Jean-Baptiste Astruc M2 Digital Economics

Homework 2 Written Answers Exercise 2 Part C

Exercise 2 Part C

Another contract mechanism that we could think of would be the Beacon Proxy pattern. In this mechanism, a beacon holds the current implementation address so that one or multiple proxies can obtain it. Thus, whenever an update is done, there is only a need to upgrade the beacon. All the proxies pointing to that beacon will automatically go towards the new implementation. In other words, they will get the new implementation from the beacon and route their delegatecall to the new implementation.
In a Transparent model, all the proxies need to be upgraded individually by changing the implementation address. For the UUPS model, the proxy stores the implementation address as well, but the upgrade logic is held by the implementation contract.
You would use UUPS or transparent models when you only need one or few contracts to be upgraded, and the Beacon Proxy patter when you would need multiple contracts to be updated.
The Beacon Proxy pattern is very efficient at dealing with many proxies but introduces a pretty important vulnerability which corresponds to the beacon, being a single point of failure and a target for attacks.