Stablecoins

Dan Boneh
Everything is a contract:

- Contracts manage state variables
- Contracts have functions that can be called externally
- Can inherit code from other contracts  \( \text{contract A is B,C} \)
- Types of contracts:  \text{contract, interface, abstract, library}

Global objects:  \text{block, msg, tx}
An example: ERC20 tokens

- https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md

- A standard API for fungible tokens. (ERC-721 for non-fungible tokens)

- An ERC20 token is itself a smart contract that maintains all user balances:
  
  ```solidity
  mapping(address => uint256) internal _balances;
  ```

- A standard interface allows other contracts to interact with every ERC20 token. No need for special logic for each token.
ERC20 token interface

function **transfer**(address _to,  uint256 _value) external returns (bool);

function **transferFrom**(address _from,  address _to,  uint256 _value) external returns (bool);

function **approve**(address _spender,  uint256 _value) external returns (bool);

function **totalSupply**() external view returns (uint256);

function **balanceOf**(address _owner) external view returns (uint256);

function **allowance**(address _owner, address _spender) external view returns (uint256);
How are ERC20 tokens transferred?

```solidity
contract ERC20 is IERC20 {

    mapping (address => uint256) internal _balances;

    function transfer(address _to, uint256 _value) external returns (bool) {
        require(_balances[msg.sender] >= _value, "ERC20_INSUFFICIENT_FUNDS");
        _balances[msg.sender] -= _value;
        _balances[_to] += _value;
        emit Transfer(msg.sender, _to, _value); // write log message
        return true;
    }
}
```

Tokens can be minted by a function `mint(address _to, uint256 _value) onlyOwner;`
Anyone can read ERC20 _balances[]

Transaction Hash: 0x6b85ca95e484d94503d1276456bfc32cc55f6fdb8bb231ff83....

Tells the USDC contract to transfer 10,010.00 USDC from Circle’s account to 0x7656159E42209A95b77aD374d...

| Storage Address: 0x4d3e7741e6c98c0c469419f9fe57ec622d7b26345802d22d17415768760f8 |
| Before: Hex → 0x00000000000000000000000000000000000000000000000000000000000000000000000000000000 |
| After: Hex → 0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000 |

Storage Address: 0x57d18af793d7300c4ba46d192ec7aa095070d6e6c52c687c6d0d92fb8532b305

| Before: Hex → 0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000 |
| After: Hex → 0x000000000000000000000000000000000000000000000000000000000000000000000000000000000000 |

(recipient’s entry)

(Circle’s entry)

(Circle’s balance after)

(etherscan.io)
Calling other contracts

Addresses can be cast to contract types.

```solidity
address _token;

ERC20Token tokenContract = ERC20Token(_token);
```

To call the “transfer” function of contract at address _token:

```solidity
tokenContract.transfer(_to, _value);
```
The world of DeFi

- Compound (lending)
- Uniswap (exchange)
- Aave (lending)

On-chain contracts:

- ERC20 coin A
- ERC20 coin B
- ERC20 coin C
- ERC721 NFT D

borrow 100 B from Compound

transfer 100 from me to Alice

Borrow 100 B from Compound transfer 100 from me to Alice
The world of DeFi

- Compound (lending)
- Uniswap (exchange)
- Aave (lending)

on-chain contracts

ERC20 coin A
ERC20 coin B
ERC20 coin C
ERC721 NFT D

Exchange 10B for 20C
Stable Coins
Stable Coins

A cryptocurrency designed to trade at a fixed price

• Examples: $1 \text{ coin} = 1 \text{ USD}, \quad 1 \text{ coin} = 1 \text{ EUR}, \quad 1 \text{ coin} = 1 \text{ USDX}$

Goals:

• Integrate real-world currencies into on-chain applications

• Enable people without easy access to USD, to hold and trade a USD-equivalent asset
### Types of stable coins

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Algorithmic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collateralized</strong></td>
<td>Custodial stablecoins (USD Coin)</td>
<td>Synthetics (DAI)</td>
</tr>
<tr>
<td><strong>Undercollateralized</strong></td>
<td>Central bank (digital) currency</td>
<td>Undercollateralized stablecoins</td>
</tr>
</tbody>
</table>
Custodial stablecoins

- Deploy ERC-20 contract
- 100 USD minted to Bob
- 35 USD minted to Alice
- State: Bob: 100, Alice: 35
- Balances:
  - 100 USD
  - 35 USD
Custodial stablecoins

pay Carol 15$ : transfer(Bob → Carol, 15) (and gas fee)

Transfers are done on-chain (custodian is not involved)

135 USD

Bob: 850
Alice: 35
Carol: 15

_balances
Custodial stablecoins

withdraw 60 USD

60 USD

burn(Bob, 60)

Custodian

135 USD

Bob: 85
Alice: 35
Carol: 15

_balances
# Two Examples

<table>
<thead>
<tr>
<th></th>
<th>Coins issued</th>
<th>24h volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>USDC</td>
<td>43.93 B</td>
<td>2.56 B</td>
</tr>
<tr>
<td>USDT</td>
<td>68.45 B</td>
<td>31.98 B</td>
</tr>
</tbody>
</table>
Some issues

Custodian keeps treasury in a traditional bank
• Must be audited to ensure treasury is available
• What happens if a hack steals treasury?

Custodian has strong powers:
• Can censor customers / refuse withdrawal requests
• Custodian can remove funds from user balances
Synthetics
The challenge: can we build a non-custodial stable coin?

• Collateral has to be a crypto currency like ETH
• The problem: ETH is not stable vs. USD
  [preferably not using USDC ... re-introduce custodian]

MakerDAO: building a stablecoin from an unstable asset
  Goal: 1 DAI = 1 USD
Two types of tokens:

- **DAI**: the stable coin (price the last year: 0.99 - 1.01 USD)
- **MKR**: anyone can buy MKR and earn interest.
  
  used for governance and to stabilize DAI in an emergency

---

Amount of DAI minted: 6.2 B (Oct. 2022)
Amount of MKR: 1 B
Alice wants to pay Bob in DAI (stable), but she has 1 ETH (unstable)

⇒ She creates a vault on the MakerDAO contract. She has:

- a **wallet**: for funds that she controls
- a **vault**: for locked funds

<table>
<thead>
<tr>
<th><strong>Alice’s Wallet</strong></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token</strong></td>
<td><strong>Balance</strong></td>
<td><strong>USD value</strong></td>
</tr>
<tr>
<td>ETH</td>
<td>1</td>
<td>$3000</td>
</tr>
<tr>
<td>DAI</td>
<td>0</td>
<td>$0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Alice’s Vault</strong></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token</strong></td>
<td><strong>Balance</strong></td>
<td><strong>USD value</strong></td>
</tr>
<tr>
<td>ETH</td>
<td>0</td>
<td>$0</td>
</tr>
<tr>
<td>DAI</td>
<td>0</td>
<td>$0</td>
</tr>
</tbody>
</table>
Alice locks up 1 ETH in her MakerDAO vault

<table>
<thead>
<tr>
<th>Alice’s Wallet</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token</strong></td>
<td><strong>Balance</strong></td>
<td><strong>USD value</strong></td>
</tr>
<tr>
<td>ETH</td>
<td>0</td>
<td>$0</td>
</tr>
<tr>
<td>DAI</td>
<td>0</td>
<td>$0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alice’s Vault</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token</strong></td>
<td><strong>Balance</strong></td>
<td><strong>USD value</strong></td>
</tr>
<tr>
<td>ETH</td>
<td>1</td>
<td>$3000</td>
</tr>
<tr>
<td>DAI</td>
<td>0</td>
<td>$0</td>
</tr>
</tbody>
</table>
MakerDAO: minting

She can use her locked ETH as collateral to borrow DAI into her wallet

130% collateralization ⇒ she can mint (borrow) up to 2300 DAI

- Alice can now pay Bob in DAI from her wallet
- she can repay her debt at any time at get her 1 ETH back
Alice pays interest on her borrowed DAI: \textit{stability fee}

- Most of the fee goes to DAI holders (via DAI savings rate: \textit{DSR})
- Some of the fee is paid as interest to MKR holders

<table>
<thead>
<tr>
<th>Alice’s Vault, at time $T+1$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token</strong></td>
<td><strong>Balance</strong></td>
<td><strong>USD value</strong></td>
</tr>
<tr>
<td>ETH</td>
<td>1</td>
<td>$3000</td>
</tr>
<tr>
<td>DAI</td>
<td>-2001</td>
<td>-$2001</td>
</tr>
</tbody>
</table>

debt increases over time
The DAI Savings Rate (DSR)

Anyone holding DAI can lock it up in the MakerDAO DSR contract

• **DSR**: the interest rate on DAI locked in the DSR contract
• Users can withdraw DAI from the DSR contract at any time

Why DSR? Encourages institutions to hold their idle assets in DAI
Stability mechanism

DAI trading below 1$ ⇒ *stability fee* and *DSR* are raised
⇒ minters are encouraged to repay their loan
⇒ reduces the liquid supply of DAI
⇒ DAI price goes up

DAI trading above 1$ ⇒ *stability fee* and *DSR* are lowered
⇒ minters are encouraged to mint more DAI
⇒ increases the liquid supply of DAI
⇒ DAI price goes down
Why collateral? Ensures Bob pays back his DAI debt.

- stability fee goes up $\Rightarrow$ Bob wants to repay DAI to get his collateral.

What if Vault debt exceeds 130% collateral?

<table>
<thead>
<tr>
<th>Bob’s vault at time $T + 100$</th>
<th>liquidation</th>
<th>Bob’s vault at time $T + 101$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token</strong></td>
<td><strong>Balance</strong></td>
<td><strong>USD value</strong></td>
</tr>
<tr>
<td>ETH</td>
<td>1</td>
<td>$3000$</td>
</tr>
<tr>
<td>DAI</td>
<td>$-2400$</td>
<td>$-2400$</td>
</tr>
</tbody>
</table>

insufficient collateral (>130%) Bob’s ETH collateral is auctioned off

(returns – fees) are used to pay off Bob’s debt until 130% is achieved.
In practice
The World of NFTs

Griffin Duniaif
The 4 Questions

1. What are NFTs?
2. Why are NFTs important?
3. What can be built on top of NFTs?
4. Where does this all go?
What are NFTs?

Token ownership of a digital asset
• Digital artwork, video game spaceship, virtual plot of land

No two NFTs are the same: they aren't mutually exchangeable

NFTs are defined by their:
• History, utility, appearance, cultural importance, etc.
Ownership of Digital Assets

What does owning a digital asset even mean?
- NFTs function identically to legal deeds

What is a deed?
- Legal contract that transfers "title" from one party to another
- Title is the legal status of owning property
- Title grants you certain legal rights:
  - Right to reside, renovate, accrue capital gains, etc.
NFTs as Deeds

Owning an NFT grants you title to digital property
• Owners of Bored Apes get access to a social network, verification on twitter, certain IP rights, etc.
• Axie holders get access to a game and earnings

A deed has tangible benefits and utility
• You are not buying a picture
• You are buying title, granting NFT-specific rights
How Do You Get an NFT?

Two methods:
1. Find the owner and barter
2. Go to a marketplace and bid

Purchase NFT from owner: Who sends first? Asset or money?
NFT Marketplaces

NFT marketplaces built as a collection of smart contracts
• OpenSea, Rarible, Axie, etc.

Why smart contract marketplace?
• Trustless: Don't need to give 3rd party custody of your asset
• Permissionless: Anyone with wallet can purchase or list
• Atomic: Payment for asset and delivery of asset done in one Tx
What Do We Have So Far?

A system of **property and exchange** for digital assets

- Facilitating efficient and verifiable property transactions
- This enables businesses to be built on top of NFTs
- Thereby driving the economic growth of the ecosystem
In the real world our economy rests upon a system of **property and exchange**

Without it:

- How would you mortgage your house?
- How would you buy or sell shares in a company?

National scale businesses rely on clear ownership and low-cost verifiability

- Banks, insurance companies, brokerages would look very, very different without such guarantees
Why NFTs?

They bring true **ownership** to the internet:
- You can resell an NFT
- You can accrue capital gains
- Emergent peer-to-peer activity

Enables a digital **commerce** layer:
- No confusion and conflicting claims (chain is source of truth)
- No platform risk (can't rewrite history or revoke ownership)
- Low-cost (can verify provenance with simple query)
- Composable (can use NFTs and services as Lego pieces)
The Service Layer
Gaming Guilds

One of the first inter-game financial institutions (Yield Guild Games)

The idea:

Source capital from LPs (by issuing a token)

=> Buy up swathes of virtual land, avatars, in-game items

=> Generate revenue by leasing assets to players

=> Pay LPs dividends

=> Take % spread

=> Accrue capital gains on the underlying assets
## Gaming Guilds

YGG's NFT holdings as of July, 2021

<table>
<thead>
<tr>
<th>NFTS OWNED</th>
<th>SUB-TOTAL (US$)</th>
<th>NUMBER</th>
<th>VALUE (US$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-total</td>
<td>$10,195,357</td>
<td>19,460</td>
<td></td>
</tr>
<tr>
<td>Founders Coin</td>
<td>$121,800</td>
<td>5</td>
<td>$24,360</td>
</tr>
<tr>
<td>Axies</td>
<td>$6,634,993</td>
<td>18,079</td>
<td>$367</td>
</tr>
<tr>
<td>Axie Land</td>
<td>$2,327,675</td>
<td>235</td>
<td>$9,905</td>
</tr>
<tr>
<td>Sandbox Land</td>
<td>$192,960</td>
<td>180</td>
<td>$1,072</td>
</tr>
<tr>
<td>Zed Run Horses</td>
<td>$267,960</td>
<td>86</td>
<td>$3,116</td>
</tr>
<tr>
<td>Embersword Land</td>
<td>$80,000</td>
<td>16</td>
<td>$5,000</td>
</tr>
<tr>
<td>F1 Delta Common Key</td>
<td>$6,890</td>
<td>130</td>
<td>$53</td>
</tr>
<tr>
<td>F1 Delta Time Parts</td>
<td>$68,110</td>
<td>328</td>
<td>$208</td>
</tr>
<tr>
<td>League of Kingdom Land</td>
<td>$315,219</td>
<td>386</td>
<td>$816.63</td>
</tr>
<tr>
<td>Cometh Ships</td>
<td>$9,350</td>
<td>10</td>
<td>$935</td>
</tr>
<tr>
<td>Splitterlands Regions and Untamed Booster Packs</td>
<td>$40,000</td>
<td>6,252</td>
<td>$6.40</td>
</tr>
<tr>
<td>Guild of Guardians Guilds (inc. 1 Mythic)</td>
<td>$130,400</td>
<td>71</td>
<td>$491</td>
</tr>
</tbody>
</table>
Gaming Guilds

Guilds are a form of internet-native holding company.

Guilds were one of the first institutions to popularize digital asset leases.
Virtual Real Estate Developers

Everyrealm: An internet-native **REIT** (real estate investment trust)

The idea:
- Raise money from LPs
  - => Buy up virtual land
  - => Develop structures and experiences people will pay for
  - => Distribute payments to LPs and take % of spread
  - => In future, sell developed property for capital gains
Develop Virtual Land?

What does this even mean? That can't require nearly as much funding as a real property development, right?

Challenge for everyone: turn a cube into a digital city.
Recent AAA title development costs:

- Cyberpunk 2077: $174M
- Battlefield 4: $100M
- Shadow of the Tomb Raider: $75M

Why?

- Graphical fidelity increasing -> development difficulty not decreasing
- Users want massively multiplayer and/or more immersive simulations
Virtual Land Development

A few of Everyrealm's developments

**METAJUKU**
A shopping district in Decentraland developed by Everyrealm. Inspired by Harajuku, a district in Tokyo known as the center of Japanese street fashion.

**THE BUILDING**

Metajuku is a 16,000 sq ft project built with a pedestrian-friendly open space at its center. Retail stores line both sides of the open center atrium. The district is located at the coordinates 94, 23 in Decentraland.

The development was designed by Austin-based architect Martin Quenz and developed by Everyrealm’s global team of real estate and game developers. The Band store was designed by Zagreb-based architecture firm BERO.

**FROSTBITTE VILLA**

Frostbitte is the most rare of all three villa types. A futuristic pairing of brutalism and serenity. Frostbitte is a stunning cliffside property that offers peaceful seclusion in the most beautiful yet icy setting. A bohemian, cantilevered above hangs dramatically from the rocky cliffs across the sea, attaching itself to your property by zipline. If jaw-dropping gatherings are your cup of cocoa, then this is the absolute perfect place to host one. Glass walls provide open, panoramic views out to your expanding empire in the metaverse, and the ice beach will wow even the most weathered explorers.
The NFT Cost Issue

NFTs are growing in popularity but remain expensive:

• Users who want to own NFTs but don't have the upfront capital are left with two options: save up or rent

• Starting an NFT business is impossible if you don't have cash on hand or access to LPs

• Expanding a business is difficult; the only method of fundraising is getting LPs
Credit Providers

Halliday: An internet-native credit provider

The idea:

Aggregate capital from LPs
=> Finance game assets over a fixed period
=> Give users immediate access to NFTs
=> Receive payments over time
=> Pass earnings back to LPs and take % cut
Credit Providers

Request details
Purchase NFT
Drago #13959

Access rights
Payments

Requests later payment for Drago #13959

Marketplace
Credit Providers

Drago 19358
Fully paid off

Paid: $131.93
Remaining: $0.00

Payment schedule
- Paid on Sep 30, 2022: $32.99
  Card Payment
- Paid on Sep 30, 2022: $32.98
  Card Payment
- Paid on Sep 30, 2022: $32.98
  Card Payment
- Paid on Sep 30, 2022: $32.98
  Card Payment

Withdraw Asset
Why is credit important?

- Credit puts excess capital to work, fueling growth
- Lowers the economic barrier to ownership
- Allows small businesses to start and expand

Credit is foundational to any modern economy
**Problem:** A REIT wants to create a digital mega-structure, but they don't have cash on hand to pay for it or enough staff to manage the properties.

What can they do?

- Collect cash they have for a down payment
- => Mortage the virtual property through a *credit provider*
- => Build the virtual mega-structure
- => Lease out subsections of the development to *guilds* who will sublet to individuals
- => Collect payments from *guilds* and other tenants
- => Use revenue to pay off the cost of the loan
- => Sell the development for capital gains
Where is this going?

This is only the beginning

We've covered only three services enabled by NFTs:

- Virtual holding companies
- Digital REITs
- Blockchain-native credit providers
Where is this going?

There is the rest of the stack to build:

• Intergame trade and commerce
• Unified identity protocols
• Virtual world regulatory entities and governing bodies
• Unbundling: Digital construction companies, virtual-only architecture firms, interior design firms, so much more
• Bundling: Game developer and interoperability layer
Where is this going?

A thought experiment:

Industry + Atoms = NYC
Where is this going?
Next lecture: Decentralized Exchanges (DeX)