

Creating Background Services in ASP.NET Core 6 and .NET 6

Adding Background Tasks to ASP.NET Core Applications



Steve Gordon

.NET Engineer and Microsoft MVP

@stevejgordon | www.stevejgordon.co.uk

Welcome!





Goals

- Learn to develop background tasks in ASP.NET Core and build .NET worker services
- Become productive as quickly as possible



Version Check



This course was created by using:

- ASP.NET Core 6
- .NET 6
- C# 10
- Visual Studio 2022



Version Check



This course is 100% applicable to:

- .NET 6 and .NET 7
- ASP.NET Core 6 and ASP.NET Core 7

Most code will work fine with older supported versions of ASP.NET Core



Relevant Notes



A note on frameworks and libraries:

- A new version of .NET releases each year
- The background services libraries change very little between versions
- Microsoft aim to maintain backward compatibility between releases



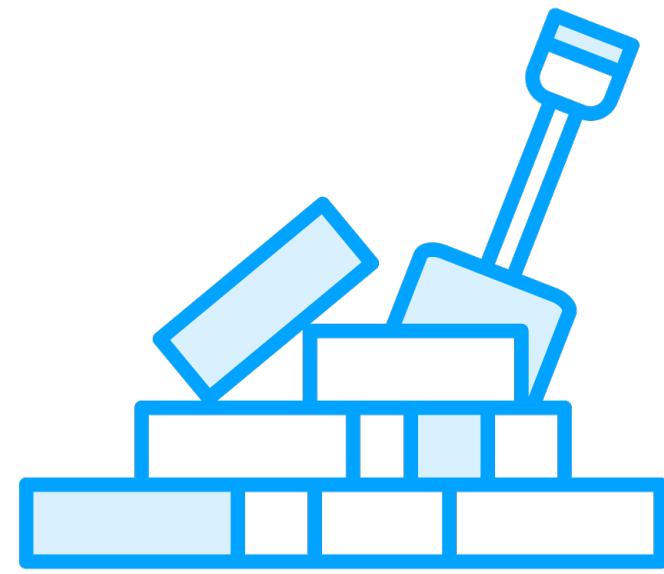
Overview

Create hosted services in ASP.NET Core

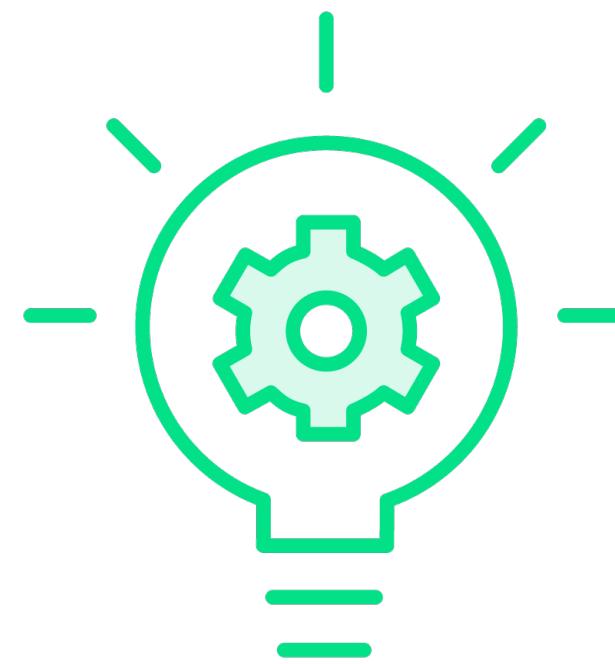
- Interval-based workload
- Transfer data between web requests and hosted services
- Register hosted services



Later in This Course



Building .NET worker services



Advanced hosted service concepts



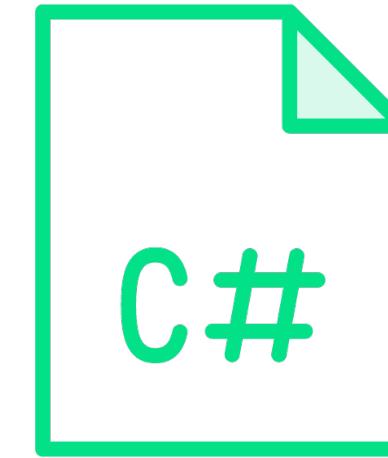
Running worker services in production



Course Prerequisites



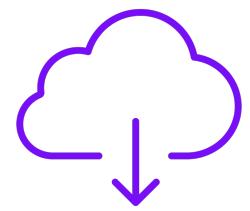
**Strong knowledge of .NET and
ASP.NET Core**



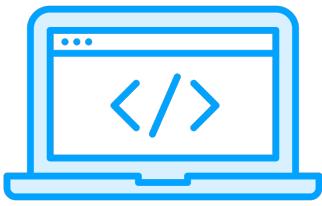
**Experience with coding
using C#**



Follow Along



Download the exercise files



The solution requires the latest .NET 6.0.x SDK
<http://dot.net>



An IDE such as Visual Studio Community Edition or an editor such as Visual Studio Code



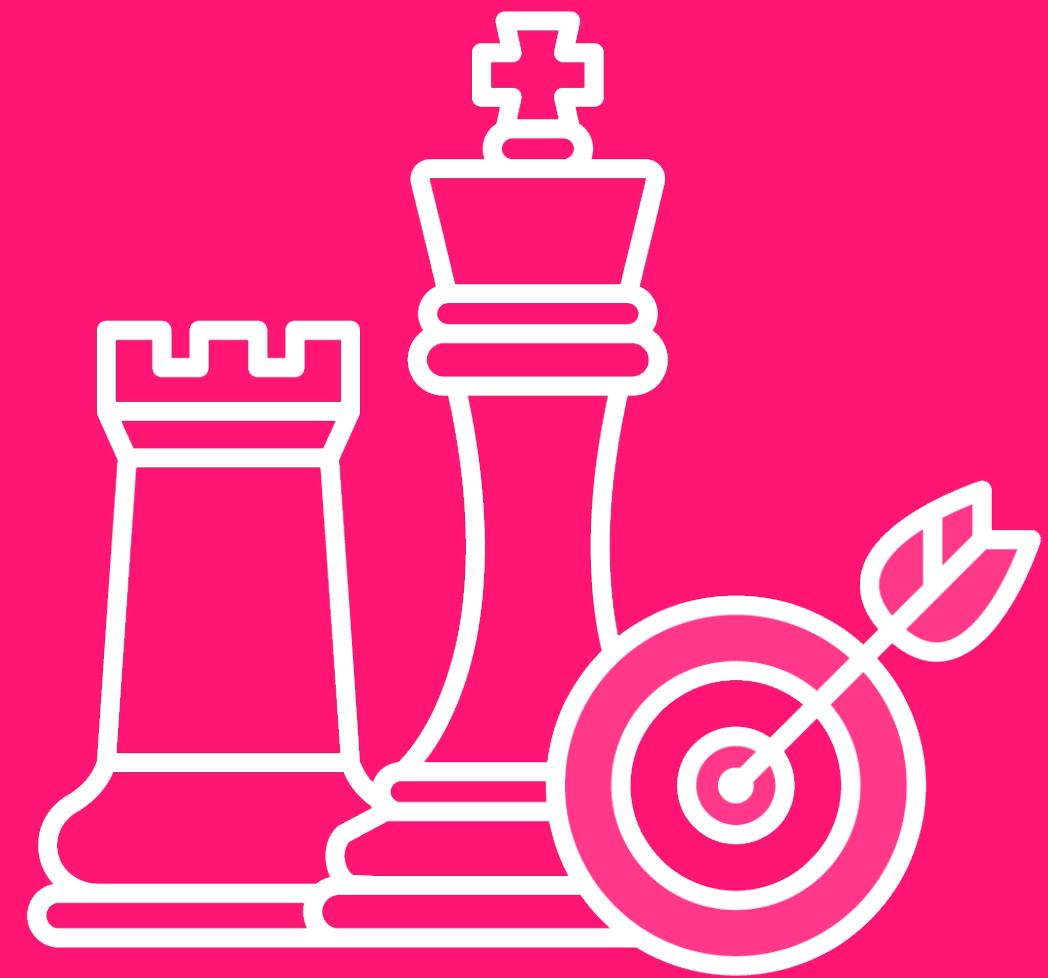
Let's Get Started...





Introducing the Tennis Booking Application





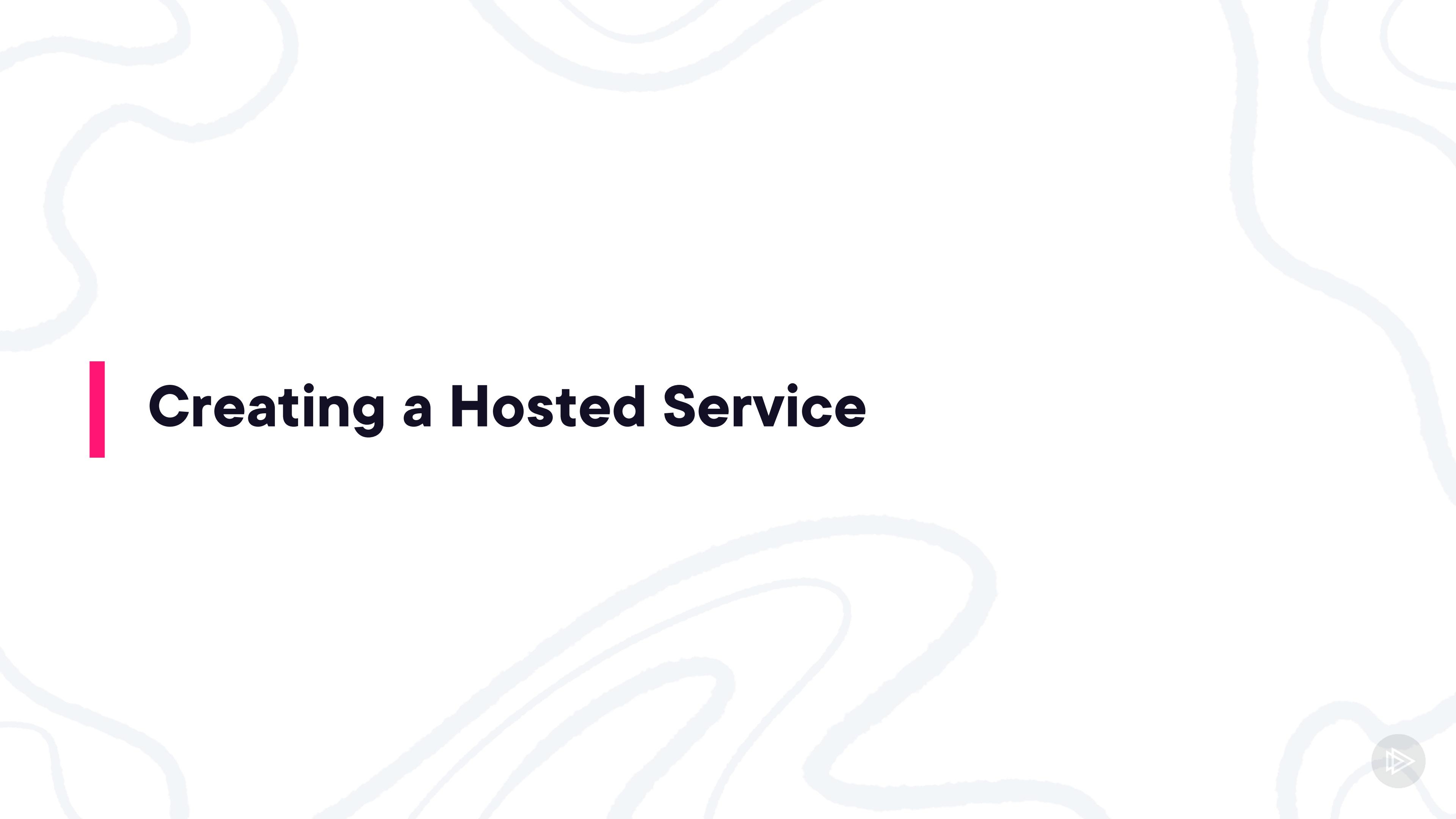
Focus

Demonstrating and explaining the
functionality of hosted services and worker
services.



We won't cover ASP.NET
Core fundamentals in this
course.





Creating a Hosted Service

Create a hosted service

- Periodically perform background work



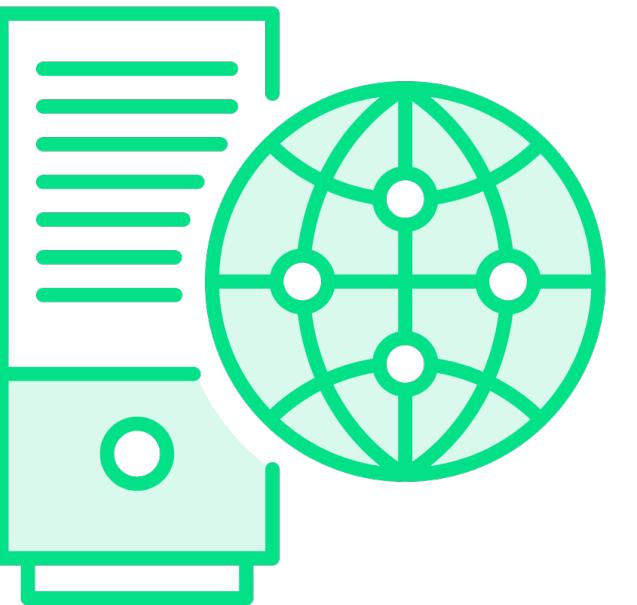
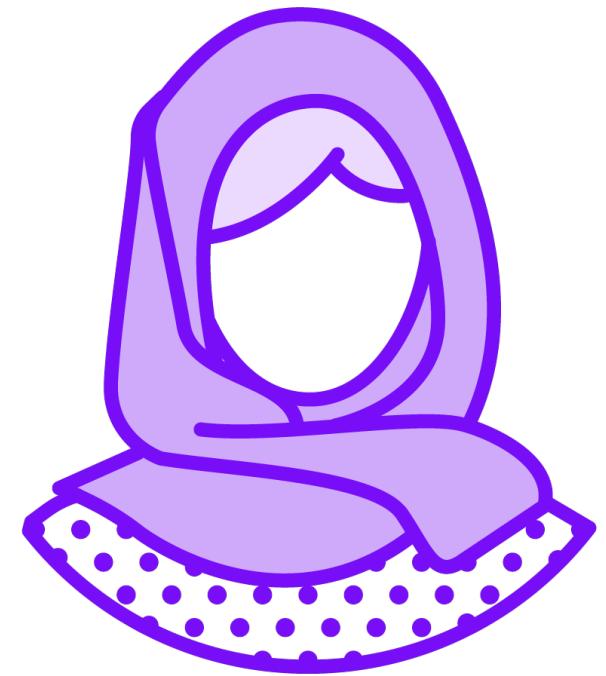
Hosted Service | Background Service



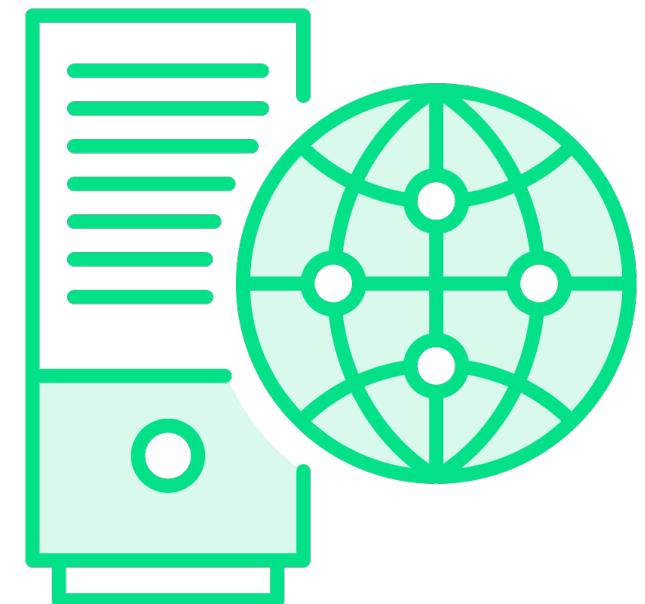
These terms are used interchangeably



Home Page Requests



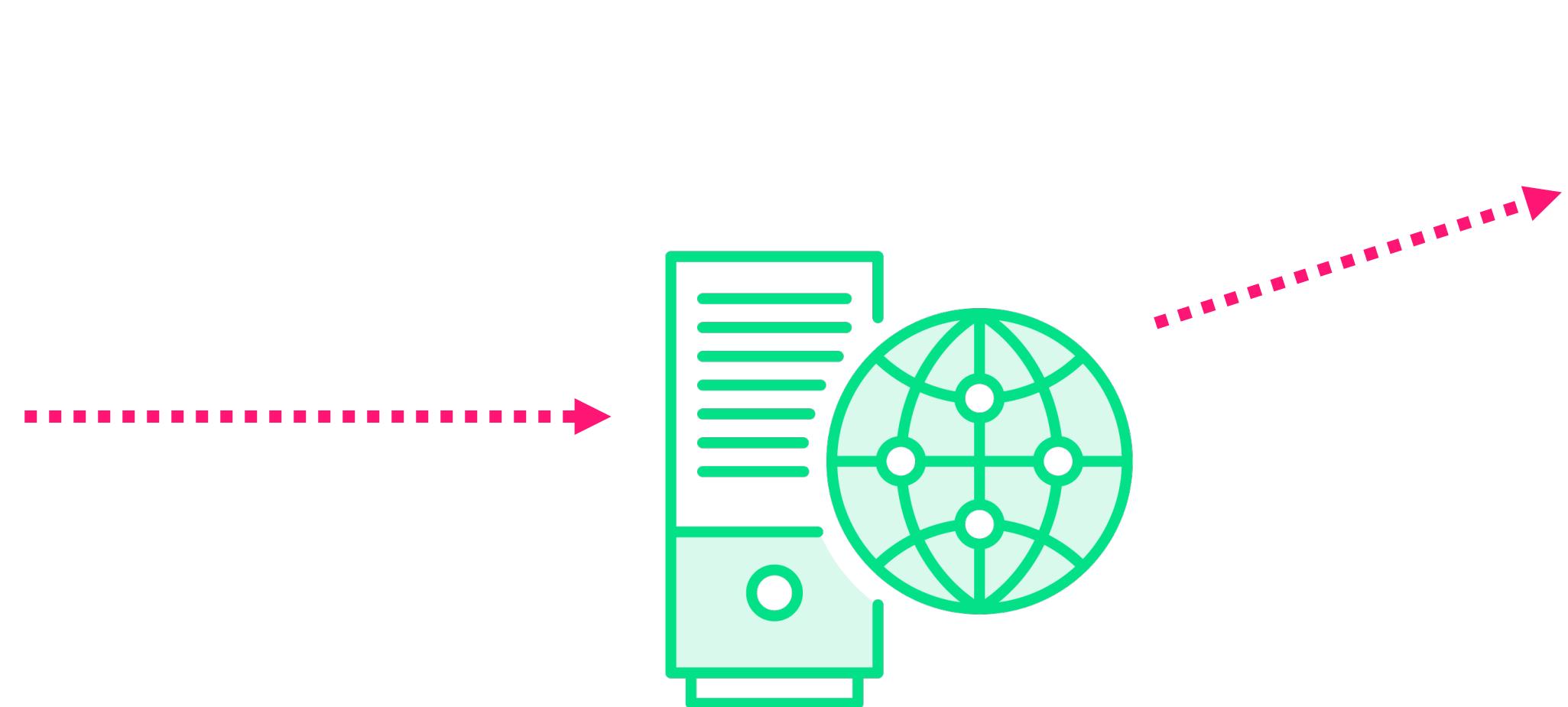
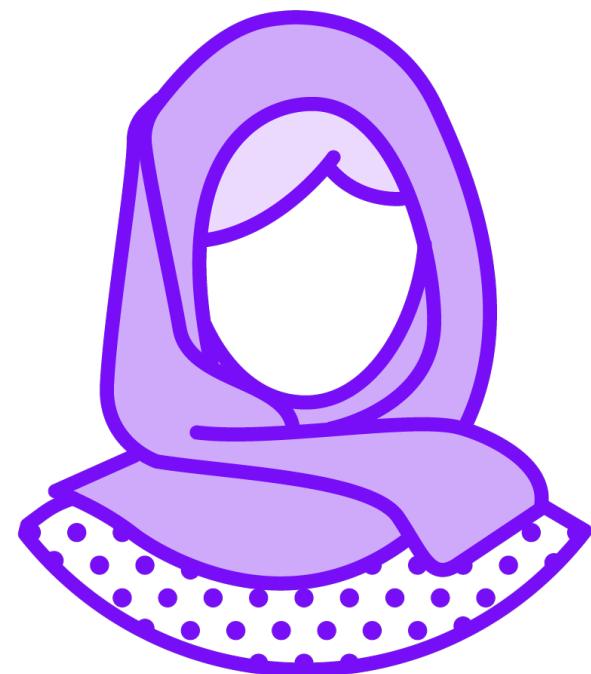
Cache



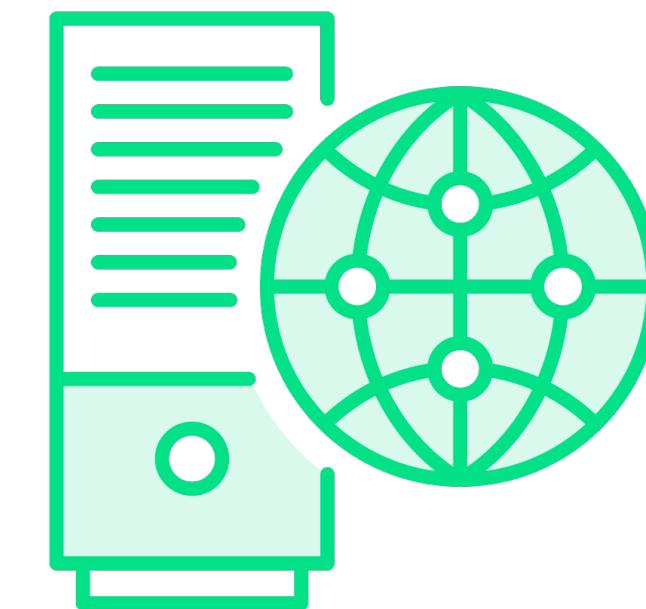
Weather API



Home Page Requests



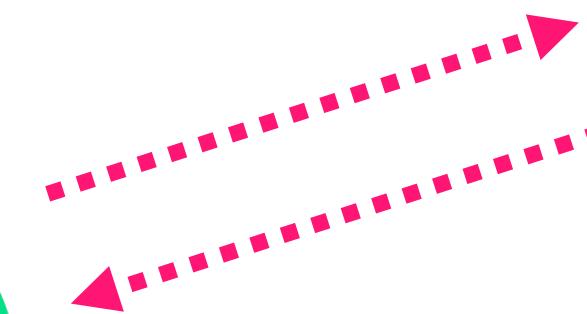
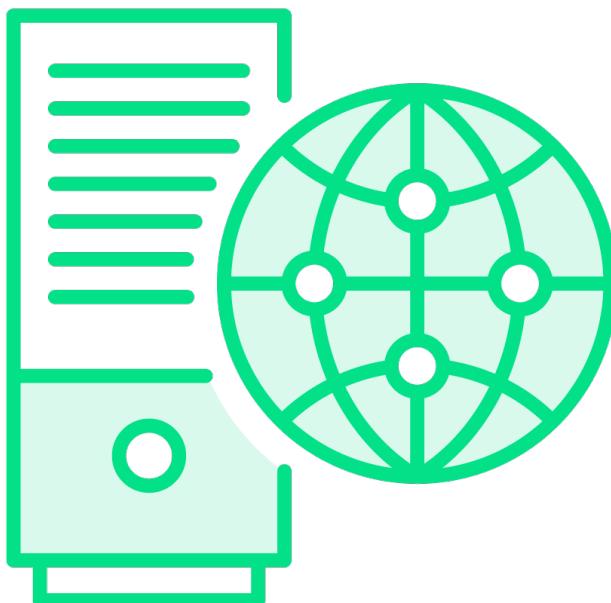
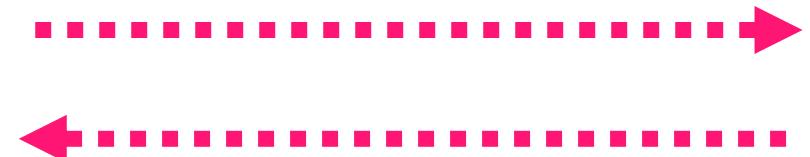
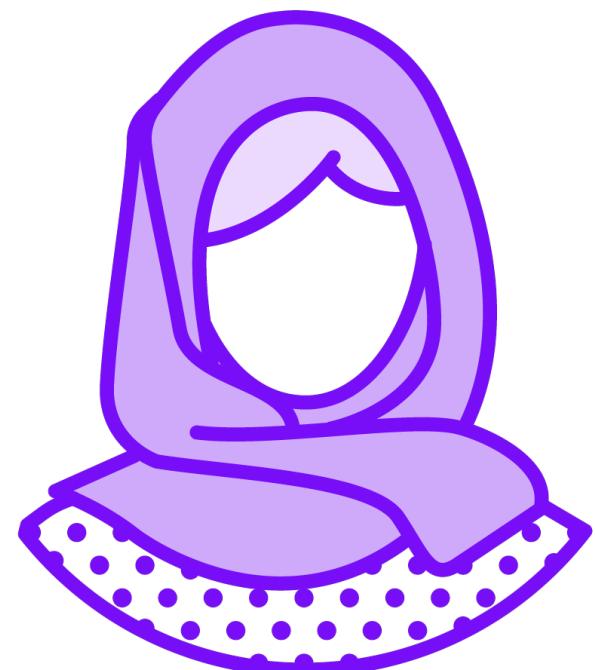
Cache



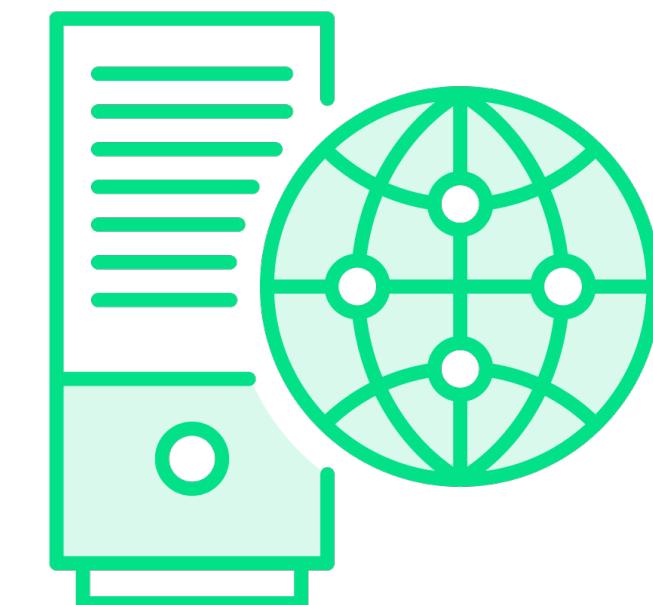
Weather API



Home Page Requests



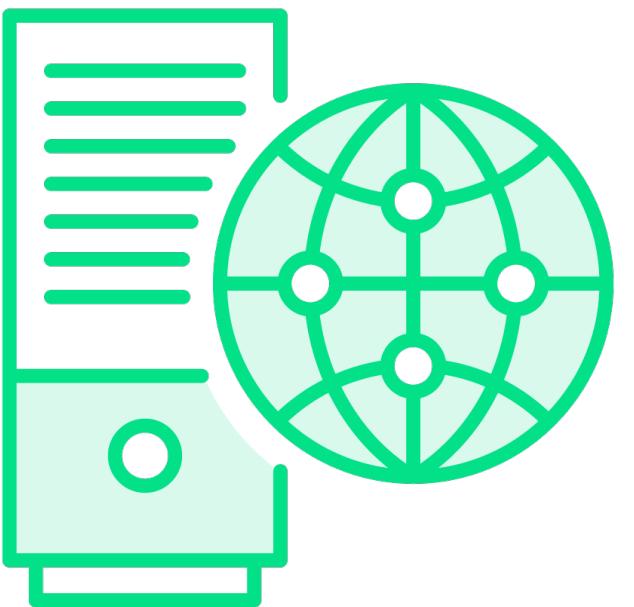
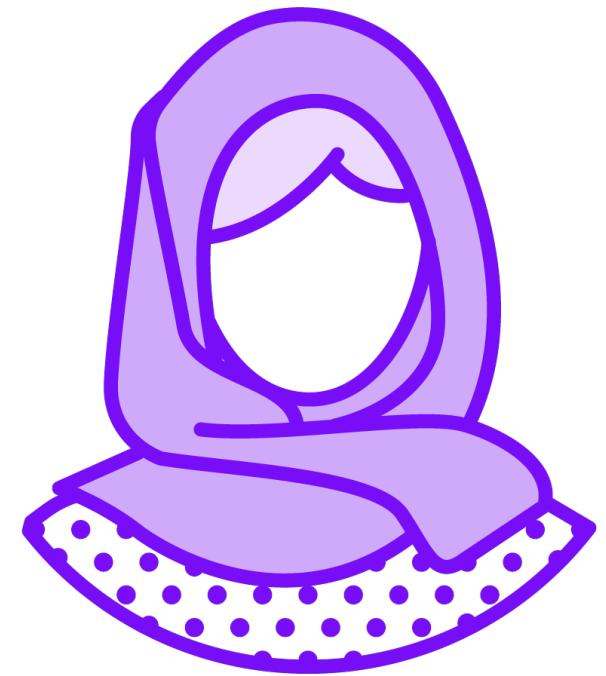
Cache



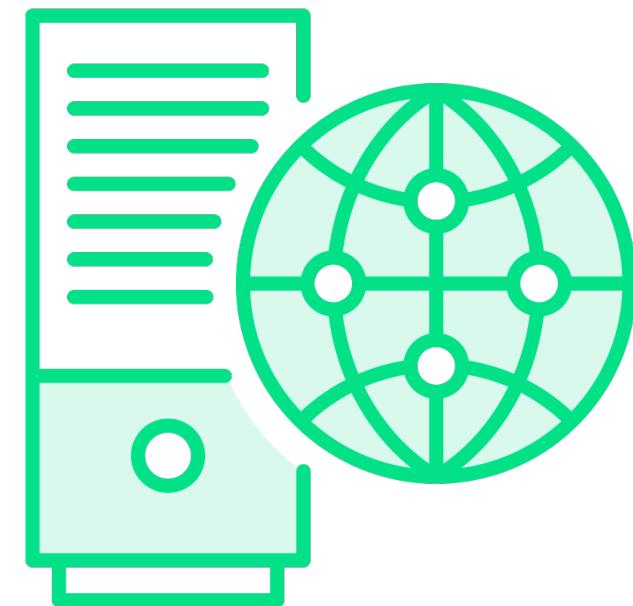
Weather API



Home Page Requests



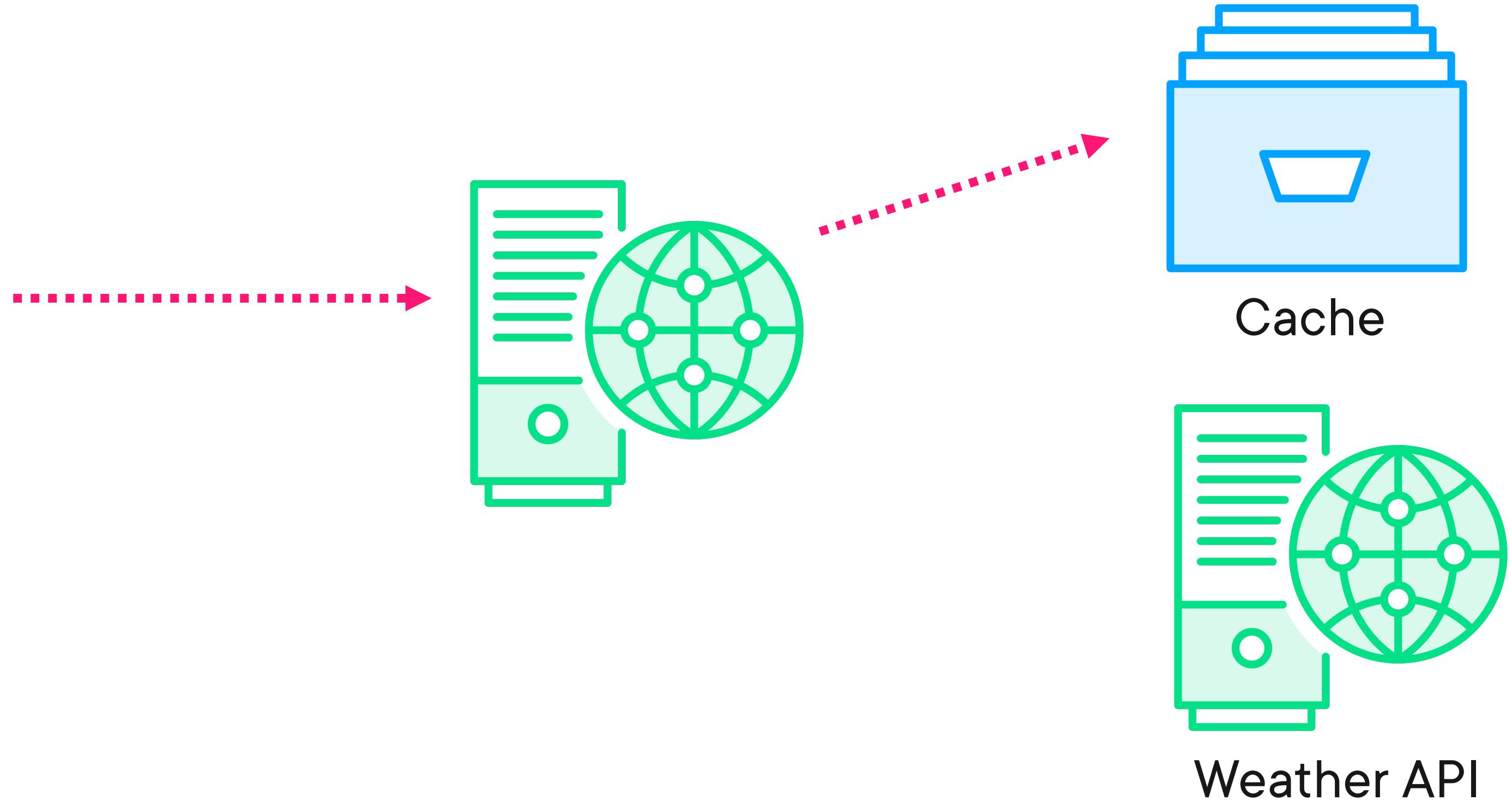
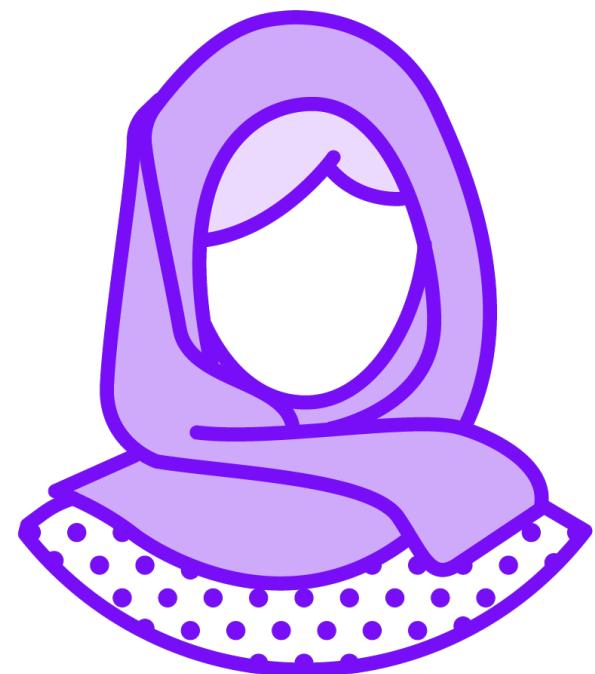
Cache



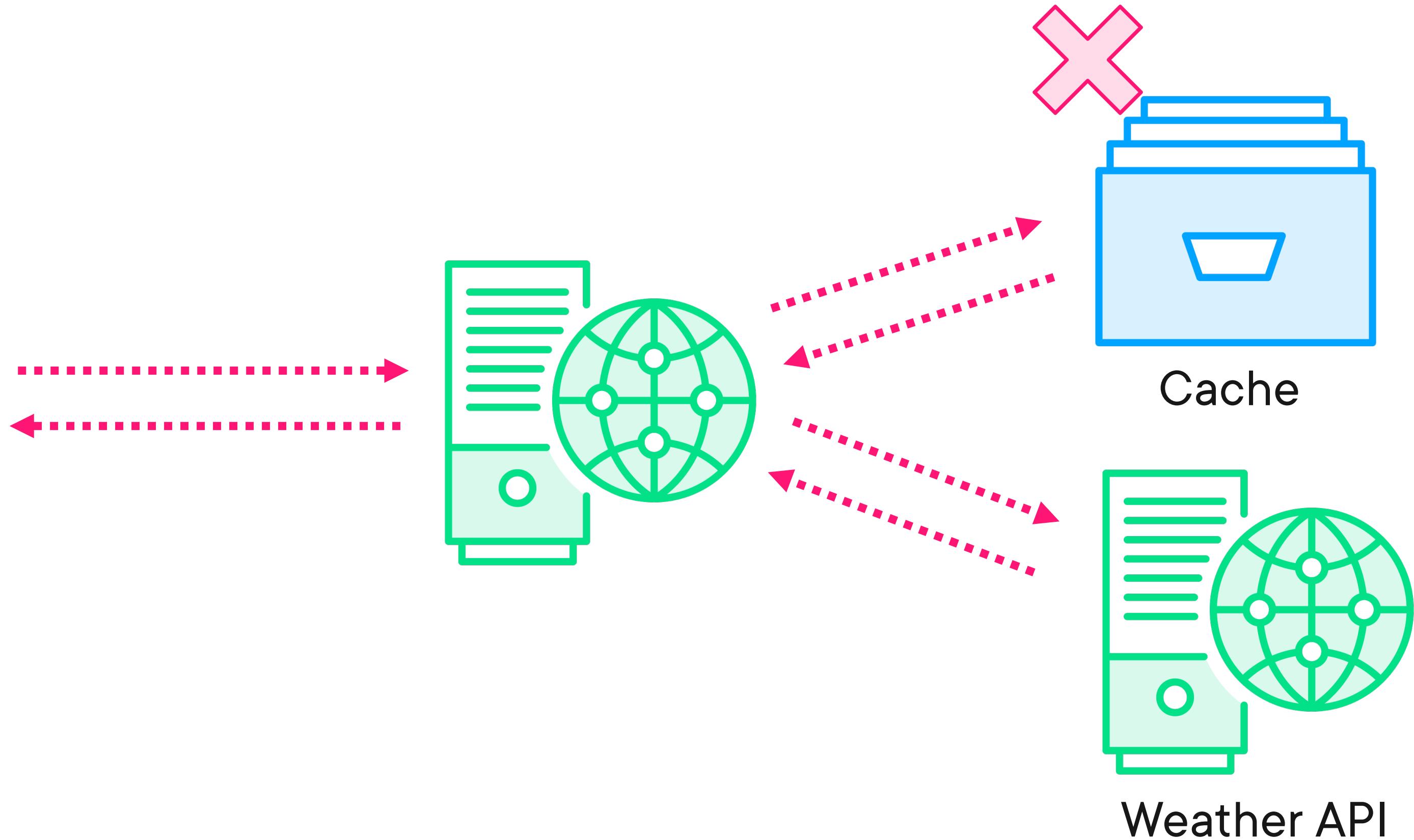
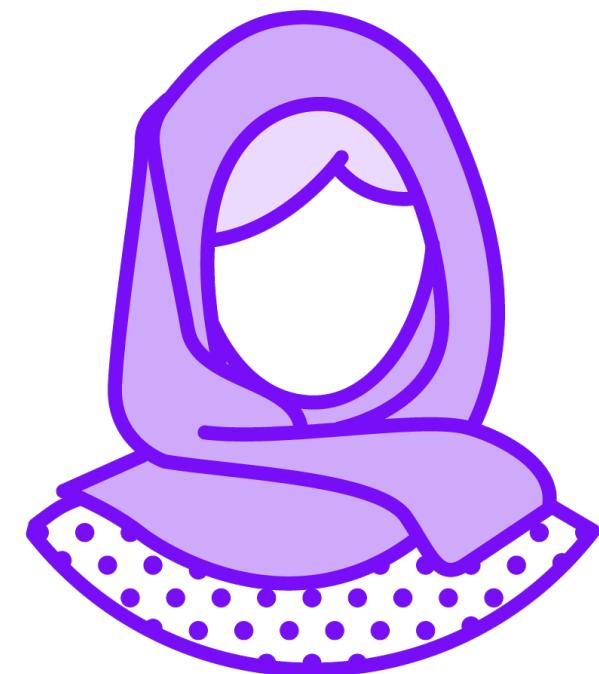
Weather API



Home Page Requests



Home Page Requests



Weather Data Caching

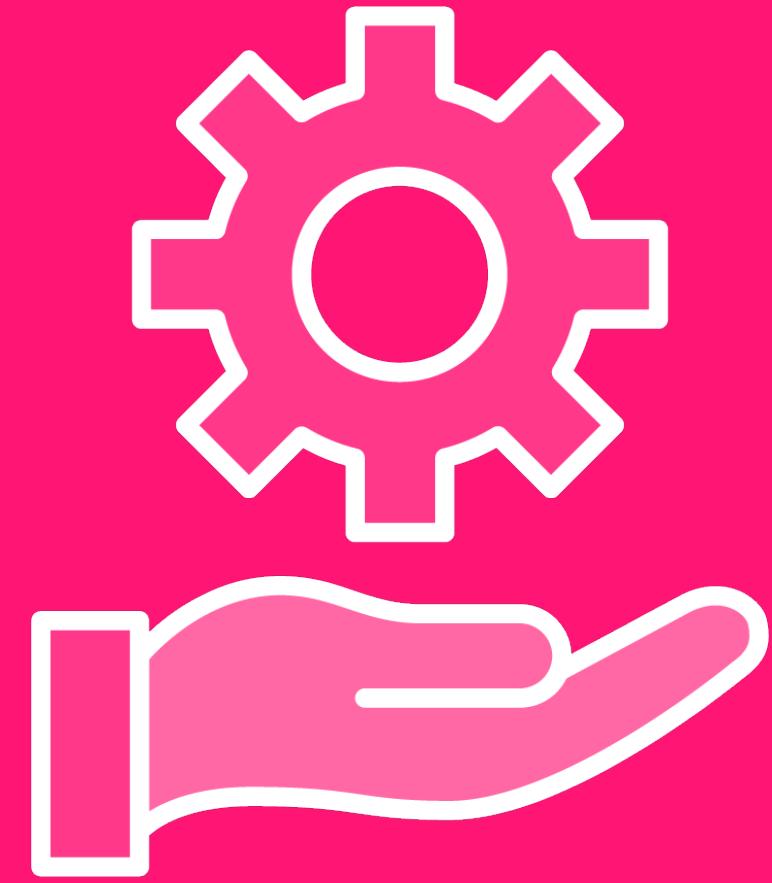


- When the cache expires, the subsequent user request is impacted**
- Forecast data is cached for a relatively long period**
- Data can become stale if cached for too long**





What are Hosted Services?

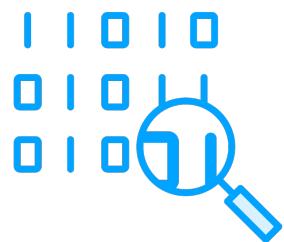


Hosted Services

Provide a clean pattern for executing code, within an ASP.NET Core application, outside of the request pipeline. Each service is started as an asynchronous background Task.



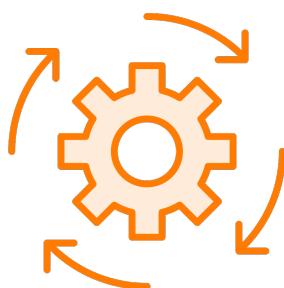
Use Cases



Polling for data from an external service



Responding to external messages or events



Performing data-intensive work, outside of the request lifecycle





Background Service

Abstract base class for creation of background services

BackgroundService.cs

```
public abstract class BackgroundService : IHostedService, IDisposable
{
    protected BackgroundService();
    public virtual void Dispose();
    public virtual Task StartAsync(CancellationToken cancellationToken);
    public virtual Task StopAsync(CancellationToken cancellationToken);
    protected abstract Task ExecuteAsync(CancellationToken stoppingToken);
}
```

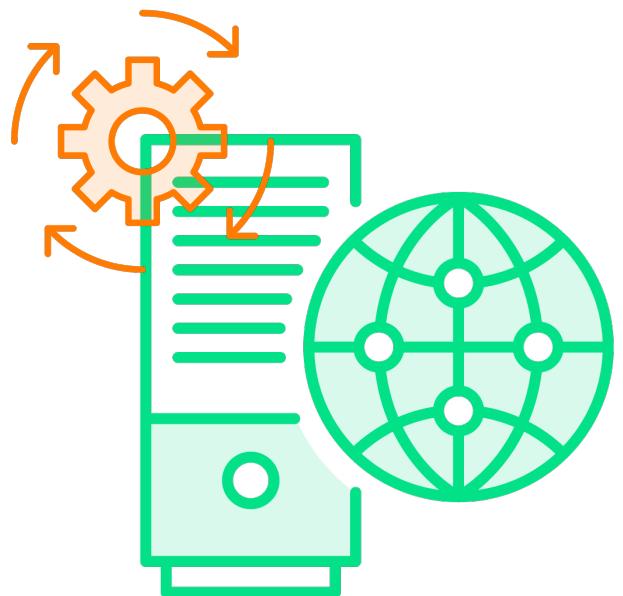
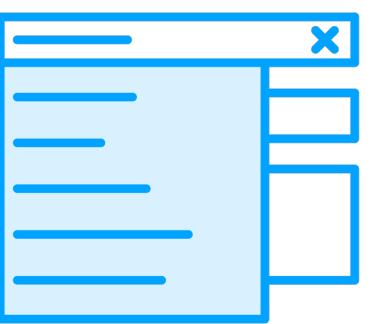
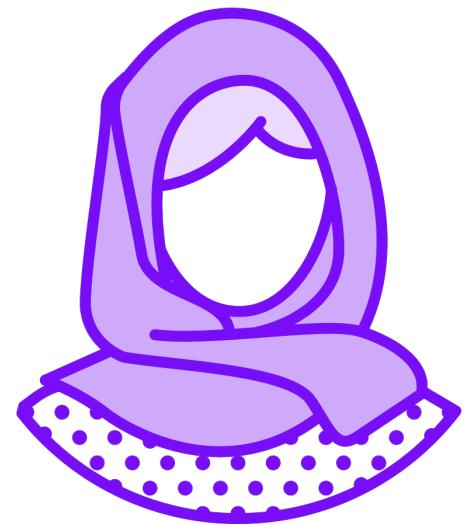


Coordinate between requests and hosted services

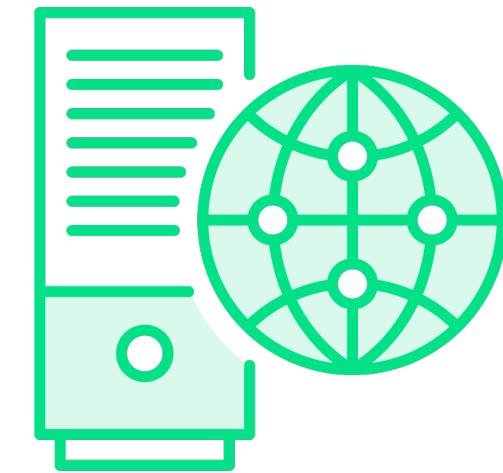
- Use ‘Channels’ to transfer data**
- Improve response times**



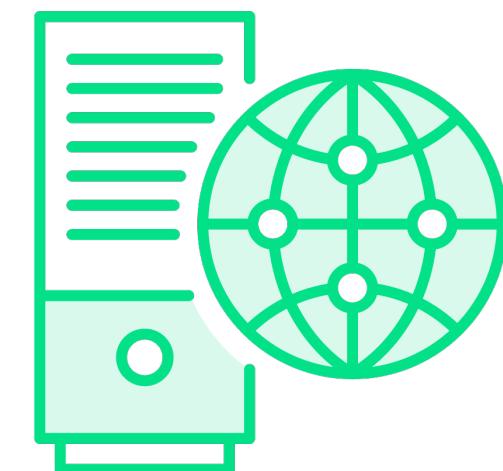
v1 Workflow



Tennis
Bookings



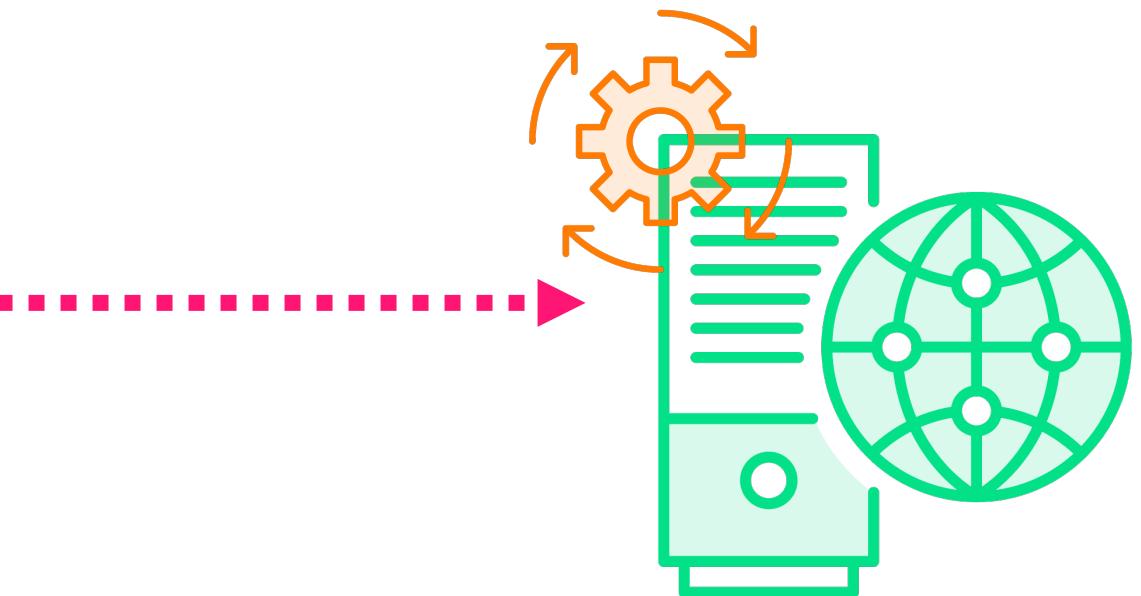
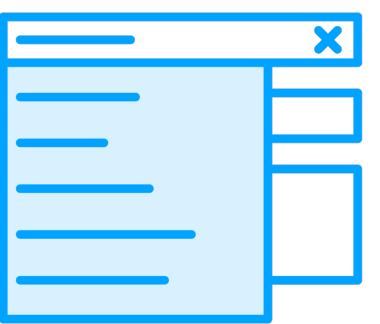
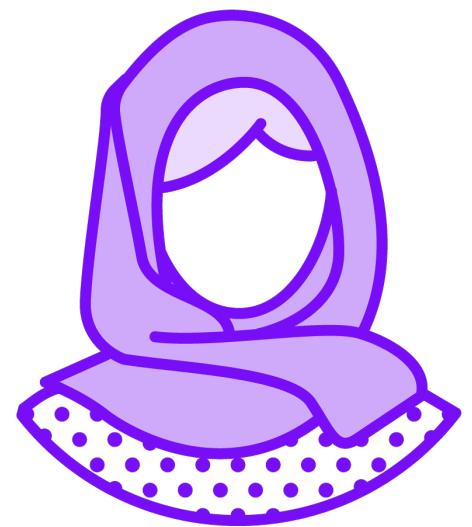
API



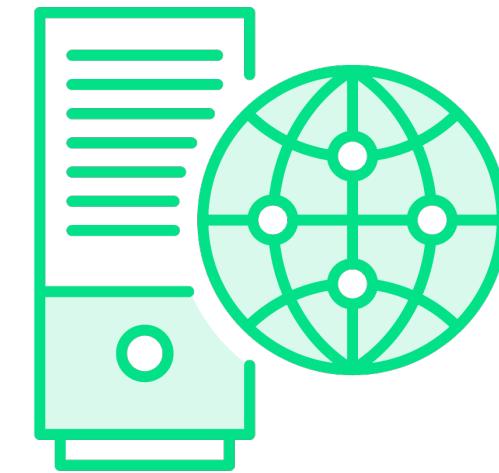
API



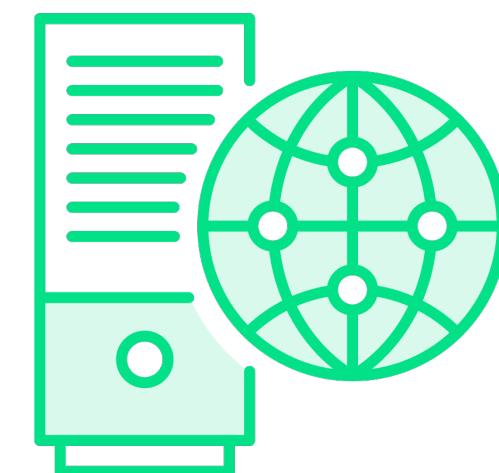
v1 Workflow



Tennis
Bookings



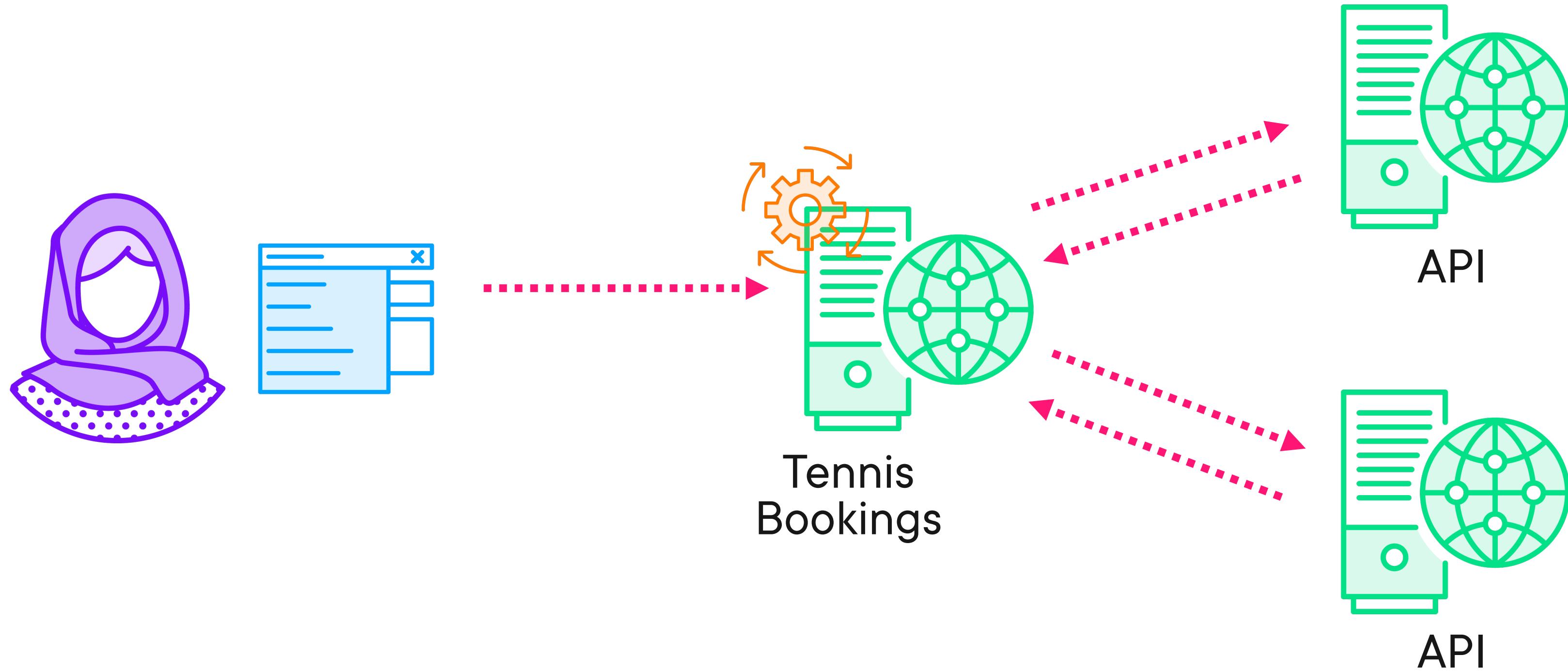
API



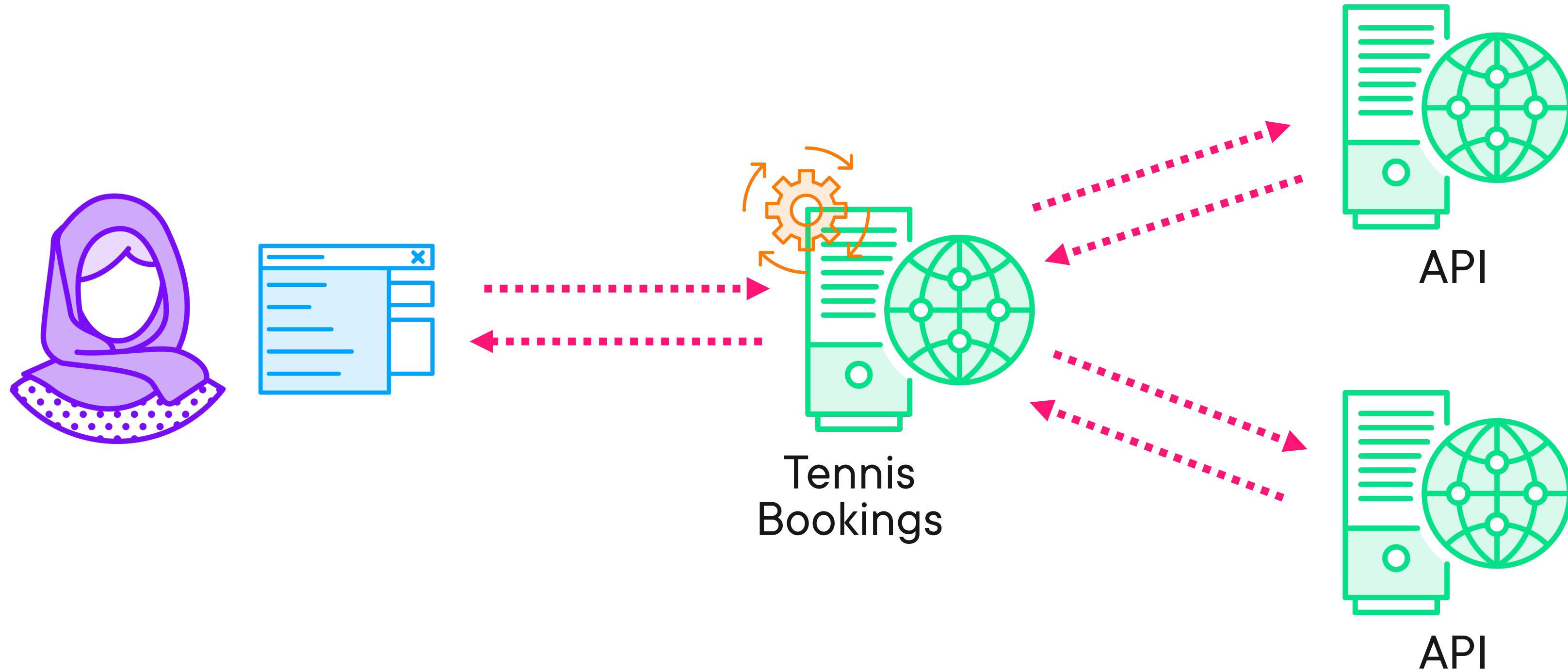
API



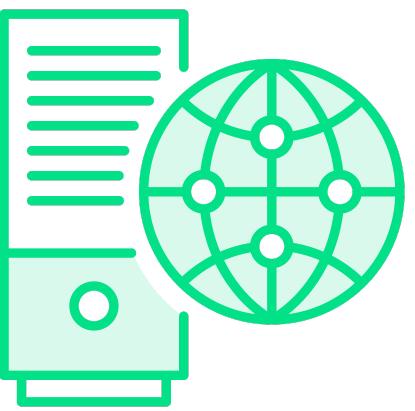
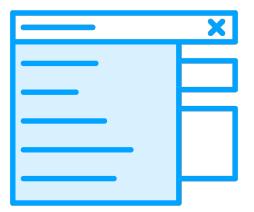
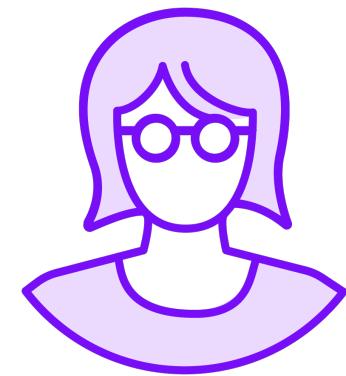
v1 Workflow



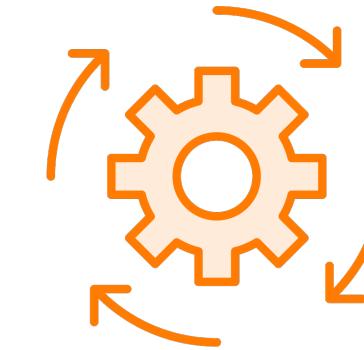
v1 Workflow



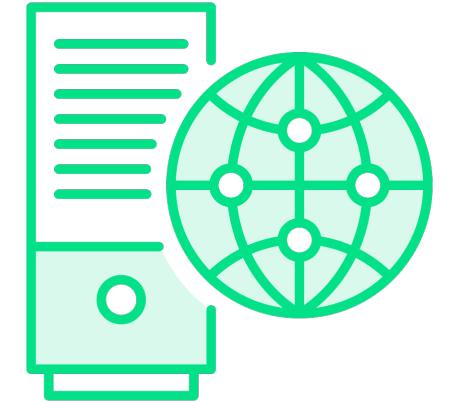
v2 Workflow



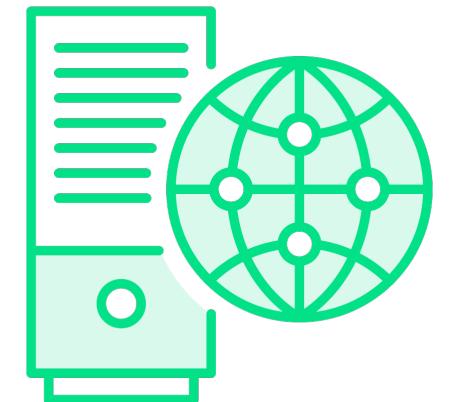
Tennis
Bookings



Hosted
Service



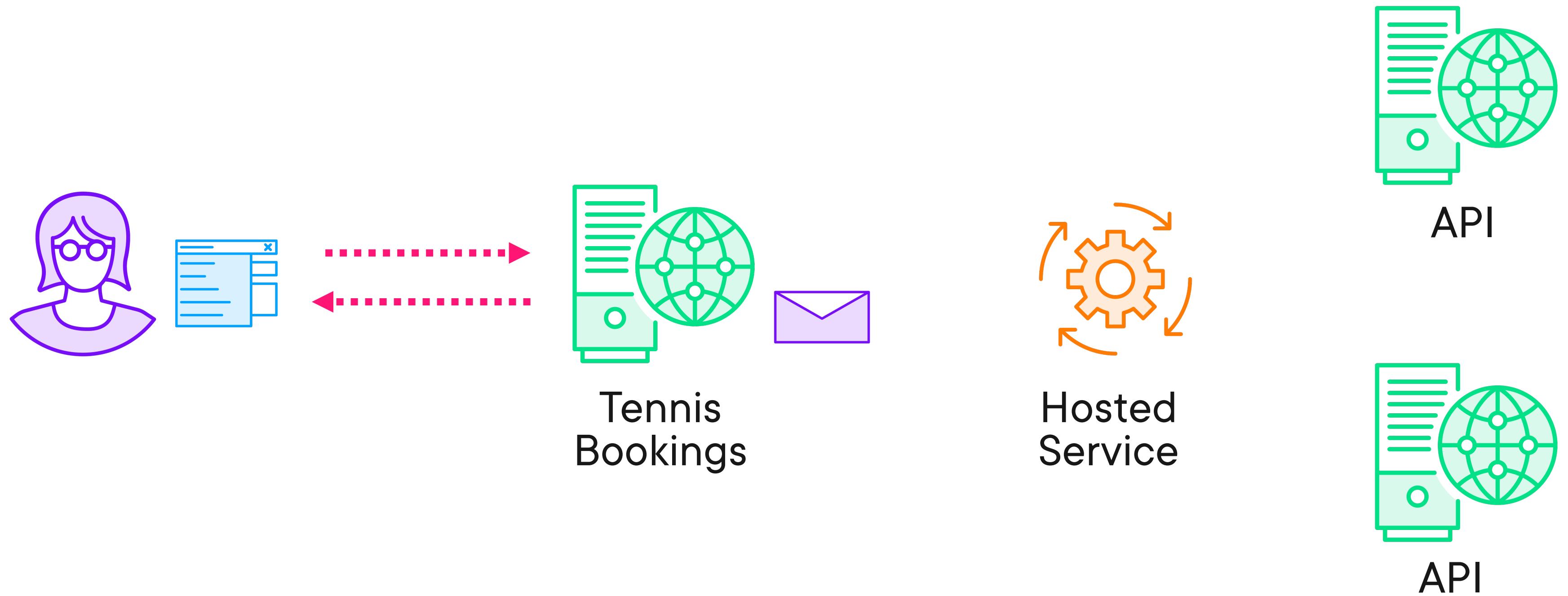
API



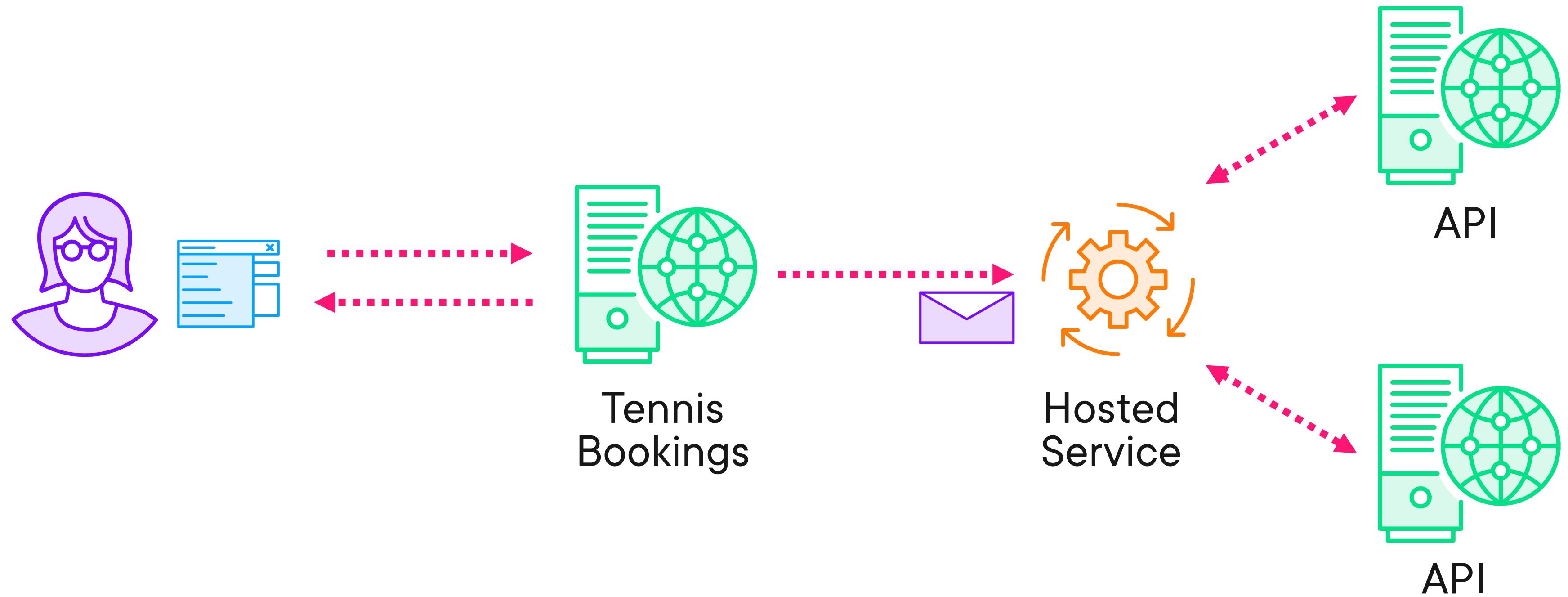
API



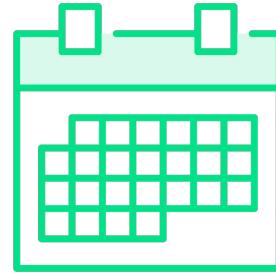
v2 Workflow



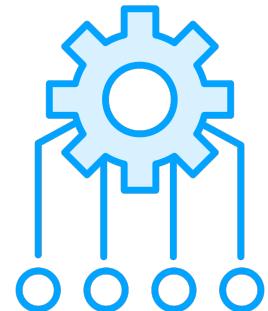
v2 Workflow



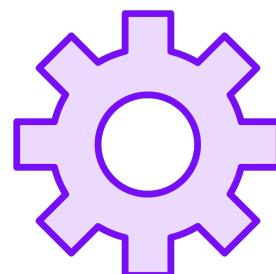
System.Threading.Channels



Released in May 2018



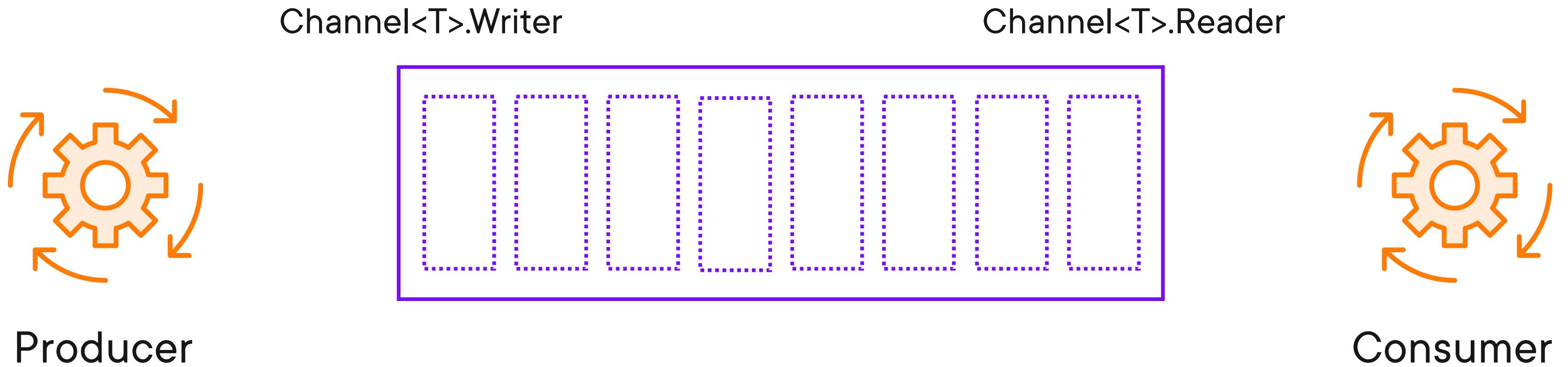
Thread-safe data transfer



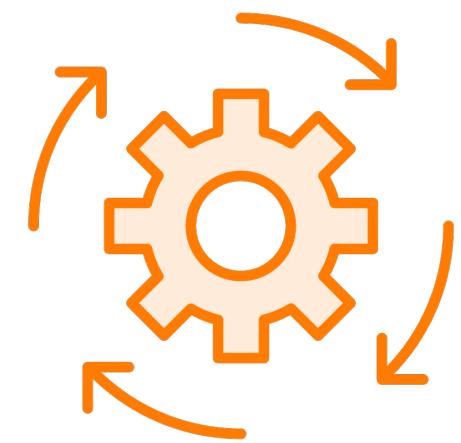
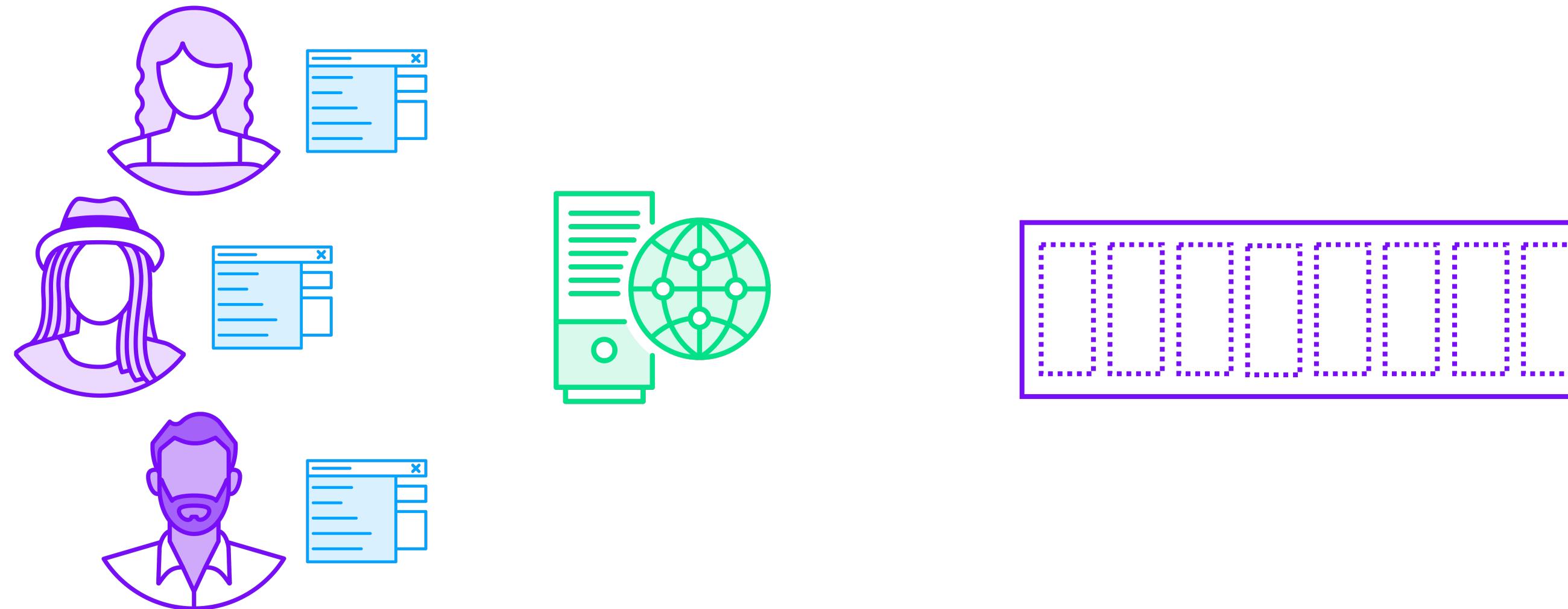
Have one or more producer, and one or more consumer



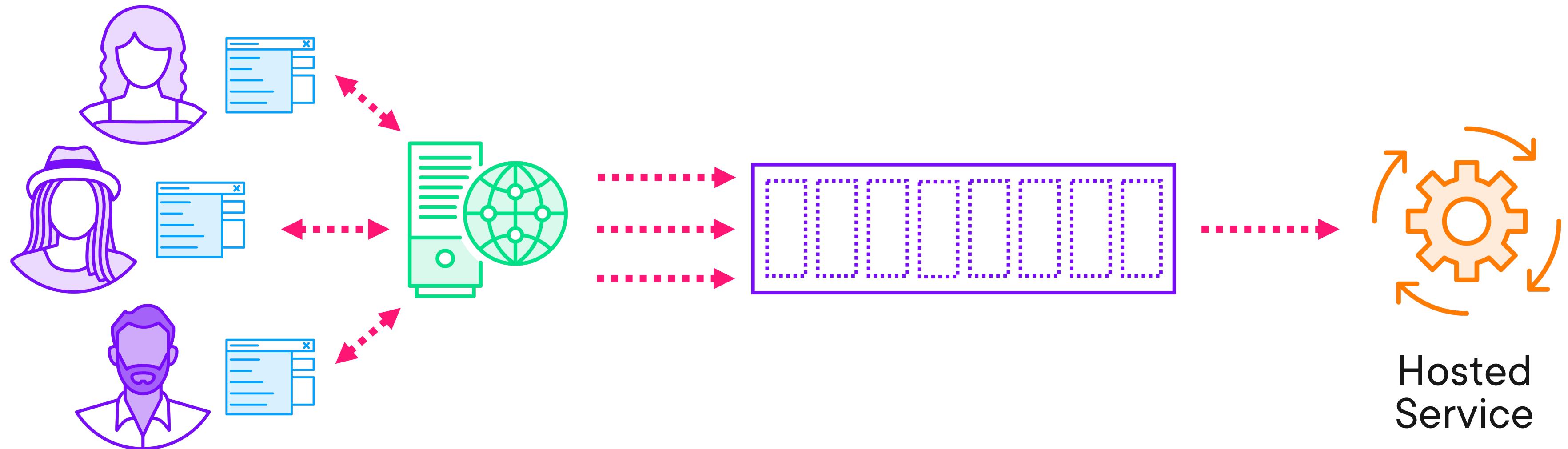
Producers and Consumers



Request to Hosted Service Coordination



Request to Hosted Service Coordination



Process data from a channel

- Create a new `BackgroundService`
- Read filenames from the channel
- Load and process uploaded files

Consume scoped services





Learn More

Dependency Injection in ASP.NET Core 6

Steve Gordon



Summary

IHostedService interface

BackgroundService abstract class

Implemented a hosted service for cache population

Passed data to hosted services from web requests

- Used a channel to transfer request data to a hosted service



Up Next:

Building .NET Worker Services

