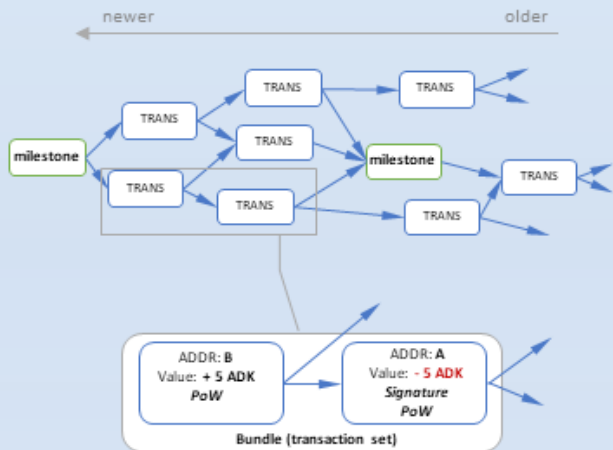
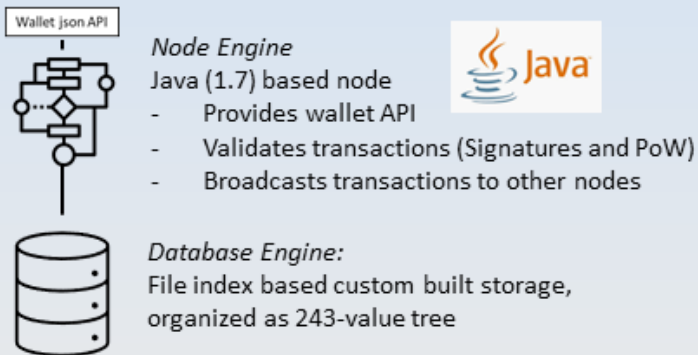


ADK v1 – Current Live Mesh

ADK (v1) Mesh Structure – OLD (Current) ADK Mesh



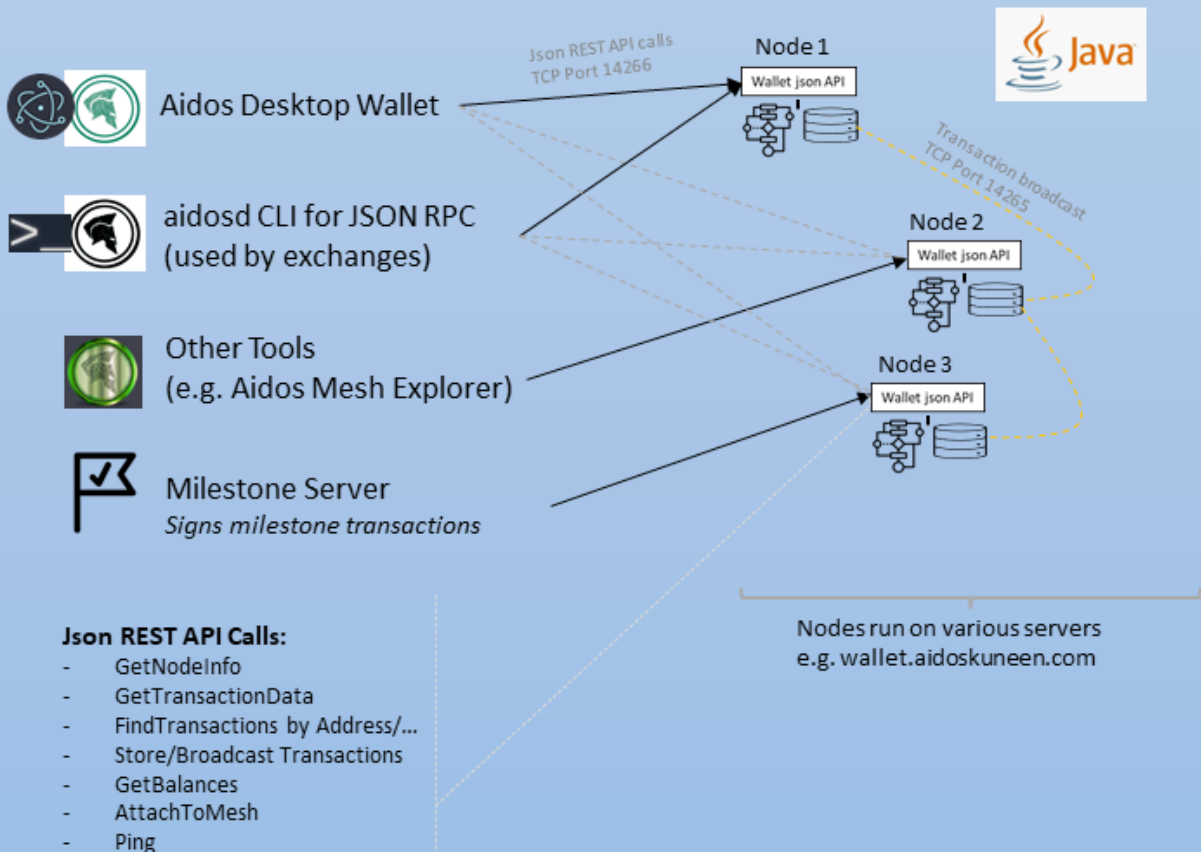
ADK (v1) Java Node



ADK Wallets / Tools

-  Aidos Desktop Wallet
-  aidosd CLI for JSON RPC (used by exchanges)
-  Other Tools (e.g. Aidos Mesh Explorer)
-  Milestone Server
Sigs milestone transactions

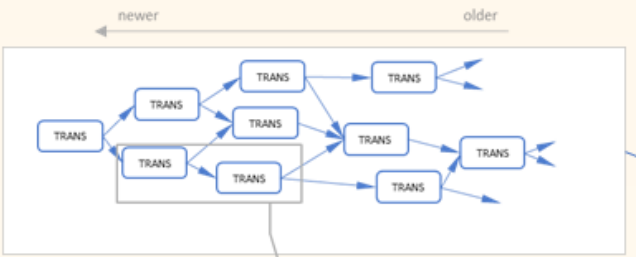
AidosKuneen ADK v1 Node Network



Json REST API Calls:

- GetNodeInfo
- GetTransactionData
- FindTransactions by Address/...
- Store/Broadcast Transactions
- GetBalances
- AttachToMesh
- Ping

ADK v2.x (2021) ADKGo SmartNode



Bundles are confirmed **directly** (near-instantly) by

- PoA Validator Nodes (v2.0) (i.e. Milestone Nodes)
- PoS Validator Nodes (v2.1) ~Q3 2022

- Aidos Desktop Wallet
- aidosd CLI for JSON RPC (used by exchanges)
- Other Tools (e.g. Aidos Mesh Explorer)
- Mobile Wallets (PoW simplified)
- Metamask
- Ledger Hardware Wallet
- Smart Contract Development Tools (Remix, solC, ...)
- Any other ERC20 / web3j compatible Wallet/Tool

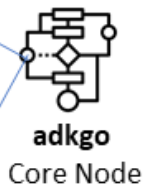
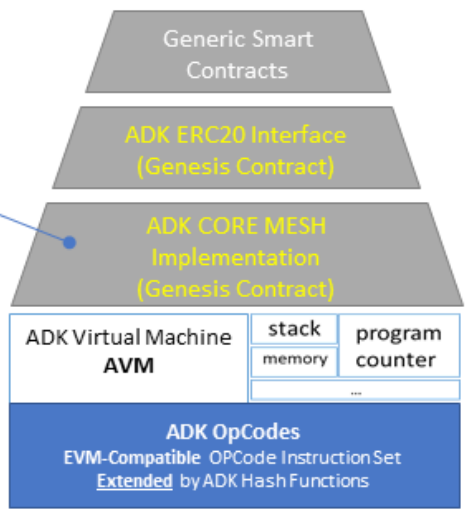
ADK JSON REST API
(ADK v1 compatible!)

- *getNodeInfo*
- *GetTransactionData*
- *FindTransactions by Address/...*
- *Store/Broadcast Transactions*
- *GetBalances*
- *Ping*

web3 compatible JSON RPC

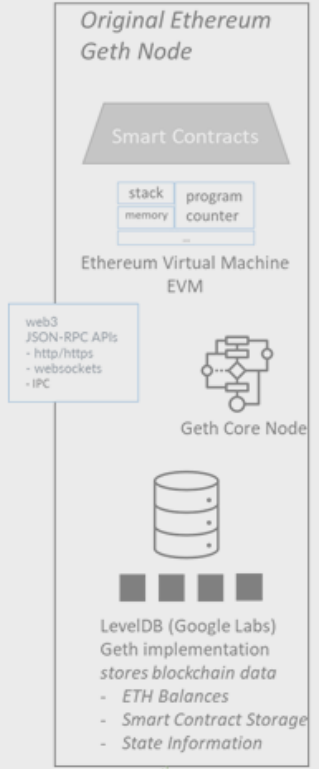
- *http/https*
- *websockets*

ADK Go SmartNode



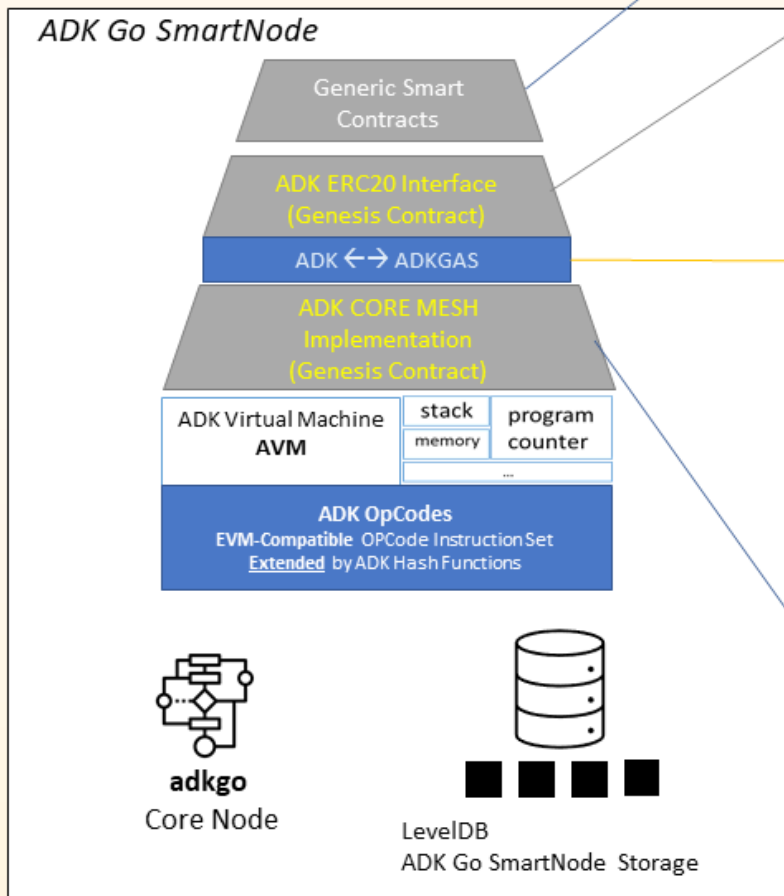
LevelDB (Google Labs)
ADK Go SmartNode Storage:
ADK Balances
ADKGAS Balances
ADK MESH State
ADK Contract States Generic

For reference, the original Go Geth Ethereum Node:



- Forked modules:
- LevelDB Storage
 - Node Fast Sync
 - EVM base
 - Web3 APIs

ADK (v2) Genesis Contracts & Smart Contracts



ADK SMART CONTRACTS

Anyone is able to deploy new custom smart contracts, and use any functions included in the AVM (= Ethereum EVM compatible + new ADK hash functions).

This includes **zkSnark precompiled contracts (e.g. ZoCrates)**

ADK ERC20 Compatible Balance Genesis Contract

ADKGAS Cost: Free for ADK PoW (Wallet) transactions and simple ERC20 balance transfer transactions

- Maintains AZ9 address and 0xHEX (reusable) ADK balances
- Provides ERC20 compatible function calls (i.e. for integration with ERC20 compatible tools like Metamask/Ledger, ...)

ADK Mesh / ADK ERC balance ↔ ADKGAS Converter Contract

What is ADKGAS:

ADKGAS is the equivalent to "ETH" on the Ethereum blockchain. ADKGAS is required to perform transactions on the ADKv2 Mesh that are not covered by the Aidos Client Wallet PoW. While (most) transactions related to the ADK GENESIS contracts will require NO gas, new custom smart contract deployments will require some ADKGAS. ADKGAS can be converted 1:1 from/to ADK at any time, and will be the incentive for future PoS node operators.

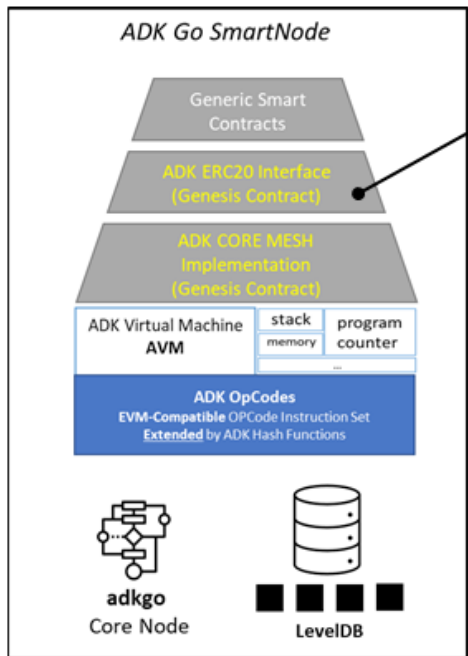
NOTE: ADKGAS can also be used in order to transact with ADK in a similar way as ETH on Ethereum (i.e. not in ERC20 token style, but directly, e.g. via non-ERC20 smart contracts or other ETH-style tools).

ADK Mesh / Transactions Genesis Contract

ADKGAS Cost: Free - 'paid' through **ADKWallet client mesh PoW**

- Implements ENTIRE ADK MESH protocol
- Validates Bundle/Transaction Client PoW (for ADKGAS free transactions)
- Checks mesh consistency
- Executes ADK balance transfers/transactions
- Manages transfer from/to AZ9 based (single use) addresses to/from 0xHEX reusable ERC20 addresses

ADK (v2) Balance Transactions OVERVIEW



ADK JSON REST API (ADK v1 compatible!)

- GetNodeInfo
- GetTransactionData
- FindTransactions by Address/...
- Store/Broadcast Transactions
- GetBalances
- Ping

web3 compatible JSON RPC

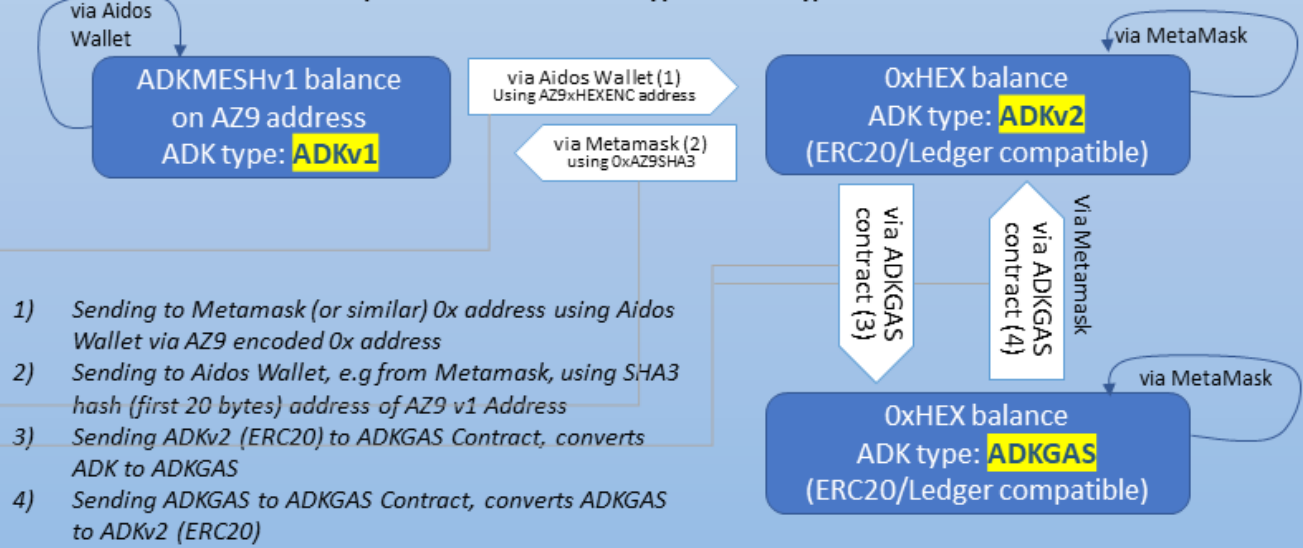
- http/https
- websockets

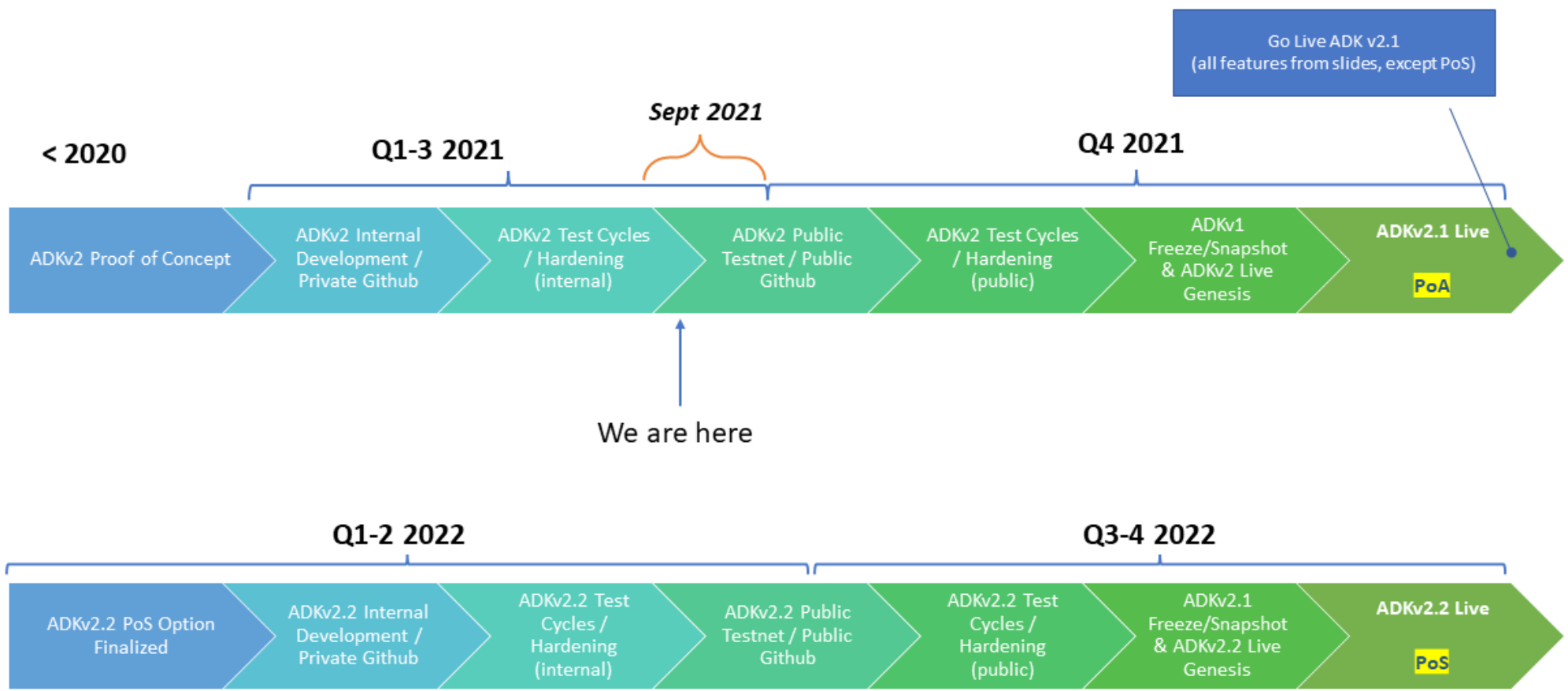
ADK Balance Holding Addresses (Balance Sheet), and available Address Types

Address	Address Type	Balance	ADK Balance nominated as
ADFHLKJYBSHGDKASDH[...]LUY9HSJK	ADKMESHv1	1234.56	ADKv1
0x23eE67eb3767fB64D5E165aF2E32F4bc08b9E0E7	0xHEX	789.12	ADKGAS
0x44C3a47eEaC0E9e7E665c6b44d00e9e4a4B15e81	0xHEX	345.6789	ADKv2
ZER0XADDRESS599KKKJAKNE[...]9999999999999999	AZ9xHEXENC	10000	
0xAA0A14ed74aef1b116d7905784E745F71579cab3	0xAZ9SHA3		
[...]			
ADAJKHFLD9SJADHLKAJSASDH[...]JUSABEVW	ADKMESHv1	1	ADKv1
0x661858F5A9e236df827Ff491847076607335A774	0xHEX		ADKv2
Total:		25,000,000	ADK Total Supply (CONSTANT)

ADKMESHv1	81 (90) char ADK Mesh address, single spend (winternitz), v1 type, Aidos Wallet compatible
0xHEX	Reusable ADK address, ERC20 (Metamask, etc) compatible, ADK & ADKGAS address
AZ9xHEXENC	0xHEX address encoded in AZ9 format (1:1 en/decodable), -> can be used in Aidos Wallet to transfer ADKMESHv1 address balance to 0xHEX address (note the "ZER0XADDRESS" prefix). Conversion tools/algorithm will be published & added to the wallet
0xAZ9SHA3	SHA3 (keccak) encoded AZ9 address (first 20 bytes) -> can be used in Metamask (or any other ERC20 compatible wallet) to transfer back to ADKMESHv1 address. Conversion tools/algorithm will be published & added to the wallet

Transfer options from one address type to other types:





FAQ

Q: Is ADKGo running on Ethereum?

A: No, while ADKGo core modules are based (originally forked) from GoGeth, it is using an extended OPCODE instruction set and new custom genesis contracts, which implement the ENTIRE ADK mesh protocol. As this includes the DAG client PoW components, any ADK transactions remain free (i.e. not cost any ADKGAS). ADKGo utilizes all the latest development that also went into Ethereum over the last years (i.e. we are not reimplementing the wheel) for some core functions such as database storage, node synchronization, web3j APIs, EVM/AVM virtual machines, and future PoS support, BUT IS AN ENTIRELY SEPARATE CHAIN with AVM DAG implementation, allowing certain transactions to remain fee-free (i.e. balance transfers).

Q: Will I have to transfer my ADK from the old ADKv1 to the new ADKv2.1? Do I need to update my wallet?

A: No, all balances will be transferred via a snapshot when v2 goes live (live-net). The new ADKGo v2.1 mesh API is fully compatible with ADKv1, so you will be able to use your existing wallets/seeds/etc. We may publish a newer faster version of the wallet, as a faster PoW will be supported, but you can still use the current (original) wallet if you prefer.

Q: Is ADKGo related to the wADK ERC20 contract on Ethereum? Is that the same?

A: NO! ADKGo will be the new NATIVE ADK mesh/chain. wADK on Ethereum is wrapped ADK on the Ethereum blockchain, so these are two different things. Currently, the transfer from/to ADK Mesh to/from wADK (on Ethereum) is not trustless, but after the move of ADKv1 to ADKGo (ADK v2 Mesh), due to ADK's new Smart Contract capabilities, the cross-chain bridge between ADKGo and Ethereum will be able to be executed in a trust-less manner (via atomic swaps and similar), especially because EVM (Ethereum) is a subset of the AVM (ADK Virtual Machine)

Q: What is PoA, and when will ADKGo transition to PoS. What is the incentive to run a future PoS node?

A: As you may know, the ADK mesh is currently validated through milestone transactions generated by the Aidos Foundation- managed milestone server. This was necessary while the decentralized DAG consensus mechanism was being worked on. ADKGo will solve the consensus conundrum through implementation of decentralized milestone nodes. Milestone node block signing privileges will be managed and allocated through the future PoS implementation, which will be based on a vote/delegation mechanism (We are also exploring alternative options such as integration into existing beacon chains, which would allow for even better cross-chain capabilities in the future. Design drafting and feasibility exploration on this option is ongoing). At this stage PoS is estimated to be going live ~Q3 2022. Between now and PoS an alternative consensus mechanism "PoA" Proof of Authority will be used, which means that milestone nodes will need to be authorized by the Aidos Foundation. The advantage over the current single milestone server is that MULTIPLE distributed milestone servers will be able to be operated (not only by AF), further strengthening the resilience and increase of speed of the ADKGo Mesh.

Q: What is ADKGAS? Is this like gas on Ethereum?

A: The main difference between ADK on ADKGo and ETH on Ethereum is that on ADKGo the PRIMARY 'currency' is ADK, implemented as part of a precompiled GENESIS CONTRACT with ERC20 interface. ADK is basically Token # 1 on the ADKv2 Mesh. ADKGAS on the other hand is what ETH is on the Ethereum blockchain. ADKGAS will be needed in order to pay gas (so yes, similar to gas on Ethereum) for custom contract deployments. BUT NOTE: unlike Ethereum, on ADKGo NO GAS IS REQUIRED for ADK balance transactions, as that is covered 100% through the DAG mesh protocol and client wallet POW. ADKGAS will only be required for smart contract deployments or other non-standard contract calls. ADKGAS will be used as the incentive for future PoS node operators. ADKGAS and ADK are 1:1 interchangeable via the ADKGAS exchange genesis contract i.e. if you have 123 ADK you can convert it from/to 123 ADKGAS (no gas required). This also allows users to transfer and transact ADKGAS in a similar way as ETH on Ethereum, and also use the corresponding smart contract features (e.g. use the 'ether' Solidity statement to transact ADKGAS, etc.). Note that ADKGAS usage will remain extremely low, unless you are planning to deploy MASSIVE smart contracts (and even then cost will be negligible compared e.g. to Ethereum gas cost, especially as standard balance transactions are always 0 adkgas).