



Weaving the ILP fabric into BigchainDB/ascrbe.io

Dimitri De Jonghe

ascrbe[®] **BIGCHAIN**DB

BIGCHAIN DB



Throughput

>1,000,000 writes/s
~100,000 transactions/s



Latency

<100 ms



Capacity

Petabytes with each
node adding 48TB



Scalability

Performance increases as
nodes are added



Query

Database is fully
queryable



Decentralization

Federated
non-anonymous participation

Interledger allows devs to decentralize the cloud using a decentralized web stack.

Centralized

Partly
Decentralized

Fully
Decentralized

Apps



Proc'ing



FS



DB



Partly Dec. Apps

ascribe®

Proc'ing



FS



Dec.
DB/Ledger



Dec. Apps



Dec. Proc'ing



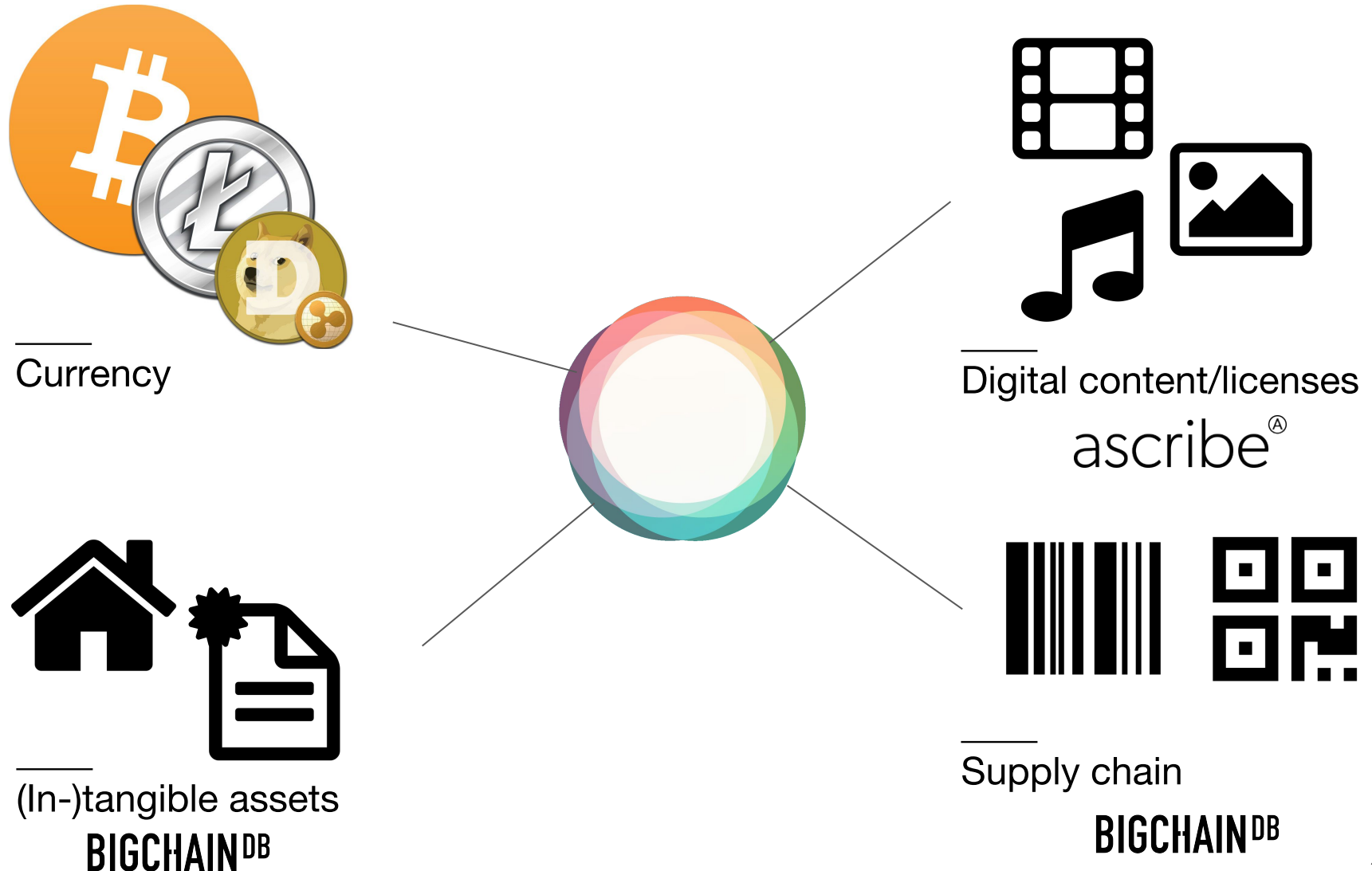
Dec. FS



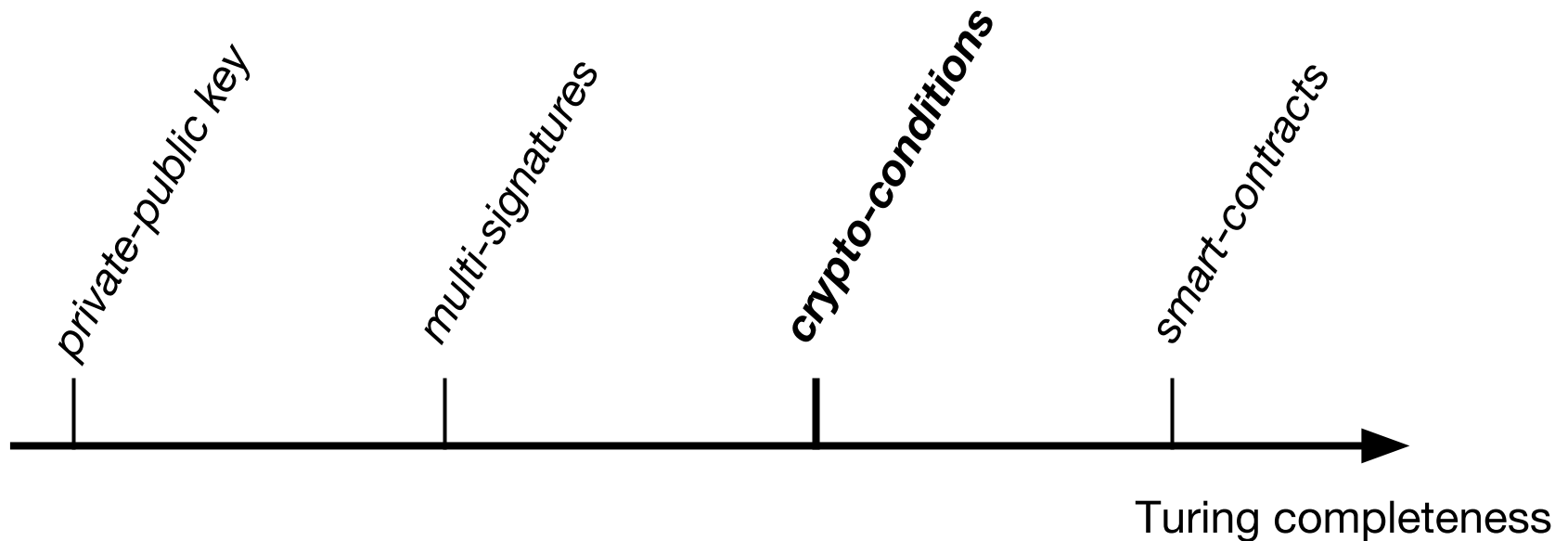
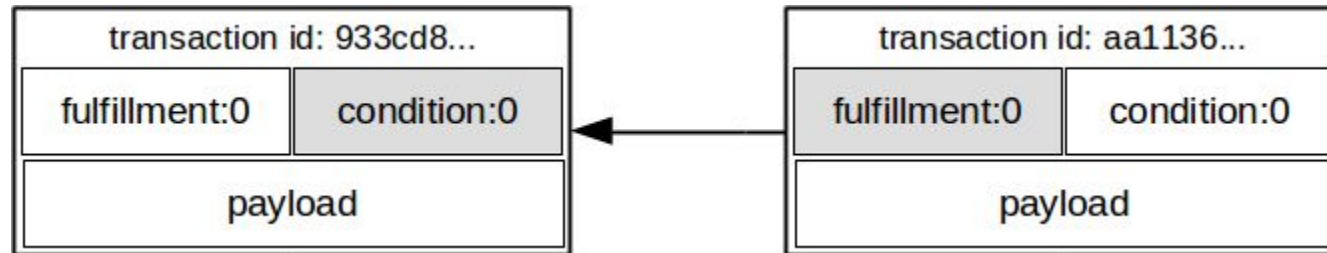
Dec.
DB/Ledger



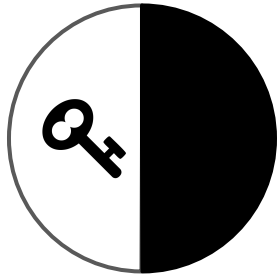
Interledger allows BigchainDB/ascibe to interexchange between fungible asset repositories



BigchainDB uses ILP crypto-conditions as a **safe** alternative to smart contracts



BigchainDB uses ILP crypto-conditions as a **safe** alternative to smart contracts



```
{
  "id": "933cd83a419d2735822a2154c84176a2f419cbd449a74b94e592ab807af23861",
  "transaction": {
    "conditions": [
      {
        "cid": 0,
        "condition": {
          "details": {
            "bitmask": 32,
            "public_key": "BwuhqQX8FPsmqYiRV2CSZYWwsSwgSSQQFHjqxKEuqkPs",
            "signature": None,
            "type": "fulfillment",
            "type_id": 4
          },
          "uri": "cc:4:20:oqXTWvR3afHHX80a0084kZxS6nH4GEBXD4Vw8Mc5iBo:96"
        },
        "new_owners": ["BwuhqQX8FPsmqYiRV2CSZYWwsSwgSSQQFHjqxKEuqkPs"]
      }
    ],
    "data": {
      "hash": "872fa6e6f46246cd44afdb2ee9cfae0e72885fb0910e2bcf9a5a2a4eadb417b8",
      "payload": {"msg": "Hello BigchainDB!"}
    },
    "fulfillments": [
      {
        "current_owners": ["3LQ5dTiddXymDhNzETB1rEkp4mA7fEV1Qeiu5ghHiJm9"],
        "fid": 0,
        "fulfillment": "cf:4:Iq-BcczwnaM2UpF-TDPdwK8fQ6IXkD_6uJaxBzd984yx...",
        "input": None
      }
    ],
    "operation": "CREATE",
    "timestamp": "1460981667.449279"
  },
  "version": 1
}
```

condition = cc.Ed25519Fulfillment()
condition.public_key = "BwuhqQ..."

optionally provide condition structure
condition.to_dict()

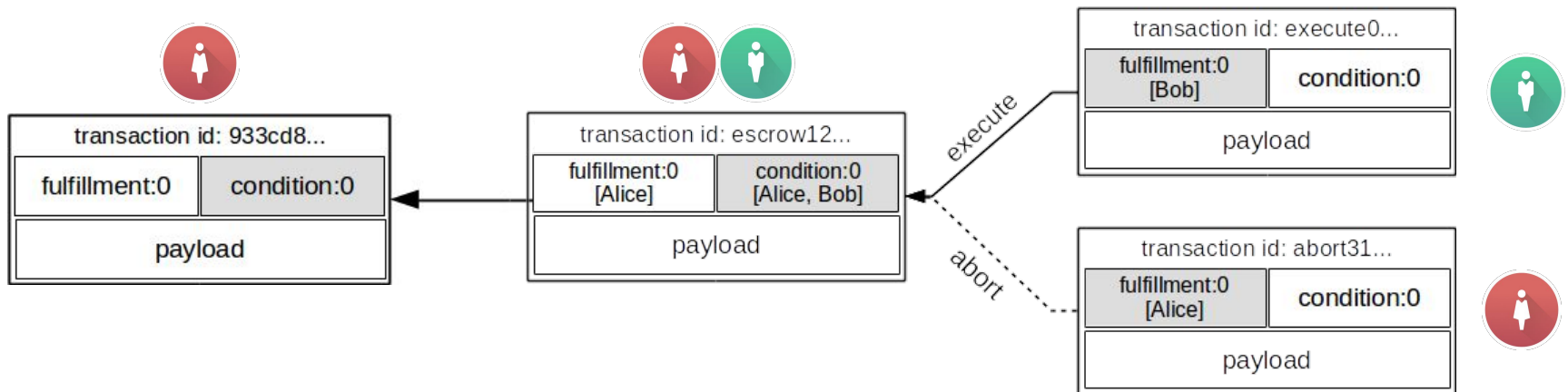
condition.condition_uri

fulfillment = \
 input.conditions[0].from_dict()
fulfillment.sign(message, private_key)

fulfillment.serialize_uri()

Crypto-conditions as building blocks for *ledger-provided escrow*

BigchainDB is **append only**, how to provide **escrow**?



```
if timeout_condition.validate(utcnow()):
    execute_fulfillment.validate(msg) == True
    abort_fulfillment.validate(msg) == False
else:
    execute_fulfillment.validate(msg) == False
    abort_fulfillment.validate(msg) == True
```

Crypto-conditions as building blocks for *ledger-provided escrow*

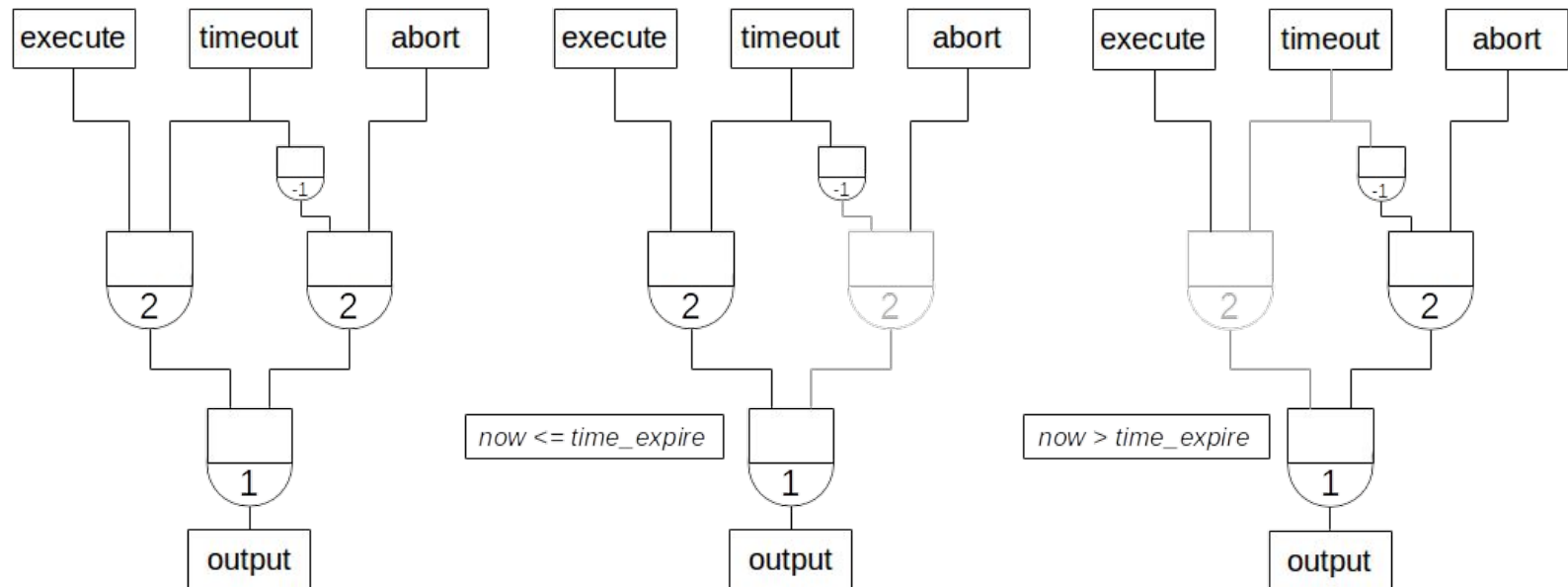
Timeout-condition

`now() < expiry_time`

experimental

Inverter

`if fact == True: output = False`



Utils for crypto-conditions

JSON/dict serialization:

`to_dict(), from_dict()`

Queryability of complex branches:

get/update/insert/remove subconditions

`get_subcondition_path_for_type(type_id):`

...

return subcondition, indices

```
fulfillment.subconditions[indices[0]]['body'] \
    .subconditions[indices[1]]['body'] \
    .subconditions[indices[2]]['body']
```

