Cryptocurrency News?

Markets and ICOs

How do you obtain cryptocurrency?

- You have to buy it!
- You can buy cryptocurrency using any exchange that accepts your fiat, or you can buy cryptocurrency using other cryptocurrency
    - Buying cryptocurrencies with other cryptocurrencies is also known as trading, and we will be diving into this today
- Exchanges are centralized markets for trading cryptocurrency
- We will dive into this in detail today

How do you get rid of cryptocurrency?

- You have to sell it!
- Exchanges are also useful for this too — although not every exchange lets you cash out in USD

How do cryptocurrencies obtain value?

- A cryptocurrency's value is determined by how much people are willing to buy it for — similar to a stock
- When a cryptocurrency has USD value x, and there are y coins in circulation, we say the cryptocurrency has a **market cap** of xy
    - For example, at the time of this writing, 1 Bitcoin is worth approx. $8000. Since there are currently approx. 17 million bitcoin in existence right now, we say Bitcoin's market cap is 17,000,000 * 8000 = 136,000,000,000 or $136 billion
- Market cap is driven mainly by price, since circulating supply of coins usually increases very slowly (and in some cases not at all, if the total supply of the coin was issued at launch)
    - Gives an overview of how much money is in a currency

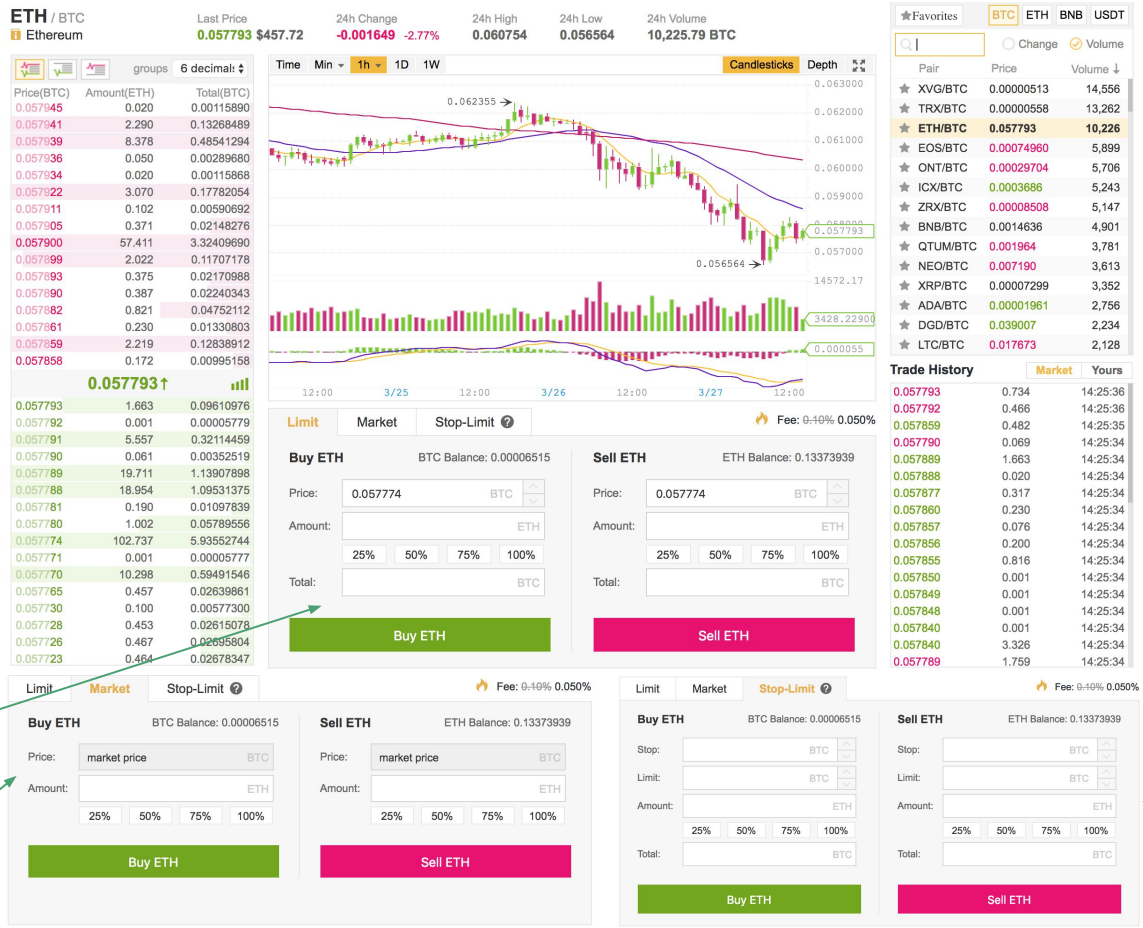
trading pair

current sell orders

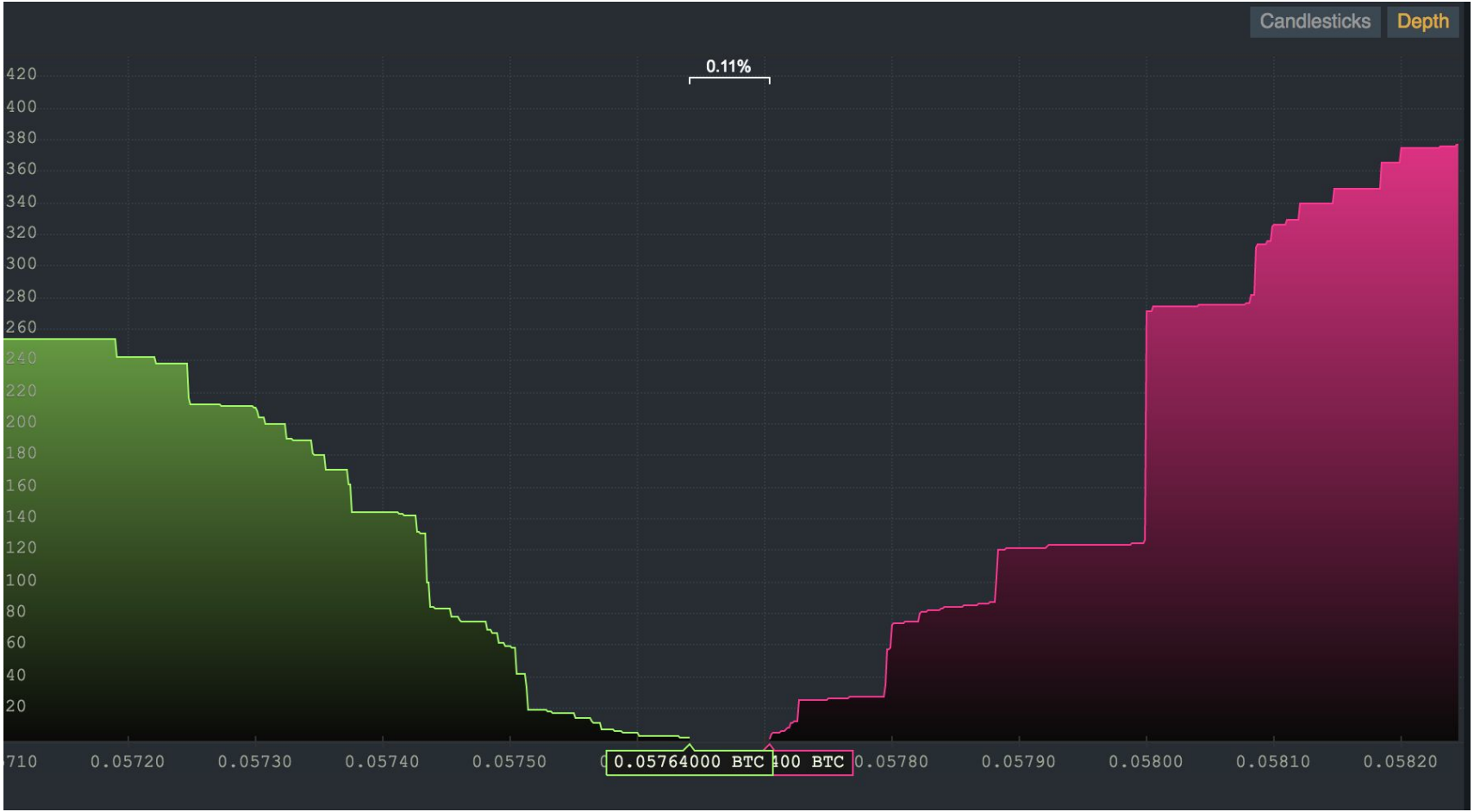
current buy orders

buy/sell with limit order

buy/sell at market



Stop-limit buy/sell

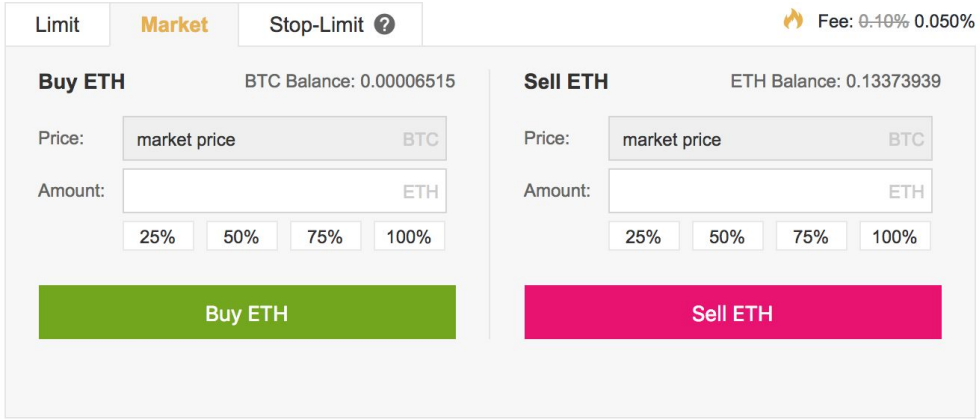

Candlesticks
Depth
0.11%
420
400
380
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0.05764000 BTC
0.05720
0.05730
0.05740
0.05750
0.05780
0.05790
0.05800
0.05810
0.05820

Buying/Selling cryptocurrency

- When you want to buy or sell cryptocurrency on an exchange, you must place an **order**
- These orders essentially broadcast a message to all other traders saying, “Someone would like to buy/sell cryptocurrency x using cryptocurrency y at price z”
- Once the order is placed, you must wait for it to be **filled**
- This means that once you hit the “buy” button, it does not necessarily mean that you’ve instantly made the purchase
- Let’s look at the different types of orders that can be made on exchanges

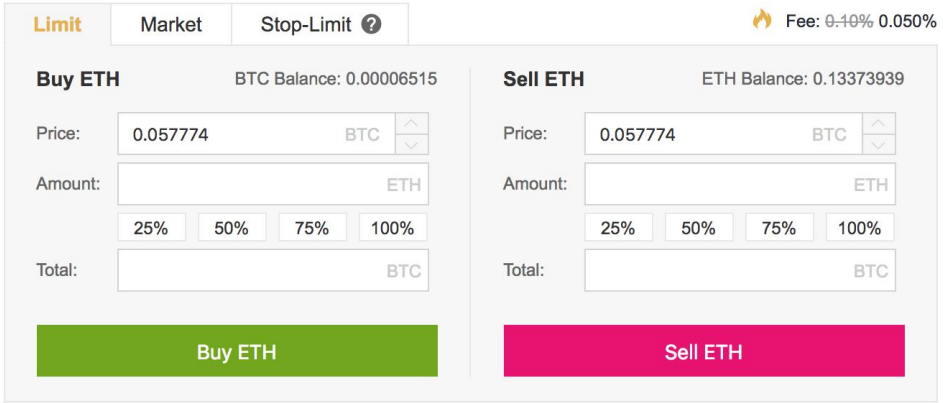
Market Buy/Sell

- When you place an order to buy or sell at market, your order will be filled at the best price possible (lowest buy, highest sell), no matter what the price is.
- If the price of the coin you are trying to purchase goes up as your order is being filled, you are forced to pay that possible upcharge



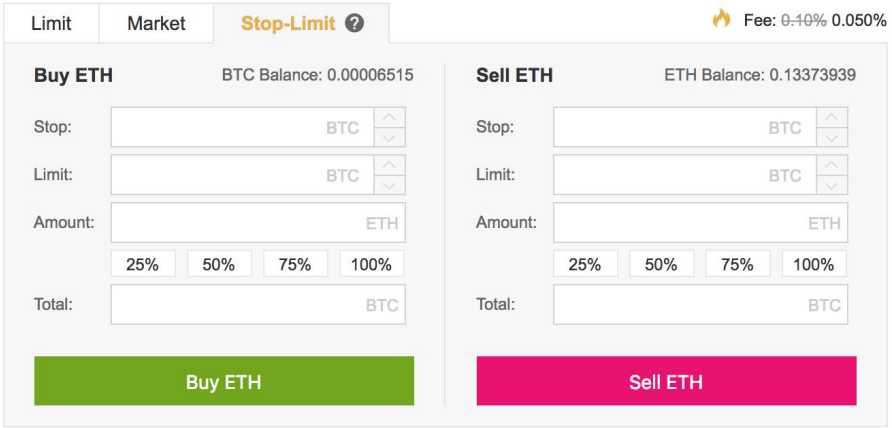
Limit Buy/Sell

- When you place an order at a limit x that you specify, your order will (likely) be filled when the price reaches that x or better (lower for buy, higher for sell)
- This type of order is beneficial in that you can always expect a certain price for your trade or better, but you do not have the market order guarantee of your purchase happening immediately (or ever)



Stop-Limit Buy/Sell

- With stop-limit orders, the user sets a stop price. When this stop price is reached, a subsequent limit-order is filled to make your purchase.
- Useful if you want to make a purchase when price reaches a certain point
  - For example, if I wanted to buy bitcoin when it reaches 10,000 again, I could put a stop-limit in with my stop at 10,000, allowing me to have that order filled in the future



Other types of orders

- Fill or Kill — this type of order specifies that if I want to buy 1 BTC at price x, then my order should fill the entire bitcoin, not just a fraction. If only fractional fills are available at the moment, kill the order
- Immediate or Cancel — this type of order specifies that if I want to buy 1 BTC at price x, then as much of my order as possible should be filled immediately — if my entire order can’t be filled immediately, cancel the rest of the order (i.e., I might be able to purchase 0.4 BTC at price x immediately, but no other orders will fill the other 0.6 BTC so I cancel the rest of the order and walk away with 0.4 BTC)

How do cryptocurrencies launch?

- Several methods — most common is ICO
- ICO stands for Initial Coin Offering
- Usually consists of people buying future coins using an existing cryptocurrency, and then receiving those coins when the network launches
    - This attempts to ‘start’ the market of the coin. Assuming the ICO is bought, the digital currency is now in the hands of adopters, investors, etc, and effectively has a price
- Two main types of ICO:
    - ERC20 tokens — coins built on top of the Ethereum network
    - Non-ERC20 tokens — coins that launch their own separate blockchain from every other coin
    - Because non-ERC20 coins are all unique and can have vastly different ICO strategies, we will focus on ERC20 in this course

ICO

- How does the ICO price of a coin get set?
  - Usually in terms of fraction of another coin (i.e., 0.1 ETH per coin)
  - This price is largely arbitrary but is loosely based on the total supply of the coin
- How does the ICO keep track of who to distribute coins to?
  - Recall the idea of private keys

ERC20 Tokens

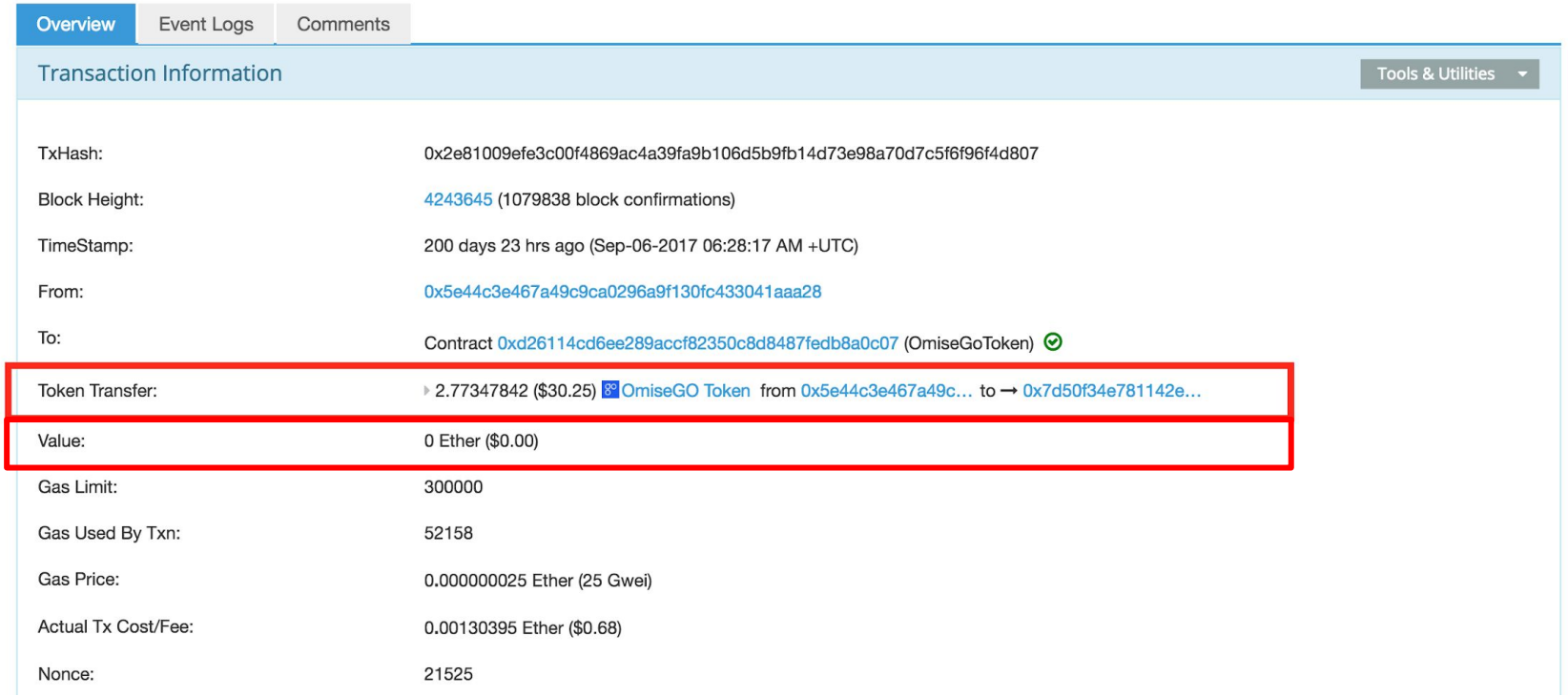
- ERC stands for Ethereum Request for Comments, which is a set of standard guidelines for tokens built on Ethereum’s network
- 20 is just the version of ERC that is currently being used by a plethora of popular coins
- ERC20 defines certain functions and behaviors that tokens have to implement

ERC20 Tokens

- ERC20 defines the following functions for tokens to implement:
  - totalSupply — gets total supply of the coin
  - balanceOf(address a) — gets the account balance of account with address a
  - transfer(address a, uint256 value) — transfer value coins to address a
  - transferFrom(address src, address dest, uint256 value) — transfer value coins from address src to address dest
  - approve(address spender, uint256 value) — allow spender to withdraw up to value coins from your account
  - allowance(address owner, address spender) — returns the amount which sender is allowed to withdraw from owner

ERC20 Tokens

- So how are ERC20 tokens bought and then distributed?
- Remember ERC20 tokens are built on top of Ethereum’s network
- This means your Ethereum address is also your token address
- You send your ether to the ICO, they can just send the token right back to the same address upon release


Overview
Event Logs
Comments
Transaction Information
Tools & Utilities
TxHash:
0x2e81009efe3c00f4869ac4a39fa9b106d5b9fb14d73e98a70d7c5f6f96f4d807
Block Height:
4243645 (1079838 block confirmations)
TimeStamp:
200 days 23 hrs ago (Sep-06-2017 06:28:17 AM +UTC)
From:
0x5e44c3e467a49c9ca0296a9f130fc433041aaa28
To:
Contract 0xd26114cd6ee289accf82350c8d8487fedb8a0c07 (OmiseGoToken)
Token Transfer:
2.77347842 ($30.25) OmiseGO Token from 0x5e44c3e467a49c… to 0x7d50f34e781142e…
Value:
0 Ether ($0.00)
Gas Limit:
300000
Gas Used By Txn:
52158
Gas Price:
0.000000025 Ether (25 Gwei)
Actual Tx Cost/Fee:
0.00130395 Ether ($0.68)
Nonce:
21525

ERC20 Tokens

- What does it mean to “send your ether to the ICO”?
- The ICO sets up a **smart contract** on Ethereum’s network
- Smart contracts have ethereum addresses, and run whenever they receive **any** amount of ethereum, including 0 (as we saw in the last picture)
- We will get into smart contracts in a couple weeks when we move into Ethereum
- Smart contracts are very powerful tools for executing payments on a network

Summary

- Exchanges act as intermediaries for trading cryptocurrencies back and forth
- Several types of orders are available depending on exactly how you want to execute a trade
- Cryptocurrencies mostly launch using Initial Coin Offerings (ICOs)
  - Can be ERC20 or non-ERC20
- ERC20 Tokens run on top of Ethereum’s network using smart contracts
- Your Ethereum address is also your ICO address for distribution
  - All addresses for ERC20 tokens are also valid Ethereum addresses
  - This is useful when you want to own many different cryptocurrencies — “one wallet fits all”

Final Project

- Out today, due at the end of the semester, 12/07 at midnight
- You will implement your own smart contract by devising your own applications which meets the requirements in the project's README
- Next week, we will work with Solidity in class to give a better overview of the language itself — be sure to bring your laptops!

Questions?