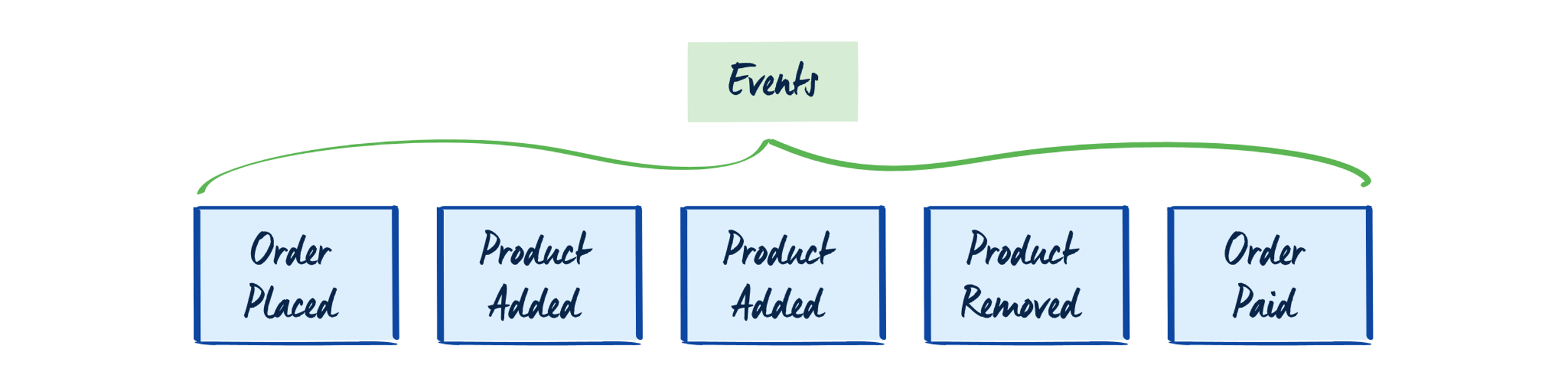
**Event Sourcing**

در event sourcing از مفهوم eventها برای مدل کردن و نگهداری داده ها استفاده می کنیم در واقع در event sourcing ، دستورات ( command ) های لاجیک در دیتابیس ذخیره می شوند به گونه ای که ثبت هر تغییر در دامنه معادل می شود با یک رویداد .

**نکته :** رویداد ها تغییر ناپذیر هستند ( remove و update ) نداریم .

موضوعی که در event sourcing وجود دارد، موضوع last state یا وضعیت آخر است. از طریق اجرای event هایی که در سیستم و بر روی یک رکورد اتفاق افتاده، می توانیم به وضعیت فعلی آن رکورد برسیم . در این رویکرد وضعیت دیتا بروزرسانی نمی شود با هر تغییر وضعیت ، وضغیت جدید append می شود .

فرض کنید یک سیستم ثبت سفارش داریم و رکودری در دیتابیس برای این مورد ذخیره شده است . در حالت معمولی سیستم به این گونه طراحی می شود که رکورد دیتابیس بر حسب وضعیت سفارش بروز رسانی ( update ) می شود و یا اگر سفارش حذف شود شاید به صورت فیزیکی رکورد حذف شود . اما در مدل سازی به شیوه event sourcing ما دستوراتی که روی سفارش رخ داده است را ثبت می کنیم .



1 – سفارش اول ثبت شد

2 – سفارش اول ویرایش شد

3 - پشتیبان اول وضعیت سفارش را مشاهده شده قرار داد

4 - پشتیبان دوم سفارش را به provider ارسال کرد .

5 - Provider سفارش را دریافت کرد .

. . .

**نکته :** Event ها با هر جزئیاتی می توانند ذخیره شوند .

event sourcing را به عنوان یک system architecture در نظر نمی گیریم . استفاده از event sourcing برای کل سامانه یک over engineering تلقی می شود . ( در واقع یک الگوی طراحی نرم افزار است ) دراین الگو نرم افزار ما به دو بخش Command و Query تقسیم می شود. هر آن چیزی که باعث تغییر در وضعیت سیستم ما شود، یک command محسوب می شود. یک command می تواند از سمت کاربر، سیستم بیرونی و یا در ازای رخ دادن یک event ارسال شود.

**قانون اولی که برای commandها وجود دارد این است که باید به صورت امری نوشته شوند.**

**قانون دوم این هست که برای ایجاد command ها باید بر روی Business تمرکز کنیم.**

**زمانی که که event sourcing وارد پروژه می شود، اولین موردی که در تحلیل سیستم حذف می شود، Data Structure و چگونگی نگهداری داده ها است.**

**بیشتر به این مساله فکر می کنیم که چه eventهایی در سیستم داریم. روش هایی برای موضوع شناخت domainها مثل EventStorming داریم که از مفهوم event modeling استفاده می کنند.**

**Event Sourcing کجا ها خوبه ؟**

* زمانی که تاریخچه دیتا برامون خیلی مهمه ( همه رویداد ها و اتفاقات ثبت می شود با این رویکرد می توانیم گزارش بگیریم که در دوسال پیش ساعت فلان دقیقا چه اتفاقی افتاد . )
* زمانی که performance خیلی برامون مهمه ... **سوالی که پیش میاد خوندن گزارش ها ( reply کردن event ها ) چطور ممکنه عملکرد خوبی داشته باشه ؟**

event sourcing در حوزه performance به دو موضوع Write و Read می پردازد نوشتن داده ها در event sourcing به شکل append می باشد و رکوردی update نمی شود و سرعت بالایی دارد. اما عملیات خواندن در زمینه performance بیشتر مورد توجه قرار می گیرد به همین دلیل موضوع CQRS را اکثر اوقات در کنار event sourcing می بینیم

CQRS الگویی هست که تمرکز آن روی جداسازی مکانیسم خواندن از نوشتن هست ( Command و Query )

یکی از علت هایی که ما از الگوی CQRS بهره می بریم مانور روی عملکرد سیستم با تمرکز رو IO هست .

منابع :

<https://git.ir/udemy-net-microservices-cqrs-event-sourcing-with-kafka>

<https://www.eventstore.com/event-sourcing#Benefits-of-Event-Sourcing>