**Предпосылка для выполнения задания:**

1. **на схеме в задании не указано, что API User передает в MES API какой-то файл с 3D-моделью, который MES API должен анализировать;**
2. **но судя по описанию проблем - расчет стоимости по запросу от API User - это такая же тяжеловесная операция, как и расчет стоимости по 3D-модели от Shop API;**
3. **Поэтому, предполагаем, что API User передает в MES API какой-то файл (или формат данных), время анализа которого сопоставимо с временем анализа 3D-модели;**
4. **Таким образом, расчет стоимости на основе запроса от API User также в среднем занимает у MES API 2-3 мин;**

## 1. Анализ проблем системы:

1. Освоение новых рынков сбыта (увеличение количества конечных потребителей (customer’ов)) =>
   1. увеличение нагрузки на сервисы: Shop API, CRM API, 3d files storage, MES API =>
   2. жалобы со стороны клиентов (customer’ов):
      1. долго не получают свои заказы - то есть заказ был сделан, обещали, что работы выполнят за 3 недели, прошло уже несколько месяцев, а работа все еще не выполнена;
   3. на что клиенты НЕ жалуются:
      1. НЕТ жалоб на скорость работы internet shop. То есть работа UI для оформления заказа пользователей устраивает
2. Освоение новых рынков сбыта (работа с другими продавцами-посредниками, которые сами продают изделия конечным потребителям (customer’ам)) =>
   1. открытие API нашей MES-системы другим продавцам-посредникам =>
   2. увеличение нагрузки на сервисы: MES API, CRM API =>
   3. жалобы со стороны пользователей API (API User):
      1. долго не получают заказы;
   4. жалобы со стороны клиентов (customer’ов):
      1. открытие API для сторонних продавцов произошло после “увеличения количества конечных потребителей за счет освоения новых рынков сбыта”;
      2. поэтому удалось отследить следующую динамику - конечные потребители (customer’ы) и до открытия API жаловались на долгое получение своих заказов. А после открытия API для API User’ов customor’ы начали еще больше жаловаться на просроченные заказы
   5. жалобы со стороны операторов (operator):
      1. дашборд с фильтрами заказов по статусам, который операторы используют для работы с заказами, долго прогружается;
      2. то есть операторы не могут оперативно отслеживать статусы взятых ими заказов;

Когда заказ считается оформленным:

1. с точки зрения customer:
   1. Customer **считает свой заказ оформленным, когда он нажал на кнопку “Сделать заказ”. То есть проблема нашей системы где-то после того, как заказ перешел в статус “SUBMITTED”.**
2. с точки зрения API User:
   1. API User считает свой заказ оформленным после того, как его собственный покупатель подтвердил оформление заказа. То есть:
      1. сначала API User приходит со своим форматом данных, по которому наш MES API считает стоимость заказа;
      2. далее, покупатель со стороны API User подтверждает заказ;
      3. API User повторно приходит в MES API, сообщая что заказ был подтвержден;

## 2. Существующие проблемные места

1. MES API - судя по жалобам со стороны пользователей системы (Customer, Api User, Operator) наиболее проблемным местом является MES API. Текущая реализации бэкенда в совокупности с текущей инфраструктурной конфигурацией не справляется со своими задачами, о чем свидетельствует:
   1. Customer - долго не получает **оформленные заказы** (то есть проблема во flow после статуса SUBMITTED)
   2. API User - долго не получают **оформленные заказы** (API User на своей стороне принимает заказы от своих покупателей, а на неш API приходит тогда, когда у него уже подтвердили оформление заказа. В нашей статусной модели API User приходит с SUBMITTED-заказами);
   3. Operator - проблема с отображением **оформленных заказов** в MES-системе (все заказы после статуса MANUFACTURING\_APPROVED);
2. судя по текущему описанию каких-то очевидных проблем здесь и сейчас у компании нет - все основные проблемы вокруг того, что после перехода заказа в статус SUBMITTED все потребители системы жалуются на проблемы - каждый на свои проблемы;

## 3. Потенциальные проблемные места

1. Shop API
   1. Судя по предоставленным данным, нагрузка от customer’ов растет линейно - в среднем в каждый месяц количество заказов увеличивается на 100. Увеличение количества оформленных заказов напрямую коррелирует со следующими метриками:
      1. DAU;
      2. RPS;
      3. QPS;
   2. Как может выглядеть типичное действие пользователя и количество запросов, которое нужно сделать к Shop API:
      1. создал новый заказ (INITIATED) - 3 запроса (например, пользователю прежде чем инициировать заказ нужно получить какую-то стартовую страницу с данными);
      2. создал заказ с помощью конструктора 3D-модели (FILE\_UPLOADED) (количество запросов зависит от реализации конструктора, но предположим, что как минимум нужно запросить сам конструктор, отправить результат и еще два запроса в запас. Весь остальной конструктор - это реализация на клиентской стороне. Итого - 4 запроса);
      3. подтвердил заказ (SUBMITTED - 3 запроса - сама кнопка и какие-то еще побочные действия)
      4. Итого, каждый оформленный заказ - это 3 + 4 + 3 = 10 запросов;
   3. также, учтем неоформленные заказы - далеко не все пользователи оформляют заказы, но используют Shop API. Предположим, что неоформивших заказ в пять раз больше, чем оформивших заказ. То есть на каждого оформившего заказ приходится пятеро неоформивших.
   4. тогда: увеличение количества оформленных заказов на 100 приводит к увеличению отправленных запросов на 1000 + 5000 = 6000.
   5. **То есть, нагрузка на Shop API растет на 6000 запросов в месяц**
      1. действительные цифры можно снять с Яндекс метрики, судя по описанию задания этот сервис подключен к интернет-магазину;
   6. увеличение RPS составит: 6000/30/12 = 17 запросов в секунду в месяц;
   7. на текущий момент это не выглядит критичным, но за этим стоит понаблюдать

## 4. Инициативы, необходимые для устранения нежелательных ситуаций

1. MES API:
   1. Вынести функционал расчета стоимости (Calculates price of the order) в отдельный сервис. Назовем этот сервис: **Price Calculation API.** MES APIбудет взаимодействовать с **Price Calculation API** асинхронно - то есть MES API асинхронно поручает **Price Calculation API** посчитать стоимость, а сам идет обрабатывать дальше другие запросы;
   2. Кеширование заказов - закешировать заказы с текущими статусами (описание см в разделе по кешированию);
   3. Обновить систему мониторинга:
      1. Сейчас подключена Яндекс Метрика, которая анализирует только сайт;
      2. необходимо добавить мониторинг со стороны запросов от API User
2. Shop API:
   1. внедрить мониторинг для отслеживания роста количества запросов со стороны customer’ов. На данный момент нет критической скорости роста количества запрсов от пользователей, но все может измениться и за этим нужно следить;
   2. внедрить клиентское кеширование для Shop и Shop API:
      1. одна из функциональностей Shop API: “provide list of shop items” - по сути, предоставить список изделий, доступных к продаже, которая включает в себя и количество доступных остатков к продаже;
      2. эти данные можно закешировать, проставив инвалидацию
3. Вся система:
   1. повышение наблюдаемости (observability) - добавление мониторинга, трейсинга для компонент системы;

## 5. Приоритет инициатив

1. **Приоритет 1** - MES API - вынести функционал расчета стоимости в отдельный сервис (**Price Calculation API**). Связать MES API с **Price Calculation API** асинхронно:
   1. расчет цены как на основе 3D-модели от customer’а, так и на основе данных от API User’а - это тяжеловесная для CPU операция (CPU-Bound);
   2. помимо этой тяжеловесной операции MES API занимается ответами на относительно легковесные запросы - функциональности: Assigns orders on operators и Shows list of orders;
   3. судя по flow процессов: Operator’ы работают с функциональностями Assigns orders on operators и Shows list of orders;
   4. таким образом, если вынести функциональность Calculates price of the order из MES API, то вполне возможно, что проблема operator’ов решится сама собой - чтобы это проверить нужно добавить мониторинг;

P.s. - чтобы не терять клиентов уже сейчас можно попробовать увеличить количество инстансов MES API в облачной инфраструктуре. До тех пор, пока не будет произведен вынос функциональности расчета цены в **Price Calculation API** количество инстансов MES API должно быть увеличено и установлен load balancer.

1. **Приоритет 2** - MES API и **Price Calculation API** - повысить наблюдаемость (observability) - добавить мониторинг, трейсинг:
   1. мониторинг MES API нужно добавлять параллельно с тем, как будет выноситься функционал Calculates price of the order в **Price Calculation API** - так как судя по жалобам именно место системы не справляется с нагрузкой;
   2. для только что добавленного сервиса **Price Calculation API** также вводим мониторинг метрик;
   3. также нужно посмотреть что там с логированием - возможно понадобится более подробное логирование;
2. **Приоритет 3** - Shop API - повысить наблюдаемость (observability) - добавить мониторинг, трейсинг:
   1. далее, повышаем наблюдаемость в сервисе Shop API - так как нагрузка на него растет каждый месяц, то необходимо следить за его метриками;

Остальные инициативы:

1. вся система - повысить наблюдаемость (observability) - добавить мониторинг, трейсинг, логирование
   1. судя по жалобам - все остальные сервисы чувствуют себя стабильнее, поэтому что-то срочно менять там не требуется, но нужно ввиду растущей популярности бизнеса нужно увеличить наблюдаемость системы, чтобы следить за метриками;
2. кеширование:
   1. MES API - в месте, где операторы жалуются на медленный дашборд можно рассмотреть вариант с кешированием;
   2. Shop API - ввиду растущей популярности сервиса имеет смысл рассмотреть кеширование в Shop API - для функциональности provide list of shop items
3. CRM API:
   1. жалоб на его работу пока нет, но в связи с ростом бизнеса проблемы могут и появится. Нужно следить за метриками;
4. повышение масштабируемости системы:
   1. на основе метрик смотрим как загружены сервисы и какие именно функциональности в них являются более нагруженными;
   2. в первую очередь увеличиваем количество инстансов сервисов (в том числе устанавливаем load balancer);
   3. далее, при необходимости, можно вынести функциональность из сервисов;
5. настройка CD/CDP:
   1. CI уже настроен, а на релизный стенд и прод-стенд деплой производится вручную;