

ARBuilder Technical Brief

Project Overview

ARBuilder is an AI-powered development assistant for the Arbitrum ecosystem. It transforms natural language prompts into:

- Stylus smart contracts (Rust)
- Cross-chain SDK implementations (asset bridging and messaging)
- Full-stack dApps (contracts + backend + indexer + oracle + frontend + wallet integration)
- Orbit chain deployment assistance

Core Technology: Retrieval-Augmented Generation (RAG) pipeline

Delivery Method: MCP server integration with Cursor/VS Code

Baseline Repositories

Repo	Link
ICP Coder	https://github.com/Quantum3-Labs/icp-coder
Stacks Builder	https://github.com/Quantum3-Labs/stacks-builder

Reusable Components:

- RAG pipeline architecture
 - Vector database setup
 - MCP server integration (Cursor/VS Code)
 - Document ingestion pipeline
 - Prompt engineering framework
 - Documentation structure
-

Milestones

M1: Ingestion of Documentation and Code, & Stylus Smart Contract Builder

Deadline Completion

12/26/2025 70%

Tasks:

- Implement ARBuilder based on RAG Architecture
- Collect Arbitrum Stylus documentation and curated code/project examples
- Data cleaning and preprocessing
- Build ingestion pipeline and embed into vector database
- Develop natural-language to Rust Stylus code generation
- Enable Q&A, design, writing, and testing of Stylus smart contracts
- Integrate with Cursor/VS Code via MCP server + manifest
- Produce tutorial video (installation, configuration, usage)
- Create README and user guide

Outcomes:

- ARBuilder v0.1 released
 - IDE integration (Cursor/VS Code via MCP)
 - Open-source under MIT license
 - Tutorial video + README + user guide
-

M2: Arbitrum SDK Integration

Deadline Completion

01/09/2026 70%

Tasks:

- Expand AI assistant for cross-chain messaging: Ethereum → Arbitrum, Arbitrum → Ethereum
- Add support for asset bridging: EthBridger, Erc20Bridger, EthL1L3Bridger

Outcomes:

- ARBuilder v0.2 released
 - Cross-chain messaging support
 - Token bridging support
-

M3: Full dApp Builder

Deadline **Completion**

01/23/2026 20%

Tasks:

- Research and enhance RAG with Hybrid approach (BM25 + vector embeddings)
- Explore advanced solutions: Agentic RAG, Multi-Stage RAG, Feedback-Driven RAG
- Select and implement one additional RAG solution
- Extend ARBuilder to generate scaffolded dApps:
 - Smart contracts (Stylus)
 - Backend (NestJS/Express)
 - Indexer (subgraphs)
 - Oracle integration (Chainlink)
 - Frontend (Next.js/React)
 - Wallet integration (wagmi, viem, RainbowKit)
 - UI styling (DaisyUI)
- Improve performance and accuracy

Outcomes:

- ARBuilder v0.3 released
 - End-to-end dApp scaffolding from natural language
-

M4: Orbit Chain Integration

Deadline **Completion**

02/06/2026 10%

Tasks:

- Further improve core generation engine for Stylus and Arbitrum SDK
- Enable Orbit chain deployment assistance:
 - Chain configuration
 - Validator setup
 - Custom gas tokens
 - Fee parameters
 - Migration guidance
- Expand support for Orbit SDK applications

Outcomes:

- ARBuilder v0.4 released
- Orbit chain deployment guidance
- Orbit SDK code generation

M5: Unified AI Assistant Experience & Final Report

Deadline Completion

02/12/2026 0%

Tasks:

- Benchmark performance across query difficulty levels (basic, intermediate, advanced)
- Track adoption metrics
- Deliver final report with results, benchmarks, future roadmap

Outcomes:

- ARBuilder v1.0 released (unified assistant)
- Final report delivered
- Public adoption metrics achieved

KPIs:

Category	Target
Developer Reach	≥10 unique developers
	≥100 successful code generation requests
Community	≥20 GitHub stars
	≥5 community contributions (PRs, issues)
	≥3 documented case studies
Ecosystem	≥5 projects deployed to Arbitrum mainnet

Data Sources

Stylus (M1)

Official Documentation:

- <https://arbitrum.io/stylus>
- <https://docs.arbitrum.io/get-started/overview>

Curated Resources:

- <https://github.com/OffchainLabs/awesome-stylus>

Official Examples:

- <https://github.com/OffchainLabs/stylus-chess/tree/update>
- <https://github.com/OffchainLabs/stylus-by-example>
- <https://github.com/OffchainLabs/stylus-hello-world>
- <https://github.com/OffchainLabs/stylus-quickstart-vending-machine>
- <https://github.com/ArbitrumFoundation/stylus-workshop-gol>

Production Codebases:

- <https://github.com/fluidity-money/9lives.so>
- <https://github.com/fluidity-money/long.so>
- <https://github.com/OpenZeppelin/rust-contracts-stylus>
- <https://github.com/renegade-fi/renegade-contracts>
- <https://github.com/stylus-developers-guild/reentrancy-transient-storage>

Community Projects:

- <https://github.com/hammertoe/ArbitrumOnchainAgent>
- <https://github.com/philogicae/ethbuc2025-gyges>
- <https://github.com/Oluwatobilobaake/erc6909-with-arbitrum-stylus>
- <https://github.com/hummusonrails/fortune-generator>

Blog/Articles:

- <https://blog.arbitrum.io/how-thirdweb-uses-arbitrum-stylus-to-power-the-next-wave-of-on-chain-apps/>

Arbitrum SDK (M2)

- <https://github.com/OffchainLabs/arbitrum-sdk>

Orbit SDK (M4)

- <https://github.com/OffchainLabs/arbitrum-orbit-sdk>

- <https://docs.superposition.so/>
-

Technical Notes

Version Handling

System should handle:

- New version releases with features, deprecations, breaking changes
- Migration suggestions for old code
- Version detection in codebase to provide relevant assistance

RAG Architecture

M1/M2: Basic RAG with vector embeddings

M3 onwards: Hybrid RAG

- Vector embeddings (semantic similarity)
 - BM25 (keyword-based)
 - Reciprocal Rank Fusion (RRF) for ranking
 - Multi-retrieval system
-

Repository

Code hosted under Q3Labs GitHub organization.