Meta Transactions:

Gassless / Etherless / Subsidized Transacting on Ethereum

Why Ether Exists?

Halting Problem:

*“determining, from a description of an arbitrary computer program and an input, whether the program will **finish running** (i.e., halt) or continue to **run forever**.”*

- https://en.wikipedia.org/wiki/Halting_problem

Story Time


ETHBERLIN
HACKATHON / WORKSHOPS / TALKS
SEPTEMBER 7 – 9, 2018 // BERLIN. PART OF THE ETHGLOBAL COMMUNITY.

Anatomy of an Ethereum Transaction

```
{
    // Required unless deploying a contract (in which case omit)
    to: addressOrName,  // the target address or ENS name

    // These are optional/meaningless for call and estimateGas
    nonce: 0,           // the transaction nonce
    gasLimit: 0,        // the maximum gas this transaction may spend
    gasPrice: 0,        // the price (in wei) per unit of gas

    // These are always optional (but for call, data is usually specified)
    data: "0x",         // extra data for the transaction, or input for call
    value: 0,           // the amount (in wei) this transaction is sending
    chainId: 3          // the network ID; usually added by a signer
}
```

https://docs.ethers.io/ethers.js/html/api-providers.html#transaction-requests

Anatomy of an Ethereum Transaction

Hash it

```
{
    // Required unless deploying a contract (in which case omit)
    to: addressOrName,  // the target address or ENS name

    // These are optional/meaningless for call and estimateGas
    nonce: 0,           // the transaction nonce
    gasLimit: 0,        // the maximum gas this transaction may spend
    gasPrice: 0,        // the price (in wei) per unit of gas

    // These are always optional (but for call, data is usually specified)
    data: "0x",         // extra data for the transaction, or input for call
    value: 0,           // the amount (in wei) this transaction is sending
    chainId: 3          // the network ID; usually added by a signer
}
```

https://docs.ethers.io/ethers.js/html/api-providers.html#transaction-requests

tag

Anatomy of an Ethereum Transaction

```
{
    // Required unless deploying a contract (in which case omit)
    to: addressOrName,  // the target address or ENS name

    // These are optional/meaningless for call and estimateGas
    nonce: 0,           // the transaction nonce
    gasLimit: 0,        // the maximum gas this transaction may spend
    gasPrice: 0,        // the price (in wei) per unit of gas

    // These are always optional (but for call, data is usually specified)
    data: "0x",         // extra data for the transaction, or input for call
    value: 0,           // the amount (in wei) this transaction is sending
    chainId: 3          // the network ID; usually added by a signer
}
```

https://docs.ethers.io/ethers.js/html/api-providers.html#transaction-requests

Hash it

Sign it

Anatomy of an Ethereum Transaction

```
{
    // Required unless deploying a contract (in which case omit)
    to: addressOrName,  // the target address or ENS name

    // These are optional/meaningless for call and estimateGas
    nonce: 0,           // the transaction nonce
    gasLimit: 0,        // the maximum gas this transaction may spend
    gasPrice: 0,        // the price (in wei) per unit of gas

    // These are always optional (but for call, data is usually specified)
    data: "0x",         // extra data for the transaction, or input for call
    value: 0,           // the amount (in wei) this transaction is sending
    chainId: 3          // the network ID; usually added by a signer
}
```

https://docs.ethers.io/ethers.js/html/api-providers.html#transaction-requests

Hash it

Sign it

Send to network

Anatomy of an Ethereum Transaction

```
{
    // Required unless deploying a contract (in which case omit)
    to: addressOrName,  // the target address or ENS name

    // These are optional/meaningless for call and estimateGas
    nonce: 0,           // the transaction nonce
    gasLimit: 0,        // the maximum gas this transaction may spend
    gasPrice: 0,        // the price (in wei) per unit of gas

    // These are always optional (but for call, data is usually specified)
    data: "0x",         // extra data for the transaction, or input for call
    value: 0,           // the amount (in wei) this transaction is sending
    chainId: 3          // the network ID; usually added by a signer
}
```

https://docs.ethers.io/ethers.js/html/api-providers.html#transaction-requests

Hash it

Sign it

Send to network

Wait till mined

What if…

```
{
    // Required unless deploying a contract (in which case omit)
    to: addressOrName,  // the target address or ENS name

    // These are optional/meaningless for call and estimateGas
    nonce: 0,           // the transaction nonce
    gasLimit: 0,        // the maximum gas this transaction may spend
    gasPrice: 0,        // the price (in wei) per unit of gas

    // These are always optional (but for call, data is usually specified)
    data: "0x",         // extra data for the transaction, or input for call
    value: 0,           // the amount (in wei) this transaction is sending
    chainId: 3          // the network ID; usually added by a signer
}
```

hash and sign

include as **data** in new transaction

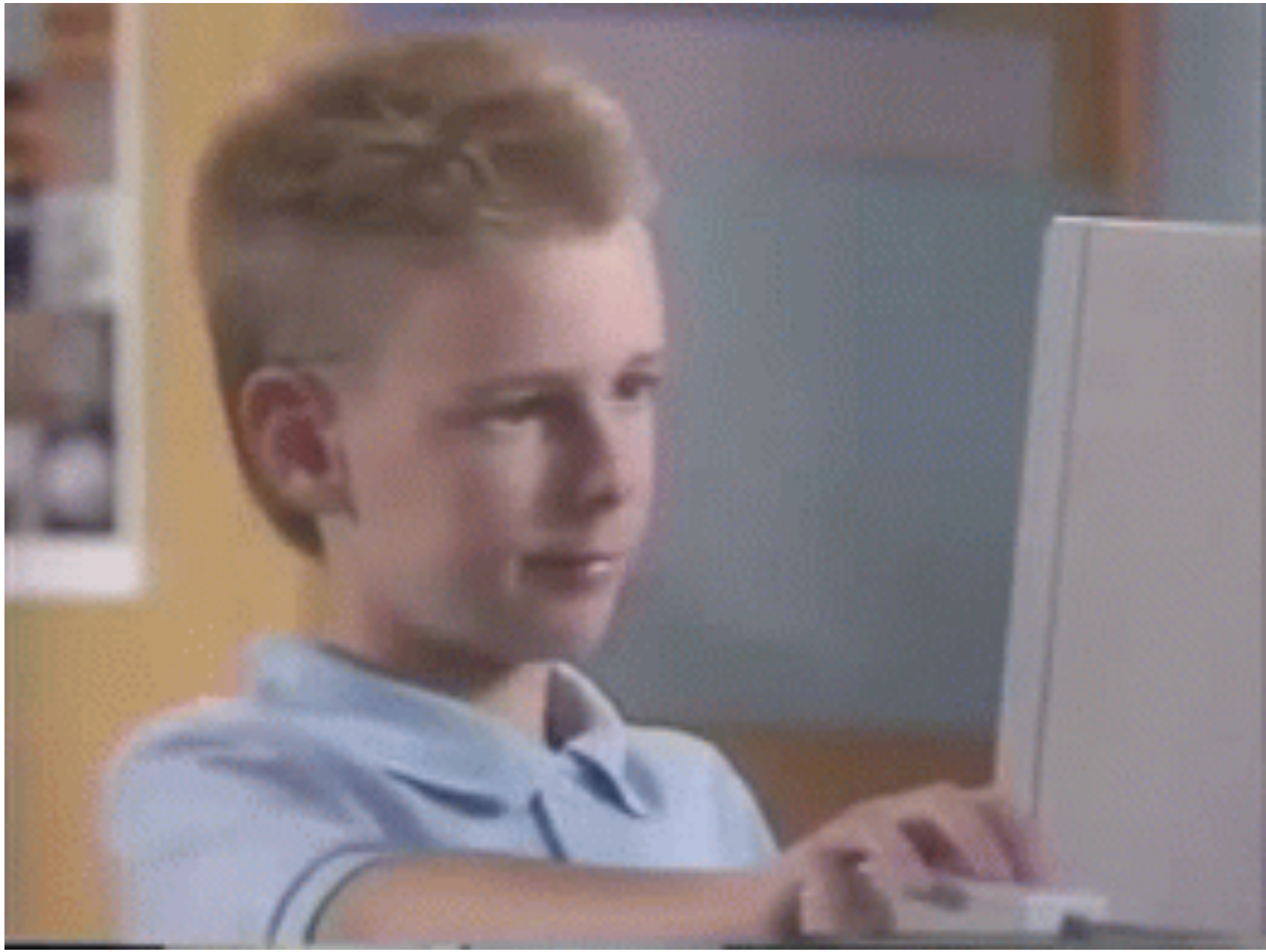

```
{
    // Required unless deploying a contract (in which case omit)
    to: addressOrName,  // the target address or ENS name

    // These are optional/meaningless for call and estimateGas
    nonce: 0,           // the transaction nonce
    gasLimit: 0,        // the maximum gas this transaction may spend
    gasPrice: 0,        // the price (in wei) per unit of gas

    // These are always optional (but for call, data is usually specified)
    data: "0x",         // extra data for the transaction, or input for call
    value: 0,           // the amount (in wei) this transaction is sending
    chainId: 3          // the network ID; usually added by a signer
}
```

What if…

Someone else foots the gas bill



Normal Tx

Sign **transaction**

Protocol verifies signature

Contract **verifies** `msg.sender`

Protocol handles replay protection

Meta Tx

Sign **data**

Contract verifies signature

Contract **agnostic** to `msg.sender`

Contract handles replay protection

Demo Time

https://github.com/NoahMarconi/metatransactions

```
mapping (address => uint256) private _nonces;
```

```
function payloadToSign(
  address sender,
  uint256 value,
  address spender,
  uint256 nonce
) public
  view
  returns (bytes32 payload)
{
  return ECDSA.toEthSignedMessageHash(
    keccak256(abi.encodePacked(
      sender,          // Token Owner.
      address(this), // Token address (replay protection).
      value,           // Number of tokens.
      spender,         // Address being approved to spend.
      nonce            // Local sender specific nonce (replay protection).
    ))
  );
}
```

```
function verifyApproval(
  address sender,
  bytes32 payload,
  bytes signature
) public
  pure
  returns (bool)
{
  address recoveredAddress = ECDSA.recover(
    ECDSA.toEthSignedMessageHash(payload),
    signature
  );

  return recoveredAddress == sender;
}
```

```
function metapprove(
  address sender,
  uint256 value,
  address spender,
  uint256 nonce,
  bytes signature
) public returns (bool success) {

  // Verify and increment nonce.
  require(getNonce(sender) == nonce);
  _nonces[sender] = _nonces[sender].add(1);

  // Verify signature.
  bytes32 payload = payloadToSign(sender, value, spender, nonce);
  require(verifyApproval(sender, payload, signature));

  // Standard approve.
  require(spender != address(0));

  _allowed[sender][spender] = value;
  emit Approval(sender, spender, value);
  return true;
}
```

To Note

Replay protection

Smart contract now responsible for work the protocol would have done for you.

Fees

Can charge a fee in ETH or in tokens.

Other Approaches

Merkle Root Redemption

See: https://blog.ricmoo.com/merkle-air-drops-e6406945584d?gi=5c1c7f05c466

One Time Address

See: https://medium.com/@weka/how-to-send-ether-to-11-440-people-187e332566b7

tag

Further Reading

https://metatx.io/

https://eips.ethereum.org/EIPS/eip-1077

https://medium.com/coinmonks/gasless-transactions-f75382095c4f

https://github.com/jpitts/eth-community-discussions/blob/master/meta-transactions.md

https://tagloyalty.com

Thank You
for you time today

We would love to hear from you

Morgan Kelly
647.402.1640
morgan@tagloyalty.com

Noah Marconi
647.669.5538
noah@tagloyalty.com