ethereum

vienna

The Road to the Frontier

Initial Code Release

Difficult to build

Outdated dependencies

Horrific UI

No documentation

No White/Yellow Paper

First Meetup

Hangout with Mihai Alisie

No one read the white paper

Vitalik @ ethereum-vienna

CEBEXPO VIENNA 2014

Ethereum Ljubljana

<PLACEHODLER/>

Simple account contract

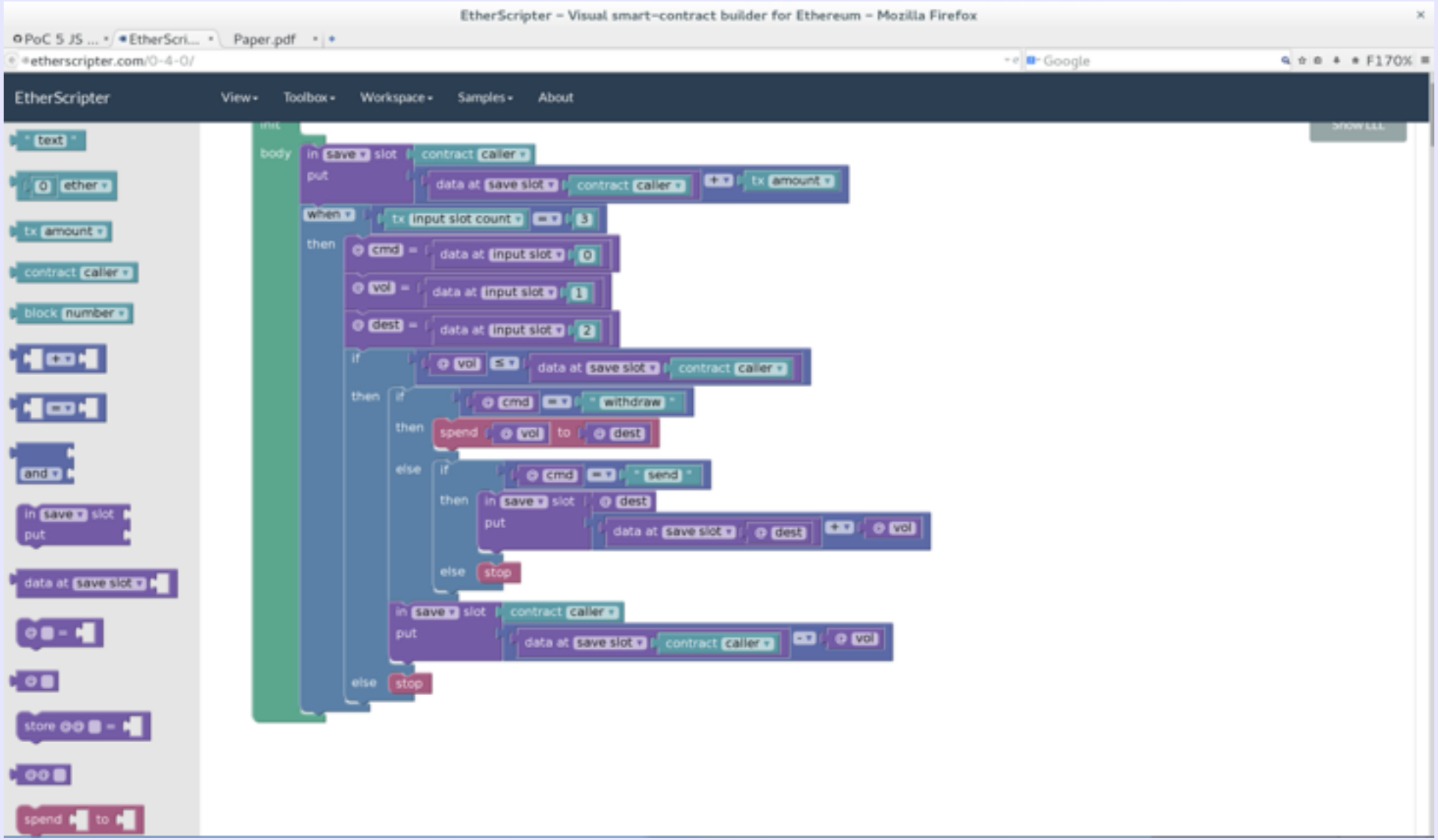
Useless

but easy to implement

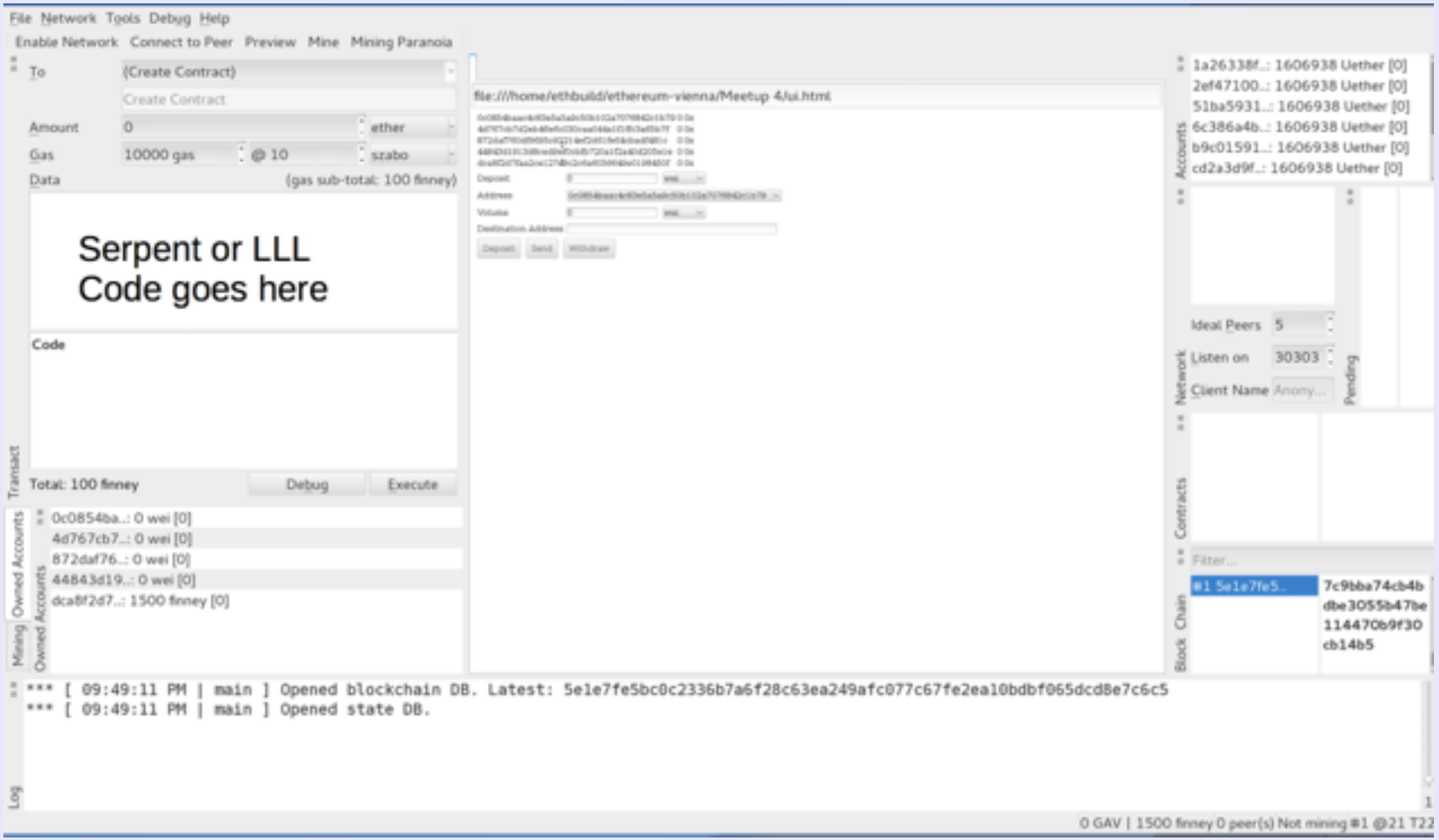
only lll working

```
{
  (return 0 (lll
    {
      (sstore (caller) (+ (sload (caller)) (callvalue)))
      (when (= (div (calldatasize) 32) 3)
       {
        [cmd]:(calldataload 0)
        [vol]:(calldataload 32)
        [dest]:(calldataload 64)
        (if (<= @vol (sload (caller)))
         {
          (if (= @cmd "withdraw")
           {
            (call (- (GAS) 100) @dest @vol 0 0 0 0)
           }
           {
            (if (= @cmd "send")
             {
              (sstore @dest (+ (sload @dest) @vol))
             }
             {
              (stop)
             }
            )
           }
          )
          (sstore (caller) (- (sload (caller)) @vol))
         }
         {
          (stop)
         }
        )
       }
      )
    }
  0))
}
```

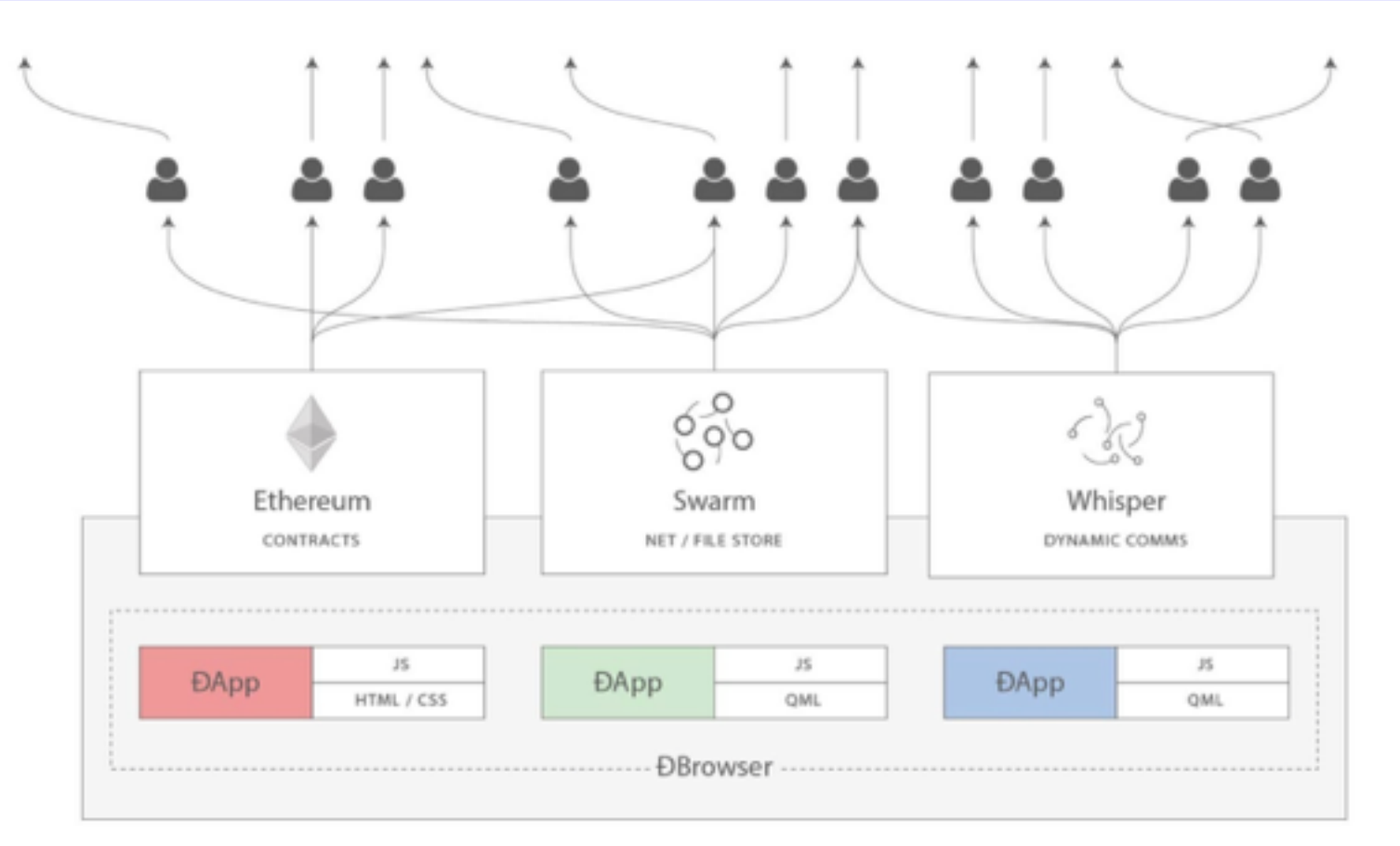
EtherScripter



First placehodler UI



The Ethereum Experience
Web 3.0



Crowdfunding

Delayed several months due to legal concerns

31.529 BTC raised (~12.5m USD at the time)

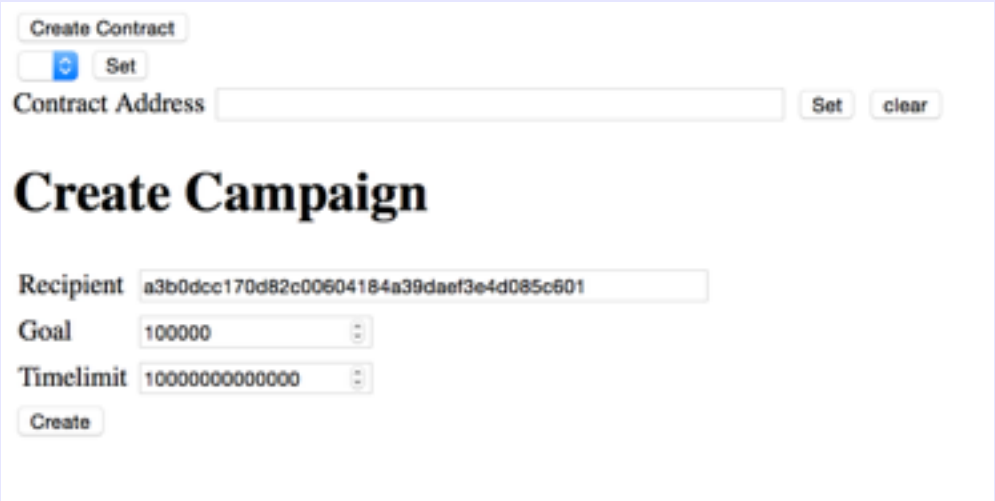
Over 9000 transactions

2nd (now 3rd) biggest crowdfunding campaign

Crowdfunding

Serpent

UI for Vitalik's contract




```
# Start a campaign, data [0, id, recipient, goal, time limit]
if msg.data[0] == 0:
    id = msg.data[1] * 2 ^ 128
    if contract.storage[id]:
        return(0)
    contract.storage[id] = msg.data[2] # Campaign recipient
    contract.storage[id + 1] = msg.data[3] # Goal
    contract.storage[id + 2] = block.timestamp + msg.data[4] # Time limit
    contract.storage[id + 3] = id + 5 # Index of next contribution
    return(msg.data[1])

# Contribute to a campaign [1, id]
elif msg.data[0] == 1:
    id = msg.data[1] * 2^128

    # Update contribution total
    total_contributed = contract.storage[id + 4] + msg.value
    contract.storage[id + 4] = total_contributed

    # Record new contribution
    sub_index = contract.storage[id + 3]
    contract.storage[sub_index] = msg.sender
    contract.storage[sub_index + 1] = msg.value
    contract.storage[id + 3] = sub_index + 2

    # Enough funding?
    if total_contributed >= contract.storage[id + 1]:
        send(contract.storage[id], total_contributed)
        v = id
        f = sub_index + 2
        while v < f:
            contract.storage[v] = 0
            v += 1
        return(1)
```

Crowdfunding

Serpent2

Functions

Data Structures

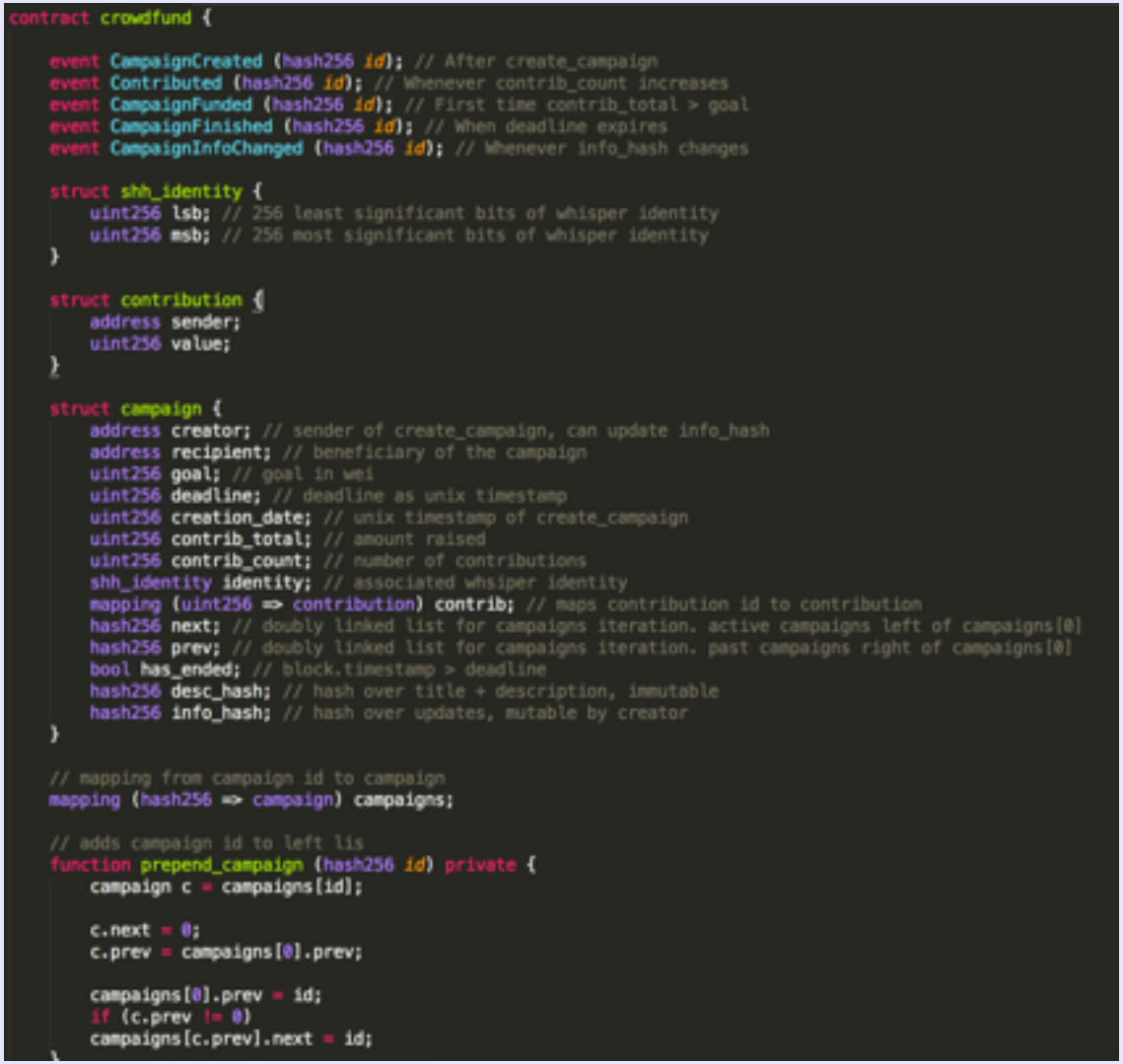
```
data campaigns[2^80](recipient, goal, deadline, contrib_total, contrib_count, contribs[2^50](sender, value))

def create_campaign(id, recipient, goal, timelimit):
    if self.campaigns[id].recipient:
        return(0)
    self.campaigns[id].recipient = recipient
    self.campaigns[id].goal = goal
    self.campaigns[id].deadline = block.timestamp + timelimit

def contribute(id):
    # Update contribution total
    total_contributed = self.campaigns[id].contrib_total + msg.value
    self.campaigns[id].contrib_total = total_contributed

    # Record new contribution
    sub_index = self.campaigns[id].contrib_count
    self.campaigns[id].contribs[sub_index].sender = msg.sender
    self.campaigns[id].contribs[sub_index].value = msg.value
    self.campaigns[id].contrib_count = sub_index + 1

    # Enough funding?
    if total_contributed >= self.campaigns[id].goal:
        send(self.campaigns[id].recipient, total_contributed)
        self.clear(id)
        return(1)

    # Expired?
    if block.timestamp > self.campaigns[id].deadline:
        i = 0
        c = self.campaigns[id].contrib_count
        while i < c:
            send(self.campaigns[id].contribs[i].sender, self.campaigns[id].contribs[i].value)
            i += 1
        self.clear(id)
        return(2)

def progress_report(id):
    return(self.campaigns[id].contrib_total)

def free_id ():
    c = 0
    while 1:
        if self.campaigns[c].recipient == 0:
            return(c)
        else
            c = c+1
```

Crowdfunding

Solidity

web3.js finally gained some stability

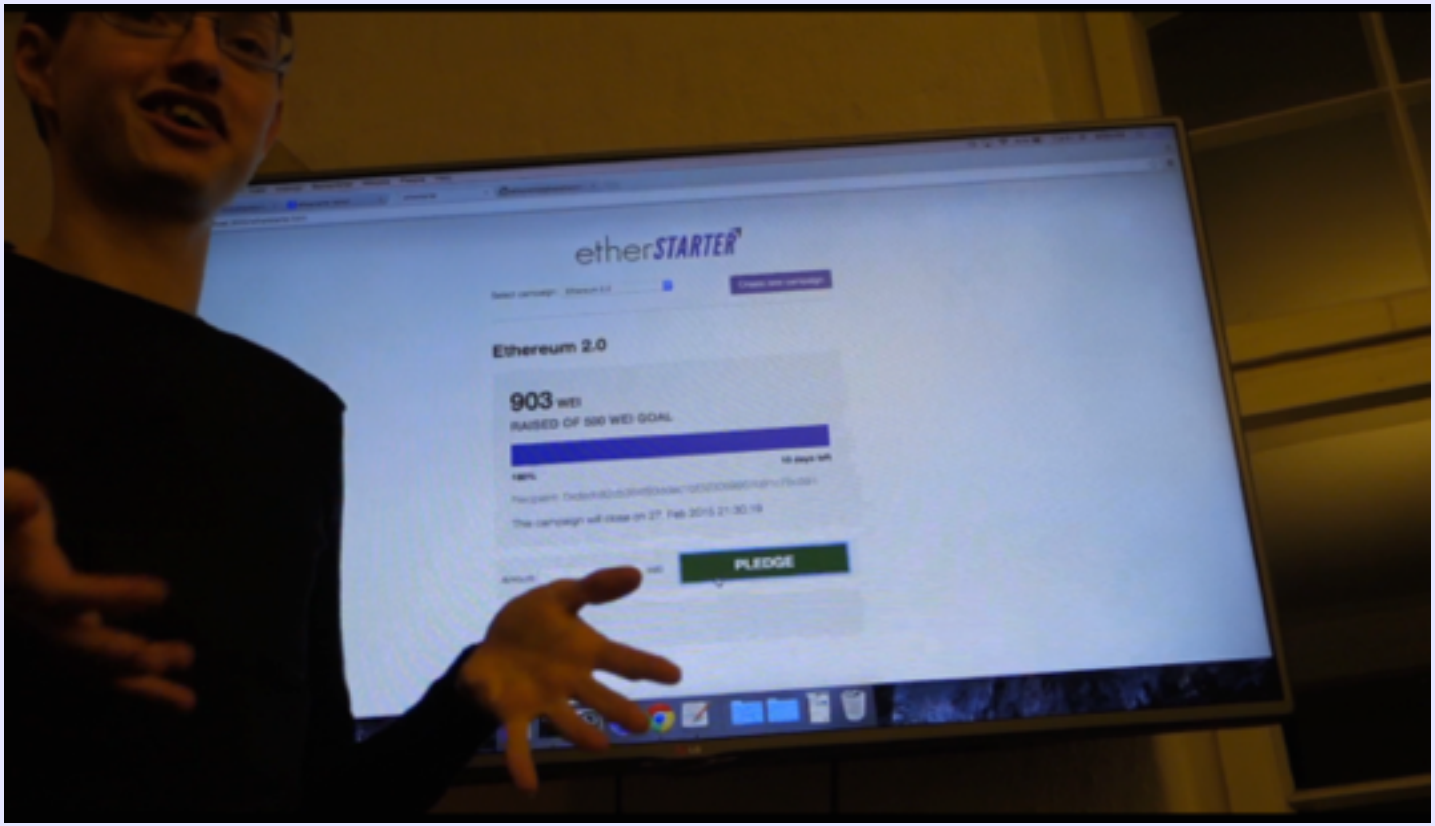
whisper was working



```
contract crowdfund {

    event CampaignCreated (hash256 id); // After create_campaign
    event Contributed (hash256 id); // Whenever contrib_count increases
    event CampaignFunded (hash256 id); // First time contrib_total > goal
    event CampaignFinished (hash256 id); // When deadline expires
    event CampaignInfoChanged (hash256 id); // Whenever info_hash changes

    struct shh_identity {
        uint256 lsb; // 256 least significant bits of whisper identity
        uint256 msb; // 256 most significant bits of whisper identity
    }

    struct contribution {
        address sender;
        uint256 value;
    }

    struct campaign {
        address creator; // sender of create_campaign, can update info_hash
        address recipient; // beneficiary of the campaign
        uint256 goal; // goal in wei
        uint256 deadline; // deadline as unix timestamp
        uint256 creation_date; // unix timestamp of create_campaign
        uint256 contrib_total; // amount raised
        uint256 contrib_count; // number of contributions
        shh_identity identity; // associated whsiper identity
        mapping (uint256 => contribution) contrib; // maps contribution id to contribution
        hash256 next; // doubly linked list for campaigns iteration. active campaigns left of campaigns[0]
        hash256 prev; // doubly linked list for campaigns iteration. past campaigns right of campaigns[0]
        bool has_ended; // block.timestamp > deadline
        hash256 desc_hash; // hash over title + description, immutable
        hash256 info_hash; // hash over updates, mutable by creator
    }

    // mapping from campaign id to campaign
    mapping (hash256 => campaign) campaigns;

    // adds campaign id to left lis
    function prepend_campaign (hash256 id) private {
        campaign c = campaigns[id];

        c.next = 0;
        c.prev = campaigns[0].prev;

        campaigns[0].prev = id;
        if (c.prev != 0)
        campaigns[c.prev].next = id;
    }
```

ETHDEV Berlin

EtherStarter @ ETHDEV Berlin



Crowdfunding

Solidity

MetaStarter

EtherStarter

Cross Contract

Interaction

```
/// @notice Create a campaign for `goal` wei for recipient `recipient`
/// @dev Create a campaign and register it with MetaStarter. Requires at Meta
/// @param recipient Recipient for the raised funds
/// @param goal Minimum value required for payout
/// @param deadline Deadline for campaign
/// @param identity_lsb 256 least significant bits of the associated whisper
/// @param identity_msb 256 most significant bits of the associated whisper i
/// @param desc_hash Hash of the description for the campaign
/// @return true if campaign was created, false if not
function create_campaign (address recipient, uint256 goal, uint256 deadline,
    if (deadline < block.timestamp) return;
    if (goal == 0) return;

    var id = metastarter.register_campaign.value(msg.value) (msg.sender, desc

    if (id != 0) {

        Campaign c = campaigns [id];

        c.recipient = recipient;
        c.goal = goal;
        c.deadline = deadline;

        metastarter.modify_status (id, CampaignStatus.STARTED);

        return true;
    }

    return false;
}
```

May 25th Hard Fork

Was inadvertently by a dev (no grand prize)

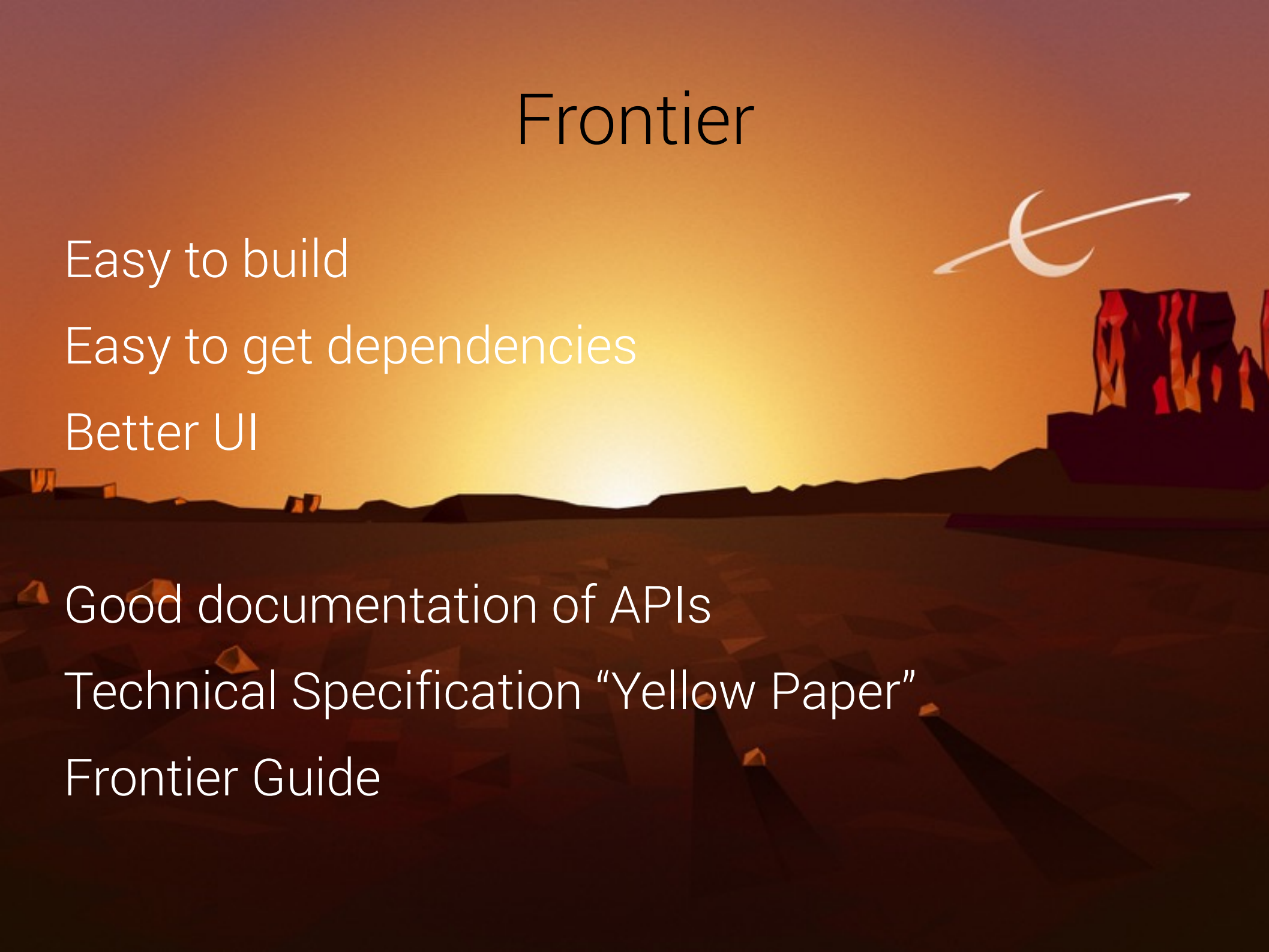
go and cpp implement CALL differently

Yellow Paper amended with a clarification

geth modified for new specification

Several issues with blockchain import discovered

Frontier



Easy to build

Easy to get dependencies

Better UI

Good documentation of APIs

Technical Specification “Yellow Paper”

Frontier Guide

Future

Meetups will be monthly again

Next one very likely during Frontier

Missing EtherStarter functionality exposed in UI

Many DApps in development

Best ones will be showcased in Meetup

Future

DAPP Name	Description	Site	Who?	Platform	Status	Last Update
Cosmo	Meteor dapp for building and vetting solid	https://github.com/SilentCicero/meteor-dapp-cosmo	Nick Dodson	Ethereum POC9	7. Working Prototype	4/23/2015
MintChalk	In-browser smart contract building / publis	http://www.mintchalk.com/	James Alexander Levy	Serpent 1.0	7. Working Prototype	4/23/2015
EtherEx	Decentralized Exchange	http://etherex.org/	caktux	Ethereum POC9	7. Working Prototype	4/22/2015
sleth	Slot Machine	https://github.com/jorisbontje/sleth	Joris Bontje	Ethereum POC9	7. Working Prototype	4/22/2015
btcrelay	Bitcoin Blockchain Relay	https://github.com/ethers/dapp-bin/tree/btcrelay/btcrela	Joseph Chow	Ethereum POC9	7. Working Prototype	4/22/2015
Ethergit	Blockchain explorer	http://dev.ethergit.com/	ConsenSys / Roman Mand	Ethereum	7. Working Prototype	4/22/2015
WeiFund	Crowdfunding Platform	http://weifund.io/	Nick Dodson	Ethereum POC9	7. Working Prototype	4/22/2015
dapp pricefeed	(Gold) price feed	https://github.com/SilentCicero/meteor-dapp-pricefeed	Nick Dodson	Ethereum POC9	7. Working Prototype	4/22/2015
Spritzle	Fractional investment platform for Ethere	https://github.com/psalami/spritzle	Patrick Salami	Ethereum POC9 ?	6. Demo	4/23/2015
Bit Vote	Voting with time on the blockchain	http://bitvote.github.io	Aaron Bale	Ethereum ?	6. Demo	4/23/2015
Adept	IBM/Samsung IoT Project	https://www.youtube.com/watch?v=U1XQPlqyP7A	John Cohn	Ethereum POC5?	6. Demo	4/23/2015
TrustDavis	Reputation system	https://github.com/BlockchainSociety/TrustDavis	Joris Bontje & Jarrad Hope	Web demo	6. Demo	4/22/2015
Augur	Decentralized Prediction Market	http://www.augur.net/	Jack Peterson, Joey Krug,	Ethereum POC9	6. Demo	4/22/2015
Project Groundhog	Social Network	https://www.youtube.com/watch?v=WFeJYv3PSaI	Conrad Bars	Ethereum ?	6. Demo	4/22/2015
Whisper Chat Client	Group chat	https://github.com/ethereum/meteor-dapp-whisper-chat	Fabian Vogelsteller	Ethereum Metropolis	6. Demo	4/22/2015
Dapp Catalog	Dapp Catalog	https://github.com/ethereum/meteor-dapp-catalog	Fabian Vogelsteller, Alex va	Ethereum Metropolis	6. Demo	4/22/2015
Wallet Dapp	Ethereum Wallet	https://github.com/ethereum/meteor-dapp-wallet	Fabian Vogelsteller, Alex va	Ethereum Metropolis	6. Demo	4/22/2015
cryptocoinwatch	Crypto currency datafeed	https://github.com/EtherCasts/cryptocoinwatch	Joris Bontje	Ethereum POC7	6. Demo	4/22/2015
Ethereum Prediction Market	Prediction market	http://atomrigs.blogspot.com	Atomrigs	Ethereum POC8	6. Demo	4/22/2015
Serpent.py	A Python Implementation of the Serpent F	https://github.com/BlockchainSociety/serpent.py	Jarrad Hope	Ethereum POC9	5. Work In Progress	4/28/2015
Swarm	Distributed File Storage	https://github.com/ethersphere/go-ethereum/tree/bzz	Daniel Nagy	Ethereum Metropolis	5. Work In Progress	4/28/2015
EtherMarket	Decentralized Marketplace	https://github.com/ethermarket/ethermarket	Iuri Matias / Ryan Casey	Ethereum POC9	5. Work In Progress	4/28/2015
atomic-swap	Atomic cross-chain trading	https://github.com/zack-bitcoin/ethereum-atomic-swap	Zack Hess	Ethereum / Bitcoin	5. Work In Progress	4/22/2015
TrustlessPrivacy	Interoperable electronic health records	http://www.trustlessprivacy.com	sam@trustlessprivacy.com	Solidity	5. Work In Progress	5/27/2015