



Instant Loan Solution designed for Banks

Eureka! Team Members:

Riccardo Pasti

Ivan Brunato

Michele Bersani

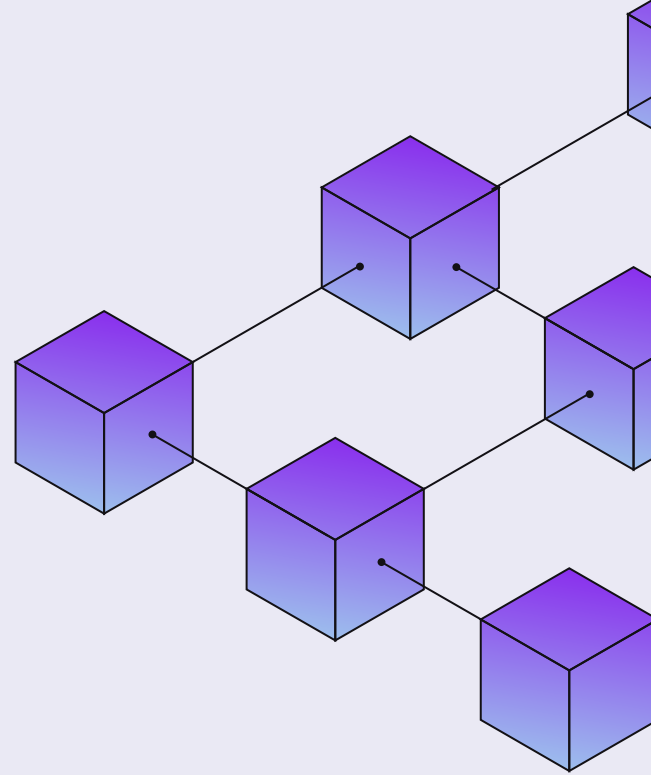
Riccardo D'Angelo

Giuseppe Pipitone

Use Case

Eureka! is a **white-label blockchain solution** that aims to be implemented into commercial banks credit offers to their retail customers.

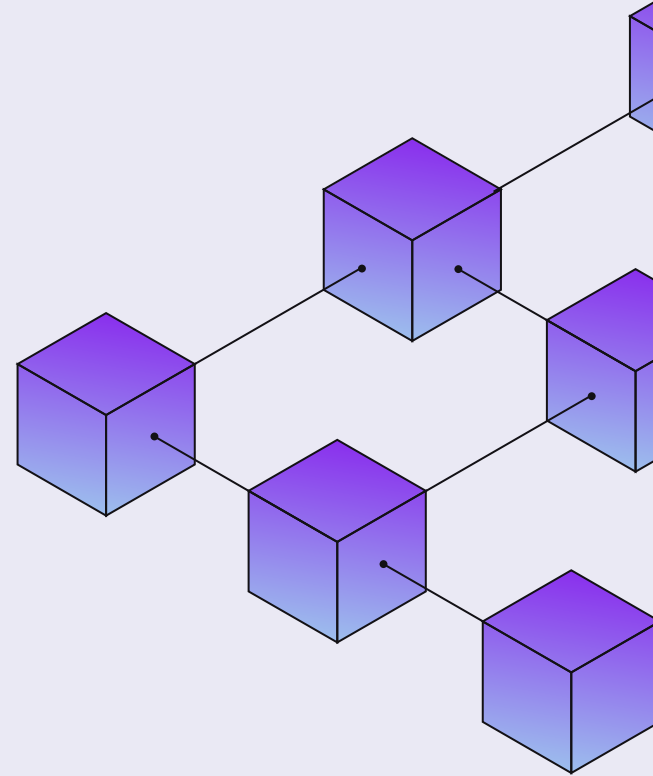
Eureka! enables banks to provide **instant collateralized loans by tokenizing customers' balance liquidity.**



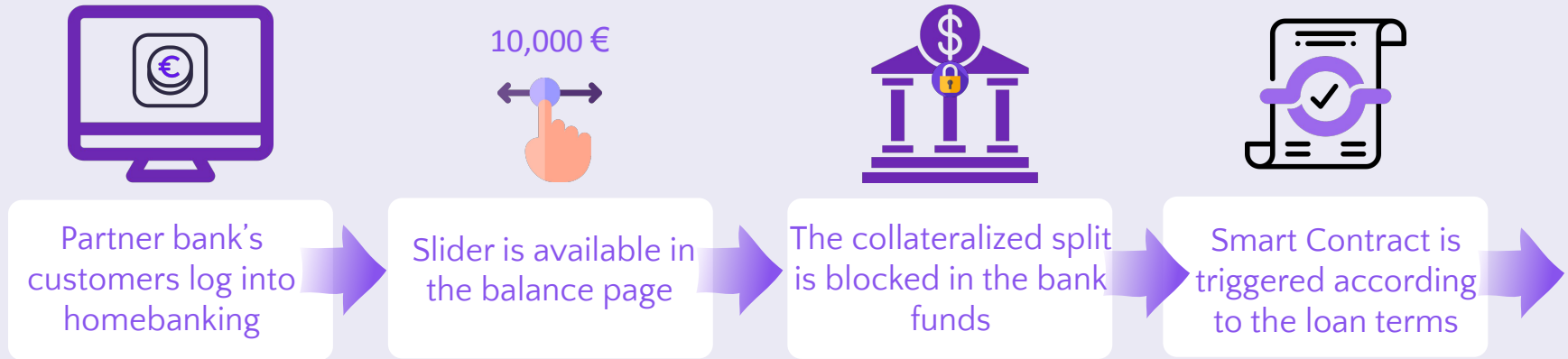
Asset Class

Tokenizing Liquid Assets in Bank Accounts

- The **Eureka!** token is a direct tokenization of each Euro allocated as collateral.
- Allowing customers to collateralize **liquid** assets in their accounts and receive instant loans based on the collateral LTV, up to 90%.



How Eureka! works?



Once the banking software is connected to the smart contract, a new feature can be rolled up into customers homebanking

How Eureka! works



Eureka! token is minted 1:1 to the customer wallet



Interest gets accrued every term as minted Eureka!



Each repayment proportionally burns deposited tokens

Once the customer wallet is empty, the smart contract instantly triggers and interacts with the banking software to **unlock the collateralized liquidity**, finally terminating the loan cycle.

Utility and Features



Instant Loans

Thanks to the network speed and homebanking white label solution the customer reaches a new level of financial experience.



Mitigated Risk

Due to the sovra-collateralized loan nature, the bank can enforce liquidation of the amount according to its loan term and recover all of the credit.



Scalability

By adopting an Ethereum sidechain solution, scalability is at the center of the value proposition, without blockchain “trilemma” tradeoffs.

Utility and Features



No Enforcement Cost

Since the collateral cannot be withdrawn or spent, but can be liquidated, there is no need of enforcement and all the costs linked: legal fees, lost interest, collateral depreciation and time wasting.



Better Rates

Thanks to a lower risk associated with the loan, the bank can provide better rates for credit on top of fixed and transparent interests to the customers.



On-Chain Records

All the loans transactions are stored with their metadata, allowing the bank to retrieve truthful data from the blockchain.

Revenue Model



One-Time Implementation Fee

Charged to the bank when they first adopt the solution. It covers the initial set-up, customization to align with the bank's existing systems and products.



% on total value locked

Essentially, the business takes a small percentage of the total amount of funds that are being managed through the platform.



Transactions Fees

Flat and transparent fee for each transaction triggered using Eureka! tokens.

What is the cost for a bank to recover a 10.000 € debt in Italy?

Total cost for the debt recovery procedure starts at **2629.50 €**



```
graph TD; A["Total cost for the debt recovery procedure starts at 2629.50 €"] --> B["Payment injunction to the debtor by formal letter from the lawyer: €100"]; A --> C["Court procedure: unified contribution to register the case: 118 € revenue stamp: 8 € average lawyer's fee for current legislation (Ministerial Decree 55/2014): 2,403.50 €"];
```

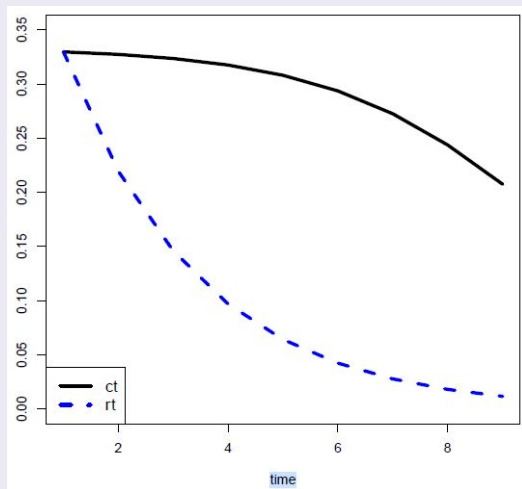
Payment injunction to the debtor by formal letter from the lawyer: €100

Court procedure:

- unified contribution to register the case: 118 €
- revenue stamp: 8 €
- **average lawyer's fee for current legislation (Ministerial Decree 55/2014): 2,403.50 €**

Tokenization: Reinventing Debt Recovering

Due to the statistical **increase in Non Performing Loans**, today it's better to prevent **debt recovery** by creating new bank loan guarantees, such as Eureka solution.

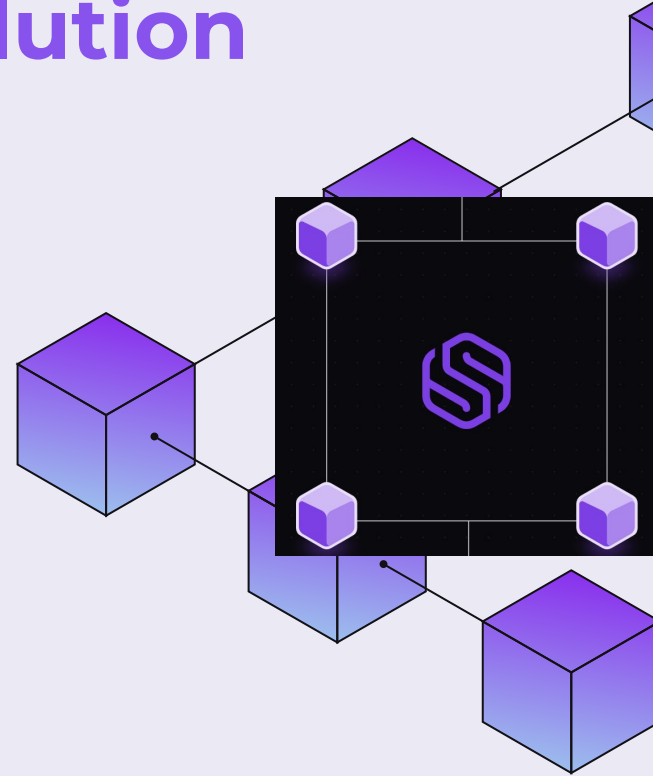


The recovery curve of an NPL portfolio, considering $T = 9$ years of delay in payment from the default. The recovery curve is expressed both in terms of recovery rate rt (dashed blue line) and conditional recovery rate ct (solid black line) over the years.

At the end of this procedure, it must be verified that the debtor has economic assets to pay the remaining debt, finalizing **a really time-consuming operation.**

Technological Solution

- We adopt a private blockchain network (Polygon Supernets) for security and data privacy compliance.
- Polygon Supernets is a Polygon's solution to build and power app-chains. Supernets are powered by Polygon's cutting-edge EVM client technology.
- Programmable in Solidity.
- Resources needed include blockchain developers, finance experts, business developers and oracle set up.

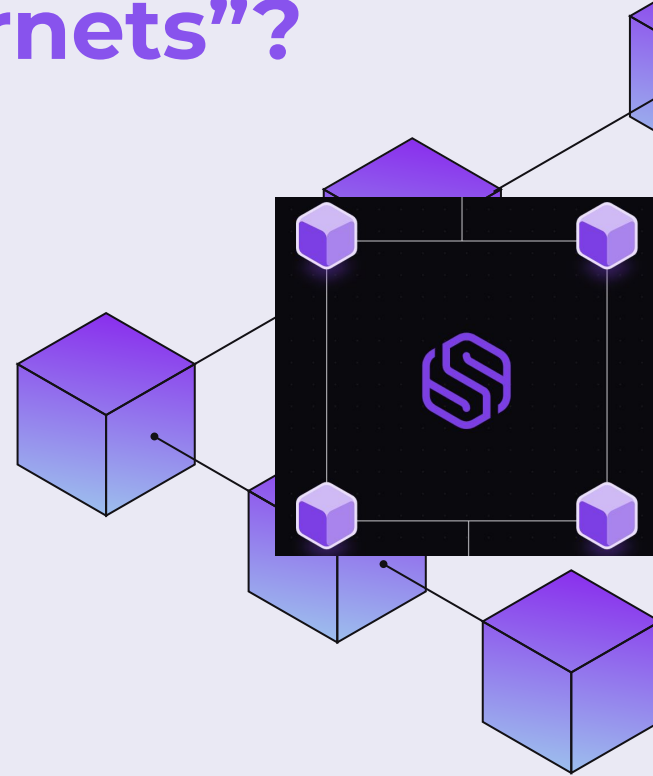


[Build and power App-chains with industry-leading tech & partner ecosystem](#)

[Deploy a Smart Contract Using Remix](#)

Why “Polygon Supernets”?

- Framework for blockchain development.
- Native bridge connection with Polygon PoS mainnet, inheriting its security and capabilities.
- PolyBFT consensus mechanism.
- Offer node and deployment infrastructure, indexers, explorers, oracles, and many tools for building and deploying Supernets and their apps.
- Cloud deployment available - taking all the advantages without thinking about managing their own infrastructure.



[What are Polygon Supernets?](#)

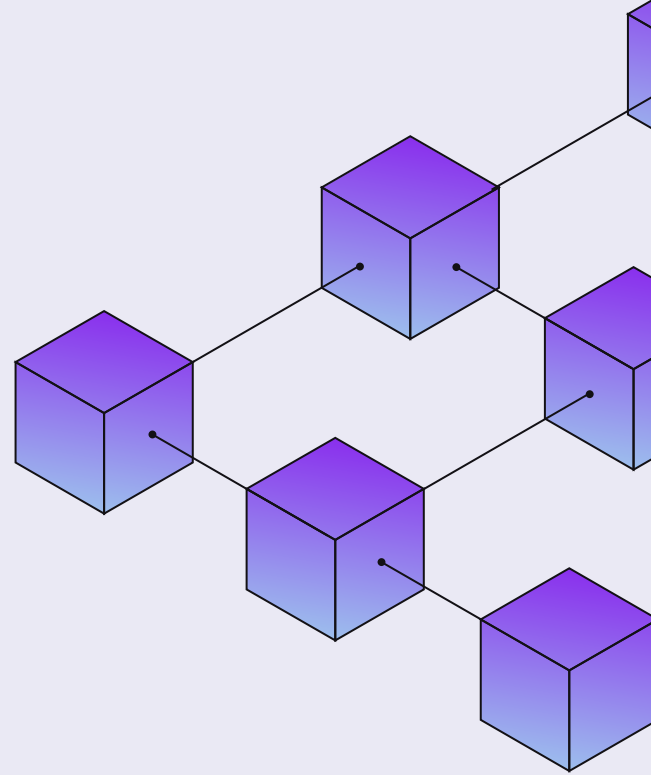
[PolyBFT - Polygon Byzantine Fault Tolerance](#)

Tokenomics

“Eureka” is an ERC-20 token, compatible with Polygon and Ethereum technologies. These tokens are private so it means that you can’t trade them for other Ethereum-based digital currencies.

- Tokenomics

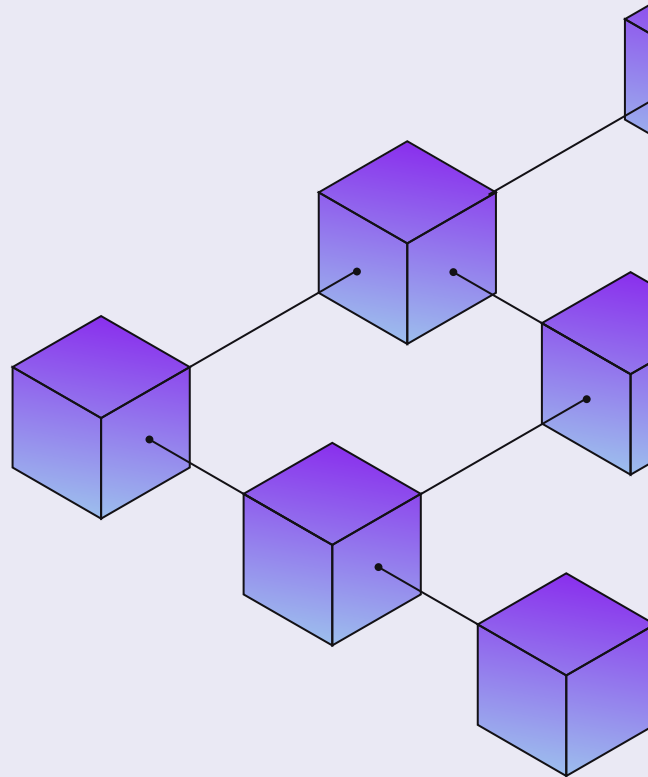
Every “Eureka!” token minted will be burned after the payment of the same value by the customer. If the customer won’t respect the conditions of the smart contract or the interest added to the loan reaches the blocked liquidity, the tokens will be burnt and the bank will take the money blocked by the customer.



Blockchain

It'll be a private blockchain, verifiable only by the bank and its customers.

- Generated using a EVM, each node will have a “validator private key” and a “networking private key”.
- The “Genesis block” will be configured with the block limit, consensus mechanism and ID of the chain.
- Start the node.

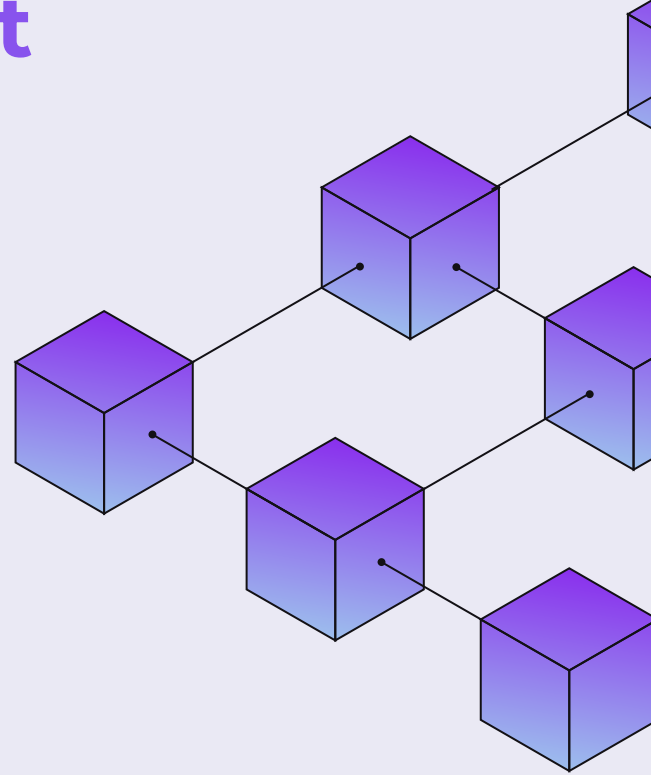


Smart Contract

- Smart Contract:

When the customer accepts the contract, a wallet is generated and associated with the customer's CC. An oracle will query whether the characteristics of the contract can be respected or not.

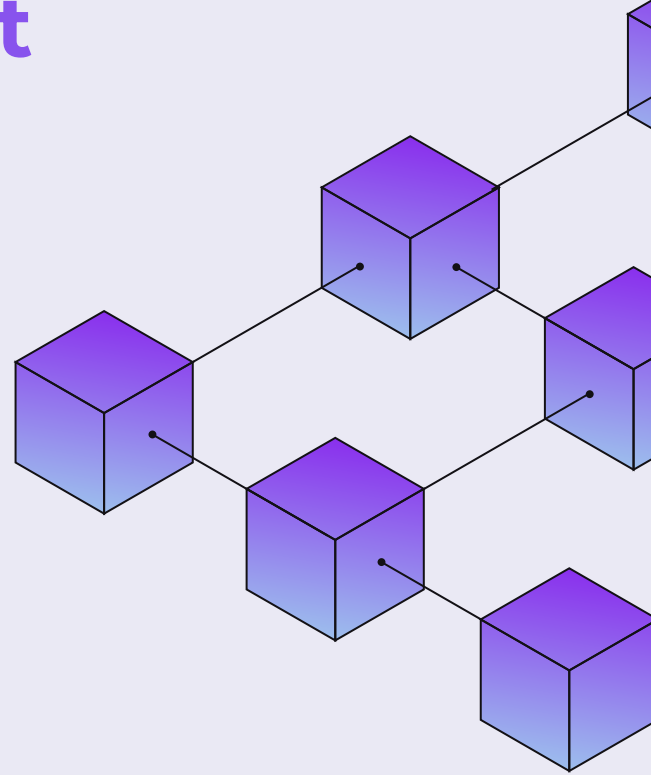
Then the smart contract generates 1 Eureka = 1€ and then the customer receives the amount in euros of the requested loan together with the number of tokens that guarantee the blocked liquidity.



Smart Contract

These tokens will then be "burned" as the customer pays the loan, unlocking the initial liquidity, or if the customer fails to comply with the conditions of the smart contract, this capital will no longer be available to the customer, leaving it totally managed by the bank.

Here it is an simple example of a smart contract with 10.000€ blocked in order to receive a 90% loan with the same value even in Eureka:



[What are Polygon Supernet?PolyBFT - Polygon Byzantine Fault Tolerance](#)

Market Sizing



All customers of traditional banking services who need fast and small loans



Bank customers comfortable with digital banking services but not with DeFi



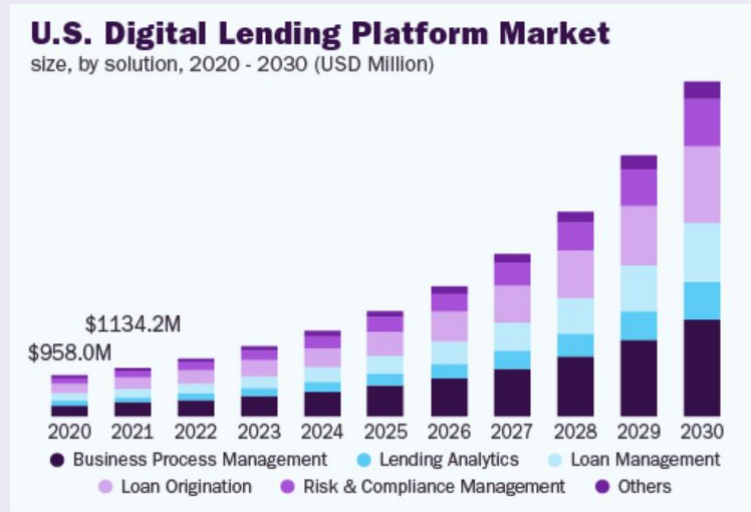
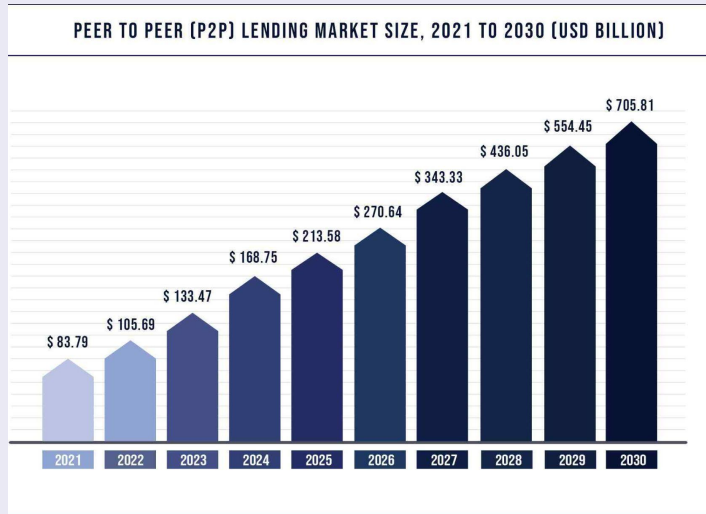
Customers of partner banks open to innovative banking solutions on blockchain



Focus on partner banks willing to pilot our system in order to simplify lending services

A rapidly growing market

Rising demand for convenient and efficient lending solutions



Competitive Landscape

Direct competitors: Traditional banks, P2P lending platforms

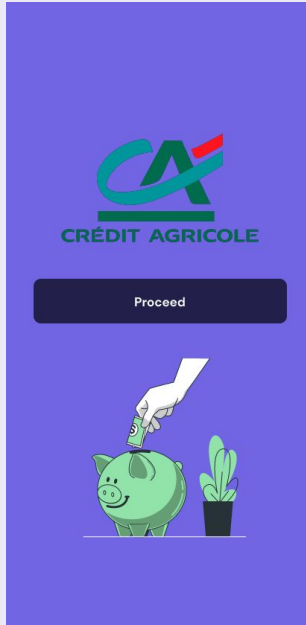
Indirect competitors: Fintech startups, DeFi lending platforms, Savings and Loan Associations, Credit Unions

Global Digital Lending Platform Market

share, by end-use, 2022 (%)



UX Mock-UP



[Figma Demo](#)



Smart Contract



[Link GitHub](#)

Conclusions

- Become our very first Bank Partner to provide simple and fast collateralized loans through Eureka!
- Asked Investment: 100'000 €
- Outcome: MVP connected with Partner's banking system and go-to-market design strategy

