The BridgeApp design consists of 3 main packages:

1. **Controllers package –** holds classes/controllers responsible for waiting for HTTP requests from WebService and so calling appropriate methods on LoraService.
2. **Application package –** presents two main functionalities of BridgeApp, such as:
   1. **Handling data from LoraWAN -** receiving data from LoraWAN, translating received data and sending translated data to the corresponding MongoDB collection
   2. **Handling requests from WebService –** responding properly to the HTTP requests, by pushing appropriate data to LoraWAN

The translation of data from bytes to lists and vice versa is handled strictly by LoraTranslator static class.

The HandleDevice class represents the thread for each of the devices that LoraService is controlling.

In order to keep in LoraTranslator the responsibility of choosing proper Repository interface, to which to send received data, the visitor pattern is used in the following way:

1. When the data needs to be translated from bytes array into List<Element> object, LoraTranslator is able to determine to what specific data object (located in data package) it needs to be translated.
2. When the visitor, in this case the MongoRepository class is called to save received data to Mongo, it doesn’t know about to what specific Mongo to save it. Therefore it “visits” each element in the list.
3. Every element then accept/react in specific way by calling appropriate method from Visitor interface back on the MongoRepository class
4. **Persistence package –** holds specific interfaces for all necessary collections in MongoDB

NOT SURE WHAT MORE TO SAY