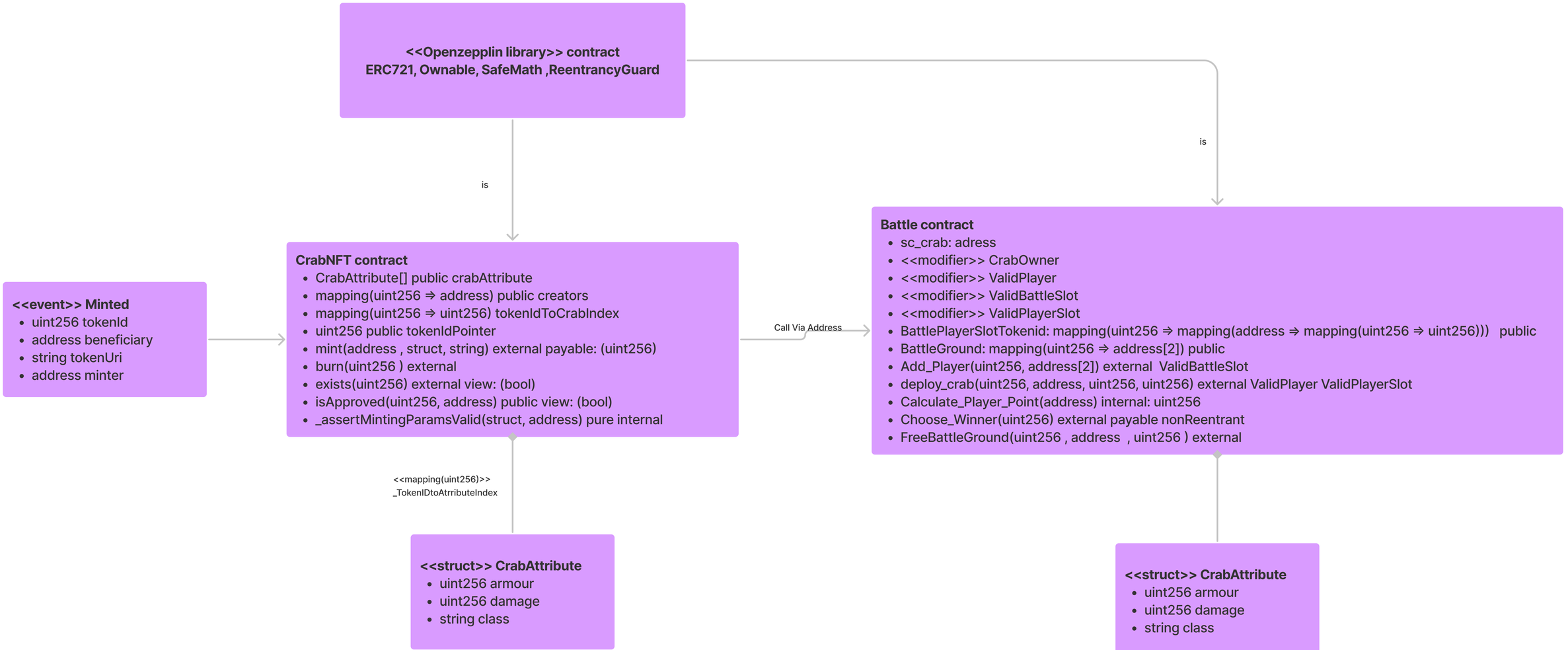
<<Openzepplin library>> contract
ERC721, Ownable, SafeMath ,ReentrancyGuard
is
is
<<event>> Minted
uint256 tokenId
address beneficiary
string tokenUri
address minter
CrabNFT contract
CrabAttribute[] public crabAttribute
mapping(uint256 ⇒ address) public creators
mapping(uint256 ⇒ uint256) tokenIdToCrabIndex
uint256 public tokenIdPointer
mint(address , struct, string) external payable: (uint256)
burn(uint256) external
exists(uint256) external view: (bool)
isApproved(uint256, address) public view: (bool)
_assertMintingParamsValid(struct, address) pure internal
Call Via Address
Battle contract
sc_crab: adress
<<modifier>> CrabOwner
<<modifier>> ValidPlayer
<<modifier>> ValidBattleSlot
<<modifier>> ValidPlayerSlot
BattlePlayerSlotTokenid: mapping(uint256 ⇒ mapping(address ⇒ mapping(uint256 ⇒ uint256))) public
BattleGround: mapping(uint256 ⇒ address[2]) public
Add_Player(uint256, address[2]) external ValidBattleSlot
deploy_crab(uint256, address, uint256, uint256) external ValidPlayer ValidPlayerSlot
Calculate_Player_Point(address) internal: uint256
Choose_Winner(uint256) external payable nonReentrant
FreeBattleGround(uint256 , address , uint256) external
<<mapping(uint256)>>
_TokenIDtoAtrributeIndex
<<struct>> CrabAttribute
uint256 armour
uint256 damage
string class
<<struct>> CrabAttribute
uint256 armour
uint256 damage
string class