

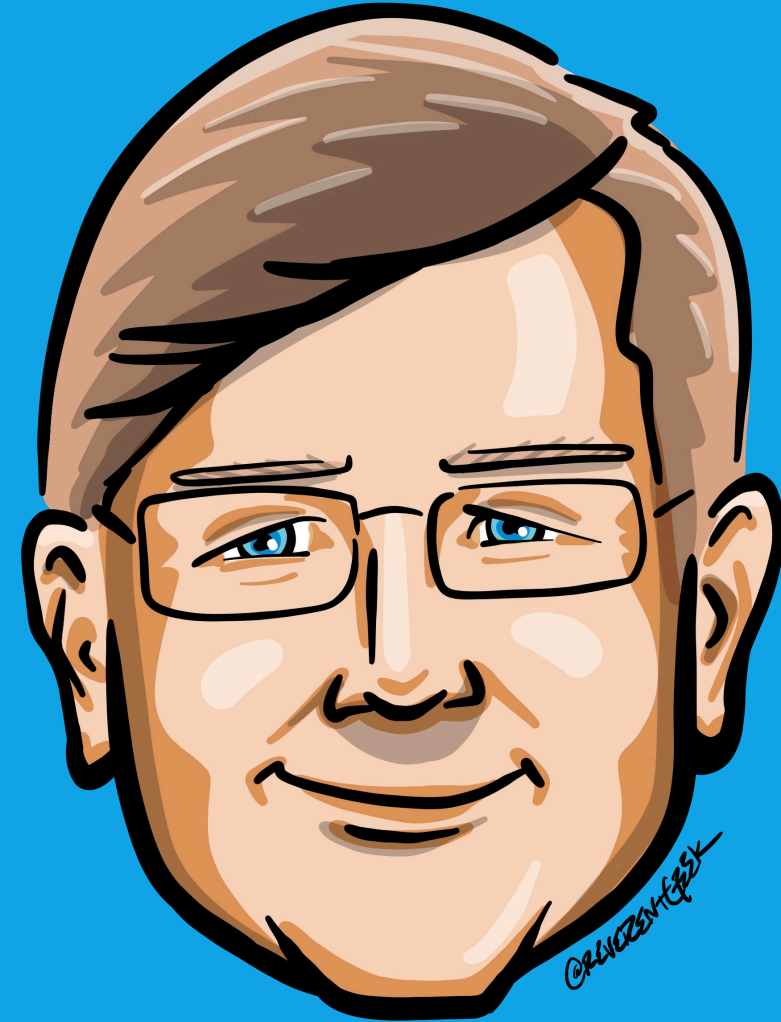


BUILDING EVENT-DRIVEN MICROSERVICES

**PRAIRIE
DEV CON**

Who is Chad Green

Director of Architecture
Louisville, KY



Preamble

Building Event-Driven Microservices

Monolith

Enterprise Architecture

UI

Order Processing

Payment Processing

Inventory Management

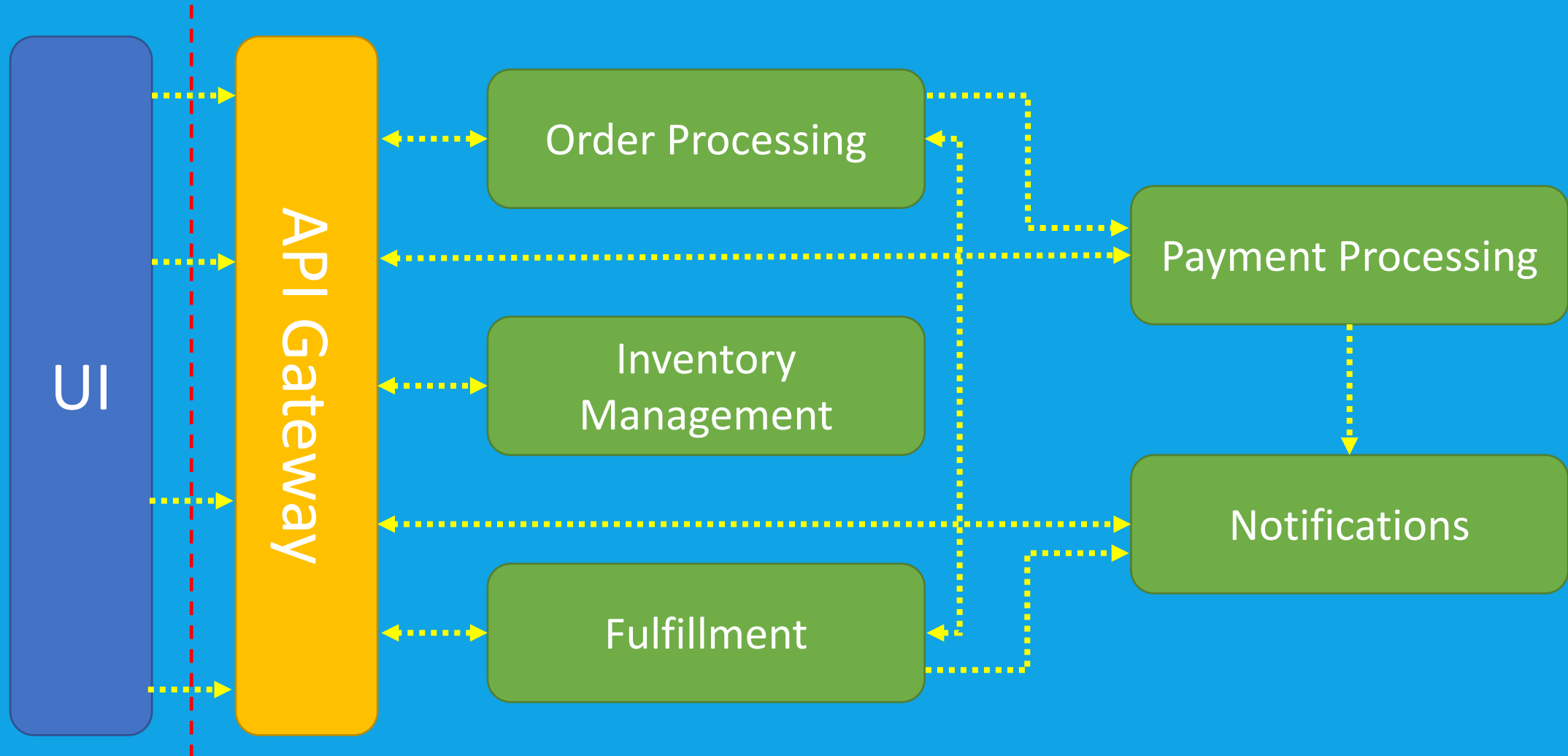
Notification

Fulfillment

Database

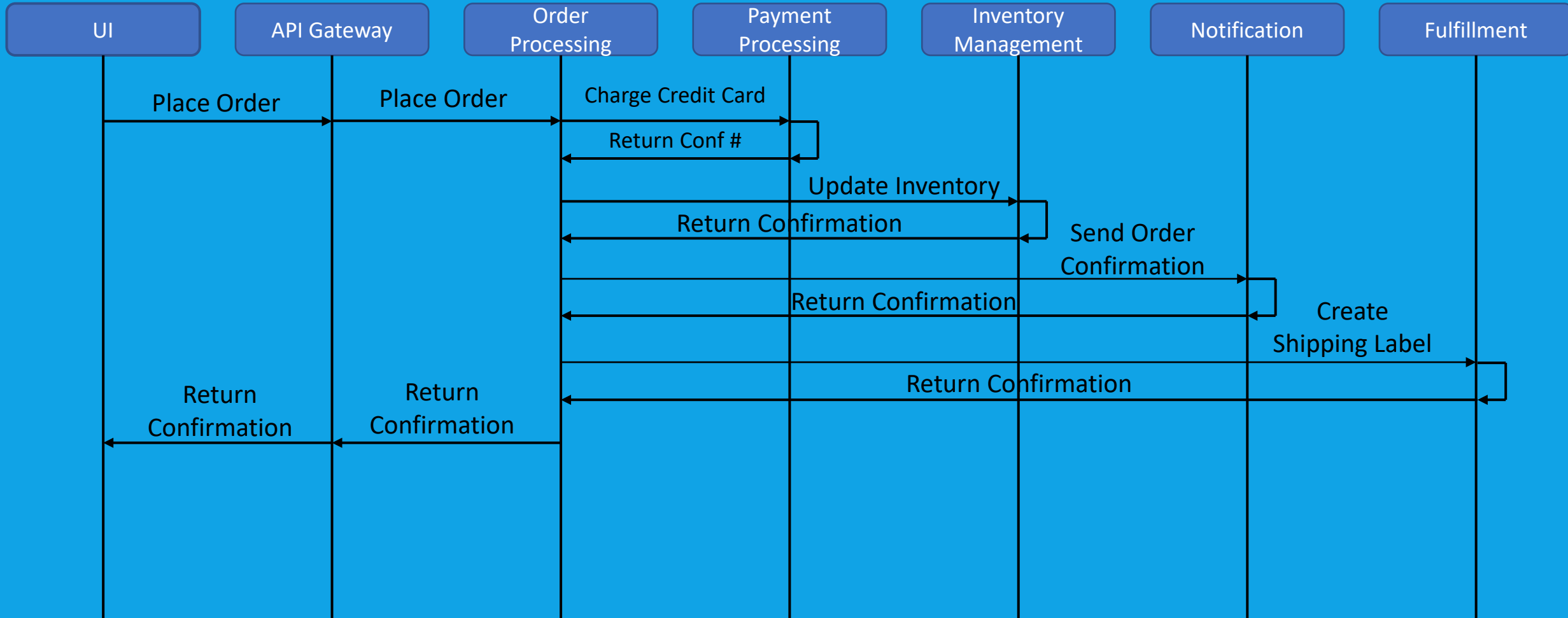
Microservices

Enterprise Architecture



Process Flow

Microservices



Event-Driven Architecture

Building Event-Driven Microservices

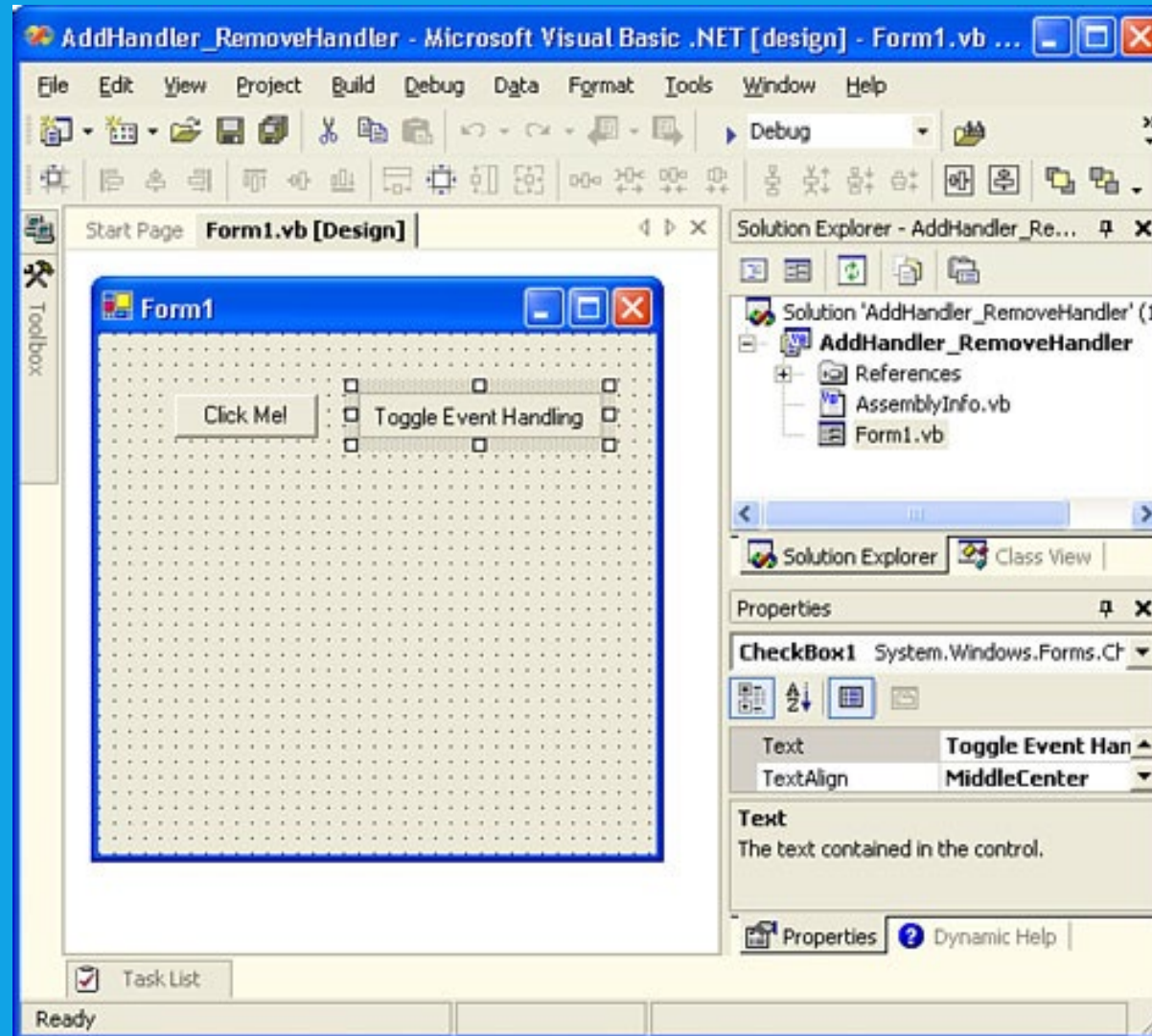
Event-Driven Architecture



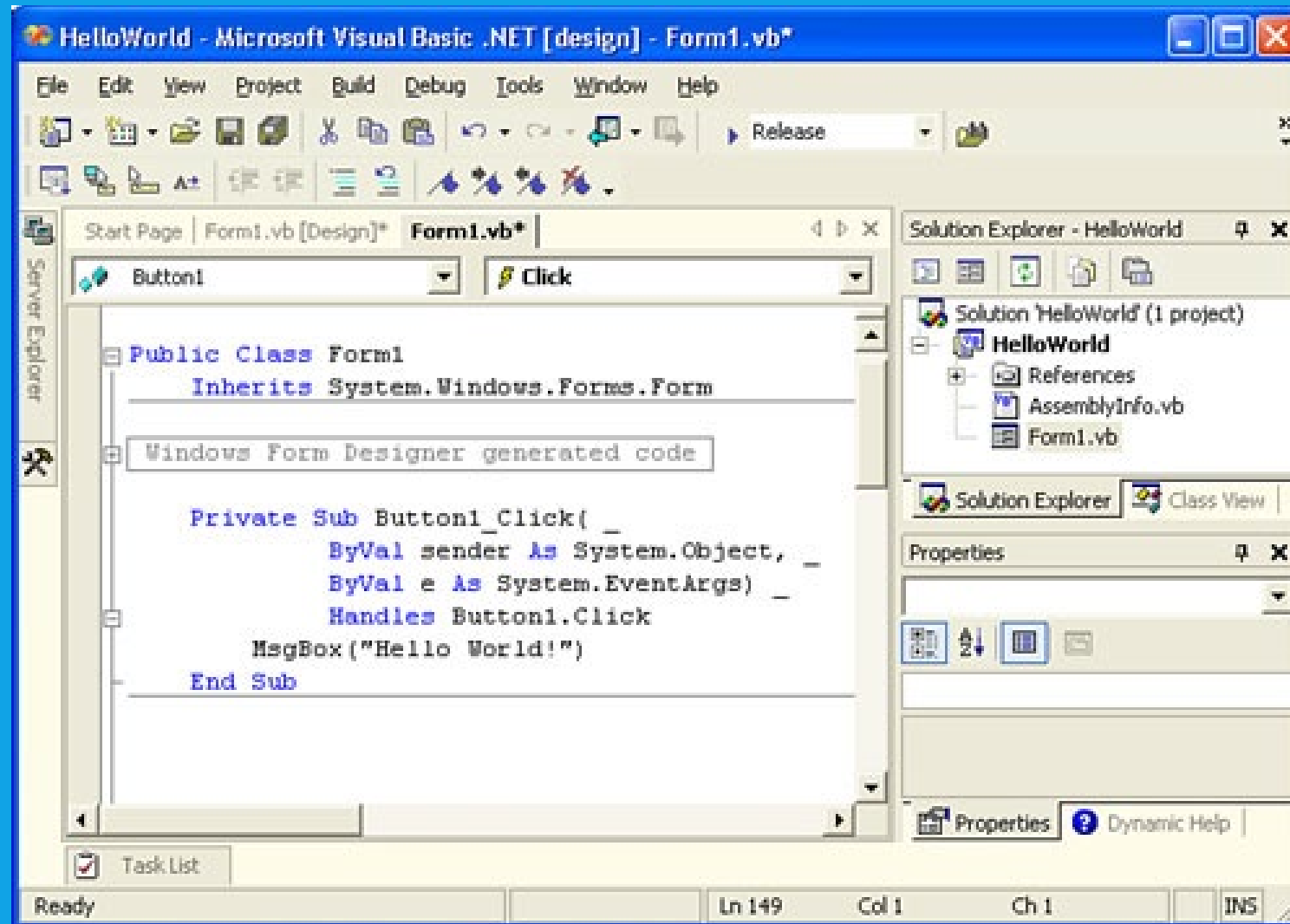
A software architecture pattern promoting the production, detection, consumption of, and reaction to **events**.

- Wikipedia -

Event-Driven Architecture



Event-Driven Architecture



Event-Driven Architecture



Event-driven architecture (EDA) is a design paradigm in which a software component executes in response to receiving one or more event notifications.

EDA is more loosely coupled than client/server paradigm because the component that sends the notification doesn't know the identity of the receiving components at the time of compiling

- Garner -

Event-Driven Architecture

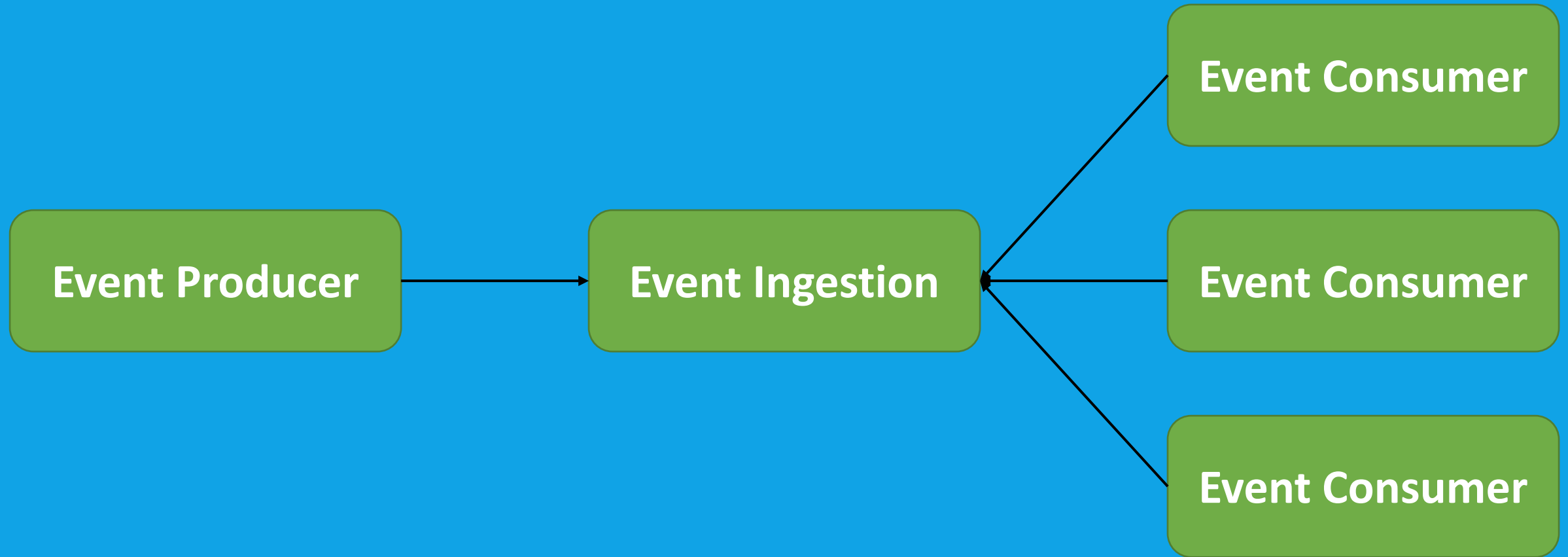


Event-driven architecture (EDA) is a design paradigm in which a software component executes in response to receiving one or more event notifications.

EDA is more loosely coupled than client/server paradigm because the **component that sends the notification doesn't know the identity of the receiving components** at the time of compiling

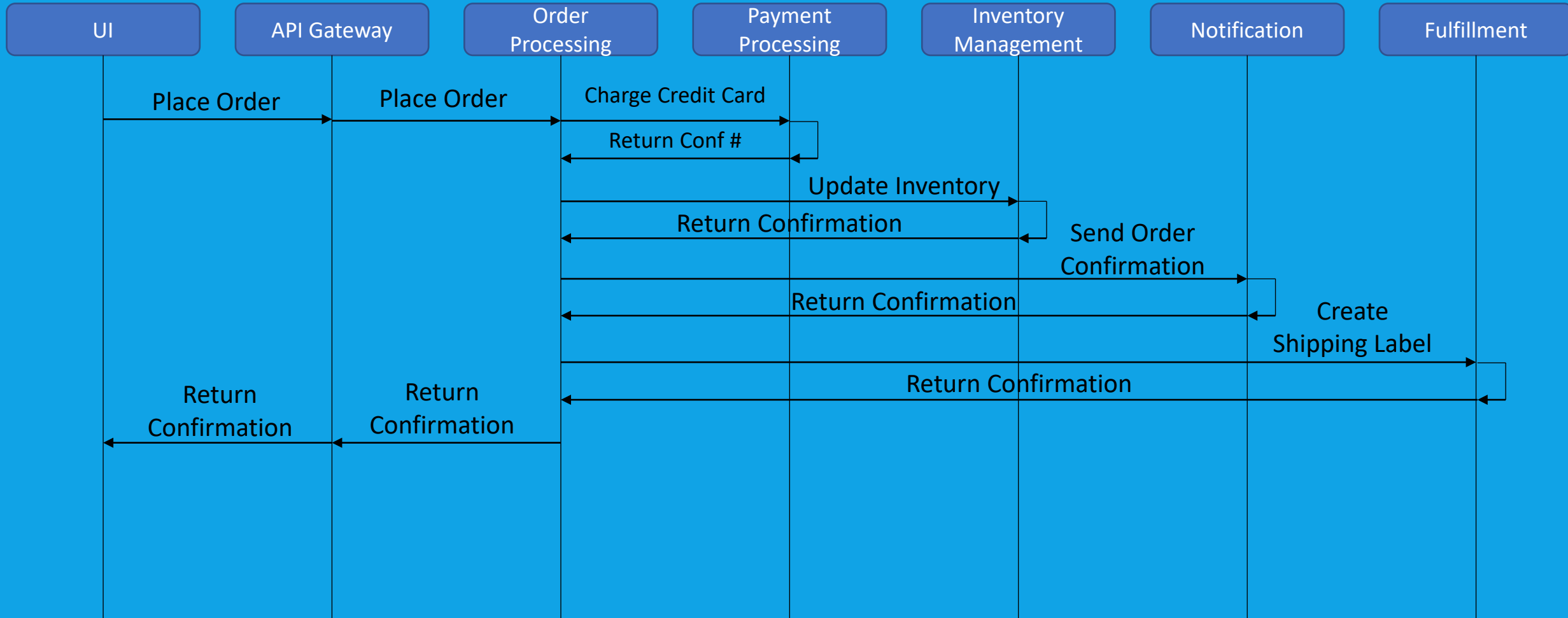
- Garner -

Event-Driven Architecture



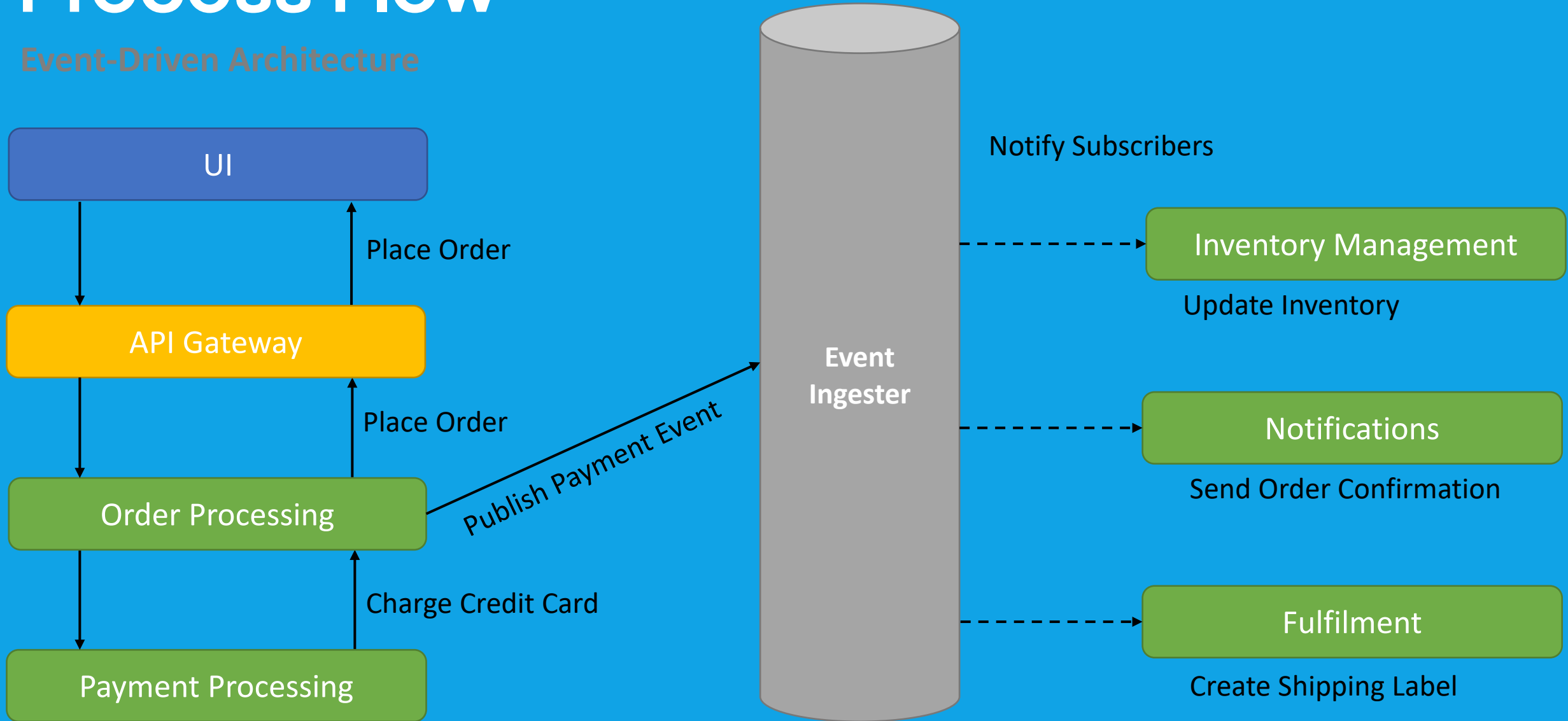
Process Flow

Microservices



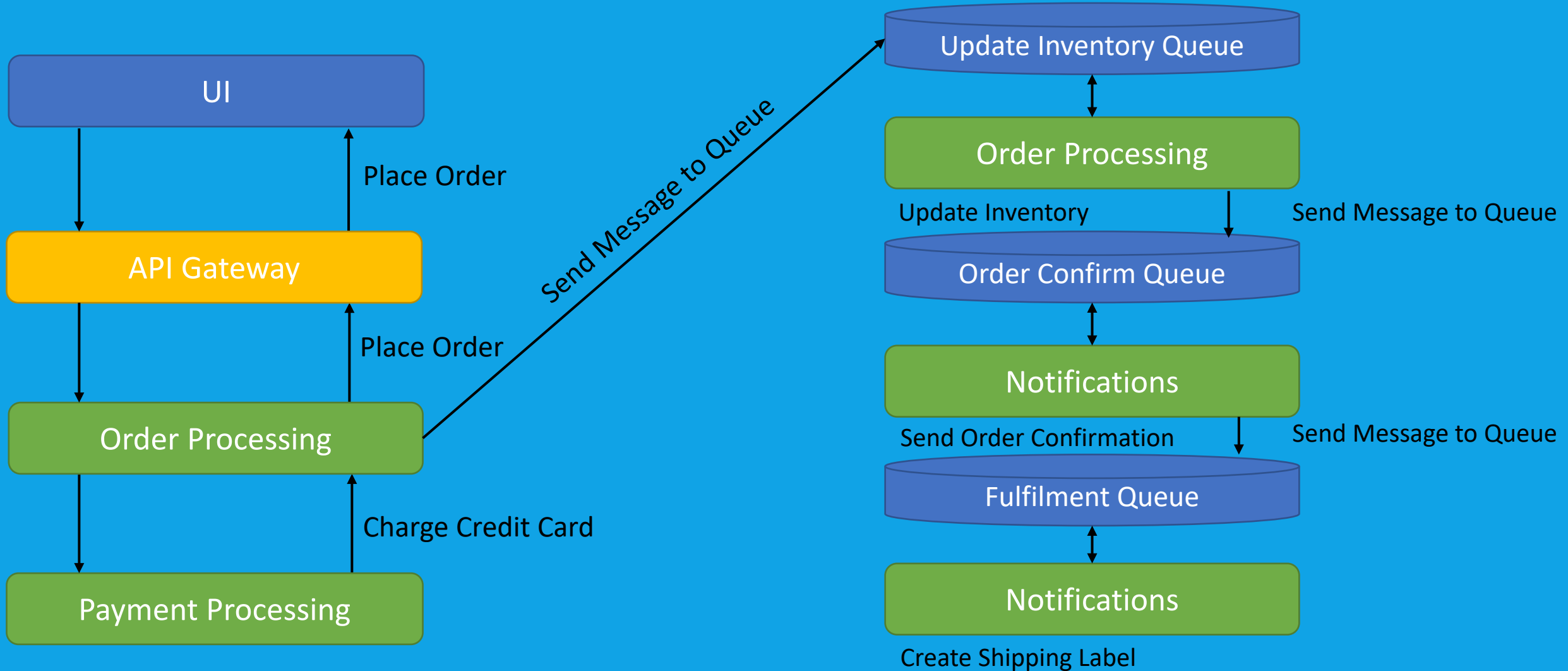
Process Flow

Event-Driven Architecture



Not Queue Based Processing

Event-Driven Architecture

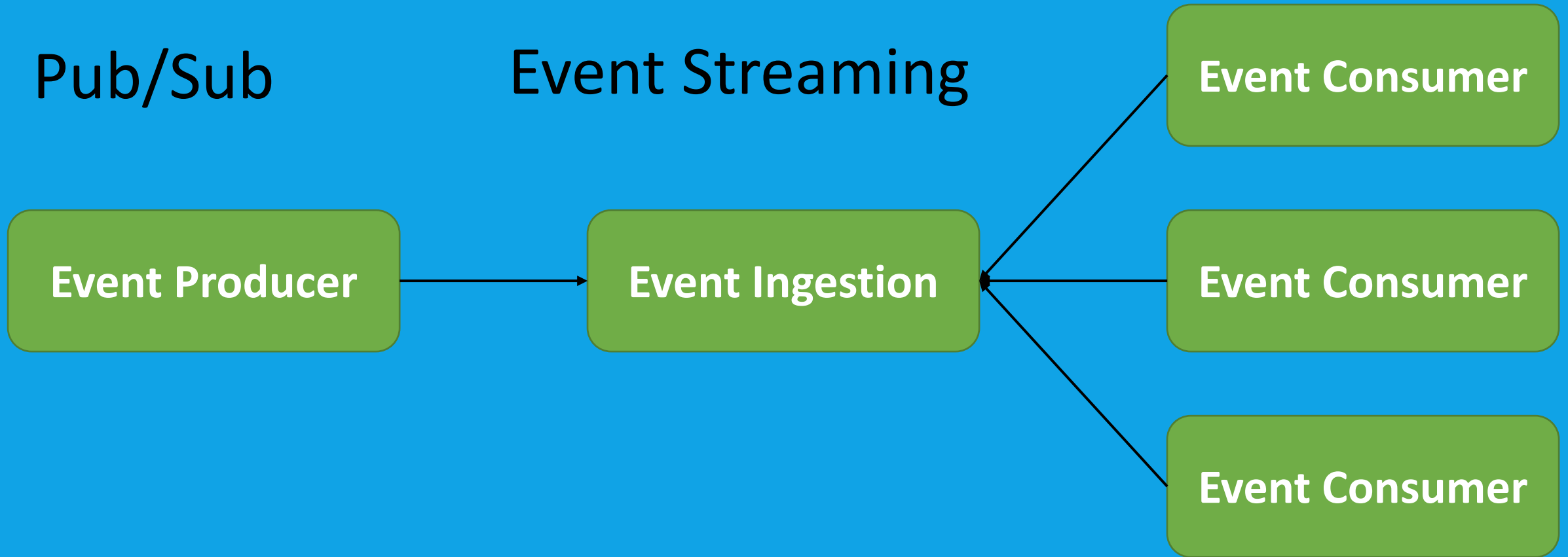


Event Consumption Models

Event-Driven Architecture

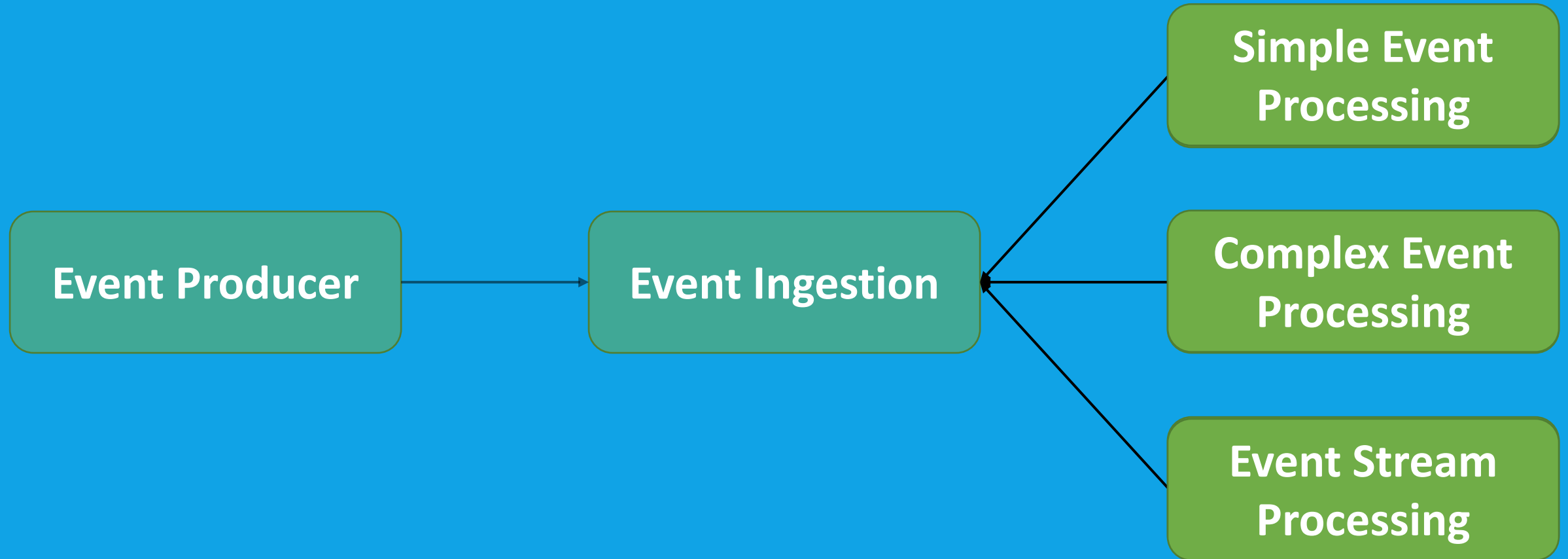
Pub/Sub

Event Streaming



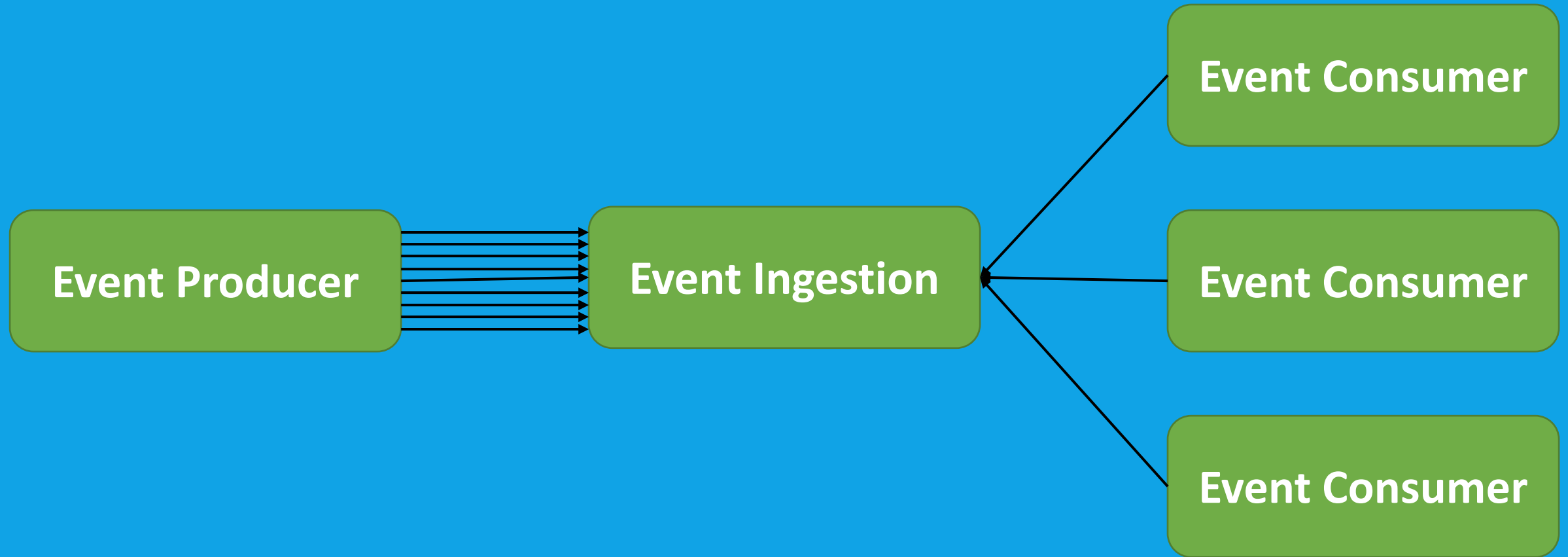
Consumer Processing Variations

Event-Driven Architecture



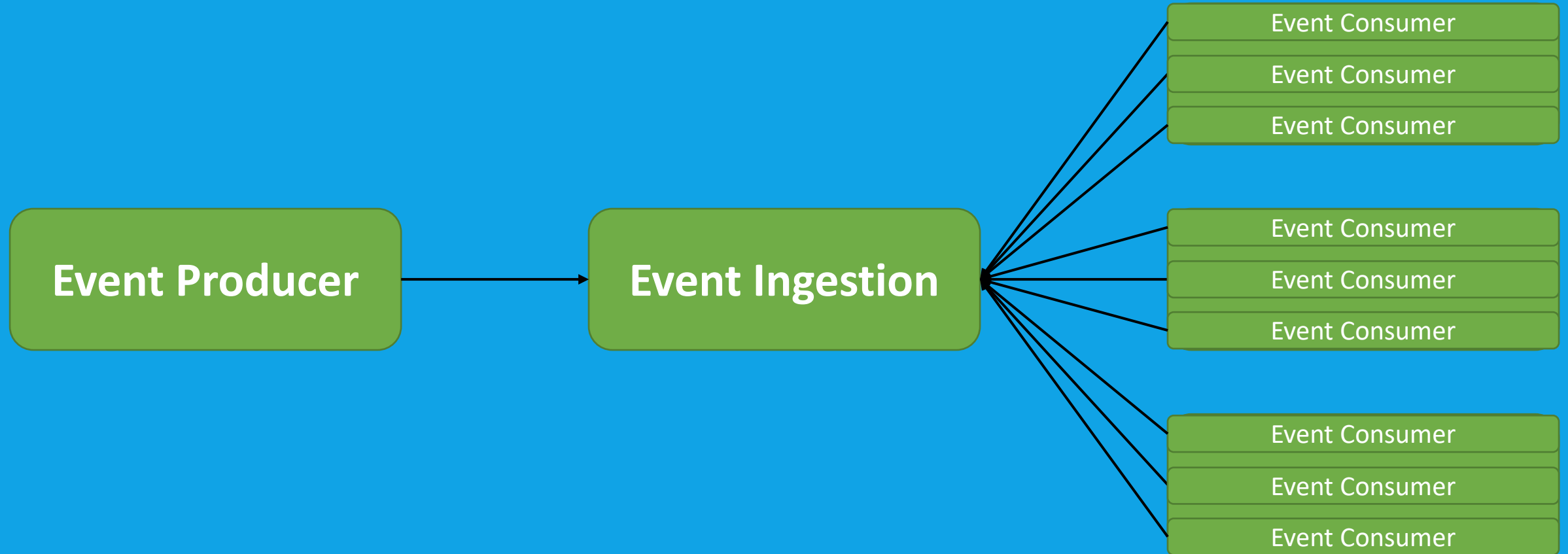
External Event Sources

Event-Driven Architecture



Multiple Consumer Instances

Event-Driven Architecture



When to use this architecture

Event-Driven Architecture

Multiple Subsystems

Real-Time Processing

**Complex Event
Processing**

**High Volume/Velocity
Data**

Benefits

Event-Driven Architecture

Decoupling



Encapsulation



Responsive



Scalable/Distributed



Independence



Drawbacks

Event-Driven Architecture

Steep Learning Curve



Complexity



Loss of Transactionality



Lineage



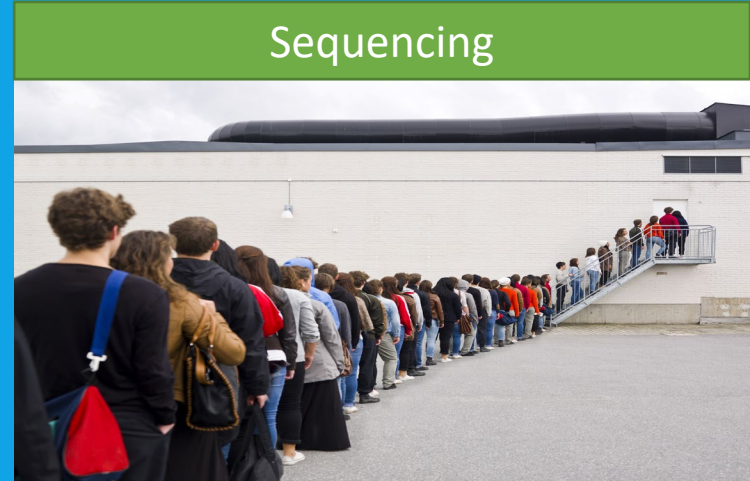
Limitations

Event-Driven Architecture

Guaranteed Delivery



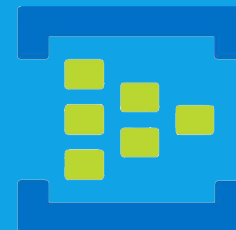
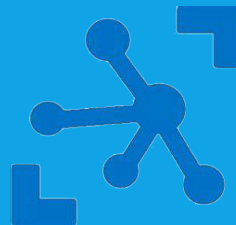
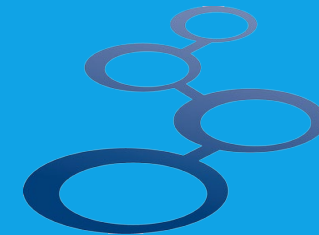
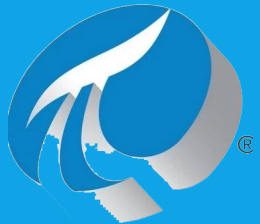
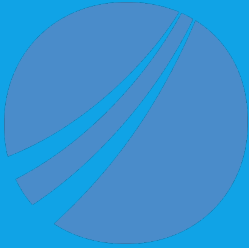
Sequencing



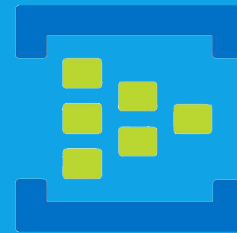
Implementation Options

Building Event-Driven Microservices

Implementation Options

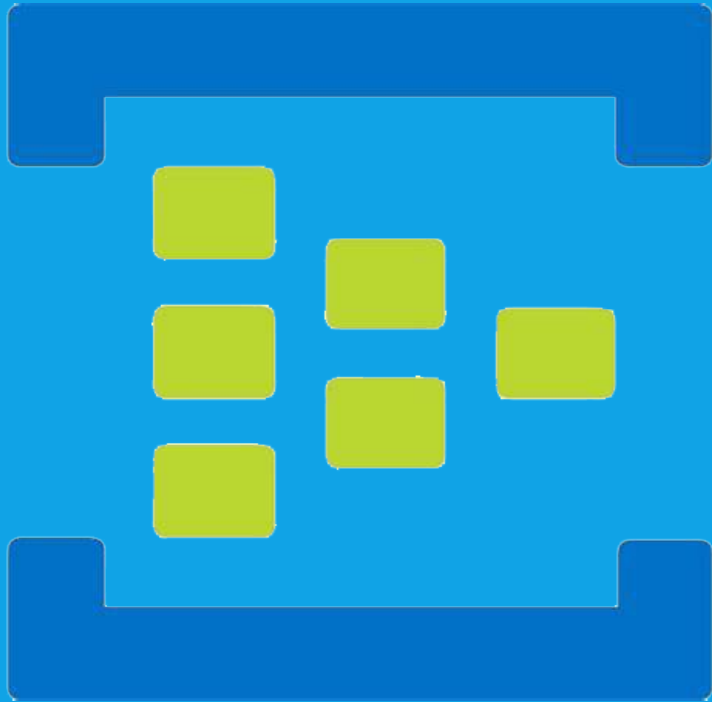


Implementation Options



Azure Event Hubs

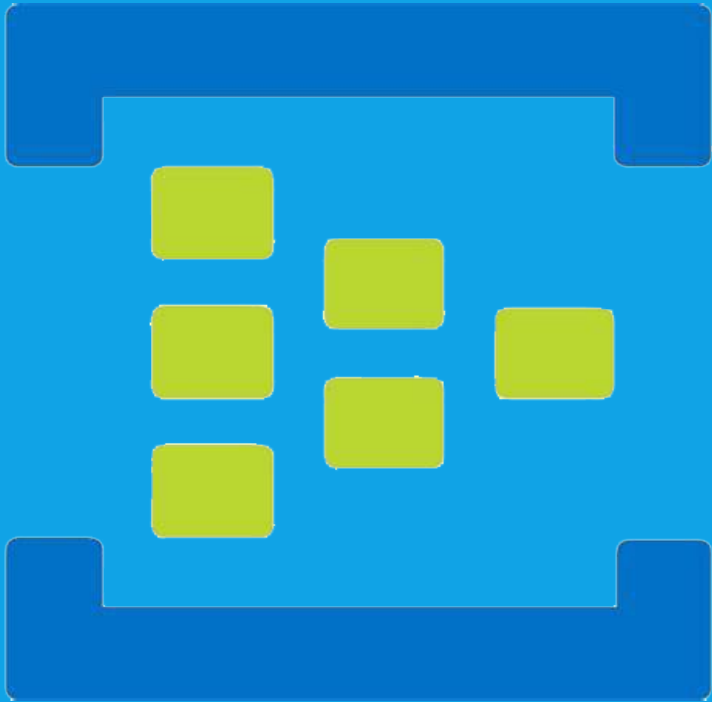
Simple, secure, and scalable real-time data ingestion



Fully managed, real-time data ingestion service that is simple, trusted, and scalable

Why choose Event Hubs?

Azure Event Hubs



Simple



Secure



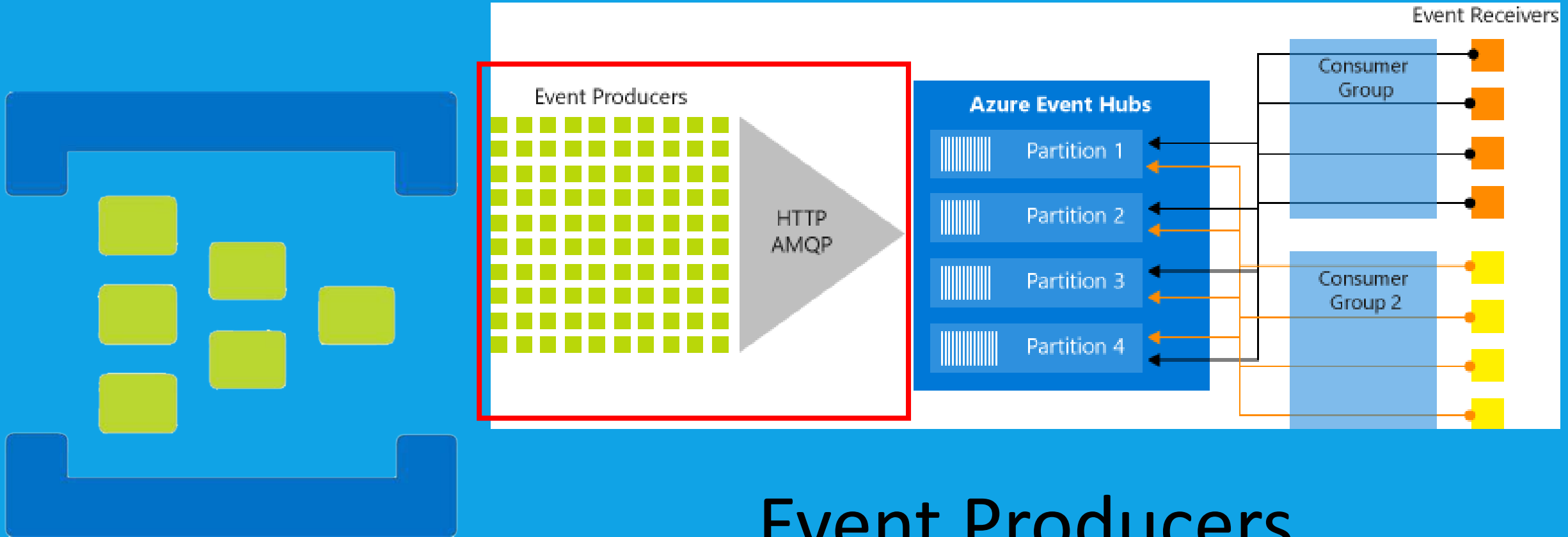
Scalable



Open

Key Architecture Components

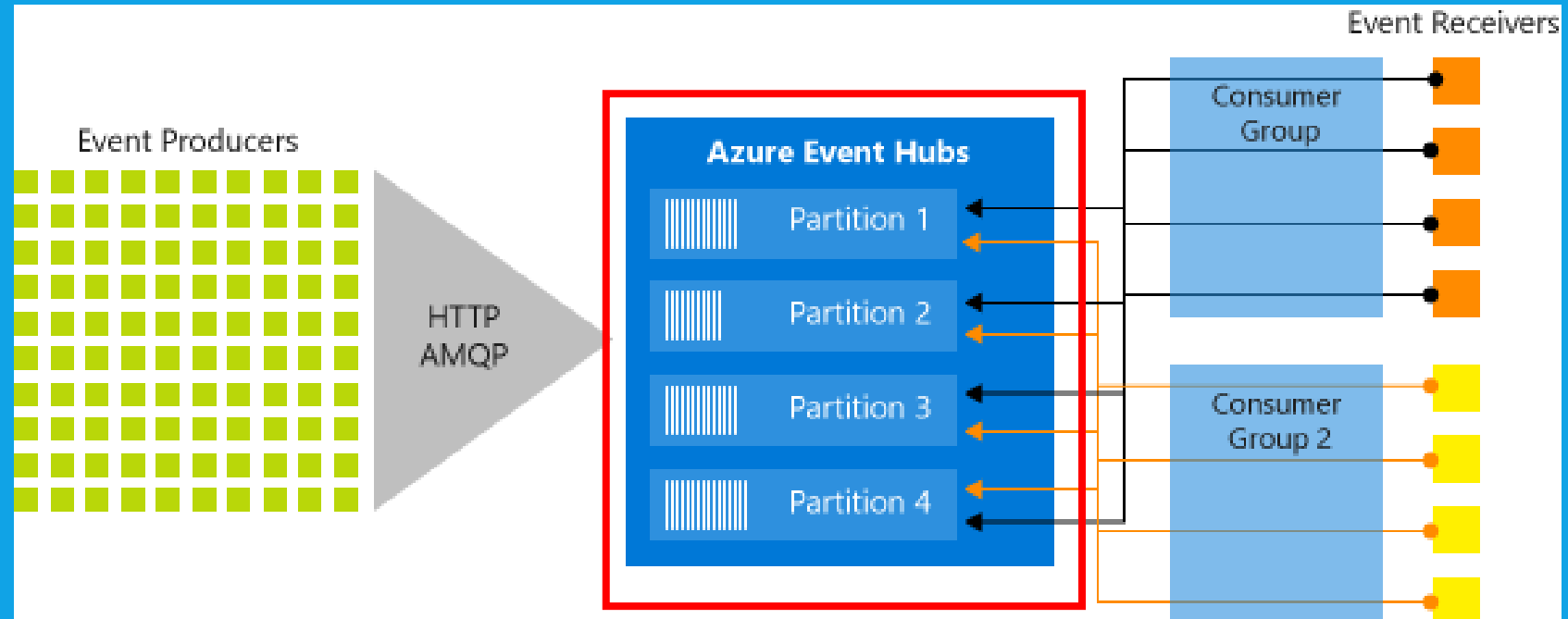
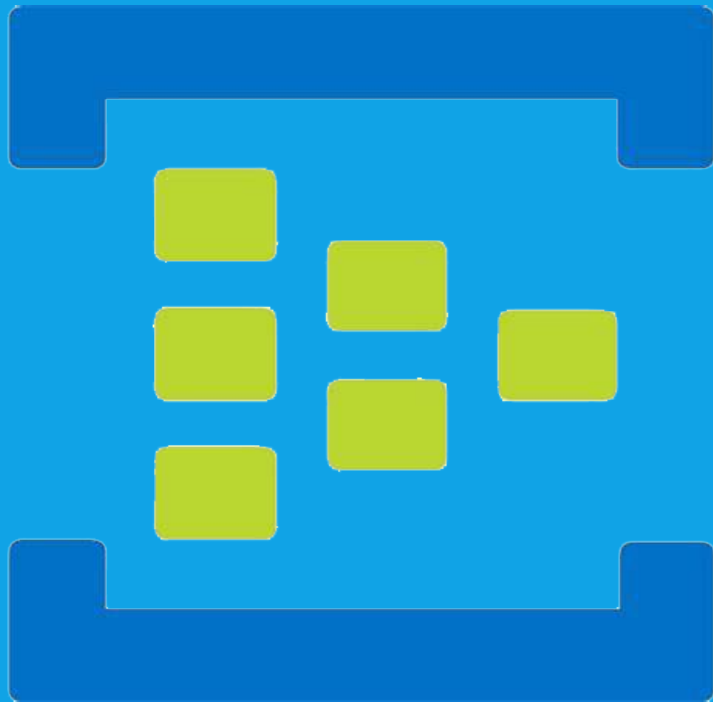
Azure Event Hubs



Event Producers

Key Architecture Components

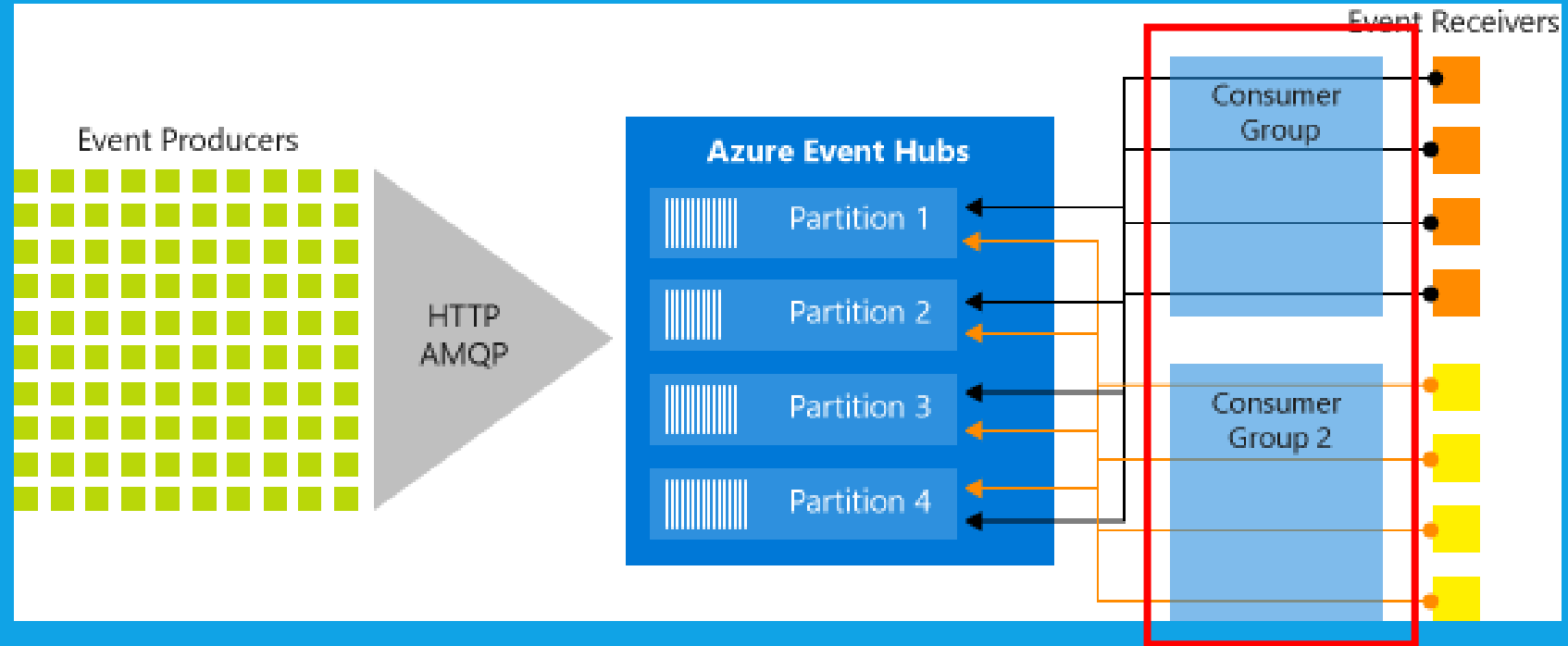
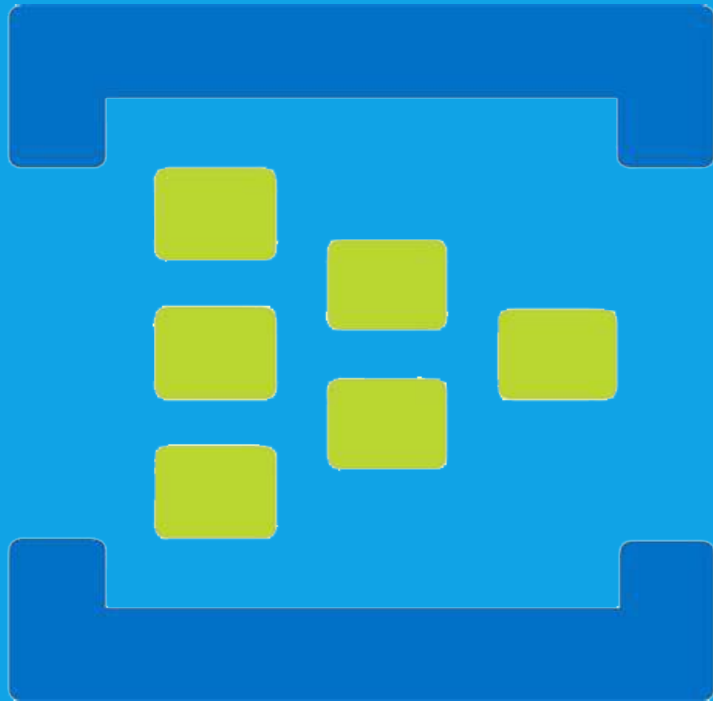
Azure Event Hubs



Partitions

Key Architecture Components

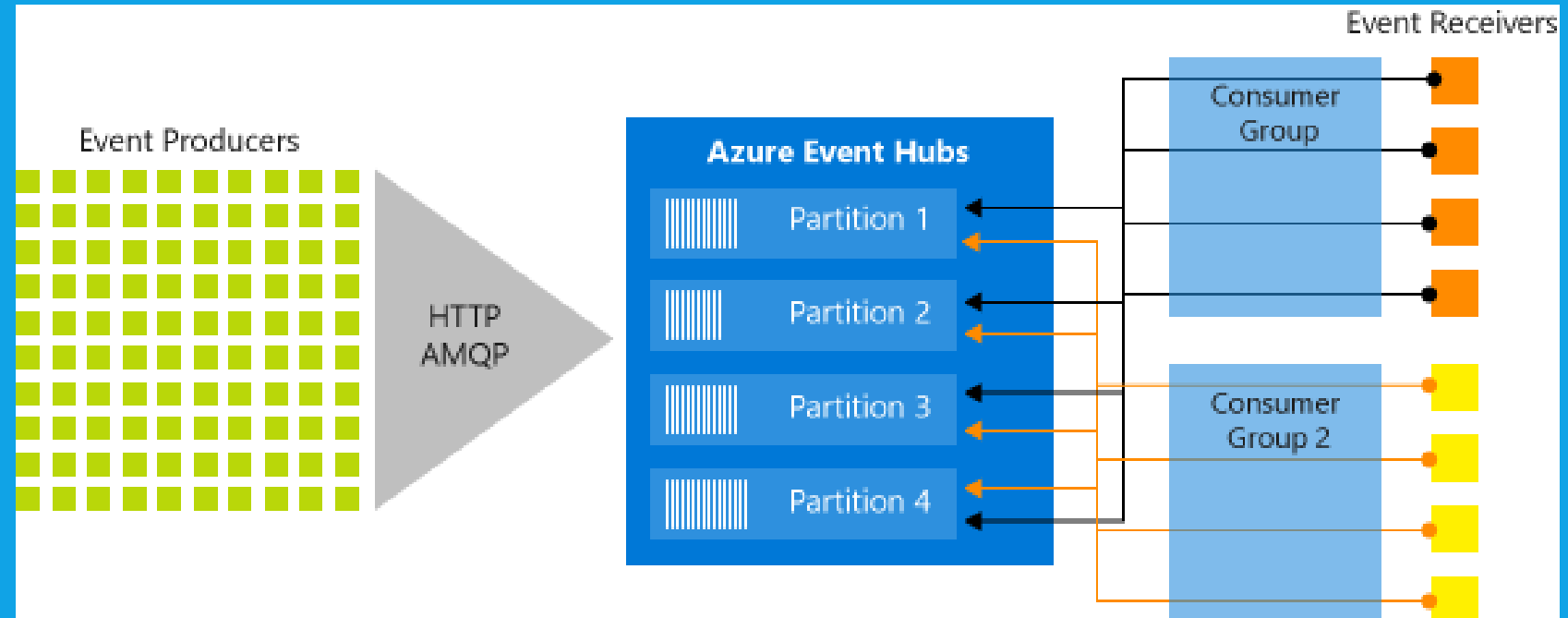
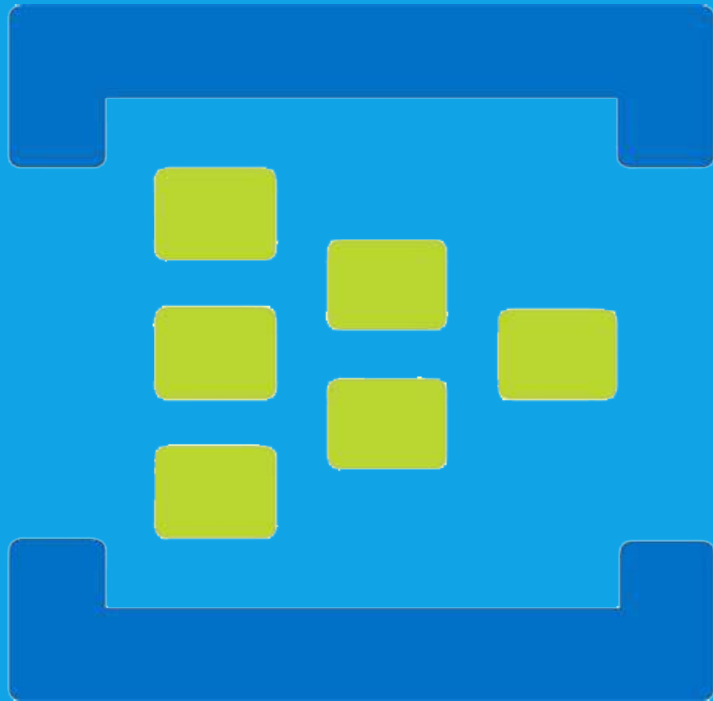
Azure Event Hubs



Consumer Groups

Key Architecture Components

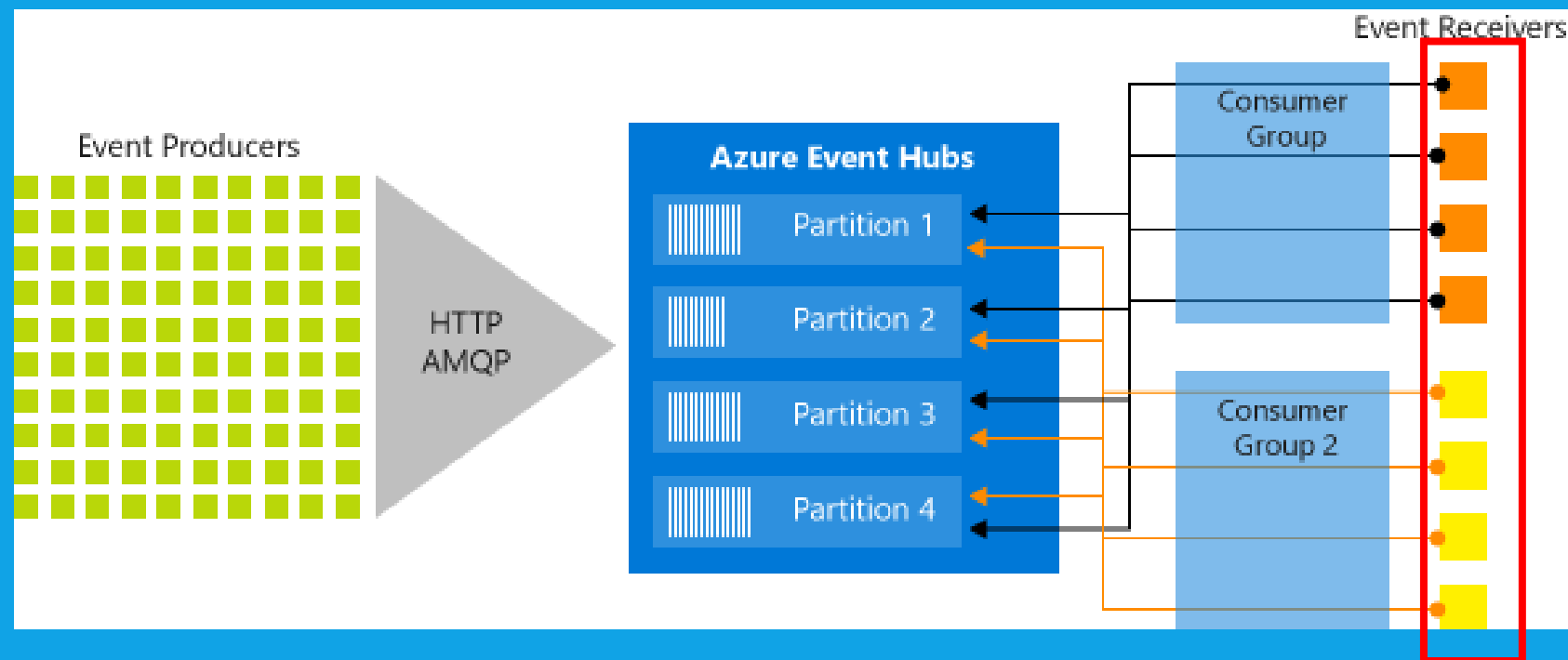
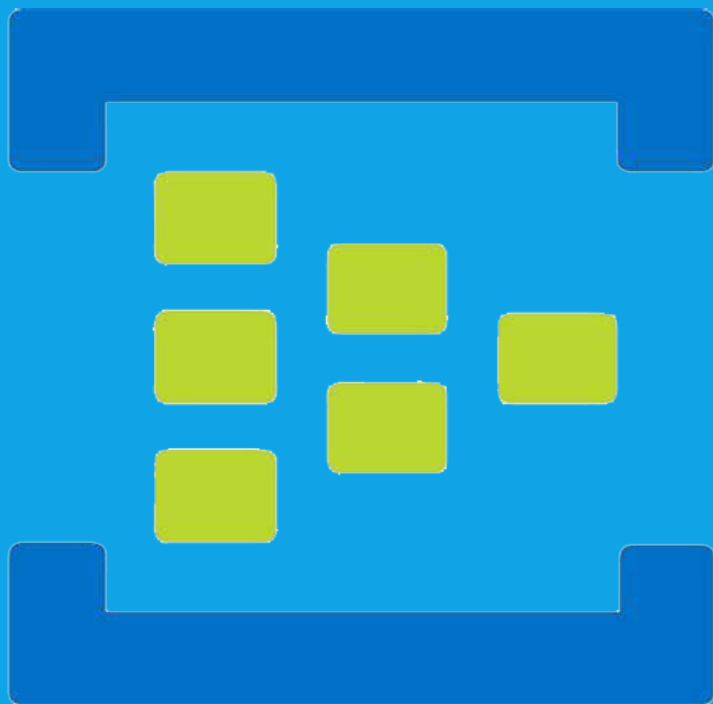
Azure Event Hubs



Throughput Units

Key Architecture Components

Azure Event Hubs

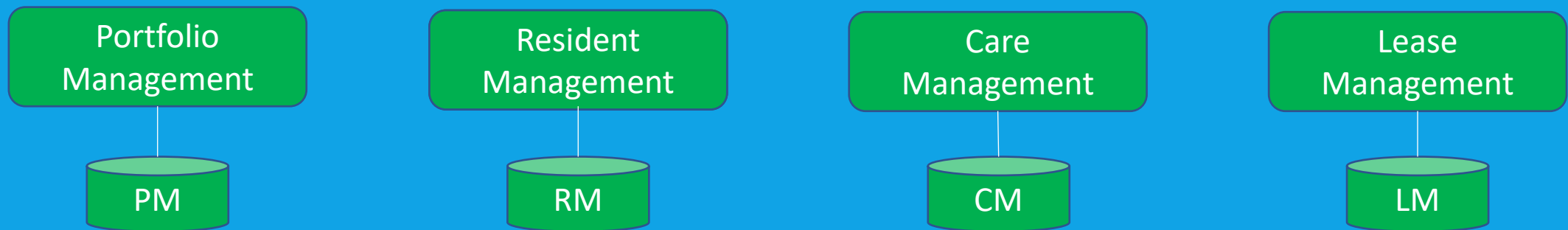


Event Receivers

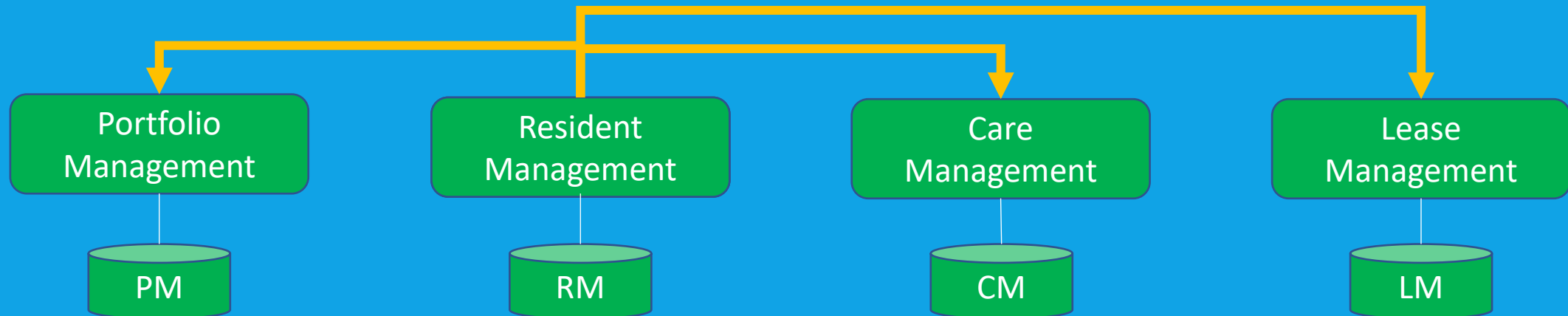
Demonstration

Building Event-Driven Microservices

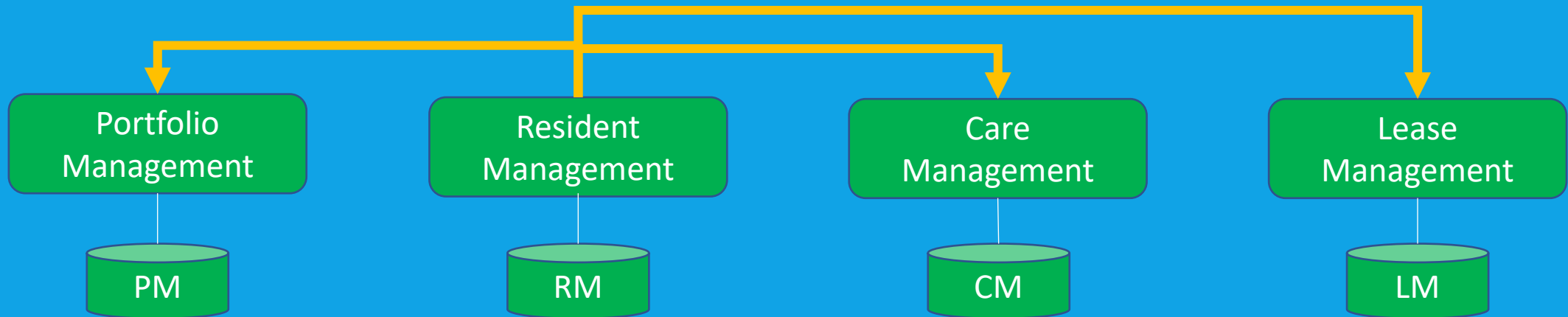




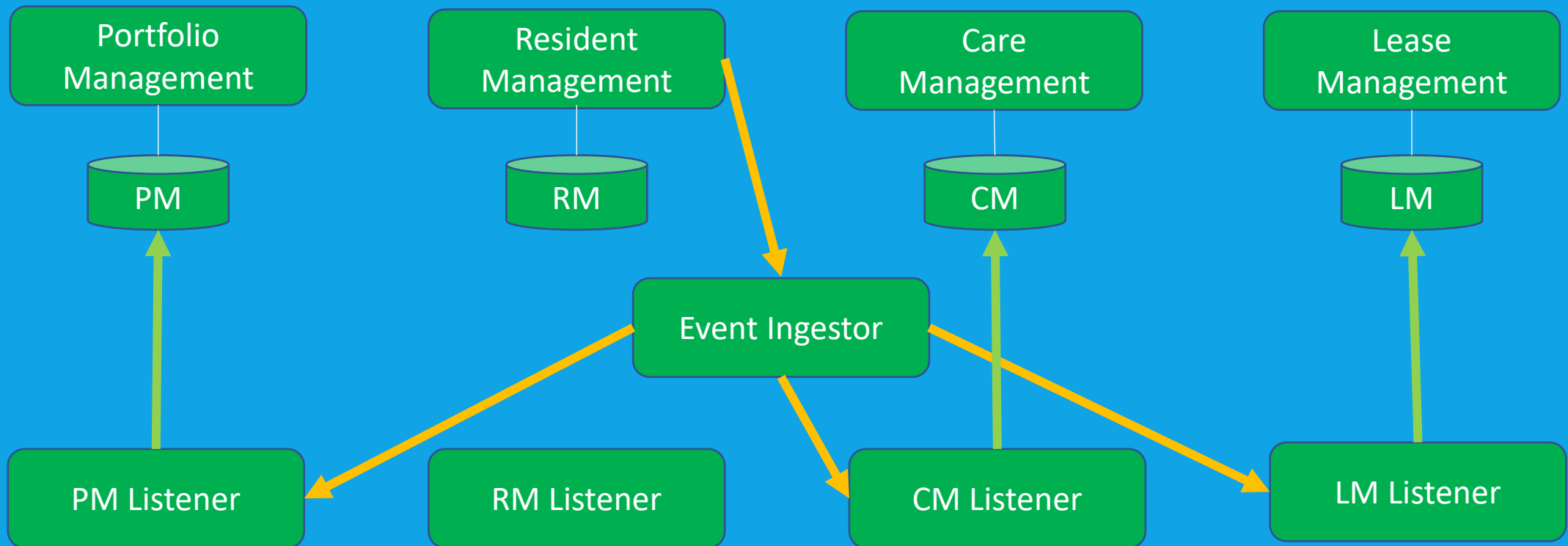
Resident Move-In



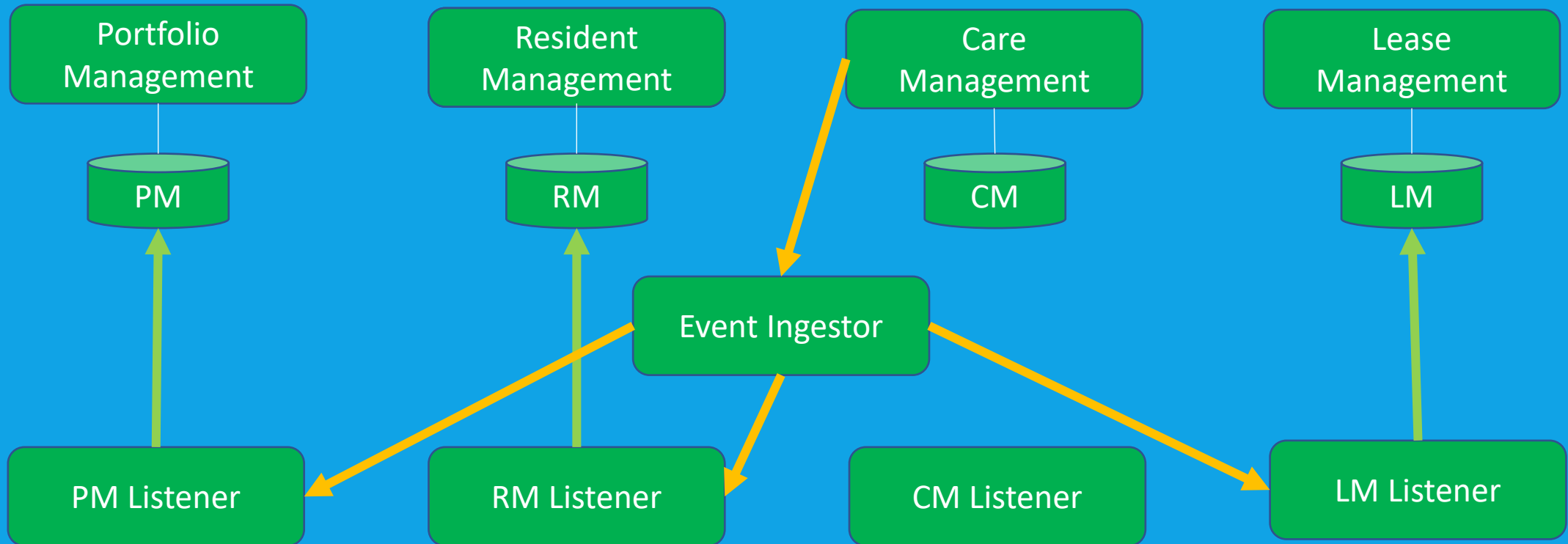
Resident Move-In



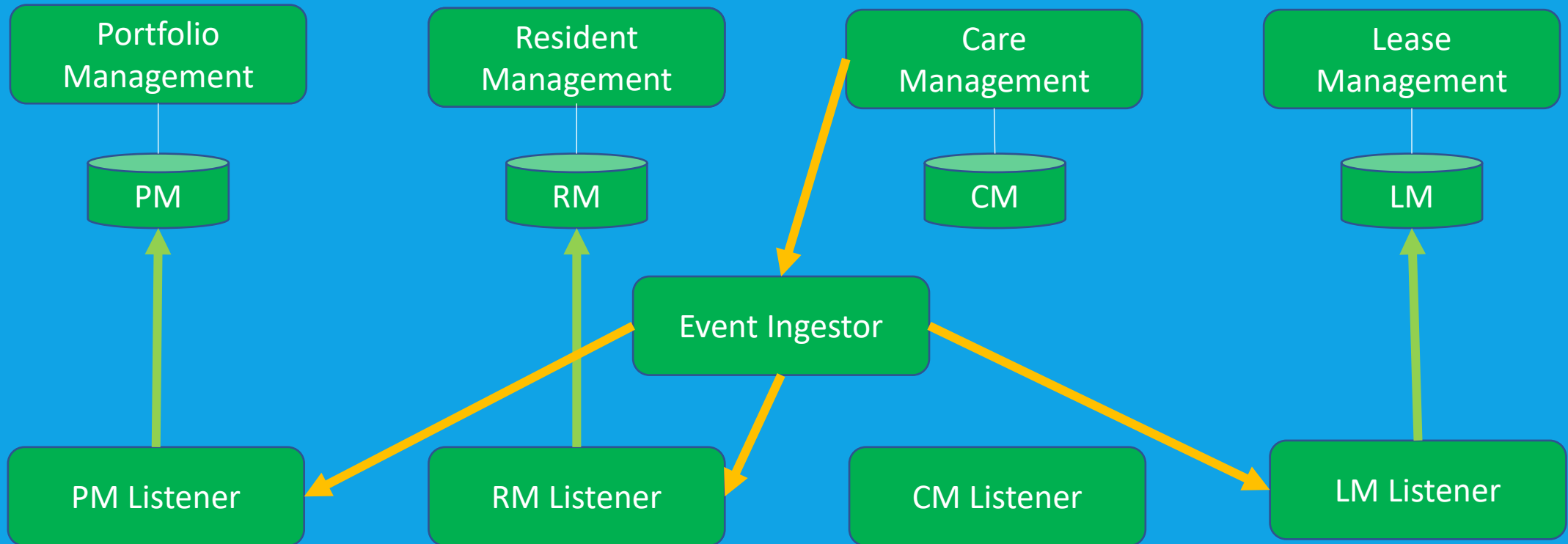
Resident Move-In



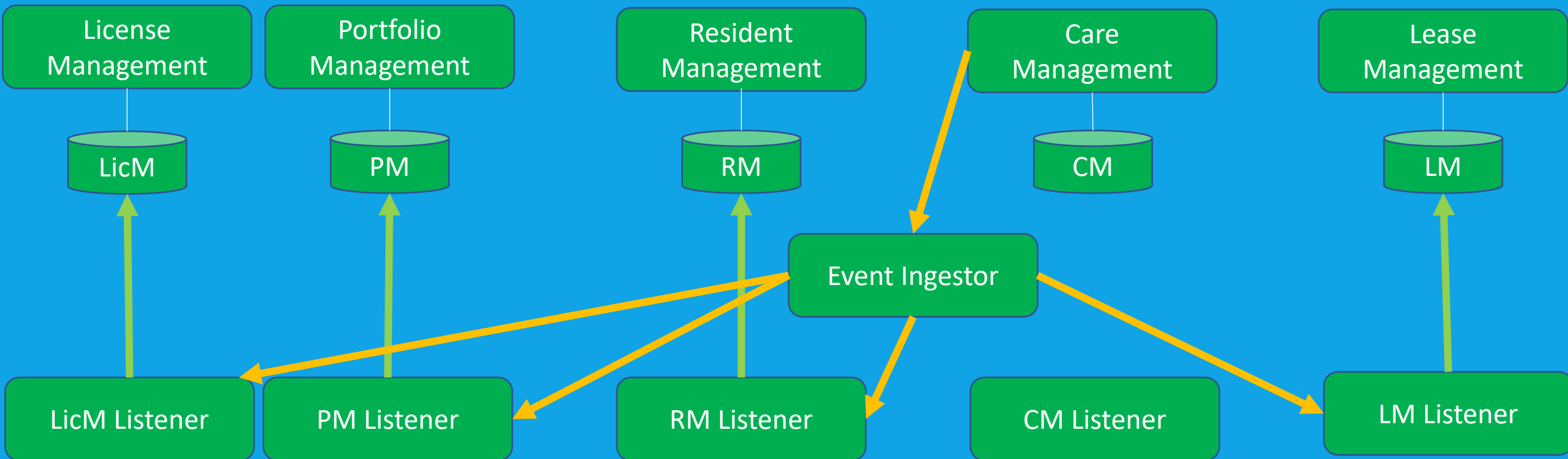
Upgrade Care



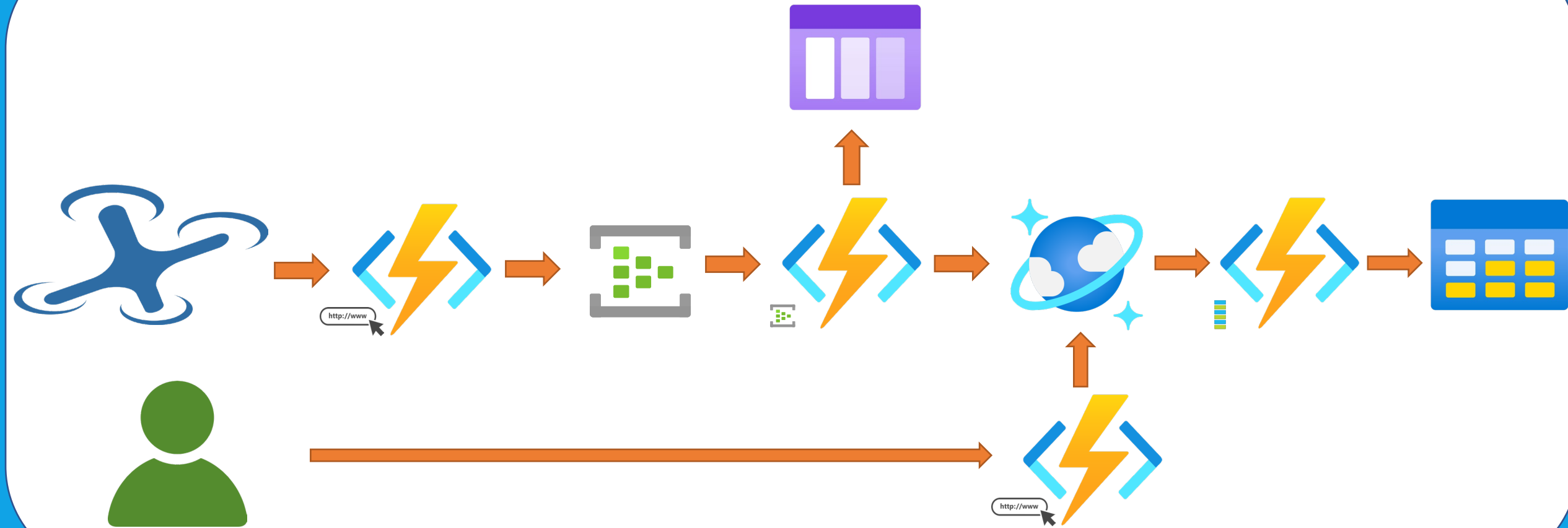
Upgrade Care



Upgrade Care



Drone Telemetry



Summary

Building Event-Driven Microservices

Event-Driven Architecture

Summary



Event-driven architecture (EDA) is a design paradigm in which a software component executes in response to receiving one or more event notifications.

EDA is more loosely coupled than client/server paradigm because the **component that sends the notification doesn't know the identity of the receiving components** at the time of compiling

- Garner -

SWOT

Summary

Strengths

- Decoupling
- Encapsulation
- Responsive
- Scalable /
Distributed
- Independence

SWOT

Summary

Strengths

- Decoupling
- Encapsulation
- Responsive
- Scalable / Distributed
- Independence

Weaknesses

- Steep Learning Curve
- Complexity
- Loss of Transactionality
- Lineage

SWOT

Summary

Strengths

- Decoupling
- Encapsulation
- Responsive
- Scalable / Distributed
- Independence

Weaknesses

- Steep Learning Curve
- Complexity
- Loss of Transactionality
- Lineage

Opportunities

- Multiple Subsystems
- Real-Time Processing
- Complex Event Processing
- High Volume / Velocity Data

SWOT

Summary

Strengths

- Decoupling
- Encapsulation
- Responsive
- Scalable / Distributed
- Independence

Weaknesses

- Steep Learning Curve
- Complexity
- Loss of Transactionality
- Lineage

Opportunities

- Multiple Subsystems
- Real-Time Processing
- Complex Event Processing
- High Volume / Velocity Data

Threats

- No Guaranteed Delivery
- Potential Sequencing Issues

SWOT

Summary

Strengths

- Decoupling
- Encapsulation
- Responsive
- Scalable / Distributed
- Independence

Weaknesses

- Steep Learning Curve
- Complexity
- Loss of Transactionality
- Lineage

Opportunities

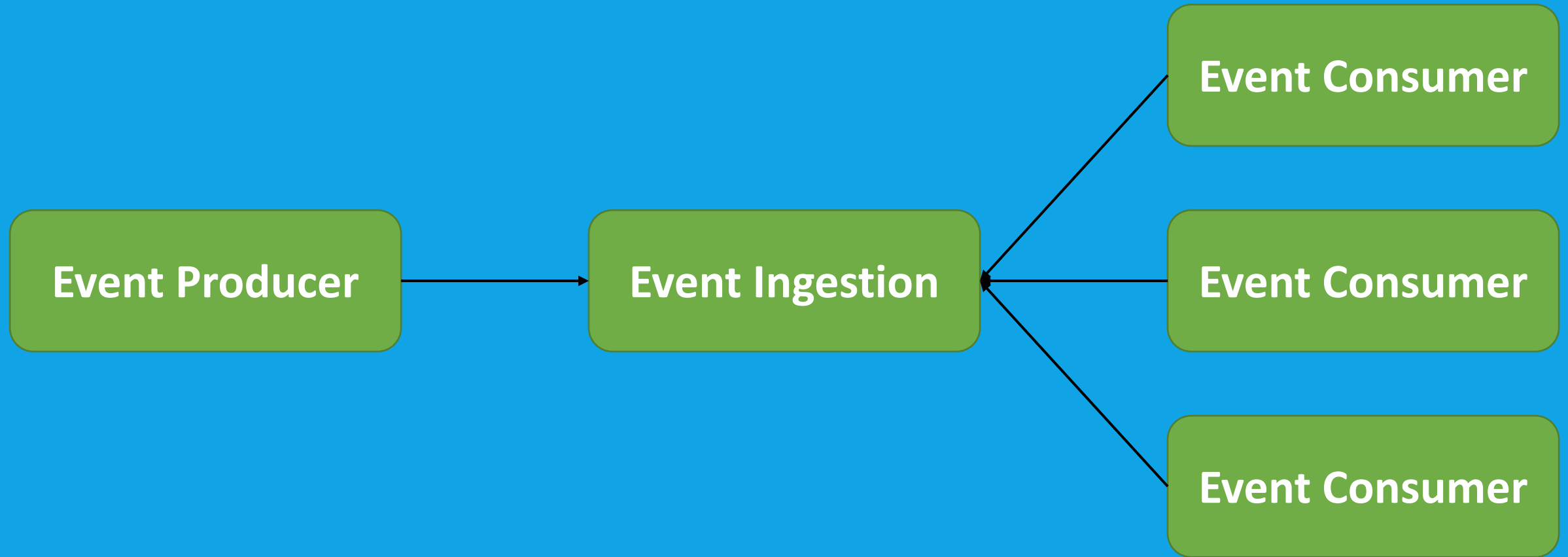
- Multiple Subsystems
- Real-Time Processing
- Complex Event Processing
- High Volume / Velocity Data

Threats

- No Guaranteed Delivery
- Potential Sequencing Issues

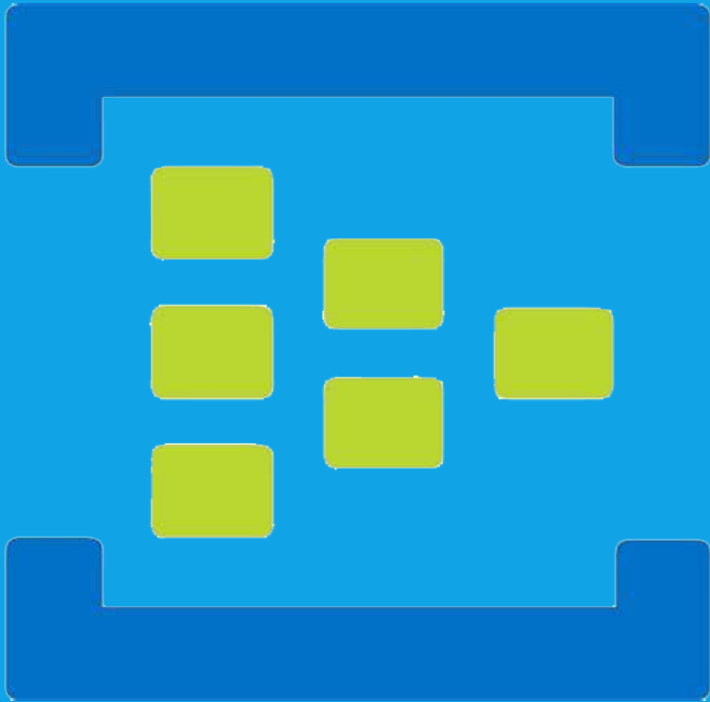
Event-Driven Architecture

Summary



Azure Event Hubs

Summary



Fully managed, real-time data ingestion service that is simple, trusted, and scalable

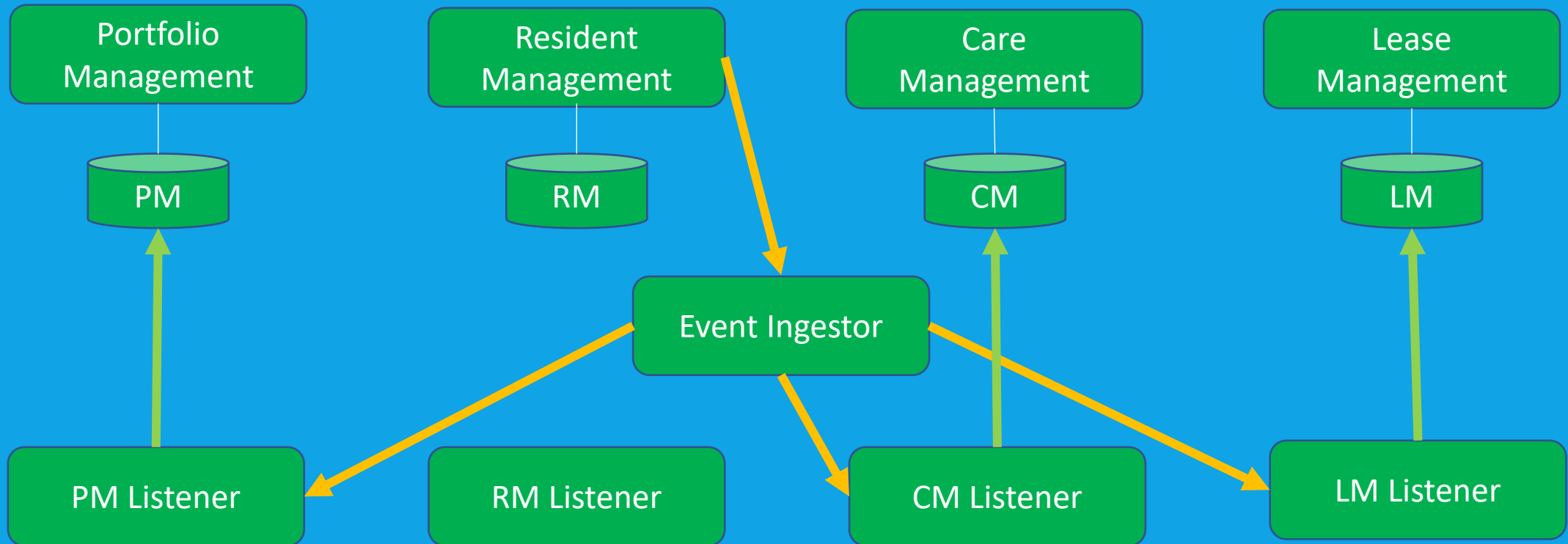
Simple

Secure

Scalable

Open

Real-World Demonstrations



Thank You

✉ chadgreen@chadgreen.com

💬 TaleLearnCode

🌐 ChadGreen.com

🐦 ChadGreen & TaleLearnCode

🌐 ChadwickEGreen

