The goal is to demonstrate a Hyperledger Fabric Application Management Suite on Salesforce.

We will use the following objects:

* Dapps\_\_Network\_\_c
* Dapps\_\_Participant\_\_c
* Dapps\_\_Channel\_\_c
* Dapps\_\_Channel\_Partcipant\_\_c
* Asset
* Dapps\_\_Registry\_\_c
* Dapps\_\_Registry\_Item\_\_c
* Dapps\_\_Transaction
* Business\_Network\_Event\_\_e
* Transaction\_Event\_\_e
* Channel\_\_e

The next thing we need to do create the infrastructure layer using a VM on a cloud platform using docker containers for the peers. Once we have the network established we can begin to build out the business network.To model the business network use the language below.When finished designing you network, deploy you composer application.When you have launched the application you can then manually send transactions to the network using the Salesforce Lightning Expirience. The interface is dynamically generated based on what Smart Contract you choose to use.

You can choose from ***RootStock, Ethereum, Tezos, Lisk***and ***Hyperledge****r*.

When you choose the contract it hits all the ledger when deployed. This is a IPFS driven engine that automatically graphs the state change and distributes it to nodes while resolving the state change in the other nodes.This is the way that companies will be able to interoperate and create a google like graph database of all the transactions that have occurred and an immutable web that versions the data, the exact content when hashed.

How can we do a demo in few days that completely integrates the Hyperledger data and the Salesforce data while running in when VM and multitenant. If we can create an environment when we are ha ving to buy nodes to cover the cot of users then we are in a really good shape. This means that the amount of users using the platform created the network. It will be an entire new ecosystem based on our applications and the infrastructure will be run by us. The ASIC cloud infrastfuture we will be providing will completely cover the cost due to specialized chip demand.