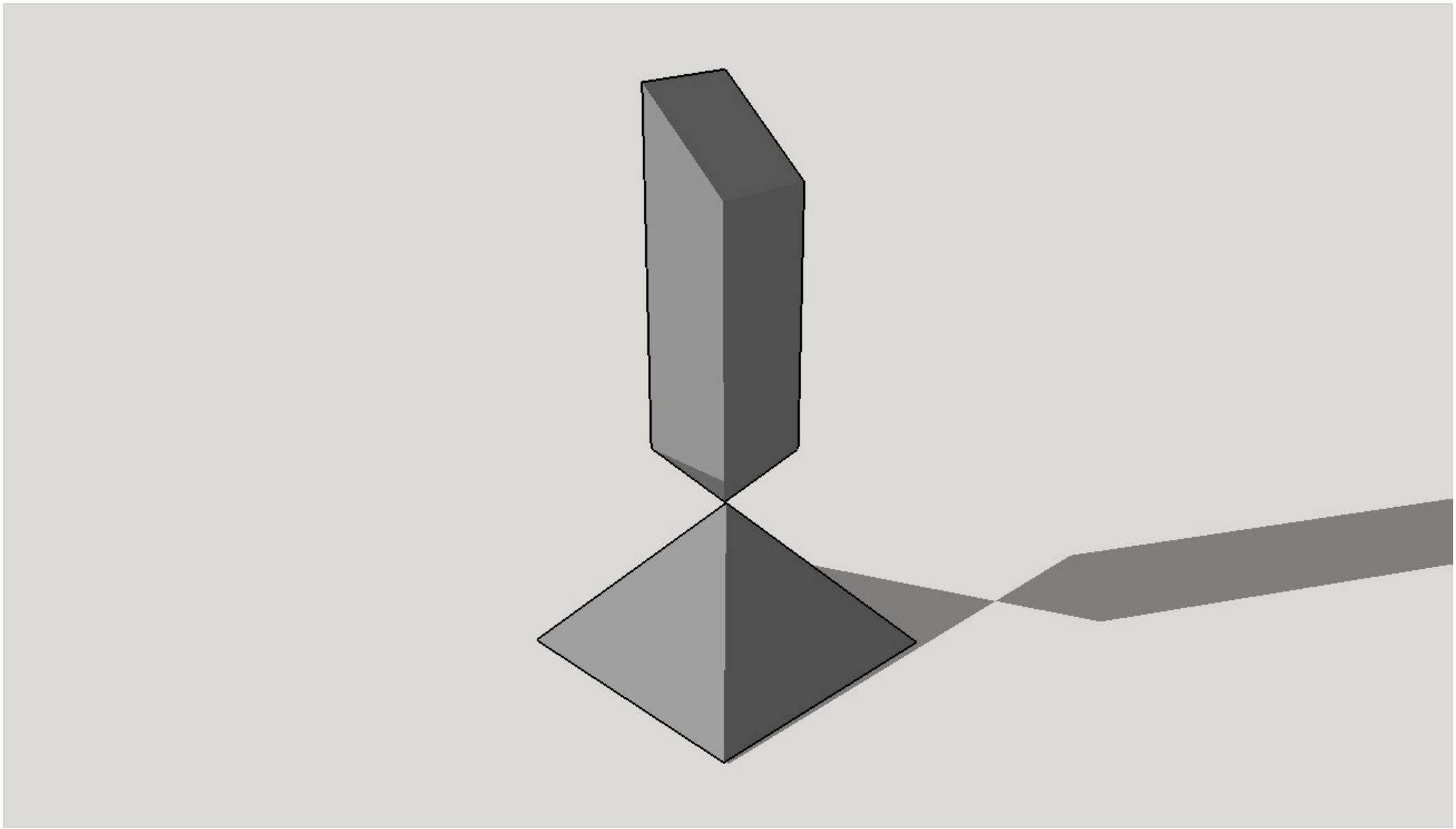
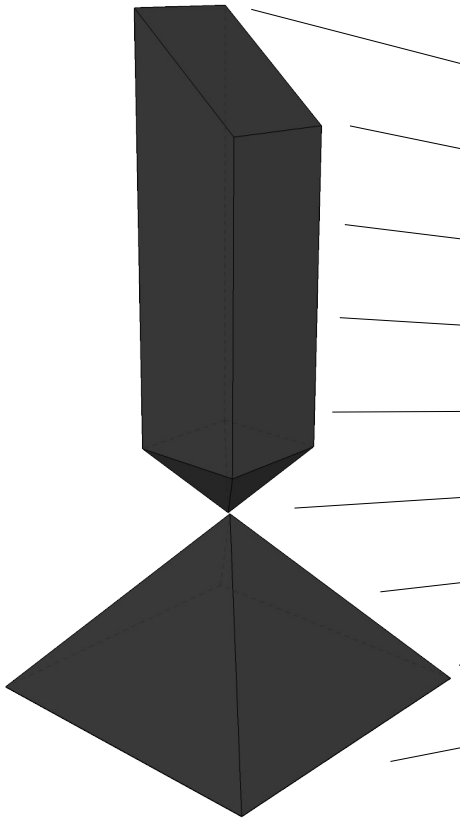
TiiQu – Trust Quotient



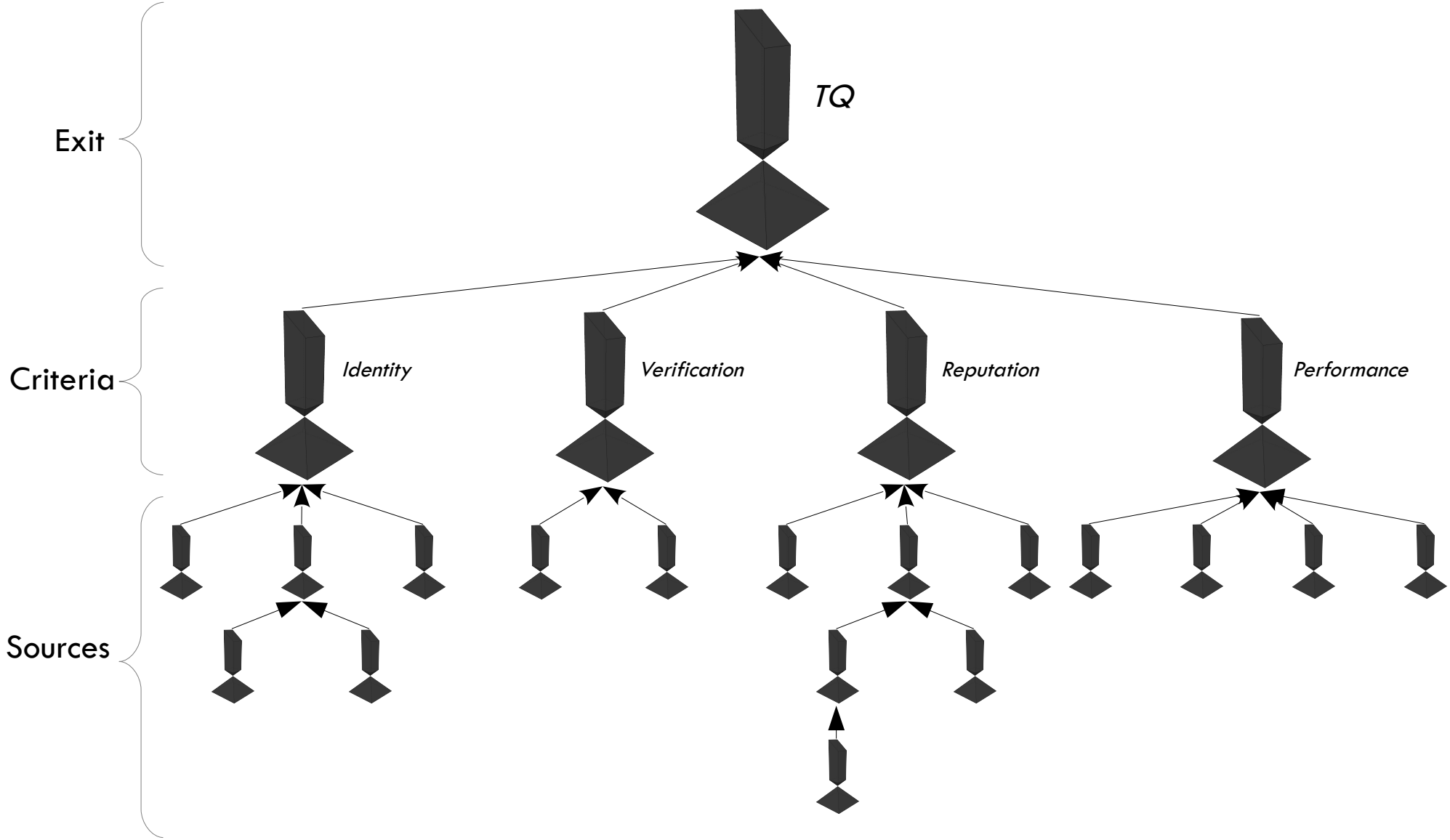
The TQ Node

Data Structure



- node_id - (uint)
- name - (string)
- parent_id – (uint) **nullable**
- value – (int) **nullable**
- value_range_from – (int)
- value_range_to – (int)
- weight – (int) **1 – 100, nullable, default(100)**
- samples – (int) **nullable**
- source – (string) **nullable**

A TQ Typical Tree



TQ Node / Tree Details

All nodes in a TQ Tree are structurally identical giving the overall TQ network several advantages:

- *A TQ score can be calculated not only for individuals but also for groups of individuals by chaining trees.*
- *A TQ tree can be appended to theoretically infinite levels.*
- *TQ nodes are agnostic to their related nodes and so can be connected in multiple configurations*

Not all TQ Trees will be structured the same, most will likely have common elements:

- *A single exit*
- *A set of criteria with the exit as parent*
- *A set of sources with a criteria or another source as parent*

Nodes recursively seek all descendants when calculating their output value.

A node's output value is the product of the weighted average of its children and its value, if set.

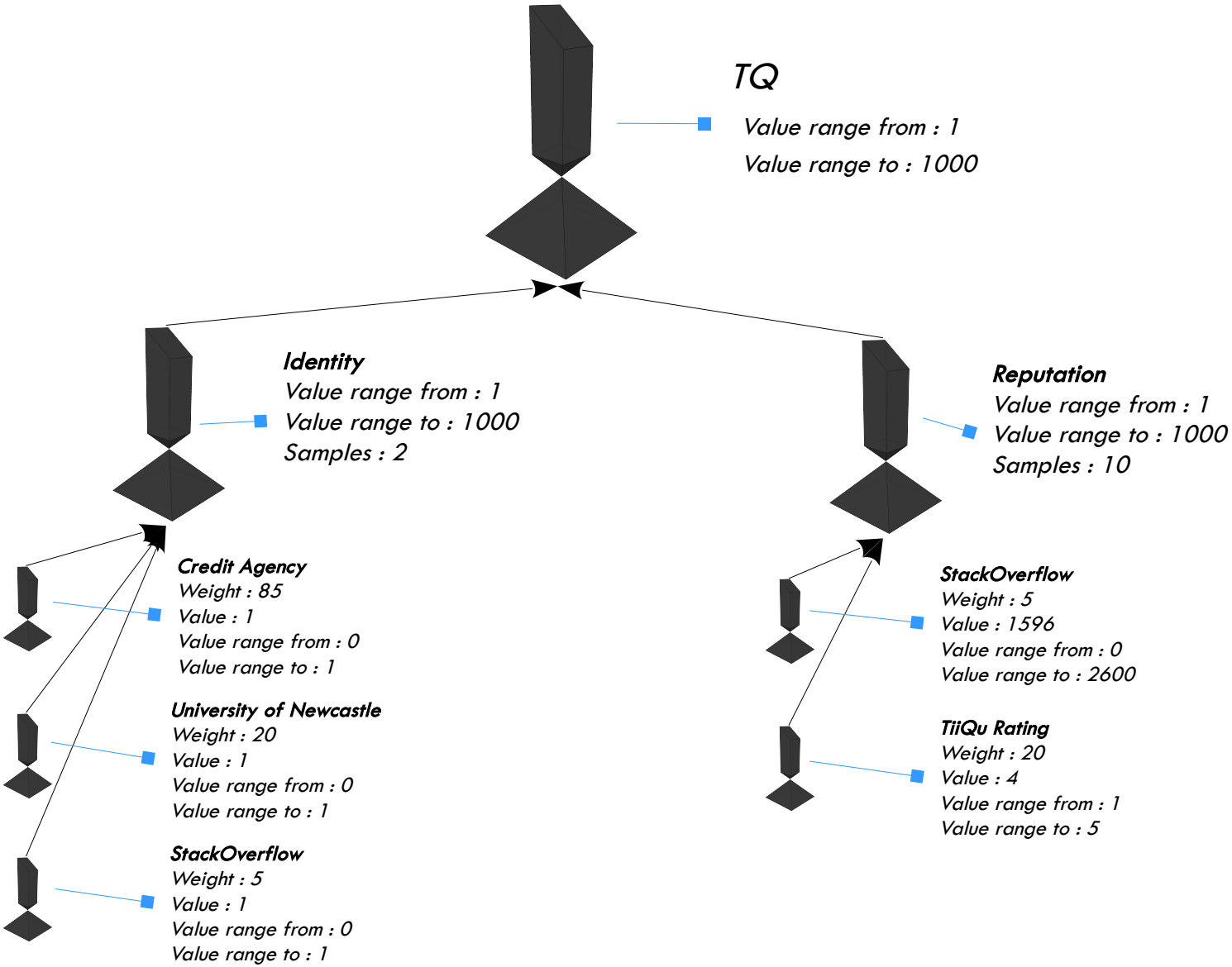
Reputation based sources with an unlimited range use the median value as the upper range, meaning values will calculated as a deviation from the median.

A weight is the most impact a value can have on its parent,

Samples set the minimum number of child nodes a node must have before weighting ratio is normalised

Weight value, because total weights may exceed 100

TQ Calculation Example



TQ Calculation Example

Identity – sample : 2
All children values
[
[85, 1, 0, 1],
[20, 1, 0, 1],
[5, 1, 0, 1]
]

Reputation – sample : 10
All children values
[
[5, 1596, 0, 2600],
[20, 4, 1, 5]
]

Weighting Ratio (wr) = Number of children (c) / Sample (s)
c >= s | s = null → Wr = 1
0.2 = 2 / 10

Weight Value (wv) = Total weights / 100 . Minimum value 1
0.25 = (20 + 5) / 100
∵ 0.25 < 1 → wv = 1

Units (ru) in range
ru = vrt – vrf + 1
5 = 5 – 1 + 1

Percentage (P) the value against the range units
P = (v / ru)
0.8 = (4 / 5)

Weighted Percentage
wP = P * w * wr / wv
3.2 = 0.8 * 20 * 0.2 / 1

∴

Identity – sample : 2
wr = 1
wv = 1.1

Reputation – sample : 10
wr = 0.2
wv = 1

Apply Percentage value calc against range and value
Identity
[
[85, 1],
[20, 1],
[5, 1]
]

Reputation
[
[5, 0.614],
[20, 0.8]
]

Apply Weighted Percentage calc against Percentage
Identity
[
[77.27],
[18.18],
[4.55]
]

Reputation
[
[0.614],
[3.2]
]

Sum the results
Identity : 100
Reputation : 3.814

Apply summed results to node range:
ru / 100 = range ratio
v * ratio = applied ratio
applied ratio + vrf = applied value

1000 / 100 = 10
3.814 * 10 = 38.14
38.14 + 1 = 39.14

Final values
Identity: 1000
Reputation : 39.14

Apply the full process to the exit node
TQ -
[
[100, 1000, 1, 1000],
[100, 39.14, 1, 1000]
]

TQ -
[
[100, 500],
[100, 19.57]
]

Final result
TQ = 519.57

Potential Problems

- Represent more sources bonus calculation, because 100 5/5 is better than 10 5/5.

  Solution is a separate subnode that is a sibling of the score node. Example structure: `TQ` > `Reputation` > `TiiQu Rating` > `Score` & `Count`
  Any node that needs to value quantity of sources can then split the

- Check that a super high deviation from mean score doesn't break the whole system. Ie stackoverflow of 3000000 against a median of 2600 with a weight of 5 = 1153.85 against a score that should be under 100