

Morpho

Adrien Laversanne-Finot
adrien.lf@morpho.xyz

20/03/26

What is Morpho

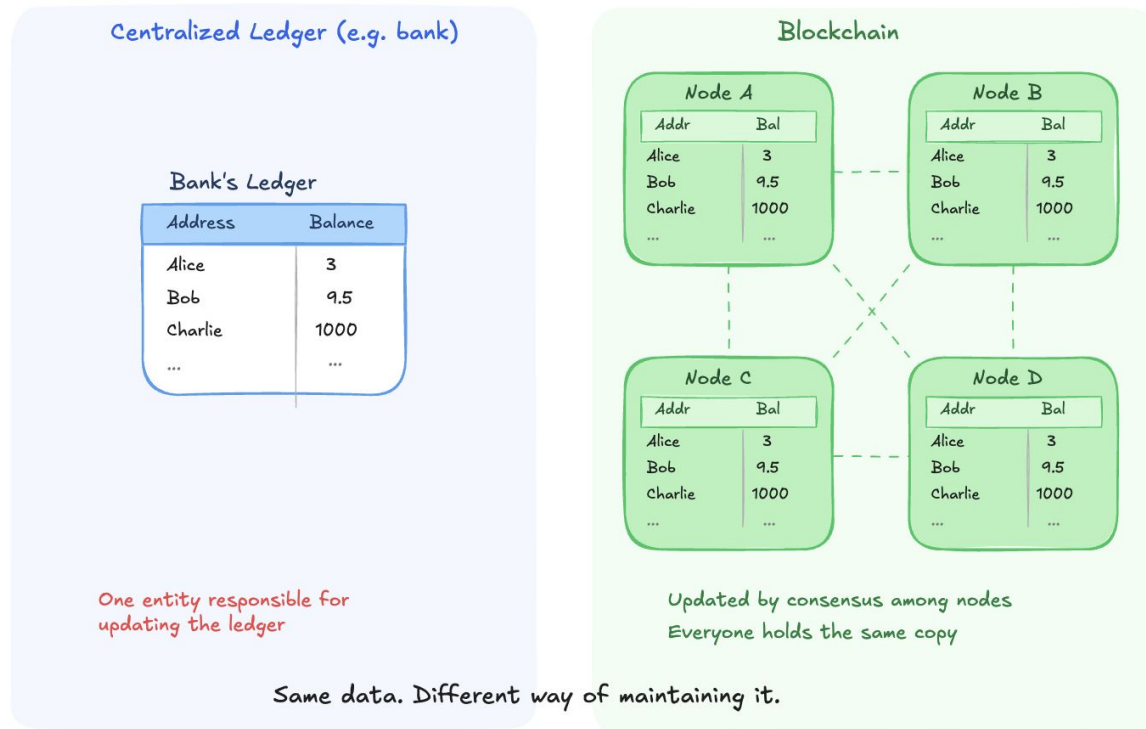
- Morpho was started in 2021. Now a team of about 60 people.
- Morpho is developing an infrastructure for lending and borrowing on the blockchain.
- This infrastructure can be used by actors to issue loans or to lend money.
- Some of the largest fintechs, banks or asset managers are already using Morpho (e.g. Coinbase, Société Générale...).

What I do at Morpho

- Part of the protocol team that is responsible for designing and implementing the different versions of the protocol.
- A lot of focus on design and security the protocol as developing a financial protocol on a decentralized system is tricky!

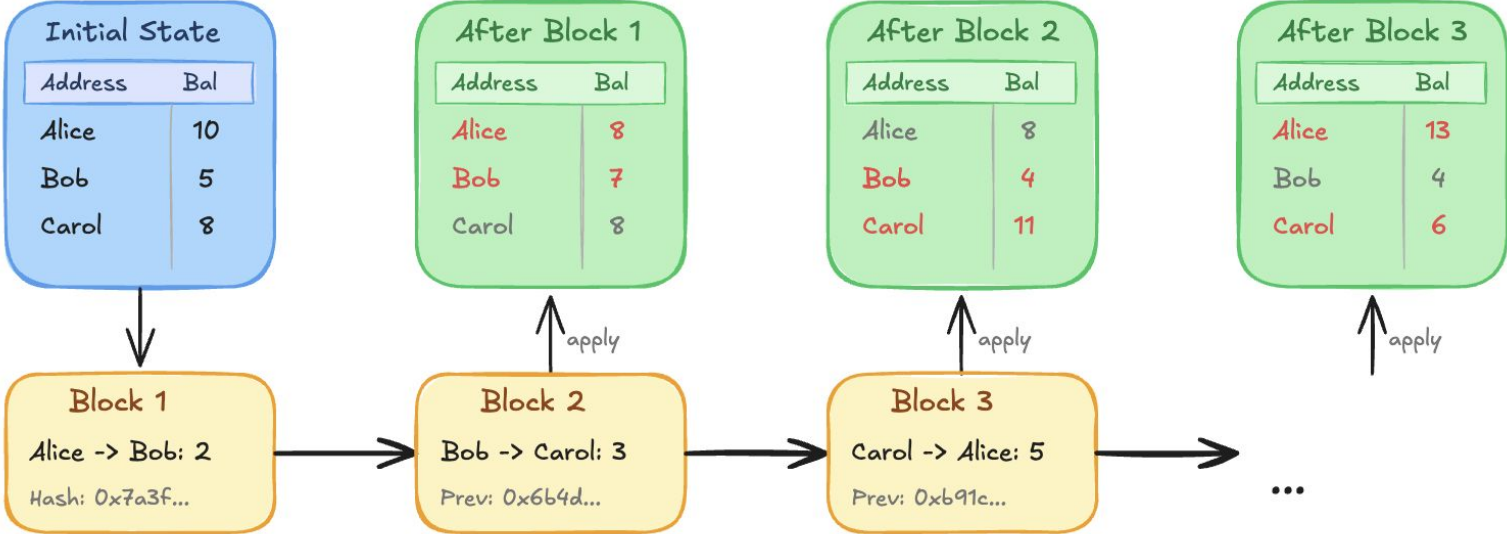
Blockchain basics

Blockchain basics



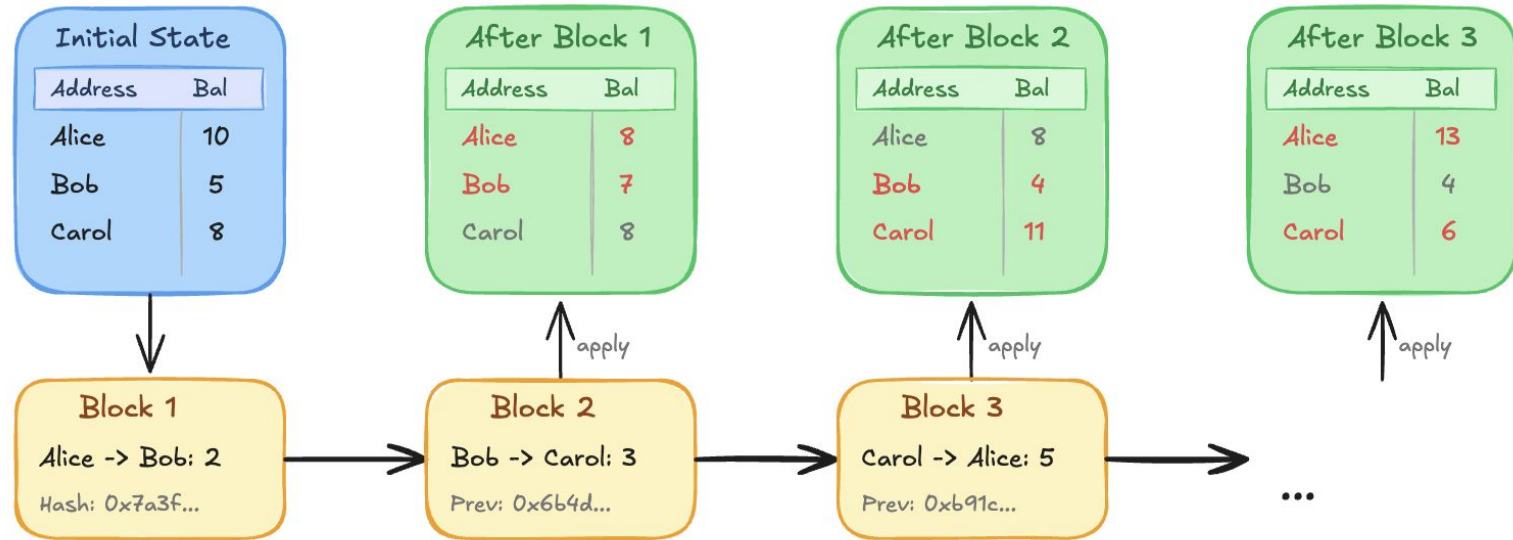
- A shared ledger that no single entity controls
- Shared ledger updated according to consensus rules (e.g. everyone agrees, BFT, Nakamoto)

Blockchain basics



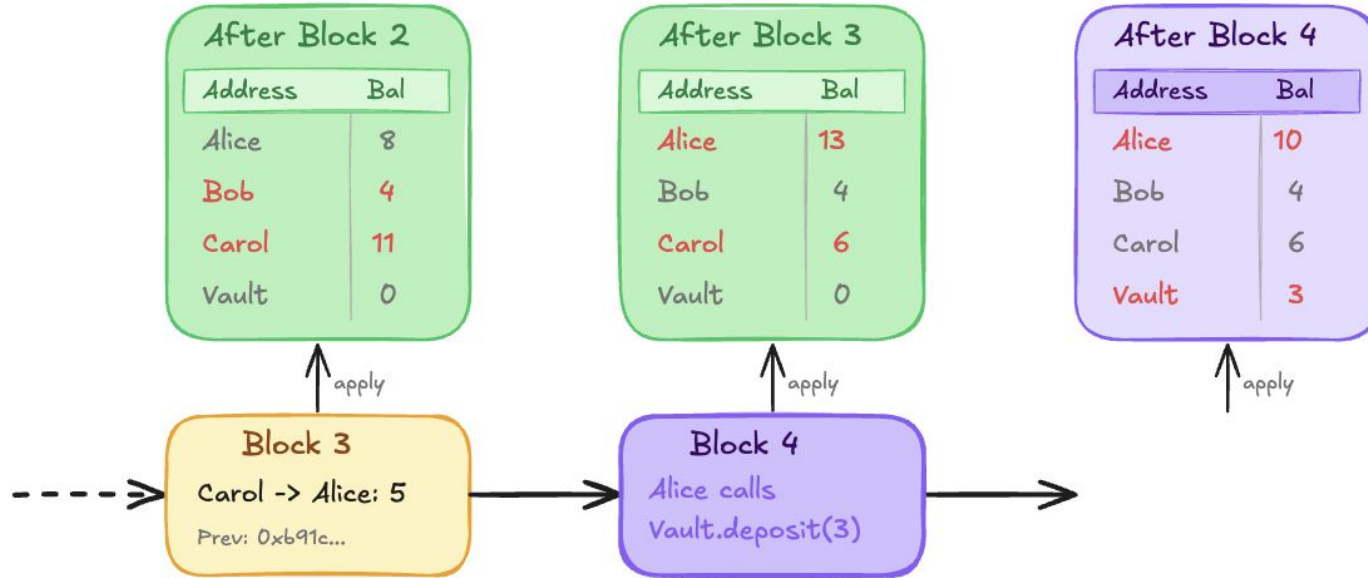
- Ledger is updated through transactions.
- Transactions are grouped into blocks, chained cryptographically.

Blockchain basics



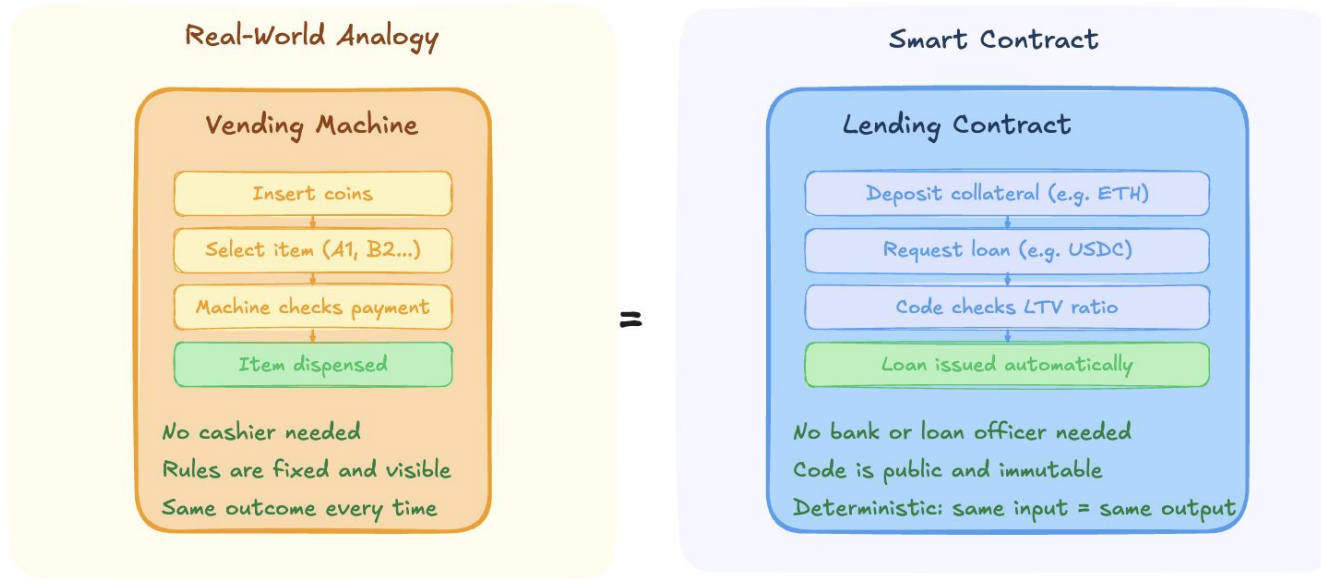
- Thousands of nodes worldwide each hold a full copy
- Consensus mechanism ensures everyone agrees on the same state
- Once written, data is practically immutable

Blockchain basics: Smart Contracts



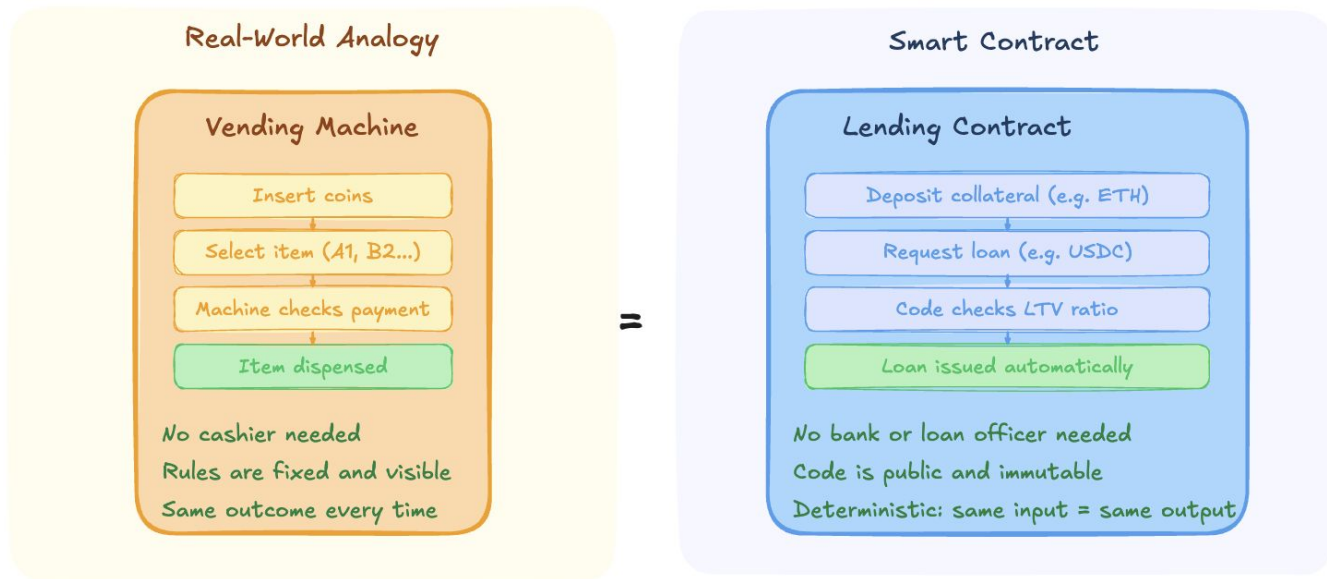
- The ledger can be also updated using Smart Contracts
- A smart contract is just another address in the ledger, controlled by code.

Blockchain basics: Smart Contracts



- Smart contract: code deployed on the blockchain
- Anyone can interact with (permissionless)
- It can hold and manage assets according to its programmed rules (trustless)
- Once deployed, the code cannot be changed (immutability)
- Anyone can read the code and verify what it does (transparency)

Blockchain basics: Smart Contracts



- Anyone can participate, 24/7, globally.
- Settlement in seconds, not days.
- Neutral infrastructure that competing parties can use.

Blockchain basics: A first (useless) Smart Contract

```
contract SimpleVault {
    mapping(address => uint256) public balances;

    function deposit() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount, "Not enough balance");
        balances[msg.sender] -= amount;
        payable(msg.sender).transfer(amount);
    }
}
```

- Simple contract that just stores Ethereum
- Morpho: instead of just storing balances, the contract also handles lending and borrowing logic.

Blockchain basics: some examples

Name		Category	TVL 
> 1	 Aave 20 chains		 \$25.747b
2	 Lido 5 chains	Liquid Staking	 \$19.36b
3	 EigenCloud 1 chain	Restaking	\$9.192b
4	 Binance staked ETH 2 chains	Liquid Staking	 \$7.738b
> 5	 Sky 1 chain		 \$7.542b
> 6	 Morpho 33 chains	Lending	 \$6.884b
> 7	 Ethena 1 chain		  \$6.663b

- Part of Decentralized Finance is to replicate financial services in a blockchain environment.
- Come with specific challenges due to the decentralized nature of blockchain (permissionless, limited resources, ...).
- Examples:
 - Swap one currency for another (Decentralized Exchange, DEX)
 - Borrow one asset against another (Lending Borrowing)
 - NFTs, Stablecoins...

Morpho Architecture: Markets



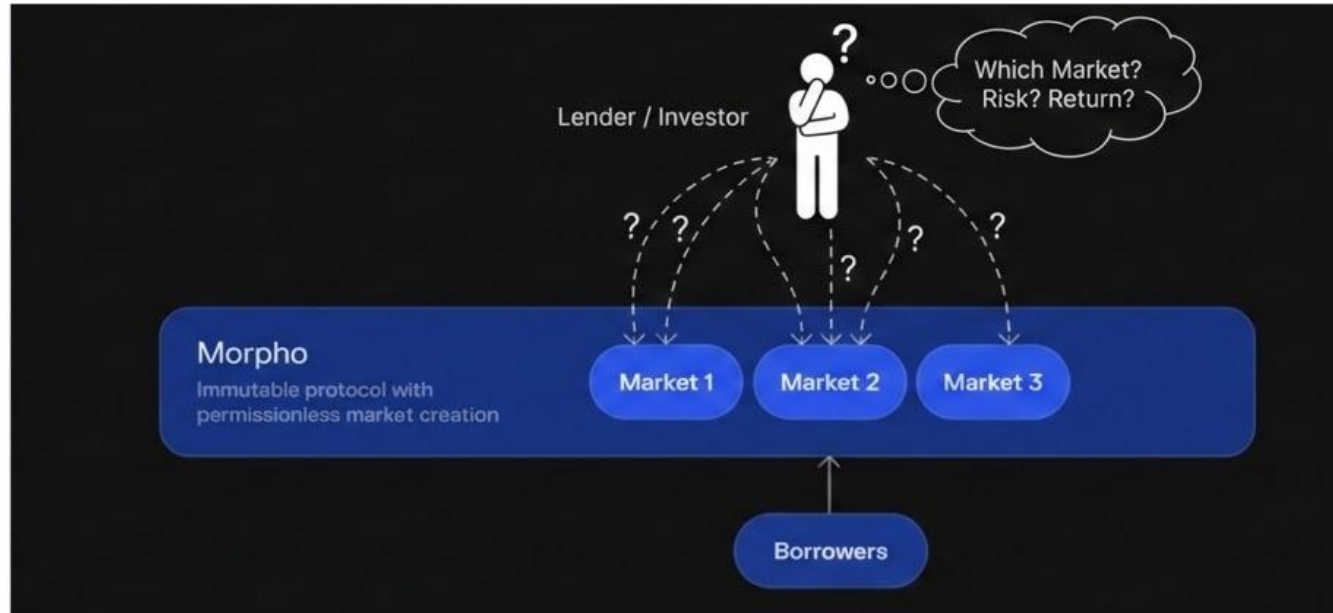
- Markets are primitive lending markets (e.g. different type of loans, different assets)
 - Example: USD loan backed by Bitcoin collateral.
- Anyone can create markets.
- Markets are shared: different lenders and borrowers interact with the same market as long as it's the same type of loans.

Morpho Architecture: Markets



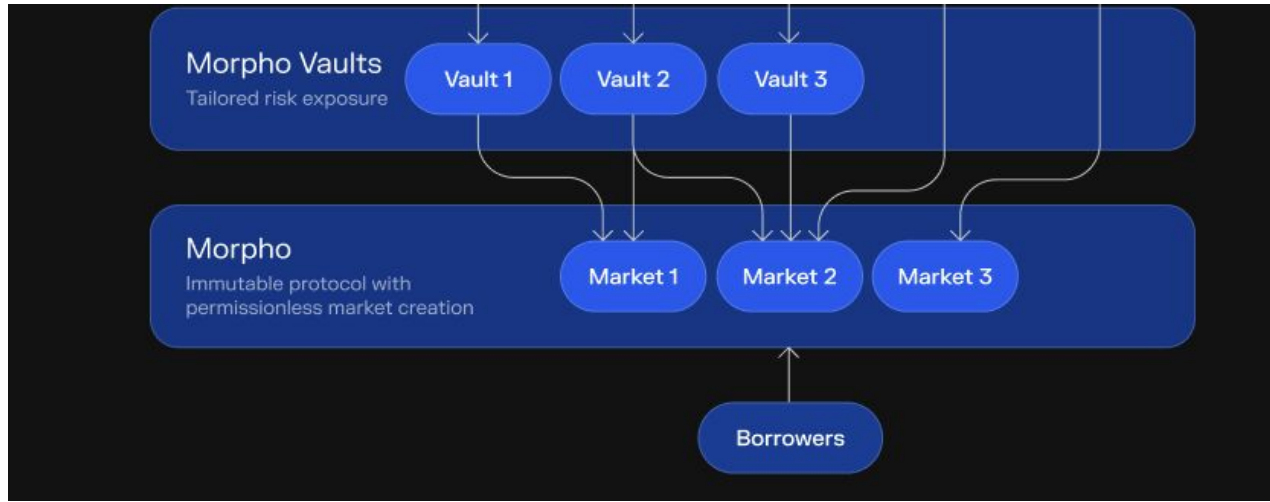
- Borrowers can tap directly in markets to borrow.

Morpho Architecture: Markets



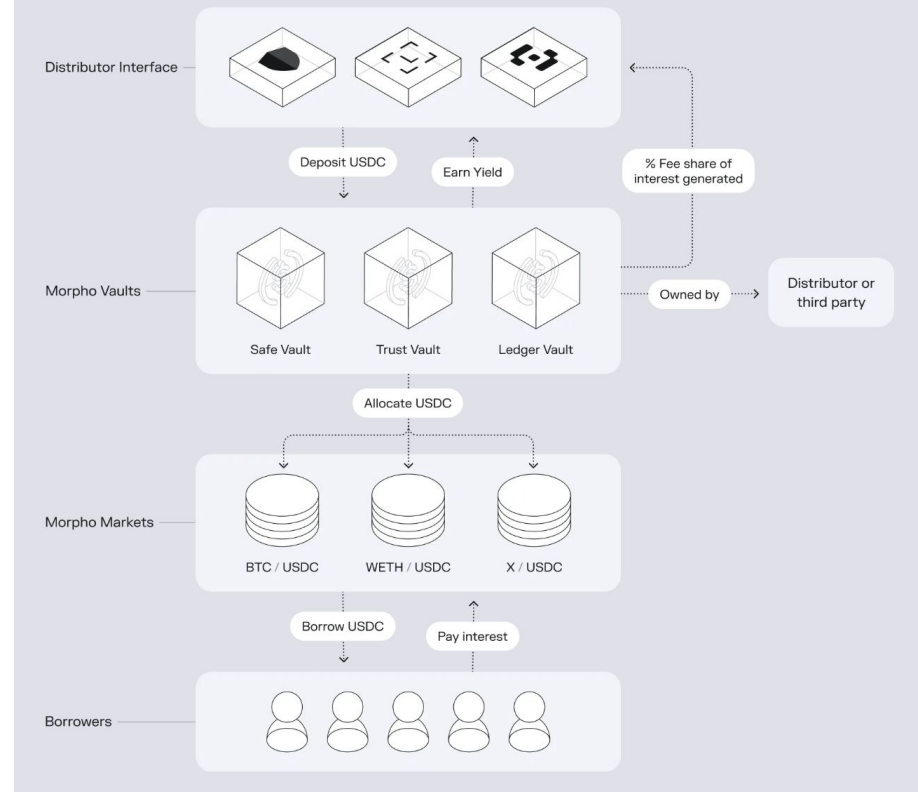
- Assessing the risk of different markets is difficult: it requires constant monitoring as risk change over time, it's operationally difficult...
- Some entities have expertise in assessing risk of different opportunities (banks, asset managers, ...).

Morpho Architecture: Vaults



- Vaults allow users to delegate management of their assets to external parties (curator)
- Curator allocate funds between different markets based on their risk assessment.
- There's no "one size fits all" risk profile. Some people want very secure products, some want to target higher yield. Depositors choose between vaults based on their risk appetite.
- Markets are shared between different vaults: competing vaults deposit into shared markets, no liquidity fragmentation, borrowers access a "global liquidity".

Morpho Architecture



- Fintechs can give users access to “earn products” using vaults under the hood (e.g. Société Générale, Coinbase, ...).
- They can own or chose a vault that is in line with their risk requirements.

Markets

- If you take a loan in the real life, you go to a bank and ask your banker for the terms (interest, duration, ...). If you don't repay, your assets get seized, you go to jail, ...
- What does it mean to create lending markets on a smart contract (remember that everything is set algorithmically!)?
 - How to ensure that people repay?
 - How to set interest rate?
 - Prices: how to accurately get prices for the assets?

Vaults

- If you have some assets in a bank account, the banker cannot use it freely (regulation, sanctions, ...)
- What does it means to create vaults on a smart contract?
 - How to ensure that the curator cannot walk out with the user funds?
 - How to enforce a given risk policy?
- Morpho Vaults are **non-custodial**: users can always opt-out of the vault, regardless of curators actions.

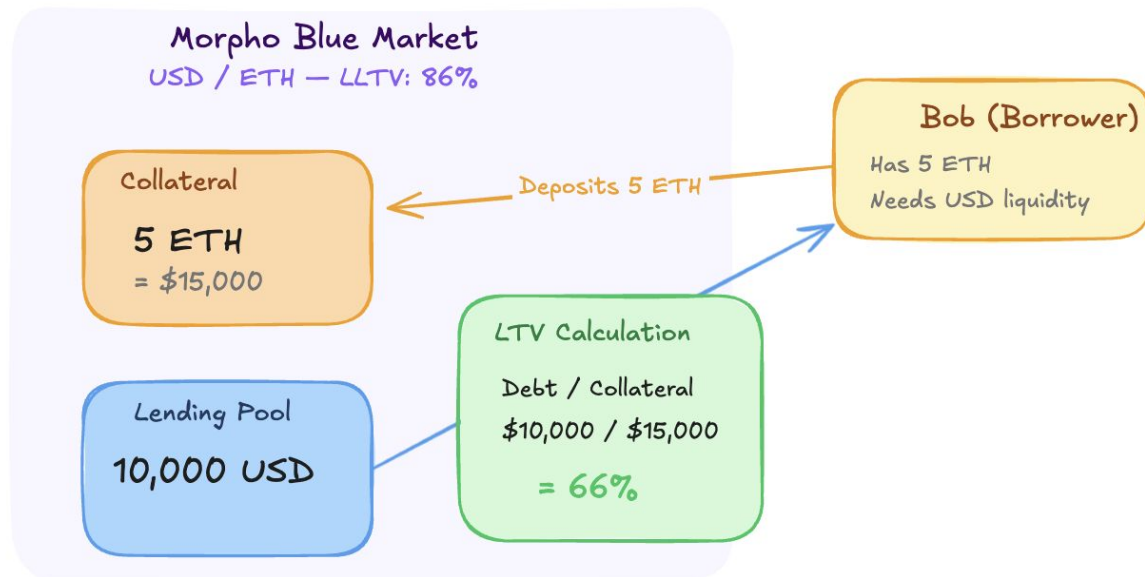
Markets: Loans

Lender	Amount Lent
Alice	10
Bob	15
...	...

- A loan is some sort of promise of repayment backed by some trust assumption [1].
- One simple trust assumption is collateral: you put \$150 worth of assets to borrow \$100.
- Morpho only does over collateralized loans (for now).
- Very similar to the repo market in essence.

[1] <https://morpho.org/blog/the-price-of-trust/>

Markets: Loans



- All loans are over collateralized (e.g. you put \$2000 worth of Ethereum and borrow \$1000 of USD).
- Borrowers can only borrow up to some loan to value (LTV = loan value / collateral value).
- Markets are defined by the loan asset, the collateral, the LTV (and the source price).

Markets: Loans

```
function borrow(
    MarketParams memory marketParams,
    uint256 assets,
    uint256 shares,
    address onBehalf,
    address receiver
) external returns (uint256, uint256) {
    Id id = marketParams.id();
    require(market[id].lastUpdate != 0, ErrorsLib.MARKET_NOT_CREATED);
    require(UtilsLib.exactlyOneZero(assets, shares), ErrorsLib.INCONSISTENT_INPUT);
    require(receiver != address(0), ErrorsLib.ZERO_ADDRESS);
    // No need to verify that onBehalf != address(0) thanks to the following authorization check.
    require(_isSenderAuthorized(onBehalf), ErrorsLib.UNAUTHORIZED);

    _accrueInterest(marketParams, id);

    if (assets > 0) shares = assets.toSharesUp(market[id].totalBorrowAssets, market[id].totalBorrowShares);
    else assets = shares.toAssetsDown(market[id].totalBorrowAssets, market[id].totalBorrowShares);

    position[id][onBehalf].borrowShares += shares.toUint128();
    market[id].totalBorrowShares += shares.toUint128();
    market[id].totalBorrowAssets += assets.toUint128();
    Accounting

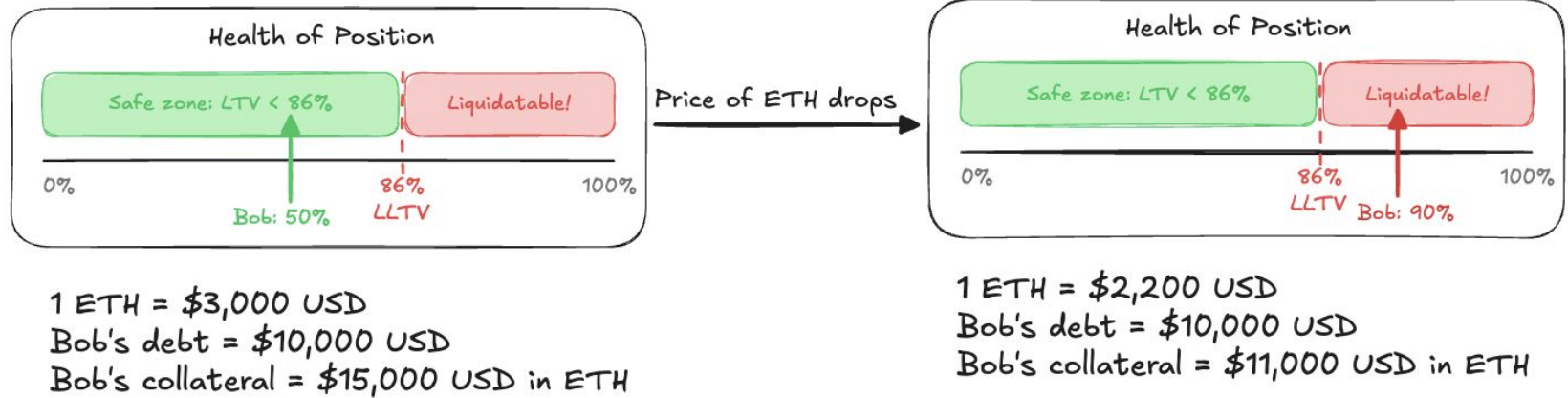
    require(_isHealthy(marketParams, id, onBehalf), ErrorsLib.INSUFFICIENT_COLLATERAL);
    require(market[id].totalBorrowAssets <= market[id].totalSupplyAssets, ErrorsLib.INSUFFICIENT_LIQUIDITY);
    Check LTV

    emit EventsLib.Borrow(id, msg.sender, onBehalf, receiver, assets, shares);

    IERC20(marketParams.loanToken).safeTransfer(receiver, assets);
    Transfer funds

    return (assets, shares);
}
```

Markets: Liquidations



- Price of collateral can change! If value of collateral fall below the loan value, lenders would lose.
- When the position is above the liquidation threshold (LLTV), anyone can seize the collateral by repaying the debt (and sell it): it's a function of the smart contract.

Markets: Liquidations



- Liquidators are incentivized to do so: they can take all collateral but only repay part of the loan.

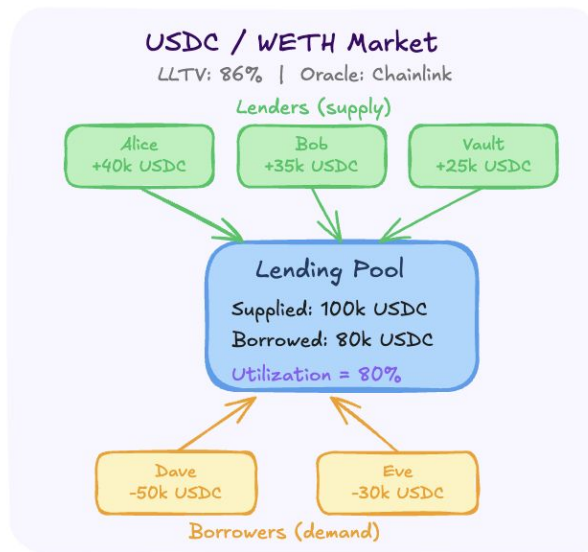
Actor	Before	After	Net result
Bob	5 WETH (\$11k) - 10,000 USDC debt	0 WETH 0 USDC debt	Lost ~\$1,000 (liquidation penalty)
Charlie	10,000 USDC 0 WETH	0 USDC 5 WETH (\$11,000)	Gained ~\$1,000 (liquidation incentive)
Lenders	10,000 USDC lent at risk if Bob defaults	10,000 USDC returned + interest earned	Protected (debt fully repaid)

Interest Rate

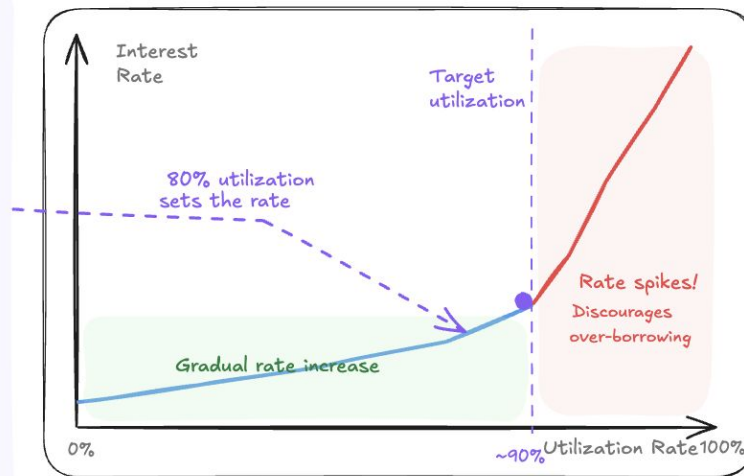


- Interest cannot be set once and for all. Interests vary a lot depending on macroeconomic.
- Could be a parameter that is set regularly but complex operationally, does not scale well to many different loans.

Interest Rate

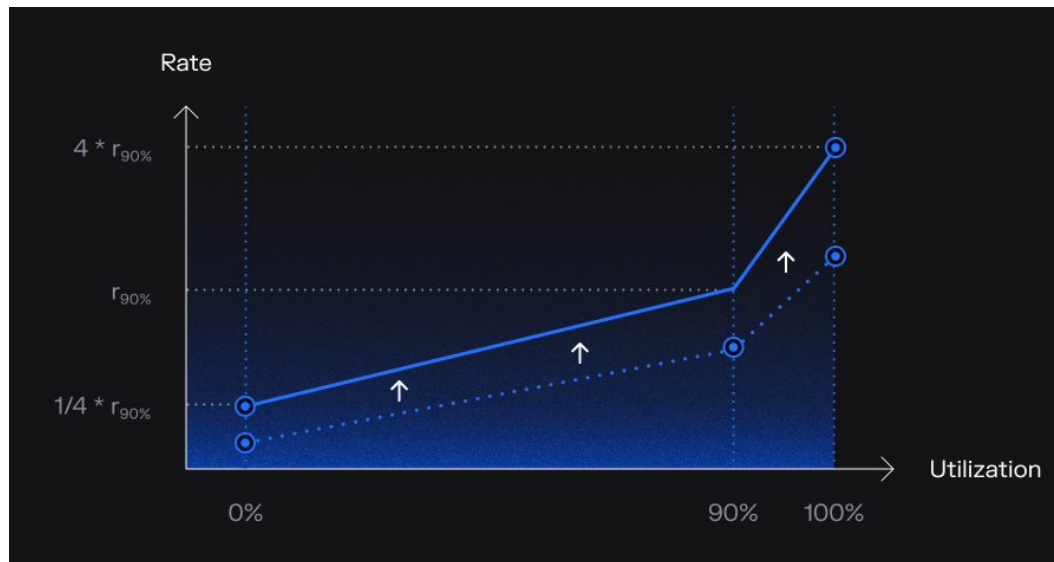


The Utilization Rate Model



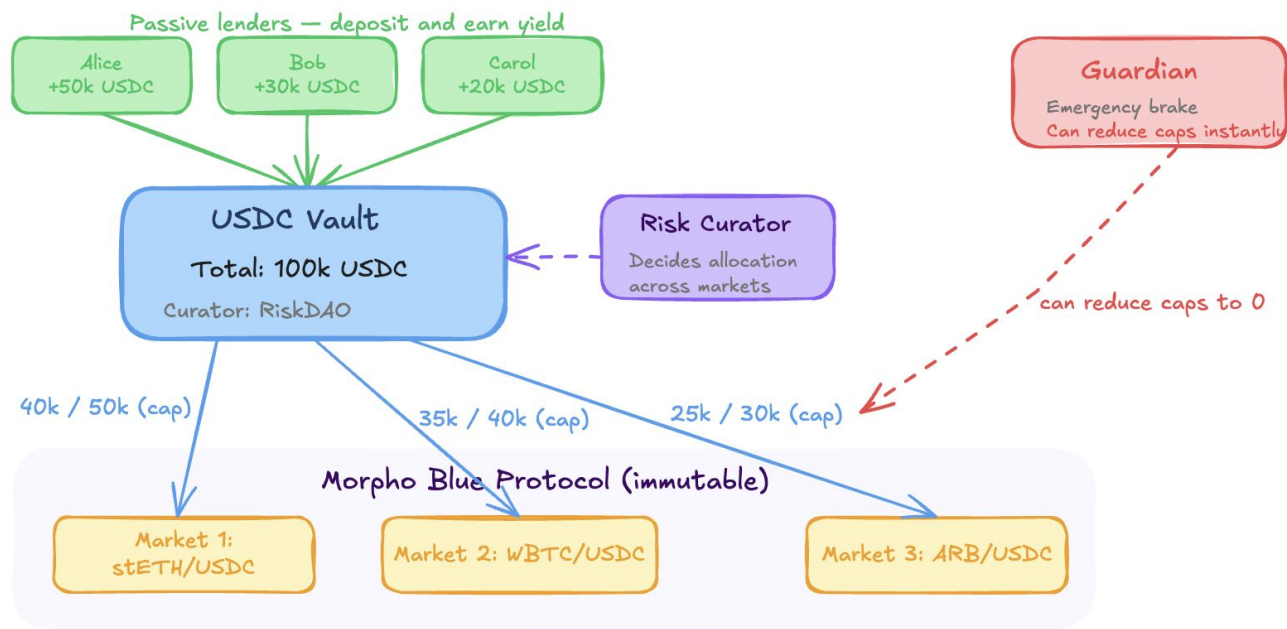
- Interest is variable and depend on utilisation.
- Rate increase slowly until target utilisation rate (90%), then very steeply.
- Self-balance: high-rates attract more lenders and push borrowers to repay. Utilization drops and rates decrease, and conversely for low rates

Interest Rate



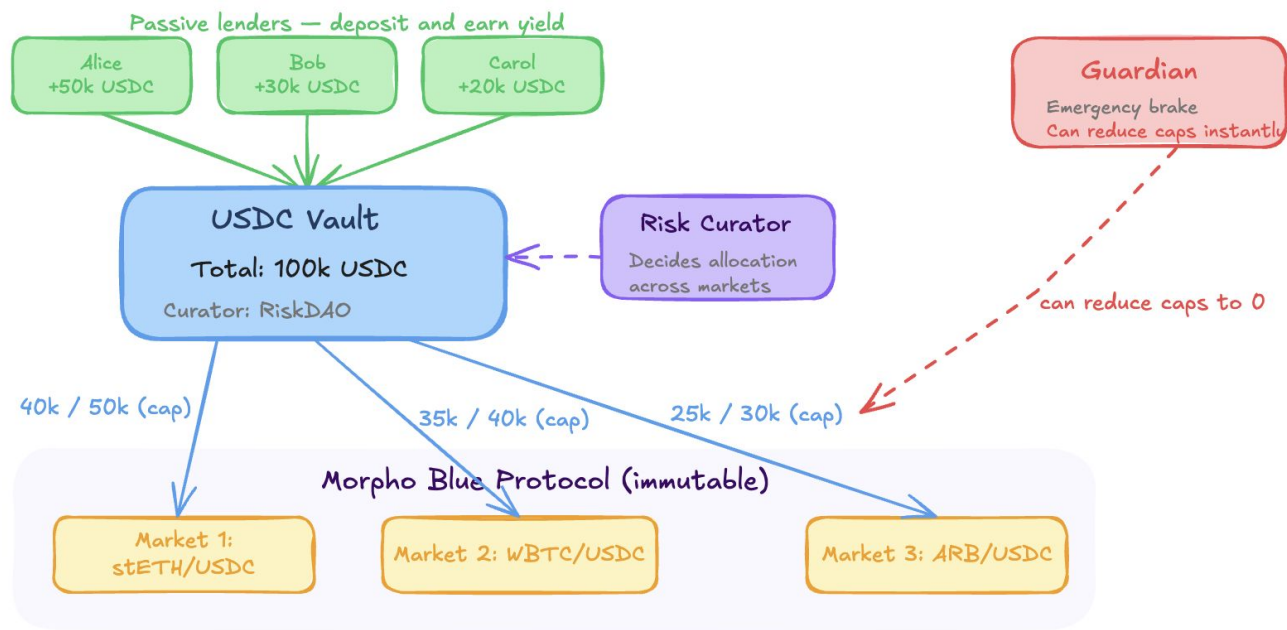
- It's not enough to have a static curve as the rate at target utilization is asset dependent (lenders want to get paid more for riskier collaterals or LTV).
- The curve is adaptive: if the current utilization stays above the target utilization, the rate at target utilisation will increase and inversely if utilization is below target utilization [1].

Vaults



- Morpho Vaults are **non-custodial**: users can always opt-out of the vault, regardless of curators actions.
- Achieved using a combination of different roles (owner, curator, allocator, guardian), with segregated responsibilities and timelocks for actions.

Vaults



- Curator: define the strategy (markets, caps for all markets).
- Allocator: allocate inside strategy defined by the curator.
- Guardian: can remove markets in case of emergency.
- Timelock: roles cannot take risky actions immediately, it takes a cooldown period.
- Example: A curator cannot allocate to a new market instantaneously, it takes time. If depositors do not agree with the curators action, they have opportunity to get out.

Limitations and perspectives

- Interest rates are determined by the IRM. Works but is not the most efficient design.
 - Rates can be volatile. Lead to unpredictable for borrowers.
 - Underutilization of capital (90% target rate).
 - Can't have fixed rate.
 - Can't use different collaterals. Or no collateral.
- Now developing a new fixed rate protocol that works as zero coupon bonds.
 - Market driven rates.



Thank you!