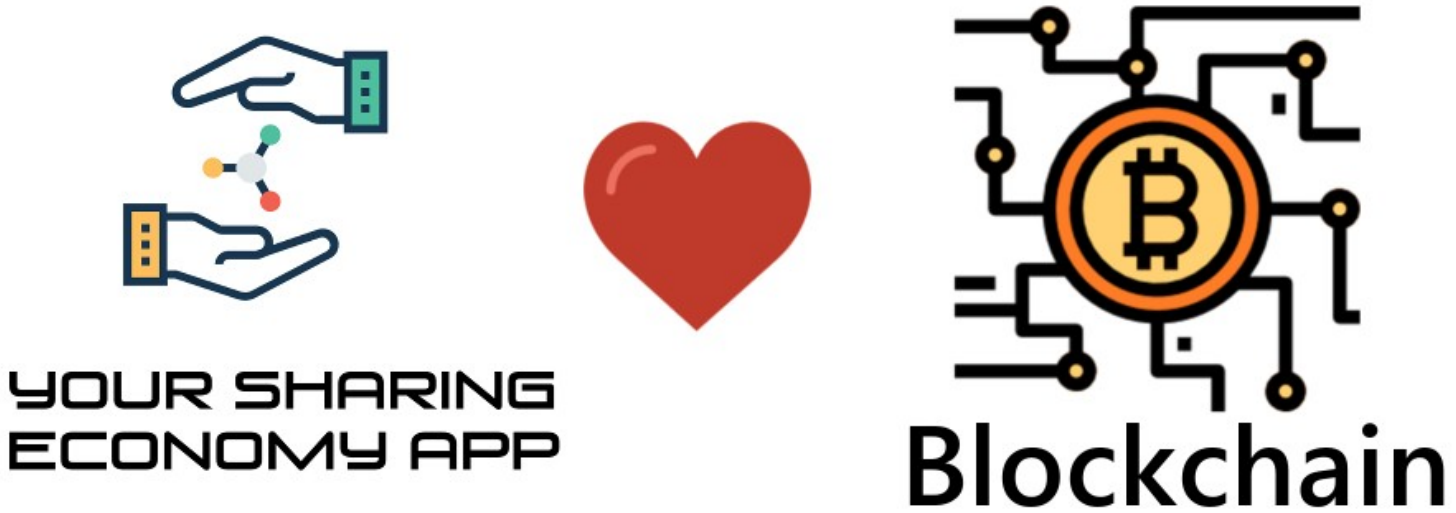
Case Study: Sharing Economy App + Blockchain



Presentation

- **Esteban Gallardo**
- **Freelance Programmer**
- **Specialized in VR/AR, networking apps**
- **Expert in Unity, C#, Java, PHP and anything to make a project come true**
- **Aspiring Entrepreneur :)**
- **Email: esteban@yourvrexperience.com**
- **Slack: https://yourvrexperience.slack.com**
- **GitHub: https://github.com/EstebanGameDevelopment**

3º Industrial Revolution

- I will keep it short:
  - The third industrial revolution is to understand that nowdays everything is connected in this world.
  - Projects are becoming more and more complex, and they can't be done by single teams but by a network of teams working synchronized.
  - The future workflow will be to get the most from the connections.
- The best reference I can give you is this video. If you really want to participate in a better future you have to see it:
  - https://www.youtube.com/watch?v=QX3M8Ka9vUA

Choose your own path

- The first part of the presentation we are going to focus to understand the use of C# Nbitcoin DLL to perform multiple operations with the Blockchain technology.
- After that we are going to see a real case project where we use this technology for a Sharing Economy App.
- Both parts are independents from each other, so you can leave at anytime if you are not interested in the topic.

Code C# NBitcoin

YOUR
BITCOIN
MANAGER

BitcoinController

- First we are going to run examples, then we are going to examine the code of that examples
- Download the repo:
  - https://github.com/EstebanGameDevelopment/YourBitcoinController
- Create a new Unity project and paste the code in "Assets" folder

Bitcoin Controller Example Manager

- Run BasicManager.unity scene
  - Explanation of Test Net and Main Net for Bitcoin
  - Open Visual Studio: BasicManager.cs and show the private keys we are using
  - Get Balance of all the accounts
  - Get transaction summary of an account
  - Perform a transaction

Bitcoin Controller
Example Create Keys

- Run CreateKeys.unity scene:
  - Create a new key
  - Add funds:
    - https://testnet.manu.backend.hamburg/faucet
  - Replace the key in the BasicManager.cs
  - Get balance of key

Bitcoin Controller Example SignData

- Run the SignTextData.unity scene:
  - Sign a text data
  - Verify the text data
- Run the SignImageData.unity scene:
  - Sign an image hash
  - Verify an image hash

Code C# BitcoinController Wallet Information

- Let's check how we are getting the balance of an Bitcoin Wallet.
  - BitcoinController.cs
    - decimal GetBalance(string _privateKey, bool _dispatchEvent = false)
- We are going to examine of we get all the information about all the transactions done on an wallet key address
  - BitcoinController.cs
    - void GetAllInformation(string _publicKeyAdress)

Code C# BitcoinController.cs Execute Transaction

- Let's take a look to the c# code used to make a payment.
  - BitcoinController.cs
    - ExecuteTransaction(string _title, decimal _finalFeeAmount, params PaymentModel[] _payments)
      - var customerPrivateKey = new BitcoinSecret(m_currentPrivateKey);
      - GetUnspentCoins(customerPrivateKey.GetAddress(), false);
      - GetTransactionInputCoins(customerPrivateKey.GetAddress(), coinsInCustomerWallet, totalAmountToPay);
      - Transaction customerTransaction = new Transaction();
      - AddInputs(customerTransaction, customerPrivateKey, coinsToSpendInVideo, totalAmountToPay);
      - AddOutputs(customerTransaction, customerPrivateKey, _finalFeeAmount, _payments, totalAmountToPay, coinsToSpendInVideo);
      - AddOutputMessage(customerTransaction, _title);
      - SignTransaction(customerTransaction, customerPrivateKey);
      - BroadcastTransaction(clientQBitNinja, customerTransaction);

Code C# BitcoinController.cs Signature Data

- Let's see how we can use our keys to sign the data for authentication purposes
  - string SignTextData(string _data, string _currentPrivateKey)
  - bool VerifySignedData(string _dataOriginal, string _dataSigned, string _publicKey)

Code C# Nbitcoin

- Exchange Information:
  - EVENT_BITCOINCONTROLLER_JSON_EXCHANGE_TABLE
    - https://blockchain.info/ticker
- Fee Information:
  - EVENT_BITCOINCONTROLLER_JSON_FEE_TABLE
    - https://bitcoinfees.earn.com/api/v1/fees/recommended

YOUR SHARING
ECONOMY APP

Create your own
Sharing Economy Application

1. Download

- Unity SDK
- Import the package YourSharingEconomyApp from Unity Asset Store
- Switch Scripting Runtime Version to:
  - **.NET 4.x Equivalent**
- Import the packages:
  - Facebook SDK package
  - Unity IAP
  - Flatcalendar package

2. XAMPP

- XAMPP install

  https://www.apachefriends.org/download.html

- Change port to avoid collision (usually Skype)

  - [XAMPP Installation Folder]/apache/conf/httpd.conf.
  - Listen 8080

3. Create Database

- Open PHPMyAdmin:
  - http://localhost:8080/phpmyadmin
- Create a database called "**yoursharingeconomyapp**" (you can call it later like you want but let's keep it this way in this tutorial)
- Import database from
  - Assets/YourSharingEconomyApp/SERVER/DATABASE

4. PHP

- Copy PHP into the folder:
  - SOURCE:
    - Assets/YourSharingEconomyApp/SERVER/PHP
- TO:
    - c:/xampp/htdocs/yoursharingeconomyapp

5. Basic Server Configuration

- Open configuration file and set up DATABASE and constants:

- c:/xampp/htdocs/yoursharingeconomyapp/
  - ConfigurationYourSharingEconomyApp.php
    - LOCAL SERVER CONNECTION
    - DATABASE CONNECTION
    - URL SERVICES

6. Basic Configuration Client

- Open **ScreenController.cs** and set the constant:

- **public const string URL_BASE_PHP = "http://localhost:8080/yoursharingeconomyapp/";**

7. Run the App and Register by Email

- Run the application and register a new user.
- PLEASE, USE A PASSWORD YOU CAN REMEMBER, THIS IS FOR TESTING SO, "123" IS OK :)
- For the example there is not a email verification process to ease this tutorial. If you want to enable the email verification process just check the file **ConfigurationYourSharingEconomyApp.php** and set variable:
  - $ENABLE_EMAIL_SERVER = 1

8. Create multiple accounts

- We are going to create another accounts. First, go here to clear local data:

- "Tools->Your Sharing Economy Tools->Clear PlayerPrefs"

- Repeat the process to create the another account so we will have two accounts. One will be the provider of services, the other will be the customer

9. Account Switcher

- Open file **YourSharingEconomyAppTools.cs.**
- In this file we are going to write down the email/password we have used before.
- This tool will allow to switch between account fast to be able to develop without problems.
- Let's do a fast test switching between the profiles we have previously created.

10. Create a request for service

- Next, one of our accounts is going to be a customer looking for a service.

- Now we are going to create a request.

11. Provider makes a proposal

- Next, we are going to take the rol of the provider of services and make a proposal for the previous job created.

- First, we go to the profile section, we activate that we are a provider of services and we can fill our profile.

- Next, we will search for work and make a proposal

12. Client accepts the proposals

- We go back to the client and accept the proposal of the provider

13. Provider finishs job

- It's up to the provider to finish the job, so we switch to his profile and we post an image with the results

14. Blockchain

- Time for the Blockchain
- We will create a Bitcoin wallet which we will use for 2 things:
  - First, make a payment using Bitcoins
  - Second, using the power of the Blockchain to verify the authenticity of the job performed and that the two parts had agreed to the description of that job.
- We are going to use TestNet in order to use the tools that the Blockchain offers without the need to spend real money.

15. Provider Sets Up Blockchain Address

- Before proceeding to the payment, we have to set up the Bitcoin Key Address of the provider in order to be able to execute the payment and be able to use the our signature on the data.
- In the section Profile we can create a new key.
- REMEMBER TO PRESS SEND EMAIL TO SAVE YOUR PRIVATE KEY ADDRESS.

16. Client Validation By Payment

- It's time for the customer to pay and verify that the picture belongs to the work done.
- We will press the button Payment.
- Since we didn't have set up a Blockchain Key Address we set up one now and we add funds.
- REMEMBER TO PRESS SEND EMAIL TO SAVE YOUR PRIVATE KEY ADDRESS.
- We proceed with the payment.

17. Client Scores Provider

- After the payment is completed the customer will have the option to score the provider.
- By scoring the work of the provider we acknowledge that the work relationship between provider and customer has been completed.
- This data is going to be signed with the Blockchain address we previouly created and that will prove that the customer has agreed with the current request completed and his feedback.

18. Provider Scores Customer

- Back to the service provider he is also able to provide some feedback about the customer to balance the things.
- Again the data is going to be signed so that way we agreed with the final conclusion of the job done.

19. Signed Deal

- We have used the Blockchain keys we previously generated to sign the data as a proof of authenticity.
- Thanks to the Blockchain we can sign any data with our address to prove that data belong to us.

20. Scores for both

- Now the score of both client and service provider has been added to their respective global scores.
- This scores are in the profiles of both the provider and the customer so the other customer and provider can get a reference of these persons.

Create your Custom solution

Custom Texts

- The whole thing is prepared in order to be able to make few changes and customize your own solution.
- First, let's see the file with all the texts. We keep it simple:
  - we just chage the title
  - Replace the word "Provider" by your own professional
- Now, let's do some customization of critical images. Copy and paste these images.
  - _CUSTOMIZE*.*
  - Assets\YourSharingEconomyApp\CLIENT\Resources\images

And... It's done!!

- You have your own sharing economy app!!!

....

You have a MVP

- ...Well, of course, it's not finished, but at least you know that you have a template that it's easy to customize
- Thanks to this template you can develop a MVP (Minimal Viable Product) at a really low cost
- You are free to use it and it can help to any community or providers of services who want to create a wider network of connections