



Aplicació descentralitzada de monitoratge de salut a través de wearables

Alumne: Manuel Martínez Díaz
Màster Universitari en Enginyeria Informàtica

Consultor: Félix Freitag

8 de gener de 2020



[Aquesta obra està subjecta a una llicència de Reconeixement-NoComercial 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Aplicació descentralitzada de monitoratge de salut a través de wearables</i>
Nom de l'autor:	<i>Manuel Martínez Díaz</i>
Nom del consultor:	<i>Félix Freitag</i>
Data de lliurament (mm/aaaa):	<i>01/2020</i>
Àrea del Treball Final:	<i>Aplicacions i Sistemes Distribuïts</i>
Titulació:	<i>Màster Universitari en Enginyeria Informàtica</i>
Resum del Treball:	
<p>Els metges poden diagnosticar malalties comprovant les constants de la nostre salut, com la pressió arterial, la temperatura corporal, la saturació d'oxigen en sang o el cor amb un electrocardiograma. Existeixen dispositius IOTs, wearables, que mesuren algunes d'aquestes constants de salut i faciliten tindre un històric de dades de salut, que podrien ajudar als metges a detectar malalties, fins i tot, abans que els pacients presentin símptomes. Les dades han de complir el reglament de protecció de dades, com la GDPR i al treball s'ha analitzat la seva seguretat, la privadesa i la disponibilitat, tant per sistemes existents com pels nous basats en tecnologia de llibre major distribuït, DLT. S'argumenta la idoneïtat del framework Hyperledger Fabric per desplegar aplicacions distribuïdes, que requereixen tractar una gran quantitat de dades de forma privada i compartir-les, ja sigui amb el metge o de forma anònima amb investigadors. Tanmateix, s'explica el disseny d'una solució d'aplicació de monitoratge de constants de salut amb wearables, així com el desenvolupament d'una prova de concepte bàsica i un pla de proves que ha validat l'execució sobre Hyperledger Fabric. Una aplicació descentralitzada que requereixi tractar grans quantitats de dades, amb capacitat de privadesa i el complir la GDPR, és viable amb Hyperledger Fabric, que disposa d'un gran suport de desenvolupadors i independent de fabricants.</p>	

Abstract:

Doctors can diagnose diseases by checking our health constants, such as blood pressure, body temperature, blood oxygen saturation or heart condition with an electrocardiogram. There are IoTs devices, as wearables, that measure some of these health constants and provide a history of health data, which could help doctors to detect illness even before patients have symptoms. Data must comply with data protection regulations, such as GDPR, and at this work its security, privacy and availability have been analyzed, both for existing systems and new ones based on Distributed Ledger Technology (DLT). It's argued the suitability of the Hyperledger Fabric framework for deploying distributed applications, whose requires the processing of large amounts of data privately, and with the ability of sharing them, either with the doctor or anonymously with researchers. It's also explained the design of a wearables health constant monitoring application solution, as well as the development of a basic concept test and a test plan that has validated its execution on Hyperledger Fabric. A decentralized application that requires processing large amounts of data, with privacy and GDPR compliance, is viable with Hyperledger Fabric, which has great developer support and is vendor independent.

Paraules clau:

Sistemes distribuïts, DLT, aplicació descentralitzada, Hyperledger, wearables, salut, IoT.

Índex

1	Introducció.....	1
1.1	Context i justificació del Treball.....	1
1.2	Objectius del Treball.....	1
1.3	Enfocament i mètode seguit.....	2
1.4	Planificació del Treball.....	2
1.5	Breu sumari de productes obtinguts.....	4
1.6	Breu descripció dels altres capítols de la memòria.....	4
2	Fase de preparació.....	5
2.1	Estudi de la tecnologia i dispositius.....	5
2.1.1	Tecnologies de llibre major distribuït (DLT).....	5
2.1.1.1	Introducció als sistemes.....	5
2.1.1.2	Sistemes descentralitzats - DLT.....	9
2.1.2	Emmagatzematge distribuït.....	16
2.1.3	Compartició de dades selectiva i anònima.....	17
2.1.3.1	Compartició de dades selectiva.....	17
2.1.3.2	Compartició de dades anònimes.....	19
2.1.4	Dispositius <i>wearables</i>	20
2.2	Disseny de la solució.....	22
2.2.1	Requeriments de la solució.....	22
2.2.2	Criteris d'acceptació de la solució.....	22
2.2.3	Selecció de tecnologies i maquinari.....	22
3	Execució.....	26
3.1	Desenvolupament de la POC.....	26
3.1.1	Provisió d'entorn de desenvolupament.....	26
3.1.2	Desenvolupament Back-End.....	33
3.1.2.1	Estructura de dades.....	33
3.1.2.2	Mètodes.....	35
3.1.2.3	Chaincode / Smart Contract.....	36
3.1.2.4	Servidor API.....	37
3.1.3	Desenvolupament Front-End.....	38
3.2	Proves de la PoC.....	40
3.2.1	Definició del pla de proves.....	40
3.2.2	Criteris d'acceptació del pla de proves.....	41
3.2.3	Execució del pla de proves.....	42
3.2.4	Anàlisi de les proves.....	43
4	Conclusions.....	44
5	Bibliografia.....	45
6	Annexos.....	47
6.1	Codi del Chaincode / Smart Contract.....	47
6.2	Codi del servidor API.....	49
6.3	Codi del client.....	50
6.4	Codi del Pla de proves.....	54
6.5	Resultat d'instal·lar Hurley.....	57
6.6	Resultat de comanda 'hurl new'.....	60
6.7	Resultat de generar certificats necessaris per HLF.....	79
6.8	Resultats del pla de proves.....	85
6.8.1	Resultat de proves a la línia de comandes.....	85
6.8.2	Resultat de proves amb fitxer.....	87

Llista de figures

Figura 1: Diagrama de Gantt	3
Figura 2: Fases de l'arquitectura Hyperledger Fabric.....	12
Figura 3: Components principals d'arquitectura Hyperledger Fabric.....	13
Figura 4: Disseny d'alt nivell de la solució	25

Llista de taules

Taula 1: Fortaleses i debilitats d'un sistema a nivell lògic.....	7
Taula 2: Fortaleses i debilitats d'un sistema a nivell físic.....	7
Taula 3: Fortaleses i debilitats d'un sistema a nivell de governança.....	8
Taula 4: Dispositius compatibles amb Bluetooth Low Energy (Smart).....	21
Taula 5: Dispositius no identificats com compatibles amb Bluetooth Low Energy (Smart).....	21
Taula 6: Maquinari de la solució.....	26
Taula 7: Programari de la solució.....	27

1 Introducció

1.1 Context i justificació del Treball

El benestar i la qualitat de vida no només depenen d'una vida saludable, sinó també d'altres factors aliens al nostre control, com les malalties hereditàries o ambientals com la contaminació de l'aire.

Com a mesura de prevenció la Organització Mundial de la Salut [1] recomana la realització de revisions mèdiques periòdiques en funció de diversos factors com l'edat i el gènere, però també segons el teu historial mèdic. Amb la detecció primerenca de malalties, algunes es poden evitar, d'altres es poden curar i les incurables es poden tractar aplicant teràpies que millorin la qualitat de vida del pacient.

Algunes de les malalties que es podrien detectar fent un seguiment mèdic serien la diabetis i tensió arterial alta. Segons l'estudi Di@betes [2] de 5048 adults el 37,4% eren hipertensos sense diagnosticar i segons un estudi del International Society of Hypertension [3], està previst que l'any 2025 hi hagi 1,56 bilions d'adults amb hipertensió arterial, unes xifres que es podria reduir si s'apliquessin mesures proactives. Fer avenços en aquest sentit, no només permetria millorar la salut de la població, sinó que reduiria la despesa pública sanitària. Segons el Ministeri de Sanitat, serveis socials e igualtat [4], la despesa del sistema sanitari espanyola a l'any 2016 va ser de 66.678 milions d'euros, equivalent al 6,0% del PIB.

Així, la implementació de mecanismes que ajudin a realitzar un seguiment continu de forma senzilla i ràpida podria impactar de manera positiva:

- Reduint el número de persones afectades per alguna d'aquestes malalties cròniques i, per tant, millorar la seva qualitat de vida.
- Reduint les despeses públiques en aquesta matèria.
- Donant accés als metges i investigadors a les dades de constants de salut que actualment no estan disponibles.
- Recollint dades mèdiques individuals disponibles digitalment de forma segura i privada.

A l'actualitat, la informació de cada pacient només es desa en sistemes que estan connectats parcialment amb l'objectiu de recollir l'historial mèdic de cada persona. Aquests dades per sí soles no ens permeten conèixer més enllà de situacions individuals. És quan les ajuntem amb d'altres, que podem obtenir coneixement mitjançant el seu estudi i per tant, es converteixen en una font de valor per a la identificació d'indicadors que redueixin els temps de detecció d'algunes malalties.

En aquest sentit, la tecnologia pot esdevenir una aliada. L'ús de dispositius que recullen dades com les passes, freqüència cardíaca, pressió arterial o saturació d'oxigen en sang, ja són una realitat. Una aplicació descentralitzada que faci ús de tecnologies de llibre major distribuït (DLT), ens permet recollir aquestes dades, emmagatzemar-les de forma segura i, finalment, fer-les accessibles a tercers si així ho desitgem.

1.2 Objectius del Treball

Els objectius d'aquest treball són els següents:

- Adquirir coneixements de les tecnologies DLT, les aplicacions descentralitzades, la compartició de dades anonimitzades de forma selectiva i els dispositius connectats al cos que recullen dades. Es posarà èmfasi en el tractament de les dades, el seu valor, la seva seguretat i la seva privadesa.

- Dissenyar una solució per al cas proposat, d'una aplicació distribuïda sobre un sistema DLT que permeti el monitoratge de la salut amb dispositius connectats al cos, wearables.
- Desenvolupar una prova de concepte de la solució, amb els requisits mínims per tal de comprovar la seva viabilitat sobre el sistema DLT.

1.3 Enfocament i mètode seguit

Tot i que durant el màster s'han estudiat sistemes distribuïts, l'emmagatzematge de dades, la manca d'informació de projectes en l'àmbit descrit, fa necessària aprofundir tant en els sistemes actuals com els nous basats en tecnologies DLTs, per adquirir el coneixement que ens permeti dissenyar solucions adequades a cada projecte.

Per tal de comprovar la viabilitat de la solució, es requereix el desenvolupament d'una prova de concepte i l'execució d'un pla de proves. Encara que existeixen proves de concepte, que es poden utilitzar com a referència per aquest treball, es realitzarà una de nova per tal d'aprofundir en el coneixement del seu funcionament i adequar-la als requisits del projecte.

Per tant, es seguirà el següent ordre: planificació, estudi de tecnologia i dispositius, disseny de la solució, desenvolupament d'una PoC, execució de proves i elaboració de la memòria.

1.4 Planificació del Treball

Partint dels objectius s'identifiquen els estats intermedis del treball i les fites per tal d'assolir els objectius marcats. En aquest treball s'identifiquen 4 fases o línies de treball: Inici, preparació, execució i tancament, on s'inclouen les tasques següents:

- Fase d'inici:
 - Es descriu el Pla de Treball, amb el context i la justificació del treball, els seus objectius, l'abast, els riscos i la planificació de tasques per complir l'abast.
 - La documentació generada en aquesta fase és el lliurable de la PAC1.
- Fase de preparació:
 - Estudi de la tecnologia i dispositius: es cerca informació i s'estudien els sistemes distribuïts basats en tecnologia DLT, com obtenir dades de dispositius *wearables* i com emmagatzemar-les de forma distribuïda, necessari pel disseny de la solució.
 - Disseny de la solució: es descriu els requeriments de la solució, els seus criteris d'acceptació i la selecció de tecnologies i maquinari, que ens permet conèixer quines eines desplegar al nostre entorn de treball. També es descriu el disseny de la solució amb els seus actors i rols, necessari pel desenvolupament.
 - La documentació generada en aquesta fase és correspon al lliurable de la PAC2.
- Fase d'execució:
 - Desenvolupament de la PoC: es descriu el desplegament de l'entorn de treball, es detalla la solució a baix nivell amb la definició de dades, necessària pel desenvolupament del *Back-End*, el *Front-End* i el *chaincode* / smart contract.
 - Proves de la PoC: es defineix un pla de proves i els resultats que s'esperen, els quals s'executen sobre la PoC per tal d'analitzar el seu resultat, necessaris per extreure conclusions.
 - Conclusions i treballs futurs: es descriuen les conclusions del treball, tenint en compte els coneixements i PoC realitzada, així com, les futures línies de treball a seguir.
 - La documentació generada en aquesta fase és el lliurable de la PAC3.

- Fase de tancament:
 - Elaboració de la memòria: es documenta la memòria del treball on s'inclou informació rellevant al disseny i el seu desenvolupament.
 - Elaboració de la presentació: es documenta la presentació, així com el vídeo on s'explica la presentació. També s'inclou l'informe d'autoavaluació i el codi desenvolupat.
 - La documentació generada en aquesta fase és correspon al Lliurament Final.

A partir d'aquestes tasques es determinen les fites i els lliurables que recullen el realitzat, que serveixen per validar la correcte direcció del treball i el seguiment de les tasques. Aquest treball consta de quatre lliurables, corresponents a la PAC1, PAC2, PAC3 i un lliurament final:

- PAC1: Descripció del treball, objectius i planificació.
- PAC2: Descripció de tecnologies i disseny de la solució.
- PAC3: Desenvolupament d'una PoC, proves de la PoC i resultats.
- Lliurament final: Memòria del treball i presentació del treball.

Posteriorment i tenint present les dependències de les tasques, l'esforç estimat i les fites, es realitza una planificació temporal amb un diagrama de Gantt, veure Figura 2: Fases de l'arquitectura Hyperledger Fabric. Es considera que com a recurs del treball només hi ha una persona a temps parcial sense dies festius, per tant les tasques es realitzen de forma seqüencial, excepte algunes que requereixin simultàniament degut a la seva interacció, com les tasques de desenvolupament de la PoC. La planificació s'ha realitzat amb l'eina [5] que permet gestionar les tasques del projecte i mostrar una vista Gantt.

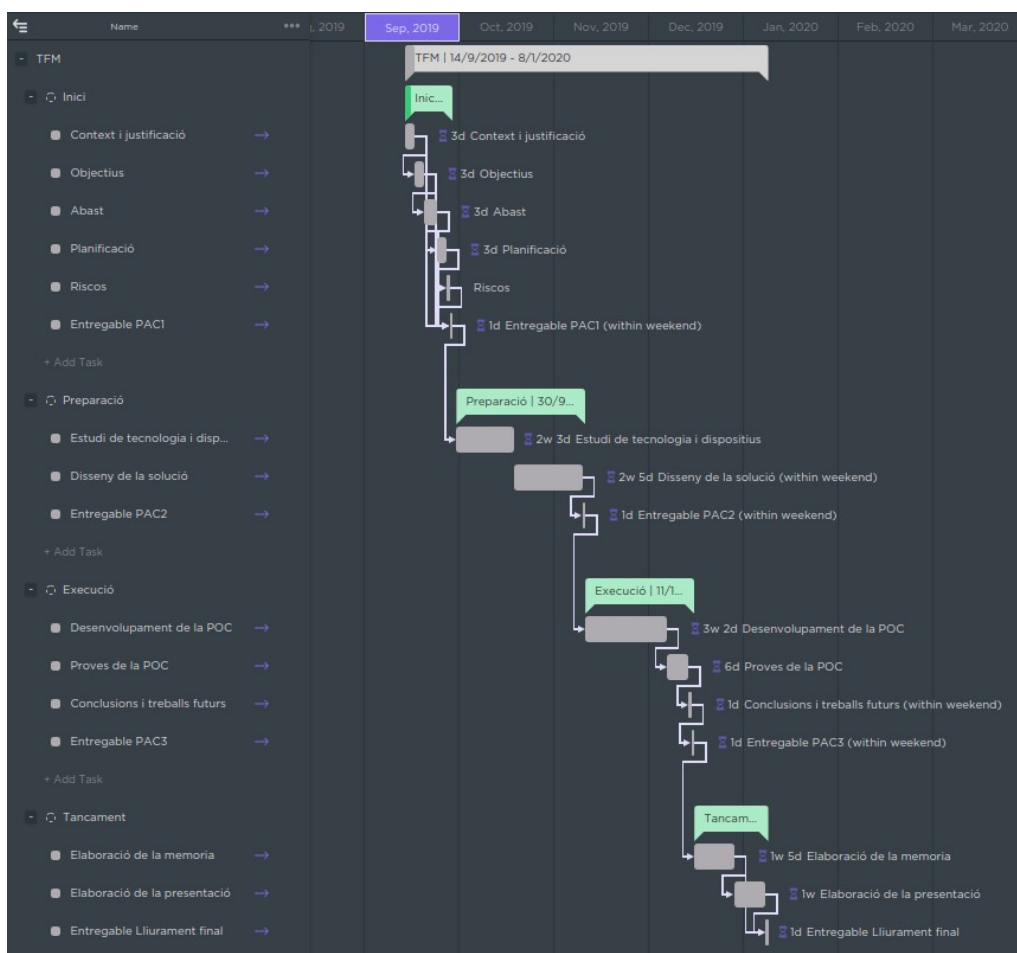


Figura 1: Diagrama de Gantt - Font: elaboració pròpia

1.5 Breu sumari de productes obtinguts

- Memòria, és el present document on es descriu tota la documentació del treball.
- PACs, són els documents on s'explica l'avanç del treball respecte al Pla de Treball.
- PoC de l'aplicació, és el programari back-end que s'executa sobre Hyperledger Fabric.
- Chaincode: és el programari que conté la lògica del negoci, conté la definició dels actius (model amb la seva estructura de dades) i els controladors amb les instruccions permeses per modificar els actius. S'executaran a la plataforma Hyperledger.

1.6 Breu descripció dels altres capítols de la memòria

A continuació es descriu breument el contingut dels capítols següents:

- Fase de preparació: es proporciona una visió global de la tecnologia DLT, els seus dispositius wearables, així com el disseny de la solució de l'aplicació distribuïda proposada al treball.
- Fase d'execució: es descriu el desplegament de l'entorn de treball i el detall de la solució. També s'explica el desenvolupament d'una PoC mínima que permeti validar i executar un pla de proves.
- Conclusions: S'expliquen les conclusions d'aquest treball amb els problemes detectats i treballs futurs.
- Bibliografia: Fonts consultades per a la realització del treball.
- Annexes: S'inclouen informació complementaria, com les guies d'instal·lació de l'entorn de desenvolupament, resultat detallat del pla de proves o codi del programari.

2 Fase de preparació

A la fase de preparació primer de tot s'investigarán els sistemes distribuïts basats en tecnologia DLT, així com els dispositius *wearables* amb els quals obtindrem dades i finalment com emmagatzemar-les de forma distribuïda. Un cop tenim una base teòrica de les eines disponibles, es prendran els requeriments, i tenint-los en compte, dissenyarem una solució.

2.1 Estudi de la tecnologia i dispositius

Alhora d'estudiar les tecnologia DLT identificarem les diferències de les solucions existents, tipus de DLTs, les seves fortaleses i debilitats. També es profunditzarà com s'emmagatzemen les dades, com es tracta la seva privacitat i la forma de compartir de forma selectiva.

També s'estudiarán els dispositius IOTs, com *wearables*, i la seva interoperabilitat amb sistemes DLTs per tal d'emmagatzemar les dades.

2.1.1 Tecnologies de llibre major distribuït (DLT)

Les tecnologies de llibre major distribuït (DLT), són un conjunt de tecnologies que permeten crear un sistema amb un registre de dades i transaccions consensuades que es sincronitza, replica i comparteix a través d'una xarxa distribuïda de nodes [6].

La funcionalitat principal de les DLT és connectar ordenadors que poden estar geogràficament distribuïts, que formen una xarxa en la que comparteixen dades, que validaran i registraran de manera consensuada sense necessitat d'un tercer actor que actuï com administrador del sistema i validi el registre de dades.

Per conèixer que ens aporta les DLT, estudiarem els sistemes existents, tant els que porten molt de temps utilitzant-se i són prou madurs per entorns productius com els més recents i els que encara estan en desenvolupament. Conèixer el passat, present i futur immediat d'aquests sistemes, ens permetrà decidir quina solució encaixa més als nostres requeriments o si cal optar per solucions híbrides.

2.1.1.1 Introducció als sistemes

Per tal de dissenyar la solució d'aquest treball, és convenient conèixer els sistemes existents, no només els que utilitzen tecnologia DLT, d'aquesta forma tindrem una visió general que ens permetrà justificar la elecció de tecnologies respecte a solucions més madures i provades.

2.1.1.1.1 Tipus de sistemes

Primer de tot, els sistemes es classifiquen en dos grans tipus:

- Centralitzats: un sistema amb un node central que processa totes les transaccions dels clients.

- Descentralitzats: dos o més nodes interconnectats entre sí, els quals processen les mateixes transaccions i consensuen l'estat del sistema. Cal tenir en compte que el grau de distribució del disseny lògic, físic o governança pot afectar negativament entre elles.

2.1.1.1.2 Com dissenyar un sistema

Per tal de dissenyar un sistema, és necessari fer-se les següents preguntes, que volem que faci el nostre sistema? Com ho farà? Quins elements tindrà? La primera pregunta fa referència als requeriments del projecte i les dues següents al disseny que ha de complir els requeriments. De les dues preguntes de disseny, per una banda tenim el com farà el sistema per complir els requeriments, on tractarem la lògica i com es governa el sistema i per un altra banda, quins elements tindrà on tractarem els dispositius necessaris i com s'interconnectaran per suportar la lògica i governança del sistema. Per tant, utilitzarem l'estratègia de separar el disseny en 3 components [7], per tal que el nostre sistema compleixi els requeriments del projecte:

- Disseny lògic: com s'estructuren les dades en els equips del sistema i quins fluxos.
- Disseny físic: quants equips físics i quina és la seva connexió.
- Governança: quantes persones i/o organitzacions controlen els equips del sistema.

Cada un dels 3 component del disseny podria ser de tipus centralitzat o descentralitzat, amb independència del tipus que siguin els altres dos components. Encara que un sistema amb una governança centralitzada, tendeixen per simplicitat a que el disseny físic i/o lògic també sigui centralitzat. Per tant, són possibles diverses combinacions com el cas de BitTorrent que amb un disseny lògic centralitzat el seu disseny físic és descentralitzat. És important tenir present que no sempre un sistema que es promociona com a descentralitzat, ho sigui totalment, i que es poden assolir diversos nivells de centralització/descentralització per a cada component que afectaran al compliment dels requisits.

2.1.1.1.3 Fortaleses i debilitats segons disseny i tipus de sistema

Per conèixer quines possibilitats ens donarà un sistema, segons el seu disseny i el tipus de sistema esmenats prèviament, analitzarem i descriurem les fortaleses i debilitats dels tres components que utilitzarem per fer el disseny respecte al tipus de (des)centralització [8], [9].

Disseny Lògic		
Centralitzat	Fortaleza	<ul style="list-style-type: none"> • La consistència de dades és elevada perquè per a cada transacció existeix només un únic registre. Assegura que tot el sistema treballa amb el mateix objectiu, al estar centralitzat. • La seguretat del sistema és de fàcil gestió perquè està centralitzada en un únic equip.
	Debilitat	<ul style="list-style-type: none"> • La privacitat de les dades es pot veure compromesa degut a que les dades passen a través d'un únic punt i poden ser recol·lectades fàcilment en cas que hi accedeixi un actor no autoritzat.
Descentralitzat	Fortaleza	<ul style="list-style-type: none"> • La privacitat augmenta si el registre de dades es replica parcialment entre els nodes o només conté les comprovacions de les dades, i en cas d'accés no autoritzat només veurà les dades d'aquell equip. Quants més nodes més privacitat.

	Debilitat	<ul style="list-style-type: none"> • La privacitat pot veure's disminuïda en cas que les dades es repliquin a tots els nodes, encara que siguin privades, enfront d'un accés no autoritzat a qualsevol dels nodes. • La consistència de dades es pot veure reduïda degut al temps elevat per a la sincronització de les dades del sistema. • Cost elevat de desenvolupament de programari degut a la seva complexitat i manteniment de la coordinació entre els nodes. • El rendiment del sistema pot disminuir degut a la sincronització. Quants més nodes del sistema, menys rellevància tenen les transaccions que cada node envia al sistema.
--	-----------	---

Taula 1: Fortaleses i debilitats d'un sistema a nivell lògic

Disseny Físic		
Centralitzat	Fortaleza	<ul style="list-style-type: none"> • Alta eficiència de l'equip central i gran fiabilitat del sistema, degut a que no depèn d'altres equips. S'elimina el temps de sincronització de dades i l'espai d'emmagatzematge de replicar-les. La gestió i manteniment és senzilla al estar centralitzat.
	Debilitat	<ul style="list-style-type: none"> • La disponibilitat es pot veure afectada degut a que totes les transaccions es concentren en un únic equip i pot provocar saturació. • La tolerància de partició en cas de fallida és nul·la perquè quan deixa de funcionar l'equip central es perd l'accés al servei. Segons el mecanisme de copia de seguretat emprat pot suposar pèrdua de dades. • La escalabilitat és reduïda perquè suposaria actualitzar l'equipament. • Cost de la solució elevat degut a equipament central ha d'esser específic, enlloc d'un equip estàndard.
Descentralitzat	Fortaleza	<ul style="list-style-type: none"> • La disponibilitat es garanteix amb replicació de les dades. • La tolerància a partició en cas de fallida d'un equip només afectarà a aquella part del sistema. • La escalabilitat es pot aconseguir amb clústers d'equips estàndards i econòmics enlloc de servidors potents i més cars. Una millor utilització de recursos perquè permet balancejar la càrrega entre equips, evitant saturacions de connexions i processament. • Millor rendiment en alguns casos al apropar les dades a l'usuari.
	Debilitat	<ul style="list-style-type: none"> • La consistència de les dades es veu afectada en cas de l'existència de repliques de les dades. • Cost elevat degut a recursos addicionals i el seu manteniment, com l'espai d'emmagatzematge de dades als nodes. • El risc de seguretat i el seu cost augmenta degut a l'existència d'equips en ubicacions diferents.

Taula 2: Fortaleses i debilitats d'un sistema a nivell físic

Governança		
Centralitzat	Fortaleza	<ul style="list-style-type: none"> Facilita garantir el compliment de les regles al existir un únic control central.
	Debilitat	<ul style="list-style-type: none"> L'existència d'un únic actor que modifiqui les normes i/o dades sense coneixement de la resta d'usuaris en benefici propi.
Descentralitzat	Fortaleza	<ul style="list-style-type: none"> Està repartida entre un número de nodes i dificulta la censura o canvis en benefici propi perquè caldria posar d'acord a la resta de nodes.
	Debilitat	<ul style="list-style-type: none"> Sense un control global que revisi el comportament dels nodes pot suposar un problema en la coordinació dels mateixos. Requereix d'un desenvolupament complex i madur. Pot disminuir el rendiment global del sistema, perquè els nodes han d'arribar a un consens en la validació de transaccions. Duplicació de càrrega de treball per validar transaccions i espai d'emmagatzematge.

Taula 3: Fortaleses i debilitats d'un sistema a nivell de governança

Amb aquestes dades podem fer un resum del que ens permet la descentralització i el que podem perdre respecte a dissenys més centralitzats.

Què millora un sistema descentralització enfront d'un centralitzat? (Lògic) Privacitat de dades on es repliquen parcialment o només comprovacions de les mateixes.

- (Físic) Disponibilitat del sistema.
- (Físic) Tolerància a partició del sistema.
- (Físic) Escalabilitat.
- (Governança) Dificulta la censura i corrupció.

Què perdem amb un sistema descentralitzat?

- (Lògic) Privacitat de dades si es repliquen a tots els nodes.
- (Lògic-Físic-Governança) Rendiment del sistema.
- (Lògic-Governança) Cost de desenvolupament elevat.
- (Lògic) Consistència de dades inferior.
- (Físic-Governança) Cost elevat de recursos i el seu manteniment.

A continuació, aprofundim a les característiques dels sistemes que s'han tractat per tal de veure alternatives:

La privacitat de dades, és un aspecte molt important sobretot amb dades sensibles com podem ser les de salut, ens trobem que la descentralització amb rèpliques o registre de dades compartit, garanteix una alta disponibilitat i tolerància a partició del sistema, però com les dades estan replicades els possibles atacants podrien accedir a gran part o a totes les dades només accedint a un equip. En aquest cas caldria analitzar solucions per millorar la privacitat, com que el registre de dades estigui limitat a uns equips, que sigui privat i només accessible amb permisos, o que els usuaris tinguin les dades localment i al registre de dades compartit es

dessin els valors que permetin comprovar la veracitat de les dades dels usuaris. Algunes d'aquestes alternatives, tot i que millorarien la privacitat, consistència de dades i rendiment del sistema, també faria que el sistema sigui més centralitzat a nivell lògic, físic i de governança amb les seves debilitats.

La consistència de dades en sistemes distribuïts es veu afectada per la necessitat de consensuar les transaccions pels equips validadors i la seva sincronització a les diferents rèpliques de dades. Això implica, respectivament, un cost de recursos computacionals i temps que repercuteix negativament en el rendiment del sistema. Per tal de resoldre-ho existeixen solucions com una consistència de dades eventual que no garanteix que les dades disponibles siguin les últimes.

La escalabilitat i el rendiment d'un sistema distribuït es veu limitat pel disseny lògic, físic i governança, per exemple tenim a Bitcoin o Ethereum que encara que no són totalment distribuïts que al tindre un registre de dades compartit, arriben a un màxim de 15 transaccions per segon. En sistemes amb un registre de dades compartit, existeixen solucions com reduir el número d'equips que validen i registren les transaccions, reduint els recursos emprats i temps, a costa d'una governança centralitzada en un número determinat d'equips, els quals podrien ser manipulats o posar-se d'acord per un benefici en comú. Aquesta solució és adient en entorns professionals on diverses empreses han de compartir dades. També es poden crear sistemes paral·lels encarregats d'alleugerar la càrrega del sistema principal, mitjançant l'agrupació de transaccions perquè posteriorment es consolidaran al sistema principal.

En un sistema de monitoratge on es rebin dades de wearables, podem esperar que es generin una gran quantitat de dades de forma constant, que necessitem de processar i emmagatzemar.

Com a conclusió de l'anàlisi de les fortaleces i debilitats dels sistemes distribuïts, tenim que aquests sistemes ens aporten beneficis però en detriment de limitacions importants que caldrà considerar alhora de dissenyar la solució, per tal d'assolir un compromís per complir els requeriments del projecte.

2.1.1.2 Sistemes descentralitzats - DLT

En sistemes amb tecnologia DLT, els recursos de dades, processament i governança del sistema, estan repartits entre equips del sistema els quals treballen per tal d'aconseguir uns objectius globals. Aquest objectius poden ser un sistema amb alta disponibilitat de dades que estarà replicat i no un únic equip, o obtenir un sistema amb una gran tolerància a la partició de la xarxa degut a que la governança del sistema no està centralitzada i en cas de talls de comunicació podrà seguir treballant. Un altre objectiu pot ser repartir la càrrega de treball per tal d'incrementar el rendiment de processament o d'ample de banda d'un sistema que estigui limitat per les característiques d'un únic equip. Encara que una excessiva distribució dels recursos del sistema o la seva governança pot afectar negativament al sistema.

2.1.1.2.1 Components bàsics d'un sistema DLT

- Registre de dades compartit: és on s'emmagatzemen les transaccions validades pels nodes i cada node del sistema té una còpia. Les còpies als nodes han d'estar sincronitzades per tal que treballin amb l'estat correcte de les dades.
- Nodes: s'ón els elements del sistema que poden tindre diferents rols, com els descrits tot seguit. Rol d'administrador que s'encarrega de controlar l'accés al sistema i gestionar-ho. Rol peticionari que proposa actualitzacions al registre de dades. Rol validador que valida les transaccions proposades. Rol d'auditor que només pot llegir el registre de dades sense fer cap proposta.
- Criptografia: s'utilitza per la gravació, manteniment i securització de les transaccions entre nodes. Això permet autenticar els nodes, que els nodes validin transaccions proposades i facilita el consens per actualitzar el registre de dades compartit. Cada node pot crear transaccions d'una forma segura sense la necessitat d'una autoritat central.
- Consens: s'utilitzen mecanismes de consens descentralitzat enlloc de confiança perquè els nodes validin transaccions i acordin afegir-les al registre de dades compartit. Els mecanismes de consens més comuns són PoW (*Proof-of-Work*), PoS (*Proof-of-Stake*), DPOS (*Delegated Proof-of-Stake*), *Byzantine Fault Tolerance* o DAG (*Directed Acyclic Graphs*).

2.1.1.2.2 Què aporta els DLT?

Els sistemes més estesos són els centralitzats o distribuïts que no utilitzen un registre de dades compartit i sincronitzat degut a la seva recent incorporació. Però que ens aporten les DLT respecte als sistemes actuals?

- Disponibilitat: Al disposar cada node del sistema el registre de dades compartit, existeix una alta disponibilitat. Podríem, fins i tot, plantejar-nos prescindir de plans de recuperació del sistema.
- Immutabilitat: Una vegada s'afegeixen les dades al registre compartit no es poden modificar i es pot verificar que no han sigut modificades.
- Transparència: les transaccions són visibles pels nodes participants per tal que les validin i no hi ha una única autoritat que governa el sistema i valida les transaccions.
- Seguretat i resistència a atacs: S'utilitza la criptografia amb claus públiques i mecanismes de consens per tal de donar seguretat i confiança als usuaris del sistema, en front d'atacs al sistema.
- Automatitzacions i contractes: Es poden afegir al registre de dades, codi de programa amb condicions i accions, que actuaran com a contractes digital. Quan les condicions es compleixen de forma automàtica s'executaran les accions programades, reduint gestions i millorant temps i confiança dels usuaris del contracte digital.
- Organitzacions autònomes: Permeten que els nodes participin en el sistema realitzant les mateixes tasques que la resta, sense una autoritat central que la governi perquè aquesta tasca està repartida entre els nodes.

2.1.1.2.3 Que cal superar?

- Tecnologia en desenvolupament: les DLT estan en una etapa de desenvolupament on encara requereixen temps per arribar a implementacions robustes i fiables, per tant la seva adopció és lenta. Tant Ethereum com Hyperledger són les més consolidades.
- Escalabilitat: les solucions DLT actuals tenen limitacions de volum de transaccions per segon, encara que s'estan desenvolupant solucions, algunes d'elles sacrificant transparència i governança.
- Latència: la mida del registre compartit creix. L'augment de nodes pot impactar en el temps de resposta entre nodes i el temps de validació és gran.
- Seguretat: poden existir vulnerabilitats que encara es desconeixen i que en un futur permetin visualitzar dades encriptades. En cas de registres privats, la seguretat de xarxa i encriptació del sistema és la del node més vulnerable, degut a que tota la informació està a tots els nodes.
- Integració: existeixen diferents tipus de DLT i sistemes actuals que s'haurien de poder comunicar per tal de facilitar la integració en les infraestructures actuals i posterior migració en cas que es requereixi.
- Privacitat: les dades que es distribueixen a través dels nodes del sistema poden tindre informació sensible i per tant seran visibles pels nodes que validen transaccions. També caldrà que els sistemes compleixin les diferents regulacions segons la ubicació dels nodes.
- Confidencialitat: si el registre de dades té informació confidencial, existeix el risc que un atacant accedeixi a les dades.
- Legals: segons el tipus de transaccions s'ha de complir diferents normatives, la llei de protecció de dades, la GDPR [10] amb el dret a l'oblit o la validesa d'un contracte digital en l'àmbit notarial o judicial i en l'àmbit bancari tenim la CNMV.

2.1.1.2.4 Tipus de DLTs

Podem classificar els DLT segons la seva estructura de dades en BlockChain, DAG i d'altres com HoloChain.

2.1.1.2.4.1 BlockChain

Aquest sistema utilitza una estructura de blocs on cada bloc està identificat per un únic *hash* relacionat amb el bloc anterior, això fa impossible modificar les dades sense que invalida els següents blocs i ens dona la seva característica de Immutabilitat.

Existeixen nodes participants, anomenats miners, que veuen les transaccions del sistema i les validen. El mètode de consens es basa en la selecció dinàmica de líder, mentre que el mètode de selecció/ponderació, que defineix l'ordre i la validesa, utilitzat principalment és PoW encara que d'altres com EOS utilitza DPOS. Degut a que les transaccions son vistes pels miners es considera un sistema transparent.

El seu tipus d'accés a la xarxa pot ser públic (*permissionless*) o privat (*permissioned*) només accessibles amb permisos.

Existeixen *Smart Contracts* en Ethereum i Chaicode en Hperledger per automatitzar transaccions sempre que es compleixin les condicions incloses al *Smart Contract*. Degut a les característiques de BlockChain fa que els principals casos d'us estiguin relacionats amb la necessitat d'enregistrar transaccions o esdeveniments, seguiment d'actius o votacions. El

rendiment de Blockchain és d'unes 5tps (transaccions per segon) a Bitcoin, 15tps a Ethereum i fins a 100000tps en EOS.

Com exemple d'implementació de Blockchain, tenim Bitcoin, Ethereum o EOS. El disseny de Bitcoin és de lògica centralitzada perquè el sistema es comporta com un únic equip (l'estat del sistema és únic perquè s'acorda entre els nodes), és de disseny físic descentralitzat perquè no hi ha un únic punt de fallida i d'una governança descentralitzada perquè no hi ha un control central.

A continuació aprofundim en Hyperledger, i concretament en **Hyperledger Fabric** que és un framework de nivell empresarial per desenvolupar sistemes distribuïts per a Blockchains privades o també anomenades permissionades.

Hyperledger Fabric

Permet executar aplicacions distribuïdes en un sistema blockchain, és modular i permet auditar l'històric de transaccions perquè el registre de dades només permet afegir dades.

L'arquitectura de Fabric és separa en tres fases 'execute-order-validate', veure Figura 2: Fases de l'arquitectura Hyperledger Fabric, on el flux de les transaccions al estar separat en fases possibilita que s'executin en diferents identitats del sistema.

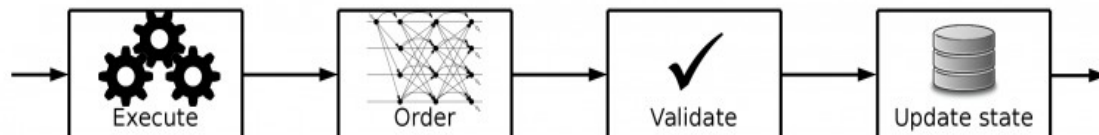


Figura 2: Fases de l'arquitectura Hyperledger Fabric

Font: <https://www.ibm.com/blogs/research/2018/02/architecture-hyperledger-fabric/>

Una aplicació distribuïda de Fabric consisteix en dues parts, *Smart contract* i *Endorsement policy*:

1. *Smart contract*, anomenat *chaincode* dins Hyperledger, és on està la lògica del negoci que s'executa durant la fase d'execució. Es defineixen els actius i les instruccions de transaccions per modificar l'actiu. També existeixen *system chaincodes* per gestionar el sistema BC i el seu manteniment.
2. *Endorsement policy*, política d'aprovació, la qual s'avalua en la fase de validació. Aquestes polítiques només les poden modificar l'administrador del sistema.

Una blockchain Fabric consisteix en diferents nodes o peers que formen la xarxa, els qual poden tindre els rols de *Client*, *Peer* o *Orderer*, veure Figura 3: Components principals d'arquitectura Hyperledger Fabric. Cadascun dels nodes té una identitat, gestionada per el module Membership Service Provider (MSP).

Tipus de nodes en Hyperledger Fabric:

- **Client**, que envien transaccions de proposta per a la seva execució, ajuden a organitzar la fase d'execució i informen de les transaccions perquè s'ordenin.
- **Peers**, que executen i validen les transaccions proposades. El chaincode de la transacció determina quins nodes l'executen i la validen. Aquest nodes s'anomenen *Endorsing peers* o *Endorsers*.
- **Ordering Service Nodes (OSN) o orderers**, que estableixen l'ordre de totes les transaccions en la BC Fabric.

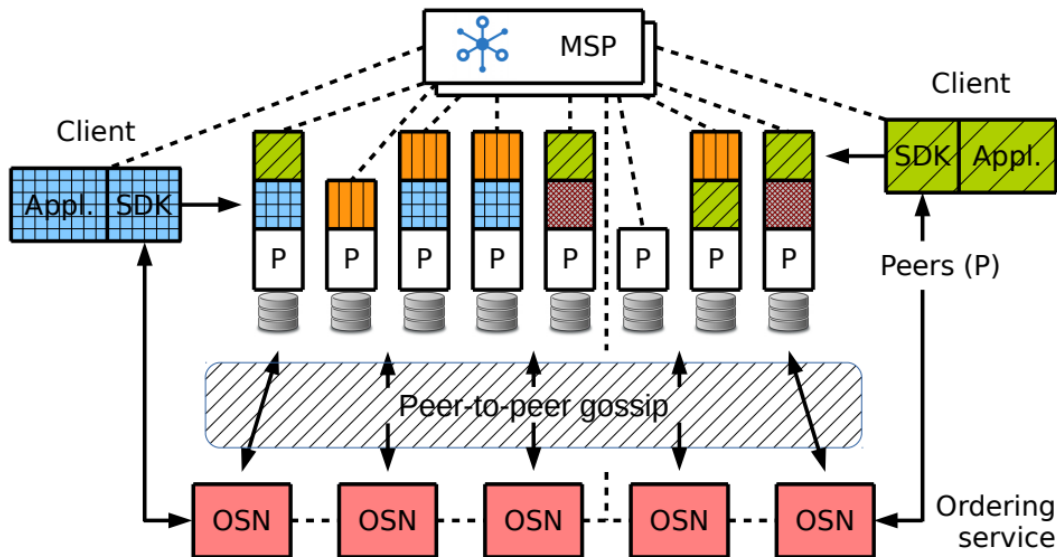


Figura 3: Components principals d'arquitectura Hyperledger Fabric

Font: <https://arxiv.org/pdf/1801.10228.pdf>

Components principals d'arquitectura Hyperledger Fabric [11]:

- **Membership Service:** S'encarrega de registrar les identitats del nodes i està format per un component a cada node, on pot autenticar transaccions, verificar la integritat de les transaccions, signar i validar i autenticar altres operacions BC. Per defecte s'implementa mètodes de l'estàndard PKI per autenticació basada en signatures digitals i pot utilitzar certificats comercials. Fabric aporta un generador de certificats, Fabric-CA.
- **Ordering Service:** gestiona múltiples canals i a cadascun dona els serveis següents:
 - *Atomic broadcast*, per establir l'ordre de les transaccions, implementar la difusió i servir crides.
 - *Reconfiguration of a channel*, per modificar el canal enviant per difusió una transacció per actualitzar la configuració.
 - *Access control*, quan el servei d'ordenació es considera fiable, permet restringir transaccions de difusió i rebre blocs de determinats nodes (*clients* o *peers*).

Al primer block de cada canal es defineixen les propietats del Ordering Service. Existeixen diferents nodes (OSNs) que es comuniquen a través un canal de sistema. El servei *Atomic broadcast* es dona amb Apache Kafka, encara que hi han dues més disponibles, Solo que és centralitzat que s'utilitza per desenvolupament i un altre que és una PoC basada en *BFT-SMaRT*.

- **Peer Gossip**; degut a que l'ordenació i validació estan desacoblades, es focalitza en informar el més eficient possible a tots els nodes d'un canals del resultat de l'execució per a la seva validació. S'utilitza el protocol *gossip data dissemination* encarregat de les següents funcions:
 - Gestiona el descobriment de *peers* i canals dels membres.
 - Disseminar dades a tots els *peers* d'un mateix canal, perquè tinguin les dades correctes.
 - Agilitzar la incorporació de nous *peers* permetent intercanvi directe de dades entre els *peers*.
- **Ledger**, és el registre de dades de totes les transaccions d'estat, que són realitzades com a resultat de cridar un chaincode pels participants del sistema, també ens podem referir a aquesta acció com a transaccions. Cada transacció genera un conjunt de parelles key-values que s'afegix al registre de dades. Aquest registre està compost d'una blockchain per emmagatzemar de forma immutable una seqüència de registres en blocs, i una base de dades amb el estat actual de les dades. Hi ha un registre de dades per canal i cada peer manté una copia per a cadascun dels canals que és membre.
- **Chaincode execution**, és realitza en un entorn aïllat de la resta de peers i chaincodes que s'executin, que suporten llenguatges com Go, Java i Node. La comunicació entre el chaincode i el peer és a través de missatges gRPC. Per altre banda, els chaincode de sistemes s'executen als peers i serveixen per modificar la configuració d'un canal, que inclou, les definicions dels nodes participants, les adreces de xarxa dels OSNs, configuració de la implementació de consens i el servei d'ordenació, regles que regulen l'accés a les operacions del servei d'ordenació (difusió i lliurament) i regles que regulen la forma en que la configuració del canal es pot modificar.

Tots els actors del sistema, com les aplicacions de client, els administradors i peers entre d'altres, s'han de registrar en un servei anomenat Membership Service Provider (MSP), perquè totes les transaccions de la xarxa són validades i signades. Cadascun dels actors tindrà una identitat digital encapsulada en un certificat digital X.509. Es disposa del Servei Fabric CA que genera dinàmicament certificats.

2.1.1.2.4.2 **Directed Acyclic Graph (DAG)**

L'estructura de dades es gestiona a través de gràfics acíclics dirigits, que estan formats per nodes i connexions, els nodes són les transaccions que fan referència a dos transaccions anteriors i relacionat amb una nova transacció que dona una ordenació a la seqüència de transaccions. El procés de validació és donat per la mateixa transacció, on la seva validació és donada pel número de transaccions que tenen darrera. El consens s'arriba pel consens general de les transaccions i no de les validacions de les transaccions.

La validació es realitza pels membres del sistema, amb una prova que té un cost computacional petit i sense necessitat de miners i les seves comissions.

El seu tipus d'accés a la xarxa pot ser públic (*permissionless*) com IOTA o privat (*permissioned*) només accessibles amb permisos com Swirlds Hasgraph.

Com exemple d'una implementació de DAG tenim NXT, Tangle de IOTA i Byteball. El rendiment és superior a Blockchain on NXT teòricament pot realitzar 367,200 transaccions per dia equivalent a 4,25 tps [12], FANTOM amb 300,000 tps teòriques i Tangle de IOTA he mesurat amb l'eina Tanglemonitor [13] en el moment de la creació d'aquest document, per tindre 1.4 tps i temps de confirmació de 3 minuts.

Existeixen *Smart Contracts* en algunes implementacions com Byteball o FANTOM per automatitzar transaccions sempre que es compleixen les condicions del *Smart Contract*. Degut a les característiques de DAG fa que els principals casos d'ús estiguin relacionats amb el processament de dades, planificació o cerca de millor ruta.

Com succeïx a Blockchain amb l'atac 51%, en sistemes DAG on només és necessari que el 33% de transaccions siguin fraudulentament. Quan la utilització del sistema, volum de transaccions, el cost és reduït, per tant a mesura que s'incrementa l'ús del sistema els atacs es fan més costosos i menys atractius per l'atacant, per exemple un atac seria rendible si el cost de realitzar les transaccions és superior al benefici obtingut. Per tal d'evitar-ho és necessari una figura centralitzada que previngui activitats malicioses, la qual es podria retirar en cas que el sistema aconseguís un gran volum de transaccions. Aquesta mesura a mida que augmenti l'ús del sistema serà més costosa de recursos i menys atractiu per l'atacant.

Una implementació de DAG que requereix mencionar és Hashgraph degut a que utilitza el protocol *gossip to gossip* però no és *open source*.

L'estructura està organitzada en esdeveniments generats per membres del sistema, aquests esdeveniments contenen els hash de dos esdeveniments anteriors (un del mateix membre i un d'un altre membre), informació de les noves transaccions que el membre genera en al moment i la signatura digital del membre.

Per arribar a consens, Hashgraph utilitza el protocol *Gossip* i altres tècniques com *Virtual Voting* i *Gossip about Gossip* que permet informar dels esdeveniments a tots els membres del sistema. El seu mètode de consens es basa en un procés de votació, mentre que el mètode de ponderació/selecció, per validar les transaccions, s'aconsegueix amb el mateix mecanisme de consens *Gossip about Gossip* i no en proves de validació dels membres com succeïx a Blockchain. Al utilitzar tolerància a falles binzantines, les transaccions es validen quan més de dos terços dels nodes son conscients d'aquesta transacció.

Com exemple de sistemes, tenim les privades Swirlds o NOIA amb un rendiment limitat per l'ample de banda [14] i la pública Hedera Hashgraph, amb un rendiment de fins a 500,000 tps [15].

2.1.1.2.4.3 **Holochain**

L'estructura del sistema Holochain [16] està formada per nodes que interaccionen entre ells i aquests nodes o usuaris tenen tres components: una aplicació, dades d'usuari (Source Chain) i una còpia de l'origen d'altres cadenes amb DHT (*Distributed Hash Table*). Les dades d'un node o usuari s'emmagatzemen en el ordinador del propi usuari [17].

Cada transacció es signada criptografiant i verificada per tots els usuaris involucrats en la transacció, considerant que cadascú dels usuaris d'una transacció valida la seva i la comparteix amb la resta del sistema, vol dir que les totes les dades del sistema han sigut validades per algú altre . Si un node propaga o valida una transacció amb dades errònies, el node es bloqueja i s'envia un missatge als altres nodes i així es té una llista d'actors fraudulents.

El seu tipus d'accés a la xarxa pot ser públic, privat o híbrid. Encara que es comparteixen les dades i permet crear sistemes públics, privats o híbrids que siguin interoperables entre ells. Per aconseguir privacitat es poden utilitzar entrades privades que no siguin compartides sota un control d'accés. També es configurable tindre permisos de lectura o escriptura. Per tant, Holochain permet crear aplicacions amb una governança i privacitat configurable per adequar-la als requeriments de l'aplicació. Un altre característica és que al centrar les dades en els usuaris, de forma nativa suporta la regulacions europea GDPR que en cas que les dades siguin de caràcter sensible serà un requisit [18].

El rendiment mesurat en tps creix linealment amb el número de nodes, degut a que la validació de les transaccions és independent entre elles i per tant a més nodes generant transaccions, més capacitat disposa el sistema per validar.

Està preparat per realitzar transaccions *online* i *offline*, i que els usuaris segueixin treballant encara que estiguin fora de la xarxa. Això permetre un millor aprofitament de recursos, per exemple en entorns IOT on vulguem optimitzar la comunicació dels dispositius. Holochain és una solució *open source* i encara està en una fase de desenvolupament inicial.

2.1.2 **Emmagatzematge distribuït**

En el cas d'aplicacions distribuïdes que utilitzin plataformes basades en DLT, les dades s'ubiquen en un registre de dades compartit i es sincronitzen a través de la plataforma segons la solució implementada.

Existeixen diferents plataformes o protocols descentralitzats on desplegar aplicacions: Sistemes basat en Blockchain, com Ethereum, Hyperledger o EOS que les dades es repliquen als nodes.

- Sistemes basats en DAG, com el protocols IOTA on el registre de dades és l'arbre sencer i o la plataforma bytball que pot tenir tot l'arbre o una part que contengui les dades suficients per funcionar.
- Holochain on les dades les tenen els usuaris de forma aleatoria altres nodes.

El cost d'utilització de la plataforma varia segons la implementació que facin del sistema DLT, i depèn de si els recursos del sistema són de propietat o no, com exemples tenim que el model a Ethereum hi han comisions per transacció, a EOS els recursos es lloguen i a Hyperledger s'han

d'adquirir. Ethereum utilitza el concepte de 'gas cost' per quantificar el cost dels recursos utilitzats per validar una transacció, mentre que a EOS es compren i venen recursos (RAM, CPU i ample de banda), tant si s'utilitzen com sinó. També existeix Hyperledger on els recursos de la plataforma s'han d'aprovisionar, desplegar i mantenir.

Per tant, en projectes on és requereix d'emmagatzemar grans quantitats d'informació, imatges o vídeos, per tal de reduir el número de transaccions i l'espai del registre de dades del sistema DLT, cal anar a solucions on al registre DLT es dessin la referència a la dada, el seu hash i altres dades com signatures, claus públiques, els permisos, mentres que la dada estarà en un sistema d'emmagatzematge distribuït. Per tant ens trobem les següents possibilitats:

- Serveis on es pot llogar espai com Amazon S3, MaidSafe, Storj.io, SIA o Filecoin els quals permeten encriptació de dades.
- Protocol IPFS, encara que de forma nativa no permet encriptar les dades, es pot utilitzar encriptació asimètrica [19]. L'usuari que vol compartir una dada ha d'encriptar-la amb una clau pública del usuari destinatari perquè aquest pugui descriptar amb la seva clau privada. Encara que un usuari accedeixi al fitxer no podrà llegir-ho sense la clau privada.

2.1.3 Compartició de dades selectiva i anònima

Les organitzacions, els dispositius electrònic i les persones al interactuar amb màquines, utilitzen dades que poden ser personals, bancaries, mèdiques i també generen continguts com comentaris en xarxes socials o d'operacions en diversos àmbits. Totes aquestes dades és consideren una representació digital de qui ets i que fas, també anomenada Identitat Digital.

En aquestes comunicacions, s'envien dades que poden contenir informació que els usuaris no vulguin divulgar, per mantenir la seva privacitat o només divulgar de forma anònima.

- La privacitat dels usuaris és la habilitat de quedar-se per un mateix, les seves dades i activitats que realitza, sense que la resta tingui coneixement.
- L'anonimat consisteix en revelar dades d'un usuari o les seves activitats sense que un tercer pugui associar-les a l'usuari al qual pertanyen.

Tant la privacitat dels usuaris com l'anonimat de les dades estan regulats a nivell Europeu per la *General Data Protection Regulation* (GDPR). En cas que no s'afegeixin dades personals al registre DLT, aquest sistema podrà complir la GDPR.

Per tant, per tal de potencia la privacitat caldrà minimitzar les dades aportades en les comunicacions i perquè les dades siguin anònimes caldrà definir quin tipus de dades no es poden compartir perquè no es puguin correlar amb els usuaris.

2.1.3.1 Compartició de dades selectiva

Tant en solucions de registres de dades amb accés públic com privat, és revelen dades que afecten a la privacitat. Les empreses per oferir els seus serveis, emmagatzemen dades amb accés privat que poden accedir ells o tercers en cas que cedeixin les dades. Aquestes dades en el cas de Google poden ser [20] nom complet, nom d'usuari, paraula de pas, gènere i data

de naixement, número de telèfon, número de targeta de crèdit, domicili d'entrega, històric d'ubicació GPS, històric de paraules cercades, entre d'altres dades, per tant en cas d'un accés no autoritzat per part d'un atacant es veurà compromès la privacitat de l'usuari i conseqüentment també el seu anonimats. En sistemes que emmagatzemin dades com número de seguretat social, DNI o historial mèdic on surten les malalties d'un pacient, l'impacte d'un accés no autoritzat és encara més gran. L'encriptació és utilitzada per tal de securitzar aquestes dades i així millorar la privacitat en cas d'un accés no autoritzat, però és en els intercanvis de dades on és necessari conèixer les dades per tal de validar la informació, com per exemple quan s'accedeix al correu electrònic amb l'usuari i paraula de pas.

Si s'emmagatzemen les dades en un registre compartit, en cas que sigui públic serà accessible per tothom, per tant quan existeixen dades sensibles es tendeix a solucions d'accés privat on només accediran els actors que tinguin permisos.

Amb l'objectiu de potenciar la privacitat, la solució pot utilitzar dels següents components o mecanismes: Registre de dades d'accés privat amb permisos d'accés.

- Registre de dades d'accés públic per a dades no sensibles.
- Registre de dades ubicat en l'usuari i còpies de seguretat encryptades, que podrien estar en un sistema d'emmagatzematge distribuït o privat, fora del registre públic.
- Mecanismes criptogràfics, com *zero-knowledge*, per verificar informació sense haver de donar dades.

Zero-knowledge, *ZKP*, permet verificar una informació sense haver de donar la informació, com per exemple que ets major d'edat sense compartir la resta de dades del teu DNI, si tens més de 100€ a la compte corrent per fer un pagament sense indicar tot l'històric de transaccions o si tens permès l'accés a un servei sense revelar la teva paraula de pas. Hyperledger Indy utilitza *ZKP* [21].

De les diferents solucions existents, he identificat que les següents permeten controlar l'accés a dades i per tant la possibilitat de complir la GDPR:

1. Hyperledger:

Disposa dels següents projectes que poden encaixar amb la GDPR:

- Fabric, permet crear sistemes privats i permisionats. Utilitza col·leccions de dades privades, limitant l'accés de les dades a determinats actors o limitant la vida de la dada, mantenint els *hashes* del block però la dada serà inaccessible.
- Indy, permet crear sistemes públics i permisionats on gestionar identitats digitals.

2. Hedera hashgraph

Permet modificar l'estructura dels blocs per emmagatzemar fitxers que es cercaran pel seu *hash*. Existeixen identificadors de fitxers *File ID* de forma que es poden crear noves versions que s'associarien a aquest identificador. Permet esborrar informació per usuaris que tinguin permisos, per exemple una llicència entre un usuari i una administració en cas que caduqui podrien deshabilitar la llicència esborrant el *hash*.

3. Holochain

Permet crear aplicacions amb privacitat configurable. Existeix un control d'accés configurable per tindre permisos de lectura o escriptura i també existeixen entrades privades [18] .

Hedera hasgraph i Ehetreum utilitzen registre de dades públics, Holochain és híbrid, una part pública i un altre privada, i només Hyperledger Fabric és privada. Des del punt de vista de privadesa, els sistemes amb registre de dades privats són més robusts que els públics.

Respecte a l'estat de maduresa de les tres solucions i el seu suport, en el moment de la creació del document a github Holochain indica que no està preparat per producció, no es garanteix que el codi sigui segur i no hi ha casos d'èxit, Hedera hasgraph té 33 dApps funcionant i Hyperledger té el suport de 'The Linux Foundation' qui hostatja més de 100 projectes opensource de més de 100 companyies amb desenes de milers de desenvolupadors, amb un cost estimat de 16 bilions de dollars. Al gener de 2018 es va realitzar un cas d'estudi per a 'Change Healthcare' [22], on utilitzant Fabric van processar fins a 50 milions de transaccions diàries i 550 transaccions per segon.

Com a conclusions, Hyperledger Fabric és opensource i independent de fabricants per tant adoptable per qualsevol actor que s'afegeixi, amb un gran suport de desenvolupadors, amb capacitats de privadesa i possibilitats de complir la GDPR.

2.1.3.2 Compartició de dades anònimes

Quines dades compartirem?

Les dades són anònimes fins que algú les relaciona amb una identitat. Per tant la nostre identitat digital no ha de contenir dades personals. Però amb això és suficient? Això depèn de les dades que es comparteixin, degut a que existeixen tècniques que permeten relacionar aquestes dades anònimes amb dades públiques per identificar usuaris. Per exemple, un usuari ha compartit de forma anònima la seva ubicació GPS i de forma pública amb el seu nom complet ha compartit fotografies o *Tweets* que incloguin la ubicació GPS, d'aquesta forma es podria arribar a relacionar ubicacions i fotos per identificar una persona. Si l'atacant disposa d'informació privada d'alguna empresa com Facebook o Google podria relacionar aquest nom amb ubicacions GPS per conèixer si ha anat a una església o mesquita, a un local que és *Gay Friendly*, a una clínica de reproducció assistida o a la seu d'un partit polític o relacionar amb l'històric de cerques per si estàs interessat quins son els tractament d'una malaltia determinada.

Tota aquesta informació donada de forma anònima pot arribar crear un perfil d'una persona que atenta a la seva privacitat.

Com compartirem les dades?

L'encriptació de dades amb clau privada i protocols *zero-knowledge proof* (ZKP) per emmagatzematge permet compartir les dades de forma anònima. Els usuaris encriptaran les dades en els seus dispositius prèviament a enviar-les, no fer-ho en un servei cloud perquè s'enviarien sense encriptar i podrien ser interceptades.

També es pot explorar la utilització de permisos en blocs que les dades a compartir sense cap informació personal, per tal de limitar l'accés als actors amb qui vulguem compartir-les.

Tanmateix, es pot analitzar el limitar la vida de la dada, ja sigui amb smartcontracts o funcionalitats del sistema DLT com per exemple Hyperledger Fabric que disposa de idemix (Identity Mixer) amb ZKP [23], que ofereix als clients autenticació de les transaccions de forma anònima, per exemple que pertanyes a una organització o posesixes una sèrie d'atributs. Si es limita l'accés de les dades durant la nostre estància al hospital, durant una entrevista de treball o quan treballem en una empresa, aconseguirem que estigui disponible el temps que desitgem.

2.1.4 Dispositius wearables

En l'actualitat existeixen gran varietat de dispositius *wearables*, els quals permeten mesurar freqüència cardíaca (BPM), concentració d'oxigen en sang (SpO2), pressió arterial (BP), glucosa en sang (FBS) i fer electrocardiogrames (ECG). Alguns d'aquest dispositius tenen aplicacions per sistema operatiu Android o IOS per *smartphones*, que es connecten per Bluetooth per tal de visualitzar les dades.

En un futur s'espera que apareguin dispositius *wearables* que amb el contacte de la pell mesurin el nivell de glucosa o altres indicadors per detectar problemes de salut.

Per tal de recollir dades d'aquests dispositius ens trobem que no existeix una API o estàndard per obtenir les dades, però les connexions BLE (*Bluetooth Low Energy*) també anomenades *Bluetooth Smart*, estan basades en especificacions GATT (*Generic Attributes*) [24] que defineixen la comunicació i enviament de dades i que els atributs existents es poden consultar [aquí](#). D'aquesta forma serà possible identificar les dades dels dispositius. Hi han projectes que utilitzen BLE per extreure dades com Gadgetbridge [25]. Per tant, ens centrarem en dispositius amb connectivitat Bluetooth Low Energy.

La cerca s'ha fet a Google i a Amazon per veure la disponibilitat i cost dels dispositius a Espanya. Les característiques s'han obtingut de les webs de cada fabricant. A continuació es mostra una primera Taula 4: Dispositius compatibles amb Bluetooth Low Energy (Smart). amb els dispositius compatibles amb BLE i una segona Taula 5: Dispositius no identificats com compatibles amb Bluetooth Low Energy (Smart). amb altres que no està comprovat que ho siguin o són interessants perquè mesuren glucosa en sang o la pressió arterial amb més fiabilitat que els sensors òptics.

Model	Connectivitat	BPM	ECG	SpO2	BP	FBS	Preu [€]	Observacions a Amazon
Xiaomi Miband 2	BT 4.2 BLE	✓	✗	✗	✗	✗	19,19	Opinions bones.
Xiaomi Miband 3	BT 4.2 BLE	✓	✗	✗	✗	✗	24,30	Opinions bones.
Amazfit Bip	BT 4.0 BLE	✓	✗	✗	✗	✗	69,94	Opinions bones.
TagoBee TB09	BT 4.2 BLE	✓	✗	✓	✓	✗	23,90	Opinions diferents.
Huawei Honor Band 5	BT 4.2 BLE	✓	✗	✗	✗	✗	29,90	Opinions bones.
Garmin Vivosmart 4	Bluetooth Smart	✓	✗	✓	✗	✗	125,91	Opinions bones.

Apple Watch Series 5	Bluetooth 5.0	✓	✓	✗	✗	✗	449,90	Opinions bones.
Withings Move ECG	BLE	✓	✓	✗	✗	✗	129,95	Opinions bones.
Omron HeartGuide	BLE	✓	✗	✗	✓	✗	499	Només disponible als Estats Units. Preu en dollars.

Taula 4: Dispositius compatibles amb Bluetooth Low Energy (Smart).

Model	Connectivitat	BPM	ECG	SpO2	BP	FBS	Preu [€]	Observacions a Amazon
Abbott FreeStyleLibre	NFC	✗	✗	✗	✗	✓	59,90	Serveis per a 14 dies.
Apple One Drop Chrome Blood Glucose Monitoring Kit	Bluetooth 5.0	✗	✗	✗	✗	✓	69,95	Només disponible als Estats Units. Preu en dollars.
Withings BPM Connect	Bluetooth	✗	✗	✗	✓	✗	96,45	L'única opinió és bona.
Withings BPM Core	Bluetooth	✗	✓	✗	✓	✗	249,95	L'única opinió és bona.
Zeblaze Neo	Bluetooth 4.0	✓	✗	✗	✓	✗	41,79	Sense opinions.
Zeblaze Hybride 2	Bluetooth 4.0	✓	✗	✗	✓	✗	41,56	Sense opinions.

Taula 5: Dispositius no identificats com compatibles amb Bluetooth Low Energy (Smart).

Existeixen dispositius com el Withings BPM [26], Zeblaze Neo o Zeblaze Hybride 2 que mesuren pressió arterial o el Zeblaze VIBE 3 ECG que realitza electrocardiogrames [27], però no indiquen compatibilitat amb BLE. Els nous models de Xiaomi necessiten registrar-se a la aplicació 'Mi Fit' i privilegis de *root* al smartphone, per tal d'obtenir la clau de vincle necessària per connectar-se al dispositiu, segons Gadgetbridge [25].

La fiabilitat de dades dels dispositius analitzats és limitada perquè majoritàriament utilitzen sensors òptics, menys precisos que els mètodes tradicionals. El ritme cardíac es calcula a partir de llum vermella i infraroja, la pressió arterial es mesura en el canell que és menys fiables que els que ho fan al braç com el Withings BPM. Els electrocardiogrames utilitzen 1 derivació enlloc de les 12 del mètode tradicional, per tant només podrien arribar a detectar una arrítmia del cor, però no altres malalties.

Per concloure, hi ha una mancança de dispositius compatibles amb estàndards per obtenir dades, com el BLE, per tant es fa difícil trobar-hi dispositius que aglutini diverses mesures alhora, com la pressió arterial, la concentració d'oxigen en sang i electrocardiograma que ens aportarien informació per estimar la salut d'una forma més fiable. Tampoc he trobat dispositius per mesurar la glucosa en sang compatibles amb BLT, i els existents com el FreeStyleLibre [28] amb NFC només duren 14 dies, o el One Drop d'Apple [29] amb Bluetooth 5.0 que no es comercialitza a Europa. En aquest treball les polseres quantificadores més adients, són compatibles amb BLE, com les Xiaomi miband 2 ó 3 [30].

2.2 Disseny de la solució

Tenint present els objectius d'aquest treball i el coneixement adquirit definirem els requeriments de la solució, els seus criteris d'acceptació, així com també el disseny de la solució.

2.2.1 Requeriments de la solució

La solució ha de complir els següent requisits:

- L'aplicació ha d'esser descentralitzada i utilitzar Ethereum o Hyperledger.
- Recollir dades des d'un dispositiu *IoT* físic o simulat.
- Emmagatzemar una gran quantitat de dades privades de forma segura i privada.
- Visualitzar les dades recollides pel dispositiu *IoT* físic o simulat.
- Les dades recollides s'han de poder compartir amb tercers de forma anònima.
- El tractament de les dades ha de complir la GDPR.

2.2.2 Criteris d'acceptació de la solució

Per tal d'acceptar la solució ha de complir els següents criteris:

- L'aplicació sigui desplegable en un entorn de proves o una plataforma descentralitzada.
- Connectar amb un dispositiu *IoT* físic o simulat.
- Recollir un tipus de dades d'un dispositiu *IoT* físic o simulat.
- Desar dades en un sistema d'emmagatzematge distribuït.
- Llegir dades d'un sistema d'emmagatzematge distribuït.
- Compartir dades privades amb un altre usuari.
- Un usuari comparteixi dades privades amb un altre de forma anònima.
- Complir amb el tractament de dades *General Data Protection Regulation* (GDPR).

2.2.3 Selecció de tecnologies i maquinari

A partir d'analitzar els requeriments i amb l'estudi de les diferents tecnologies proposaré un disseny de la solució a alt nivell que compleixi els requeriments.

- L'aplicació ha d'esser descentralitzada i utilitzar Ethereum o Hyperledger: Qualsevol de les dues tecnologies permet desplegar aplicacions sobre xarxes Blockchains descentralitzades. Per tant, s'analitzarà la resta de requeriments per escollir la més apropiada al nostre cas.
- Recollir dades des d'un dispositiu wearable físic o simulat: Els dispositius wearables no utilitzen una mateixa normativa o API estàndard, el navegador web Chrome i alguns dispositius són compatibles amb Bluetooth Low Energy (BLE), i basades en especificacions GATT (Generic Attributes) que defineixen la comunicació i enviament de dades i, per tant, ens facilitarà el desenvolupament. Per fer-ho, necessitarem:
 - Un dispositiu wearable compatible amb BLE que reculli dades de salut dels usuaris, com el Xiaomi MiBand2 que és el més econòmic.
 - Una programari com a interfície d'usuari (UI) per registrar els dispositius i associar-los als usuaris. Existeix Angular, React i Vue que permeten crear aplicacions webs, de les quals es selecciona React perquè té una corba d'aprenentatge inferior a

- Angular i similar a Vue, té el suport de Facebook i està molt demandada a l'entorn laboral.
- Un programari que atengui peticions de la UI i dades dels dispositius a través d'una API. Amb l'objectiu de desplegar-ho ràpidament, tenim frameworks com Laravel per PHP, Django o Flask per Python, Spring per Java o Sinatra per Ruby, es selecciona Express per a Node.js que permet crear ràpidament aplicacions, en aquest cas APIs.
 - Un dispositiu amb navegador web Chrome, connectivitat BLE i compatible amb les especificacions GATT, com un Android, Chrome OS, Mac o Windows 10. Utilitzaré un dispositiu amb Windows 10, connector Bluetooth BLE i navegador web Chrome.
 - Un programari que envii i llegeixi dades a la Blockchain, en el cas de Hyperledger Fabric disposa de kits per desenvolupar aplicacions i SmartContracts, tant per a Node.js com Java. Per desenvolupar l'aplicació s'utilitzarà el Hyperledger Fabric SDK for Node.js i per agilitzar el desenvolupament de SmartContracts utilitzarem el framework Convector Smart Contract que treballa amb Node.js. S'utilitzarà Hurley que facilita les proves desplegament Hyperledger Fabric.
- Emmagatzemar una gran quantitat de dades privades de forma segura i privada: Degut a la gran quantitat de dades que generaran els dispositius wearables, Ethereum no podria assumir el volum de tps ni el cost associat, per tant s'utilitzarà Hyperledger el qual té més rendiment de tps, els recursos són de propietat i permeten dades privades. Degut al volum de dades i la necessitat de la seva privadesa serà necessari el següent:
 - Un programari/API que envii dades a un emmagatzematge extern, que en el nostre cas primer s'utilitzarà el registre de dades privades de Hyperledger Fabric per adquirir coneixement del funcionament i si hi ha temps s'implementaria en un emmagatzematge extern. Per la primera part s'utilitzarà el Hyperledger Fabric SDK for Node.js i per la segona Express per a Node.js que permet crear ràpidament APIs.
 - Visualitzar les dades recollides pel dispositiu wearable físic o simulat: La interfície d'usuari ha de permetre fer peticions de lectura de les dades dels dispositius IOT i visualitzar-les. Per tant necessitem:
 - Una UI que demani dades a l'aplicació/API i les visualitzi. El Framework React disposa de llibreries que permeten visualitzar dades com Victory.
 - Un programari que llegeixi les dades emmagatzemades i les retorni a la UI per la seva visualització. Per llegir les dades s'utilitzarà el Hyperledger Fabric SDK for Node.js en cas que estigui a Hyperledger Fabric o una aplicació si està en un emmagatzematge extern, per la segona part de retornar les dades a la UI s'utilitzarà React.
 - Les dades recollides s'han de poder compartir amb tercers de forma anònima: La interfície d'usuari ha de permetre informar a la plataforma Blockchain que comparteixi un determinat grup de dades. S'utilitzarà Hyperledger Fabric SDK for Node.js per implementar Identity Mixer.
 - El tractament de les dades ha de complir la GDPR: Dels projectes Hyperledger tenim Fabric i Indy. El primer és una blockchain privada amb comunicacions privades on existeix un cas d'ús, mentre que Indy és pública i està dissenyada per la gestió d'identitat i la privacitat. S'utilitzarà Hyperledger Fabric perquè permet limitar l'accés de les dades a determinats actors o limitar la vida de la dada, mantenint els hashes del block però la dada serà inaccessible.

Adicionalment es necessitarà un equip amb sistema operatiu Windows o Linux on desplegar l'entorn de desenvolupament i la xarxa Hyperledger Fabric en local. A banda del programari indicat anteriorment, com a gestor de versions s'utilitza Git, i com editor de codi es selecciona Visual Studio Code, compatible amb els frameworks seleccionats i on podré testear les aplicacions i els SmartContracts.

A continuació es resumeix el llistat de tecnologies de programari i maquinari escollit per a la solució:

- Programari:
 - Hyperledger Fabric com a plataforma de l'aplicació descentralitzada.
 - React per la interfície d'usuari (UI) que rebrà peticions dels usuaris i dades de la Blockchain. La llibreria Victory de React per visualitzar dades a la UI.
 - Hyperledger Fabric SDK for Node.js per enviar i llegir dades de la Blockchain, i compartir dades de forma anònima.
 - Convector Smart Contract per crear SmartContracts/chaincodes i Hurley per fer les proves sobre Hyperledger Fabric.
 - Express i Node.js per desenvolupar l'aplicació que rebrà peticions de la UI, rebrà dades dels dispositius IoTs, les enviarà a la Blockchain/emmagatzematge extern i llegirà dades de la Blockchain.
 - Visual Studio Code com editor de codi.
- Maquinari:
 - Equip amb sistema operatiu distribució de Linux per desplegar entorn de desenvolupament.
 - Dispositiu amb Windows 10, connector Bluetooth BLE i navegador web Chrome per recopilar dades del dispositiu IoT.
 - Dispositiu wearable, Xiaomi Miband3.

Identifiquem els actors i els seus rols:

Els actors poden ser qualsevol usuari o sistema extern que interactui amb el sistema, però no són part del sistema.

- Usuaris que interaccionen amb el sistema Fabric. Existeixen tres tipus d'usuari, els dos primers a través d'una aplicació web i el tercer a través :
 - Usuaris amb wearables que volen demanar al sistema donar d'alta dispositius IoTs, que enregistri les dades enviades pels dispositius IoTs i autoritzar que comparteixi determinades dades seves amb altres usuaris.
 - Usuaris consumidors de dades, que volen llegir les dades d'un o més pacients, com professionals de la salut per tal d'avaluar l'estat de salut o investigadors per tal de fer anàlisi de dades.
 - Usuaris que interaccionen amb el sistema Fabric directament per administrar el sistema.
- Interfície d'usuari que permeti als usuaris realitzar peticions i que s'enviïn a l'aplicació API.
- Dispositiu IoT que enviaran les dades de la freqüència cardíaca al sistema Fabric.
- Aplicació API que rebrà peticions d'usuaris a través de l'aplicació web i les enviarà al sistema Fabric.
- Emmagatzematge distribuït extern que rebrà peticions del sistema Fabric.
- Un administrador del sistema que registri l'alta de nous usuaris.

Un cop tenim les tecnologies que utilitzarem, els actors i rols que representen, definim a alt nivell l'esquema de la solució, veure Figura 4: Disseny d'alt nivell de la solució - Font: Elaboració pròpia..

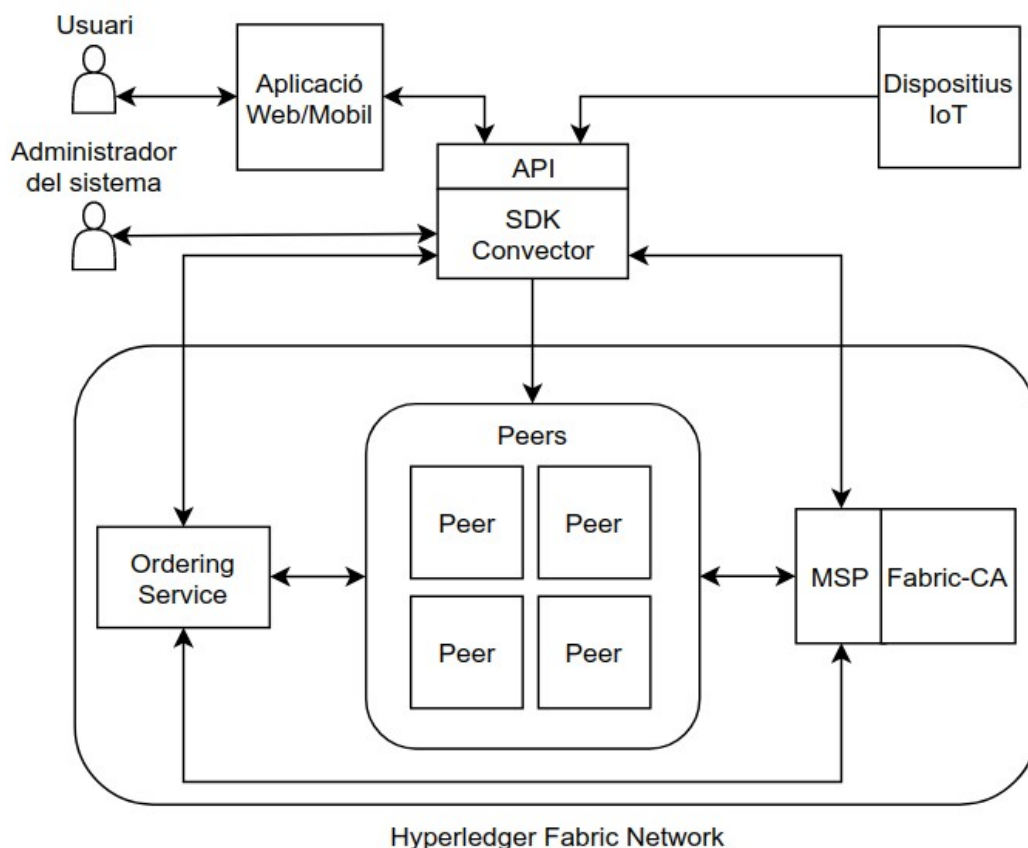


Figura 4: Disseny d'alt nivell de la solució - Font: Elaboració pròpia.

Per escriure el chaincode necessitarem els models amb la seva estructura de dades. S'ha analitzat pels diferents actors:

- Usuari: Els atributs són l'identificador de l'usuari, l'organització i array d'identitats.
- Dispositiu IoT: Els atributs són la data de creació, identificador de l'usuari associat.
- Dada IoT: Identificador del dispositiu IoT que ha generat la dada i la data de creació.

Estratègia de configuració del sistema Hyperledger Fabric serà la següent:

- Creació de dues organitzacions, una pels usuaris amb wearables i els dispositius IoTs (OrgIoT) i una segona pels usuaris consumidors de dades (OrgConsD).
- Creació d'un canal entre les dues organitzacions.
- Cada organització disposarà de dos peers, per disposar de redundància.
- L'administrador del sistema instal·larà el chaincode en els peers que inclourà la política d'aprovació.
- Els administradors de cada organització instanciaran el chaincode als peers.

3 Execució

A la fase d'Execució, primer muntarem l'entorn de treball per tal de desenvolupar la PoC i també detallarem la solució a baix nivell.

Posteriorment definirem el pla de proves i els resultats esperats, que executarem sobre la PoC desenvolupada, per tal d'analitzar el seu resultat i l'assoliment dels criteris d'acceptació del pla de proves.

Finalment s'explicaran les conclusions de la PoC i futures línies de treball que serien interessants seguir.

3.1 Desenvolupament de la POC

Primer instal·larem les eines per disposar de l'entorn de treball per desenvolupar la PoC i desplegar-la.

3.1.1 Provisió d'entorn de desenvolupament

L'entorn de desenvolupament consta de maquinari, descrit a la Taula 6: Maquinari de la solució, sobre el que instal·larem el programari, descrit a la Taula 7: Programari de la solució, necessari per desenvolupar la PoC i executar el pla de proves. A les taules següents, per a cada element de maquinari i programa s'especifica si és requisit per a la PoC desenvolupada.

De maquinari disposem del següent:

Maquinari	Detall	Requerit a la PoC
Equip amb sistema operatiu Linux	Ordinador de sobretaula amb processador Intel i7-4770, 16 GB de RAM, 14 GB d'espai lliure d'emmagatzematge i sistema operatiu Ubuntu 18.04.3 LTS amb kernel 5.0.0-36-generic.	Sí, s'utilitzarà per executar les eines de desenvolupament.
Dispositiu amb Windows 10, connector Bluetooth BLE i navegador web Chrome	Portàtil marca Lenovo, model 31-80 amb Windows 10 Pro v1903 SO v18362.476, drivers Bluetooth 'Qualcomm Atheros QCA9377 Bluetooth 4.1' v10.0.0.278, 'Enumarador de Bluetooth LE de Microsoft' v10.0.18362.1 i navegador web Chrome v78.0.390-108 (64 bits).	No, necessari per registrar dispositiu wearable i recollir dades.
Dispositiu wearable, Xiaomi Miband2.	Disposarem tant del Xiaomi Miband2 com del Xiaomi Amazfit Bip.	No, necessari per registrar dispositiu wearable i recollir dades.

Taula 6: Maquinari de la solució

De programari tenim el següent:

Programari	Requerit	Instal·lat	Requerit
cURL	<ul style="list-style-type: none"> • Sense especificar 	<ul style="list-style-type: none"> • curl v7.58.0 	Sí, per la instal·lació de nodejs, docker i Hyperledger Fabric.
Docker ce i Docker compose	<ul style="list-style-type: none"> • Docker version 17.06.2-ce or greater • Docker Compose version 1.14.0 or greater 	<ul style="list-style-type: none"> • Docker v18.09.7, build 2d0083d • Docker-compose v1.17.1, build unknown 	Sí, per executar Hurley i Hyperledger Fabric.
Nodejs , NPM i Express	<ul style="list-style-type: none"> • Nodejs Runtime 8.9.4 and higher or 10.15.3 and higher • NPM • Express 	<ul style="list-style-type: none"> • Nodejs v10.18.0 • npm v6.13.4 • Express v4.17.1 	Sí, per desenvolupar el servidor API que rebrà petició del client front-end.
React	<ul style="list-style-type: none"> • Sense especificar 	<ul style="list-style-type: none"> • React v16.12.0 	Sí, per desenvolupar el client front-end
Llibreria victory per a React	<ul style="list-style-type: none"> • Sense especificar 	<ul style="list-style-type: none"> • Victory v33.1.7 	No, necessari per visualitzar les dades.
Convector Smart Contract	<ul style="list-style-type: none"> • Sense especificar 	<ul style="list-style-type: none"> • Convector v1.1.7 • Convector-cli v1.1.7 	Sí, per codificar el chaincode.
Hurley	<ul style="list-style-type: none"> • Sense especificar 	<ul style="list-style-type: none"> • Hurley v1.1.2 • Fabric 1.4.0 	Sí, per executar el pla de proves sobre Hyperleger Fabric.
Git	<ul style="list-style-type: none"> • Sense especificar 	<ul style="list-style-type: none"> • Git v2.17.1 	Sí, per controlar les versions del codi.
Visual Studio Code	<ul style="list-style-type: none"> • Sense especificar 	<ul style="list-style-type: none"> • Code v1.40.2 	Sí, com editor de codi.
Hyperledger Fabric	<ul style="list-style-type: none"> • Hyperledger Fabric 1.4.4 • Go version 1.12.x • Phython 2.7 	<ul style="list-style-type: none"> • Hurley v1.1.2 • Go v1.13.5 linux/amd64 • Python 2.7.15+ 	No, s'utilitza Hurley per realitzar el pla de proves.
Hyperledger Fabric application SDK for Node.js i Gulp	<ul style="list-style-type: none"> • Sense especificar 	<ul style="list-style-type: none"> • Fabric-sdk-node v1.4.4 • Gulp v4.0.2 	No, s'utilitza Hurley per realitzar el pla de proves.

Taula 7: Programari de la solució

A continuació, s'expliquen breument les eines i la seva instal·lació. Als programari on no s'especifica la versió requerida, s'ha instal·lat la última versió LTS o estable disponible.

CURL:

Descripció: Permet descarregar arxius de servidors remots.

Requisit: És necessari per a la instal·lació de nodejs, docker i Hyperledger.

Instal·lació:

```
manelmdiaz@desktop-i7-linux:~$ sudo apt install curl
manelmdiaz@desktop-i7-linux:~$ curl -V
```

Docker ce i Docker compose

Descripció: Plataforma per crear, compartir i executar aplicacions amb contenidors.

Requisit: És prerequisit per a l'execució de Hurley i Hyperledger Fabric.

Instal·lació:

```
manelmdiaz@desktop-i7-linux:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
manelmdiaz@desktop-i7-linux:~$ sudo apt-get install docker-ce
manelmdiaz@desktop-i7-linux:~$ docker -v
```

Afegim el nostre usuari al grup d'usuaris docker

```
manelmdiaz@desktop-i7-linux:~$ sudo usermod -a -G docker $USER
```

Inicialitzem docker

```
manelmdiaz@desktop-i7-linux:~$ sudo systemctl start docker
manelmdiaz@desktop-i7-linux:~$ sudo systemctl enable docker
```

Nodejs, NPM i Express

Descripció: Nodejs és un entorn d'execució per a Javascript, npm és el sistema de gestió de paquets per a nodejs i Express és un mòdul que necessita de Nodejs 10.0 o superior que ens permet crear un servidor API ràpidament.

Requisit: És necessari per desenvolupar el servidor API que rebrà peticions del client front-end.

Instal·lació:

Afegim el repositori de la versió 10 de nodejs que és l'última versió LTS disponible, i per tant un suport actiu de 18 mesos, i inclou npm. Posteriorment instal·lem.

```
manelmdiaz@desktop-i7-linux:~$ curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash
-
manelmdiaz@desktop-i7-linux:~$ sudo apt-get install -y nodejs
manelmdiaz@desktop-i7-linux:~$ node -v
```

S'instal·la Express globalment i es crea el servidor API.

```
manelmdiaz@desktop-i7-linux:~$ sudo npm install express-generator -g
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ npx express-generator servidor
```

S'instal·len les dependències.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ cd api
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/servidor$ npm install
```

Per tal de configurar un proxy per tal de redirigir les peticions API al backend Express, es faran les dues tasques següents:

1. Crear el fitxer server.js amb el següent contingut:

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/client$ nano server.js
const express = require('express');
const app = express();
const port = process.env.PORT || 8000;

// console.log that your server is up and running
app.listen(port, () => console.log(`Listening on port ${port}`));

// create a GET route
app.get('/express_backend', (req, res) => {
  res.send({ express: 'YOUR EXPRESS BACKEND IS CONNECTED TO REACT' });
});
```

2. Al fitxer package.json afegim la línia, per configurar el proxy:

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/client$ nano package.json
"proxy": "http://localhost:8000"
```

React

Descripció: És una biblioteca Javascript per construir interfícies d'usuari.

Requisit: És necessari per desenvolupar la interfície d'usuari, client front-end, que l'usuari utilitzarà per interactuar amb els dispositius wearables i la infraestructura Hyperledger.

Instal·lació:

Instal·lem create-react-app globalment, que ens permet crear aplicacions React.

```
manelmdiaz@desktop-i7-linux:~/TFM$ sudo npm install -g create-react-app
```

Posteriorment creem una aplicació base.

```
manelmdiaz@desktop-i7-linux:~$ cd wearhealth-poc
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ mkdir client
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ create-react-app client
```

S'instal·len les dependències.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ cd client
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/client$ npm install
```

Instal·larem 'react-router-dom', per tal de crar rutes en aplicacions React.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/client$ npm install --save react-router-dom
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/client$ npm install react-responsive-modal --save
```

Llibreria victory

Descripció: És una llibreria de React que permet visualitzar dades en aplicacions React.

Requisit: És necessari per facilitar la visualització de dades en aplicacions React.

Instal·lació:

```
manelmdiaz@desktop-i7-linux:~$ sudo npm install -g victory --save
```

Convector Smart Contract i Hurley

Descripció:

- Convector Smart Contracts, és un framework Javascript per crear SmartContracts/chaincodes.
- Hurley, permet configurar fàcilment un entorn de desenvolupament Hyperledger on provar els chaincodes.

Requisit: És necessari per crear chaincodes i executar el pla de proves sobre Hyperledger Fabric.

Instal·lació:

S'instal·la Go i python, requisits per Hurley i Hyperledger Fabric. Posteriorment instal·lem globalment convector-cli i hurley.

```
manelmdiaz@desktop-i7-linux:~$ sudo apt-get install golang python
manelmdiaz@desktop-i7-linux:~$ sudo npm i -g @worldsibu/convector-cli
manelmdiaz@desktop-i7-linux:~$ npm i -g @worldsibu/hurley
```

Creem un projecte a la carpeta convector amb el chaincode participant.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ conv new convector -c participant
```

Instal·lem les dependències.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ cd convector
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ npm i -g
```

Git

Descripció: Eina de control de versions de codi.

Requisit: És necessari per controlar les versions al desenvolupar.

Instal·lació:

Prèviament al web github.com s'ha creat un repositori anomenat 'wearhealth-poc' amb el nom d'usuari ManelMartinezDiaz i email manelmdiaz@uoc.edu.

Instal·lem Git.

```
manelmdiaz@desktop-i7-linux:~$ sudo apt install git
manelmdiaz@desktop-i7-linux:~$ git --version
```

Creem un repositori local de la carpeta wearhealth-poc.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ echo "# wearhealth-poc" >> README.md
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ git init
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ git add README.md
```

Configurem les dades de la conta d'usuari de github creada prèviament.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ git config --global user.name
ManelMartinezDiaz
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ git config --global user.email
manelmdiaz@uoc.edu
```

Afegim els arxius al HEAD de la copia local fent un commit.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ git commit -m "first commit"
```

Ens connectem al repositori existent amb remote add origin.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ git remote add origin
git@github.com:ManelMartinezDiaz/wearhealth-poc.git
```

Enviam els fitxers al repositori remot de github amb push -u origin master.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc$ git push -u origin master
```

Visual Studio Code

Descripció: És un editor de codi que permet crear i depurar aplicacions.

Requisit: És necessari per desenvolupar la PoC.

Instal·lació:

Importem les claus de Microsoft GPG i afegim el repositori de Visual Studio code. Posteriorment actualitzem amb apt per llegir el nou repositori, instal·lem Visual Studio Code i comprovem la versió instal·lada.

```
manelmdiaz@desktop-i7-linux:~$ wget -q https://packages.microsoft.com/keys/microsoft.asc -O- |
sudo apt-key add -
manelmdiaz@desktop-i7-linux:~$ sudo add-apt-repository "deb [arch=amd64]
manelmdiaz@desktop-i7-linux:~$ sudo apt update
manelmdiaz@desktop-i7-linux:~$ sudo apt install code
manelmdiaz@desktop-i7-linux:~$ code -v
```

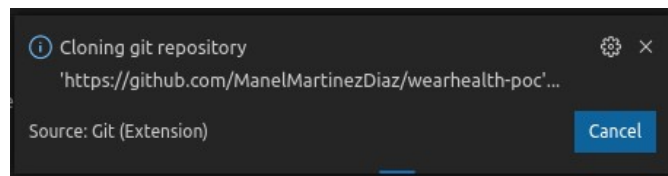
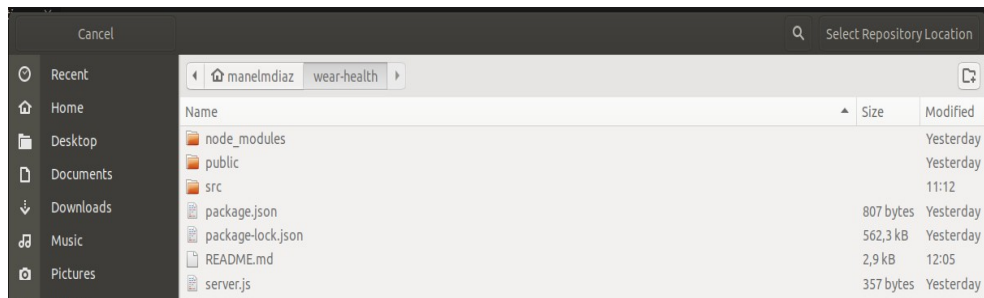
Iniciem Visual Studio Code i configurem Git al visual studio Code per gestionar les versions de codi, a mesura que desenvolupem.

```
manelmdiaz@desktop-i7-linux:~$ code
```

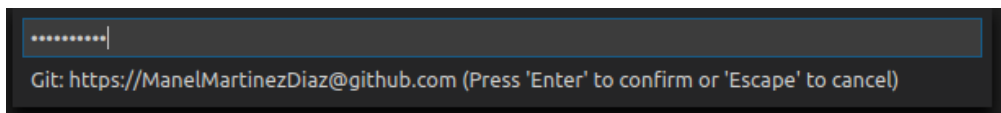
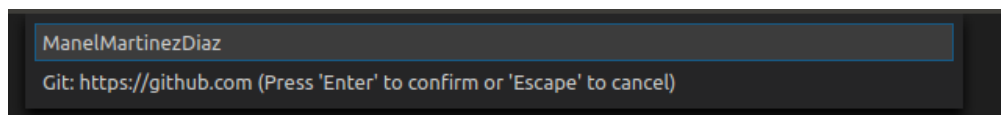
Al menú, anem a View/command palette i teclegem:

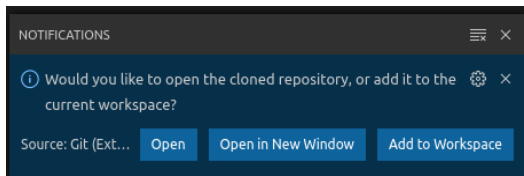
```
>git clone
>https://github.com/ManelMartinezDiaz/wearhealth-poc
```

Ens demana la carpeta local, seleccionem la carpeta de l'usuari i ens mostra que s'està clonant el repositori.



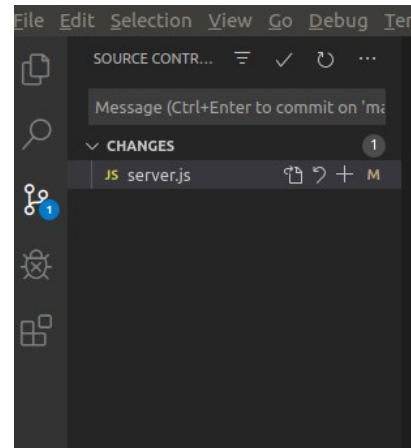
Ens demana l'usuari, després la paraula de pas i, a continuació, l'afegim al Workspace amb 'Add to Workspace', que prèviament s'ha creat al web github.com.





Per tal de disposar a Visual Code Studio les capetes del repositori, seleccionar 'Add to Workspace' quan ens ho preguntin per tal d'afegir-ho a l'espai de treball.

Per tal de desar al HEAD del repositori local, al menú esquerra, la imatge de Source Control i símbol +. Posteriorment incloure un missatge de commit i el símbol de verificació.



Per pujar-ho a github.com a la barra inferior clic al símbol d'actualitzar. Et demanarà l'usuari i la paraula de pas a la barra superior.



Per tal d'executar comandes git, com crear branques, fer merge a d'altres branques, anem a View/command palette i teclegem la comanda corresponent:

- Git: Create Branch From
- Git: Merge Branch
- Git: Commit All
- Git: Push

Hyperledger Fabric + Go + Python

Descripció: plataforma per desenvolupar solucions i aplicacions DLT permissionades.

Requisit: És necessari per executar aplicacions distribuïdes sobre Hyperledger Fabric.

Instal·lació:

Es descarrega i s'instal·la Hyperledger Fabric.

```
manelmdiaz@desktop-i7-linux:~$ curl -sSL http://bit.ly/2ysbOFE | bash -s
```

Verifiquem extrem a extrem la provisió d'un exemple de xarxa Fabric de dues organitzacions, amb dos peers cadascun i un «solo» ordering service. Posteriorment l'aturarem per alliberar recursos.

```
manelmdiaz@desktop-i7-linux:~$ cd fabric-samples/first-network
manelmdiaz@desktop-i7-linux:~/fabric-samples/first-network$ ./byfn.sh up
```

Aturem la plataforma HLF per alliberar recursos.

```
manelmdiaz@desktop-i7-linux:~/fabric-samples/first-network$ ./byfn.sh down
```

Hyperledger Fabric application SDK for Node.js i gulp

Descripció: plataforma per desenvolupar solucions i aplicacions DLT permissionades.

Requisit: És necessari per executar aplicacions distribuïdes per Hyperledger Fabric.

Instal·lació:

Clonem el projecte «fabric-sdk-node» i a la carpeta arrel executem diversos comandes, per instal·lar dependències, generar fitxers requerits per testejar la instal·lació.

```
manelmdiaz@desktop-i7-linux:~/$ git clone "https://gerrit.hyperledger.org/r/fabric-sdk-node"
manelmdiaz@desktop-i7-linux:~/$ cd fabric-sdk-node
manelmdiaz@desktop-i7-linux:~/fabric-sdk-node$ npm install
```

S'instal·la Gulp, generem el material crypto necessari pels test i executem els tests.

```
manelmdiaz@desktop-i7-linux:~/$ npm install -g gulp
manelmdiaz@desktop-i7-linux:~/fabric-sdk-node$ gulp install-and-generate-certs
manelmdiaz@desktop-i7-linux:~/fabric-sdk-node$ npm test
```

3.1.2 Desenvolupament Back-End

Del desenvolupament del BackEnd, s'ha definit l'estructura de dades, els mètodes necessaris i un chaincode bàsic que permet registrar nous participants, consultar les seves dades i modificar-les (això darrer només ho podrà fer un administrador). Finalment s'ha instal·lat i inicialitzat el chaincode a Hyperledger Fabric amb Hurley [31], [32].

Per tal de desenvolupar el back-end definim l'estructura de dades, els mètodes a utilitzar i el chaincode a instal·lar a Hyperledger Fabric. Finalment desenvolupem el servidor API amb els mètodes definits que recepcionarà les peticions del client front-end i s'enviaran a Hyperledger Fabric.

3.1.2.1 Estructura de dades

Es defineix a partir d'analitzar l'ús de l'aplicació pels diferents actors:

Usuaris, que poden ser els propietaris dels dispositius IoT, els consumidors de dades pròpies o de tercers o administradors del sistema.

- string user_profile_hash -> Dades de l'usuari que s'emmagatzemen en un sistema extern i a la Blockchain només es guardarà el valor hash de les dades, per tant habilita el compliment amb la GDPR.
- string role -> rol de l'usuari per distingir-ho, si és admin, usuari generador de dades o consumidor.
- certificate identities -> Els usuaris tindran un llistat de certificats per identificar-se, els quals es guardaran en un objecte 'certificate'.
- string added_by_key, date added_at i date updated_at -> s'enregistrerà la clau pública de l'organització que l'afegeix, així com la data de creació i d'actualització quan s'escaigui.

```

Structure User {
string user_profile_hash;
string role; // pot ser admin, producer or consumer
certificate identities;
string added_by_key;
date added_at;
date updated_at;
}

```

```

Structure Certificate {
integer id;
string key;
}

```

```

structure user_profile {
string email;
}

```

Dispositiu IoT, que poden mesurar diverses dades, com pressió arterial o ritme cardíac:

- string device_profile_hash -> Dades del dispositiu IoT s'emmagatzemen en un sistema extern i a la Blockchain només es guardarà el valor hash de les dades, per tant habilita el compliment amb la GDPR.
- L'estructura de dades del device_profile és la següent:
 - BD_ADDR, com identificador únic del dispositiu de 48 bits.
 - service, per identificar el tipus de dada, com pressió arterial, ritme cardíac. Utilitza 16 bits per a serveis BLE.
 - vendor, fabricant del dispositiu
 - model, model del dispositiu
- certificate identities ->Els dispositius IoT també tindran un llistat de certificats per identificar-se, els quals es guardaran en un objecte 'certificate'.
- string added_by_key, date added_at i date updated_at -> s'enregistrarà la clau pública de l'usuari que afegeix el dispositiu IoT, així com la data de creació i d'actualització quan s'escaigui.

```

Structure Device{
string device_profile_hash;
certificate identities;
string added_by_key;
date added_at;
date updated_at;
}

```

```

structure device_profile {
uint6 BD_ADDR; //48 bits for bluetooth identification
uint4 service; // 16-bit for BLE Services
}

```

Dada IoT:

- string device_data_hash -> Dades del dispositiu IoT s'emmagatzemen en un sistema extern i a la Blockchain només es guardarà el valor hash de les dades, per tant habilita el compliment amb la GDPR. L'estructura de dades del device_data és la següent:
 - BD_ADDR, com identificador únic del dispositiu de 48 bits.
 - service, per identificar el tipus de dada, com pressió arterial, ritme cardíac. Utilitza 16 bits per a serveis BLE.

- values, on s'inclou la data de quan s'ha fet la mesura i el valor de la mateixa. La mida és variable.
- string added_by_key, date added_at i date updated_at -> s'enregistrerà la clau pública del dispositiu IoT que afegeix les dades, així com la data de creació i d'actualització quan s'escaigui.

```
Structure Data{
  string device_data_hash;
  string added_by_key;
  date added_at;
  date updated_at;
}
structure device_data {
  uint6 BD_ADDR; //48 bits for bluetooth identification
  uint4 service; // 16-bit for BLE Services
  string values;
}
```

Organitzacions: Separarem els usuaris en dues organitzacions, 'OrgIoT que generen dades amb els seus dispositius i 'OrgConsD' que són usuaris externs que volen consultar dades, com metges, asseguradores o investigadors.

- string description -> Nom complet de l'organització.
- string role -> rol de l'usuari per distingir-ho.
- certificate identities -> Les organitzacions tindran diversos certificats per identificar-se, els quals es guardaran a l'objecte 'certificate'.
- string added_by_key, date added_at i date updated_at -> s'enregistrerà la clau pública de l'organització que l'afegeix, així com la data de creació i d'actualització quan s'escaigui.

```
Structure Organization {
  string description;
  certificate identities;
  string added_by_key;
  date added_at;
  date updated_at;
}
```

3.1.2.2 Mètodes

Els mètodes que necessitem són els següents:

- addUser, per registrar nous usuaris. L'organització corresponent l'haurà de donar d'alta perquè se li associi a l'organització.
- addDevice, per registrar nous dispositius que enviïn dades a HLF. L'usuari propietari serà l'únic amb permisos.
- addData, per afegir dades a HLF. Els dispositius seran els únics amb permisos.
- readUser, per llegir dades d'un usuari. Els usuaris tindran permisos.
- readData, per llegir dades dels dispositius IoT. Tindran permisos els usuaris propietaris de les dades o si l'usuari té permís del propietari de les dades.
- setUser, per actualitzar dades d'un usuari. L'usuari serà l'únic amb permisos.
- setOrganization, per actualitzar dades d'una organització. L'organització serà l'única amb permisos.

3.1.2.3 Chaincode / Smart Contract

El *chaincode* desenvolupat amb Convector Smart Contract consta de fitxers, el model i controlador corresponent al actiu i al smart contract respectivament [33], [31], [34], els quals s'ubiquen a la carpeta `packages/participant-cc/src`. El codi sencer està al capítol 6.1 Codi del Chaincode / Smart Contract.

En el model definim el nom del participant, el nom de la seva organització MSP i el seu fingerprint del certificat X.509. A continuació es mostra el codi per definir el participant, on s'especifica si el valor només es pot llegir (`@ReadOnly()`), si és requerit (`@Required()`) i com es valida (`@validate()`).

```
export class Participant extends ConvectorModel<Participant> {
  @ReadOnly()
  @Required()
  public readonly type = "io.worldsibu.participant";

  @Required()
  @Validate(yup.string())
  public name: string;

  @ReadOnly()
  @Validate(yup.string())
  public msp: string;

  @Required()
  @Validate(yup.array(x509Identities.schema()))
  public identities: Array<FlatConvectorModel<x509Identities>>;
}
```

Al controlador creem els següents mètodes:

- `register` → registre l'usuari sol·licitant, afegint la identitat de l'usuari, que conté el id, nom, msp i certificat.
- `getParticipantById` → demanar el model de participant pel seu identificador.
- `changeIdentity` → actualitza el certificat del participant.

A continuació es mostra el codi del mètode `getParticipantById`.

```
@Invokable()
public async getParticipantById(
  @Param(yup.string()) id: string
): Promise<Participant> {
  const participant = await Participant.getOne(id);
  // a non-null or defined result doesn't mean that it is valid, the following check must
  still be done
  if (!participant || !participant.id) {
    return null;
  }
  return participant;
}
```

Per tal de carregar el chaincode a HLF, primer hem de crear un package. És necessari tindre en execució tant docker com Hurley.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ npm run cc:package -- participant
org1
```

Per iniciar Hyperledger Fabric amb hurl, primer netegem si existeix una instància executant i instal·lem el chaincode participant.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ hurl clean; hurl new
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ npm run cc:start -- participant
```

```
[hurley] - installing smart contract located at /home/manelmdiaz/wearhealth-poc/convector/chaincode-participant
Installing Chaincode participant version 1.0 at org1
2020-01-11 01:52:31.281 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-01-11 01:52:31.281 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-01-11 01:52:31.332 CET [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
Installed Chaincode participant version 1.0 at org1
Installing Chaincode participant version 1.0 at org2
2020-01-11 01:52:31.417 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-01-11 01:52:31.417 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-01-11 01:52:31.463 CET [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
Installed Chaincode participant version 1.0 at org2
Instantiating Chaincode at org1 for channel chl
It may take a few minutes depending on the chaincode dependencies
2020-01-11 01:52:41.557 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escc
2020-01-11 01:52:41.558 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
Instantiated Chaincode at org1
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$
```

El resultat ha sigut satisfactori i ja podem enviar propostes de transaccions per a la seva execució a Hyperledger Fabric.

3.1.2.4 Servidor API

S'ha creat l'esquelet del servidor API i al fitxer server.js, s'ha especificat el port 8000 on fer les peticions i s'han definit els endpoints que estaran accessibles [35], [36], els quals corresponen a cadascun dels mètodes indicats al capítol 3.1.2.2 Mètodes. A continuació es mostra el codi del mètode readUser que estarà accessible a l'enllaç <http://localhost:8000/api/readUser>. S'han configurat valors estàtics que ens permetran validar el funcionament. El codi sencer està al capítol 6.2 Codi del servidor API.

```
// An api endpoint that returns data of the user
app.get('/api/readUser', (req,res) => {
  var list = ["user1", "user2", "user3"];
  res.json(list);
  console.log('Sent list of items');
});
```

Per iniciar el servidor API, s'executa la següent comanda:

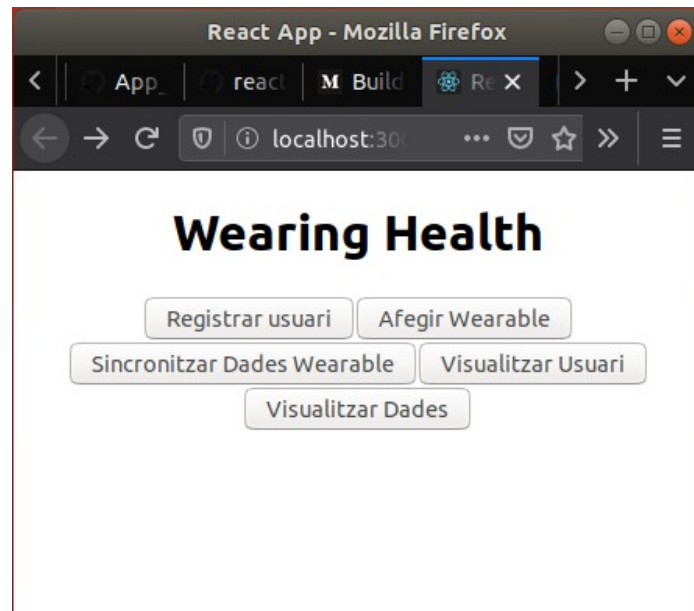
```
manelmdiaz@desktop-i7-linux:~/wear-health/server$ node server.js
```

Per validar que el servidor API respon a peticions, s'he fet la crida al API del mètode readUser amb la comanda curl i, tal com s'espera, ha retornat els valors definits al mètode.

```
["item1","item2","item3"]manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ curl
http://localhost:8000/api/readUser
["user1","user2","user3"]manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$
```

3.1.2.5 Desenvolupament Front-End

S'ha creat l'esquelet de l'aplicació client i servidor API [37], i provat que es comuniquen. Posteriorment s'ha actualitzat el client per tal de mostrar les funcions que permet, encara que no és funcional i resta desenvolupar un cop estigui desenvolupat el backend i comprovada la connexió amb Hyperledger Fabric. S'ha creat el següent disseny amb botons per a diverses accions:



Per a cada acció o pàgina a mostrar al client, com la pàgina principal, registrar usuari o visualitzar usuari entre d'altres, s'ha de crear un fitxer .js a la carpeta /home/manelmdiaz/wearhealth-poc/client/src/App/pages. La pàgina principal és defineix al fitxer Homepage.js, que enllaça a cadascuna de les pàgines a través de botons que criden a les diferents pàgines [38]. En aquesta PoC s'ha desenvolupat el botó 'Visualitzar Usuari' i la pàgina corresponent UserRead.js, per tal de fer proves. El codi sencer està al capítol 6.3 Codi del client.

Al homepage.js es defineix el botó 'Visualitzar Usuari' que obra la pàgina UserRead.js amb el codi:

```
<Link to={'./UserRead'}>
  <button variant="raised">
    Visualitzar Usuari
  </button>
</Link>
```

Al UserRead.js es defineix la crida al servidor API readUser i s'executa de forma async a la espera de resposta amb l'array de dades [39].

```
callApi = async (a) => {
  let response;
  response = await fetch('/api/readUser');
  const body = await response.json(); if (response.status !== 200) throw
  Error(body.message); return body;
};
```

```
// Retrieves the list of items from the Express app
readUser(){
  this.callApi()
  .then(res => {
    let user = res;
    this.setState({user: res});
  })
  .catch(err => console.log(err));
}
```

Per tal que el client conegui les pàgines existents, i mostri la pagina corresponent, s'han d'importar i indicar les rutes al fitxer /home/manelmdiaz/wearhealth-poc/client/src/App/App.js, amb el codi:

```
import { Route, Switch } from 'react-router-dom';
import Homepage from './pages/Homepage';
import UserRead from './pages/UserRead';
```

```
<div>
  <Switch>
    <Route exact path="/" component={Homepage}/>
    <Route path="/UserRead" component={UserRead}/>
  </Switch>
</div>
```

La definició de l'estil es desa a /home/manelmdiaz/wearhealth-poc/client/src/App/App.css. Exemple d'un estil:

```
.App-header {
  background-color: #222;
  height: 150px;
  padding: 20px;
  color: white;
}
```

Per tal d'iniciar el client React i el servidor api, s'executa el següent:

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/client$ npm start
```

```
Compiled successfully!

You can now view client in the browser.

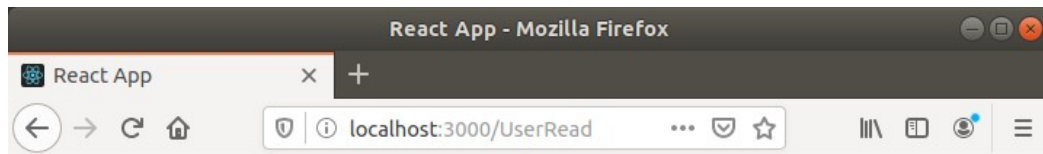
Local:      http://localhost:3000/
On Your Network:  http://10.0.2.15:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/server$ node server.js
```

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/server$ node server.js
App is listening on port 8000
```

Per validar que el client s'inicia i repcepciona correctament les dades servidor API, s'ha executat 'Visualitzar Usuari'.



User to register

user1
user2
user3

3.2 Proves de la PoC

Primer instal·larem les eines per disposar de l'entorn de treball per desenvolupar la PoC i desplegar-la.

3.2.1 Definició del pla de proves

Les proves són les següents:

- Instal·lar i inicialitzar un chaincode.
- Registrar un participant.
- Llegir les dades d'un participant registrat.
- No es permet registrar un participant amb un id existent.
- No es permet canviar la identitat d'un participant si el sol·licitant no té l'atribut admin.
- Modificar la identitat d'un participant si el sol·licitant té l'atribut admin.

Aquestes proves es realitzen directament sobre Hyperledger Fabric i estan adequades al controlador del chaincode desenvolupat per la PoC. S'ha creat el fitxer `/home/manelmdiaz/wearhealth-poc/convector/packages/participant-cc/tests/participantController.spec.ts`.

Per tal d'interactuar amb Hyperledger s'han de cridar els mètodes a través del controlador, en aquest cas s'utilitza participantCtrl on es declara i defineix amb el controlador client 'ConvectorControllerClient' [34]. També s'han creat dades fictícies, com identitats (*fingerprints*) i certificats per a l'usuari administrador i participants, que s'utilitzen per fer comprovacions enlloc de consultar-ho al MSP. El codi sencer està al capítol 6.4 Codi del Pla de proves. A continuació es mostra part del codi on es fan els imports, es defineix participantCtrl i s'inicialitza el controlador.

```
import { MockControllerAdapter } from '@worldsibu/convector-adapter-mock';
import { ClientFactory, ConvectorControllerClient } from '@worldsibu/convector-core';
import { Participant, ParticipantController } from '../src';

describe('Participant', () => {
  let mockAdapter: MockControllerAdapter;
  let adapter: MockControllerAdapter;
  let participantCtrl: ConvectorControllerClient<ParticipantController>;

  before(async () => {
    // Mocks the blockchain execution environment
    adapter = new MockControllerAdapter();
    participantCtrl = ClientFactory(ParticipantController, adapter);

    await adapter.init([
      {
        version: '1*',
        controller: 'ParticipantController',
        name: join(__dirname, '..')
      }
    ]);
  });
});
```

Per a cada prova es desenvolupen el codi a testejar, en el següent cas és comprova el registrar un participant, s'utilitza el mètode register() i es comprova que la el participant existeixi amb les dades fictícies d'identitat.

```
it("hauria de registrar un participant", async () => {
  // Create participant1
  await participantCtrl.register("Participant1", "Participant1Name");
  const participant1 = await participantCtrl
    .getParticipantById ("Participant1")
    .then (result => {
      return new Participant(result);
    });

  expect(participant1).to.include(
    {"id":"Participant1","type":"io.worldsibu.participant","name":"Participant1Name","msp"
    ::"dummymspId"}
  );

  expect(participant1.identities).to.deep.members(
    [{"fingerprint":fakeFingerprint1,"status":true}]
  );
});
```

3.2.2 Criteris d'acceptació del pla de proves

La finalització amb resultat satisfactori de les proves, és el criteri per acceptar el pla de proves.

3.2.3 Execució del pla de proves

S'han realitzat les següents proves directament sobre Hyperledger Fabric.

1. A través de la línia de comanda:

Llistat de proves:

- Prova.1.1: Instal·lar i inicialitzar un chaincode.

Execució:

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ hurl clean; hurl new
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ npm run cc:start -- participant
```

Resultat: Finalització amb èxit. El detall del resultat es troba al capítol 6.8.1-Resultat de proves a la línia de comandes.

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ npm run cc:start -- participant
> convector@1.0 cc:install /home/manelmdiaz/wearhealth-poc/convector
> f() { hurl install ${2:-$1} node -P ./chaincode-$1; }; f "participant"

[hurley] - installing smart contract located at /home/manelmdiaz/wearhealth-poc/convector/chaincode-participant
Installing Chaincode participant version 1.0 at org1
2020-01-01 21:27:07.242 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escv
2020-01-01 21:27:07.242 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-01-01 21:27:07.293 CET [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
Installed Chaincode participant version 1.0 at org1
Installing Chaincode participant version 1.0 at org2
2020-01-01 21:27:07.383 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escv
2020-01-01 21:27:07.383 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
2020-01-01 21:27:07.421 CET [chaincodeCmd] install -> INFO 003 Installed remotely response:<status:200 payload:"OK" >
Installed Chaincode participant version 1.0 at org2
Instantiating Chaincode at org1 for channel ch1
It may take a few minutes depending on the chaincode dependencies
2020-01-01 21:27:17.487 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using default escv
2020-01-01 21:27:17.487 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using default vscc
Instantiated Chaincode at org1
```

2. Fitxer de proves 'participantController.spec.ts'

Llistat de proves:

- Prova.2: Registrar un participant.
- Prova.3: Llegir les dades d'un participant registrat.
- Prova.4: No es permet registrar un participant amb un id existent.
- Prova.5: No es permet canviar la identitat d'un participant si el sol·licitant no té l'atribut admin.
- Prova.6: Modificar la identitat d'un participant si el sol·licitant té l'atribut admin.

Execució:

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ npm test
```

Resultat:

Finalització amb èxit. El detall del resultat es troba al capítol 6.8.2-Resultat de proves amb fitxer.

```
participant-cc: 5 passing (133ms)
lerna success run Ran npm script 'test' in 1 package in 2.3s:
lerna success - participant-cc
```


3.2.4 Anàlisi de les proves

Les proves han sigut bàsiques i consisteixen en instal·lar un chaincode a Fabric, la seva instanciació en un dels canals, enregistrar un participant, canviar la seva identitat i mostrar les dades d'un participant.

En quan a seguretat, s'ha validat que no es permeti registrar un participant ja existent o canviar la identitat si no és un usuari amb rol d'administrador, per tant es podria modelar perquè només sigui el participant propietari d'aquesta identitat.

4 Conclusions

Aquest treball m'ha permès obtenir una visió de l'estat actual de les tecnologies DLT. Des d'un punt de vista de negoci, considero que les tecnologies DLT ens poden permetre assolir nous requisits en projectes que els sistemes actuals no ho permeten o millorar els existents, en l'àmbit de la logística, les entitats financeres i la salut, com la confiança entre actors, l'automatització d'acords, evitar frauds, millorar el temps en tràmits i la seguretat de dades en compliment amb la llei GDPR. Des d'un punt de vista tècnic, el coneixement adquirit del funcionament a alt nivell de diferents tecnologies DLT i l'aprofundiment en el disseny d'una blockchain privada com és Hyperledger Fabric, m'ha ajudat a entendre el seu comportament i limitacions tecnològiques, necessari alhora de dissenyar una solució.

Els dispositius IoTs wearables, actualment no poden substituir a instruments mèdics, degut a la seva menor fiabilitat, encara que la recollida de dades i el potencial ús per científics i altres actors, pot impactar positivament en l'interès de millorar els wearables. En aquest cas, és clar, la sinergia amb la intel·ligència artificial que s'enriqueix de les dades massiva que generen els dispositius IoTs i la informació resultant que podria donar als diferents actors.

Considero que existeixen diversos reptes que s'han d'aprofundir, entre el quals el més important és la seguretat i privacitat de les dades, degut a que les dades encriptades no s'esborren, en un futur es puguin desencriptar o relacionar amb l'usuari propietari, gràcies a la computació quàntica o falles de seguretat. Segons un estudi del *National Institute of Standards and Technology* [40], la computació quàntica farà obsoletes els sistemes de claus públiques, i l'encriptació simètrica necessitaria paraules de pas més llargues. Llavors es pot pensar que solucions més descentralitzades com Holochain seran més segures i privades, però dependrà de la seguretat dels dispositius i de l'ús de l'usuari.

Per tant, a partir de la recerca d'informació realitzada en aquest treball i l'existència del cas 'Change Healthcare' [22] amb capacitat de tractar gran quantitat de dades, considero que és viable la solució dissenyada al treball per realitzar una aplicació distribuïda del monitoratge de salut amb wearables.

Per finalitzar, voldria mencionar que encara que hi ha disponible informació del programari utilitzat per aquesta PoC, ha sigut un repte aconseguir desenvolupar la PoC per realitzar proves contra Hyperledger Fabric i, resta pendent una gran part de la solució per desenvolupar. Per tant tenim les següents línies de treball futures:

- L'aplicació sigui desplegable en un entorn de proves o una plataforma descentralitzada.
- Connectar amb un dispositiu IoT físic o simulat.
- Recollir un tipus de dades d'un dispositiu IoT físic.
- Desar dades en un sistema d'emmagatzematge distribuït.
- Llegir dades d'un sistema d'emmagatzematge distribuït.
- Compartir dades privades amb un altre usuari.
- Un usuari comparteixi dades privades amb un altre de forma anònima.
- Pla de proves per validar compliment del tractament de dades General Data Protection Regulation (GDPR).
- Estudi de les implicacions de la computació quàntica en la seguretat de dades encriptades.

5 Bibliografía

- [1] «Organización Mundial de la Salud», 2019. [En línea]. Disponible a: <https://www.who.int/es/about/what-we-do/>. [Consulta: 23-set-2019].
- [2] «Prevalencia, diagnóstico, tratamiento y control de la hipertensión arterial en España. Resultados del estudio Di@bet.es», *Revista Española de Cardiología*, vol. 69, núm. 6, p. 572-578, juny 2016.
- [3] «International Society of Hypertension», 2019. [En línea]. Disponible a: <http://ish-world.com>. [Consulta: 22-set-2019].
- [4] «Estadística de Gasto Sanitario Público», 2016. [En línea]. Disponible a: <http://www.mscbs.gob.es/estadEstudios/estadisticas/docs/EGSP2008/egspPrincipalesResultados.pdf>. [Consulta: 20-set-2019].
- [5] *Clickup - Eina col·laborativa i de gestió de projectes*. 2018.
- [6] H. F. A. Gary B. Wills, «Intersections between IoT and distributed ledger», *Advances in Computers*, p. 1-41, 2019.
- [7] Vitalik Buterin, «The Meaning of Decentralization», 2017. [En línea]. Disponible a: <https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274>. [Consulta: 19-oct-2019].
- [8] Parikshit Hooda, «Comparison – Centralized, Decentralized and Distributed Systems», 2018. [En línea]. Disponible a: <https://www.geeksforgeeks.org/comparison-centralized-decentralized-and-distributed-systems/>. [Consulta: 20-nov-2019].
- [9] Jiyad Ahsan, «Centralized vs. Decentralized: The Best (and worst) of Both Worlds», 2018. [En línea]. Disponible a: <https://hackernoon.com/centralization-vs-decentralization-the-best-and-worst-of-both-worlds-7bfdd628ad09>. [Consulta: 15-oct-2019].
- [10] *[GDPR] Reglamento general de protección de datos*. 2016.
- [11] E. A. Jason Yellick, «Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains», abr-2018. [En línea]. Disponible a: <https://arxiv.org/pdf/1801.10228.pdf>. [Consulta: 05-oct-2019].
- [12] Nxt community, «Nxt Whitepaper», 12-jul-2014. [En línea]. Disponible a: <https://whitepaperdatabase.com/wp-content/uploads/2018/03/Nxt-NXT-Whitepaper.pdf>. [Consulta: 05-oct-2019].
- [13] «Tangle Monitor», *TangleMonitor - Live visualisation and metrics of the IOTA Tangle*. [En línea]. Disponible a: <https://tanglemonitor.com/>. [Consulta: 05-oct-2019].
- [14] «Swirlds Solutions», *Solutions - Swirlds*, 2019. [En línea]. Disponible a: <https://www.swirlds.com/solutions/>. [Consulta: 07-oct-2019].
- [15] L. B. Paul Madsen, «Hedera Whitepaper», feb-2019. [En línea]. Disponible a: <https://www.hedera.com/hh-whitepaper-v1.5-190219.pdf>. [Consulta: 10-oct-2019].
- [16] E. H.-B. Arthur Brock, «HoloChain WhitePaper», 15-feb-2018. [En línea]. Disponible a: <https://github.com/holochain/holochain-proto/blob/whitepaper/holochain.pdf>. [Consulta: 15-oct-2019].
- [17] David Atkinson, «Learning more about Holochain», *Holochain*, 2019. [En línea]. Disponible a: <https://holo.host/wp-content/uploads/holochain-learning-more.pdf>. [Consulta: 15-oct-2019].
- [18] Arthur Brock, «Holochain has way more privacy options than blockchain», 29-març-2018. [En línea]. Disponible a: <https://twitter.com/artbrock/status/979501125354188800?lang=es>. [Consulta: 17-oct-2019].
- [19] «Asymmetric Encryption», ago-2019. [En línea]. Disponible a: <https://www.gnupg.org/>. [Consulta: 15-oct-2019].
- [20] «GOOGLE PRIVACY POLICY», 22-gen-2019. [En línea]. Disponible a: https://www.gstatic.com/policies/privacy/pdf/20190122/f3294e95/google_privacy_policy_en_eu.pdf. [Consulta: 25-oct-2019].
- [21] Hyperledger, «Hyperledger Indy -Zero-Knowledge Proof», *Research Paper: Validating Confidential Blockchain Transactions with Zero-Knowledge Proof*, 2019. [En línea].

- Disponible a: <https://www.hyperledger.org/blog/2019/04/25/research-paper-validating-confidential-blockchain-transactions-with-zero-knowledge-proof>. [Consulta: 15-oct-2019].
- [22] «Case Study:Change Healthcare using Hyperledger Fabric», gen-2018. [En línea]. Disponible a: https://www.hyperledger.org/wp-content/uploads/2019/06/Hyperledger_CaseStudy_ChangeHealthcare_Printable_6.19.pdf. [Consulta: 02-nov-2019].
- [23] Hyperledger, «MSP Implementation with Identity Mixer», 2019. [En línea]. Disponible a: <https://hyperledger-fabric.readthedocs.io/en/release-1.4/idemix.html>. [Consulta: 05-oct-2019].
- [24] Bluetooth SIG, «GATT Characteristics», *GATT Characteristics*, 2019. [En línea]. Disponible a: <https://www.bluetooth.com/specifications/gatt/characteristics/>. [Consulta: 20-oct-2019].
- [25] A. S. Daniele Gobbetti, *Gadgetbridge*.
- [26] «www.withings.com», 2019. [En línea]. Disponible a: www.withings.com. [Consulta: 15-oct-2019].
- [27] «www.zublaze.com», 2019. [En línea]. Disponible a: www.zublaze.com. [Consulta: 15-oct-2019].
- [28] «Sistema flash de monitorización de glucosa», 2019. [En línea]. Disponible a: www.freestylelibre.es/libre/. [Consulta: 15-oct-2019].
- [29] Apple Inc., «One Drop Chrome Blood Glucose Monitoring Kit», 2019. [En línea]. Disponible a: <https://www.apple.com/shop/product/HMN02LL/A/one-drop-chrome-blood-glucose-monitoring-kit>. [Consulta: 15-oct-2019].
- [30] «Xiaomi», 2019. [En línea]. Disponible a: www.mi.com/es/. [Consulta: 15-oct-2019].
- [31] Covalent, «Tutorial - Back end - Convector Suite», 2019. [En línea]. Disponible a: <https://docs.covalentx.com/article/95-tutorial-back-end>. [Consulta: 25-des-2019].
- [32] Covalent, «Tutorial - Smart Contract», 2019. [En línea]. Disponible a: <https://docs.covalentx.com/article/89-tutorial>. [Consulta: 25-des-2019].
- [33] Covalent, «Covalent Documentation - Convector Smart contracts and Hurley», 2019. [En línea]. Disponible a: <https://docs.covalentx.com/article/71-getting-started>. [Consulta: 25-oct-2019].
- [34] Aboubakar Koïta, «Develop a Hyperledger Fabric application with Convector Suite», 15-maig-2019. [En línea]. Disponible a: <https://medium.com/@aboubakarkoita/develop-a-hyperledger-fabric-application-with-convector-suite-1a2fa8f01011>. [Consulta: 02-nov-2019].
- [35] KC Tam, «An Implementation of API Server for Hyperledger Fabric Network», 04-set-2019. [En línea]. Disponible a: <https://medium.com/@kctheservant/an-implementation-of-api-server-for-hyperledger-fabric-network-8764c79f1a87>. [Consulta: 25-nov-2019].
- [36] Varun Raj, «Setting up RESTful API Server for Hyperledger Fabric With NodeJS SDK», 26-març-2018. [En línea]. Disponible a: <https://www.skcript.com/svr/setting-up-restful-api-server-for-hyperledger-fabric-with-nodejs-sdk/>. [Consulta: 20-nov-2019].
- [37] Rudy Aldair Rocha González, «Tutorial de Node JS API + create-react-app [+ES6, +Heroku Deploy]», 05-abr-2018. [En línea]. Disponible a: <https://medium.com/@rudyrochag/tutorial-de-node-js-api-create-react-app-12aa8b03b6f9>. [Consulta: 20-nov-2019].
- [38] Jack Hickey, «Build an insurance application with Hyperledger Composer and React.js Part 2». [En línea]. Disponible a: <https://medium.com/coinmonks/build-an-insurance-application-with-hyperledger-composer-and-react-js-part-2-da27a9dfdee5>.
- [39] Robin Wieruch, «How to manage React State with Arrays - RWieruch», 04-oct-2018. [En línea]. Disponible a: <https://www.robinwieruch.de/react-state-array-add-update-remove>. [Consulta: 26-des-2019].
- [40] L. C. Daniel Smith-Tone, «Report on Post-Quantum Cryptography», 2016.

6 Annexos

6.1 Codi del Chaincode / Smart Contract

Codi del model

Ruta: /home/manelmdiaz/wearhealth-poc/convector/packages/participant-cc/src/participant.model.ts

```
import * as yup from "yup";
import {
  ConvectorModel,
  ReadOnly,
  Required,
  Validate,
  FlatConvectorModel
} from "@worldsibu/convector-core-model";

export class x509Identities extends ConvectorModel<x509Identities>{
  @ReadOnly()
  public readonly type = 'io.worldsibu.examples.x509identity';

  @Validate(yup.boolean())
  @Required()
  status: boolean;
  @Validate(yup.string())
  @Required()
  fingerprint: string;
}

export class Participant extends ConvectorModel<Participant> {
  @ReadOnly()
  @Required()
  public readonly type = "io.worldsibu.participant";

  @Required()
  @Validate(yup.string())
  public name: string;

  @ReadOnly()
  @Validate(yup.string())
  public msp: string;

  @Required()
  @Validate(yup.array(x509Identities.schema()))
  public identities: Array<FlatConvectorModel<x509Identities>>;
}
```

Codi del controlador

Ruta: /home/manelmdiaz/wearhealth-poc/convector/packages/participant-cc/src/participant.controller.ts

```
import * as yup from 'yup';

import {
  Controller,
  ConvectorController,
  Invokable,
  Param,
  BaseStorage
} from '@worldsibu/convector-core';

import { ChaincodeTx } from '@worldsibu/convector-platform-fabric';
```

```

import { Participant } from './participant.model';
import { ClientIdentity } from 'fabric-shim';

@Controller('participant')
export class ParticipantController extends ConvectorController<ChaincodeTx> {
  @Invokable()
  public async register(
    @Param(yup.string())
    id: string,
    @Param(yup.string())
    name: string
  ) {
    // Retrieve to see if exists
    const existing = await Participant.getOne(id);

    if (!existing || !existing.id) {
      let participant = new Participant();
      participant.id = id;
      participant.name = name || id;
      participant.msp = this.tx.identity.getMSPID();
      // Create a new identity
      participant.identities = [{
        fingerprint: this.sender,
        status: true
      }];
      console.log(JSON.stringify(participant));
      await participant.save();
    } else {
      throw new Error('Identity exists already, please call changelIdentity fn for updates');
    }
  }

  @Invokable()
  public async getParticipantById(
    @Param(yup.string()) id: string
  ): Promise<Participant> {
    const participant = await Participant.getOne(id);
    // a non-null or defined result doesn't mean that it is valid, the following check must still be done
    if (!participant || !participant.id) {
      return null;
    }
    return participant;
  }

  @Invokable()
  public async changelIdentity(
    @Param(yup.string())
    id: string,
    @Param(yup.string())
    newIdentity: string
  ) {
    // Check permissions
    let isAdmin = this.tx.identity.getAttributeValue('admin');
    console.log(this.tx.identity);
    console.log(isAdmin);
    let requesterMSP = this.tx.identity.getMSPID();

    // Retrieve to see if exists
    const existing = await Participant.getOne(id);
    console.log('Existing participant:');
    console.log(existing);
    if (!existing || !existing.id) {
      throw new Error('No identity exists with that ID');
    }

    console.log(`existing.msp=${existing.msp} requesterMSP=${requesterMSP}`);
    if (existing.msp != requesterMSP) {
      throw new Error('Unauthorized. MSPs do not match');
    }
  }
}

```

```

console.log(`isAdmin=${isAdmin}`);
if (!isAdmin) {
  throw new Error('Unauthorized. Requester identity is not an admin');
}

// Disable previous identities!
existing.identities = existing.identities.map(identity => {
  identity.status = false;
  return identity;
});

// Set the enrolling identity
existing.identities.push({
  fingerprint: newIdentity,
  status: true
});
await existing.save();
}
@Invokable()
public async get(
  @Param(yup.string())
  id: string
) {
  const existing = await Participant.getOne(id);
  if (!existing || !existing.id) {
    throw new Error(`No identity exists with that ID ${id}`);
  }
  return existing;
}
}
}

```

6.2 Codi del servidor API

Codi del fitxer server.js del servidor API.

Ruta /home/manelmdiaz/wearhealth-poc/server/server.js

```

const express = require('express');
const path = require('path');

const app = express();
const port = 8000;

// Serve the static files from the React app
app.use(express.static(path.join('/home/manelmdiaz/wearhealth-poc/client/build')));

// An api endpoint that add a user
app.get('/api/addUser', (req,res) => {
  var list = ["item1", "item2", "item3"];
  res.json(list);
  console.log('Send new user to register');
});

// An api endpoint that add a device
app.get('/api/addDevice', (req,res) => {
  var list = ["item1", "item2", "item3"];
  res.json(list);
  console.log('Send new user to register');
});

// An api endpoint that returns data of the user
app.get('/api/readUser', (req,res) => {
  var list = ["user1", "user2", "user3"];
  res.json(list);
  console.log('Sent list of items');
});

```

```

// An api endpoint that add a data
app.get('/api/addData', (req,res) => {
  var list = ["item1", "item2", "item3"];
  res.json(list);
  console.log('Send new user to register');
});

// An api endpoint that returns data
app.get('/api/readData', (req,res) => {
  var list = ["item1", "item2", "item3"];
  res.json(list);
  console.log('Sent list of items');
});

// An api endpoint that change user's data
app.get('/api/setUser', (req,res) => {
  var list = ["item1", "item2", "item3"];
  res.json(list);
  console.log('Sent list of items');
});

// An api endpoint that change organization's data
app.get('/api/setOrganization', (req,res) => {
  var list = ["item1", "item2", "item3"];
  res.json(list);
  console.log('Sent list of items');
});

</Route path='/UserAdd' component={UserAdd}>
</Route path='/DeviceAdd' component={DeviceAdd}>
</Route path='/DataAdd' component={DataAdd}>
</Route path='/UserRead' component={UserRead}>
</Route path='/DataRead' component={DataRead}>

// Handles any requests that don't match the ones above
app.get('*', (req,res) =>{
  res.sendFile(path.join('/home/manelmdiaz/wearhealth-poc/client/public/index.html'));
});

app.listen(port);

console.log('App is listening on port ' + port);

```

6.3 Codi del client

Codi del fitxer App.js del client.

Ruta /home/manelmdiaz/wearhealth-poc/client/src/App/App.js

```

import React, { Component } from 'react';
import { Route, Switch } from 'react-router-dom';
import './App.css';
import Homepage from './pages/Homepage';
import UserAdd from './pages/UserAdd';
import DeviceAdd from './pages/DeviceAdd';
import DataAdd from './pages/DataAdd';
import UserRead from './pages/UserRead';
import DataRead from './pages/DataRead';

class App extends Component {

```



```

state = {
  name: "joe",
  assets: [],
  user: {}
}

```

```

render() {
  const App = () => (
    <div>
      <Switch>
        <Route exact path="/" component={Homepage}/>
        <Route path="/UserAdd" component={UserAdd}/>
        <Route path="/DeviceAdd" component={DeviceAdd}/>
        <Route path="/DataAdd" component={DataAdd}/>
        <Route path="/UserRead" component={UserRead}/>
        <Route path="/DataRead" component={DataRead}/>
      </Switch>
    </div>
  )
  return (
    <Switch>
      <App/>
    </Switch>
  );
}
}

```

```
export default App;
```

Codi del fitxer Homepage.js del client.

Ruta /home/manelmdiaz/wearhealth-poc/client/src/App/pages/Homepage.js

```

import React, { Component } from 'react';
import { Link } from 'react-router-dom';
import './App.css';

```

```

class Homepage extends Component {
  render() {
    return (
      <div className="App">
        <h1 className={styles.headerStyle} >Wearing Health</h1>
        { /* Link to User.js */ }
        <Link to='./UserAdd'>
          <button variant="raised">
            Registrar usuari
          </button>
        </Link>

        { /* Link to Data.js */ }
        <Link to='./DeviceAdd'>
          <button variant="raised">
            Afegir Wearable
          </button>
        </Link>

        { /* Link to Data.js */ }

```

```

<Link to='./DataAdd'>
  <button variant="raised">
    Sincronitzar Dades Wearable
  </button>
</Link>

  { /* Link to Data.js */ }
  <Link to='./UserRead'>
    <button variant="raised">
      Visualitzar Usuari
    </button>
  </Link>

  { /* Link to Data.js */ }
  <Link to='./DataRead'>
    <button variant="raised">
      Visualitzar Dades
    </button>
  </Link>

</div>
);
}
}

//PropTypes
Homepage.propTypes = {
}

export default Homepage;

```

Codi de la pàgina UserRead.js

Ruta /home/manelmdiaz/wearhealth-poc/client/src/App/pages/UserRead.js

```

import React, { Component } from 'react';
import './App.css';

class UserRead extends Component {
  // Initialize the state
  constructor(props){
    super(props);
    this.state = {
      list: ['test', 'test']
    }
  }

  // Fetch the list on first mount
  componentDidMount() {
    this.readUser();
  }

  callApi = async (a) => {
    let response;
    response = await fetch(`/api/userRead`);
    const body = await response.json(); if (response.status !== 200) throw Error(body.message);
    return body;
  };

  // Retrieves the list of items from the Express app
  readUser(){
    this.callApi()
    .then(res => {
      let list = res.list;
      this.setState({ list })
    })
  }
}

```

```

    })
    .catch(err => console.log(err));
  }

  render() {
    const { list } = this.state;

    return (
      <div className=".headerStyle">
        <h1>User to register</h1>
        { /* Check to see if any items are found */ }
        { list.length ? (
          <div>
            { /* Render the list of items */ }
            { list.map((item) => {
              return(
                <div>
                  {item}
                </div>
              );
            }) }
          </div>
        ) : (
          <div>
            <h2>No User to register</h2>
          </div>
        )
      )
    );
  }
}

export default UserRead;

```

Codi del fitxer App.css del client.

Ruta /home/manelmdiaz/wearhealth-poc/client/src/App/App.css

```

.App {
  text-align: center;
}

.App-logo {
  animation: App-logo-spin infinite 20s linear;
  height: 80px;
}

.App-header {
  background-color: #222;
  height: 150px;
  padding: 20px;
  color: white;
}

.App-title {
  font-size: 1.5em;
}

.App-intro {

```



```

"azELMAKGA1UEBhMCMVVMxEzARBgNVBAgTCkNhbgGimb3JuaWEXFjAUBGNVBAcTDVNh\n"
+
"biBGcmFuY2lzY28xDzANBgNVBAsTBmNsaWVudDEeMBwGA1UEAwwVZXNlcjFab3Jn\n" +
"MS5odXJsZXkubGFIMFkwEwYHkoZlZjOCAQYIKoZlZjODACDQgAE5QS5zZd5kllr\n" +
"lCceMAshpkryJr3LKlev/fblhc76C6x6jfbWsYx4eilqDKGmGtoP/DL/ubiHtWxW\n" +
"ncRs5tuu7KNNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCAAwKwYDVR0\n"
n" +
"BCQwloAgOrfdQBvYqeJMP2kSeYMs454SgMM0UMxVMX3smJhq1T0wCgYIKoZlZjOE\n" +
"AwlDSAawRQlhAKuLQTEpu7OUJVepcKR8/4agjQzP5m5dbyOhZUPI7HKzAiBromln\n" +
"dH9+KtMkM6VNbtSP54kS5idQg+1IXSal76P98A==\n" +
"-----END CERTIFICATE-----\n";
const fakeAdminCert =
"-----BEGIN CERTIFICATE-----\n" +
"MIIC7DCCApOgAwIBAgIUc3DffC8hY03iz6zRC6GZQUch7EwCgYIKoZlZjOEAwIw\n" +
"cTELMAGKGA1UEBhMCMVVMxEzARBgNVBAgTCkNhbgGimb3JuaWEXFjAUBGNVBAcTDVNh\n"
+
"biBGcmFuY2lzY28xGDAWBGNVBAoTD29yZzEuaHVybGV5LmxhYjEjEjEjEjEjEj\n"
+
"Y2Eub3JnMS5odXJsZXkubGFIMB4XDTE5MDUwNjA4NDEwMFoXDTIwMDUwNTA4NDYw\n"
+
"MFowfzELMAKGA1UEBhMCMVVMxZAVBgNVBAgTDk5vcnRlENhcm9saW5hMRQwEgYD\n"
+
"VQKKEwtleXBkcmxIZGdlcjEwMA0GA1UECXMGY2xpZW50MAAsGA1UECXMeb3JnMTAS\n" +
"BgNVBAsTC2RlcGFydG1lbnQxMQ8wDQYDVQQDEwZhZG1pbjIwWTATBgqhkJOPQIB\n" +
"BggqhkJOPQMBBwNCAATdhgd0fRPq4AYSvS9tiS7vcZamCG3PDAb0QM4UGyFADdW\n" +
"RsQjglz2/MnlD4rLkU6srIAJUHdZI+QYGGkDhZiBo4H6MIH3MA4GA1UdDwEB/wQ\n" +
"AwIHgDAMBGNVHRMBAf8EAJAAMB0GA1UdDgQWBBSbHq5DcRccBt0+y4miDuzL0q80\n"
+
"8TArBgNVHSMJEJAigCD9XKUjIbuooHek1fmgE768dWtKHdGppGn8v/YEeBbyDAR\n" +
"BgNVHREECjAlgGZ1YnVudHUweAYIKgMEBQYHCAEEbHsiYXR0cnMiOnsiYWRtaW4i\n" +
"OiJ0cnVliwiaGYuQWZmaWxpYXRpb24iOiJvcmcxLmRlcGFydG1lbnQxliwiaGYu\n" +
"RW5y2xsbWVudEIEljoIYWRtaW4yYliwiaGYuVHlwZSI6ImNsaWVudCJ9fTAKBgq\n" +
"hkjoPQQDAgNHADBEAiAzUQos0hPVPf3DuZaCW3gX+LlxL2G5d7iY1ZU1murgwlg\n" +
"dkQIssMaMwkireuglUubT/Chee4jfGnhJqffnG+qCHs=\n" +
"-----END CERTIFICATE-----\n";

before(async () => {
  // Mocks the blockchain execution environment
  adapter = new MockControllerAdapter();
  participantCtrl = ClientFactory(ParticipantController, adapter);

  await adapter.init([
    {
      version: '*',
      controller: 'ParticipantController',
      name: join(__dirname, '..')
    }
  ]);
});

it("hauria de registrar un participant", async () => {
  // Create participant1
  await participantCtrl.register("Participant1", "Participant1Name");
  const participant1 = await participantCtrl
    .getParticipantById ("Participant1")
    .then (result => {
      return new Participant(result);
    });
  expect(participant1).to.include(
    {"id": "Participant1", "type": "io.worldsibu.participant", "name": "Participant1Name", "msp": "dummy mspld"}
  );
  expect(participant1.identities).to.deep.members(
    [{"fingerprint": "fakeFingerprint1", "status": true}
  ]);
});

```

```

it("hauria de llegir dades d'un participant registrat", async() => {
  // Get participant1 information
  await participantCtrl.getParticipantById("Participant1");
  const participant1 = await participantCtrl
    .getParticipantById("Participant1")
    .then (result =>{
      return new Participant(result);
    });
  expect(participant1).to.include(
    {"id": "Participant1", "name": "Participant1Name"}
  );
});

it("no hauria de registrar un participant si existeix el id", async () => {
  // Create participant1 ja existent
  await rejects(participantCtrl.register("Participant1", "Participant1Name"));
});

it("no hauria de permetre canviar la identitat d'un participant si el participant no té atribut admin",
  async () => {
  // Create participant1 ja existent
  await rejects(participantCtrl.changeIdentity("Participant1", fakeFingerprint2));
});

it("Hauria de permetre modificar la identitat d'un participant si el participant té atribut admin",
  async () => {
  //Create Participant2
  await participantCtrl.register("Participant2", "Participant2Name");

  const participant2 = await participantCtrl
    .getParticipantById ("Participant2")
    .then (result => {
      return new Participant(result);
    });
  expect(participant2).to.include({
    id: "Participant2",
    name: "Participant2Name",
  });
  expect(participant2.identities).to.deep.members(
    [{"fingerprint":fakeFingerprint1,"status":true}]
  );
  // Change identity of Participant
  // the method check that an admin identity is required to change the identity of a participant
  (adapter.stub as any).usercert = fakeAdminCert;
  await participantCtrl.changeIdentity(
    "Participant2",
    fakeFingerprint2
  );

  const participant2Updated = await participantCtrl
    .getParticipantById ("Participant2")
    .then (result => {
      return new Participant(result);
    });
  expect(participant2Updated).to.include({
    id: "Participant2",
    name: "Participant2Name"
  });
  expect(participant2.identities).to.deep.members(
    [{"fingerprint":fakeFingerprint1,"status":true}]
  );
  expect(participant2Updated.identities).to.deep.members(
    [
      {"fingerprint":fakeFingerprint1,"status":false},

```

```

    {"fingerprint":fakeFingerprint2,"status":true}
  ]
  );
});
});

```

6.5 Resultat d'instal·lar Hurley

```

manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector/chaincode-participant$ npm i -g
@worldsibu/hurley
npm WARN deprecated core-js@2.6.11: core-js@<3 is no longer maintained and not
recommended for usage due to the number of issues. Please, upgrade your dependencies to the
actual version of core-js@3.
npm WARN deprecated ts-simple-ast@11.3.0: NOTICE: ts-simple-ast has been renamed to ts-
morph and version reset to 1.0.0. Switch at your leisure...
npm WARN deprecated hoek@4.2.1: This version has been deprecated in accordance with the
hapi support policy (hapi.im/support). Please upgrade to the latest version to get the best
features, bug fixes, and security patches. If you are unable to upgrade at this time, paid support is
available for older versions (hapi.im/commercial).
/usr/bin/hurl -> /usr/lib/node_modules/@worldsibu/hurley/dist/command.js

> @ampretia/x509@0.4.0 install /usr/lib/node_modules/@worldsibu/hurley/node_modules/fabric-
shim/node_modules/@ampretia/x509
> node-gyp rebuild

make: Entering directory '/usr/lib/node_modules/@worldsibu/hurley/node_modules/fabric-shim/
node_modules/@ampretia/x509/build'
  CXX(target) Release/obj.target/x509/src/addon.o
  CXX(target) Release/obj.target/x509/src/x509.o
../src/x509.cc:      In      function      'Nan::NAN_METHOD_RETURN_TYPE
verify(Nan::NAN_METHOD_ARGS_TYPE)':
../src/x509.cc:56:65:  warning:  'v8::String::Utf8Value::Utf8Value(v8::Local<v8::Value>)' is
deprecated: Use Isolate version [-Wdeprecated-declarations]
  std::string cert_path = *String::Utf8Value(info[0]->ToString());
                          ^
                          |
                          v
In file included from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:26:0,
  from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/node.h:63,
  from ../include/addon.h:4,
  from ../include/x509.h:5,
  from ../src/x509.cc:1:
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:2892:28: note: declared here
  explicit Utf8Value(Local<v8::Value> obj);
                          ^
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8config.h:324:3: note: in definition of
macro 'V8_DEPRECATED'
  declarator __attribute__((deprecated(message)))
  ~~~~~
../src/x509.cc:57:68:  warning:  'v8::String::Utf8Value::Utf8Value(v8::Local<v8::Value>)' is
deprecated: Use Isolate version [-Wdeprecated-declarations]
  std::string ca_bundlestr = *String::Utf8Value(info[1]->ToString());
                          ^
                          |
                          v
In file included from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:26:0,
  from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/node.h:63,
  from ../include/addon.h:4,
  from ../include/x509.h:5,
  from ../src/x509.cc:1:
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:2892:28: note: declared here
  explicit Utf8Value(Local<v8::Value> obj);
                          ^
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8config.h:324:3: note: in definition of
macro 'V8_DEPRECATED'
  declarator __attribute__((deprecated(message)))
  ~~~~~
../src/x509.cc:      In      function      'Nan::NAN_METHOD_RETURN_TYPE
get_alt_names(Nan::NAN_METHOD_ARGS_TYPE)':

```

```

./src/x509.cc:119:57: warning: 'v8::Local<v8::Object> v8::Value::ToObject() const' is deprecated:
Use maybe version [-Wdeprecated-declarations]
  Local<Object> exports(try_parse(parsed_arg)->ToObject());
                        ^
In file included from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/node.h:63:0,
  from ../include/addon.h:4,
  from ../include/x509.h:5,
  from ./src/x509.cc:1:
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:10046:15: note: declared here
Local<Object> Value::ToObject() const {
  ^~~~~
./src/x509.cc:      In function      'Nan::NAN_METHOD_RETURN_TYPE
get_subject(Nan::NAN_METHOD_ARGS_TYPE)':
./src/x509.cc:132:57: warning: 'v8::Local<v8::Object> v8::Value::ToObject() const' is deprecated:
Use maybe version [-Wdeprecated-declarations]
  Local<Object> exports(try_parse(parsed_arg)->ToObject());
                        ^
In file included from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/node.h:63:0,
  from ../include/addon.h:4,
  from ../include/x509.h:5,
  from ./src/x509.cc:1:
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:10046:15: note: declared here
Local<Object> Value::ToObject() const {
  ^~~~~
./src/x509.cc:      In function      'Nan::NAN_METHOD_RETURN_TYPE
get_issuer(Nan::NAN_METHOD_ARGS_TYPE)':
./src/x509.cc:145:57: warning: 'v8::Local<v8::Object> v8::Value::ToObject() const' is deprecated:
Use maybe version [-Wdeprecated-declarations]
  Local<Object> exports(try_parse(parsed_arg)->ToObject());
                        ^
In file included from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/node.h:63:0,
  from ../include/addon.h:4,
  from ../include/x509.h:5,
  from ./src/x509.cc:1:
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:10046:15: note: declared here
Local<Object> Value::ToObject() const {
  ^~~~~
./src/x509.cc:      In function      'Nan::NAN_METHOD_RETURN_TYPE
parse_cert(Nan::NAN_METHOD_ARGS_TYPE)':
./src/x509.cc:158:57: warning: 'v8::Local<v8::Object> v8::Value::ToObject() const' is deprecated:
Use maybe version [-Wdeprecated-declarations]
  Local<Object> exports(try_parse(parsed_arg)->ToObject());
                        ^
In file included from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/node.h:63:0,
  from ../include/addon.h:4,
  from ../include/x509.h:5,
  from ./src/x509.cc:1:
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:10046:15: note: declared here
Local<Object> Value::ToObject() const {
  ^~~~~
In      file      included      from
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/openssl/asn1.h:16:0,
  from ../include/x509.h:11,
  from ./src/x509.cc:1:
./src/x509.cc: In function 'v8::Local<v8::Value> try_parse(const string&)':
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/openssl/bio.h:481:34: warning: value
computed is not used [-Wunused-value]
# define BIO_set_close(b,c)    (int)BIO_ctrl(b,BIO_CTRL_SET_CLOSE,(c),NULL)
  ^~~~~~
./src/x509.cc:395:5: note: in expansion of macro 'BIO_set_close'
  BIO_set_close(ext_bio, BIO_CLOSE);
  ^~~~~~
./src/x509.cc: In function 'v8::Local<v8::Value> parse_date(ASN1_TIME*)':
./src/x509.cc:459:75: warning: 'v8::Local<v8::Object> v8::Value::ToObject() const' is deprecated:
Use maybe version [-Wdeprecated-declarations]
  Nan::New<String>("Date").ToLocalChecked().ToLocalChecked()->ToObject();
  ^

```



```

In file included from /home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/node.h:63:0,
    from ../include/addon.h:4,
    from ../include/x509.h:5,
    from ../src/x509.cc:1:
/home/manelmdiaz/.cache/node-gyp/10.18.0/include/node/v8.h:10046:15: note: declared here
Local<Object> Value::ToObject() const {
    ^~~~~
SOLINK_MODULE(target) Release/obj.target/x509.node
COPY Release/x509.node
make: Leaving directory '/usr/lib/node_modules/@worldsibu/hurley/node_modules/fabric-shim/
node_modules/@ampretia/x509/build'

>                                     grpc@1.17.0                                install
/usr/lib/node_modules/@worldsibu/hurley/node_modules/fabric-shim/node_modules/grpc
> node-pre-gyp install --fallback-to-build --library=static_library

node-pre-gyp WARN Using request for node-pre-gyp https download
[grpc] Success: "/usr/lib/node_modules/@worldsibu/hurley/node_modules/fabric-shim/
node_modules/grpc/src/node/extension_binary/node-v64-linux-x64-glibc/grpc_node.node" is
installed via remote

> pkcs11js@1.0.19 install /usr/lib/node_modules/@worldsibu/hurley/node_modules/pkcs11js
> node-gyp rebuild

make: Entering directory '/usr/lib/node_modules/@worldsibu/hurley/node_modules/pkcs11js/build'
CXX(target) Release/obj.target/pkcs11/src/main.o
CXX(target) Release/obj.target/pkcs11/src/dl.o
CXX(target) Release/obj.target/pkcs11/src/const.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/error.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/v8_convert.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/template.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/mech.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/param.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/param_aes.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/param_rsa.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/param_ecdh.o
CXX(target) Release/obj.target/pkcs11/src/pkcs11/pkcs11.o
CXX(target) Release/obj.target/pkcs11/src/async.o
CXX(target) Release/obj.target/pkcs11/src/node.o
SOLINK_MODULE(target) Release/obj.target/pkcs11.node
COPY Release/pkcs11.node
make: Leaving directory '/usr/lib/node_modules/@worldsibu/hurley/node_modules/pkcs11js/build'

> grpc@1.21.1 install /usr/lib/node_modules/@worldsibu/hurley/node_modules/grpc
> node-pre-gyp install --fallback-to-build --library=static_library

node-pre-gyp WARN Using request for node-pre-gyp https download
[grpc] Success: "/usr/lib/node_modules/@worldsibu/hurley/node_modules/grpc/src/node/
extension_binary/node-v64-linux-x64-glibc/grpc_node.node" is installed via remote

> core-js@2.6.11 postinstall /usr/lib/node_modules/@worldsibu/hurley/node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript
standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or
Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

> ejs@2.7.4 postinstall /usr/lib/node_modules/@worldsibu/hurley/node_modules/ejs
> node ./postinstall.js

```

```

Thank you for installing EJS: built with the Jake JavaScript build tool (https://jakejs.com/)

> spawn-sync@1.0.15 postinstall
/usr/lib/node_modules/@worldsibu/hurley/node_modules/spawn-sync
> node postinstall

npm WARN notsup Unsupported engine for fabric-shim-crypto@1.4.4: wanted:
{"node":>=8.9.0,"npm":>=5.5.1"} (current: {"node":>=10.18.0,"npm":>=6.13.4"})
npm WARN notsup Not compatible with your version of node/npm: fabric-shim-crypto@1.4.4
npm WARN notsup Unsupported engine for fabric-shim@1.4.4: wanted:
{"node":>=8.9.0,"npm":>=5.5.1"} (current: {"node":>=10.18.0,"npm":>=6.13.4"})
npm WARN notsup Not compatible with your version of node/npm: fabric-shim@1.4.4

+ @worldsibu/hurley@1.1.2
added 1117 packages from 874 contributors in 55.103s

```

6.6 Resultat de comanda 'hurl new'

```

manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector/chaincode-participant$ hurl new
[hurley] - Hurley collects anonymous usage statistics to improve the tool over time
[hurley] - About to create binaries
[hurley] - Created and saved binaries
[hurley] - About to run binaries
Installing Hyperledger Fabric binaries
Binaries exist already
Checking IMAGES
==> Pulling fabric Images
==> FABRIC IMAGE: peer

1.4.0: Pulling from hyperledger/fabric-peer
Digest: sha256:9707c97f787de1d4d6dd60994d6b8ea2e5cc28b0f42e6849df3fb41c64b41372
Status: Image is up to date for hyperledger/fabric-peer:1.4.0
docker.io/hyperledger/fabric-peer:1.4.0
==> FABRIC IMAGE: orderer

1.4.0: Pulling from hyperledger/fabric-orderer
Digest: sha256:644265186b4887c7d9dcb91895124cceed3c0125c2c4f9eadc421dc9555d7495
Status: Image is up to date for hyperledger/fabric-orderer:1.4.0
docker.io/hyperledger/fabric-orderer:1.4.0
==> FABRIC IMAGE: tools

1.4.0: Pulling from hyperledger/fabric-tools
Digest: sha256:aee256916d0cb938d8023fa32ed2745991d32cfe79018e360f1720707ebfbd5
Status: Image is up to date for hyperledger/fabric-tools:1.4.0
docker.io/hyperledger/fabric-tools:1.4.0
==> Pulling fabric ca Image
==> FABRIC CA IMAGE

1.4.0: Pulling from hyperledger/fabric-ca
Digest: sha256:c1dce534d9e9202697e0aaad7c5521d958700fda0b05127dafb9333c22e15f74
Status: Image is up to date for hyperledger/fabric-ca:1.4.0
docker.io/hyperledger/fabric-ca:1.4.0
==> Pulling thirdparty docker images
==> THIRDPARTY DOCKER IMAGE: couchdb

0.4.14: Pulling from hyperledger/fabric-couchdb
Digest: sha256:021c7e4a5047432d892fbd7d5220d3049ff4e8b436fd481bb08e41871f1aac7
Status: Image is up to date for hyperledger/fabric-couchdb:0.4.14
docker.io/hyperledger/fabric-couchdb:0.4.14

==> List out hyperledger docker images
hyperledger/fabric-tools 1.4.0 0a44f4261a55 11 months ago 1.56GB
hyperledger/fabric-tools latest 0a44f4261a55 11 months ago 1.56GB
hyperledger/fabric-orderer 1.4.0 54f372205580 11 months ago 150MB
hyperledger/fabric-orderer latest 54f372205580 11 months ago 150MB

```

hyperledger/fabric-peer	1.4.0	304fac59b501	11 months ago	157MB
hyperledger/fabric-peer	latest	304fac59b501	11 months ago	157MB
hyperledger/fabric-ca	1.4.0	1a804ab74f58	11 months ago	244MB
hyperledger/fabric-ca	latest	1a804ab74f58	11 months ago	244MB
hyperledger/fabric-couchdb	0.4.14	f14f97292b4c	14 months ago	1.5GB
hyperledger/fabric-couchdb	latest	f14f97292b4c	14 months ago	1.5GB

```

[hurley] - Ran binaries
[hurley] - About to create configtxyaml
[hurley] - Created and saved configtxyaml
[hurley] - About to create cryptoconfigyaml
[hurley] - Created and saved cryptoconfigyaml
[hurley] - About to create cryptoconfigsh
[hurley] - Created and saved cryptoconfigsh
[hurley] - Running cryptoconfigsh
org1.hurley.lab
org2.hurley.lab
2019-12-29 22:49:29.064 CET [common.tools.configtxgen] main -> WARN 001 Omitting the
channel ID for configtxgen for output operations is deprecated. Explicitly passing the channel ID
will be required in the future, defaulting to 'testchainid'.
2019-12-29 22:49:29.064 CET [common.tools.configtxgen] main -> INFO 002 Loading
configuration
2019-12-29 22:49:29.067 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-29 22:49:29.067 CET [common.tools.configtxgen.localconfig] Load -> INFO 004 Loaded
configuration: /home/manelmdiaz/hyperledger-fabric-network/configtx.yaml
2019-12-29 22:49:29.070 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 005 orderer type: solo
2019-12-29 22:49:29.070 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
006 Loaded configuration: /home/manelmdiaz/hyperledger-fabric-network/configtx.yaml
2019-12-29 22:49:29.070 CET [common.tools.configtxgen.encoder] NewChannelGroup -> WARN
007 Default policy emission is deprecated, please include policy specifications for the channel
group in configtx.yaml
2019-12-29 22:49:29.070 CET [common.tools.configtxgen.encoder] NewOrdererGroup -> WARN
008 Default policy emission is deprecated, please include policy specifications for the orderer
group in configtx.yaml
2019-12-29 22:49:29.070 CET [common.tools.configtxgen.encoder] NewOrdererOrgGroup ->
WARN 009 Default policy emission is deprecated, please include policy specifications for the
orderer org group OrdererOrg in configtx.yaml
2019-12-29 22:49:29.071 CET [common.tools.configtxgen.encoder] NewOrdererOrgGroup ->
WARN 00a Default policy emission is deprecated, please include policy specifications for the
orderer org group org1MSP in configtx.yaml
2019-12-29 22:49:29.071 CET [common.tools.configtxgen.encoder] NewOrdererOrgGroup ->
WARN 00b Default policy emission is deprecated, please include policy specifications for the
orderer org group org2MSP in configtx.yaml
2019-12-29 22:49:29.071 CET [common.tools.configtxgen] doOutputBlock -> INFO 00c
Generating genesis block
2019-12-29 22:49:29.071 CET [common.tools.configtxgen] doOutputBlock -> INFO 00d Writing
genesis block
2019-12-29 22:49:29.102 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-29 22:49:29.106 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/hyperledger-fabric-network/configtx.yaml
2019-12-29 22:49:29.109 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-29 22:49:29.109 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/hyperledger-fabric-network/configtx.yaml
2019-12-29 22:49:29.109 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-29 22:49:29.109 CET [common.tools.configtxgen.encoder] NewApplicationGroup ->
WARN 006 Default policy emission is deprecated, please include policy specifications for the
application group in configtx.yaml
2019-12-29 22:49:29.109 CET [common.tools.configtxgen.encoder] NewApplicationOrgGroup ->
WARN 007 Default policy emission is deprecated, please include policy specifications for the
application org group org1MSP in configtx.yaml
2019-12-29 22:49:29.109 CET [common.tools.configtxgen.encoder] NewApplicationOrgGroup ->
WARN 008 Default policy emission is deprecated, please include policy specifications for the
application org group org2MSP in configtx.yaml

```

```

2019-12-29 22:49:29.110 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
009 Writing new channel tx
2019-12-29 22:49:29.143 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-29 22:49:29.146 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/hyperledger-fabric-network/configtx.yaml
2019-12-29 22:49:29.148 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-29 22:49:29.148 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/hyperledger-fabric-network/configtx.yaml
2019-12-29 22:49:29.148 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
005 Generating anchor peer update
2019-12-29 22:49:29.149 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
006 Writing anchor peer update
2019-12-29 22:49:29.184 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-29 22:49:29.188 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/hyperledger-fabric-network/configtx.yaml
2019-12-29 22:49:29.192 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-29 22:49:29.192 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/hyperledger-fabric-network/configtx.yaml
2019-12-29 22:49:29.192 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
005 Generating anchor peer update
2019-12-29 22:49:29.192 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
006 Writing anchor peer update
[hurley] - Ran cryptoconfigsh
[hurley] - Building compose
[hurley] - Builided compose
[hurley] - Saving compose
[hurley] - Saved compose
[hurley] - Creating network profiles
[hurley] - Cleaning .hfc-org1
[hurley] - Cleaning .hfc-org2
[hurley] - Creating for org1
[hurley] - Creating for org2
[hurley] - Creating for org1 inside Docker
[hurley] - Creating for org2 inside Docker
[hurley] - Created network profiles
[hurley] - Creating network restart script
[hurley] - Saved network restart script
[hurley] - Running network restart script
Creating network "net_hurley_dev_net" with the default driver
Creating volume "net_shared" with default driver
Creating couchdb.peer0.org1.hurley.lab ...
Creating orderer.hurley.lab ...
Creating ca.org2.hurley.lab ...
Creating couchdb.peer0.org2.hurley.lab ...
Creating ca.org1.hurley.lab ...
Creating couchdb.peer0.org1.hurley.lab
Creating couchdb.peer0.org2.hurley.lab
Creating orderer.hurley.lab
Creating ca.org2.hurley.lab
Creating couchdb.peer0.org2.hurley.lab ... done
Creating peer0.org2.hurley.lab ...
Creating couchdb.peer0.org1.hurley.lab ... done
Creating peer0.org1.hurley.lab ...
Creating peer0.org2.hurley.lab ... done
Creating ch1 channel block in peer peer0.org1.hurley.lab
2019-12-29 21:49:35.496 UTC [main] InitCmd -> WARN 001 CORE_LOGGING_LEVEL is no
longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:35.496 UTC [viperutil] getKeysRecursively -> DEBU 002 Found
map[string]interface{} value for peer.BCCSP
2019-12-29 21:49:35.497 UTC [viperutil] unmarshalJSON -> DEBU 003 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:35.497 UTC [viperutil] getKeysRecursively -> DEBU 004 Found real value for
peer.BCCSP.Default setting to string SW

```

```

2019-12-29 21:49:35.497 UTC [viperutil] getKeysRecursively -> DEBU 005 Found
map[string]interface{} value for peer.BCCSP.SW
2019-12-29 21:49:35.497 UTC [viperutil] unmarshalJSON -> DEBU 006 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:35.497 UTC [viperutil] getKeysRecursively -> DEBU 007 Found real value for
peer.BCCSP.SW.Hash setting to string SHA2
2019-12-29 21:49:35.497 UTC [viperutil] unmarshalJSON -> DEBU 008 Unmarshal JSON: value
is not a string: 256
2019-12-29 21:49:35.497 UTC [viperutil] getKeysRecursively -> DEBU 009 Found real value for
peer.BCCSP.SW.Security setting to int 256
2019-12-29 21:49:35.497 UTC [viperutil] getKeysRecursively -> DEBU 00a Found
map[string]interface{} value for peer.BCCSP.SW.FileKeyStore
2019-12-29 21:49:35.497 UTC [viperutil] unmarshalJSON -> DEBU 00b Unmarshal JSON: value
cannot be unmarshalled: unexpected end of JSON input
2019-12-29 21:49:35.497 UTC [viperutil] getKeysRecursively -> DEBU 00c Found real value for
peer.BCCSP.SW.FileKeyStore.KeyStore setting to string
2019-12-29 21:49:35.497 UTC [viperutil] getKeysRecursively -> DEBU 00d Found
map[string]interface{} value for peer.BCCSP.PKCS11
2019-12-29 21:49:35.498 UTC [viperutil] unmarshalJSON -> DEBU 00e Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:35.498 UTC [viperutil] getKeysRecursively -> DEBU 00f Found real value for
peer.BCCSP.PKCS11.Library setting to <nil> <nil>
2019-12-29 21:49:35.498 UTC [viperutil] unmarshalJSON -> DEBU 010 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:35.498 UTC [viperutil] getKeysRecursively -> DEBU 011 Found real value for
peer.BCCSP.PKCS11.Label setting to <nil> <nil>
2019-12-29 21:49:35.498 UTC [viperutil] unmarshalJSON -> DEBU 012 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:35.498 UTC [viperutil] getKeysRecursively -> DEBU 013 Found real value for
peer.BCCSP.PKCS11.Pin setting to <nil> <nil>
2019-12-29 21:49:35.499 UTC [viperutil] unmarshalJSON -> DEBU 014 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:35.499 UTC [viperutil] getKeysRecursively -> DEBU 015 Found real value for
peer.BCCSP.PKCS11.Hash setting to <nil> <nil>
2019-12-29 21:49:35.499 UTC [viperutil] unmarshalJSON -> DEBU 016 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:35.499 UTC [viperutil] getKeysRecursively -> DEBU 017 Found real value for
peer.BCCSP.PKCS11.Security setting to <nil> <nil>
2019-12-29 21:49:35.499 UTC [viperutil] getKeysRecursively -> DEBU 018 Found
map[string]interface{} value for peer.BCCSP.PKCS11.FileKeyStore
2019-12-29 21:49:35.499 UTC [viperutil] unmarshalJSON -> DEBU 019 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:35.500 UTC [viperutil] getKeysRecursively -> DEBU 01a Found real value for
peer.BCCSP.PKCS11.FileKeyStore.KeyStore setting to <nil> <nil>
2019-12-29 21:49:35.500 UTC [viperutil] EnhancedExactUnmarshalKey -> DEBU 01b
map[peer.BCCSP:map[Default:SW SW:map[Hash:SHA2 Security:256
FileKeyStore:map[KeyStore:]] PKCS11:map[Pin:<nil> Hash:<nil> Security:<nil>
FileKeyStore:map[KeyStore:<nil>] Library:<nil> Label:<nil>]]]
2019-12-29 21:49:35.500 UTC [bccsp_sw] openKeyStore -> DEBU 01c KeyStore opened at [/etc/
hyperledger/msp/users/Admin@org1.hurley.lab/msp/keystore]...done
2019-12-29 21:49:35.502 UTC [bccsp] initBCCSP -> DEBU 01d Initialize BCCSP [SW]
2019-12-29 21:49:35.502 UTC [msp] getPemMaterialFromDir -> DEBU 01e Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/signcerts
2019-12-29 21:49:35.502 UTC [msp] getPemMaterialFromDir -> DEBU 01f Inspecting file
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/signcerts/Admin@org1.hurley.lab-
cert.pem
2019-12-29 21:49:35.504 UTC [msp] getPemMaterialFromDir -> DEBU 020 Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/cacerts
2019-12-29 21:49:35.504 UTC [msp] getPemMaterialFromDir -> DEBU 021 Inspecting file
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/cacerts/ca.org1.hurley.lab-cert.pem
2019-12-29 21:49:35.504 UTC [msp] getPemMaterialFromDir -> DEBU 022 Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/admincerts
2019-12-29 21:49:35.504 UTC [msp] getPemMaterialFromDir -> DEBU 023 Inspecting file
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/admincerts/Admin@org1.hurley.lab-
cert.pem
2019-12-29 21:49:35.504 UTC [msp] getPemMaterialFromDir -> DEBU 024 Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts

```

2019-12-29 21:49:35.504 UTC [msp] getMspConfig -> DEBU 025 Intermediate certs folder not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts]. Skipping. [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts: no such file or directory]

2019-12-29 21:49:35.504 UTC [msp] getPemMaterialFromDir -> DEBU 026 Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlscacerts

2019-12-29 21:49:35.504 UTC [msp] getPemMaterialFromDir -> DEBU 027 Inspecting file /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlscacerts/tlscacerts.org1.hurley.lab-cert.pem

2019-12-29 21:49:35.504 UTC [msp] getPemMaterialFromDir -> DEBU 028 Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts

2019-12-29 21:49:35.505 UTC [msp] getMspConfig -> DEBU 029 TLS intermediate certs folder not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts]. Skipping. [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts: no such file or directory]

2019-12-29 21:49:35.505 UTC [msp] getPemMaterialFromDir -> DEBU 02a Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls

2019-12-29 21:49:35.505 UTC [msp] getMspConfig -> DEBU 02b crls folder not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls]. Skipping. [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls: no such file or directory]

2019-12-29 21:49:35.505 UTC [msp] getMspConfig -> DEBU 02c MSP configuration file not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/config.yaml]: [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/config.yaml: no such file or directory]

2019-12-29 21:49:35.505 UTC [msp] newBccspMsp -> DEBU 02d Creating BCCSP-based MSP instance

2019-12-29 21:49:35.505 UTC [msp] New -> DEBU 02e Creating Cache-MSP instance

2019-12-29 21:49:35.505 UTC [msp] loadLocaMSP -> DEBU 02f Created new local MSP

2019-12-29 21:49:35.505 UTC [msp] Setup -> DEBU 030 Setting up MSP instance org1MSP

2019-12-29 21:49:35.505 UTC [msp.identity] newIdentity -> DEBU 031 Creating identity instance for cert -----BEGIN CERTIFICATE-----
MIICDCCAFogAwIBAgIQQ7TyVkcpJS+f2MkExfvaozAKBggqhkJOPQQDAjBxMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcms5pYTEWMBQGA1UEBxMNU2FuIEZyYW5jaXNjbzEYMBYGA1UEChMPb3JnMS5odXJsZXkubGFiMRswGQYDVQQDEXJjYS5vcmcxLmhhcmxleS5sYWVwHhcNMjE0MDAwMjE0NDAwWjBxMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcms5pYTEWMBQGA1UEBxMNU2FuIEZyYW5jaXNjbzEYMBYGA1UEChMPb3JnMS5odXJsZXkubGFiMRswGQYDVQQDEXJjYS5vcmcxLmhhcmxleS5sYWVwWTATBgcqhkjOPQIBBggqhkJOPQMBBwNCAASf74s3k5eu1gRBP1qicgmfZE6rjzYxJvW/Dfw7xmlfutAnPH7LJ/Ql1vpW7RH2G1MrUsCj4mAKL7n2RbsYD7ylo20wazAOBgNVHQ8BAf8EBAMCAAYwHQYDVR0IBBYwFAYIKwYBQUHAWIGCCsGAQUFBwMBMA8GA1UdEwEB/wQFMAMBAf8wKQYDVR0BQCIEIM8+w7YCAGZe7e5dHQVfUeVdLQj9Pv9KXI3yBcud8H+MAoGCCqGSM49BAMCA0cAMEQCIB5IsqHu8ljnipluXUS1m6bP95aRMqukgf2jYVI3SYMAiB6v5zNdtc9/pNnIB+M2H+uIL5gTizjEfx8bWCNMxc7UQ==
-----END CERTIFICATE-----

2019-12-29 21:49:35.506 UTC [msp.identity] newIdentity -> DEBU 032 Creating identity instance for cert -----BEGIN CERTIFICATE-----
MIICDCCAc6gAwIBAgIRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZj0EAWlwTELMAKGA1UEBhMVCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBGNVBAcTDVNhbiBGcmFuY2lyZ28xGDAWBgNVBAoTD29yZzEuaHVVybGV5LmhhcmxleS5sYWVwHhcNMjE0MDAwMjE0NDAwWjBxMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcms5pYTEWMBQGA1UEBxMNU2FuIEZyYW5jaXNjbzEYMBYGA1UEChMPb3JnMS5odXJsZXkubGFiMRswGQYDVQQDEXJjYS5vcmcxLmhhcmxleS5sYWVwWTATBgcqhkjOPQIBBggqhkJOPQMBBwNCAASf74s3k5eu1gRBP1qicgmfZE6rjzYxJvW/Dfw7xmlfutAnPH7LJ/Ql1vpW7RH2G1MrUsCj4mAKL7n2RbsYD7ylo20wazAOBgNVHQ8BAf8EBAMCAAYwHQYDVR0IBBYwFAYIKwYBQUHAWIGCCsGAQUFBwMBMA8GA1UdEwEB/wQFMAMBAf8wKQYDVR0BQCIEIM8+w7YCAGZe7e5dHQVfUeVdLQj9Pv9KXI3yBcud8H+MAoGCCqGSM49BAMCA0cAMEQCIB5IsqHu8ljnipluXUS1m6bP95aRMqukgf2jYVI3SYMAiB6v5zNdtc9/pNnIB+M2H+uIL5gTizjEfx8bWCNMxc7UQ==
-----END CERTIFICATE-----

2019-12-29 21:49:35.515 UTC [msp.identity] newIdentity -> DEBU 033 Creating identity instance for cert -----BEGIN CERTIFICATE-----
MIICDCCAc6gAwIBAgIRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZj0EAWlwTELMAKGA1UEBhMVCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBGNVBAcTDVNhbiBGcmFuY2lyZ28xGDAWBgNVBAoTD29yZzEuaHVVybGV5LmhhcmxleS5sYWVwHhcNMjE0MDAwMjE0NDAwWjBxMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcms5pYTEWMBQGA1UEBxMNU2FuIEZyYW5jaXNjbzEYMBYGA1UEChMPb3JnMS5odXJsZXkubGFiMRswGQYDVQQDEXJjYS5vcmcxLmhhcmxleS5sYWVwWTATBgcqhkjOPQIBBggqhkJOPQMBBwNCAASf74s3k5eu1gRBP1qicgmfZE6rjzYxJvW/Dfw7xmlfutAnPH7LJ/Ql1vpW7RH2G1MrUsCj4mAKL7n2RbsYD7ylo20wazAOBgNVHQ8BAf8EBAMCAAYwHQYDVR0IBBYwFAYIKwYBQUHAWIGCCsGAQUFBwMBMA8GA1UdEwEB/wQFMAMBAf8wKQYDVR0BQCIEIM8+w7YCAGZe7e5dHQVfUeVdLQj9Pv9KXI3yBcud8H+MAoGCCqGSM49BAMCA0cAMEQCIB5IsqHu8ljnipluXUS1m6bP95aRMqukgf2jYVI3SYMAiB6v5zNdtc9/pNnIB+M2H+uIL5gTizjEfx8bWCNMxc7UQ==
-----END CERTIFICATE-----

```
MS5odXJsZXkubGFIMFkwEwYHKOZlZj0CAQYIKoZlZj0DAQcDQgAEittt4NVKG4er
98vMAyDj3ROx2RCqsF2AugnQY7lft1exXoRctfSgJSRssqpYTvWJ5ijyrtgJR7qf
ryKB91EMe6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAgzz7DtgICBI7t7l0dBUW4S90s6OP0+/0peXflFy53wf4wCgYIKoZlZj0E
AwIDSAAwRQlhAN84ywXWxmAsdFULxMQqKOq78V0g1e1SyO6/U+SGZms8AiAbnSP/
/qep5onXRpRIWnmGOe4jSiXFehP3uW1I2hXjg==
-----END CERTIFICATE-----
2019-12-29 21:49:35.515 UTC [bccsp_sw] loadPrivateKey -> DEBU 034 Loading private key
[003d407522234792c36bacdc1fc0d312cc28fc07f8e3733590b0ca7861044502] at
[/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/keystore/003d407522234792c36bacdc
1fc0d312cc28fc07f8e3733590b0ca7861044502_sk]...
2019-12-29 21:49:35.516 UTC [msp.identity] newIdentity -> DEBU 035 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwIBAgIRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZj0EAWlwTEL
MAKGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWExFjAUBGNVBACjDVNhbG
cmFuY2lyZ28xGDAWBgNVBAoTD29yZzEuaHVybGV5LmXhYjE5b3JnMS5odXJsZXkubGFIMB4XDTE5MTIyOTIxNDQwMFoXDTI5MTIyNjlxNDQwMFow
azELMAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWExFjAUBGNVBACjDVNhb
biBGcmFuY2lyZ28xGDAWBgNVBAAsTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
MS5odXJsZXkubGFIMFkwEwYHKOZlZj0CAQYIKoZlZj0DAQcDQgAEittt4NVKG4er
98vMAyDj3ROx2RCqsF2AugnQY7lft1exXoRctfSgJSRssqpYTvWJ5ijyrtgJR7qf
ryKB91EMe6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAgzz7DtgICBI7t7l0dBUW4S90s6OP0+/0peXflFy53wf4wCgYIKoZlZj0E
AwIDSAAwRQlhAN84ywXWxmAsdFULxMQqKOq78V0g1e1SyO6/U+SGZms8AiAbnSP/
/qep5onXRpRIWnmGOe4jSiXFehP3uW1I2hXjg==
-----END CERTIFICATE-----
2019-12-29 21:49:35.516 UTC [msp] setupSigningIdentity -> DEBU 036 Signing identity expires
at 2029-12-26 21:44:00 +0000 UTC
2019-12-29 21:49:35.516 UTC [msp] Validate -> DEBU 037 MSP org1MSP validating identity
2019-12-29 21:49:35.517 UTC [main] SetOrdererEnv -> WARN 038 CORE_LOGGING_LEVEL is
no longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:35.517 UTC [msp] GetDefaultSigningIdentity -> DEBU 039 Obtaining default
signing identity
2019-12-29 21:49:35.517 UTC [grpc] DialContext -> DEBU 03a parsed scheme: ""
2019-12-29 21:49:35.517 UTC [grpc] DialContext -> DEBU 03b scheme "" not registered, fallback
to default scheme
2019-12-29 21:49:35.517 UTC [grpc] watcher -> DEBU 03c ccResolverWrapper: sending new
addresses to cc: [{orderer.hurley.lab:7050 0 <nil>}]
2019-12-29 21:49:35.517 UTC [grpc] switchBalancer -> DEBU 03d ClientConn switching balancer
to "pick_first"
2019-12-29 21:49:35.517 UTC [grpc] HandleSubConnStateChange -> DEBU 03e
pickfirstBalancer: HandleSubConnStateChange: 0xc000151570, CONNECTING
2019-12-29 21:49:35.521 UTC [grpc] HandleSubConnStateChange -> DEBU 03f
pickfirstBalancer: HandleSubConnStateChange: 0xc000151570, READY
2019-12-29 21:49:35.521 UTC [channelCmd] InitCmdFactory -> INFO 040 Endorser and orderer
connections initialized
2019-12-29 21:49:35.522 UTC [msp] GetDefaultSigningIdentity -> DEBU 041 Obtaining default
signing identity
2019-12-29 21:49:35.522 UTC [msp] GetDefaultSigningIdentity -> DEBU 042 Obtaining default
signing identity
2019-12-29 21:49:35.522 UTC [msp.identity] Sign -> DEBU 043 Sign: plaintext:
0AB6060A076F7267314D535012AA062D...53616D706C65436F6E736F727469756D
2019-12-29 21:49:35.522 UTC [msp.identity] Sign -> DEBU 044 Sign: digest:
7E1C1488C2AC81FC7D146EDCBC8C369D5406493E01F4E81B32B38A638792C873
2019-12-29 21:49:35.522 UTC [msp] GetDefaultSigningIdentity -> DEBU 045 Obtaining default
signing identity
2019-12-29 21:49:35.522 UTC [msp] GetDefaultSigningIdentity -> DEBU 046 Obtaining default
signing identity
2019-12-29 21:49:35.522 UTC [msp.identity] Sign -> DEBU 047 Sign: plaintext:
0AE7060A0F08021A0608EFBEA4F00522...33DDC6A4C61A93614B60946AE341FE1B
2019-12-29 21:49:35.522 UTC [msp.identity] Sign -> DEBU 048 Sign: digest:
0073F37A6EE0CD2075963001F4A0FEE440831EE0176C7C4C0220A287A3466CF8
2019-12-29 21:49:35.522 UTC [grpc] DialContext -> DEBU 049 parsed scheme: ""
2019-12-29 21:49:35.522 UTC [grpc] DialContext -> DEBU 04a scheme "" not registered, fallback
to default scheme
2019-12-29 21:49:35.522 UTC [grpc] watcher -> DEBU 04b ccResolverWrapper: sending new
addresses to cc: [{orderer.hurley.lab:7050 0 <nil>}]
```

```

2019-12-29 21:49:35.523 UTC [grpc] switchBalancer -> DEBU 04c ClientConn switching balancer
to "pick_first"
2019-12-29 21:49:35.523 UTC [grpc] HandleSubConnStateChange -> DEBU 04d
pickfirstBalancer: HandleSubConnStateChange: 0xc0004d8f20, CONNECTING
2019-12-29 21:49:35.524 UTC [grpc] HandleSubConnStateChange -> DEBU 04e
pickfirstBalancer: HandleSubConnStateChange: 0xc0004d8f20, READY
2019-12-29 21:49:35.551 UTC [msp] GetDefaultSigningIdentity -> DEBU 04f Obtaining default
signing identity
2019-12-29 21:49:35.551 UTC [msp] GetDefaultSigningIdentity -> DEBU 050 Obtaining default
signing identity
2019-12-29 21:49:35.551 UTC [msp.identity] Sign -> DEBU 051 Sign: plaintext:
0AE7060A0F08051A0608EFBEA4F00522...EE30306221CB12080A021A0012021A00
2019-12-29 21:49:35.551 UTC [msp.identity] Sign -> DEBU 052 Sign: digest:
7D44845ED5D6DD039DF2BAD2BA9CA5265D6F413686A257B85F72C5B777F1FBC4
2019-12-29 21:49:35.573 UTC [cli.common] readBlock -> INFO 053 Received block: 0
Joining ch1 channel on peer peer0.org1.hurley.lab
++ docker exec peer0.org1.hurley.lab peer channel join -b /shared/ch1.block
2019-12-29 21:49:46.317 UTC [main] InitCmd -> WARN 001 CORE_LOGGING_LEVEL is no
longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:46.317 UTC [viperutil] getKeysRecursively -> DEBU 002 Found
map[string]interface{} value for peer.BCCSP
2019-12-29 21:49:46.318 UTC [viperutil] unmarshalJSON -> DEBU 003 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:46.318 UTC [viperutil] getKeysRecursively -> DEBU 004 Found real value for
peer.BCCSP.Default setting to string SW
2019-12-29 21:49:46.318 UTC [viperutil] getKeysRecursively -> DEBU 005 Found
map[string]interface{} value for peer.BCCSP.SW
2019-12-29 21:49:46.318 UTC [viperutil] unmarshalJSON -> DEBU 006 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:46.318 UTC [viperutil] getKeysRecursively -> DEBU 007 Found real value for
peer.BCCSP.SW.Hash setting to string SHA2
2019-12-29 21:49:46.318 UTC [viperutil] unmarshalJSON -> DEBU 008 Unmarshal JSON: value
is not a string: 256
2019-12-29 21:49:46.318 UTC [viperutil] getKeysRecursively -> DEBU 009 Found real value for
peer.BCCSP.SW.Security setting to int 256
2019-12-29 21:49:46.318 UTC [viperutil] getKeysRecursively -> DEBU 00a Found
map[string]interface{} value for peer.BCCSP.SW.FileKeyStore
2019-12-29 21:49:46.318 UTC [viperutil] unmarshalJSON -> DEBU 00b Unmarshal JSON: value
cannot be unmarshalled: unexpected end of JSON input
2019-12-29 21:49:46.318 UTC [viperutil] getKeysRecursively -> DEBU 00c Found real value for
peer.BCCSP.SW.FileKeyStore.KeyStore setting to string
2019-12-29 21:49:46.319 UTC [viperutil] getKeysRecursively -> DEBU 00d Found
map[string]interface{} value for peer.BCCSP.PKCS11
2019-12-29 21:49:46.319 UTC [viperutil] unmarshalJSON -> DEBU 00e Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.319 UTC [viperutil] getKeysRecursively -> DEBU 00f Found real value for
peer.BCCSP.PKCS11.Library setting to <nil> <nil>
2019-12-29 21:49:46.319 UTC [viperutil] unmarshalJSON -> DEBU 010 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.319 UTC [viperutil] getKeysRecursively -> DEBU 011 Found real value for
peer.BCCSP.PKCS11.Label setting to <nil> <nil>
2019-12-29 21:49:46.320 UTC [viperutil] unmarshalJSON -> DEBU 012 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.320 UTC [viperutil] getKeysRecursively -> DEBU 013 Found real value for
peer.BCCSP.PKCS11.Pin setting to <nil> <nil>
2019-12-29 21:49:46.320 UTC [viperutil] unmarshalJSON -> DEBU 014 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.320 UTC [viperutil] getKeysRecursively -> DEBU 015 Found real value for
peer.BCCSP.PKCS11.Hash setting to <nil> <nil>
2019-12-29 21:49:46.320 UTC [viperutil] unmarshalJSON -> DEBU 016 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.321 UTC [viperutil] getKeysRecursively -> DEBU 017 Found real value for
peer.BCCSP.PKCS11.Security setting to <nil> <nil>
2019-12-29 21:49:46.321 UTC [viperutil] getKeysRecursively -> DEBU 018 Found
map[string]interface{} value for peer.BCCSP.PKCS11.FileKeyStore
2019-12-29 21:49:46.321 UTC [viperutil] unmarshalJSON -> DEBU 019 Unmarshal JSON: value
is not a string: <nil>

```



```
2019-12-29 21:49:46.321 UTC [viperutil] getKeysRecursively -> DEBU 01a Found real value for peer.BCCSP.PKCS11.FileKeyStore.KeyStore setting to <nil> <nil>
2019-12-29 21:49:46.321 UTC [viperutil] EnhancedExactUnmarshalKey -> DEBU 01b map[peer.BCCSP:map[SW:map[Security:256 FileKeyStore:map[KeyStore:] Hash:SHA2] PKCS11:map[Label:<nil> Pin:<nil> Hash:<nil> Security:<nil> FileKeyStore:map[KeyStore:<nil>] Library:<nil>] Default:SW]]
2019-12-29 21:49:46.322 UTC [bccsp_sw] openKeyStore -> DEBU 01c KeyStore opened at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/keystore]...done
2019-12-29 21:49:46.322 UTC [bccsp] initBCCSP -> DEBU 01d Initialize BCCSP [SW]
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 01e Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/signcerts
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 01f Inspecting file /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/signcerts/Admin@org1.hurley.lab-cert.pem
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 020 Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/cacerts
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 021 Inspecting file /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/cacerts/ca.org1.hurley.lab-cert.pem
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 022 Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/admincerts
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 023 Inspecting file /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/admincerts/Admin@org1.hurley.lab-cert.pem
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 024 Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts
2019-12-29 21:49:46.322 UTC [msp] getMspConfig -> DEBU 025 Intermediate certs folder not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts]. Skipping. [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts: no such file or directory]
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 026 Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsacerts
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 027 Inspecting file /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsacerts/tlsca.org1.hurley.lab-cert.pem
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 028 Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts
2019-12-29 21:49:46.322 UTC [msp] getMspConfig -> DEBU 029 TLS intermediate certs folder not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts]. Skipping. [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts: no such file or directory]
2019-12-29 21:49:46.322 UTC [msp] getPemMaterialFromDir -> DEBU 02a Reading directory /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls
2019-12-29 21:49:46.322 UTC [msp] getMspConfig -> DEBU 02b crls folder not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls]. Skipping. [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls: no such file or directory]
2019-12-29 21:49:46.322 UTC [msp] getMspConfig -> DEBU 02c MSP configuration file not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/config.yaml]: [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/config.yaml: no such file or directory]
2019-12-29 21:49:46.323 UTC [msp] newBccspMsp -> DEBU 02d Creating BCCSP-based MSP instance
2019-12-29 21:49:46.323 UTC [msp] New -> DEBU 02e Creating Cache-MSP instance
2019-12-29 21:49:46.323 UTC [msp] loadLocaMSP -> DEBU 02f Created new local MSP
2019-12-29 21:49:46.323 UTC [msp] Setup -> DEBU 030 Setting up MSP instance org1MSP
2019-12-29 21:49:46.323 UTC [msp.identity] newIdentity -> DEBU 031 Creating identity instance for cert -----BEGIN CERTIFICATE-----
MIICDCCAfOgAwIBAgIQ7TtyVkcPjS+f2MkExfvaozAKBggqhkJOPQQDAjBxMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTEwMBQGA1UEBxMNU2FulEZy
YW5jaXNjbzEYMBYGA1UEChMPb3JnMS5odXJsZXkubGFjMRswGQYDVQQDEjJjY55v
cmcxLmh1cmxleS5sYWlwHhcNMTkxMjI1MjE0NDAwWhcNMjI1MjI1MjE0NDAwWjBx
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTEwMBQGA1UEBxMNU2F
u
IEZyYW5jaXNjbzEYMBYGA1UEChMPb3JnMS5odXJsZXkubGFjMRswGQYDVQQDEjJj
YS5vcmxLmh1cmxleS5sYWlwWTATBgqhkJOPQIBBgqhkJOPQMwBwNCAASf74s3
k5eu1gRBP1qicgmfZE6rjzYxJvW/Dfw7xmlfutAnPH7LJ/Ql1vpW7RH2G1MrUsCj
4mAKL7n2RbsYD7ylo20wazAOBgNVHQ8BAf8EBAMCAAYwHQYDVR0IBBYwFAYIKwYB
BQUHAWIGCCsGAQUFBwMBMA8GA1UdEwEB/wQFMAMBAf8wKQYDVROBCEIM8+w7YC
AgZe7e5dHQVfUeVdLQj9Pv9KXl3yBcud8H+MAoGCCqGSM49BAMCA0cAMEQCIB5I
sqHu8ljnipLuXUS1m6bP95aRMqukgef2jYVI3SYMAiB6v5zNdtc9/pNnIB+M2H+u
```

```
IL5gTizjEfX8bWCNMxc7UQ==
-----END CERTIFICATE-----
2019-12-29 21:49:46.323 UTC [msp.identity] newIdentity -> DEBU 032 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwIBAgIRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZjOEAwlwcTEL
MAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBgNVBACQTDVNhbiBG
cmFuY2lzY28xGDAWBgNVBAoTD29yZzEuaHVybGV5LmXhYjEubmBkGA1UEAxMSY2Eu
b3JnMS5odXJsZXkubGFuMjIzY28xGDAWBgNVBAAsTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
azELMAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBgNVBACQTDVNh
biBGcmFuY2lzY28xGDAWBgNVBAAsTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
MS5odXJsZXkubGFuMjIzY28xGDAWBgNVBAQDAQgAEittt4NVK64er
98vMAydJ3ROx2RCqsF2AugnQY7lft1exXoRctfSgJSRssqpYTvWJ5ijyrtgjR7qf
ryKB91EMe6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAgzz7DtgICBI7t7I0dBUW4S90s6OP0+/0peXfIFy53wf4wCgYIKoZlZjOE
AwIDSAAwRQlhAN84ywXWxmAsdFULxMQqKOq78V0g1e1SyO6/U+SGZms8AiAbnSP/
/qep5onXRpRIWnmGOe4jSiXFehP3uW1I2hXjg==
-----END CERTIFICATE-----
2019-12-29 21:49:46.331 UTC [msp.identity] newIdentity -> DEBU 033 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwIBAgIRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZjOEAwlwcTEL
MAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBgNVBACQTDVNhbiBG
cmFuY2lzY28xGDAWBgNVBAoTD29yZzEuaHVybGV5LmXhYjEubmBkGA1UEAxMSY2Eu
b3JnMS5odXJsZXkubGFuMjIzY28xGDAWBgNVBAAsTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
azELMAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBgNVBACQTDVNh
biBGcmFuY2lzY28xGDAWBgNVBAAsTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
MS5odXJsZXkubGFuMjIzY28xGDAWBgNVBAQDAQgAEittt4NVK64er
98vMAydJ3ROx2RCqsF2AugnQY7lft1exXoRctfSgJSRssqpYTvWJ5ijyrtgjR7qf
ryKB91EMe6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAgzz7DtgICBI7t7I0dBUW4S90s6OP0+/0peXfIFy53wf4wCgYIKoZlZjOE
AwIDSAAwRQlhAN84ywXWxmAsdFULxMQqKOq78V0g1e1SyO6/U+SGZms8AiAbnSP/
/qep5onXRpRIWnmGOe4jSiXFehP3uW1I2hXjg==
-----END CERTIFICATE-----
2019-12-29 21:49:46.332 UTC [bccsp_sw] loadPrivateKey -> DEBU 034 Loading private key
[003d407522234792c36bacdc1fc0d312cc28fc07f8e3733590b0ca7861044502] at
[/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/keystore/003d407522234792c36bacdc
1fc0d312cc28fc07f8e3733590b0ca7861044502_sk]...
2019-12-29 21:49:46.332 UTC [msp.identity] newIdentity -> DEBU 035 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwIBAgIRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZjOEAwlwcTEL
MAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBgNVBACQTDVNhbiBG
cmFuY2lzY28xGDAWBgNVBAoTD29yZzEuaHVybGV5LmXhYjEubmBkGA1UEAxMSY2Eu
b3JnMS5odXJsZXkubGFuMjIzY28xGDAWBgNVBAAsTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
azELMAkGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBgNVBACQTDVNh
biBGcmFuY2lzY28xGDAWBgNVBAAsTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
MS5odXJsZXkubGFuMjIzY28xGDAWBgNVBAQDAQgAEittt4NVK64er
98vMAydJ3ROx2RCqsF2AugnQY7lft1exXoRctfSgJSRssqpYTvWJ5ijyrtgjR7qf
ryKB91EMe6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAgzz7DtgICBI7t7I0dBUW4S90s6OP0+/0peXfIFy53wf4wCgYIKoZlZjOE
AwIDSAAwRQlhAN84ywXWxmAsdFULxMQqKOq78V0g1e1SyO6/U+SGZms8AiAbnSP/
/qep5onXRpRIWnmGOe4jSiXFehP3uW1I2hXjg==
-----END CERTIFICATE-----
2019-12-29 21:49:46.332 UTC [msp] setupSigningIdentity -> DEBU 036 Signing identity expires
at 2029-12-26 21:44:00 +0000 UTC
2019-12-29 21:49:46.332 UTC [msp] Validate -> DEBU 037 MSP org1MSP validating identity
2019-12-29 21:49:46.333 UTC [main] SetOrdererEnv -> WARN 038 CORE_LOGGING_LEVEL is
no longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:46.333 UTC [msp] GetDefaultSigningIdentity -> DEBU 039 Obtaining default
signing identity
2019-12-29 21:49:46.334 UTC [grpc] DialContext -> DEBU 03a parsed scheme: ""
2019-12-29 21:49:46.334 UTC [grpc] DialContext -> DEBU 03b scheme "" not registered, fallback
to default scheme
2019-12-29 21:49:46.334 UTC [grpc] watcher -> DEBU 03c ccResolverWrapper: sending new
addresses to cc: [{peer0.org1.hurley.lab:7051 0 <nil>}]
2019-12-29 21:49:46.334 UTC [grpc] switchBalancer -> DEBU 03d ClientConn switching balancer
to "pick_first"
2019-12-29 21:49:46.334 UTC [grpc] HandleSubConnStateChange -> DEBU 03e
pickfirstBalancer: HandleSubConnStateChange: 0xc000324450, CONNECTING
```

```

2019-12-29 21:49:46.335 UTC [grpc] HandleSubConnStateChange -> DEBU 03f
pickfirstBalancer: HandleSubConnStateChange: 0xc000324450, READY
2019-12-29 21:49:46.335 UTC [channelCmd] InitCmdFactory -> INFO 040 Endorser and orderer
connections initialized
2019-12-29 21:49:46.336 UTC [msp.identity] Sign -> DEBU 041 Sign: plaintext:
0AB4070A5C08011A0C08FABEA4F00510...CDD8BD6738F91A080A000A000A000A00
2019-12-29 21:49:46.336 UTC [msp.identity] Sign -> DEBU 042 Sign: digest:
905CCD37B643A26D8D3CE5BD152720014B061E7B88C5B598A80E6E556B5A5336
2019-12-29 21:49:46.469 UTC [channelCmd] executeJoin -> INFO 043 Successfully submitted
proposal to join channel
++ echo pass
+ output=pass
+ set +x
Joining ch1 channel on peer peer0.org2.hurley.lab
++ docker exec peer0.org2.hurley.lab peer channel join -b /shared/ch1.block
2019-12-29 21:49:46.830 UTC [main] InitCmd -> WARN 001 CORE_LOGGING_LEVEL is no
longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:46.831 UTC [viperutil] getKeysRecursively -> DEBU 002 Found
map[string]interface{} value for peer.BCCSP
2019-12-29 21:49:46.831 UTC [viperutil] unmarshalJSON -> DEBU 003 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:46.831 UTC [viperutil] getKeysRecursively -> DEBU 004 Found real value for
peer.BCCSP.Default setting to string SW
2019-12-29 21:49:46.831 UTC [viperutil] getKeysRecursively -> DEBU 005 Found
map[string]interface{} value for peer.BCCSP.SW
2019-12-29 21:49:46.831 UTC [viperutil] unmarshalJSON -> DEBU 006 Unmarshal JSON: value
is not a string: 256
2019-12-29 21:49:46.831 UTC [viperutil] getKeysRecursively -> DEBU 007 Found real value for
peer.BCCSP.SW.Security setting to int 256
2019-12-29 21:49:46.831 UTC [viperutil] getKeysRecursively -> DEBU 008 Found
map[string]interface{} value for peer.BCCSP.SW.FileKeyStore
2019-12-29 21:49:46.831 UTC [viperutil] unmarshalJSON -> DEBU 009 Unmarshal JSON: value
cannot be unmarshalled: unexpected end of JSON input
2019-12-29 21:49:46.831 UTC [viperutil] getKeysRecursively -> DEBU 00a Found real value for
peer.BCCSP.SW.FileKeyStore.KeyStore setting to string
2019-12-29 21:49:46.832 UTC [viperutil] unmarshalJSON -> DEBU 00b Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:46.832 UTC [viperutil] getKeysRecursively -> DEBU 00c Found real value for
peer.BCCSP.SW.Hash setting to string SHA2
2019-12-29 21:49:46.832 UTC [viperutil] getKeysRecursively -> DEBU 00d Found
map[string]interface{} value for peer.BCCSP.PKCS11
2019-12-29 21:49:46.832 UTC [viperutil] unmarshalJSON -> DEBU 00e Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.832 UTC [viperutil] getKeysRecursively -> DEBU 00f Found real value for
peer.BCCSP.PKCS11.Security setting to <nil> <nil>
2019-12-29 21:49:46.832 UTC [viperutil] getKeysRecursively -> DEBU 010 Found
map[string]interface{} value for peer.BCCSP.PKCS11.FileKeyStore
2019-12-29 21:49:46.832 UTC [viperutil] unmarshalJSON -> DEBU 011 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.833 UTC [viperutil] getKeysRecursively -> DEBU 012 Found real value for
peer.BCCSP.PKCS11.FileKeyStore.KeyStore setting to <nil> <nil>
2019-12-29 21:49:46.833 UTC [viperutil] unmarshalJSON -> DEBU 013 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.833 UTC [viperutil] getKeysRecursively -> DEBU 014 Found real value for
peer.BCCSP.PKCS11.Library setting to <nil> <nil>
2019-12-29 21:49:46.833 UTC [viperutil] unmarshalJSON -> DEBU 015 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.833 UTC [viperutil] getKeysRecursively -> DEBU 016 Found real value for
peer.BCCSP.PKCS11.Label setting to <nil> <nil>
2019-12-29 21:49:46.834 UTC [viperutil] unmarshalJSON -> DEBU 017 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.834 UTC [viperutil] getKeysRecursively -> DEBU 018 Found real value for
peer.BCCSP.PKCS11.Pin setting to <nil> <nil>
2019-12-29 21:49:46.834 UTC [viperutil] unmarshalJSON -> DEBU 019 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:46.834 UTC [viperutil] getKeysRecursively -> DEBU 01a Found real value for
peer.BCCSP.PKCS11.Hash setting to <nil> <nil>

```

```

2019-12-29 21:49:46.834 UTC [viperutil] EnhancedExactUnmarshalKey -> DEBU 01b
map[peer.BCCSP:map[Default:SW SW:map[Hash:SHA2 Security:256
FileKeyStore:map[KeyStore:]] PKCS11:map[FileKeyStore:map[KeyStore:<nil>] Library:<nil>
Label:<nil> Pin:<nil> Hash:<nil> Security:<nil>]]]
2019-12-29 21:49:46.834 UTC [bccsp_sw] openKeyStore -> DEBU 01c KeyStore opened at [/etc/
hyperledger/msp/users/Admin@org2.hurley.lab/msp/keystore]...done
2019-12-29 21:49:46.834 UTC [bccsp] initBCCSP -> DEBU 01d Initialize BCCSP [SW]
2019-12-29 21:49:46.834 UTC [msp] getPemMaterialFromDir -> DEBU 01e Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/signcerts
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 01f Inspecting file
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/signcerts/Admin@org2.hurley.lab-
cert.pem
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 020 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/cacerts
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 021 Inspecting file
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/cacerts/ca.org2.hurley.lab-cert.pem
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 022 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/admincerts
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 023 Inspecting file
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/admincerts/Admin@org2.hurley.lab-
cert.pem
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 024 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/intermediatecerts
2019-12-29 21:49:46.835 UTC [msp] getMspConfig -> DEBU 025 Intermediate certs folder not
found at [/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/intermediatecerts]. Skipping.
[stat /etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/intermediatecerts: no such file or
directory]
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 026 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsacerts
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 027 Inspecting file
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsacerts/tlsca.org2.hurley.lab-cert.pem
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 028 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsintermediatecerts
2019-12-29 21:49:46.835 UTC [msp] getMspConfig -> DEBU 029 TLS intermediate certs folder
not found at [/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsintermediatecerts].
Skipping. [stat /etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsintermediatecerts: no
such file or directory]
2019-12-29 21:49:46.835 UTC [msp] getPemMaterialFromDir -> DEBU 02a Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/crls
2019-12-29 21:49:46.835 UTC [msp] getMspConfig -> DEBU 02b crls folder not found at
[/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/crls]. Skipping. [stat
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/crls: no such file or directory]
2019-12-29 21:49:46.835 UTC [msp] getMspConfig -> DEBU 02c MSP configuration file not
found at [/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/config.yaml]: [stat
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/config.yaml: no such file or directory]
2019-12-29 21:49:46.836 UTC [msp] newBccspMsp -> DEBU 02d Creating BCCSP-based MSP
instance
2019-12-29 21:49:46.836 UTC [msp] New -> DEBU 02e Creating Cache-MSP instance
2019-12-29 21:49:46.836 UTC [msp] loadLocaMSP -> DEBU 02f Created new local MSP
2019-12-29 21:49:46.836 UTC [msp] Setup -> DEBU 030 Setting up MSP instance org2MSP
2019-12-29 21:49:46.836 UTC [msp.identity] newIdentity -> DEBU 031 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICTTCCAfOgAwIBAgIQLwG3le49FkbtixviNjAd3TAKBggqhkJOPQQDAjBxMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTEwMNU2FulEZY
YW5jaXNjbzEYMBYGA1UEChMPb3JnMi5odXJsZXkubGFIMRswGQYDVQQDEExjYS5v
cmcyLmh1cmxleS5sYWlwHhcNMTkxMjMjE0NDAwWhcNMTkxMjMjE0NDAwWjBx
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTEwMNU2F
u
IEZY
YW5jaXNjbzEYMBYGA1UEChMPb3JnMi5odXJsZXkubGFIMRswGQYDVQQDEExj
YS5vcmcyLmh1cmxleS5sYWlwWTATBgqhkJOPQIBBggqhkJOPQMBBwNCAASd5CT4
5GIYATnLuGufmYVG45216hXRuIP4bpQwJyAMia3yBxJ7B+3t8JCp5jTg2gVK3
DmL0UEW9owRzl0xNo20wazAOBgNVHQ8BAf8EBAMCAAYwHQYDVR0IBBYwFAIKwYB
BQUHAWIGCCsGAQUFBwMBMA8GA1UdEwEB/wQFMAMBAf8wKQYDVR0OBCIEINYHI7yR
gXv24T4njyONz9uCHgiL1f8t6lqdzBeqieGtMAoGCCqGSM49BAMCA0gAMEUCIQCb
rxr+Mn+lq59QPPIAD7JNvk58Ldy7z8L6FGW2wyYdnglgl32YycgyYv77LIHb9MV/
MNzocWvyQ7c6JaQ1kN5oovs=
-----END CERTIFICATE-----

```

```
2019-12-29 21:49:46.836 UTC [msp.identity] newIdentity -> DEBU 032 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwIBAgIRAJk8IJKIFe0/oohn/WkXZkwCgYIKoZlZj0EAWlwTEL
MAKGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWExFjAUBGNVBAcTDVNhbiBG
cmFuY2lzY28xGDAWBGNVBAoTD29yZzluaHVybGV5LmXhYjEhYjEhYjEhYjEhYjEh
b3JnMi5odXJsZXkubGFIMB4XDTE5MTIyOTIxNDQwMFOxDTI5MTIyNjIxNDQwMFow
azELMAKGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWExFjAUBGNVBAcTDVNh
biBGcmFuY2lzY28xGDAWBGNVBAcTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
Mi5odXJsZXkubGFIMFkwEwYHkoZlZj0CAQYIKoZlZj0DAQcDQgAESaZiI3Pntosm
wjd+I6nBk9dUhYC2cJJYv2JvivyT1YqE4xOVCDM9SxuGFLnbtFul/JI9EC0oso+pJ
IrSawyCDvKNNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAg1geXvJGBe/bhPiePI43P24leClvV/y3qWp3MF6qJ4a0wCgYIKoZlZj0E
AwIDSAAwRQlhAjc3KXwJZtsUHYwy42SBXZKsrdOXPhb7LHY1faRaWLKAiBZ7jfs
2boT1MKNKRpEaAUnm7iNevzvGMZy+pMxBexohw==
-----END CERTIFICATE-----

2019-12-29 21:49:46.850 UTC [msp.identity] newIdentity -> DEBU 033 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwIBAgIRAJk8IJKIFe0/oohn/WkXZkwCgYIKoZlZj0EAWlwTEL
MAKGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWExFjAUBGNVBAcTDVNhbiBG
cmFuY2lzY28xGDAWBGNVBAoTD29yZzluaHVybGV5LmXhYjEhYjEhYjEhYjEhYjEh
b3JnMi5odXJsZXkubGFIMB4XDTE5MTIyOTIxNDQwMFOxDTI5MTIyNjIxNDQwMFow
azELMAKGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWExFjAUBGNVBAcTDVNh
biBGcmFuY2lzY28xGDAWBGNVBAcTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
Mi5odXJsZXkubGFIMFkwEwYHkoZlZj0CAQYIKoZlZj0DAQcDQgAESaZiI3Pntosm
wjd+I6nBk9dUhYC2cJJYv2JvivyT1YqE4xOVCDM9SxuGFLnbtFul/JI9EC0oso+pJ
IrSawyCDvKNNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAg1geXvJGBe/bhPiePI43P24leClvV/y3qWp3MF6qJ4a0wCgYIKoZlZj0E
AwIDSAAwRQlhAjc3KXwJZtsUHYwy42SBXZKsrdOXPhb7LHY1faRaWLKAiBZ7jfs
2boT1MKNKRpEaAUnm7iNevzvGMZy+pMxBexohw==
-----END CERTIFICATE-----

2019-12-29 21:49:46.851 UTC [bccsp_sw] loadPrivateKey -> DEBU 034 Loading private key
[0407e312f42a9d2e4047c347d4326564ef99c4f7ab5b30c7a5469e7e1443c66d] at
[/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/keystore/0407e312f42a9d2e4047c347d
4326564ef99c4f7ab5b30c7a5469e7e1443c66d_sk]...

2019-12-29 21:49:46.851 UTC [msp.identity] newIdentity -> DEBU 035 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwIBAgIRAJk8IJKIFe0/oohn/WkXZkwCgYIKoZlZj0EAWlwTEL
MAKGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWExFjAUBGNVBAcTDVNhbiBG
cmFuY2lzY28xGDAWBGNVBAoTD29yZzluaHVybGV5LmXhYjEhYjEhYjEhYjEhYjEh
b3JnMi5odXJsZXkubGFIMB4XDTE5MTIyOTIxNDQwMFOxDTI5MTIyNjIxNDQwMFow
azELMAKGA1UEBhMCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWExFjAUBGNVBAcTDVNh
biBGcmFuY2lzY28xGDAWBGNVBAcTBmNsaWVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
Mi5odXJsZXkubGFIMFkwEwYHkoZlZj0CAQYIKoZlZj0DAQcDQgAESaZiI3Pntosm
wjd+I6nBk9dUhYC2cJJYv2JvivyT1YqE4xOVCDM9SxuGFLnbtFul/JI9EC0oso+pJ
IrSawyCDvKNNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAg1geXvJGBe/bhPiePI43P24leClvV/y3qWp3MF6qJ4a0wCgYIKoZlZj0E
AwIDSAAwRQlhAjc3KXwJZtsUHYwy42SBXZKsrdOXPhb7LHY1faRaWLKAiBZ7jfs
2boT1MKNKRpEaAUnm7iNevzvGMZy+pMxBexohw==
-----END CERTIFICATE-----

2019-12-29 21:49:46.851 UTC [msp] setupSigningIdentity -> DEBU 036 Signing identity expires
at 2029-12-26 21:44:00 +0000 UTC

2019-12-29 21:49:46.852 UTC [msp] Validate -> DEBU 037 MSP org2MSP validating identity

2019-12-29 21:49:46.853 UTC [main] SetOrdererEnv -> WARN 038 CORE_LOGGING_LEVEL is
no longer supported, please use the FABRIC_LOGGING_SPEC environment variable

2019-12-29 21:49:46.853 UTC [msp] GetDefaultSigningIdentity -> DEBU 039 Obtaining default
signing identity

2019-12-29 21:49:46.854 UTC [grpc] DialContext -> DEBU 03a parsed scheme: ""

2019-12-29 21:49:46.854 UTC [grpc] DialContext -> DEBU 03b scheme "" not registered, fallback
to default scheme

2019-12-29 21:49:46.854 UTC [grpc] watcher -> DEBU 03c ccResolverWrapper: sending new
addresses to cc: [{peer0.org2.hurley.lab:7051 0 <nii>}]

2019-12-29 21:49:46.854 UTC [grpc] switchBalancer -> DEBU 03d ClientConn switching balancer
to "pick_first"

2019-12-29 21:49:46.854 UTC [grpc] HandleSubConnStateChange -> DEBU 03e
pickfirstBalancer: HandleSubConnStateChange: 0xc000082f70, CONNECTING

2019-12-29 21:49:46.855 UTC [grpc] HandleSubConnStateChange -> DEBU 03f
pickfirstBalancer: HandleSubConnStateChange: 0xc000082f70, READY
```

```
2019-12-29 21:49:46.856 UTC [channelCmd] InitCmdFactory -> INFO 040 Endorser and orderer
connections initialized
2019-12-29 21:49:46.856 UTC [msp.identity] Sign -> DEBU 041 Sign: plaintext:
0AB4070A5C08011A0C08FABEA4F00510...CDD8BD6738F91A080A000A000A000A00
2019-12-29 21:49:46.856 UTC [msp.identity] Sign -> DEBU 042 Sign: digest:
7F7DF0EF0DFA8296C5BC78B9A24AC815FE78AEDA1047E2313AA66ECEAB622C14
2019-12-29 21:49:46.992 UTC [channelCmd] executeJoin -> INFO 043 Successfully submitted
proposal to join channel
++ echo pass
+ output=pass
+ set +x
Creating ch1 anchor block in peer peer0.org1.hurley.lab
2019-12-29 21:49:47.362 UTC [main] InitCmd -> WARN 001 CORE_LOGGING_LEVEL is no
longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:47.363 UTC [viperutil] getKeysRecursively -> DEBU 002 Found
map[string]interface{} value for peer.BCCSP
2019-12-29 21:49:47.363 UTC [viperutil] getKeysRecursively -> DEBU 003 Found
map[string]interface{} value for peer.BCCSP.SW
2019-12-29 21:49:47.363 UTC [viperutil] unmarshalJSON -> DEBU 004 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:47.363 UTC [viperutil] getKeysRecursively -> DEBU 005 Found real value for
peer.BCCSP.SW.Hash setting to string SHA2
2019-12-29 21:49:47.363 UTC [viperutil] unmarshalJSON -> DEBU 006 Unmarshal JSON: value
is not a string: 256
2019-12-29 21:49:47.363 UTC [viperutil] getKeysRecursively -> DEBU 007 Found real value for
peer.BCCSP.SW.Security setting to int 256
2019-12-29 21:49:47.363 UTC [viperutil] getKeysRecursively -> DEBU 008 Found
map[string]interface{} value for peer.BCCSP.SW.FileKeyStore
2019-12-29 21:49:47.363 UTC [viperutil] unmarshalJSON -> DEBU 009 Unmarshal JSON: value
cannot be unmarshalled: unexpected end of JSON input
2019-12-29 21:49:47.363 UTC [viperutil] getKeysRecursively -> DEBU 00a Found real value for
peer.BCCSP.SW.FileKeyStore.KeyStore setting to string
2019-12-29 21:49:47.364 UTC [viperutil] getKeysRecursively -> DEBU 00b Found
map[string]interface{} value for peer.BCCSP.PKCS11
2019-12-29 21:49:47.364 UTC [viperutil] unmarshalJSON -> DEBU 00c Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.364 UTC [viperutil] getKeysRecursively -> DEBU 00d Found real value for
peer.BCCSP.PKCS11.Security setting to <nil> <nil>
2019-12-29 21:49:47.364 UTC [viperutil] getKeysRecursively -> DEBU 00e Found
map[string]interface{} value for peer.BCCSP.PKCS11.FileKeyStore
2019-12-29 21:49:47.364 UTC [viperutil] unmarshalJSON -> DEBU 00f Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.364 UTC [viperutil] getKeysRecursively -> DEBU 010 Found real value for
peer.BCCSP.PKCS11.FileKeyStore.KeyStore setting to <nil> <nil>
2019-12-29 21:49:47.365 UTC [viperutil] unmarshalJSON -> DEBU 011 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.365 UTC [viperutil] getKeysRecursively -> DEBU 012 Found real value for
peer.BCCSP.PKCS11.Library setting to <nil> <nil>
2019-12-29 21:49:47.365 UTC [viperutil] unmarshalJSON -> DEBU 013 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.365 UTC [viperutil] getKeysRecursively -> DEBU 014 Found real value for
peer.BCCSP.PKCS11.Label setting to <nil> <nil>
2019-12-29 21:49:47.365 UTC [viperutil] unmarshalJSON -> DEBU 015 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.365 UTC [viperutil] getKeysRecursively -> DEBU 016 Found real value for
peer.BCCSP.PKCS11.Pin setting to <nil> <nil>
2019-12-29 21:49:47.365 UTC [viperutil] unmarshalJSON -> DEBU 017 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.366 UTC [viperutil] getKeysRecursively -> DEBU 018 Found real value for
peer.BCCSP.PKCS11.Hash setting to <nil> <nil>
2019-12-29 21:49:47.366 UTC [viperutil] unmarshalJSON -> DEBU 019 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:47.366 UTC [viperutil] getKeysRecursively -> DEBU 01a Found real value for
peer.BCCSP.Default setting to string SW
2019-12-29 21:49:47.366 UTC [viperutil] EnhancedExactUnmarshalKey -> DEBU 01b
map[peer.BCCSP:map[SW:map[Hash:SHA2 Security:256 FileKeyStore:map[KeyStore:]]
```

```

PKCS11:map[Label:<nil> Pin:<nil> Hash:<nil> Security:<nil> FileKeyStore:map[KeyStore:<nil>]
Library:<nil>] Default:SW]]
2019-12-29 21:49:47.366 UTC [bccsp_sw] openKeyStore -> DEBU 01c KeyStore opened at [/etc/
hyperledger/msp/users/Admin@org1.hurley.lab/msp/keystore]...done
2019-12-29 21:49:47.366 UTC [bccsp] initBCCSP -> DEBU 01d Initialize BCCSP [SW]
2019-12-29 21:49:47.366 UTC [msp] getPemMaterialFromDir -> DEBU 01e Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/signcerts
2019-12-29 21:49:47.366 UTC [msp] getPemMaterialFromDir -> DEBU 01f Inspecting file
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/signcerts/Admin@org1.hurley.lab-
cert.pem
2019-12-29 21:49:47.366 UTC [msp] getPemMaterialFromDir -> DEBU 020 Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/cacerts
2019-12-29 21:49:47.366 UTC [msp] getPemMaterialFromDir -> DEBU 021 Inspecting file
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/cacerts/ca.org1.hurley.lab-cert.pem
2019-12-29 21:49:47.366 UTC [msp] getPemMaterialFromDir -> DEBU 022 Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/admincerts
2019-12-29 21:49:47.366 UTC [msp] getPemMaterialFromDir -> DEBU 023 Inspecting file
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/admincerts/Admin@org1.hurley.lab-
cert.pem
2019-12-29 21:49:47.366 UTC [msp] getPemMaterialFromDir -> DEBU 024 Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts
2019-12-29 21:49:47.366 UTC [msp] getMspConfig -> DEBU 025 Intermediate certs folder not
found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts]. Skipping.
[stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/intermediatecerts: no such file or
directory]
2019-12-29 21:49:47.367 UTC [msp] getPemMaterialFromDir -> DEBU 026 Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsacerts
2019-12-29 21:49:47.367 UTC [msp] getPemMaterialFromDir -> DEBU 027 Inspecting file
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsacerts/tlsca.org1.hurley.lab-cert.pem
2019-12-29 21:49:47.367 UTC [msp] getPemMaterialFromDir -> DEBU 028 Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts
2019-12-29 21:49:47.367 UTC [msp] getMspConfig -> DEBU 029 TLS intermediate certs folder
not found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts].
Skipping. [stat /etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/tlsintermediatecerts: no
such file or directory]
2019-12-29 21:49:47.367 UTC [msp] getPemMaterialFromDir -> DEBU 02a Reading directory
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls
2019-12-29 21:49:47.367 UTC [msp] getMspConfig -> DEBU 02b crls folder not found at
[/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls]. Skipping. [stat
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/crls: no such file or directory]
2019-12-29 21:49:47.367 UTC [msp] getMspConfig -> DEBU 02c MSP configuration file not
found at [/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/config.yaml]: [stat
/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/config.yaml: no such file or directory]
2019-12-29 21:49:47.367 UTC [msp] newBccspMsp -> DEBU 02d Creating BCCSP-based MSP
instance
2019-12-29 21:49:47.367 UTC [msp] New -> DEBU 02e Creating Cache-MSP instance
2019-12-29 21:49:47.367 UTC [msp] loadLocaMSP -> DEBU 02f Created new local MSP
2019-12-29 21:49:47.367 UTC [msp] Setup -> DEBU 030 Setting up MSP instance org1MSP
2019-12-29 21:49:47.368 UTC [msp.identity] newIdentity -> DEBU 031 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICTDCCAfOgAwIBAgIQ77TyVkcPjS+f2MkExfvaozAKBggqhkJOPQQDAjBxMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTEwMBQGA1UEBxMNU2FulEZy
YW5jaXNjbzEYMBYGA1UEChMPb3JnMS5odXJsZXkubGFjMRswGQYDVQQDEJjYS5v
cmcxLmh1cmxleS5sYWlwHhcNMTkxMjIjE0NDAwWhcNMjIjE0NDAwWjBx
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcn5pYTEwMBQGA1UEBxMNU2F
u
IEZyYW5jaXNjbzEYMBYGA1UEChMPb3JnMS5odXJsZXkubGFjMRswGQYDVQQDEJj
YS5vcmcxLmh1cmxleS5sYWlwWTATBgcqhkJOPQIBBgqhkJOPQMBBwNCAASf74s3
k5eu1gRBP1qcgmfZE6rjzYxJvW/Dfw7xmlfutAnPH7LJ/Ql1vpW7RH2G1MrUsCj
4mAKL7n2RbsYD7ylo20wazAOBgNVHQ8BAf8EBAMCAAYwHQYDVR0IBBYwFAYIKwYB
BQUHAWIGCCsGAQUFBwMBMA8GA1UdEwEB/wQFMAMBAf8wKQYDVR0OBCIEIM8+w7YC
AgZe7e5dHQVFuEvdLOjj9Pv9KXI3yBcud8H+MAoGCCqGSM49BAMCA0cAMEQCIB5I
sqHu8ljnpluXUS1m6bP95aRMqkgef2jYVI3SYMAiB6v5zNdtc9/pNnIB+M2H+u
IL5gTizjEfx8bWCNMxc7UQ==
-----END CERTIFICATE-----
2019-12-29 21:49:47.368 UTC [msp.identity] newIdentity -> DEBU 032 Creating identity instance
for cert -----BEGIN CERTIFICATE-----

```

```
MIICKDCCAc6gAwlBAglRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZj0EAwlcTEL
MAKGA1UEBhMVCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBGNVBACtDVNhbG
cmFuY2lyZ28xGDAWBgNVBAoTD29yZzEuaHVybGV5LmXhYjE5bWVkbG91UEAxMSY2Eu
b3JnMS5odXJsZXkubGFIMB4XDTE5MTlyOTIxNDQwMFoXDTI5MTlyNjIxNDQwMFow
azELMAKGA1UEBhMVCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBGNVBACtDVNhb
biBGcmFuY2lyZ28xGDAWBgNVBAAsTBmNsaVVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
MS5odXJsZXkubGFIMFkwEwYHkoZlZj0CAQYIKoZlZj0DAQcDQgAEiitt4NVKG4er
98vMAydJ3ROx2RCqsF2AugnQY7lft1exXoRctfSgJSRssqpYTvWJ5ijyrtgjR7qf
ryKB91EMe6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAgzz7DtgICBI7t7I0dBUW4S90s6OP0+/0peXfIFy53wf4wCgYIKoZlZj0E
AwIDSAAwRQlhAN84ywXWxmAsdFULxMQqKOq78V0g1e1SyO6/U+SGZms8AiAbnSP/
/qep5onXRpRIWnmGOe4jSiXFehP3uW1I2hXjg==
-----END CERTIFICATE-----
2019-12-29 21:49:47.377 UTC [msp.identity] newIdentity -> DEBU 033 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwlBAglRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZj0EAwlcTEL
MAKGA1UEBhMVCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBGNVBACtDVNhbG
cmFuY2lyZ28xGDAWBgNVBAoTD29yZzEuaHVybGV5LmXhYjE5bWVkbG91UEAxMSY2Eu
b3JnMS5odXJsZXkubGFIMB4XDTE5MTlyOTIxNDQwMFoXDTI5MTlyNjIxNDQwMFow
azELMAKGA1UEBhMVCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBGNVBACtDVNhb
biBGcmFuY2lyZ28xGDAWBgNVBAAsTBmNsaVVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
MS5odXJsZXkubGFIMFkwEwYHkoZlZj0CAQYIKoZlZj0DAQcDQgAEiitt4NVKG4er
98vMAydJ3ROx2RCqsF2AugnQY7lft1exXoRctfSgJSRssqpYTvWJ5ijyrtgjR7qf
ryKB91EMe6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAgzz7DtgICBI7t7I0dBUW4S90s6OP0+/0peXfIFy53wf4wCgYIKoZlZj0E
AwIDSAAwRQlhAN84ywXWxmAsdFULxMQqKOq78V0g1e1SyO6/U+SGZms8AiAbnSP/
/qep5onXRpRIWnmGOe4jSiXFehP3uW1I2hXjg==
-----END CERTIFICATE-----
2019-12-29 21:49:47.377 UTC [bccsp_sw] loadPrivateKey -> DEBU 034 Loading private key
[003d407522234792c36bacdc1fc0d312cc28fc07f8e3733590b0ca7861044502] at
[/etc/hyperledger/msp/users/Admin@org1.hurley.lab/msp/keystore/003d407522234792c36bacdc
1fc0d312cc28fc07f8e3733590b0ca7861044502_sk]...
2019-12-29 21:49:47.377 UTC [msp.identity] newIdentity -> DEBU 035 Creating identity instance
for cert -----BEGIN CERTIFICATE-----
MIICKDCCAc6gAwlBAglRAI9s7+LL1opCjLct6HDjBoQwCgYIKoZlZj0EAwlcTEL
MAKGA1UEBhMVCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBGNVBACtDVNhbG
cmFuY2lyZ28xGDAWBgNVBAoTD29yZzEuaHVybGV5LmXhYjE5bWVkbG91UEAxMSY2Eu
b3JnMS5odXJsZXkubGFIMB4XDTE5MTlyOTIxNDQwMFoXDTI5MTlyNjIxNDQwMFow
azELMAKGA1UEBhMVCVVMxEzARBgNVBAGTCkNhbGlmb3JuaWEeFjAUBGNVBACtDVNhb
biBGcmFuY2lyZ28xGDAWBgNVBAAsTBmNsaVVudDEeMBwGA1UEAwwVQWRtaW5Ab3Jn
MS5odXJsZXkubGFIMFkwEwYHkoZlZj0CAQYIKoZlZj0DAQcDQgAEiitt4NVKG4er
98vMAydJ3ROx2RCqsF2AugnQY7lft1exXoRctfSgJSRssqpYTvWJ5ijyrtgjR7qf
ryKB91EMe6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAgzz7DtgICBI7t7I0dBUW4S90s6OP0+/0peXfIFy53wf4wCgYIKoZlZj0E
AwIDSAAwRQlhAN84ywXWxmAsdFULxMQqKOq78V0g1e1SyO6/U+SGZms8AiAbnSP/
/qep5onXRpRIWnmGOe4jSiXFehP3uW1I2hXjg==
-----END CERTIFICATE-----
2019-12-29 21:49:47.378 UTC [msp] setupSigningIdentity -> DEBU 036 Signing identity expires
at 2029-12-26 21:44:00 +0000 UTC
2019-12-29 21:49:47.378 UTC [msp] Validate -> DEBU 037 MSP org1MSP validating identity
2019-12-29 21:49:47.378 UTC [main] SetOrdererEnv -> WARN 038 CORE_LOGGING_LEVEL is
no longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:47.381 UTC [msp] GetDefaultSigningIdentity -> DEBU 039 Obtaining default
signing identity
2019-12-29 21:49:47.382 UTC [grpc] DialContext -> DEBU 03a parsed scheme: ""
2019-12-29 21:49:47.382 UTC [grpc] DialContext -> DEBU 03b scheme "" not registered, fallback
to default scheme
2019-12-29 21:49:47.382 UTC [grpc] watcher -> DEBU 03c ccResolverWrapper: sending new
addresses to cc: [{orderer.hurley.lab:7050 0 <nil>}]
2019-12-29 21:49:47.382 UTC [grpc] switchBalancer -> DEBU 03d ClientConn switching balancer
to "pick_first"
2019-12-29 21:49:47.382 UTC [grpc] HandleSubConnStateChange -> DEBU 03e
pickfirstBalancer: HandleSubConnStateChange: 0xc000356300, CONNECTING
2019-12-29 21:49:47.383 UTC [grpc] HandleSubConnStateChange -> DEBU 03f
pickfirstBalancer: HandleSubConnStateChange: 0xc000356300, READY
2019-12-29 21:49:47.383 UTC [channelCmd] InitCmdFactory -> INFO 040 Endorser and orderer
connections initialized
```



```

2019-12-29 21:49:47.384 UTC [msp] GetDefaultSigningIdentity -> DEBU 041 Obtaining default
signing identity
2019-12-29 21:49:47.384 UTC [msp] GetDefaultSigningIdentity -> DEBU 042 Obtaining default
signing identity
2019-12-29 21:49:47.385 UTC [msp.identity] Sign -> DEBU 043 Sign: plaintext:
0AB6060A076F7267314D535012AA062D...2A0641646D696E732A0641646D696E73
2019-12-29 21:49:47.385 UTC [msp.identity] Sign -> DEBU 044 Sign: digest:
8CEA8A5E98E71362F7288EB83AF20288A0B8F957A7805A8ADAAD81397A75A943
2019-12-29 21:49:47.385 UTC [msp] GetDefaultSigningIdentity -> DEBU 045 Obtaining default
signing identity
2019-12-29 21:49:47.385 UTC [msp] GetDefaultSigningIdentity -> DEBU 046 Obtaining default
signing identity
2019-12-29 21:49:47.385 UTC [msp.identity] Sign -> DEBU 047 Sign: plaintext:
0AE7060A0F08021A0608FBBEA4F00522...74BAC472591624E0D605BEE7933BD67F
2019-12-29 21:49:47.385 UTC [msp.identity] Sign -> DEBU 048 Sign: digest:
86E0FC396FAA35B317232DA1A329D514917CE7223138E6187E9578AEA66CC731
2019-12-29 21:49:47.385 UTC [grpc] DialContext -> DEBU 049 parsed scheme: ""
2019-12-29 21:49:47.385 UTC [grpc] DialContext -> DEBU 04a scheme "" not registered, fallback
to default scheme
2019-12-29 21:49:47.385 UTC [grpc] watcher -> DEBU 04b ccResolverWrapper: sending new
addresses to cc: [{orderer.hurley.lab:7050 0 <nil>}]
2019-12-29 21:49:47.386 UTC [grpc] switchBalancer -> DEBU 04c ClientConn switching balancer
to "pick_first"
2019-12-29 21:49:47.386 UTC [grpc] HandleSubConnStateChange -> DEBU 04d
pickfirstBalancer: HandleSubConnStateChange: 0xc0003e8370, CONNECTING
2019-12-29 21:49:47.387 UTC [grpc] HandleSubConnStateChange -> DEBU 04e
pickfirstBalancer: HandleSubConnStateChange: 0xc0003e8370, READY
2019-12-29 21:49:47.399 UTC [channelCmd] update -> INFO 04f Successfully submitted channel
update
Creating ch1 anchor block in peer peer0.org2.hurley.lab
2019-12-29 21:49:47.794 UTC [main] InitCmd -> WARN 001 CORE_LOGGING_LEVEL is no
longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:47.794 UTC [viperutil] getKeysRecursively -> DEBU 002 Found
map[string]interface{} value for peer.BCCSP
2019-12-29 21:49:47.795 UTC [viperutil] getKeysRecursively -> DEBU 003 Found
map[string]interface{} value for peer.BCCSP.PKCS11
2019-12-29 21:49:47.795 UTC [viperutil] unmarshalJSON -> DEBU 004 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.795 UTC [viperutil] getKeysRecursively -> DEBU 005 Found real value for
peer.BCCSP.PKCS11.Hash setting to <nil> <nil>
2019-12-29 21:49:47.795 UTC [viperutil] unmarshalJSON -> DEBU 006 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.795 UTC [viperutil] getKeysRecursively -> DEBU 007 Found real value for
peer.BCCSP.PKCS11.Security setting to <nil> <nil>
2019-12-29 21:49:47.795 UTC [viperutil] getKeysRecursively -> DEBU 008 Found
map[string]interface{} value for peer.BCCSP.PKCS11.FileKeyStore
2019-12-29 21:49:47.796 UTC [viperutil] unmarshalJSON -> DEBU 009 Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.796 UTC [viperutil] getKeysRecursively -> DEBU 00a Found real value for
peer.BCCSP.PKCS11.FileKeyStore.KeyStore setting to <nil> <nil>
2019-12-29 21:49:47.796 UTC [viperutil] unmarshalJSON -> DEBU 00b Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.796 UTC [viperutil] getKeysRecursively -> DEBU 00c Found real value for
peer.BCCSP.PKCS11.Library setting to <nil> <nil>
2019-12-29 21:49:47.796 UTC [viperutil] unmarshalJSON -> DEBU 00d Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.796 UTC [viperutil] getKeysRecursively -> DEBU 00e Found real value for
peer.BCCSP.PKCS11.Label setting to <nil> <nil>
2019-12-29 21:49:47.796 UTC [viperutil] unmarshalJSON -> DEBU 00f Unmarshal JSON: value
is not a string: <nil>
2019-12-29 21:49:47.796 UTC [viperutil] getKeysRecursively -> DEBU 010 Found real value for
peer.BCCSP.PKCS11.Pin setting to <nil> <nil>
2019-12-29 21:49:47.797 UTC [viperutil] unmarshalJSON -> DEBU 011 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:47.797 UTC [viperutil] getKeysRecursively -> DEBU 012 Found real value for
peer.BCCSP.Default setting to string SW

```

```

2019-12-29 21:49:47.797 UTC [viperutil] getKeysRecursively -> DEBU 013 Found
map[string]interface{} value for peer.BCCSP.SW
2019-12-29 21:49:47.797 UTC [viperutil] unmarshalJSON -> DEBU 014 Unmarshal JSON: value
is not a string: 256
2019-12-29 21:49:47.797 UTC [viperutil] getKeysRecursively -> DEBU 015 Found real value for
peer.BCCSP.SW.Security setting to int 256
2019-12-29 21:49:47.797 UTC [viperutil] getKeysRecursively -> DEBU 016 Found
map[string]interface{} value for peer.BCCSP.SW.FileKeyStore
2019-12-29 21:49:47.797 UTC [viperutil] unmarshalJSON -> DEBU 017 Unmarshal JSON: value
cannot be unmarshalled: unexpected end of JSON input
2019-12-29 21:49:47.797 UTC [viperutil] getKeysRecursively -> DEBU 018 Found real value for
peer.BCCSP.SW.FileKeyStore.KeyStore setting to string
2019-12-29 21:49:47.797 UTC [viperutil] unmarshalJSON -> DEBU 019 Unmarshal JSON: value
cannot be unmarshalled: invalid character 'S' looking for beginning of value
2019-12-29 21:49:47.797 UTC [viperutil] getKeysRecursively -> DEBU 01a Found real value for
peer.BCCSP.SW.Hash setting to string SHA2
2019-12-29 21:49:47.797 UTC [viperutil] EnhancedExactUnmarshalKey -> DEBU 01b
map[peer.BCCSP:map[PKCS11:map[Pin:<nil> Hash:<nil> Security:<nil>
FileKeyStore:map[KeyStore:<nil> Library:<nil> Label:<nil>] Default:SW SW:map[Security:256
FileKeyStore:map[KeyStore:] Hash:SHA2]]]
2019-12-29 21:49:47.798 UTC [bccsp_sw] openKeyStore -> DEBU 01c KeyStore opened at [/etc/
hyperledger/msp/users/Admin@org2.hurley.lab/msp/keystore]...done
2019-12-29 21:49:47.798 UTC [bccsp] initBCCSP -> DEBU 01d Initialize BCCSP [SW]
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 01e Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/signcerts
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 01f Inspecting file
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/signcerts/Admin@org2.hurley.lab-
cert.pem
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 020 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/cacerts
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 021 Inspecting file
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/cacerts/ca.org2.hurley.lab-cert.pem
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 022 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/admincerts
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 023 Inspecting file
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/admincerts/Admin@org2.hurley.lab-
cert.pem
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 024 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/intermediatecerts
2019-12-29 21:49:47.798 UTC [msp] getMspConfig -> DEBU 025 Intermediate certs folder not
found at [/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/intermediatecerts]. Skipping.
[stat /etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/intermediatecerts: no such file or
directory]
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 026 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsacerts
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 027 Inspecting file
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsacerts/tlsca.org2.hurley.lab-cert.pem
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 028 Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsintermediatecerts
2019-12-29 21:49:47.798 UTC [msp] getMspConfig -> DEBU 029 TLS intermediate certs folder
not found at [/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsintermediatecerts].
Skipping. [stat /etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/tlsintermediatecerts: no
such file or directory]
2019-12-29 21:49:47.798 UTC [msp] getPemMaterialFromDir -> DEBU 02a Reading directory
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/crls
2019-12-29 21:49:47.799 UTC [msp] getMspConfig -> DEBU 02b crls folder not found at
[/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/crls]. Skipping. [stat
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/crls: no such file or directory]
2019-12-29 21:49:47.799 UTC [msp] getMspConfig -> DEBU 02c MSP configuration file not
found at [/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/config.yaml]: [stat
/etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/config.yaml: no such file or directory]
2019-12-29 21:49:47.799 UTC [msp] newBccspMsp -> DEBU 02d Creating BCCSP-based MSP
instance
2019-12-29 21:49:47.799 UTC [msp] New -> DEBU 02e Creating Cache-MSP instance
2019-12-29 21:49:47.799 UTC [msp] loadLocaMSP -> DEBU 02f Created new local MSP
2019-12-29 21:49:47.799 UTC [msp] Setup -> DEBU 030 Setting up MSP instance org2MSP

```

2019-12-29 21:49:47.799 UTC [msp.identity] newIdentity -> DEBU 031 Creating identity instance for cert -----BEGIN CERTIFICATE-----
MIICTTCCAfoGawlBAglQLwG3le49FkbtlxviNjAd3TAKBggghkjOPQQDAjBxMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcml5pYUWMBQGA1UEBxMNU2FulEZy
YW5jaXNjbzEYMBYGA1UEChMPb3JnMi5odXJsZXkubGFIMRswGQYDVQQDEwJYYS5v
cmcyLmh1cmxleS5sYWlwHhcNMTkxMjI5MjE0NDAwWhcNMjI5MjE0NDAwWjBx
MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcml5pYUWMBQGA1UEBxMNU2F
u

IEZyYW5jaXNjbzEYMBYGA1UEChMPb3JnMi5odXJsZXkubGFIMRswGQYDVQQDEwJY
YS5vcmcyLmh1cmxleS5sYWlwWTATBgcqhkiOPQIBBggghkiOPQMBBwNCAASd5CT4
5GIYATnLuGufmYVG45216hXRuIP4bpQwJyAMia3yBxJ7B+3t8JcP5jTxg2gVK3
DmL0UEW9owRzI0xNo20wazAOBgNVHQ8BAf8EBAMCAaYwHQYDVR0IBBYwFAYIKwYB
BQUHAWIGCCsGAQUFBwMBMA8GA1UdEwEB/wQFMAMBAf8wKQYDVR0OBCIEIYNHI7yR
gXv24T4njyONz9uCHgiL1f8t6lqdzBeqieGtMAoGCCqGSM49BAMCA0gAMEUCIQCb
rxr+Mn+lq59QPPIAD7JNvk58Ldy7z8L6FGW2wyYdnglgl32YycgyYv77LIHb9MV/
MNzocWvyQ7c6JaQ1kN5oovs=
-----END CERTIFICATE-----

2019-12-29 21:49:47.799 UTC [msp.identity] newIdentity -> DEBU 032 Creating identity instance for cert -----BEGIN CERTIFICATE-----

MIICKDCCAc6gAwlBAglRAJk8lJLKIFe0/oohn/WkXZkwCgYIKoZlZj0EAWlwcTEL
MAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEwFjAUBGNVBAcTDFVHbiBG
cmFuY2lzY28xGDAWBgNVBAoTD29yZzluaHVybGV5LmxhYjE5bGMBkGA1UEAxMSY2Eu
b3JnMi5odXJsZXkubGFIMB4XDTE5MTlyOTI5NDQwMFoXDTI5MTlyNjlxNDQwMFow
azELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEwFjAUBGNVBAcTDFVH
biBGcmFuY2lzY28xGDAWBgNVBAoTD29yZzluaHVybGV5LmxhYjE5bGMBkGA1UEAxMSY2Eu
b3JnMi5odXJsZXkubGFIMFkwEwYHKOZlZj0CAQYIKoZlZj0DAQcDQgAESaZtI3Pntosm
wjd+l6nBk9dUhYC2cJJYv2JvivyT1YqE4xOVCDM9SxuGFLnbtFul/JI9EC0oso+pJ
lrSawyCDvKNNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAg1geXvJGBE/bhPiePI43P24leClvV/y3qWp3MF6qJ4a0wCgYIKoZlZj0E
AwIDSAAwRQlhAJc3KXwJZtsUHYwy42SBXZKsrvDOXPhb7LHY1faRaWLKAiBZ7jfs
2boT1MKNKRpeAaUnm7iNevzvgMZy+pMxBexohw==
-----END CERTIFICATE-----

2019-12-29 21:49:47.808 UTC [msp.identity] newIdentity -> DEBU 033 Creating identity instance for cert -----BEGIN CERTIFICATE-----

MIICKDCCAc6gAwlBAglRAJk8lJLKIFe0/oohn/WkXZkwCgYIKoZlZj0EAWlwcTEL
MAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEwFjAUBGNVBAcTDFVHbiBG
cmFuY2lzY28xGDAWBgNVBAoTD29yZzluaHVybGV5LmxhYjE5bGMBkGA1UEAxMSY2Eu
b3JnMi5odXJsZXkubGFIMB4XDTE5MTlyOTI5NDQwMFoXDTI5MTlyNjlxNDQwMFow
azELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEwFjAUBGNVBAcTDFVH
biBGcmFuY2lzY28xGDAWBgNVBAoTD29yZzluaHVybGV5LmxhYjE5bGMBkGA1UEAxMSY2Eu
b3JnMi5odXJsZXkubGFIMFkwEwYHKOZlZj0CAQYIKoZlZj0DAQcDQgAESaZtI3Pntosm
wjd+l6nBk9dUhYC2cJJYv2JvivyT1YqE4xOVCDM9SxuGFLnbtFul/JI9EC0oso+pJ
lrSawyCDvKNNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAg1geXvJGBE/bhPiePI43P24leClvV/y3qWp3MF6qJ4a0wCgYIKoZlZj0E
AwIDSAAwRQlhAJc3KXwJZtsUHYwy42SBXZKsrvDOXPhb7LHY1faRaWLKAiBZ7jfs
2boT1MKNKRpeAaUnm7iNevzvgMZy+pMxBexohw==
-----END CERTIFICATE-----

2019-12-29 21:49:47.808 UTC [bccsp_sw] loadPrivateKey -> DEBU 034 Loading private key [0407e312f42a9d2e4047c347d4326564ef99c4f7ab5b30c7a5469e7e1443c66d] at [etc/hyperledger/msp/users/Admin@org2.hurley.lab/msp/keystore/0407e312f42a9d2e4047c347d4326564ef99c4f7ab5b30c7a5469e7e1443c66d_sk]...

2019-12-29 21:49:47.809 UTC [msp.identity] newIdentity -> DEBU 035 Creating identity instance for cert -----BEGIN CERTIFICATE-----

MIICKDCCAc6gAwlBAglRAJk8lJLKIFe0/oohn/WkXZkwCgYIKoZlZj0EAWlwcTEL
MAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEwFjAUBGNVBAcTDFVHbiBG
cmFuY2lzY28xGDAWBgNVBAoTD29yZzluaHVybGV5LmxhYjE5bGMBkGA1UEAxMSY2Eu
b3JnMi5odXJsZXkubGFIMB4XDTE5MTlyOTI5NDQwMFoXDTI5MTlyNjlxNDQwMFow
azELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWEwFjAUBGNVBAcTDFVH
biBGcmFuY2lzY28xGDAWBgNVBAoTD29yZzluaHVybGV5LmxhYjE5bGMBkGA1UEAxMSY2Eu
b3JnMi5odXJsZXkubGFIMFkwEwYHKOZlZj0CAQYIKoZlZj0DAQcDQgAESaZtI3Pntosm
wjd+l6nBk9dUhYC2cJJYv2JvivyT1YqE4xOVCDM9SxuGFLnbtFul/JI9EC0oso+pJ
lrSawyCDvKNNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB/wQCMAAwKwYDVR0j
BCQwloAg1geXvJGBE/bhPiePI43P24leClvV/y3qWp3MF6qJ4a0wCgYIKoZlZj0E
AwIDSAAwRQlhAJc3KXwJZtsUHYwy42SBXZKsrvDOXPhb7LHY1faRaWLKAiBZ7jfs
2boT1MKNKRpeAaUnm7iNevzvgMZy+pMxBexohw==
-----END CERTIFICATE-----

```

2019-12-29 21:49:47.809 UTC [msp] setupSigningIdentity -> DEBU 036 Signing identity expires
at 2029-12-26 21:44:00 +0000 UTC
2019-12-29 21:49:47.809 UTC [msp] Validate -> DEBU 037 MSP org2MSP validating identity
2019-12-29 21:49:47.810 UTC [main] SetOrdererEnv -> WARN 038 CORE_LOGGING_LEVEL is
no longer supported, please use the FABRIC_LOGGING_SPEC environment variable
2019-12-29 21:49:47.810 UTC [msp] GetDefaultSigningIdentity -> DEBU 039 Obtaining default
signing identity
2019-12-29 21:49:47.810 UTC [grpc] DialContext -> DEBU 03a parsed scheme: ""
2019-12-29 21:49:47.810 UTC [grpc] DialContext -> DEBU 03b scheme "" not registered, fallback
to default scheme
2019-12-29 21:49:47.810 UTC [grpc] watcher -> DEBU 03c ccResolverWrapper: sending new
addresses to cc: [{orderer.hurley.lab:7050 0 <nil>}]
2019-12-29 21:49:47.810 UTC [grpc] switchBalancer -> DEBU 03d ClientConn switching balancer
to "pick_first"
2019-12-29 21:49:47.810 UTC [grpc] HandleSubConnStateChange -> DEBU 03e
pickfirstBalancer: HandleSubConnStateChange: 0xc0000836d0, CONNECTING
2019-12-29 21:49:47.811 UTC [grpc] HandleSubConnStateChange -> DEBU 03f
pickfirstBalancer: HandleSubConnStateChange: 0xc0000836d0, READY
2019-12-29 21:49:47.812 UTC [channelCmd] InitCmdFactory -> INFO 040 Endorser and orderer
connections initialized
2019-12-29 21:49:47.812 UTC [msp] GetDefaultSigningIdentity -> DEBU 041 Obtaining default
signing identity
2019-12-29 21:49:47.812 UTC [msp] GetDefaultSigningIdentity -> DEBU 042 Obtaining default
signing identity
2019-12-29 21:49:47.812 UTC [msp.identity] Sign -> DEBU 043 Sign: plaintext:
0AB6060A076F7267324D535012AA062D...2A0641646D696E732A0641646D696E73
2019-12-29 21:49:47.812 UTC [msp.identity] Sign -> DEBU 044 Sign: digest:
7CA40F2524B9D4F99C3C37231619F6D2835EA0A11B6C10CA10853914795DAE87
2019-12-29 21:49:47.812 UTC [msp] GetDefaultSigningIdentity -> DEBU 045 Obtaining default
signing identity
2019-12-29 21:49:47.812 UTC [msp] GetDefaultSigningIdentity -> DEBU 046 Obtaining default
signing identity
2019-12-29 21:49:47.812 UTC [msp.identity] Sign -> DEBU 047 Sign: plaintext:
0AE7060A0F08021A0608FBBEA4F00522...601437AA58991B7BE69395692B9EF006
2019-12-29 21:49:47.812 UTC [msp.identity] Sign -> DEBU 048 Sign: digest:
C90A6CDBBC34AE728E9EE4B734EF44D08E7B105DD32CE519A01804D1A29C7078
2019-12-29 21:49:47.812 UTC [grpc] DialContext -> DEBU 049 parsed scheme: ""
2019-12-29 21:49:47.813 UTC [grpc] DialContext -> DEBU 04a scheme "" not registered, fallback
to default scheme
2019-12-29 21:49:47.813 UTC [grpc] watcher -> DEBU 04b ccResolverWrapper: sending new
addresses to cc: [{orderer.hurley.lab:7050 0 <nil>}]
2019-12-29 21:49:47.813 UTC [grpc] switchBalancer -> DEBU 04c ClientConn switching balancer
to "pick_first"
2019-12-29 21:49:47.813 UTC [grpc] HandleSubConnStateChange -> DEBU 04d
pickfirstBalancer: HandleSubConnStateChange: 0xc000157880, CONNECTING
2019-12-29 21:49:47.814 UTC [grpc] HandleSubConnStateChange -> DEBU 04e
pickfirstBalancer: HandleSubConnStateChange: 0xc000157880, READY
2019-12-29 21:49:47.824 UTC [channelCmd] update -> INFO 04f Successfully submitted channel
update
Registering admin for org1
Registering admin for org2
Registering user1 for org1
Registering user1 for org2
[hurley] - Ran network restart script
[hurley] - ***** Success!
[hurley] - Complete network deployed at /home/manelmdiaz/hyperledger-fabric-network
[hurley] - Setup:
  - Channels deployed: 1
    * ch1
  - Organizations: 2
    * org1:
      - channels:
        * ch1
      - users:
        * admin
        * user1

```

```
* org2:
  - channels:
    * ch1
  - users:
    * admin
    * user1
```

[hurley] - You can find the network topology (ports, names) here: /home/manelmdiaz/hyperledger-fabric-network/docker-compose.yaml

6.7 Resultat de generar certificats necessaris per HLF

```
manelmdiaz@desktop-i7-linux:~/TFM/fabric-sdk-node$ gulp install-and-generate-certs
2019-12-07T21:59:10.405Z - info: [TestUtil]:

*****
*****
*
* Using temp dir: /home/manelmdiaz/TFM/fabric-sdk-node/test/temp
*
*****
*****

#####
# debug log: /home/manelmdiaz/TFM/fabric-sdk-node/test/temp/debug.log
#####

[22:59:10] Using gulpfile ~/TFM/fabric-sdk-node/gulpfile.js
[22:59:10] Starting 'install-and-generate-certs'...
[22:59:10] Starting 'get-crypto-binaries-amd64'...
--2019-12-07 22:59:10--
https://nexus.hyperledger.org/content/repositories/releases/org/hyperledger/fabric/hyperledger-
fabric/linux-amd64-1.4.0/hyperledger-fabric-linux-amd64-1.4.0.tar.gz
Resolving nexus.hyperledger.org (nexus.hyperledger.org)... 199.19.212.106
Connecting to nexus.hyperledger.org (nexus.hyperledger.org)|199.19.212.106|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 53389792 (51M) [application/x-gzip]
Saving to: '/tmp/fabric-binaries/hyperledger-fabric-linux-amd64-1.4.0.tar.gz'

hyperledger-fabric-linux-amd64- 100%
[=====>] 50,92M 2,04MB/s
in 41s

2019-12-07 22:59:52 (1,24 MB/s) - '/tmp/fabric-binaries/hyperledger-fabric-linux-amd64-
1.4.0.tar.gz' saved [53389792/53389792]

bin/
bin/get-docker-images.sh
bin/orderer
bin/peer
bin/configtxlator
bin/discover
bin/idemixgen
bin/cryptogen
bin/configtxgen
config/
config/orderer.yaml
config/core.yaml
config/configtx.yaml
[22:59:53] Finished 'get-crypto-binaries-amd64' after 43 s
[22:59:53] Starting 'generate-test-certs'...
```

```

Crypto-gen scripts running based on binaries location /tmp/fabric-binaries/bin
Creating new crypto material and tx blocks from within directory /home/manelmdiaz/TFM/fabric-
sdk-node/test/fixtures/crypto-material/config-base
Generating base crypto-material and channel tx files....
org1.example.com
org2.example.com
2019-12-07 23:00:06.334 CET [common.tools.configtxgen] main -> WARN 001 Omitting the
channel ID for configtxgen for output operations is deprecated. Explicitly passing the channel ID
will be required in the future, defaulting to 'testchainid'.
2019-12-07 23:00:06.334 CET [common.tools.configtxgen] main -> INFO 002 Loading
configuration
2019-12-07 23:00:06.390 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:06.390 CET [common.tools.configtxgen.localconfig] Load -> INFO 004 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-base/
configtx.yaml
2019-12-07 23:00:06.399 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 005 orderer type: solo
2019-12-07 23:00:06.399 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
006 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-base/configtx.yaml
2019-12-07 23:00:06.428 CET [common.tools.configtxgen] doOutputBlock -> INFO 007
Generating genesis block
2019-12-07 23:00:06.428 CET [common.tools.configtxgen] doOutputBlock -> INFO 008 Writing
genesis block
2019-12-07 23:00:06.548 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:06.557 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-base/
configtx.yaml
2019-12-07 23:00:06.568 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:06.568 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-base/configtx.yaml
2019-12-07 23:00:06.568 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:06.574 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:06.681 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:06.690 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-base/
configtx.yaml
2019-12-07 23:00:06.699 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:06.699 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-base/configtx.yaml
2019-12-07 23:00:06.699 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:06.702 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:06.816 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:06.825 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-base/
configtx.yaml
2019-12-07 23:00:06.834 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:06.834 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-base/configtx.yaml
2019-12-07 23:00:06.834 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:06.839 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx

```

```

2019-12-07 23:00:06.943 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:06.952 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-base/
configtx.yaml
2019-12-07 23:00:06.961 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:06.961 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-base/configtx.yaml
2019-12-07 23:00:06.961 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:06.965 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:07.070 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:07.079 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-base/
configtx.yaml
2019-12-07 23:00:07.090 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:07.090 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-base/configtx.yaml
2019-12-07 23:00:07.090 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:07.094 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:07.196 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:07.205 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-base/
configtx.yaml
2019-12-07 23:00:07.214 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:07.214 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-base/configtx.yaml
2019-12-07 23:00:07.214 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:07.218 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
Generating crypto-material complete, now renaming keys...
Renaming keys complete

Creating new channel update tx blocks from within directory /home/manelmdiaz/TFM/fabric-sdk-
node/test/fixtures/crypto-material/config-update
2019-12-07 23:00:07.894 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:07.906 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-update/
configtx.yaml
2019-12-07 23:00:07.917 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:07.917 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-update/configtx.yaml
2019-12-07 23:00:07.917 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
005 Generating anchor peer update
2019-12-07 23:00:07.917 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
006 Writing anchor peer update
2019-12-07 23:00:08.026 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:08.035 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/config-update/
configtx.yaml

```

```

2019-12-07 23:00:08.046 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:08.046 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/crypto-material/
config-update/configtx.yaml
2019-12-07 23:00:08.046 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
005 Generating anchor peer update
2019-12-07 23:00:08.046 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
006 Writing anchor peer update

Crypto-gen scripts running based on binaries location /tmp/fabric-binaries/bin
Creating new crypto material and tx blocks from within directory /home/manelmdiaz/TFM/fabric-
sdk-node/test/ts-fixtures/crypto-material/config-base
Generating base crypto-material and channel tx files....
org1.example.com
org2.example.com
2019-12-07 23:00:20.562 CET [common.tools.configtxgen] main -> WARN 001 Omitting the
channel ID for configtxgen for output operations is deprecated. Explicitly passing the channel ID
will be required in the future, defaulting to 'testchainid'.
2019-12-07 23:00:20.562 CET [common.tools.configtxgen] main -> INFO 002 Loading
configuration
2019-12-07 23:00:20.574 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:20.575 CET [common.tools.configtxgen.localconfig] Load -> INFO 004 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
base/configtx.yaml
2019-12-07 23:00:20.584 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 005 orderer type: solo
2019-12-07 23:00:20.584 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
006 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-base/configtx.yaml
2019-12-07 23:00:20.610 CET [common.tools.configtxgen] doOutputBlock -> INFO 007
Generating genesis block
2019-12-07 23:00:20.610 CET [common.tools.configtxgen] doOutputBlock -> INFO 008 Writing
genesis block
2019-12-07 23:00:20.718 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:20.727 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
base/configtx.yaml
2019-12-07 23:00:20.736 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:20.736 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-base/configtx.yaml
2019-12-07 23:00:20.736 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:20.743 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:20.851 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:20.859 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
base/configtx.yaml
2019-12-07 23:00:20.869 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:20.869 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-base/configtx.yaml
2019-12-07 23:00:20.869 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:20.873 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:20.998 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration

```



```

2019-12-07 23:00:21.007 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
base/configtx.yaml
2019-12-07 23:00:21.016 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:21.016 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-base/configtx.yaml
2019-12-07 23:00:21.016 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:21.019 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:21.127 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:21.136 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
base/configtx.yaml
2019-12-07 23:00:21.145 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:21.145 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-base/configtx.yaml
2019-12-07 23:00:21.145 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:21.149 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:21.262 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:21.271 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
base/configtx.yaml
2019-12-07 23:00:21.280 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:21.280 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-base/configtx.yaml
2019-12-07 23:00:21.280 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:21.284 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:21.396 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:21.404 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
base/configtx.yaml
2019-12-07 23:00:21.413 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:21.413 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-base/configtx.yaml
2019-12-07 23:00:21.413 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:21.418 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
Generating crypto-material complete, now renaming keys...
Renaming keys complete

Creating new channel update tx blocks from within directory /home/manelmdiaz/TFM/fabric-sdk-
node/test/ts-fixtures/crypto-material/config-v2
2019-12-07 23:00:22.107 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:22.121 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-v2/
configtx.yaml
2019-12-07 23:00:22.133 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo

```

```

2019-12-07 23:00:22.133 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-v2/configtx.yaml
2019-12-07 23:00:22.133 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:22.137 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:22.251 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:22.260 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-v2/
configtx.yaml
2019-12-07 23:00:22.271 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:22.271 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-v2/configtx.yaml
2019-12-07 23:00:22.271 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
005 Generating new channel configtx
2019-12-07 23:00:22.274 CET [common.tools.configtxgen] doOutputChannelCreateTx -> INFO
006 Writing new channel tx
2019-12-07 23:00:22.383 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:22.393 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-v2/
configtx.yaml
2019-12-07 23:00:22.404 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:22.404 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-v2/configtx.yaml
2019-12-07 23:00:22.404 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
005 Generating anchor peer update
2019-12-07 23:00:22.404 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
006 Writing anchor peer update

Creating new channel update tx blocks from within directory /home/manelmdiaz/TFM/fabric-sdk-
node/test/ts-fixtures/crypto-material/config-update
2019-12-07 23:00:22.529 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:22.543 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
update/configtx.yaml
2019-12-07 23:00:22.556 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:22.556 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-update/configtx.yaml
2019-12-07 23:00:22.556 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
005 Generating anchor peer update
2019-12-07 23:00:22.556 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
006 Writing anchor peer update
2019-12-07 23:00:22.670 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:22.680 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
update/configtx.yaml
2019-12-07 23:00:22.689 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:22.689 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-update/configtx.yaml
2019-12-07 23:00:22.689 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
005 Generating anchor peer update
2019-12-07 23:00:22.689 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
006 Writing anchor peer update

```

```

2019-12-07 23:00:22.794 CET [common.tools.configtxgen] main -> INFO 001 Loading
configuration
2019-12-07 23:00:22.804 CET [common.tools.configtxgen.localconfig] Load -> INFO 002 Loaded
configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-material/config-
update/configtx.yaml
2019-12-07 23:00:22.815 CET [common.tools.configtxgen.localconfig] completeInitialization ->
INFO 003 orderer type: solo
2019-12-07 23:00:22.815 CET [common.tools.configtxgen.localconfig] LoadTopLevel -> INFO
004 Loaded configuration: /home/manelmdiaz/TFM/fabric-sdk-node/test/ts-fixtures/crypto-
material/config-update/configtx.yaml
2019-12-07 23:00:22.815 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
005 Generating anchor peer update
2019-12-07 23:00:22.815 CET [common.tools.configtxgen] doOutputAnchorPeersUpdate -> INFO
006 Writing anchor peer update

Generating fabricca cetificates and keys in directory
/home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/fabricca
Can't load /home/manelmdiaz/.rnd into RNG
140235580486080:error:2406F079:random number generator:RAND_load_file:Cannot open
file:./crypto/rand/randfile.c:88:Filename=/home/manelmdiaz/.rnd
Generating a RSA private key
.....+++++
.....+++++
writing new private key to &apos;/home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/fabricca/
test.key&apos;
-----
Can't load /home/manelmdiaz/.rnd into RNG
139704084668864:error:2406F079:random number generator:RAND_load_file:Cannot open
file:./crypto/rand/randfile.c:88:Filename=/home/manelmdiaz/.rnd
Generating a RSA private key
.....+++++
.....+++++
writing new private key to &apos;/home/manelmdiaz/TFM/fabric-sdk-node/test/fixtures/fabricca/
enroll-key.pem&apos;
-----
Generation complete
[23:00:23] Finished &apos;generate-test-certs&apos; after 30 s
[23:00:23] Finished &apos;install-and-generate-certs&apos; after 1.2 min

```

6.8 Resultats del pla de proves

6.8.1 Resultat de proves a la línia de comandes

```

manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ npm run cc:start -- participant

> convector@0.1.0 cc:start /home/manelmdiaz/wearhealth-poc/convector
> f() { npm run cc:package -- $1; npm run cc:install $1 $2; }; f "participant"

> convector@0.1.0 cc:package /home/manelmdiaz/wearhealth-poc/convector
> f() { npm run lerna:build; chaincode-manager --update --config ./$.1.config.json --output
./chaincode-$1 package; }; f "participant"

> convector@0.1.0 lerna:build /home/manelmdiaz/wearhealth-poc/convector
> lerna run build

lerna notice cli v3.13.4
lerna info Executing command in 1 package: "npm run build"
lerna ERR! npm run build exited 2 in 'participant-cc'
lerna ERR! npm run build stdout:

> participant-cc@0.1.0 build /home/manelmdiaz/wearhealth-poc/convector/packages/participant-
cc

```

```

> npm run clean && tsc

> participant-cc@0.1.0 clean /home/manelmdiaz/wearhealth-poc/convector/packages/participant-cc
> rimraf dist client

../../node_modules/@types/chai-as-promised/index.d.ts(14,43): error TS2694: Namespace 'Chai' has no exported member 'ChaiPlugin'.

lerna ERR! npm run build stderr:
npm ERR! code ELIFECYCLE
npm ERR! errno 2
npm ERR! participant-cc@0.1.0 build: `npm run clean && tsc`
npm ERR! Exit status 2
npm ERR!
npm ERR! Failed at the participant-cc@0.1.0 build script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!   /home/manelmdiaz/.npm/_logs/2020-01-01T20_35_32_108Z-debug.log

lerna ERR! npm run build exited 2 in 'participant-cc'
npm ERR! code ELIFECYCLE
npm ERR! errno 2
npm ERR! convector@0.1.0 lerna:build: `lerna run build`
npm ERR! Exit status 2
npm ERR!
npm ERR! Failed at the convector@0.1.0 lerna:build script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!   /home/manelmdiaz/.npm/_logs/2020-01-01T20_35_32_171Z-debug.log

> convector@0.1.0 cc:install /home/manelmdiaz/wearhealth-poc/convector
> f() { hurl install ${2:-$1} node -P ./chaincode-$1; }; f "participant"

[hurley]      -      installing      smart      contract      located      at
/home/manelmdiaz/wearhealth-poc/convector/chaincode-participant
Installing Chaincode participant version 1.0 at org1
2020-01-01 21:35:34.365 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using
default escc
2020-01-01 21:35:34.365 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using
default vscc
2020-01-01 21:35:34.417 CET [chaincodeCmd] install -> INFO 003 Installed remotely
response:<status:200 payload:"OK" >
Installed Chaincode participant version 1.0 at org1
Installing Chaincode participant version 1.0 at org2
2020-01-01 21:35:34.488 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using
default escc
2020-01-01 21:35:34.489 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using
default vscc
2020-01-01 21:35:34.528 CET [chaincodeCmd] install -> INFO 003 Installed remotely
response:<status:200 payload:"OK" >
Installed Chaincode participant version 1.0 at org2
Instantiating Chaincode at org1 for channel ch1
It may take a few minutes depending on the chaincode dependencies
2020-01-01 21:35:44.595 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 001 Using
default escc
2020-01-01 21:35:44.595 CET [chaincodeCmd] checkChaincodeCmdParams -> INFO 002 Using
default vscc
Instantiated Chaincode at org1

```

6.8.2 Resultat de proves amb fitxer

```
manelmdiaz@desktop-i7-linux:~/wearhealth-poc/convector$ npm test --
> convector@0.1.0 test /home/manelmdiaz/wearhealth-poc/convector
> npm-run-all -s lerna:test

> convector@0.1.0 lerna:test /home/manelmdiaz/wearhealth-poc/convector
> lerna run test --stream

lerna notice cli v3.13.4
lerna info Executing command in 1 package: "npm run test"
participant-cc: > participant-cc@0.1.0 test
/home/manelmdiaz/wearhealth-poc/convector/packages/participant-cc
participant-cc: > mocha -r ts-node/register tests/*.spec.ts --reporter spec
participant-cc: Participant
participant-cc: info: [Chaincode] ===== Instantiated Chaincode chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b7508a70-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args:
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b75126b0-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: initControllers,
[{"version":"**", "controller":"ParticipantController", "name":"/home/manelmdiaz/wearhealth-poc/
convector/packages/participant-cc"}]
participant-cc: debug: [Chaincode] ===== START : initControllers =====
participant-cc: Found config in param [ { version: '**',
participant-cc:   controller: 'ParticipantController',
participant-cc:   name:
participant-cc:     '/home/manelmdiaz/wearhealth-poc/convector/packages/participant-cc' } ]
participant-cc: debug: [Chaincode] ===== END : initControllers =====
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b754d030-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: participant_register,Participant1,Participant1Name
participant-cc: debug: [Chaincode] ===== START : participant_register =====
participant-cc:
{"id":"Participant1","type":"io.worldsibu.participant","name":"Participant1Name","msp":"dummymsp
Id","identities":
[{"fingerprint":"B6:0B:37:7C:DF:D2:7A:08:0B:98:BF:52:A4:2C:DC:4E:CC:70:91:E1","status":true}}]
participant-cc: debug: [Chaincode] ===== END : participant_register =====
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b757b660-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: participant_getParticipantById,Participant1
participant-cc: debug: [Chaincode] ===== START : participant_getParticipantById
=====
participant-cc: debug: [Chaincode] ===== END : participant_getParticipantById
=====
participant-cc:   ✓ hauria de registrar un participant
participant-cc: (node:13814) MaxListenersExceededWarning: Possible EventEmitter memory leak
detected. 11 uncaughtException listeners added. Use emitter.setMaxListeners() to increase limit
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b7598b20-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: participant_getParticipantById,Participant1
participant-cc: debug: [Chaincode] ===== START : participant_getParticipantById
=====
participant-cc: debug: [Chaincode] ===== END : participant_getParticipantById
=====
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b75a4e70-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: participant_getParticipantById,Participant1
participant-cc: debug: [Chaincode] ===== START : participant_getParticipantById
=====
participant-cc: debug: [Chaincode] ===== END : participant_getParticipantById
=====
participant-cc:   ✓ hauria de llegir dades d'un participant registrat
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
```

```

participant-cc: info: [Chaincode] Transaction ID: b75b86f0-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: participant_register,Participant1,Participant1Name
participant-cc: debug: [Chaincode] ===== START : participant_register =====
participant-cc: error: [Chaincode]
participant-cc: { message:
participant-cc:   'Identity exists already, please call changelidentity fn for updates',
participant-cc:   stack:
participant-cc:     'Error: Identity exists already, please call changelidentity fn for updates\n    at
ParticipantController.<anonymous>
(/home/manelmdiaz/wearhealth-poc/convector/packages/participant-cc/src/
participant.controller.ts:41:13)\n
(/home/manelmdiaz/wearhealth-poc/convector/node_modules/tslib/tslib.js:136:27)\n    step
(/home/manelmdiaz/wearhealth-poc/convector/node_modules/tslib/tslib.js:117:57)\n    at
Object.next (/home/manelmdiaz/wearhealth-poc/convector/node_modules/tslib/tslib.js:107:62)' }
participant-cc:   ✓ no hauria de registrar un participant si existeix el id
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b75dd0e0-2cd0-11ea-b652-958020a09749
participant-cc:   info: [Chaincode] Args:
participant_changeldentity,Participant1,56:74:69:D7:D7:C5:A4:A4:C5:2D:4B:7B:7B:27:A9:6A:A8:
6A:C9:26:FF:8B:82
participant-cc: debug: [Chaincode] ===== START : participant_changeldentity
=====
participant-cc: ClientIdentity {
participant-cc:   stub:
participant-cc:     ChaincodeMockStub {
participant-cc:       name: 'Participant',
participant-cc:       cc:
participant-cc:         Chaincode {
participant-cc:           logger: [EventEmitter],
participant-cc:           initialized: true,
participant-cc:           participant: ParticipantController {},
participant-cc:           participant_register: [Function],
participant-cc:           participant_getParticipantById: [Function],
participant-cc:           participant_changeldentity: [Function],
participant-cc:           participant_get: [Function] },
participant-cc:       usercert:
participant-cc:         '-----BEGIN CERTIFICATE-----
MIIB6TCCAY+gAwIBAgIUHkmY6fRP0ANTvzaBwKCKMZZPUnUwCgYIKoZlZjOEAwIwGzEZMBc
GA1UEAxMQZmFicmljLWNhLXNlcnZlcjAeFw0xNzA5MDgwMzQyMDBaFw0xODA5MDgwMzQy
MDBaMB4xHDAaBgNVBAMTE015VGZzdFVzZXJXaXRoQXR0cnMwWTATBgqhkJOPQIBBggq
hkjOPQMBBwNCAATmB1r3CdWVvOOP3opB3DjJnW3CnN8q1ydiRdzmuA6A2rXKzPltHvYbbSqI
SZJubsy8gVL6GYgYXNdu69RzzFF5o4GtMIGqMA4GA1UdDwEB/
wQEAwICBDAwBgNVHRMBAf8EAjAAMB0GA1UdDgQWBbTYKLTAvJJK08OMVGwIhMjMQp2Dr
jAfBgNVHSMEGDAWgBTEs/
52DeLePPx1+65VhgTwu3/2ATAiBgNVHREEGzAZghdBbmlscy1NYWNCb29rLVByby5sb2NhbD
AmBggqAwQFBGclAQQAeyJhdHRycyl6eyJhdHRyMSI6InZhbDEifX0wCgYIKoZlZjOEAwIDSAAwR
QlhAPuEqWUpsvTTvBqLR5JeQSctJuz3zaqGRqSs2iW+QB3FAiAIP0mGwKcgSGRMMBvaqLyt
BYo9v3hRt1r8j8vN0pMcg=-----END CERTIFICATE-----',
participant-cc:   txID: 'b75dd0e0-2cd0-11ea-b652-958020a09749',
participant-cc:   state:
participant-cc:     Map {
participant-cc:       Chaincode:
participant-cc:         <Buffer 5b 7b 22 76 65 72 73 69 6f 6e 22 3a 22 2a 22 2c 22 63 6f 6e 74 72
6f 6c 6c 65 72 22 3a 22 50 61 72 74 69 63 69 70 61 6e 74 43 6f 6e 74 72 6f 6c 6c 65 ... >,
participant-cc:       Participant1:
participant-cc:         <Buffer 7b 22 69 64 22 3a 22 50 61 72 74 69 63 69 70 61 6e 74 31 22 2c 22
74 79 70 65 22 3a 22 69 6f 2e 77 6f 72 6c 64 73 69 62 75 2e 70 61 72 74 69 63 69 70 ... >,
participant-cc:       transientMap: Map {},
participant-cc:       privateCollections: Map {},
participant-cc:       event: Map {},
participant-cc:       history: Map { Chaincode: [Array], Participant1: [Array] },
participant-cc:       invokables: Map {},
participant-cc:       mspld: 'dummymspld',
participant-cc:       stateValidation: Map {},
participant-cc:       privatedataValidation: Map {},
participant-cc:       logger:
participant-cc:         EventEmitter {

```

```

participant-cc:   _events: [Object: null prototype] {},
participant-cc:   _eventsCount: 0,
participant-cc:   _maxListeners: undefined,
participant-cc:   transports: [Object],
participant-cc:   _names: [Array],
participant-cc:   padLevels: false,
participant-cc:   levels: [Object],
participant-cc:   error: [Function],
participant-cc:   warn: [Function],
participant-cc:   info: [Function],
participant-cc:   verbose: [Function],
participant-cc:   debug: [Function],
participant-cc:   silly: [Function],
participant-cc:   id: null,
participant-cc:   level: 'info',
participant-cc:   emitErrs: false,
participant-cc:   stripColors: false,
participant-cc:   exitOnError: false,
participant-cc:   exceptionHandlers: {},
participant-cc:   profilers: {},
participant-cc:   rewriters: [],
participant-cc:   filters: [] },
participant-cc:   args:
participant-cc:   [ 'Participant1',
participant-cc:   '56:74:69:D7:D7:C5:A4:A4:C5:2D:4B:7B:7B:27:A9:6A:A8:6A:C9:26:FF:8B:82' ],
participant-cc:   signedProposal: {},
participant-cc:   txTimestamp:
participant-cc:   { wrappers_: null,
participant-cc:     messageId_: undefined,
participant-cc:     arrayIndexOffset_: -1,
participant-cc:     array: [],
participant-cc:     pivot_: 1.7976931348623157e+308,
participant-cc:     convertedPrimitiveFields_: {} },
participant-cc:   mspld: 'dummymspld',
participant-cc:   cert:
participant-cc:   { version: 2,
participant-cc:     subject: { commonName: 'MyTestUserWithAttrs' },
participant-cc:     issuer: { commonName: 'fabric-ca-server' },
participant-cc:     serial: '1E4998E9F44FD00353BF3681C0A0A431964F5275',
participant-cc:     notBefore: 2017-09-08T03:42:00.000Z,
participant-cc:     notAfter: 2018-09-08T03:42:00.000Z,
participant-cc:     subjectHash: '9cd7895b',
participant-cc:     signatureAlgorithm: 'ecdsa-with-SHA256',
participant-cc:     fingerprint:
participant-cc:     'B6:0B:37:7C:DF:D2:7A:08:0B:98:BF:52:A4:2C:DC:4E:CC:70:91:E1',
participant-cc:     publicKey: { algorithm: 'ecdsa-with-SHA256' },
participant-cc:     altNames: [ 'Anils-MacBook-Pro.local' ],
participant-cc:     extensions:
participant-cc:     { keyUsage: 'Certificate Sign',
participant-cc:       basicConstraints: 'CA:FALSE',
participant-cc:       subjectKeyIdIdentifier:
participant-cc:       'D8:28:B4:C0:BC:92:4A:D3:C3:8C:54:6C:08:86:33:10:A6:8D:83:AE',
participant-cc:       authorityKeyIdIdentifier:
participant-cc:       'keyid:C4:B3:FE:76:0D:E2:DE:3C:FC:75:FB:AE:55:86:04:F0:BB:7F:F6:01',
participant-cc:       subjectAlternativeName: 'DNS:Anils-MacBook-Pro.local',
participant-cc:       '1.2.3.4.5.6.7.8.1: {"attrs":{"attr1":"val1"}}' },
participant-cc:     attrs: { attr1: 'val1' },
participant-cc:     id: 'x509:/CN=MyTestUserWithAttrs:/CN=fabric-ca-server' }
participant-cc:   null
participant-cc:   Existing participant:
participant-cc:   Participant {
participant-cc:     _id: 'Participant1',
participant-cc:     _type: 'io.worldsibu.participant',
participant-cc:     _name: 'Participant1Name',
participant-cc:     _msp: 'dummymspld',
participant-cc:     _identities:

```

```

participant-cc: [ { fingerprint:
participant-cc:   'B6:0B:37:7C:DF:D2:7A:08:0B:98:BF:52:A4:2C:DC:4E:CC:70:91:E1',
participant-cc:   status: true } ] }
participant-cc: existing.msp=dummymspld requesterMSP=dummymspld
participant-cc: isAdmin=null
participant-cc:   ✓ no hauria de permetre canviar la identitat d'un participant si el participant no
té atribut admin
participant-cc: error: [Chaincode]
participant-cc: { message: 'Unauthorized. Requester identity is not an admin',
participant-cc: stack:
participant-cc:   'Error: Unauthorized. Requester identity is not an admin\n      at
ParticipantController.<anonymous>
(/home/manelmdiaz/wearhealth-poc/convector/packages/participant-cc/src/
participant.controller.ts:84:13)\n      at
step
(/home/manelmdiaz/wearhealth-poc/convector/node_modules/tslib/tslib.js:136:27)\n      at
Object.next (/home/manelmdiaz/wearhealth-poc/convector/node_modules/tslib/tslib.js:117:57)\n
at fulfilled (/home/manelmdiaz/wearhealth-poc/convector/node_modules/tslib/tslib.js:107:62) }
participant-cc: debug: [Chaincode] ===== START : participant_register =====
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b75f5780-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: participant_register,Participant2,Participant2Name
participant-cc:
{"id":"Participant2","type":"io.worldsibu.participant","name":"Participant2Name","msp":"dummymsp
ld","identities":
[{"fingerprint":"B6:0B:37:7C:DF:D2:7A:08:0B:98:BF:52:A4:2C:DC:4E:CC:70:91:E1","status":true}}]
participant-cc: debug: [Chaincode] ===== END : participant_register =====
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b76041e0-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: participant_getParticipantById,Participant2
participant-cc: debug: [Chaincode] ===== START : participant_getParticipantById
=====
participant-cc: debug: [Chaincode] ===== END : participant_getParticipantById
=====
participant-cc: debug: [Chaincode] ===== START : participant_changelidentity
=====
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b761c880-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args:
participant_changelidentity,Participant2,56:74:69:D7:D7:C5:A4:A4:C5:2D:4B:7B:7B:27:A9:6A:A8:
6A:C9:26:FF:8B:82
participant-cc: ClientIdentity {
participant-cc: stub:
participant-cc: ChaincodeMockStub {
participant-cc:   name: 'Participant',
participant-cc:   cc:
participant-cc:     Chaincode {
participant-cc:       logger: [EventEmitter],
participant-cc:       initialized: true,
participant-cc:       participant: ParticipantController {},
participant-cc:       participant_register: [Function],
participant-cc:       participant_getParticipantById: [Function],
participant-cc:       participant_changelidentity: [Function],
participant-cc:       participant_get: [Function] },
participant-cc:   usercert:
participant-cc:     '-----BEGIN CERTIFICATE-----\n
nMIIC7DCCApOgAwIBAgIUc3DffC8hY03iz6zRC6GZQUch7EwCgYIKoZlZj0EAwIwI\
ncTELMaKGA1UEBhMCMVVMxEzARBGNVBAgTCKNhbGlmb3JuaWEwEjFjAUBGNVBAcTDVNH\
nbiBGcmFuY2lzY28xGDAWBgNVBAoTD29yZzEuaHVybGV5LmXhYjEhMBkGA1UEAxMS\
nY2Eub3JnMS5odXJsZXkubGFIMB4XDTE5MDUwNjA4NDUwMjEwMDUwNjA4NDUwMjEw\
nMDFowfzELMAKGA1UEBhMCMVVMxZzAVBgNVBAGTDk5vcnRlENhcm9saW5hMRQwEgYD\
nVQKKEwtleXBicmxlZGdlcjEwMA0GA1UECXMGMGY2xpZW50MAsGA1UECxEb3JnMTAS\
nBgNVBAsTC2RlcGFydG1lbnQxMQ8wDQYDVQQDEWZhZG1pbjIwWTATBgqhkjOPQIB\
nBggqhkiJOPQMBBwNCAATdhgd0fRPq4AYSvS9tiS7vcZamCG3PDAAb0QM4UGyFADdWi\
nRsQjglz2/Mnld4rLkU6srIAJUhdZI+QYGGkDhZIBo4H6MIH3MA4GA1UdDwEB/wQE\
nAwIHgDAMBGNVHRMBAf8EAJAAMB0GA1UdDgQWBBSbHq5DcRCcBt0+y4miDuzLOq80\
n8TArBgNVHSMEJDAigCD9XKUjIbuooHek1fmgbe768dWTkHdGpqGn8v/YEeBbyDAR\
nBgNVHREECjAlggZ1YnVudHUweAYIKgMEBQYHCAEEbHsiYXR0cnMiOnsiYWRtaW4i

```



```

nOiJ0cnVliiwiiaGYuQWZmaWxpYXRpb24iOiJvcmcxLmRlcGFydG1lbnQxliwiiaGYu\
nRW5yb2xsbWVudEEljoiYWRtaW4yliwiiaGYuVHlwZSI6ImNsaWVudCJ9fTAKBggq\
nhkjOPQQDAgNHADBEAiAzUQos0hPVPf3DuZaCW3gX+LxL2G5d7iY1ZUh1murgwlg\
ndkQlssMaMwkireuglUubT/Chee4jFgnhJqffnG+qCHs=\n-----END CERTIFICATE-----\n',
participant-cc: txID: 'b761c880-2cd0-11ea-b652-958020a09749',
participant-cc: state:
participant-cc: Map {
participant-cc: Chaincode:
participant-cc: <Buffer 5b 7b 22 76 65 72 73 69 6f 6e 22 3a 22 2a 22 2c 22 63 6f 6e 74 72
6f 6c 6c 65 72 22 3a 22 50 61 72 74 69 63 69 70 61 6e 74 43 6f 6e 74 72 6f 6c 6c 65 ... >,
participant-cc: Participant1:
participant-cc: <Buffer 7b 22 69 64 22 3a 22 50 61 72 74 69 63 69 70 61 6e 74 31 22 2c 22
74 79 70 65 22 3a 22 69 6f 2e 77 6f 72 6c 64 73 69 62 75 2e 70 61 72 74 69 63 69 70 ... >,
participant-cc: Participant2:
participant-cc: <Buffer 7b 22 69 64 22 3a 22 50 61 72 74 69 63 69 70 61 6e 74 32 22 2c 22
74 79 70 65 22 3a 22 69 6f 2e 77 6f 72 6c 64 73 69 62 75 2e 70 61 72 74 69 63 69 70 ... >,
participant-cc: transientMap: Map {},
participant-cc: privateCollections: Map {},
participant-cc: event: Map {},
participant-cc: history:
participant-cc: Map {
participant-cc: Chaincode: [Array],
participant-cc: Participant1: [Array],
participant-cc: Participant2: [Array] },
participant-cc: invokables: Map {},
participant-cc: mspld: 'dummymspld',
participant-cc: stateValidation: Map {},
participant-cc: privatedataValidation: Map {},
participant-cc: logger:
participant-cc: EventEmitter {
participant-cc: _events: [Object: null prototype] {},
participant-cc: _eventsCount: 0,
participant-cc: _maxListeners: undefined,
participant-cc: transports: [Object],
participant-cc: _names: [Array],
participant-cc: padLevels: false,
participant-cc: levels: [Object],
participant-cc: error: [Function],
participant-cc: warn: [Function],
participant-cc: info: [Function],
participant-cc: verbose: [Function],
participant-cc: debug: [Function],
participant-cc: silly: [Function],
participant-cc: id: null,
participant-cc: level: 'info',
participant-cc: emitErrs: false,
participant-cc: stripColors: false,
participant-cc: exitOnError: false,
participant-cc: exceptionHandlers: {},
participant-cc: profilers: {},
participant-cc: rewriters: [],
participant-cc: filters: [] },
participant-cc: args:
participant-cc: [ 'Participant2',
participant-cc: '56:74:69:D7:D7:C5:A4:A4:C5:2D:4B:7B:7B:27:A9:6A:A8:6A:C9:26:FF:8B:82' ],
participant-cc: signedProposal: {},
participant-cc: txTimestamp:
participant-cc: { wrappers_: null,
participant-cc: messageId_: undefined,
participant-cc: arrayIndexOffset_: -1,
participant-cc: array: [],
participant-cc: pivot_: 1.7976931348623157e+308,
participant-cc: convertedPrimitiveFields_: {} },
participant-cc: mspld: 'dummymspld',
participant-cc: cert:
participant-cc: { version: 2,

```

```

participant-cc: subject:
participant-cc: { countryName: 'US',
participant-cc: stateOrProvinceName: 'North Carolina',
participant-cc: organizationName: 'Hyperledger',
participant-cc: organizationalUnitName: 'department1',
participant-cc: commonName: 'admin2' },
participant-cc: issuer:
participant-cc: { countryName: 'US',
participant-cc: stateOrProvinceName: 'California',
participant-cc: localityName: 'San Francisco',
participant-cc: organizationName: 'org1.hurley.lab',
participant-cc: commonName: 'ca.org1.hurley.lab' },
participant-cc: serial: '720DC37DF0BC858D378B3EB3442E8665051C87B1',
participant-cc: notBefore: 2019-05-06T08:41:00.000Z,
participant-cc: notAfter: 2020-05-05T08:46:00.000Z,
participant-cc: subjectHash: '44324676',
participant-cc: signatureAlgorithm: 'ecdsa-with-SHA256',
participant-cc: fingerprint:
participant-cc: '8D:B9:F2:E7:77:CB:A9:A3:B9:0D:B7:C8:F1:FE:70:16:42:3B:BA:0D',
participant-cc: publicKey: { algorithm: 'ecdsa-with-SHA256' },
participant-cc: altNames: [ 'ubuntu' ],
participant-cc: extensions:
participant-cc: { keyUsage: 'Digital Signature',
participant-cc: basicConstraints: 'CA:FALSE',
participant-cc: subjectKeyIdentifier:
participant-cc: '9B:1E:AE:43:71:10:9C:06:DD:3E:CB:89:A2:0E:EC:CB:3A:AF:34:F1',
participant-cc: authorityKeyIdentifier:
participant-cc:
'keyid:FD:5C:A5:23:21:BB:A8:A0:77:A4:D5:F9:A0:6C:4E:FA:F1:D5:93:90:77:46:A6:A1:A7:F2:FF:
D8:11:E0:5B:C8',
participant-cc: subjectAlternativeName: 'DNS:ubuntu',
participant-cc: '1.2.3.4.5.6.7.8.1':
participant-cc:
{"attrs":
{"admin":"true","hf.Affiliation":"org1.department1","hf.EnrollmentID":"admin2","hf.Type":"client"}} }
},
participant-cc: attrs:
participant-cc: { admin: 'true',
participant-cc: 'hf.Affiliation': 'org1.department1',
participant-cc: 'hf.EnrollmentID': 'admin2',
participant-cc: 'hf.Type': 'client' },
participant-cc: id:
participant-cc: 'x509::/C=US/ST=North
Carolina/O=Hyperledger/OU=client+OU=org1+OU=department1/CN=admin2::/C=US/
ST=California/L=San Francisco/O=org1.hurley.lab/CN=ca.org1.hurley.lab' }
participant-cc: true
participant-cc: Existing participant:
participant-cc: Participant {
participant-cc: _id: 'Participant2',
participant-cc: _type: 'io.worldsibu.participant',
participant-cc: _name: 'Participant2Name',
participant-cc: _msp: 'dummymspld',
participant-cc: _identities:
participant-cc: [ { fingerprint:
participant-cc: 'B6:0B:37:7C:DF:D2:7A:08:0B:98:BF:52:A4:2C:DC:4E:CC:70:91:E1',
participant-cc: status: true } ] }
participant-cc: existing.msp=dummymspld requesterMSP=dummymspld
participant-cc: isAdmin=true
participant-cc: info: [Chaincode] ===== Invoked Chaincode Chaincode =====
participant-cc: info: [Chaincode] Transaction ID: b7634f20-2cd0-11ea-b652-958020a09749
participant-cc: info: [Chaincode] Args: participant_getParticipantById,Participant2
participant-cc: debug: [Chaincode] ===== END : participant_changeldentity
=====
participant-cc: debug: [Chaincode] ===== START : participant_getParticipantById
=====
participant-cc: debug: [Chaincode] ===== END : participant_getParticipantById
=====

```

participant-cc: ✓ Hauria de permetre modificar la identitat d'un participant si el participant té atribut admin
participant-cc: 5 passing (136ms)
lerna success run Ran npm script 'test' in 1 package in 2.3s:
lerna success – participant-cc