**مستند نهایی معماری – یکپارچهسازی فروشگاهآنالین با سیستم حسابداری محک**

**C1 – System Context .1**

**:توضیح کوتاه**

این الیه بازیگران خارجی و سیستم اصلی را نشان میدهد. کاربران (مشتریها) و مدیر سیستم برای اعالنها، وSMS با فروشگاه آنالین تعامل دارند. درگاه پرداخت برای تراکنشها، سرویسسیستم حسابداری محک برای همگامسازی مالی حضور دارد. این نمودار دیدی سطح باال ازاکوسیستم کلی سیستم ارائه میدهد و مشخص میکند که فروشگاه آنالین ما با چه سیستمهاو موجودیتهای خارجی در ارتباط است.

**):External Actors( بازیگران خارجی**

|  |  |  |
| --- | --- | --- |
| کاربران نهایی که از فروشگاه آنالین بازدید کرده، محصوالت را | **):Customers( مشتریان** | • |

مشاهده، انتخاب و خریداری میکنند. تعامالت اصلی آنها شامل مرور محصوالت،افزودن به سبد خرید، ثبت سفارش و پرداخت است.

مسئول مدیریت فروشگاه آنالین، شامل**):System Administrator( مدیر سیستم** •مدیریت محصوالت، سفارشات، کاربران، تنظیمات و نظارت بر عملکرد سیستم.

|  |  |  |
| --- | --- | --- |
| سرویس خارجی که مسئولیت پردازش امن | **):Payment Gateway( درگاه پرداخت** | • |

تراکنشهای مالی و دریافت وجه از مشتریان را بر عهده دارد.

|  |  |  |
| --- | --- | --- |
| سرویس خارجی برای ارسال اعالنهای مربوط به | **:SMS )SMS Service( سرویس** | • |

سفارش، پرداخت، ارسال و سایر رویدادهای مهم به مشتریان و مدیران.

سیستم مالی و حسابداری**):Mahak Accounting System( سیستم حسابداری محک** • خارجی که وظیفه ثبت و مدیریت تراکنشهای مالی، فاکتورها، انبارداری و سایر امورحسابداری را بر عهده دارد. این سیستم منبع اصلی دادههای مالی و حسابداری است که نیاز به همگامسازی با فروشگاه آنالین دارد.

**):Main System( سیستم اصلی**

سیستم اصلی که شامل تمام بخشهای مورد نیاز برای**):Online Store( فروشگاه آنالین** • برای مشتریان و مدیران، مدیریتUI/UX عملیات یک فروشگاه اینترنتی، از جمله

محصوالت، مدیریت سفارشات، سبد خرید، سیستم پرداخت، مدیریت کاربران و منطقتجاری است.

**):Interactions( ارتباطات**

مشتریان با رابط کاربری فروشگاه آنالین تعامل دارند. • مدیران سیستم نیز از طریق رابط کاربری مدیریتی با فروشگاه آنالین تعامل دارند. • فروشگاه آنالین با درگاه پرداخت برای انجام عملیات پرداخت ارتباط برقرار میکند. • برای ارسال اعالنها ارتباط برقرار میکند.SMS فروشگاه آنالین با سرویس •فروشگاه آنالین با سیستم حسابداری محک برای همگامسازی دادههای مالی و عملیاتی •(مانند ثبت فاکتورها، بهروزرسانی وضعیت سفارش، مدیریت موجودی) ارتباط برقرار میکند. این ارتباط بخش کلیدی پروژه حاضر است.

**تصویر:**   
Placeholder   
*(این تصویر یک نمای کلی از تعامالت سیستم اصلی با بازیگران خارجی را نمایش میدهد. در" به عنوان سیستم مرکزی قرار گرفته و فلشها نشاندهنده جریانOnline Store" ،این تصویرMahak" ", وCustomers", "Administrator", "Payment Gateway", "SMS Service" ارتباط با" هستند.)Accounting System*

**C2 – Container Diagram .2**

**:توضیح کوتاه**

،API Gateway/BFF، PostgreSQL نمایش تمام سرویسها/کانتینرهای اصلی معماری شامل وRabbitMQ، Mahak Sync Worker، Mahak API Adapter، Object Storage )MinIO/S3( . ارتباط و جریان داده بین این اجزا نمایش داده شده است. این نمودار معماریFrontend مانند میکروسرویسها یا برنامههایDeployable Units ًسیستم را در سطح کانتینرها (معموالمستقل) نشان میدهد.

**):Containers( کانتینرها**

|  |  |  |
| --- | --- | --- |
| . رابط کاربری فروشگاه آنالین که برای مشتریان و مدیران قابل دسترسی است | **:Frontend** | • |

این کانتینر مسئول نمایش اطالعات، دریافت ورودی کاربر و ارسال درخواستها به است.API Gateway از طریقBackend **:API Gateway / BFF )Backend for Frontend(** نقطه ورودی واحد برای تمامی • . این کانتینر وظایف مسیریابی درخواستها بهFrontend درخواستهای ) و مجوزدهیAuthentication( میکروسرویسهای مناسب، احراز هویت

. را بر عهده داردFrontend )، و همچنین تجمیع یا تبدیل دادهها برایAuthorization( بهینه شدهAPI تضمین میکند که هر رابط کاربری (مثال ً وب، موبایل) یکBFF وجودمختص خود را دریافت کند.

**(مثال):Online Store Core Service** یک سرویس اصلی که منطق تجاری هسته • فروشگاه آنالین را پیادهسازی میکند (مدیریت محصوالت، سبد خرید، سفارشات، ) تعامل دارد.PostgreSQL( کاربران). این سرویس معموال ً با دیتابیس اصلی **:Mahak Sync Worker** •،Mahak سرویس تخصصی مسئول دریافت داده از سیستم  
 پردازش آن، و اعمال تغییرات در سیستم فروشگاه آنالین یا بالعکس. این کانتینر با برای ذخیره دادههایPostgreSQL و باMahak برای ارتباط باMahak API Adapter .همگامسازی شده تعامل دارد**:Mahak API Adapter** کانتینری که وظیفه واسطهگری و تبدیل درخواستها و پاسخهای • را بر عهده دارد. این کانتینرMahak های خارجی سیستمAPI بین معماری داخلی ما و سیستمAPI مشکالت مربوط به پروتکلها، فرمت دادهها و مدیریت خطا در سطح را مدیریت میکند.Mahak **:PostgreSQL )Database(** پایگاه داده رابطهای اصلی که دادههای فروشگاه آنالین • (محصوالت، مشتریان، سفارشات، تاریخچه تراکنشها و غیره) را ذخیره میکند.

**:RabbitMQ )Message Broker(** سیستم صف پیام که برای ارتباط ناهمگام بین •سرویسها استفاده میشود. این سیستم برای اطمینان از تحویل پیامها، پیادهسازی برایDead-Letter Queues )DLQ( وRetry ، مدیریت صفهایPub/Sub الگوهای پردازش خطا حیاتی است.

**:Object Storage )MinIO/S3 Compatible(** سرویس ذخیرهسازی اشیاء که برای •نگهداری فایلهای بزرگ مانند تصاویر محصوالت، مستندات، یا الگها استفاده میشود.

**):Interactions( ارتباطات**

ارتباط برقرار میکند.API Gateway باFrontend • Online Store مربوطه (مانندBackend درخواستها را به سرویسهایAPI Gateway • .) مسیریابی میکندCore Service . برای ذخیره و بازیابی دادهها تعامل دارندPostgreSQL باBackend سرویسهای • ارتباط برقرارMahak با سیستمMahak API Adapter از طریقMahak Sync Worker • .میکند دریافت کرده و به پردازش ادامه میدهد.RabbitMQ پیامها را ازMahak Sync Worker • . ذخیره میکندPostgreSQL نتایج پردازش خود را درMahak Sync Worker • و سایر سرویسهای احتمالی (برای ارسالMahak Sync Worker بینRabbitMQ • .رویدادها یا صفبندی وظایف) ارتباط برقرار میکند برای ذخیره فایلهای رسانهای مورد استفاده قرار میگیرد.Object Storage •

**:تصویر**  
Placeholder

*(این تصویر ساختار کانتینرها و ارتباطات اصلی بین آنها را نشان میدهد. کانتینرها به صورتبلوکهای مجزا نمایش داده شدهاند و خطوط ارتباطی جریان داده و فراخوانیها را مشخصمیکنند.)*

**C3 – Component Diagram )Mahak Sync .3 )Worker**

**:توضیح کوتاه**

RowVersion Tracker، Bulk شامل ماژولهایMahak Sync Worker جزئیات داخلیUpsert Engine )Conflict Resolution(، Logical Delete ،)Paging (باFetcher/Saver . این نمودار به بررسیData Transformer )، وDLQ (باHandler، Retry & Error Manager .) میپردازدMahak Sync Worker( اجزای داخلی یک کانتینر کلیدی

**:Mahak Sync Worker )Internal Components( اجزای داخلی**

**:Data Transformer** به فرمتMahak API مسئول تبدیل فرمت دادههای دریافتی از •قابل فهم و استفاده برای سیستم داخلی ما و بالعکس. این شامل نگاشت فیلدها، تبدیل انواع داده و استانداردسازی است.

)Bulk( ماژولی که مسئولیت استخراج دستهای**:Bulk Fetcher/Saver )with Paging(** • و ذخیره دستهای آنها در دیتابیسAPI Adapter از طریقMahak دادهها از سیستم (صفحهبندی) برای مدیریت حجم باالی دادههاPaging محلی را بر عهده دارد. استفاده از ضروری است.Memory Overload و جلوگیری از **:RowVersion Tracker** یاRowVersion( مکانیزمی برای پیگیری نسخههای دادهها • دریافت میشود. این به ما امکان میدهد تا تشخیص دهیمMahak ) که ازTimestampکدام رکوردها تغییر کردهاند و نیاز به بهروزرسانی دارند، و از بارگذاری مجدد دادههای بدون تغییر جلوگیری میکند.

•  
**:Upsert Engine )Conflict Resolution(** قلب منطق همگامسازی. این موتور مسئول در دیتابیس محلی بر اساس دادههایUpsert" )Update or Insert(" انجام عملیات) را پیادهسازی میکند؛Conflict Resolution( دریافتی است. همچنین منطق حل تعارضمثال ً در صورت بروز تداخل بین دادههای جدید و دادههای موجود، تصمیم میگیرد کهکدام نسخه اولویت دارد.

**:Logical Delete Handler**) دادهها. بهSoft Delete( مسئولیت مدیریت حذف منطقی •" را در رکورد مربوطه تنظیمIsDeleted" جای حذف فیزیکی رکوردها، این ماژول یک فلگ میکند تا نشان دهد دادهها دیگر فعال نیستند اما تاریخچه آنها حفظ شود. این برای حفظ یکپارچگی دادهها و امکان بازیابی ضروری است.

|  |  |
| --- | --- |
| **:Retry & Error Manager )with DLQ(** بخشی حیاتی برای مدیریت خطاها و پایداری) باRetry( سیستم. این ماژول وظیفه دارد تا عملیات ناموفق را مجددا ً امتحان کند | • |

استفاده از مکانیزمهای صفبندی. در صورت شکست مکرر یک عملیات، آن را به یک منتقل میکند تا توسط تیم فنی بررسی و رسیدگیDead-Letter Queue )DLQ( صفشود. این از دست رفتن دادهها و توقف فرآیند همگامسازی جلوگیری میکند.

**:Scheduler Integration**) هماهنگScheduler( ماژولی که با سیستم زمانبندی •میشود تا فرآیند همگامسازی را در فواصل زمانی مشخص اجرا کند.

**):Data Flow( جریان داده**

را برای شروع فرآیند همگامسازی فراخوانی میکند.Scheduler، Mahak Sync Worker .1، دادههای جدید یا تغییرRowVersion Tracker با استفاده ازBulk Fetcher/Saver .2 .) واکشی میکندMahak API Adapter (از طریقMahak API یافته را از دادهها را به فرمت داخلی تبدیل میکند.Data Transformer .3، دادههای تبدیل شده را دریافت کرده و با استفاده از منطق حل تعارضUpsert Engine .4 . را در دیتابیس انجام میدهدUpsert عملیات

|  |  |
| --- | --- |
| دادههای حذف شده منطقی را مدیریت میکند.Logical Delete Handler  وارد عمل شده وRetry & Error Manager ،در صورت بروز هرگونه خطا در طول فرآیند منتقل میکند.DLQ فرآیند را مجددا ً امتحان میکند یا به | .5 .6 |

**تصویر:**   
Placeholder  
 *را با اجزای مختلف و نحوه تعاملMahak Sync Worker (این تصویر ساختار داخلی کانتینرآنها با یکدیگر نشان میدهد. فلشها جریان پردازش و داده را در داخل این کانتینر مشخصمیکنند.)*

**C4 – Code / Implementation Details .4**

**:توضیح کوتاه**

Go، PostgreSQL 15، RabbitMQ، JWT یاNET 8. تکنولوژیها و ابزار استفاده شده شامل. این نمودارHybrid برای اسکیل، و استفاده از معماریAuth، MinIO/S3، Kubernetes .) مورد استفاده در کل معماری را مشخص میکندTech Stack( جزئیات فنی و پشته فناوری

**:)Technologies & Tools( تکنولوژیها و ابزار**

**:Backend Language/Framework** •**:NET 8 / Go.** هر دو زبان گزینههای قدرتمندی برای توسعه میکروسرویسها ◦

هستند.

،)، اکوسیستم قویASP.NET Core( ارائه دهنده فریمورک غنی**:NET 8.** ▪

. مناسب برای توسعهCloud-Native عملکرد باال، و پشتیبانی عالی از توسعه.Mahak Sync Worker وAPI Gateway، Backend Services **:Go** زبان سبک، با کارایی باال، مناسب برای توسعه سرویسهای ▪

NET. باال و مقیاسپذیری. میتواند جایگزین یا مکملConcurrencyباباشد.

**:Database** •**:PostgreSQL 15** پایگاه داده رابطهای قدرتمند، قابل اعتماد، متنباز و با پشتیبانی ◦

.Replication وJSONB، Partitioning گسترده از ویژگیهای پیشرفته مانندانتخاب ایدهآل برای دادههای ساختاریافته و نیمهساختاریافته.

**:Message Broker** •**:RabbitMQ** قابل اعتماد و با قابلیتهای فراوان که ازMessage Broker یک ◦

) پشتیبانی میکند. برایPub/Sub، Queues الگوهای مختلف پیامرسانی (مانند بسیار مناسب است.DLQ وRetry اجرای ناهمگامسازی، مدیریت **:API Gateway / Authentication** •**:JWT Auth )JSON Web Token(** استاندارد امن برای انتقال اطالعات بین طرفین ◦

API . برای احراز هویت و مجوزدهی درJSON به صورت امن به عنوان یک شیء و سرویسها استفاده میشود.Gateway **:Object Storage** •**:MinIO / S3 Compatible** سرور متنباز با کاراییObject Storage یکMinIO ◦

، سازگار است. امکان ذخیرهسازی تصاویر محصوالتS3 آمازونAPI باال است که بافایلهای الگ و سایر دادههای غیرساختاریافته را فراهم میکند.

**:Orchestration & Deployment** •،)Deployment( پلتفرم متنباز برای اتوماسیون استقرار**:Kubernetes** ◦

) و مدیریت برنامههای کاربردی کانتینری. برای اطمینان از درScaling( مقیاسدهی و مقیاسپذیری خودکارSelf-healing ،)High Availability( دسترس بودن باالسرویسها حیاتی است.

|  |  |
| --- | --- |
| **:Architecture Style :Hybrid Architecture** ترکیبی از رویکردهای مختلف، که در اینجا احتماال ً به | •  ◦ |

معنای استفاده از معماری میکروسرویس برای بخشهای اصلی و منطق تجاری، و ازMahak ) با سیستمAsync Integration( استفاده از یکپارچهسازی ناهمگام است. این رویکرد انعطافپذیری و مقیاسپذیری راMessage Broker طریقفراهم میکند.

**:)Rationale( دالیل انتخاب**

امکان مقیاسدهی افقی و عمودیKubernetes **):Scalability( قابلیت مقیاسپذیری** • نیز برای توسعه سرویسهای مقیاسپذیرGo وNET Core. .سرویسها را فراهم میکند بهینه شدهاند.

باKubernetes ، وDLQ وRetry با ویژگیهایRabbitMQ **):Reliability( قابلیت اطمینان** •، به افزایش قابلیت اطمینان سیستمRolling Updates وSelf-healing مکانیزمهای کمک میکنند.

معماری میکروسرویس امکان توسعه، استقرار و**):Flexibility( انعطافپذیری** •Go یاNET 8. مقیاسدهی مستقل هر سرویس را میدهد. انتخاب زبانهای مدرن مانند نیز این انعطافپذیری را افزایش میدهد.

**):Data Management( مدیریت داده** • یک دیتابیس قوی برای مدیریتPostgreSQL   
 راهکاری استانداردMinIO/S3 . استMahak Sync Worker دادههای فروشگاه آنالین و برای ذخیرهسازی فایل است.

راهی امن و استاندارد برای مدیریت احراز هویت است.JWT Auth **):Security( امنیت** •

**:تصویر**  
Placeholder   
*(این تصویر نمایانگر پشته فناوری مورد استفاده در معماری کلی پروژه است. اجزای مختلف، زبانهای برنامهنویسی و ابزارهای استقرار نمایش دادهMessage Broker ،مانند پایگاه دادهشدهاند.)*

**Sequence Diagram – BulkSync Simple Flow .5**

**:توضیح کوتاه**

نسخه سادهشده جریان همگامسازی دادهها برای ارائه به تیم غیر فنی. شامل فازهای درخواست، دریافت پاسخ، و ذخیرهسازی در دیتابیس محلی. این نمودار به زبان سادهMahak API داده ازنحوه کارکرد همگامسازی دستهای را توضیح میدهد.

**):Key Phases( فازهای اصلی**

**:Mahak API )Request Data from Mahak API( درخواست داده از** .1

درون آن) تشخیص میدهد که نیاز بهScheduler سیستم فروشگاه آنالین (یا یک ◦

وجود دارد.Mahak همگامسازی دادهها با سیستم

|  |  |
| --- | --- |
| ) ارسال میشود تاMahak API Adapter (از طریقMahak API درخواستی بهمجموعهای از دادههای مورد نیاز (مثال ً لیست محصوالت یا فاکتورهای جدید) | ◦ |

واکشی شود. این درخواست شامل پارامترهایی برای تعیین بازه زمانی یا فیلترهای

دیگر است.

|  |  |
| --- | --- |
| **):Receive Response( دریافت پاسخ**  پاسخ درخواست را ارسال میکند. این پاسخ شامل دادههایMahak API ) ارائهXML یاJSON درخواستی است که معموال ً در قالبی استاندارد (مانند | .2 ◦ |

میشود.

) دریافت شود اگر حجم دادهها زیادPaging( ممکن است پاسخ در چندین بخش ◦

باشد.

**):Process and Store in Local Database( پردازش و ذخیرهسازی در دیتابیس محلی** .3

دادههای دریافتی ابتدا در صورت نیاز پردازش یا تبدیل میشوند (مثال ً فرمتدهی ◦

مجدد).

سپس، این دادهها در پایگاه داده محلی فروشگاه آنالین ذخیره میشوند. بسته به ◦

" (درج دادههای جدید) یاInsert" ماهیت دادهها، این عملیات میتواند

" (بهروزرسانی دادههای موجود) باشد.Update"

**:)Participants( شرکتکنندگان**

. نهادی که درخواست همگامسازی را آغاز میکند**:Scheduler/Online Store** •

.Mahak API واسط ارتباط با**:Mahak API Adapter** •

. سیستم حسابداری محک**:Mahak API** •

. پایگاه داده فروشگاه آنالین**:Local Database** •

**):Simple Flow( جریان ساده**

|  |  |  |  |
| --- | --- | --- | --- |
| Mahak API Adapter: درخواست | >- | Scheduler/Online Store | .1 |

)Fetch Data Request( واکشی دادهها

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| API )Send API : ارسال درخواست | Mahak API | >- | Mahak API Adapter | .2 |

)Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Response( : پاسخ حاوی دادهها | Mahak API Adapter | >- | Mahak API | .3 |

)with Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mahak Sync (یا | Scheduler/Online Store | >- | Mahak API Adapter | .4 |

)Forward Fetched Data( ): ارسال دادههای واکشی شدهWorker

|  |  |  |  |
| --- | --- | --- | --- |
| Local Database: عملیات ذخیرهسازی | >- | Scheduler/Online Store | .5 |

)Save Operation - Insert/Update(

**:تصویر**

Placeholder   
*(این نمودار توالی سادهای از ارتباطات بین موجودیتها برای همگامسازی دستهای را نشانمیدهد، با تمرکز بر جنبههای کلی جریان.)*

**Sequence Diagram – BulkSync Detailed Flow .6**

**:توضیح کوتاه**

،Deleted Flag ، مدیریتRowVersion شامل استفاده ازBulk Sync جزئیات فنی فرآیندTimeout/Rate ، و سناریوهایDLQ وRetry ، پیادهسازیConflict Resolution باUpsert .. این نمودار به الیههای فنی و پیچیدگیهای همگامسازی میپردازدLimit

**:)Technical Details( جزئیات فنی**

|  |  |  |
| --- | --- | --- |
| برای جلوگیری از پردازش دادههای قدیمی یا از دست رفته، هر | **:RowVersion استفاده از** | • |

) است.Timestamp یاRowVersion دارای یک شناسه نسخه (مانندMahak رکورد در دریافت شده استفاده میکند تاRowVersion در درخواستهای بعدی، سیستم از آخرینفقط دادههای جدیدتر یا تغییر یافته را درخواست کند.

)Logical Delete( ممکن است از فلگ حذف منطقیMahak **:Deleted Flag مدیریت** •برای نشان دادن اینکه یک رکورد دیگر فعال نیست، استفاده کند. سیستم ما نیز باید اینفلگ را در هنگام همگامسازی رعایت کرده و در پایگاه داده محلی خود نیز به طور منطقی رکورد را غیرفعال کند.

**:Conflict Resolution باUpsert** هنگام ذخیره دادهها، اگر رکوردی با همان کلید اصلی • در پایگاه داده محلی وجود داشته باشد، سیستم باید تصمیم بگیرد که آیا آن را بهروزرسانی کند یا خیر. اگر دادههای دریافتی و دادههای موجود در پایگاه داده محلیتفاوتهای قابل توجهی داشته باشند (مثال ً تاریخچه تغییرات متفاوت)، ممکن است نیاز ) باشد.Conflict Resolution( به منطق پیچیدهتری برای حل تعارض

|  |  |  |
| --- | --- | --- |
| با خطا مواجه شود، یا اگر پردازشMahak API اگر ارتباط با | **:DLQ وRetry پیادهسازی** | • |

یک دسته از دادهها با مشکل روبرو شود (مانند خطای اعتبارسنجی، یا مشکل در ) را برای آنRetries( دیتابیس)، فرآیند نباید متوقف شود. سیستم باید تالشهای مجددعملیات انجام دهد. اگر پس از چندین تالش، عملیات همچنان ناموفق باشد، آن را به یک منتقل میکند تا بعدا ً توسط یک تیم فنی بررسی شود.Dead-Letter Queue )DLQ( ممکن است محدودیت زمانیMahak API **:Timeout/Rate Limit سناریوهای** • )Rate Limit( ) برای پاسخگویی یا محدودیت تعداد درخواست در واحد زمانTimeout( ،داشته باشد. سیستم ما باید بتواند این شرایط را مدیریت کرده و در صورت نیاز درخواستها را با تاخیر ارسال کند یا تالشهای مجدد با فاصله زمانی بیشتر انجام دهد.

**:)Participants( شرکتکنندگان**

. مسئول اصلی اجرای فرآیند همگامسازی**:Mahak Sync Worker** •

.Mahak API واسط ارتباطی با**:Mahak API Adapter** •

. سرویس خارجی**:Mahak API** •

.DLQ وRetry برای مدیریت**:Message Broker )RabbitMQ(** •

. پایگاه داده فروشگاه آنالین**:Local Database** •

**):Detailed Flow( جریان دقیق**

|  |  |  |  |
| --- | --- | --- | --- |
| Mahak API Adapter: درخواست واکشی | >- | Mahak Sync Worker | .1 |

RowVersion )Fetch Data Request with Last RowVersion( دادهها با آخرین

|  |  |  |  |
| --- | --- | --- | --- |
| Mahak API، با در نظرAPI : ارسال درخواست | >- | Mahak API Adapter | .2 |

Rate Limits )Send API Request, respecting Rate Limits( گرفتن

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RowVersion : پاسخ حاوی دادهها و | Mahak API Adapter | >- | Mahak API | .3 |

)Response with Data & New RowVersion( جدید

|  |  |  |  |
| --- | --- | --- | --- |
| به& Retry | ، دهدAPI Timeout اگرMahak Sync Worker **:Scenario** | | ◦ |
|  |  |
| . اطالع میدهد | | Error Manager |
| Mahak Sync Worker): دریافت دادهها در بستههاBulk Fetcher/Saver( | | | .4 |

)Receive Data in Batches(

|  |  |  |  |
| --- | --- | --- | --- |
| )Transform Data( ): تبدیل دادههاData Transformer( | | Mahak Sync Worker | .5 |
| |  | | --- | | Local | | درUpsert ): تالش برایUpsert Engine( | Mahak Sync Worker | .6 |

|  |  |  |
| --- | --- | --- |
| Database | |  |
| ،)Conflict ناموفق بود (مانندUpsert اگر**:Scenario** | | ◦ |
| Conflict Resolution ) از مکانیزمUpsert Engine( | Mahak Sync Worker |

.استفاده میکند

|  |  |  |
| --- | --- | --- |
| ،)Network Issue خطا داد (مانند | اگرLocal Database **:Scenario** | ◦ |

Mahak Sync Worker) پیام خطا را پردازشRetry & Error Manager(

.میکند

Mahak Sync Worker): اعمال حذف منطقی درLogical Delete Handler( .7

.صورت لزوم

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **مدیریت خطا:** | | | | .8 ◦ |
| Retry & Error( | | در صورت خطا در هر مرحله، Mahak Sync Worker | |
|  |
| . میفرستد | Message Broker | | درRetry ) عملیات را به صفManager |
| در DLQ پس از چند بار تالش ناموفق، عملیات بهMessage Broker منتقل | | | | ◦ |

میشود.

**تصویر:**

Placeholder

*(این نمودار جزئیات فنی و مراحل پردازش در هنگام همگامسازی دستهای را نشان میدهد، با*

*تمرکز بر مدیریت خطا، نسخهها و تعارضات.)*

**. تکنولوژیها و دلیل انتخاب7**

انتخاب تکنولوژیها برای هر پروژه، تأثیر بسزایی بر موفقیت، مقیاسپذیری، نگهداری و کارایی

آن دارد. در این پروژه، دالیل زیر مبنای انتخاب تکنولوژیهای مطرح شده بودهاند:

**:PostgreSQL 15** •

کاهش هزینههای الیسنس و امکان دسترسی به کد منبع برای**:رایگان و متنباز** ◦

سفارشیسازی در صورت نیاز.

|  |  |  |
| --- | --- | --- |
| اطمینان از اجرای آسان و یکپارچه در | **:Kubernetes/سازگاری با لینوکس** | ◦ |

محیطهای مدرن استقرار.

امکان ذخیره و کوئری گرفتن از دادههای نیمهساختاریافته**:JSONB پشتیبانی از** ◦

ها) به صورت کارآمد در کنار دادههای رابطهای.API (مانند دادههای دریافتی از

،Full-Text Search، GIS Extension، Partitioning مانند**قابلیتهای پیشرفته:** ◦

و انواع دادههای سفارشی که انعطافپذیری باالیی را ارائه میدهند.Replication

یکی از پایدارترین و قابل اعتمادترینPostgreSQL **:قابلیت اطمینان و بلوغ** ◦

.سیستمهای مدیریت پایگاه داده رابطهای در جهان است

**:RabbitMQ** •

امکان انتشار پیامها به چندین مشترک، که برای**:Pub/Sub پشتیبانی از** ◦

اطالعرسانی رویدادها بین سرویسهای مختلف بسیار مفید است.

|  |  |  |
| --- | --- | --- |
| مکانیزم داخلی برای مدیریت خطاها و تالشهای مجدد، | **:DLQ وRetry صفهای** | ◦ |

که به طور قابل توجهی قابلیت اطمینان فرآیندهای ناهمگام را افزایش میدهد.

پروتکلهای انتقال پیام مطمئن، اطمینان حاصل میکنند**تضمین تحویل پیامها:** ◦

که پیامها حتی در صورت بروز مشکل در شبکه یا سرویس، از بین نمیروند.

**قابلیت کار با پروتکلهای مختلف:**MQTT وAMQP، STOMP پشتیبانی از ◦

انعطافپذیری را در انتخاب روشهای ارتباطی فراهم میکند.

قابلیت مقیاسدهی افقی برای مدیریت حجم باالی پیامها.**مقیاسپذیری:** ◦

**:MinIO / S3 Compatible** •

ایدهآل برای ذخیرهسازی فایلهای حجیم مانند تصاویر**:راهکار ذخیرهسازی شیءگرا** ◦

محصوالت، ویدئوها، یا فایلهای الگ که نیازی به کوئریهای پیچیده ندارند.

|  |  |  |
| --- | --- | --- |
| کار میکنند،AWS S3 امکان استفاده از ابزارها و کتابخانههایی که با | **:S3 سازگار با** | ◦ |

و همچنین سهولت مهاجرت به سرویسهای ابری دیگر در آینده.

|  |  |  |
| --- | --- | --- |
| کنترل کامل بر دادهها و زیرساخت | **):On-Premise( متنباز و قابل استقرار در محل** | ◦ |

ذخیرهسازی.

طراحی شده برای مدیریت حجم عظیمی از دادهها.**مقیاسپذیری باال:** ◦

**:NET 8 / Go.** •

هر دو زبان، به خصوص در نسخههای جدیدتر، عملکردی بسیار باال در**:عملکرد باال** ◦

اجرای کد و مدیریت درخواستها ارائه میدهند که برای یک فروشگاه آنالین و

سرویسهای همگامسازی حیاتی است.

**سازگاری با ساختار ماژوالر و میکروسرویس:**ASP.NET فریمورکهای مدرن مانند ◦

، توسعه و مدیریت میکروسرویسها را تسهیلGo ، و ماهیتNET 8. درCore

.میکنند

**اکوسیستم قوی:**Go . دارای جامعه کاربری بزرگ و کتابخانههای فراوان استNET. ◦

.نیز به دلیل سادگی و کارایی باال محبوبیت زیادی دارد

امکان توسعه و استقرار بر روی**:Cross-Platform قابلیت توسعه** ◦

).Windows, Linux, macOS( سیستمعاملهای مختلف

|  |  |  |
| --- | --- | --- |
| استاندارد صنعتی برای احراز هویت بر اساس توکن، که | **:JWT Auth** | • |
| **:احراز هویت امن و ساده** | ◦ |

) را بین کالینت و سرور فراهم میکند.Stateless( امکان ارتباط بدون نیاز به جلسه

|  |  |  |
| --- | --- | --- |
| اطالعات الزم برای احراز هویت را در خود دارند وJWT توکنهای | **قابلیت حمل:** | ◦ |

میتوانند بین سرویسهای مختلف منتقل شوند.

تضمین عدم دستکاری توکن و اطمینان از منبع آن.**امضای دیجیتال:** ◦

**:Kubernetes** •

امکان خودکارسازی استقرار، مقیاسدهی، و**:مدیریت و اسکیل سرویسها** ◦

مدیریت برنامههای کاربردی در کانتینرها.

**:Zero Downtime Deploy** امکان بهروزرسانی سرویسها بدون ایجاد وقفه در ◦

سرویسدهی به کاربران.

خودکارسازی بازیابی سرویسهای از کار افتاده یا ناپایدار. **:Self-healing قابلیت** ◦

) برایCPU، RAM( تخصیص و مدیریت بهینه منابع سختافزاری**مدیریت منابع:** ◦

سرویسها.

|  |  |  |
| --- | --- | --- |
| فراهم کردن زیرساخت الزم برای اجرای | **:Microservices پلتفرم استاندارد برای** | ◦ |

معماری میکروسرویس.

**. نمای کلی جریان همگامسازی8**

فرآیند همگامسازی دادهها بین فروشگاه آنالین و سیستم حسابداری محک، ستون فقراتارتباطی این دو سیستم است. این جریان به صورت ناهمگام و با هدف حفظ یکپارچگی دادههایمالی و عملیاتی بین پلتفرمها طراحی شده است.

**جریان اصلی:**

|  |  |
| --- | --- |
| **):Scheduling( زمانبندی** ) داخلی در فروشگاه آنالین، بر اساس یکScheduler( یک سیستم زمانبندی دقیقه، یا هر ساعت) اجرای فرآیند15 برنامه از پیش تعیین شده (مثال ً هر | .1 ◦ |

همگامسازی را آغاز میکند.

هدف از این زمانبندی، اطمینان از بهروز بودن دادهها در هر دو سیستم بدون ◦

های خارجی یا پایگاه دادهها است.API بارگذاری بیش از حد بر روی

|  |  |  |  |
| --- | --- | --- | --- |
| **:Mahak API درخواست داده از** | | | .2 |
| . را فعال میکند | Mahak Sync Worker | ، سرویس Scheduler | ◦ |
| Mahak Sync Worker با استفاده از آخرین وضعیت ثبت شده (مانند | | | ◦ |

دریافتی) در پایگاه داده داخلی خود،Timestamp یاRowVersion آخرین

(از طریق Mahak API درخواستی را بهMahak API Adapter) ارسالمیکند.

|  |  |
| --- | --- |
| این درخواست برای دریافت دادههایی است که از زمان آخرین همگامسازی تغییرکردهاند یا جدیدا ً ایجاد شدهاند (مثال ً محصوالت جدید، سفارشات تکمیل شده، | ◦ |
| تغییرات انبار).  در Timeout وRate Limit در این مرحله، مکانیزمهای مدیریتMahak API ◦ | |

|  |  |  |
| --- | --- | --- |
|  | |  |
| Mahak فعال هستند تا از بروز خطا به دلیل محدودیتهای سیستم | Adapter |

جلوگیری شود.

|  |  |
| --- | --- |
| **دریافت و پردازش اولیه دادهها:** پاسخ را ارسال میکند که شامل مجموعهای از دادهها است. اینMahak API | .3 ◦ |

.) و صفحهبندی شده دریافت شوندBatch( دادهها ممکن است به صورت دستهای

|  |  |  |
| --- | --- | --- |
| Data Transformer دادههای دریافتی را ابتدا در | Mahak Sync Worker | ◦ |

پردازش میکند تا فرمت آنها برای استفاده در سیستم داخلی مناسب شود. اینشامل نگاشت فیلدها، تبدیل انواع داده و اطمینان از صحت دادهها است.

**:)RabbitMQ( ارسال به صف پیام** .4

دادههای پردازش شده در قالب پیامهایی (هر پیام شامل یک یا چند رکورد) به ◦ ارسال میشوند.RabbitMQ صف پیام ارسال به صف پیام، به سیستم اجازه میدهد تا پردازش اصلی را به صورت ◦ وRetry ناهمگام و با قابلیت اطمینان باالتری انجام دهد. همچنین امکان مدیریت را فراهم میکند.DLQ

**:Mahak Sync Worker )Worker Pool( پردازش توسط** .5

چندین نمونه از Mahak Sync Worker به صورت موازی به صف پیام گوش ◦

میدهند و پیامها را دریافت میکنند.

، یک دسته از دادهها را پردازش میکند:Worker هر ◦**:Upsert Engine**" را در پایگاه داده محلیUpdate" " یاInsert" عملیات ▪ فروشگاه آنالین انجام میدهد. در صورت وجود رکورد مشابه، از منطق برای اطمینان از درستی دادهها استفاده میکند.Conflict Resolution ▪**:Logical Delete Handler** نشاندهنده حذف منطقیMahak اگر دادههای باشند، این مکانیزم در پایگاه داده داخلی نیز رکورد مربوطه را غیرفعال میکند.

|  |  |
| --- | --- |
| **مدیریت خطا و تالشهای مجدد:** در پایگاه داده) خطاییUpsert اگر در هنگام پردازش یک دسته از دادهها (مانند . فعال میشود رخ دهد، Retry & Error Manager ). برای این کار، پیام بهRetry( سعی میکند عملیات را مجددا ً انجام دهدWorker | .6 ◦  ◦ |

. بازگردانده میشودRabbitMQ درRetry صف Dead-Letter اگر پس از چندین تالش، عملیات همچنان ناموفق باشد، پیام به ◦ منتقل میشود. این پیامها توسط تیم عملیات یا توسعهدهندگانQueue )DLQ( .برای بررسی علت اصلی خطا جمعآوری میشوند

**ثبت وضعیت و الگینگ:** .7

تمام عملیات همگامسازی، موفقیتها و شکستها در الگهای سیستم ثبت ◦ میشوند.

پردازش شده برای هر نوع داده، در پایگاهTimestamp یاRowVersion آخرین ◦داده داخلی ذخیره میشود تا در چرخه همگامسازی بعدی مورد استفاده قرار گیرد.

این فرآیند ناهمگام تضمین میکند که سیستم فروشگاه آنالین همواره با دادههای مالی و هماهنگ باشد، ضمن اینکه از قابلیت اطمینان باال و مدیریتMahak عملیاتی به روز از سیستمخطای مؤثر برخوردار است.