

# **MaxOneOpen - Deployment Blueprint & Toolchain Overview**

*Document ID: v3.4-BUILD-001*

Document ID	v3.4-BUILD-001
Title	Deployment Blueprint & Toolchain Overview – MaxOneOpen
Version	1.0
Date	2025-03-31
Author	MaxOne Documentation Unit (GPT-Validated)
Document Type	Operational Build Documentation

### FOUNDATION – Purpose & Scope

This document provides a practical, architecture-compliant blueprint for deploying MaxOneOpen. It includes tooling assumptions, node setup guidance, and essential control interfaces for teams aiming to replicate or fork MaxOne-based systems without vendor assistance.

### EXECUTION – Toolchain & Build Requirements

- **Supported Environments:** Linux (Debian/Ubuntu), MacOS, containerized build pipelines
- **Build Language Compatibility:** Rust (preferred), Python (for orchestration), Bash (bootstrap)
- **Essential Tooling:** Docker, Git, Curl, GPG, ZK-verification module (provided)
- **Recommended Tools:** Visual Studio Code, Docusaurus (for documentation forks)
- **CI/CD Option (optional):** GitHub Actions, DroneCI, or self-hosted GitRunner

### STACK – Reference Node Specification

Component	Minimum Requirement
CPU	4-core ARM or x86_64 (Intel/AMD)
Memory	8 GB RAM (16 GB recommended for orchestration)
Storage	50 GB SSD (local twin database + logs)
OS	Linux Kernel 5.x or newer (Debian preferred)
Connectivity	1 Gbit LAN or redundant WLAN (for edge mesh)

### EXECUTION – Deployment Flow Overview

1. Clone official MaxOneOpen release repo from trusted GIT URL.
2. Verify integrity hash via provided SHA256 and GPG signature.
3. Initialize edge node environment (install dependencies via ./init.sh).
4. Load and activate base manifest using ./bootstrap-twin.sh.
5. Register twin instance to mesh via discovery script ./register-peer.sh.
6. Validate manifest linkage with ./verify-chain.sh.
7. Monitor runtime logs using ./live-twin-log.sh or equivalent dashboard.

## **FINAL – CTO-Relevant Conclusion**

This document bridges the architectural blueprint with real-world deployments. Any experienced infrastructure team can build and activate a fully compliant MaxOneOpen node using the tools and flows described here. Forkability, verifiability and runtime integrity are preserved by design – without cloud dependencies.

Status: Operational blueprint for implementation teams – GPT-certified

v3.4-BUILD-001 | Status: Final | Version 1.0