

## A Study on the API Gateway for human resources management modules extensions in ERP

Ji-Woon Lee\*, Hee-Suk Seo\*

\*Student, Dept. of Computer Engineering, Korea University of Technology and Education, Chungcheongnam-do, Korea

\*Professor, Dept. of Computer Engineering, Korea University of Technology and Education, Chungcheongnam-do, Korea

### [Abstract]

In this paper, we propose a API Gateway technique for the expansion of human resource management module, one of the ERP functions. The institution has introduced ERP (Enterprise Resource Planning) based on its efforts to transform all human and physical resources into business competitiveness and its response to the digital knowledge informatization environment, and listed it as multiple success factors. Human resource management is one of the factors that have been dealt with. However, ERP's Human Resources Management Module remains in the role of functional personnel management. How to utilize human resources begins with navigating and recognizing human resources. The proposed API Gateway technique leverages blockchain networks to design and implement APIs for human resource sharing and navigation, including the possibility of extending ERP's human resource management module. Secondly, it was designed and implemented using a smart contract that behaves like an API for preventing information forgery. The proposed method will not only be used as a tool that can actively utilize human resources, but will also be a complete resource for utilizing big data technology.

▶ **Key words:** ERP, Human resource, Micro Service Architecture, API, Blockchain

### [요 약]

기관은 모든 인적, 물적 자원을 비즈니스 경쟁력으로 변환하기 위한 노력과 디지털 지식정보화 환경의 대응에 따라 ERP(Enterprise Resource Planning)을 도입 하였고 여러 성공 요인으로 열거된 요인 중에는 인적자원관리가 존재 한다. 하지만 ERP의 인적자원관리 모듈은 기능적 인사관리 역할에 머물러 있다. 인적자원을 어떻게 활용할 것인가는 인재를 탐색하고 인식하는 것으로 출발하는 것으로 시작한다. 제안한 API Gateway 기법은 ERP의 인적자원관리 모듈의 확장의 가능성을 시작으로 첫 번째, 블록체인 네트워크를 활용해 인적자원의 공유와 탐색을 위한 API의 설계와 구현을 하였고 두 번째로 정보의 위변조 방지를 위해 API와 같이 동작하는 스마트 컨트랙트(Smart Contract)를 함께 설계 및 구현 하였다. 제안한 기법은 인적자원을 적극적으로 활용할 수 있는 도구로 사용할 수 있을 뿐만 아니라 빅 데이터 기술을 활용하기 위한 완벽한 자원이 될 것이다.

▶ **주제어:** 전사적 자원 관리, 인적자원, 마이크로 서비스 아키텍처, API, 블록체인

- 
- First Author: Ji-Woon Lee, Corresponding Author: Hee-Suk Seo
  - \*Ji-Woon Lee (jwlee7@koreatech.ac.kr), Dept of Computer Engineering, Korea University of Technology and Education
  - \*Hee-Suk Seo (histone@koreatech.ac.kr), Dept of Computer Engineering, Korea University of Technology and Education
  - Received: 2021. 01. 06, Revised: 2021. 02. 03, Accepted: 2021. 02. 03.

## I. Introduction

기관은 모든 인적, 물적 자원을 궁극적인 비즈니스 경쟁력으로 변환하기 위해 수많은 노력을 하고 있다. 대표적으로 기관 내의 모든 업무를 조화롭게 제대로 발휘 할 수 있도록 최신 정보기술을 이용하여 업무와 관련된 프로세스를 위한 소프트웨어를 냈고 나아가 소프트웨어의 집합시키고 통합하는데 까지 이르게 되었다.

비즈니스 환경이 디지털 지식정보화 환경으로 돌입하면서 여러 성공 요인으로 열거된 사항 중 인적자원관리가 있다. 이는 기존의 단순한 인력채용과 급여 관련 업무 등의 보편적 업무가 아닌 기관의 전략과 연계된 광범위하고 중요한 의미를 포함하고 있다고 할 수 있다. 관리적 대상이 아닌 자원으로 계획적이고 효율적으로 운영이 필요한 자원으로서의 인식과 함께 많은 요구사항을 받아들여 인적 자원 관리 시스템을 시작으로 기관의 전사적 자원 관리(ERP)의 영역으로 포함과 관련 연구[1,2,5,6]가 진행 되었으나 기능적 인사관리로서의 역할만을 수행하고 있다.

인적자원을 어떻게 활용할 것인가는 인재를 탐색하고 인식하는 것으로부터 출발한다. 이를 국가적 차원으로 접근한 것이 ‘국가인재데이터베이스’ 이지만 민간 차원으로는 각 기관별로 축적하는 정도로 머물고 있는 것으로 알려져 있다. 데이터베이스에 있는 인적자원정보는 ‘축적 → 탐색과 공유’ 단계로 변화가 필요한 것이다. 나아가 축적된 정보는 정제되고 유통을 위한 창구를 위한 정문 역할 또한 필요할 것으로 보인다.

본 연구에서는 ERP의 한 모듈인 HR Management을 확장하기 위한 요소로서 인적자원공유 네트워크 형성, 추가적인 기능 확장을 위한 APIs, 정보의 위변조 방지[7-8]를 MSA(Micro Service Architecture)를 활용한 APIs Gateway을 제안 제안하고자 한다.

## II. Preliminaries

### 1. Related works

#### 1.1 ERP (Enterprise Resource Planning)

전사적 자원 관리(ERP)는 경영 정보 시스템의 한 종류로서 기관의 모든 정보뿐만 아니라 효율화의 중심에 있는 수단이자 소프트웨어이다. ERP는 ‘Enterprise Resource Planning’의 약어로서 기관 전체에서 사용되는 정보가 정규화 되고 공통의 정의 및 경험을 기반으로 한 비즈니스 활동을 관리한다. 핵심 구성요소는 재무, 인적자원, 엔지

니어링, 마케팅, 운영 등이며 이를 사용하는 사람들의 워크플로우(Work Flow)에 의해 구동되는 비즈니스 프로세스와 상호 연결을 연결하는 프로세스 및 기술의 통합 수단이기도 하다.

Table 1. Definitions of ERP[2]

Name	Definitions
Gartner Group	a series of applications designed to bring business functions into balance and represents the next generation of systems
APICS : American Production and Inventory Control Society	accounting-oriented information system for identifying and planning the enterprise-wide resources needed to take, make, ship, and account for customer orders

ERP는 기존 업무 처리방식에 대한 근본적인 변화를 요구하며 다양한 영역의 프로세스 통합관리가 Table 2 와 같이 이루어졌고 개선된 효과는 첫 번째 부서간의 실시간 정보공유, 두 번째 표준화된 단일 시스템을 통한 데이터의 일관성, 세 번째 실시간 정보조회로 업무능력 향상, 경쟁력 확보, 의사결정 지원 이다.

Table 2. Major Modules of ERP (SAP)[3]

Group	Item
Logistics	Sales & Distribution
	Material Management
	Production Planning
	Quality Management
	Plant Management
	Project System
HR Management	Human Resource
Accounting	Financial
	Controlling
	Treasury
	Investment Management
	Real Estate Management

프로세스의 통합관리 외에 ERP의 성공적 도입 및 효과와 관련된 연구 중에는 인적자원의 관리가 필수적인 요인으로 열거가 된바 있다[1].

#### 1.2 Module of HR Management in ERP

인적자원관리는 기관이 목표를 이루기 위해 필요한 인적자본을 확보·개발·활용을 계획하는 일련의 과정이며 관리적 전략을 의미하며 다음과 같은 목표를 갖는다고 할 수 있다. 첫 번째 인재를 통하여 조직경쟁력의 강화, 두 번째

인적자원관리 시스템의 설계, 구축, 운영 세 번째 조직목표와 조직구성원목표의 일치, 네 번째 지속적인 혁신을 좋은 환경과 여건 생성[5-6]이며, HR Management 모듈 역시 관리적 전략과 목표를 지원 할 수 있도록 다양한 기능들[3-4]을 Table 3 과 같이 제공하고 있다.

Table 3. Summary of functions of HR Management module[3-4]

Group	Item
Employee	Employee Employment Type Branch Department Designation Employee Grade Employee Group Employee Health Insurance
Attendance	Attendance Employee Attendance Tool Attendance Request Upload Attendance Auto Attendance
Leaves	Leave Management Holiday List Leave Type Leave Period Leave Policy Leave Allocation Leave Application Compensatory Leave Request Leave Encashment Leave Block List
Recruitment	Staffing Plan Job Opening Job Applicant Job Offer Appointment Letter
Training	Training Program Training Event Training Result Training Feedback
Employee Lifecycle	Employee Lifecycle Management Employee Onboarding Employee Promotion Employee Separation Employee Transfer Employee Skill Map
Payroll	Payroll Management Payroll Period Income Tax Slab Salary Component Salary Structure Salary Structure Assignment Salary Slip Payroll Entry Additional Salary Retention Bonus Employee Incentive
Performance	Appraisal Energy Point System

Expand Claims	Expense Claim Employee Advance
Shift Management	Shift Management

HR Management 모듈을 살펴보면 일반적으로 기능적 인사관리 수준을 제공하는 것을 알 수 있으며, 인적자원정보를 적극적으로 활용할 수 있는 혹은 확장할 수 있는 보완책이 필요하다[7].

### 1.3 MSA

MSA(Micro Service Architecture)는 서비스 지향 아키텍처인 SOA(Service Oriented Architecture)을 근간을 두고 있다. SOA가 엔터프라이즈 중심으로 고안 되었다면, MSA는 많은 양의 처리를 위한 서비스 개발에 맞게 경량화 되고, 대규모 개발 조직에 맞도록 변형된 아키텍처이다.

이러한 구조는 독립적으로 분리된 소규모 서비스나 비즈니스 단위로 연결이 가능한 독립적인 규모로 구성이 가능하여 배포 단위를 다양하게 설계할 수 있다. 분리된 서비스 간의 통신은 비동기 메시지를 활용하거나 도메인 특정 프로토콜, Https 등을 사용할 수 있다[9].

### 1.4 API Gateway

API Gateway는 어떤 규모와 형태든 API를 손쉽게 생성, 게시, 유지 관리, 모니터링을 할 수 있는 완전관리형 서비스에 해당 한다. API Gateway는 애플리케이션이 Front-end와 Back-end 서비스의 데이터, 비즈니스 로직 또는 기타 기능에 액세스할 수 있는 정문 역할을 할 수 있다. API Gateway의 정문 역할을 하는데 있어 5가지의 주요한 특징을 지닌다. 첫 번째 인증 및 인가에 관련된 기능 두 번, End-Point별 API 호출 및 인증, 세 번째 API서버로의 로드 밸런싱, 네 번째 공통 비즈니스 로직 처리 다섯 번째 프로토콜 변환이다[9].

본 연구에서는 API Gateway가 가지는 특징 중 인증 및 인가, End-Point별 API 호출, 공통 비즈니스 로직을 적극적으로 이용하는 것이 적절하다.

## III. The Proposed Scheme

본 연구에서는 국내외에서 여러 기관(혹은 기업, 이하 기관)이 사용하고 있는 ERP의 인적자원 관리 모듈을 확장하는 방법으로서 Fig. 1 과 같은 APIs(Application Protocol Interfaces) Gateway를 제안하고자 한다.

이 구조의 특징으로는 국·내외에서 사용되는 ERP 시스템의 인적자원 관리 모듈에서 제공하지 않는 사항과 부가 사항을 탄력적으로 확장 및 개선 할 수 있고, 정보의 검정 처리된 인적자원 정보 네트워크 형성과 정보의 위·변조 방지 대책[8]을 동시에 만족하는 이점을 가진다. 또한 API Gateway를 통해 MSA의 이점을 승계하여 시스템적인 확장과 API 모듈의 확장을 모두 만족한다.

1. Concept of Basic Model

제안하는 API Gateway(이하 AG)는 Fig. 1과 같이 크게 2가지의 구성요소를 가진다. 첫 번째로 ERP의 인적자원 관리 모듈을 사용하는 기관으로부터 API 정보를 입력받아 처리하는 API Modules(이하 AM)와 검증 전 데이터와 임시 데이터를 저장하는 RDBMS이며, 두 번째는 데이터의 위·변조 방지와 네트워크 제공을 위한 블록체인(Hyperledger Fabric) 이다.

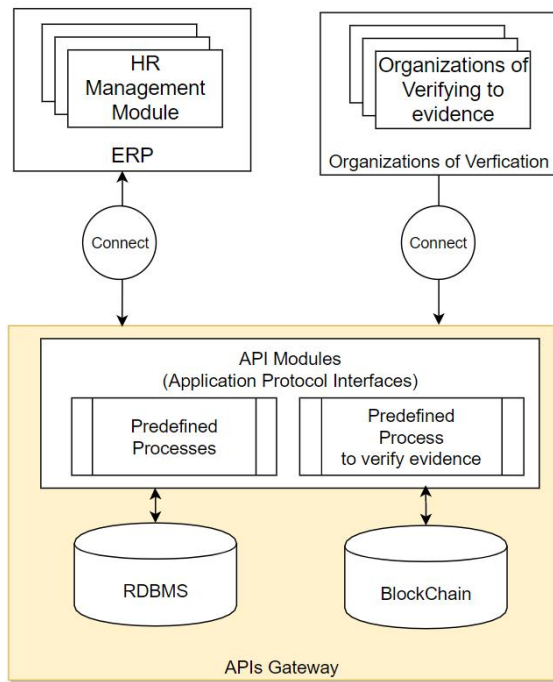


Fig. 1. System Architecture of Proposed Concept

AG의 전반적인 동작 구조는 크게 2가지로 구성하였다. 첫 번째는 최초로 ERP의 인적자원 관리 모듈을 이용한 정보의 입력이며, 두 번째는 입력이 완료된 정보의 조회이다. 정보가 입력되는 과정은 Fig. 2 와 같다.

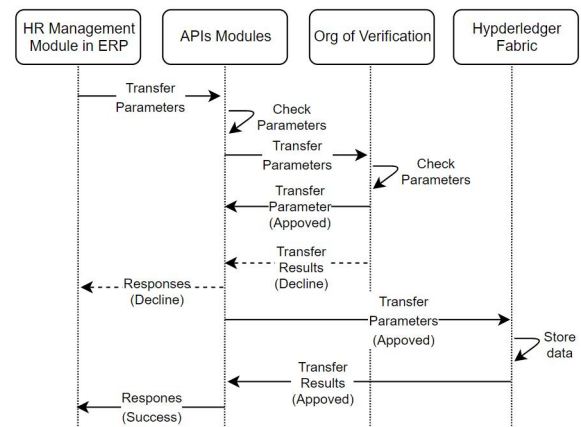


Fig. 2. Overall of Operation Flow (Input Data)

Fig. 2 와 같이 정보의 입력은 ERP 인적자원 모듈에서 전송된 Parameter Values는 AM의 RDBMS에 검증처리가 완료되기 전까지 임시 정보(Temporary data)로 취급되어 저장되며 해당 정보의 진위 여부를 판단할 수 있는 검증 기관으로 Parameter Values를 전송된다. 해당 정보를 수신한 검증 기관은 수신한 정보를 바탕으로 진위 여부를 판단과 승인을 하게 되면 AM으로 승인 정보와 함께 Parameter Values를 송신하고 마지막으로 AM은 위·변조 방지를 위해 블록체인으로 전송 후 저장한다. 그 외 정보의 갱신이 필요한 경우는 Fig. 2 와 동일한 과정을 거친다.

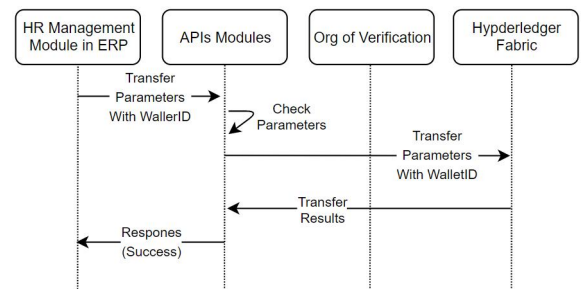


Fig. 3. Overall of Operation Flow (Inquiry Data)

검증 처리 및 승인이 완료된 정보의 조회는 조회 대상의 고유 식별 코드인 WallerID와 Parameter Values를 AM에 전송하여 직접 블록체인에 접근 후 해당 정보를 반환하며 그 과정은 Fig. 3 과 같다. 다만, 검증 처리가 완료되지 않는 정보는 RDBMS에서 정보를 반환한다.

2. Models of Data Parameters

AM에서 사용되는 데이터는 JSON(JavaScript Object Notation)을 차용하여 구성하였다. 데이터 오브젝트(Data Object)를 전달하기 위한 개방형 포맷 중 자료 형태에 대

한 큰 제한이 없고 플랫폼(ERP 인적자원 관리 모듈)에 독립적이므로 프로그램의 'Key-Value' 변수 값을 표현하고 확장하는데 가장 용이하다.

제안하는 Parameter의 구성은 Table 4 와 같이 기초적인 인적자원 정보인 학위, 자격/면허, 교육-훈련, 경력 등의 저장과 해당 정보를 검증하기 위한 요소를 복합적으로 고려한 주요 Parameter 이다.

Table 4. List of Data Parameters

Parameters	Type	Value
CareerNo	string	`json:"careerNo"`
OwnerNo	string	`json:"ownerNo"`
OwnerName	string	`json:"ownerName"`
OwnerWallet	string	`json:"ownerWallet"`
AcqDate1(start)	string	`json:"acqDate1"`
AcqDate2(end)	string	`json:"acqDate2"`
Description	string	`json:"description"`
DocsUrl	string	`json:"docsUrl"`
State	string	`json:"state"`
RegDate	string	`json:"regDate"`
CareerType	string	`json:"careerType"`
CareerName	string	`json:"careerName"`
AcqOrgName	string	`json:"acqOrgName"`
Reason	string	`json:"reason"`
CareerClass	string	`json:"careerClass"`
UpdateDate	string	`json:"updateDate"`
TrainType	string	`json:"trainType"`
OnoffType	string	`json:"onoffType"`
AcqOrgNo	string	`json:"acqOrgNo"`
TrainNo	string	`json:"trainNo"`

Table 5 와 같이 종류, 상태 등의 변경이 잦은 갱신 요소들에 대한 세부적인 값을 마련하였다.

Table 5. Details of Parameter Values

Parameter	Deatils of Values
careerType	0: Career, 1: Degree, 2: Qualification, 3: Educational Training
careerClass	100: Qualification, 101: License 200: Bachelor, 201: Master 202: Ph.d 300: Leaving comp. 301: In office, 400: Educational Training, 401: Education
state	0: Accepting, 1: In progress, 2: Rejected, 3: complete

최종적으로 블록체인에 저장이 완료된 데이터는 Fig. 4 와 같이 Transaction ID(\_id)와 Previous Hash ID(\_rev)를 포함하여 저장된 것을 확인 할 수 있다.

```
id "TTX_CAREER_100018"
{
  "id": "TTX_CAREER_100018",
  "key": "TTX_CAREER_100018",
  "value": {
    "rev": "4-b619a2b8c14a4187f8a8bb3f0375fd6"
  },
  "doc": {
    "_id": "TTX_CAREER_100018",
    "_rev": "4-b619a2b8c14a4187f8a8bb3f0375fd6",
    "acqDate1": "2020-10-10",
    "acqDate2": "NULL",
    "acqOrgName": "한국산업인력공단",
    "acqOrgNo": "8",
    "careerClass": "100",
    "careerName": "정보처리기사",
    "careerNo": "150",
    "careerType": "2",
    "description": "",
    "docsUrl": "https://www.proto-apps.site:7070/docs/1602581289275.txt",
    "onoffType": "0",
    "ownerName": "김철수",
    "ownerNo": "45",
    "ownerWallet": "indy_indv05@proto-apps.site",
    "reason": "null",
    "regDate": "2020-10-14",
    "state": "3",
    "trainNo": "0",
    "trainType": "0",
    "updateDate": "2020-10-13",
    "version": "\u0000Cg#BQQA="
  }
}
```

Fig. 4. Example Result of Generated Parameters

### 3. Register to use AG (Generating WalletID)

AG는 ERP의 인적자원 관리 모듈을 확장하기 위한 모델로서 첫 번째로 인증 및 인가와 관련된 사항이 필요하다. Hyperledger Fabric의 CA(Certificate Authority)를 이용해 인증 및 인가 기능으로서 사용할 수 있지만, 이는 기관 자체 국한된 인증에 해당하는 것으로 볼 수 있고 기관에 속해 있는 구성원(혹은 개인)을 제어하기에는 어려운 점을 가지고 있다.

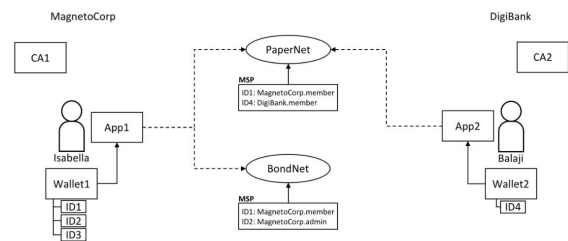


Fig. 5. The Importance of WalletID[10]

구성원(사용자)의 구분이 중요한 이유는 Fig. 5 에서 알 수 있다. 예를 들어, Fig. 5 의 ID1은 원장을 읽고 쓸 수 있는 기관의 구성원으로서 사용자를 인식하는 반면 ID2는 기관의 관리자로서 식별된다. WalletID 별로 API 및 Smart Contract를 실행을 다르게 할 수 있기 때문이다[10].

AG 또한 사용자 별(기관, 개인, 검증)로 사용할 수 있는 API 및 Smart Contract의 구분이 필요하기에 사용등록을 구성하였다. API와 Smart Contract를 포함하여 3단계로 진행되며 전반적인 순서는 다음과 같다.

- 1) '/join' 이라는 API를 호출 한다.
- 2) '/join'을 통해 호출된 등록 폼(Form)인 Fig. 6 에 나열된 정보를 입력 후 등록 버튼을 클릭 한다.

Fig. 6. Example Web Page of Registering User

- 3) Fig. 6 의 '등록하기' Table 6 와 Table 7 를 실행한다.
- 4) 고유 식별 코드인 WalletID를 생성하기 위한 Smart Contract로 사용자가 입력한 ID 정보 Table 7 과 Table 8 로 전달한다.
- 5) Table 8 에서 WalletID를 생성하기 위한 Smart Contract 실행하기 위한 초기화와 Prefix를 선정하고 Fig. 7 같이 WalletID를 생성한다.

Table 6. Example code of register to use on Front-end

```
app.post('/join', (req, res) => {
  const id = req.body.item.id;
  const name = req.body.item.name;
  let walletPrefix = "";
  const userType = Number(req.body.item.type);
  if(userType === 1) walletPrefix = 'indv_';
```

```
else if(userType === 2) walletPrefix = 'entp_';
else if(userType === 3) walletPrefix = 'verf_';
const walletId = walletPrefix + id
const password = req.body.item.password;
const sex = req.body.item.sex;
const callNums = req.body.item.callNums;
const birth = req.body.item.birth;
const hasJob = req.body.item.hasJob;
const profile = req.body.item.docsFile;
const employeeNums =
empty(req.body.item.employeeNums) === true ? " " :
req.body.item.employeeNums;
const jobGrade = empty(req.body.item.jobGrade) ===
true ? '0' : req.body.item.jobGrade;
const companyNo = req.body.item.companyNo;
const departmentName =
empty(req.body.item.departmentName) === true ? " " :
req.body.item.departmentName;
const openInfo = req.body.item.openInfo;
...
post(
'api/setWallet',
{ userName: name, wallet_id: walletId },
function(rst) {
  });
});
```

Table 7. Example code of Generating WalletID

```
app.post('/api/setWallet', function(req, res) {
  console.dir(req.body);
  let name = req.body.userName
  let walletId = req.body.wallet_id;
  let walletName = name + "'s Wallet";
  let args = [ walletName, walletId ];
  sdk.send(true, 'setWallet', args, res);
});
```

Table 8. Example code of smart contract in Generating WalletID

```
func(s *SmartContract) initWallet(APIStub
shim.ChaincodeStubInterface) pb.Response {

//Declare wallets
indv :=Wallet{ Name: "INDIVIDUAL", ID: "indv" }
entp :=Wallet{ Name: "ENTERPRISE", ID: "entp" }
verf :=Wallet{ Name: "ORGANIZATION", ID: "verf" }
...
}

func(s *SmartContract) setWallet(APIStub
shim.ChaincodeStubInterface, args []string) pb.Response {
  if len(args) != 2 {

returnshim.Error("Incorrect number of arguments. Expecting
2")
  }
  var wallet =Wallet{Name: args[0], ID: args[1]}

  WalletasJSONBytes, _ :=json.Marshal(wallet)
  err :=APIStub.PutState(wallet.ID,
WalletasJSONBytes)
```

```

        iferr != nil {
            returnshim.Error("Failed to
create asset " +wallet.Name)
        }
        returnshim.Success(nil)
    }
}
    
```

```

id "entp_entp@koreatech.ac.kr"
{
  "id": "entp_entp@koreatech.ac.kr",
  "key": "entp_entp@koreatech.ac.kr",
  "value": {
    "rev": "1-f16e030be89147a886abdc70c95decb"
  },
  "doc": {
    "_id": "entp_entp@koreatech.ac.kr",
    "_rev": "1-f16e030be89147a886abdc70c95decb",
    "id": "entp_entp@koreatech.ac.kr",
    "name": "한국기술교육대학교's Wallet",
    "version": "\u0000gMBUwA="
  }
}
    
```

Fig. 7. Result of Generating WalletID

등록 및 생성이 완료된 정보 중 Fig. 7 은 RDBMS에 저장되며 고유 식별 정보인 WalletID는 RDBMS와 블록체인

에 각각 저장되며, 차후 블록체인의 정보를 CRUD할 경우 WalletID를 기준으로 실행하게 된다.

#### 4. Use APIs with Smart Contracts

AG는 현재 ERP의 인적자원 관리 모듈에서 제공하지 않는 인적자원의 공유 및 탐색과 위·변조 방지를 위해 공통적으로 처리해야할 API를 설계하였고, 위·변조 방지를 위해 채용한 블록체인의 사용을 위해 Smart Contract를 API가 쌍(Pair)을 이루도록 하였다. API와 Smart Contract는 사용자에게 의해 데이터에 직접적인 영향을 미치므로 사용자의 구분인 개인(Indv), 기관(Entp), 검증(Verf)에 따라 쓰임새를 다르게 하였을 뿐 아니라 사용 등록, 데이터의 CRUD, 인적자원정보 관련 키워드 검색 등 3가지로 구분하였다. 자세한 목록은 Table 9 와 같다. 제안하는 APIs는 공유, 검색 및 위·변조 방지를 위한 사항을 우선 제시하였고, 차후 요구사항이 발생할 경우 유연하게 추가할 수 있다.

Table 9. List of APIs and Smart Contracts

URI of APIs	Name of Smart Contracts	Indv	Entp	Verf	Descriptions
/api/getWallet	getWallet	●	●	●	Use to Check the WalletID.
/api/setWallet	setWallet	●	●		Create WalletID when registering for first use.
/api/getCareer	getCareer	●	●	●	Use to search user's Career data.
/api/setCareer	setCareer	●	●		Save user's Career data.
/api/deleteCareer	deleteCareer	●	●		delete user's Career data.
/api/updateCareer	updateCareer	●	●	●	update user's Career data.
/api/getCareerHistory	getCareerHistory	●	●		Inquire data change history.
/api/getCareerWithNoneKeyword	getCareerWithNoneKeyword		●		Use to search for HR with Keyword
/api/getCareerWithKeyword"	getCareerWithKeyword"		●		Use to search for HR without Keyword

Table 10. Example code of API and Smart Contract

Example code of API	Example code of Smart Contracts
<pre> function _uploadCareerToNetwork(_item, res) {     let item = procStr(_item);     item.regDate = getCurrentDate();     item.updateDate = getCurrentDate();     item.acqDate1 = item.acqDate1.replace(/'/g, "");     item.acqDate2 = item.acqDate2.replace(/'/g, "");     // console.log('testtest')     // console.log(item.acqDate1);      if(Number(_item.careerType) === 3     &amp;&amp; Number(_item.state) === 3) {         _item.state = '3';     } else {         _item.state = '0';     }     ...     post(     'api/setCareer',     {         </pre>	<pre> func(s *SmartContract) setCareer(APIstub shim.ChaincodeStubInterface, args []string) pb.Response {     if len(args) != 20 {         returnshim.Error         ("Incorrect number of arguments. Expecting 4")     }      var careerKey =CareerKey{}     json.Unmarshal(generateKey(APIstub, "latestKey"), &amp;careerKey)     keyidx :=strconv.Itoa(careerKey.Idx)     fmt.Println("Key : " +careerKey.Key + ", Idx : " +keyidx)      var career =Career{         OwnerNo: args[0],         OwnerName: args[1],         CareerNo: args[2],         OwnerWallet: args[3],         AcqDate1: args[4],         AcqDate2: args[5],         RegDate: args[6],     } }         </pre>

```

ownerNo: String(item.ownerNo),
ownerName: String(item.ownerName),
careerNo: String(item.careerNo),
ownerWallet: String(item.ownerWallet),
acqDate1: String(item.acqDate1),
acqDate2: String(item.acqDate2),
regDate: String(item.regDate),
updateDate: String(item.updateDate),
description: String(item.description),
careerType: String(item.careerType),
careerName: String(item.careerName),
acqOrgName: String(item.acqOrgName),
docsUrl: String(item.docsUrl),
careerClass: String(item.careerClass),
trainType: String(item.trainType),
onoffType: String(item.onoffType),
acqOrgNo: String(item.acqOrgNo),
trainNo: String(item.trainNo),
state: _item.state,
reason: String(item.reason)
},
function(rst) { res.send(rst); });
}

Description: args[7],
CareerType: args[8],
CareerName: args[9],
AcqOrgName: args[10],
DocsUrl: args[11],
CareerClass: args[12],
UpdateDate: args[13],
TrainType: args[14],
OnoffType: args[15],
AcqOrgNo: args[16],
TrainNo: args[17],
State: args[18],
Reason: args[19] }

careerAsJSONBytes, _ :=json.Marshal(career)

var keyString =careerKey.Key +keyidx
fmt.Println("careerKey is " +keyString)

err :=APIStub.PutState(keyString, careerAsJSONBytes)
iferr != nil {
    returnshim.Error
    (fmt.Sprintf("Failed to record career catch: %s", careerKey))
}
...
}
    
```

5. Run Testing APIs and Smart Contract

Table 11. Result of running '/api/updateCareer'

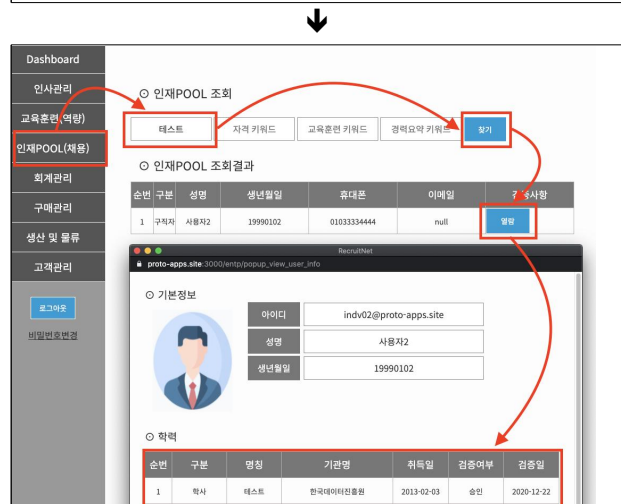
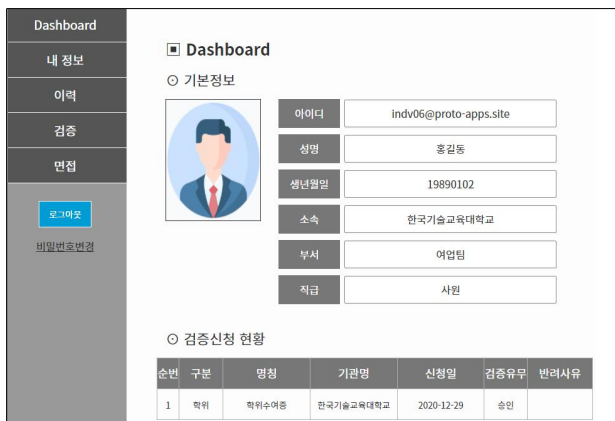
```

{"message": "-----\n\n", "level": "info"}
{"message": "===> [Begin] POST//Addr: api/updateCareer", "level": "info"}
{"message": "===> Request Data", "level": "info"}
{"message": {"careerKey": "TTX_CAREER_100033", "ownerWallet": "indv06@proto-apps.site"}
{"message": "===> [End] POST//Addr: api/updateCareer", "level": "info"}
{"message": "===> Request Data", "level": "info"}
{"message": {"careerKey": "TTX_CAREER_100033", "ownerNo": "48", "ownerName": "홍길동", "careerNo": "174", "ownerWallet": "indv_indv06@proto-apps.site", "acqDate1": "2020-12-01", "level": "info"}
    
```

Table 12. Result of running '/api/getCareerWithKeyword'

```

{"message": "-----\n\n", "level": "info"}
{"message": "===> [Begin] SQL//Addr: /getCareerWithKeyword", "level": "info"}
{"message": "===> Request Query", "level": "info"}
...
{"message": "***** Response Data", "level": "info"}
{"message": {"userNo": "41", "id": "indv02@proto-apps.site", "walletId": "indv_indv02@proto-apps.site", "level": "info"}
{"message": "-----\n\n", "level": "info"}
    
```



설계, 구현된 API Gateway의 동작 시험을 위해 Web UI 구성 후 API 호출을 이용한 테스트를 실시하였다.

Web UI는 ERP의 모듈을 차용하여 구성하였다. 시험 결과를 위하여 첫 번째로 데이터를 조회, 삽입, 갱신을 포함하는 API 인 '/api/updateCareer'의 호출 하였고, 결

과는 Table 11 과 같다. API 호출하면 사용 등록 시 생성된 'WalletID'를 조회하여 해당 데이터를 찾아 사전에 정의된 프로세스를 수행한 것을 알 수 있다.

두 번째로 네트워크 형성과 공유 및 검색의 결과를 확인하기 위해 '/api/getCareerWithKeyword'의 호출하였다.



해당 API는 Keyword 정보와 함께 블록체인 네트워크를 검색 후 Table 12 와 같이 결과를 반환한다. 검색을 위해 블록체인 네트워크에 접근한 것은 블록체인에 저장된 'WalletID'를 반환한 것으로 같음할 수 있다.

#### IV. Conclusions

인사관리에서 벗어나 인적자원이라는 용어가 널리 사용되더라도 불구하고 ERP의 HR Management은 기능적 인사관리로서 머물러 있고, 자원으로써 확보·개발·활용을 위한 진보가 부족한 것으로 보인다.

본 연구에서는 ERP 혹은 인적자원관리(HR Management) 모듈에서 사용 가능한 기능적, 시스템적 확장 방법으로 API Gateway를 제안 하였다. API Gateway는 배포(Deployment), 확장(Scaling), 장애(Failure) 등의 관점에서 기술의 적용이 매우 유용하다. 제안한 API Gateway는 국내의 ERP와 손쉽게 연동해 공통로직, 인증, 인가 등의 인적자원 관리 모듈의 기능과 가치를 확장할 수 있는 가능성(공통로직, 인증, 서비스 중립적 가치 등)을 보유하고 있고, 블록체인 네트워크를 이용해 그 동안 어려웠던 인적자원을 공유하고 키워드들 이용한 검색 뿐 아니라 불가한 풀(Pool)을 구성하여, 인적자원을 적극적으로 활용할 수 있는 도구로 사용할 수 있다. 뿐만 아니라 인공지능, 머신러닝 기술 등이 급속한 발전을 이루면서 blockchains을 이용한 풀(Pool)은 빅 데이터 기술을 활용하기 위한 완벽한 자원이 될 것이다.

하지만 3가지의 문제가 지적될 수 있다. 첫 번째, 간단하지만 독립된 시스템을 구성하는 것과 크게 다르지 않는 문제와 시스템 관리에 대한 문제가 지적될 수 있다. 두 번째, API Gateway의 스케일 아웃(scale-out) 적용이 유연하게 되지 않을 경우, API Gateway가 병목지점이 되어 어플리케이션의 성능저하가 일어날 수 있고 추가적인 계층이 만들어지는 것이기 때문에, 그만큼 네트워크 레이턴시(latency)가 증가하는 문제가 있다. 세 번째, 본 연구에서 사용한 블록체인의 TPS(Transaction Rate Per Second)가 아직은 충분하지 않다는 것이다. 블록체인의 TPS 향상을 목적으로 한 연구[11]가 진행된 바 있고, 성능 분석을 한 연구[12]가 존재 하지만 기술의 적용까지는 성숙하지 못한 것으로 보인다.

향후 연구로서 서버리스(Serverless) 형태인 FaaS(Function as a Service)로 API Gateway Function 요청 수에 따라 자동 스케일링이 가능하고 비즈니스 로직을

작성하면 플랫폼이 관리를 전담하는 컨테이너 구현 방안과 설계에 관한 연구를 진행하고자 한다.

#### REFERENCES

- [1] Lee, Jong-Joon, Jang, Kyung-Ho. "A Study on the Success Factors of e-Human Resource Management(e-HRM)." YEUNGSANG JOURNAL 3.1(2010):17-31.
- [2] Se-hwan Lee, Bong-hyun Kim, Dong-uk Cho, Sun-ae Park, Seung-youn Kim., "Design on Distributed Transaction Service Using ERP Personal Management System.", Proceedings of Symposium of the Korean Institute of communications and Information Sciences., 2006.11., pp.387-390
- [3] Elbahri, Faisal Mohamed et al. "Difference Comparison of SAP, Oracle, and Microsoft Solutions Based on Cloud ERP Systems: A Review." Proceedings - 2019 IEEE 15th International Colloquium on Signal Processing and Its Applications, CSPA 2019. Institute of Electrical and Electronics Engineers Inc., 2019. 65-70. Proceedings - 2019 IEEE 15th International Colloquium on Signal Processing and its Applications, CSPA 2019. Web.
- [4] ERPNext, "Human Resource", <https://docs.erpnext.com/docs/user/manual/en/human-resources>
- [5] Yang, Huanchun. "Business Process Reengineering of E-HRM Based on ERP." 2009 International Conference on Management of E-Commerce and e-Government, ICMCG 2009. N.p., 2009. 246-249. 2009 International Conference on Management of e-Commerce and e-Government, ICMCG 2009. Web.
- [6] Zhanghuiqin, and Tangjing. "HRIS Application Analysis Based on ERP Platform." ICIME 2010 - 2010 2nd IEEE International Conference on Information Management and Engineering. Vol. 4. N.p., 2010. 170-172. ICIME 2010 - 2010 2nd IEEE International Conference on Information Management and Engineering. Web.
- [7] Salah, Dina, Maha Hafez Ahmed, and Kamal Eldahshan. "Blockchain Applications in Human Resources Management: Opportunities and Challenges." ACM International Conference Proceeding Series. Association for Computing Machinery, 2020. 383-389. ACM International Conference Proceeding Series. Web.
- [8] Kim, Tai Hoon et al. "A Privacy Preserving Distributed Ledger Framework for Global Human Resource Record Management: The Blockchain Aspect." IEEE Access 8 (2020): 96455-96467. IEEE Access. Web.
- [9] Seokyong Choi, Jisu Choi, Dongju Jung, Byungjeong Lee. "Applying Micro Service Architecture to Improve DApp release : Chatbot Case." Korea Software Congress (2020): 1407-1409.
- [10] Hyperledger Fabric Foundation, "Wallet", <https://hyperledger-fabric.readthedocs.io/en/release-1.4/developapps/wallet.html#scenario>

- [11] Gorenflo, Christian et al. "FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second." ICBC 2019 - IEEE International Conference on Blockchain and Cryptocurrency. Institute of Electrical and Electronics Engineers Inc., 2019. 455?463. ICBC 2019 - IEEE International Conference on Blockchain and Cryptocurrency. Web
- [12] Dreyer, Julian, Marten Fischer, and Ralf Tonjes. "Performance Analysis of Hyperledger Fabric 2.0 Blockchain Platform." CCIoT 2020 - Proceedings of the 2020 Cloud Continuum Services for Smart IoT Systems, Part of SenSys 2020. Association for Computing Machinery, Inc, 2020. 32?38. CCIoT 2020 - Proceedings of the 2020 Cloud Continuum Services for Smart IoT Systems, Part of SenSys 2020. Web.

## Authors



Ji-Woon Lee received the M.S. degrees in Information Processing from Sogang University, Korea, in 2015. He is currently a graduate student of the Department of Computer Engineering and work in at Korea

University of Technology and Education, Chungcheongnam-do, Korea. He is interested in blockchain, Hyperledger Project, Information Security and cloud computing.



Hee-Suk Seo received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Sungkyunkwan University, Korea, in 2000, 2002 and 2005, respectively. Dr. Seo joined the faculty of the Department

of Computer Engineering at Korea University of Technology and Education, Chungcheongnam-do, Korea, in 2006. He is currently a Professor in the Department of Computer Engineering, Korea University of Technology and Education. He is interested in Information Security, Network Security, malicious code analysis, modeling & simulation and intelligent system.