Общество с ограниченной ответственностью "СмартГеосистемс"

УДК: 004
Регистрационный №122012100320-0
Инв. №2023/01-1

УТВЕРЖДАЮ
Генеральный директор
Кобилов Т. А.

ОТЧЕТ
о выполнении НИОКР по теме:
«Разработка средств и сервисов информационной поддержки навигации автономных судов»
(договор №778ГРНТИС5/72421 от 23.12.2021)

(заключительный)

Руководитель работ Кобилов Т. А.

Москва, 2023

1. Реферат

ГЕОИНФОРМЦИОННАЯ СИСТЕМА (ПЛАТФОРМА) РЕАЛЬНОГО ВРЕМЕНИ GEOKASKAD MARINE.

Настоящий отчет подготовлен как итог выполнения работы по разработке средств и сервисов информационной поддержки навигации автономных судов.

Цель работы – разработка опытного образца геоинформационной системы (платформы) реального времени, выполняющей интеграцию гетерогенных сервисов морской отрасли в целях информационного обеспечения автономного судовождения данными радиолокационных систем (РЛС), Глобальной морской системы связи при бедствии (ГМССБ), автоматизированной идентификационной системы (АИС) или автоматической системы обмена данными (АСОД), оптических систем наблюдения, погодных станций, а также геопривязанных данных дистанционного зондирования Земли ("космоснимков") и иных геопривязанных данных и сервисов, обеспечивающих работу бортовых и береговых навигационных систем.

Результатом выполнения работ является:

- опытный образец геоинформационной интегрирующей платформы, и входящие в него программные компоненты:
- модуль входных интерфейсов;
- модуль предварительной обработки и фильтрации данных;
- модуль выходных интерфейсов;
- модуль логирования и контроля метрик производительности работы системы;
- подсистема хранения данных;
- модуль администрирования;
- модуль комплексной обработки событий;
- модуль географических информационных систем.

На опытный образец геоинформационной интегрирующей платформы разработан комплект программной документации в составе:

- Описание программы;
- Руководство оператора;
- Руководство системного программиста;
- Программа и методика предварительных пилотных испытаний опытного образца гео-информационной интегрирующей платформы;
- Программа и методика приемочных испытаний опытного образца геоинформационной интегрирующей платформы
- Акт изготовления опытного образца геоинформационной интегрирующей платформы;
- Протоколы предварительных пилотных испытаний опытного образца геоинформационной платформы;
- Акт доработки опытного образца геоинформационной интегрирующей платформы;
- Протоколы приемочных испытаний опытного образца геоинформационной интегрирующей платформы.

На каждый программный компонент, входящий в состав опытного образца геоинформационной интегрирующей платформы разработан комплект программной документации в составе:

- Частное техническое задание на разработку программного модуля;
- Описание программы;
- Программы и методики испытаний (тестирования);
- Протоколы тестирования.

Исполнитель считает, что работы выполнены в срок, соответствуют требованиям

технического задания, и позволяют достичь плановых показателей реализации инновационного проекта.

Москва, 2023

СПИСОК ИСПОЛНИТЕЛЕЙ

Должность, ученая степень, ученое звание	Подпись	Инициалы и фамилия
Руководитель проекта, генеральный директор ООО «СмартГеосистемс», кандидат технических наук	 (подпись)	Т. А. Кобилов
Директор по развитию бизнеса ООО «СмартГеосистемс»	 (подпись)	О. Е. Бурынин
Технический директор ООО «СмартГеосистемс»	 (подпись)	Е. А. Никулин
Начальник отдела геоинформационных систем ООО «СмартГеосистемс»	 (подпись)	Л. В. Афанасьева
Аналитик отдела разработки программного обеспечения ООО «СмартГеосистемс»	 (подпись)	А. Ю. Филиппова

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ 3

ОПРЕДЕЛЕНИЯ 7

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 9

ВВЕДЕНИЕ 10

1 Сроки и порядок поэтапных испытаний 12

2 Разработка частных технических заданий на программные модули геоинформационной интегрирующей платформы 13

3 Разработка архитектуры и компонентов программного обеспечения Модуля входных интерфейсов 14

3.1 Назначение и краткое описание 14

3.2 Обоснование логической структуры 15

3.2.1 Обоснование используемых методов и алгоритмов 15

3.2.2 Компоненты модуля и взаимодействия между ними 15

3.2.3 Связи модуля с другими компонентами платформы 17

3.3 Программное обеспечение 17

3.3.1 Обоснование выбора языка программирования 17

3.3.2 Обоснование программной архитектуры 18

3.4 Разработка программного обеспечения 19

4 Разработка архитектуры и компонентов программного обеспечения Модуля предварительной обработки и фильтрации данных 24

4.1 Назначение и краткое описание 24

4.2 Обоснование логической структуры 25

4.2.1 Обоснование используемых методов и алгоритмов 25

4.2.2 Компоненты модуля и взаимодействия между ними 25

4.2.3 Связи модуля с другими компонентами платформы....27
4.3 Программное обеспечение....27
4.3.1 Обоснование выбора языка программирования....27
4.3.2 Обоснование программной архитектуры....28
4.3.3 Разработка программного обеспечения....28
5 Разработка архитектуры и компонентов программного обеспечения Модуля выходных интерфейсов....32
5.1 Назначение и краткое описание....32
5.2 Обоснование логической структуры....32
5.2.1 Обоснование используемых методов и алгоритмов....32
5.2.2 Компоненты модуля и взаимодействия между ними....33
5.2.3 Связи модуля с другими компонентами платформы....34
5.3 Программное обеспечение....35
5.3.1 Обоснование выбора языка программирования....35
5.3.2 Обоснование программной архитектуры....35
5.3.3 Разработка программного обеспечения....35
6 Разработка архитектуры и компонентов программного обеспечения Модуля логирования и контроля метрик производительности работы системы....39
6.1 Назначение и краткое описание....39
6.2 Обоснование логической структуры....40
6.3 Программное обеспечение....41
6.3.1 Обоснование выбора языка программирования....41
6.3.2 Разработка программного обеспечения....41
7 Разработка архитектуры и компонентов программного обеспечения Подсистемы хранения данных....50

Москва, 2023

7.1 Назначение и краткое описание 50
7.2 Обоснование логической структуры 50
7.2.1 Обоснование используемых методов и алгоритмов 50
7.2.2 Компоненты модуля и взаимодействия между ними 53
7.2.3 Связи модуля с другими компонентами платформы 53
8 Разработка архитектуры и компонентов программного обеспечения модуля администрирования 55
8.1 Назначение и краткое описание 55
8.2 Обоснование логической структуры 55
8.2.1 Обоснование используемых методов и алгоритмов 55
8.2.2 Компоненты модуля и взаимодействия между ними 56
8.2.3 Связи модуля с другими компонентами платформы 57
8.3 Программное обеспечение 58
8.3.1 Обоснование выбора языка программирования 58
8.3.2 Обоснование программной архитектуры 59
8.3.3 Разработка программного обеспечения 60
9 Разработка архитектуры и компонентов программного обеспечения модуля комплексной обработки событий 68
9.1 Назначение и краткое описание 68
9.2 Обоснование логической структуры 68
9.2.1 Обоснование используемых методов и алгоритмов 68
9.2.2 Компоненты модуля и взаимодействия между ними 69
9.2.3 Связи модуля с другими компонентами платформы 70
9.3 Программное обеспечение 70
9.3.1 Обоснование выбора языка программирования 70

9.3.2 Обоснование программной архитектуры 71

9.3.3 Разработка программного обеспечения 71

10 Разработка архитектуры и компонентов программного обеспечения Модуля географических информационных систем 79

10.1 Назначение и краткое описание 79

10.2 Обоснование логической структуры 80

10.2.1 Обоснование используемых методов и алгоритмов 80

10.2.2 Компоненты модуля и взаимодействия между ними 80

10.2.3 Связи модуля с другими компонентами платформы 80

10.3 Программное обеспечение 80

10.3.1 Обоснование выбора языка программирования 81

10.3.2 Обоснование программной архитектуры 81

10.3.3 Разработка программного обеспечения 82

11 Разработка Программ и методик тестирования программных модулей геоинформационной интегрирующей платформы 85

12 Проведение тестирования разработанных программных модулей геоинформационной интегрирующей платформы 86

13 Разработка опытного образца геоинформационной интегрирующей платформы 87

14 Разработка Программы и методик предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы 89

15 Проведение предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы 90

16 Доработка программной документации геоинформационной интегрирующей платформы по результатам ее пилотных испытаний 91

17 Разработка Программы и методик приемочных испытаний опытного образца геоинформационной интегрирующей платформы 92

18 Проведение приемочных испытаний опытного образца геоинформационной интегрирующей платформы 93

19 Подача заявки в Федеральную службу по интеллектуальной собственности на оформление интеллектуальной собственности 94

ЗАКЛЮЧЕНИЕ.....96

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 97

Москва, 2023

ОПРЕДЕЛЕНИЯ

В настоящем отчете применяют следующие термины с соответствующими определениями:

Архитектура модуля/подсистемы/системы – это совокупность решений об организации программного обеспечения, обеспечивающих его соответствие функциональным и нефункциональным требованиям, включающая, как решения, связанные с выбором структурных элементов и их интерфейсов, так и характером обмена данными между этими элементами.

Протокол передачи данных – это набор определённых правил или соглашений интерфейса логического уровня, который определяет обмен данными между различными программами. Эти правила задают единообразный способ передачи сообщений и обработки ошибок.

Конструкторская документация – это комплекс основных видов конструкторских документов, таких, как, чертежи деталей, спецификации сборочных единиц, комплексов и комплектов. Также к конструкторским документам относят схемы, ведомости, технические условия и др. Правила комплектации и оформления конструкторской документации определяются стандартом ЕСКД.

Комплекс – это совокупность чего-либо, объединённого вместе, имеющего общее предназначение, и отвечающего какой-либо определённой общей цели. В данном документе понятие Комплекс обозначает совокупность программного обеспечения единой системы управления (ЕСУ) и интегрированной инженерной программной платформы (ИИПП).

Методика - документ, определяющий совокупность правил, определяющих порядок действий исполнителей и их ответственность в рамках какой-либо области рассматриваемых бизнес-процессов, является основой для разработки регламентов.

Регламент – это совокупность правил, определяющих порядок действий исполнителей и их ответственность в рамках рассматриваемых бизнес-процессов.

ИТ-система – это система, предназначенная для хранения, поиска и обработки информации.

Протокол TCP, или Transmission Control Protocol – это Протокол управления передачей. Протокол TCP предоставляет транспортные услуги, отличающиеся от услуг UDP. Вместо ненадежной доставки дейтаграмм без установления соединений, он обеспечивает гарантированную доставку с установлением соединений в виде байтовых потоков.

Протокол UDP, или User Datagram Protocol – это Протокол Пользовательских Дейтаграмм), который обеспечивает обслуживание без установления соединения, таким образом UDP не гарантирует доставку или проверки последовательности для любой дейтаграммы.

Трёхуровневая архитектура (трёхзвенная архитектура, англ. three-tier) – это архитектурная модель программного комплекса, предполагающая наличие в нём трёх типов компонентов (уровней, звеньев): клиентских приложений (с которыми работают пользователи), серверов приложений (с которыми работают клиентские приложения) и серверов баз данных (с которыми работают серверы приложений).

Клиент (слой клиента в составе трехуровневой архитектуры) — это интерфейсный (обычно графический) компонент комплекса, предоставляемый конечному пользователю.

Этот уровень не должен иметь прямых связей с базой данных (по требованиям безопасности и масштабируемости), быть нагруженным основной бизнес-логикой (по требованиям масштабируемости) и хранить состояние приложения (по требованиям надёжности). На этот уровень обычно выносится только простейшая бизнес-логика: интерфейс авторизации, алгоритмы шифрования, проверка вводимых значений на допустимость и соответствие формату, несложные операции с данными (сортировка, группировка, подсчёт значений), уже загруженными на терминал.

Сервер приложений (слой сервера приложений в составе трехуровневой архитектуры) – это слой, который располагается на втором уровне, на нём сосредоточена большая часть бизнес-логики. Вне его остаются только фрагменты, экспортируемые на клиента (терминалы), а также элементы логики, погруженные в базу данных (хранимые процедуры и триггеры). Реализация данного компонента обеспечивается связующим программным обеспечением. Серверы приложений проектируются таким образом, чтобы добавление к ним дополнительных экземпляров обеспечивало горизонтальное масштабирование производительности программного комплекса и не требовало внесения изменений в программный код приложения.

Сервер баз данных (слой данных в составе трехуровневой архитектуры) – это слой, который обеспечивает хранение данных и выносится на отдельный уровень, реализуется, как правило, средствами систем управления базами данных, подключение к этому компоненту обеспечивается только с уровня сервера приложений.

Функциональное тестирование – тестирование программного обеспечения на соответствие заданным функциональным требованиям.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем отчете применяют следующие обозначения и сокращения, приведенные в таблице ниже (Таблица 1).

Таблица 1 - Обозначения и сокращения

Термин	Описание
API	От англ. Application Programming Interface — «программный интерфейс приложения») — описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
AtoN	Аббревиатура от англ. Aids to Navigation, представляющие собой виртуальные и физические средства навигационного оборудования (буи и проч.).
CCTV	Аббревиатура от англ. Closed Circuit Television – это система замкнутого телевидения, то есть такая система, где видео не транслируется на общедоступные сети, а демонстрируется только на специальном приемнике. В ней используются устройства записи, обработки, хранения и воспроизведения видеоинформации.
MMSI	Аббревиатура от англ. Maritime Mobile Service Identity - Идентификатор морской подвижной службы.
ОС	Операционная система
OGC	От англ. Open Geospatial Consortium – Открытый геопространственный консорциум.
REST API	Термин состоит из двух аббревиатур, которые расшифровываются следующим образом. API и REST (Representational State Transfer) — это способ создания API с помощью протокола HTTP. В русском языке – «передача состояния представления».
RTSP	Аббревиатура от англ. real time streaming protocol – прикладной протокол, предназначенный для использования в системах, работающих с мультимедийными данными
VSAT	Малая наземная станция спутниковой связи, используемая, в том числе, в сфере морской отрасли.
АИС	Автоматизированная идентификационная система
АСОД	Автоматическая система обмена данными
БС АИС	Базовая станция АИС
ГМССБ	Глобальная морская система связи при бедствии
ДЗЗ	Дистанционное зондирование Земли
ПО	Программное обеспечение
РЛС	Радиолокационные системы
С-АИС	Спутниковая автоматизированная идентификационная система
СУБД	Система управления базами данных.
СУДС	Система управления движением судов.

ВВЕДЕНИЕ

Настоящий научно-технический отчет «Заключительный научно-технический отчет о выполнении работ по разработке средств и сервисов информационной поддержки навигации автономных судов» разработан в соответствии с пунктом 5 технического задания (Приложение № 1 Договора (Соглашения) от «23» декабря 2021 г. №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ, далее - Договор).

В соответствии с Договором Грантополучатель должен разработать опытный образец геоинформационной системы (платформы) реального времени, выполняющей интеграцию гетерогенных сервисов морской отрасли в целях информационного обеспечения автономного судовождения данными радиолокационных систем (РЛС), Глобальной морской системы связи при бедствии (ГМССБ), автоматизированной идентификационной системы (АИС) или автоматической системы обмена данными (АСОД), оптических систем наблюдения, погодных станций, а также геопривязанных данных дистанционного зондирования Земли ("космоснимков") и иных геопривязанных данных и сервисов, обеспечивающих работу бортовых и береговых навигационных систем (далее – Опытный образец).

В ходе разработки Опытного образца были выполнены следующие работы:

согласование с пилотной площадкой реализации проекта-маяка: сроки и порядок поэтапных (макет, опытный образец, опытная партия) испытаний на ней продукции/услуг, создаваемой по результатам выполнения НИОКР (описание результатов работ приведено в разделе 1 настоящего документа);

разработка частных технических заданий на программные модули геоинформационной интегрирующей платформы (описание результатов работ приведено в разделе 2 настоящего документа);

разработка архитектуры и компонентов программного обеспечения в составе:

- модуль входных интерфейсов (описание результатов работ приведено в разделе 3 настоящего документа);
- модуль предварительной обработки и фильтрации данных (описание результатов работ приведено в разделе 4 настоящего документа);
- модуль выходных интерфейсов (описание результатов работ приведено в разделе 5 настоящего документа);
- модуль логирования и контроля метрик производительности работы системы (описание результатов работ приведено в разделе 6 настоящего документа);
- подсистема хранения данных (описание результатов работ приведено в разделе 7 настоящего документа);
- модуль администрирования (описание результатов работ приведено в разделе 8 настоящего документа);
- модуль комплексной обработки событий (описание результатов работ приведено в разделе 9 настоящего документа);
- модуль географических информационных систем (описание результатов работ приведено в разделе 10 настоящего документа).

разработка Программ и методик тестирования программных компонентов геоинформационной интегрирующей платформы (описание результатов работ приведено в разделе 11 настоящего документа);
проведение тестирования разработанных программных модулей геоинформационной интегрирующей платформы (описание результатов работ приведено в разделе 15 настоящего документа);
разработка опытного образца геоинформационной интегрирующей платформы (описание результатов работ приведено в разделе 13 настоящего документа);
разработка Программы и методик предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы (описание результатов работ приведено в разделе 14 настоящего документа);
проведение предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы (описание результатов работ приведено в разделе 15 настоящего документа);
доработка программной документации геоинформационной интегрирующей платформы по результатам ее пилотных испытаний (описание результатов работ приведено в разделе 16 настоящего документа);
разработка Программы и методик приемочных испытаний опытного образца геоинформационной интегрирующей платформы (описание результатов работ приведено в разделе 17 настоящего документа);
проведение приемочных испытаний опытного образца геоинформационной интегрирующей платформы (описание результатов работ приведено в разделе 18 настоящего документа);
подача заявки в Федеральную службу по интеллектуальной собственности на оформление интеллектуальной собственности (описание результатов работ приведено в разделе 19 настоящего документа).

СРОКИ И ПОРЯДОК ПОЭТАПНЫХ ИСПЫТАНИЙ

В ходе выполнения работ было подписано Соглашение о проведении пилотных испытаний опытного образца геоинформационной системы (платформы) на площадке на базе СУДС Региональной системы безопасности мореплавания (РСБМ) Финского залива.

Соглашение о проведении пилотных испытаний опытного образца геоинформационной интегрирующей платформы было заключено с Закрытым акционерным обществом «Институт телекоммуникаций» (ЗАО «Институт телекоммуникаций»), являющимся головным исполнителем по государственному Контракту от 12 ноября 2021 г. № 21411.1810190019.09.016 на выполнение ОКР «Разработка системы управления движением судов (СУДС) нового поколения с использованием технологии е-Навигации», шифр «Маяк-СУДС».

Предметом Соглашения является взаимодействие Сторон по организации проведения пилотных испытаний опытного образца геоинформационной системы (платформы) реального времени, выполняющей интеграцию гетерогенных сервисов морской отрасли в целях информационного обеспечения автономного судовождения данными радиолокационных систем (РЛС), глобальной морской системы связи при бедствии (ГМССБ), автоматизированной идентификационной системы (АИС) или автоматической системы обмена данными (АСОД), оптических систем наблюдения, погодных станций, а также геопривязанных данных дистанционного зондирования Земли ("космоснимков") и иных геопривязанных данных и сервисов, обеспечивающих работу бортовых и береговых навигационных систем (далее – «Геоинформационная интегрирующая платформа» или «GEOKASKAD Marine») на базе СУДС Региональной системы безопасности мореплавания (РСБМ) Финского залива в рамках проведения работ по ОКР «Разработка системы управления движением судов (СУДС) нового поколения с использованием технологии е-Навигации», шифр «Маяк-СУДС».

В целях оценки потенциала использования GEOKASKAD Marine в рамках систем управления движением судов (СУДС), в том числе, морских автономных надводных средств (МАНС) с использованием технологии е-Навигации принято решение реализовать мероприятия, определенные в Регламенте пилотных испытаний опытного образца геоинформационной системы (платформы) реального времени (Приложение № 1 к Соглашению).

В соответствии с Регламентом, предварительно запланированы следующие этапы:

Разработка опытного образца геоинформационной интегрирующей платформы в срок до 1 ноября 2022 г.

Разработка Программы и методик предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы в срок до 7 ноября 2022 г.

Проведение предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы в срок до 15 ноября 2022 г.

Доработка программной документации и геоинформационной интегрирующей платформы по результатам ее пилотных испытаний в срок до 1 декабря 2022 г.

Разработка Программы и методик приемочных испытаний опытного образца геоинформационной интегрирующей платформы в срок до 1 декабря 2022 г.

Проведение приемочных испытаний опытного образца геоинформационной интегрирующей платформы в срок до 5 декабря 2022 г.

Соглашение о проведении пилотных испытаний геоинформационной интегрирующей платформы приводятся в Приложении Т (обязательное). Соглашение о проведении пилотных испытаний.

Москва, 2023

РАЗРАБОТКА ЧАСТНЫХ ТЕХНИЧЕСКИХ ЗАДАНИЙ НА ПРОГРАММНЫЕ МОДУЛИ ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ

В ходе выполнения работ этапа 1 Договора (Соглашения) от «23» декабря 2021 г. №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ, были проведены работы по разработке частных технических заданий на программные модули, входящие в состав геоинформационной интегрирующей платформы GEOKASKAD Marine.

При формулировании требований к программным компонентам принимались во внимание потребности компаний, работающих в сфере морской отрасли Российской Федерации, в том числе, принимающих участие в проектах-маяках, проводимых Министерством промышленности и торговли Российской Федерации.

Принимая во внимание востребованность разрабатываемой платформы, по совокупности требований к программным модулям геоинформационной интегрирующей платформы, между Обществом с ограниченной ответственностью «СмартГеосистемс» (ООО «СмартГеосистемс») и Закрытым акционерным обществом «Институт телекоммуникаций» (ЗАО «Институт телекоммуникаций»), являющимся головным исполнителем по государственному Контракту от 12 ноября 2021 г. № 21411.1810190019.09.016 на выполнение ОКР «Разработка системы управления движением судов (СУДС) нового поколения с использованием технологии е-Навигации», шифр «Маяк-СУДС», было заключено соглашение о проведении пилотных испытаний опытного образца геоинформационной системы (платформы) реального времени № 05.2022/СГС/01 от «13» мая 2022 года.

Москва, 2023

РАЗРАБОТКА АРХИТЕКТУРЫ И КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЯ ВХОДНЫХ ИНТЕРФЕЙСОВ

Назначение и краткое описание

Модуль входных интерфейсов предназначен для обеспечения возможности подключения данных РЛС, АИС или АСОД, оптических систем наблюдения, погодных станций, а также геопривязанных данных дистанционного зондирования Земли ("космоснимков") и иных геопривязанных данных и сервисов, обеспечивающих работу бортовых и береговых навигационных систем к Морской геоинформационной интегрирующей платформе реального времени GEOKASKAD Marine, разрабатываемой в рамках Договора (Соглашения) №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ.

Разработанный в рамках выполнения первого этапа Модуль входных интерфейсов, обеспечивает возможность подключения следующих типов данных:

данные РЛС, поступающих в реальном масштабе времени от стороннего специального программного обеспечения Радарных процессоров, предоставляющего:
данные радарных целей;
данные интегрированных целей;
данные радарного видео.

данные сети наземных и спутниковых АИС/АСОД;

данные метеорологических станций;

данные видеонаблюдения;

электронные морские навигационные карты, географически привязанные данные ДЗЗ, данные, предоставляемые цифровыми сервисами краткосрочного и долгосрочного прогнозов метеорологической обстановки, а также иных релевантных данных, предоставляемых сторонними или смежными системами в соответствии со стандартами OGC.

Для обеспечения процесса подключения источников Модуль входных интерфейсов поддерживает следующие функциональные возможности (методы):

возможность отображения перечня подключенных источников, а также параметров их подключения;

возможность отображения перечня подключенных источников, имеющих географическую привязку, на цифровой карте;

возможность создания (подключения) нового источника данных;

возможность редактирования ранее подключенных источников данных;

возможность отключения ранее подключенного источника данных.

Данные, поступающие из подключенных источников, преобразуются ко внутреннему унифицированному представлению, либо ретранслируются пользователю/клиентской системе, без изменений в случае, если клиентская система предполагает использование исходных технологических форматов и протоколов данных.

Подключенные к платформе GEOKASKAD Marine при помощи Модуля входных интерфейсов данные, доступны Модулю предварительной обработки и фильтрации данных, Модулю выходных интерфейсов, а также могут передаваться в подсистему хранения данных для последующего воспроизведения и анализа.

Обоснование логической структуры

Обоснование используемых методов и алгоритмов

Для разработки Модуля входных интерфейсов была выбрана микросервисная архитектура. В разделе 3.3.2 приводится обоснование принятого решения.

Каждый компонент Модуля входных интерфейсов реализован как отдельный контейнер, с использованием технологии Docker.

Выбор был обусловлен следующими факторами:

независимость рабочего окружения. Контейнеры позволяют упаковать в единый образ приложение и все его зависимости: библиотеки, системные утилиты и файлы настройки. Это упрощает перенос приложения на другую инфраструктуру.

изоляция и безопасность. Контейнер — это набор процессов, изолированных от основной операционной системы. Приложения работают только внутри контейнеров, и не имеют доступа к основной операционной системе. Это повышает безопасность приложений, потому что они не смогут случайно или умышленно навредить основной системе. Если приложение в контейнере завершится с ошибкой, это никак не затронет основную ОС.

автоматизация развертывания приложений и масштабируемость. Контейнеры упрощают развертывание приложений. В классическом подходе для установки ПО может потребоваться выполнение серии действий. Контейнеры позволяют полностью автоматизировать этот процесс, так как включают в себя все нужные зависимости и порядок выполнения действий. Также контейнеры упрощают развертывание на нескольких серверах. В классическом подходе для того, чтобы развернуть одно и то же приложение на нескольких машинах, нужно будет повторять одни и те же действия. Контейнеры избавляют от этой рутинной работы и позволяют автоматизировать развертывание.

Компоненты Модуля входных интерфейсов запускаются с использованием их точек входа, как отдельные, слабо связанные компоненты микросервисной архитектуры.

Компоненты модуля и взаимодействия между ними

Модуль входных интерфейсов включает в состав программные компоненты, описанные в следующей таблице.

Таблица 2 – Компоненты Модуля входных интерфейсов

	Наименование компонента	Описание модуля
	AIS TCP Client Receiver	TCP-клиент[1] приема данных АИС в формате NMEA 0183 [**3**]. TCP подключение формируется на основе параметров IP-адреса (или домена) и порта для подключения.

[1] Клиент, в контексте Модуля входных интерфейсов, означает, что подключение к источнику данных происходит, как к TCP или UDP серверу, с указанием данных о подключении. Оборудование, выступающее в рамках информационного обмена, как TCP/UDP сервер, поставляет данные в соответствующий клиент модуля.

	AIS TCP Server Receiver	TCP-сервер[2] приема данных АИС в формате NMEA 0183.
	AIS UDP Server Receiver	UDP-сервер приема данных АИС в формате NMEA 0183. UDP подключение формируется на основе broadcast адреса, порта и IP-адреса интерфейса.
	AIS ORBCOMM Receiver	Специализированный компонент приема данных, предназначенный для приема данных С-АИС от одного из ведущих поставщиков данных С-АИС – компании ORBCOMM.
	AIS ExactEarth Receiver	Специализированный компонент приема данных, предназначенный для приема данных С-АИС от одного из ведущих поставщиков данных С-АИС – компании Spire[3].
	Asterix062 Receiver	UDP-клиент данных РЛС в формате ASTERIX Cat 062 [**4**].
	Integrated Targets Receiver	UDP-клиент приема данных интегрированных[4] целей РЛС.
	Radar Targets Receiver	UDP-клиент приема данных целей РЛС, поступающих в форматах, отличных от ASTERIX Cat 062.
	Meteo TCP Client Receiver	TCP-клиент получения данных с метеорологических станций с использованием NMEA-сентенций. В рамках разработки и тестирования, были использованы метеорологические станции, фиксирующие и передающие следующие метеорологические элементы: – Температура (град. С); – Давление (hPa); – Влажность (%); – Скорость ветра (м/с); – Направление ветра (град. от направления на север); – Осадки (мм). Данные метеорологических элементов поступают в виде текстовых сообщений, пример которого приводится в Приложении Б. В этом же приложении доступно описание элементов, входящих в состав сентенций.
	Radar Video Receiver	HTTP/HTTPS клиент приема одномоментных срезов данных («сцен») радарного видео, передаваемых в формате JPG.
	RTSP Receiver	Программный компонент, обеспечивающий прием данных видеонаблюдения. Описание протокола

[2] Сервер, в контексте Модуля входных интерфейсов, означает, что данные в него поступают от специализированного программного обеспечения GEOKASKAD Edge, устанавливаемого на стороне источника данных и, ответственного за предварительную обработку данных и их передачу в сервер модуля.

[3] Источник данных С-АИС до его приобретения компанией Spire, известный, как exactEarth.

[4] Под интегрированными, понимаются цели, подтвержденные данными АИС.

		доступно в [**5**], а также по адресу https://datatracker.ietf.org/doc/html/rfc7826
	OGC Proxy	Программный компонент, предназначенный для проксирования геопространственных данных, поступающих от OGC сервисов. В рамках работ Этапа 2 Договора, предусматривается передача данных в Модуль географических информационных систем (ГИС)

Компоненты Модуля входных интерфейсов представлены на иллюстрации в Приложении А1.

Связи модуля с другими компонентами платформы

Модуль входных интерфейсов непосредственно взаимодействует со следующими модулями платформы реального времени GEOKASKAD Marine:

модуль предварительной обработки и фильтрации данных;

модуль выходных интерфейсов;

подсистема хранения данных;

модуль логирования и контроля метрик производительности работы системы;

модуль администрирования.

Данные, поступающие в платформу реального времени посредством Модуля входных интерфейсов, сохраняются в оперативную базу данных Подсистемы хранения данных, проходят предварительную обработку и фильтрацию, а затем предоставляются пользователям и/или сторонним и смежным ИТ-системам посредством модуля выходных интерфейсов.

Управление логами и метриками, в том числе, Модуля входных интерфейсов, осуществляется Модулем логирования и контроля метрик производительности работы системы.

Модуль администрирования обеспечивает настройку параметров Модуля входных интерфейсов, с использованием визуального пользовательского интерфейса.

Общая схема взаимодействия между компонентами платформы GEOKASKAD Marine представлена на иллюстрации в Приложении А1.

Программное обеспечение

В данном подразделе приводятся основные результаты разработки и тестирования программного обеспечения, входящего в состав Модуля входных интерфейсов. Более подробная информация о разработанном программном обеспечении и его тестировании, представлены в [1] и [2].

Обоснование выбора языка программирования

При выборе языка программирования для Модуля входных интерфейсов принимались во внимание следующие требования к языку программирования.

Простота синтаксиса, низкий уровень вхождения. Данное требование обусловлено тем, что при последующей эксплуатации платформы GEOKASKAD Marine, высока

вероятность возникновения необходимости создания новых адаптеров для подключения новых типов входных данных и поддержки новых протоколов.

Поддержка многопоточности. Должна обеспечиваться поддержка многопоточности на уровне самого языка ввиду того, что Модуль входных интерфейсов должен поддерживать асинхронную работу с большим количеством высоконагруженных потоков данных;

Высокая производительность. В соответствии с количественными параметрами, предъявляемыми Техническим заданием (Приложение 1 к Договору), Модуль входных интерфейсов должен обеспечивать бесперебойную работу при частоте поступления сообщений С-АИС не менее 500 пакетов NMEA 0183 в секунду. К системе может быть подключено несколько таких источников, а в перспективе возможно расширение сферы использования платформы GEOKASKAD Marine на т. н. «судовой IoT», где частота поступления пакетов данных может быть еще выше.

Кроссплатформенность. Модуль входных интерфейсов может устанавливаться, как на собственных вычислительных мощностях, так и на мощностях заказчика, выбор операционной системы которым, зачастую обусловлен другими используемыми им программными продуктами. В связи с этим, наличие кроссплатформенности значительно повышает конкурентоспособность решения.

В перечень кандидатов были выбраны такие языки программирования, как C++, Python, Java и Go. Некоторые из альтернативных вариантов были исключены ввиду ограничений в части требований кроссплатформенности.

C++, несмотря на высокие показатели, демонстрируемые в многопоточных высоконагруженных приложениях, имеет достаточно высокий порог вхождения.

Python, несмотря на простоту и широкое распространение, имеет недостаточно высокую производительность, не обеспечивая требуемых характеристик в части обработки данных в реальном времени.

Сильные стороны Java проявляются при разработке корпоративных приложений ввиду развитой экосистеме разработки. При этом язык имеет достаточно высокий порог вхождения и при высоких нагрузках предъявляет достаточно высокие требования к аппаратному обеспечению.

Язык программирования Go соответствует всем перечисленным выше требованиям и получил на сегодняшний день достаточно высокое распространение. В связи с приведенным анализом, для программной реализации Модуля входных интерфейсов был выбран именно Go.

При этом, высокоуровневое координирование низкоуровневых потоковых процессов, обработчики которых реализованы на Go было решено реализовывать на языке программирования Java ввиду возможной будущей интеграции разрабатываемой платформы в существующие корпоративные программные экосистемы.

Таким образом, для реализации модуля входных интерфейсов была выбрана гибридная реализация с использованием языков программирования Go и Java.

Обоснование программной архитектуры

Для реализации Модуля входных интерфейсов была принята микросервисная архитектура ввиду того, что она характеризуется следующими преимуществами.

Независимость и автономность. Микросервисная архитектура позволяет создавать независимые самодостаточные, автономно работающие команды, ответственные за реализацию конкретной бизнес-логики. В случае Модуля входных интерфейсов, каждый процесс приема и декодирования входных потоков, должен быть независимым от других потоков данных.

Возможность применения различных платформ и технологий. Несмотря на то, что в рамках опытного образца платформы GEOKASKAD Marine для разработки модулей обработки был выбран Go, достаточно высока вероятность того, что в будущем возникнет необходимость использовать специализированные, в том числе, проприетарные библиотеки, разработанные на других языках программирования.

Масштабируемость. Масштабируемость микросервисной архитектуры демонстрирует высокую эффективность в обеспечении отказоустойчивости и адаптируемости системы к повышению плотности входных потоков данных (за счет горизонтального масштабирования).

Управляемость и стабильность работы. Данный пункт является следствием независимости и автономности процессов – возможные дефекты в реализации одного микросервиса, или ошибки во входном потоке данных, не окажут влияния на работу остальных, что существенно в работе платформы реального времени GEOKASKAD Marine.

Разработка программного обеспечения

Технические средства

В соответствии с требованиями технического задания, Модуль входных интерфейсов, должен запускаться на одном из четырех виртуальных, или физических серверов платформы, имеющем следующие технические характеристики:

CPU 12 ядер 2,4 ГГц и выше;

RAM 24 Гб и выше;

SSD 200 Гб и выше.

Программные средства

Для функционирования Модуля входных интерфейсов необходимо следующее программное обеспечение:

операционная система Windows или Linux (Kernel>=5.3);

средства для контейнеризации Docker (> = 19.03) и Docker-Compose (> = 1.22);

программные библиотеки, необходимые для работы со стандартными типами данных и протоколов (NMEA, и др.)

высокопроизводительное NoSQL хранилище данных «ключ-значение», предназначенное для промежуточного хранения входных пакетов данных, в случае необходимости их преобразования в форматы внутреннего представления платформы GEOKASKAD Marine. Учитывая требования высокой производительности системы, было принято решение использовать средства, работающие непосредственно в оперативной памяти компьютера. На данный момент одним из самых распространенных решений

такого рода является Redis. В целях обеспечения программной совместимости с прочими компонентами платформы, был выбран Redis (> =6.0);

очередь сообщений, используемая в качестве шины данных, преобразованных во внутренние форматы представления. В качестве программного обеспечения, реализующего очередь сообщений, в силу высокой производительности, распространенности, а также устойчивости и горизонтальной масштабируемости, было выбрано ПО Apache Kafka (> = 2.13);

для организации высоконадежной распределенной координации приложений, предлагается использование Apache ZooKeeper (> = 3.7);

данные морской отрасли, а также, предоставляющие их источники, имеют геопространственную привязку. Для производительной работы с геоданными была выбрана геопространственная база данных Tile38 с открытым исходным кодом;

для хранения файловых данных используется высокопроизводительное ПО с открытым исходным кодом Minio (>=RELEASE.2022-03-0).

Запуск программы

Запуск Модуля входных интерфейсов осуществляется, как запуск контейнеризированного приложения с использованием технологий Docker и Docker-Compose.

Все контейнеры должны запускаться со следующими обязательными параметрами:

name: некое значимое имя, содержащее тип контейнера и уникальный идентификатор (например flow_id, ip_port или просто UUID);

restart-policy: unless-stopped;

network: geoflow.

Пример команды запуска контейнера:

```
docker run --env SOURCE_HOST=94.20.63.114 --env SOURCE_PORT=64782 --env SOURCE_NAME=94.20.63.114_64782 --name ais_receiver_94.20.63.114_64782 --network geoflow --restart unless-stopped registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-receiver-latest
```

Таблица 3 – Точки входа для запуска контейнеров Модуля входных интерфейсов.

Наименование компонента	Точка входа
AIS TCP Client Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-receiver-latest
AIS TCP Server Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-receiver-server-latest
AIS UDP Server Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-receiver-udp-server-latest
AIS ORBCOMM Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-orbcomm-receiver-latest
AIS ExactEarth Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-exactearth-receiver-latest

Asterix062 Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:asterix62-receiver
Integrated Targets Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:integrated-targets-receiver
Radar Targets Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:radar-targets-receiver
Meteo TCP Client Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:meteo-receiver
Radar Video Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:radar-video-receiver
RTSP Receiver	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:rtsp-receiver
OGC Proxy	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ogc-proxy

Запуск контейнера приема данных АИС с использованием точки входа, приводится в примере ниже.

Контейнер ais-receiver:

registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-receiver-latest

Обязательные параметры в виде переменных окружения:

SOURCE_HOST: [строка] host или ip источника данных;

SOURCE_PORT: [целое] порт источника данных;

SOURCE_NAME: [строка] название источника данных (для мониторинга и для фильтрации).

Необязательные параметры в виде переменных окружения:

KAFKA_HOST: [строка] host или ip Apache Kafka;

REDIS_HOST: [строка] host или ip Redis;

CONSUL_ADDRESS: [строка] адрес Consul в формате host:port.

Значения по умолчанию:

KAFKA_HOST: kafka

REDIS_HOST: redis

CONSUL_ADDRESS: consul:8500

Аналогично выполняется запуск остальных контейнеров с использованием их точек входа.

Входные данные

В соответствии с требованиями Технического задания (Приложение 1 к Договору, в связи с работой Модуля входных интерфейсов, Система должна поддерживать следующие виды входных воздействий:

действия оператора, в рамках человеко-машинного взаимодействия, направленные на подключение источников входных данных различного типа;

входные данные, поступающие от средств и сервисов информационной поддержки автономного судовождения.

В целях реализации данного требования Модуль входных интерфейсов предоставляет пользователю следующие функциональные возможности:

Создание и настройка источника данных

РЛС;

АИС или АСОД;

оптических систем наблюдения;

погодных станций;

геопривязанных данных, предоставляемых в соответствии со стандартами OGC;

получение настроек подключенного источника данных;

удаление (отключение) подключенного ранее источника данных;

отображение подключенного источника данных в случае, если у него заданы координаты.

Таблица 4 – Используемые форматы входных данных.

Тип источника	Используемый формат данных
Данные АИС	Стандартом распространения данного типа информации, как для передачи данных С-АИС, так и для передачи данных БС АИС, принят формат NMEA 0183 [**3**]. Требования, связанные с соответствием отраслевым стандартам, определяет использование NMEA 0183 в Модуле входных интерфейсов.
Данные РЛС – радарное видео	В текущей реализации и с учетом технических характеристик используемого радарного процессора, данные радарного видео получаются в формате JPG путем вызова REST API по протоколу HTTP/HTTPS.
Данные РЛС – радарные цели	В текущей реализации принят единый формат представления целей РЛС, как подтвержденных (интегрированные), так и не подтвержденных данными АИС. Используемая структура пакетов представлена в Приложении В.
Данные РЛС – интегрированные цели	
Метеорологические измерения, поступающие с метеостанций	В качестве формата передачи данных метеонаблюдений, в силу его применения в большинстве метеорологических станций, используется формат текстовых сентенций, пример и описание которого представлены в Приложении Б.
Данные видеонаблюдения	Для работы с данными видеонаблюдения, используется потоковый протокол реального времени RTSP (англ. real time streaming protocol, сокр. RTSP) — прикладной протокол, предназначенный для использования в системах, работающих с мультимедийными данными, и позволяющий удалённо управлять потоком данных с сервера, предоставляя возможность выполнения команд, таких как запуск (старт), приостановку (пауза) и остановку (стоп) вещания (проигрывания) мультимедийного содержимого, а также доступа по времени к файлам, расположенным на сервере [**5**].
Данные ДЗЗ и иные географически привязанные	В качестве протокола представления географически привязанных данных и сервисов, включая данные ДЗЗ,

данные, и сервисы морской отрасли	предоставляемых сторонними и смежными системами, используются стандарты OGC WMS [**6**]. Использование формата обусловлено повсеместным распространением и поддержкой данного формата практически всеми современными геоинформационными системами и сервисами.

Подключаемые к Модулю входных интерфейсов данные не проходят дополнительной обработки и поступают на вход его компонентов посредством TCP, UDP или HTTP/HTTPS подключения.

Выходные данные

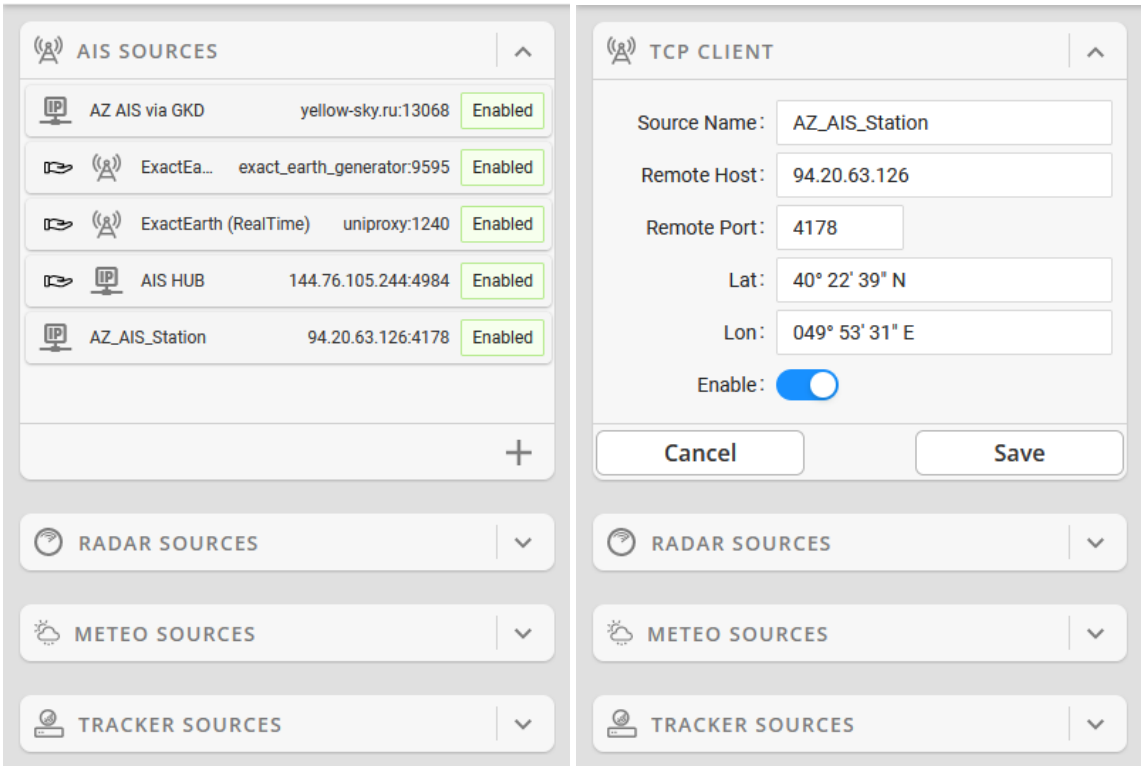
Модуль входных интерфейсов поддерживает два типа представления выходных данных:

без преобразования. Используется в случае необходимости ретрансляции данных без изменения в исходных технологических форматах, путем их прямой передачи в Модуль выходных интерфейсов;

с преобразованием к унифицированному представлению. Данный формат используется при необходимости передачи входных пакетов иным компонентам платформы реального времени в виде объектного представления через очередь сообщений.

В целях описания схем представления объектов используются механизм сериализации-десериализации данных, основанный на схемах Apache Avro (AVRO – схемы). Взаимодействие пользователя с модулем входных интерфейсов осуществляется посредством веб-консоли модуля администрирования. На рис.1 (а) и (б) представлен список подключенных посредством Модуля входных интерфейсов источников данных АИС (а) и интерфейс подключения источника посредством TCP (б).

Рисунок 1. Подключение данных АИС посредством модуля входных интерфейсов



(а) Список АИС-подключений (б) Добавление нового источника

РАЗРАБОТКА АРХИТЕКТУРЫ И КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЯ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ И ФИЛЬТРАЦИИ ДАННЫХ

Назначение и краткое описание

Модуль предварительной обработки и фильтрации данных предназначен для выполнения программируемой обработки в реальном, или приближенном к реальному времени, данных, передаваемых при помощи Модуля входных интерфейсов, подключенными средствами и сервисами, включая их фильтрацию, преобразование моделей данных там, где такая обработка возможна и требуется, исходя из особенностей поступающей информации.

Перечень типов данных, поступающих на вход Модулю предварительной обработки и фильтрации данных, совпадает с перечнем типов данных, подключаемых Модулем входных интерфейсов, а именно: данные РЛС, АИС или АСОД, оптических систем наблюдения, погодных станций, а также геопривязанных данных дистанционного зондирования Земли ("космоснимков") и иных геопривязанных данных и сервисов, обеспечивающих работу бортовых и береговых навигационных систем к Морской геоинформационной интегрирующей платформе реального времени GEOKASKAD Marine, разрабатываемой в рамках Договора (Соглашения) №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ.

При этом, необходимость проведения предварительной обработки, фильтрации и преобразование моделей данных относится только к потоковым данным и не включает в себя данные видеонаблюдения и электронные морские навигационные карты, географически привязанные данные ДЗЗ, данные, предоставляемые цифровыми сервисами краткосрочного и долгосрочного прогнозов метеорологической обстановки, а также иные релевантные данные, предоставляемые сторонними или смежными системами в соответствии со стандартами OGC.

Для обеспечения процесса предварительной обработки и фильтрации Модуль поддерживает следующие функциональные возможности (методы):

возможность отображения перечня каналов данных, каждый из которых включает в себя описание используемых механизмов предварительной обработки и фильтрации данных;

возможность создания нового потока, характеризующегося набором настроек предварительной обработки и фильтрации данных, в том числе
фильтрации по атрибутивным данным;
фильтрации по геопространственной привязке;

возможность редактирования ранее созданного потока;

возможность запуска существующего потока данных, что приводит к обеспечению приема данных от Модуля входных интерфейсов, их обработки в соответствии с настройками предварительной обработки и фильтрации, передачи отфильтрованных и преобразованных данных на вход Модулю выходных интерфейсов;

возможность остановки ранее запущенных потоков;

возможность удаления существующих потоков данных, если в них нет необходимости.

Данные, обрабатываемые Модулем предварительной обработки и фильтрации данных, передаются Модулю выходных интерфейсов.

Обоснование логической структуры

Обоснование используемых методов и алгоритмов

Для разработки Модуля предварительной обработки и фильтрации данных была выбрана микросервисная архитектура. Обоснование микросервисной архитектуры аналогично обоснованию архитектуры Модуля входных интерфейсов и приводится в разделе 3.3.2 соответствующего раздела настоящего отчета.

В соответствии с принятыми архитектурными решениями, как в случае Модуля входных интерфейсов, каждый компонент Модуля предварительной обработки и фильтрации данных реализован как отдельный контейнер, с использованием технологии Docker.

Выбор, как и в случае Модуля входных интерфейсов, был обусловлен следующими факторами:

независимость рабочего окружения контейнеризированного приложения Модуля предварительной обработки и фильтрации данных;

изоляция и безопасность, ввиду отделенности контейнеризированного приложения Модуля предварительной обработки и фильтрации данных от остальных контейнеров и операционной системы (включая общесистемные приложения и сервисы);

автоматизация развертывания приложений, входящих в состав Модуля предварительной обработки и фильтрации данных, а также масштабируемость.

Компоненты Модуля предварительной обработки и фильтрации данных запускаются с использованием их точек входа, как отдельные, слабо связанные компоненты микросервисной архитектуры.

Компоненты модуля и взаимодействия между ними

Модуль предварительной обработки и фильтрации данных включает в состав программные компоненты, описанные в следующей таблице.

Таблица 5 – Компоненты Модуля предварительной обработки и фильтрации данных

	Наименование компонента	Описание модуля
	AIS Filter	Данный компонент предназначен для выполнения фильтрации данных с использованием атрибутивных полей, поступающих с данными NMEA 0183 [**3**], а также, с использованием фильтрации на основе географической привязки с использованием поступающих данных о широте и долготе цели. Следует отметить, что данные АИС включают в себя 27 типов сообщений, среди которых для тестирования Модуля предварительной обработки и фильтрации данных, были использованы:

		– Данные отчета о местоположении (сообщения типов 1–3) судов AIS Class A; – Данные отчета о рейсе судна (сообщение типа 5)[5]; – Данные AtoN (сообщение типа 21)[6]. Для работы Модуля предварительной обработки и фильтрации данных могут использоваться данные, принимаемые следующими компонентами Модуля входных интерфейсов (см. 3.2.2): AIS TCP Client Receiver, AIS TCP Server Receiver, AIS UDP Server Receiver, AIS ORBCOMM Receiver, AIS ExactEarth Receiver.
	Radar Targets Filter	Данный компонент предназначен для выполнения фильтрации данных РЛС, которые могут поступать, как в специализированных форматах, поддерживаемых конкретным ПО радарного процессора, экстрактора и интегратора целей, так и с использованием данных ASTERIX Cat 062 [**4**]. Как и в случае с данными АИС, цели РЛС могут фильтроваться по атрибутивным параметрам, а также на основании местоположения. Для работы Модуля предварительной обработки и фильтрации данных могут использоваться данные, принимаемые следующими компонентами Модуля входных интерфейсов (см. 3.2.2): Asterix062 Receiver, Radar Targets Receiver.
	Integrated Targets Filter	Данный компонент предназначен для предварительной обработки и фильтрации данных, принимаемых компонентом Integrated Targets Receiver Модуля входных интерфейсов. Как и в случае с данными АИС, цели РЛС могут фильтроваться по атрибутивным параметрам, а также на основании местоположения, причем, в случае интегрированных целей, имеющих подтверждение пакетами данных АИС, становится доступным расширенный набор полей, входящих в состав сообщения 5 АИС.
	Scan View Filter	Специализированный фильтр, предназначенный для работы с данными, предоставляемыми ПО Scan View компании DspNor, Норвегия. Тестирование осуществлялось в рамках апробации Модуля в составе пилотного проекта по ретрансляции данных с судов, несущих на борту РЛС

[5] Для выполнения комплексной фильтрации, как с использованием информации о местоположении, так и с использованием информации о рейсе, Модулем предварительной обработки и фильтрации, в реальном времени осуществляется связывание пакетов данных сообщений 1–3 и сообщений 5 с использованием поля MMSI.

[6] Также были проведены тесты на данных AtoN, поступающих в составе сообщения 21 АИС и содержащих в себе, как информацию о местоположении, так и свойства данного средства навигационного оборудования.

		и передающих результаты мониторинга акватории по каналам VSAT.
	Meteo Raw Filter	Компонент, предназначенный для предварительной обработки исходных метеорологических данных, принимаемых Модулем входных интерфейсов при помощи компонента Meteo TCP Client Receiver.
	Meteo Sentence Filter	Компонент, предназначенный для предварительной обработки и фильтрации метеорологических данных после их декодирования на основе информации о типе метеорологических элементов, принимаемых Модулем входных интерфейсов при помощи компонента Meteo TCP Client Receiver.

Компоненты Модуля предварительной обработки и фильтрации данных представлены на иллюстрации в Приложении А1.

Связи модуля с другими компонентами платформы

Модуль предварительной обработки и фильтрации данных непосредственно взаимодействует со следующими модулями платформы реального времени GEOKASKAD Marine:

модуль входных интерфейсов;

модуль выходных интерфейсов;

модуль логирования и контроля метрик производительности работы системы;

модуль администрирования.

Данные, поступающие в платформу реального времени посредством Модуля входных интерфейсов, сохраняются в оперативную базу данных Подсистемы хранения данных, проходят предварительную обработку и фильтрацию, а затем предоставляются пользователям и/или сторонним и смежным ИТ-системам посредством модуля выходных интерфейсов.

Управление логами и метриками, в том числе, Модуля входных интерфейсов, осуществляется Модулем логирования и контроля метрик производительности работы системы.

Модуль администрирования обеспечивает настройку параметров Модуля входных интерфейсов, с использованием визуального пользовательского интерфейса.

Общая схема взаимодействия между компонентами платформы GEOKASKAD Marine представлена на иллюстрации в Приложении А1.

Программное обеспечение

В данном подразделе приводятся основные результаты разработки и тестирования программного обеспечения, входящего в состав Модуля предварительной обработки и фильтрации данных. Более подробная информация о разработанном программном обеспечении и его тестировании, представлены в [7] и [8].

Обоснование выбора языка программирования

Выбор языка программирования для реализации ПО Модуля предварительной обработки и фильтрации данных, аналогичен анализу, проведенному для Модуля входных интерфейсов, ход и результаты которого приводятся в разделе 3.3.1 , и основывались на следующих факторах:

Простота синтаксиса, низкий уровень вхождения;

Поддержка многопоточности;

Высокая производительность;

Кроссплатформенность.

Анализ производился с учетом возможностей и характеристик следующих языков программирования: C++, Python, Java и Go.

Для реализации Модуля предварительной обработки и фильтрации данных, была выбрана гибридная реализация с использованием языков программирования Go и Java.

Обоснование программной архитектуры

Как и в случае Модуля входных интерфейсов (см. 3.3.2), для реализации Модуля предварительной обработки и фильтрации данных была принята микросервисная архитектура ввиду того, что она характеризуется следующими преимуществами:

Независимость и автономность;

Возможность применения различных платформ и технологий, включая открытые и проприетарные библиотеки, разработанные на других языках программирования;

Масштабируемость, позволяющая обеспечить высокую эффективность в части отказоустойчивости и адаптируемости системы к повышению плотности входных потоков данных (за счет горизонтального масштабирования).

Управляемость и стабильность работы в следствии от возможных ошибок, возникающих в других модулях платформы, а также в общесистемных компонентах.

Разработка программного обеспечения

Технические средства

В соответствии с требованиями технического задания, Модуль предварительной обработки и фильтрации данных, должен запускаться на одном из четырех виртуальных, или физических серверов платформы, имеющем следующие технические характеристики:

CPU 12 ядер 2,4 ГГц и выше;

RAM 24 Гб и выше;

SSD 200 Гб и выше.

Программные средства

Для функционирования Модуля предварительной обработки и фильтрации данных необходимо следующее программное обеспечение:

операционная система Windows или Linux (Kernel>=5.3);

средства для контейнеризации Docker (> = 19.03) и Docker-Compose (> = 1.22);

программные библиотеки, необходимые для работы со стандартными типами данных и протоколов (NMEA, и др.)

высокопроизводительное NoSQL хранилище данных «ключ-значение», предназначенное для промежуточного хранения входных пакетов данных, в случае необходимости их преобразования в форматы внутреннего представления платформы GEOKASKAD Marine. Учитывая требования высокой производительности системы, было принято решение использовать средства, работающие непосредственно в оперативной памяти компьютера. На данный момент одним из самых распространенных решений такого рода является Redis. В целях обеспечения программной совместимости с прочими компонентами платформы, был выбран Redis (> =6.0);

очередь сообщений, используемая в качестве шины данных, преобразованных во внутренние форматы представления. В качестве программного обеспечения, реализующего очередь сообщений, в силу высокой производительности, распространенности, а также устойчивости и горизонтальной масштабируемости, было выбрано ПО Apache Kafka (> = 2.13);

для организации высоконадежной распределенной координации приложений, предлагается использование Apache ZooKeeper (> = 3.7);

данные морской отрасли, а также, предоставляющие их источники, имеют геопространственную привязку. Для производительной работы с геоданными была выбрана геопространственная база данных Tile38 с открытым исходным кодом.

Запуск программы

Запуск Модуля предварительной обработки и фильтрации данных, как запуск контейнеризированного приложения с использованием технологий Docker и Docker-Compose.

Все контейнеры должны запускаться со следующими обязательными параметрами:

name: некое значимое имя, содержащее тип контейнера и уникальный идентификатор (например flow_id, ip_port или просто UUID);

restart-policy: unless-stopped;

network: geoflow.

Пример команды запуска контейнера:

```
docker run --env SOURCE_HOST=94.20.63.114 --env SOURCE_PORT=64782 --env
SOURCE_NAME=94.20.63.114_64782 --name ais_receiver_94.20.63.114_64782 --network geoflow -
-restart unless-stopped
registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-filter-latest
```

Таблица 6 – Точки входа для запуска контейнеров Модуля предварительной обработки и фильтрации данных.

Наименование компонента	Точка входа
AIS Filter	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/ geoflowkernel:ais-filter-latest
Radar Targets Filter	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/ geoflowkernel:asterix62-filter-latest и

	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:radar-targets-filter-latest
Integrated Targets Filter	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:integrated-targets-filter-latest
Scan View Filter	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:scanview-filter-latest
Meteo Raw Filter	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:meteo-filter-latest
Meteo Sentence Filter	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:meteo-sentence-filter-latest

Запуск контейнера фильтрации данных АИС с использованием точки входа, приводится в примере ниже.

Контейнер ais-filter:

registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-filter-latest

Обязательные параметры в виде переменных окружения:
FLOW_ID: [строка] идентификатор потока (для связи фильтров и выходов);
FILTER_EXP: [строка] выражение для фильтрации пакетов;
FILTER_GEOM_TYPE: [строка] тип географической фильтрации. возможные значения: "", "WITHIN", "OUTSIDE";

FILTER_GEOM: [строка] GeoJSON геометрия полигона.

Необязательные параметры в виде переменных окружения:

KAFKA_HOST: [строка] host или ip Apache Kafka;

REDIS_HOST: [строка] host или ip Redis;

CONSUL_ADDRESS: [строка] адрес Consul в формате host:port.

Значения по умолчанию:

REMOVE_DUPLICATES: no

KAFKA_HOST: kafka

REDIS_HOST: redis

CONSUL_ADDRESS: consul:8500

Аналогично выполняется запуск остальных контейнеров с использованием их точек входа.

Входные данные

В соответствии с требованиями Технического задания (Приложение 1 к Договору, в связи с работой Модуля предварительной обработки и фильтрации данных, Система должна поддерживать следующие виды входных воздействий:

Действия оператора, в рамках человеко-машинного взаимодействия, направленные на программирование обработки и преобразования данных, поступающих из источников различного типа.

В целях реализации данного требования Модуль предварительной обработки и фильтрации данных предоставляет пользователю следующие функциональные возможности:

получение списка каналов обработки и предоставления данных;

создание и настройка каналов в части предварительной обработки и фильтрации данных, включая

атрибутивные критерии фильтрации;

геопространственный критерии фильтрации;

получение настроек ранее созданного канала в части правил обработки и фильтрации данных;

удаление созданного ранее канала обработки данных;

отображение правил обработки фильтрации в графическом виде (диаграмма обработки данных).

Выходные данные

Модуль предварительной обработки и фильтрации данных поддерживает два типа представления выходных данных:

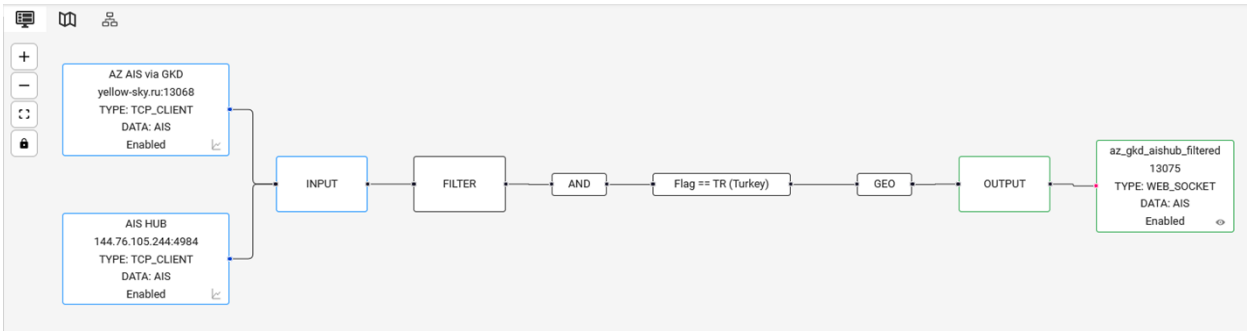
без преобразования. Используется в случае необходимости фильтрации[7] с сохранением пакетов поступающей информации без изменения в исходных технологических форматах, путем их последующей передачи в Модуль выходных интерфейсов;

с преобразованием к унифицированному представлению. Данный формат используется при необходимости передачи входных пакетов иным компонентам платформы реального времени в виде объектного представления через очередь сообщений.

В целях описания схем представления объектов используются механизм сериализации-десериализации данных, основанный на схемах Apache Avro (AVRO – схемы).

На рис.2 представлен интерфейс визуального программирования модуля предварительной обработки и фильтрации данных через веб-консоль модуля администрирования платформы.

Рисунок 2. Интерфейс визуального программирования модуля предварительной обработки и фильтрации данных.



[7] Примером может служить фильтрация данных АИС по геопространственному критерию (например, заданной акватории), с предоставлением данных в исходном технологическом формате NMEA 0183.

РАЗРАБОТКА АРХИТЕКТУРЫ И КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЯ ВЫХОДНЫХ ИНТЕРФЕЙСОВ

Назначение и краткое описание

Модуль выходных интерфейсов предназначен для предоставления данных принимаемых, унифицируемых, обрабатываемых и фильтруемых при помощи компонентов платформы GEOKASKAD Marine конечным пользователям, а также сторонним и смежным ИТ-системам с использованием распространенных протоколов и форматов данных.

Перечень типов данных, предоставляемых Модулем выходных интерфейсов, включает в себя данные РЛС, АИС или АСОД, оптических систем наблюдения, погодных станций, а также геопривязанных данных дистанционного зондирования Земли ("космоснимков") и иных геопривязанных данных и сервисов, обеспечивающих работу бортовых и береговых навигационных систем к Морской геоинформационной интегрирующей платформе реального времени GEOKASKAD Marine, разрабатываемой в рамках Договора (Соглашения) №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ.

Для обеспечения процесса предоставления данных конечным пользователям, а также сторонним и смежным ИТ-системам Модуль поддерживает следующие функциональные возможности (методы):

возможность отображения перечня выходных интерфейсов канала обработки информации;

возможность создания нового выходного интерфейса, включая

TCP-клиент

TCP-сервер

UDP-клиент

Web-сокет

возможность редактирования перечня выходных интерфейсов.

Данные, выдаваемые Модулем выходных интерфейсов, являются тем информационным потоком, который используемся платформой, как для информационного обеспечения автономного судовождения, так и для последующей обработки в рамках процесса комплексной обработки событий реального времени.

Обоснование логической структуры

Обоснование используемых методов и алгоритмов

Для разработки Модуля выходных интерфейсов была выбрана микросервисная архитектура. Обоснование микросервисной архитектуры аналогично обоснованию архитектуры Модуля входных интерфейсов и приводится в разделе 3.3.2 соответствующего раздела настоящего отчета.

В соответствии с принятыми архитектурными решениями, как в случае Модуля входных интерфейсов, каждый компонент Модуля выходных интерфейсов реализован как отдельный контейнер, с использованием технологии Docker.

Выбор, как и в случае Модуля входных интерфейсов, был обусловлен следующими факторами:

независимость рабочего окружения контейнеризированного приложения Модуля выходных интерфейсов;

изоляция и безопасность, ввиду отделенности контейнеризированного приложения Модуля выходных интерфейсов от остальных контейнеров и операционной системы (включая общесистемные приложения и сервисы);

автоматизация развертывания приложений, входящих в состав Модуля выходных интерфейсов, а также масштабируемость.

Компоненты Модуля выходных интерфейсов запускаются с использованием их точек входа, как отдельные, слабо связанные компоненты микросервисной архитектуры.

Компоненты модуля и взаимодействия между ними

Модуль выходных интерфейсов включает в состав программные компоненты, описанные в следующей таблице.

Таблица 7 – Компоненты Модуля выходных интерфейсов

	Наименование компонента	Описание модуля
	AIS TCP Client Sender	TCP-клиент[8] предоставления данных АИС в формате NMEA 0183 [**3**]. TCP подключение формируется на основе параметров IP-адреса (или домена) и порта для подключения.
	AIS TCP Server Sender	TCP-сервер[9] предоставления данных АИС в формате NMEA 0183.
	AIS UDP Client Sender	UDP-сервер предоставления данных АИС в формате NMEA 0183. UDP подключение формируется на основе broadcast адреса, порта и IP-адреса интерфейса.
	AIS WebSocket Service	Клиент, предназначенный для предоставления высоконагруженных потоковых данных с использованием технологии web-сокетов.
	Integrated Target Kafka Publisher	Клиент, предназначенный для публикации пакетов интегрированных целей в унифицированном представлении, задаваемом AVRO-схемами, в очередь сообщений Apache Kafka.
	Scan View Kafka Publisher	Клиент, предназначенный для публикации пакетов целей (радарных, АИС и интегрированных), предоставляемых ПО Scan View компании DspNor,

[8] Клиент, в контексте Модуля входных интерфейсов, означает, что подключение к источнику данных происходит, как к TCP или UDP серверу, с указанием данных о подключении. Оборудование, выступающее в рамках информационного обмена, как TCP/UDP сервер, поставляет данные в соответствующий клиент модуля.

[9] Сервер, в контексте Модуля входных интерфейсов, означает, что данные в него поступают от специализированного программного обеспечения GEOKASKAD Edge, устанавливаемого на стороне источника данных и, ответственного за предварительную обработку данных и их передачу в сервер модуля.

		Норвегия, в унифицированном представлении, задаваемом AVRO-схемами, в очередь сообщений Apache Kafka.
	Meteo Raw TCP Server Sender	TCP-сервер публикации исходных данных с метеорологических станций с использованием NMEA-сентенций.
	Meteo Sentence TCP Server Sender	TCP-сервер, предназначенный для публикации сентенций метеорологических элементов в составе: – Температура (град. C); – Давление (hPa); – Влажность (%); – Скорость ветра (м/с); – Направление ветра (град. от направления на север); – Осадки (мм).
	Scan View Radar Video Service	HTTP/HTTPS клиент публикации последовательности одномоментных срезов данных («сцен» на момент запроса) радарного видео, передаваемых в формате JPG.
	WebRTC Video Service	Программный компонент, обеспечивающий предоставление данных видеонаблюдения, поступающих в платформу при помощи Модуля входных интерфейсов. Описание протокола доступно в [**5**], а также по адресу https://datatracker.ietf.org/doc/html/rfc7826
	PelcoD Remote Control Service	Дополнительный компонент, предназначенный для организации управления камерами видеонаблюдения при помощи протокола PelcoD и PelcoD Extended.
	ONVIF Remote Control Service	Дополнительный компонент, предназначенный для организации управления камерами видеонаблюдения при помощи протокола ONVIF.

Компоненты Модуля выходных интерфейсов представлены на иллюстрации в Приложении А1.

Связи модуля с другими компонентами платформы

Модуль выходных интерфейсов непосредственно взаимодействует со следующими модулями платформы реального времени GEOKASKAD Marine:

модуль входных интерфейсов;

модуль предварительной обработки и фильтрации данных;

модуль логирования и контроля метрик производительности работы системы;

модуль администрирования.

Данные, поступающие в платформу реального времени посредством Модуля входных интерфейсов, сохраняются в оперативную базу данных Подсистемы хранения данных, проходят предварительную обработку и фильтрацию, а затем предоставляются пользователям и/или сторонним и смежным ИТ-системам посредством модуля выходных интерфейсов.

Управление логами и метриками, в том числе, Модуля входных интерфейсов, осуществляется Модулем логирования и контроля метрик производительности работы системы.

Модуль администрирования обеспечивает настройку параметров Модуля входных интерфейсов, с использованием визуального пользовательского интерфейса.

Общая схема взаимодействия между компонентами платформы GEOKASKAD Marine представлена на иллюстрации в Приложении А1.

Программное обеспечение

В данном подразделе приводятся основные результаты разработки и тестирования программного обеспечения, входящего в состав Модуля выходных интерфейсов. Более подробная информация о разработанном программном обеспечении и его тестировании, представлены в [9] и [10].

Обоснование выбора языка программирования

Выбор языка программирования для реализации ПО Модуля выходных интерфейсов, основывается на аналогичном анализе, проведенном для Модуля входных интерфейсов ход и результаты которого приводятся в разделе 3.3.1 и основываются на следующих факторах:

Простота синтаксиса, низкий уровень вхождения;

Поддержка многопоточности;

Высокая производительность;

Кроссплатформенность.

Анализ производился с учетом возможностей и характеристик следующих языков программирования: C++, Python, Java и Go.

Для реализации Модуля выходных интерфейсов, была выбрана гибридная реализация с использованием языков программирования Go и Java.

Обоснование программной архитектуры

Как и в случае Модуля входных интерфейсов (см. 3.3.2), для реализации Модуля выходных интерфейсов была принята микросервисная архитектура ввиду того, что она характеризуется следующими преимуществами:

Независимость и автономность;

Возможность применения различных платформ и технологий, включая открытые и проприетарные библиотеки, разработанные на других языках программирования;

Масштабируемость, позволяющая обеспечить высокую эффективность в части отказоустойчивости и адаптируемости системы к повышению плотности входных потоков данных (за счет горизонтального масштабирования).

Управляемость и стабильность работы в следствии от возможных ошибок, возникающих в других модулях платформы, а также в общесистемных компонентах.

Разработка программного обеспечения

Технические средства

В соответствии с требованиями технического задания, Модуль выходных интерфейсов, должен запускаться на одном из четырех виртуальных, или физических серверов платформы, имеющем следующие технические характеристики:

CPU 12 ядер 2,4 ГГц и выше;

RAM 24 Гб и выше;

SSD 200 Гб и выше.

Программные средства

Для функционирования Модуля выходных интерфейсов необходимо следующее программное обеспечение:

операционная система Windows или Linux (Kernel>=5.3);

средства для контейнеризации Docker (> = 19.03) и Docker-Compose (> = 1.22);

очередь сообщений, используемая в качестве шины данных, преобразованных во внутренние форматы представления. В качестве программного обеспечения, реализующего очередь сообщений, в силу высокой производительности, распространенности, а также устойчивости и горизонтальной масштабируемости, было выбрано ПО Apache Kafka (> = 2.13);

для организации высоконадежной распределенной координации приложений, предлагается использование Apache ZooKeeper (> = 3.7).

Запуск программы

Запуск Модуля выходных интерфейсов, выполняется, как запуск контейнеризированного приложения с использованием технологий Docker и Docker-Compose.

Все контейнеры должны запускаться со следующими обязательными параметрами:

name: некое значимое имя, содержащее тип контейнера и уникальный идентификатор (например flow_id, ip_port или просто UUID);

restart-policy: unless-stopped;

network: geoflow.

Пример команды запуска контейнера:

```
docker run -p 10000:9000 --env OUT_PORT=9000 --env FLOW_ID=ATONS --name
ais_pub_tcp_server_ATONS --network geoflow
registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-pub-tcp-server-
latest
```

Таблица 8 – Точки входа для запуска контейнеров Модуля выходных интерфейсов.

Наименование компонента	Точка входа
AIS TCP Client Sender	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/ geoflowkernel:ais-pub-tcp-client

AIS TCP Server Sender	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-pub-tcp-server
AIS UDP Client Sender	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-pub-udp-client
AIS WebSocket Service	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-pub-ws-server
Integrated Target Kafka Publisher	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:integrated-pub-kafka
Scan View Kafka Publisher	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:scanview-pub-kafka
Meteo Raw TCP Server Sender	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:meteo-simple-pub-tcp-server
Meteo Sentence TCP Server Sender	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:meteo-sentence-pub-tcp-server
Scan View Radar Video Service	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:radar-video-pub-http
WebRTC Video Service	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:video-pub-webrtc
PelcoD Remote Control Service	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:video-rcs-pelcod
ONVIF Remote Control Service	registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:video-rcs-onvif

Запуск контейнера публикации данных АИС с использованием TCP-сервера с использованием точки входа, приводится в примере ниже.

Контейнер ais-pub-tcp-server:

registry.gitlab.com/smartgeosystem/smartgeostack/smartgeoflow/geoflowkernel:ais-pub-tcp-server-latest

Обязательные параметры в виде переменных окружения:

FLOW_ID: [строка] идентификатор потока (для связи фильтров и выходов).

Необязательные параметры в виде переменных окружения:

OUT_PORT: [целое] TCP порт для подключений

KAFKA_HOST: [строка] host или ip Apache Kafka;

REDIS_HOST: [строка] host или ip Redis;

CONSUL_ADDRESS: [строка] адрес Consul в формате host:port.

Значения по умолчанию:

OUT_PORT: 9000

KAFKA_HOST: kafka

REDIS_HOST: redis

CONSUL_ADDRESS: consul:8500

Аналогично выполняется запуск остальных контейнеров с использованием их точек входа.

Входные данные

В соответствии с требованиями Технического задания (Приложение 1 к Договору, в связи с работой Модуля выходных интерфейсов, Система должна поддерживать следующие виды входных воздействий:

Действия оператора, в рамках человеко-машинного взаимодействия, направленные на подключение пользователей/клиентских систем-потребителей информации.

В целях реализации данного требования Модуль выходных интерфейсов предоставляет пользователю следующие функциональные возможности:

получение списка выходных интерфейсов для данного информационного потока;

создание и настройка выходного интерфейса для данного информационного потока;

получение настроек ранее созданного выходного интерфейса для данного информационного потока;

отключение, или полное удаление ранее созданного выходного интерфейса для данного информационного потока.

Выходные данные

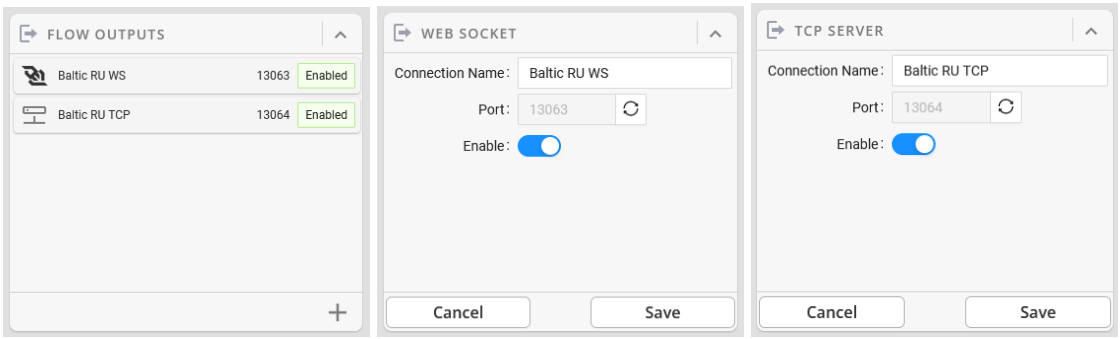
Модуль выходных интерфейсов поддерживает три типа представления выходных данных:

с использованием технологических стандартов и протоколов, принятых в сфере морской отрасли;

с преобразованием к унифицированному представлению. Данный формат используется при необходимости передачи входных пакетов иным компонентам платформы реального времени в виде объектного представления через очередь сообщений. В целях описания схем представления объектов используются механизм сериализации-десериализации данных, основанный на схемах Apache Avro (AVRO – схемы);

с использованием общих форматов представления данных, широко распространенных в сфере информационных технологий. Сюда можно отнести, в том числе, данные видеонаблюдения, поступающие с камер CCTV.

Рисунок 3. Параметры выходных интерфейсов.



(а) список интерфейсов (б) веб-сокет (в) TCP-подключение

На рис. 3 представлен визуальный интерфейс управления модулем выходных интерфейсов, предоставляемый веб-консолью модуля администрирования. На рисунке

представлены (а) список выходных интерфейсов различного типа, автоматически сформированных при создании канала (потока данных); (б) параметры подключения типа веб-сокет; (в) параметры TCP-подключения.

РАЗРАБОТКА АРХИТЕКТУРЫ И КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЯ ЛОГИРОВАНИЯ И КОНТРОЛЯ МЕТРИК ПРОИЗВОДИТЕЛЬНОСТИ РАБОТЫ СИСТЕМЫ

Назначение и краткое описание

Разработка архитектуры и компонентов программного обеспечения Модуля логирования и контроля метрик производительности работы системы выполняется в составе работ по Разработке опытного образца геоинформационной системы (платформы) реального времени, выполняющей интеграцию гетерогенных сервисов морской отрасли в целях информационного обеспечения автономного судовождения данными радиолокационных систем (РЛС), Глобальной морской системы связи при бедствии (ГМССБ), автоматизированной идентификационной системы (АИС) или автоматической системы обмена данными (АСОД), оптических систем наблюдения, погодных станций, а также геопривязанных данных дистанционного зондирования Земли ("космоснимков") и иных геопривязанных данных и сервисов, обеспечивающих работу бортовых и береговых навигационных систем.

Модуль предназначен для обеспечения логирования и сбора метрик производительности работы системы, включая автоматизированную подготовку экранов отображения информации в целях визуализации информации, характеризующей работу компонентов опытного образца геоинформационной системы (платформы) реального времени GEOKASKAD Marine, как в части сообщений об ошибках и системных оповещениях (логи), так и в части показателей текущей производительности работы модулей, входящих в состав платформы.

Разработанный Модуль обладает следующими функциональными возможностями:
Модуль обеспечивает создание, хранение и редактирование шаблонов для отображения модулей системы, включая:

- вид, форму и размеры схематического изображения модулей системы,
- пиктограммы модулей системы,
- значения метрик и статусов,
- цветовую окраску изображения для различных статусов.

Модуль обеспечивает создание, хранение и редактирование правил формирования экранов отображения информации, включая:

- расположение схематических изображений модулей системы,
- количество схематических изображений для каждого модуля системы,
- связи между изображениями модулей системы,
- элементы управления экраном отображения информации.

Модуль обеспечивает автоматизированное формирование программных кодов экранов отображения информации на основе правил формирования и шаблонов модулей.
Модуль обеспечивает возможность визуализации сформированных программных кодов экранов отображения информации в модуле логирования и контроля метрик без дополнительных доработок.
Модуль обеспечивает выдачу по протоколу http сформированных программных кодов экранов отображения информации для последующей визуализации в модуле логирования и контроля метрик.

Москва, 2023

Таким образом, Модуль решает следующие задачи:
Хранение и учет программных кодов шаблонов экранов отображения информации;
Контроль изменения количества и состава работающих в Системе компонентов;
Подготовка программного кода экранов отображения информации о состоянии Системы и ее компонентов;
Загрузка подготовленных экранов отображения в модуль контроля метрик.

Обоснование логической структуры

Программа состоит из основной подпрограммы, хранилища шаблонов, трех специализированных подпрограмм и служебной подпрограммы управления. Структура Программы без основной и служебной подпрограмм представлена на рисунке, представленном в Приложении Л.

После запуска Программа (основная подпрограмма) читает файл настроек (см. Входные данные), после чего запускает подпрограмму контроля состава компонентов Системы и служебную подпрограмму управления.

Подпрограмма контроля состава компонентов Системы при первом запуске по соответствующему программному интерфейсу (API Prometheus) получает данные о составе и количестве компонентов Системы, после чего запускает подпрограмму подготовки экранов отображения информации. Далее, с заданной периодичностью (или по команде оператора) подпрограмма контроля состава проверяет изменения состава компонентов Системы и при наличии изменений снова запускает подпрограмму подготовки экранов.

Подпрограмма подготовки экранов отображения информации при запуске подгружает из хранилища шаблонов необходимые для формирования экранов шаблоны, формирует по заданным алгоритмам программные коды экранов отображения информации и запускает подпрограмму загрузки экранов.

Подпрограмма загрузки экранов по соответствующему программному интерфейсу (API Grafana) загружает сформированные программные коды экранов в модуль логирования и контроля показателей работы Системы.

Служебная подпрограмма управления Модулем обеспечивает взаимодействие оператора Системы с Модулем при необходимости ручного управления или проверки его работоспособности.

Программный код каждой подпрограммы хранится в отдельном файле. Список файлов с программными кодами Программы приведен в таблице.

Таблица 9 – Компоненты (подпрограммы), входящие в состав Модуля логирования и контроля метрик производительности работы системы.

	Наименование	Описание
	geocas.js	Основная подпрограмма
	controller.js	Подпрограмма контроля состава
	generator.js	Подпрограмма формирования экранов
	loader.js	Подпрограмма загрузки экранов
	web.js	Служебная подпрограмма управления

Хранилище шаблонов представляет собой файловое хранилище, организованное средствами операционной системы, в котором содержатся программные коды шаблонов типовых компонентов Системы, которые используются для формирования экранов отображения информации.

Каждый шаблон содержит в себе информацию о типе компонента, источнике данных для компонентов этого типа, параметрах контроля и их пороговых значениях, а также, графическое изображение компонента Системы.

Общая схема работы Модуля логирования и контроля метрик производительности работы в составе системы во взаимодействии с другими модулями и подсистемами, представлена в Приложении А1.

Программное обеспечение

Обоснование выбора языка программирования

ПО Модуля логирования и контроля метрик подразумевает организацию доступа к мониторингу платформы GEOKASKAD Marine дистанционно, при помощи сети интернет и современных средств отображения информации в веб-приложении.

В связи с этим, для разработки приложения используется язык программирования JavaScript на базе движка NodeJS.

Преимуществами NodeJS являются:

полнофункциональная разработка с использованием JavaScript;

оптимальная скорость работы приложений;

большое количество бесплатных инструментов;

кроссплатформенность.

Высокая скорость работы движка NodeJS обусловлена тем, что он разработан на основе V8, реализованного на C++ и используется для компиляции функций, написанных на JavaScript, в машинный код, выполняя компиляцию с высокой скоростью.

NodeJS поддерживает асинхронную обработку запросов, что важно при многопользовательском режиме использования приложения. Также в NodeJS поддерживается событийная модель, позволяющая разрабатывать современные веб-приложения.

Разработка программного обеспечения

Технические средства

В соответствии с требованиями технического задания, Модуль входных интерфейсов, должен запускаться на одном из четырех виртуальных, или физических серверов платформы, имеющем следующие технические характеристики:

CPU 12 ядер 2,4 ГГц и выше;

RAM 24 Гб и выше;

SSD 200 Гб и выше.

Программные средства

Программа работает на программной платформе NodeJS версии не ниже 10.24 под управлением операционной системы из семейства Linux.

В работе Программа использует сторонние программные компоненты платформы NodeJS. Список используемых сторонних программных компонентов и их назначение приведены в таблице.

Таблица 10 – Перечень используемых сторонних программных компонентов.

	Наименование	Описание
	fs	Работа с файловой системой операционной системы
	xmlbuilder2	Формирование программных кодов экранов отображения информации
	axios	Взаимодействие с другими модулями Системы через соответствующие программные интерфейсы (API)
	http	Организация удаленного управления Модулем через web-интерфейс (http-запросы)
	querystring	Обработка и разбор параметров запросов удаленного управления

Для удаленного взаимодействия по сети с другими модулями Системы Программа использует стандартные протоколы коммуникации, такие как TCP/IP и ICMP, работа которых обеспечивается встроенными средствами операционной системы.

Запуск программы

Запуск программы осуществляется по команде операционной системы из консоли в директории, в которой установлена программа.

Для запуска Программы нужно подключиться в терминальном режиме к серверу, на котором установлена Программа, перейти в соответствующую директорию, например, используя

cd /prog/dir

и подать команду:

для запуска Программы в режиме отладки с выводом отладочной информации на экран:

node geocas.js

для запуска Программы в обычном режиме:

nohup node geocas.js > /dev/null 2>&1 &

Входные данные

1.1.1.1.1. Основная подпрограмма

Входные данные Программы (основной подпрограммы) хранятся в файле настроек и имеют следующий вид:

```
{
```

```
    "storage":
          {"dir": "boilerplates"
          },
         "web":
          {"host": "0.0.0.0",
           "port": 8080
          },
         "prometheus":
          {"url": "http://localhost:8081",
           "accessKeyFile": ".geocas_prometheus_key",
           "database": "dbName"
          },
        "loki":
          {"url": "http://localhost:3200",
           "accessKeyFile": ".geocas_loki_key"
          },
         "grafana":
          {"url": "http://localhost:8082",
           "accessKeyFile": ".geocas_grafana_key",
           "dashboardId": "xxxxxxxxx",
           "panelName": "Схема системы"
          }
}
```

Описание каждого из параметров приводится в таблице.

Таблица 11 – Описание полей данных, поступающих на вход основной подпрограмме Модуля логирования и контроля метрик производительности работы системы.

	Группа	Параметр	Описание
	storage	dir	Директория файлового хранилища
	web	host	Сетевой адрес для подпрограммы управления
		port	Порт для подпрограммы управления
	prometheus	url	Адрес для доступа к модулю сбора метрик
		accessKeyFile	Ключ доступа
		database	Имя базы данных
	loki	url	Адрес для доступа к модулю сбора логов
		accessKeyFile	Ключ доступа
	grafana	url	Адрес для доступа к модулю контроля метрик
		accessKeyFile	Ключ доступа
		dashboardId	Код экрана отображения
		panelName	Заголовок экрана отображения

1.1.1.1.2. Подпрограмма контроля состава

Входные данные подпрограммы контроля состава аналогичны входным данным основной подпрограммы из группы prometheus и loki.

1.1.1.1.3. Подпрограмма формирования экранов

В качестве входных данных подпрограмма формирования экранов использует выходные данные от подпрограммы контроля состава, выходные данные хранилища шаблонов, а также данные файла настроек из группы storage.

1.1.1.1.4. Подпрограмма формирования экранов

В качестве входных данных подпрограмма загрузки экранов использует выходные данные от подпрограммы формирования экранов, а также данные файла настроек из группы grafana.

Выходные данные

1.1.1.1.5. Подпрограмма контроля состава

Подпрограмма на выходе формирует сгруппированный по блокам и столбцам JSON-список компонентов Системы, по которым осуществляется сбор метрик.

```
[[{"name": <string>,
        "type": <string>,
        "elems": [{"name": <string>,
                "id": <string or integer>,
                "value": <boolean>}
               ]
  }]]
```

Компоненты одного типа объединяются в блоки. Блоки по заданному в подпрограмме алгоритму объединяются в столбцы. Описание параметров приведено в таблице.

Таблица 12 – Описание полей данных, поступающих на выход подпрограммы контроля состава.

	Группа	Параметр	Описание
	Блок	name	Название блока для отображения на экране
		type	Тип компонентов в блоке
		elems	Список компонентов Системы
	Компонент	name	Название компонента для отображения на экране
		id	Идентификатор компонента в базе хранения метрик
		value	Флаг отображения значения метрики на экране

1.1.1.1.6. Хранилище шаблонов

JSON-структура, описывающая в формате grafana:
метрики для компонентов Системы указанного типа;

их пороговые значения;
цвета отображения компонента на экране.
Далее приводится пример такой JSON-структуры:


```
{
 "order": 1,
 "pattern": "PROBE-1",
 "unit": "short",
 "type": "number",
 "metricType": "serie",
 "alias": "Probe 1",
 "refId": "PROBE-1",
 "column": "Time",
 "hidden": false,
 "aggregation": "current",
 "decimals": 2,
 "colors": [
  "rgba(50, 172, 45, 0.97)",
  "rgba(237, 129, 40, 0.89)",
  "rgba(245, 54, 54, 0.9)"
 ],
 "reduce": true,
 "dateFormat": "YYYY-MM-DD HH:mm:ss",
 "thresholds": [
  50,
  70
 ],
 "stringThresholds": [
  "/.*/",
  "/.*/"
 ],
 "invert": true,
 "gradient": false,
 "overlayIcon": false,
 "tooltip": true,
 "tooltipLabel": "",
 "tooltipColors": false,
 "tooltipOn": "a",
 "tpDirection": "v",
 "tpGraph": true,
 "tpGraphSize": "100%",
 "tpGraphType": "line",
 "tpGraphLow": null,
 "tpGraphHigh": null,
 "tpGraphScale": "linear",
 "shapeProp": "id",
```

```
        "shapeRegEx": false,
        "shapeData": [
          {
            "pattern": "PROBES-BLOCK-ELEMENT-1",
            "hidden": false,
            "style": "fillColor",
            "colorOn": "a"
          },
          {
            "pattern": "PROBES-BLOCK",
            "hidden": false,
            "style": "strokeColor",
            "colorOn": "wc"
          }
        ],
        "textProp": "id",
        "textRegEx": false,
        "textData": [
          {
            "pattern": "PROBES-BLOCK-ELEMENT-1-VALUE",
            "hidden": false,
            "textReplace": "content",
            "textPattern": "/.*/",
            "textOn": "wmd"
          }
        ],
        "linkProp": "id",
        "linkRegEx": true,
        "linkData": [],
        "eventProp": "id",
        "eventRegEx": false,
        "eventData": [],
        "mappingType": 1,
        "valueData": [],
        "rangeData": [],
        "sanitize": false
}
```

XML-структура, описывающая в формате drawio графическое изображение компонента Системы на экране:

```
        <?xml version="1.0"?>
        <mxGraphModel>
          <root>
            <mxCell id="0"/>
            <mxCell id="1" parent="0"/>
```

```
            <mxCell id="PROBES-BLOCK" value="Probes"
style="rounded=1;whiteSpace=wrap;html=1;fontColor=#737373;fillColor=#FFFFFF;stroke
Color=#32ae2d;verticalAlign=top;strokeWidth=3;" parent="1" vertex="1">
              <mxGeometry x="20" y="20" width="220" height="450" as="geometry"/>
            </mxCell>
            <mxCell id="PROBES-BLOCK-ELEMENT-1" value="Probe 1"
style="rounded=1;whiteSpace=wrap;html=1;fontColor=#737373;fillColor=#FFFFFF;stroke
Color=#919191;" parent="PROBES-BLOCK" vertex="1">
              <mxGeometry x="20" y="30" width="120" height="40" as="geometry"/>
            </mxCell>
            <mxCell id="PROBES-BLOCK-ELEMENT-1-VALUE" value="value"
style="text;html=1;strokeColor=none;fillColor=none;align=center;verticalAlign=middle;whit
eSpace=wrap;rounded=0;fontColor=#737373;" parent="PROBES-BLOCK" vertex="1">
              <mxGeometry x="160" y="40" width="40" height="20" as="geometry"/>
            </mxCell>
          </root>
</mxGraphModel>
```

1.1.1.1.7. Подпрограмма формирования экранов

JSON-структура, описывающая в формате grafana экран отображения информации (пример):

```
          {
           "annotations": {
            "list": [
             {
              "builtIn": 1,
              "datasource": "-- Grafana --",
              "enable": true,
              "hide": true,
              "iconColor": "rgba(0, 211, 255, 1)",
              "name": "Annotations & Alerts",
              "type": "dashboard"
             }
            ]
           },
           "editable": true,
           "gnetId": null,
           "graphTooltip": 0,
           "id": 14,
           "links": [],
           "panels": [
            {
             "datasource": "TestData DB",
             "fieldConfig": {
```

```
      "defaults": {
        "custom": {}
      },
      "overrides": []
    },
    "flowchartsData": {
      "flowcharts": [
        {
          "allowDrawio": false,
          "bgColor": null,
          "center": false,
          "csv": "",
          "download": false,
          "editorTheme": "dark",
          "editorUrl": "https://www.draw.io",
          "enableAnim": true,
          "grid": false,
          "lock": true,
          "name": "Main",
          "scale": false,
          "tooltip": true,
          "type": "xml",
          "url": "http://<YourUrl>/<Your XML/drawio file/api>",
          "xml": "",
          "zoom": "100%"
        }
      ]
    },
    "format": "short",
    "graphId": "flowchart_23763571993",
    "gridPos": {
      "h": 14,
      "w": 14,
      "x": 0,
      "y": 0
    },
    "id": 23763571993,
    "newFlag": false,
    "pluginVersion": "7.3.7",
    "rulesData": {
      "rulesData": []
    },
    "targets": [],
  "schemaVersion": 26,
  "style": "dark",
```

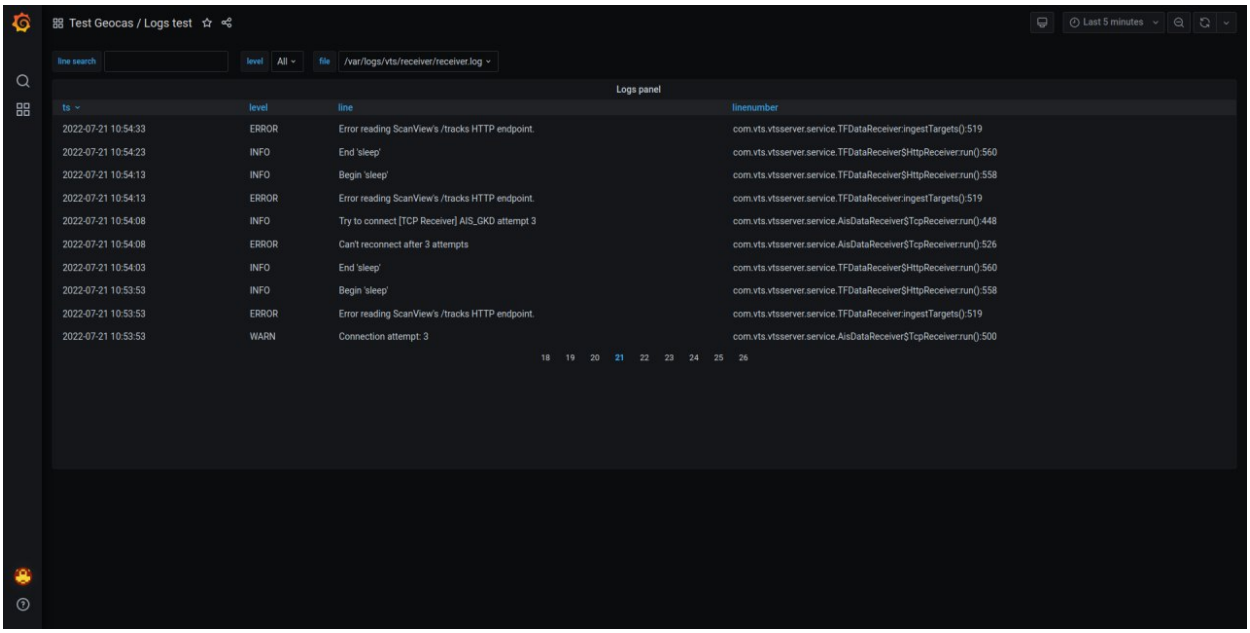
```
        "tags": [],
        "templating": {
          "list": []
        },
        "time": {
          "from": "now-6h",
          "to": "now"
        },
        "timepicker": {},
        "timezone": "",
        "title": "Flowchart test",
        "uid": "xxxxxxxxx",
        "version": 67
}
```


Модуль логирования и контроля метрик производительности работы системы предоставляет возможность программирования метрик и их визуального отображения при помощи средств Grafana. Структура диаграмм, визуализирующих получаемые метрики полностью задается настройками модуля.

На рис. 4 представлен дашборд визуализации метрик работы СУДС, основанной на разрабатываемой геоинформационной платформе реального времени. Каждый блок отображаемой диаграммы (а) своим цветом сигнализирует о состоянии компонента системы. Также модуль предоставляет доступ к логам соответствующего компонента (б), позволяя администратору системы, при необходимости, разобраться в причине ошибок и сбоев в работе.

Рисунок 4. Визуальный интерфейс дашбордов Grafana, формируемых модулем логирования и контроля метрик производительности работы системы.

(а) дашборд компонентов системы, отображающихся в соответствии с метриками



(б) интерфейс доступа к логам компонентов

РАЗРАБОТКА АРХИТЕКТУРЫ И КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОДСИСТЕМЫ ХРАНЕНИЯ ДАННЫХ

Назначение и краткое описание

Подсистема хранения данных предназначена для хранения данных, поступающих от источников данных, подключенных посредством Модуля входных интерфейсов платформы GEOKASKAD Marine.

Подсистема хранения данных обеспечивает два различных механизма хранения данных геоинформационной интегрирующей платформе реального времени GEOKASKAD Marine, разрабатываемой в рамках Договора (Соглашения) №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ:

Хранение потоковых данных сенсоров и иного измерительного оборудования в реляционной СУБД с поддержкой построения индекса по времени получения пакетов данных;

Хранение объектного представления данных, таких, как радарное видео, ДЗЗ и проч.

Для хранения данных первого типа была выбрана открытая реляционная система управления базами данных, с расширением TimescaleDB и PostGIS.

В качестве объектного хранилища используется открытое ПО Minio. В последующих разделах приводится обоснование данного выбора.

Обоснование логической структуры

Обоснование используемых методов и алгоритмов

При анализе выбора используемых средств хранения данных и реляционного и нереляционного типа, был проведен сравнительный анализ существующих средств с точки зрения оптимальности их применения в составе модуля хранения данных платформы реального времени GEOKASKAD Marine.

Было выделено два принципиально отличающихся механизма хранения данных, востребованных в составе платформы:

Хранение неструктурированной, или слабо структурированной информации, такой, как растровые данные радарного видео[10].

Хранение структурированной информации, когда хранению подлежит серия относительно небольших по размеру информационных блока с определенной структурой, детерминируемой тем или иным форматом или протоколом. Примерами таких данных являются пакеты данных АИС, целей РЛС, метеорологическая информация и иная телеметрия, где каждый поступающий пакет данных имеет заданный набор полей и временную привязку.

[10] В перспективе может возникнуть необходимость хранения данных видеонаблюдения, а также промежуточного хранения данных, публикуемых OGC-сервисами, однако, это не требуется на данном этапе.

Рассмотрим хранилища различных типов.

Блочные хранилища, часто используемые для виртуализации, имеют высокую скорость и надежность работы – хост-адаптер шины данных позволяет разгрузить процессор и освобождает его ресурсы для выполнения других задач. Недостатками блочного хранилища являются высокая стоимость и сложность в управлении. Также блочные хранилища имеют существенные ограничения по объему метаданных, которые могут относиться к хранимым объектам. Любую дополнительную информацию необходимо обрабатывать на уровне приложений и БД.

Файловые хранилища просты – файл имеет некоторое имя, а также некоторый набор метаданных, а затем размещается в каталогах и подкаталогах. Файловые хранилища обычно дешевле по сравнению с блочными системами, а их иерархическая структура удобна при обработке небольших объемов данных, в том числе, интерактивно. Поэтому с их помощью организуются системы совместного использования файлов и системы локального архивирования. Файловые хранилища также имеют ограничения в части объемов хранимой метаинформации.

Объектные хранилища хорошо масштабируются, поэтому объемы хранимых в них данных могут достигать петабайтов информации. По статистике, объем неструктурированных данных во всем мире на 2020 составляет 44 зеттабайт, что в 10 раз превышает показатели 2013. Объектные хранилища, благодаря своей возможности работать с растущими объемами данных, стали стандартом для большинства из самых популярных облачных сервисов. Такие сервисы часто используются для хранения медиафайлов и резервных копий, что показывает адаптированность для хранения такого рода неструктурированной информации.

Объектные хранилища оснащены внутренними механизмами для проверки корректности файлов и другими функциями, обеспечивающими доступность данных.

На данном этапе развития платформы, выделен один тип неструктурированных данных, требующих хранения в объектном хранилище – данные радарного видео, поступающие с РЛС.

Ко второй категории данных, структурированных и, имеющих привязку ко времени измерения и получения пакета, необходимо хранить в реляционном или нереляционном (NoSQL) хранилище для обеспечения возможности его использования для последующих ретроспективы и аналитических выборок.

Был проведен сравнительный анализ реляционного хранилища на базе открытой СУБД PostgreSQL и нереляционных систем Apache Cassandra и Apache HBase.

Сравнение нереляционных хранилищ Apache Cassandra и Apache HBase приводят к следующим результатам, приведенным в таблице.

Таблица 13 – Сравнение Apache Cassandra и Apache HBase[11].

	Критерий	Apache HBase	Apache Cassandra
	Архитектура	Кластер HBase работает по принципу ведущий/ведомый (master/slave), когда главный сервер (Master Node) управляет остальными узлами. Поэтому	Apache Cassandra, организована в виде кольца равнозначных узлов, по которым распределены данные. Побочным эффектом такой отказоустойчивости является

[11] По материалам Школы больших данных: https://www.bigdataschool.ru/news/cassandra-hbase-difference.html

		отказ главного сервера влечет за собой сбой всего кластера. При этом, несмотря на наличие глобальной точки отказа, такая архитектура позволяет обеспечить согласованность данных на всех распределенных узлах.	проблема согласованности данных, когда одна и та же информация на разных узлах может отличаться. Для борьбы с этим предусмотрены настраиваемые уровни согласованности (consistency levels).
	Инфраструктура	Основана на Hadoop - для работы HBase необходимы компоненты Apache Hadoop: распределенная файловая система HDFS для хранения данных и служба Zookeeper для координации работ между сервисами, управления их конфигурациями и синхронизацией.	Самодостаточна и не зависит от дополнительных внешних компонент и файловых хранилищ.
	Средства работы с данными	HBase предоставляет более широкий набор возможностей: REST и другие API-интерфейсы Java, а также внешние SQL-решения, такие, как Apache Phoenix, Drill, Hive и Cloudera Impala, позволяющие работать с данными, хранящимися в HBase, как с реляционными таблицами.	Имеет собственный SQL-подобный язык запросов CQL, позволяющий выполнять простейшие выборки по определённому условию. Операторы добавления и обновления информации отсутствует. Драйверы с поддержкой CQL реализованы для языков Python, Java, Ruby, PHP, JavaScript и Perl
	Поддержка индексации	В HBase индексация возможна только по одному полю – первичному ключу. Добавление второго индекса возможно только с использованием сторонних продуктов, таких, как Apache Phoenix.	Cassandra поддерживает вторичные индексы, создаваемые при помощи CQL-выражения CREATE INDEX
	ACID-транзакции	Не обеспечивает все ACID-требования к транзакциям (атомарность, согласованность, изолированность, долговечность).	Cassandra обеспечивает ACID-транзакции на уровне одной записи, т. е. для набора столбцов с одним ключом.

Приведенное сравнение позволяет, при выборе NoSQL хранилища, отдать предпочтение Apache Cassandra, в основном, учитывая ее более высокий уровень независимости от использования сторонних продуктов. При этом возможности Cassandra в части обеспечения сложных выборок данных (как и в случае любых NoSQL хранилищ), ограничены и не позволят осуществлять сложной аналитики.

Одним из самых распространенных открытых решений, при этом, обладающих всеми свойствами профессиональной СУБД, является СУБД PostgreSQL. Основными недостатками базовой версии PostgreSQL является отсутствие в ней поддержки

геопространственных данных и временных рядов. Оба этих недостатка можно преодолеть с использованием плагинов TimescaleDB и PostGIS.

Расширение PostGIS разрабатывается в соответствии со спецификациями OGC, поэтому соответствует всем необходимым требованиям по хранению данных, имеющих геопространственную привязку, в платформе GEOKASKAD Marine.

Расширение TimescaleDB позволяет PostgreSQL работать с большими временными рядами данных, аналогично тому, как это делают NoSQL хранилища (нагрузочные стандартные нагрузочные тесты для временных рядов доступны на https://github.com/timescale/tsbs и показывают достаточно высокие характеристики), но при этом, оставляя возможность выполнения классических SQL-запросов.

В Приложении Н приводится скорость записи в строках в секунду, которую обеспечивает расширение TimescaleDB (также для сравнения показана деградация производительности при использовании базовой версии PostgreSQL без расширений). Представленная на иллюстрации средняя скорость записи в 120 000 строк в секунду, удовлетворяет требованиям по нагрузке, предъявляемым техническим заданием.

В связи с вышеизложенным анализом было принято решение использовать в качестве основного средства для хранения СУБД PostgreSQL с расширением TimescaleDB. В перспективе, если в этом возникнет необходимость, будет рассмотрена возможность дополнения платформы GEOKASKAD Marine средствами поддержки Apache Cassandra.

Компоненты модуля и взаимодействия между ними

Модуль выходных интерфейсов включает в состав программные компоненты, описанные в следующей таблице.

Таблица 14 – Компоненты Модуля выходных интерфейсов

	Наименование компонента	Описание модуля
	Minio	Объектное хранилище на основе открытого ПО Minio.
	PostgreSQL	СУБД PostgreSQL, дополненная расширениями TimescaleDB и PostGIS.

Компоненты Модуля выходных интерфейсов представлены на иллюстрации в Приложении А1.

Связи модуля с другими компонентами платформы

Модуль выходных интерфейсов непосредственно взаимодействует со следующими модулями платформы реального времени GEOKASKAD Marine:

модуль входных интерфейсов (на получение данных);

модуль предварительной обработки и фильтрации данных (на предоставление данных);

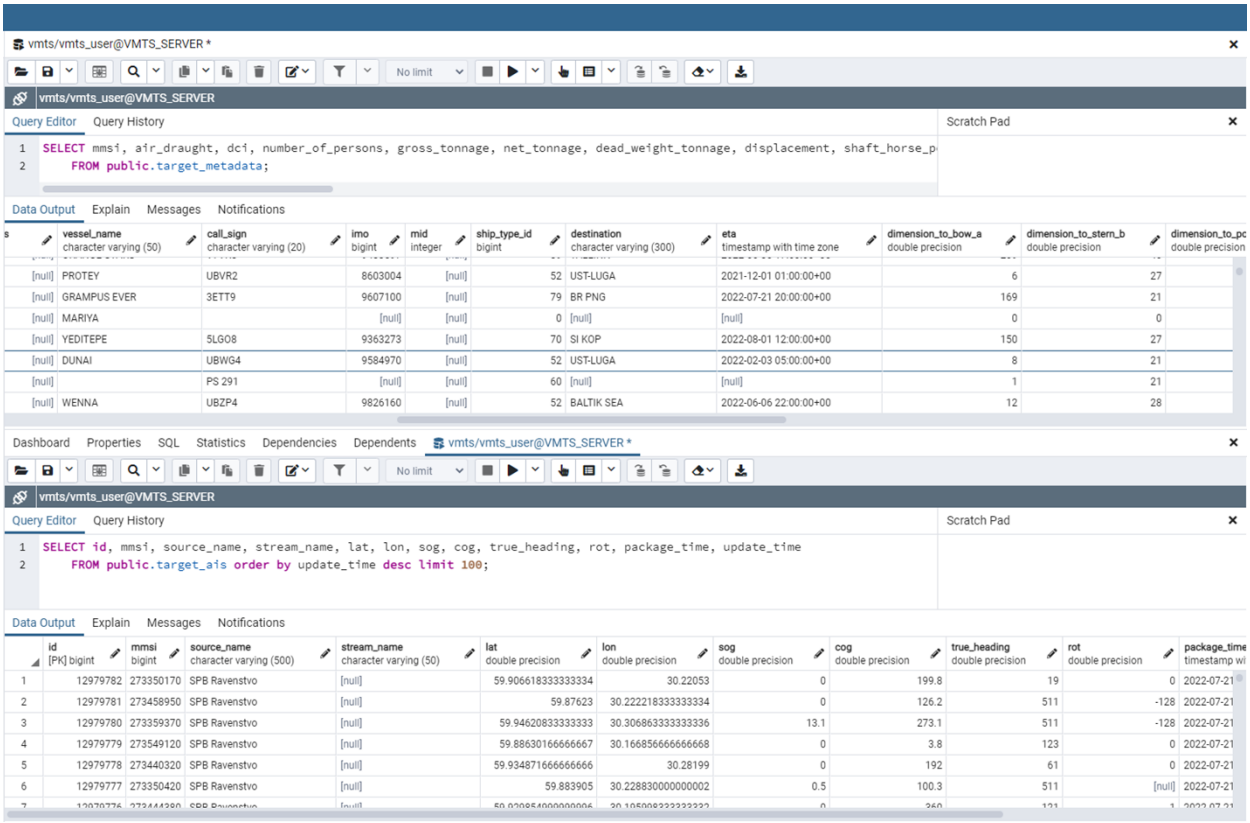
модуль выходных интерфейсов (на предоставление данных).

Модуль логирования и контроля метрик может быть использован для сбора и визуализации метрик СУБД. Администрирование БД через Модуль администрирования не предусмотрено, однако он может хранить в БД вспомогательную информацию.

Общая схема взаимодействия между компонентами платформы GEOKASKAD Marine представлена на иллюстрации в Приложении А1.

Подсистема хранения данных не имеет собственного визуального интерфейса, однако содержимое рабочих таблиц может быть визуализировано при помощи любого стандартного клиента доступа к данным PostgreSQL, например, pgAdmin (см. рис.6).

Рисунок 6. Примеры выборки данных из таблиц с данными AIS Position Report (1-3 сообщение АИС) и AIS Static and Voyage Report (5 сообщение АИС) при помощи SQL-запроса.



РАЗРАБОТКА АРХИТЕКТУРЫ И КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЯ АДМИНИСТРИРОВАНИЯ

Назначение и краткое описание

Модуль администрирования предназначен для централизованного управления всеми модулями платформы GEOKASKAD Marine путем взаимодействия с контейнерами других модулей и подсистем в рамках микросервисной архитектуры.

Модуль администрирования имеет гибкую расширяемую архитектуру и на данном этапе развития Морской геоинформационной интегрирующей платформе реального времени GEOKASKAD Marine, разрабатываемой в рамках Договора (Соглашения) №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ, обеспечивает управление следующими модулями и подсистемами:

Модуль входных интерфейсов;

Модуль предварительной обработки и фильтрации данных;

Модуль выходных интерфейсов;

Подсистема хранения данных.

Модули входных интерфейсов, предварительной обработки и фильтрации данных, а также выходных интерфейсов управляются посредством визуального интерфейса Модуля администрирования в то время, как управление настройками Подсистемы хранения данных осуществляется посредством конфигурационных файлов, ввиду отсутствия необходимости регулярного взаимодействия пользователя с данными настройками.

Модуль администрирования предназначен для обеспечения гибкого управления всеми программными модулями, входящими в состав опытного образца Платформы, включая:

контроль прав доступа пользователей Платформы к системным настройкам;

визуализацию текущих настроек модулей и компонентов, подключенных к Модулю администрирования с учетом прав доступа;

редактирование и сохранение изменений текущих настроек модулей и компонентов, подключенных к Модулю администрирования с учетом прав доступа;

фиксацию изменений и времени изменения настроек компонентов Платформы с целью возможного последующего анализа причин сбоев в работе системы;

интеграция с модулем логирования и контроля метрик производительности работы системы с целью визуализации логов и метрик подключенных компонентов Платформы.

Обоснование логической структуры

Обоснование используемых методов и алгоритмов

При выборе архитектуры Модуля администрирования, следует учитывать следующее:

Принимая во внимание современные тенденции развития ИТ-отрасли, Модуль администрирования должен иметь эргономичный пользовательских интерфейс, доступный посредством технологии тонкого клиента;

Модуль администрирования должен иметь в своем составе средства баз данных, обеспечивающие возможность персистентного хранения выбранных настроек модулей, входящих в состав платформы;

Модуль администрирования должен обеспечивать серверные механизмы взаимодействия с управляемыми им компонентами платформы GEOKASKAD Marine.

Учитывая вышеперечисленные факторы, для разработки Модуля администрирование была выбрана трехуровневая, или трехзвенная архитектура.

Для реализации слоя клиента была разработана веб-консоль Модуля администрирования, реализованная с использованием тонкого клиента и взаимодействие с которой происходит при помощи окна браузера.

Слой данных в составе Модуля администрирования был разработан с использованием открытой СУБД PostgreSQL. Данный выбор обусловлен широким распространением данной БД, а также ее использованием в рамках Подсистемы хранения данных. Использование PostgreSQL позволит переиспользование СУДБ без усложнения архитектура платформы в целом, а также, не расширяя состав используемого общесистемного программного обеспечения, необходимого для функционирования платформы реального времени GEOKASKAD Marine.

Компоненты модуля и взаимодействия между ними

Модуль выходных интерфейсов включает в состав программные компоненты, описанные в следующей таблице.

Таблица 15 – Компоненты Модуля администрирования

	Наименование	Описание
	Admin Service	Серверное приложение (слой сервера приложений) в составе:
	auth	Модуль[12] авторизации, обеспечивающий разграничение прав доступа к механизмам администрирования платформы GEOKASKAD Marine на основании имени пользователя и пароля.
	config	Модуль конфигурирования серверного слоя Модуля администрирования, включающий в себя: – средства конфигурирования модуля, как приложения; – средства конфигурирования механизмов разграничения доступа; – средства конфигурирования параметров и настроек приложения в зависимости от авторизованного пользователя.
	dao	Модуль организации уровня доступа к данным – взаимодействует со слоем хранения данных в рамках трехзвенной архитектуры Модуля администрирования.
	exception	Модуль обработки ошибок.

[12] В рамках данной таблицы компонентов Модуля администрирования, под модулями понимаются программные модули в составе Admin WebClient и Admin Service.

	helpers	Модуль вспомогательных средств, используемый прочими модулями для выполнения однотипных процедур.
	payload	Модуль формирования данных, передаваемых Модулем администрирования контейнерам прочих модулей и подсистем платформы GEOKASKAD Marine при их конфигурировании, запуске и остановке.
	web.api	Модуль контроллера веб-приложения, обеспечивающий взаимодействие между Admin WebClient и Admin Service посредством REST API.
	Admin WebClient	Клиентское приложение (слой клиента) в составе следующих основных[13] модулей:
	profile	Модуль, обеспечивающий ввод группы настроек, отвечающих за – источники данных, – фильтры и настройки предварительной обработки, – выходные интерфейсы.
	controls	Модуль, обеспечивающий работу основных инструментов, включающий, как базовые (такие, как ввод значений и параметров), так и специализированные (отображение диаграмм/блок-схем правил фильтрации, картографическое представление) инструменты веб-консоли.
	panel	Модуль, обеспечивающий визуальное представление панелей web-приложения.
	Container Environment (Docker)	Среда, обеспечивающая работу с контейнерами, на основе технологии Docker, параметры и конфигурация развертывания и запуска которых задается скриптами docker-compose.

Укрупненная схема компонентов Модуля администрирования, а также их взаимодействие в составе платформы, представлена на иллюстрации в Приложении А1.

Связи модуля с другими компонентами платформы

Модуль администрирования непосредственно взаимодействует со следующими модулями платформы реального времени GEOKASKAD Marine:

модуль входных интерфейсов;

модуль предварительной обработки и фильтрации данных;

модуль выходных интерфейсов;

модуль логирования и контроля метрик производительности работы системы.

Данные, поступающие в платформу реального времени посредством Модуля входных интерфейсов, сохраняются в оперативную базу данных Подсистемы хранения данных, проходят предварительную обработку и фильтрацию, а затем предоставляются пользователям и/или сторонним и смежным ИТ-системам посредством модуля выходных интерфейсов.

[13] В таблице представлены только основные модули, т. е. исключены модули, имеющие служебное внутреннее назначение.

Управление логами и метриками, в том числе, Модуля входных интерфейсов, осуществляется Модулем логирования и контроля метрик производительности работы системы.

Модуль администрирования обеспечивает настройку параметров Модуля входных интерфейсов, с использованием визуального пользовательского интерфейса, основанного на тонком клиенте, доступном при помощи браузера.

Общая схема взаимодействия между компонентами платформы GEOKASKAD Marine представлена на иллюстрации в Приложении А1.

Программное обеспечение

В данном подразделе приводятся основные результаты разработки и тестирования программного обеспечения, входящего в состав Модуля администрирования. Более подробная информация о разработанном программном обеспечении и его тестировании, представлены в [13] и [14].

Обоснование выбора языка программирования

При выборе языка программирования для Модуля администрирования принимались во внимание следующие требования к языку программирования.

Поддержка современных технологий web-разработки, в связи с необходимостью разработки клиентской составляющей Модуля администрирования на основе технологии тонкого клиента, запускаемого в браузере;

Кроссплатформенность. Модуль администрирования может устанавливаться, как на собственных вычислительных мощностях, так и на мощностях заказчика, выбор операционной системы которым, зачастую обусловлен другими используемыми им программными продуктами. В связи с этим, наличие кроссплатформенности значительно повышает конкурентоспособность решения;

Встроенная поддержка современных средств разграничения доступа к функциональным возможностям приложения;

Поддержка высокого уровня интеграции с иными корпоративными системами, в связи с высокой вероятностью их совместного использования в составе программных комплексов порта и иных объектов морской отрасли.

В перечень кандидатов были выбраны такие языки программирования, как C++, Python и Java. Некоторые из альтернативных вариантов были исключены ввиду ограничений в части требований кроссплатформенности.

При выборе основы для клиентской составляющей, учитывалось повсеместное распространение и универсальность языка программирования JavaScript, поэтому выбор средств разработки клиентской составляющей ограничивался выбором программного фреймворка и библиотек разработки. В силу широкого распространения и большого по численности сообщества разработчиков, для реализации клиентской составляющей был выбран фреймворк ReactJS.

C++, несмотря на высокую производительность, больше ориентирован на разработку системных, нежели корпоративных и web-приложений.

Python, несмотря на простоту и широкое распространение, имеет недостаточно высокую производительность, не обеспечивая требуемых характеристик в части обработки данных в реальном времени.

Сильные стороны Java проявляются при разработке корпоративных приложений ввиду развитой экосистеме разработки, что является значимым фактором при его использовании в составе серверной части Модуля администрирования. Кроме того, при разработке приложений на языке программирования Java возможно использование программного фреймворка Spring Boot, который включает в себя такие необходимые для модуля компоненты, как

- Spring Security для организации современных механизмов разграничения доступа,
- Spring JDBC для организации работы с СУБД, в том числе, выбранной СУБД PostgreSQL.

Таким образом, для реализации модуля входных интерфейсов была выбрана гибридная реализация с использованием языков программирования Java и JavaScript с использованием фреймворков Spring Boot и ReactJS, соответственно.

Обоснование программной архитектуры

В целях анализа и обоснования выбора трёхуровневой архитектуры, следует выделить следующие ее достоинства:

- масштабируемость – серверный слой и слой хранения данных относительно независимы друг от друга, что позволяет обеспечить, при необходимости, масштабирование системы, как в части повышения производительности работы СУБД, так и в части повышение ресурсов, доступных для компонентов серверного слоя;
- конфигурируемость – изолированность уровней друг от друга позволяет эффективно переконфигурировать систему при возникновении сбоев, или при плановом обслуживании на одном из уровней;
- высокая безопасность с точки зрения механизмов разграничения доступа, в том числе, с использованием ролевой модели;
- высокая надёжность – также следует из изолированности уровней, позволяя производить, как плановое обслуживание, так и обслуживание, при возникновении сбоев;
- низкие требования к скорости сетевого канала между терминалами (веб-приложением) и сервером приложений;
- низкие требования к производительности и техническим характеристикам клиентских компонентов и, как следствие снижение их стоимости.

При этом трехуровневая архитектура имеет следующие недостатки, являющиеся следствиями вышеперечисленных достоинств:

- более высокая сложность создания приложений, по сравнению с одноуровневыми и двухуровневыми системами. Однако, меньшая сложность, по сравнению с микросервисной архитектурой;
- сложнее в разворачивании и администрировании – может нивелироваться использованием Docker-контейнеров, упрощая и автоматизируя процедуру

развертывания. Использование тонкого клиента значительно упрощает развертывание клиентской составляющей ПО Модуля администрирования;

- высокие требования к производительности серверов приложений и сервера базы данных, а, значит, и высокая стоимость серверного оборудования – в данном случае, Модуль администрирования представляет собой визуально управляемый контекст выполнения прочих, управляемых им компонентов платформы GEOKASKAD Marine, что позволяет обеспечить работу модуля на достаточно ограниченных ресурсах;
- высокие требования к скорости канала (сети) между сервером базы данных и серверами приложений – не имеют на сегодняшний день решающего значения ввиду существенного и непрерывного роста производительности систем облачных вычислений. Так, используемая для разработки опытного образца платформы облачная архитектура Яндекс.Облако, позволяет обеспечить не менее 100 Мбит, а чаще до нескольких Гбит в части пропускной способности сетевого соединения между серверным слоем и слоем базы данных.

В связи с приведенным выше анализом, для реализации Модуля администрирования была выбрана трехуровневая (или трехзвенная) архитектура.

Разработка программного обеспечения

Технические средства

В соответствии с требованиями технического задания, Модуль администрирования, должен запускаться на одном из четырех виртуальных, или физических серверов платформы, имеющем следующие технические характеристики:

CPU 12 ядер 2,4 ГГц и выше;

RAM 24 Гб и выше;

SSD 200 Гб и выше.

Программные средства

Для функционирования Модуля администрирования необходимо следующее программное обеспечение:

операционная система Windows или Linux (Kernel>=5.3);

средства для контейнеризации Docker (> = 19.03) и Docker-Compose (> = 1.22);

система управления базами данных PostgreSQL (> = 13);

специализированная библиотека OpenLayers, предназначенная для визуализации картографической информации в составе web-приложения;

разработка клиентской части используется программный фреймворк ReactJS;

серверная составляющая разработана на языке программирования Java (>=8) и запускается с использованием запущенной на серверной стороне java-машины.

В целях реализации требований к Модулю администрирования, в части персистентного хранения настроек источников данных, слой баз данных содержит таблицы для их хранения (см. Приложение Г). Ниже приводятся описания части из них.

Источники данных АИС

Таблица 16 – Описание модели данных и таблицы sources_ais, предназначенной для хранения пользовательских источников категории АИС[14]

	Наименование поля	Тип поля	Обнуляемость	Описание
	id	int8	NO	Идентификатор
	source_name	varchar	NO	Название
	sources_type	ais_sources_type	NO	Тип
	remote_host	varchar	YES	Адрес хоста
	remote_port	int4	YES	Порт хоста
	lat	float8	YES	Координата, широта
	lon	float8	YES	Координата, Долгота
	user_id	int8	NO	Идентификатор пользователя, создавшего запись
	creation_time	timestamptz	NO	Фактическое время создания записи
	update_time	timestamptz	YES	Фактическое время последнего обновления записи
	deleted	int2	NO	Признак удаленной записи
	global	bool	NO	Признак глобального источника
	enable	bool	NO	Признак активности источника

Источники данных РЛС

Таблица 17 – Описание модели данных и таблицы sources_radar, предназначенной для хранения пользовательских источников категории РЛС[15]

	Наименование поля	Тип поля	Обнуляемость	Описание
	id	int8	NO	Идентификатор
	source_name	varchar	NO	Название
	lat	float8	YES	Координата, широта
	lon	float8	YES	Координата, долгота
	user_id	int8	NO	Идентификатор пользователя, создавшего запись
	creation_time	timestamptz	NO	Фактическое время создания записи

[14] Включает в себя все источники АИС и С-АИС, как поступающие с отдельных БС АИС, так и с глобальных сервисов, если такие подключены к платформе GEOKASKAD Marine.

[15] Включая, как интегрированные, так и неинтегрированные данные, неподтвержденные целями АИС.

	update_time	timestamptz	YES	Фактическое время последнего обновления записи
	deleted	int2	NO	Признак удаленной записи
	global	bool	NO	Признак глобального источника
	enable	bool	NO	Признак активности источника

Источники метеорологической информации

Таблица 18 – Описание модели данных и таблицы sources_meteo, предназначенной для хранения пользовательских источников метеорологической информации

	Наименование поля	Тип поля	Обнуляемость	Описание
	id	int8	NO	Идентификатор
	source_name	varchar	NO	Название
	sources_type	ais_sources_type	NO	Тип
	remote_host	varchar	YES	Адрес хоста
	remote_port	int4	YES	Порт хоста
	lat	float8	YES	Координата, широта
	lon	float8	YES	Координата, Долгота
	user_id	int8	NO	Идентификатор пользователя, создавшего запись
	creation_time	timestamptz	NO	Фактическое время создания записи
	update_time	timestamptz	YES	Фактическое время последнего обновления записи
	deleted	int2	NO	Признак удаленной записи
	global	bool	NO	Признак глобального источника
	enable	bool	NO	Признак активности источника

Также, постоянного хранения в БД требует информация о потоках данных, включая настройку входов, выходов и фильтров. Структура этих таблиц и связи между ними, представлены в Приложении Д к настоящему отчету, а назначения таблиц преведены в следующем списке:

flows – Общая информация о пуле потоков;

flow_sources – Настройки контейнеров, обрабатывающих потоки из внешних источников;

flow_filter – Настройки контейнера, фильтрующего потоковую информацию;

flow_outputs - Настройки контейнеров, выдающих информацию в нормализованном виде.

Запуск программы

Запуск программы (Docker-контейнера) осуществляется следующей командой:
docker-compose up -d

Остановка программы (Docker-контейнера) осуществляется командой:
docker-compose stop

Описание конфигурационного файла docker-compose.yaml приводится в Приложении М.

Входные данные

Взаимодействие между клиентской и серверной составляющей Модуля администрирования происходит при помощи вызова методов REST API, входные наборы к данным, к которым, представлены отдельно, для каждого их методов.

1.1.1.1.8. Источники АИС

Создание/изменение источника
POST request: /api/sources/ais
RequestBody (входные данные):

```
{
   "id": null,                    //Идентификатор источника
   "sourceName": "test-exact-earth",  //Название источника
   "sourceType": "EXACT_EARTH",          //Тип источника
   "remoteHost": "127.0.0.1",        //IP хоста источника
   "remotePort": 123456,                 //Port хоста источника
   "lat": 40.1253,                //Координата местонахождения источника (широта)
   "lon": 60.123,                 //Координата местонахождения источника (долгота)
   "global": false                //Флаг: глобальный(true)/нет(false) источник
   "enable": true                 //Флаг: активен(true)/не активен(false)
}
```

Удаление источника
DELETE request: /api/sources/ais/{id}
{id} – Идентификатор пользовательского источника

1.1.1.1.9. Источники РЛС

Создание/изменение источника
POST request: /api/sources/radar
RequestBody (входные данные):

```
{
```

```
  "id": null,                    //Идентификатор источника
  "sourceName": "test-exact-earth",  //Название источника
  "lat": 40.1253,                //Координата местонахождения источника (широта)
  "lon": 60.123,                 //Координата местонахождения источника (долгота)
  "global": false                //Флаг: глобальный(true)/нет(false) источник
  "enable": true                 //Флаг: активен(true)/не активен(false)
}
```

Удаление источника
DELETE request: /api/sources/radar/{id}
{id} – Идентификатор пользовательского источника

1.1.1.1.10. Источники метеорологической информации

Создание/изменение источника
POST request: /api/sources/meteo
RequestBody (входные данные):

```
{
  "id": null,                        //Идентификатор источника
  "sourceName": "meteo",     //Название источника
  "sourceType": "METEO",  //Тип источника
  "remoteHost": "127.0.0.1", //IP источника
  "remotePort": 123456,              //Port источника
  "lat": 40.1253,                //Координата местонахождения источника (широта)
  "lon": 60.123,                 //Координата местонахождения источника (долгота)
  "global": false                //Флаг: глобальный(true)/нет(false) источник
  "enable": true                 //Флаг: активен(true)/не активен(false)
}
```

Удаление источника
DELETE request: /api/sources/meteo/{id}
{id} – Идентификатор пользовательского источника

1.1.1.1.11. Работа с потоком данных

Создание/изменение потока
POST request: /api/flow
RequestBody (входные данные):

```
{
  "id": 1570,
  "flowType": "AIS",
  "name": "test_flow3333",
  "sources": [
```

```
    {
      "aisId": 1570,
      "aisSourcesType": "EXACT_EARTH",
      "sourceAisId": 1022,
      "connectionName": "exact_earth",
      "remoteHost": null,
      "remotePort": null,
      "enable": true
    }
  ],
  "outputs": [
    {
      "aisId": 1570,
      "aisOutputType": "WEB_SOCKET",
      "connectionName": "test",
      "remoteHost": null,
      "remotePort": 13014,
      "enable": true
    }
  ],
  "filters": {
    "aisId": 1570,
    "filterSql": "(msg_type==1)",
    "deduplicate": false,
    "filterGeo": {
      "type": "Polygon",
      "coordinates": [
        [
          [45.570438,47.836027],
          [55.085450,47.974595],
          [55.916859,36.334872],
          [46.124711,36.196304],
          [46.124711,36.196304],
          [45.570438,47.836027]
        ]
      ]
    },
    "filterGeoType": "WITHIN",
    "jsonTree": null
  },
  "sourcesIds": [
    1022
  ],
  "running": false,
  "userId": null
```

```
}
```

Описание полей объекта, передаваемого для создания потока:

"Id" – Идентификатор в системе

"flowType" – Тип пула (AIS, METEO, RADAR)

"name" – Название потока

"sources" – список настроек контейнеров, обрабатывающих внешние источники

- "aisId" – Идентификатор в системе
- "aisSourcesType" – Тип источника,
- "sourceAisId" – Идентификатор пользовательского источника,
- "connectionName" – Название соединения
- "remoteHost" – Адрес хоста источника
- "remotePort" – Порт хоста источника

"filters": - контейнер, обрабатывающий/накладывающий фильтр на потоковые данные

- "aisId" - Идентификатор в системе
- "filterSql" – SQL выражение в строковом виде
- "deduplicate" – Признак дедупликации
- "filterGeo" -
  "type": - Тип геообъекта (например POLYGON)
  "coordinates": Координаты геообъекта
- "filterGeoType" – геопространственный фильтр "NONE", "WITHIN", "OUTSIDE"

"outputs" - список настроек контейнеров, формирующих исходящие данные

- "aisId" - Идентификатор в системе
- "aisOutputType": Тип передачи (например WEB_SOCKET)
- "connectionName" – Название соединения
- "remoteHost" - Адрес хоста отдачи данных
- "remotePort" – Порт хоста отдачи данных

Запуск потока

POST request: /api/flow/run/{flowId}

{flowId} – Идентификатор потока в системе.

Остановка потока

POST request: /api/flow/stop/{flowId}

{flowId} – Идентификатор потока в системе.

Выходные данные

Модуль не предполагает выдачи выходных данных, за исключением экранных форм, визуализируемых клиентской составляющей программы:

Цифровая карта – компонент пользовательского интерфейса, ответственного за визуализацию источников данных, имеющих географическую привязку (см. иллюстрацию в Приложении Е);

Панель источников данных, выводит перечень подключенных источников данных различного типа (см. иллюстрацию в Приложении Е);

Панель настройки потоков данных, выводит перечень текущих параметров потока, также служит для их настройки (см. иллюстрацию в Приложении Ж). Общие настройки потока также визуализируются в виде диаграммы потока;

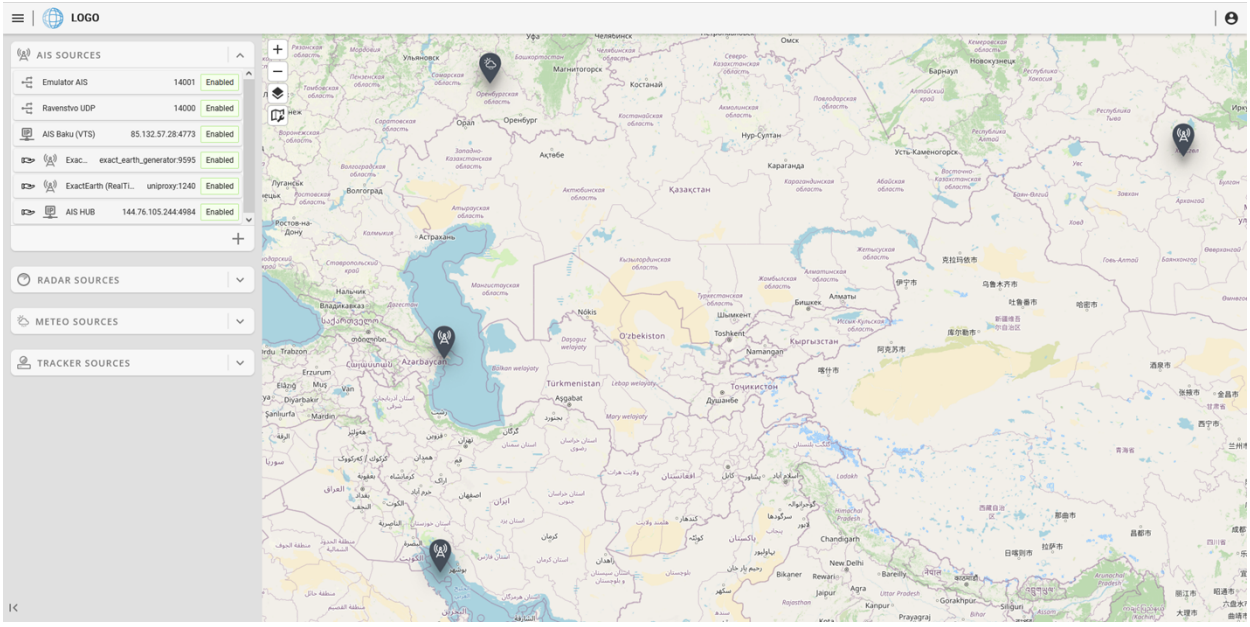
Панель потоков – компонент пользовательского интерфейса, отображающий перечень потоков, а также статус каждого из них, включен/выключен (см. иллюстрацию в Приложении З);

Интерфейс управления камерами видеонаблюдения – интерфейс, показывающий на карте все подключенные к системе камеры видеонаблюдения, их статус, зону обслуживания (см. иллюстрацию в Приложении И);

Интерфейс визуализации видеопотока – компонент пользовательского интерфейса, отображающий видеопоток с камер CCTV (см. иллюстрацию в Приложении К).

На рис.5 представлен визуальный интерфейс веб-консоли Модуля администрирования, предоставляющего доступ ко всем другим модулям платформы GEOKASKAD Marine. На карте представлены маркеры источников (в данном примере представлены базовые станции АИС и метеостанция), имеющих географическую привязку.

Рисунок 5. Визуальный интерфейс веб-консоли Модуля администрирования, предоставляющего доступ ко всем другим модулям платформы GEOKASKAD Marine.



РАЗРАБОТКА АРХИТЕКТУРЫ И КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЯ КОМПЛЕКСНОЙ ОБРАБОТКИ СОБЫТИЙ

Назначение и краткое описание

Модуль комплексной обработки событий предназначен для организации программируемого выявления событий заданного типа, соответствующего стандартным навигационным тревогам и оповещениям, утвержденным Приказом от «23» июля 2015 г. № 226 Министерства транспорта Российской Федерации, и формирования программных сообщений, клиентами которых могут выступать прочие программные компоненты платформы GEOKASKAD Marine, а также иные сторонние и смежные системы.

Модуль комплексной обработки событий функционирует в составе Морской геоинформационной интегрирующей платформе реального времени GEOKASKAD Marine, разрабатываемой в рамках Договора (Соглашения) №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ.

Модуль комплексной обработки событий поддерживает следующие утвержденные типы тревог и оповещений:

превышение максимального темпа изменения скорости целей (судов),

превышения максимального темпа изменения курса целей (судов),

скорость цели (судна) в пределах заданной зоны выше максимального допустимого значения,

скорость цели (судна) в пределах заданной зоны ниже минимального допустимого значения,

вход цели (судна) в заданную зону,

выход цели (судна) из заданной зоны,

пребывание цели (судна) в пределах заданной зоны,

опасное сближение целей (судов),

начало движения неподвижной цели (судна),

сближение цели (судна) с заданной точкой.

Данные, выдаваемые Модулем комплексной обработки событий, помещаются в очередь сообщений под управлением открытого программного обеспечения Apache Kafka и могут быть получены и десериализованы в соответствии с описанными далее моделями сообщений, всеми подключенными клиентскими приложениями.

Обоснование логической структуры

Обоснование используемых методов и алгоритмов

Для разработки Модуля комплексной обработки событий была выбрана микросервисная архитектура. Обоснование микросервисной архитектуры аналогично обоснованию архитектуры Модуля входных интерфейсов и приводится в разделе 3.3.2 соответствующего раздела настоящего отчета.

В соответствии с принятыми архитектурными решениями, как в случае Модуля входных интерфейсов, каждый компонент Модуля комплексной обработки событий реализован как отдельный контейнер, с использованием технологии Docker.

Выбор, как и в случае Модуля входных интерфейсов, был обусловлен следующими факторами:

независимость рабочего окружения контейнеризированного приложения Модуля комплексной обработки событий;

изоляция и безопасность, ввиду отделенности контейнеризированного приложения Модуля комплексной обработки событий от остальных контейнеров и операционной системы (включая общесистемные приложения и сервисы);

автоматизация развертывания приложений, входящих в состав Модуля комплексной обработки событий, а также масштабируемость.

Компоненты Модуля комплексной обработки событий запускаются с использованием их точек входа, как отдельные, слабо связанные компоненты микросервисной архитектуры.

Компоненты модуля и взаимодействия между ними

Модуль комплексной обработки событий включает в состав программные компоненты, описанные в следующей таблице.

Таблица 19 – Компоненты Модуля комплексной обработки событий

	Наименование компонента	Описание модуля
	GEOKASKAD Event Processor. Core	Основная серверная составляющая Модуля комплексной обработки событий, отвечающая за получения данных о движении целей из Модуля выходных интерфейсов, через очередь сообщений Apache Kafka, выявление событий на основе введенных правил и передачи сформированных сообщений в БД и очередь. Для программирования правил комплексной обработки событий Модуль предоставляет программные интерфейсы, работающие по технологии REST API.
	GEOKASKAD Event Processor. In-memory cache.	Система краткосрочного хранения информации о движении целей в оперативной памяти, предназначенная для быстрого доступа к этим данными и построенная на основе открытого программного обеспечения Tile38. Использует геопространственные индексы для оптимизации работы с данными, имеющими географическую привязку, в реальном времени, или времени, приближенном к реальному.
	GEOKASKAD Event Processor. Database.	СУБД, основанная на открытом ПО PostgreSQL и, предназначенная для долгосрочного и среднесрочного хранения информации Модуля комплексной обработки событий в составе платформы GEOKASKAD Marine.

	GEOKASKAD Event Processor. Database metrices exporter.	Средство для экспорта метрик, связанных с работой СУБД PostgreSQL в открытое ПО Prometheus в составе Модуля логирования и контроля метрик производительности работы системы.

Компоненты Модуля комплексной обработки событий, а также их взаимодействие в составе геоинформационной интегрирующей платформы в целом, показаны на иллюстрации в Приложении А1.

Связи модуля с другими компонентами платформы

Модуль комплексной обработки событий непосредственно взаимодействует со следующими модулями платформы реального времени GEOKASKAD Marine:

модуль выходных интерфейсов;

модуль логирования и контроля метрик производительности работы системы.

Данные, поступающие в платформу посредством Модуля входных интерфейсов и, проходящие предобработку при помощи Модуля предварительной обработки и фильтрации данных, затем поступают в модуль выходных интерфейсов, откуда и попадают на вход Модуля комплексной обработки событий посредством очереди сообщений.

Управление логами и метриками, в том числе, Модуля комплексной обработки событий, осуществляется Модулем логирования и контроля метрик производительности работы системы.

Общая схема взаимодействия между компонентами платформы GEOKASKAD Marine представлена на иллюстрации в Приложении А1.

Программное обеспечение

В данном подразделе приводятся основные результаты разработки и тестирования программного обеспечения, входящего в состав Модуля комплексной обработки событий. Более подробная информация о разработанном программном обеспечении и его тестировании, представлены в [**18**] и [**19**].

Обоснование выбора языка программирования

При выборе языка программирования для Модуля комплексной обработки событий принимались во внимание следующие требования к языку программирования.

Кроссплатформенность. Модуль комплексной обработки событий может устанавливаться, как на собственных вычислительных мощностях, так и на мощностях заказчика, выбор операционной системы которым, зачастую обусловлен другими используемыми им программными продуктами. В связи с этим, наличие кроссплатформенности значительно повышает конкурентоспособность решения;

Встроенная поддержка современных средств разграничения доступа к функциональным возможностям приложения;

Поддержка высокого уровня интеграции с иными корпоративными системами, в связи с высокой вероятностью их совместного использования в составе программных комплексов порта и иных объектов морской отрасли.

В перечень кандидатов были выбраны такие языки программирования, как C++, Python и Java. Некоторые из альтернативных вариантов были исключены ввиду ограничений в части требований кроссплатформенности.

C++, несмотря на высокую производительность, больше ориентирован на разработку более низкоуровневых системных, нежели корпоративных высокоуровневых приложений.

Python, несмотря на простоту и широкое распространение, имеет недостаточно высокую производительность, не обеспечивая требуемых характеристик в части обработки данных в реальном времени.

Сильные стороны Java проявляются при разработке корпоративных приложений ввиду развитой экосистеме разработки, что является значимым фактором при его использовании в составе серверной части Модуля комплексной обработки событий. Кроме того, при разработке приложений на языке программирования Java возможно использование программного фреймворка Spring Boot, который включает в себя такие необходимые для модуля компоненты, как

- Spring Security для организации современных механизмов разграничения доступа,
- Spring JDBC для организации работы с СУБД, в том числе, выбранной СУБД PostgreSQL.

Таким образом, для реализации модуля комплексной обработки событий была выбрана гибридная реализация с использованием языка программирования Java с использованием фреймворка Spring Boot.

Обоснование программной архитектуры

Как и в случае Модуля входных интерфейсов (см. 3.3.2), для реализации Модуля комплексной обработки событий была принята микросервисная архитектура ввиду того, что она характеризуется следующими преимуществами:

Независимость и автономность;

Возможность применения различных платформ и технологий, включая открытые и проприетарные библиотеки, разработанные на других языках программирования;

Масштабируемость, позволяющая обеспечить высокую эффективность в части отказоустойчивости и адаптируемости системы к повышению плотности входных потоков данных (за счет горизонтального масштабирования).

Управляемость и стабильность работы в следствии от возможных ошибок, возникающих в других модулях платформы, а также в общесистемных компонентах.

Разработка программного обеспечения

Технические средства

В соответствии с требованиями технического задания, Модуль комплексной обработки событий, должен запускаться на одном из четырех виртуальных, или физических серверов платформы, имеющем следующие технические характеристики:

CPU 12 ядер 2,4 ГГц и выше;

RAM 24 Гб и выше;

SSD 200 Гб и выше.

Программные средства

Для функционирования Модуля комплексной обработки событий необходимо следующее программное обеспечение:

операционная система Windows или Linux (Kernel>=5.3);

средства для контейнеризации Docker (> = 19.03) и Docker-Compose (> = 1.22);

очередь сообщений, используемая в качестве шины данных, преобразованных во внутренние форматы представления. В качестве программного обеспечения, реализующего очередь сообщений, в силу высокой производительности, распространенности, а также устойчивости и горизонтальной масштабируемости, было выбрано ПО Apache Kafka (> = 2.13);

для организации высоконадежной распределенной координации приложений, предлагается использование Apache ZooKeeper (> = 3.7).

Запуск программы

Подготовка к установке

Перед развертыванием Системы необходимо произвести следующие подготовительные действия:

1. Проверить наличие в системе необходимых пакетов, выполнив команды в терминале:

```
docker --version
docker-compose --version
```

2. Убедиться, что пользователь, под которым производится развертывание Системы, входит в системную группу docker (вывод команды должен содержать название этой группы). В противном случае все команды при работе с системой будет необходимо выполнять в режиме sudo:

```
groups
```

3. Скопировать любым удобным способом архив geokaskad_event_processing.zip с конфигурационными файлами в домашнюю директорию пользователя.

Установка ПО

Система развертывается в виде docker контейнеров. Для установки необходимо выполнить следующие действия в терминале:

1. Создать директорию для конфигурационных файлов и файла docker-compose, выполнив следующие команды (заменить USERNAME на имя пользователя в ОС):

```
sudo mkdir –p /opt/geokaskad/event_processing
sudo chown USERNAME /opt/geokaskad/event_processing
```

2. Распаковать архив и скопировать конфигурационные файлы и файл docker-compose в ранее созданную директорию:

```
cd ~
```

```
unzip ~/geokaskad_event_processing.zip
cp ~/geokaskad_event_processing/*.* /opt/geokaskad/event_processing/
```

3. При необходимости отредактировать конфигурационные файлы ep-db.env и geokaskad-ep-server.env в директории /opt/geokaskad/event_processing с учетом соответствия следующих параметров:

Таблица 20 – Параметры конфигурации подключения к БД для модуля комплексной обработки событий.

Файл ep-db.env	Файл geokaskad-ep-server.env
POSTGRES_DB	SPRING_DATASOURCE_URL
POSTGRES_USER	SPRING_DATASOURCE_USERNAME
POSTGRES_PASSWORD	SPRING_DATASOURCE_PASSWORD

4. Создать соответствующие docker volumes:

```
docker volume create ep-kafka-data
docker volume create ep-db-data
docker volume create ep-tile38-data
```

5. Аутентифицироваться в docker registry, содержащем необходимые образы (заменить USERNAME и PASSWORD на выданные):

```
docker login -u USERNAME -p PASSWORD docker.smartgeosystems.com
```

6. Получить необходимые образы, выполнив команду:

```
cd /opt/geokaskad/event_processing
docker-compose pull
```

7. Установить пароль для Tile38, указанный в конфигурационном файле geokaskad-ep-server.env (параметр SPRING_TILE38_PASSWORD), выполнив команды:

```
cd /opt/geokaskad/event_processing
docker-compose up -d ep-tile38
docker-compose exec -T ep-tile38 /usr/bin/tile38_set_pass.sh
PASSWORD_FROM_CONFIG
```

Запуск и остановка системы

Для запуска системы необходимо выполнить следующие команды в терминале:

```
cd /opt/geokaskad/event_processing
docker-compose up -d
```

Для остановки системы необходимо выполнить следующие команды в терминале:

```
cd /opt/geokaskad/event_processing
docker-compose stop
```

Проверка корректности установки

Для проверки корректности установки Системы, после ее запуска необходимо проверить:

1. В списке запущенных контейнеров (в выводе команды docker ps) присутствую контейнеры с именами event_processing_ep-db_1, event_processing_ep-kafka_1,

event_processing_ep-tile38_1, event_processing_geokaskad-ep-server_1, event_processing_geokaskad-mon-ep-db-exporter_1, а также статус контейнеров UP.

2. В браузере корректно открывается адрес (заменить SERVER_IP на ip или имя сервера, на котором развернута система): http://SERVER_IP:8191/swagger-ui/index.html

Удаление ПО

Для полного удаления необходимо выполнить следующие команды в терминале:

1. Остановить и удалить контейнеры и виртуальную сеть:

```
cd /opt/geokaskad/event_processing
docker-compose down
```

2. Удалить docker volumes:

```
docker volume rm ep-kafka-data
docker volume rm ep-db-data
docker volume rm ep-tile38-data
```

3. Удалить директорию с конфигурационными файлами и файлом docker-compose:

```
rm -r -f /opt/geokaskad/event_processing
```

Входные данные

Модуль комплексной обработки событий поддерживает следующие входные воздействия, посредством доступа к программным интерфейсам, реализованным по технологии REST API:

Запрос на получение списка всех географических областей, используемых для выявления событий, параметризация которых включает в себя геопространственную привязку;

Запросы на создание, редактирование (изменение) и удаление указанных географических областей, используемых для выявления событий, параметризация которых включает в себя геопространственную привязку;

Запросы для работы с правилами (создание, изменение, удаление) следующих типов: превышение максимального темпа изменения скорости целей, превышения максимального темпа изменения курса целей, скорость цели в пределах заданной зоны выше максимального допустимого значения, скорость цели в пределах заданной зоны ниже минимального допустимого значения, вход цели в заданную зоны, выход цели из заданной зоны, пребывание цели в пределах заданной зоны, опасное сближение целей, начало движения неподвижной цели, сближение с заданной точкой;

Запрос на получение полного перечня введенных правил, на основании которых происходит дедектирование событий.

Выходные данные

Модуль комплексной обработки событий поддерживает следующие выходные данные и реакции:

Список всех географических областей, используемых для выявления событий, параметризация которых включает в себя геопространственную привязку в формате JSON;

Сообщения о статусе выполнения запросов на создание, редактирование (изменение) и удаление указанных географических областей, используемых для выявления событий, параметризация которых включает в себя геопространственную привязку, а также ответы методов REST API в формате JSON;

Сообщения о статусе выполнения запросов на работе с правилами (создание, изменение, удаление) следующих типов: превышение максимального темпа изменения скорости целей, превышения максимального темпа изменения курса целей, скорость цели в пределах заданной зоны выше максимального допустимого значения, скорость цели в пределах заданной зоны ниже минимального допустимого значения, вход цели в заданную зоны, выход цели из заданной зоны, пребывание цели в пределах заданной зоны, опасное сближение целей, начало движения неподвижной цели, сближение с заданной точкой, а также ответы методов REST API в формате JSON;

Сообщение о статусе выполнения запроса на получение полного перечня введенных правил, на основании которых происходит дедектирование событий, а также ответ методов REST API в формате JSON;

Сообщения по результатам событий, формируемых на основе правил, помещаемые в очередь сообщений на базе Apache Kafka и имеющие следующие модели данных.

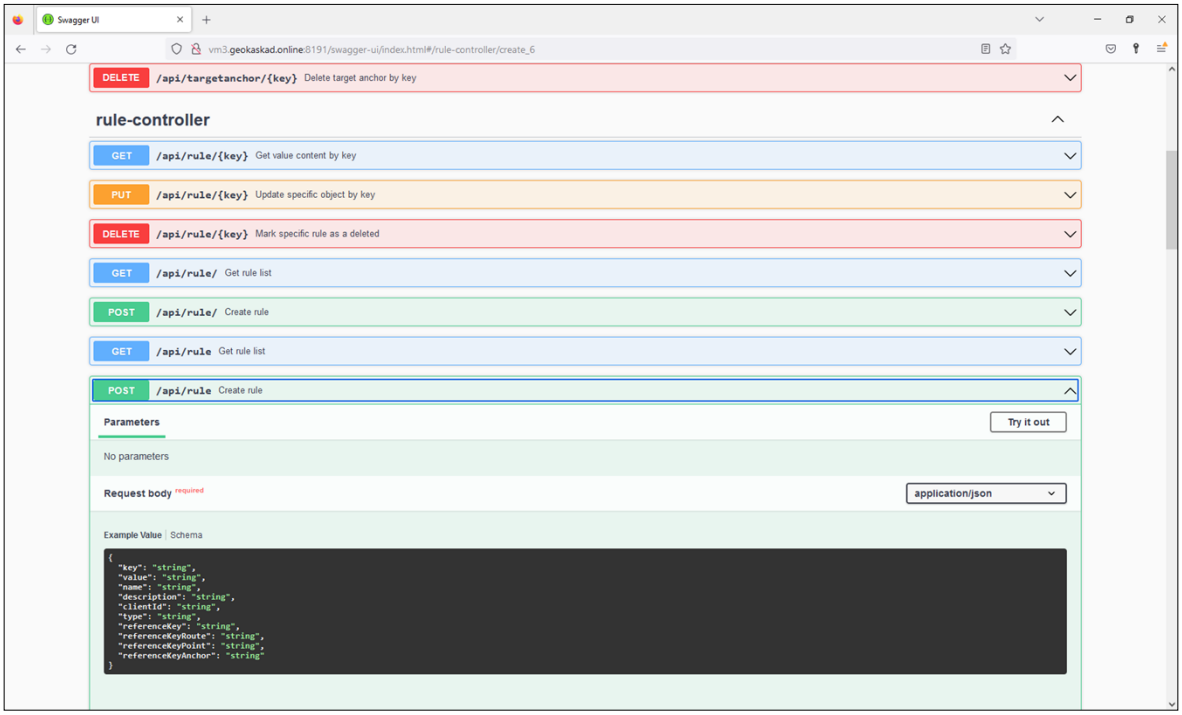
Таблица 21 – Модели данных области (зоны), используемой Модулем комплексной обработки при обработке событий.

Key	String		Ключ зоны
Name	String		Название зоны
description	String		Описание зоны
Type	String	Enum: [UNDEFINED, VESSEL_TRAFFIC_CONTROL, RESTRICTION, RESPONSIBILITY, ARRANGEMENT, NAVIGATIONAL_WARNING])	Тип зоны, заполняется из ограниченного списка значений
objectType	String	Enum: [REPORT_LINE, POLYGON, CIRCLE, RECTANGLE, POINT, NOTES])	Тип объекта, заполняется из ограниченного списка значений
clientId	String		Идентификатор системы клиента, отправившего запрос на создание зоны
Wkt	String		Геометрия зоны в формате WKT
Rules	RuleDto []	Список правил (RuleDto)	Список включенных правил к создаваемой зоне

Таблица 22 – Модели данных правил обработки событий, используемых Модулем комплексной обработки.

Key	String		Ключ правила
Value	String	Json формат	Значение определяющее правило
Name	String		Название правила
description	String		Описание правила
clientId	String		Идентификатор системы клиента, отправившего запрос на создание зоны
type	String	Enum: [TRACKING_ON, TRACKING_OFF, MAX_SPEED_CHANGE, MAX_COURSE_CHANGE, MAX_SPEED, MIN_SPEED, ENTERING_ZONE, LEAVING_ZONE, WITHIN_ZONE, DOMAIN_BREACH, CPA_TCPA, ANCHOR_DRAG, GROUND_ALARM, GROUNDING_ON_ROUTE, APPROACH_WP, APPROACH_WP_ROUTE, OFF_XTE, COURSE_DEVIATION, ENTER_ROUTE, DNG_DTE, EXIT_ROUTE, CROSSING_LINE, START_MOTION]	Тип правила
referenceKey	String		Ключ зоны, которому относится правило

Рисунок 6. Swagger-интерфейс модуля комплексной обработки событий.



На рис. 6 Swagger-интерфейс модуля комплексной обработки событий, облегчающий интеграцию сторонних и смежных систем с Модулем комплексной обработки событий геоинформационной интегрирующей платформы GEOKASKAD Marine.

РАЗРАБОТКА АРХИТЕКТУРЫ И КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МОДУЛЯ ГЕОГРАФИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ

Назначение и краткое описание

Модуль географических информационных систем (ГИС), обеспечивает представление потоковых данных, поступивших в геоинформационную интегрирующую платформу и публикуемых для сторонних и смежных систем в соответствии со стандартами ГИС, а также, в соответствии со стандартами S-10x.

Публикация данных происходит посредством ГИС-сервера GEOKASKAD Maritime GIS, разработанного в рамках внебюджетного софинансирования НИОКР со стороны ООО «СмартГеосистемс».

При этом, Модуль географических информационных систем (ГИС), в рамках платформы, предназначен для обеспечения представления потоковых данных системы в соответствии со стандартами ГИС, а также, в соответствии со стандартами S-10x, включая:

Чтение данных о движении целей АИС и РЛС, сохраняемых ПХД в реляционную базу данных на основе открытой СУБД PostgreSQL с расширением PostGIS в реальном времени, или времени, приближенном к реальному в целях формирования текущего среза данных о движении судов;

Преобразование данных о текущей динамике целей АИС и РЛС, полученных из подсистемы хранения данных в формат, совместимый с протоколами OGC, и публиковать их в соответствии с протоколом WMS;

Преобразование данных о текущей динамике целей АИС и РЛС и их публикация в соответствии со стандартами S-10x при помощи моделей, предоставляемых Платформой.

При этом, преобразованные Модулем географических информационных систем (ГИС) данные публикуются сервером GEOKASKAD Maritime GIS.

На данном этапе, среди всего перечня перспективных стандартов серии S-10x является стандарт обмена динамическими маршрутами RTZ v.1.2 и стандарт обмена навигационными оповещениями S-124. Остальные стандарты находятся на стадии доработки и согласования специализированными экспертными группами Международной морской организации.

При формировании описаний стандартов серии S-10x Международной морской организацией принят подход, основанный на разработке XSD-схем для каждого из стандартов. Основываясь на этом подходе, при разработке Модуля ГИС, использован подход динамического генерирования моделей данных на основе предоставленных и подключенных к платформе GEOKASKAD Marine XSD-схем каждого из используемых стандартов.

Модуль ГИС функционирует в составе Морской геоинформационной интегрирующей платформе реального времени GEOKASKAD Marine, разрабатываемой в рамках Договора (Соглашения) №778ГРНТИС5/72421 о предоставлении гранта на проведение научно-исследовательских и опытно-конструкторских работ.

Москва, 2023

Обоснование логической структуры

Обоснование используемых методов и алгоритмов

В соответствии с подходами Международной морской организации и рабочих групп, ответственных за разработку стандартов серии S-10x, описание утвержденных стандартов в виде XSD-схем.

Для разработки Модуля принято решение об использовании Java Architecture for XML Binding (JAXB), который позволяет Java разработчикам ставить в соответствие Java классы и XML представления. JAXB предоставляет две основные возможности: маршаллирование Java объектов в XML и обратно, то есть демаршализация из XML-структуры обратно в Java объект.

Иными словами, JAXB позволяет хранить и извлекать данные в памяти в любом XML-формате, без необходимости выполнения определённого набора процедур загрузки и сохранения XML с использованием утвержденных XSD-схем.

JAXB особенно полезен, когда спецификация является сложной и меняющейся, что характерно для стандартов серии S-10x, находящихся в стадии разработки и развития. При изменении XSD-схемы XML определений, для синхронизации их с определениями Java может потребоваться много времени, а также могут быть допущены ошибки и искажения.

Автоматизация процесса при помощи JAXB, позволяет избавить Модуль ГИС от ошибок такого рода.

Компоненты модуля и взаимодействия между ними

Модуль ГИС является самостоятельным модулем в составе геоинформационной интегрирующей платформы.

Модуля ГИС, а также его взаимодействие в составе геоинформационной интегрирующей платформы в целом, показаны на иллюстрации в Приложении А1.

Связи модуля с другими компонентами платформы

Модуль ГИС непосредственно взаимодействует со следующими компонентами геоинформационной интегрирующей платформы GEOKASKAD Marine:

Модуль выходных интерфейсов, обеспечивающий предоставление гепривязанных данных в составе платформы сторонним и смежным системам, а также, подсистемам платформы;

Подсистема хранения данных;

а также ГИС сервер в составе платформы, GEOKASKAD Maritime GIS.

Общая схема взаимодействия между компонентами платформы GEOKASKAD Marine представлена на иллюстрации в Приложении А1.

Программное обеспечение

В данном подразделе приводятся основные результаты разработки и тестирования программного обеспечения, входящего в состав Модуля ГИС. Более подробная

информация о разработанном программном обеспечении и его тестировании, представлены в [**20**] и [**21**].

Обоснование выбора языка программирования

При выборе языка программирования для Модуля ГИС, принимались во внимание следующие требования к языку программирования.

Кроссплатформенность. Модуль ГИС может устанавливаться, как на собственных вычислительных мощностях, так и на мощностях заказчика, выбор операционной системы которым, зачастую обусловлен другими используемыми им программными продуктами. В связи с этим, наличие кроссплатформенности значительно повышает конкурентоспособность решения;

Встроенная поддержка современных средств разграничения доступа к функциональным возможностям приложения;

Поддержка высокого уровня интеграции с иными корпоративными системами, в связи с высокой вероятностью их совместного использования в составе программных комплексов порта и иных объектов морской отрасли.

В перечень кандидатов были выбраны такие языки программирования, как C++, Python и Java. Некоторые из альтернативных вариантов были исключены ввиду ограничений в части требований кроссплатформенности.

C++, несмотря на высокую производительность, больше ориентирован на разработку более низкоуровневых системных, нежели корпоративных высокоуровневых приложений.

Python, несмотря на простоту и широкое распространение, имеет недостаточно высокую производительность, не обеспечивая требуемых характеристик в части обработки данных в реальном времени.

Сильные стороны Java проявляются при разработке корпоративных приложений ввиду развитой экосистеме разработки, что является значимым фактором при его использовании в составе Модуля ГИС.

Дополнительным фактором в пользу разработки Модуля ГИС на языке программирования Java, является описанная выше библиотека JAXB, обеспечивающая эффективные маршаллирование и демаршализацию данных, с использованием утвержденных XSD-схем.

Таким образом, для реализации Модуля ГИС, было принято решение об использовании языка программирования Java.

Обоснование программной архитектуры

Как и в случае Модуля входных интерфейсов (см. 3.3.2), для реализации Модуля ГИС была принята микросервисная архитектура ввиду того, что она характеризуется следующими преимуществами:

Независимость и автономность;

Возможность применения различных платформ и технологий, включая открытые и проприетарные библиотеки, разработанные на других языках программирования;

Масштабируемость, позволяющая обеспечить высокую эффективность в части отказоустойчивости и адаптируемости системы к повышению плотности входных потоков данных (за счет горизонтального масштабирования).

Управляемость и стабильность работы в следствии от возможных ошибок, возникающих в других модулях платформы, а также в общесистемных компонентах.

В случае Модуля ГИС, являющегося отдельным модулем без сложной внутренней структуры, он сам выступает в роли отдельного микросервиса в составе платформы.

Разработка программного обеспечения

Технические средства

В соответствии с требованиями технического задания, Модуль ГИС, должен запускаться на одном из четырех виртуальных, или физических серверов платформы, имеющем следующие технические характеристики:

CPU 12 ядер 2,4 ГГц и выше;

RAM 24 Гб и выше;

SSD 200 Гб и выше.

Программные средства

Для функционирования Модуля ГИС необходимо следующее программное обеспечение:

операционная система Windows или Linux (Kernel>=5.3);

средства для контейнеризации Docker (> = 19.03) и Docker-Compose (> = 1.22);

очередь сообщений, используемая в качестве шины данных, преобразованных во внутренние форматы представления. В качестве программного обеспечения, реализующего очередь сообщений, в силу высокой производительности, распространенности, а также устойчивости и горизонтальной масштабируемости, было выбрано ПО Apache Kafka (> = 2.13);

для организации высоконадежной распределенной координации приложений, предлагается использование Apache ZooKeeper (> = 3.7).

Запуск программы

Запуск Модуля ГИС осуществляется путем динамического подключения модуля к ГИС-серверу GEOKASKAD Maritime GIS в составе платформы.

Входные данные

На данном этапе развития и согласования форматов семейства S-100x, Модуль ГИС получает на вход следующие данные:

Данные динамических маршрутов в формате RTZ версии 1.2, модель поддерживается МЭК (IEC 61174:2015);

Данные навигационных оповещений в формате S-124.

В следующей таблице представлены описания основных классов, входящих в состав модели динамических маршрутов RTZ.

Таблица 23 – Описания основных классов, входящих в состав модели RTZ.

	Класс	Описание
	Route	Основной класс модели данных RTZ, предназначенный для представления динамического маршрута в целом.
	RouteInfo	Класс, предназначенный для представления метаданных маршрута (наименование, автор, и т. п.) в рамках модели данных RTZ.
	Waypoints	Коллекция путевых точек динамического маршрута, а также методы для работы с ней.
	Waypoint	Экземпляр путевой точки, задающий ее основные характеристики, включая геопространственную привязку.
	Schedules	Коллекция расписаний планируемого движения по данному динамическому маршруту, а также методы для работы с ней.
	Schedule и ScheduleElement	Экземпляр расписания, задающий ее основные характеристики, включая ожидаемый график движения судна по маршруту.
	Leg	Сегмент динамического маршрута, соединяющий последовательные путевые точки.

В следующей таблице представлены описания основных классов, входящих в состав модели данных S-124.

Таблица 24 – Описания основных классов, входящих в состав модели S-124.

	Класс	Описание
	WarningType	Перечислимый тип, задающий один из следующих типов оповещений: – LOCAL_NAVIGATIONAL_WARNING – локальные оповещения, связанные с внутренними водами в пределах некоторой юрисдикции или порта; – COASTAL_NAVIGATIONAL_WARNING – береговые оповещения; – SUB_AREA_NAVIGATIONAL_WARNING – навигационные оповещения, связанные с подрайонами некоторых юрисдикций; – NAVAREA – навигационное оповещение; – NAVAREA_NO_WARNING – сообщение, уведомляющее об отсутствии оповещений в заданной области навигации; – SUB_AREA_NO_WARNING – сообщение, уведомляющее об отсутствии оповещений в заданном подрайоне некоторой юрисдикции; – COASTAL_NO_WARNING – признак отсутствия береговых оповещений; – LOCAL_NO_WARNING – признак отсутствия локальных оповещений в пределах некоторой юрисдикции, или порта;

		– NAVAREA_IN_FORCE_BULLETIN – список идентификаторов навигационных оповещений, остающихся активными; – SUB_AREA_IN_FORCE_BULLETIN – список идентификаторов оповещений по подрайону, остающихся активными; – COASTAL_IN_FORCE_BULLETIN – список идентификаторов береговых оповещений, остающихся активными; – LOCAL_IN_FORCE_BULLETIN – список идентификаторов локальных оповещений, остающихся активными;
	DatasetType	Класс обеспечивает представление информации, связанной с типом набора данных в формате S-124. Поддерживаются следующие типы наборов данных: – Point – точка; – MultiPoint – мульти-точка; – Curve – кривая; – CompositeCurve – кривая, состоящая из нескольких независимых сегментов; – OrientableCurve – кривая с задаваемым параметром ориентации; – Surface – поверхность; – Polygon – полигональная область.
	WarningInformationType	Обеспечивает представление текстовой части информации, относящейся к навигационному предупреждению, и включает в себя заголовок, основной текст, а также код языка навигационного предупреждения.
	PointOrSurface, CurveOrSurface, PointCurveSurface, GM_Point, GM_Curve и GM_Surface	Представление точки, кривой, или поверхности в навигационном предупреждении в соответствии со стандартом S-124 в формате GML.
	TextPlacementType	Задает способ отображения текстовой информации на области навигационного предупреждения (точка вывода, выравнивание и масштаб, на котором отображается текстовая информация).

Выходные данные

Модуль ГИС, путем использования настраиваемых при помощи XSD-схем процессов маршаллинга и демаршализации сообщений, обеспечивает симметричность в части входных и выходных данных.

Следовательно, при помощи модуля ГИС возможен, как импорт данных, поступивших на вход платформы, так и преобразование данных платформы для их

отправки в соответствии со стандартами сторонним и смежным системам, а также конечным пользователям.

Москва, 2023

РАЗРАБОТКА ПРОГРАММ И МЕТОДИК ТЕСТИРОВАНИЯ ПРОГРАММНЫХ МОДУЛЕЙ ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ

В соответствии с требованиями технического задания и календарным планом Договора, была выполнена разработка программ и методик тестирования программных модулей геоинформационной интегрирующей платформы.

Утвержденные программы и методики испытания (тестирования) программных модулей геоинформационной интегрирующей платформы представлены в следующих приложениях к настоящему научно-техническому отчету:

- Приложение Р1 (обязательное). Программа и методика испытания (тестирования) Модуля входных интерфейсов;
- Приложение Р3 (обязательное). Программа и методика испытания (тестирования) Модуля предварительной обработки и фильтрации данных;
- Приложение Р5 (обязательное). Программа и методика испытания (тестирования) Модуля выходных интерфейсов;
- Приложение Р7 (обязательное). Программа и методика испытания (тестирования) Модуля логирования и контроля метрик производительности работы системы;
- Приложение Р9 (обязательное). Программа и методика испытания (тестирования) Модуля администрирования;
- Приложение Р11 (обязательное). Программа и методика испытания (тестирования) Подсистемы хранения данных.
- Приложение Р13. Программа и методика испытания (тестирования) Модуля комплексной обработки событий;
- Приложение Р15. Программа и методика испытания (тестирования) Модуля географических информационных систем.

В разделе 12 представлена информация о результатах тестирования программных модулей в соответствии с разработанными программами и методиками.

ПРОВЕДЕНИЕ ТЕСТИРОВАНИЯ РАЗРАБОТАННЫХ ПРОГРАММНЫХ МОДУЛЕЙ ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ

В соответствии с требованиями технического задания и календарным планом этапа 1 Договора, было произведено тестирование разработанных программных модулей геоинформационной интегрирующей платформы в соответствии с программами и методиками, перечисленными в разделе 8 настоящего научно-технического отчета.

По результатам проведенных испытаний были подписаны протоколы, представленные в следующих приложениях:

- Приложение Р2 (обязательное). Протокол тестовых испытаний Модуля входных интерфейсов;
- Приложение Р4 (обязательное). Протокол тестовых испытаний Модуля предварительной обработки и фильтрации данных;
- Приложение Р6 (обязательное). Протокол тестовых испытаний Модуля выходных интерфейсов;
- Приложение Р8 (обязательное). Протокол тестовых испытаний Модуля логирования и контроля метрик производительности работы системы;
- Приложение Р10 (обязательное). Протокол тестовых испытаний Модуля администрирования;
- Приложение Р12 (обязательное). Протокол тестовых испытаний Подсистемы хранения данных.
- Приложение Р16 (обязательное). Протокол тестовых испытаний Модуля комплексной обработки событий;
- Приложение Р18 (обязательное). Протокол тестовых испытаний Модуля географических информационных систем.

По итогам всех проведенных испытаний были получены заключения о соответствии модулей требованиям технического задания к Договору гранта и об отсутствии выявленных ошибок.

РАЗРАБОТКА ОПЫТНОГО ОБРАЗЦА ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ

По результатам анализа требований, предъявляемых к системам управления движением судов (СУДС), Модули геоинформационной интегрирующей платформы GEOKASKAD Marine были объединены в программы для ЭВМ в соответствии со следующей таблицей.

Таблица 25 – Вхождение модулей платформы в программы для ЭВМ в составе опытного образца геоинформационной интегрирующей платформы.

№ №	Программа для ЭВМ	Модули геоинформационной интегрирующей платформы
1	GEOKASKAF DataFlow	Модуль входных интерфейсов, Модуль предварительной обработки и фильтрации данных, Модуль выходных интерфейсов
2	GEOKASKAD Maritime GIS[16]	-
3	GEOKASKAD Event Processing	Модуль комплексной обработки событий
4	GEOKASKAD Marine	Модуль логирования и контроля метрик производительности работы системы, Подсистема хранения данных, Модуль администрирования, Модуль географических информационных систем (ГИС)

Опытный образец платформы геоинформационной интегрирующей платформы «GEOKASKAD Marine» размещен на четырех серверах со следующими характеристиками:

- CPU 12 ядер 2,4 ГГц;
- RAM 24 Гб;
- SSD 200 Гб.

Таблица 26 – Размещение компонентов платформы по виртуальным серверам стенда.

№№	Сервер	Разворачиваемые компоненты
1	Сервер №1	– GEOKASKAD DataFlow в составе модулей: – Модуль входных интерфейсов – Модель предварительной обработки и фильтрации данных – Модуль выходных интерфейсов
2	Сервер №2	– GEOKASKAD Maritime GIS

[16] Сервер ГИС «GEOKASKAD Maritime GIS» не включает в себя модули платформы, перечисленные в календарном плане Договора и разрабатывался в рамках внебюджетного софинансирования со стороны ООО «СмартГеосистемс» для обеспечения отказоустойчивой работы с данными ГИС, включая электронные навигационные карты (ЭНК) в формате S-57 и, поддержки визуализации ЭНК в соответствии со стандартами S-52 Международной морской организации.

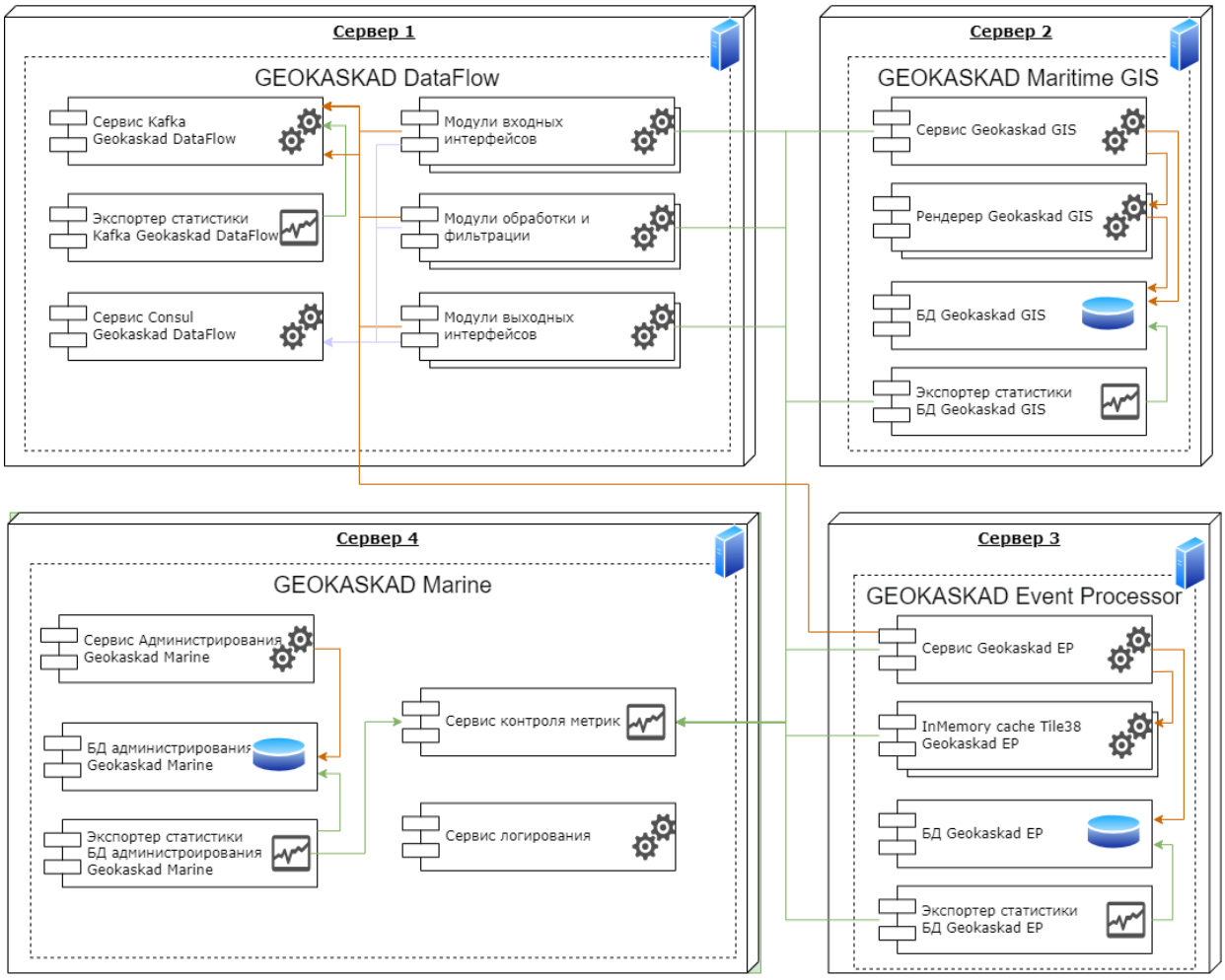
3	Сервер №3	– GEOKASKAD Event Processor в составе модуля комплексной обработки событий
4	Сервер №4	– GEOKASKAD Marine в составе модулей: – Модуль логирования и контроля метрик производительности работы системы, – Модуль администрирования, – Модуль географических информационных систем (ГИС) – Подсистема хранения данных.

Виртуальные сервера размещаются на серверах датацентра сервиса Яндекс.Облако и объединены вычислительной сетью скоростью 100 Мбит. Развертывание модулей платформы осуществлялось в соответствии с описаниями, представленными в главах, соответствующих каждому из модулей. Акт изготовления опытного образца геоинформационной интегрирующей платформы представлен в Приложение С (обязательное). Акт изготовления опытного образца

На следующем рисунке представлена обобщенную схему опытного образца с учетом распределения компонентов платформы по серверам виртуального стенда.

Рисунок 7. Обобщенная схема опытного образца с учетом установки на сервера стенда.

Схема развертывания компонентов



Москва, 2023

РАЗРАБОТКА ПРОГРАММЫ И МЕТОДИК ПРЕДВАРИТЕЛЬНЫХ ПИЛОТНЫХ ИСПЫТАНИЙ ОПЫТНОГО ОБРАЗЦА ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ

В соответствии с требованиями технического задания и календарным планом Договора, была выполнена разработка программы и методик предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы.

В соответствии с соглашением о проведении пилотных испытаний (см. Приложение Т (обязательное). Соглашение о проведении пилотных испытаний.), подписанных между ООО «СмартГеосистемс» и ЗАО «Институт телекоммуникаций», являющимся головным исполнителем по государственному Контракту от 12 ноября 2021 г. № 21411.1810190019.09.016 на выполнение ОКР «Разработка системы управления движением судов (СУДС) нового поколения с использованием технологии е-Навигации», шифр «Маяк-СУДС», было запланировано проведение испытаний и были разработаны и согласованы программы и методики испытаний.

Утвержденные программы и методики пилотных испытаний опытного образца геоинформационной интегрирующей платформы представлены в следующем приложении к настоящему научно-техническому отчету:

- Приложение Ц1 (обязательное). Программа и методики предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы;

В разделе 15 представлена информация о результатах проведения пилотных испытаний опытного образца геоинформационной интегрирующей платформы в соответствии с разработанными программами и методиками.

Москва, 2023

ПРОВЕДЕНИЕ ПРЕДВАРИТЕЛЬНЫХ ПИЛОТНЫХ ИСПЫТАНИЙ ОПЫТНОГО ОБРАЗЦА ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ

В соответствии с требованиями технического задания и календарным планом Договора, были проведены предварительные пилотные испытания в соответствии с программами и методиками, ссылка на которые приводится в разделе 14 настоящего научно-технического отчета.

По результатам проведенных испытаний были подписаны акт проведения пилотных испытаний, а также протоколы, представленные в следующем приложении:

- Приложение Ц2 (обязательное). Акт и протоколы проведения пилотных испытаний.

По итогам испытаний, представленный на пилотные испытания опытный образец геоинформационной интегрирующей платформы соответствует требованиям ТЗ к Договору (Соглашению) от 23 декабря 2021 г. №778ГРНТИС5/72421 о предоставлении гранта на проведение НИОКР по теме: «Разработка средств и сервисов информационной поддержки навигации автономных судов» в объеме проверок, предусмотренных документом «Программа и методики предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы «GEOKASKAD Marine».

Получены следующие рекомендации комиссии по пилотным испытаниям:

- Приступить к регистрации разработанного программного обеспечения в Едином реестре российских программ для электронных вычислительных машин и баз данных.
- Продолжить развитие платформы в целях обеспечения совместимости с сертифицированной операционной системой Astra Linux SE со встроенными верифицированными средствами защиты информации.

Было дано заключение, что опытный образец геоинформационной интегрирующей платформы «GEOKASKAD Marine» может быть представлен на приемочные испытания.

ДОРАБОТКА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ ПО РЕЗУЛЬТАТАМ ЕЕ ПИЛОТНЫХ ИСПЫТАНИЙ

По результатам проведения пилотных испытаний, доработка программной документации геоинформационной интегрирующей платформы не требуется.

Была произведена доработка программного обеспечения геоинформационной интегрирующей платформы GEOKASKAD Marine на виртуальных машинах облачного вычислительного сервиса «Яндекс.Облако».

По результатам проведения предварительных пилотных испытаний, замечаний выявлено не было, в связи с чем, доработка опытного образца геоинформационной интегрирующей платформы включала в себя:

- Приведение БД платформы в исходное состояние путем очистки ее таблиц;
- Удаление кэшей, используемых компонентами программного обеспечения.

Результаты доработки опытного образца геоинформационной интегрирующей платформы отражены в соответствующем Акте (см. Приложение Ч (обязательное) Акт доработки опытного образца геоинформационной интегрирующей платформы).

РАЗРАБОТКА ПРОГРАММЫ И МЕТОДИК ПРИЕМОЧНЫХ ИСПЫТАНИЙ ОПЫТНОГО ОБРАЗЦА ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ

В соответствии с требованиями технического задания и календарным планом Договора, была выполнена разработка программы и методик приемочных испытаний опытного образца геоинформационной интегрирующей платформы.

Утвержденные программы и методики приемочных испытаний опытного образца геоинформационной интегрирующей платформы представлены в следующем приложении к настоящему научно-техническому отчету:

- Приложение Ш1 (обязательное). Программа и методики приемочных испытаний опытного образца геоинформационной интегрирующей платформы;

В разделе 18 представлена информация о результатах проведения приемочных испытаний опытного образца геоинформационной интегрирующей платформы в соответствии с разработанными программами и методиками.

Москва, 2023

ПРОВЕДЕНИЕ ПРИЕМОЧНЫХ ИСПЫТАНИЙ ОПЫТНОГО ОБРАЗЦА ГЕОИНФОРМАЦИОННОЙ ИНТЕГРИРУЮЩЕЙ ПЛАТФОРМЫ

В соответствии с требованиями технического задания и календарным планом Договора, были проведены приемочные испытания в соответствии с программами и методиками, ссылка на которые приводится в разделе 17 настоящего научно-технического отчета.

По результатам проведенных испытаний были подписаны акт проведения приемочных испытаний, а также протоколы, представленные в следующем приложении:

- Приложение Ш2 (обязательное). Акт и протоколы проведения приемочных испытаний.

По итогам испытаний, представленный на приемочные испытания опытный образец геоинформационной интегрирующей платформы соответствует требованиям ТЗ к Договору (Соглашению) от 23 декабря 2021 г. №778ГРНТИС5/72421 о предоставлении гранта на проведение НИОКР по теме: «Разработка средств и сервисов информационной поддержки навигации автономных судов» в объеме проверок, предусмотренных документом «Программа и методики приемочных испытаний опытного образца геоинформационной интегрирующей платформы «GEOKASKAD Marine».

ПОДАЧА ЗАЯВКИ В ФЕДЕРАЛЬНУЮ СЛУЖБУ ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ НА ОФОРМЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

По итогам выполнения научно-исследовательских и опытно-конструкторских работ по Договору (Соглашению) от 23 декабря 2021 г. №778ГРНТИС5/72421 о предоставлении гранта на проведение НИОКР по теме: «Разработка средств и сервисов информационной поддержки навигации автономных судов», были разработаны следующие объекты интеллектуальной собственности:

- **Программа для ЭВМ «GEOKASKAD DataFlow».** Свидетельство о государственной регистрации № 2022681824. Заявка № 2022681464. Дата поступления 14 ноября 2022 г. Дата государственной регистрации в Реестре программ для ЭВМ 16 ноября 2022 г.

  Краткая аннотация к программе: Программа предназначена для сбора, интеграции, предварительной обработки, фильтрации и предоставления данных РЛС, АИС/АСОД, видеонаблюдения, метео и иных потоковых данных и сервисов морской отрасли, а также геопривязанных данных. Программа может использоваться в составе СУДС как интеграционная шина для унификации подключения, предварительной обработки и фильтрации данных морской отрасли в соответствии с отраслевыми стандартами. Функциональные возможности: при помощи модуля входных интерфейсов позволяет подключить источники данных; используя модуль предварительной обработки и фильтрации, позволяет очистить, а также отфильтровать данные по атрибутивным и пространственно-временным критериям; модуль выходных интерфейсов программы обеспечивает предоставление результирующих данных пользователям и сторонним/смежным системам. Программа «GEOKASKAD DataFlow», входящая в состав геоинформационной интегрирующей платформы «GEOKAKSAD Marine», включает в себя следующие модули, разработанные в ходе выполнения НИОКР:
  - Модуль входных интерфейсов;
  - Модуль предварительной обработки и фильтрации данных;
  - Модуль выходных интерфейсов.
- **Программа для ЭВМ «GEOKASKAD Maritime GIS».** Свидетельство о государственной регистрации № 2022681557. Заявка № 2022681545. Дата поступления 15 ноября 2022 г. Дата государственной регистрации в Реестре программ для ЭВМ 15 ноября 2022 г.

  Краткая аннотация к программе: Программа предназначена для работы с геоданными, имеющими преимущественное применение в сфере морской отрасли, включая электронные навигационные карты, векторные и растровые геопривязанные слои, включая данные дистанционного зондирования Земли (ДЗЗ) а также информацию, предоставляемую сторонними и смежными системами в форматах открытого геопространственного консорциума (OGC), а также универсальной гидрографической модели данных S-100. Программа может использоваться для информационного обеспечения систем, применяющих электронные

навигационные карты, данные ГИС, ДЗЗ, и др. и требующих, в том числе, визуального отображения географически привязанного материала. Функциональные возможности программы включают в себя импорт геоданных различного формата, управление этими данными, а также их публикацию и предоставление конечным пользователям, а также сторонним и смежным системам. Программа «GEOKASKAD Maritime GIS», входящая в состав геоинформационной интегрирующей платформы «GEOKAKSAD Marine» и, выполняющая функции ГИС-сервера в ее составе, взаимодействует с Модулем географических информационных систем (ГИС), обеспечивающим представление потоковых данных в соответствии со стандартами ГИС, а также, в соответствии со стандартами S-10x, разработанным в ходе выполнения НИОКР.

- **Программа для ЭВМ «GEOKASKAD Event Processing».** Свидетельство о государственной регистрации № 2022685454. Заявка № 2022685304. Дата поступления 19 декабря 2022 г. Дата государственной регистрации в Реестре программ для ЭВМ 23 декабря 2022 г.

  Краткая аннотация к программе: Программа предназначена для выполнения конфигурируемой обработки событий, связанных с движением судов, включая их скорости, координаты, направление движения, взаимное расположение, а также их расположение относительно тех или иных объектов инфраструктуры, и генерирования тревог и оповещений на основе этой информации, в реальном времени, или времени, приближенном к реальному. Программа может использоваться в сфере морской отрасли для формирования навигационных тревог и оповещений, а также для автоматизированного контроля судоходства, в том числе, в составе систем управления движением судов (СУДС). Функциональные возможности включают в себя программирование правил детектирования событий следующего типа: вход в заданную зону и выход из нее, нарушение скоростного режима, значительные изменения скорости и направления движения судов, опасные сближения судов, нарушения маршрутного движения. Программа «GEOKASKAD Event Processing», входящая в состав геоинформационной интегрирующей платформы «GEOKAKSAD Marine», включает в себя Модуль комплексной обработки событий, разработанный в ходе выполнения НИОКР.
- **Программа для ЭВМ «GEOKASKAD Marine».** Свидетельство о государственной регистрации № 2022685968. Заявка № 2022686316. Дата поступления 30 декабря 2022 г. Дата государственной регистрации в Реестре программ для ЭВМ 30 декабря 2022 г.

  Краткая аннотация к программе: Программа предназначена для управления контейнеризированными модулями геоинформационной системы (платформы) реального времени, выполняющей интеграцию гетерогенных сервисов морской отрасли в целях информационного обеспечения судовых и береговых служб. Платформа предоставляет для входящих в нее модулей входных/выходных интерфейсов, предварительной обработки и фильтрации данных, сложной обработки событий, а также

модулей ГИС, административную консоль, подсистему метрик и логов, а также подсистему хранения данных. Программа может использоваться в сфере морской отрасли в составе таких систем, как СУДС, логистические системы, системы контроля судозаходов и др. Функциональные возможности включают в себя управление модулями платформы при помощи веб-консоли, регистрацию данных в подсистеме хранения данных, логирование метрик производительности работы системы, а также предоставление данных. Программа для ЭВМ «GEOKASKAD Marine» является программной имплементацией геоинформационной интегрирующей платформы в целом и обеспечивает согласованную работу всех компонентов и программ, входящих в ее состав, включая в себя следующие модули, разработанные в ходе выполнения НИОКР:

- Модуль логирования и контроля метрик производительности работы системы;
- Подсистему хранения данных;
- Модуль администрирования.

ЗАКЛЮЧЕНИЕ

Настоящий документ содержит заключительный научно-технический отчет по ходу разработки опытного платформы реального времени GEOKASKAD Marine, предназначенной для информационного обеспечения автономного судовождения данными, поступающими из различных сторонних и смежных ИТ-систем, а также оборудования.

В ходе выполнения работ были сформулированы требования, подготовлены частные технические задания, а также разработаны архитектура и программное обеспечение следующих модулей/подсистем платформы:

модуль входных интерфейсов;

модуль предварительной обработки и фильтрации данных;

модуль выходных интерфейсов;

модуль логирования и контроля метрик производительности работы системы;

подсистема хранения данных;

модуль администрирования;

модуль комплексной обработки событий;

модуль географических информационных систем.

Помимо информационного обеспечения автономного судоходства, перечисленные выше модули могут использоваться для информационного обеспечения иных систем, используемых в сфере морской отрасли, например СУДС.

В ходе выполнения работ было подписано Соглашение о проведении пилотных испытаний опытного образца геоинформационной системы (платформы) на площадке на базе СУДС Региональной системы безопасности мореплавания (РСБМ) Финского залива.

По результатам выполнения работ отмечается, что работы выполнены в срок и в соответствии с требованиями, предъявляемыми к разрабатываемому научно-техническому продукту – платформе GEOKASKAD Marine.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

При формировании настоящего отчета были использованы следующие источники:

1. «Модуль входных интерфейсов. Описание программы» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин, Н.Б. Ишанов), 2022 г.
2. «Модуль входных интерфейсов. Программа и методика испытаний» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин, Н.Б. Ишанов), 2022 г.
3. NMEA 0183 - Standard for Interfacing Marine Electronic Devices. National Marine Electronics Association – January 1, 2002.
4. EUROCONTROL Specification for Surveillance Data Exchange ASTERIX Part 9 Category 62 SDPS Track Messages. EUROCONTROL-SPEC-0149-9. 11 December 2020.
5. ONVIF™ Streaming Specification, Version 21.12, December 2021.
6. Web Map Service Implementation Specification. Open GIS Consortium Inc. Reference number of this OpenGIS® project document: OGC 01-068r3. 2002-01-16.
7. «Модуль предварительной обработки и фильтрации данных. Описание программы» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин, Н.Б. Ишанов), 2022 г.
8. «Модуль предварительной обработки и фильтрации данных. Программа и методика испытаний» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин, Н.Б. Ишанов), 2022 г.
9. «Модуль выходных интерфейсов. Описание программы» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин, Н.Б. Ишанов), 2022 г.
10. «Модуль выходных интерфейсов. Программа и методика испытаний» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин, Н.Б. Ишанов), 2022 г.
11. «Модуль логирования и контроля метрик производительности работы системы. Описание программы» / ООО «Сатис Софт», Рук. А.А. Байтин, (А.О. Армяков, Е.А. Нестеров), 2022 г.
12. «Модуль логирования и контроля метрик производительности работы системы. Программа и методика испытаний» / ООО «Сатис Софт», Рук. А.А. Байтин, (А.О. Армяков, Е.А. Нестеров), 2022 г.
13. «Модуль администрирования. Описание программы» / ООО «АЙТИ ИНТЕЛЛЕКТ», Рук. Р.Р. Шарафутдинов, (Д.В. Блинова, И.Г. Тукаев), 2022 г.
14. «Модуль администрирования. Программа и методика испытаний» / ООО «АЙТИ ИНТЕЛЛЕКТ», Рук. Р.Р. Шарафутдинов, (Д.В. Блинова, И.Г. Тукаев), 2022 г.
15. «Подсистема хранения данных. Описание программы» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин), 2022 г.
16. «Подсистема хранения данных. Программа и методика испытаний» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин), 2022 г.
17. «Промежуточный научно-технический отчет о результатах выполнения работ Этапа 1. РАЗРАБОТКА СРЕДСТВ И СЕРВИСОВ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ НАВИГАЦИИ АВТОНОМНЫХ СУДОВ» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин и др.)
18. «Модуль комплексной обработки событий. Описание программы» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин), 2022 г.

19. «Модуль комплексной обработки событий. Программа и методика испытаний» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин), 2022 г.
20. «Модуль географических информационных систем (ГИС)». Описание программы» / ООО «АЙТИ ИНТЕЛЛЕКТ», Рук. Р.Р. Шарафутдинов, 2022 г.
21. «Модуль географических информационных систем (ГИС)». Программа и методика испытаний» / ООО «АЙТИ ИНТЕЛЛЕКТ», Рук. Р.Р. Шарафутдинов, 2022 г.
22. «Геоинформационная интегрирующая платформа GEOKASKAD Marine. Описание программы» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин), 2022 г.
23. «Геоинформационная интегрирующая платформа GEOKASKAD Marine. Руководство оператора» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин), 2022 г.
24. «Геоинформационная интегрирующая платформа GEOKASKAD Marine. Руководство системного программиста» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (О.Е. Бурынин, Е.А. Никулин), 2022 г.
25. «Программа и методика предварительных пилотных испытаний опытного образца геоинформационной интегрирующей платформы GEOKASKAD Marine» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (Е.А. Никулин), 2022 г.
26. «Программа и методика приемочных испытаний опытного образца геоинформационной интегрирующей платформы GEOKASKAD Marine» / ООО «СмартГеосистемс», Рук. Т.А. Кобилов, (Е.А. Никулин), 2022 г.

Реквизиты

Грантополучатель:
Общество с ограниченной ответственностью "СмартГеосистемс" (ООО "СмартГеосистемс")

ИНН/КПП:7718922540/775101001

Юридический адрес:
108811, город Москва, км Киевское шоссе 22-й (п Московский), домовлад 4 стр. 1 корпус а, офис 53 блок 620а

Фактический адрес:
108811, город Москва, км Киевское шоссе 22-й (п Московский), домовлад 4 стр. 1 корпус а, офис 53 блок 620а

Тел: +79163350732 (моб.)

Банковские реквизиты:
Расчетный счет: 40702810238000284035 в банке ПАО СБЕРБАНК (МОСКВА), БИК 044525225, к/с 30101810400000000225

Генеральный директор

Кобилов Т. А.

Документ подписан УКЭП
Серийный номер:
01CBE19C0012AF3BB84907ECD3B108F225
Владелец: ООО "СМАРТГЕОСИСТЕМС" Кобилов Тимур Алиевич
Действие сертификата: с 16.09.2022 по 16.12.2023

Москва, 2023