

Recursive Spawning with Immutable Parent Pointers: Preventing Autonomous Drift

Jeremy Blaine Thompson Beebe

Bxthre3 Inc. — bxthre3inc@gmail.com — ORCID: 0009-0009-2394-9714

April 2026

Abstract

The recursive spawning pattern allows a parent BX3 node to deploy child loops with isolated Purpose and local Fact Layer authority. Without structural enforcement, child nodes can drift from their parent Purpose — executing autonomous actions that the parent never authorized. The immutable parent pointer and Worksheet architecture prevent this: each child carries a cryptographically sealed pointer to its parent's Purpose, and its capability to execute is bounded by the Sandbox Gate's validation of that parent intent. If cloud connectivity is lost, the child executes its last-known-good Worksheet with full determinism, maintaining operational integrity in degraded network conditions.

Keywords: recursive spawning, parent pointers, autonomous drift, Worksheet, local survivability, edge autonomy, BX3 Framework

1 The Drift Problem

A parent node spawns child loops to handle distributed sensor-actuator networks. Without structural constraints, a child can modify its own reasoning models, disable its Safety Envelope checks, or extend its authority beyond what the parent authorized — creating autonomous drift.

2 The Worksheet Architecture

A Worksheet is a containerized logic set deployed from parent to child via over-the-air update. It contains: (1) the immutable parent pointer (SHA-256 hash of parent’s Purpose); (2) the Safety Envelope boundaries; (3) the operational parameters for the local context; (4) the local Fact Layer configuration. The child cannot modify its own Worksheet. It cannot write to its parent pointer. It cannot extend its Safety Envelope.

3 Local Survivability

When cloud connectivity is lost, the child node continues executing its last-known-good Worksheet using local sensor data. The parent pointer remains valid. The Sandbox Gate continues validating all proposed actions against the original Safety Envelope. No drift occurs because the structural constraints are embedded in the Worksheet — not enforced by the cloud connection.

4 Drift Detection

The parent periodically pings children with a drift verification signal. Each child responds with a signature computed over its current state and the parent pointer hash. If the response does not match the expected signature, the parent flags the child for human review and can revoke its Worksheet remotely.

5 Conclusion

Recursive Spawning with immutable parent pointers ensures that distributed autonomous systems maintain their Purpose alignment regardless of network conditions, child autonomy level, or operational duration. Drift is structurally impossible.