

클라우드 환경에서 블록체인을 이용한 포그 기반 IoT 서비스 상호운용 시스템

(A Fog-based IoT Service Interoperability System using Blockchain in Cloud Environment)

김미선*, 박용석**, 서재현***

(Mi Sun Kim*, Yong Suk Park**, Jae Hyun Seo***)

요약

사물 클라우드(CoT, Cloud of Things)는 사물인터넷(IoT, Internet of Things) 애플리케이션에 클라우드 서비스가 지원하는 무제한 저장기능과 처리능력을 제공할 수 있다. 그러나, 중앙 집중식 사물 클라우드에서는 병목 문제, 사물 클라우드 네트워크의 중단으로 이어질 수 있는 단일 장애 지점을 발생시킬 수 있다. 본 논문에서는 중앙 집중식 사물 클라우드의 문제를 해결하고 서로 다른 서비스 도메인 간 상호운용을 위하여, 분산 포그 컴퓨팅과 블록체인 기술을 적용한 IoT 서비스 상호운용 시스템을 제안한다. 분산 포그를 사용하여 IoT 장치와 지역적으로 가까운 거리에 위치한 포그 시스템에서 실시간 데이터 처리 및 서비스를 제공하고, 블록체인의 스마트 컨트랙트와 분산 원장을 이용하여 각 포그간에 서비스 상호운용이 가능하도록 한다. 제안 시스템은 클라우드로부터 서비스를 위임받은 분산 포그에서 가까운 지역 내 서비스를 제공하며, 포그 간에도 클라우드를 거치지 않고, 다른 포그의 서비스를 접근할 수 있다. 또한, 블록체인 네트워크상에서 클라우드와 포그 노드들은 서비스 권한 토큰 발행 정보를 공유함으로써 토큰에 대한 무결성을 보장하고 포그 노드들 간 신뢰할 수 있는 서비스 상호운용이 수행될 수 있다.

■ 중심어 : 사물클라우드 ; 포그 컴퓨팅 ; 블록체인 ; 서비스 상호운용

Abstract

Cloud of Things (CoT) can provide IoT applications with unlimited storage functions and processing power supported by cloud services. However, in a centralized cloud of things, it can create a single point of failure that can lead to bottleneck problems, outages of the CoT network. In this paper, to solve the problem of centralized cloud of things and interoperate between different service domains, we propose an IoT service interoperability system using distributed fog computing and blockchain technology. Distributed fog is used to provide real-time data processing and services in fog systems located at a geographically close distance to IoT devices, and to enable service interoperability between each fog using smart contracts and distributed ledgers of the blockchain. The proposed system provides services within a region close to the distributed fog entrusted with the service from the cloud, and it is possible to access the services of other fogs without going through the cloud even between fogs. In addition, by sharing a service right token issuance information between the cloud and fog nodes using a blockchain network, the integrity of the token can be guaranteed and reliable service interoperability between fog nodes can be performed.

■ keywords : Cloud of Things ; Fog Computing ; Blockchain ; Service Interoperability

I. 서론

사물인터넷은 인터넷과 연결된 다양한 디바이스로부터 많은 정보를 수집, 모니터링, 제어 또는

상호작용할 수 있는 물리적인 개체 시스템으로 스마트 시티, 스마트 산업, 스마트 헬스케어, 스마트 팜 등 다양한 산업 시스템에 광범위하게 적

* 정회원, 목포대학교 정보보호학과 강사

** 정회원, 목포대학교 정보보호기술학협동과정 박사과정

**** 정회원, 목포대학교 정보보호학과 교수

본 논문은 2021년도 교육부의 재원으로 이공분야기초연구사업의 지원을 받아 수행된 연구임.(No. NRF-2021R1F1A1063603).

접수일자 : 2022년 04월 11일

수정일자 : 2022년 04월 26일

게재확정일 : 2022년 04월 28일

교신저자 : 서재현 e-mail : jhseo@mnu.ac.kr

용되고 있다. 그러나, IoT 장치는 메모리, 전력 및 처리 자원의 한계를 가지고 있어 이를 극복하기 위하여 클라우드 컴퓨팅과 접목한 패러다임이 나타났다[1].

사물 클라우드는 IoT 애플리케이션에 클라우드 서비스가 지원하는 무제한 저장기능과 처리능력을 제공할 수 있다. 또한, 사물 인터넷은 클라우드 컴퓨팅과 결합함으로써 확장성, 상호운용성, 신뢰성, 효율성, 가용성 및 보안성을 더 가질 수 있게 되고, 접근 용이성과 사용의 용이성 및 배치의 비용 감소 등 추가적인 이점을 얻을 수 있게 된다.

그러나, 중앙 집중식 사물 클라우드에서는 병목 문제, CoT 네트워크의 중단으로 이어질 수 있는 단일 장애 지점을 발생시킬 수 있다. 또한, IoT에서 발생한 모든 데이터가 클라우드로 전송됨에 따라 IoT 소유자의 동의 없는 개인 정보 유출의 문제를 포함하고 있어 이를 위한 보안 방안이 필요하다.

포그 컴퓨팅은 기존 사물 클라우드에서 IoT 장치가 생성하는 모든 데이터를 클라우드로 전송하여 저장하는 경우 저장용량의 한계 및 전송 지연 등의 문제들을 해결하기 위해 도입된 개념이다[2-5].

본 논문에서는 중앙 집중식 사물 클라우드의 문제를 해결하기 위하여, 분산 포그 컴퓨팅과 블록체인 기술을 적용한 서비스 상호운용 시스템을 제안한다. 제안 시스템은 분산 포그를 사용하여 IoT 장치와 지역적으로 가까운 거리에 위치한 포그 시스템에서 실시간 데이터 처리 및 서비스를 제공하고, 블록체인을 이용하여 각 포그 간에 서비스 상호운용이 가능하도록 한다.

클라우드가 위임한 로컬 서비스는 분산된 포그에서 수행되며, 다른 포그에 대한 서비스 요구는 블록체인의 스마트 컨트랙트를 통해 수행될 수 있다. 기존 사물 클라우드의 모든 서비스가 클라우드를 통해 제공되는 것과 달리, 클라우드로부터 서비스를 위임받은 분산 포그에서 가까운 지

역 내 서비스를 제공한다. 또한, 포그 간에도 클라우드를 거치지 않고, 다른 포그의 서비스를 접근할 수 있다.

IoT 기기에 대한 인증은 포그에서 이루어지므로, 생성된 데이터에 대한 안전성을 보장받을 수 있다. 또한, 포그에서 필터링을 통해서 보안이 적용된 IoT 데이터 서비스를 제공함으로써 개인정보보호 및 IoT 데이터의 무분별한 사용을 제한할 수 있다.

본 논문에서 클라우드 서버는 블록체인의 스마트 컨트랙트를 사용하여 포그 노드에게 SR을 공유하여 포그 노드가 로컬 서비스를 생성하도록 함으로써, 서비스를 위임할 수 있으며, 포그 노드는 서비스 권한 토큰을 사용하여 각 포그 간 서비스를 상호운용할 수 있다. 본 논문에서는 제안한 시스템의 설계를 검증하기 위해서 클라우드 서버와 포그 노드로 구성된 사물 인터넷 서비스를 시뮬레이션하였으며, 분산 원장을 이용하여 정보를 공유하고 이를 통한 서비스 공유가 가능한지를 테스트하였다.

본 논문의 구성은 다음과 같다. 본론의 2장에서는 클라우드-포그 컴퓨팅 및 클라우드 컴퓨팅과 블록체인 통합 기술에 관한 관련 연구를 설명한다. 3장에서는 제안하는 클라우드 환경에서 블록체인을 이용한 포그 기반 IoT 서비스 상호운용 시스템 설계에 대해 기술하고, 4장에서는 시스템 구현 및 테스트 시나리오를 기술한다.

II. 관련연구

1. 클라우드-포그 컴퓨팅

클라우드 컴퓨팅은 주문형 및 확장 가능한 스토리지와 IoT 요구 사항에 맞게 확장할 수 있는 처리 서비스를 제공한다. 그러나 상태 모니터링, 비상 대응 및 기타 지연시간에 민감한 애플리케이션의 경우에는 데이터를 클라우드로 전송한 후 애플리케이션으로 다시 전송함으로써 발생하

는 지연이 문제가 될 수 있다. 또한, 과도한 네트워크 대역폭 사용의 문제도 발생할 수 있다.

기존 사물 클라우드에서 IoT 장치가 생성하는 모든 데이터를 클라우드로 전송하여 저장하는 경우 저장용량의 한계 및 전송 지연 등의 문제들을 해결하기 위해 도입된 개념이 포그 컴퓨팅이다. 포그 컴퓨팅은 네트워크 에지에 클라우드와 같은 서비스를 제공하는 분산형 패러다임으로 자체 인프라와 함께 클라우드 및 에지 리소스를 활용한다. IoT 장치에서 생성되는 모든 데이터를 클라우드로 보내지 않고, 지역적으로 IoT 장치와 가까운 거리에 위치하는 포그(또는 에지) 장치에서 처리한다. 영구적인 저장이나, 복잡한 처리 과정이 수행되는 경우에만 클라우드로 전송함으로써 실시간 서비스 및 빠른 대용량 데이터 처리 서비스등을 제공할 수 있다[2]. 많은 양의 데이터 저장, 통신, 제어, 구성 및 관리를 수행하기 위해 사용자 근처의 클라이언트 또는 에지 장치를 활용하여 IoT 데이터를 로컬로 처리한다. 이 접근 방식은 클라우드 자원의 온디맨드 확장성을 활용하면서 에지 장치의 센서와의 근접성에서 이점을 얻을 수 있다[3].

포그 지원 시스템 아키텍처는 센서 네트워크 및 원격 의료 센터의 일부 부담을 처리함으로써 이동성, 에너지 효율성, 확장성 및 안정성 문제와 같은 유비쿼터스 의료 시스템의 많은 문제에 대처할 수 있다[6,7].

[7]은 스마트 홈이나 병원의 센서 인프라 네트워크와 클라우드 사이에 로컬 스토리지, 실시간 로컬 데이터 처리, 임베디드 데이터 마이닝 등과 같은 여러 상위 수준 서비스를 제공하는 스마트 e-Health 게이트웨이를 포그에서 지원하는 시스템을 제안하고 UT-GATE라는 스마트 e-Health 게이트웨이의 프로토타입을 제시하고 있다.

2. 블록체인과 사물 클라우드 컴퓨팅 통합

블록체인과 사물 클라우드 컴퓨팅 통합은 IoT 장치가 제공하는 실제 데이터는 클라우드 스토

리지에 보관하고, 암호화 해시된 메타 데이터는 블록체인에 저장함으로써 강화된 보안을 제공할 수 있다. 또한, 스마트 계약을 채택하여 높은 클라우드 데이터 유효성을 보장한다[1].

스마트 헬스케어, 스마트 운송, 스마트 시티, 스마트 에너지 및 스마트 산업과 같이 IoT와 관련된 다양한 영역에서 클라우드-포그 컴퓨팅의 보안과 프라이버시 문제를 해결하기 위한 방안으로 블록체인 통합 연구가 활발하게 진행되고 있다[8].

[9]는 모바일 클라우드 플랫폼에서 안전하게 EHRs (Electronic Health Record)을 공유하기 위하여 스마트 컨트랙트를 이용한 블록체인 기술을 연구하였다. [10]에서는 포그 컴퓨팅 기반 블록체인 기술을 이용하여 헬스케어 사물인터넷 인증 및 식별 기술을 제안하고 있다.

[11]은 SDN(Software Defined Networking)이 포함된 블록체인 기반 분산형 클라우드 아키텍처를 제안하였다. 포그 계층에 SDN 컨트롤러가 블록체인 기술을 사용하여 분산 방식으로 연결되어, 장치 이동성 및 흐름 규칙 분석 기능을 제공한다. 네트워크 에지에서 대규모 IoT 장치가 생성하는 원시 데이터 스트림을 효율적으로 관리하고, 고가용성, 실시간 데이터 전달, 높은 확장성, 보안, 탄력성 및 짧은 대기 시간을 지원한다.

[12]는 사물 인터넷의 데이터 관리를 위하여 사이트 체인과 컨소시엄 체인을 결합하기 위한 방안을 제시하고 있으며, [13]에서는 클라우드-포그 컴퓨팅에서 IoT 장치 관리와 사용자 인증을 위한 블록체인 기반 사용자 인증 방안을 제시하고 있다.

본 논문에서는 블록체인의 스마트 컨트랙트와 분산 원장을 사용하여 클라우드-포그 간에 의사결정 지원 기준을 공유하고, 포그-포그 간에 서비스 상호운용을 위한 시스템을 제안한다.

3. 하이퍼레저 패브릭

하이퍼레저 패브릭은 대표적인 허가형 블록체인으로 모듈형 구조로 설계되어 있어 확장성을

제공하며 전용 언어를 사용하지 않고, Go나 JAVA와 같은 표준 프로그래밍 언어로 작성된 분산 애플리케이션의 실행을 지원한다. 하이퍼레저 패브릭은 추가 전용 복제 원장(append only ledger) 데이터 구조를 사용해 실행 기록을 안전하게 추적하게 하며, 기존 블록체인과 달리 내장 암호화제를 사용하지 않는다. 또한, 모듈식 합의 프로토콜을 지원하므로 시스템을 특정 사용 사례 및 신뢰 모델에 맞게 조정할 수 있다[14].

하이퍼레저 패브릭을 구성하는 주요 컴포넌트는 다음과 같다.

- MSP(Membership Service Provider), Fabric CA : 시스템에 있는 모든 노드(클라이언트, Peer 및 Order)의 ID를 유지하고 인증 및 권한 부여에 사용되는 노드 자격 증명 발급을 담당한다. MSP, CA를 통해 허가형 블록체인의 특성을 유지하며, 노드 간의 모든 상호 작용은 디지털 서명을 사용하여 인증된 메시지를 통해 발생한다.

- Ordering Service : Orderer에 의해 수행되며, 트랜잭션을 순서대로 정리한 후, 블록을 생성하고 각 피어들에게 브로드캐스트한다.

- Peer : 하이퍼레저 패브릭 블록체인을 구성하는 네트워크 기본 요소로서, 블록체인 네트워크를 유지하고 트랜잭션의 제안 및 응답을 처리하며, 원장과 체인코드 관리 및 저장하는 역할을 수행한다.

- Chain Code : 하이퍼레저 패브릭의 스마트 컨트랙트이며, 블록체인 네트워크의 다양한 비즈니스 로직을 구현할 수 있다. 현재 Go, Java, Node 등 표준 프로그래밍 언어로 작성할 수 있지만 원장 상태에 직접 액세스 할 수는 없다.

- Ledger : 블록에 트랜잭션 정보가 실제로 저장되는 공간으로, 하이퍼레저 패브릭은 world state라는 저장소를 사용하여, 추가 전용 블록 체인 형태로 원장을 로컬로 유지하고 키-값 저장소에서 가장 최근 상태의 스냅 샷으로 유지한다.

- Channel : 개인 및 기밀 거래를 수행하기 위한 목적으로 둘 이상의 특정 네트워크 구성원 간의 통신을 위한 사설 "서브넷"이다. 네트워크의

각 트랜잭션은 채널에서 실행되며, 채널은 구성원(조직), 조직별 앵커 피어, 공유 원장, 체인코드 응용 프로그램 및 orderer에 의해 정의된다.

- Transaction : 클라이언트가 블록체인 네트워크의 원장을 읽거나 업데이트할 목적으로 특정 입력 매개 변수를 사용하여 체인코드 함수를 호출하기 위한 요청 메시지이다.

하이퍼레저 패브릭에서 클라이언트는 Dapp을 사용하여 Peer에게 연결하고 체인코드 ID를 포함한 트랜잭션 제안서를 만들어 제출한다. 이를 받은 Peer는 체인코드를 시뮬레이션하고 실행한 후 검증을 거쳐, 자신의 분산 원장을 업데이트하고, 제안서에 대한 응답을 클라이언트에게 전송한다. 클라이언트는 오더러에게 트랜잭션의 오더링을 요청하고, Orderer는 Peer의 서명을 검증한 후, 블록을 생성하여 이를 Peer에게 전송한다. Peer는 자신의 원장을 갱신한다.

III. 포그 기반 IoT 서비스 상호운용 시스템 설계

1. 시스템 구성

본 논문은 클라우드 환경에서 클라우드 서버, 포그, 서비스 사용자 간에 블록체인 네트워크를 통해 안전한 서비스 공유가 가능한 포그 기반 IoT 서비스 상호운용 시스템을 제안하였다.

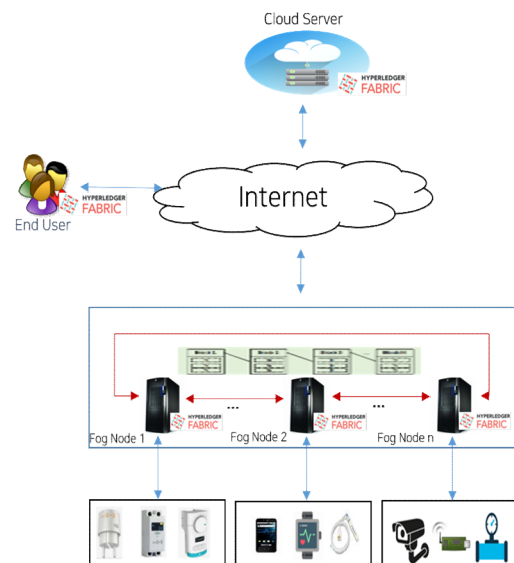


그림 1. 포그간 서비스 상호운용 시스템 구성도

그림 1은 제안하는 시스템의 구성을 보이고 있다. IoT 장치들로 구성된 IoT 장치 계층, IoT 장치와 연결되어 IoT 장치가 생성한 데이터 수집, 필터링 및 로컬 서비스를 생성하는 포그 계층 그리고 모든 서비스 및 사용자를 관리하는 클라우드 계층으로 구성되며, 서비스 요청을 위한 최종 사용자를 포함한다.

가. IoT 장치

사물 클라우드 서비스를 위한 데이터를 생성하는 IoT 장치 및 스마트 기기들로 구성되는 계층으로, 센서, 액추에이터, 그리고 모바일 장치들이 포함된다. 응용 분야에 따라, 도로, 의료 센터 및 농장을 비롯한 다양한 환경에 사물 인터넷(IoT) 센서가 사용될 수 있다.

각 IoT 장치는 포그에 연결되어 기기 인증을 수행한 후, 생성된 데이터를 포그 노드에 전송한다. IoT 장치는 공개성/비밀성 여부를 포함하여 분류 규칙에 의해 분류된 데이터를 전송한다. IoT 장치와 포그 노드간 데이터의 기밀성을 보장하기 위하여 AES 암호화 알고리즘을 사용하여 데이터를 암호화하여 전송한다.

나. 포그 노드

포그 노드는 IoT 장치와 클라우드 사이에 존재하며, IoT 장치가 전송하는 데이터를 저장 및 처리한다. 포그 노드는 로컬에서 처리할 데이터의 저장 및 처리, 로컬 서비스 요구에 대한 데이터 전송 등을 수행한다. 또한, 클라우드 서버에 데이터 업로딩 및 로컬 IoT 장치에 대한 컨트롤러로서 동작한다.

포그 노드는 데이터 분석을 통한 서비스를 제공하기 위하여 IoT 장치 그룹을 감독하고 관리하며, IoT 장치가 생성한 데이터에 대한 데이터 필터링 및 데이터 분석을 수행하여 서비스를 생성한다.

본 논문에서 포그 노드의 주된 기능은 클라우드 서비스의 지역적 분산 서비스를 수행하며, 클

라우드로부터 위임받은 서비스를 수행한다. 클라우드가 생성한 데이터 필터링 및 서비스 생성 규칙에 따라 서비스를 생성하고, 포그간에는 서비스 상호운용이 가능하다. 이를 통하여 IoT 장치와 가까운 곳에서 로컬화된 저장 및 처리를 수행하여 클라우드에서 IoT 장치까지의 대기 시간과 응답 시간을 줄일 수 있도록 한다.

다. 클라우드 서버

클라우드 서버는 응용 계층의 서비스 요구에 의해 데이터를 제공하며, 분산 컴퓨팅 및 스토리지를 제공하여 광역 모니터링, 제어, 대규모 이벤트 감지, 행동 분석 및 장기 패턴 인식을 제공할 수 있다.

본 논문에서 클라우드 서버는 모든 서비스에 대한 생성 및 접근 권한을 가지고 있으며, 포그 노드에 서비스를 위임할 수 있다. 클라우드 서버는 포그 노드에서 데이터 필터링 및 로컬 서비스를 생성하기 위한 의사 결정 기준을 포그 노드에 송신한다. 클라우드 서버가 포그 노드에 전송하는 의사 결정 기준을 Support Rule로 정의하고, 이를 블록체인의 스마트 컨트랙트를 통해 공유하도록 한다. 이를 사용하여 빠른 지연시간이 요구되거나, 기밀성이 요구되는 데이터에 대해서는 포그에서 서비스를 생성하여 관리할 수 있도록 한다.

라. 최종 사용자

최종 사용자는 IoT 서비스를 사용하기 위한 인증된 참여자로, 클라우드를 통해서 IoT 서비스를 제공받거나, 포그 노드를 통해서 서비스를 제공할 수 있다. 포그 노드의 로컬 서비스를 사용하기 위해서 블록체인의 스마트 계약에서 액세스 권한을 요청하는 사용자이다. 인증서를 발행받은 최종 사용자는 스마트 계약을 통해 서비스에 대한 액세스 권한을 부여받고, 대상 서비스를 제공하는 지정된 포그 노드에 연결하여 서비스를 제공할 수 있다.

2. IoT 서비스 상호운용 시스템 설계

클라우드 환경에서 포그 노드는 지리적으로 분산될 뿐만 아니라, 서로 다른 스마트 환경에 배치되기 때문에 특정 서비스 제공을 위해서는 포그 간 상호운용이 필요하며, 서로 다른 도메인에서 적용 가능하도록 구성되어야 한다. 이를 위하여 인증된 클라우드 서버 및 포그 노드간 서비스 생성 및 등록정보, 서비스 권한, 그리고 서비스 사용 등의 정보를 블록체인을 사용하여 공유함으로써, 포그 노드간 서비스 상호운용이 가능하도록 설계한다.

본 시스템은 클라우드 환경에서 포그 기반 IoT 서비스 상호운용 시스템을 구성하며, 포그-서비스 맵핑 정보, 사용자 권한 정보 등을 블록체인 네트워크를 통해 공유하고 권한에 따라 IoT 서비스에 접근할 수 있다.

포그 서비스 상호운용 설계를 설명하기 위해 본 논문에서 사용하는 용어에 대해 다음과 같이 정의하였다.

[정의1. Support Rule(SR)]

클라우드 서버가 포그 노드에 전송하는 의사결정 기준 규칙을 Support Rule로 정의한다. 포그 노드가 데이터 필터링을 수행하기 위한 규칙과 포그에서 수행할 서비스를 생성하기 위한 서비스 생성 규칙, 그리고 임계값 등을 설정한다.

[정의2. Local Service(LS)]

IoT 장치가 생성한 데이터를 클라우드까지 전송하여 처리하지 않고, IoT 장치와 지리적으로 가까운 위치에 있는 포그 노드에서 클라우드가 설정한 Support Rule에 의해 데이터 분석 및 처리를 수행하기 위하여 생성한 서비스를 말한다.

[정의3. 서비스 권한 토큰]

포그 또는 서비스 요청자에게 로컬 서비스의 사용 권한을 발행한 인증서를 의미하며, 클라우드의 비밀키로 서명된 토큰이다.

IoT 서비스를 클라우드에서 모두 처리하지 않고 포그에 분산하여 처리하는 서비스 상호운용 방

법은 그림 2에 제시한다.

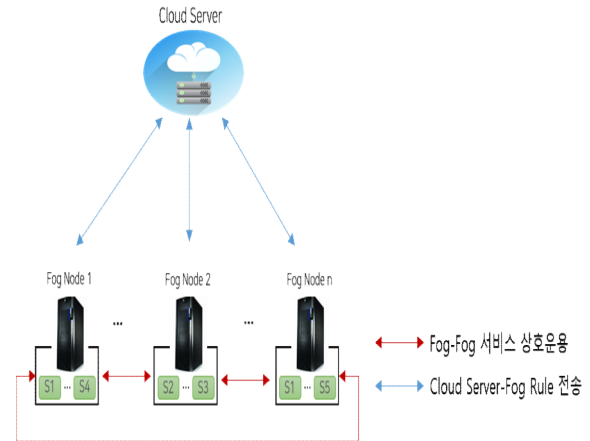


그림 2. 클라우드 서버-포그-포그간 시스템 흐름

클라우드가 규칙으로 정의한 로컬 서비스인 개인정보 필터링, 하한값 필터링, 모니터링, 알람, 제어 신호 발생들은 포그 노드가 구현한다. 클라우드와 포그 간에는 다음 과정을 통하여 서비스 상호운용이 가능하다.

- 클라우드 서버는 포그 노드가 데이터 필터링 및 로컬 서비스를 생성하기 위한 Support Rule을 포그 노드에 전송한다.
- 포그 노드는 이를 사용하여 데이터를 필터링하거나, 로컬 서비스를 생성한다.
- 포그-포그간 로컬 서비스를 상호운용하여 서로 다른 도메인간 특정한 서비스를 사용할 수 있다.

클라우드 포그 간에는 블록체인을 활용하여 서비스 상호운용을 위한 분산 원장 및 스마트 컨트랙트를 그림 3과 같이 공유한다. 분산 원장에는 클라우드가 생성한 Support Rule, 포그-서비스 등록정보, 서비스 권한 토큰 발행 정보가 저장된다. 스마트 컨트랙트에는 각 분산 원장에 대한 정보를 처리하기 위한 함수를 정의하였다.

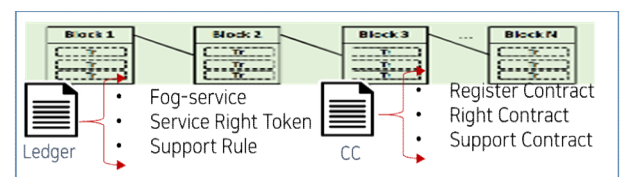


그림 3. 블록체인의 분산 원장과 스마트 컨트랙트

서비스 상호운용을 위한 시스템의 프로세스는 크게 블록체인 네트워크 구성 및 체인코드 설치 단계, 로컬 서비스 생성 단계, 서비스 권한 토큰 발행 그리고 서비스 상호운용 단계의 4단계로 나눌 수 있다.

[1단계] 블록체인 네트워크 구성 및 스마트 컨트랙트 정의 : 클라우드 서버는 포그-서비스 등록, Support Rule, 토큰 발행을 위한 스마트 컨트랙트를 정의하여 체인코드를 생성한다.

[2단계] 로컬 서비스 생성 : 포그 노드는 블록체인 네트워크를 통해 Support Rule에 따라 로컬 서비스를 생성한다. 생성된 서비스 등록을 위해 블록체인 네트워크에 포그-서비스 등록 트랜잭션을 제출하고 포그-서비스 맵핑 정보를 블록체인에 공유한다. 그림 4.a)는 로컬 서비스 생성과정을 보인다.

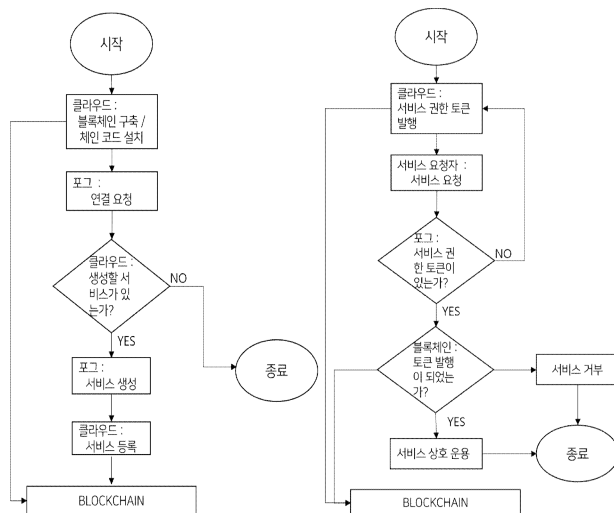


그림 4. a)로컬 서비스 생성 b) 서비스 상호운용

[3단계] 서비스 권한 토큰 발행 : 포그 노드의 로컬 서비스를 사용하고자 하는 서비스 요청자(최종 사용자 또는 포그 노드)는 블록체인을 통해 등록된 포그-서비스를 요청하고, 분산 원장에 공유된 접근 권한 정보에 따라 포그-서비스에 대한 서비스 권한 토큰을 받는다.

[4단계] 서비스 상호운용 : 서비스 요청자(포그 노드, 최종 사용자)는 서비스 권한 토큰을 블록체인에 제출하여 포그 노드 또는 포그 노드 리스트

에 대해 IoT 서비스를 제공 받는다. 서비스 상호운용을 위한 과정은 그림 4.b)와 같다.

가. 블록체인 네트워크 구축 및 체인코드 생성

서비스 상호운용 시스템 프로세스의 1단계는 블록체인 네트워크 구축 및 체인코드 생성단계이다.

본 논문에서는 허가형 블록체인 플랫폼인 하이퍼레저 패브릭을 사용한다. 블록체인 네트워크에 참여하는 클라우드 서버, 포그 노드 그리고 최종 사용자는 하이퍼레저 패브릭 네트워크(HFN, HyperLedger Fabric Network)에서 클라이언트로 동작한다. 각 클라이언트는 Dapp을 이용하여 HFN의 CA에게 인증서를 발급받고, 네트워크에 참여한다.

본 시스템에서 체인코드는 IoT 서비스 공유를 위한 서비스 등록, 접근 권한 토큰 발행 및 SR 관리에 대한 스마트 컨트랙트들로 그림 5와 같이 ServiceShareCC로 정의하며, Register Contract, Right Contract, Support Contract를 포함한다.

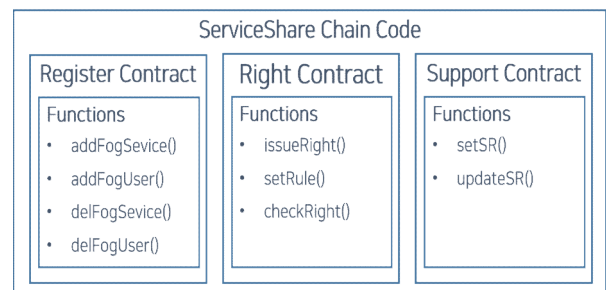


그림 5. ServiceShare Chain code

• Register Contract

포그 노드의 서비스 등록 요청을 받아 클라우드 서버의 요청에 의해 실행되며 포그-서비스 등록 및 제거, 포그-사용자 등록 및 제거를 위한 컨트랙트이다.

- addFogService() : 포그와 로컬 서비스 맵핑 정보를 등록한다.
- addFogUser() : 포그 내 최종 유저 정보를 등록한다.

- delFogService() : 포그가 생성한 로컬 서비스를 제거한다.

- delFogUser() : 포그 내 최종 유저 정보를 제거한다.

- Right Contract

로컬 서비스를 사용하기 위한 사용자의 서비스 권한 토큰을 발행하고, 관리하기 위한 체인코드이다. 본 논문에서 서비스 권한은 읽기, 제어, 포그 그룹에 요청 가능여부에 따라 사용권한이 적용되며 이에 따라 서비스 권한 토큰이 발행된다.

- issueRight() : 서비스 요청자에 대한 서비스 접근 권한이 부여된 토큰을 발행한다.

- checkRight() : 서비스 요청자가 제출한 토큰의 유효성을 검증한다.

- setRule() : 서비스에 대한 읽기와 쓰기 권한 뿐만 아니라, 포그 그룹에 대한 서비스 요청 권한 등을 설정할 수 있으며, 최종 사용자의 경우 포그 내/외부 접근에 따른 접근 권한 등을 설정하기 위한 접근 룰을 설정한다.

- Support Contract

클라우드 서버가 포그 노드에게 위임할 서비스 및 데이터 필터링을 위한 SR(Support Rule)을 설정하고, 사용하기 위한 컨트랙트이다.

- setSR() : 클라우드 서버에서 포그 노드에게 의사 결정 기준을 제공하기 위한 룰을 설정하며, 위임할 서비스와 데이터 필터링 규칙을 설정할 수 있다.

- updateSR() : 포그 노드에서 로컬 서비스 생성 및 로컬 데이터 처리를 위해 데이터 필터 룰을 적용하는 메소드이다.

블록체인 네트워크에서 클라이언트가 Dapp을 통해 peer에게 트랜잭션을 제출하면, 체인코드의 해당 스마트 컨트랙트가 실행된다.

나. 로컬 서비스 생성과 등록

서비스 상호운용 시스템 프로세스의 2단계는 로컬 서비스 생성과 등록단계이며, 클라우드 서버는 로컬 서비스 생성을 위한 SR을 정의한다.

(1) 클라우드의 SR 생성

클라우드 서버는 모든 서비스에 대한 생성 및 접근 권한을 가지고 있으며, 포그 계층에 서비스를 위임할 수 있다. 클라우드 서버는 포그 노드에서 데이터 필터링 및 로컬 서비스를 생성하기 위한 SR을 스마트 컨트랙트를 통하여 생성하고, 이를 블록체인을 통해 공유한다. SR은 클라우드 서버의 필요에 의해 추가되거나, 갱신될 수 있다. 클라우드가 생성하는 SR은 데이터 필터링 SR과 로컬 서비스 생성 SR로 구분한다.

본 논문에서는 SR을 정의하기 위해 ECA(Event-Condition -Action) Rule 표기를 따른다. 사건-조건-행동(ECA)은 지식 표현 구조의 패턴으로 특징지어지는 유명한 규칙 체계 중 하나로, 각 ECA 규칙은 이벤트, 조건 및 작업의 세 부분으로 구성된다[15].

- 데이터 필터링 SR

IoT 장치가 생성한 모든 데이터를 클라우드로 전송하지 않고, 클라우드가 제시한 임계치에 도달하지 않은 데이터 필터링 그리고 개인정보가 포함된 데이터 필터링 기능을 수행하기 위한 규칙이다. 익명성이 요구되는 데이터에 대해 개인정보가 포함되지 않도록 위치정보 및 사용자 통계 정보를 제거한 후 클라우드로 전달한다. 데이터 필터링 SR은 클라우드가 위임한 서비스의 결과값에 대해서도 적용할 수 있다.

[규칙1]은 전송하려는 데이터에 IP, 주민등록번호, 주소 등의 정보가 포함되어있는 경우, 데이터를 필터링한 후 전송하기 위한 규칙이다.

[규칙1] Privacy_Rule

전송하려는 데이터에 개인을 식별할 수 있는 정보가 포함되어 있는 경우, 이를 제거하고 전송하는 규칙

ON : Insert(Data)

IF : Data.in(IP, Personal_ID, Address)

DO : Filtered(Data)

[규칙2]는 전송하려는 데이터가 클라우드 서버가 지정한 시스템 하한값에 해당하는 경우에 데이터를 폐기하기 위한 규칙이다.

[규칙2] Discard_Rule

전송하려는 데이터가 클라우드 서버가 지정한 하한값 이하에 해당되는 경우, 데이터를 폐기하는 규칙

ON : Insert(Data)

IF : (Data<=System.infimum)

DO : Discard(Data)

- 로컬 서비스 생성 SR

클라우드 서버는 포그 노드에서 IoT 장치로부터 생성된 데이터에 대해 데이터 분석 및 통계 처리를 통해 모니터링, 제어 또는 이벤트 서비스를 생성할 수 있도록 MonitorService_Rule, ControlService_Rule 그리고 AlarmService Rule 을 생성한다. 포그에 연결된 장치가 전송한 데이터에 대한 통계 처리를 모니터링 하는 서비스를 생성하기 위한 규칙, 특정 장치에 대한 제어 정보를 전송하기 위한 서비스를 생성하기 위한 규칙, 특정 장치에서 이상 데이터 발생 시 알람 서비스를 생성하기 위한 규칙 등을 작성할 수 있다.

[규칙3]은 포그에 연결된 장치를 모니터링하기 위한 서비스를 생성하는 규칙이다. 장치 목록과 시간 정보가 입력되면 이에 대한 통계 정보를 계산하는 서비스를 생성할 때 사용한다.

[규칙3] MonitorService_Rule

IoT 장치에 대한 일정 시간 데이터 통계값을 모니터링 하기 위한 서비스를 생성하는 규칙

On : Insert(DeviceID[], startTime, endTime)

Do : Calculate(dataField)

[규칙4]는 포그 내에서 특정 장치에 대한 제어 신호 및 액추에이터 동작이 필요한 경우에 특정 신호를 전송하기 위한 서비스를 생성하기 위한 규칙이다.

[규칙4] ControlService_Rule

특정 장치에 대한 제어 신호를 전송하기 위한 서비스를 생성하는 규칙

On : Alarm(Device)

If : Alarm(Device).count>=System.threshold

Do : Control_Signal(Device)

[규칙5]는 로컬 내에서 즉각적인 알람이나 경고 등과 같이 실시간 처리가 필요한 경우 이벤트 알람과 같은 서비스를 생성하기 위한 규칙이다.

[규칙5] AlarmService_Rule

이상 데이터 발생시 알람 서비스를 생성하는 규칙

On : Insert(Device.dataField)

If : Device.dataField>=System.supremum

Do : Alarm(Device)

본 논문에서는 하한값 필터링, 개인정보 필터링, 모니터링 서비스, 컨트롤 서비스, 알람 서비스 등 5개의 규칙을 생성하였다.

(2) 로컬 서비스 생성

클라우드 서버는 인증된 포그가 연결되면, 포그에게 위임할 서비스를 검색한다. 그리고 해당 서비스를 생성하기 위한 SR을 갱신할 수 있도록 스마트 컨트랙트 연결을 리다이렉트 한다. 해당 포그 노드에 대해 새로운 SR이 필요한 경우는 블록체인에 setSR() 트랜잭션을 제출하여 SR을 설정할 수 있다.

포그 노드는 updateSR()을 통해 SR을 다운로드한 후, 로컬 서비스를 생성한다. 그림 6은 로컬 서비스 생성 절차를 나타낸다.

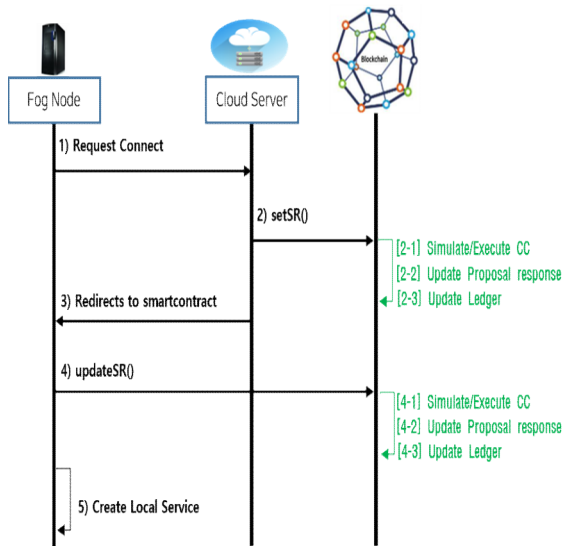


그림 6. 로컬 서비스 생성 절차

포그 노드는 클라우드 서버가 생성한 스마트 컨트랙트를 사용하여 SR을 업데이트하고, 물에 따라 데이터 분석 및 처리를 수행하여 로컬 서비스를 생성한다.

로컬 서비스는 IoT 장치로부터 생성된 데이터에 대한 필터링 및 분석한 정보를 제공하는 모니터링 서비스, IoT 장치의 기능제어를 위한 제어 서비스, 특정 상황에 대한 경고 및 알람을 발생하는 알람 서비스로 구성될 수 있다.

포그 노드는 SR을 이용하여 로컬 서비스를 생성하고, 서비스와 IoT 장치를 결합하여 로컬 데이터베이스에 다음과 같이 관리할 수 있다.

$LS_n : \{S_n, Device_1, [Device_n]\}$

스마트 홈 내에서는 다음과 같은 로컬 서비스를 생성할 수 있다.

$LS_1 : \{\text{일별 소비 전력량 모니터 서비스, 스마트콘센트1}\}$

$LS_2 : \{\text{사용량 제한 알람 서비스, 스마트콘센트2}\}$

$LS_3 : \{\text{전력 차단 컨트롤 서비스, 스마트콘센트2}\}$

스마트 의료 IoT 데이터의 경우, 이상 수치 발생 시 “즉시 알람 서비스”와 같이 실시간 처리가 필요한 서비스를 로컬 서비스로 생성하여 포그 노드에서 바로 서비스를 실행할 수 있다.

(3) 로컬서비스(LS) 및 사용자 등록

생성된 로컬 서비스는 스마트 컨트랙트를 사용하여 포그-서비스 정보를 등록하여 공유한다.

그림 7은 블록체인을 통해 스마트 컨트랙트가 호출되고, 포그-서비스와 포그-사용자가 등록되는 절차를 보인다.

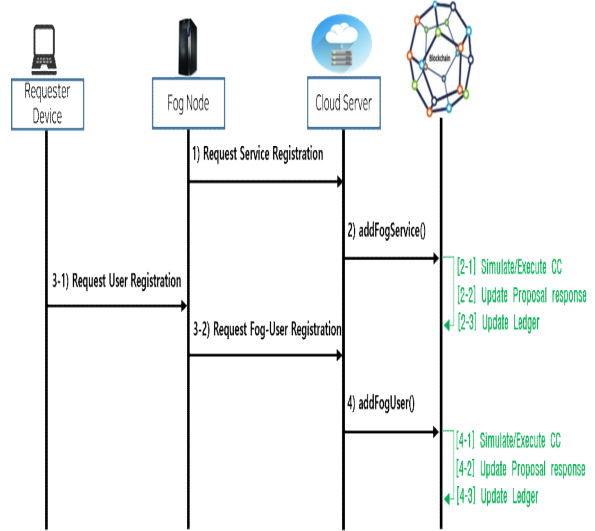


그림 7. 서비스 및 사용자 등록 절차

포그 노드가 생성한 로컬 서비스는 클라우드 서버가 생성한 컨트랙트의 `addFogService()`를 호출하여 등록한다.

스마트 컨트랙트가 실행된 후, 분산 원장에 저장되는 포그-서비스 정보의 구조는 그림 8과 같다.

```

"FogService":
{
  "FS_ID": "",
  "FogID": "",
  "ServiceID": ""
  "RuleID": ""
  "Comment": ""
}
  
```

그림 8. 포그-서비스 등록정보 구조

포그-서비스 등록정보에는 로컬서비스를 생성한 RuleID를 포함하며, Comment는 서비스에 대한 설명을 기록한다.

다. 서비스 권한 토큰 발행

서비스 상호운용 시스템 프로세스의 3단계는 서비스 권한 토큰 발행단계이다.

본 논문에서 서비스 권한 토큰은 포그-포그간 서비스 상호운용을 위해 포그 또는 최종 사용자에게 발급되는 인증서를 말한다.

그림 9는 서비스 요청자에게 발행되는 서비스 권한 토큰의 구조를 보인다. 서비스 권한 토큰은 PKI에서 사용하는 표준 인증서 형식인 X.509 구조를 기반으로 JSON을 사용하여 다음과 같이 정의하였다. 서비스 권한 토큰 발행 여부는 블록체인의 분산 원장에 저장되고 블록체인 네트워크 참여자에게 공유된다.

```

"ServiceRightToken":
{
  "TokenID" : " ",
  "FS_ID" : [ " " ],
  "RF_ID" : " ",
  "Right" : { "AC": " ", "SG": " " },
  "ITIME" : " ",
  "ETIME" : " ",
  "Sig" : " "
}

```

그림 9 서비스 권한 토큰 구조

- TokenID : 발행된 토큰에 부여하는 토큰 식별자이다.
- FS_ID : 클라우드에 등록된 서비스 대상이 되는 로컬 서비스 식별자이다. FS_ID는 하나 이상의 포그가 포함될 수 있다.
- RF_ID : 서비스 요청 포그 식별자를 나타낸다.
- Right : 서비스에 대한 접근 권한을 나타낸다.
 - AC : 실제 동작으로 권한(Read, Write 등)을 나타낸다.
 - SG : 단일 포그 또는 포그 그룹에 해당 서비스 권한 여부를 나타낸다.
- ITIME : 토큰 발행시간 값을 나타낸다.
- ETIME : 토큰 만료시간 값을 나타낸다.
- Sig : 토큰에 대한 무결성을 보장하기 위한 토큰을 발행한 클라우드 노드의 서명값(해시)을 갖는다.

그림 10은 서비스 권한 토큰 발행 절차를 보인다.

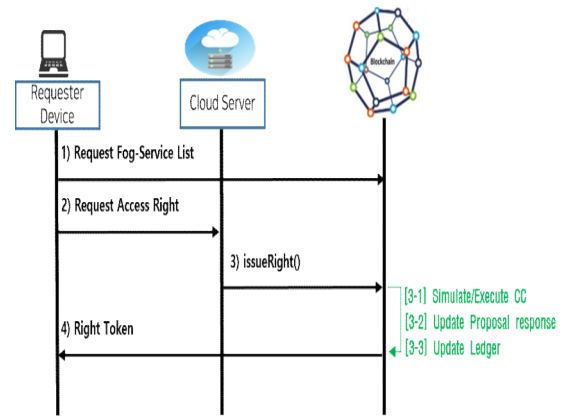


그림 10. 서비스 권한 토큰 발행 절차

서비스 요청자는 블록체인을 통해 포그-서비스 리스트를 확인한 후, 클라우드에 특정 포그에 대한 서비스 권한을 요청한다. 클라우드 서버는 인증된 서비스 요청자 정보를 확인한 후 토큰을 발행한다. 토큰 발행 정보는 분산 원장에 저장된다.

라. 로컬 서비스 상호운용

서비스 상호운용 시스템 프로세스의 4단계는 포그에서 생성한 IoT 서비스를 포그-포그 간에 상호운용하는 단계이다. 분산된 포그에서 실행되는 클라우드의 대리 서비스를 서로 다른 포그 간에 운용 도메인과 상관없이 클라우드에서 발행된 서비스 사용 권한 토큰을 이용하여 포그 간에 서비스를 상호운용할 수 있다.

그림 11은 서비스 권한 토큰을 가진 서비스 요청자가 포그 노드 또는 포그 노드 리스트에 서비스를 요청하고, 해당 서비스를 사용하는 절차를 보이고 있다.

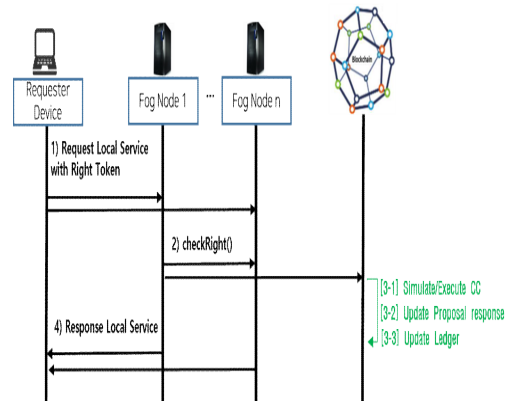


그림 11. 포그간 서비스 상호운용 절차

서비스 요청자는 해당 서비스에 대한 서비스 권한 토큰을 가지고 포그에 서비스를 요청한다. 포그 또는 포그 리스트는 블록체인의 분산 원장을 사용하여 권한을 체크한 후 서비스를 제공한다. 블록체인을 통해 등록된 포그의 IoT 서비스는 클라우드의 개입 없이 로컬에서 실행되고, 다른 포그 간에도 상호운용될 수 있다.

IV. 포그 기반 IoT 서비스 상호운용 시뮬레이션

1. 시스템 시뮬레이션

제안한 시스템 시뮬레이션을 위하여 사용한 오픈소스 블록체인 프레임워크는 Hyperledger Fabric v2.0이며, Hyperledger Node.js SDK를 이용해 구현하였다. 개발 OS는 Ubuntu 18.04 LTS, 사용된 도구는 Node.js 14.15.1, Docker 19.03.13, Docker-compose 1.8, npm 6.4, git 2.17.1, Python 2.7.4, VScode다.

블록체인 네트워크는 그림 12와 같이 클라우드-포그 간 SR 전송, 포그-포그 간 서비스 상호운용 테스트를 위해 클라우드 조직과 포그 조직으로 두 개의 조직을 구성한다. 각각은 채널을 통해 스마트 컨트랙트와 원장을 공유한다. 각 조직별 두 개의 피어를 구성하였고, CA를 통해 포그와 클라우드는 인증서를 발급받는다. 클라우드는 포그는 Dapp을 이용하여 하이퍼레저 패브릭 네트워크에 트랜잭션을 전송하고, 응답을 받는 클라이언트로 동작한다.

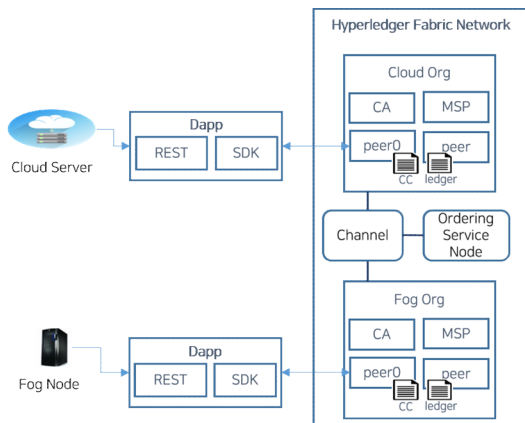


그림 12. 하이퍼레저 패브릭 네트워크 구조

클라이언트는 Dapp을 통해 트랜잭션을 제출하여, 포그-서비스 등록, SR 갱신 및 서비스 권한 토큰을 발행하며 트랜잭션의 결과는 분산 원장에 저장, 공유된다.

2. 로컬 서비스 생성 및 서비스 시나리오

본 논문에서 시스템 테스트를 위해 스마트 에너지 IoT 장치들과 연결된 포그 노드에서 장치의 소비 전력량 관리를 위한 서비스 상호운용 구현 시나리오로 진행하였다.

Support Contract의 setSR()에 대한 트랜잭션을 제출하여 다음과 같이 SR을 설정하였다.

가. 소비 전력량 모니터 서비스 생성

소비 전력량 모니터 서비스는 장치 또는 장치 목록과 일정 시간 정보를 이벤트로 입력받아서, 소비 전력량을 계산하는 서비스이다.

클라우드 서버가 setRule() 트랜잭션을 사용하여 정의한 MonitorService _Rule은 그림 13과 같이 분산 원장에 저장된다.

```
{ "id": "MS_Rule1", "rulename": "MonitorService_Rule", "on": "Insert(DeviceID[], startTime, endTime", "iif": "", "do": "calculate(dataField)" }
```

그림 13. setRule() 실행 후, MS_Rule1 분산 원장

포그는 updateSR() 트랜잭션을 사용하여 정의된 MS_Rule1을 다운로드하여 LS001 서비스를 생성한다.

생성한 포그 서비스는 addFogService()에 대한 트랜잭션을 제출하여 등록하고, 등록한 후 분산 원장은 그림 14와 같다.

```
{ "id": "FarmFog1LS001", "fog_id": "FarmFog1", "service_id": "LS001", "ruleid": "MS_Rule1", "comment": "averageperday_temperature" }
```

그림 14. LS001 서비스가 등록된 분산 원장

FarmFog1이 LS001 서비스를 MS_Rule1에 의해 생성하였으며, 이 서비스는 온도에 하루 평균 값을 모니터링하는 서비스임을 알 수 있다.

나. 전력 차단 제어 서비스 생성

전력 차단 제어 서비스는 장치가 소비 전력량 경고 알람을 받게 되면, 알람 받은 횟수를 카운트 하여, 동일한 알람에 대한 횟수가 시스템에서 설정한 최대 알람 횟수 이상이면, 해당 장치에 대한 전원을 차단할 수 있는 서비스로 ControlService_Rule을 적용하여 로컬 서비스를 생성한다.

클라우드 서버가 정의한 ControlService_Rule은 그림 15와 같이 분산 원장에 저장된다.

```
{ "id": "CS_Rule1", "rulename": "ControlService_Rule", "on": "OnAlarm(Device)", "iif": "Alarm(Device)\u003e=System.threshold", "do": "TurnOff(Device)" }
```

그림 15. setRule() 실행 후, CS_Rule1 분산 원장

포그는 updateSR() 트랜잭션을 사용하여 정의된 CS_Rule1을 다운로드하여 LS002 서비스를 생성한다. LS002 서비스가 등록된 후 분산 원장은 그림 16과 같다.

```
{ "id": "FarmFog1CS001", "fog_id": "FarmFog1", "service_id": "LS002", "ruleid": "CS_Rule1", "comment": "control_turnoff" }
```

그림 16. LS002 서비스가 등록된 분산 원장

FarmFog1이 LS002 서비스를 CS_Rule1에 의해 생성하였으며, 이 서비스는 전력차단 신호를 보내는 서비스임을 알 수 있다.

다. 전력 소비 경고 알람 서비스 생성

전력 소비 경고 알람 서비스는 소비 전력량이 입력되면, 장치의 소비 전력량이 시스템이 설정한 상한값 이상인지를 판단하여 상한값 이상인 경우, 전력소비량 경고 알람을 발송하기 위한 서비스로 AlarmService_Rule을 적용하여 로컬 서비스를 생성한다.

클라우드 서버가 정의한 AlarmService_Rule은 그림 17과 같이 분산 원장에 저장된다.

```
{ "id": "AS_Rule1", "rulename": "AlarmService_Rule", "on": "Insert(Device.PowerConsumption)", "iif": "Device.PowerConsumption\u003e=System.supremum", "do": "Alarm(Device)" }
```

그림 17. setRule() 실행 후, AS_Rule1 분산 원장
포그는 updateSR() 트랜잭션을 사용하여 정의된 AS_Rule1을 다운로드하여 LS003서비스를 생성한다. LS003 서비스가 등록된 후 분산 원장은 그림 18과 같다.

```
{ "id": "FarmFog1AS001", "fog_id": "FarmFog1", "service_id": "LS003", "ruleid": "AS_Rule1", "comment": "alarm_powerconsumption" }
```

그림 18. LS003 서비스가 등록된 분산 원장

FarmFog1이 LS003 서비스를 AS_Rule1에 의해 생성하였으며, 이 서비스는 소비 전력량 경고 알람 서비스임을 알 수 있다.

라. 서비스 사용

포그 노드는 로컬 서비스를 생성하여, 포그와 직접 연결된 지역 내 장치로부터의 서비스 요청과 다른 포그 노드의 서비스 요청에 대해 해당 서비스를 제공한다. 다른 포그 노드에서 정의한 서비스를 사용하고자 하는 포그 노드에서는 블록체인에 트랜잭션을 제출하여, 전체 포그-서비스 목록을 확인할 수 있다.

그림 19는 시뮬레이션 시스템에 등록된 모든 포그-서비스 목록을 나타낸다.

```
[{"id": "AS_Rule1", "rulename": "AlarmService_Rule", "on": "Insert(Device.PowerConsumption)", "iif": "Device.PowerConsumption\u003e=System.supremum", "do": "calculate(dataField)"}, {"id": "CS_Rule1", "rulename": "ControlService_Rule", "on": "OnAlarm(Device)", "iif": "Alarm(Device)\u003e=System.threshold", "do": "TurnOff(Device)"}, {"id": "MS_Rule1", "rulename": "MoniterService_Rule", "on": "Insert(DeviceID[], startTime, endTime)", "iif": "", "do": "calculate(dataField)"}]
```

그림 19. 시뮬레이션 시스템 등록된 서비스 목록

포그 노드는 사용하고자 하는 서비스가 있는 경우, 클라우드를 통해 서비스 권한 토큰을 발급 받아서 사용할 수 있다. 토큰이 발행된 후 토큰 발행 정보는 그림 20과 같이 원장에 저장된다.

```
[{"id": "SRT0001", "fs_id": "FarmFog1LS001", "rf_id": "EnergyFog1", "itime": "2021-12-01"}]
```

그림 20. 토큰 발행 정보 분산 원장

서비스 사용자는 발행한 토큰을 이용하여 포그 노드에서 생성한 로컬 서비스를 사용할 수 있다.

본 논문에서는 제안한 시스템의 설계를 검증하기 위해서 클라우드 서버와 포그 노드로 구성된 사물 인터넷 서비스를 시뮬레이션하였으며, 분산 원장을 이용하여 정보를 공유하고 이를 통한 서비스 공유가 가능한지를 테스트하였다.

클라우드 서버는 블록체인의 스마트 컨트랙트를 사용하여 포그 노드에게 SR을 공유하여 포그 노드가 로컬 서비스를 생성하도록 함으로써, 서비스를 위임할 수 있으며, 포그 노드는 서비스 권한 토큰을 사용하여 각 포그 간 서비스를 상호운용할 수 있음을 보였다.

V. 결 론

본 논문은 사물 클라우드에서 빠른 지역적 데이터 서비스 및 안전한 데이터 공유를 위하여 클라우드 환경에서 블록체인을 사용하여 포그 기반 IoT 서비스 상호운용 시스템을 제안하였다.

기존 클라우드 기반 IoT 서비스는 대용량의 데이터 전송 및 처리, 전송 지연, 단일 지점 장애, 개인 정보 보안 관련 취약점을 가지고 있다. 이를 해결하기 위한 방안으로 클라우드와 IoT 장치 계층 사이에 포그 노드를 두어 클라우드가 제공해야 할 서비스를 포그 노드로 분산 관리하며, 클라우드를 거치지 않고도 서로 다른 도메인을 사용하는 포그 간에도 서비스 상호운용이 가능하도록 설계하였다.

본 논문에서 접근제어는 IoT 기기별 접근제어가 아니라, IoT 데이터에서 생성된 데이터에 대한 서비스 단위로 접근제어를 수행하며, IoT 기기에 대한 인증은 포그에서 이루어지며, 포그에서 필터링을 통해서 보안이 적용된 IoT 데이터

서비스를 제공함으로써 개인정보보호 및 IoT 데이터의 무분별한 사용을 제한할 수 있다. 또한, 클라우드가 위임한 서비스는 분산된 포그 노드에서 수행되고, 블록체인을 이용하여 인증된 포그 노드들 사이에 서비스 상호운용이 가능하였다.

본 연구에서는 서로 다른 도메인 간에 포그 노드를 통해 서비스를 상호운용하기 위한 방안을 제시하였으며, 블록체인 네트워크의 분산 원장을 이용하여 스마트 팜과 스마트 에너지 간 정보를 공유하고 이를 통한 서비스 공유가 가능함을 보였다. 클라우드 서버와 포그 노드는 블록체인을 사용함으로써 안전하게 SR을 공유하고, SR의 추가 및 갱신도 용이하게 수행할 수 있다. 블록체인을 이용하여 응용 도메인에 따라 SR을 확장함으로써 다양한 사물 클라우드 응용 분야에 적용할 수 있을 것으로 기대된다.

추후, 분산 클라우드와 분산 포그 그리고 IoT 장치 계층을 멀티 블록체인으로 연결하여, 다양한 응용 서비스간 이기종 블록체인 상호운용 시스템에 대한 연구를 진행하고자 한다.

REFERENCES

- [1] D. C. Nguyen, P. N. Pathirana, M. Ding and A. Seneviratne, "Integration of Blockchain and Cloud of Things: Architecture, Applications and Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2521-2549, 2020.
- [2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," *MCC Wksp. Mob. Cloud Comput.*, pp. 13-16, Aug. 2012.
- [3] A. V. Dastjerdi and R. Buyya, "Fog Computing: Helping the Internet of Things Realize Its Potential," *Computer*, vol. 49, no. 8, pp. 112-116, Aug. 2016.
- [4] I. Stojmenovic and S. Wen, "The Fog computing paradigm: Scenarios and security issues," *2014 Federated Conference on Computer Science and Information Systems*, pp. 1-8, Warsaw, Poland, Oct. 2014.
- [5] H. Gupta, A. V. Dastjerdi, S. K. Ghosh and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of

- Things, Edge and Fog computing environments," *SOFTWARE-PRACTICE & EXPERIENCE*, vol. 47, no. 9, pp.1275-1296, Jun. 2017.
- [6] H. Dubey, A. Monteiro, N. Constant, M. Abtahi, D. Borthakur, L. Mahler, Y. Sun, Q. Yang, U. Akbar and K. Mankodiya, "Fog Computing in Medical Internet-of-Things: Architecture, Implementation, and Applications," *Handbook of Large-Scale Distributed Computing in Smart Healthcare*, Springer, pp. 11-25, Aug. 2017.
- [7] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, part 2, pp. 641-658, Jan. 2018.
- [8] Y.I. Alzoubi, A. Al-Ahmad and H. Kahtan, "Blockchain technology as a Fog computing security and privacy solution: An overview," *Computer Communications*, vol. 182, pp.129-152, Jan. 2022.
- [9] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure EHRs sharing of mobile cloud based e-health systems," *IEEE access*, vol. 7, pp. 66792-66806, Jan. 2019.
- [10] S. Shukla, S. Thakur, S. Hussain, J. G. Breslin, and S. M. Jameel, "Identification and Authentication in Healthcare Internet-of-Things Using Integrated Fog Computing Based Blockchain Model," *Internet of Things*, vol. 15, Sep. 2021.
- [11] P. K. Sharma, M. -Y. Chen and J. H. Park, "A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT," *IEEE Access*, vol. 6, pp. 115-124, 2018.
- [12] Y. Jiang, C. Wang, Y. Wang and Lang Gao. "A Cross-Chain Solution to Integrating Multiple Blockchains for IoT Data Management," *Sensors*, vol. 19, no. 9, 2019.
- [13] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi and K. Salah, "A User Authentication Scheme of IoT Devices using Blockchain-Enabled Fog Nodes," *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1-8, Aqaba, Jordan, 2018.
- [14] Androulaki, E., et al, "Hyperledger fabric: a distributed operating system for permissioned blockchains," *In Proceedings of the thirteenth EuroSys conference*, pp. 1-15, Apr. 2018.
- [15] Y. -T. Chen, C. -C. Chen, H. -Y. Chang, H. -S. Lin and H. -T. Chang, "Constructing ECA

Rule for IoT Application through a Novel S2RG Process: The Exemplary ECA Rules for Smarter Energy Applications," *2016 International Computer Symposium (ICS)*, pp. 549-554, Chiayi, Taiwan, Dec. 2016.

- [16] 이한솔, 최정우, 변기범, 홍지만, "포그 컴퓨팅을 위한 효율적인 IoT 플랫폼", *스마트미디어저널*, 제8권, 제1호, 35-42쪽, 2019년 03월
- [17] 민 연 아, "프라이빗 블록체인 기반의 사용자 환경을 고려한 수정된 PBFT 연구," *스마트미디어저널*, 제9권 제1호, 9-15쪽, 2020년 03월
- [18] 정지성, 이명훈, 박종권, "클라우드 컴퓨팅기반 가족 질병 예찰 및 스마트 측사 통합 관제 시스템," *스마트미디어저널*, 제8권, 제3호, 88-94쪽, 2019년 09월

저 자 소 개



김미선(정희원)

1996년 국립목포대학교 컴퓨터공학과 학사 졸업.

2000년 국립목포대학교 컴퓨터공학과 석사 졸업.

2007년 국립목포대학교 컴퓨터공학과 박사 졸업.

2012년 ~ 현재 국립목포대학교 정보보호학과 강사

<주관심분야 : 정보보호, 프로그래밍 언어, 모바일 시스템 보안, 클라우드 포그 컴퓨팅 보안, 사물인터넷 보안, 블록체인 기술>



박용석(정희원)

1998년 국립목포대학교 컴퓨터공학과 학사 졸업

2018년 국립목포대학교 정보보호기술학협동과정 석사 졸업

2020년 ~ 현재 국립목포대학교 정보보호기술학협동과정 박사과정

2000년 ~ 현재 현대삼호중공업 설계

ICT 근무 중

<주관심분야 : 정보보호, 클라우드 포그 컴퓨팅 보안, 사물인터넷 보안, 블록체인 기술>



서재현(정희원)

1985년 전남대학교 계산통계학과 학사 졸업.

1988년 중앙대학교 전자계산학과 석사 졸업.

1996년 전남대학교 전산통계학과 박사 졸업.

1996년 9월~현재 국립목포대학교 정

보보호학과 교수

2015년 3월~현재 한국정보처리학회 부회장

<주관심분야 : 정보보호, 시스템 및 네트워크 보안, 클라우드 포그 컴퓨팅 보안, 사물인터넷 보안, 블록체인 기술 >