ПО ИНИТ СервисТерминал (INIT ServiceTerminal)

Инструкция по установке экземпляра программного обеспечения

ООО «ИНИТ-плюс софт»

25 сентября 2026

Оглавление

Программа для ЭВМ «INIT ServiceTerminal» 4

Экземпляр для проведения экспертной проверки при включении в Единый реестр российских программ для электронных вычислительных машин и баз данных 4

1. Назначение документа 4

2. Результат установки 4

3. Границы устанавливаемого программного обеспечения 5

4. Данные, предоставляемые эксперту 5

5. Системные требования 6

5.1. Аппаратные требования 6

5.2. Программные требования 6

5.3. Сетевые требования 7

6. Состав дистрибутива 7

7. Получение и проверка дистрибутива 8

8. Подготовка операционной системы 8

8.1. Проверка параметров системы 8

8.2. Установка базовых пакетов 8

8.3. Создание служебных пользователей и каталогов 8

9. Установка зависимостей 9

9.1. Установка ASP.NET Core Runtime 10.0 9

9.2. Установка и настройка RabbitMQ 9

10. Установка компонентов 10

10.1. Распаковка дистрибутива 10

10.2. Права доступа 10

10.3. Установка сертификата доверенного УЦ 10

11. Настройка компонентов 11

11.1. Настройка `config.json` 11

11.2. Настройка Arkaim Pay 12

11.3. Настройка Vector-agent 12

12. Настройка служб systemd 13

12.1. Служба Arkaim Pay 13

12.2. Служба INIT ServiceTerminal 13

12.3. Служба Vector-agent 14

12.4. Регистрация и запуск служб 14

13. Настройка режима киоска 14

14. Проверка установленного экземпляра 15

14.1. Автоматические проверки 15
14.2. Визуальная проверка 16
14.3. Контрольный функциональный сценарий 16
15. Диагностика 16
15.1. Журналы служб 16
15.2. Типовые неисправности 17
15.3. Сбор диагностического архива 17
16. Управление службами 17
17. Обновление и восстановление 18
18. Критерии успешной установки 18
19. Контрольный лист эксперта 19
20. Сведения о технической поддержке 19
21. История изменений документа 19

Программа для ЭВМ «INIT ServiceTerminal»

Экземпляр для проведения экспертной проверки при включении в Единый реестр российских программ для электронных вычислительных машин и баз данных

Реквизит	Значение
Наименование программного обеспечения	INIT ServiceTerminal
Условное обозначение	IST
Версия программного обеспечения	4.0.2026.0921
Версия документа	1.0
Правообладатель	ООО "ИНИТ-плюс софт"
Дата документа	24 сентября 2026
Целевая платформа	Linux x86-64
Эталонная операционная система	РЕД ОС 8, x86-64

---

1. Назначение документа

Настоящий документ устанавливает порядок получения, проверки, установки, настройки и запуска экземпляра программы для ЭВМ **INIT ServiceTerminal**, предоставленного для проведения экспертной проверки в целях включения сведений о программном обеспечении в Единый реестр российских программ для электронных вычислительных машин и баз данных.

Инструкция позволяет эксперту:

1. подготовить эталонную среду на базе РЕД ОС;
2. установить зависимости и компоненты, необходимые для работы экземпляра;
3. развернуть INIT ServiceTerminal без сборки из исходного кода;
4. настроить соединения с RabbitMQ, Arkaim, АСУ АЗС, конфигуратором и системой мониторинга;
5. запустить приложение и киоск-браузер;
6. проверить работоспособность установленного экземпляра;
7. собрать диагностические сведения при возникновении ошибки.

Для выполнения основной процедуры требуются права `root` или возможность выполнения команд через `sudo`.

2. Результат установки

После успешного выполнения инструкции на машине терминала работают:

- серверное приложение `InitServiceTerminalApp`;
- пользовательский интерфейс Blazor Server, доступный локально по адресу `http://127.0.0.1:5000`;
- локальный RabbitMQ с поддержкой MQTT over WebSocket;

- контроллер периферийного оборудования Arkaim Pay;
- агент Vector для сбора эксплуатационных метрик;
- киоск-браузер, автоматически открывающий пользовательский интерфейс терминала.

Работоспособность экземпляра подтверждается успешным запуском служб, доступностью локального веб-интерфейса, подключением MQTT-клиентов к RabbitMQ и прохождением контрольного сценария из раздела 14.

3. Границы устанавливаемого программного обеспечения

Заявляемым программным обеспечением является **INIT ServiceTerminal**, размещаемый в каталоге `IST-linux`.

Компонент	**Назначение**	**Статус относительно заявляемого ПО**
`IST-linux/InitServiceTerminalApp`	Серверное приложение и пользовательский интерфейс терминала	Заявляемое ПО
`IST-linux/config.json`	Конфигурация экземпляра	Конфигурационный файл заявляемого ПО
`IST-linux/init-terminal.service`	Описание службы systemd	Эксплуатационный файл заявляемого ПО
Configurator	Централизованное создание и изменение конфигурации	Внешний программный компонент
RabbitMQ	Локальный брокер сообщений	Сторонний инфраструктурный компонент
Arkaim Pay	Контроллер периферийного оборудования	Внешний программный компонент
Vector	Сбор и передача эксплуатационной телеметрии	Сторонний инфраструктурный компонент
Chromium / Яндекс Браузер	Отображение пользовательского интерфейса	Сторонний программный компонент
.NET Runtime / ASP.NET Core Runtime	Среда выполнения	Сторонний программный компонент

Configurator, RabbitMQ, Arkaim Pay, Vector, браузер и .NET Runtime необходимы для воспроизведения полной рабочей конфигурации, но не относятся к собственным функциональным модулям INIT ServiceTerminal.

4. Данные, предоставляемые эксперту

До начала установки правообладатель предоставляет эксперту:

Данные	**Значение или способ предоставления**
Ссылка на дистрибутив	`https://owncloud.init-plus.com/index.php/s/FabagOrlQBt6bkm`

Данные	Значение или способ предоставления
Имя архива	`IST-4.0.9770.26212-linux.7z`
Контрольная сумма SHA-256	`679E40F9ECE532DC9FD03AF6704C1C88CE54EF94702FBDF78610C325C12B7DD0`
Учётные данные RabbitMQ	Передаются отдельно по защищённому каналу
Адрес и токен тестовой АСУ АЗС	Передаются отдельно по защищённому каналу
Адрес тестового конфигуратора	`config.json`
Сертификат доверенного УЦ	`INIT_NETWORK-CA.crt` в составе дистрибутива
Идентификатор тестового терминала	`test-001`
Контакт технической поддержки	`it@init-plus.com`

Секреты, пароли и токены не публикуются в настоящем документе. Они предоставляются эксперту отдельно и вносятся в конфигурацию непосредственно перед запуском.

5. Системные требования

5.1. Аппаратные требования

Параметр	Минимальное значение	Рекомендуемое значение
Архитектура процессора	x86-64	x86-64, Intel Core i5 или аналог
Количество вычислительных ядер	2	4 и более
Оперативная память	4 ГБ	8 ГБ и более
Свободное место на диске	10 ГБ	20 ГБ и более
Сетевой интерфейс	100 Мбит/с	1 Гбит/с
Разрешение экрана	в соответствии с интерфейсом терминала	штатное разрешение дисплея киоска

Для промышленной эксплуатации рекомендуется подключение через источник бесперебойного питания.

5.2. Программные требования

Компонент	Требование
Операционная система	РЕД ОС 8 x86-64; допускается Astra Linux x86-64 при установке эквивалентных пакетов
Среда выполнения	ASP.NET Core Runtime 10.0.x
Брокер сообщений	RabbitMQ с модулями `rabbitmq_mqtt`, `rabbitmq_web_mqtt` и `rabbitmq_prometheus`
Браузер	Chromium или Яндекс Браузер
Менеджер служб	systemd
Системное время	Синхронизировано средствами ОС

SDK .NET и средства компиляции на целевой машине не требуются.

5.3. Сетевые требования

Источник	Получатель	Протокол	Порт	Назначение
Киоск-браузер	INIT ServiceTerminal	HTTP / SignalR	5000	Пользовательский интерфейс
INIT ServiceTerminal	RabbitMQ	MQTT over WebSocket	15675	Команды, ответы и события периферии
Arkaim Pay	RabbitMQ	MQTT или MQTT over WebSocket	По конфигурации Arkaim	Обмен сообщениями с контроллером
INIT ServiceTerminal	АСУ АЗС	HTTP(S) / JSON	7472 или по конфигурации	Работа с заказами
INIT ServiceTerminal	Configurator	HTTP(S)	По конфигурации	Получение `config.json`
INIT ServiceTerminal	Vector	OTLP/gRPC	4317	Логи, метрики и трассировки
Vector-agent	RabbitMQ	Prometheus HTTP	15692	Получение метрик RabbitMQ
Vector-agent	Vector	Vector native	6000	Передача эксплуатационных метрик

Порт `5000` рекомендуется привязывать только к `127.0.0.1`. Остальные сетевые соединения должны быть разрешены межсетевым экраном только между фактически взаимодействующими узлами.

6. Состав дистрибутива

После распаковки архив должен иметь следующую структуру:

```
IST/
├── IST-linux/
│   ├── InitServiceTerminalApp
│   ├── config.json
│   ├── INIT_NETWORK-CA.crt
│   ├── init-terminal.service
│   ├── start_kiosk.sh
│   └── kiosk.desktop
├── Configurator/
├── Vector/
│   ├── vector-agent.toml
│   └── vector-agent.service
├── Arkaim/
│   ├── ArkaimPay
│   └── default.conf
└── checksums.sha256
```

Фактический дистрибутив должен соответствовать опубликованной структуре. Если отдельные сторонние компоненты распространяются самостоятельными архивами, рядом с основным дистрибутивом публикуется перечень архивов и их контрольных сумм.

7. Получение и проверка дистрибутива

Загрузите архив по ссылке из раздела 4 и поместите его, например, в `/tmp/ist-install`.

```
sudo install -d -m 0755 /tmp/ist-install
cd /tmp/ist-install
```

После загрузки вычислите контрольную сумму:

В следующих командах в качестве примера используется имя `IST-1.0.0-linux-x64.tar.gz`. Замените его на фактическое имя полученного архива.

```
sha256sum IST-1.0.0-linux-x64.tar.gz
```

Полученное значение должно полностью совпадать с опубликованной суммой `<SHA256_АРХИВА>`. При несовпадении установку необходимо прекратить и повторно получить дистрибутив у правообладателя.

Если в комплект включён файл `checksums.sha256`, проверку можно выполнить командой:

```
sha256sum --check checksums.sha256
```

Все строки проверки должны завершиться результатом `OK`.

8. Подготовка операционной системы

8.1. Проверка параметров системы

```
cat /etc/os-release
uname -m
hostnamectl
timedatectl status
ip address
```

Ожидается архитектура `x86_64`, корректные дата и время, настроенное имя узла и рабочее сетевое соединение.

8.2. Установка базовых пакетов

Для РЕД ОС:

```
sudo dnf install -y curl tar unzip ca-certificates jq
```

Для Astra Linux:

```
sudo apt-get update
sudo apt-get install -y curl tar unzip ca-certificates jq
```

8.3. Создание служебных пользователей и каталогов

Создайте отдельных пользователей для приложения, Arkaim и Vector:

```
sudo useradd --system --user-group --home-dir /opt/IST/IST-linux --shell /usr/sbin/nologi
n ist
sudo useradd --system --user-group --home-dir /opt/IST/Arkaim --shell /usr/sbin/nologin a
rkaim
```

```
sudo useradd --system --user-group --home-dir /var/lib/vector-agent --shell /usr/sbin/nol
ogin vector
```

Если пользователь уже существует, команда вернёт соответствующее сообщение; повторно создавать пользователя не требуется.

Создайте каталоги:

```
sudo install -d -m 0755 /opt/IST
sudo install -d -o vector -g vector -m 0750 /var/lib/vector-agent/data
sudo install -d -o root -g vector -m 0750 /etc/vector
```

9. Установка зависимостей

9.1. Установка ASP.NET Core Runtime 10.0

Сначала проверьте уже установленные среды выполнения:

```
dotnet --list-runtimes
```

Требуется строка вида:

```
Microsoft.AspNetCore.App 10.0.x [/usr/share/dotnet/shared/Microsoft.AspNetCore.App]
```

Если пакет доступен в репозитории РЕД ОС:

```
sudo dnf install -y aspnetcore-runtime-10.0
```

Для Astra Linux при наличии пакета в подключённом репозитории:

```
sudo apt-get update
sudo apt-get install -y aspnetcore-runtime-10.0
```

Если пакет отсутствует, установите runtime из официального дистрибутива .NET:

```
curl -fsSL https://dot.net/v1/dotnet-install.sh -o /tmp/dotnet-install.sh
sudo bash /tmp/dotnet-install.sh \
  --channel 10.0 \
  --runtime aspnetcore \
  --install-dir /usr/share/dotnet
sudo ln -sfn /usr/share/dotnet/dotnet /usr/bin/dotnet
```

Повторите проверку:

```
dotnet --list-runtimes
```

9.2. Установка и настройка RabbitMQ

Установите RabbitMQ из репозитория, разрешённого для используемой редакции РЕД ОС:

```
sudo dnf install -y rabbitmq-server
sudo systemctl enable --now rabbitmq-server.service
```

Включите необходимые модули:

```
sudo rabbitmq-plugins enable rabbitmq_management
sudo rabbitmq-plugins enable rabbitmq_mqtt
```

```
sudo rabbitmq-plugins enable rabbitmq_web_mqtt
sudo rabbitmq-plugins enable rabbitmq_prometheus
sudo systemctl restart rabbitmq-server.service
```

Создайте отдельную учётную запись для обмена с терминалом и Arkaim. Вместо значений в угловых скобках используйте реквизиты, переданные эксперту:

```
sudo rabbitmqctl add_user 'MQTT_USER' 'MQTT_PASSWORD'
sudo rabbitmqctl set_permissions -p / 'MQTT_USER' '.*' '.*' '.*'
```

Не используйте стандартную учётную запись `guest` для интеграционных соединений.

Проверьте состояние брокера и модулей:

```
sudo rabbitmq-diagnostics ping
sudo rabbitmq-plugins list --enabled
sudo ss -lntp | grep -E ':(15675|15692)\b'
```

Ожидается ответ `Ping succeeded`, а также прослушивание портов `15675` и `15692`.

10. Установка компонентов

10.1. Распаковка дистрибутива

```
sudo tar -xzf /tmp/ist-install/IST-1.0.0-linux-x64.tar.gz -C /opt
```

После распаковки должен существовать файл:

```
test -x /opt/IST/IST-linux/InitServiceTerminalApp
```

Команда завершается без вывода и с кодом `0`.

10.2. Права доступа

```
sudo chown -R root:ist /opt/IST/IST-linux
sudo chmod 0750 /opt/IST/IST-linux
sudo chmod 0750 /opt/IST/IST-linux/InitServiceTerminalApp
sudo chmod 0640 /opt/IST/IST-linux/config.json

sudo chown -R root:arkaim /opt/IST/Arkaim
sudo chmod 0750 /opt/IST/Arkaim
sudo chmod 0750 /opt/IST/Arkaim/ArkaimPay
sudo chmod 0640 /opt/IST/Arkaim/default.conf
```

Пользователи `ist` и `arkaim` получают возможность читать конфигурацию и запускать соответствующие исполняемые файлы, но не изменять файлы программы.

10.3. Установка сертификата доверенного УЦ

Для РЕД ОС:

```
sudo install -m 0644 \
  /opt/IST/IST-linux/INIT_NETWORK-CA.crt \
  /etc/pki/ca-trust/source/anchors/INIT_NETWORK-CA.crt
sudo update-ca-trust extract
```

Для Astra Linux:

```
sudo install -m 0644 \
  /opt/IST/IST-linux/INIT_NETWORK-CA.crt \
  /usr/local/share/ca-certificates/INIT_NETWORK-CA.crt
sudo update-ca-certificates
```

Сертификат необходим для проверки TLS-сертификата АСУ АЗС. Отключать проверку TLS в конфигурации приложения не следует.

11. Настройка компонентов

11.1. Настройка `config.json`

Основной файл конфигурации расположен по адресу:

```
/opt/IST/IST-linux/config.json
```

Перед редактированием создайте резервную копию:

```
sudo cp /opt/IST/IST-linux/config.json /opt/IST/IST-linux/config.json.initial
```

Отредактируйте файл:

```
sudoedit /opt/IST/IST-linux/config.json
```

Подключение к конфигуратору

Раздел `terminal_config.configurator_settings`:

```
"configurator_settings": {
  "configurator_api_url": "http://CONFIGURATOR_HOST:PORT",
  "config_polling_interval_seconds": 0,
  "full_reload_required": true
}
```

Параметр	Назначение
`configurator_api_url`	Адрес сервера конфигуратора
`config_polling_interval_seconds`	Интервал опроса в секундах; `0` — проверка только при запуске
`full_reload_required`	`true` — перезапуск при изменении настроек; `false` — применение без полного перезапуска, если параметр поддерживается

Если конфигуратор не используется, сохраните локальный `config.json` с корректными параметрами. Недоступность конфигуратора при запуске не должна препятствовать использованию локальной конфигурации.

Подключение к АСУ АЗС

В разделе `terminal_config.asu_api_client_settings` укажите идентификатор терминала, адрес API и JWT-токен в соответствии с параметрами тестового стенда:

```
"asu_api_client_settings": {
  "terminal_id": "TERMINAL_ID"
}
```

Идентификатор `terminal_id` должен совпадать с тегом `resource.terminal.id` в конфигурации Vector-agent.

Подключение к RabbitMQ и Arkaim

Раздел `mqtt_settings`:

```
"mqtt_settings": {
  "client_id": "BlazorClient-TERMINAL_ID",
  "keep_alive_interval": 60,
  "username": "MQTT_USER",
  "password": "MQTT_PASSWORD",
  "topic": "apay/topic",
  "web_socket_uri": "ws://127.0.0.1:15675/ws"
}
```

Значения `username`, `password`, `topic` и параметры брокера должны совпадать с настройками RabbitMQ и Arkaim Pay. Для каждого одновременно работающего терминала рекомендуется уникальный `client_id`.

После редактирования проверьте JSON:

```
jq empty /opt/IST/IST-linux/config.json
```

Отсутствие вывода означает, что синтаксис JSON корректен.

11.2. Настройка Arkaim Pay

Отредактируйте файл:

```
/opt/IST/Arkaim/default.conf
```

В конфигурации должны быть указаны тот же RabbitMQ, учётная запись и MQTT-топик, что и в `mqtt_settings` INIT ServiceTerminal. Параметры взаимодействия Arkaim с конкретными устройствами задаются в соответствии с составом тестового стенда.

11.3. Настройка Vector-agent

Установите Vector из архива, входящего в дистрибутив, либо используйте совместимую версию, указанную в ведомости сторонних компонентов. Исполняемый файл должен быть доступен как `/usr/bin/vector`.

Проверьте версию:

```
/usr/bin/vector --version
```

Скопируйте конфигурацию:

```
sudo install -o root -g vector -m 0640 \
  /opt/IST/Vector/vector-agent.toml \
  /etc/vector/vector-agent.toml
```

В блоке `[transforms.enrich_metrics]` укажите:

```
.tags."resource.terminal.id" = "TERMINAL_ID"
.tags."resource.host.name" = "HOSTNAME"
```

`TERMINAL_ID` должен совпадать с `terminal_id` в `config.json`, а `HOSTNAME` — с результатом команды `hostname`.

Проверьте конфигурацию:

```
cd /var/lib/vector-agent
sudo -u vector /usr/bin/vector validate \
  --config-toml /etc/vector/vector-agent.toml
```

12. Настройка служб systemd

12.1. Служба Arkaim Pay

Создайте `/etc/systemd/system/arkaim-pay.service`:

```
[Unit]
Description=Arkaim Pay peripheral controller
Wants=network-online.target rabbitmq-server.service
After=network-online.target rabbitmq-server.service

[Service]
Type=simple
User=arkaim
Group=arkaim
WorkingDirectory=/opt/IST/Arkaim
ExecStart=/opt/IST/Arkaim/ArkaimPay
Restart=on-failure
RestartSec=5s
StandardOutput=journal
StandardError=journal

[Install]
WantedBy=multi-user.target
```

12.2. Служба INIT ServiceTerminal

Создайте `/etc/systemd/system/init-terminal.service`:

```
[Unit]
Description=INIT ServiceTerminal
Wants=network-online.target rabbitmq-server.service arkaim-pay.service
After=network-online.target rabbitmq-server.service arkaim-pay.service

[Service]
Type=simple
User=ist
Group=ist
WorkingDirectory=/opt/IST/IST-linux
ExecStart=/opt/IST/IST-linux/InitServiceTerminalApp
Environment=ASPNETCORE_ENVIRONMENT=Production
Environment=ASPNETCORE_URLS=http://127.0.0.1:5000
Environment=DOTNET_ROOT=/usr/share/dotnet
```

```
Restart=on-failure
RestartSec=5s
KillSignal=SIGINT
TimeoutStopSec=30s
StandardOutput=journal
StandardError=journal
SyslogIdentifier=init-terminal

[Install]
WantedBy=multi-user.target
```

12.3. Служба Vector-agent

Установите unit-файл из дистрибутива:

```
sudo install -o root -g root -m 0644 \
  /opt/IST/Vector/vector-agent.service \
  /etc/systemd/system/vector-agent.service
```

В секции `[Service]` должны использоваться:

```
User=vector
Group=vector
```

12.4. Регистрация и запуск служб

```
sudo systemctl daemon-reload
sudo systemctl enable --now rabbitmq-server.service
sudo systemctl enable --now vector-agent.service
sudo systemctl enable --now arkaim-pay.service
sudo systemctl enable --now init-terminal.service
```

Проверьте состояния:

```
sudo systemctl --no-pager --full status rabbitmq-server.service
sudo systemctl --no-pager --full status vector-agent.service
sudo systemctl --no-pager --full status arkaim-pay.service
sudo systemctl --no-pager --full status init-terminal.service
```

Для всех служб ожидается состояние `active (running)`.

13. Настройка режима киоска

Режим киоска запускается в графическом сеансе отдельного пользователя, например `kiosk`. Служебный пользователь `ist` для этого не используется.

Создайте пользователя графического сеанса в соответствии с политикой целевой ОС и установите поддерживаемый браузер. Затем разместите скрипт `/opt/IST/IST-linux/start_kiosk.sh`:

```
#!/usr/bin/env bash
set -u

IST_UI_URL="http://127.0.0.1:5000"
IST_BROWSER="yandex-browser-stable"
IST_PROFILE_DIR="/var/lib/ist-kiosk/browser-profile"
```

```
until curl --silent --fail --max-time 2 "$IST_UI_URL" >/dev/null; do
  sleep 2
done

exec "$IST_BROWSER" \
  --kiosk "$IST_UI_URL" \
  --user-data-dir="$IST_PROFILE_DIR" \
  --no-first-run \
  --disable-infobars \
  --noerrdialogs
```

Для Chromium замените значение `IST_BROWSER` на фактическое имя команды, например `chromium`.

Создайте каталог профиля и назначьте его пользователю киоска:

```
sudo install -d -o kiosk -g kiosk -m 0700 /var/lib/ist-kiosk/browser-profile
sudo chmod 0755 /opt/IST/IST-linux/start_kiosk.sh
```

Создайте файл автозапуска `/home/kiosk/.config/autostart/ist-kiosk.desktop`:

```
[Desktop Entry]
Type=Application
Name=INIT ServiceTerminal Kiosk
Exec=/opt/IST/IST-linux/start_kiosk.sh
Hidden=false
NoDisplay=false
X-GNOME-Autostart-enabled=true
```

Настройте права:

```
sudo install -d -o kiosk -g kiosk -m 0755 /home/kiosk/.config/autostart
sudo chown kiosk:kiosk /home/kiosk/.config/autostart/ist-kiosk.desktop
sudo chmod 0644 /home/kiosk/.config/autostart/ist-kiosk.desktop
```

После входа пользователя `kiosk` в графический сеанс браузер ожидает готовности приложения и открывает его в полноэкранном режиме.

14. Проверка установленного экземпляра

14.1. Автоматические проверки

Проверьте службы одной командой:

```
systemctl is-active \
  rabbitmq-server.service \
  vector-agent.service \
  arkaim-pay.service \
  init-terminal.service
```

Ожидается четыре строки `active`.

Проверьте веб-интерфейс:

```
curl --fail --silent --show-error \
  --output /dev/null \
  http://127.0.0.1:5000
```

Команда должна завершиться с кодом `0`.

Проверьте порты:

```
sudo ss -lntp | grep -E ':(5000|15675|15692)\b'
```

Проверьте соединения RabbitMQ:

```
sudo rabbitmqctl list_connections \
  name user peer_host peer_port state
```

После запуска IST и Arkaim должны присутствовать соответствующие MQTT-соединения.

14.2. Визуальная проверка

1. Войдите в графический сеанс пользователя `kiosk`.
2. Убедитесь, что браузер автоматически открыл `http://127.0.0.1:5000`.
3. Убедитесь, что интерфейс отображается без сообщения об ошибке загрузки.
4. Проверьте отображение состояния терминала и доступных ТРК.
5. Убедитесь, что отсутствует сообщение о потере связи с RabbitMQ или Arkaim.

14.3. Контрольный функциональный сценарий

Контрольный сценарий выполняется на тестовом стенде без проведения реального платежа либо с использованием предоставленного тестового платёжного контура:

1. выбрать доступную ТРК;
2. выбрать вид топлива;
3. выбрать способ оплаты, разрешённый для тестового стенда;
4. задать сумму;
5. проверить переход интерфейса к следующему этапу сценария;
6. отменить операцию до фактического списания денежных средств либо завершить тестовую операцию по инструкции стенда;
7. проверить возврат интерфейса на начальный экран;
8. убедиться, что операция и диагностические события появились в журналах.

Успешное выполнение подтверждает, что пользовательский интерфейс, прикладная логика и интеграция с компонентами тестового стенда работают согласованно.

15. Диагностика

15.1. Журналы служб

```
sudo journalctl -u init-terminal.service -n 200 --no-pager
sudo journalctl -u arkaim-pay.service -n 200 --no-pager
sudo journalctl -u rabbitmq-server.service -n 200 --no-pager
sudo journalctl -u vector-agent.service -n 200 --no-pager
```

Наблюдение в реальном времени:

```
sudo journalctl -u init-terminal.service -f
```

15.2. Типовые неисправности

Проявление	Возможная причина	Проверка и действие
`InitServiceTerminalApp` не запускается	Нет ASP.NET Core Runtime 10.0	Выполнить `dotnet --list-runtimes`
Ошибка чтения `config.json`	Ошибка JSON или недостаточные права	Выполнить `jq empty ...`; проверить `ls -l`
Браузер не открывает интерфейс	Служба не запущена или порт 5000 не слушается	Проверить `systemctl status` и `ss -lntp`
Нет соединения с RabbitMQ	Неверные реквизиты, модуль Web MQTT отключён	Проверить плагины, порт 15675 и `list_connections`
Нет связи с Arkaim	Различаются брокер, учётная запись или топик	Сопоставить `config.json` и `default.conf`
Ошибка TLS при обращении к АСУ	Не установлен УЦ или неверно системное время	Проверить сертификат и `timedatectl status`
Нет метрик RabbitMQ	Не включён `rabbitmq_prometheus` или недоступен порт 15692	Проверить плагины и `curl http://127.0.0.1:15692/metrics`
Vector-agent не запускается	Ошибка TOML или права на каталог данных	Выполнить `vector validate`; проверить владельца каталогов

15.3. Сбор диагностического архива

Для обращения в техническую поддержку сохраните журналы без файлов, содержащих пароли и токены:

```
IST_DIAG_DIR="/tmp/ist-diagnostics"
install -d -m 0700 "$IST_DIAG_DIR"

sudo journalctl -u init-terminal.service --since today \
  > "$IST_DIAG_DIR/init-terminal.log"
sudo journalctl -u arkaim-pay.service --since today \
  > "$IST_DIAG_DIR/arkaim-pay.log"
sudo journalctl -u rabbitmq-server.service --since today \
  > "$IST_DIAG_DIR/rabbitmq.log"
sudo journalctl -u vector-agent.service --since today \
  > "$IST_DIAG_DIR/vector-agent.log"

sudo tar -czf /tmp/ist-diagnostics.tar.gz -C /tmp ist-diagnostics
```

Перед передачей архива необходимо проверить, что журналы не содержат JWT-токены, пароли, персональные данные и иные сведения ограниченного доступа.

16. Управление службами

Перезапуск после изменения конфигурации:

```
sudo systemctl restart init-terminal.service
```

Остановка всех компонентов:

```
sudo systemctl stop init-terminal.service
sudo systemctl stop arkaim-pay.service
sudo systemctl stop vector-agent.service
```

Отключение автоматического запуска:

```
sudo systemctl disable init-terminal.service
sudo systemctl disable arkaim-pay.service
sudo systemctl disable vector-agent.service
```

После изменения unit-файла:

```
sudo systemctl daemon-reload
sudo systemctl restart init-terminal.service
```

17. Обновление и восстановление

Перед обновлением сохраните конфигурацию и текущий дистрибутив:

```
sudo systemctl stop init-terminal.service
sudo systemctl stop arkaim-pay.service
sudo cp -a /opt/IST /opt/IST.backup-before-update
```

После этого распакуйте новую версию в отдельный каталог, перенесите только документированные параметры конфигурации, проверьте контрольные суммы и повторите проверки из раздела 14.

Если новая версия не запускается, остановите её и восстановите каталог `/opt/IST` из `/opt/IST.backup-before-update`, после чего запустите прежние службы. Автоматическая миграция конфигурации допускается только в том случае, если она предусмотрена документацией конкретной версии.

18. Критерии успешной установки

Экземпляр считается успешно установленным, если одновременно выполняются следующие условия:

- контрольная сумма дистрибутива совпала с опубликованной;
- ASP.NET Core Runtime 10.0 обнаруживается командой `dotnet --list-runtimes`;
- службы RabbitMQ, Vector-agent, Arkaim Pay и INIT ServiceTerminal находятся в состоянии `active`;
- RabbitMQ прослушивает порты Web MQTT и Prometheus;
- локальный интерфейс отвечает по `http://127.0.0.1:5000`;
- киоск-браузер отображает интерфейс INIT ServiceTerminal;
- в RabbitMQ присутствуют подключения IST и Arkaim;
- приложение получает конфигурацию либо корректно использует локальный `config.json`;
- доступна тестовая АСУ АЗС или предусмотренный правообладателем имитатор;

- контрольный функциональный сценарий завершается без необработанной ошибки;
- журналы не содержат повторяющихся критических ошибок запуска.

19. Контрольный лист эксперта

Проверка	Результат
Получен дистрибутив указанной версии	☐
SHA-256 совпадает с опубликованным значением	☐
Системные требования выполнены	☐
Установлен ASP.NET Core Runtime 10.0	☐
RabbitMQ установлен, необходимые модули включены	☐
`config.json` заполнен и проходит проверку `jq`	☐
Сертификат доверенного УЦ установлен	☐
Службы зарегистрированы и запущены	☐
Интерфейс доступен на `127.0.0.1:5000`	☐
IST и Arkaim подключены к RabbitMQ	☐
Киоск-режим запускается	☐
Контрольный сценарий выполнен	☐
Диагностические журналы доступны	☐

20. Сведения о технической поддержке

Параметр	Значение
Организация	ООО "ИНИТ-плюс софт"
Адрес электронной почты	support@init-plus.com
Телефон	+7 800 100-46-48
Время работы	08:00 - 17:00 (МСК)

При обращении необходимо указать версию INIT ServiceTerminal, версию операционной системы, идентификатор терминала, время возникновения ошибки и приложить диагностические журналы после проверки их содержимого.

21. История изменений документа

Версия	Дата	Описание
1.0	25.09.2026	Переработана инструкция для проведения экспертной проверки: унифицирована эталонная установка, определены границы ПО, добавлены проверка дистрибутива, критерии приёмки и диагностика