

Мессенджер и ВКС

Архитектура и описание системы

Назначение документа	6
Дополнительная документация	6
Архитектура Системы	7
Инсталляция на одну виртуальную машину	7
Кластерная инсталляция	9
Возможные интеграции	13
Используемые токены и протоколы	15
Технические данные и требования	16
Как использовать системы виртуализации	17
Поддерживаемые операционные системы	18
Совместимость	18
Необходимые доступы и настройки	18
Мониторинг и журналирование	22
Резервное копирование	23
Базы данных	23
Сервисы Системы	25
Сервисы фронтенда	25
Front	25
Saws	25
Пользовательская сессия и события	26
Boss	26
Scribl	26
Feeddog	27
Cox	27
Cronos	28
Пуш-уведомления	29
Push_me_goiproto	29
Push_me_gosender	29
VKTPusher	29

Аутентификация, авторизация и антибрутфорс	30
Варианты доступа к Системе	30
Border	33
Keycloak	33
Nomail	34
Tokeeper	34
Fagus	34
Pocus	35
Focus	35
Janus	35
Vcl	36
Voice-relayicq	36
Recorder	36
Callevnts	37
Чаты, группы	37
Mchat-st	38
Gbld-st & Gbld-mchat	39
Drama	39
Prof-st	39
Nick	40
Sn-st	40
Chatexpo	40
Icon	41
Badger	41
Fate	42
Papker	42
Planer	43
Треды (обсуждения)	44
Bobbin	44
Flux	44
Сервисы для работы с прикрепленными файлами, аватарами и стикерами	45
Go-files	45

Krueger	45
Lincoln	45
Store-httpd	46
Store-mq	46
Статусы	46
Gradus	46
Голосования	46
Polls-api	46
Polls-updater	46
Опросы	47
Survey	47
Spamy	48
Поиск	48
Beagle	48
Levitan	49
Cmini	49
Jimny	49
Mini	50
Задачи	50
Blitz	50
Структура организаций	51
Staffd	51
Мини-аппы	52
Giraffe	52
Avon	52
Whoost	53
Approach	53
Gulliver	54
Zelda	54
Godmod	55
Боты	55
Stickers_bot	55

Botex	55
Syneva	55
Everest	56
Интеграция с VK Workmail	56
Stentor	56
Mecca	56
Skywalker	57
Интеграция с антивирусом	57
Kowalski	57
Интеграция с MAX	58
Pigeon	58
Интеграция с DLP-системами SearchInform и InfoWatch (до версии 25.2)	59
Plumber	59
Resort	59
Интеграция с DLP-системами (начиная с версии 25.2)	60
Vahter	61
Multifora	61
Tourniquet	62
Watchman	62
Мониторинг	62
VictoriaMetrics	62
Grafana	63
Групповые политики	64
Zastava	64
Vipper	65
Chat-gp	65
Pacman	65
Сервисы инфраструктуры	66

Назначение документа

В документе описана архитектура Мессенджера.

Документ предназначен для использования следующими специалистами:

- системные администраторы;
- специалистами по информационной безопасности.

Дополнительная документация

[Инструкция по настройке Single Sign-On аутентификации](#) — в документе представлена информация о сервисах Мессенджер и ВКС, обеспечивающих Single Sign-On аутентификацию.

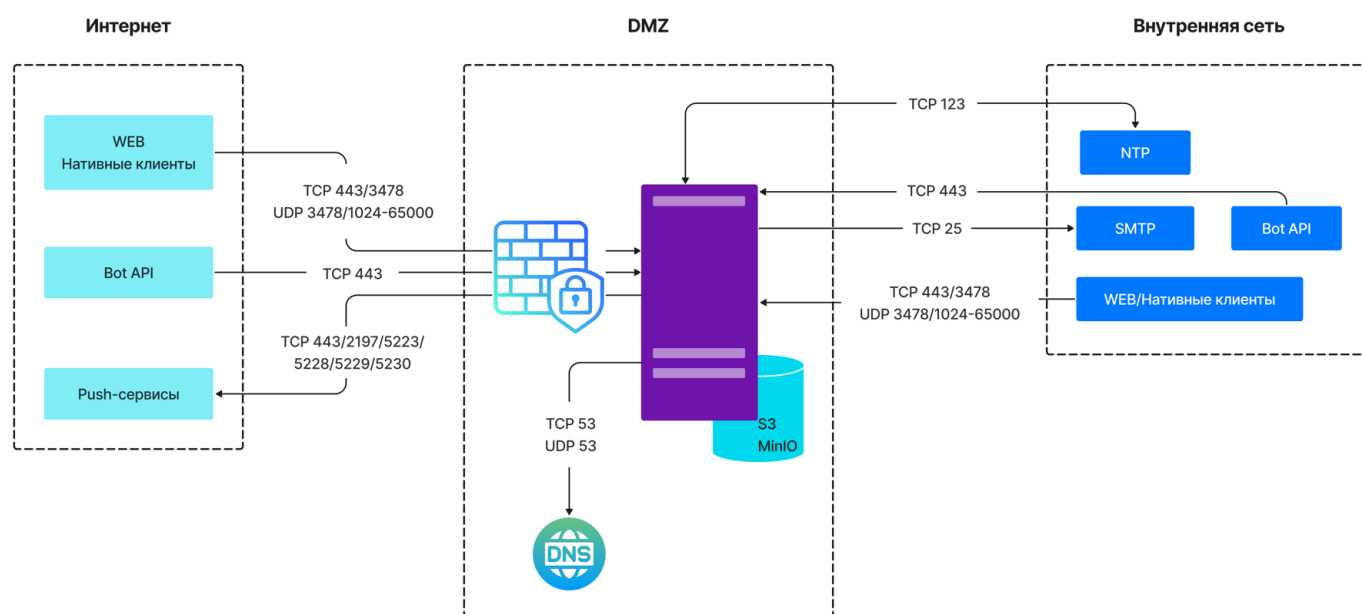
Архитектура Системы

Мессенджер (далее – Система) может быть установлен как на площадке заказчика, так и в облачном сервисе VK Cloud Solutions.

Систему можно установить как в полностью закрытый контур, отрезанный от интернета, чтобы мессенджер работал только внутри локальной сети, так и в контур с возможностью работы из внешних сетей.

Инсталляция на одну виртуальную машину

На схеме ниже представлена схема инсталляции на одну виртуальную машину в периметре организации заказчика:



Хранилище S3

Внутреннее хранилище вложений чатов. MinIO — хранилище, совместимое с S3 API.

По умолчанию в составе дистрибутива мессенджера представлен MinIO. MinIO не масштабируется и не резервируется. При необходимости кластера MinIO он может быть поставлен отдельно и требует отдельных аппаратных ресурсов.

NTP-серверы

Используется для синхронизации времени. Возможно использование внешних серверов, если нет сложностей с прохождением сетевых фильтров.

SMTP-сервер

Для аутентификации в мессенджере используется OTP via Email. Необходим SMTP-сервер, на котором разрешена отправка почтовых сообщений для данной виртуальной машины.

WEB/Нативные клиенты

Приложение VK WorkSpace, установленное на одной из допустимых платформ. Сервер Мессенджер принимает входящие сообщения от этого приложения, а также отправляет ответы. Основное взаимодействие осуществляется через протокол HTTPS (443/TCP).

Для работы звонков используются следующие порты:

- 3478 TCP и UDP — входящие соединения для протоколов STUN и TURN в р2р звонках;
- 1024+ UDP — входящие и исходящие соединения для RTP трафика в групповых звонках

Bot API

Приложение VK WorkSpace поддерживает реализацию IM-ботов посредством Bot API. Документация по Bot API расположена по ссылке: <https://myteam.mail.ru/botapi/>.

Push сервисы

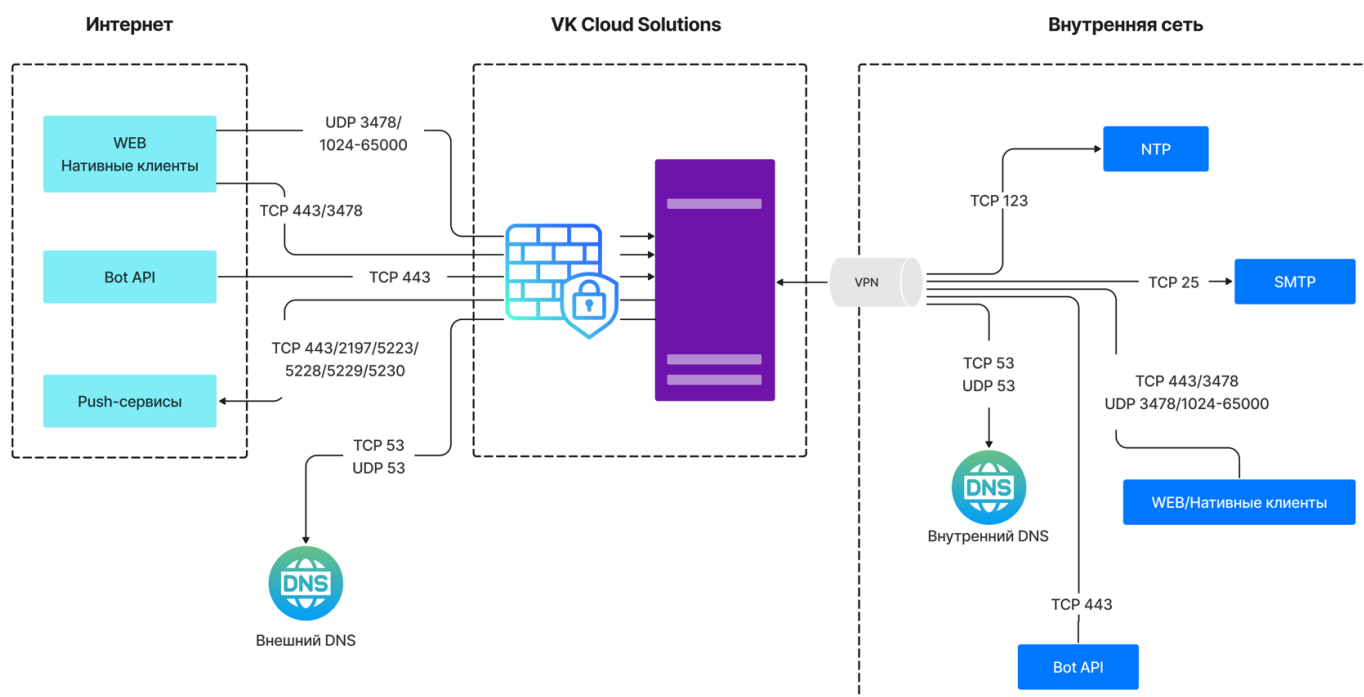
Внешние сервисы Apple и Google для отправки push-сообщений на мобильные платформы. Серверы Мессенджера, требуются исходящие соединения к этим сервисам и не требуется входящих соединений.

Обеспечение отказоустойчивости

Отказоустойчивость обеспечивается средствами виртуализации Заказчика.

Инсталляция в облаке

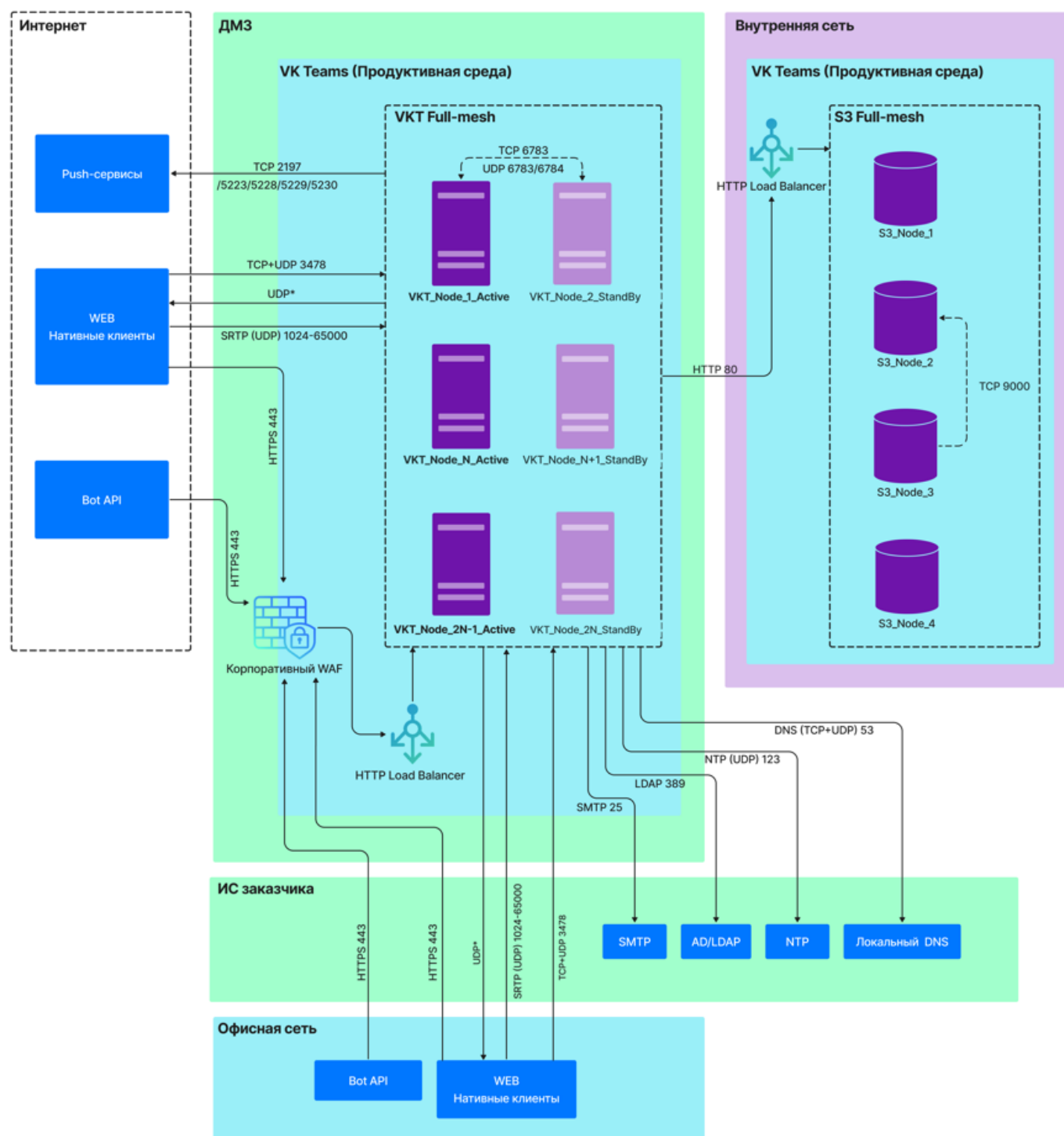
На схеме ниже представлена инсталляция Мессенджера в облачном сервисе (на примере VK Cloud Solutions):



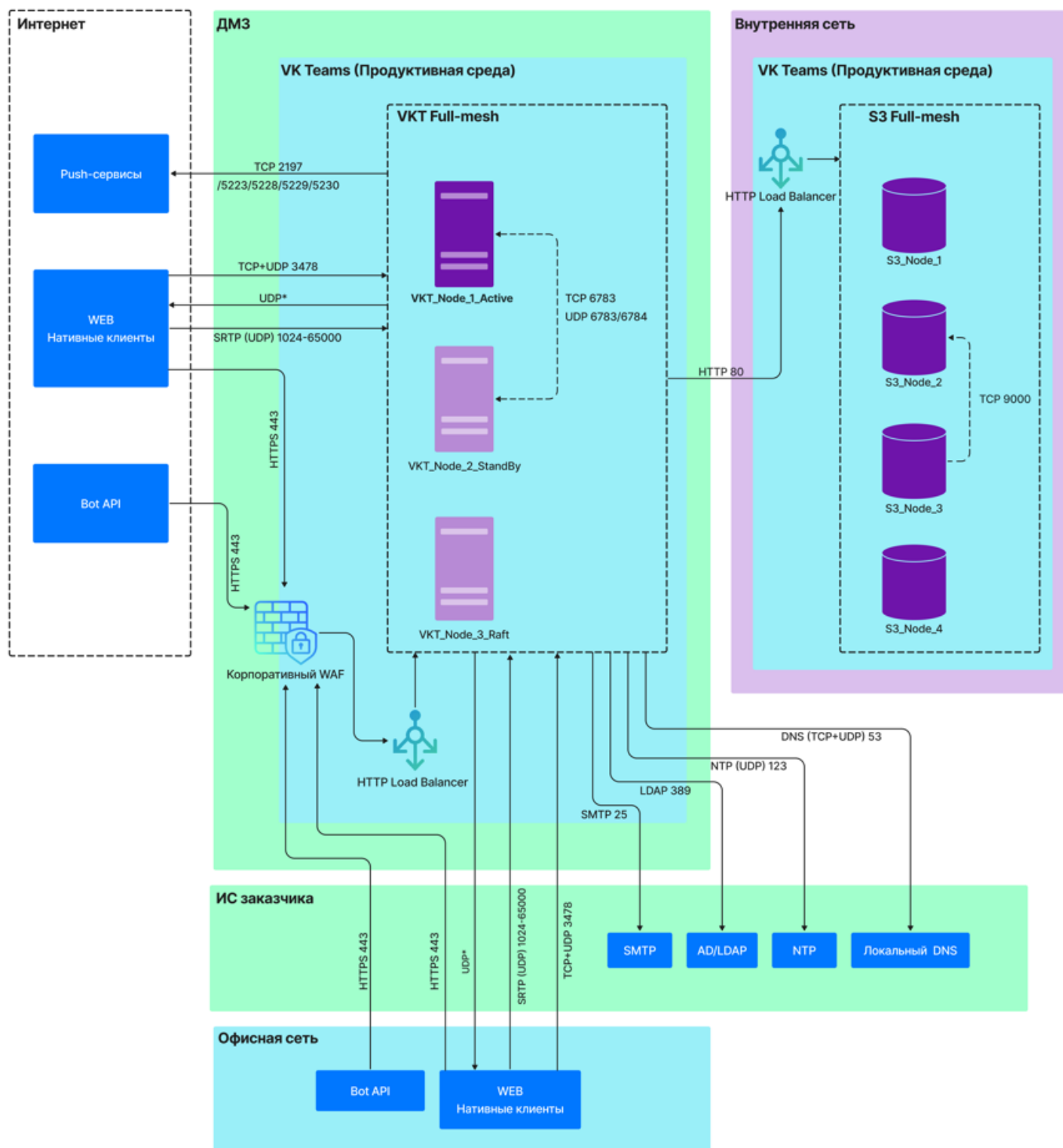
Кластерная инсталляция

Кластерная инсталляция предусматривает возможность реализации отказоустойчивой конфигурации. Система может быть горизонтально масштабирована.

На схеме ниже представлен способ поставки Мессенджера в масштабируемом, отказоустойчивом варианте. Минимальный размер кластера — 2 шарда (4 вычислительных узла), максимальный не ограничен.



На схеме ниже представлена одношардовая горизонтально не масштабируемая отказоустойчивая конфигурация.



На схеме в качестве примера приведен кластер S3, поставляемый продуктом (реализуется на базе MinIO). Также может быть использован любой S3 API совместимый продукт заказчика (например, VKCS S3).

Full-mesh — полносвязанная сетевая топология, в которой каждый вычислительный узел связан со всеми остальными по портам:

- в кластере VKT — TCP 6783, UDP 6783/ 6784;
- в кластере S3 — TCP 9000.

На границах межсетевых зон по решению заказчика могут быть расположены TCP/UDP Firewall. Корпоративный WAF устанавливается по желанию заказчика.

В случае использования MinIO в качестве кластера S3, необходимо дополнительное балансирование трафика от кластера VKT между всеми узлами данного кластера (HTTP Load Balancer на диаграмме). Данный балансировщик должен быть настроен в соответствии с политикой Least Connections, и иметь DNS-имя. Подробнее см. <https://min.io/docs/minio/linux/operations/install-deploy-manage/expand-minio-deployment.html#networking-and-firewalls/>.

Так как продукт использует Virtual-hosted style для доступа к S3 API, то также необходимо добавить Wildcard DNS-запись для S3-сервера, например, **CNAME *.some-s3-server.com → some-s3-server.com**.

Решения относятся к классу автономных и разворачиваются на базе предоставленных Заказчиком вычислительных ресурсов (виртуальных машин) в виде цельных образов виртуальных машин (virtual-appliance).

На текущий момент не предполагается, что клиентская часть и серверная часть (сервисы бизнес-логики, сервисы БД, т.п.) могут быть разнесены на различные узлы.

Решения поставляются в виде OVA и/или Qcow2 образов ВМ.

Ландшафт развертывания

В рамках проекта может быть использован двухуровневый ландшафт развертывания Системы:

- среда тестирования/пред-продуктивная среда;
- продуктивная среда.

Пред-продуктивная среда разворачивается аналогично продуктивной среде, требования к оборудованию пред-продуктивной среды совпадают с таковыми для продуктивной среды за возможным уменьшением горизонтального масштаба (числа шардов).

Обеспечение отказоустойчивости

Отказоустойчивость обеспечивается по схеме 2N, где N – число шардов (число единиц горизонтально-масштабированной поставки), что обусловлено внутренней архитектурой решения. При N=1 необходимое число узлов для обеспечения отказоустойчивости — 3 (вырожденный случай).

Основные узлы кластера условно делятся на два вида:

- обеспечивающие обработку рабочей нагрузки (Active);
- обеспечивающие резервный вычислительный ресурс для первых (Stand-by).

При N=1 также имеет место дополнительный Raft узел, обеспечивающий функционирование raft-сервисов. Для N>1 raft-мастеры располагаются на любых трех вычислительных узлах кластера.

Перехват управления доступа (клиентского трафика) между Active и Stand-by узлами при отказе Active-узла обеспечивается автоматически на стороне клиента.

Отказоустойчивость каждого компонента в рамках одного дата-центра, обеспечивается в автоматическом режиме (только сам failover). Восстановление части компонентов после аварии и переключения может потребовать ручного вмешательства, согласно инструкции.

В случае резервирования сервиса в нескольких дата-центрах переключение между ДЦ производится в ручном режиме, силами Заказчика, согласно инструкции.

Установка обновлений

Установка обновлений должна происходить последовательно – по очереди обновляется каждая виртуальная машина кластера. При таком подходе кластер не обязательно останавливать, минимизируется риск потерь и дополнительных используемых ресурсов.

Установка образов виртуальных машин обновлённого продуктового релиза на узлы кластера производится силами Заказчика.

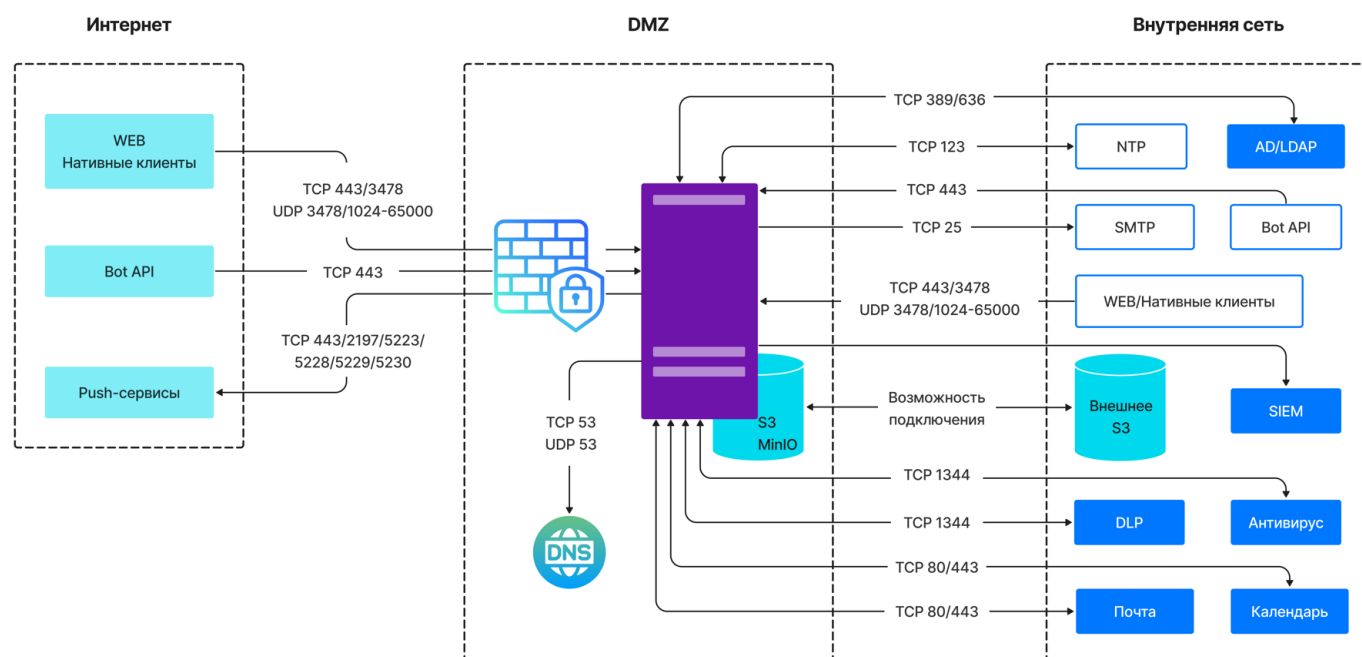
Настройка, интеграция виртуальных машин с предустановленными продуктовыми образами, миграция данных при обновлении в течение проекта лежит в зоне ответственности Подрядчика.

Процесс установки обновлений (для каждого узла кластера):

1. Снятие нагрузки с обновляемой ВМ;
2. Резервирование данных с ВМ;
3. Подготовка к обновлению ВМ;
4. Остановка ВМ;
5. Установка обновленного образа ВМ, предоставляемого ВК;
6. Запуск ВМ;
7. Настройка ВМ после установки «чистого» продуктового образа ВМ:
 - a. Конфигурация;
 - b. Инсталляция агентов мониторинга и резервирования;
8. Возобновление нагрузки на обновленную ВМ.

Возможные интеграции

На схеме ниже представлены возможные интеграции с Мессенджер и ВКС:



Антивирус

Используются внешние решения (например, Dr.Web, Kaspersky) с целью экономии ресурсов клиента. Интеграция с антивирусом возможна по протоколу ICAP (подробнее см. описание сервиса [Kowalski](#)).

SIEM

Система управления информационной безопасностью и событиями безопасности. Сервисы Мессенджер и ВКС логируют события безопасности, которые могут быть интегрированы с корпоративной подсистемой SIEM по протоколу syslog. Для интеграции с SIEM используется мультиплатформенный инструмент сбора и централизации журналов NXLog (подробнее см. документ «Руководство по администрированию»).

DLP-системы

Система для предотвращения утечки данных. Интеграция с DLP-системами возможна по протоколу ICAP. В DLP-систему отправляется переписка, события ботов и вложенные файлы (подробнее см. описание сервиса [Plumber](#)).

LDAP, ActiveDirectory

Система работает со службами каталогов по протоколу LDAP, клиент предоставляет только путь и пароль на чтение. Из инфраструктуры клиента в Систему выгружаются и сохраняются пользователи, попадающие под фильтр запроса. В дополнение к данным из AD в мессенджере можно вручную создавать пользователей, используя YML-файл либо веб-интерфейс сервиса Keycloak.

Необходимо предоставить доступ к LDAP-серверу, а также настройки для соединения с LDAP-сервером: bind_dn, user_dn, URL, password, CA-сертификат. Возможна работа без LDAP, с заведением пользователей внутри Системы.

Внешнее хранилище S3

Система предоставляет возможность использовать хранилище S3 заказчика при его наличии, а также хранить файлы большого размера в хранилище S3 VK WorkMail.

WAF (Web application firewall)

Реализуется посредством корпоративной инфраструктуры Заказчика. Осуществляет фильтрацию входящего HTTP трафика, а также акселерацию SSL трафика.

Почта /Календарь

При наличии инсталляции VK WorkMail.

Интеграция с внешним Push-провайдером

На текущий момент отправка push-уведомлений поддерживается только для платформ iOS и Android. В остальных случаях предусмотрено получение уведомлений клиентскими инсталляциями посредством технологии long-polling.

Используемые токены и протоколы

Использование авторизационных токенов

Система поддерживает следующие способы авторизации:

- по aimsid (идентификатор сессии) — для авторизации пользователей;
- по prism-токену/ k-служебному trusted-токену (akes) — доверенный токен для внутренних коммуникаций.

Использование проприетарных протоколов в системе

- IPROS — бинарный протокол асинхронного обмена сообщениями между сервисами Мессенджер и ВКС. Почти все взаимодействие сервисов построено на нем. В него встроены механизм обнаружения сервисов и средства для управления инсталляцией;
- Iproto — протокол обмена сообщениями между сервисами Мессенджер и ВКС. Не имеет системы адресации. Используется также в БД Tarantool.

Использование глобальных стандартов и протоколов

Для организации аудио- и видеозвонков используется технология WebRTC.

Сжатие аудио — кодек Opus. Сжатие видео — кодек H.264.

Передача медиапотокa осуществляется с помощью протокола SRTP (TLS 1.2/1.3).

Поддерживаются следующие опции шифрования **звонков**:

- AES128_CM_SHA1_80.
- AES128_CM_SHA1_32.
- AEAD_AES_256_GCM.
- AEAD_AES_128_GCM.

Месседжинг — протокол TLS 1.2/1.3.

Поддерживаются следующие опции шифрования **месседжинга**:

- ECDHE-ECDSA-AES128-GCM-SHA256.
- ECDHE-RSA-AES128-GCM-SHA256.
- ECDHE-ECDSA-AES256-GCM-SHA384.
- ECDHE-RSA-AES256-GCM-SHA384.
- ECDHE-ECDSA-CHACHA20-POLY1305.
- ECDHE-RSA-CHACHA20-POLY1305.
- DHE-RSA-AES128-GCM-SHA256.
- DHE-RSA-AES256-GCM-SHA384.

Технические данные и требования

В случае распределенной инсталляции требования к вычислительным ресурсам (виртуальным машинам) для продуктивной среды рассчитываются индивидуально для Заказчика. Свяжитесь с представителями VK для помощи с расчетом сайзинга.

Требования к standalone-инсталляции представлены ниже.

Для виртуальной машины под регистри и установщик VK WorkSpace

Минимальный рекомендуемый объем памяти SSD — 300 ГБ.

vCPU — 4.

RAM — 4 ГБ.

Для виртуальной машины под Мессенджер

Минимальные требования к серверу в зависимости от количества пользователей:

Количество пользователей	vCPU	RAM, GB	SSD, GB	S3, GB / год
Тестовая установка				
1 000	22	56*	400 root 150 GB data 250 GB	-
Продуктивная установка				
От 1 до 2 000	22	56*	400: root 150 GB data 250 GB	500

* — указано минимальное количество доступной памяти. Для функционирования инсталляции общей памяти на инстансе должно быть больше этого значения минимум на 7%

Минимальные требования к SSD-дискам:

Операций чтения, IOPS	Операций записи, IOPS	Максимальная задержка операции, мс
10 000	5 000	< 1

Для корректной работы сервиса Федерация необходимо добавить на серверы каждой инсталляции дополнительные вычислительные ресурсы:

- 10% от имеющихся мощностей vCPU
- 1 ГБ SSD
- 2 ГБ RAM на каждую тысячу федеративных пользователей. Ожидаемый прирост RAM — около 1 ГБ в год в зависимости от количества сообщений федерации

Как использовать системы виртуализации

vCPU

Обязательная поддержка Time Stamp Counter (TSC). Проверить наличие можно поиском флага `constant_tsc` в `/proc/cpuinfo`. Любой современный процессор поддерживает эту технологию, однако иногда этого регистра нет внутри виртуальной машины. В этом случае необходимо правильно настроить систему виртуализации.

Не допускайте переподписку. Суммарные vCPU на хосте не должны превышать количество физических ядер, выделенных всем виртуальным машинам. При этом не рекомендуется считать Hyper-Threading полноценными ядрами.

Не выделяйте одной виртуальной машине количество ядер больше, чем количество ядер на физическом сокете.

RAM

Не назначайте суммарную vRAM выше физической RAM хоста.

Механизмы экономии памяти

Не включайте механизмы ballooning и сжатия памяти.

swap

Не используйте swap — как на гипервизоре, так и внутри виртуальных машин.

Резервирования ресурсов виртуальных машин

Устанавливайте всю выделенную память и процессоры в резерв для виртуальных машин системы.

Хранилище

Не используйте тонкие диски (диски типа Thin) — диски с отложенным выделением пространства на СХД.

Входящий трафик

TCP — 10 Мбит/с; UDP — 10 Мбит/с.

Поддерживаемые операционные системы

На виртуальных машинах под Мессенджер должна быть установлена РЕД ОС — версия 7.3с, версия ядра — не выше 6.1.162.

Требований к операционной системе для виртуальной машины под регистри/установщик не предъявляется.

Совместимость

- ПО VMware версий 6.x – 7.
- Любые системы виртуализации, основанные на KVM, например OpenStack.
- VK Cloud Solutions.

Необходимые доступы и настройки

Следующие подсистемы должны быть реализованы посредством корпоративной инфраструктуры Заказчика:

- HTTP LoadBalancer.
- МСЭ — межсетевой экран.
- Корпоративный DNS.
- Корпоративный NTP.
- Корпоративный SMTP.

Также могут быть реализованы при необходимости:

- Корпоративный AD - Active Directory.
- Корпоративный WAF.

Доступы к данным ресурсам должны быть открыты.

Внешний IPv4 адрес

Адрес может быть поднят как внутри виртуальной машины, так и проброшен через NAT. Преобразование сетевых адресов (NAT) должно быть вида 1-в-1 (сеть в сеть), то есть с сохранением номера порта. Иначе видео- и голосовые звонки могут не работать.

Роутинг исходящих соединений

Необходим для отправки push-сообщений (через сервисы Apple, Google) и для работы голосовых и видеозвонков.

Настройки DNS-зоны

Завести в DNS-зоне имена хостов, которые будут смотреть на внешний IP, выделенный в рамках инсталляции.

Доменные имена:

Тип записи	Запись
A	vkteams.mycorp.com 10.20.30.40
или:	
CNAME	*.vkteams.mycorp.com vkteams.mycorp.com
CNAME	admin.vkteams.mycorp.com vkteams.mycorp.com
CNAME	api.vkteams.mycorp.com vkteams.mycorp.com
CNAME	biz.vkteams.mycorp.com vkteams.mycorp.com
CNAME	calendar.vkteams.mycorp.com vkteams.mycorp.com
CNAME	call.vkteams.mycorp.com vkteams.mycorp.com
CNAME	для версий до 25.2 — di.vkteams.mycorp.com vkteams.mycorp.com для версий 25.2 и выше — kc.vkteams.mycorp.com vkteams.mycorp.com
CNAME	dl.vkteams.mycorp.com vkteams.mycorp.com
CNAME	files-b.vkteams.mycorp.com vkteams.mycorp.com
CNAME	mobile-calendar.vkteams.mycorp.com vkteams.mycorp.com
CNAME	s.vkteams.mycorp.com vkteams.mycorp.com
CNAME	stentor.vkteams.mycorp.com vkteams.mycorp.com
CNAME	u.vkteams.mycorp.com vkteams.mycorp.com
CNAME	ub.vkteams.mycorp.com vkteams.mycorp.com
CNAME	webim.vkteams.mycorp.com vkteams.mycorp.com

SSL-сертификат

Предварительно на сервер должен быть установлен SSL-сертификат. Для SSL/TLS необходимы fullchain-сертификаты для доменных имен, указанных выше.

Входящие соединения на стороне сервера Мессенджер и ВКС

Открыть порты: 80/TCP, 443/TCP, 3478/TCP + UDP, UDP-порты выше 1024.

Исходящие соединения на стороне сервера Мессенджер и ВКС

Открыть доступ для серверов отправки уведомлений: необходимо обеспечить доступ к серверам Google и Apple для отправки и корректной работы push-уведомлений на мобильных платформах Android и iOS.

	Apple iOS	Google Android
Ссылка на официальную документацию	https://support.apple.com/en-us/HT203609	https://firebase.google.com/docs/cloud-messaging/concept-options#messaging-ports-and-your-firewall
Порты	TCP 5223 для связи с APNs TCP 443, TCP 2197 для отправки уведомлений APNs	TCP 5228 TCP 5229 TCP 5230 TCP 443
IP-адреса или доменные имена	Преимущественно необходимо открыть исходящий доступ к подсети 17.0.0.0/8, более точно: <u>IPv4:</u> 17.249.0.0/16 17.252.0.0/16 17.57.144.0/22 17.188.128.0/18 17.188.20.0/23 <u>IPv6:</u> 2620:149:a44::/48 2403:300:a42::/48 2403:300:a51::/48 2a01:b740:a42::/48	Необходимо открыть исходящий доступ к нижеследующим доменным именам (могут меняться с течением времени): mtalk.google.com mtalk4.google.com mtalk-staging.google.com mtalk-dev.google.com alt1-mtalk.google.com alt2-mtalk.google.com alt3-mtalk.google.com alt4-mtalk.google.com alt5-mtalk.google.com alt6-mtalk.google.com alt7-mtalk.google.com alt8-mtalk.google.com android.apis.google.com device-provisioning.googleapis.com firebaseinstallations.googleapis.com Альтернативно возможно открытие доступа к конкретным IP-адресам, однако их список

	Apple iOS	Google Android
		намного шире и может меняться с течением времени: https://www.gstatic.com/ipranges/goog.json

Исходящие соединения на стороне клиента

Разрешить подключение: 80/TCP, 443/TCP, 3478/TCP + UDP, UDP-порты выше 1024.

Мониторинг и журналирование

Мониторинг бизнес-метрик сервисов решения

Высокоуровневые метрики сервисов решения могут отслеживаться администратором на дашбордах встроенного GUI Grafana (подробнее см. описание сервиса [Grafana](#)).

Мониторинг состояния сервисов

Часть базовых метрик, отвечающих за состояние инфраструктурных сервисов, могут быть получены из продуктового сервиса [VictoriaMetrics](#) (Kubernetes, Docker, отдельные демоны некоторых сервисов, ОС).

Остальные метрики (health, некоторые низкоуровневые бизнес-метрики) могут быть получены из вывода python скрипта:

```
/usr/share/check-mk-agent/local/local_check_exec.py
```

(более подробно см. документ [Руководство по администрированию](#)).

Мониторинг состояния оборудования

Низкоуровневые метрики состояния виртуальных машин, а также стороннего используемого прикладного ПО контролируется подсистемой мониторинга Заказчика самостоятельно (более подробно см. документ [Руководство по администрированию](#)).

Резервное копирование

При инсталляции на одну виртуальную машину предусматриваются следующие способы обеспечения резервирования:

- Корпоративная подсистема резервирования Заказчика;
- Резервирование на уровне образов виртуальных машин при помощи систем виртуализации, например, VMware and Veeam.

Для кластерной установки Системы предлагается совместное использование обеих рассматриваемых ниже методик резервирования:

- Резервирование дампов баз данных;
- Резервирование цельных снимков виртуальных машин.

Частота и время запуска процесса определяется Заказчиком самостоятельно. Рекомендуется исходить из анализа востребованности ресурсов корпоративного мессенджера внутрикорпоративными пользователями.

Базы данных

Используемые СУБД хранят данные непосредственно на локальных виртуальных машинах, за исключением данных вложений сообщений (прикрепляемые файлы – фотографии, видео, документы и т.п.). Последние хранятся в подсистеме S3, физически отделённой от остального кластера.

В качестве подсистемы S3 в контуре заказчика может быть развёрнуто OpenSource решение с минимально необходимым набором функциональности, апробированное продуктовой командой. Альтернативно Мессенджер и ВКС может интегрироваться с S3-совместимой подсистемой, например, VK Cloud S3.

Физическое вынесение хранимых данных во внешнюю систему хранения данных на текущий момент не представляется возможным.

Мессенджер и ВКС использует собственные и сторонние БД.

Собственные БД:

1. Tarantool. В БД Tarantool хранятся все основные данные: профили пользователей, сессии авторизации, внутренние данные по Системе и работе компонентов.
2. Kust. БД Kust используется для хранения чатов.

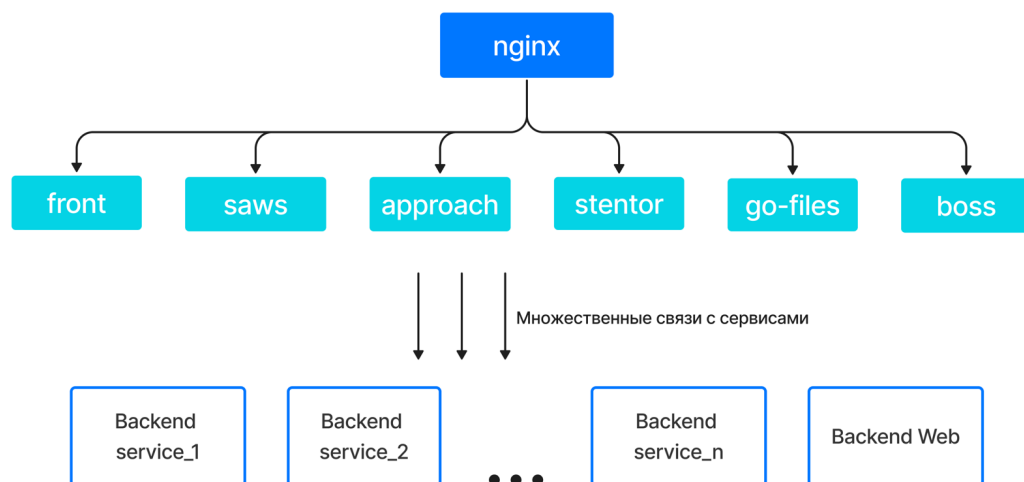
Сторонние БД:

1. ClickHouse — система сбора и хранения статистики.
2. Etcd — KV хранилище глобальных настроек.
3. MySQL-files — БД для сервиса Go-files (хранение файлов, в т.ч. стикеров).
4. MySQL-keycloak — БД для сервиса Keycloak.
5. MySQL-store — БД для сервиса Store (хранение стикеров, витрины стикеров).
6. Vault — защищенное хранилище секретов.
7. Consul — KV хранилище.

В базах данных MySQL хранятся внутренние технические данные.

Сервисы Системы

Все входные запросы пользователей поступают в балансировщик и перенаправляется на сервис Nginx, который, в свою очередь, перенаправляет запрос на нужный сервис.



Описание сервисов Мессенджер и ВКС представлено ниже.

Сервисы фронтенда

Front

Сервис является входной точкой в кластер для интерфейса `u.<DOMAIN>`. Перед сервисом Front установлен сервис Nginx для терминирования HTTPS-запросов.

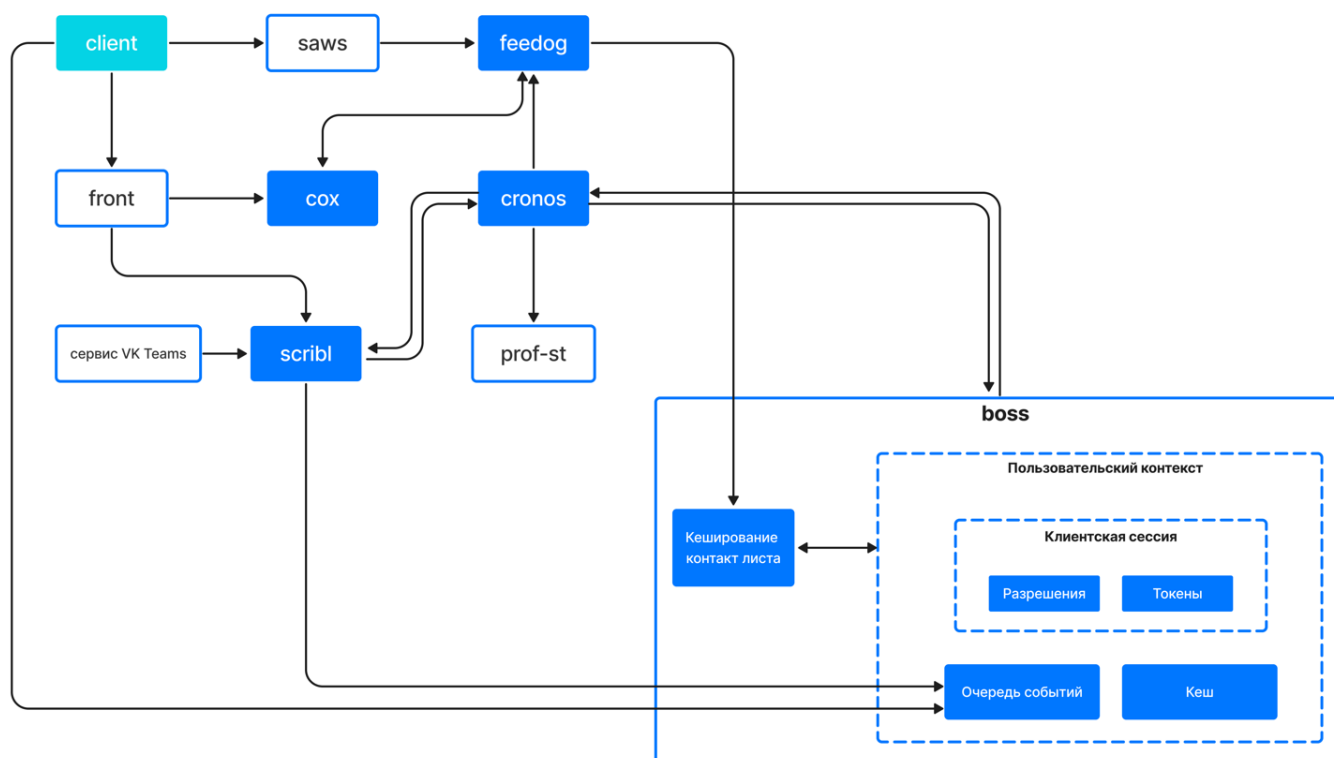
Взаимодействует со многими сервисами кластера, которые обрабатывают запрос. Выполняет различные проверки входных и выходных данных. Авторизует запросы пользователей.

Saws

Принимает HTTP-запросы пользователей с домена `u.<USER_DOMAIN>`, самые значимые из которых: аутентификация в Системе, начало сессии мессенджера и отправка сообщения, а также запросы работы с контакт-листом, внутренние запросы от веб части.

Сервис Saws взаимодействует со многими сервисами кластера. Перед Saws установлен сервис Nginx, терминирующий HTTPS и балансирующий нагрузку.

Пользовательская сессия и события



Boss

Хранит пользовательские сессии в оперативной памяти. Хранит очередь событий и отдает их подключенным клиентам. Хранит токены для отправки push-сообщений. Кеширует контакт лист пользователей.

Все данные хранятся в оперативной памяти, но может восстанавливать сессию из сервиса Feeddog.

Scribl

Универсальный сервис подписок. Данные о подписках хранит, используя БД Tarantool.

Сервис Front по протоколу IPROS инициирует подписку в сервисе Scribl. Сервис Scribl в соответствующем сервисе получает данные о текущем состоянии объекта, на который выполняется подписка, создает в БД Tarantool запись (подписку) и отправляет состояние в сервис Boss.

Когда клиента нужно оповестить о событии, случившемся по подписке, сторонний сервис передает событие в сервис Scribl, сервис Scribl создает событие в сервисе Boss.

Время жизни каждого типа подписок регулируется конфигурационным файлом. Также есть eternal подписки — они существуют от начала сессии в сервисе Saws до ее завершения в сервисе Boss.

Получает запросы от следующих сервисов:

- Front — информация о пользователе;
- Prof-st — информация о пользователе, изменение должности;

- Saws;
- Boss;
- Gradus — статус пользователя;
- Kowalski — статус проверки файла на вирусы;
- Focus — информация о звонке в чате;
- Icon — аватар пользователя или чата;
- Gbld-mchat — обновление состояния обсуждения, обновление состояния диалога;
- Gbld-st — обновление состояния диалога;
- Bobbin — обновление состояния обсуждения;
- Blitz — информация о состоянии задачи, информация об изменении списка задач, информация об изменении задач, удовлетворяющих фильтру, количество непрочитанных задач;
- Месса — количество непрочитанных писем;
- Skywalker — статус файла в облаке (сохранен или нет).

Feedog

Работа с контакт-листом пользователей и адресными книгами:

- добавление, обновление, удаление контактов;
- добавление телефонных книг;
- хранение пользовательских сессий;
- хранение привязанного номера телефона;
- хранение идентификатора аватара пользователя;
- хранение статистики по пользователю (время регистрации, количество сообщений).

Получает запросы от следующих сервисов:

- Boss — для уведомления об изменении контакт-листа;
- Сох — для сохранения изменений в контакт-листе;
- Mchat-st — для удаления из участников чата при удалении чата из контакт-листа;
- Gbld-mchat — для миграции истории при перемещении аккаунта.

Отправляет запросы в следующие сервисы:

- Boss — для получения и изменения контакт-листа, сохранения сессий;
- Saws — для добавления, обновления, удаления контактов, добавления телефонных книг.

Данные хранит, используя сервис Сох.

Сох

Хранилище контакт-листов пользователей и адресных книг. Данные хранит, используя БД KUST.

Отправляет запросы в следующие сервисы:

- Stdб;
- Prof-st;
- Sn-st;
- Feedog.

Получает запросы от следующих сервисов:

- Boss;
- Chatexpo;
- Feedog;
- Front;
- Jimny;
- Mchat-st;
- Mini;
- Prof-st;
- Gbld-st.

Cronos

Сервис получает состояние пользователя и хранит время, когда пользователь в последний раз был online.

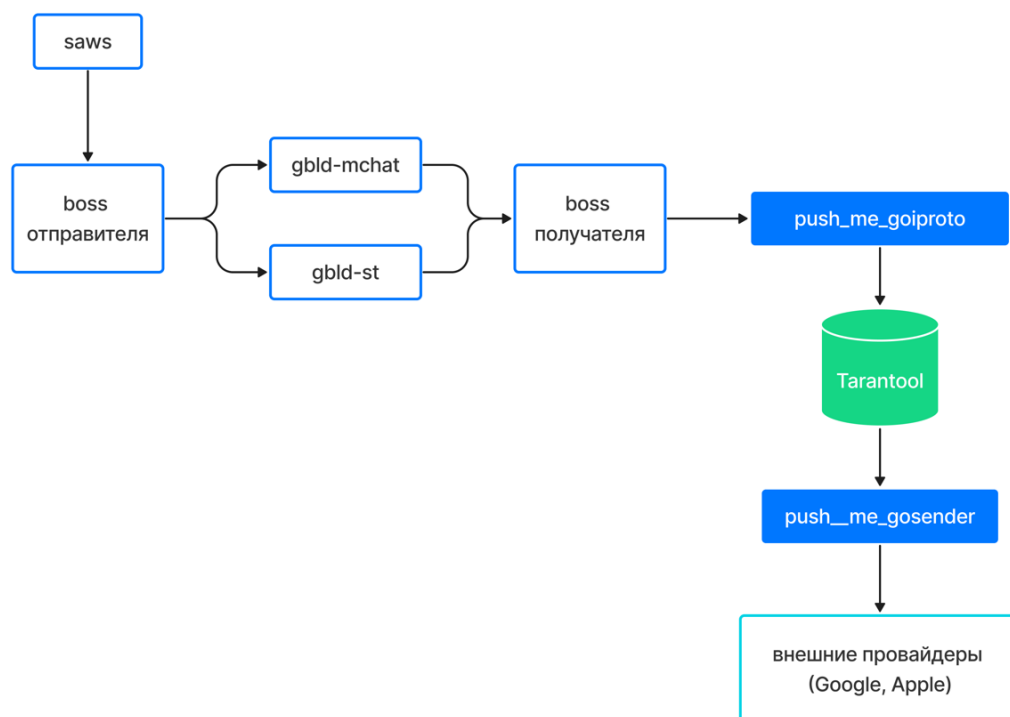
Отправляет запросы в следующие сервисы:

- Scribl — чтобы сервис Scribl отправил в клиентское приложение информацию, когда пользователь в последний раз был online;
- Feedog — чтобы проверить необходимость отправки подписчикам информации о времени, когда пользователь в последний раз был online;
- Prof-st — проверка статуса пользователя (удален/не удален), проверка когда пользователь в последний раз был online.

Получает запросы от следующих сервисов:

- Boss — установить время, когда пользователь в последний раз был online, установить статус у пользователя (заблокирован/не заблокирован);
- Scribl — получить время, когда пользователь в последний раз был online, и состояние пользователя;
- Prof-st — установить статус у пользователя (удален/не удален).

Пуш-уведомления



Push_me_goiprot

Сервис Boss получателя сообщения отправляет пуш-уведомление в сервис Push_me_goiprot по протоколу iprot. Далее пуш-уведомление попадает в очередь, реализованную в Tarantool.

Push_me_gosender

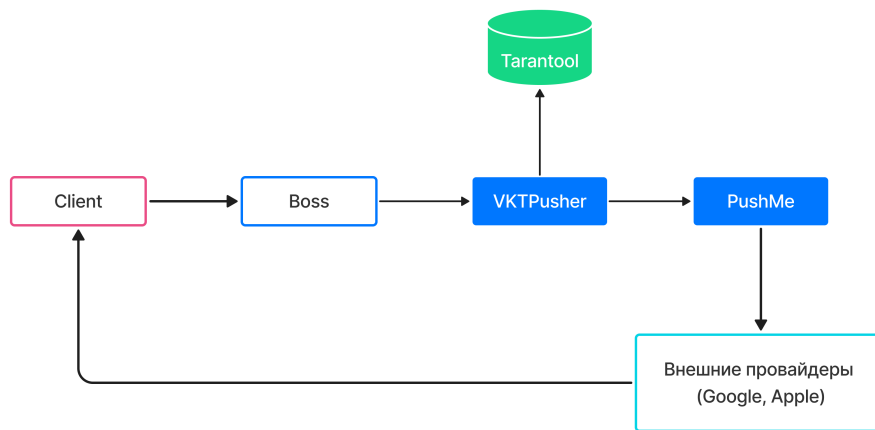
Сервис Push_me_gosender забирает из очереди пуш-уведомление, обрабатывает его (возможен вариант, когда сервис кладет в очередь полуобработанную версию пуш-уведомления) и отправляет провайдеру пуш-уведомлений.

VKTPusher

Сервис предназначен для формирования пуш-уведомлений для iOS и Android в формате, поддерживаемом клиентским приложением.

Сервис используется, если надо скрыть контент пуш-уведомлений на мобильных устройствах пользователей. В сервисе скрываются чувствительные данные и формируется контекст для правильной интерпретации на клиентских приложениях перед отправкой пуш-уведомления в APNs (Apple) или FCM (Google).

На схеме ниже представлена работа пуш-уведомлений при включенной настройке скрытия контента:



Клиент отправляет сообщение. Сервис Boss формирует контекст пуш-уведомления и отправляет его в сервис VKTPusher. VKTPusher формирует пуш-уведомление с текстом сообщения, именами отправителя/получателя и отправляет его в сервис Push_me_goiproto. Сервис Push_me_goiproto отправляет запрос в APNs (Apple) или FCM (Google). Пуш-уведомление приходит на клиент и отображается в UI.

Также сервис используется для отправки пуш-уведомлений через провайдера RuStore.

Взаимодействует с сервисами:

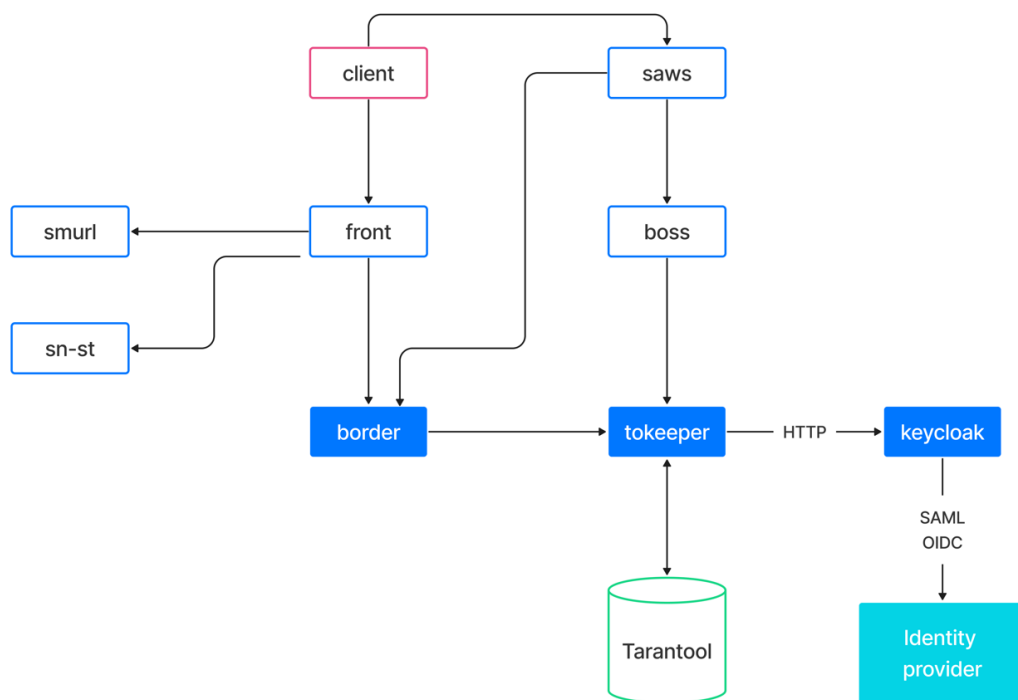
- Push_me_goiproto — отправляет контекст пуш-уведомления в сервис Push_me_goiproto по протоколу iproto по алгоритму Round Robin.
- Boss — получает от сервиса Boss контекст пуш-уведомления.

Аутентификация, авторизация и антибрутфорс

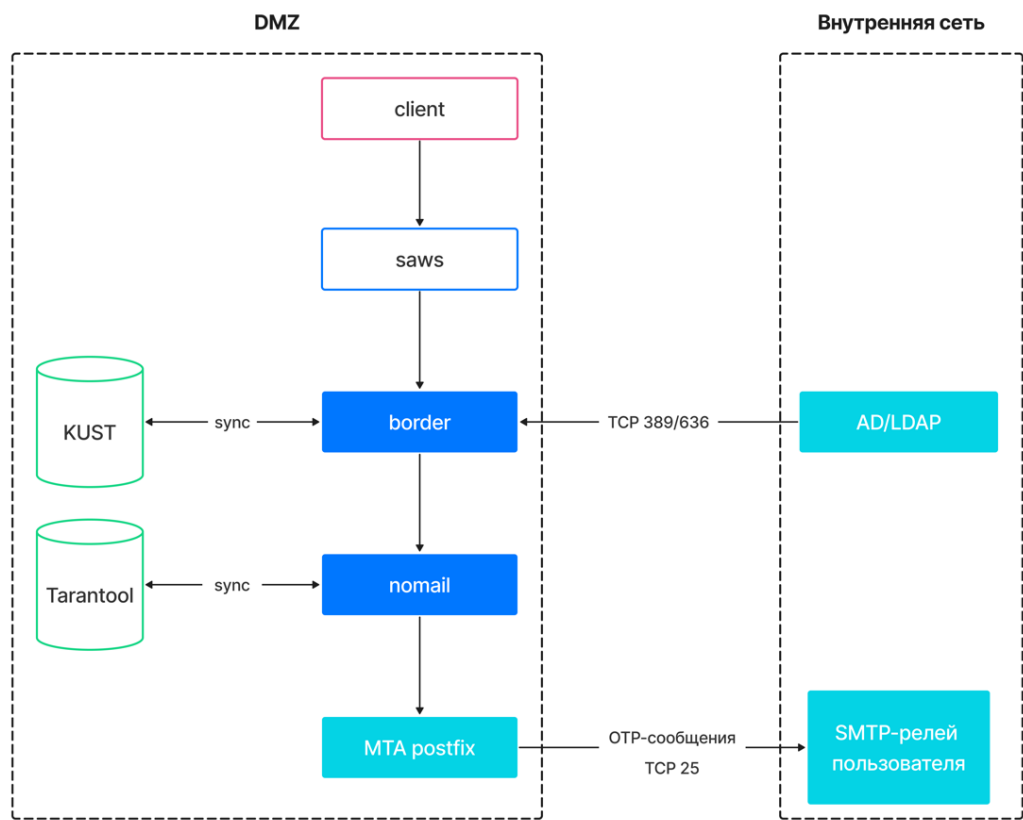
Варианты доступа к Системе

Представленные ниже варианты используются совместно.

Вариант 1: SSO-аутентификация (описание процесса настройки и диаграмма запросов представлены в документе [Инструкция по настройке Single Sign-on аутентификации](#)):



Вариант 2: Аутентификация с помощью одноразовых кодов (OTP), отправляемых на почту пользователя.
 При этом адрес почты для отправки совпадает с логином пользователя:



Ниже представлена диаграмма запросов для аутентификации при помощи OTP.



1. Пользователь вводит свой email в клиенте;
2. Клиент генерирует POST-запрос на сервис Saws, в котором передается тип токена (otp_via_email) и email пользователя;
3. Сервис Saws определяет по типу токена запроса, что должен быть отправлен OTP-код и отправляет запрос в сервис Border;
4. Сервис Border на основе полученного email генерирует OTP-код и отправляет данные (email и OTP-код) в сервис отправки email - Nomail;
5. Сервис Nomail сохраняет у себя полученный OTP-код и email и перенаправляет запрос в postfix-сервис для отправки OTP-кода пользователю;
6. Пользователь получает OTP-код и проходит процедуру аутентификации в Мессенджер и ВКС в форме входа в приложение вводит полученный код;

7. Клиент создает запрос с ранее введенным email и полученным OTP-кодом, устанавливает тип токена longTerm и передает запрос к сервису Saws;
8. Сервис Saws на основе полученных данных передает email и OTP-код в виде пароля в сервис Border;
9. Сервис Border проверяет на корректность полученный email и OTP-код в сервисе Nomail;
10. В случае успешной проверки введенного кода и email сервис Border генерирует токен авторизации и передает его в сервис Saws;
11. Сервис Saws передает сгенерированный токен авторизации в клиентское приложение;
12. После получения токена авторизации клиентское приложение отправляет запрос на создание сессии сервису Saws;
13. Сервис Saws отправляет запрос на проверку полученного токена в сервис Border;
14. В случае корректности сервис Border сообщает сервису Saws о валидности токена;
15. Затем сервис Saws отправляет запрос в сервис Boss на создание сессии для клиентского приложения;
16. Сервис Boss на основе полученного авторизационного токена и email генерирует токен aimsid и передает его сервису Saws;
17. Сервис Saws в свою очередь проксирует полученный токен aimsid в клиентское приложение.

Ниже представлено описание сервисов Мессенджер и ВКС, отвечающих за аутентификацию, авторизацию и антибрутфорс.

Border

Генерирует OTP-токены. Хранит и генерирует список авторизационных токенов для каждого пользователя, проверяет уже выданные авторизационные токены. Получает токены из cookie Почты. Данные хранит используя БД KUST.

Взаимодействует со следующими сервисами:

- Saws — сервис Border принимает от сервиса Saws IPROS-сообщения для авторизации пользователей, сервис Saws принимает от сервиса Border авторизационные токены, токены из почтовых cookie;
- Nomail — для возможности генерировать OTP-токены;
- Keycloak — посредством сервиса Keycloak осуществляется механизм проверки OTP-токенов;
- Sn-st – сервис Sn-st получает данные от сервиса Border для проверки/инвалидации OTP-токена;
- Boss – сервис Boss получает данные от сервиса Border для инвалидации OTP-токена.

Keycloak

Сервис-хранилище данных о пользователях. Все данные хранятся в БД MySQL.

Сервис осуществляет синхронизацию пользователей по протоколу LDAP. Хранит OTP-токены, полученные от сервиса Nomail.

Nomail

Основной сервис аутентификации внутри Системы. Получает данные о пользователях от сервиса Keycloak. Может работать как самостоятельно, так и совместно с сервисом Keycloak. Все данные сервиса Nomail хранятся в БД Tarantool.

MTA postfix принимает OTP-сообщения от сервиса Nomail и перенаправляет эти сообщения на SMTP-релей пользователя.

Примечание

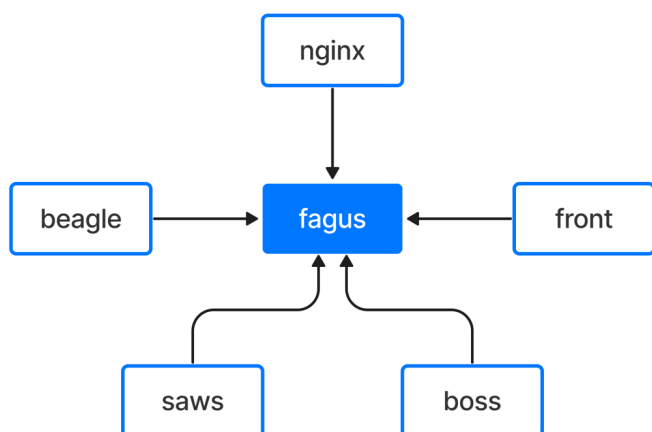
При наличии проблемы с синхронизацией пользователей ошибка, скорее всего, относится к сервису Keycloak. Иногда поломка сервиса Keycloak приводит к удалению его копии базы пользователей. В таком случае сервис Nomail не получает данные о пользователях и разлогинивает всех.

Tokenkeeper

Stateless сервис для хранения, обновления и проверки валидности JSON Web Token для настройки Single Sign-on аутентификации.

Хранит, обновляет и проверяет токены. Сервис хранит данные, используя БД Tarantool. БД Tarantool отслеживает токены, период жизни которых истек, и автоматически удаляет их из базы.

Fagus



Сервис анти-брутфорс. Хранит rate-лимиты пользователей.

Использует БД KUST, чтобы сохранять значения для rate-лимитов, настроенных на продолжительное время, между рестартами. Лимиты настраиваются в таблицах сервиса Stdb.

Умеет блокировать IP или клиента на указанный промежуток времени. Все настройки rate-лимитов задаются отдельно для каждого метода и отдельно для IP и rid.

По умолчанию сервис Fagus не проверяет rate-лимит для id, если в запросе передан признак, что инициатор запроса не авторизован. Чтобы включить проверку id для неавторизованных запросов есть настройка в конфигурационном файле.

Взаимодействует с сервисами:

- Beagle — для проверки rate-лимитов при поиске по номеру телефона.
- Nginx;
- Front;
- Saws;
- Boss.

Pocus

Прокси-сервис для шардирования и отказоустойчивости инстансов сервиса Focus.

Сопоставляет идентификатор каждого группового звонка с инстансом сервиса Focus, который его обслуживает. Если для идентификатора звонка не найден инстанс, заводит новый контекст для этого идентификатора и выбирает инстанс из доступных (по алгоритму Round-robin).

Focus

Сервер групповых звонков.

Отвечает за хранение списка комнат, участников комнат, обслуживает процесс их добавления/удаления. Экспортирует виртуальные контакты вида <guid>@focus, где guid - идентификатор звонка (конференции).

Обменивается с клиентами сигнальными сообщениями с помощью сервиса Boss.

Управляет инстансами сервиса Janus.

С помощью сервиса VCL проверяет валидность ссылки при входе клиента в звонок по ссылке.

Janus

Медиа-сервер групповых звонков.

На нем осуществляется маршрутизация, микширование, перекодирование медиа трафика. Управляется сервисом Focus, но также сообщает ему о статусе подсоединенных клиентов.

Клиент взаимодействует с сервисом Janus напрямую, согласно методу [ICE](#).

Таким образом, сервер для сервиса Janus должен иметь 2 интерфейса:

- Публичный (PublicIP), по которому клиенты взаимодействует с сервисом Janus. Он должен быть открыт для протоколов UDP и TCP в обе стороны.

- Приватный (PrivateIP), по которому осуществляется управление от сервиса Focus и скриптов запуска/остановки по REST API.

Сервис может иметь относительно большую загрузку CPU из-за режима перекодирования видео, шифрования и высокого пакет-рейта.

Vcl

Вспомогательный сервис для звонков по ссылке. Создает/хранит ссылки на звонок (uuid, создатель, время создания) и параметры конференции.

Взаимодействует с сервисом Mchat-st для создания чатов в звонках.

Voice-relayicq

Обеспечивает работу STUN – помогает клиентам видеть их внешние IP-адреса для сбора ICE-кандидатов.

Обеспечивает работу TURN – при недоступности прямого подключения клиентов друг к другу в р2р звонок, проксирует через себя медиа-трафик.

Используется для создания идентификаторов звонка.

Recorder

Сервис, управляющий записью конференций. Принимает запросы по управлению записью от сервиса Focus по протоколу IPROS.

Для работы сервиса используются утилиты:

1. dumpy — VoIP-клиент, который подключается к звонку и сохраняет ivf- и rtp-пакеты. При загрузке аватаров участников звонка используются запросы с prism-токеном.
2. rmp — компонент для конвертации rtp-дампов + мета-информации в итоговый mp4-файл.

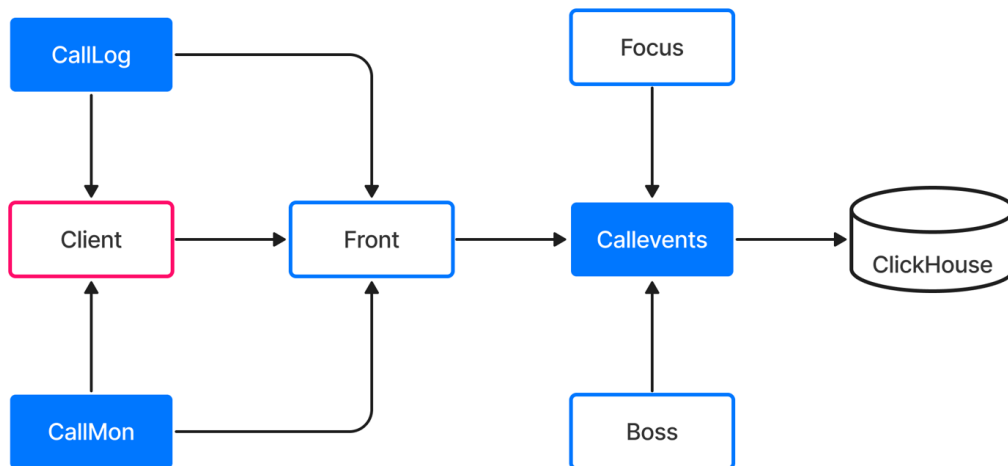
Сервис хранит информацию о текущей записи в RAM (id комнаты, модератор).

Во время активной записи звонка возрастает нагрузка на CPU и диск. Примерный размер дампов за один час - 850 Мб. Фаза рендеринга (дамп rtp-пакета в .mp4 файл) также потребляет CPU, но приоритет данного процесса можно снизить.

Готовое видео с записью загружается в сервис Go-files по протоколу HTTP. После успешной загрузки директория с дампами очищается, т.е. сервис потребляет дисковое пространство только в активной фазе записи.

Сервис Recorder использует Bot API, чтобы отправлять модератору файл с записью конференции.

Callevents



Сервис для мониторинга и статистики звонков.

Хранит список конференций, участников, сессий. Данные хранит используя БД ClickHouse.

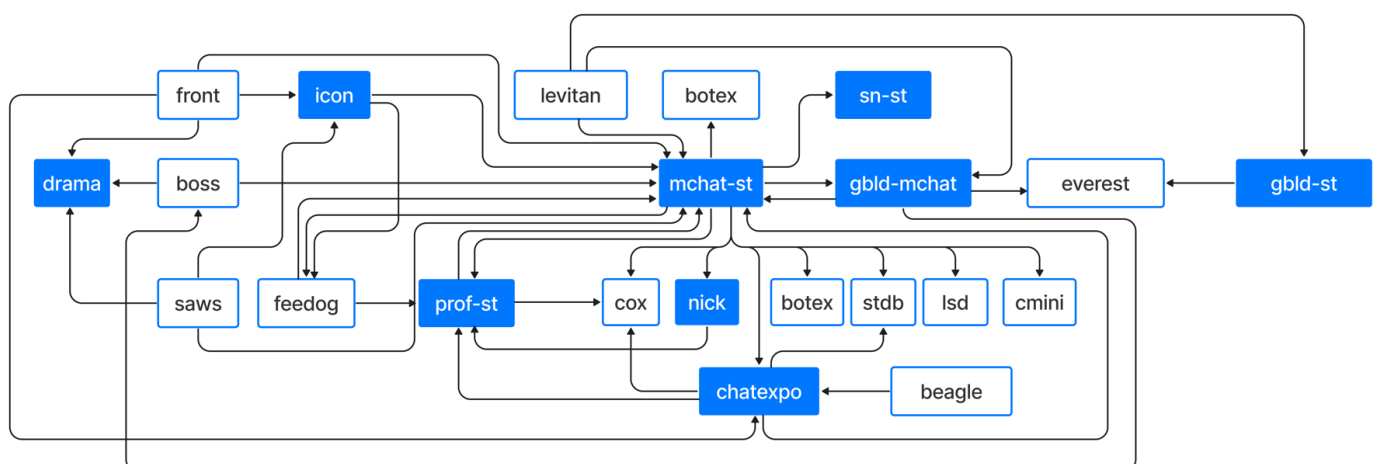
Принимает данные от следующих сервисов по протоколу IPROS:

- Focus — начало и конец конференции и сессии для групповых звонков и для звонков по ссылке;
- Boss — начало и конец конференции и сессии для р2р-звонков;
- Front — статистика звонков.

Отвечает на запросы от следующих сервисов (по протоколу IPROS):

- мини-апп для администраторов CallMon — запрос статистики звонков;
- мини-апп CallLog — список звонков.

Чаты, группы



Mchat-st

Сервис-хранилище групповых чатов.

Хранит анкеты чатов, списки участников и их прав, заблокированных, ожидающих входа. Данные хранит, используя БД KUST.

Обрабатывает пользовательские запросы на изменения анкеты, добавление участников и т.д.

Проксирует запросы на сохранение сообщений из сервиса Boss в сервис Gbld-mchat.

Принимает запросы от следующих сервисов по протоколу IPROS:

- Boss:
 - сохранение сообщений — сервис Mchat-st делает коррекцию некоторых полей, далее сохраняет в сервис Gbld-mchat;
 - создание чата;
 - удаление участников чата;
 - добавление участников чата;
- Botex - отправляет сообщение от ботов (в формате сервиса Boss);
- Feedog:
 - при удалении чата из контакт-листа просит удалить участника из сервиса Mchat-st;
 - перемещение участников чата во время перемещения аккаунта;
- Front:
 - все клиентские запросы с u.<DOMAIN>
 - доверенные запросы разработчиков с prism-токеном;
- Gbld-mchat:
 - проверка прав участников чата при получении истории;
 - синхронизацияписка участников;
- Levitan - проверка прав участников чата при получении файлов галереи;
- Prof-st - отдает анкету чата как профиль;
- Saws:
 - проверяет права участника чата для смены аватара;
 - уведомление об изменении аватара.

Отправляет запросы в следующие сервисы по протоколу IPROS:

- Botex — проверка ботов на возможность добавления в чат;
- Stdb — подписка на параметры;
- Feedog — обновление контакт-листов;

- Gbld-mchat:
 - проксирование сообщений от сервиса Boss;
 - генерация и сохранение служебных сообщений;
- Sn-st — логин в id;
- Nick — регистрация/изменение названия чата;
- Drama — удаление черновика сообщения пользователя, если он покидает чат;
- Prof-st:
 - id в логин;
 - получение профиля пользователя.

Gbld-st & Gbld-mchat

Сервисы для хранения переписки между пользователями и доставке новых сообщений (апдейтов) до получателей. Данные хранит, используя БД KUST.

- Gbld-st — обрабатывает и хранит сообщение в личных переписках пользователей,
- Gbld-mchat — обрабатывает и хранит сообщения в групповых чатах.

Drama

Сервис для хранения черновиков. Данные хранит, используя БД Tarantool.

Сервис хранит следующую информацию о черновиках:

- автор черновика;
- в каком диалоге установлен черновик;
- время установки черновика;
- тело черновика;
- установлен ли черновик для обсуждения.

Принимает запросы от следующих сервисов:

- Front — установка черновиков и получение истории;
- Saws — сброс черновиков при отправке сообщения;
- Boss — получает черновики при старте сессии;
- Mchat-st — удаление черновика сообщения пользователя, если он покидает чат.

Prof-st

Хранение анкеты пользователя (имя, фамилия и т.д.) и получение по id в логин пользователя. Данные хранит, используя БД KUST.

Отправляет данные в большинство сервисов кластера.

Импортирует анкеты из сервиса Feedog.

Nick

Сервис хранилище уникальных никнеймов для ботов и групповых чатов. Хранит привязку никнейма и идентификатора пользователя, а также историю изменений никнейма. Данные хранит, используя БД KUST.

Никнейм, хранимый в сервисе Nick, дублирует поле из профиля (сервис Prof-st) или поле из чата (сервис Mchat-st).

При изменении никнейма, сначала происходит изменение в сервисе Nick, затем в сервисах Prof-st/Mchat-st в зависимости от учетной записи.

Сервисы Prof-st и Mchat-st при изменении своего поля формируют update, который служит признаком регистрации никнейма. Если update не было, то никнейм считается не зарегистрированным, то есть свободным.

Sn-st

Сервис-хранилище скриннеймов пользователей. Данные хранит, используя БД Tarantool.

Chatexpo

Сервис рекомендаций и поиска чатов/ботов/каналов (витрина). Витрина конфигурируется менеджером через различные web API. Настройки чатов хранятся в БД Tarantool.

Витрина состоит из двух блоков:

1. Геочаты (опционально).
2. Региональные чаты.

Гео чаты конфигурируются при помощи кластеров, у которых есть координата и радиус. Если координата пользователя (получаем от клиента) попадает в окружность кластера, пользователю отображаются чаты внутри этого кластера с заданным порядком.

Далее после них отображаются региональные чаты. Если точка пользователя попадает в 2 или более кластера, отображаются все чаты из всех кластеров с сохранением порядка; кластеры сортируются по приближению.

Региональные чаты показываются всегда и зависят только от региона пользователя, который вычисляется по геолокации IP.

Принимает запросы от сервисов по протоколу IPROS:

- Beagle — поиск;
- Front — выдача витрины.

Отправляет запросы в сервисы по протоколу IPROS:

- Stdб — подписка на параметры;
- Cox — получение контакт-листа отправителя;
- Prof-st — получение анкеты (для возраста), логин в id.
- Mchat-st — анкеты чатов, списки участников.

Icon

Хранит аватары пользователей и групп.

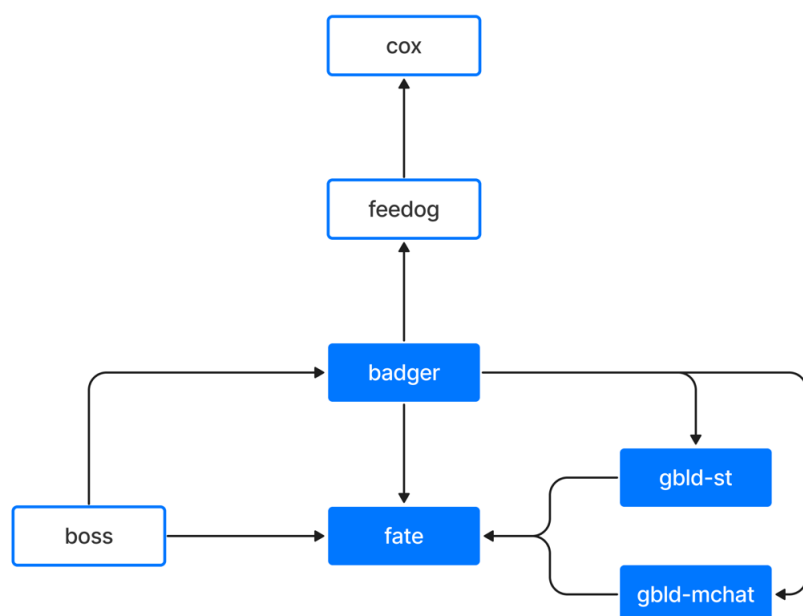
Принимает запросы от сервисов:

- Front — новые клиенты;
- Saws — старые клиенты/

Отправляет запросы в сервисы:

- Mchat-st — проверки прав при добавлении аватара группы;
- Feeddog — привязка аватара к пользователю.

Badger



Сервис хранит количество непрочитанных сообщений у пользователей по всем их контактам и отдает это число (с учетом непрочитанных сообщений в чатах с отключенными уведомлениями).

Принимает запросы от сервисов по протоколу IPROS:

- Boss — отправляет в сервис сумму непрочитанных сообщений в чатах пользователя.

Отправляет запросы в сервисы по протоколу IPROS:

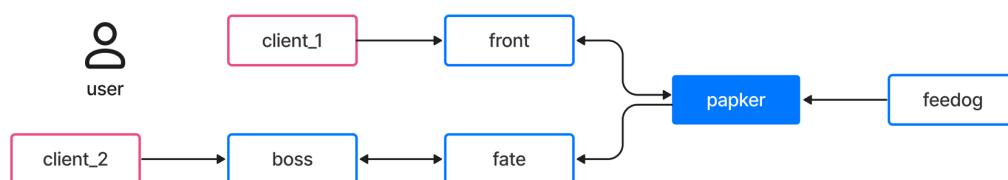
- Fate — получает от сервиса информацию для обновления числа непрочитанных сообщений для пользователя;
- Feedog — получает от сервиса контакт-лист пользователя и информацию об отключенных уведомлениях для контактов пользователя;
- Gbld-st/ Gbld-mchat — получает от сервисов сообщения для пользователя.

Fate

В сервисе реализована очередь для данных, поступающих от сервиса Gbld-st/ Gbld-mchat к сервисам:

- Boss;
- Badger;
- Papker.

Papker



В сервисе хранятся папки с чатами для каждого пользователя.

Сервис предоставляет возможность:

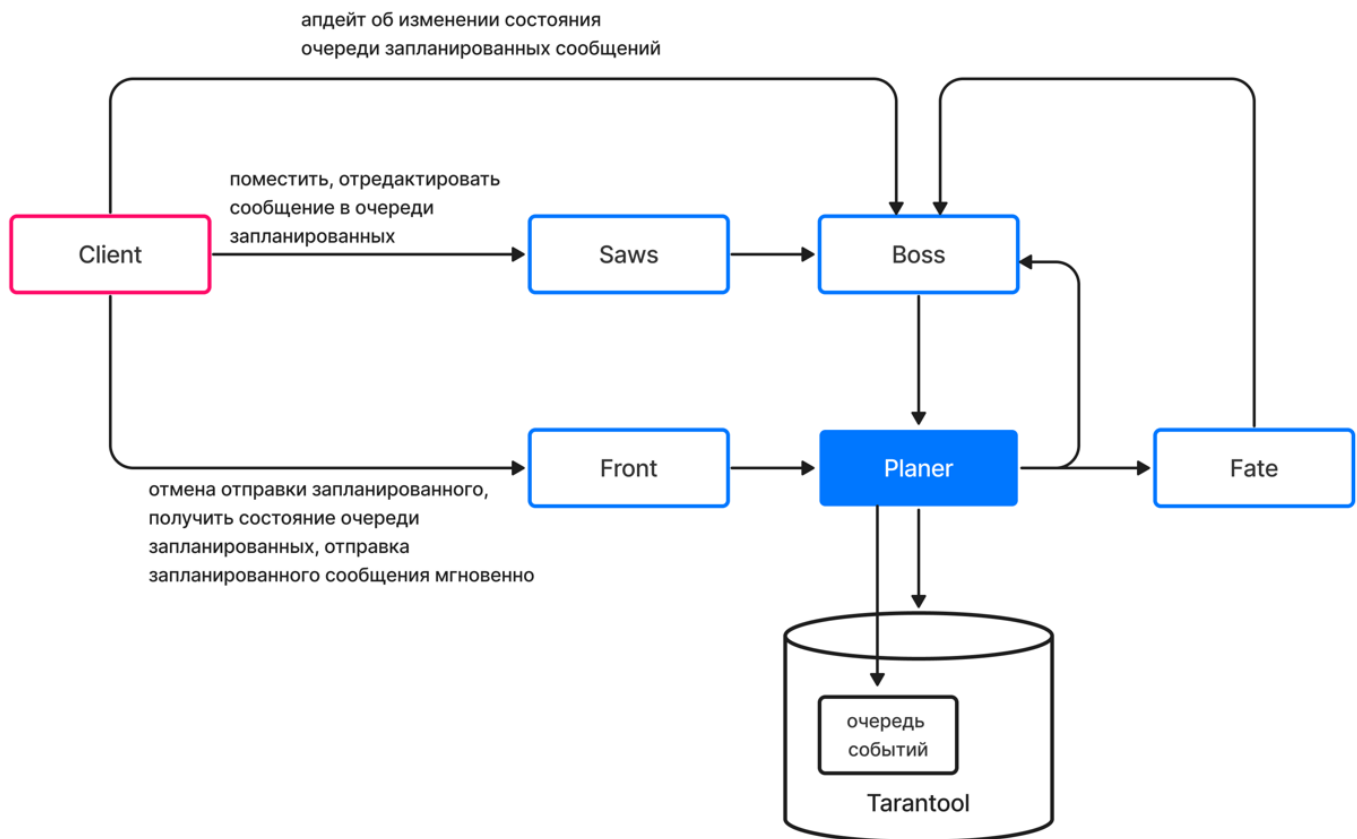
- создавать новую папку — как с чатами, так и без чатов (пустая папка);
- редактировать папку;
- удалять папку;
- менять порядок папок.

Существует лимит на количество папок, который может быть изменен администратором организации.

Взаимодействует с сервисами:

- Boss — передает сервису Boss перечень папок пользователей при старте сессии.
- Fate — через сервис Fate сервис Papker оповещает сервис Boss об обновлении состояния папок.
- Feedog — получает от сервиса Feedog обновления контакт-листа пользователя.
- Front — принимает от сервиса Front запросы о папках пользователей.

Planer



Сервис запланированных сообщений. Хранит очереди запланированных сообщений. Позволяет добавлять, удалять, редактировать запланированные сообщения, а так же инициирует их рассылки. Данные хранит, используя БД Tarantool.

В БД Tarantool есть очередь событий, которую наполняет сама БД. Существует два типа событий:

- на отправку запланированного сообщения;
- на отправку апдейта об изменении состояния очереди запланированных сообщений.

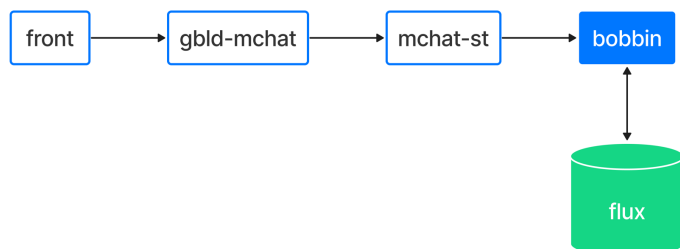
Клиентское приложение получает апдейты об изменении состояния очереди запланированных сообщений при помощи запроса в сервис Boss.

Planer взаимодействует с сервисами:

- Boss:
 - при создании нового запланированного сообщения;
 - при отправке запланированного сообщения получателю сообщения;
 - при рассылке начальных состояний непустых очередей запланированных сообщений после старта сессии;
 - при редактировании запланированного сообщения.
- Front:
 - при получении истории запланированных сообщений;
 - при отправке запланированного сообщения из очереди мгновенно;

- при отмене отправки запланированного сообщений и его удалении из очереди.
- Fate — отправка апдейта об изменении состояния очереди запланированных сообщений.

Треды (обсуждения)



Bobbin

Хранит метаинформацию о тредах, а именно — тип треда и родительское сообщение. Для каждого подписчика также хранится количество непрочитанных сообщений и непрочитанных упоминаний.

Процесс создания треда затрагивает следующие сервисы:

- Front;
- Gbld-Mchat;
- Mchat-St;
- Bobbin.

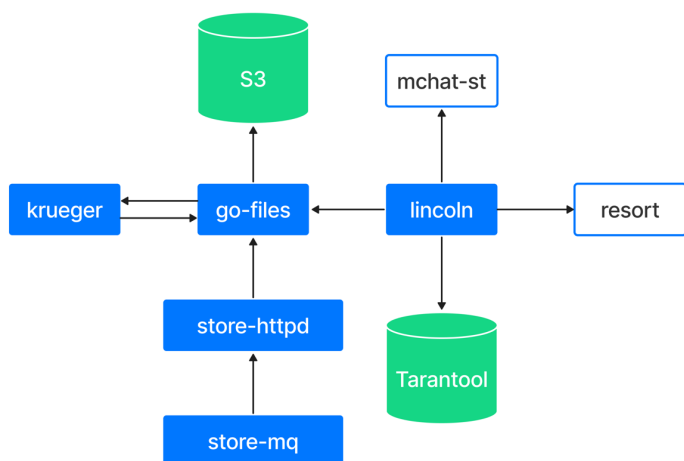
В сервисе Gbld-mchat производится транзакция, которая создает тред, либо возвращает id треда, если на сообщении уже есть тред.

Сервис Mchat-st создаёт анкеты треда. В сервис Bobbin доносится информация о новом треде.

Flux

Хранилище потоков тредов.

Сервисы для работы с прикрепленными файлами, аватарами и стикерами



Go-files

Сервис для работы с файлами, аватарами, стикерами. Хранит метаданные о файлах, которые хранятся в S3.

Позволяет загружать на сервер и скачивать файлы, готовит превью файлов.

Krueger

Сервис сжимает картинку для подготовки превью видео и картинки к в чате .

Производит нарезку картинок по требованию. Сервис Go-files отправляет файл, который необходимо нарезать, сервис Krueger обрабатывает запрос и возвращает ответ.

Lincoln

Сервис для подсчета ссылок на файлы. Данные хранит, используя БД Tarantool.

Порядок работы с задачами:

- получает задачу от сервиса Resort;
- проверяет консистентность данных задачи;
- сохраняет информацию о данных (добавление, удаление или редактирование данных) в БД Tarantool;
- выбирает статус выполнения задачи и отправляет его в сервис Resort.

Отправляет данные в следующие сервисы:

- Resort;
- Mchat-St;
- Go-files.

Store-httpd

Сервис для хранения и раздачи стикеров.

Store-mq

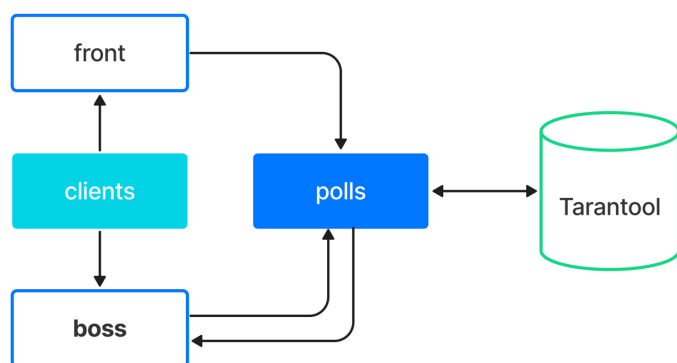
Сервис для хранения и раздачи стикеров (очередь).

Статусы

Gradus

Сервис для хранения статусов. Данные хранит, используя БД Tarantool.

Голосования



Polls-api

Предоставляет интерфейс для взаимодействия с опросами по протоколу IPROS.

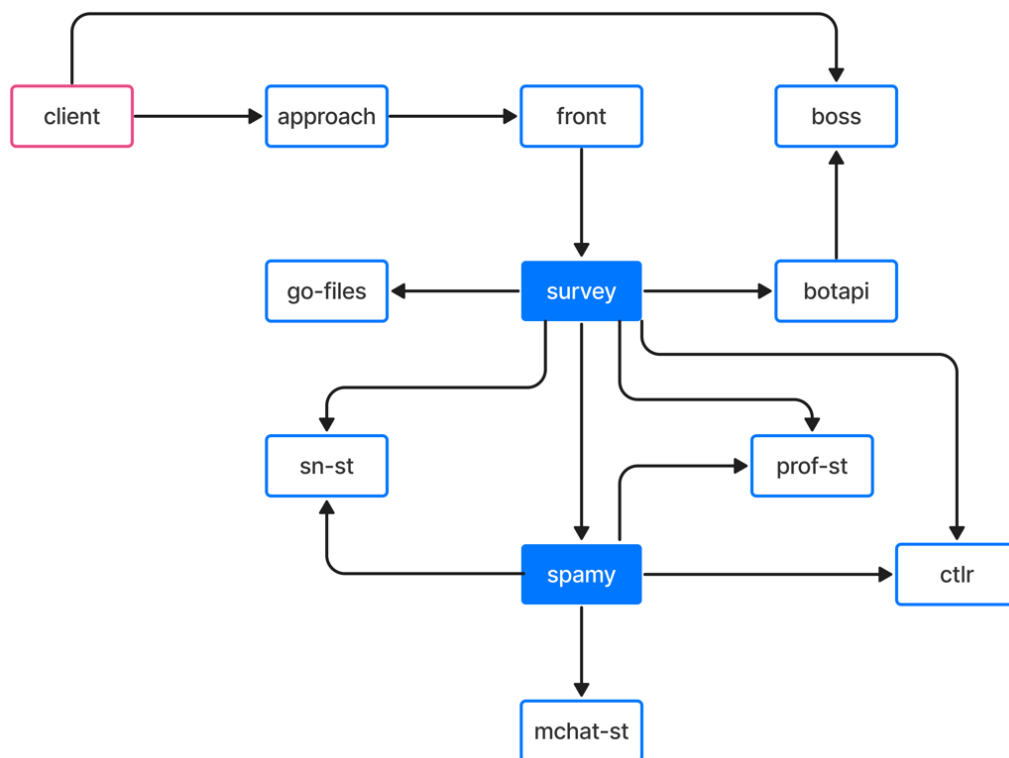
В качестве хранилища использует БД Tarantool. В сервисе также есть логика, которая реализует доставку клиентскому приложению изменений по опросам.

Polls-api при получении запроса расшифровывает IPROS, валидирует данные, затем выполняет запрос в БД Tarantool. БД Tarantool обрабатывает ответ, преобразует его в IPROS и отвечает.

Polls-updater

Сервис Polls-updater берет из очереди задачи на доставку событий с обновленными данными опроса, запрашивают данные из БД Tarantool и отправляет сообщения в сервис Boss.

Опросы



Survey

Сервис проведения опросов, отвечает за создание/хранение и обработку голосований по опросам. Данные хранит, используя БД Tarantool.

Сервис обрабатывает:

- IPROS-сообщения для администрирования опросов —создание, редактирование, обработку;
- данные по прохождению опросов.

Получает запросы от сервисов:

- Front — список групп и личных сообщений создателя опроса для добавления в опрос.

Отправляет запросы в сервисы:

- Spamy — получение списка пользователей из групп и личных сообщений создателя опроса для добавления в опрос;
- Bot API — отправка через Bot API в сервис Boss приглашения для прохождения опроса;
- Go-files — получение картинки опроса, если сообщение от бота отправляется с картинкой;
- Sn-st — получение внутреннего идентификатора пользователя по email;
- Prof-st — получение анкеты пользователя по внутреннему идентификатору;
- Ctrl — получение карты сервисов и регистрация в Ctrl.

Spamy

Сервис предоставляет создателю опроса перечень респондентов для добавления в опрос. Данные хранит, используя БД Tarantool.

Респонденты могут быть выбраны из двух источников:

- Групповые чаты, в которых есть создатель опроса.
- Личные сообщения — пользователи, которым может писать создатель опроса.

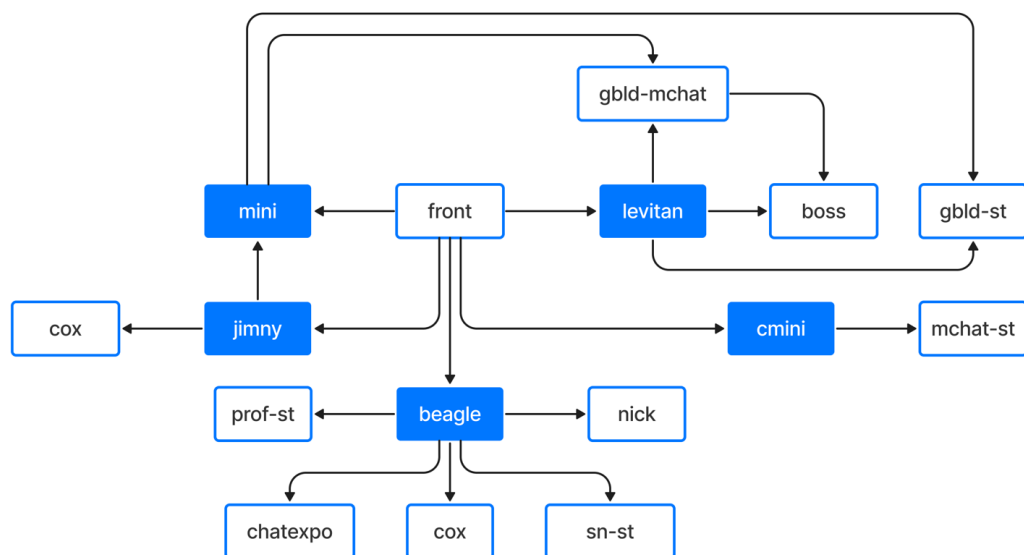
Получает запросы от сервисов:

- Survey — список пользователей из групп и личных сообщений создателя опроса для добавления в опрос.

Отправляет запросы в сервисы:

- Ctlr — получение карты сервисов и регистрация в Ctlr;
- Sn-st — получение внутреннего идентификатора пользователя по email;
- Prof-st — получение анкеты пользователя по внутреннему идентификатору;
- Mchat-st — получение списка пользователей группы.

Поиск



Beagle

Фронтенд для поиска, используется для поиска по полным совпадениям различных идентификаторов.

Входящие запросы по протоколу IPROS:

- Front — поиск;

Исходящие запросы по протоколу IPROS:

- Chatexpo — проксирует запрос поиска чатов;
- Сох — получение контакт-листа;
- Nick — поиск по полному совпадению ника;
- Prof-st — получение анкеты;
- Sn-st — логин в id.

Levitan

Индексер сообщений, содержащих элементы галереи (ссылки, фото, видео и т.д.).

Хранит в себе элементы галереи, события над сообщениями — личными и общими. Данные хранит, используя БД KUST.

Получает данные из сервисов Gbld-st и Gbld-mchat . Далее данные обрабатываются сервисом Levitan и сохраняются в БД KUST, после чего используются в диалоге.

Рассылает уведомления об изменениях галереи через сервис Gbld-mchat (для групповых чатов) и напрямую через сервис Boss (для личных сообщений).

Принимает запросы от сервиса Front для обработки пользовательского запроса на получения галереи.

Cmini

Сервис поиска анкет чатов по именам чатов. Данные хранит, используя БД KUST.

Анкеты сервисов передаются от сервиса Mchat-st.

Jimny

Statefull сервис поиска истории в чатах и группах.

Обслуживает запросы поиска по всем диалогам указанного пользователя. Хранит поисковую сессию в оперативной памяти. Сессия хранится для пагинации результатов поиска. Умеет восстанавливать сессию из токена, при этом тексты сообщений потребуются запросить повторно.

Обслуживает входящие запросы поиска от сервиса Front. Делает запросы в сервис Mini для каждого диалога в полученном из сервиса Сох списка. После получения списка вхождений из сервиса Mini, формирует выдачу постранично для всех диалогов. Запрашивает из сервиса Mini необходимые тексты сообщений для заполнения первой страницы.

Может сильно нагружать сервис Mini, так как для каждого поискового запроса выполняются поиск для большого числа пользователей. На момент написания документации число может превышать 20 000. При поиске медленных запросов стоит обратить внимание на сервис Mini.

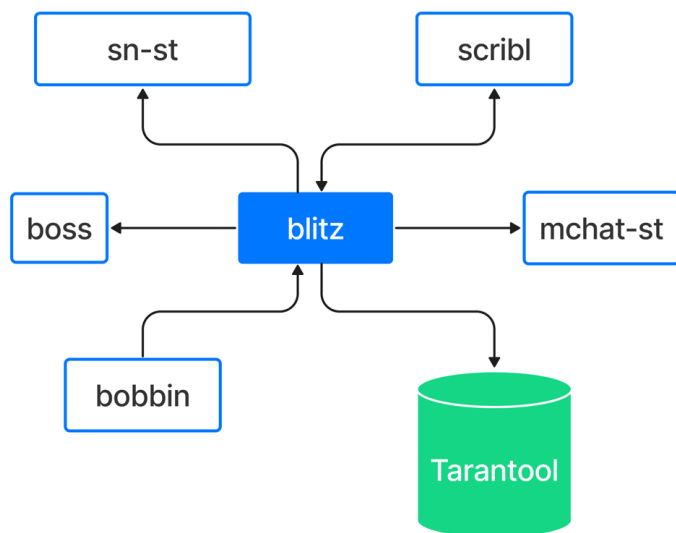
Mini

Индексер сообщений пользователей. Данные хранит, используя БД KUST.

Получает данные от сервисов Gbld-st и Gbld-mchat и индексирует токены из сообщений, ссылки, а также сниппеты. Ссылки разделяет на категории (фото, видео и т.д.) для последующего поиска по типам.

Отвечает на запросы поиска по истории от сервисов Front и Jimny.

Задачи



Blitz

Хранит в себе задачи, созданные пользователями, и подписки пользователей на задачи.

БД Tarantool хранит в себе все задачи и обратный индекс «пользователь → задача» для возможности запроса всех задач конкретного пользователя.

Имеется встроенный бот для управления уведомлениями о дедлайне задач. Задачи на отправку уведомлений о дедлайне получает из очереди хранящейся в БД Tarantool. Стандартные уведомления не рассылаются пользователям, которые осуществили настройку уведомлений, за исключением уведомления в момент дедлайна.

Поддерживается логика по отправке уведомлений в зависимости от статуса задачи и типа пользователя. Создателю задачи не приходят уведомления, если задача находится в статусах: «Готова», «Закрыта», «Отклонена». Исполнителю задачи не приходят уведомления, если задача находится в статусе «Закрыта». Для остальных подписчиков задачи уведомления не приходят, если задача находится в статусах: «Готова», «Закрыта».

Получает запросы от сервисов:

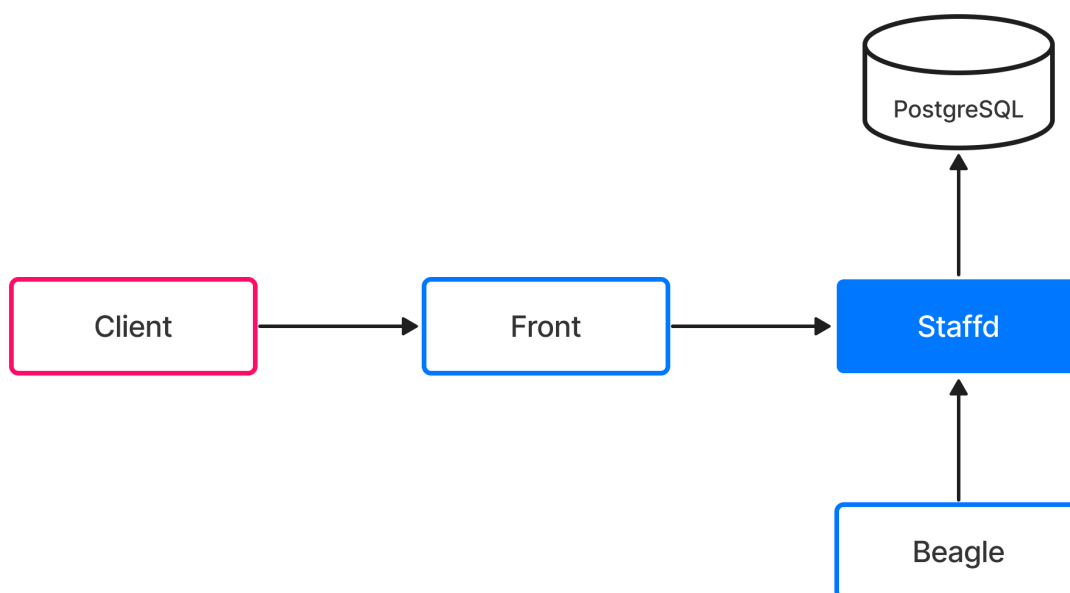
- Bobbin — подписка/отписка;
- Scribl — подписка об изменении задач.

Отправляет запросы в сервисы:

- Mchat-st — создание треда и уведомление в тред;
- Scribl — уведомление об изменении задач;
- Sn-st — резолвинг sn;
- Boss — проверка идентификатора сессии.

Структура организаций

Staffd



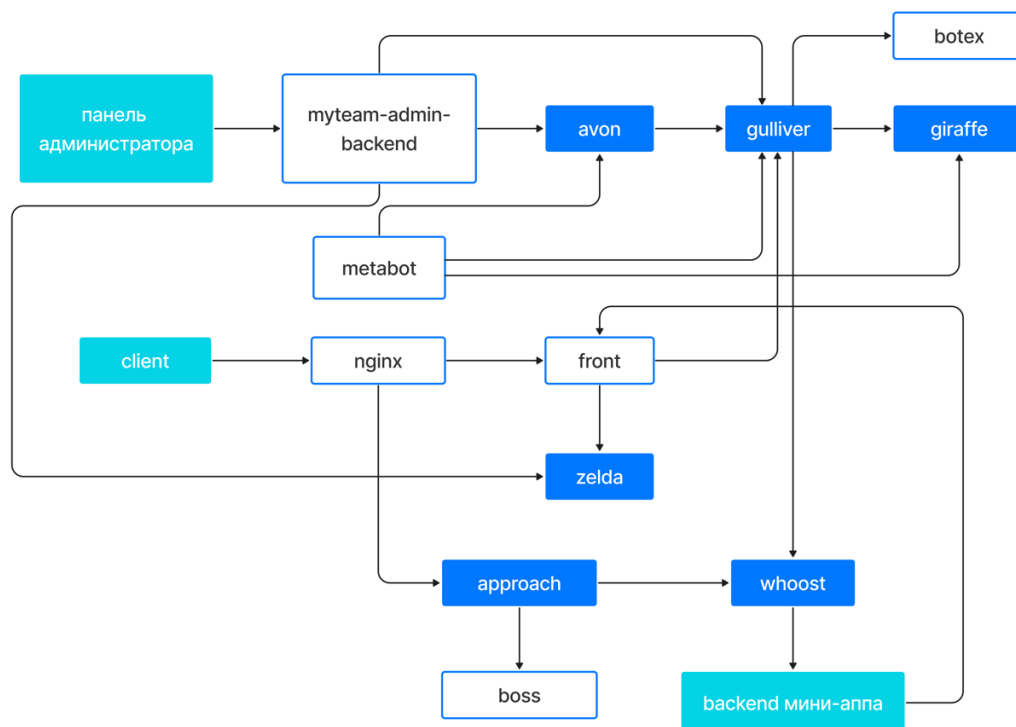
Сервис для хранения положения сотрудников в иерархии подразделений компании. Данные хранит, используя БД PostgreSQL.

Взаимодействует с сервисами:

- Front:
 - при создании, редактировании, удалении структуры организации;
 - при создании, редактировании, удалении подразделений;
 - при добавлении сотрудников в оргструктуру, их редактировании, перемещении и удалении;
 - при поиске, просмотре сотрудников в оргструктуре и просмотре иерархии оргструктуры;
 - при выгрузке и загрузке данных об оргструктуре.

- Beagle:
- при поиске сотрудников в оргструктуре.

Мини-аппы



Giraffe

Сервис для хранения информации о списке мини-аппов пользователя.

Данный сервис дополняет функционал сервисов Avon, Gulliver. Данные хранит, используя БД Tarantool.

Получает запросы от сервисов:

- Gulliver — для добавления, удаления мини-аппа из списка пользователя;
- Метабот — для получения списка мини-аппов пользователя.

Avon

Сервис для хранения списка мини-аппов инсталляции.

Поддерживает запросы на чтение, добавление, изменение и удаление списка мини-аппов. Данные хранит, используя БД Tarantool.

Получает запросы от сервисов:

- Myteam-admin-backend — при создании, удалении, изменении и получении информации о мини-аппах.

- Metabot — при создании и удалении мини-аппов.

Отправляет запросы в сервисы:

- Gulliver — во время удаления или получения данных о мини-аппе.

Whoost

Сервис для работы с мини-аппами.

Сервис хранит следующую информацию о вебхуке:

- Screenname владельца вебхука;
- URL вебхука;
- самоподписанный-сертификат в формате .PEM.

Данные хранит, используя БД Tarantool.

Аутентификация бэкенда при запросах к вебхуку выполняется при помощи SSL-сертификата бэкенда, который сгенерировал пользователь.

Подключение Мессенджер и ВКС к бэкэнду вебхука осуществляется с использованием mTLS.

Сервис Whoost выполняет запросы с конечными короткоживущими сертификатами, выпускаемыми для него по запросу в сервисе Vault и хранящиеся исключительно в оперативной памяти до момента завершения работы сервиса либо до момента перевыпуска сертификата. В сервисе Vault может храниться как корневой клиентский сертификат Мессенджер и ВКС (с приватным ключом для подписания конечных сертификатов), так и промежуточный сертификат (также с приватным ключом).

Отправляет запросы в сервисы:

- Vault (по протоколу HTTP(S);
- Backend вебхуков (по протоколу HTTPS).

Получает запросы от сервиса:

- Approach.

Approach

Stateless сервис. Проксирует запросы фронтенда мини-аппов, запускаемых в клиентском приложении VK WorkSpace, в бэкенд мини-аппа с проверкой авторизации.

Запросы балансируются с сервиса Nginx при поступлении запроса на Miniapp-host.

Получает запросы от сервисов:

- Nginx.

Отправляет запросы в сервисы:

- Boss;
- Gulliver;
- Whoost.

Gulliver

Сервис для хранения метаданных о мини-аппах. Хранит информацию в БД Tarantool.

Поддерживает запросы на чтение, добавление, изменение и удаление мини-аппов.

Получает запросы от сервисов:

- Approach — для получения информации о мини-аппе;
- Front — для получения информации о мини-аппе;
- Avon — во время удаления и получения информации о мини-аппе;
- Metabot — при создании, изменении и удалении мини-аппов;
- Myteam-admin-backend — при создании и изменении информации о мини-аппе.

Отправляет запросы в сервисы:

- Whoost:
 - для удаления вебхуков, во время удаления мини-аппа из хранилища;
 - для добавления вебхуков, при создании мини-аппа;
 - для получения информации об URL бэкенда и сертификате, во время получения данных о мини-аппе;
- Botex — при добавлении или изменении доносит до сервиса Botex ID мини-аппа;
- Giraffe:
 - для добавления мини-аппа в список мини-аппов пользователя при создании мини-аппа;
 - для удаления мини-аппа из списка мини-аппов пользователя при удалении мини-аппа.

Zelda

Сервис для хранения и конфигурации наборов ссылок, распределенных по доменам. Данные хранит, используя БД Tarantool.

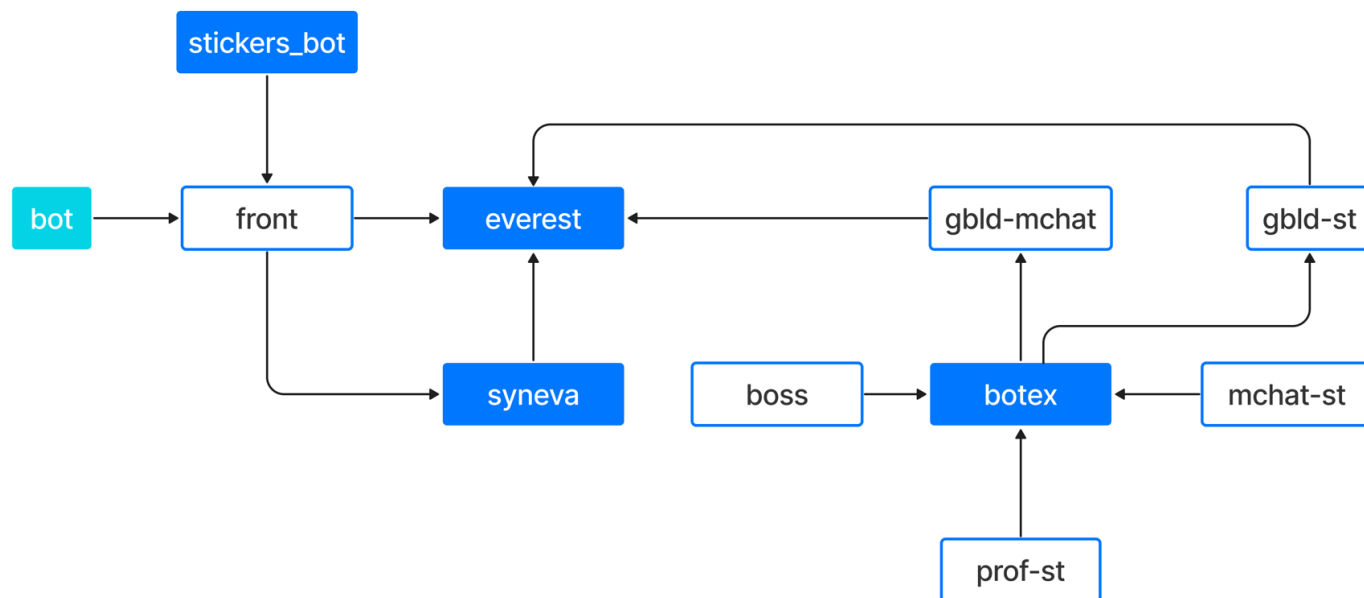
Получает запросы от сервисов:

- Front — для получения списка ссылок мини-аппа;
- Myteam-admin-backend — для добавления/удаления/получения списка ссылок мини-аппа.

Godmod

Сервис хранит информацию о мини-аппах и содержит методы API для настройки работы мини-аппов и конфигурирования системы.

Боты



Stickers_bot

Бот для управления стикерами и стикерпаками.

Botex

Сервис Botex:

- взаимодействует с ботами по вебхукам;
- хранит списки ботов для каждого владельца;
- получает информацию о владельцах ботов от сервиса Prof-st.

У сервиса Botex нет специализированной статистики, так как они представляют собой диспетчер запросов и простое хранилище информации о ботах. Всю необходимую информацию можно получить по статистике IPROS-пакетов и статистике БД KUST.

Syneva

Хранит контексты асинхронных запросов к ботам. Данные хранит, используя БД Tarantool.

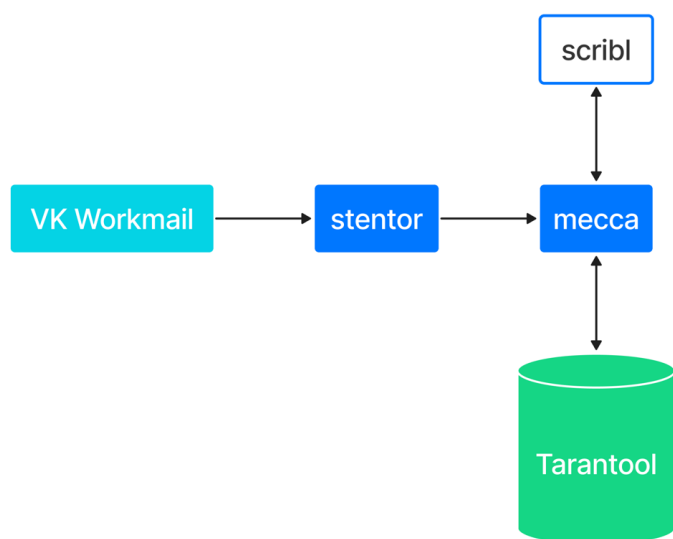
Everest

Поддерживает запросы на сохранение событий, чтение и удаление. Используется для инициированной сервером доставки информации клиентам. Данные хранит, используя БД Tarantool.

Получает запросы от следующих сервисов:

- Для сохранения событий:
 - Gbld-st;
 - Gbld-mchat;
 - Scribl.
- Для получения событий:
 - Front.

Интеграция с VK Workmail



Stentor

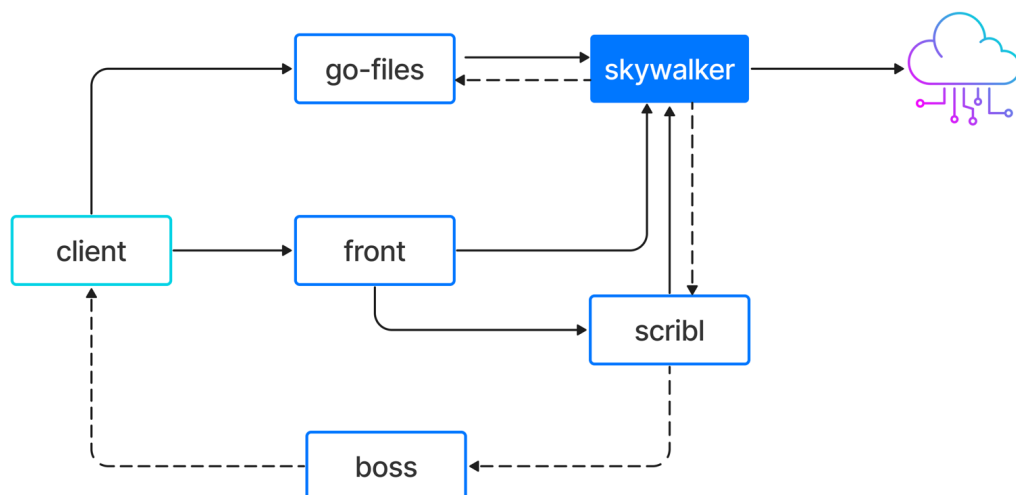
Сервис для интеграции с VK Workmail.

Mecca

Сервис для хранения количества непрочитанных писем для пользователя.

Данные хранит, используя БД Tarantool.

Skywalker



Сервис для интеграции с облаком. Позволяет сохранять файлы из мессенджера в облако и открывать их (генерирует ссылку на файл в облаке). Данные хранит, используя БД Tarantool.

Файлы, загруженные ранее в Систему, могут быть сохранены в облако пользователя через интерфейс клиентского приложения VK Workspace.

Файл сохраняется в облако с тем же именем, которое было указано при загрузке файла в мессенджер. Файл из мессенджера связывается с соответствующим файлом в облаке, и после сохранения может быть «открыт» по кнопке в виджете файла в интерфейсе клиента. Под «открытием» подразумевается переход в браузер и отображение файла в облаке. Это приводит к тому, что каждый файл может быть сохранен пользователем в облако только один раз.

Пользователи могут иметь доступ к файлам из других федераций.

Взаимодействует с сервисами:

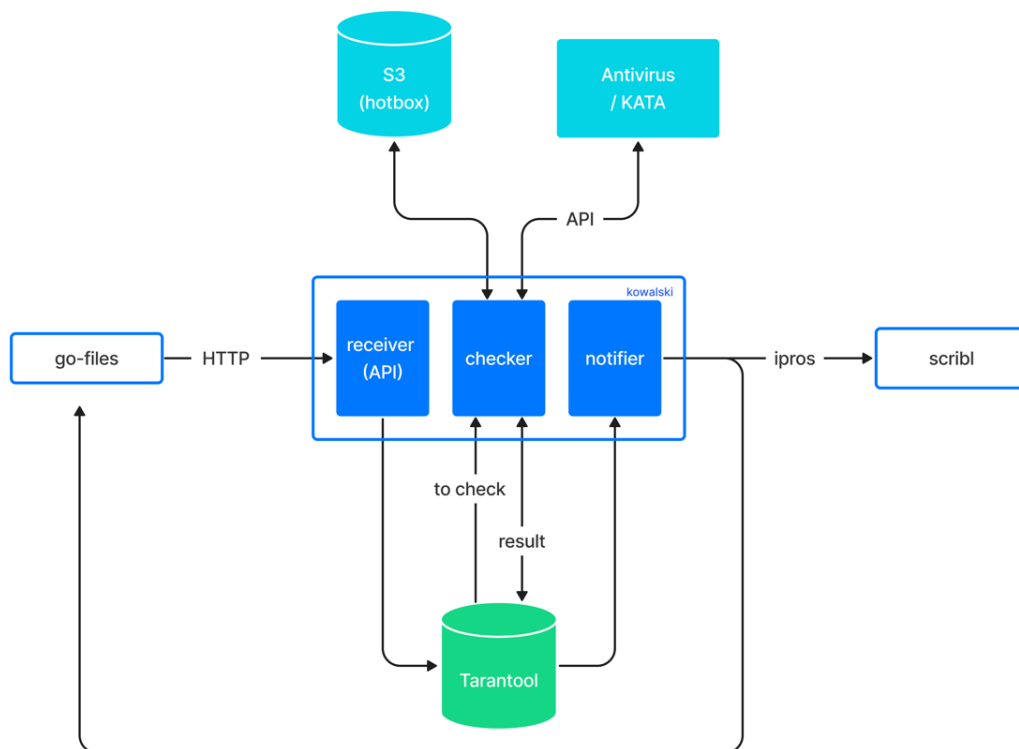
- Front — через IPROS API;
- Go-files — через внутренний HTTP API;
- Scribl — через IPROS API;
- С облаком через Server-Side API.

Интеграция с антивирусом

Kowalski

Сервис для интеграции с антивирусом.

Сервис проверяет файлы антивирусом, работает на собственных локальных очередях и общается с файловым бэкендом `ub.<DOMAIN>` и универсальным сервисом подписок Scribl.

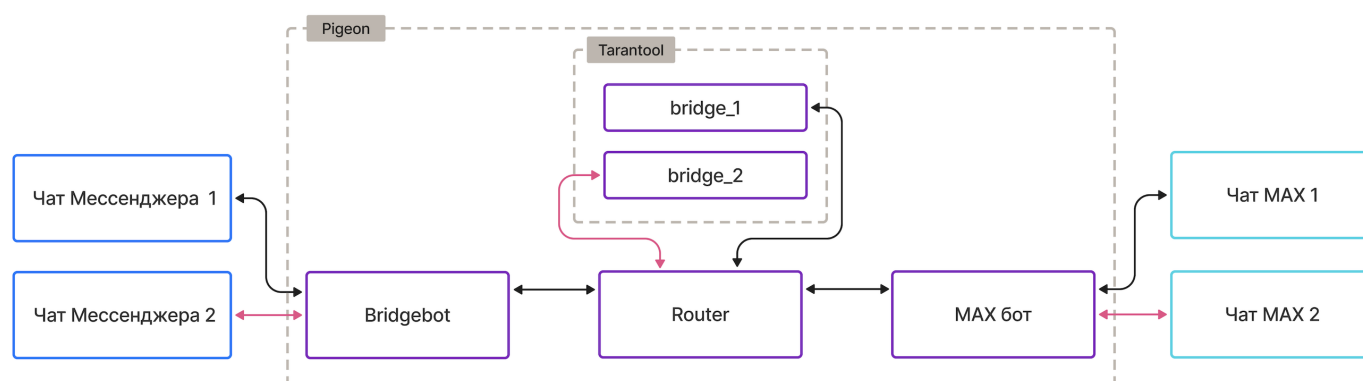


После загрузки файла в сервис Go-files идёт HTTP-запрос в сервис Kowalski с информацией о файле. Сервис помещает этот файл к себе в очередь на проверку.

Подробное описание антивирусной проверки представлено в документе «Руководство по администрированию».

Интеграция с MAX

Pigeon



Взаимодействие между чатами выполняется с помощью сервиса Pigeon, который перенаправляет сообщения из бота Мессенджера в бот MAX и в обратном направлении.

Сервис Pigeon состоит из:

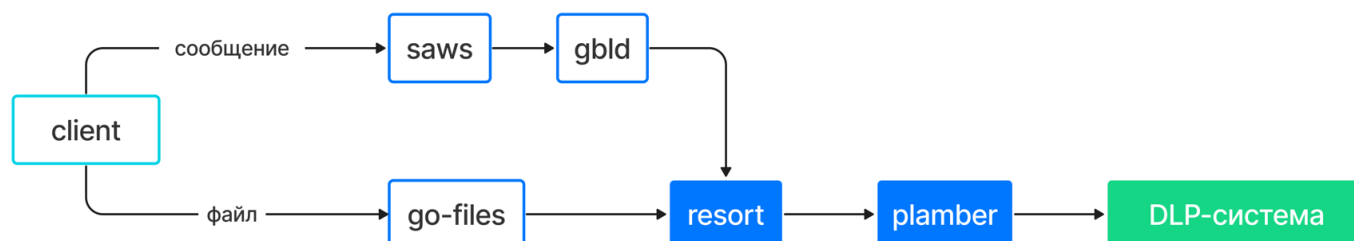
- Tarantool — хранилище всех соединений между чатами.
- Router — перенаправляет запросы в Tarantool для поиска установленной связи по id.

- Чат-боты — для установки связи между чатами.

Описание взаимодействия представлено в документе [Руководство по администрированию](#).

Интеграция с DLP-системами SearchInform и InfoWatch (до версии 25.2)

Plumber



Сервис для интеграции с DLP-системой заказчика. Сервис Plumber обрабатывает очередь из сообщений и файлов для отправки их в DLP-систему.

Сервис Plumber отправляет два типа запросов в DLP-систему:

- message — обычное текстовое сообщение;
- file — файл, загруженный пользователем.

Данные отправляются в очередь сервиса Resort при сохранении сообщения в сервисе Gbld-mchat (для групповых чатов) или Gbld-st (для 1to1), при сохранении файла в сервисе Go-files. Далее сервис Plumber получает данные от сервиса Resort и отправляет во внешнюю DLP-систему.

Выше представлена схема взаимодействия сервисов Мессенджер и ВКС при отправке сообщений/файлов во внешнюю DLP-систему заказчика.

Подробное описание отправки данных в DLP-систему представлено в документе «Руководство по администрированию».

Resort

Сервис Resort — это универсальная очередь для задач.

Поддерживает методы:

- добавить задачу;
- взять задачу;
- поставить статус выполнения задачи:
 - подтвердить успешное выполнение;

- заморозить задачу при невозможности исполнения;
- вернуть задачу в очередь при временной ошибке.

Принимает запросы от сервисов:

- Gbld-st + Gbld-mchat — для добавления задачи;
- Drama — для добавления задачи;
- Lincoln — для выполнения задачи.

Интеграция с DLP-системами (начиная с версии 25.2)

Схема отправки текстовых сообщений в DLP-систему:

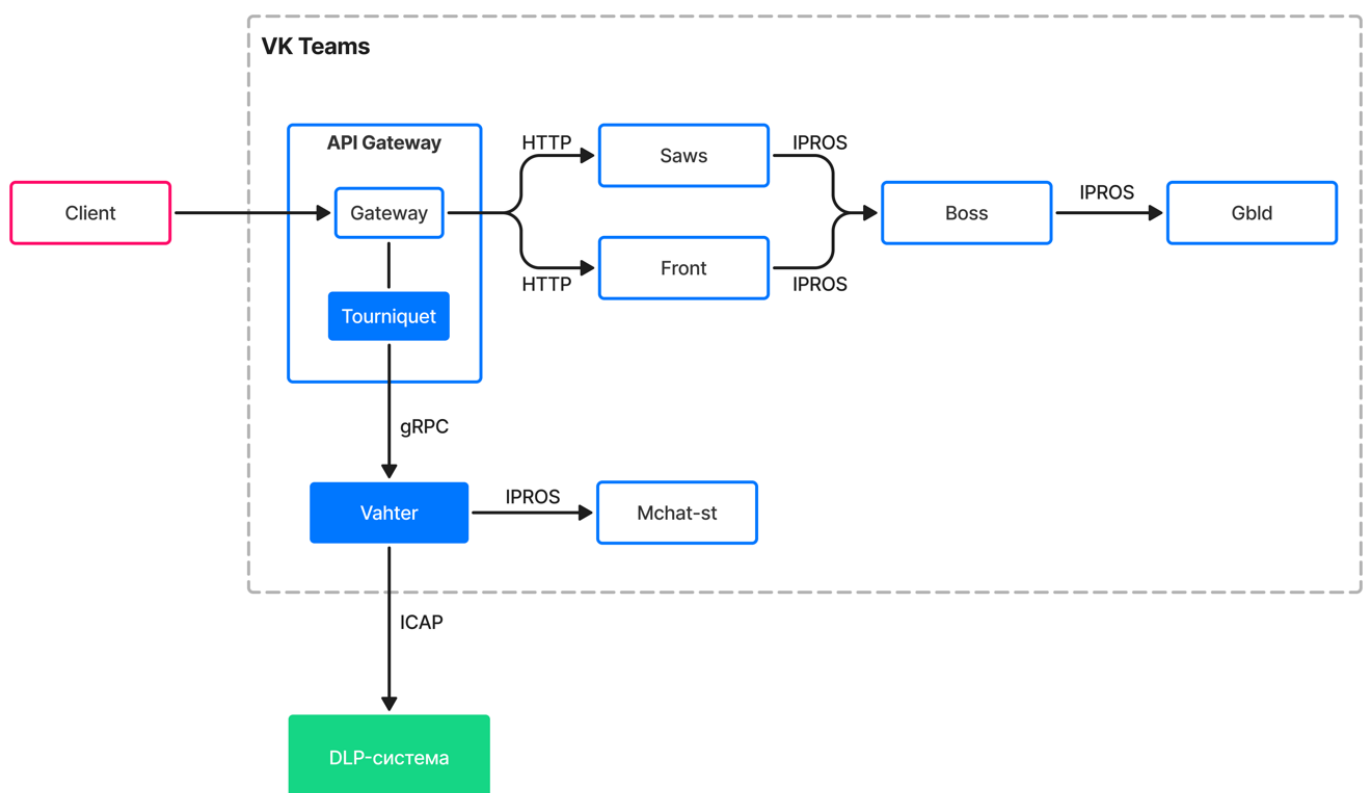
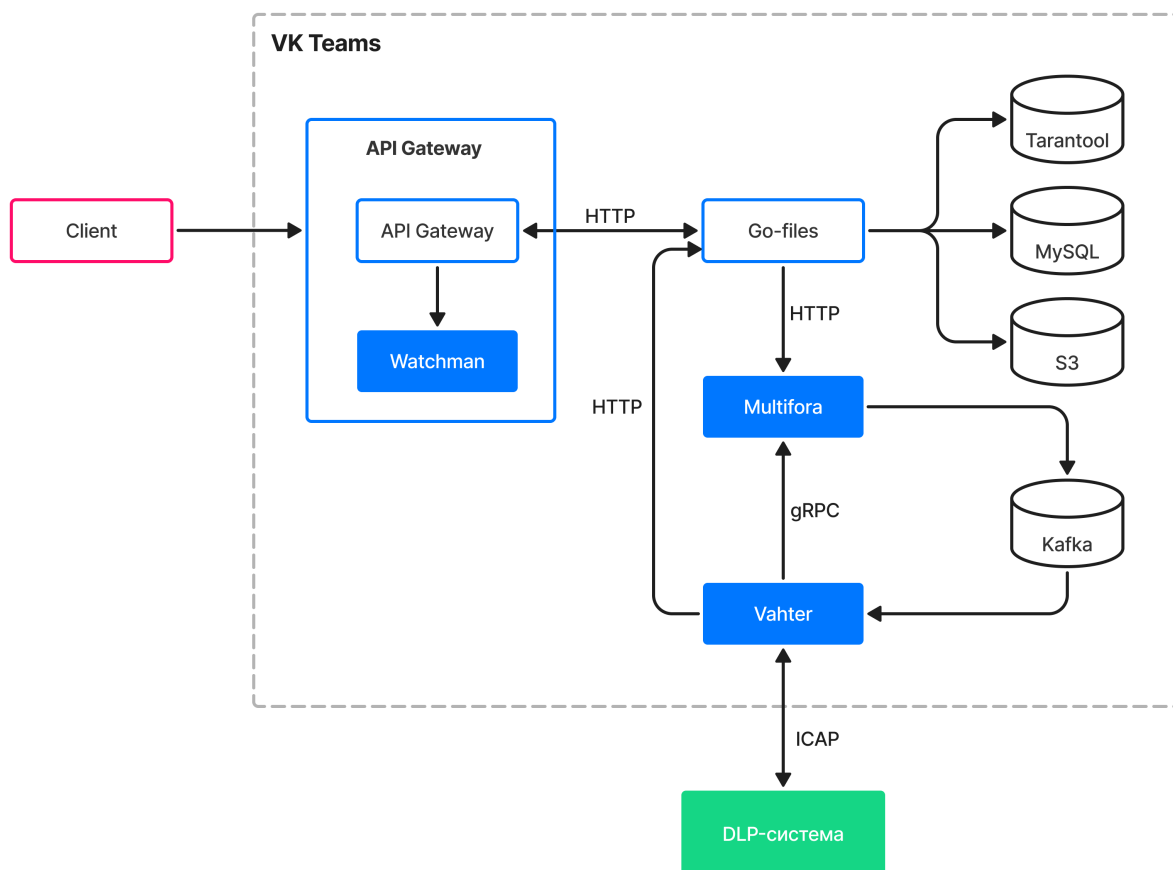


Схема отправки файлов в DLP-систему:



Vahter

Сервис предназначен для интеграции различных систем безопасности, в том числе и DLP через протокол ICAP. Сервис выполняет роль прокси, который получает отправляемые в мессенджере сообщения и их содержимое от API Gateway через встроенную Tourniquet middleware, дополняет запрос информацией из других сервисов Мессенджер и ВКС и отправляет получившийся запрос на проверку в DLP-систему, далее проксирует вердикт обратно в Gateway и либо пропускает сообщение дальше, либо блокирует его отправку.

Multifora

В момент завершения загрузки файла сервис Go-files сигнализируют сервису Multifora о том, что файл загружен и готов к проверке.

Сервис Multifora составляет запрос и кладет событие в Kafka для дальнейшей отправки в сервис Vahter для проверки файла в DLP-системе. Использование Kafka для передачи сообщений между компонентами системы позволяет реализовать асинхронную обработку данных, что влияет на производительность и устойчивость к скачкам нагрузки. Обеспечивает отказоустойчивость благодаря репликации и распределенной архитектуре, уменьшает вероятность потери данных.

Сервис Vahter формирует ICAP-запрос в DLP-систему, «выкачивает» содержимое файла из сервиса Go-files и передаёт его в теле ICAP-запроса в DLP-систему.

В результате взаимодействия с DLP-системой сервис Vahter устанавливает в сервисе Go-files результат проверки. Результат состоит из ответа (OK, Blocked) и уровня доступа к файлу (Internal, External).

Tourniquet

Модуль Tourniquet отвечает за проверку текстовых сообщений. В контексте проверки файлов он отвечает за подписи к файлам, комментарии при цитировании и пересылке сообщений, имеющие файлы и прочее.

Tourniquet встраивается в HTTP-сервер, обрабатывающий входящие запросы от клиентских приложений. Если от сервиса Vahter был получен положительный результат проверки, запрос проходит в сервис Saws и дальше так же, как если бы DLP-система отсутствовала.

Если был получен отрицательный результат проверки, запрос не будет пропущен дальше по процессу обработки запроса. Это гарантирует, что заблокированное сообщение не будет сохранено на сервере и не сможет быть доставлено до других клиентов системы.

Watchman

Модуль Watchman отвечает за сопоставление запросов пользователей на получение доступа к файлам с уровнем доступа, выставляемом согласно системе правил. Анализирует заголовки пользовательского запроса, используя информацию из них (User Agent и IP) для поиска соответствия в конфигурируемом списке правил. В результате анализа сервис Watchman выставляет дополнительный заголовок User-Level — в нём проставляется уровень конфиденциальности запроса - External или Internal.

Мониторинг

VictoriaMetrics

VictoriaMetrics — это специальное средство для мониторинга систем, включающее:

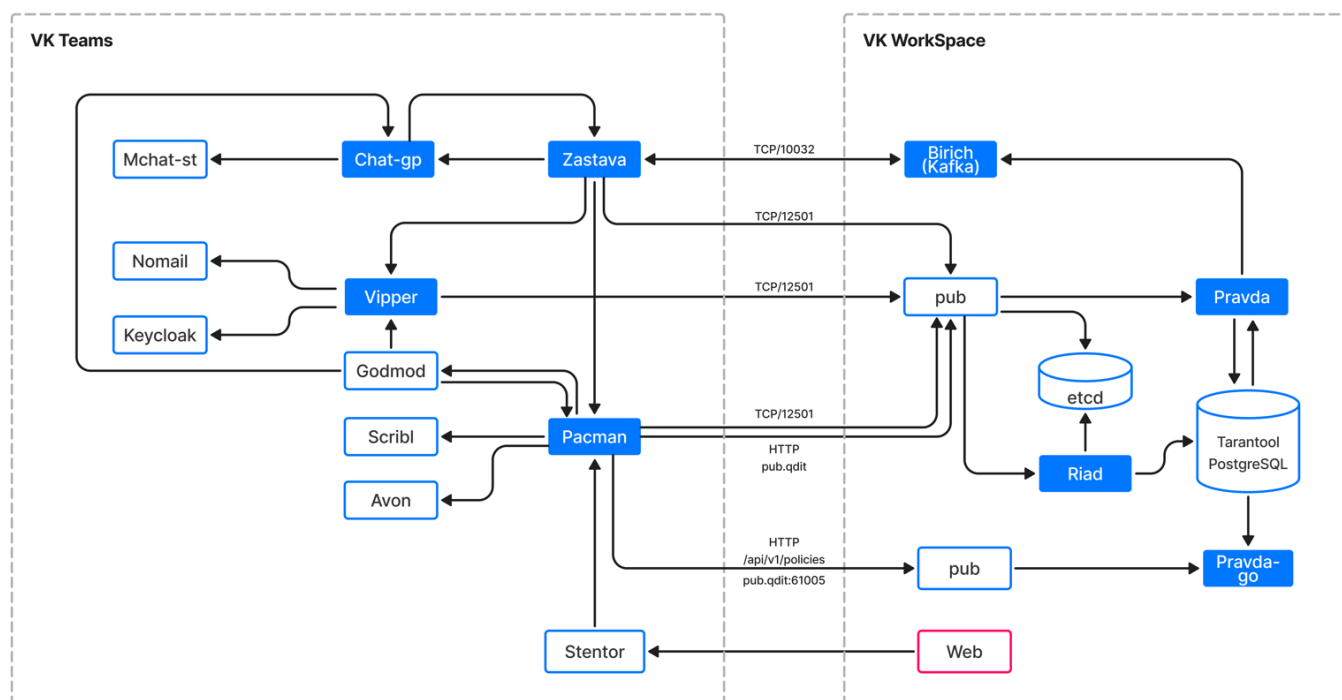
- базу данных метрик (временных рядов);
- извлечение метрик из целевых объектов;
- визуализацию метрик;
- оповещение при выходе метрик за допустимые пределы.

VictoriaMetrics извлекает метрики из целевых объектов по протоколу HTTP, поэтому в сервисах имеется специальная конечная точка для целей обмена данным с системой мониторинга.

Следующие бизнес-метрики доступны из продуктового [GUI Grafana](#):

1. MAU online (MV) — Сумма уникальных пользователей, которые были online в течение месяца.
2. DAU online (MV) — Сумма уникальных пользователей, которые были online в течение дня.
3. MAU получатели (MV) — Сумма уникальных пользователей, которые получили одно и более сообщений в течение месяца.
4. MAU отправители (MV) — Сумма уникальных пользователей, которые отправили одно и более сообщений в течение месяца.
5. DAU получатели (MV) — Сумма уникальных пользователей, которые получили одно и более сообщений в течение дня.
6. DAU отправители (MV) — Сумма уникальных пользователей, которые отправили одно и более сообщений в течение дня.
7. Количество сообщений за день — Сумма отправленных сообщений за день.
8. Количество звонков за день — Сумма совершенных звонков за день.
9. Количество сообщений за минуту — Количество отправленных сообщений за минуту.
10. Количество звонков за минуту — Количество состоявшихся звонков за минуту.
11. Активность за день — Отношение суммы отправленных сообщений к сумме уникальных пользователей, которые отправили одно и более сообщений; Отношение суммы уникальных пользователей, которые получили одно и более сообщений к сумме уникальных пользователей, которые отправили одно и более сообщений; Отношение суммы состоявшихся звонков к сумме уникальных пользователей, которые сделали один и более успешных звонков.
12. Группы и каналы — Сумма групп, в которых было отправлено одно и более сообщений в течение дня; Сумма групп, созданных за день.
13. Аккаунты и блокировки — Сумма аккаунтов, которые были online за день; Сумма аккаунтов, заблокированных за день.
14. Количество регистраций за день — Количество новых пользователей за день.

Групповые политики



Zastava

Сервис отвечает за взаимодействие с административной панелью VK WorkSpace и передачу событий групповых политик в сервисы Мессенджер и ВКС

Когда на пользователя/группу пользователей назначается или снимается групповая политика, или когда политика изменяется, сервис Pravda на стороне административной панели VK WorkSpace создает событие и отправляет в Kafka. Сервис Zastava разбирает сообщения от Kafka и направляет по протоколу IPROS в сервисы Мессенджер и ВКС.

Для конфигурирования доступа к целевым сервисам сервис Zastava использует хранилище конфигураций групповых политик в etcd (на стороне VK WorkSpace).

Данные хранит, используя БД Tarantool.

Отправляет запросы в следующие сервисы:

- Pacman (по протоколу IPROS) — для политики доступности разделов клиентского приложения VK WorkSpace.
- Chat-gp (по протоколу IPROS) — для политики обязательных чатов Мессенджер и ВКС.
- Vipper (по протоколу IPROS) — для политики привилегий.
- etcd (по протоколу TCP через pub) — получает конфигурацию групповых политик и инициализирует групповые политики.
- Kafka (по протоколу TCP) — для получения обновлений по групповым политикам.

Vipper

Сервис отвечает за реализацию групповой политики привилегий.

Получает сообщения от сервиса Zastava на изменение информации о принадлежности пользователя к группе в Keycloak. Далее отправляет запрос в сервис Nomail на обновление данных о пользователе в БД Tarantool.

Получает запросы от следующих сервисов (по протоколу IPROS):

- Godmod — при анонсировании групповой политики.
- Zastava — сообщения о новых действиях в рамках политики привилегий.

Отправляет запросы в следующие сервисы:

- Nomail (по протоколу IPROS) — для обновления информации о принадлежности пользователя к группе привилегированных пользователей.
- Keycloak (по протоколу HTTP) — для обновления информации о принадлежности пользователя к группе привилегированных пользователей.
- etcd (через pub по протоколу TCP) — инициализация конфигураций групповых политик.

Chat-gp

Сервис отвечает за реализацию групповой политики обязательных чатов и каналов.

Получает от сервиса Zastava сообщения с действиями, которые требуется выполнить (добавить пользователя в чат/удалить из чата) и выполняет эти действия.

Получает запросы от следующих сервисов (по протоколу IPROS):

- Zastava — сообщения о новых действиях в рамках политики обязательных чатов и каналов Мессенджер и ВКС.
- Godmod — при анонсировании групповой политики.

Отправляет запросы в следующие сервисы:

- Mchat-st (по протоколу IPROS) — для добавления/удаления юзеров в чаты/каналы в рамках политики обязательных чатов.
- etcd (через pub по протоколу TCP) — инициализация конфигураций групповых политик.

Расман

Сервис отвечает за реализацию групповой политики веб-аппов и мини-аппов.

Формирует список доступных пользователю мини-аппов с учётом их порядка для клиентского приложения VK WorkSpasena основе ответов от сервисов:

- Avon — передает информацию о всех мини-аппах инсталляции.

- Pravda (по протоколу HTTP через pub) — основной API для взаимодействия с сервисами групповых политик на стороне VK WorkSpace. Расман получает информацию об активной политике доступности разделов для пользователя.
- Pravda-go (по протоколу HTTP через pub) — API для чтения групповых политик пользователей.
- Godmod – для формирования политики по умолчанию с учетом конфигурационного файла сервиса Godmod.
- Scribl (по протоколу IPROS) — для отправки изменений о политике доступности разделов на клиентские приложения VK WorkSpace.
- etcd (по протоколу TCP через pub).

Получает запросы от следующих сервисов (по протоколу IPROS):

- Godmod — при анонсировании политики.
- Zastava — сообщения о новых действиях в рамках политики доступности разделов.
- Stentor — для получения информации о политике «по умолчанию».

Сервисы инфраструктуры

- Nginx.
- Orchestrator.
- Pushgateway.
- Vector.
- Weave.
- CoreDNS.
- Ctlr — хранит в себе и отдает сервисам топологию распределения данных по кластеру.
- Kittenhouse — локальный прокси сервис для БД ClickHouse, обеспечивающий буферизацию запросов, распределение нагрузки и переотправку данных.
- Krtek — сервис отправки клиентских логов.
- Consul — сервис для поддержания обнаружения сервисов и распределенного хранилища ключ-значение.
- Myteam-admin-backend — предоставляет API для администрирования мини-аппов. Средством администрирования является Панель администратора VK WorkSpace.
- Stdby — сервис хранения конфигураций. Данные хранит, используя БД KUST.

🕒 3 сентября 2026 г.