coding-bootcamps.com

Hyperledger Fabric for System Admin Vs Developers

Coding Bootcamps

By Sergio Torres
from Coding Bootcamps

Certificaciones Hyperledger Fabric y Tipos de Carreras

Webinar

Resumen

¿Cuánto sabes de
Hyperledger Fabric?

About Presenter

- **Sergio** es ingeniero informático especializado en estudiar el dato tanto en su comienzo con el Big Data para actualmente trabajar con Blockchain.
- Está elaborando el libro Hands-on Smart Contract Development with Hyperledger Fabric de O'Reilly Media junto con otros autores.
- Ha impartido numerosos cursos a lo largo de estos tres años sobre Hyperledger Fabric tanto en universidades como en centros privados.
- Cuenta con dos masters en Big Data, uno especializado en Business Analytics y otro en arquitectura técnica.
- Su comienzo con Blockchain parte desde cero en el conocimiento de la tecnología Hyperledger Fabric.

Esquema

- Hyperledger Fabric puntos fuertes
- HF Componentes y Arquitectura
- HF para Administradores de Sistemas
- HF para Desarrolladores

1. Arquitectura permisionada
2. Altamente modular
3. Consenso seleccionable
4. Model abierto de smart contract
5. Confirmación de baja latencia
6. Configuración de privacidad de datos

7. Multilenguaje de chaincodes: java, nodejs y go
8. Compatible con la EVM y con solidity
9. Diseñado para mejora continua de versiones de sus componentes
10. Administración y versionados de smart contracts
11. Modelo flexible de políticas de endorsement entre las organizaciones de la red
12. Datos consultados vía query

1. Peer
2. Ordering service
3. Fabric CA (Certificate Authority)
4. Fabric ledger
5. Channel
6. Smart Contracts or chaincodes
7. Endorsement policy
8. Membership services provider (MSP)

Peer

- Unidad principal de red HF
- El Peer mantiene una copia exacta de la cadena (ledger)
- Tipos de Peer
  - Anchor Peer
  - Leader Peer
  - Endorsing Peer
  - Committing peer

Peer

- Los nodos pueden conigurarse, crearse, actualizarse o eliminarse.
- Exponen una serie de métodos mediante su API para la administración de estas configuraciones.

Ordering Service

- A veces nombrado como Orderer

- Es el mecanismo con el que las aplicaciones y los nodos interactúan con el resto para garantizar la consistencia de los datos almacenados en los ledgers, es conocido como ***orderer***.

- Orderer Types
  - Solo (deprecated in v2.x)
  - Kafka (deprecated in v2.x)
  - Raft (recommended)

Ordering Service

- Orderer envía los mensajes confirmados mediante broadcast dentro del canal a todos los nodos.Todos los nodos que están escuchando, reciben el mensaje con el mismo orden.

- El servicio de orderer ordena las transacciones por orden de llegada para todos los canales de la red.

- Una vez ordenada la transacción, los registros confirmados de los nodos se agrupan y asignan como parte del bloque para ese canal de comunicación.

Ordering Service- Raft VS Kafka

- **Raft es más fácil de instalar.**
- **Kafka y Zookeeper no están diseñados para funcionar en redes grandes**.
- **Raft es soportado de manera nativa**, lo que significa que los usuarios deben obtener las imágenes necesarias y aprender a usar Kafka y ZooKeeper por su cuenta.

Ordering Service- Raft VS Kafka

- Mientras que Kafka utiliza un grupo de servidores (llamados "intermediarios de Kafka") y el administrador de la organización que controla el orderer especifica cuántos nodos quieren usar en un canal en particular, Raft permite a los usuarios especificar qué nodos de orderer se implementarán en cada canal.

- Raft es el primer paso para la implementación en Fabric de **Byzantine Fault Tolerant** (BFT) ordering service.

Fabric Certificate Authority

- **Fabric CA** es un proveedor de CA Root privado capaz de administrar las identidades digitales de los participantes de Fabric que tienen el formato de certificados X.509.

- Debido a que la CA de Fabric es una CA personalizada dirigida a las necesidades de la CA Root de Fabric, no es inherentemente capaz de proporcionar certificados SSL para uso general/automático en los navegadores.

Fabric Certificate Authority

- Implementación personalizada de la autoridad de certificación
- Alternativa a cryprogen
- Ayuda en
  - Registros de identidad
  - Enrollment

Ledger

- Almacena información verdaderamente importante sobre objetos comerciales

- Ledger se basa en dos partes, world state y blockchain el cual es el log complete de transacciones

- World State tiene dos opciones
  - LevelDB (Default)
  - CouchDB

Ledger

- El ledger es el registro a prueba de manipulaciones de todas las transiciones en Fabric, ya que las transiciones de estado son el resultado de invocaciones de chaincodes ("transacciones") enviadas por las partes participantes.

- Cada transacción es un conjunto **key-value pairs** que están confirmadas en el ledger como creaciones, modificaciones o actualizaciones.

Ledger Features

- Consultar y actualizar el ledger mediante búsquedas basadas en claves, consultas de rango y consultas de claves compuestas
- Consultas de solo lectura con un lenguaje de consulta enriquecido (si se usa CouchDB como base de datos de estado)
- Consultas de historial de solo lectura: consulta el historial del ledger para una clave, lo que permite escenarios de procedencia de datos

Ledger Features

- Las transacciones consisten en las versiones de clave/valor que se leyeron en el chaincode (conjunto de lectura) y los clave/valor que se escribieron en el chaincode (conjunto de escritura)
- Las transacciones contienen firmas de todos los peers que las respaldan y se envían al servicio de orderer.
- Las transacciones se ordenan en bloques y se "entregan" desde un servicio de orederer a los peers en un canal.

Ledger Features

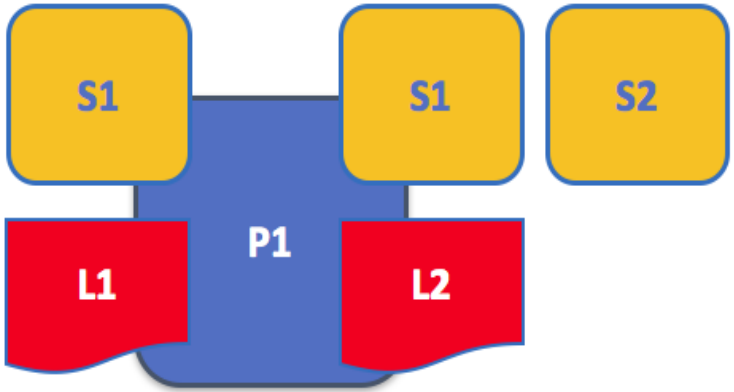
- Los peers validan las transacciones contra las políticas de endorser y hacen que se cumplan.
- Antes de agregar un bloque, se realiza una verificación de versiones para garantizar que los estados de los activos que se leyeron no hayan cambiado desde el tiempo de ejecución del chaincode.
- Existe inmutabilidad una vez que una transacción se valida y confirma.

Ledger Features

- El ledger de un canal contiene un bloque de configuración que define políticas, listas de control de acceso y otra información pertinente.

- Los canales contienen instancias de MSP que permiten que los materiales criptográficos se deriven de diferentes autoridades de certificación.

Multiple Ledgers



Channel

- Mecanismo mediante el cual los peers interesados dentro de una red blockchain pueden colaborar y comunicarse.
- Un canal de Hyperledger Fabric es una “subred” privada de comunicación entre dos o más miembros específicos de la red, con el propósito de realizar transacciones privadas y confidenciales.
- En HF, tenemos canal de sistema y canal de aplicación

Channel Characteristics

- Un canal lo definen los miembros (organizaciones), los anchor peers de cada organización, el ledger, las aplicaciones de los chaincodes y los nodos de servicio de orderer.

- Cada transacción en la red se ejecuta en un canal, donde cada parte debe estar autenticada y autorizada para realizar transacciones en ese canal.

- Cada nodo que se une a un canal tiene su propia identidad proporcionada por un proveedor de servicios de membresía (MSP), que autentica a cada nodo en sus servicios en canal.

Channel Characteristics

- La elección de un nodo líder para cada organización en un canal determina qué nodo se comunica con el servicio de orderer en nombre de la organización.

- **Ningún dato del ledger puede pasar de un canal a otro**. Esta separación de libros de contabilidad, por canal, se define e implementa mediante el chaincode de configuración, el servicio de MSP y el protocolo de difusión de datos.

Smart Contracts

- Smart contracts no son más que acuerdos comerciales encapsulados en el código.

- Smart Contract vs Chaincode
  - Los Smart Contracts están destinados a contener la lógica que se ejecuta e interactúa con el ledger para administrar y mantener el estado del activo.

  - Chaincode rige los aspectos administrativos de los contratos inteligentes, así es como los Smart Contracts se empaquetan en un chaincode y luego se implementan en la red.

Smart Contracts…

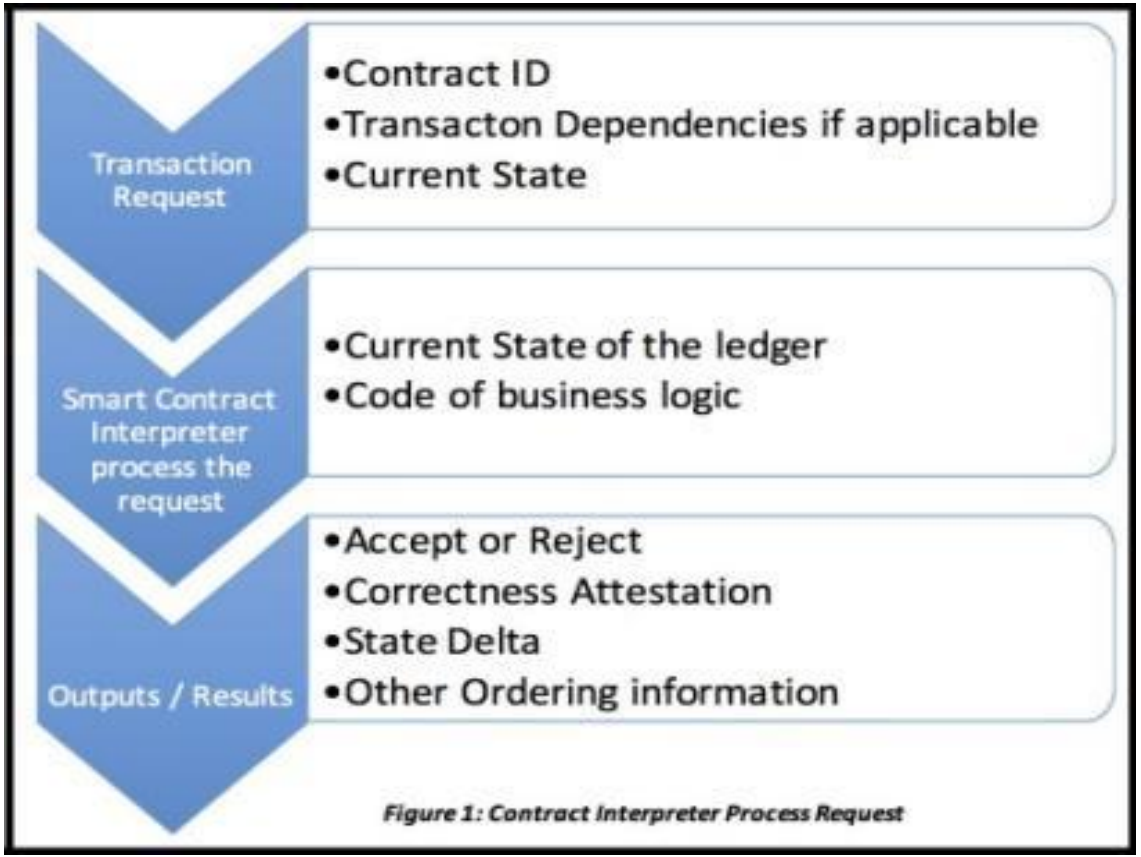
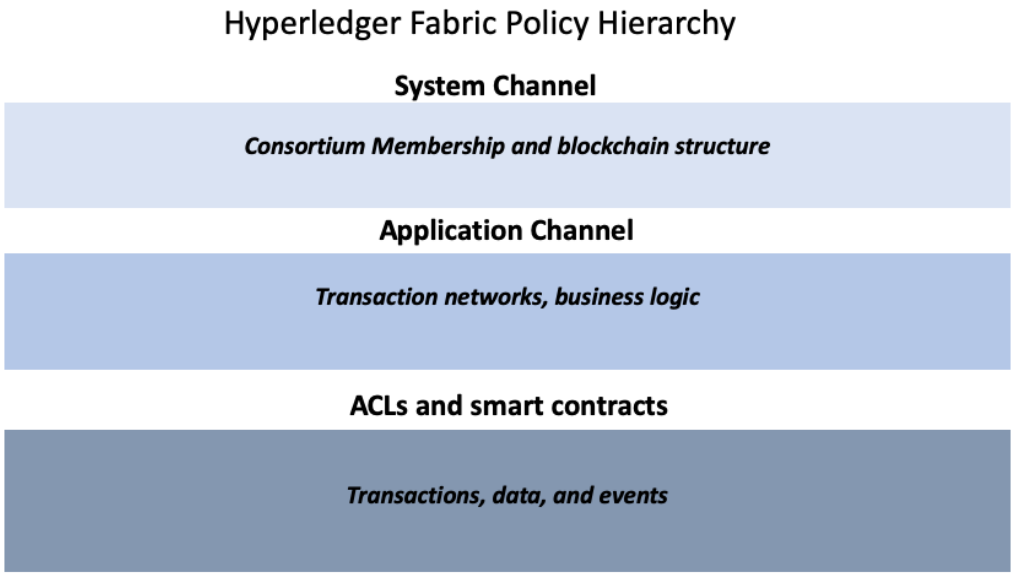


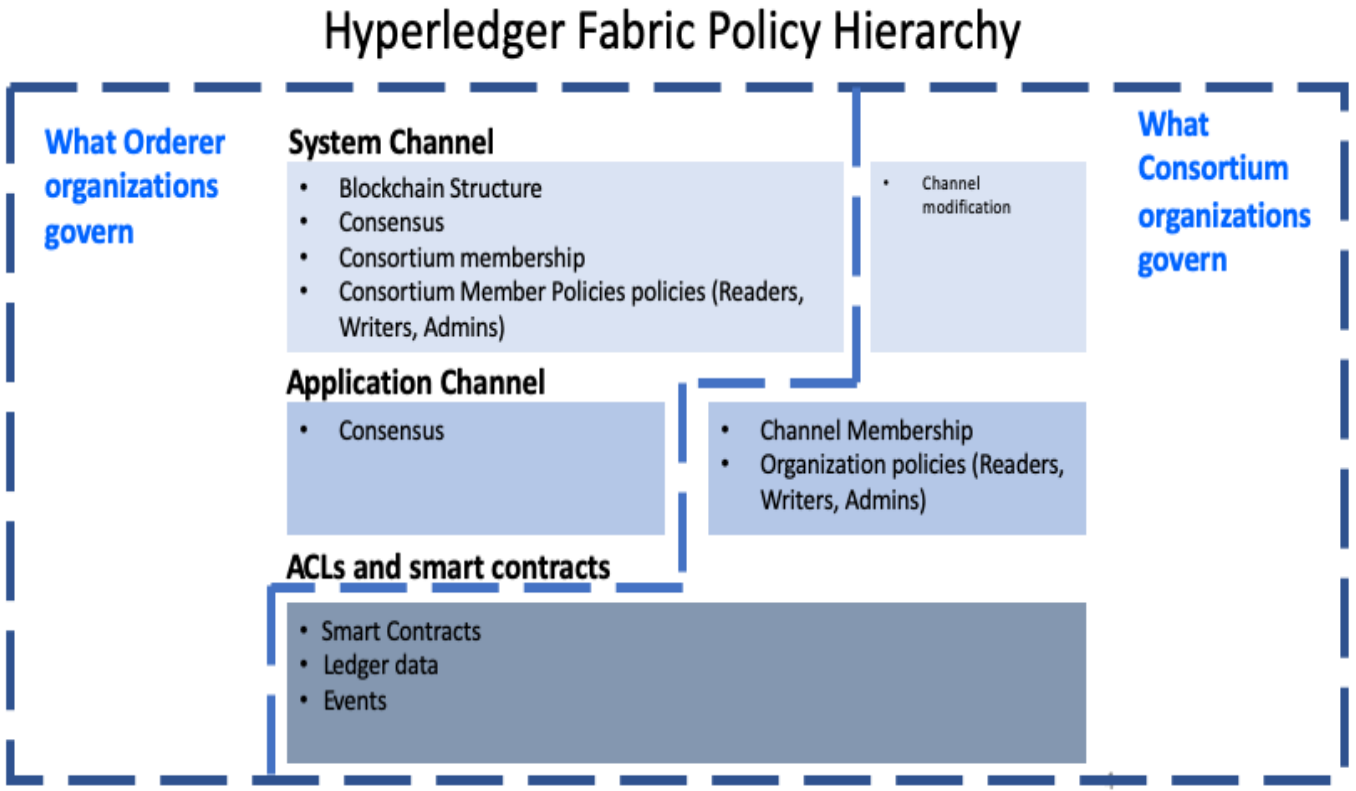
Figure 1: Contract Interpreter Process Request

Fabric Policies

Las políticas se implementan en diferentes niveles de una red Fabric. Cada dominio de política gobierna diferentes aspectos del funcionamiento de una red



Fabric Policies



Endorsement Policy

- Define el conjunto más pequeño de organizaciones que deben respaldar una transacción para que sea válida.
- Descrita o establecida durante la instanciación del chaincode.
- Los directores se describen como 'MSP.ROLE', donde MSP representa el ID de MSP requerido y ROLE representa uno de los cuatro roles aceptados: miembro, administrador, cliente y par.

Endorsement Policy Syntax

Ejemplos:

- 'Org0.admin': any administrator of the Org0 MSP
- 'Org1.member': any member of the Org1 MSP
- 'Org1.client': any client of the Org1 MSP
- 'Org1.peer': any peer of the Org1 MSP

La sintaxis es la siguiente: **EXPR(E[, E...])**

Donde EXPR es cualquier AND, OR, or OutOf, and E es principal (con la sintaxis descrita anteriormente) u otra llamada anidada a EXPR.

Membership Service Provider (MSP)

- Un mecanismo que proporciona a las organizaciones un conjunto de roles y permisos dentro de la red.
- La implementación del requisito de MSP es un conjunto de carpetas que se agregan.
- MSP Types
  - Local MSP
  - Channel MSP

Membership Service Provider (MSP)

- **Las CA** emiten identidades generando una clave pública y privada que forma un par de claves que se puede utilizar para probar la identidad.

- Debido a que una clave privada nunca se puede compartir públicamente, se requiere un mecanismo para habilitar esa prueba, que es donde entra el MSP. Por ejemplo, un par usa su clave privada para firmar digitalmente, o respaldar, una transacción.

- **El MSP es el mecanismo que permite que el resto de la red confíe y reconozca esa identidad sin revelar la clave privada del miembro**.

Membership Service Provider (MSP)

- **Mientras que las CA generan los certificados que representan identidades, el MSP contiene una lista de identidades autorizadas.**

- Además, es el MSP el que convierte una identidad en un rol al identificar los privilegios específicos que tiene un actor en un nodo o canal.

- **Tenga en cuenta que cuando un usuario está registrado con una CA de Fabric, se debe asociar un rol de administrador, peer, cliente, orderer u organización con el usuario representante.**

- El rol de administrador de Sistema sirve:

Administrar e interactuar con los chaincodes, administrar pares y opere funciones básicas de nivel de CA.

Ser administrador de un sistema de HF implica una buena comprensión de la topología de la red de HF, las operaciones del chaincode, la administración de identidades, los permisos, cómo y dónde configurar el registro de componentes y mucho más.

- Las tareas del Administrador del Sistema son:

- Application Lifecycle Management
- Install and Configure Network
- Diagnostics and Troubleshooting including logs
- Membership Service Provider
- Network Maintenance and Operations

- Requisitos técnicos del administrador:

Desde el punto de vista técnico, para convertirse en un administrador de sistemas HF profesional, uno debe adquirir un conocimiento básico de la línea de comandos de Linux, bash, Docker y contenedores (como Kubernetes), NoSQL, CouchDB, blockchain y DLTs.

- Examen de certificación para ser administrador:

Certified Hyperledger Fabric Administrator (CHFA) se recomienda encarecidamente la certificación.

- El rol del Fabric Developer es para:

  En comparación con el administrador del sistema, la tarea principal de los desarrolladores de HF es trabajar en un componente principal de HF, que es el Smart contract o chaincode. En resumen, los desarrolladores deben diseñar, desarrollar, probar e implementar chaincodes de HF en el nodo.

- Las tareas del desarrollador de Fabric son:

- Definir funciones de transacciones
- Ejecutar consultas sencillas
- Crear consultas complejas
- Definir activos utilizando pares clave-valor
- Identificar datos potencialmente privados
- Incorporar la recopilación de datos privados
- Envíe, evalúe y consulte la transacción invocando los Smart contracts

- Requisitos técnicos para Fabric developers:

Los desarrolladores de HF deben tener una base sólida o experiencia en uno de los siguientes lenguajes de programación: JavaScript, Java, Go o Python. El marco JS más famoso que utilizan actualmente los desarrolladores de Fabric es Node.JS.

- Examen de certification para Fabric developers:

Certified Hyperledger Fabric Developer es altamente recomendable.

Recapitulación

Qué hemos aprendido en este webinar?

- Plataforma donde puedes seguir aprendiendo sobre Blockchain, gracias a sus cursos online.

  - Introducción a la programación JavaScript
  - Introducción a Node.JS, Express.JS y MongoDB
  - Introducción al Blockchain en Español
  - Introducción a Ethereum Blockchain en Español

Coding
Bootcamps

- Plataforma donde puedes seguir aprendiendo sobre Blockchain, gracias a sus cursos online.

  - Aprenda programación Solidity con ejemplos
  - Introducción a Hyperledger Fabric Blockchain en Español
  - Hyperledger Fabric Blockchain para el administrador del sistema

Coding
Bootcamps

Contacto

- Puedes contactar por email: info@myhsts.org
- Puedes contactar por Linkedin: https://es.linkedin.com/in/sergiotorrespalomino
- Nos puedes seguir en Linkedin: https://www.linkedin.com/showcase/coding-and-technology-classes

Q/A

coding-bootcamps.com

Thank you

Coding
Bootcamps