ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 71806— 2024

Цифровая промышленность

УНИФИЦИРОВАННАЯ АРХИТЕКТУРА OPC

Часть 1

Обзор и концепции

(IEC/TR 62541-1:2020, NEQ)

Издание официальное

Москва
Российский институт стандартизации
2025

Предисловие

1 РАЗРАБОТАН Ассоциацией «Цифровые инновации в машиностроении» и Федеральным государственным бюджетным учреждением «Российский институт стандартизации» (ФГБУ «Институт стандартизации»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 306 «Измерения, управление и автоматизация в промышленных процессах»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 29 ноября 2024 г. № 1802-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений международного документа IEC/TR 62541-1:2020 «Унифицированная архитектура OPC. Часть 1. Обзор и концепции» (IEC/TR 62541-1:2020 «OPC unified architecture — Part 1: Overview and concepts», NEQ)

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.rst.gov.ru)

Содержание

Введение

Система стандартов в цифровой промышленности имеет важное значение для управления процессами цифровой трансформации и создания умных производств, основанных на интеграции и интероперабельности различных автоматизированных систем управления, технологического оборудования, промышленных роботов, средств оснащения, контроллеров и датчиков.

Настоящий стандарт является первой частью серии стандартов «Цифровая промышленность. Унифицированная архитектура OPC». Все части данной серии стандартов входят в систему стандартов в цифровой промышленности.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Цифровая промышленность

УНИФИЦИРОВАННАЯ АРХИТЕКТУРА OPC

Часть 1

Обзор и концепции

Digital industry. OPC unified architecture. Part 1. Overview and concepts

Дата введения — 2025—02—01

1 Область применения

В настоящем стандарте представлены обзор и концепции унифицированной архитектуры открытой платформы взаимодействия (OPC UA — Open Platform Communication Unified Architecture), обеспечивающей универсальный механизм обмена данными в промышленных системах контроля и управления.

В настоящий стандарт включены термины и определения, дано описание других частей серии стандартов «Цифровая промышленность. Унифицированная архитектура OPC», совместно с которыми он должен использоваться.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 59799 Умное производство. Модель эталонной архитектуры Индустрии 4.0 (RAMI 4.0)

ГОСТ Р 70988 Система стандартов в цифровой промышленности. Основные положения. Общие требования к системе

ГОСТ Р 70992 Цифровая промышленность. Интеграция и интероперабельность систем. Термины и определения

ГОСТ Р 71807 Цифровая промышленность. Унифицированная архитектура OPC. Часть 2. Модель безопасности

ГОСТ Р 71808 Цифровая промышленность. Унифицированная архитектура OPC. Часть 3. Модель адресного пространства

ГОСТ Р 71809 Цифровая промышленность. Унифицированная архитектура OPC. Часть 4. Сервисы

ГОСТ Р 71810 Цифровая промышленность. Унифицированная архитектура OPC. Часть 5. Информационная модель

ГОСТ Р 71811 Цифровая промышленность. Унифицированная архитектура OPC. Часть 6. Сопоставления

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать версию этого стандарта с уче-

Издание официальное

том всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящем стандарте применены термины по ГОСТ Р 59799, ГОСТ Р 70988 и ГОСТ Р 70992, а также следующие термины с соответствующими определениями:

3.1.1 **адресное пространство** (adress space): Набор информации, которую сервер делает видимой для своих клиентов.

Примечание — Описание содержания и структуры адресного пространства сервера согласно ГОСТ Р 71808.

3.1.2 **атрибут** (attribute): Базовая характеристика узла.

Примечание — Все атрибуты определяются OPC UA и не могут быть определены клиентами или серверами. Атрибуты являются единственными элементами в адресном пространстве, которые могут иметь значения данных.

3.1.3 **базовая система** (underlying system): Аппаратные или программные платформы, существующие как самостоятельная сущность.

Примечания

1 Приложения UA зависят от существования сущности, чтобы выполнять услуги UA. Однако сущность не зависит от приложений UA.

2 Аппаратные и программные платформы включают физическое оборудование, встроенное программное обеспечение, операционную систему, сетевое взаимодействие, не относящиеся к UA приложения, равно как и другие приложения UA. Распределенная система управления, ПЛК/устройство и UA-сервер являются примерами базовой системы.

3.1.4 **брокер** (broker): Программный модуль-посредник, который направляет сетевые сообщения от издателей к подписчикам.

Примечание — Брокеры являются составными частями связующего ПО, ориентированного на обработку сообщений.

3.1.5 **издатель** (publisher): Сущность, отправляющая сетевые сообщения в связующее ПО, ориентированное на обработку сообщений.

Примечание — Издатель является оригинальным приложением OPC UA или приложением, которое обладает данными лишь о связующем ПО, ориентированном на обработку сообщений и правилах кодирования сетевых сообщений и сообщений набора данных.

3.1.6 **информационная модель** (information model): Организационная структура, определяющая, характеризующая и связывающая информационные ресурсы данной системы или набора систем.

Примечание — Основная модель адресного пространства поддерживает представление информационных моделей адресного пространства. Описание базовой информационной модели OPC UA представлено в ГОСТ Р 71810.

3.1.7 **класс узла** (node class): Класс узла в адресном пространстве.

Примечание — Классы узла определяют метаданные для компонентов объектной модели OPC UA. Они также определяют конструкции, такие как представления, которые используются для организации адресного пространства.

3.1.8 **клиент** (client): Программное приложение, которое отправляет сообщения на серверы OPC UA, в соответствии со службами, указанным в данном наборе спецификаций.

3.1.9 **метод** (method): Вызываемая программная функция, являющаяся компонентом объекта.

3.1.10 **модель «издатель-подписчик»** (publisher-subscriber): Используемый OPC UA вариант шаблона обмена сообщениями «публикация-подписка».

3.1.11 **набор данных** (data set): Список наименованных значений данных.

Примечание — Набор данных состоит из полей событий или значений переменных.

3.1.12 **набор служб** (service set): Группа связанных служб.

3.1.13 **обнаружение** (discovery): Процесс, с помощью которого клиент получает информацию о серверах, включая информацию о конечных точках и безопасности.

3.1.14 **объект/экземпляр объекта** (object/object instance): Узел, представляющий физический или абстрактный элемент системы.

Примечание — Объекты моделируются с помощью объектной модели OPC UA. Примерами объектов являются системы, подсистемы и устройства. Объект определяется как экземпляр типа объекта.

3.1.15 **переменная** (variable): Узел, содержащий какое-либо значение.

3.1.16 **подписка** (subscription): Определяемая клиентом конечная точка в сервере, используемая для возврата уведомлений клиенту.

Примечание — «Подписка» — это общий термин, описывающий набор узлов, выбранных клиентом, который сервер периодически отслеживает на наличие какого-либо состояния и для которого сервер отправляет уведомления клиенту при обнаружении состояния.

3.1.17 **подписчик** (subscriber): Объект, получающий сообщения наборов данных из связующего ПО, ориентированного на обработку сообщений.

Примечание — Подписчик является собственным приложением OPC UA или приложением, которое знает только о связующем ПО, ориентированном на обработку сообщений и правилах декодирования сетевых сообщений и сообщений наборов данных. Подписка в модели OPC UA «клиент-сервер» имеет иное значение, чем подписчик в модели «издатель-подписчик».

3.1.18 **представление** (view): Определенное подмножество адресного пространства, которое представляет интерес для клиента.

3.1.19 **приложение OPC UA** (OPC UA application): Клиент, вызывающий службы OPC UA, или сервер, выполняющий опции этих служб, или же подписчик или издатель OPC UA.

3.1.20 **программа** (program): Исполняемый объект, который при вызове немедленно возвращает ответ, указывающий на начало выполнения, а затем возвращает промежуточные и окончательные результаты через подписки, определенные клиентом во время вызова.

3.1.21 **профиль** (profile): Конкретный набор возможностей, на соответствие которым может претендовать сервер.

Примечание — Каждый сервер может заявить о соответствии более чем одному профилю.

3.1.22 **связующее ПО, ориентированное на обработку сообщений** (message oriented middleware): Инфраструктура, поддерживающая отправку и прием сетевых сообщений между распределенными системами.

Примечание — Приложение OPC UA поддерживает различные типы инфраструктур связующего ПО, ориентированного на обработку сообщений, и протоколов, таких как AMQP, MQTT или UDP с IP мультикаст. Другие типы, такие как DDS или XMPP, также могут быть интегрированы в модель OPC UA «издатель-подписчик».

3.1.23 **сигнал предупреждения** (alarm): Тип события, связанный с состоянием, которое требует подтверждения.

Примечание — Описание аварийных сигналов приведено в ГОСТ Р 71807.

3.1.24 **служба** (service): Вызываемая опция на сервере.

Примечание — Службы определены в ГОСТ Р 71809. Служба аналогична вызову метода в языке программирования или опции в контракте WSDL веб-сервисов.

3.1.25 **событие** (event): Общий термин, используемый для описания события, имеющего определенное значение в системе или ее компоненте.

3.1.26 **совокупность** (aggregate): Функция, вычисляющая производные значения из необработанных данных.

Примечание — Необработанные данные получаются из исторических или буферизованных источников в реальном времени данных. Обычные совокупности включают средние значения по заданному временному диапазону, минимум по временному диапазону и максимум по временному диапазону.

3.1.27 **сервер** (server): Программное приложение, воплощающее и демонстрирующее службы, указанные в данном наборе спецификаций.

3.1.28 **сертификат** (certificate): Структура данных с цифровой подписью, содержащая открытый ключ и идентификационные данные клиента или сервера.

3.1.29 **сессия** (session): Длительное логическое соединение между клиентом и сервером.

Примечание — Сессия сохраняет информацию о состоянии между вызовами сервиса от клиента к серверу согласно ГОСТ Р 71811.

3.1.30 **сетевое сообщение** (network message): Сообщения набора данных и заголовок для упрощения доставки, маршрутизации, безопасности и фильтрации.

Примечания

1 Издатель передает сетевое сообщение связующему ПО, ориентированному на обработку сообщений (транспортный уровень) для доставки сообщения набора данных подписчикам.

2 В мире обмена сообщениями термин «сообщение» используется с различными значениями. Издатель понимает сообщение как неизменную полезную нагрузку, переданную связующему программному обеспечению, ориентированному на обработку сообщений. Подписчик понимает сообщение не только как неизменную полезную нагрузку от отправителя, но и как различные аннотации, попутно предоставляемые связующему программному обеспечению, ориентированному на обработку сообщений. Чтобы избежать путаницы, термин <сообщение цифровых данных системы> используется для обозначения сообщения, поставляемого издателем для набора данных, а термин <сетевое сообщение> используется для обозначения сообщения набора данных плюс секции для аннотаций в его начале и конце.

3.1.31 **сообщение** (message): Единица данных, передаваемая между клиентом и сервером, которая представляет собой конкретный запрос или ответ службы.

3.1.32 **сообщение набора данных** (data set message): Полезная нагрузка сетевого сообщения, созданная на основе набора данных.

Примечание — Сообщение набора данных — это неизменная полезная нагрузка сетевого сообщения, переданная в связующее ПО, ориентированное на обработку сообщений (транспортный уровень) для доставки издателем. Подписчик получает сообщение набора данных как полезную нагрузку сетевого сообщения от издателя как полезную нагрузку с дополнительными заголовками, которые попутно предоставляются связующим ПО, ориентированным на обработку сообщений.

3.1.33 **сообщение-уведомление** (notification message): Сообщение, опубликованное подпиской и содержащее одно или несколько уведомлений.

3.1.34 **состояние** (condition): Общий термин, являющийся следствием события.

Примечание — Состояние отображает состояние системы или одного из ее компонентов и всегда имеет определенный статус.

3.1.35 **ссылка** (reference): Прямая связь (именованный указатель) одного узла с другим.

3.1.36 **стек связи** (communication stack): Многоуровневый набор программных модулей между приложением и аппаратным обеспечением, обеспечивающий различные функции для кодирования, шифрования и форматирования отправляемого сообщения, а также для декодирования, дешифрования и распаковки полученного сообщения.

3.1.37 **тип объекта** (object type): Узел, представляющий определение типа объекта.

Примечание — Узел, содержащий ссылку, является исходным узлом, а ссылающийся узел — целевым узлом. Все ссылки описываются определениями типа ссылок.

3.1.38 **тип ссылки** (reference type): Узел, представляющий определение типа ссылки.

Примечание — Тип ссылки определяет семантику ссылки. Название типа ссылки определяет, как исходные узлы связаны с целевыми узлами, и обычно отражает операцию между ними например «А содержит В».

3.1.39 **уведомитель событий** (event notifier): Специальный атрибут узла, указывающий на то, что клиент может подписаться на этот конкретный узел, чтобы получать уведомления о возникновении событий.

3.1.40 **уведомление** (notification): Общий термин для данных, сообщающих об обнаружении события или изменения значения атрибута. Уведомления посылаются в виде сообщения-уведомления.

3.1.41 у**зел** (node): Основной компонент адресного пространства.

3.1.42 **элемент мониторинга** (monitoredItem): Определяемая клиентом сущность на сервере, используемая для мониторинга атрибутов или уведомителей событий на предмет новых значений или событий и генерации уведомлений для них.

3.2 Сокращения

В настоящем стандарте применены следующие сокращения:

A&E — сигналы и события (Alarms and Events);

AMQP — открытый протокол для передачи сообщений между компонентами системы (Advanced Message Queuing Protocol);

API — программный интерфейс приложения (Application Programming Interface);

COM — модель компонентного объекта (Component Object Model);

DA — доступ к данным (Data Access);

DCS — распределенная система управления (Distributed Control System);

DDS — служба распределения данных (Data Distribution Service);

HDA — доступ к данным за прошлые периоды (Historical Data Access);

HMI (ЧМИ) — человеко-машинный интерфейс (Human-Machine Interface);

JSON — нотация объектов JavaScript (JavaScript Object Notation);

LDAP — легковесный протокол доступа к каталогам (Lightweight Directory Access Protocol);

MES — система управления производственными процессами (Manufacturing Execution System);

MQTT — передача телеметрии очереди сообщений (Message Queue Telemetry Transport);

OPC — OPC Foundation (некоммерческая промышленная ассоциация) ранее была акронимом для «OLE for Process Control». Акроним больше не используется;

PLC — программируемый логический контроллер (Programmable Logic Controller);

SCADA — диспетчерское управление и сбор данных (Supervisory Control And Data Acquisition);

UA — унифицированная архитектура (Unified Architecture);

WSDL — язык описания веб-сервисов (Web Services Definition Language);

XML — расширяемый язык разметки (Extensible Mark-up Language);

XMPP — расширяемый протокол обмена сообщениями и информацией о присутствии (ExtensibleMessaging and Presence Protocol);

MOM — промежуточное программное обеспечение, ориентированное на сообщения (Message Oriented Middleware).

4 Структура серии OPC UA

4.1 Организация спецификации

В ГОСТ Р 71807, ГОСТ Р 71808, ГОСТ Р 71809, ГОСТ Р 71810 и ГОСТ Р 71811 показаны основные возможности OPC UA. Службы, работающие согласно структуре адресного пространства, определяют модель публикации-подписки OPC UA. В дополнение к модели клиент-сервер, определенной для сервисов согласно ГОСТ Р 71809, [1], [2], [3] и [4], применяют основные возможности конкретных типов доступа, ранее рассматривавшихся в отдельных спецификациях OPC COM, таких как доступ к данным (DA), сигналы тревоги и события (A&E) и доступ к данным за прошлые периоды (HDA). [5] описывает механизмы обнаружения для OPC UA, а [6] — способы объединения данных.

Настоящая спецификация организована в виде многокомпонентной спецификации, как показано на рисунке 1.

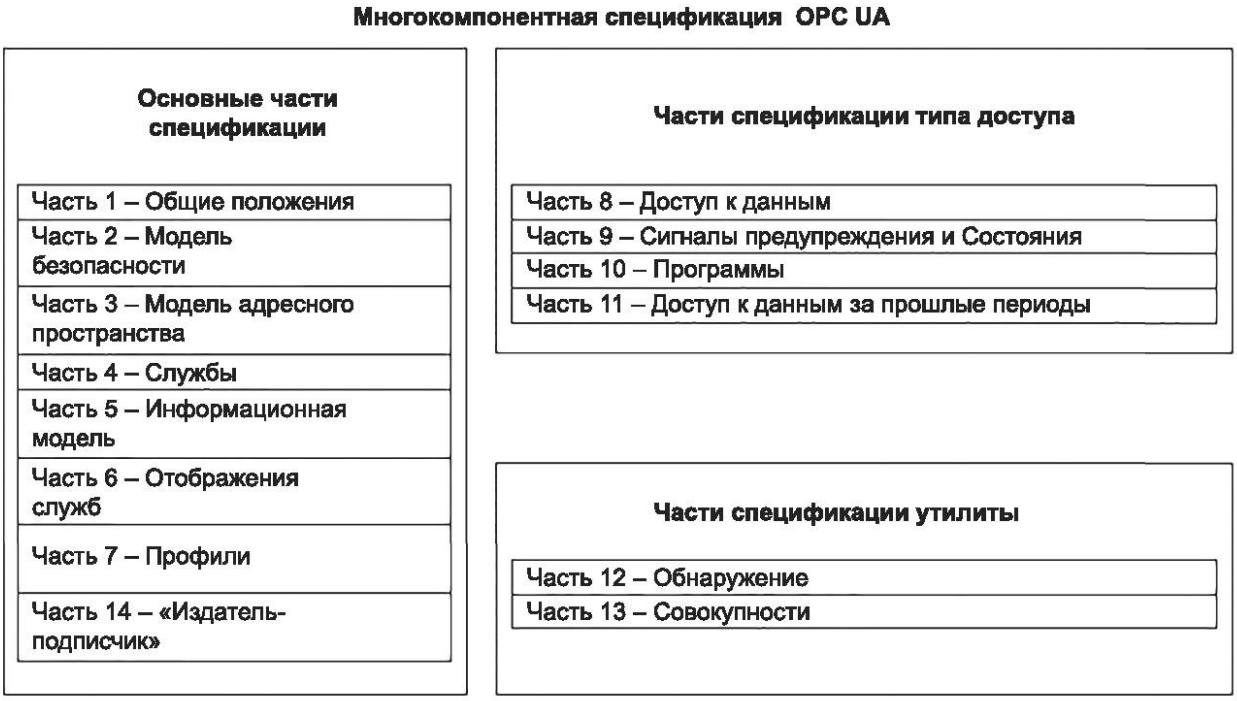

Рисунок 1 — Организация спецификации OPC UA

4.2 Основные части спецификации

Часть 1 — Общие положения

Настоящий стандарт устанавливает общие положения OPC UA.

Часть 2 — Модель безопасности

ГОСТ Р 71807 устанавливает модель для обеспечения безопасности взаимодействия между приложениями OPC UA.

Часть 3 — Модель адресного пространства

ГОСТ Р 71808 устанавливает содержание и структуру адресного пространства сервера.

Часть 4 — Службы

ГОСТ Р 71809 устанавливает службы, предоставляемые серверами.

Часть 5 — Информационная модель

ГОСТ Р 71810 устанавливает типы и их взаимосвязи, определенные для серверов.

Часть 6 — Отображения

ГОСТ Р 71811 устанавливает отображения транспортных протоколов и кодировок данных, поддерживаемых OPC UA.

Часть 7 — Профили

[7] устанавливает профили, доступные для приложений OPC UA. Эти профили представляют собой группы функциональных возможностей, которые используются для сертификации уровня соответствия. Приложения OPC UA следует тестировать на соответствие профилям.

4.3 Части спецификации типа доступа

Часть 8 — Доступ к данным

[1] определяет использование OPC UA для доступа к данным.

Часть 9 — Сигналы предупреждения и состояния

[2] определяет использование поддержки OPC UA для доступа к сигналам предупреждения и состояниям. Базовая система включает в себя поддержку простых событий; данная спецификация расширяет эту поддержку, чтобы включить поддержку сигналов предупреждения и состояний.

Часть 10 — Программы

[3] определяет поддержку OPC UA для доступа к программам.

Часть 11 — Доступ к данным за прошлые периоды

[4] определяет использование OPC UA для доступа к данным за прошлые периоды. Данный доступ включает в себя как данные, так и события за прошлые периоды.

4.4 Части спецификации утилиты

Часть 12 — Обнаружение

[5] определяет, как серверы обнаружения работают в различных сценариях, и описывает как клиенты и серверы UA должны взаимодействовать с ними. Он также определяет, как связанная с UA информация должна быть доступна с использованием общих протоколов службы каталогов, таких как LDAP.

Часть 13 — Совокупности

[6] определяет, как вычислять и возвращать агрегаты, такие как минимум, максимум, среднее и т. д. Совокупности могут использоваться с текущими и историческими данными.

Часть 14 — «Издатель-подписчик»

[8] определяет коммуникационную модель «издатель-подписчик» унифицированной архитектуры OPC (OPC UA). Модель связи «издатель-подписчик» определяет модель публикации-подписки OPC UA в дополнение к модели «клиент-сервер», определенной службами в [7].

5 Обзор

5.1 Область применения UA

OPC UA применим к компонентам во всех промышленных областях, таких как промышленные датчики и исполнительные механизмы, системы управления, системы управления производством и системы планирования ресурсов предприятия, включая промышленный Интернет вещей (IIoT), межмашинное взаимодействие (M2M), а также системы индустрии 4.0 и Китая 2025. Эти системы предназначены для обмена информацией и использования команд и управления промышленными процессами. OPC UA, определяют общую модель инфраструктуры для облегчения такого обмена информацией. OPC UA определяет:

- информационную модель для представления структуры, поведения и семантики;
- модель сообщений для взаимодействия между приложениями;
- модель связи для передачи данных между конечными точками;
- модель соответствия для обеспечения совместимости между системами.

5.2 Общие положения

OPC UA — независимый от платформы стандарт, с помощью которого различные типы систем и устройств взаимодействуют, отправляя Сообщения с запросами и ответами между клиентами и серверами или сетевые сообщения между издателями и подписчиками по различным типам сетей. Он поддерживает надежную, безопасную связь, которая гарантирует идентичность приложений OPC UA и противостоит атакам [6].

В модели клиент-сервер OPC UA определяет наборы служб, которые предоставляют серверы, а отдельные серверы указывают клиентам, какие наборы служб они поддерживают. Информация передается с помощью типов данных, определенных OPC UA и поставщиком, а серверы определяют объектные модели, которые клиенты динамически обнаруживают. Серверы должны предоставлять доступ к актуальным данным и данным за прошлые периоды, а также к сигналам предупреждения и состояниям для оповещения клиентов о важных изменениях согласно ГОСТ Р 71809. OPC UA должен иметь связь с различными коммуникационными протоколами, допустимо, что бы данные кодировались различными способами, позволяющими сочетать портативность и эффективность.

В дополнение к модели клиент-сервер, OPC UA определяет механизм передачи информации от издателей подписчикам с помощью модели «издатель-подписчик» [8].

5.3 Цели разработки

OPC UA обеспечивает согласованную, интегрированную модель адресного пространства и служб, что позволяет серверу встраивать данные, сигналы предупреждения, события и историю операций в свое адресное пространство и предоставлять доступ к ним с помощью встроенного набора служб. Эти службы также содержат встроенную модель безопасности.

OPC UA должна давать возможность серверам предоставлять клиентам определения типов для объектов, доступ к которым осуществляется из адресного пространства, что определяет применение информационных моделей для описания содержимого адресного пространства. OPC UA позволяет представлять данные в различных форматах, включая бинарные структуры и документы XML или JSON. Формат данных определяется OPC, другими стандартными организациями или поставщиками. Через адресное пространство клиенты запрашивают у сервера метаданные, описывающие формат данных. Клиенты, не имеющие запрограммированных знаний о форматах данных, могут определить их и правильно использовать во время выполнения заданий.

OPC UA добавляет поддержку множества отношений между узлами, не ограничиваясь единственной иерархией. Сервер представляет данные в различных иерархиях в соответствии с тем, как обычно хотят видеть эти данные различные клиенты. Такая гибкость в сочетании с поддержкой определений типов делает OPC UA применимым к широкому спектру проблемных областей. Как показано на рисунке 2, OPC UA ориентирован не только на интерфейсы SCADA, PLC и DCS, но и на обеспечение большей совместимости между функциями более высокого уровня.

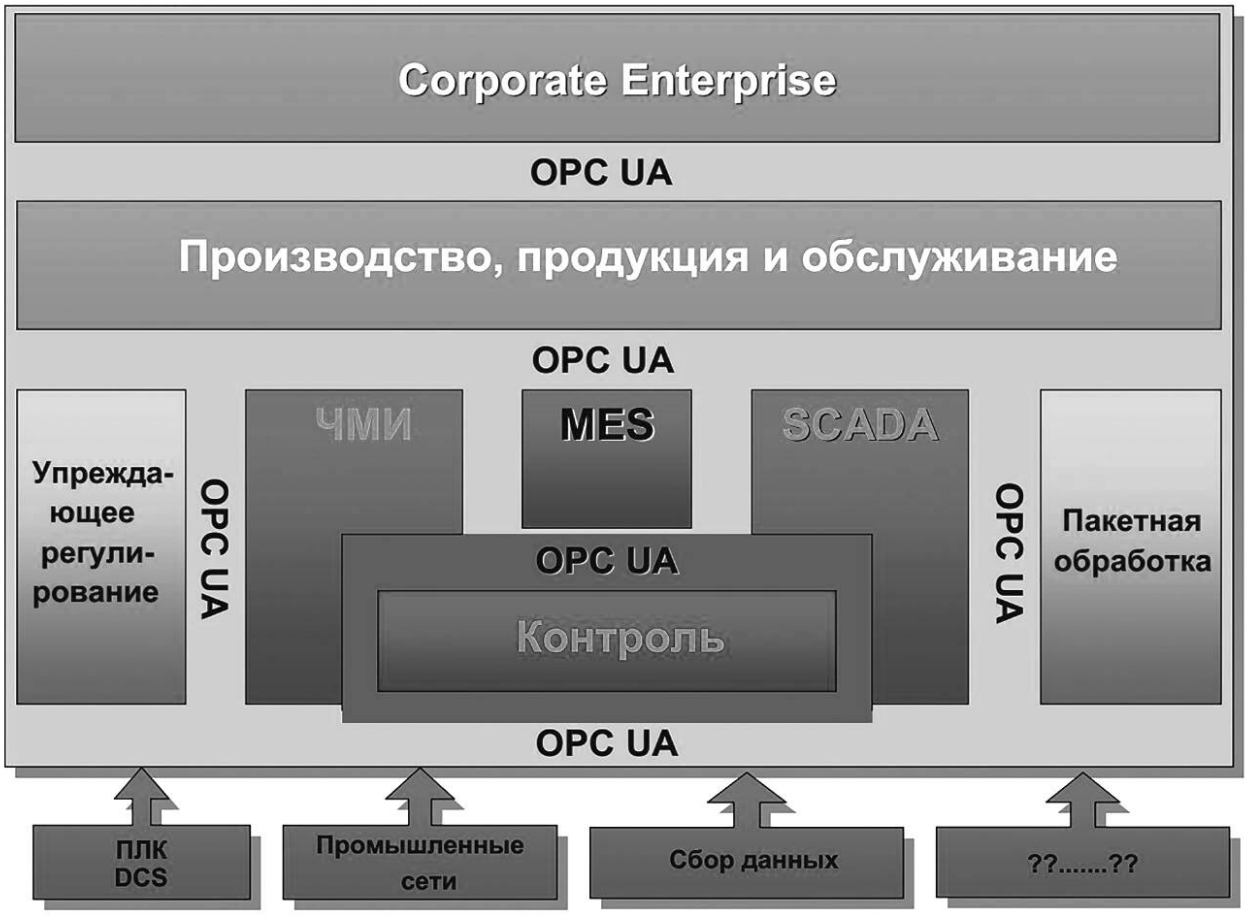

Рисунок 2 — Целевые приложения OPC UA

OPC UA разработан для обеспечения надежности публикуемых данных. Главной особенностью всех серверов OPC является возможность публикации данных и уведомлений о событиях. OPC UA предоставляет клиентам механизмы для быстрого обнаружения и восстановления после сбоев связи, связанных с этими передачами, без необходимости ожидания длительных тайм-аутов, характерных для базовых протоколов.

OPC UA разработана для поддержки широкого спектра серверов, от ПЛК на производстве до корпоративных серверов. Эти серверы характеризуются широким диапазоном размеров, производительности, платформ исполнения и функциональных возможностей. OPC UA определяет полный набор

возможностей, в то время как серверы реализуют лишь отдельные подмножества этих возможностей. Для обеспечения совместимости OPC UA определяет подмножества, называемые профилями, соответствия которым должны обеспечиваться серверами. Клиенты могут обнаружить профили сервера и настроить свое взаимодействие с сервером на основе профилей. Профили определены в [7].

Спецификации OPC UA имеют многоуровневую структуру, чтобы изолировать основную структуру от базовой вычислительной технологии и сетевой передачи. Это позволяет при необходимости адаптировать OPC UA к будущим технологиям, не отказываясь от базовой конструкции. Отображения и кодировки данных описаны в [1]. Определены три кодировки данных:

- XML/text,
- UA Binary,
- JSON.

Кроме того, определены протоколы:

- OPC UA TCP,
- HTTPS,
- WebSockets.

Приложения OPC UA, поддерживающие несколько вариантов передачи информации и кодировок, должны позволять конечным пользователям принимать решения о компромиссах между производительностью и совместимостью на этапе развертывания, а не доверять принятие этих компромиссов поставщику OPC на этапе разработки продукта.

OPC UA разработан как путь перехода для клиентов и серверов OPC, основанных на технологии Microsoft COM. Существующие данные, отображаемые серверами OPC COM (DA, HDA и A&E), должны быть совместимы и отображаться через OPC UA. Поставщики должны иметь возможность перейти к разработке своих продуктов сразу на OPC UA или использовать внешние модули для преобразования из OPC COM в OPC UA и наоборот. Каждая из предыдущих спецификаций OPC определяла свою собственную модель адресного пространства и свой собственный набор служб. Характеристики OPC UA должны объединять предыдущие модели в единое интегрированное адресное пространство с единым набором служб.

Модель OPC UA «издатель-подписчик» открывает новые области применения OPC UA. Ниже приведены примеры использования модели «издатель-подписчик»:

- конфигурируемая одноранговая связь между несколькими контроллерами и между контроллерами и программируемыми терминалами. Одноранговые системы не связаны напрямую и даже не должны знать о существовании друг друга. Для обмена данными требуется такой фиксированный временной интервал как «точка-точка» или многоточечное соединение;
- асинхронные рабочие процессы. Например, приложение для обработки заказов имеет возможность поместить заказ в очередь сообщений или шину корпоративных служб;
- ведение журналов операций в нескольких системах. Например, датчики или исполнительные устройства должны записывать журналы операций для систем мониторинга, HMI, архивного приложения для последующего запроса и так далее;
- серверы, представляющие службы или устройства, могут передавать данные в приложения, размещенные в облаке; например в серверы бэкенда и аналитические системы больших данных для оптимизации систем и профилактического обслуживания.

Модель «издатель-подписчик» не привязана к конкретной системе обмена сообщениями. Она сопоставима с различными системами. Например, модель «издатель-подписчик» с UDP приспособлена для частой передачи небольших объемов данных в производственных системах. Она также позволяет осуществлять обмен данными в конфигурациях «один к одному» и «один ко многим». Другим примером является использование устоявшихся протоколов обмена сообщениями с кодировкой данных JSON, поддерживающее путь облачной интеграции и позволяющее обрабатывать информацию в современных потоковых и пакетных аналитических системах.

5.4 Встроенные модели и сервисы

5.4.1 Модель безопасности

5.4.1.1 Общие положения

Безопасность OPC UA связана с аутентификацией клиентов и серверов, аутентификацией пользователей, целостностью и конфиденциальностью сообщений, а также верификацией функционального соответствия. OPC UA не определяет обстоятельства, при которых требуются различные механизмы

защиты. Эта спецификация очень важна, но она определяется разработчиками системы на конкретном объекте и может быть уточнена в других стандартах.

Модель безопасности OPC UA предоставлена в ГОСТ Р 71807, где указано, что меры безопасности выбирают и настраивают для соответствия потребностям безопасности конкретной установки. Эта модель включает в себя механизмы и параметры безопасности, которые определяют механизм обмена параметрами безопасности, но не определяют способ, которым приложения используют эти параметры. Данная модель OPC UA также определяет минимальный набор профилей безопасности, которые поддерживают все ее приложения, хотя они и могут применяться не во всех конфигурациях. Профили безопасности определены в [7].

5.4.1.2 Обнаружение и установление сессии

Безопасность на уровне приложения обеспечивается защищенным каналом связи, который активен в течение всей сессии приложения и обеспечивает целостность всех переданных сообщений, проходить аутентификацию только один раз при установлении сессии приложения. Механизмы обнаружения серверов и установления защищенных каналов связи и прикладных сессий описаны в настоящем стандарте и в ГОСТ Р 71811. Дополнительная информация о процессе обнаружения описана в [7].

При установлении сессии приложения клиента и сервера согласовывают безопасный канал связи. Для идентификации клиента и сервера используются цифровые сертификаты (ITU X.509). Сервер дополнительно аутентифицирует пользователя и утверждает последующие запросы на доступ к объектам сервера.

5.4.1.3 Аудит

OPC UA включает поддержку журналов аудита безопасности с возможностью сопоставления между журналами аудита клиента и сервера. При обнаружении проблемы защиты на сервере может быть найдена и изучена соответствующая запись в журнале аудита клиента. OPC UA также предоставляет серверам возможность генерировать уведомления о событиях аудита клиентам, для их обработки и регистрации. OPC UA определяет параметры аудита безопасности, которые могут быть включены в записи журнала аудита и в уведомления о событиях аудита. Не все серверы и клиенты предоставляют все функции аудита. Профили, приведенные в [7], указывают, какие функции поддерживаются.

5.4.1.4 Безопасность передачи данных

Безопасность OPC UA дополняет инфраструктуру безопасности, предоставляемую большинством платформ, поддерживающих веб-службы.

Безопасность уровня передачи данных может использоваться для шифрования и подписи сообщений. Шифрование и подписи защищают от раскрытия информации и обеспечивают целостность сообщений. Возможности шифрования обеспечиваются базовой технологией связи, используемой для обмена сообщениями между приложениями OPC UA. [7] определяет алгоритмы шифрования и подписи, которые должны использоваться для данного профиля.

5.4.2 Интегрированная модель адресного пространства

Набор объектов и сопутствующая информация, которые сервер делает доступными для клиентов, называется его адресным пространством. Адресное пространство OPC UA отображает свое содержимое как набор узлов, соединенных ссылками.

Примитивные характеристики узлов описываются определенными OPC атрибутами. Атрибуты — это единственные элементы сервера, которые имеют значения данных. Типы данных, определяющие значения атрибутов, могут быть простыми или сложными.

Узлы в адресном пространстве классифицируются в соответствии с их использованием и значением. Классы узлов определяют метаданные для адресного пространства OPC UA. [5] определяет классы узлов OPC UA.

Базовый класс узла определяет общие для всех узлов атрибуты, позволяющие идентифицировать, классифицировать и называть узлы. Каждый класс узла наследует эти атрибуты и может дополнительно определять собственные атрибуты.

Для обеспечения совместимости клиентов и серверов адресное пространство OPC UA иерархически структурировано, причем верхние уровни структуры одинаковы для всех серверов. Хотя узлы в адресном пространстве обычно доступны через иерархию, они могут иметь ссылки друг на друга, что позволяет адресному пространству представлять взаимосвязанную сеть узлов. Модель адресного пространства определена в ГОСТ Р 71808.

Серверы могут размещать адресное пространство в представлениях для упрощения доступа клиентов. В 6.3.4.3 более подробно описаны представления адресного пространства.

5.4.3 Интегрированная объектная модель

Объектная модель OPC UA обеспечивает согласованный, интегрированный набор классов узлов для представления объектов в адресном пространстве. Эта модель представляет объекты с точки зрения переменных, событий и методов, а также их отношений с другими объектами. ГОСТ Р 71811 описывает эту модель.

Объектная модель OPC UA позволяет серверам предоставлять определения типов для объектов и их компонентов. Определения типов могут быть подклассами. Они также могут быть общими или принадлежать конкретной системе. Типы объектов могут быть определены организациями по стандартизации, поставщиками или конечными пользователями.

Эта модель позволяет интегрировать данные, сигналы предупреждения, события, а также их историю в единый сервер. Например, серверы могут представлять датчик температуры как объект, состоящий из значения температуры, набора параметров тревоги и соответствующего набора предельных значений.

5.4.4 Интегрированные службы

Интерфейс между клиентами и серверами определяется как набор служб. Эти службы организованы в логические группы, называемые наборами служб. Наборы служб представлены в 6.4 и определены в ГОСТ Р 71809.

Службы OPC UA дают клиентам две возможности. Они позволяют клиентам отправлять запросы на серверы и получать от них ответы. Они также позволяют клиентам подписываться на серверы для получения уведомлений. Сервер использует уведомления для сообщения о таких событиях, как сигналы предупреждения, изменения значений данных, события и результаты выполнения программы.

В целях эффективности сообщения OPC UA могут быть закодированы в виде текста (XML или JSON) или в двоичном формате. Передача сообщений может проводиться с использованием нескольких базовых протоколов передачи, например TCP или HTTP. Серверы могут предоставлять различные кодировки и протоколы передачи, как определено в ГОСТ Р 71810.

5.5 Сессии

Взаимодействие клиент-сервер OPC UA требует наличия модели состояния. Информация о состоянии поддерживается внутри сессии приложения. Примерами информации о состоянии являются подписки, учетные данные пользователя и точки продолжения для операций, охватывающих несколько запросов.

Сессии определяются как логические соединения между клиентами и серверами. Серверы имеют ограничения по числу одновременных сессий в зависимости от доступности ресурсов, а также лицензионных и иных факторов. Сессия не зависит от базовых протоколов связи. Сбои в этих протоколах не приводят к автоматическому завершению сеанса. Сессии прекращаются на основании запроса клиента или сервера, а также на основании бездействия клиента. Интервал времени бездействия согласовывается во время установления сессии.

6 Концепции систем

6.1 Обзор модели клиент-сервер

Архитектура систем OPC UA моделирует клиентов и серверы как взаимодействующих партнеров. Каждая система может содержать несколько клиентов и серверов. Каждый клиент может одновременно взаимодействовать с одним или несколькими серверами, и каждый сервер может одновременно взаимодействовать с одним или несколькими клиентами. Приложение может объединять компоненты сервера и клиента для обеспечения взаимодействия с другими серверами и клиентами, согласно 6.3.7.

Клиенты и серверы представлены в 6.2 и 6.3, на рисунке 3 показана архитектура с объединенным сервером и клиентом.

Рисунок 3 — Архитектура системы OPC UA

6.2 Клиенты OPC UA

Клиентская архитектура OPC UA моделирует конечную точку клиента во взаимодействиях клиента и сервера. На рисунке 4 показаны основные элементы типичного клиента и их взаимосвязь.

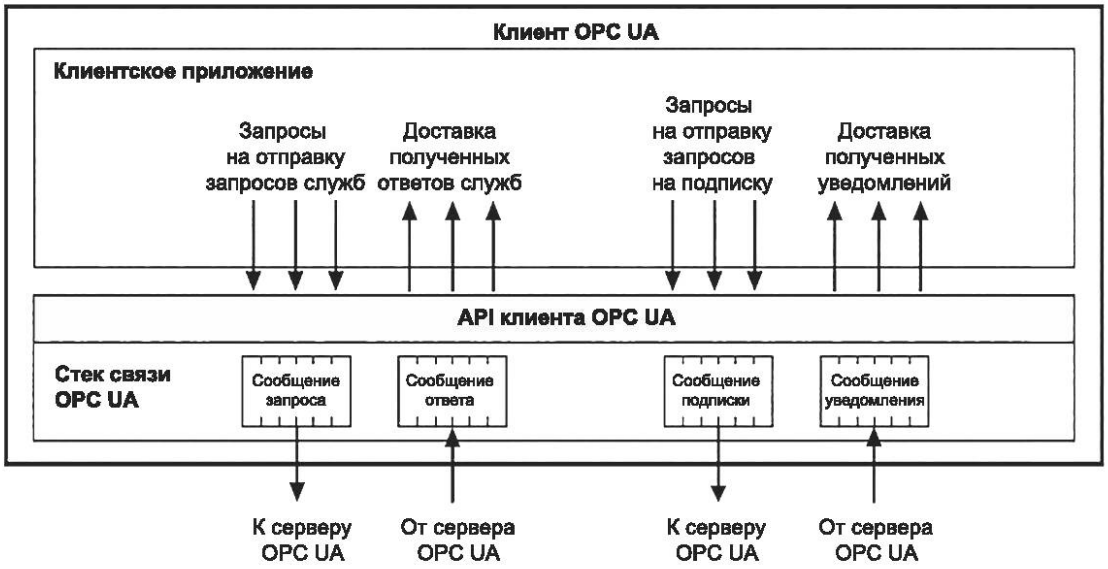

Рисунок 4 — Архитектура клиента OPC UA

Клиентское приложение — это код, который реализует функцию клиента. Оно использует API клиента для отправки и получения запросов служб OPC UA и ответов на сервер. Службы OPC UA описаны в 6.4 и определены в ГОСТ Р 71809.

«API Клиента» является внутренним интерфейсом, который изолирует код клиентского приложения от стека связи OPC UA. Стек связи OPC UA преобразует вызовы API клиента в сообщения и отправляет их через базовую сущность связи на сервер по запросу клиентского приложения. Стек связи OPC UA также получает ответ и сообщения уведомления от базовой сущности связи и доставляет их клиентскому приложению через API клиента.

6.3 Серверы OPC UA

6.3.1 Общие положения

Архитектура сервера OPC UA моделирует конечную точку сервера во взаимодействиях клиента и сервера. На рисунке 5 показаны основные элементы сервера и их взаимосвязь между собой.

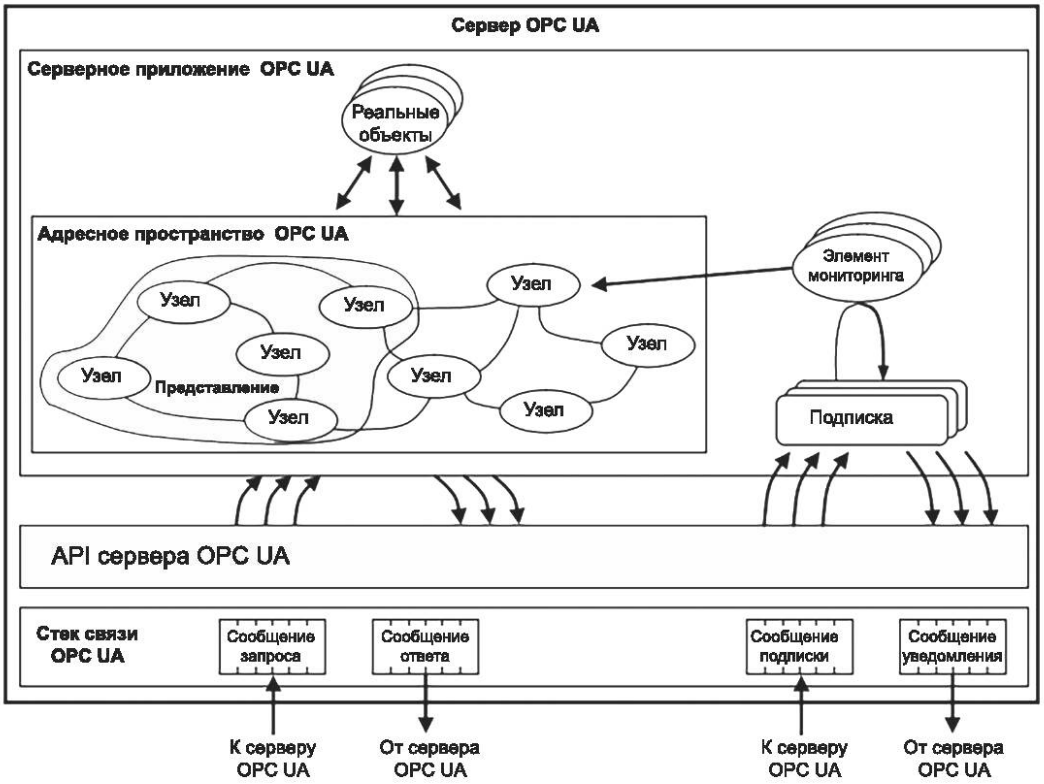

Рисунок 5 — Архитектура сервера OPC UA

6.3.2 Реальные объекты

Реальные объекты — это физические или программные объекты, доступные серверному приложению, или которые оно обслуживает внутри. Примерами являются физические устройства и диагностические счетчики.

6.3.3 Серверное приложение

Серверное приложение — это код, реализующий функцию сервера. Оно использует API сервера для отправки и получения сообщений OPC UA от клиентов. «API сервера» является внутренним интерфейсом, который изолирует код приложения сервера от стека связи OPC UA.

6.3.4 Адресное пространство OPC UA

6.3.4.1 Узлы адресного пространства

Адресное пространство моделируется как набор узлов, доступных клиентам с использованием служб (интерфейсов и методов) OPC UA. Узлы в адресном пространстве используются для представления реальных объектов, их определений и ссылок друг на друга.

6.3.4.2 Организация адресного пространства

Для последовательного создания адресного пространства из взаимосвязанных узлов ГОСТ Р 71808 содержит детали «строительных блоков» метамодели. Серверы могут свободно организовывать свои узлы в адресном пространстве. Использование ссылок между узлами позволяет серверам организовывать адресное пространство в иерархию, полную ячеистую сеть узлов или любое возможное сочетание.

В адресном пространстве OPC UA определяет узлы, ссылки и их организацию. Некоторые профили не требуют реализации всех узлов UA согласно ГОСТ Р 71810.

6.3.4.3 Представления адресного пространства

Представление — подмножество адресного пространства. Представления используются для ограничения узлов, которые сервер делает видимыми для клиента, тем самым ограничивая размер адресного пространства для запросов на обслуживание, отправляемых клиентом. Представлением по умолчанию является все адресное пространство. Серверы опционально определяют другие представления. Представления видны через адресное пространство, и клиенты могут просматривать представления, чтобы определить их структуру. Представления часто являют собой иерархии, которые удобны клиентам для навигации и представления в виде дерева.

6.3.4.4 Поддержка информационных моделей

Адресное пространство OPC UA поддерживает информационные модели согласно ГОСТ Р 71810. Эта поддержка обеспечивается через:

- ссылки на узлы, которые позволяют объектам в адресном пространстве быть связанными друг с другом;
- узлы типов объекта, которые предоставляют семантическую информацию для реальных объектов (определения типов);
- узлы типов объектов для поддержки подклассов определений типов;
- определения типов данных, представленных в адресном пространстве, которые позволяют использовать отраслевые типы данных;
- сопутствующие стандарты OPC UA, позволяющие отраслевым группам определять представление конкретных информационных моделей в адресном пространстве сервера.

6.3.5 Сущности подписки

6.3.5.1 Элементы мониторинга

Элементы мониторинга — сущности на сервере, созданные клиентом, которые отслеживают узлы адресного пространства и их реальные аналоги. При обнаружении изменения данных или возникновения события/сигнала предупреждения, они генерируют уведомление, которое передается клиенту посредством подписки.

6.3.5.2 Подписки

Подписка — конечная точка на сервере, которая публикует уведомления для клиентов. Клиенты контролируют скорость публикаций путем отправки сообщения публикации.

6.3.6 Интерфейс службы OPC UA

6.3.6.1 Общие положения

Службы, определенные для OPC UA, представлены в 6.4 и определены в ГОСТ Р 71809.

6.3.6.2 Службы запроса/ответа

Службы запроса/ответа — задачи на одном или нескольких узлах в адресном пространстве и для возврата ответа.

6.3.6.3 Службы подписки

Служба подписки вызывается через интерфейс службы OPC UA с целью периодической отправки уведомлений клиентам. Уведомления включают события, сигналы предупреждения, изменения данных и выходные данные программ.

6.3.7 Межсерверные взаимодействия

Межсерверное взаимодействие в модели «клиент-сервер» — это взаимодействие, при котором один сервер выступает в качестве клиента другого сервера. Межсерверное взаимодействие позволяет разрабатывать серверы, которые:

а) обмениваются информацией друг с другом на одноранговой основе. Реализация данной технологии может включать избыточные или удаленные серверы, которые используются для поддержки определений общесистемного типа (см. рисунок 6);

б) объединяются в многоуровневую архитектуру серверов для обеспечения:

- агрегации данных с серверов нижнего уровня;
- высокоуровневой структуры данных для клиентов;
- клиентские интерфейсы концентратора, обеспечивающие единую точку доступа к нескольким базовым серверам.

Взаимодействие между серверами представлено на рисунке 6.

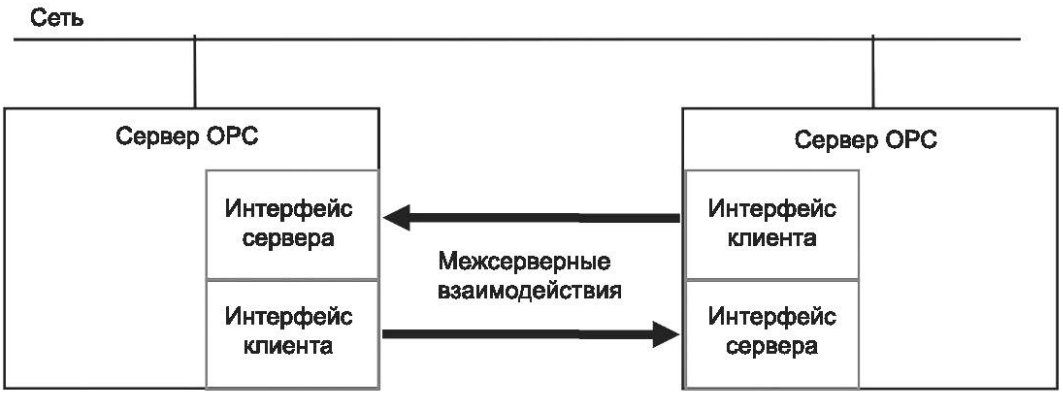

Рисунок 6 — Одноранговое взаимодействие между серверами

Подобные одноранговые взаимодействия могут быть реализованы с помощью модели OPC UA «издатель-подписчик», где каждое одноранговое приложение может одновременно являться как издателем, так и подписчиком.

Схема объединения серверов для вертикального доступа к данным на предприятии представлена на рисунке 7.

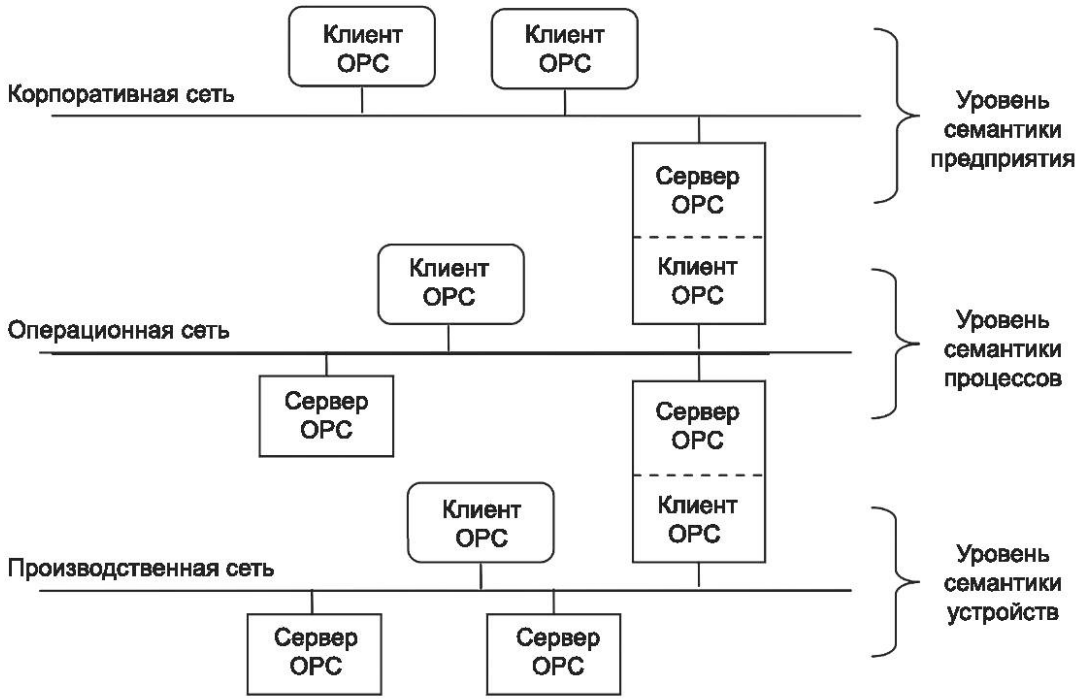

Рисунок 7 — Пример объединения серверов

6.4 Резервирование

OPC UA предоставляет структуры данных и службы, которые с применением стандартизированных характеристик определяют возможность резервирования. Резервирование применимо для повышения доступности, отказоустойчивости и балансировки нагрузки. Резервирование клиента, службы и

сети формально определено в ГОСТ Р 71809. Профилями, представленными в [7], определяется как поддерживается резервирование приложением OPC UA.

Требуемые сигналы, поступающие от клиента и сервера, связаны с двумя режимами резервирования сервера: прозрачным и непрозрачным. Обязанности клиента и сервера при использовании прозрачного или непрозрачного резервирования определены в ГОСТ Р 71809.

Серверы, поддерживающие непрозрачное резервирование, могут также поддерживать управляемое клиентом распределение нагрузки. Состояние сервера, включая его способность обслуживать запросы, в совокупности определяется как уровень службы. Определение уровня службы приведено в ГОСТ Р 71809. В ГОСТ Р 71810 представлены четыре поддиапазона уровня службы и примеры их использования.

6.5 Публикация-подписка

При использовании модели «издатель-подписчик» приложения OPC UA не обмениваются запросами и ответами напрямую [8]. Вместо этого, издатели посылают сообщения связующему ПО, ориентированному на обработку сообщений, не зная о том, доступны ли там какие-либо подписчики. Аналогично, подписчики выражают интерес к определенным типам данных, и обрабатывают сообщения, содержащие эти данные, без учета возможностей и функций издателей.

Связующее ПО, ориентированное на обработку сообщений — это программная или аппаратная инфраструктура, поддерживающая отправку и получение сообщений между распределенными системами. От связующего ПО, ориентированного на обработку сообщений, зависит, как будет реализовано это распределение.

Чтобы охватить большое количество способов использования, модель OPC UA «издатель-подписчик» поддерживает варианты связующего ПО, ориентированного на обработку сообщений [2]. К ним относятся:

- формат без брокера, где связующее ПО, ориентированное на обработку сообщений, является сетевой инфраструктурой, способной направлять сообщения на основе датаграм. Подписчики и издатели используют датаграммные протоколы, такие как мультикаст UDP;
- формат, основанный на брокерах, где связующее ПО, ориентированное на обработку сообщений, является брокером. Подписчики и издатели используют стандартные протоколы обмена сообщениями, такие как AMQP или MQTT, для связи с брокером. Все сообщения публикуются в очереди (например, темы, узлы), которые раскрывает брокер, а подписчики эти очереди прослушивают. Брокер переводит сообщения из формального протокола обмена сообщениями издателя в формальный протокол обмена сообщениями подписчика.

Модель «издатель-подписчик» используется для передачи сообщений между различными компонентами системы, при этом компонентам не нужно знать идентификаторы друг друга.

Издатель заранее настроен на отправку конкретных данных. Установления соединения между издателем и подписчиком не происходит.

Знание о том, кто является подписчиками, и пересылка опубликованных данных подписчикам перекладывается на связующее ПО, ориентированного на обработку сообщений. Издатель не знает и должен определять наличие и число подписчиков. Требования к усилиям и ресурсам издателя предсказуемы и не зависят от количества подписчиков. [8] описывает детали модели OPC UA «издатель-подписчик».

6.6 Взаимодействие моделей

Модели «издатель-подписчик» и «клиент-сервер» основаны на информационной модели OPC UA. Модель «издатель-подписчик» легко внедряется как серверу, так и клиенту, где издатель является сервером (владельцем информации), а подписчик — клиентом. С помощью модели OPC UA «клиент-сервер» информационная модель конфигурации «издатель-подписчик» способствует конфигурации издателей и подписчиков. На рисунке 8 показано одно приложение OPC UA, которое выступает в роли и сервера, и издателя.

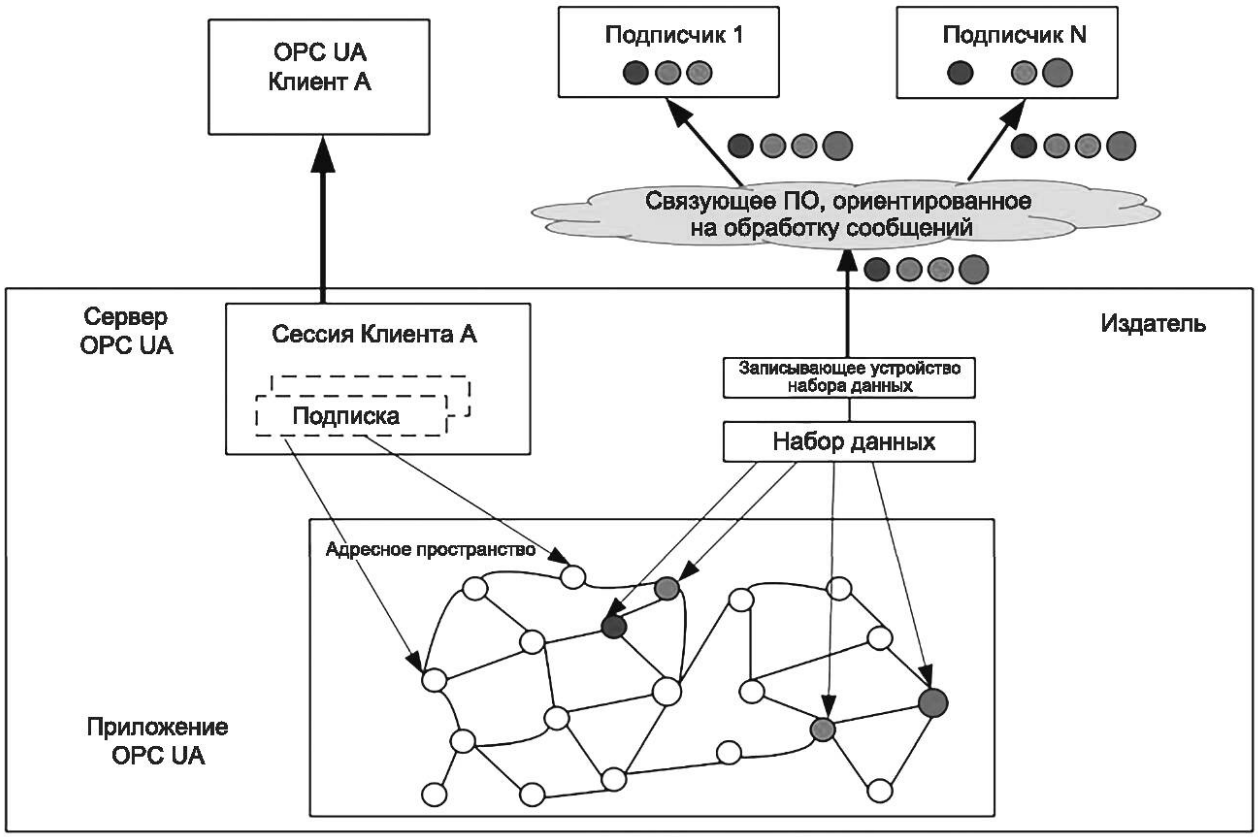

Рисунок 8 — Интегрированные модели «клиент-сервер» и «издатель-подписчик»

7 Наборы служб

7.1 Общие положения

Службы OPC UA разделены на наборы служб, каждый из которых определяет логическую группу служб для доступа к аспекту сервера. Наборы служб описаны в 7.2—7.11. Наборы служб и их службы определены в ГОСТ Р 71809. Поддерживает ли сервер набор служб или конкретную услугу в наборе служб, определяется его профилем. Профили представлены в [7].

7.2 Набор служб обнаружения

Данный набор служб определяет службы, используемые для обнаружения в системе доступных серверов. Он также обеспечивает способ, с помощью которого клиенты могут прочитать конфигурацию безопасности, необходимую для подключения к серверу. Службы обнаружения реализуются отдельными серверами и выделенными серверами обнаружения. Хорошо известные выделенные серверы обнаружения предоставляют клиентам возможность обнаружить все зарегистрированные серверы. В [5] описано, как использовать службы обнаружения с выделенными серверами обнаружения.

7.3 Набор служб защищенного канала

Данный набор служб определяет службы, используемые для открытия канала связи, который обеспечивает конфиденциальность и целостность всех сообщений, передаваемых на сервер. Базовые концепции безопасности UA определены в ГОСТ Р 71807.

Службы защищенного канала отличаются от других служб, поскольку они не реализуются непосредственно приложением OPC UA. Вместо этого они предоставляются стеком связи, на котором построено приложение OPC UA. Приложения OPC UA просто должны проверять что безопасный канал активен, каждый раз, когда они получают сообщение. ГОСТ Р 71811 описывает применение служб защищенного канала в различных типах стеков связи.

Защищенный канал — длительное логическое соединение между одним клиентом и одним сервером. Данный канал поддерживает набор ключей, которые известны только клиенту и серверу и используются для аутентификации и шифрования сообщений, передаваемых по сети. Службы защищенного канала позволяют клиенту и серверу безопасно согласовывать используемые ключи.

Точные алгоритмы, используемые для аутентификации и шифрования сообщений, описаны в политиках безопасности сервера. Такие политики открываются с помощью набора служб обнаружения. Клиент выбирает соответствующую конечную точку, которая поддерживает желаемую политику безопасности сервера при создании защищенного канала.

При общении через защищенный канал клиент и сервер проверяют, все ли входящие сообщения были подписаны и/или зашифрованы в соответствии с политикой безопасности. Приложение UA должно проигнорировать любое сообщение, не соответствующее политике безопасности канала.

Безопасный канал отделен от сеанса UA-приложения; доступ к сеансу UA-приложения может осуществляться только через безопасный канал. UA-приложение определяет, какой <защищенный канал> связан с каждым сообщением. Стек связи, обеспечивающий механизм защищенного канала, но не позволяющий приложению узнать, какой <защищенный канал> был использован для данного сообщения, не может быть использован для реализации набора служб <защищенный канал>.

Взаимосвязь между сеансом приложения UA и защищенным каналом показана на рисунке 9. Приложения UA используют стек связи для обмена сообщениями. Службы защищенного канала применяются для установления <защищенного канала> между двумя стеками связи, что позволяет им безопасно обмениваться сообщениями. Приложения UA используют набор служб сессии для установления сессии приложения UA.

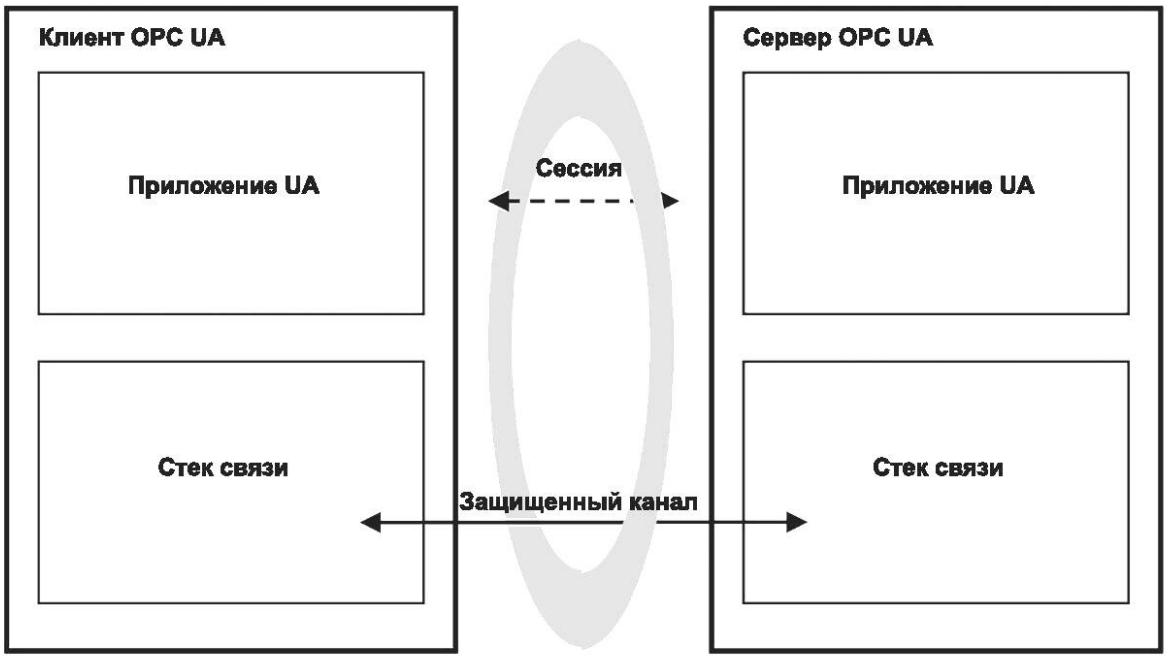

Рисунок 9 — Защищенный канал и службы сессии

7.4 Набор служб сессии

Набор служб определяет службы, используемые для установления соединения на прикладном уровне в контексте сессии от имени конкретного пользователя.

7.5 Набор служб управления узлами

Набор служб управления узлами позволяет клиентам добавлять, изменять и удалять узлы в адресном пространстве службы предоставляют интерфейс для конфигурирования серверов согласно ГОСТ Р 71809.

7.6 Набор служб представлений

Представления являются общедоступными, создаваемыми сервером подмножествами адресного пространства. Все адресное пространство по умолчанию, является представлением и поэтому службы представлений работают со всем адресным пространством. Развитие версий этой спецификации предполагает работу службы для создания определяемых клиентом представлений.

Набор служб представлений позволяет клиентам обнаруживать узлы в представлении путем просмотра. Просмотр позволяет клиентам перемещаться вверх и вниз по иерархии или следовать ссылкам между узлами, содержащимися в представлении. Таким образом, просмотр также позволяет клиентам обнаруживать структуру представления.

7.7 Набор служб запросов

Набор служб запросов позволяет пользователям получать доступ к адресному пространству без просмотра и без знания логической структуры для внутреннего хранения данных.

Запрос позволяет клиентам выбирать подмножество узлов в представлении на основе некоторых критериев фильтра, предоставляемых клиентом. Узлы, выбранные из представления в результате выполнения запроса, называются набором результатов.

Серверы могут столкнуться с трудностями при обработке запросов, требующих доступа к данным во время исполнения, например к данным устройства. Такая обработка связана с ресурсоемкими операциями или значительными задержками. В таких случаях сервер может счесть необходимым отклонение запроса.

7.8 Набор служб атрибутов

Набор служб атрибутов используется для чтения и записи значений атрибутов. Атрибуты являются простейшими характеристиками узлов OPC UA. Они не могут быть определены клиентами или серверами. Атрибуты — единственные элементы адресного пространства, которым допустимо иметь значения данных. Особый атрибут, атрибут значения, используется для определения значения переменных.

7.9 Набор служб методов

Методы представляют собой вызовы функций объектов. Они определены в настоящем стандарте и ГОСТ Р 71808. Методы вызываются и возвращаются после завершения, неважно, успешного или неудачного. Время применения методов может варьироваться в зависимости от выполняемой ими функции.

Набор служб методов определяет средства вызова методов. Метод всегда является компонентом объекта. Обнаружение осуществляется с помощью служб просмотра и запроса. Клиенты обнаруживают методы, поддерживаемые сервером, просматривая имеющиеся объекты, которые идентифицируют поддерживаемые ими методы.

Поскольку методы могут контролировать некоторые аспекты производственной работы, вызов метода зависит от воздействий окружающей среды или других условий. Это особенно актуально при попытке повторного вызова метода сразу после завершения его использования. Условия, необходимые для вызова метода, могут не вернуться в состояние, позволяющее повторный запуск метода. Кроме того, некоторые методы могут поддерживать одновременные вызовы, в то время как другие могут иметь только один вызов, выполняемый в данный момент времени.

7.10 Набор служб элементов мониторинга

Набор служб элементов мониторинга используется клиентом для создания и обслуживания элементов мониторинга. Элементы мониторинга отслеживают переменные, атрибуты и уведомители событий. Они отслеживают переменные на предмет изменения значения или состояния, атрибуты — на предмет изменения значения, а уведомители событий — на предмет вновь созданных отчетов о сигналах предупреждения и событиях.

Каждый элемент мониторинга идентифицирует элемент для мониторинга и подписку, которая будет использоваться для периодической публикации уведомлений для клиента (см. 7.11). Каждый элемент мониторинга также определяет скорость мониторинга (измерения) элемента, а для переменных и уведомителей событий — критерии фильтрации для генерации уведомления. Критерии фильтра для атрибутов задаются их определениями атрибутов согласно ГОСТ Р 71809.

Если частота измерения элемента мониторинга выше, чем частота публикации подписки, то элемент мониторинга должен быть настроен на очередь всех уведомлений или на очередь только последних уведомлений для передачи по подписке. В последнем случае размер очереди равен единице.

Службы элементов мониторинга также определяют режим мониторинга. Режим мониторинга настраивается так, чтобы отключить измерения и отчетность, включить только измерения или включить и измерения, и отчетность. Если измерения включены, сервер производит измерения элемента. Каждое измерение оценивается, чтобы определить, следует ли генерировать уведомление. Если да, то уведомление ставится в очередь. Если включена функция отчетности, очередь становится доступной подписке для передачи.

Элементы мониторинга должны быть настроены для запуска отчетов других элементов мониторинга. В этом случае режим мониторинга элементов отчета устанавливается только на измерение, и когда инициирующий элемент генерирует уведомление, любые уведомления в очереди для элементов отчета становятся доступными подписке для передачи.

7.11 Набор служб подписки

Набор служб подписки используется клиентом для создания и обслуживания подписок. Подписки — это сущности, которые периодически публикуют сообщения уведомления для назначенного им элемента мониторинга (см. 7.10). Сообщение уведомления содержит общий заголовок, за которым следует серия уведомлений. Формат уведомлений зависит от типа элемента мониторинга (то есть переменных, атрибутов и уведомлений о событиях).

После создания существование подписки не зависит от сессии клиента с сервером. Это позволяет одному клиенту создать подписку, а второму, возможно, резервному клиенту, получать от нее сообщения уведомления.

Для защиты от неиспользования клиентами подписки имеют настроенный срок службы, который клиенты периодически обновляют. Если какой-либо клиент не может продлить срок действия, срок действия истекает, и сервер закрывает подписку. Когда подписка закрыта, все элементы мониторинга, назначенные подписке, удаляются.

Подписки включают функции, поддерживающие обнаружение и восстановление потерянных сообщений. Каждое сообщение уведомления содержит порядковый номер, который позволяет клиентам обнаружить пропущенные сообщения. Если в течение интервала поддержания активности нет уведомлений для отправки, сервер отправляет сообщение подтверждения активности, содержащее порядковый номер следующего отправленного сообщения уведомления. Если клиенту не удается получить сообщение по истечении интервала поддержания активности или если он обнаружил пропуск сообщения, то он должен запросить у сервера повторную отправку одного или нескольких сообщений.

Библиография

[1] МЭК 62541-8:2020 Унифицированная архитектура OPC. Часть 8. Доступ к данным (OPC Unified Architecture — Part 8: Data Access)

[2] МЭК 62541-9:2020 Унифицированная архитектура OPC. Часть 9. Аварийные сигналы и условия (OPC Unified Architecture — Part 9: Alarms and Conditions)

[3] МЭК 62541-10:2020 Унифицированная архитектура OPC. Часть 10. Программы (OPC Unified Architecture — Part 10: Programs)

[4] МЭК 62541-11:2020 Унифицированная архитектура OPC. Часть 11. Исторический доступ (OPC Unified Architecture — Part 11: Historical Access)

[5] МЭК 62541-12:2020 Унифицированная архитектура OPC. Часть 12. Обнаружение и глобальные службы (OPC unified architecture — Part 12: Discovery and global services)

[6] МЭК 62541-13:2020 Унифицированная архитектура OPC. Часть 13. Агрегаты (OPC Unified Architecture — Part 13: Aggregates)

[7] МЭК 62541-7:2020 Унифицированная архитектура OPC. Часть 7. Профили (OPC unified architecture — Part 7: Profiles)

[8] МЭК 62541-14:2020 Унифицированная архитектура OPC. Часть 14. PubSub (OPC unified architecture — Part 14: PubSub)

УДК 004.03:006.354

ОКС 35.240.99
31.180

Ключевые слова: цифровая промышленность, открытая платформа взаимодействия, унифицированная архитектура, доступ к данным, сервер, плата ввода-вывода

Редактор *Н.А. Аргунова*
Технический редактор *И.Е. Черепкова*
Корректор *И.А. Королева*
Компьютерная верстка *Е.А. Кондрашовой*

Сдано в набор 20.12.2024. Подписано в печать 09.01.2025. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 3,26. Уч.-изд. л. 2,77.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
www.gostinfo.ru info@gostinfo.ru