**Пункт 1.**

В качестве БД возьмем реляционную MySQL (или Postgres).

Схема данных:

-Таблица catalog (группы, категории, подкатегории)

-Таблица products (товары)

-Таблица warehouses (склады)

-Таблица product\_warehouses (количество товаров на складах)

-Таблица orders (заказы)

-Таблица users (пользователи)

-Таблица product\_fields (дополнительные поля товаров)

Хранение медиаконтента:

Медиаконтент будем хранить в отдельной файловой системе (например, Amazon S3 или Google Cloud Storage).

Обоснование:

В качестве бд выбрали реляционную, так как в подобных бд удобно хранить структурированные данные. В нашем случае это - структура каталога товаров, данные о товарах, складах, заказах и т.д.

В product\_fields сможем структурированно хранить дополнительные поля товаров, которые могут различаться. Это обеспечит гибкость и возможность использования сложных запросов.

Медиаконтент выносим в отдельную файловую систему для разгрузки основной бд и обеспечения высокой доступности и масштабируемости в будущем.

Плюсы:

1. Высокая производительность для типичных запросов к интернет-магазину
2. Поддержка транзакций и отношений между таблицами
3. Гибкость и расширяемость благодаря таблице product\_fields
4. Высокая доступность и масштабируемость медиаконтента

Минусы:

1. Более сложная настройка и администрирование по сравнению с NoSQL-базами данных
2. Может требовать больше ресурсов для хранения больших объемов данных

**Пункт 2.**

Взаимодействие компонентов:

1. Пользователи взаимодействуют с веб-сервером через пользовательский интерфейс.
2. Веб-сервер обращается к базе данных интернет-магазина для получения и сохранения данных о товарах, заказах и пользователях.
3. Административный модуль также обращается к базе данных интернет-магазина для создания и редактирования товаров.
4. Внешняя логистическая система используется для обработки доставки заказов.
5. Медиаконтент (изображения товаров) хранится в отдельной файловой системе.