

Миграция Почты по IMAP

Миграция Почты по IMAP

Оглавление

Сокращения и определения	3
Общее описание компонентов	3
Collector	3
Rima-tar	3
Rpopdb	4
Scheduler	4
Mailapi	4
Instant	4
Conkee	4
Picker	4
Irina-tar	5
Rico-tar	5
Hermes	5
Storage	5
Первая сборка с помощью instant	5
Процесс сборки почты по расписанию	8
Мгновенная сборка по IDLE	12

Сокращения и определения

Наименование	Определение
IDLE	Команда в IMAP, которая слушает IMAP-сервер и ожидает от него сообщения в реальном времени
IMAP-сервер	Сервер внешней почты, обмен с которым осуществляется по протоколу IMAP
Маппинг	Определение соответствия между различными моделями данных
UpUID	UID письма в папке на стороне IMAP-сервера, которое мы скачали последним из новых писем пользователя
DownUID	UID письма в папке на стороне IMAP-сервера, до которого мы успели скачать старые письма.

Общее описание компонентов

Collector

Основной демон в RIMAP, отвечающий за периодическую сборку писем из очереди по приоритетам и обработку уведомлений от сервиса **storage**. Напрямую обращается в IMAP, взаимодействует с базами данных: **rima-tar**, **rpopdb**, **rico-tar**, **hermes**, через **arbuzzapi** обращается к БД пользователей.

Rima-tar

БД Tarantool, выполняющая роль очереди задач для сборщиков почты. Из этой очереди сервис **collector** забирает задачи с приоритетами:

- **sync (0)** — периодическая синхронизация почтового ящика;
- **medium (512)** — исполнение нотификаций от **storage** (перемещение письма в корзину, отметки о прочтении и т.п.);
- **high (1024)** — выкачивание писем нового пользователя, дозагрузка писем при скролле папки.

На схемах обозначена как **rima**.

Rpopdb

БД MySQL, используемая для хранения метаданных всех сборщиков, настроек и лимитов отдельных IMAP-серверов.

Сборщики добавляются через **mailapi**. На основе этой базы **scheduler** ставит задачи в **collector** через **rima**. К этой же БД обращаются демоны **collector** для проверки существования сборщика.

На схемах и в описании обозначена как **rpop**.

Scheduler

Демон, отвечающий за постановку задач на сбор почты.

Раз в 15 минут (настраивается в конфигурационном файле) отправляет запрос в **rpopdb** и запускает цикл, в котором запрашивает все IMAP-сборщики и разбивает их на пакеты данных, а затем перемешивает их.

Ставит периодические задачи в **rima-tar** для исполнения сервисом **collector**.

Mailapi

Основной API сервиса Почта. Отправляет в **rpopdb** информацию о новых сборщиках.

Instant

Демон, предназначенный для работы с командой IDLE, с установленной периодичностью ставит задачи на синхронизацию сборщиков.

Совместно с **conkee** образует сервис **picker**, обеспечивая постоянное соединение с IMAP-сервером по IDLE. При появлении нового письма сразу забирает его.

Conkee

Прокси-сервис для поддержки непрерывного соединения **instant** с внешним IMAP-сервисом и не закрывает соединение без необходимости, что позволяет выпонять перезапуск **instant** без создания волны массовых переустановок соединений.

Picker

Общее название сервисов **instant** и **conkee**, образующих сервис отслеживания изменений во внешних ящиках.

В интерфейсе установщика наименования **instant** и **conkee** отсутствуют, есть только **picker**. В схемах, напротив, использованы названия сервисов **instant** и **conkee**.

Irina-tar

Tarantool для управления мгновенными сборщиками, хранит в себе данные с номером шарда и распределением шардов по мгновенным сборщикам (**instant**).

На схемах и в описаниях обозначен как **irina**.

Rico-tar

Tarantool для хранения параметров сборок: время последней успешной сборки и общее их количество.

На схемах и в описаниях обозначен как **rico**.

Hermes

Tarantool, в котором хранится состояние синхронизации по всем папкам всех сборщиков: соответствия ID в **storage** с абсолютным путем в IMAP.

Также в **hermes** приведены данные: сколько писем в папке скачано вверх (upUID) и вниз (downUID).

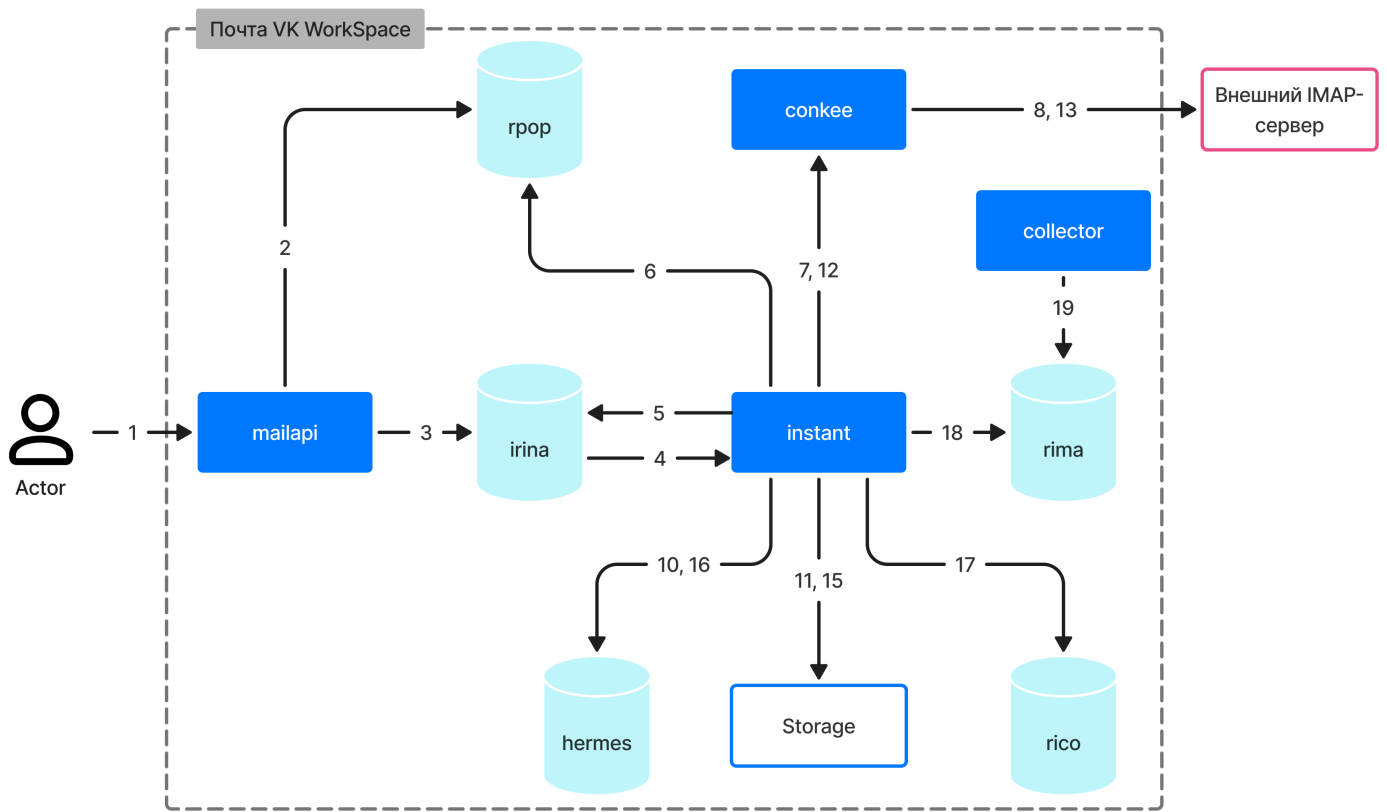
Storage

Условное наименование нескольких хранилищ данных разного назначения. Отправляют в БД **rima** нотификации по изменениям в почтовых ящиках пользователей: установка флагов, перемещение писем, их удаление и т.д.

Первая сборка с помощью instant

Первая сборка происходит после авторизации администратором нового пользователя или после того, как пользователь самостоятельно прервал сессию, а затем выполнил вход повторно (тогда пользователь также будет считаться новым). Как только пользователь стал авторизованным, выполняется мгновенная выгрузка первых 100 писем из папки Inbox (настраивается в конфигурационном файле).

Общая схема потоков данных при первой сборке приведена на рисунке ниже:



Описание потоков данных приведено в таблице ниже:

Номер потока	Источник	Описание потока	Получатель
1	Actor	Передаёт новый сборщик для покладки в БД	mailapi — основной API Почты VK WorkSpace
2	mailapi	Кладёт новый сборщик	rpop — БД MySQL, хранящая данные сборщиков
3	mailapi	Отправляет на привязку к мгновенному сборщику	irina — хранит в себе данные о мгновенных сборщиках с номером шарда и о распределении шардов по серверам
4	irina	Оповещает о новом сборщике	instant — демон мгновенной сборки писем
5	instant	Забирает нового пользователя	irina
6	instant	Проверяет сборщик	rpop
7	conkee		instant

Номер потока	Источник	Описание потока	Получатель
		Отдает команду на подключение к IMAP	
8	conkee	Соединяется с сервером IMAP, забирает список папок (протокол IMAP)	IMAP-сервер
9	conkee	Передает список папок	instant
10	instant	Отдает на внесение в БД	hermes — БД, в которой хранится состояние синхронизации по всем папкам всех сборщиков
11	instant	Сохраняет список папок	storage — хранилище данных
12	instant	Запрашивает список писем	conkee — прокси, держит непрерывное соединение с сервером IMAP
13	conkee	Забирает список писем	сервер IMAP
14	conkee	Передает список писем	instant
15	instant	Сохраняет письма	storage
16	instant	Обновляет статус сборки папки	hermes
17	instant	Обновляет тайминги	rico — БД Tarantool, хранит тайминги сборок
18	instant	Ставит задачу на сборку по расписанию	rima — очередь задач для сборщиков почты
19	collector	Слушает очередь на сборку по расписанию	rima

Описание взаимодействия:

1. После создания нового сборщика (его создает прикладной администратор) информация о нем через **mailapi** попадает в **rpop**, а затем отправляется в **irina**. **Irina** привязывает пользователя к

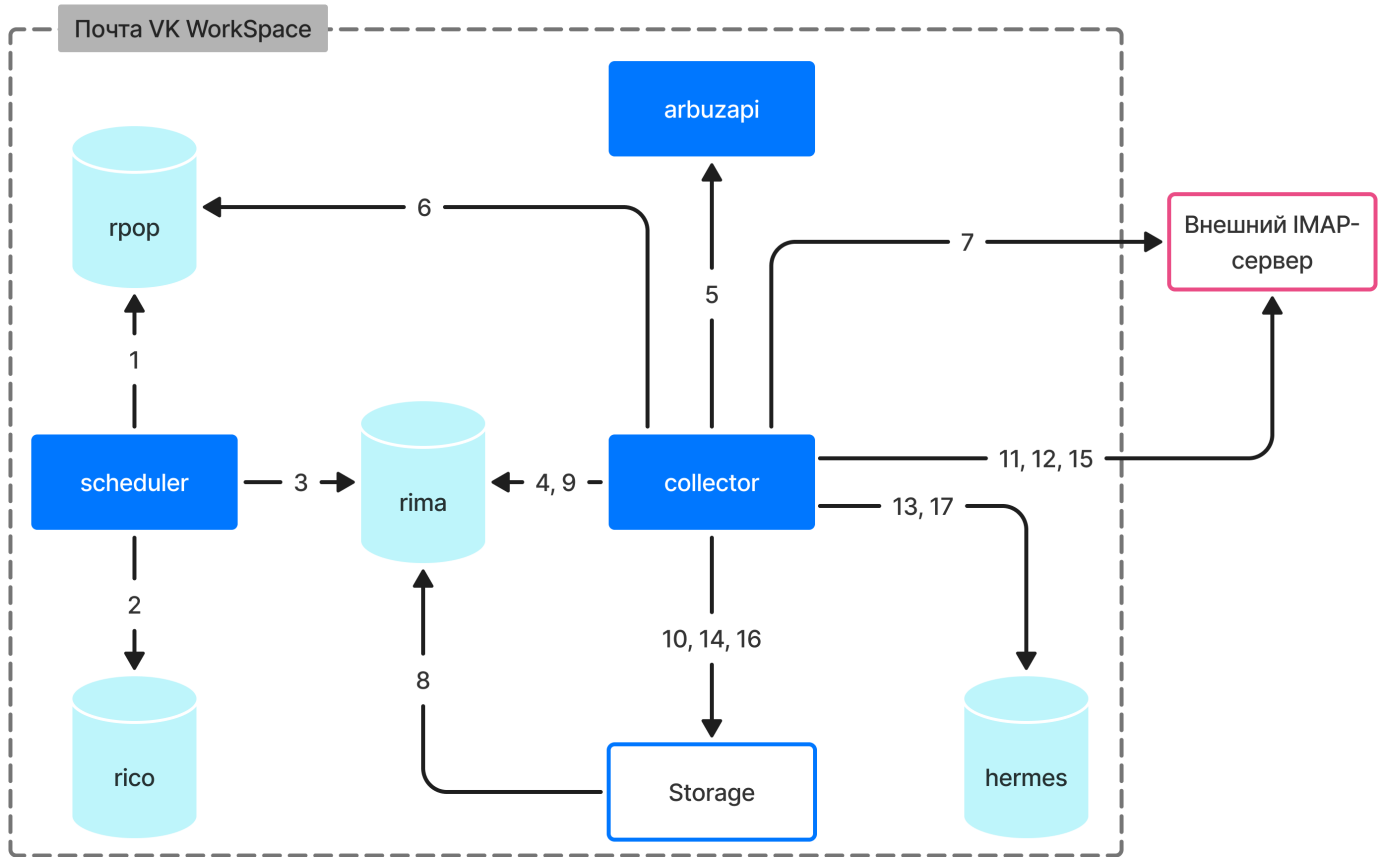
конкретному **instant** и отправляет ему оповещение, что необходимо собрать письма нового пользователя.

2. **Instant** забирает этого пользователя и проверяет в **rpop**, существует ли такой сборщик. Если сборщик отсутствует, вернется ошибка, и **instant** начнет обрабатывать следующий сборщик.
3. После успешной проверки сборщика необходимо установить соединение с IMAP через **conkee**. **Instant** (поток 6) отдает **conkee** команду на подключение к IMAP.
4. Когда соединение по IMAP установлено, **instant** проводит первую синхронизацию — выгружает из IMAP список папок, добавляет их абсолютные пути **hermes** и сохраняет в **storage**.
5. Вторым этапом необходимо выгрузить первые 100 писем из папки Inbox.
6. **Instant** так же обращается к **conkee** (поток 11), а тот, в свою очередь, выгружает письма из IMAP и передает их в **instant**.
7. Далее **instant** сохраняет письма в **storage** (поток 15) и обновляет в **hermes** downUID (сколько писем скачано).
8. После этого **instant** выставляет время последней сборки в **rico**.
9. При первой сборке **instant** также ставит задачу в **rma** на сборку писем из других папок с помощью **collector** с наивысшим приоритетом (1024).

Процесс сборки почты по расписанию

После создания и настройки сборщиков выполняется сборка по расписанию, для выполнения миграции основного объема писем.

Общая схема потоков данных при сборке почты по расписанию приведена на рисунке ниже:



Описание потоков данных приведено в таблице ниже:

Номер потока	Источник	Описание потока	Получатель
1	scheduler	Забирает список сборщиков	rrpop — БД MySQL, хранящая данные сборщиков
2	scheduler	Проверяет сроки сборки по ящикам	rico — БД Tarantool, хранит тайминги сборок
3	scheduler	Ставит задачи на сборку	rima — очередь задач для сборщиков почты
4	collector	Забирает задачи из очереди	rima
5	collector	Проверяет наличие пользователя (протокол gRPC).	arbutapi — gRPC-прокси, изолирующее MySQL базы данных
6	collector	Проверяет наличие сборщиков	rrpop — БД MySQL, хранящая данные сборщиков
7	collector		сервер IMAP

Номер потока	Источник	Описание потока	Получатель
		Авторизовывается (протокол IMAP)	
8	storage	Доносит нотификации со стороны VK WorkSpace.	rima
9	collector	Проверяет, не было ли новых изменений	rima
10	collector	Забирает новые письма из VK WorkSpace	storage — хранилище данных
11	collector	Доносит все изменения на внешний сервер	сервер IMAP
12	collector	Запрашивает изменения в папках с сервера IMAP (протокол IMAP)	сервер IMAP
13	collector	Маппинг папок	hermes — БД, в которой хранится состояние синхронизации по всем папкам всех сборщиков
14	collector	Сохраняет изменения в папках	storage
15	collector	Запрашивает письма с сервера IMAP	сервер IMAP
16	collector	Сохраняет новые письма	storage
17	collector	Обновляет маркеры скачивания писем (up/down UID)	hermes

Развернутое описание потоков данных:

1. Раз в 15 минут (периодичность настраивается в конфигурационном файле) **scheduler** обращается к **rpop** за списком ID сборщиков, которые делит на пакеты данных.
2. Далее выполняется проверка таймингов (когда последний раз была сборка по каждому из сборщиков) в **rico**.

3. После проверки таймингов **scheduler** исключает сборщики, по которым не пришло время ставить задачи. Затем ставит задачи в очередь **rima**.
4. **Collector** забирает задачи из очереди **rima** в свою оперативную память.
5. **Collector** запрашивает у **arbuzapi**, существует ли ещё пользователь.
6. **Collector** запрашивает у **rpop**, существует ли нужный сборщик. Если сборщик отсутствует, вернется ошибка, и **collector** обращается за следующим сборщиком в **rima**.
7. Также **collector** забирает у **rpop** пароль для авторизации в IMAP. С паролем **collector** пытается авторизоваться под пользователем по IMAP. Если авторизация прошла неуспешно, задача на сборку возвращается в **rima**. Следующая задача — донести до сервера IMAP изменения, которые произошли на стороне Почты VK WorkSpace.
8. Затем **storage** отправляет все нотификации (установка/снятие флагов, перемещение писем между папками и т.п.) в очередь **rima**.

 **Примечание**

Потоком 8 обозначено донесение изменений из Почты VK WorkSpace, так как это укладывается в логический порядок схемы (перед потоками 9 и 10). Изменения со стороны Почты из **storage** могут добавляться в очередь **rima** в любой момент времени.

9. **Collector** второй раз забирает задачи из **rima** на случай новых изменений.
10. После этого **collector** выполняет запрос в **storage** для сбора новых писем, созданных на стороне Почты VK WorkSpace с момента последней синхронизации.
11. Далее **collector** отправляет на сторону IMAP-сервера все изменения, которые произошли в Почте VK WorkSpace.

 **Примечание**

Если возникают какие-то конфликты, они решаются в пользу IMAP.

12. Далее **collector** выполняет запрос к IMAP-серверу для получения списка папок.
13. После получения списка папок **collector** обращается в **hermes** для маппинга, так как в IMAP папки определяются по абсолютному пути, а в **storage** используются ID. **Collector** обрабатывает список папок в **hermes** и сравнивает с тем, что получил из IMAP. Папки, которые есть в IMAP, но нет в Почте VK WorkSpace, добавляются, а те, которых нет в IMAP, но есть в Почте, удаляются.
14. После маппинга изменений, если они были, сохраняются в **storage**.

 **Примечание**

Если папка новая, **collector** отправляет запрос на получение максимального ID в **hermes** и присвоит новый ID со значением n+1. То же самое самое произойдет и при сохранении в **storage**.

15. После этого **collector** выгружает новые письма по всем папкам с сервера IMAP.
16. Далее **collector** сохраняет их в **storage**.

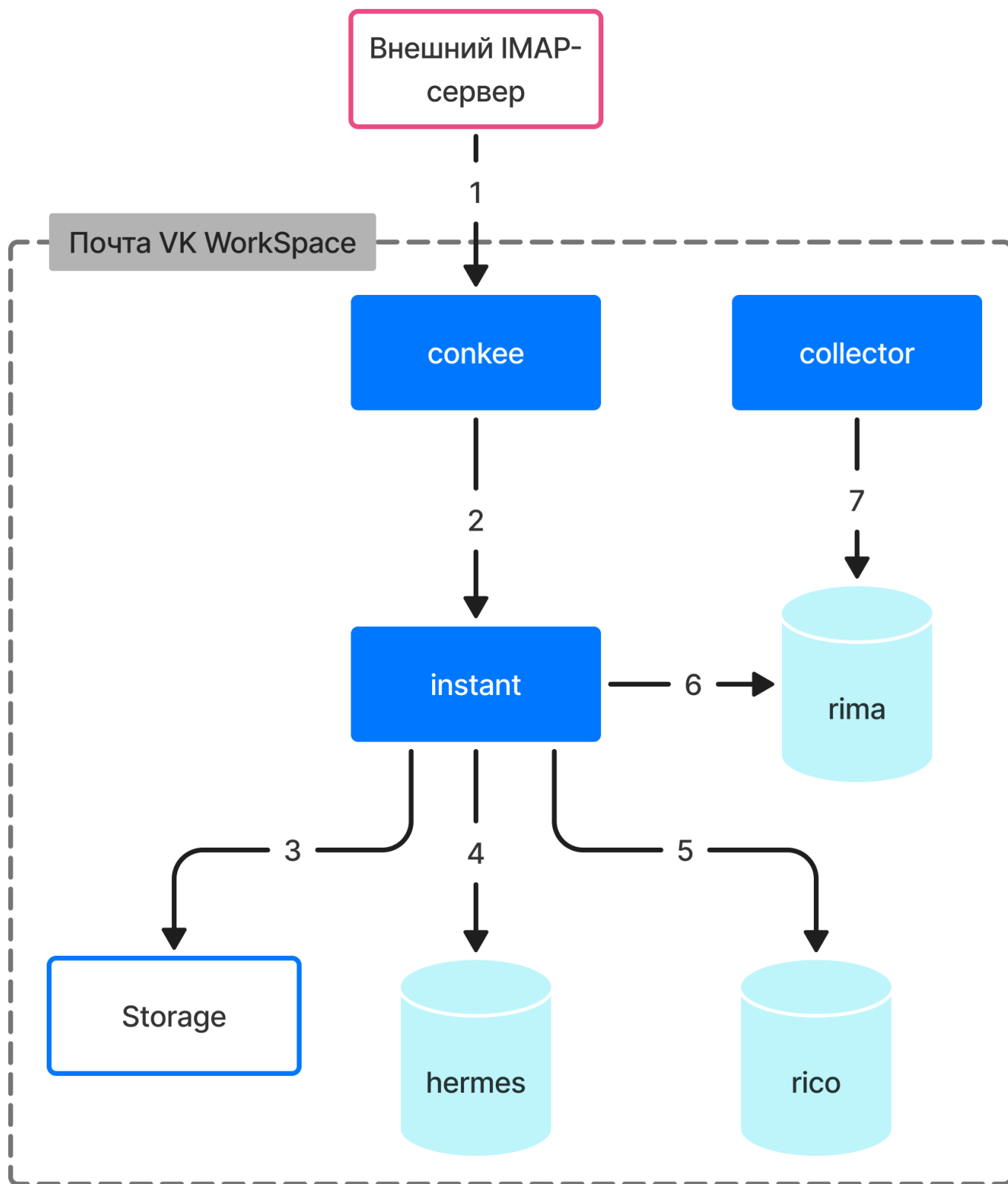
17. **Collector** отправляет данные для обновления маркеров скачивания писем (up/down UID) в **hermes**.

 **Примечание**

Scheduler работает в бесконечном цикле до момента отключения и полного перехода на мгновенную сборку.

Мгновенная сборка по IDLE

Общая схема потоков данных при сборке почты по IDLE приведена на рисунке ниже:



Описание потоков данных приведено в таблице ниже:

Номер потока	Источник	Описание потока	Получатель
1	сервер IMAP	Отправка пуш-уведомлений	conkee — прокси, держит непрерывное соединение с сервером IMAP

Номер потока	Источник	Описание потока	Получатель
2	conkee	Передача нового письма	instant — демон мгновенной сборки писем
3	instant	Сохранение письма	storage — хранилище данных
4	instant	Обновление состояния сборки для новых писем.	hermes — БД, в которой хранится состояние синхронизации по всем папкам всех сборщиков
5	instant	Передача времени последней сборки	rico — БД Tarantool, хранит тайминги сборок
6	instant	Отправка запроса на постановку задачи на сборку по расписанию	rima — очередь задач для сборщиков почты
7	collector	Поддержка постоянного подключения, для осуществления периодической сборки писем из очереди по расписанию	rima

После первой сборки **instant** через **conkee** держит постоянное соединение по IDLE и ждет новых писем. Как только новое письмо от IMAP поступает, **conkee** отправляет его в **instant**, а затем сохраняет в **storage**. Когда письмо сохранено, **instant** обновляет upUID (сколько писем скачано сверху) в **hermes**. Далее **instant** (поток 5) отправляет запрос в **rico** и обновляет время последней сборки из конкретного ящика.

Далее **instant** ставит задачи в **rima** на сборку с помощью **collector**, если:

- не успел что-то скачать;
- прервалось соединение по IMAP;
- существуют иные проблемы доступа к ящику.

Collector забирает задачи из **rima** и скачивает письма в зависимости от приоритета.

🕒 14 августа 2025 г.