

Домашнее задание №1

Диаграммы

- [MCF - Event Storming](#)
- [MCF - Data Model](#)
- [MCF - Communications](#)

Описание

- **Основной сервис-монолит** с модулями:
 - **Управления Тасками** - обеспечивает жизненный цикл тасок для воркеров
 - **Биллинга** - проводит еженедельный расчёт по выполненным таскам, применяет скидки для клиентов, штрафы для воркеров, выставляет счета, производит выплаты воркерам (через интеграцию с Golden Hat)
 - **Складского учёта расходников** - хранит данные об остатках, нахождении на складе, трекает заказы fur-tune печенек к Bakery-подрядчику
 - **Контроля Качества Работ** - ведёт учёт произведённых звонков и опросов клиентов по отменённым таскам, хранит выдвигаемые гипотезы по улучшению и результаты экспериментов по их применению
 - Единое хранилище данных (реляционная СУБД)
 - Очереди сообщений для асинхронного взаимодействия при выполнении задач (заявки в Пекарню, отправка переводов в Golden Hat, обсчёт каждого воркера при биллинге и т.д.)
- **Сервис найма воркеров**
 - публичный сайт, где можно откликнуться на вакансию и получить набор тестов для оценки пригодности кандидата
 - база данных сотрудников (можно использовать и как AAA сервис) - тут все и воркеры, и менеджеры, и кладовщики, и контролёры
- **Сервис-тотализатор**
 - скрытый от внешнего мира сервис с частным сайтом
 - доступ на сайт только у 15 менеджеров-участников
 - на сайте каждый менеджер может сделать по 1 ставке на каждую таску (успех/провал + сумма)
 - получает от монолита список тасок и обновление статусов по ним (асинхронно, через очередь сообщений)
 - при переводе конкретной таски в финальный статус (completed, cancelled) запускается процедура расчёта выигрыша ставок - результаты присылаются участникам по e-mail - учётчик записывает себе в тетрадку. Расчёты в БД не хранятся, чтобы ничего не вскрылось даже в случае утечке данных
- **Сервис отправки e-mail сообщений**
 - получает задания на отправку сообщений от всех остальных частей системы
 - хранит и управляет шаблонами сообщений, чтобы соблюдался единый корпоративный стиль
 - обеспечивает гарантированную отправку сообщения (повторные попытки пока не будет восстановлена связь с почтовым сервером)
 - умеет отправлять сообщения в несколько потоков при высокой нагрузке

Характер коммуникаций

- Сервис Найма -> Монолит - асинхронный, Сервис Найма выгружает данные о новом сотруднике, ответ Монолита на его работу никак не сказывается.
- Монолит -> Тотализатор - асинхронный, через очередь сообщений. Асинхронный, потому что ответ Тотализатора не блокирует работу Монолита. Может быть небольшой лаг между заведением таски в Монолите и попаданием ее в Тотализатор, но исходим из того, что время обработки очереди существенно меньше, чем время выполнения любой таски (дольше будут печенки доставляться)

- Вся коммуникация с Сервисом отправки сообщений - асинхронная. Сам характер связи по e-mail подразумевает асинхронность. Отправка писем не блокирует не один из сервисов в системе

•

Спорные места

- Размещение складского учета в монолите - может в будущем создавать проблемы, за счёт двойного объема запросов к API сервиса - со стороны воркеров и со стороны кладовщиков (нагрузкой от менеджеров и контроллеров пока пренебрегаем)
- Размещение биллинга в монолите - с ростом числа воркеров и тасок процесс еженедельного биллинга будет занимать все больше времени (процессорного в том числе), что будет негативно сказываться на работе остальных модулей в монолите Оба этих модуля - кандидаты на отдельные сервисы при условии развития бизнеса и необходимости масштабирования. На текущий момент технические требования покрываются легко и монолитом