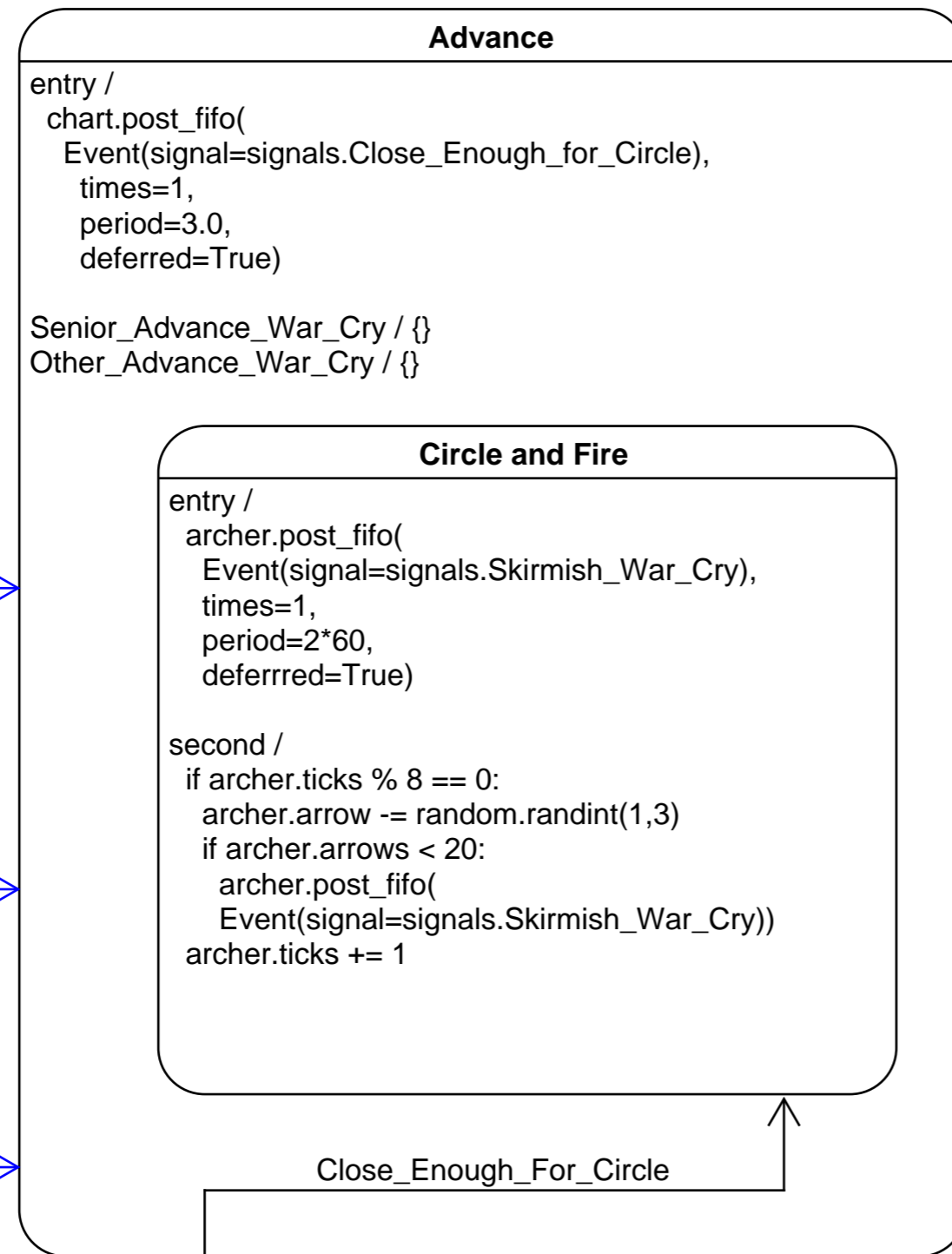


## Deceit\_in\_Detail\_Tactic (marshaled)

```

entry /
# Identify next marshal point
# Load horse with arrows
archer.arrows = HorseArcher.MAXIMUM_ARROW_CAPACITY
archer.ticks = 0
archer.post_fifo(Event(signal=signals.Second, times=0, period=archer.to_time(1.0), deferred=True))
second /
archer.ticks += 1
exit / archer.cancel_event(Event(signal=signals.Seconed))
Senior_Advance_War_Cry as e \ archer.post_fifo(e)
Senior_Skirmish_War_Cry as e \ archer.post_fifo(e)
Senior_Retreat_War_Cry as e \ archer.post_fifo(e)

```



```

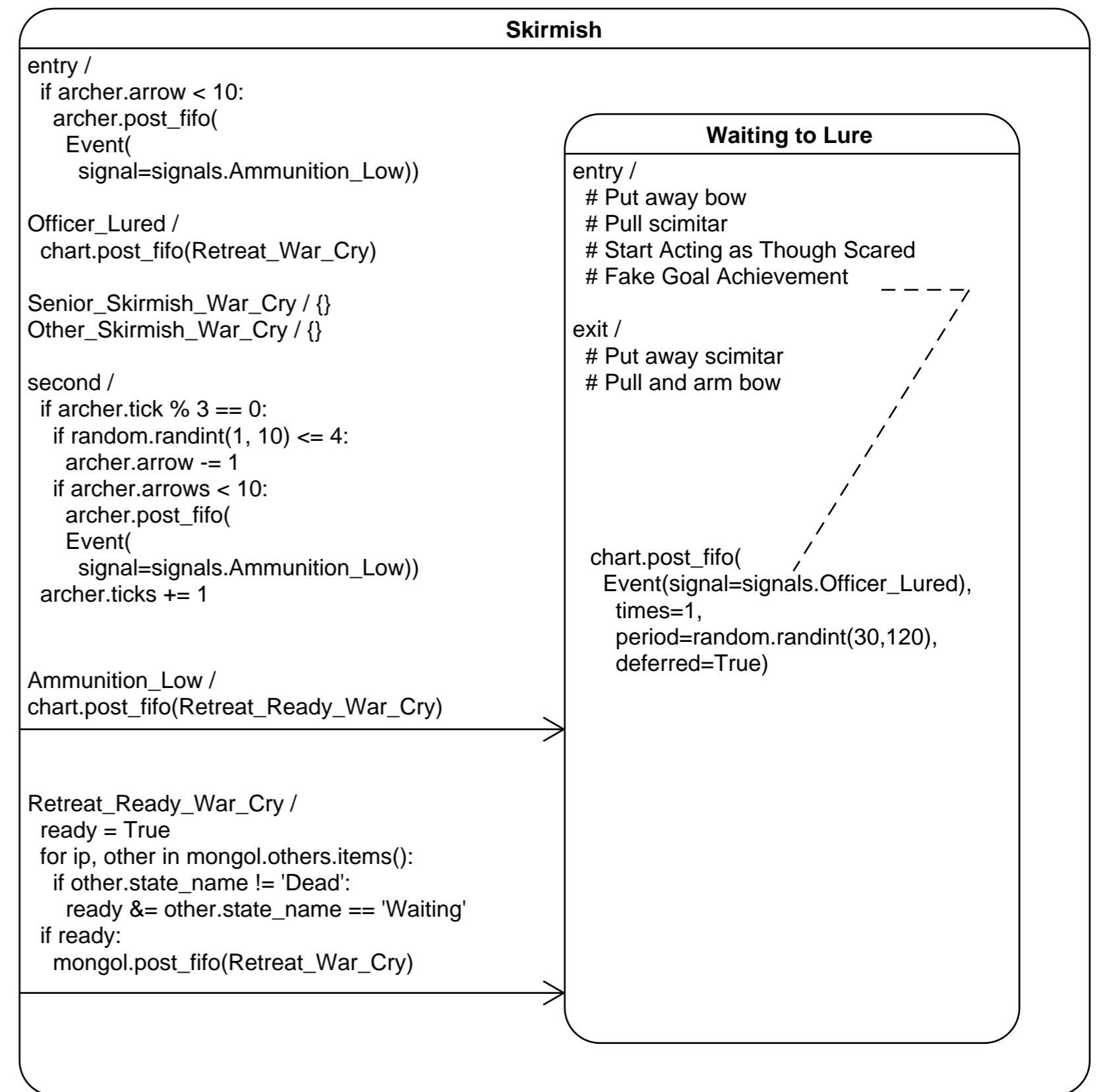
Advance_War_Cry as e /
archer.yell(e)
for ip, other in archer.others.items():
other.dispatch(e)

```

```

Other_Advance_War_Cry as e, ip: /
archer.post_fifo(
Event(
signal=/
signals.Advance_War_Cry))
archer.other[ip].dispatch(e)

```



```

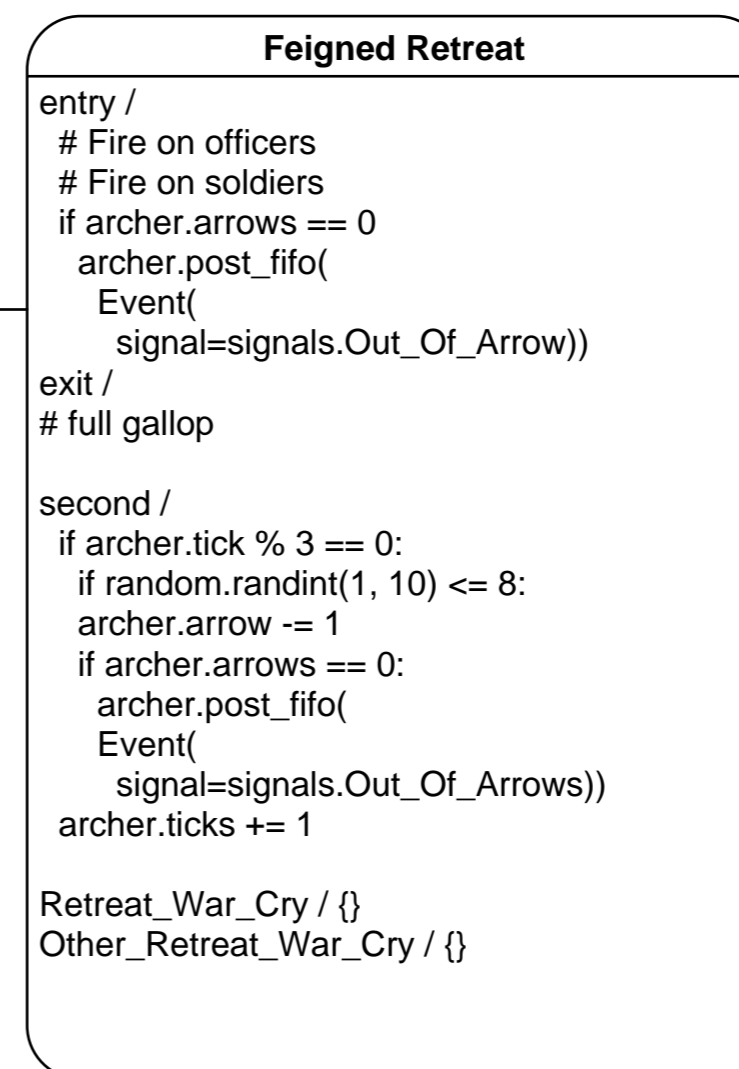
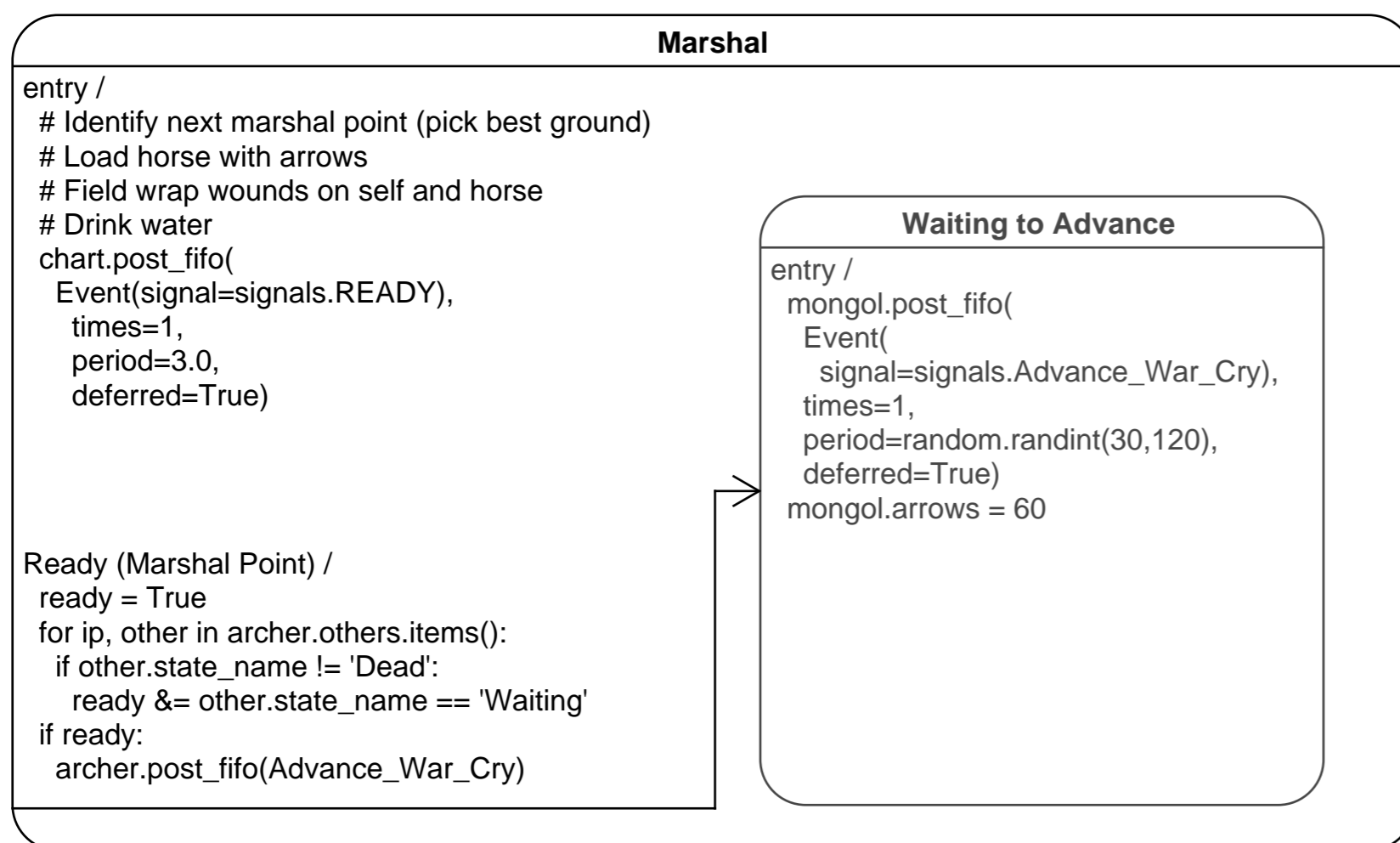
Other_Skirmish_War_Cry as e, ip \
archer.post_fifo(
Event(
signal=/
signals.Skirmish_War_Cry))
mongol.other[ip].dispatch(e)

```

```

Skirmish_War_Cry /
archer.yell(Skirmish_War_Cry)

```



```

Retreat_War_Cry as e /
archer.yell(e)
for ip, other in mongol.others.items():
other.dispatch(e)

```

```

Other_Retreat_War_Cry as e, ip:
mongol.other[ip].dispatch(e)

```

Out\_Of\_Arrows