# Taller Blockchain

## Diseño y despliegue de una red blockchain (2da parte)

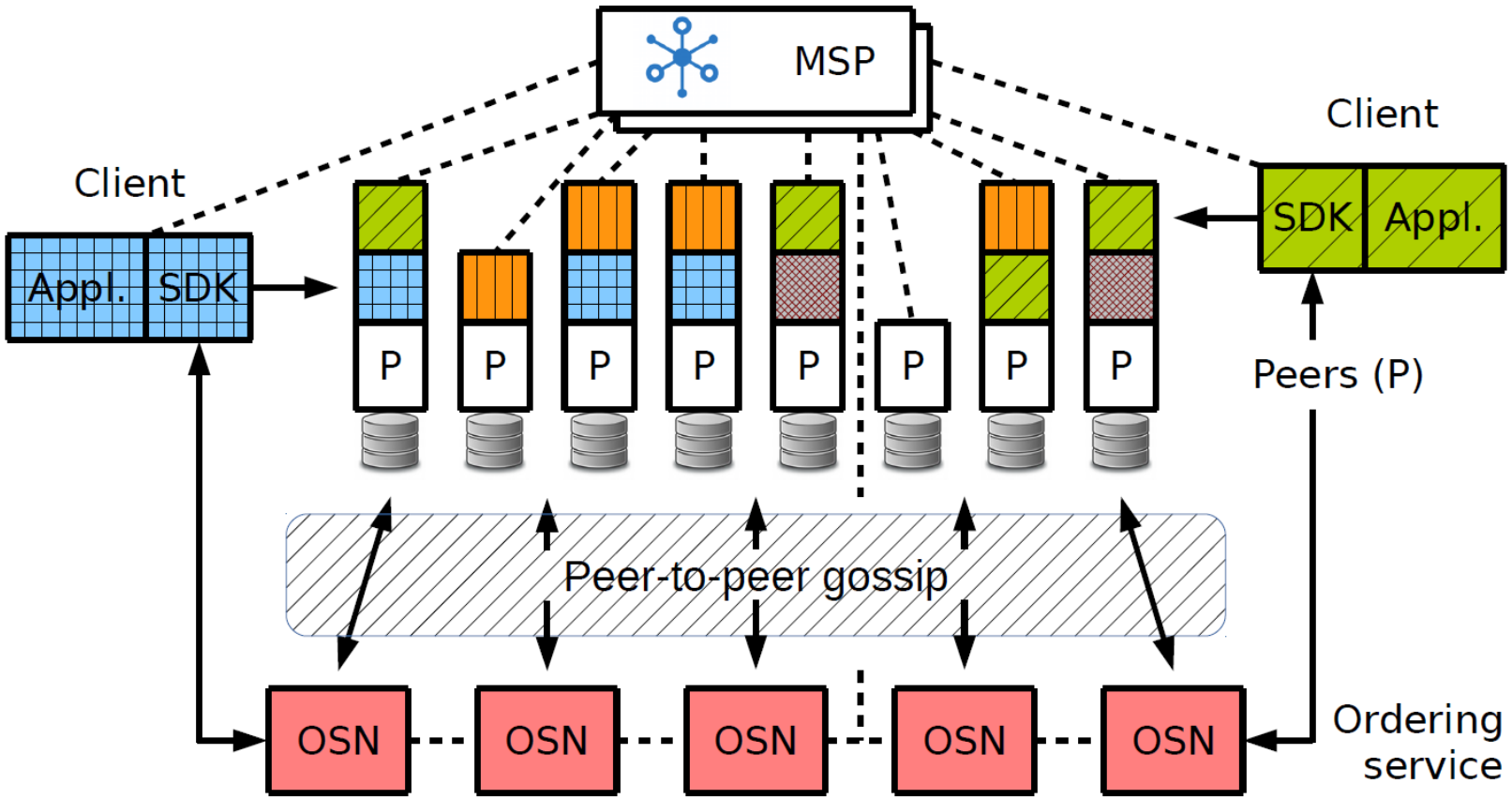
Dr. Iván S. Razo-Zapata
Dr. Alberto F. Martínez

Día 4, Viernes 5 de Marzo, 2021

# Agenda

- Despliegue de red blockchain con scripts automatizados (2da parte)
- Método de diseño
- Conformación de equipos de trabajo
- Herramientas de diseño
- Trabajo en equipos - diseño general
- Presentación
- Trabajo en equipos - arquitectura de solución

# Hyperledger Fabric



# Principales archivos de configuración

- crypto-config.yaml
- configtx.yaml *
- base/docker-compose-base.yaml
- base/peer-base.yaml **
- docker-compose-cli.yaml
- docker-compose-couch.yaml
- docker-compose-ca.yaml

- * Políticas de membresia
- ** Posiblemente nada que modificar

# Principales archivos de ejecución

- byfn.sh
- scripts/script.sh
- scripts/utils.sh
- ccp-generate.sh

# Principales archivos de ejecución - byfn.sh

- networkUp
  - Genera Certs con cryptogen
  - Genera Channel Artifacts con configtxgen
  - docker-compose
- networkDown

```
function networkUp() {

  checkPrereqs
  # Start the Network
  # generate artifacts if they don't exist
  if [ ! -d "crypto-config" ]; then
    generateCerts
    replacePrivateKey
    generateChannelArtifacts
  fi

  COMPOSE_FILES="-f ${COMPOSE_FILE}"
  if [ "${CERTIFICATE_AUTHORITIES}" == "true" ]; then
    COMPOSE_FILES="${COMPOSE_FILES} -f ${COMPOSE_FILE_CA}" #peer0.org1.risks.org #author1.department1.university1.org
    export BYFN_CA1_PRIVATE_KEY=$(cd crypto-config/peerOrganizations/department1.university1.edu/ca && ls *_sk)
    export BYFN_CA2_PRIVATE_KEY=$(cd crypto-config/peerOrganizations/department2.university2.edu/ca && ls *_sk)
    export BYFN_CA3_PRIVATE_KEY=$(cd crypto-config/peerOrganizations/department3.university3.edu/ca && ls *_sk)
  fi
  COMPOSE_FILES="${COMPOSE_FILES} -f ${COMPOSE_FILE_COUCH}"
  echo "############################################################"
  echo "######                    Starting Network                ######"
  echo "############################################################"
  echo "$COMPOSE_FILES"
  # Start the Network
  IMAGE_TAG=$IMAGETAG docker-compose ${COMPOSE_FILES} up -d 2>&1
  docker ps -a
  if [ $? -ne 0 ]; then
    echo "ERROR !!!! Unable to start network"
    exit 1
  fi
```

# Principales archivos de ejecución

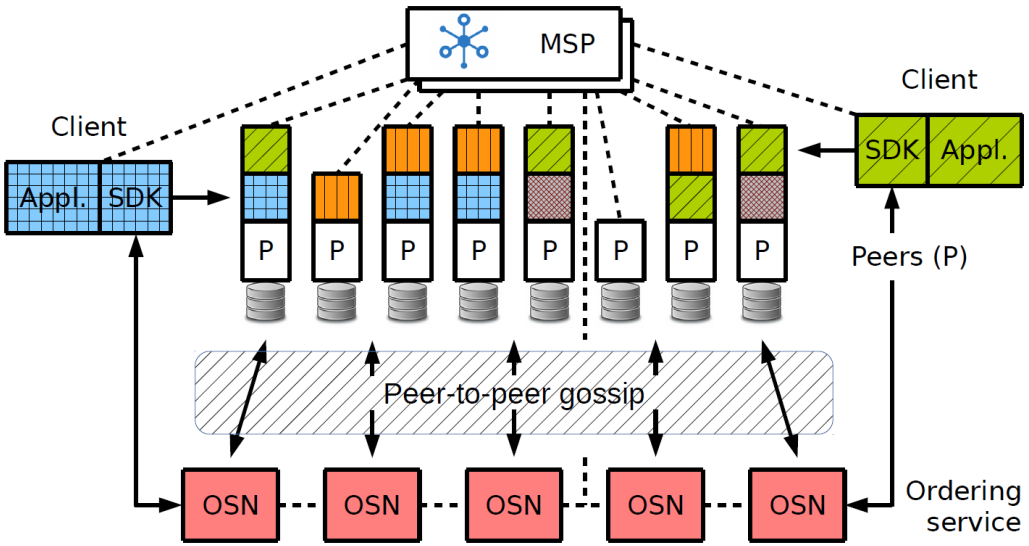
- networkUp
- Resuelve dependencias del chaincode
- Ejecuta script script-s/script.sh
- Ejecución es en **cli**

```
# NEW For v2.0
echo "Valor de NO_CHAINCODE ... "
echo $NO_CHAINCODE

# Getting dependencies for Chaincode
# Careful with location ... needs to be updated
if [ "${NO_CHAINCODE}" != "true" ]; then
  echo Vendoring Go dependencies ...
  pushd ../chaincode/abstore/go
  #GO111MODULE=on
  go mod vendor
  popd
  echo Finished vendoring Go dependencies
fi

# now run the end to end script
docker exec cli scripts/script.sh $CHANNEL_NAME_1 $CHANNEL_NAME_2 $CLI_DELAY $LANGUAGE $CLI_TIMEOUT $VERBOSE $NO_CHAINCODE
if [ $? -ne 0 ]; then
  echo "ERROR !!!! Test failed"
  exit 1
fi
```

# Hyperledger Fabric - MSP




◇ Membership Service Provider (MSP)

◇ Responsable de asociar nodos con sus indentidades criptograficas

◇ Mantiene la naturaleza permisionada de Fabric

Membership service provider. Ayuda a determinar que elemento tiene derechoa a qué en la red (control de acceso y de recursos). Algunos tendrán derecho a:

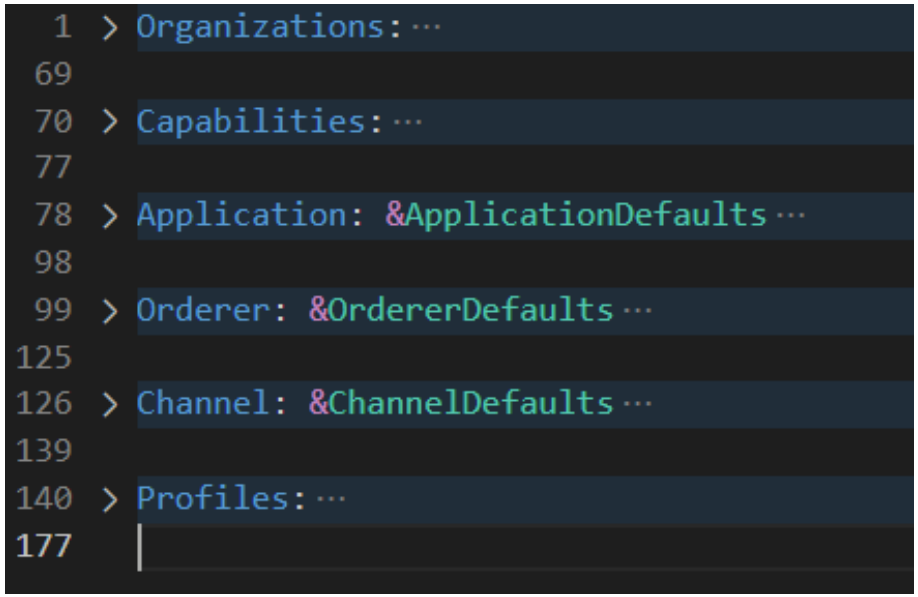
- Leer.
- Escribir.
- Validar transacciones.

## Policies

- Conjunto de reglas de operación e interacción que existen en entre organizaciones
- Relacionada con la red Blockchain, es un conjunto de reglas de operación e interacción que existen en una red Blockchain
- configtx.yaml

## Elementos en configtx.yaml

- Organization
- Capabilities
- Application
- Orderer
- Channel
- Profiles



```
Profiles:

    RedPruebaOrdererGenesis:
        <<: *ChannelDefaults
        Orderer:
            <<: *OrdererDefaults
            Organizations:
                - *OrdererOrg
            Capabilities:
                <<: *OrdererCapabilities
        Consortiums:
            CienciaConsortium:
                Organizations:
                    - *Org1
                    - *Org2
                    - *Org3
    RedPruebaChannel1:
        Consortium: CienciaConsortium
        <<: *ChannelDefaults
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *Org1
                - *Org2
                - *Org3
            Capabilities:
                <<: *ApplicationCapabilities
    RedPruebaChannel2:
        Consortium: CienciaConsortium
        <<: *ChannelDefaults
        Application:
            <<: *ApplicationDefaults
            Organizations:
                - *Org2
                - *Org3
            Capabilities:
                <<: *ApplicationCapabilities
```

# Hyperledger Fabric

```
Organizations:
    - &OrdererOrg
        Name: OrdererOrg
        ID: OrdererMSP
        MSPDir: crypto-config/ordererOrganizations/ciencia.edu/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Writers:
                Type: Signature
                Rule: "OR('OrdererMSP.member')"
            Admins:
                Type: Signature
                Rule: "OR('OrdererMSP.admin')"
    - &Org1
        Name: Org1MSP
        ID: Org1MSP
        MSPDir: crypto-config/peerOrganizations/department1.university1.edu/msp
        Policies:
            Readers:
                Type: Signature
                Rule: "OR('Org1MSP.admin', 'Org1MSP.peer', 'Org1MSP.client')"
            Writers:
                Type: Signature
                Rule: "OR('Org1MSP.admin', 'Org1MSP.client')"
            Admins:
                Type: Signature
                Rule: "OR('Org1MSP.admin')"
        AnchorPeers:
            - Host: peer0.department1.university1.edu
              Port: 5051
```

```
Application: &ApplicationDefaults
    Organizations:
    Policies:
        Readers:
            Type: ImplicitMeta
            Rule: "ANY Readers"
        Writers:
            Type: ImplicitMeta
            Rule: "ANY Writers"
        Admins:
            Type: ImplicitMeta
            Rule: "MAJORITY Admins"
        LifecycleEndorsement:
            Type: Signature
            Rule: "OR('Org1MSP.admin.peer','Org3MSP.peer')"
        Endorsement:
            Type: Signature
            Rule: "OR('Org1MSP.admin.peer','Org3MSP.peer')"
    Capabilities:
        <<: *ApplicationCapabilities
```

# Orden para su modificación

- crypto-config.yaml
- configtx.yaml
- base/docker-compose-base.yaml
- base/peer-base.yaml *
- docker-compose-cli.yaml
- docker-compose-couch.yaml
- docker-compose-ca.yaml

- * Posiblemente nada que modificar

# Orden para su modificación

- Pruebas de operación
- docker-compose-couch.yaml
- docker-compose-cli.yaml

# *i*SIM – An Integrated Design Method for New Service Development

*Prof. Eng Chew*

UNIVERSITY OF TECHNOLOGY SYDNEY

*Iván S. Razo-Zapata*

LUXEMBOURG INSTITUTE OF SCIENCE AND TECHNOLOGY

EE-Challenge 2017

1

Chew, E. K. (2016). i SIM: An integrated design method for commercializing service innovation. Information Systems Frontiers, 18(3), 457-478.

# Conformación de equipos de trabajo

- Opción 1: CIDESI, CIATEQ, UAQ
- Opción 2:
  - Speed dating
  - 3 a 4 equipos

# Presentación por equipos

- Componentes - iSIM
  - Sector
  - Clientes / usuarios
  - Concepto general de la solución (service concept design)
  - Arquitectura (ver Fabric)
  - Actividades a ejecutar (Service Activity System)
  - Beneficios esperados
- Componentes - Fabric
  - Orderer
  - Organizaciones y sus peers
  - Canales
  - Assets

# Miro

https://miro.com/welcomeonboard/
xquuvU9nCMWFIHLtDmW5rbqkKjTWCRfsFectB7DRlejI2AzDvebAtf936k7ib0