

?????????????: ??????
???????????? - ?????

????????

Эта статья разбирает архитектуру обработки заданий печати, копирования и сканирования.

???? ? ??????

ПринтМенеджер нужен для обработки заданий печати, сканирования и копирования, а также для авторизации пользователя на устройстве.

????????????

- **ПринтМенеджер** — сервер с описанными ниже компонентами.
- **АРМы сотрудников** — источник заданий. В зависимости от сценария печати на АРМ работает либо Клиент ПМ с локальной очередью печати рабочей станции, либо универсальный PostScript-драйвер, настроенный на виртуальный принтер, находящийся на сервере печати (см. сценарии печати ниже).
- **Встроенное приложение** (на принтере/МФУ) — управляет заданиями печати/копирования/сканирования и авторизацией пользователя.
- **Внешнее устройство авторизации** — опционально, для аутентификации на принтере/МФУ без встроенного приложения.

????? ????????? (??? ?
?????????)

- **PostgreSQL** — данные пользователей, информация об устройствах, настройки печати, атрибуты заданий печати, копирования, сканирования, статусы заданий. В отличие от мониторинга, здесь **нет ClickHouse** — нет необходимости хранить большой объём сырых данных от МФУ.

- **Redis** — брокер сообщений между Django, планировщиком и Celery, как и в мониторинге.
- **Django** — бэкэнд: принимает запросы, создаёт задания, управляет авторизацией и настройками, отдаёт данные для админ-панели ПринтМенеджера.
- **Celery** — тяжёлые и долгие операции: обработка PDF-документов, отправка заданий в печать, работа с очередями CUPS, работа с внешними интеграциями.
- **Планировщик** — периодические задачи: очистка старых заданий, проверка очередей печати, обновление статуса устройств, уведомления.
- **Nginx** — реверс-прокси, принимает запросы от всех внешних компонентов и передаёт их в Django.

??????????, ????????????? ????
????????????????

Компонент	Роль
Сервер печати CUPS	Обрабатывает очереди печати, работает с драйверами принтеров, отправляет задания на физическое устройство.
Файловый сервер CUPS	Позволяет эффективно скачивать задания из CUPS.
Локальное хранилище образов документов	Директория на диске сервера ПринтМенеджера. Хранит образы документов (PDF/PostScript). В базе данных не рекомендуется хранить документы, поэтому для них выделено отдельное хранилище, а в PostgreSQL хранятся только метаданные (ссылка на задание, атрибуты задания: дуплекс, цветность и т.д.), а образ лежит в хранилище.
Временное объектное хранилище	Транзитная зона для МФУ, которые не умеют отправлять данные напрямую в систему и работают только через FTP/SMB/сетевые папки. Celery забирает образ оттуда и передаёт дальше по цепочке в постоянное хранилище.
Конвертер- сервер	Нужен для МФУ, которые не поддерживают встроенное приложение и используют внешнее устройство авторизации (например, кардридер). Конвертер формирует авторизационные данные и приводит их к формату, понятному Django.

????????? 1: ??????? ?????? ????
??

На АРМ пользователя установлен Клиент ПМ — компонент, работающий с очередью печати рабочей станции, управляющий принтерами на АРМ и отвечающий за применение экономии тонера. Путь задания при печати через клиента ПМ:

1. Пользователь отправляет документ на печать. Задание попадает в **локальную очередь печати** рабочей станции.
2. Клиент ПМ забирает задание из локальной очереди, **конвертирует его в PDF** — основной формат для дальнейшей обработки в Printum — и извлекает атрибуты (формат, дуплекс, количество страниц, цветность, ориентация, разбор по копиям).
3. Клиент ПМ добавляет задание в серверную очередь печати ПринтМенеджера. На сервере Django создаёт внутреннее задание и кладёт метаданные в PostgreSQL; ПринтМенеджер проверяет и применяет к заданию правила печати.
4. Клиент ПМ запрашивает у ПринтМенеджера настройки оптимизации печати (например, параметры экономии тонера), применяет их и отправляет обработанный образ документа на сервер в локальное хранилище образов.
5. Дальнейший путь зависит от типа печати:
 - **Прямая печать** — задание уходит в сервер печати CUPS, CUPS отправляет его на МФУ.
 - **Отложенная печать** — задание остаётся в очереди ПринтМенеджера. Пользователь авторизуется во встроенном приложении на МФУ (PIN-код или карта), приложение запрашивает список заданий; после выбора пользователем и нажатия «Печать» задание уходит в CUPS, CUPS — на МФУ.
6. После отправки на печать ПринтМенеджер переводит задание в архив, обновляет статистику, которая потом подгружается мониторингом.

???????? 2: ?????????????? ??????

Настраивается через виртуальный принтер со ссылкой напрямую на CUPS ПринтМенеджера, без установки Клиента ПМ на АРМ. К виртуальному принтеру подключается один из универсальных PostScript-драйверов: Xerox Global Print Driver PostScript, Konica Minolta Universal PS, HP Universal Printing PS.

Путь задания при печати без клиента ПМ:

1. Универсальный драйвер формирует PostScript-задание и отправляет его напрямую в CUPS ПринтМенеджера, минуя Django на этом шаге.
2. Django периодически опрашивает CUPS; обнаружив новое задание, создаёт собственное внутреннее задание и заполняет его атрибутами, разобрав пришедший PostScript-файл.
3. Дальнейшая обработка зависит от настройки **USE_PS_PRINTING**: если она выключена — задание конвертируется в PDF; если включена, но для задания

действуют правила по количеству страниц, цветности или автоудалению после печати — задание всё равно конвертируется в PDF; если таких правил нет — конвертация не выполняется, и задание остаётся в PostScript.

4. ПринтМенеджер применяет правила печати к заданию; метаданные сохраняются в PostgreSQL, образ — в хранилище образов (локальное или внешнее).
5. Исходное PostScript-задание удаляется из CUPS — дальше система работает только со своей внутренней копией задания.
6. Дальнейший путь зависит от типа печати:
 - **Прямая печать** — сформированное задание отправляется в CUPS, CUPS передаёт его на МФУ.
 - **Отложенная печать** — задание остаётся в очереди ПринтМенеджера до авторизации пользователя во встроенном приложении на МФУ, дальше — как в сценарии с Клиентом ПМ.
7. После печати — архивирование задания, обновление статистики и синхронизация с мониторингом как обычно.

?????? ?????? ?????? (IMAP)

Помимо клиентской/бесклиентской печати с АРМ, есть сценарий отправки задания на специальный почтовый ящик. Периодическая задача Celery проверяет этот ящик по протоколу IMAP; обнаружив новое письмо с вложением, система забирает его и обрабатывает как обычное задание печати — метаданные в PostgreSQL, образ в локальное хранилище, дальнейшая авторизация и печать не отличаются от прочих сценариев.

??????? ?????????? ?????????????? ????????? ??????????????????

- **SMTP-сервер** — уведомления, в том числе о том, что задание заблокировано правилами печати.
- **IMAP-сервер** — печать через почту и гостевая печать (см. выше).
- **FTP/SMB-папка для выгрузки архива** — периодическая выгрузка образов документов из локального хранилища для освобождения места на сервере; метаданные при этом остаются доступны через мониторинг (после синхронизации), а сами образы — в архивной папке.
- **Сетевая папка для отсканированных документов** — используется при сценарии «сканирование в папку»: образ идёт во временное объектное хранилище, Celery обрабатывает и через Django/nginx также кладёт копию в сетевую папку.
- **NFS-хранилище** — в базовой конфигурации **не используется**. Оно появляется только в схеме "Кластер", где хранилище должно быть общим для нескольких ПринтМенеджеров.

Базовая конфигурация: что такое базовая конфигурация и зачем она нужна?

Базовая конфигурация (Сингл) — это один сервер. Если в инфраструктуре несколько ПринтМенеджеров, это происходит по одной из двух причин: нужна отказоустойчивость/производительность или несколько географически разнесённых филиалов.

Варианты базовой конфигурации

- [Поддерживаемые сценарии печати](#)
 - [Отказоустойчивость и горизонтальное масштабирование ПринтМенеджера](#)
-