COALA IP Protocol Specification

Agenda I

- Introduction
    - Goals
    - Ideas
- The LCC Framework
- Technical details
    - What is IPLD?
    - What is Linked Data? + useful RDF schemata
    - What is the Interledger Protocol?

Agenda II

- The COALA IP Protocol
    - Transformations from LCC RRM
    - Evaluations and Challenges
    - Roadmap

Introduction

Goals

- A licensing framework for digital assets that:
    - Is easily approachable by all participants *(devs, rights holders, copyright societies, …)*
    - Is easily extensible and future-proof
    - Guarantees immutability and tamper-resistance
    - Is blockchain-agnostic
    - Is free (free as in FoS/FOSS) for everyone to participate and use

Ideas

- Let’s use:
  - **Linked Data**, it’s easily extensible
  - **IPLD**, it’s integrity is cryptographically verifiable
  - **ILP**, it allows COALA IP to live on many ledgers
  - The **LCC framework**, it’s concise and applicable

History

- COALA - spinout from MIT & Harvard for blockchain workshops. Meet 3-4 times per year.
- IP working group started in fall 2015. Meet at each COALA workshop. Communication in between.
- Contributors from COALA, IPFS, Ujo / Consensys, Mycelia, ascribe / BigchainDB, more. Relations to mediachain, SoundCloud, Open Music Initiative, W3C, more.

The LCC Framework

The LCC Framework

Catalogue of documents (main ones used):

- LCC Entity Model (short: LCC EM)
- LCC Rights Reference Model (short: LCC RRM)
- LCC Ten Targets for a Rights Network
- LCC Principles of Identification

⇒ style of writing: “implementation-**agnostic**”

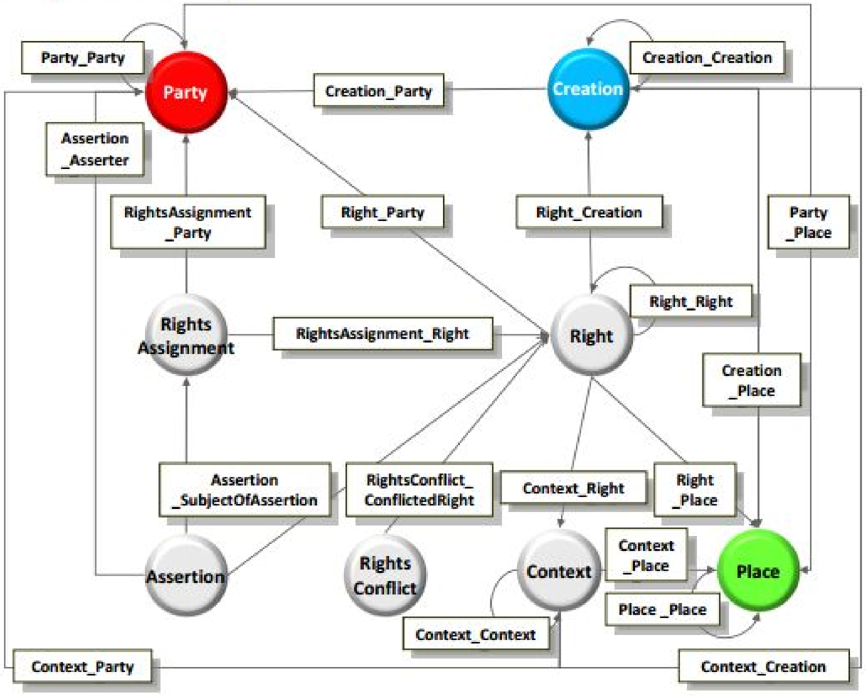
The LCC Rights Reference Model

- Represent IP rights digitally

- Data model on top of the LCC EM

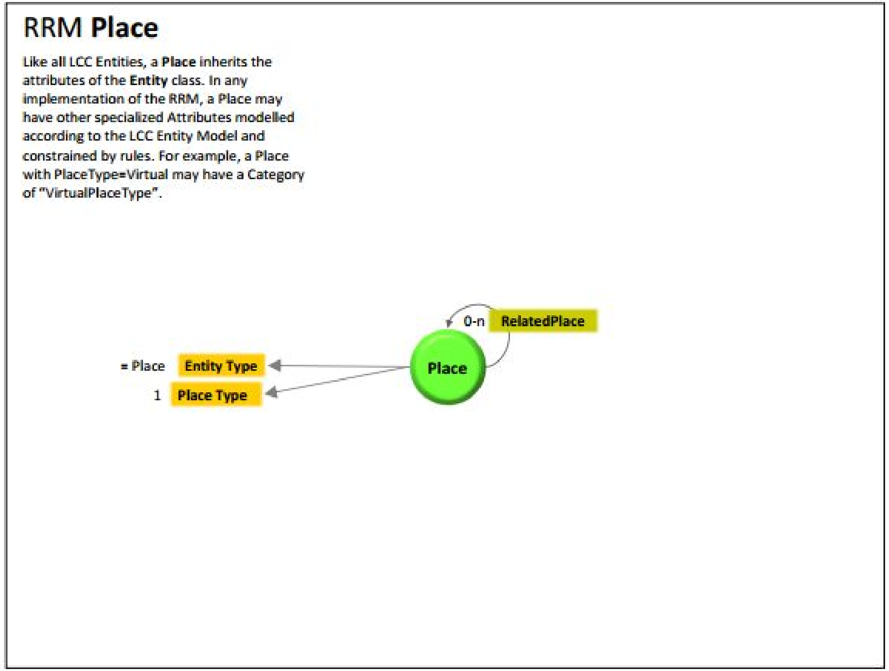
⇒ 7 (main) entities

RRM **Entity and Link Types**



LCC RRM: Place

- A localizable or virtual place
- For example:
  - *“New York City”; or*
  - *“https://newyork.city”*

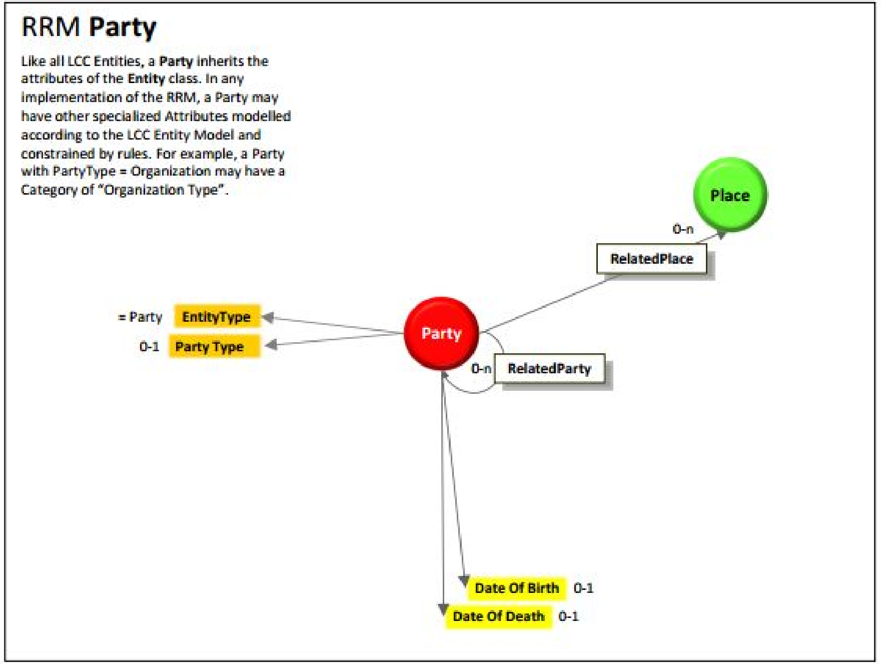


RRM **Place**

Like all LCC Entities, a **Place** inherits the attributes of the **Entity** class. In any implementation of the RRM, a Place may have other specialized Attributes modelled according to the LCC Entity Model and constrained by rules. For example, a Place with PlaceType=Virtual may have a Category of “VirtualPlaceType”.

LCC RRM: Party

- An individual or a group of individuals (organization)
- Represents: right holders, licensors, users
- For example:
  - *"Andy Warhol"; or*
  - *"Warner Bros. Entertainment"*



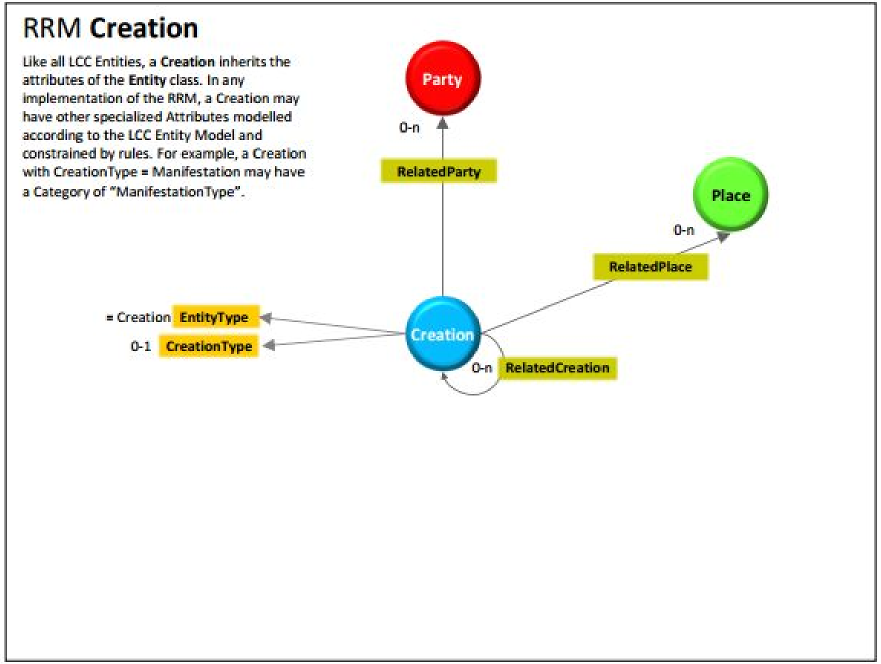
RRM **Party**

Like all LCC Entities, a **Party** inherits the attributes of the **Entity** class. In any implementation of the RRM, a Party may have other specialized Attributes modelled according to the LCC Entity Model and constrained by rules. For example, a Party with PartyType = Organization may have a Category of "Organization Type".

LCC RRM: Creation

- Smth (in)directly made by human beings
- Separation:
  - Work and;
  - Manifestation
- For example:

“32 Campbell’s Soup Cans” (by Andy Warhol)



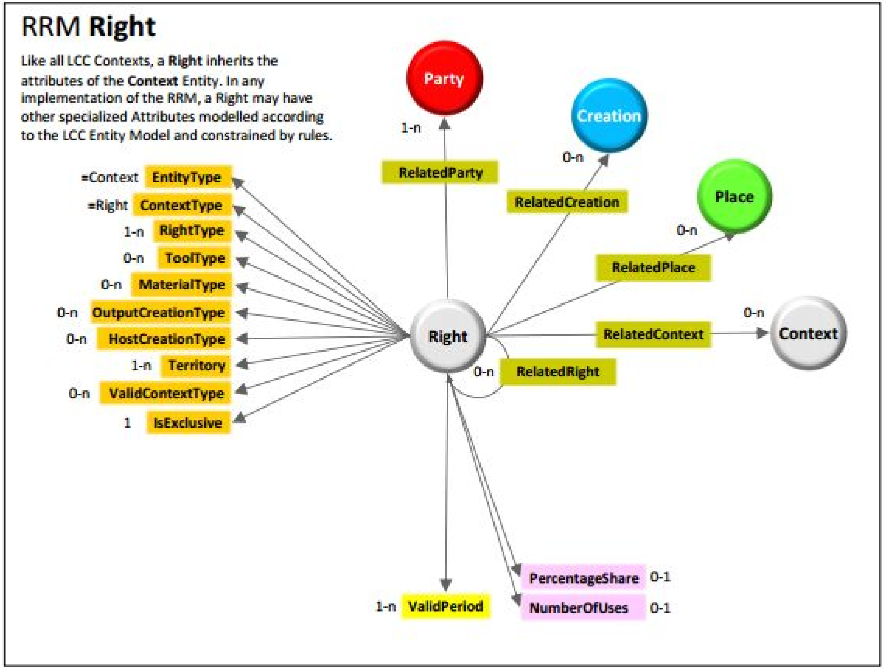
RRM **Creation**

Like all LCC Entities, a **Creation** inherits the attributes of the **Entity** class. In any implementation of the RRM, a Creation may have other specialized Attributes modelled according to the LCC Entity Model and constrained by rules. For example, a Creation with CreationType = Manifestation may have a Category of “ManifestationType”.

LCC RRM: Right

- Set of permissions that entitle a Party to do something with a Creation
- For example:

  “Andy Warhol controls all Rights to 32 Campbell’s Soup Cans”



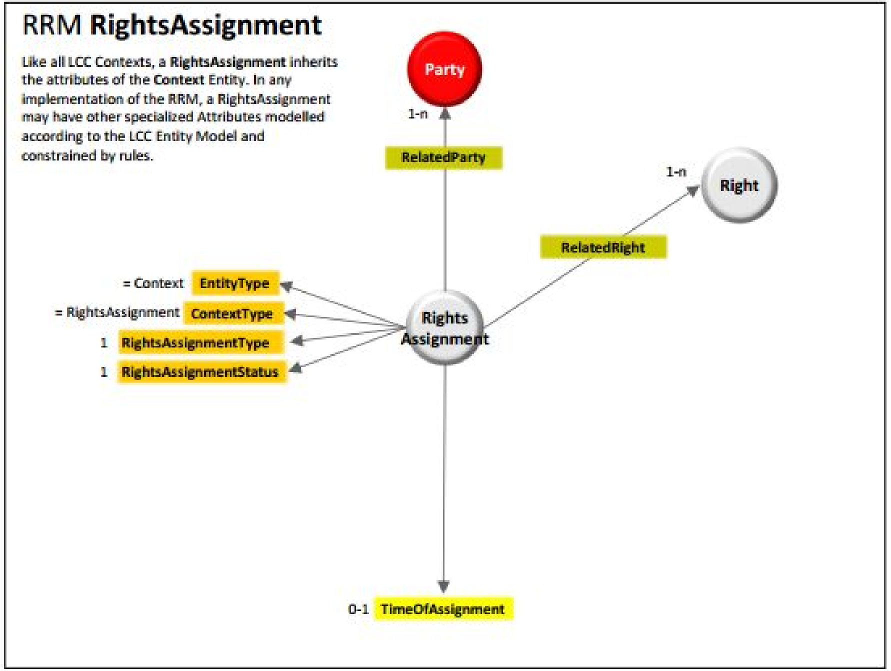
RRM **Right**

Like all LCC Contexts, a **Right** inherits the attributes of the **Context** Entity. In any implementation of the RRM, a Right may have other specialized Attributes modelled according to the LCC Entity Model and constrained by rules.

LCC RRM: RightsAssignment

- Every possible way to “transfer” a Right from Party to Party
- For example:

  “I, Andy Warhol declare that ‘32 Campbell’s Soup Cans’ shall be published under CC free use license”



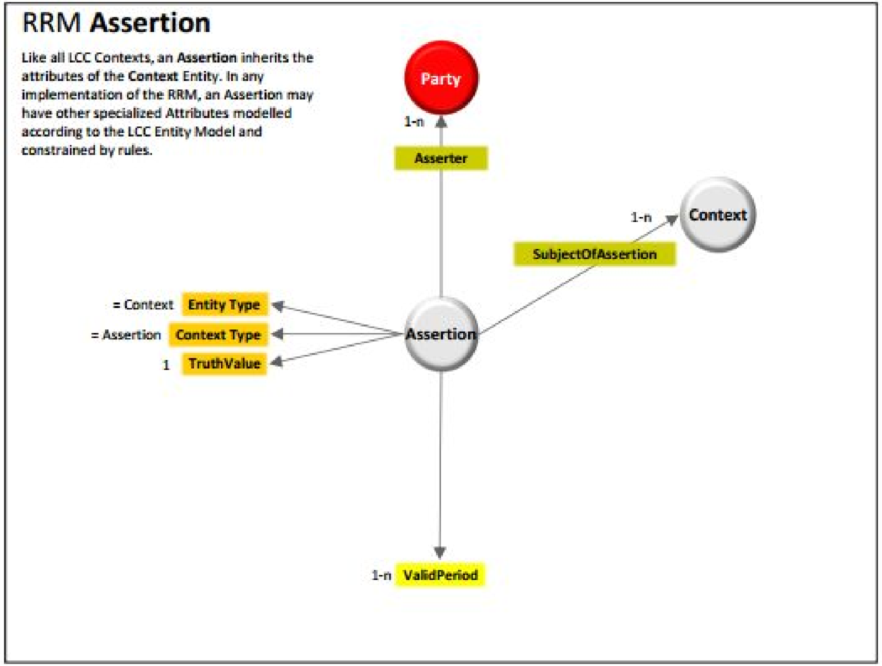
RRM **RightsAssignment**

Like all LCC Contexts, a **RightsAssignment** inherits the attributes of the **Context** Entity. In any implementation of the RRM, a RightsAssignment may have other specialized Attributes modelled according to the LCC Entity Model and constrained by rules.

LCC RRM: Assertion

- A claim made about the substance of a Right
- For example:

  “I, the MoMa, New York, claim that Andy Warhol is the righteous creator of ‘32 Campbell’s Soup Cans’”



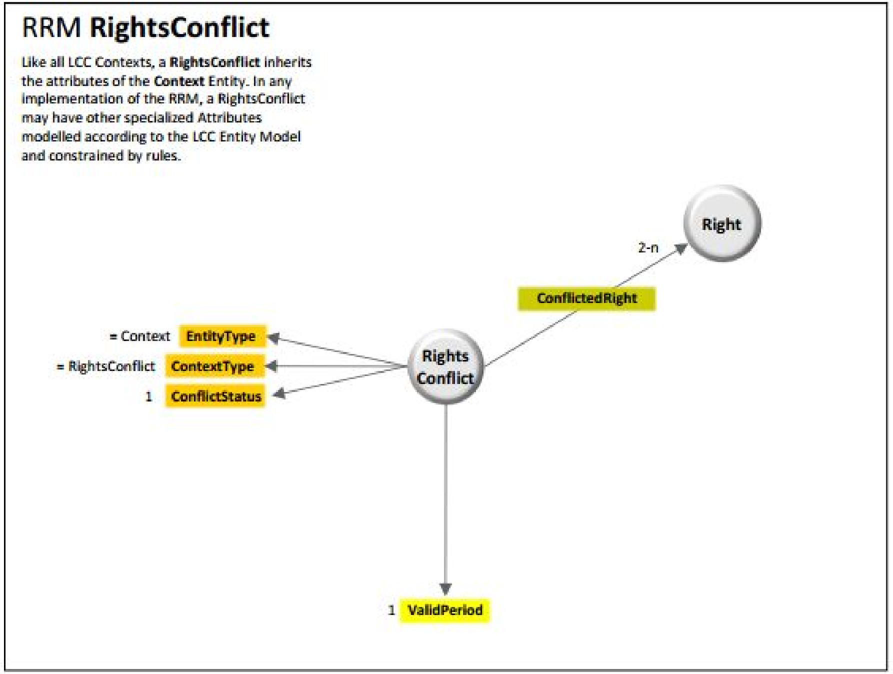
RRM Assertion

Like all LCC Contexts, an **Assertion** inherits the attributes of the **Context** Entity. In any implementation of the RRM, an Assertion may have other specialized Attributes modelled according to the LCC Entity Model and constrained by rules.

LCC RRM: RightsConflict

- A statement of disagreement over a Right
- For example:

  *“I, Tim, declare that Andy Warhol is **NOT** the righteous creator of ‘32 Campbell’s Soup Cans’”*



RRM **RightsConflict**

Like all LCC Contexts, a **RightsConflict** inherits the attributes of the **Context** Entity. In any implementation of the RRM, a RightsConflict may have other specialized Attributes modelled according to the LCC Entity Model and constrained by rules.

Technical Details

Main technical components

- IPLD: Merkle-linking of JSON objects
- JSON-LD: URI-linking of JSON objects
- Interledger Protocol: Linking of ordered transactions on blockchains
- Some canonical identity protocol for the web (specifics: unclear as of now - “Reboot the Web of Trust”?)

What is IPLD?

What is IPLD?

- Merkle-linking JSON objects
    - Cryptographic integrity-checking of data

- Merkle-paths JSON objects
    - Content-addressable data/storage

Merkle-Linking example I

```
import ipld

In [2]: person = {

...:     "givenName": "Andy",

...:     "familyName": "Warhol",

...:     "birthDate": "1928-08-06"

...: }

In [3]: serialized_person = ipld.marshal(person) # serialize using CBOR

Out[3]: b'\xa3ibirthDatej1928-08-06jfamilyNamefWarholigivenNamedAndy'

In [4]: ipld.multihash(serialized_person) # hash CBOR value and get a hash digest

Out[4]: ‘QmRinxtytQFizqBbcRfJ3i1ts617W8AA8xt53DsPGTfisC’
```

Merkle-Linking example II

```
In [5]: creation = {

    "name":"32 Campbell's Soup Cans",

    "dateCreated": "01-01-1962",

    "exampleOfWork": "https://en.wikipedia.org/wiki/.../media/File:
Campbells_Soup_Cans_MOMA.jpg",

    "author": { "/": "QmRinxtytQFizqBbcRfJ3i1ts617W8AA8xt53DsPGTfisC" } # see
resulting hash prev slide

}

In [6]: serialized_creation = ipld.multihash(ipld.marshal(creation))

Out[6]: ‘QmfMLNLyJZgvSPkNMvsJspRby2oqP6hWZ8Nd2PvKLhudmK’
```

Merkle-paths example

- Resolve path to specific set of information

```
In [7]: ipld.resolve
('/ipfs/QmfMLNLyJZgvSPkNMvsJspRby2oqP6hWZ8Nd2PvKLhudmK/author')

Out[7]:

{“givenNamen”: “Andy”,

 “familyName”: “Warhol”,

 “birthDate”: “1928-08-06”}
```

Why IPLD?

Benefits:

- Crypto-integrity checking
- Content-addressing
- Inter-ledger resolvability
- Canonicalized ID
- Immutability
- Future-proof (multihash)

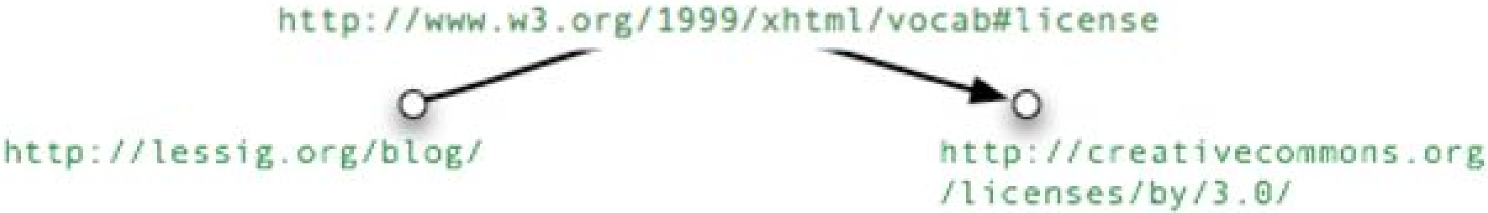
Caveats:

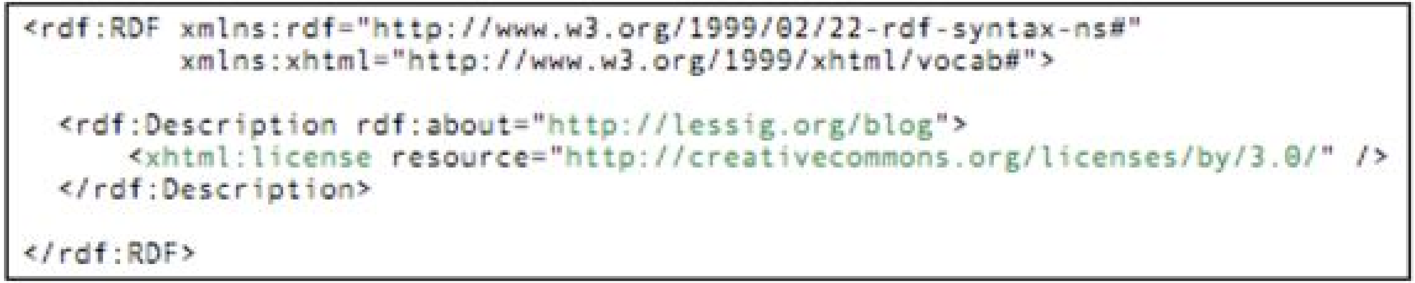
- Non-standard protocols (multi-x)
- Breakage with established protocols (e. g. URI)
- Non-compatible with LD ontology

What is Linked Data?

Resource Description Framework (short: RDF)

- A way to express assertions in a schematic way


http://www.w3.org/1999/xhtml/vocab#license
http://lessig.org/blog/
http://creativecommons.org
/licenses/by/3.0/



```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:xhtml="http://www.w3.org/1999/xhtml/vocab#">

  <rdf:Description rdf:about="http://lessig.org/blog">
      <xhtml:license resource="http://creativecommons.org/licenses/by/3.0/" />
  </rdf:Description>

</rdf:RDF>
```

What is JSON-LD?

- Data structure serialize RDF in JSON

```
{
  "@type": "http://schema.org/Person",
  "@id": "http://example.com/data/AndyWarhol",
  "givenName": "Andy",
  "familyName": "Warhol",
}
```

```
http://example.com/data/AndyWarhol
  http://www.w3.org/1999/02/22-rdf-syntax-ns#type
    http://schema.org/Person .
```

Why JSON-LD?

Benefits:

- Approachable (compared to XML/RDF)
- Extensible
- Well-shaped data
- Maybe: Cross-usage from *Semantic Web*

Caveats:

- Assumption: Data is mutable
- Incompabilities with IPLD

Useful RDF Schemata

Useful RDF schemata I

- LCC RRM Party:
  - schema.org/Person
  - schema.org/Organization

- LCC RRM Creation:
  - schema.org/CreativeWork
  - And its subtypes: Book, Movie, MusicComposition

Useful RDF schemata II

- LCC RRM Place:
  - schema.org/Place

- LCC RRM Assertion:
  - schema.org/AssessAction
  - Additionally: Web of Trust Ontology

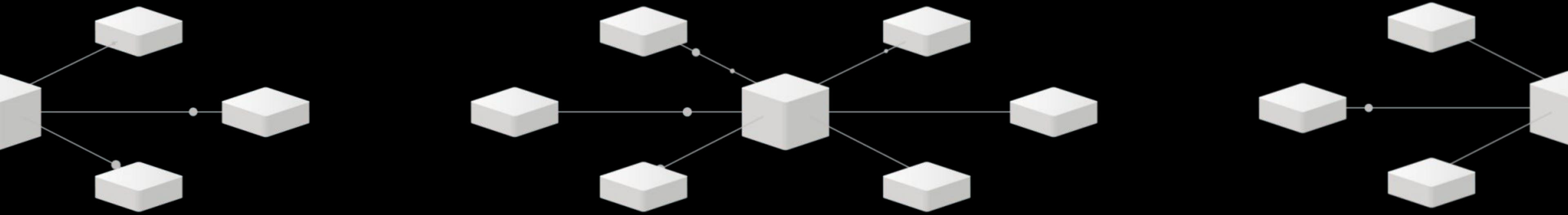
What is the Interledger Protocol?

Interledger

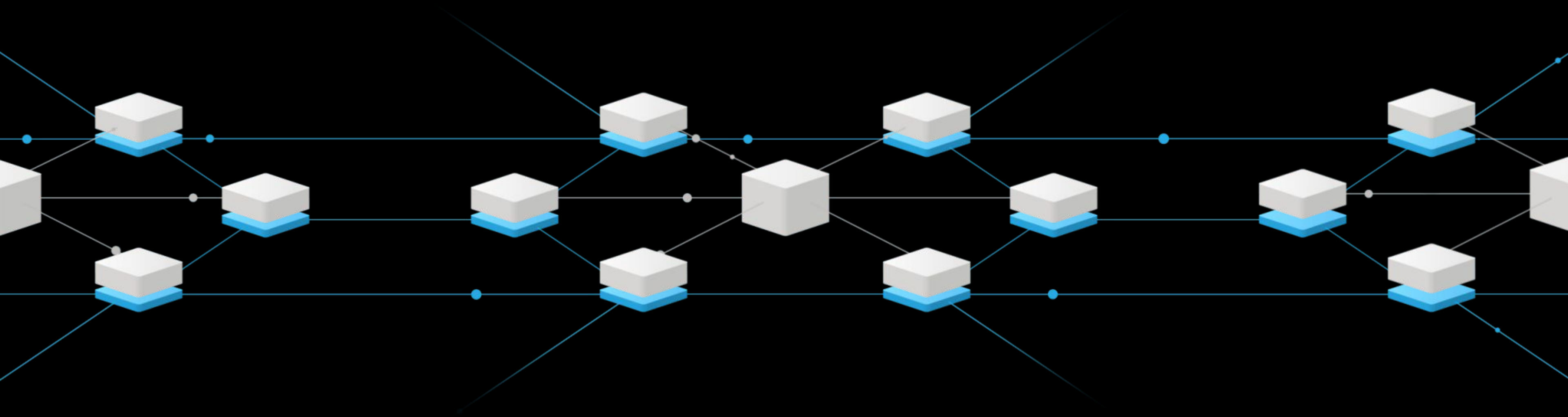
The protocol for connecting blockchains

The World Will **Never Agree** On a Single Ledger

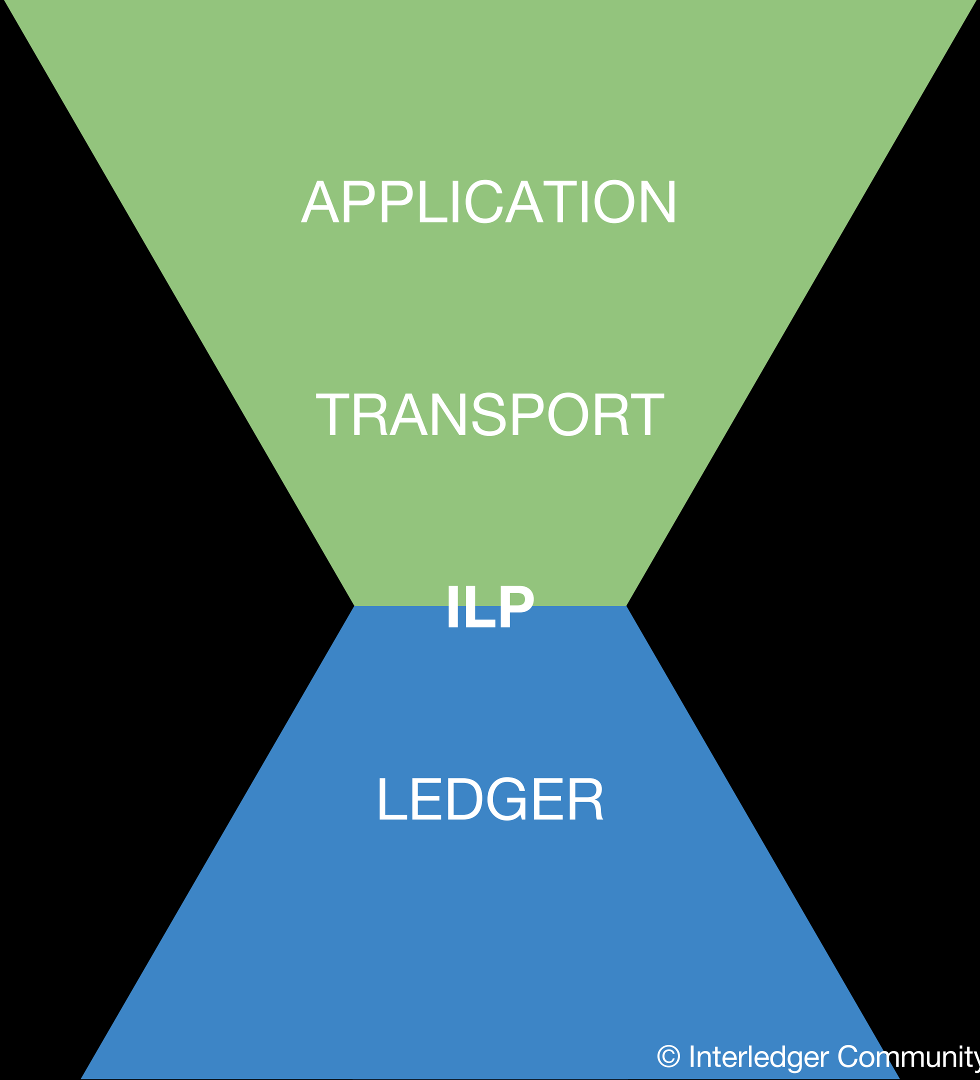
Learning From History



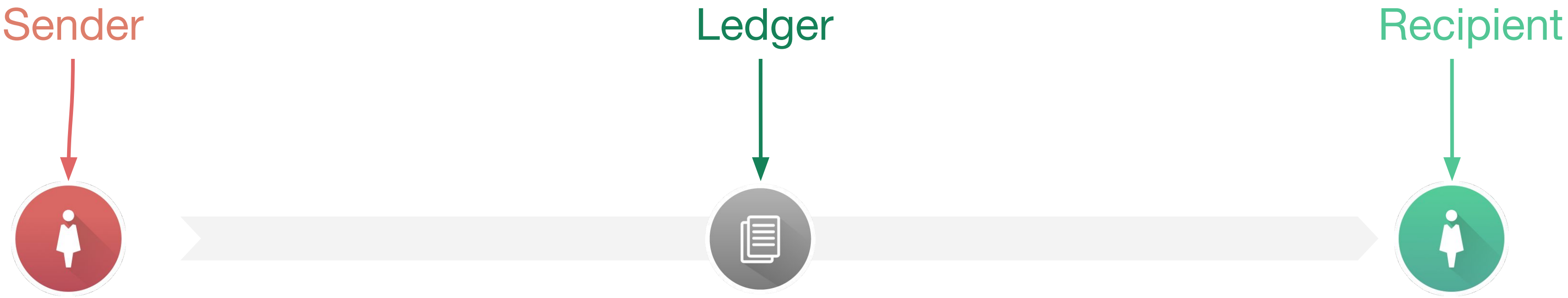
*Inter*net: Network of Networks



Interledger Architecture

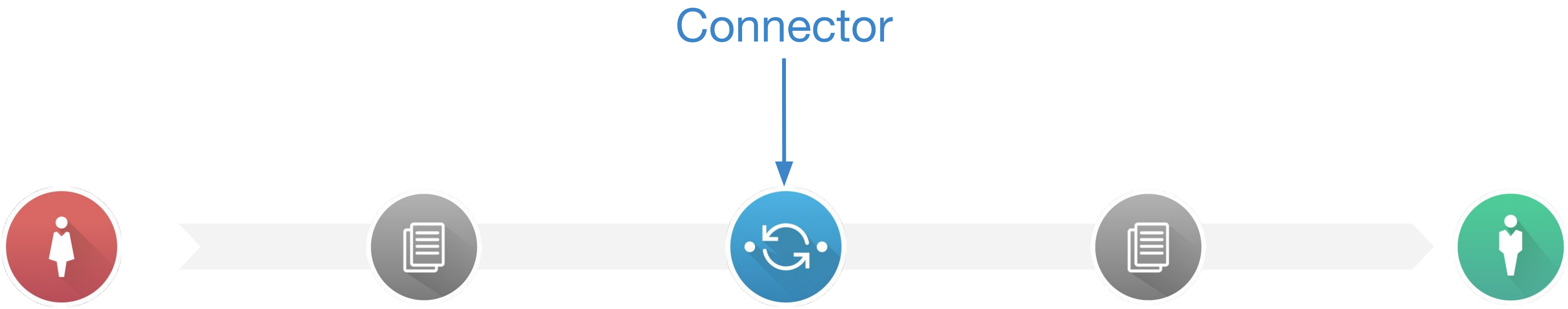


Ledgers track accounts and balances

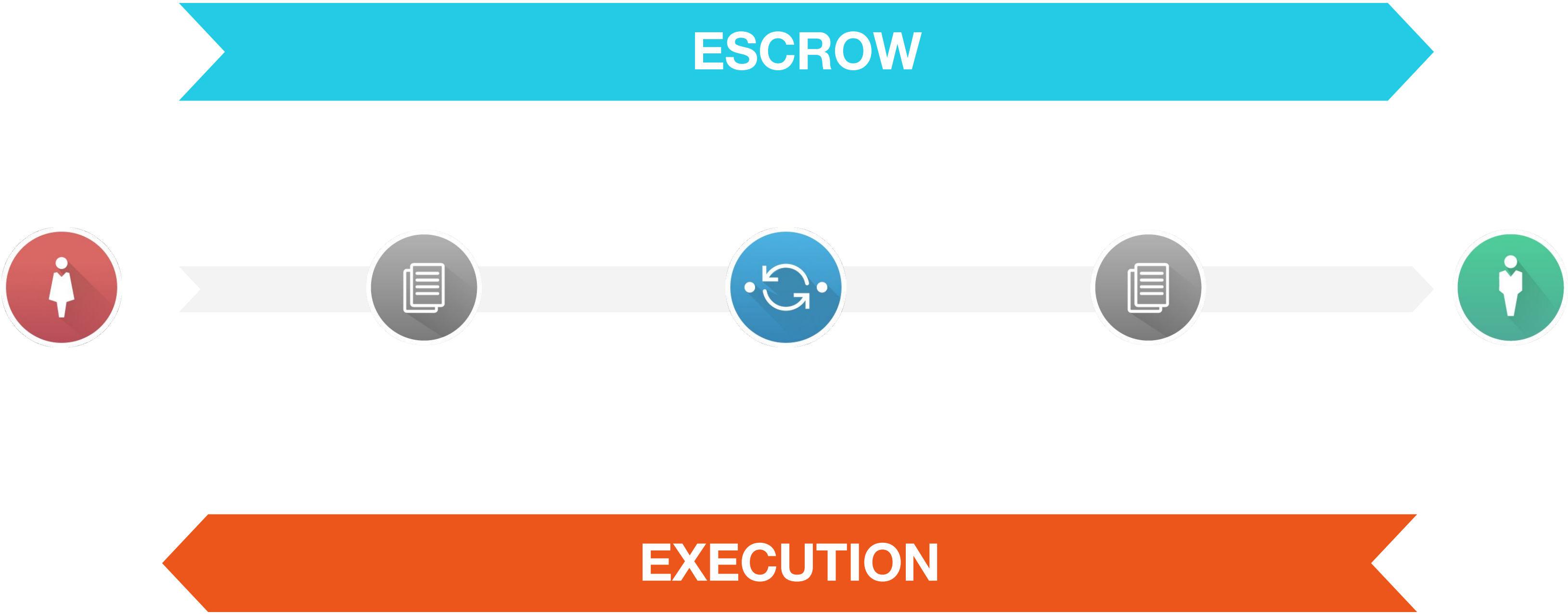


But not everyone is on the same ledger

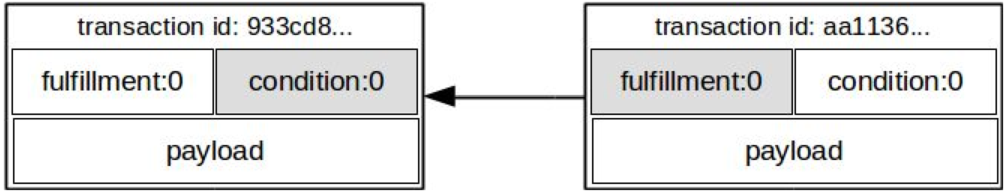
Connectors relay assets



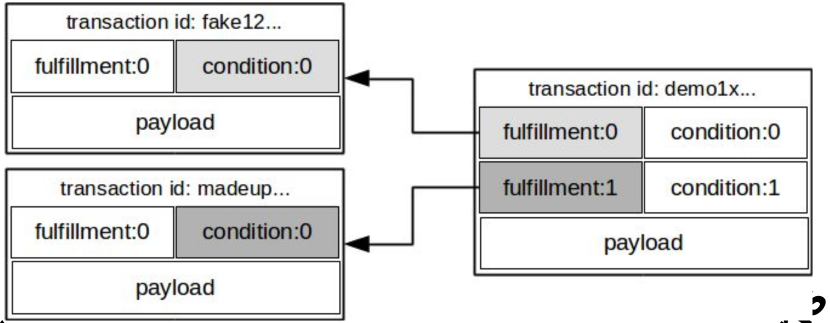
Transfers are escrowed L2R, executed R2L



Assets with crypto-conditions



Assets with crypto-conditions



transaction id: fake12...

fulfillment:0 | condition:0

payload

transaction id: madeup...

fulfillment:0 | condition:0

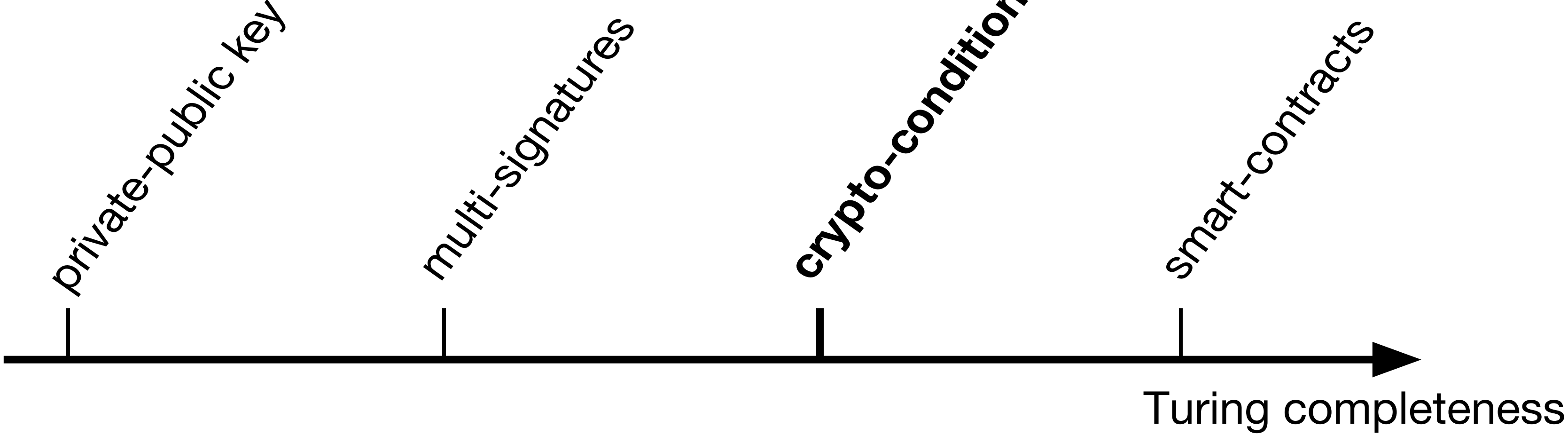
payload

transaction id: demo1x...

fulfillment:0 | condition:0

fulfillment:1 | condition:1

payload



private-public key

multi-signatures

crypto-conditions

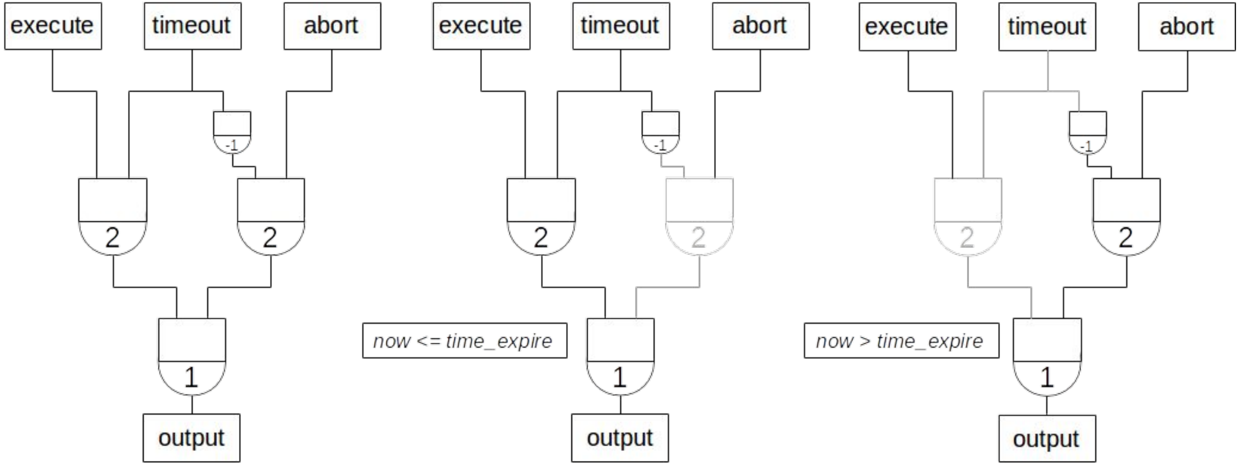
smart-contracts

Turing completeness

Assets with crypto-conditions

Boolean logic with signatures

Designed by the *Interledger* community



COALA IP Protocol

What is the COALA IP Protocol?

- A community-driven *minimum-viable set of data* for Intellectual Property licensing (RDF schema definitions)

- A free and open *messaging* protocol for license-transactions (Interledger & IPLD)

COALA IP: Place

```
{
    "@type": { "/": "<hash pointing to RDF-Schema of Place>" },
    "geo": {
        "@type": { "/": "<hash pointing to RDF-Schema of GeoCoordinates>" },
        "latitude": "40.75",
        "longitude": "73.98"
    },
    "name": "Empire State Building"
}
```

COALA IP: Party (only Individual)

```
{

    "@type": { "/": "<hash pointing to RDF-Schema of Individual>" },

    "givenName”: “Andy”,

    “familyName”: “Warhol”,

    “birthDate”: “1928-08-06”,

    “deathDate”: “1987-02-22”

    // and any other arbitrary meta data

    // TDB: Let’s use an established identity protocol here

}
```

COALA IP: Creation

```
{

    "@type": { "/": "<hash pointing to RDF-Schema of Creation>" },

    "name”: “Lord of the Rings”,

    “author”: { "/": "<hash pointing to the Author>" }

}
```

COALA IP: Creation (a *digital* Manifestation)

```
{   "@type": { "/": "<hash pointing to RDF-Schema of Manifestation>" },
    "name": "The Fellowship of the Ring",
    "creation": { "/": "<hash pointing to the Creation>" },
    "digital_work": { "/": "<hash pointing to a file on e.g. IPFS>" },
    "fingerprints": [
        "Qmbs2DxMBraF3U8F7vLAarGmZaSFry3vVY5zytuN3BxwaY",
        "<multihash/multifingerprint value>"
    ],
    "locationCreated": "<URI pointing to a Place object>"
}
```

COALA IP: Creation (a *physical* Manifestation)

```
{   "@type": { "/": "<hash pointing to RDF-Schema of Manifestation>" },

    "name": "The Fellowship of the Ring",

    "creation": { "/": "<hash pointing to the Creation>" },

    "datePublished": "29-07-1954",

    "locationCreated": "<URI pointing to a Place object>"

}
```

COALA IP: Right

```
{   "@type": { "/": "<hash pointing to RDF-Schema of Right>" },

    “usages”: “all|copy|play|stream|...”,

    “territory”: { “/”: “<hash pointing to a Place>” },

    “context”: “inflight|inpublic|commercialuse…”,

    “exclusive”: true|false,

     ...

    "manifestation": { "/": "<hash pointing to the Manifestation>" },

    "license": { "/": "<hash pointing to the License>" }

}
```

COALA IP: RightsAssignment

- A special case: RightsAssignments must be stored in an *ordered fashion*

- Store on an **Interledger Protocol compliant ledger**
    - Provenance of assets (chain of events)
    - *True* ownership of assets (priv and pub key)
    - Enhanced transfers (escrowed, multi-sig)

⇒ e.g. BigchainDB (implements IPLD and ILP)

Intermezzo:
RightsAssignment on BigchainDB

BigchainDB: RightsAssignment example I

- Add Right to payload of transaction to create it

```
# Create a Right and write it to BigchainDB

In [1]: tx_create = b.create_transaction(<node_pub>, <creator_pub>, None, 'CREATE',
payload=<Minimum viable data set of a Right>)

In [2]: tx_create_signed = b.write_transaction(b.sign_transaction(tx_create,
<creator_priv>))
```

BigchainDB: RightsAssignment example II

```
In [5]: tx_create_signed

Out[5]: {"id":"aa11365317cb89bfdae2375bae76d6b8232008f8672507080e3766ca06976dcd",

         "transaction": {

             "conditions": [...],

             "data": <Right>,

             "fulfillments": [...],

             "operation": "CREATE",

             "timestamp": "1460981671.262047"

    }

}
```

BigchainDB: RightsAssignment example III

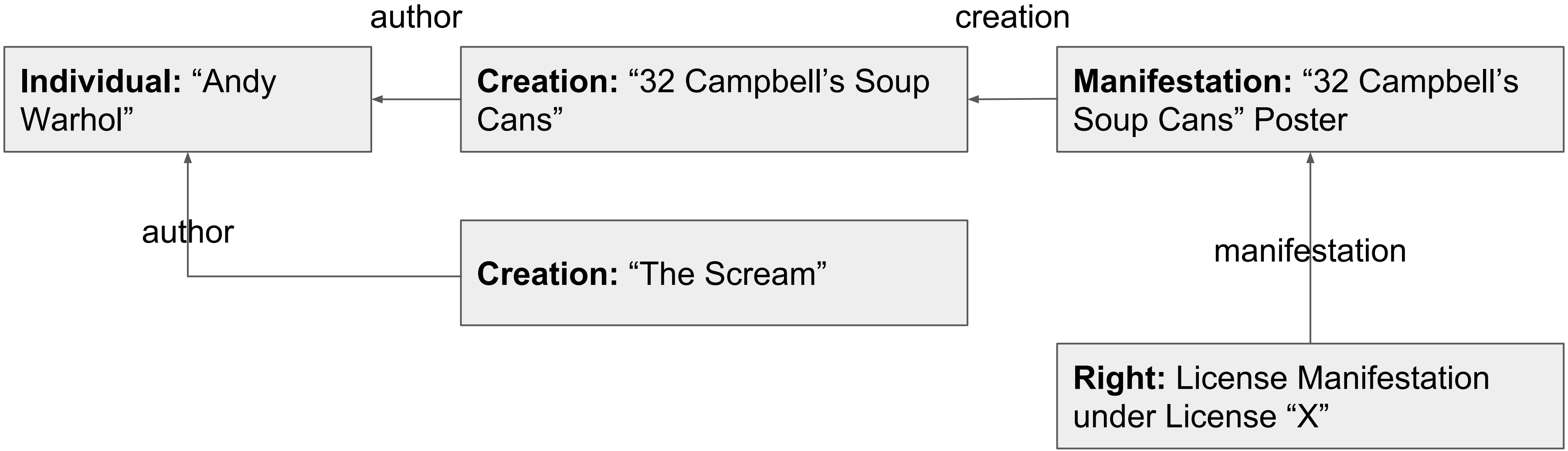
- Transfer the asset representing the Right

```
# Transfer a Right with BigchainDB

In [3]: tx_transfer = b.create_transaction(<creator_pub, <transferee_pub>, tx_signed
[‘id’], 'TRANSFER')

In [4]: tx_transfer_signed = b.write_transaction(b.sign_transaction(tx_transfer,
<creator_pub>))
```

BigchainDB: RightsAssignment example IV

```
In [5]: tx_transfer_signed

Out[5]: {"id":"aa11365317cb89bfdae2375bae76d6b8232008f8672507080e3766ca06976dcd",

         "transaction": {

             "conditions": [...],

             "fulfillments": [...],

             "operation": "TRANSFER",

             "timestamp": "1460981677.472037"

    }

}
```

COALA IP: Assertion I



author

creation

Individual: “Andy Warhol”

Creation: “32 Campbell’s Soup Cans”

Manifestation: “32 Campbell’s Soup Cans” Poster

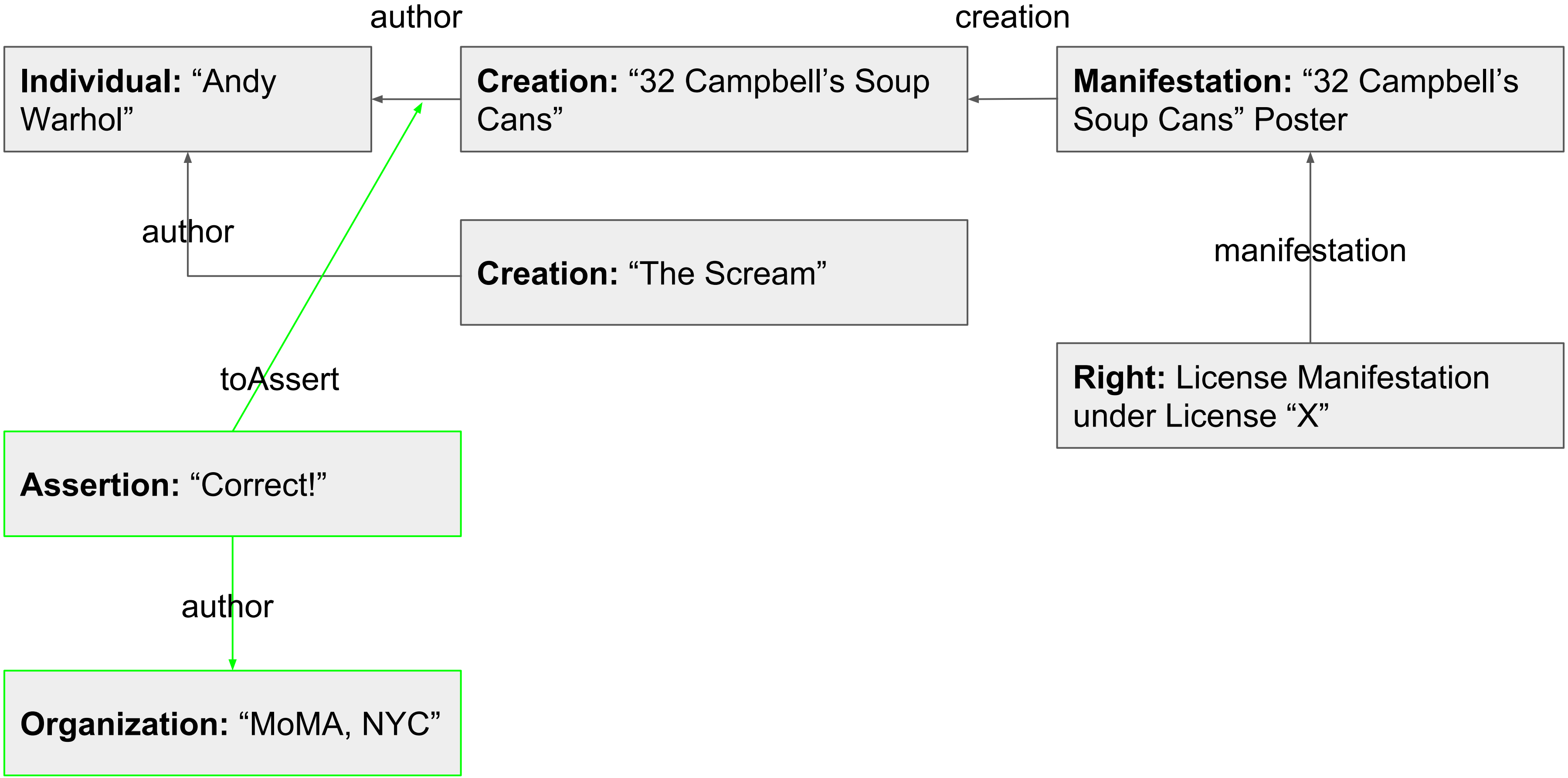
author

Creation: “The Scream”

manifestation

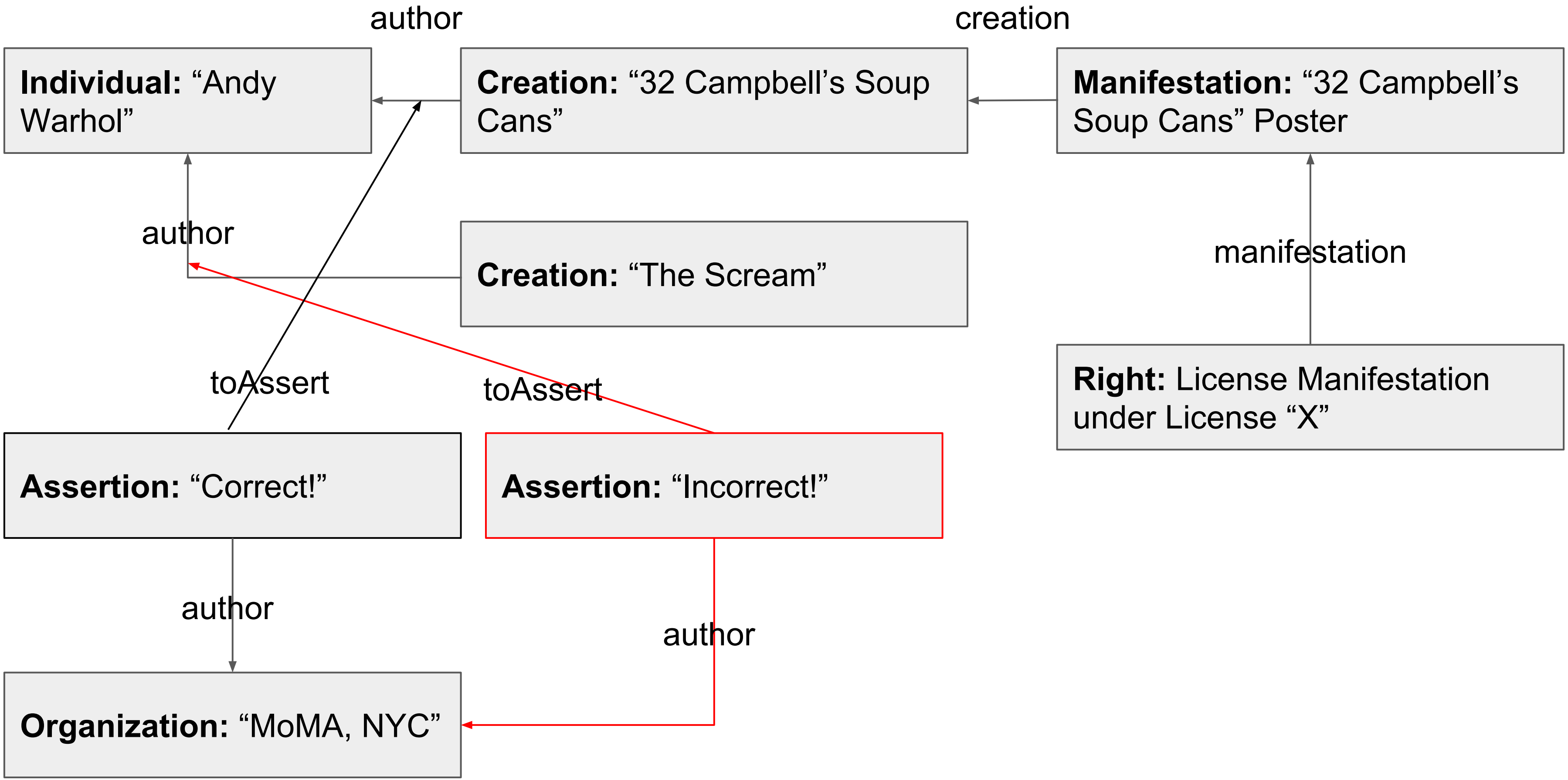
Right: License Manifestation under License “X”

COALA IP: Assertion II



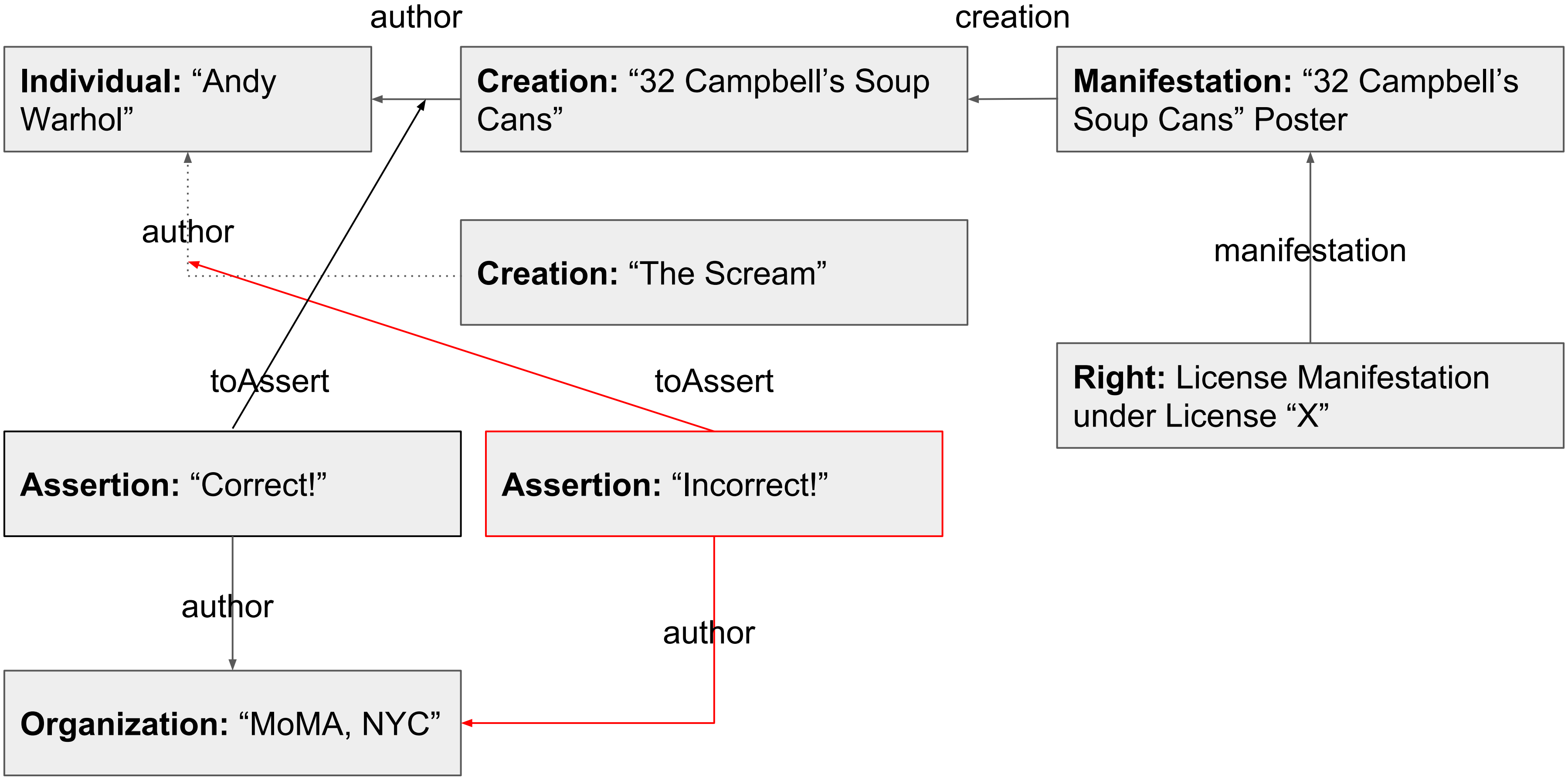
author
creation
Individual: “Andy Warhol”
Creation: “32 Campbell’s Soup Cans”
Manifestation: “32 Campbell’s Soup Cans” Poster
author
Creation: “The Scream”
manifestation
Right: License Manifestation under License “X”
toAssert
Assertion: “Correct!”
author
Organization: “MoMA, NYC”

COALA IP: Assertion III



author
creation
Individual: "Andy Warhol"
Creation: "32 Campbell's Soup Cans"
Manifestation: "32 Campbell's Soup Cans" Poster
author
Creation: "The Scream"
manifestation
toAssert
toAssert
Right: License Manifestation under License "X"
Assertion: "Correct!"
Assertion: "Incorrect!"
author
author
Organization: "MoMA, NYC"

COALA IP: Assertion III



author
creation
Individual: “Andy Warhol”
Creation: “32 Campbell’s Soup Cans”
Manifestation: “32 Campbell’s Soup Cans” Poster
author
Creation: “The Scream”
manifestation
Right: License Manifestation under License “X”
toAssert
toAssert
Assertion: “Correct!”
Assertion: “Incorrect!”
author
author
Organization: “MoMA, NYC”

Evaluation

Challenges

- Immutable ledgers and data structure: Backlinks only
- Directed Graph: Unreachable nodes?
- How can ILP and IPLD’s efforts be combined further?
- The missing link: How does **identity** fit in?

Interoperability

- With data *stores* through IPLD, compatibility with:
  - IPFS
  - Mediachain
  - ...
- With ledgers through IPLD & ILP compatibility with:
  - BigchainDB
  - Ripple
  - Bitcoin? Ethereum?

Extensibility

- With RDF-based data structure:
  - Community-curatable *minimum viable data*

- With Interledger Protocol:
  - Exciting escrow opportunities (Money ←escrow → digital license)

Other achievements

- Both human and machine readable (compare to ascribe’s SPOOL protocol)
- Hyperlinked media: Exciting times for crawlers
- Fairly compliant with parts of the LCC framework

Roadmap & Organization

Roadmap

- * - July ‘16: Finish COALA IP spec.
- July ‘16 - Sept. ‘16: COALA IP reference implementation
- After Sept. ‘16:
    - Use COALA IP reference impl. in prod. systems
    - Take COALA IP specs to IETF/W3C/???

Organization

- Soon, there will be:
  - A spec hosted publicly on Github
  - A Github organization
  - A mailing list

⇒ Interested? Feedback? Ideas?: Please talk to us!

Thanks for listening!
Questions?