News

1. Faster download using Merkel Trees http://news.mit.edu/2019/vault-faster-more-efficient-cryptocurrency-0124
2. DeFi Hack, $320M in Ethere! https://blockworks.co/in-second-largest-defi-hack-ever-blockchain-bridge-loses-320m-ether/

Economics of Wealth

1. How is Wealth Created

    1. What is your time worth

        - $102,204 ($115,831 with MS in CoSc) per year in 5 years
        - 2x - you work twice as hard (80+ hrs a week)
        - 3x - you are 3 times as productive becasue you “skip the bs”
        - 3x - you work on someting important - an actual technology with productivity benefits = 18x as much

        This means 18 x 100,000 = 1.8 Mill a year = for a couple of years.

    2. What are the risks

        - Bad product market fit
        - Can’t raise capital
        - Wrong people
        - Business Fails

    “Leaverage” * “Measurability” = Wealth

    Leaverage is Technology Leaverage is New finanical instrument Leaverage is Other peoples money Leaverage is compounding of interest

    Measurability is Smallness

    1. Fallacies

        - Fixed pie falicy (I win you loose)
        - I can’t sell
        - FOP / FOF
        - Security Blanket
        - That you should “avoid” risk

2. Where is blockchain in this

    1. Attestation of Documents / Digital Assets
    2. Financial Instruments
    3. Productivity in Industry
    4. Supply Chain
    5. Financial Cleaning
    6. Shared Data

Blocks in our Go Code

Transactions in Blockchain

Data Structure from .../block/block.go:

```
// BlockType is a single block in the block chain.
type BlockType struct {
  Index         int                           // position of this
                                              // block in the
                                              // chain, 0, 1, ...
  Desc          string                        // if "genesis" str.
                                              // then this is a
                                              // genesis block.
  ThisBlockHash hash.BlockHashType            //
  PrevBlockHash hash.BlockHashType            // This is 0 len.
                                              // if this is a
                                              // "genesis" block
  Nonce         uint64                        //
  Seal          hash.SealType                //
  MerkleHash    hash.MerkleHashType          // Hw 03
  Tx            []*transactions.TransactionType // Tx for Block
}
```

Data Structure from .../transactions/tx.go:

```
type TransactionType struct {
  TxOffset       int               // The pos. of this in the block.
  Input          []TxInputType     // Set of inputs to a transaction
  Output         []TxOutputType    // Set of outputs to a tranaction
  SCOwnerAccount addr.AddressType  // ... for SmartContracts ...
  SCAddress      addr.AddressType  // ... for SmartContracts ...
  SCOutputData   string            // ... for SmartContracts ...
  Account        addr.AddressType  //
  Signature      lib.SignatureType //  Used in HW 5 - Signature
  Message        string            //  Used in HW 5 - Message
  Comment        string            //
}

type TxInputType struct {
  BlockNo     int // Which block is this from
  TxOffset    int // The transaction in the block.
                 // In the block[BlockHash].Tx[TxOffset]
  TxOutputPos int // Position of the output in the transaction.
                 // In the block[BlockHash].Tx[TxOffset].
                 // Output[TxOutptuPos]
  Amount      int // Value $$
}

type TxOutputType struct {
  BlockNo     int              // Which block is this in
  TxOffset    int              // Which transaction in this block.
                               // block[this].Tx[TxOffset]
  TxOutputPos int              // Pos. of the output in this block.
                               // In the  block[this].Tx[TxOffset].
                               // Output[TxOutptuPos]
  Account     addr.AddressType // Acctount funds go to (If this is
                               // ""chagne"" then this is the same
                               // as TransactionType.Account
  Amount      int              // Amoutn to go to accoutn
}
```