Weaving the ILP fabric into BigchainDB/ascribe.io

Dimitri De Jonghe
BIGCHAINDB

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.
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

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

# optionally provide condition structure
condition.to_dict()

condition.condition_uri

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

fulfillment.serialize_uri()
```
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**

\[ \text{now}() < \text{expiry\_time} \Rightarrow \text{experimental} \]

**Inverter**

\[ \text{if fact == True: output = False} \]
Utils for crypto-conditions

**JSON/dict serialization:**

```python
to_dict(), from_dict()
```

**Queryability of complex branches:**

```python
def get_subcondition_path_for_type(type_id):
    ...
    return subcondition, indices
```

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