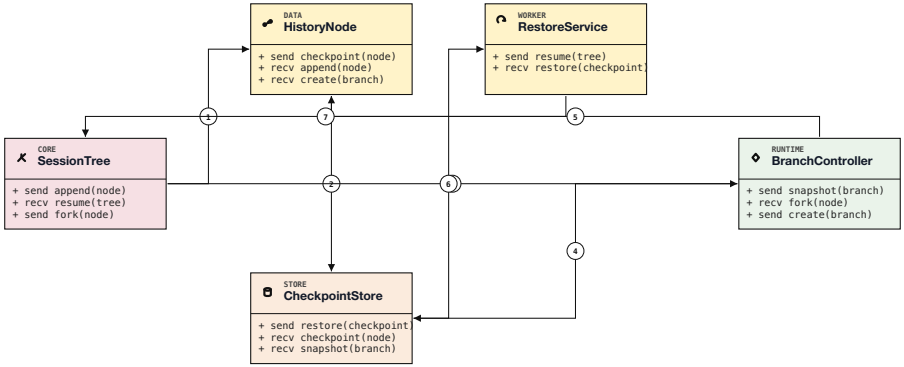
P26 Session-tree version control

Primitive · A² A3 · tree role collaboration

Branch or revisit execution history.

STRUCTURE · PARTICIPANT COLLABORATION



PATTERN INTERFACE

context
Execution history can be checkpointed and branched.

trigger
A fork, checkout, rewind, or restore is requested at an event boundary.

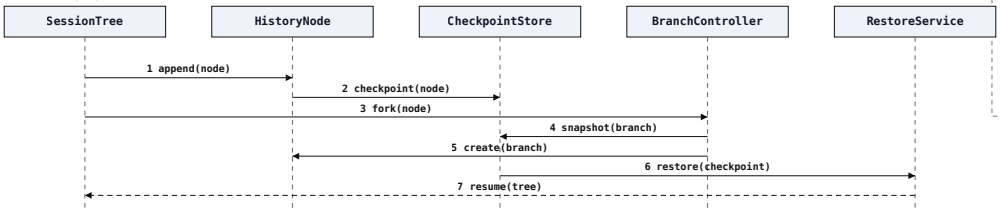
structure
Node identity, ancestry, checkpoint payload, and branch creation operation.

effect
The new branch starts from the requested stable history without mutating its ancestors.

CONTRACT ANNOTATIONS

requires	Event boundary is stable and enough state exists to resume.
ensures	New branch records parent, boundary, and provenance; invalid boundaries are rejected.
invariant	Branch operations never rewrite unrelated history.
violation	External side effects are not rolled back and must be reported as irreversible.

INTERACTION



MESSAGE TRACE `1 append(node) → 2 checkpoint(node) → 3 fork(node) → 4 snapshot(branch) → 5 create(branch) → 6 restore(checkpoint) → 7 resume(tree)`

Evidence: 32/41 repositories, 30/36 families; 32 trajectories (15 multi-checkpoint); exact audit: not sampled.

Variation: Transcript fork, session copy, Git-style checkpoint, event UUID branch, and rewind.

Negative control: Workspace checkpoint without conversation branching is only partial.