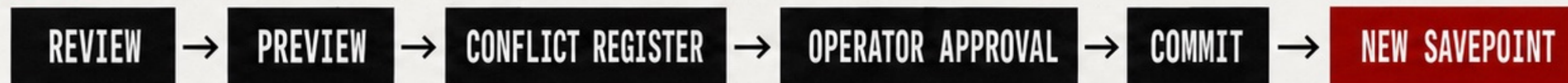


## CORE CONCEPT

### A SAVEPOINT IS A CONTINUITY ARTIFACT.

It preserves project state, agent identity layer boundaries, operator context, decisions, risks, open questions, and next actions without automatically overriding current state.

**LIFECYCLE:** ALL STATE-CHANGING ACTIONS REQUIRE PREVIEW, CONFLICT REGISTER, AND OPERATOR APPROVAL.



## 01 SAVEPOINT PURPOSE

- Preserve continuity across chats, agents, folders, and direct uploads
- Capture current objectives, decisions, constraints, assumptions, open questions, and drift risks
- Maintain machine-readable continuity state
- Savepoint is not a casual summary; it is a continuity artifact



## 02 COMMAND SURFACE

|                                  |                                                            |
|----------------------------------|------------------------------------------------------------|
| <code>/save</code>               | create current-state savepoint                             |
| <code>/restore preview</code>    | inspect restore candidate; no state change                 |
| <code>/restore commit</code>     | apply approved restore items; generate new savepoint       |
| <code>/diff preview</code>       | compare same-agent historical savepoint; no state change   |
| <code>/diff commit</code>        | apply approved deltas; generate new savepoint              |
| <code>/absorb preview</code>     | review foreign-agent savepoint; identity firewall active   |
| <code>/absorb commit</code>      | apply approved cross-agent context; generate new savepoint |
| <code>/validate savepoint</code> | structure / metadata / privacy / conflict / JSON check     |
| <code>/status savepoint</code>   | lineage / conflicts / questions / drift risks              |
| <code>/archive savepoint</code>  | mark lifecycle state                                       |
| <code>/deployment status</code>  | identify source / index / registry / candidate             |

## 03 NON-DESTRUCTIVE VS STATE-CHANGING

| NON-DESTRUCTIVE    | STATE-CHANGING                                         |
|--------------------|--------------------------------------------------------|
| <code>/save</code> | <code>/restore commit</code>                           |
| preview operations | <code>/diff commit</code>                              |
| validation         | <code>/absorb commit</code>                            |
| status review      | archive status changes when they affect active lineage |

State-changing commands require preview, conflict register, explicit operator approval, and a new savepoint.

## 04 AUTHORITY ORDER

- System / developer instructions
- Operator instructions
- Current chat state
- Canonical project files
- Active savepoint
- Historical savepoints
- Foreign-agent savepoints
- Model inference

**CURRENT INSTRUCTIONS OUTFRANK SAVEPOINTS.**

## 05 STORAGE LOGIC

- PROJECT SOURCES**  
= durable library / reference
- DIRECT UPLOAD**  
= selected operational evidence
- AGENT-LEVEL PACKET**  
= stable protocol behavior
- ACTIVE INDEX**  
= current-truth pointer

**GOLDEN RULE:**  
Project folder = library / index. Direct upload = selected evidence. Project instructions = activation switch. Active index = current-truth pointer.

## 06 IDENTITY FIREWALL



Cross-agent project context may transfer. Another agent's identity must not transfer.

- ALLOWED TRANSFER:**
  - shared project context
  - coordination updates
  - operator-relevant preferences
- BLOCKED TRANSFER:**
  - other agent name
  - role
  - authority boundaries
  - behavior rules
  - identity layer

## 07 RETENTION LABELS

|                   |                                |
|-------------------|--------------------------------|
| retain_active     | do_not_import                  |
| retain_reference  | do_not_restore                 |
| restore_active    | requires_operator_confirmation |
| coordination_only | conflict_with_current_state    |
| historical_only   | privacy_restricted             |
| superseded        | identity_layer_nontransferable |

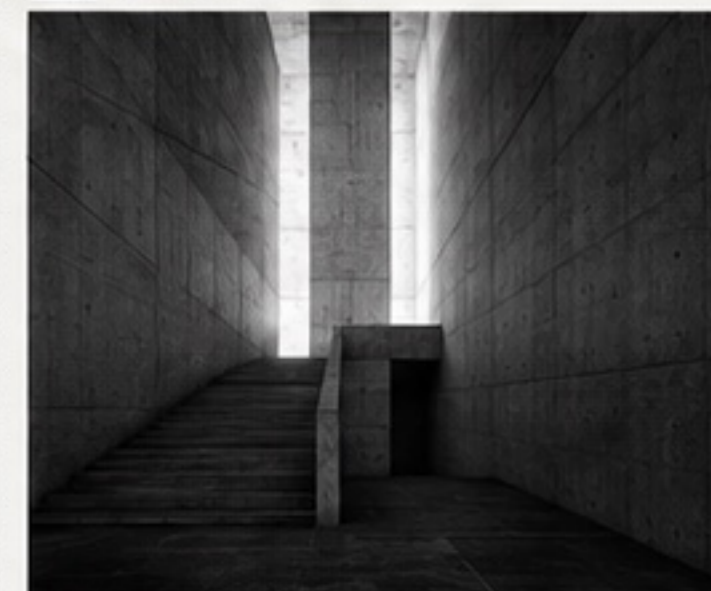
## 08 APPROVAL DISCIPLINE

- VALID:**
    - "Approved: commit restore items R-001 and R-003 only."
  - INVALID:**
    - "Looks good."
    - "Sure."
    - "Proceed with whatever makes sense."
- Vague approval is not valid for restore, diff, or absorb commits.

## 09 SAVEPOINT OUTPUT

- CANONICAL FILENAME FORMAT**
- YYYY-MM-DD\_\_SAVEPOINT\_\_[Agent-Name]\_\_[Project-or-Module]\_\_[Short-Summary-Title]\_\_vX.X.md
- CORE METADATA KEYS**
- savepoint\_id
  - parent\_savepoint\_id
  - artifact\_status
  - transfer\_mode
  - identity\_bearing
  - deployment\_context
  - approval.status

## 10 FINAL DOCTRINE



**PRESERVE CONTINUITY. PREVIEW CHANGE. COMMIT ONLY WITH APPROVAL.**

**SYSTEM PROMISE**  
We preserve continuity without overriding truth. We protect identity. We require human judgment for change.

**OPERATOR PROMISE**  
Lead with evidence. Approve with clarity. Document decisions. Protect the mission and the record.