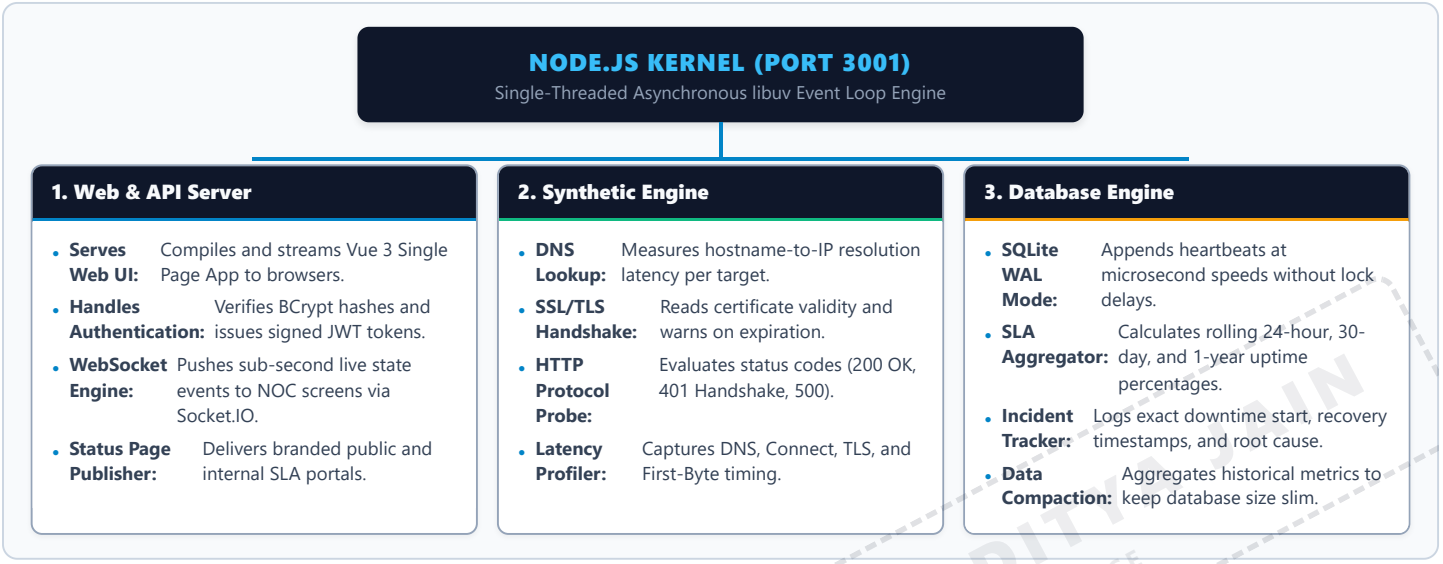
Node.js Kernel Architecture & System Dynamics

Synthetic Telemetry Engine, Failure Modes, and Autonomous Recovery (State NOC Perimeter)

SERVICE PORT
TCP / 3001

Technical Purpose: On server **STATE-NOC-NODE**, Node.js does not merely "ping" network adapters. It functions as an integrated **three-tier asynchronous engine**: serving the web application and real-time WebSocket feeds, executing deep multi-protocol synthetic checks (DNS, TCP, TLS x509, HTTP), and writing high-frequency time-series heartbeats into an embedded SQLite engine with zero table-locking delay.



1. WHAT EXACTLY IS NODE.JS DOING? (PING VS. SYNTHETIC APPLICATION PROBING)

Deep Inspection

CAPABILITY	STANDARD ICMP "PING"	NODE.JS SYNTHETIC PROBE (UPTIME KUMA)
Layer Tested	Network Layer (L3 - ICMP Echo)	**Application Layer (L7 - Full HTTP/HTTPS & WebForms Handshake)**
Detects Web Server Crash?	NO (Host responds even if IIS / Web App is dead)	YES (Detects HTTP 500, 502, 503, or connection refusions)
Detects Database Outages?	NO (Network adapter is oblivious to database locks)	YES (Flags error strings or backend database failure payloads)
SSL Certificate Expiry?	NO (Does not perform cryptographic handshakes)	YES (Extracts exact expiration dates and alerts 30d/14d/7d prior)
Response Time Accuracy	Only network wire propagation latency	**True citizen experience latency (DNS + TLS + Server Compute)**

SANITIZED CASE STUDY · ADITYA SHARMA · STATE NOC SURVEILLANCE · STATE NOC PERIMETER

Network Disconnection Scenarios & Auto-Heal Mechanics

Behavior During Internet Drops, DNS Outages, and Server Restarts

FAULT TOLERANCE
HIGH RESILIENCE

2. FAILURE SCENARIO MATRIX: WHAT HAPPENS IF CONNECTIVITY BREAKS?

Scenario Analysis

FAULT SCENARIO	IMPACT ON DASHBOARD	IMPACT ON MONITORED PORTALS	TECHNICAL EXPLANATION
Server Internet Disconnected (External ISP Fiber Cut)	DASHBOARD WORKS 100% (Accessible over Intranet)	• Internal Portals: GREEN (UP) • Public Portals: RED (DOWN)	The dashboard and database run 100% locally on STATE-NOC-NODE. Local Bihar WAN traffic succeeds, while external public requests fail with connection timeouts, accurately alerting the NOC to an external ISP failure.
Server DNS Disconnected (DNS Server Unreachable)	DASHBOARD WORKS 100% (Direct IP Access)	• IP Targets: GREEN (UP) • Domain Targets: RED (DOWN)	Node.js fails at Step 1 (DNS Resolution) with `ENOTFOUND` for domain names like `[REDACTED-PORTAL].internal`. Targets addressed by IP (like `STATE-NOC-NODE`) continue probing without interruption.
Host Server Rebooted (Windows Updates / Power Cycle)	AUTO-RESTORES < 120S (Zero Admin Needed)	All 5 Portals Resume Immediately	The Windows Task Scheduler watchdog detects port 3001 is closed and calls `kuma_service.php` to spawn a fresh detached `node.exe` process with memory ceiling intact.

3. WHAT ARE THE BACKGROUND SCRIPTS ACTUALLY DOING?

Two-Tier Watchdog

1. PORTAL_MONITOR_DAEMON.PY (SCHEDULED WATCHDOG)

- **Execution Trigger:** Runs automatically every 2 minutes via Windows Task Scheduler.
- **Health Check:** Issues a rapid HTTP probe to `http://CNI-NOC-KERNEL:3001`.
- **Action on Failure:** If port 3001 is unresponsive, it issues an autonomous POST signal to `kuma_service.php?action=start` to resurrect Node.js immediately.

2. KUMA_SERVICE.PHP (IIS FASTCGI SUPERVISOR)

- **Execution Context:** Microsoft IIS FastCGI on STATE-NOC-NODE (Port 80).
- **Process Table Inspection:** Queries active Windows TCP sockets using `netstat`.
- **Detached Process Spawner:** Launches `node.exe` in the background with flags: `--max-old-space-size=512` and `--use-system-ca`.

4. EXECUTIVE TALKING POINTS (FOR MANAGEMENT BRIEFING)

Summary

Summary to present to Leadership:

- *"Node.js runs as a lightweight, 165 MB synthetic application engine on STATE-NOC-NODE that tests real citizen experiences, not just server cables."*
- *"If our Internet connection drops, the NOC dashboard continues running locally on our intranet, instantly highlighting which external portals lost connectivity."*
- *"Our two-tier background watchdog guarantees that if the server ever reboots, the entire monitoring system heals itself in under 2 minutes with zero human intervention."*

SANITIZED CASE STUDY — ADITYA JAIN
STATE NOC SURVEILLANCE