# **Задание 4. Логирование**

## **Анализ системы в контексте планирования логирования**

### **Необходимые логи уровня INFO и ERROR по элементам системы**

1. **MES API**
   * **Создание заказа**:
     + Уровень: INFO
     + Данные: время, идентификатор клиента, детали заказа.
   * **Расчет стоимости заказа**:
     + Уровень: INFO
     + Данные: время, номер заказа, рассчитанная стоимость.
   * **Изменение статуса заказа в MES**:
     + Уровень: INFO
     + Данные: время, номер заказа, старый статус, новый статус.
   * **Ошибка выполнения заказа**:
     + Уровень: ERROR
     + Данные: время, номер заказа, сообщение об ошибке.
2. **Shop API**
   * **Создание заказа**:
     + Уровень: INFO
     + Данные: время, идентификатор клиента, номер заказа, детали заказа.
   * **Изменение статуса заказа**:
     + Уровень: INFO
     + Данные: время, идентификатор клиента, номер заказа, новый статус.
   * **Ошибка при загрузке 3D модели**:
     + Уровень: ERROR
     + Данные: время, номер заказа, сообщение об ошибке.
3. **CRM API** (опционально)
   * **Изменение статуса заказа**:
     + Уровень: INFO
     + Данные: время, идентификатор клиента, номер заказа, новый статус.
   * **Подтверждение отправки заказа**:
     + Уровень: INFO
     + Данные: время, номер заказа, статус подтверждения.
   * **Ошибка отправки заказа**:
     + Уровень: ERROR
     + Данные: время, номер заказа, сообщение об ошибке.

### **Использование других уровней логирования кроме INFO**

* **ERROR**: Необходимо использовать для логирования ошибок, которые могут повлиять на функциональность системы или вызвать сбой. Например, ошибки при загрузке 3D-моделей или ошибки при работе с RabbitMQ. Эти логи помогут выявить и устранить проблемы в работе системы.
* **WARN**: Можно использовать для предупреждений о потенциальных проблемах, которые не приводят к сбоям, но могут указывать на необходимость внимания. Например, если заказ не может быть обработан из-за временного ограничения.
* **DEBUG**: Можно использовать в dev-окружении в процессе разработки или анализа, особенно для сложных взаимодействий между компонентами (например, в MES API).

Элементы, для которых стоит применить логирование, выделены на схеме 4.1

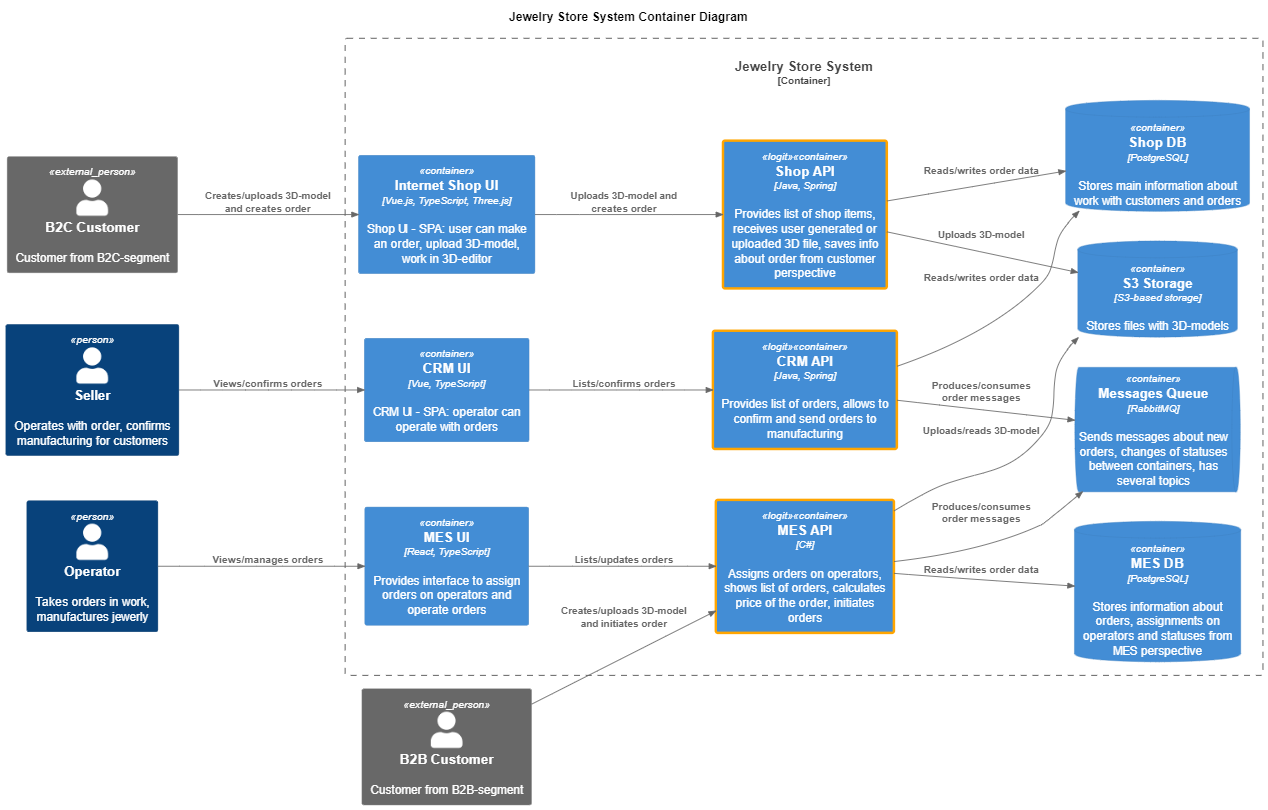


Схема 4.1 Контейнерная диаграмма ASIS для системы Jewelry Store System с выделением элементов для логирования

## **Мотивация**

Логирование необходимо для повышения надёжности системы и удобства для пользователей. Внедрение логирования решает несколько ключевых задач:

1. **Отладка и диагностика**: Логи позволяют разработчикам и инженерам наиболее точно и быстро выявлять и анализировать проблемы в работе системы, что значительно сокращает время на диагностику и исправление ошибок.
2. **Аудит и безопасность**: Логирование действий пользователей и бизнес-процессов позволяет отслеживать несанкционированные действия, возможные DDOS-атаки и обеспечивать безопасность системы.
3. **Улучшение пользовательского опыта**: Анализ логов помогает выявлять проблемные зоны в взаимодействии пользователей с системой, что позволяет улучшать интерфейсы и процессы.

### **Технические и бизнес-метрики, на которые повлияет внедрение логирования**

**Технические метрики:**

1. **Скорость нахождения и исправления ошибок**

* **Как влияет логирование**: Логирование позволяет оперативно выявлять причины ошибки в системе, таких как некорректная обработка заказов или сбои при расчёте стоимости. При возникновении проблемы логи дадут точную информацию о том, где произошла ошибка, что ускорит её диагностику.
* **Результат для компании**: Снижение времени на исправление критических багов позволит ускорить процесс обработки заказов, уменьшить количество жалоб клиентов и улучшить надёжность системы.

1. **Производительность системы**

* **Как влияет логирование**: Логирование даёт возможность анализировать производительность различных частей системы (например, расчёт стоимости в MES или взаимодействие с базой данных Shop). Это помогает выявлять узкие места и оптимизировать работу системы.
* **Результат для компании**: Оптимизация производительности сократит время выполнения операций и улучшит пользовательский опыт как клиентов, так и сотрудников, особенно в MES и CRM.

1. **Время расчёта стоимости**

* **Как влияет логирование**: Логи позволяют отслеживать длительность процессов расчёта стоимости в MES API для разных моделей. Это поможет определить, какие модели требуют больше ресурсов, и оптимизировать процесс.
* **Результат для компании**: Снижение времени расчёта стоимости сократит задержки в обработке заказов, особенно для сложных 3D-моделей, что положительно скажется на удовлетворённости клиентов.

**Бизнес-метрики:**

1. **Удовлетворённость клиентов**

* **Как влияет логирование**: Сокращение времени на нахождение и устранение проблем благодаря логированию уменьшит количество сбоев и задержек при обработке заказов. Это улучшит взаимодействие клиентов с системой и повысит их удовлетворённость.
* **Результат для компании**: Меньшее количество жалоб и более высокий уровень удовлетворённости клиентов приведут к увеличению повторных заказов и снижению оттока клиентов.

1. **Соблюдение SLA**

* **Как влияет логирование**: Логи позволяют выявлять причину существенной проблемы до того, как проблема начнёт приводить к массовым сбоям. Оперативное выявление причины и предложение, по крайней мере, временного обходного решения позволит обойтись без ограничения доступа пользователей к системе и сократить время на исправление последствий.
* **Результат для компании**: Снижение количества простоев системы позволит соблюсти SLA, обеспечить стабильную работу приложений и избежать потерь, связанных с задержками или простоями в производстве.

### **Приоритетные системы для настройки логирования и трейсинга**

Поскольку команда не сможет реализовать логирование и трейсинг всех систем одновременно, необходимо определить приоритетные направления:

**Логирование и трейсинг обязательны:**

1. **MES API.** Это наиболее сложная часть системы и потенциальное узкое место. Логирование необходимо для точечного контроля на сервисом. Без него также нельзя обойтись в виду запланированного разделения сервиса на несколько инстансов под различные виды задач.
2. **Shop API.** Онлайн-магазин является первой точкой контакта с клиентами. Для своевременного выявления проблем и их причин необходимо владеть информацией о заказе с самого начала его обработки. Без логирования здесь также нельзя обойтись в виду запланированного перенаправления запросов B2B-клиентов c MES API к Shop API.

**Можно ограничиться трейсингом:**

1. **CRM API** управляет обработкой заказов. Логи помогут лучше отслеживать статус заказов, выявлять проблемы и оптимизировать процессы обслуживания клиентов. Хотя функционал CRM API достаточно простой, и можно обойтись трейсингом.

## **Предлагаемое решение**

1. **Выбор системы логирования**Основные технологии для логирования (**ELK Stack**):
   * **Elasticsearch** будет использоваться для хранения и поиска логов.
   * **Logstash** будет собирать, обрабатывать и отправлять логи в Elasticsearch.
   * **Kibana** предоставит удобный интерфейс для визуализации и анализа логов.
2. **Необходимые доработки**Для интеграции ELK с сервисами необходимо добавить/настроить клиентские библиотеки Logstash для Shop API (Java/Spring), MES API (.NET) и CRM API (опционально, Java/Spring).
3. **Политика безопасности в отношении логов**
   * **Внедрение единой системы аутентификации (SSO)**: Для доступа к системе логирования только сотрудники компании смогут использовать свои корпоративные учетные записи. Это позволит централизовать управление доступом и сократить риски, связанные с потерянными или украденными учетными данными. Также можно предусмотреть двухфакторную аутентификацию.
   * **Использование системы ролей**: Доступ к логам будет предоставляться только сотрудникам с соответствующими ролями, например:
   * **Поддержка** (Support): могут видеть все логи, исключая чувствительные данные.
   * **Разработчики**: могут видеть только логи, связанным с их подсистемами, исключая чувствительные данные.
   * **Администраторы**: могут видеть все логи, включая чувствительные данные.
   * **Сегментация доступа по зонам безопасности**. Доступ к системе логирования будет ограничен только внутренними сетями компании, с минимизацией количества внешних точек входа. Сотрудники, работающие удаленно, смогут получить доступ только через VPN с шифрованием.
   * **Ограничение по IP-адресам**. Настройка белых списков IP-адресов для доступа к системе логирования. Это означает, что доступ к системе возможен только из сетей, находящихся под управлением компании.
   * **Журналирование действий**. Вся активность в системе логирования будет журналироваться, включая входы в систему, доступ к логам, изменения конфигурации и настройки системы. Это позволит отслеживать любые подозрительные действия и оперативно реагировать на потенциальные угрозы.
   * **Мониторинг и защита от аномалий**. Система должна иметь возможность обнаруживать аномальные паттерны запросов (например, подозрительно большое количество запросов на чтение данных), что может быть признаком попытки взлома или DDOS-атаки.
   * **Обновления безопасности**. Все компоненты системы логирования (ELK) должны регулярно обновляться для устранения уязвимостей и улучшения их безопасности.
4. **Работа с чувствительными данными**:  
   Логи не должны содержать чувствительные данные (например, пароли, номера кредитных карт). Если необходимо зафиксировать какую-либо информацию, содержащую чувствительные данные, она должна быть зашифрована или маскированной перед записью в логи.
5. **Политика хранения логов**
   * **Структура хранения**:
     + Для каждой системы (MES, CRM, Shop) будет создан отдельный индекс в Elasticsearch для логов.
     + Логи будут структурированы по типам (например, ошибки, информационные, отладочные).
   * **Сроки хранения**:
     + Логи будут храниться в системе в течение 90 дней.
     + По истечении этого срока старые логи будут автоматически удаляться, чтобы освободить место для новых.
   * **Размер логов**:
     + Необходимо установить ограничение на размер индекса — например, 30 ГБ на индекс, после чего будет происходить ротация логов.
     + Для поддержания производительности Elasticsearch рекомендуется настроить партиционирование индексов по временным меткам (например, ежедневно).
6. **Система анализа логов**Для превращения системы сбора логов в систему анализа логов необходимо реализовать выявление аномалий и алертинг.  
   Для этого потребуется установить сервис алертинга (ElastAlert), либо настроить существующий сервис (алертинг в Kibana), либо провести интеграцию с общим сервисом алертинга (Grafana Alerting). В нашем случае больше подойдёт последнее решение, т.к. оно не требует установки дополнительных сервисов, дополнительных расходов и позволяет настраивать все алерты централизованно.

Необходимо:

* + Настроить отправку данных из Elasticsearch в Prometheus.
  + Настроить соответствующие панели и дашборды в Grafana.
  + Настроить Grafana Alerting для алертинга по выявленным аномалиям:
    - Резкий рост количества логов уровня ERROR от одного или нескольких сервисов. Может указывать на проблему в коде или инфраструктуре.
    - Резкий рост количества логов одного типа (например, лог создания заказа) уровня INFO от одного из сервисов. Может указывать на DDoS-атаку или бот-атаку.
    - Отсутствие логов от одного или нескольких сервисов в течение заданного времени. Может указывать на проблему в коде или инфраструктуре.

Новые компоненты и связи отражены на схеме 4.2.

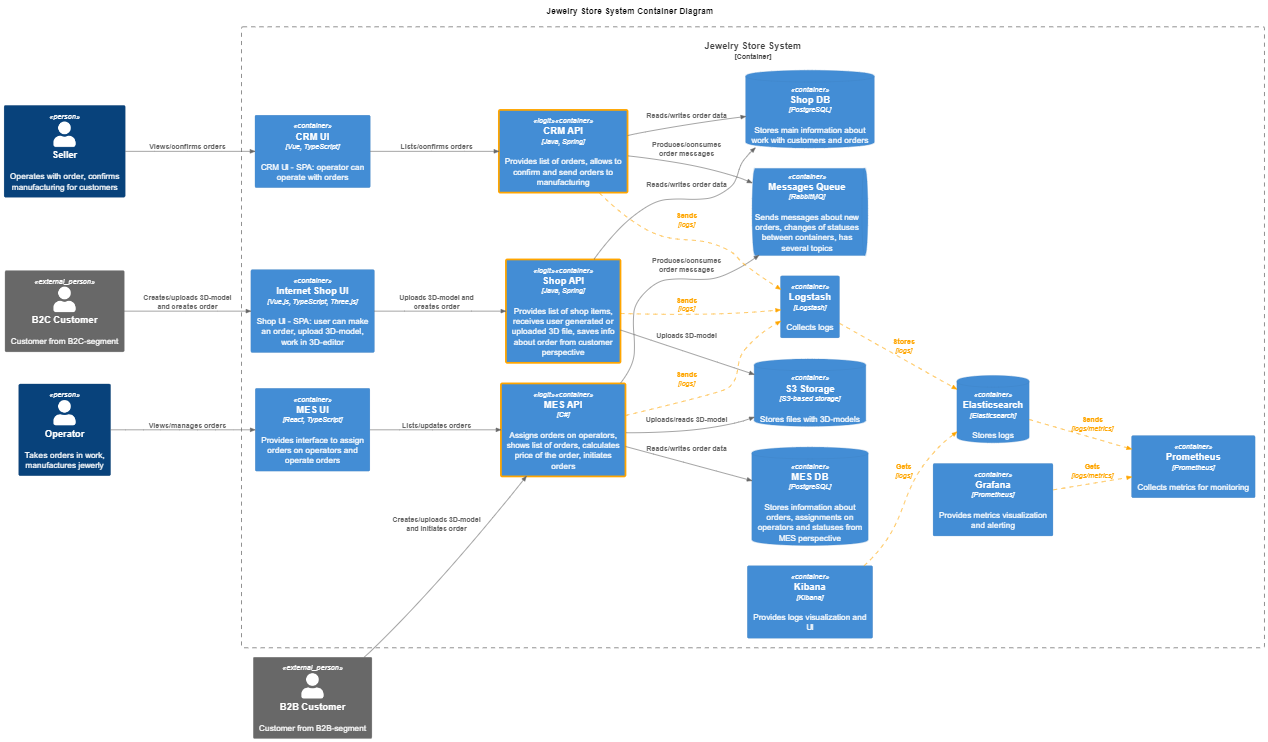


Схема 4.2 Контейнерная диаграмма для системы Jewelry Store System с новыми компонентами для логирования

## **Анализ критериев для выбора технологии для работы с логами**

Сравнение технологий по значимым критериям:

| **Критерии** | **ELK (Elasticsearch, Logstash, Kibana)** | **OpenSearch** | **Splunk** |
| --- | --- | --- | --- |
| **Лицензия** | Elastic License (+/-) | Apache License, Version 2.0 (+) | Проприетарная (-) |
| **Стоимость** | Бесплатна, но облачные DBaaS решения могут обойтись дороже из-за несвободной лицензии (+/-) | Бесплатна и с открытым исходным кодом (+) | Платная, стоимость может быть высокой (-) |
| **Функциональность** | Имеется вся необходимая функциональность (+) | Имеется вся необходимая функциональность, но может поддерживать меньше плагинов и интеграций (+/-) | Функциональность большая, но явно избыточная (+) |
| **Масштабируемость** | Хорошо масштабируется (+) | Хорошо масштабируется, но ещё не зарекомендовала себя в работе с большими объёмами данных (+/-) | Хорошо масштабируется, но это требует значительных затрат как на лицензирование, так и на инфраструктуру (-) |
| **Поддержка** | Подробная документация в открытом доступе, развитое сообщество, коммерческая поддержка для предприятий (+) | Поддерживается Amazon, но как относительно новая технология, имеет меньше сторонних ресурсов, чем ELK (+/-) | Профессиональная поддержка от производителя и активное сообщество, но доступ к полной поддержке может потребовать значительных затрат (+/-) |
| **Простота использования и интеграции** | Простая интеграция со множеством различных источников данных (+) | Простая интеграция со множеством различных источников данных, но как относительно новая технология, может потребовать дополнительных усилий для интеграции (+/-) | Проще в использовании только для больших организаций с готовыми ресурсами (-) |
| **Плюсы/Минусы** | 6/2 | 6/4 | 3/5 |

По соотношению плюсов и минусов и с учётом реальных потребностей компании наиболее подходящим решением однозначно будет ELK.