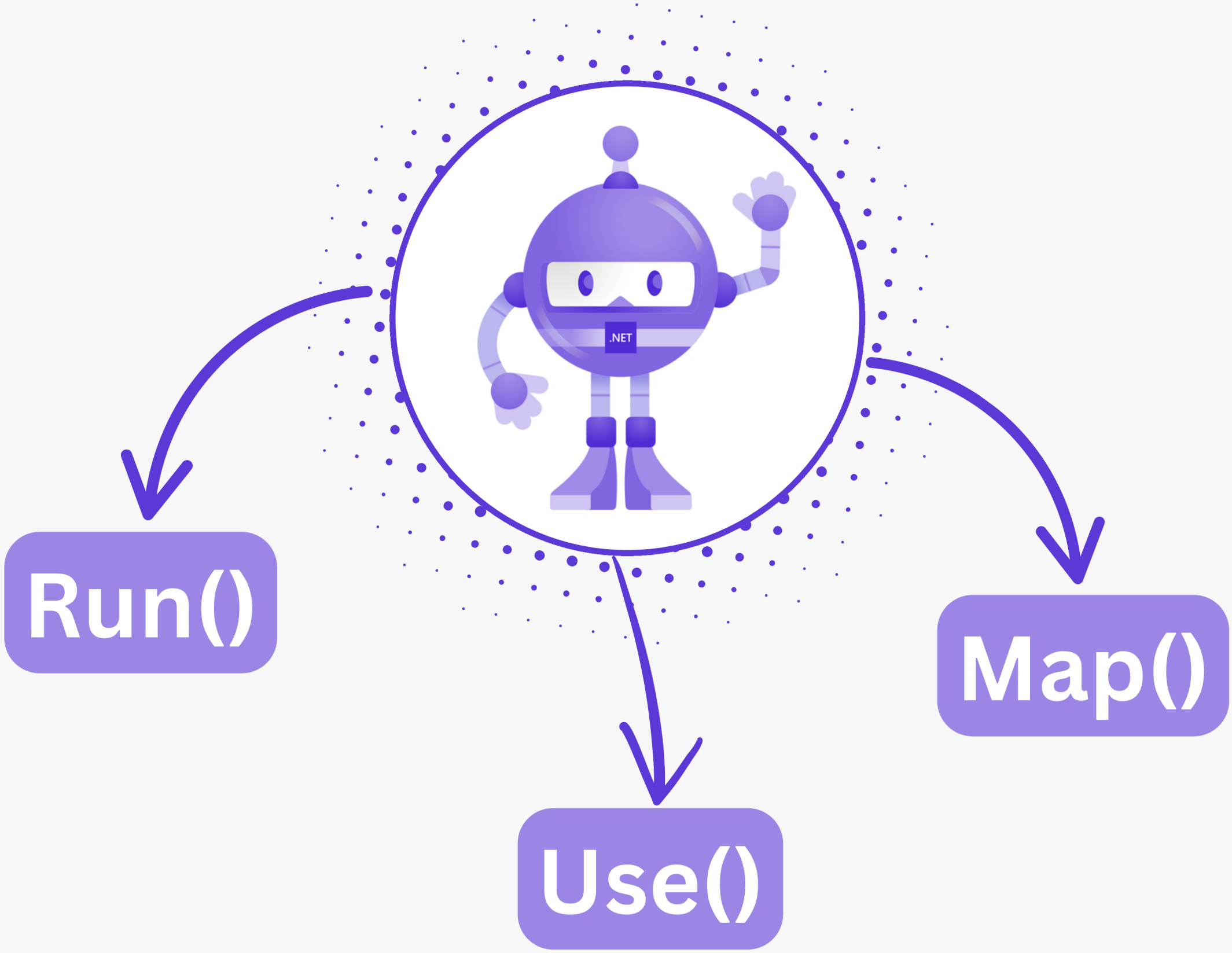
C#

Request Handling Functions in .NET



SHREYANS PADMANI

Use() Method

- The Use method is used to add middleware components to the pipeline. It allows passing the request to the next middleware component.
- It can call the next middleware component in the pipeline.
- Used for tasks like logging, authentication, exception handling, etc.
- If it does not call next(), it short-circuits the request.

```
app.Use(async (context, next) =>
{
    Console.WriteLine("Middleware 1: Before Next");
    // Calls the next middleware in the pipeline
    await next();
    Console.WriteLine("Middleware 1: After Next");
});
```

Map() Method

- The Map method is used for branching the request pipeline based on the request path.
- It routes requests to different middleware branches.
- Commonly used for modular request handling (e.g., API versioning, feature-based middleware).
- Middleware in the branch runs only if the request path matches.

```
app.Map("/hello", appBuilder =>
{
    appBuilder.Run(async context =>
    {
        await context.Response.WriteAsync("Hello World!");
    });
});
```

Run() Method

- The Run method is used to terminate the middleware pipeline. It does not call the next middleware; instead, it processes the request and sends a response.
- It does not call the next middleware (acts as a terminal middleware).
- Used for simple request handling when no further middleware is needed.
- Typically used at the end of the pipeline.
- Since Run does not pass control to the next middleware, no other middleware after it executes.

```
app.Run(async (context) =>
{
    await context.Response.WriteAsync("Final Response.");
});
```

Combined Code Snippet

```
app.Use(async (context, next) =>
{
    Console.WriteLine("Middleware 1: Before Next");
    await next();
    Console.WriteLine("Middleware 1: After Next");
});

app.Map("/hello", appBuilder =>
{
    appBuilder.Run(async context =>
    {
        await context.Response.WriteAsync("Hello from /hello");
    });
});

app.Run(async (context) =>
{
    await context.Response.WriteAsync("Final Middleware");
});
```