



МИНИСТЕРСТВО ЮСТИЦИИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ЗАРЕГИСТРИРОВАНО

Регистрационный № 86587
от 22 мая 2025 г.

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

(МИНЦИФРЫ РОССИИ)

ПРИКАЗ

16.12.2025

№ 1174

Москва

Об утверждении Требований к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», 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 подпунктом 3 пункта 9 статьи 56.2 Федерального закона от 7 июля 2003 г. № 126-ФЗ «О связи», подпунктом 5.2.55 пункта 5 Положения о Министерстве цифрового развития, связи и массовых коммуникаций Российской Федерации, утвержденного постановлением Правительства Российской Федерации от 2 июня 2008 г. № 418, п р и к а з ы в а ю :

1. Утвердить прилагаемые Требования к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», 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.

2. Признать утратившим силу приказ Минкомсвязи России от 5 ноября 2019 г. № 646 «Об утверждении Требований к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих номер автономной системы, 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»

(зарегистрирован Министерством юстиции Российской Федерации 22 января 2020 г., регистрационный № 57223).

Министр

A handwritten signature in black ink, appearing to read 'М.И. Шадаев', written in a cursive style.

М.И. Шадаев

УТВЕРЖДЕНЫ
приказом Министерства
цифрового развития, связи
и массовых коммуникаций Российской
Федерации
от 16.12.2025 № 1174

ТРЕБОВАНИЯ

к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», 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. Настоящие Требования к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-коммуникационной сети «Интернет», 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 (далее – Требования к ПТС), устанавливают правила применения программных и технических средств, используемых собственниками или иными владельцами технологических сетей связи, имеющими уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), в эксплуатируемых им информационных системах (далее – ИС).

2. При реализации программных и технических средств, используемых собственниками или иными владельцами технологических сетей связи, имеющими уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет» (далее – ПТС), в эксплуатируемых им ИС должен быть использован один из вариантов их построения:

- а) отдельный аппаратно-программный комплекс;
- б) отдельный программный модуль в ИС;

в) комбинированный вариант, предусматривающий совместное использование элементов в соответствии с подпунктами «а» и «б» настоящего пункта.

3. Для каждого варианта построения собственниками или иными владельцами технологических сетей связи, имеющими уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет», должны реализовываться требования по информационной безопасности и защите от несанкционированного доступа к информации, связанной с проведением оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации, в соответствии с подпунктами «з» и «и» пункта 16 Требований к ПТС. ПТС должны входить в состав узлов связи технологических сетей связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет».

II. Функциональные требования к ПТС

4. Посредством использования ПТС осуществляются поиск, обработка и передача на пункт управления уполномоченных государственных органов, осуществляющих оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации (далее – ПУ), по их запросу или в автоматическом режиме информации, хранящейся в ИС и передаваемой по технологическим сетям связи.

5. Перечень доступной для поиска информации, хранящейся в ИС и передаваемой по технологическим сетям связи, и схема ее представления должны согласовываться с уполномоченными государственными органами, осуществляющими оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации.

6. Взаимодействие ПТС с ПУ должно осуществляться по единому каналу передачи данных.

7. Пропускная способность канала передачи данных между ПТС 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 ПТС 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

8. Посредством использования ПТС обеспечивается:

- а) подключение к ПУ в соответствии с пунктом 10 Правил взаимодействия собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», с уполномоченными государственными органами, осуществляющими оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации, утвержденных постановлением Правительства Российской Федерации от 29 октября 2019 г. № 1385;
- б) реализация интерфейса взаимодействия между ПТС и ПУ в соответствии с приложением № 1 к Требованиям к ПТС;
- в) прием от ПУ поисковых запросов и запросов постановки объектов на контроль;
- г) поиск информации, хранящейся в ИС, в соответствии с поступившими с ПУ поисковыми запросами и запросами постановки объектов на контроль. Для ПТС, осуществляющих поиск информации в ИС АС со среднесуточным объемом новых данных свыше 1 Гбайт, устанавливаются требования к критериям поиска в соответствии с пунктом 21 приложения № 1 к Требованиям к ПТС;
- д) передача на ПУ от ПТС данных в соответствии с поступившими с ПУ поисковыми запросами (допускается страничная передача) и запросами постановки объектов на контроль;
- е) прием от ПУ критериев поиска и передача на ПУ статистической, текстовой, мультимедийной, звуковой, графической информации в исходном (декодированном) виде, хранящейся в ИС и отбираемой по критериям поиска, без дополнительной обработки (далее – неформатированные данные);
- ж) предоставление информации, хранящейся в схеме данных, при выполнении запросов от ПУ согласно приложению № 2 к Требованиям к ПТС;
- з) описание схемы данных в соответствии с перечнем базовых типов согласно приложению № 3 к Требованиям к ПТС;
- и) передача на ПУ схемы данных в соответствии с поступившим с ПУ запросом согласно приложению № 4 к Требованиям к ПТС;
- к) задание параметров поиска в соответствии с перечнем входных объектов согласно приложению № 5 к Требованиям к ПТС;
- л) защита от несанкционированного доступа со стороны производителей ПТС, неавторизованных пользователей, технического персонала, третьих лиц к хранящейся в ПТС информации и информации, непосредственно связанной с проведением уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации;
- м) информирование ПУ о попытках несанкционированного доступа к ПТС,

если ПТС исполнен в виде отдельного аппаратно-программного комплекса:

- к данным, связанным с проведением уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации с использованием команд или сервисных программ;

- о резервном копировании данных, связанных с проведением уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации;

- к ПТС через интерфейсы, не предусмотренные для доступа к ним;

- о вскрытии корпуса технических средств, с использованием которых обеспечивается проведение уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечение безопасности Российской Федерации;

н) информирование ПУ о попытках несанкционированного доступа к ПТС, если ПТС исполнен в виде отдельного программного модуля в ИС:

- к данным, связанным с проведением уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации, с использованием команд или сервисных программ;

- о резервном копировании данных, связанных с проведением уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации;

о) информирование ПУ о попытках несанкционированного доступа к ПТС, если ПТС имеет комбинированный вариант исполнения:

- к данным, связанным с проведением уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации, с использованием команд или сервисных программ;

- о резервном копировании данных, связанных с проведением уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации;

- к ПТС через интерфейсы, не предусмотренные для доступа к ним;

- о вскрытии корпуса технических средств, с использованием которых обеспечивается проведение уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечение безопасности Российской Федерации;

- о контроле работоспособности и загруженности ПТС;

- о контроле за соблюдением предоставленных прав доступа к хранящейся в ПТС информации;

- о круглосуточном удаленном доступе со стороны операторов ПУ к хранящейся в ИС информации по интерфейсу взаимодействия ПУ и ПТС в соответствии с приложением № 1 к Требованиям к ПТС;

п) ведение в автоматическом режиме системных журналов, содержащих информацию о работе ПТС и данные, связанные с проведением уполномоченными государственными органами оперативно-разыскных мероприятий или обеспечением безопасности Российской Федерации, включая сведения:

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

- о фактах получения и выполнения поисковых запросов и запросов постановки

объектов на контроль ПУ (только идентификаторы запросов, время получения и (или) выполнения, результат выполнения (успешно или неуспешно));

об ответах на поисковые запросы и запросы постановки объектов на контроль ПУ;

о текущей конфигурации ПТС, системного и прикладного ПО;

об изменениях в конфигурации ПТС, системного и прикладного ПО;

о сбоях в работе ПТС, системном и прикладном ПО;

об изменениях схемы данных;

о доступе персонала к программному обеспечению и оборудованию ПТС;

о сохранности и доступности для дальнейшего использования ранее накопленных данных при модернизации ПТС;

об автоматическом определении способа выполнения поступивших с ПУ поисковых запросов (в режиме реального времени или в отложенном режиме) в соответствии с пунктом 20 приложения № 1 к Требованиям к ПТС в соответствии с заданными временными параметрами;

о временном хранении результатов выполнения поисковых запросов в отложенном режиме согласно пункту 12 Требований к ПТС.

9. Функционирование ПТС не должно оказывать влияние на работоспособность ИС и технологические сети связи.

10. ПТС не должны иметь иных интерфейсов взаимодействия, кроме интерфейсов взаимодействия с ИС и ПУ.

11. ПТС, применяемые в распределенных технологических сетях связи и (или) нескольких технологических сетях связи, должны иметь функционал выполнения поисковых запросов для тех технологических сетей связи, которые заданы с ПУ в запросе.

III. Требования, предъявляемые к ПТС в части обеспечения временных характеристик обработки запроса и поиска информации

12. Максимальное время предварительной обработки информации с момента ее поступления в ПТС до момента, когда она становится доступной для выполнения поисковых запросов с ПУ, а также максимальное время выполнения поисковых запросов и запросов постановки объектов на контроль определяются в процессе внедрения ПТС в сети связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет». В процессе внедрения ПТС в сети связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет» должны быть установлены временные параметры для определения способа выполнения поисковых запросов (в режиме реального времени или в отложенном режиме в соответствии с пунктом 20 приложения № 1 к Требованиям к ПТС) и максимальное время хранения результатов выполнения поисковых запросов в отложенном режиме. Мероприятие по определению указанных временных параметров должно включаться в план

мероприятий по внедрению ПТС и согласовываться с уполномоченными государственными органами, осуществляющими оперативно-разыскные мероприятия или обеспечение безопасности Российской Федерации, в соответствии с постановлением Правительства Российской Федерации от 29.10.2019 № 1385 «Об утверждении Правил взаимодействия собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», с уполномоченными государственными органами, осуществляющими оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации».

13. Временные параметры, указанные в пункте 12 Требований к ПТС, могут корректироваться собственниками или иными владельцами технологических сетей связи, имеющими уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет», во время эксплуатации ПТС по согласованию с уполномоченными государственными органами, осуществляющими оперативно-разыскные мероприятия или обеспечение безопасности Российской Федерации.

Приложение № 1
к Требованиям к сетям и средствам связи
собственников или иных владельцев
технологических сетей связи, имеющих
уникальный идентификатор совокупности
средств связи и иных технических средств
в информационно-телекоммуникационной
сети «Интернет», 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 Минцифры России
от 16 декабря 2025 г. № 1174

**ТРЕБОВАНИЯ,
предъявляемые к интерфейсу взаимодействия
между пунктами управления уполномоченных государственных органов,
осуществляющих оперативно-разыскную деятельность или обеспечение
безопасности Российской Федерации, и программными и техническими
средствами, используемыми собственниками или иными владельцами
технологических сетей связи, имеющими уникальный идентификатор
совокупности средств связи и иных технических средств в информационно-
телекоммуникационной сети «Интернет»**

1. ПТС подключаются к ПУ в точках подключения выделенных каналов связи.
2. Для передачи данных между ПУ и ПТС должен использоваться набор протоколов TCP/IP.
3. Для организации линий (каналов) связи, соединяющих ПТС и ПУ, должен быть использован сетевой интерфейс физического уровня протокола взаимодействия открытых систем.
4. Посредством использования ПТС обеспечивается возможность создания виртуальной сети VPN (Virtual Private Network) между ПТС и ПУ.
5. Посредством использования ПТС обеспечивается взаимодействие с оборудованием ПУ на транспортном уровне протокола взаимодействия открытых систем (ПТС должен являться сервером).
6. При построении ПТС должны соблюдаться согласованные с уполномоченными государственными органами, осуществляющими оперативно-разыскные мероприятия или обеспечение безопасности Российской Федерации, способы защиты линий (каналов) связи. Соединение ПУ и ПТС должно устанавливаться по протоколу TLS. При установлении соединения ПУ и ПТС

должны осуществлять взаимную аутентификацию. В случае невозможности аутентифицировать ПУ или ПТС ТСП-соединение разрывается автоматически. Для аутентификации ПУ и ПТС используются сертификаты в формате международного стандарта ITU X.509 для инфраструктуры открытого ключа.

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

- 1) от ПУ в ПТС поисковых запросов и запросов постановки объектов на контроль;
- 2) от ПТС на ПУ результатов выполнения поисковых запросов и запросов постановки объектов на контроль;
- 3) от ПУ в ПТС запросов на получение неформатированных данных (далее – запросы неформатированных данных);
- 4) от ПТС на ПУ ответов на запросы неформатированных данных;
- 5) от ПУ в ПТС запросов для получения схемы данных, статуса поисковых запросов в отложенном режиме согласно пункту 20 настоящего приложения к Требованиям к ПТС и сигналов о функционировании ПТС в соответствии с пунктом 29 настоящего приложения к Требованиям к ПТС;
- 6) от ПТС на ПУ текущей схемы данных, статуса поисковых запросов в отложенном режиме и сигналов о функционировании ПТС;
- 7) от ПУ в ПТС запросов о текущей конфигурации оборудования, системного и прикладного ПО ПТС, а также их состоянии (далее – запросы мониторинга);
- 8) от ПТС на ПУ ответов на запросы мониторинга.

8. Единый канал передачи данных создается для подключения к ПТС в виде ТСП-соединений. Посредством использования ПТС должен выполняться мониторинг данного порта для создания ТСП-соединений с ПУ.

9. ПУ должен осуществлять попытки установления подключения к ПТС в соответствии с задаваемым интервалом по предоставленному ПТС ТСП-порту.

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

11. В ПТС должны поддерживаться четыре интерфейса (endpoints) взаимодействия с ПУ:

- 1) «/query» – для получения от ПУ поисковых запросов и специальных запросов 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 в соответствии с пунктом 20 настоящего приложения к Требованиям к ПТС;
- 2) «/subscription» – для получения от ПУ запросов постановки объектов на контроль и передачи на ПУ результатов их выполнения, а также для передачи на ПУ статуса выполнения поисковых запросов ПТС, которые выполняются в отложенном режиме согласно пункту 20 настоящего приложения к Требованиям к ПТС, и сигналов о функционировании ПТС согласно пункту 29 настоящего приложения к Требованиям к ПТС;
- 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 конечной точки «/subscription» – по протоколу WebSocket. ПУ и ПТС для конечной точки «/subscription» должны поддерживать сетевое соединение с помощью механизмов протокола WebSocket.

13. Собственники или иные владельцы технологических сетей связи, имеющие уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет», обязаны реализовать механизм запросов и ответов в ПТС для конечных интерфейсов «/query» и «/subscription» с использованием версии языка описания данных и запросов GraphQL. Для кодирования содержимого запросов и ответов необходимо применять формат JSON.

14. Посредством использования ПТС должен обеспечиваться функционал передачи данных на языке GraphQL по протоколу WebSocket до конечной точки «/subscription». В соответствии с требованиями, предъявляемыми к формату передачи данных на языке GraphQL по протоколу WebSocket для конечной точки «/subscription», приведенными в приложении № 6 к Требованиям к ПТС.

15. Посредством использования ПТС должен обеспечиваться функционал передачи данных до конечной точки «/metric». В соответствии с требованиями, предъявляемыми к формату передачи данных для конечной точки «/metric», приведенными в приложении № 7 к Требованиям к ПТС.

16. Собственники или иные владельцы технологических сетей связи, имеющие уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет», обязаны реализовать механизм запросов и ответов в ПТС для конечной точки «/download» с использованием протокола HTTP 1.1. ПТС для данной конечной точки должны поддерживать механизм передачи данных по частям и механизм частичных запросов.

17. Если результаты выполнения поисковых запросов и запросов постановки объектов на контроль для конечных точек «/query» и «/subscription» содержат неформатированные данные (файлы), то ответ ПТС должен содержать вместо непосредственно неформатированных данных (файлов) ссылки (HTTP URI в полном формате согласно RFC7230 `http(s)://<IP-адрес или доменное имя ПТС>:<порт>/download/<уникальный путь к файлу>`) для каждого файла для конечной точки «/download». В ПТС должна поддерживаться функциональная возможность автоматического формирования уникальной ссылки для каждого файла.

18. Для получения неформатированных данных, ссылки на которые содержатся в ответе ПТС на поисковый запрос, посредством использования ПУ для каждого файла автоматически инициируется запрос на конечную точку «/download» отдельные (от поисковых запросов) запросы получения неформатированных данных (HTTP GET-запросы) в ПТС, содержащие ссылку на запрашиваемые данные.

19. Для отправления на ПУ текущей схемы данных посредством использования ПТС необходимо получить запрос «getSchema» согласно приложению № 4 к Требованиям к ПТС на конечную точку «/query». В ответ посредством использования ПТС автоматически направляется схема данных, описанная на языке SDL GraphQL.

20. Поисковые запросы должны выполняться в ПТС в двух режимах: режиме реального времени и отложенном режиме (далее – отложенный поисковый запрос). Режим выполнения поисковых запросов определяется ПТС в соответствии с установленными временными параметрами согласно пункту 12 Требований к ПТС и параметрами запроса (количество запрашиваемых полей и критериев поиска), количества данных, хранящихся в ИС, и производительностью ПТС. Посредством использования ПТС принимаются все поисковые запросы на конечную точку «/query».

21. В ПТС должна поддерживаться функциональная возможность выполнения поисковых запросов только с критериями поиска, заданными на верхнем уровне запроса (без вложенных объектов «and», «or», «not»), включая базовые фильтры, согласно пункту 8 приложения № 2 к Требованиям к ПТС при следующих условиях:

по согласованию с уполномоченными государственными органами, осуществляющими оперативно-разыскные мероприятия или обеспечение безопасности Российской Федерации в соответствии с правилами взаимодействия, если среднесуточный объем новых данных, поступающих в ИС, составляет от 1 до 10 Гбайт;

без согласования с уполномоченными государственными органами, осуществляющими оперативно-разыскные мероприятия или обеспечение безопасности Российской Федерации, если среднесуточный объем новых данных, поступающих в ИС АС, превышает 10 Гбайт.

22. В ПТС до получения поисковых запросов с ПУ должно обеспечиваться выполнение запроса «statusOfflineRequest» (запрос «Subscription» языка GraphQL) через интерфейс «/subscription» по протоколу WebSocket в соответствии с приложением № 4 к Требованиям к ПТС.

23. В режиме реального времени посредством использования ПТС должно автоматически поддерживаться сетевое соединение, по которому поступил запрос от ПУ, и результаты выполнения запроса должны передаваться от ПТС на ПУ по тому же сетевому соединению, не позднее установленных временных параметров согласно пункту 22 настоящего приложения к Требованиям к ПТС.

24. Посредством использования ПТС обеспечивается выполнение отложенного поискового запроса по следующему сценарию:

- 1) получение от ПУ поискового запроса на конечную точку «/query»;
- 2) автоматическое определение способа выполнения запроса;
- 3) автоматическая генерация уникального идентификатора для поискового запроса в рамках данной ПТС;
- 4) автоматический возврат на ПУ идентификатора отложенного поискового запроса и информации о том, что запрос выполняется в отложенном режиме;
- 5) отправка в непрерывном режиме уведомления на ПУ (путем ответа на запрос «statusOfflineRequest») на интерфейс «/subscription» по протоколу WebSocket согласно пункту 22 настоящего приложения к Требованиям к ПТС об изменениях статуса выполнения отложенного поискового запроса.

Посредством использования ПТС должны поддерживаться 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 сервером;

«CANCELED» – запрос отменен со стороны ПУ;

6) направление на ПУ уведомления о завершении выполнения отложенного поискового запроса (статус «READY»). С ПУ направляется специальный запрос «getOfflineRequest» согласно приложению № 4 к Требованиям к ПТС на конечную точку «/query» с указанием идентификатора запроса. В ПТС должна поддерживаться функция отправления результатов выполнения отложенного запроса постранично в соответствии с подпунктом 2 пункта 8 приложения № 2 к Требованиям к ПТС;

7) после получения запроса «getOfflineRequest» в ПТС должна быть обеспечена передача на ПУ результатов выполнения отложенного поискового запроса в режиме реального времени в формате JSON. Структура результатов выполнения отложенного поискового запроса должна соответствовать структуре исходного поискового запроса, полученного на ПТС через интерфейс «/query»;

8) в момент выполнения отложенного запроса в ПТС должна обеспечиваться возможность прерывания данной операции, после получения запроса «cancelOfflineRequest», в соответствии с приложением № 4 к Требованиям к ПТС на интерфейс «/query» с указанием идентификатора запроса. В ответ посредством ПТС должен быть направлен результат выполнения данного запроса.

25. Посредством использования ПТС должно обеспечиваться автоматическое удаление промежуточных результатов прерванного отложенного запроса.

26. В ПТС должно обеспечиваться хранение результатов выполнения отложенных запросов согласно пункту 22 настоящего приложения к Требованиям к ПТС или немедленное их удаление при получении запроса.

Для удаления результатов выполнения отложенного поискового запроса на ПТС в ПТС должно быть предусмотрено получение от ПУ запроса «delOfflineRequest» в соответствии с приложением № 4 к Требованиям к ПТС на интерфейс «/query» с указанием идентификатора запроса. В ответ посредством использования ПТС обеспечивается автоматическая отправка результатов выполнения данного специального запроса на ПУ.

27. В ПТС должна поддерживаться постраничная передача результатов выполнения поисковых запросов в соответствии с параметрами, задаваемыми с ПУ («offset» и «cursor»). По согласованию с уполномоченными государственными органами, осуществляющими оперативно-разыскные мероприятия или обеспечение безопасности Российской Федерации в соответствии с правилами взаимодействия, собственниками или иными владельцами технологических сетей связи, имеющими уникальный идентификатор совокупности средств связи и иных технических средств в сети «Интернет», на ПТС допустима реализация только одного из видов постраничной передачи результатов.

28. В ПТС должна обеспечиваться функциональная возможность получения запросов постановки объектов на контроль от ПУ через интерфейс «/subscription» по протоколу WebSocket. После получения запроса с ПТС автоматически направляются на ПУ результаты выполнения данного запроса по мере появления новых данных (с момента получения запроса) в ИС, соответствующих критериям отбора, заданным в запросе ПУ. Передача результатов выполнения запросов постановки объектов на контроль осуществляется в непрерывном режиме, без дополнительных запросов от ПУ.

29. Для снятия объектов с контроля на ПТС должно быть получено сообщение «Complete» от ПУ в соответствии с подпунктом 6 пункта 8 приложения № 6 к Требованиям к ПТС на интерфейс «/subscription» по протоколу WebSocket. После получения сообщения «Complete» должно прекратиться исполнение запроса постановки объекта на контроль и передача на ПУ результатов выполнения данного запроса.

30. В ПТС должен обеспечиваться автоматический контроль функционирования собственных параметров и передача на ПУ следующих сигналов:

- 1) «Перезапуск программного обеспечения» («RESTARTDB»);
- 2) «Попытка несанкционированного доступа» («UNAUTHORIZEDACCESS»);
- 3) «Критическая ошибка программного обеспечения, потеря данных, дальнейшая работа невозможна» («CRITICALERROR»);
- 4) «Серьезная ошибка программного обеспечения, потеря данных, но дальнейшая работа возможна» («MAJORERROR»);
- 5) «Незначительная ошибка программного обеспечения, данные не потеряны, дальнейшая работа возможна» («MINORERROR»);
- 6) «Изменение схемы данных» («SCHEMACHANGED»);
- 7) «Предупреждение о проблеме с контролируемыми параметрами (метриками)» («METRICALERTS»). Для данного сигнала ПТС должны возвращать дополнительную информацию о предупреждениях, указанную в приложении № 4 к Требованиям к ПТС. Список контролируемых метрик и предупреждений о проблемах с ними определяется разработчиком ПТС и согласовывается с уполномоченными государственными органами, осуществляющими оперативно-разыскные мероприятия или обеспечение безопасности Российской Федерации.

31. Для отправки на ПУ сигналов в ПТС необходимо предусмотреть функциональную возможность исполнения запроса от ПУ «trap» (запрос Subscription языка GraphQL) через интерфейс «/subscription» по протоколу WebSocket. Посредством использования ПТС автоматически в непрерывном режиме (без дополнительных запросов от ПУ) отправляются на ПУ сигналы сразу после их возникновения на ПТС. Если в момент получения запроса «trap» на ПТС есть сигналы, которые не были отправлены ранее ПУ, то посредством использования ПТС должна быть обеспечена передача на ПУ данных сигналов.

32. В ПТС должны быть предусмотрены 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

Код ошибки	Описание ошибки
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)
INTERNAL_SERVER_ERROR	Внутренняя ошибка

33. В ПТС должно обеспечиваться выполнение всех запросов мониторинга и неформатированных данных, поступающих на ПТС от ПУ, в режиме реального времени.

Приложение № 2

к Требованиям к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», 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 Минцифры России от 16 декабря 2025 г. № 1174

ТРЕБОВАНИЯ

к программным и техническим средствам, используемым собственниками или иными владельцами технологических сетей связи, имеющими уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», в части предоставления информации, хранящейся в схеме данных, при выполнении запросов от пунктов управления уполномоченных государственных органов, осуществляющих оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации

1. Данные, хранящиеся в ИС, должны быть описаны с помощью схемы данных на языке SDL GraphQL (далее – схема данных).

2. Произвольный тип данных «А» в схеме данных должен иметь связь с произвольным типом данных «В», если произвольный тип данных «А» содержит поле произвольного типа данных «В» или поле, являющееся списком типа данных «В».

3. Произвольный тип данных «С» в схеме данных должен иметь двухстороннюю связь с произвольным типом данных «D», если произвольный тип данных «С» связан с произвольным типом данных «D» и произвольный тип данных «D» связан с произвольным типом данных «С».

4. Для схемы данных разработчиком ПТС должен быть определен перечень базовых типов данных на языке GraphQL, содержащих информацию, которая должна быть доступна для поиска в ПТС по запросам ПУ. Перечень базовых типов данных (приложение № 3 к Требованиям к ПТС) должен содержать параметры идентифицирующих признаков объектов физического мира.

5. Для схемы данных разработчиком ПТС определяется перечень сервисных типов на языке GraphQL, отражающих информацию о схеме данных, или описывающих HTTP URI для неформатированных данных,

или предназначенных для получения статуса отложенных поисковых запросов или сигналов о функционировании ПТС. Посредством использования ПТС должен поддерживаться перечень сервисных типов, приведенный в приложении № 4 к Требованиям к ПТС.

6. Пользовательский тип данных на языке GraphQL, не являющийся базовым сервисным типом или встроенным типом языка GraphQL, должен содержать идентифицирующий признак объекта физического мира. Пользовательские типы используются для описания данных, содержащихся в ИС и отличных от данных, описываемых базовыми типами. Требования к пользовательским типам должны соответствовать пункту 6 настоящего приложения к Требованиям к ПТС.

7. Схема данных в части базовых, пользовательских и сервисных типов должна соответствовать следующим требованиям:

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

2) поля пользовательского типа, которые содержат ссылки (HTTP URI) на неформатированные данные (файлы), должны представляться сервисным типом «Media»;

3) остальные поля пользовательских типов должны представляться в том числе встроенными типами языка GraphQL;

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 пользовательского типа	результат выполнения поискового запроса для данного пользовательского типа
--------	-------------------------------	--

Программный код (тип поля представления результатов) выполнения поисковых запросов:

```
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 к ПТС	для задания параметров поиска конкретных полей пользовательского типа, представленных базовыми типами

Программный код входного объекта для пользовательского типа должен иметь следующую структуру:

```
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 № 5 к Требованиям к ПТС;

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 № 4 к Требованиям к ПТС.

8. Схема данных в части стандартных типов «Query», «Subscription» языка GraphQL должна соответствовать следующим требованиям:

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

«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
все имена пользовательских типов, имеющие хотя бы одно поле базового типа	тип представления результатов выполнения поисковых запросов, соответствующий пользовательскому типу	задание параметров поиска и постраничной передачи результатов поиска для пользовательского типа

Поле типа «QueryMessage», содержащее все имена пользовательских типов, имеющее хотя бы одно поле базового типа, заданное списком пользовательского типа, должно иметь следующие аргументы:

«filter» – задает параметры поиска и представляется 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.

Программный код типа QueryMessage должен иметь следующую структуру:

```
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) тип «Subscription» должен содержать следующие поля для задания запросов постановки объектов на контроль:

Имя поля	Тип поля	Назначение поля
все имена пользовательских типов, имеющие хотя бы одно поле базового типа	список пользовательского типа	задание параметров постановки объектов на контроль для пользовательского типа

Каждое поле в типе «Subscription», представленное списком пользовательского типа, должно иметь аргумент «filter» для задания параметров

постановки объектов на контроль и представляться входным объектом для данного пользовательского типа.

Программный код типа «Subscription» должен иметь следующую структуру:

```
type Subscription {
  {{userType1}}(filter: {{UserTypeFilterInput1}}!): [{{UserType1}}!]
  ...
  {{userTypeN}}(filter: {{UserTypeFilterInputN}}!): [{{UserTypeN}}!]
},
```

в котором:

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

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

4) типы «Query» и «Subscription» должны содержать все запросы, соответствующие сервисным типам, которые приведены в приложении № 4 к Требованиям к ПТС.

9. В ПТС должны поддерживаться переменные типа «variables» языка GraphQL

в поисковых запросах и запросах постановки объектов на контроль, получаемых от ПУ. В ПТС должна обеспечиваться возможность использования актуальных перечней базовых, сервисных типов и входных объектов для поиска.

10. Использование базовых, сервисных типов и входных объектов для задания параметров поиска, не предусмотренными Требованиями к ПТС, должно согласовываться во время внедрения ПТС с уполномоченными государственными органами, осуществляющими оперативно-разыскные мероприятия или обеспечение безопасности Российской Федерации.

11. Наименование пользовательского типа и его полей должно соответствовать следующим требованиям:

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

имена пользовательских типов пишутся с заглавной буквы и могут состоять из нескольких слов, написанных слитно без пробелов, каждое слово должно быть написано с заглавной буквы;

имена полей пользовательских типов пишутся со строчной буквы и могут состоять из нескольких слов, написанных слитно без пробелов, каждое слово, кроме первого слова, должно быть написано с заглавной буквы.

Приложение № 3
к Требованиям к сетям и средствам связи
собственников или иных владельцев
технологических сетей связи, имеющих
уникальный идентификатор совокупности
средств связи и иных технических средств
в информационно-телекоммуникационной
сети «Интернет», 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 Минцифры России
от 16 декабря 2025 г. № 1174

ПЕРЕЧЕНЬ
базовых типов (кодов) для описания схемы данных

№ п/п	Описание	Базовый тип (код)
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 "" 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 """ Область """ region: String """ Район, муниципальный округ """ zone: String """ Город, поселок, 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 ФИО "" type PersonBase { "" Неструктурированная информация о ФИО "" fullName: String! "" Имя "" name: String "" Отчество (при наличии) "" middleName: String "" Фамилия "" lastName: String } </pre>
12.	Номер водительского удостоверения	<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	<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.	<p>Геометрические пространственно-временные данные: точка, массив точек, информация о месте и времени, полигон, мультиполигон, геотрек</p>	<pre> """ Базовый тип пространственной информации: точка """ type PointBase { """ Долгота """ longitude: Float! """ Широта """ latitude: Float! } """ Базовый тип пространственной информации: последовательность точек (линия) """ type LineBase { """ 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.	<p>Сведения о юридических лицах и индивидуальных предпринимателях</p>	<pre> "" Базовый тип сведений о юридических лицах и индивидуальных предпринимателях "" type OrganizationInfoBase { "" Полное наименование юридического лица или ИНН индивидуального предпринимателя "" nameFull: String "" Сокращенное наименование юридического лица (при наличии) "" nameSmall: String "" ОГРН или ОГРНИП "" grn: GRNBase! "" Идентификационный номер налогоплательщика "" inn: InnBase "" Информация о банковских данных "" bankInfo: BankAccountInfoBase "" Адрес "" address: AddressBase "" Информация о номере телефона "" msisdn: [MsisdnBase!] "" Сайт в сети «Интернет» "" webSite: URLBase "" Электронная почта "" 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 } "" </pre>
-----	---	--

		<pre> Базовый тип ОГРН или ОГРНИП """" type GRNBase { """" Государственный регистрационный номер – ОГРН и ОГРНИП """" grn: String! } """" 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 банковского счета """" account: String """" Корреспондентский счет """" corrAccount: String """" Номер банковской 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 } """" </pre>

		<pre> Базовый тип банковских переводов """" type BankTransferInfoBase { """" Сумма перевода """" amount: String! """" 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 в формате IPv4 (192.0.2.235) или IPv6 (1050:0000:0000:0000:0005:0600:300c:326b, 1050:0:0:0:5:600:300c:326b, ff06::c3) """" 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/идентификатор пользователя	<pre> "" Базовый тип имени/идентификатора пользователя "" type LoginBase { "" Имя/идентификатор пользователя "" login: String! } </pre>

Приложение № 4

к Требованиям к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», 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 Минцифры России от 16 декабря 2025 г. № 1174

ПЕРЕЧЕНЬ**сервисных типов для описания схемы данных**

№ 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. Ответом являются данные (результат выполнения запроса), представленные в формате JSON в соответствии с изначальным GraphQL запросом. Если отложенный запрос с указанным идентификатором не найден, то должно быть возвращено

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
			<pre> type Query { """" Удаление результатов отложенного запроса """" delOfflineRequest(requestID: String!): Boolean! } </pre>	<p>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> <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 ошибки «GraphQL».</p> <p>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 </pre>	<p>Прерывание (отмена) выполнения отложенного запроса по его идентификатору.</p>

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
			<pre> cancelOfflineRequest (requestID: String!): Boolean! } </pre>	<p>Если прерывание произошло успешно, то должен быть возвращен ответ «True».</p> <p>Если прерывание не удалось выполнить, то должен быть возвращен ответ «False» и описание ошибки «GraphQL».</p> <p>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 отложенных запросов	<pre> type RequestInfo { Идентификатор отложенного запроса requestID: String! Статус отложенного запроса status: RequestStatus Дата и время начала выполнения отложенного запроса startsAt: DateTimeBase } </pre>	<pre> type Subscription { Запрос статуса отложенных запросов statusOfflineRequest: RequestInfo! } </pre>	<p>Запрос статуса отложенных запросов. Ответом является сообщение типа «RequestInfo».</p>

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
		<p>Возможные статусы выполнения отложенного запроса " "" "" ""</p> <pre>enum RequestStatus { " "" "" "" Запрос еще не выполняется " "" "" "" NOTSTARTED " "" "" "" Запрос выполняется " "" "" "" RUNNING " "" "" "" Запрос выполнен " "" "" "" READY " "" "" "" Выполнение запрос прервано ПТС " "" "" "" ABORTED " "" "" "" Запрос отменен ПУ " "" "" "" CANCELED } </pre>		
4.	HTTP URI для неформатированных данных	<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 ПТС>:<порт>/download/<уникальный путь к файлу> " "" "" "" link: String! } </pre>	Отсутствует	Отсутствует

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
5.	Сигналы о функционировании и ПТС	<pre> """ Сервисный тип для сигналов о функционировании ПТС """ type Trap { """ Тип сигнала """ type: TrapType! """ Описание сигнала """ message: String """ Дата и время возникновения сигнала """ startsAt: DateTimeBase! """ Дополнительная информация о предупреждениях для контролируемых метрик (для сигнала типа «METRICALERTS») """ metricalerts: MetricAlerts } """ Возможные типы сигналов """ enum TrapType { """ Перезапуск программного обеспечения """ RESTARTDB """ Попытка несанкционированного доступа """ UNAUTHORIZEDACCESS """ Критическая ошибка """ CRITICALERROR """ Серьезная ошибка """ MAJORERROR """ Незначительная ошибка """ MINORERROR """ Изменение схемы данных """ SCHEMACHANGED """ </pre>	<pre> type Subscription { """ Запрос получения сигналов о функционировании ПТС """ trap: Trap! } </pre>	<p>Запрос получения сигналов о функционировании ПТС. Ответом является сообщение типа «Trap». Если тип сигнала «Предупреждение о проблеме с контролируемыми параметрами (METRICALERTS)», то ПТС должны возвращать дополнительную информацию о предупреждениях в поле «metricalerts».</p>

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
		<pre> 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 { </pre>		

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
		<pre> """ 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! } """ Ключ-значение """ type KeyValueRef { """ Ключ """ key: String! </pre>		

№ п/п	Описание типа	Сервисный тип	Запрос	Описание запроса
		"" Значение "" value: String! }		

Приложение № 5

к Требованиям к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», 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 Минцифры России от 16 декабря 2025 г. № 1174

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 хотя бы одно поле базового типа или связанного с другим пользовательским типом, имеющим хотя бы одно поле базового типа.

2. Каждый входной объект для базового типа должен содержать поля «and», «or» и «not», соответствующим логическим функциям «И», «ИЛИ» и «НЕ», а также может содержать поля для функций, приведенных в таблице № 1 настоящего приложения к Требованиям к ПТС. В случае если числовая функция применяется к строкам, то на множестве строк подразумевается лексикографический порядок.

Таблица № 1

№ п/п	Функция		Описание
1.	Операции сравнения	lt	Меньше, чем
2.		le	Меньше либо равно
3.		eq	Равно
4.		in	В [списке]
5.		between	Между значениями min и max

№ п/п	Функция		Описание
6.		ge	Больше либо равно
7.		gt	Больше
8.	Регулярные выражения	regexr	Регулярное выражение в формате POSIX Basic Regular Expression
9.	Поиск по тексту	allofterms	Соответствие строкам, содержащим все указанные термины в произвольном порядке, без учета регистра
10.		anyofterms	Соответствие строкам, содержащим любой из указанных терминов в произвольном порядке, без учета регистра
11.		alloftext	Полнотекстовый поиск по всему полю
12.		anyoftext	Полнотекстовый поиск по любой части поля
13.	Геопоиск	near	Соответствие объектам, предикат которых находится не более, чем на расстоянии «distance» метров от точки «point»
14.		within	Соответствие объектам, предикат которых находится в пределах полигона, заданного в качестве аргумента
15.		contains	Соответствует объектам, предикат которых содержит указанные точку или полигон
16.		intersects	Соответствует объектам, предикат которых имеет непустое пересечение с указанным полигоном или мультиполигоном
17.	Геопоиск (для геотрека)	nearTime	Соответствие объектам, предикат которых находится не более, чем на расстоянии «distance» от точки «point», при этом время, в которое предикат находился в данной области, ограничено фильтром по времени

№ п/п	Функция		Описание
18.		withinTime	Соответствие объектам, предикат которых находится в пределах полигона, заданного в качестве аргумента, при этом время, в которое предикат находился в данной области, ограничено фильтром по времени
19.		intersectsTime	Соответствует объектам, предикат которых имеет непустое пересечение с заданным полигоном или мультиполигоном, при этом время, в которое предикат находился в данной области, ограничено фильтром по времени

3. Для описания входных объектов, соответствующих пользовательским типам, должны использоваться входные объекты для базовых типов, приведенные в таблице № 2 настоящего приложения к Требованиям к ПТС.

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: StringFilter 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: StringFilter and: [ImeiBaseFilter] or: [ImeiBaseFilter] not: ImeiBaseFilter } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
3.	type MsisdnBase	<pre> """" Входной объект для базового типа номера абонента сети сотовой связи """" input MsisdnBaseFilter { """" Фильтр для номера абонента сети сотовой связи """" msisdn: StringFilter 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>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
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: StringFilter 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: StringFilter "" Фильтр для почтового индекса, zip-кода "" zip: StringFilter "" Фильтр для страны "" country: StringFilter "" Фильтр для области "" region: StringFilter "" Фильтр для района, муниципального округа "" zone: StringFilter "" Фильтр для города, поселка, деревни "" city: StringFilter "" Фильтр для улицы "" streetStringFilter } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
		<pre> """" Фильтр для дома, строения """" building: StringFilter """" Фильтр для корпуса """" buildsect: StringFilter """" Фильтр для квартиры, офиса """" apartment: StringFilter 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: StringFilter """" Фильтр для серии паспорта """" series: StringFilter and: [PassportBaseFilter] or: [PassportBaseFilter] not: PassportBaseFilter } </pre>
11.	type PersonBase	<pre> """" Входной объект для базового типа фамилии, имени, отчества (при наличии) """" input PersonBaseFilter { """" Фильтр для неструктурированной информации о фамилии, имени, отчестве (при наличии) """" fullName: StringFilter """" Фильтр для имени """" name: StringFilter """" Фильтр для отчества (при наличии) """" middleName: StringFilter """" Фильтр для фамилии """" lastName: StringFilter and: [PersonBaseFilter] or: [PersonBaseFilter] not: PersonBaseFilter } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
12.	type DrivingLicenseNumberBase	<pre> """ Входной объект для базового типа номера водительского удостоверения """ input DrivingLicenseNumberBaseFilter { """ Фильтр для номера водительского удостоверения """ drivingLicenseNumber: StringFilter 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: StringFilter and: [VehicleGosNumberBaseFilter] or: [VehicleGosNumberBaseFilter] not: VehicleGosNumberBaseFilter } </pre>
14.	type InnBase	<pre> """ Входной объект для базового типа идентификационного номера налогоплательщика """ input InnBaseFilter { """ Фильтр для значения идентификационного номера налогоплательщика """ inn: StringFilter and: [InnBaseFilter] or: [InnBaseFilter] not: InnBaseFilter } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
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>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
16.	type PointBase	<pre> """ Входной объект для базового типа пространственной информации: точки """ 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! """ Широта в градусах """ latitude: Float! } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
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>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
18	type PointTimeBase	<pre> """" Входной объект для базового типа пространственно- временной информации: точки в пространстве с 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 } """" Входной объект: пространственный фильтр включения заданной точки или полигона """" input ContainsFilter { </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
		<pre> """ 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 точек или полигона """ contains: ContainsFilter """ Фильтр для мультиполигона: пересечение с заданным полигоном """ intersects: IntersectsFilter and: [MultiPolygonBaseFilter] or: [MultiPolygonBaseFilter] not: MultiPolygonBaseFilter } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
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 указанных промежутках времени и расстояния """ intersectsTrack: IntersectsTrackFilter! and: [GeoTrackBaseFilter] or: [GeoTrackBaseFilter] not: GeoTrackBaseFilter } """ Входной объект: пространственно-временной фильтр нахождения около заданной точки в указанных пределах в заданное время """ input NearTimeFilter { distance: Float! point: PointRef! </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
		<pre> time: DateTimeBaseFilter! } """ Входной объект: пространственно-временной фильтр нахождения внутри заданного полигона в указанное время """ input WithinTimeFilter { polygon: PolygonRef! time: DateTimeBaseFilter! } """ Входной объект: пространственно-временной фильтр пересечения с заданным полигоном в указанное время """ input IntersectsTimeFilter { polygon: PolygonRef time: DateTimeBaseFilter! } """ Входной объект: пространственно-временной фильтр пересечения с заданным треком в указанных интервалах расстояния и времени """ input IntersectsTrackFilter { geotrack: GeoTrackRef! deltaTime: DeltaTimeRef! distance: Float! } """ Входной объект: период времени в секундах """ input DeltaTimeRef { second: Int! } """ Входной объект: трек для использования в фильтрах """ input GeoTrackRef { pointsInTime: [PointTimeRef!] } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
22.	type BankAccountInfoBase	<pre> "" Входной объект для базового типа банковских данных "" input BankAccountInfoBaseFilter { "" Фильтр для имени банка "" bankName: StringFilter "" Фильтр для номера счета "" account: StringFilter "" Фильтр для номера корреспондентского счета "" corrAccount: StringFilter "" Фильтр для номера карты "" cardNumber: StringFilter "" Фильтр для банковского идентификационного кода "" rcbic: StringFilter "" Фильтр для КПП "" kpp: StringFilter and: [BankAccountInfoBaseFilter] or: [BankAccountInfoBaseFilter] not: BankAccountInfoBaseFilter } </pre>
23.	type BankTransferInfoBase	<pre> "" Входной объект для базового типа банковских переводов "" input BankTransferInfoBaseFilter { "" Фильтр для отправителя "" from: BankAccountInfoBaseFilter "" Фильтр для получателя "" to: BankAccountInfoBaseFilter "" Фильтр для суммы перевода "" amount: FloatFilter "" Фильтр для даты и времени перевода "" date: DateTimeBaseFilter and: [BankTransferInfoBaseFilter] or: [BankTransferInfoBaseFilter] not: BankTransferInfoBaseFilter } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
24.	type OrganizationInfoBase	<pre> """ Входной объект для базового типа сведений о юридических лицах и индивидуальных предпринимателях """ input OrganizationInfoBaseFilter { """ Фильтр для полного наименования организации """ nameFull: StringFilter """ Фильтр для сокращенного наименования организации (при наличии) """ nameSmall: StringFilter """ Фильтр для ОГРН организации """ grn: GRNBaseFilter """ Фильтр для ИНН организации """ inn: InnBaseFilter """ Фильтр для ИНН иностранный организации """ vat: StringFilter """ Фильтр для банковских данных организации """ bankInfo: BankAccountInfoBaseFilter """ Фильтр для номера телефона организации """ msisdn: MsisdnBaseFilter """ Фильтр для сайта организации (при наличии) """ webSite: URLBaseFilter """ Фильтр для электронной почты организации (при наличии) """ email: EmailBaseFilter """ 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: StringFilter """ </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
		<pre> Фильтр для ЕГРЮЛ """" egrul: StringFilter """" Фильтр для ЕГРИП """" egrip: StringFilter and: [OrganizationInfoBaseFilter] or: [OrganizationInfoBaseFilter] not: OrganizationInfoBaseFilter } """" Входной объект для базового типа ОГРН или ОГРНИП """" input GRNBaseFilter { """" Фильтр для ОГРН или ОГРНИП """" grn: StringFilter 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 { """" Фильтр для должности """" position: StringFilter """" Фильтр для ИНН лица """" inn: InnBaseFilter """" Фильтр для фамилии, имени, отчества (при наличии) """" name: PersonBaseFilter and: [OrganizationRepresentativeBaseFilter] or: [OrganizationRepresentativeBaseFilter] not: OrganizationRepresentativeBaseFilter } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
26.	type IpAddressBase	<pre> """ Входной объект для базового типа IP-адреса """ input IpAddressBaseFilter { """ Фильтр подсетей, которые входят в искомую подсеть Для указания подсети используется CIDR-нотация (10.10.10.0/24) При отсутствии явного указания маски подсети принимается значение 32 (для IPv4) и 128 (для IPv6) """ inSubnet: String } """ Входной объект для базового типа IP-адреса (поиск подсетей) """ input IpSubnetBaseFilter { """ Фильтр подсетей, которые входят в искомую подсеть Для указания подсети используется CIDR-нотация (10.10.10.0/24) При отсутствии явного указания маски подсети принимается значение 32 (для IPv4) и 128 (для IPv6) """ inSubnet: String """ Фильтр подсетей, в которые входит искомая подсеть Для указания подсети используется CIDR-нотация При отсутствии явного указания маски подсети принимается значение 32 (для IPv4) и 128 (для IPv6) """ outSubnet: String } </pre>
27.	type NetworkPeerBase	<pre> """ Входной объект для базового типа информации об участнике сетевого соединения """ input NetworkPeerBaseFilter { """ Фильтр для IP-адреса """ ip: IpAddressBaseFilter """ Фильтр для порта """ port: IntFilter """ Фильтр для номера протокола """ protocolNumber: IntFilter and: [NetworkPeerBaseFilter] or: [NetworkPeerBaseFilter] not: NetworkPeerBaseFilter } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
28.	type URLBase	<pre> """ Входной объект для базового типа URL-адреса """ input URLBaseFilter { """ Фильтр для URL-адреса """ url: StringFilter 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: StringFilter and: [DomainNameBaseFilter] or: [DomainNameBaseFilter] not: DomainNameBaseFilter } </pre>
30.	type LoginBase	<pre> """ Входной объект для базового типа имени и (или) идентификатора пользователя """ input LoginBaseFilter { """ Фильтр для имени и (или) идентификатора пользователя """ login: StringFilter 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 """ В [списке] """ in: [Int] """ Меньше либо равно """ le: Int """ Меньше, чем """ } </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
		<pre> 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! } "" Входной объект: используемые функции для фильтрации значений типа Int по полному совпадению "" input IntExactFilter { "" Равно "" eq: 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 "" Меньше, чем "" lt: Float "" Больше либо равно "" ge: Float "" Больше, чем "" </pre>

№ п/п	Базовый тип и (или) скалярный тип	Входные объекты для задания параметров поиска
		<pre> 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: используемые функции для фильтрации значений типа String """ input StringFilter { """ Равно """ eq: String """ В [списке] """ in: [String] """ Регулярное выражение в формате POSIX Basic Regular Expression """ regexp: String } """ Входной объект: используемые функции для фильтрации значений типа String по полному совпадению """ input StringExactFilter { """ Равно """ eq: String } </pre>

4. Посредством использования ПТС должна быть обеспечена функциональная возможность поиска информации об IP-подсетях в рамках атрибута с типом «IpAddressBase» с использованием входного типа «IpSubnetBaseFilter». В случаях соответствия маски подсети числам 32 или 128 допустимо использование входного типа «IpAddressBaseFilter». В атрибутах входных типов «IpAddressBaseFilter» и «IpSubnetBaseFilter» IP-подсети должны указываться с использованием CIDR-нотации.

Приложение № 6

к Требованиям к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», 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 Минцифры России от 16 декабря 2025 г. № 1174

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

1. ПТС и ПУ для передачи данных на языке GraphQL по протоколу WebSocket должны использовать протокол graphql-transport-ws.
2. Посредством ПТС должно обеспечиваться взаимодействие с ПУ 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:
 - «id» – идентификатор запроса, использующийся для последующего определения ответов ПТС и привязки их к запросам ПУ;
 - «payload» – дополнительные данные.
4. В любой момент времени посредством использования ПТС должна обеспечиваться возможность выполнения нескольких запросов ПУ.
5. В ПТС должен быть обеспечен автоматический разрыв соединения по протоколу WebSocket при возникновении ошибки (сбоя в работе).
6. В ПТС должна поддерживаться возможность корректного завершения соединения по протоколу WebSocket с кодом 1000 и причиной закрытия «Normal Closure».
7. В ПТС должна поддерживаться возможность открытия сессии в соответствии с подпунктом 1 пункта 8 настоящего приложения к Требованиям к ПТС.

8. Посредством использования ПТС должно быть обеспечено взаимодействие с ПУ по протоколу WebSocket с использованием следующих сообщений:

1) сообщение «ConnectionInit» – направляется ПУ в адрес ПТС для открытия сессии. ПУ может передавать дополнительную информацию о сессии в поле «payload». ПТС должны получить данное сообщение в течение заданного периода времени (определяется во время внедрения ПТС и согласовывается с уполномоченными государственными органами, осуществляющими оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации). Если ПУ не отправило данное сообщение в течение заданного периода времени, то ПТС должны закрыть соединение по протоколу WebSocket с кодом 4408 и причиной закрытия «Connection initialization timeout». Если ПТС получили более одного сообщения «ConnectionInit», то они должны закрыть соединение по протоколу WebSocket с кодом 4429 и причиной закрытия «Too many initialization requests».

Программный код сообщения «ConnectionInit» (поле «payload» опциональное) должен иметь следующую структуру:

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

2) сообщение «ConnectionAck» – подтверждает создание сессии и направляется ПТС в адрес ПУ в ответ на сообщение «ConnectionInit». ПТС могут передавать дополнительную информацию о сессии в поле «payload». ПУ после получения данного сообщения может отправлять запросы на языке GraphQL с помощью сообщений «Subscribe» в соответствии с подпунктом 3 пункта 8 настоящего приложения к Требованиям к ПТС.

Программный код сообщения «ConnectionAck» (поле «payload» опциональное) должен иметь следующую структуру:

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

3) сообщение «Subscribe» – направляется ПУ в адрес ПТС с запросом на языке GraphQL в поле «query» поля «payload». Поле «id» должно содержать уникальный идентификатор запроса, формируемый ПУ. Если запрос с таким идентификатором находится на выполнении, то ПТС должны обеспечить завершение соединения 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 ПТС. В случае, когда ПТС получил данное сообщение до отправления сообщения «ConnectionAck», ПТС должны обеспечить завершение соединения по протоколу WebSocket с кодом 4401 и причиной закрытия «Unauthorized».

Программный код сообщения «Subscribe» (поля «operationName», «variables», «extensions» опциональные) должен иметь следующую структуру:

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

4) сообщение «Next» – направляется ПТС в адрес ПУ с результатами выполнения запроса, ранее направленного ПУ в сообщении «Subscribe». Поле «id» должно содержать идентификатор запроса, ранее сформированный ПУ. Поле «payload» должно содержать результаты выполнения запроса в формате аналогичном для конечной точки «/query» согласно спецификации языка GraphQL.

Программный код сообщения «Next» должен иметь следующую структуру:

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

5) сообщение «Error» – направляется ПТС в адрес ПУ с информацией об ошибке в ходе выполнения запроса, ранее направленного ПУ в сообщении «Subscribe». Поле «id» должно содержать идентификатор запроса, ранее сформированный ПУ. Поле «payload» должно содержать информацию об ошибке в формате согласно спецификации языка GraphQL. После возникновения ошибки ПТС должен прекращать выполнение запроса и не направлять в сторону ПУ больше никаких сообщений.

Программный код сообщения «Error» должен иметь следующую структуру:

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

6) сообщение «Complete» – может направляться в обоих направлениях. Поле «id» должно содержать идентификатор запроса, ранее сформированный ПУ. При завершении выполнения запроса с указанным идентификатором ПТС должны отправить в адрес ПУ сообщение «Complete». Если ранее ПТС отправили сообщение «Error», то данное сообщение не отправляется. Отправка данного сообщения от ПУ в адрес ПТС должна прервать выполнение запроса инициированного на стороне ПУ с указанным идентификатором.

Программный код сообщения «Complete» должен иметь следующую структуру:

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

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

Приложение № 7

к Требованиям к сетям и средствам связи собственников или иных владельцев технологических сетей связи, имеющих уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет», 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 Минцифры России от 16 декабря 2025 г. № 1174

ТРЕБОВАНИЯ,

предъявляемые к запросам мониторинга и формату их передачи по интерфейсу взаимодействия между пунктами управления уполномоченных государственных органов, осуществляющих оперативно-разыскную деятельность или обеспечение безопасности Российской Федерации, и программными и техническими средствами, используемыми собственниками или иными владельцами технологических сетей связи, имеющими уникальный идентификатор совокупности средств связи и иных технических средств в информационно-телекоммуникационной сети «Интернет»

1. Посредством использования ПТС должна поддерживаться возможность отправления на ПУ следующей информации о структуре и функционировании ПТС по запросу ПУ:

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

2) об установленном в ПТС общесистемном программном обеспечении, перечне и состоянии программных модулей в составе ПО ПТС;

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

2. ПТС по запросу ПУ должны предоставлять следующую информацию в части структуры и состава ПТС, состава и состояния интерфейса взаимодействия ПТС с ПУ:

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

2) идентификацию интерфейсов подключения оборудования ПТС друг к другу;

3) параметры для серверного оборудования:
общий и занятый объем оперативной памяти;

количество сетевых интерфейсов с их идентификацией и нагрузкой;
 общее количество и загрузку процессоров;
 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 («штатное функционирование», «сбой», «не функционирует»), текстовую расшифровку сбоя.

3. Посредством использования ПТС по запросу ПУ должна быть предоставлена информация в части точек подключения ПТС к ИС, интерфейсов ввода информации в ПТС, содержащая:

1) перечень точек подключения к технологической сети связи и точек ввода информации в ПТС с их идентификацией;

2) для каждой точки подключения посредством ПТС предоставляется информация о (об):

состоянии точки подключения (ввода информации) («штатное функционирование», «сбой», «не функционирует»), текстовую расшифровку сбоя;

объеме информации, поступающей в секунду;

периоде времени, в течение которого на точку подключения и (или) ввода информации в ПТС не поступала информация.

4. ПТС по запросу ПУ должны предоставлять следующую информацию в части состава программного обеспечения ПТС и его состояния:

1) перечень установленного общесистемного программного обеспечения с его идентификацией;

2) информацию для общесистемного программного обеспечения:

идентификатор ПТС, на котором установлено программное обеспечение;

наименование программного обеспечения;

состояние программного обеспечения («штатное функционирование», «сбой», «не функционирует»), текстовую расшифровку сбоя;

3) перечень установленного ПО ПТС с его идентификацией;

4) информацию для программного обеспечения ПТС:

идентификатор ПТС, на котором установлено программное обеспечение;

назначение (определяется разработчиком ПТС);

состояние («штатное функционирование», «сбой», «не функционирует»), текстовую расшифровку сбоя;

список контролируемых параметров (определяется разработчиком ПТС).

5. Информация о функционировании и состоянии ПТС, указанная в пунктах 1 – 4 настоящего приложения к Требованиям к ПТС, должна быть представлена в виде временных рядов.

6. Временные ряды должны иметь следующий обязательный набор меток:

systemtype – относится ли метрика к оборудованию или программному обеспечению. Данный временной ряд должен 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 должны быть заданы единицы измерения и описания их назначения на русском языке.

8. Для получения на ПУ значений метрик посредством ПТС обеспечивается возможность отправки на ПУ запросов мониторинга на конечную точку «/metric».

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

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

11. ПТС 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 в соответствии с пунктом 13 настоящего приложения к Требованиям к ПТС;

503 Service Unavailable – выполнение запроса прервано на стороне ПТС.

12. Ответы на запросы мониторинга должны иметь следующий формат:

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

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

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

в котором:

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

поле «data» – результат выполнения запроса, структура поля зависит от конечной точки, на которую был отправлен запрос (в соответствии с пунктом 13 настоящего приложения к Требованиям к ПТС);

поля «errorType» и «error» – соответственно тип и описание ошибки, возникшей при выполнении запроса (поле «status» имеет значение «error»);

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. ПТС должны реализовывать внутри конечной точки «/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 конечная точка должна поддерживать HTTP методы GET и POST с параметрами:

«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.

Программный код для поля «data» для ответа на запросы мониторинга для данной конечной точки должен иметь следующую структуру:

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

Формат поля «result» должен зависеть от значения поля «resultType» согласно пункту 14 настоящего приложения к Требованиям к ПТС;

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 конечная точка должна поддерживать HTTP методы GET и POST с параметрами:

«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.

Программный код для поля «data» для ответа на запросы мониторинга для данной конечной точки должен иметь следующую структуру:

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

Формат поля «result» должен соответствовать подпункту 1 пункта 14 настоящего приложения к Требованиям к ПТС;

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 конечная точка должна поддерживать HTTP методы GET и POST с параметрами:

«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 конечной точки 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 конечная точка должна поддерживать HTTP методы GET и POST с параметрами:

«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 конечной точки 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 конечная точка должна поддерживать HTTP метод GET с параметрами:

«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 конечной точки 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>"].
```