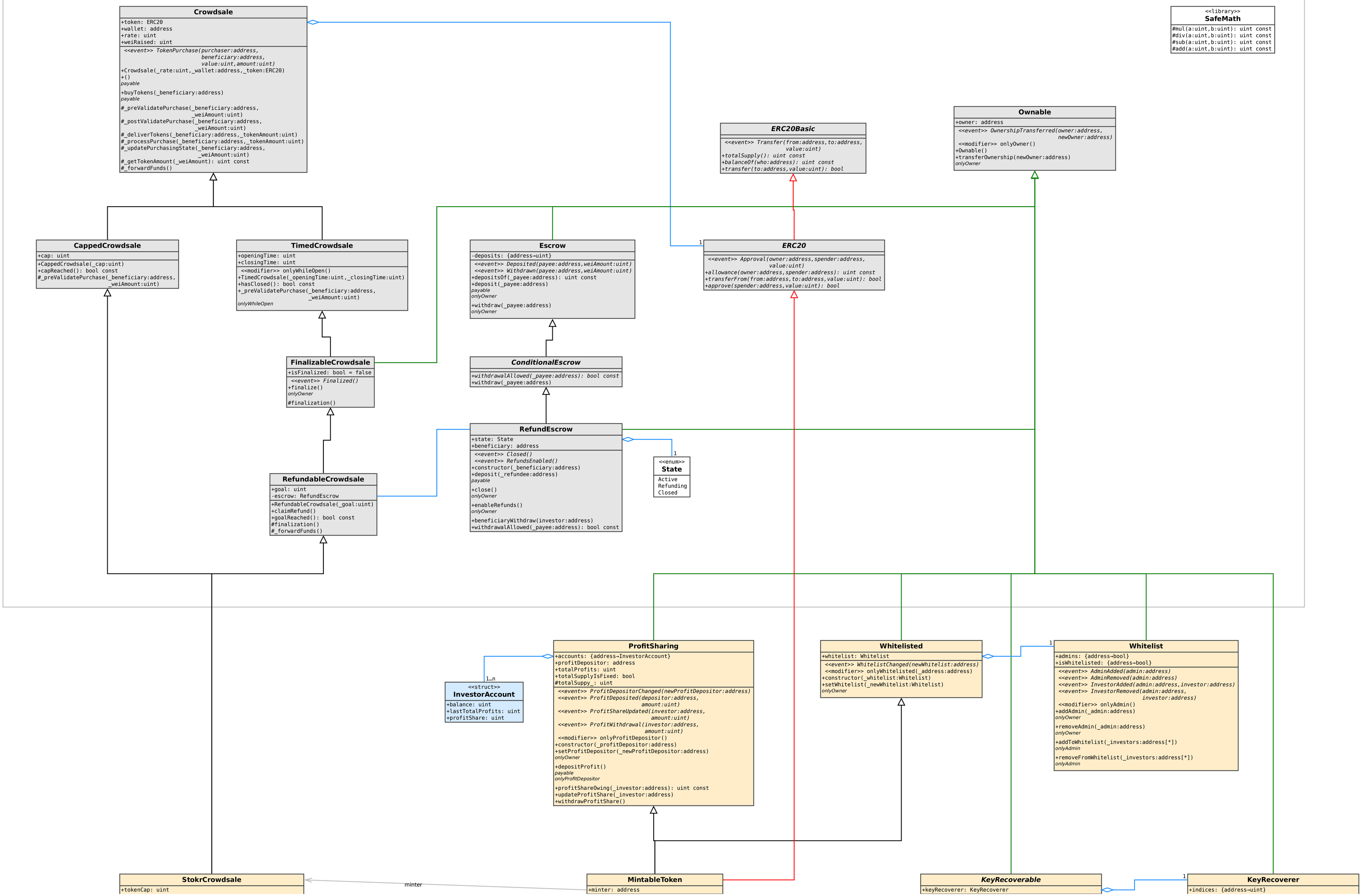
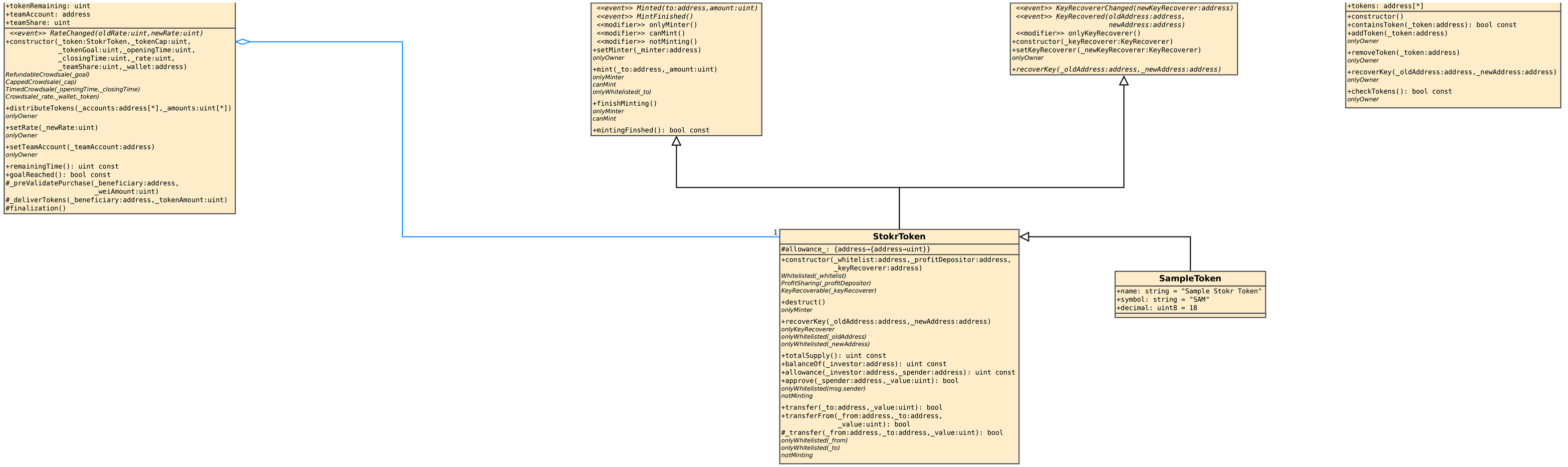
OpenZeppelin Solidity Framework v1.12.0




+tokenRemaining: uint
+teamAccount: address
+teamShare: uint
<<event>> RateChanged(oldRate:uint,newRate:uint)
+constructor(_token:StokrToken,_tokenCap:uint,
_tokenGoal:uint,_openingTime:uint,
_closingTime:uint,_rate:uint,
_teamShare:uint,_wallet:address)
RefundableCrowdsale(_goal)
CappedCrowdsale(_cap)
TimedCrowdsale(_openingTime,_closingTime)
Crowdsale(_rate,_wallet_token)
+distributeTokens(_accounts:address[*],_amounts:uint[*])
onlyOwner
+setRate(_newRate:uint)
onlyOwner
+setTeamAccount(_teamAccount:address)
onlyOwner
+remainingTime(): uint const
+goalReached(): bool const
#_preValidatePurchase(_beneficiary:address,
_weiAmount:uint)
#_deliverTokens(_beneficiary:address,_tokenAmount:uint)
#finalization()
1
<<event>> Minted(to:address,amount:uint)
<<event>> MintFinished()
<<modifier>> onlyMinter()
<<modifier>> canMint()
<<modifier>> notMinting()
+setMinter(_minter:address)
onlyOwner
+mint(_to:address,_amount:uint)
onlyMinter
canMint
onlyWhitelisted(_to)
+finishMinting()
onlyMinter
canMint
+mintingFinshed(): bool const
<<event>> KeyRecovererChanged(newKeyRecoverer:address)
<<event>> KeyRecovered(oldAddress:address,
newAddress:address)
<<modifier>> onlyKeyRecoverer()
+constructor(_keyRecoverer:KeyRecoverer)
+setKeyRecoverer(_newKeyRecoverer:KeyRecoverer)
onlyOwner
+recoverKey(_oldAddress:address,_newAddress:address)
+tokens: address[*]
+constructor()
+containsToken(_token:address): bool const
+addToken(_token:address)
onlyOwner
+removeToken(_token:address)
onlyOwner
+recoverKey(_oldAddress:address,_newAddress:address)
onlyOwner
+checkTokens(): bool const
onlyOwner
StokrToken
#allowance_: {address→{address→uint}}
+constructor(_whitelist:address,_profitDepositor:address,
_keyRecoverer:address)
Whitelisted(_whitelist)
ProfitSharing(_profitDepositor)
KeyRecoverable(_keyRecoverer)
+destruct()
onlyMinter
+recoverKey(_oldAddress:address,_newAddress:address)
onlyKeyRecoverer
onlyWhitelisted(_oldAddress)
onlyWhitelisted(_newAddress)
+totalSupply(): uint const
+balanceOf(_investor:address): uint const
+allowance(_investor:address,_spender:address): uint const
+approve(_spender:address,_value:uint): bool
onlyWhitelisted(msg.sender)
notMinting
+transfer(_to:address,_value:uint): bool
+transferFrom(_from:address,_to:address,
_value:uint): bool
#_transfer(_from:address,_to:address,_value:uint): bool
onlyWhitelisted(_from)
onlyWhitelisted(_to)
notMinting
SampleToken
+name: string = "Sample Stokr Token"
+symbol: string = "SAM"
+decimal: uint8 = 18