

**Платформа автоматизации процессов управления
корпоративными топливными картами на базе технологий
распределенного реестра и смарт-контрактов**

Описание процессов, обеспечивающих
поддержание жизненного цикла

Листов 13

2023

СОДЕРЖАНИЕ

Содержание	2
Введение.....	3
1 Общие сведения.....	4
1.1 Назначение программы	4
1.2 Системные требования	4
2 Жизненный цикл программного продукта	5
2.1 Стадии разработки и внедрения программного продукта	5
2.1.1 Стадия инициирования.....	5
2.1.2 Стадия проектирования.....	5
2.1.3 Стадия разработки	5
2.1.4 Стадия тестирования	6
2.1.5 Стадия внедрения и сопровождения.....	7
2.1.5.1 Этап развертывания Платформы в контуре заказчика	7
2.1.5.2 Поддержка пользователей.....	7
2.1.5.3 Обновление ПО	7
2.1.5.4 Устранение неисправностей	7
3 Требования к персоналу и пользователям платформы.....	11
3.1 Требования к пользователям	11
3.2 Требования к квалификации персонала, порядку его подготовки и контроля знаний и навыков	11
4 Обращение в службу технической поддержки	13
5 Контактная информация производителя	14
5.1 Юридическая информация.....	14
5.2 Контактная информация службы технической поддержки.....	14

ВВЕДЕНИЕ

В данном документе приводится описание основных процессов, обеспечивающих поддержание жизненного цикла платформы автоматизации процессов управления корпоративными топливными картами на базе технологий распределенного реестра и смарт-контрактов 1.0.0 (далее – Платформа «CardLedger», Платформа).

В настоящем документе приведены общие сведения и системные требования, предъявляемые к Платформе, описание стадий жизненного цикла Платформы, требованию к уровню подготовки обслуживающего персонала и пользователям и порядка обращения в службу поддержки Платформы «CardLedger».

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение программы

Платформа автоматизации процессов управления корпоративными топливными картами на основе технологии распределенного реестра и смарт-контрактов предназначена для обработки, учета и хранения операций заправки по корпоративным топливным картам. Данные операций заправки записываются и хранятся в системе распределенного реестра в виде транзакций.

В качестве технологии распределенного реестра используется отечественная система распределенного реестра «InnoChain». Система распределенного реестра «InnoChain» обеспечивает исполнение смарт-контрактов и хранение данных децентрализованным способом. Страница проекта системы распределенного реестра «InnoChain»: <https://innopolis.university/innochain/>.

1.2 Системные требования

Перечень характеристик, предъявляемые к аппаратному обеспечению серверной части, приведен в таблице 1.

Таблица 1 – Минимальные характеристики аппаратного обеспечения серверной части

Компонент	Характеристика
Процессор	Количество ядер: 8 Intel не ниже 5 поколения (или схожий по характеристикам) не менее 8 потоков частота не менее 3.6 ГГц
Оперативная память	Установленный объем не менее 32 Гб
Жесткий диск	Установленный объем не менее 100 Гб

Перечень характеристик, предъявляемые к аппаратному обеспечению пользователей (рабочим местам), приведен в таблице 2Таблица .

Таблица 2 – Минимальные характеристики рабочего места

Наименование показателя	Значение
Процессор	Intel не ниже 5 поколения (или схожий по характеристикам) частота не менее 2.5 ГГц
Оперативная память	не менее 4 Гб
Свободное дисковое пространство	не менее 2 Гб
Разрешение экрана	1920×1080

2 ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ПРОДУКТА

2.1 Стадии разработки и внедрения программного продукта

Можно выделить следующие стадии разработки и внедрения Платформы «CardLedger»:

- Инициирование;
- Проектирование;
- Разработка;
- Тестирование;
- Внедрение и сопровождение.

2.1.1 Стадия инициирования

На данной стадии была сформирована концепция Платформы «CardLedger», которая определяла назначение, цели и задачи, верхнеуровневую функциональную и техническую архитектуру.

2.1.2 Стадия проектирования

На данной стадии была спроектирована техническая и функциональная архитектура Платформы «CardLedger», описаны основные процессы работы Платформы, сформирована структура базы данных, сформировано техническое задание, которое определяло основные функциональные и нефункциональные требования к Платформе и календарный план ее разработки.

2.1.3 Стадия разработки

На данной стадии осуществлялась программная разработка Платформы, которая состояла из следующих основных этапов:

- Разработка моделей смарт-контрактов на языке HOL4.
- Идентификация и формальная верификация свойств надежности моделей смарт-контрактов.
- Реализация смарт-контрактов на языке CakeML.
- Настройка и оптимизация смарт-контрактов.
- Разработка веб-приложения.
- Разработка средств интеграции с внешними информационными системами.
- Разработка подсистемы управления большими данными.
- Разработка и интеграция программного обеспечения для оптимизации процессов КТК на основе больших данных для топливного оператора.
- Разработка и интеграция программного обеспечения для оптимизации процессов КТК на основе больших данных для сетей АЗС.

– Разработка сервиса непрерывного анализа и оптимизации процессов КТК на основе обработки больших данных.

– Формирование документации на Платформу.

2.1.4 Стадия тестирования

На данной стадии проводилось тестирование Платформы, которое включало в себя следующие виды тестирования:

– Ручное тестирование — выполнение тестировщиком прохода тестового цикла вручную.

– Автоматизированное тестирование — автоматический проход тестового цикла, с последующим автоматическим уведомлением заинтересованных лиц о результатах.

– Смешанное тестирование — вариант объединения ручного и автоматизированного тестирования.

– Дымовое тестирование — простейший вид тестирования, основанный на определении успешности сборки платформы из ветви исходного кода, находящейся в разработке и проверке основных сценариев работы платформы.

– Модульное тестирование — основанный на проверке работоспособности функций, методов и свойств в условиях их нормального и ошибочного исполнения. Это тестирование проводится на уровне исходного кода каждого существующего элемента (функции).

– Интеграционное тестирование — после разработки тестов на отдельные классы необходимо проведение теста в рамках одного исполняемого процесса.

– Сквозное тестирование — оценка работоспособности компонентов платформы на уровне взаимодействия нескольких отдельных исполняемых процессов. На данном этапе тестируется функционирование клиент-серверных систем, их взаимодействие внутри и с внешними компонентами. Определяются функции макроуровня, надежности, производительности, координации.

– Нагрузочное тестирование — определение и проверка характеристик производительности платформы в заданной конфигурации оборудования и набора данных.

– Тестирование безопасности — определение ролей и проверка списка функций платформы, доступных для каждой роли. Может осуществляться на уровне интерфейса, на уровне компонента, на уровне базы данных, на уровне модуля и на сетевом уровне. Включает проверку методов шифрования данных при хранении и передаче, отказа доступа к запрещенным функциям, перехвата данных, подделки удостоверения личности, отказа в обслуживании и других атак.

Была сформирована программа и методика испытаний, которая определяла подход и порядок проведения испытаний работы Платформы.

Далее, в соответствии с программой и методикой испытаний были проведены испытания, которые подтвердили корректность работы Платформы. По результатам испытаний были оформлены акт и протокол испытаний.

2.1.5 Стадия внедрения и сопровождения

На данной стадии осуществляются следующие этапы:

- Развертывание Платформы в контуре заказчика и проведение обучения пользователей.
- Поддержка пользователей.
- Обновление ПО.
- Устранение неисправностей.

2.1.5.1 Этап развертывания Платформы в контуре заказчика

На данном этапе осуществляется развертывание Платформы на серверных мощностях заказчика, а именно разворачивание узла системы распределенного реестра и веб-приложения при необходимости (решение принимается в конкретных случаях с учетом модели предоставления услуг).

Проводится обучение пользователей и далее осуществляется запуск программного обеспечения в эксплуатацию.

2.1.5.2 Поддержка пользователей

Поддержка пользователей осуществляется в объеме, порядке и сроках, установленных договором с заказчиком. Стандартный порядок обращения в службу поддержки представлен в разделе 4 настоящего документа.

2.1.5.3 Обновление ПО

Обновление программного обеспечения осуществляется в случае необходимости устранения выявленных ошибок или в случае его модернизации с целью внедрения нового функционала и повышения эффективности.

2.1.5.4 Устранение неисправностей

Неисправности, выявленные в ходе эксплуатации Платформы, могут быть устранены системным администратором.

В случае возникновения неисправностей на Платформе при установке и настройке, системный администратор может исправить их, следуя рекомендациям, представленным в Таблице 3. Если неисправности возникли в работе графического интерфейса отображения

журнала работ, системный администратор может устранить причины, следуя инструкциям, описанным в Таблице 4. Если неисправности возникли при эксплуатации веб-приложения, системный администратор может устранить причины, следуя инструкциям, описанным в Таблице 5. Если возникли сбои в работе сети системы распределенного реестра, системный администратор может устранить причины, следуя инструкциям, описанным в Таблице 6.

При невозможности устранения неисправностей, выявленных в ходе эксплуатации Платформы, либо необходимости в доработке, заказчик направляет производителю запрос в соответствии с разделом 4 настоящего документа.

Таблица 3 – Неисправности при установке и настройке системы распределенного реестра

Проблема	Причина	Решение
При установке появляется ошибка “Command docker not found”	На сервере не установлена утилита “docker”	Необходимо повторить процедуру установки утилиты “docker”
При установке появляется ошибка “Command docker-compose not found”	На сервере не установлена утилита “docker-compose”	Необходимо повторить процедуру установки утилиты “docker-compose”
При установке появляется одна из следующих ошибок: 1. ERROR: for node_api_1 Cannot create container for service api: logger: no log driver named 'loki' is registered 2. ERROR: for node_api_1 Cannot create container for service node: logger: no log driver named 'loki' is registered	Плагин “loki” не установлен или не активирован	Необходимо повторить процедуру установки плагина “loki”. Затем необходимо запустить плагин командой в терминале: docker plugin enable loki:latest

Таблица 4 – Сообщения при эксплуатации графического интерфейса для просмотра журнала работ системы распределенного реестра

Текст сообщения	Описание	Действия
При открытии страницы графического интерфейса в вебк-браузере появляется ошибка “Страница не найдена”	Система не запущена либо не открыт доступ к серверу внешним пользователя	1. Чтобы убедиться, что система запущена необходимо найти в списке строку с названием node_grafana_1 после выполнения следующей команды на проблемном

		<p>сервере: <code>\$ docker container stats</code></p> <ol style="list-style-type: none"> Если не удалось найти строку после выполнения команды, то необходимо повторить процедуру установки системы на машине Если строку удалось найти, то необходимо проверить настройки сервера на предмет доступности сетевых портов для внешних подключений
--	--	---

Таблица 5 – Сообщения программисту при эксплуатации веб-приложения

Текст сообщения	Описание	Действия
<p>При запросе к API в ответ приходит ошибка вида: <pre>{ "jsonrpc": "2.0", "error": { "code": -32601, "message": "Method not found" }, "id": 1 }</pre></p>	<p>В запросе указано неправильное название функции в поле method</p>	<p>Необходимо сверить поле method запроса со списком поддерживаемых функций</p>
<p>При запросе к API в ответ приходит ошибка вида: <pre>{ "jsonrpc": "2.0", "error": { "code": -32000, "message": "Server error" }, "id": 1 }</pre></p>	<p>В запросе указано неправильное значение в поле params</p>	<p>Необходимо удостовериться в правильности передаваемых параметров</p>
<p>При запросе к API в ответ приходит ошибка вида: <pre>{ "jsonrpc": "2.0", "result": { "id": 1, "old_len": 33, "msg":</pre></p>	<p>Используются несовместимые компоненты системы</p>	<p>Необходимо обратиться с заявкой в техническую поддержку для получения актуальной версии системы</p>

<pre>{'Error': "Incorrect message id returned from the node"}}, 'id': 1}</pre>		<p>согласно разделу 4 настоящего документа.</p>
--	--	---

Таблица 6 – Сообщения из журнала работ при сбоях в работе Платформы

Текст сообщения	Описание	Действия
<p>В журнале работ отображается ошибка Blockstore.BlockStoreInconsistentState</p>	<p>Данные в базе данных не прошли проверку на корректность при запуске системы</p>	<ol style="list-style-type: none"> 1. Необходимо удостовериться, что база данных не была изменена вручную сотрудниками, имеющими доступ к серверу. 2. Для восстановления работы системы на отдельном сервере сети достаточно перезапустить систему распределенного реестра для синхронизации с другими машинами в сети и продолжения работы: <pre>\$ docker-compose -f docker-compose.yml -p node down \$ docker-compose -f docker-compose.yml -p node up -d</pre> 3. Если проблема повторяется, то необходимо отправить заявку в службу технической поддержке и приложить журнал работ согласно разделу 4 настоящего документа

3 ТРЕБОВАНИЯ К ПЕРСОНАЛУ И ПОЛЬЗОВАТЕЛЯМ ПЛАТФОРМЫ

Персонал Платформы состоит из следующих категорий:

- пользователи;
- обслуживающий персонал.

К обслуживающему персоналу могут относиться специалисты потенциальных заказчиков, выполняющие функции администрирования функций Платформы, СУБД, серверов, специалисты по информационной безопасности, техническому обслуживанию и др.

Численность персонала должна определяться, исходя из количества необходимых АРМ на всех уровнях управления Платформы и объемов выполняемых работ, и должна быть достаточной для функционирования Платформы.

3.1 Требования к пользователям

Пользователь должен обладать базовыми навыками работы с персональным компьютером и веб-браузером. Специальные требования к компетенциям пользователя не предъявляются.

Перед началом работы с Платформой пользователь должен пройти обучение и изучить руководство пользователя.

3.2 Требования к квалификации персонала, порядку его подготовки и контроля знаний и навыков

К обслуживающему персоналу Платформы помимо наличия базовых навыков работы на персональном компьютере предъявляются специальные квалификационные требования:

- знание принципов работы технологии распределенных реестров и смарт-контрактов;
- знание принципов построения СУБД;
- навыки работы с серверным и телекоммуникационным оборудованием;
- расширенные знания в области поддержки пользователей;
- знание основ и опыт администрирования операционных систем, серверов приложений и серверов баз данных.

Основными обязанностями специалиста по техническому обслуживанию являются:

- модернизация, настройка и мониторинг работоспособности комплекса технических средств (серверов, рабочих станций);
- конфигурирование и настройка программно-технических средств Платформы;
- диагностика типовых неисправностей;
- замена базовых узлов периферийных устройств, имеющих ограниченный ресурс;
- настройка локальной компьютерной сети и Интернета;

- контроль доступа к сетевым ресурсам;
- настройка сетевого окружения.

Обслуживающий персонал, выполняющий функции администрирования функциональных возможностей Платформы, кроме перечисленных навыков, должен:

- обладать знаниями в предметной области;
- обладать знаниями и навыками операций архивирования и восстановления данных;
- обеспечивать выполнение функций регистрации пользователей и настройки прав доступа.

Основными обязанностями специалиста информационной безопасности являются:

- установка, настройка и мониторинг работоспособности средств защиты информации;
- контроль доступа к информационным ресурсам Платформы;
- контроль доступа к сетевым ресурсам.

Специалист информационной безопасности должен обладать высоким уровнем квалификации в области обеспечения информационной безопасности и практическим опытом выполнения работ по установке, настройке и администрированию используемых на Платформе средств защиты информации.

Порядок подготовки обслуживающего персонала и пользователей Платформы должен предусматривать их обязательную подготовку перед проведением опытной эксплуатации на объектах заказчика.

4 ОБРАЩЕНИЕ В СЛУЖБУ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ

В случае выявления проблемы в работе Платформы пользователь может направить запрос в службу поддержки на электронную почту university@innopolis.ru. В запросе необходимо указать следующие данные:

- тема запроса;
- описание проблемы;
- скриншот (при наличии).

После получения запроса служба поддержки выполняет следующие действия:

1) Проводится анализ выявленной проблемы и, в случае если проблема не требует детального анализа и ответ для ее решения может быть дан сразу, пользователю оказывается техническая и/или методическая поддержка для устранения/обхода проблемы.

2) В случае если проблема не может быть разрешена сразу, проблема воспроизводится командой службы поддержки. Если выявленная проблема не является ошибкой, пользователю оказывается техническая и/или методическая поддержка для устранения/обхода проблемы.

3) В случае если проблема пользователя является ошибкой Платформы, то сотрудники команды разработки оценивают сложность и критичность ее исправления. Далее ошибка ставится в очередь исполнения к следующему релизу Платформы.

4) Непосредственное исправление ошибки, с внесением информации о её исправлении в перечень изменений в новом релизе.

5) После доставки ответа запрос считается завершенным, и находится в таком состоянии до получения подтверждения от заказчика о решении инцидента. В случае аргументированного несогласия заказчика с завершением запроса выполнение запроса продолжается (переходит к шагу 1). В случае отсутствия ответа заказчика о завершении запроса в течение 10 рабочих дней запрос считается автоматически закрытым.

5 КОНТАКТНАЯ ИНФОРМАЦИЯ ПРОИЗВОДИТЕЛЯ

5.1 Юридическая информация

Название компании: АНО ВО «Университет Иннополис»;

Юр. адрес: 420500, Российская Федерация, Республика Татарстан, город Иннополис, улица Университетская, д. 1;

ОГРН: 1121600006142

ИНН: 1655258235

5.2 Контактная информация службы технической поддержки

Связаться со специалистами службы технической поддержки можно одним из следующих способов:

– телефон: 8 (843) 203-92-53;

– e-mail: university@innopolis.ru.

Фактический адрес размещения инфраструктуры разработки: Российская Федерация, 420500, Республика Татарстан, город Иннополис, улица Университетская, д. 1.

Фактический адрес размещения разработчиков: Российская Федерация, 420500, Республика Татарстан, город Иннополис, улица Университетская, д. 1.

Фактический адрес размещения службы поддержки: Российская Федерация, 420500, Республика Татарстан, город Иннополис, улица Университетская, д. 1.

Фактический адрес размещения серверов: Российская Федерация, 420500, Республика Татарстан, город Иннополис, улица Университетская, д. 1.