스마트 블록체인 기반 암호화폐 결제 서비스 애플리케이션 개발

*강태신 **최지원 ***이태규

평택대학교

*teri98@ptu.ac.kr **tina6900@ptu.ac.kr ***tglee@ptu.ac.kr

Development of a Smart Blockchain-based Cryptocurrency Payment Service Application

*Kang, Tae-Shin **Choi, Ji-Won ***Lee, Tae-Gyu

Pyeongtaek University

요약

최근 다양한 산업분야에서 디지털 사용자 트랜잭션의 보안성 강화와 디지털 자산의 거래 비용 최소화를 동시에 실현하기 위해 블록체인 기술을 점진적으로 확대하고 있다. 특히 무인 자동화 시스템으로의 전환이라는 과도기적 상황과 코로나-19 대유행이 맞물려 비대면 시장에 대한 관심이 커졌고, 비대면 시장의 수익 극대화와 안정적인 비대면 서비스 실현을 위해 블록체인 시스템과의 결합이 초점화되고 있다. 본 캡스톤디자인 보고서는 블록체인 기반 암호화폐 결제 서비스 애플리케이션을 설계하고 구현하는 과정을 기술한다. 특히 새로운 알트코인으로 평택코인과 관련된 앱 생태계 구축 과정, 그리고 평택코인 결제 서비스 지원 애플리케이션을 개발을 통한 스마트 암호화폐 결제 애플리케이션의 서비스 초기 모델 구현 과정에 대해 기술한다.

1. 작품의 개발 동기

최근 금융 분야를 비롯한 여러 산업 분야에서 블록체인 기술을 이용하여 사용자의 디지털 결제 트랜잭션의 무결성 및 여러 보안 요소를 강화하고 있다. 이와 동시에 거래 비용을 최소화하기 위해 금융 분야를 비롯하여 다양한 산업 분야에 블록체인 기술의 적용점을 점진적으로 확대하고 있다. 이러한 금융 시장의 적극적인 투자 속에서 전 세계적으로 디지털 화폐의 상용화에 관한 연구가 진행 중이고, 여러 전문가들이 10년 이내에 디지털 화폐가 법정 화폐를 대체할 것으로 예측하고 있다. 국내에서도 디지털 화폐에 관해 연구 및 검토가 진행 중이나, 다른 선진국들에 비해서는 아직 부족한 상황이다. 따라서 본 작품에서는 디지털 화폐, 스마트 결제 시스템에 대응하기 위해 스마트 블록체인 기반 암호화폐 결제 서비스 애플리케이션 개발에 초점을 맞춘다. 특히 KIOSK, POS 등의 무인 판매 및 결제 시스템에 암호화폐 결제 모듈을 이식하여 암호화폐를 통한 실시간 결제 서비스를 제공하는 것을 목표로 한다.

본 캡스톤디자인 보고서 먼저 2장에서 작품의 개발 방법과 개발 프로세스에 관해 기술한다. 3장에서는 작품의 구현 결과를 그림을 중심으로 배치하고, 그림에서 보이는 결과에 관해 설명한다. 마지막으로, 4장에서 본 작품에 대한 결론과 기대효과를 기술한다.

2. 개발 방법 및 개발 프로세스

본격적인 개발 방법 및 개발 프로세스 설명에 앞서 먼저 개발을 진행한 PC 환경을 소개한다. 개발 PC의 환경은 다음과 같다.

1) OS: Windows 10 Pro/Linux Ubuntu 20.04 LTS
2) CPU: AMD Ryzen 9 3900X 12-Core Processor 3.80GHz
3) RAM: Corsair Vengeance RGB PRO 64GB 3600Mhz
4) GPU: NVIDIA GeForce RTX 3060 Ti 8GB
5) Storage: Samsung PM981A 1TB M.2 NVMe x 2
6) Power: SEASONIC FOCUS GOLD GM-750 Modular

본 작품은 크게 4개의 부분으로 나누어 개발하였다. 다음 표 1은 각 개발 분야의 개발환경을 표로 나타낸 것이다.

작품	Pyeongtaek coin Core	Pyeongtaek -wallet	평택코인 Web	평택코인 KIOSK
운영체제	Linux Ubuntu 20.04 LTS	Windows 10 Pro	Windows 10 Pro	Windows 10 Pro
기반 소스 코드	Litecoin Core v0.18	Loafwallet-android v0.19	-	-
프로그램 언어	C++	Java, C	Javascript (Node.js)	Java, C
컴파일러	gcc+	C, java compiler	-	C, java compiler
개발 도구	Bash, Visual Studio Code	Android Studio, Visual Studio Code	Visual Studio Code, cmd	Android Studio, Visual Studio Code
동작 환경	Linux, Mac OS, Windows	Android	웹이 동작하는 모든 기기	Android

표 1 각 분야별 개발환경

본 작품의 최종목표인 블록체인 기반 암호화폐 결제 서비스 애플리케이션 개발을 위해서는 새로운 암호화폐 개발 및 생태계 구축이 요구된다. 따라서 우리는 새로운 암호화폐인 평택코인을 개발하였고, 평택코인과 관련된 앱 생태계를 구축하였다. 평택코인은 라이트코인 기반 파생 암호화폐이다. 라이트코인은 2011년 10월에 MIT 출신의 Charlie Lee가 개발한 P2P 암호화폐의 하나로 비트코인과 유사한 방식으로 운영되는 암호화폐이다. 라이트코인은 비트코인보다 작고 가벼운 암호화폐를

추구하며 개발되었으며, 비트코인의 오픈소스 코드를 포크하여 개발하였다. 즉, 라이트코인은 비트코인의 대안화폐(Altcoin)이며, 평택코인도 비트코인과 라이트코인의 대안화폐이다. 평택코인은 기존의 암호화폐의 장점은 살리고, 단점 및 한계점은 일부 보완하여 매개변수를 새로 정의하였다. 다음 표 2는 평택코인과 주요 암호화폐 간 매개변수의 비교를 나타낸다. 다음과 같이 설계된 평택코인의 매개변수를 기반으로 평택코인 코어를 구축하고, 확장 및 응용 애플리케이션을 개발하였다.

매개변수 \ 코인이름	Bitcoin	Litecoin	Pyeongtaekcoin
통화단위	BTC	LTC	PTC
	mBTC	mLTC/lites	mPTC
	µBTC	µLTC/phothons	µPTC
	satoshi	litoshi/bits	pyeongtoshi
블록 생성 시간	10Min	2.5Min	1Min
최대 블록 크기	1MB	4MB	4MB
총 발행량	21,000,000	84,000,000	10,612,000,000
초기 채굴 보상	50	50	1000
반감기	210,000블록	840,000블록	존재하지 않음
P2P 포트번호	8333	9333	3333
RPC 포트번호	8332	9332	3332
코인 성숙도	100	100	100

표 2 주요 암호화폐와 평택코인 간 비교

2.1 Pyeongtaekcoin-core

평택코인 코어는 블록체인 기반 암호화폐 결제 서비스 애플리케이션 개발의 근간이 되는 부분으로 PC에서 구동되는 클라이언트이다. 평택코인 코어는 블록체인의 모든 트랜잭션과 블록을 저장하는 풀 노드(Full Node)이다. 평택코인 코어는 라이트코인 코어의 오픈소스 코드를 포크하여 개발하였다.

평택코인 코어의 개발 프로세스는 크게 라이브러리 설치 등의 개발 환경 구축, 이름 및 매개변수 변경 등을 포함한 평택코인 개발, 컴파일 및 클라이언트 실행 테스트로 나눌 수 있다. 평택코인 코어의 개발 프로세스는 다음과 같다.

1) 패키지 및 컴파일 도구 설치

소스코드 컴파일 및 빌드를 위한 컴파일 도구와 패키지를 설치한다. 다음 표 3은 평택코인에 사용되는 의존 라이브러리를 나타낸다.

라이브러리	목적	설명
libssl	암호화	난수 생성, 타원 곡선 암호
libboost	유틸리티	스레딩, 데이터 구조 등을 위한 라이브러리
libevent	네트워킹	운영체제 독립 비동기 네트워킹
miniupnpc	UPnP 지원	방화벽 통과 지원
libdb4.8	Berkeley DB	지갑 저장소(지갑 활성화 시 필요)
qt	GUI	GUI 툴킷(GUI 활성화 시 필요)
protobuf	지불	지불에 사용되는 데이터 교환형식
libqrencode	QR 코드 생성	QR 코드 생성(GUI 활성화 시 필요)
univalue	유틸리티	JSON 파싱 및 인코딩
libzmq3	ZMQ 알림	ZMQ 알림 생성 허용

표 3 평택코인 빌드에 사용되는 의존 라이브러리

2) 라이트코인 오픈소스 다운로드 및 코인 이름 변경

라이트코인 소스를 내려받고, Bash 명령어를 활용하여 코인의 이름을 평택코인으로 일괄 변경한다. 화폐단위도 모두 변경한다.

3) P2P, RPC 포트 및 매직넘버 변경

포트넘버와 매직넘버를 평택코인 설계에 맞게 변경한다.

4) 제네시스 블록 생성 및 매개변수 변경

제네시스 블록 생성 프로그램을 이용하여 설계된 매개변수에 맞추어 제네시스 블록을 생성한다. 제네시스 블록은 블록체인의 최초 블록으로 위조 및 변조될 수 없다. 생성된 제네시스 블록을 평택코인 코드 내에 하드 코딩한다.

6) DNS 시드 노드 및 체크 포인트 제거

라이트코인을 포크하여 개발하고 있으므로 기존의 DNS 시드 노드 및 체크 포인트는 라이트코인의 값이다. 평택코인은 새로운 코인이므로 이를 모두 제거한다. 평택코인이 지속적으로 업데이트 될 때마다 체크 포인트가 추가될 것이며, 사용 노드가 많아진다면 네트워크 이용자의 ip를 제공하는 DNS 시드 노드가 추후에 구축될 것이다.

7) 매개변수를 평택코인의 설계에 맞추어 변경

블록 생성 시간, 최대 블록 크기, 총 발행량 등의 설계한 매개변수를 바탕으로 평택코인의 소스코드를 갱신한다.

8) 아이콘 변경

라이트코인의 아이콘을 평택코인에 맞추어 변경한다. 평택코인의 앞 글자를 딴 알파벳 P를 로고로 활용한다.

9) 컴파일 및 빌드

컴파일 및 빌드를 수행하여 응용 프로그램을 만들어낸다. 이는 Linux에서만 응용 프로그램이 정상적으로 구동되기 때문에 다른 운영체제에서 작동하는 프로그램을 만들기 위해서는 크로스 컴파일이 요구된다. 크로스 컴파일을 통해 Windows, Mac OS, Linux ARM에서 구동 가능한 프로그램을 추가적으로 빌드하였고, 이는 정상 작동한다.

10) 평택코인 클라이언트 실행 및 테스트

평택코인 클라이언트를 여러 운영체제에서 실행하고, 각 클라이언트를 연결하여 테스트한다. 테스트에는 pyeongtaek-wallet과 Web 인터페이스 간의 상호 연결을 포함한다. 각 클라이언트는 TCP 연결을 기반의 분산 합의 메커니즘을 통해 시스템을 운영해나간다. 각 노드는 채굴을 수행할 수 있고, 작업 증명 알고리즘을 통해 블록의 유효성을 검증한다.

2.2 Pyeongtaek-wallet

Pyeongtaek-wallet은 SPV 기반 모바일 클라이언트로 loafwallet의 안드로이드 오픈소스에 기반한다. SPV(Simple Payment Verification) 노드는 모든 블록체인을 저장하지 않고도 트랜잭션을 검증하는 노드이다. 스마트폰을 비롯한 휴대용 기기 또는 IoT 기기에 블록체인의 모든 트랜잭션과 블록을 저장하는 것은 어려움이 따르기에 일반적으로 이런 기기에는 SPV 기반의 클라이언트를 사용한다. loafwallet은 라이트코인에서 공식으로 지원하는 SPV 기반 모바일 애플리케이션으로 breadwallet이라는 오픈소스를 포크하여 개발하였다.

Pyeongtaek-wallet의 개발 프로세스도 평택코인 코어와 마찬가지로 크게 SDK 설치 등의 개발 환경 구축, 이름 및 매개변수 변경 등을 포함한 평택코인 개발, 컴파일 및 클라이언트 실행 테스트로 나눌 수 있다. Pyeongtaek-wallet의 개발 프로세스는 평택코인 코어의 개발과정과 유사하나, Pyeongtaek-wallet은 풀 노드가 아닌 경량 노드이기 때문에 평택코인 코어에 존재하는 모든 매개변수가 존재하지 않고, 개발 언어도 다르기 때문에 일부분에서 차이점이 존재한다. 예를 들어, Pyeongtaek-wallet은 빌드를 위해 라이브러리를 사용하는 대신 안드로이드 스튜디오를 통해 SDK와 NDK를 설치하고, 이후 개발에 필요한 SDK와 NDK의 버전을 설정한다. 또한 컴파일 및 빌드를 수행하여 만들어지는

설치 프로그램의 형식이 안드로이드 운영체제에서 동작하는 apk이다. Pyeongtaek-wallet을 테스트할 때는 평택코인 코어와의 연동이 정상적으로 되는지, 그리고 애플리케이션 내의 기능들이 정상적으로 작동하는지를 중점적으로 확인한다. 예를 들어 QR코드를 찍어 해당 주소로 평택코인이 정상적으로 전송되는지를 확인한다.

2.3 평택코인 Web

평택코인 코어는 강력한 기능을 제공하지만, 블록체인 분야의 전문가 또는 숙련가가 아니라면 코어의 기능을 이해하고 사용하기 쉽지 않고, 사용자 인터페이스의 직관성이 떨어진다는 문제점이 있다. 이를 해결하기 위해 평택코인 코어의 일부 기능을 평택코인의 RPC-API를 이용하야 Web으로 구현한다. Web으로 구현하기 때문에 인터페이스의 수정 및 기능 추가가 자유롭다는 특징을 지닌다.

평택코인 Web을 구현하기 위해서는 평택코인 Web 인터페이스를 출력하기 위한 메인 웹서버와 RPC-API를 호출하기 위한 API서버, 이렇게 두 개의 서버를 구축해야 한다. 두 개의 서버는 모두 node.js로 구현하였으며, 프론트엔드에서 fetch API를 사용하여 비동기 통신을 구현하였다. 평택코인 Web의 프론트엔드는 fetch API를 활용하여 사전에 구축한 API 서버와 통신하고, API 서버는 RPC-API를 통해 평택코인 코어 클라이언트에 프론트엔드로부터 요청받은 정보를 요청한다. 평택코인 코어 클라이언트로부터 응답이 정상적으로 도착했다면, 이를 다시 프론트엔드에 전송하는 중개 임무를 수행한다. 평택코인 Web의 프론트엔드는 응답받은 데이터를 사전에 정의된 규칙에 맞게 Web 페이지에 출력한다.

2.4 평택코인 기반 무인 판매 및 결제 시스템 (KIOSK)

평택코인 기반 무인 판매 및 결제 시스템은 판매 시스템을 통해 구매할 상품을 고르고, 고른 상품을 평택코인으로 결제한다. 기본 시스템은 기존의 무인 판매기와 유사하나, 블록체인 기반의 분산 결제 프로토콜을 사용한다는 차이점이 존재한다.

평택코인 기반 무인 판매 및 결제 시스템은 안드로이드 환경에서 구동된다. 즉, 이 시스템을 사용하려는 KIOSK와 POS 등에서 안드로이드 운영체제를 지원해야 사용할 수 있다. 본 작품에서 사용한 결제 방식은 KIOSK에 QR코드를 출력하고, 소비자가 Pyeongtaek-wallet을 실행하여 QR 코드를 촬영하는 방식이다. QR 코드를 Pyeongtaek-wallet에서 인식한 즉시, 판매자의 지갑 주소와 결제 금액이 소비자의 애플리케이션에 자동으로 입력되며, 소비자는 보내기 버튼을 눌러 편리하게 결제할 수 있다.

3. 구현 결과

서비스 유지 및 정상적인 서비스 제공을 위해 평택코인 코어 클라이언트는 평택코인 네트워크 내에서 상시 동작해야 하며, 2개 이상의 노드가 지속적으로 운영되어야 한다. 이를 실현하기 위해 워크스테이션 2대에 평택코인 코어를 설치하고, 이를 지속적으로 운영하고 있다.

다음 그림 1, 그림 2는 평택코인 코어의 GUI를 통해 평택코인 코어 클라이언트가 정상 동작함을 나타낸다. 그림 3에서 평택코인 코어가 설치되어 있는 워크스테이션과 정상적으로 네트워크에 연결됐음을 확인할 수 있다.

그림 1 평택코인 코어 메인 인터페이스

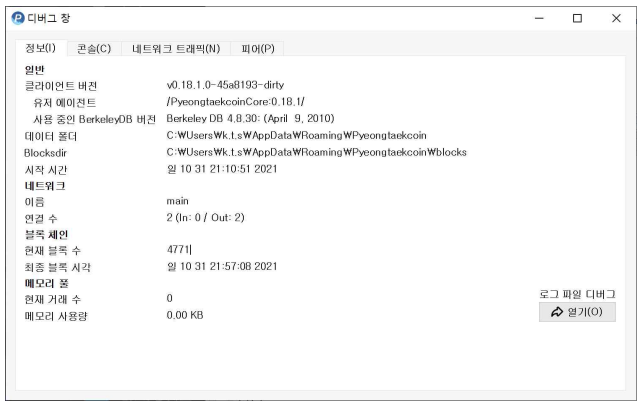

그림 2 평택코인 코어의 상태 정보 확인 인터페이스

그림 3 평택코인 코어 클라이언트에 연결된 노드 확인 인터페이스

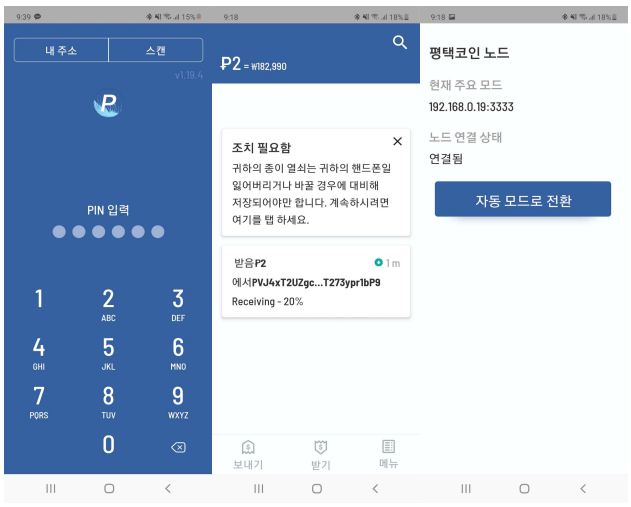

그림 4 Pyeongtaek-wallet 메인 인터페이스 및 연결 상태 화면

그림 4, 그림 5는 Pyeongtaek-wallet의 내부 인터페이스와 기능을 나타낸다. Pyeongtaek-wallet은 보안을 위해 애플리케이션에 접속할 때 PIN 번호 또는 지문인식으로 잠금을 해제해야 한다. 잠금을 해제하면 Pyeongtaek-wallet 내에 들어있는 잔액과 내 최근 트랜잭션 내용이 출력된다. 받기 탭을 누르면 Pyeongtaek-wallet의 지갑주소가 출력되고, 지갑주소와 지불 요청 금액이 QR코드로 변환된다. 보내기 탭에서는 자

신의 지갑주소에서 다른 사람의 지갑으로 송금할 수 있는 서비스를 제공한다. 스캔 버튼을 눌러 수신자의 QR코드를 촬영한 즉시 판매자의 지갑 주소와 판매자가 지불 요청한 금액이 자동으로 입력된다.

그림 5 Pyeongtaek-wallet 받기 보내기 인터페이스

그림6, 그림7은 평택코인 Web의 내부 인터페이스를 나타낸 것이다. 홈 인터페이스는 평택코인 코어의 메인 인터페이스와 유사하지만, 부가적인 기능이 추가되었다. 예를 들어 메인 페이지에서 블록체인 정보 자세히 보기를 클릭하면 알림창 형태로 블록체인 정보에 관한 내용이 출력된다. 다른 탭도 마찬가지로 평택코인 코어의 각 탭과 유사한 기능을 제공하며, 일부 부가적인 기능을 제공한다. 예를 들어 트랜잭션 탭에서 조회되는 트랜잭션의 수를 지정할 수 있는 부가적인 기능을 제공한다.

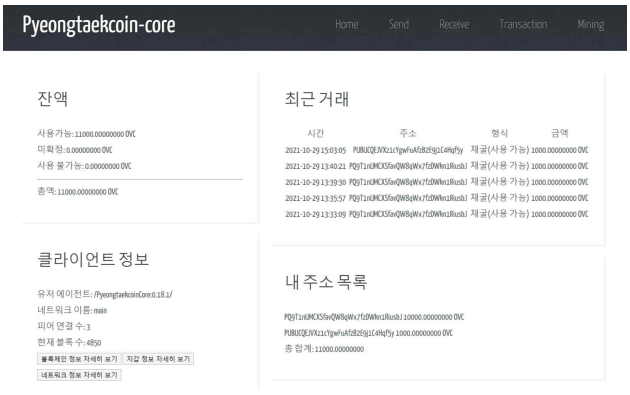

그림 6 평택코인 Web 홈 인터페이스

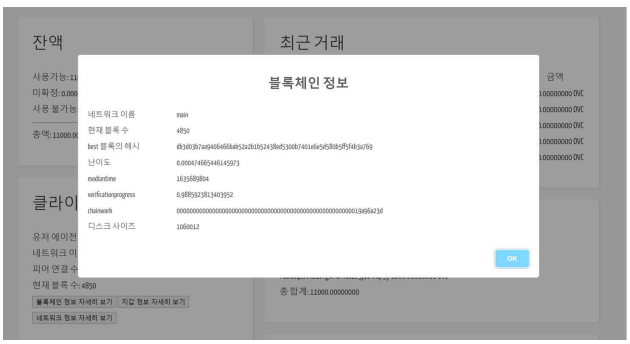

그림 7 평택코인 Web 정보 알림창

그림 8은 평택코인 Kiosk의 내부 인터페이스와 기능을 나타낸다.

첫 번째 인터페이스는 주문 방법을 선택하는 UI를 제공한다. 두 번째 인터페이스는 Scrollview를 통해 메뉴를 수평 방향으로 넘겨 볼 수 있으며 선택한 음료는 주문 리스트 view로 메뉴 이름과 가격을 확인할 수 있다. 주문을 완료하면 QR결제 버튼을 클릭하고 총 금액과 총 금액에 대한 QR코드가 보여진다. 평택코인 Kiosk는 Pyeongtaek-wallet과 같이 출력된 QR코드를 촬영한 즉시 소비자의 지갑에서 키오스크로 총 금액을 송금할 수 있다.

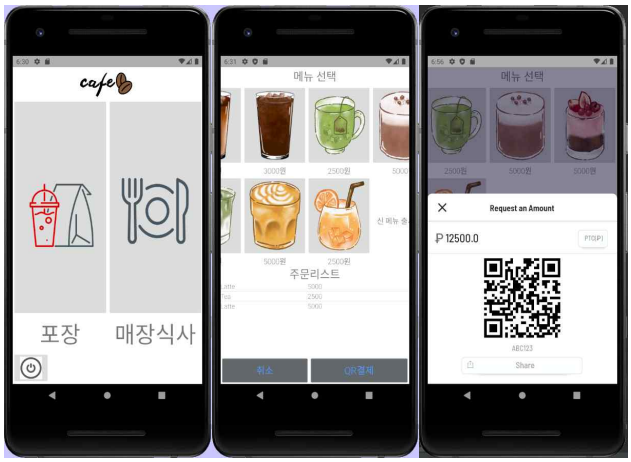

그림 8 평택코인 Kiosk 주문 및 결제 인터페이스

4. 작품의 기대효과

본 작품은 기존 암호화폐를 포크하여 새로운 알트코인인 평택코인을 개발하고, 이에 관련된 앱 생태계를 구축하였다. 또한 KIOSK, POS 등에 적용 가능한 결제 서비스 및 결제 서비스 지원 애플리케이션을 개발함으로써 스마트 암호화폐 결제 애플리케이션 서비스 초기 모델을 구현했다. 이러한 초기 모델의 기능을 고도화하여 스마트 계약, 스마트 거래, 스마트 결제 등의 스마트 서비스로 확장할 수 있을 것으로 기대된다. 또한 이종 자산 간의 교환 서비스 구현을 통해 최근 주목받고 있는 NFT, 분산 자산거래, 메타버스 등의 다양한 모델의 인증 및 결제 서비스 지원을 위한 초기 스마트 앱 플랫폼으로 확장할 수 있을 것으로 보인다. 향후 개발은 평택코인 플랫폼을 실제 무인 판매 및 결제 시스템에 적용모델을 구현한다. 이후, 암호화폐 거래소와 유사한 기능을 수행하는 이종 자산 간의 교환 서비스 개발을 통해 평택코인 플랫폼을 NFT, 메타버스 등 스마트 플랫폼에서의 거래로 확장하려 한다.

※ 본 과제(결과물)는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된 사회맞춤형 산학협력 선도대학(LINC+) 육성사업의 연구결과입니다.