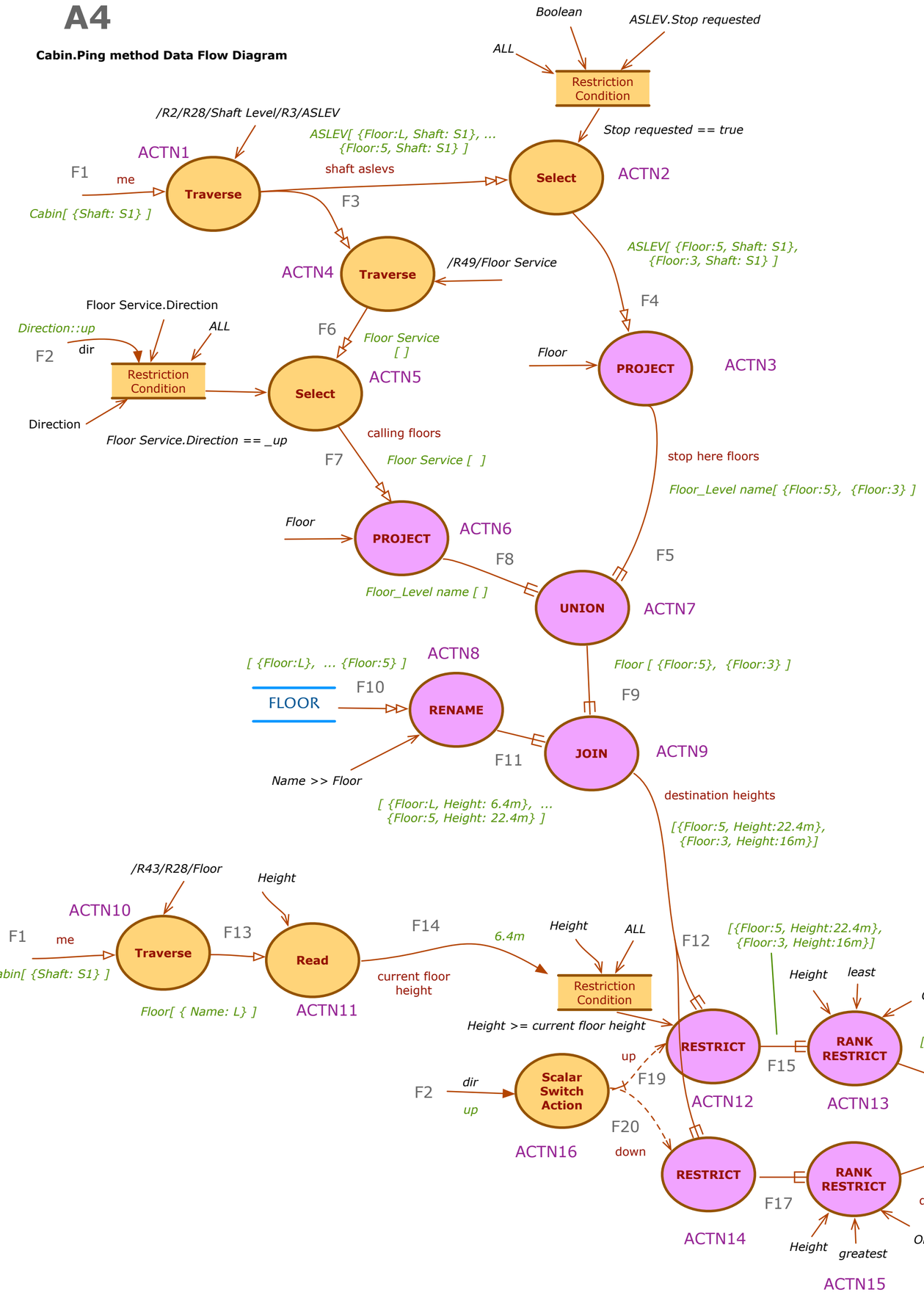
Cabin - Ping DFD

Leon Starr
2025-6-3 / v0.5.1

A4

Cabin.Ping method Data Flow Diagram

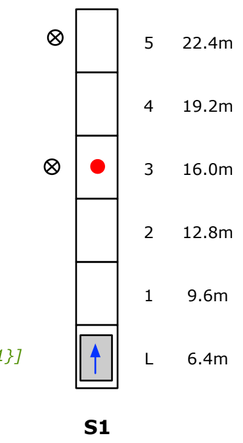



```
--
Cabin.Ping( dir: Direction ) : Accessible Shaft Level
--

//
// Search ahead of the Cabin in the indicated direction
// return the nearest ASL with a stop request and/or a Floor Service calling
// in that direction. If none is found, return the empty set.

// Everywhere I could stop
shaft aslevs ..= /R2/R28/Shaft Level/R3/Accessible Shaft Level
// Pending stop and call requests
stop here floors #= shaft aslevs( Stop requested ).Floor
calling floors #= shaft aslevs/R49/Floor Service(Direction : ^dir ).Floor

// Take the union of all requests and join with Floor to get a table of heights for each
destination heights #= ((stop here floors + calling floors) ## Floor[Name >> Floor])

// Find all Accessible Shaft Levels ahead of the cabin
current floor height = /R43/R28/Floor.Height
^dir? {
    _up:
        // Get the lowest floor at or above the cabin
        nearest dest ||= destination heights(Height >= current floor height)(1, ^-Height)
    _down:
        // Get the highest floor below the cabin
        nearest dest ||= destination heights(Height < current floor height)(1, ^+Height)
}
// Since you can't have two floors at the same height for the same shaft, we have
// at most one row in our table, nonetheless we set cardinality to 1
// to clarify that we have a single tuple result

// Convert to either zero or one instance of ASLEV class and provide as output
=>> Accessible Shaft Level( Floor: nearest dest.Floor; Shaft )
```

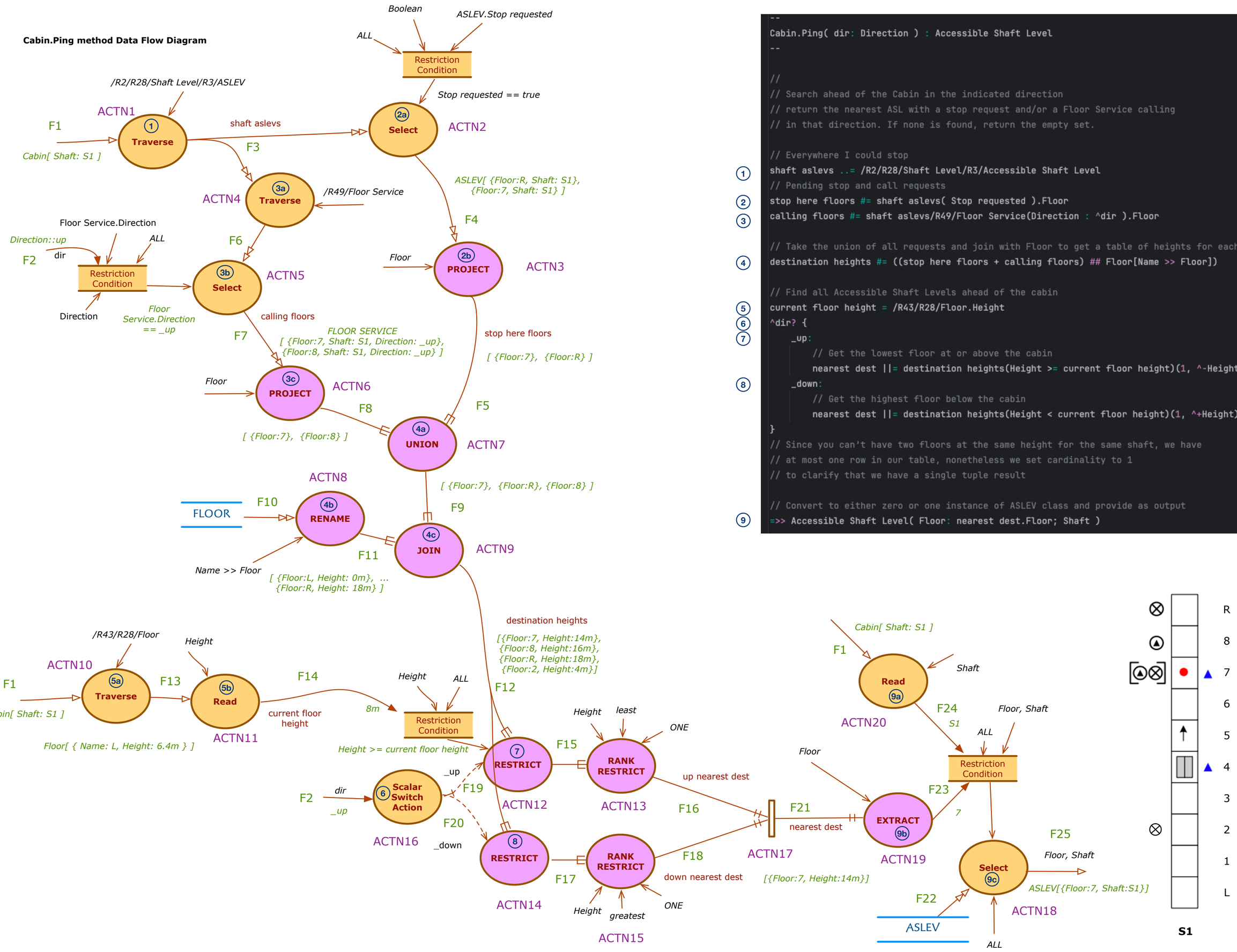
1 2 3 4 5 6 7 8 9

Cabin at Floor L with two stop requests pending at Floors 3 and 5.

No floor up/down calls pending



Cabin.Ping method Data Flow Diagram




```
--
Cabin.Ping( dir: Direction ) : Accessible Shaft Level
--

//
// Search ahead of the Cabin in the indicated direction
// return the nearest ASL with a stop request and/or a Floor Service calling
// in that direction. If none is found, return the empty set.

// Everywhere I could stop
(1) shaft aslevs ..= /R2/R28/Shaft Level/R3/Accessible Shaft Level
// Pending stop and call requests
(2) stop here floors #= shaft aslevs( Stop requested ).Floor
(3) calling floors #= shaft aslevs/R49/Floor Service(Direction : ^dir ).Floor

// Take the union of all requests and join with Floor to get a table of heights for each
(4) destination heights #= ((stop here floors + calling floors) ## Floor[Name >> Floor])

// Find all Accessible Shaft Levels ahead of the cabin
(5) current floor height = /R43/R28/Floor.Height
(6) ^dir? {
(7)     _up:
            // Get the lowest floor at or above the cabin
            nearest dest ||= destination heights(Height >= current floor height)(1, ^-Height)
(8)     _down:
            // Get the highest floor below the cabin
            nearest dest ||= destination heights(Height < current floor height)(1, ^+Height)
}
// Since you can't have two floors at the same height for the same shaft, we have
// at most one row in our table, nonetheless we set cardinality to 1
// to clarify that we have a single tuple result

// Convert to either zero or one instance of ASLEV class and provide as output
(9) =>> Accessible Shaft Level( Floor: nearest dest.Floor; Shaft )
```