An event-driven architecture uses events to trigger and communicate between decoupled services and is common in modern applications built with microservices. An event is a change in state, or an update, like an item being placed in a shopping cart on an e-commerce website. Events can either carry the state (the item purchased, its price, and a delivery address) or events can be identifiers (a notification that an order was shipped).

معماري رويداد محور از رويداد ها براي راه اندازي و ايجاد ارتباط بين سرويس هاي جدا استفاده ميكند و در اپلیکیشن های مدرن ساخته شده با میکرو سرویس ها بسیار رایج هستند . یک رویداد یک تغییر یا بروز رسانی وضعیت است ، مانند یک کالا که درسبد خرید یک وب سایت تجارت الکترونیک قرار دارد . رویدادها می‌توانند دارای وضعیت باشند (کالای خریداری‌شده، قیمت آن و آدرس تحویل) یا رویدادها می‌توانند شناسه (اعلان ارسال سفارش) باشند.

Event-driven architectures have three key components: event producers, event routers, and event consumers. A producer publishes an event to the router, which filters and pushes the events to consumers. Producer services and consumer services are decoupled, which allows them to be scaled, updated, and deployed independently

معماری رویداد محور سه بخش کلیدی دارد : رویداد تولیدکنندگان ، رویداد روترها ، و رویداد مصرف کنندگان . یک تولید کننده رویدادی را به سمت روتر نشر می دهد که آن رویداد را فیلتر کرده و به سمت مصرف کننده می فرستد . سرویس های تولید کننده و سرویس های مصرف کننده از هم جدا هستند که به آن ها اجازه می‌دهد به صورت مستقل مقیاس گذاری ، بروز رسانی و به کار گرفته شوند .

Benefits of an event-driven architecture

مزایای معماری رویداد محور

### Scale and fail independently

### 1-به طور مستقل مقایسه و شکست بخورند

### By decoupling your services, they are only aware of the event router, not each other. This means that your services are interoperable, but if one service has a failure, the rest will keep running. The event router acts as an elastic buffer that will accommodate surges in workloads.

### به دلیل جدا بودن سرویس های شما، آنها فقط از رویداد های روتر آگاه هستند .نه از رویداد های هم این بدان معناست که سرویس‌های شما قابلیت همکاری دارند، اما اگر یکی از سرویس‌ها با مشکل مواجه شود، بقیه به کار خود ادامه می‌دهند. رویداد روتر به عنوان یک بافر قابل ارتجاع عمل میکند که با افزایش بارکاری جلوگیری می‌کند.

### Develop with agility

### 2-توسعه چابک

### You no longer need to write custom code to poll, filter, and route events; the event router will automatically filter and push events to consumers. The router also removes the need for heavy coordination between producer and consumer services, speeding up your development process.

### دیگر نیازی به نوشتن کد سفارشی برای نظر سنجی ، فیلتر و مسیر یابی رویداد ها ندارید . رویداد روتر به طور اتوماتیک فیلتر و ارسال میکند رویداد ها رو به مصرف کنندگان . روتر همچنین نیاز به هماهنگی شدید بین خدمات تولید کننده و مصرف کننده را از بین می برد و روند توسعه شما را سرعت می بخشد.

### 3- Audit with ease

### 3-تسهیل در حسابرسی

### An event router acts as a centralized location to audit your application and define policies. These policies can restrict who can publish and subscribe to a router and control which users and resources have permission to access your data. You can also encrypt your events both in transit and at rest.

### روتر رویداد به عنوان یک مکان متمرکز برای بررسی برنامه شما و تعریف خط مشی ها عمل می کند.این خط مشی ها می توانند انحصار ایجاد کنند که چه کسی . همچنین می توانید رویدادهای خود را هم در حالت حمل و نقل و هم در حالت استراحت رمزگذاری کنید.

### 4-cut costs

### 4-کاهش هزینه ها

### Event-driven architectures are push-based, so everything happens on-demand as the event presents itself in the router. This way, you’re not paying for continuous polling to check for an event. This means less network bandwidth consumption, less CPU utilization, less idle fleet capacity, and less SSL/TLS handshakes.

### معماری رویداد محور مبتنی بر فشار است ، بنابراین همه چیز بر اساس تقاضا اتفاق می‌افتد زیرا رویداد‌ها خود را در روتر ها نمایش می‌دهد . این به معنای کاهش مصرف پهنای باند شبکه ، استفاده کمتر CPU ، ظرفیت ناوگانی کمتر .

### نحوه کارکرد معماری رویداد محور

### نحوه کار این معماری را با ذکر یک مثال توضیح می دهیم

### C:\Users\user\Desktop\للل.jpg.png