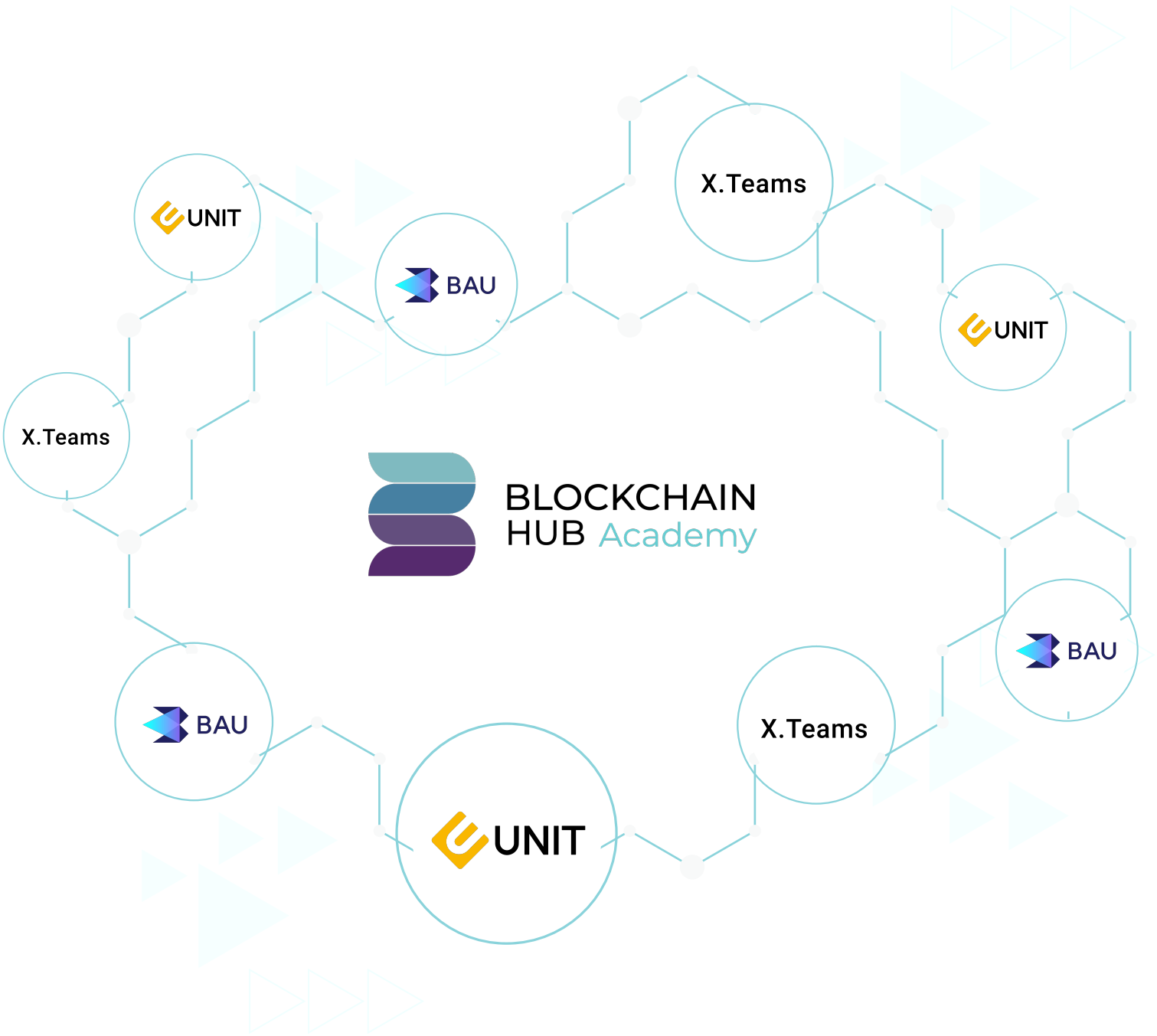
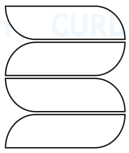

UNIT
BAU
X.Teams
UNIT
X.Teams
BLOCKCHAIN
HUB Academy
BAU
BAU
X.Teams
UNIT

PoSt-Satoshi era

PS: 1
Created: 2019



BLOCKCHAIN
HUB Academy

CONTENTS

GENRAL INSTRUCTIONS

Technical organization:

Submit your finished tasks to the GIT repository.
https://xteams-gitlab.unit.ua/module-N-login

Only work submitted to the repository will be considered for mentor-evaluation. Any extraneous files will be considered against you if this is not justified by any serious cause.

Deadline:

Access to the repository for making records closes after 7 days since tasks were presented at 09:01:00 AM GMT+2

XBIP: 5 - 25:02:2018 at 09:01:00 AM GMT+2

PART 00: Introduction

Congratulations, you've completed Satoshi Way and gained a lot of skills. Satoshi left great legacy and time was changing..The second loud case in the Blockchain community became Ethereum. This project proposed itself as world decentralized computer, which guaranteed code is law based on the blockchain technology. Programmers could apply different scenarios by smart contracts. Tool for this purposes is Solity, turing complete programming language.

Also, Ethereum provided the first ICO experience. On that moment it seemed incredible opportunity to fund project by decentralization way. That's about the light side of ICO. Hereinafter, ICO's became not so trust way to funding and the light side turned over to the dark.

In spite of this process, it gave rise to new standards of development in decentralized environment. In general, it's a questions about robustness of Turing complete programming languages with decentralized approach "code is law". In particular, ICO bring to Ethereum new token standards like ERC20, tokenomy, patterns and security best practices, first major hacks and stupid bugs in production. We cannot avoid this phenomenon and give you a chance to take part in this piece of history by creating your own ICO for educational purposes. Also, it'll bring you experience in Ethereum smart contract development by real historical cases, deeping dive into Solidity language and other stuff.

PART 01: Weekly task

This week you have to build blockchain application using the Ethereum Blockchain. It’s a one project for the full first week of the PoST-Satoshi era. This project includes implementation the most typical (and maybe scammy) ICO. It’ll bring you a lot of Solidity development skills, by the way.

We need to figure out this topic ONCE and close it, keeping developers knowledge for further development.

Imagine, you are Solidity developer and you have ICO requirements specification from one of the projects. Please provide your technical solution with smart contract on the Ethereum. The reviewer will have to compile the project in Remix IDE, deploy it and check it by calling different methods from online IDE interface. Keep in mind, how to connect from online IDE to you localhost blockchain and how to manage pre-generated keys (in the case of Ganache usage).

PART 02: Technical onboarding

For the Ethereum smart contracts development please use Solidity language. The code environment could be different - you can use online IDE like Remix or customize your code editor. For signing transactions and other interactions with Ethereum blockchain you can use Metamask wallet. For the development environment you can use Truffle framework. This one helps your design, test, and deploy secure smart contracts. Run your personal private blockchain on the localhost by Ganache and you can deploy smart contract to this one by simple scripts. Remix also includes deploy functionality to the localhost or any chosen node from different networks (mainnet, testnets). Feel free use any available developers helpers and scripts, Solidity libraries and contracts from open source (eg. OpenZeppelin). Mocha testing framework and Chai for assertions to provide tools for testing and help you to check the business logic during the development process (eg. revert time in the blockchain, wow).

PART 03: Requirements specification

Our smart contract's built for distributing our token to investors through private sales, presales, ICO crowd-sales and post-ICO. The code's on solidity last version and based on ERC20 standard. This contract will have full of features to support us perfectly in the business flows. The basic name of token is "Shitcoin", but you can provide your own version.

1. Actors

Some actors will interact with our smart contract as below:

Contract owner
The address will be used to deploy smart contract to main net. It will have full of permissions to execute functions.

Contract admin
The address will be used as administrator. It will have permissions to execute the business functions related to token allocation and distribution.

Contract portal
The address will be used just for automatic KYC or confirm private investor through our portal.

Private investor
The address' of private investor. It will be added to private list by smart contract.

Investor in whitelist
The address joins in presales and ICO. It's completed KYC verification.

Investor not in whitelist
The address joins in presales and ICO. It's not completed KYC verification.

1. Features
Our smart contract will have so many functions authorized for relevant actors following the table below:

Function	Description	Public	_Self	Owner	Admin	Portal	All + investor
GreenX	Constructor function			X			
transfer	ERC20 standard function	X					X
transferFrom	ERC20 standard function	X					X
approve	ERC20 standard function	X					X
allowance	ERC20 standard function	X					X
balanceOf	ERC20 standard function	X					X
()	Payable function to distribute token	X	X				X
getCurrentState	Get current state of sales campaign	X					X
issueTokenForPrivate Investor	To distribute token to private investor		X				
issueTokenForPresale	To distribute token to normal investors joined presales		X				
issueTokenForICO	To distribute token to normal investors joined ICO		X				

trackdownInvestedEther	To track invested amount of Ether of investors not completed KYC		X				
issueToken	To distribute token to investor and transfer ETH to our wallet		X				
addToWhitelist	To add new addresses to whitelist	X		X	X	X	
removeFromWhitelist	To remove addresses from whitelist	X		X	X	X	
addPrivateInvestor	To add new addresses to private list	X		X	X	X	
removePrivateInvestor	To remove addresses from private list	X		X	X	X	
startPrivateSale	To start private sales	X		X	X		
startPreSale	To start presales	X		X	X		
endPreSale	To end presales	X		X	X		
startICO	To start ICO	X		X	X		
endICO	To end ICO	X		X	X		
setPrivateSalePrice	To set price before starting private sales	X		X	X		
setPreSalePrice	To set price before starting presales	X		X	X		
setICOPrice	To set standard price before starting ICO	X		X	X		
revokeToken	Payable function to revoke token	X		X	X		
activateContract	To activate contract	X		X			
deactivateContract	To deactivate contract	X		X			
enableTokenTransfer	To enable transferring of token	X		X			
changeFundKeeper	To change the ETH wallet	X		X			
changeAdminAddress	To change admin address	X		X			
changePortalAddress	To change portal address	X		X			

changeFounderAddress	To change founder address	X		X	X		
changeTeamAddress	To change team address	X		X	X		
changeReservedAddress	To change reserved address	X		X	X		
allocateTokenForFounder	To allocate tokens to founder	X		X	X		
allocateTokenForTeam	To allocate tokens to team	X		X	X		
moveAllAvailableToken	To move all tokens remaining after ICO to external address	X		X	X		
allocateReservedToken	To allocate reserved token to external address	X		X	X		

3. Workflow

Token Allocation & Distribution

Owner will deploy smart contract to create token and contract to follow the allocation & distribution policies below

Token Summary	
Number of token created	500 000 000
ICO Price per token	$0.20
ETH Price used for calculation	$600.00
Shitcoin tokens per ETH	3000
Theoretical market cap	$100 000 000
Token Issued for ICO	300 000 000

Total token distribution		Token amount
Team	16%	80 000 000
Public Sales	60%	300 000 000
Advisor & Partners	10%	50 000 000
Bonuses for early token investors	4%	20 000 000
Reserved tokens	10%	50 000 000
	100%	500 000 000

Tokens reserved for team and investors		Token amount
Seed investor	4%	20 000 000
Founders (LOCKED - 12 minutes vesting)	10%	50 000 000
Reserved	10%	50 000 000
Advisors	10%	50 000 000
Team (LOCKED - 12 minutes vesting)	6%	30 000 000
	100%	500 000 000

Token distribution during ICO Rounds

Round	Bonuses	Projected Sale	Token Price	Token Issued
Private Sale	60%	6 000 000	$0.13	48 000 000
Pre-Sale	30%	5 000 000	$0.15	32 500 000
ICO - Phase 1	20%	10 000 000	$0.17	60 000 000
ICO - Phase 2	10%	10 000 000	$0.18	55 000 000
ICO - Phase 3	0%	10 000 000	$0.20	50 000 000
Future donations				54 000 000
				300 000 000

There’re 3 rounds including private sales, crowd presales and official ICO.

For private sale, the private investors can buy token with bonus 60%, this round will last to the end of ICO.

For crowd-sales, the normal investors are required to complete KYC to buy token with bonus 30%.

The official ICO will happen for 9 minutes including 3 rounds and last 3 minutes for each one.

1st round : bonus 20%
2nd round : bonus 10%
3rd round : bonus 0%

For reserved tokens

We have 200 mil tokens for reservation, these tokens will be allocated following the details below

- Seed investor : 4 % tokens will be available right away after ICO ended.
- Advisor : 10% tokens will be available right away after ICO ended.
- Founder : 10% tokens will be vested for 12 minutes for 3 times
  20% after ICO ended
  30% after 6 minutes later
  50% after 12 minutes later
- Team : 6% tokens will be vested for 12 months for 3 times
  20% after ICO ended
  30% after 6 minutes later
  50% after 12 minutes later

4. Main workflow for private sales, presales, ICO and post-ICO

Contract owner and admin / portal will call the relevant functions of smart contract to complete all steps in this main workflow.

Add To Private List
Admin can add private investors' addresses to private list in contract by calling the function "addToPrivateList". The addresses in private list can buy tokens with bonuses and policies for private investors.

Add To White List
Normal investors completed KYC verification on our portal, then the portal will call the contract function "addToWhiteList" to add their addresses to whitelist in smart contract.
The addresses in whitelist can buy tokens with bonuses and policies for KYC investors relevantly with each sales round. The investors not completed the KYC verification can buy tokens similarly as KYC investors without referral bonuses, and their tokens will be revoked by smart contract after finishing ICO. Our KYC verification will happened continuously from private sales to 15 days after finishing ICO, the investors need to complete this process in this period to be added to whitelist, after that, they're non-KYC investors.

Sale Tokens

The admin will call contract functions through sales campaign to control it, generally, our process will be
- The admin will set sale price
- The admin will start sale round
- The investor will send ETH to contract
- Contract will transfer ETH to wallet
- Contract will check investor's address to clarify the suitable cases to distribute the relevant tokens with bonuses on price to investor
- Contract will track the raised ETH of non-KYC addresses to be used for revoking process later.
- The admin will end sale round

Revoke Tokens

- For addresses not in private list and whitelist, we will do revoking tokens after KYC ended, the process will be
- The admin will check the total invested ETH we need to return.
- This amount of ETH will be transferred from wallet to admin address
- The admin will call contract function "revokeToken" to return ETH to relevant address and revoke tokens to contract.

Note:
The variant amount of refunded ETH by transaction fee will be calculated for investors.

Allocate Tokens

- After ICO ended, tokens can be allocated to external address for relevant allocations with specific conditions, the general process will be
- For founder, team, reserved token
- The admin will set the relevant external addresses for allocating
- The admin will get the value of addresses to make sure that they're exact
- The admin will allocate the relevant tokens to the relevant addresses.
- For remaining tokens in sales
- The admin will allocate the available sales tokens to the relevant external address

Enable Transferring

After ICO and post-ICO completed successfully, the admin will call function to enable transferrable function for tokens.