



iExec

Workshop - EthCC 2019

iExec V3 developer edition

<https://iex.ec>

Introduction

Jean-Charles Cabelguen

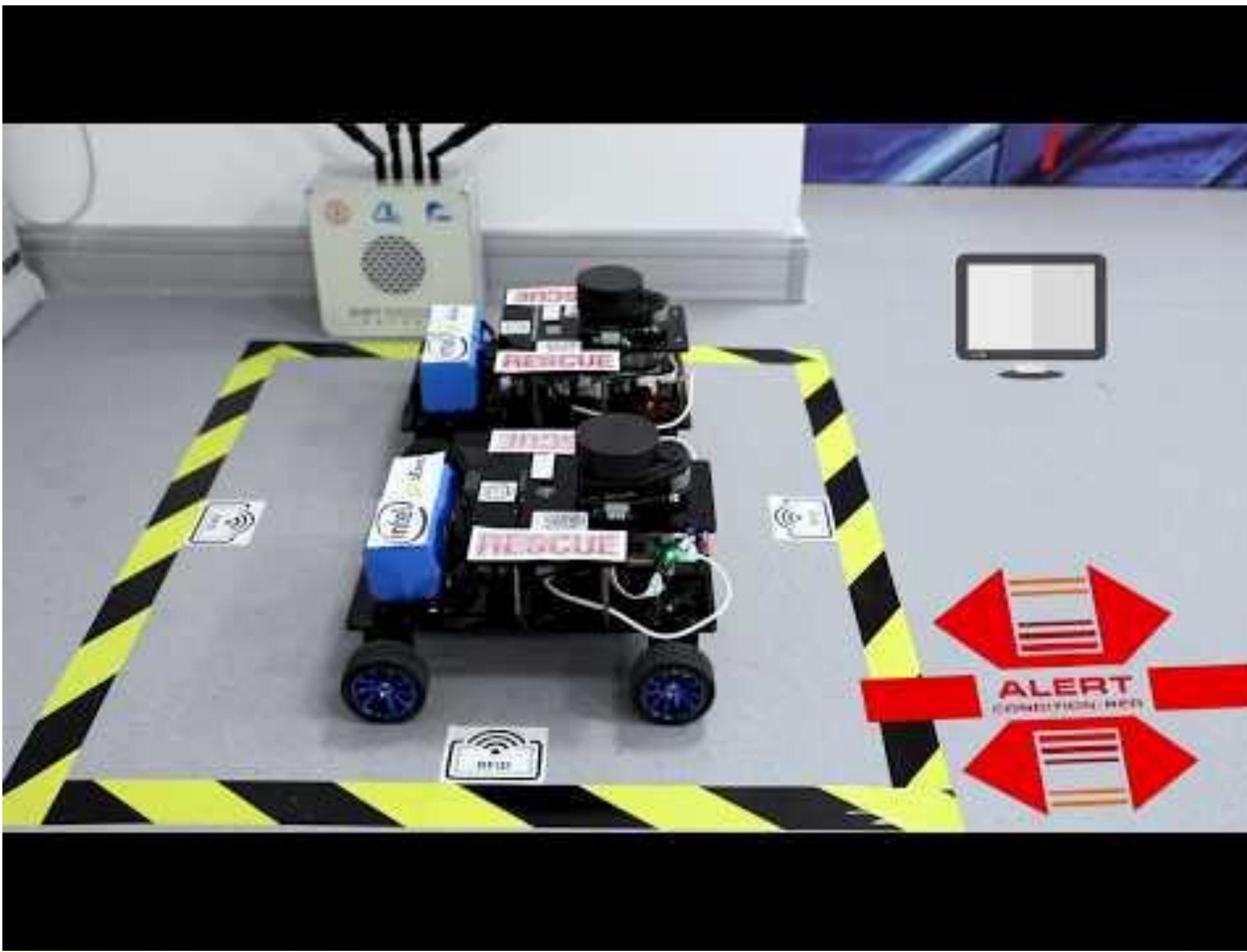


Agenda

- What's new for the V3 alpha
- Onchain/offchain consensus
- TEE for security and data monetisation
- Smart contract and reputation
- Hands on #1: setup a worker
- Hands on #2: building a dapp
- Hands on #3: marketplace for computing resources

Use cases - Smart cities

Video MWC2019



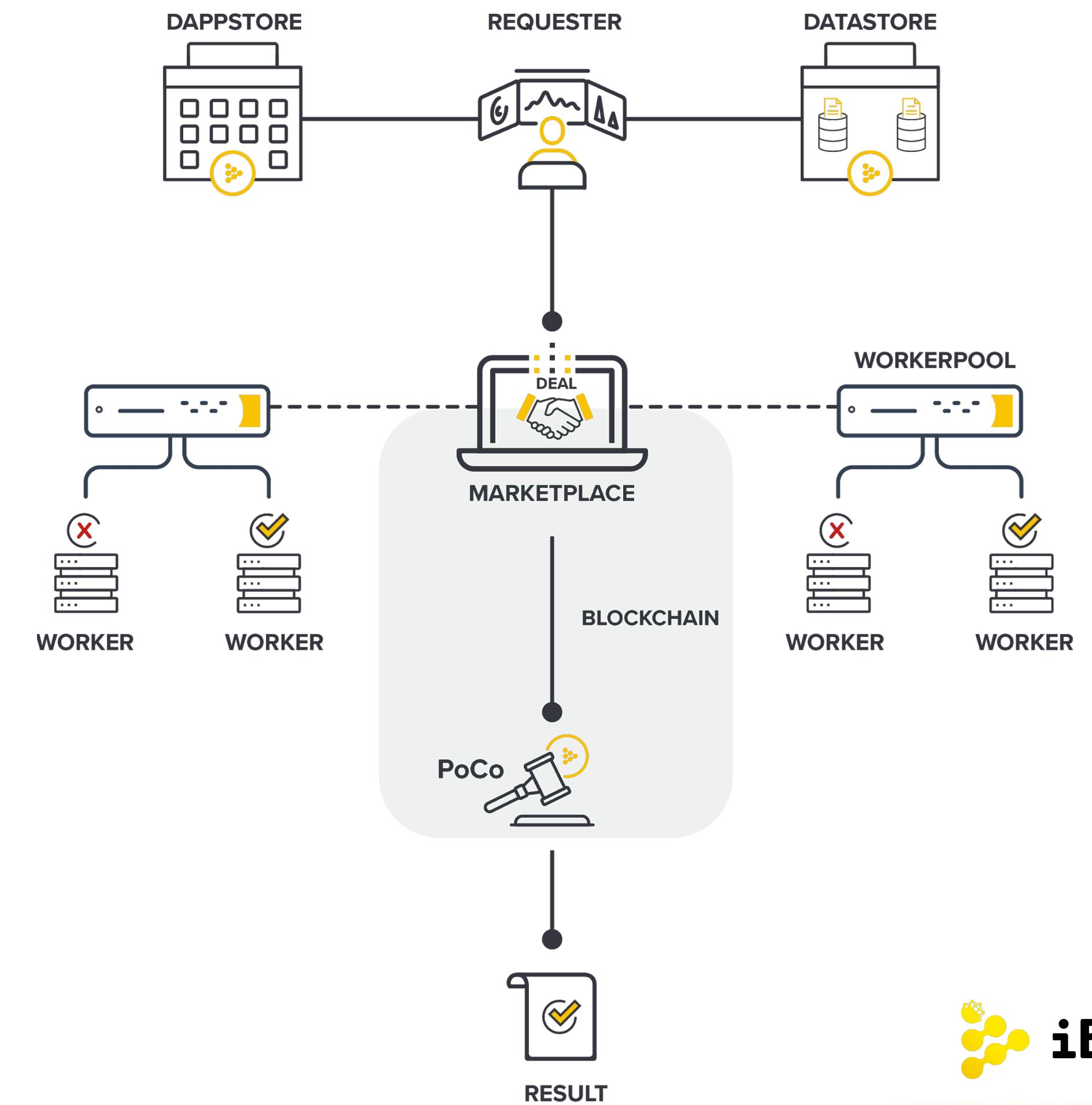
Introducing iExec V3-alpha

Hadrien Croubois



iExec V3 - Enterprise Edition

Data Monetization
Security & Privacy
Access Control
Batch Execution (Bag-of-Task)
...
many more



What's new in iExec V3?

Open Decentralized Brokering

Objective: UX & Scalability

Solution: Hybrid construction

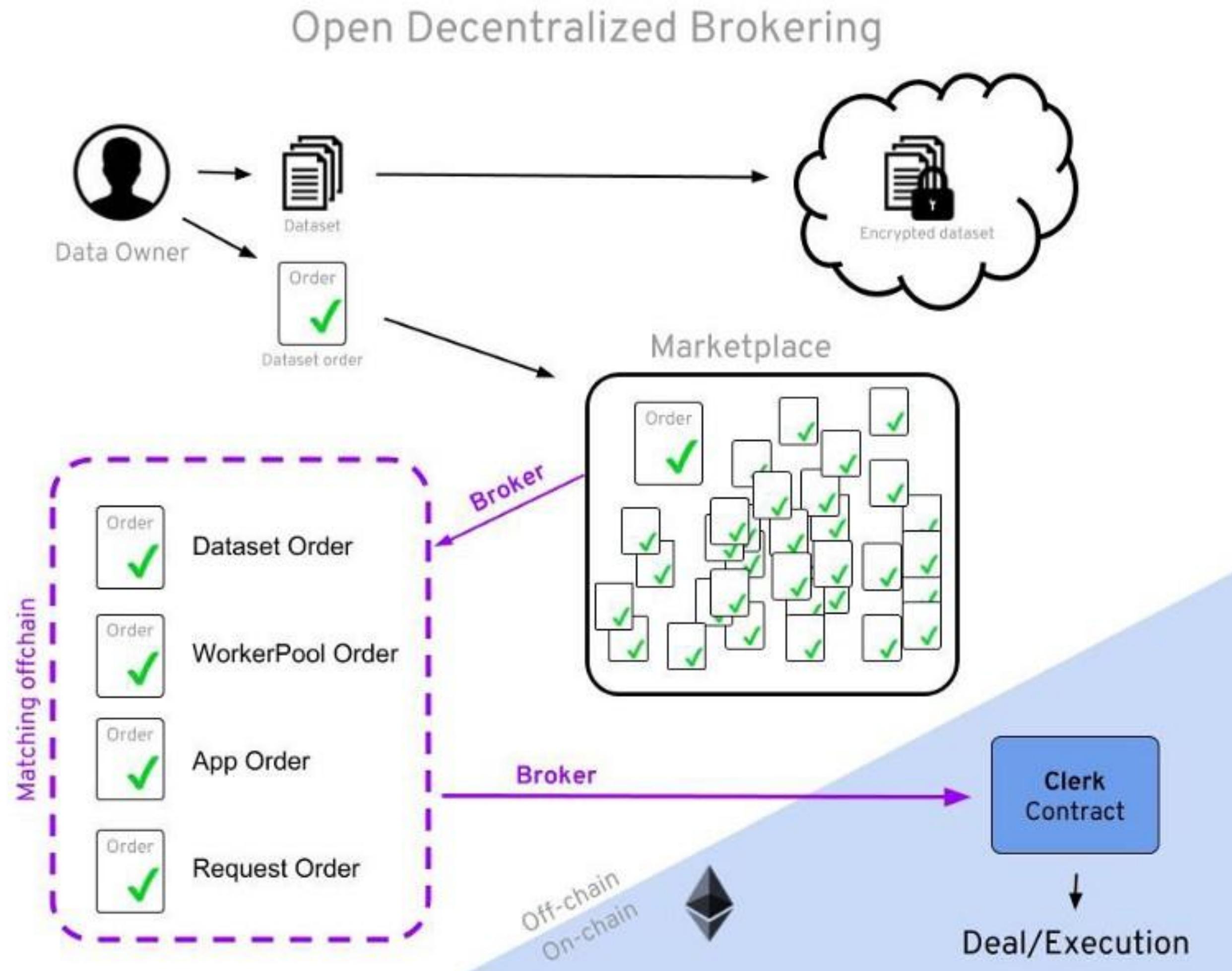
Support signed orders similarly to ox (EIP712)

Many market possibilities (ASK / BID)

Future proof (mostly)



What's new in iExec V3?



What's new in iExec V3?

iExec Core

Objective: Reducing the technological debt

Solution: A brand new middleware

Goodbye XtremWeb-HEP
Hello iExec Core

Designed from the ground up with Blockchain in mind



What's new in iExec V3?

Marketplace, Explorer and SDK

Objective: Easy access to all the features of V3
Solution: New front-end tools

Marketplace
Explorer
SDK



Laptop time ! Setup a Worker

Francois Braciard & Vladimir Ostapenco

- Download next slides and commands here :
 - <https://github.com/iExecBlockchainComputing/iexec-workshop/tree/ethcc2019>



Requirements

- **Git**
 - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
- **Docker**
 - <https://docs.docker.com/install/>
- **Docker-compose**
 - <https://docs.docker.com/compose/install/>
- **nodejs >6**
 - <https://nodejs.org/en/download/>

Plan :

- **Prepare your worker wallet**
- **Launch worker**
- **Worker DappNode package ready too**

Install SDK and Prepare Wallet(1)

- Install iExec SDK
<https://github.com/iExecBlockchainComputing/iexec-sdk/tree/next>
 - using npm : **npm -g install iexec@next**
 - using docker :
 - <https://github.com/iExecBlockchainComputing/iexec-sdk/tree/next#using-docker>
- Initialize SDK environment
 - **iexec --version**
 - **create and go into work directory**
 - **iexec init**
 - This command creates a wallet in:
 - linux : `~/.ethereum/keystore`
 - mac : `~/Library/Ethereum/keystore`
 - windows : `~/AppData/Roaming/Ethereum/keystore`



Install SDK and Prepare Wallet(2)

- Get Kovan ETH
 - **iexec wallet show** => address: __HERE____
 - ask ETH kovan for your address : givemekovaneth@iex.ec
- In **chain.json** add kovan "hub": "0x759b25b358b9f9c18812a69c0agcf8b5a11c2e2d"
 - curl <https://raw.githubusercontent.com/iExecBlockchainComputing/iexec-workshop/ethcc2019/chain.json> --output chain.json
- Get Kovan RLC with iExecSDK
 - **iexec wallet getRLC**
 - Please enter your password to unlock your wallet
 - ✓ Faucets responses:
 - faucet.iex.ec :
 - ok: true
 - message: Successfully requested 200 nRLC. Your position in the waiting list is 1/10
- Deposit RLC to stake
 - **iexec account deposit 200**
 - ✓ deposited 200 nRLC to your iExec account
- Now you wallet is prepared you can launch iExec Worker
 - **iexec wallet show**
 - **iexec account show**



Launch Worker !

- In your working directory git clone :
 - `git clone https://github.com/iExecBlockchainComputing/iexec-worker-deploy.git`
- `cd iexec-worker-deploy/`
- edit **docker-compose.yml**
- Change env variables values
 - `IEXEC_WORKER_NAME` - choose any name
 - `IEXEC_WALLET_JSON` - wallet content in 1 line from:
 - **linux** : `~/.ethereum/keystore`
 - **mac** : `~/Library/Ethereum/keystore`
 - **windows** : `~/AppData/Roaming/Ethereum/keystore`
 - `IEXEC_WALLET_PASSWORD` - wallet password set during "iexec init" phase
(remove space trick : `cat your_wallet | awk -v RS= '{\$1=\$1}1' | tr -d "[:space:]"`)
- `docker-compose up -d`
- `docker-compose logs -f`
- Visualize your worker in grafana: <https://v3-kovan-pool.iex.ec/>



Propaganda break



#ServeYourself

#RunYourOwnNode

#RunYourOwnWorker

#ServeYourFriends

#VPNShareYourNode

#RunYourOwnScheduler



iExec Worker and Scheduler : DappNode* Package ready !

- iExec Worker can be launched on DappNode
 - <https://github.com/iExecBlockchainComputing/DAppNodePackage-iexec-worker>
- iExec Scheduler can be launched on DappNode :
 - https://github.com/iExecBlockchainComputing/DAppNodePackage-iexec-schedule_r-server
 - https://github.com/iExecBlockchainComputing/DAppNodePackage-iexec-schedule_r-db

*<https://dappnode.io/>

DappNode Package runnings

Packages

Packages		
kovan version 0.1.2	RUNNING	<button>Open</button> <button>Remove</button>
dappnodepackage-iexec-scheduler-server version /ipfs/QmXNveMyRtaqnzJgu6D2xAvuYNYkkNufxtQtBvUgJHCRD3	RUNNING	<button>Open</button> <button>Remove</button>
dappnodepackage-iexec-scheduler-db version /ipfs/QmcdpDyybtDzBM1yvUwQkk6bb8KxeqDNtoWmXTyroBxGtp	RUNNING	<button>Open</button> <button>Remove</button>
dappnodepackage-iexec-worker version /ipfs/Qmf5bkBwAtpPigPsnTjrBPqXeJSo9uSEZFz4w8dnHVcBQo	RUNNING	<button>Open</button> <button>Remove</button>



iExec Worker on DappNode

DAppNode

ADMIN UI

- Dashboard
- Activity
- Devices
- Installer
- Packages
- System
- Sdk

packages dappnodepackage-iexec-worker.public.dappnode.eth

Stats

Description: DappNode package for iExec Worker
Version: /ipfs/Qmf5bkBwAtpPigPsnTjrBPqXeJSoS9uSEZFz4w8dnHVcBQo
Use link: <my.dappnodepackage-iexec-worker.public.dappnode.eth>

Volumes:

- /var/run/docker.sock: (bind)

Links: no links
Ports: no ports

Logs

Auto-refresh logs Display timestamps [Scroll to bottom](#)

Lines 200

Search Filter...

```
2019-03-01 10:13:20.214 INFO 8 --- [l-4311-thread-6] c.iexec.worker.feign.CustomFeignClient : COMPUTING [chainTaskId:0x20acd4f1bacc59913fa0d6662ba6ebfab0e8de854c444ebaae823e5dc51bc6c]
2019-03-01 10:13:20.458 INFO 8 --- [l-4311-thread-6] c.s.docker.client.DefaultDockerClient : Starting container with Id: d1ddf88cce4af5123656cb041f5687aa99423ee31627a06b2d68fdf2cff1880b
2019-03-01 10:13:21.937 INFO 8 --- [l-4311-thread-6] c.i.worker.docker.CustomDockerClient : Computation started [taskId:0x20acd4f1bacc59913fa0d6662ba6ebfab0e8de854c444ebaae823e5dc51bc6c,
2019-03-01 10:13:22.943 INFO 8 --- [l-4311-thread-6] c.i.worker.docker.CustomDockerClient : Computation running [taskId:0x20acd4f1bacc59913fa0d6662ba6ebfab0e8de854c444ebaae823e5dc51bc6c,
2019-03-01 10:13:23.949 INFO 8 --- [l-4311-thread-6] c.i.worker.docker.CustomDockerClient : Computation running [taskId:0x20acd4f1bacc59913fa0d6662ba6ebfab0e8de854c444ebaae823e5dc51bc6c,
2019-03-01 10:13:24.958 INFO 8 --- [l-4311-thread-6] c.i.worker.docker.CustomDockerClient : Computation running [taskId:0x20acd4f1bacc59913fa0d6662ba6ebfab0e8de854c444ebaae823e5dc51bc6c,
2019-03-01 10:13:25.964 INFO 8 --- [l-4311-thread-6] c.i.worker.docker.CustomDockerClient : Computation running [taskId:0x20acd4f1bacc59913fa0d6662ba6ebfab0e8de854c444ebaae823e5dc51bc6c,
2019-03-01 10:13:26.973 INFO 8 --- [l-4311-thread-6] c.i.worker.docker.CustomDockerClient : Computation running [taskId:0x20acd4f1bacc59913fa0d6662ba6ebfab0e8de854c444ebaae823e5dc51bc6c,
2019-03-01 10:13:26.983 INFO 8 --- [l-4311-thread-6] c.i.worker.docker.CustomDockerClient : Computation completed [taskId:0x20acd4f1bacc59913fa0d6662ba6ebfab0e8de854c444ebaae823e5dc51bc6c]
```



Build an iExec dApp

Eric Rodriguez

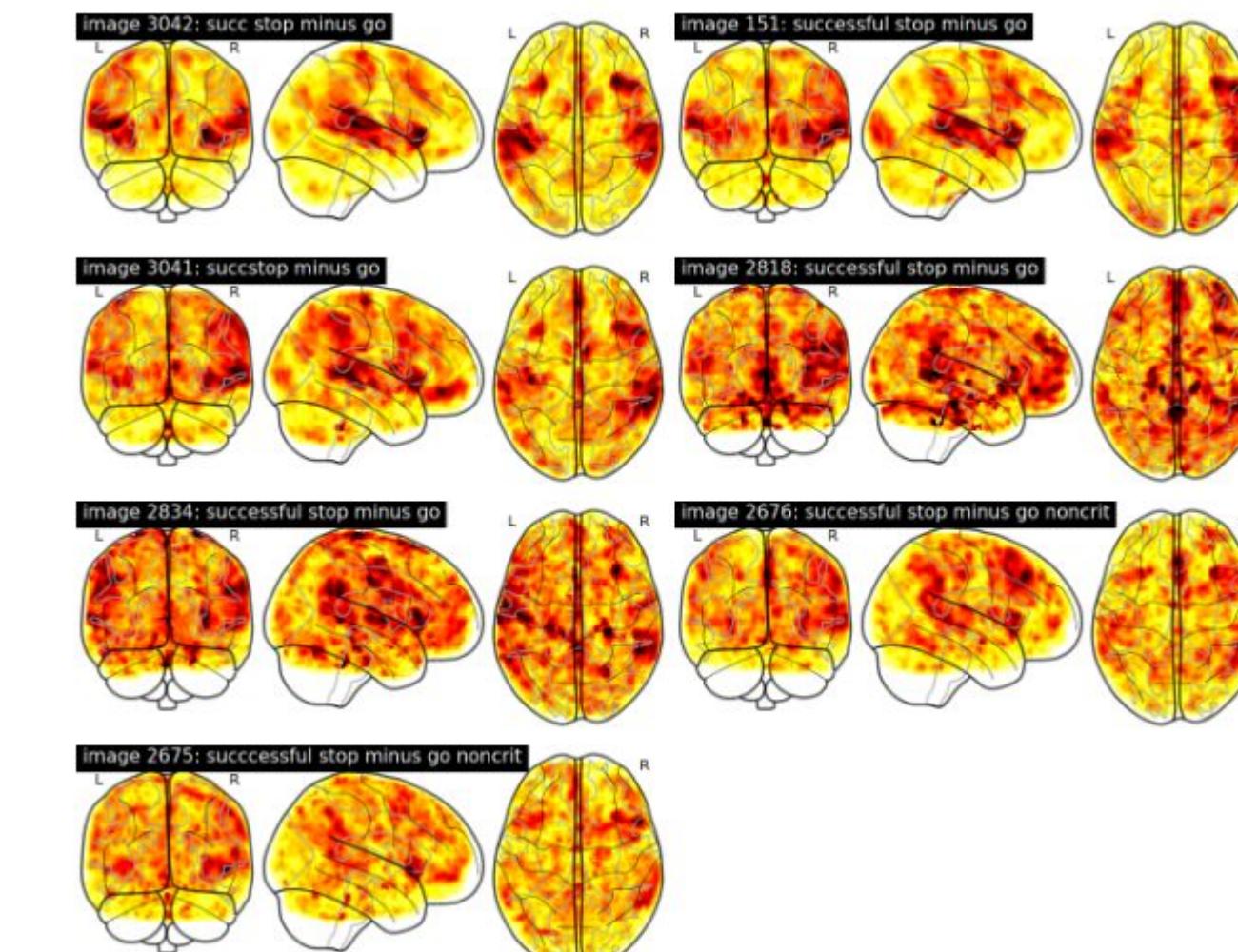


Neuroimaging processing

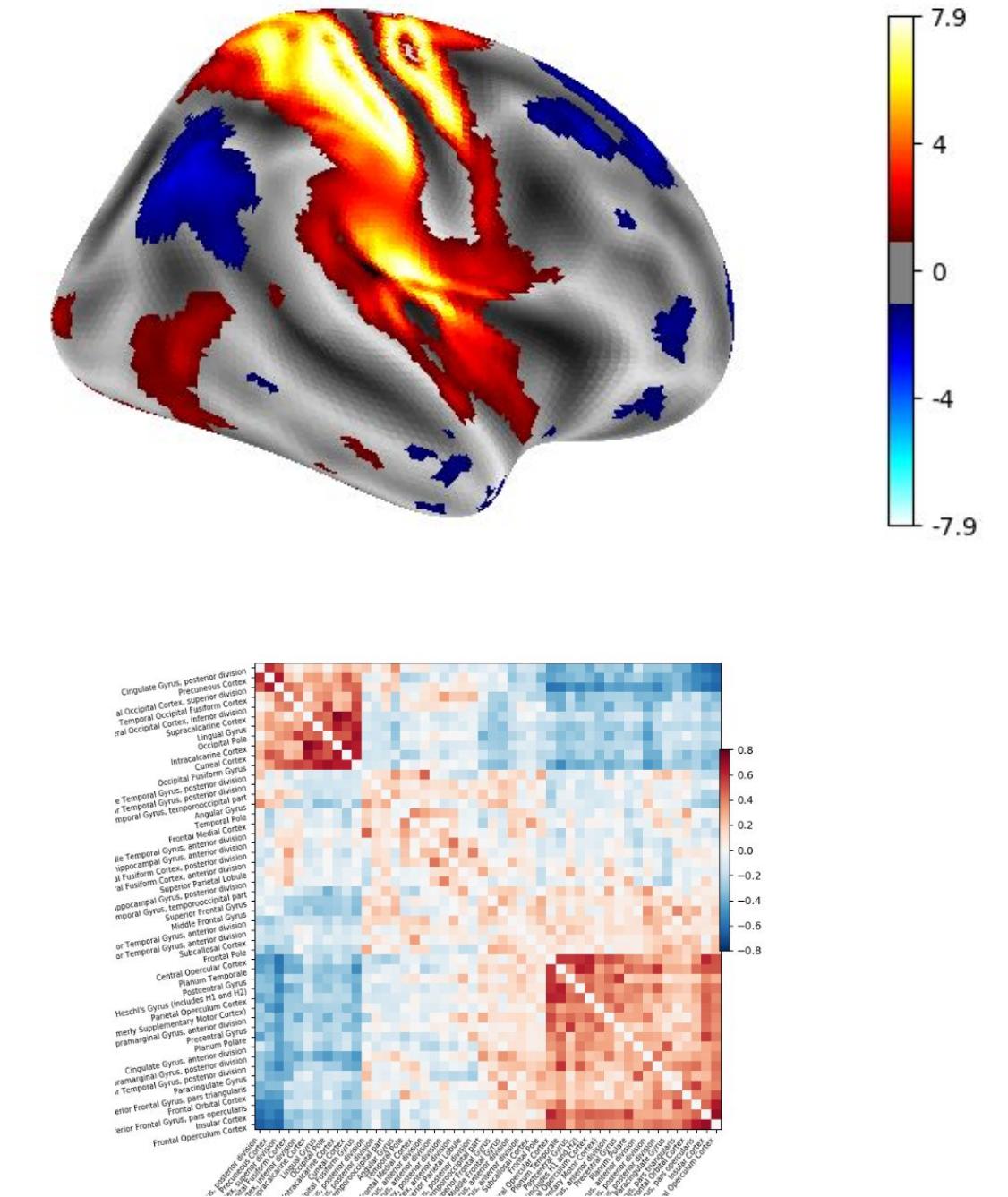


Nilearn:
a framework for
statistical learning on
NeuroImaging data

based on scikit learn



Surface right hemisphere



<https://nilearn.github.io/>

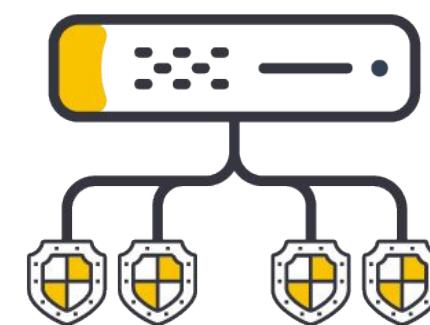


Enhance application

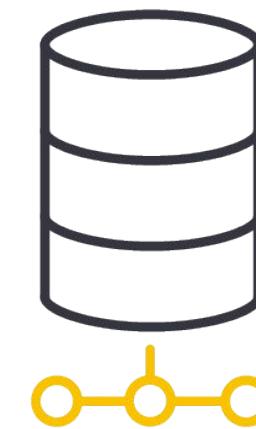
Allows new use cases:

Private usage

Compatible with dataset renting



WORKERPOOL/SGX



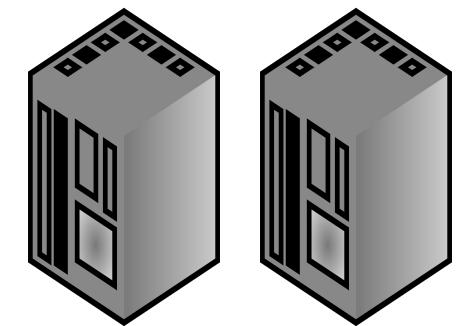
DATASET



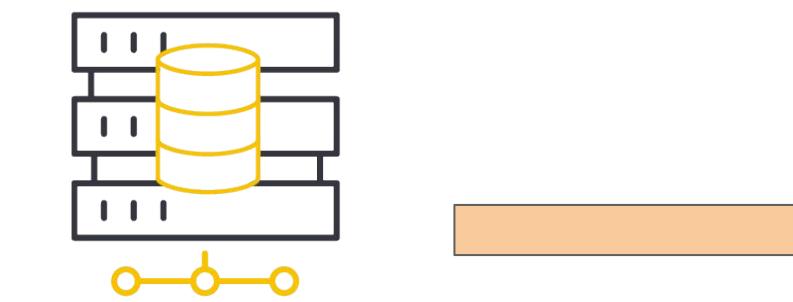
Prepare your dapp

Command line
application (server mode)

Cloud
providers



Data Management



SCIENTIFIC DATA



/iexec_in
compute
/iexec_out
.EXE



Verified result with the PoCo

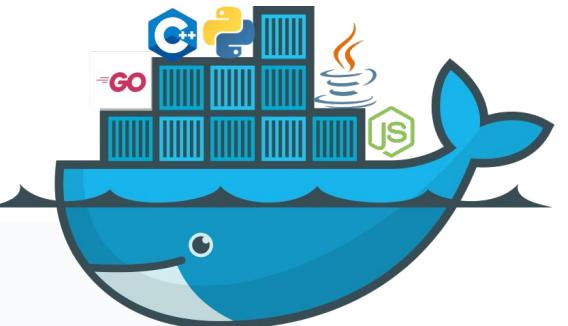
Consolidated result with a high level of trust

Application must be deterministic

Under the developer responsibility

For nilearn app,
md5sum of the results in *iexec_out/consensus.iexec*





Docker application

Docker container contains code and dependencies

Lightweight
Fast
Portable

For SGX enclaves, we will provide tools to transform the container.

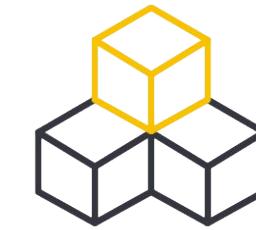
18 lines (12 sloc) | 433 Bytes

```
1 # Use an official Python runtime as a parent image
2 FROM python:3.6-slim
3
4 RUN apt-get update && apt-get install -y wget zip
5
6 # Set the working directory to /app
7 WORKDIR /
8
9 # Install any needed packages
10 RUN pip install scikit-learn nilearn matplotlib
11
12 # Add program file and data management
13 ADD plot_3d_and_4d_niimg_nogui.py .
14 ADD test.sh .
15
16 # Run app.py when the container launches
17 CMD ["./test.sh", "/iexec_out", "image_"]
```



Register and publish

<https://v3.explorer.iex.ec/kovan>



BLOCKCHAIN

Blockchain registration

Last apps deployed				
Time	TxHash	App	Owner	Name
a day a...	0xf507...	0x1739...	0x7800...	nilearn
a day a...	0xa959...	0x59fB...	0x7800...	nilearn



DAPP STORE

Dapp Store registration (not mandatory)



App is ready ... easy process!

You Build

FEW LINES OF
CODE

BLOCKCHAIN
COMPLEXITY
IS HIDDEN

Additionnal Features

PRIVATE RUN

MONETIZE APPS:
AUTOMATIC
PAYMENTS

CONNECT TO SMART CONTRACTS

ACCESS CONTROL

NO
DEPLOYMENT



Conclusion

Developer focuses on what really matters?

develop APPLICATION

Next step:

bags of tasks (ex: for optimization)

GPU computing

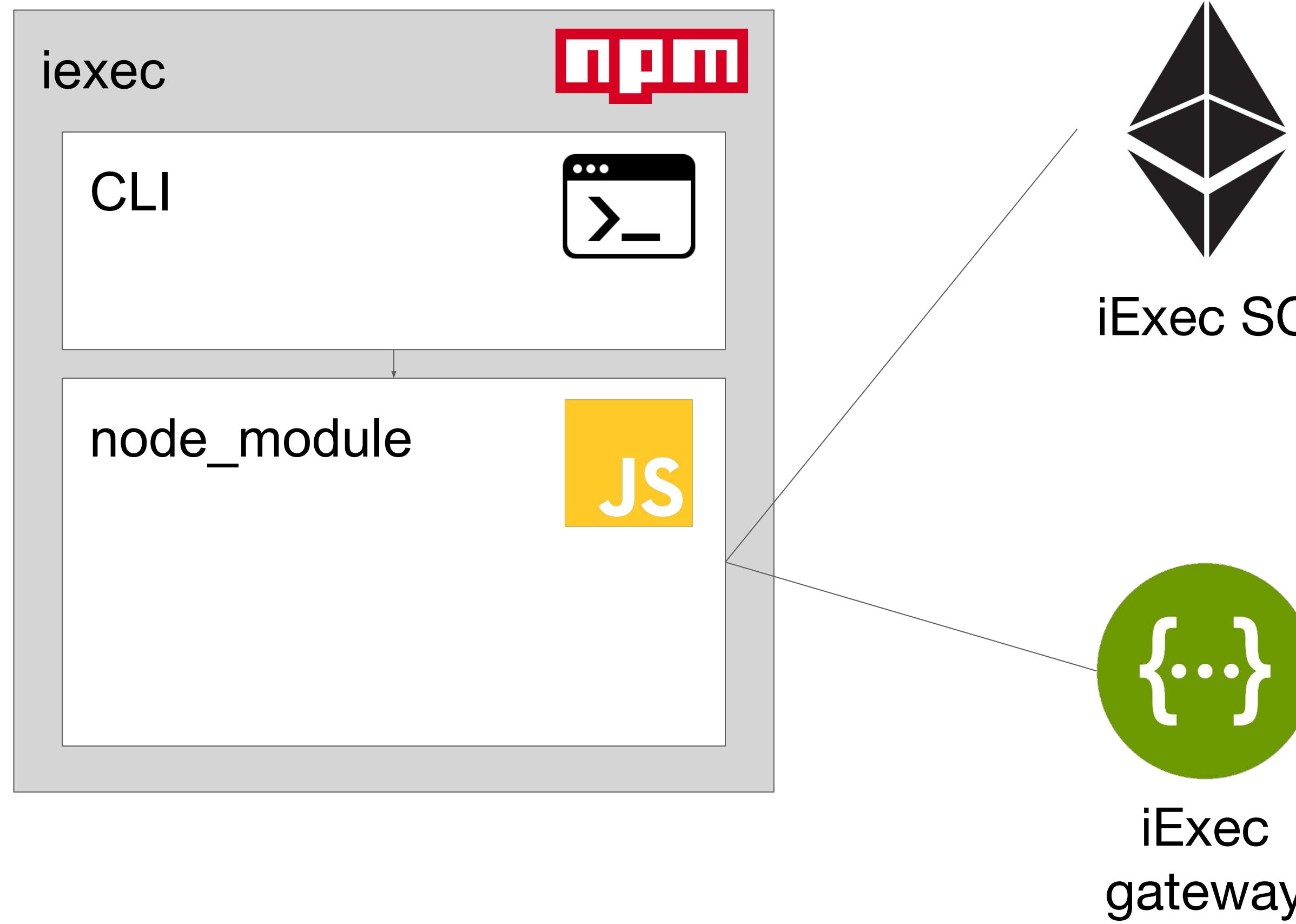


Tools for interacting with iExec v3

Pierre Jeanjacquot



Tools for developers: iExec SDK

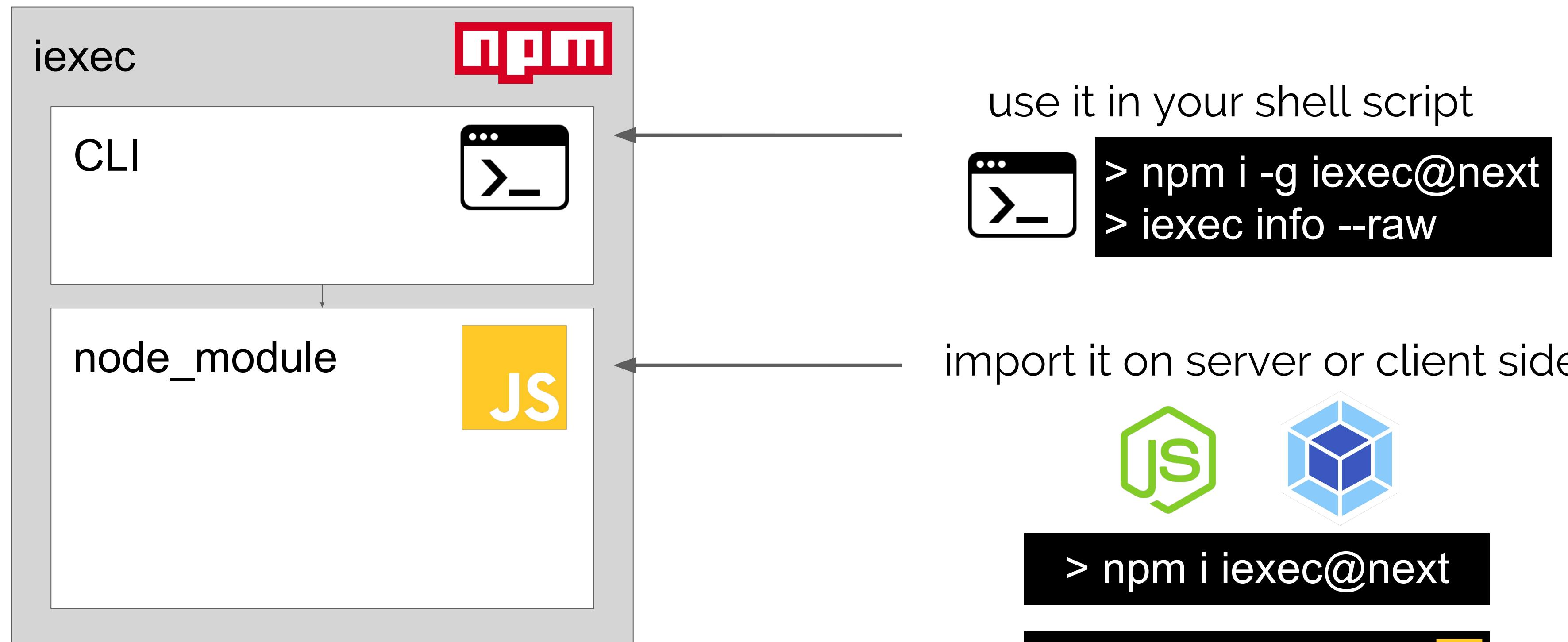


- Assets management
- Resource deployment
- iExec Smart contracts inspection
- Deal registration

- Order publication
- Public orders inspection
- Orderbook



Build on the top of iExec SDK



Deploy and sell your resource with iExec SDK

What we will do:

- Deploy a prepared docker app on iexec
- Set app selling conditions
- Publish a sell apporder

Requirements:

- Docker
- iexec sdk CLI
 - npm i -g iexec@next
 - chain.json configured
 - wallet with kovan Eth



Deploy and sell your resource with iExec SDK

Deploy your app on iExec

Pull the image of your app from dockerhub

- > docker pull iexechub/vanitygen:1.0.1

Init a new app template

- > iexec app init

Edit the app in iexec.json

- > iexec wallet show # find your address
- checksum is your app digest 0x prefixed
- edit iexec.json

deploy the app

- > iexec app deploy
- > cat deployed.json

```
$ docker pull iexechub/vanitygen:1.0.1
1.0.1: Pulling from iexechub/vanitygen
Digest: sha256:fa69ce67286e5f2d209ee91a1e7a7ada7e3b1c63be6982024769221ea8c316c0
Status: Image is up to date for iexechub/vanitygen:1.0.1

$ iexec wallet show
? Using wallet
UTC--2019-02-28T18-08-28.819000000Z--F048eF3d7E3B33A465E0599E641BB29421f7Df92
...
address: 0xF048eF3d7E3B33A465E0599E641BB29421f7Df92

$ iexec app init
✓ Saved default app in "iexec.json", you can edit it:
...

$ vi iexec.json
$ cat iexec.json
...
{
  "app": {
    "owner": "0xF048eF3d7E3B33A465E0599E641BB29421f7Df92",
    "name": "VanityGen",
    "type": "DOCKER",
    "multiaddr": "registry.hub.docker.com/iexechub/vanitygen:1.0.1",
    "checksum": "0xfa69ce67286e5f2d209ee91a1e7a7ada7e3b1c63be6982024769221ea8c316c0",
    "mrenclave": ""
  }
}

$ iexec app deploy
① using chain [kovan]
...
✓ Deployed new app at address 0x07d50ab03e7D17D1148a9277fCA703833E86785D
```

Deploy and sell your resource with iExec SDK

Sell your app on iExec marketplace

make sure your app is deployed

- > iexec app show <appAddress>

generate an apporder

- > iexec order init --app
 - edit apporder appprice in iexec.json
 - > iexec order sign --app

publish your order on the marketplace

- > iexec order publish --app

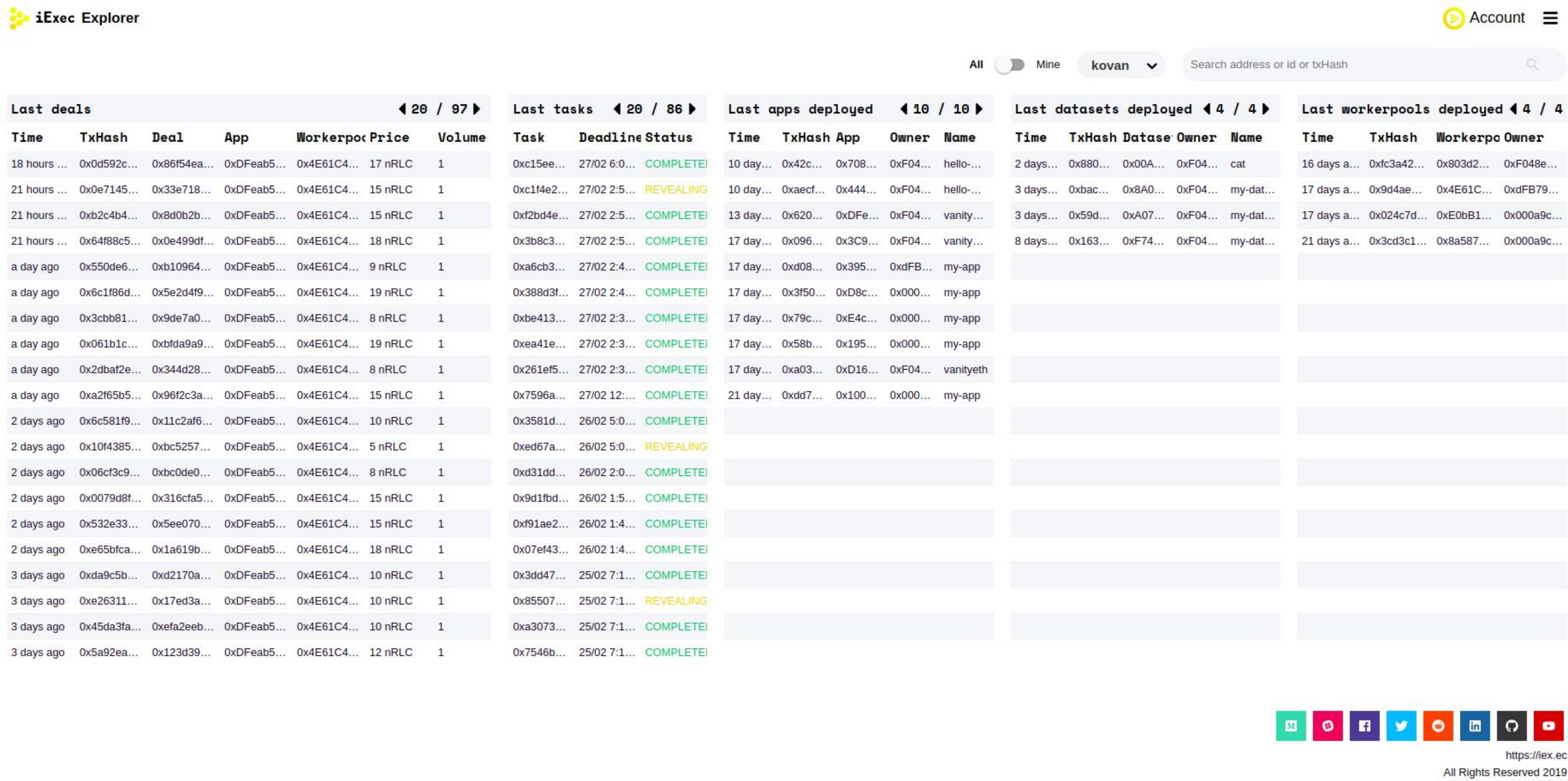
show the published orders for your app

- > iexec orderbook app <appAddress>

Congratulation ! Your app is available on iExec !



Explore iExec Smart contracts with iExec explorer



The screenshot shows the iExec Explorer interface with several sections of data:

- Last deals**: A table with columns: Time, TxHash, Deal, App, Workerpool, Price, Volume. It shows 20 out of 97 deals.
- Last tasks**: A table with columns: Task, Deadline, Status. It shows 20 out of 86 tasks.
- Last apps deployed**: A table with columns: Time, TxHash, App, Owner, Name. It shows 10 out of 10 apps.
- Last datasets deployed**: A table with columns: Time, TxHash, Dataset, Owner, Name. It shows 4 out of 4 datasets.
- Last workerpools deployed**: A table with columns: Time, TxHash, Workerpool, Owner. It shows 4 out of 4 workerpools.

At the bottom, there are social media sharing icons and links to the website (<https://iex.ec>) and copyright information ("All Rights Reserved 2019").

Explorer v3-alpha is live: <https://v3.explorer.iex.ec>



Explore deployed resources with iExec explorer

> iexec app deploy

All Mine kovan Search address or id or txHash

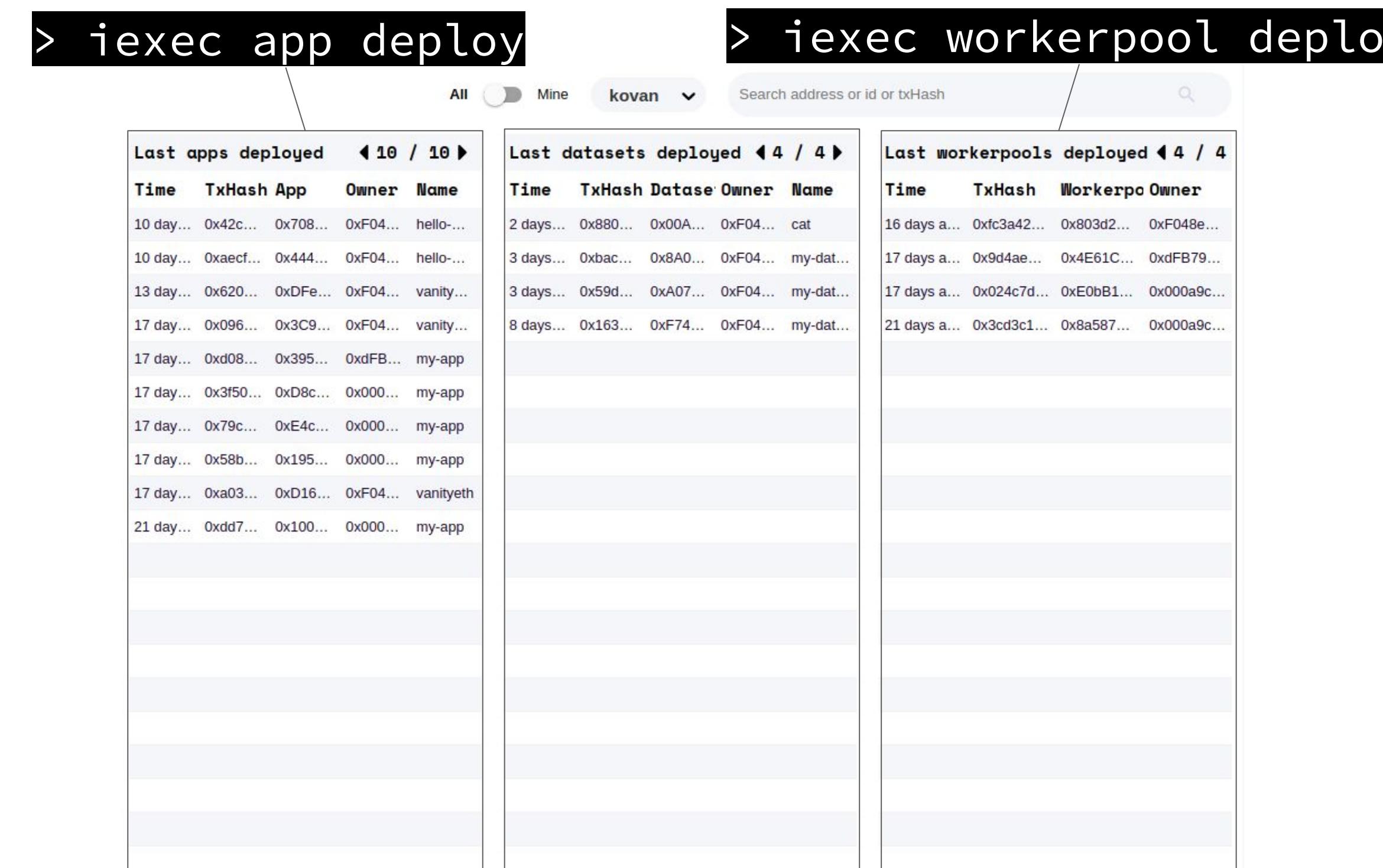
Last apps deployed ▶ 10 / 10 ▶				
Time	TxHash	App	Owner	Name
10 day...	0x42c...	0x708...	0xF04...	hello-...
10 day...	0xaecf...	0x444...	0xF04...	hello-...
13 day...	0x620...	0xDFe...	0xF04...	vanity...
17 day...	0x096...	0x3C9...	0xF04...	vanity...
17 day...	0xd08...	0x395...	0xFB...	my-app
17 day...	0x3f50...	0xD8c...	0x000...	my-app
17 day...	0x79c...	0xE4c...	0x000...	my-app
17 day...	0x58b...	0x195...	0x000...	my-app
17 day...	0xa03...	0xD16...	0xF04...	vanityeth
21 day...	0xdd7...	0x100...	0x000...	my-app

> iexec workerpool deploy

Last workerpools deployed ▶ 4 / 4 ▶				
Time	TxHash	Workerpo	Owner	Name
16 days a...	0xfc3a42...	0x803d2...	0xF048e...	cat
3 days...	0xbac...	0x8A0...	0xF04...	my-dat...
3 days...	0x59d...	0xA07...	0xF04...	my-dat...
8 days...	0x163...	0xF74...	0xF04...	my-dat...

> iexec dataset deploy

<https://iex.ec>
All Rights Reserved 2019



Explorer v3-alpha is live: <https://v3.explorer.iex.ec>



Buy computation on iExec marketplace

Why a marketplace?

- Allow computation price to fluctuate
- Always get the best price for your computation

How does it works?

- Orders are published on the marketplace
- A deal is settled when orders are matched
- Once the deal is settled, the computation begin

Every computation begins with a deal



Buy computation on iExec marketplace

The screenshot displays the iExec Marketplace v3-alpha interface. At the top, there are tabs for Cat1, Cat2, Cat3 (highlighted in yellow), and Cat4, followed by TEE. Below the tabs is a search bar with placeholder text "Search for computation".

The main area features a chart titled "nRLC/WORK" showing price fluctuations over time. The Y-axis ranges from 0 to 38.

On the right side, there are two tables:

- Order Book:** A table showing open orders for nRLC/WORK. It includes columns for Hash, Price, Workerpool, Trust, and Volume. Some entries are truncated with "Last 8 nRLC ↴".
- Recent Trades:** A table showing completed trades. It includes columns for ID, Price, Time, Workerpool, and Volume.

At the bottom, there are sections for "My Trades", "My Open Request Orders", and "My Open Workerpool Orders". The "Fill Market Order" section contains fields for Order Hash, Request Order Hash, Volume, Dapp Address, and Work Params, along with "Buy computation at market price" and "Sell computation at market price" buttons.

Marketplace v3-alpha is live: <https://v3.market.iex.ec>



Buy computation on iExec marketplace

The screenshot displays the iExec Marketplace v3-alpha interface. At the top, there are tabs for Cat1, Cat2, Cat3 (highlighted), Cat4, and TEE. Below the tabs is a search bar with placeholder text "computation price real time Ask and Bid".

The main area features a chart titled "nRLC/WORK" showing computation price over time. The Y-axis ranges from 0 to 38. The chart shows a fluctuating yellow line with several sharp peaks and troughs.

On the right side, there are two tables:

- Order Book:** A table showing the latest 8 nRLC orders. It includes columns for Hash, Price, Workerpool, Trust, and Volume.
- Recent Trades:** A table showing recent trades. It includes columns for ID, Price, Time, Workerpool, and Volume.

At the bottom, there are sections for "My Trades", "My Open Request Orders", and "My Open Workerpool Orders". The "My Trades" section lists two recent trades with IDs, Price, Time, Workerpool, and Volume. The "Fill Market Order" and "Place Limit Order" forms allow users to buy or sell computation at market price, specifying Order Hash, Request Order Hash, Volume, Dapp Address, and Work Params.

Marketplace v3-alpha is live: <https://v3.market.iex.ec>



Buy computation on iExec marketplace

The screenshot displays the iExec Marketplace v3-alpha interface. At the top, there are tabs for Cat1, Cat2, Cat3 (highlighted in yellow), and Cat4, followed by TEE. Below the tabs, a search bar contains the text "nRLC/WORK". A large chart area shows a line graph titled "view computation prices and trends" with a Y-axis ranging from 2 to 38. The chart displays a fluctuating line with several peaks and troughs. To the right of the chart are two tables: "Order Book" and "Recent Trades". The "Order Book" table lists 20 entries with columns for Hash, Price, Workerpool, Trust, and Volume. The "Recent Trades" table lists 20 entries with columns for ID, Price, Time, Workerpool, and Volume. At the bottom, there are sections for "My Trades", "My Open Request Orders", and "My Open Workerpool Orders", each showing a table of 2 entries. On the right side, there are two buttons: "Buy computation at market price" and "Sell computation at market price".

Hash	Price	Workerpool	Trust	Volume
0xba0c8f3c617d...	15	0x4E61C418cFcF...	1	1
0x5080a6a73765...	14	0x4E61C418cFcF...	1	1
0xb69e283a418b...	12	0x4E61C418cFcF...	1	1
0x22c54610c5ef...	12	0x4E61C418cFcF...	1	1
0xcbddfb16e108...	12	0x4E61C418cFcF...	1	1
0x219fbe85c3da0...	11	0x4E61C418cFcF...	1	1
0xf9b08707356e...	10	0x4E61C418cFcF...	1	1
0x84a6290086f4...	9	0x4E61C418cFcF...	1	1
0x77a9f9ddd6a24...	9	0x4E61C418cFcF...	1	1
0x301ef5b152b2e...	9	0x4E61C418cFcF...	1	1
0xc998053eb08...	8	0x4E61C418cFcF...	1	1
0xb5242cdfd8998...	3	0xF048eF3d7E3B...	0	1
0x13219f58a6e19...	3	0xF048eF3d7E3B...	0	1
0xd47803724fad...	3	0xF048eF3d7E3B...	0	1
0xd6706b3e6294...	2	0xF048eF3d7E3B...	0	1
0x84a1a96069af...	2	0xF048eF3d7E3B...	0	1
0x45bf7442f6712...	2	0xF048eF3d7E3B...	0	1
0xc7698012d600...	1	0xF048eF3d7E3B...	0	1
0x1e5eee5706bb...	1	0xF048eF3d7E3B...	0	1
0x6ad4fb9618763...	1	0xF048eF3d7E3B...	0	1
0x50dd517a9de8...	1	0xF048eF3d7E3B...	0	1
0x115f6aec41ec2...	1	0xF048eF3d7E3B...	0	1

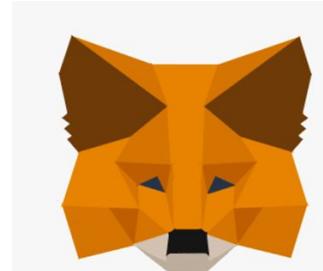
ID	Price	Time	Workerpool	Volume
0xa9ecd2973b71b0b14b3293af5c0d9031bee0744a76d994...	8	11:18:32	0x4E61C418cFcF080DbCA9544AAd64572e68BD9802	1
0xb1096448c72f9df742f805b5fe657e49b5885de1f2a9340c...	9	13:04:44	0x4E61C418cFcF080DbCA9544AAd64572e68BD9802	1

Marketplace v3-alpha is live: <https://v3.market.iex.ec>

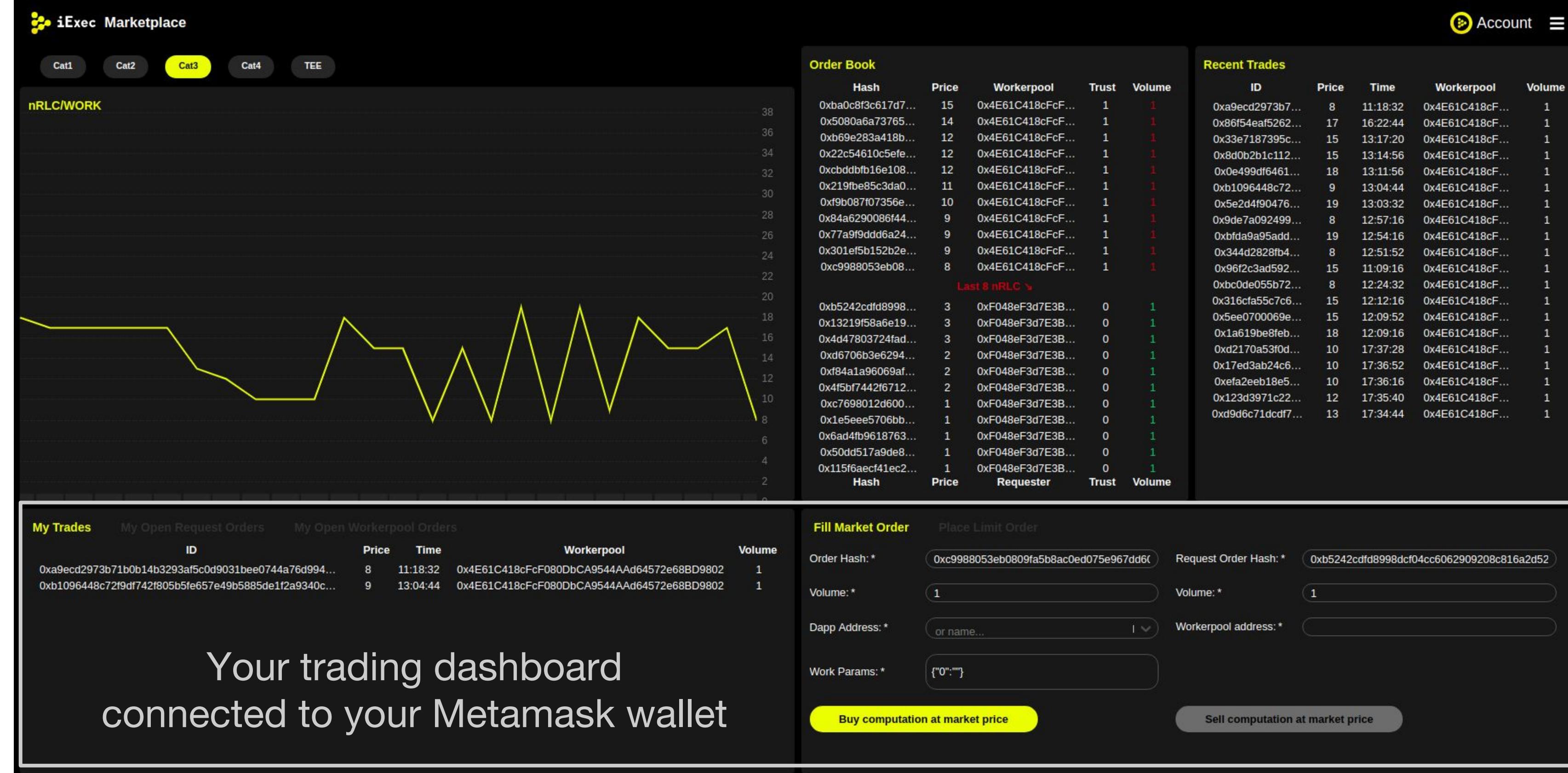


Buy computation on iExec marketplace

Required
Metamask plugin



METAMASK



The screenshot shows the iExec Marketplace interface. At the top, there are tabs for Cat1, Cat2, Cat3 (highlighted in yellow), Cat4, and TEE. Below the tabs is a search bar with placeholder text "Search for computation or workerpool". A sidebar on the left lists categories: nRLC/WORK, WORKERS, and REQUESTERS. The main area displays two tables: "Order Book" and "Recent Trades". The Order Book table shows a list of computation offers with columns for Hash, Price, Workerpool, Trust, and Volume. The Recent Trades table shows a list of completed trades with columns for ID, Price, Time, Workerpool, and Volume. At the bottom, there are sections for "My Trades", "My Open Request Orders", and "My Open Workerpool Orders", each showing a list of items with columns for ID, Price, Time, Workerpool, and Volume. On the right, there is a form for placing a market order with fields for Order Hash, Request Order Hash, Volume, Dapp Address, and Work Params. Buttons for "Buy computation at market price" and "Sell computation at market price" are also present. A central message says "Your trading dashboard connected to your Metamask wallet".



Marketplace v3-alpha is live: <https://v3.market.iex.ec>

Buy computation on iExec marketplace

Buy computation at market price for immediate execution

OR

Place limit order get your computation when the market meets your bid

Fill Market Order **Place Limit Order**

Order Hash: *	0xc9988053eb0809fa5b8ac0ed075e967dd60831c5f00363
Volume: *	1
Dapp Address: *	0xDFeab54290E804318F0a82217cEa88E98a629B6c
Work Params: *	{"0": "my command line"}

Buy computation at market price

ORDER
MATCHED:
DEAL

My Trades **My Open Request Orders** **My Open Workerpool Orders**

ID	Price	Time	Workerpool	Volume
0xa9ecd2973b71b0b14b3...	8	11:18:32	0x4E61C418cFcF080DbC...	1

COMPUTATION STARTS

Fill Market Order **Place Limit Order**

Volume: *	1
Workerpool Price: *	5
Dapp Address: *	0xDFeab54290E804318F0a82217cEa88E98a629B6c
Dapp Price: *	1
Trust: *	0
Work Params: *	{"0": "my command line"}

Place computation buy limit order

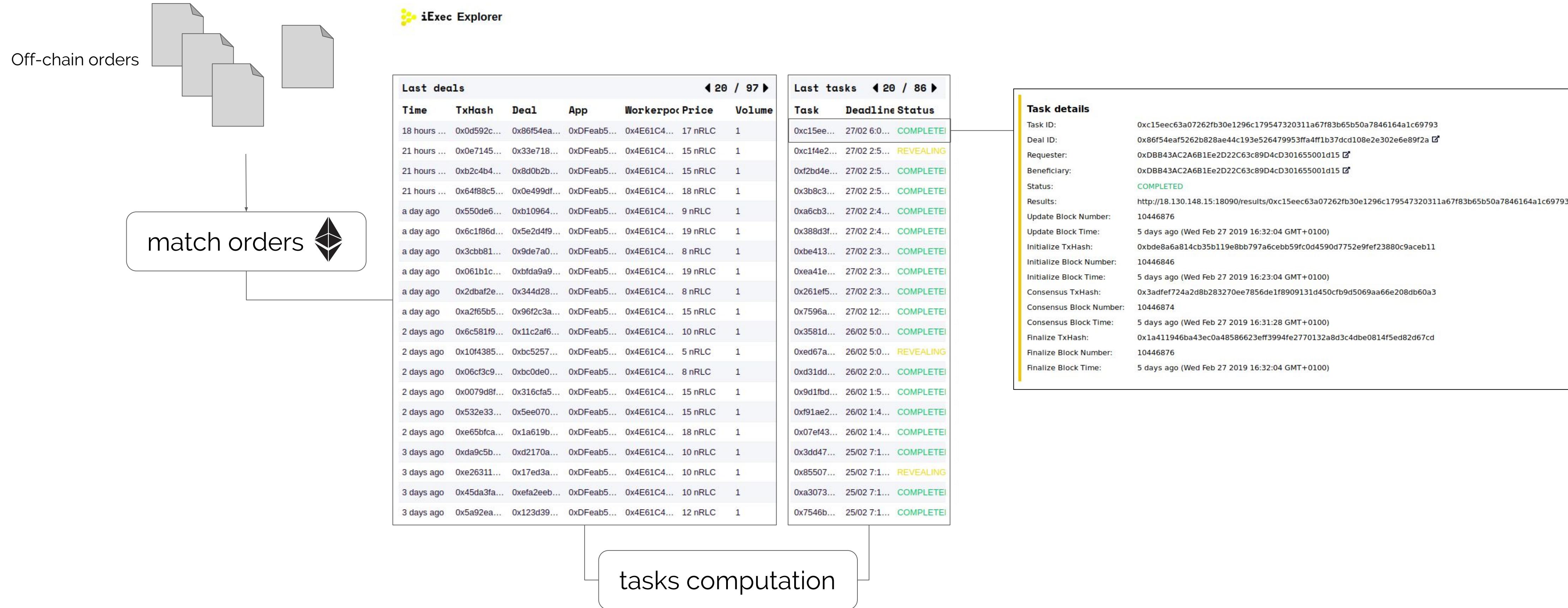
My Trades **My Open Request Orders** **My Open Workerpool Orders**

Hash	Params	Cat.	Trust	Price	Remaining	Cancel
0x50dd517a9de8a16fb5...	{"0": "1RLC"}	3	0	1	1/1	X

ORDER
WAITING
TO BE
MATCHED



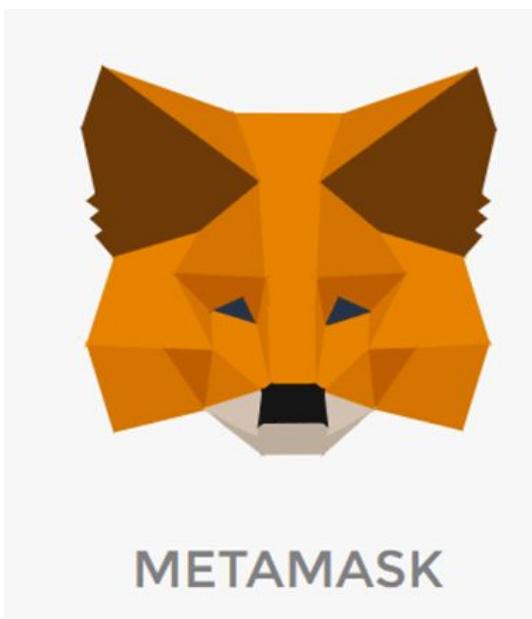
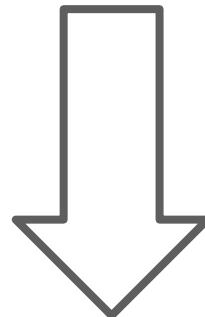
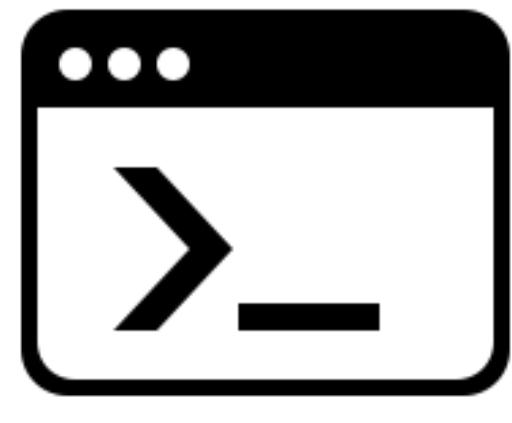
Follow the computations with iExec explorer



Explorer v3-alpha is live: <https://v3.explorer.iex.ec>



Hands on: let's buy some computation



Import your wallet in Metamask

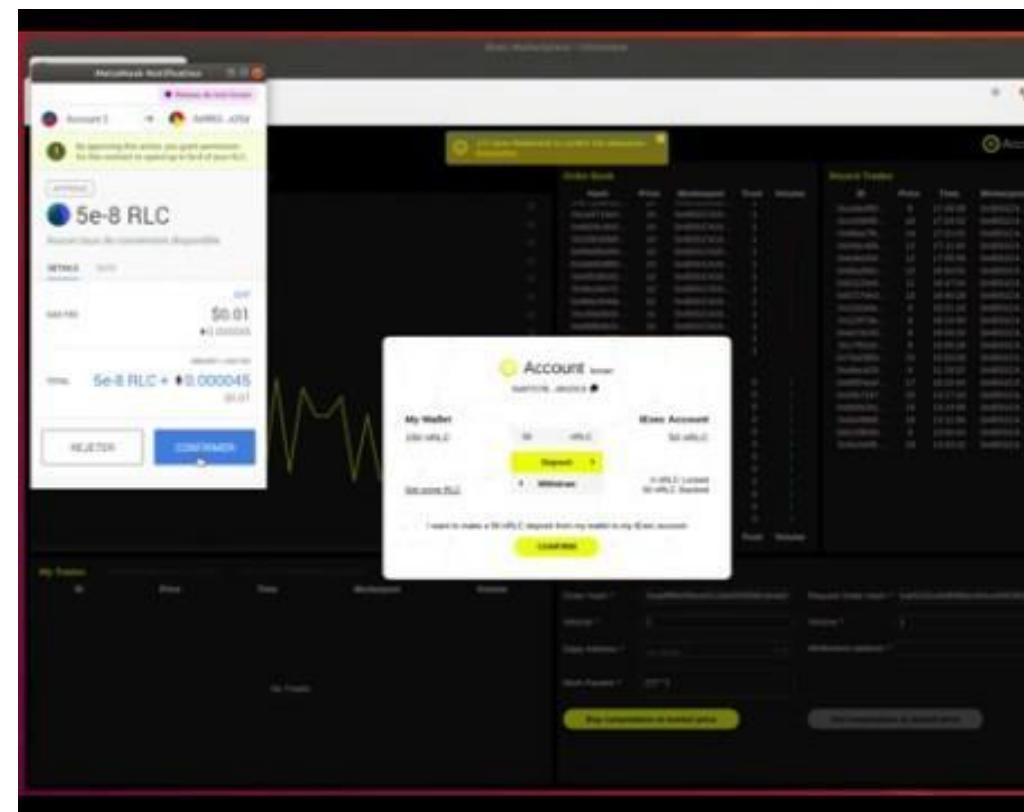
- Get your wallet private key from the SDK
 > `iexec wallet show --show-private-key`
- Copy the private key from the stdout
- Open Metamask in your browser and import your key

View your iExec account in the browser

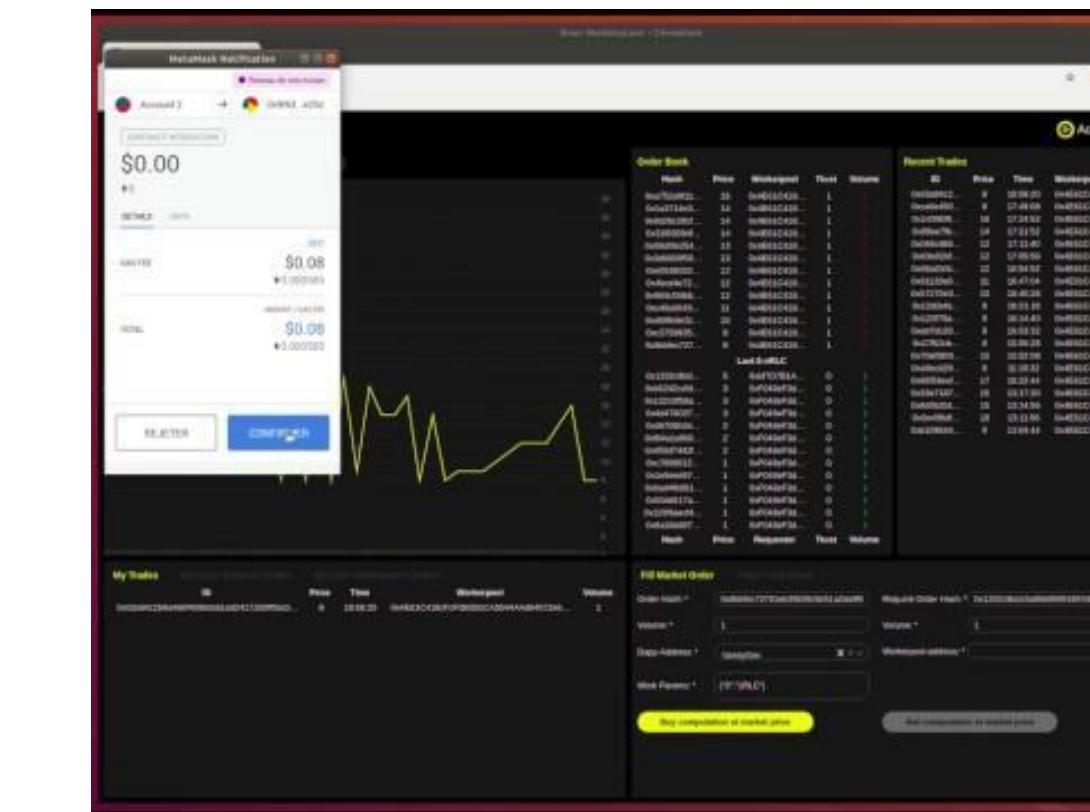
- Go to <https://v3.market.iex.ec>
- Unlock Metamask
- Switch Metamask on kovan network
- Click Account to manage your account
- Make sure you have some nRLC stacked

Buy computation on iExec marketplace

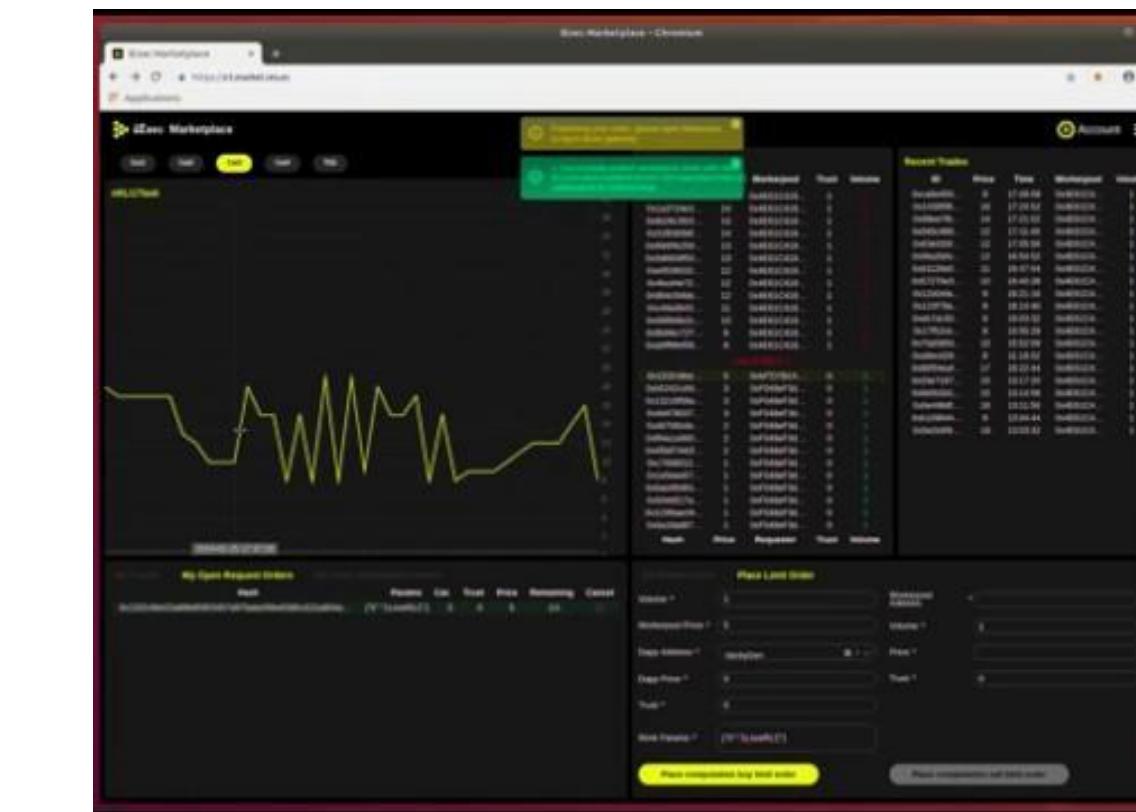
Top up your account



Buy at the market



Place limit order



Marketplace v3-alpha is live: <https://v3.market.iex.ec>



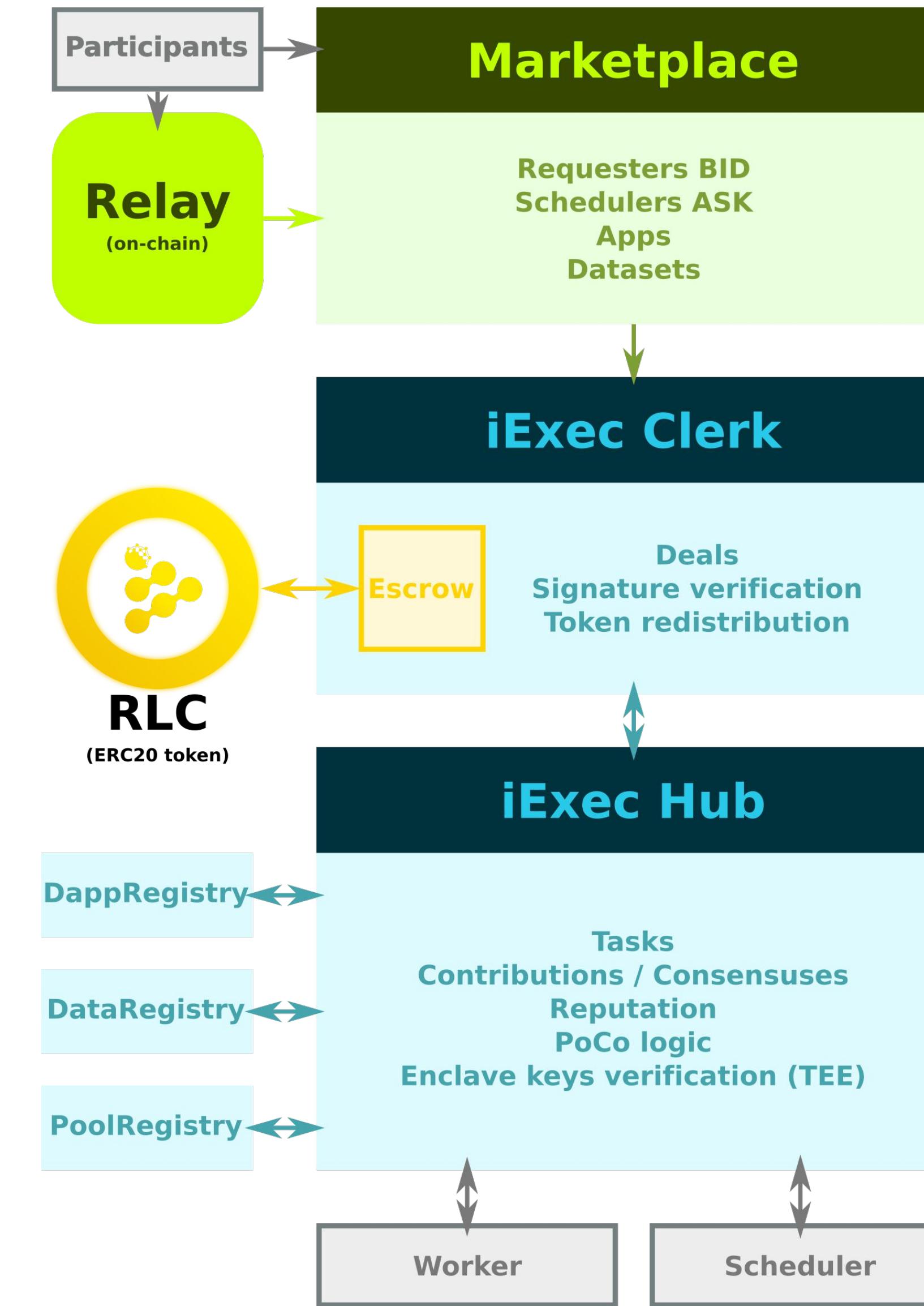
Below the Deck

Hadrien Croubois



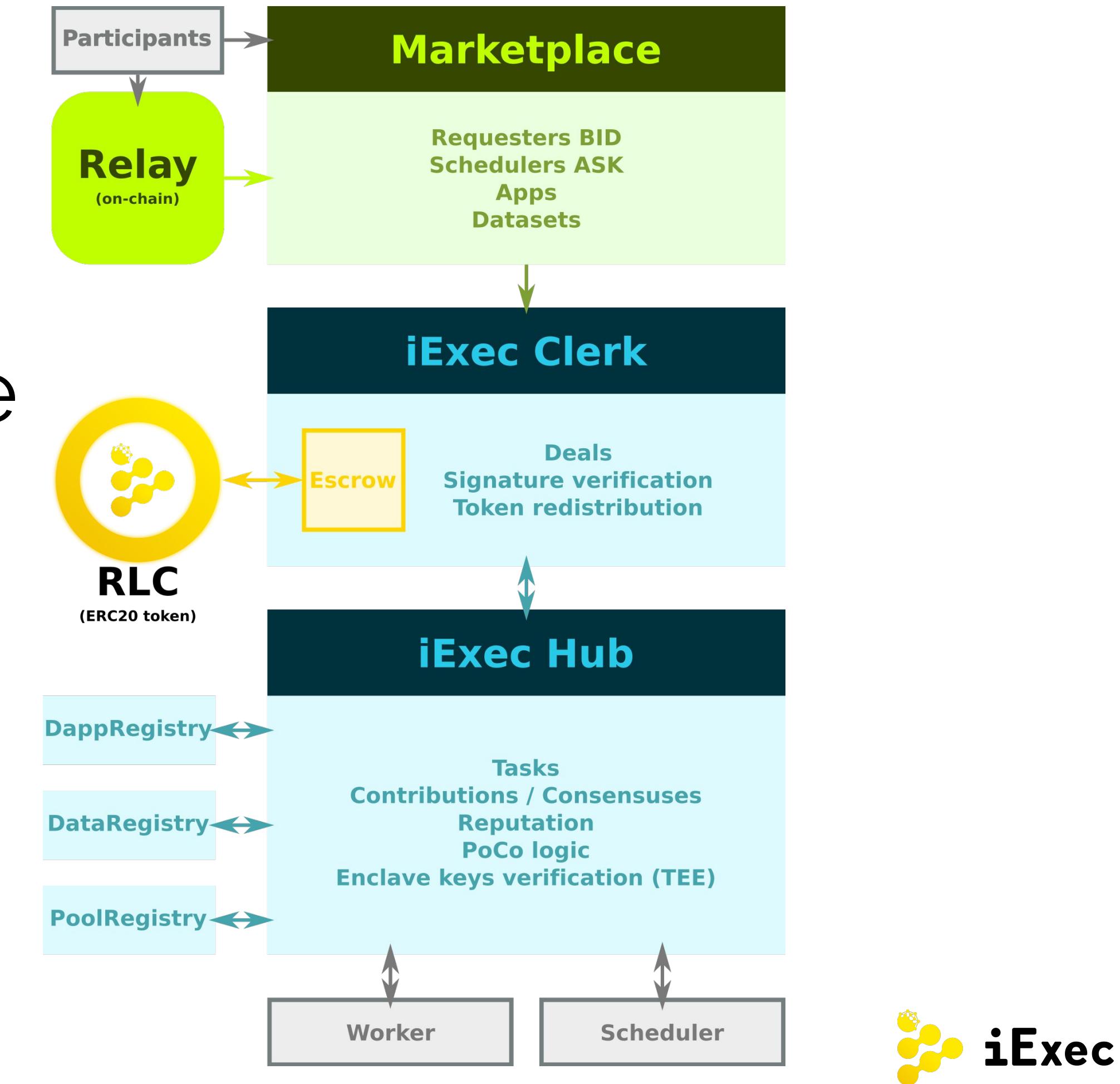
Below the Deck

- Off-chain Marketplace
- On-chain Clerk
 - Escrow
 - Deals
- On-chain Hub
 - Tasks
 - PoCo consensus
 - Workers reputation



Requester Smart Contracts

- SCs cannot sign message
 - SCs can send a transaction to the Clerk to “presign” an order.
- SCs cannot use the SDK
 - SCs can use an on-chain Relay.



Below the Deck: PoCo

Proof-of-contribution

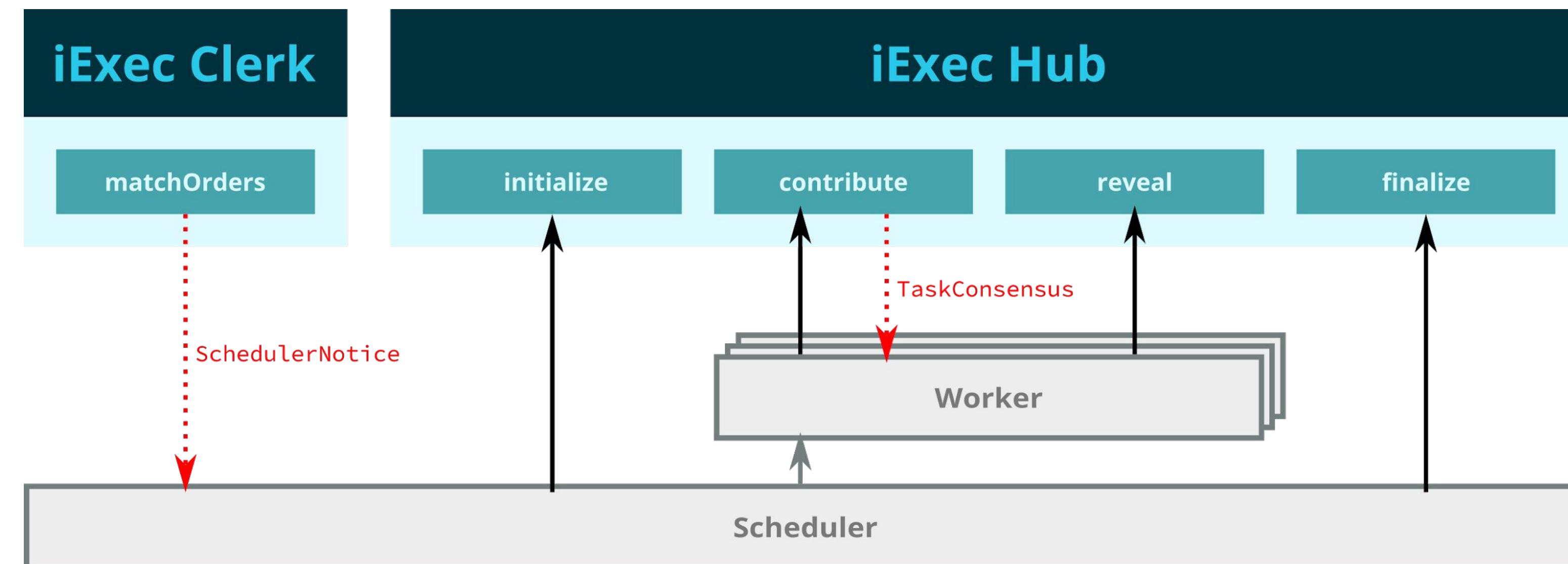
Objective: Certifying result correctness

Solution: A protocol based on game theory

Incentives provided by staking and reputation

Customizable trust parameters

Below the Deck: PoCo



Below the Deck: PoCo

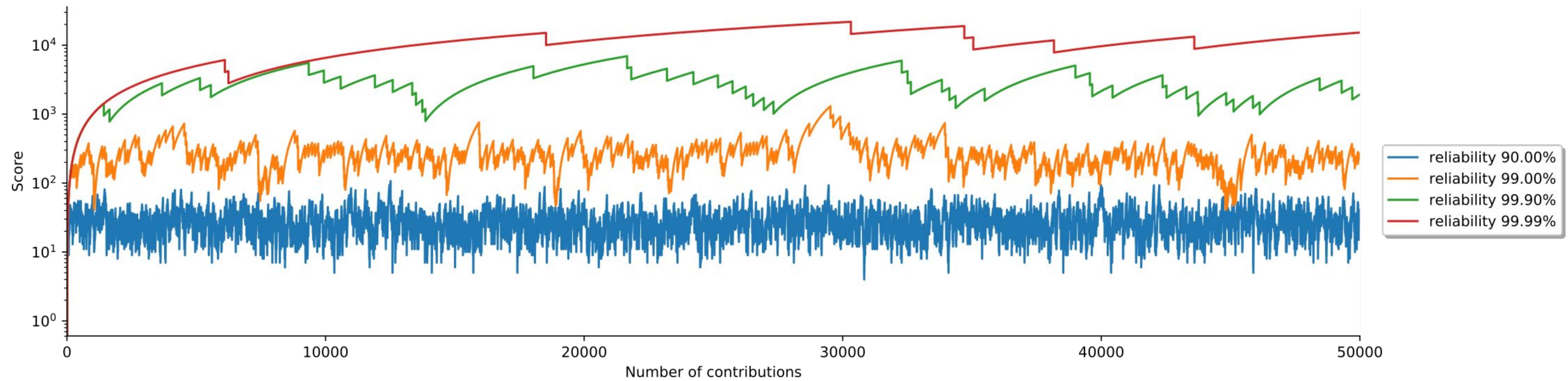
How is a consensus detected by the PoCo?

Start by the workers' reputation

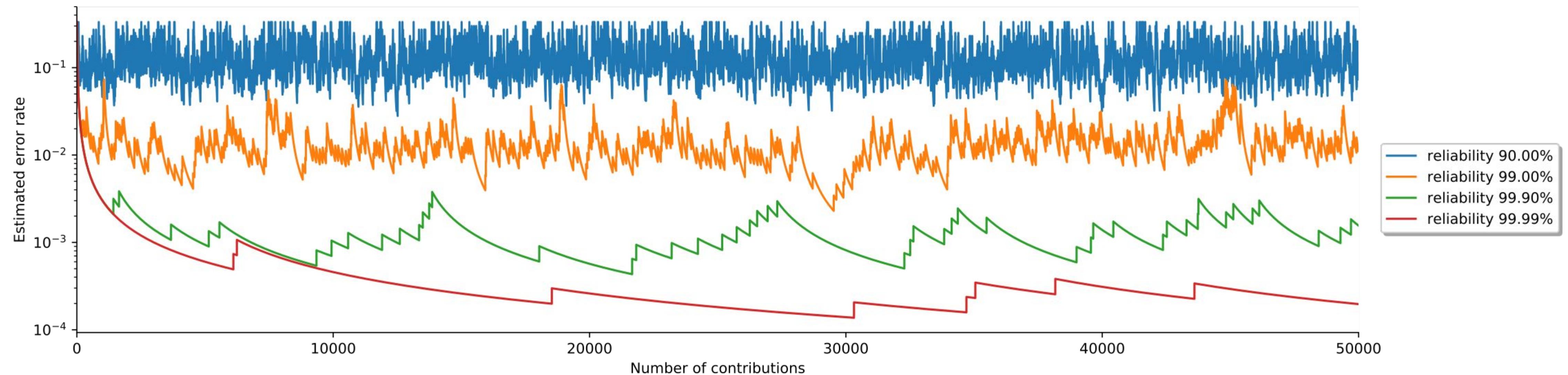
Score

- +1 for (replicated) valid contribution;
- 33% of the worker's score for bad contributions.

Below the Deck: Worker reputation



Below the Deck: Worker reputation



Below the Deck: PoCo

How is a consensus detected by the PoCo?

Combine workers' reputation into results' likelihood.

Replicate until QoS threshold is reached.

Below the Deck: PoCo

Validity domain

Workers are selected randomly;

No-one controls more than 51% of the workerpool;

Application can produce a deterministic result digest.

Below the Deck: TEE support

The PoCo also supports TEE

Complementary to the trust-based verification;

Data security & End-to-End encryption;

Works with non deterministic applications.

Data security with iExec E2E Trusted Execution

Charles de Mauroy



Challenges

- Decentralized cloud computing marketplace: everyone can participate, data and result on public repositories
- Confidentiality of data and outputs is a big concern for users
- How to ensure confidentiality, but stay easy to use?

In-transit security with integrated encryption

- First step: encrypt your data before pushing them
- Ensure confidentiality even on a remote repository
- One single CLI command: iexec encryptedpush. Handles encryption, pushes the data to a remote repository

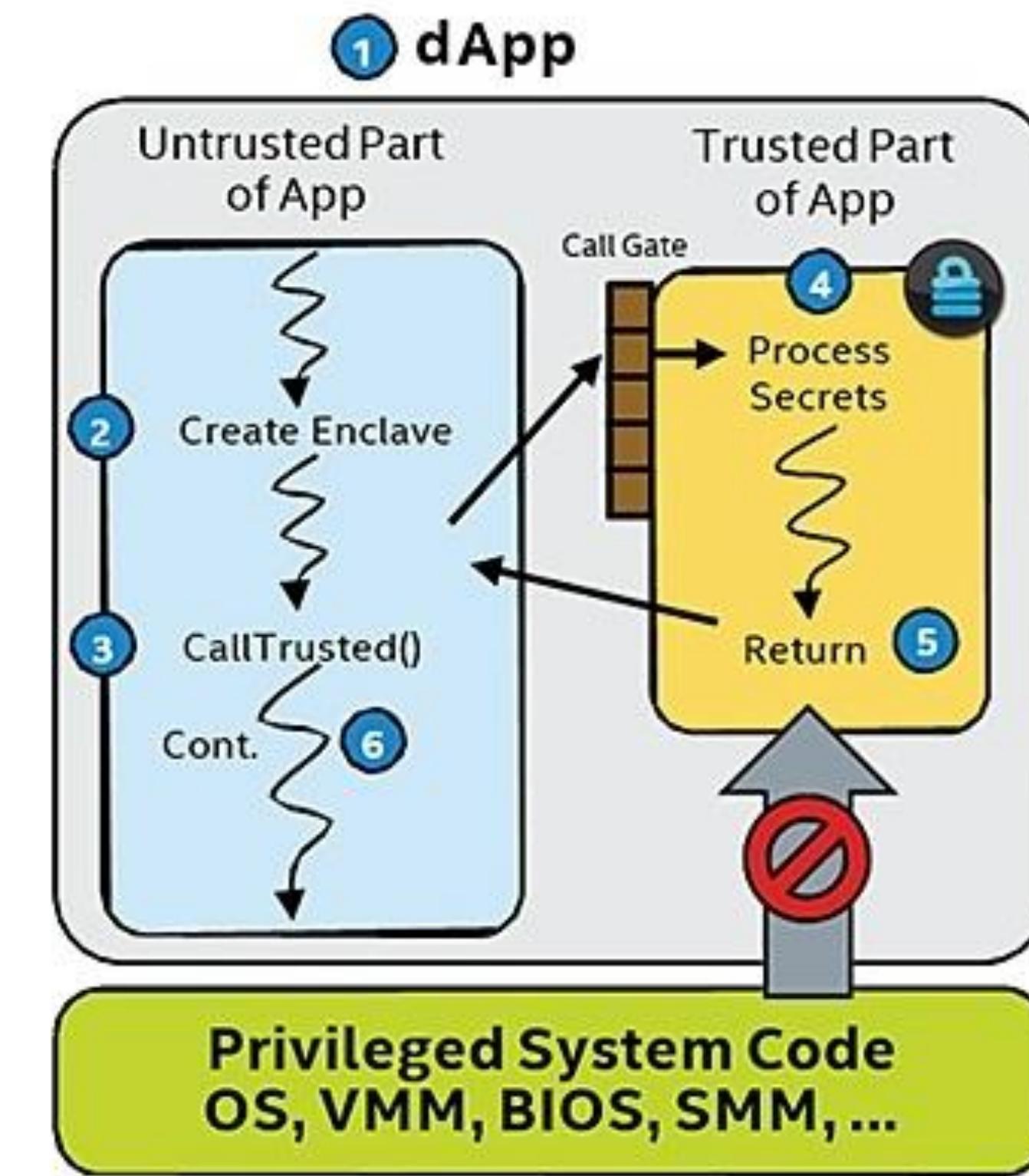
Problem: how to ensure security during computation?

- Decentralized setting: the machine owner's may have bad intention, or simply be curious
- Yet at some point, you need to decrypt the data to use them
- **Our saviour: Trusted Execution with Intel SGX!**



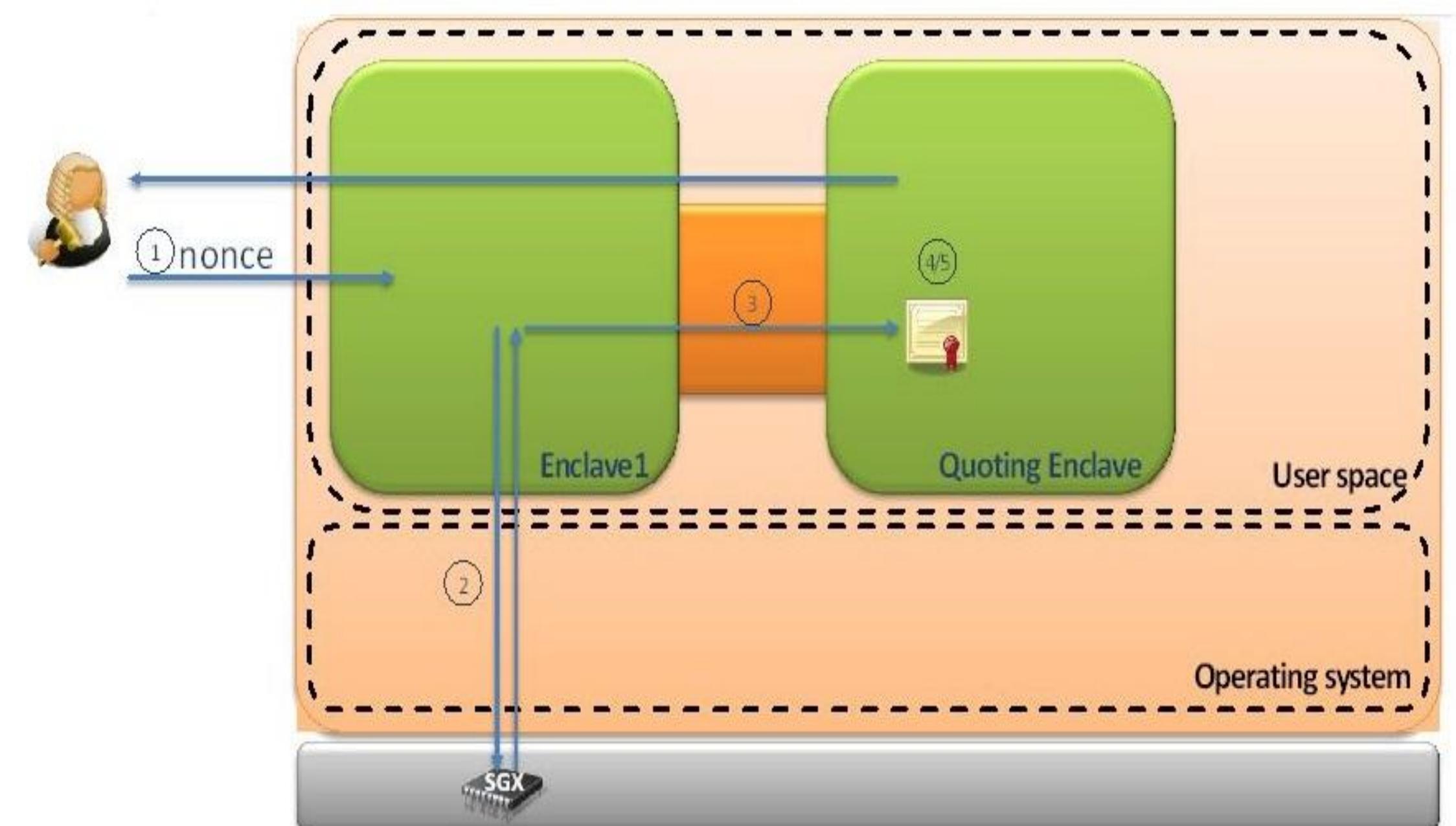
Intel SGX: what it is and how we use it (1)

- Set of CPU instructions to create enclaves in RAM, that no one can access - except code from the enclave itself
- Ensures total confidentiality of data during computation - decryption happens only inside the enclave



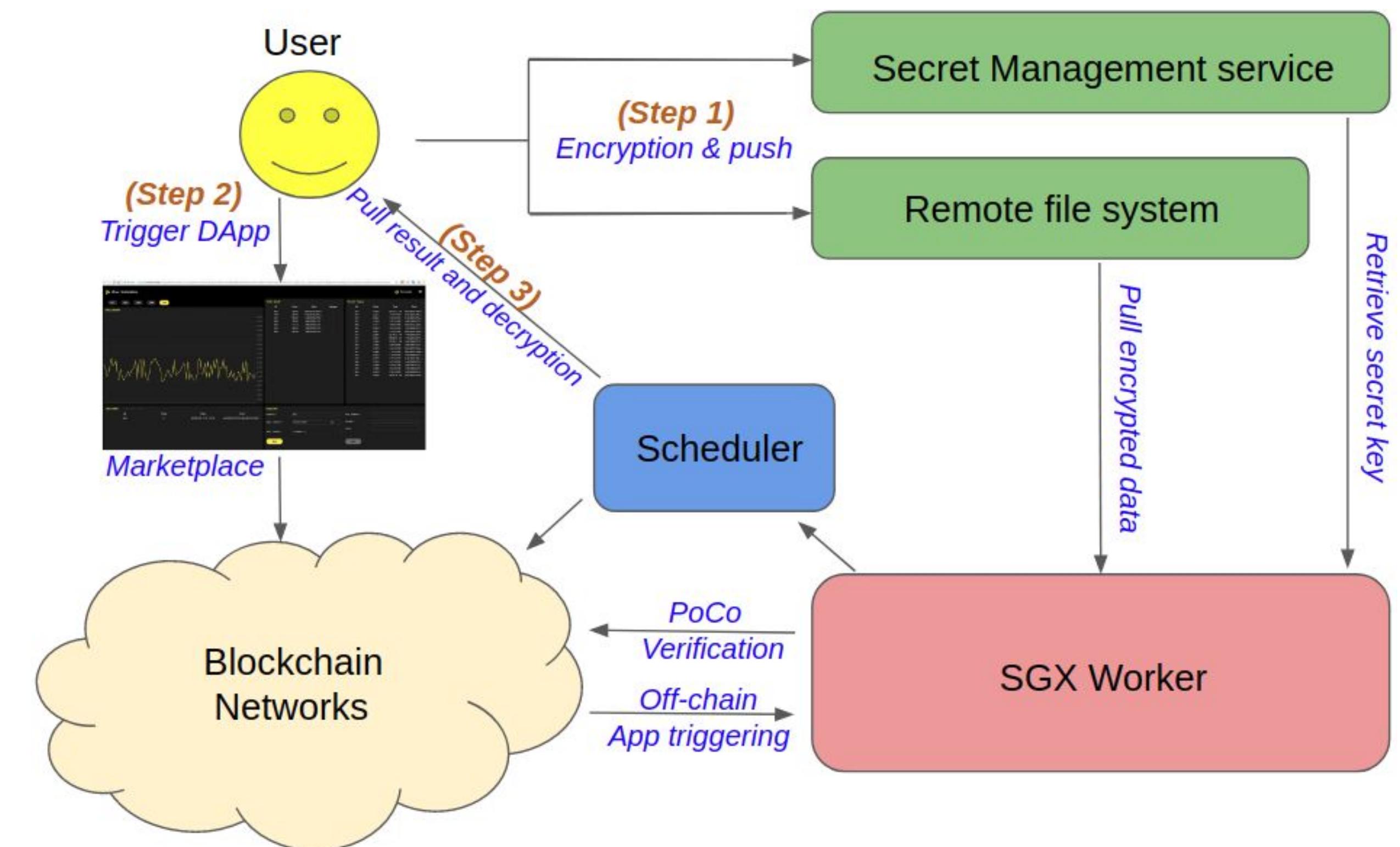
Intel SGX: what it is and how we use it (2)

- A crucial feature: **remote attestation**
- Cryptographic proof that the code inside the enclave is the one expected
- Ensures we trust the data to the right code
- Ensures the results are correct.



Fully integrated with smart contracts/PoCo

- Secret management service performs all the checks by requesting data from smart contracts
- After computation, verification by the PoCo using Intel Attestation Service (indirectly)



What changes for the end user/computation requester?

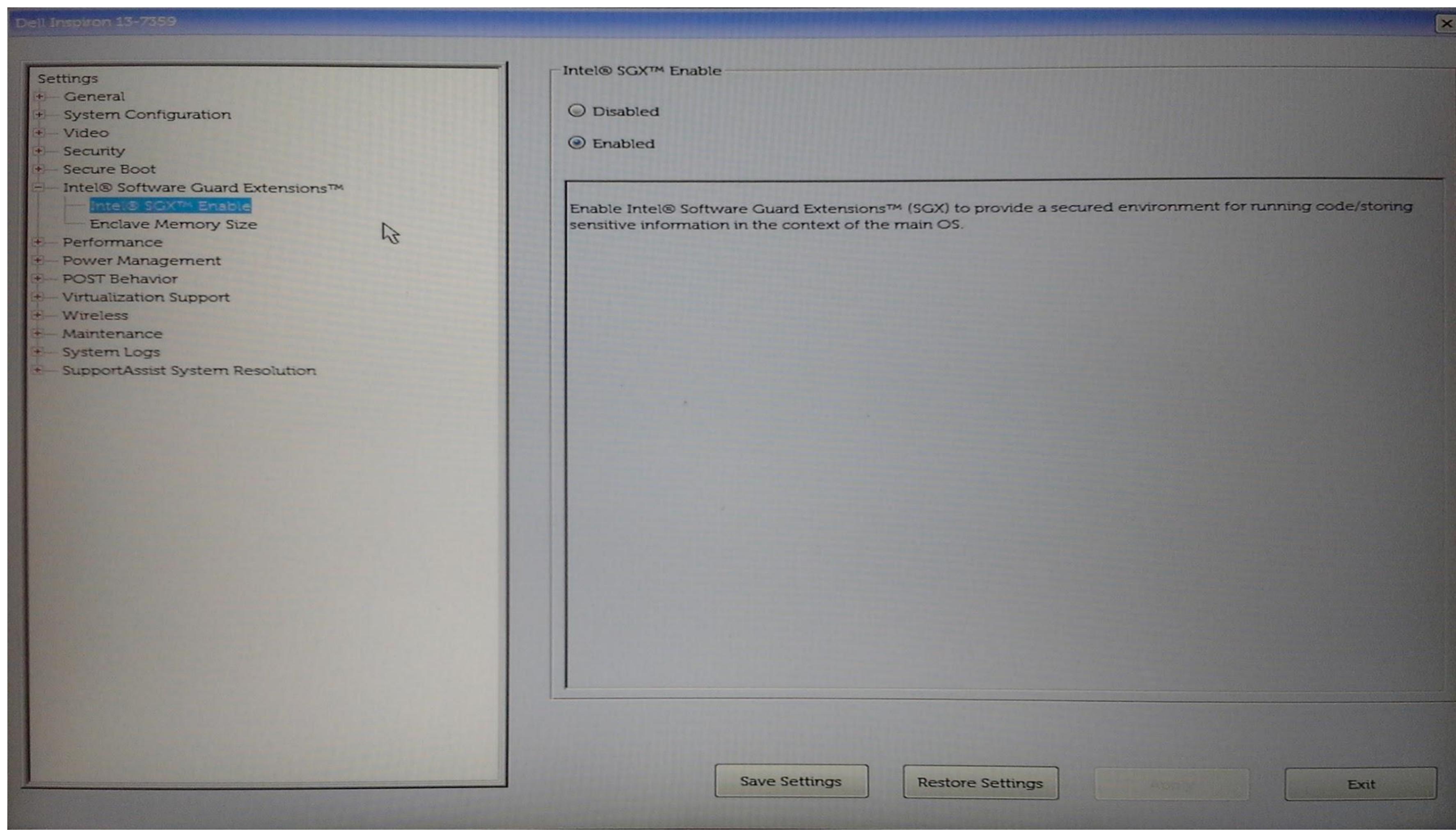
- **Nothing!**
- Just need to choose an SGX enabled application.
- Decrypt your result with a single command

What changes for the data owner?

- Nothing!
- Encrypt and push your data transparently with iExec SDK.

What changes for the worker?

- Needs to enable SGX: one setting in BIOS



What changes for the worker?

- Needs to install drivers. One command line:

```
curl -fssl https://raw.githubusercontent.com/SconeDocs/SH/master/install_sgx_driver.sh | bash
```

What changes for the dApp developer?

- Need to package runtime libraries to run native applications with good performance
- Currently working with Scone, a research project from TU Dresden.
- Easy to package with Docker: just start from one of our images

Use cases - Medical data

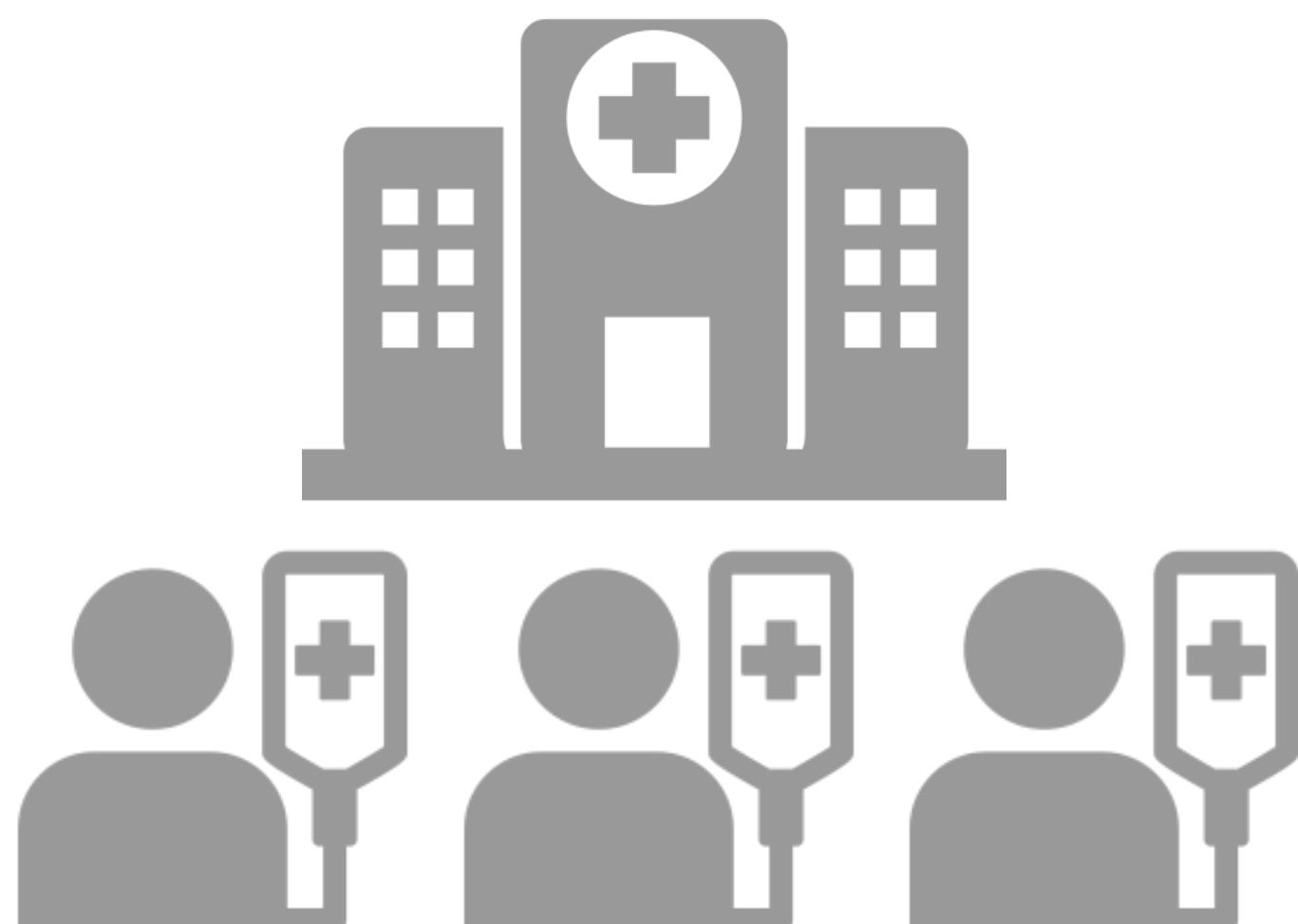
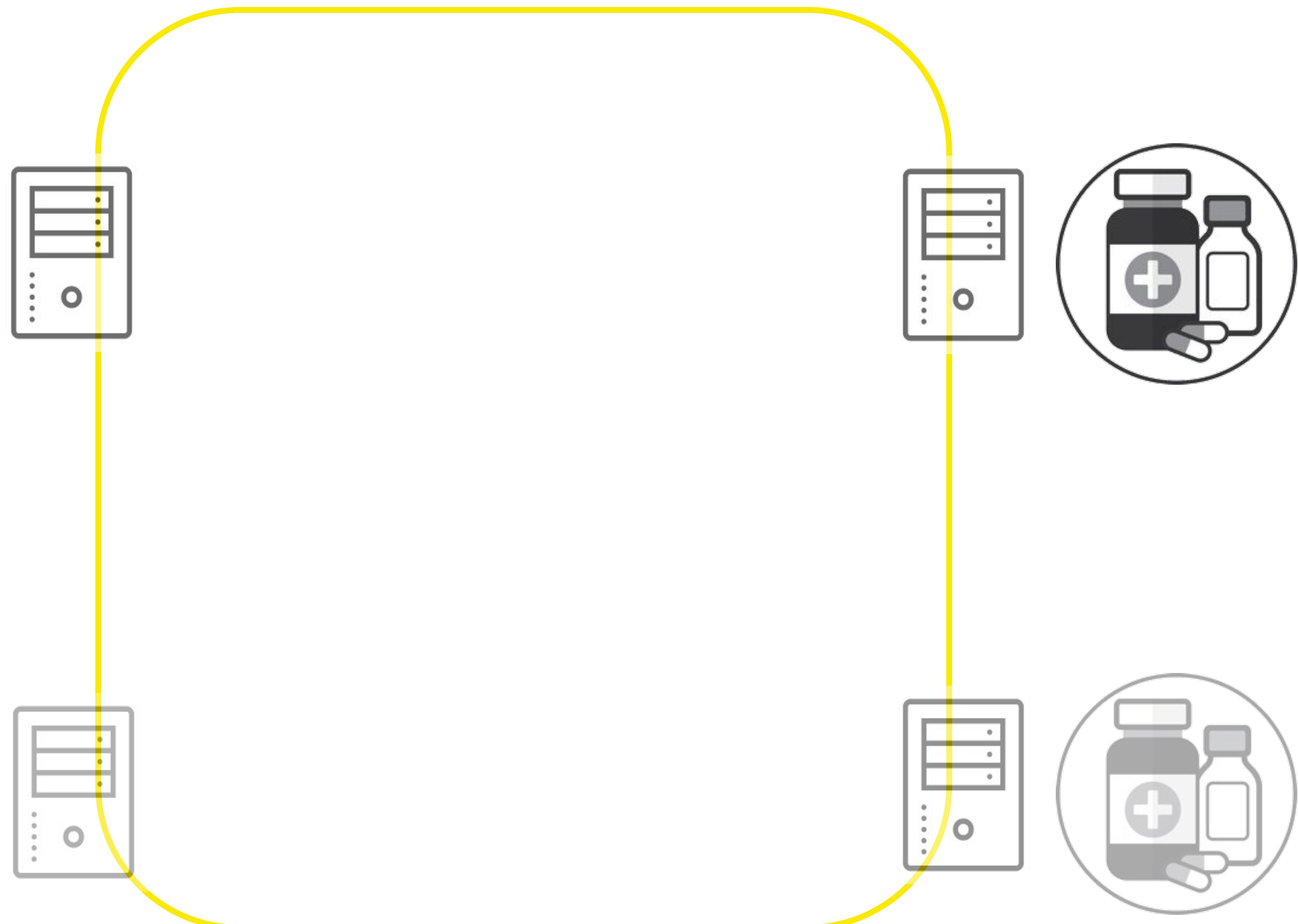
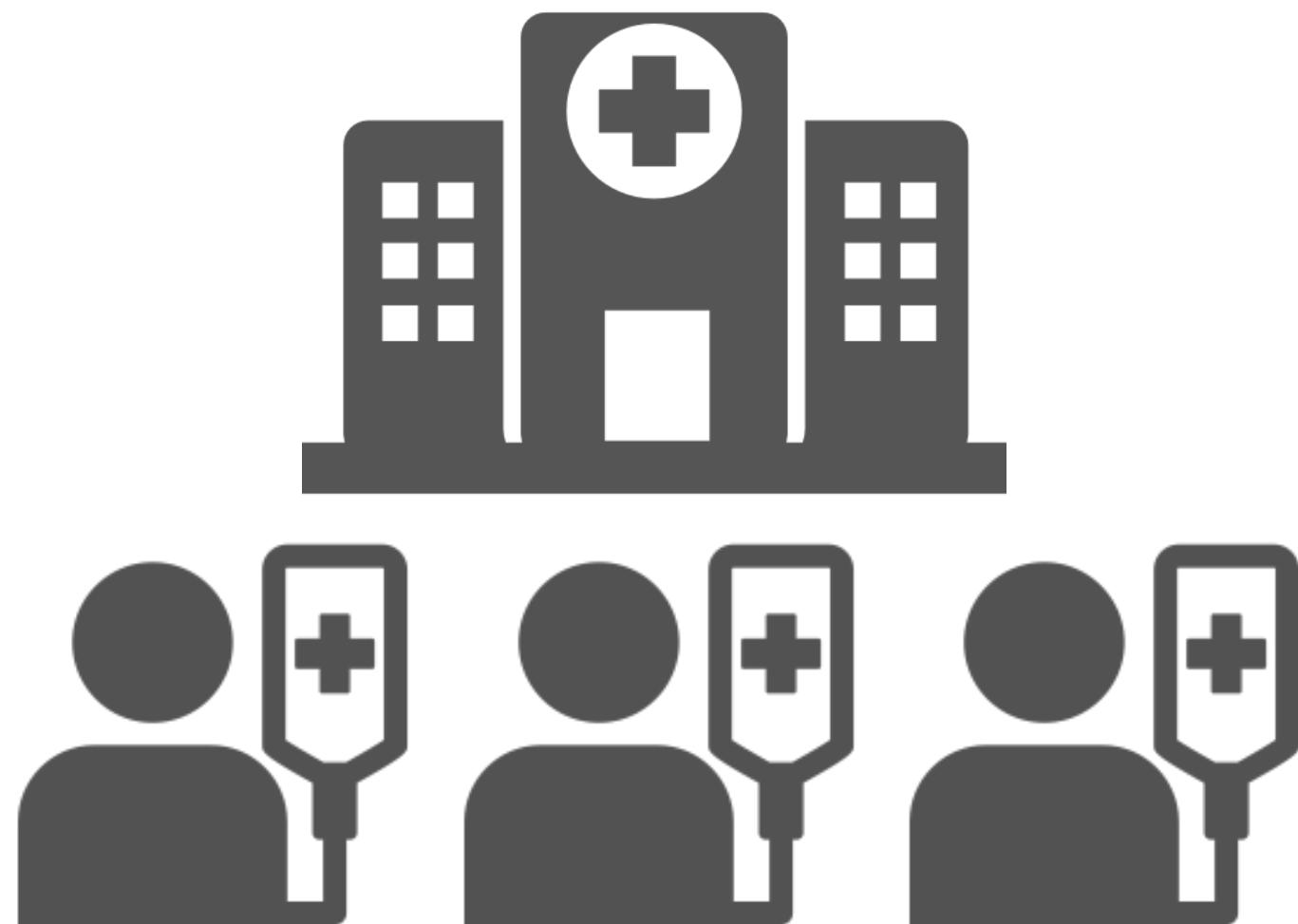
Huge market

Classic blockchain exemple

Complex legal environment

Hospitals and patients

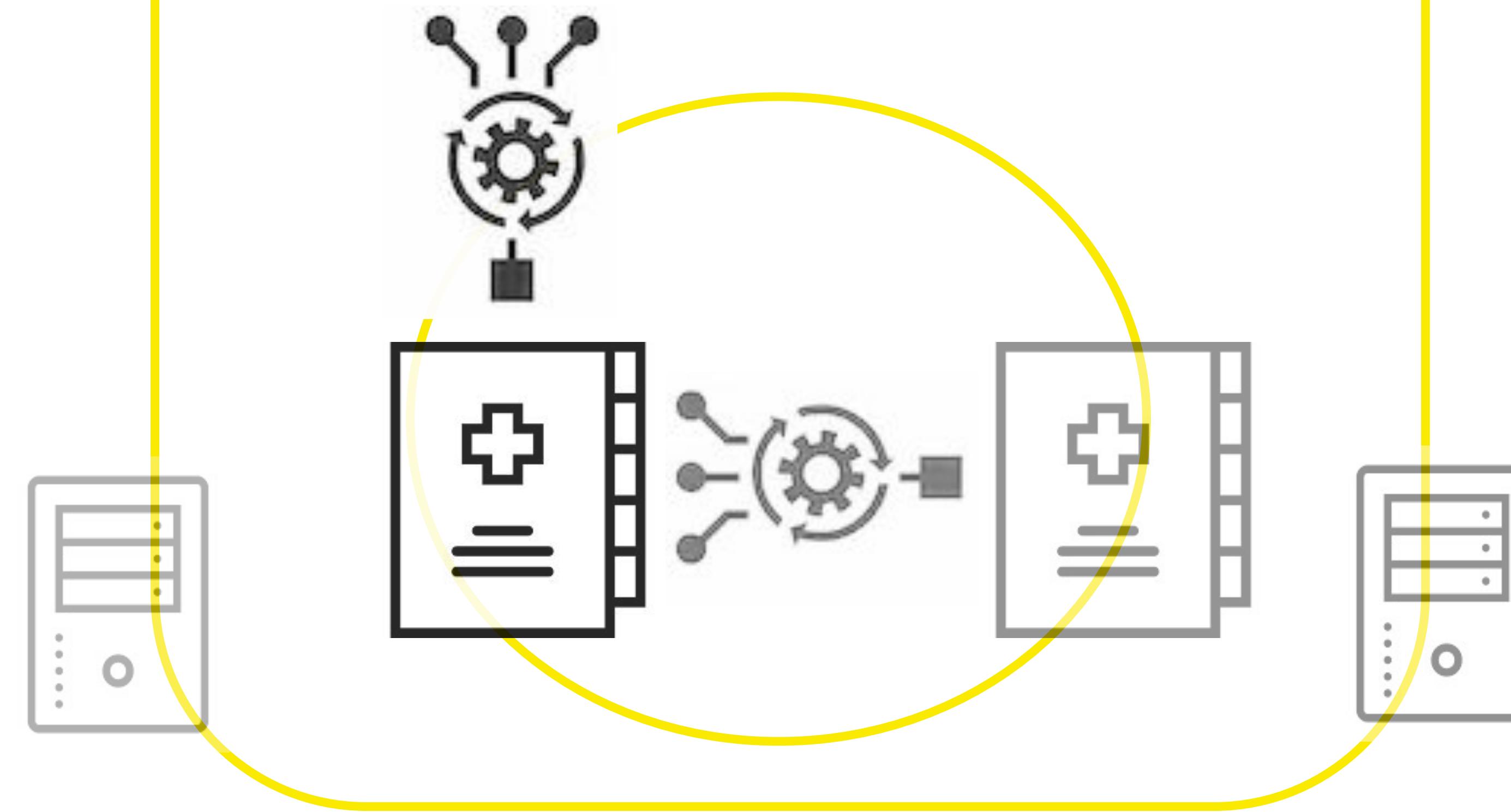
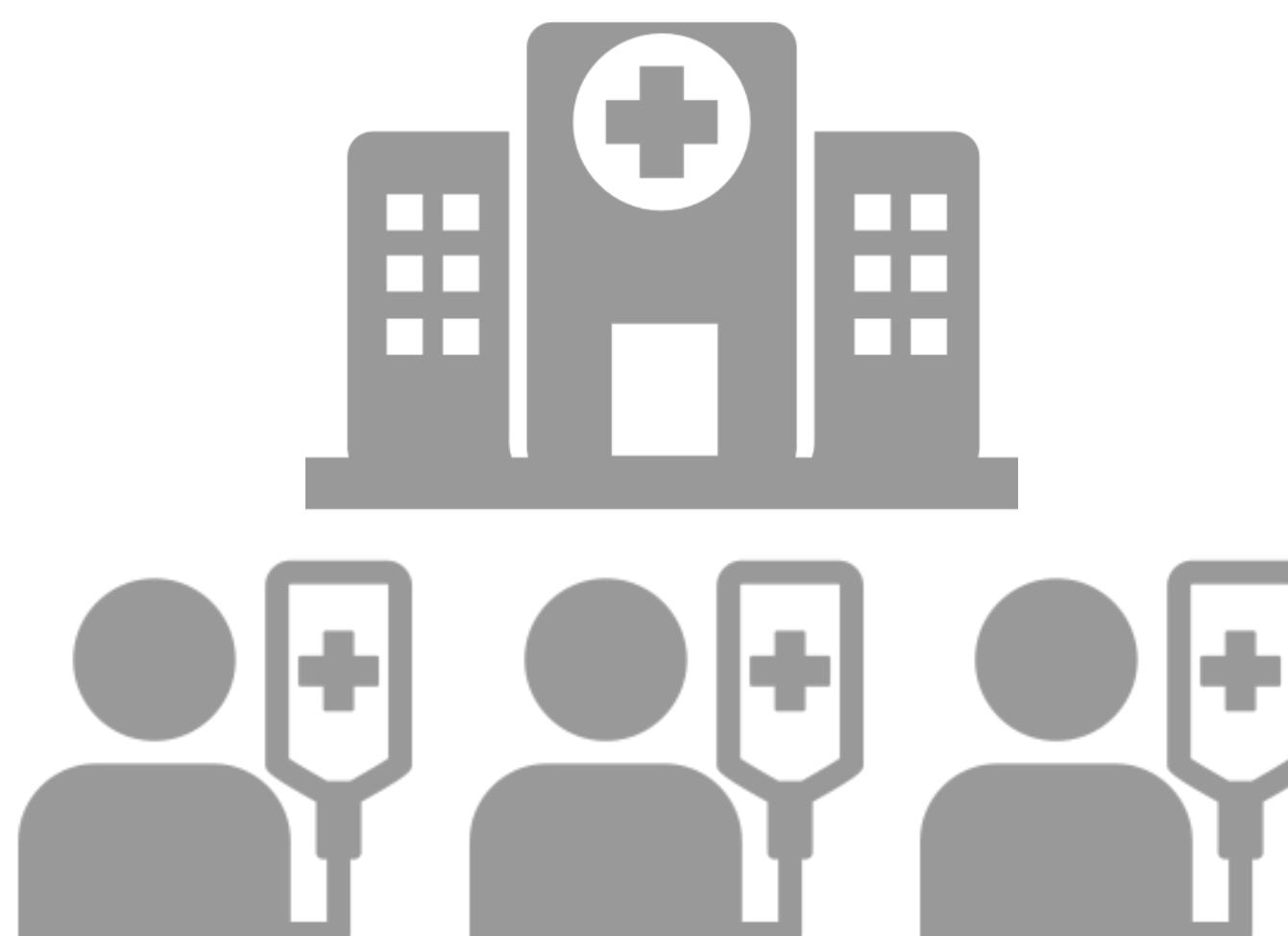
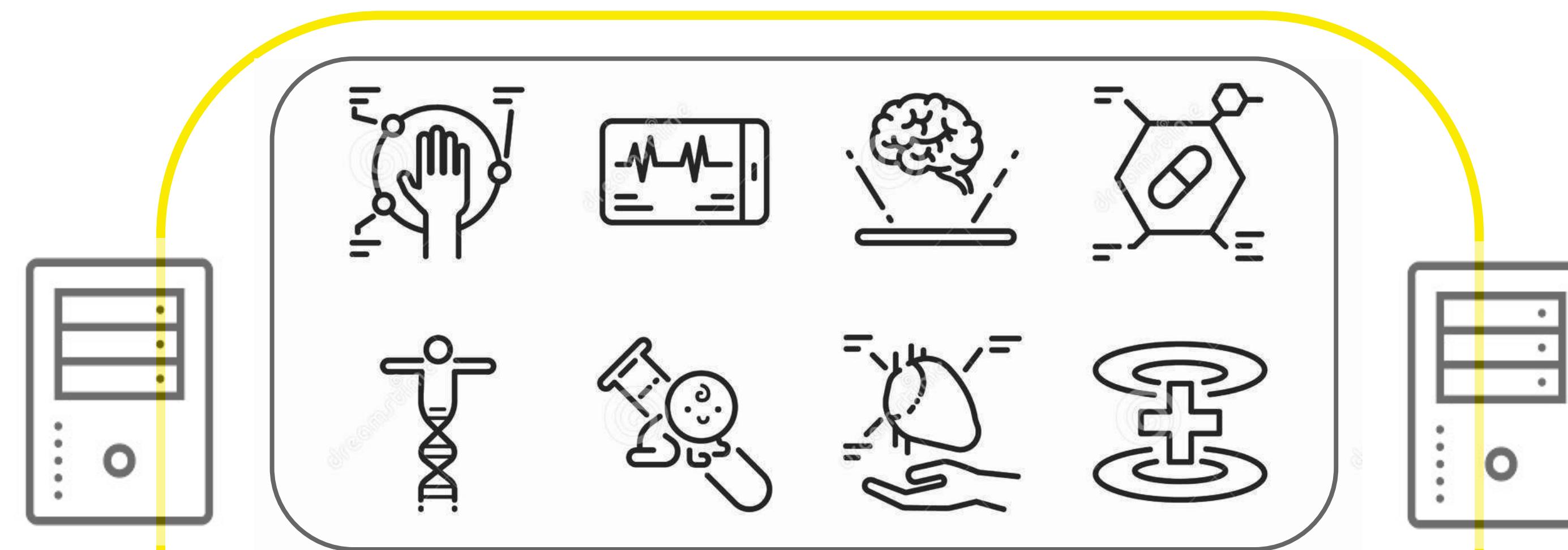
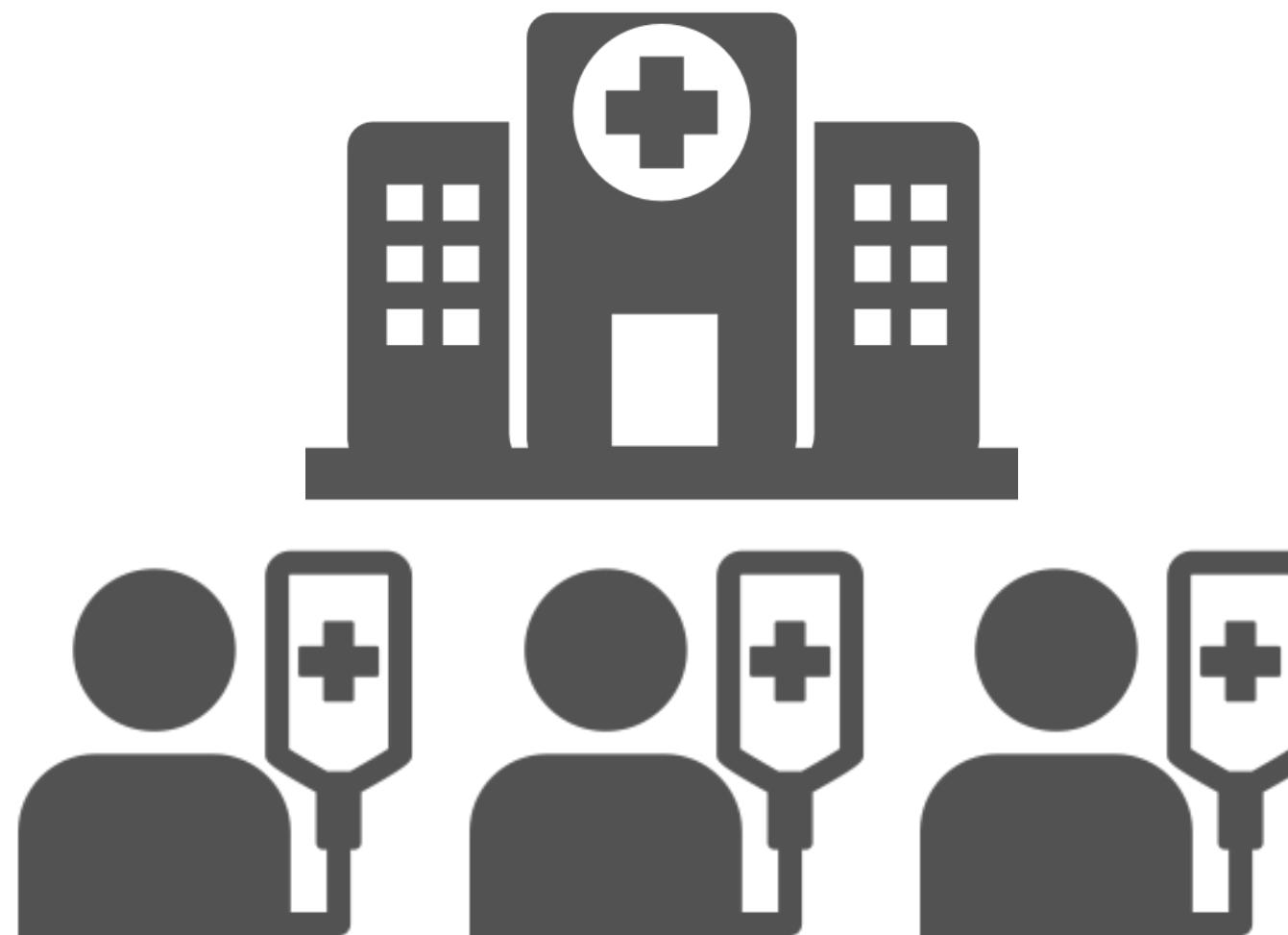
Pharmaceutical companies



Hospitals and patients

Blockchain + Intel SGX + Distributed Computing

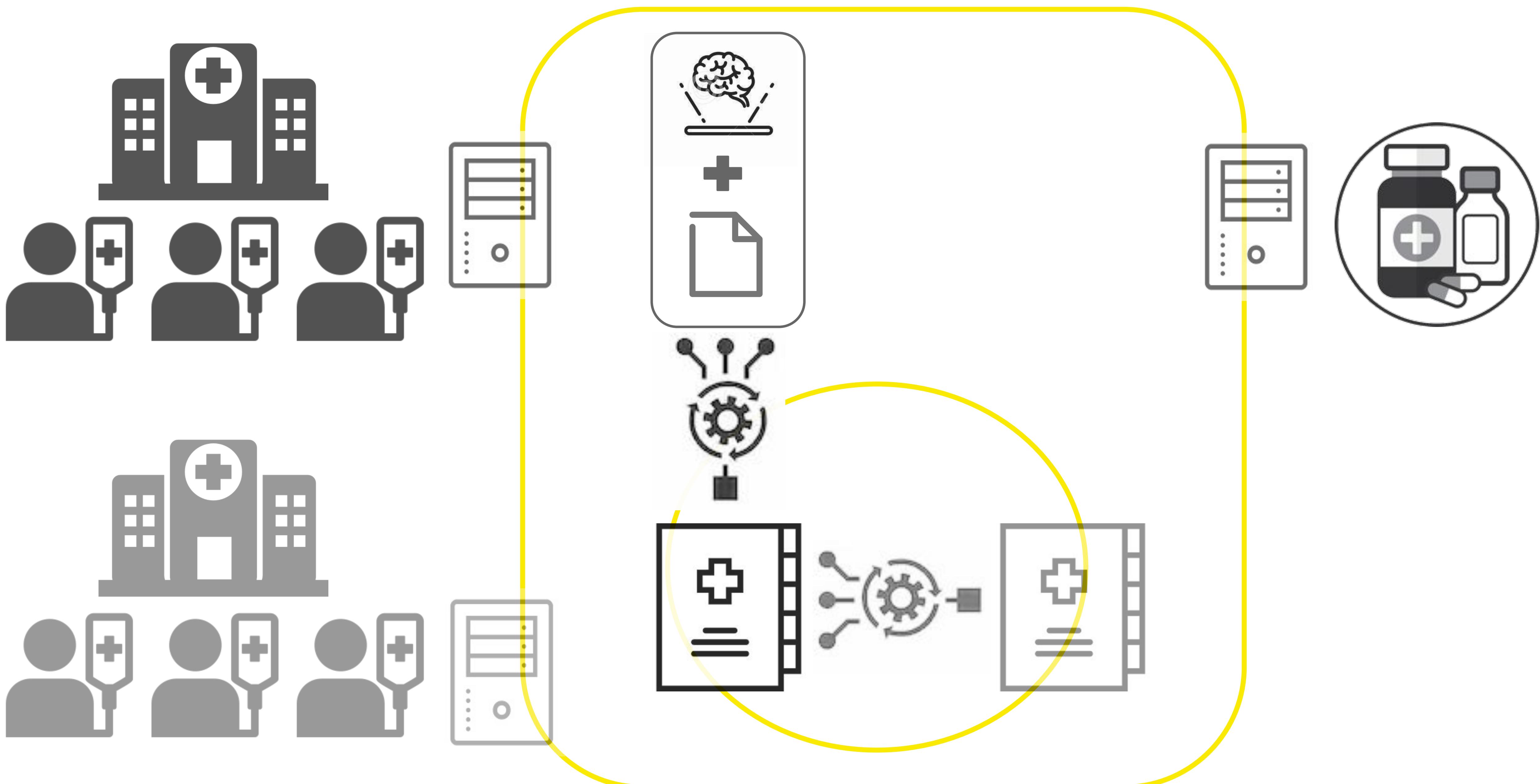
Pharmaceutical companies



Hospital A & B and patients

Blockchain + Intel SGX + Distributed Computing

Pharmaceutical
companie A



iExec flow

Business Iteration A: **Data renting for AI training**

- Requester = Pharmaceutical company A
- Data owner = Hospitals A & B and patients
- Dapps owner = Pharmaceutical company A
- Worker pool = Hospitals A & B

iExec flow

Business Iteration A: **Data renting for AI training**

- Requester = Pharmaceutical company A
- Data owner = Hospitals A & B and patients
- Dapps owner = Pharmaceutical company A
- Worker pool = Hospitals A & B

Business Iteration B: **AI renting**

- Requester = Pharmaceutical company B OR/AND Hospital C OR/AND Hospitals A & B
- Dapps owner = Pharmaceutical company A
- Data owner = requester
- Worker pool = Intel SGX Cloud

iExec flow

Business Iteration C: **AI reinforcement**

- Requester = Pharmaceutical company A
- Data owner for reinforcement of the AI model: Hospitals A & B and patients
- Dapps owner = Pharmaceutical company A
- Worker pool = Hospitals A & B

iExec flow

Business Iteration C: **AI reinforcement**

- Requester = Pharmaceutical company A
- Data owner for reinforcement of the AI model: Hospitals A & B and patients
- Dapps owner = Pharmaceutical company A
- Worker pool = Hospitals A & B

Business Iteration D: **AI renting**

- Requester = Pharmaceutical company B OR/AND Hospital C OR/AND Hospitals A & B
- Dapps owner = Pharmaceutical company A
- Data owner = requester
- Worker pool = Intel SGX Cloud

iExec flow

Business Iteration C: **AI reinforcement**

- Requester = Pharmaceutical company A
- Data owner for reinforcement of the AI model: Hospitals A & B and patients
- Dapps owner = Pharmaceutical company A
- Worker pool = Hospitals A & B

Business Iteration D: **AI renting**

- Requester = Pharmaceutical company B OR/AND Hospital C OR/AND Hospitals A & B
- Dapps owner = Pharmaceutical company A
- Data owner = requester
- Worker pool = Intel SGX Cloud

iExec Network effect



Conclusion

Decentralized Market for Computing Resources

Provide a secure solution to trade:

- Datasets
- Dapps
- Computing power

V3 alpha now - V3 in May 2019



Key ideas

For developers

- **easy to deploy** decentralized unstoppable apps
- **low maintenance** of infrastructure
- **monetization** of computing resources
- **feeless** and freedom of pricing
- **off-chain solution** with high level of trust for decentralized applications

Key ideas

For developers

- **easy to deploy** decentralized unstoppable apps
- **low maintenance** of infrastructure
- **monetization** of computing resources
- **feeless** and freedom of pricing
- **off-chain solution** with high level of trust for decentralized applications

For Businesses

- **ownership management** of computing resources
- **real-time monitoring** and auditing of transactions
- **monetization** of computing resources
- **new business paradigms**

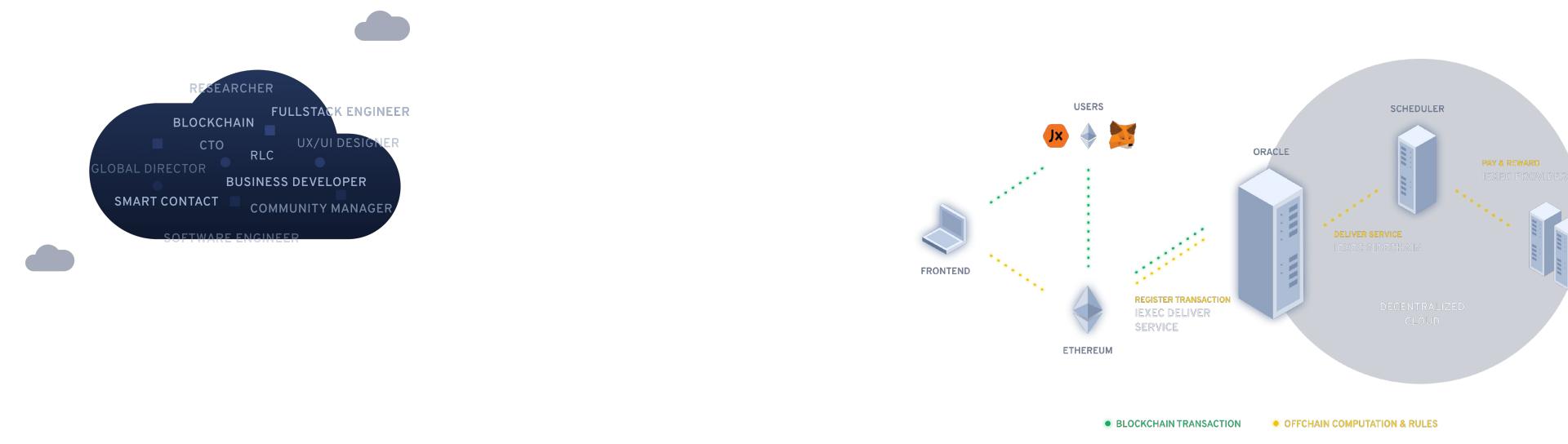
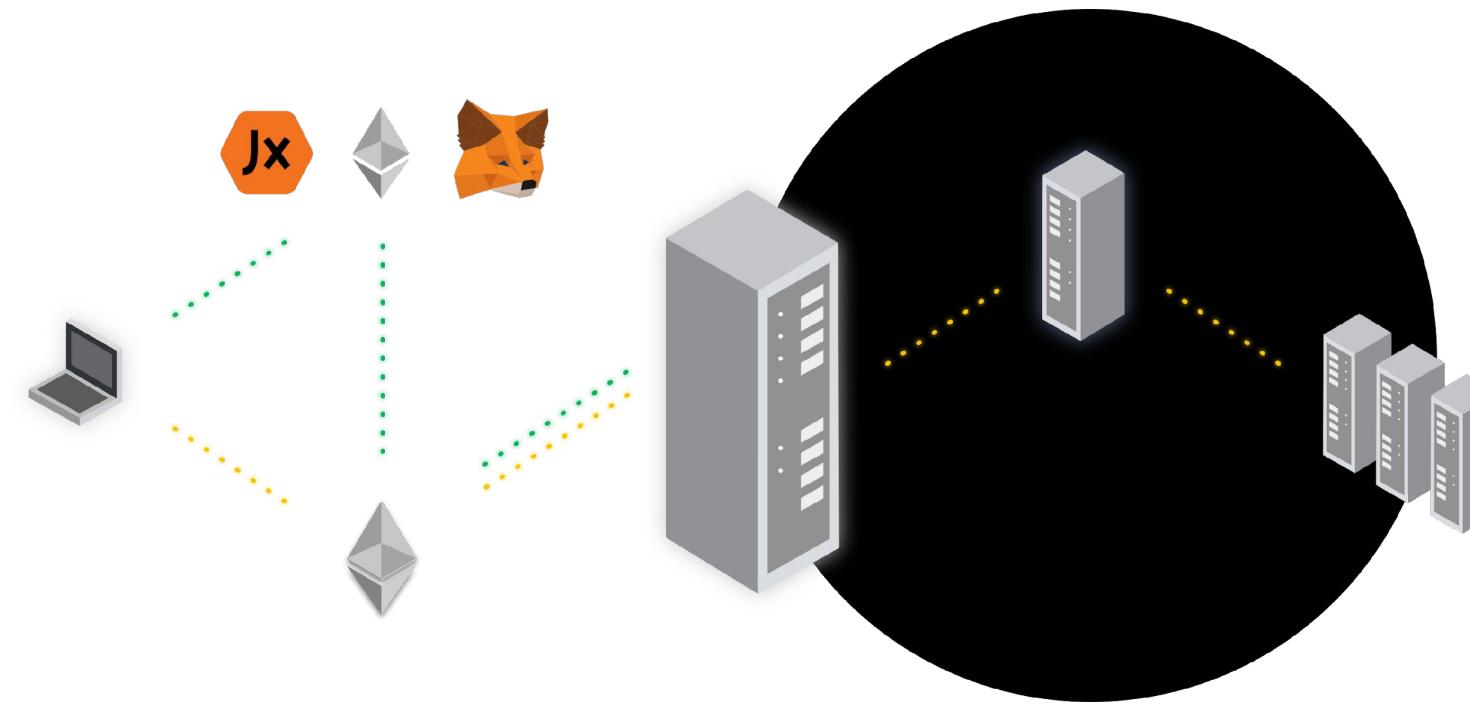
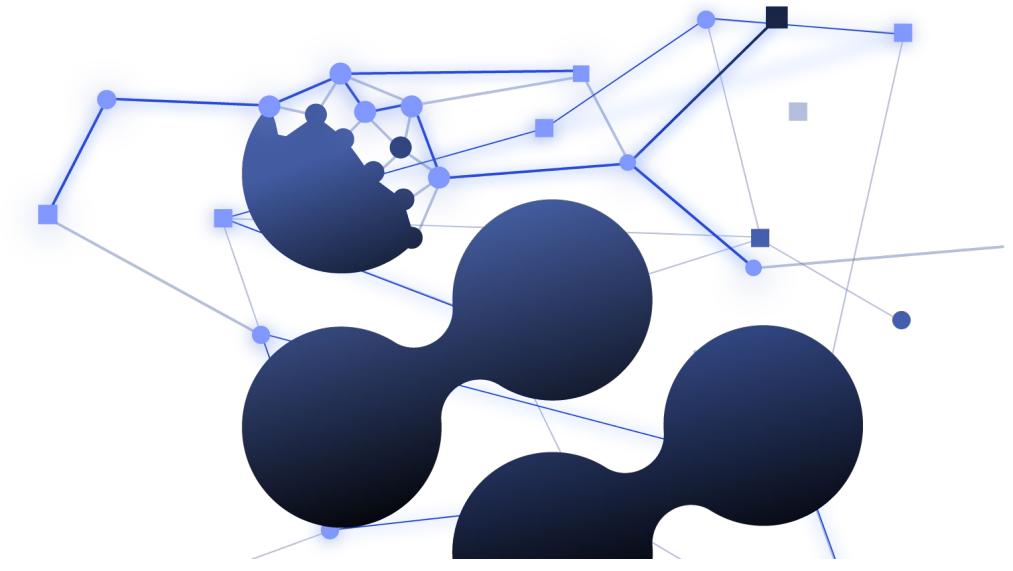
Questions

<https://iex.ec>

<https://docs.iex.ec>



Elements



Typographie Google

Raleway - Titre et Text

Slides backup JCC

Decentralized Marketplace for Computing Resources

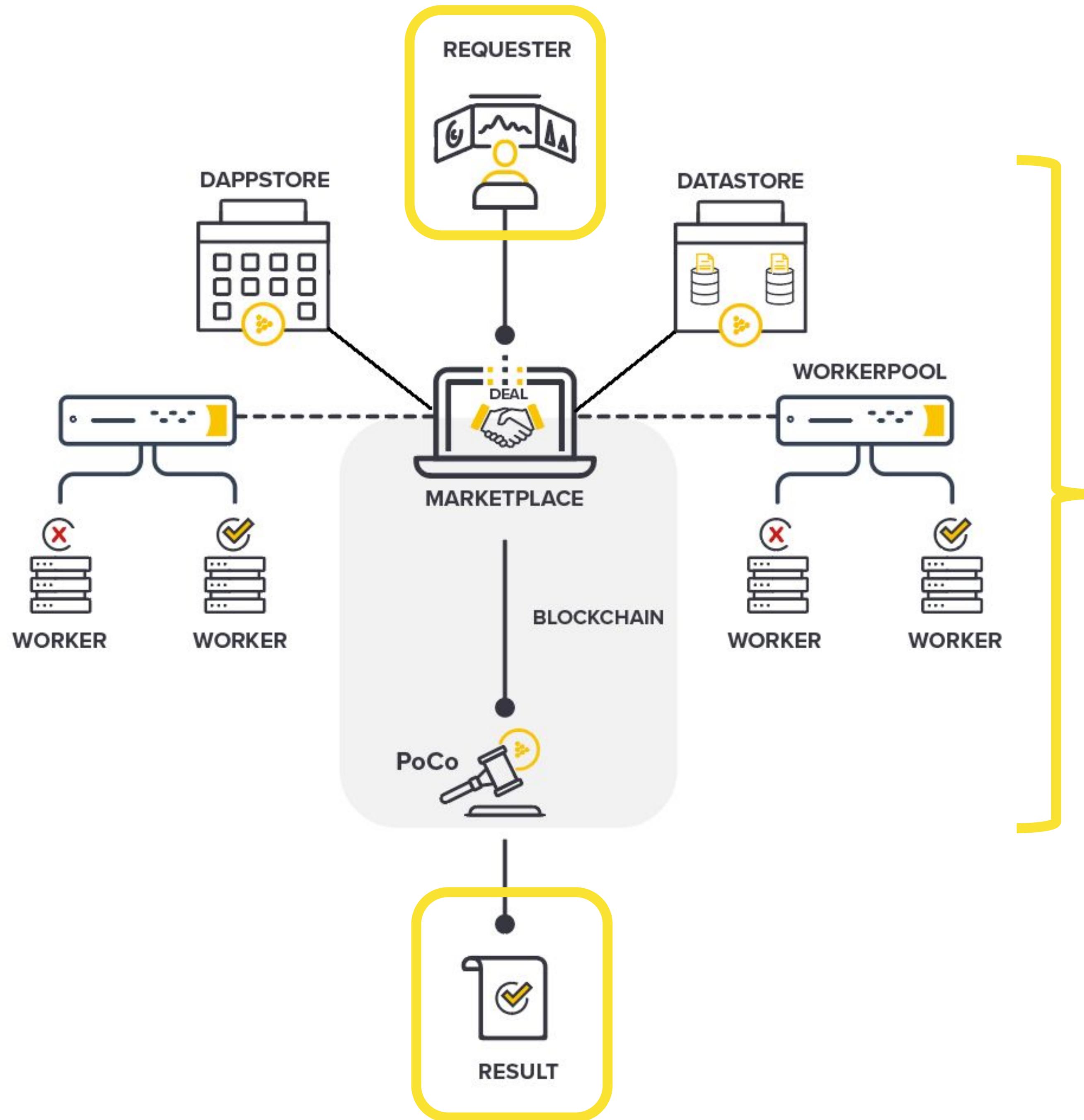
Provide a secure solution to trade:

- Datasets
- Dapps
- Computing power

Decentralized Marketplace for Computing Resources

By leveraging state of the art technologies in:

- Blockchain
- Onchain/offchain consensus
- Hardware enclaves
- Grid computing



iExec

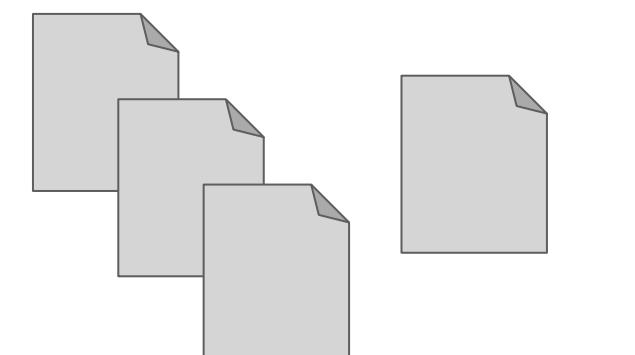
- Blockchain
- Onchain/offchain consensus
- Hardware enclaves
- Grid computing



Use cases - Med Tech

Video demo RSA2019

Explore decentralized stack with iExec explorer



match orders

Explore computing on iExec

iExec Explorer

> iexec app deploy

All Mine kovan Search address or id or txHash

Last deals ▶ 20 / 97 ▶

Time	TxHash	Deal	App	Workerpo	Price	Volume
18 hours ...	0xd592c...	0x86f54ea...	0xDfeab5...	0x4E61C4...	17 nRLC	1
21 hours ...	0xe7145...	0x33e718...	0xDfeab5...	0x4E61C4...	15 nRLC	1
21 hours ...	0xb2c4b4...	0x8d0b2b...	0xDfeab5...	0x4E61C4...	15 nRLC	1
21 hours ...	0x64f88c5...	0xe499df...	0xDfeab5...	0x4E61C4...	18 nRLC	1
a day ago	0x550de6...	0xb10964...	0xDfeab5...	0x4E61C4...	9 nRLC	1
a day ago	0x6cf86d...	0x5e2d4f9...	0xDfeab5...	0x4E61C4...	19 nRLC	1
a day ago	0x3ccb81...	0x9de7a0...	0xDfeab5...	0x4E61C4...	8 nRLC	1
a day ago	0x061b1c...	0xbfd9a9...	0xDfeab5...	0x4E61C4...	19 nRLC	1
a day ago	0x2dbaf2e...	0x344d28...	0xDfeab5...	0x4E61C4...	8 nRLC	1
a day ago	0xa2f65b5...	0x96f2c3a...	0xDfeab5...	0x4E61C4...	15 nRLC	1
2 days ago	0x6c581f9...	0x11c2af6...	0xDfeab5...	0x4E61C4...	10 nRLC	1
2 days ago	0x10f4385...	0xbc5257...	0xDfeab5...	0x4E61C4...	5 nRLC	1
2 days ago	0x06cf3c9...	0xbc0de0...	0xDfeab5...	0x4E61C4...	8 nRLC	1
2 days ago	0x0079d8f...	0x316cfa5...	0xDfeab5...	0x4E61C4...	15 nRLC	1
2 days ago	0x532e33...	0x5ee070...	0xDfeab5...	0x4E61C4...	15 nRLC	1
2 days ago	0xe65bfca...	0x1a619b...	0xDfeab5...	0x4E61C4...	18 nRLC	1
3 days ago	0xda9c5b...	0xd2170a...	0xDfeab5...	0x4E61C4...	10 nRLC	1
3 days ago	0xe26311...	0x17ed3a...	0xDfeab5...	0x4E61C4...	10 nRLC	1
3 days ago	0x45da3fa...	0xefa2eeb...	0xDfeab5...	0x4E61C4...	10 nRLC	1
3 days ago	0x5a92ea...	0x123d39...	0xDfeab5...	0x4E61C4...	12 nRLC	1

Last tasks ▶ 20 / 86 ▶

Task	Deadline	Status
0xc15ee...	27/02 6:0...	COMPLETE
0xc1f4e2...	27/02 2:5...	REVEALING
0xf2bd4e...	27/02 2:5...	COMPLETE
0x3b8c3...	27/02 2:5...	COMPLETE
0xa6cb3...	27/02 2:4...	COMPLETE
0x388d3f...	27/02 2:4...	COMPLETE
0xbe413...	27/02 2:3...	COMPLETE
0xea41e...	27/02 2:3...	COMPLETE
0x261ef5...	27/02 2:3...	COMPLETE
0xa2f65b5...	27/02 12:...	COMPLETE
0x3581d...	26/02 5:0...	COMPLETE
0xed67a...	26/02 5:0...	REVEALING
0xd31dd...	26/02 2:0...	COMPLETE
0x9d1fb...	26/02 1:5...	COMPLETE
0xf91ae2...	26/02 1:4...	COMPLETE
0x07ef43...	26/02 1:4...	COMPLETE
0x3dd47...	25/02 7:1...	COMPLETE
0x85507...	25/02 7:1...	REVEALING
0xa3073...	25/02 7:1...	COMPLETE
0x7546b...	25/02 7:1...	COMPLETE

Last apps deployed ▶ 10 / 10 ▶

Time	TxHash	App	Owner	Name
10 day...	0x42c...	0x708...	0xF04...	hello...
10 day...	0xaecf...	0x444...	0xF04...	hello...
13 day...	0x620...	0xDFe...	0xF04...	vanity...
17 day...	0x096...	0x3C9...	0xF04...	vanity...
17 day...	0xd08...	0x395...	0xF04...	my-app
17 day...	0x3f50...	0xD8c...	0x000...	my-app
17 day...	0x79c...	0xE4c...	0x000...	my-app
17 day...	0x58b...	0x195...	0x000...	my-app
17 day...	0xa03...	0xD16...	0xF04...	vanityeth
21 day...	0xdd7...	0x100...	0x000...	my-app

Last datasets deployed ▶ 4 / 4 ▶

Time	TxHash	Dataset	Owner	Name
2 days...	0x880...	0x00A...	0xF04...	cat
3 days...	0xbac...	0x8A0...	0xF04...	my-dat...
3 days...	0x59d...	0xA07...	0xF04...	my-dat...
8 days...	0x163...	0xF74...	0xF04...	my-dat...

Last workerpools deployed ▶ 4 / 4 ▶

Time	TxHash	Workerpo	Owner
16 days a...	0xfc3a42...	0x803d2...	0xF048e...
17 days a...	0x9d4ae...	0xE61C...	0xdFB79...
17 days a...	0x024c7d...	0xE0bB1...	0x000a9c...
21 days a...	0x3cd3c1...	0x8a587...	0x000a9c...

Explore resources deployed on iExec

task computation

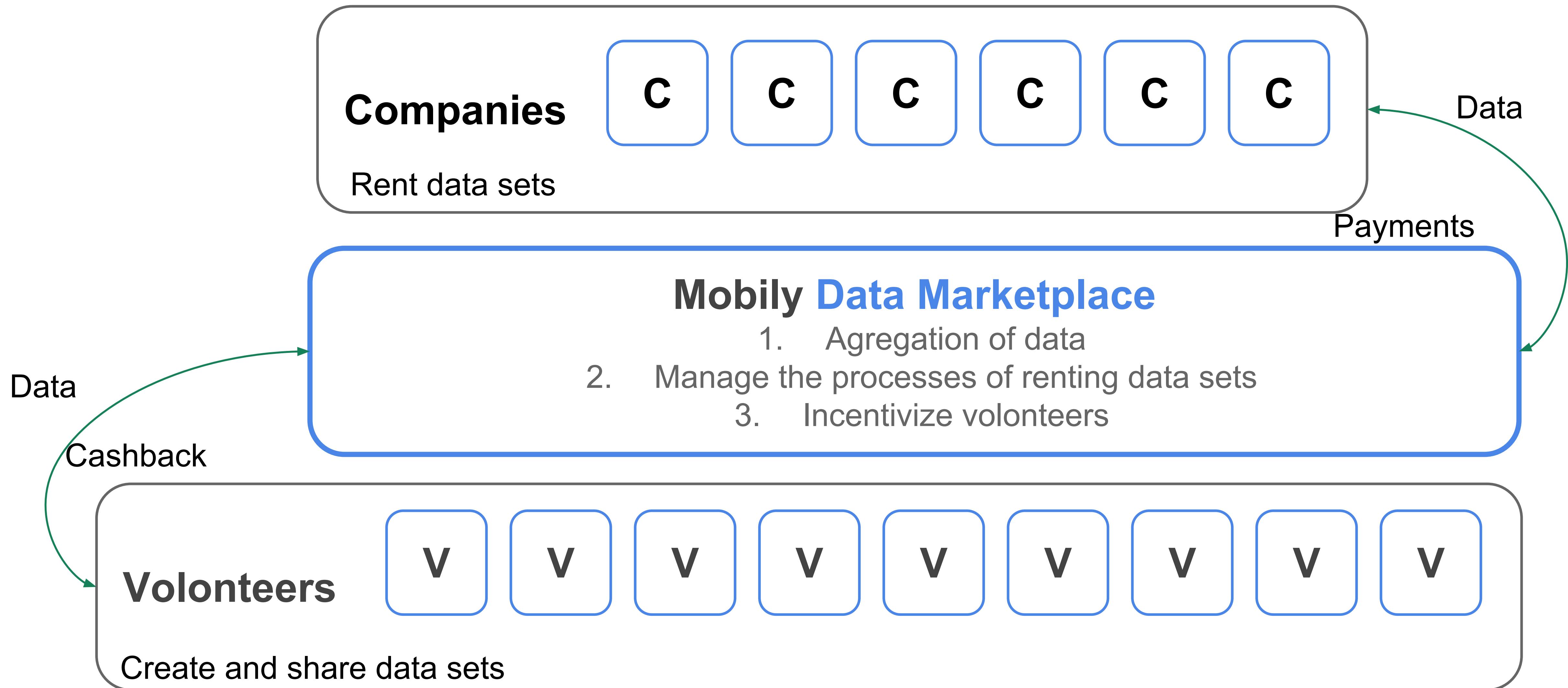
> iexec dataset deploy

<https://v3.explorer.iex.ec>

All Rights Reserved 2019



Explorer v3-alpha is live: <https://v3.explorer.iex.ec>



FREKENCY

Clients de FREKY

Restaurants, offices de tourisme, hôpitaux...

R

O

H

FREKY

Louent et monétise la data

Data

Rémunération

Data

Cashback

MobilySmart

1. Agrège de la data
2. Permet la location de cette data
3. Récompense les volontaires

V

V

V

V

V

V

V

V

V

Volontaires

Créent et mettent à disposition de la data de la mobilité