# Мотивация

Мониторинг обеспечит следующие преимущества:

* Ускорение процесса поиска проблем и диагностики
* Предотвращение проблем с производительностью и надежностью системы и своевременное реагирование на них
* Анализ потребляемых ресурсов позволит увидеть пути и места их оптимизации
* Мониторинг времени отклика и реагирование на него позволит снизить неудовлетворенность пользователей
* Возможность оценки степени влияния обновлений, новых фич на стабильность и производительность системы

# Выбор подхода к мониторингу

В целом, нет необходимости строго придерживаться одного подхода к мониторингу, а наоборот, комбинировать их согласно здравому смыслу.

**RED**

Подойдёт для мониторинга API всех трёх систем. Фокус на запросах, успешности и времени обработки. Например,

* Время ответа API
* Коэффициент ошибок
* Количество запросов API

**USE**

Подойдёт для мониторинга MES в части тяжёлых операций обработки 3D-моделей и других аспектов других приложений. Например:

* Количество активных соединений
* Загрузка процессора и памяти
* Использование дискового пространства

**Четыре золотых сигнала**

* Размер очереди RabbitMQ
* Процент попадания в кеш
* Время выполнения запросов в БД

# Отслеживаемые метрики

В общем, все перечисленные метрики полезны для мониторинга и должны быть применены:

* Number of dead-letter-exchange letters in RabbitMQ
  + Показывает количество сообщений, которые не смогли быть обработаны
  + Ярлыки: Имя очереди, причина попадания в DLX
* Number of message in flight in RabbitMQ
  + Позволяет понять, сколько сообщений находится в обработке. Может показать недостаток потребителей или проблемы с производительностью
  + Ярлыки: Имя очереди
* Number of requests (RPS) for API
  + Мониторинг нагрузки на каждую из систем
  + Ярлыки**:** Имя сервиса, конечная точка API, HTTP-метод
* Response time (latency) for API
  + Оценка производительности и обнаружение задержек
  + Ярлыки**:** Имя сервиса, конечная точка API
* Number of HTTP 500 for API
  + Указывает на критические ошибки в API
  + Ярлыки**:** Имя сервиса, конечная точка API, причина ошибки.
* Number of simultanious sessions for API
  + Показывает количество одновременно активных пользователей, что помогает в планировании нагрузки
  + Ярлыки**:** Имя сервиса, регион пользователя
* CPU % for API
  + Отслеживает использование CPU для анализа и предотвращения перегрузки
  + Ярлыки**:** Имя узла, имя сервиса.
* Memory Utilisation for API
  + Позволяет выявлять утечки памяти и корректировать использование ресурсов
  + Ярлыки**:** Имя узла, имя сервиса
* Memory Utilisation for db instance
  + Позволяет оценить, достаточно ли ресурсов для базы данных.
  + Ярлыки**:** Имя базы данных
* Number of connections for shop db instance
  + Предотвращение исчерпания пула соединений и связанных с этим ошибок.
  + Ярлыки**:** Имя базы данных, тип подключения (read/write).
* Size of S3 storage
  + Мониторинг использования хранилища для своевременного управления объемом данных
  + Ярлыки**:** Имя хранилища
* Size of shop instance
  + Отслеживание роста базы данных
  + Ярлыки**:** Имя базы данных.
* Kb transferred (received) for API
  + Помогает понять уровень входящего трафика и его распределение.
  + Ярлыки**:** Имя сервиса, источник трафика.
* Kb provided (sent) for API
  + Анализирует исходящий трафик для выявления возможных аномалий или утечек данных.
  + Ярлыки**:** Имя сервиса, цель трафика.