



ITMO UNIVERSITY

BAL — modelling framework for Blockchain and Quantum Network"

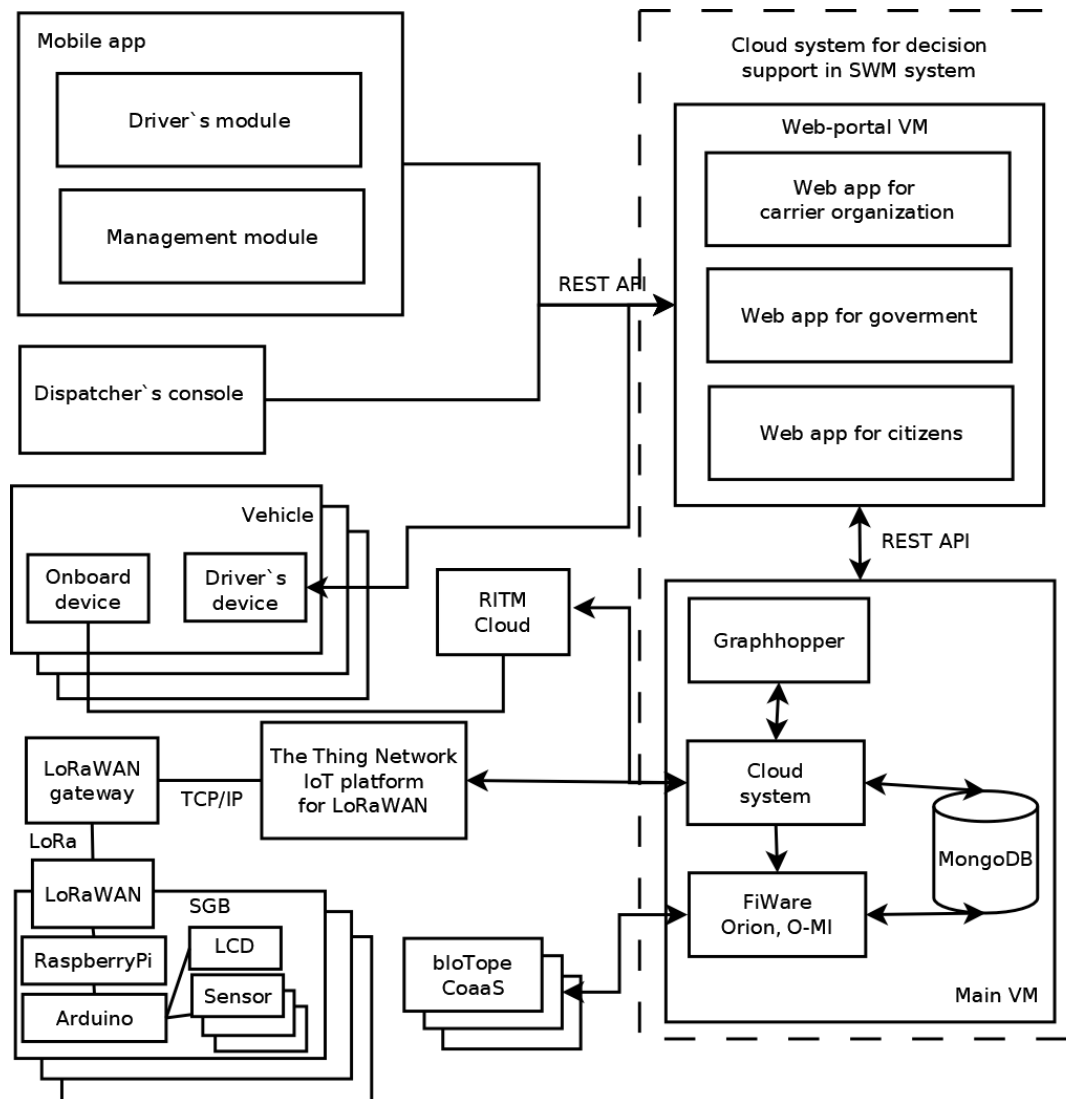
sadov@mail.ifmo.ru

Oleg Sadov

<http://sdn.ifmo.ru/>



bloTope SWM general architecture



Main components:

- Cloud system
- Web apps
- Mobile apps
- Onboard devices
- SGB

Communication via:

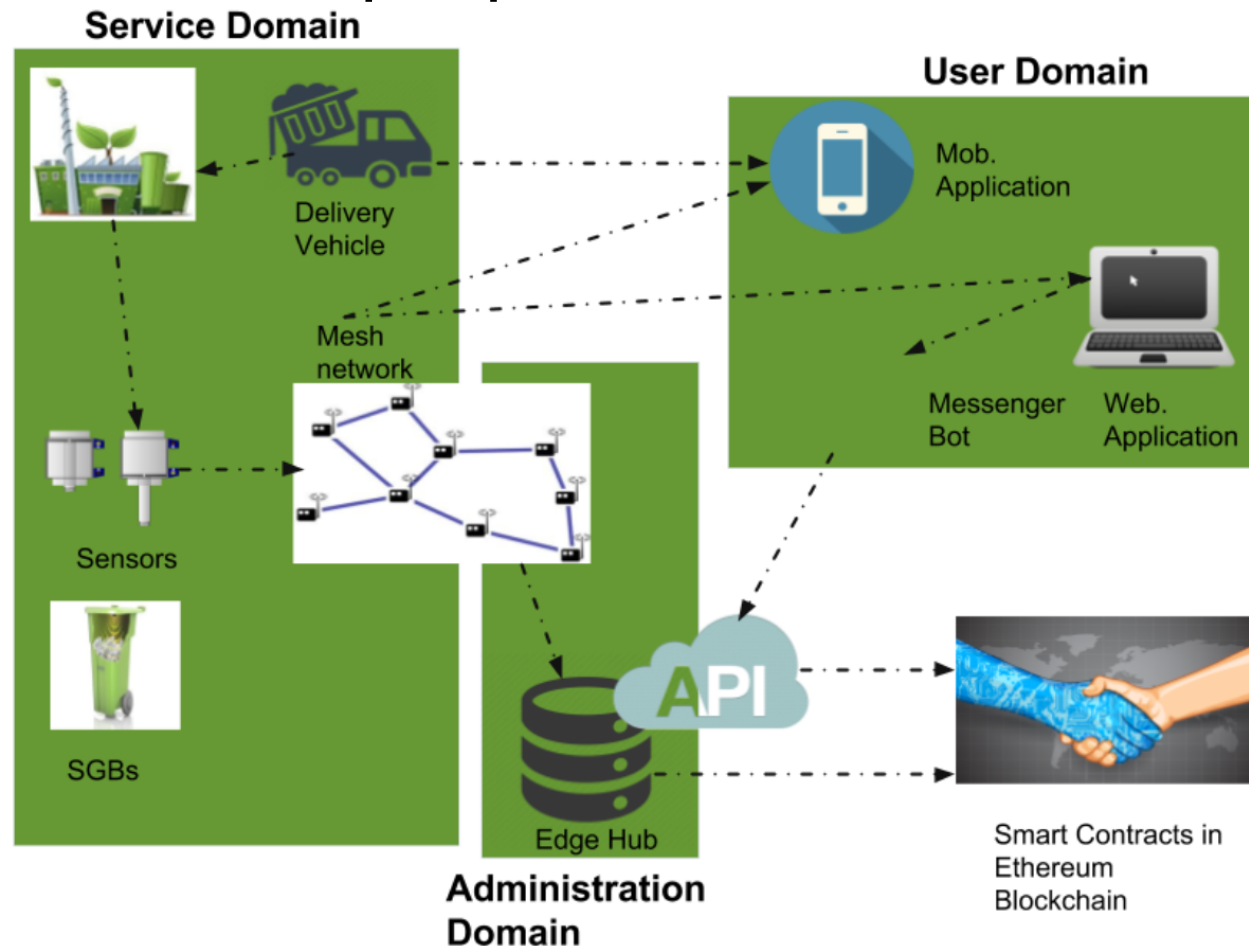
- REST API
- LoRaWAN
- OMI
- NGSI



ITMO UNIVERSITY

Smart City

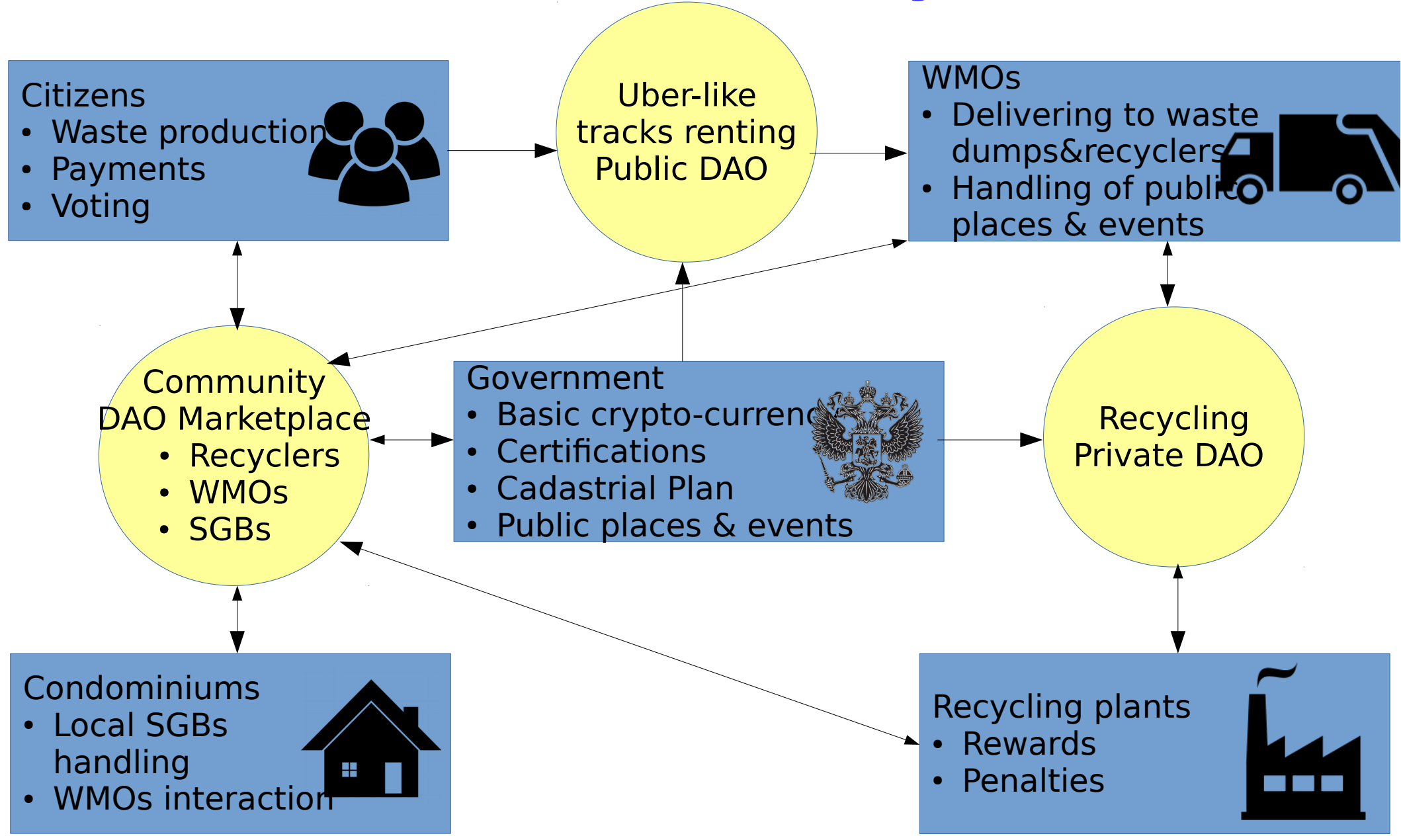
The test data used in the works should correspond to the standard data of a modern metropolis with a population of more than one million people.





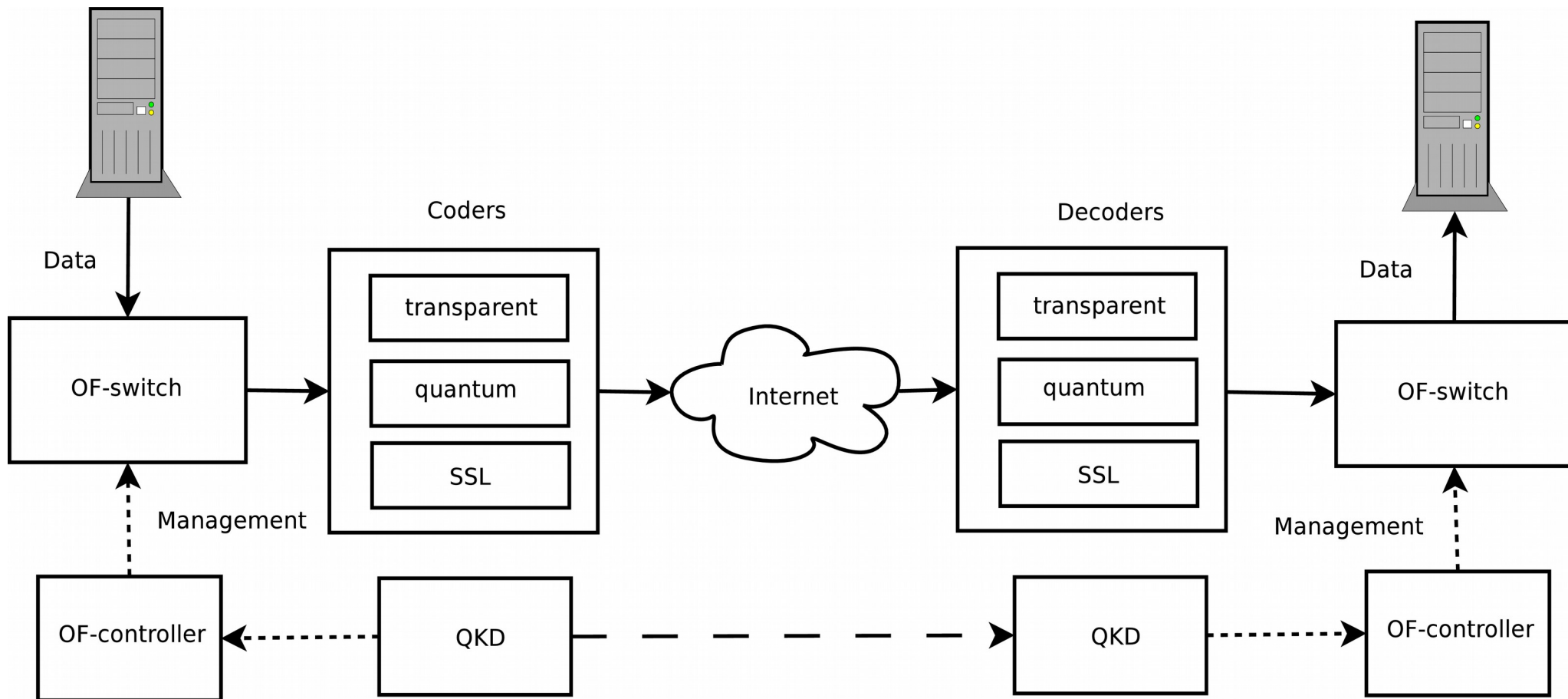
ITMO UNIVERSITY

Blockchain Ecosystem



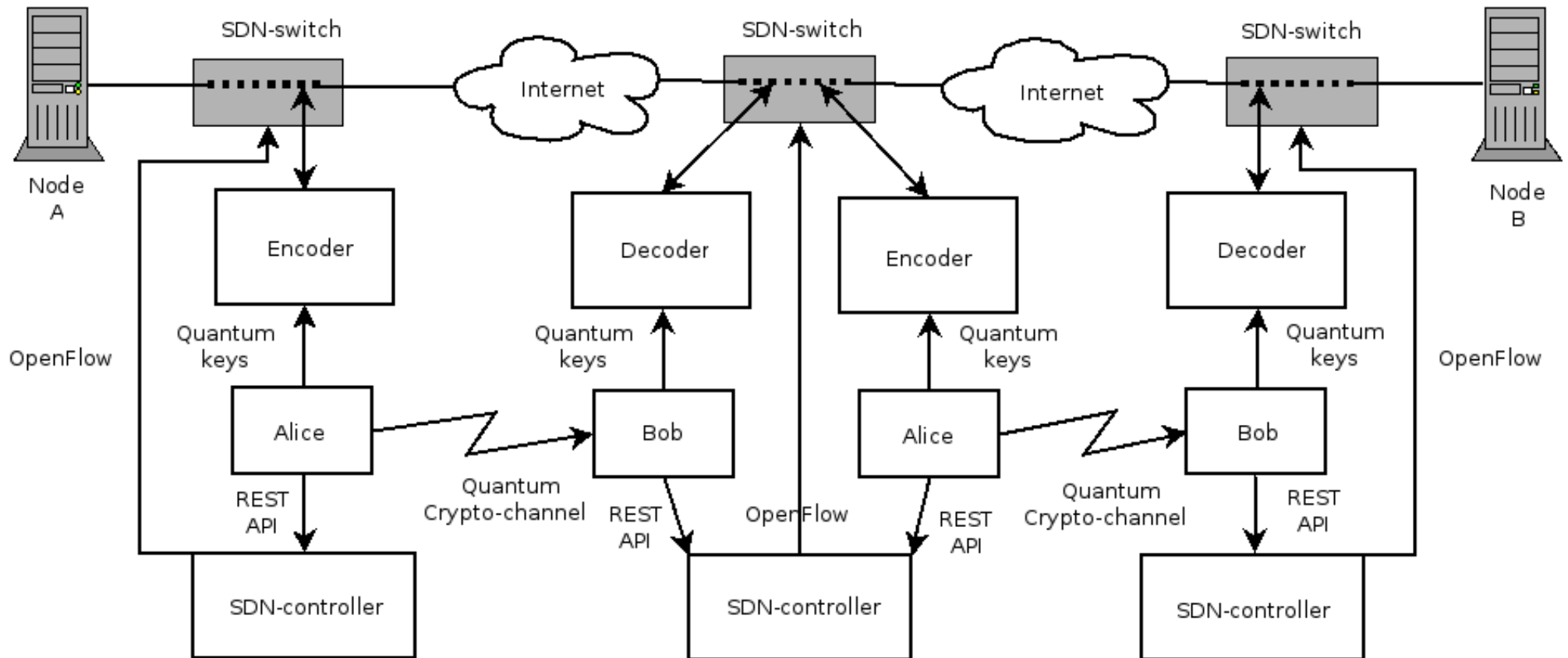


Virtual Cryptography Channel





Quantum Network





ITMO UNIVERSITY

SafeNet project





BAL – Block[Chain] Alchemy Lab

<http://balchemistrylab.gitlab.io/> – source code & wiki

<http://downloads.naulinux.ru/static/Projects/BAL/> – VM images

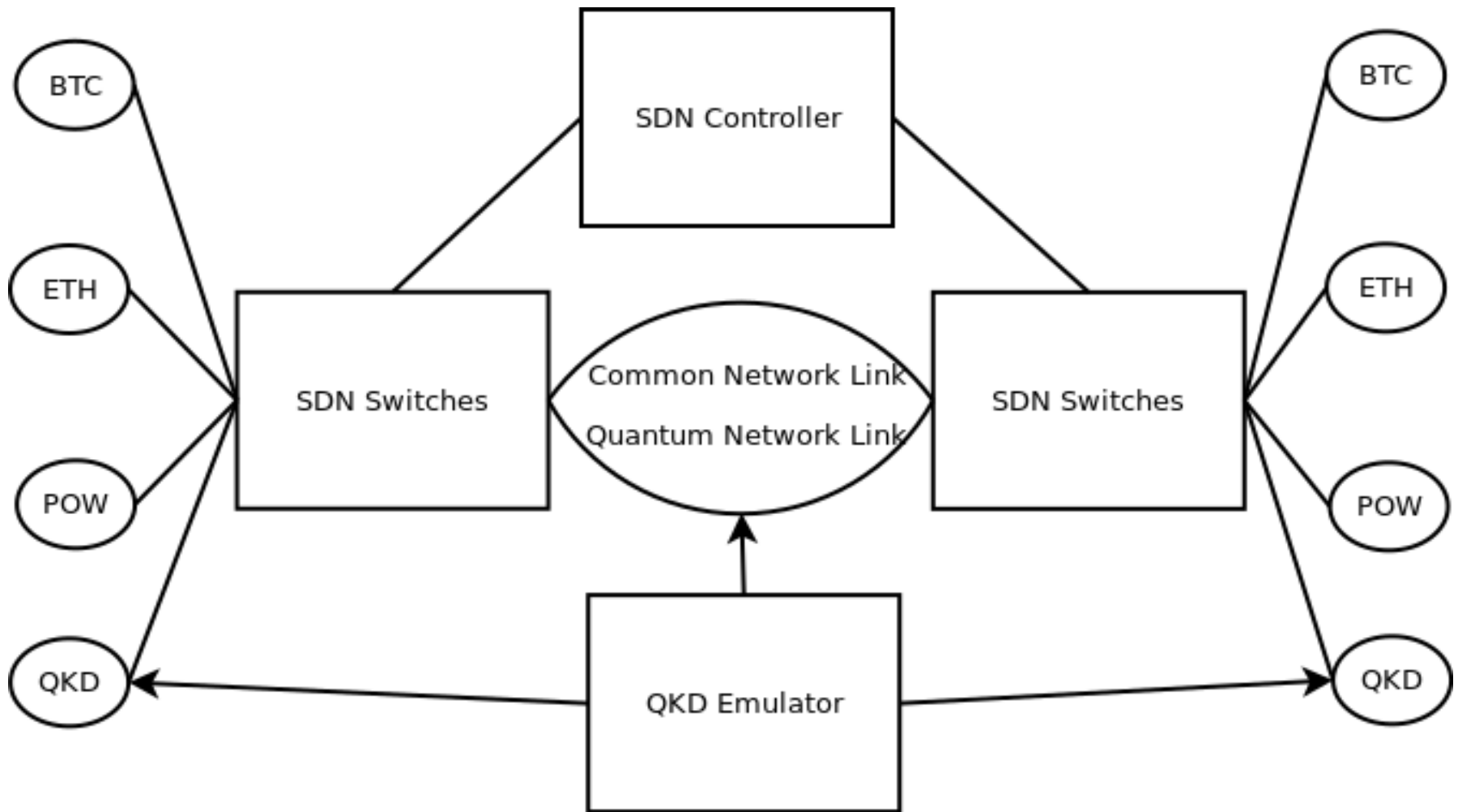
The goal of the project "Block[Chain] Alchemy Lab" – creation of a toolkit for educational and R&D activities for blockchain-related tasks. This toolkit consist from programming framework and tools for modelling different blockchain and not only blockchain (indicated by the optionality of "chain") network environments.

At present, the implemented Mininet-based Python framework with classes provides a simple abstraction for interacting with Block[Chains]. Local nodes are simply one or more processes on the local machine.

- BCNode is a Node that is running an block[chain] application.
- EthNode is a BCNode that is running an Geth application.
- BtcNode is a BCNode that is running an Bitcoin application.
- POWNode is a BCNode that is running an POWBlockChain. Difficulty may be set as a parameter
- QNode is a BCNode that is running an QuantumBlockChain application.



Block[Chain] Alchemy Framework





BAL – Examples

Different types of hosts:

`mn --custom=balmn.py --host=pow:2`

`btc` — `bcnode.BtcNode`

`eth` — `bcnode.EthNode`

`pow` — `bcnode.POWNode`

`qkd` — `bcnode.Qnode`

Test with quantum links:

`mn --custom=bal/QKCustom.py --link=qk --topo=tree,depth=2,fanout=3`

QKD topology:

`mn --custom balmn.py --topo qtopo --mac --controller remote --switch
OVS`