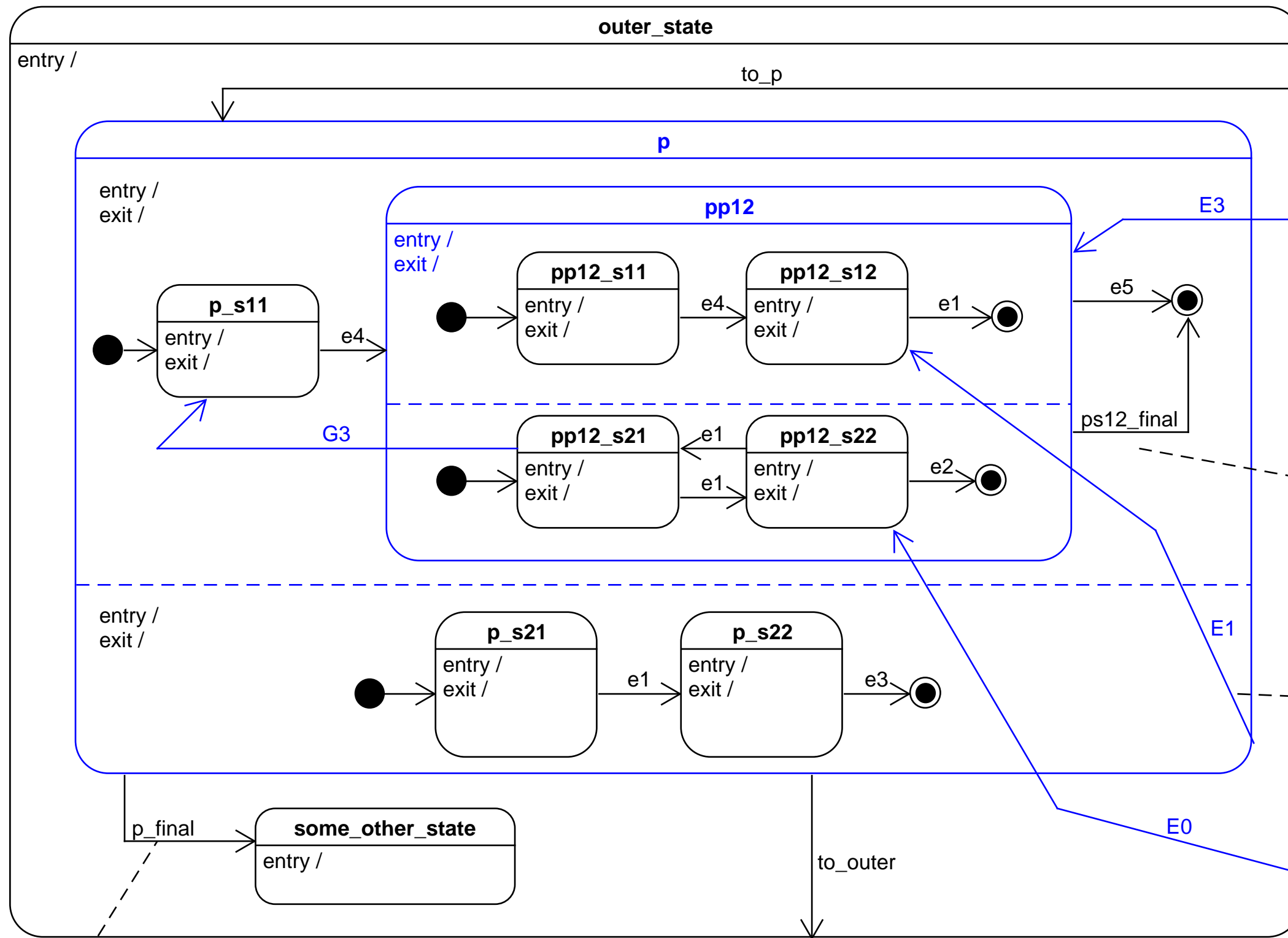


# Orthogonal Regions Diagram



sent when s1Final and s2Final states have been entered

sent when ps12\_s1Final and ps12\_s2Final states have been entered

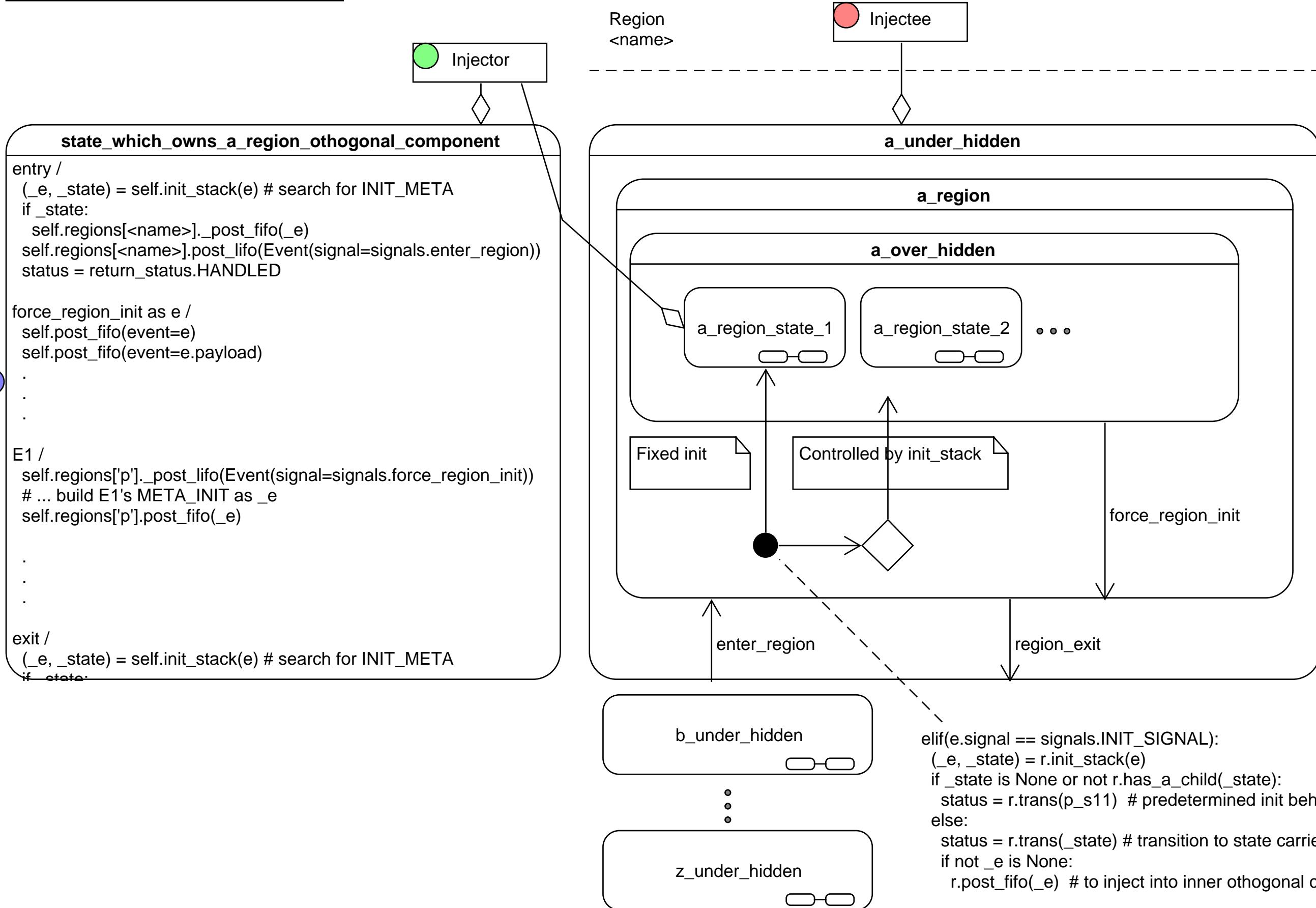
```
# E3
eeee = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=None, state=pp12, source_event=e)
)
eee = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=eeee, state="pp12", source_event=e)
)
ee = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=eee, state=pp12, source_event=e)
)
_e = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=ee, state="p", source_event=e)
)
```

```
# E1
eee = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=None, state=pp12_s12, source_event=e)
)
ee = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=eee, state=pp12, source_event=e)
)
_e = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=ee, state=pp12, source_event=e)
)
```

```
# E0
eeee = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(
        event=None, state=pp12_s22, source_event=e)
)
eee = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=eeee, state=pp12, source_event=e)
)
ee = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=eee, state=pp12, source_event=e)
)
_e = Event(
    signal=signals.INIT_META,
    payload=INIT_META_PAYLOAD(event=ee, state=pp12, source_event=e)
)
```

state just has to return True

state must be callable



```
elif(e.signal == signals.INIT_SIGNAL):
    (_e, _state) = r.init_stack(e)
    if _state is None or not r.has_a_child(_state):
        status = r.trans(p_s11) # predetermined init behavior based on chart
    else:
        status = r.trans(_state) # transition to state carried in META_INIT_PAYLOAD
    if not _e is None:
        r.post_fifo(_e) # to inject into inner othogonal compoent if there
```