

Validator Operator Guide

Platform & Architecture · Published by UnyKorn LLC (Wyoming), technology provider · September 2026

DEMONSTRATION DOCUMENT — NOT AN OFFERING, NOT A BOUND CONTRACT. All entities, identifiers, reserves, serials, vaults, policies, counterparties and figures in this document are illustrative and synthetic. UnyKorn LLC is not a broker-dealer, exchange, alternative trading system, investment adviser, bank, trust company, qualified custodian, money transmitter, insurer or issuer, and does not hold client assets or funds. No entity described here is registered with the SEC, FINRA, the FCA or any other regulator. No refinery, vault operator, insurer, broker, auditor or qualified person named or implied is a counterparty to UnyKorn LLC. Nothing here is an offer to sell or a solicitation of an offer to buy any security, or investment, legal, tax or accounting advice.

DGX VALIDATOR OPERATOR & NODE INFRASTRUCTURE GUIDE

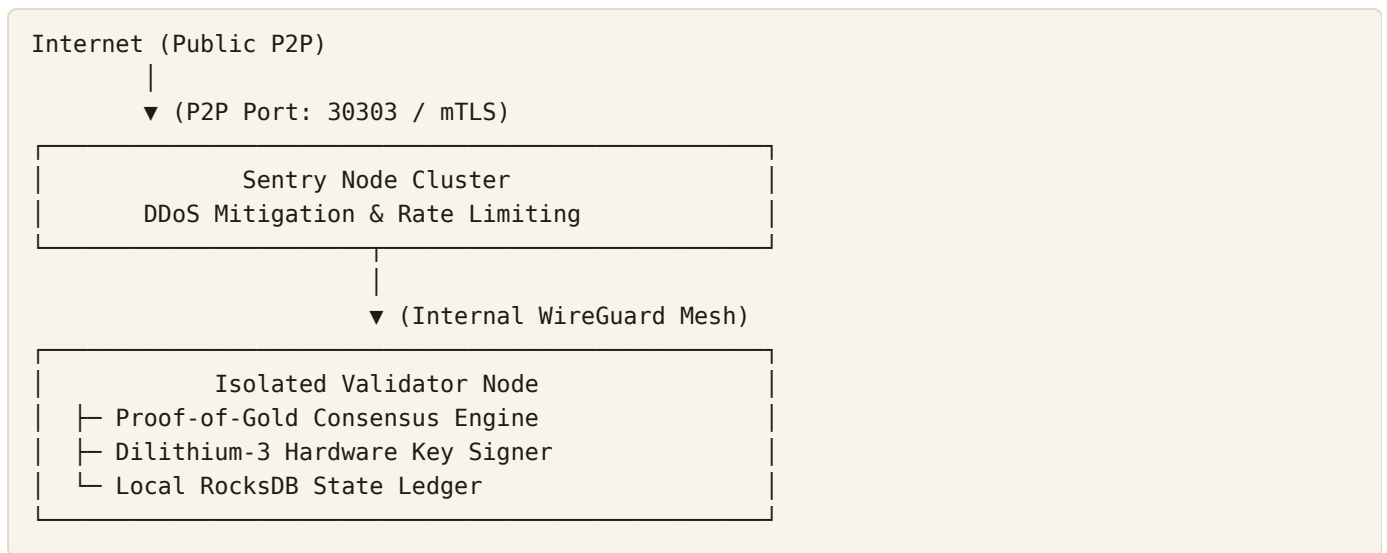
Proof-of-Gold Byzantine Fault Tolerance (PoG-QBFT) Node Deployment

1. Minimum Hardware Requirements

For production institutional validators and consensus node operators:

- **CPU:** 16 physical cores / 32 threads (AMD EPYC or Intel Xeon Ice Lake)
- **RAM:** 64 GB ECC DDR4/DDR5
- **Storage:** 2 TB NVMe PCIe 4.0 (sustained IOPS > 50,000)
- **Network:** 1 Gbps redundant symmetric fiber uplink
- **Hardware Security Module (HSM):** YubiHSM 2 or Nitrokey HSM for Dilithium-3 private key storage
- **Operating System:** Ubuntu Server 22.04 LTS or RHEL 9 (hardened kernel)

2. Node Architecture & Isolation



3. Step-by-Step Node Deployment

Step 1: Clone and Compile the Node

```
git clone https://github.com/FTHTrading/DGX.git
cd DGX
cargo build --release -p dgx-node
```

Step 2: Generate Post-Quantum Validator Keypair

```
./target/release/dgx-node keygen --algorithm dilithium-3 --output /etc/dgx/validator.key
```

Step 3: Configure `dgx-node.toml`

```
[network]
listen_address = "0.0.0.0:30303"
network_id = "dgx-mainnet-1"
bootnodes = [
    "/ip4/34.205.29.55/tcp/30303/p2p/QmDGXBootnodeZurich",
    "/ip4/18.211.197.115/tcp/30303/p2p/QmDGXBootnodeLondon"
]

[consensus]
consensus_type = "PoG-QBFT"
block_time_seconds = 1
validator_key_file = "/etc/dgx/validator.key"
min_bonded_gold_oz = 1000

[rpc]
enabled = true
listen_address = "127.0.0.1:8545"
cors_domains = ["https://unykorn.org", "https://unykorn.ai"]

[storage]
database_path = "/var/lib/dgx/chaindata"
```

Step 4: Systemd Service Configuration

Create `/etc/systemd/system/dgx-node.service` :

```
[Unit]
Description=DGX Sovereign L1 Validator Daemon
After=network.target

[Service]
User=dgx
Group=dgx
Type=simple
ExecStart=/usr/local/bin/dgx-node --config /etc/dgx/dgx-node.toml
Restart=always
RestartSec=3
LimitNOFILE=65535

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
sudo systemctl enable --now dgx-node
sudo journalctl -u dgx-node -f
```

4. Slashing Conditions & Validator Governance

- **Double-Signing:** Immediate forfeiture of 20% of bonded gold stake and permanent ejection from validator set.
- **Extended Downtime:** 0.05% stake penalty for every 1,000 unvalidated consecutive blocks.
- **Reserve Default:** If a validator's bonded depository physical gold fails quarterly audit, node voting power drops to zero automatically.