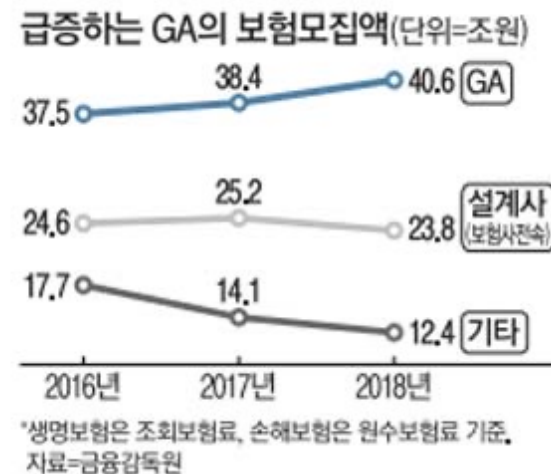


블록체인을 활용한 새로운 보험대리점 수수료 지급 방식 구현

한성대학교 20213201 박재훈

서론

- 과거에는 보험사 전속 설계사만이 소속 보험사의 보험을 판매
- 2005년 보험대리점 (General Agency, GA) 등장 이후 다양한 보험사의 보험 판매
- 보험대리점의 규모는 나날이 커지고 있음



서론

- 보험대리점의 규모가 커졌으나 보험사에서의 수수료 내역 지급 방식은 모두 상이

이름	사원번호	수수료	판매내역
...

보험사A

사원번호	일자별	종류	수수료
...

보험사B

사원번호	생년월일	이름	수수료
...

보험사C

서론

- 대부분 메일을 통해서 보내는 방식 (보안성 떨어짐)
- 본 논문에서는 이에 대한 새로운 블록체인 시스템 구현

블록체인

- 사토시 나카모토가 비트코인 개발 당시 개발한 것
- 데이터를 블록 형태로 묶어서 관리
- P2P 네트워크 상의 데이터 위/변조를 합의 알고리즘을 통해 방지

하이퍼레저 패브릭

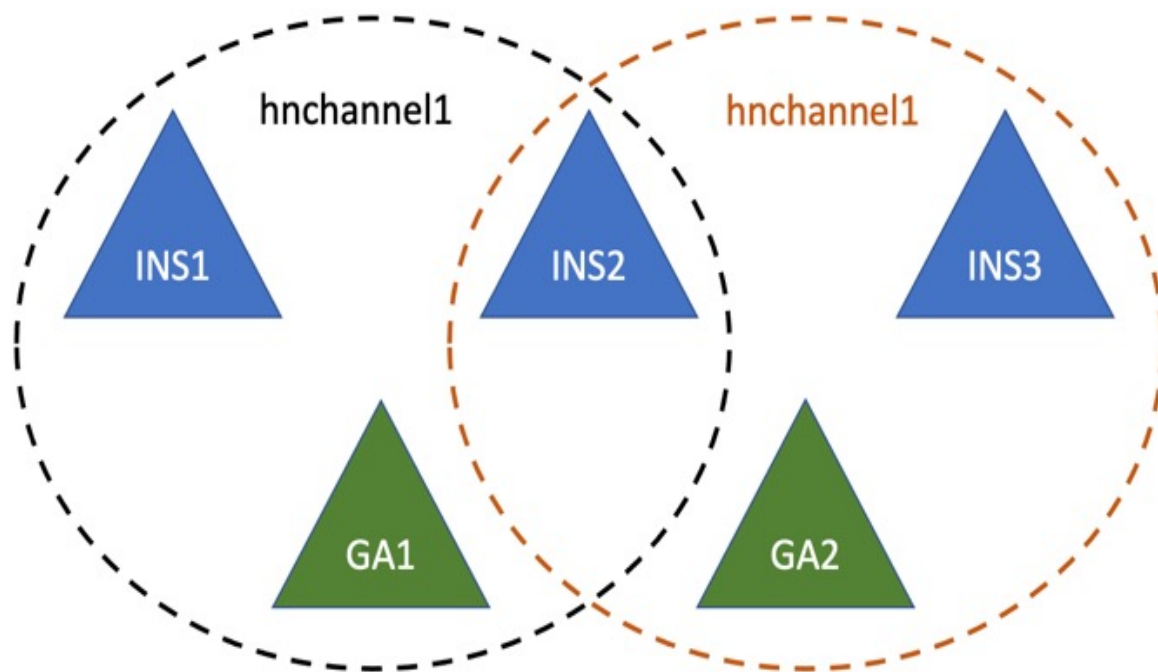
- 리눅스에서 주도 중인 하이퍼레저 프로젝트에서 나온 프레임워크로, IBM이 개발
- ‘체인코드’를 통해 스마트 컨트랙트 호스팅
- 본 논문에서는 2.2.2 LTS 버전 이용

도커

- 실행환경이나 프로그램을 ‘이미지’로 만든 뒤, ‘컨테이너’라는 단위로 추상화
- 하이퍼레저 패브릭에서는 여러 구성 요소를 컨테이너로 관리
- 피어, 데이터베이스, CA 등이 도커를 통해 구동

보험대리점 수수료 지급 시스템

- 보험대리점 하나가 여러 보험사와 연계할 수 있음
- 보험대리점 하나당 채널 1개 생성
- 5개 조직
 - Ins1, Ins2, Ins3 (보험사)
 - Ga1, Ga2 (보험대리점)
- 2개 채널
 - hnchannel1 (Ins1, Ins2, Ga1)
 - hnchannel2 (Ins2, Ins3, Ga2)



보험대리점 수수료 지급 시스템

- TypeScript
- CouchDB

Organization	Peer port	CA port	DB port
ins1	7051	7054	5984
ins2	8051	8054	6984
ins3	9051	9054	7984
ga1	10051	10054	8984
ga2	11051	11054	9984
orderer	12051	12054	-

보험대리점 수수료 지급 시스템

- class FeeData

Field	Type	Description
key	string	key value
date	string	date when executed
num	string	same as key value
companyCode	string	unique code of a company
product	string	name of a product
fee	number	fee value

보험대리점 수수료 지급 시스템

- getAll
- get
- create
- edit

보험대리점 수수료 지급 시스템

- getAll

```
async getAll(context: Context) {
  const startKey = '';
  const endKey = '';
  const results: FeeData[] = [];

  for await (const {key, value} of context.stub.getStateByRange(startKey, endKey)) {
    const strValue = Buffer.from(value).toString('utf8');
    let record: FeeData;
    try {
      record = JSON.parse(strValue);
      record.key = key;
    } catch(err) {
      console.log(err);
      record = new FeeData();
      record.key = key;
    }
    results.push(record);
  }
  return results;
}
```

보험대리점 수수료 지급 시스템

- get

```
async get(context: Context, key: string) {  
  const byteItem = await context.stub.getState(key);  
  if ( !byteItem || byteItem.length === 0 ) {  
    throw new Error(`error on loading data '${key}'`);  
  }  
  try {  
    const fee = <FeeData> JSON.parse( byteItem.toString() );  
    return fee;  
  } catch(err) {  
    console.log(err);  
    return undefined;  
  }  
}
```

보험대리점 수수료 지급 시스템

- create

```
async create(context: Context, companyCode: string, product: string, fee: number) {
  try {
    const f: FeeData = {
      date: dateFormat(new Date(context.stub.getTxTimestamp().nanos)),
      num: this.key(context),
      companyCode: companyCode,
      product: product,
      fee: fee
    };
    f.key = f.num;
    await context.stub.putState(f.key, stateValue(f));
    console.log(`new fee data generated`);

  } catch(err) {
    console.log(err);
    return undefined;
  }
}
```

보험대리점 수수료 지급 시스템

- edit

```
async edit(context: Context, key: string, field: string, value: string) {
  const byteItem = await context.stub.getState(key);
  if ( !byteItem || byteItem.length === 0 ) {
    throw new Error(`error on loading data '${key}'`);
  }
  try {
    const f = <FeeData> JSON.parse( byteItem.toString() );
    const keys = f.key.split('-');
    let sequence = 1
    if ( keys[keys.length - 1].includes('E') ) {
      sequence = parseInt( keys[keys.length - 1].substring(1) ) + 1;
      keys.length -= 1;
      keys.push('E' + sequence);
    }
    const newKey = keys.join('-');
    f.key = newKey;
    f.num = newKey;
    f[field] = value;
    await context.stub.putState(f.key, stateValue(f));
    console.log(`fee data edited`);

  } catch(err) {
    console.log(err);
    return undefined;
  }
}
```

참고문헌

- [1] “공룡 된 'GA'...보험시장 절반 삼켰다”, 매일경제 [Internet] available: <https://bit.ly/3d9HLVF>
- [2] Nakamoto, Satoshi. Bitcoin: A peer-to-peer electronic cash system.
- [3] Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine generals problem."
- [4] Beckert, Bernhard, et al. "Formal specification and verification of Hyperledger fabric chaincode." 3rd Symposium on Distributed Ledger Technology (SDLT-2018) co-located with ICFEM. 2018.

Q & A

