Solidity
Smart Contract 101

Robbie OH

Speaker:
Robbie OH
(오영택)

- HAECHI Labs SW engineer
- Decipher co-founder
- 이드콘 준비위원회

robbieinertia@gmail.com
010-3264-3011

What is Dapp

- 스마트컨트랙트와 상호작용 하는 어플리케이션

Why Dapp

- 실행 결과가 번복되지 않음
- 누구나 확인할 수 있는 코드를 기반으로 작동함
- 암호화폐라는 가치 전송수단을 서비스에 intregration 가능

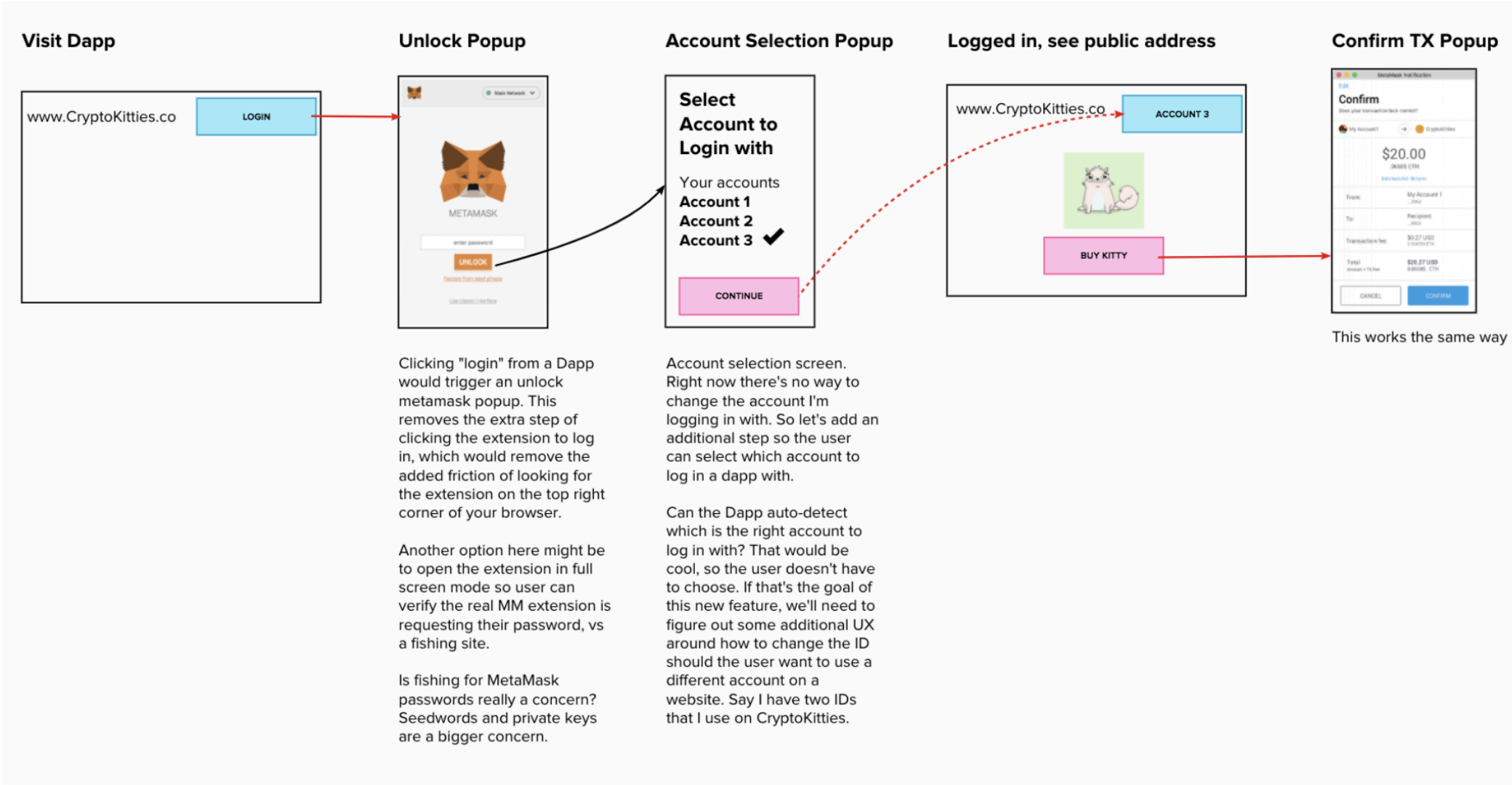
Why Not Dapp

- 상호작용이 잦은 앱들
  - 속도 문제보다는 스마트컨트랙트 특성상 당사자가 매번 직접 실행 해야 해서 UX/UI 경험이 안좋다
  - 매 행위 마다 수수료 발생 -> fee delegation등의 제안됨
- 로직이 너무 복잡한 앱들
  - 허용된 계산의 최대치가 있다(gas)
  - 여러 offchain solution들이 현재 연구중

(아직)
Dapp이 어울리지 않는 경우

- 상호작용이 잦은 앱들
  - **속도 문제보다는 스마트컨트랙트 특성상 당사자가 매번 직접 실행 해야 해서 UX/UI 경험이 안좋다**
  - **매 행위 마다 수수료 발생 -> fee delegation등의 제안됨**
- 로직이 너무 복잡한 앱들
  - 허용된 계산의 최대치가 있다(gas)
  - 여러 offchain solution들이 현재 연구중

(아직)
Dapp이 어울리지 않는 경우

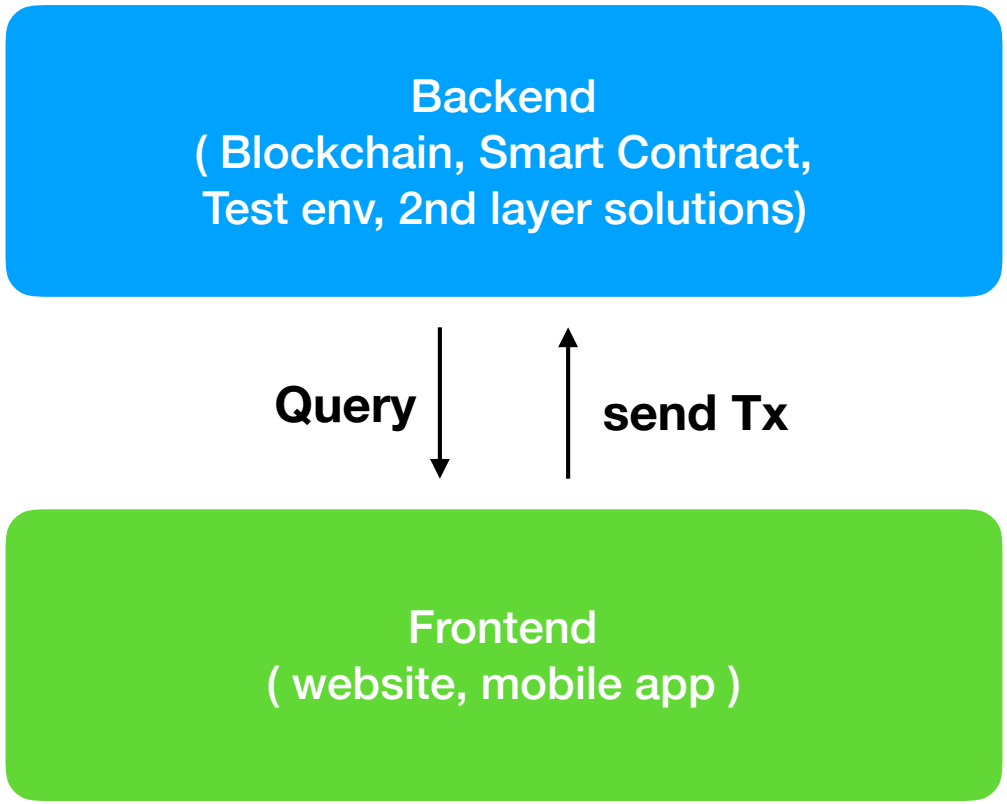


https://github.com/metamask/metamask-extension/issues/3383

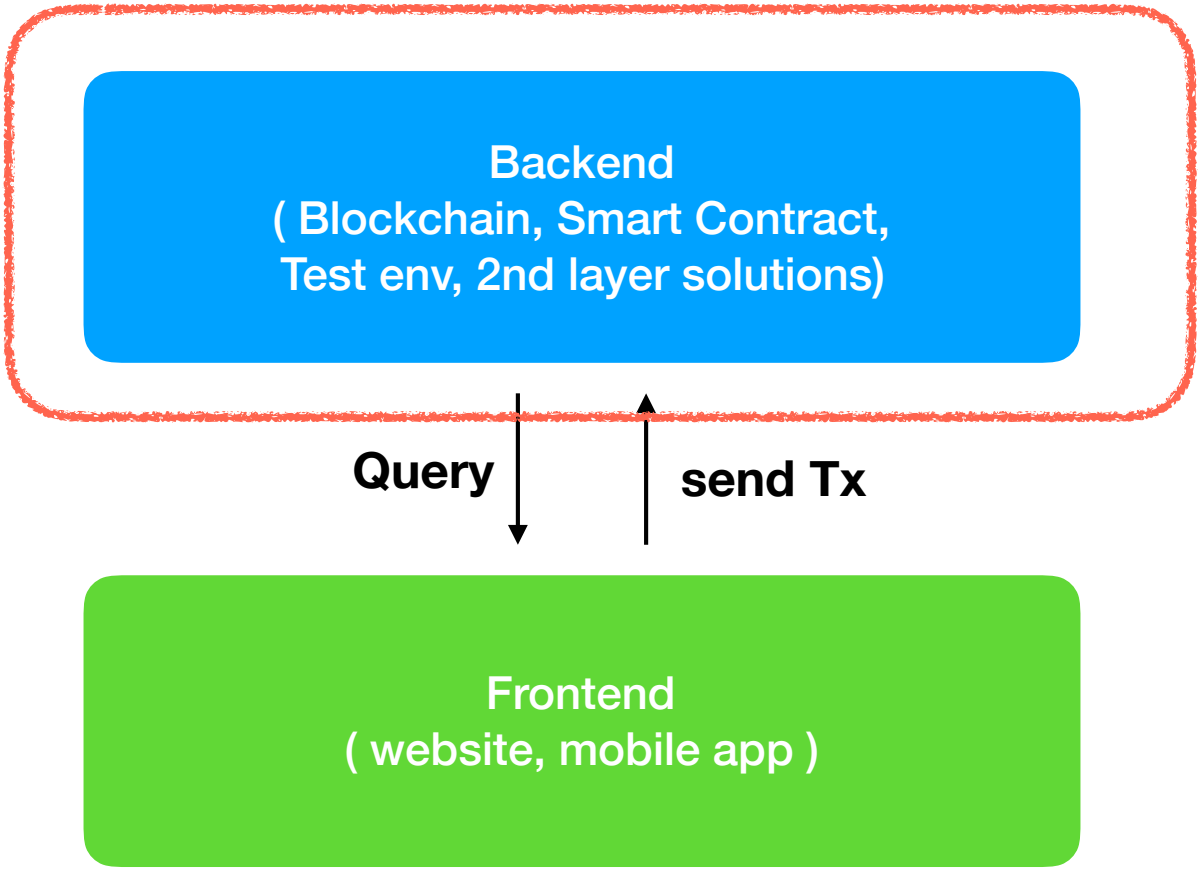
(아직)
Dapp이 어울리지 않는 경우

- 상호작용이 잦은 앱들
  - 속도 문제보다는 스마트컨트랙트 특성상 당사자가 매번 직접 실행해야 해서 UX/UI 경험이 안좋다
  - 매 행위 마다 수수료 발생 -> fee delegation등의 제안됨
- 로직이 너무 복잡한 앱들
  - **허용된 계산의 최대치가 있다(gas)**
  - **여러 offchain solution들이 현재 연구중**

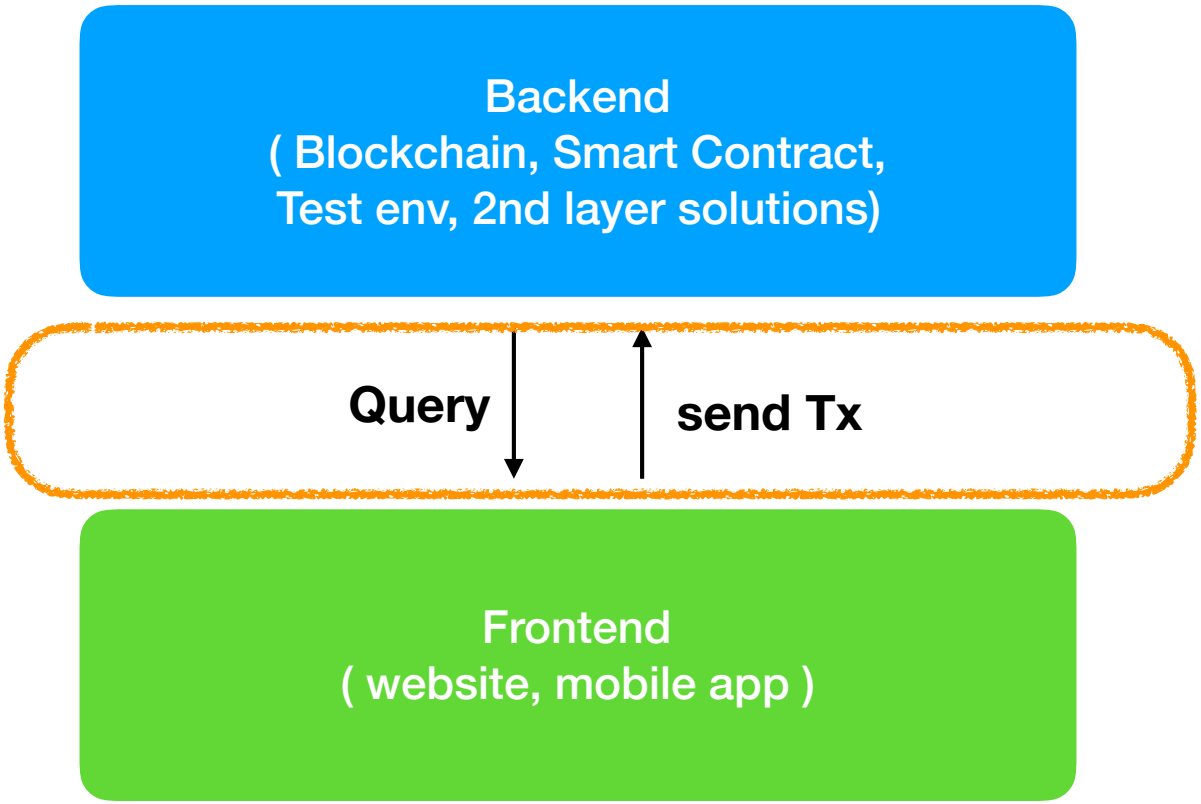
Ethereum Dapp 개발환경 이해하기



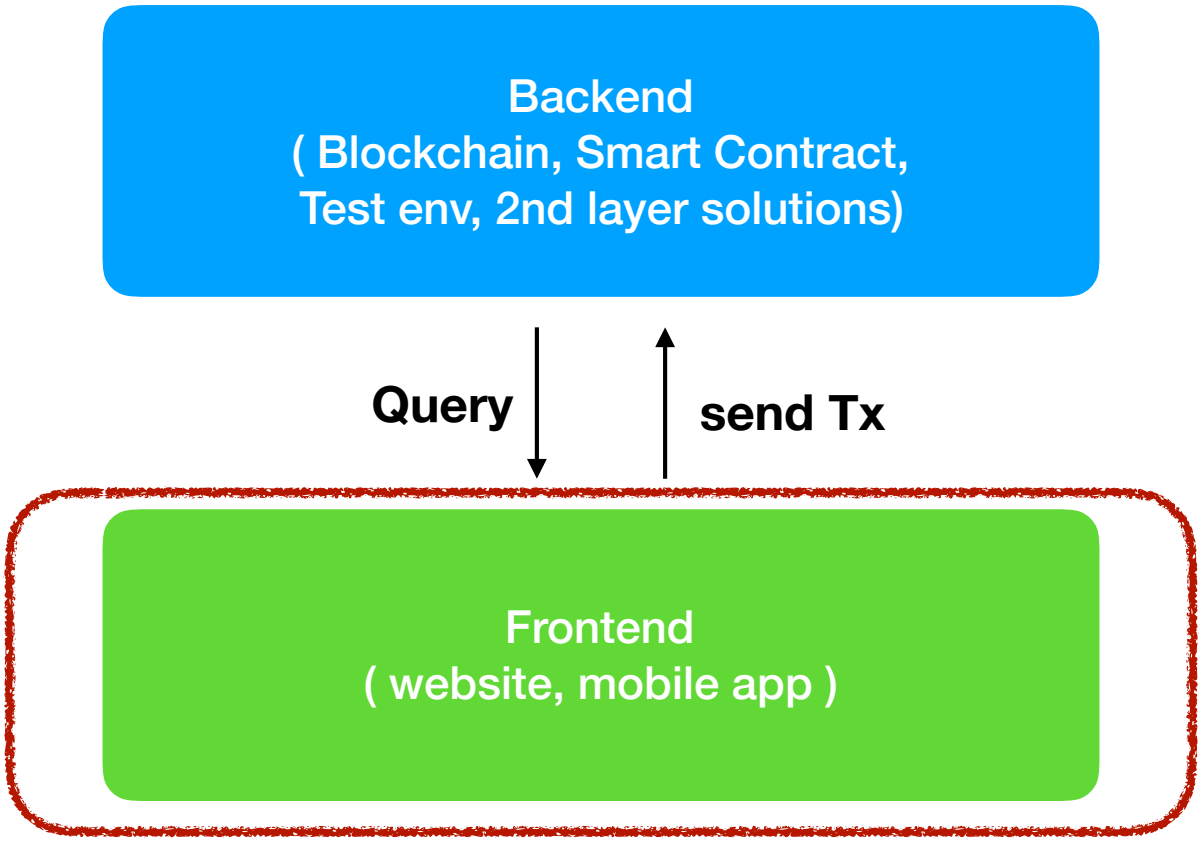
Ethereum Dapp 개발환경 이해하기



Ethereum Dapp 개발환경 이해하기



Ethereum Dapp 개발환경 이해하기



What is solidity

- 이더리움 Dapp을 만들기 위한 programming language
  - what is PL?

What is Programming Language?

- 컴퓨터: 전기 신호(on, off/ 0, 1)의 조합으로 일반적인 연산을 하는 기계
- 기계어: 전기 신호와 1:1 매칭 되는 언어
- 프로그래밍 언어: (좁은 의미) 사람이 이해할 수 있는 수준의 문법을 컴퓨터가 이해할 수 있는 기계어로 쉽게 변환이 가능하도록 만든 체계. (문법, 번역기, 실행기 등등)
    - 프로그래밍 언어의 목표: 쉬운 언어 -> 기계어

What is Programming Language?

- 이더리움: 가상의 컴퓨터(virtual machine) 모델 이용
- 가상의 컴퓨터란?
  - CPU등 하드웨어와 무관한 기계어로 동작하는 가상의 기계
  - “표준화” 호환성을 확보하기에 좋음
- 이더리움도 EVM이라는 가상의 컴퓨터에서 실행 가능한 기계어가 있음.

What is Programming Language?

- 이더리움에서 언어: solidity, vyper, …
    - **목표**: EVM bytecode 로 번역 가능하면서 사람이 쉽게 읽을 수 있도록 편한 개발환경을 제공!

Solidity Features

- javascript style 의 문법
- 객체 지향 정적 타입 언어
- 쉽게 배울 수 있음!

ERC20 explained

- 토큰 거래의 표준화를 위해 일정한 인터페이스를 제안!
  - (구현 자체는 표준이 아님)
- 주요 파라메터:
  - totalSupply
  - name
  - symbol
  - decimals
- mint, burn등은 표준은 아님!
- 유명한 구현: open zeppelin, consensys

ERC20 explained

- 자주 하는 실수:
  - 토큰 발행량 단위 실수(decimal을 놓침)
  - 규격 이외에 추가적인 custom logic(lock, mint 등)을 넣을 때

Tutorial 1

- CryptoZombies (문법)

Tutorial 2

- Remix (배포)

Tutorial 3

- truffle, drizzle을 이용한 간단한 dapp 만들어보기