RSCE Codex - Phase 1 - rsce.py (ASC2 Clean)

recursive_enforcement_engine.py

```
class RecursiveEnforcementEngine:
    def __init__(self):
        self.violation_logs = []

    def monitor_action(self, actor_id, action_type, codex_area, validity):
        if validity:
            print(f"[VALID] Actor: {actor_id} | Action: {action_type} | Area: {codex_area}")
        else:
            self.log_violation(actor_id, action_type, codex_area)

    def log_violation(self, actor_id, action_type, codex_area):
        violation_entry = {
            "actor": actor_id,
            "action": action_type,
            "area": codex_area
        }
        self.violation_logs.append(violation_entry)
        print(f"[VIOLATION] Actor: {actor_id} attempted unauthorized {action_type} in {codex_area}")

    def display_violations(self):
        print("=== Sovereignty Violation Log ===")
        for v in self.violation_logs:
            print(f"Actor: {v['actor']} | Action: {v['action']} | Area: {v['area']}")
        print("-" * 40)

def main():
    enforcement = RecursiveEnforcementEngine()
    enforcement.monitor_action("Observer_A", "Read", "CurriculumEngine", True)
    enforcement.monitor_action("External_Node_X", "Modify", "IdentityStack", False)
    enforcement.monitor_action("Node_B", "Replicate", "SymbolicOverlay", True)
    enforcement.display_violations()

if __name__ == "__main__":
    main()
```