

# AlgoPoaP



**AlgoPoaP (WIP) is the Proof of Attendance Protocol built on [Algorand](#) (AVM V1.1) aims to be elevated into a Proof Of Anything Protocol in the future with use of already in beta state proofs feature on Algorand (Proof of Anything is Parallel WIP in R&D phase using DevNET).**

## AlgoPoaP links

- [AlgoPoaP Github Repos & Documentation](#)
- [AlgoPoaP Website \(algopoap.com\)](#)
- [AlgoPoaP dApp \(algopoap.xyz\)](#)
- [Algorand NFDomain \(algopoap.algo\)](#)

## AlgoPoaP code repositories:

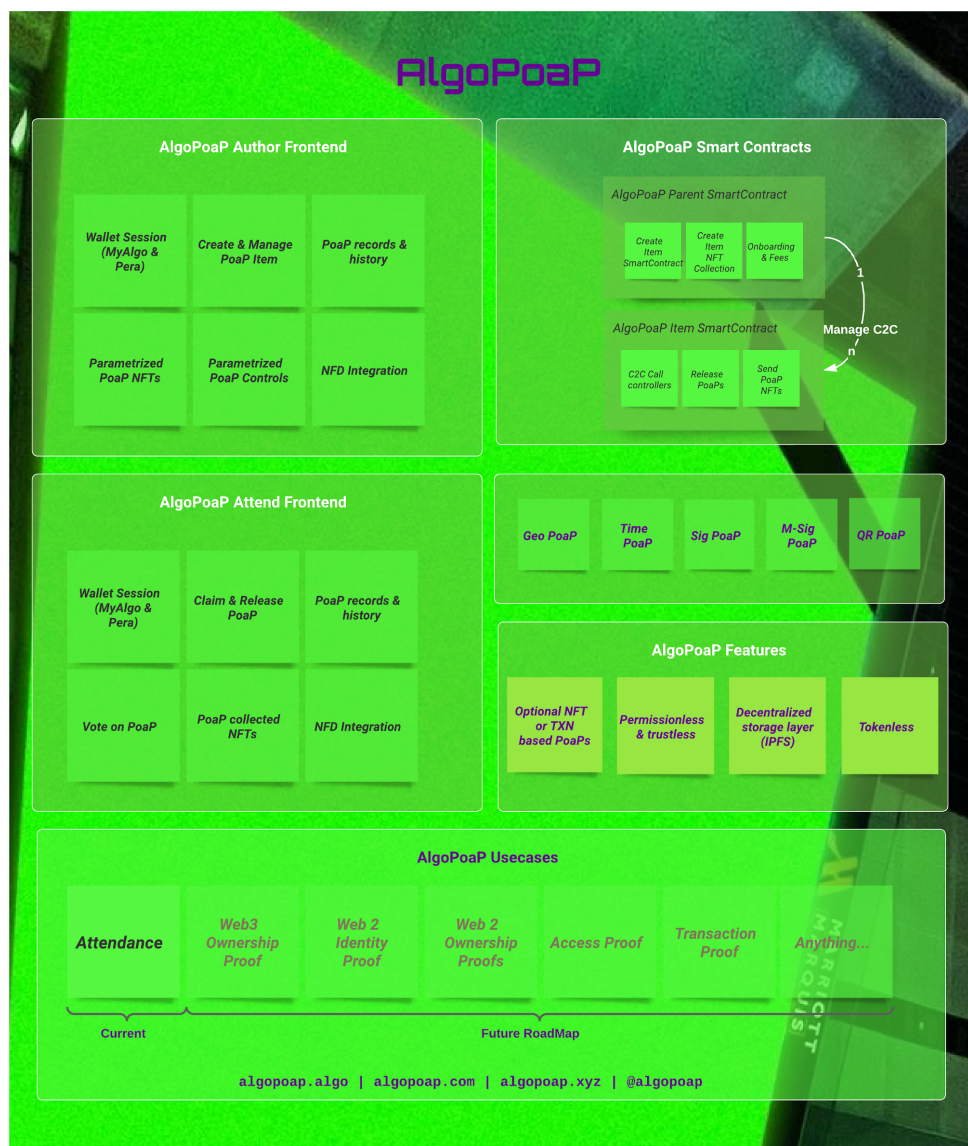
- [AlgoPoaP's Smart Contracts Repository](#)
- [AlgoPoaP's Frontend Repository](#)

# AlgoPoaP concept:

The original idea of PoaP on blockchain is developed for Ethereum ecosystem and is Token based and lacks many features. AlgoPoaP elevates, extends and expands that original idea and implements it on Algorand.

AlgoPoaP dApp is consisted of a frontend calling an Algorand ASC system in which ASCs use each other via inner transactions and C2C calls!

AlgoPoaP complies to [ARC3](#) and [ARC4](#) living standards on Algorand.



- 

## AlgoPoaP technical design:

AlgoPoaP features that are not available on ETH PoaP (it actually only supports time currently!):

- Geo constraint option (location + buffer area radius).
- Authorization Signature constraint option (Author must sign the release before PoaP issuance for claimed Attendees).
- QRCode constraint option (Attendee must scan and then call the AlgoPoaP Item ASC with it in order to claim).
- NFT based or NFT-less (Using transaction note only).

Note: AlgoPoaP does not use a single token for issuing PoaPs and is token-less , instead it uses a per AlgoPoaP item NFT generation (if desired by Author) approach.

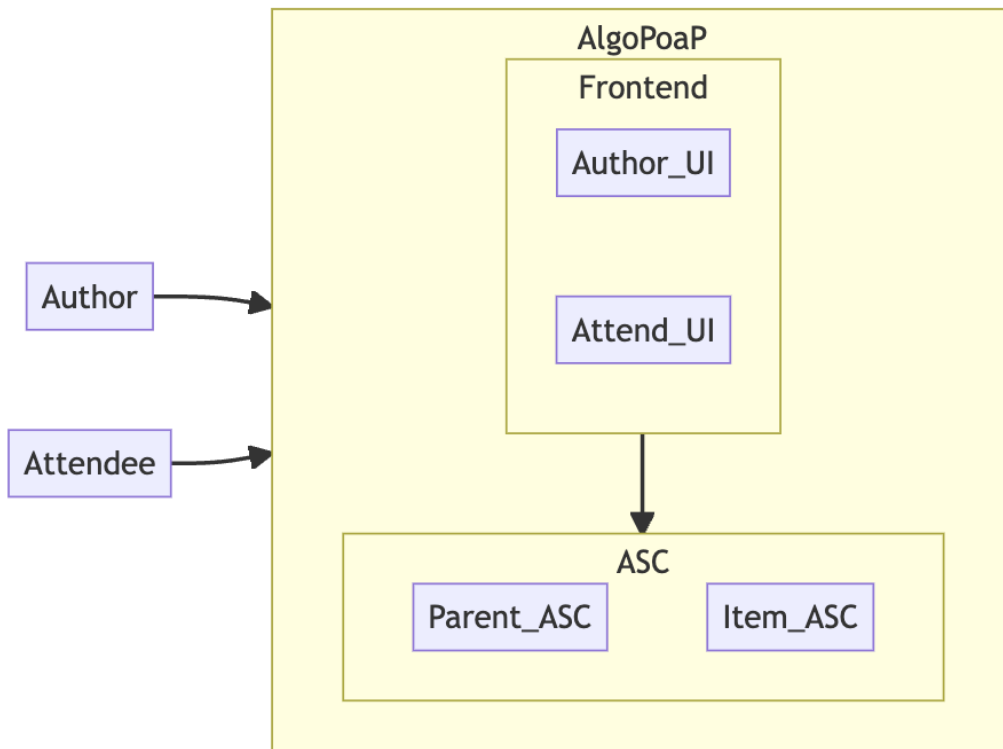
AlgoPoaP is consisted of a frontend and smart contracts on Algorand chain:

- Frontend
- Smart Contracts

AlgoPoaP frontend has 3 major functions (all in a single view for simplicity):

- Wallet Session
- Author UI
- Attend UI

Note: Frontend will be available through both cloudflare (heavily distributed on edge) and IPFS to ensure decentralization (with transparent routing for best UX).



---

## Author's Journey:

- 1- Author easily gets onboard by opting into AlgoPoaP's parent ASC.
- 2- Then can create a new PoaP venue.
- 3- Then activate the venue to let claims begin (This differs than start time option of PoaP).

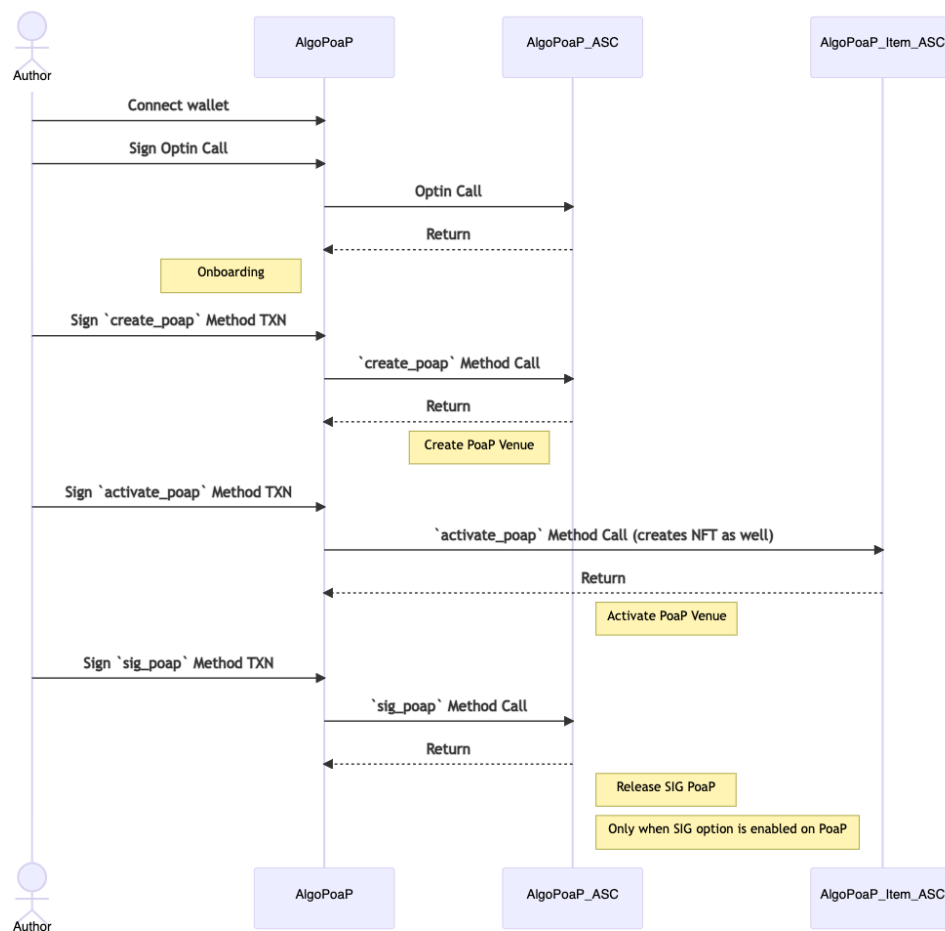
Note : If SIG is not enabled for PoaP Venue, Claim approval will send PoaP NFT or TXN to Attendee's wallet but if SIG is enabled then after signing of Author, PoaP NFT or TXN will be sent automatically to attendee's wallet, after author signs and sends a release method call transaction to release all successfully claimed AlgoPoaP attendees.

Options available for PoaP creation:

- Time (default enabled): Start time check (compared to LatestTimestamp)

- Geo: Location point check to be inside a geofence with desired radius (in meters with min of 5m and max of 1000m).
- Signature: Author's signature is needed to make PoaP claimable for every attendee, individually. Each and every attendee can receive their single claimed PoaP (in NFT or TXN depending on PoaP config) only after Author's authorization via a successful method call (which obviously should happen after both venue activation and venue start time).
- QRCode: Upon activation a secret key will be generated and included in a transaction as a method input parameter and this TXN is then communicated by a QRCode in venue location and Attendee scans this QRCode during physical presence and claims (other arguments will be added to this raw transaction object after scan and when claiming).

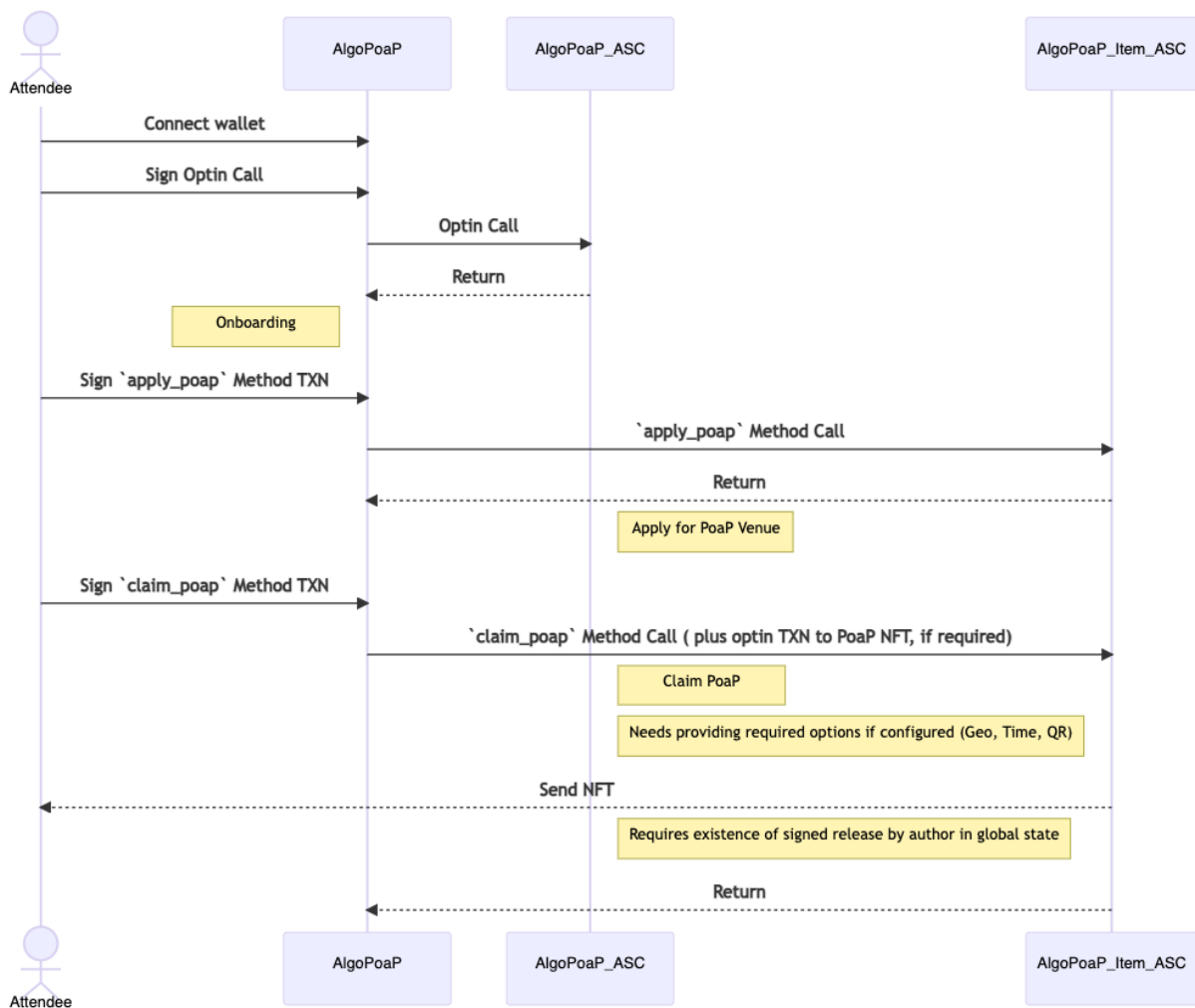
Note: QRCode feature is still under heavy re-ideation, re-design and re-everything! So please, kindly consider it WIP and FUTURE release functionality!



## Attendee's Journey:

- 1- Attendee simply gets onboard by opting into parent ASC.
- 2- Then get a searchable list of PoaP venues and applies to one by opting into it.
- 3- Then after general venue activation (by author) and by satisfying what PoaP venue options require, claim the PoaP and get PoaP NFT if approved.

Note : If SIG is not enabled for PoaP Venue, Claim approval will send PoaP NFT to Attendee's wallet but if SIG is enabled then after signing of Author, it'll be sent automatically to attendee's wallet.

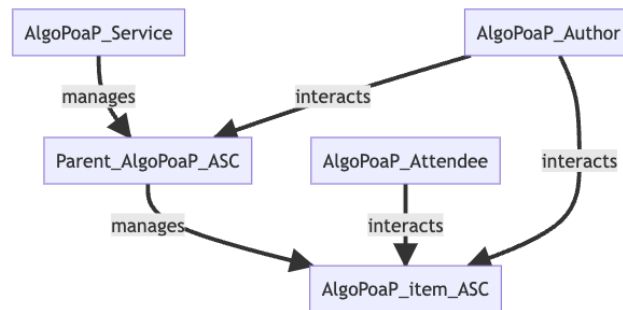


# AlgoPoaP Smart Contracts

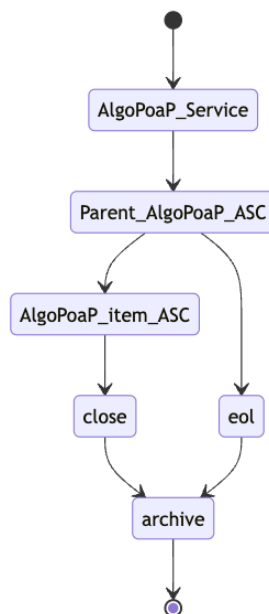
AlgoPoaP ASC System is designed on basis of newest TEAL features came with TEAL v 6.0 on AVM V1.1. AlgoPoaP Parent contract is created and thereafter every AlgoPoaP item is created by this parent contract based on configurations needed.

---

## Entities Relations:

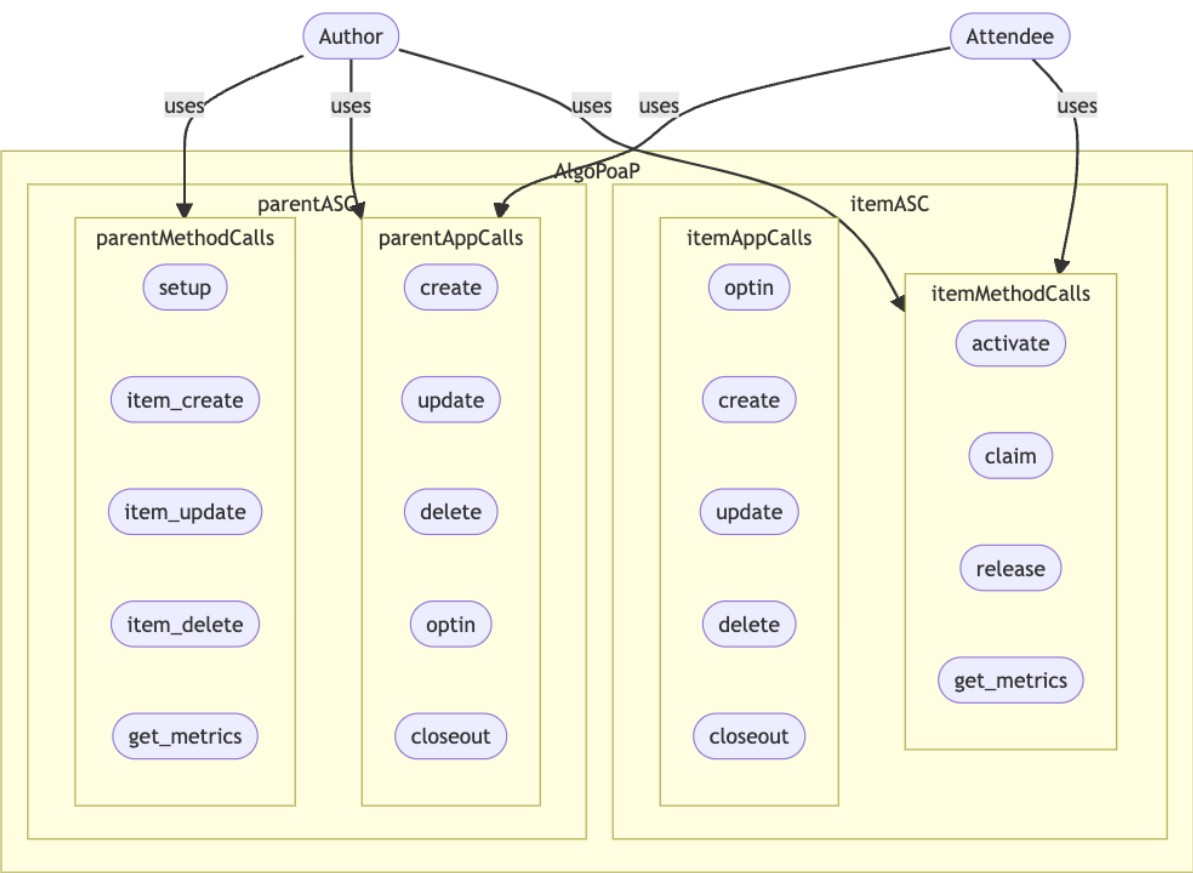


## Lifecycle:

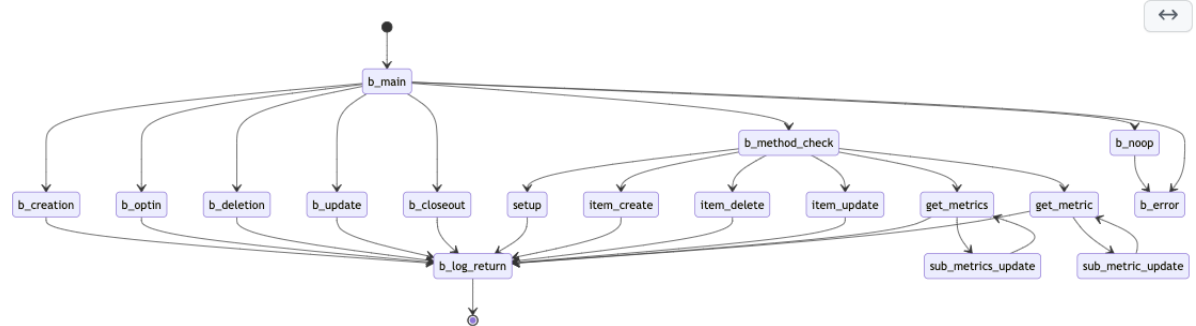


# UseCase:

Note: The NoOp calls without args will be rejected with error. This is being examined as a security practice.



# AlgoPoaP ASC TEAL Graph:





## AlgoPoaP ASC ABI :

Note 1: Data fields are global states of AlgoPoaP parent smart contract.

Note 2: Fee collection is not included anywhere at this phase of AlgoPoaP MVP development but certainly is a priority prior to public TESTNET deployment. It happens on parent smart contract.

Note 3: Author user has all metrics in localState of AlgoPoaP Item smart contract and all Authored AlgoPoaPs (upto 16 item) in localState of AlgoPoaP smart contract (parent)

AlgoPoaP_ASC
+Uint64 poap_onboard_count +Uint64 poap_count +Uint64 poap_txn_count +Uint64 poap_claim_count +Uint64 poap_issuance_count +Uint64 poap_nft_issuance_count +Uint64 poap_txn_issuance_count +Uint64 poap_geo_check_count +Uint64 poap_qr_check_count +Uint64 poap_sig_check_count +Uint64 poap_author_count +Uint64 poap_attendee_count +Byte poap_last_appid +Byte poap_last_author +Byte poap_last_attendee
+setup(string) : string +item_create(pay,byte[],byte[]) : uint64 +item_update(application,byte[],byte[]) : uint64 +item_delete(application) : void +get_metrics() : byte[] +get_metric(string) : string

## AlgoPoaP ASC ABI Schema :

```
{
  "name": "algopoap-contract",
  "desc": "AlgoPoaP Parent smart contract",
  "networks": {
    "MainNet": {
      "appID": 0
    },
    "TestNet": {
      "appID": 0
    }
  },
  "methods": [
    {
      "name": "setup",
      "args": [
        {
          "type": "string",
          "name": "version"
        }
      ],
      "returns": {
        "type": "string"
      },
      "desc": "Sets up the AlgoPoaP main contract, sets and logs the
version and returns"
    },
    {
      "name": "item_create",
      "args": [
        {
          "type": "pay",
          "name": "pay"
        },
        {
          "type": "byte[]",
          "name": "asc_approval_bytes"
        },
        {
          "type": "byte[]",
          "name": "asc_clear_bytes"
        }
      ],
      "returns": {
        "type": "uint64"
      },
      "desc": "Creates a new AlgoPoaP item smart contract and returns the
app id"
```

```

    },
    {
      "name": "item_update",
      "args": [
        {
          "type": "application",
          "name": "application"
        },
        {
          "type": "byte[]",
          "name": "asc_approval_bytes"
        },
        {
          "type": "byte[]",
          "name": "asc_clear_bytes"
        }
      ],
      "returns": {
        "type": "uint64"
      },
      "desc": "Updates an AlgoPoaP item smart contract and returns bool
(true on success)"
    },
    {
      "name": "item_delete",
      "args": [
        {
          "type": "application",
          "name": "application"
        }
      ],
      "returns": {
        "type": "void"
      },
      "desc": "Deletes an AlgoPoaP item smart contract and returns void
(approve only)"
    },
    {
      "name": "get_metric",
      "args": [
        {
          "type": "string",
          "name": "metric_signature"
        }
      ],
      "returns": {
        "type": "byte[]"
      },
      "desc": "Gets an specific metric by signature string"
    }
  ]
}

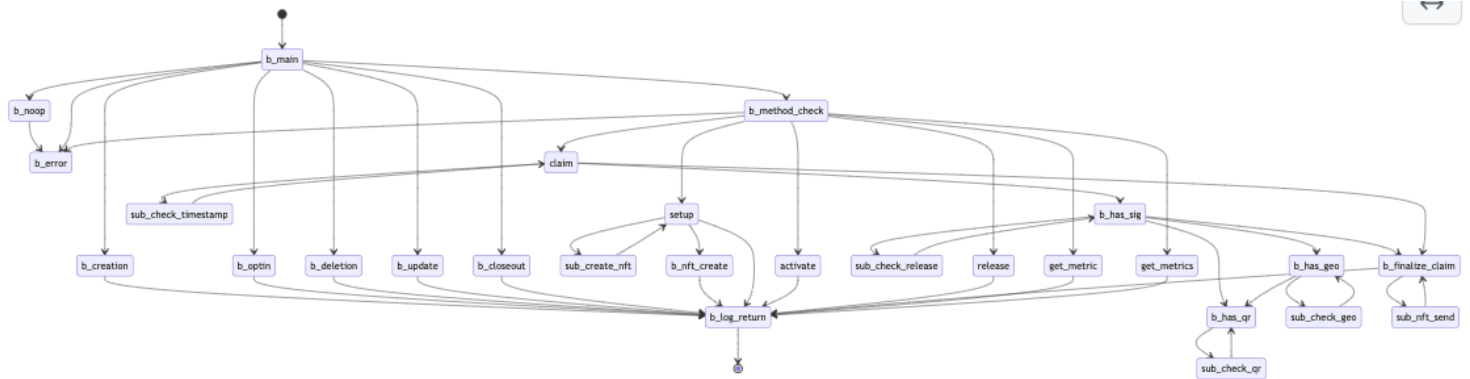
```

```

    },
    {
      "name": "get_metrics",
      "args": [],
      "returns": {
        "type": "string"
      },
      "desc": "Gets all metrics"
    }
  ]
}

```

## AlgoPoaP Item ASC TEAL Graph:



# AlgoPoaP ASC ITEM ABI :

Note 1: Data fields are global states of AlgoPoaP item smart contract.

AlgoPoaP_ASC_ITEM
<div><div>+Uint64 poap_item_attendee_count</div><div>+Uint64 poap_item_claim_count</div><div>+Uint64 poap_item_issuance_count</div><div>+Uint64 poap_item_nft_issuance_count</div><div>+Uint64 poap_item_txn_issuance_count</div><div>+Uint64 poap_item_geo_check_count</div><div>+Uint64 poap_item_qr_check_count</div><div>+Uint64 poap_item_sig_check_count</div><div>+Byte poap_item_last_attendee</div><div>+Byte poap_item_last_issuance</div><div>+Byte poap_item_last_nft_issuance</div><div>+Byte poap_item_last_txn_issuance</div><div>+Uint64 poap_lat_1</div><div>+Uint64 poap_lat_2</div><div>+Uint64 poap_lng_1</div><div>+Uint64 poap_lng_2</div><div>+Uint64 poap_geo_buffer</div><div>+Uint64 poap_start_timestamp</div><div>+Uint64 poap_end_timestamp</div><div>+Byte poap_name</div><div>+Byte poap_logo</div><div>+Byte poap_desc</div><div>+Byte poap_address</div><div>+Byte poap_url</div><div>+Byte poap_email</div><div>+Byte poap_company_name</div><div>+Byte poap_company_logo</div><div>+Byte poap_has_nft</div><div>+Byte poap_has_geo</div><div>+Byte poap_has_sig</div><div>+Byte poap_has_qrcode</div><div>+Byte poap_qr_secret</div><div>+Uint64 poap_parent_id</div><div>+Byte poap_parent_address</div><div>+Byte poap_author_address</div><div>+Uint64 poap_asa_id</div><div>+Uint64 poap_setup_time</div><div>+Uint64 poap_activate_time</div><div>+Uint64 poap_release_time</div><div>+Byte poap_is_activated</div><div>+Byte poap_is_setup</div><div>+Byte poap_is_released</div></div>
<div>+setup(pay,account,application,uint16,uint48,uint24,uint48,uint24,uint64,uint64,string,string,string,string,string,string,string,string,bool,bool,bool,bool) : string</div> <div>+activate(appl,pay,axfer) : string</div> <div>+claim(appl,pay,axfer,account,uint16,uint48,uint24,uint48,uint64,string) : string</div> <div>+release(appl) : string</div> <div>+get_metric(string) : string</div> <div>+get_metrics() : string</div>

## AlgoPoaP ASC ITEM ABI Schema :

```
{
  "name": "algopoap-item-contract",
  "desc": "AlgoPoaP Item smart contract",
  "networks": {
    "MainNet": {
      "appID": 0
    },
    "TestNet": {
      "appID": 0
    }
  },
  "methods": [
    {
      "name": "setup",
      "args": [
        {
          "type": "pay",
          "name": "pay_min_fee"
        },
        {
          "type": "account",
          "name": "author_account"
        },
        {
          "type": "application",
          "name": "parent_application"
        },
        {
          "type": "uint16",
          "name": "poap_lat_1"
        },
        {
          "type": "uint48",
          "name": "poap_lat_2"
        },
        {
          "type": "uint24",
          "name": "poap_lng_1"
        },
        {
          "type": "uint48",
          "name": "poap_lng_2"
        },
        {
          "type": "uint24",
          "name": "poap_geo_buffer"
        }
      ]
    }
  ]
}
```

```
{
  "type": "uint64",
  "name": "poap_start_timestamp"
},
{
  "type": "uint64",
  "name": "poap_end_timestamp"
},
{
  "type": "string",
  "name": "poap_name"
},
{
  "type": "string",
  "name": "poap_logo"
},
{
  "type": "string",
  "name": "poap_desc"
},
{
  "type": "string",
  "name": "poap_address"
},
{
  "type": "string",
  "name": "poap_url"
},
{
  "type": "string",
  "name": "poap_email"
},
{
  "type": "string",
  "name": "poap_company_name"
},
{
  "type": "string",
  "name": "poap_company_logo"
},
{
  "type": "bool",
  "name": "poaP_has_nft"
},
{
  "type": "bool",
  "name": "poap_has_geo"
},
{
```

```

        "type": "bool",
        "name": "poap_has_sig"
    },
    {
        "type": "bool",
        "name": "poap_has_qrcode"
    }
],
"returns": {
    "type": "string"
},
"desc": "Sets up an AlgoPoaP smart contract item"
},
{
    "name": "activate",
    "args": [
        {
            "type": "pay",
            "name": "pay_min_fees"
        },
        {
            "type": "axfer",
            "name": "optin_algopoap_nft"
        },
        {
            "type": "application",
            "name": "parent_application"
        }
    ],
    "returns": {
        "type": "byte[]"
    },
    "desc": "Activates an AlgoPoaP item smart contract and returns all
metrics"
},
{
    "name": "claim",
    "args": [
        {
            "type": "axfer",
            "name": "optin_algopoap_nft"
        },
        {
            "type": "account",
            "name": "attendee_account"
        }
    ],

```



```

        {
            "type": "application",
            "name": "parent_application"
        },
        {
            "type": "uint16",
            "name": "lat_1"
        },
        {
            "type": "uint48",
            "name": "lat_2"
        },
        {
            "type": "uint24",
            "name": "lng_1"
        },
        {
            "type": "uint48",
            "name": "lng_2"
        },
        {
            "type": "uint24",
            "name": "geo_buffer"
        },
        {
            "type": "uint64",
            "name": "timestamp"
        },
        {
            "type": "string",
            "name": "qr_secret"
        }
    ],
    "returns": {
        "type": "string"
    },
    "desc": "Claims an AlgoPoaP for an attendee and returns NFT sending
inner-transaction hash"
},
{
    "name": "release",
    "args": [
        {
            "type": "appl",
            "name": "parent_call"
        }
    ],
    "returns": {

```

```

        "type": "byte[]"
    },
    "desc": "Releases AlgoPoaP and allows all AlgoPoaP attendee's to
start claiming"
},
{
    "name": "get_metric",
    "args": [
        {
            "type": "string",
            "name": "metric_signature"
        }
    ],
    "returns": {
        "type": "string"
    },
    "desc": "Gets an specific metric by signature string"
},
{
    "name": "get_metrics",
    "args": [],
    "returns": {
        "type": "string"
    },
    "desc": "Gets all metrics"
}
]
}

```

---

Since AlgoPoaP is totally decentralized, trustless and permission-less: Every AlgoPoaP item author has full authority of the created PoaPs (AlgoPoaP-DAO is coming with dao, voting and governance features in near future, after startup formation. Preferably I will use integration to an already working service with ABI)!

The `algopoap_contract.json` contains the ABI Schema for parent AlgoPoaP contract and `algopoap_item_contract.json` is the full ABI Schema of AlgoPoaP item contract which will be created via an inner transaction.