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 سرویس‌ها و نقاط اتصال میان آن‌ها ۴

۵ تحلیل سناریوهای خطا و افزایش تاب‌آوری ۵

۶ پایش سیستم و گذرگاه کنترل ۵

۷ جمع‌بندی ۵

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 است، طوری که عملیات خواندن، غنی‌سازی داده (افزودن زمان دریافت[2])، اعتبارسنجی، مدیریت خطاهای احتمالی و گزارش‌دهی دوره‌ای وضعیت سیستم را به‌صورت کاملا 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]

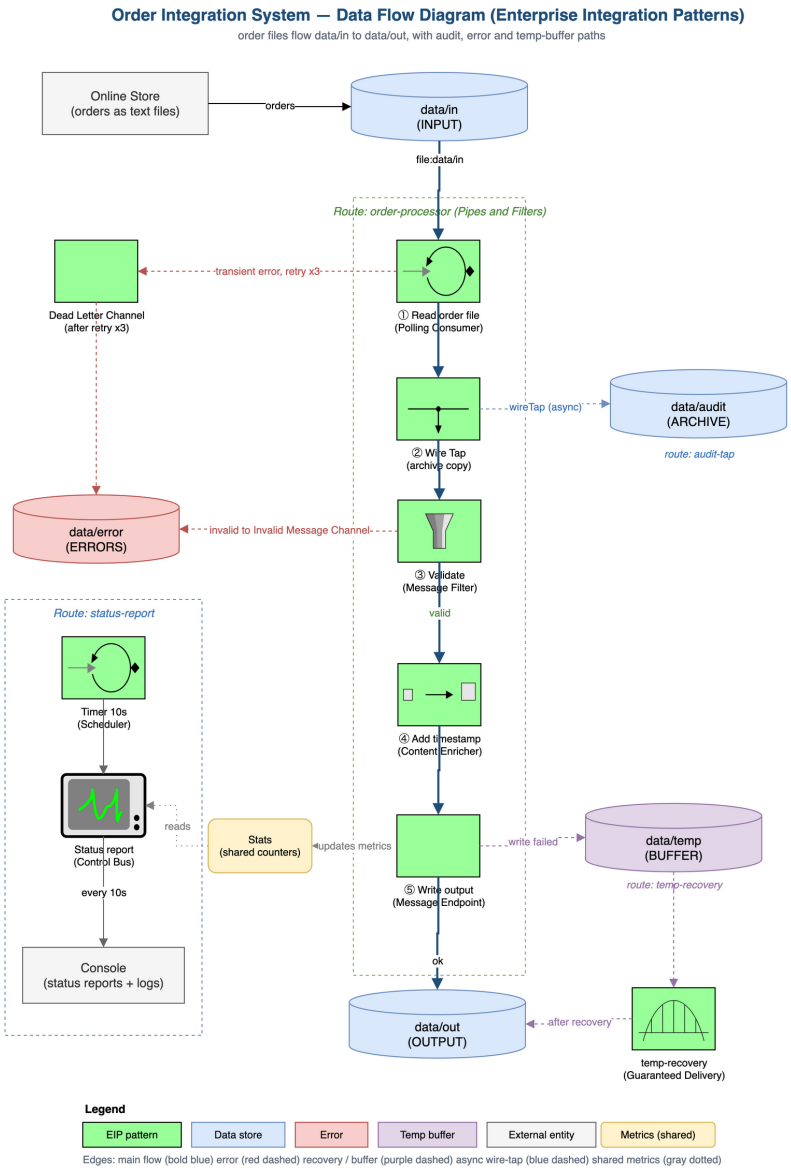
این پروژه در سطح داده و بر پایه‌ی ترکیبی از دو سبک معماری طراحی شده است:

- **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 یک پیام و مسیرهای پردازش را کانال پیام در نظر می‌گیرد. با این ترکیب می‌توانیم از مزایای اتصال سست[8] در این بستر استفاده کنیم.

[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 (حاشیه‌نویسی‌شده با الگوهای EIP)

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 جدید، آن را به‌عنوان یک پیام وارد سیستم می‌کند.

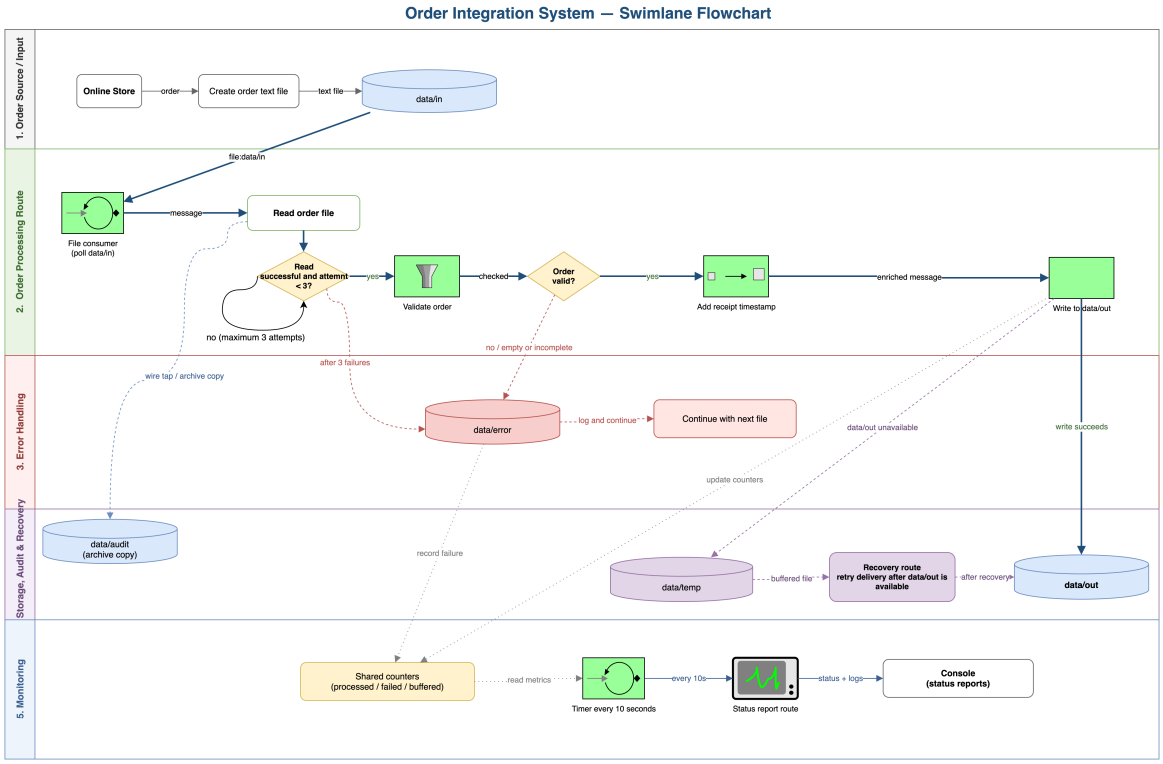
[9] Pipes and Filters
[10] Polling Consumer

- **شنود پیام:**[11] برای نگه‌داشتن سابقه، درست پس از خواندن فایل و پیش از اعتبارسنجی و غنی‌سازی، یک کپی از پیام به‌صورت ناهمگام به پوشه‌ی بایگانی (data/audit) فرستاده می‌شود تا از همه‌ی پیام‌های دریافتی یک لاگ داشته باشیم. این پوشه یک انتخاب طراحی اختیاری است و چون کپی ناهمگام انجام می‌شود، مسیر اصلی را کند نمی‌کند.
- **اعتبارسنجی و فیلتر پیام:**[12] این مؤلفه کار اعتبارسنجی را انجام می‌دهد. در صورتی که فایل ورودی خالی یا ناقص باشد، پیام نامعتبر شناخته شده و از طریق الگوی کانال پیام‌های نامعتبر[13] به پوشه‌ی data/error هدایت می‌شود؛ در غیر این صورت به گام بعد می‌رود.
- **غنی‌ساز پیام:**[14] سفارش‌های معتبر وارد این گام می‌شوند. سیستم زمان دریافت را از ساعت داخلی سرور (به‌عنوان منبع غنی‌سازی از محیط) می‌گیرد و به محتوای پیام اضافه می‌کند.
- **نقطه‌ی پایانی پیام:**[15] در نهایت، پیام غنی‌شده به پوشه‌ی خروجی (data/out) که نقطه‌ی پایانی سیستم است منتقل و ذخیره می‌شود.
- **تحویل تضمین‌شده:**[16] در صورت خطای خروجی، داده‌ها در پوشه‌ی موقت (data/temp) ذخیره می‌شوند و مسیر temp-recovery پس از رفع مشکل آن‌ها را به پوشه‌ی خروجی برمی‌گرداند.
- **پایش و گزارش‌گیری:**[17] سرویسی موازی با فرآیند اصلی که برای نظارت، هر ۱۰ ثانیه گزارش وضعیت سامانه را در کنسول چاپ می‌کند.

دو مسیر خطای این سیستم — کانال پیام‌های نامعتبر (برای فایل‌های خالی یا ناقص) و کانال پیام‌های مرده (برای خطاهای موقت پس از سه تلاش) — دو الگوی جدا هستند که در عمل در یک پوشه‌ی واحد (data/error) کنار هم قرار می‌گیرند. جزئیات این دو، همراه با جعبه‌های Dead Letter Channel و temp-recovery در نمودار، در بخش تحلیل سناریوهای خطا توضیح داده شده است.

در ادامه، نمودار Swimlane تکمیلی فرآیند یکپارچه‌سازی سفارش‌ها در تصویر ۲ ارائه شده است.

[11] Wire Tap
[12] Message Filter
[13] Invalid Message Channel
[14] Content Enricher
[15] Message Endpoint
[16] Guaranteed Delivery
[17] Control Bus



تصویر ۲: نمودار Swimlane فرآیند یکپارچه‌سازی سفارش‌های فروشگاه اینترنتی

۴ سرویس‌ها و نقاط اتصال میان آن‌ها

سیستم از سه سرویس تشکیل شده که چارچوب Apache Camel آن‌ها را به هم وصل و با هم هماهنگ می‌کند:

- **سرویس فایل:** مسئول خواندن سفارش‌ها از پوشه‌ی ورودی (data/in) و نوشتن خروجی در پوشه‌های data/out، data/audit، data/error و data/temp است. نقاط اتصال این سرویس، نقاط پایانی فایل[۱۸] در ابتدا و انتهای مسیرها هستند.
- **سرویس پردازشگر:** منطق اعتبارسنجی و غنی‌سازی (افزودن زمان دریافت) را اجرا می‌کند و در میانه‌ی مسیر order-processor از طریق کانال‌های داخلی به سرویس فایل متصل می‌شود.
- **سرویس زمان‌بندی:** مسیر مستقل status-report را اجرا می‌کند و از طریق گذرگاه کنترل، خروجی‌اش را در کنسول نمایش می‌دهد.

دو وابستگی اصلی هم وجود دارد: مسیر audit-tap که از طریق کانال مستقیم[۱۹] به سرویس فایل وصل است، و مسیر temp-recovery که میان پوشه‌ی موقت و پوشه‌ی خروجی رابط است. سرویس زمان‌بندی از دو سرویس دیگر مستقل است و فقط شمارنده‌های مشترک را می‌خواند؛ همین اتصال سست باعث می‌شود پردازش در برابر خطاها بدون توقف ادامه پیدا کند.

[18] File Endpoints
[19] direct:audit

۵ تحلیل سناریوهای خطا و افزایش تاب‌آوری

برای پایدارماندن در مقیاس وسیع، سیستم سه سناریوی خطا را با کمک الگوهای EIP مدیریت می‌کند:

- **تأخیر یا خطای موقت در خواندن:** در صورت بروز تأخیر یا خطای موقت هنگام خواندن فایل، سازوکار تحویل مجدد[۲۰] تا ۳ بار تلاش می‌کند؛ در صورت تداوم خطا، پیام بدون متوقف‌کردن سیستم به کانال پیام‌های مرده در data/error منتقل می‌شود.
- **عدم دسترسی به پوشه‌ی خروجی:** در صورت شکست در نوشتن روی پوشه‌ی خروجی، داده‌ها در پوشه‌ی موقت (data/temp) ذخیره می‌شوند. فایل تا زمانی که پوشه‌ی خروجی در دسترس نباشد در data/temp باقی می‌ماند و در هر چرخه‌ی بازیابی مجددا تلاش می‌شود؛ پس از رفع مشکل، به مسیر اصلی بازمی‌گردد.
- **فایل خالی یا ناقص:** توسط الگوی کانال پیام‌های نامعتبر[۲۱]، این فایل‌ها جداسازی شده و مستقیما به پوشه‌ی خطا فرستاده می‌شوند تا پردازش سایر فایل‌ها بدون وقفه ادامه یابد.

۶ پایش سیستم و گذرگاه کنترل

برای نظارت بر سلامت و کارایی سیستم، یک فرآیند موازی و مستقل بر پایه‌ی 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.

نکته این است که مسیر پایش و مسیر پردازش هیچ کانال پیامی با هم ندارند و کاملا مستقل‌اند؛ تنها راه ارتباطشان همین شمارنده‌های مشترک است که در نمودار با عنصر Stats نشان داده شده: مسیر پردازش این شمارنده‌ها را به‌روزرسانی می‌کند و مسیر پایش آن‌ها را می‌خواند. چون این ارتباط از جنس حالت مشترک است و نه پیام، در نمودار به‌جای کانال پیام با یال‌های نقطه‌چین خاکستری رسم شده است. همین جدایی باعث می‌شود پایش حتی وقتی مسیر اصلی متوقف یا کند شود باز هم کار کند و زنده‌بودن سامانه را گزارش دهد — که دقیقا هدف الگوی گذرگاه کنترل است.

۷ جمع‌بندی

در این فاز از پروژه، سناریوی فروشگاه اینترنتی به‌عنوان یک یکپارچه‌سازی درون‌سازمانی[۲۲] در سطح داده و بر پایه‌ی رویکرد استخراج، تبدیل و بارگذاری[۲۳] 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.

[20] Retry / Redelivery
[21] Invalid Message Channel
[22] Application-to-Application (A2A)
[23] Extract, Transform, Load (ETL)