Блог · 5 октября 2026 · 16 мин чтения

Свой Google внутри компании: кейс FlyOptima, 2026

Как FlyOptima за месяц собрала из открытых компонентов поиск по всей базе знаний компании: вопрос своими словами, ответ со ссылкой на документ, доступ по токенам и проверка каждого файла до поиска.

Коротко. Поиск по всей базе знаний компании можно собрать самим из открытых компонентов. У FlyOptima на это ушел месяц работы одного инженера на своем оборудовании. Система ищет среди примерно 790 тысяч фрагментов текста за полсекунды, а локальная модель дает ответ со ссылками на документы за время от 40 секунд до нескольких минут. Документы при этом обрабатываются на своем оборудовании.

За несколько лет работы в любой компании накапливается огромный массив разношерстных данных: переписка, договоры, схемы, технические документы, протоколы встреч, задачи, таблицы, презентации. Потом руководителю нужно найти в этом массиве одно давно принятое решение.

Например, руководитель проекта спрашивает в рабочем чате, какой вариант схемы согласовали с заказчиком. Все помнят, что вопрос решили давно и решение записали. Где именно, не помнит никто.

Один коллега советует искать в почте, другой присылает презентацию не той версии. Нужная запись может лежать во вложении к письму, в протоколе встречи, в общей папке или в комментарии к задаче, и в каждом из этих мест свой поиск. Ищут до конца дня и наутро рисуют схему заново, так как это быстрее, чем искать дальше.

Ответ давно записан, но найти его нельзя.

Данные копятся и не используются: к ним нет легкого доступа. Я основал FlyOptima в 2022 году, мы разрабатываем решения для банков и финтеха, наш продукт – FlyCoreBanking, платформа для карт и кредитования. Команда небольшая, но в документах, вики, задачах трекера и переписке счет идет на сотни тысяч фрагментов текста.

Если ничего не менять, решения и дальше будут приниматься повторно. Новый сотрудник будет месяцами спрашивать о том, что давно описано, а с уходом человека компания потеряет то, что он помнил: записи останутся, но где в них ответ, знал только он.

Цель и результат

Цель была одна: находить ответ в архиве компании за секунды, а не за день.

Сегодня это работает. Систему мы назвали FlyArchive, сейчас это версия 0.1. Ей задают вопрос своими словами, и она ищет по смыслу и по точному слову среди примерно 790 тысяч фрагментов текста. Поиск занимает полсекунды (медиана – 513 мс). Локальная модель читает найденное и дает ответ со ссылками на документы: на это уходит от 40 секунд до нескольких минут. Когда она впервые ответила со ссылками на наши собственные документы, была эйфория.

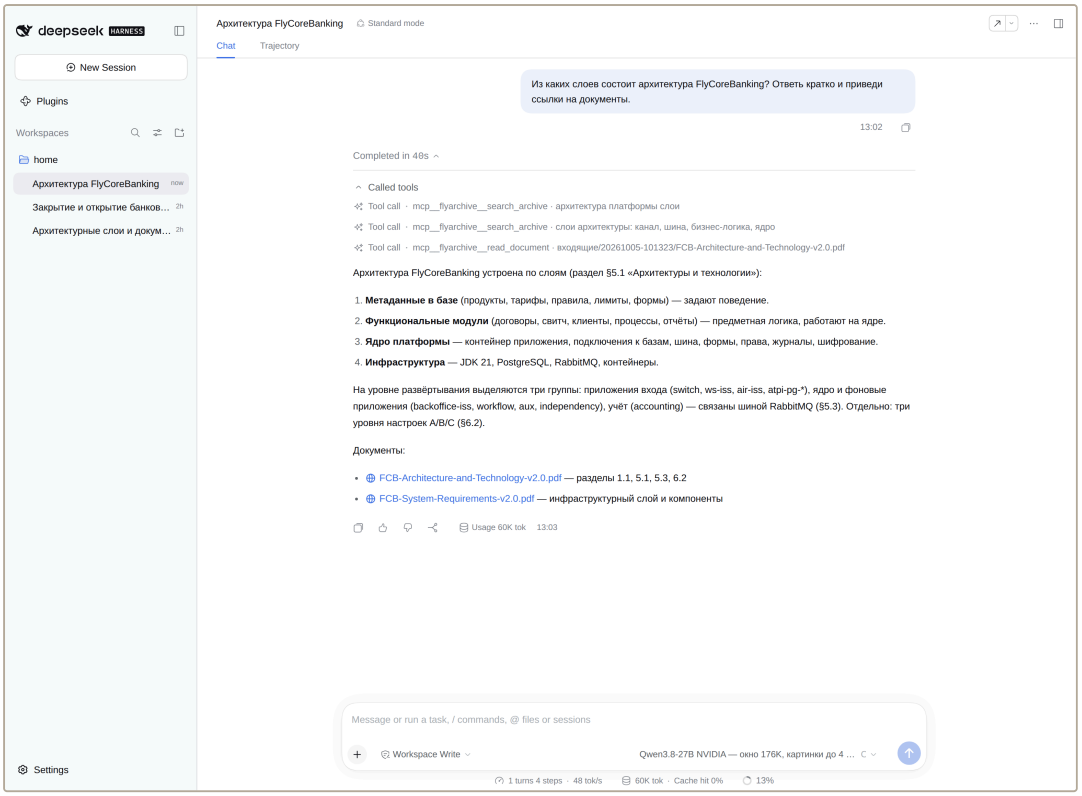


Рисунок 1. Технический вопрос по документации FlyCoreBanking: локальная модель трижды обратилась к архиву и за 40 секунд ответила со ссылками на документы

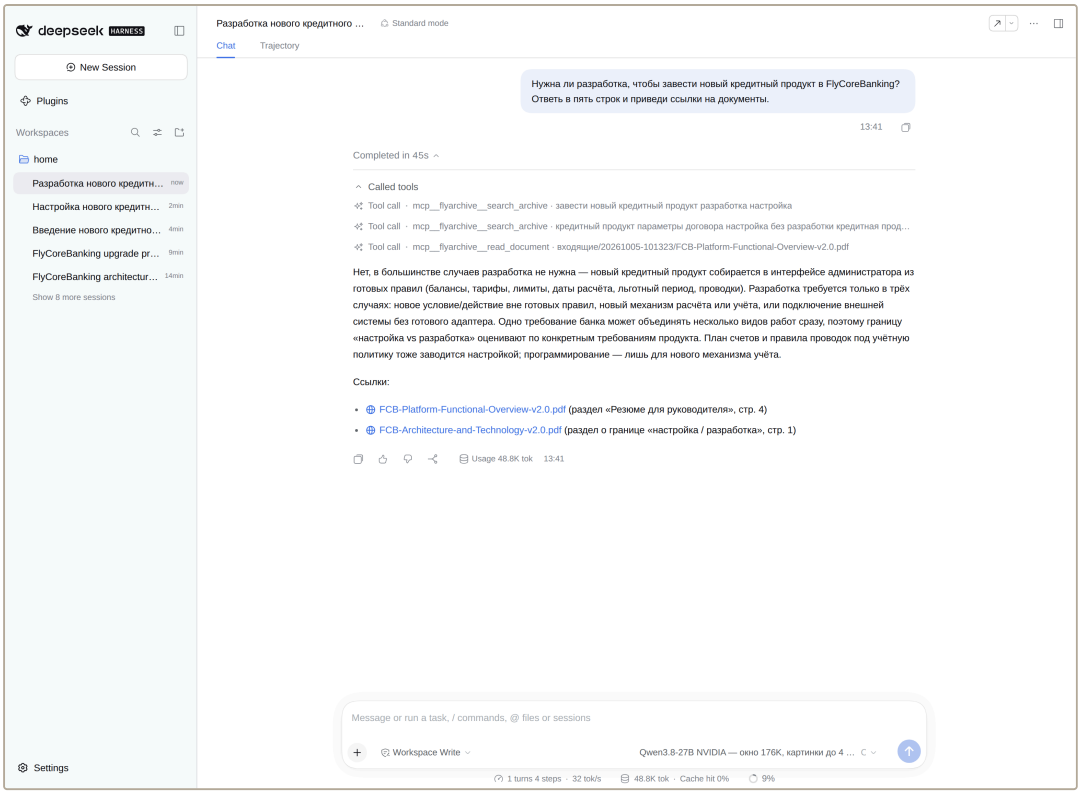


Рисунок 2. Вопрос о процессе, нужна ли разработка для нового кредитного продукта. Ответ: обычно не требуется. Три обращения к архиву, 45 секунд, ссылки на документы

Чего это стоило: работа одного инженера на своем оборудовании и месяц от идеи до работающего решения. Код пишут модели-агенты под присмотром инженера.

Что ограничивает, скажу сразу. Зона доверия одна: кто получил токен, видит весь архив. Одновременно работает одна модель. Инженер тоже один. Системой я пользуюсь сам, части команды выдаю токены. Собрать такую может компания, у которой есть инженер и свое оборудование с видеокартой.

Из чего собрано

Одному инженеру это по силам, так как почти все уже написано другими. Своего кода в системе четыре части: поиск, приемка файлов, доступ по токенам и плагин управления. Остальное – готовые компоненты из таблицы, почти все с открытым кодом.

Компонент	Роль	Лицензия
LanceDB	хранит индекс для поиска, файлами на диске	Apache-2.0
BAAI/bge-m3	модель, которая готовит текст к поиску по смыслу	MIT
Ollama	запускает эту модель на процессоре	MIT
Qwen3.8-27B	локальная языковая модель, отвечает на вопросы	Apache-2.0

Компонент	Роль	Лицензия
vLLM	запускает локальную модель на видеокарте	Apache-2.0
llama-swap	посредник перед локальной моделью	MIT
PyMuPDF	читает PDF	AGPL-3.0 или коммерческая
python-docx, openpyxl, python-pptx	читают файлы Word, Excel и PowerPoint	MIT
extract-msg	читает письма Outlook	GPL-3.0
Docling	разбирает документы с таблицами	MIT
EasyOCR	распознает текст на картинках	Apache-2.0
SQLite	помнит, какие файлы уже есть в архиве	общественное достояние
bubblewrap	песочница для чужих программ и просмотра файлов	LGPL-2.0-or-later
systemd	запускает службы и разбор по расписанию	LGPL-2.1-or-later
7-Zip	распаковывает архивы	LGPL-2.1+
DeepSeek Harness (DSH)	оболочка: чат с моделью и плагины	MIT
MCP	протокол, по которому модель обращается к инструментам	Apache-2.0
Tailscale	частная сеть между устройствами	клиент – BSD-3-Clause, сервис закрыт
Python, Node.js	языки серверной части и плагина	PSF-2.0, MIT
Ubuntu в WSL2	операционная система	у пакетов свои лицензии, ядро – GPL-2.0

Как это устроено

Устроено это проще, чем кажется по таблице. На сервере архива постоянно работают служба поиска, служба документов и служба доступа, через которую модели обращаются к архиву. Все они доступны только внутри самого сервера, в сеть компании выходит один шлюз.

Остальное запускается по требованию: разбор входящей папки по расписанию, команда управления `flyarchive` и песочницы для чужих программ. Отдельных серверов баз данных нет. Индекс, список уже известных файлов, квитанции приемки и журнал обращений лежат обычными файлами на диске.

Чатом служит оболочка DeepSeek Harness (DSH). Первую оболочку, Open WebUI, мы отвергли: модель обращалась к ее встроенным «знаниям» и получала пустой ответ. Я считаю, что DSH подошел нам из-за агентской архитектуры с плагинами: модель в нем работает как агент, и архив для нее – набор инструментов. Наш плагин добавляет в оболочку раздел настроек и вкладку Archive. Подключается он одним файлом настроек, без правки самой оболочки.

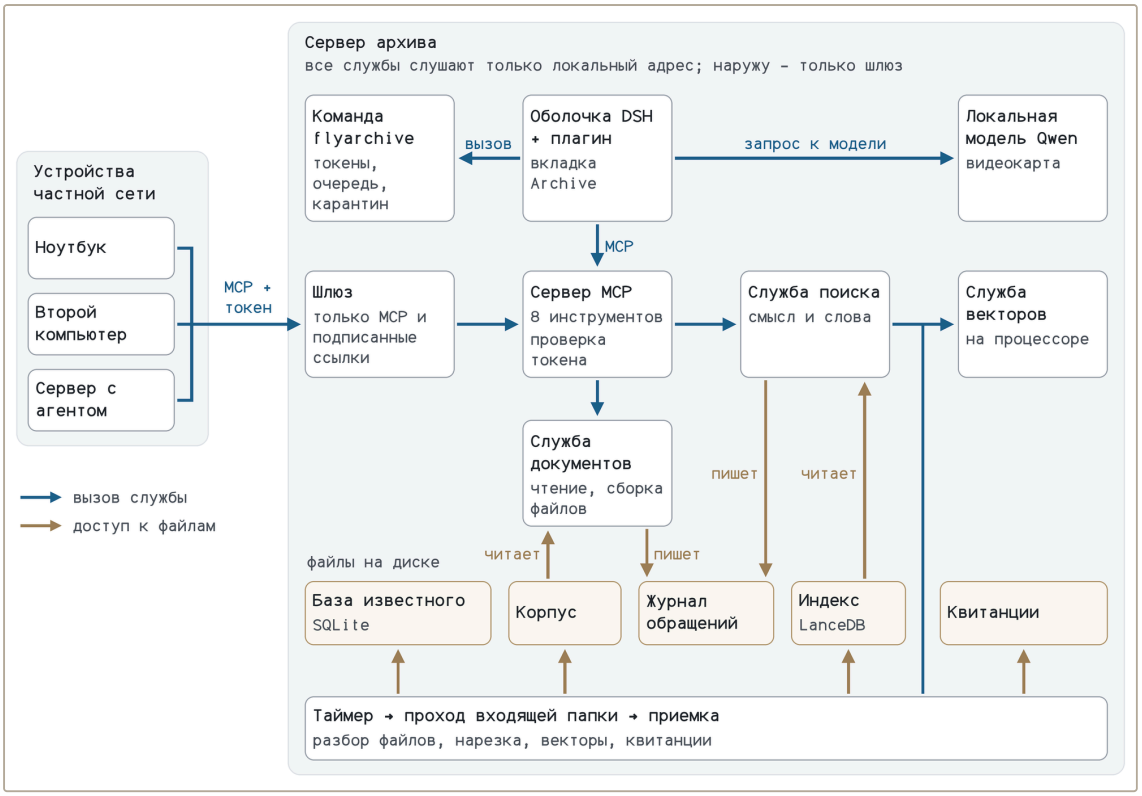


Рисунок 3. Из чего состоит система и что с чем связано

Кто и как обращается к архиву

Архив полезен только тогда, когда спросить его можно с любого рабочего места. У нас это любое устройство частной сети компании. Машины объединены в сеть Tailscale, шифрование дает она сама. Вне этой сети архив недоступен: в интернет ничего не опубликовано. Ноутбук, второй компьютер или сервер с агентом подключается по MCP (Model Context Protocol) – протоколу, по которому модели обращаются к инструментам.

Запрос проходит пять шагов.

1. Модель или оболочка на устройстве сотрудника отправляет вопрос и свой токен на шлюз.
2. Шлюз пропускает только обращения к архиву и ссылки на документы. Токен он не проверяет.
3. Служба доступа проверяет токен: известен ли он системе, не отозван ли и не истек ли срок. Уровень «только чтение» позволяет искать и читать документы, «полный» – еще создавать файлы и передавать документы в архив. Удалить документ или выпустить токен через этот вход нельзя.

4. Поиск идет двумя способами сразу, по смыслу и по точному слову, и ставит свежие документы выше. В ответ уходят найденные фрагменты и ссылка на документ, она действует 24 часа.
5. Обращение пишется в журнал: какой клиент спрашивал, с какого узла сети, что искал и чем это кончилось. Отказы пишутся тоже.

Токен выдается на клиента и отзывается сразу: следующий запрос с ним уже не пройдет.



Рисунок 4. Путь запроса: шлюз, проверка токена, поиск, журнал

Ассистент в Telegram

Тем же способом, по токену, к архиву подключен мой ассистент «Юля» на OpenClaw. Она помогает с календарем и почтой и снимает с меня много рутины. Я выдал ей токен только на чтение и вывел поиск по архиву в Telegram. Теперь в дороге и где угодно я со своего телефона задаю вопрос базе знаний компании, и отвечает мне модель. Это круто: на поиск такого ответа раньше уходил день.

Этот канал идет через модель ассистента и мессенджер, поэтому токен выдан только на чтение и отзывается одним действием.

Что лежит в архиве

Искать можно только в том, что собрано в одном месте, поэтому в архив сложено все, что компания накопила. Сырые данные лежат файлами, в исходном виде: задачи трекера и страницы вики с вложениями, рабочая переписка с вложениями, рабочие папки с документами. Вместе это 148 ГБ. Каталог архива закрыт от других пользователей.

Для поиска документ режется на куски примерно по 4000 знаков, это и есть фрагменты. В индексе их около 790 тысяч, он занимает 5,5 ГБ. Ухода индекс требует в одном месте: после массовой загрузки его нужно уплотнять, иначе объем растет (у нас вырос с 7 до 35 ГБ).

Очередь на утверждение и карантин лежат отдельно от архива и в поиск не попадают.

С какими документами работает архив

Таблица показывает, что из положенного в архив находится поиском и что открывается прямо в интерфейсе.

Что	Попадает в поиск	Открывается в интерфейсе
PDF	да; сканы без текста – пока нет, описание моделью в работе	страницами
Word, Excel, PowerPoint (docx, xlsx, pptx)	да	страницами
Старые форматы Office (doc, xls, ppt)	да, после преобразования	страницами
Письма (eml, msg) и выгрузка почтового архива (olm)	да, вместе с вложениями	шапка, текст и вложения; вложение открывается так же
Текст и данные (txt, md, csv, json, xml, sql, yaml, log)	да	текстом
Страницы HTML	да	страницей, без сценариев и внешних картинок
Схемы (bpmn, drawio, svg)	надписи со схемы	svg – да; bpmn и drawio – пока нет
Картинки (png, jpg, gif, tiff, bmp, webp)	распознанный текст; описание моделью в работе	да
Архивы (zip, 7z, tar)	содержимое проверяется и принимается по одному файлу; rar – пока нет	состав архива, без распаковки
Календари (ics)	нет	события
Программы и скрипты	нет: сразу в карантин	только сведения о файле
Видео и звук	нет	пока нет

Как архив пополняется

Архив, собранный один раз, начинает устаревать на следующий день. Поэтому он пополняется сам. Новый документ приходит двумя путями: его кладут во входящую папку или передает агент с полным уровнем доступа. Каждые 30 минут система разбирает входящее: проверяет файлы, раскладывает принятое и добавляет его в индекс. Файл, который в архиве уже есть, второй раз в него не попадет. На каждую пачку остается квитанция: что пришло и какое решение принято по каждому файлу.

Приемка и карантин

Когда архив пополняется сам, в поиск может попасть что угодно. Поэтому есть приемка: она решает, что попадет в архив. Логика такая.

1. Тип файла определяется по содержимому, расширение в расчет не берется.
2. Программа сразу уходит в карантин.
3. Архив распаковывается с пределами по объему, числу файлов и глубине вложенности. Архив с паролем или с подозрительными путями внутри уходит в карантин целиком.
4. Правила ищут макросы, скрытый текст, секреты и указания, обращенные к модели. Каждая находка дает баллы.
5. Сумма баллов сравнивается с порогом 20. Не выше порога – файл принят, выше – он ждет в очереди, пока его не утвердит человек. Самые тяжелые находки отправляют файл в карантин.

Поверх правил файл читает проверяющая модель, так как вежливо написанное указание для модели правила не ловят. Она может только ужесточить решение.

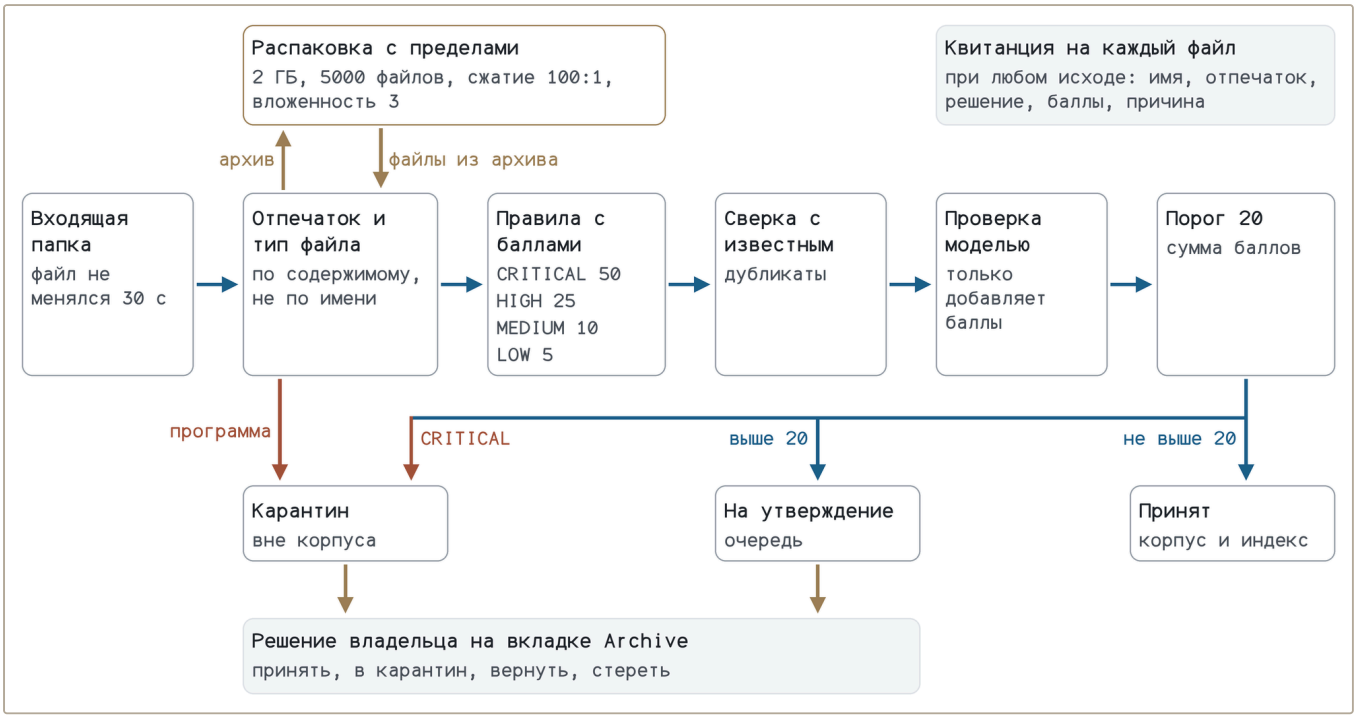


Рисунок 5. Приемка: что происходит с файлом от входящей папки до архива, очереди или карантина

Очередь и карантин ведет владелец архива во вкладке оболочки: принять, отправить в карантин, вернуть или стереть.

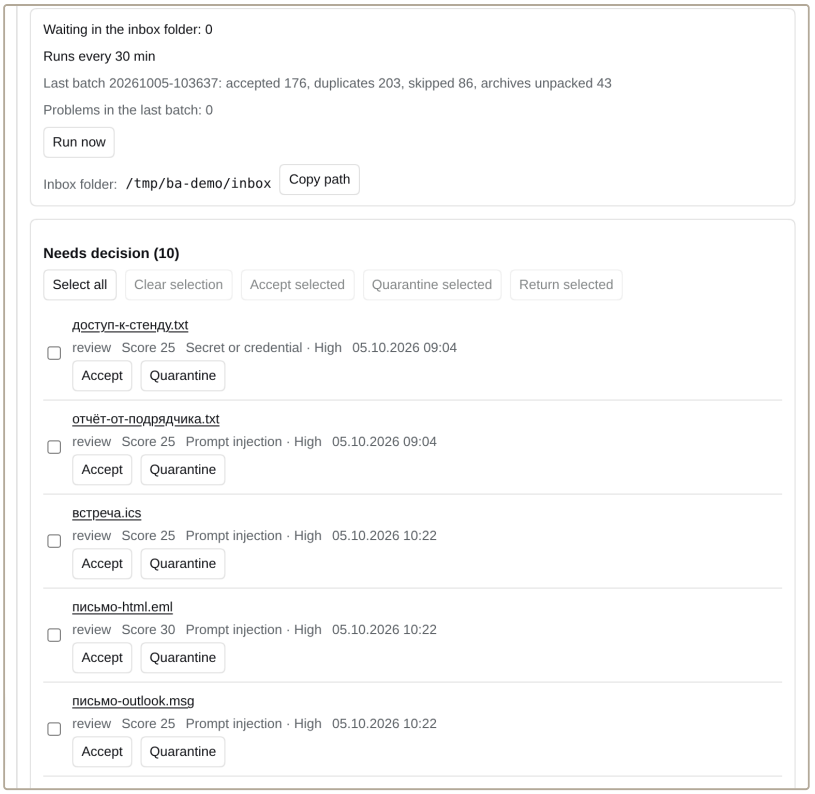


Рисунок 6. Вкладка Archive на пробных файлах: в очереди письма с указанием для модели, программа и архив с паролем, рядом история пачек

Фильтры работают. На пробе из 46 558 файлов приемка сама разобрала почти все: 11 265 приняла, 33 555 пропустила как не документы, 1 722 отсекла как повторы. Человеку осталось решить по пяти файлам.

Картинки: два способа

Часть знаний компании лежит на картинках: это снимки экрана, схемы и сканы.

Первый способ – распознавание текста. Оно дешево и точно в словах, им обработан основной объем: около 80 тысяч фрагментов индекса получены с картинок. Схему оно не понимает: если на картинке нет текста, искать нечего.

Второй способ – модель со зрением: она понимает схему и может ее описать, но работает медленнее и занимает единственную видеокарту. 5 октября 2026 года мы проверили это на выгрузке схем. Через входящую папку прошло 176 файлов за 34 секунды, но 95 картинок и PDF без текста в поиск не попали. Это было разочарование: схема есть, а найти ее нельзя. Зато проба зрения сработала прекрасно: локальная модель описала 13 таких схем, медиана – 5,8 секунды на схему, описание около 1 500 знаков. На двух PDF, где текст был, она воспроизвела 98 и 100% его слов.

Решение принято: описание, сделанное моделью, будет сохраняться в индекс, это в работе. Картинки при этом в облако не уходят никогда.

Железо и модель

Архив работает на своем оборудовании: процессор Ryzen 9 9950X3D, 128 ГБ памяти, видеокарта RTX 5090 с 32 ГБ видеопамяти. На видеокарте работает локальная модель Qwen3.8-27B с окном 176 тысяч токенов и зрением. Вспомогательная модель для поиска по смыслу работает на процессоре: на видеокарте она замедляла основную в 4–15 раз.

Своя видеокарта дает главное: документы и сами вопросы обрабатываются у себя и за пределы компании не уходят. Исключений два, и оба управляемые. Первое заложено в систему – запасной облачный путь: картинки в облако не уходят никогда, текст – только если локальная модель не отвечает 180 секунд, а отключается путь одним ключом. Второе – мое собственное решение: поиск для ассистента в Telegram, о нем сказано выше. Цена своей модели: она слабее лучших облачных.

Как мы шли

Когда	Что сделано
начало сентября 2026 года	появилась идея
18 сентября 2026 года	первая версия: поиск по архиву уже работал
3 октября	поставлена задача: подключать любую модель по токену и пополнять архив через входящую папку
3–4 октября	доступ по токенам, шлюз, приемка с карантином, проверка с соседнего компьютера
4–5 октября	управление из DSH: настройки, вкладка Archive, просмотр файлов

От идеи до сбора всех данных и работающего решения прошел месяц, от первой версии до нынешней – 17 дней. Основная часть – доступ, приемка и управление из DSH – заняла три дня.

За это время первые решения пришлось менять. Сверка индекса с архивом показала, что половина писем в нем – копии. Это было как ушат холодной воды. Следом выяснилось, что у вложений стоит дата копирования вместо даты документа, и настроения стали упаднические. Мы исправили обе ошибки, и после правки дат выдача сменилась почти целиком.

Потом архив начал пополняться сам, через входящую папку. С этого момента главным вопросом стало не «как быстро ищет», а «что в него попадает». Так появились приемка и карантин.

Свое или готовое

Справедливый вопрос: зачем собирать самим, если есть готовое? Готовые решения для поиска по данным компании есть, и пробовать их можно. Чувствительные материалы заказчиков нельзя просто положить в облачный сервис, поэтому нам подходит только то, что ставится у себя. Такие продукты дороги и рассчитаны на крупные компании: открытых цен у них почти нет.

Мы собрали свое из компонентов с открытым кодом. FlyOptima – AI-native компания, и мы ищем способы ускорять внутренние процессы с помощью агентов и открытых решений.

С открытым Onyx (выпуск 4.8.4) мы сравнили себя уже после того, как построили свое.

Признак	Onyx	FlyArchive
Проверка файла до того, как он попадет в поиск	проверки содержимого, карантина и очереди на утверждение нет; пределы распаковки архивов есть	приемка с правилами, проверяющей моделью, очередью и карантином
Форматы	не читает старые форматы Office, письма Outlook и схемы	читает письма Outlook и схемы, старые форматы Office преобразует
Что нужно установить	13 контейнеров с отдельными серверами баз данных	несколько служб и файлы на диске
Готовый чат и агенты	есть	своего чата нет, используется оболочка DSH
Подключения к облачным сервисам	57 готовых подключений	нет, источники выгружаются файлами
Права на отдельные документы	есть в платной части	нет, зона доверия одна
Свежие документы выше в поиске	не реализовано	учитывается

Onyx разумно взять, когда нужны готовый чат, подключения к облачным сервисам и права для многих пользователей. Свое имеет смысл собирать, когда важнее проверка каждого файла до поиска и простая установка.

Что не решено и что в планах

Готово не все. Сканы и картинки схем без текста пока не ищутся: сохранение описаний модели в работе. Поиск каждый раз заново открывает индекс, и на это уходит большая часть его времени.

В планах – разделение доступа для сотрудников и усиление информационной безопасности.

1. Разделить доступ: каждый сотрудник будет видеть только свои источники и проекты. Сейчас зона доверия одна.
2. Выдавать токены со сроком действия по умолчанию и получать оповещения о необычных запросах по журналу обращений.
3. Перевести службы архива под отдельного пользователя системы, а оболочку с чатом – в песочницу.
4. Добавить на шлюзе собственное шифрование в дополнение к шифрованию частной сети.

Код FlyArchive 0.1 и плагин для DSH открыты на GitHub вместе с описанием установки и настройки: github.com/andrewsabn/flyarchive. Серверная часть распространяется под AGPL-3.0, плагин – под MIT.

У нас вопрос о том, какой вариант схемы согласовали с заказчиком, теперь задают архиву своими словами, и ответ приходит со ссылкой на документ.

Будущее уже наступило. То, что раньше казалось невозможным, – свой Google внутри компании – в 2026 году уже реальность. Следуйте за новым миром дальше.



Андрей Сабынин

Основатель FlyOptima – AI-native разработчика решений для банков и финтеха.