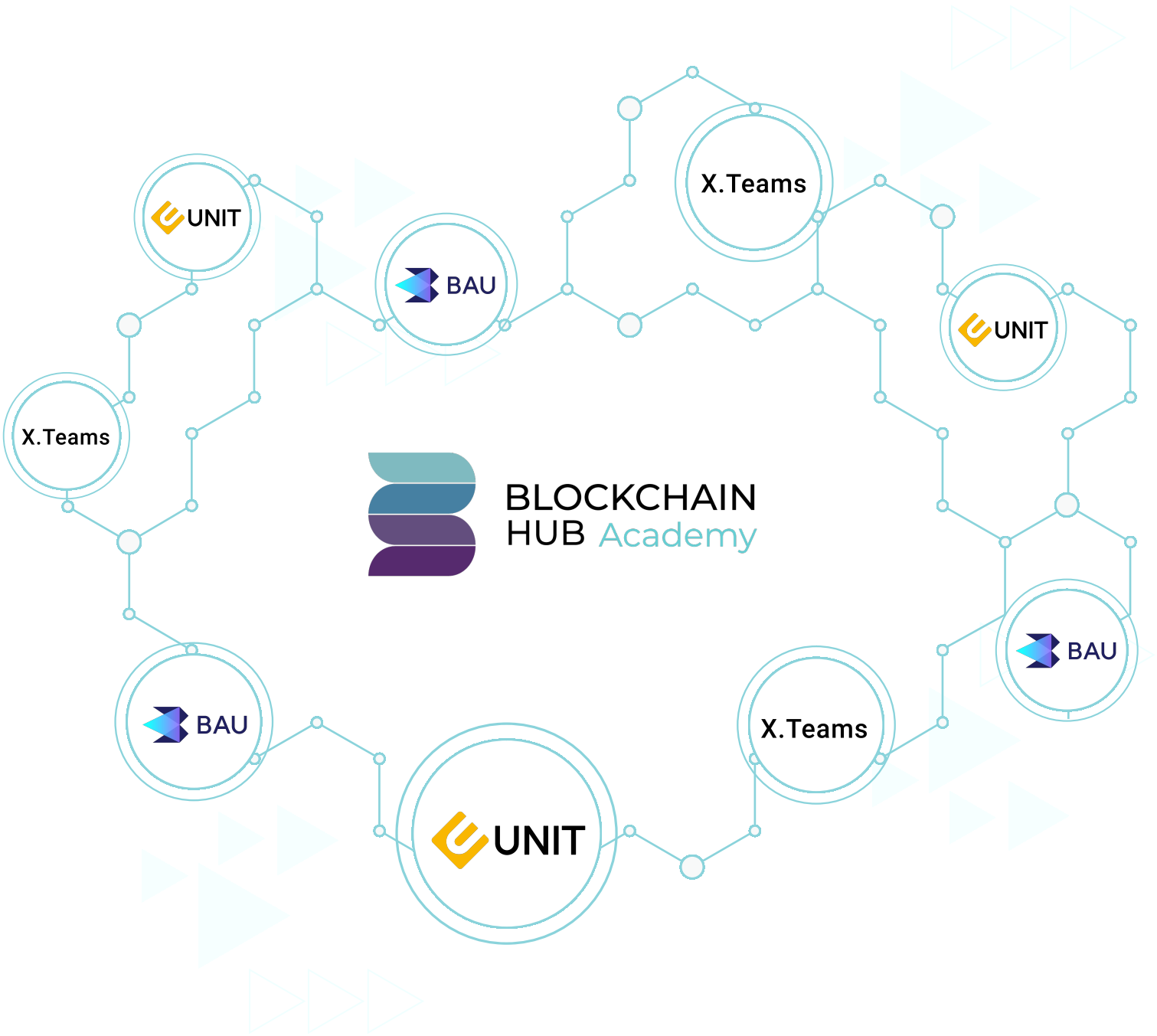
UNIT
BAU
X.Teams
UNIT
X.Teams
BLOCKCHAIN
HUB Academy
BAU
BAU
X.Teams
UNIT

Satoshi way

XBIP: 4
Title: Final Countdown
Created: 2019

CONTENTS

GENERAL 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: 4 - 11:02:2018 at 09:01:00 AM GMT+2

INTRODUCTION

This week you will have a chance to implement prototype of application from real world use case. This application need to be developed using Bitcoin feachers, implemented in your Pitcoin project. You can choose one of them project below. Also, this week contains one group task.

PART 00: Pitcoin mainnet

It’s a time to get an experience, using the Pitcoin like a first crypto anarchists. Those guys were pioneers of Bitcoin mining on CPU, code development, debugging and supporting. This task means group interaction, because the blockchain really can live based on state machine replication - method for implementing a fault-tolerant service by replicating servers and coordinating client interactions with server replicas.

On this moment you have to have working Pitcoin full node with different levels of implementation, according to your progress. In this step you can earn some point by the running Pitcoin full blockchain node with other student implementations.

You can initiate or find and join the group with your level of Pitcoin project implementation. For this one you can versionized your transaction in your group.

Launch network

The task is run Pitcoin blockchain infrastructure on the Raspberry Pi with several full nodes. Leave you nodes running on the Raspberry Pi, especially before the review

- Increase difficulty to avoid instant mining process in a config before starting
- Launch several nodes with premining mode and synchronize it
- Make a fun with broadcasting transactions and exploring the blockchain state changes

Attack network

A 51% attack is a potential attack on the bitcoin network whereby an organization is somehow able to control the majority of the network mining power (hashrate). In this task you can emulate this attack.

- Launch several malicious nodes (with more hash power then Pitcoin mainnet) with premine flag, which based on new settings - edit total supply and first mining reward transaction by increasing this constants in ten times
- Synchronize it between each other
- Connect to valid Pitcoin mainnet and try to replace blockchain with malicious chain
- Make fun with broadcasting malicious transaction and get profit with a large mining reward fee

The review will contain previous interactions and will be happened with running node on Raspberry Pi.

PART 01: Personal project

The final countdown of Satoshi Way it’s implementation of the case, which could be used in the real world. Please choose one of below, according to your preferences. All the projects are equal to each other in context of review points. Most of your personal project should be covered with tests.

xWallet

This project means wallet implementation. Feel free to use previous code from transaction builder, signing, broadcasting process from previous codebase. Yeah, Don’t Repeat Yourself is partly broken in this way, but it assumps this application need to be separate from Pitcoin full node (like in real life - some wallets have separate implementation from Bitcoin full node, eg Electrum and Bitcoin Core integration).

There will be following options, implemented in **xWallet** from scratch. Please don’t use additional libraries for this key management tasks:

- Implementation of BIP32 - Hierarchical Deterministic Wallets
- Implementation of BIP39 - Mnemonic code for generating deterministic keys
- User interaction with cli. Use previous command for transaction and key management and update for importing WIF and getting. Also add new command for getting 10 receiving addresses and 10 change addresses, derived from one seed phrase
- Getting wallet balance with combination of all assets on the wallet addresses

The following options allow to be implemented with production-ready Bitcoin python libraries. You can integrate it to the xWallet for the following features:

- Add multisignature functional for your wallet. It means creating multisig script address for 2 of 2 keys
- Signed and broadcast any type of transactions (P2PKH and P2SH) to the Bitcoin testnet
- Add appropriate command for this multisig features to your wallet cli

Payment processor

This project is a chance to implement tool for automated actions like receiving BTC from a different users. You have to develop a payment processor to handle deposit process. User can get the invoice for a payment or get new the own address from a payment processor and send Bitcoin to this address.

There will be backend payment processor application, which interact with Bitcoin testnet and application database. For this project you can use any of Bitcoin libraries and tools.

Application can receive necessary information by the web requests. You can choose database according to your preference. It's for saving payment process data like account name, belonged address, amount of deposit BTC etc.

Despite on the payment type - invoice or deposit, every user will have a personal unique bitcoin address during the payment process and associated with him. Each address pre-generated from the one seed phrase and stored into DB (total amount is 100).

The payment flow:

- User send request with payload (json format). It's include one of two types of payment: bitcoin invoice or bitcoin deposit, account name, sender address that will be use to send Bitcoins (private key is under user control) and amount. Last two parameters not required in case of bitcoin deposit.
- Payment processor make a DB record with this user data and associate unique bitcoin address from pre-generated set.
- In case of bitcoin invoice type, after receiving request, payment processor has to build the unsigned raw transaction (according to the request payload) with user address for payment, amount and personal receiving address, previously getting from payment processor and send this transaction in response. Also, receiving address associated with the user and stored in DB with pending status
- In case of bitcoin deposit type, payment processor return unique bitcoin address in response and storing receiving address into DB with pending status.
- Payment service has to follow updates from the Bitcoin testnet. All of receiving address with pending status should be scanned into the Bitcoin testnet network, and after two confirmations (production ready solution is six, but we try to decrease waiting time for prototyping aims) its status change to confirmed and deposit BTC value in DB, associated with user should be also updated

For the more convenient interaction with payment processor, you can make a Python helper for send web requests with payload, get responses. For signing unsigned raw transaction you can use Electrum wallet or another public available software for this aims.

Pitcoin Blockchain eXplorer

This case is above the core development and cover the user interface features. It’s implementation of Pitcoin blockchain explorer. This means separate frontend app, which can getting data through the public web api routers of Pitcoin network.

The eXplorer is a single page application. Feel free to use any kinds of frontend tools and frameworks like React, Angular, Vue or native JavaScript. There be following options of eXplorer application:

- Provide data about blocks in details, transactions in blocks, details in transaction from the Pitcoin network
- Providing getting block information by the height in search input
- Transaction linked mechanism for blockchain exploring

You can get inspiration looking on the blockchain.com or blockstream.info explorers. If you don’t have appropriate API for the data, don’t hesitate to implement it in your Pitcoin full node on the API side. Design fully up to you, it could be simple functional design.