Plan for Adding Salesforce Integration and Extending to Other Catalogs

1. Adding Salesforce Integration

To integrate Salesforce with the existing customer sync system, the following steps will be taken:

1. Create Salesforce Service Module:
    - Develop a new service file (salesforce_service.py) to handle API interactions with Salesforce, similar to the existing stripe_service.py.
    - Implement functions for customer creation, updates, and retrieval.
2. Modify Sync Workers:
    - Update the existing sync_worker.py to support processing Salesforce-related events in addition to Stripe.
    - Introduce event types such as customer_created_salesforce and customer_updated_salesforce.
3. Kafka Event Routing:
    - Extend the Kafka producer to push events to specific topics for Salesforce.
    - Modify the Kafka consumer to handle messages for both Stripe and Salesforce by checking event types.
4. Webhook Handling:
    - Configure a webhook endpoint (/salesforce/webhook) to receive real-time updates from Salesforce.
    - Implement processing logic to update the internal database accordingly.
5. Configuration Management:
    - Add Salesforce API credentials and settings in the settings.py file for easy configuration.

---

2. Extending to Other Catalogs (Invoices, Orders, etc.)

The system can be extended to support other entities such as invoices by following these approaches:

1. Modular Service Design:
    - Introduce new modules like invoice_service.py to handle interactions with external platforms (e.g., Stripe, Salesforce).
    - Define standard methods for CRUD operations across different catalogs.

2. Unified Event Handling:
    - Define common event structures for different catalogs (e.g., customer_event, invoice_event).
    - Ensure the Kafka producer and consumer can process different event types dynamically.
3. Scalable Database Schema:
    - Modify existing database models to accommodate new catalogs while maintaining relationships with customer records.
    - Use separate tables for invoices, orders, and other entities with proper indexing.
4. API Enhancements:
    - Introduce additional API endpoints for new catalogs to allow CRUD operations via RESTful services.
    - Maintain consistent API structure to ensure easy integration with front-end systems.
5. Logging and Monitoring:
    - Extend logging utilities to track sync operations for each catalog type separately.
    - Implement monitoring tools (e.g., Prometheus, Grafana) to analyze sync performance across different entities.