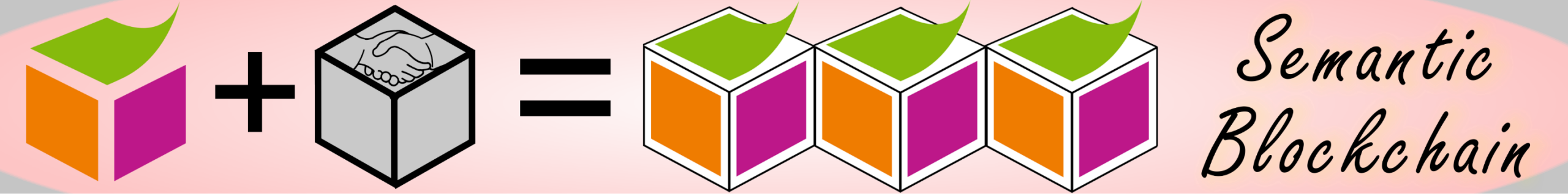


Semantic Blockchain

(STARTING POINT)

Semantic Web

ethereum

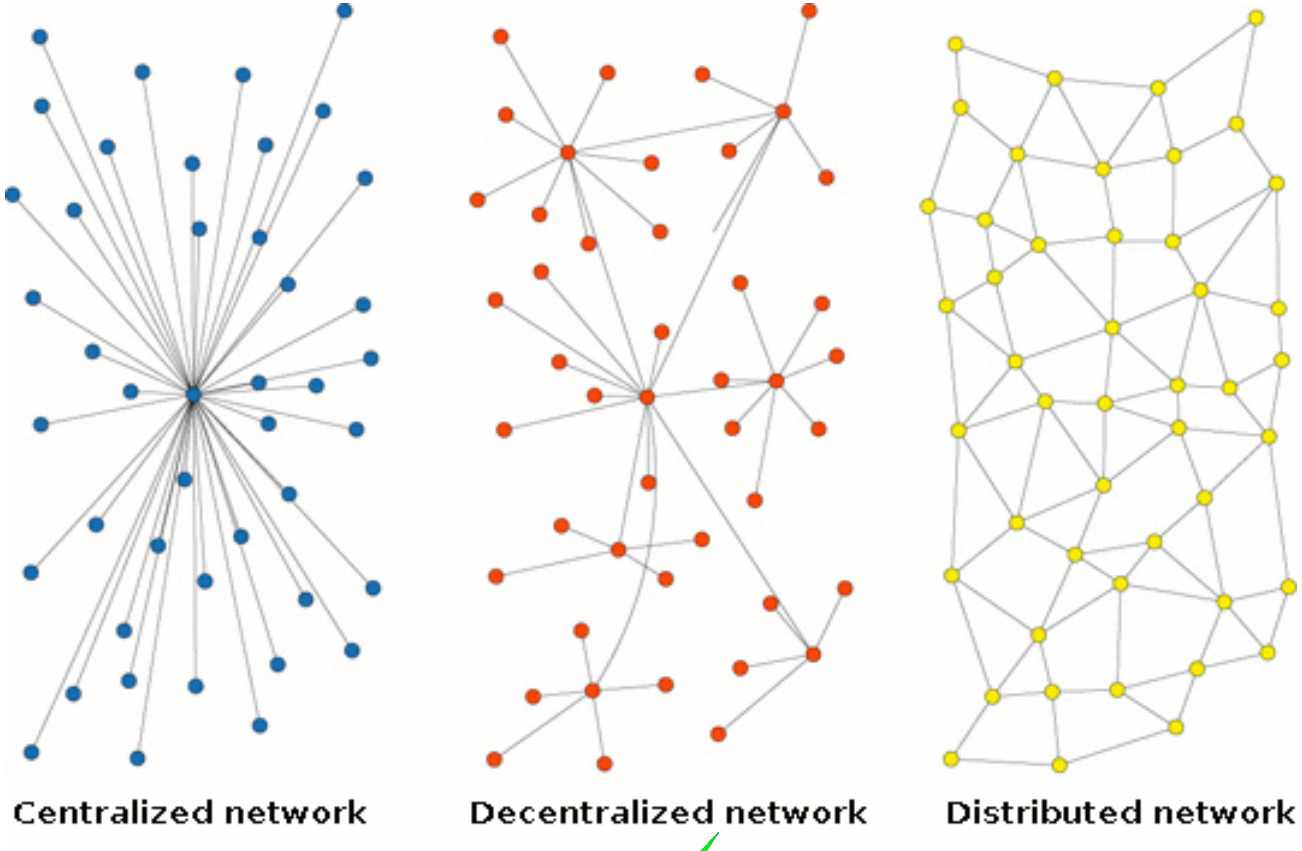
By: Héctor E. Ugarte R.

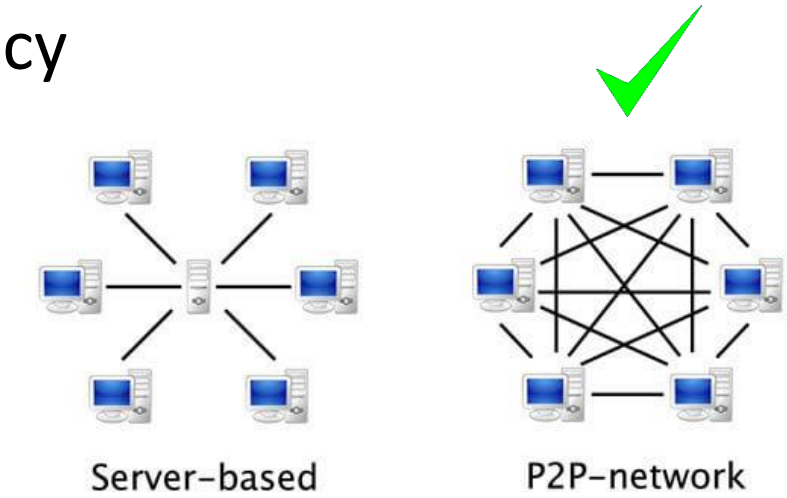
https://semanticblocks.wordpress.com/

1. BACKGROUND

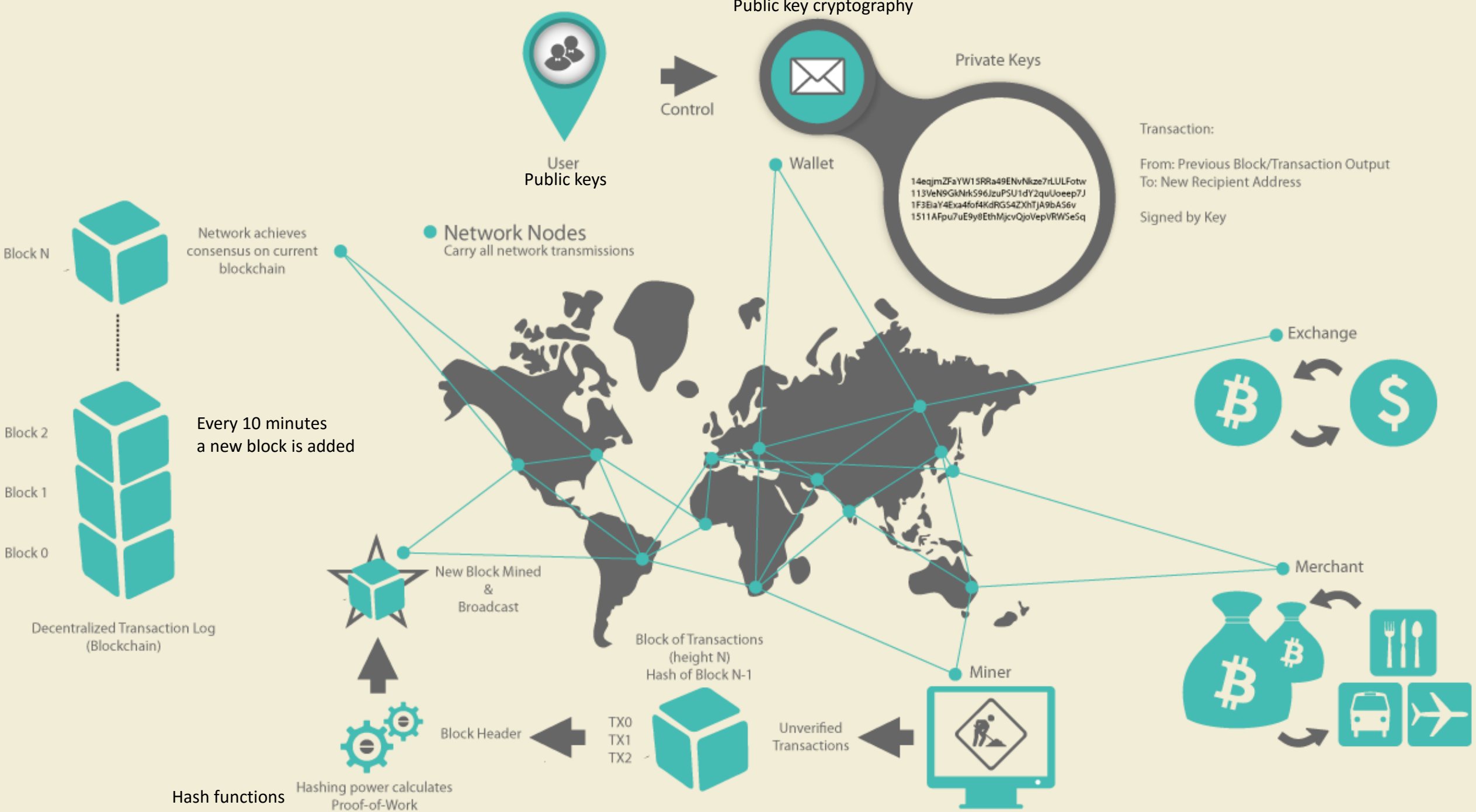
Bitcoin 1.0 bitcoin

- The world's first decentralized digital currency





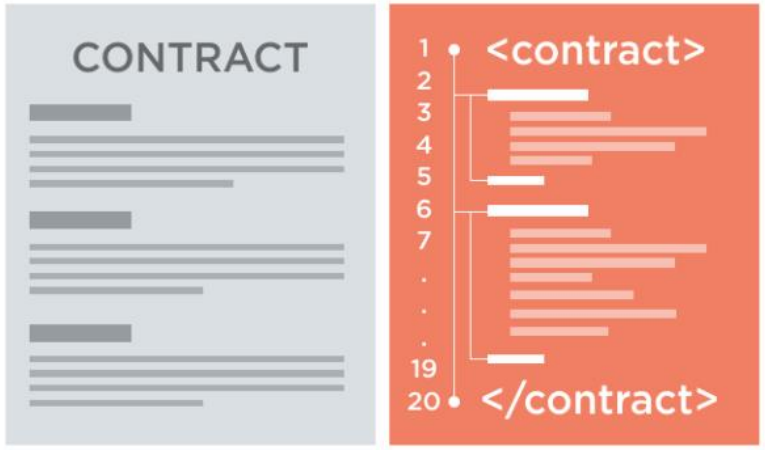
- A solution to the central authority problem
- Achieve concensus without a central authority



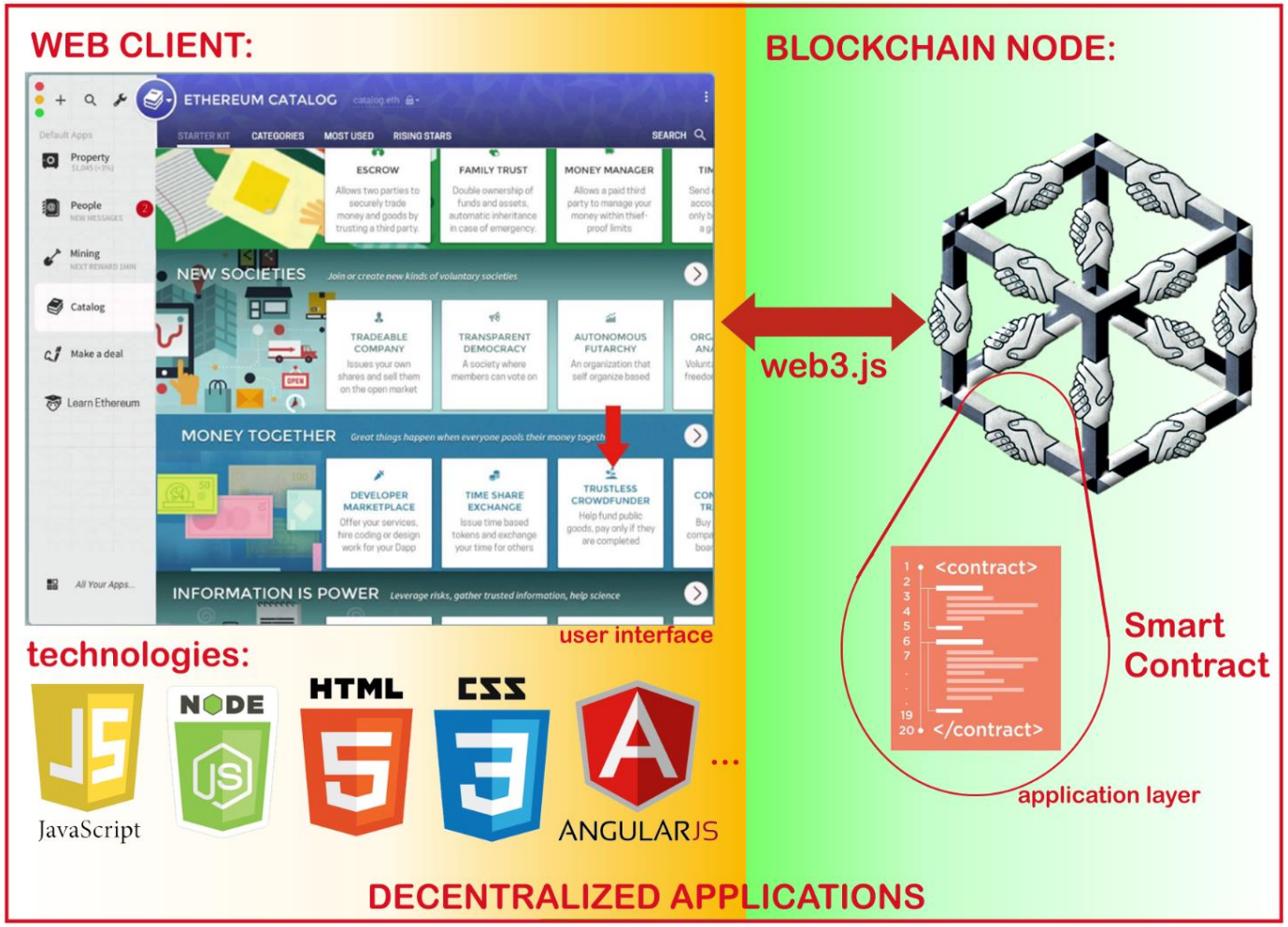
SOURCE: Andreas Antonopoulos. Mastering bitcoin: Unlocking digital cryptocurrencies. O'Reilly,Sebastopol, CA, 2015.

Bitcoin 2.0: ethereum

- The application of decentralized public ledgers for purposes other than just digital currencies.
- There are many (metacoins and others), but Ethereum includes a programmable smart contract platform.
- SMART CONTRACT: programs and protocols to facilitate the automated performance of a contract. Different possible languages, supported one: SOLIDITY



Ethereum Decentralized Application (DApp)



SOURCE: Hector Ugarte. Semantic web on/with the blockchain - Multilayered architecture on Ethereum, https://semanticblocks.wordpress.com/2016/03/29/multilayered-architecture-on-ethereum/ 2016.

2. ANALYSIS

Why Ethereum over Bitcoin? (metacoins)

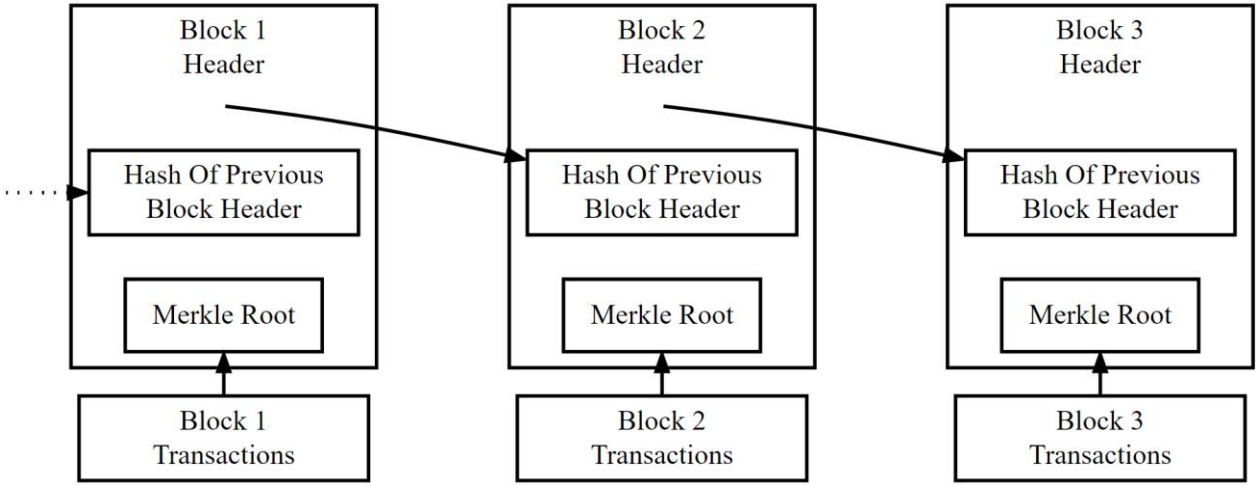
- General purpose.
- Turing-complete, universal scripting language.
- Fees - regulate its Turing-complete functionality and prevent abusive transactions transaction fee for each computational step of script execution.
- Mining algorithms: Different more energy friendly “Proof of stake”
- GHOST - GHOST is a new block propagation protocol that allows blockchains to have much faster block confirmation times, ideally in the range of 3-30 seconds.

Why blockchain technologies?

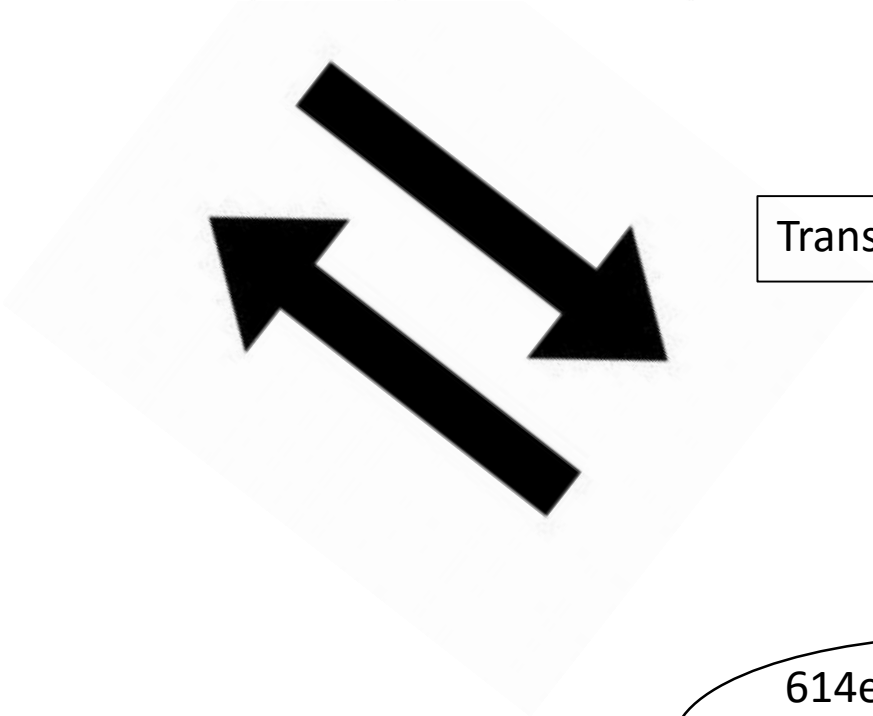
- Interoperable/Robust: A modular, interoperable platform that eliminates the possibility of double spending.
- Cost-efficient/Reduce costs: A solution to drastically reduce costs by eliminating the need for “handling companies” to be audited
- Real-time and agile: A fast and highly accessible sign-up means quick deployment
- Public/Transparency/censorship resilent/ Incorruptible/ Inmutable / Auditable/Traceability : The openness of the platform enables innovation and could achieve bottom-up transparency in supply chains instead of burdensome top-down audits. Ethics can be hard-coded. An auditable record that can be inspected and used by companies, standards organizations, regulators, and customers alike
- Guaranteed continuity: The elimination of any central operator ensures inclusiveness and longevity.
- Trustless
- Super distributed Security: You would have to hack each node in the chain to hijack the system
- proof of uniqueness
- ownership of data

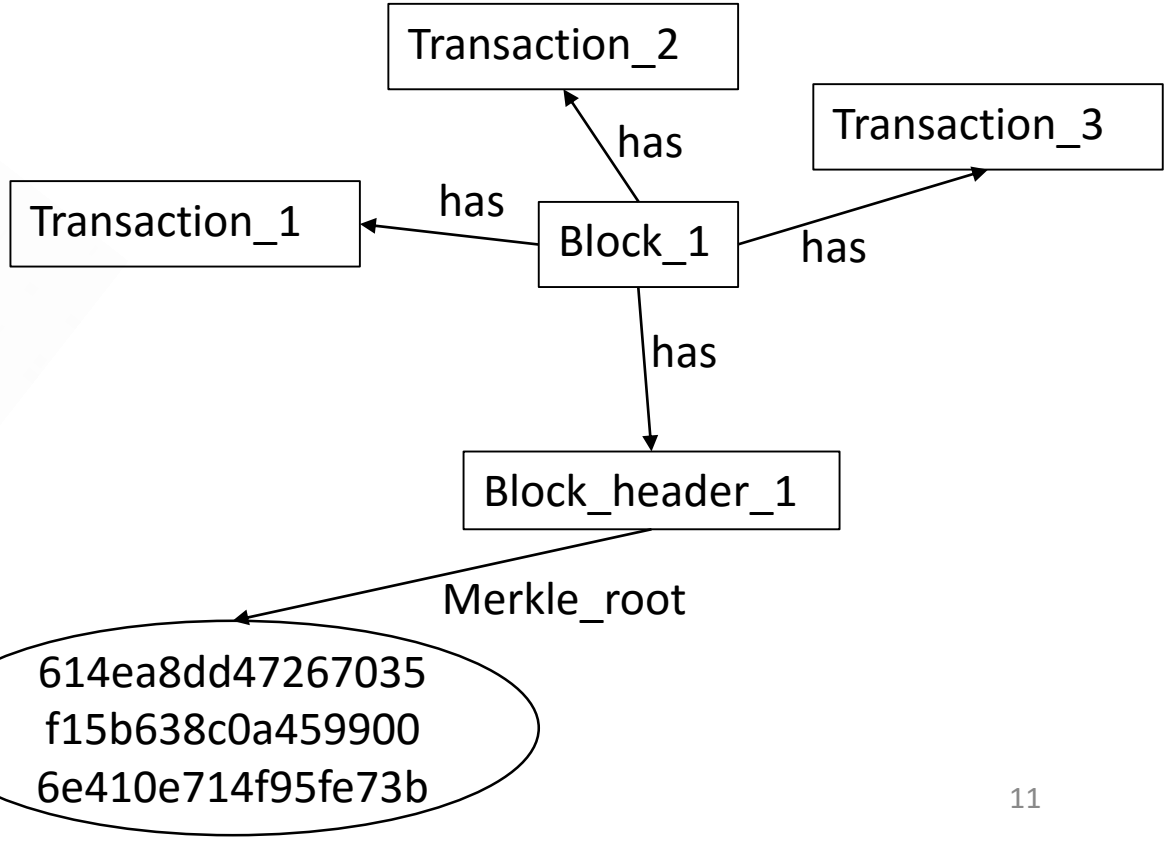
Why Semantic Web?

- Uniform Resource Identifiers (URIs) used to identify documents and also concepts (people, places, things, abstract/intangible concepts) and properties / data relationships **(consistency).**
- Resource Description Framework (RDF) provides a W3C standard way to write simple logical statements about relationships. **(Standardization).**
- Ontologies are like data dictionaries with additional logical annotations Multiple ontologies can co-exist and be used in parallel. It's also easy to cross-reference between them. **(Standarization).**
- SPARQL query language enables a query to combine machine-readable data from multiple sources and also allows new data relationships to be constructed from existing data. **(Linking and mappings).**



The Symbiosis





3. PROPOSED SOLUTION

Proposed solutions

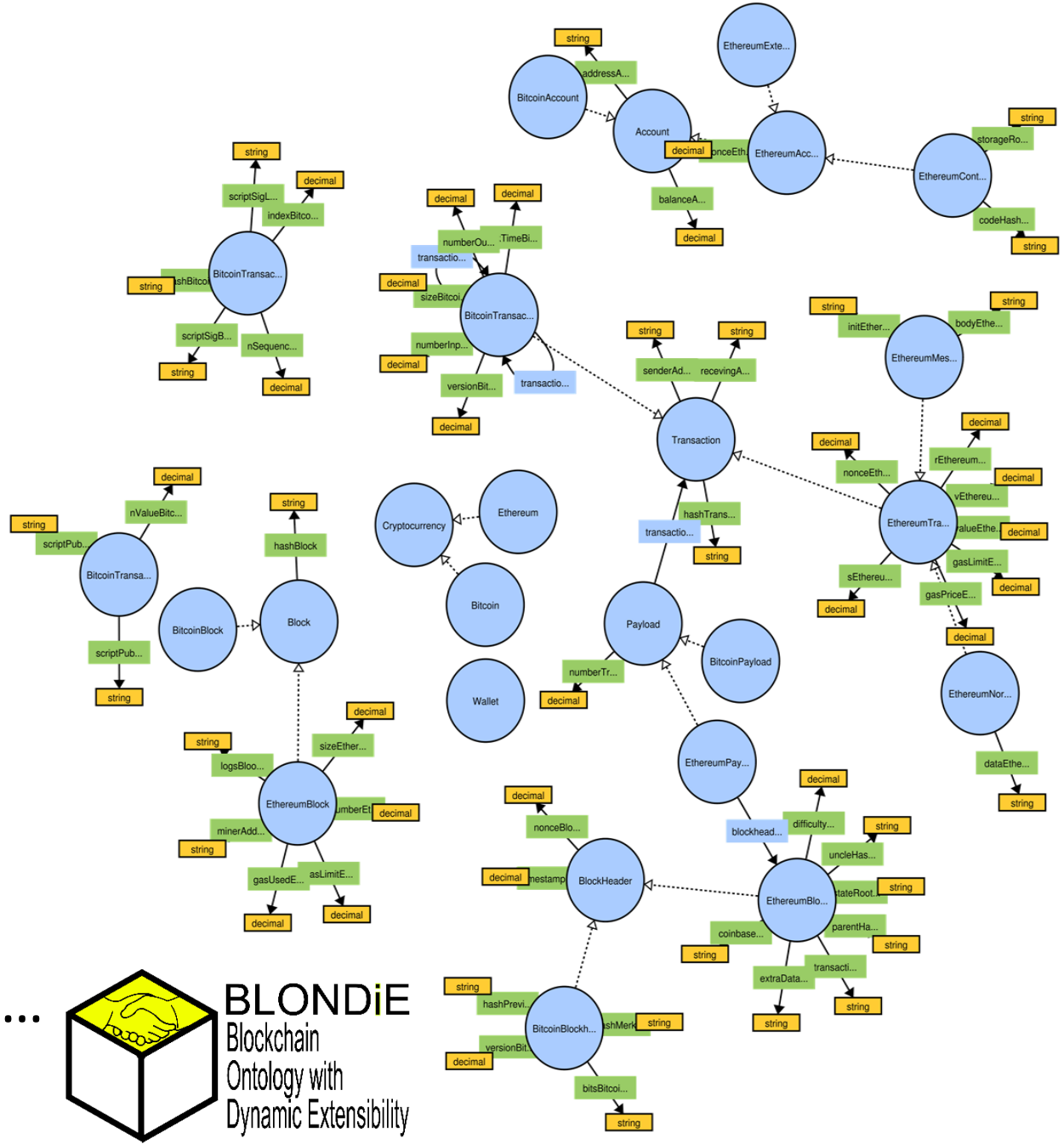
1. Developed **Blondie** ontology to explore Bitcoin and Ethereum blockchain

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.example.com/blondie.owl>
  a owl:Ontology ;
  rdfs:comment """BLONDIE: Bitcoin Ontology with Dynamic Extensibility
Ontology for Cryptocurrencies""" .

<http://www.example.com/blondie.owl#blockheaderListEthereumPayload>
  a owl:ObjectProperty ;
  rdfs:domain <http://www.example.com/blondie.owl#EthereumPayload> ;
  rdfs:range <http://www.example.com/blondie.owl#EthereumBlockheader> .

<http://www.example.com/blondie.owl#transactionInputListBitcoinTransaction>
  a owl:ObjectProperty ;
  rdfs:domain <http://www.example.com/blondie.owl#BitcoinTransaction> ;
  rdfs:range <http://www.example.com/blondie.owl#BitcoinTransaction> .
```



Proposed solutions

Evaluation:

A basic mapping between JSON RPC (using Ethereum JSON RPC API) and RDF (using **Blondie** vocabulary)

```
// Result
{
"id":1,
"jsonrpc":"2.0",
"result": {
    "number": "0x1b4", // 436
    "hash": "0xe670ec64341771606e55d6b4ca35a1a6b75ee3d5145a99d05921026d1527331",
    "parentHash": "0x9646252be9520f6e71339a8df9c55e4d7619deeb018d2a3f2d21fc165dde5eb5",
    "nonce": "0xe04d296d2460cfb8472af2c5fd05b5a214109c25688d3704aed5484f9a7792f2",
    "sha3Uncles": "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
    "logsBloom": "0xe670ec64341771606e55d6b4ca35a1a6b75ee3d5145a99d05921026d1527331",
    "transactionsRoot": "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421
    "stateRoot": "0xd5855eb08b3387c0af375e9cdb6acfc05eb8f519e419b874b6ff2ffda7ed1dff",
```

```
:Blockheader01 a :EthereumBlockheader;
        :parentHashEthereumBlockhader "0x9646252be9520f6e71339a8df9c55e4d7619deeb018d2a3f221fc165dde5eb5" ;
:stateRootEthereumBlockheader "0x56e81f171bcc55a6fff8345e692c0f86e5b48e01b996cadc001622fb5e363b421".
```

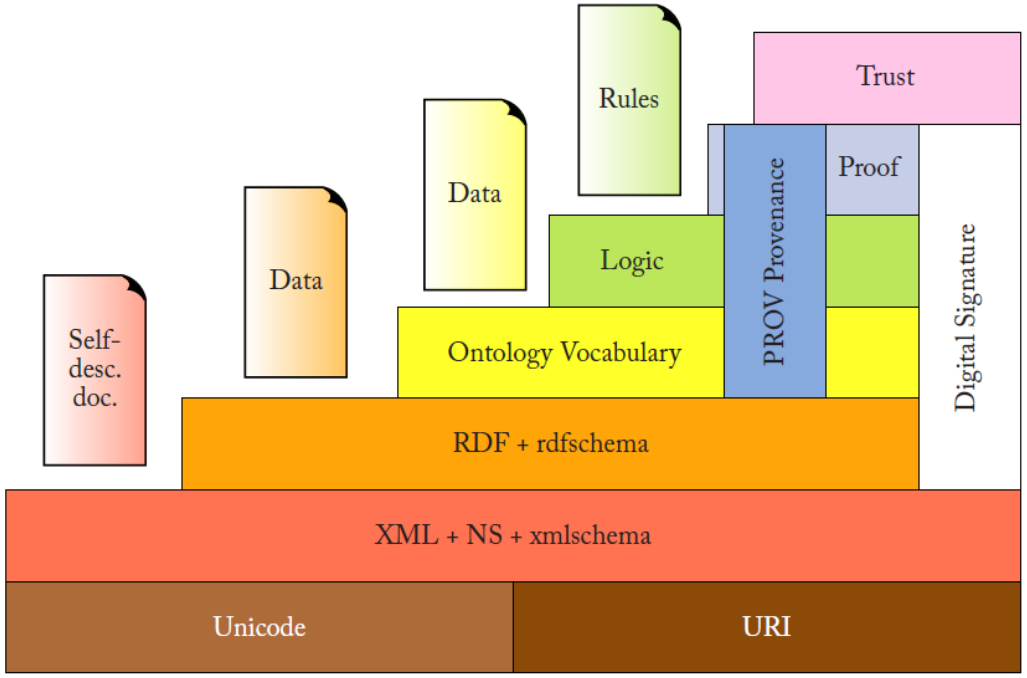
Proposed solutions

2. Implement a usecase on Ethereum

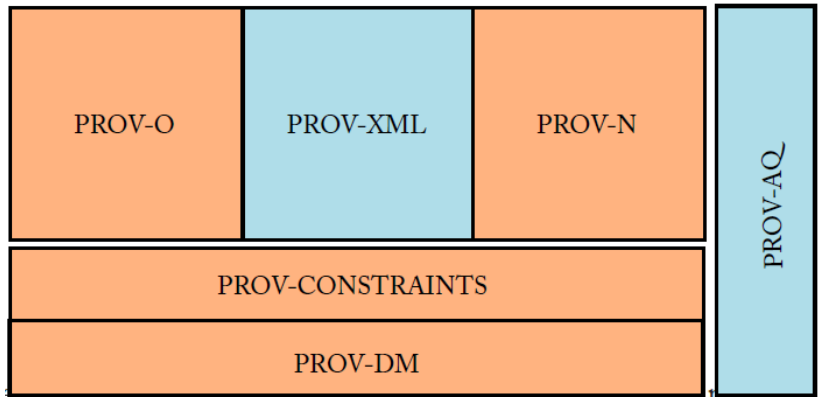
Chosen usecase: “PROVENANCE on the Supply Chain”

“Provenance is defined as a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing.”

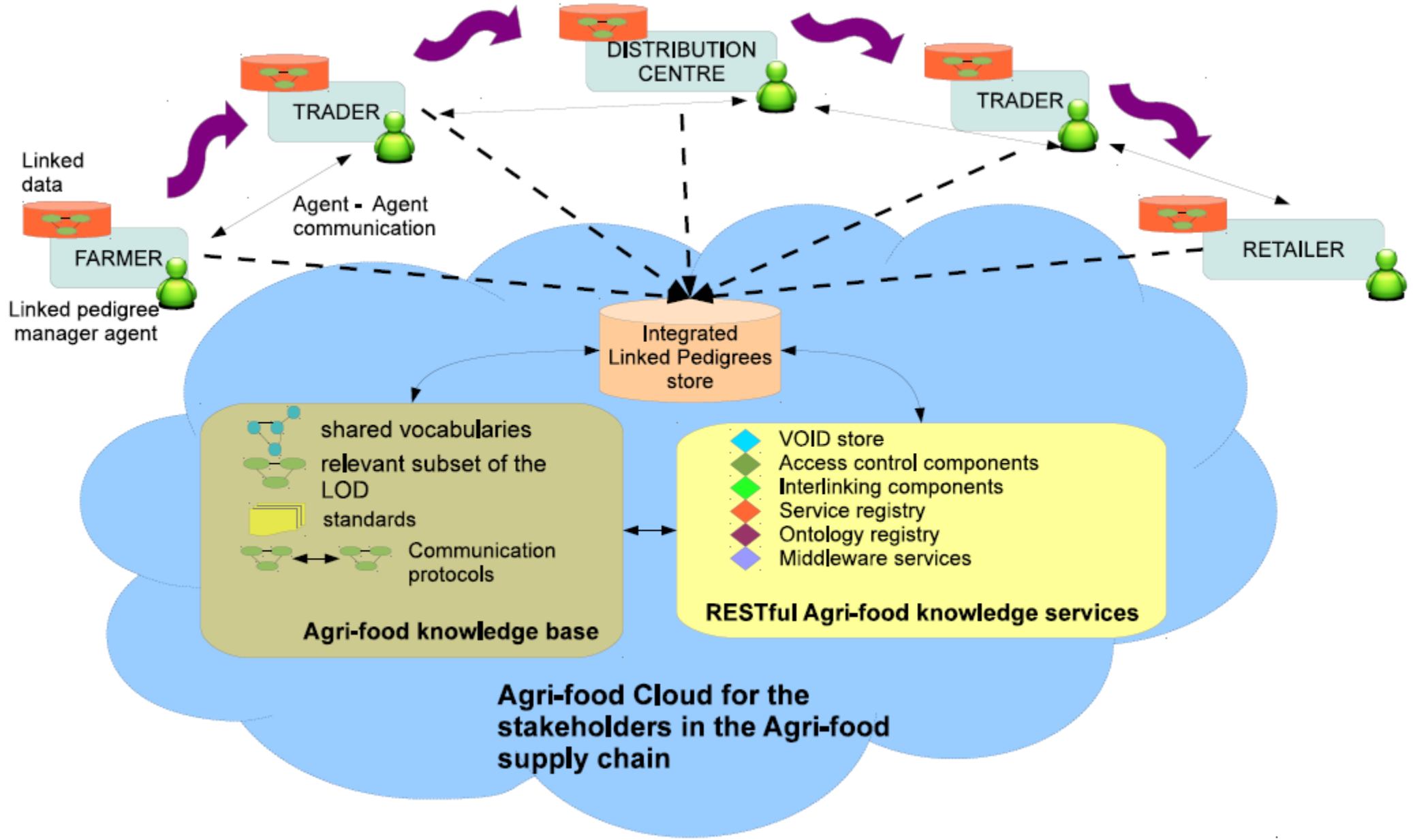
W3C (World Wide Web Consortium) Provenance Working Group’s



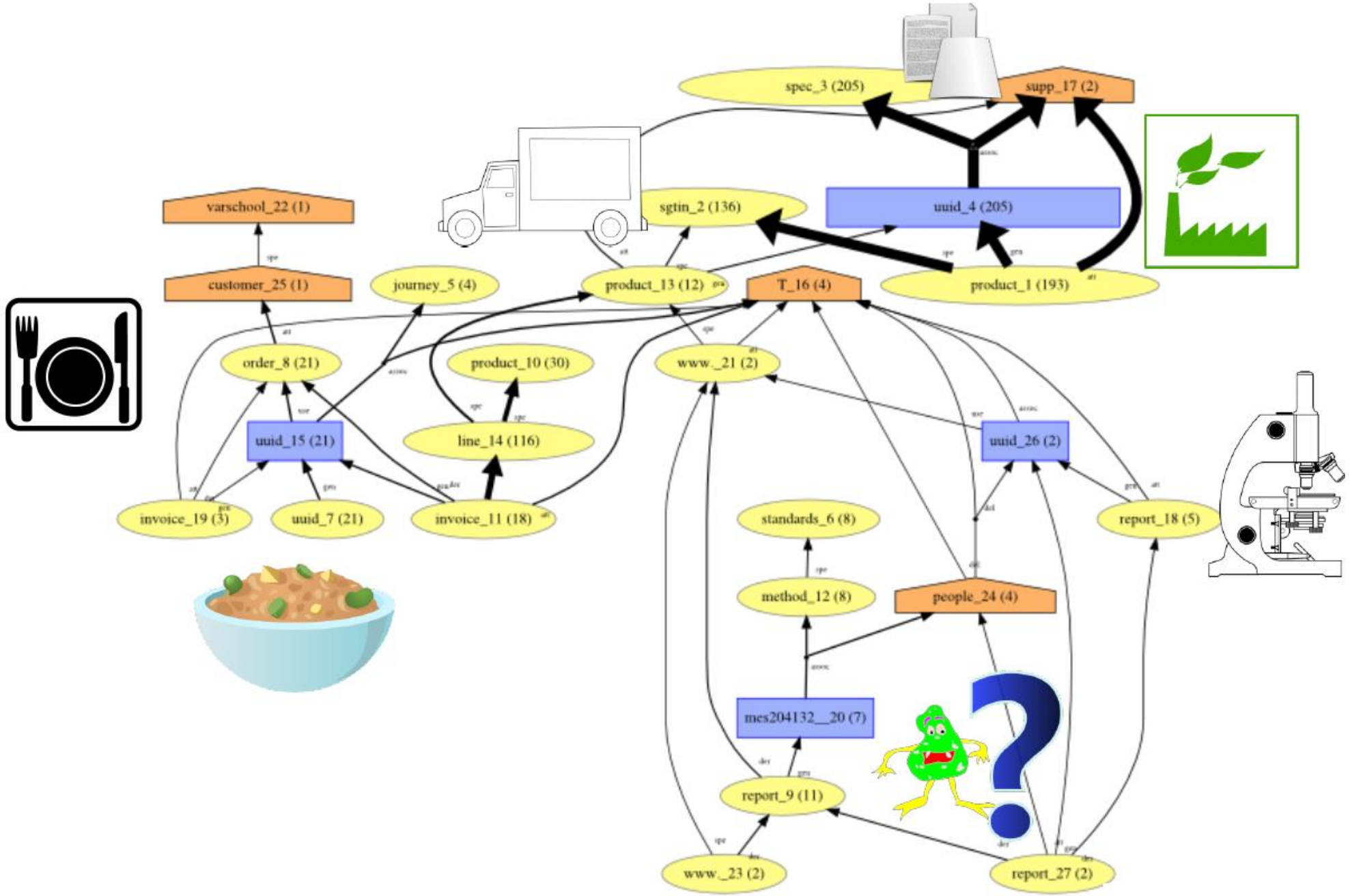
Provenance in the Semantic Web Layer Cake Diagram.



Simplified view of PROV specifications



SOURCE: Christopher Brewster. Semantic blockchains in the supply chain. Aston University, 2015.

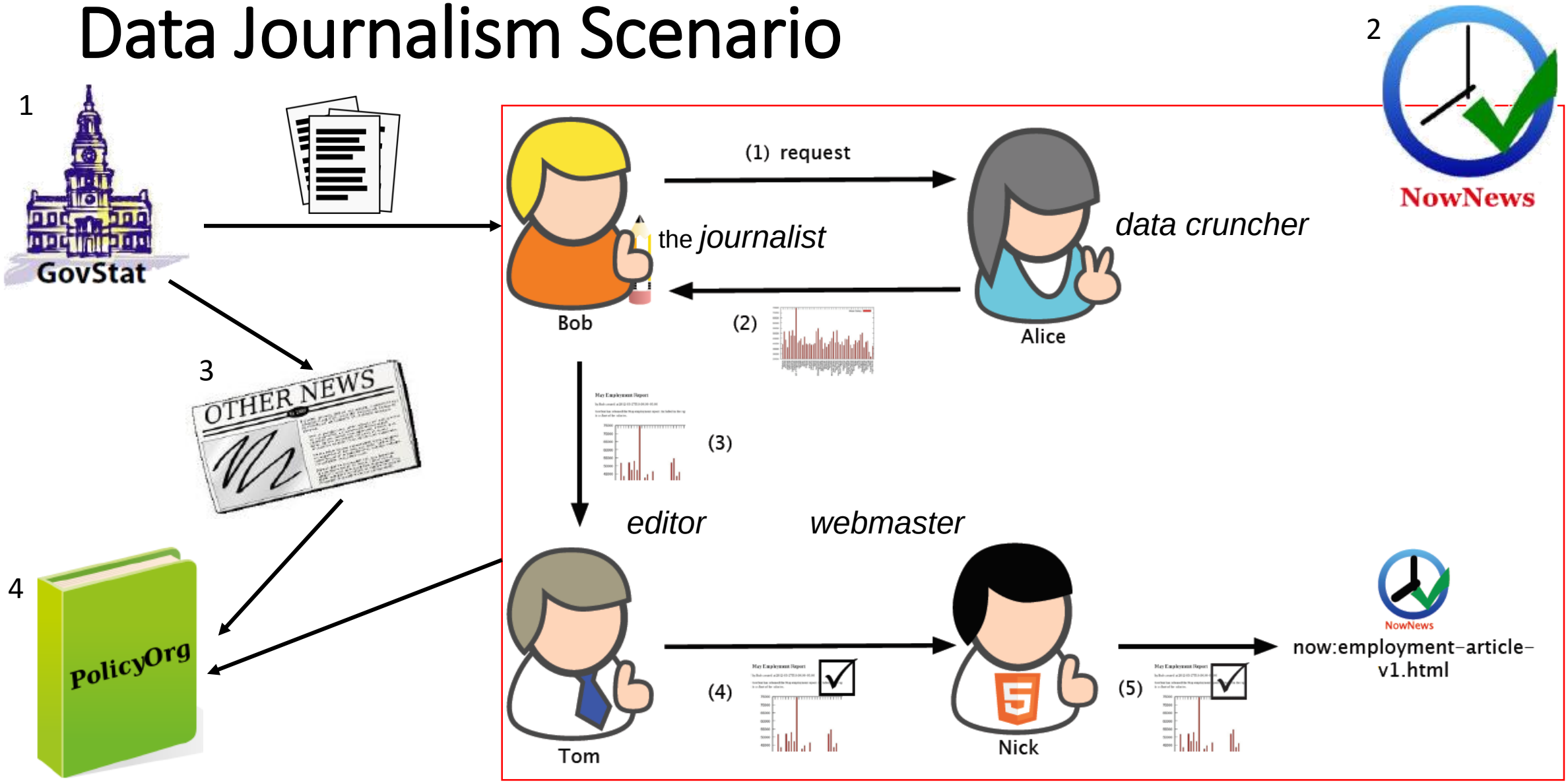


SOURCE: Luc Moreau. Enabling Provenance on the Web Standardization and Research Questions. Web and Internet Science. University of Southampton, 2015.

The Project Provenance Ltd

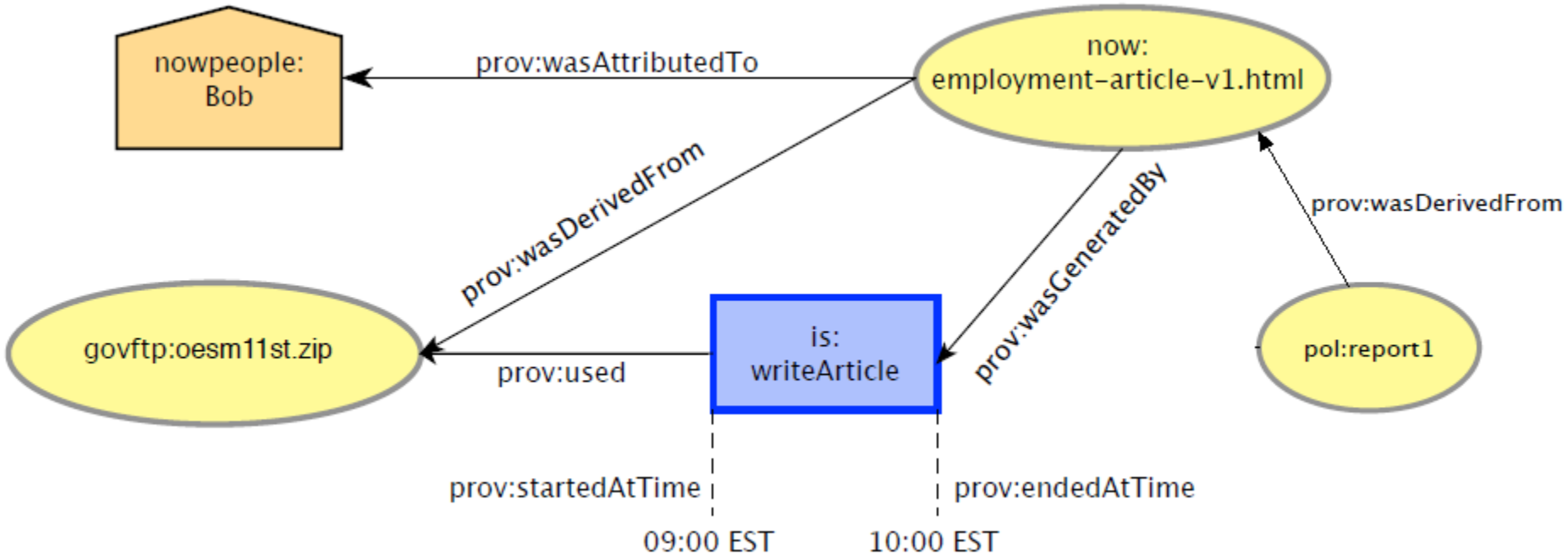
- Prototype that uses block chain technology to enable secure traceability of certifications and other salient information in supply chains.

Data Journalism Scenario



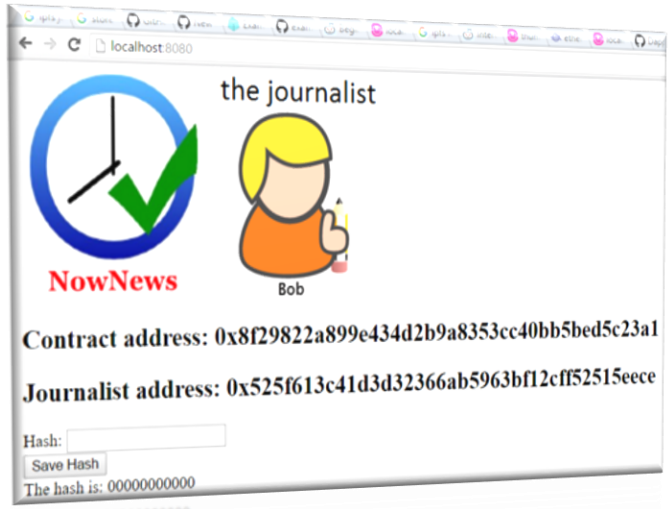
SOURCE: Luc Moreau, Paul Groth. Provenance : an introduction to PROV. Morgan & Claypool, 2013.

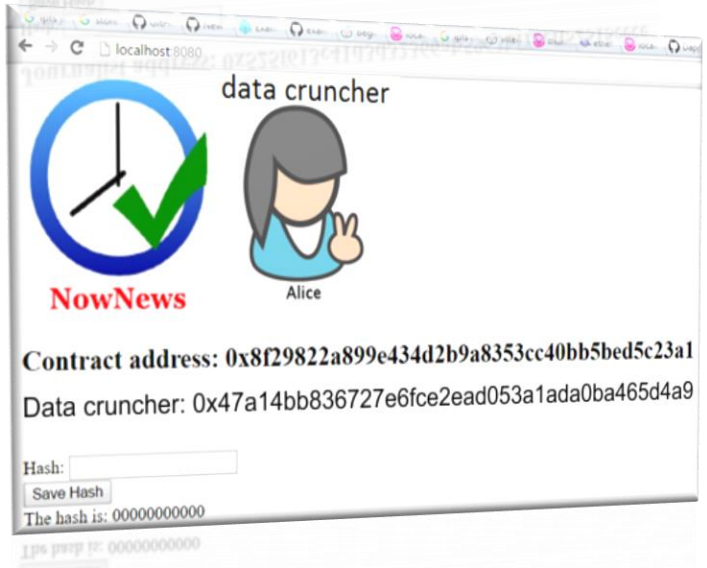
Individuals diagram

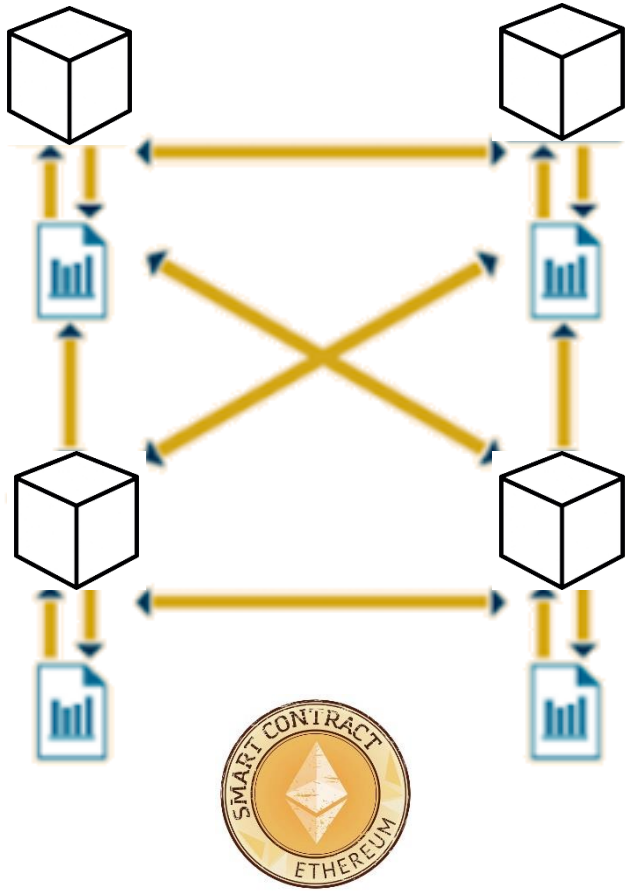


Using The PROV Ontology (PROV-O)

Data Journalism Đapp Schema







Future Work

- Ethereum Contract ABI (Application binary interface)

```
contract Test {
function Test(){ b = 0x1234567890123456789012345678901 2; }
event Event(uint indexed a, bytes32 b)
event Event2(uint indexed a, bytes32 b)
function foo(uint a) { Event(a, b); }
bytes32 b;
}
```


```
[{
"type":"event",
"inputs": [{"name":"a","type":"uint256","indexed":true},{"name":"b","type":"bytes32","indexed":false}],
"name":"Event"
}, {
"type":"event",
"inputs": [{"name":"a","type":"uint256","indexed":true},{"name":"b","type":"bytes32","indexed":false}],
"name":"Event2"
}, {
"type":"event",
"inputs": [{"name":"a","type":"uint256","indexed":true},{"name":"b","type":"bytes32","indexed":false}],
"name":"Event2"
}, {
"type":"function",
"inputs": [{"name":"a","type":"uint256"}],
"name":"foo",
"outputs": []
}]
```

?

Future Work

- Ethereum Natural Specification Format

```
/// @notice Send `(valueInmGAV / 1000).fixed(0,3)` GAV from the account of
/// `message.caller.address()`, to an account accessible only by `to.address()
/// @dev This should be the documentation of the function for the developer docs
/// @param to The address of the recipient of the GavCoin
/// @param valueInmGav The GavCoin value to send
function send(address to, uint256 valueInmGAV) {
  if (balances[message.caller] >= valueInmGAV) {
    balances[to] += valueInmGAV;
    balances[message.caller] -= valueInmGAV;
  }
```

•@title: This is a title that should describe the contract and go above the contract definition
•@author: The name of the author of the contract. Should also go above the contract definition.
•@notice: Represents user documentation. This is the text that will appear to the user to notify him of what the function he is about to execute is doing
•@dev: Represents developer documentation. This is documentation that would only be visible to the developer.
•@param: Documents a parameter just like in doxygen. Has to be followed by the parameter name.
•@return: Documents the return type of a contract's function.