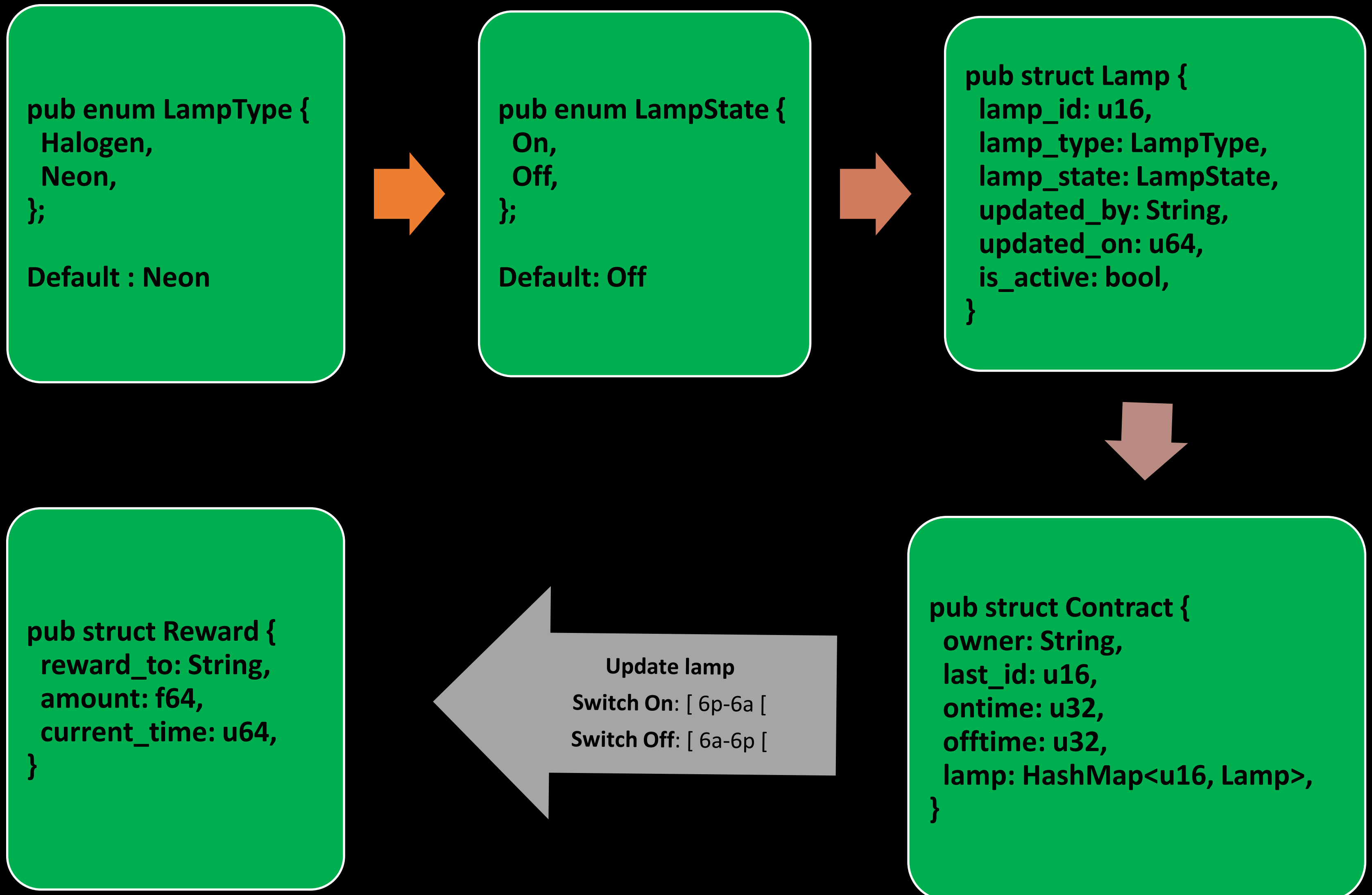
Streetlights are public asset; it would be good if public can take its maintenance also in their hands ...

NCD : Lamps

try.vsb@gmail.com

Contract elements



```
pub enum LampType {
  Halogen,
  Neon,
};

Default : Neon
```

```
pub enum LampState {
  On,
  Off,
};

Default: Off
```

```
pub struct Lamp {
  lamp_id: u16,
  lamp_type: LampType,
  lamp_state: LampState,
  updated_by: String,
  updated_on: u64,
  is_active: bool,
}
```

```
pub struct Contract {
  owner: String,
  last_id: u16,
  ontime: u32,
  offtime: u32,
  lamp: HashMap<u16, Lamp>,
}
```

Update lamp

Switch On: [6p-6a [

Switch Off: [6a-6p [

```
pub struct Reward {
  reward_to: String,
  amount: f64,
  current_time: u64,
}
```

Contract methods (1)

1. pub fn ***new_lamp***(&mut self, state: String, ltype: String) -> u16 { … } : Called by **Owner**. Persists a new instance of lamp.

2. pub fn ***gen_lamp***(&self, state: String, ltype: String) -> (u16, Lamp) { .. } : It's a **private** method called by 'new_lamp' method, it generates an instance of new lamp and returns to the caller.

3. pub fn ***enable_lamp***(&mut self, lamp_id: u16) { … } : Called by **Owner** to disable a lamp.

4. pub fn ***disable_lamp***(&mut self, lamp_id: u16) { … } : Called by **Owner** to enable a lamp.

5. pub fn ***change_owner***(&mut self, new_owner: String) -> bool { … }: Called by **Owner** to transfer the ownership to new account.

6. pub fn ***get_owner***(&self) -> &str { … } : To retrieve current owner

7. pub fn ***get_count***(&self) -> u16 { … } : Count of # of lamps present in contract

8. pub fn ***get_lamp***(&self, lamp_id: u16) -> Option<&Lamp> { … } : To fetch details for a single Lamp using Lamp id

9. pub fn ***get_all_lamps***(&self) -> &HashMap<u16, Lamp> { .. } : To get the list of all the lamps

Contract methods (2)

10. pub fn ***can_set_lamp_state***(&self, new_state: String) -> bool { … } : This method calls the “***get_epoc_hour”*** function” to get current hour of the days, then it checks if the ‘new_state’ parameter (On or Off operation) can be done at this time of day

11. pub fn ***set_lamp_state***(&mut self, lamp_id: u16, new_state: String) -> bool { … } : This method is used to set the state of the lamp, it calls “***can_set_lamp_state”*** and if the outcome is ‘true’ then it updates the state of the lamp. It also updates **‘updated_by’ and ‘updated_on’** fields, these two fields are used to calculate rewards.

12. pub fn ***calculate_reward*(&self, reward_for: String) -> Option<HashMap<u16, Reward>> { … } :** This method is used to calculate rewards for the user who set the lamps state.

Possible value for **‘reward_for’** are

‘Self’ : Check reward for the caller only

‘All’ : Check reward for all users. Only owner of the contract can call this method with ‘All’ parameter

Near CLI Contract calls

Create new lamp

```
near call one.x0x.testnet new_lamp '{\"ltype\": \"Neon\", \"state\": \"Off\"}' –accountId one.x0x.testnet
```

Get all lamp

```
near view one.x0x.testnet get_all_lamps
```

Get lamp

```
near view one.x0x.testnet get_lamp '{\"lamp_id\": 2}’
```

Get Owner

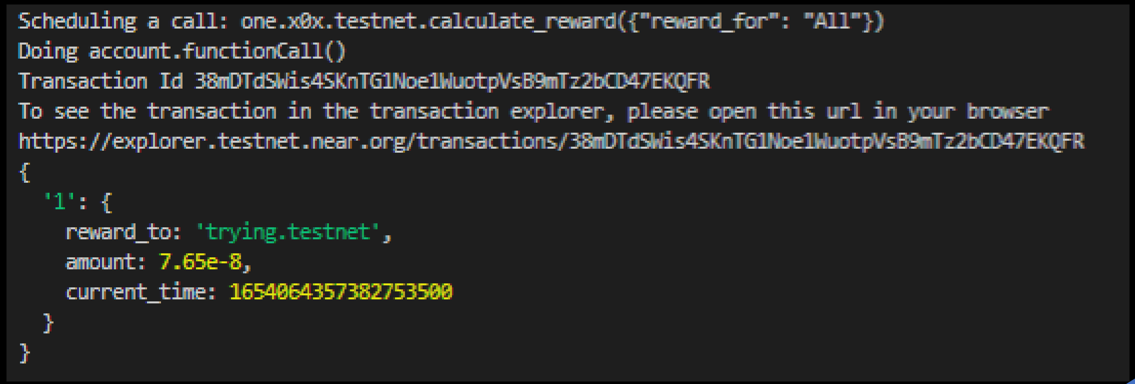
```
near view one.x0x.testnet get_owner
```

can_set_lamp_state

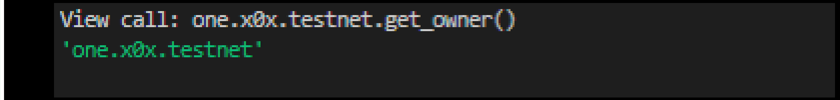
```
near view one.x0x.testnet can_set_lamp_state  '{\"ned_for\": \"All\"}' --accountId one.x0x.testnetw_state\": \"On\"}’
```

Calculate reward

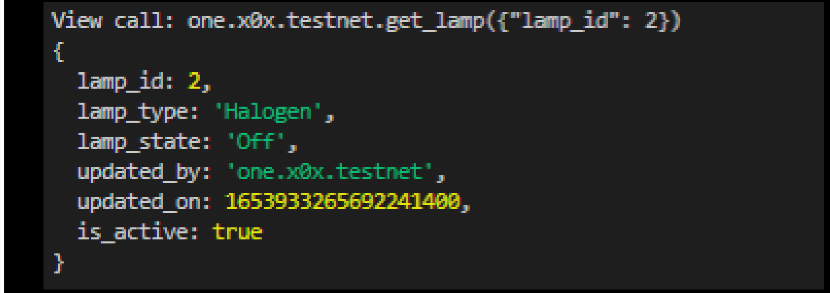
```
near call one.x0x.testnet calculate_reward '{\"reward_for\": \"All\"}'  --accountId one.x0x.testnet
```



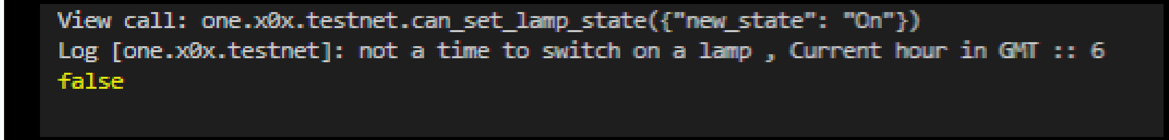
```
Scheduling a call: one.x0x.testnet.calculate_reward({"reward_for": "All"})
Doing account.functionCall()
Transaction Id 38mDTdSWis4SKnTG1Noe1WuotpVsB9mTz2bCD47EKQFR
To see the transaction in the transaction explorer, please open this url in your browser
https://explorer.testnet.near.org/transactions/38mDTdSWis4SKnTG1Noe1WuotpVsB9mTz2bCD47EKQFR
{
  '1': {
    reward_to: 'trying.testnet',
    amount: 7.65e-8,
    current_time: 1654064357382753500
  }
}
```



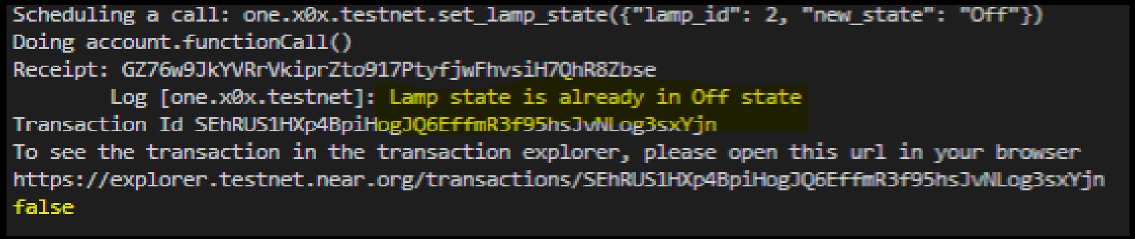
```
View call: one.x0x.testnet.get_owner()
'one.x0x.testnet'
```



```
View call: one.x0x.testnet.get_lamp({"lamp_id": 2})
{
  lamp_id: 2,
  lamp_type: 'Halogen',
  lamp_state: 'Off',
  updated_by: 'one.x0x.testnet',
  updated_on: 1653933265692241400,
  is_active: true
}
```



```
View call: one.x0x.testnet.can_set_lamp_state({"new_state": "On"})
Log [one.x0x.testnet]: not a time to switch on a lamp , Current hour in GMT :: 6
false
```



```
Scheduling a call: one.x0x.testnet.set_lamp_state({"lamp_id": 2, "new_state": "Off"})
Doing account.functionCall()
Receipt: GZ76w9JkYVRrVkiprZto917PtyfjwFhvsiH7QhR8Zbse
        Log [one.x0x.testnet]: Lamp state is already in Off state
Transaction Id SEhRUS1HXp4BpiHogJQ6EffmR3f95hsJvNLog3sxYjn
To see the transaction in the transaction explorer, please open this url in your browser
https://explorer.testnet.near.org/transactions/SEhRUS1HXp4BpiHogJQ6EffmR3f95hsJvNLog3sxYjn
false
```

Stats Gallery :
https://stats.gallery/testnet/one.x0x.testnet/contract

Once more time,

I would like to thank you all for those two wonderful sessions

NEAR Certified Developer - May 20, 2022 @ 2p-4p UTC (2 hrs)

NEAR Certified Developer - Mar 21, 2022 @ 9a-11a UTC (2 hrs)