Below is the FINAL DELIVERABLE for the requested scope.

📌 CLAWBOLT — Final Phase Deliverables

---

1. Project Rationale

CLAWBOLT is designed to bridge human authority, AI assistance, and system-level automation in a controlled, observable, and reversible way.

Modern AI tools can reason, but must not act without consent when actions involve:

- File access
- System modification
- Long-running execution
- Privileged (sudo) commands

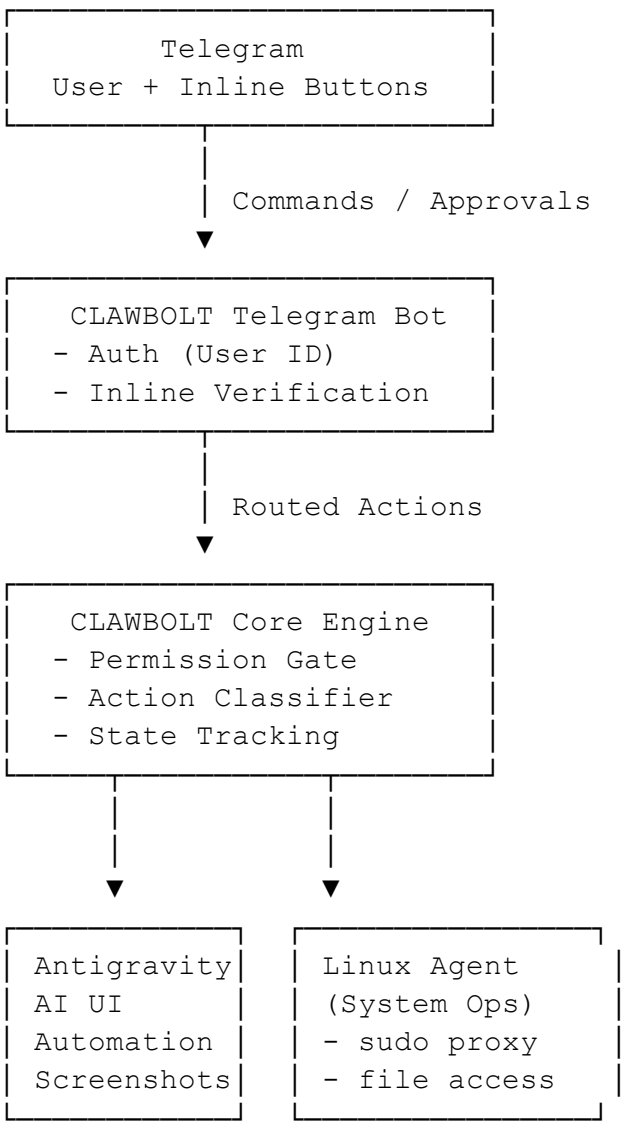
CLAWBOLT enforces a human-in-the-loop model, using Telegram as the decision surface and communication channel, while Antigravity AI provides reasoning and guidance — never silent execution.

This architecture ensures:

- Accountability
- Transparency
- Auditability
- Remote yet safe system control

---

2. Architecture Diagram (Logical)



---

3. Agent Action Verification (Inline Buttons)

Trigger Conditions

When Antigravity AI requests:

- File access (read/write/delete)
- System changes
- Long execution (> configurable threshold)

- Any sudo command

Telegram Inline UI

⚠️ AI REQUESTS HUMAN APPROVAL

```
Action Type: System Change
Requested By: Antigravity AI
Command Summary: Modify system configuration
```

[✅ ALLOW ONCE]
[🔁 ALLOW THIS CONVERSATION]
[❌ DENY]

Behavior

- ALLOW ONCE
    - Executes only the current request
    - Permission expires immediately
- ALLOW THIS CONVERSATION
    - Temporarily whitelists similar actions
    - Automatically revoked on conversation end
- DENY
    - Blocks execution
    - AI is notified of rejection

All decisions are logged.

---

4. Sudo & Password Execution Policy

When sudo is Required

- System configuration changes
- Service restart
- Package install
- Privileged file access

Execution Flow

1. AI proposes action
2. CLAWBOLT classifies it as privileged
3. Telegram verification buttons appear
4. If approved:
    - User is prompted for system password
5. Password is:
    - Injected via stdin (echo "$PASS" | sudo -S)
    - Stored in memory only
    - Time-limited
6. Command executes
7. Output + screenshot sent to Telegram

❗ No sudo command is ever executed without explicit user consent

---

5. systemd Autostart Service

File: /etc/systemd/system/clawbolt.service

```
[Unit]
Description=CLAWBOLT AI Control Agent
After=network.target graphical.target

[Service]
Type=simple
User=clawbolt
WorkingDirectory=/opt/clawbolt
ExecStart=/usr/bin/python3 core/main.py
Restart=always
RestartSec=5
EnvironmentFile=/opt/clawbolt/.env

[Install]
WantedBy=multi-user.target
```

Purpose

- Automatically starts CLAWBOLT on boot
- Survives crashes
- Ensures Telegram control is always available

---

6. Hardened sudoers Configuration

File: /etc/sudoers.d/clawbolt

```
clawbolt ALL=(ALL) NOPASSWD: \
/usr/bin/systemctl restart antigravity, \
/usr/bin/systemctl status antigravity

# Everything else requires password
```

Security Notes

- Minimal whitelisting
- No blanket NOPASSWD
- All other privileged actions trigger password request + Telegram approval

---

7. Full Project README.md (Final)

```
#  CLAWBOLT

CLAWBOLT is a Telegram-driven control system for Antigravity AI and
Linux system automation,
designed with strict human-in-the-loop execution.

## Features
- Real-time AI interaction via Telegram
- Screenshot-based transparency
- Human approval for sensitive actions
- Secure sudo execution with user consent
- Agent commands independent of AI availability

## Safety Model
- No silent execution
- No persistent password storage
- Inline approval buttons for all risky actions
- Full visibility through screenshots and logs

## Supported Actions
- AI-assisted reasoning (Antigravity)
- File inspection
- System monitoring
- Controlled system changes
- Project snapshot export

## Requirements
- Linux Mint 22.3
- Python 3.10+
- Telegram Bot Token
- Antigravity AI installed locally

## Startup
```

```bash
sudo systemctl enable clawbolt
sudo systemctl start clawbolt
```

Disclaimer

CLAWBOLT is designed for authorized systems only.

The user retains full responsibility for all executed actions.

---

8. Work Breakdown Structure (WBS)

Phase 1 — Control Plane
- Telegram bot
- User authentication
- Command routing

Phase 2 — AI Integration
- Antigravity UI automation
- Screenshot capture
- Response streaming

Phase 3 — Verification Layer
- Action classification
- Inline approval UI
- Conversation-scoped permissions

Phase 4 — Privilege Control
- sudo proxy
- Password handling
- Secure execution flow

Phase 5 — Deployment & Hardening
- systemd service
- sudoers hardening
- Documentation
- Audit logging

---

9. Final Outcome

CLAWBOLT achieves:

✅ Remote AI-assisted control
✅ Explicit human authorization
✅ Safe privileged execution
✅ Transparent system visibility

**AI assists.
Humans decide.
Systems obey.**

If you want next:
- Inline keyboard Python code
- Action classifier logic
- sudo proxy implementation
- Threat model

Just say **which one**.