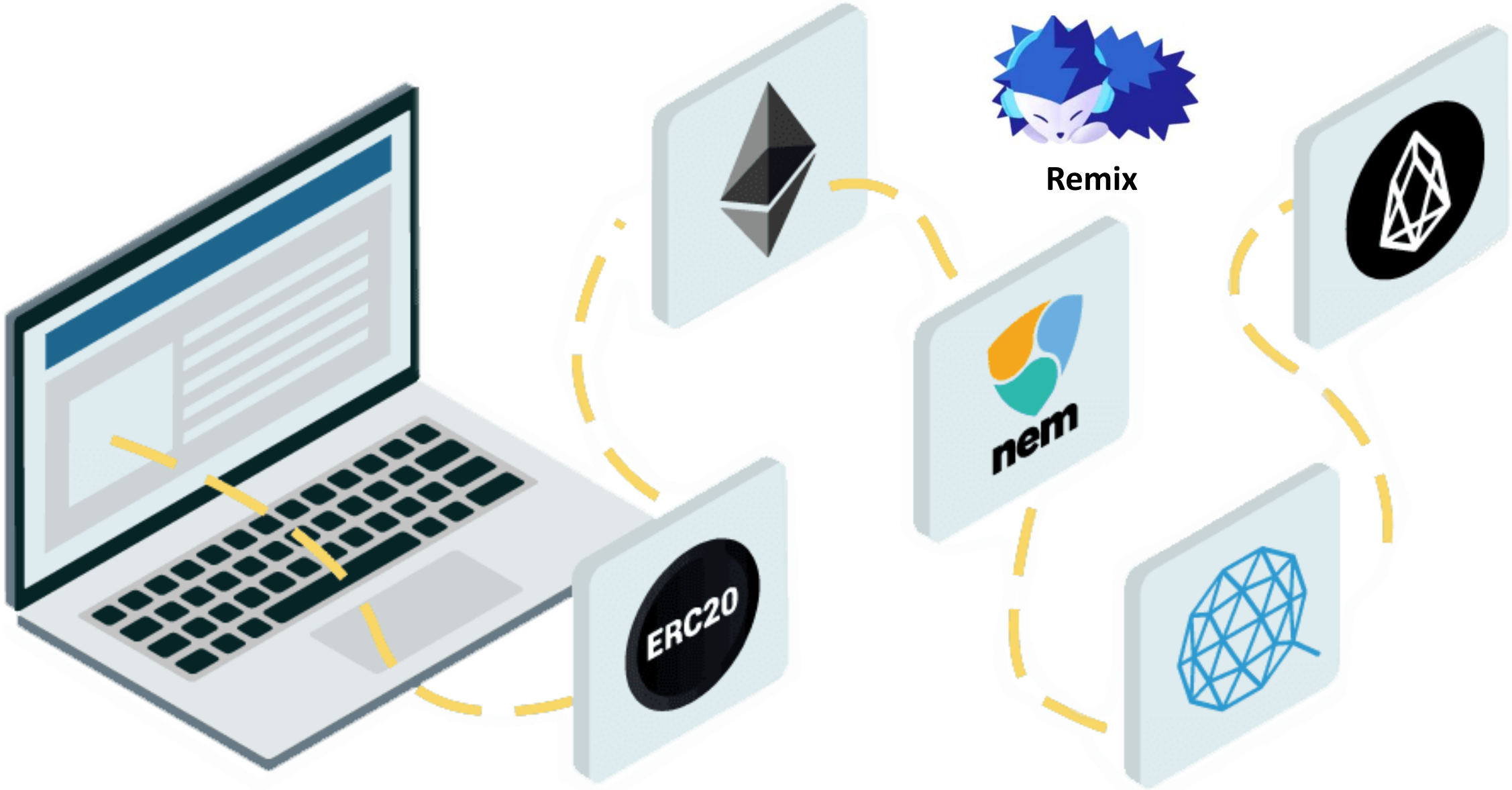
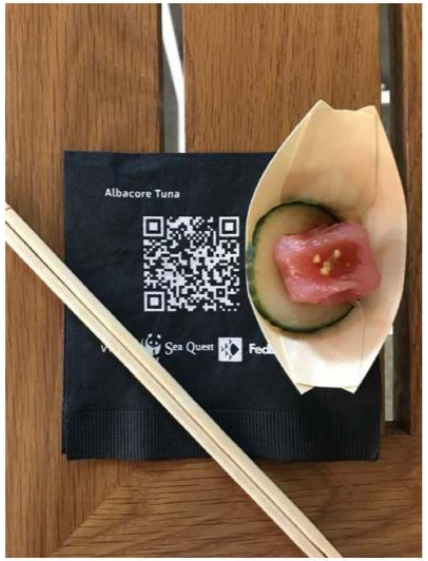
Smart Contract Development



Use Case and General Requirements

- USE CASE
  - Fish Supply Chain Management (FSCM)
- GENERAL REQUIREMENT
  - Develop a smart contract on Remix for FSCM with the following requirements:
    - ✓ Token minting, burning, buying, and transfer
    - ✓ On-chain storage of fish-related information/evidence
    - ✓ Information traceability
    - ✓ Fair trading between a seller and a buyer



Specific Development Requirements

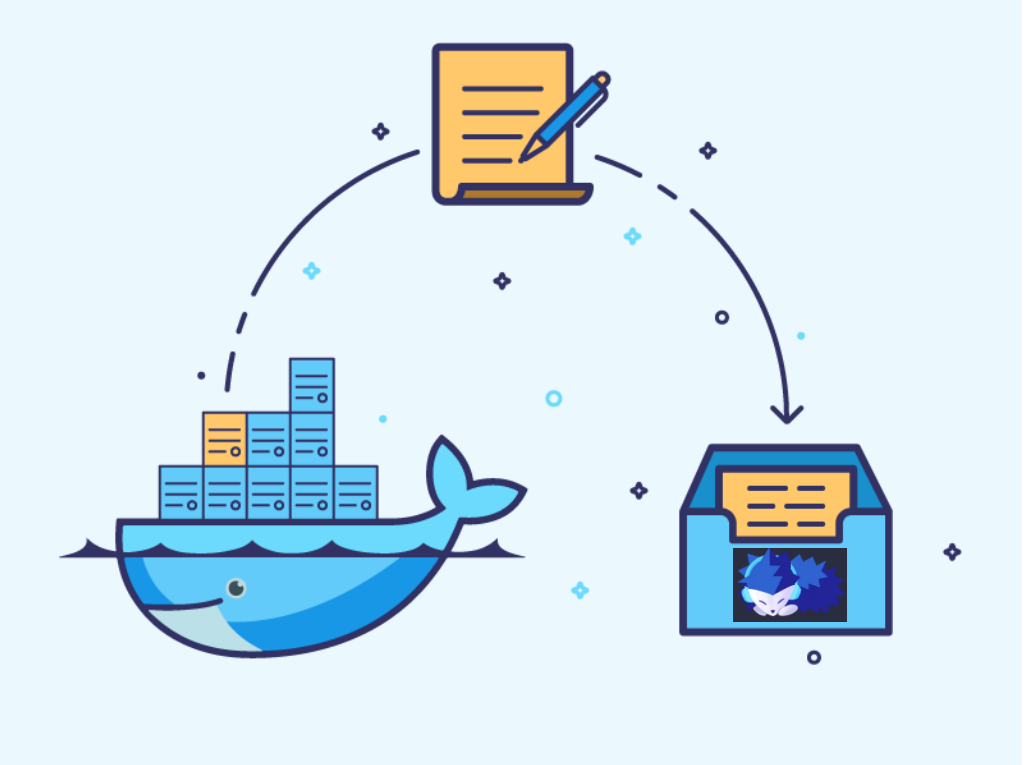
GENERAL REQUIREMENTS	SPECIFICATION
▪ Token minting, burning, buying, and transfer	▪ Define fish tokens, including name, token owner, symbol, and initial supply
	▪ Participants can buy tokens from the token owner using their account ether (100 wei = 1 fish token)
	▪ Capability of token minting and burning (limited to the token owner); token transfer between two participants
▪ On-chain storage of fish-related information	▪ Information includes time, temperature, location, species, safety (i.e., whether the temperature does not exceed the predefined threshold)...Temperature and location information should be captured/recorded over time
▪ Retrieval of fish-related information	▪ Participants can retrieve fish information based on the fish's identifier or query a fish's historical location based on the fish's identifier and a timestamp
▪ Fair trading between a seller and a buyer	▪ Assume that the price of each fish is 1 fish token, in blockchain-based fish trading. The seller can get the fish token if he provides qualified fish. The buyer can get the fish if he made the payment

Class 6: Group Exercise Report

Group members:

Smart Contract Design

- Requirement Analysis
- Preliminary Design
- Detailed Design
  - Data structure
  - Variables and functionalities



Please refer to the Solidity tutorial for the smart contract design:
https://docs.soliditylang.org/en/v0.8.9/
https://www.youtube.com/watch?v=p3C7jljTXaA&list=PLbbtODcOYIoE0D6fschNU4rqtGFRpk3ea&index=1

Some Suggestions

- Use Solidity function transfer() to transfer ether between accounts (for buying tokens)
- Define a modifier to check if a function caller is a certain account

For example, only the token owner can mint tokens, a modifier can be defined as follows

```
modifier OnlyTokenOwner() {
require(msg.sender == tokenOwner, “caller is not the token owner.”);
_;   }
```

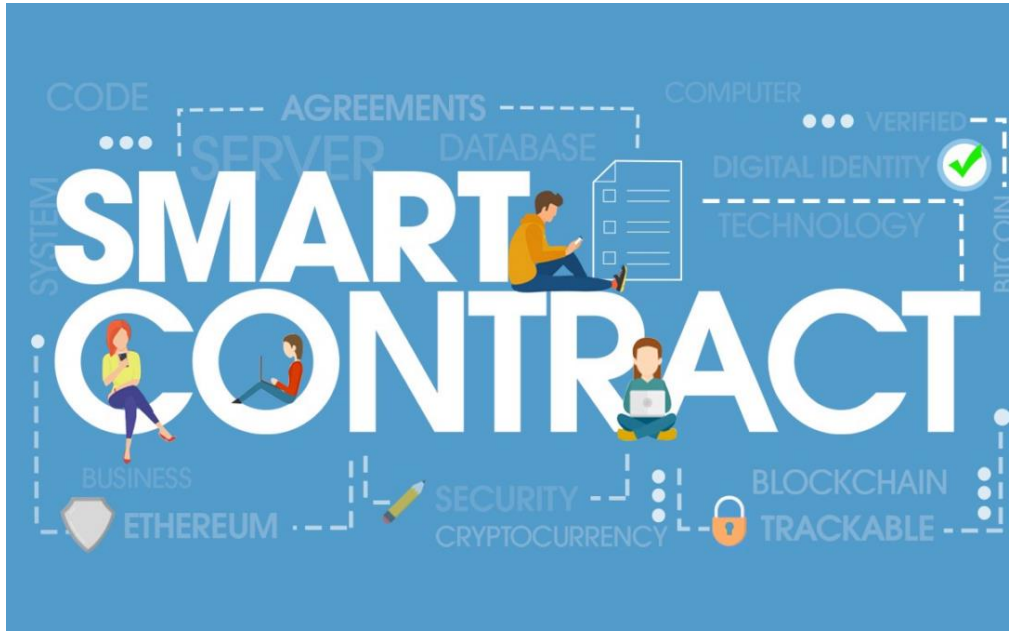
Then a mint function can be defined below

```
function mint(uint256 _amount) public OnlyTokenOwner {
….}
```

Before executing the mint function, it will check if msg.sender == tokenOwner.

Demonstration

- Smart Contract Overview
- Compile and Deploy the Contract to JavaScript VM on Remix
- Interact with the Contract



Summary and Feedback

- Challenges and problems in developing such a smart contract on Remix
- From the entire FSCM perspective, what else should be considered in terms of functionalities?


Thank you!