

Пояснения по поводу архитектуры системы

Почему была выбрана микросервисная архитектура?

Широкий функционал

В системе много функционала, которые могут логически быть не связаны друг с другом и в некоторых ситуациях даже никак не пересекаться

В системе много разных пользователей, причем это не просто разные роли с разными правами, а именно пользователи с разным функционалом: клиенты, воркеры, менеджеры (к ним еще могут относиться работники отдела проверки качества), кандидаты в воркеры и сотрудники. Все они работают с разными сущностями и по своему отдельному флоу

Интеграции с внешними системами

У каждой внешней системы могут быть свои способы интеграции, например, REST, SOAP и т.д. и было бы более удобно, если бы мы вынесли эти интеграции в разные независимые сервисы

Сервис загрузки внешних клиентов. Данный сервис предназначен чисто для интеграции с внешней системой – Harpur sat box, из которой мы будем выгружать всех пользователей в нашу систему. Данный сервис через брокер выгружает данные о клиентах в сервис Аутентификации и авторизации и в сервис по работе с услугами. Почему через брокер? Во-первых гарантия доставки данных, во-вторых, он может кидать сообщения в одну очередь из которой одновременно два сервиса будут вычитывать, в-третьих, при массовой загрузке новых пользователей брокер будет служить буфером в риковые нагрузки – сообщения будут просто накапливаться в очереди, а консьюмеры постепенно будут их обрабатывать.

Сервис по работе с услугами. Сервис, в котором менеджер сможет заводить/редактировать типы услуг, а клиент сможет заводить заявки на выполнение этих услуг. По сути клиент работает только с услугами/заявками у которых свой жизненный цикл по сравнению с теми джобами, которые делают воркеры (своя статусная модель, свои параметры и т.д.). Поэтому я подумал, что эти сервисы можно разделить между собой. Данный сервис в синхроне вызывает сервис по поиску воркеров для услуг, почему в синхроне – вызов по АПИ своего рода триггер для старта поиска воркера.

Сервис по поиску воркеров для услуг. Отдельный сервис, так как в нем живет отдельная логика и алгоритм по поиску воркеров. Так же, по сути, нигде в других сервисах особо не нужна информация об активных воркерах. Как только сервис нашел воркера для услуги он через брокер сообщает инфу об этом в сервис по работе с услугами для воркеров и в сервис для ставок. Почему через брокер? Гарантия доставки, нет необходимости быстро реагировать или что-то триггерить/вызывать при этом срочно + из одной очереди будут вычитывать сразу два сервиса

Сервис для найма воркеров. Отдельный сервис для работы с кандидатами, в котором много своеобразной логики: работа самого кандидата, менеджера, который может накидывать ему тесты и редактировать пул тестов, сама логика проверки тестов и принятия

решения о том, годен кандидат или нет. Сервис через брокер отдает инфу о новых воркерах в сервис по поиску воркеров для услуг. По сути не нагружаем сервис по поиску, который и без этого высоконагружен.

Сервис по работе с услугами для воркеров. Отдельный сервис в котором отдельная логика со своей статусной моделью и флоу, которые сильно отличаются от услуг. Интеграция в синхроне с сервисом отдела расходников. По сути по кнопке создаем заявку в другом сервисе, не хотелось бы задержек и возни с брокером когда на UI нужно четко отрисовать создалась ли заявка по кнопке или нет.

Сервис для отдела расходников. Тоже отдельный, так как там отдельные работники, которые тоже работают с отдельной сущностью у которой тоже своя статусная модель и свой флоу. Вынес отдельную интеграцию с сервисом, который интегрируется с компанией по производству печенек. Отдельный сервис на случай, если таких компаний в будущем будет больше или же мы захотим заказывать не только печенки у этой компании.

Биллинг. Отдельный сложный сервис. Думаю, что лучше работать в балансной схеме – т.е. у нас в системе у каждого клиента/воркера есть свой баланс. Почему так? А потому что выплачиваем мы деньги клиентам и воркерам раз в неделю и причем только тогда, когда штрафов меньше чем успехов и т.д. Так или иначе придется подсчитывать сколько денег нужно выплатить и я предлагаю эту сумму считать не тогда, когда нужно выплачивать, а каждый раз убавляя/прибавляя денег на баланс.

Также отдельные сервисы для интеграции с эквайрингом и с Золотой шляпой (в самом начале писал про внешние сервисы).

Сервис для ставок. Отдельная логика у сервиса, работает в черную, можно будет скрыть при аудите. Нет интеграций с биллингом.

Нотификейшен сервис и отдельные сервисы для составления писем и по отправки email. По сути логика работы такая: сервис, которому нужно отправить email-нотификацию кидает сообщение в очередь (все кидают в одну очередь), что нужно отправить по такому-то шаблону с такими-то данными письмо. Нотификейшен сервис идет в синхроне в сервис для составления письма, который на выходе отдает файл с письмом и далее эти данные кидаются в другую очередь уже из которой сервис по отправке писем вычитывает инфу и отправляет письмо. Вынес отдельно сервис по составлению писем, так как там тоже может быть отдельная логика в будущем по составлению кастомизированных шаблонов, не только текст, но и например формат файла и т.д. Ну и файлы эти могут отправляться не только по email.

Почему вообще отдельный нотификейшен сервис? Потому что в будущем могут быть не только нотификации по почте, но и через СМС и прочее – некий оркестратор.

Сервис аутентификации и авторизации. Безопасность, упрощение работы с пользователями (которых много, но у которых внутри в будущем могут быть разные роли и права), упрощение интеграций и т.д.

Отдельные клиенты. По сути объяснил это выше – разные пользователи с совершенно разными задачами и работающие с разными сущностями.

Ну и отдельный **прокси-сервис Менеджерка** для клиента в котором работают менеджеры и сотрудники отдела проверки качества. Нужен так как в слишком много сервисов бы иначе с фронта пришлось бы ходить и приходилось бы много логики делать на фронте, либо усложнять сервисы. А так вся логика ляжет на этот прокси-сервис.