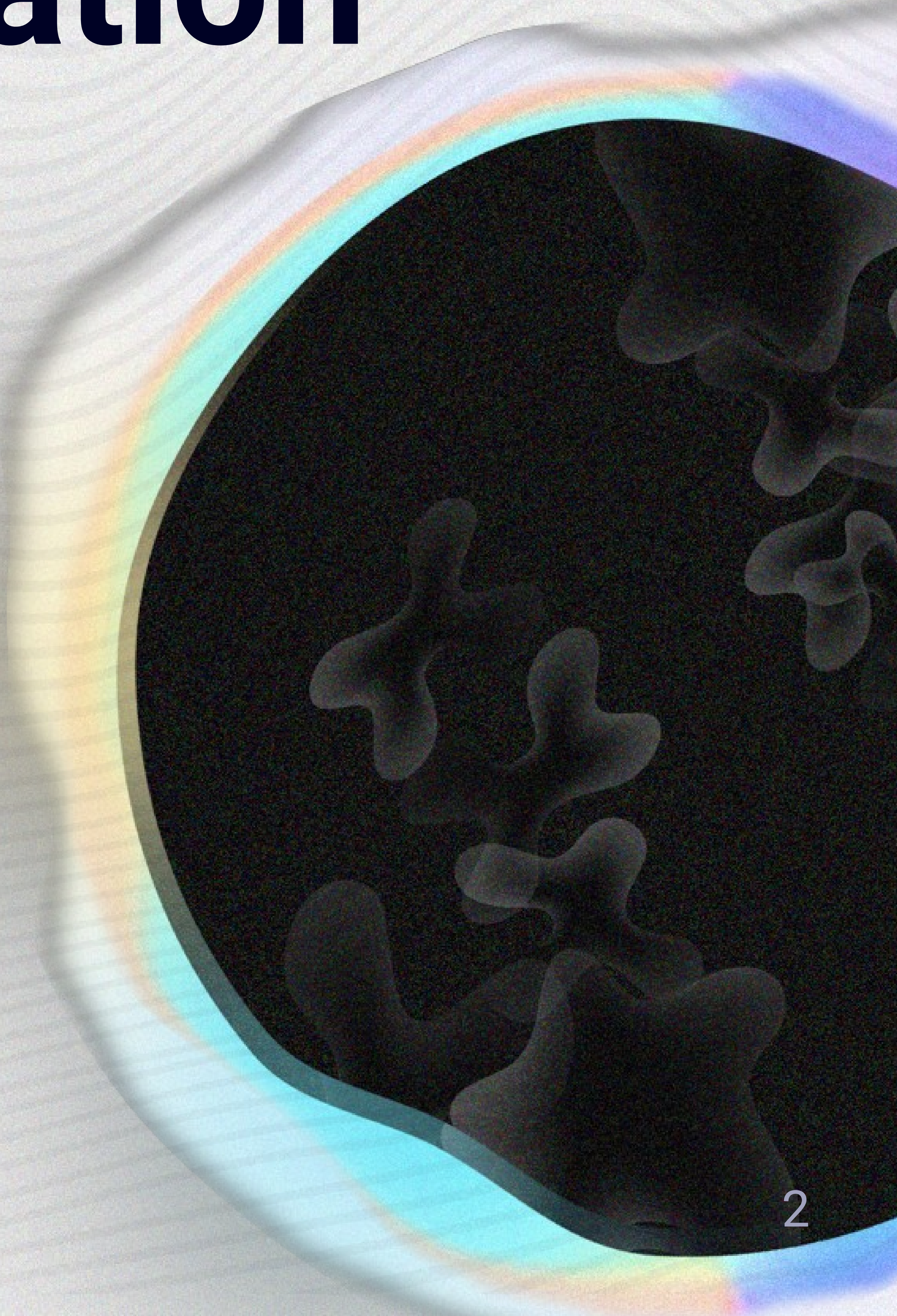
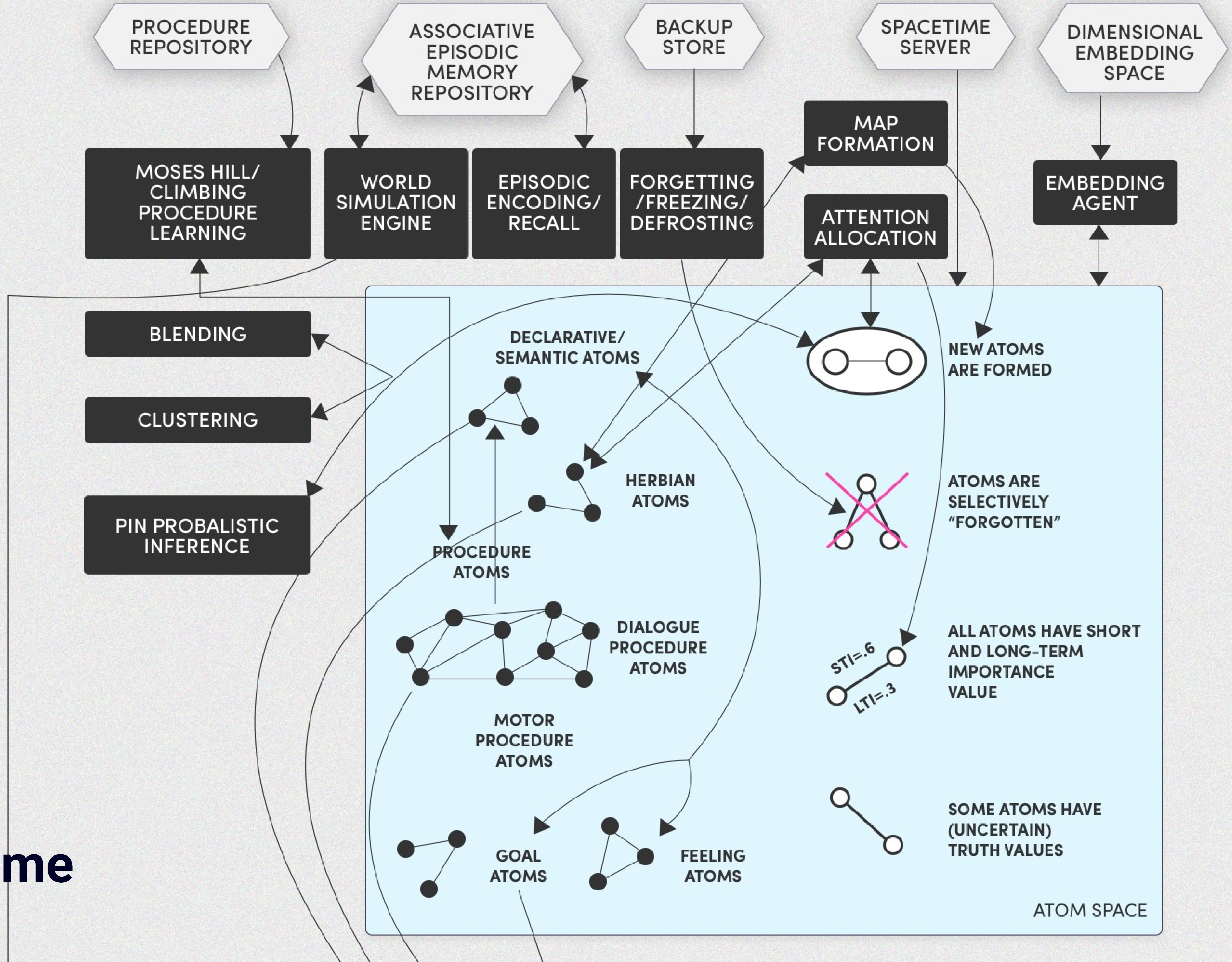
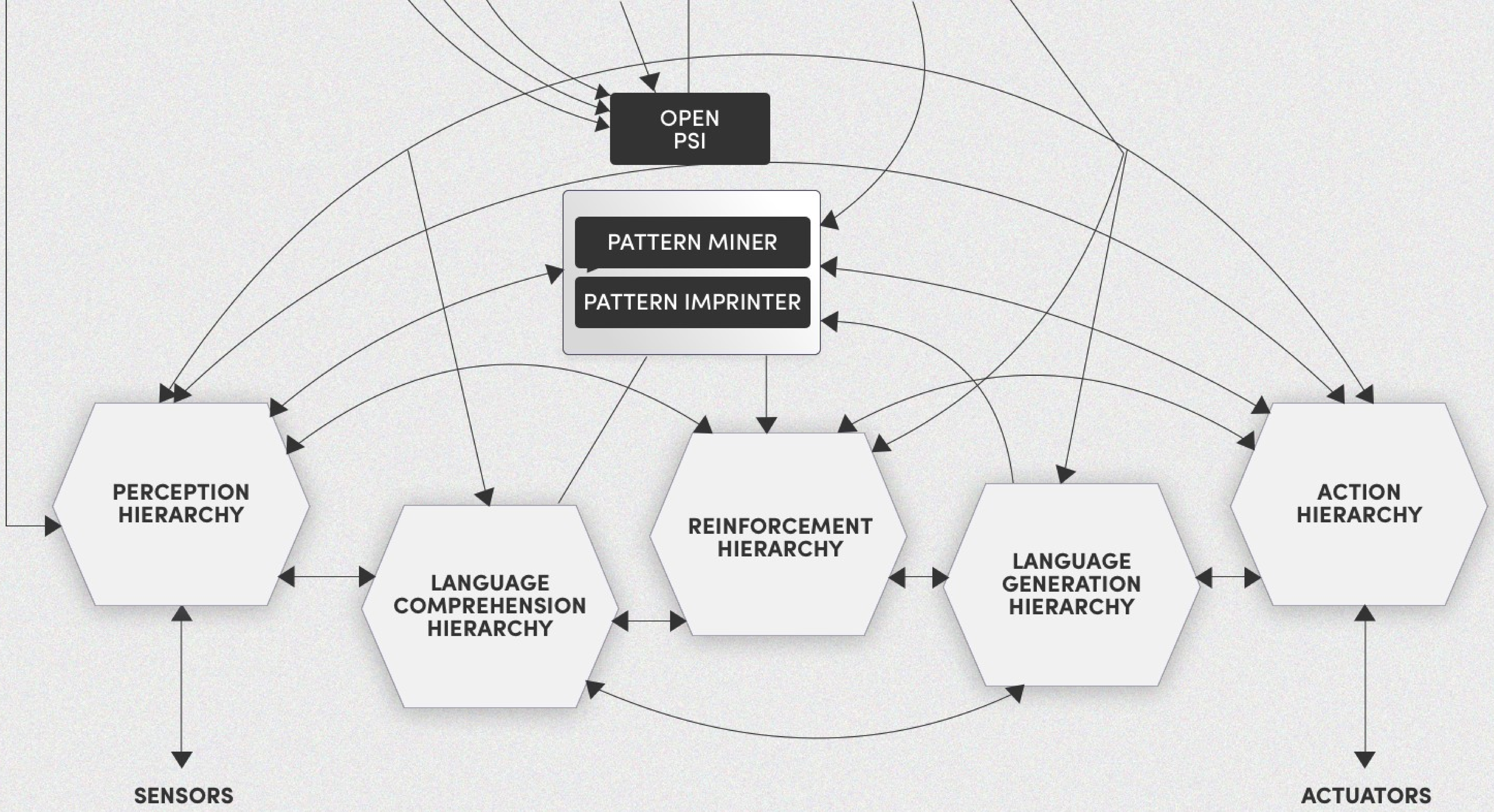
Neural-Symbolic Integration Using OpenCog



Anatoly Belikov
abelikov@singularitynet.io

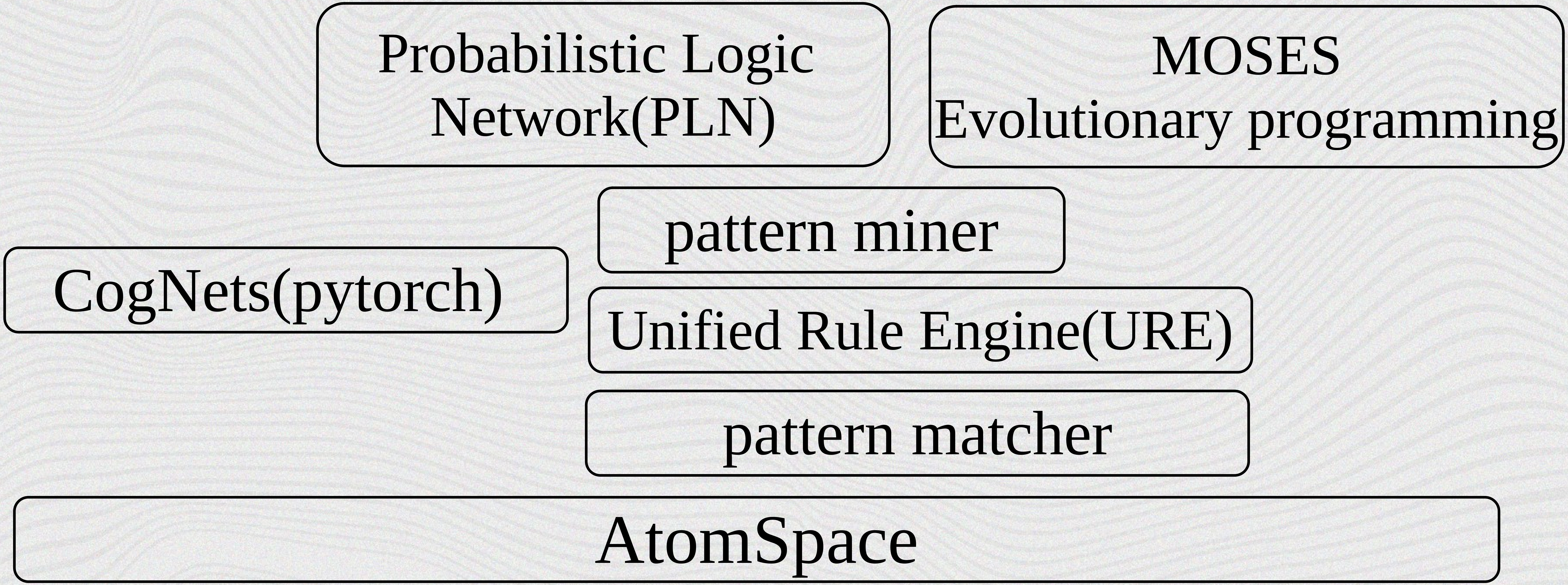

PROCEDURE REPOSITORY
ASSOCIATIVE EPISODIC MEMORY REPOSITORY
BACKUP STORE
SPACETIME SERVER
DIMENSIONAL EMBEDDING SPACE
MAP FORMATION
MOSES HILL/ CLIMBING PROCEDURE LEARNING
WORLD SIMULATION ENGINE
EPISODIC ENCODING/ RECALL
FORGETTING /FREEZING/ DEFROSTING
ATTENTION ALLOCATION
EMBEDDING AGENT
BLENDING
CLUSTERING
PIN PROBALISTIC INFERENCE
DECLARATIVE/ SEMANTIC ATOMS
HERBIAN ATOMS
PROCEDURE ATOMS
DIALOGUE PROCEDURE ATOMS
MOTOR PROCEDURE ATOMS
GOAL ATOMS
FEELING ATOMS
NEW ATOMS ARE FORMED
ATOMS ARE SELECTIVELY "FORGOTTEN"
STI=.6
LTI=.3
ALL ATOMS HAVE SHORT AND LONG-TERM IMPORTANCE VALUE
SOME ATOMS HAVE (UNCERTAIN) TRUTH VALUES
ATOM SPACE

CogPrime


OPEN PSI
PATTERN MINER
PATTERN IMPRINTER
PERCEPTION HIERARCHY
LANGUAGE COMPREHENSION HIERARCHY
REINFORCEMENT HIERARCHY
LANGUAGE GENERATION HIERARCHY
ACTION HIERARCHY
SENSORS
ACTUATORS

CogPrime

OpenCog



Probabilistic Logic Network(PLN)

MOSES
Evolutionary programming

pattern miner

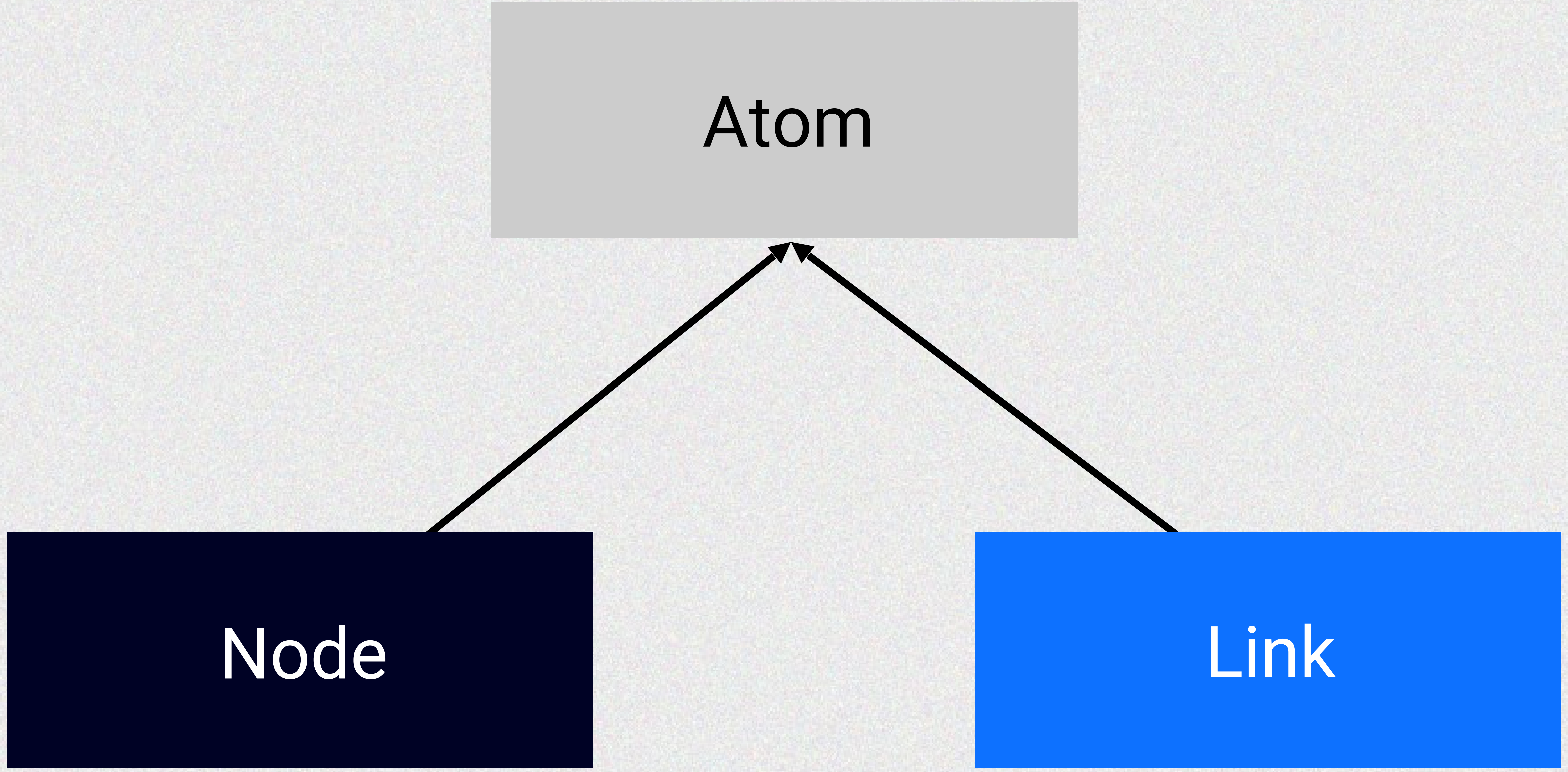
CogNets(pytorch)

Unified Rule Engine(URE)

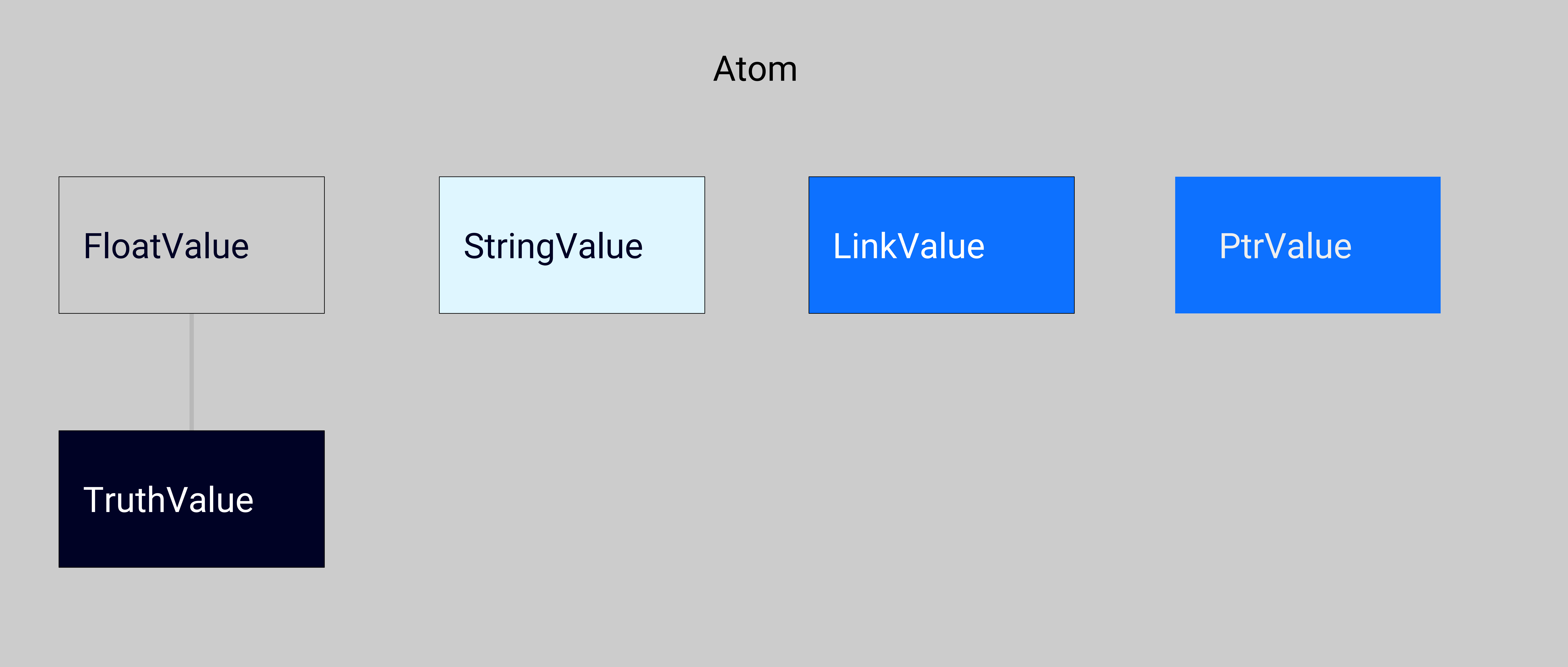
pattern matcher

AtomSpace

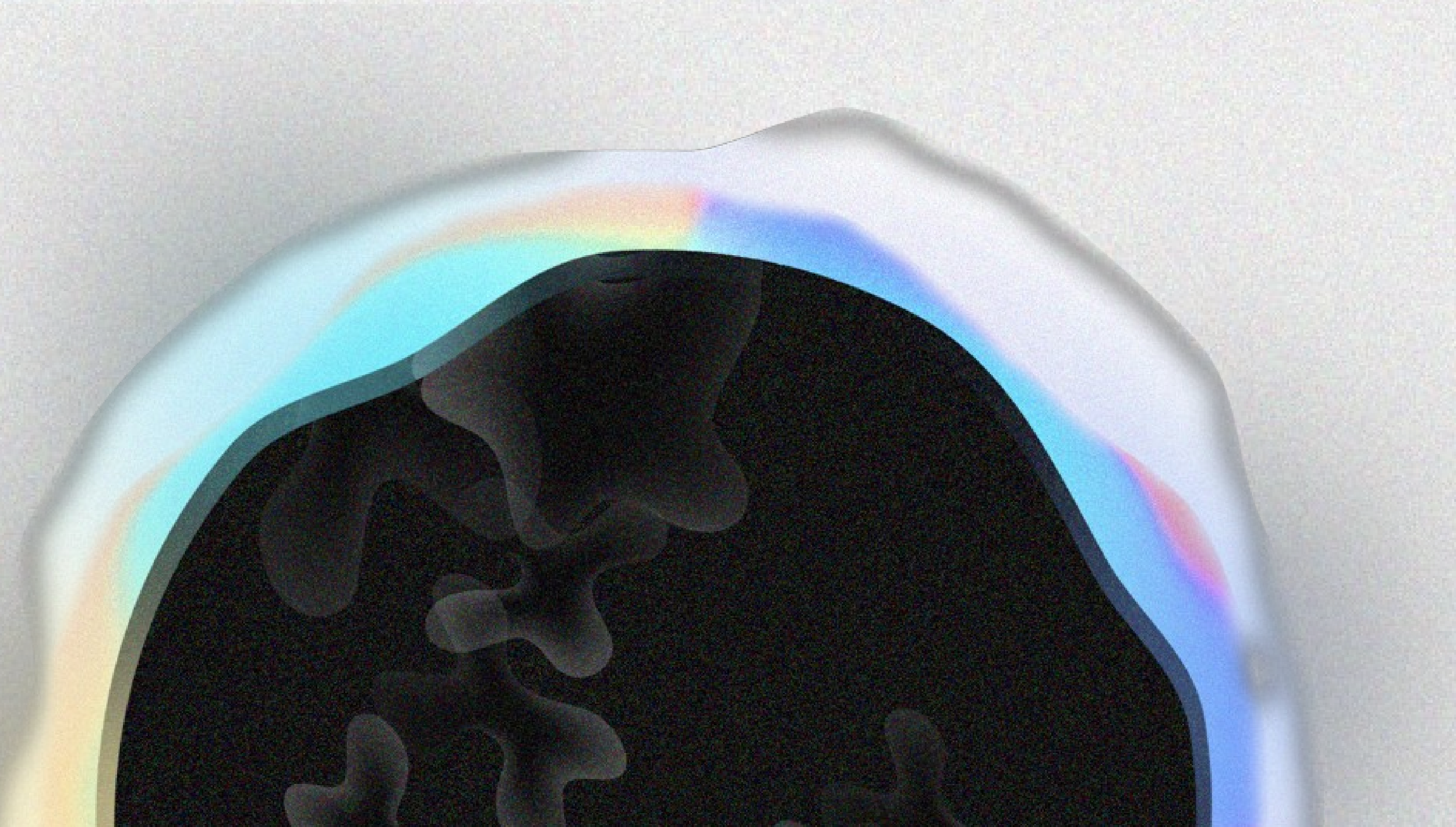
Types of data in atomspace: Atoms



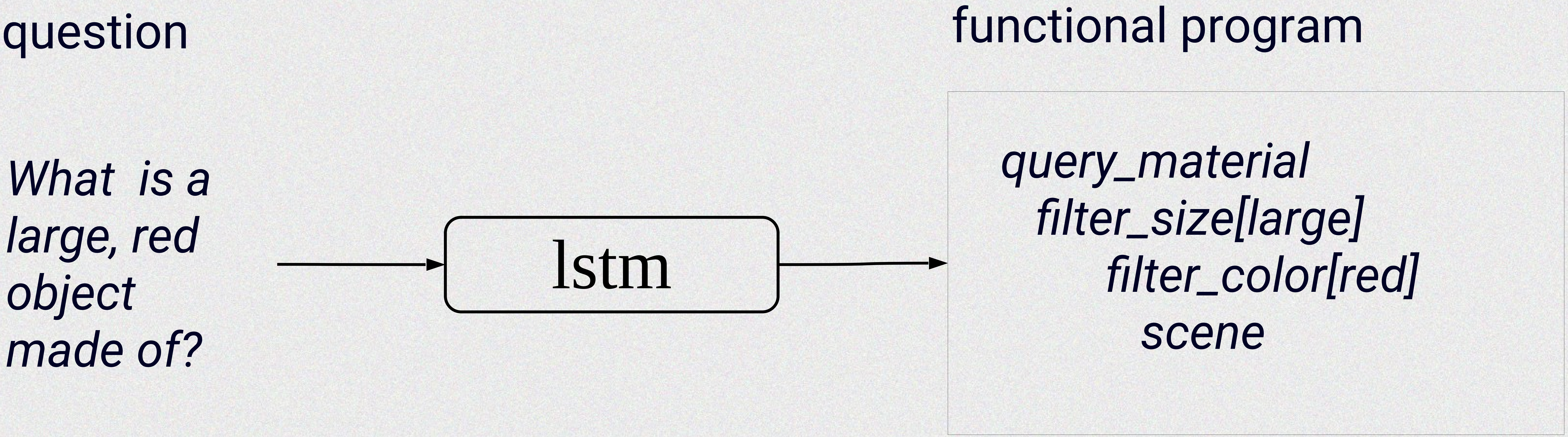
Types of data in atomspace: Values



Code walkthrough and demonstration



Module networks for VQA



question

What is a large, red object made of?

lstm

functional program

```
query_material
  filter_size[large]
      filter_color[red]
         scene
```

Mascharka, David, et al. "Transparency by design: Closing the gap between performance and interpretability in visual reasoning."

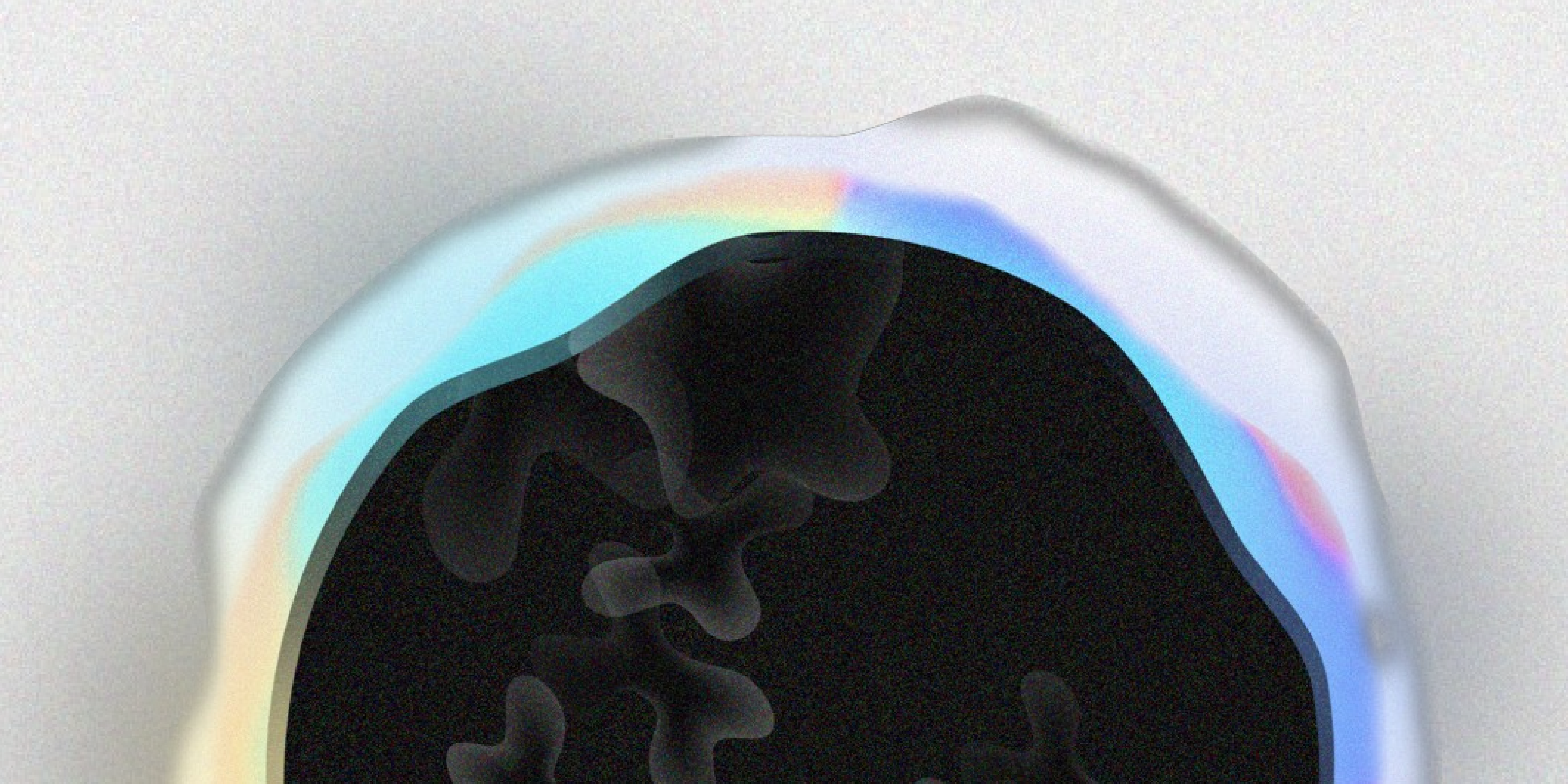
Module networks for VQA

functional program

```
query_material
  filter_size[large]
      filter_color[red]
         scene
```

execution

```
out1 = filter_color(red, attention, features)
out2 = filter_size(large, out1, features)
result = classify(material, out2, features)
```

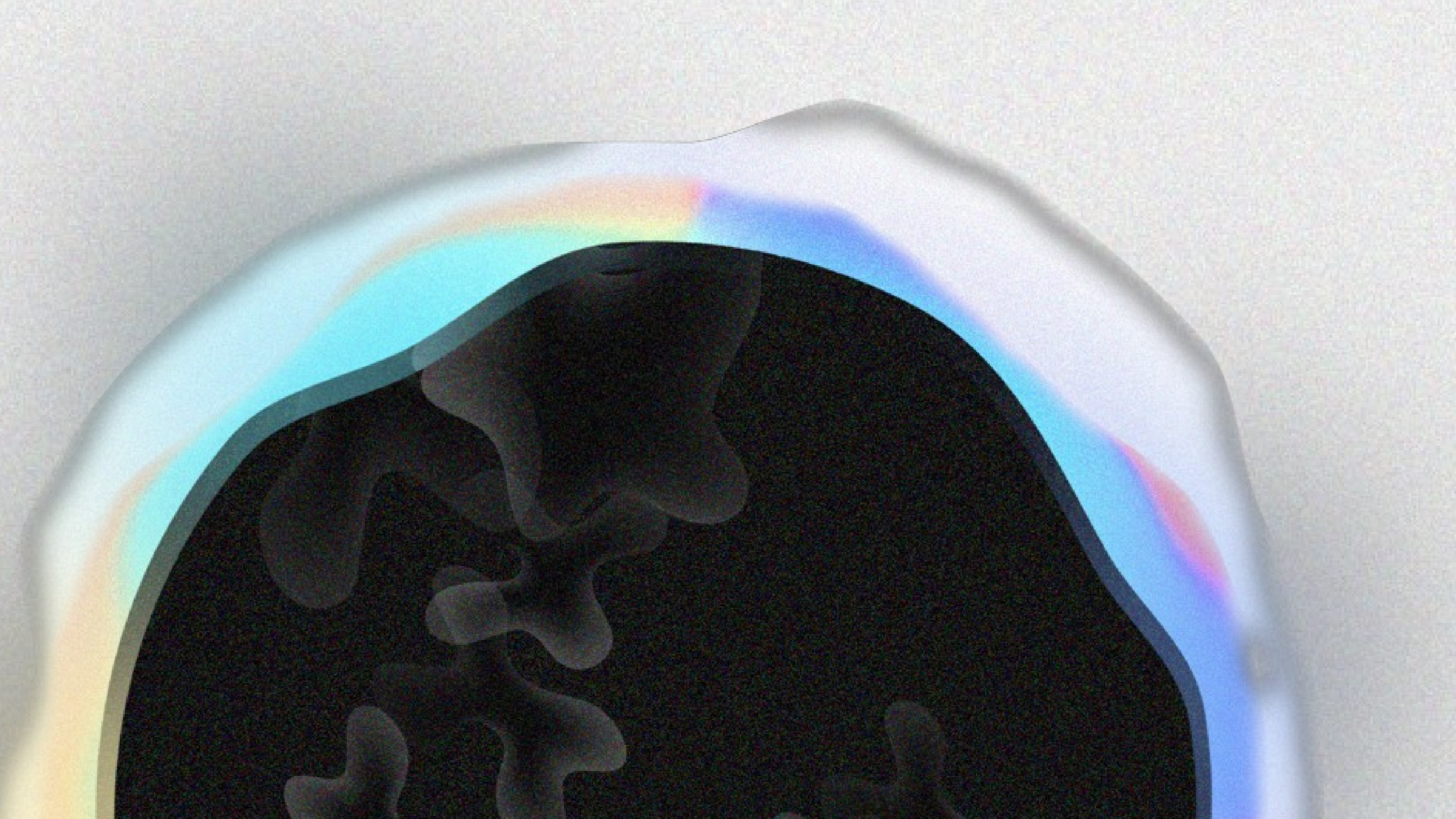


Using knowledge base

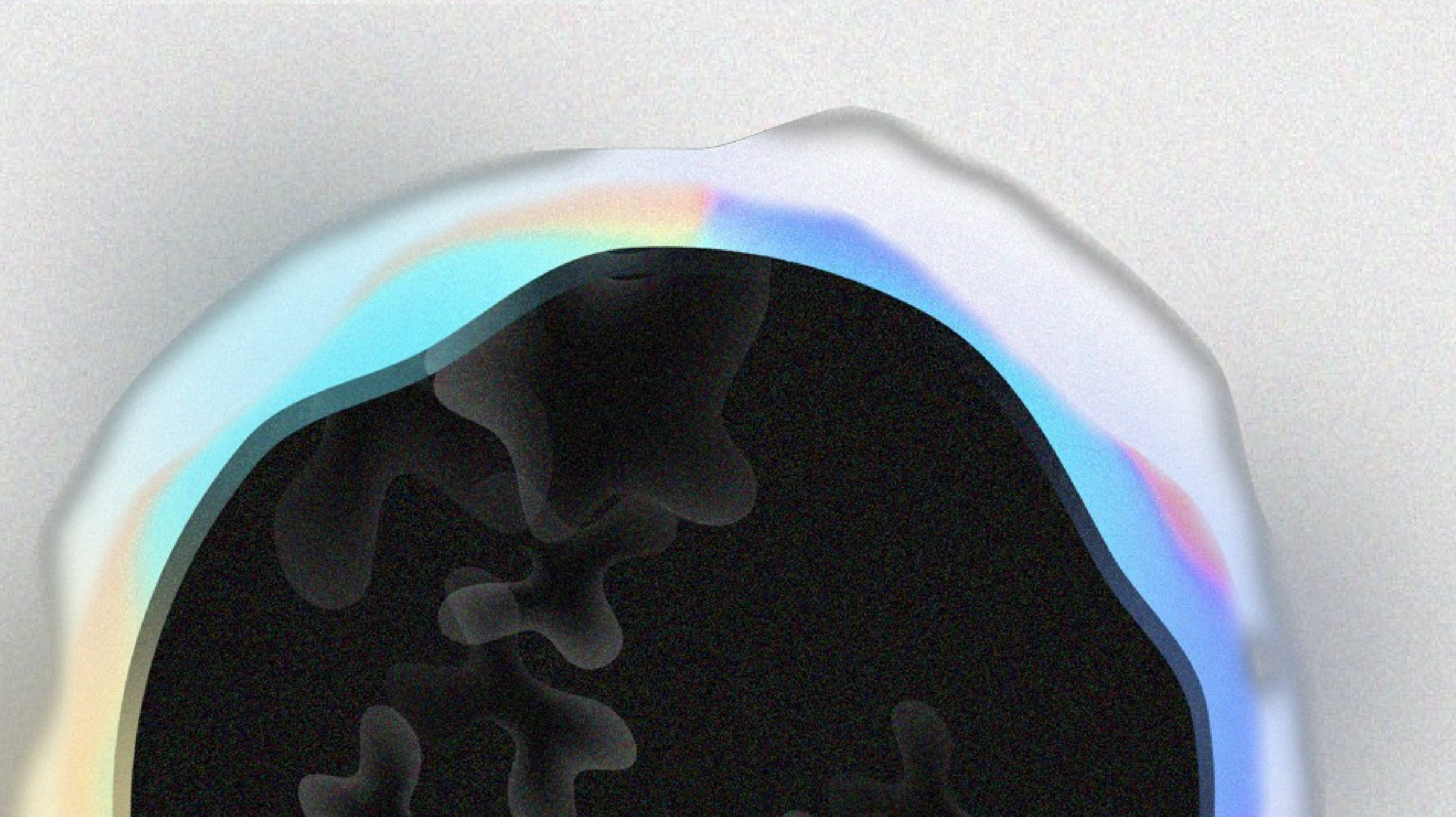
functional program

```
query_material
  filter_size[large]
      filter_color[red]
        scene
```

```
out1 = filter_color(red, attention, features)
out2 = filter_size(large, out1, features)
out3 = filter_material(Variable(“X”),
                         out2, features)
result = argmax(out3)
```



Code walkthrough and demonstration



Functional vs Logical programs for VQA

Is there a large, rubber object that is red?

Nested functional program:

```
nonempty(
   filter(red,
         filter(large,
                  filter(rubber, Attention, Features)
                  , Features)
         Features))
```

Logical program:

```
filter(red, X, Features),
filter(large, Y, Features),
filter(rubber, Z, Features),
intersect(X, Y, Z, Result),
nonempty(Result).
```

GQA-opencog – logical programs for VQA

Is the horse on the edge of water brown and small?

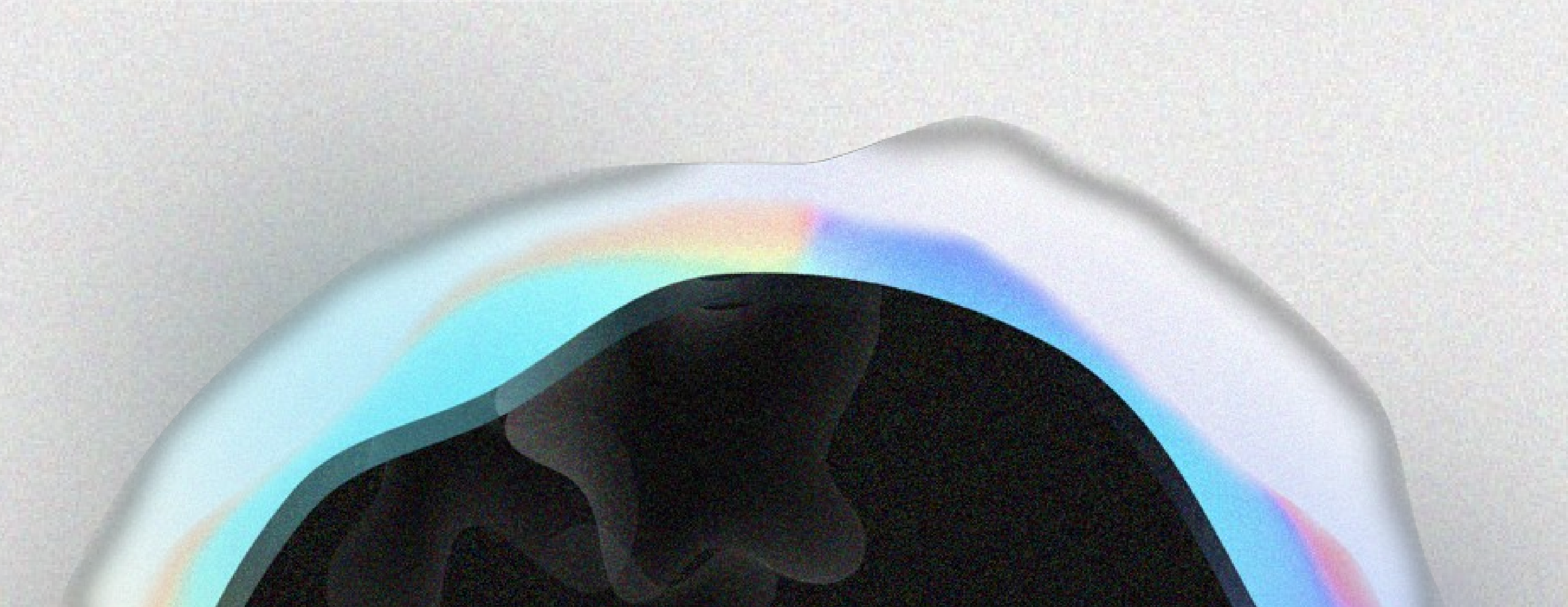
```
[{"argument": "water (447019)",
 "dependencies": [],
 "operation": "select"},
{"argument": "horse,on the edge of,s (447018)",
 "dependencies": [0],
 "operation": "relate"},
{"argument": "brown",
 "dependencies": [1],
 "operation": "verify color"},
{"argument": "small ",
 "dependencies": [1],
 "operation": "verify size"},
{"argument": "", "dependencies": [2, 3],
"operation": "and"}]
```

GQA-opencog – logical programs for VQA

Is the horse on the edge of water brown and small?

has_color(brown, $Y) and **on($Y, $Z)** and **edge_of($Z, $X)** and **object(horse, $Y)** and **object(water, $X)** and **has_size(small , $Y)**

https://github.com/noskill/GQA-opencog



TorchPLN

TTensorTruthValue – subclass of torch.Tensor

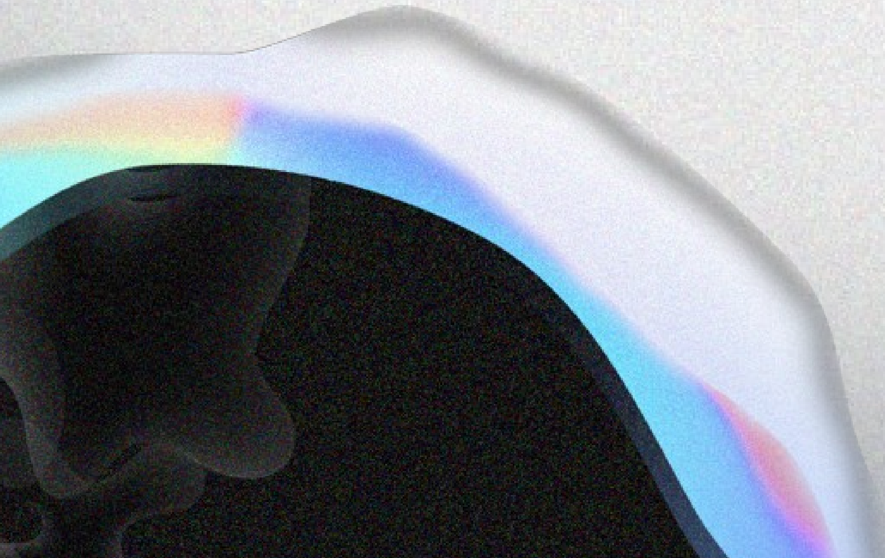
scheme

```
(define (precise-modus-ponens-strength-formula sA sAB snotAB)
        (+ (* sAB sA) (* snotAB (negate sA))))
```

python

```
def precise_modus_ponens_strength_formula(sA, sAB, snotAB):
      return sAB * sA + snotAB * (1 - sA)
```

https://github.com/singnet/pln



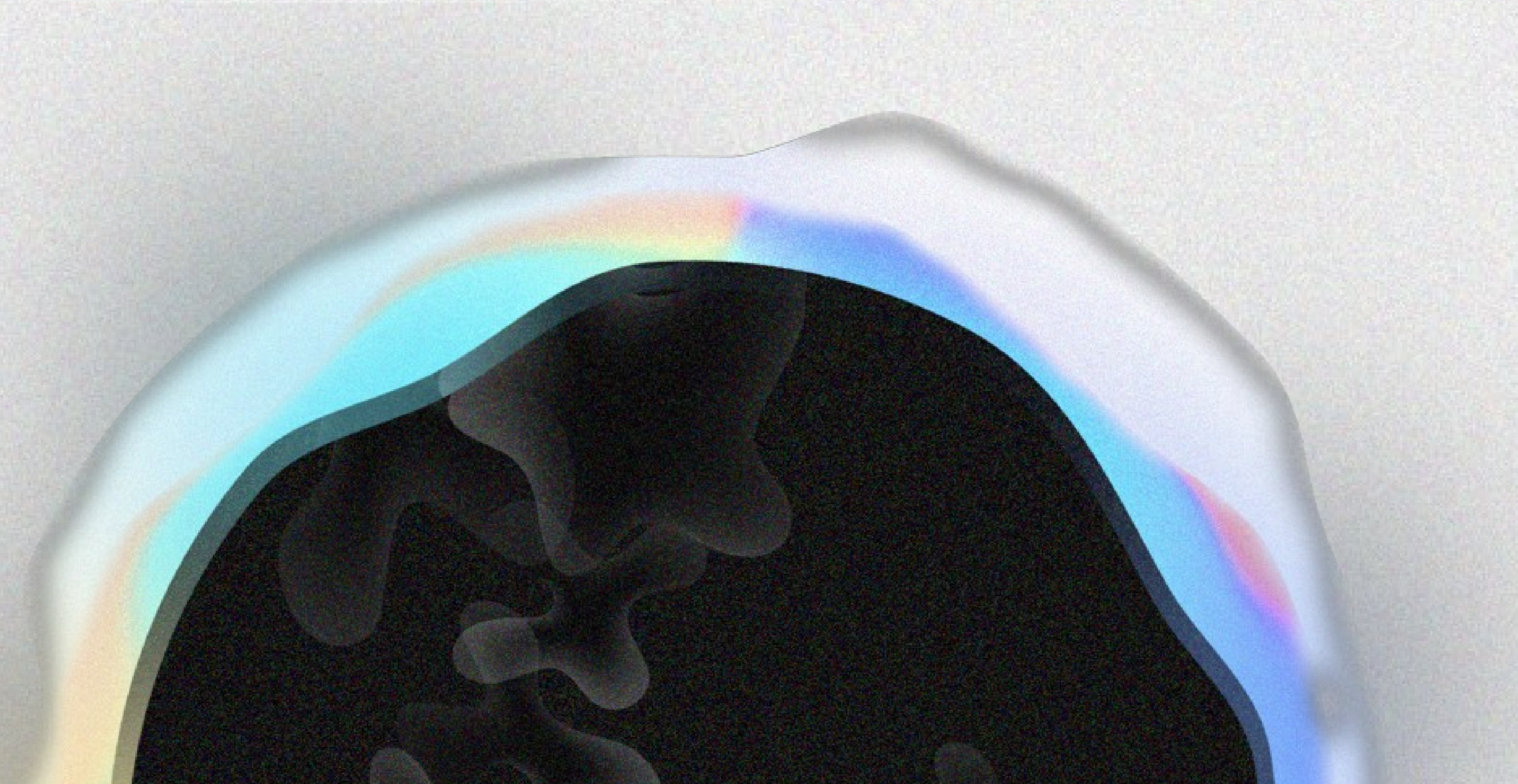
Integrating opencog with neural networks

CogNets

Passing arbitrary python objects between ExecutionOutputLinks

Allows to express computation graph as pytorch expression

Allows to integrate and update ontologies



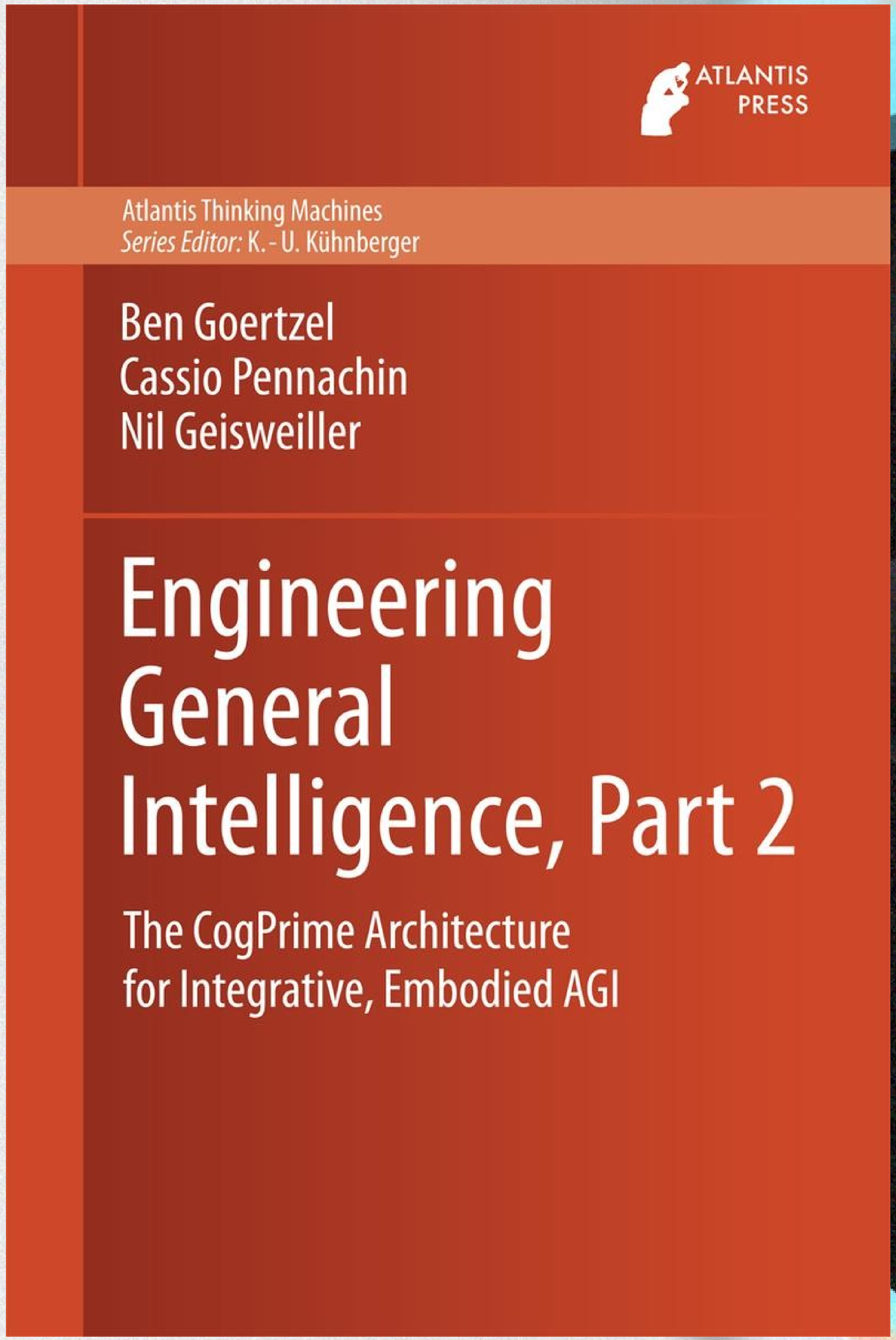
Resources

wiki.opencog.org/

github.com/singnet/atomspace

github.com/singnet/semantic-vision

blog.singularitynet.io



ATLANTIS PRESS

Atlantis Thinking Machines
Series Editor: K.-U. Kühnberger

Ben Goertzel
Cassio Pennachin
Nil Geisweiller

Engineering General Intelligence, Part 2

The CogPrime Architecture for Integrative, Embodied AGI

SingularityNET