Nimbus Architecture :

**Audit**: Framework has the ability to create an Audit Configuration for leaf parameters to generate audit history on

state change of the annotated parameter. @Audit is used for this.

**Entity state:** First need to create Core Domain entity. We can embed nested entities. We can perform actions on entity by listeners and later we can persist it.

**Domain Model:** For every domain entity, there are set of nested entities which we can consider as model. Model is represented with @Model.

**Config:** Actions are performed on entity using config information. General categorization are : Entity config, Rules Config and work flow Config. Set of config annotations are available to perform different calls.

**Rule Config:** Rule config file which contains details on how the state of the form variables are set through web sockets.

Ex: hello.drl

To know the framework where classes are configured, use application.yml file.

WorkFlow: This is managed with bpm process.

* Configurations like BPM engine, Mongo config, process config and web config will be added at config path.

**Major Technologies in Nimbus Framework:**

Three major technologies used in this frame work are:

1. BPM
2. Springboot
3. Mongo DB

**BPM**:

1. Entity lifecycle can be managed with BPM.
2. The framework provides the ability to back a business or view entity using a workflow. There might be scenarios where an entity needs to traverse through a series of business steps. Having a BPM defined through standard BPMN construct can help with configuring the business steps and will also provide a visual depiction of the bsuiness configuration.
3. If we want to execute a stateless BPM as a function, we need to do following.
   1. Create a BPMN process with a unique process id
   2. Define the config url with action as **\_process**,**fn** as **\_bpm** and **processId** as the name of the process id. Example **@Config(url="/p/patient:<!/.m/id!>/\_process?fn=\_bpm&processId=createcaseforpatient")**

**SpringBoot**:

1. Using spring boot, we can develop stand alone applications in faster manner using default tomcat server configured.

It’s the extended frame work of spring modular framework. To boot strap spring application written here we will write one java class with annotation @SpringBootApplication

**Mongo DB:**

1. MongoDB is a cross-platform and open-source document-oriented database, a kind of NoSQL database. As a NoSQL database, MongoDB shuns the relational database's table-based structure to adapt JSON-like documents that have dynamic schemas which it calls BSON.
2. Nimbus framework uses Mongo DB for dynamic db operations.
3. Data will be persisted with annotations @Domain,@Repo.

Ex Query: &criteria=testdsl.age.between(1,11)