Безопасность стенда Pilot

Сетевые правила, секреты, права доступа и данные, которые обрабатывают компоненты стенда.

1. Кратко

Стек работает внутри сети организации и не требует интернета в работе. Из наших сервисов люди работают в браузере только с веб-админкой PilotProAdmin; веб-интерфейсы GitLab, Grafana и Pilot-myAdmin принадлежат стороннему ПО. Ещё три наших сервиса внутренние и служебные. Остальное это продукты АСКОН и открытое ПО. Паролей в этом документе нет: они находятся в файле .env и в PASSWORDS.md комплекта.

От службы ИБ нужны решения:

№	Что решить	Где описано
1	Открыть сетевые доступы, в том числе всем рабочим местам Pilot к сборщику статистики	Разделы 4 и 5
2	Согласовать сбор статистики с логином, ФИО и именем компьютера, срок хранения 30 дней уже задан (раздел 7)	Раздел 7
3	Выбрать вид токенов разработчиков для реестра пакетов GitLab и их срок действия	Раздел 6
4	Принять заявление разработчика о недекларируемых возможностях или назначить свою проверку	Раздел 8
5	Принять решение по известным рискам: работа без TLS, доступ runner к Docker, секреты открытым текстом на диске, права Maintainer в GitLab, вход в админку только администратором Pilot	Раздел 9

2. Что входит в стек

Наши компоненты

Компонент	Где работает	Что делает	Кто пользуется
Веб-админка pilot-proadmin	Сервер	Выкладывает автоматизации, отчёты и модули в Pilot через GitLab: ручная выкладка идёт после успешного pipeline, выкладка из CI подтверждается токеном задания и ключом контура. Правит настройки серверов Pilot (раздел «Серверы»). Сравнивает контуры, откатывает версии, запускает миграцию	Разработчики, администраторы
Миграция pilot-migration	Сервер	Копирует данные между контурами. Своего интерфейса нет	Только админка

Компонент	Где работает	Что делает	Кто пользуется
Сборщик report-metrics-collector	Сервер	Принимает статистику и пишет её в базу pilot_metrics	Рабочие места, серверные модули
Сборщики automation-metrics-*	Сервер	Раз в 5 минут читают лог автоматизаций Pilot и пишут в pilot_metrics	Только сервисы
ServerHost	Внутри Pilot-Server	Загружает и обновляет серверные модули без перезапуска сервера	Pilot-Server
Pilot.ClientSdk	Клиент Pilot	Общие функции клиентских модулей и отправка статистики	Все пользователи Pilot
Reports	Клиент Pilot	Функции отчётов, статистика построения, сохранение отчётов в GitLab	Все пользователи; GitLab только у разработчиков отчётов
debug	Машина разработчика	Запускает серверный модуль под отладчиком на данных контура. Запись на сервер только с подтверждением в консоли	Разработчики серверных модулей

Стороннее ПО

Версии сторонних продуктов закреплены в docker-compose.yml тегом и контрольной суммой (digest) и не меняются сами. Исключения: RustFS закреплён только тегом (образ приходит через docker load, у загруженного образа digest нет, целостность проверяет IMAGES.txt; контейнер работает не от root, UID 10001, с no-new-privileges; данные лежат в именованном томе Docker, а не в папке хоста, и права на них настраивать не нужно), а образы наших сервисов (pilot-proadmin, pilot-migration, pilot-report-metrics, pilot-automation-metrics) собираются из исходников и называются :latest. При сборке в них записываются метки org.opencontainers.image.revision и created (коммит и время сборки), идентификатор каждого образа указан в файле IMAGES.txt комплекта. RustFS 1.0.x вышел недавно (лицензия Apache-2.0); его совместимость с Pilot-Server и поддержку вендором Pilot хранилищ, отличных от MinIO, мы не проверяли.

Продукт	Версия	Назначение
Pilot-Server, Pilot-myAdmin Web (АСКОН)	release, по digest	Сервер Pilot и его веб-администрирование
PostgreSQL	16.15	Базы Pilot и pilot_metrics

Продукт	Версия	Назначение
RustFS	1.0.5	Файловый архив Pilot (S3), поставляется в образе комплекта
GitLab CE, GitLab Runner	19.3.0	Исходники, CI, реестр пакетов, вход по OAuth
Grafana, Prometheus, Loki, Promtail	11.2.2, 2.55.1, 3.2.1, 3.2.1	Панели, метрики, логи
node-exporter, cAdvisor	1.8.2, 0.49.1	Нагрузка машин и контейнеров
nginx	1.27.5	Мост к сервисам на других машинах, только при разнесённой установке

3. Доступ в интернет

Для работы стека интернет не нужен, исходящий доступ можно полностью закрыть. Ниже указано, что происходит при установке и в работе.

Компонент	Обращения в интернет	Что нужно сделать
PilotPro Admin, сервер	Нет. Обращается только к Pilot, GitLab и миграции по адресам из настроек. Запросы к GitLab идут с тайм-аутом 15 с и без переходов на другой хост. До первой настройки серверов на странице входа можно указать адрес Pilot вручную, после настройки поле убрано	Ничего
PilotPro Admin, браузер пользователя	Нет. CodeMirror 5.65.16 лежит в самой админке (папка wwwroot/vendor/codemirror, лицензия MIT рядом), с внешних адресов скрипты и стили не загружаются. Политика CSP разрешает скрипты только с самой админки	Ничего. Редактор работает без интернета, с подсветкой синтаксиса
Миграция, сборщики статистики	Нет	Ничего
Клиентские и серверные модули	Нет. Только сборщик статистики и, у разработчиков отчётов, GitLab	Ничего
Grafana, Loki	По умолчанию отправляют статистику использования и проверяют обновления. В комплекте и в автономных compose-файлах репозитория это отключено, в Grafana отключён и Gravatar	Ничего

Компонент	Обращения в интернет	Что нужно сделать
GitLab	Параметрами compose отключены отправка статистики и Gravatar, но они действуют для новой базы: в базе, восстановленной из копии, значения берутся из копии. Проверку версий, а после восстановления и эти параметры нужно проверить вручную в Admin / Settings	См. «GitLab без интернета»
Установка	Docker скачивает образы с registry.ascon.ru, Docker Hub, gcr.io и registry.gitlab.com	Комплект без интернета содержит все образы: его готовит поставщик, тип комплекта указан во второй строке IMAGES.txt
CI и разработчики	Образы заданий и пакеты NuGet берутся из интернета	Без интернета образы лежат на runner, пакеты в реестре GitLab. Проверки автоматизаций и отчётов (ci-checks) без интернета работают только с собственным образом CHECKS_IMAGE, в котором есть python3 и git: по умолчанию задание ставит git командой apk add из интернета
Pilot-Server и другие сторонние продукты	Поведение определяет производитель	Закрыть исходящий доступ на межсетевом экране

> Настройки сервисов не заменяют сетевой контроль. Надёжнее запретить исходящий доступ машин стека на межсетевом экране: стек продолжит работать.

GitLab без интернета

GitLab CE и Runner могут работать в закрытой сети при двух условиях: образы и пакеты принесены заранее, а для проверок ci-checks задан собственный образ CHECKS_IMAGE с python3 и git. Образы приносятся один раз в комплекте. Пакеты NuGet и библиотеки DLL загружает ответственный сотрудник в реестр пакетов GitLab, а разработчики и CI получают их только оттуда. Это даёт ещё и контроль: в закрытый контур попадает только то, что загрузил ответственный. Порядок действий описан в документе «GitLab без интернета».

4. Кому какой доступ

Кто	Что делает	Куда нужен доступ	Чем входит
Пользователь Pilot	Работает в Pilot	Pilot prod 5546, сборщик 5310	Учётка Pilot
Тестировщик	Проверяет изменения	То же и Pilot test 5547, stage 5549	Учётка Pilot

Кто	Что делает	Куда нужен доступ	Чем входит
Разработчик модулей, автоматизаций, отчётов	Пишет код и отправляет на проверку	GitLab 8080 и 2224, админка 5300, Pilot test и stage	Учётка GitLab; в админку входит учёткой администратора сервера Pilot; токен read_api для пакетов
Выпускающий	Выводит изменения на stage и prod	То же и Pilot prod 5546	Учётка GitLab (роль не ниже Developer в группе, для слияния в main и release/* Maintainer) и учётка администратора сервера Pilot
Администратор Pilot	Управляет пользователями и правами Pilot	myAdmin 5601-5603	Учётка администратора Pilot
Администратор стека	Устанавливает и обслуживает сервисы	SSH или RDP на машины, служебные порты с самой машины	Учётка ОС; .env и PASSWORDS.md с правами 600 на Linux и ограниченным доступом на Windows, права DATA_ROOT на Windows закрывает администратор
Руководители, аналитики	Смотрят статистику	Grafana 3000	Учётка Grafana
Служба ИБ	Контроль и журналы	Grafana 3000, журналы в Loki: логи контейнеров, в том числе входы в админку и GitLab; журнал входов в ОС только на Linux, категория auth определяется по тексту строк и не заменяет журнал ОС	Учётка Grafana

Вход в админку. Админка принимает только учётную запись администратора сервера Pilot: логин и пароль проверяются на серверах из настроек (адрес сервера на странице входа выбрать нельзя, ручной ввод остаётся только при первом запуске, пока серверы не заданы, и требует ключ установки ADMIN_SETUP_KEY; без ключа такой вход закрыт), затем нужен вход в GitLab. Любой, кто работает в админке, имеет на сервере Pilot права администратора. Для автоматизаций админка работает от имени вошедшего администратора сервера, для отчётов и модулей от имени учётки, введённой на странице раздела, а при выкладке из CI под служебной учёткой Pilot контура.

Права внутри админки. Сохранение общих настроек, весь раздел «Миграция» (запуск, расписания, подключения, проверка подключений с сохранением) и сохранение «Серверов» доступны только сессии, вошедшей администратором Pilot контура prod (если прод не задан, администратором любого настроенного сервера, до первой настройки любым). Остальные сессии видят эти страницы только для чтения, секреты замаскированы, запись даёт ошибку 403. Любое изменение в GitLab и выкладка на серверы (в том числе все операции серверных модулей) требуют роли не ниже Developer в группе и разрешены только при подключённом GitLab и заданной группе: при ошибке GitLab или ошибке определения роли действие запрещено.

Выкладка из CI. Анонимные маршруты /api/ci/modules/deploy и /api/ci/server-modules/deploy требуют два доказательства: заголовок X-Job-Token (токен задания $CI_JOB_TOKEN, который

админка проверяет в GitLab запросом GET /api/v4/job) и заголовок X-Deploy-Key (ключ контура, выведенного из проверенной ветки: dev это test, release/* stage, main и master prod). Ветка, коммит, pipeline, проект и автор берутся из ответа GitLab, а не из полей запроса: несовпавшие поля дают 400, проект вне группы модулей этого вида 403, неверный токен 401. Ключи раздельные: MODULES_DEPLOY_KEY_TEST, _STAGE, _PROD. В GitLab ключи STAGE и PROD имеют флаги Masked и Protected, поэтому их получают только задания защищённых веток main и release/*. Старый общий ключ принимается только для test и только если ключ test не задан. Проверки библиотек (check_libraries) требуют только токен задания с той же проверкой проекта и группы.

Остаточные риски выкладки. Защита опирается на защиту веток в GitLab и на состав Maintainer. Ветка release/* принимает push от Maintainer (админка коммитит в неё версию и CHANGELOG.md), поэтому Maintainer может выложить код на стейдж, а слить в main (прод) может любой Maintainer. Ключ test защищён только флагом Masked: его видит любое задание группы. Владелец группы и администратор GitLab могут менять переменные и правила защиты. После выкладки код модуля выполняется в процессе Pilot-Server или клиента Pilot. Проекты, которые создаёт админка, получают настройку Pipelines must succeed, а репозиторий автоматизации или отчёта со своим .gitlab-ci.yml без подключения общих проверок при первой публикации получает общий ci_config_path (предупреждение пишется в журнал).

Защита от перебора и журнал. После 10 неудачных попыток (вход, ввод учёток контуров, неверные ключи и токены CI) с одного адреса за 5 минут админка отвечает 429 на 5 минут. Адрес клиента виден, если админка не за обратным прокси или включено ADMIN_TRUST_PROXY=true (включайте, только если порт админки недоступен в обход прокси). Админка пишет в журнал контейнера pilot-proadmin, который читает Promtail и который виден в Loki: входы в Pilot и GitLab, неудачные входы (без паролей), смены общих настроек (имена полей, не значения), отказы 403 и разрешённые изменения в защищённых разделах, неверные ключи и токены CI. Журнал операций (список, страница операции и её состояние) видит администратор прода, а остальные только свои операции: запущенные под их логином Pilot или их пользователем GitLab (выкладка из CI записывается на пользователя GitLab, запустившего job). Чужая операция отвечает 404. В журнале могут быть сообщения об ошибках и хвост лога задания GitLab; сам журнал хранится в data/jobs.json без шифрования.

Заголовки и cookie. Страницы админки отдаются с Content-Security-Policy (скрипты только свои с одноразовым nonce, без unsafe-inline; редактор CodeMirror отдаётся самой админкой), X-Content-Type-Options: nosniff, Referrer-Policy и X-Frame-Options. Обработчики событий вынесены из страниц, файлы /downloads отдаются только после входа. Cookie входа получает флаг Secure, когда запрос пришёл по HTTPS; за обратным прокси это работает при ADMIN_TRUST_PROXY=true или https-адресе в ADMIN_PUBLIC_URL. Допустимые значения заголовка Host по умолчанию не ограничены (AllowedHosts=*): сузить их можно переменной ADMIN_ALLOWED_HOSTS, перечислив через «;» все имена, по которым открывают админку, и host.docker.internal (по нему обращается CI модулей); запрос с другим Host получает 400.

5. Сетевые правила

По умолчанию весь трафик идёт по HTTP без TLS, см. раздел 9. Для админки и Grafana HTTPS включается необязательно: сертификат и ключ кладутся в папку certs установочной папки, установка включает HTTPS на тех же портах (HTTP на них перестаёт работать). Сертификат и ключ копируются в DATA_ROOT/tls с правами 600 для пользователя контейнера (на Windows доступ ограничен текущим пользователем, SYSTEM и администраторами). Порты указаны по умолчанию и меняются в .env.

Вход от людей и рабочих мест

№	Откуда	Куда, порт TCP	Зачем
1	Рабочие места Pilot	Pilot-Server prod, 5546	Работа в Pilot
2	Рабочие места Pilot	Сборщик, 5310: POST /api/module-events, /api/report-events	Статистика модулей и отчётов
3	Разработчики, тестировщики	Pilot-Server test 5547, stage 5549	Проверка изменений
4	Разработчики, администраторы	Админка, 5300	Выкладка, сравнение, откат, миграция
5	Разработчики; клиент Pilot разработчиков отчётов	GitLab, 8080 и 2224	Исходники, CI, пакеты, вход
6	Администраторы Pilot	myAdmin, 5601, 5602, 5603	Администрирование Pilot
7	Руководители, аналитики, ИБ	Grafana, 3000	Панели и журналы

Служебные порты

PostgreSQL 5433-5435, RustFS 9000-9001, 9010-9011 и 9020-9021, миграция 5320, Prometheus 9090, Loki 3100, node-exporter 9100, cAdvisor 9101. При установке на одну машину они слушают только 127.0.0.1 и снаружи недоступны. Автономные compose-файлы репозитория ведут себя так же: порт миграции публикуется только на 127.0.0.1 (`MIGRATION_BIND`), Prometheus на `INTERNAL_BIND` или 127.0.0.1. Внутри машины сервисы общаются через сети Docker: у каждого контура своя сеть.

> **На Linux правила ufw и firewalld не закрывают опубликованные порты Docker.** Docker добавляет свои правила раньше них. Запреты нужно задавать в цепочке DOCKER-USER или на внешнем межсетевом экране и проверять подключением с другой машины. Пример находится в документе по установке, раздел 2.

Дополнительно при установке на разные машины

Файл правил брандмауэра. Без него служебные порты при INTERNAL_BIND доступны всей сети, а у Prometheus, Loki и экспортёров нет пароля. Поэтому при заданном INTERNAL_BIND установка сама пишет в установочную папку готовый файл правил: firewall-linux.sh для Linux, firewall-windows.ps1 для Windows. Файл разрешает эти порты только адресам машин стенда из .env (PROD_HOST, TEST_HOST, STAGE_HOST, METRICS_HOST, ADMIN_HOST, GITLAB_HOST_IP) и самой машине, остальным закрывает. Порты админки, Grafana, Pilot и GitLab файл не затрагивает. Установка файл только создаёт и печатает путь к нему, сама ничего не применяет: правила применяет администратор после просмотра (проверка без применения: sh firewall-linux.sh --test, снятие: --remove). На Linux правила стоят в цепочке DOCKER-USER; их работа проверена на настоящем iptables (трафик через подмену адреса, как у опубликованных портов Docker). Файл для Windows проверен только по синтаксису PowerShell, на живой машине его действие не проверялось. В правила входят только IPv4-адреса, имена хостов пропускаются с предупреждением.

Параметр INTERNAL_BIND открывает служебные порты на IP машины. Они слушают сеть без аутентификации: откройте их только для перечисленных источников. В конце установки скрипт печатает список служебных портов машины:

Откуда	Куда, порт TCP	Зачем
Машина admin	Контуры: Pilot 5546, 5547, 5549	Работа админки и миграции
Машина admin	Контуры: PostgreSQL 5433-5435, RustFS 9000, 9010, 9020	Копирование данных миграцией
Машина admin	Контуры: SMB 445 или NFS 2049	Чтение настроек контуров. Миграция прод только читает, но раздел «Серверы» сохраняет настройки и прода, поэтому для него нужна запись
Машина admin	GitLab, 8080	Ветки, merge request, pipeline, вход; проверка токена задания при выкладке из CI
GitLab Runner	Админка, 5300	Выкладка модулей из CI: ключ контура и токен задания
Машины test, stage, metrics	Прод: PostgreSQL, 5433	База статистики pilot_metrics
Машина metrics	Все машины: 9100, 9101	Сбор нагрузки
Все машины	Машина metrics: Loki, 3100	Отправка логов
Контуры	Машина metrics: сборщик, 5310	Статистика серверных модулей

6. Учётные записи и секреты

Как входят в сервисы

Сервис	Вход
Админка	Логин и пароль администратора сервера Pilot, затем GitLab через OAuth. Запасной вариант: Personal Access Token. Своих учётных записей нет. Токены хранятся в сессии на сервере, cookie зашифрованы (ключи шифрования лежат в данных админки файлами без дополнительной защиты)
Pilot-Server, myAdmin	Учётные записи Pilot
GitLab	Учётные записи GitLab
Grafana	Свои учётные записи, саморегистрация отключена
RustFS	Root-учётка контура для обслуживания (в .env её ключи по-старому называются MINIO_USER и MINIO_PASSWORD). Pilot должен работать под учёткой pilot с доступом только к своему бакету: такую учётку нужно создать после установки

Сервис	Вход
Миграция	Ключ MIGRATION_API_KEY, один на админку и сервис, сравнивается по хешам. После 10 неверных ключей с одного адреса за 5 минут ответ 429 на 5 минут. Защищайте его как ключ к проду: в сервисе хранятся учётные данные всех контуров
Сборщик статистики	Ключ COLLECTOR_API_KEY. Если он задан, нужен и для приёма событий, и для сводок GET /api/report-events/summary и /api/module-events/summary (/health открыт). При первой установке на одну машину ключ создаётся, в остальных случаях он пуст и запросы принимаются без ключа, см. раздел 9. Размер тела запроса ограничен 10 МБ (больше отклоняется), параметр hours сводок не больше 8784
Prometheus, Loki	Без аутентификации, порты открыты только сервисам. Управление Prometheus по HTTP (--web.enable-lifecycle) выключено

OAuth-приложения GitLab

	Веб-админка	Reports
Кто входит	Пользователь админки в браузере	Разработчик отчётов из клиента Pilot
Тип	Confidential: секрет хранится на сервере	Public без секрета, PKCE
Redirect URI	http://<админка>:5300/GitLab/Callback: адрес берётся из GITLAB_OAUTH_REDIRECT_URI, иначе из ADMIN_PUBLIC_URL, иначе из адреса запроса	http://127.0.0.1:7890/callback, адрес рабочего места
Scope	api	api
Хранение токена	Сессия на сервере	Рабочее место, DPAPI: %AppData%\ASCON\Reports\gitlab

Scope api нужен, потому что приложения создают ветки, коммиты и merge request от имени пользователя.

Токены для реестра пакетов

Вариант	Права	Плюсы и минусы
Личный токен разработчика (рекомендуется)	read_api	Персональный, отзывается при увольнении, срок задаётся
Deploy Token группы	read_package_registry	Проще, только чтение пакетов, но один общий секрет и нужна ротация

Пароль токена на рабочем месте шифруется через DPAPI. Загружать пакеты должен один ответственный, в CI для чтения используется токен задания. Шаблон CI передаёт его только источникам NuGet, хост которых равен `CI_SERVER_HOST`, и записывает открытым текстом в конфигурацию NuGet внутри одноразового контейнера задания. Пакет Pilot.ClientSdk публикуют задания `nuget_*` проекта под токеном задания.

Где какие секреты

Секрет	Для чего	Где хранится
Администратор Pilot-Server	Вход в Pilot, миграция	.env; config.json миграции (DATA_ROOT/services/migration) открытым текстом
postgres, pilotuser	Базы Pilot	.env, строка подключения в settings.xml
metrics_writer, metrics_reader	База pilot_metrics: запись (сборщики статистики) и только чтение (Grafana)	.env (METRICS_WRITER_PASSWORD, METRICS_READER_PASSWORD)
RustFS root, pilot	Файловый архив	.env; ключ в settings.xml зашифрован средствами Pilot
MIGRATION_API_KEY	Админка / миграция	.env
MODULES_DEPLOY_KEY_TEST, _STAGE, _PROD, служебные учётки выкладки	CI / админка / Pilot	.env машины admin; переменные группы GitLab DEPLOY_KEY_TEST, _STAGE (Masked) и _PROD (Masked и Protected)
COLLECTOR_API_KEY	Клиенты / сборщик	.env, файлы настроек модулей
OAuth Client ID и Secret	Вход в админку	.env
GitLab root, gitlab-secrets.json	GitLab	Данные GitLab
Администратор Grafana	Grafana	.env (начальный), база Grafana
Ключи Data Protection	Шифрование cookie входа, токенов форм и паролей Pilot в памяти сессий админки	Данные админки (services/proadmin/keys), файлы ключей без шифрования. У каждой установки свои: в комплект с data ключи не попадают, при установке из data создаются заново; из бэкапа своей машины возвращаются. На Linux папка закрыта правами 700

> Комплект с папкой data содержит данные стенда и все пароли открытым текстом, в том числе файл PASSWORDS.md. Передавайте его как секрет, доступ к .env и DATA_ROOT оставьте только администратору стека. Установка ограничивает права: .env и PASSWORDS.md получают права 600 на Linux (на Windows доступ у пользователя, SYSTEM и администраторов), на Linux закрываются каталоги секретов в DATA_ROOT; права DATA_ROOT на Windows установка не меняет. При первой установке на одну машину пароли-заглушки из .env.example заменяются случайными, на нескольких машинах установка останавливается, пока заглушки не заменены. Открытым текстом лежат пароли администраторов Pilot и ключи S3 в config.json миграции (права 600 действуют только в Linux-контейнере) и пароль PostgreSQL в settings.xml (ключ S3 там зашифрован средствами Pilot); защищайте резервные копии и сетевые папки.

7. Сбор статистики и персональные данные

Статистику отправляет клиент Pilot на рабочем месте, поэтому всем нужен доступ к сборщику. Без него Pilot работает как обычно, но статистика с этого места теряется.

Источник	Как собирается	Куда пишется
Клиентские модули	События копятся в памяти и отправляются пачкой раз в 15 секунд, на диск не пишутся	metrics.module_events
Отчёты	Событие на каждое построение	metrics.report_events
Серверные модули	События загрузки и действий, пачкой раз в 15 секунд	metrics.module_events
Автоматизации	Лог сервера Pilot читается раз в 5 минут, только на test и stage. Повторное чтение того же лога дубли не создаёт, исключение: события, записанные прежней версией сборщика	metrics.automation_events

Что отправляется: модуль и его версия, действие, результат, длительность, число объектов, тип и текст ошибки, а также логин, ФИО, имя компьютера, адрес сервера Pilot и версия клиента. Данные объектов, атрибуты и содержимое документов клиентские и серверные модули не отправляют. Текст ошибки может содержать имена объектов, если их включил автор модуля; он усекается: в Reports это текст корневого исключения до 1000 символов, в Pilot.ClientSdk до 2000, в серверных модулях до 1000. Исключение составляет сборщик автоматизаций: в metrics.automation_events он пишет значения expected и was из сообщений об ошибках и до 200 символов тела ошибки, поэтому туда могут попасть значения атрибутов объектов. Адрес сборщика в модулях задаёт администратор: в поставляемом Reports.config.json адреса пусты, пока statsUrl пуст, статистика выключена; выпущенные ранее версии модуля собраны с адресами исходной машины.

Логин, ФИО и имя компьютера относятся к персональным данным работников. Нужно решение ИБ: достаточно ли внутреннего уведомления. Срок хранения владелец стека определил: 30 дней. Сборщики сами удаляют события старше срока из metrics.report_events, metrics.module_events и metrics.automation_events при старте и затем не реже раза в час (сборщик автоматизаций проверяет при каждом запуске, раз в 5 минут), пачками по 5000 строк. Срок задаётся переменной STATS_RETENTION_DAYS в .env (по умолчанию 30); значение 0 отключает удаление, и сборщики пишут об этом предупреждение в журнал. Ошибка удаления попадает в журнал и работу сборщика не останавливает. Резервные копии базы pilot_metrics, сделанные до удаления, хранят старые строки до собственной ротации. Журналы Loki хранятся 720 часов (30 дней, retention_period в конфигурации Loki).

8. Недекларируемые возможности

Что означает этот раздел

Недекларируемые возможности (НДВ) это функции ПО, не описанные в документации, в результате использования которых могут быть нарушены конфиденциальность, доступность или целостность данных. Формальный контроль отсутствия НДВ проводит аккредитованная лаборатория.

> **Сертификата или заключения лаборатории на компоненты нет.** Этот раздел содержит заявление разработчика, полный перечень функций и результаты самопроверки. Он не заменяет сертификацию. Если система подпадает под обязательные требования, например как значимый объект КИИ или ГИС, нужна проверка по этим требованиям.

Что умеют наши компоненты

Здесь перечислены функции, чувствительные с точки зрения безопасности. Других таких функций при самопроверке не найдено.

Компонент	Что может делать	Чем ограничено
Pilot ProAdmin	Записывает автоматизации, отчёты, типы и модули в Pilot. Создаёт ветки, коммиты, merge request, теги и Release в GitLab и защищает ветку dev. Правит файл настроек сервера Pilot server-settings.conf всех контуров, включая прод (каталоги смонтированы на запись). Распаковывает zip модулей: запись до 256 МБ, архив до 1 ГБ	Права вошедшего пользователя в Pilot и GitLab (роль не ниже Developer, при ошибке определения роли действие запрещено), успешный pipeline. Настройки, миграция и серверы только для администратора прода. Docker и перезапуск сервера админке недоступны
Pilot ProAdmin, выкладка из CI	Записывает модуль в Pilot под служебной учёткой контура. Ветка, коммит, pipeline, проект и автор берутся из ответа GitLab	Токен задания, проверяемый в GitLab, и ключ контура (MODULES_DEPLOY_KEY_TEST, _STAGE, _PROD), выведенного из проверенной ветки; проект должен лежать в группе модулей этого вида; после 10 неудач 429. Пустой ключ отключает выкладку на контур
Миграция	Останавливает и запускает базу приёмника, заменяет её данные, удаляет из архива приёмника файлы, которых нет в источнике, заменяет метаданные и автоматизации, пишет server-settings.conf, перестраивает индекс атрибутов, пишет пробные объекты в архивы, выгружает базы в файлы вместе с хешами паролей пользователей, импортирует присланные базы	Ключ API. Запись в прод запрещена проверкой меток контуров: перенос блокируется, если у приёмника тот же адрес Pilot, settings.xml или папка архива, что у прода. Файловые тома прода смонтированы только для чтения. Защита от общей папки данных PostgreSQL работает только при праве на pg_control_*, без него остаётся предупреждение. Учётные данные прода в сервисе есть (открытым текстом в config.json)
Сборщики статистики	Читают лог автоматизаций, принимают события, пишут их в pilot_metrics	Сборщикам автоматизаций смонтирован только каталог логов контура (чтение), они работают не от root. Сборщик событий при заданном ключе требует его и для сводок, тело запроса до 10 МБ

Компонент	Что может делать	Чем ограничено
ServerHost	Загружает сборки серверных модулей в процесс Pilot-Server и выполняет их код	Модули выкладывает админка: выкладка из CI подтверждается токеном задания и ключом контура, ключи stage и prod доступны только заданиям защищённых веток. Имя модуля проверяется (буквы, цифры, _, -, ., до 100 знаков), архив не больше 512 МБ после распаковки и 5000 файлов (200 МБ в сжатом виде), текст ошибки в статистику до 1000 знаков. Сам код хост не проверяет и выполняет любую сборку, выложенную админкой
Pilot.ClientSdk	Отправляет статистику. В окне настроек проверяет доступность сборщика	Адрес задаётся настройкой
Reports	Отправляет отчёты в GitLab, хранит токен, открывает браузер для входа. Для поиска токена вызывает git credential fill, если токена нет: это чтение сохранённого в Git токена к адресу GitLab	Только по действию разработчика отчётов, только адрес GitLab из настроек. Ссылки из ответов GitLab открываются в браузере только по http и https; git.exe ищется только по каталогам PATH, ожидание ответа до 5 с; токен при http-адресе GitLab передаётся открытым текстом, в лог раз за сеанс пишется предупреждение; отчёт отправляется в ветку dev или hotfix/… от main, не в main
debug	Подключается к Pilot и запускает модуль на машине разработчика	Учётка разработчика. Запись только при apply (в файле настроек или ключом --apply) и после подтверждения в консоли с показом сервера и логина; ключ --yes отключает вопрос, без подтверждения код возврата 1, в неинтерактивной среде без --yes запись не выполняется. Пароля в файле настроек нет, образец поставляется с apply: false
Скрипт запуска Pilot-Server	При старте контейнера меняет settings.xml по файлу настроек	Только 7 перечисленных параметров и только значения true или false

Что проверено

Самопроверка исходного кода выполнена 5 ноября 2026 года; перечень в этом разделе обновлён 10 ноября 2026 года с учётом внесённых исправлений. Проверялись proadmin, Клиентские модули, Reports, Серверные расширения и сборщики статистики. Проверка выполнена поиском по коду, это не аудит и не полный статический анализ.

Что искали	Как	Результат
Адреса в интернете	Поиск URL в исходниках и конфигурации	cdnjs.cloudflare.com в редакторе больше не используется: CodeMirror лежит в самой админке, см. раздел 3
Сетевые клиенты	Поиск HttpClient, WebRequest, TcpClient	Обращения только к Pilot, GitLab, миграции и сборщику; запросы админки к GitLab идут с тайм-аутом 15 с и без переходов на другой хост
Посторонняя телеметрия	Поиск telemetry, analytics, sentry, applicationinsights, просмотр зависимостей	Внешней телеметрии нет. Собственная статистика идёт только в свой сборщик
Встроенные учётные записи и пароли	Поиск строковых значений password, secret, token, apikey в коде	В рабочем коде не найдено. Пароли-заглушки были в .env.example, compose и скриптах: при первой установке на одну машину установка заменяет их случайными, на нескольких машинах останавливается, пока они не заменены; подстановки minioadmin для S3 миграции убраны. В compose остаются запасные значения служебных переменных, поэтому проверяйте .env после установки. Тесты не проверялись
Запуск сторонних программ	Поиск Process.Start	Только открытие браузера и git credential fill в Reports
Доступ без входа в админке	Поиск AllowAnonymous	/health, /Login, /Error, /Process (справка), выход. Маршруты /api/ci/* проверяют токен задания в GitLab и ключ контура. Файлы /downloads только после входа

Что можно проверить самостоятельно

1. **Сеть.** Запретите исходящий доступ машин стека и журналируйте попытки соединений. Стек должен работать, а попытки покажут всё, что идёт наружу.
2. **Исходный код.** Код лежит в репозитории проекта (монорепозиторий). Образы собираются из Dockerfile его папок, поэтому их можно собрать заново и сравнить поведение. Для сборки нужен доступ к mcr.microsoft.com, nuget.org и PyPI (в образе сборщика автоматизаций закреплён psycopg2-binary==2.9.10).
3. **Статический анализ.** Проведите его своим инструментом, ориентир для процесса: ГОСТ Р 71207-2024.
4. **Зависимости.** Постройте перечень компонентов и проверьте его по базе уязвимостей: dotnet list package --include-transitive и сканер образов.
5. **Сторонние продукты.** Сведения о сертификации и поведении Pilot запрашивайте у АСКОН, об остальных продуктах у их производителей. Наша проверка их не охватывает.

Заявление разработчика

Заполняется и подписывается ответственным лицом разработчика после своей проверки.

Разработчик заявляет, что в компонентах стека, перечисленных в разделе 2, версии от _____, на момент передачи: функции, чувствительные с точки зрения безопасности, описаны в настоящем разделе; разработчиком намеренно не внесены скрытые учётные записи, средства удалённого управления и функции передачи данных за пределы сети организации. Заявление не является сертификатом соответствия и не распространяется на стороннее ПО.

Организация	Должность, ФИО	Подпись, дата

9. Известные риски

Риск	Почему важно	Что можно сделать
Нет TLS	Пароли Pilot, токены GitLab и статистика идут по сети открытым текстом; токен GitLab в Reports при http-адресе тоже. Cookie входа получает флаг Secure только при HTTPS	Включить HTTPS для админки и Grafana (сертификат и ключ в папке certs, установка включает его сама); для GitLab настроить external_url с https и сертификат; порядок настройки HTTPS у Pilot-Server описан у поставщика, до этого поставить перед ним обратный прокси с HTTPS и сегментировать сеть
Runner имеет доступ к docker.sock	Равносильно root на машине GitLab: код из pipeline может влиять на все контейнеры этой машины	Вынести runner на отдельную машину, защитить ветки (main, release/*), ограничить, кто может менять CI
Promtail, cAdvisor и node-exporter с широким доступом	Читают логи и данные машины, cAdvisor работает в режиме privileged	Закрыть их порты от всех, кроме metrics, или отключить
Сборщик принимает данные без ключа	Любой в сети может записать ложную статистику и, пока ключ пуст, читать сводки	Задать COLLECTOR_API_KEY и прописать его в настройках модулей. При первой установке на одну машину ключ создаётся, при установке с данными и на нескольких машинах его нужно задать самому; при пустом ключе сборщик пишет предупреждение при старте. Тело запроса ограничено 10 МБ
Группа docker на Linux	Участие в группе равносильно правам root на машине	Включать в неё только администратора стека

Риск	Почему важно	Что можно сделать
Один .env на всех машинах	При разнесённой установке компрометация машины test раскрывает пароли прода	На каждой машине оставить только нужные пароли
Роли PostgreSQL для статистики	Сборщики пишут под ролью metrics_writer (владелец базы pilot_metrics), Grafana читает под metrics_reader (только SELECT, источник помечен только для чтения). У обеих ролей нет прав на базы Pilot, у pilotuser нет прав на pilot_metrics. Роли ходят в PostgreSQL по сети без TLS. База Pilot, созданная после запуска установки, остаётся доступной для захода (без прав на данные) до следующего запуска установки	Закрыть порт PostgreSQL брандмауэром, оставить доступ только машинам стека (готовый файл правил установка пишет при INTERNAL_BIND); пароли ролей менять через shared/rotate-contour-secrets
Сетевые папки контуров	В settings.xml есть пароль PostgreSQL. Папки доступны машине admin, раздел «Серверы» пишет server-settings и прода	Открыть только машине admin, отдельная учётка; прод только для чтения, если настройки прода не правятся из раздела «Серверы»
Мост без шифрования	Контейнеры link-* пересылают трафик между машинами как есть	Выделенная сеть между серверами или шифрованный туннель
Выгрузки и бэкапы	Папки export, бэкапы и data содержат данные и секреты	Ограничить доступ, хранить отдельно, удалять ненужное
Секреты открытым текстом	Пароли администраторов Pilot и ключи S3 в config.json миграции (права 600 только в Linux-контейнере), пароль PostgreSQL в settings.xml, .env, PASSWORDS.md, ключи шифрования админки в файлах без защиты; всё это попадает в резервные копии	Закрыть доступ к DATA_ROOT и ./data средствами ОС (на Windows права DATA_ROOT установка не меняет), шифровать копии и ограничивать доступ к ним
Права Maintainer в GitLab	Maintainer сливает в main (прод) и пишет в release/* (стейдж), ему доступны Protected-ключи STAGE и PROD. Ключ test защищён только флагом Masked. «Pipelines must succeed» оценивается по .gitlab-ci.yml ветки merge request, который меняет автор	Ограничить число Maintainer и владельцев групп, включить ревью merge request, следить за переменными групп
Вход в админку администратором Pilot	Каждый, кто работает в админке, входит учёткой администратора сервера Pilot; администратор прода к тому же меняет общие настройки, миграцию и серверы	Ограничить круг таких учёток, следить за журналом контейнера админки: входы, неудачные входы, смены настроек
Миграция хранит учётные данные прода	Ключ MIGRATION_API_KEY равносилен доступу к учёткам всех контуров. Запрет записи в прод держится на метках контуров и сверке адресов; защита от общей папки данных PostgreSQL работает только при праве на pg_control_*	Хранить ключ только на машине admin, закрыть порт 5320 брандмауэром, выдать подключению прода отдельную учётку с минимальными правами

Риск	Почему важно	Что можно сделать
Версии .NET в стеке	Образы админки, миграции и сборщика метрик собраны на .NET 10, его поддержка действует до ноября 2028 года (дату проверьте на сайте Microsoft). Серверные расширения (ServerHost, Debug) собираются под .NET 8 и работают внутри Pilot-Server: их среда зависит от образа Pilot-Server, а его выбирает АСКОН. Поддержка .NET 8 закончилась 10 ноября 2026 года, обновлений безопасности для неё не будет	Пересобирать образы сервисов при выходе исправлений безопасности .NET. Версию .NET в образе Pilot-Server уточнить у АСКОН и запросить образ на поддерживаемой версии
Скрипты отчётов получают доступ к Pilot	Модуль Reports передаёт скриптам отчётов (из GitLab и Pilot) сервисы клиентского SDK: IObjectsRepository, ISearchService, IObjectModifier, IFileProvider, диалоги. Скрипт отчёта может читать и менять объекты и файлы в Pilot с правами пользователя, открывшего отчёт	Доверять только отчётам из проверенных репозиториев GitLab (защита веток, ревью merge request), ограничить права пользователей Pilot по принципу минимальных прав