

블록체인을 활용한 카페 이용시간 관리 시스템

한성대학교 IT융합공학부 20213201 박재훈

서론

- COVID-19 바이러스가 누그러짐에 따라 카페 매장 내 취식이 가능해짐

- 1시간 이용 제한 권고 (보건복지부)

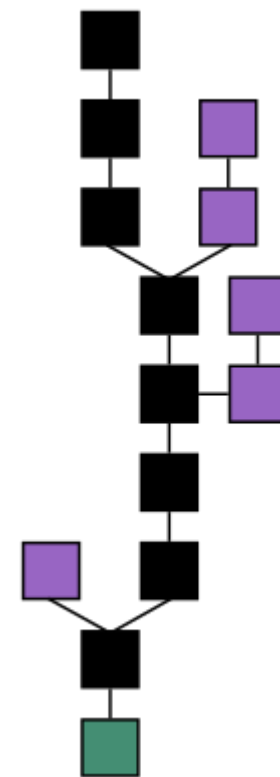
식당·카페(무인카페 포함)	<ul style="list-style-type: none">▶ 2인 이상이 커피·음료류, 디저트류만을 주문했을 경우 매장 내 머무르는 시간을 1시간으로 제한(강력 권고)▶ 식당·카페 모두 21시~익일 05시까지 포장·배달만 허용▶ 테이블 또는 좌석 한 칸을 띄워 매장 좌석의 50%만 활용하되, 이를 준수하기가 어려울 경우 ①[고정폭 빈 칸]테이블 간 1m 거리 간 칸막이/가림막 설치 중 한 가지 준수(시설 면적 50㎡ 이상)▶ 뷔페의 경우 공용접게접시수저 등 사용 전후 손소독제 또는 비닐장갑 사용 음식 담기 위한 대가사 이용자 간 간격 유지
----------------	---

- 권고사항이 잘 지켜지지 않을 뿐더러, 제한할 마땅한 수단도 없음

- 본 논문에서 블록체인을 활용한 카페 이용시간 관리 시스템 구현

관련 연구 - 블록체인

- 중앙 서버에서의 관리가 아닌 Peer-to-Peer 방식의 네트워크
- 블록 형태로 피어를 묶어 데이터를 관리
- 참여자 전원이 동일한 장부를 지님
- 모두의 장부를 지속 확인함으로써 무결성 유지



관련 연구 - 하이퍼레저 패브릭

- 리눅스 재단에서 주도하는 ‘하이퍼레저 프로젝트’에 IBM이 참여하여 개발한 프레임워크
- CFT 기반의 합의 프로토콜 이용
- 기업 비즈니스 구현에 적합
- 체인코드(chaincode)를 이용하여 스마트 컨트랙트 구현

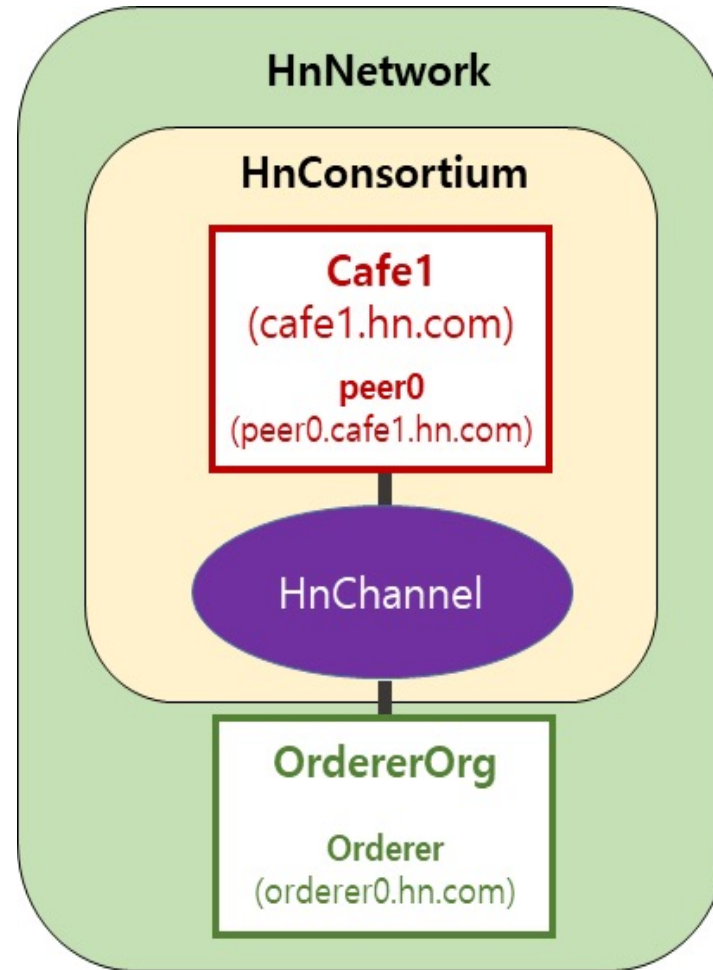


카페 이용 제한 시스템

- Hyperledger Fabric 2.3.0
- TypeScript
- CouchDB
- Docker

카페 이용 제한 시스템

- 네트워크 구성



카페 이용 제한 시스템

- DB 정의 (IO 클래스)

필드	형식	설명
key	string	키값
cafe	string	카페명
phone	string	핸드폰 번호
in_ms	number	입장시간
out_ms	number	퇴장시간
agreeToOfferInfo	boolean	정보제공동의여부

카페 이용 제한 시스템

- gettingIn 메소드
- 입장 시 IO 객체 생성

```
async gettingIn(context: Context, cafe: string, phone: string, agreeToOfferInfo: boolean) {  
  if ( !agreeToOfferInfo ) {  
    console.log('not available without agreeing to offer the information');  
    return;  
  }  
  try {  
    const io: IO = {  
      key: this.generateKey(),  
      cafe: cafe,  
      phone: phone,  
      in_ms: new Date().getTime(),  
      out_ms: -1,  
      agreeToOfferInfo: agreeToOfferInfo  
    };  
    await context.stub.putState(io.key, stateValue(io));  
    console.log('new IO generated');  
  } catch(err) {  
    console.log(err);  
  }  
}
```


카페 이용 제한 시스템

- gettingOut
- 퇴장 시 IO 객체 수정

```
async gettingOut(context: Context, phone: string) {
  try {
    const res = context.stub.getQueryResult(JSON.stringify({
      selector: {
        phone: phone
      },
      limit: 1
    }));
    if ( !res ) throw 'could not find the given phone number';
    for await (const {key, value} of res) {
      const io = toItem(value);
      io.out_ms = new Date().getTime();

      if ( io.out_ms - io.in_ms > 60 * 60 * 1000 ) {
        console.log(`the customer '${io.phone}' stayed in the cafe '${io.cafe}' over 1 hour`);
      }
      await context.stub.putState(io.key, stateValue(io));
      return;
    }
  } catch(err) {
    console.log(err);
  }
}
```

카페 이용 제한 시스템

- checkDisobeyed
- 수칙 어긴 사람을 확인

```
async checkDisobeyed(context: Context) {
  const ts = new Date().getTime();
  const res = context.stub.getQueryResult(JSON.stringify({
    selector: {
      $and: [
        {
          out_ms: { $lt: 0 },
        }, {
          in_ms: { $lt: ts - (60 * 60 * 1000) }
        }
      ]
    }
  }));
  const disobeyeds: IO[] = [];
  for await (const {key, value} of res) {
    disobeyeds.push(toItem(value));
  }
  console.log(`${disobeyeds.length} customer(s) existing`);
  return disobeyeds;
}
```

결론 및 향후 연구

- 본 논문에서의 시스템을 통해 COVID-19에 대한 좀 더 강경한 대응 기대
- 향후 백엔드 및 프론트엔드 구현을 통해 애플리케이션으로서의 활용 기대

Q & A

