

Developing an Ethereum blockchain application

**Created by:
Nikolaos-Petros Triantafyllidis**

Things to get out of the way

Things to get out of the way

**Who the frack
am I?**

Things to get out of the way

**Who the frack
am I?**

**What the frack is
Ethereum?**

Things to get out of the way

**Who the frack
am I?**

**What the frack is
Ethereum?**

**I cannot explain
everything...**

Timeline



1. Research Question

Can Ethereum be directly used to rapidly deploy meaningful and sufficiently performing trusted applications with added value over traditional approaches?

2. Installing Ethereum

OS X: OK!

Ubuntu: OK!

Debian: Are you kidding me?

Timeline

0. Motivation

- Hype: 'The Internet how it was supposed to be'.
- More hype: \$18.5M funding
- Contracts: The 'safeguards' of liberty.

1. Research Question

Can Ethereum be directly used to rapidly deploy meaningful and sufficiently performing trusted applications with added value over traditional approaches?

2. Installing Ethereum

OS X: OK!

Ubuntu: OK!

Debian: Are you kidding me?

Timeline



1. Research Question

Can Ethereum be directly used to rapidly deploy meaningful and sufficiently performing trusted applications with added value over traditional approaches?

2. Installing Ethereum

OS X: OK!

Ubuntu: OK!

Debian: Are you kidding me?

Timeline



1. **Research Question**

Can Ethereum be directly used to rapidly deploy meaningful and sufficiently performing trusted applications with added value over traditional approaches?

2. **Installing Ethereum**

OS X: OK!

Ubuntu: OK!

Debian: Are you kidding me?



3. Understanding Ethereum

- WHERE IS THE DOCUMENTATION????
- Ether = the internal currency
- Blockchain = our motivating factor
- Smart contracts, smart 'democracy'

4. Understanding Ethereum



- Contract \approx OO class
- Instance \approx OO object
- Transaction = interaction with anything
- Gas = nothing is for free



5. Understanding Ethereum

- Ethereum \neq Bitcoin
- Ethash (Dagger-Hashimoto)
- 12 sec. block time
- Genesis block???

4. Understanding Ethereum

- Contract ≈ OO class
- Instance ≈ OO object
- Transaction = interaction with anything
- Gas = nothing is for free

3. Understanding Ethereum

- WHERE IS THE DOCUMENTATION????
- Ether = the internal currency
- Blockchain = our motivating factor
- Smart contracts, smart 'democracy'

5. Understanding Ethereum

- Ethereum ≠ Bitcoin
- Ethash (Dagger-Hashimoto)
- 12 sec. block time
- Genesis block???

4. Understanding Ethereum

- Contract ≈ OO class
- Instance ≈ OO object
- Transaction = interaction with anything
- Gas = nothing is for free

3. Understanding Ethereum

- WHERE IS THE DOCUMENTATION????
- Ether = the internal currency
- Blockchain = our motivating factor
- Smart contracts, smart 'democracy'

5. Understanding Ethereum

- Ethereum ≠ Bitcoin
- Ethash (Dagger-Hashimoto)
- 12 sec. block time
- Genesis block???

7. Setting up a test network

- 'Useless peer'...!!!
- Peers largely out of sync
- Nodes mining frantically
- Raise the initial mining difficulty

6. Setting up a test network

- No 'real' ether to spend
- Using 'counterfeit' ether
- Nodes can't peer :(
- 'Use a large network ID'
- ...not too large

8. Setting up a test network

- Turn off peer discovery
- Add peers statically
- Network stabilizes
- Send sample transactions
- SUCCESS!!!

7. Setting up a test network

- 'Useless peer'...!!!
- Peers largely out of sync
- Nodes mining frantically
- Raise the initial mining difficulty

6. Setting up a test network

- No 'real' ether to spend
- Using 'counterfeit' ether
- Nodes can't peer :(
- 'Use a large network ID'
- ...not too large

8. Setting up a test network

- Turn off peer discovery
- Add peers statically
- Network stabilizes
- Send sample transactions
- SUCCESS!!!

7. Setting up a test network

- 'Useless peer'...!!!
- Peers largely out of sync
- Nodes mining frantically
- Raise the initial mining difficulty

6. Setting up a test network

- No 'real' ether to spend
- Using 'counterfeit' ether
- Nodes can't peer :(
- 'Use a large network ID'
- ...not too large

8. Setting up a test network

- Turn off peer discovery
- Add peers statically
- Network stabilizes
- Send sample transactions
- SUCCESS!!!

10. A simple contract

- Broadcast some messages
- Only the recipient can read their own messages
- (Not really...)

9. Setting up a registrar

- At least 3 different guides
- This makes no sense
- 'With a little help from my friends'
- Works, no idea why...

11. An advanced contract

- A court of law for civil cases
- Are you kidding?
- Plaintiff, defendant, bench, jury, treasurer.
- 'Justice' token
- Adversary system (debate, vote)

10. A simple contract

- Broadcast some messages
- Only the recipient can read their own messages
- (Not really...)

9. Setting up a registrar

- At least 3 different guides
- This makes no sense
- 'With a little help from my friends'
- Works, no idea why...

11. An advanced contract

- A court of law for civil cases
- Are you kidding?
- Plaintiff, defendant, bench, jury, treasurer.
- 'Justice' token
- Adversary system (debate, vote)

10. A simple contract

- Broadcast some messages
- Only the recipient can read their own messages
- (Not really...)

9. Setting up a registrar

- At least 3 different guides
- This makes no sense
- 'With a little help from my friends'
- Works, no idea why...

11. An advanced contract

- A court of law for civil cases
- Are you kidding?
- Plaintiff, defendant, bench, jury, treasurer.
- 'Justice' token
- Adversary system (debate, vote)

13. Writing setup scripts

- Instantiate a registrar**
- Register entities**
- Distribute ether**
- Compile and instantiate case contract**
- Watchbots.js to consume events**

12. Developing the contract

- 'Solidity'**
- Variables, mappings**
- Constant Functions**
- Transactional Functions**
- Events**
- Modifiers**

14. Executing the contract

- Assume roles for each participating node**
- Run the setup scripts**
- Interact with the contract and see events flowing in**
- Mostly works as designed**

13. Writing setup scripts

- Instantiate a registrar
- Register entities
- Distribute ether
- Compile and instantiate case contract
- Watchbots.js to consume events

12. Developing the contract

- 'Solidity'
- Variables, mappings
- Constant Functions
- Transactional Functions
- Events
- Modifiers

14. Executing the contract

- Assume roles for each participating node
- Run the setup scripts
- Interact with the contract and see events flowing in
- Mostly works as designed

13. Writing setup scripts

- Instantiate a registrar
- Register entities
- Distribute ether
- Compile and instantiate case contract
- Watchbots.js to consume events

12. Developing the contract

- 'Solidity'
- Variables, mappings
- Constant Functions
- Transactional Functions
- Events
- Modifiers

14. Executing the contract

- Assume roles for each participating node
- Run the setup scripts
- Interact with the contract and see events flowing in
- Mostly works as designed

16. Drawing more conclusions

- Debugging is not as agile as we're used to
- Everything needs to be mined
- Everything costs gas
- Live network: gas = money

15. Drawing a conclusion

- Dev < Ops
- Setting up: 2.5 Weeks
- Developing: 2.5 Days
- Very intuitive language

17. Drawing further conclusions

- Great leverage for 'trustless' applications
- Very powerful concepts
- Huge lack of documentation
- Low project maturity
- Great potential

16. Drawing more conclusions

- Debugging is not as agile as we're used to
- Everything needs to be mined
- Everything costs gas
- Live network: gas = money

15. Drawing a conclusion

- Dev < Ops
- Setting up: 2.5 Weeks
- Developing: 2.5 Days
- Very intuitive language

17. Drawing further conclusions

- Great leverage for 'trustless' applications
- Very powerful concepts
- Huge lack of documentation
- Low project maturity
- Great potential

16. Drawing more conclusions

- Debugging is not as agile as we're used to
- Everything needs to be mined
- Everything costs gas
- Live network: gas = money

15. Drawing a conclusion

- Dev < Ops
- Setting up: 2.5 Weeks
- Developing: 2.5 Days
- Very intuitive language

17. Drawing further conclusions

- Great leverage for 'trustless' applications
- Very powerful concepts
- Huge lack of documentation
- Low project maturity
- Great potential

19. Our future

- Fully develop the court contract
- Proper benchmarking (performance, cost)
- Evaluate security
- Use contracts to research open problems

18. Ethereum future

- Whisper
- Swarm
- Mist
- Ethereum itself

19. Our future

- Fully develop the court contract
- Proper benchmarking (performance, cost)
- Evaluate security
- Use contracts to research open problems

18. Ethereum future

- Whisper
- Swarm
- Mist
- Ethereum itself

19. Our future

- Fully develop the court contract
- Proper benchmarking (performance, cost)
- Evaluate security
- Use contracts to research open problems

18. Ethereum future

- Whisper
- Swarm
- Mist
- Ethereum itself

K thx

Ask me stuff!