполнения запроса ТС ОРМ прекращают выполнение запроса и не направляют в сторону ПУ никаких сообщений.

Формат сообщения «Error»:

```

{
  "id": "<unique-operation-id>",
  "type": "error",
  "payload": "GraphQLError"
}

```

6) сообщение «Complete». Данное сообщение может направляться в обоих направлениях. Поле «id» должно содержать уникальный идентификатор запроса, ранее сформированный ПУ. Отправка данного сообщения ТС ОРМ в адрес ПУ означает завершение выполнения запроса с соответствующим идентификатором. В случае отправки ТС ОРМ сообщения «Error» до сообщения «Complete», данное сообщение не отправляется. Отправка данного сообщения ПУ в адрес ТС ОРМ означает, что ПУ прерывает выполнение запроса с соответствующим идентификатором и с данного момента не принимает результаты его выполнения.

Формат сообщения «Complete:

```

{
  "id": "<unique-operation-id>",
  "type": "complete"
}

```

9. Получение ПУ или ТС ОРМ иных сообщений, отличных от указанных в пункте 8 настоящего приложения, должно приводить к закрытию соединения по протоколу WebSocket с кодом 4400 и причиной закрытия, содержащей описание ошибки в типе или формате сообщения.

Приложение № 8
к требованиям к вычислительной мощности,
используемой провайдером хостинга, для
проведения уполномоченными
государственными органами, осуществляющими
оперативно-разыскную деятельность или
обеспечение безопасности Российской
Федерации, в случаях, установленных
федеральными законами, мероприятий в целях
реализации возложенных на них задач,
утвержденным приказом Министерства
цифрового развития, связи и массовых
коммуникаций Российской Федерации
от 1 ноября 2023 г. № 935

Требования, предъявляемые к запросам мониторинга и формату их передачи по интерфейсу взаимодействия между пунктом управления и техническими средствами, обеспечивающими выполнение установленных действий при проведении оперативно-разыскных мероприятий

1. ТС ОРМ должны предусматривать возможность получения ПУ следующей информации о структуре и функционировании ТС ОРМ по запросу ПУ:

1) о структуре и составе ТС ОРМ, составе и состоянии интерфейса взаимодействия ТС ОРМ с ПУ;

2) об установленном в ТС ОРМ общесистемном ПО, перечне и состоянии программных модулей в составе ПО ТС ОРМ;

3) о точках подключения ТС ОРМ к ИС и интерфейсах ввода информации в ТС ОРМ.

2. ТС ОРМ по запросу ПУ должны представлять следующую информацию в части структуры и состава ТС ОРМ, состава и состояния интерфейса взаимодействия ТС ОРМ с ПУ:

1) перечень коммутационного и серверного оборудования, средств хранения данных с его идентификацией;

2) идентификацию интерфейсов подключения оборудования ТС ОРМ друг к другу;

3) параметры для серверного оборудования:

общий и занятый объем оперативной памяти;

количество сетевых интерфейсов с их идентификацией, нагрузку;

общее количество процессоров, загрузку;

общий объем дискового пространства, объем свободного пространства;

4) параметры технических средств хранения данных:

перечень модулей, составляющих средства хранения данных с их идентификацией;

для каждого входящего в состав средств хранения данных модуля: общий объем дискового пространства, объем свободного дискового пространства и состояние модуля (функционирует штатно, произошел сбой, модуль не функционирует), текстовую расшифровку сбоя.

3. ТС ОРМ по запросу ПУ должны представлять информацию в части точек подключения ТС ОРМ к ИС, интерфейсов ввода информации в ТС ОРМ, содержащую:

1) перечень точек подключения к ИС провайдера хостинга и точек ввода информации в ТС ОРМ с их идентификацией;

2) для каждой точки подключения предоставляет информацию о: состоянии точки подключения (ввода информации) (функционирует штатно, произошел сбой, модуль не функционирует), текстовую расшифровку сбоя; объеме информации, поступающей в секунду; периоде времени, в течение которого на точку подключения и (или) ввода информации в ТС ОРМ не поступала информация.

4. ТС ОРМ по запросу ПУ должны представлять следующую информацию в части состава ПО ТС ОРМ и его состояния:

1) перечень установленного общесистемного ПО с его идентификацией;

2) информацию для общесистемного ПО:

идентификатор ТС ОРМ, на котором установлено ПО;

наименование ПО;

состояние ПО (функционирует штатно, произошел сбой, модуль не функционирует), текстовую расшифровку сбоя;

3) перечень установленного ПО ТС ОРМ с его идентификацией;

4) информацию для ПО ТС ОРМ:

идентификатор ТС ОРМ, на котором установлено ПО;

назначение (определяется разработчиком ТС ОРМ);

состояние (функционирует штатно, произошел сбой, модуль не функционирует), текстовую расшифровку сбоя;

список контролируемых параметров.

5. Информация о функционировании и состоянии ТС ОРМ, указанная в пунктах 1 – 4 настоящего приложения, должна быть представлена в виде временных рядов (далее – метрик).

6. Метрики должны иметь следующий обязательный набор меток (пара ключ-значение), в которых:

`systemtype` – относится ли метрика к оборудованию или ПО. Может принимать одно из значений «hardware» или «software»;

`moduleid` – идентификатор модуля;

`modulename` – наименование модуля;

`moduletype` – тип модуля;

`blocknumber` – номер блока оборудования;

`groupname` – наименование группы метрик;

`softwareparentmoduleid` – идентификатор модуля ПО, в состав которого входит модуль с данной метрикой;

hardwareparentmoduleid – идентификатор модуля оборудования, в состав которого входит модуль с данной метрикой.

7. Для метрик также должны быть заданы единицы измерения и описания их назначения на русском языке.

8. Для получения значений метрик ПУ направляет ТС ОРМ запросы мониторинга на интерфейс «/metric».

9. Для кодирования содержимого запросов мониторинга и ответов ТС ОРМ должен применяться формат JSON.

10. ТС ОРМ при успешном выполнении запросов мониторинга должны возвращать HTTP код 2xx.

11. ТС ОРМ при возникновении ошибки во время выполнения запросов мониторинга должны в ответе возвращать поле «error» с информацией об ошибке и следующие HTTP коды:

400 Bad Request – в запросе отсутствуют или некорректно заданы параметры;

422 Unprocessable Entity – неправильный формат запроса на языке PromQL согласно пункту 13 настоящего приложения;

503 Service Unavailable – выполнение запроса прервано на стороне ТС ОРМ.

12. Ответы на запросы мониторинга должны иметь следующий формат:

```
{
  "status": "success" | "error",
  "data": <data>,

  "errorType": "<string>",
  "error": "<string>",

  "warnings": [<string>]
}
```

;

в котором:

1) поле «status» – статус выполнения запроса, может принимать одно из двух значений: «success» или «error»;

2) поле «data» – результат выполнения запроса, структура поля зависит от интерфейса, на которую был отправлен запрос (согласно пункту 13 настоящего приложения);

3) поля «errorType» и «error» – соответственно тип и описание ошибки, возникшей при выполнении запроса. Данные поля должны быть заданы только, если поле «status» имеет значение «error»;

4) поле «warnings» – незначительные ошибки (предупреждения), возникшие во время выполнения запроса, но существенно не повлиявшие на его выполнение. При этом поле «data» может содержать частичные результаты выполнения запроса.

13. ТС ОРМ должны реализовывать внутри интерфейса «/metric» следующие дополнительные конечные точки:

1) «/api/v1/query» – предназначена для приема запросов мониторинга от ПУ в целях получения метрик для конкретного значения времени, а также

передачи на ПУ результатов выполнения данных запросов. Данная интерфейс должна поддерживать HTTP методы GET и POST.

Принимаемые параметры:

«query» – запрос на языке PromQL;

«time» (опционально) – дата и время в формате согласно RFC3339 или Unix-время;

«duration» (опционально) – период времени в формате согласно Time Durations языка PromQL.

Поле «data» для ответа на запросы мониторинга для данной интерфейса должно иметь следующий вид:

```
{
  "resultType": "matrix" | "vector" | "scalar" | "string",
  "result": <value>
}
```

Формат поля «result» зависит от значения поля «resultType» согласно пункту 14 настоящего приложения;

2) «/api/v1/query_range» – предназначена для приема запросов мониторинга от ПУ в целях получения метрик для диапазона времени, а также передачи на ПУ результатов выполнения данных запросов. Данная интерфейс должна поддерживать HTTP методы GET и POST.

Принимаемые параметры:

«query» – запрос на языке PromQL;

«start» – дата и время начала диапазона в формате согласно RFC3339 или Unix-время;

«end» – дата и время конца диапазона в формате согласно RFC3339 или Unix-время;

«step» – временной шаг запроса в формате согласно Time Durations языка PromQL или число с плавающей точкой;

«duration» (опционально) – период времени в формате согласно Time Durations языка PromQL.

Поле «data» для ответа на запросы мониторинга для данной интерфейса должно иметь следующий вид:

```
{
  "resultType": "matrix",
  "result": <value>
}
```

Формат поля «result» представлен в подпункте 1 пункта 14 настоящего приложения;

3) «/api/v1/series» – предназначена для приема запросов мониторинга от ПУ в целях получения списка метрик, которые имеют заданный набор меток, а также передачи на ПУ результатов выполнения данных запросов. Данная интерфейс должна поддерживать HTTP методы GET и POST.

Принимаемые параметры:

«match[]» – перечень запрашиваемых меток в формате Time series Selectors языка PromQL;

«start» – дата и время начала диапазона в формате согласно RFC3339 или Unix-время;

«end» – дата и время конца диапазона в формате согласно RFC3339 или Unix-время.

Поле «data» для ответа на запросы мониторинга для данной интерфейса должно содержать перечень метрик (представляются перечнем их меток (пара ключ-значение)), удовлетворяющих критериям запроса;

4) «/api/v1/labels» – предназначена для приема запросов мониторинга от ПУ в целях получения списка меток, которые имеют заданный набор метрик, а также передачи на ПУ результатов выполнения данных запросов. Данная интерфейс должна поддерживать HTTP методы GET и POST.

Принимаемые параметры:

«match[]» (опционально) – перечень запрашиваемых метрик в формате Time series Selectors языка PromQL;

«start» (опционально) – дата и время начала диапазона в формате согласно RFC3339 или Unix-время;

«end» (опционально) – дата и время конца диапазона в формате согласно RFC3339 или Unix-время.

Поле «data» для ответа на запросы мониторинга для данной интерфейса должно содержать перечень названий меток, удовлетворяющих критериям запроса;

5) «/api/v1/label/<label_name>/values» – предназначена для приема запросов мониторинга от ПУ в целях получения значений заданных меток, а также передачи на ПУ результатов выполнения данных запросов. Данная интерфейс должна поддерживать HTTP метод GET.

Принимаемые параметры:

«match[]» (опционально) – перечень запрашиваемых метрик в формате Time series Selectors языка PromQL;

«start» (опционально) – дата и время начала диапазона в формате согласно RFC3339 или Unix-время;

«end» (опционально) – дата и время конца диапазона в формате согласно RFC3339 или Unix-время.

Поле «data» для ответа на запросы мониторинга для данной интерфейса должно содержать перечень значений меток, удовлетворяющих критериям запроса.

14. Поле «result» в ответах на запросы мониторинга может иметь один из следующих форматов:

1) если поле «resultType» имеет значение «matrix», то поле «result» должно иметь следующий формат:

```
[
  {
    "metric": {"<label_name>": "<label_value>", ... },
    "values": [[<unix_time>, "<sample_value>"], ... ],
    "histograms": [[<unix_time>, <histogram>], ... ]
  }
]
```

```

    },
    ...
]

```

;

2) если поле «resultType» имеет значение «vector», то поле «result» должно иметь следующий формат:

```

[
  {
    "metric": {"<label_name>": "<label_value>", ... },
    "value": [<unix_time>, "<sample_value>"],
    "histogram": [<unix_time>, <histogram>],
  },
  ...
]

```

3) если поле «resultType» имеет значение «scalar», то поле «result» должно иметь следующий формат:

```
[<unix_time>, "<scalar_value>"]
```

;

4) если поле «resultType» имеет значение «string», то поле «result» должно иметь следующий формат:

```
[<unix_time>, "<string_value>"]
```