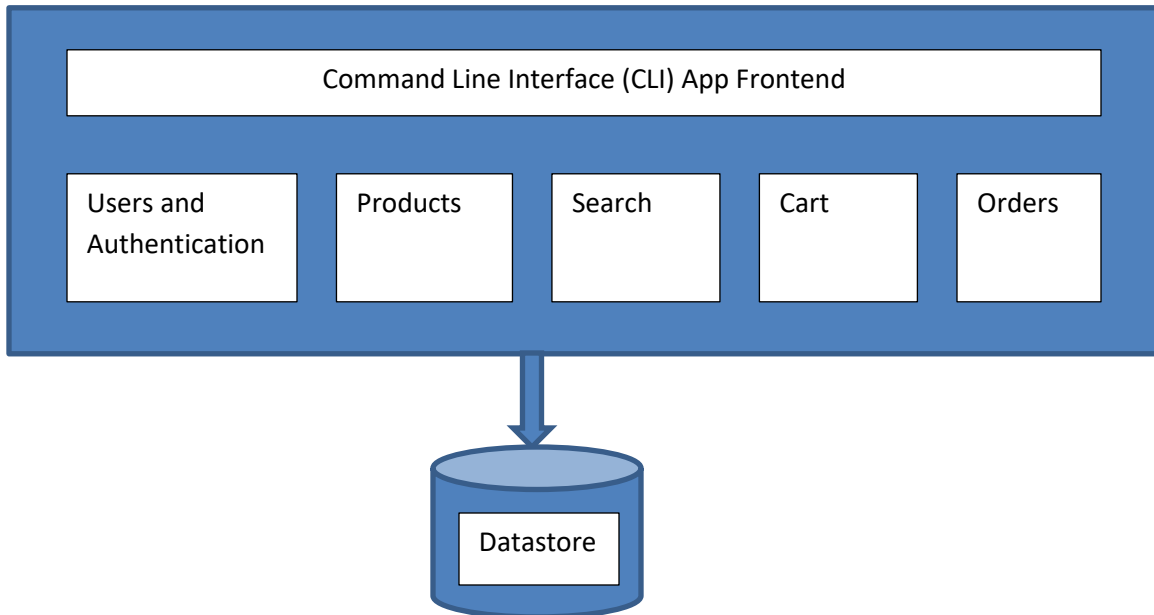


## 1. Application architecture

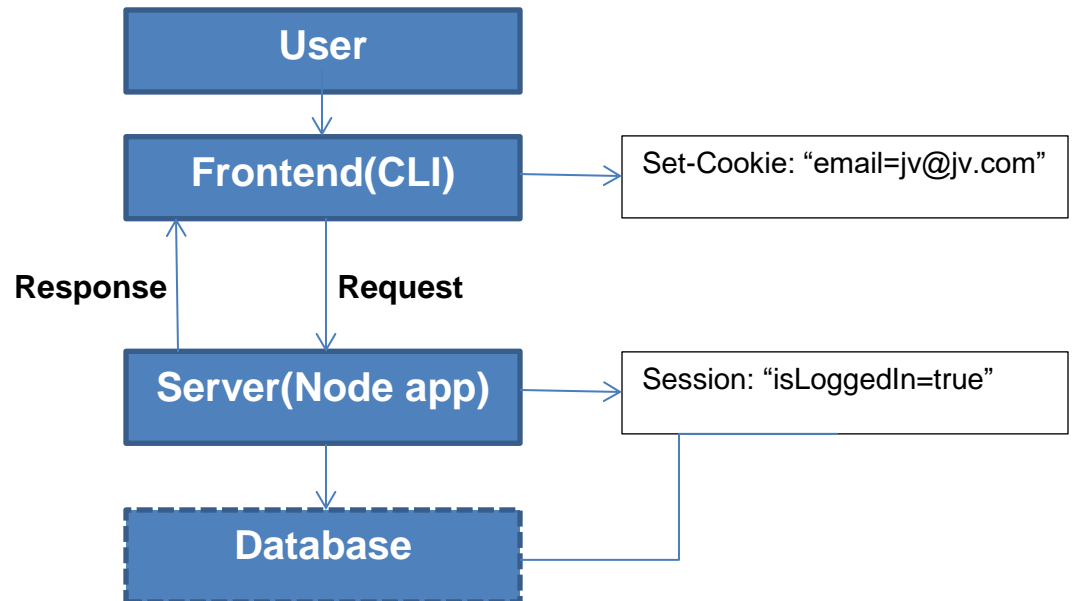
RestEasy application:



All the services are part of the same code base and run in a single process. Each service (Products, Users & Auth, Search, Cart, Orders etc.) is responsible for a specific portion of the application business logic. There is clear separation of services as implemented by the application; however every module is part of the same code base.

The application runtime is Node.js which is a JavaScript runtime. Express is the http server library used for the REST APIs. APIs are created with the standard Express HTTP verb style methods and routes. Express helps build the application in terms of HTTP requests and responses.

## 2. Application design:



### Application stack:

- Node.js – application runtime
- Express – http server library
- MongoDB - database
- Mongoose ODM – object-document mapping framework for Node.js and MongoDB

Node.js

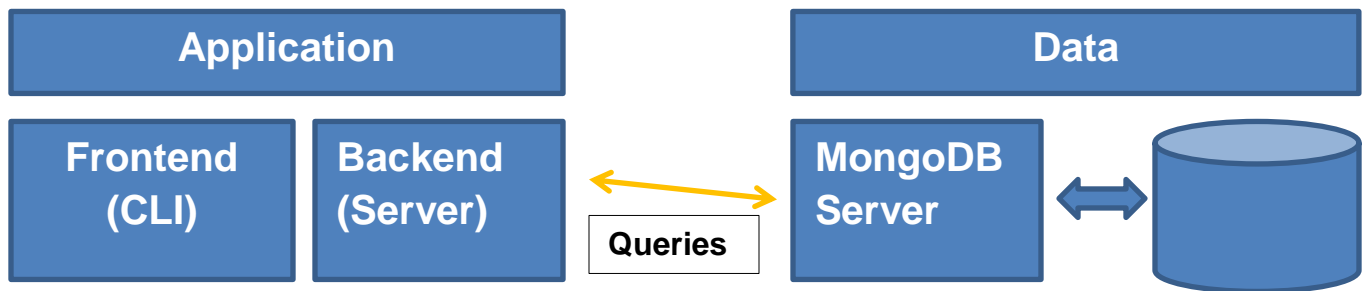
```
const doc = new MyModel({ name: 'foo', tags: ['bar'] });
await doc.save();
```

MongoDB

```
rs:PRIMARY> db.tests.findOne()
{
  "_id" : ObjectId("5c5c52fd785cdd08cedf1fbc"),
  "tags" : [
    "bar"
  ],
  "name" : "foo",
  "--v" : 0
}
rs:PRIMARY>
```

## Basic flow:

1. User sends the request from command-line by entering commands.
2. The client program running on CLI sends request to the backend APIs.
3. The server checks for user authentication and if user passes auth, user is logged in to the system and can issue further commands.
4. In the data layer, connection, data access, queries and results are handled by Mongoose ODM
5. On completion of logic execution response is sent back to the CLI
6. CLI renders the response accordingly



## DB Schema:

