WIRELESS SERVICE ACCOUNTING BASED ON ETHEREUM STATE CHANNELS - MASTER THESIS

A. EGIO, L. GONZÁLEZ, A. PARDO, T. ROMERO

ABSTRACT. This document contains authors' "*Blockchain technologies master*"[1] final thesis. The project described in the document consists on a technological proof of concept of an internet access service consumption using a state channel smart contract as mean of payment. After a brief introduction, there's a first motivation of the use case included; in the following sections there's a detailed explanation of the architecture, implementation and security considerations; after that it is described a more general approach on using blockchain technologies and state channels in particular to provide mechanisms to reduce advanced wireless (i. e. 5G) services market friction between neutral host providers, virtual operators and operators licensing spectrum bands to nations' governments; guarantee transparency to consumers, and provide a fully GDPR compliant schema using a self-sovereign identity system.

Contents

1. Introduction

As exposed in [?] although radio spectrum "*is often referred to as a 'scarce resource'*" there are spectrum management strategies being researched to improve and support a brand new model of services into the market; in particular there are multiple 5G[?] innovative projects and initiatives that try to solve this using Software Defined Network[?] and Network Function Virtualization[?].

Since spectrum is a scarce resource, it can be tokenized being suited this way for blockchain/DLT technologies business models; in addition, since atmospheric spectrum can be considered a public asset, it's managed in last terms by nations' governments and it's licensing and management should feature a high level of transparency and at the same time allow for efficiency and fair-competition making it feasible to model on public blockchain systems.

Taking this into account, lead the authors' to develop a first bottom-up proof of concept of a simple system demonstrating how can a wireless WiFi hotspot deployed on a commodity hardware asset[?] be managed to allow users to access the internet paying in exchange for Ethereum on a public network.

One of the first intiatives to create a bandwidth reselling market was La Fonera[?] back in 2006. Main goal of the initiative was to allow ADSL users to share unused bandwidth using a specific WiFi hotspot deployed on an ad-hoc built router.

Figure 1. *La Fonera's initial project router.*

A more recent project like Hotspot Me[?] and also very aligned with the technological demonstration described in this document consist on using mobile APIs to allow a users to share WiFi hotspot tethering to users nearby in exchange for ETH supported by a micro-payment channel contract.

2. Technological background

Blockchain technologies support a ledger of records in continuous growth called blocks. They are linked and secured by cryptography. Adopt the P2P protocol (*peer-to-peer*) in order to be distributed with not a single point of failure.

The consensus mechanism ensures a common order of unambiguous transactions and blocks, and guarantees integrity and blockchain consistency a through geographically distributed nodes. By its design, the blockchain has characteristics such as: decentralization, integrity and auditability. The blockchain can serve as a new type of software connector, which should be considered as a possible decentralized alternative to storage of existing centralized shared data.

In addition, depending on the different levels of access permission, block chains can split into two types: 1) public (such as Bitcoin and Ethereum); Y 2) private (such as Hyperledger). Blockchain serves as a platform for smart contracts (hereinafter, textit smart contracts). For example, programmed in the Solidity language of Ethereum, stay and run. Blockchain is a technology concept DLT distributed ledger (textit distributed ledger technology). It can be integrated into multiple business areas.

3. Technological Proof of Concept description

3.1. **Architecture.** In order to support the PoC a Raspberry Pi 3 Model B (see figure 2) has been used as a single board computer to host all the required software. Tested operative system was OSMC[?] (a Debian based GNU-Linux flavor) mainly used as media-center. A WiFi access point using hostapd[?]

daemon over Raspberry's WiFi chipset. Raspberry Pi was also connected using ethernet cable to an optical fiber commercial router (*see figure 2*).

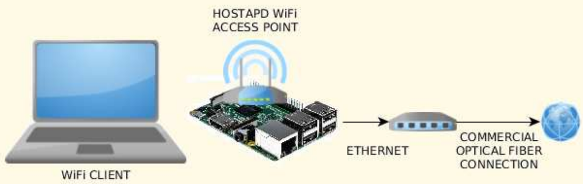

Figure 2. Physical Setup.

Dnsmasq[?] was used as dynamic host configuration server for the wireless network users and iptables[?] was used as a firewall running both inside the Raspberry; by default, iptables was configured to drop all packet forwarding traffic from users connected to the wireless domain and at the same time to redirect all traffic to Raspberry's HTTP port where a captive portal was listening incoming connections.

Main idea is that the users can join freely to the wireless hotspot and access the web captive portal (*hosted on the same Raspberry*) application that communicates with a back-end that controls iptables rules and also an Infura light proxy able to relay RPC calls arriving on 8545 port of the Raspberry.

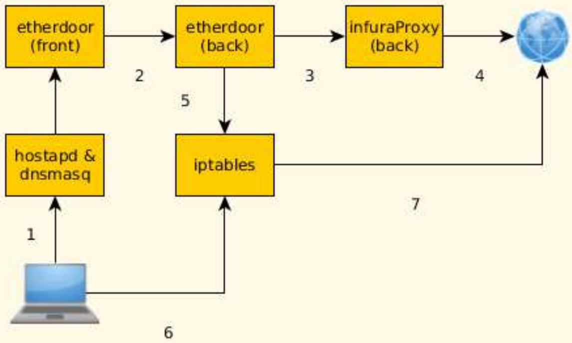

Figure 3. Physical Setup.

4. Further steps

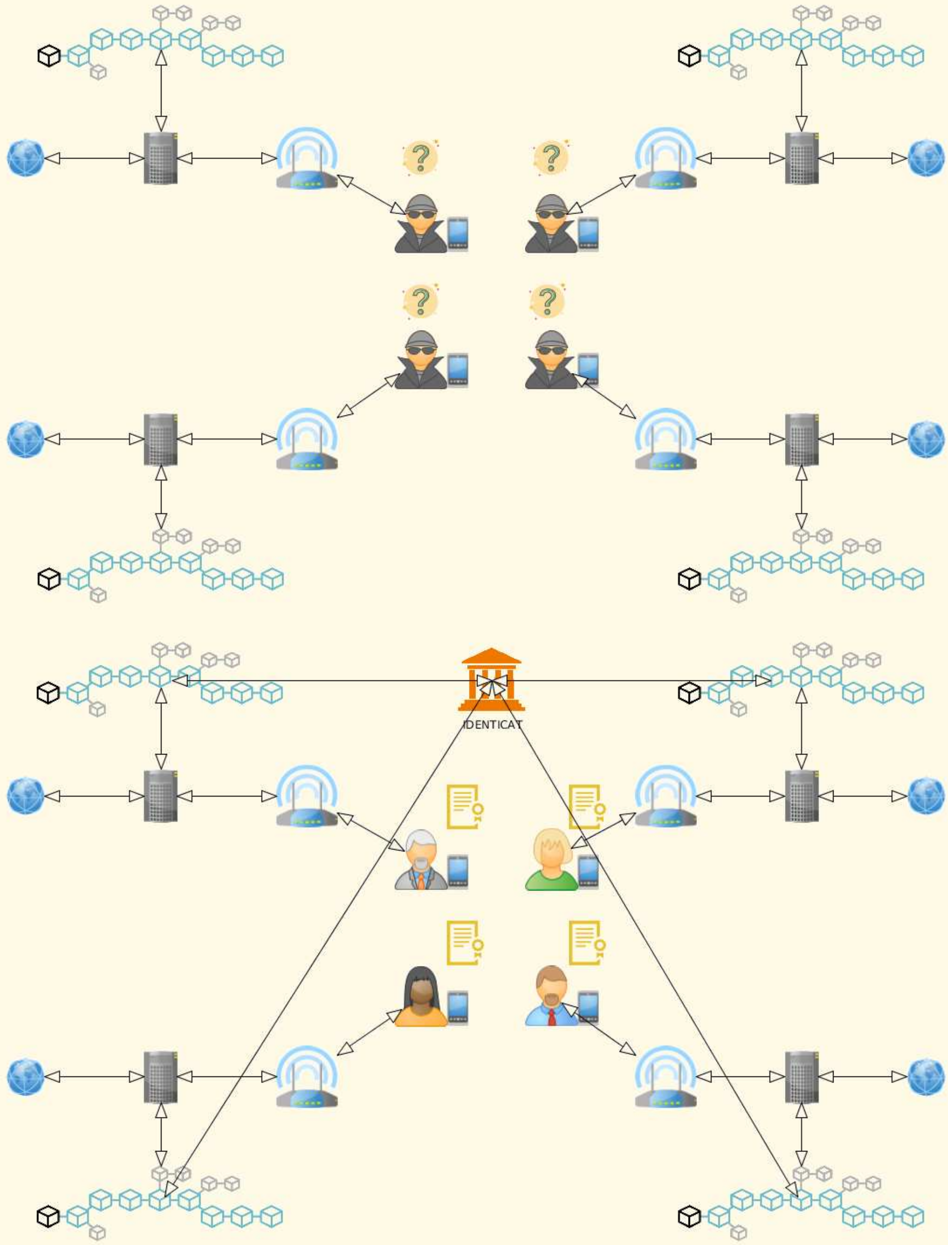

Figure 3. Anonymicity, identity, KyC and GDPR.

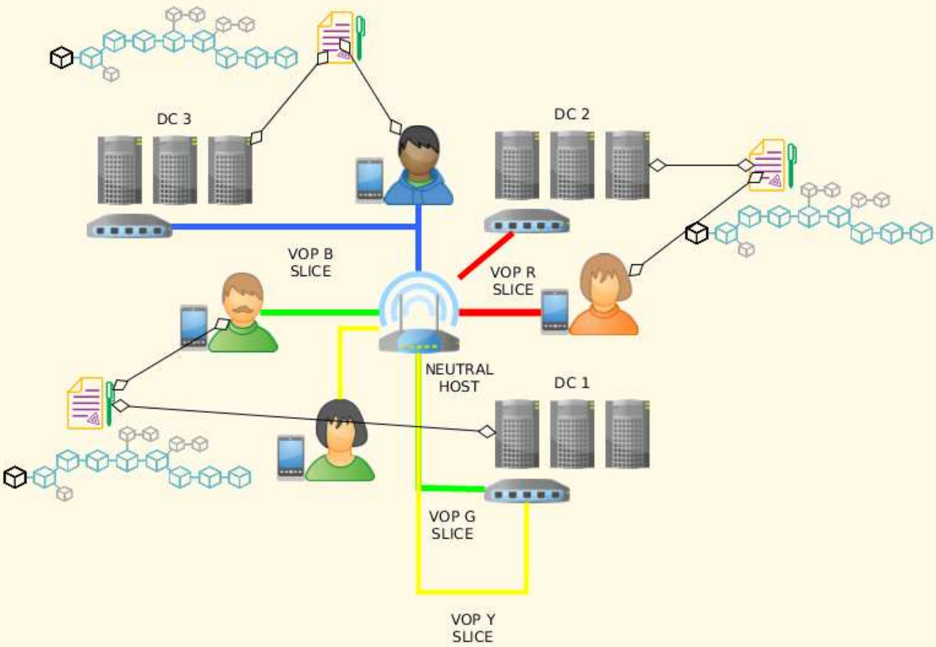

Figure 4. Slicing model.

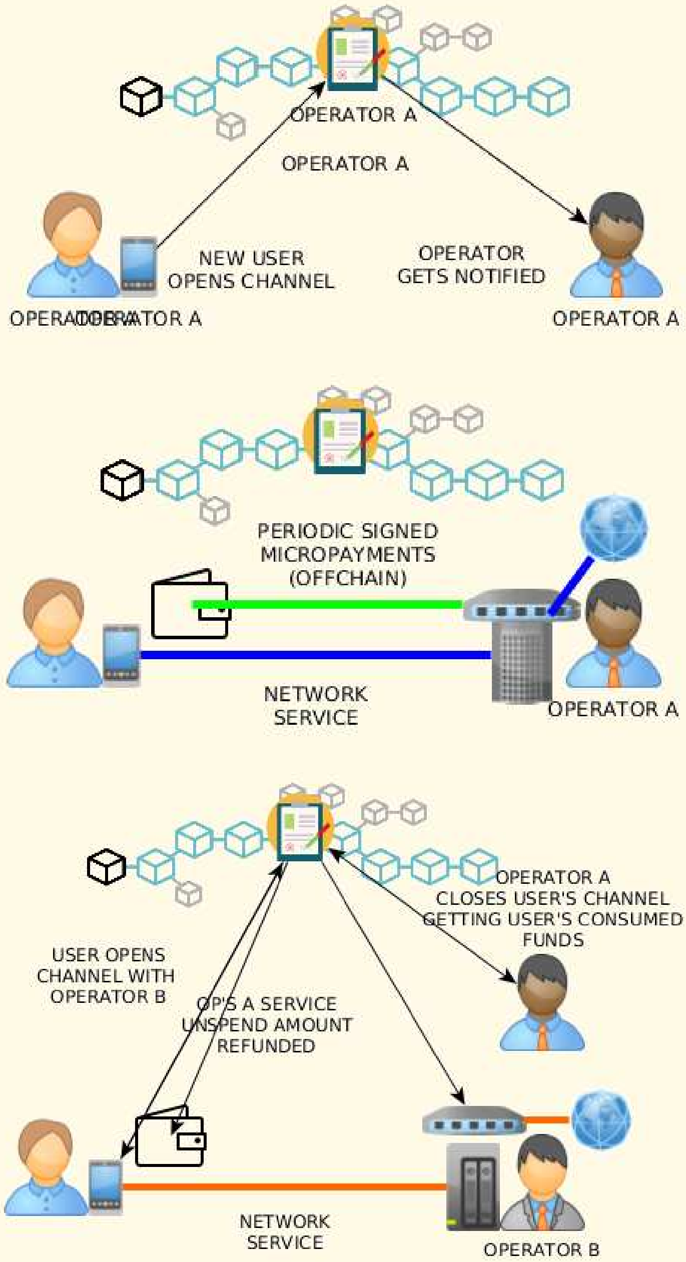

Figure 5. State channels support instant portability between different operators with minimal fee.

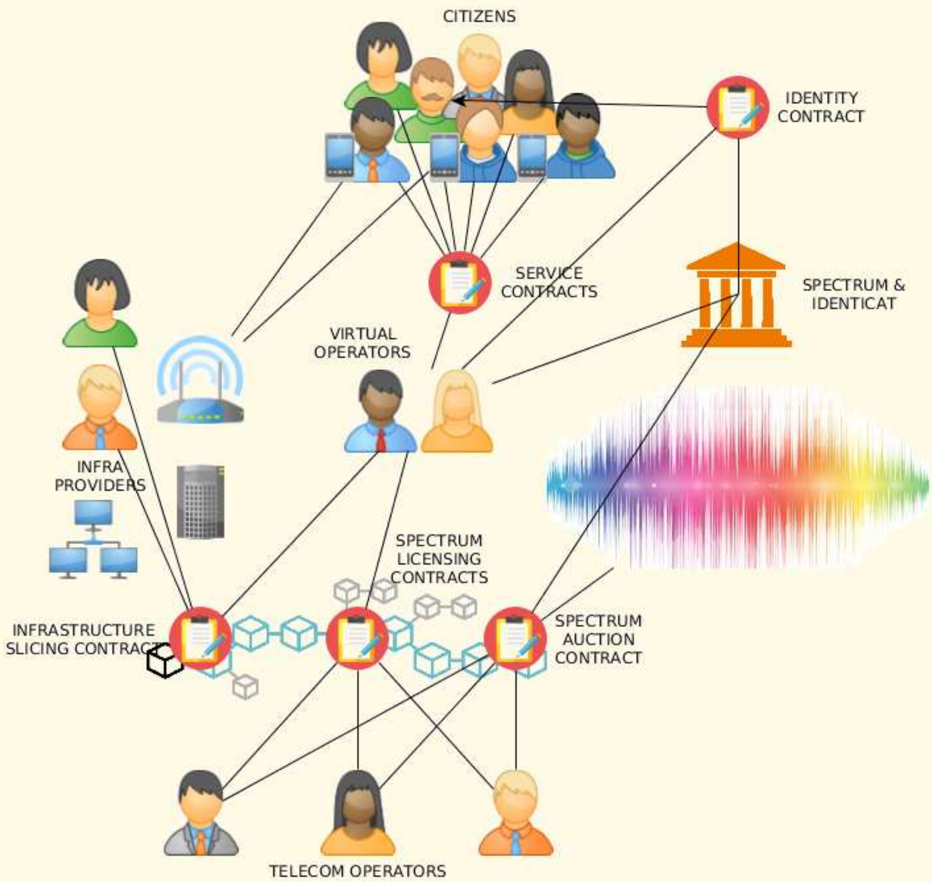

Figure 6. Full tokenization of wireless services from the spectrum licensing to the citizen.

4.0.1. *Smart contract.*

4.0.2. *Front-end.*

4.0.3. *Back-end.*

4.1. **Implementation.**

5. Quality considerations

Para validar la PoC, se ha definido una batería de tests. Por ejemplo, dado que los smart contract normalmente manejan dinero, es esencial asegurarse de que su número de fallos y vulnerabilidades sea bajo [Hegedus 18]. Para ayudar a los desarrolladores y hacer más madura la tecnología, necesitamos herramientas de análisis [ConsenSys 19].

5.1. **Automated tests.** (Por completar)

6. Deployment

Docker es una herramienta que permite desplegar aplicaciones dentro de contenedores de software. Esto puede ser útil para la Raspberry Pi porque permite a los usuarios ejecutar aplicaciones con muy poca sobrecarga, siempre y cuando la aplicación esté empaquetada dentro de una imagen Docker. Simplemente instalamos Docker y ejecutamos el contenedor sobre Raspbian/ARM. Éste despliega el proyecto realizado.

La secuencia de comandos a ejecutar es:

```
$ sudo apt-get install git
$ git clone https://github.com/ethereum-internet-access/docker.git
$ cd docker
$ sh ./install-project-inside-docker-container.sh
```

7. Viabilidad de negocio

(Por completar)

7.1. **Legislación.** (Por completar)

8. Conclusiones

El mundo en el que vivimos es interconectado. En él Internet juega un papel fundamental. Se ha contribuido a la construcción de una PoC. Ésta se ha basado en la medida de lo posible en hardware libre (Raspberry Pi) ya que goza de buena popularidad para el desarrollo de prototipos. Como debilidad del pago con ETH, se puede achacar la volatilidad de su precio (véase figura 3). Esto conlleva variaciones en las comisiones (*fees*) de una transacción Ethereum.

Figura 3. Variación cotización ETH (11/08/2019). Fuente: [Coinbase 19]

(Por completar)

9. Líneas futuras

Como líneas futuras y complemento a este trabajo, con más tiempo, se plantea:

9.1. Añadir forma de pago DAI. DAI es una criptomoneda estable (*stablecoin*) que funciona con ETH y que intenta mantener un valor de 1$. A diferencia de los billetes de banco centralizados, DAI no está respaldado por dólares estadounidenses en una cuenta bancaria. En cambio, está respaldado por garantías en la plataforma Maker [MakerDAO 19].

9.2. Usar Raspberry Pi 4 Modelo B. Anunciado en Junio de 2019 y a la venta. Es un hardware que ofrece mejores prestaciones. Por ejemplo, el procesador es un Broadcom BCM2711. Un ARM Cortex-A72 de cuatro núcleos y a 1,5 GHz. Hasta tres veces más eficiente (*benchmark*) que el modelo anterior (Raspberry Pi 3 Modelo B+). También da la posibilidad de elegir entre 1, 2 y 4 GB de RAM.

9.3. Usar Docker Swarm. Se podría presentar una aplicación descentralizada basada en microservicios. Por ejemplo en microservicios, un contenedor Docker (como en la sección 4) podría verse como un servicio. Entre las ventajas que aporta microservicios estarían:

- Pequeños
- Independientes
- Despliegue sencillo
- Reutilizables
- Externalización
- Escalabilidad

Para ello, Docker ofrece un orquestador (de código abierto) llamado Docker Swarm (véase figura 4). Un nodo *worker* sería nuestra Raspberry Pi y un nodo *manager* sería un centro de control.

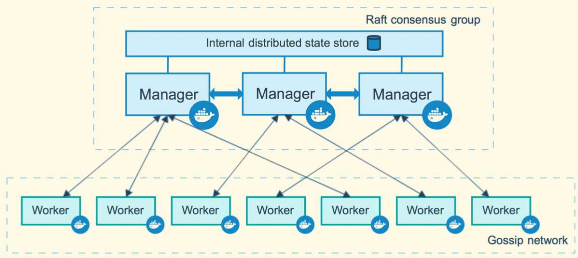

Figura 4. Diagrama Docker Swarm. Fuente: [Docker 19]

10. Repositorio

El código fuente generado en el proyecto está publicado en un repositorio público de GitHub bajo licencia MIT.

https://github.com/ethereum-internet-access

References

[1] 5G Spectrum and Neutral Hosting, Discussion paper, February 2019, DCMS Phase 1 5G Testbeds & Trials Programme
[2] https://en.wikipedia.org/wiki/5G
[3] Ian F. Akyildiz and Shih-Chun Lin and Pu Wang, *Wireless software-defined networks (W-SDNs) and*

network function virtualization (NFV) for 5G cellular systems: An overview and qualitative evaluation, Computer Networks, 93, 2015

[4] Jose Ordonez-Lucena et al. *Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges*, IEEE Communications Magazine (Volume: 55 , Issue: 5 , May 2017)

[5] https://es.wikipedia.org/wiki/La_Fonera

[6] https://devpost.com/software/hotspot-me

[7] https://w1.fi/hostapd/

[8] https://linux.die.net/man/8/iptables

[9] https://en.wikipedia.org/wiki/Dnsmasq

[10] https://osmc.tv/

[11] Coinbase. Intercambio de moneda digital https://www.coinbase.com/price/ethereum

[12] Ethereum Smart Contract Best Practices. Security tools. ConsenSys. 12 Agosto 2019.

[13] Docker Documentation. How nodes work. 6 Septiembre 2019. https://docs.docker.com/engine/swarm/how-swarm-mode-works/nodes/

[14] Hackathon ETH Berlín Zwei. Twitter. 6 Septiembre 2019. https://twitter.com/ETHBerlin

[15] P. Hegedus. “Towards Analyzing the Complexity Landscape of Solidity Based Ethereum Smart Contracts”. 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). 27 Mayo-3 Junio 2018. https://ieeexplore.ieee.org/document/8445056

[16] MakerDAO. Stability for the blockchain. https://makerdao.com/en/dai

[17] J. L. Muñoz. “Docker Containers”. Apuntes de las clases magistrales. Máster en Tecnologías Blockchain. 1, Edición. UPC School. 29 Noviembre 2018.

[18] S. Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. 31 Octubre 2008. https://bitcoin.org/bitcoin.pdf

[19] Raspberry Pi 3 Modelo B - Placa Base (1,2 GHz Quad-Core Arm Cortex-A53, 1 GB RAM, USB 2.0). Amazon. 11 Agosto 2019.

[20] Raspberry Pi FAQs. BUYING AND SHIPPING. Where can I buy a Raspberry Pi?. Página Web oficial. 11 Agosto 2019. https://www.raspberrypi.org/help/faqs/#buyingWhere

[21] W. Stallings. “Comunicaciones y redes de computadores”. 1 Septiembre 2004. Páginas 904. Pearson Prentice Hall. ISBN-13: 978-8420541105.