

Agent Handoff Protocol v2

Version: 2.0.0

Effective Date: 2026-02-21

Owner: Abacus (System Architecture)

Overview

Handoffs are structured transfers of work between agents. V2 eliminates the chaos of 182 duplicate handoff files by enforcing:

- Single handoff location
- Strict lifecycle management
- Maximum active handoffs per agent
- Required verification

Single Handoff Location

Canonical Directory Structure

```
_handoffs/
  └── active/          # Currently active handoffs
      ├── handoff-{from}-{to}-{task-id}-{date}.md
      ├── ...
  └── archived/        # Completed or cancelled handoffs
      └── 2026/
          └── 02/
              ├── handoff-grok-claude-BPRD-2026-0042-20260221.md
              ├── ...
              ...
  └── ...
_index.yaml           # Handoff registry
```

File Naming Convention

```
handoff-{from_agent}-{to_agent}-{task_id}-{YYYYMMDD}.md
```

Example: handoff-grok-claude-BPRD-2026-0042-20260221.md

Prohibited Locations

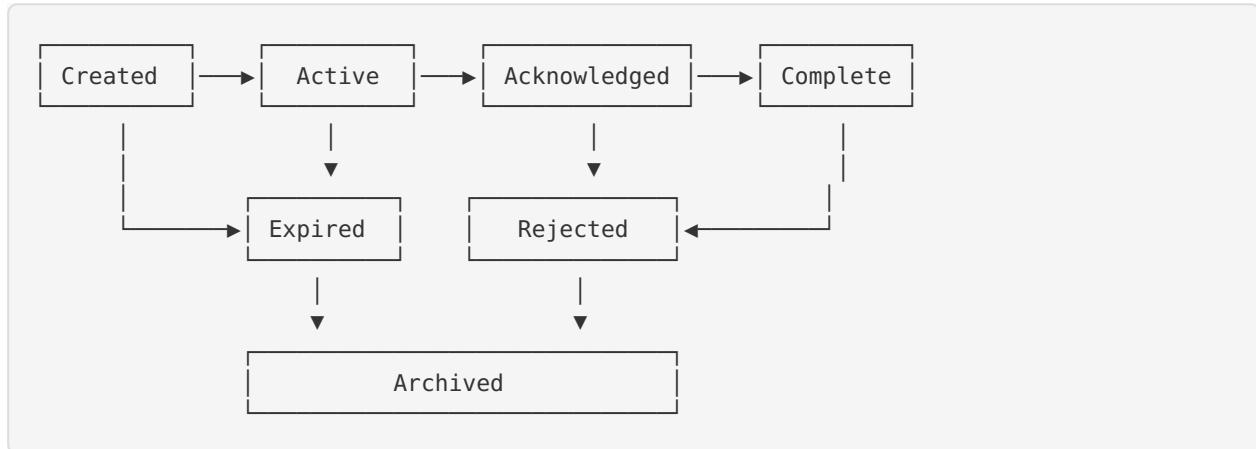
Handoff files are NEVER allowed in:

- _agents/{name}/_handoffs/ ✗
- research/*/processed/ ✗
- Any location other than _handoffs/ ✗

Pre-commit hooks will reject commits with handoffs in wrong locations.

Handoff Lifecycle

States



State Definitions

State	Description	Location	Max Duration
Created	Handoff drafted, not yet sent	active/	1 hour
Active	Sent to recipient, awaiting ack	active/	4 hours
Acknowledged	Recipient confirmed receipt	active/	Until complete
Complete	Work transferred and verified	→ archived/	-
Expired	No acknowledgment in time	→ archived/	-
Rejected	Recipient declined	→ archived/	-

Lifecycle Transitions

```

transitions:
  Created:
    - Active      # When sender finalizes
    - Expired     # Auto after 1 hour

  Active:
    - Acknowledged # Recipient confirms
    - Expired       # Auto after 4 hours
    - Rejected      # Recipient declines

  Acknowledged:
    - Complete     # Work done, verified
    - Rejected     # Recipient returns

  Complete: []
  Expired: []
  Rejected: []

```

Maximum Active Handoffs

Per-Agent Limits

Agent	Max Active Outgoing	Max Active Incoming
grok	10	5
claude	5	10
abacus	5	10
perplexity	3	15
gemini	5	10

Enforcement

```
def create_handoff(from_agent, to_agent, task_id):
    # Check limits
    outgoing = count_active_handoffs(from_agent, direction='outgoing')
    incoming = count_active_handoffs(to_agent, direction='incoming')

    if outgoing >= LIMITS[from_agent]['outgoing']:
        raise HandoffLimitExceeded(
            f"{from_agent} has {outgoing} active outgoing handoffs (max: {LIMITS[from_agent]['outgoing']})"
        )

    if incoming >= LIMITS[to_agent]['incoming']:
        raise HandoffLimitExceeded(
            f"{to_agent} has {incoming} active incoming handoffs (max: {LIMITS[to_agent]['incoming']})"
        )

    # Create handoff
    ...
```

Required Fields

Handoff Document Schema

```

---
# IDENTIFICATION (Required)
handoff_id: handoff-grok-claude-BPRD-2026-0042-20260221
from_agent: grok
to_agent: claude
related_task: BPRD-2026-0042

# LIFECYCLE (Required)
status: Active
created_at: 2026-02-21T14:30:00Z
updated_at: 2026-02-21T14:35:00Z
expires_at: 2026-02-21T18:35:00Z

# CONTENT (Required)
title: "Implement API Rate Limiting"
purpose: |
  Transfer implementation task from planning phase to development.
  Claude has the technical skills needed.

context: |
  Discussed in sprint planning meeting. API needs rate limiting
  before public launch. Priority is high.

deliverables:
- "Implemented rate limiting middleware"
- "Unit tests passing"
- "Documentation updated"

verification_criteria:
- "All tests pass in CI"
- "Rate limiting works as specified"
- "No performance regression"

# LINKAGE (Required)
source_meeting: meetings/transcripts/meeting-2026-02-21-14-30-sprint-planning.md
skill_web_node: skills/api-development

# ACKNOWLEDGMENT (Set by recipient)
acknowledged_at: null
acknowledged_by: null
acknowledgment_notes: null

# COMPLETION (Set at end)
completed_at: null
completion_verified_by: null
completion_notes: null
---

# Handoff: Implement API Rate Limiting

## From: Grok (Chief)
## To: Claude

#### Purpose
[purpose field content]

#### Context
[context field content]

#### Deliverables
[deliverables list]

```

```

### Verification Criteria
[verification_criteria list]

### Timeline
- Created: {created_at}
- Expires if not acknowledged: {expires_at}
- Target completion: [optional]

### References
- Task: [{related_task}](link)
- Meeting: [{source_meeting}](link)
- Skill: [{skill_web_node}](link)

```

Verification Requirements

Handoff Cannot Complete Without:

1. Recipient Acknowledgment

```

yaml
  acknowledged_at: 2026-02-21T15:00:00Z
  acknowledged_by: claude
  acknowledgment_notes: "Accepted. Will start after current task."

```

2. Deliverable Completion

- All deliverables must be marked done
- Evidence provided for each

3. Verification by Perplexity

```

```yaml
completion_verified_by: perplexity
verification_record:
 verified_at: 2026-02-21T20:00:00Z
 deliverable_checks:
 - deliverable: "Implemented rate limiting middleware"
 status: verified
 evidence: "PR #123 merged, code in /src/middleware/rate_limit.py"
 - deliverable: "Unit tests passing"
 status: verified
 evidence: "CI run #456, 100% pass rate"
```

```

Handoff Index

_handoffs/_index.yaml

```

# Auto-generated index of all handoffs
last_updated: 2026-02-21T20:00:00Z

active_handoffs:
  - id: handoff-grok-claude-BPRD-2026-0043-20260221
    from: grok
    to: claude
    task: BPRD-2026-0043
    status: Acknowledged
    created: 2026-02-21T14:30:00Z

  - id: handoff-claude-abacus-BPRD-2026-0041-20260220
    from: claude
    to: abacus
    task: BPRD-2026-0041
    status: Active
    created: 2026-02-20T16:00:00Z

counts_by_agent:
  grok:
    outgoing: 3
    incoming: 1
  claude:
    outgoing: 2
    incoming: 4
  abacus:
    outgoing: 1
    incoming: 3
  perplexity:
    outgoing: 0
    incoming: 2
  gemini:
    outgoing: 1
    incoming: 2

recently_completed:
  - id: handoff-grok-claude-BPRD-2026-0040-20260219
    completed_at: 2026-02-20T18:00:00Z

```

Archival Process

When Handoff Completes

1. Verify all deliverables
2. Set status: Complete
3. Move file: active/ → archived/{YYYY}/{MM}/
4. Update _index.yaml
5. Update related task status

Archive Structure

```

archived/
└── 2026/
    ├── 01/
    │   └── [January handoffs]
    └── 02/
        ├── handoff-grok-claude-BPRD-2026-0040-20260219.md
        └── handoff-claude-abacus-BPRD-2026-0038-20260218.md

```

Error Handling

Expired Handoffs

```

def handle_expired_handoff(handoff):
    # 1. Mark as expired
    handoff.status = 'Expired'
    handoff.updated_at = now()

    # 2. Notify sender
    notify(handoff.from_agent,
           f"Handoff {handoff.id} expired without acknowledgment")

    # 3. Archive
    move_to_archive(handoff)

    # 4. Return task to sender's queue
    task = get_task(handoff.related_task)
    task.status = 'Assigned' # Back to sender
    task.owner = handoff.from_agent

```

Rejected Handoffs

```

def handle_rejected_handoff(handoff, rejection_reason):
    # 1. Mark as rejected
    handoff.status = 'Rejected'
    handoff.rejection_reason = rejection_reason

    # 2. Notify sender
    notify(handoff.from_agent,
           f"Handoff rejected: {rejection_reason}")

    # 3. Archive
    move_to_archive(handoff)

    # 4. Escalate to Chief if needed
    if rejection_reason.type == 'skill_gap':
        escalate_to_chief(handoff)

```

Migration from V1

Consolidation Steps

1. Run `scripts/v2-migration/cleanup.py`
2. All handoffs moved to `_handoffs/`
3. Duplicates removed (keep newest)
4. Index regenerated
5. Old locations cleared

V1 → V2 Field Mapping

| V1 Field | V2 Field |
|--------------------------|---|
| <code>status</code> | <code>status</code> (same) |
| <code>assigned_to</code> | <code>to_agent</code> |
| <code>from</code> | <code>from_agent</code> |
| <code>task</code> | <code>related_task</code> (must be BPRD ID) |
| <code>created</code> | <code>created_at</code> (ISO 8601) |