Graphs in OpenCog, Webstructor, Aigents and other places and cases

Anton Kolonin
SingularityNET
2018, May

Problem of Graph Visualization:

Trying to make it **useful**,
we make it **complex**
making it **not usable**.

Visual Representation Options

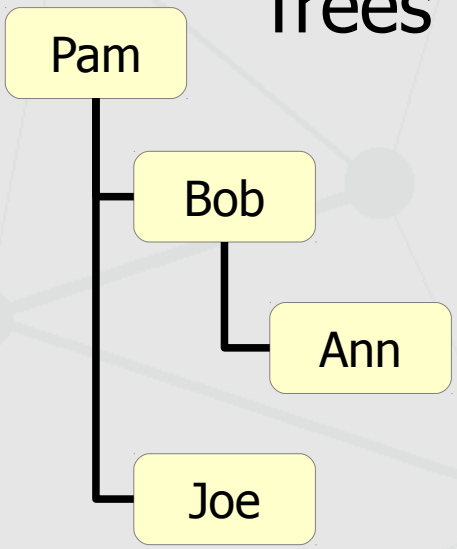
Tables

Person	Parent
Pam	
Bob	Pam
Joe	Pam
Ann	Bob

Person	Uncle
Ann	Joe

Not readable?

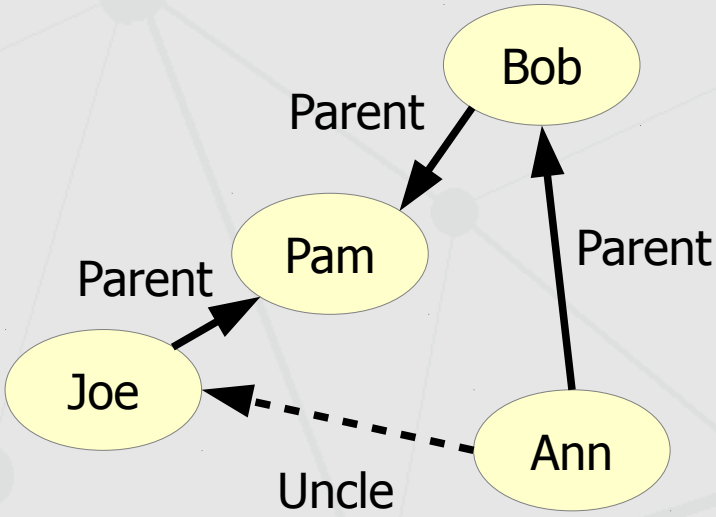
Trees



Where is Uncle?

Too messy?

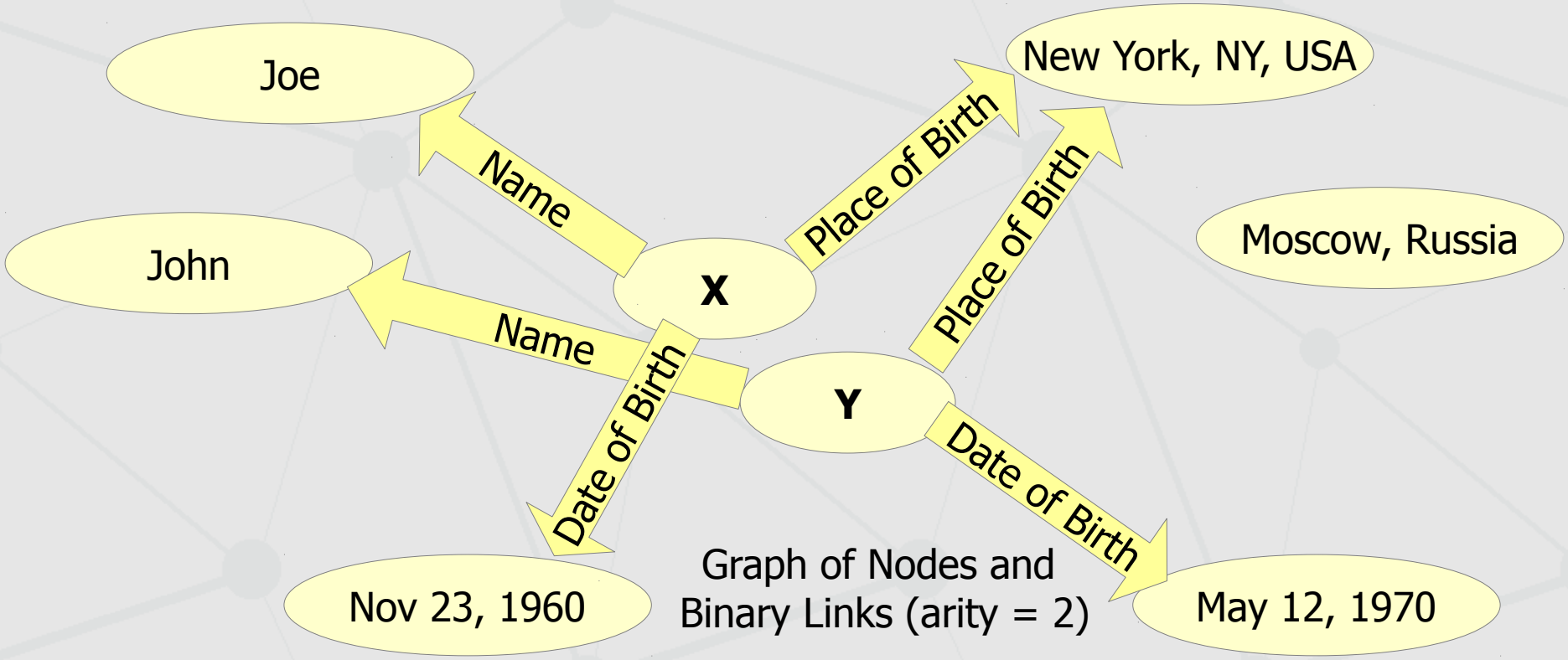
Graphs



Relational Database vs. Graph

Table of relations in relational database

Name	Date of Birth	Place of Birth
Joe	May 12, 1970	New York, NY, USA
John	Nov 23, 1960	New York, NY, USA



Graph of Nodes and Binary Links (arity = 2)

Graph as Relational Database

Nodes

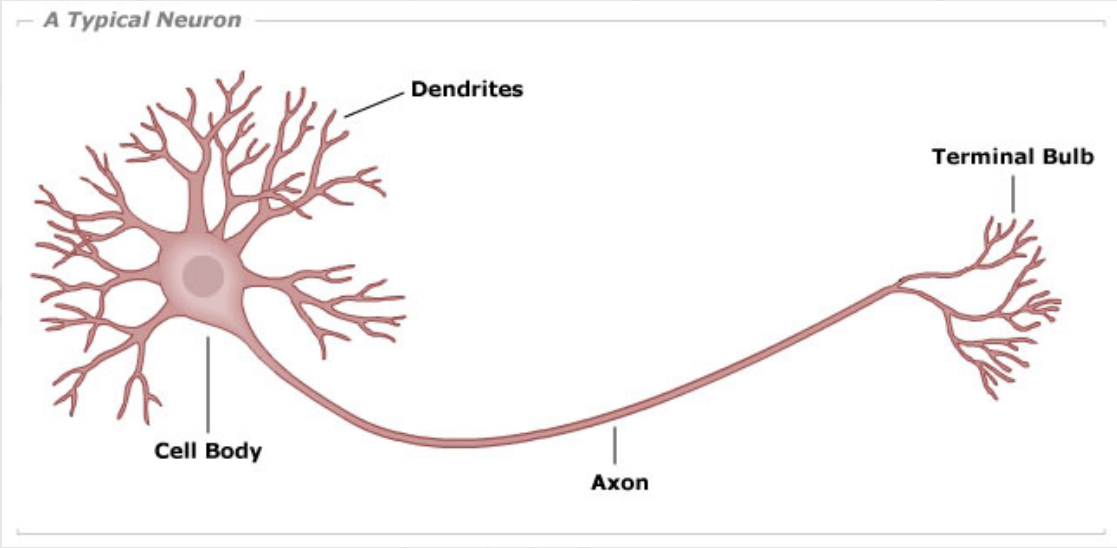
Id	Label
1	X
2	Y
3	John
4	Joe
5	Nov 23, 1960
6	May 12, 1970
7	New York, NY, USA
8	Moscow, Russia

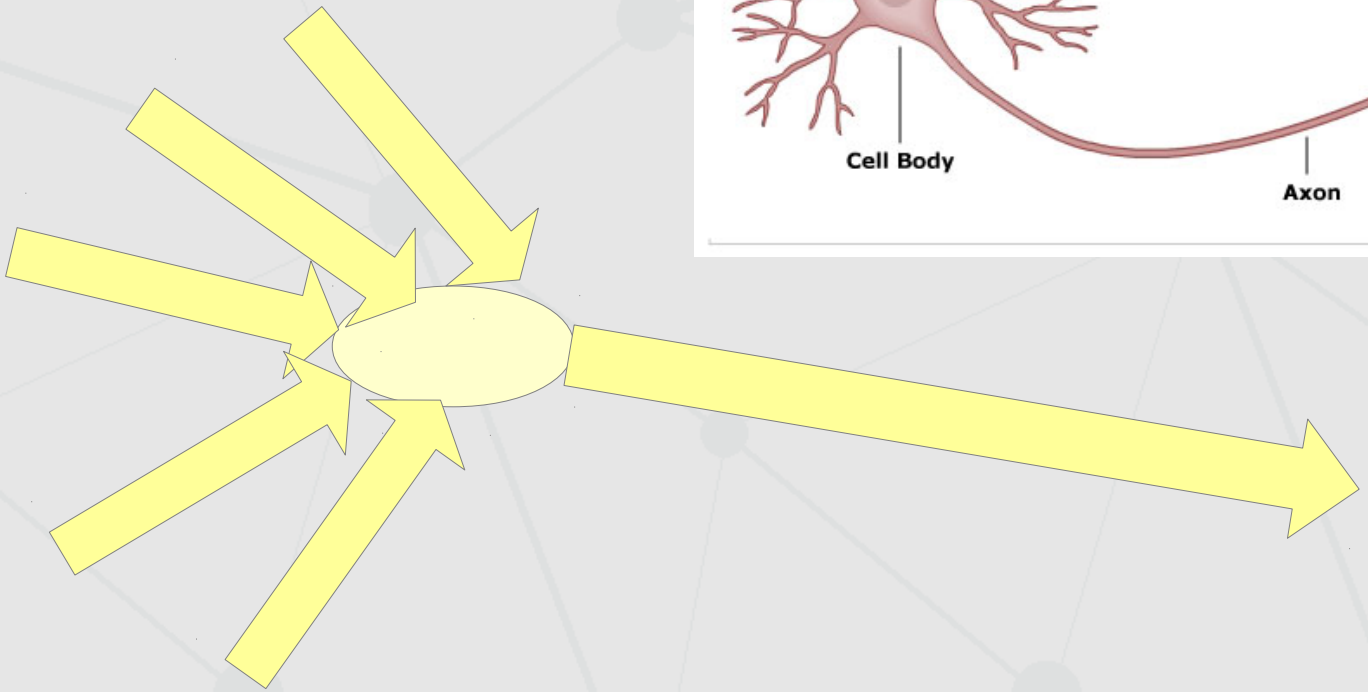
Binary Links

From	To	Label
1	4	Name
2	3	Name
1	5	Date of Birth
2	6	Date of Birth
1	7	Place of Birth
1	7	Place of Birth

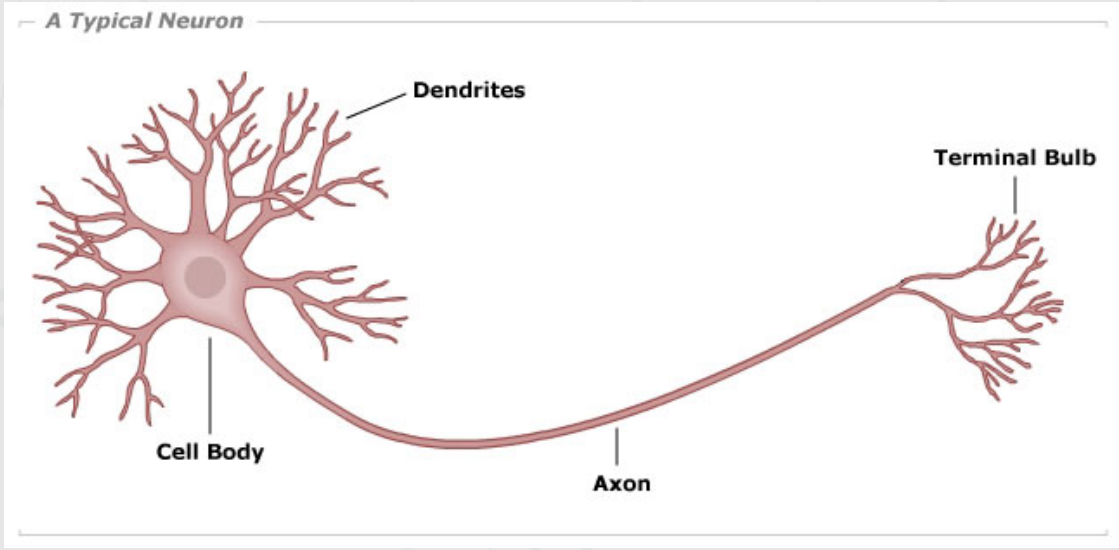
Binary links in relational database are connecting relations while tables holding these links are called relationships

Neuron vs. Graph: Node and Links

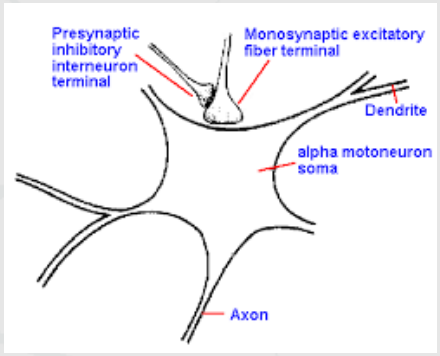




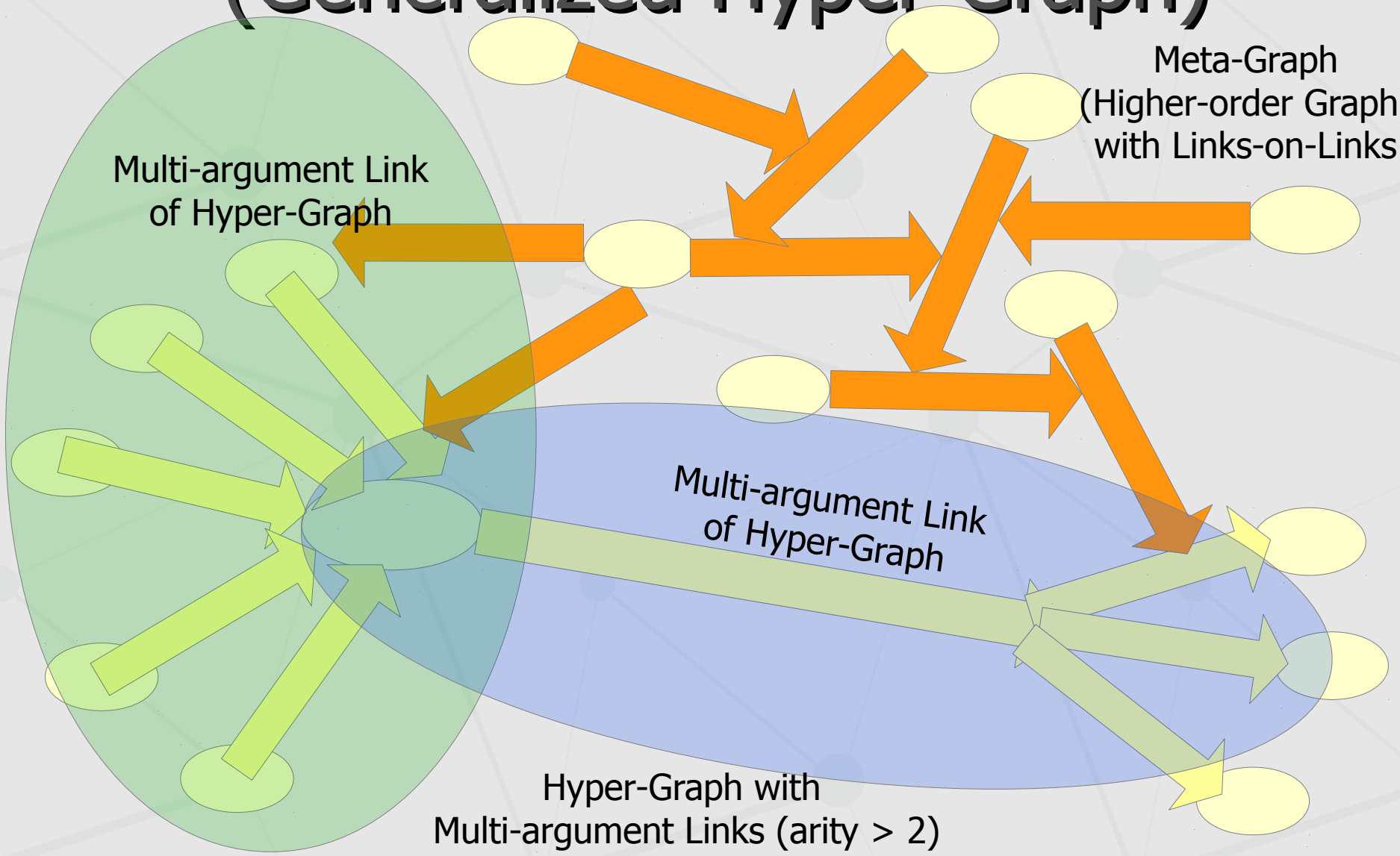
Neuron vs.Graph:Dendrites & Axons - Multi-argument Links (arity > 2)



Neuron vs. Graph: Synapses on Synapses – Links-on-Links

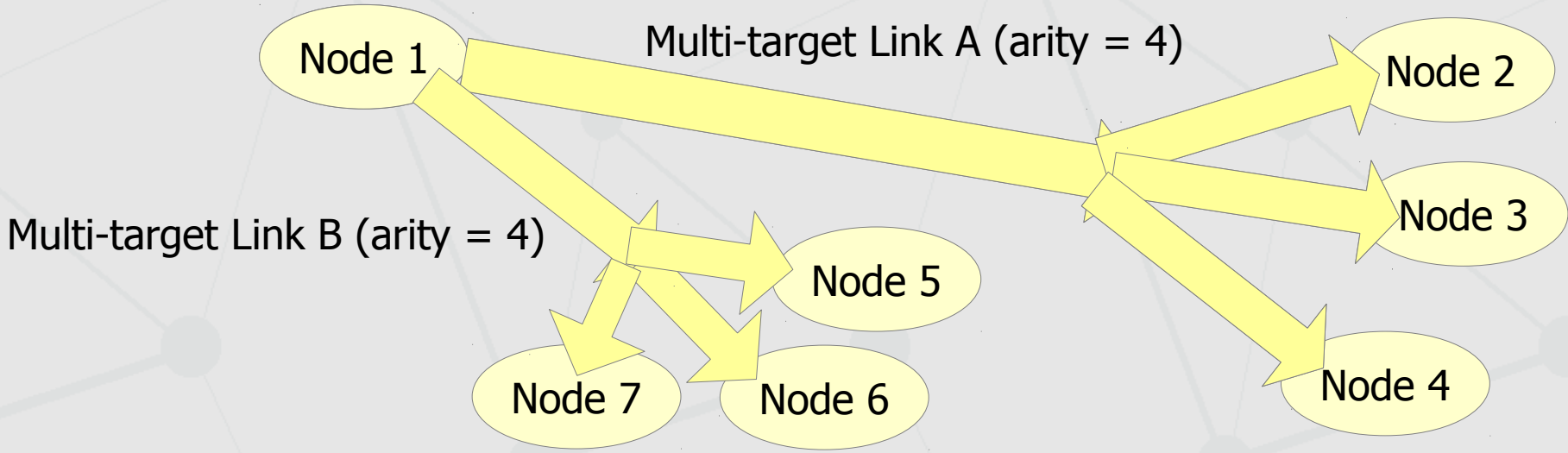


Hyper-Graph and Meta-Graph (Generalized Hyper-Graph)

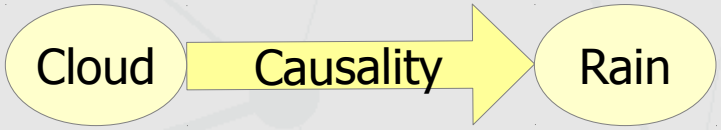


N-ary Multi-argument Links as N-ary Relations in Relational database

	Argument 1	Argument 2	Argument 3	Argument 4
Multi-target Link A (arity = 4)	Node 1 source	Node 2 target	Node 3 target	Node 4 target
Multi-target Link B (arity = 4)	Node 1 source	Node 5 target	Node 6 target	Node 7 target

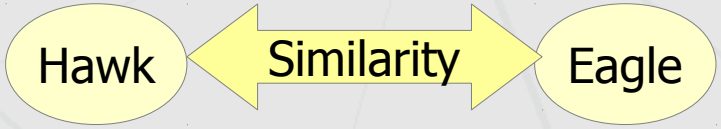


Ordered (directed) and Unordered (undirected) Links

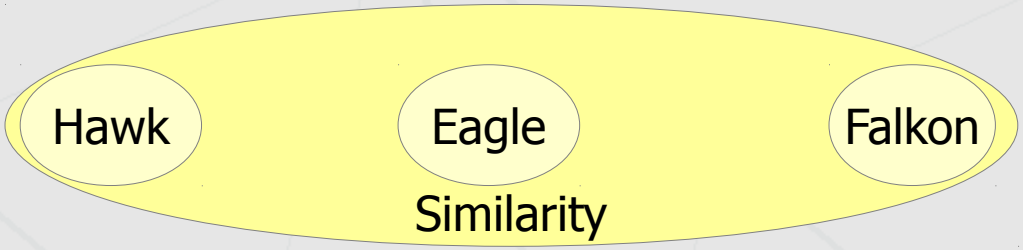


Ordered (directed) Link, arity = 2
Cloud is "source"
Rain is "target"
Real Link

Ordered (directed) Link, arity = 3
Cloud is "source"
Rain is "source" and "target"
Flood is "target"
Sequence or Series

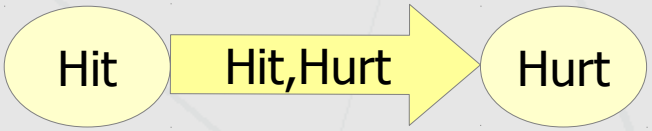


Unordered (undirected) Link, arity = 2
Just Pair

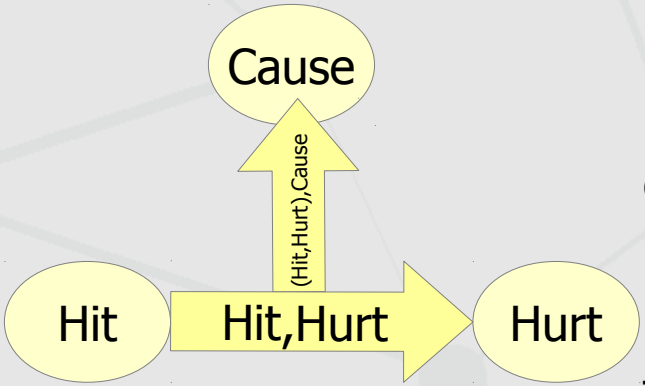


Unordered (undirected) Link, arity = 3
Set

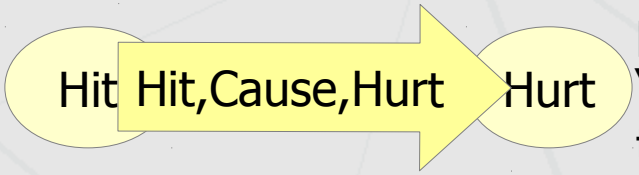
Link Type and Arity – ways to go



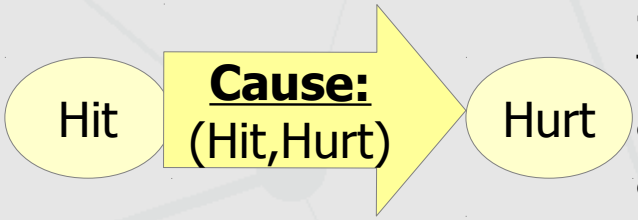
Binary typeless link (from, to) with arity = 2



Binary link (from, to) with arity = 2 where type is expressed with extra binary link to link type node (needs higher-order links-on-links capability)



Ternary triple (from, type, to) with arity = 3 where type is part of relation, corresponds to "term" subject-predicate-object in natural language processing and "triple" in most of conventional graph databases (RDF triplet stores)



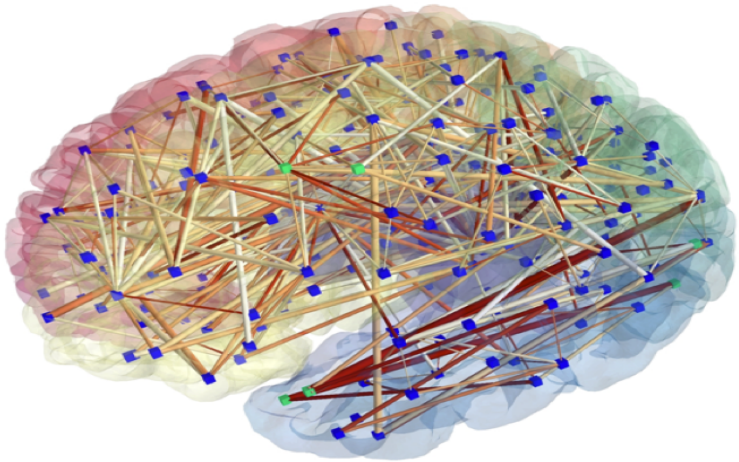
Separate predicate (type) and list of its arguments (from, to) with arity = 2 corresponds to "term" in predicate logic and RDF statement in modern semantic web programming and implemented in OpenCog and Webstructor, dealing with arbitrary arities, specific to those predicate types

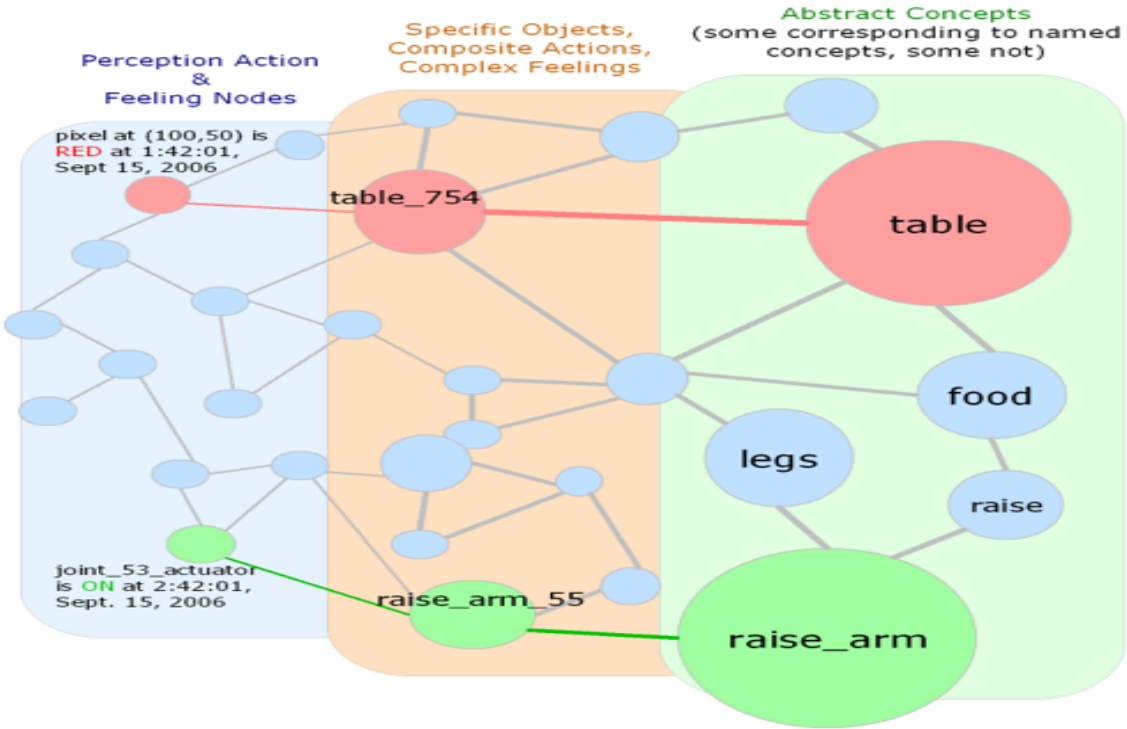
Takeaways, thus far

- Same data can be represented as set of relations in a relational database or nodes and links in a binary graph;
- Graph of binary links can be stored in relational database;
- Simplified neural net can be represented with nodes and binary links;
- More realistic neural net can be represented with multi-argument links;
- Each realistic neuron can be described as two multi-argument links;
- There is no single structure in graph theory representing real neuron;
- Realistic neural net involves links affecting other nodes, not links between nodes only;
- Direction of multi-arguments link and roles of its arguments are not determined by its structure and need extra knowledge or assumptions, involving order of arguments in the link as a tuple;
- Multi-argument link of N-arguments can be thought as N-ary relation of relational database;
- Different graph representations can be used to represent the same piece of information.

OpenCog's "AtomSpace"

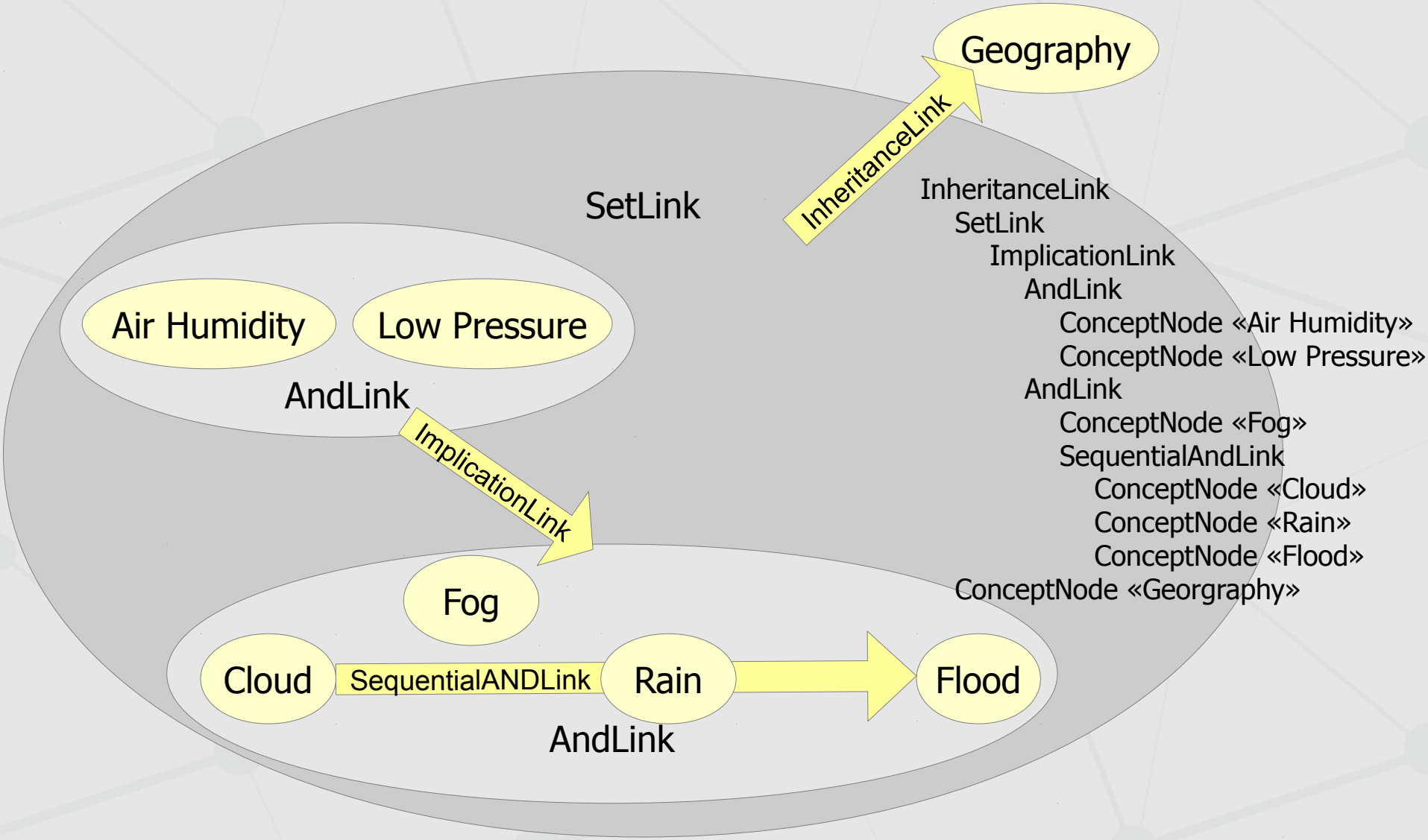
Implements Generalized Hyper-graph and Meta-graph, so each directed/undirected link may link together any number of atoms, where atom could be either node (arity = 0) or any other link with any arity, including unordered N-ary links representing subgraphs as their elements.



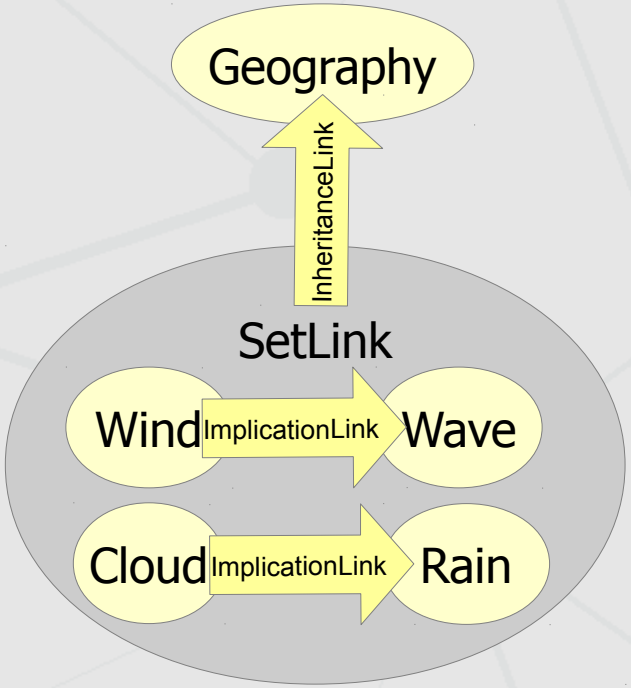


https://github.com/opencog/atomspace

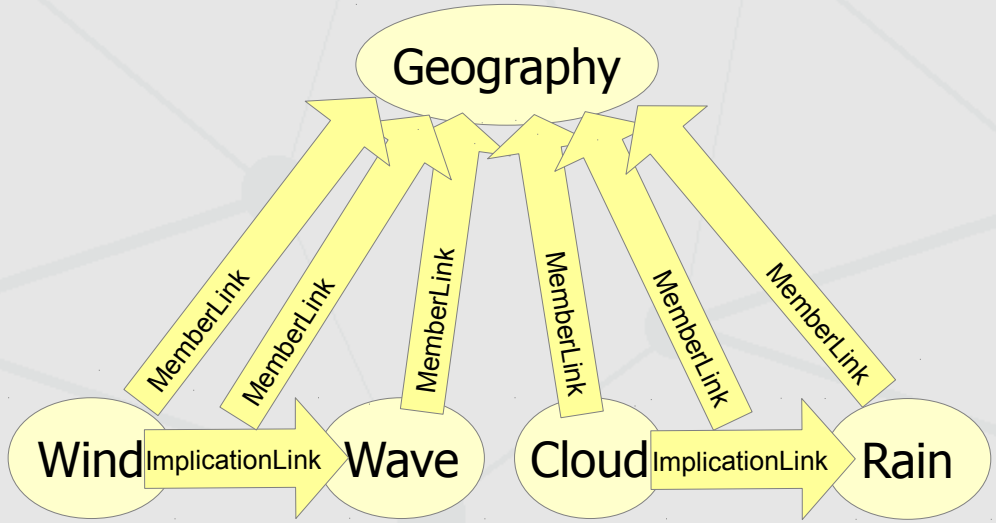
AtomSpace - Generalized (Link-as-Node) Hyper-Graph is a Meta-Graph (in Atomese)



AtomSpace - Meta-Graphs with Hyper-Graphs and Links-on-Links



InheritanceLink
  SetLink
    ImplicationLink
      ConceptNode «Wind»
      ConceptNode «Wave»
    ImplicationLink
      ConceptNode «Cloud»
      ConceptNode «Rain»
  ConceptNode «Georgraphy»



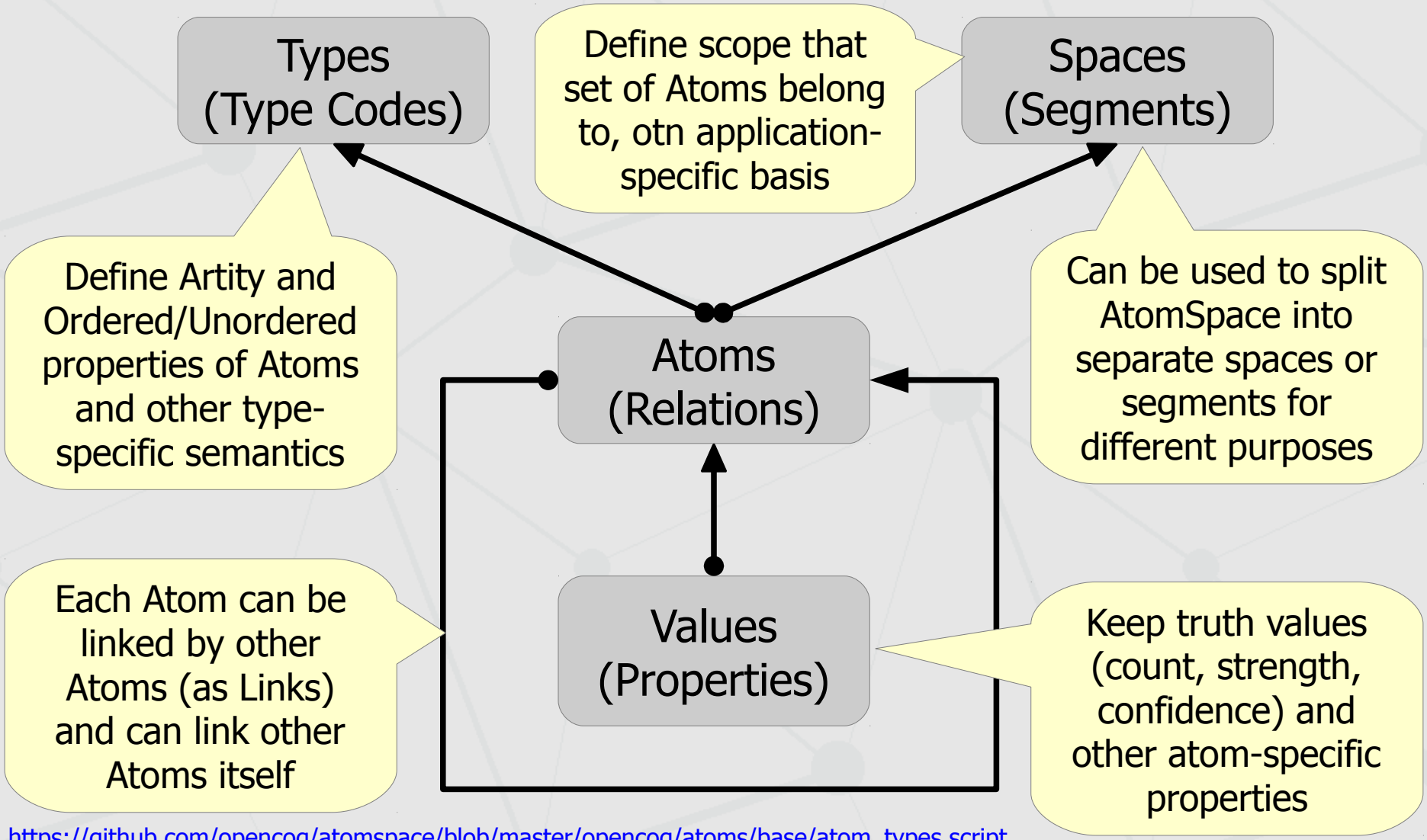
MemberLink
  ConceptNode «Wind»
  ConceptNode «Georgraphy»
MemberLink
  ConceptNode «Wave»
  ConceptNode «Georgraphy»

MemberLink
  ImplicationLink
    ConceptNode «Wind»
    ConceptNode «Wave»
  ConceptNode «Georgraphy»

MemberLink
  ConceptNode «Cloud»
  ConceptNode «Georgraphy»
MemberLink
  ConceptNode «Rain»
  ConceptNode «Georgraphy»

MemberLink
  ImplicationLink
    ConceptNode «Cloud»
    ConceptNode «Rain»
  ConceptNode «Georgraphy»

OpenCog AtomSpace, what's inside

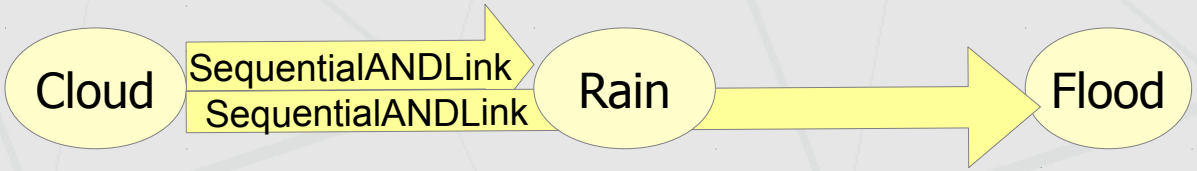


https://github.com/opencog/atomspace/blob/master/opencog/atoms/base/atom_types.script

OpenCog Atoms - example

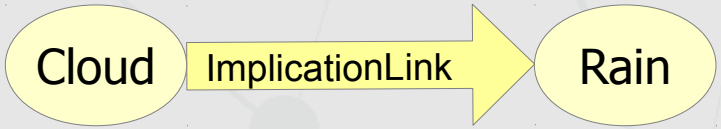
Id	Type	Space	Name (Label)	Level in Meta-Graph	Arity	Argu-ments	Truth Value: Count, Strength, Confidence
11	SequentialANDLink	5	-	2	2	13, 14	22 0.5 (22/44=50%) 1.0 («surely»)
12	SequentialANDLink	5	-	2	3	13, 14, 15	11 0.25 (11/44=25%) 0.5 («probably»)
13	ConceptNode	5	«Cloud»	0	0	-	44 1.0 1.0 («fact»)
14	ConceptNode	5	«Rain»	0	0	-	22 1.0 1.0 («fact»)
15	ConceptNode	5	«Flood»	0	0	-	11 1.0 0.5 («just news»)

Atom Type is used to infer if atom is link and if it is Ordered/directed or Unordered/undirected

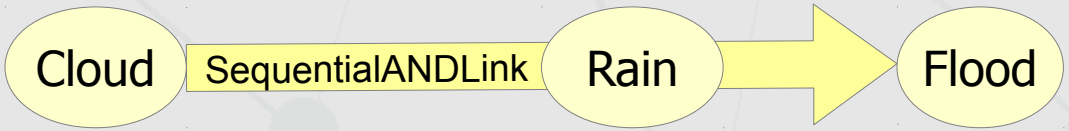


https://github.com/opencog/atomspace/blob/master/opencog/atoms/base/atom_types.script

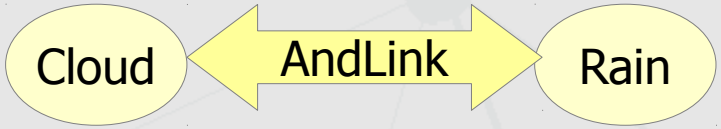
Ordered (directed) and Unordered (undirected) Links



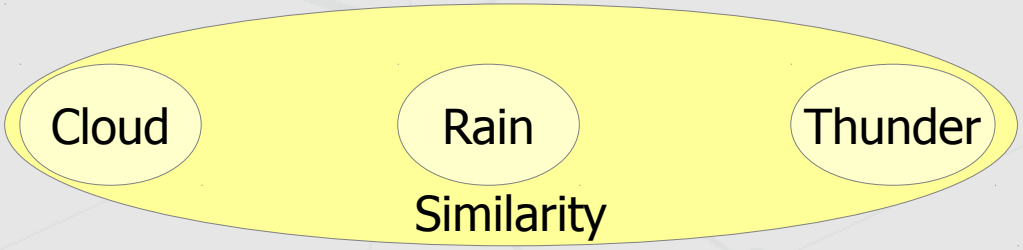
ImplicationLink
ConceptNode ”Cloud”
ConceptNode ”Rain”



SequentialANDLink
ConceptNode ”Cloud”
ConceptNode ”Rain”
ConceptNode ”Flood”



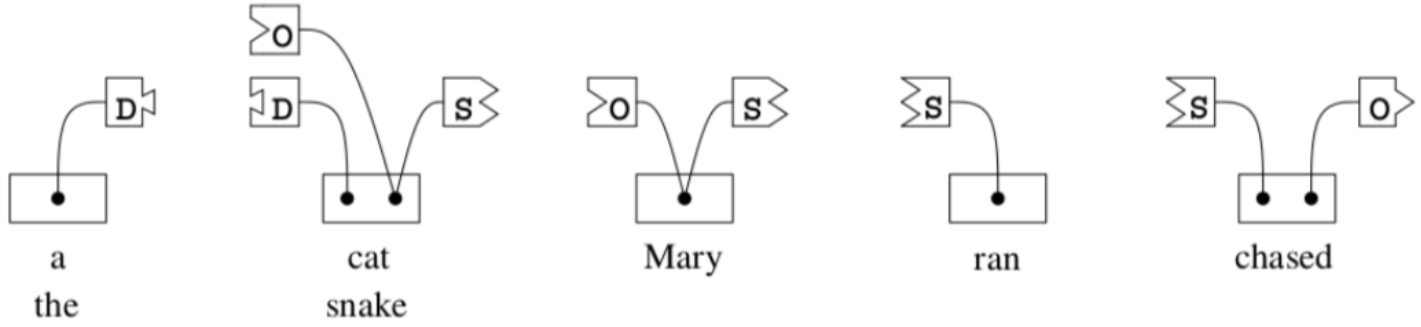
AndLink
ConceptNode ”Cloud”
ConceptNode ”Rain”



AndLink
ConceptNode ”Cloud”
ConceptNode ”Rain”
ConceptNode ”Thunder”

expressed in Atomese language

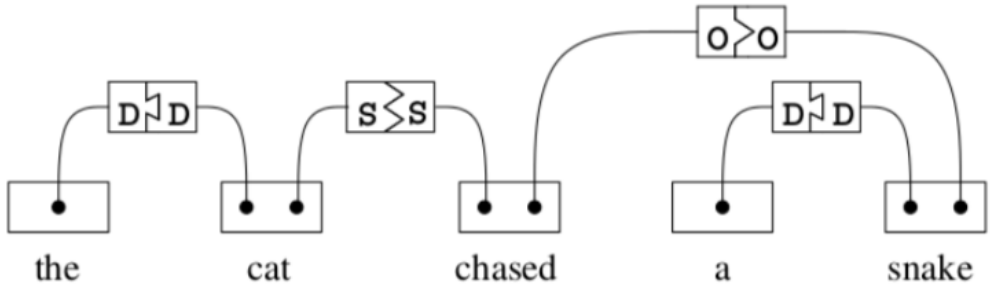
OpenCog Natural Language Graphs (Link Grammar – Disjuncts & Connectors)



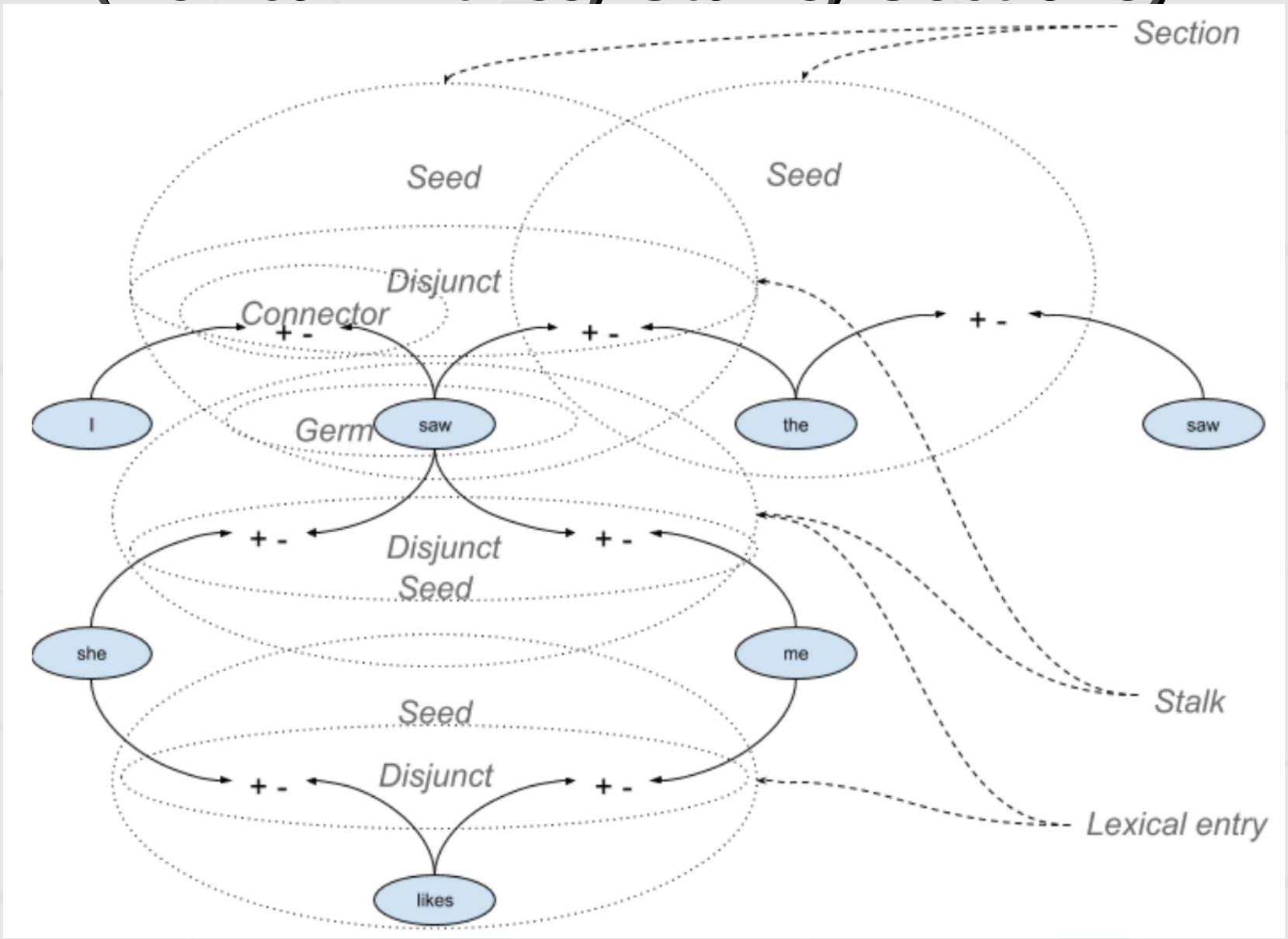
An illustration of Link Grammar connectors and disjuncts. The connectors are the jigsaw-puzzle-shaped pieces; connectors are allowed to connect only when the tabs fit together. A disjunct is the entire (ordered) set of connectors for a word. As lexical entries appearing in a dictionary, the above would be written as

```
a the: D+;
cat snake: D- & (S+ or O-);
Mary: O- or S+;
ran: S-;
chased S- & O+;
```

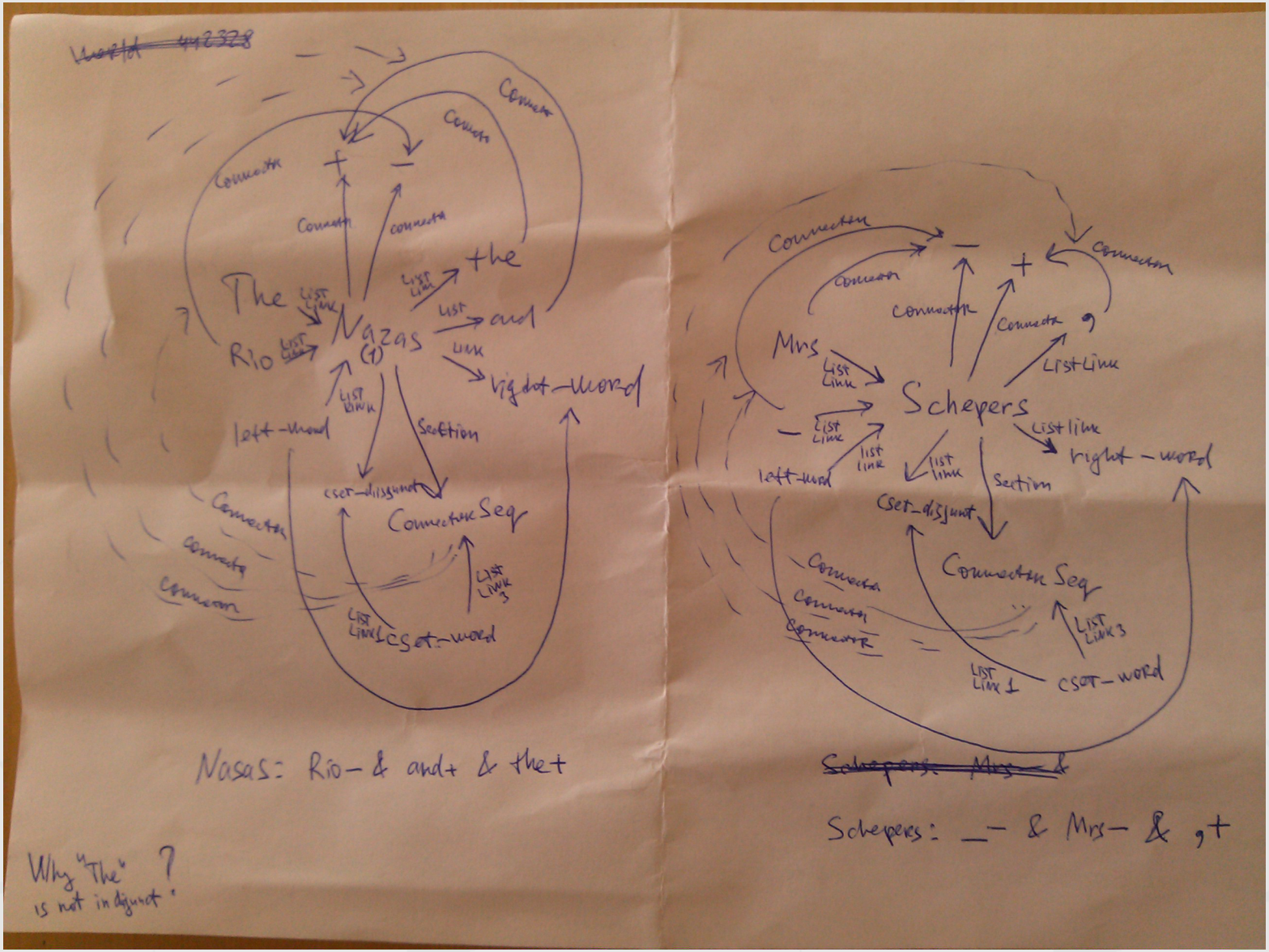
Note that although the symbols ‘‘&’’ and ‘‘or’’ are used to write down disjuncts, these are ***not*** Boolean operators, and do ***not*** form a Boolean algebra. They do form a non-symmetric compact closed monoidal algebra. The diagram below illustrates puzzle pieces, assembled to form a parse:



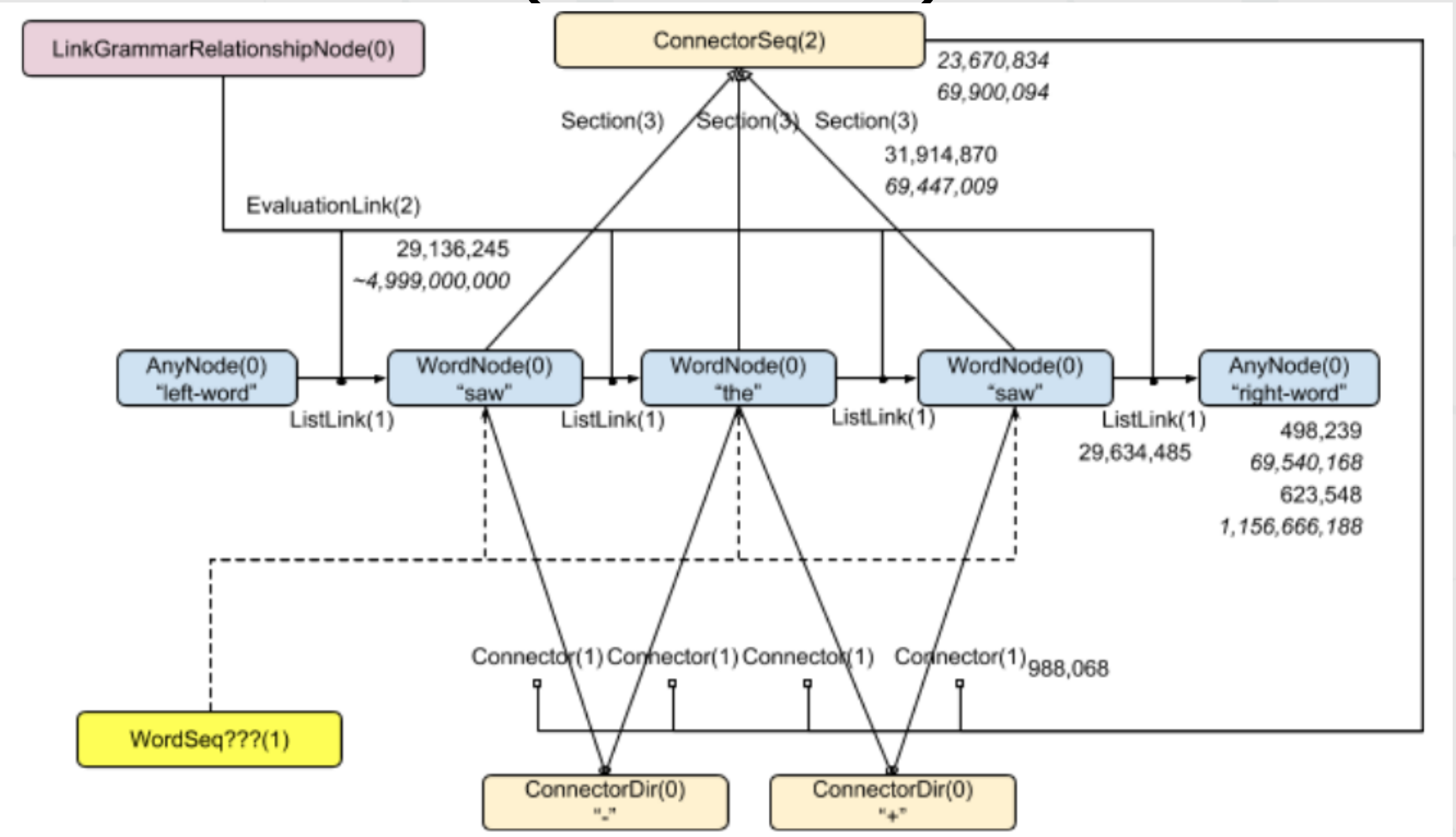
OpenCog Natural Language Graphs (Lexical Entries, Stalks, Sections)



OpenCog Natural Language Graphs (under the hood)



OpenCog Natural Language Graphs (real Atoms)



Legend:

AtomType (Atom Order)

"Atom Label"

Number of Atoms — unique Atoms

Atom Counts — how much times the Atom is observed

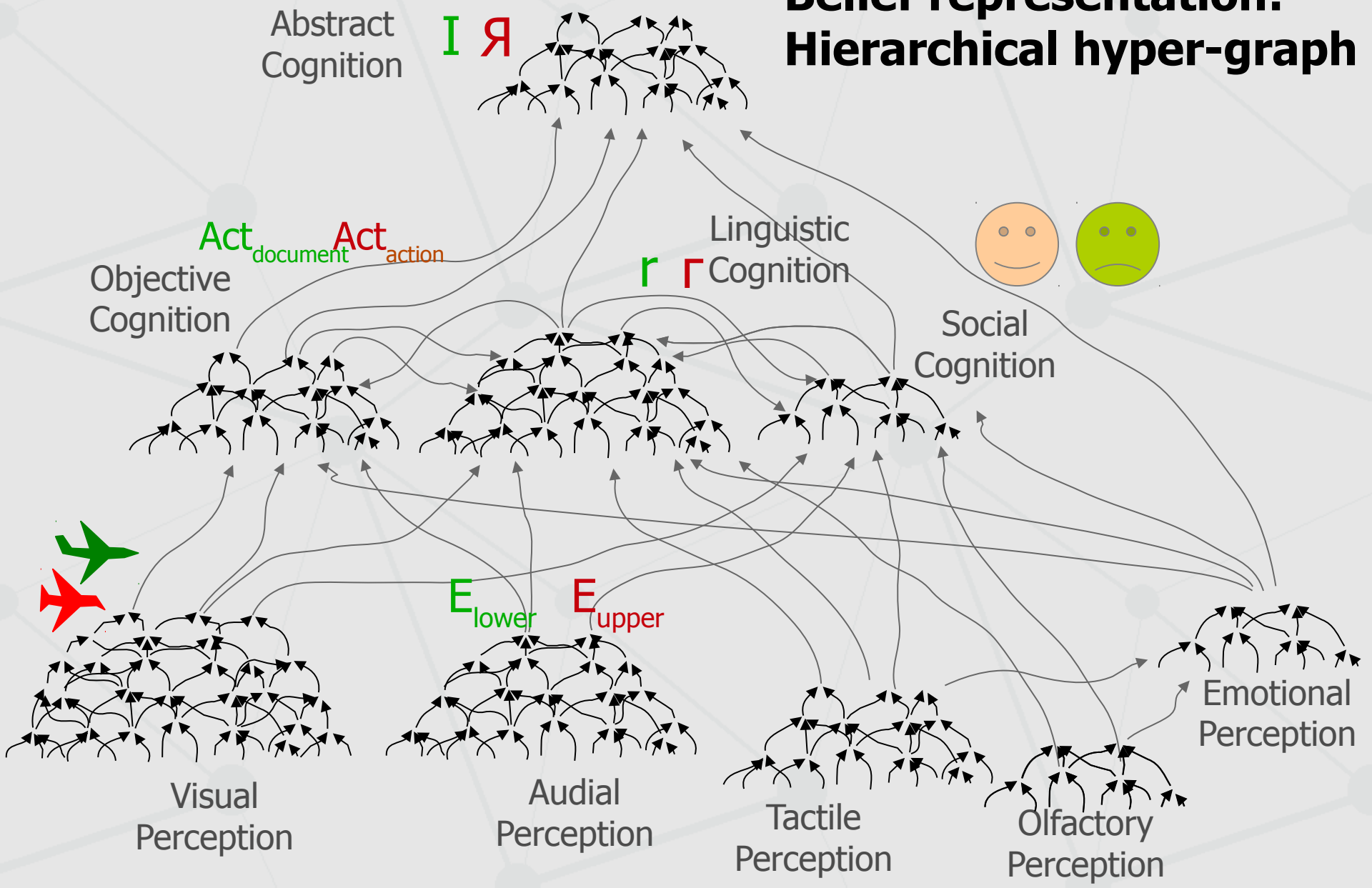
The Emerging World Wide Mind

Anton Kolonin
Webstructor project
2013, August

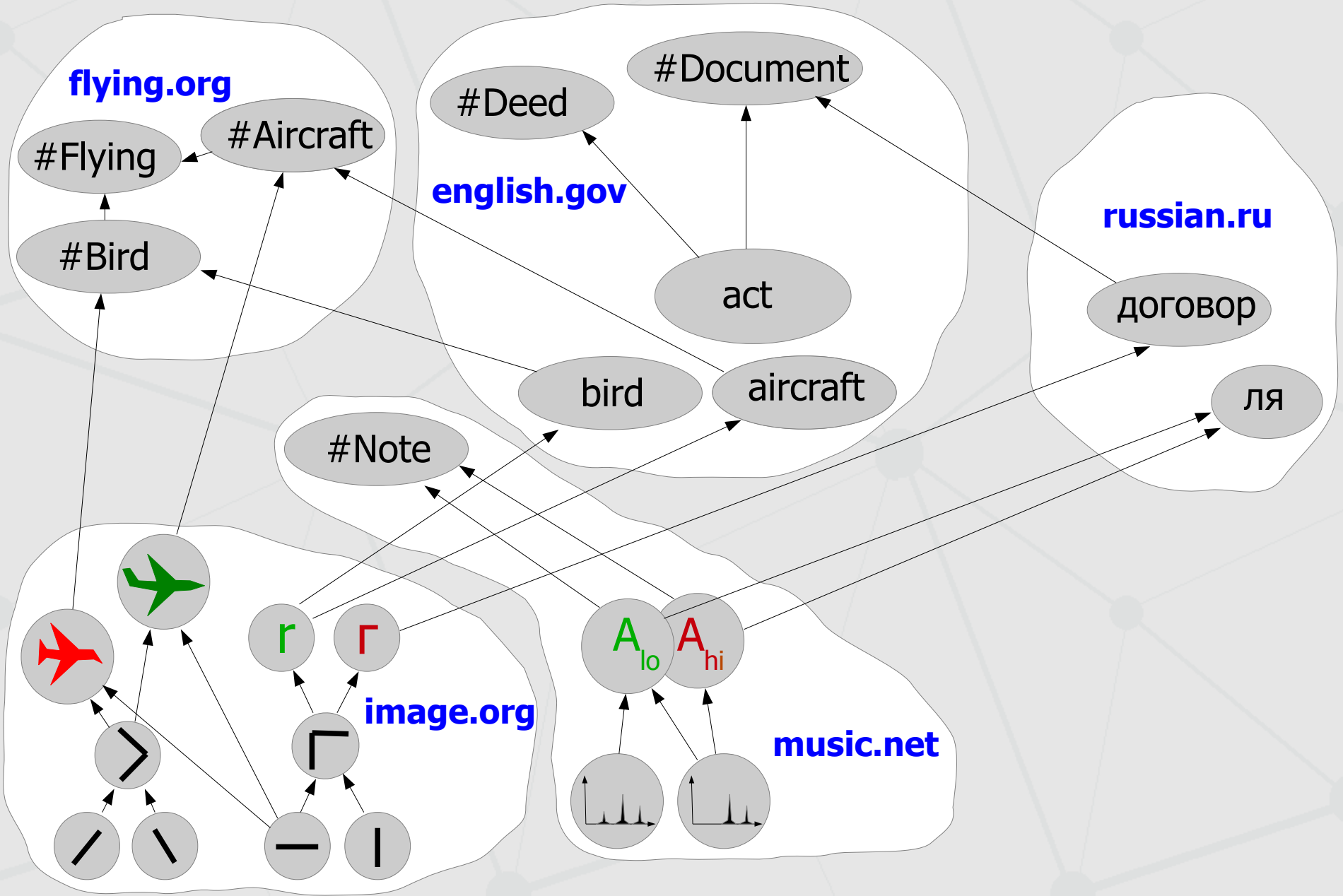
(Slides from earlier presentation on the matter)

Hierarchy of recognition algorithms

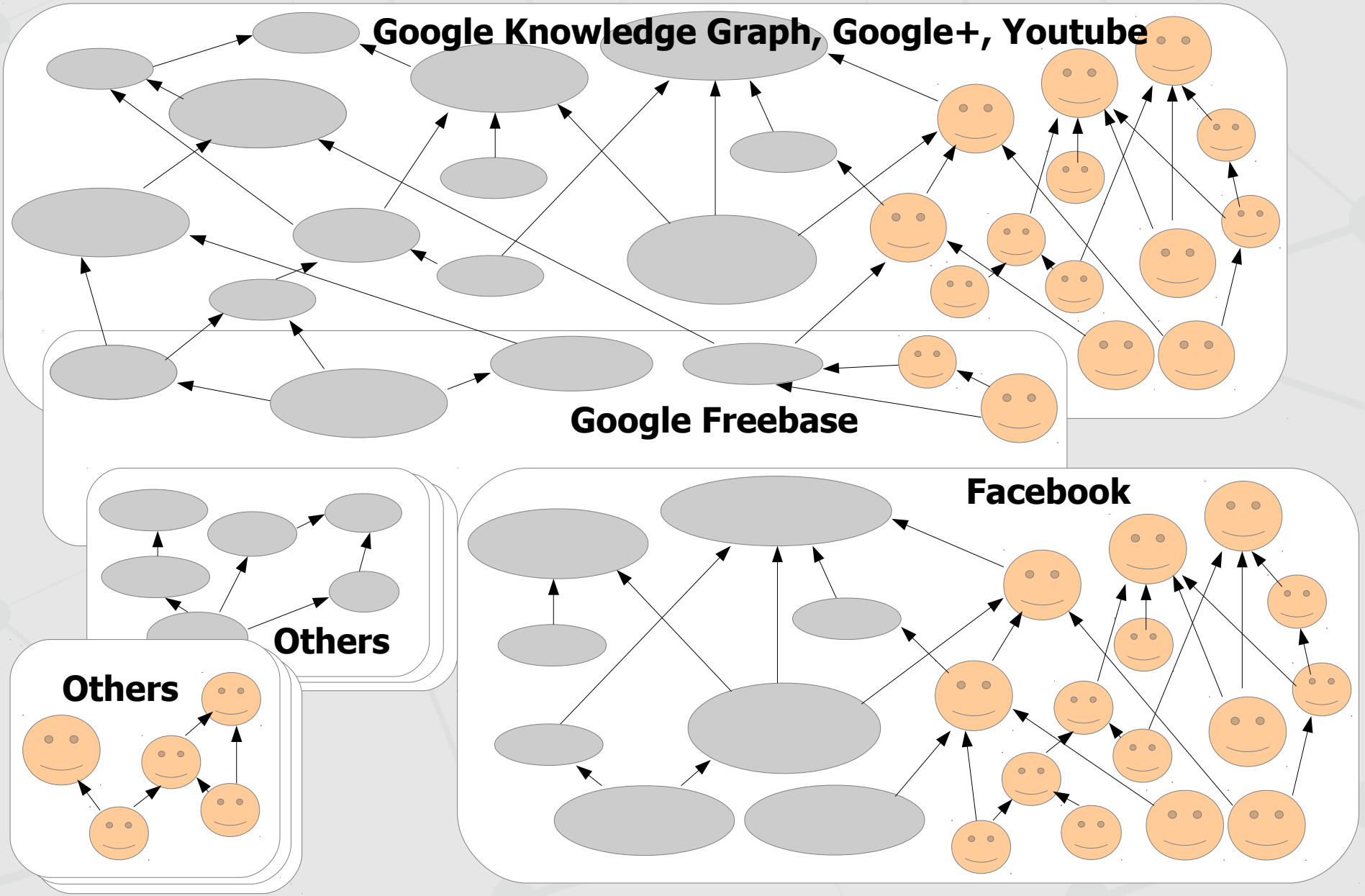
Belief representation:
Hierarchical hyper-graph



Semantic nets and web – as planned



Semantic nets and web - today



Ontologies - Status by 2018 *(Incomplete, thus far)*

- http://schema.org/ - small and compact de-facto standard schema of semantic relations (link types) for describing information published on the web pages in graph form

- https://developers.google.com/freebase/ - formerly open-source «ontology of everything» purchased by Google, exposed for community for crowdsourcing and made private later incorporating it in Google Knowledge Graph, including schema and upper ontology and semanctics factual database

- https://en.wikipedia.org/wiki/Upper_ontology - Upper Ontologies are different ontologies describing basic fundamental concepts and their relationships (schemas) under different licensing terms

- https://www.wikidata.org/ - public open source repository of common world knowledge, places, facts and evens in format of semantic graph

Distributed knowledge engineering Webstructor system

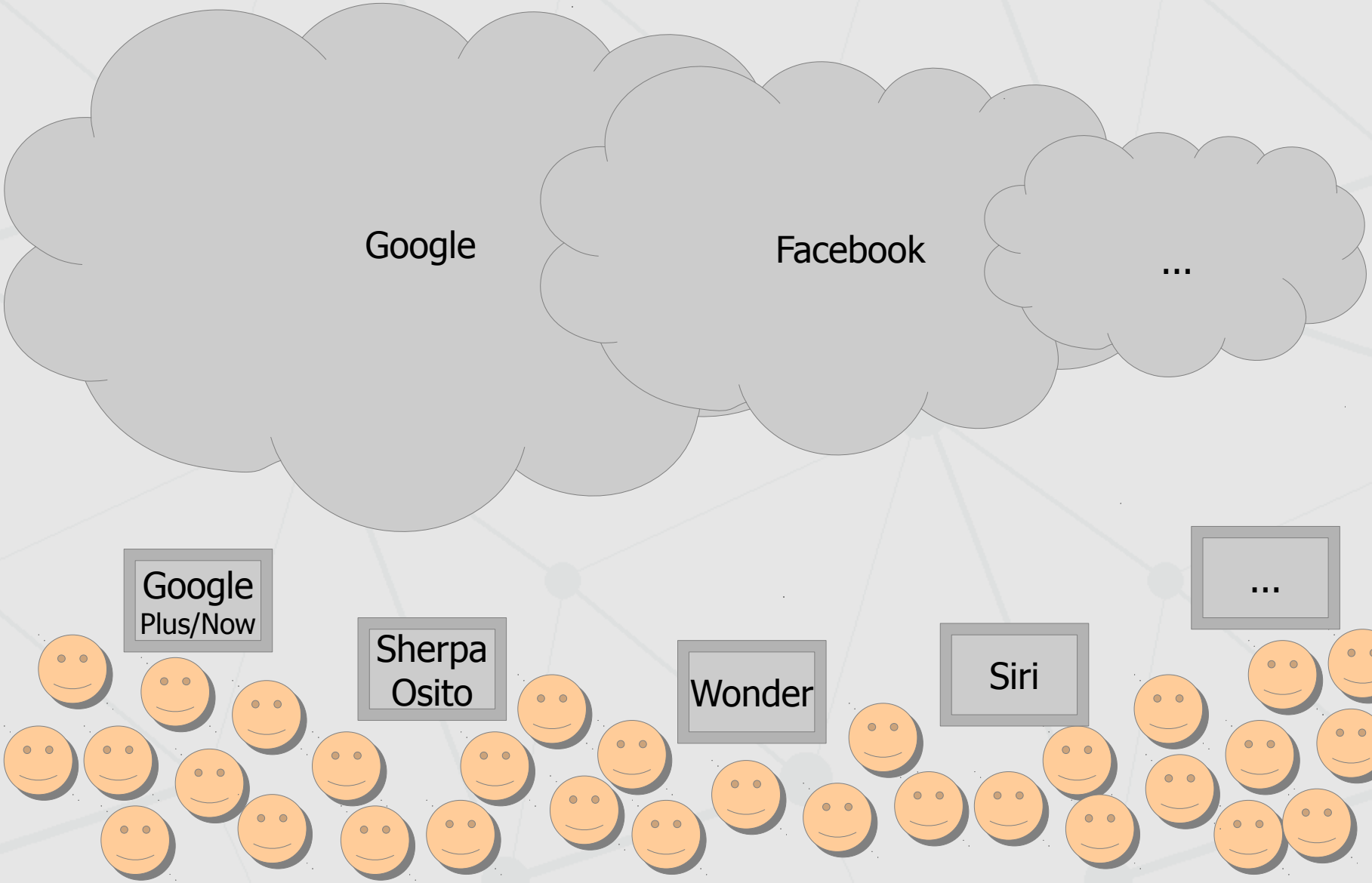
http://www.webstructor.net/

Anton Kolonin, 2013

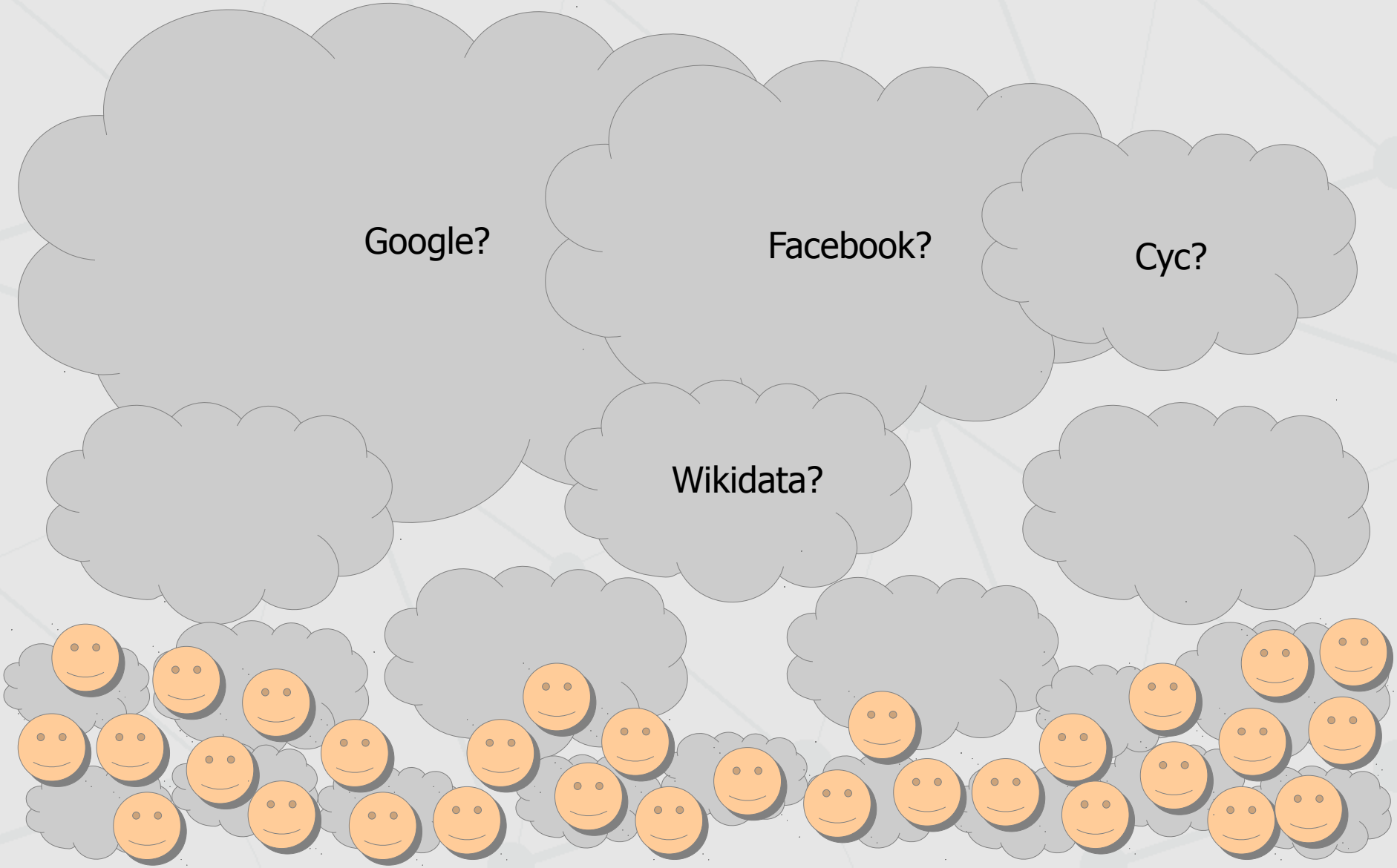
- Decentralized globalization model
  - Requirements
  - Agent specialization
  - Topologies and functional clusters
  - Social evidence-based knowledge model
- Knowledge representation in graphs
- Webstructor system
  - Architecture and supported topologies
  - Hyper-graphs and subgraphs
  - Present applications
  - Object-Relational Language (ORL)
  - Project history

Distributed knowledge engineering

Centralized model



Distributed knowledge engineering
Decentralized model



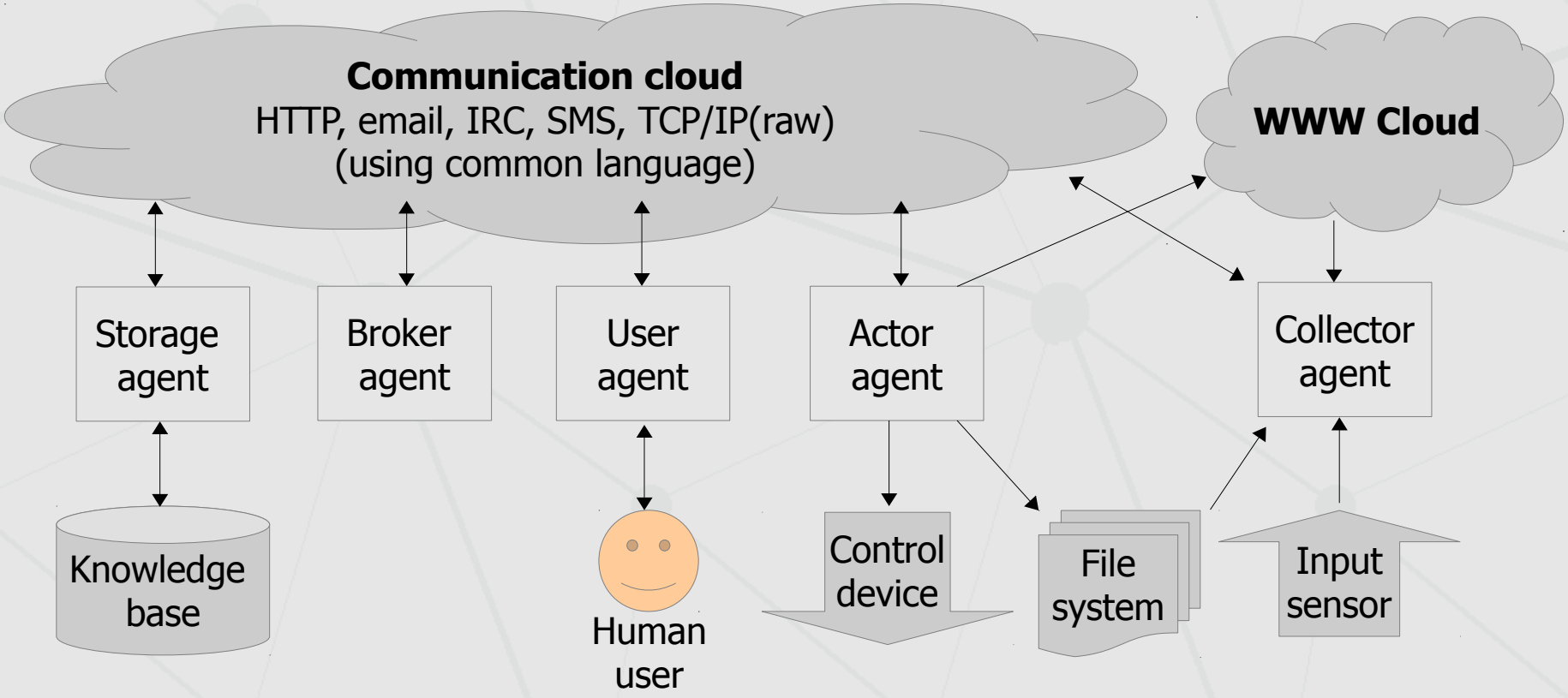
Distributed knowledge engineering
Decentralized model
Requirements

- Network of computer **agents** as functional structure and rules of dynamic self-organization:
  - rich **historical memory** shared by communicating computer agents (e.g. accessible public banks of information);
  - rich **sensory environment** driving the communication and accessible **means of gathering novel information** (e.g. search, browsing and messaging against peer computer agents);
  - for an agent, **ability to explicitly expose its own knowledge** indicating confidence, proprietary rights and privacy levels of it;
  - unrestricted **fertility of diverse behavioral patterns** (i.e. computational algorithms) exposed by agents (capable for evolution upon feedback);
  - ease of peer-to-peer communication by means of **unified language** based on the same upper ontology (i.e. open knowledge transfer/manipulation protocol);
  - legal definition of the **responsibility for computer agent's actions** (e.g. search results, browse requests and messages) delegated to the person or corporation operating the agent hardware.

Distributed knowledge engineering

Decentralized model

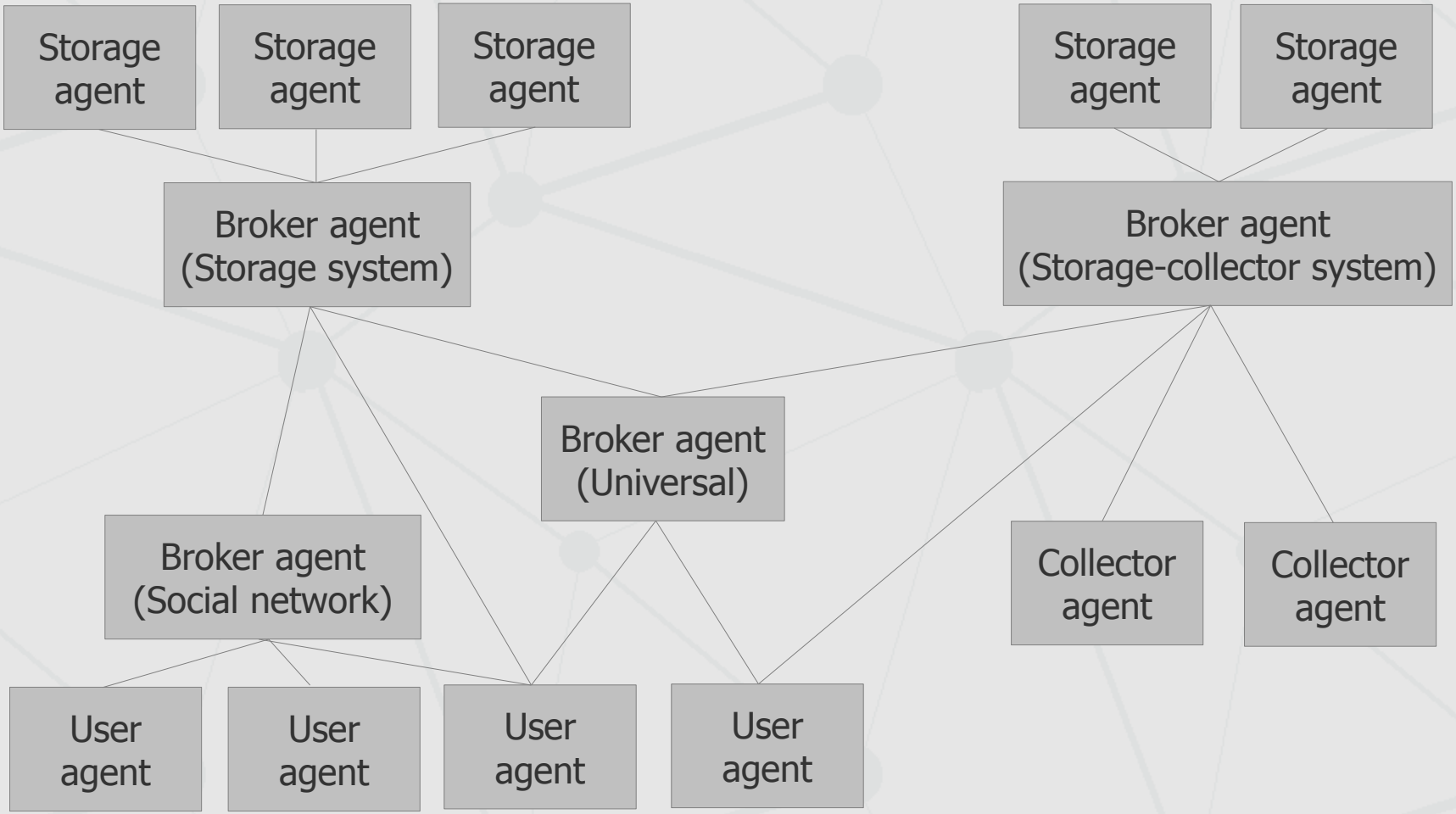
Agent specialization



Distributed knowledge engineering

Decentralized model

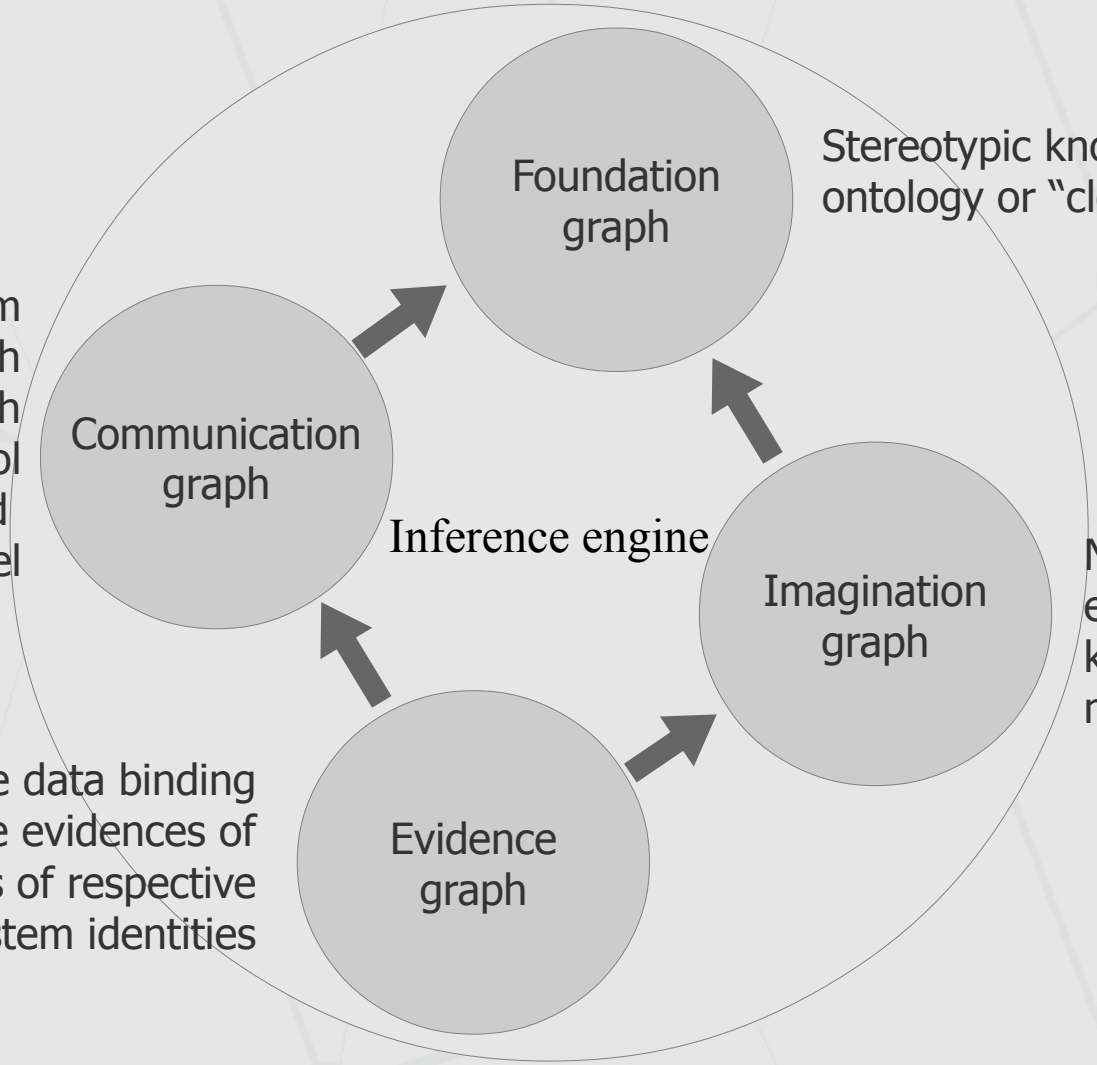
Topologies and functional cluster



Distributed knowledge engineering

Decentralized model

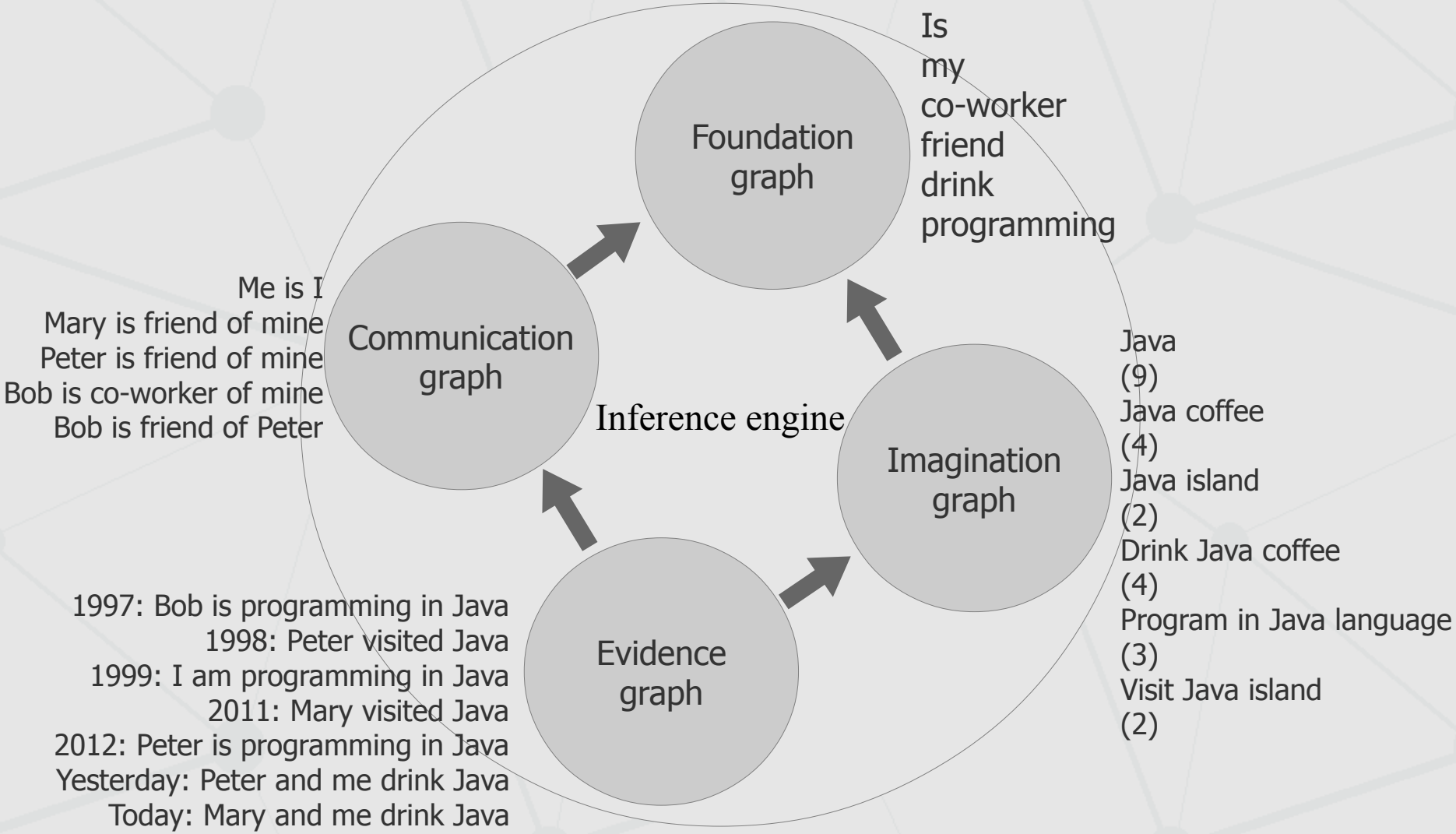
Social evidence-based data model



Distributed knowledge engineering

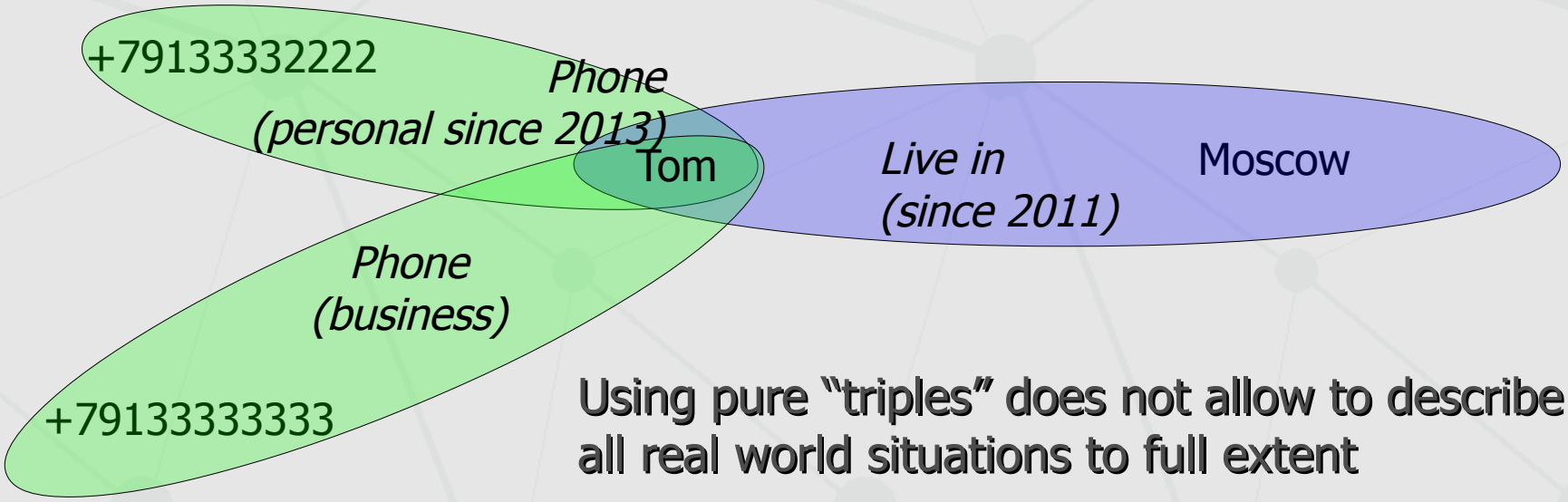
Decentralized model

Social evidence-based data model



Distributed knowledge engineering
Knowledge representation in graphs

Tom lives in Moscow since 2011, his phones:
+7913333222 (personal since 2013), +79133333333 (business)

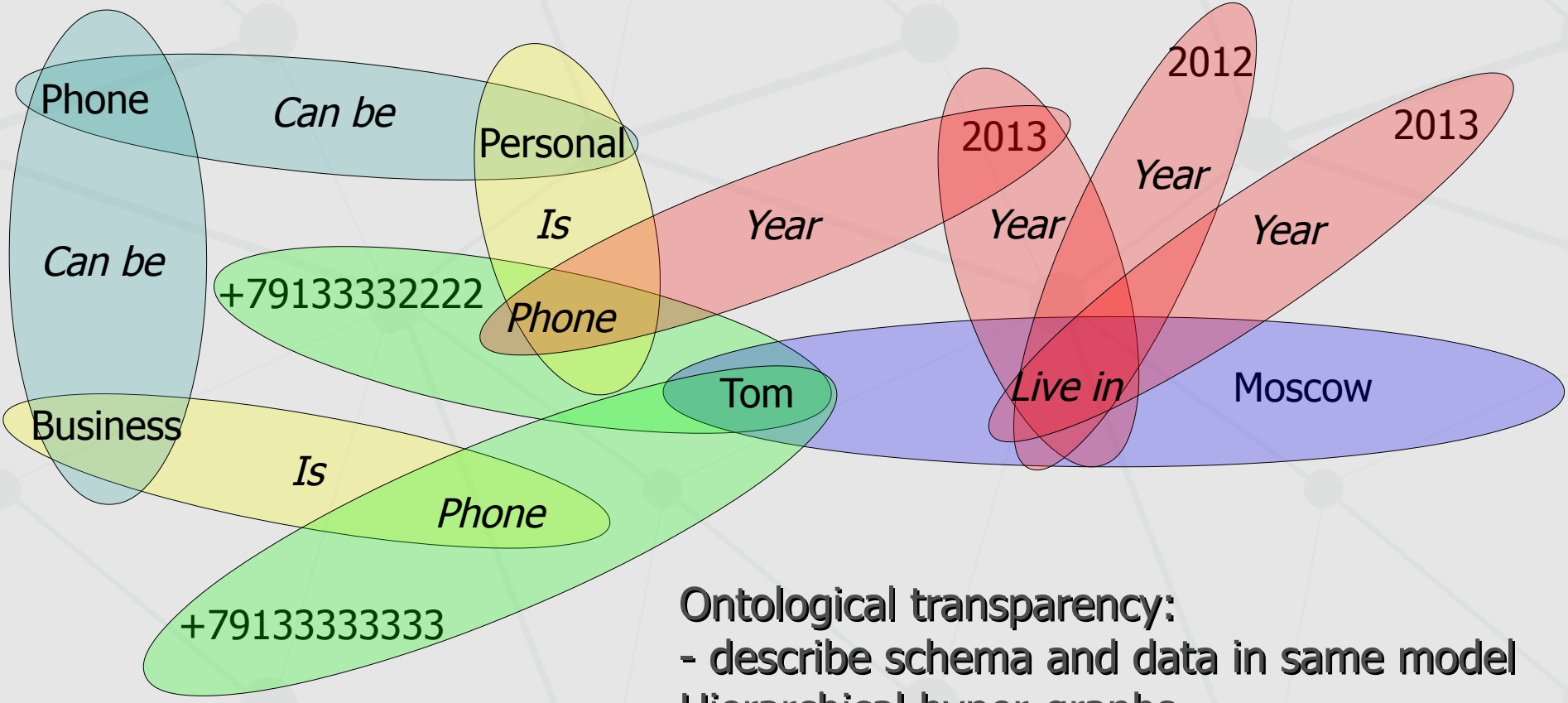


Using pure "triples" does not allow to describe all real world situations to full extent

Distributed knowledge engineering

Knowledge representation in graphs

Tom lives in Moscow since 2011, his phones:
+7913333222 (personal since 2013), +79133333333 (business)

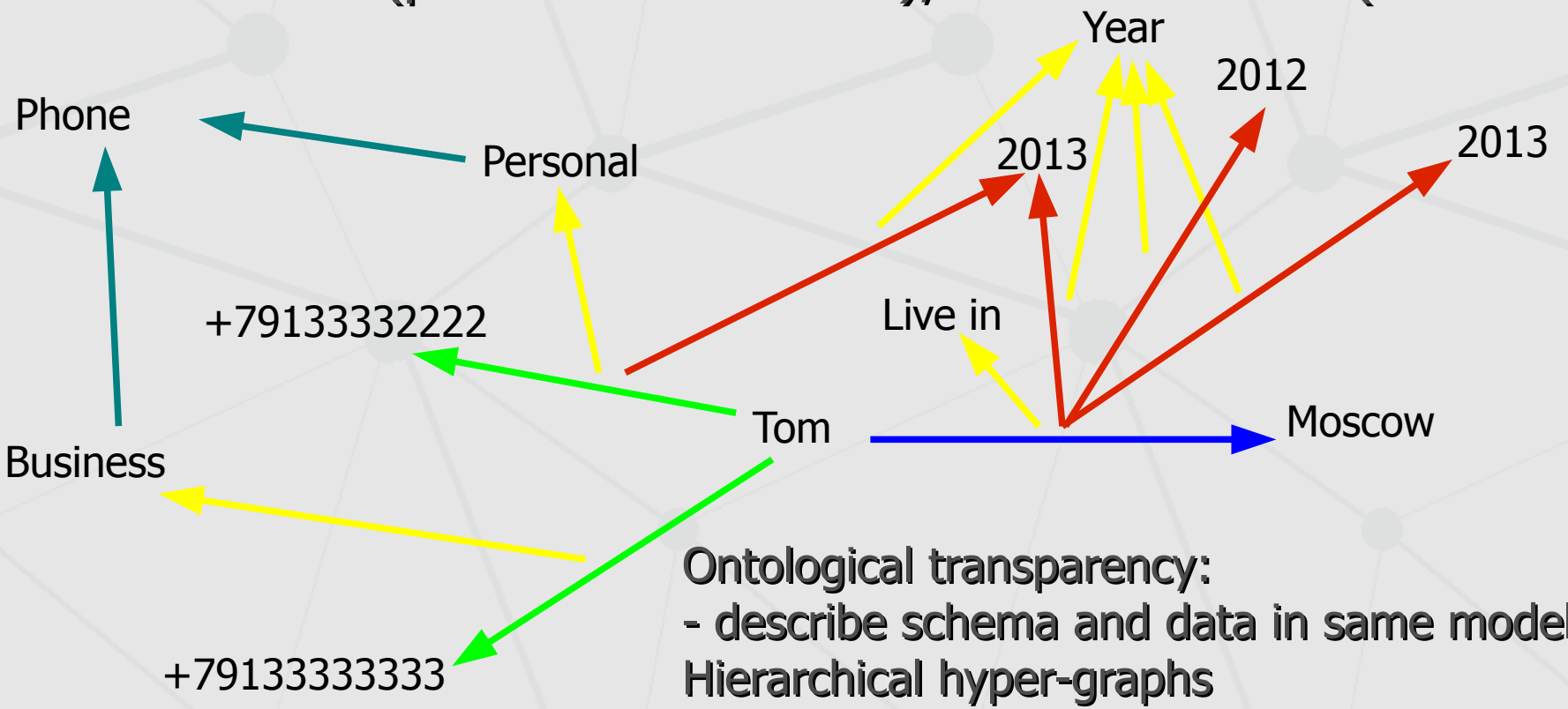


Ontological transparency:
- describe schema and data in same model
Hierarchical hyper-graphs
- involving links in other links

Distributed knowledge engineering

Knowledge representation in graphs

Tom lives in Moscow since 2011, his phones:
+7913333222 (personal since 2013), +79133333333 (business)



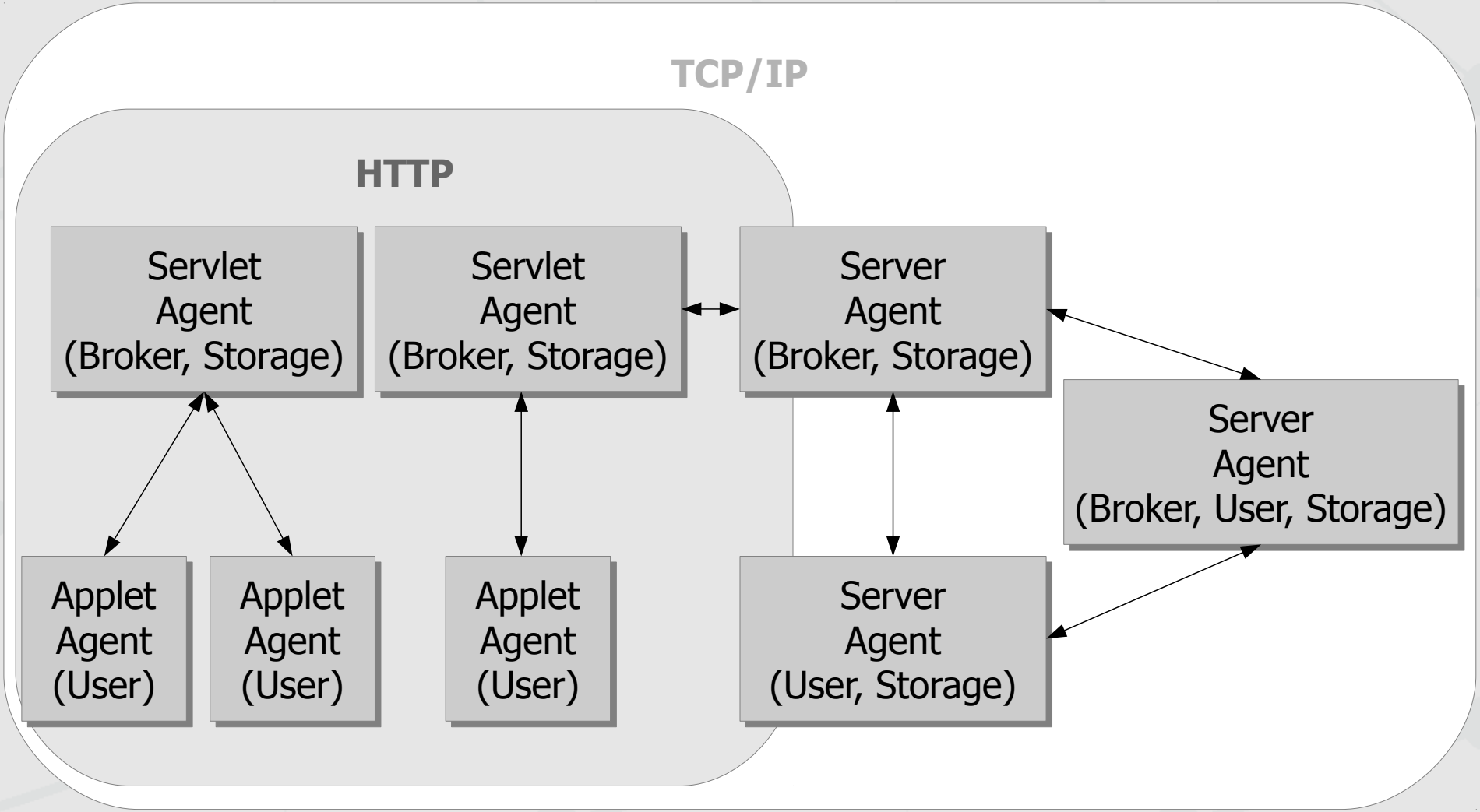
Ontological transparency:
- describe schema and data in same model

Hierarchical hyper-graphs
- involving links in other links

Ultimate normalization
- down to binary relations (theoretically)

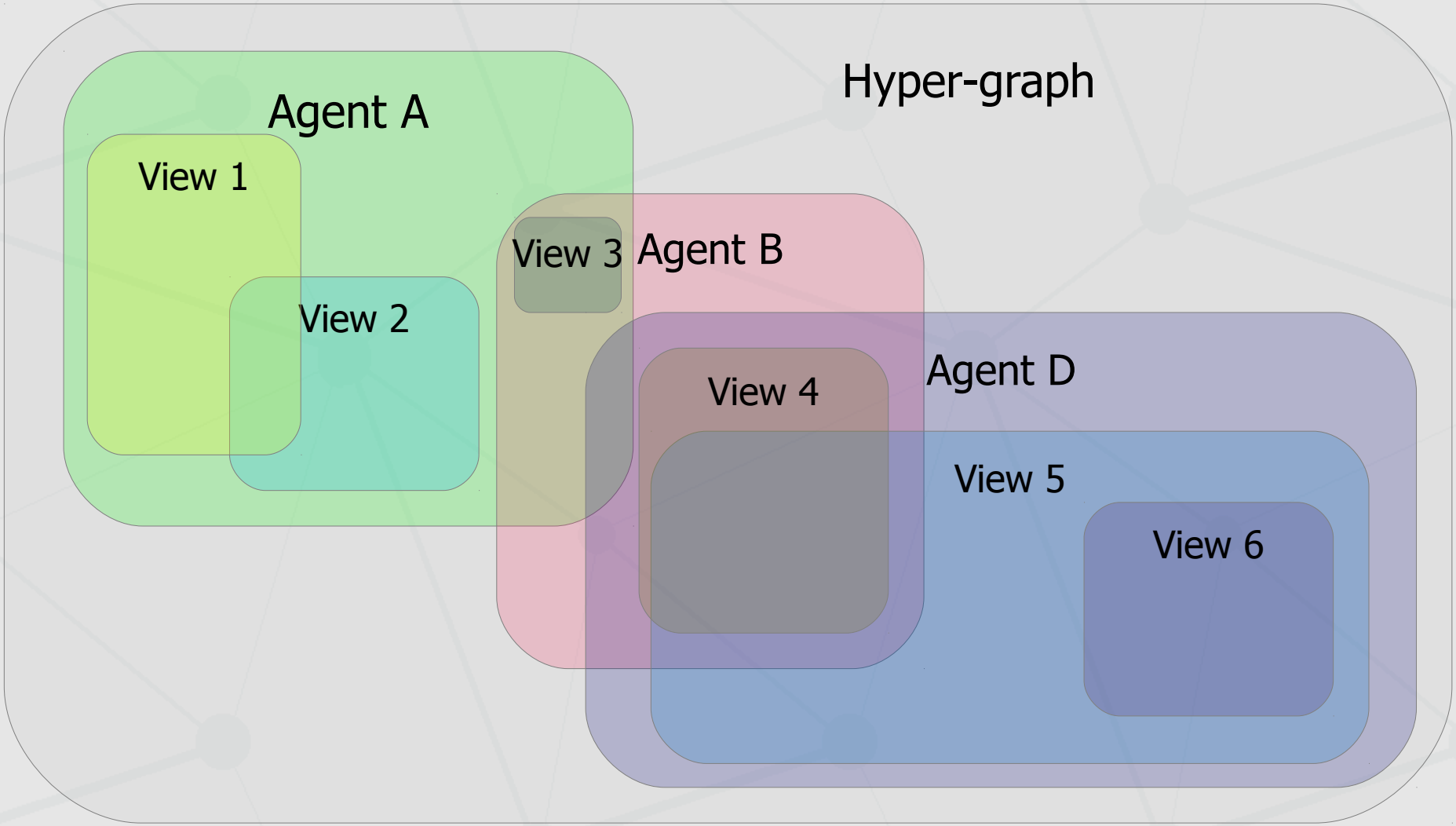
Distributed knowledge engineering Webstructor system

Architecture and possible topologies



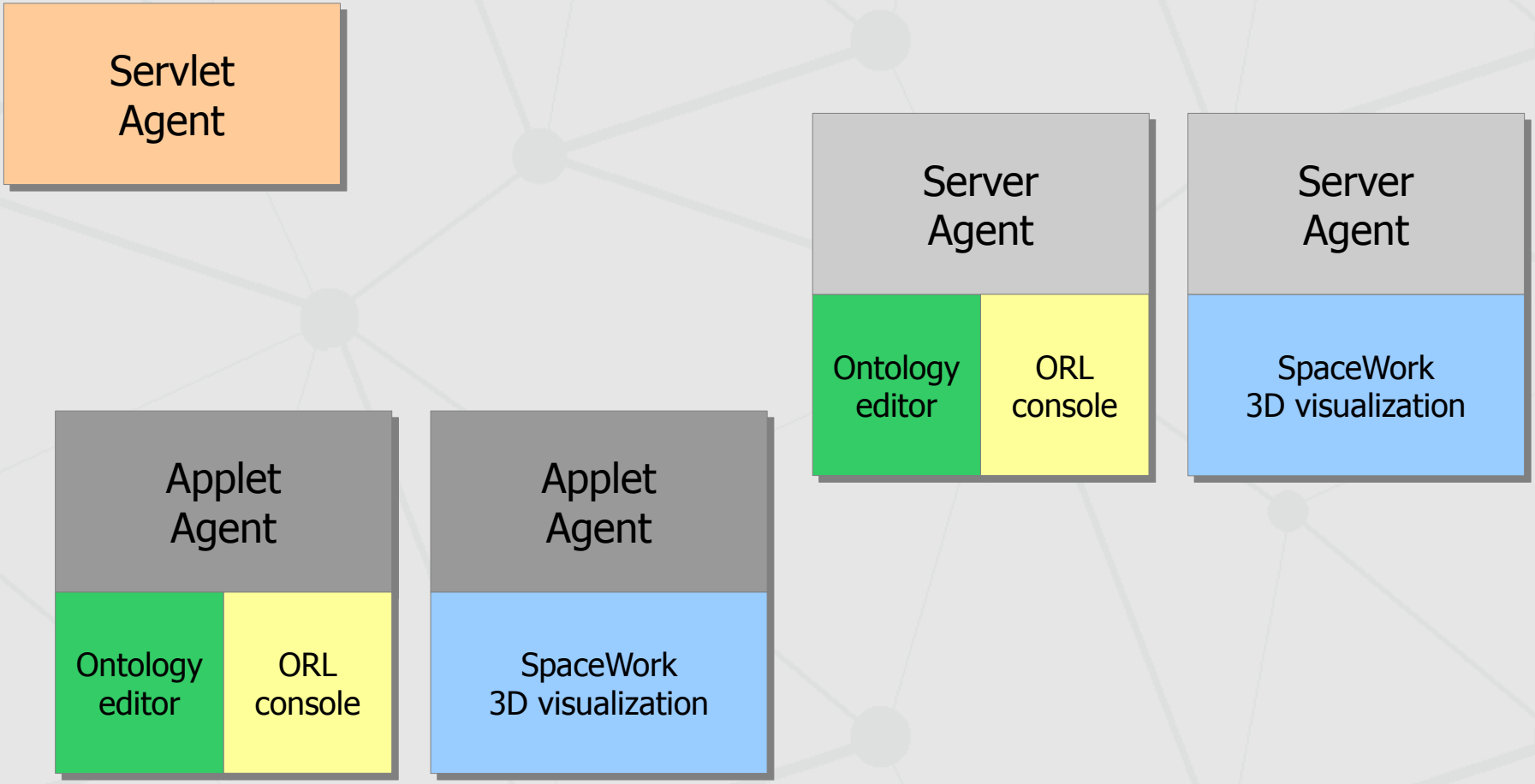
Distributed knowledge engineering Webstructor system

Global hyper-graph and subgraphs of agents



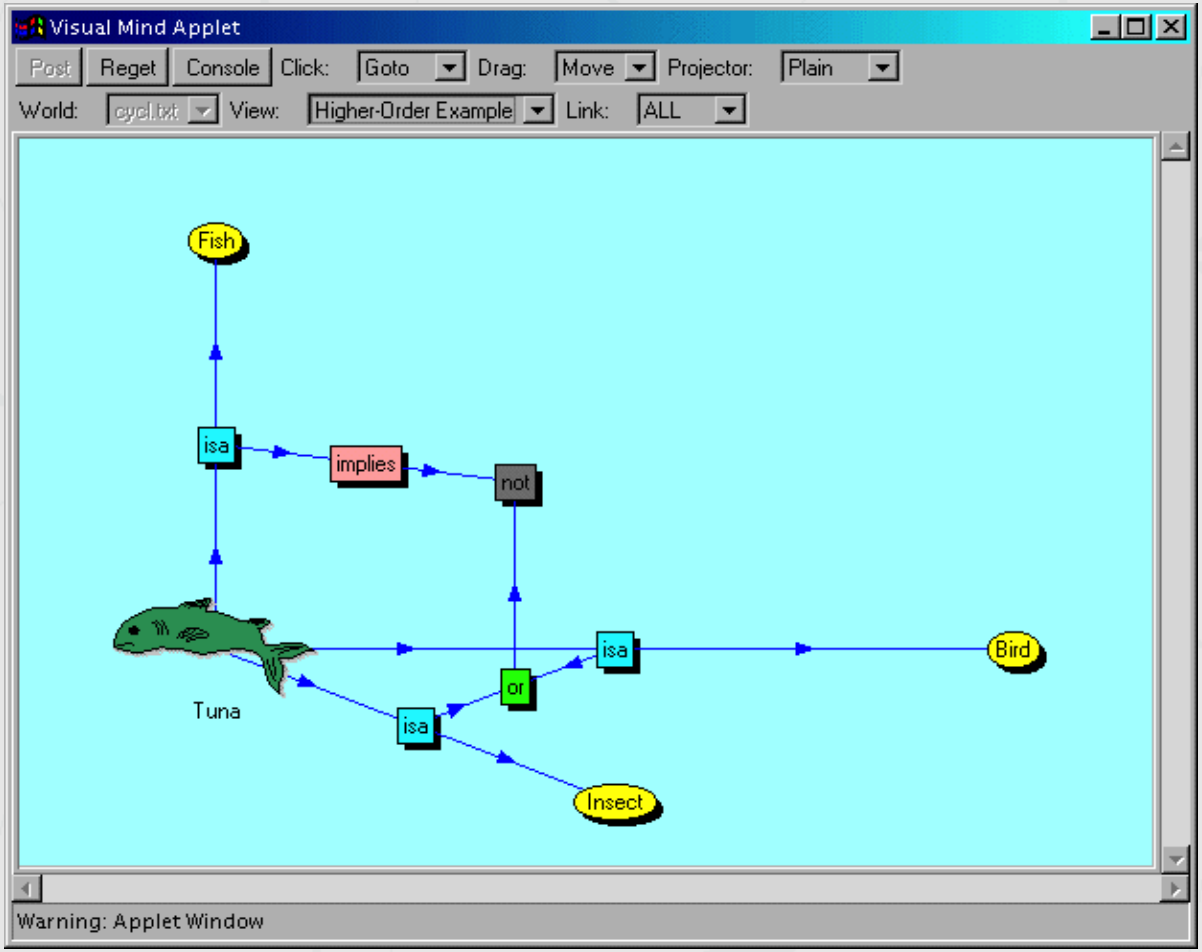
Distributed knowledge engineering Webstructor system

Existing applications



Distributed knowledge engineering Webstructor system

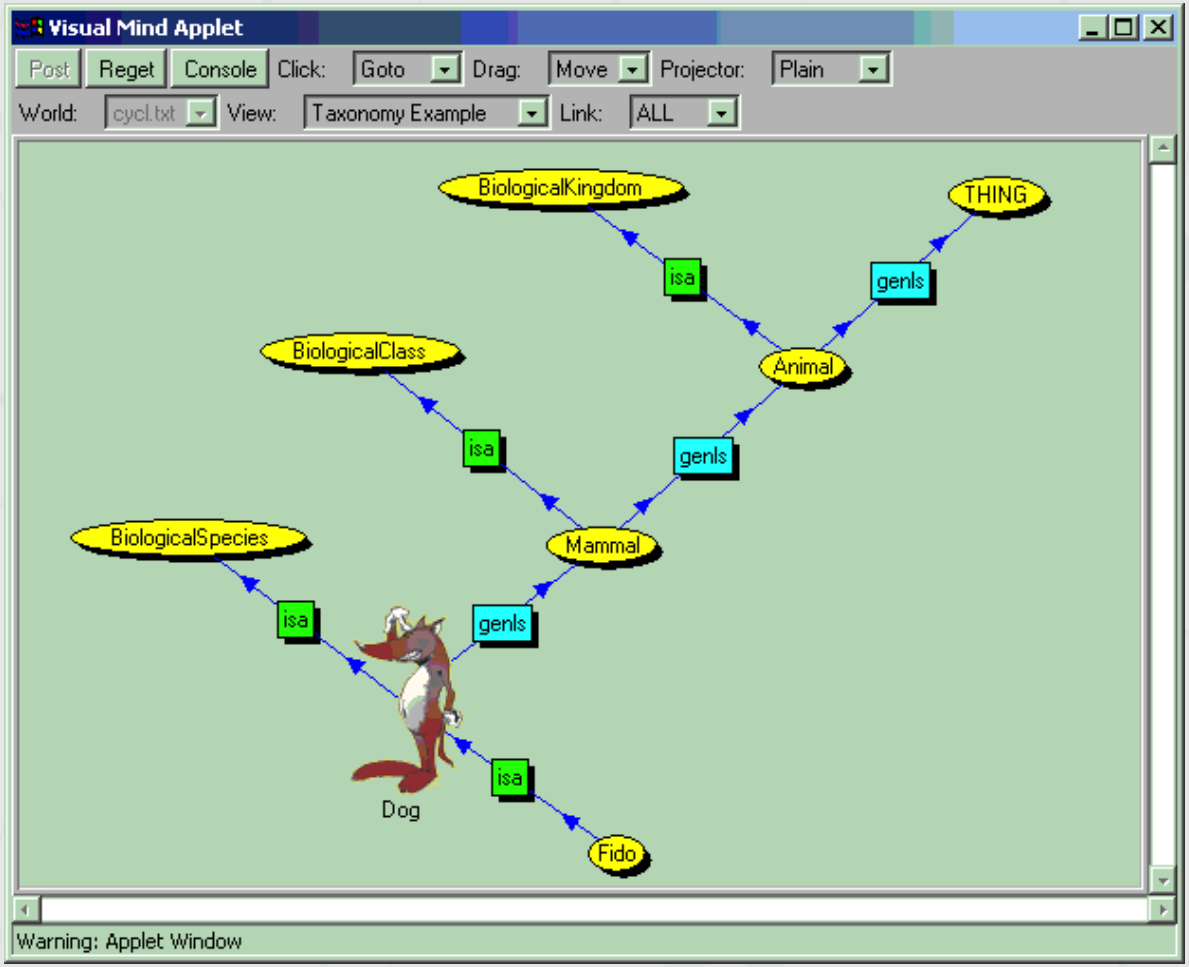
Visual ontology and logical formulae editor



«If tuna is a fish, that implies it is not an insect or a bird.»

Distributed knowledge engineering Webstructor system

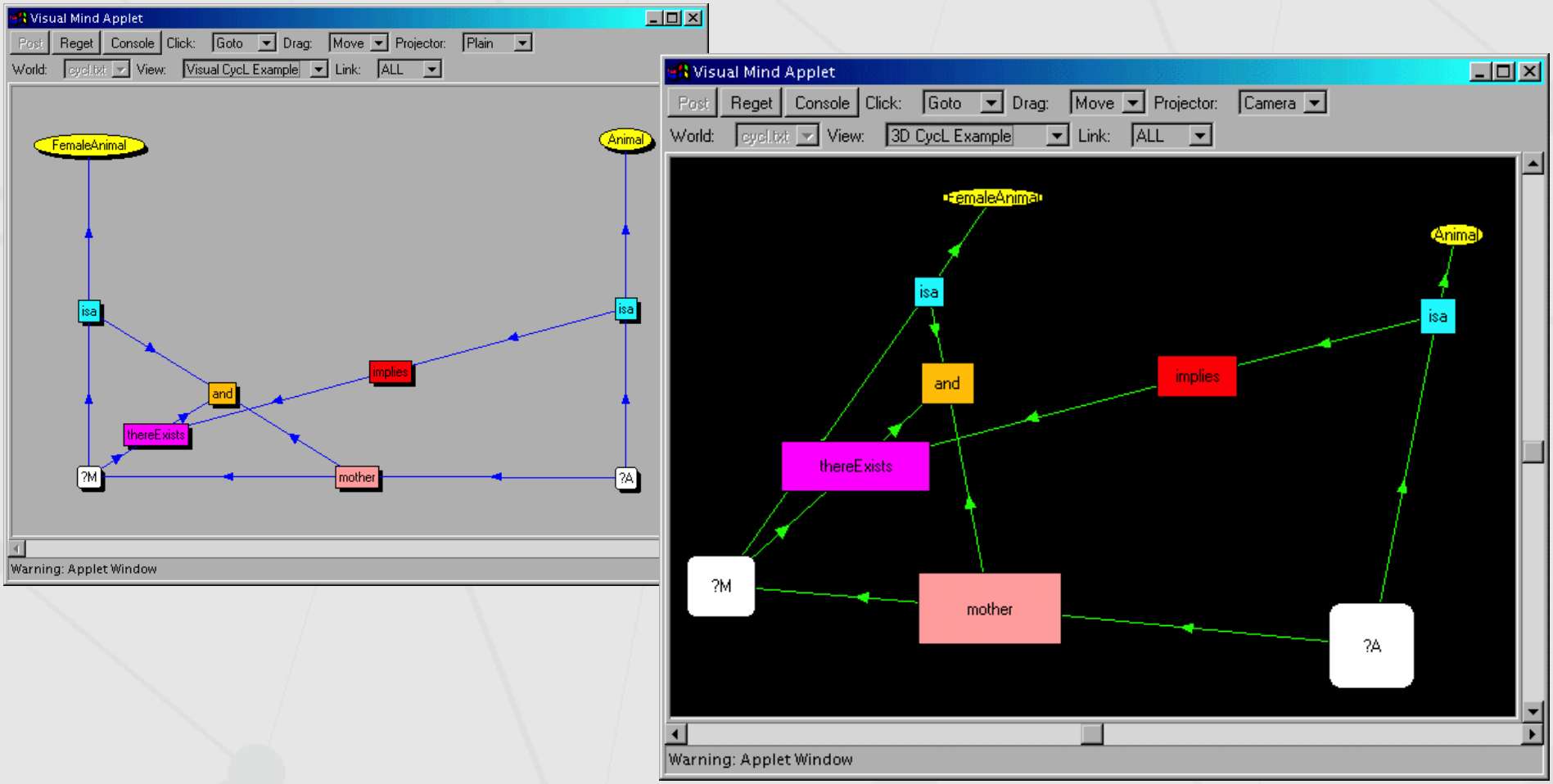
Representing Cyc «micro-theory»



Fragment of «biological kingdom» of Cyc «upper ontology»

Distributed knowledge engineering Webstructor system

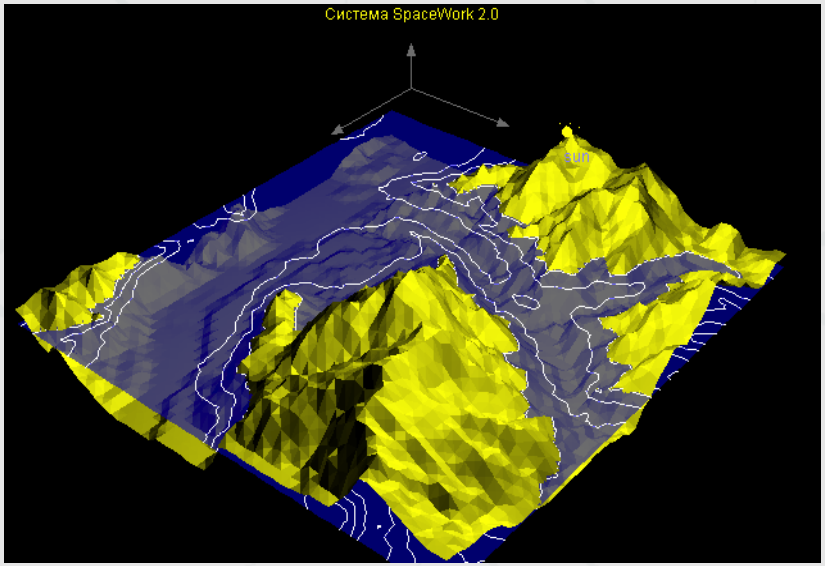
Visual editor of CycL assertions

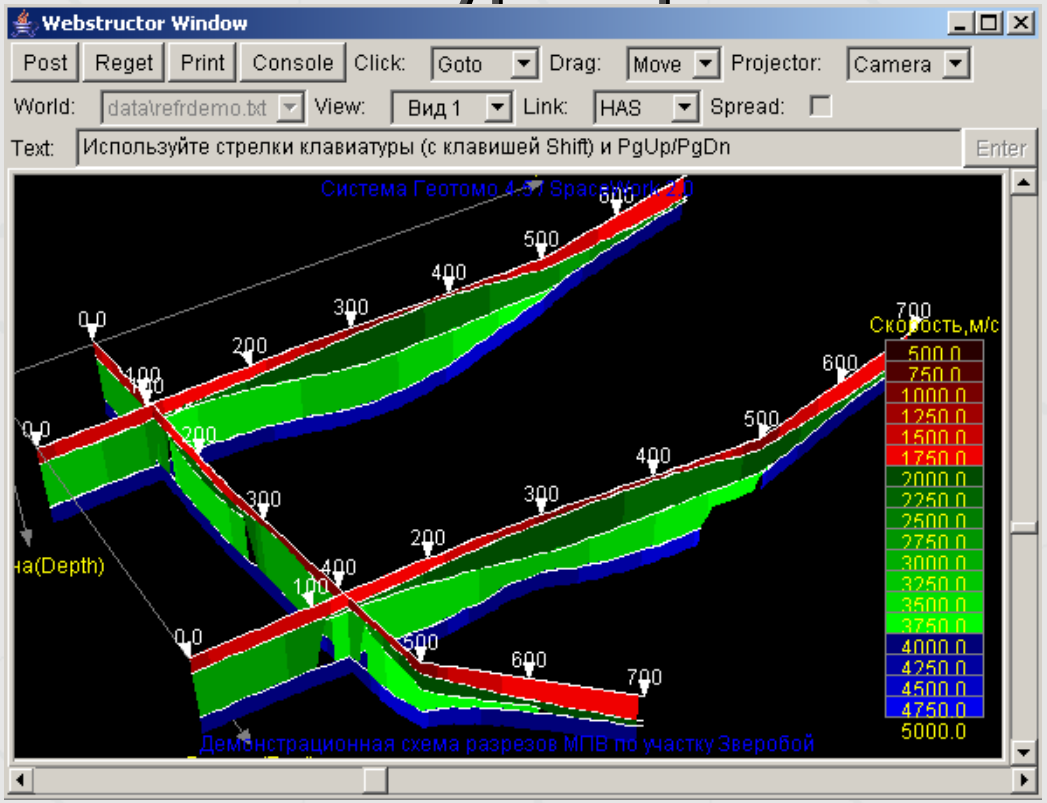


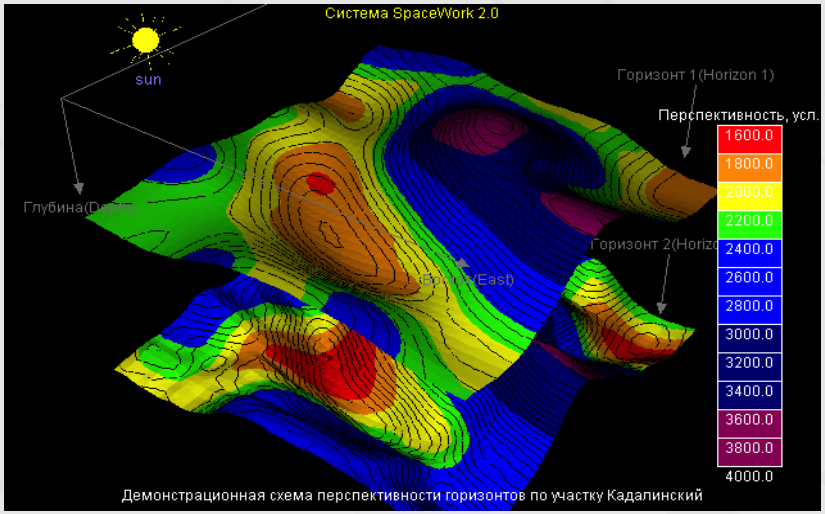
(implies (isa ?A Animal) (thereExists ?M (and (mother ?A ?M) (isa ?M FemaleAnimal))))

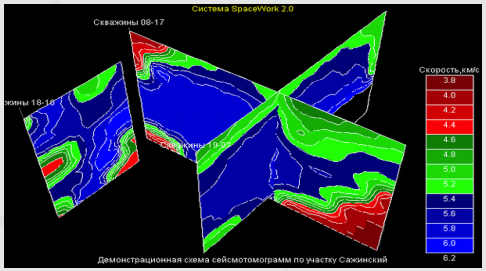
Distributed knowledge engineering Webstructor system

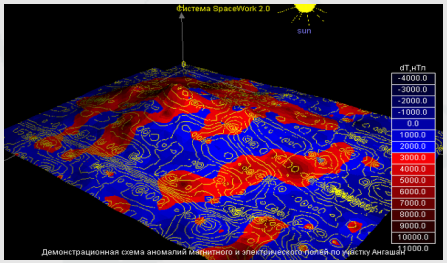
Visualization of multi-dimensional hyper-spaces











Distributed knowledge engineering Webstructor system

Object Relational Language (ORL)

- Syntax assumes few fundamental objects such as **thing**, **property**, **name**, numeric or literal **constant**, **array**, **set** (where a set can be either **mandatory** or **optional**) and **query**.
- Within particular implementation of the language, there can be specific scope of terms describing an application object model as **keywords**.
- Description of any schema (classes, attributes, etc.) is done in the same linguistic space as description of data objects and values – ontological transparency.
- Description of functional schemata (functions, methods and operators) is also possible in the same linguistic space however compact (scripting) notation is also possible.
- Centric feature of language is query (somewhat resembling structured query applicable to relational model), which is used as reference (instead of pointers or identifiers) describing structured data as well as functional schemata.
- Enables flexible expression of any sorts of hyper-graphs.

Distributed knowledge engineering Webstructor system

Object Relational Language (ORL)

English	ORL
Here are the items A, B and C where A has properties X and Y while B and C are in relationship Z.	ITEM A,B,C;; A HAS (X), (Y);; B Z(C);;
In order to reach goal 1 one needs condition 2 and 3 to be held true while 2 can be true only if condition 4 happens.	CONDITION C2,C3,C4;; GOAL G1 REQUIRES (C2),(C3);; CONDITION(C2) REQUIRES (C4);;
Each morning need to perform this and that in order, having such and such done at once next.	PROCESS TIME “8:00”; REPEAT (DAILY); ORDER DO THIS, DO THAT;, FORK DO SUCH, DO SUCH;;;
What is that my stuff you mentioned yesterday or the day before?	STUFF(OWNER (ME), UPDATE (AUTHOR (YOU), {TIME “2013-03-22”, TIME “2013-03-21”}).TELL;
What were the relationships between P and Q last year?	PROPERTY(OWNER (P), THING (Q), TIME “2012”).TELL;
Let me know once they roll out next version of the product.	DO EMAIL TO “me@at.org”;; WHEN PRODUCT(VENDOR (THEY)).VERSION CHANGE;;

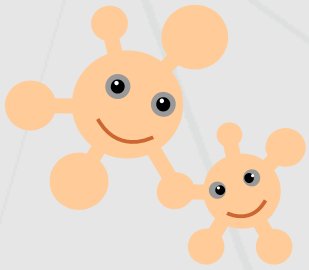
Distributed knowledge engineering Webstructor system

Project History

- 1995-1996 CTC Company
  - Semantic graph employed to fully describe the operational space of a software system to carry out data management, inter-personal interactions, interactive form processing, report generation and action script development.
- 1997-1999 ProPro Group
  - Object relational language (ORL) for inter-agent communication developed to enable development of corporate business automation system for stock exchange domain.
- 2001 Webstructor Project
  - Agent software for peer-to-peer knowledge creation and interchange developed as part of Webstructor project, based on ORL.
- 2006 IT Solutions, Ltd.
  - **Virtual 3D environment** Space Work for the purpose of visualization and sharing of complex scientific data created on basis of Webstructor environment.
- Future plans
  - Implement social evidence-based data model with multi-lingual support, full ORL specification and open-source reference implementation.

http://www.webstructor.net/

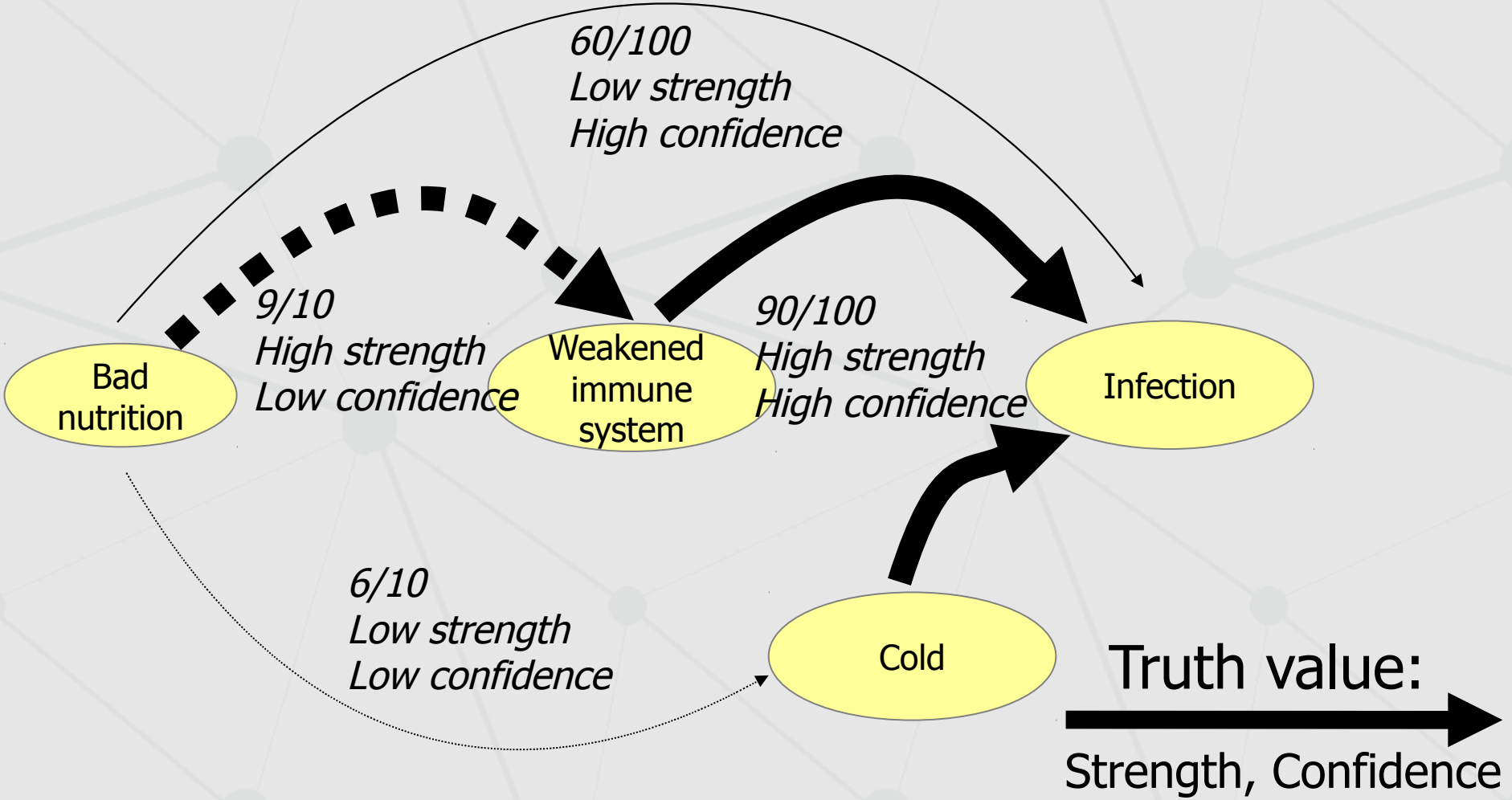
Distributed knowledge engineering and (social) evidence-based knowledge representation in multi-agent systems



Anton Kolonin
Aigents Group
http://aigents.com
KESW-2015 Conference

(Slides from earlier presentation on the matter)

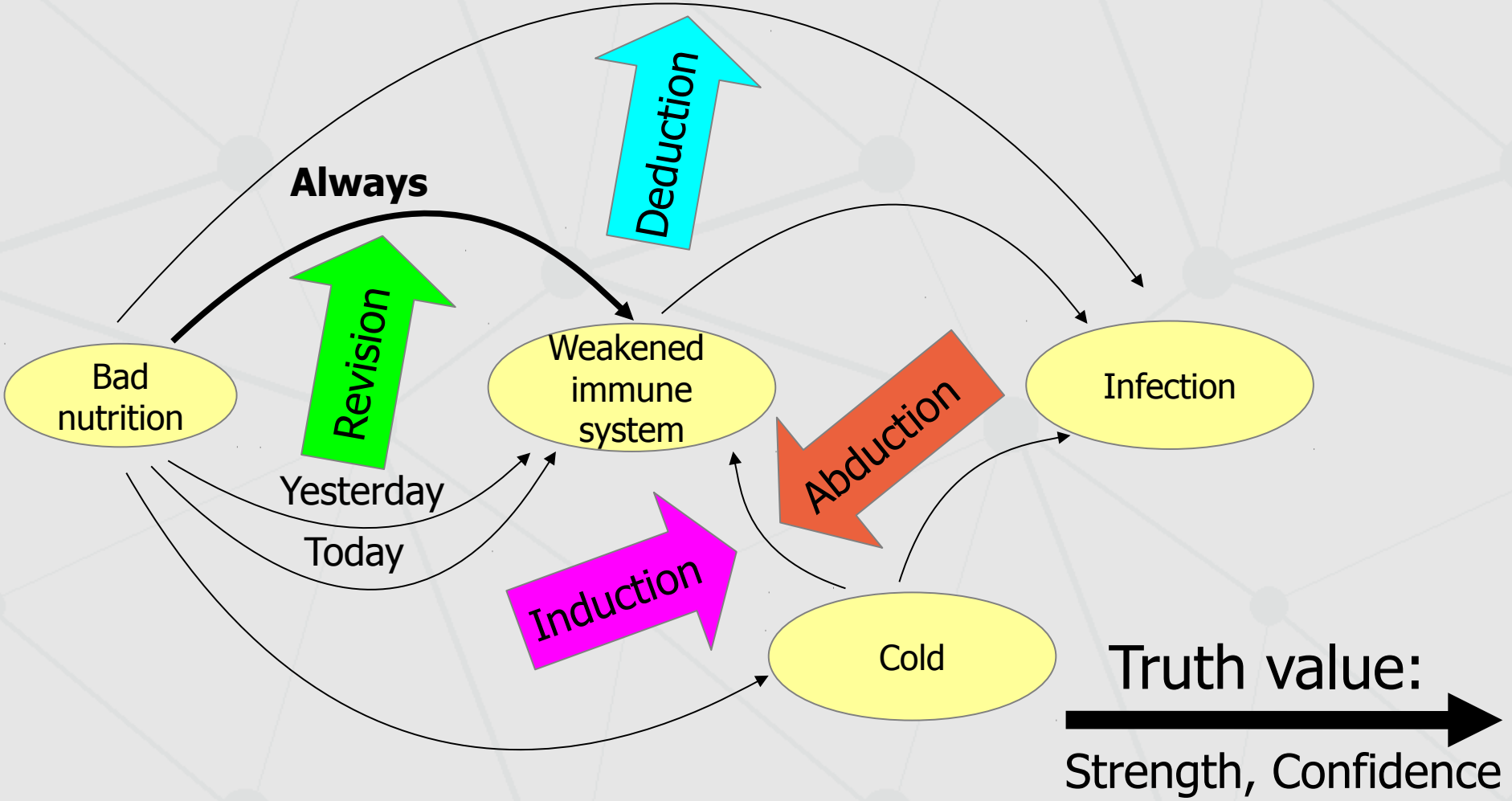
Fuzzy Logic: Separating Strength and Confidence



P.Wang (1993,2000,2013)

Distributed knowledge engineering and (social) evidence-based knowledge representation in multi-agent systems

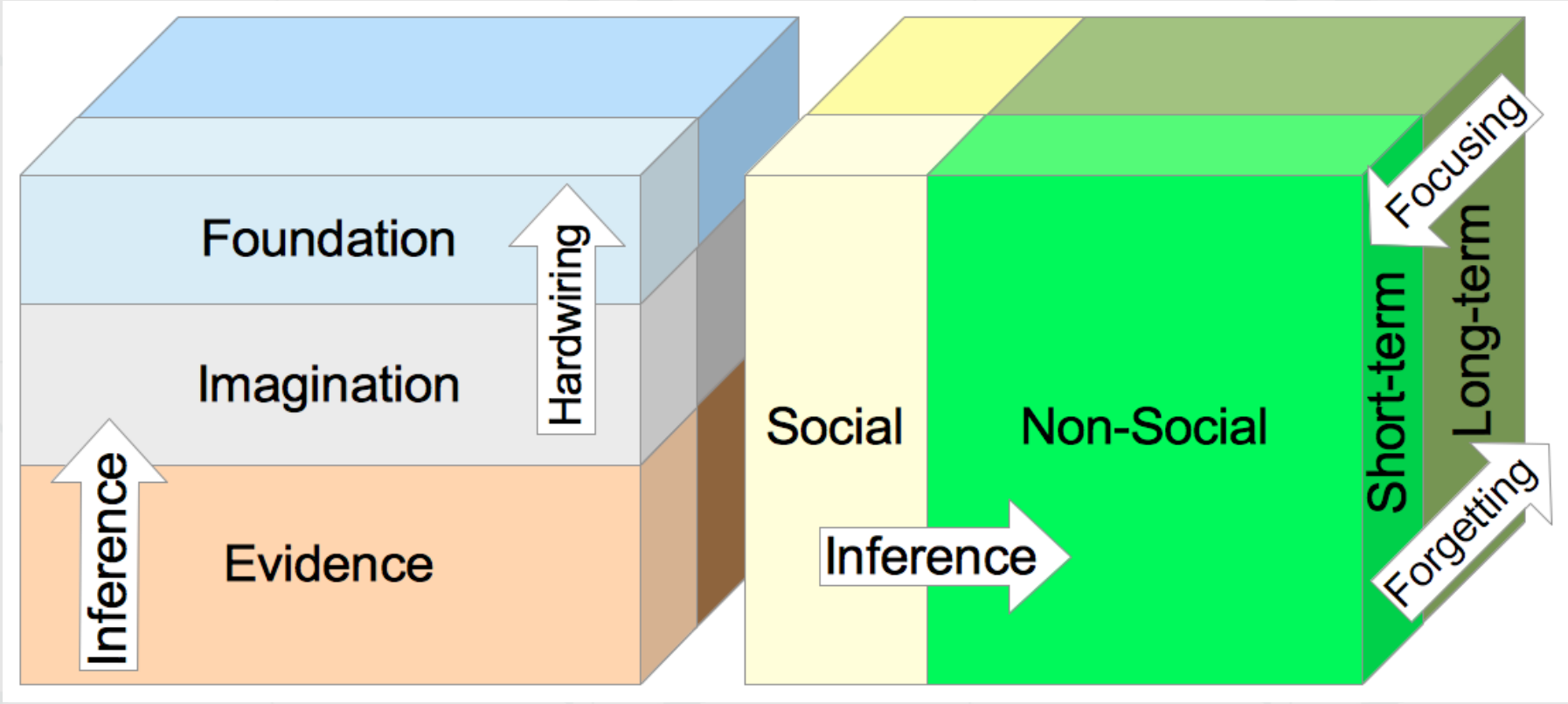
Fuzzy Logic: Non-Axiomatic Reasoning System (NARS)



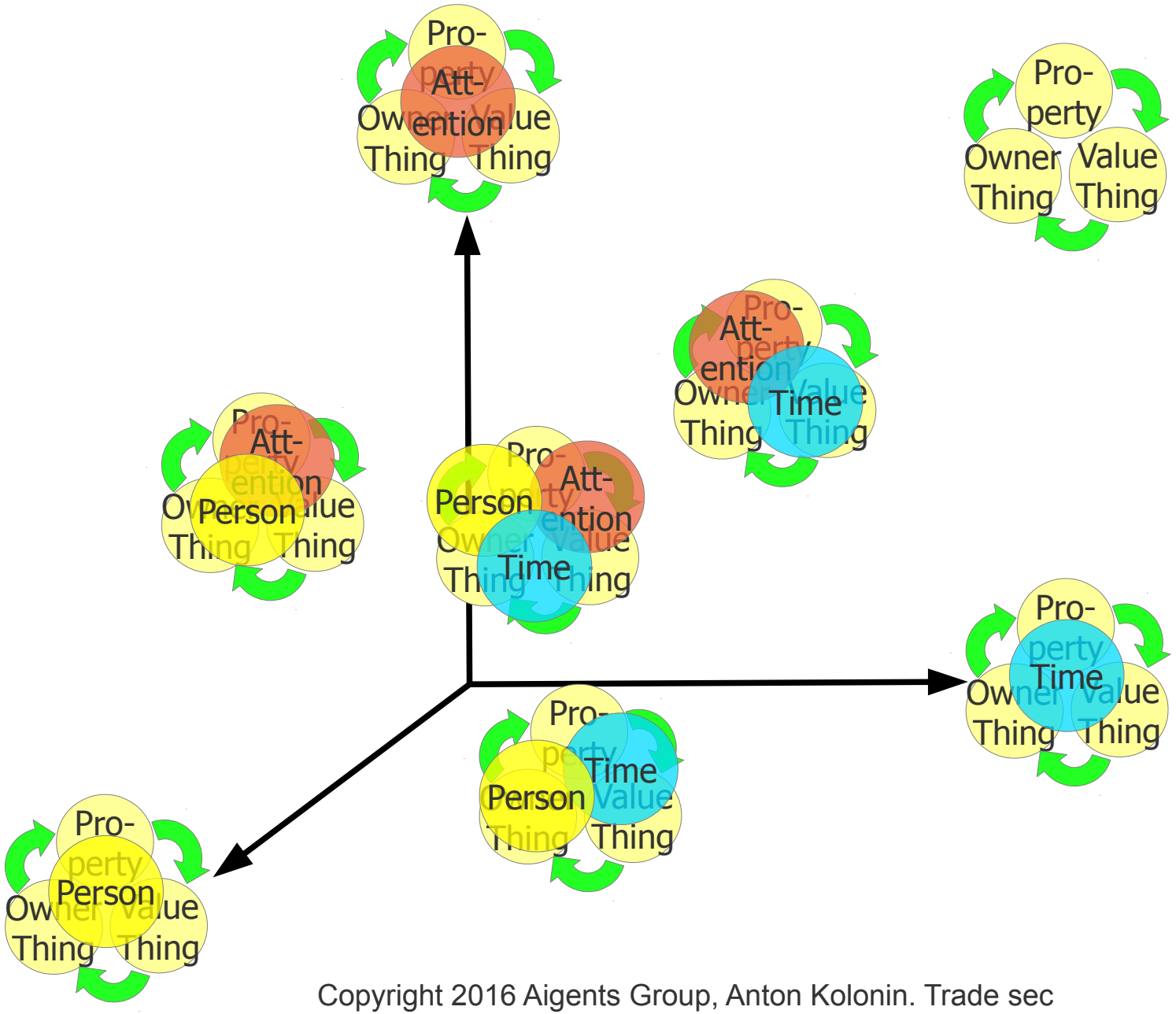
P.Wang (1993,2000,2013)

Three different segments of the graph representation of the entire set of system knowledge.

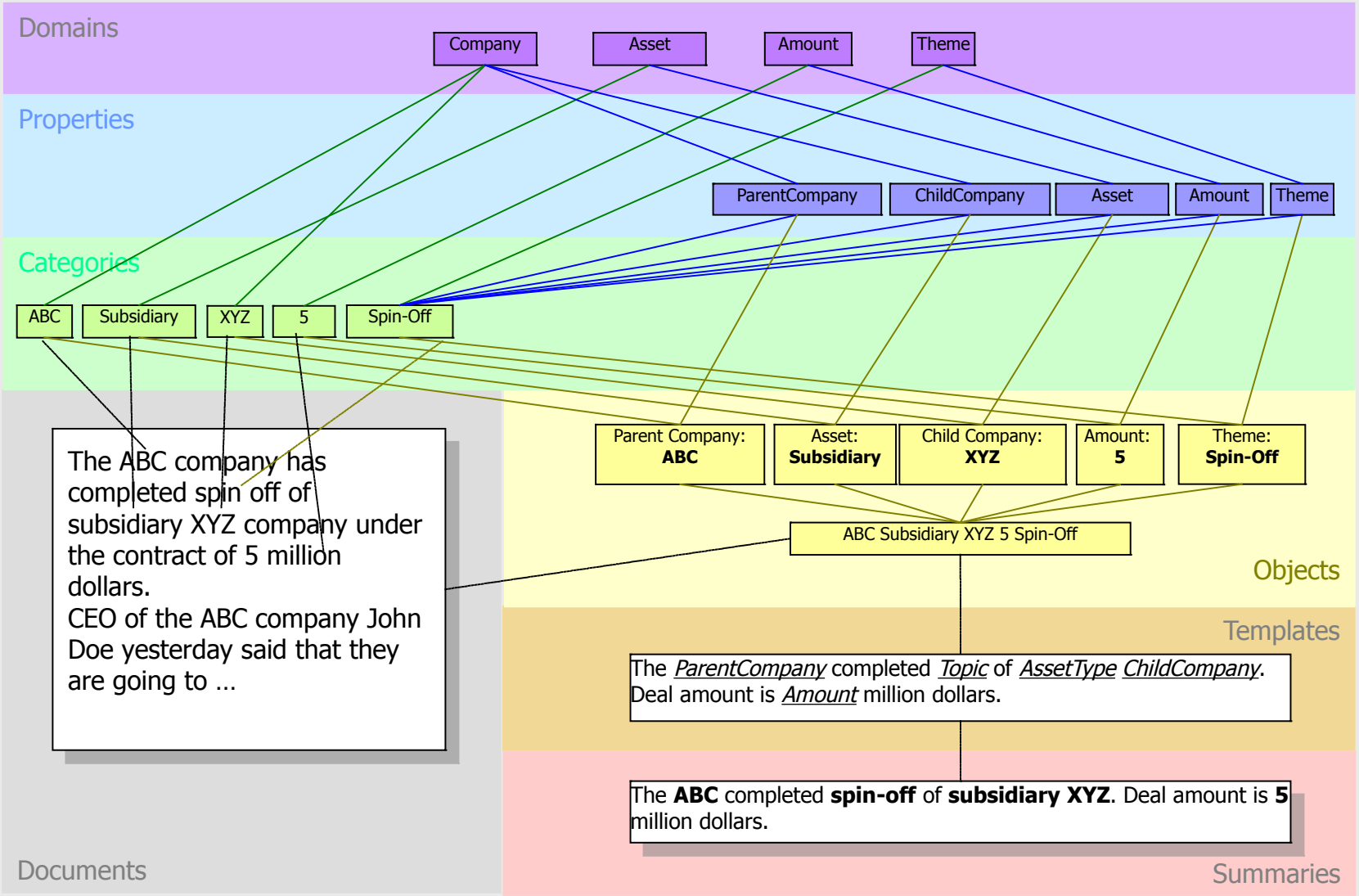
Top layer - “foundation graph” of basic knowledge, which is necessary for social system to be shared by all of its members in order to communicate. Middle layer - “imagination graph” keeping “inferred” knowledge. Bottom layer - “evidence graph” containing everyday life-time experiences.



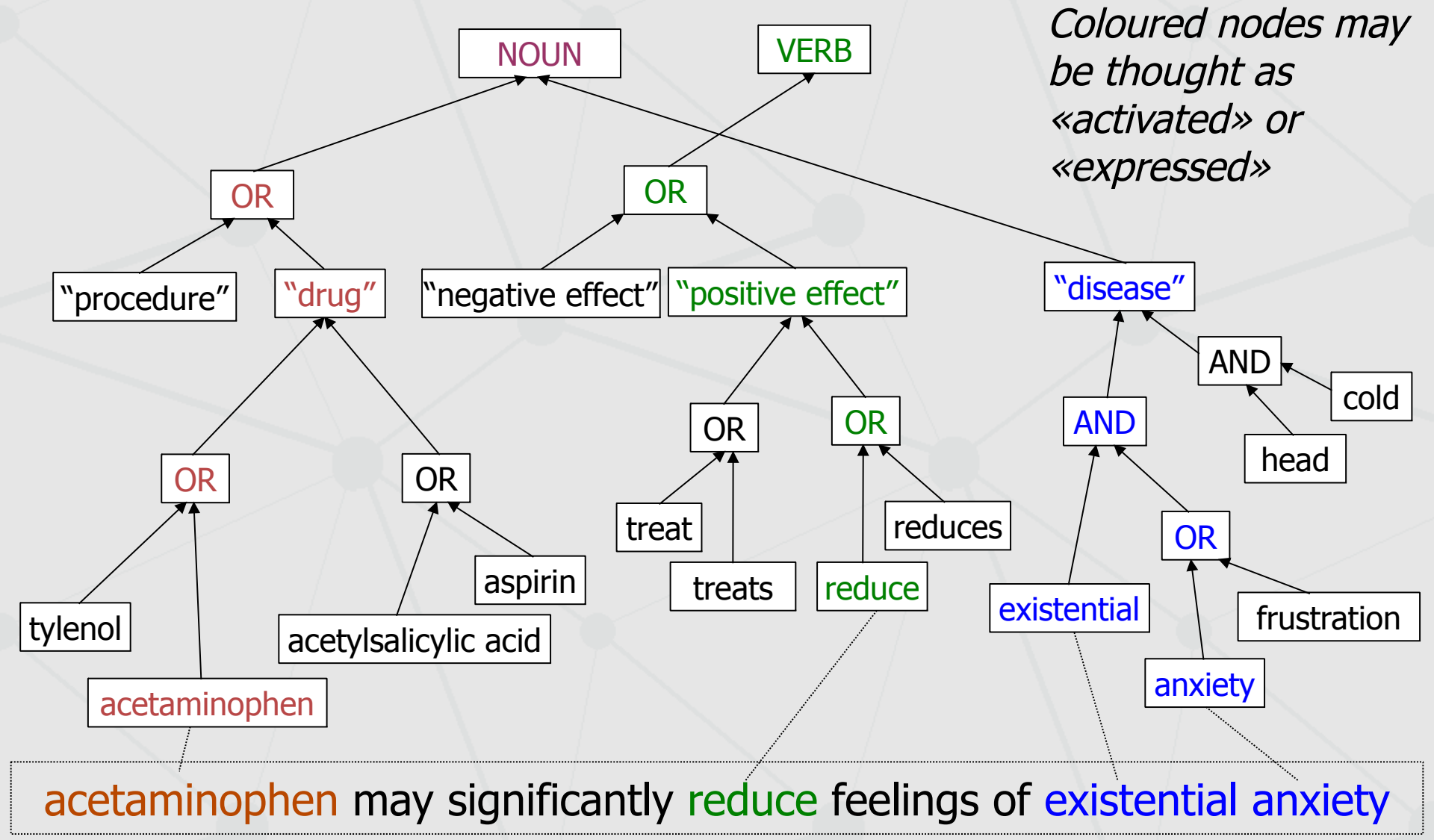
The social-time-attention continuum of knowledge



Using Graphs with Texts: Content Extraction Example



Using Graphs with Texts: Representing Linguistic Structures



Coloured nodes may be thought as «activated» or «expressed»

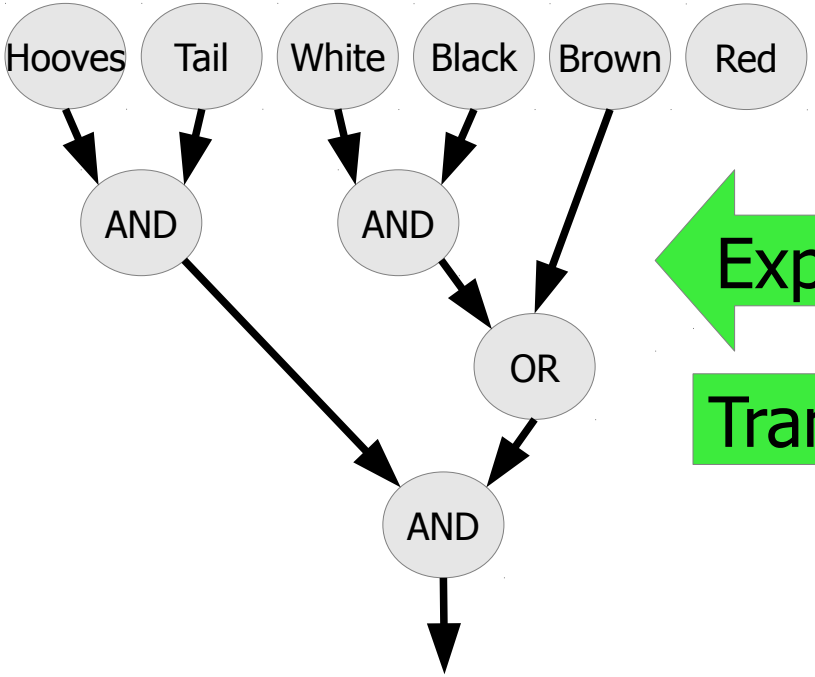
Grammar & Ontology Graph

Hand giving

Grammar & Ontology Graph Expression

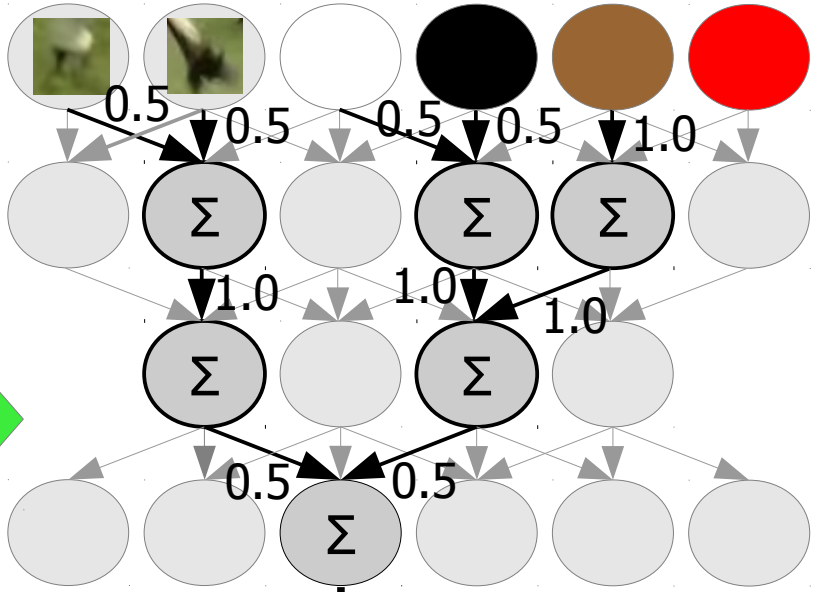
Hand giving

Bridging the Symbolic-Subsymbolic gap for "explainable AI" and "transfer learning"



Explain

Transfer



(Hooves AND Tail) AND
((White and Black) OR Brown)

=> Horse

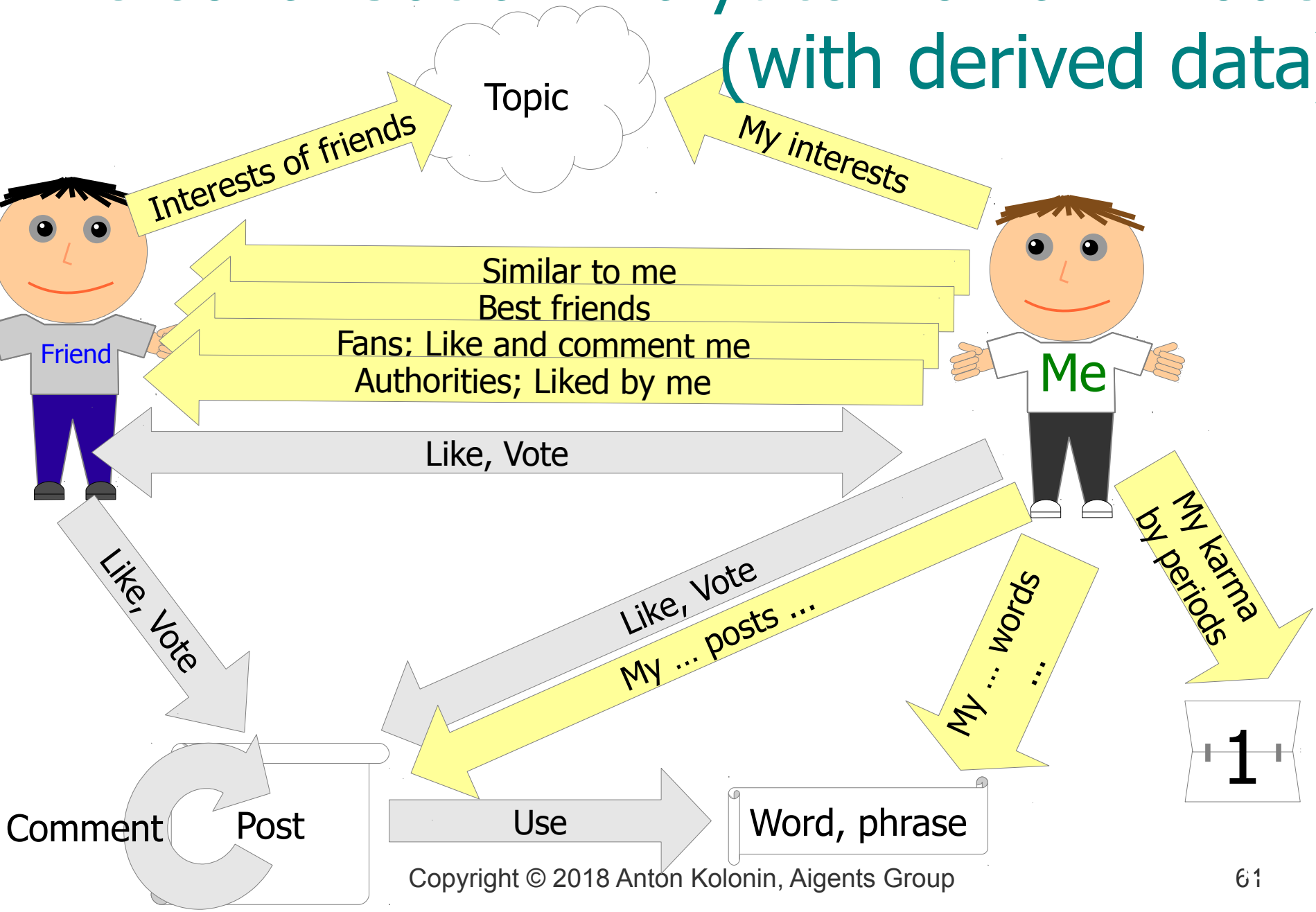
Personalized Social Connectivity and Reputation

Monitoring dynamics in online networks with Aigents platform

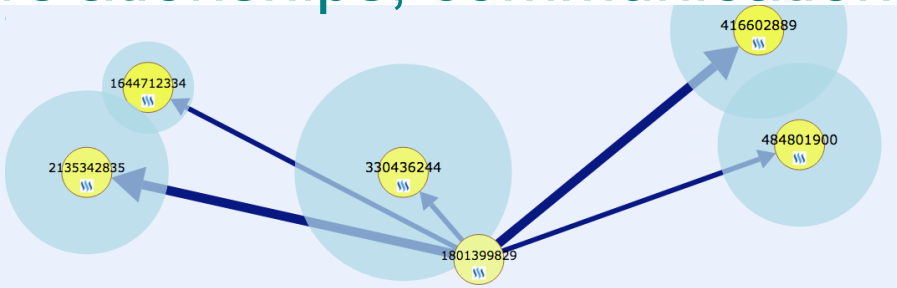
Anton Kolonin
akolonin@aigents.com

(Slides from earlier presentation on the matter)

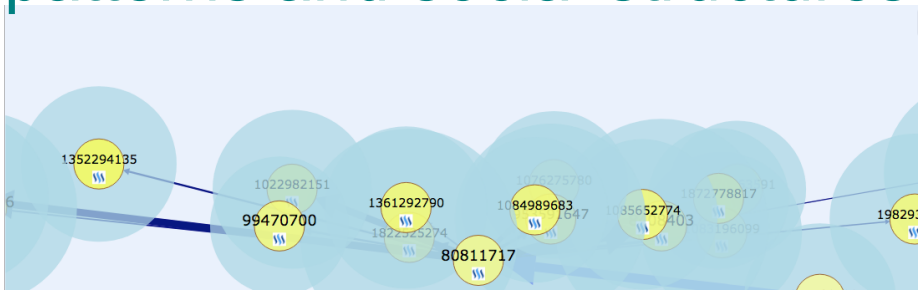
Personal Social Analytics Domain Model (with derived data)



Case 2: Helping users to understand themselves better and perform more efficiently online – using their tracks in social networks and online resources, capture their interests, relationships, communication patterns and social structures.

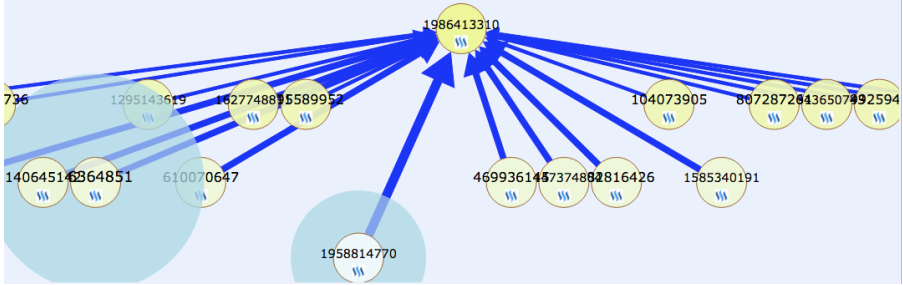


Social type: follower

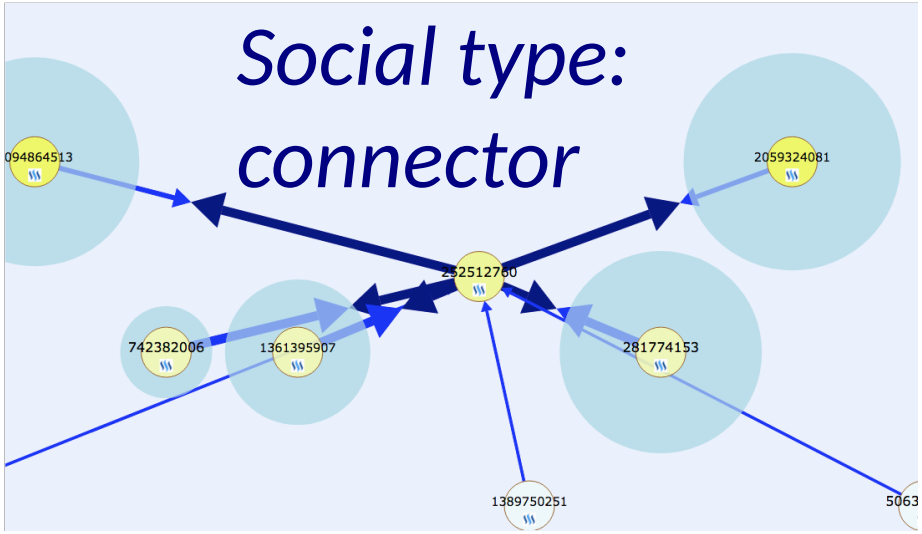


Social type: peer

Social type: opinion leader

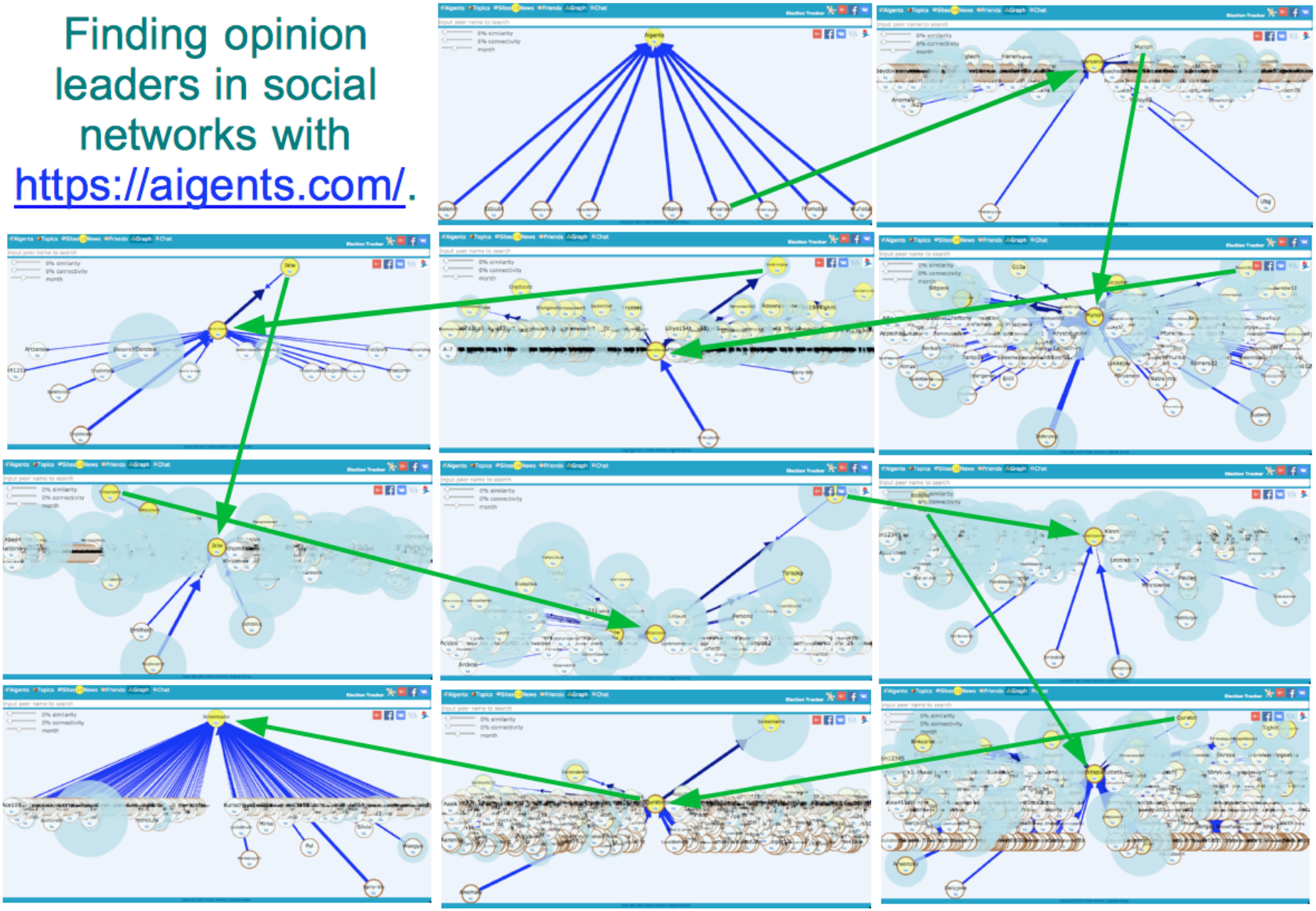


Social type: connector

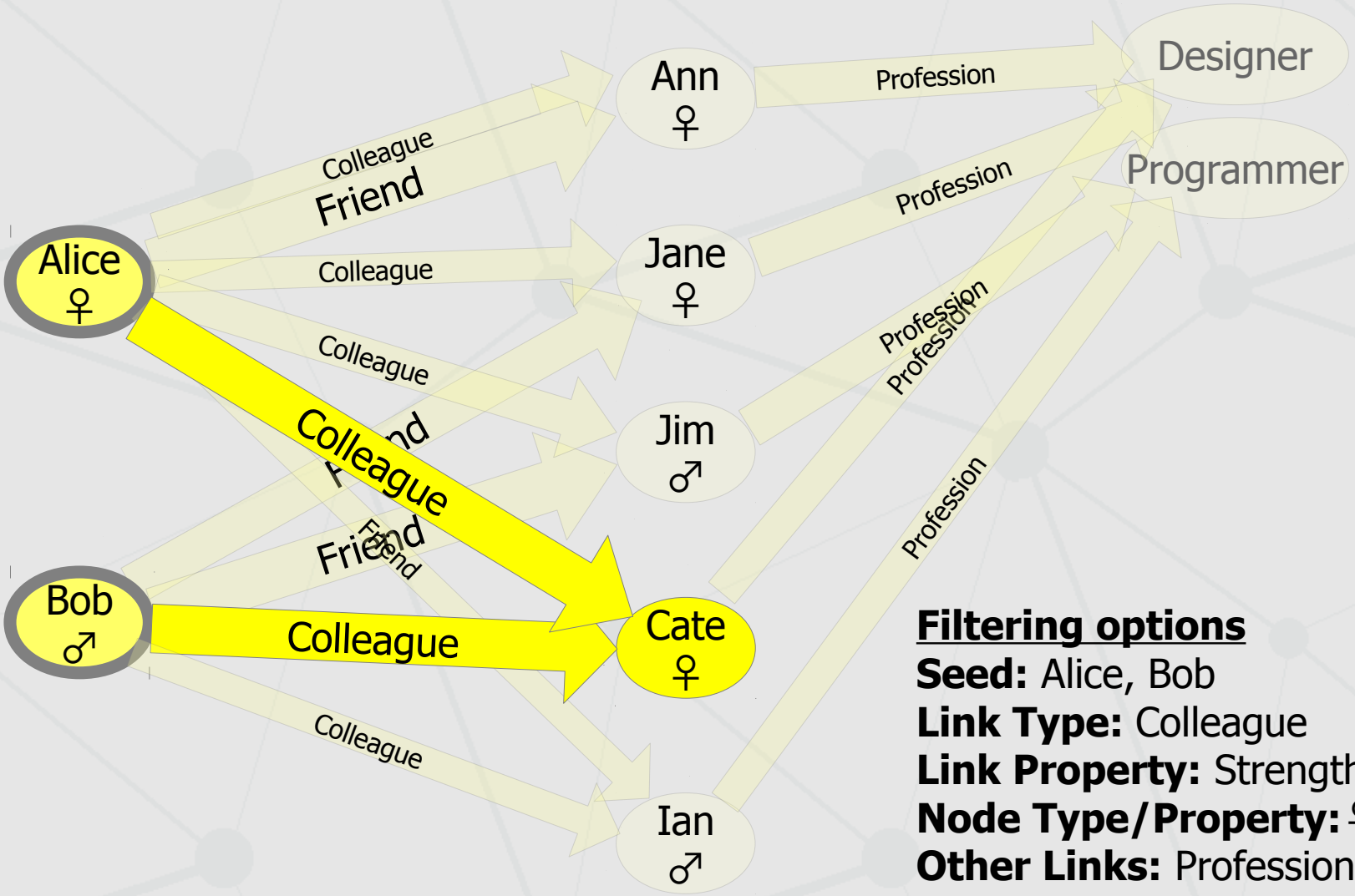


Case 3:

Finding opinion leaders in social networks with https://aigents.com/.



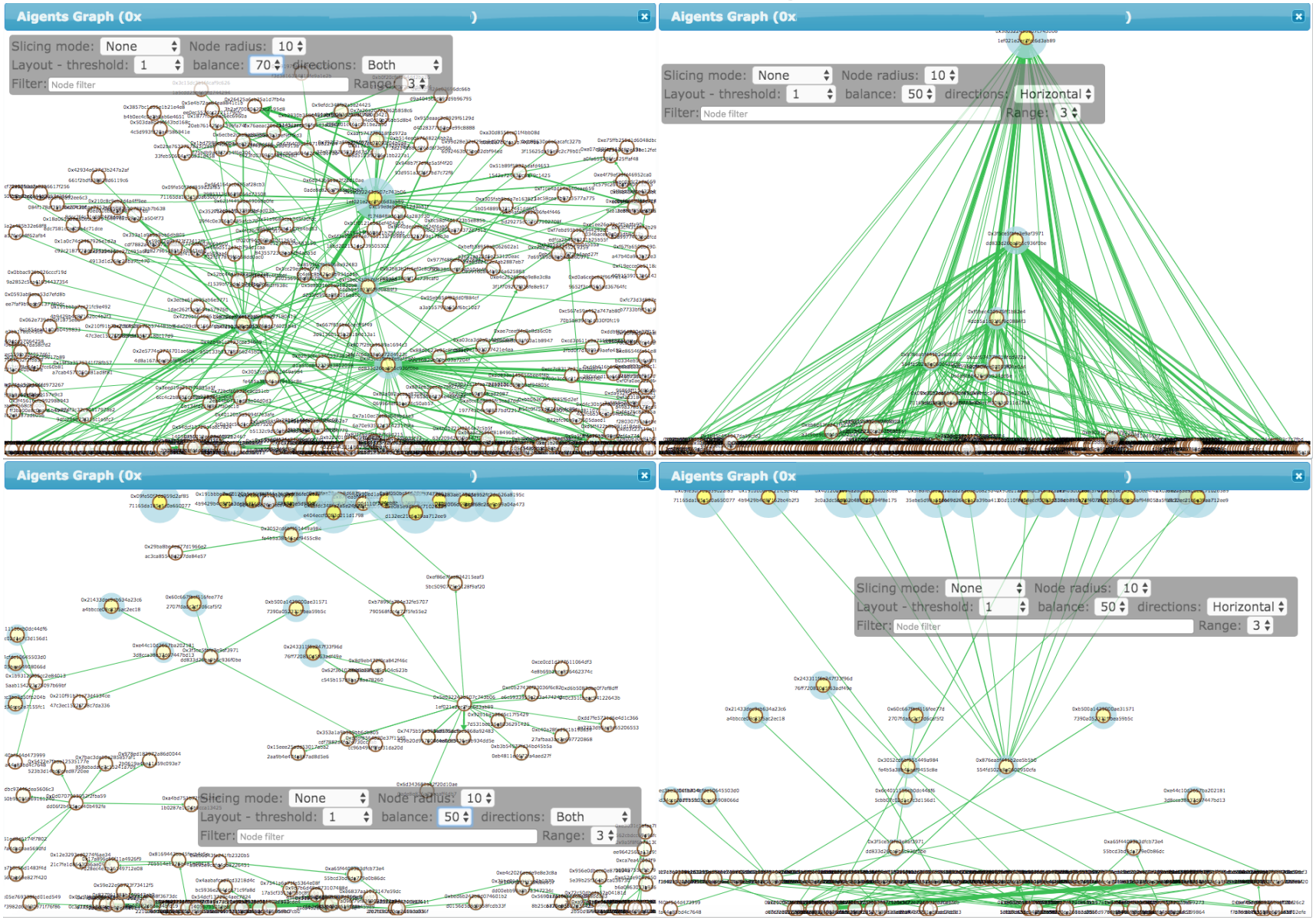
Graph Builder/Filter Concept



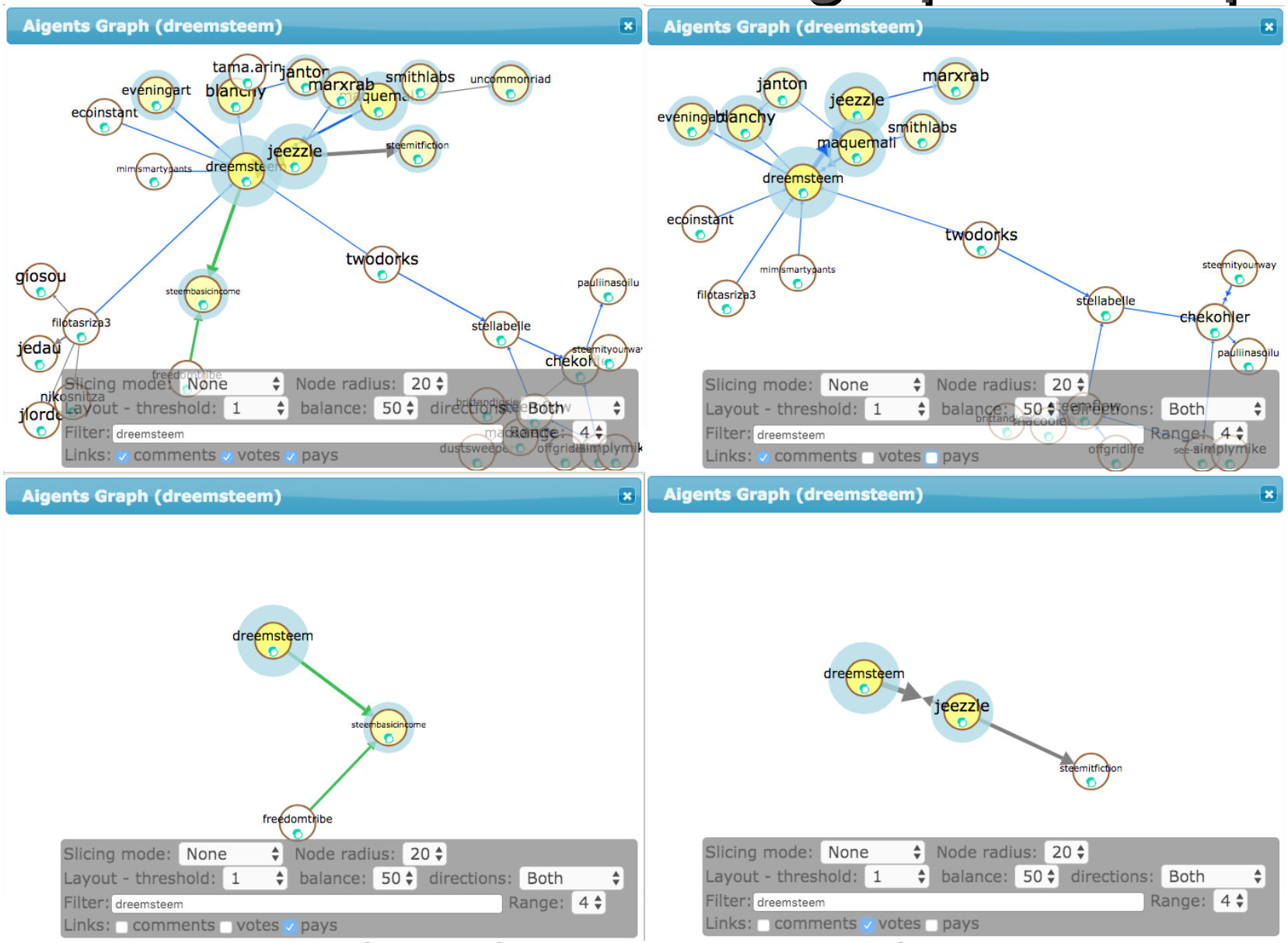
Filtering options

Seed: Alice, Bob
Link Type: Colleague
Link Property: Strength=High
Node Type/Property: ♀
Other Links: Profession=Designer
Link Range: 1

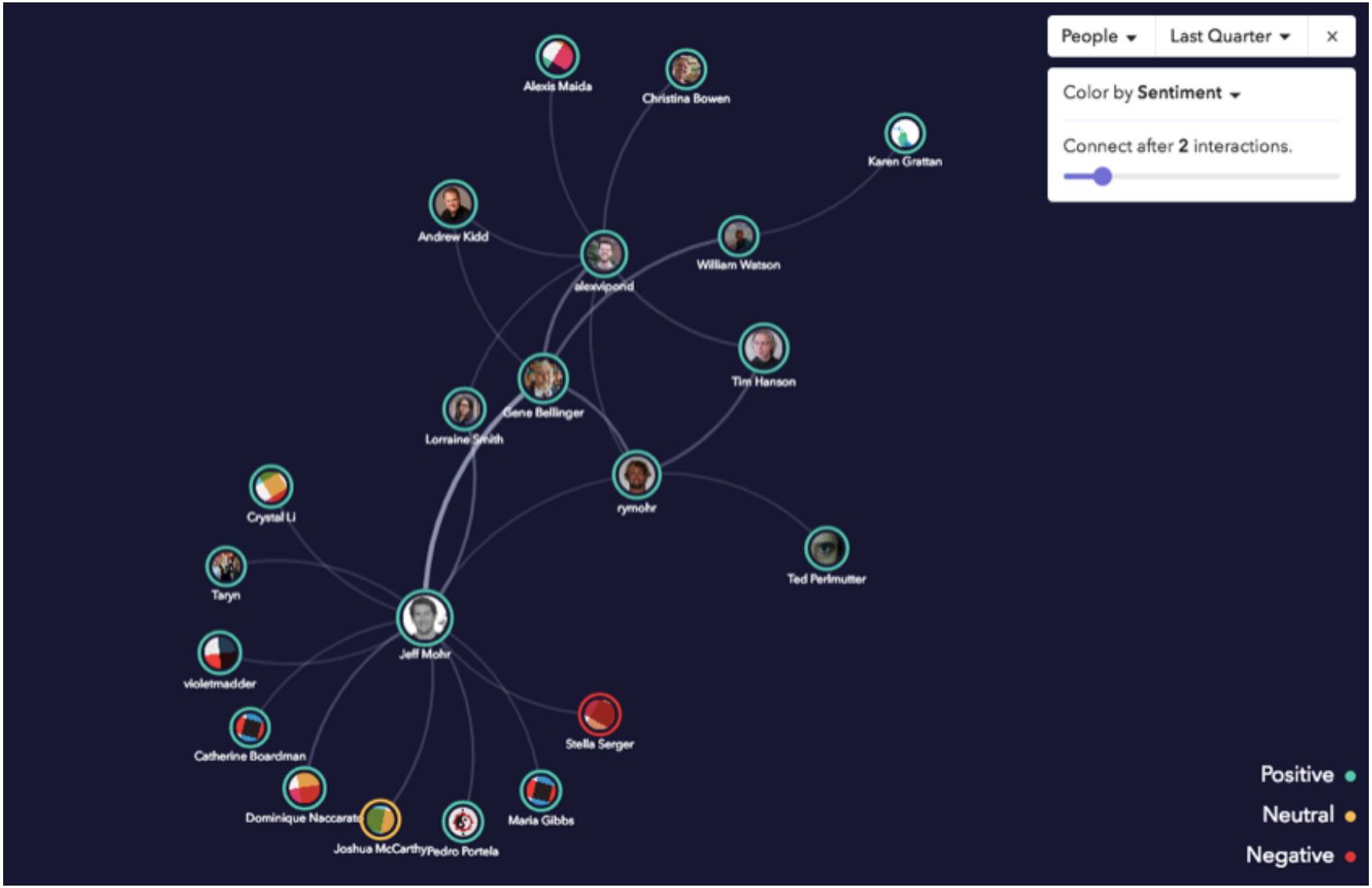
Ethereum cash flow graph example



Steemit social network graph example



Slack social graph example



https://compasshq.com/

Foundation Ontologies (simplex)