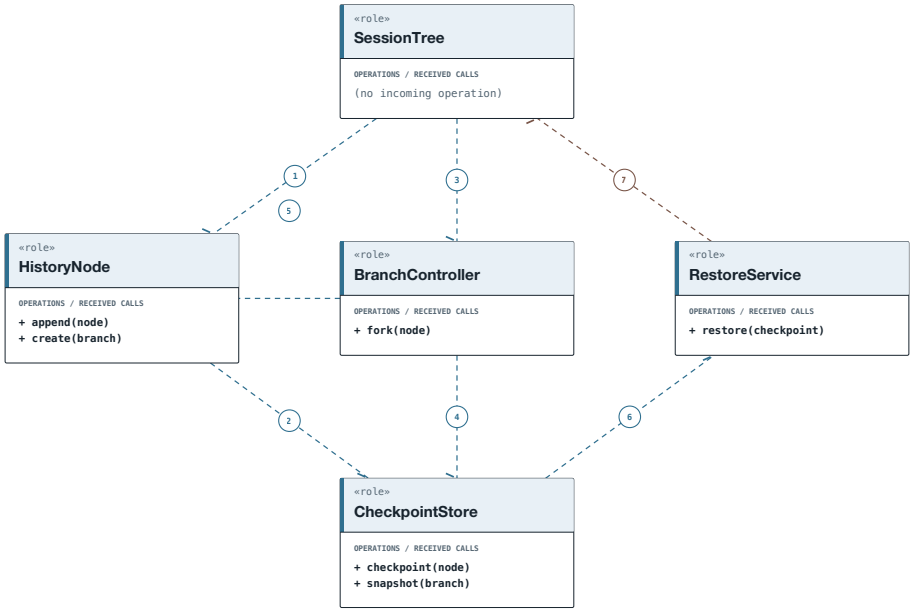
P26 Session-tree version control

Primitive / UML CLASS VIEW / ANALYTICAL MODEL

Branch or revisit execution history.

UML CLASS VIEW

dashed arrow: call dependency · brown: return/failure · analytical, not an observed trace



POINTCUT / 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 / REQUIRED BEHAVIOR

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.

RELATIONSHIPS / ANALYTICAL CALLS

number · caller → callee · message

1 SessionTree → HistoryNode append(node)
2 HistoryNode → CheckpointStore checkpoint(node)
3 SessionTree → BranchController fork(node)
4 BranchController → CheckpointStore snapshot(branch)
5 BranchController → HistoryNode create(branch)
6 CheckpointStore → RestoreService restore(checkpoint)
7 RestoreService ↩ SessionTree resume(tree)

VARIATION / TEST / NEGATIVE CONTROL

VARIATION	Transcript fork, session copy, Git-style checkpoint, event UUID branch, and rewind.
TEST_OBLIGATION	Test valid fork, invalid boundary, branch isolation, restore, and side-effect caveat.
NEGATIVE_CONTROL	Workspace checkpoint without conversation branching is only partial.