



# Implantación de la tecnología Blockchain en infraestructura Serverless

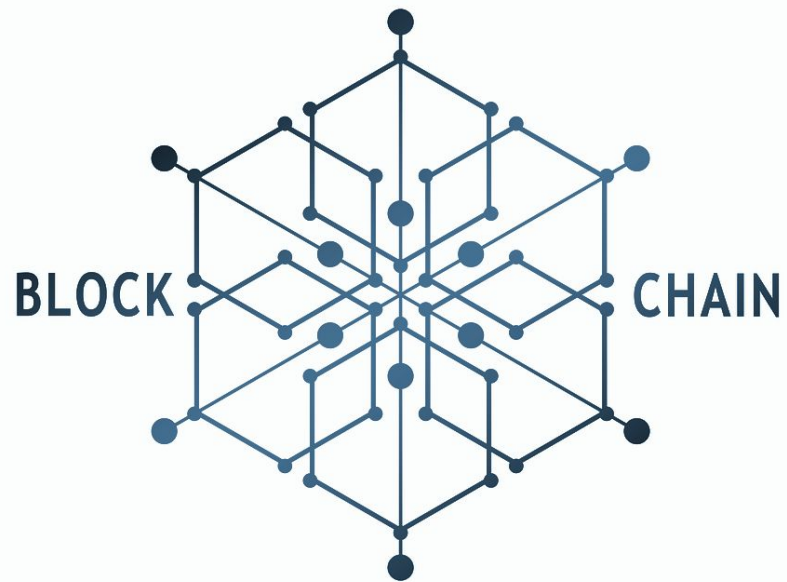
Daniel Ortiz Sánchez

Máster en Ingeniería Informática  
Facultad de Informática  
Universidad Complutense de Madrid

Trabajo Fin de Máster  
Director: José Luis Vázquez Poletti  
Dept. de Arquitectura de Computadores y Automática

# Indice

---



1

**Resumen y Objetivos**

2

**Estado del Arte**

3

**Solución**

4

**Demostración**

5

**Conclusiones y Futuro**



01

# Resumen y Objetivos

Introducción al TFM y a la solución ofrecida.  
Explicación de qué se ha querido conseguir.

# Introduction

---

- Everything turns around the union of **blockchain** and **serverless** technologies.
- It is about abstracting functionality from the first by using the second.
- Keep all its virtues intact.
- Example of **banking application**, similar to the operation of existing ₿.
- It should be understood as a **starting point**, being aware of its limits.

# Main objectives



**Combine Blockchain technology with Serverless.**

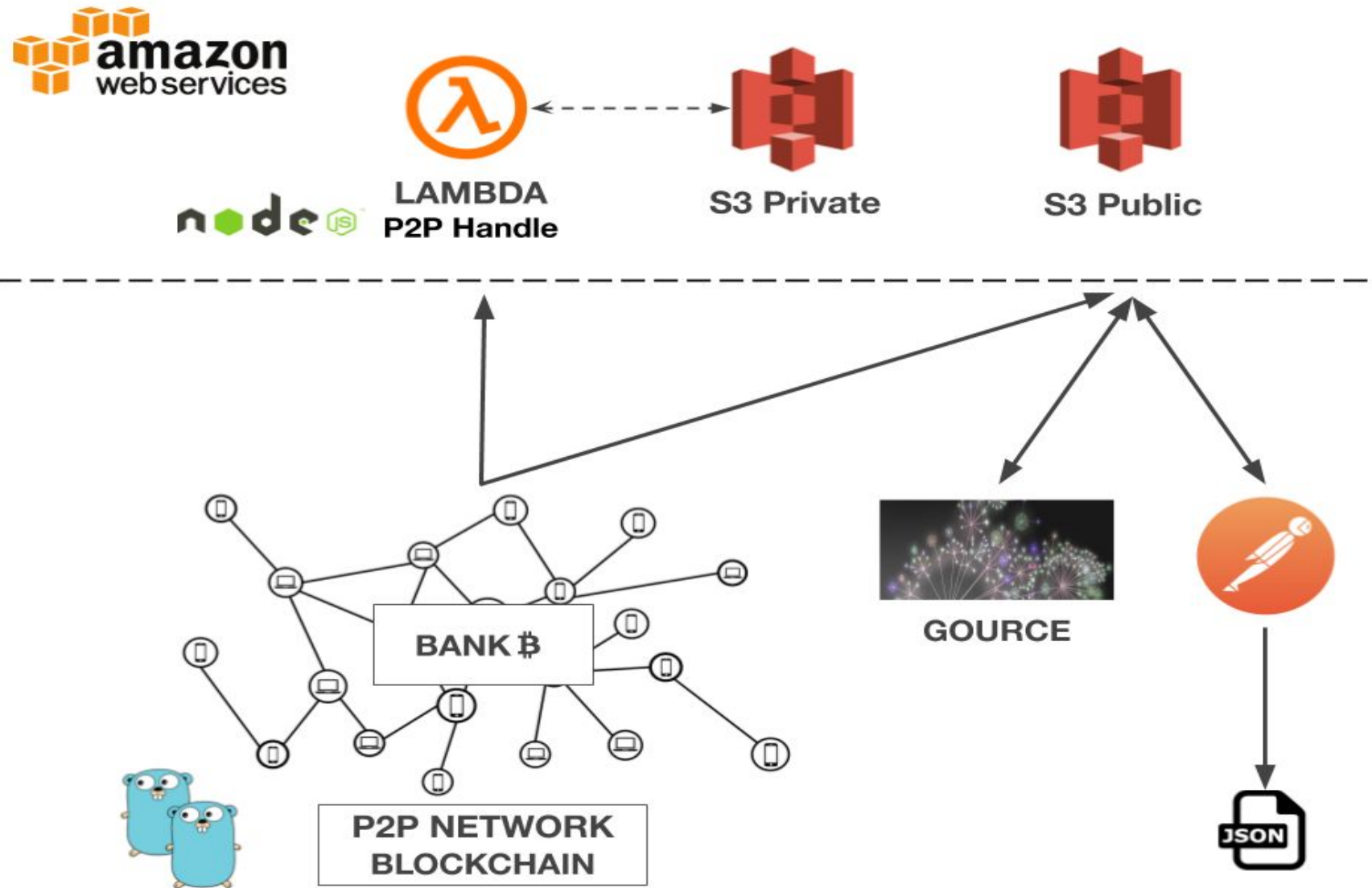


**Build real example with full deployment.**

# Secondary objectives

- 1 Stay within the free layer of cloud services.
- 2 Demonstrate the advantages of using both technologies together.
- 3 Be able to show the internal functioning and movements produced in the network.

# Esquema general





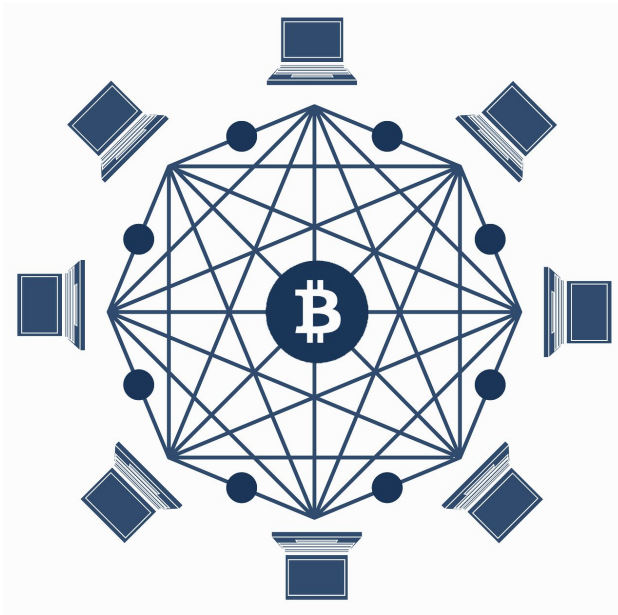
02

# Estado del Arte

Descripción del estado actual de las tecnologías usadas e introducción a los conceptos del TFM.



# Estado del Arte



## Blockchain

Explicación de la tecnología y de sus fundamentos.

## Funcionamiento

Funcionamiento principal de la tecnología blockchain.

## Soluciones

Soluciones actuales del mundo blockchain.

## \*aaS Serverless

Evolución de la nube hasta llegar a Serverless.

# Blockchain: ¿Qué es?

---

- Libro de registros cuya información es infalsificable.
- **Base de datos distribuida y segura.**
- Almacena transacciones y las agrupa en bloques.
- Cada bloque está relacionado con el anterior.

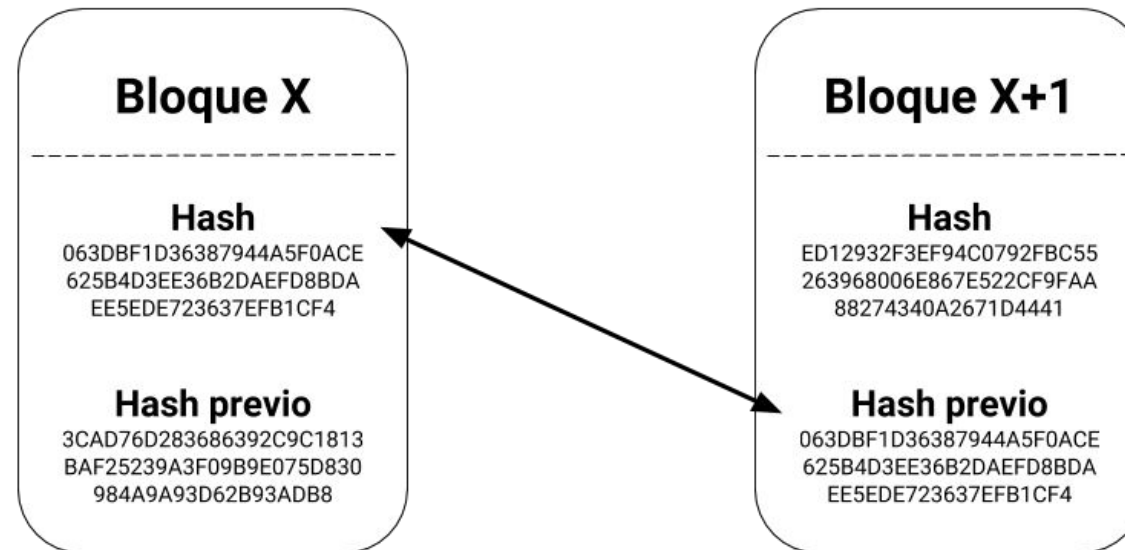


# Blockchain: Características principales

- 1 Es una red p2p. Todos los usuarios son iguales.
- 2 Distribuida. Todos los usuarios tienen copia de los datos.
- 3 Descentralizada. No hay un poder central.
- 4 Inmutabilidad de los datos. No se puede borrar la historia.

# Blockchain: Funcionamiento

- **Hash en los bloques:** PoW y PoS.
- **Minado:** usuarios especiales, recompensas e importancia.
- **Transacciones:** cuenta pública y cuenta privada.



# Soluciones Existentes

## Bitcoin

Primera bitcoin, nace junto con la tecnología blockchain.

## Ethereum

Por norma general, mejora a Bitcoin y añade Smart Contracts.





# Soluciones Existentes

## **Hyperledger**

Orientado hacia el mundo empresarial y redes privadas.

## **Nuevas soluciones:**

AWS Blockchain, IBM Blockchain, Libra (Fb), Fintechs...



# \*aaS: IaaS, PaaS y SaaS

1

**IaaS:** Infraestructura como servicio.

Rápida escalada. AWS EC2 o Google Compute Engine.

2

**PaaS:** Plataforma como servicio.

Despliegues y ciclos de vida. OpenShift.

3

**SaaS:** Software como servicio.

Aplicaciones en nube. GSuite, Slack o Salesforce.

# \*aaS: FaaS, BaaS y Serverless

---

- **Función como servicio.** Se delega parte de la funcionalidad a la nube.
- Ejecución de código mediante eventos o respuestas a peticiones.
- Solo ámbito local. Normalmente se usa combinado con otros servicios.







03

# Solución

Descripción de la solución aportada.  
Cómo funciona y porqué de esa manera.

# Solución



## Tecnologías

Explicación de las tecnologías usadas.

## AWS

Qué herramientas de AWS y cómo se han usado.

## Cliente

Explicación de las opciones y menús disponibles.

## Salidas

Explicación de las dos formas de obtener salidas.

# AWS: IAM y S3

- **S3**: sistema de almacenamiento de Amazon en buckets.
- Operaciones a objetos mediante peticiones (SDK) o mediante web.
- Se usan 2 buckets: uno público (para consumir) y uno privado (de trabajo).
- Necesarias configuraciones independientes por bucket.
- **IAM**: necesario para crear roles de acceso desde el exterior.



# AWS: Lambda

---

- Permite implementar **microservicios** como funciones abstraídas.
- Coste por solicitud a la llamada. Suele reducir coste.
- Soporte para múltiples lenguajes. En nuestro caso Node.js.
- Encargado de controlar el estado de los nodos en la red p2p.
- **CloudWatch:** herramienta de logs para ayuda al depurar las funciones.



# Cliente

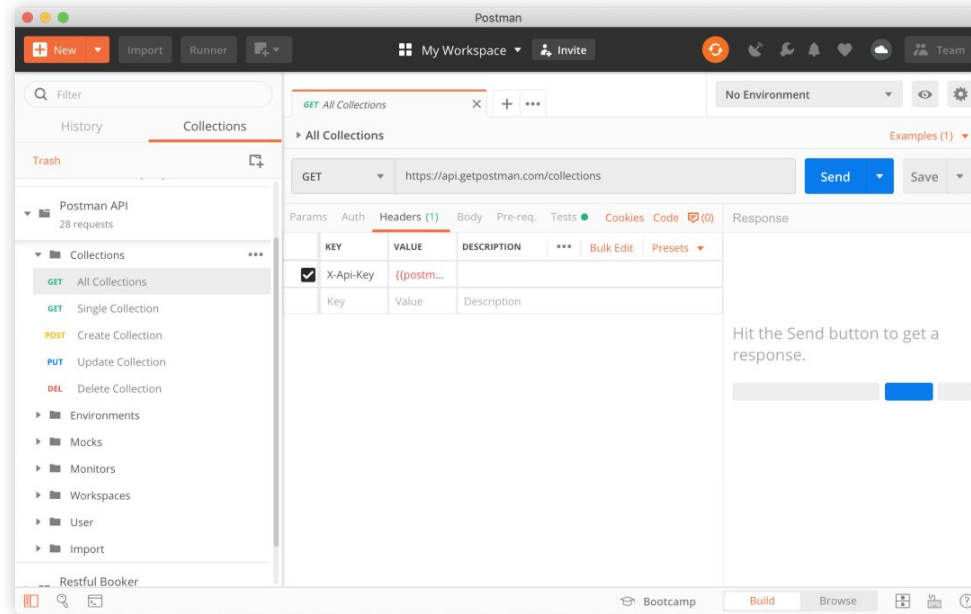
---

- Desarrollado en lenguaje **Go** ayudándose de la librería **LibP2P**.
- Cada usuario debe tener una copia del cliente para conectarse.
- Contiene estructuras de datos y comunicaciones con AWS.
- Se encarga de las funciones principales.
- Medio por el cual el usuario realiza transferencias.



# Postman

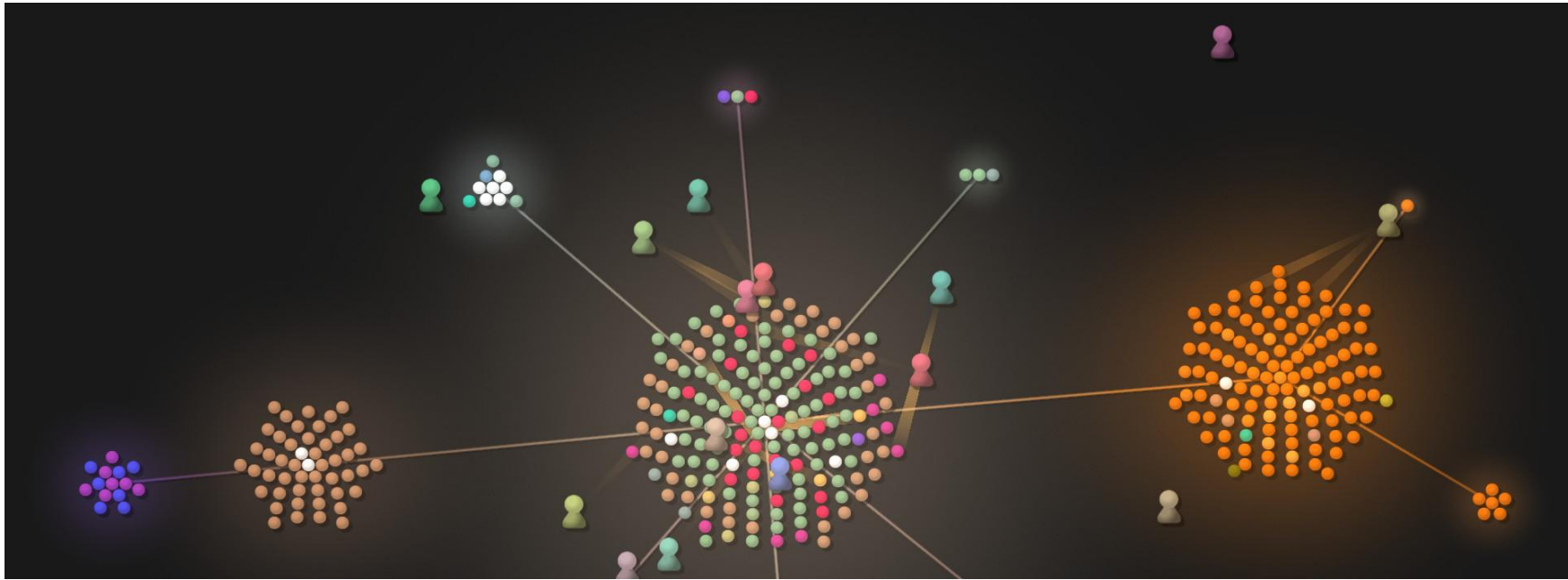
- Herramienta para realizar peticiones **HTTP**. Ideal para probar y depurar API REST.
- Permite agrupar las llamadas en colecciones para su uso posterior.



# Gource

---

- Herramienta de visualización de archivos de log. Originaria de proyectos software.
- Video temporal en el que se muestran las distintas interacciones.





04

# Demostración

Ejecución del caso de uso de la solución.









05

# Conclusiones y Futuro

Conclusiones y reflexiones del TFM.  
Líneas a seguir para el futuro.

# Conclusions

---

-  The main objective of combining blockchain with serverless has been achieved.
-  It has been demonstrated that diverse functionalities can be externalised to serverless computing.
-  A blockchain network has been developed from scratch.
-  The relevance of this project is highlighted in two fundamental points.

# Trabajo Futuro



## Red Global

Ir más allá de las redes usadas en las demostraciones y casos de uso.

## Minado y Log

Implantar algoritmos de minado PoW o PoS externalizado en funciones serverless.

## Aplicación

Construcción de aplicación web/móvil mediante la que interactuar con el cliente.

## Smart Contracts

Implantar Smart Contracts similar a otras tecnologías ya existentes.



Daniel Ortiz Sánchez

*danortiz@ucm.es*

**Muchas gracias**