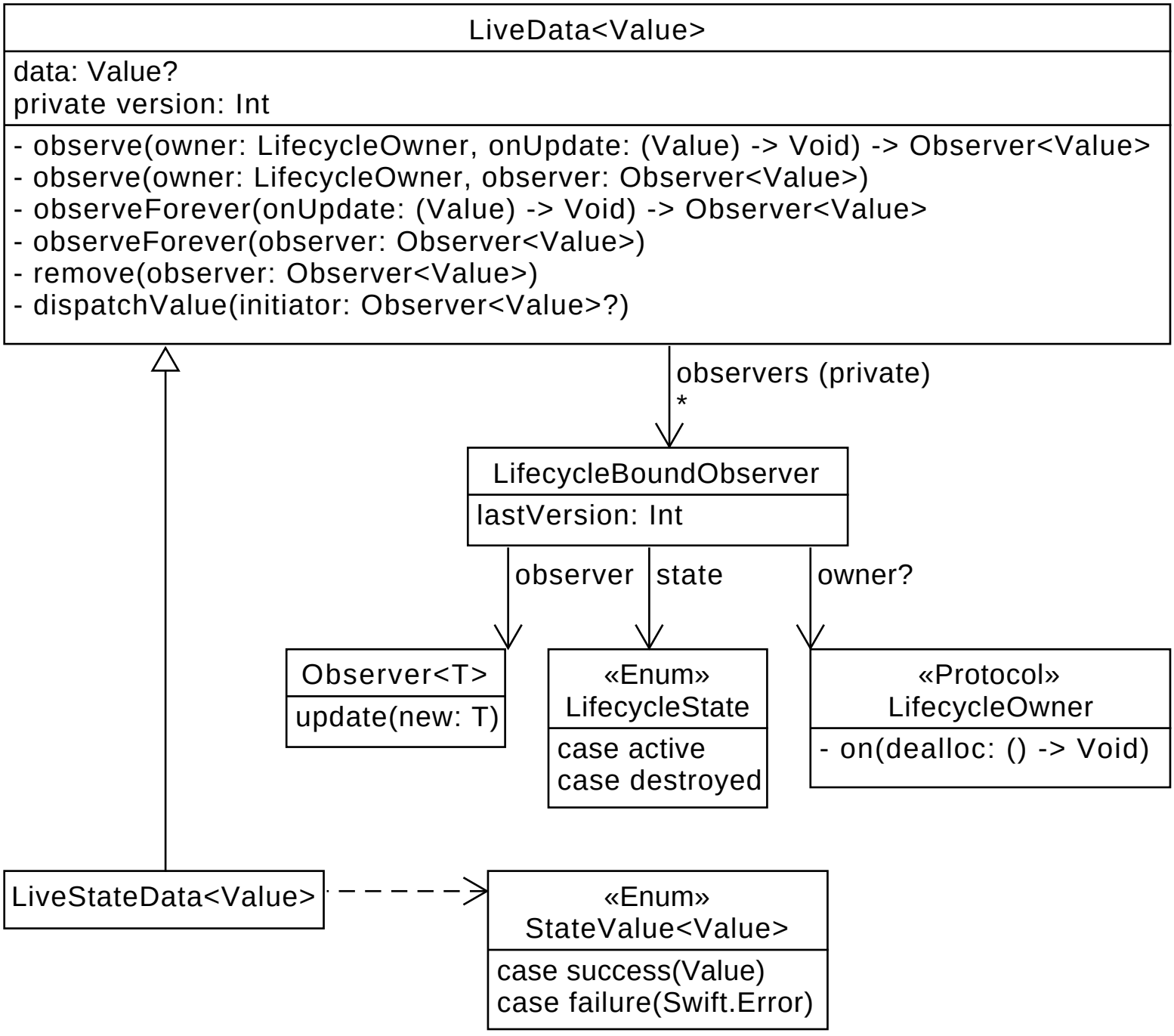
LiveData<Value>
data: Value?
private version: Int
- observe(owner: LifecycleOwner, onUpdate: (Value) -> Void) -> Observer<Value>
- observe(owner: LifecycleOwner, observer: Observer<Value>)
- observeForever(onUpdate: (Value) -> Void) -> Observer<Value>
- observeForever(observer: Observer<Value>)
- remove(observer: Observer<Value>)
- dispatchValue(initiator: Observer<Value>?)
observers (private)
*
LifecycleBoundObserver
lastVersion: Int
observer
state
owner?
Observer<T>
update(new: T)
«Enum»
LifecycleState
case active
case destroyed
«Protocol»
LifecycleOwner
- on(dealloc: () -> Void)
LiveStateData<Value>
«Enum»
StateValue<Value>
case success(Value)
case failure(Swift.Error)