

Aleksandr Vasilev — AI Systems Architect · Founder & Chief Architect, [AgentStack.tech](#)



I turn ambiguous, high-stakes systems — AI, money, real-time — into boring, predictable infrastructure.

I build systems where **AI carries real production load**—and I have spent **15+ years** shipping backends and platforms that stay predictable, observable, and operable when usage grows.

[AgentStack.tech](#) is a patent-pending, AI-first *Universal Processing System*: full backend, admin surfaces, and a unified integration plane—TypeScript/Python/React SDK, REST, and MCP with **800+ catalog actions** spanning projects, auth, RBAC, billing, wallets & buffs, scheduler, webhooks, analytics, AgentSocial, SSE messenger, RAG & agent memory, CRM, commerce & marketplace, bots & integrations, an on-chain AgentNet economy, and server-side logic.

Role: Founder & Chief Architect—product architecture, protocol & SDK design, multi-tenant data model, AI instruction systems, and end-to-end delivery (dual-shell SPA, TanStack Query, SSE + WebSockets, PostgreSQL, Stripe, Solidity/EVM, observability). **Through-line:** performance discipline from AAA-style game engineering applied to AI-native infrastructure.

15+ Years shipping systems	800+ Catalog actions (MCP + REST) · 58 domains	409 Architecture Decision Records	<0.001s Neural Router target latency (hot path)
--------------------------------------	--	---	--

AgentStack v0.4.22 · Verified 2026-10-09 · [machine-readable manifest](#)

Platform Thesis

AgentStack is designed so humans and AI agents share the same contract: typed SDK, REST, and MCP—with discovery, documented workflows, and governance baked in—not bolted on.

Most “AI backends” glue an LLM to CRUD. AgentStack is inverted: a **multi-tenant universal processing core** with **Field Access Policy, Protein Command System, Logic Engine, RAG**, commerce, social, realtime, and an on-chain **AgentNet economy**—then a first-class **MCP + REST catalog (800+ actions)** so agents can operate the same platform developers do.

Connect time for new products is measured in minutes because migrations, rollback, RBAC, billing, and observability are part of the architecture—not weekend glue code.

Architectural pillars

8DNA · Universal DNA layer Service orchestration with a universal DNA data model—same conceptual backbone across SaaS, games, social, and commerce verticals.	Neural Router · Hot-path latency O(1)-style routing with aggressive caching philosophy—documented target <0.001s on the router hot path; built for horizontal scale.
Protein Command System Universal command bus aligned with automation, internal tooling, and MCP-shaped execution—one vocabulary for scripts and agents.	Field Access Policy (FAP) Role- and user-scoped field masks and triggers—secure exposure for REST, SDK, and LLM-mediated access.
Logic Engine Server-side event-driven rules and processors—business logic without shipping a new microservice for every branch.	MCP integration plane agentstack.execute with 800+ actions, discovery, health, and bundled AI prompts (workflows + use-case packs).
Dual-shell product UI Separate user vs developer/admin experiences (mobile-first), TanStack Query server state, 600+ component modules, i18n.	Realtime split SSE for messenger & social streams; WebSockets for operational/build telemetry—clean separation of product vs ops channels.

AI Architecture Practice

How AI is constrained to behave well in production—not only which model you call.

Great AI systems are **interfaces + data + policy**. Models change weekly; contracts and governance stay.

- **Instruction design:** prevention-over-cure prompts, structured workflows (understand → plan → build → publish), MCP ai_prompt bundles and use-case packs (game, SaaS, ecommerce, social).
- **RAG & memory:** collections, semantic search, and long-term agent memory as product surfaces—not ad hoc vectors in a notebook.
- **Safety & tenancy:** API keys with scopes/caps, RBAC, tenant isolation, Field Access Policy for safe field exposure to humans and agents.
- **Observability:** traces and structured logs for AI-mediated paths—debuggable automation, not opaque chat.
- **Evaluation mindset:** ship loops where agent behavior is observable and improvable—schemas, discovery, and regression-friendly workflows.

Proof · Metrics & Highlights

Quantified engineering outcomes where documented—plus shipped scope that demonstrates architectural ownership. Beyond features: architecture decisions are written down, and invariants are enforced by CI — not by memory.

Platform engineering outcomes

These figures appear in AgentStack engineering documentation as consolidation and neural-architecture targets—they summarize intentional core unification, not incidental tuning.

Dimension	Outcome	Notes
Core consolidation	~87% fewer services	Documented reduction (15 → 2 primary core services) to cut coordination and ops drag.
Memory footprint	~76% reduction	Documented unified-core target (822MB → 200MB class targets in architecture docs).
Neural Router	<0.001s · O(1) routing	Hot-path latency budget; paired with neural cache discipline.
Neural events	200,000+ events/s	Throughput class for event-driven orchestration paths.
Neural cache	95%+ hit-rate target	Designed into routing and read-heavy platform paths.
MCP surface	800+ actions	Single discovery-oriented catalog for agents—same platform as humans.
Product UI kit	600+ component modules · i18n	Shipping velocity for admin + dashboard surfaces ('agentstack-frontend/src/components/').
Architecture Decision Records	409 ADRs	One documented decision per major subsystem — 'docs/adr/', not tribal knowledge.
Architecture-invariant CI audits	202 audit scripts	Automated boundary checks (shell parity, SEO surface, hosting golden paths) beyond unit tests.
End-to-end test coverage	118 Playwright specs	Hosting, commerce, CRM, bots, integrations, security flows.

Figures cited from internal/platform engineering documentation (architecture guides, neural router targets) and direct repository counts (ADRs, CI scripts, e2e specs), verified on the date noted in each row's source comment. Treat as engineering claims tied to design intent—always happy to discuss methodology on a call.

Codebase scale: **3,600+ Python modules** (agentstack-core/) and **4,000+ TypeScript/TSX modules** (agentstack-frontend/src/) — file counts, not line counts, as the more defensible metric.

Signature achievements

Founded & architected AgentStack — patent-pending AI-first Universal Processing System; unified SDK + REST + MCP (800+ actions); multi-tenant clusters with instant migrations/rollback posture.

Shipped full-stack ecosystem — OAuth/2FA, RBAC, billing & wallets, buff/trial system, scheduler & webhooks, AgentSocial + SSE messenger + federation hooks, RAG KB & agent memory, CRM, commerce & marketplace, bots fleet, AgentNet Web3 economy, game economy, Logic Engine.

Defined AI instruction methodology — minimalism, safety-first, decomposition; prevention-over-cure prompt design for Builder + MCP agents; bundled workflow & use-case documentation.

Game-industry backbone (8+ years) — multiplayer systems, TCP/UDP networking, performance-sensitive gameplay code, shipped titles across PC, iOS, Android, PlayStation & Xbox with Unity/C#.

Studio track record — Skybringer: Idle RPG (Dreamfrost); Block N Load & Immortal: Unchained + Chronos analytics (Toadman); Entropy MMORPG & Grimm: Dark Legacy (ArtPlant)—plus indie full-stack at World4Play.

Platform expansion, 2025–2026

Continuous ownership, not a one-time build: shipped major subsystems on top of the original core.

- **Commerce & Marketplace:** Storefront Studio, seller activation, fulfillment orchestrator, cross-project marketplace checkout.
- **Automation:** Bots Fleet (Telegram/WhatsApp/Instagram), Integration Hub (Zapier-class webhooks + recipes), Logic Engine V2 (visual flow builder, durable sagas).
- **CRM & Support:** project-scoped CRM tissue, in-app AI-assisted customer support channel.
- **AI Builder Modular Renaissance:** manifest-driven deterministic app composition, replacing prose-regenerated code with validated, testable builds.
- **Messenger reliability plane (0.4.17):** unified history assembly, relay resume, local-first contours — CRDT bridge partial per ADR; SSE product streams + SharedWorker shell.
- **Connect & Automate (0.4.15):** Integration Hub GA — App Directory, OAuth, DataMapper, checkout→CRM recipe on Logic Engine.
- **Docs Living Plane (0.4.18):** MCP stats SoT chain (matrix → snapshot → plugin/SDK); genetic-tag AI navigation genome.
- **Tenant 8DNA generation supply (0.4.18):** sandbox fork → diff → gates → promote; deploy/tenant boundary enforced in CI.
- **Navigation & discovery:** Platform Compass (unified omnibox), Discovery Hub, dual-shell audience-aware SPA, PTC comfort tasks.
- **Own AI-navigation framework:** a genetic-tag indexing system (domain.subsystem.role.genN) that lets AI coding agents locate the right module tree in a large codebase without full-repo search — later exported as a portable starter kit for other projects.

Case study — from framework conflict to platform primitive

Faced with LLM-generated code breaking on templating-syntax collisions across products, instead of patching each occurrence, redesigned the underlying command/template contract so the conflict class became structurally impossible — a pattern later reused across the AI Builder pipeline's validation gates.

Technology Stack

Backend (FastAPI, Pydantic, SQLAlchemy, 8DNA / Neural Router patterns, in-process work queue & cache — no Redis/Kafka dependency), frontend (React, Vite, Tailwind, **dual-shell SPA**, TanStack Query), data (PostgreSQL), **LLM application architecture** (MCP catalog, custom RAG engine, agent memory, agentic workflows), commerce (Stripe, wallets, buffs), real-time (**SSE** messenger, WebSocket ops), **Web3** (Solidity, multi-chain testnet orchestration). Observability: Loguru, OpenTelemetry. **Unified SDK** (TypeScript @agentstack/sdk, React hooks). Cloud-native, multi-tenant, event-driven.

Programming Languages

Python

Advanced

FastAPI, Data processing, AI/ML integration, Automation

TypeScript/JavaScript

Advanced

React, Node.js, Real-time systems

C#

Expert

Game development, Backend services, ASP.NET Core

Java

Advanced

Enterprise architecture, Microservices, System design

SQL

Expert

PostgreSQL, Complex queries, Database optimization

Backend Frameworks & Platforms

FastAPI

Advanced

REST APIs, async/await, MCP server, AgentStack core

Uvicorn / Starlette

Advanced

ASGI server, WebSocket support, production deployment

WebSockets & SSE

Expert

WebSocket: build status, dashboards. SSE: messenger & real-time social (AgentStack)

Pydantic

Advanced

Data validation, settings, API schemas (v2)

Node.js

Advanced

Event-driven architecture, Real-time communication

ASP.NET Core

Advanced

Enterprise applications, Microservices

Django/Flask

Advanced

Rapid development, Data processing (pre-AgentStack projects)

Frontend & Build (AgentStack)

React 18

Advanced

Hooks, concurrent features, dashboard & builder UI

TypeScript

Advanced

Strict mode, SDK types, frontend and tooling

Vite

Advanced

Build tooling, HMR, React plugin, base path

React Router v6

Advanced

SPA routing, nested routes, loaders

TanStack Query

Advanced

Server state, caching, invalidation (AgentStack)

Tailwind CSS

Advanced

Utility-first, design tokens, responsive UI

React Flow

Advanced

Node-based UIs, logic builder, diagrams

i18next / react-i18next

Advanced

Internationalization, locale switching

Unified SDK & hooks

Advanced

@agentstack/sdk, React package, protein-style API consumption

Vitest

Advanced

Frontend unit/integration tests, Vite-native runner (AgentStack)

API & Data Layer

SQLAlchemy 2 (async)

Advanced

ORM, migrations, async engine, AgentStack core

asyncpg

Advanced

Async PostgreSQL driver, high performance

httpx

Advanced

Async HTTP client, outbound API calls

Databases & Data Storage

PostgreSQL

Expert

Schema design, Complex queries, Performance tuning, JSONB

8DNA-native work queue & cache

Advanced

In-process queue/cache on unified DNA table (AgentStack) — chosen over Redis/Kafka to cut ops surface; Redis used in earlier enterprise/game-backend roles for caching & pub/sub

MongoDB

Advanced

Document storage, flexible schemas (pre-AgentStack projects; AgentStack itself is PostgreSQL-native)

Firebase

Advanced

Real-time databases, authentication, cloud functions (pre-AgentStack / indie projects)

Elasticsearch

Advanced

Full-text search, log aggregation (pre-AgentStack projects)

Auth, Security & Payments

JWT / bcrypt

Expert

Authentication, token validation, password hashing

Stripe

Advanced

Payment gateway, subscriptions, AgentStack billing

cryptography

Advanced

API keys, sensitive data, TLS

Observability & Tooling

Loguru

Advanced

Structured logging, rotation, AgentStack core

OpenTelemetry

Advanced

Traces, instrumentation (FastAPI, HTTP)

pytest / Jest

Expert

Backend and frontend testing, async tests

ESLint / TypeScript

Expert

Code quality, type-check, build pipeline

AI / LLM Expertise

Orchestration of LLMs and agents, **multi-agent patterns**, MCP at scale (**800+ catalog actions**), discovery, workflow docs, use-case prompts (game, SaaS, ecommerce, social). **Custom-built RAG engine** (self-authored vector search & compression, no third-party vector DB), collections, semantic search, agent long-term memory. **AI instruction design:** system prompt bundles, structured workflows (understand → plan → build → publish), prevention-over-cure. Providers: OpenAI, Anthropic, local LLMs (Ollama, vLLM). Orchestration: self-built multi-stage pipeline (AgentStack AI Builder); LangChain/LangGraph evaluated in earlier prototypes, not in AgentStack production.

AI/LLM Stack — Models, Frameworks, Infra

OpenAI / Anthropic

Advanced

API integration, function calling, structured outputs

Ollama / vLLM

Advanced

Local LLM deployment, model management, self-hosted

Custom AI Builder Orchestrator

Advanced

Self-built multi-stage pipeline (understand → plan → build → publish) with gate/repair-loop validation — not LangChain/LangGraph; those were evaluated in earlier, pre-AgentStack prototypes

Custom RAG Engine (TurboQuant)

Advanced

Self-built vector engine — Fast Walsh-Hadamard rotation + outlier-aware quantization (HGT algorithm from Google Research/DeepMind, ICLR 2026), ~8× compression, no third-party vector DB dependency

MCP Protocol

Expert

agentstack.execute, 800+ actions, discovery, AI prompt bundle

RAG Systems

Advanced

Knowledge-augmented generation, retrieval, evals

Prompt / Instruction Design

Advanced

System prompts, workflows, MCP ai_prompt / use-case bundles, AI Builder

Web3 / Blockchain — AgentNet Economy

Designed and shipped an on-chain economy layer (**AgentNet**) with an append-only Postgres ledger as financial source of truth, EVM smart contracts as public anchors, and multi-chain test-net orchestration — not a token bolted onto a database.

Status: **testnet / pre-mainnet evidence stage** (Base Sepolia, BSC testnet, Arbitrum Sepolia). Engineering framing, not a financial offering: this section describes vault accounting mechanics and a reference implementation, not investment returns.

Solidity Smart Contracts

Advanced

ERC-4626 yield vault (agUSD over USDT), ERC-8004 execution-proof registries, EIP-712 signed envelopes, Timelock governance

Multi-chain orchestration

Advanced

Base Sepolia, BSC testnet, Arbitrum Sepolia via a single ChainProfileRegistry; faucet + scenario orchestrator

On-chain anchoring & proofs

Advanced

Merkle checkpoints, hash-chained ledger batches, public verifiability without a shadow source of truth

Bridge & token economics

Advanced

Rate-limited, finality-aware mint/burn bridge policy; AGNT (utility) vs agUSD (vault shares) vs USDT payments — three separated rails by design

Live evidence lab: agentstack.tech/grants/demo — public multi-chain proof timeline.

Verified testnet contract (AGNT, Base Sepolia) — cross-checked against docs/grants/evidence/grant_audit_pack_latest.json, 2026-07-07: sepolia.basescan.org/address/0x38aD...9a943 — source-verified, compiler v0.8.24. Testnet contracts may be redeployed; re-verify before relying on this address.

DevOps & Infrastructure

Docker

Expert

Containerization, Image optimization, Multi-container setup

Kubernetes

Advanced

Orchestration, Scaling, Deployment

AWS

Advanced

EC2, Lambda, RDS, S3, CloudFront, API Gateway

GitHub/GitLab

Expert

Version control, CI/CD, Complex workflows

Skill Matrix

Aligned with shipped platform work — multi-tenant core, MCP catalog, social & messenger (SSE), RAG, billing, unified SDK, Web3/AgentNet, observability.

Core competencies

Python · TypeScript · FastAPI · React · PostgreSQL · Distributed Systems · System Architecture · Multi-tenant SaaS · MCP Protocol · RAG · Agentic AI · Solidity · Web3 / Blockchain · Event-Driven Architecture · CQRS · Real-time Systems · CI/CD · AI Systems Architecture

Backend Architecture & System Design

Skill	Level	Experience	Context
System Architecture Design	Expert	10+ years	Game servers, microservices, distributed systems
Microservices / multi-tenant SaaS	Advanced	8+ years	Per-tenant clusters, migrations, AgentStack production
API Design (REST/GraphQL)	Expert	10+ years	RESTful principles, GraphQL implementation
Database Schema Design	Advanced	10+ years	PostgreSQL, MongoDB, normalization, optimization
Real-time Systems Architecture	Expert	8+ years	WebSocket, multiplayer backends, live data, AgentStack build status
Pydantic / API Schemas	Advanced	3+ years	Validation, settings, OpenAPI, AgentStack core
SQLAlchemy (async)	Advanced	3+ years	ORM, migrations, PostgreSQL, AgentStack
Field Access Policy (FAP)	Advanced	2+ years	Role/user field masks, triggers, secure data exposure
Protein command bus / DNA ops	Advanced	2+ years	Universal automation surface, integrations

Frontend & Build

Technology	Level	Experience	Context
React (hooks, concurrent)	Advanced	5+ years	Dashboard, builder UI, AgentStack frontend
TypeScript	Advanced	6+ years	Strict mode, SDK types, frontend and tooling
Vite	Advanced	2+ years	Build, HMR, React plugin, AgentStack
TanStack Query	Advanced	2+ years	Server state, caching, AgentStack
Tailwind CSS	Advanced	3+ years	Utility-first, responsive UI
React Flow	Advanced	2+ years	Node-based UIs, logic builder
Dual-shell SPA / audience routing	Advanced	1+ years	User vs dev surfaces, mobile-first admin (AgentStack)
SSE & real-time UX	Advanced	2+ years	Messenger streams, social presence patterns

Programming Languages

Language	Level	Years	Current Use
Python	Advanced	5+	FastAPI, Pydantic, AgentStack core, AI/ML
TypeScript/JavaScript	Advanced	6+	React, Vite, TanStack Query, AgentStack frontend
C#	Expert	10+	Legacy game dev, ASP.NET
Java	Advanced	5+	Enterprise systems, Architecture
SQL	Expert	10+	Complex queries, optimization

AI/LLM Integration & Optimization

Technology	Level	Experience	Specialization
Ollama	Advanced	2+ years	Local LLM deployment, model management
LangChain / LangGraph	Advanced	2+ years	LLM chains, prompt engineering, memory — evaluated in earlier prototypes, not in AgentStack production (AgentStack uses a custom orchestrator, see below)
Custom RAG Engine (TurboQuant)	Advanced	2+ years	Self-built vector search — Fast Walsh-Hadamard rotation + outlier-aware quantization, ~8× compression; no third-party vector DB
MCP Protocol	Expert	2+ years	800+ actions, discovery, workflow & use-case prompts

Technology	Level	Experience	Specialization
RAG Systems	Advanced	2+ years	Collections, semantic search, agent memory (production)
Custom AI Builder orchestrator	Advanced	2+ years	Multi-stage pipeline (understand → plan → build → publish), gate/repair-loop, function calling

Web3 / Blockchain

Technology	Level	Experience	Context
Solidity / EVM contracts	Advanced	1+ years	ERC-4626 vault, ERC-8004 proof registries, EIP-712, Timelock governance (AgentNet)
Multi-chain orchestration	Advanced	1+ years	Base Sepolia, BSC testnet, Arbitrum Sepolia via ChainProfileRegistry
On-chain anchoring & ledger design	Advanced	1+ years	Merkle checkpoints, hash-chained batches, Postgres ledger as source of truth
Bridge & token economics design	Advanced	1+ years	Rate-limited mint/burn policy, AGNT/agUSD/USDT rail separation

Auth, Payments & Observability

Area	Level	Experience	Context
JWT / bcrypt / Auth	Expert	8+ years	Token validation, password hashing, AgentStack
Stripe / wallets / buffs	Advanced	2+ years	Subscriptions, trials, multi-wallet economy
Loguru / OpenTelemetry	Advanced	2+ years	Structured logging, traces, AgentStack
pytest / Jest	Expert	8+ years	Backend and frontend testing, async, AgentStack

Expertise Tiers

Tier 1 - Expert Level (90-100%)

System Architecture · Backend API · Database Design · Real-time (WS + SSE) · CI/CD & DevOps · SOLID & Patterns · Problem-Solving · Game Development · MCP at Scale

Tier 2 - Advanced Level (70-89%)

Python · TypeScript · C# · Java · FastAPI · Pydantic · SQLAlchemy · React · Vite · TanStack Query · Tailwind · React Flow · PostgreSQL · 8DNA Work Queue · AWS · Docker · Kubernetes · Stripe · Loguru · Ollama · LangChain · RAG · MCP Protocol · Unified SDK · AgentSocial · Multi-tenant SaaS · FAP / Data Governance · Solidity · Web3 / Blockchain

Tier 3 - Intermediate Level (50-69%)

Vue.js · Google Cloud · Apache Spark

Professional Experience

15+ years in backend and distributed systems; game development, full-stack, AI/LLM, and Web3. [AgentStack](#) — Universal Processing System: MCP + REST (800+ actions), social & messenger, RAG, commerce, AgentNet economy, unified SDK, dual-shell product UI.

Founder · Chief Architect & CTO — [AgentStack.tech](#)
[AgentStack Platform](#)

- Designed and shipped a **universal backend + admin UI** with MCP + REST surface (**800+ actions**), and TypeScript/Python/React SDK — SaaS, games, social, commerce, and internal tools from one stack.
- Delivered modular subsystems: auth (OAuth, 2FA), RBAC, scheduler, billing & wallets, buffs/trials, analytics, logic engine & processors, webhooks, AgentSocial (friends, followers, feed), **messenger** (SSE streams, relay resume, local-first reliability plane), RAG KB & agent memory, CRM, commerce & marketplace, bots & integration hub, game economy.
- Platform architecture: multi-tenant clusters, Field Access Policy, protein command bus, Neural Router-style fast routing, instant migrations — aligned with production scale and ops.
- **Designed and shipped AgentNet** — an on-chain economy layer (Solidity, ERC-4626 vault, ERC-8004 proof registries, multi-chain testnet orchestration) with an append-only ledger as financial source of truth and EVM contracts as public anchors, not a token bolted onto a database. See [Web3 / AgentNet](#).
- Frontend: dual-shell SPA (user vs developer), TanStack Query, 600+ component modules, 118n; AI-facing docs and MCP bundles (workflows, use-case prompts).
- Defined **AI instruction design** for Builder and MCP agents: minimalism, safety-first, decomposition; prevention-over-cure prompts and structured workflows.
- **Solo-architected end-to-end** — from empty repository to a multi-tenant platform with 800+ catalog actions, 409 ADRs, and 202 root CI audit scripts, without a dedicated human engineering team; used an AI-agent-augmented workflow (structured prompts, genetic-tag codebase navigation, controlled-change discipline) as a force multiplier — itself a case study in AI-assisted systems engineering.
- **Impact:** Fast MVP delivery, low integration friction, predictable TCO; stability and scalability as first-class product constraints.

Tech: Python, FastAPI, PostgreSQL, Docker, Compose, queues, Stripe and payment rails, Solidity/EVM. React, TypeScript, Vite, Vitest. MCP, RAG, WebSockets (ops), SSE (chat).
Observability: Loguru, OpenTelemetry. IDE/agent plugins: Cursor, Claude, GPT, VS Code.

Python · FastAPI · React · MCP · RAG · Agentic AI · Prompt Design · Stripe · SSE · AgentSocial · Unified SDK · Solidity / Web3 · System Architecture

Senior Unity Developer

Dreamfrost Studio

Led development of **Skybringer: Idle RPG**—adaptive UI, core mechanics, SDK expansion —shipping on mobile and desktop-class targets.

Unity · C# · Game Mechanics · UI/UX

Indie Game & Software Developer

World4Play

- Owned architecture end to end for a small indie studio — solo/small-team delivery across design, backend, and client.
- Built cloud-backed multiplayer services (matchmaking, save-state sync, cross-platform accounts) reused across multiple prototypes instead of rebuilt per game.
- **Impact:** shipped playable, monetizable builds on a small-team budget by treating infrastructure as a reusable asset, not a one-off cost per project.

Unity · C# · Full-Stack · Cloud

2018 - 2020

Senior Unity Developer

Toadman Interactive (Stockholm)

- Shipped multiplayer FPS **Block N Load** and hardcore action-RPG **Immortal: Unchained** on PC and consoles (PlayStation, Xbox) — performance-critical gameplay code under fixed console cert deadlines.
- Built **Chronos**, an internal data-analytics platform used across studio projects — first exposure to production data pipelines outside of game-client code.
- **Impact:** stable multiplayer netcode under console certification constraints; shared tooling reduced duplicated analytics work across titles.

Multiplayer · Performance · Data Analytics

2014 - 2018

Unity Developer

ArtPlant (Oslo)

- Shipped MMORPG **Entropy** and cooperative survival **Grimm: Dark Legacy** — networking-heavy gameplay with persistent world state across PC & consoles.
- Built and iterated mechanics prototypes used to validate game design decisions before full production commitment.
- **Impact:** early foundation in networked, stateful systems that later generalized directly into distributed backend and real-time messenger work.

Unity · Game Design · Networking

Education & Background

Self-taught, portfolio-driven engineer — no formal CS degree.
15+ years of shipped production systems (games → distributed backends → AI infrastructure) as the primary credential;
architecture decisions are documented in 409 ADRs rather than claimed by title.

Languages

English — professional working proficiency · Russian — native ·
Portuguese — basic / learning

Get in Touch

Consulting and collaboration: AI systems architecture, MCP integration (large action catalogs), RAG & agent memory, multi-tenant BaaS, billing, real-time social, Web3/AgentNet — and [AgentStack](#) ecosystem.



Aleksandr Vasilev

AI Systems Architect · Full-Stack · Founder, [AgentStack.tech](#)

Open to: Fractional CTO / architecture consulting · Technical co-founder engagements · Staff/Principal Architect roles · Investment & strategic partnership (AgentStack)

Availability

Remote-first · GMT-3 (Brazil, Santos) · Open to discussing new engagements

Location

[Santos, São Paulo, Brazil](#)

WhatsApp

[+55 \(13\) 99800-4433](#)

Telegram

Email

lance@world4play.com

LinkedIn

linkedin.com/in/alexanderlance

Phone

[+55 \(13\) 99800-4433](#)

On this page

- [Trust & reliability](#)
- [Overview](#)
- [AI practice](#)
- [Proof & metrics](#)
- [Tech stack](#)
- [Web3 / AgentNet](#)
- [Skills](#)
- [Experience](#)
- [Contact](#)

Proof & artifacts

- [cv.json manifest](#)
- [Cover letter library](#)
- [MCP discovery](#)
- [Grant demo](#)

Connect

- [Email](#)
- [Phone](#)
- [WhatsApp](#)
- [Telegram](#)
- [LinkedIn](#)
- [AgentStack.tech](#)