UPDATE -

This Constitutional AIP is now live on [Tally

](https://www.tally.xyz/gov/arbitrum/proposal/108288822474129076868455956066667369439381709547570289793612729242368710728616) - voting starts on Thursday, August 1st, at 13:46 UTC.

There has been an error in 2 URLs on the text of the Onchain AIP on Tally, and the correct URLs are listed below.

To clarify, the payload and custom upgrade actions in the AIP are still correct - the only errors were in the description of the proposal.

Wrong Addresses:

- Arb1 SetArbOS31VersionAction: SetArbOS31VersionAction | Address 0x7ed9C3A779BE8b742AbFC17a2F15353ecBcE3e00 | Arbitrum One
- Nova SetArbOS31VersionAction: SetArbOS31VersionAction | Address 0x75C5a4532102DDFa44527B382990C384Ec1dD57D | Arbiscan

Correct Addresses:

- Arb1 SetArbOS31VersionAction: SetArbOS31VersionAction | Address 0xaF81C82Ec98f86D0017d78cD66F1026f1A5Cf1Db | Arbitrum One
- Nova SetArbOS31VersionAction: SetArbOS31VersionAction | Address 0x6dD43360d2a69BB9FfFC5349F2511f2A3bCbC2da | Arbiscan

This update has also been highlighted at the very top of the first post. We apologize for this error, and are happy to answer any questions that delegates and the DAO might have.

Type: Constitutional

Abstract

This AIP proposes the activation of ArbOS 31

on Arbitrum One and Arbitrum Nova. This ArbOS upgrade brings a number of improvements, including:

- Activating Arbitrum Stylus to enable developers to build the next generation of Rust or C++ applications on the EVM.
- New precompile for verifying the secp256r1 elliptical curve (as part of RIP-7212)
- Change to fee collection on Arbitrum Nova, making it easier for ArbitrumDAO to manage and access the funds.

This AIP combines the following three temperature checks:

- AIP: Activate Stylus and Enable Next-Gen WebAssembly Smart Contracts
- AIP: Support RIP-7212 for Account Abstraction Wallets
- AIP: Nova Fee Router Proposal

A new ArbOS 31 “Bianca” was released between the time of the temperature check votes and submitting this proposal. ArbOS 31 release builds upon ArbOS 30 and includes new optimizations that were discovered during rigorous testing and feedback from Stylus teams. The ArbOS upgrade is shipped as a new Nitro node release alongside new upgrades to the rollup contracts for Arbitrum One and Arbitrum Nova.

Note: ArbOS 31 “Bianca” will be the canonical ArbOS version for the “Bianca” family of releases.

Since the DAO has already signaled support for the adoption of all three changes in all three temperature checks, this proposal will proceed to an on-chain AIP on Tally next.

Changes included

1. Activation of Arbitrum Stylus: a new WebAssembly-based (WASM) VM to support smart contracts written in Rust and C++

Stylus is an upgrade to the node software that powers all Arbitrum chains. This upgrade introduces a new WASM-based Virtual Machine (VM) that runs alongside the EVM. This enables developers to write smart contracts in new programming languages, like Rust, that are more efficient than Solidity smart contracts.

Stylus is a first-of-its-kind technology resulting from breakthrough engineering efforts in the Arbitrum ecosystem. Unlike other alternative VMs, the Stylus VM is not a replacement for the EVM & is instead purely additive to the EVM. This means that Stylus contracts and EVM contracts are fully interoperable. The two VMs work together to facilitate state transitions, each playing their part in executing their respective bytecode. Support for more memory efficient and safer languages will unlock a new generation of applications that were previously impossible to build on the EVM.

The Stylus VM and fraud prover were originally developed as a fork of theNitro codebase before. The Stylus codebase has since been audited (Trail of Bits audit report) and merged back into the Nitro codebase:

- GitHub - OffchainLabs/stylus: Stylus VM and Fraud Prover
- Merged into the Nitro codebase in:Arbitrum Stylus by rachel-bousfield · Pull Request #2242 · OffchainLabs/nitro · GitHubandStylus v2 by tsahee · Pull Request #2425 · OffchainLabs/nitro · GitHub
- Merged into the Nitro codebase in:Arbitrum Stylus by rachel-bousfield · Pull Request #2242 · OffchainLabs/nitro · GitHubandStylus v2 by tsahee · Pull Request #2425 · OffchainLabs/nitro · GitHub
- GitHub - OffchainLabs/stylus-geth
- Merged into Offchain Labs' fork of Geth in:Arbitrum Stylus by rachel-bousfield · Pull Request #305 · OffchainLabs/go-ethereum · GitHub
- Merged into Offchain Labs' fork of Geth in:Arbitrum Stylus by rachel-bousfield · Pull Request #305 · OffchainLabs/go-ethereum · GitHub
- GitHub - OffchainLabs/stylus-contracts
- Merged into the Nitro-contracts codebase in:Arbitrum Stylus by rachel-bousfield · Pull Request #170 · OffchainLabs/nitro-contracts · GitHub
- Merged into the Nitro-contracts codebase in:Arbitrum Stylus by rachel-bousfield · Pull Request #170 · OffchainLabs/nitro-contracts · GitHub

Stylus contracts can be written using the Stylus SDK, which employs Solidity-equivalent ABIs and storage patterns to ensure cross-language interoperability. For example, existing Solidity DEXs can list Rust tokens without modification. New SDKs for additional programming languages can be added over time. Current SDK repositories:

- Rust SDK: GitHub - OffchainLabs/stylus-sdk-rs: Rust Smart Contracts on Arbitrum
- C/C++ SDK: GitHub - OffchainLabs/stylus-sdk-c: C/C++ Smart Contracts on Arbitrum
- Stylus CLI: GitHub - OffchainLabs/cargo-stylus: Cargo subcommand for developing Arbitrum Stylus projects in Rust

If you would like to better understand the lifecycle of a Stylus contract, head over toA Gentle Introduction: Stylus. For teams who are curious to learn more about how Stylus is expected to interact with their project's existing infrastructure, we encourage folks to check out this Stylus launch ecosystem one-pager.

Note: Support for Stylus contract verification is currently under development by major block explorers. We expect that contract verification will be fully supported before ArbOS 31 Bianca gets activated on Arbitrum One and Arbitrum Nova, pending the DAO's decision to adopt this proposal.

2. Addition of a new pre-compile, to Arbitrum Nitro, that reduces the costs of verifying the secp256r1 elliptic curve as part of RIP-7212

Passkeys offer a solution that removes the need for a web3 user to personally store a private key for their wallet. Passkeys accomplish this by leveraging WebAuthn, a global standard for passwordless authentication used by Google, Facebook, Microsoft, and all major web browsers. The private key generated when creating a passkey can be encrypted and can only be decrypted using a specialized hardware module called the Secure Enclave. The Secure Enclave ensures a user's private key can never leave the device, transforming any compatible device into a hardware wallet. Users can authorize transactions with biometric features like Touch ID or Face ID when using passkey-based wallets for key management. These qualities add flexibility and significantly improve UX while maintaining high security.

Adding support for RIP-7212 decreases the costs of verifying the secp256r1 curve on-chainby 99% when compared to current implementations, making them more feasible for everyday use and enabling dApp developers and protocols to offer their users improved UX on Arbitrum One and Arbitrum Nova. Without a precompile, verifying this signature on-chain is extremely expensive. Passkey-based wallets offer a better level of security than a typical EOA and seamless cross-device support. Many wallets, and notably, apps using embedded wallets, have been requesting this feature for over a year.

The specifications of RIP-7212, including test cases, can be found in theRIP repository. If approved, Arbitrum One and Arbitrum Nova will use this specification as the reference for implementation. The Ethereum Magicians Forum discusses design decisions, iterations, and the transformation of the proposal from an EIP (Ethereum Improvement Proposal) to a RIP.

This pre-compile is part of Nitro 3.1.0 and was added to our fork of Go Ethereum inAdd Precompile for secp256r1 conditionally based on ArbOS version by anodar · Pull Request #303 · OffchainLabs/go-ethereum · GitHub and has been upstreamed into Nitro 3.1.0. Nitro 3.1.0 is the minimum supported version of Nitro for ArbOS 31 "Bianca".

3. Update all Nova fund distributors to use a series of "fee routers" instead, while keeping the final destination as the ArbitrumDAO Treasury

Today, the ArbitrumDAO's portion of the transaction fees from Arbitrum Nova are sent to theCore Governance L1 Timelock address, which is accessible via the core governance system. This setup is disadvantageous because any time the ArbitrumDAO wishes to spend/move the funds, a roundtrip, constitutional proposal must be passed before the DAO.

This proposal updates the setup such that all Arbitrum Nova transaction fees, that are currently sent to the Core Governance L1 Timelock Address, are instead sent to a system of "fee routers" that automatically send all funds to the ArbitrumDAO treasury. Benefits of this new setup include:

- Lower quorum requirements for moving these funds (3% vs. the current 5%)
- No ~2 week delay to spend funds (since a constitutional vote would not be needed, as per theLifecycle and anatomy of an AIP)

- Simpler accounting and bookkeeping, since all the funds would be in the ArbitrumDAO treasury.

This new, proposed system of “fee routers” results in a fee collection lifecycle as follows:

1. A distributeRewards

function call is made on the RewardDistributor

contract, sending funds to a ChildtoParentRouter

contract

1. Either upon receiving funds or via a call to routeFunds

, the ChildToParentRouter

creates an L2-to-L1 message which sends the contract’s full Ether balance.

1. The L2-to-L1 message is executed, transferring the Ether to a ParentToChildRouter

contract on L1.

1. routeFunds

is called on ParentToChildRouter

, creating a retryable ticket which transfers its full Ether balance to the DAO Treasury on Arbitrum One.

Note that the addresses deployed during theSnapshot vote for the ParentToChildRewardRouter

and ChildToParentRewardRouter

have been re-deployed at 0x40Cd7D713D7ae463f95cE5d342Ea6E7F5cF7C999

and 0x36D0170D92F66e8949eB276C3AC4FEA64f83704d

, respectively.

Implementation

All of the above changes were independently audited. The list below contains all relevant audit reports:

- Arbitrum Stylus audit report: publications/reviews/2024-05-offchain-arbitrumstylus-securityreview.pdf at master · trailofbits/publications · GitHub
- ArbOS 30 audit: publications/reviews/2024-04-offchain-arbos-30-nitro-upgrade-securityreview.pdf at master · trailofbits/publications · GitHub
- ArbOS 31 audit: publications/reviews/2024-04-offchain-arbos-31-securityreview.pdf at master · trailofbits/publications · GitHub
- Nova Fee router audit: Smart Contract Audit - Offchain Labs Fund Distribution - ChainSecurity

The canonical version of ArbOS 31 “Bianca” this proposal aims to adopt is implemented in the Arbitrum Nitro git commit hash 7d1d84c75db7fd26d27d24ffb75f8b1c93d4f980 and can be viewed in: Merge pull request #2485 from OffchainLabs/fix-disable-p2p · OffchainLabs/nitro@7d1d84c · GitHub.

The current full code diff can be viewed via this link:Comparing consensus-v20...consensus-v31 · OffchainLabs/nitro · GitHub.

ArbOS 31 “Bianca” will be shipped as part of a future release of Nitro. Node operators may see a small increase in latency for Stylus eth_call

queries, if ArbOS 31 is approved for activation on mainnet by the ArbitrumDAO.

Upgrade Action smart contracts

The Action smart contracts used to execute the on-chain upgrade can be viewed inArbos 31 actions by godzillaba · Pull Request #296 · ArbitrumFoundation/governance · GitHub.

Action contract deployments for ArbOS 31 and Stylus:

- Mainnet OneStepProofEntry (v2.0.0) (new osp) deployed at: OneStepProofEntry | Address 0xa328BAF257A937b7934429a5d8458d98693C6FC7 | Etherscan
- Mainnet OneStepProofEntry (v1.3.0) (cond osp) deployed at: OneStepProofEntry | Address 0x83fA8eD860514370fbcC5f04eA7969475F48CfEb | Etherscan
- Mainnet ChallengeManager (v2.0.0) deployed at: ChallengeManager | Address 0x914B7b3053B35B84A24df08D7c9ceBCaEA4E2948 | Etherscan
- Arb1 CacheManger deployed at: TransparentUpgradeableProxy | Address 0x51dedbd2f190e0696afbee5e60bfde96d86464ec | Arbitrum One

$ cast storage 0x51dEDBD2f190E0696AFbEE5E60bFdE96d86464ec 0xb53127684a568b3173ae13b9f8a6016e243e63b6e8ee1178d6a717850b5d6103 --rpc-url=https://arb1.arbitrum.io/rpc

0x000000000000000000000000db216562328215e010f819b5abe947bad4ca961e (Arb One Proxy Admin)

- Nova CacheManger deployed at: TransparentUpgradeableProxy | Address 0x20586F83bF11a7cee0A550C53B9DC9A5887de1b7 | Arbitrum Nova

$ cast storage 0x20586F83bF11a7cee0A550C53B9DC9A5887de1b7 0xb53127684a568b3173ae13b9f8a6016e243e63b6e8ee1178d6a717850b5d6103 --rpc-url=https://nova.arbitrum.io/rpc

0x000000000000000000000000f58ea15b20983116c21b05c876cc8e6cdae5c2b9 (Nova Proxy Admin)

- V31 Wasm Root: 0x260f5fa5c3176a856893642e149cf128b5a8de9f828afec8d11184415dd8dc69
- Deployed actions: Arbos 31 actions by godzillaba · Pull Request #296 · ArbitrumFoundation/governance · GitHub
- ArbOneAIPArbOS31UpgradeChallengeManagerAction: ArbOneAIPArbOS31UpgradeChallengeManagerAction | Address 0x19b715cf310c28c9020e53aaa11ce9df42e718b5 | Etherscan
- NovaAIPArbOS31UpgradeChallengeManagerAction: NovaAIPArbOS31UpgradeChallengeManagerAction | Address 0x658afc9d5ec4476fa6bb7033ea465f9901fbff27 | Etherscan
- ArbOneAIPArbOS31AddWasmCacheManagerAction: ArbOneAIPArbOS31AddWasmCacheManagerAction | Address 0xb040b105A4a0C7a9CC290164AcCBC32855368322 | Arbitrum One
- NovaAIPArbOS31AddWasmCacheManagerAction: NovaAIPArbOS31AddWasmCacheManagerAction | Address 0x61703Bf337341f2e09d96Dd6488c2907718A8E26 | Arbitrum Nova
- Arb1 SetArbOS31VersionAction: SetArbOS31VersionAction | Address 0xaF81C82Ec98f86D0017d78cD66F1026f1A5Cf1Db | Arbitrum One (CORRECTED ADDRESS)
- Nova SetArbOS31VersionAction: SetArbOS31VersionAction | Address 0x6dD43360d2a69BB9FfFC5349F2511f2A3bCbC2da | Arbiscan (CORRECTED ADDRESS)**
- ArbOneAIPArbOS31UpgradeChallengeManagerAction: ArbOneAIPArbOS31UpgradeChallengeManagerAction | Address 0x19b715cf310c28c9020e53aaa11ce9df42e718b5 | Etherscan
- NovaAIPArbOS31UpgradeChallengeManagerAction: NovaAIPArbOS31UpgradeChallengeManagerAction | Address 0x658afc9d5ec4476fa6bb7033ea465f9901fbff27 | Etherscan
- ArbOneAIPArbOS31AddWasmCacheManagerAction: ArbOneAIPArbOS31AddWasmCacheManagerAction | Address 0xb040b105A4a0C7a9CC290164AcCBC32855368322 | Arbitrum One
- NovaAIPArbOS31AddWasmCacheManagerAction: NovaAIPArbOS31AddWasmCacheManagerAction | Address 0x61703Bf337341f2e09d96Dd6488c2907718A8E26 | Arbitrum Nova
- Arb1 SetArbOS31VersionAction: SetArbOS31VersionAction | Address 0xaF81C82Ec98f86D0017d78cD66F1026f1A5Cf1Db | Arbitrum One (CORRECTED ADDRESS)
- Nova SetArbOS31VersionAction: SetArbOS31VersionAction | Address 0x6dD43360d2a69BB9FfFC5349F2511f2A3bCbC2da | Arbiscan (CORRECTED ADDRESS)**

Action contracts, deployments, and Fee Router contracts for the Fee Router AIP:

The Fee Router contracts can be found here: fund-distribution-contracts/src/FeeRouter at v1.0.1 · OffchainLabs/fund-distribution-contracts · GitHub

- Note that during the Snapshot vote, a different set of Fee Router contracts were shared. Due to optimizations that have been made since then, the above v1.0.1 Fee Router contracts will be used for this proposal. A full code diff of the two sets of Fee Router contracts can be viewed here: Comparing 61f4f60384e2ecba8250287dfb2778ce30bd82b0...v1.0.1 · OffchainLabs/fund-distribution-contracts · GitHub
- Upgrade Action Contract: governance/src/gov-action-contracts/AIPs/AIPNovaFeeRoutingAction.sol at 8e730975dbf4403a38dc0270fcc0c56bdee80c42 · ArbitrumFoundation/governance · GitHub
- Deployments:
- AIPNovaFeeRoutingAction: AIPNovaFeeRoutingAction | Address 0x849E360a247132F961c9CBE95Ba39106c72e1268 | Arbitrum Nova
- Mainnet ParentToChildRewardRouter: ParentToChildRewardRouter | Address 0x40Cd7D713D7ae463f95cE5d342Ea6E7F5cF7C999 | Etherscan
- AIPNovaFeeRoutingAction: AIPNovaFeeRoutingAction | Address 0x849E360a247132F961c9CBE95Ba39106c72e1268 | Arbitrum Nova
- Mainnet ParentToChildRewardRouter: ParentToChildRewardRouter | Address 0x40Cd7D713D7ae463f95cE5d342Ea6E7F5cF7C999 | Etherscan

Nova ArbChildToParentRewardRouter: ArbChildToParentRewardRouter | Address 0x36D0170D92F66e8949eB276C3AC4FEA64f83704d | Arbitrum Nova

Verifying the ArbOS code differences

To verify the ArbOS code differences, this [notion page

](https://arbitrumfoundation.notion.site/ArbOS-31-Bianca-Verifying-the-ArbOS-code-difference-3c1fcdde85824447b1700ac43613fdcd) contains steps to build the WASM module root on that git tag, which produces the WASM module root 0x8b104a2e80ac6165dc58b9048de12f301d70b02a0ab51396c22b4b4b802a16a4, which is what the rollup contract's wasmModuleRoot() method returns for both Arbitrum One and Arbitrum Nova.

These steps will be included on the onchain AIP on Tally in entirety to ensure consistency with previous ArbOS proposals.