

1 Структура программного обеспечения

В данном разделе описывается структура программного обеспечения ПУМ.

ПУМ состоит из следующих подсистем:

а) Подсистема управления:

- 1) Модуль управления ПДУ и КРК
- 2) Модуль управления трафиком, топологией и технологическими параметрами .

б) Подсистема мониторинга:

- 1) Модуль мониторинга ПДУ и КРК;
- 2) Модуль мониторинга оборудования ККП;

в) Подсистема технического учёта:

- 1) Модуль учёта оборудования;
- 2) Модуль учёта логических ресурсов;

г) Подсистема взаимодействия с внешними системами:

- 1) Модуль экспорта данных во внешние системы;
- 2) Модуль обработки запросов API в реальном времени;

д) Подсистема контроля доступа;

е) АРМ администратора;

ж) ЛСУ

- 1) Агенты ЛСУ

Ине. № подл.	Подп. и дата
Взам. инв. №	Ине. № дубл.
Подп. и дата	

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					6

Модуль управления трафиком, топологией и технологическими параметрами должен обеспечивать:

- а) Формирование топологии сети в графическом виде, отображение всех сетевых узлов и связей между ними;
- б) управление ресурсами ККП:
 - 1) автоматический расчет емкости сети
 - 2) определение технической возможности включения новых потребителей (клиентов) ККП без ухудшения параметров и уровня обслуживания для подключенных ранее потребителей (клиентов)
 - 3) приоритезация пользователей и трафика. Устанавливаются три уровня приоритета: низкий, обычный, высокий. Трафик ключевой информации должен всегда иметь высокий приоритет. Принципы управления трафиком должны строиться на использовании резервных каналов передачи данных и управлением их пропускной способностью.

Установка модуля управления трафиком, топологией и технологическими параметрами МКС производится в ЦУМ.

- в) Устанавливаются три уровня приоритета: низкий, обычный, высокий.
- г) Трафик ключевых данных всегда имеет высокий приоритет.
- д) Принципы управления трафиком строятся на использовании резервных каналов передачи данных (при их наличии) и управлением их пропускной способностью.

Модуль управления трафиком, топологией и технологическими параметрами ККП состоит из двух частей: для выполнения на ЦУМ и для выполнения на ЛСУ.

- а) Функционал модуля в ЦУМ обеспечивает:
 - 1) Инициализацию модуля;

Име. № подл.	
Подп. и дата	
Взам. инв. №	
Име. № дубл.	
Подп. и дата	

Изм.	Лист	№ докум.	Подп.	Дата	

Модуль мониторинга ПДУ и КРК обеспечивает прием следующих данных из ВС:

- прием данных о результатах обработки событий в привязке к инцидентам и работам для информирования администратора ПУМ. ВС направляет в модуль мониторинга данные о всех работах и/или инцидентах, связанных с событиями, которые модуль мониторинга оборудования ранее отправил в ВС.

Установка модуля мониторинга ПДУ и КРК производится в ЦУМ.

2.2.2 Модуль мониторинга оборудования ККП

Модуль мониторинга оборудования ККП, обеспечивает мониторинг, сбор, хранение и обработку следующей информации ККП:

- температуры внутри аппаратных модулей узлов ККП;
- результаты контроля работоспособности вентиляторов и источников питания;
- параметры утилизации ресурсов;
- результатов тестирования программных компонентов ПКРК,
- иной информации, определенной в эксплуатационной документации на КРК (покупное изделие), используемых в ПКРК ККП в соответствии с руководством по эксплуатации производителя (с учётом требований Регулятора).

Модуль мониторинга оборудования обеспечивает визуализацию (вывод через единый графический интерфейс администратора сети КРК) следующих параметров ККП:

- скорость генерации квантовых ключей между каждой парой ПОУ и ОУ и сравнение с заданным порогом SLA для каждого из клиентских сервисов;
- QBER между каждой парой ПОУ и ОУ;

Име. № подл.	Подп. и дата
Взам. инв. №	Ине. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					10

- частота обновления КЗК между парой узлов;
- иную информацию.

Модуль мониторинга оборудования взаимодействует с внешней системой ВС, которая предназначена для обеспечения единого непрерывного жизненного цикла управления технологическими сетями передачи данных.

Модуль мониторинга оборудования обеспечивает передачу следующих данных в ВС:

- передачу событий на основе данных мониторинга оборудования в привязке к конфигурационным единицам (событие – это любой выход значений параметров любого компонента за допустимые пределы и /или расхождение в составе оборудования вПУМ и ВС).

Модуль мониторинга оборудования обеспечивает прием следующих данных из ВС:

- прием данных о результатах обработки событий в привязке к инцидентам и работам для информирования администратора ПУМ. ВС направляет в модуль мониторинга данные о всех работах и/или инцидентах, связанных с событиями, которые модуль мониторинга оборудования ранее отправил в ВС.

Установка модуля мониторинга оборудования производится в ЦУМ.

2.3 Подсистема технического учёта

2.3.1 Модуль учета оборудования

Модуль учета оборудования осуществляет:

- автоматическую загрузку состава оборудования ККП с возможностью заполнения места расположения, координат,

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					11

номеров коммерческих заказов и прочей дополнительной информации.

- создания объектов, загруженных из внешних систем с возможностью внесения описания.

Модуль учета оборудования взаимодействует с внешней системой ВС, которая предназначена для обеспечения единого непрерывного жизненного цикла управления технологическими сетями передачи данных. Компоненты, оборудование, соединения компонентов описываются в ВС. Все описания из ВС реплицируются в модуле учета оборудования.

Модуль учета оборудования обеспечивает передачу следующих данных в ВС:

- регулярную передачу инвентаризационных данных, полученных с оборудования.

Модуль учета оборудования должен обеспечивать прием следующих данных в ВС:

- прием данных о типах, моделях и экземплярах конфигурационных единиц, соединениях между конфигурационными единицами.

Установка модуля учета оборудования производится в ЦУМ.

2.3.2 Модуль учета логических ресурсов

Модуль учета логических ресурсов осуществляет:

- автоматическую загрузку состава логических ресурсов ККП с возможностью заполнения дополнительной информации.
- создания объектов, загруженных из внешних систем с возможностью внесения описания.

Модуль учета логических ресурсов взаимодействует с внешней системой ВС, которая предназначена для обеспечения единого непрерывного жизненного цикла управления технологическими сетями передачи данных.

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					12

- Обработка запроса с использованием диагностической информации с соответствующих подсистем и компонентов ;
- Возврат полученных в ходе обработке данных инициатору запроса.

Запросы формируются в формате JSON (YAML). Содержательные поля запроса:

Поле запроса	Значение поля запроса
TIMESTAMP	Дата формирования запроса в секундах, начиная с 1.01.1970 (UNIX timestamp)
SYSNAME	Имя устройства PDU согласно установленной нотации
PARAMNAME_1	Название параметра №1
...	...
PARAMNAME_n	Название параметра №n

Формат ответа:

Поле ответа	Значение поля ответа
TIMESTAMP	Дата формирования ответа в секундах, начиная с 1.01.1970 (UNIX timestamp)
SYSNAME	Имя устройства PDU согласно установленной нотации
PARAMNAME_1	Значение параметра №1
...	...
PARAMNAME_n	Значение параметра №n

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

- доступ пользователей в систему через графический интерфейс автоматизированного рабочего места;
- регистрацию пользователя (администратора) в системе с использованием подтверждения со стороны администратора и назначения группы, в которую включён пользователь. Данные пользователя автоматически вносятся в OpenLDAP;
- контроль учетных данных пользователей;
- внесение изменений в учётные данные пользователей через графический интерфейс;
- определение принадлежности пользователей к группам;
- запись событий входа и взаимодействия с системой в системные логи;
- блокировка и разблокировка пользователей;
- автоматизация создания учетных данных в системах OpenLDAP и Kerberos для дальнейшего использования устройствами системы;
- интеграция в модули управления и мониторинга квантовой коммуникационной платформы;
- единую точку входа по протоколу CAS 2.0, в которой используются данные из OpenLDAP.
- обеспечение отказоустойчивой работы за счёт репликации данных.

Установка подсистемы контроля доступа производится в ЦУМ.

2.6 ЛСУ и Агенты ЛСУ

На Локальном сервере управления (далее по тексту ЛСУ) устанавливаются программные агенты, которые выполняют функции, соответствующие наименованиям агентов:

Ине. № подл.	Подп. и дата	Взам. инв. №	Ине. № дубл.	Подп. и дата						Лист
										19
					Изм.	Лист	№ докум.	Подп.	Дата	

- Агент модуля учёта логических ресурсов основанный на заимствованном ПО fusion-inventory
- Агент подсистемы мониторинга, основанный на заимствованном ПО Zabbix-agent
- Агент подсистемы управления, включая модуль взаимодействия с СПДУ

Все агенты, выполняющиеся на ЛСУ, взаимодействуют с ЦУМ напрямую и не производят сохранение каких-либо данных непосредственно на ЛСУ.

Установка агентов ЛСУ производится на ЛСУ.

2.7 ПО АРМ администратора

Автоматизированное рабочее место администратора (АРМ администратора) реализует функции взаимодействия ККП с администраторами с помощью графического интерфейса. Графический интерфейс реализуется в браузере администратора и может быть выполнен на рабочем месте – ПК, имеющем доступ к сети управления. Серверная часть ПО АРМ администратора выполняется на ЦУМ и позволяет производить авторизацию пользователей, отображать окна графического интерфейса с учётом доступных ролей в системе за счёт использования подсистемы контроля доступа.

ПО АРМ администратора обеспечивает:

- генерацию окон графического интерфейса
- отображение окон в браузере администратора по запросу
- отправку действий пользователя на исполнение подсистеме управления ККП
- отображение результата выполнения действий подсистемой управления ККП в графическом виде
- интеграцию с подсистемой мониторинга и технического учёта для отображения их данных

Установка ПО АРМ администратора производится в ЦУМ.

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					20

3 Методы и средства разработки программного обеспечения

3.1 Подсистема управления

3.1.1 Модуль управления ПДУ и КРК

Модуль проектируется и разрабатывается с использованием только свободно распространяемых компонент в среде ОС Linux на языке ООП Python3. При этом используется набор стандартных свободно распространяемых библиотек на языке Python, а также набор разрабатываемых библиотек.

3.1.2 Модуль управления трафиком, топологией и технологическими параметрами

Модуль проектируется и разрабатывается с использованием только свободно распространяемых компонент в среде ОС Linux на языках программирования – perl, bash, C. При этом используется набор стандартных свободно распространяемых библиотек, а также набор разрабатываемых нами библиотек.

3.2 Подсистема мониторинга

3.2.1 Модуль мониторинга ПДУ и КРК

При проектировании и разработке программного обеспечения ПУМ ККП предпочтение отдавалось свободно распространяемым программным продуктам, уже доказавшим свою пригодность при решении подобных задач.

Для осуществления автоматической диагностики, мониторинга и формирования предупреждений в реальном времени по событиям в ККП использовалась заимствованная система Zabbix. Все полученные данные мониторинга хранятся в базе данных Zabbix'a. Для отслеживания отклонения данных от приемлемых значений используются Zabbix триггеры. При срабатывании триггера создается событие и выполняются действия по уведомлению о данном событии.

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					21

В качестве языка разработки программного обеспечения используется язык python в среде Linux. Для передачи данных мониторинга ПДУ и КРК на Zabbix сервер используется модуль Zabbix'a для python «py-zabbix». Для формирования списка аварий использовалась заимствованная система GLPI, которая получает сведения об авариях из Zabbix с помощью программного пакета для python «glpi-sdk-python».

Для взаимодействия с внешней системой ВС используется API интерфейс через брокер сообщений Kafka (служебный модуль коммутации данных ВС) и Apache ServiceMix.

3.2.2 Модуль мониторинга оборудования

При проектировании и разработке программного обеспечения ПУМ ККП предпочтение отдавалось свободно распространяемым программным продуктам, уже доказавшим свою пригодность при решении подобных задач.

Для обеспечения мониторинга, сбора, хранения и обработки информации ККП использовалась заимствованная система Zabbix. Все полученные данные мониторинга хранятся в базе данных Zabbix'a. Для отслеживания отклонения данных от приемлемых значений используются Zabbix триггеры. При срабатывании триггера создается событие и выполняются действия по уведомлению о данном событии.

В качестве языка разработки программного обеспечения использовался python в среде Linux. Для передачи в Zabbix сервер данных мониторинга узлов и оборудования, установленного на ККП, используется модуль Zabbix'a для python «py-zabbix». Для формирования списка аварий используется заимствованная система GLPI, которая получает сведения об авариях из Zabbix с помощью программного пакета для python «glpi-sdk-python».

Для взаимодействия с внешней системой ВС используется API интерфейс через брокер сообщений Kafka (служебный модуль коммутации данных ВС) и Apache ServiceMix.

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					22

3.3 Подсистема технического учёта

3.3.1 Модуль учета оборудования

При проектировании и разработке программного обеспечения ПУМ ККП предпочтение отдавалось свободно распространяемым программным продуктам, уже доказавшим свою пригодность при решении подобных задач.

Создание модуля учета оборудования выполнялось с помощью заимствованной системы GLPI. Для обнаружения состава оборудования ККП и его загрузки в GLPI использовался плагин Fusioninventory.

В качестве языка разработки программного обеспечения использовался python в среде Linux. Для создания объектов в GLPI, загруженных из внешних систем, использовался программный пакет для python «glpi-sdk-python».

Для взаимодействия с внешней системой BC используется API интерфейс, организованный с использованием брокера сообщений Kafka (служебный модуль коммутации данных BC) и Apache ServiceMix.

3.3.2 Модуль учета логических ресурсов

При проектировании и разработке программного обеспечения ПУМ ККП предпочтение отдавалось свободно распространяемым программным продуктам, уже доказавшим свою пригодность при решении подобных задач.

Создание модуля учета логических ресурсов выполнялось с помощью заимствованной системы GLPI. Для обнаружения состава логических ресурсов ККП и его загрузки в GLPI использовался плагин Fusioninventory.

В качестве языка разработки программного обеспечения использовался python в среде Linux. Для создания объектов в GLPI, загруженных из внешних систем, использовался программный пакет для python «glpi-sdk-python».

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					23

Для взаимодействия с внешней системой ВС используется API интерфейс, организованный с использованием брокера сообщений Kafka (служебный модуль коммутации данных ВС) и Apache ServiceMix.

3.4 Подсистема взаимодействия с внешними системами

3.4.1 Модуль экспорта данных во внешние системы

Модуль проектируется и разрабатывается с использованием только свободно распространяемых компонент в среде ОС Linux на языке ООП Python3. При этом используется набор стандартных свободно распространяемых библиотек на языке Python, а также набор разрабатываемых нами библиотек.

Для взаимодействия с внешней системой ВС используется API интерфейс, организованный с использованием брокера сообщений Kafka (служебный модуль коммутации данных ВС) и Apache ServiceMix.

3.4.2 Модуль обработки запросов API в реальном времени

Модуль проектируется и разрабатывается с использованием только свободно распространяемых компонент в среде ОС Linux на языке ООП Python3. При этом используется набор стандартных свободно распространяемых библиотек на языке Python, а также набор разрабатываемых нами библиотек.

Для приема запросов API от внешних систем в реальном времени используется брокер системы сообщений RabbitMQ. Для работы с брокером RabbitMQ на языке Python 3 используется библиотека pika.

3.5 Подсистема контроля доступа

Подсистема аутентификации использует в качестве контроллера и части клиентских библиотек заимствованный продукт Kerberos.

Подсистема авторизации использует в качестве контроллера и клиентских библиотек заимствованный продукт OpenLDAP.

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					24

Для обеспечения единой точки входа при авторизации в системе используется сервер CAS 2.0, который получает данных о пользователе от сервера OpenLDAP.

Создание подсистемы контроля доступа выполнено с использованием программного продукта nano из состава дистрибутива CentOS 8.

Управление изменениями подсистемы выполнено с использованием системы контроля версий git.

3.6 Агенты ЛСУ

Агенты каждой подсистемы, установленные на ЛСУ, разрабатываются теми же методами и средствами, которые применяются для разработки соответствующих подсистем. Методы и средства разработки агентов ЛСУ описаны в следующих подразделах:

- Агент модуля учёта логических ресурсов – в п. 3.2
- Агент подсистемы мониторинга – в п 3.3
- Агент подсистемы управления с модулем взаимодействия с СПДУ – в п 3.1.

3.7 ПО АРМ администратора

ПО АРМ администратора использует библиотеки Django на языке Python для генерации окон графического интерфейса. Окна графического интерфейса выполнены в виде шаблонов страниц формата HTML. Отображение данных в шаблонах производится с использованием информации из подсистемы управления и подсистемы мониторинга.

Отображение графического интерфейса выполняется с использованием браузера из состава дистрибутива CentOS 8.

Для обеспечения взаимодействия с подсистемой управления используется очередь сообщений rabbitmq.

Име. № подл.	Подп. и дата
Взам. инв. №	Име. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					25

Перечень принятых сокращений

API	Application programming interface
АРМ	Автоматизированное рабочее место
БД	База данных
ВМ	Виртуальная машина
КЗК	Квантово-защищенные ключи
ККП	Квантовая коммуникационная платформа цифровой экономики
КРК	Квантовое распределение ключей
ЛСУ	Локальный сервер управления
ПДУ	Подсистема организации доверенных опорных узлов квантовой связи
ПКРК	Подсистема квантового распределения ключей
ПО	Программное обеспечение
ПУМ	Подсистема управления и мониторинга квантовой сетью
СКЗИ	Система защиты информации
ЦУМ	Центр Управления и Мониторинга

Име. № подл.	
Подп. и дата	
Взам. инв. №	
Име. № дубл.	
Подп.	

Изм.	Лист	№ докум.	Подп.	Дата

Лист

28

Список используемых источников

- [1] “Техническое задание на разработку цифрового аппаратно-программного решения «Квантовая коммуникационная платформа цифровой экономики» к Договору от 27.03.2020 № 006/20.”
- [2] “Частное техническое задание «Разработка и создание подсистемы управления и мониторинга (ПУМ) квантовой коммуникационной платформы (ККП)», СЧ ОКР, шифр «Стрела-ПУМ».”
- [3] “CentOS Documentation.” <https://docs.centos.org/en-US/docs/>.
- [4] “Handbook:AMD64 Full Installation.”
<https://wiki.gentoo.org/wiki/Handbook:AMD64/Full/Installation/ru>.

Инев. № подл.	Подп. и дата	Взам. инв. №	Инев. № дубл.	Подп. и дата		Лист
						29
Изм.	Лист	№ докум.	Подп.	Дата		

