ChatGPT

Gorbagana (Solana Fork) – Network & Developer Stack

Network Status: Gorbagana is a Solana-fork meme blockchain that has moved from testnet into mainnet. Official sources confirm the Gorbagana mainnet went live on **June 28, 2025** [1]. (Prior to that, the project ran two testnet phases: a “Testnet v1” proof-of-concept and then a “Testnet v2 / Devnet” for stable development [2] [3].) Gorbagana publishes its own RPC endpoint and explorer: the **RPC URL** is `https://rpc.gorbagana.wtf/` and the official block explorer is **explorer.gorbagana.wtf** [4]. A chain status dashboard is available at **status.gorbagana.wtf** [5]. (These resources are all listed on the Gorbagana docs site.) In short, Gorbagana is now running live (mainnet) with full network infrastructure – developers can point to the custom RPC and use the block explorer for on-chain data.

Smart-Contract Frameworks: As a **Solana** codebase fork, Gorbagana supports Solana-compatible smart contracts and tooling. The project advertises “full compatibility with Solana Program Library (SPL)” [6], meaning existing Solana programs (written in Rust via Anchor or plain Solana BPF) should run unmodified. In practice, developers **can use the standard Solana toolchain (Rust + Anchor, or other Solana frameworks like Seahorse)** and deploy to Gorbagana by targeting the Gorbagana RPC. (For example, community developers have already built Anchor-based games on Gorbagana testnet.) Solana’s native BPF programs in Rust are supported; newer frameworks (e.g. the Python-based Seahorse for Anchor, or Solidity-to-BPF via Solang) should likewise work as they rely on Solana’s runtime. The docs specifically note that Gorbagana inherits Solana’s runtime (PoH consensus, Sealevel parallel execution, etc.) [7], so any smart contract written for Solana (Anchor, SPL token programs, DeFi contracts, etc.) can be redeployed on Gorbagana. The only planned difference is the rebranding of SPL to “Trash Program Library (TPL)” in the future [6].

Wallet Support: Gorbagana is designed to work with Solana wallets via a custom RPC. The **Backpack Wallet** (desktop extension or mobile app) explicitly supports Gorbagana out of the box: the official docs walk users through adding the Gorbagana network in Backpack by entering the RPC `https://rpc.gorbagana.wtf/` [8]. In fact, “Backpack is currently the primary supported wallet for accessing the Gorbagana network” [9]. (On desktop, Backpack will label native testnet GOR as “SOL” in the UI, but it functions as the gas token on Gorbagana [10].) Other Solana wallets could in principle be pointed at the Gorbagana RPC, though as of now official guidance emphasizes Backpack. Phantom does *not* natively include Gorbagana, but it can show the $GOR token on Solana mainnet (see Phantom token info [11]) – to actually use Phantom for on-chain transactions, you’d need to switch the RPC (which Phantom’s UI does not yet support). In summary, **Backpack (Magic Eden’s wallet) is the recommended client**; developers can also use any Solana wallet adapter or SDK by configuring the custom RPC.

Randomness Oracles: Gorbagana itself does not currently list any built-in randomness source. There is no official Chainlink integration announced. Because Gorbagana runs Solana’s runtime, randomness must come from off-chain oracles as on Solana. The standard approach on Solana is to use a VRF oracle such as Switchboard. In fact, Switchboard “offers a developer-friendly VRF for the Solana ecosystem” [12], and Gorbagana developers can call Switchboard VRF (or similar service) from their programs. Gorbagana’s docs do not mention randomness, so developers should follow Solana best practices: request randomness via a VRF oracle (e.g. Switchboard VRF or a future Chainlink Solana VRF if it arrives) and verify it on-chain. No native on-chain random primitive exists (as on Solana, blockhashes are predictable to validators), so VRF oracles are the recommended solution.

Developer Tools & Docs: Gorbagana provides official developer resources. The **Gorbagana docs site** (https://docs.gorbagana.wtf) is the central developer portal [4]. It includes network access details (RPC, explorer, faucet), FAQ, and a roadmap. There is a **Gorbagana CLI** in development (eventually to manage keys, deploy programs, etc.), but for now developers should use the standard Solana CLI pointed at the Gorbagana RPC [13]. GitHub hosts the code and SDKs: the `gorbagana-dev` organization (https://github.com/gorbagana-dev) contains repos like **gorbagana-web3.js** (a JavaScript SDK for Gorbagana) [14], a pending `gorbagana-spl-token-cli` tool, and a Gorbagana block-explorer frontend. Token standards on Gorbagana follow Solana's SPL model; e.g. the SPL Token program and related CLI/tools are compatible (the team even plans a "Trash Token CLI" for management).

Resources (for developers): Gorbagana's official links include: - **Website/Docs:** docs.gorbagana.wtf – technical documentation and guides [15] [4].
- **Explorer:** explorer.gorbagana.wtf – official block explorer for transactions/blocks [4].
- **RPC Endpoint:** `https://rpc.gorbagana.wtf/` (set this in your CLI or wallet) [16].
- **GitHub:** github.com/gorbagana-dev – source code and SDKs (web3.js, CLI tools, etc.) [14].
- **Wallets:** Backpack Wallet (install from backpack.app and add Gorbagana RPC) [8].
- **Social / Announcements:** Follow the Gorbagana team on X/Twitter (@Gorbagana_chain) and join their Telegram for updates.

In summary, Gorbagana offers a Solana-compatible developer experience: you can deploy Rust/Anchor programs, use Solana wallets (via custom RPC), and interact with the chain using Solana-like tooling. Official docs and repositories provide all the necessary endpoints and instructions [4] [13]. With the mainnet live, developers should point their tools (CLI, Anchor, web3 libraries) at the Gorbagana RPC and proceed much as they would on Solana.

Sources: Gorbagana documentation and announcements [1] [7] [6] [9] [8] [13] [12] [14]. These official sources detail the network status, tech stack, and dev tooling for Gorbagana.

---

[1] Gorbagana (GOR) - Gorbagana Mainnet - 28 Jun 2025 — TradingView News
https://www.tradingview.com/news/coinmarketcal:b0afb2a12094b:0-gorbagana-gor-gorbagana-mainnet-28-jun-2025/

[2] [6] [7] What is Gorbagana? · Gorbagana Documentation
https://docs.gorbagana.wtf/

[3] Testnet v2 (Devnet) · Gorbagana Documentation
https://docs.gorbagana.wtf/testnet-v2-devnet.html

[4] [5] Explorer & Tools · Gorbagana Documentation
https://docs.gorbagana.wtf/network-access/explorer-and-tools.html

[8] [9] [10] [16] Connecting to Testnet · Gorbagana Documentation
https://docs.gorbagana.wtf/network-access/connecting-to-testnet.html

[11] Gorbagana (GOR) Price Chart - Buy and Sell on Phantom
https://phantom.com/tokens/solana/J2Sd7BX5Q4vqdxJvSLNZJgnFnFF11tfNzwBCb29Xpump

[12] Verifiable Randomness Functions | Solana
https://solana.com/developers/courses/connecting-to-offchain-data/verifiable-randomness-functions

[13] CLI Installation & Usage · Gorbagana Documentation
https://docs.gorbagana.wtf/development/cli-installation-and-usage.html

[14] gorbagana dev · GitHub
https://github.com/gorbagana-dev

[15] What is Gorbagana
https://learn.backpack.exchange/articles/what-is-gorbagana