TSI TRANSPORTA UN SAKARU INSTITŪTS

iDEA HUB

ArgentoLab

IDEA PROPOSAL: ***payment solution on blockchain***

NATIONAL DEVELOPMENT PLAN 2020

EUROPEAN UNION
European Regional Development Fund

INVESTING IN YOUR FUTURE

Company profile

Industry:

- IT, Innovation’s development

Directions for innovation solutions

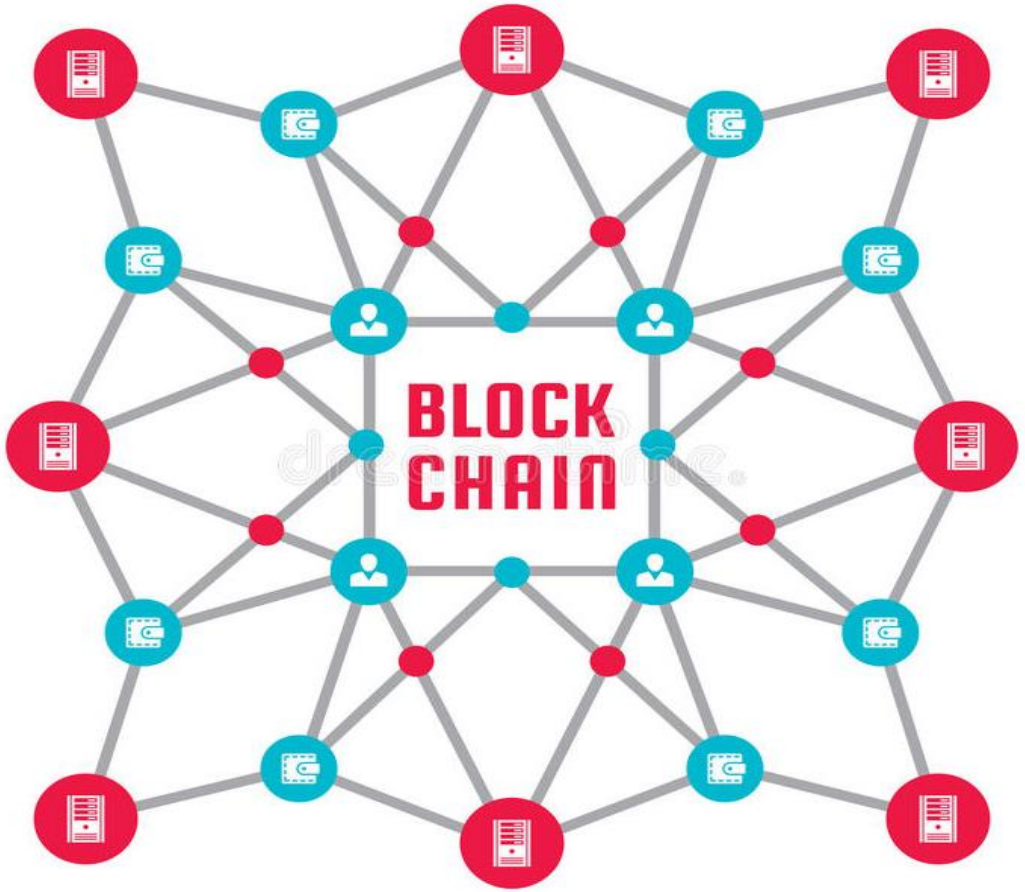
- Biomedicine technologies
- Complex digitalization
- Blockchain solutions

BACKGROUND

- Payment system works in non-custodian mode, and it never touches customers or merchant keys. Payment system changes commission from each transaction paid to the merchant. This functionality is backed by smart contract technology deployed to the Ethereum network.

- Payment system should allow merchant to accept stable coins, available on Ethereum blockchain. Payment system generates new blockchain address for each transaction by deploying new smart contract.

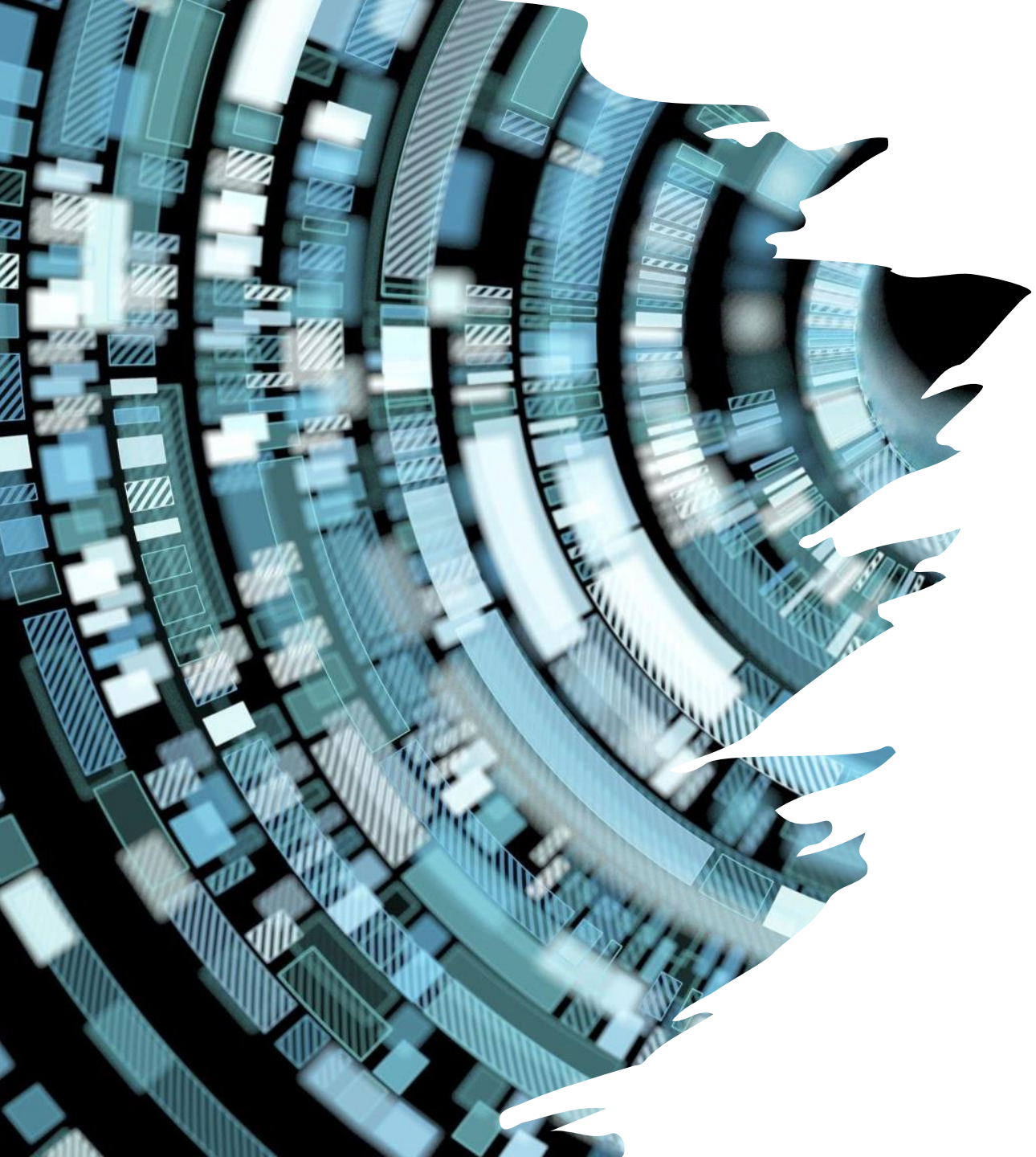
- Customer transfers token to smart contract address to make the purchase. Payment system monitors transfers to that blockchain address and transfers it to the merchant wallet by calling smart contract settlement function.



PROBLEM:

- The payment system should be gas-efficient; transactions should be executed as fast as possible. Deploying smart-contracts upfront adds additional, unnecessary costs before a customer even transfers any money.

- It is required to develop the most gas-efficient solution that allows accepting payments to smart-contract addresses



CHALLENGE:

- Deploying a factory contract that stores all the payment contract sources and can instantiate new contracts can significantly reduce new contract deployment costs (Factory Contract – Blockchain Patterns (csiro.au))

- Use minimal proxy contract. A minimal proxy contract allows deploying an exact copy of another existing Contract. Instead of deploying a new contract every time we need to perform the transaction, we can use a clone of the contract deployed for the first time, which can significantly reduce gas costs.

- Use CREATE2 for contract deployment. Using Ethereum CREATE2 opcode, it is possible to precompute the contract address without deploying it. It allows us to deploy only those contracts paid by the customer and avoid additional costs for deploying unused contracts for every operation.

SOLUTION VISION

- The solution has a payment system smart contract written in solidity and smart contract unit tests
- Hardhat development tools are recommended for contract and unit test development
- The smart contract should use all 3 recommended techniques for gas optimization: Factory Pattern, Minimal proxy contract, and CREATE2 opcode
- Additionally, a sample tool (c# or javascript) should demonstrate the generation of the contract address of the chain and complete payment with contract deployment