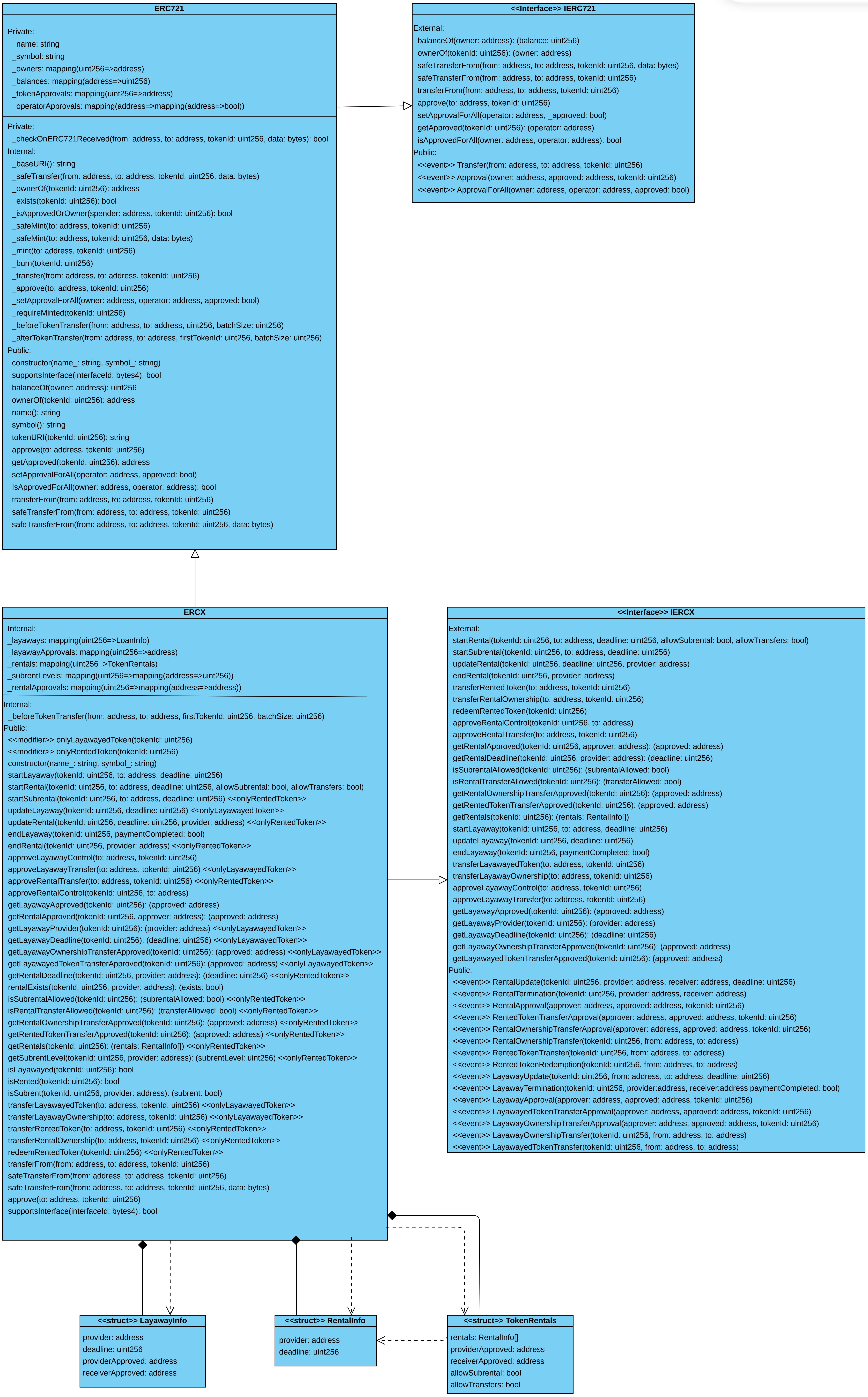

```
ERC721

Private:
  _name: string
  _symbol: string
  _owners: mapping(uint256=>address)
  _balances: mapping(address=>uint256)
  _tokenApprovals: mapping(uint256=>address)
  _operatorApprovals: mapping(address=>mapping(address=>bool))

Private:
  _checkOnERC721Received(from: address, to: address, tokenId: uint256, data: bytes): bool
Internal:
  _baseURI(): string
  _safeTransfer(from: address, to: address, tokenId: uint256, data: bytes)
  _ownerOf(tokenId: uint256): address
  _exists(tokenId: uint256): bool
  _isApprovedOrOwner(spender: address, tokenId: uint256): bool
  _safeMint(to: address, tokenId: uint256)
  _safeMint(to: address, tokenId: uint256, data: bytes)
  _mint(to: address, tokenId: uint256)
  _burn(tokenId: uint256)
  _transfer(from: address, to: address, tokenId: uint256)
  _approve(to: address, tokenId: uint256)
  _setApprovalForAll(owner: address, operator: address, approved: bool)
  _requireMinted(tokenId: uint256)
  _beforeTokenTransfer(from: address, to: address, uint256, batchSize: uint256)
  _afterTokenTransfer(from: address, to: address, firstTokenId: uint256, batchSize: uint256)
Public:
  constructor(name_: string, symbol_: string)
  supportsInterface(interfaceId: bytes4): bool
  balanceOf(owner: address): uint256
  ownerOf(tokenId: uint256): address
  name(): string
  symbol(): string
  tokenURI(tokenId: uint256): string
  approve(to: address, tokenId: uint256)
  getApproved(tokenId: uint256): address
  setApprovalForAll(operator: address, approved: bool)
  IsApprovedForAll(owner: address, operator: address): bool
  transferFrom(from: address, to: address, tokenId: uint256)
  safeTransferFrom(from: address, to: address, tokenId: uint256)
  safeTransferFrom(from: address, to: address, tokenId: uint256, data: bytes)
```

```
<<Interface>> IERC721

External:
  balanceOf(owner: address): (balance: uint256)
  ownerOf(tokenId: uint256): (owner: address)
  safeTransferFrom(from: address, to: address, tokenId: uint256, data: bytes)
  safeTransferFrom(from: address, to: address, tokenId: uint256)
  transferFrom(from: address, to: address, tokenId: uint256)
  approve(to: address, tokenId: uint256)
  setApprovalForAll(operator: address, _approved: bool)
  getApproved(tokenId: uint256): (operator: address)
  isApprovedForAll(owner: address, operator: address): bool
Public:
  <<event>> Transfer(from: address, to: address, tokenId: uint256)
  <<event>> Approval(owner: address, approved: address, tokenId: uint256)
  <<event>> ApprovalForAll(owner: address, operator: address, approved: bool)
```

```
ERCX

 Internal:
 _layaways: mapping(uint256=>LoanInfo)
 _layawayApprovals: mapping(uint256=>address)
 _rentals: mapping(uint256=>TokenRentals)
 _subrentLevels: mapping(uint256=>mapping(address=>uint256))
 _rentalApprovals: mapping(uint256=>mapping(address=>address))

Internal:
  _beforeTokenTransfer(from: address, to: address, firstTokenId: uint256, batchSize: uint256)
Public:
  <<modifier>> onlyLayawayedToken(tokenId: uint256)
  <<modifier>> onlyRentedToken(tokenId: uint256)
  constructor(name_: string, symbol_: string)
  startLayaway(tokenId: uint256, to: address, deadline: uint256)
  startRental(tokenId: uint256, to: address, deadline: uint256, allowSubrental: bool, allowTransfers: bool)
  startSubrental(tokenId: uint256, to: address, deadline: uint256) <<onlyRentedToken>>
  updateLayaway(tokenId: uint256, deadline: uint256) <<onlyLayawayedToken>>
  updateRental(tokenId: uint256, deadline: uint256, provider: address) <<onlyRentedToken>>
  endLayaway(tokenId: uint256, paymentCompleted: bool)
  endRental(tokenId: uint256, provider: address) <<onlyRentedToken>>
  approveLayawayControl(to: address, tokenId: uint256)
  approveLayawayTransfer(to: address, tokenId: uint256) <<onlyLayawayedToken>>
  approveRentalTransfer(to: address, tokenId: uint256) <<onlyRentedToken>>
  approveRentalControl(tokenId: uint256, to: address)
  getLayawayApproved(tokenId: uint256): (approved: address)
  getRentalApproved(tokenId: uint256, approver: address): (approved: address)
  getLayawayProvider(tokenId: uint256): (provider: address) <<onlyLayawayedToken>>
  getLayawayDeadline(tokenId: uint256): (deadline: uint256) <<onlyLayawayedToken>>
  getLayawayOwnershipTransferApproved(tokenId: uint256): (approved: address) <<onlyLayawayedToken>>
  getLayawayedTokenTransferApproved(tokenId: uint256): (approved: address) <<onlyLayawayedToken>>
  getRentalDeadline(tokenId: uint256, provider: address): (deadline: uint256) <<onlyRentedToken>>
  rentalExists(tokenId: uint256, provider: address): (exists: bool)
  isSubrentalAllowed(tokenId: uint256): (subrentalAllowed: bool) <<onlyRentedToken>>
  isRentalTransferAllowed(tokenId: uint256): (transferAllowed: bool) <<onlyRentedToken>>
  getRentalOwnershipTransferApproved(tokenId: uint256): (approved: address) <<onlyRentedToken>>
  getRentedTokenTransferApproved(tokenId: uint256): (approved: address) <<onlyRentedToken>>
  getRentals(tokenId: uint256): (rentals: RentalInfo[]) <<onlyRentedToken>>
  getSubrentLevel(tokenId: uint256, provider: address): (subrentLevel: uint256) <<onlyRentedToken>>
  isLayawayed(tokenId: uint256): bool
  isRented(tokenId: uint256): bool
  isSubrent(tokenId: uint256, provider: address): (subrent: bool)
  transferLayawayedToken(to: address, tokenId: uint256) <<onlyLayawayedToken>>
  transferLayawayOwnership(to: address, tokenId: uint256) <<onlyLayawayedToken>>
  transferRentedToken(to: address, tokenId: uint256) <<onlyRentedToken>>
  transferRentalOwnership(to: address, tokenId: uint256) <<onlyRentedToken>>
  redeemRentedToken(tokenId: uint256) <<onlyRentedToken>>
  transferFrom(from: address, to: address, tokenId: uint256)
  safeTransferFrom(from: address, to: address, tokenId: uint256)
  safeTransferFrom(from: address, to: address, tokenId: uint256, data: bytes)
  approve(to: address, tokenId: uint256)
  supportsInterface(interfaceId: bytes4): bool
```

```
<<Interface>> IERCX

External:
  startRental(tokenId: uint256, to: address, deadline: uint256, allowSubrental: bool, allowTransfers: bool)
  startSubrental(tokenId: uint256, to: address, deadline: uint256)
  updateRental(tokenId: uint256, deadline: uint256, provider: address)
  endRental(tokenId: uint256, provider: address)
  transferRentedToken(to: address, tokenId: uint256)
  transferRentalOwnership(to: address, tokenId: uint256)
  redeemRentedToken(tokenId: uint256)
  approveRentalControl(tokenId: uint256, to: address)
  approveRentalTransfer(to: address, tokenId: uint256)
  getRentalApproved(tokenId: uint256, approver: address): (approved: address)
  getRentalDeadline(tokenId: uint256, provider: address): (deadline: uint256)
  isSubrentalAllowed(tokenId: uint256): (subrentalAllowed: bool)
  isRentalTransferAllowed(tokenId: uint256): (transferAllowed: bool)
  getRentalOwnershipTransferApproved(tokenId: uint256): (approved: address)
  getRentedTokenTransferApproved(tokenId: uint256): (approved: address)
  getRentals(tokenId: uint256): (rentals: RentalInfo[])
  startLayaway(tokenId: uint256, to: address, deadline: uint256)
  updateLayaway(tokenId: uint256, deadline: uint256)
  endLayaway(tokenId: uint256, paymentCompleted: bool)
  transferLayawayedToken(to: address, tokenId: uint256)
  transferLayawayOwnership(to: address, tokenId: uint256)
  approveLayawayControl(to: address, tokenId: uint256)
  approveLayawayTransfer(to: address, tokenId: uint256)
  getLayawayApproved(tokenId: uint256): (approved: address)
  getLayawayProvider(tokenId: uint256): (provider: address)
  getLayawayDeadline(tokenId: uint256): (deadline: uint256)
  getLayawayOwnershipTransferApproved(tokenId: uint256): (approved: address)
  getLayawayedTokenTransferApproved(tokenId: uint256): (approved: address)
Public:
  <<event>> RentalUpdate(tokenId: uint256, provider: address, receiver: address, deadline: uint256)
  <<event>> RentalTermination(tokenId: uint256, provider: address, receiver: address)
  <<event>> RentalApproval(approver: address, approved: address, tokenId: uint256)
  <<event>> RentedTokenTransferApproval(approver: address, approved: address, tokenId: uint256)
  <<event>> RentalOwnershipTransferApproval(approver: address, approved: address, tokenId: uint256)
  <<event>> RentalOwnershipTransfer(tokenId: uint256, from: address, to: address)
  <<event>> RentedTokenTransfer(tokenId: uint256, from: address, to: address)
  <<event>> RentedTokenRedemption(tokenId: uint256, from: address, to: address)
  <<event>> LayawayUpdate(tokenId: uint256, from: address, to: address, deadline: uint256)
  <<event>> LayawayTermination(tokenId: uint256, provider:address, receiver:address paymentCompleted: bool)
  <<event>> LayawayApproval(approver: address, approved: address, tokenId: uint256)
  <<event>> LayawayedTokenTransferApproval(approver: address, approved: address, tokenId: uint256)
  <<event>> LayawayOwnershipTransferApproval(approver: address, approved: address, tokenId: uint256)
  <<event>> LayawayOwnershipTransfer(tokenId: uint256, from: address, to: address)
  <<event>> LayawayedTokenTransfer(tokenId: uint256, from: address, to: address)
```

```
<<struct>> LayawayInfo

provider: address
deadline: uint256
providerApproved: address
receiverApproved: address
```

```
<<struct>> RentalInfo

provider: address
deadline: uint256
```

```
<<struct>> TokenRentals

rentals: RentalInfo[]
providerApproved: address
receiverApproved: address
allowSubrental: bool
allowTransfers: bool
```