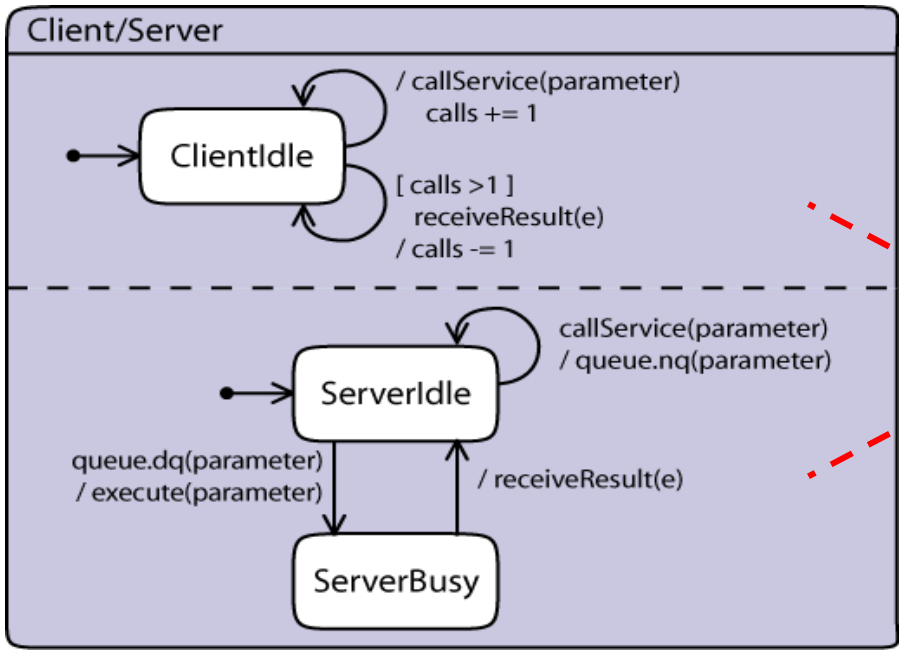
Orthogonal regions

- **Simple State**: containing no Region
- **Composite State**: containing at least one Region
  - simple composite State: exactly one
  - orthogonal composite State: at least two



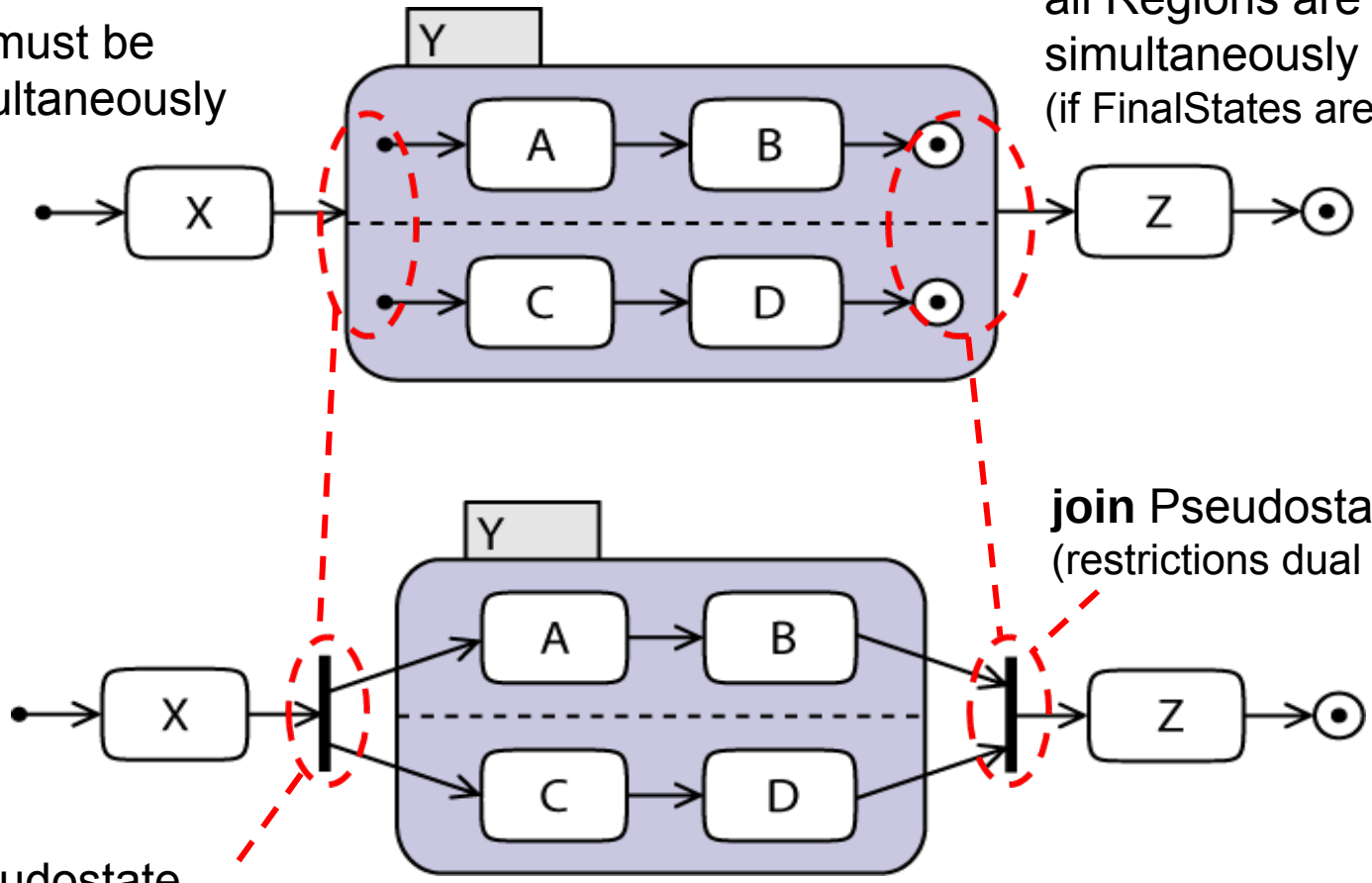
orthogonal **Region**s, both active if Client/Server active

orthogonal states are “concurrent” as a single event may trigger a transition in each orthogonal region “simultaneously”

Forks and joins

all Regions must be entered simultaneously

all Regions are left simultaneously
(if FinalStates are reached)

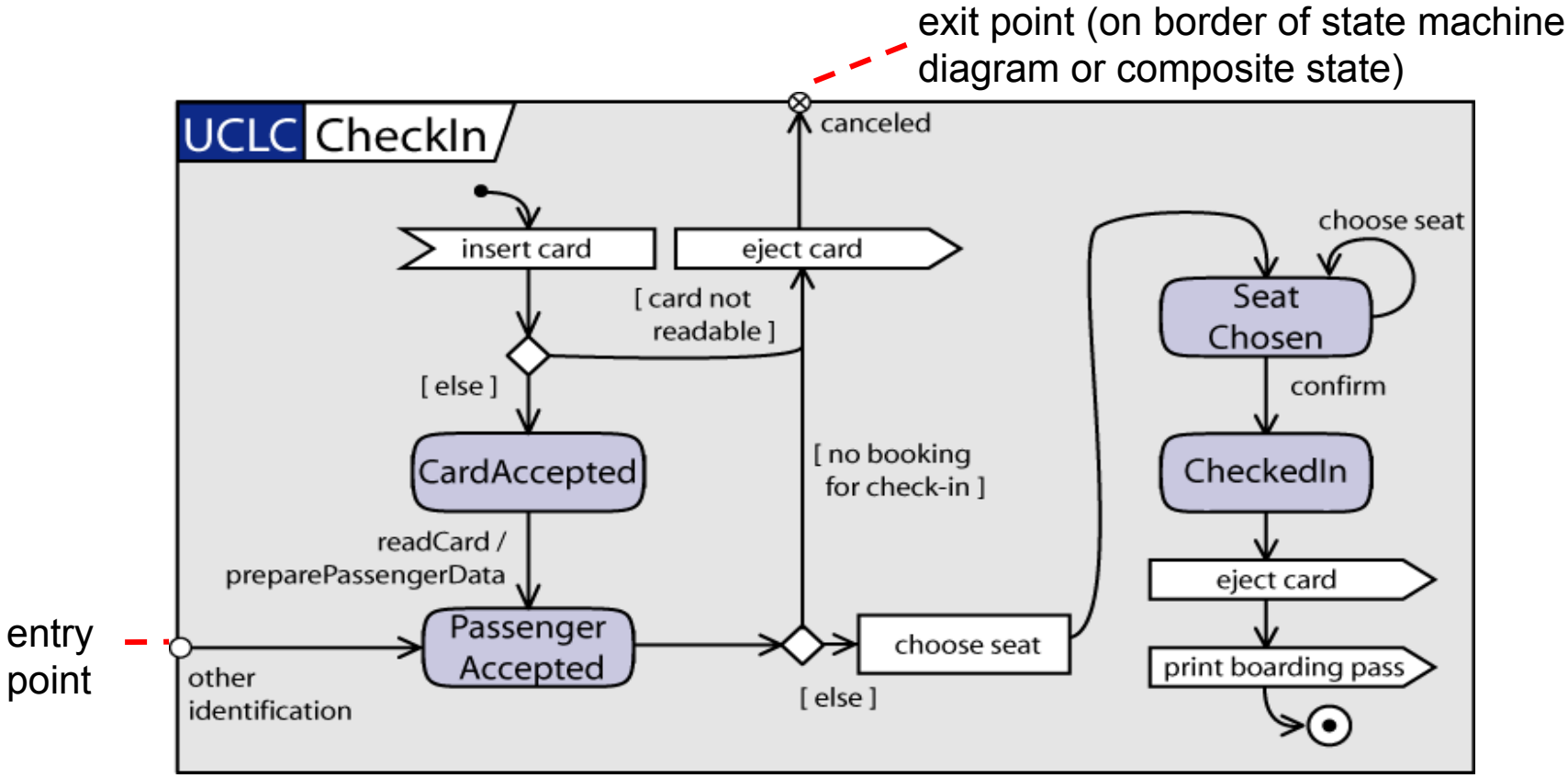


join Pseudostate
(restrictions dual to forks)

fork Pseudostate
(one incoming, at least two outgoing Transitions;
outgoing Transitions must target States in different Regions of an orthogonal State)

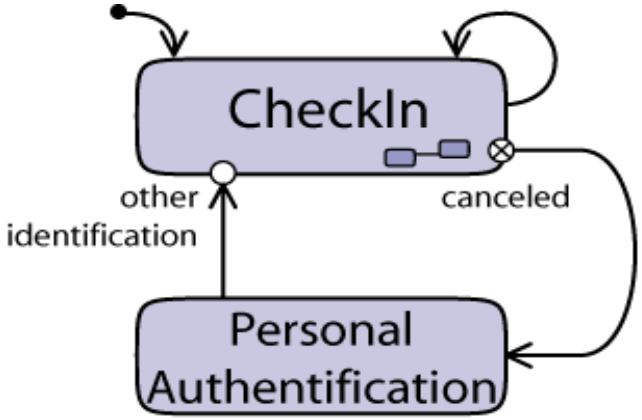
Entry and exit points (1)

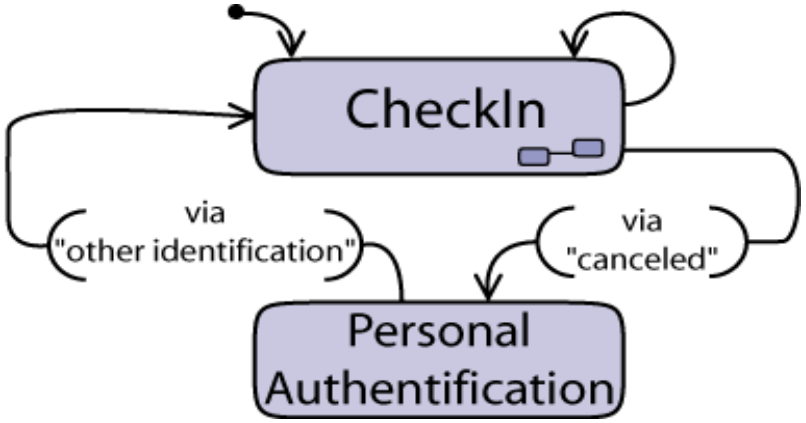
- Entry and exit points (Pseudostates)
  - provide better encapsulation of composite states
  - help avoid “unstructured” transitions



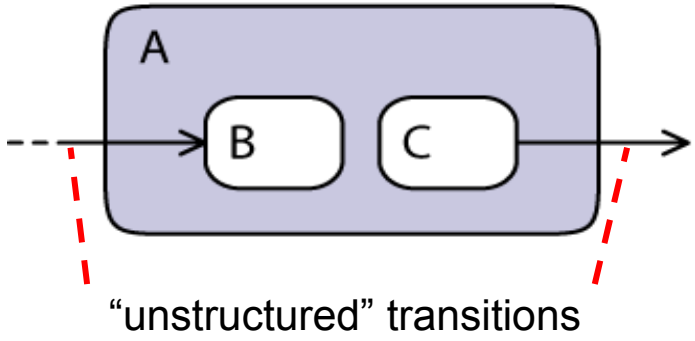
Entry and exit points (2)

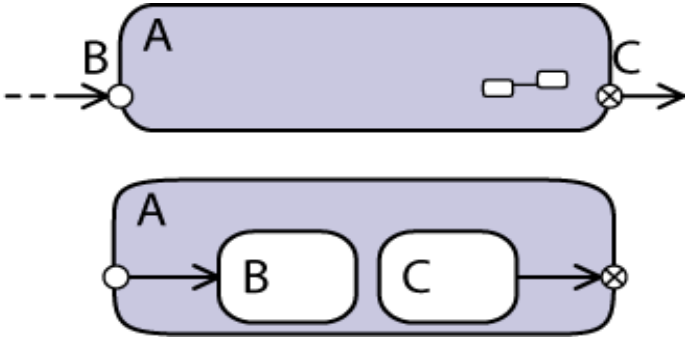
Notational alternatives





Semantically equivalent





History states

- History states represent the last active
  - substate (shallow history), or
  - configuration (deep history)

  of a region.