

R53 Data Flow Diagrams

Leon Starr

2025-9-14/ v1.0.0



Copyright © 2025, Leon Starr at

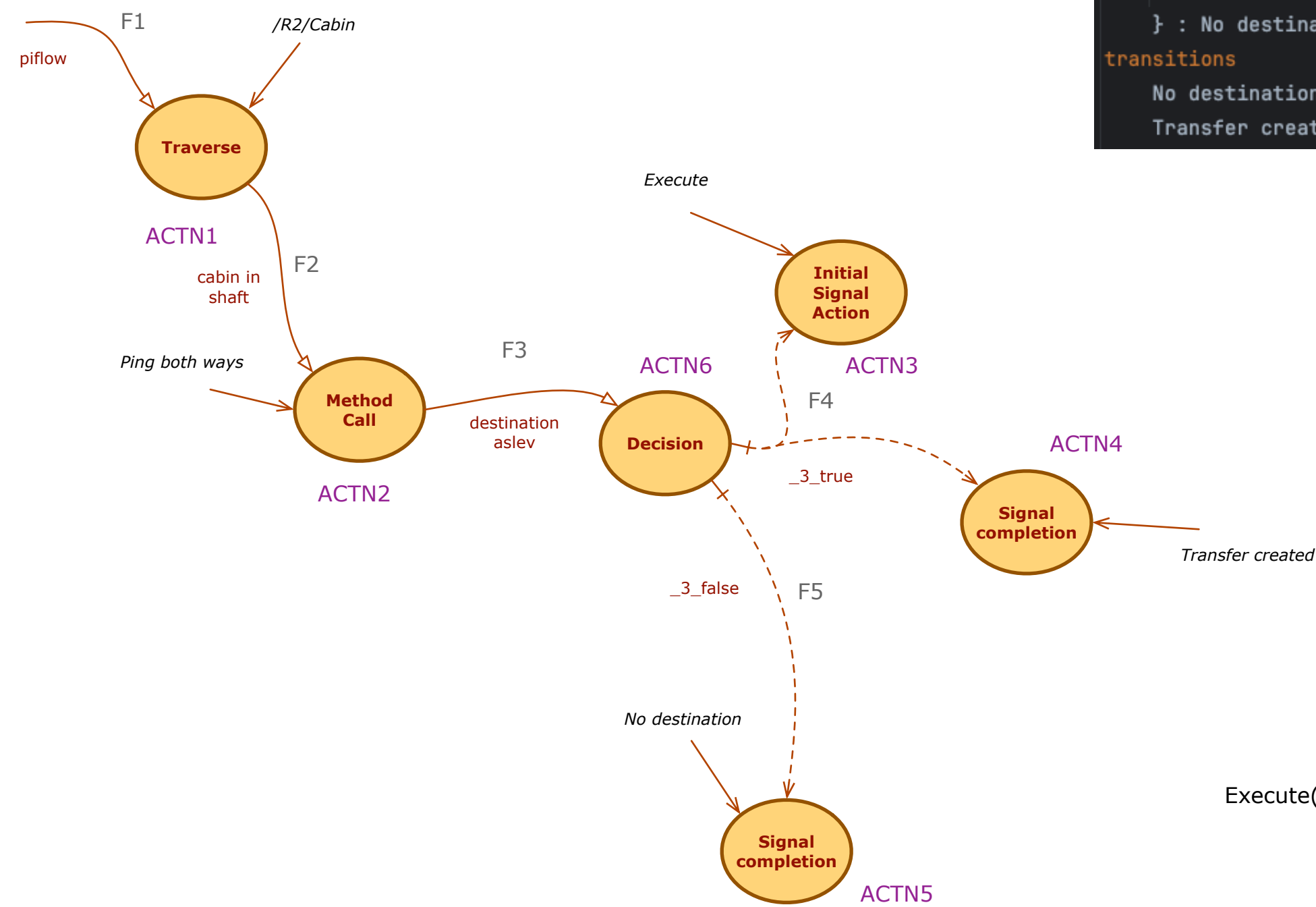
MODEL INTEGRATION, LLC

State name: NO TRANSFER

No actions

```
state NO TRANSFER
activity
    // Waiting for service request
transitions
    Service requested > Search for new destination
```

A9

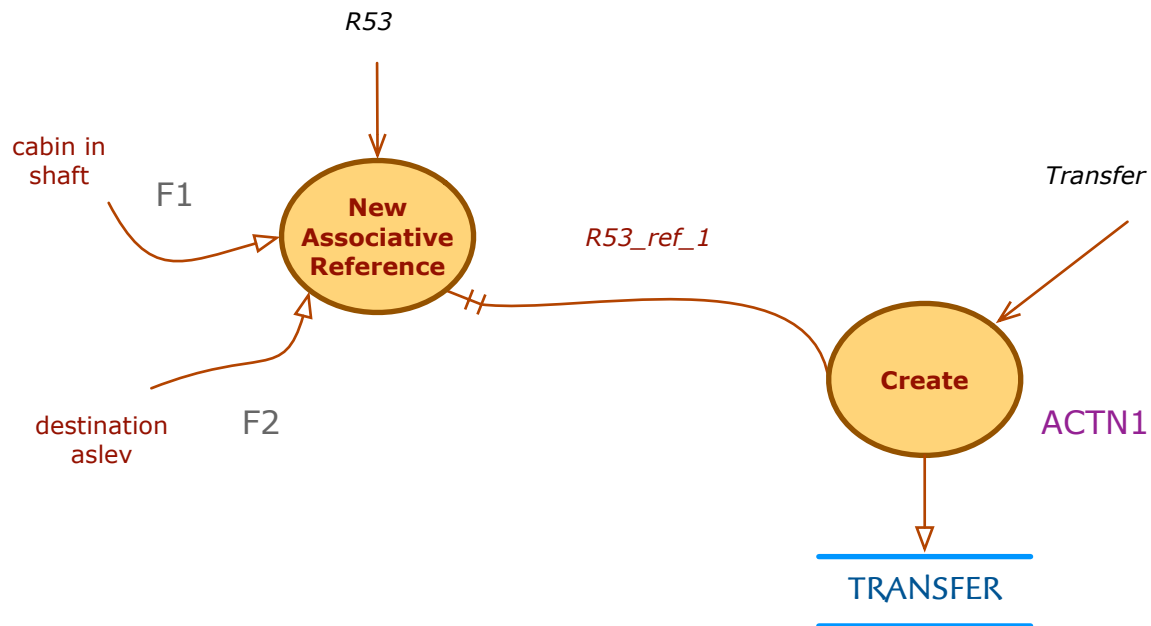


```
state Search for new destination
activity
    // The Cabin in this Shaft is stationary and has no Transfer
    cabin in shaft .= /R2/Cabin
    destination aslev .= cabin in shaft.Ping both ways()
    destination aslev? {
        Execute -> *Transfer &R53 cabin in shaft, destination aslev
        Transfer created -> me
    } : No destination -> me
transitions
    No destination > NO TRANSFER
    Transfer created > TRANSFER IN PROGRESS
```

Execute(cabin in shaft: Cabin, destination aslev: Accessible Shaft Level)

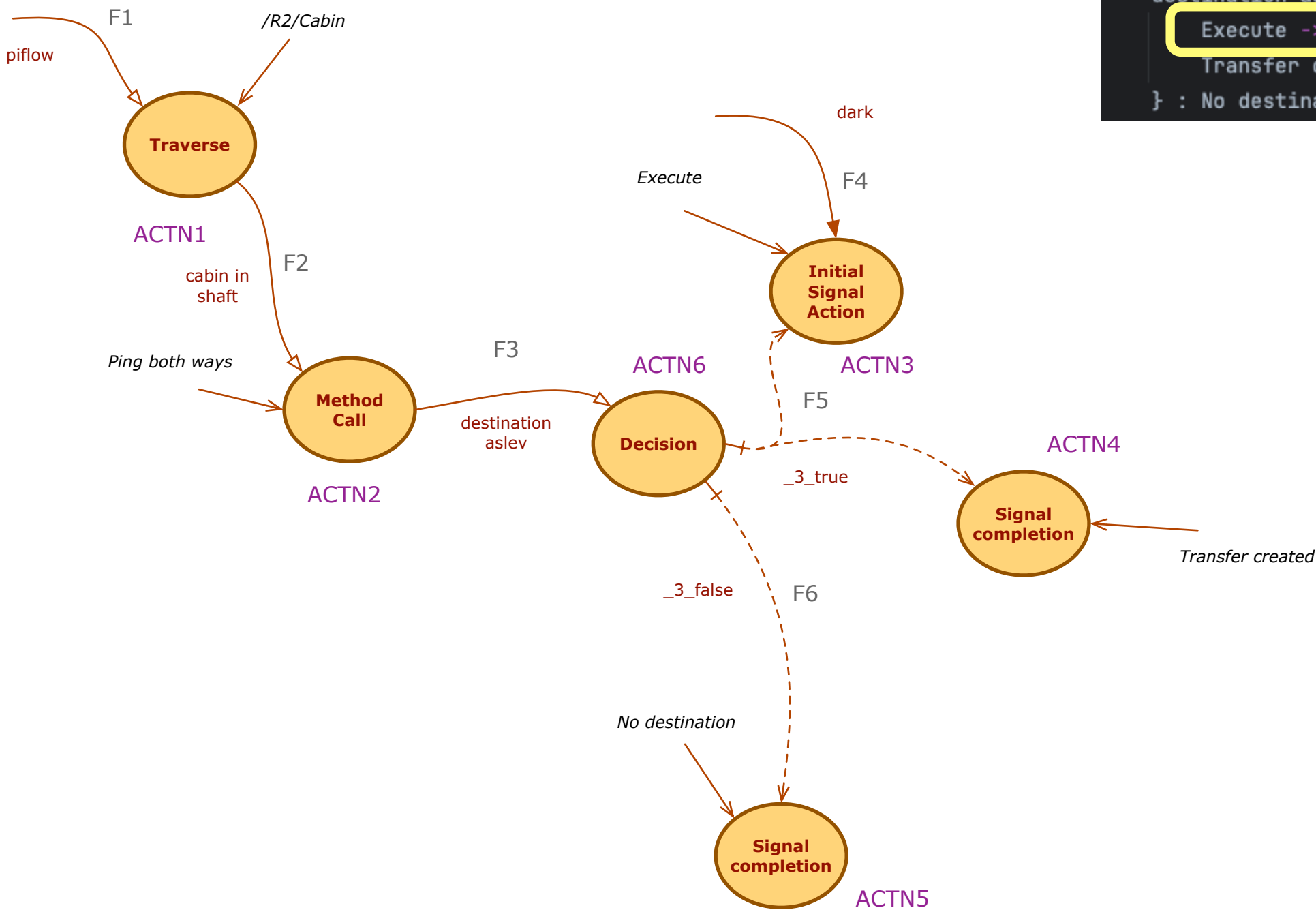
Delegated Creation Activity on Transfer

A38



A9

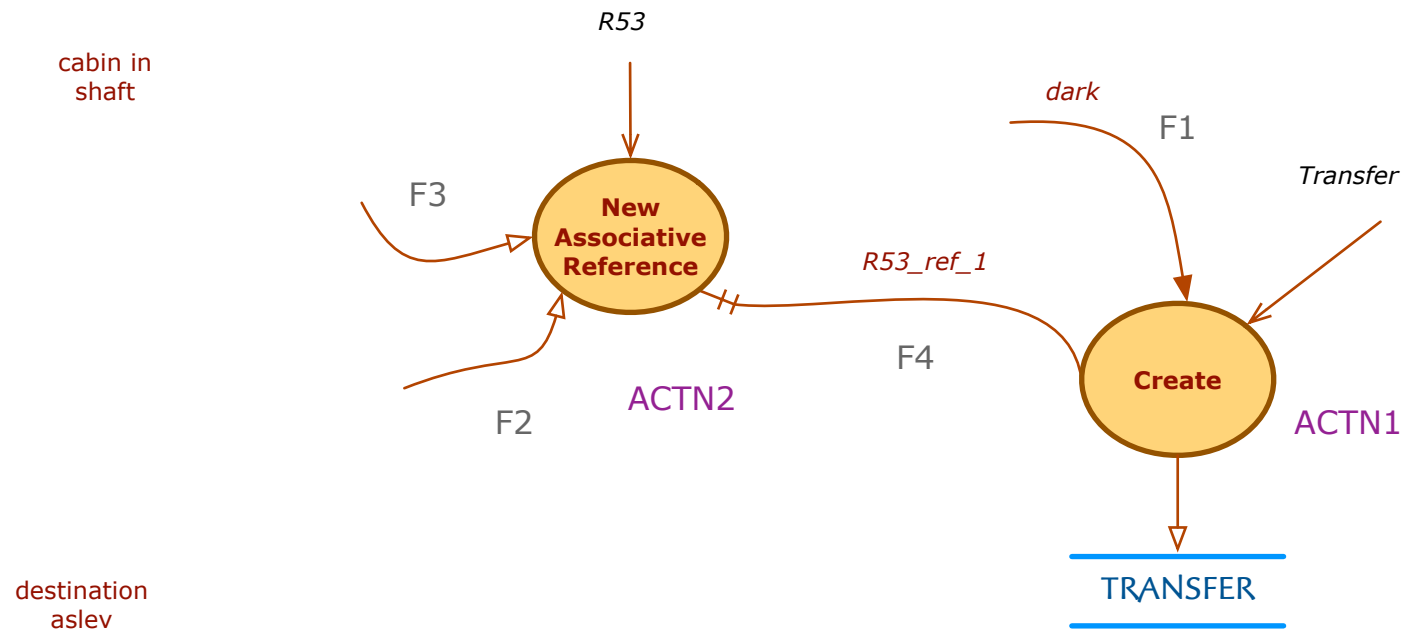
Added Tint/Color attributes to test setting of explicit and default scalar attribute values



```
state Search for new destination
activity
    // The Cabin in this Shaft is stationary and has no Transfer
    cabin in shaft . = /R2/Cabin
    destination aslev . = cabin in shaft.Ping both ways()
    destination aslev? {
        Execute -> *Transfer(Tint: _dark) &R53 cabin in shaft, destination aslev
        Transfer created -> me
    } : No destination -> me
```

A37

Added Tint/Color attributes to test setting of explicit and default scalar attribute values



State name: Search for alternate destination

A11

1

2

3

```
state Search for alternate destination
activity
  cabin in shaft .:= /R2/Cabin
  fwd = cabin in shaft.Travel direction
  xfer .:= cabin in shaft/R53/Transfer
  call in progress .:= xfer/R53/Accessible Shaft Level/R49/Floor Service( Direction: fwd )

  5 fwd dest .:= cabin in shaft.Ping( dir: fwd )
  6 fwd dest? dest aslev .:= fwd dest :
    !call in progress? {
      // Search opposite the travel direction only if not servicing
      // a floor request in the current direction
      rev = fwd.opposite
      rev dest .:= cabin in shaft.Ping( dir: rev )
      rev dest? cabin in shaft.Travel direction = rev
      // may be empty
      dest aslev .:= rev dest
    }

  7 dest aslev? {
    dest floor = dest aslev.Floor
    dest floor != xfer.Destination floor?
      Dest change request( new dest: dest floor ) -> xfer
  }

  8 Dest change evaluated -> me
```

