LIGTHWEIGHT MICROSERVICE COLLABORATION USING HTTP

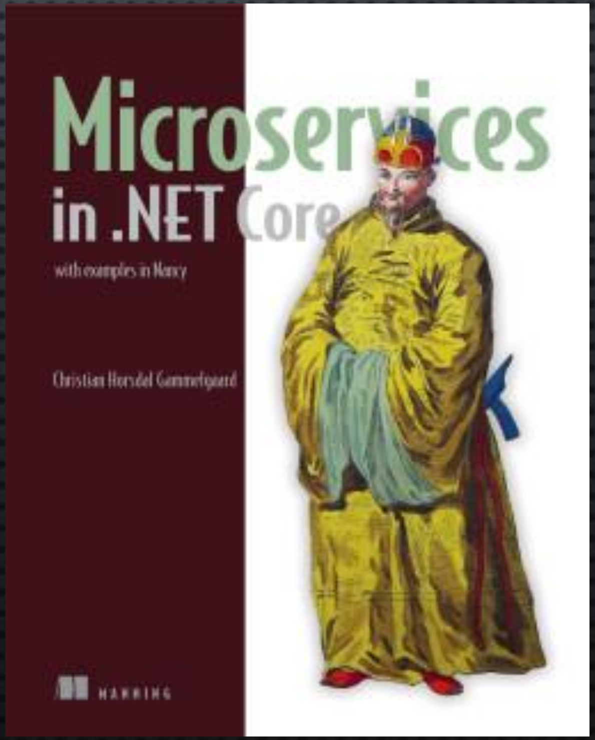
CHRISTIAN HORSDAL

@CHR_HORSDAL

ABOUT ME

Christian Horsdal
Independent Consultant

www.horsdal-consult.dk
c.horsdal@gmail.com
@chr_horsdal



NANCY
MOST VALUED
MINION

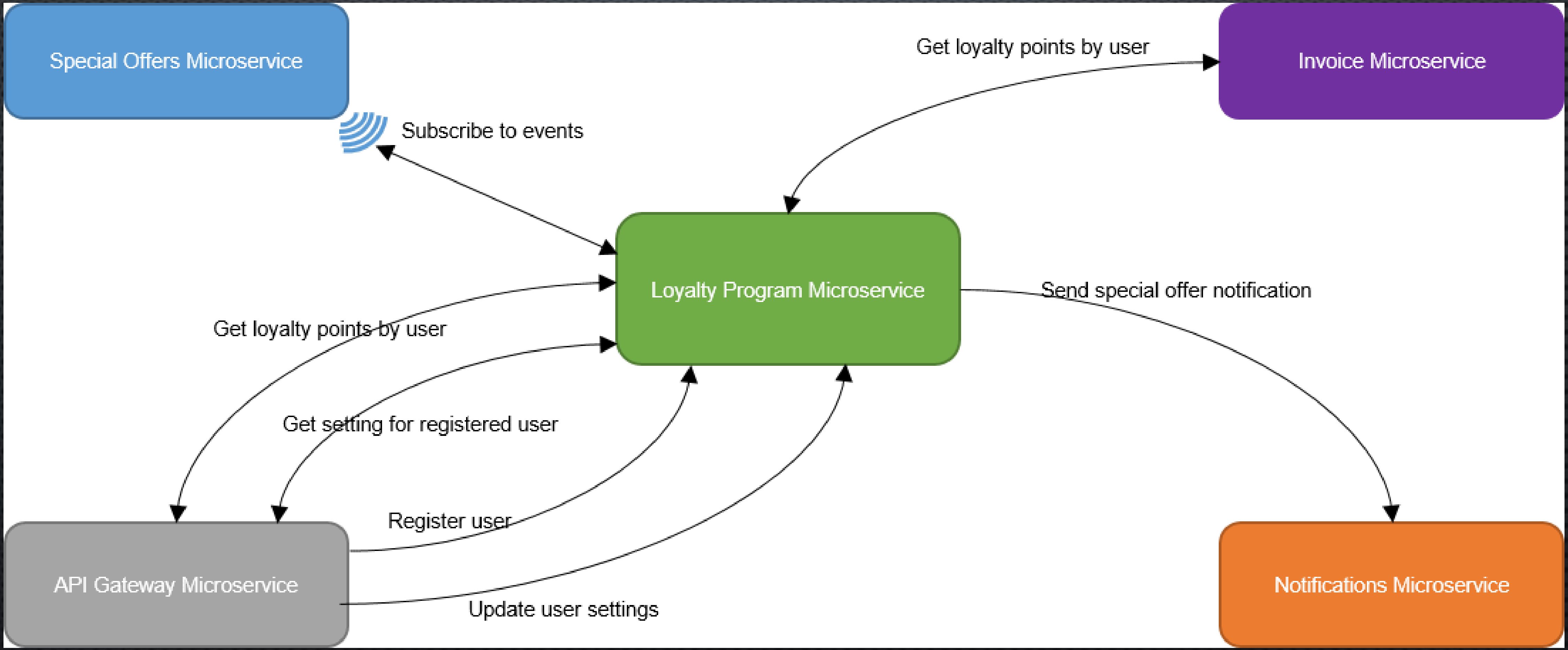
MVP
Microsoft®
Most Valuable
Professional

MICROSERVICES

INTRODUCTION TO MICROSERVICES

A *MICROSERVICE* IS A SERVICE WITH ONE, AND ONLY ONE, VERY NARROWLY FOCUSED CAPABILITY THAT A REMOTE API EXPOSES TO THE REST OF THE SYSTEM.

INTRODUCTION TO MICROSERVICES


Special Offers Microservice
Subscribe to events
Get loyalty points by user
Invoice Microservice
Loyalty Program Microservice
Send special offer notification
Get loyalty points by user
Get setting for registered user
Register user
Update user settings
API Gateway Microservice
Notifications Microservice

WHY MICROSERVICES?

- Enables continuous delivery
- Highly maintainable services
- Efficient developer workflow
- Robust by design
- Flexible scalability

WHY NOT?

- Distributed systems are hard
- 100s or more services
  - Complex production environment
  - Lots of deployments
- Refactoring across services is cumbersome

WHEN CAN YOU USE MICROSERVICE?

- Automated deploys
- Fast provisioning
- Automated builds
- Automated tests
- Logging and monitoring
- Close dev-ops relationship

THE 6 CHARACTERISTICS OF A MICROSERVICE

1. Is responsible for a single capability.
2. Is individually deployable.
3. Consists of one or more processes.
4. Owns its own data store.
5. A small team can maintain a handful of microservices.
6. Is replaceable.

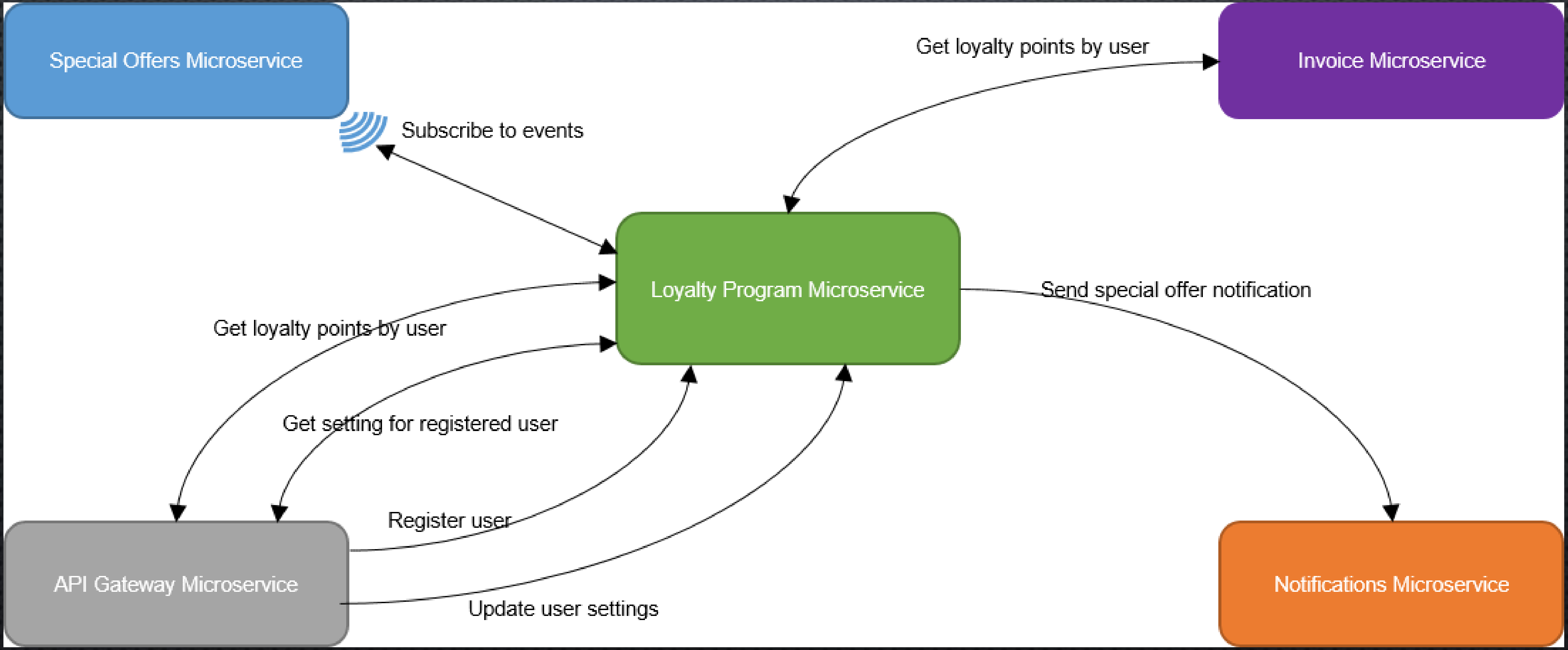
MICROSERVICE COLLABORATION

COLLABORATION IS ANY AND ALL COMMUNICATION BETWEEN MICROSERVICES

Collaboration

- Commands
    - DoStuff
- Queries
    - ReadStuff
- Events
    - StuffHappened

EXAMPLE

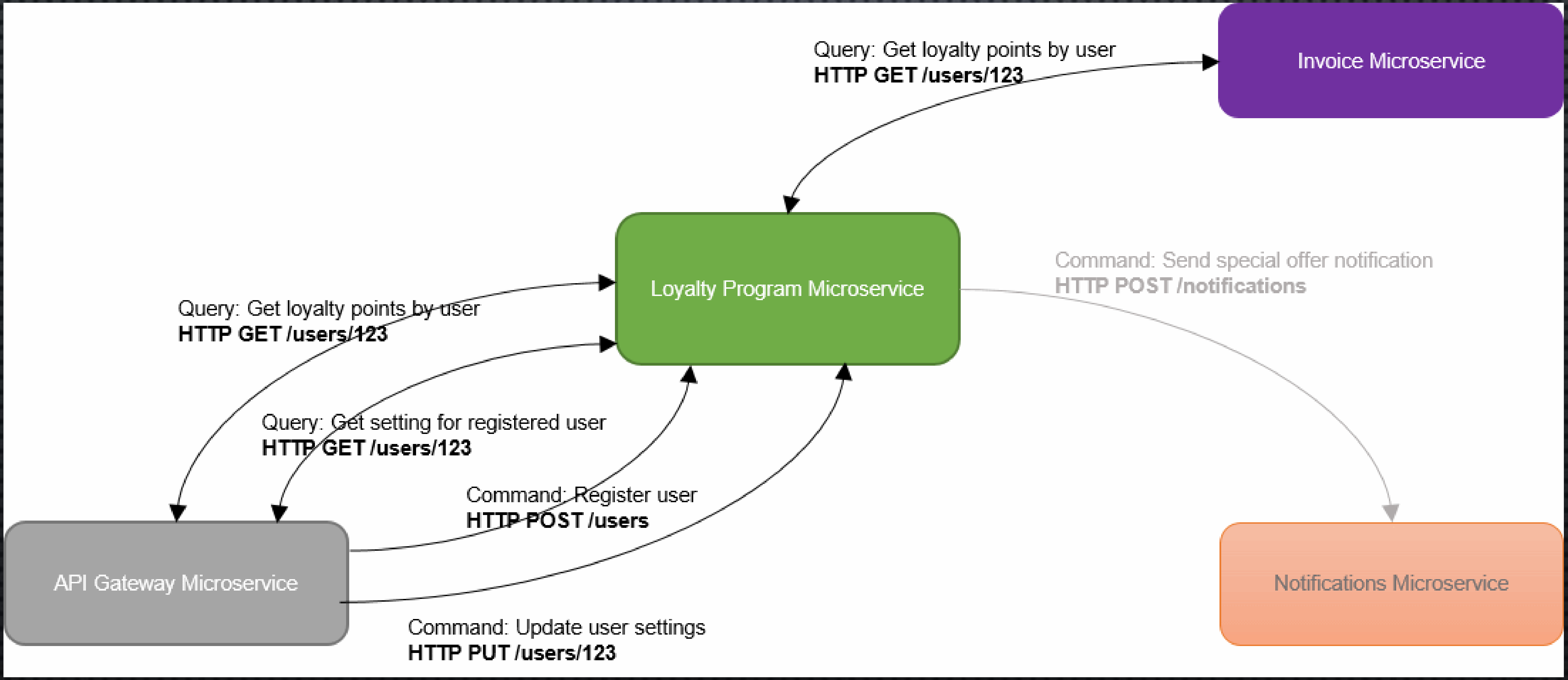

Special Offers Microservice
Invoice Microservice
Get loyalty points by user
Subscribe to events
Loyalty Program Microservice
Send special offer notification
Get loyalty points by user
Get setting for registered user
Register user
API Gateway Microservice
Update user settings
Notifications Microservice

COMMANDS AND QUERIES

COMMANDS AND QUERIES

- Synchronous
- Queries: `HTTP GET`
- Commands: `HTTP POST, HTTP PUT`

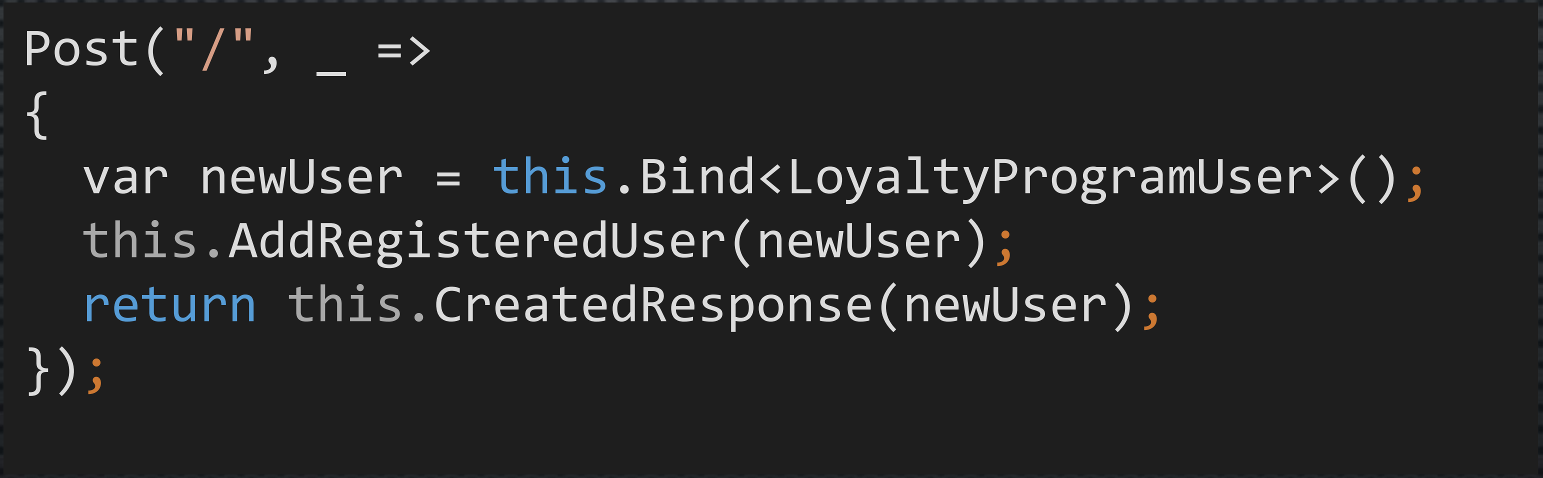
EXAMPLE: COMMANDS, QUERIES


Query: Get loyalty points by user
HTTP GET /users/123
Invoice Microservice
Loyalty Program Microservice
Command: Send special offer notification
HTTP POST /notifications
Query: Get loyalty points by user
HTTP GET /users/123
Query: Get setting for registered user
HTTP GET /users/123
Command: Register user
HTTP POST /users
API Gateway Microservice
Notifications Microservice
Command: Update user settings
HTTP PUT /users/123

EXAMPLE: QUERY

```
Get("/{userId:int}", parameters =>
{
  int userId = parameters.userId;
  if (registeredUsers.ContainsKey(userId))
    return registeredUsers[userId];
  else
    return HttpStatusCode.NotFound;
});
```

EXAMPLE: COMMAND



```
Post("/", _ =>
{
  var newUser = this.Bind<LoyaltyProgramUser>();
  this.AddRegisteredUser(newUser);
  return this.CreatedResponse(newUser);
});
```

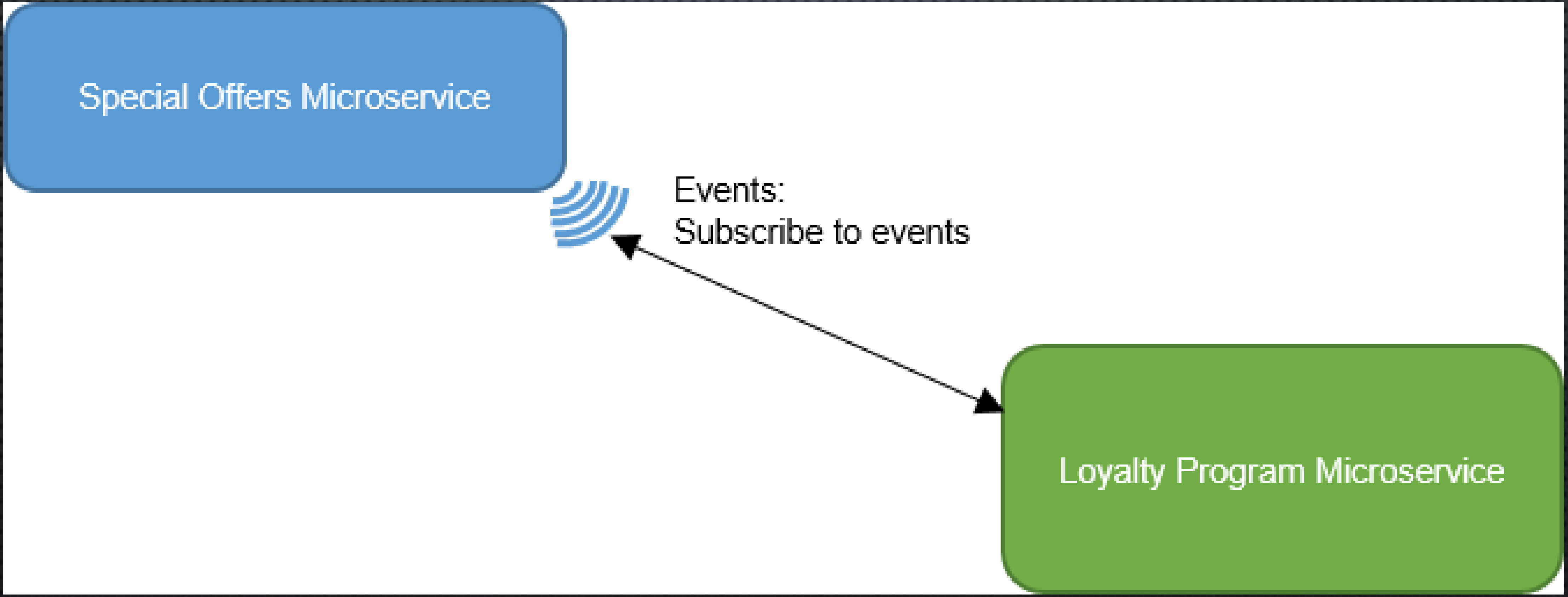
DEMO

EVenT

EVENTS

- Asynchronous
- "Stuff happened"
  - Past tense
- Publish/Subscribe

EXAMPLE: EVENTS



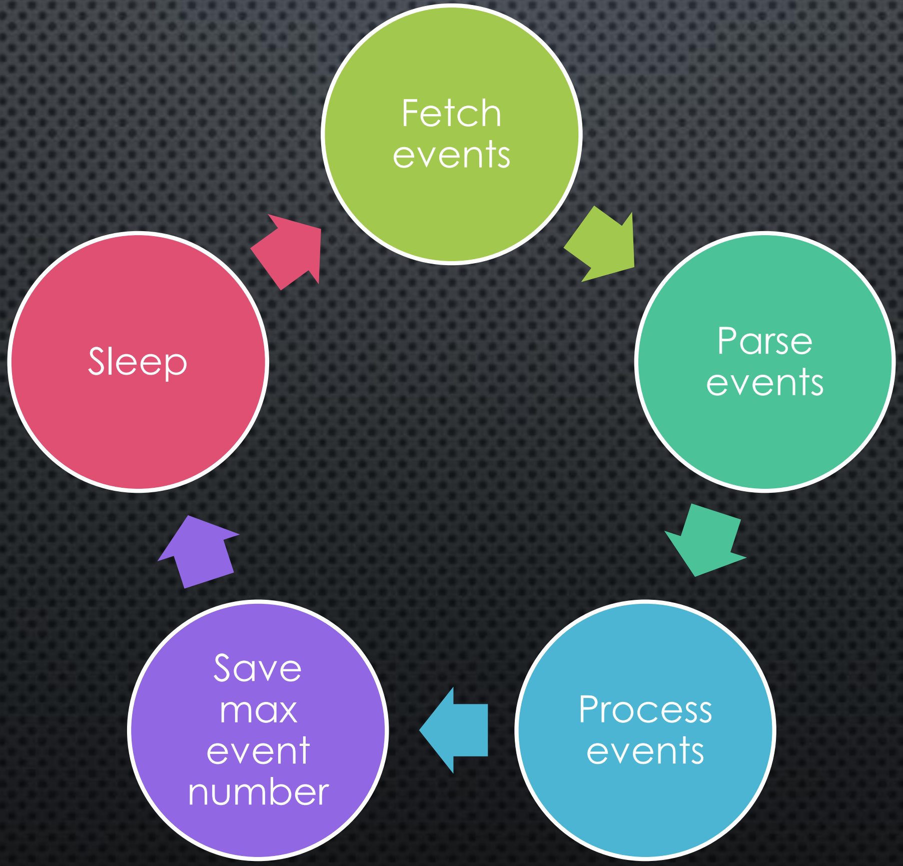
EVENT FEED

- An HTTP GET endpoint
  - Supports paging
  - HTTP GET /events?from=100&to=200
- Events raised by domain logic
- Events are stored in an event store

EXAMPLE: EVENT FEED

```
Get("/", _ =>
{
  long firstEventSequenceNumber, lastEventSequenceNumber;
  if (!long.TryParse(this.Request.Query.from.Value,
                     out firstEventSequenceNumber))
    firstEventSequenceNumber = 0;
  if (!long.TryParse(this.Request.Query.to.Value,
                     out lastEventSequenceNumber))
    lastEventSequenceNumber = long.MaxValue;

  return eventStore.GetEvents(firstEventSequenceNumber,
                              lastEventSequenceNumber);
});
```

SUBSCRIBING TO EVENTS



Subscribing to events

- Part of a microservice
  - But separate process

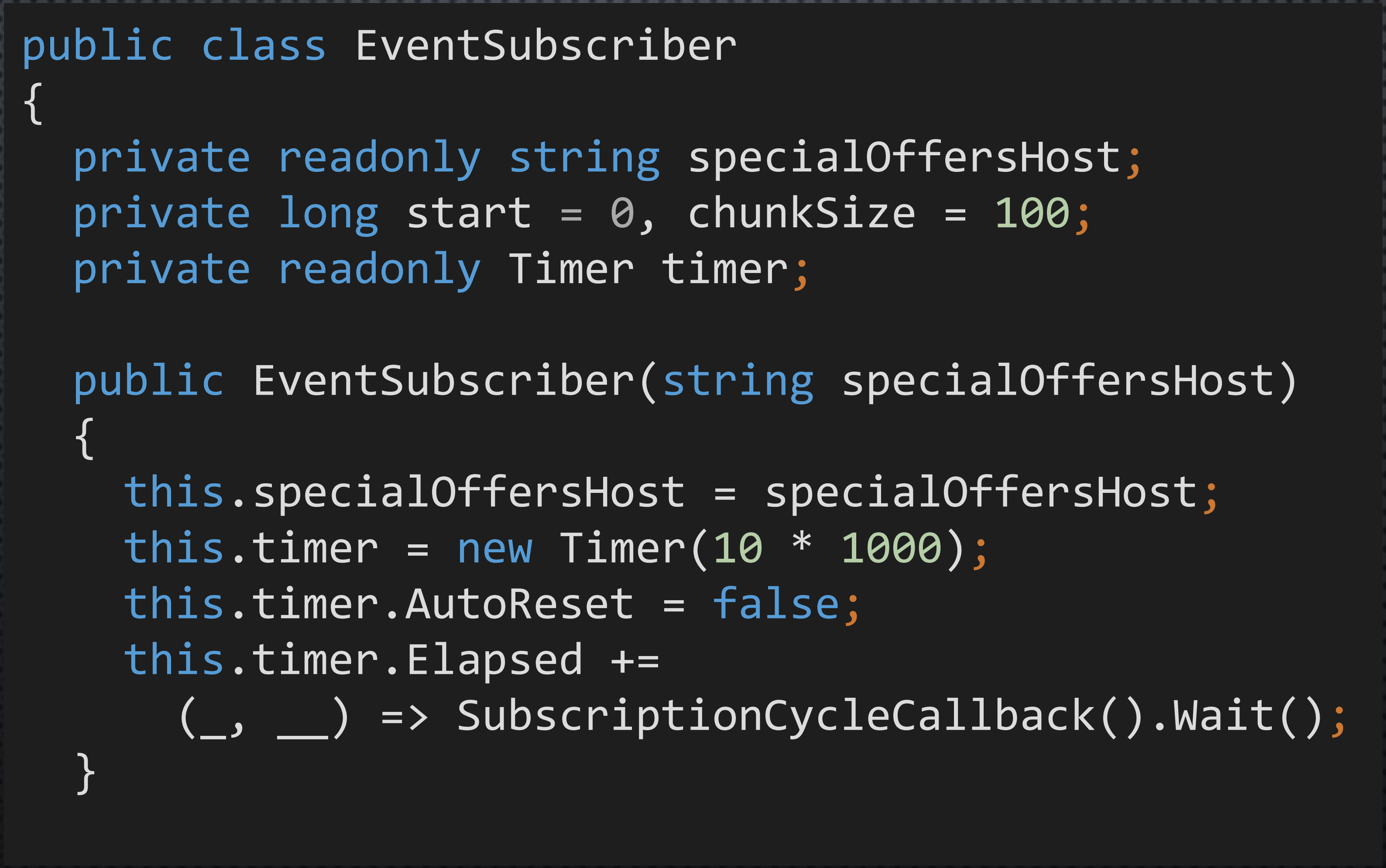
- A windows service
- A deamon

THE 6 CHARACTERISTICS

A MICROSERVICE

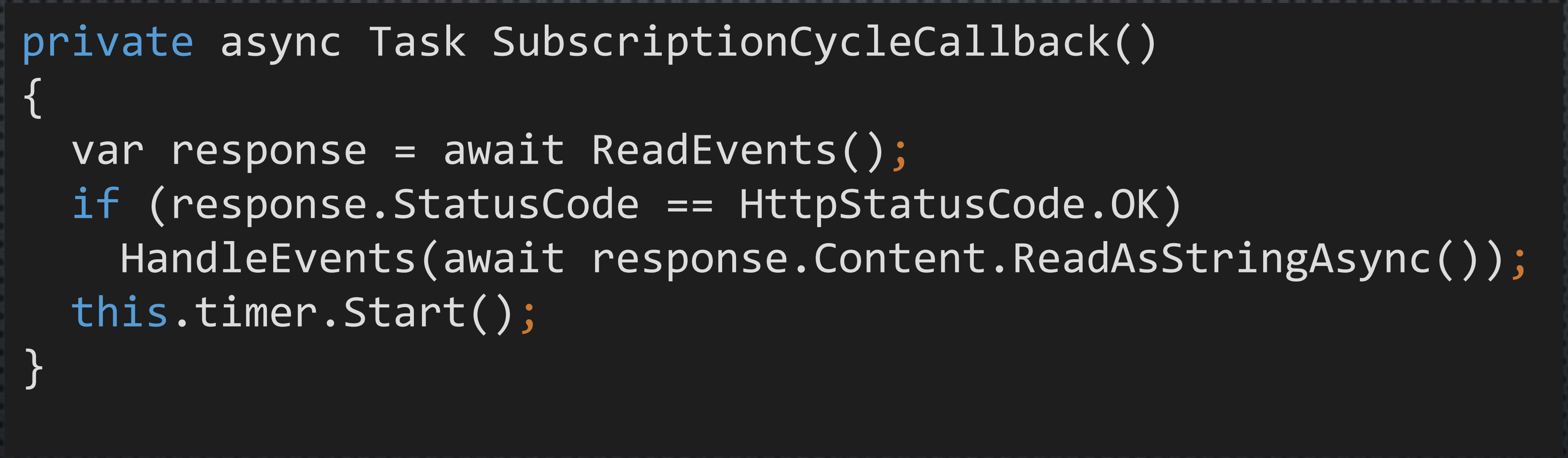
1. IS RESPONSIBLE FOR A SINGLE CAPABILITY.
2. **IS INDIVIDUALLY DEPLOYABLE.**
3. **CONSISTS OF ONE OR MORE PROCESSES.**
4. OWNS ITS OWN DATA STORE.
5. A SMALL TEAM CAN MAINTAIN A HANDFUL OF MICROSERVICES.
6. IS REPLACEABLE.

EXAMPLE: SUBSCRIBING TO EVENTS



```
public class EventSubscriber
{
  private readonly string specialOffersHost;
  private long start = 0, chunkSize = 100;
  private readonly Timer timer;

  public EventSubscriber(string specialOffersHost)
  {
    this.specialOffersHost = specialOffersHost;
    this.timer = new Timer(10 * 1000);
    this.timer.AutoReset = false;
    this.timer.Elapsed +=
      (_, __) => SubscriptionCycleCallback().Wait();
  }
```

EXAMPLE: SUBSCRIBING TO EVENTS



```
private async Task SubscriptionCycleCallback()
{
  var response = await ReadEvents();
  if (response.StatusCode == HttpStatusCode.OK)
    HandleEvents(await response.Content.ReadAsStringAsync());
  this.timer.Start();
}
```

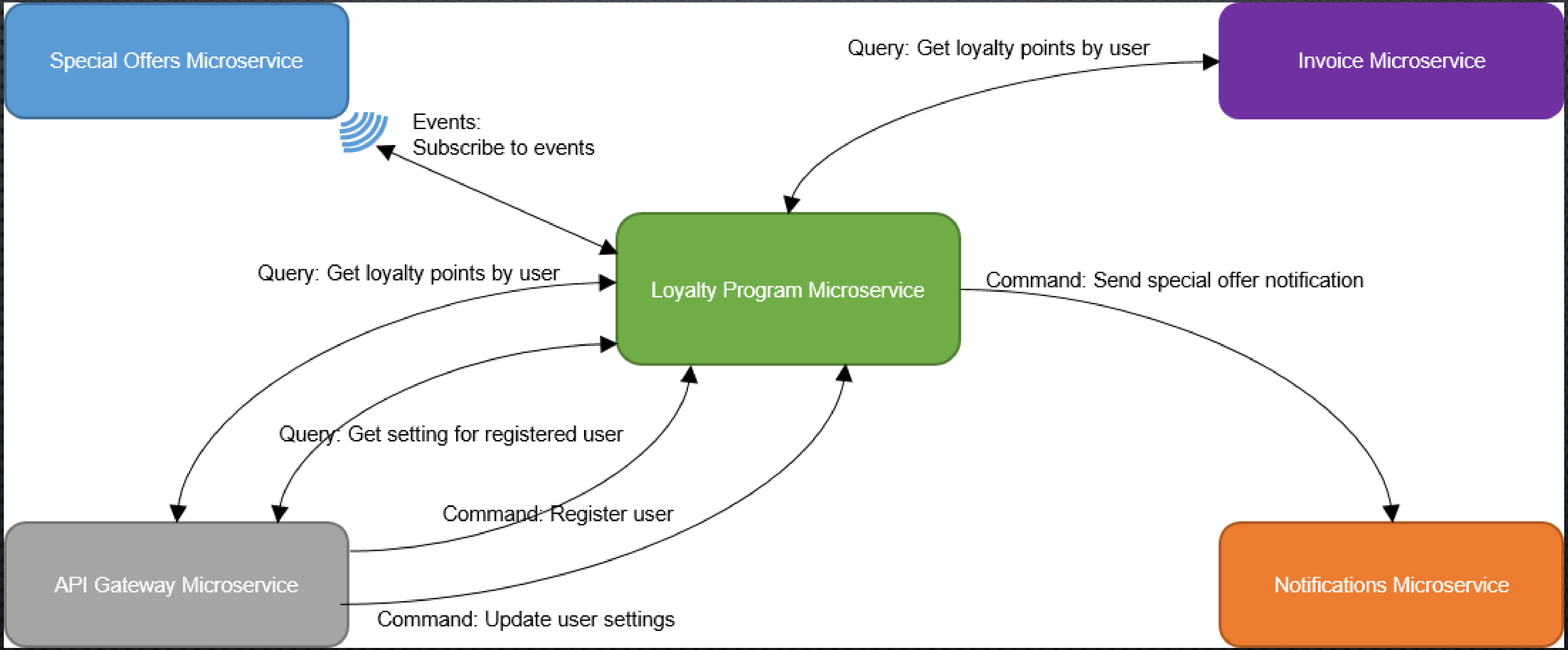
EXAMPLE: SUBSCRIBING TO EVENTS

```
private async Task<HttpResponseMessage> ReadEvents()
{
  using (var httpClient = new HttpClient())
  {
    httpClient.BaseAddress =
      new Uri($"http://{this.specialOffersHost}");
    var response = await httpClient.GetAsync(
      $"/events/?start={this.start}&end={this.start + this.chunkSize}");
    return response;
  }
}
```

EXAMPLE: SUBSCRIBING TO EVENTS

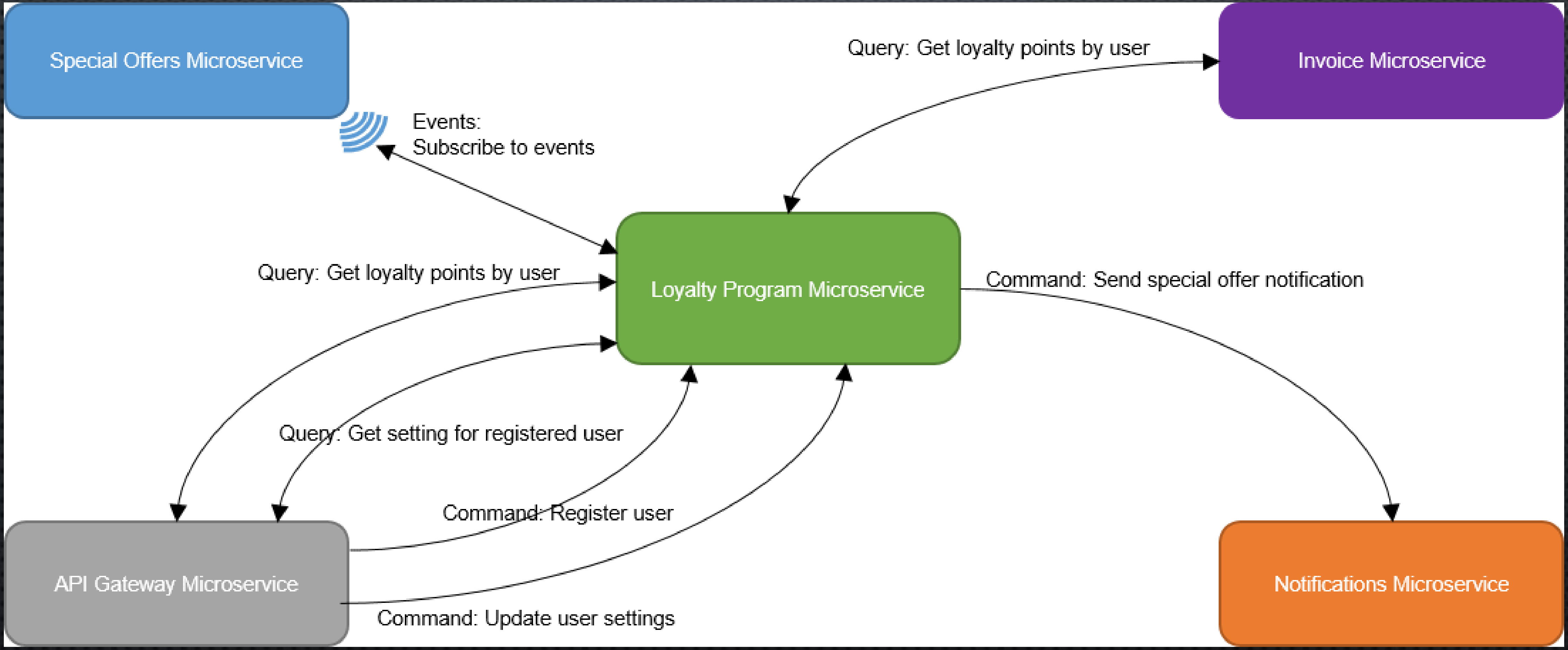
```
private void HandleEvents(string content)
{
  var events =
    JsonConvert.DeserializeObject<IEnumerable<SpecialOfferEvent>>(content);

  foreach (var ev in events)
  {
     // Inspect ev
     // Invoke domain logic
  }
}
```

EXAMPLE RECAP


Special Offers Microservice
Invoice Microservice
Query: Get loyalty points by user
Events:
Subscribe to events
Loyalty Program Microservice
Query: Get loyalty points by user
Command: Send special offer notification
Query: Get setting for registered user
Command: Register user
API Gateway Microservice
Notifications Microservice
Command: Update user settings

DEMO

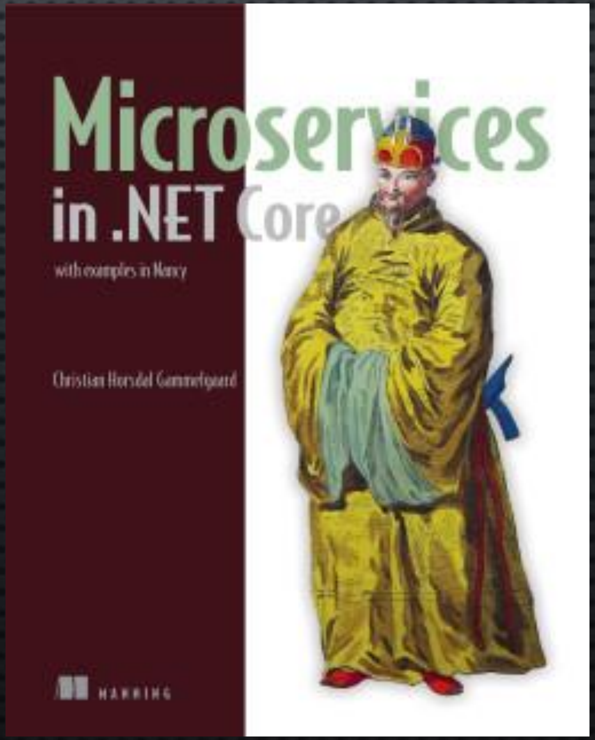
EXAMPLE RECAP


Special Offers Microservice
Events:
Subscribe to events
Query: Get loyalty points by user
Invoice Microservice
Query: Get loyalty points by user
Loyalty Program Microservice
Command: Send special offer notification
Query: Get setting for registered user
Command: Register user
API Gateway Microservice
Command: Update user settings
Notifications Microservice

ABOUT ME

Christian Horsdal
Independent Consultant

www.horsdal-consult.dk
c.horsdal@gmail.com
@chr_horsdal



NANCY
MOST VALUED
MINION

MVP
Microsoft®
Most Valuable
Professional

Links

- My blog: www.horsdal-consult.dk
- My microservices book: www.manning.com/books/microservices-in-net-core
- The Polly library: github.com/App-vNext/Polly