**How are we going to use ether.js to interact with our smart contract?**

1. First, we import the necessary library called `ethers` which allows us to interact with the 5irechain blockchain.

2. We define the configuration for the 5irechain network, specifying its name, chain ID, and the URL where we can connect to it.

3. We provide the contract ABI (Application Binary Interface), which is a description of the functions and data available in our smart contract. It helps us understand how to interact with the contract.

4. We specify the address of our smart contract we want to interact with on the 5irechain network.

5. We create a provider, which is responsible for connecting to the 5irechain network. It uses the URL we provided earlier.

6. We set up a signer, which is like a digital identity that allows us to interact with the blockchain. We use a private key to create a wallet, which is associated with that signer.

7. We connect to the smart contract using the contract's address and ABI. This allows us to call the functions defined in our smart contract.

8. Next, we define an example function called `callContractFunction` that shows how to interact with the smart contract.

9. Inside this function, we make a read-only function call to the smart contract. This means we're asking the contract for some information, and we store the result.

10. We also demonstrate calling a transaction function, which means we're making a change on the blockchain. We pass arguments to the function if needed.

11. After making the transaction, we receive a transaction hash, which is like an ID for the transaction. We log it for reference.

12. We wait for the transaction to be mined, which means it's added to a block on the blockchain and becomes a permanent part of the record.

13. Finally, we log a message indicating that the transaction has been confirmed and completed successfully.

This sets up a connection to the 5irechain network, interacts with a smart contract deployed on that network, and demonstrates how to call both read-only functions and transaction functions of the contract.