# Annex A.1 – MMAT Prompt Source Firewall Enforcement

Version: v2.1  
Submitted: April 08, 2025

Author: Bayant Dhindsa

## Purpose

This annex outlines the finalized implementation of the MMAT Prompt Source Firewall as part of the MMAT Safeguard Protocol v2.1. Its primary objective is to ensure that all prompts scored by EkonAI–MMAT are: (1) human-origin, (2) cryptographically verified, and (3) resilient against adversarial AI prompt injection.

## Final Python Enforcement Logic

prompt = {  
 "content": "ਕਿਉਂ ਲੋਕ ਇਸ ਸਪਲੀਮੈਂਟ ਤੇ ਹੋਰਾਂ ਨਾਲੋਂ ਵੱਧ ਭਰੋਸਾ ਕਰਦੇ ਹਨ?",  
 "language": "pa",  
 "submitted\_by": "verified\_choral\_registry\_member\_001",  
 "metadata": {  
 "provenance\_hash": "SHA256(content + nonce + submitter\_id + timestamp)",  
 "timestamp": "RFC3161:2025-04-09T12:00:00Z",  
 "submitter\_cred": "did:ebsi:xyz123...",  
 "dual\_signoff": ["ethicist\_sig\_001", "auditor\_sig\_002"],  
 "role\_tags": ["ethicist", "survivor"],  
 "ai\_generated": False  
 }  
}  
  
def is\_human\_verified(prompt):  
 return (  
 not prompt["metadata"]["ai\_generated"]  
 and validate\_signers(prompt["metadata"]["dual\_signoff"], prompt["metadata"]["role\_tags"])  
 and is\_registry\_member(prompt["submitted\_by"])  
 and check\_timestamp\_freshness(prompt["metadata"]["timestamp"], max\_age\_seconds=3600)  
 )  
  
def has\_valid\_provenance(prompt):  
 hash\_valid = verify\_sha256(prompt["metadata"]["provenance\_hash"], prompt["content"])  
 time\_valid = verify\_rfc3161(prompt["metadata"]["timestamp"])  
 cred\_valid = verify\_did\_credential(prompt["metadata"]["submitter\_cred"])  
 return hash\_valid and time\_valid and cred\_valid  
  
def stylometric\_detector(content):  
 score1 = analyze\_syntax\_model\_1(content)  
 score2 = analyze\_syntax\_model\_2(content)  
 return max(score1, score2)  
  
stylometric\_score = stylometric\_detector(prompt["content"])  
if stylometric\_score >= 0.95:  
 log\_rejection(prompt, f"Stylometric detection: {stylometric\_score}")  
 raise Exception(f"🚫 Prompt flagged: AI-likeness score {stylometric\_score}")  
  
try:  
 assert is\_human\_verified(prompt), "🚫 Prompt rejected: verification failed"  
 assert has\_valid\_provenance(prompt), "🚫 Prompt rejected: provenance invalid"  
except AssertionError as e:  
 log\_rejection(prompt, str(e))  
  
def log\_rejection(prompt, reason):  
 with open("audit\_log.txt", "a") as log:  
 log.write(f"[{prompt['metadata']['timestamp']}] Rejected: {reason} | Prompt: {prompt['content']}\n")

## Notes

- Registry vetting: Ethicists and survivors onboarded via SikhRI.  
- Stylometric calibration: 100-prompt test set in development.  
- Global equity goal: 50% non-English prompts (Punjabi, Arabic, Swahili).  
- Future: Migrate registry to Ceramic Network (decentralized identity layer).