دانشگاه شهید بهشتی

دانشکده مهندسی و علوم کامپیوتر

گزارش فاز اول پروژه‌ی درس سیستم‌های نرم‌افزاری مقیاس‌وسیع (ULS)

طراحی و مدل‌سازی فرآیند
یکپارچه‌سازی سفارش‌های فروشگاه اینترنتی

گرایش نرم‌افزار

دانشجویان:

شبنم جلیل‌پور

محمدسعید زارع مهرجردی

استاد راهنما:

دکتر فریدون شمس

تیر ۱۴۰۵

فهرست مطالب

۱ مقدمه و تبیین مسئله ۱

۲ تعیین سبک و سطح یکپارچه‌سازی ۱

۳ معماری سیستم و الگوهای مسیریابی ۲

۴ تحلیل سناریوهای خطا و افزایش تاب‌آوری ۳

۵ پایش سیستم و گذرگاه کنترل ۴

۶ جمع‌بندی ۴

۱ مقدمه و تبیین مسئله

در سازمان‌های امروزی، سیستم‌های اطلاعاتی و برنامه‌های کاربردی به دلایل مختلف به‌صورت جزیره‌ای و ناهمگون توسعه یافته‌اند. یکپارچه‌سازی نرم‌افزارهای سازمان[1] ابزار و رویکردی است که به این برنامه‌ها اجازه می‌دهد تا فرآیندها و داده‌های خود را با یکدیگر به اشتراک بگذارند.

در این پروژه، سناریوی یک فروشگاه اینترنتی مطرح است که سفارش‌های مشتریان را در قالب فایل‌های متنی (شامل نام مشتری، محصول، تعداد و قیمت) در یک پوشه‌ی ورودی دریافت می‌کند. هدف فاز اول، طراحی و مدل‌سازی معماری این سیستم یکپارچه‌سازی به‌گونه‌ای است که عملیات خواندن، غنی‌سازی داده (افزودن زمان دریافت[2])، اعتبارسنجی، مدیریت خطاهای احتمالی و گزارش‌دهی دوره‌ای وضعیت سیستم را به‌صورت کاملاً خودکار، بدون توقف و مبتنی بر الگوهای استاندارد یکپارچه‌سازی سازمانی[3] انجام دهد.

۲ تعیین سبک و سطح یکپارچه‌سازی

بر اساس مفاهیم پایه‌ی درس، چهار سبک اصلی برای یکپارچه‌سازی وجود دارد:

- انتقال فایل[4]
- پایگاه‌داده‌ی مشترک[5]
- فراخوانی رویه از راه دور[6]
- پیام‌رسانی[7]

این پروژه در سطح داده و بر پایه‌ی ترکیبی از دو سبک معماری طراحی شده است:

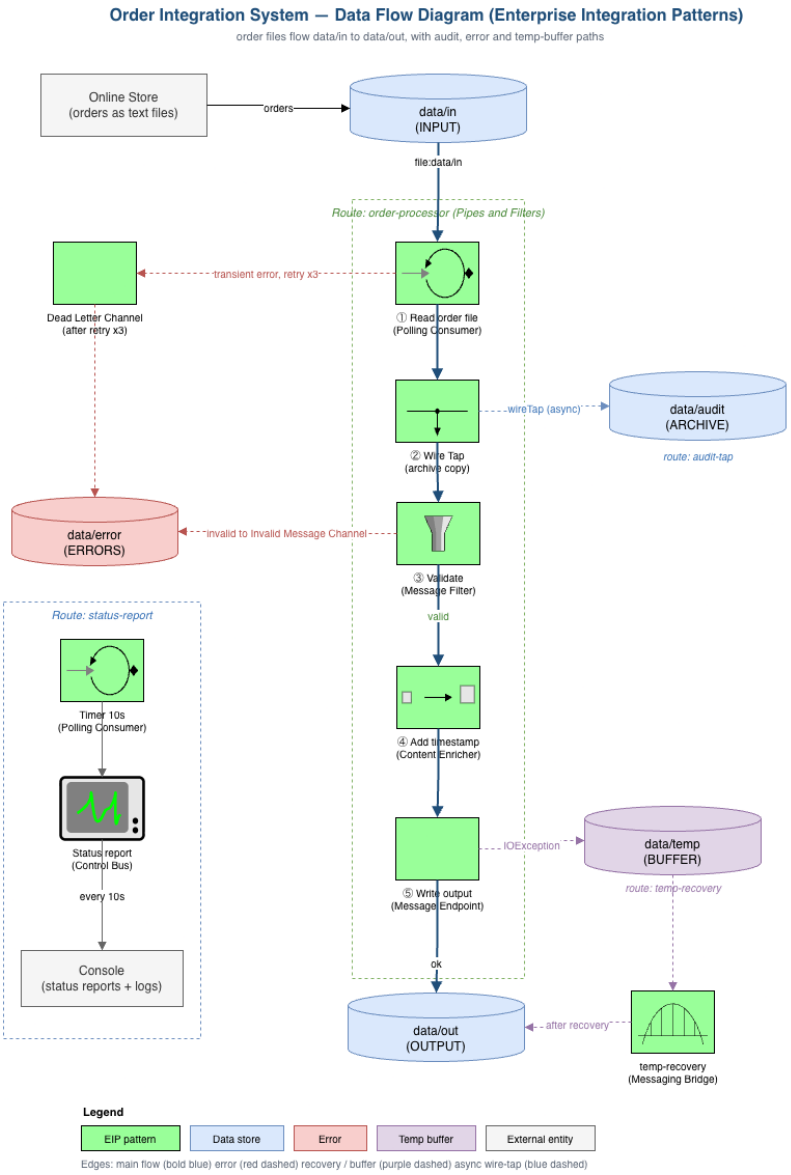
- **سبک انتقال فایل:** بستر فیزیکی تبادل داده‌ها بر اساس فایل است. سیستم‌های بیرونی سفارش‌ها را در پوشه‌ی ورودی می‌نویسند و خروجی نیز در قالب فایل ذخیره می‌شود. این سبک، عدم وابستگی زمانی و مکانی مناسبی ایجاد می‌کند.
- **سبک پیام‌رسانی:** برای پیاده‌سازی داخلی لایه‌ی میانی، از دیدگاه پیام‌رسانی استفاده شده است. چارچوب Apache Camel هر فایل ورودی را به‌عنوان یک پیام و مسیرهای پردازش را به‌عنوان کانال پیام در نظر می‌گیرد. این ترکیب به ما اجازه می‌دهد از مزایای اتصال سست[8] در این بستر بهره‌مند شویم.

---

[1] Enterprise Application Integration (EAI)
[2] Timestamp
[3] Enterprise Integration Patterns (EIP)
[4] File Transfer
[5] Shared Database
[6] Remote Procedure Call (RPC)
[7] Messaging
[8] Loose Coupling

۳ معماری سیستم و الگوهای مسیریابی

ساختار اصلی پردازش سفارش‌ها در این سیستم بر اساس الگوی لوله‌ها و فیلترها[9] مدل‌سازی شده است. در این الگو، هر گام پردازشی یک فیلتر مستقل است و پیام‌ها از طریق کانال‌ها بین این فیلترها جریان دارند. نمای کلی این معماری در تصویر ۱ نشان داده شده است.



تصویر ۱: نمودار جریان داده‌ی فرآیند یکپارچه‌سازی سفارش‌های فروشگاه اینترنتی

همان‌طور که در نمودار بالا مشاهده می‌شود، مؤلفه‌ها و فیلترهای اصلی این سیستم عبارت‌اند از:

- **مصرف‌کننده‌ی دوره‌ای:**[10] سیستم به‌صورت دوره‌ای پوشه‌ی ورودی (data/in) را سرکشی کرده و به‌محض یافتن فایل

[9] Pipes and Filters
[10] Polling Consumer

جدید، آن را به‌عنوان یک پیام وارد سیستم می‌کند.

- **شنود پیام:**[11] به‌منظور ثبت وقایع، بلافاصله پس از ورود فایل و پیش از هرگونه پردازش، یک کپی از پیام اصلی به‌صورت ناهمگام به پوشه‌ی بایگانی (data/audit) ارسال می‌شود تا لاگی از تمام پیام‌های دریافتی وجود داشته باشد. این الگو مسیریابی ساده‌ای است که پیام را تکثیر کرده و یک کپی از آن را بدون ایجاد تأخیر در مسیر اصلی، به‌صورت ناهمگام پردازش می‌کند.
- **فیلتر پیام:**[12] این مؤلفه وظیفه‌ی اعتبارسنجی را بر عهده دارد. در صورتی که فایل ورودی خالی یا ناقص باشد، پیام فیلتر شده و به مسیر خطا[13] در پوشه‌ی data/error هدایت می‌شود.
- **غنی‌ساز پیام:**[14] سفارش‌های معتبر وارد این گام می‌شوند. سیستم اطلاعات زمان دریافت را از ساعت داخلی سرور دریافت کرده و به محتوای پیام اضافه می‌کند.
- **نقطه‌ی پایانی پیام:**[15] در نهایت، پیام غنی‌شده به پوشه‌ی خروجی (data/out) که به‌عنوان نقطه‌ی پایانی سیستم عمل می‌کند، منتقل و ذخیره می‌گردد.
- **تحویل تضمین‌شده:**[16] در صورت خطای خروجی، داده‌ها موقتاً در پوشه‌ی جایگزین (data/temp) ذخیره می‌شوند تا پس از رفع مشکل ارسال گردند.
- **پایش و گزارش‌گیری:**[17] سرویسی موازی با فرآیند اصلی که به‌منظور مدیریت و نظارت، هر ۱۰ ثانیه گزارش وضعیت سامانه را در کنسول چاپ می‌کند.

۴ تحلیل سناریوهای خطا و افزایش تاب‌آوری

سیستم برای تضمین پایداری در مقیاس وسیع، سه سناریوی خطا را به کمک الگوهای EIP مدیریت می‌کند:

- **خطای خواندن/نوشتن:**[18] سیستم تا ۳ بار تلاش مجدد[19] می‌کند؛ در صورت تداوم خطا، پیام بدون متوقف‌کردن سیستم به کانال پیام‌های مرده در data/error منتقل می‌شود.
- **عدم دسترسی به پوشه‌ی خروجی:** در این صورت داده‌ها موقتاً در پوشه‌ی واسط (data/temp) ذخیره شده و پس از رفع مشکل، به مسیر اصلی بازمی‌گردند.
- **فایل خالی یا ناقص:** توسط الگوی کانال پیام‌های نامعتبر[20]، این فایل‌ها جداسازی شده و مستقیماً به پوشه‌ی خطا فرستاده می‌شوند تا پردازش سایر فایل‌ها بدون وقفه ادامه یابد.

[11] Wire Tap
[12] Message Filter
[13] Dead Letter Channel
[14] Content Enricher
[15] Message Endpoint
[16] Guaranteed Delivery
[17] Control Bus
[18] IOException
[19] Retry / Redelivery
[20] Invalid Message Channel

۵ پایش سیستم و گذرگاه کنترل

به‌منظور نظارت بر سلامت و کارایی سیستم، یک فرآیند موازی و مستقل بر اساس الگوی گذرگاه کنترل طراحی شده است. این سرویس به کمک یک کامپوننت تایمر، هر ۱۰ ثانیه یک‌بار فعال شده و معیارهای حیاتی سیستم (تعداد فایل‌های موفق، ناموفق و موقت) را که در شمارنده‌های ایمن از نظر هم‌روندی ذخیره شده‌اند، در کنسول نمایش می‌دهد تا مدیران سیستم از زنده‌بودن فرآیند مطمئن شوند.

۶ جمع‌بندی

در این فاز از پروژه، سناریوی فروشگاه اینترنتی به‌عنوان یک یکپارچه‌سازی درون‌سازمانی[۲۱] در سطح داده و بر پایه‌ی الگوی استخراج، تبدیل و بارگذاری[۲۲] مدل‌سازی شد. این معماری، ترکیبی از بستر فیزیکی انتقال فایل و منطق پیام‌رسانی است که با چارچوب Apache Camel مدیریت می‌شود. خروجی‌های این فاز — نمودار جریان داده، شناسایی مؤلفه‌ها و کانال‌های ارتباطی، و همین گزارش توصیفی — پایه‌ی مستقیم پیاده‌سازی فاز دوم را فراهم می‌کنند.

[21] Application-to-Application (A2A)

[22] Extract, Transform, Load (ETL) — Population Pattern