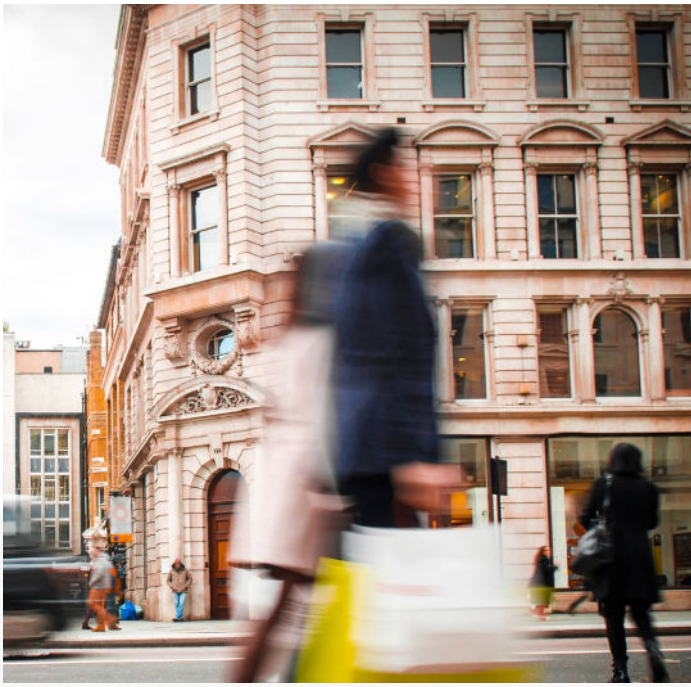
Powering a leading payment platform with event-driven architecture

Using events to enhance customer experience, automate fraud detection, and seamlessly handle complex payment journeys for retail merchants.

EQUAL EXPERTS



As the Middle East's leading provider, with annual turnover of around $350 million, our client offers a wide range of payment services.

From card issuing and merchant acquisition to payment processing and provision of in-store, online and mobile payment solutions—across multiple international markets—they manage sophisticated end-to-end payment journeys. And, they do it for all types of clients; from large enterprise and government organisations to small-medium retail businesses and micro-merchants.

The lynchpin for many of these services is an event-driven **payments platform that Equal Experts helped to build** from the ground up. The platform provides merchants with a single point of contact to initiate, process, capture and track a variety of payments spanning credit card, bank transaction, PayPal, China UnionPay, M-Pesa and more. Learn how the payment platform uses event driven architecture to provide secure, fast and reliable payments, protect against fraudulent payments, and offer a seamless experience for both merchants and customers alike.

This case study will help you to understand:

The benefits of adopting event driven architectures and asynchronous processing in retail payment gateways

How events can be used for real-time risk management and fraud detection in transactions

How event-driven architecture and microservices can prevent cascading service degradation during peak usage

Table of contents.

ABOUT THE EVENT-DRIVEN PAYMENT PLATFORM.

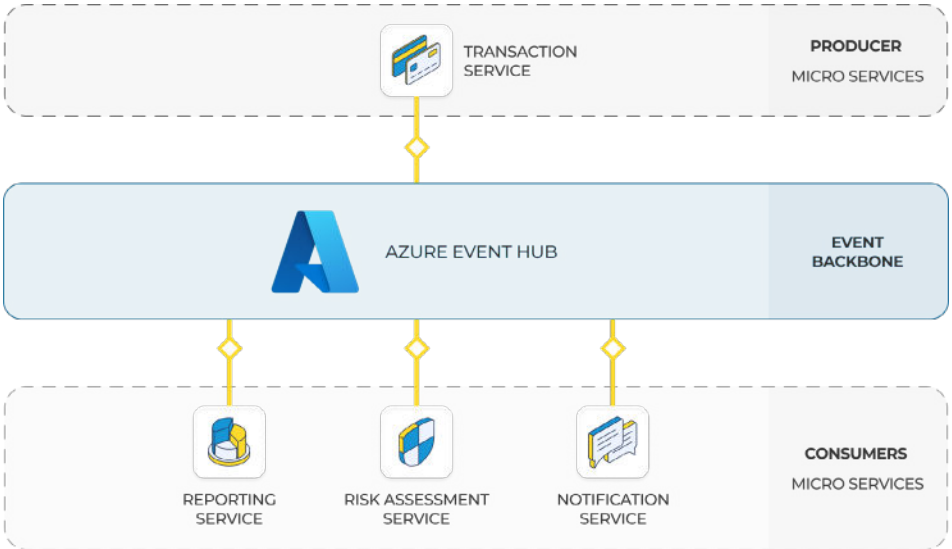
The digital payment platform is built using microservices and leverages event-driven patterns for real-time automation and insights into payment processing.

While the payment platform consists of more than 25 microservices in totality, this case study will focus on a subset of core services to highlight the value of event-driven processes that occur within the platform.

These include the:

- **Transaction service:** which produces and consumes events to facilitate customer orders and accept various payments.
- **Reporting service:** primarily used by merchants in a self-service portal, producing and consuming events for real-time display of transaction and payment information across a variety of channels.
- **Risk assessment service:** which produces and consumes events while determining the potential risk associated with orders and payments.
- **Notification service:** which produces and consumes events while automating communications to highlight the status, or changes in status, of orders and payments.

Whenever an order or payment is processed in the transaction service, it sends out a variety of events (which can be thought of as 'triggers' containing data for subsequent, automated processes) to the event backbone, Azure Event Hub. The event backbone, in turn, fires events to services which are configured to 'listen' to them. These listening services are configured to consume specific events related to their core business function; it's a 'pull'-style configuration, rather than the 'pushing' of data packets.



For example, if the transaction service processes 50 payments from one personal account number (PAN) in 10 minutes, an event is produced to notify the risk assessment service to take action to prevent the fraud (see section 03. 'Using events for real-time fraud detection'). The reporting service may 'listen' for the same event, and provide an update that a certain payment or account is potentially compromised.

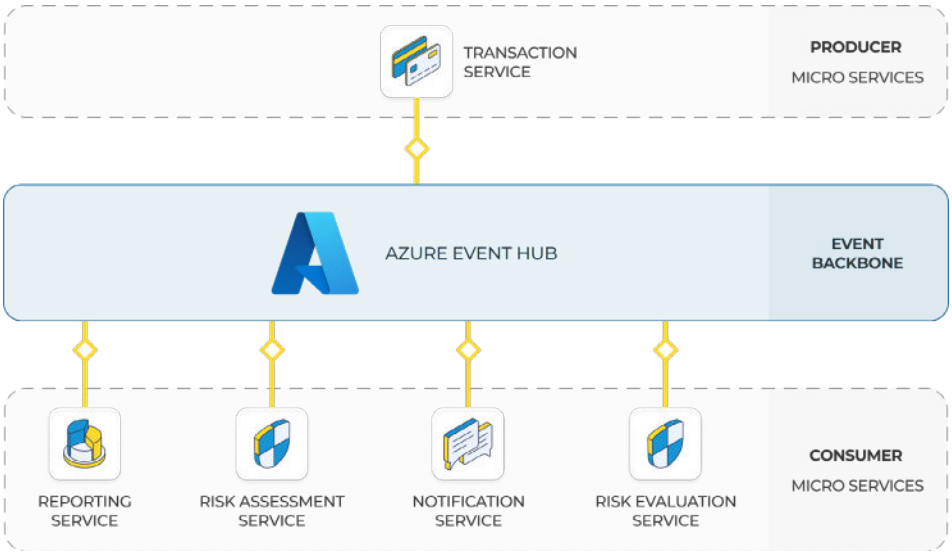
Examples of event-driven processes include:

- All payment processes in the transaction service. This means the complex workflows associated with payment submission, authorization, transitioning payments into escrow accounts, merchant capture, and credit reversal callback are all seamlessly automated and tracked in an immutable log.
- Transactional emails, SMS messages and merchant notifications are fully event-driven, ensuring both customer and merchant have real-time updates into payment status and processing.
- Advanced risk rules are evaluated using events to detect and prevent potential fraud in real-time.
- Events trigger updates to card management systems for reversal of any voided transactions.
- Recurring and subscription payments are automatically triggered using events.
- On successful completion of payment, the transaction service generates a series of events for reporting purposes. This includes everything from creating an immutable trail of payments for merchant recordkeeping, to generating reports and adding to computed dashboard data.

Events provide the power to quickly extend functionality

In addition to automating processes, improving real-time insight, and enhancing the resilience of systems, event driven architectures are infinitely—and easily—extensible.

For example, imagine a scenario where the above process requires the introduction of a gateway risk evaluation. Events are already being generated and fed into the event backbone; another event consumer (the risk evaluation service) can be configured to 'listen for' and consume that same stream of events for an entirely independent business requirement.



THE BENEFITS OF ASYNCHRONOUS PROCESSING IN PAYMENT VALIDATION.

Picture the scenario: you run an online retail store, selling high value homewares like televisions, watches and cosmetics.

A customer visits your eCommerce store and orders a TV. By submitting their card details, a series of events are immediately triggered to start validating the payment.

On payment capture, a series of events are triggered to automate notifications to both you and your customer.

While the notifications are being prepared and sent—which only takes a couple of seconds—all of your reporting is updated in real-time with detailed information highlighting the nature of the payment. You receive a notification, and you can view the information in your self-service portal.

Critically, the varying steps in these payment validation processes can occur asynchronously in event-driven architectures.

Structuring the payment process as a series of decoupled, independent mircoservices creates two crucial benefits;

1. Speed, and
2. Reliability of processing.

Firstly, the various, complex steps involved in validating payments can occur in a non-linear way. This makes things faster overall, especially when compared with traditional sequential or synchronous processing.

By their nature, event-driven architectures are 'fire and forget'. In other words, there's no compulsory requirement for a microservice to wait for a response before initiating the next step in a sequence.

For example, the transaction service might receive a payment, fire an event to the reporting service to create a log of that payment being received, and begin the process of validating that payment. This process can occur entirely independent of the reporting service acknowledging to the transaction service that a log has been created and saved in the reporting database.

These efficiencies may seem small in isolation, but at scale — when large enterprises or national retailers are **processing tens or hundreds of thousands of payments per hour** — the impact for processing and customer experience can be huge.

For example, the transaction service might receive a payment, fire an event to the reporting service to create a log of that payment being received, and begin the process of validating that payment. This process can occur entirely independent of the reporting service acknowledging to the transaction service that a log has been created and saved in the reporting database.

In addition to speed, asynchronous processing and event driven architectures break the chains of interdependencies commonly seen in synchronous structures like REST or gRPC. This drastically reduces the risk of service degradation cascading across multiple microservices.

USING EVENTS FOR REAL-TIME FRAUD DETECTION.

In addition to speed and reliability, event-driven processing allows the payment platform to screen for fraudulent transactions much faster. And even pre-empt and block those transactions before they can occur.

In most card-based frauds, stolen cards are used to make multiple, rapid transactions through different channels in quick succession. For example, a person might quickly attempt a transaction on ten different devices in five minutes using a stolen card.

Using events, the platform tracks attempted transactions against specific primary account numbers (PANs) and triggers an event for each transaction. Those events are recorded against a sliding window of time. If a specific PAN exceeds the acceptable transaction threshold within a certain time window—for example, 25 transaction attempts within a 30-minute period—another event is triggered and consumed by a fraud-risk engine for analysis. The risk engine, in turn, flags potentially fraudulent transactions in real time, and the proposed payment(s) will be prevented from progressing towards validation.

Real-time views of merchant-customer risk rules

Events are also useful for version-control style functionality and sustained visibility of merchant-customer activity.

Whenever sensitive risk rules are modified by merchants using the self-service platform—for example, they decide to block certain countries or increase their online transaction limits—the configuration service creates an event which is consumed by the notification service. The notification service triggers an event to the fraud management team to ensure full, real time visibility of any changes to the risk rules, and a more detailed, to-the-minute profile of each customers' unique context.

IMPROVING UX THROUGH EVENT-BASED INTERACTION.

While event driven architectures are enormously beneficial for internal processing and operations, they offer crucial benefits in general end user experience too.

Merchants get their money faster than ever before

Payment platforms are incredibly high throughput applications. Additionally—as a leading provider of payment solutions—our client integrates with a huge number of merchants to serve customers around the world.

By decoupling microservices and using asynchronous processing (see **section 02. 'The benefits of asynchronous processing in payment gateways'**), we can provide high speed transaction rates which are simply not possible using synchronous blocking or sequential processing. The result? Merchants get revenue delivered to their acquisition accounts faster than before, and customers get their orders processed and confirmed at lightning pace.

An event-driven self-service portal makes life easy for merchants

Merchants can access a self-service portal which offers:

- Full visibility of every transaction's status across all channels; from initial authorisation to capture. An event-driven architecture ensures merchants can access and understand all payments in play, as they're processed in real-time.
- Configurable gateway risk rules—from blocking certain countries to increasing online transaction limits—which use events to predict and protect against fraudulent transactions.
- Close-to-real time data visuals providing detailed reports, chargebacks, payments histories, and other information.

ABOUT THE TECHNOLOGY STACK.

By their nature, event-driven architectures are inherently extensible and provide a range of benefits for integration.

For example, events can mitigate instances of 'domain bleeding' by reconfiguring data in whichever way any third-party interfaces might require. This is particularly useful in a payment platform, **digital banking platform**, or any other type of complex system that requires multiple integrations.

Additionally, the payment platform's microservice applications use a Hexagon architecture pattern that follows the domain-driven design philosophy. This provides the flexibility to change or update the technology stack without modifying domains or business rule implementations.

Tools

Want to know more?

Are you interested in this project?
Or do you have one just like it?
Get in touch. We'd love to tell you more about it.