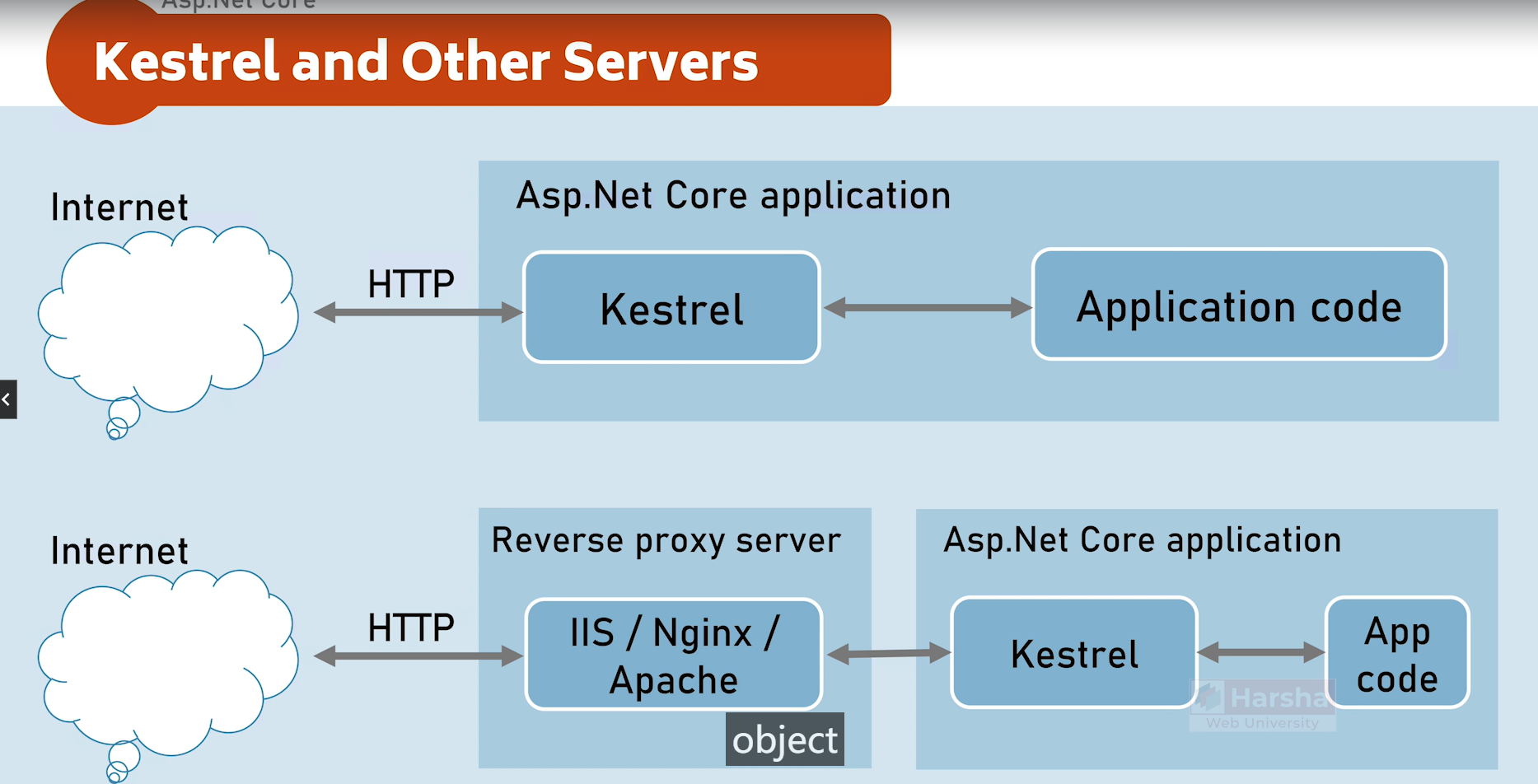
# 1. Kestrel and other servers

Development servers: after writing some asp.net core code, you will be testing and running application by the default servicer.

Production servers: after completion of the development, you need to put your codes to production servicers, which could receive the request from the internet anywhere in the world.

IIS is supported by windows operating system only, and Nginx and Apache are mainly supported by Linux, of course by window



Question:

1. What are Kestrel’s functions?

Kestrel is the default cross-platform http service for asp.net core application. It is lightweight and suitable for serving dynamic content. Typically used in conjunction with a reverse proxy for production environments and handling income Http requests and response.

1. What is the Reverse proxy server?

Kestrel doesn’t support some of advanced features that is required on the internet these days. A reverse proxy server forwards client requests to backend servers and returns the responses to the clients. Used in front of Kestrel to enhance security, load balancing, and other enterprise-level requirements.

1. Based on the flowchart at the top of image, how is the data processed?

The http request will be received from local or internet, and the Kestrel will receive that request and fills the information in the form of an Object which called http context. And it will forward it to Application code.

1. Why is the bottom one better?

First, the http request will be received by the reverse proxy server, and these severs transfer the same request to the real application server, kestrel. (In that way kestrel can receive the request from the real internet, but it won’t do that in the real-world case. Because reverse proxy server has some of advanced features that is requested on the internet today). And then the Kestrel will receive the request and forward it to the app code. It can execute some code and provide the response back to the kestrel, and kestrel will send the response back to the reverse proxy server. And then the same response will be sent back to the actual client. It means that in the real production scenarios this is the process what happens in the real world.

# 2. LaunchSettings.json

Question:

1. What is the LaunchSetting.json?

LaunchSetting.json is a configuration files in ASP.NET Core projects used to define setting for how the application is launched during development. This includes settings for different environment, URLs and other debugging options.

1. What’s the profile?

* A profile is a collection of settings which enables a particular serve to run our application when you start the project. Defines different profiles for launching the application. Each profile can have unique settings.

1. In the LauchSetting.Json file, what do the all variables represent?