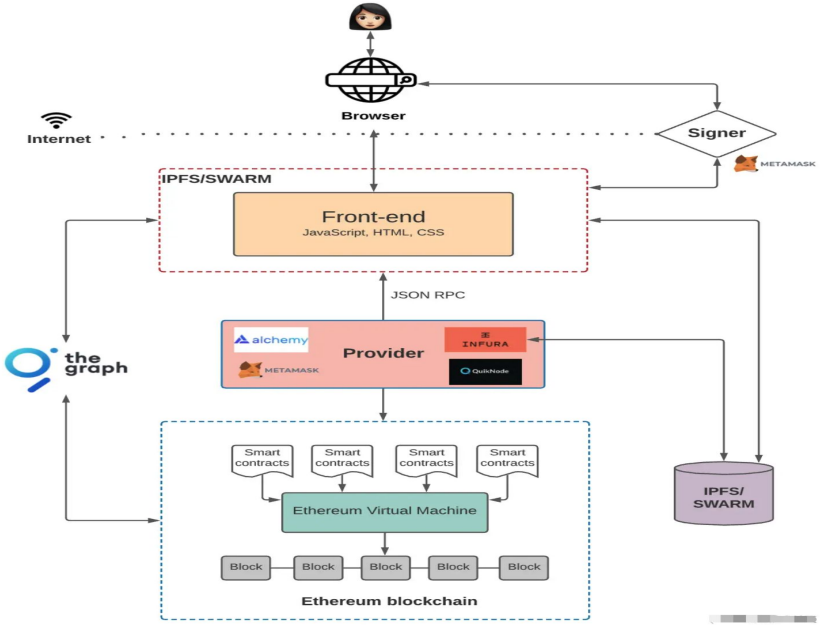
Solidity in practice

—------For Beginner

dApp - Decentralized application



requirements for developers

1. install **Node.js**
2. install **MetaMask**
3. create a **Wallet & Account**
4. testnet & rpc
    - **goerli**
    - **sepolia**
5. Faucet
    - https://goerlifaucet.com/

Tools

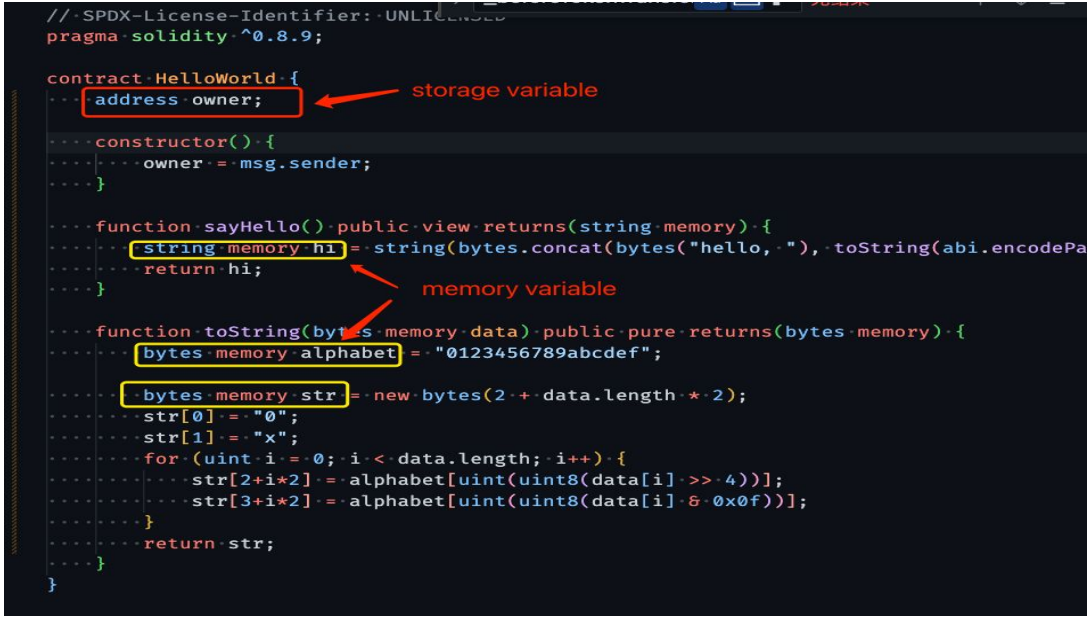
- IDE
  - remix https://remix.ethereum.org/
  - vs code
  - JetBrains
- tools
  - hardhat
  - truffle + ganache
  - foundry
- Assistants
  - block explore: **etherscan.io**
  - plugin wallet: **MetaMask**

HelloWorld

1. use **remix**
2. use **hardhat**

solidity - storage & memory variable

- storage variable
- memory variable




```
//·SPDX-License-Identifier:·UNLICENSED
pragma·solidity·^0.8.9;

contract·HelloWorld·{
····address·owner;

····constructor()·{
····owner·=·msg.sender;
····}

····function·sayHello()·public·view·returns(string·memory)·{
····string·memory·hi·=·string(bytes.concat(bytes("hello,·"),·toString(abi.encodePa
····return·hi;
····}

····function·toString(bytes·memory·data)·public·pure·returns(bytes·memory)·{
····bytes·memory·alphabet·=·"0123456789abcdef";

····bytes·memory·str·=·new·bytes(2·+·data.length·*·2);
····str[0]·=·"0";
····str[1]·=·"x";
····for·(uint·i·=·0;·i·<·data.length;·i++)·{
····str[2+i*2]·=·alphabet[uint(uint8(data[i]·>>·4))];
····str[3+i*2]·=·alphabet[uint(uint8(data[i]·&·0x0f))];
····}
····return·str;
····}
}
```

solidity - data types

- bool **true / false**
- Integers **int8/uint8, int16/uint16, …. int256 / uint256**
- Address **address(this), 0x61…DA32640,**
- bytes **bytes1, bytes2, … , bytes32**
- string
- event **event**
- enum **enum { }**
- array **uint8[]**
- mapping **mapping(uint256 => address)**
- struct **struct { }**

solidity - useful constants

- block
  - block.number
  - block.timestamp
  - ...
- msg
  - msg.sender
  - msg.value
  - msg.sig
  - msg.data
- tx
  - tx.origin
  - tx.gasprice
- address
  - address.balance
  - address.transfer / address.send
  - address.call / address.delegatecall / address.staticcall

solidity - function& modifiers

- function visibility
  - external
  - public
  - private
  - internal
- function modifiers
  - pure
  - view
  - virtual
  - override
  - payable

solidity - other keywords

- interface
  like *interface* in Java, to define methods for abstraction

- library
  a library is a set of methods, without storage variables

solidity - write test cases

- test cases is important for every line of code
- cover more scenes as possible as you can

solidity - deploy a contract

- by remix
- by scripts

solidity - inheritance

- reuse verified codes

solidity - use libraries

- business abstraction
- reuse

solidity - cross call between contracts

- conjunct with other ecosystem
- split your business to small unit, because the size of your contract is limited

solidity - eip712

All about eip712

- It is a EIP standard
- It is used for signing offchain and verifying onchain.
- More readable information for users

Scenes

- Order Book
- Conidtional authorize

solidity - upgrade a contract

Why should we upgrade a contract?

- fix bugs
- business changes
- new requirements
- ……

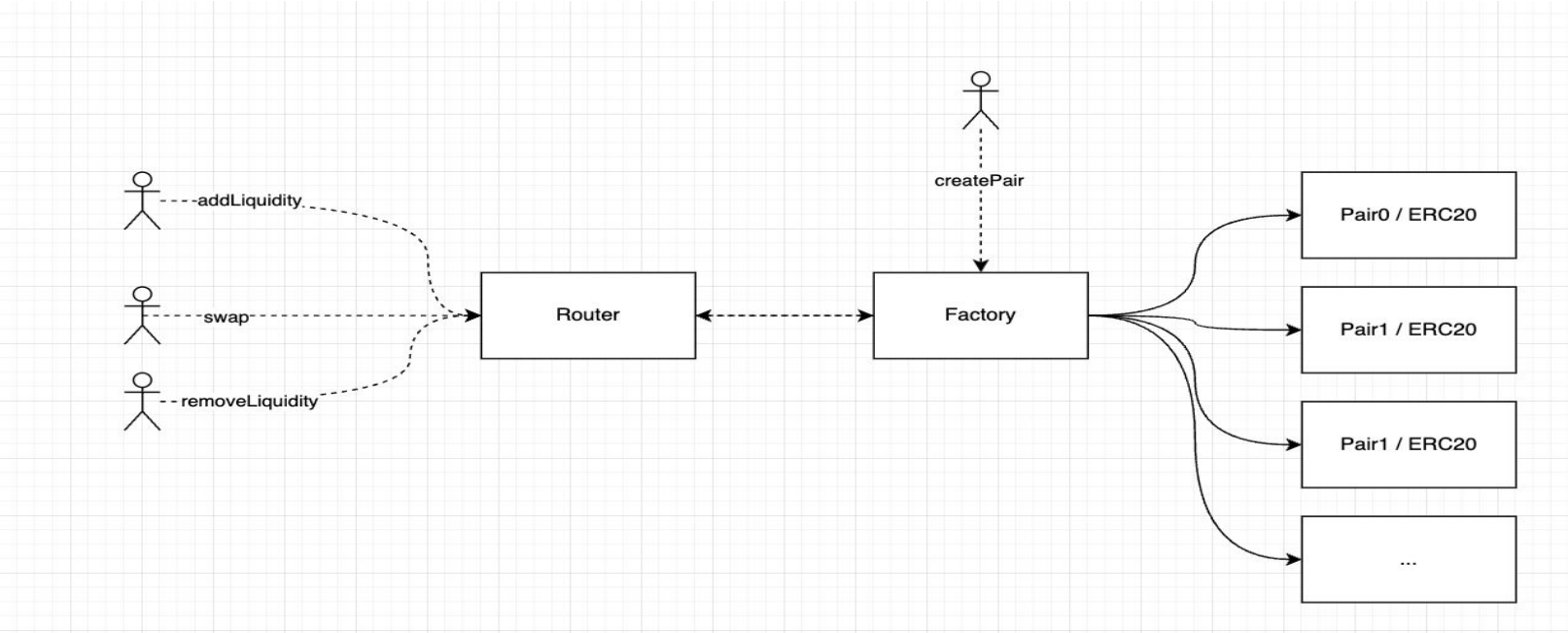
mini uniswap - functionalities

- 要实现的主要功能是什么?
- 如何与生态应用交互?
- 安全性&可扩展性?
- 可提供的附加价值?
- 社区&管理?
- 激励制度?

mini uniswap - design

1. **createPair**
   - create2
2. **addLiquidity**
3. **removeLiquidity**
   - x/y = c
4. **swap*****
   - AMM: Auto Market Maker
   - Constant Product Market Maker Model: x * y = k

mini uniswap - architecture



Dive into solidity

- Security
- EVM
- YUL: Seaport
- Cryptos
- Scenes: DAO, NFT, DID……

solidity - Q&A

Thanks