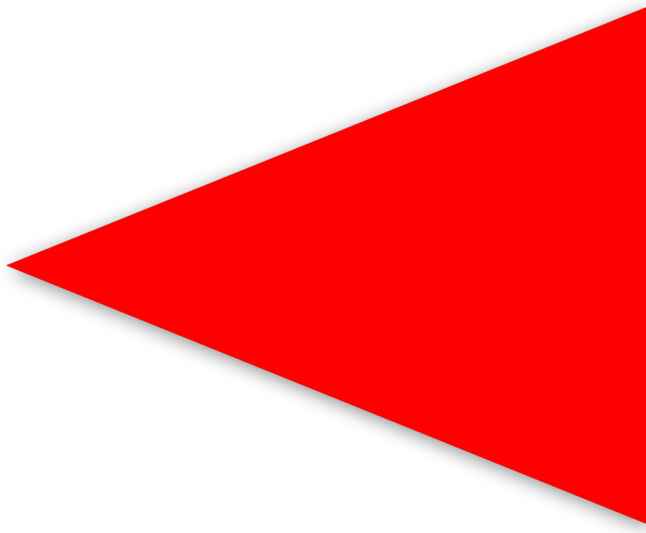
ZeroDiscovery @nd Neuropil



OpenSearchSymposium 2021 / 12.10.2021

Stephan Schwichtenberg

Our Approach

OpenSource CyberSecurity Mesh

www.neuropil.org

Milestones

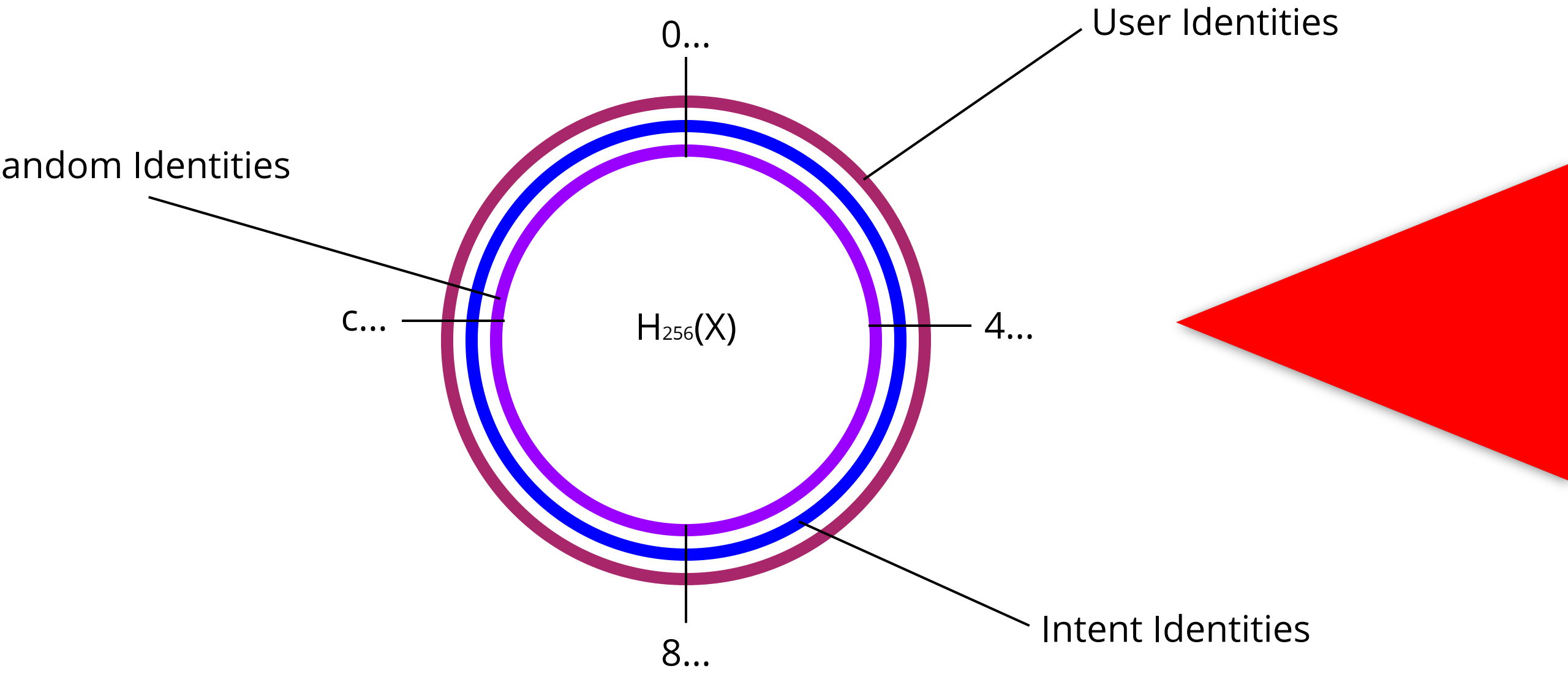
development started in 2014

2016: first exhibition @FROSCON

2019: NGI Zero / EU funded

2021: beta-release HMI 2021

looking for pilots & partners


0...
User Identities
andom Identities
c...
$H_{256}(X)$
4...
8...
Intent Identities

Use Case:

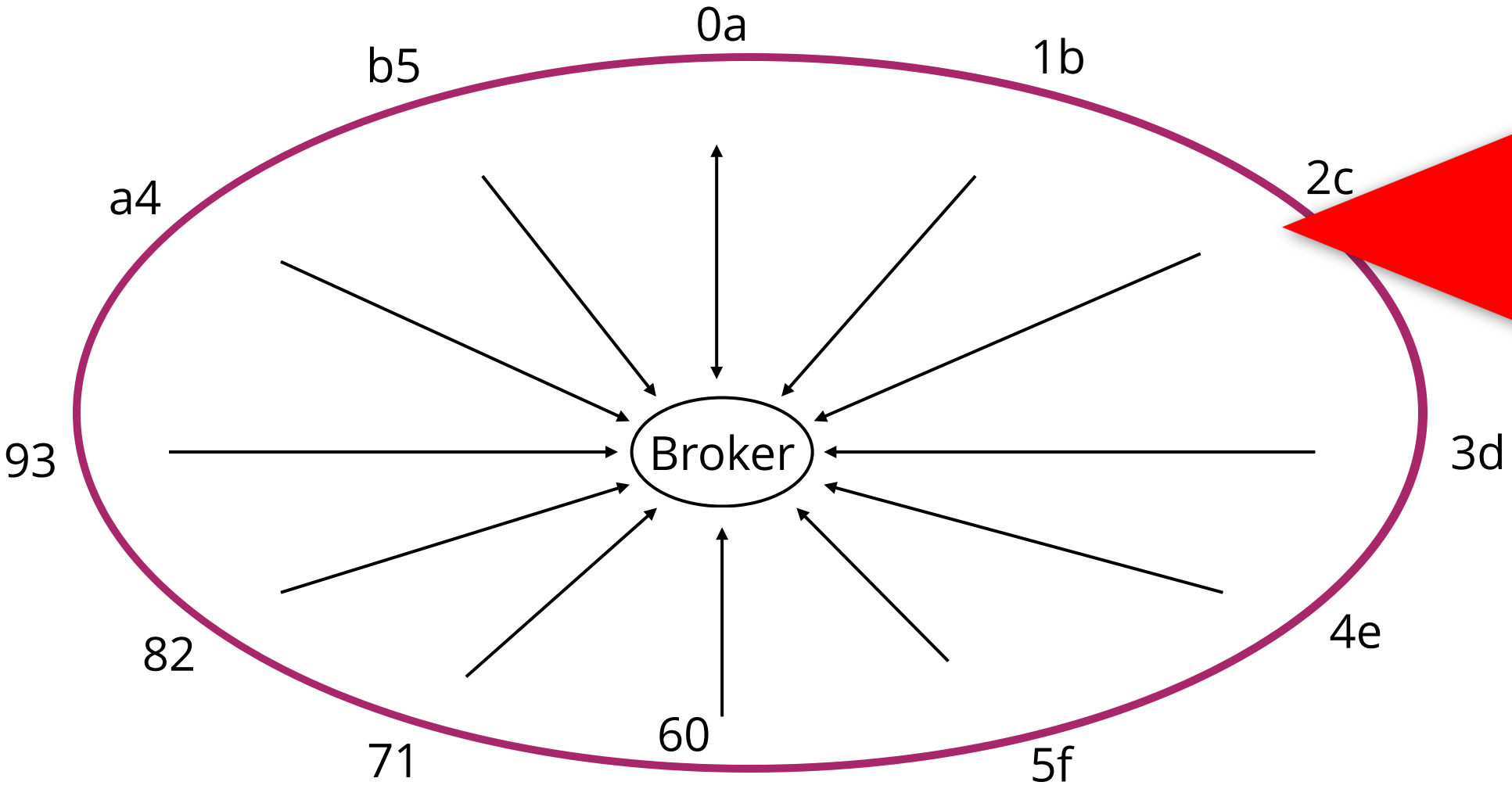
Distributed Search Engine

Neuropil is a project that wants to turn the tables on online search and discovery: instead of search solutions calling the shots, data owners decide what content is publicly searchable in the first place.

They can do this through a new messaging layer that is private and secure by design. Data owners can send cryptographic and unique so-called intent messages that state what specific information can be found where.

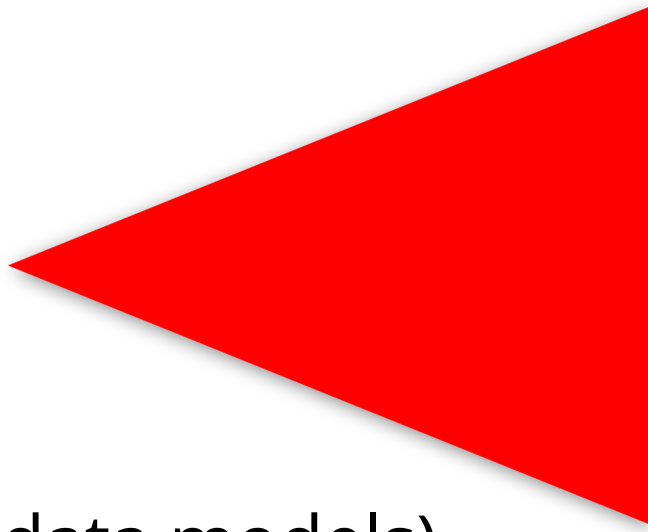
The access to the actual information or content is also controlled by data owners, for instance to provide either paid or public free content.

central broker structures

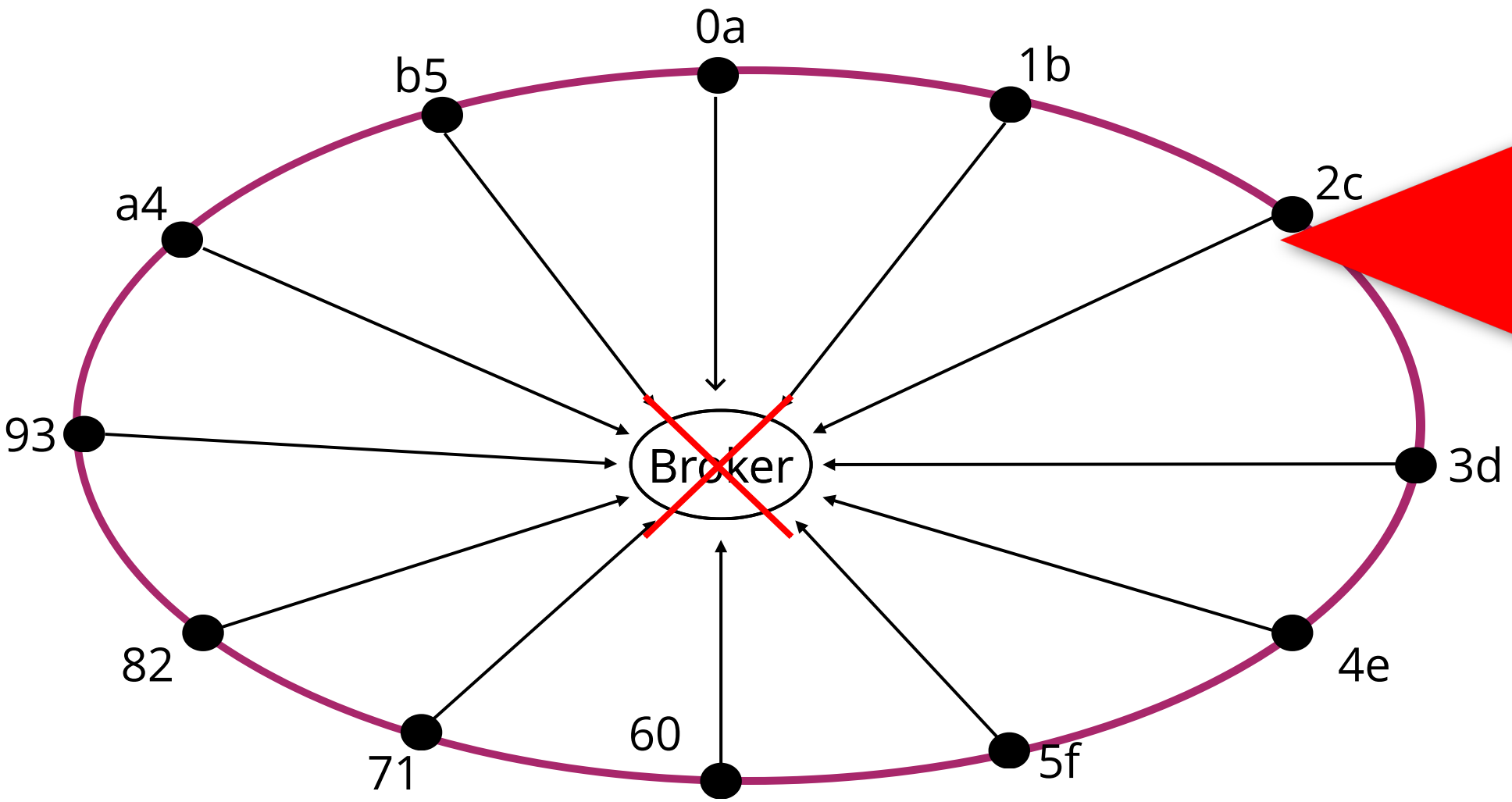


central broker structures

- has to grow with the size of connected instances
- any central broker is attackable
- information is doubled, and possibly outdated
  - crawling is waste of energy
  - legal aspects of copyright / data ownership
- can withhold or change information
- needs to understand many different languages (data models)
- we would like to search for data, not URL's
- who is the broker of all broker?
  - federated broker
  - distributed broker



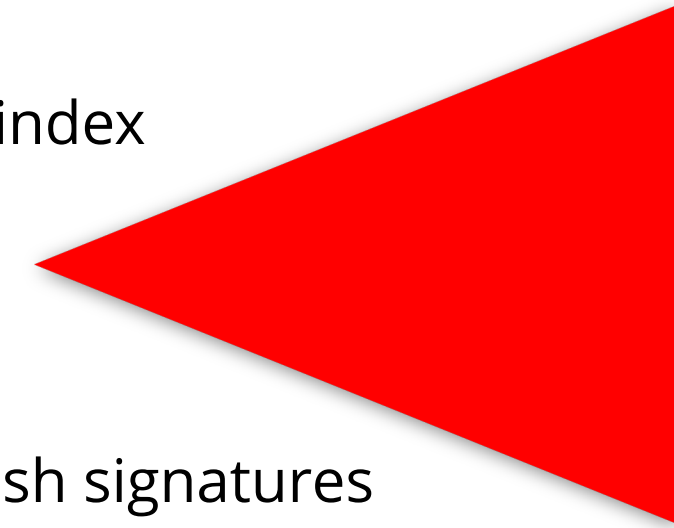
central broker structures ?



Neuropil zero search

initial idea of the NGI Zero project:

- use the virtual address space as a catchword index
- "urn:osf:search:v1" => 0fa6472ba9813c56
- "mydocument.odt" => 65c3189ab2746af0



approach works for single words / URL's / etc.:

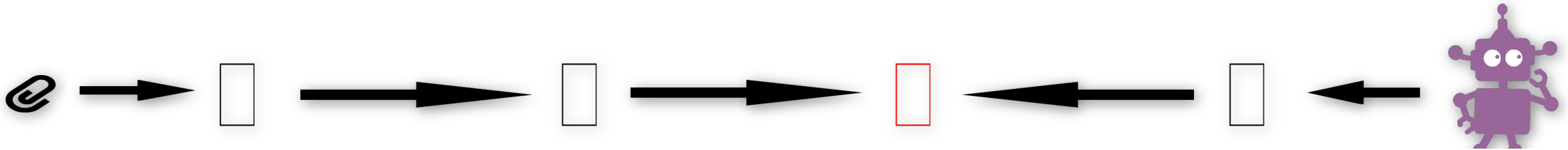
- documents contain more than one word: LSH / minhash signatures
- what about pictures and other data sets (biology / chemistry / ...)?

not every node wants to be part of a specific search index

- need additional subjects to manage search

Neuropil zero search

- what is a good "distributed" index?
  - define "search entry" attributes / data model (JSON/Ontologies)
  - how can we distribute the search entries across a DHT?
- cryptographic longterm key hashing (Schnell et. al.)
  - construct a 256-bit hash value from a vector/dataset (or document)
  - discovery through address space
- minhash signature / frequency mapping
  - use the minhash and its distribution to create a 256-bit hash
  - mapping to address space (hamming distance)



Neuropil zero search

minhash signatures:

- split text into shingles / ngrams, hash each
- min/maxhash (more efficient / less MSE / higher BAR)
- seed the minhash with cryptographic hash
- variable size possible / but has to be mod(8)
- data-dependant minhash signatures
  - fixed size, variable shingle size
  - variable size, fixed shingling

Neuropil zero search

CLKHash - Cryptographic Longterm Keys:

- is basically a bloom filter
- standardized set of identifieres (tbd for "search")
- candidate for a search entry
- natural fit with intent token / pheromone
  - pheromone is able to capture time information
  - intent token contains secured public data
- still need to find the correct clustering

Neuropil zero search

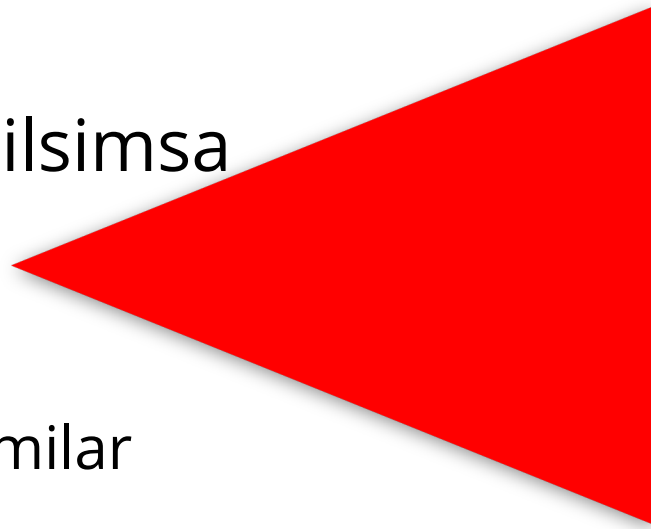
LSH - Locality Sensitive Hashing (based on minhash):

- split mmh into n-rows and b-bands
- efficiently reduce the amount of comparison
- lots of variants: TreeLSH, BoundedLSH, EnsembleLSH, ...
- but:
  - designed for target threshold (1/b)^(1/r)
  - works on a fixed set of hash tables
  - use a variable length hash

Neuropil zero search

LPH - Locality Preserving Hashing:

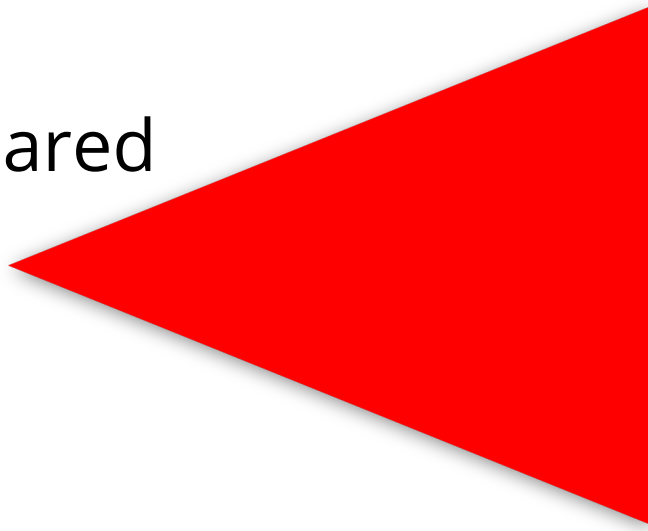
- used spam/malware detection: ssdeep / nilsimsa
  - low false positive rate / robust against attacks
- used in forensics: tlsh
  - comparing which part of two documents are similar
  - resulting hash based on threshold (median)
- data dependant hash calculation
- variable length hash



Neuropil zero search

using LSH and LPH together - 256bit hash value

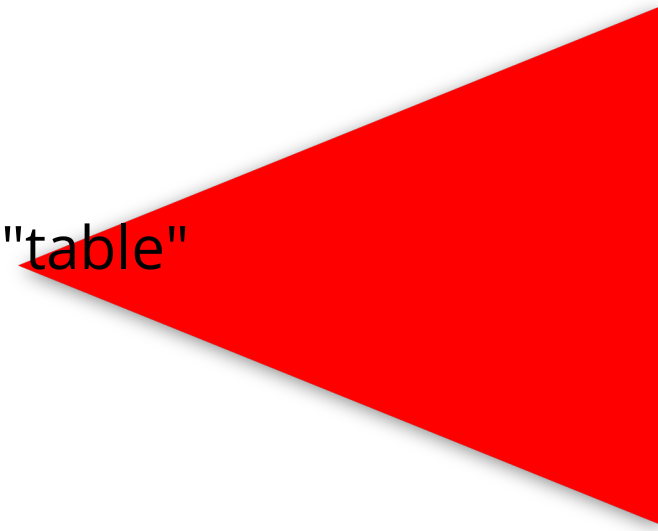
- relative importance of virtual tables can be compared
  - locally the full hamming distance is used
  - distribution is based in partial hamming distance
  - is a kind of multi-index
- easy to calculate, easy to distribute
  - using octile values (3bits per octile / assuming 85 hash tables)
  - uses a bktree implementation including binning (neighbour table seach)
  - on table hit, CLKHash'es are compared
  - can be extended with additional tables / np_index



Neuropil zero search

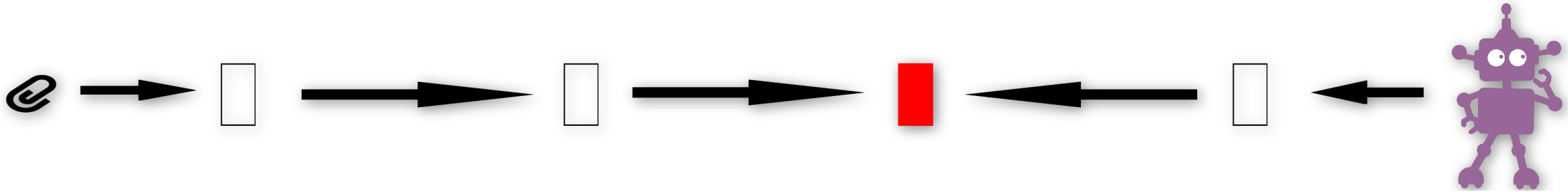
can LSH and LPH work together - 256bit hash value

- will it work with in fully distributed mode?
  - hash distance routing guarantees query of closest "table"
  - nodes can detect the required hamming distance
  - storage in multiple search nodes is guaranteed
- until now: validated locally with 600k entries
- reasonsable performance (running un-optimized code)
  - parallel execution of queries
  - code optimization (cache misses / network runtime / ...)
  - using an embedded database for bf (bitmap) comparisons (?)



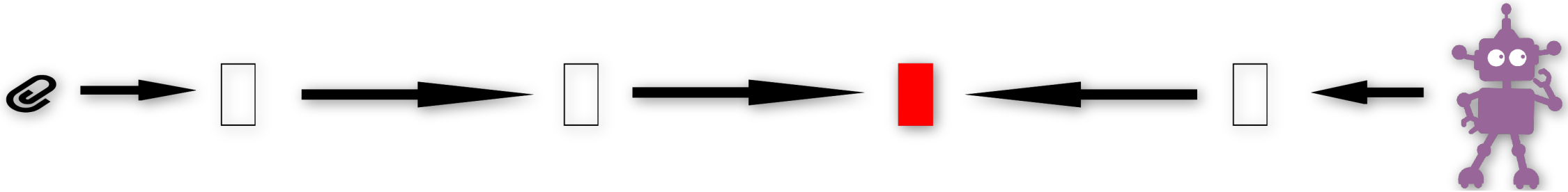
Neuropil zero search

- what is a good "distributed" index?
  - defined "search entry" data model
    - intent (CWT) token of data owner
    - claims to be used as attributes (extend e.g. for HTML ...)
    - CLKHash to represent the actual data set
    - can be used "in private": add PPAttributes (minhash)
  - defined the distributed search index for a DHT
    - NPIndex: relativ importance based on our search entry
    - 256bit data dependant hash value

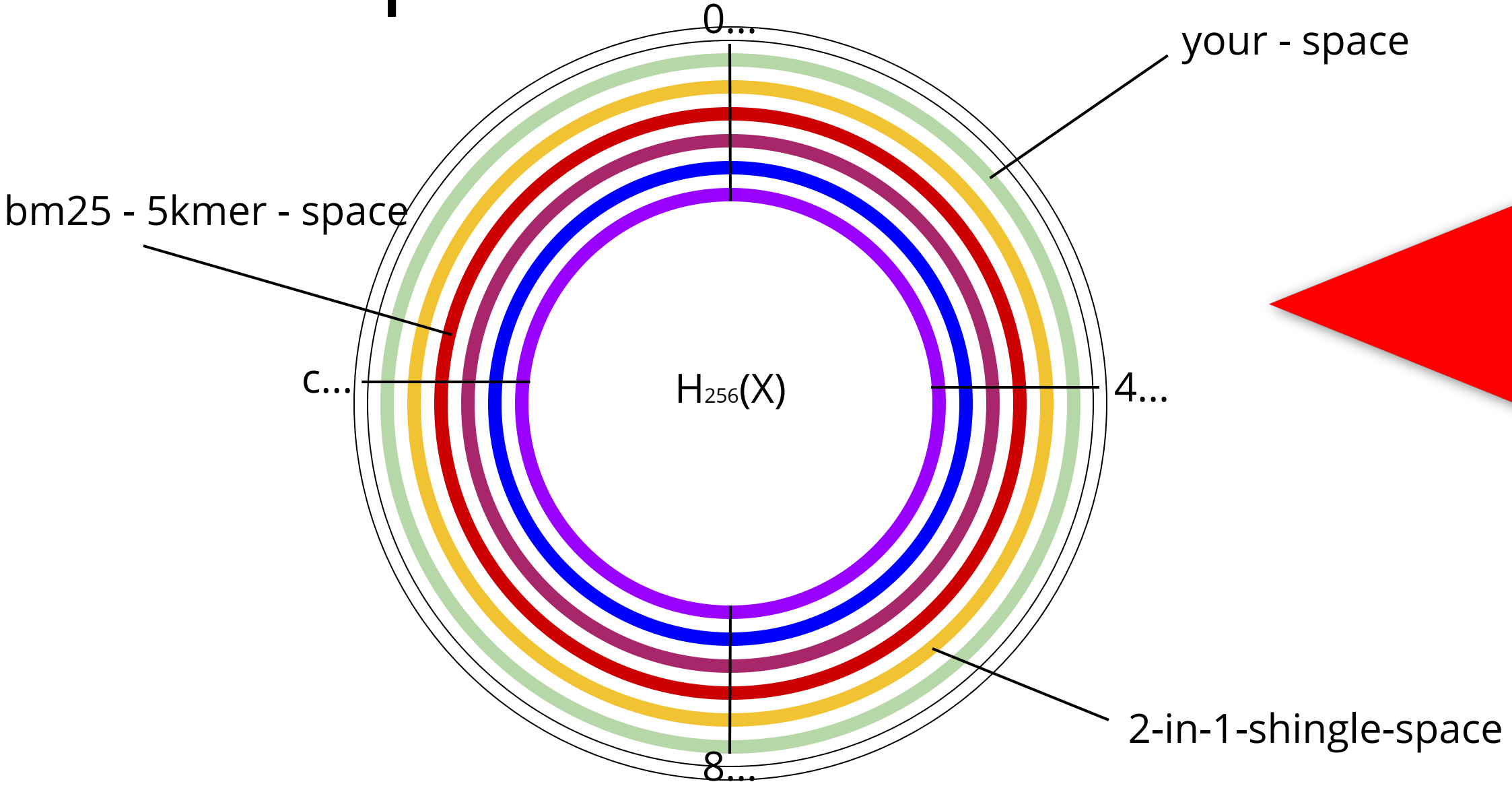


Neuropil zero search

- what is a good "distributed" index?
  - search entries will "disappear"
    - encoded time information will enable "forgetting"
    - automatically evicts malicious content
    - ensure actuality of information
  - mutual exchange of interest
    - the searcher retrieves a list of possible data sources
    - the content provider retrives a list of searchers
    - no man-in-the-middle to prevent exchange

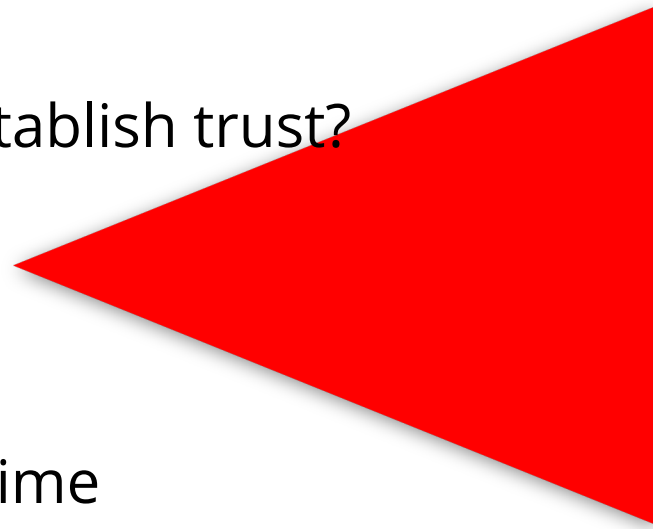


Neuropil zero search



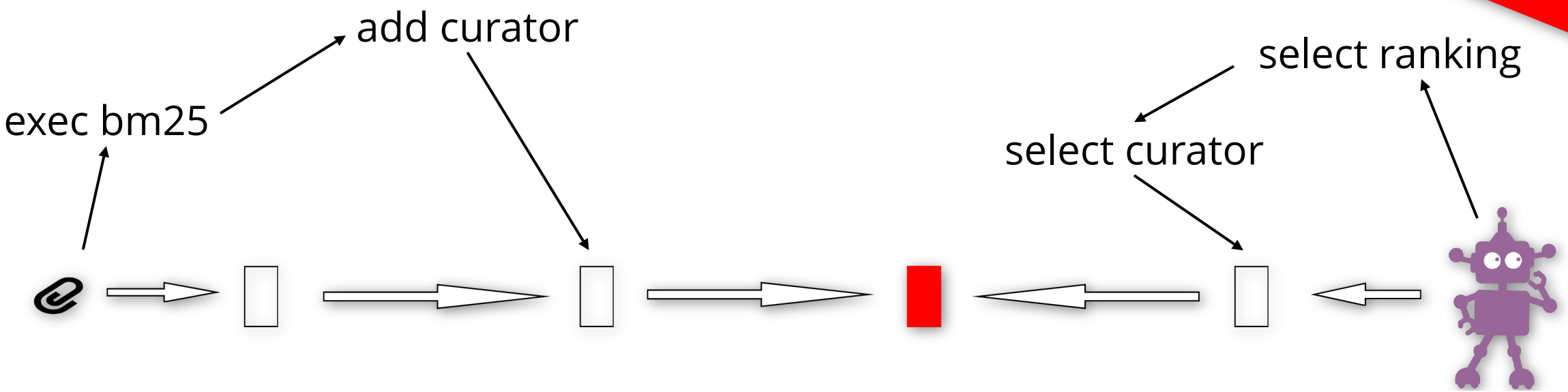
Neuropil zero search

- Open Question: Identities
  - if every participant can add search entry, how to establish trust?
  - additional "curator" signature for SEO companies
  - PKI / web of trust unsuited, but TSA is an answer
- Open Question: Time
  - all systems must have the same understanding of time
  - index is attenuated, entries will disappear after a time
- Open Question: Runtime
  - Python Binding (full support), Lua / NodeJS (partial) ...
  - WASM to execute user supplied map/reduce code?



Neuropil Zero Search

- Open Questions: Search Pipeline Collaboration
  - good understanding of the whole process is needed
  - adding a search entry (BM25 / TF-IDF / ...)
  - querying a search entry



Join Our Workshops!

pi-lar GmbH

Kreuzgasse 2-4

50667 Köln

www.pi-lar.net

info@pi-lar.net

eliza@neuropil.org

www.neuropil.org

https://www.gitlab.com/pi-lar/neuropil

neuropil@pi-lar.net