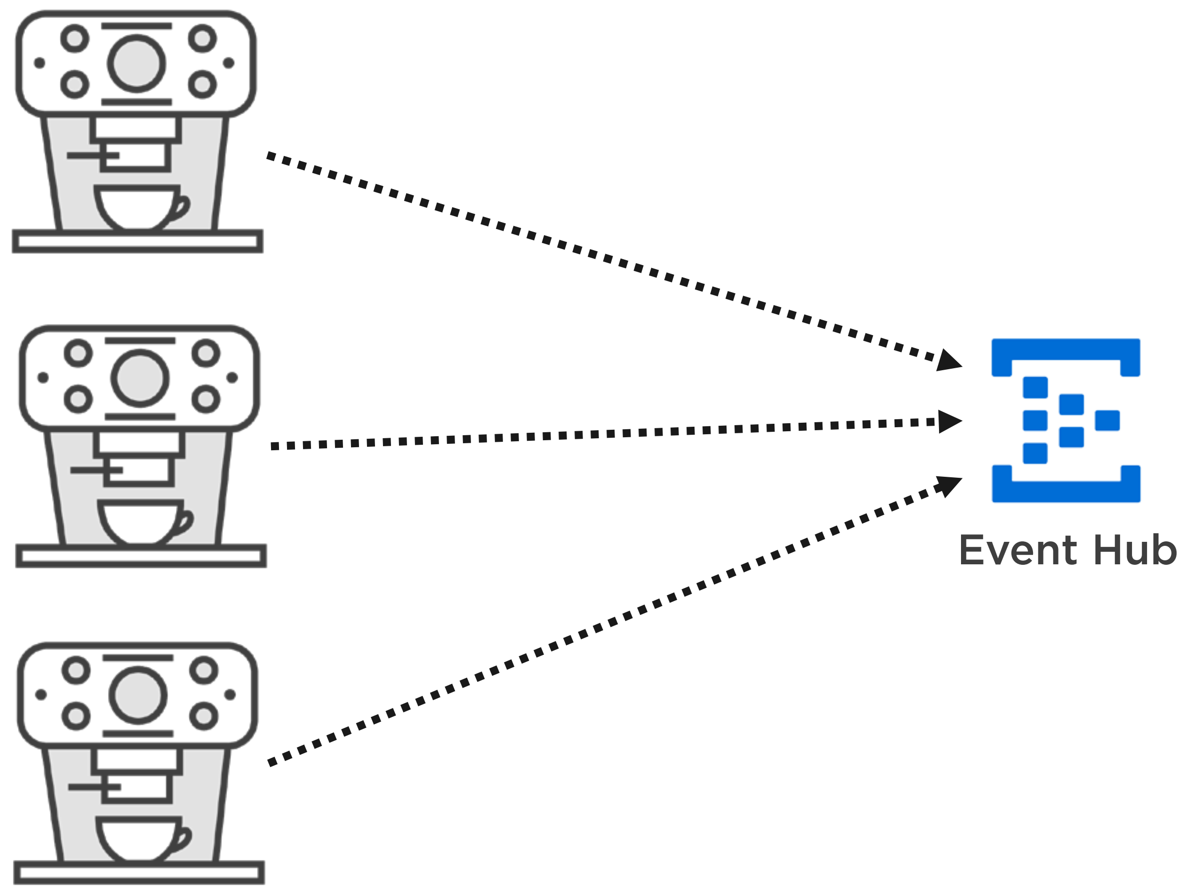
Building a WPF Simulator App to Send Events

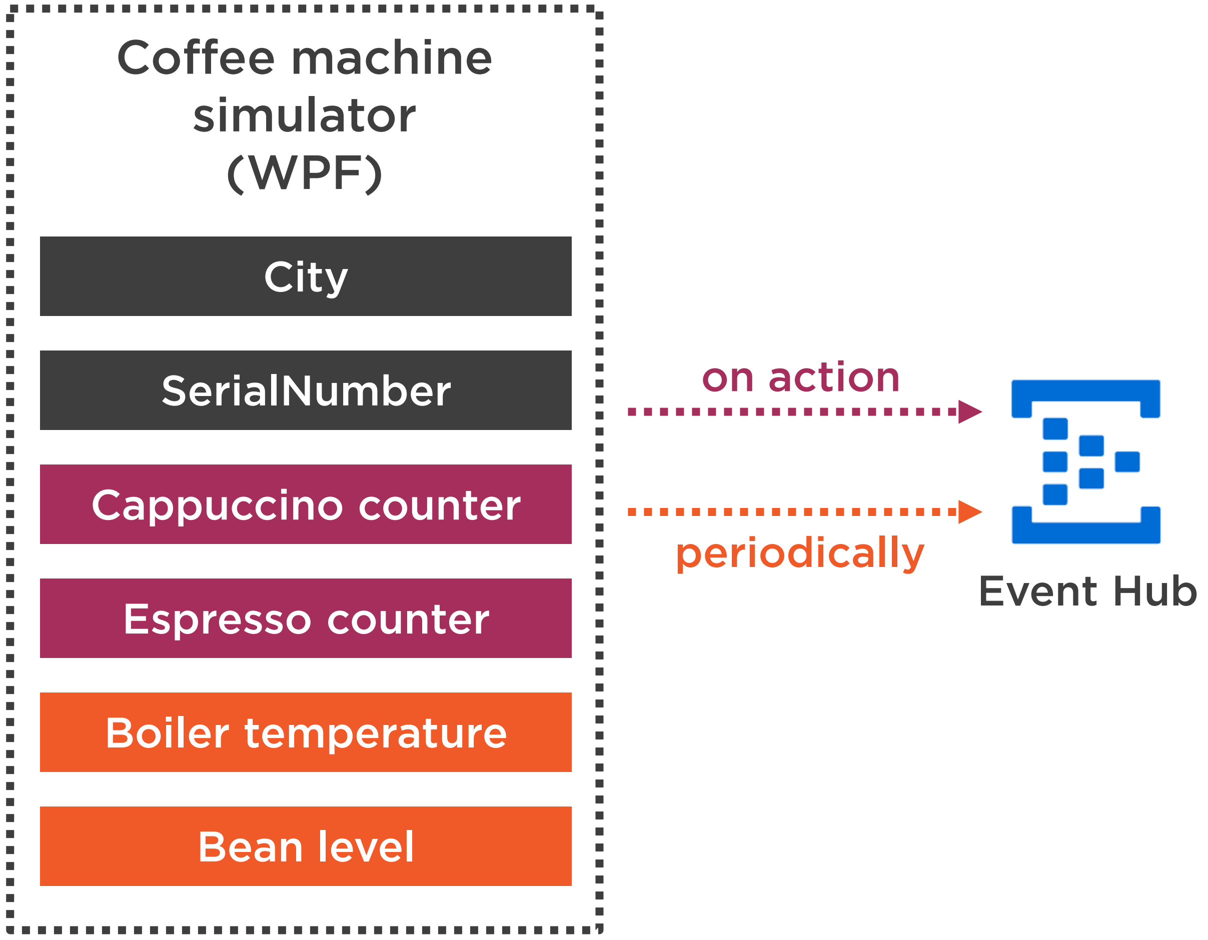
Thomas Claudius Huber
MICROSOFT MVP (WINDOWS DEVELOPMENT)

@thomasclaudiush www.thomasclaudiushuber.com

The Wired Brain Coffee Machine Scenario



The Wired Brain Coffee Machine Scenario



Module Outline

Create a WPF project

- Implement View and ViewModel

Create a .NET Standard library

- Send events to the Event Hub
- Batch multiple events for sending

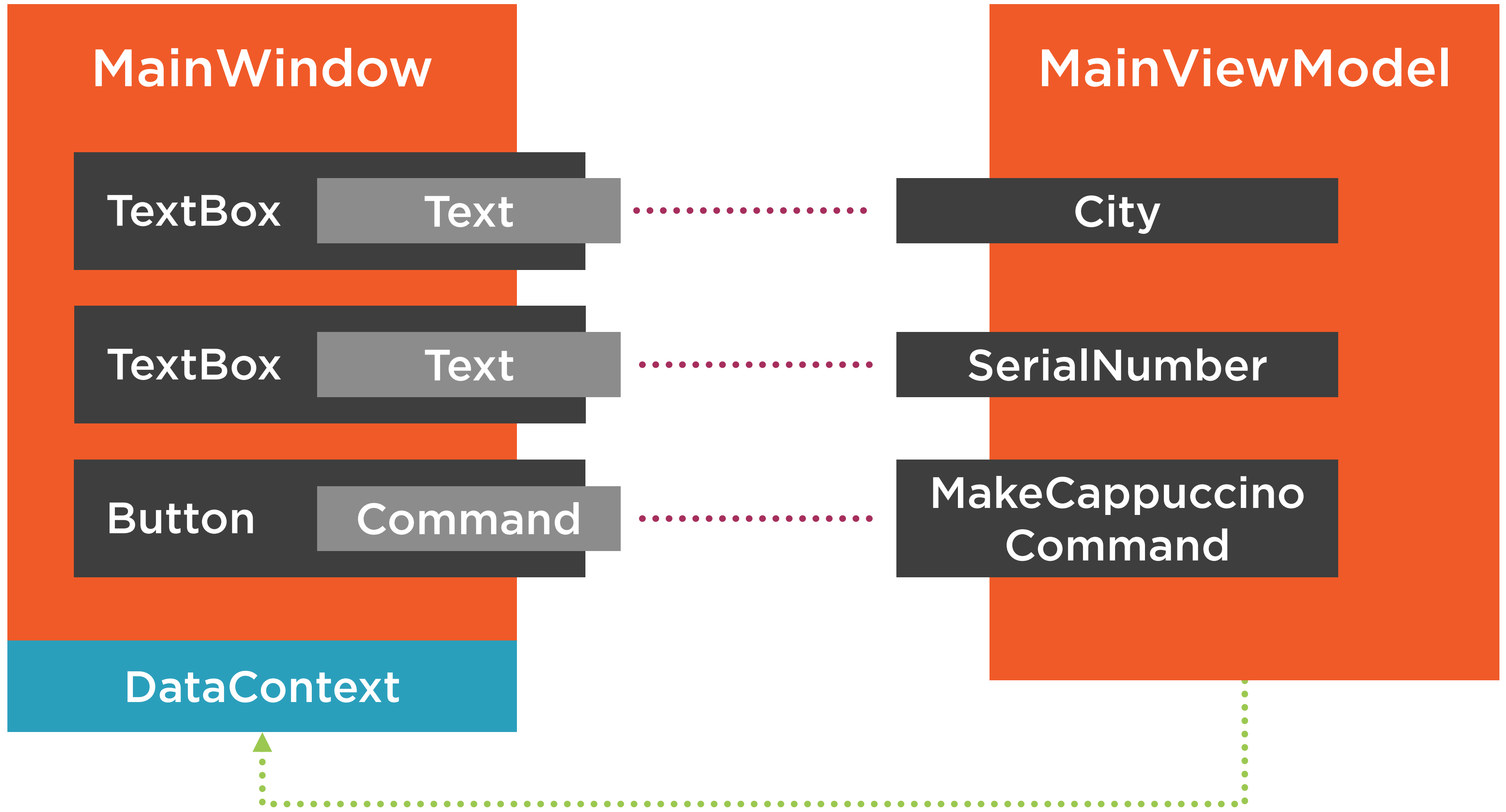
Set up a shared access policy

Demo

Create a new WPF project

Build the user interface

Implement the MainViewModel



Demo

Implement the MainViewModel

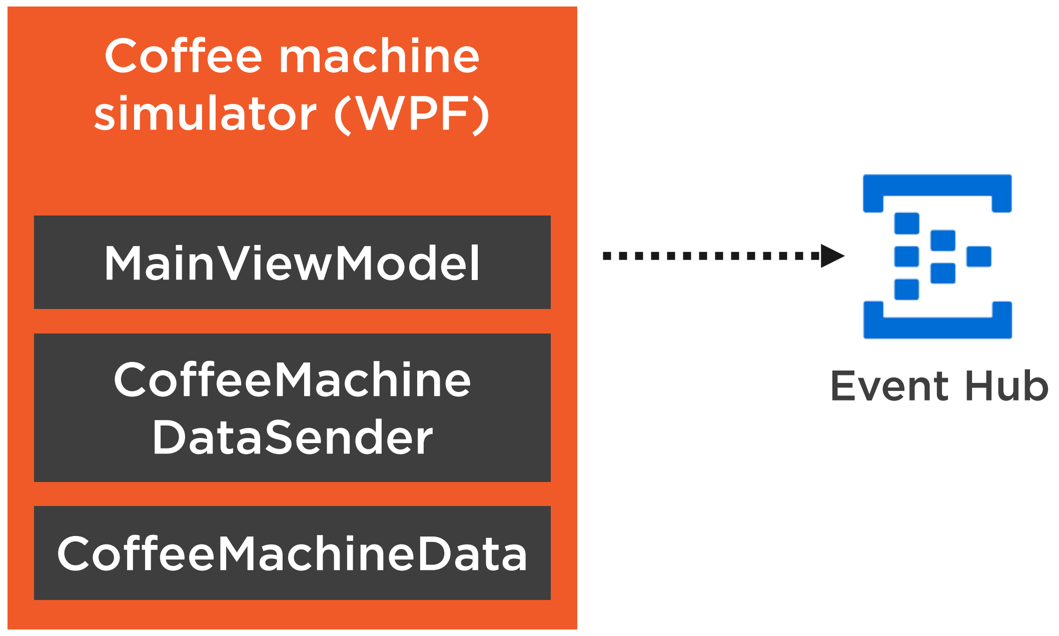
Bind the UI to the MainViewModel

Demo

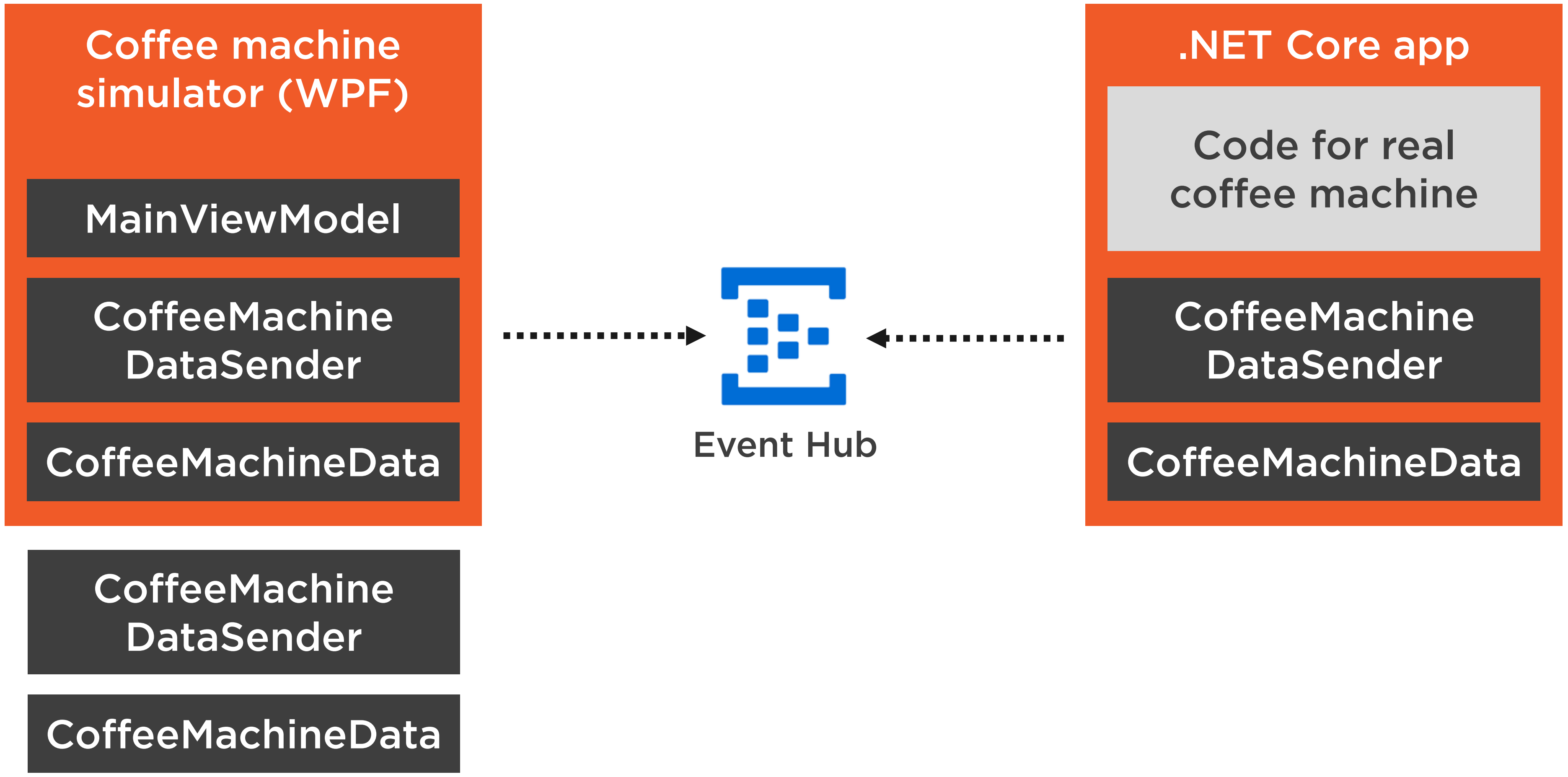
Add a CoffeeMachineData class

Create an instance
when a coffee is made

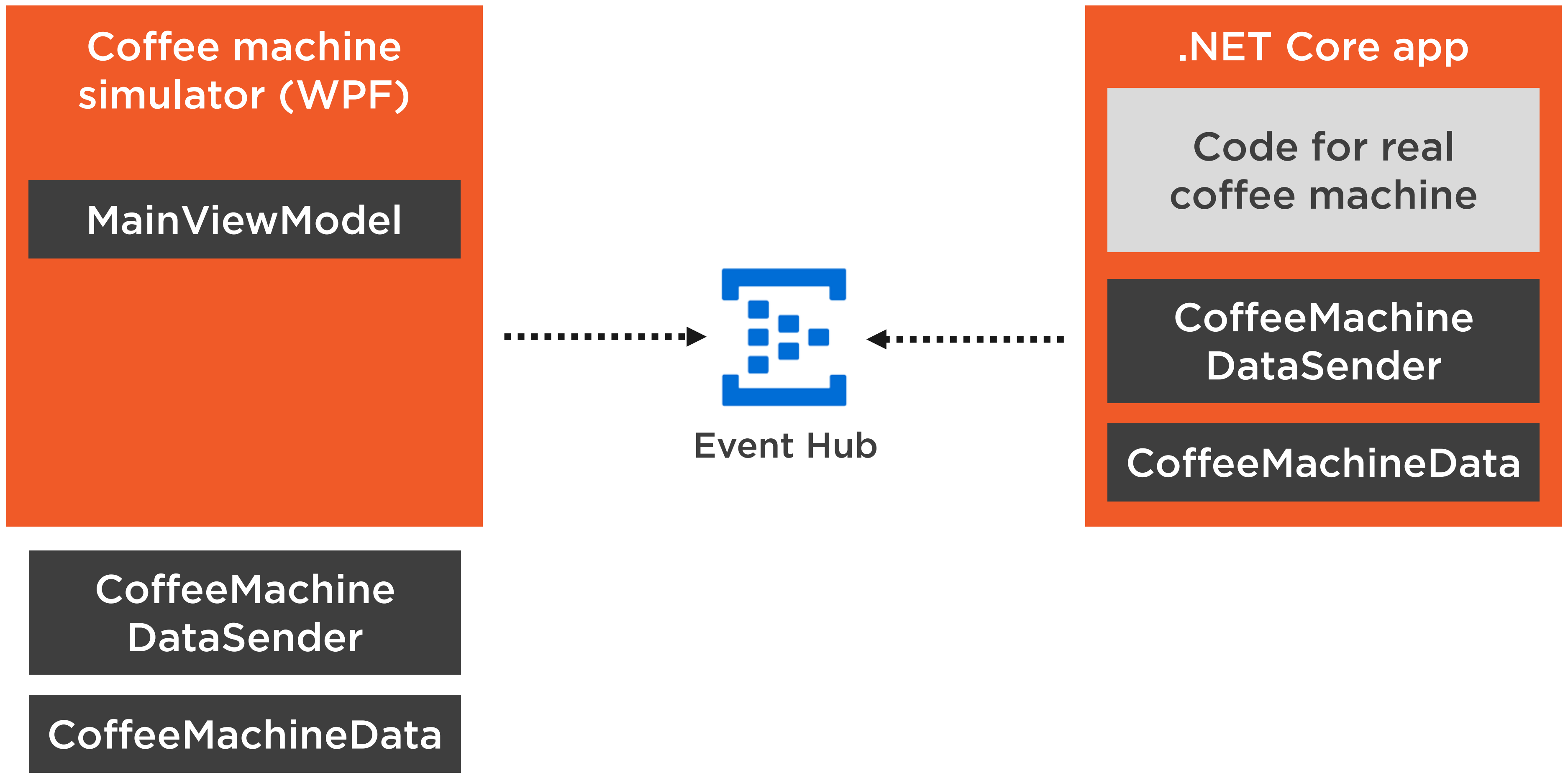
Add a .NET Standard Library to Send Events



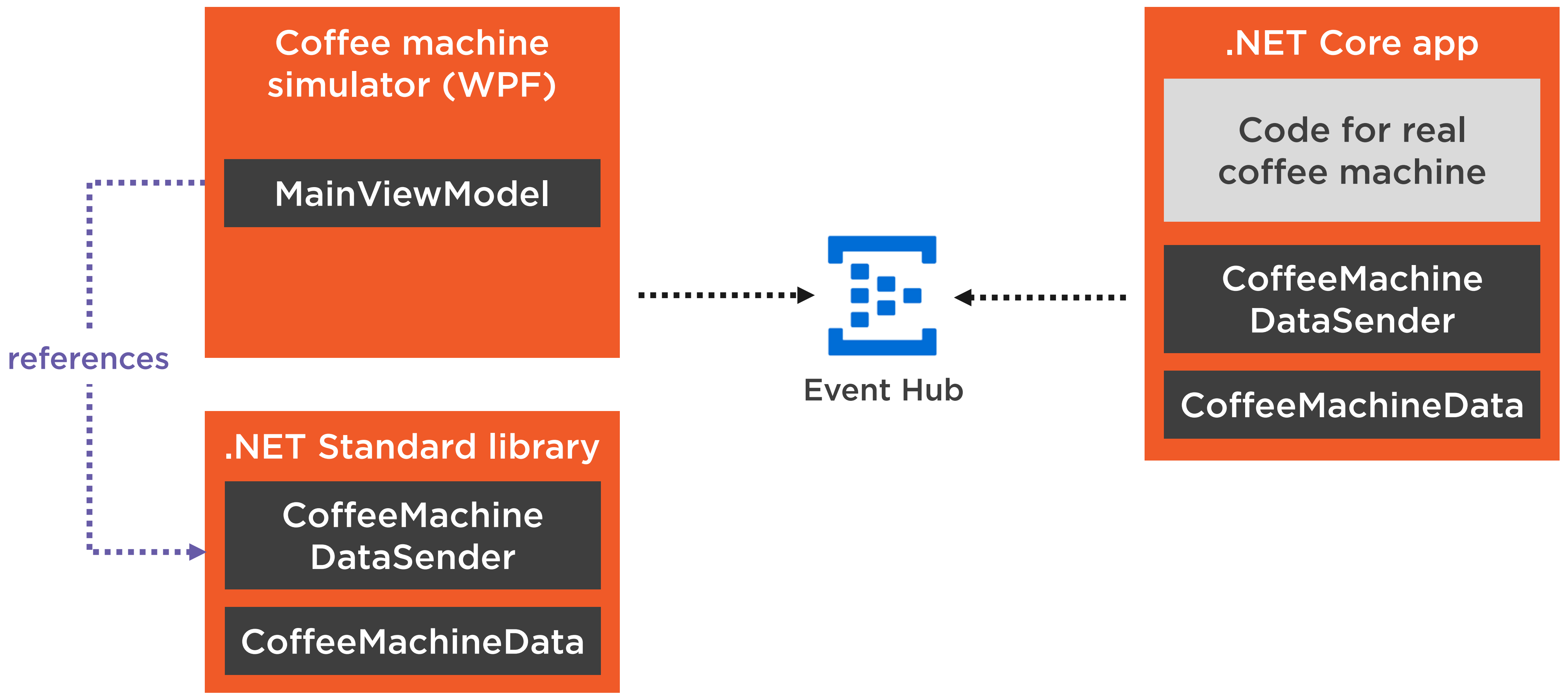
Add a .NET Standard Library to Send Events



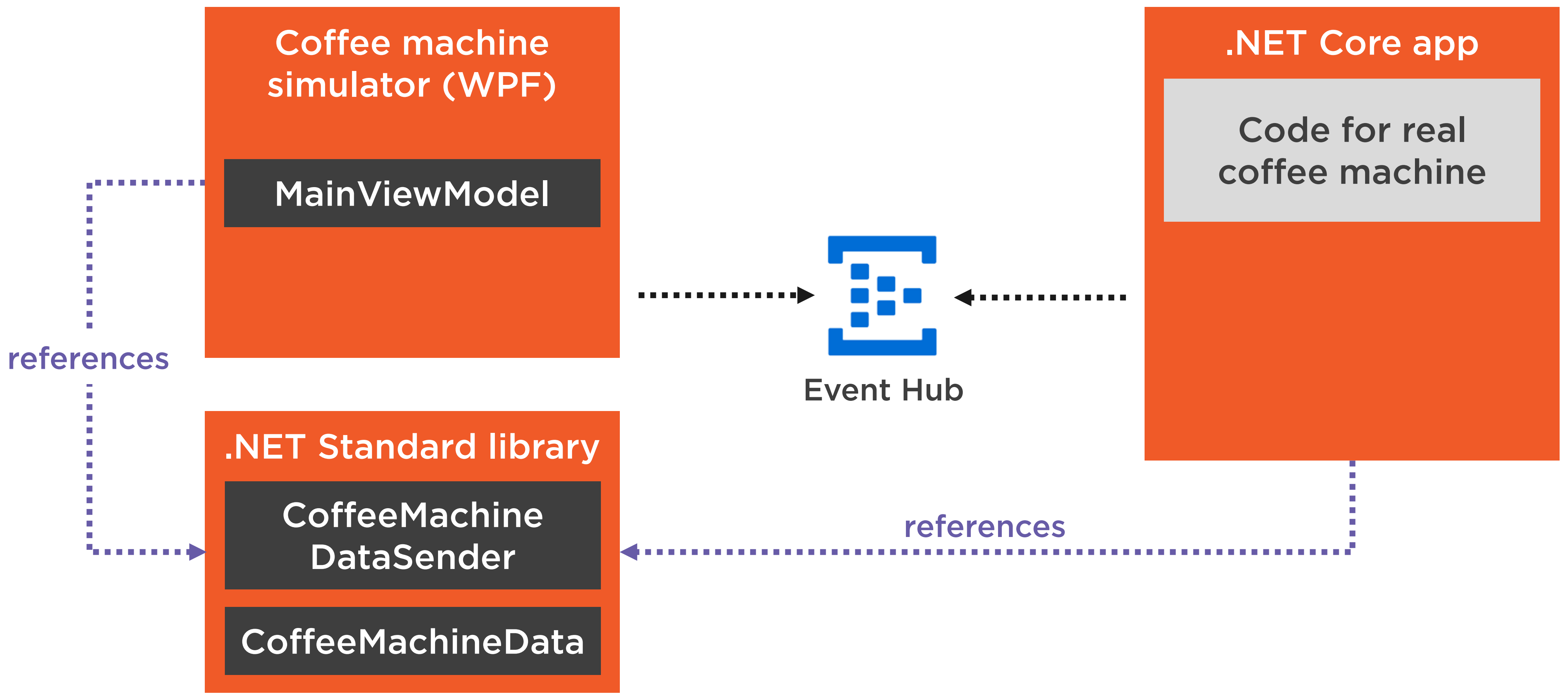
Add a .NET Standard Library to Send Events



Add a .NET Standard Library to Send Events



Add a .NET Standard Library to Send Events



Demo

Add a .NET Standard library

Move the CoffeeMachineData class to the library

Create and use a CoffeeMachineDataSender

Write the Code to Send Events

HTTPS

NuGet: Microsoft.Azure.EventHubs

Advanced Message Queueing Protocol (AMQP)

Demo

Implement the logic to send events in the CoffeeMachineDataSender class

Use Microsoft.Azure.EventHubs library

Set up a Shared Access Policy and Send Events

Shared access policy

Role-based access control

Demo

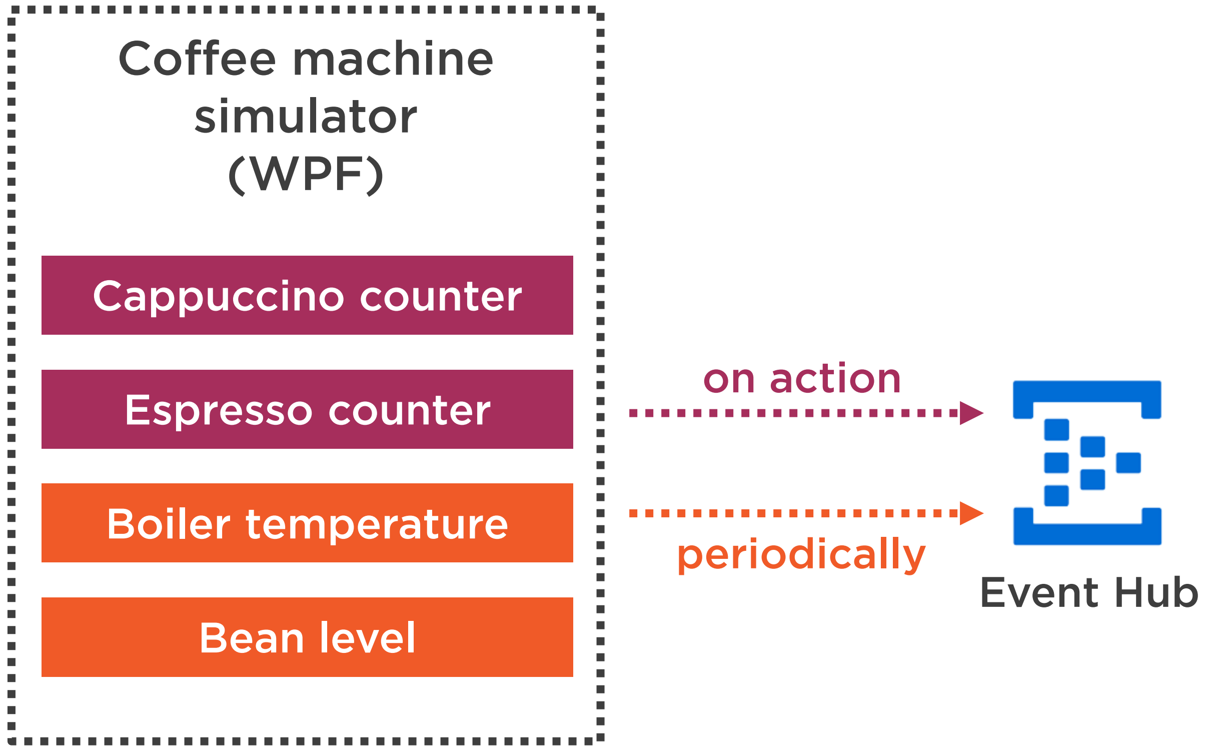
Create a shared access policy in the Azure portal

Grab the connection string and use it in the simulator app

Demo

Log sent events and exceptions in the user interface

Add and Bind Properties for Periodical Events



Demo

Add sensor properties to the MainViewModel

Bind the UI to the properties

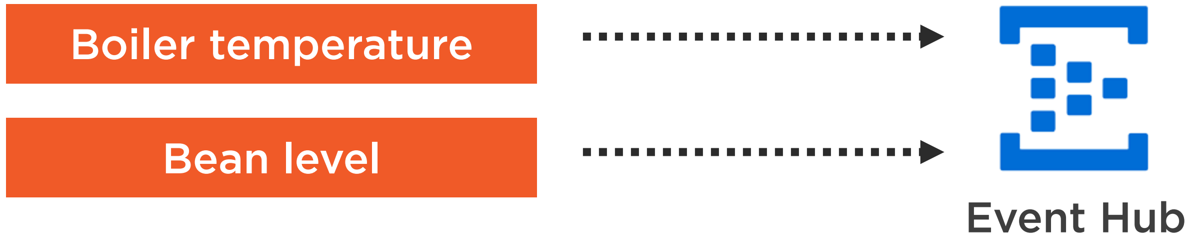
Send events periodically by using a DispatcherTimer

```
public abstract class EventHubClient
{
  public Task SendAsync(EventData eventData);




  ...
}
```

Batch Multiple Events

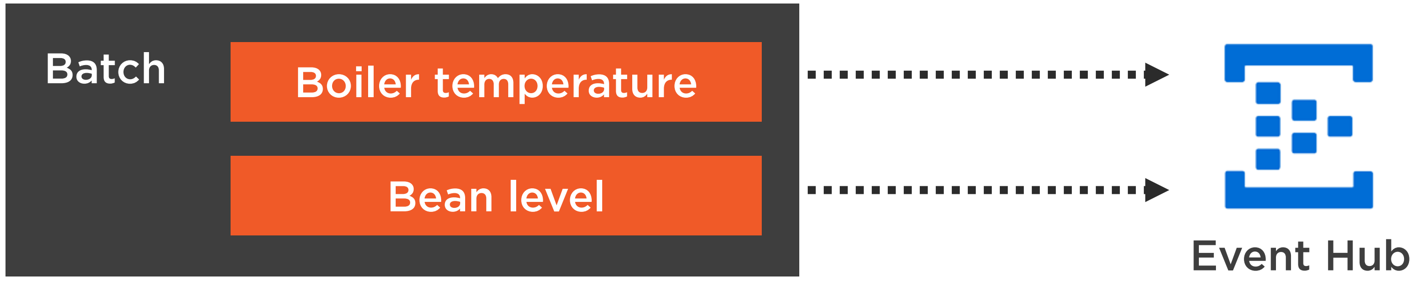


```
public abstract class EventHubClient
{
  public Task SendAsync(EventData eventData);



   ...
}
```

Batch Multiple Events

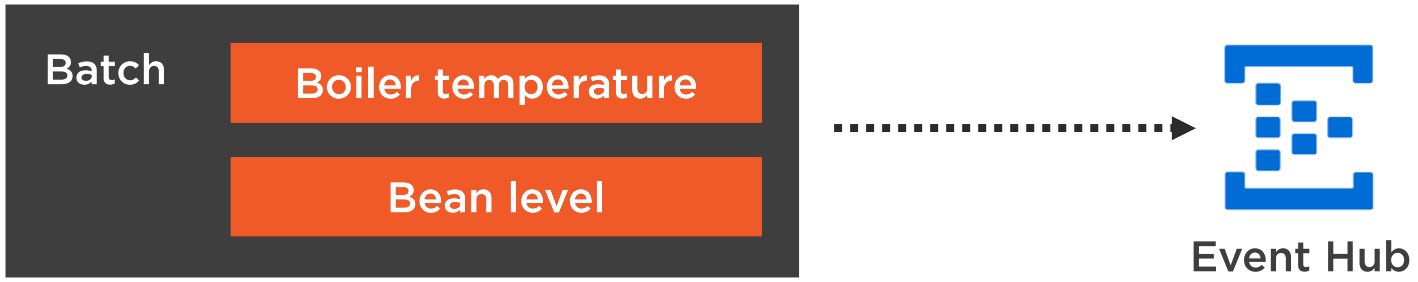


```
public abstract class EventHubClient
{
  public Task SendAsync(EventData eventData);




   ...
}
```

Batch Multiple Events

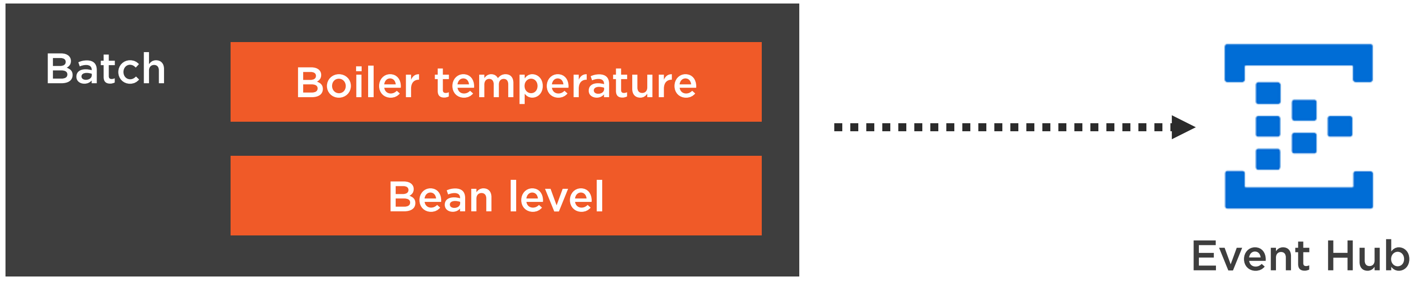


```
public abstract class EventHubClient
{
  public Task SendAsync(EventData eventData);

  public Task SendAsync(IEnumerable<EventData> eventDatas);


  ...
}
```

Batch Multiple Events

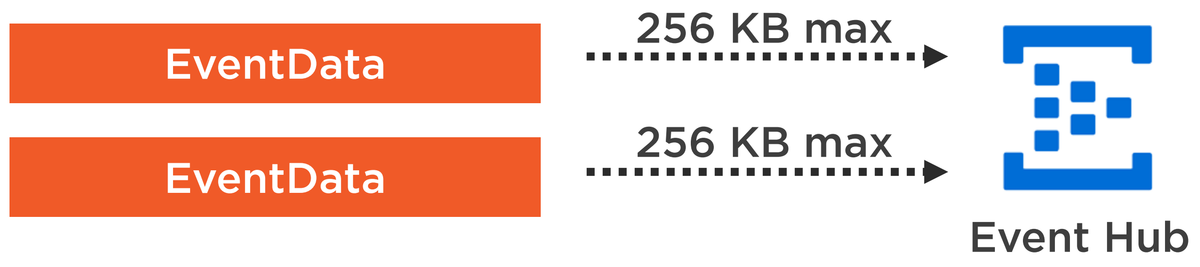


Demo

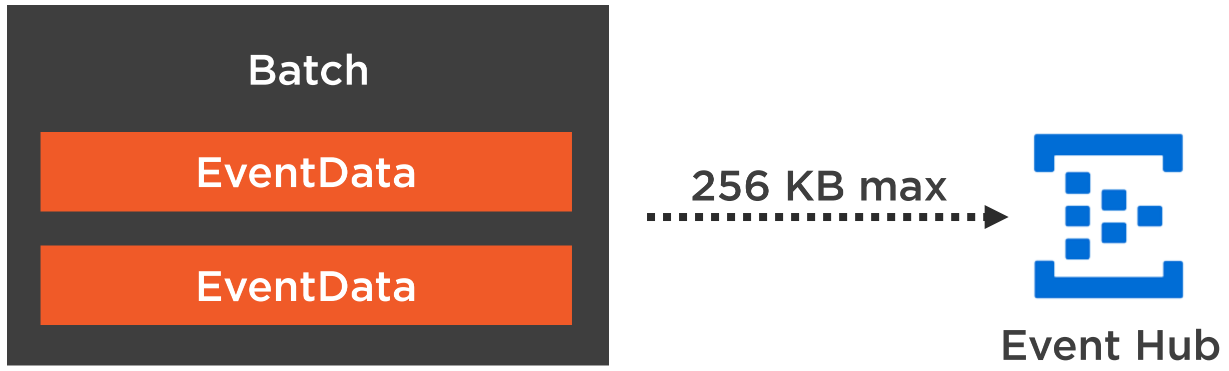
Extend the CoffeeMachineDataSender to support batching

Use the functionality in the MainViewModel

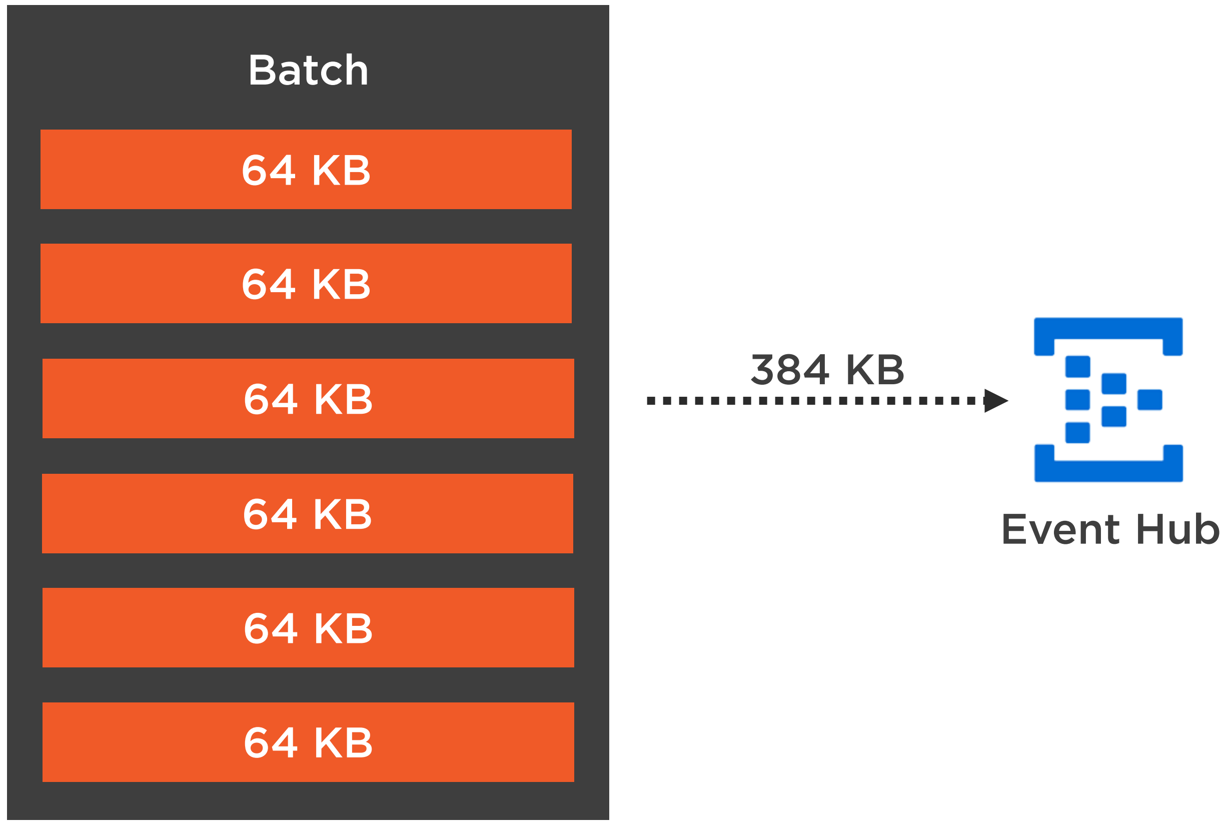
Keep the Message Size Limit When Batching



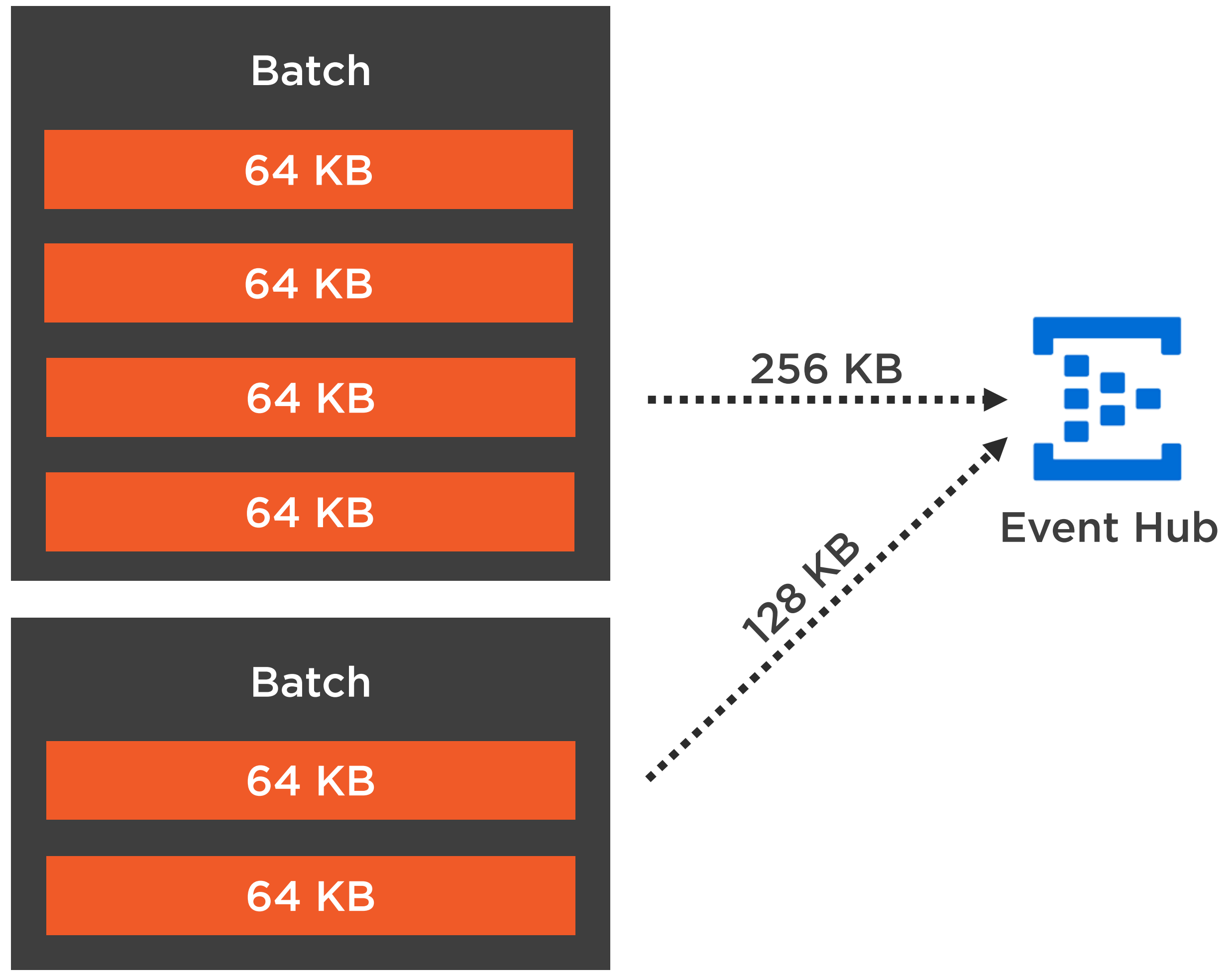
Keep the Message Size Limit When Batching



Keep the Message Size Limit When Batching



Keep the Message Size Limit When Batching



Demo

Use the EventDataBatch class to keep the message size limit

Summary

Build a coffee machine simulator with WPF

Use the Microsoft.Azure.EventHubs library

- EventHubClient to send events
- EventDataBatch to batch events

Set up a shared access policy