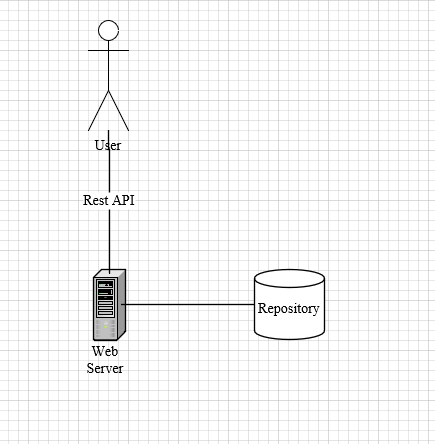
HIGH LEVEL DESIGN

Word statistics

# Application Architecture



# Web application architecture

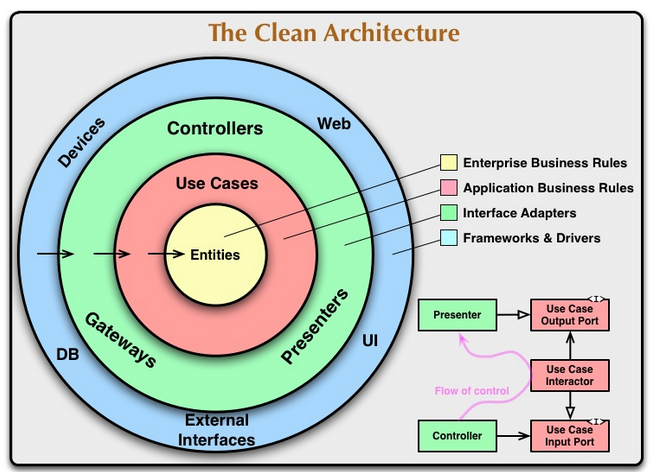
Web application based on .Net core 3.1 (LTS). Application have 2 end point

* word/analyze – this end point receives a different type of inputs
  + plain text
  + file path with data
  + url to service that return data

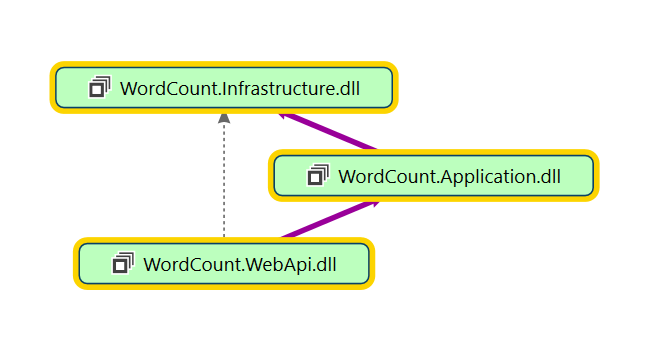
the end point should analyze data and store in for future use

* word/statistics – this end point receive an word and return number of occurrences of that word

The application architecture implements Clean Architecture design



## Dependencies



## Modules Responsibility

* WordCount.WebApi – this component is a controller it receives user request and return response. It should check/validate user input but it doesn’t have any business logic.
* WordCount.Application – this is a core module of application. It provides an API to any consumer such WebApi or some other consumer that might use it functionality

This module encapsulates business logic and should not be changed if no changes was made to business logic

* WordCount.Infrastructure – this module responsible provide an API and functionality and implementation for any Infrastructure that might be required such as consume external services (REST, WCF), implement a repository interface (Memory, File, Cache, DB)

## Logic flow

As we can see the in all cases the logic flow is a same the difference between is just data source.



So when user submit data for analysis we can run all types with in a same request



# Assumptions

1. The URL data processing is should be in GET method, it a default method if not specified
2. Exception handling handled by **ErrorHandler**.cs middleware in WebApi component, thus code doesn’t contain a try catch blocks

This is a simplest way to handle exception without crashing the client.

The correct way is to handle each operation by it exception type