



Here is a brief description of the design. First, there is a **Client** class that manages all of the boto3 libraries. The client has a **Request** and a **Widget** in it, but they are not class attributes. There is a method in the **Client** class called `process_next_request`. It takes the first request in the source bucket and loads it into the **Request** class. The **Request** class handles all of the parsing of the request. Then, it converts the request into a **Widget** by storing it in the **Widget** class. It then puts the widget in the destination bucket.

The **Widget** class is an abstract class. The **S3_Widget** and **DynamoDB_Widget** classes are instantiated depending on the method used. This uses the polymorphism of the abstract class, because the client doesn't need to know about the intricacies of converting the widget attributes to fit the desired destination. It just knows that it's a widget and processes it accordingly.