

초연결 공통 네트워킹 서비스 플랫폼 개발

심재찬^{1*} · 윤호선¹ · 문성¹ · 박평구¹ · 광병욱¹ · 류호용¹¹한국전자통신연구원 지능네트워킹연구실 책임연구원

Development of Hyper-connected Common Networking Service Platform

Jaechan Shim^{1*} · Hosun Yoon¹ · Seong Moon¹ · Pyungkoo Park¹ · Byeongok Kwak¹ · Hoyong Ryu¹¹Principal Researcher, Intelligent Network Research Section, Electronics and Telecommunications Research Institute, Daejeon 34129, Korea

[요약]

기술혁신을 통한 산업성장 동력으로 연구, 개발 및 시험의 중요성이 증가하고 있으며, ICT융합이 전체 산업으로 확산되는 디지털 트랜스포메이션 시대를 맞아 연구개발 인프라와 서비스 인프라가 결합된 새로운 융합 인프라에 대한 요구와 투자가 경쟁적으로 진행되고 있다. 기술혁신을 실현하는 고도화된 연구개발 및 서비스 통합 플랫폼의 개발 및 제공은 필연적이라 할 수 있으며, 이는 컴퓨팅 가상화 기술과 네트워크 가상화 기술을 토대로 가상화 기반 클라우드 인프라로 실현된다. 본 논문에서는 ICT 융합 제품 및 서비스 개발에 유용한 초연결 공통 네트워킹 서비스 플랫폼의 구조 및 주요 기술을 소개한다. 그리고 개발 플랫폼을 적용한 서비스 실증 인프라를 제안하며, 초연결 공통 네트워킹 서비스 플랫폼의 활용 사례를 소개한다.

[Abstract]

The significance of research, development and testing is expanding as a driving force for industrial growth through technological innovation. In the era of digital transformation, where ICT convergence permeates all industries, there is a competitive surge in demand and investment in new convergence infrastructure that combines research and development (R&D) infrastructure and service infrastructure is competitively advancing. It can be asserted that the development and provision of advanced R&D and service integration platforms that materialize technological innovation are imperative. This achievement is realized through a virtualization-based cloud infrastructure founded on computing and network virtualization technology. This paper presents the structure and key technologies of a hyper-connected common networking service platform, which is invaluable for the development of ICT convergence products and services. Additionally, we propose a service verification infrastructure using a development platform and offer illustrative examples of the use of a hyper-connected common networking service platform.

색인어 : 가상화, 쿠버네티스, 인에이블러, 서비스 오케스트레이션, 클라우드 인프라스트럭처**Keyword** : Virtualization, Kubernetes, Enabler, Service Orchestration, Cloud Infrastructure<http://dx.doi.org/10.9728/dcs.2023.24.11.2881>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 26 September 2023; Revised 11 October 2023

Accepted 16 October 2023

*Corresponding Author, Jaechan Shim

Tel: +82-42-860-6177

E-mail: jcshim@etri.re.kr

I. 서론

치열한 기술경쟁 속에서 시장선점의 핵심수단으로 기술혁신이 부각되는 가운데 기술혁신을 통한 산업성장의 견인차로 연구개발 및 시험검증 환경의 중요성이 증가하고 있다. 세계 주요국들은 기술혁신을 위한 핵심기술 축적과 자국의 산업발전 및 혁신성장을 지원하기 위하여 미래 지향적인 연구개발 인프라를 구축하고 고도화하여 산학연 공동 활용으로 기술개발을 지원하고 있다. 또한 혁신적인 기술과 지식의 발전이 가속화됨에 따라 단순 연구 수단이 아닌 기술 개발 및 상용화를 위한 서비스 인프라의 중요성 역시 더욱 증가할 전망이며, ICT 융합이 전체 산업으로 확산되는 디지털 트랜스포메이션 시대에는 연구개발 인프라와 서비스 인프라가 결합된 새로운 융합 인프라의 출현 또한 가속화될 전망이다.

기술혁신 및 산업 발전의 토대인 연구개발 및 서비스 융합 인프라는 개방형 오픈소스 플랫폼의 특성을 띠 것으로 전망된다. 또한 신규 비즈니스 모델 및 ICT 서비스를 효과적으로 개발·운영·관리하며 특정 인프라나 벤더에 종속되지 않는 개방형 오픈소스 생태계는 지속적으로 확장하고 있다. 연구개발 및 서비스 융합 인프라의 핵심 요소인 연구개발 및 서비스 통합 플랫폼은 ICT를 중심으로 산업간 융합, 기술간 융합의 보편화에 따라 단지 연구개발 플랫폼의 역할뿐 아니라 개인별, 상황별, 의미별 서비스 활용에 초점을 둔 융합서비스 플랫폼으로 통합되어 발전할 것으로 예상된다.

연구개발 및 서비스 통합 플랫폼은 가상화 기반 클라우드 인프라로 실현된다. 가상화 기반 클라우드 인프라는 컴퓨팅 가상화 기술에 네트워크 가상화 기술을 토대로 구성되며 컴퓨팅, 네트워크, 스토리지 등 하드웨어 리소스와 소프트웨어 모듈, 애플리케이션 등 소프트웨어 리소스를 사용자에게 온디맨드로 제공한다[1],[2]. 주요 선진국 및 선도 기업들에 비해 출발이 다소 늦은 국내 수요자들에게 기술혁신을 실현하는 고도화된 연구개발 및 서비스 통합 플랫폼의 활용은 필수적이며, 이러한 융합 플랫폼의 제공은 필연적이라 할 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 주요 선진국에서 활용하고 있는 연구개발 또는 서비스 인프라에 대해 살펴보고, 3장에서는 초연결 공동 네트워크 서비스 플랫폼의 구조 및 기술을 소개한다. 이어서 4장에서는 개발 플랫폼을 적용한 서비스 실증 인프라를 설명하고, 5장에서는 초연결 공동 네트워크 서비스 플랫폼을 활용한 사례들을 소개한 후, 향후 전망과 시사점으로 마무리한다.

II. 관련 연구개발 동향

2-1 OPEN BATON

OPEN BATON은 ETSI (European Telecommunications

Standards Institute) NFV (Network Function Virtualisation) MANO (Management And Orchestration) 규격[3]의 완전한 구현을 목표로 독일 프라운호퍼에서 개발한 오픈소스 플랫폼으로 도커(Docker)의 컨테이너 서비스를 이용하여 템플릿 기반으로 복잡한 서비스를 쉽게 제공하며, 특정 클라우드 환경에서 가상 네트워크 인프라의 효율적인 운영을 지원한다. 그리고 IMS (IP Multimedia Subsystem), EPC (Evolve Packet Core), 4G/5G 무선 네트워크와 같은 주요 핵심 네트워크 기능의 클라우드화와 함께 on-demand, IaaS (Infrastructure as a Service), PaaS (Platform as a Service) 서비스 제공을 위해 좀 더 자동화된 가상화 인프라 제어 기능을 제공한다[4],[5].

OPEN BATON은 기본 인프라, 소프트웨어 아키텍처, 네트워킹, 관리 및 오케스트레이션을 통합하여 전체 인프라의 성능 향상과 보안 관리에 중점을 두고 있다. OPEN BATON 프레임워크에서는 VNF (Virtual Network Function)의 생명주기 관리를 위해 VNFM (Virtual Network Function Manager)을 이용한다. 또한, 분산 이벤트 관리와 스케일링 관리를 위한 자동 스케일링 (Auto Scaling)을 통합 운영한다. 그리고 Zabbix를 활용하여 모니터링 정보를 수집하고, 자동 런타임 관리를 수행하는 결합 관리 기능을 제공하며, 오케스트레이션 로직에 추가 및 삭제가 가능한 플러그인도 제공한다[6],[7].

2-2 FIWARE

FIWARE는 미래 인터넷 애플리케이션 및 서비스를 비용 효율적으로 생성하고 제공하기 위한 혁신적인 개방형 클라우드 기반 인프라이자 솔루션이며, 특히 다양성을 수용하기 위해 GE (Generic Enabler)와 SE (Specific Enabler) 개념을 도입하고, 상황 정보를 중심으로 수집-처리-적용 사이클 구성을 쉽게 할 수 있는 프레임워크를 제공하여 스마트 서비스 솔루션 개발을 용이하게 한다[8].

FIWARE는 다수의 내부 프로젝트로 구성되어 있는데, 핵심 프로젝트는 상황관리 프로젝트인 Orion으로 GE에 해당하며, FIWARE NGSIv2 (Next Generation Service Interface version 2) 표준을 기반으로 RESTful API를 제공하여 컨텍스트 (Context) 업데이트, 요청, 구독, 배포 등의 기능을 제공한다. 또한 IoT (Internet of Thing), 로봇 등의 서비스를 위한 GE를 제공하는데, IDAS (FIWARE IoT Agent suite)는 IoT Gateway로써 다양한 프로토콜을 사용하는 IoT 장치들로부터 정보를 모은다. 이들 외에도 데이터 처리를 위한 컨텍스트 처리 시각화, 미디어 스트림과 데이터 및 API 관리를 위한 API 매니지먼트, 데이터 게시 (Data Publication), 데이터 수익화 (Data Monetization), 보안 관련 GE들도 제공한다[9].

FIWARE는 지금까지 12개 응용 도메인의 레퍼런스 아키텍처와 호환성을 가진 공통 데이터 모델을 축적하여 다양한

Smart-X 솔루션 개발에 활용할 수 있도록 제공하고 있으며, 대표적인 Smart-X 도메인에는 스마트 시티, 스마트 환경, 스마트 농식품, 스마트 센싱, 스마트 로봇, 스마트 워크, 스마트 헬스, 스마트 항공, 스마트 에너지, 스마트 관광, 스마트 제조, 스마트 재난 등이 있다[10].

2-3 5GinFIRE

2003년 미국 NSF (National Science Foundation)의 지원 하에 인터넷을 재설계하려는 FI (Future Internet) 움직임이 세계적인 핫이슈로 부상하였으며, 미국 중심의 FI가 Internet의 기능적 혁신에 집중한 반면에, 유럽은 통신을 위한 네트워크 자체를 비즈니스 관점에서 유연하게 구축할 수 있도록 하는 방향으로 전략을 추진하였는데, 그 모태가 FIRE (Future Internet Research & Experimentation)이다[11]. FIRE의 이런 움직임은 ETSI ISG (Industry Specification Group)를 중심으로 소프트웨어 기반 네트워크 (Software based Network), 즉 네트워크 구성요소를 소프트웨어화하여 주문형으로 네트워크를 재구성 할 수 있는 NFV 개념을 태동시켰다[12]. 이후 FIRE는 NFV와 SDN (Software Defined Network) 개념을 통합한 단일 아키텍처를 구상하였고, 이런 SDN/NFV 개념이 FIRE를 구성하는 기본개념이 되었으며, FIRE의 프레임워크 속에서 5G를 지향하는 다양한 기능들을 서비스 관점에서 실험하는 장으로 발전하게 되었다.

5GinFIRE는 네트워크, 스토리지, 컴퓨팅 파워 등의 하드웨어들을 가상화 기반으로 재구성하여 다양한 5G 서비스 운용실험을 제공하는 플랫폼이다[13],[14]. FIWARE와 비교해보면, FIWARE의 GE와 SE를 VxF (Virtual Anything Function)로 통합하여 VxF를 물리 기능과 가상 기능을 통칭하는 통합적 개념으로 사용한다. VxF는 클라우드에서 NFV 기능으로 실행이 될 수도 있고, 물리적인 특성을 살리기 위하여 하드웨어의 논리적인 분할을 통한 물리적 기능으로 실행될 수 있다. 5GinFIRE는 이러한 VxF를 사용하여 다양한 실험을 할 수 있도록 테스트베드를 구축할 수 있게 해준다.

III. 초연결 공통 네트워킹 서비스 플랫폼 (HNSP; Hyper-connected common Networking Service Platform)

연구개발 동향에서 살펴본 바와 같이 이들이 제공하는 솔루션들은 공통적으로 네트워킹 응용서비스 개발의 효율성을 높이기 위해 공용 소프트웨어 라이브러리 (GE, VxF 등)를 제공하고, 자동화된 가상화 기반 운용 환경을 통해 서비스를 제공하는데, HNSP 역시 컨테이너 기반 인에이블러 (Enabler) 들을 이용하여 가상 환경에서 다양한 융복합 서비스의 개발, 시험 및 물리 인프라 환경에 존재하는 PNF (Physical

Network Function)들과 연동하여 융복합 네트워킹 서비스를 제공하는 플랫폼이자 통합 인프라이다.

3-1 HNSP 구조

HNSP 시스템은 그림 1과 같이 인에이블러와 서비스 개발을 위한 ESDP (Enabler and Service Development Platform), 인에이블러 및 서비스를 운용하는 SOP (Service Orchestration Platform), 물리 인프라 (Physical Infrastructure) 그리고 이들을 관리하기 위한 관리자 (Administrator) UI/UX로 구성된다[15].

ESDP는 다시 인에이블러 제작 및 관리를 위한 EDP (Enabler Development Platform), 서비스 제작 및 관리를 위한 SCP (Service Composer Platform) 그리고 EDP와 SCP에서 개발된 인에이블러와 서비스를 관리하는 스토어 (Store)로 구성된다.

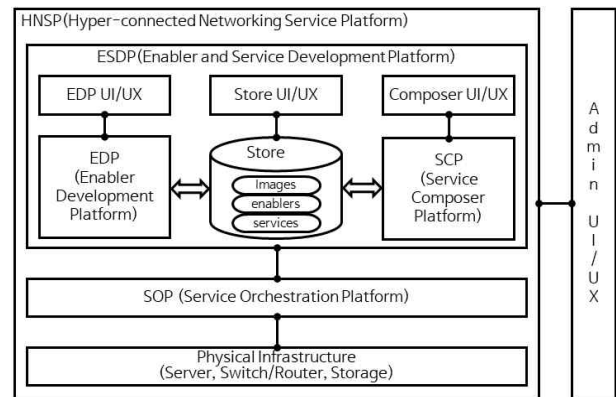


그림 1. HNSP 구조도
Fig. 1. HNSP Architecture Diagram

사용자 관점에서 HNSP를 활용하는 절차는 그림 2와 같다. 먼저 HNSP 홈페이지를 이용하여 회원가입 및 등록절차를 진행하고, 회원등록이 완료되면 EDP를 이용하여 인에이블러를 개발하고, 개발한 인에이블러를 검증하는 절차를 진행한다. 검증된 인에이블러는 서비스 제작에 사용하거나 마켓플레이스인 스토어에 등록하여 다른 사용자에게 제공된다. 사용자가 직접 개발한 인에이블러와 스토어에서 다운로드한 인에이블러를 이용하여 서비스를 제작하고, SOP를 활용하여 서비스 운영 테스트를 진행한다. 그리고 서비스 실행하여 사용자별 리소스, 장애, 로그 정보 등을 수집할 수 있다. 이때 관리자 대시보드를 이용하면 사용자별 리소스, 장애, 로그를 관리할 수 있다. 검증된 서비스 역시 마켓플레이스인 스토어에 등록하고 다른 사용자와 공유할 수 있도록 허용한다[16].

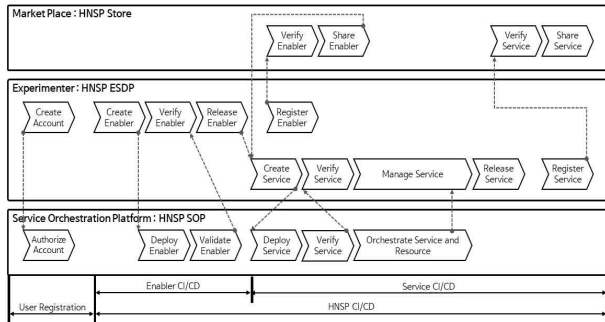


그림 2. HNSP 작업흐름
Fig. 2. HNSP workflow

HNSP의 특징은 하나의 플랫폼을 이용하여 인에이블러 및 서비스의 개발과 개발된 서비스를 사용자에게 제공할 수 있다는 점이다. 그리고 스토어를 통해 개발자들 사이에 인에이블러와 응용 서비스를 공유할 수 있다는 장점이 있다. 다음 절에서는 이러한 특징점을 가진 HNSP의 구성 요소들을 좀 더 구체적으로 소개한다.

3-2 인에이블러 및 서비스 개발 플랫폼 (ESDP)

ESDP를 구성하는 EDP, SCP, 스토어 그리고 관리자 UI/UX의 역할 및 제공 기능은 다음과 같다.

1) 인에이블러 개발 플랫폼 (EDP)

네트워크 기반 응용서비스 개발에 있어 가장 중요한 요소는 소프트웨어 모듈인 인에이블러이며, 인에이블러는 일반 인에이블러 (GE)와 특별 인에이블러 (SE)로 구분한다. GE는 HNSP 스토어를 통해 공유하는 인에이블러로 서비스 개발에 공통으로 활용되는 인에이블러이며, SE는 특정 서비스에 의존성이 강한 인에이블러이다. 물론 SE를 공동 활용을 위해 스토어에 등록하여 공개하면 GE가 되며, 이런 인에이블러를 개발하고 검증하는 플랫폼이 EDP이다.

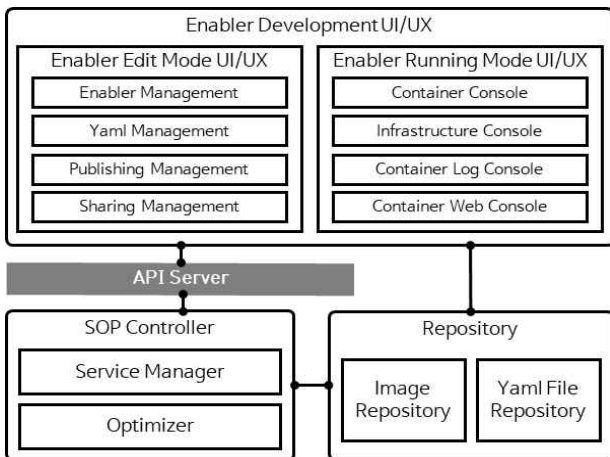


그림 3. EDP 구조도
Fig. 3. EDP architecture diagram

인에이블러 개발 및 관리를 위한 EDP는 그림 3과 같이 Enabler Development UI/UX, API 서버, SOP 컨트롤러 그리고 리포지토리로 구성된다.

인에이블러의 제작, 시험 및 관리 기능을 제공하는 EDP를 효율적으로 활용하기 위한 사용자 인터페이스인 EDP UI/UX는 편집 모드 (Enabler Edit Mode)와 실행 모드 (Enabler Running Mode)로 구분된다.

편집 모드는 인에이블러의 생성/저장/불러오기 등과 같은 인에이블러 관리 기능, 야플 파일 자동생성/수정/삭제/저장/불러오기 등과 같은 야플 관리 기능, 제작된 인에이블러를 저장소에 등록하는 퍼블리싱 관리 기능, 특정 사용자 그룹과 인에이블러를 공유하는 공유 관리 기능 등을 제공한다.

그림 4는 인에이블러 개발 UI/UX-편집 모드 화면으로 ① 인에이블러 구성 요소 편집기, ② 야플 그래픽 편집기, 그리고 ③ 야플 텍스트 편집기로 구성된다.

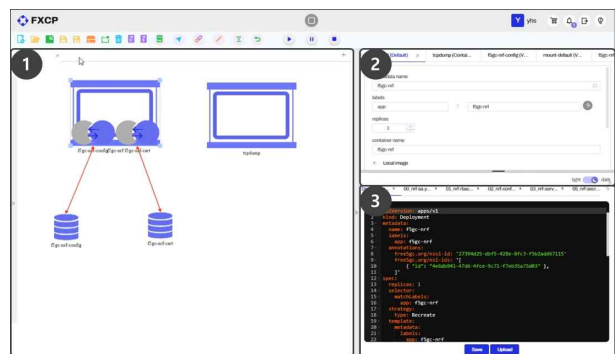


그림 4. 인에이블러 개발 UI/UX - 편집 모드
Fig. 4. Enabler development UI/UX - Edit mode

실행 모드는 컨테이너 제어를 위한 컨테이너 콘솔 기능, 인 프라스트럭처 상태 확인을 위한 인프라 콘솔 기능, 각 컨테이너 로그 정보 확인을 위한 컨테이너 로그 콘솔 기능, 웹 지원 컨테이너를 위한 웹 콘솔 기능 등을 제공한다. 그리고 SOP 컨트롤러는 서비스별 권한/상태/버전 관리를 위한 서비스 매니저 기능과 서비스 운영 최적화를 위한 옵티마이저 기능을 제공한다.

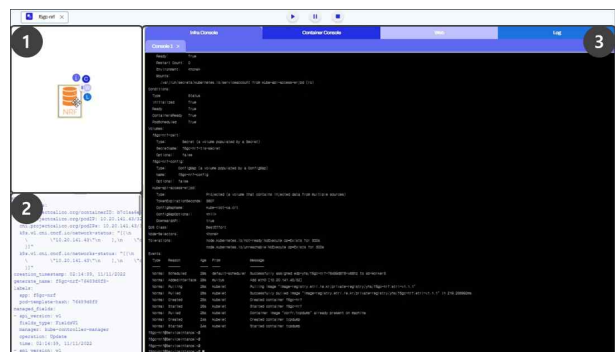


그림 5. 인에이블러 개발 UI/UX - 실행 모드
Fig. 5. Enabler development UI/UX - Running mode

그림 5는 인에이블러 개발 UI/UX-실행 모드 화면으로 ① 인에이블러 동작 상태 아이콘, ② 인에이블러 구동 절차 표시 터미널, 그리고 ③ SOP 운용 콘솔/컨테이너 콘솔/컨테이너 웹 콘솔/컨테이너 로그 출력 창으로 구성된다.

2) 서비스 개발 플랫폼 (SCP)

서비스 제작 및 관리를 위한 SCP는 그림 6과 같이 Service Composer UI/UX, API 서버, SOP 컨트롤러 그리고 리포지토리로 구성되며, 이는 EDP 구조와 매우 유사한데, 인에이블러는 원칙적으로 하나의 컨테이너로 구성되고, 서비스는 다수의 컨테이너로 구성되는 차이만 있을 뿐 이들에 대한 제어, 관리 및 운용은 동일하기 때문이다.

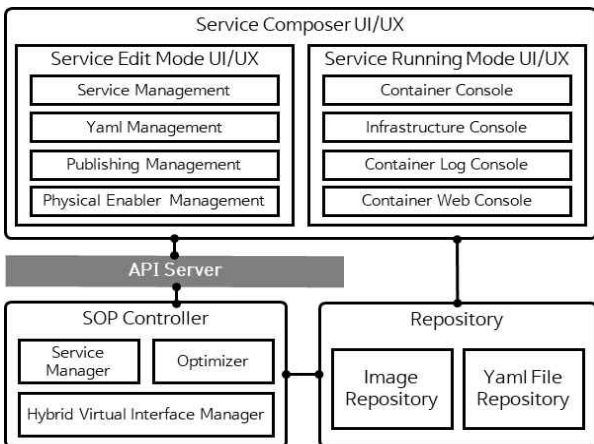


그림 6. SCP 구조도
Fig. 6. SCP architecture diagram

응용 서비스의 제작, 시험 및 관리 기능을 제공하는 SCP를 효율적으로 활용하기 위한 사용자 인터페이스인 SCP UI/UX는 편집 모드 (Service Edit Mode)와 실행 모드 (Service Running Mode)로 구성된다.

편집 모드는 서비스 생성/저장/불러오기 등과 같은 서비스 관리 기능, 야블 파일 자동생성/수정/삭제/저장/불러오기 등과 같은 야블 관리 기능, 제작된 서비스를 저장소에 등록하는 퍼블리싱 관리 기능 그리고 물리 인에이블러 생성/삭제/수정을 위한 물리 인에이블러 관리 기능 등을 제공한다.

그림 7은 서비스 제작 UI/UX-편집 모드 화면으로 ① 사용자 별 인에이블러 목록, ② 서비스 편집 창, ③ 서비스 구성 인에이블러 야블 목록 창, 그리고 ④ 야블 텍스트 편집기로 구성된다.

실행 모드는 컨테이너 제어를 위한 컨테이너 콘솔 기능, 인프라스트럭처 상태 확인을 위한 인프라 콘솔 기능, 각 컨테이너 로그 정보 확인을 위한 컨테이너 로그 콘솔 기능, 웹 지원 컨테이너를 위한 웹 콘솔 기능 등을 제공한다. 그리고 SOP 컨트롤러는 서비스별 권한/상태/버전 관리를 위한 서비스 매니저 기능, 서비스 최적화를 위한 옵티마이저 기능 그리고 P2P 가상 네트워크 관리를 위한 Hybrid VIM 기능을 제공한다.

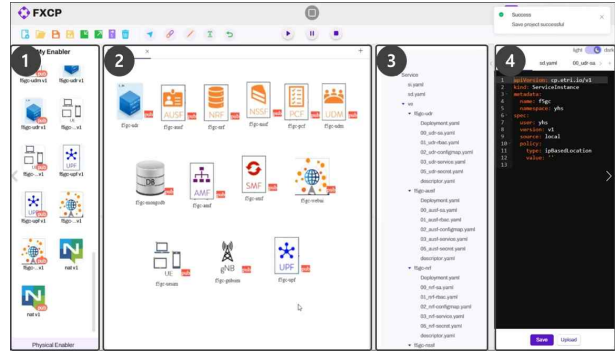


그림 7. 서비스 제작 UI/UX - 편집 모드
Fig. 7. Service Composer UI/UX - Edit Mode

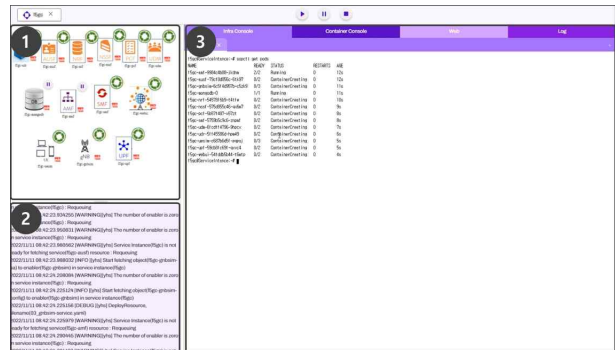


그림 8. 서비스 제작 UI/UX - 실행 모드
Fig. 8. Service composer UI/UX - Running mode

그림 8은 서비스 제작 UI/UX-실행 화면으로 ① 서비스 동작 상태 아이콘, ② 서비스 구동 절차 표시 터미널, 그리고 ③ SOP 운용 콘솔/컨테이너 콘솔/컨테이너 웹 콘솔/컨테이너 로그 출력 창으로 구성된다.

3) 스토어 (Store)

만들어진 인에이블러와 서비스는 기능 검증 후에 각 개발자들의 개인 저장소에 보관한다. 또한, 이들 인에이블러와 서비스는 다른 사용자들과 공유할 수 있는데, 이때 활용하는 저장소가 바로 스토어이다.

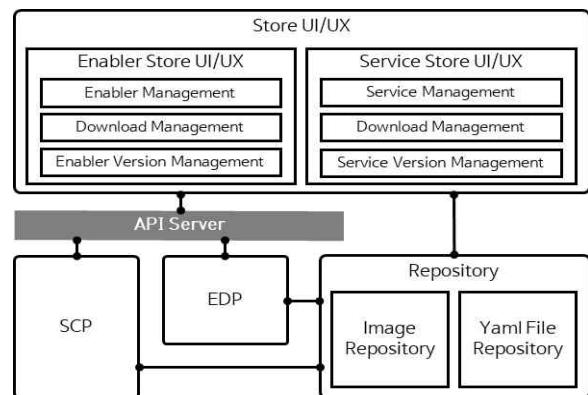


그림 9. Store 구조도
Fig. 9. Store architecture diagram

스토어 UI/UX는 인에이블러/서비스 등록/삭제 관리 기능, 사용자가 필요한 인에이블러/서비스 다운로드 관리 기능, 인에이블러/서비스 버전 관리 기능을 제공하며, 저장소 (Repository)는 HNSP 플랫폼에서 사용하는 이미지를 관리하는 이미지 저장소 (Image Repository)와 HNSP 플랫폼에서 사용하는 야플 파일을 관리하는 야플 파일 저장소 (Yaml File Repository)로 구성된다.

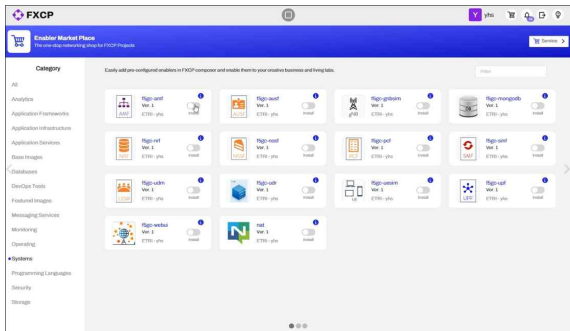


그림 10. 스토어 UI/UX
Fig. 10. Store UI/UX

인에이블러와 서비스의 공유 및 관리를 위한 스토어UI/UX는 일반 사용자용 UI/UX와 관리자용 UI/UX를 별도로 지원하며, 버전별/종류별로 인에이블러/서비스 공유 및 관리 기능, 사용자별 인에이블러/서비스 다운로드 관리 기능, 인에이블러/서비스 정보를 저장하기 위한 리퍼지토리 연동 기능 그리고 인에이블러 제작 및 관리를 수행하는 EDP 연동 기능에 대한 인터페이스를 제공한다.

4) 관리자 UI/UX

인에이블러/서비스의 제작 및 검증 플랫폼, 검증된 인에이블러/서비스의 공유 저장소, 실제 운용 환경에서 서비스 동작을 실증하기 위한 가상 물리 통합 인프라 등 HNSP에 대한 운용, 관리 및 유지보수를 위한 관리자용 UI/UX이다.

HNSP의 일반 사용자 등록 및 관리를 위한 사용자 관리 기능, 전반적인 인프라의 상태 확인을 위한 대시보드 기능, 구체적인 자원 상황 파악을 위한 자원 관리 기능, 각 인에이블러에서 출력하는 로그를 관리하고 출력하기 위한 로그 관리 기능을 제공한다.

HNSP의 다른 구성요소들과 달리 다양한 오픈 소스를 이용하여 개발하였으며, 노드, 스토리지, 네임스페이스 등과 같은 클러스터 정보, 파드, 디플로이먼트, 데몬셋, 스테이트풀셋 등과 같은 워크로드 정보, 워크로드에 접근하기 위한 서비스 정보, 설정을 위한 컨피그맵 정보, 스토리지 상세 정보, 클러스터 정보를 제외한 사용자별 정보 등을 다루는 대시보드 기능 개발에는 쿠버네티스의 대시보드를 활용하였다. 그리고 노드 자원정보 및 컨테이너 메트릭 정보의 관리를 위한 자원 정보 관리 기능 개발은 Prometheus를 이용하였으며, 자원 정보 출력 기능 개발에는 Grafana를 활용하였고, 로그 수집 및 출력 기능은 Elasticsearch, Fluentd, Kibana를 이용하였다.

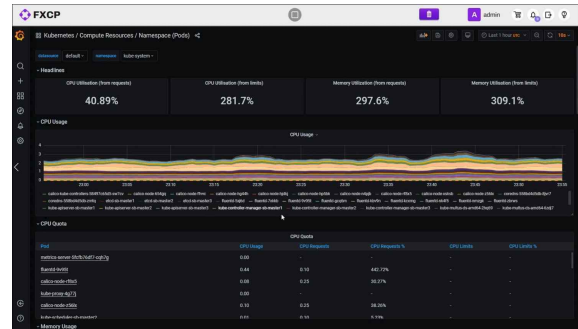


그림 11. 관리자 UI/UX - 자원관리
Fig. 11. Administrator UI/UX - Resource management

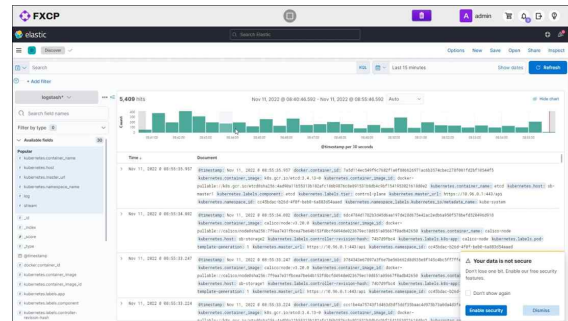


그림 12. 관리자 UI/UX - 로그
Fig. 12. Administrator UI/UX - Log

3-3 서비스 오케스트레이션 플랫폼 (SOP)

인에이블러 기반 서비스들이 인프라에서 동작하기 위해서는 가상 환경에서 동작할 수 있는 가상 개체, 즉 컨테이너로 제작되어야 하며, 이들 컨테이너의 실행을 위해 CPU, 메모리 및 네트워크 자원의 할당, 요구사항 기반 스케줄링 등 생성부터 감지, 복구 및 삭제와 같은 전반적인 라이프사이클을 관리하는 기능을 수행하는 플랫폼이 SOP이고, 주요 요구사항은 다음과 같다.

- 서비스 라이프사이클 관리 : 서비스의 시작, 운영 및 종료를 관리한다.
- 서비스 보증 : 서비스의 장애를 모니터링하고, 장애 시 복구 절차를 수행한다.
- 자원 구성 : 서비스가 필요로 하는 자원을 할당하고 설정한다.
- 동적 자원 확장 : 동적으로 자원을 관리한다.
- 데이터 수집 : 실시간으로 다양한 채널을 통해서 메트릭 데이터를 수집한다.
- 가상 인프라 관리 : 인프라 자원에 대한 할당 및 설정을 제어한다.

SOP는 그림 13과 같이 다양한 구성 요소들의 상호 연동을 통해 하부 인프라를 제어하고, 서비스를 제공한다. 따라서 운

영자는 인프라 상에서 동작하는 서비스 관리를 위해 GUI (Graphic User Interface) 또는 CLI (Command Line Interface)로 SOP에 명령을 내리고, API 서버는 명령을 해석하여 필요한 기능을 제공하는 담당 컨트롤러에 전달하며, 각 컨트롤러는 API 서버를 통해 전달된 업무를 인지하고 처리한다. 이러한 일련의 동작 과정에서 발생하는 모든 형상 정보는 etcd에 저장된다.

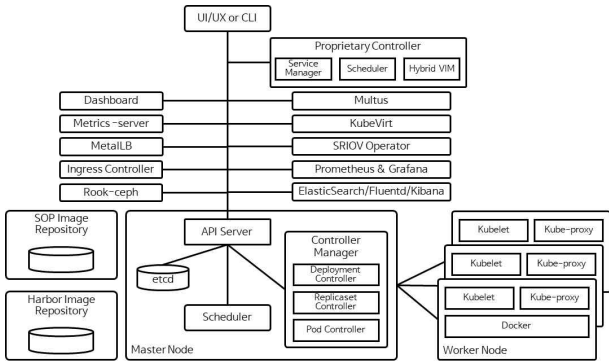


그림 13. SOP 구조도
Fig. 13. SOP Architecture Diagram

SOP를 구성하는 컨트롤러는 필요에 따라 추가 또는 대체될 수 있으며, 주요 구성요소의 역할은 다음과 같다.

- API 서버 : 클라이언트인 컨트롤러에 명령을 전달하는 중앙 서버
- 컨트롤러 매니저 : 디플로이먼트, 레플리카셋, 파드 자원 관리
- 메트릭 서버 : 시스템 자원 현황 수집
- 대시보드 : GUI 대시보드 제공
- MetalLB : 로드밸런서 유형의 서비스들에게 외부와의 네트워킹 제공
- 인그레스 컨트롤러 : HTTP 기반으로 외부와 네트워킹 제공
- Rook-ceph : 스토리지 클러스터 제공
- Multus : 가상개체들에게 다중 네트워킹 제공
- KubeVirt : 가상머신 기반 가상개체 오케스트레이션
- SRIOV operator : SRIOV 네트워킹 오케스트레이션
- Prometheus/Grafana : 자원 사용 현황 모니터링
- EFK (Elasticsearch, Fluentd, Kibana) : 데이터 저장, 로깅, 시각화
- 서비스 매니저 : 서비스의 시작과 종료 제어
- 스케줄러: 가상개체가 동작할 노드 선택 제어
- Hybrid VIM (Virtual Infrastructure Manager) : 가상개체를 물리 장치에서 실행시키는 기능 수행

컨트롤러들 중에서 서비스 매니저, 스케줄러 그리고 Hybrid VIM은 HNISP를 위해 독자 개발하였으며, 이와 같이

필요한 컨트롤러를 개발하여 적용할 수 있는 프레임워크 또한 플랫폼이 가진 특징 중 하나이다.

SOP 환경 구축 방법에는 단일 클러스터 방식과 분산 클러스터 방식이 있다. 단일 클러스터 방식은 한 공간에 SOP 클러스터를 구성하는 방법으로, 응용 서비스 시작 및 종료 기능, 자원 기반 스케줄링 기능, 배치 기반 스케줄링 기능, 네트워크 대역폭 기반 스케줄링 기능, 인에이블러 스케일링 기능, 인에이블러 마이그레이션 기능, 서비스별 가상 네트워크 대역폭 제어 기능, 서비스 라이프 사이클 연동 가상 링크 관리 기능, 서비스 가상 링크 마이그레이션 기능, 가상 머신(VM) 기반 인에이블러 관리 기능, 인에이블러-다중 네트워크 연동 기능, 인에이블러-물리 장치 연동 기능 그리고 인에이블러 물리 네트워크 연동 기능을 제공한다.

분산 클러스터 방식은 지리적으로 떨어진 공간에 SOP 환경을 구성하는 것으로 분산 배치되어 있는 클러스터 상에서 서비스들을 배포하고 관리하는 기능을 지원한다.

지리적으로 분산되어 있는 클러스터들을 서로 연동하는 방법은 계층적 클러스터 연동 방법과 비계층적 클러스터 연동 방법이 있다. 계층적 연동 방법은 중앙 클러스터, 지역 클러스터 그리고 에지 클러스터 등으로 그 규모와 위치에 따라서 계층화해서 관리하는 방법이다. 비계층적 방식은 모든 클러스터들을 동등한 레벨로 바라보며 호스트 클러스터와 멤버 클러스터로 분리해서 관리한다. 비계층적 방식은 계층적인 방식에 비해서 관리 기능이 단순한 장점이 있는데 HNISP는 비계층적인 방식을 지원하며, 분산 클러스터에서 서비스 시작 및 종료 기능, 클러스터 별 명세서 Override 기능, 분산 클러스터에서 자원 기반 인에이블러 스케줄링 기능, 분산 클러스터에서 인에이블러 마이그레이션 기능, 그리고 DNS 연동 기능을 추가로 제공한다.

IV. 초연결 리빙랩을 위한 서비스 실증 인프라 구축

4-1 초연결 서비스 실증 인프라

서비스 실증 인프라는 인에이블러와 서비스의 시험을 통한 기능 확인 및 서비스의 성능 검증 목적으로 구축하였으며, 하드웨어 장비들로 구축되는 실증 인프라의 구성은 다음과 같다.

- SOP 운용을 위한 마스터 노드 3대, 인에이블러 및 서비스 구동을 위한 워커 노드 5대, 인에이블러 이미지를 위한 이미지 저장소 1대
- SOP 컨트롤러 이미지를 위한 저장 서버 1대
- 인프라 내부 제어 메시지를 통신을 위한 네트워크 구성 스위치 1대, 인프라 외부 사용자의 통신을 위한 로드 밸런서 스위치 1대

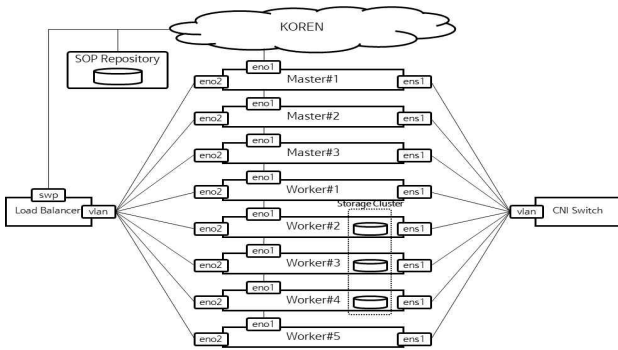


그림 14. HNSP 물리 장치 구성도
 Fig. 14. HNSP physical architecture diagram

사용자에게 제공하는 서비스 및 인프라 자체에 대한 운용 관리를 제공하는 소프트웨어 환경은 다양한 역할을 담당하는 컴포넌트들이 각자 맡은 일을 독립적으로 수행하는 SOP이다.

그림 13과 같이 컨테이너 관리 도구이자 컨테이너 오케스트레이션을 위한 대표 솔루션인 쿠버네티스를 SOP의 기본 환경으로 활용하여 부가 기능들을 추가 개발하여 적용하였으며, 다음과 같은 특징이 있다.

- 고 가용성 지원 SOP : 장애 발생 자체를 미연에 방지하고, 장애 발생에 신속히 대처할 수 있도록 고 가용성을 지원하며, 마스터 노드의 구성을 장애 발생시 절체가 가능한 Three 노드 고가용성 (HA; High Availability) 구조로 구축
- 공용 인터넷 연동 로드 밸런서 : 서비스를 이용하려면 인프라 외부에서 사용자들이 접근할 수 있어야 한다. 이를 위해 인프라에서 동작하는 서비스 인에이블러에게 공용 IP를 할당하는 기능을 제공하며, 사용자들의 요청이 증가할 때 부하분산을 위한 부하분산 기능 제공
- 스토리지 클러스터 : 동작하는 인에이블러들이 사용하는 저장 공간으로 블록 스토리지를 볼륨으로 연결해서 사용할 수 있도록 스토리지 클러스터 구축

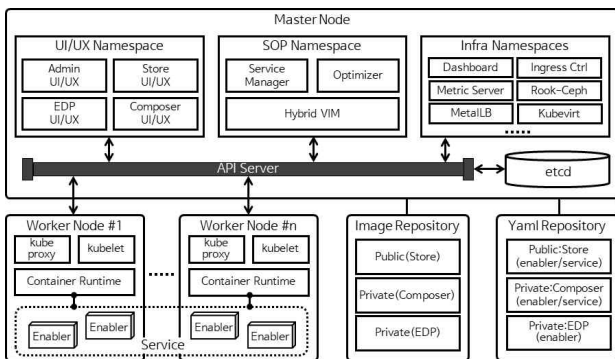


그림 15. HNSP 소프트웨어 구조도
 Fig. 15. HNSP software architecture diagram

4-2 HNSP 자동 설치 도구 (HNSP Auto-Installer)

HNSP는 내부적으로 수많은 소프트웨어 컴포넌트들이 유기적으로 결합하여 동작하도록 구성되며, 필요에 따라 다양한 컴포넌트들이 설치되어야 하고, 요구한 동작을 효과적으로 수행하기 위한 구성 및 설정이 필요하다. 하지만 이러한 작업을 운용자나 사용자가 하나씩 수작업을 통해 처리하는 것은 비효율적이다. 따라서 클라우드에 HNSP 환경을 자동으로 구축해 주는 자동 프로비저닝 도구를 개발하여 제공한다.

HNSP 자동 설치 도구는 앤서블 (Ansible) 스크립트 기반으로 컴퓨팅 장비에 원격 접속하여 미리 정해진 플레이북 (Playbook)들을 실행하는 방식을 취하며, 각 플레이북은 다양한 플레이 (Play)들을 포함하고 있고, 이 플레이들은 그 내부에 정의되어 있는 역할 (Role)과 임무 (Task)들을 수행한다. 그리고 웹 GUI를 통해 프로젝트 단위로 필요한 플레이북들을 지정한 호스트들에 직관적으로 설치하며, 설치 소프트웨어 컴포넌트들의 자동 프로비저닝 기능을 제공한다.

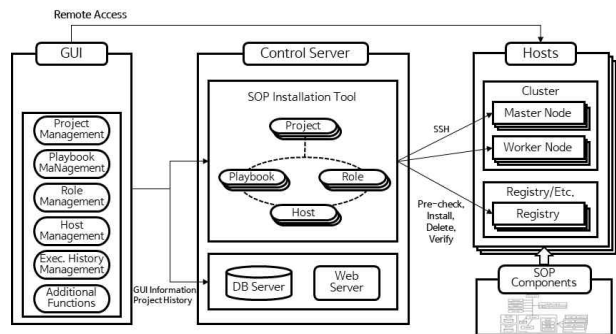


그림 16. HNSP 자동 설치 도구 개념도
 Fig. 16. HNSP auto-installer concept diagram

V. 활용 사례 (Use Case)

HNSP에서 제공하는 개발 플랫폼을 활용하여 인에이블러와 이를 이용한 서비스 개발 및 서비스 시험 환경 구축 사례를 소개한다.

5-1 5G 코어 인에이블러

개발자는 EDP를 이용하여 5G 코어 함수(5G Core Function)들을 인에이블러로 개발하게 된다. 즉, 제작 절차에 따라 각 인에이블러에 대한 컨테이너, 서비스, 볼륨, 볼륨 마운트 및 오토 스케일링 관련하여 구성 및 설정 정보들을 야믈 파일로 작성하게 되는데, 5G 코어 컴포넌트들 각각에 대해 Deployment.yaml, Service.yaml, ConfigMap.yaml, PersistentVolumeClaim.yaml 그리고 HorizontalPodAutoscaler.yaml 파일을 생성한다. 다음으로 SOP와 연동하여 동작을 검증한 후 개발자별 저장소에 저장하

고, 필요한 경우 공용 스토어에 등록하게 된다. 그림 17은 공개 소프트웨어인 Free5GC의 구성 요소들을 각각 인에이블러로 만들어 스토어에 등록한 모습이다.

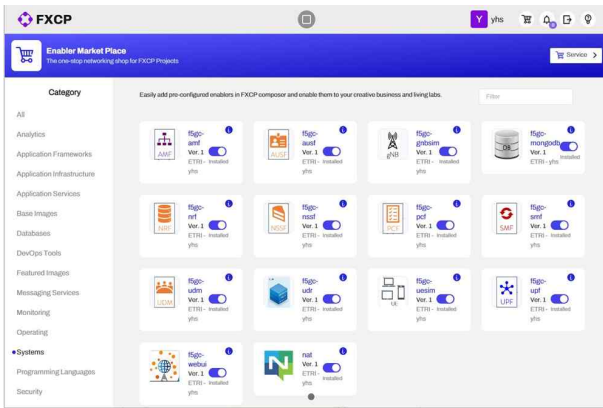


그림 17. 스토어 등록 5G 코어 인에이블러
Fig. 17. 5G Core enablers in store

5-2 5G 코어 서비스

5G 코어 함수 또는 5G 응용 서비스 개발자는 개발한 코어 함수 또는 응용 서비스 시험을 위해 5G 시험망이 필요하다. 그러나 시험망 구축에 필요한 장비 및 소프트웨어 도입 등에 많은 비용이 발생한다. 따라서 이러한 비용과 시간을 단축하고, 생산성을 향상하는 방법으로 GE로 제공하는 모바일 서비스용 인에이블러들을 이용하여 SCP에서 5G 모바일 네트워크 서비스 환경을 직접 만드는 것이다. HNSP에서는 모바일 서비스 제작용으로 오픈 소스인 Free5GC와 Open5GS 기반으로 만든 모바일 인에이블러들을 기본으로 제공하며, 특별히 Untrusted Non-3GPP Access Network용 N3IWF 인에이블러와 P4 (Programming Protocol-independent Packet Processor) 기술을 활용하여 자체 제작한 UPF (User Plane Function) 인에이블러를 제공한다.

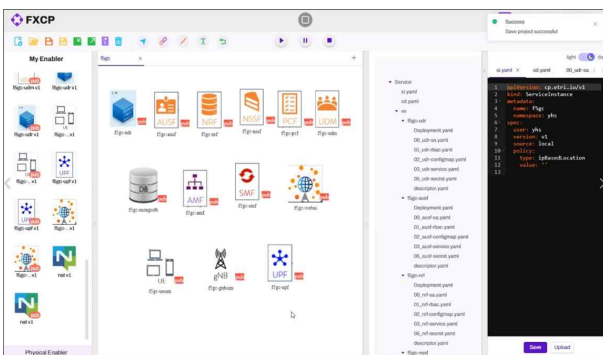


그림 18. 5G 코어 서비스 제작
Fig. 18. 5G core service authoring

5-3 Private 5G 서비스 시험 환경

초고속, 초저지연, 초연결의 특징이 있는 5G 기술을 기반으로 산업 전반에 디지털 혁신을 가져오기 위해 특정 기업이나 사용자들에게 필요한 특별한 서비스를 제공하는 사설망이 5G 특화망이다. 5G 특화망에서 활용하는 5G는 데이터 속도, 용량, 지연시간 등에서 Wi-Fi 대비 우수하며, 보안 유지 및 이동성이 보장되기에 Wi-Fi로 구현할 수 없던 다양한 특화망 서비스를 제공할 수 있으며, 대표적인 활용분야로는 스마트시티, 스마트팩토리, 자율주행로봇, 스마트농업 등이 있고, 적용 분야 및 활용에 대한 관심이 높아지고 있다.

하지만 5G 특화망에 필요한 장치나 서비스를 개발하는 기업 입장에서는 제품의 기능 및 성능 검증을 통한 상용화에 있어 중요한 시험 환경 확보가 어려운 현실이다. 그래서 합리적인 비용으로 5G 특화망 시험환경 구축이 가능해야 한다. 그림 19는 5-2에서 소개한 5G 코어 서비스를 활용하여 COTS (Commercial Off The Shelf) 물리 장치와 가상화 기반 소프트웨어를 결합한 5G 특화망 시험 환경이다.

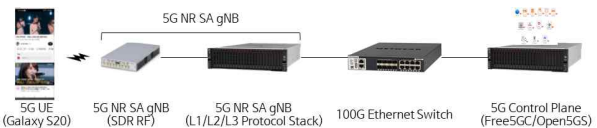


그림 19. 5G 특화망 서비스 테스트베드
Fig. 19. Private 5G service testbed

VI. 결 론

ICT 융합이 전 산업으로 확산되는 디지털 트랜스포메이션 시대에 가상화, SDN, NFV, 오픈 소스 기술과 융합한 클라우드 기반 산업 발전과 혁신 성장의 토대가 되고 있으며, 그 중심에는 연구개발과 서비스 제공을 위한 ICT 융합 인프라가 있다.

본 논문에서는 국가의 미래 성장 동력 확보를 위해 개발한 ICT 융합 인프라로서 연구 개발 및 서비스 통합 플랫폼인 초연결 공통 네트워킹 서비스 플랫폼 (HNSP)의 구조, 구성 요소 및 제공 기능에 대해 소개하였다. HNSP는 인에이블러 및 서비스의 개발과 개발된 서비스를 제공하는 일체화된 단일 플랫폼이며, 인에이블러와 응용 서비스 개발을 위해 제공하는 UI/UX 기능과 스토어를 통해 개발자들 사이에 인에이블러와 응용 서비스를 공유할 수 있는 부분은 제품과 서비스 상용화 측면에서 아주 큰 장점이다. 또한 이러한 연구개발 및 응용 서비스 플랫폼을 자동으로 구성해 주는 자동 설치 도구 역시 HNSP가 가진 장점 중 하나이다.

향후에는 HNSP를 좀 더 많이 활용할 수 있도록 활용 분야를 더욱 다양화하고 각 분야별 공통 인에이블러를 적극 개발하여 제공함으로써 이를 활용해 개발된 응용 서비스들의 상

업화가 더욱 활성화되어 기존 산업이 한층 더 고도화되고 융합 신산업이 창출되는 등 국내 ICT 융합 생태계에 활력을 불어넣는 계기가 되길 기대한다.

감사의 글

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구(No. 2019-0-00260, 초연결 공통 네트워크 서비스 연구인프라 구축)로서, 관계부처에 감사드립니다.

참고문헌

- [1] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile Cloud Computing: A Survey," *Future Generation Computer System*, Vol. 29, No. 1, pp. 84-106, January 2013. <https://doi.org/10.1016/j.future.2012.05.023>
- [2] E. D. Kim, Y. W. Woo, K. S. Ok, and E. K. Baek, "Outlook for 5G Virtualization and Integrated Automatic Control System," *OSIA S&TR Journal*, Vol. 31, No. 4, pp. 26-30, December 2018.
- [3] ETSI (European Telecommunications Standards Institute). Network Functions Virtualisation (NFV); Management and Orchestration [Internet]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf.
- [4] P. Bellavista, L. Foschini, R. Venanzi, and G. Carella, "Extensible Orchestration of Elastic IP Multimedia Subsystem as a Service Using Open Baton," in *Proceedings of the 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, San Francisco: CA, pp. 88-95, April 2017. <https://doi.org/10.1109/MobileCloud.2017.31>
- [5] G. A. Carella, M. Pauls, T. Magedanz, M. Cilloni, P. Bellavista, and L. Foschini, "Prototyping NFV-Based Multi-Access Edge Computing in 5G Ready Networks with Open Baton," in *Proceedings of IEEE Conference on Network Softwarization (NetSoft)*, Bologna, Italy, pp. 1-4, July 2017. <https://doi.org/10.1109/NETSOFT.2017.8004237>
- [6] Open Baton. An Open Source Network Function Virtualisation Orchestrator (NFVO) Fully Compliant with the ETSI NFV MANO Specification [Internet]. Available: <http://openbaton.org>.
- [7] J. C. Shim, P. K. Park, H. Y. Ryu, and T. Y. Kim, "Analysis of Trends in Hyper-connected Virtual Infrastructure Management Technology," *Electronics and Telecommunications Trends*, Vol. 35, No. 4, pp. 135-148, August 2020. <https://doi.org/10.22648/ETRI.2020.J.350412>
- [8] FIWARE. NGSI-V2: Step-by-Step [Internet]. Available: <https://fiware-tutorials.readthedocs.io/en/latest/>.
- [9] V. Heikkilä, O. Hylli, M. Nurminen, and K. Systä, Report on FIWARE Platform, CityIoT Technical Report, June 2020.
- [10] U. Ahle and J. J. Hierro, FIWARE for Data Spaces, in *Designing Data Spaces*, Cham, Switzerland: Springer, ch. 24, pp. 395-417, 2020.
- [11] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future Internet Research and Experimentation: The FIRE Initiative," *ACM SIGCOMM Computer Communication Review*, Vol. 37, No. 3, pp. 89-92, July 2007. <https://doi.org/10.1145/1273445.1273460>
- [12] K. Kaur, V. Mangat, and K. Kumar, "A Comprehensive Survey of Service Function Chain Provisioning Approaches in SDN and NFV Architecture," *Computer Communication Review*, Vol. 38, 100298, November 2020. <https://doi.org/10.1016/j.cosrev.2020.100298>
- [13] J. R. Martinez, D. R. Lopez, C. Tranoris, I. Vidal Fernández, and A. Gavras, Experimentation over Distributed 5G NFV-Based Environments, 5GinFIRE Whitepaper, December 2019.
- [14] A. P. Silva, C. Tranoris, S. Denazis, S. Sargento, J. Pereira, M. Luís, ... and D. Simeonidou, "5GinFIRE: An End-to-End Open5G Vertical Network Function Ecosystem," *Ad Hoc Networks*, Vol. 93, 101895, October 2019. <https://doi.org/10.1016/j.adhoc.2019.101895>
- [15] B. O. Kwak, H. Y. Ryu, and T. Y. Kim, "Enabler Development Infrastructure Design for Hyper-Connected Common Networking Services," in *Proceedings of International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, pp. 1323-1325, October 2021. <https://doi.org/10.1109/ICTC52510.2021.9620785>
- [16] B. O. Kwak, H. Y. Ryu, and T. Y. Kim, "Enabler Development Platform for Hyper-Connected Common Networking Services," in *Proceedings of the 13th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, pp. 2309-2311, October 2022. <https://doi.org/10.1109/ICTC55196.2022.9953000>



심재찬(Jaechan Shim)

1992년 : 충남대학교 학사 (전산학사)
1994년 : 충남대학교 대학원 (전산학석사)
2016년 : 전남대학교 대학원 (컴퓨터공학박사)

1994년~현 재: 한국전자통신연구원 책임연구원
※관심분야 : 클라우드 컴퓨팅, 고가용(HA) 시스템 등



류호용(Hoyong Ryu)

1991년 : 광운대학교 학사 (통신공학사)
1993년 : 광운대학교 대학원 (전자통신공학석사)
1999년 : 광운대학교 대학원 (전자통신공학박사)

1998년~현 재: 한국전자통신연구원 책임연구원
※관심분야 : 클라우드 컴퓨팅, 고가용(HA) 시스템, 디지털 트윈 네트워크 등



윤호선(Hosun Yoon)

1997년 : 순천향대학교 학사 (전자공학사)
1999년 : 순천향대학교 대학원 (전자공학석사)
2016년 : 순천향대학교 대학원 (정보보호박사)

2000년~현 재: 한국전자통신연구원 책임연구원
※관심분야 : 클라우드 컴퓨팅, 초저지연 네트워킹 등



문성(Seong Moon)

1996년 : 한남대학교 학사 (전산학사)
1999년 : 충남대학교 대학원 (컴퓨터공학석사)

2000년~현 재: 한국전자통신연구원 책임연구원
※관심분야 : 클라우드 컴퓨팅, 고속 네트워킹 등



박평구(Pyungkoo Park)

1999년 : 고려대학교 학사 (컴퓨터공학사)
2002년 : 고려대학교 대학원 (전산학석사)
2009년 : 충남대학교 대학원 (컴퓨터공학박사)

2001년~현 재: 한국전자통신연구원 책임연구원
※관심분야 : 클라우드 컴퓨팅, 5G, 6G 이동통신 등



곽병옥(Byeongok kwak)

1994년 : 청주대학교 학사 (전산학사)
1996년 : 충북대학교 대학원 (전산학석사)
2017년 : 충북대학교 대학원 (전산학박사)

2000년 10월~현 재: 한국전자통신연구원 책임연구원
※관심분야 : 클라우드, 가상화, ICT R&D, 미래인터넷 등