Ebike application

Requirements and specifications

It is required to create an application that allows users to register themselves and then use one of the electric bikes that are on the environment to start a ride, moving the bike's position.

Users can registers themselves and use a bike from a remote GUI, and the administrators can create electric bikes from another remote GUI.

There should be data persistence.

Should be possible to add new functionalities available to the users dynamically.

My specific requirements

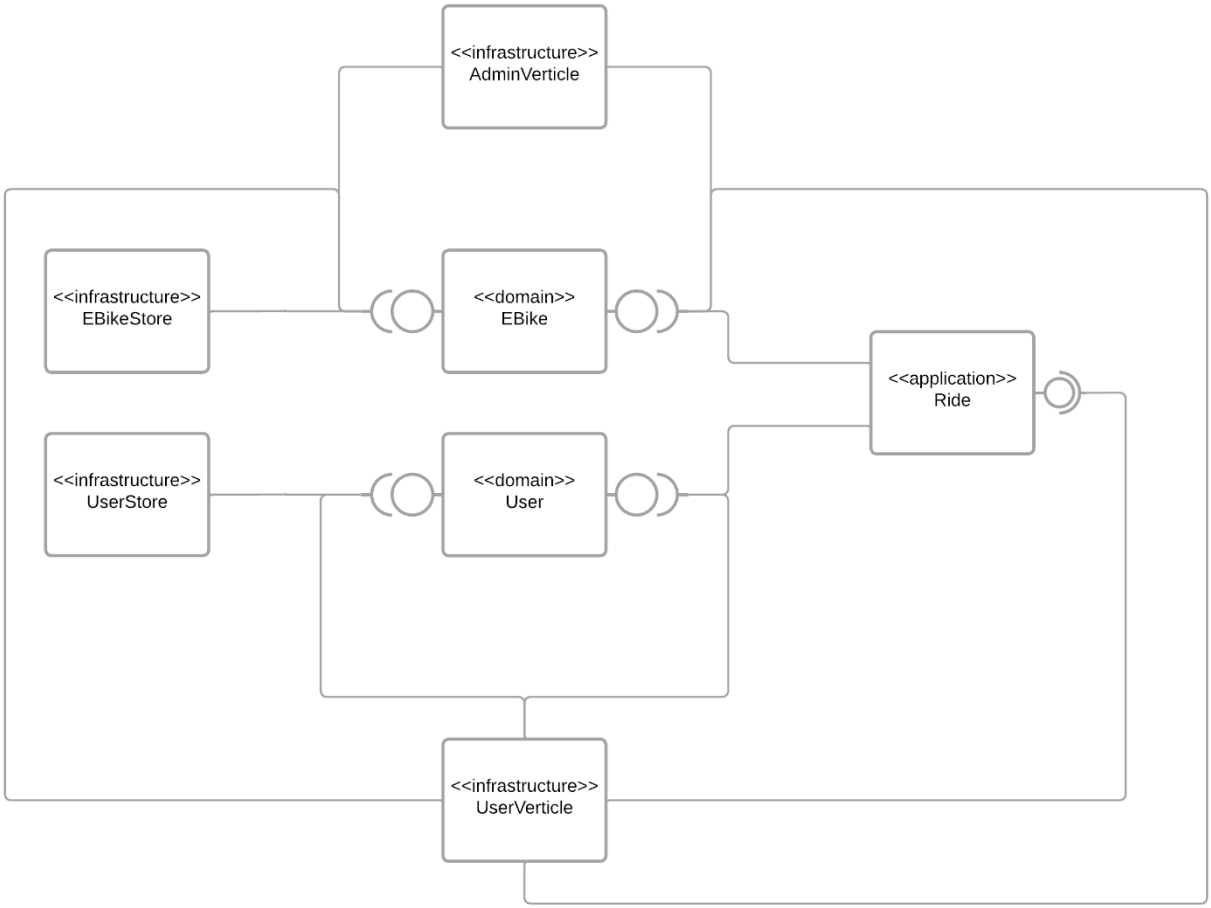
To create persistency, it has been decided to allows the user to save its data, to load its data, and to load and change the electric bikes data (with a ride).

The admin can load, save and change the electric bikes data.

Architecture

The architecture chosen is the clean architecture, with the Domain at the center that expose all the ports needed to use it. It should expose one outbound port and one inbound port.

The ride will use the inbound ports. The stores will use the outbound ports. The verticles need to use both the inbound and outbound ports to exchange messages with a user with a REST API and with a websocket (real-time).



Quality attributes scenarios

Availability:

- If a user (Source) creates a user already existing (Stimulus) on the system (Artifact) in normal operations (Environment) the system informs the user (Response) with no downtime (Response Measure).

The same as above works with creation of EBikes from the admins.

Consistency:

- If more users (Source) start riding some bikes (Stimulus) on the system (Artifact) in normal operations (Environment) the users will see the same information about the bikes (Response) at the same time (Response Measure).

Consistency, Scalability:

- If more users (Source) start riding some bikes (Stimulus) on the system (Artifact) in normal operations (Environment) the information stored will be the same (Response) regardless of how many bikes or users are being modified (Response Measure).

Fitness functions

There are two fitness functions to test the clean architecture: first, the domain should not depend on application or infrastructure, second, the application should not depend on infrastructure.