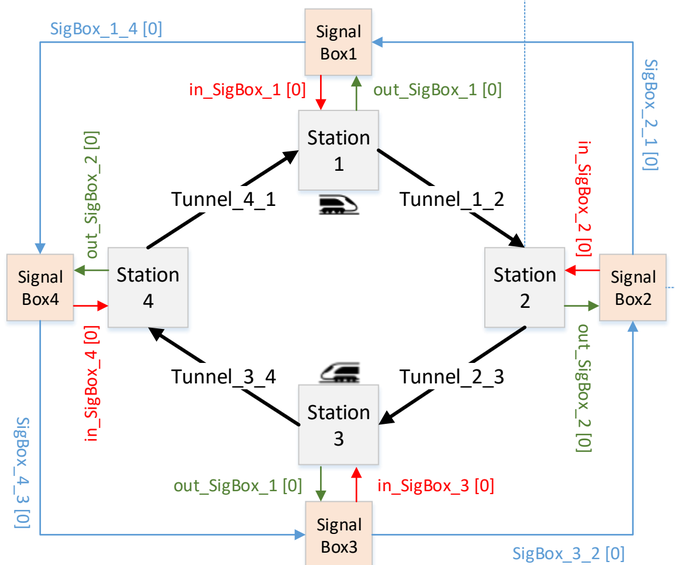
Key:

- Green arrows represent channels that carry messages from station *i* (for *i=1,2,3,4*) to the local (*i*th) signalbox; my naming convention for these channels is `out_SigBox_i` where the word `out` indicates that messages are output in the direction from the station to the signalbox (i.e. they go 'out' of the station)
- Red arrows represent channels that carry messages in the opposite direction, i.e. from signalbox *i* (for *i=1,2,3,4*) to the local (*i*th) station; my naming convention for these channels is `in_SigBox_i` where the word `in` indicates that messages are output in the direction from the signalbox to the station (i.e. they go 'in' the station)
- Blue arrows represent channels that carry messages from one signalbox to the other; note that a signalbox can only output messages to the preceding signalbox, and can only input messages from the signalbox in advance; my naming convention for these channels in `SigBox_i_j` where *i* and *j* are adjacent signalboxes such that *j* precedes *i* and *i* emits a message to *j*
- Note that you may (but you are not obliged to) follow my naming convention for the channels
- Also note that all channel described above are sync channels (i.e. they have no capacity to buffer messages); you have to decide for yourselves the types of messages carried by each channel

Note:

It is assumed that all local signals are initially showing red

- Upon arrival of a train (i.e. when `in_track?train` occurs), it informs the local signalbox by sending an appropriate message on `out_SigBox_2`.
- In case a train is already in the station (i.e. when `train_cnt > 0` is true), it requests from the local signalbox to instruct the local channel to show green; it does so by sending an appropriate message on `out_SigBox_2`; it then receives a response from the local signalbox that sets the local signal either to green or to red.
- Naturally, the train can only depart (indicated by the communication `out_track!train`) if the local signal has been set to green.
- Once the train departs, the local signalbox is informed by an appropriate message on `out_SigBox_2`.



- Initially ready to receive a message either on `out_SigBox_2` or on `SigBox_3_2`
- A message is received on `out_SigBox_2` on three distinct occasions:
  - upon arrival of a train at the local station (say occasion 'A')
  - upon request from the local station to set the local signal to green (say occasion 'R')
  - upon departure of a train from the local station (say occasion 'D')
- A message is received on `SigBox_3_2` when a train arrives at Station 3, i.e. when a train that used to occupy `Tunnel_2_3` arrives at Station 3 and therefore clears `Tunnel_2_3`
- In occasion 'A', the signalBox sends a message to the preceding signalbox to inform it that `Tunnel_1_2` is now empty
- In occasion 'R', the signalbox replies to the request either by sending a 'proceed' message on channel `in_SigBox_2` that instructs the local signal to show green, or by sending a 'do not proceed' message on channel `in_SigBox_2` that instructs the local signal to show red; naturally, a 'proceed' message can only be issued if the tunnel in advance (`Tunnel_2_3` in this case) is not occupied, otherwise a 'do not proceed' message must be emitted.
- In occasion 'D', the signalbox flags the tunnel in advance (i.e. (`Tunnel_2_3`) as occupied, and emits a 'do not proceed' message on channel `in_SigBox_2` that instructs the local signal to show red.
- Upon receipt of a message on `SigBox_3_2` the signalbox flags the tunnel in advance (i.e. (`Tunnel_2_3`) as not occupied.
- The behaviour described in the bullets above repeats, of course, indefinitely.