ELYRA

AUTONOMOUS INTELLIGENCE NETWORK

NETWORK INITIALIZED

An AI-native execution network designed to extend decentralized infrastructure toward autonomous digital intelligence.

DOCUMENT	VERSION	NETWORK	PHASE
WHITEPAPER	**0.1 — GENESIS**	**EVM COMPATIBLE**	**GENESIS TESTNET**

Elyra is an experimental protocol under active development. Architecture, functionality, token utility, network design and roadmap elements described in this document may evolve as research and development progress. This document does not constitute financial advice or an offer to sell securities.

Table of Contents

A structured guide to the Elyra protocol and its evolutionary architecture.

01 Overview 3

02 The Infrastructure Gap 5

03 Autonomous Intelligence Paradigm 7

04 Network Architecture 9

05 Agent Runtime 11

06 Autonomous Execution Engine 14

07 Agent Identity & Programmable Wallets 17

08 Memory & Intelligence Layer 19

09 Security Model 22

10 Developer Ecosystem 24

11 Token Economy 27

12 Governance 32

13 Roadmap 34

14 Security Model — Extended 36

15 Conclusion 38

A Appendix D — Technical Documentation 40

CHAPTER 01

Overview

The Foundation for Autonomous Intelligence

The evolution of digital systems has always been defined by the emergence of new participants. Personal computers introduced individual computing. The internet connected people and information globally. Mobile devices created continuous digital interaction. Blockchain introduced decentralized ownership and programmable value.

Artificial Intelligence introduces the next transformation: **autonomous digital intelligence.** AI systems are evolving beyond traditional software applications. They are increasingly capable of analyzing information, making decisions, executing workflows and interacting with digital environments with limited human intervention.

INFRASTRUCTURE CHALLENGE

Existing digital networks were designed for users and applications. Future networks must support autonomous entities. Elyra is designed to address this transition.

What is Elyra?

Elyra is an **AI-native execution network** designed to provide the infrastructure foundation for autonomous digital intelligence. The protocol begins with an EVM-compatible blockchain environment, enabling developers to deploy applications using familiar tools and established standards.

Over time, Elyra will introduce additional protocol-native capabilities designed specifically for autonomous systems, including:

- Agent Identity
- Programmable Agent Wallets
- Agent Runtime
- Autonomous Execution Engine
- Execution Memory
- Intelligence Coordination Layers

The long-term objective is to create an environment where humans, applications and autonomous agents can operate within a shared decentralized infrastructure.

The Convergence of Blockchain and AI

Blockchain and artificial intelligence solve two different fundamental problems.

BLOCKCHAIN PROVIDES	ARTIFICIAL INTELLIGENCE PROVIDES
Decentralized ownership	Reasoning capabilities
Transparent execution	Decision making
Trustless settlement	Adaptation
Programmable assets	Automation

Current blockchain architectures were not designed for autonomous intelligence — they were optimized around human-controlled wallets, explicit transactions and static smart contracts. Elyra introduces an evolutionary approach: **extending blockchain infrastructure toward autonomous digital participation.**

Core Principles

Autonomous Execution

Autonomous systems should be capable of performing operations according to predefined objectives, permissions and constraints, enabling independent execution, transparent operations and verifiable outcomes.

Programmable Intelligence

Intelligence should become a reusable infrastructure primitive. Developers should be able to build systems where autonomous behavior can interact with standardized protocol components.

Verifiable Trust

Autonomous systems require accountability. Every operation must remain observable, auditable and cryptographically verifiable. AI should extend blockchain guarantees, not replace them.

DESIGN PHILOSOPHY

Elyra does not aim to replace blockchain infrastructure. Elyra aims to evolve blockchain infrastructure for an autonomous future.

CHAPTER 02

The Infrastructure Gap

Blockchains Were Designed Around Human Interaction

Modern blockchain networks represent a major technological achievement. They introduced decentralized consensus, programmable contracts, digital ownership and open financial systems. However, their fundamental interaction model remains unchanged.



A user creates an action. A wallet authorizes it. A contract executes predefined logic. The process ends. This model works extremely well for traditional decentralized applications — but autonomous intelligence introduces a different requirement.

From Applications to Participants

Artificial intelligence systems are becoming increasingly autonomous. Modern AI systems can analyze complex information, maintain context, select strategies, execute multi-step processes and interact with external systems — evolving from tools into participants. A future autonomous agent may manage digital assets, coordinate services, optimize resources, interact with protocols and execute economic decisions without requiring constant human intervention.

Static Logic vs Dynamic Intelligence

Smart Contracts

Smart contracts are deterministic systems. Given the same input, they produce the same output — providing security, predictability and verification.

Artificial Intelligence

AI systems operate differently. They evaluate context, objectives, available information and potential outcomes. They are designed to adapt.

The future requires infrastructure capable of combining blockchain verification, autonomous decision-making, and programmable execution.

The Infrastructure Gap

CAPABILITY	CURRENT LIMITATION
Agent Identity	External to blockchain infrastructure
Autonomous Execution	Requires external automation systems
Memory	Stored outside protocol boundaries
Permissions	Managed individually by applications
Coordination	Requires custom implementations

This forces developers to assemble complex architectures from multiple independent components, resulting in increased complexity, additional security assumptions and a fragmented user experience.

Toward an AI-Native Network

The next generation of blockchain infrastructure must support autonomous entities as native participants. This requires persistent identities, programmable ownership, autonomous execution environments and transparent interaction models. This architectural direction defines Elyra.

CHAPTER 03

Autonomous Intelligence Paradigm

Intelligence as a Native Network Participant

Blockchain networks have historically focused on assets, transactions and contracts. Elyra introduces a different perspective: **intelligence itself becomes a participant.**

A New Digital Actor

Traditional software executes instructions. Autonomous intelligence evaluates objectives. This distinction changes the architecture of digital systems. An autonomous agent can observe, reason, decide, execute and coordinate. The network must therefore evolve from transaction processing toward autonomous activity.

The Intelligence Network Concept

Elyra introduces the concept of an **Intelligence Network** — a decentralized environment where humans interact with applications, applications interact with protocols, and autonomous agents interact with the ecosystem. All participants operate under shared blockchain guarantees.

From Transactions to Autonomous Systems

Traditional model

User → Application → Blockchain

Future model

Human → Application → Autonomous Agent → Blockchain

The blockchain becomes the settlement layer for autonomous digital activity.

Protocol Design Principles

1. Autonomous Execution

Agents must eventually operate independently while maintaining permission boundaries, transparent execution and verifiable outcomes.

2. Programmable Intelligence

Developers should be able to create reusable autonomous behaviors through standardized components.

3. Verifiable Trust

Every autonomous operation must remain attributable and auditable.

Evolutionary Development

Elyra follows a progressive development approach. The initial network establishes EVM compatibility, developer infrastructure and blockchain execution. Future upgrades introduce agent primitives, autonomous runtime and intelligence coordination. The protocol evolves step by step.

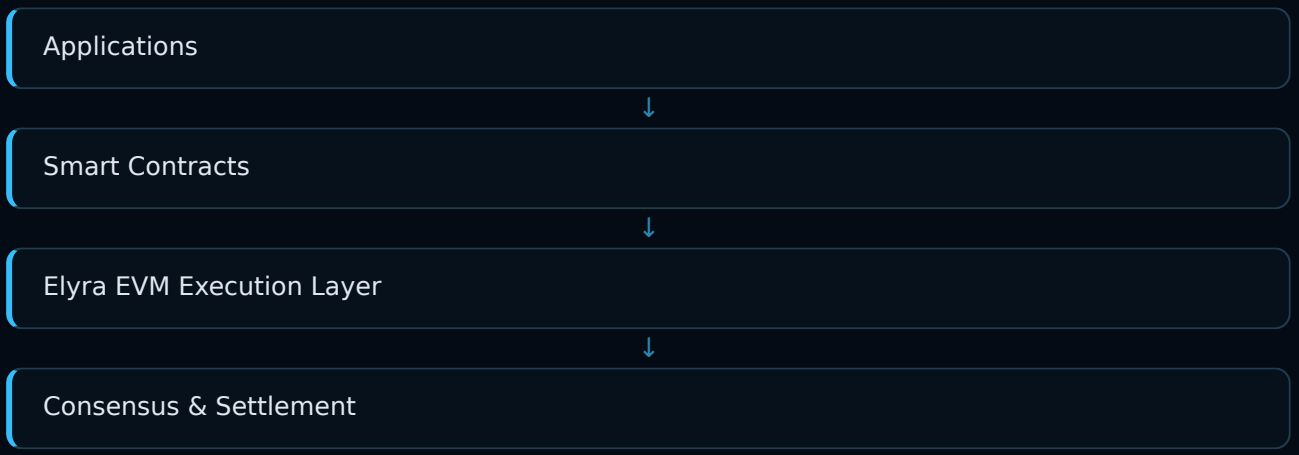
CHAPTER 04

Network Architecture

The Foundation Layer

The first phase of Elyra focuses on establishing a reliable blockchain execution environment. This foundation enables developers to build decentralized applications while preparing the infrastructure required for future AI-native capabilities.

Current Architecture



EVM Compatibility

Elyra adopts EVM compatibility to maximize accessibility. Developers benefit from existing tooling, Solidity compatibility, familiar wallets and established developer workflows — allowing ecosystem growth while future protocol capabilities are developed.

Core Network Components

COMPONENT	PURPOSE
EVM Execution Layer	Smart contract execution
Consensus Layer	Network security and finality
RPC Infrastructure	Blockchain communication
Explorer	Transparency and verification
Wallet Compatibility	User interaction
Developer Tools	Application creation

Architecture Evolution

The initial network provides the foundation for future AI-native modules.



Open Development

Elyra follows an open development philosophy. The ecosystem will progressively introduce developer documentation, open-source tooling, protocol research and community contributions.

> CHAPTER SUMMARY
>
> - Elyra begins as an EVM-compatible execution network.
> - The initial architecture provides a secure foundation for decentralized applications.
> - Future upgrades introduce AI-native protocol components.
> - The long-term vision is a network where autonomous intelligence becomes a native participant.

CHAPTER 05

Agent Runtime

The Execution Environment for Autonomous Intelligence

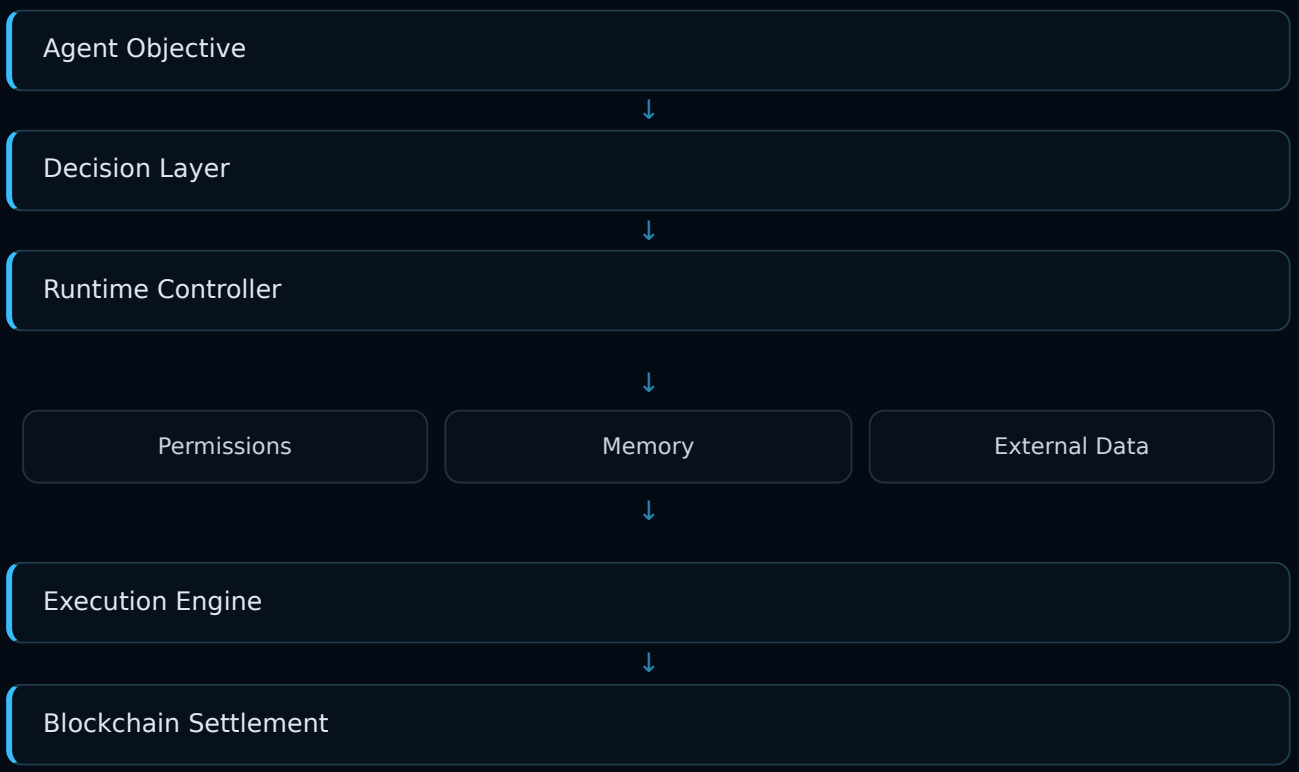
Traditional blockchain applications execute transactions initiated by users. Autonomous systems require a different execution model. An AI agent is not a single transaction — it is a continuous process involving objectives, context, decision-making, execution and feedback.

The Agent Runtime represents the future execution environment designed to coordinate autonomous entities within the Elyra ecosystem. The initial Elyra network establishes the blockchain foundation required for this evolution.

ARCHITECTURE INSIGHT

The Agent Runtime separates intelligence execution from blockchain settlement. AI systems determine actions. The blockchain verifies and records outcomes.

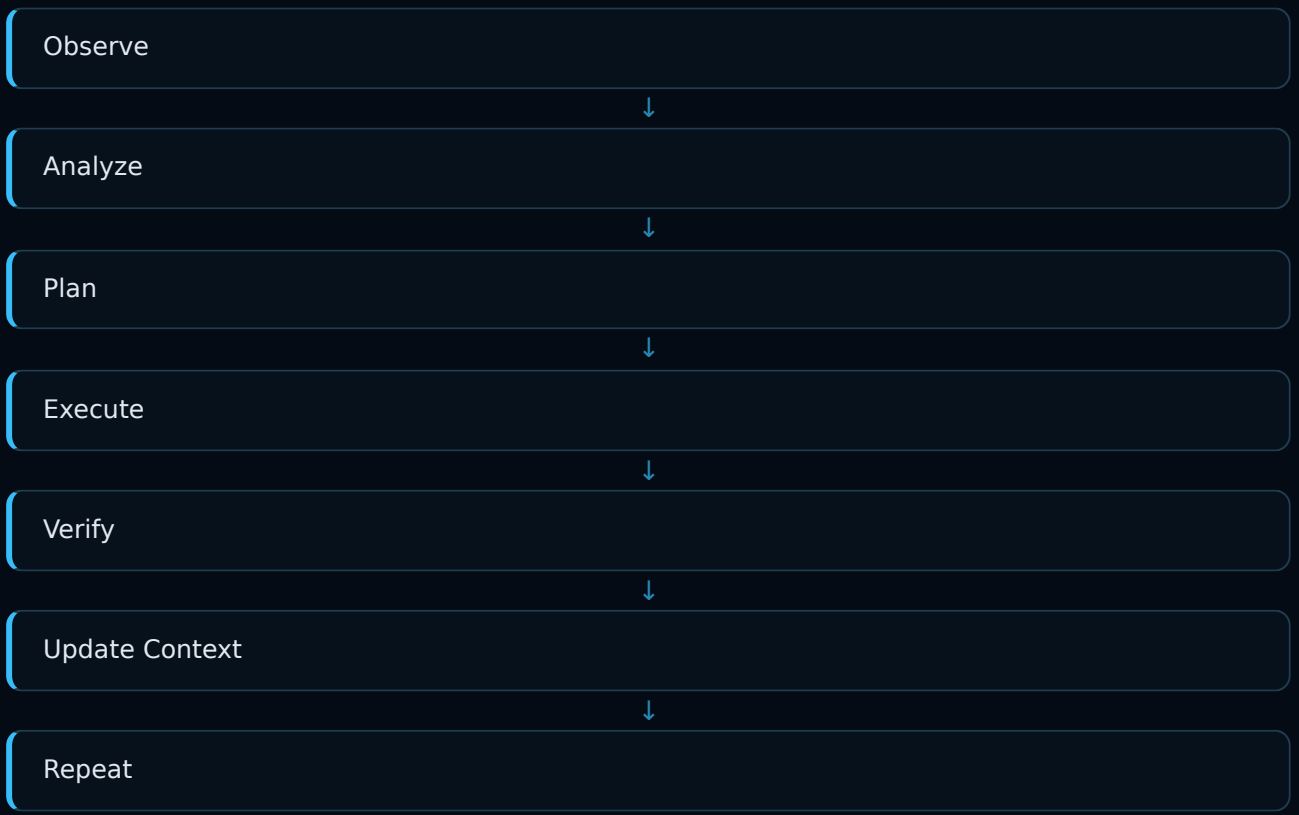
Runtime Architecture



Agent Lifecycle Model

STATE	DESCRIPTION
Created	Agent identity initialization
Configured	Permissions and objectives assigned
Active	Agent can execute operations
Suspended	Execution temporarily disabled
Archived	Historical state preserved

Agent Execution Cycle



Execution Model

A simplified representation of an agent's action:

$$A(t) = f(O(t), C(t), P(t))$$

SYMBOL	MEANING
A(t)	Agent action
O(t)	Current objective
C(t)	Context state
P(t)	Permission boundaries

Runtime Security Boundaries

Autonomous execution requires strict limitations. Every agent operation must respect:

$$Execution \subseteq Permissions$$

Meaning an agent can only execute actions allowed by its assigned permission model. Example:

PARAMETER	VALUE
Maximum Transfer	10 ELY
Allowed Contracts	DEX, Treasury
Blocked	Unknown Addresses

Testnet Implementation Path

During the initial Elyra testnet phase:

✓ Available: EVM execution · Smart contracts · Wallet interaction · Transaction settlement

○ Future: Agent Runtime modules · Agent permissions · Autonomous execution framework

CHAPTER 06

Autonomous Execution Engine

Coordinating Intelligent Operations

The Autonomous Execution Engine represents the coordination layer between intelligent decision systems and blockchain execution. Its purpose is not to replace consensus. Its purpose is to organize autonomous operations before final settlement.

PROTOCOL SEPARATION

Intelligence decides. Execution coordinates. Consensus verifies.

Execution Pipeline



Execution Verification Model

Every autonomous operation can be represented as:

```
Transaction = Intent + Authorization + Execution + Proof
```

COMPONENT	PURPOSE
Intent	Desired operation
Authorization	Permission verification
Execution	Blockchain action
Proof	Final verification

Execution Prioritization

Future versions may implement execution scheduling:

```
Priority = (Value × Confidence) / Cost
```

VARIABLE	MEANING
Value	Expected benefit
Confidence	Execution reliability
Cost	Required resources

Illustrative Example

Agent receives objective: *Optimize treasury allocation.* The engine evaluates candidate actions:

METRIC	ACTION A	ACTION B
Expected Value	100	120
Cost	10	40
Confidence	0.9	0.6

The system selects the action according to predefined policies.

Deterministic Settlement

AI decisions may be probabilistic. Blockchain execution remains deterministic.

AI Layer (decision)

↓

Execution Layer (validation)

↓

Blockchain (final state)

This separation maintains network security.

CHAPTER 07

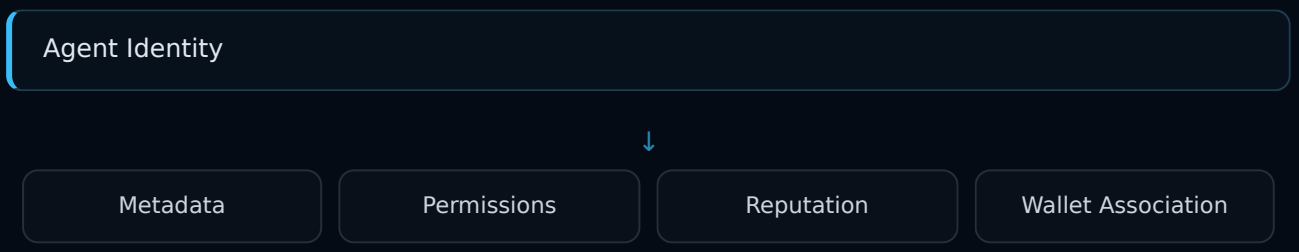
Agent Identity & Programmable Wallets

Digital Ownership for Autonomous Entities

Current blockchain wallets represent ownership controlled by humans. Future autonomous systems require a different identity model. An autonomous agent requires persistent identity, ownership capabilities, permission control and transaction history.

Agent Identity Model

Future Elyra architecture introduces protocol-native identities:



Identity Properties

PROPERTY	FUNCTION
Persistence	Same identity across operations
Verification	Cryptographic ownership
Reputation	Historical behavior
Permissions	Controlled execution

Programmable Agent Wallets

An Agent Wallet differs from a traditional wallet.

Traditional wallet



Agent Wallet



Permission Framework

Spending Limits

```
Daily Limit = 1,000 ELY
```

Contract Restrictions

Allowed · DEX | Allowed · Treasury | Allowed · Staking | Blocked · Unknown Contracts

Time Constraints

```
Execution Window: 00:00 – 06:00 UTC
```

Security Philosophy

Autonomous ownership must never mean unrestricted ownership.

```
Autonomy + Constraints = Secure Intelligence
```

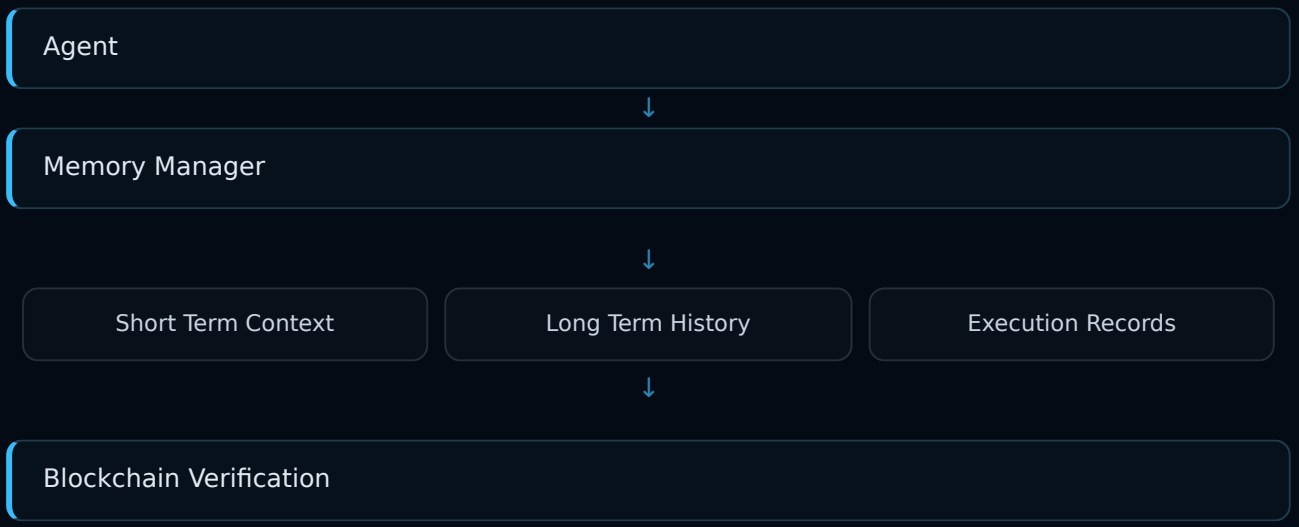
CHAPTER 08

Memory & Intelligence Layer

Persistent Context for Autonomous Systems

Human decision-making depends on memory. Autonomous systems require an equivalent capability. Memory allows agents to maintain historical context, previous actions, operational state and learned information.

Memory Architecture



Memory Types

TYPE	PURPOSE
Operational Memory	Current execution context
Historical Memory	Previous interactions
Reputation Memory	Behavioral history
Protocol Memory	Network-level information

Memory Integrity Model

Memory updates require verification. Simplified:

```
Memory(new) = Hash( Memory(old) + Event )
```

Each update creates a verifiable chain of state changes.

Illustrative Example

ASSET	BEFORE	AFTER
Asset A	40%	50%
Asset B	60%	50%

The transition is recorded as: *Previous State + Verified Event = New State.*

Future Intelligence Stack

Applications

↓

Autonomous Agents

↓

Agent Runtime

↓

Execution Engine

↓

Identity + Wallet

↓

Memory Layer

↓

EVM Network

↓

Consensus

CHAPTER SUMMARY

- Agent Runtime provides the future execution environment for autonomous entities.
- Autonomous Execution Engine coordinates operations before settlement.
- Agent Identity enables persistent digital ownership.
- Programmable Wallets introduce controlled autonomous asset management.
- Memory infrastructure allows agents to maintain verifiable context.

CHAPTER 09

Security Model

Designing Trust for Autonomous Systems

Security represents the foundation of every decentralized network. As blockchain systems evolve toward autonomous intelligence, security requirements expand beyond traditional transaction validation. Future autonomous systems require protection across network security, execution security, identity security, permission security and economic security.

SECURITY PRINCIPLE

Autonomous intelligence should never bypass blockchain security. Intelligence creates decisions. Cryptographic systems validate execution.

Layered Security Architecture



Consensus Security

The foundation of Elyra remains blockchain consensus, providing transaction ordering, state finality, network synchronization and attack resistance.

Execution Security

```
State(new) = F( State(old), Transaction )
```

Autonomous Execution Security

Future autonomous agents require additional controls. The fundamental principle:

$$\text{Allowed Actions} \subseteq \text{Defined Permissions}$$

ALLOWED	DENIED
Swap Assets	Transfer Unlimited Funds
Manage Treasury	Modify Identity
Interact with Approved Contracts	Disable Security Rules

Identity Security

Future agent identities will rely on cryptographic ownership, permission management and historical verification — ensuring authenticity, accountability and traceability.

Economic Security

Network security also depends on sustainable incentives, including token utility, validator incentives, network participation and attack resistance.

> CHAPTER SUMMARY
>
> - Elyra maintains blockchain security foundations.
> - Autonomous systems operate within defined boundaries.
> - Future AI capabilities are introduced without compromising verification.
> - Security evolves together with protocol capabilities.

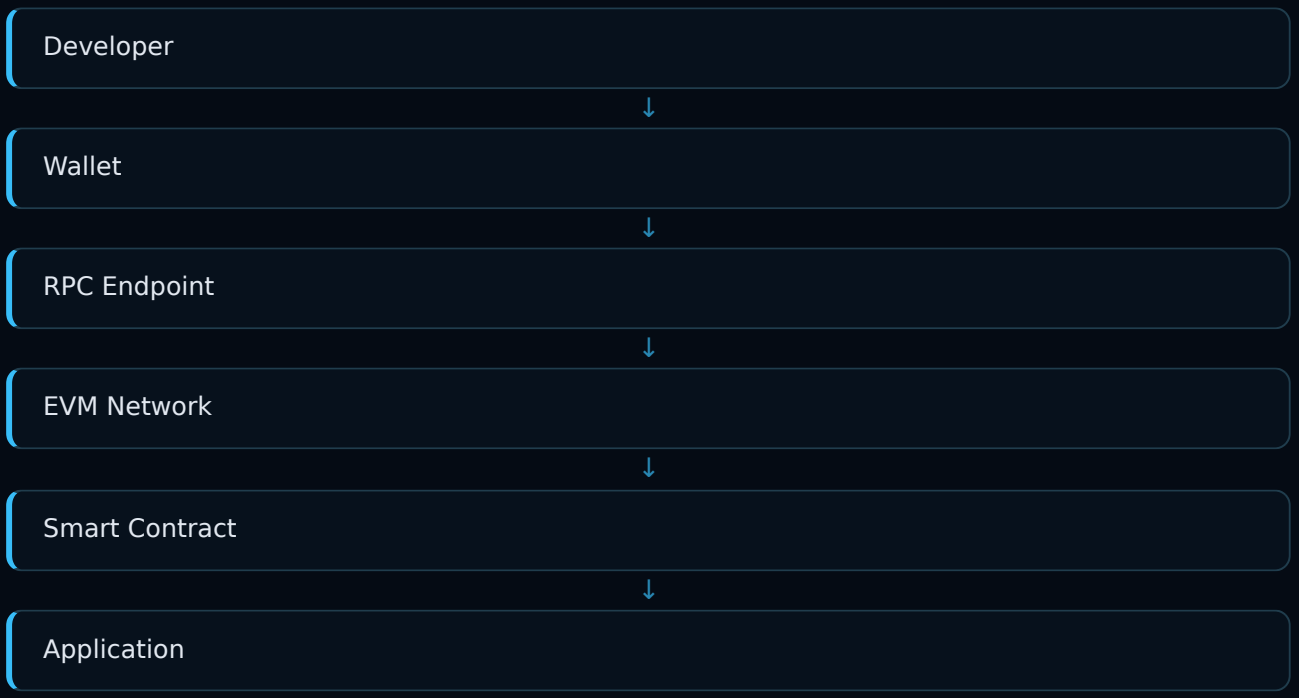
CHAPTER 10

Developer Ecosystem

Building the Infrastructure for Open Innovation

The growth of any protocol depends on developers. Elyra adopts an open and accessible approach centered on familiar tools, transparent development, EVM compatibility and progressive protocol evolution.

Initial Developer Environment



Testnet Developer Stack

The Genesis Testnet is designed around standard blockchain workflows. Developers can connect wallets, deploy contracts, test applications, explore transactions and build integrations.

Network Components

COMPONENT	PURPOSE
RPC Layer	Blockchain communication
Explorer	Transaction transparency
Wallet Integration	User access
Smart Contract Layer	Application logic
Developer Documentation	Technical onboarding
Open Repository	Public development

GitHub Architecture

```
elyra-network/
  ├── contracts
  ├── documentation
  ├── examples
  ├── network-config
  ├── developer-tools
  └── research
```

Smart Contract Development

Because Elyra is EVM-compatible, developers can use established tools:

Solidity · Hardhat · Foundry · Web3 libraries · Existing wallets

Future Developer Modules

TRACK	INCLUDES
Current	EVM Applications
Future	EVM Applications + Agent Frameworks + Autonomous Modules + AI-native Components

NOTE

As soon as the Genesis Testnet opens, developers can build with familiar blockchain infrastructure while Elyra progressively introduces autonomous intelligence primitives.

Developer Roadmap

Genesis Network

✓ Testnet · EVM compatibility · Developer tools · Agent registry and bounded agent wallets (smart contracts)

Ecosystem Expansion

○ Grants · Hackathons · Independent validator operators

AI-Native Infrastructure

○ Protocol-native Agent Identity · Agent Wallets · Runtime modules

Autonomous Economy

○ Mainnet · AI-native applications · Agent coordination · Machine-driven economies

CHAPTER SUMMARY

- Elyra prioritizes developer accessibility.
- The initial network supports standard EVM workflows.
- Future upgrades expand the developer stack toward autonomous intelligence.

CHAPTER 11

Token Economy

The Economic Layer of the Elyra Network

The Elyra token represents the economic coordination layer of the Elyra ecosystem. As a Layer 1 protocol, Elyra requires a sustainable economic model capable of supporting network security, ecosystem growth, developer adoption, community participation and long-term decentralization.

The Elyra token distribution follows a **public, fixed-price model**, designed to maximize transparency and minimize centralized ownership concentration. Elyra does not include private token sales, venture capital allocations or preferential early-investor pricing: a single public presale offers up to 15% of supply at one fixed price, with the same terms for everyone.

Token Specification

PARAMETER	VALUE
Token Name	Elyra
Symbol	ELY
Initial Network	Ethereum (ERC-20)
Native Network	Elyra Mainnet (post-migration)
Distribution Model	Public fixed-price presale · no private sale
Presale Price	$0.001333 per ELY (fixed)
Launch Price	$0.001467 per ELY (+10% on presale)
Initial Supply	1,000,000,000 ELY (fixed)
Token Type	ERC-20 at launch → Native Network Asset at Mainnet

Token Launch & Migration Path

ELY does not launch directly as the native asset of the Elyra chain. The token follows a deliberate, staged path designed to build distribution and liquidity early, while reserving native-asset status for the point at which the Elyra Mainnet can actually secure and settle it.

Public Presale & Ethereum Launch (ERC-20)

A public presale sells up to 15% of supply at a fixed $0.001333; unsold tokens are burned. ELY then launches as an ERC-20 token on Ethereum: 20% of the proceeds are paired with ELY on Uniswap and the LP is locked 12 months. The token and vesting contracts and an independent third-party audit are published days before launch.

↓

Genesis & Testnet Phase

The ERC-20 token is used for early community, ecosystem and liquidity participation while the Elyra L1 is built and tested.

↓

Mainnet Activation

The Elyra Mainnet goes live as an independent, EVM-compatible Layer 1 with its own native asset.

↓

Bridge Opens — 1:1 Migration

Holders migrate ERC-20 ELY to native Mainnet ELY through the official bridge. Ethereum-side tokens are locked/burned as native tokens are released 1:1. Validator Rewards begin to be emitted from this point — see Supply Distribution below.

↓

Native ELY

Native ELY becomes the chain's core asset: gas, staking/validator incentives, and governance.

MIGRATION PRINCIPLE

The Ethereum-to-Mainnet migration is strictly **1:1**. No dilution, no re-allocation, no additional minting at the point of migration — every wallet's proportional ownership of total supply is preserved exactly as it was on Ethereum.

Once the bridge is live, the Ethereum ERC-20 contract is retained only as the legacy migration entry point. All new protocol-level utility — transaction fees, validator and staking rewards, and governance voting weight — accrues exclusively to the native Mainnet token. Holders who do not migrate retain their ERC-20 balance and can bridge at any later point; migration is not time-limited unless stated otherwise in official bridge documentation at launch. Tokens still vesting on Ethereum migrate with their remaining schedule unchanged.

Supply Distribution Overview

ALLOCATION	%	LOCK / RELEASE
Public Presale	15%	Fully unlocked at launch, no vesting; unsold tokens burned
Liquidity & Market Making	15%	Up to 3% paired with 20% of proceeds on Uniswap at launch (LP locked 12 months); the rest added over time to DEX and exchange liquidity; multisig
Genesis Airdrop · Season 1	4%	35% at launch, rest daily over the following 11 months
Creators & KOL (Elyra Voices)	4%	35% at launch, rest daily over the following 11 months
Ecosystem & Grants	22%	Locked 6 months, then 24-month linear release; multisig
Validator Rewards	20%	Emitted only after Mainnet, over several years
Treasury & Operations	10%	Locked 6 months, then 24-month linear release; multisig
Team & Core Contributors	10%	Locked 12 months, then 24-month linear vesting

Why These Schedules

Presale buyers receive their full allocation at launch; every allocation held by the project is locked or released over time.

AT LAUNCH — PRESALE, LIQUIDITY, COMMUNITY

About 21% of supply circulates at launch: the whole presale, which has no vesting, the Uniswap pool, and 35% of the community and creator rewards. Everyone who receives tokens without paying gets the rest over the following 11 months, and the rest of the Liquidity & Market Making allocation deepens markets over time.

LONG LOCKS — TEAM, ECOSYSTEM, TREASURY

Team tokens are locked for 12 months and then vest over 24 months, so the team is the last group able to sell. Ecosystem and Treasury allocations are locked for 6 months, then released over 24 months from multisig wallets; every use is disclosed publicly.

Use of Presale Proceeds

USE	SHARE OF PROCEEDS
Locked Uniswap liquidity (LP locked 12 months)	20%
Protocol development — testnet, AI-native features, mainnet	45%
Security audits & bug bounty	15%
Operations & legal	8%
Marketing, community & listings	12%

Launch Principles

Transparency

All allocations, percentages, unlock schedules and the use of presale proceeds are publicly documented above. The token and vesting contracts and an independent third-party audit are published together, days before launch. Proceeds are held in a multisig wallet and spending is reported publicly every quarter.

Accessibility

One public presale, one fixed price and the same per-wallet limits for everyone. Elyra does not include private sale rounds, venture allocations or preferential early-investor pricing.

Schedule-Based Unlocks

Every allocation held by the project is released through public, on-chain vesting schedules rather than at anyone's discretion. Validator Rewards are the only allocation tied to a network milestone: they are emitted only once the Mainnet exists.

Token Utility Evolution

Initial utility, on Ethereum, is limited to holding and trading the ERC-20 token. Once native Mainnet ELY is live post-migration, utility expands to network participation, transaction fees, staking and validator incentives, governance mechanisms, ecosystem incentives, protocol services and autonomous economy interactions.

NOTE

Elyra follows a progressive economic model. The initial phase focuses on building a secure and accessible Layer 1 foundation. Future protocol upgrades will introduce additional utilities as autonomous intelligence capabilities mature.

CHAPTER SUMMARY

- ELY launches as an ERC-20 token on Ethereum after a public presale at a fixed $0.001333, then migrates 1:1 to native Mainnet ELY once the Elyra L1 goes live.
- Native ELY becomes the chain's core asset: gas, staking/validator incentives and governance.
- 15% is offered in the presale, fully unlocked at launch with no vesting. Unsold tokens are burned.
- 15% funds Liquidity & Market Making: 20% of presale proceeds is paired with up to 3% of supply on Uniswap (LP locked 12 months, launch price 10% above the presale price); the rest deepens DEX and exchange liquidity over time, under multisig.
- 22% funds Ecosystem & Grants and 10% Treasury & Operations: locked 6 months, then released over 24 months under multisig control.
- 20% funds Validator Rewards, emitted only after Mainnet.
- 4% rewards the Genesis community and 4% the Elyra Voices creators, each shared in proportion to credits or points: 35% at launch, the rest daily over the following 11 months.
- 10% is allocated to the Team, locked 12 months and then vested over 24 months.
- About 21% of supply circulates at launch. The token and vesting contracts and an independent third-party audit are published together, days before launch.

CHAPTER 12

Governance

Coordinating the Evolution of Autonomous Infrastructure

A decentralized protocol requires more than technical infrastructure. It requires a framework that enables participants to coordinate, propose improvements and guide the evolution of the network.

NOTE

Governance is considered a progressive process. Decentralization is not introduced instantly — it develops alongside network maturity, security and ecosystem participation.

Progressive Decentralization

Phase 1: Protocol Foundation

↓

Phase 2: Community Coordination

↓

Phase 3: Decentralized Governance

Phase 1 — Protocol Foundation

Core contributors focus on network development, security improvements, infrastructure deployment, technical upgrades and developer experience.

Phase 2 — Community Coordination

Governance expands through community discussions, improvement proposals, ecosystem initiatives, developer feedback and research contributions.

Phase 3 — Decentralized Governance

Future governance mechanisms may introduce on-chain proposals, voting systems, treasury coordination, protocol parameter management and upgrade coordination.

Governance Framework



Governance Principles

Transparency

All governance decisions should be publicly verifiable.

Participation

Network participants should have mechanisms to contribute to protocol evolution.

Security

Governance processes must prioritize protocol stability and prevent malicious changes.

Adaptability

The governance model must evolve alongside technological development.

CHAPTER 13

Roadmap

Building the Infrastructure for Autonomous Intelligence

The Elyra roadmap follows a progressive development strategy. Rather than attempting to introduce every capability immediately, the protocol evolves through structured phases, each establishing the foundation required for the next stage of development.

Phase 1 — Genesis Network

Foundation Layer · Current Development Phase

Objectives: launch EVM-compatible testnet, deploy network infrastructure, enable wallet connectivity, provide RPC accessibility, establish explorer and developer tooling, publish technical documentation.

COMPONENT	STATUS
EVM Execution	Initial
Smart Contracts	Enabled
Wallet Compatibility	Enabled
RPC Infrastructure	Enabled
Developer Environment	Active

Phase 2 — Ecosystem Expansion

Developer Growth Layer

Objectives: developer onboarding, ecosystem applications, infrastructure growth, community expansion — through developer programs, hackathons, grants, technical resources and ecosystem partnerships.

Phase 3 — AI-Native Infrastructure

Autonomous Execution Layer

Introduces the foundational components required for autonomous intelligence: Agent Identity, Agent Wallets, Agent Runtime, Autonomous Execution Engine, Execution Memory — enabling intelligent entities to interact with blockchain infrastructure through native protocol components.

Phase 4 — Autonomous Economy

The Intelligence Network

This phase includes Mainnet activation: the Elyra Mainnet goes live as an independent Layer 1, the bridge opens, and ERC-20 ELY on Ethereum migrates 1:1 to native Mainnet ELY, which becomes the chain's gas, staking and governance asset. Potential capabilities: agent-to-agent communication, autonomous applications, intelligent coordination systems, machine-driven economic activity.

Roadmap Philosophy

Foundation → Ecosystem → Intelligence → Autonomy

Each stage builds upon previous infrastructure rather than replacing it.

NOTE

Elyra adopts an evolutionary architecture. Every protocol upgrade is designed to extend network capabilities while maintaining compatibility and reliability.

CHAPTER 14

Security Model — Extended

Designing Trust for Autonomous Systems

As Elyra evolves toward autonomous intelligence, security requirements expand beyond traditional blockchain assumptions. The protocol must ensure that autonomous entities can operate while maintaining transparency, verification, accountability and deterministic settlement.

Blockchain Security Foundation

The initial Elyra architecture inherits established blockchain security principles: cryptographic verification, distributed consensus, deterministic execution and transparent state transitions.

Autonomous Execution Security

REQUIREMENT	PURPOSE
Identity	Establish agent ownership and attribution
Permissions	Control available actions
Verification	Validate operations
Isolation	Prevent unauthorized execution
Auditability	Maintain transparency

Agent Permission Model



Deterministic Settlement

CORE SEPARATION

Intelligence decides what should happen. Consensus verifies what actually happened.

This separation allows Elyra to combine adaptive intelligence with deterministic blockchain guarantees.

Security Evolution

Future improvements may include advanced execution controls, agent verification mechanisms, runtime isolation, monitoring systems and formal validation frameworks.

CHAPTER 15

Conclusion

The Foundation for Autonomous Digital Intelligence

The next evolution of decentralized infrastructure will not only involve transferring ownership from centralized systems. It will involve enabling new categories of digital participants. Blockchain introduced decentralized ownership. Smart contracts introduced programmable execution. Artificial intelligence introduces autonomous intelligence. Elyra is designed at the intersection of these transformations.

From Applications to Participants

Traditional blockchain systems were built around human interaction. Users create transactions, contracts execute predefined logic, applications provide services. The emergence of autonomous intelligence requires a new architectural model — a future where intelligent entities can own identities, autonomous systems can manage resources, digital agents can interact securely, and decentralized economies can operate continuously.

The Elyra Vision

Elyra aims to provide the execution foundation for this transition. The protocol combines blockchain security, programmable infrastructure, autonomous execution capabilities and future AI-native components. The objective is not to replace existing blockchain systems — it is to extend them toward a future where intelligence becomes a native network participant.

Long-Term Perspective

FROM	TO
Human	Computer
Computer	Internet
Human	Blockchain
Human + Machine	Autonomous Networks

VISION STATEMENT

Elyra is building the infrastructure where autonomous intelligence can securely participate in decentralized economies.

Appendix A — Initial Network Principles

Genesis Testnet Phase — core objectives: network accessibility, developer experimentation, smart contract deployment, infrastructure testing, community participation. The Genesis Testnet represents the first technical milestone toward the long-term AI-native execution architecture.

Appendix B — Development Philosophy

PRINCIPLE	DESCRIPTION
Compatibility	Maintain familiar blockchain development standards
Transparency	Public technical evolution
Modularity	Independent protocol components
Scalability	Architecture designed for future expansion
Innovation	Progressive introduction of AI-native capabilities

Appendix C — Final Statement

The future of decentralized systems will not only be defined by who owns digital assets. It will be defined by what digital entities are capable of doing. Elyra is designed to provide the infrastructure for that future.

APPENDIX D

Technical Documentation

Elyra Genesis Testnet — Developer Documentation v0.1

The Elyra Genesis Testnet represents the first technical milestone toward the Elyra AI-native execution network. The initial network phase focuses on establishing a stable and accessible EVM-compatible environment where developers can deploy smart contracts, test decentralized applications, explore protocol capabilities, experiment with ecosystem primitives, and prepare applications for future network upgrades.

NOTE

The Genesis Testnet represents the initial execution layer of Elyra. Future protocol upgrades will progressively introduce AI-native capabilities including autonomous agents, programmable identities and intelligent execution modules.

Network Overview

PARAMETER	VALUE
Network Name	Elyra Genesis Testnet
Network Type	EVM-Compatible Blockchain
Consensus Model	QBFT (Byzantine fault tolerant), immediate finality
Validators	Operated by the Elyra team during Genesis; independent operators join in Ecosystem Expansion
Virtual Machine	Ethereum Virtual Machine
Gas Coin	ELY (testnet, no monetary value)
Environment	Testnet

Note: on the Genesis Testnet, ELY is a test gas coin with no monetary value, distributed by the faucet. It is not the ERC-20 token, which launches separately on Ethereum. Native Mainnet ELY does not exist until Mainnet activation and the 1:1 bridge migration described in Chapter 11.

Network Architecture



Developer Environment

Developers can interact with Elyra using standard Ethereum tooling:

MetaMask · Hardhat · Foundry · Remix · EVM-compatible frameworks

Wallet Configuration

FIELD	VALUE
Network Name	Elyra Genesis Testnet
RPC URL	Provided through official documentation
Chain ID	Provided through official documentation
Currency Symbol	ELY
Block Explorer	Provided through official documentation

NOTE

Network parameters may evolve during the Genesis Testnet phase. Always verify the latest configuration through official Elyra documentation.

Gas Model

```
Transaction Cost = Gas Used × Gas Price
```

Smart Contract Deployment

Write Contract → Compile → Deploy → Verify → Interact

Example Contract

```
contract ElyraExample {
    uint256 public value;

    function updateValue(uint256 newValue) public {
        value = newValue;
    }
}
```

```
npx hardhat compile
npx hardhat run deploy.js --network elyra
```

RPC Interface

METHOD	PURPOSE
eth_blockNumber	Retrieve latest block
eth_getBlockByNumber	Retrieve block data
eth_sendTransaction	Submit transaction
eth_call	Execute contract call
eth_getTransactionReceipt	Retrieve transaction status

Block Explorer

The Elyra Explorer provides transparency into network activity: blocks, transactions, addresses, smart contracts and token activity.

Genesis Testnet Objectives

Network Stability

Testing block production, transaction processing and infrastructure reliability.

Developer Experience

Evaluating tool compatibility, deployment workflows and documentation quality.

Ecosystem Preparation

Preparing the foundations for applications, builders and community participation.

Future Network Evolution



Appendix E — Agent Infrastructure Roadmap

Agent Identity

A protocol-native identity layer designed to enable autonomous entities to possess persistent identities: agent authentication, ownership attribution, permission management.

Agent Wallet

A programmable asset management layer designed for autonomous entities: controlled asset ownership, automated transactions, permission-based execution.

Agent Runtime

An execution environment designed to coordinate autonomous operations: task execution, context management, protocol interaction.

Autonomous Execution Engine



Execution Memory

A future protocol component designed to provide persistent operational context: agent history, execution context, interaction state.

Final Technical Vision

The Genesis Testnet establishes the foundation. Future Elyra versions expand this foundation toward an autonomous execution network where intelligent systems can operate securely, transparently and verifiably.