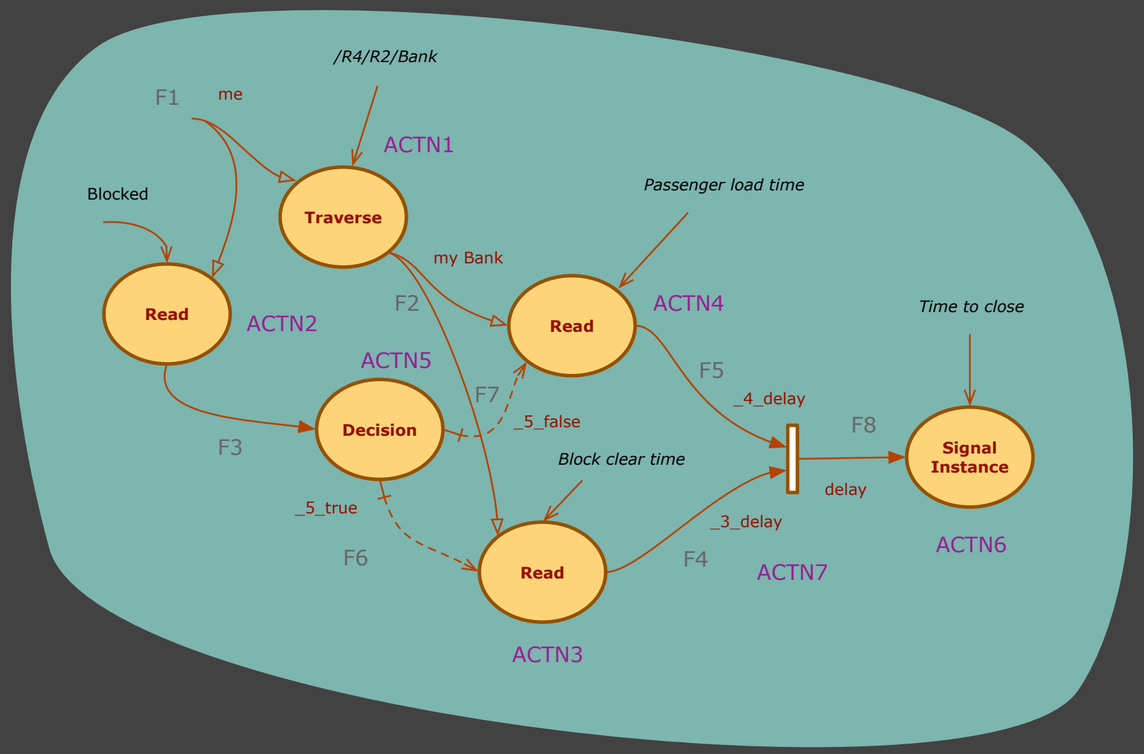
Door Data Flow Diagrams

Leon Starr
2025-10-14/ v0.9

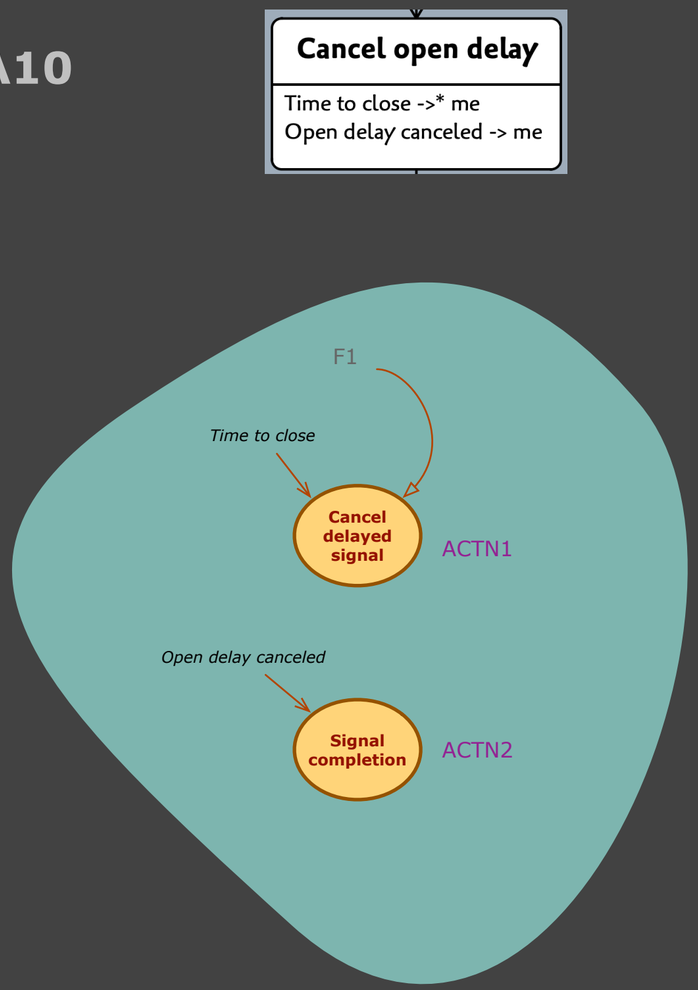
A9

OPEN

my Bank .= /R4/R2/Bank
Blocked? delay = my Bank.Block clear time :
  delay = my Bank.Passenger load time
Time to close -> me @delay

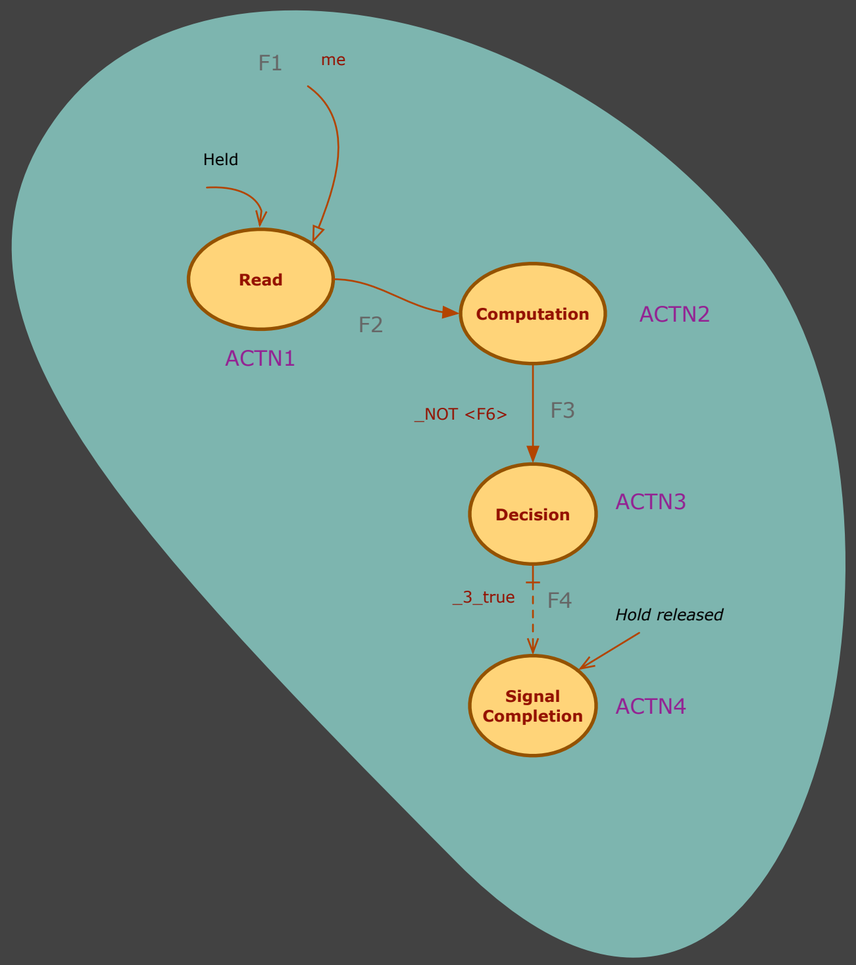


A10



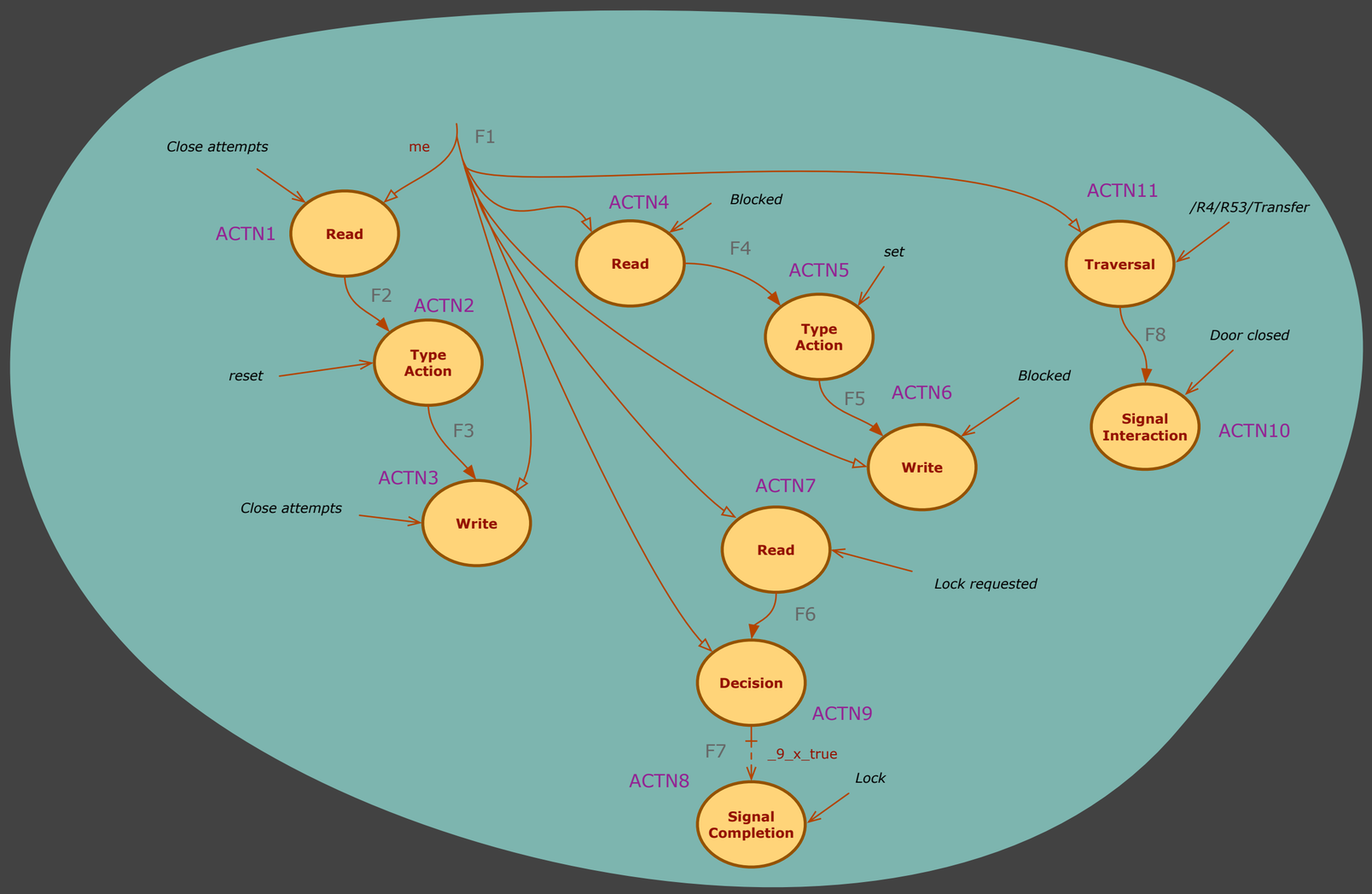
HOLDING OPEN

!Held ? Hold released -> me



CLOSED

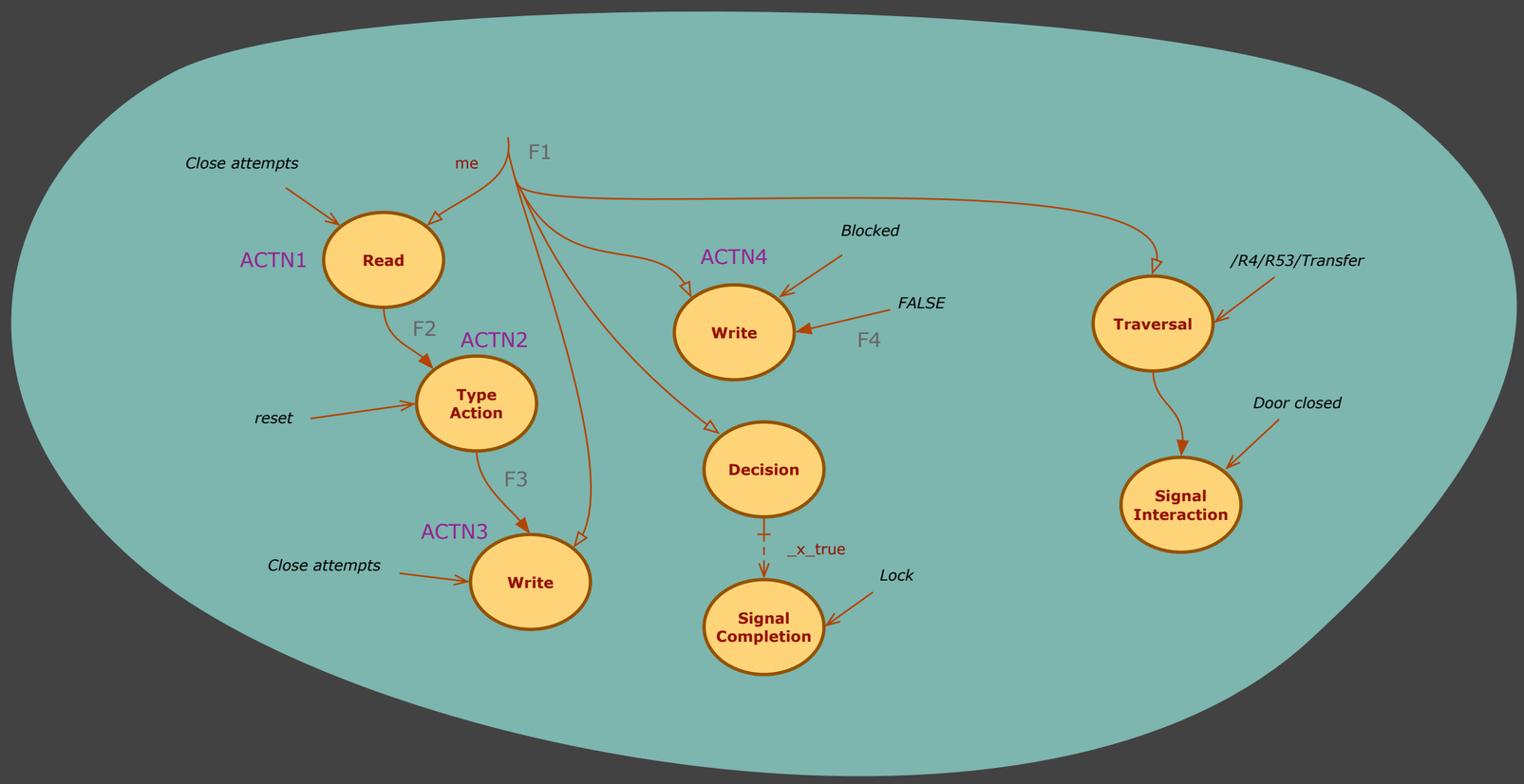
Close attempts.reset
Blocked.unset
Lock requested? Lock -> me
// May be None
Door closed -> /R4/R53/Transfer



CLOSED

Close attempts.reset
Blocked = FALSE
Lock requested? Lock -> me
// May be None
Door closed -> /R4/R53/Transfer

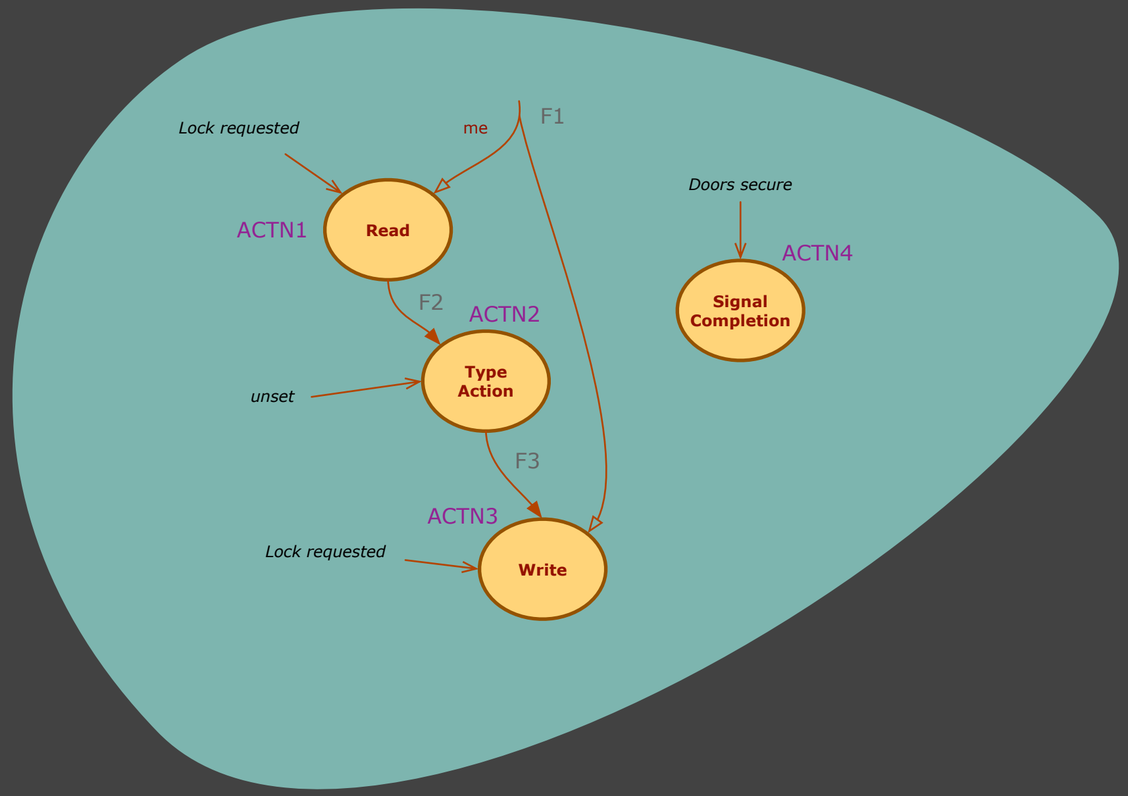
Here we set Blocked to FALSE directly instead of using the boolean set type operation, which is preferred. So we save this alternate, less preferable data flow for reference. Either approach is fine, but I like to avoid explicit constants even though TRUE/FALSE are fairly safe as constants go.



A14

LOCKED

// Tell the cabin that we're ready to go
Lock requested.unset
Doors secure -> /R4/Cabin



A15

Count block

```
Close attempts.increment
(Close attempts > /R4/R2/Bank.Max close attempts) ? Cannot close door -> me : {
   Blocked.set
   Keep trying -> me
}
```