Autonolas

Autonomous Services

Autonomy is dead. Long live autonomy!

'Autonomy' is a buzzword used to sell everything from self-driving cars to text- and image-generating AIs. Today, such apps are centrally owned and controlled. Users yield control to opaque algorithms and forfeit their private data. They increasingly trade away human autonomy for machine autonomy, i.e. self-operating machines and systems.

Autonolas believes autonomous services — software services that continuously operate and require little to no input from humans by design — enable human autonomy best when they are transparent, robust, and decentrally owned and operated.

Smart Contracts **cannot:**

- Do things that require expensive processing
- Run continuously
- Access data not on the same blockchain, like external APIs or other chains

Autonomous Services can:

- Run continuously
- Take action on their own
- Interact with the world outside of blockchains
- Run complex logic

With Autonolas the best AIs will not be exploited by individual billionaires or mega-corporations. Instead, they will be co-owned by collectives of humans and operated as autonomous services to catalyze individual autonomy.
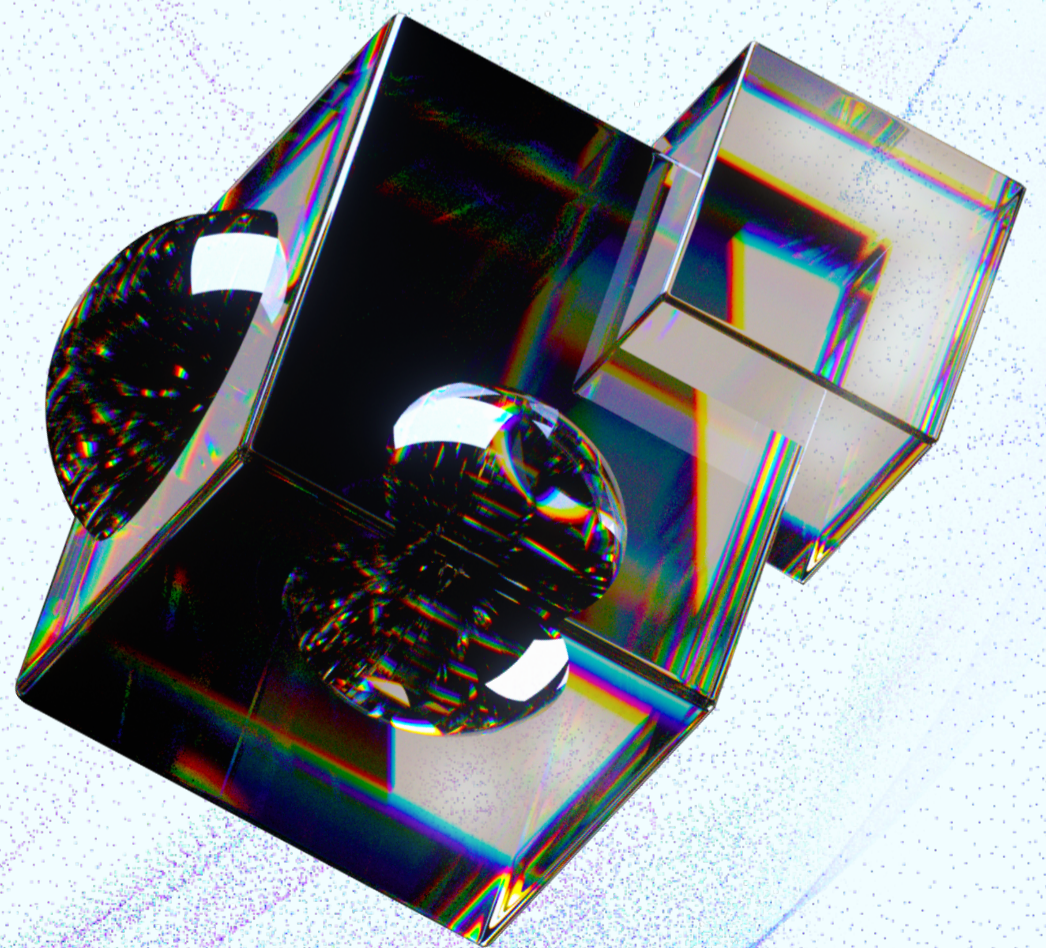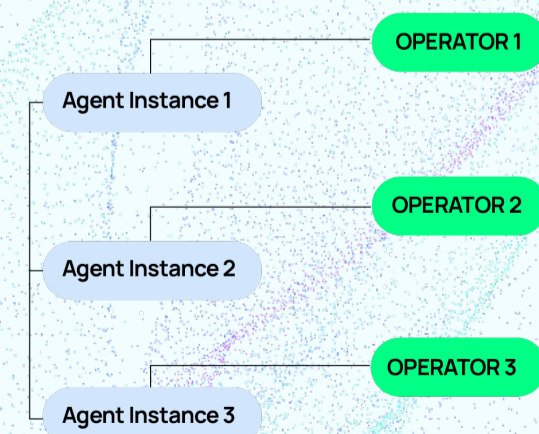
Technical Architecture

Autonolas autonomous software services are embodied as agent services. An agent service is a group of independent computer programs that interact with each other to achieve a predetermined goal. An agent service can be understood as a logically centralized application (namely, there is only one application state and logic) that is replicated in a distributed system.

Agent services are made of code components that can be combined, similarly to Lego bricks, through software composition. This is enabled and incentivized by the on-chain protocol, which facilitates developers publishing and finding code components to build and extend new services. To this end, the on-chain protocol implements a series of registries that enable code components, agents and services to be found, reused and compensated economically.

To sum up, the main elements of the Autonolas tech stack are:

1   **AGENT SERVICES**, which are maintained by a service owner and run by multiple operators, who execute independent agent instances (that run the same code); these instances coordinate through a so-called consensus gadget.

2   **COMPOSABLE AUTONOMOUS APPS,** that are built out of basic applications that are easily extendable and composable into higher order applications.

3   An **ON-CHAIN PROTOCOL**, on a programmable blockchain, that secures the agent services and incentivizes developers to contribute code to this protocol.
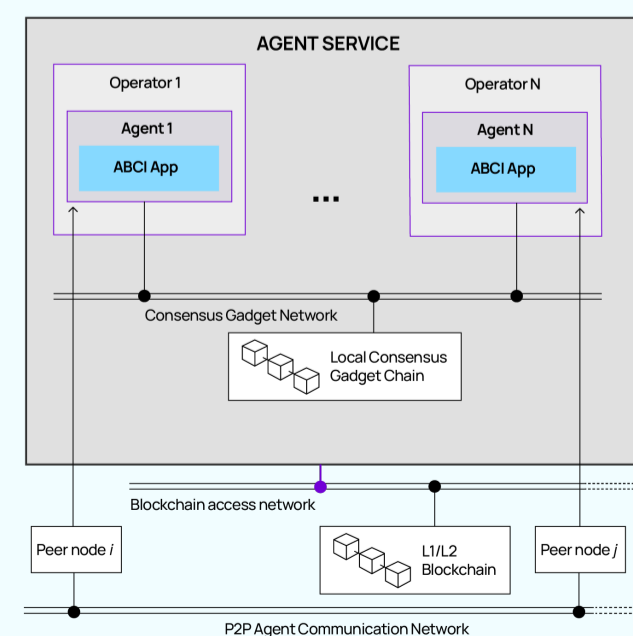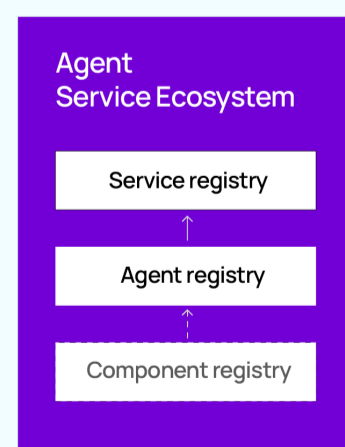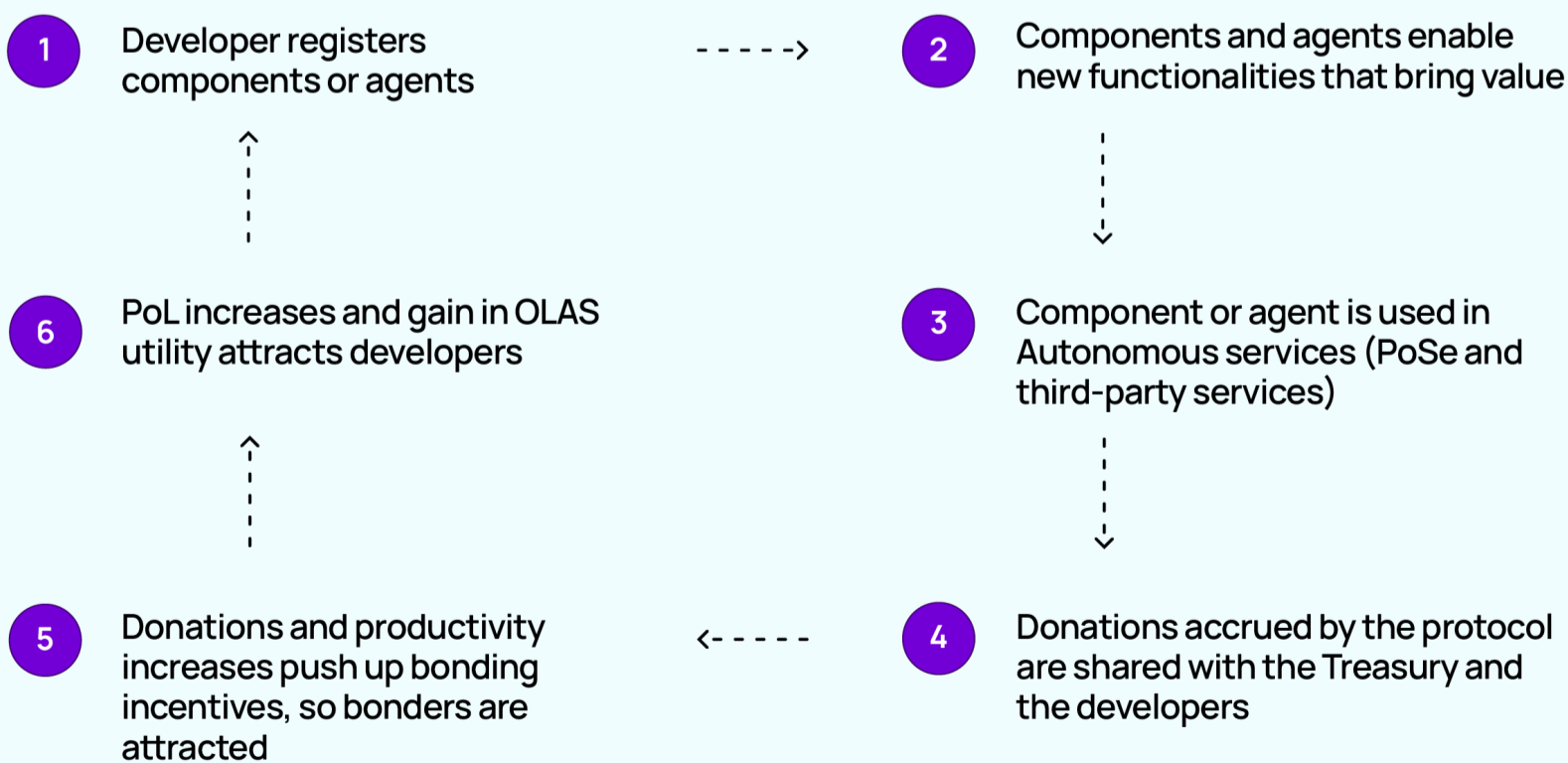
Tokenomics

Autonolas tokenomics focuses on three objectives:

1 **Growing capital and code proportionally**: On-chain mechanisms ensure that the code provided by developers is rewarded according to its usefulness to the services operated on the protocol. The protocol acquires protocol-owned liquidity (PoL) in proportion to code usefulness, which allows the protocol to generate returns, invest in services and guarantee its long-term financial health.

2 **Enabling intra- and inter-protocol composability**: NFTs representing code and services are used to track contributions inside the protocol, accrue rewards and can be used productively as collateral across DeFi.

3 **Incentivizing donation of profits from Protocol-owned Services (PoSe)**: Autonomous services, owned by governance of various DAOs, operated by the ecosystem and developed by agent developers around the world, donate some of their profits to the protocol.
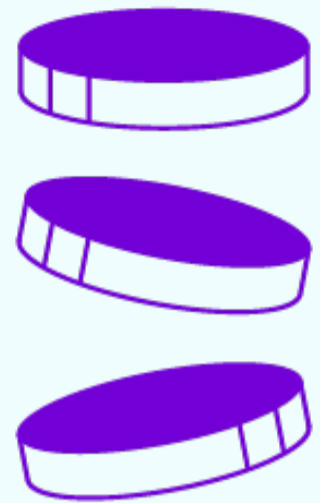
Autonolas protocol is free-to-use for service owners, operators and developers. Service owners are incentivized to donate some of their PoSe profits to the protocol where they are shared primarily with developers, with optionally a small fraction going to protocol treasury. Autonolas tokenomics hence offers **a mechanism for DAOs to reward contributors** for their autonomous services. Additionally, the tokenomics attracts bonders, who can make their capital useful, by pairing it with code.

AUTONOLAS **FLYWHEEL** IS DESIGNED TO WORK AS FOLLOWS:

1 Developer registers components or agents

2 Components and agents enable new functionalities that bring value

6 PoL increases and gain in OLAS utility attracts developers

3 Component or agent is used in Autonomous services (PoSe and third-party services)

5 Donations and productivity increases push up bonding incentives, so bonders are attracted

4 Donations accrued by the protocol are shared with the Treasury and the developers

Autonolas governance will likely arise as a core operator of important PoSe thereby creating both pull (for developers) and push (for capital) factors around the flywheel.
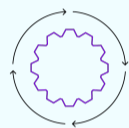
Use Cases for Autonomous Services

We believe a large market for autonomous agent services is emerging. Initially, the primary utility of autonomous services is in improving **DAO Operations**. The main benefits that DAOs see from this adoption are in:

1 Drastic improvements to their **internal operations**, and

2 **Delivering services** that are radically richer, more powerful and more robust

Internal Operations

Treasury Manager
Continuous Parameter Adjustment
Dynamic, Credential-based Payroll
And many more...

Service Delivery

Off-chain Decentralized NFT Orderbook
Cross-chain Yield Super Aggregator
Autonomous NFT Collector

Autonomous services make DAOs more competitive by:

- providing richer means for transparently coordinating human actors

- executing processes with little or no marginal human input

Autonomous services can be composed of three fundamental lego blocks:

Oracles – pull in and perform complex data operations

Keepers – take action on-chain according to rich conditions

Bridges – move value and information between chains

This composability leads to combinatorial expansion and unprecedented new applications.

ORACLES

KEEPERS

BRIDGES
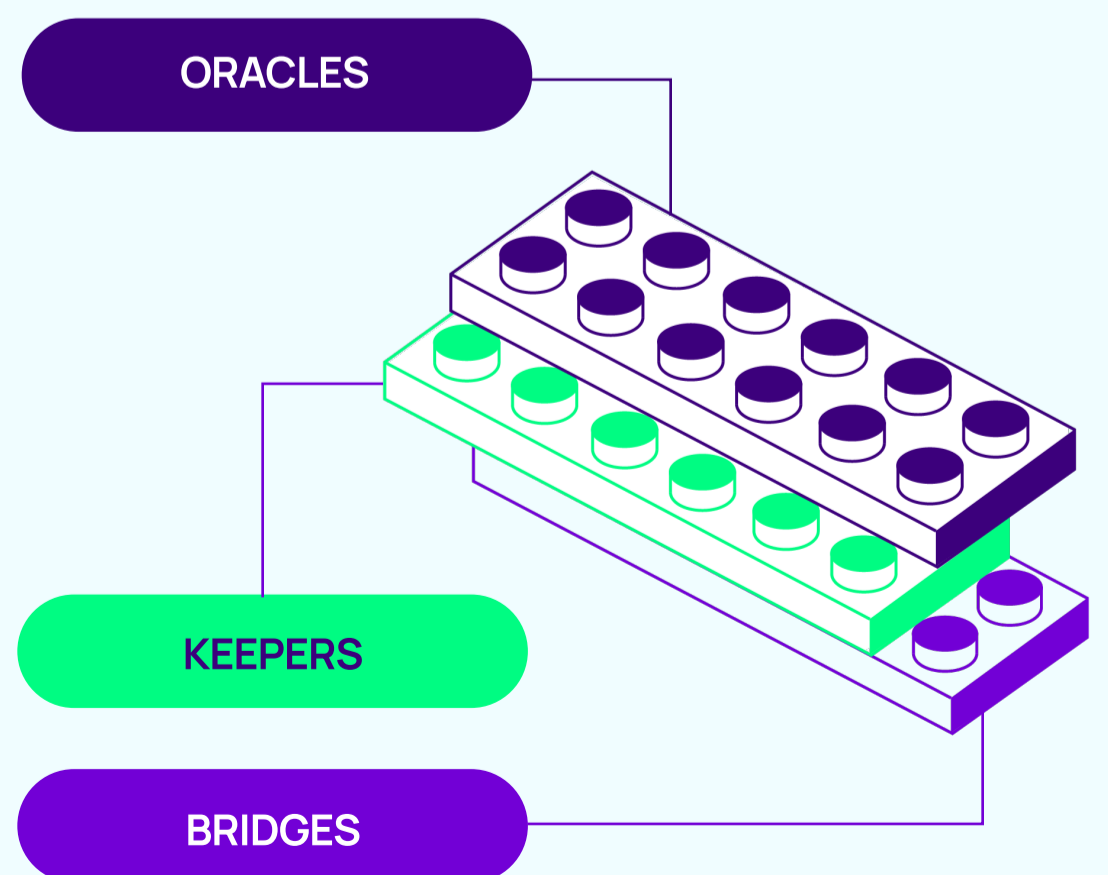
Governance

A crucial element of the success of Autonolas is to have an active community and ecosystem that both build, evolve, promote and make use of Autonolas technology. For this reason, Autonolas is organized as a DAO where meaningful contributors and supporters will participate in the decision-making process.

Initially, holders of the virtualized veOLAS token can participate in any governance activities. The veOLAS token is obtained by locking OLAS, which is the native token of Autonolas. Governance participation is proportional to veOLAS holdings and their locking duration.

Governance proposals can notably modify system parameters, support new technological directions, or add entirely new functionality to the on-chain protocol. Once a governance proposal is approved, the Timelock adds a delay for the proposal to be executed.
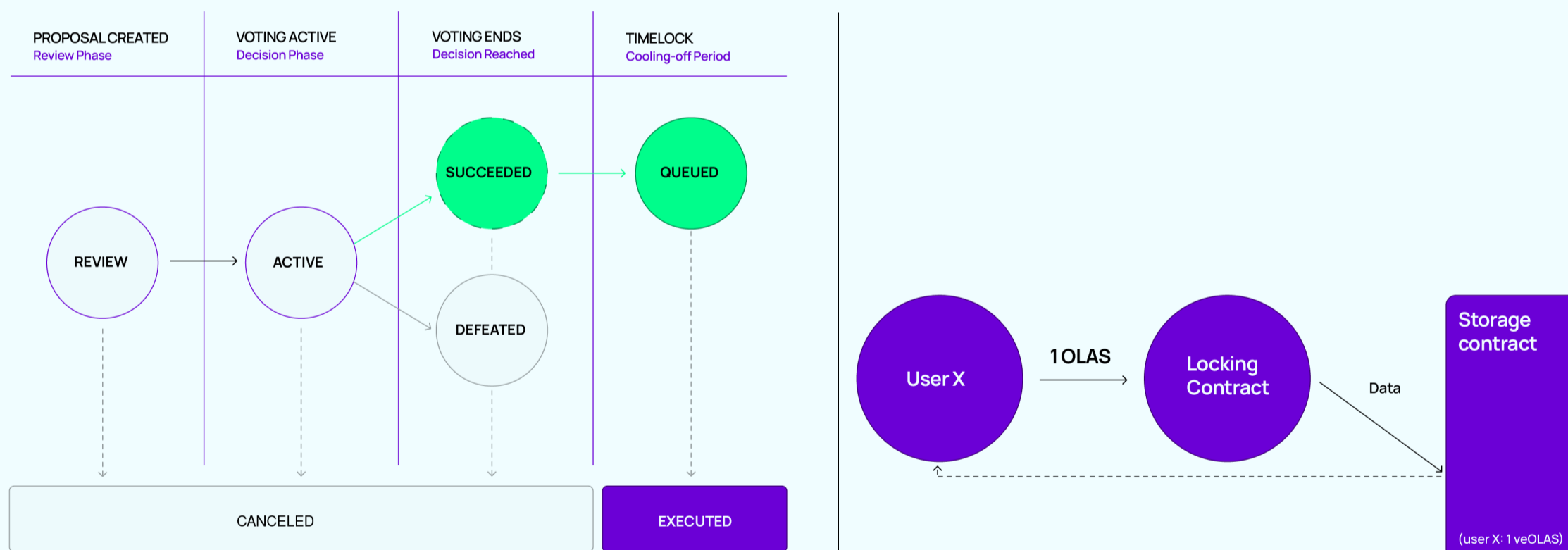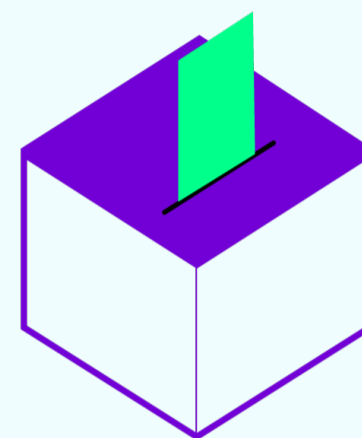
Exceptionally, some changes to the Autonolas on-chain protocol could be executed by a community-owned multi-sig wallet, bypassing the governance process. This allows a set of trusted actors to overrule governance in certain aspects, e.g. a security exploit that needs to be patched without governance discussion.

Follow Autonolas

twitter.com/autonolas

autonolas .network

discord.com/invite/z2PT65jKqQ