CRIX the Coin

Kevin Nößler

Thomas Herrdum

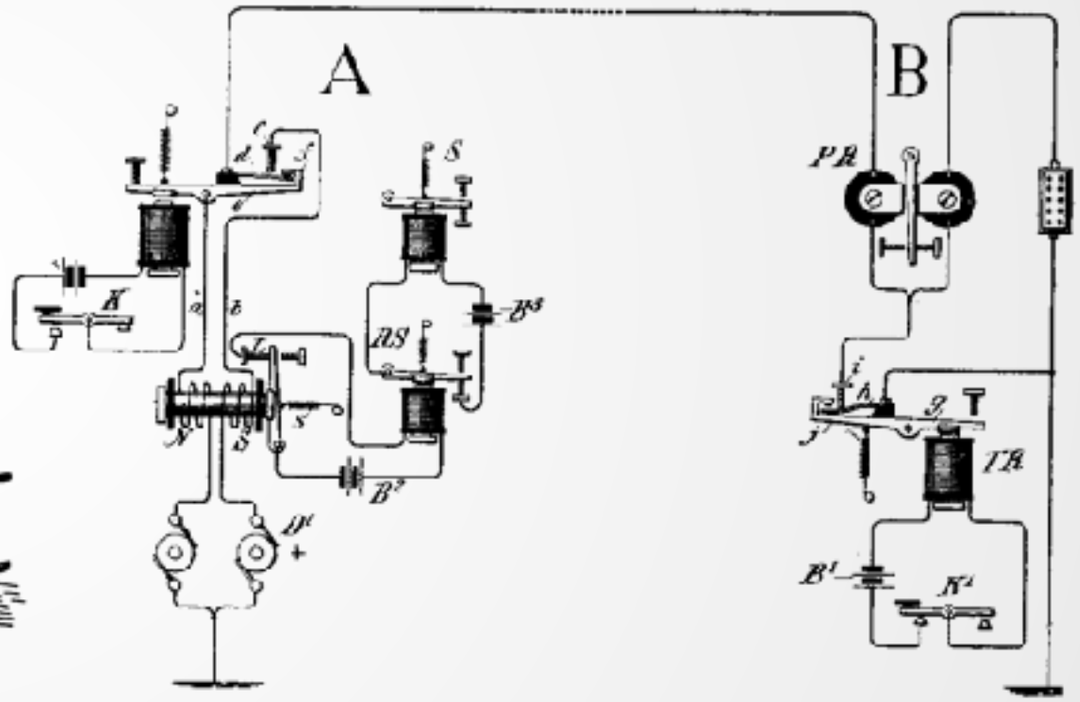
Wolfgang Karl Härdle

Ladislaus von Bortkiewicz Professor of Statistics
Humboldt-Universität zu Berlin
lvb.wiwi.hu-berlin.de

The role of money

- Unit of account
- Store of value
- Medium of exchange





Greek Coin, Gutenberg Paper Printing, Telegraphy Circuit

Allocation of scarce (appropriable, non-abundant, non-rivalrous) resources

Commodity (Gold) - Fiat (EUR) - Cryptocurrencies (CC)

Cryptocurrencies

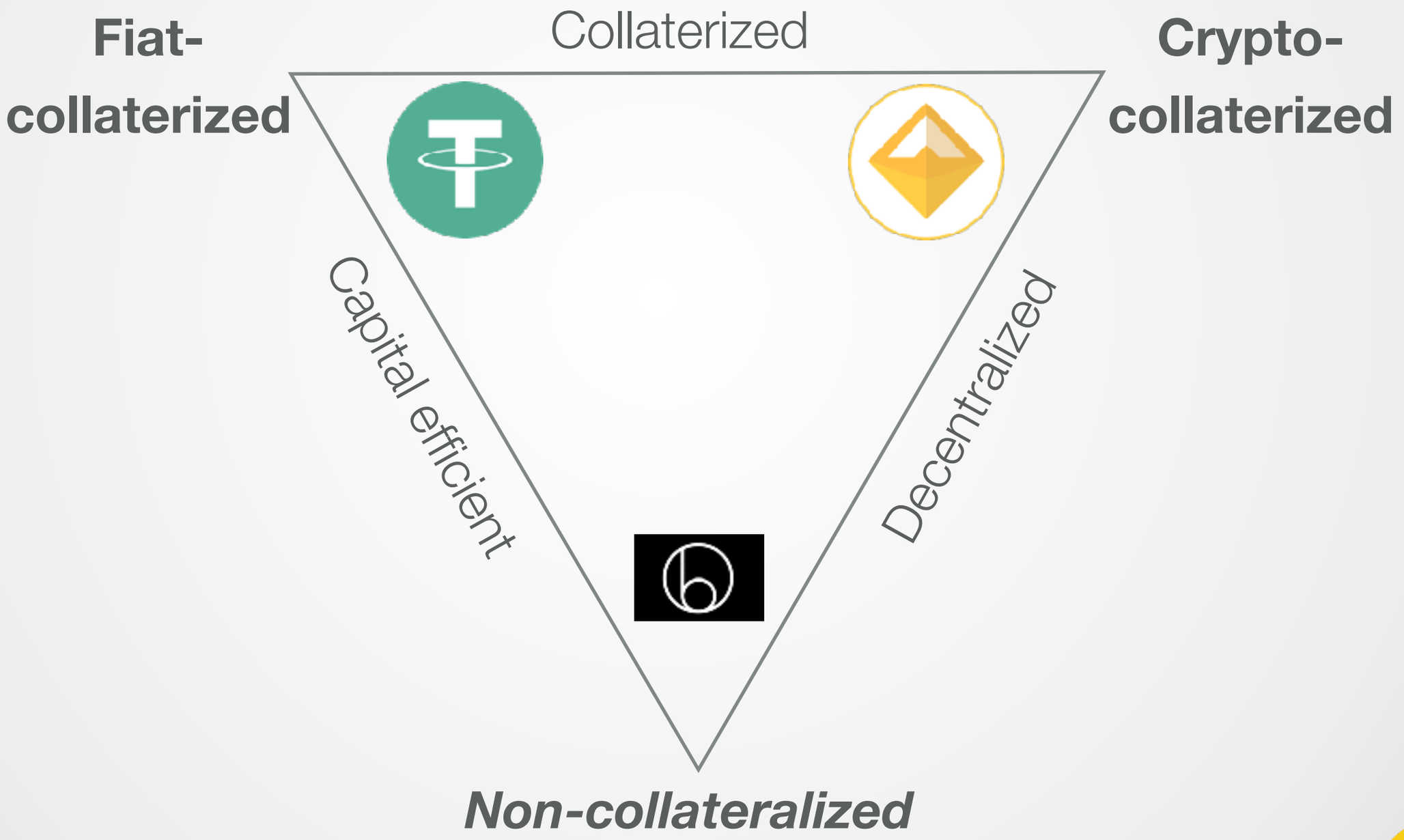
Drawbacks of cryptocurrencies:

- Unnecessary currency risk
- Difficult to transact
- Volatility
- Mainstream Adoption

Goal: Create a price stable cryptocurrency

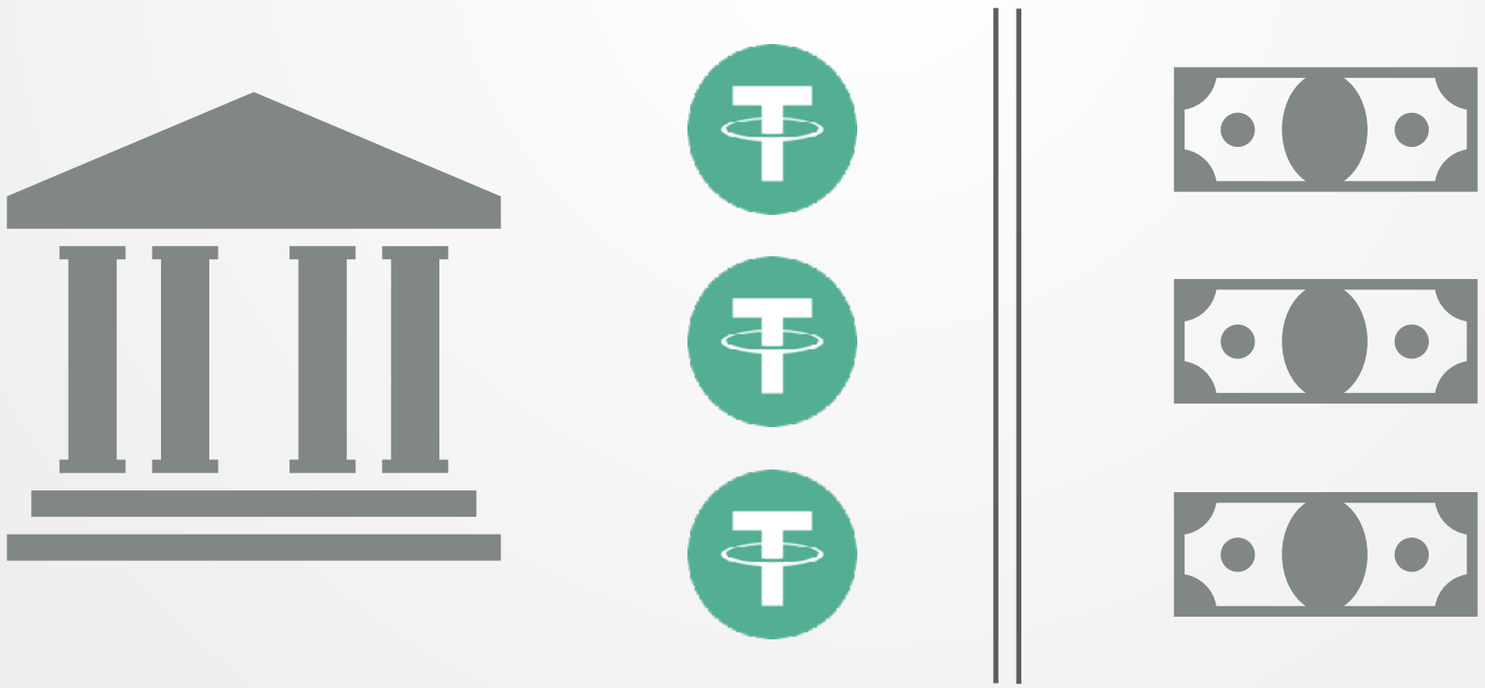
Solution: Pegg the value of a stablecoin to real world assets

Types of stablecoins:



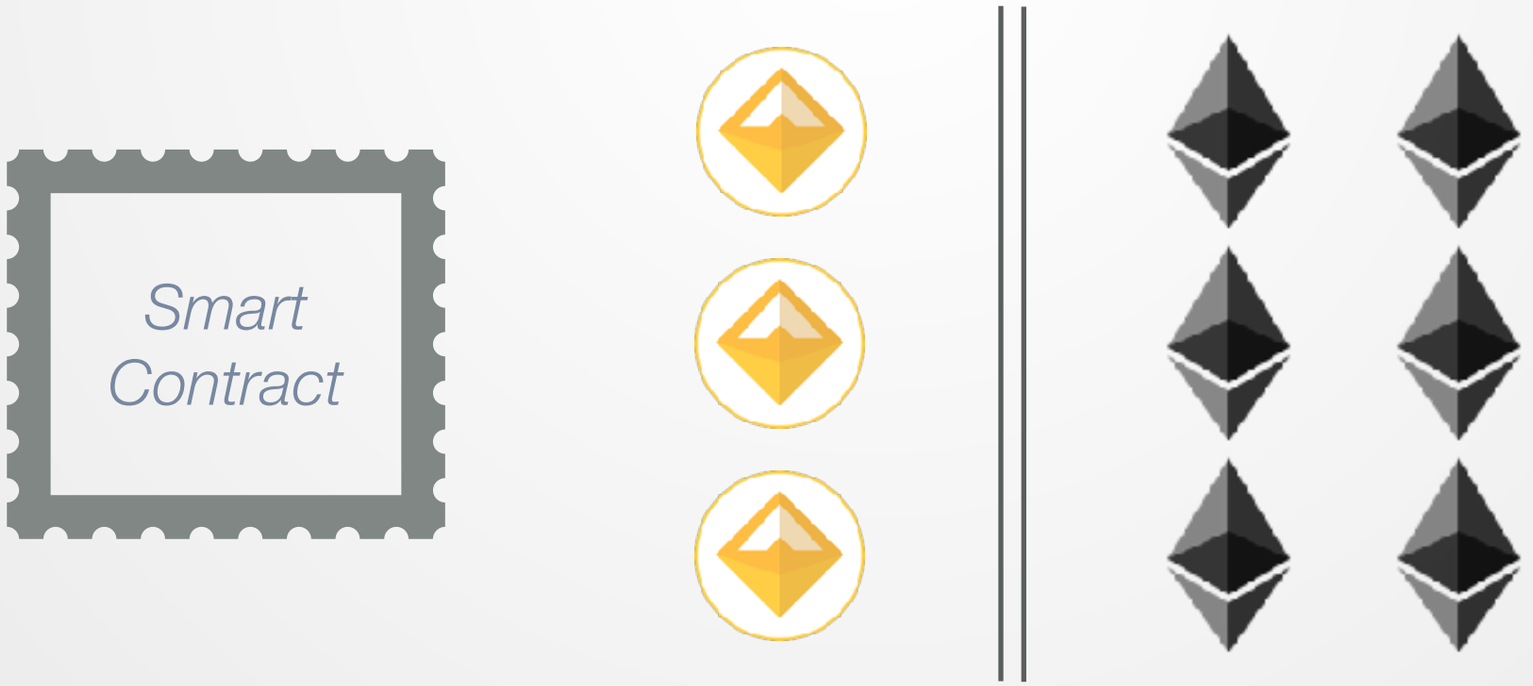
Fiat-collaterized

- Simple
- Value remains the same (price-stable, close to *100%*)
- Less vulnerable to hacks, since no collateral is held on the blockchain
- Digital representation of a currency:

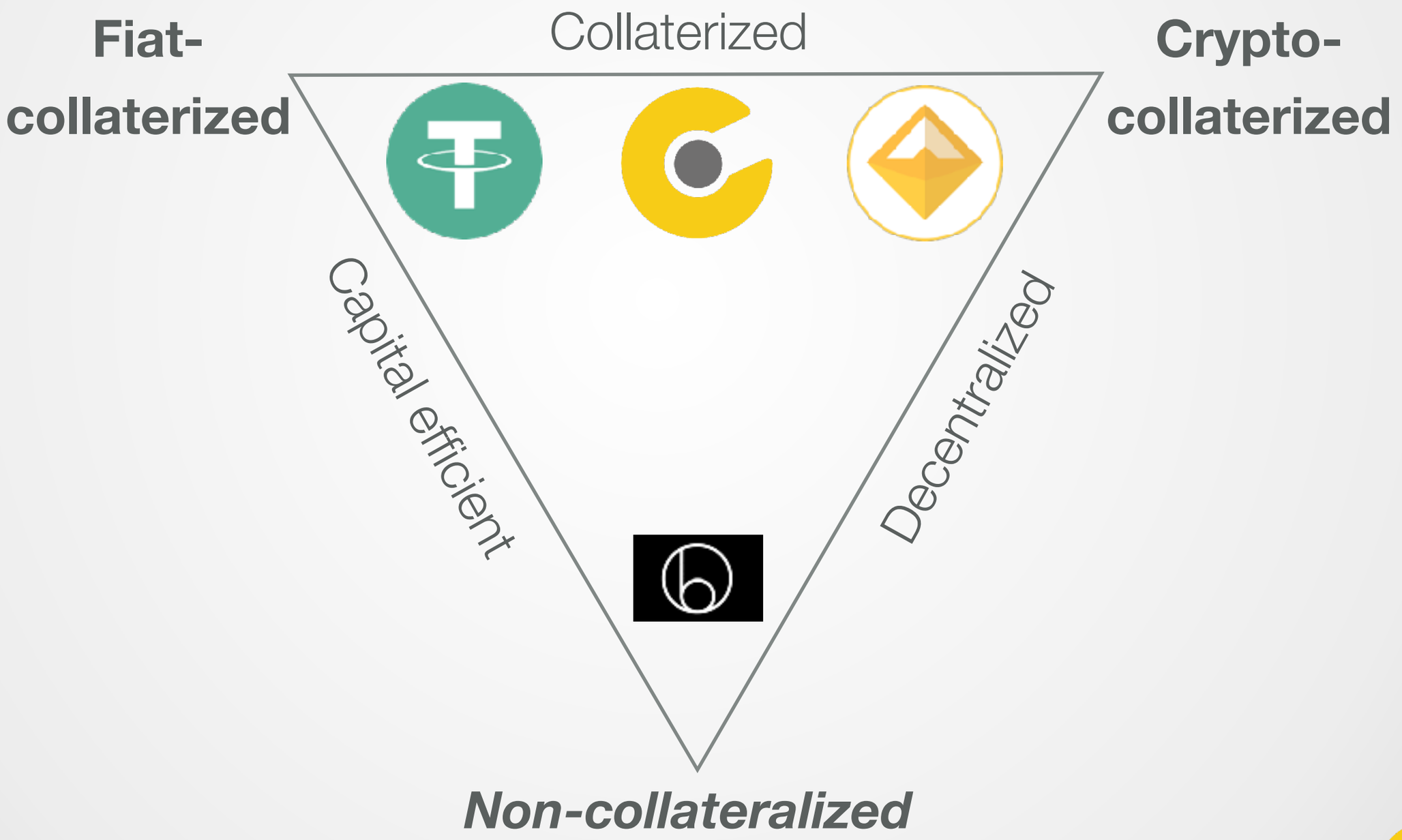


Crypto-collaterized

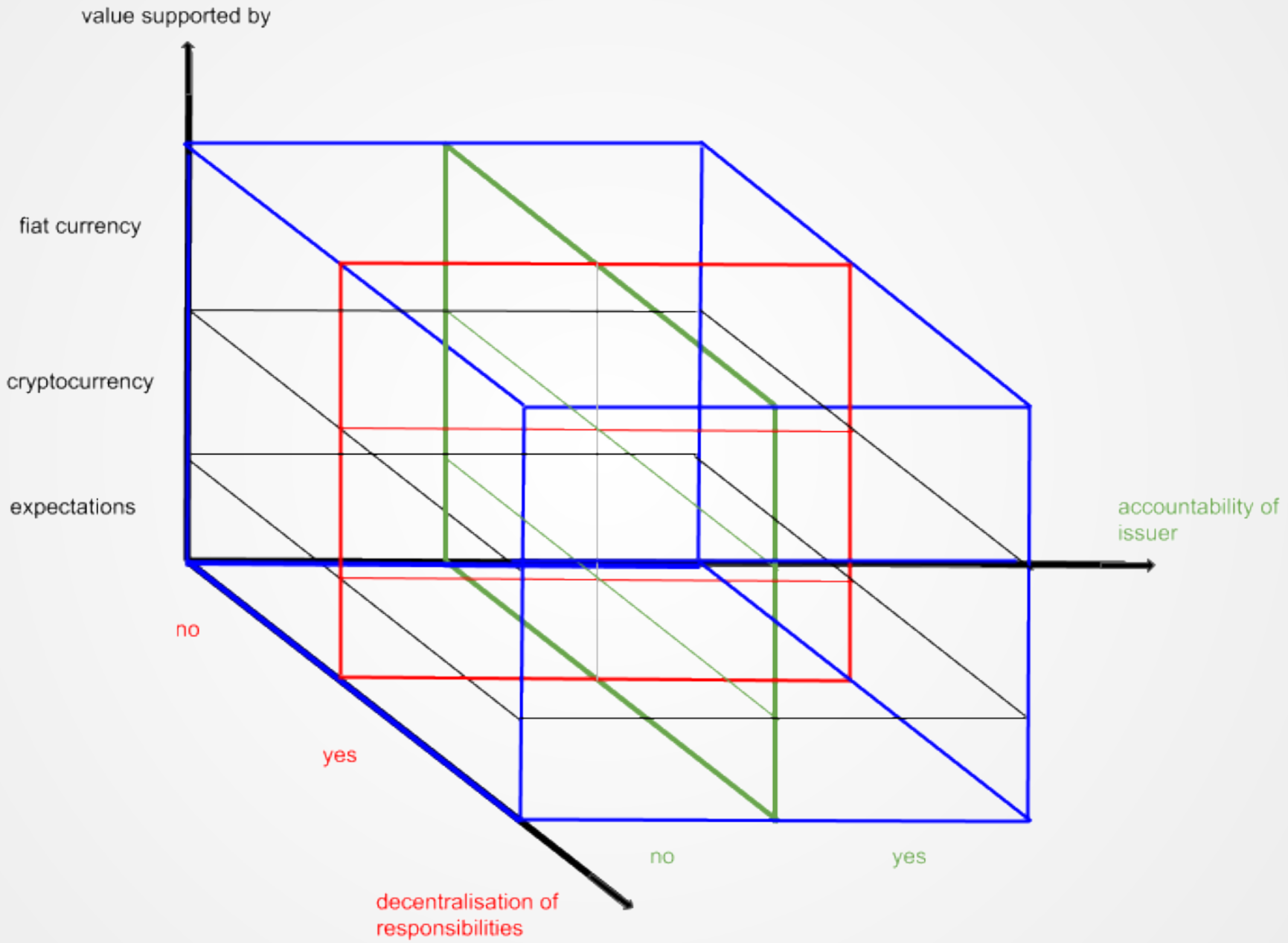
- Liquidate quickly into underlying crypto collateral
- More decentralised
- Transparency
- Collaterization Ratio:



Types of stablecoins:

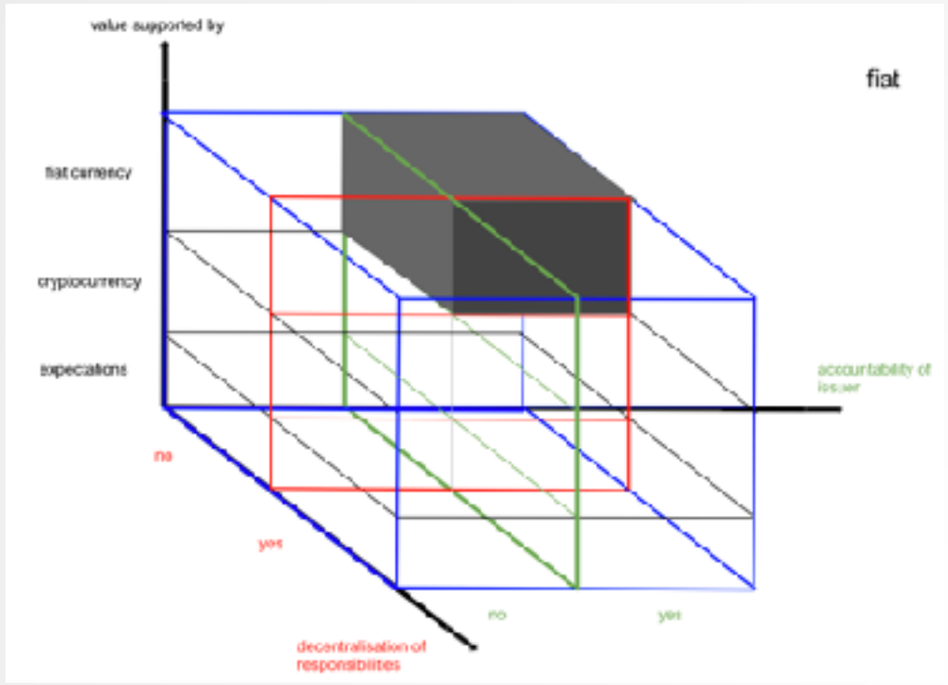


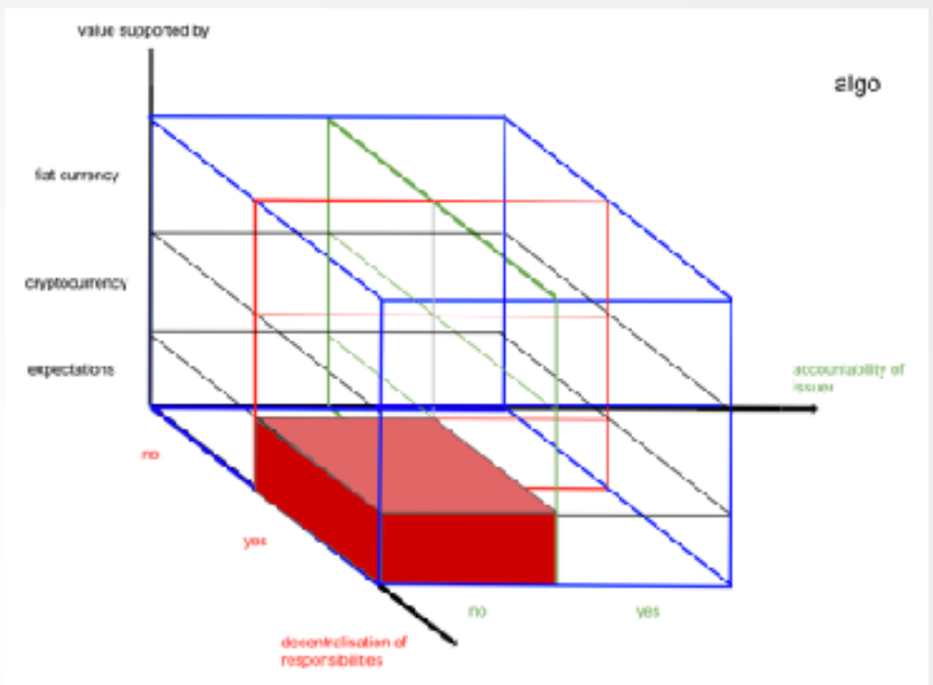
Types of stablecoins - ‘The crypto cube’

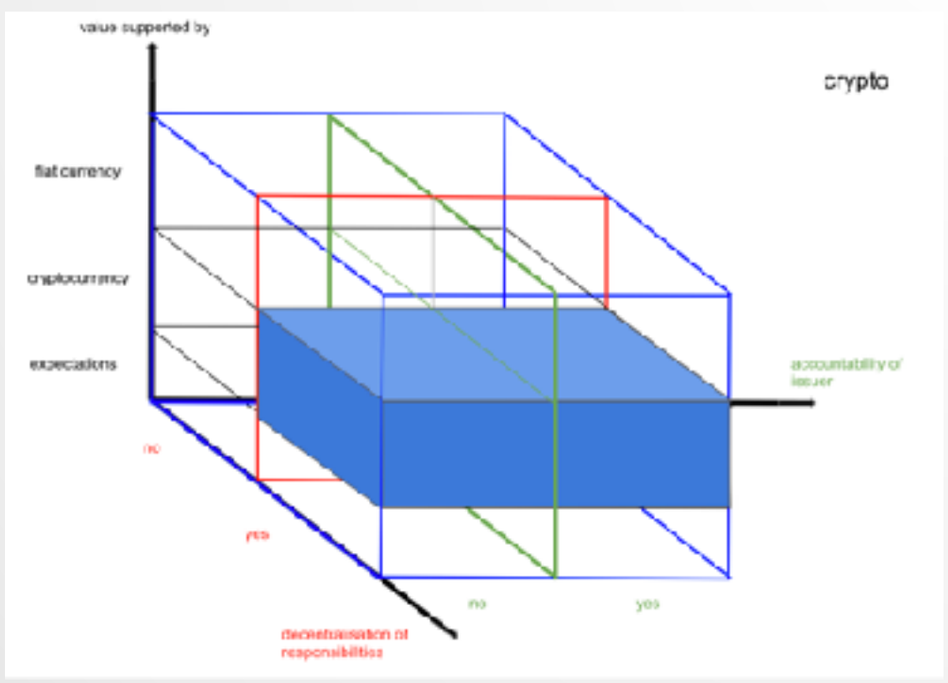


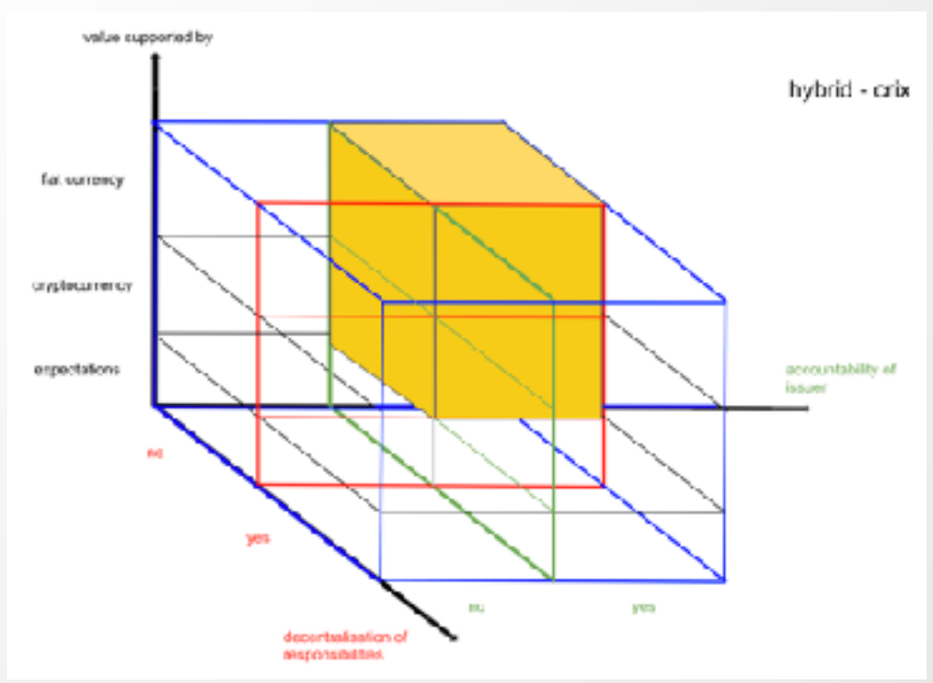
The Crypto Cube.
Adapted from 'In search for stability in crypto-assets: are stablecoins the solution?,' by Dirk Bullmann, Jonas Klemm, Andrea Pinna. 2019

Types of stablecoins - ‘The crypto cube’









Outline

1. Motivation
2. Smart Contract
   - Solidity/Remix
3. PRI-Coin rating system
   - Environment (How to get started)
   - Set-up/Transfer
   - iOS/Android App
4. Next Steps
   - White Paper (Master thesis)
   - ICO/Deployment
   - Pegging to CRIX constituents

Smart Contract

Solidity:

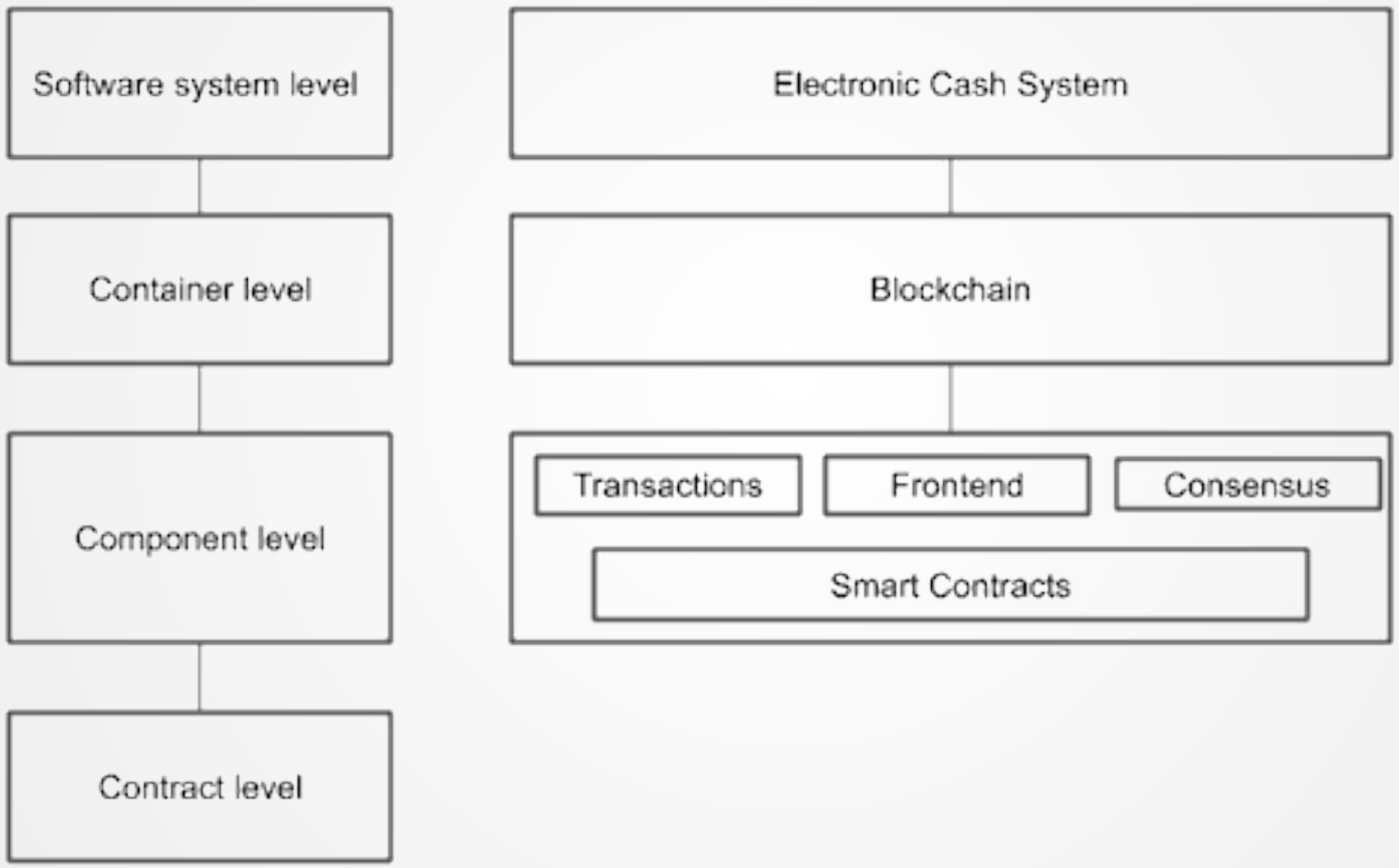
- Object-oriented, application-specific high-level programming language
- JavaScript-like syntax for developing Smart Contracts for blockchain platforms
    1. Ethereum Virtual Machine (EVM)

Remix:

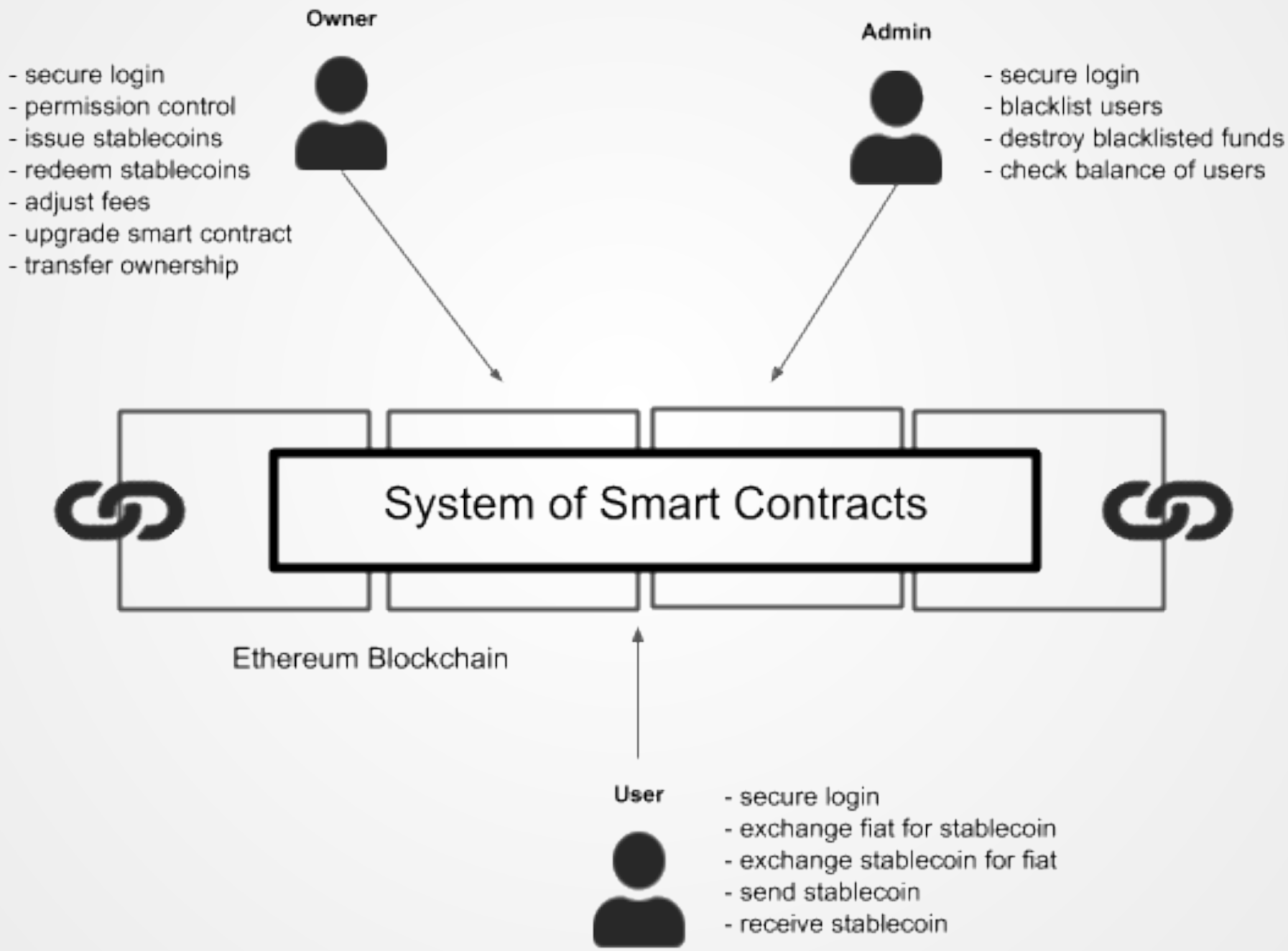
- Powerful, open source tool to write Smart Contracts
- Supports testing, debugging and deploying of smart contracts

https://remix.ethereum.org/

High-level-system overview



Overview of users



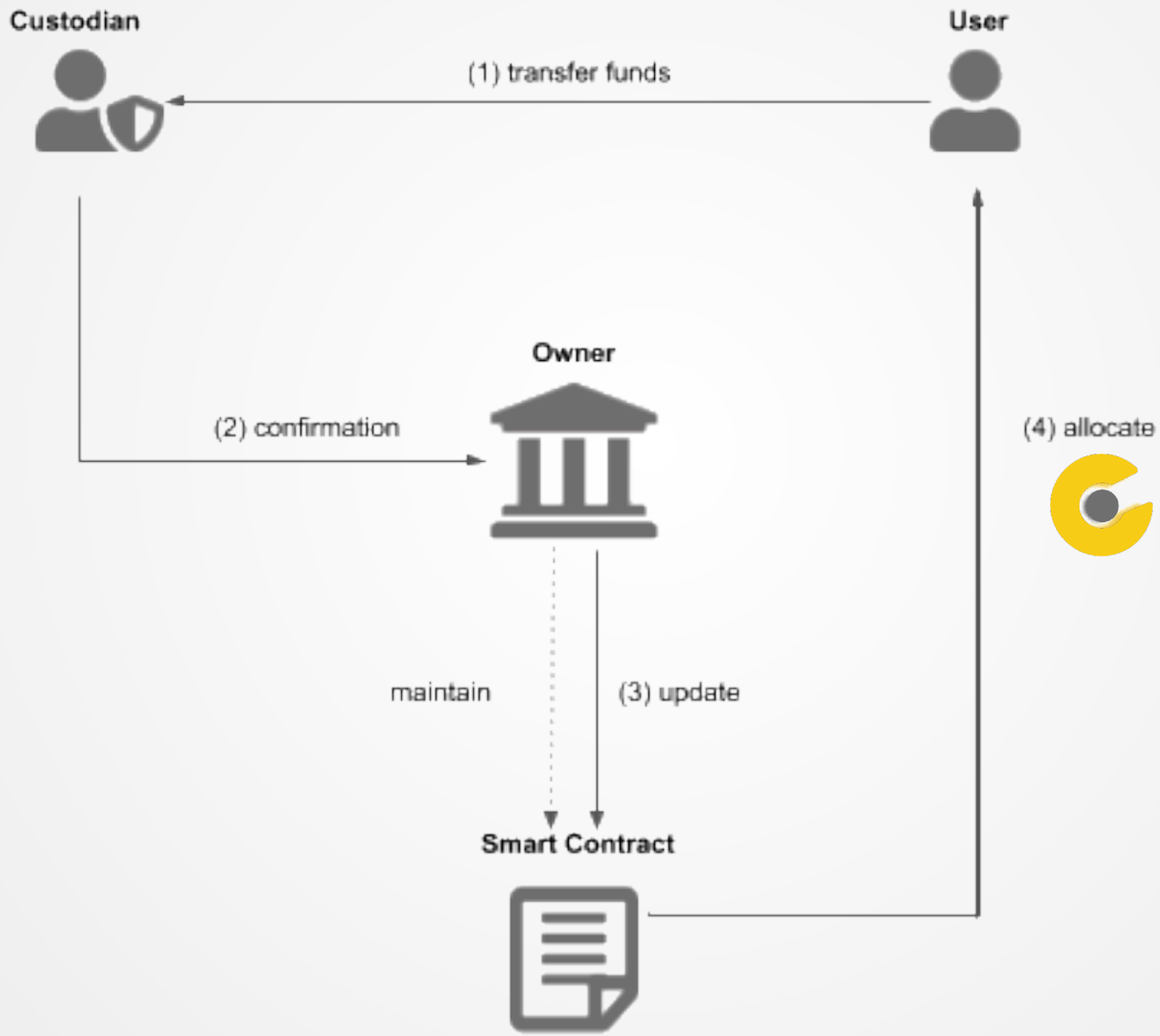
Smart Contract

- contract Ownable:
  - contract has an owner address, and provides basic authorisation control -> "user permissions”
  - contract has multiple admin addresses (added)
- library SafeMath:
  - Math operations with safety checks that throw on error.
- Functions:
  - transfer: Transfer token for a specified address.
  - balanceOf: Gets the balance of the specified address.
  - approve: Approve the passed address to spend the specified amount of tokens on behalf of msg.sender.
  - allowance: Check the amount of tokens that an owner allowed to a spender.

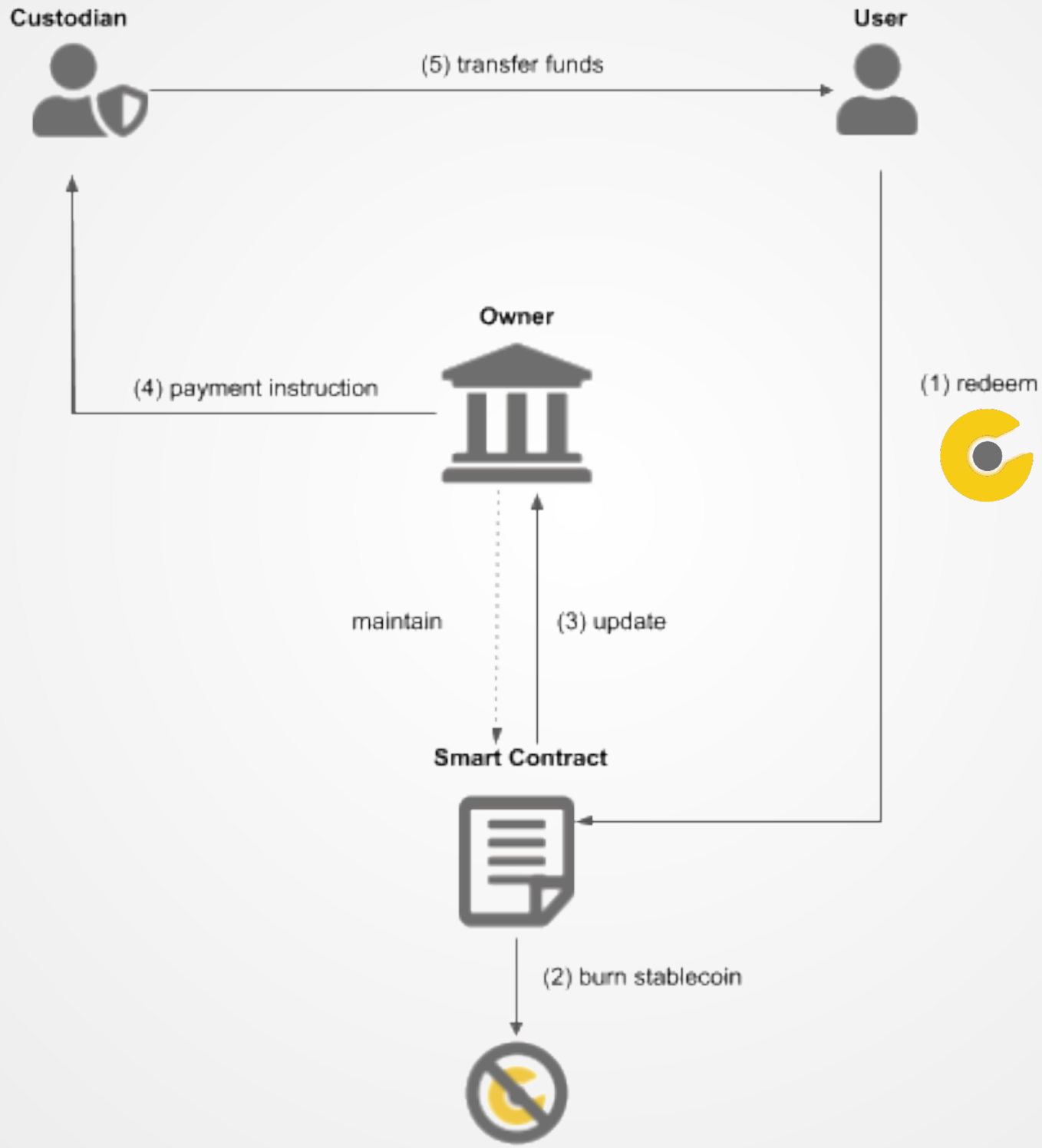
Smart Contract

- Functions (conti.):
  - issue: Issue a new amount of tokens to owner address.
  - redeem: Tokens are withdrawn from the owner address.
  - pause/unpause: Called by the owner to pause/unpause, triggers stopped/unstopped state.
  - addBlackList; removeBlackList; destroyBlackFunds.
  - upgrade: deprecate current contract and upgrade for a new contract.
  - setParams: Add fees to the contract!
- Events (e.g. Pausable, Issue, Redeem, DestroyedBlackFunds, Upgrade)
- ERC20 Token Standard

Contract level: Issuance



Contract level: Redemption

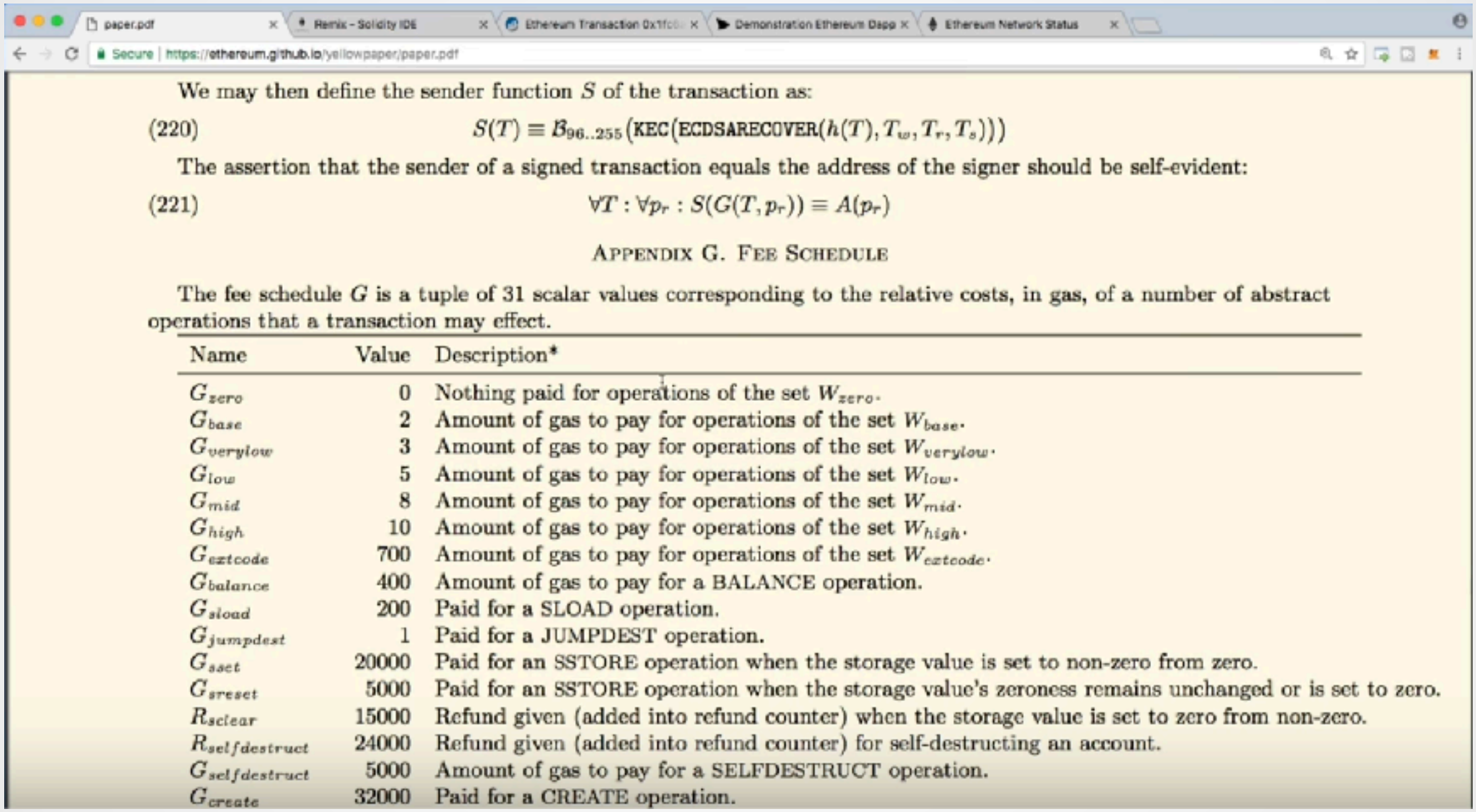


Contract level: Transfer

Ethereum Gas

- Each time a solidity contract is compiled it is converted into a sequence of **op**eration **codes** (e.g. ADD for addition)
- **Byte codes** are **similar** to **opcodes** but they are represented by hexadecimal numbers
- The Ethereum Virtual Machine (EVM - Remix) executes the byte code
- Each opcode has a fixed amount of gas assigned and is a measure of computational effort (e.g. ADD costs 3 gas, SHA256 costs 60 gas)
- Gas is the **execution fee**, paid by the **sender** of the transaction **who triggered the computation**
- Minimum gas limit = 21000 gas
- Interacting with a smart contract = 21000 gas + all executed opcodes gas
- The gas price is not fixed; It correlates positively to the current network traffic
- Current gas price: https://ethstats.net/ / https://www.rinkeby.io/#stats

Smart Contract



paper.pdf | Remix - Solidity IDE | Ethereum Transaction 0x1fc6... | Demonstration Ethereum Dapp | Ethereum Network Status

Secure | https://ethereum.github.io/yellowpaper/paper.pdf

We may then define the sender function S of the transaction as:

$$S(T) \equiv \mathcal{B}_{96..255}\left(\text{KEC}\left(\text{ECDSARECOVER}(h(T), T_w, T_r, T_s)\right)\right) \tag{220}$$

The assertion that the sender of a signed transaction equals the address of the signer should be self-evident:

$$\forall T : \forall p_r : S(G(T, p_r)) \equiv A(p_r) \tag{221}$$

Appendix G. Fee Schedule

The fee schedule G is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

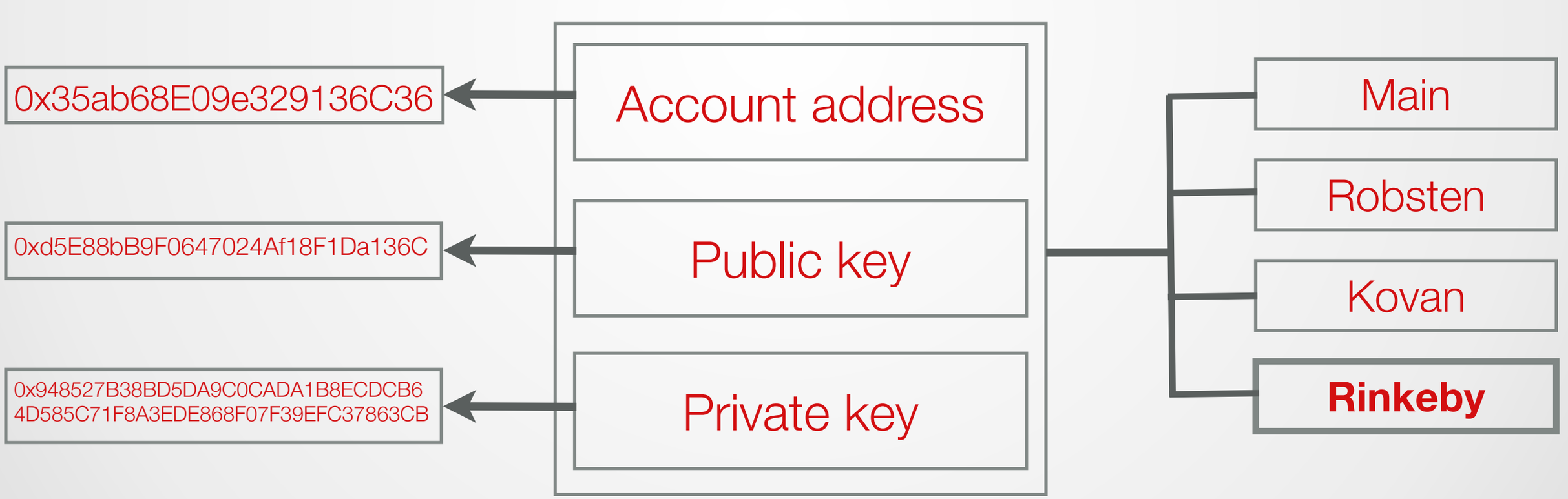
Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid}.
G_{high}	10	Amount of gas to pay for operations of the set W_{high}.
$G_{extcode}$	700	Amount of gas to pay for operations of the set $W_{extcode}$.
$G_{balance}$	400	Amount of gas to pay for a BALANCE operation.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
G_{sreset}	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{selfdestruct}$	24000	Refund given (added into refund counter) for self-destructing an account.
$G_{selfdestruct}$	5000	Amount of gas to pay for a SELFDESTRUCT operation.
G_{create}	32000	Paid for a CREATE operation.

Ethereum Gas

- When executing a transaction, maximum amount of gas can be specified
- If a transaction exceeds the gas price, changes are reversed but user will still have to pay specified amount of gas
- If the actual gas price is lower than the specified maximum amount of gas, the user will only pay for the actual used gas
- The gas limit is a security measure against circularity (bug in the smart contract) which can potentially result in unlimited amounts of gas
- Miners prefer transactions with a higher fee (merit order), transactions with a lower fee will take longer to be included in a block

PRI-coin - Create your own wallet

- Go to temporary chatroom: https://yopad.eu/p/PRI_Coin-365days
- Meta Mask is a wallet and thus the portal to the ethereum network
- Via Meta Mask we can send and receive PRI-coins
- What does it mean to have an "account"?



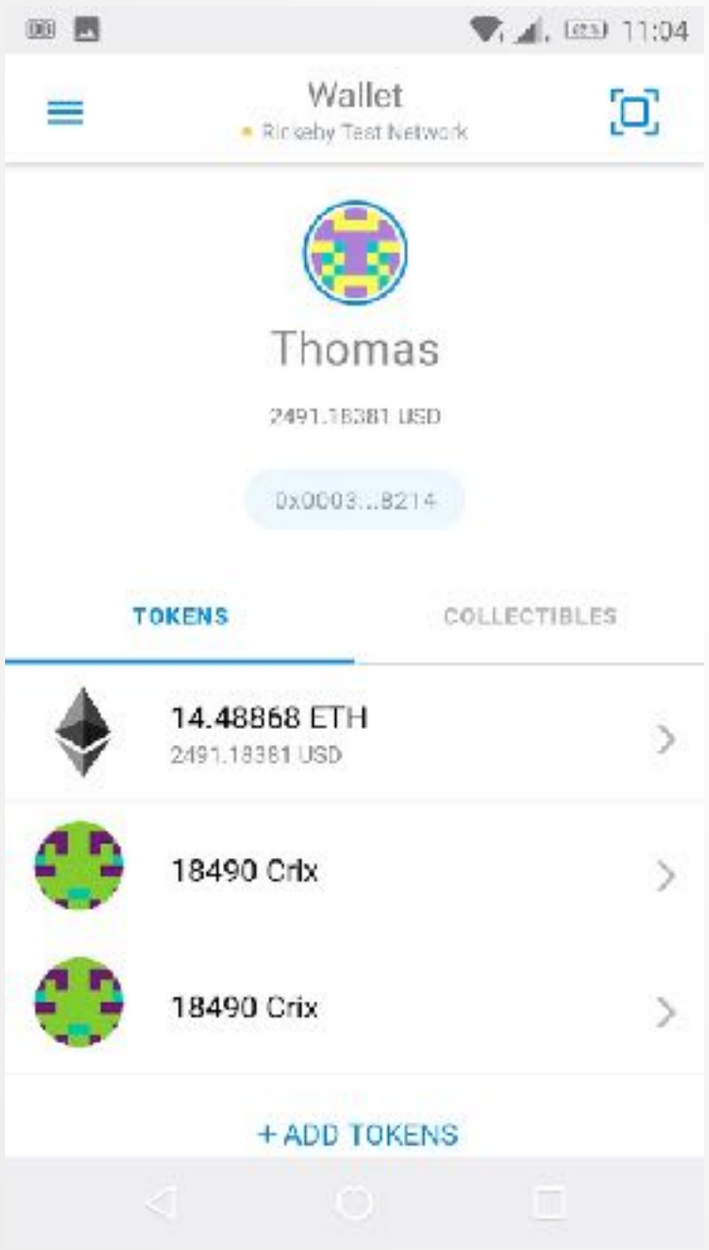
Create your own PRI-coin wallet

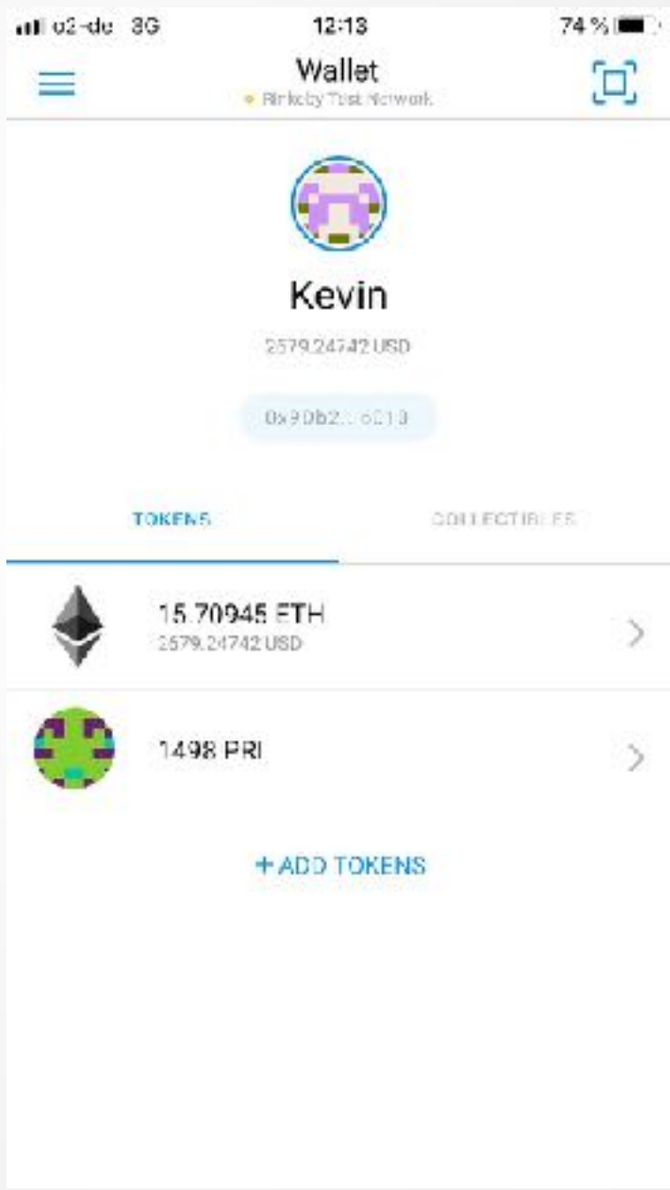
- Link to download Google Chrome Browser:
- Link to download Chrome extension 'Meta Mask':
- Go to the chatroom: https://yopad.eu/p/PRI_Coin-365days and follow the instructions.
- Contract address:

  0x0D61b2583369F98487dd6414f10FE0A35694B8c3

iOS & Android Application

- Now available on Google Play Store and Appstore (Beta):





Next Steps

- White Paper = Master thesis
- Check legal implications
- Link/Tab on the CRIX website (https://thecrix.de/)
- Further security checks (potentially external sources to raise credibility for the project)
- Plan ICO
- Get on an exchange (e.g. Coingecko, Binance, Kraken, etc.)
- Find Investors

Pegging to the CRIX constituents

TheCRIX.de

- **CR**yptocurrency **I**nde**X** / **CRIX**
- CRIX a benchmark index for crypto dynamics

- CRIX as a valuable trading tool providing a benchmark for CCs
- Used by Exchange Traded Funds and Investment funds

crix.berlin, **crix.info**