

# Consuming Events in a .NET Core Application

---



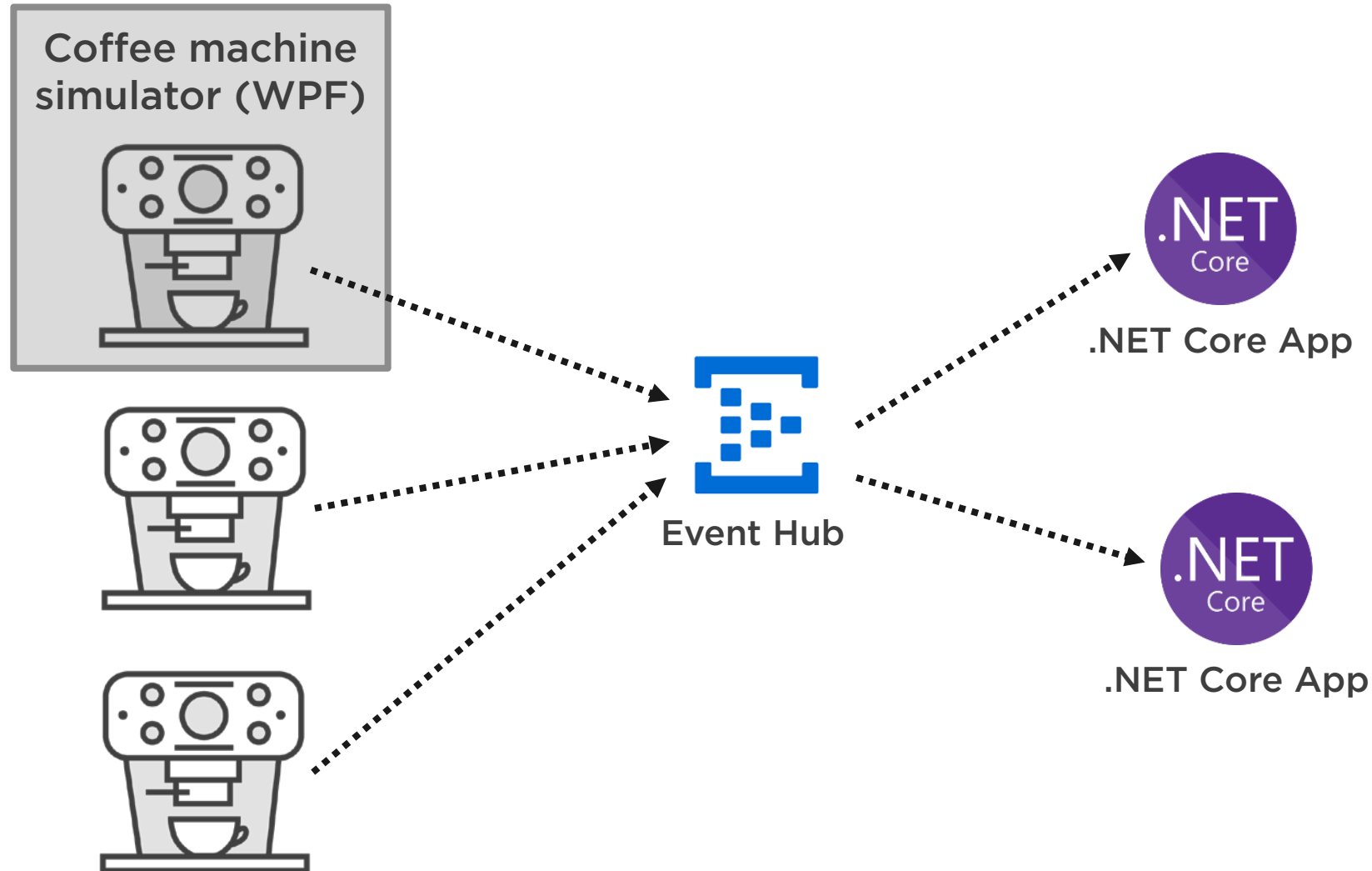
**Thomas Claudius Huber**

MICROSOFT MVP (WINDOWS DEVELOPMENT)

@thomasclaudiush [www.thomasclaudiushuber.com](http://www.thomasclaudiushuber.com)



# The Wired Brain Coffee Machine Scenario



# Module Outline



**Consume events with  
an EventHubClient**

**Consume events with an  
EventProcessorHost**

- Stores the offset in Azure
- Synchronizes reader instances  
for horizontal scaling

**Consumer groups and partitions**

# Demo



**Build a .NET Core console application and initialize an EventHubClient**

**Create a PartitionReceiver**

- Receive events from a specific partition

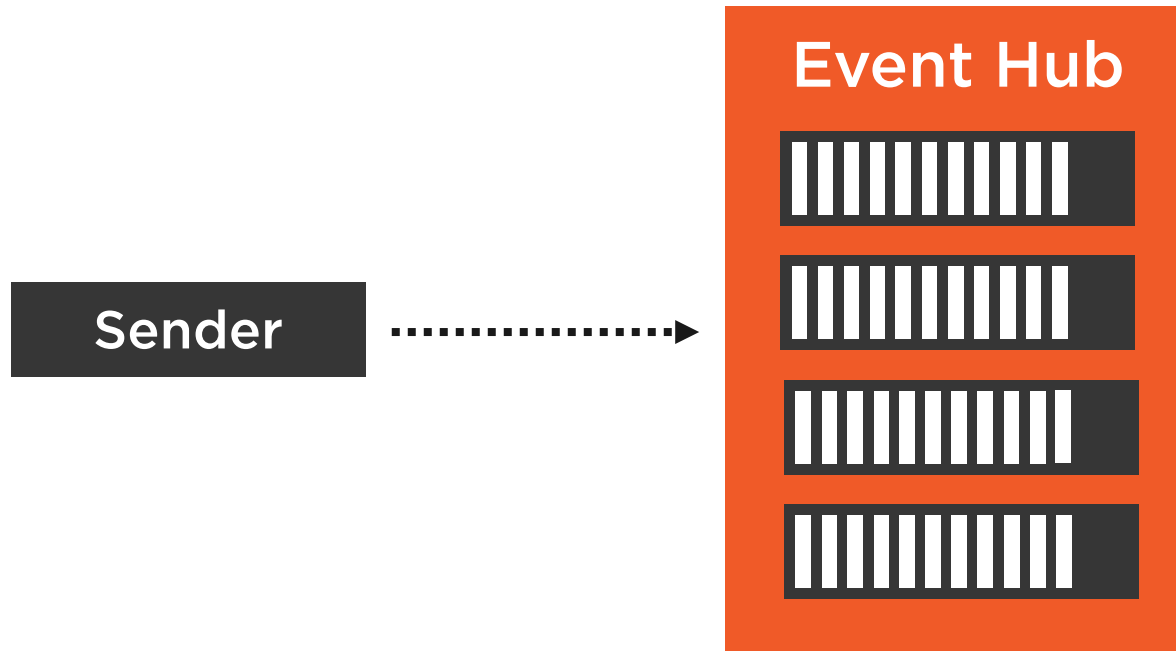
**Receive events from all partitions**

- Multiple PartitionReceivers

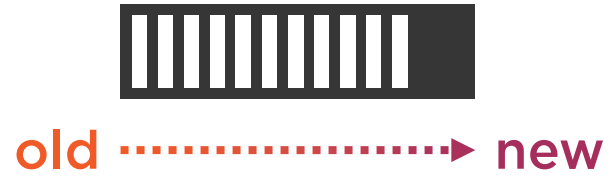
**Deserialize a CoffeeMachineData object from the received JSON**



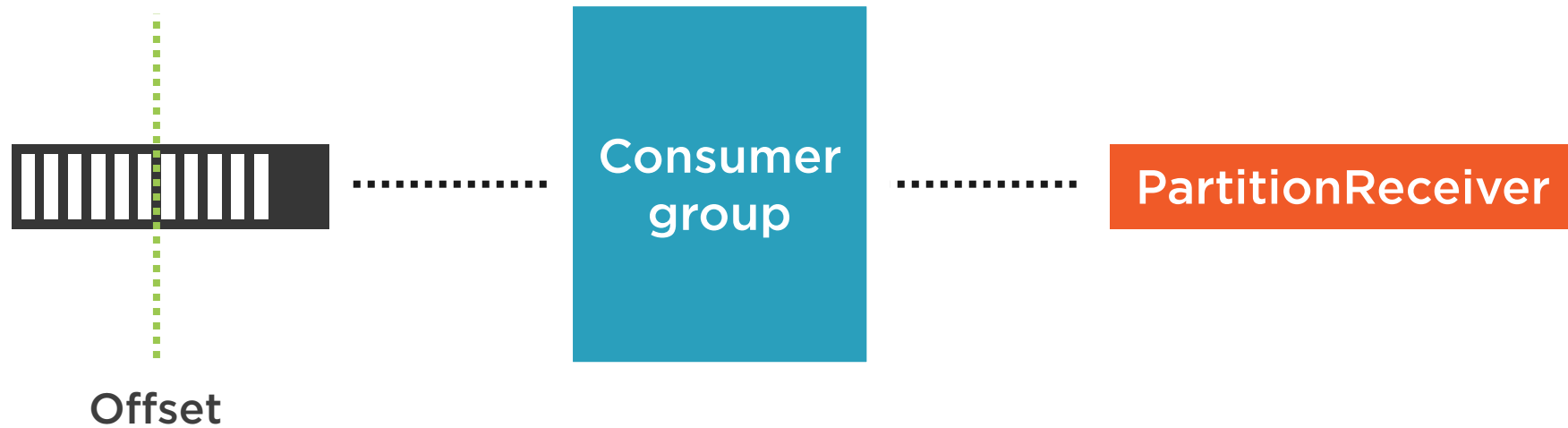
# Reading Events from a Specific Offset



# Reading Events from a Specific Offset



# Reading Events from a Specific Offset



# Demo

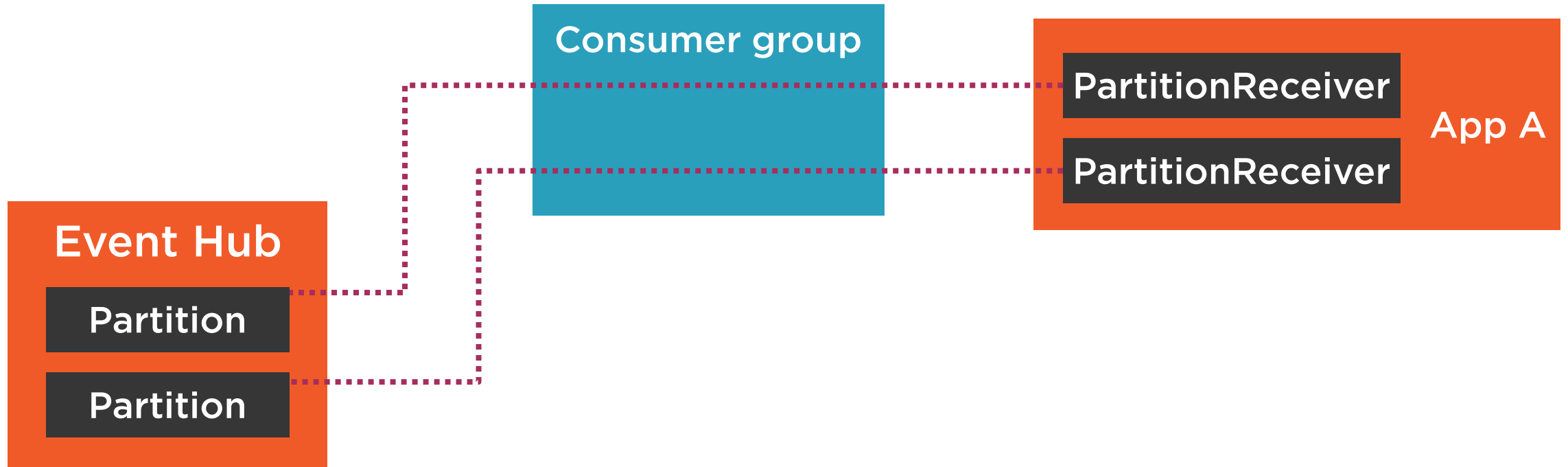


Use a partition's offset to receive events





# Understand Consumer Groups



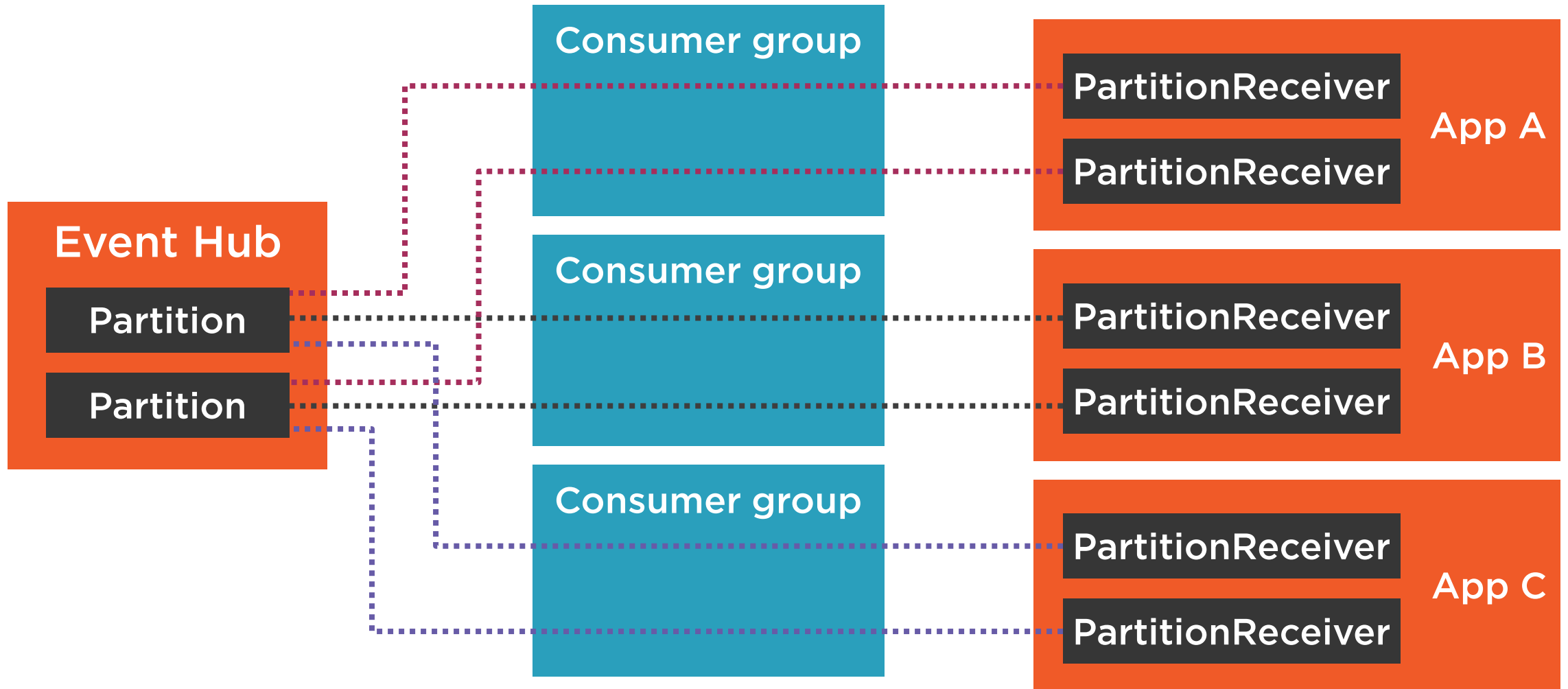
It is recommended to have  
just one active connection  
per partition  
per consumer group



Like a `select *` view  
on the Event Hub



# Understand Consumer Groups



You can create up to  
20 consumer groups  
in the Azure portal



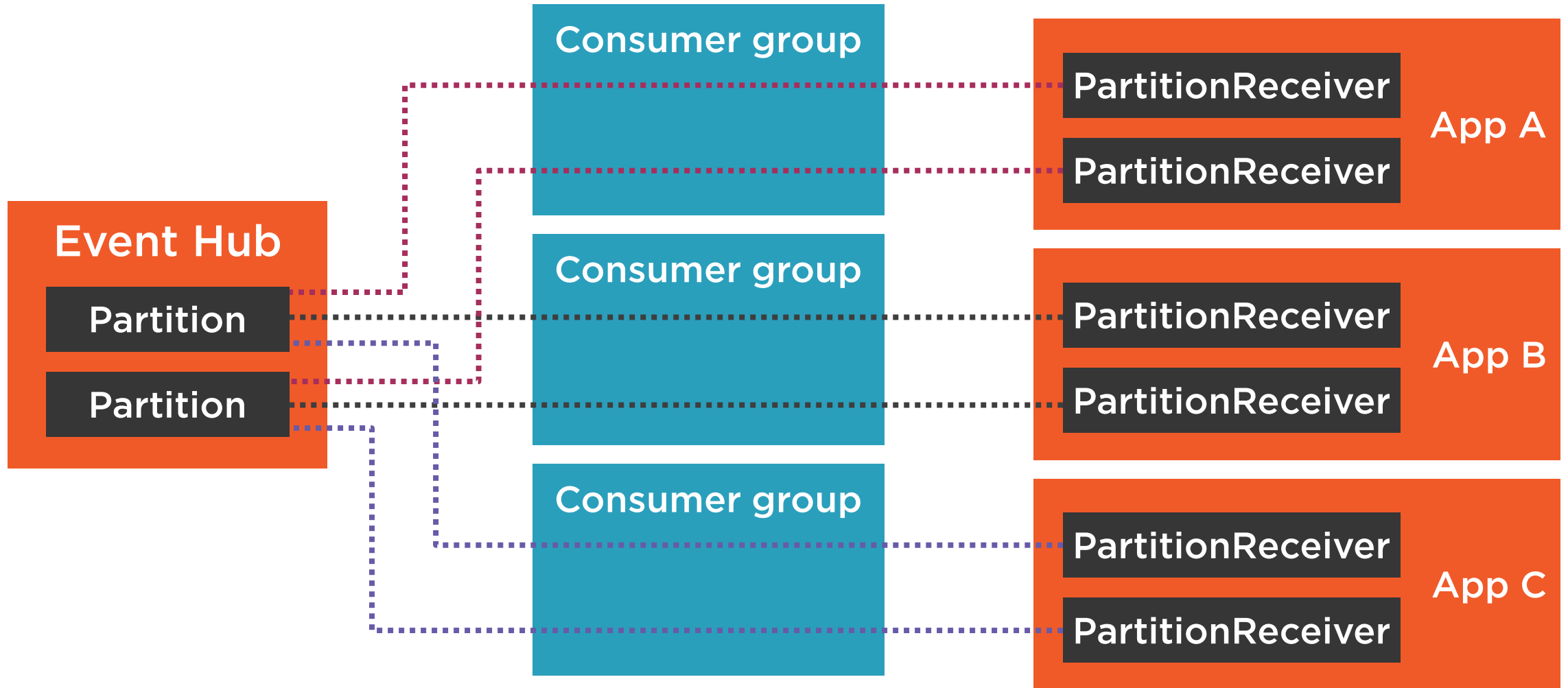
# Demo



Create a new consumer group  
in the Azure portal



# Understand More About Partitions



Set the number of your  
partitions to the anticipated  
number of concurrent  
consumers





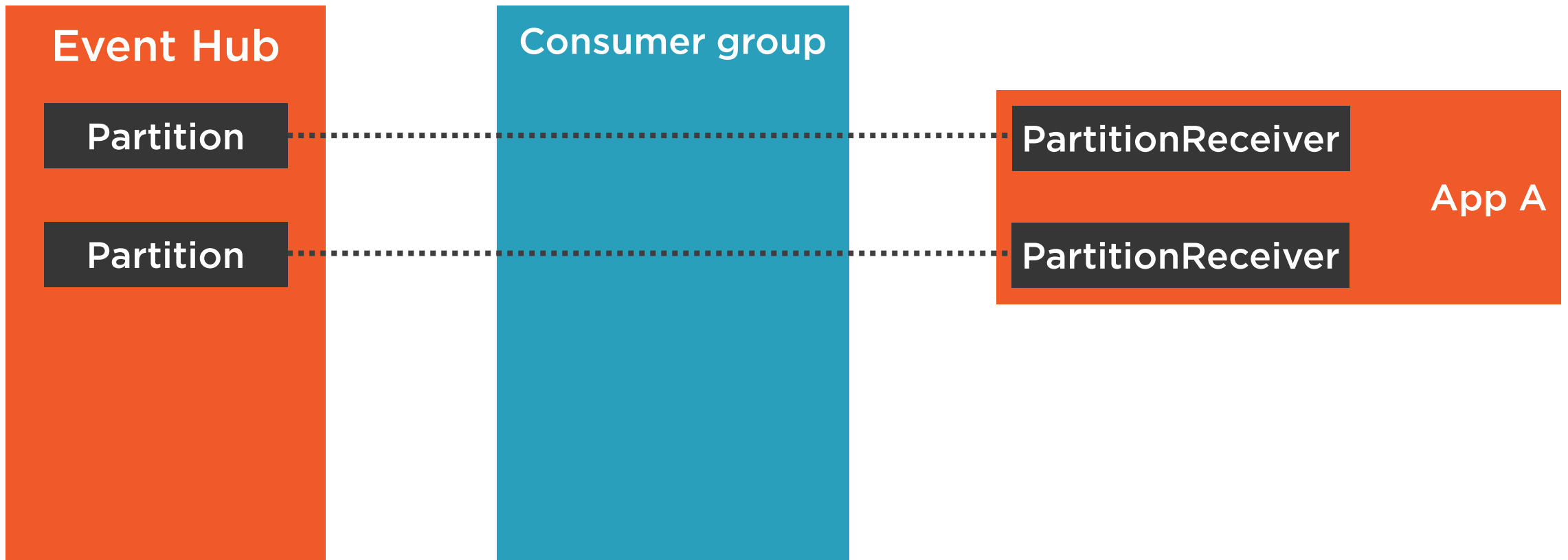
Set the number of your  
partitions to the anticipated  
number of concurrent  
consumers



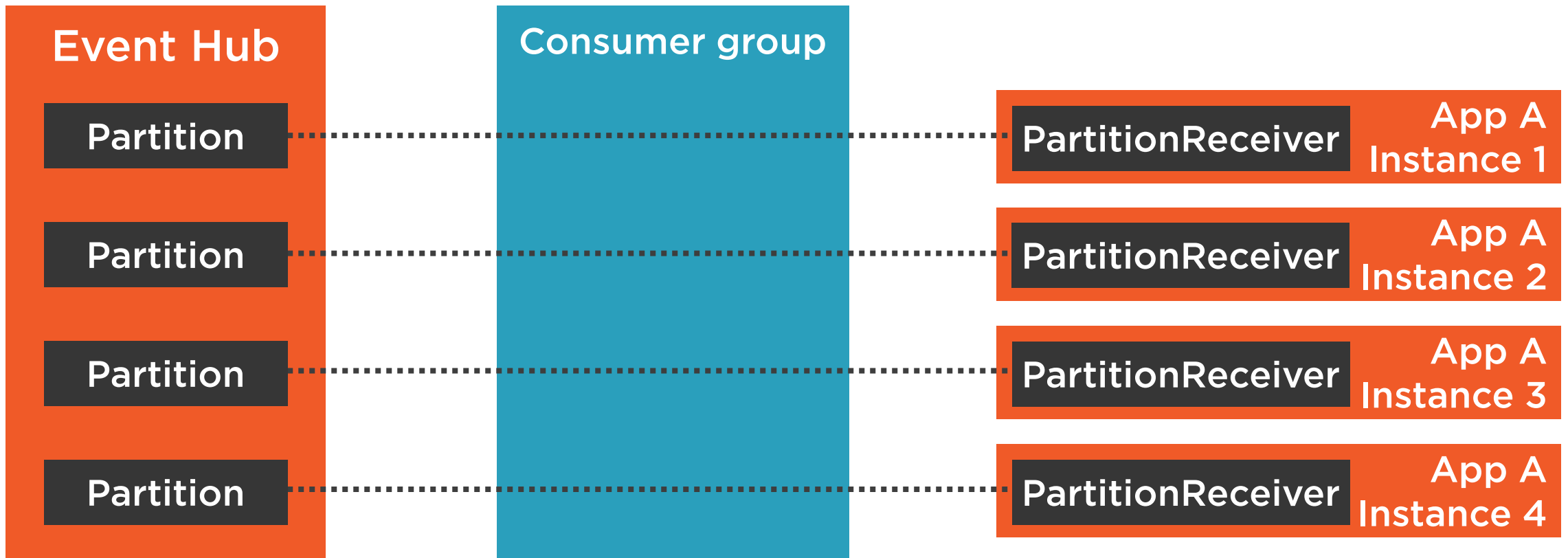
Set the number of your  
partitions to the anticipated  
number of concurrent  
reader instances  
within a consumer group



# Understand More About Partitions



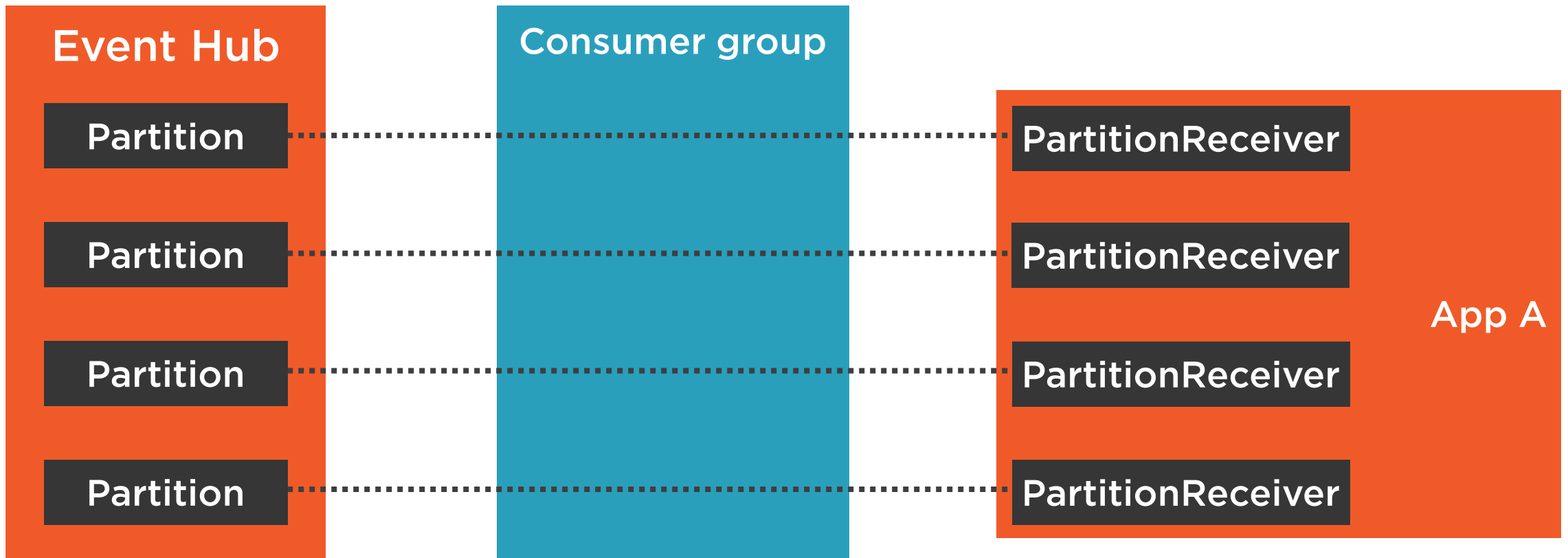
# Understand More About Partitions



Set the number of your  
partitions to the anticipated  
number of concurrent  
reader instances  
within a consumer group



# Understand More About Partitions



# Receive Events with an EventProcessorHost

**EventHubClient**

**Microsoft.Azure.EventHubs**

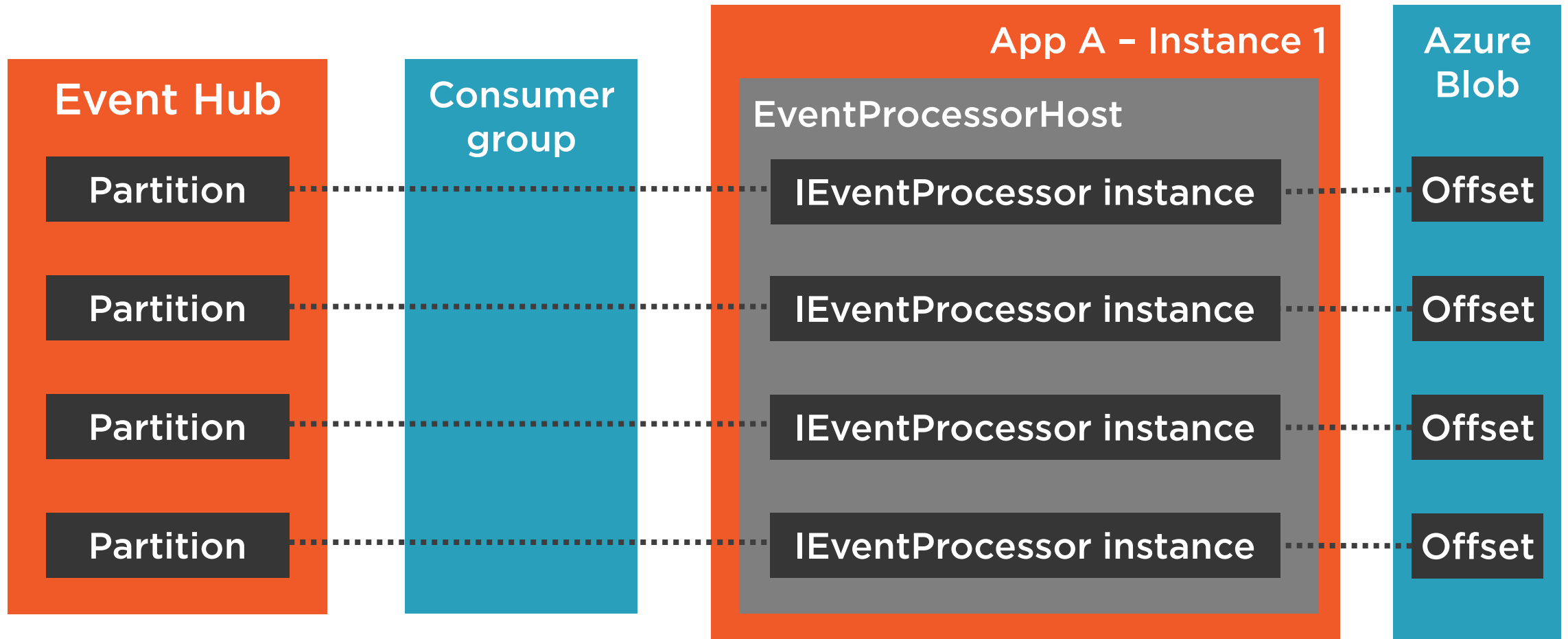
**EventProcessorHost**

**Microsoft.Azure.EventHubs.Processor**

<https://github.com/Azure/azure-event-hubs-dotnet>

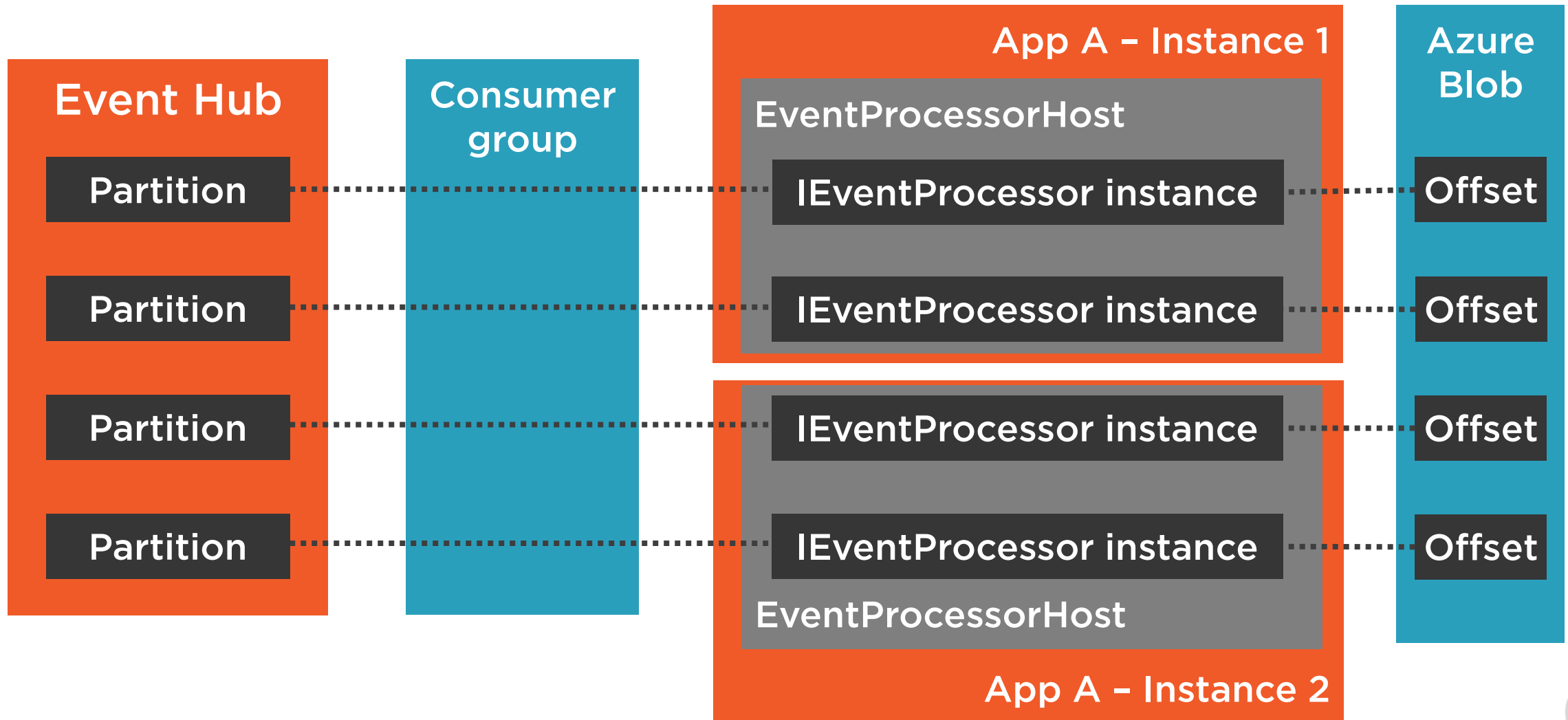


# Receive Events with an EventProcessorHost





# Receive Events with an EventProcessorHost



# Demo



**Build a .NET Core console application**  
**Implement the IEventProcessor interface**



# Demo



Register the  
**WiredBrainCoffeeEventProcessor**  
with an **EventProcessorHost**

Provide several required parameters  
to create the **EventProcessorHost**

- Event Hub connection string
- Consumer group name
- Storage connection string
- Blob container name



# Demo



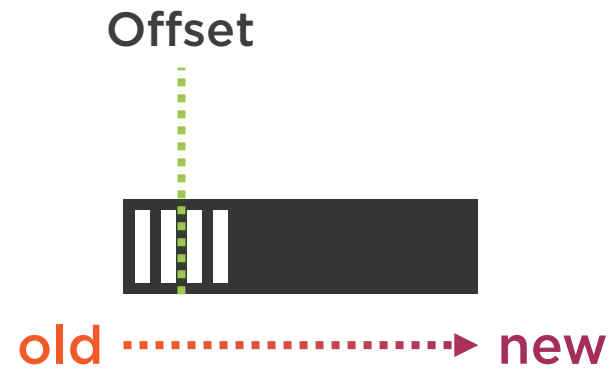
Offset handling

Look into the blob container  
used by the EventProcessorHost

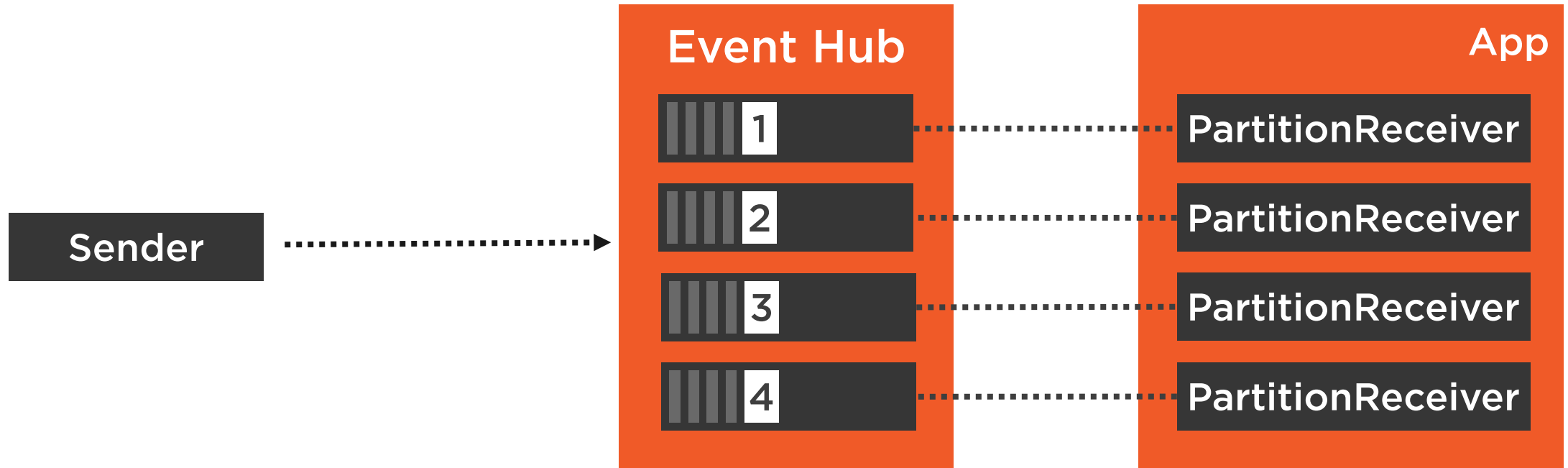
Reader synchronization



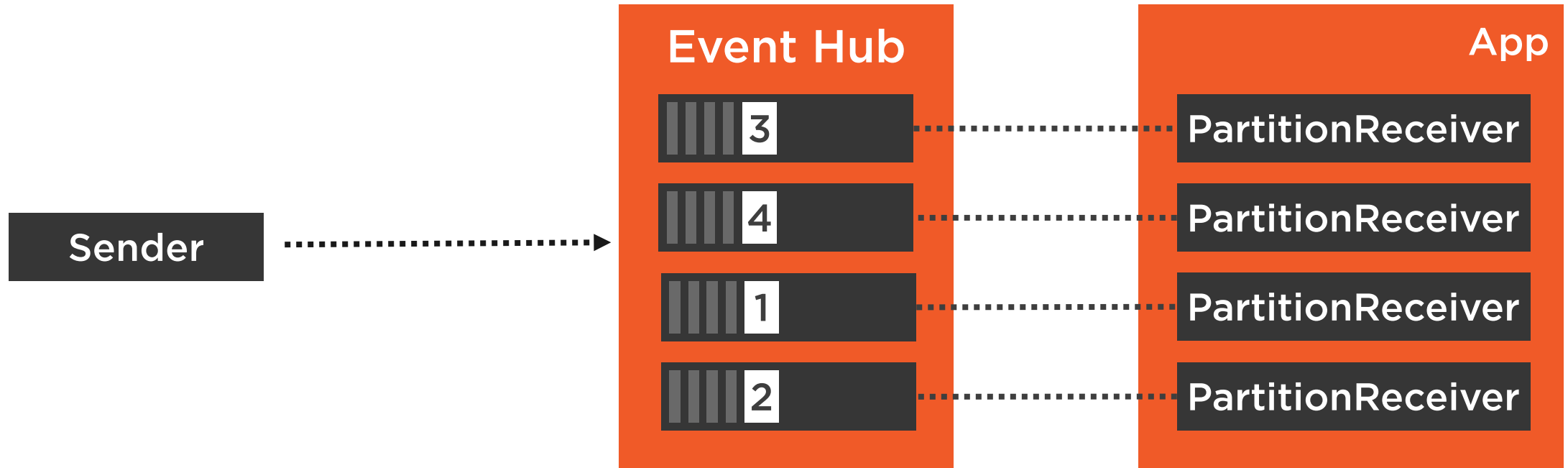
# Understand How Events Are Ordered



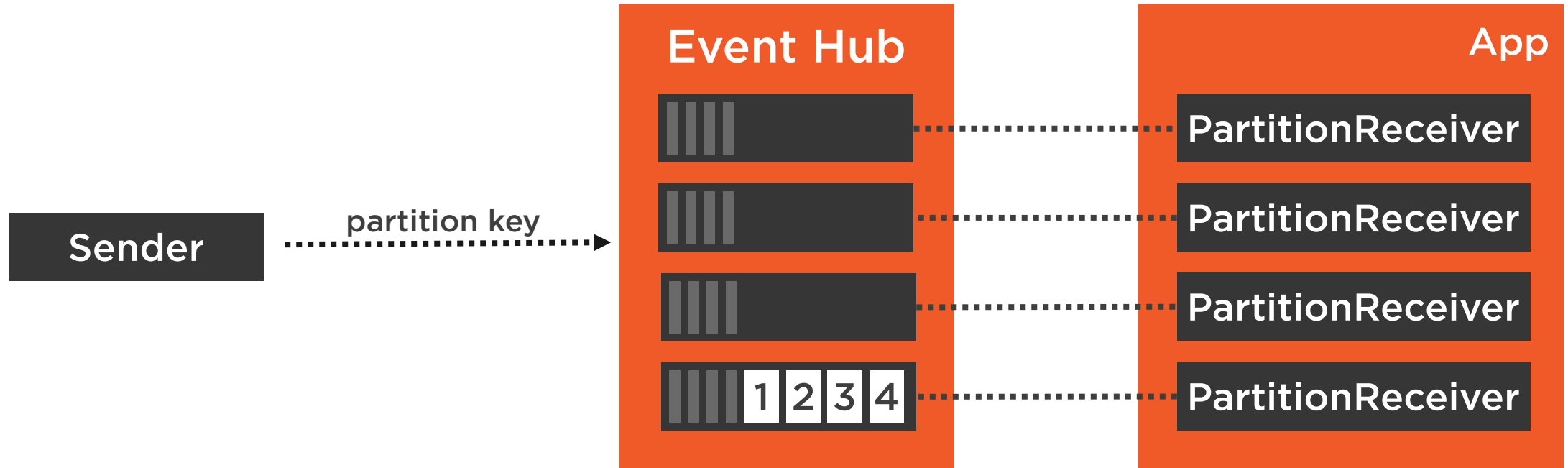
# Understand How Events Are Ordered



# Understand How Events Are Ordered



# Understand How Events Are Ordered





# Demo



Send events with a partition key



# Summary



**Consume events with  
an EventHubClient**

**Consume events with an  
EventProcessorHost**

- Stores the offset in Azure
- Synchronizes reader instances  
for horizontal scaling

**Consumer groups and partitions**

