

EAS 5830: BLOCKCHAINS

ERC-20s

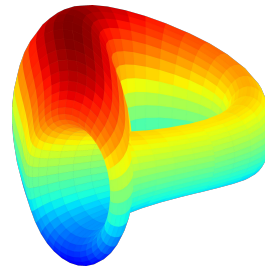
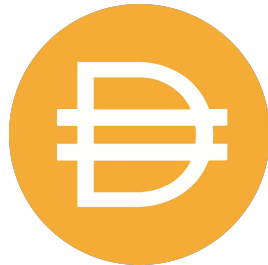
Professor Brett Hemenway Falk

Creating Tokens

- BTC (Bitcoin) - 2009
 - LTC (Litecoin) - 2011
 - DOGE (Dogecoin) - 2013
- XRP (Ripple)- 2012
- ETH (Ethereum) - 2013
- ADA (Cardano) - 2017
- TRX (Tron) - 2018
- ALGO (Algorand) - 2019
- AVAX (Avalanche) - 2019
- SOL (Solana) - 2020

ERC-20s

- The ERC-20 token standard is a voluntary standard for fungible tokens
- An ERC-20 token is a contract
- Deploying a new contract on Ethereum is easier than creating a new blockchain
- Hundreds of thousands of ERC-20s have been created



Tokens on Ethereum

- [ERC-20](#)
 - Fungible
- [ERC-721](#)
 - Non-fungible
- [ERC-1155](#)
- [ERC-777](#)
- Voluntary standards
 - Similar to HTML5



ERC-20

REQUIRED

function totalSupply() **public** view returns (uint256)

function balanceOf(address _owner) **public** view returns (uint256 balance)

function transfer(address _to, uint256 _value) **public** returns (bool success)

function transferFrom(address _from, address _to, uint256 _value) **public** returns (bool success)

function approve(address _spender, uint256 _value) **public** returns (bool success)

function allowance(address _owner, address _spender) **public** view returns (uint256 remaining)

OPTIONAL

function name() **public** view returns (string)

function symbol() **public** view returns (string)

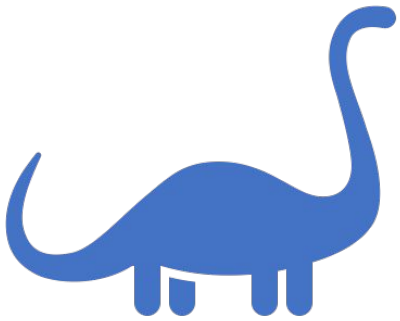
function decimals() **public** view returns (uint8)

Simple transfers



UPN

Alice	7
Bob	8
Charlie	10
Deborah	11
Elsa	4

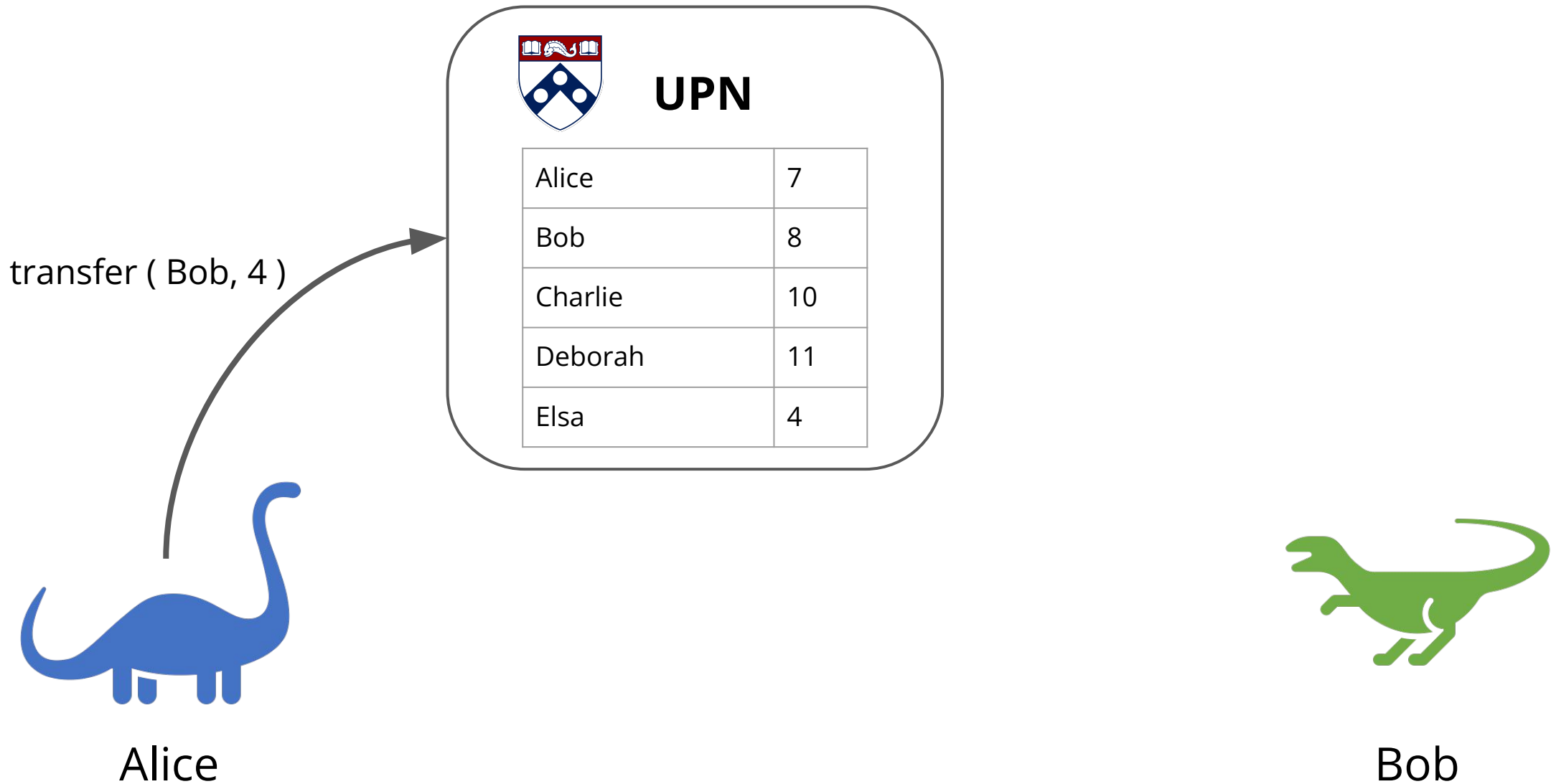


Alice

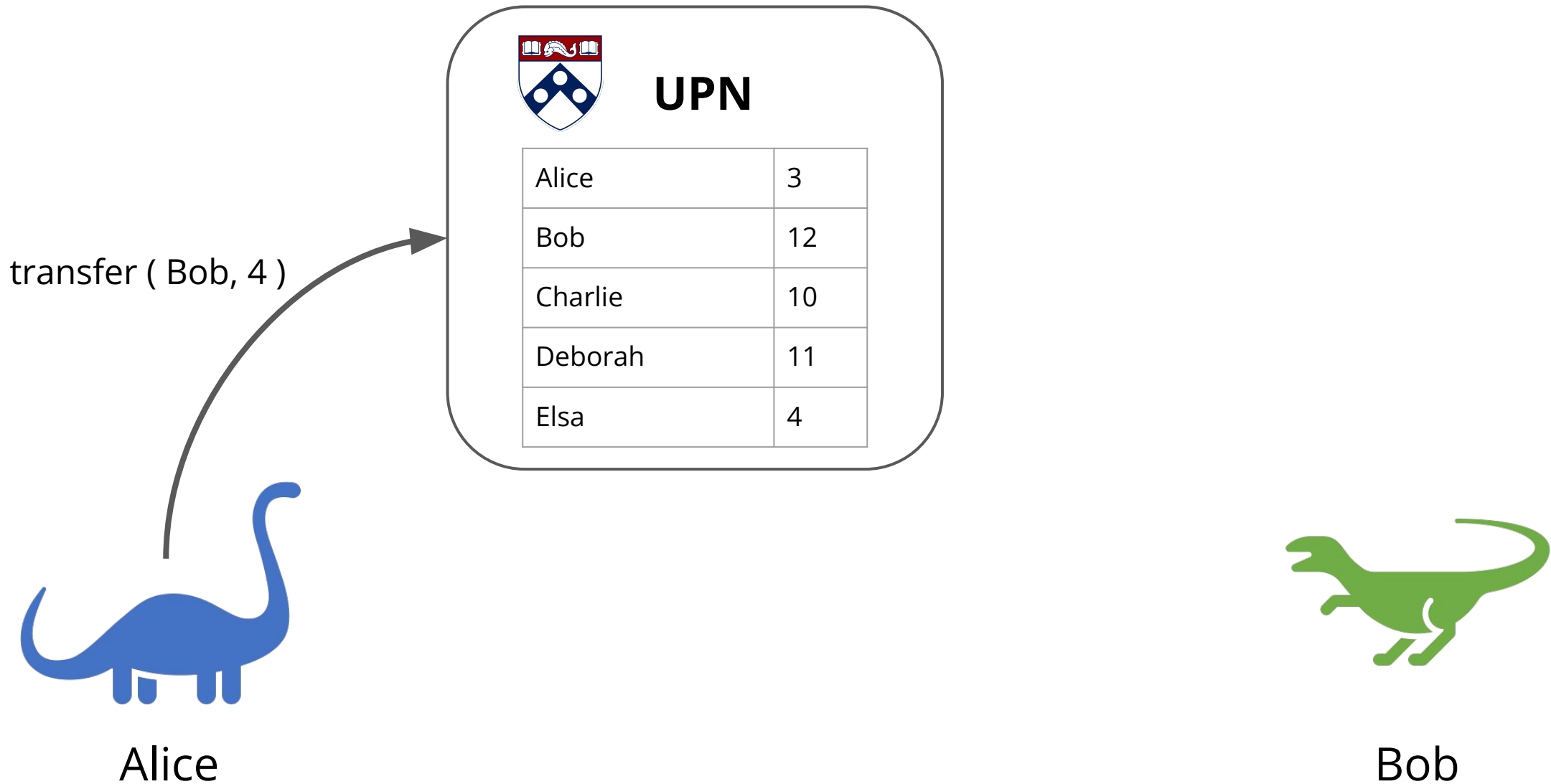


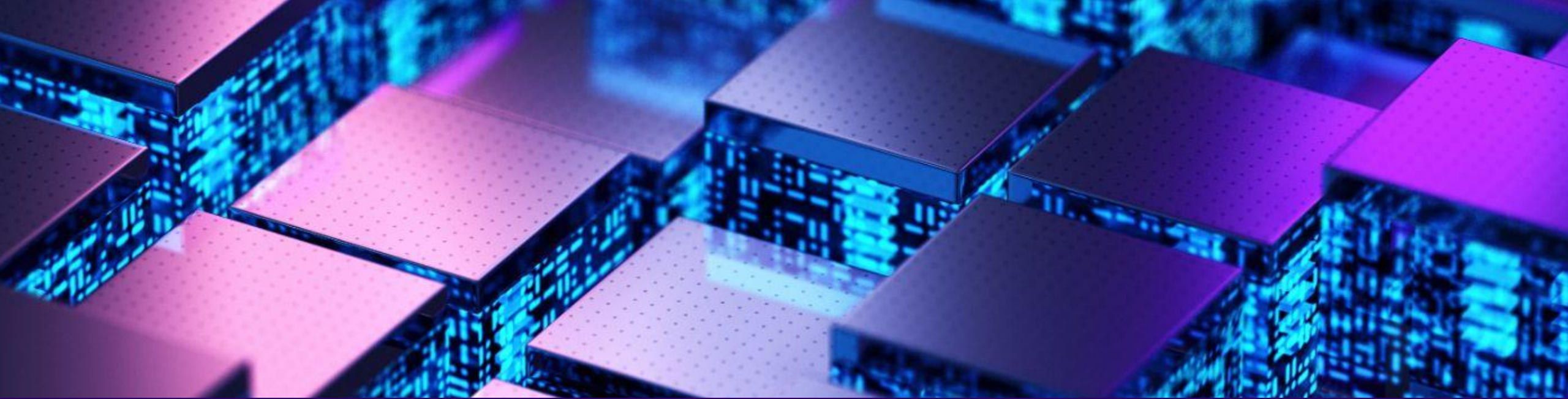
Bob

Simple transfers



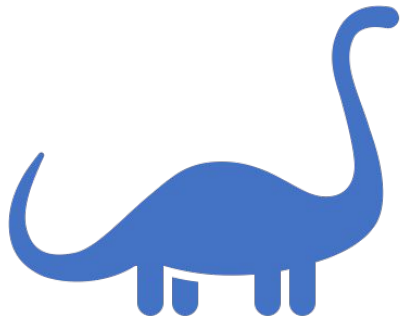
Simple transfers








Approvals

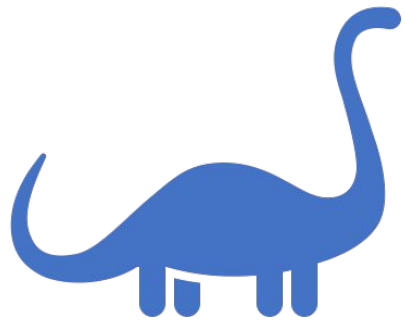
Approving transactions



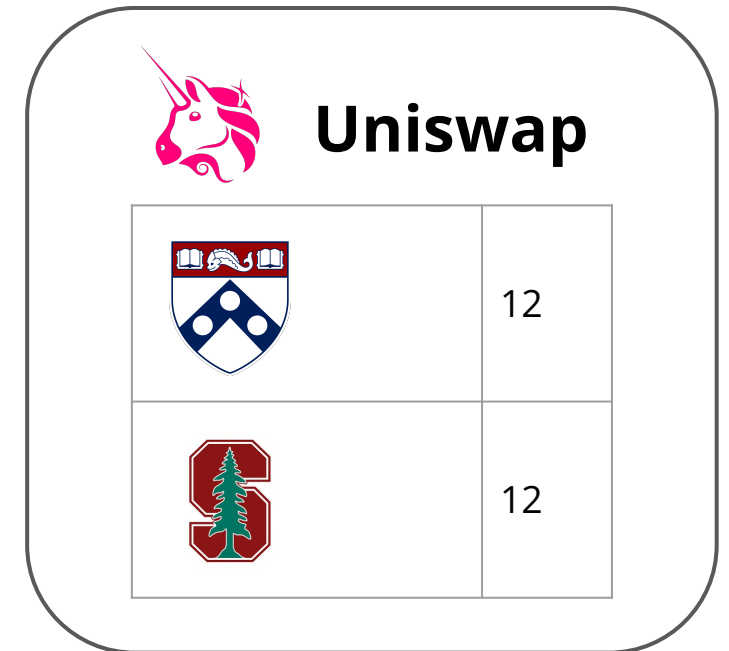
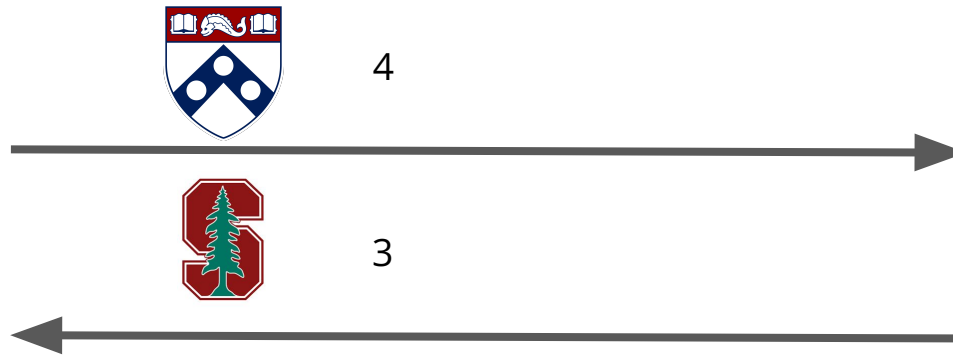
Alice

 Uniswap	
	12
	12

Approving transactions

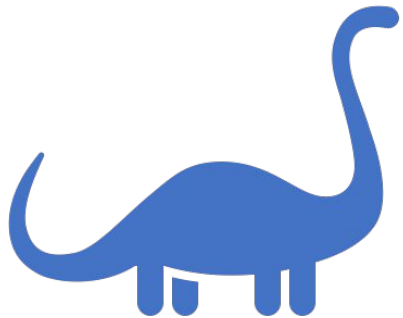


Alice



Approving transactions

transfer(, 4)



Alice





UPN

Alice	10
Bob	12
Charlie	10
Deborah	11
	12

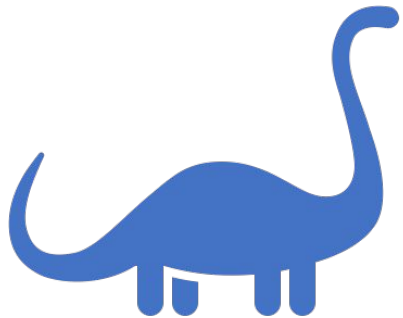


Uniswap



	12
	12

Approving transactions




transfer(, 4)



Alice

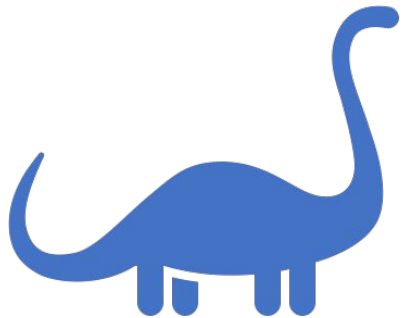
	UPN
Alice	10
Bob	12
Charlie	10
Deborah	11
	12

This doesn't call
Uniswap

	Uniswap
	12
	12

Approving transactions


approve(, 4)



Alice





UPN

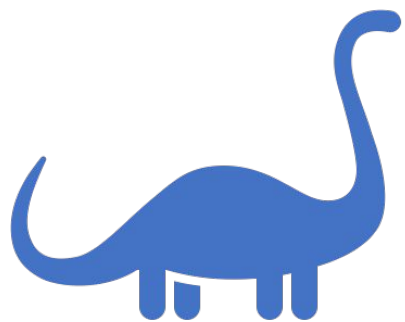
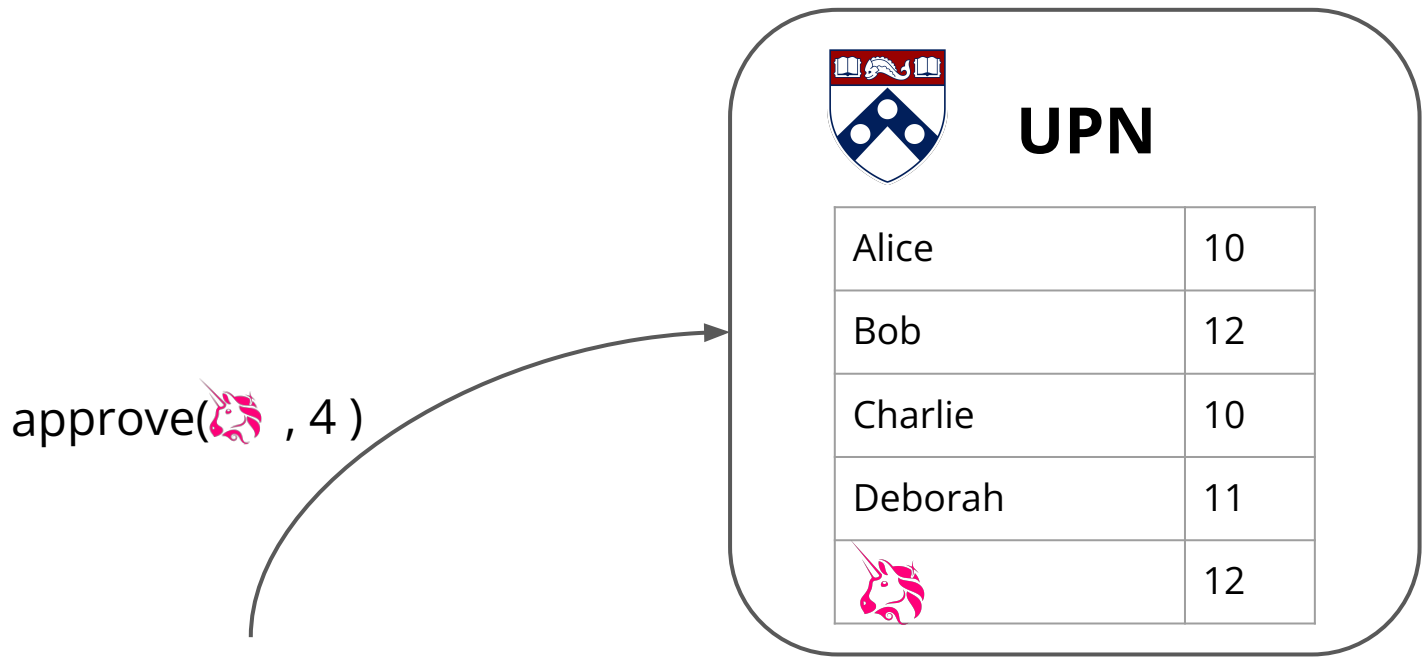
Alice	10
Bob	12
Charlie	10
Deborah	11
	12



Uniswap

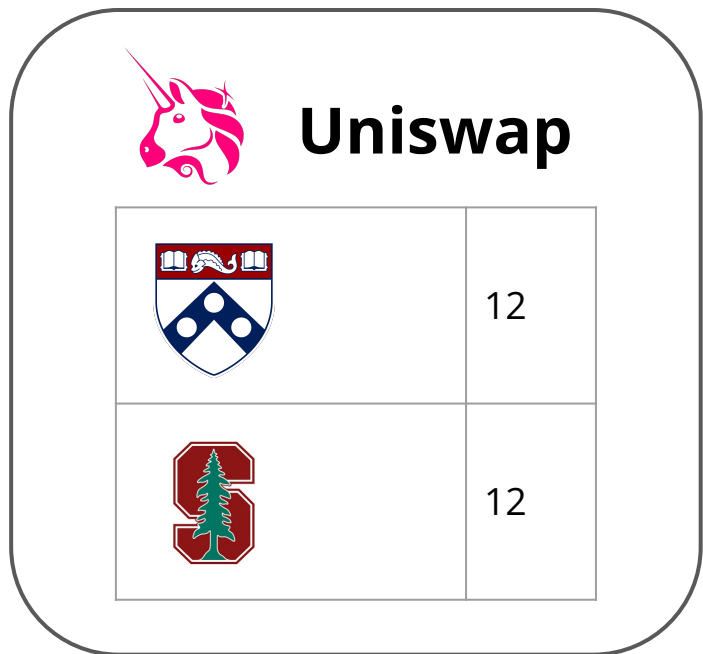
	12
	12

Approving transactions

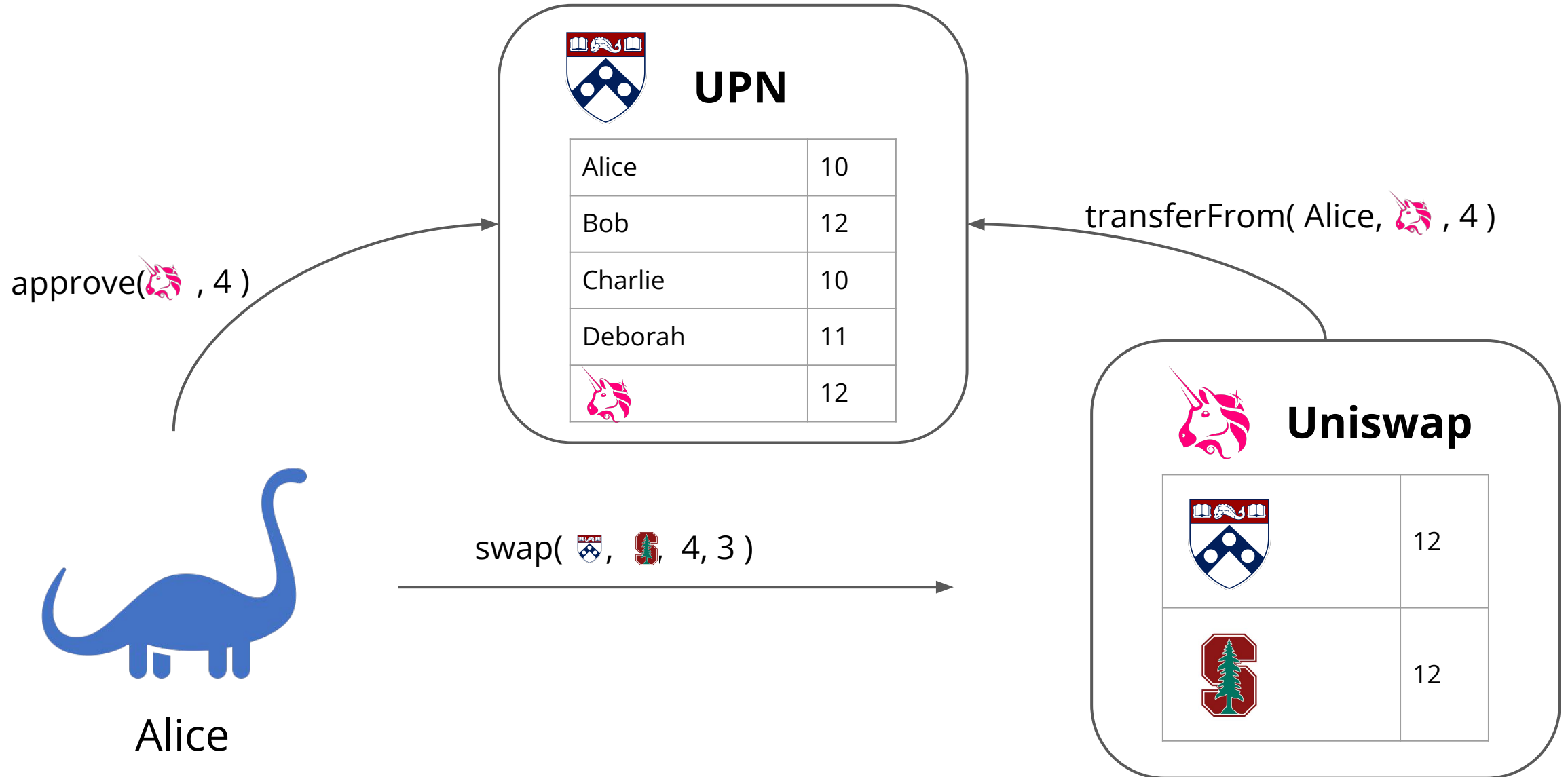


Alice

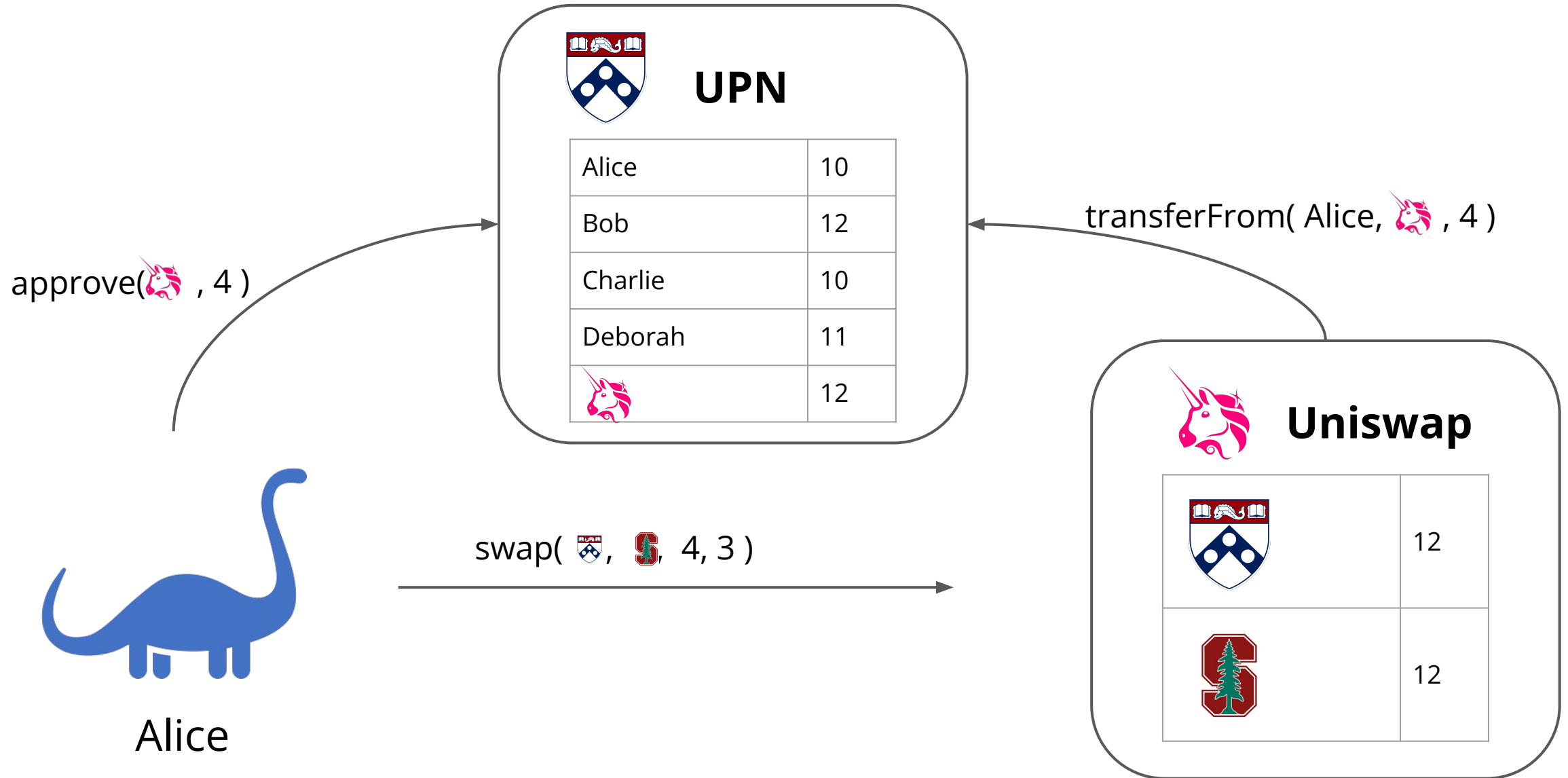
swap(, , 4, 3)



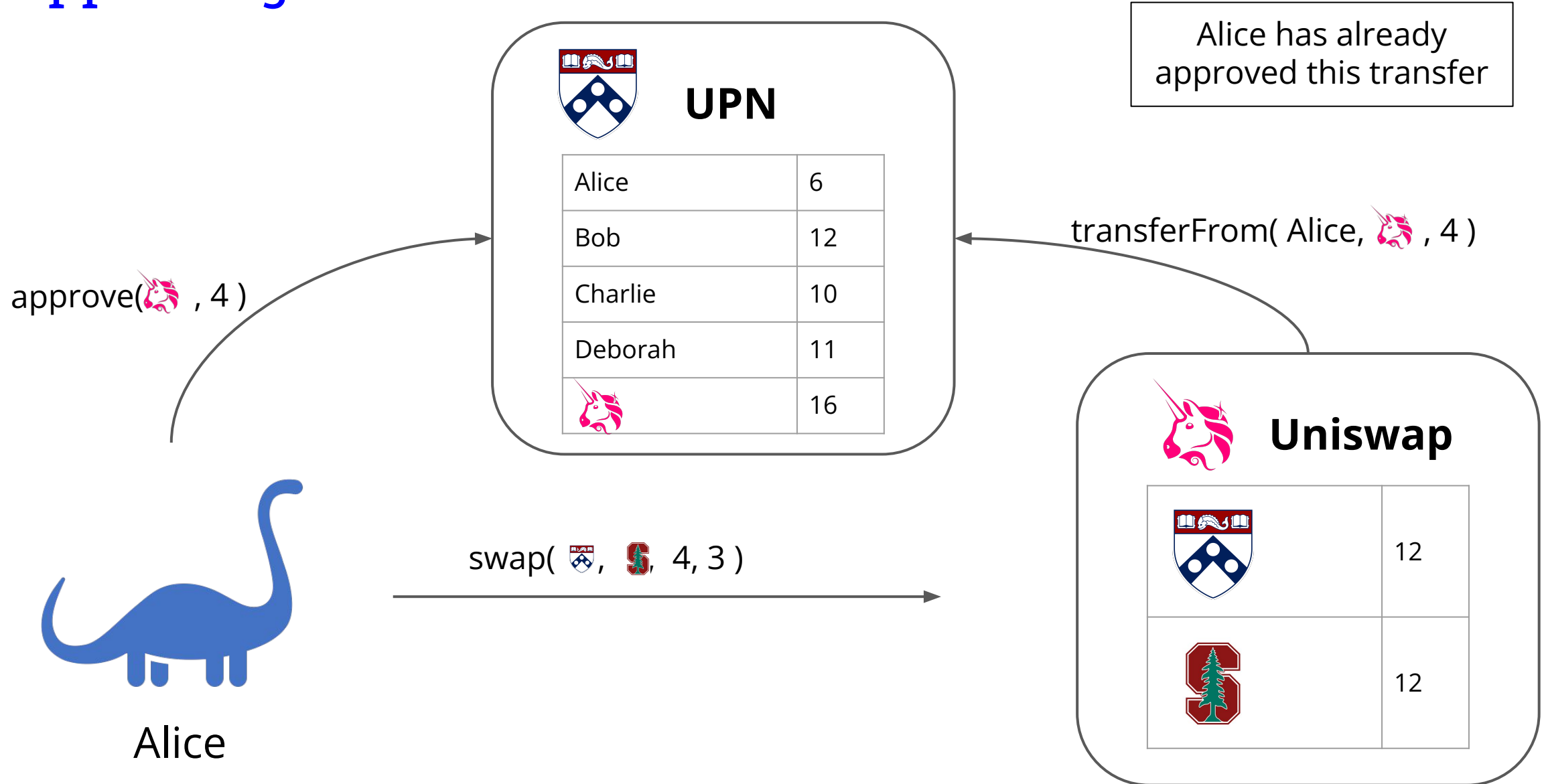
Approving transactions



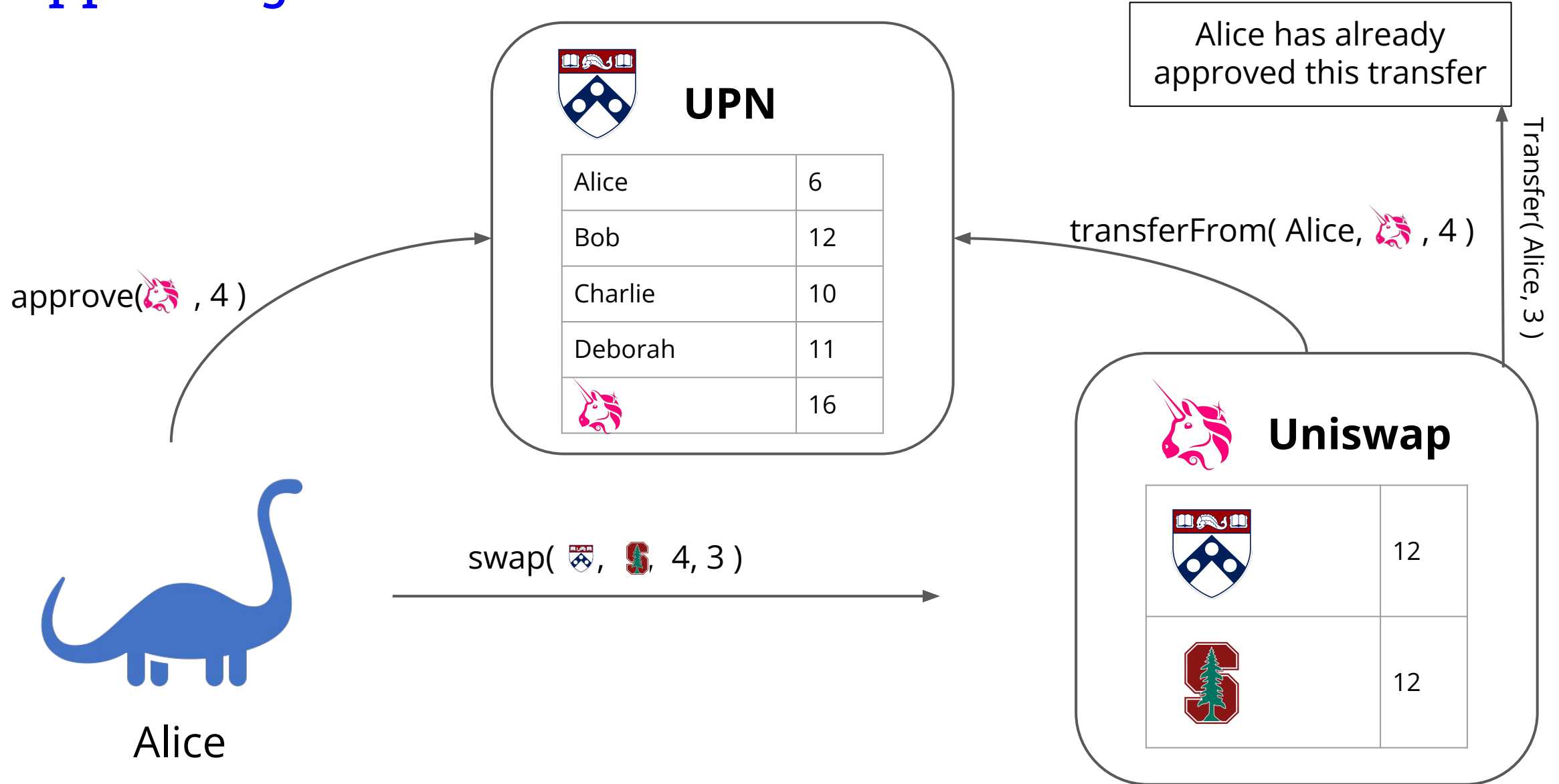
Approving transactions



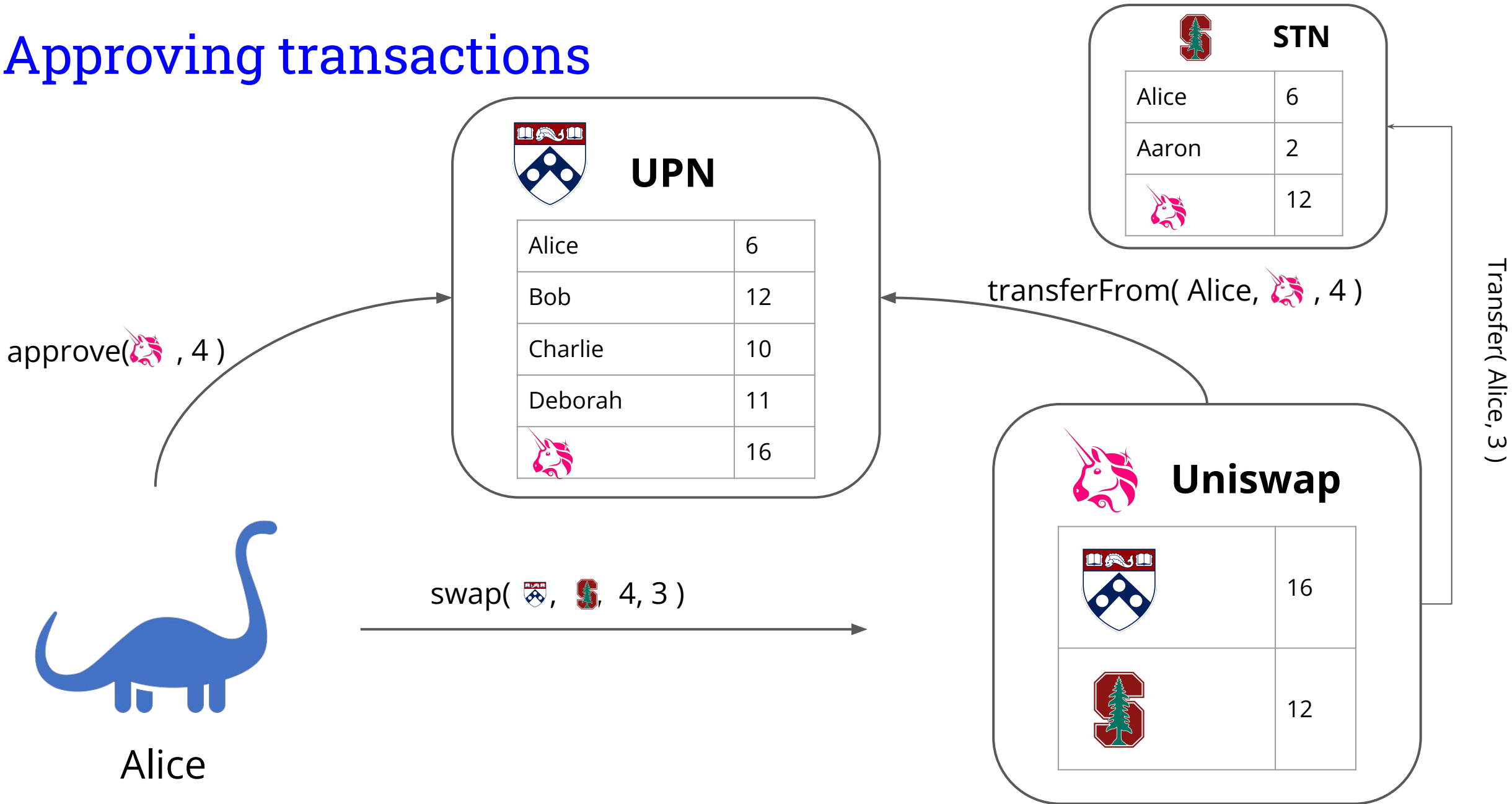
Approving transactions



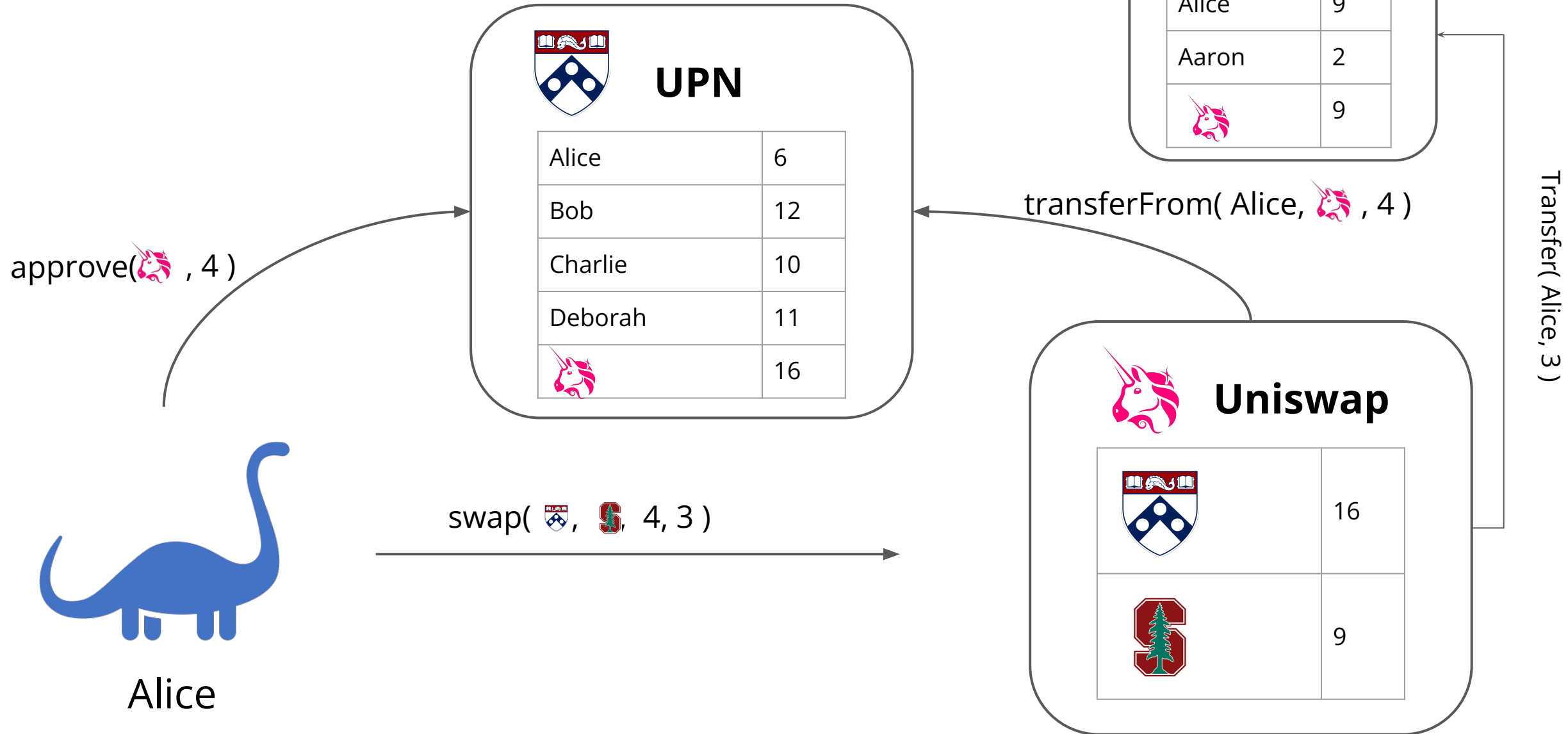
Approving transactions



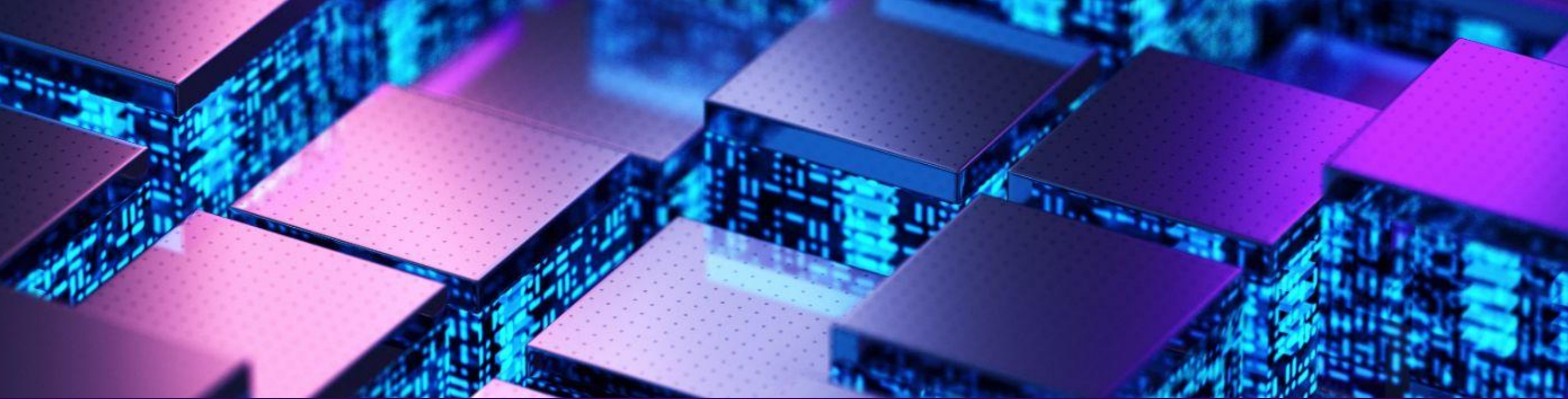
Approving transactions



Approving transactions



```
38  abstract contract ERC20 is Context, IERC20, IERC20Metadata, IERC20Errors {
39      mapping(address account => uint256) private _balances;
40
41      mapping(address account => mapping(address spender => uint256)) private _allowances;
42
43      uint256 private _totalSupply;
44
45      string private _name;
46      string private _symbol;
47  }
```



ERC20s can have other “features”

Standards

The ERC-20 standard requires a “transfer” function
It does not require that the function behave the
way you would expect

Transaction fees

- Statera token burns 1% at every transfer
 - “Every trade for Statera creates an arbitrage opportunity, which increases volume across the entire Statera ecosystem resulting in higher fees paid to liquidity providers.”
- Attackers used this “feature” to steal funds from Balancer



```
function transfer(address to, uint256 value) public returns (bool) {
    require(value <= _balances[msg.sender]);
    require(to != address(0));

    uint256 tokensToBurn = cut(value);
    uint256 tokensToTransfer = value.sub(tokensToBurn);

    _balances[msg.sender] = _balances[msg.sender].sub(value);
    _balances[to] = _balances[to].add(tokensToTransfer);

    _totalSupply = _totalSupply.sub(tokensToBurn);

    emit Transfer(msg.sender, to, tokensToTransfer);
    emit Transfer(msg.sender, address(0), tokensToBurn);
    return true;
}
```

Freezing

- Stablecoins like USDC and USDT allow admins to “freeze” user accounts

```
mapping (address => bool) public isBlackListed;  
  
function addBlackList (address _evilUser) public onlyOwner {  
    isBlackListed[_evilUser] = true;  
    AddedBlackList(_evilUser);  
}
```

Benefits

- The ERC-20 standard is necessary for DeFi
 - AMMs like Uniswap, Curve and Balancer allow users to trade ERC-20 tokens
 - Lending protocols like Aave and Compound allow users to Borrow / Lend ERC-20s



Copyright 2023 University of Pennsylvania
No reproduction or distribution without permission.