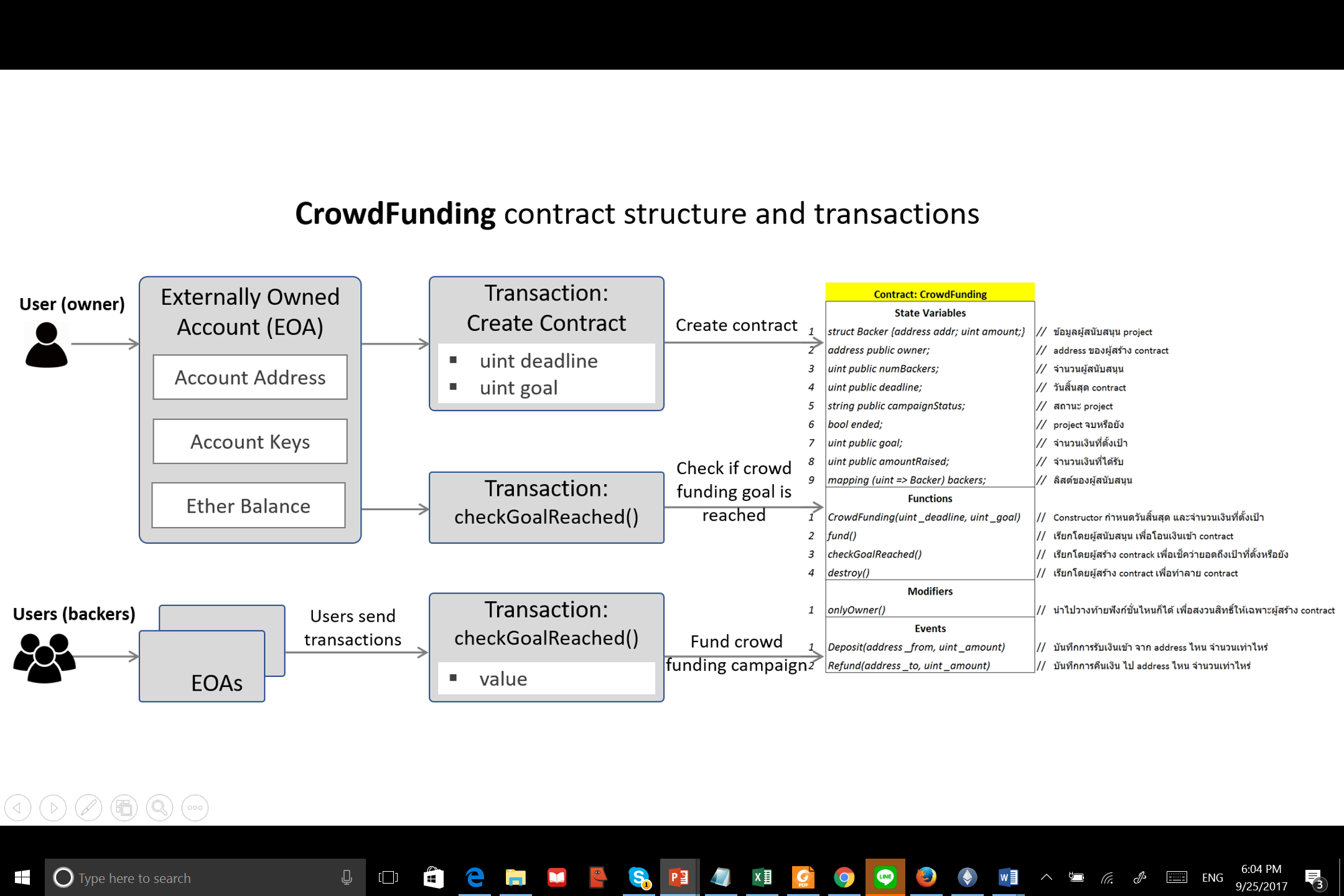
Crowdfunding Contract

This is an example of a smart contract to setup a crowdfunding campaign. With this contract, a crowdfunding campaign can be setup by specifying a campaign goal and deadline. Backers interested in supporting the funding campaign can send transactions to the contract along with the value (in Ether) to contribute to the campaign. The funds raised are held in the contract account. When the campaign ends, the owner of the campaign can check if the campaign goal has been reached. If the campaign succeeds, the funds are releases to the campaign beneficiary (or the owner of the contract). However, if the campaign fails, the backers are issued refunds.



// CrowdFunding contract implemented in Solidity

pragma solidity ^0.4.11;

contract CrowdFunding {

struct Backer {

address addr;

uint amount;

}

address public owner;

uint public numBackers;

uint public deadline;

string public campaignStatus;

bool ended;

uint public goal;

uint public amountRaised;

mapping (uint => Backer) backers;

event Deposit(address \_from, uint \_amount);

event Refund(address \_to, uint \_amount);

modifier onlyOwner {

if (msg.sender != owner) {

throw;

} else {

\_;

}

}

function CrowdFunding(uint \_deadline, uint \_goal) {

owner = msg.sender;

deadline = \_deadline;

goal = \_goal;

campaignStatus = "Funding";

numBackers = 0;

amountRaised = 0;

ended = false;

}

function fund() payable {

Backer b = backers[numBackers++];

b.addr = msg.sender;

b.amount = msg.value;

amountRaised += b.amount;

Deposit(msg.sender, msg.value);

}

function checkGoalReached () onlyOwner returns (bool ended) {

if (ended)

throw; //this function already been called

if(block.timestamp < deadline)

throw;

if(amountRaised >= goal) {

campaignStatus = "Campaign Succeeded";

ended = true;

if (!owner.send(this.balance))

throw; // if anything fails,

// this will revert the changes above

} else {

uint i = 0;

campaignStatus = "Campaign Failed";

ended = true;

while (i <= numBackers) {

backers[i].amount = 0;

if (!backers[i].addr.send(backers[i].amount))

throw;

Refund(backers[i].addr, backers[i].amount);

i++;

}

}

}

function kill() onlyOwner{

suicide(owner);

}