

CMTA AGM – Keynote

JP Aumasson – CMTA, technical committee chair – Taurus, co-founder & chief security officer

Keynote speeches are like golf,
it's what you do when you retire.

— Halvar Flake

CMTA

CMTAT

CMTAToken

CMTATokenization

Token

what is a token?

In the context of computers and digital technology, a "token" can refer to several things, but it's often used in two main ways:

1. **Authentication Token:** In computer security, a token is an object that the owner has to authenticate their identity. This can be in physical form (like a security badge or card), or it can be digital. For instance, in digital form, it could be a piece of data generated by a website that a user's computer stores and presents to the site to confirm their identity. This is how you stay logged into a website even after you close the browser.
2. **Cryptocurrency or Blockchain Token:** In the realm of cryptocurrency and blockchain technology, a token represents a unit of value issued by a private entity. Cryptocurrencies like Bitcoin and Ethereum are tokens. There are also tokens issued on top of blockchain networks, like the various tokens issued on the Ethereum network using the ERC-20 and ERC-721 standards (the latter being the basis for "Non-Fungible Tokens" or NFTs, which represent unique assets).

how can I create my blockchain token on Ethereum?

Creating a blockchain token on Ethereum involves writing and deploying a smart contract to the Ethereum network. Here is a basic guide on how you might do it. Please note, however, that creating a token should only be done if you have a good understanding of how Ethereum and smart contracts work. Mistakes in this process can lead to loss of funds or other serious issues. The guide assumes a basic understanding of Ethereum, smart contracts, and the Solidity programming language.

1. **Install MetaMask:** MetaMask is a browser extension that allows you to interact with the Ethereum network directly from your browser. It also acts as an Ethereum wallet.
2. **Get Ether:** To deploy your contract on the Ethereum network, you'll need Ether to pay for the gas fees. You can get Ether from a cryptocurrency exchange.
3. **Write Your Smart Contract:** In order to create a token, you'll need to write a smart contract that conforms to the ERC20 or ERC721 standards (for fungible and non-fungible tokens respectively). The OpenZeppelin library provides secure and community-reviewed implementations of these standards that you can use.

A smart contract is..

A smart contract is..

Not a contract

A smart contract is..

Not a contract create by AI

A smart contract is..

Just like a normal computer program,
except that multiple computers run it
instead of one computer

A smart contract is..

Just like a normal computer program,
except that multiple computers run it
instead of one computer

$$\begin{array}{l} \mathbf{x} = 0 \\ \mathbf{y} = 10 \end{array}$$

$$\begin{array}{l} \mathbf{x} = \mathbf{x} + 2 \\ \mathbf{y} = \mathbf{y} - 2 \end{array}$$



$$\begin{aligned} \mathbf{x} &= 0 \\ \mathbf{y} &= 10 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2 \end{aligned}$$



$$\begin{aligned} \mathbf{x} &= 0 \\ \mathbf{y} &= 10 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2 \end{aligned}$$



$$\begin{aligned} \mathbf{x} &= 0 \\ \mathbf{y} &= 10 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2 \end{aligned}$$



$$\begin{aligned} \mathbf{x} &= 0 \\ \mathbf{y} &= 10 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2 \end{aligned}$$



Hey other computers, do you
also get $x=2$ and $y=8$?

$$\begin{aligned}x &= 0 \\ y &= 10\end{aligned}$$

$$\begin{aligned}x &= 0 \\ y &= 10\end{aligned}$$

$$\begin{aligned}x &= 0 \\ y &= 10\end{aligned}$$

$$\begin{aligned}x &= x + 2 \\ y &= y - 2\end{aligned}$$

$$\begin{aligned}x &= x + 2 \\ y &= y - 2\end{aligned}$$

$$\begin{aligned}x &= x + 2 \\ y &= y - 2\end{aligned}$$



Yup



Me too

$$\begin{aligned} \mathbf{x} &= 0 \\ \mathbf{y} &= 10 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= 0 \\ \mathbf{y} &= 10 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= 0 \\ \mathbf{y} &= 10 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2 \end{aligned}$$



$$\mathbf{x} = 0$$
$$\mathbf{y} = 10$$

$$\mathbf{x} = \mathbf{x} + 2$$
$$\mathbf{y} = \mathbf{y} - 2$$

We agree that now
 $\mathbf{x}=2$ and $\mathbf{y}=8$



$$\mathbf{x} = 0$$
$$\mathbf{y} = 10$$

$$\mathbf{x} = \mathbf{x} + 2$$
$$\mathbf{y} = \mathbf{y} - 2$$

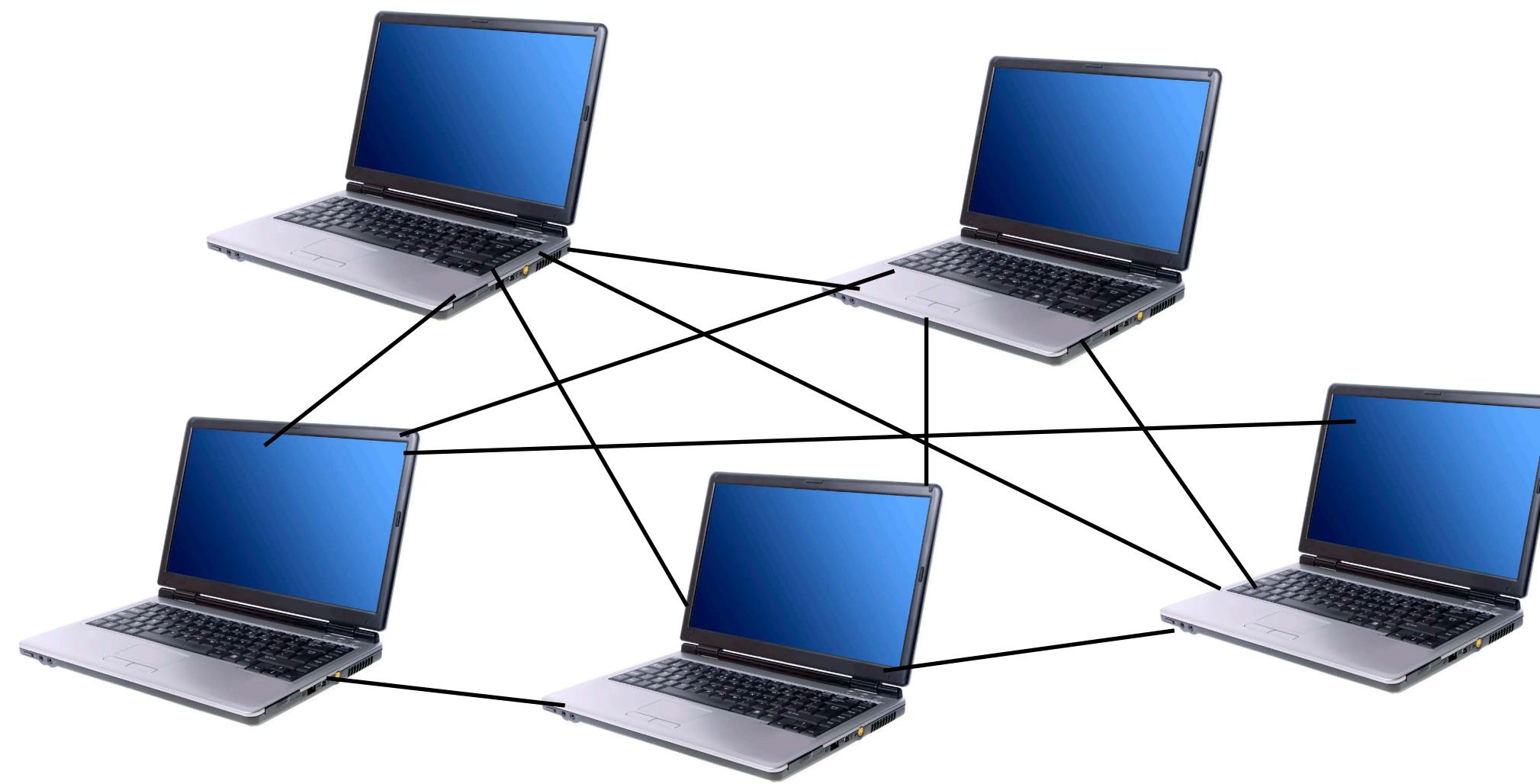
We agree that now
 $\mathbf{x}=2$ and $\mathbf{y}=8$



$$\mathbf{x} = 0$$
$$\mathbf{y} = 10$$

$$\mathbf{x} = \mathbf{x} + 2$$
$$\mathbf{y} = \mathbf{y} - 2$$

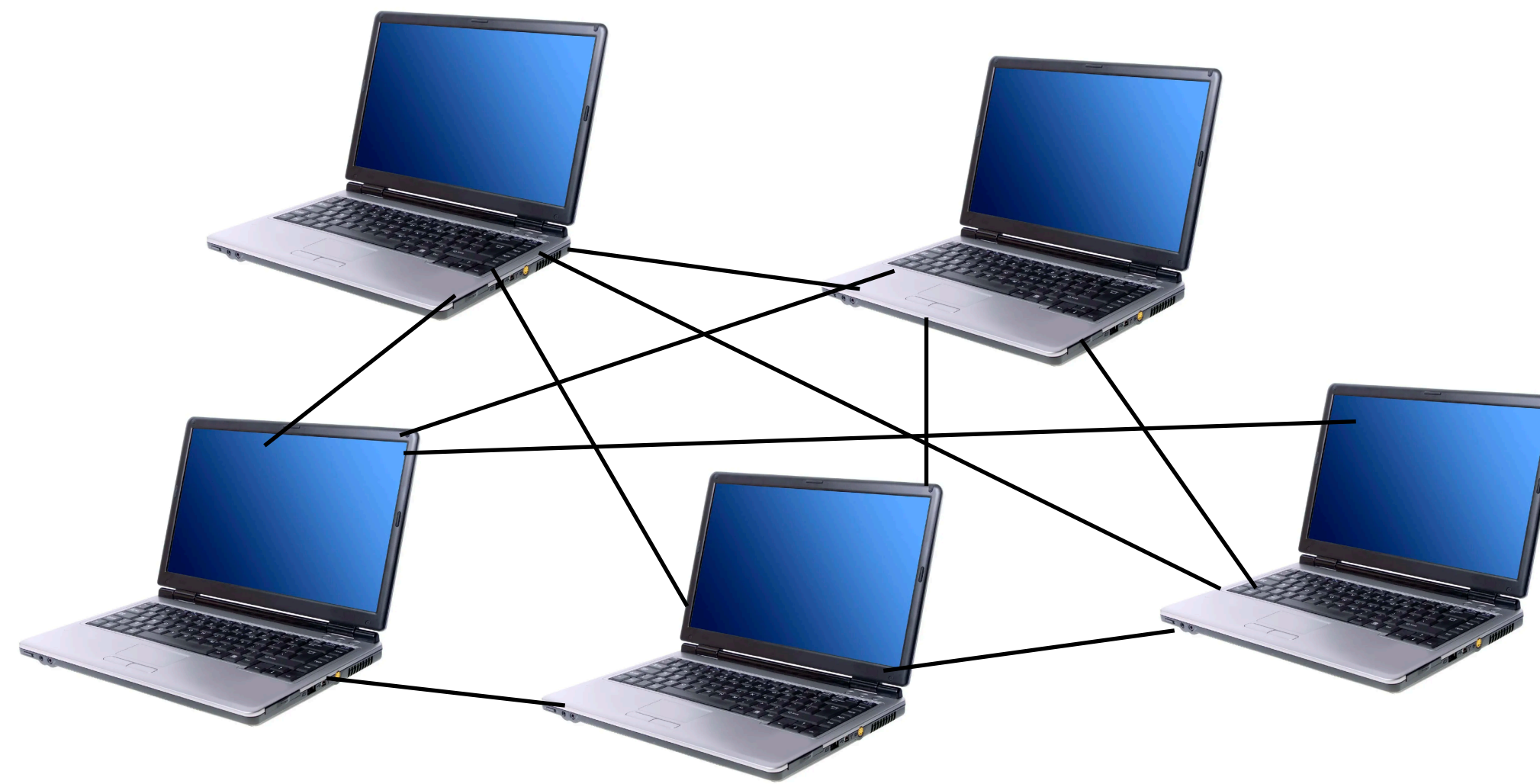
We agree that now
 $\mathbf{x}=2$ and $\mathbf{y}=8$



$$\begin{aligned} \mathbf{x} &= 0 \\ \mathbf{y} &= 10 \end{aligned}$$

$$\begin{aligned} \mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2 \end{aligned}$$

We agree that now
 $\mathbf{x}=2$ and $\mathbf{y}=8$



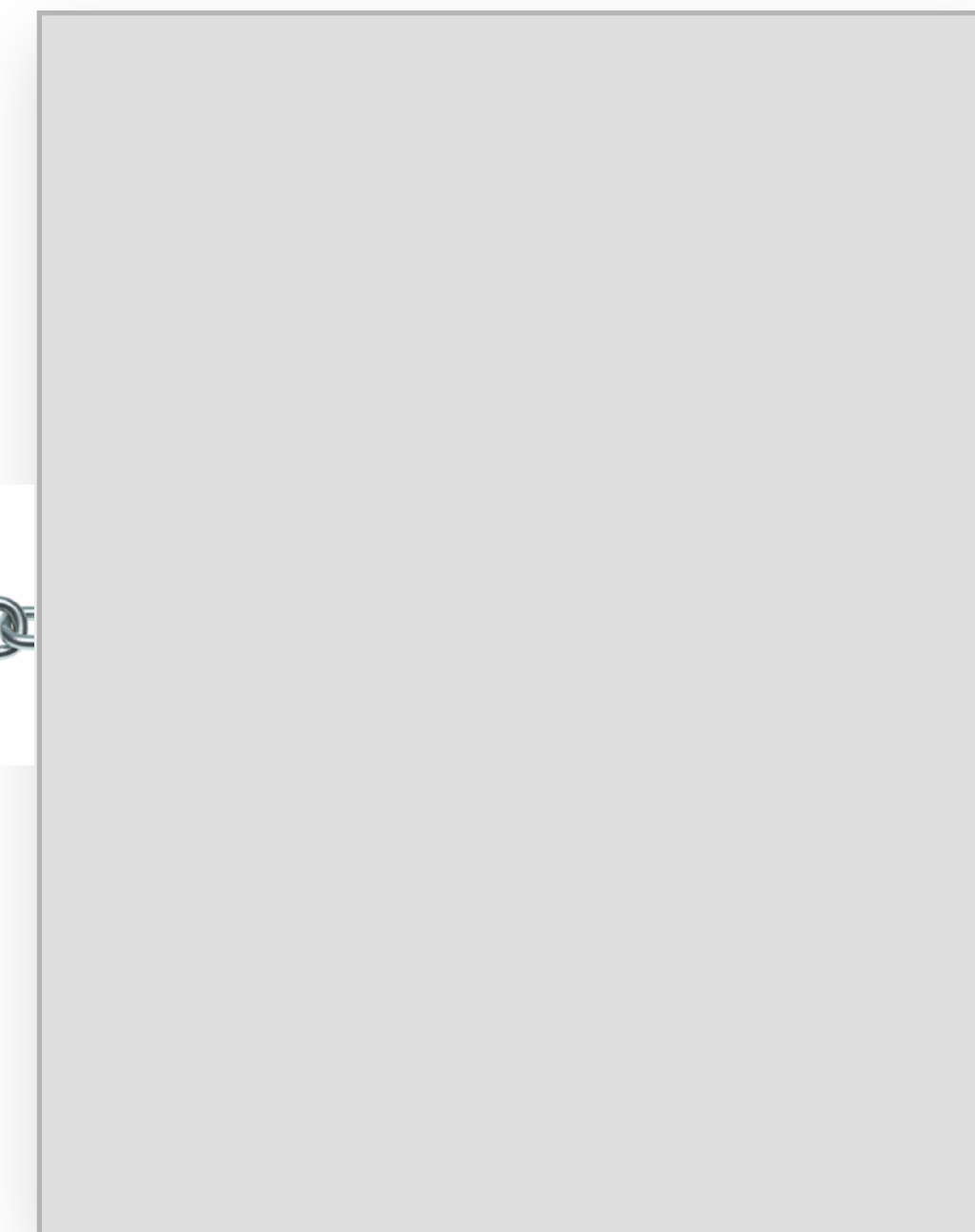
$$\begin{aligned}\mathbf{x} &= 0 \\ \mathbf{y} &= 10\end{aligned}$$

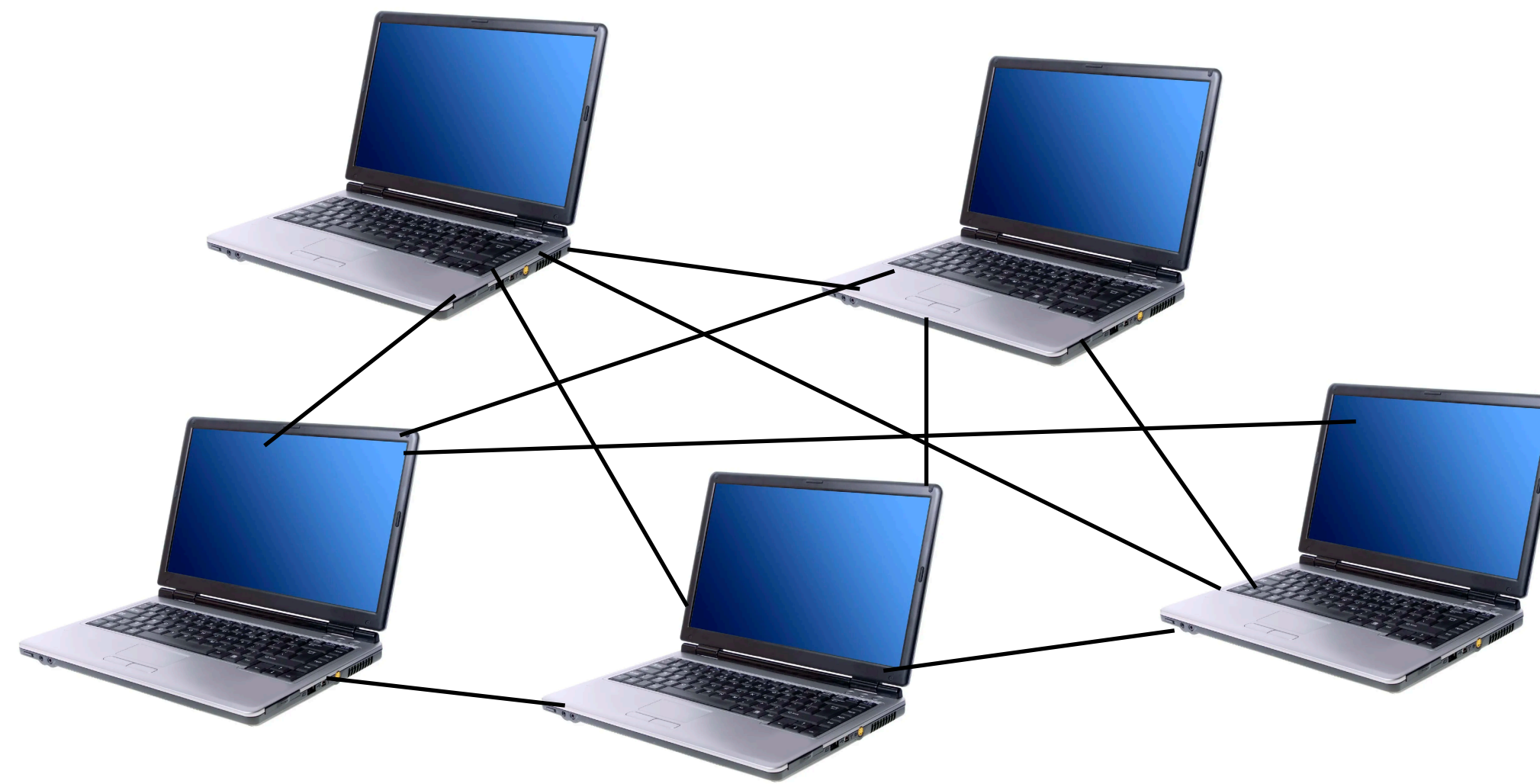
$$\begin{aligned}\mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2\end{aligned}$$

We agree that
 $\mathbf{x}=2$ and $\mathbf{y}=8$

$$\begin{aligned}\mathbf{x} &= \mathbf{x} + 4 \\ \mathbf{y} &= \mathbf{y} - 4\end{aligned}$$

We agree that
 $\mathbf{x}=6$ and $\mathbf{y}=4$





$$\begin{aligned}\mathbf{x} &= 0 \\ \mathbf{y} &= 10\end{aligned}$$

$$\begin{aligned}\mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2\end{aligned}$$

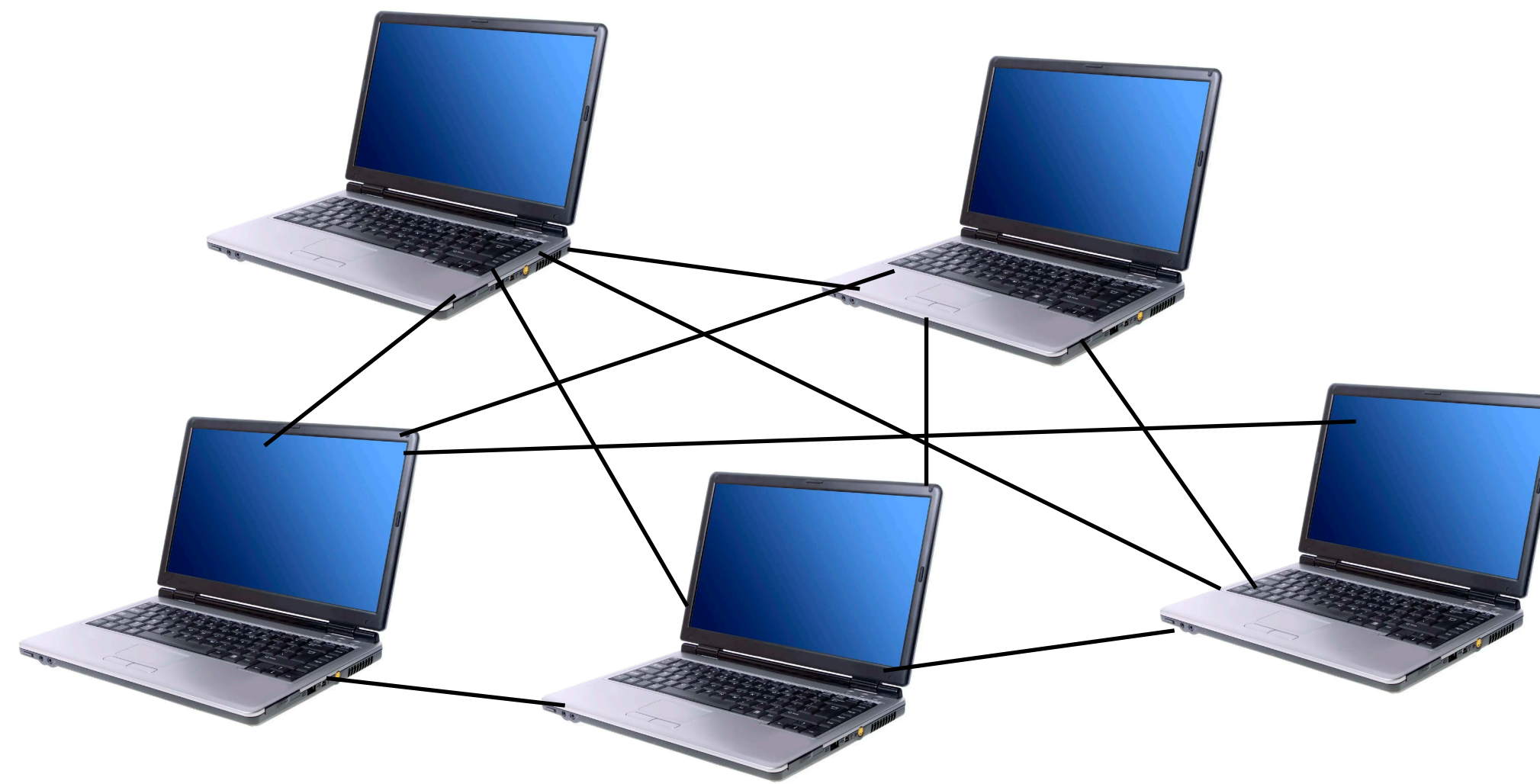
We agree that
 $\mathbf{x}=2$ and $\mathbf{y}=8$

$$\begin{aligned}\mathbf{x} &= \mathbf{x} + 4 \\ \mathbf{y} &= \mathbf{y} - 4\end{aligned}$$

We agree that
 $\mathbf{x}=6$ and $\mathbf{y}=4$

$$\begin{aligned}\mathbf{x} &= \mathbf{x} - 5 \\ \mathbf{y} &= \mathbf{y} + 5\end{aligned}$$

We agree that
 $\mathbf{x}=1$ and $\mathbf{y}=9$



"Blockchain"

=

Consensus protocol

(Rules to agree on things)

+

Database of transactions

(List of programs and data)

$$\begin{aligned}\mathbf{x} &= 0 \\ \mathbf{y} &= 10\end{aligned}$$

$$\begin{aligned}\mathbf{x} &= \mathbf{x} + 2 \\ \mathbf{y} &= \mathbf{y} - 2\end{aligned}$$

We agree that
 $\mathbf{x}=2$ and $\mathbf{y}=8$

$$\begin{aligned}\mathbf{x} &= \mathbf{x} + 4 \\ \mathbf{y} &= \mathbf{y} - 4\end{aligned}$$

We agree that
 $\mathbf{x}=6$ and $\mathbf{y}=4$

$$\begin{aligned}\mathbf{x} &= \mathbf{x} - 5 \\ \mathbf{y} &= \mathbf{y} + 5\end{aligned}$$

We agree that
 $\mathbf{x}=1$ and $\mathbf{y}=9$

A smart contract is..

Just like a normal computer program,
except that multiple computers run it
instead of one computer

```
function _transfer(address from, address to, uint256 amount) internal virtual {
    require(from != address(0), "ERC20: transfer from the zero address");
    require(to != address(0), "ERC20: transfer to the zero address");

    _beforeTokenTransfer(from, to, amount);

    uint256 fromBalance = _balances[from];
    require(fromBalance >= amount, "ERC20: transfer amount exceeds balance");
    unchecked {
        _balances[from] = fromBalance - amount;
        _balances[to] += amount;
    }
}
```

```
function _transfer(address from, address to, uint256 amount) internal virtual {  
    require(from != address(0), "ERC20: transfer from the zero address");  
    require(to != address(0), "ERC20: transfer to the zero address");  
  
    _beforeTokenTransfer(from, to, amount);  
  
    uint256 fromBalance = _balances[from];  
    require(fromBalance >= amount, "ERC20: transfer amount exceeds balance");  
    unchecked {  
        _balances[from] = fromBalance - amount;  
        _balances[to] += amount;  
    }  
}
```

But you don't have to write this yourself

 **CMTA / CMTAT** Public

Reference Solidity implementation of the CMTAT token developed by CMTA to tokenise securities in compliance with the Swiss law.

www.cmta.ch


MPL-2.0 license

32 stars 15 forks

★ Starred

👁 Unwatch ▾

[Code](#) [Issues](#) 17 [Pull requests](#) 1 [Discussions](#) [Actions](#) ⋮

 **CMTA / CMTAT-Tezos-FA2** Public

forked from [airgap-it/CMTAT-FA2](#)

Implementation for Tezos of the CMTAT token developed by CMTA to tokenise securities in compliance with the Swiss law.

```
/**
 * @dev calls the different initialize functions from the different modules
 */
function __CMTAT_init(
    address admin,
    string memory nameIrrevocable,
    string memory symbolIrrevocable,
    string memory tokenId_,
    string memory terms_,
    IEIP1404Wrapper ruleEngine_,
    string memory information_,
    uint256 flag_
) internal onlyInitializing {
    /* OpenZeppelin library */
    // OZ init_unchained functions are called firstly due to inheritance
    __Context_init_unchained();
    __ERC20_init_unchained(nameIrrevocable, symbolIrrevocable);
    // AccessControlUpgradeable inherits from ERC165Upgradeable
    __ERC165_init_unchained();
    // AuthorizationModule inherits from AccessControlUpgradeable
    __AccessControl_init_unchained();
    __Pausable_init_unchained();

    /* Internal Modules */
    __Enforcement_init_unchained();
}
```

CMTA's open token framework and code

CMTAT

NEW!

CMTAT version 2.3



github.com/CMTA/CMTAT

NEW!

CMTAT version 2.3

FREE

github.com/CMTA/CMTAT



- Debt tokenization support
- Many tech improvements
- Security audit

CMTA AGM – Keynote

Thank you!

JP Aumasson – CMTA, technical committee chair – Taurus, co-founder & chief security officer