**DESIGN DECISIONS**

We are proposing a new system DOCVER based on a Permissioned Private Blockchain for certificate registration and verification. This system is built using these technologies:

1. NodeJS
2. Angular v7.0
3. Rest API web service
4. Docker Infrastructure
5. Hyperledger Sawtooth v1.0

Here in this system is composed of simple web app consist of different pages. They are

Login page: A registered validator can get into the system through this login page. A register option is available for a new validator.By giving department name and office id, a new file is created and the validator is registered. A registered validator can login into the system by giving department name, office id and select the file which is created at the time of registration.

Docklist page: Validator can view a list of documents such as birth/death certificate and can use different features available they are Generate, Search and Verify. Generate option is used for create a new document. Search option helps to find already existed document. Verify option serves a feature to verify the authenticity of the document.

Birth registration page: This page is used for register a birth by giving a list of details. These details include Date of birth, name of child, name of father, name of mother, sex, address and registration date.

Death registration page: This page is used for register a death by giving a list of details. These details include Date of death, name of deceased, name of father/husband, name of mother, sex and address.

Search Page: This page allows to search a registered birth or death by giving a list details.

**Working :**

The address to be packaged inside the payload function, which would be further sent to the validator, through the proper REST API.

The validator, then propagate the same to the Transaction Processor for processing the transaction through protobuf serialisation.

**Frontend / Client side Design Decisions :**

1. Namespace is created using the hash of family name: ‘docver’
2. Encoded data/ Payload:It is encoding of the computation of json variables ie all the details entered in the form.
3. Address Generation: address =this.hash(this.FAMILY\_NAME).substr(0,6)+ userash + this.hash(docType).substr(0,6);

**Backend /Server side Design Decisions :**

1) The transaction received through validator is then received for

Processing

2) For the address and the message inside the payload is extracted – for the details stored and the family name is taken.

3)Then the decode data is separated and matched to produce a verification status which is returned citing success or failure. The apply function then further accept or reject(match).