

Sending Messages in Your Application with IEventAggregator



Brian Lagunas

INFRAGISTICS - MICROSOFT MVP

@brianlagunas | <https://brianlagunas.com>



Agenda



What is the IEventAggregator?

Using the IEventAggregator

Filtering Events

Unsubscribing from an Event

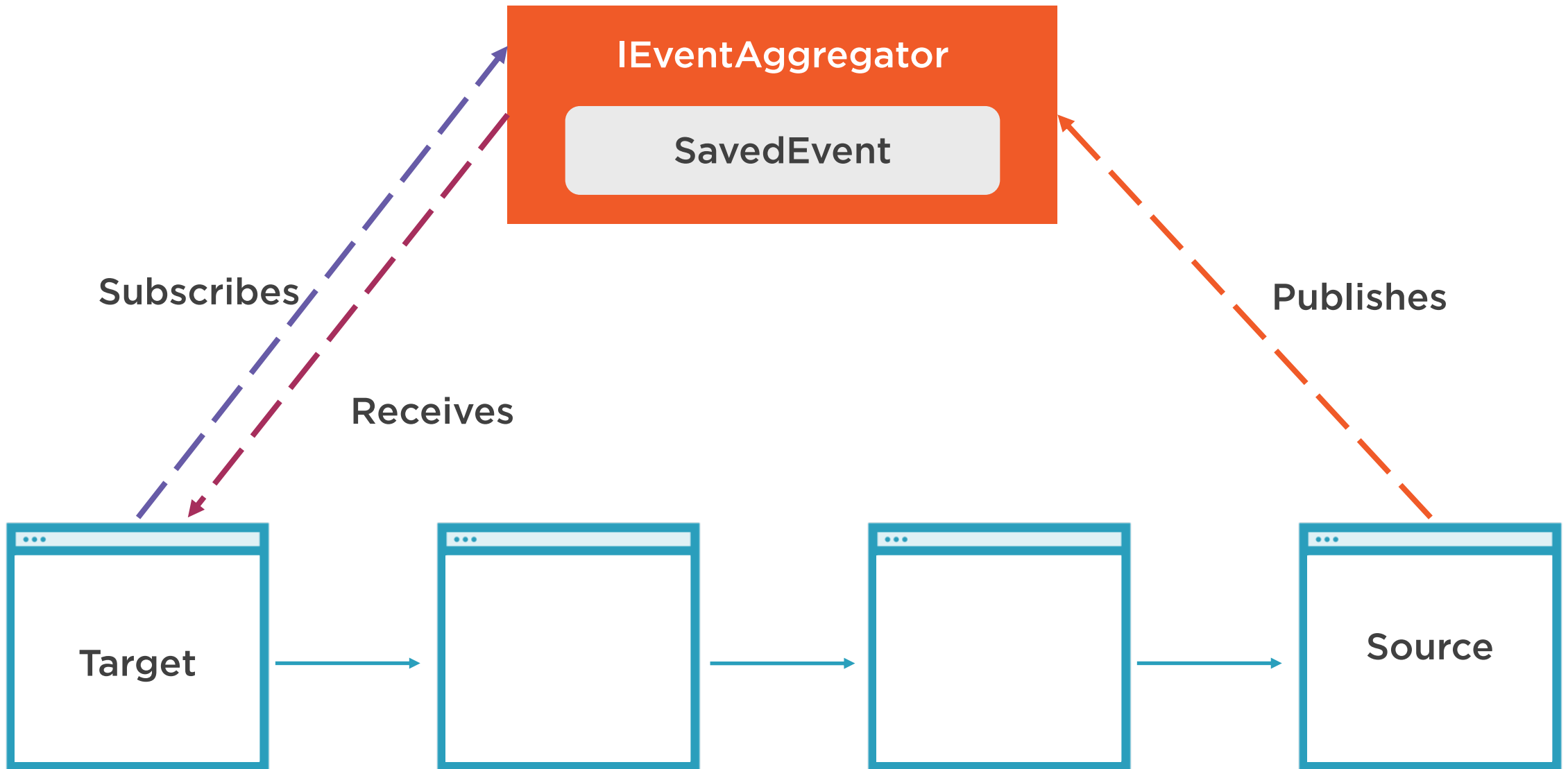


IEventAggregator

**Loosely coupled event-based
communication**

Multiple Publishers and Subscribers





IEventAggregator

Loosely coupled event-based communication

Multiple Publishers and Subscribers

Pass parameters/payloads

Filter events

They're weak events

- keepSubscriberReferenceAlive – creates a strong reference
- Unsubscribe from events



Using the IEventAggregator

//event

```
public class SavedEvent : PubSubEvent<string> { }
```

//publisher

```
IEventAggregator.GetEvent<SavedEvent>().Publish("some value");
```

//subscriber

```
IEventAggregator.GetEvent<SavedEvent>().Subscribe(SomethingSaved);
```

//delegate

```
void SomethingSaved(string parameter)
```

```
{
```

```
    //do something
```

```
}
```



Demo



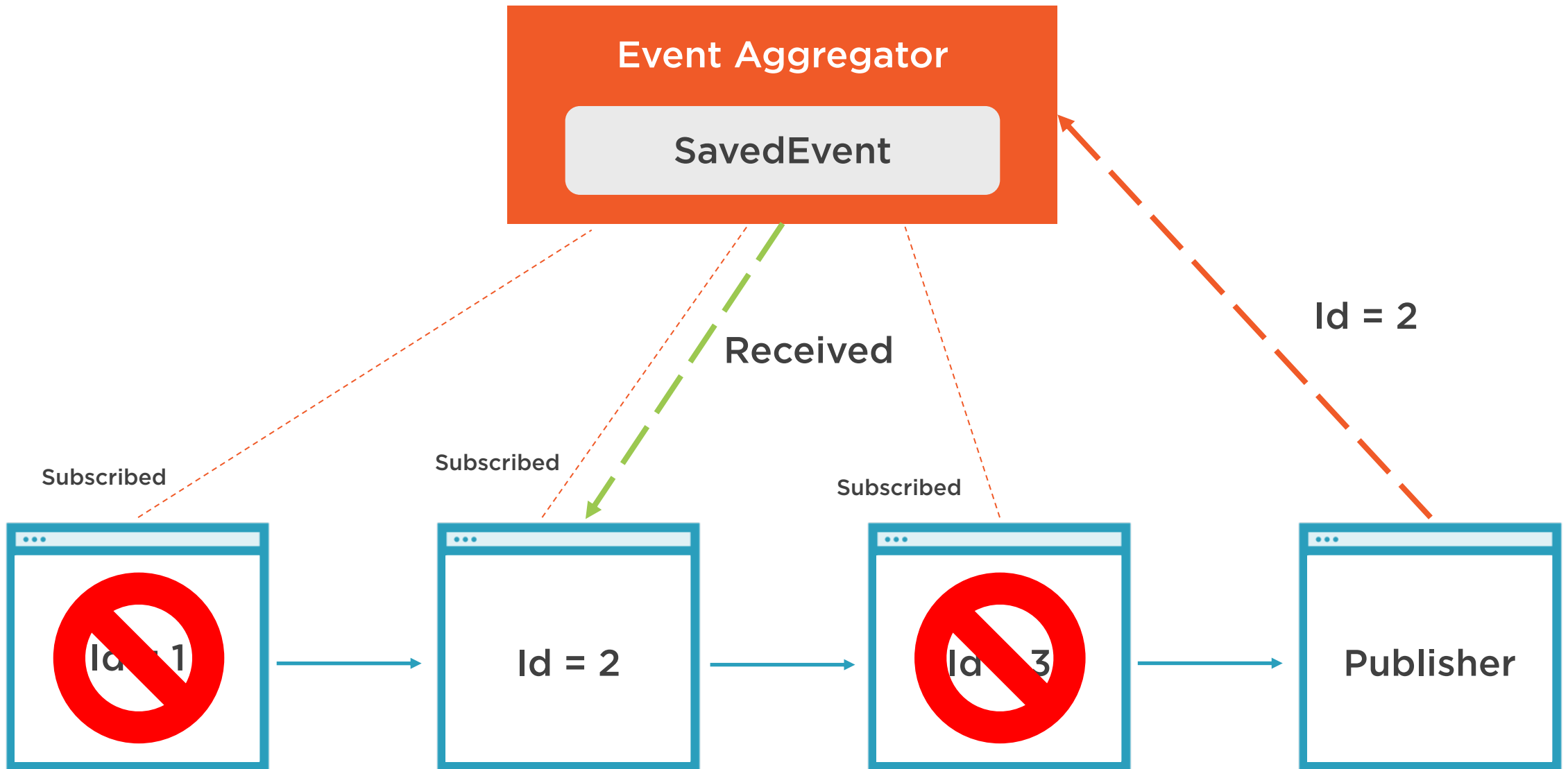
Using the IEventAggregator



Filtering Events

Subscribers may not need to handle every instance of a published event





Filtering Events

```
IEventAggregator.GetEvent<SavedEvent>().Subscribe(SomethingSaved,
```

```
void SomethingSaved(string parameter)  
{  
    //do something  
}
```



Filtering Events

```
IEventAggregator.GetEvent<SavedEvent>().Subscribe(SomethingSaved,  
    ThreadOption.Publisher, false,  
    p => p == "Brian");
```

```
void SomethingSaved(string parameter)  
{  
    //do something  
}
```



Demo



Filtering events



Unsubscribe

No longer receive events

`keepSubscriberReferenceAlive = true`

Unsubscribe

- Delegate method
- Subscription token



Unsubscribe with Delegate Method

```
var event = IEventAggregator.GetEvent<SavedEvent>();
```

```
event.Subscribe(SavedMethod);
```

```
//unsubscribe
```

```
event.Unsubscribe(SavedMethod);
```



Unsubscribe with Token

```
var event = IEventAggregator.GetEvent<SavedEvent>();
```

```
//get the token
```

```
var token = event.Subscribe(SavedEventHandler);
```

```
//unsubscribe
```

```
event.Unsubscribe(token);
```



Demo



Unsubscribing from events



Summary



What is the IEventAggregator?

Using the IEventAggregator

Filtering Events

Unsubscribing from an Event

