네트워크 설계서

1.네트워크 구조

- 생태계 구조
- -BApp 구성 및 접근 방식
- 2. 네트워크 환경설정

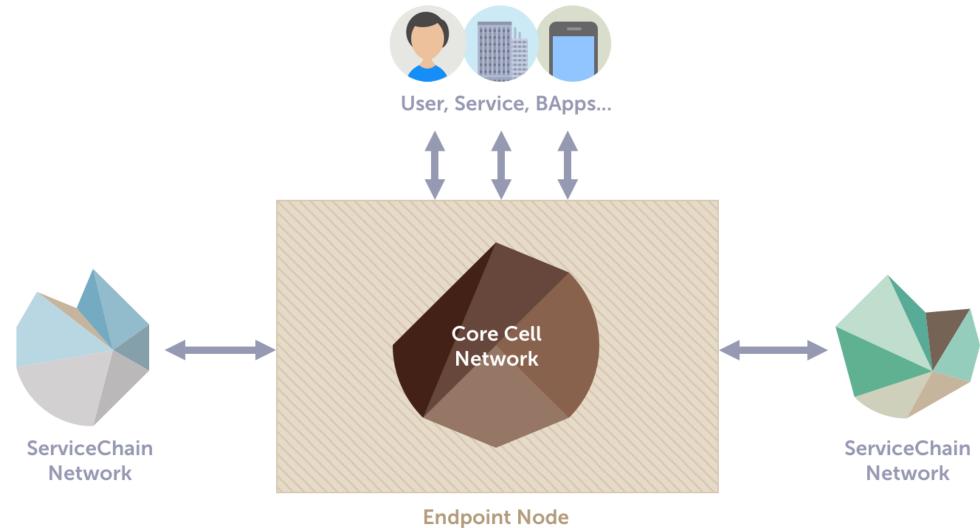
- 3. 네트워크 관계도
 - 네트워크 설계도
 - 트러플 설계도
- 4. 키, 인증서 관리
 - 키와 월렛 관리
 - 관리자 계정 관리

1.네트워크 구조

-생태계 구조

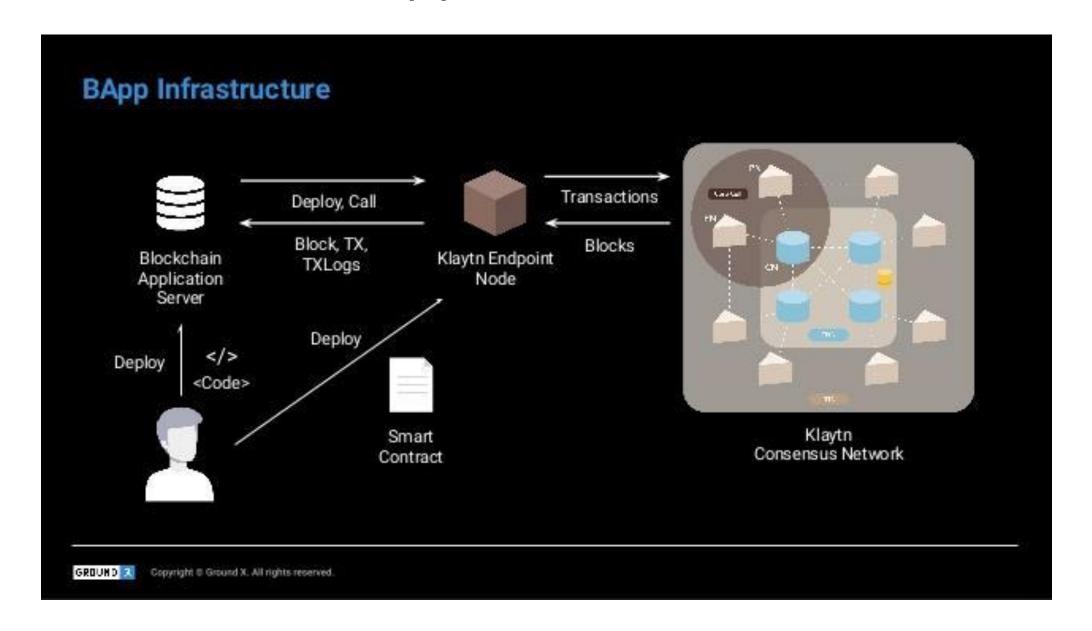
-BApp 구성 및 접근 방식

네트워크 구조 – 생태계 구조



Endpoint Node Network

네트워크 구조 - BApp 구성 및 접근 방식



2.네트워크 환경설정

- 환경설정
 프론트 엔드 react, javascript
 백엔드 Nodejs, KAS(Klaytn Api Service), Kaikas api 네트워크 구성 – truffle, caver-js
- 개발 환경 Ubuntu 18.04, windows 10, truffle v5.1.
- 개발툴 : Visual Studio Code, Git
- 인증서 & ID 관련 Kaikas api 이용
- 스테이킹 매니저 계정 관리 멀티시그 이용
- 메인넷 클라이언트 설정 KAS를 이용한 EN접근

3.네트워크 설계서

-네트워크 설계서

-트러플 네트워크 설계서

Truffle 네트워크 설계도 dapp Caver.js React Javascript (ui, ux) (action) Smart contract KAS Transaction render block block block Kaikas block

App folder (React, JavaScript) Webpack-config.js Node_modules Caver.js Truffle Truffle-config.js (RPC server)

트러플 네트워크 설계도

Contract Folder

Sol file

4.키, 인증서 관리

- 키와 월렛 관리

- 관리자 계정 관리

인증서(개인 키, 지갑) 관리





Kaikas 사용

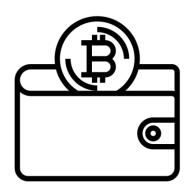
Klaytn JSON-RPC API



Cypress Port : 8217

← →

Baobab Port: 1001



```
if (typeof window.klaytn !== 'undefined') {
    // Kaikas user detected. You can now use the provider.
    const provider = window['klaytn']
}
```

```
try {
   const accounts = await klaytn.enable()
   // You now have an array of accounts!
   // Currently only one:
   // ['0xFDEa65C8e26263F6d9A1B5de9555D2931A33b825']
} catch (error) {
   // Handle error. Likely the user rejected the login console.error(error)
}
```

Manager 계정 관리 -Multi sig 설계도

제안자 Manager A



기능 실행을 위한 submit transaction



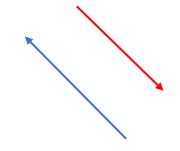
검수자 승인 시

KLAY 전송

제안 확인 요청



검수자 Manager B



제안 승인



검수자 Manager C