



Introduction

Background

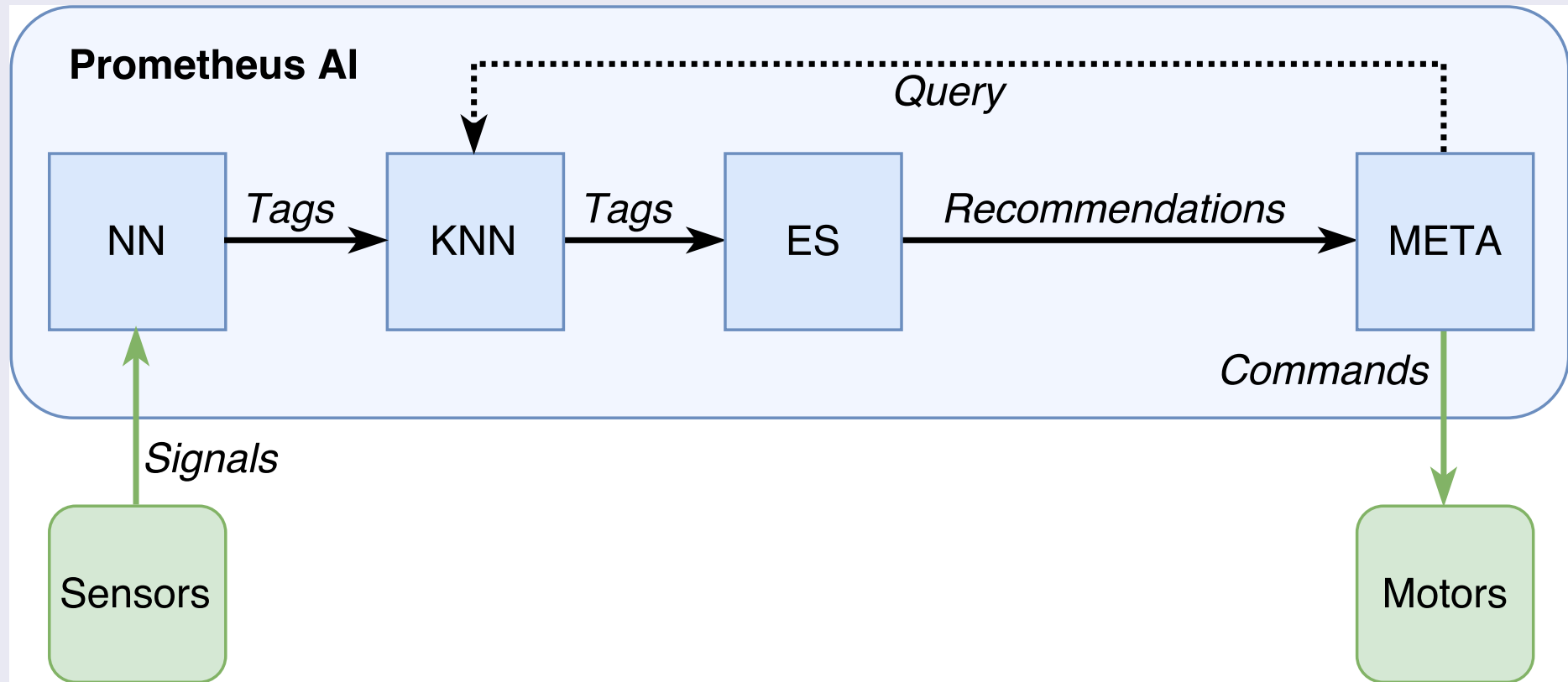


Figure 1: Prometheus AI model with labeled input and output.

Figure 2: Thinking forwards in the KNN.

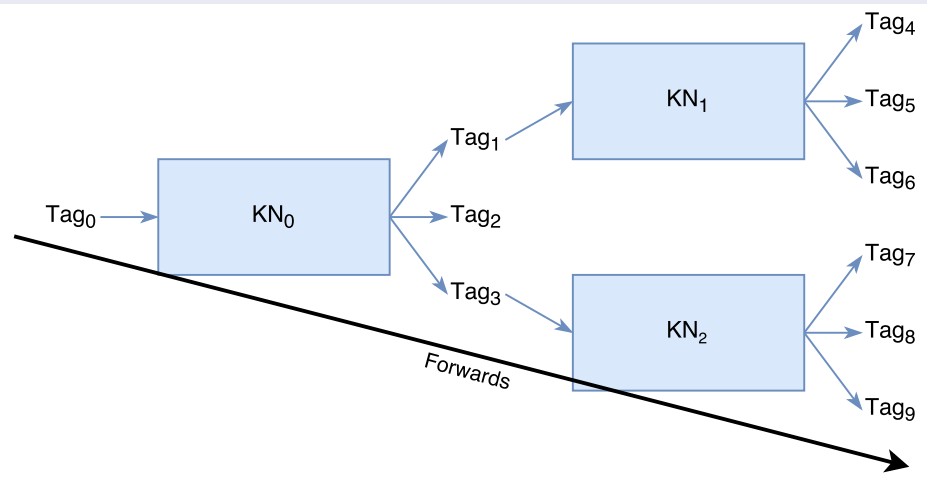


Figure 3: Thinking forwards in the KNN.

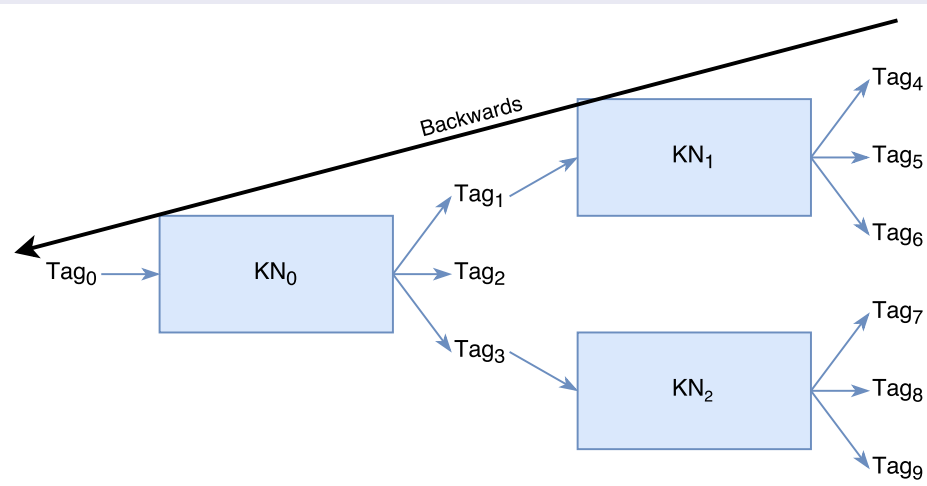


Figure 4: Thinking backwards in the KNN.

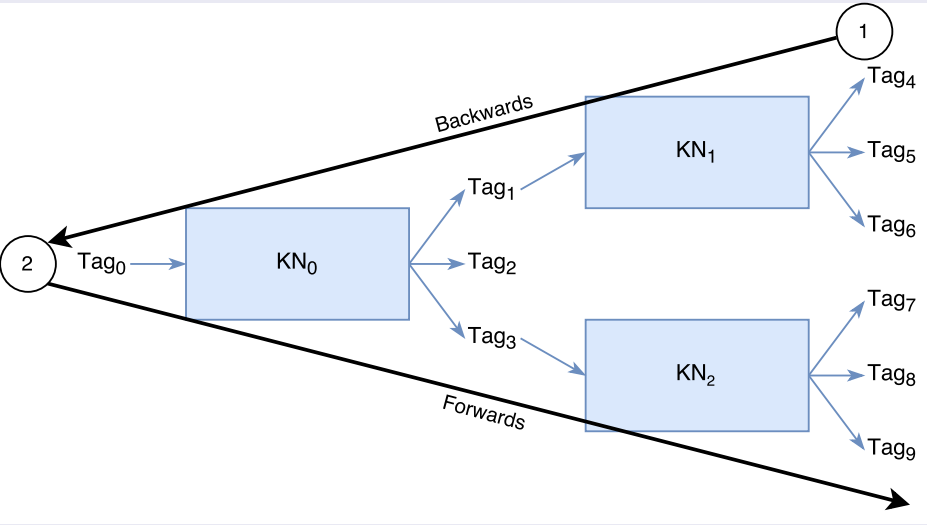


Figure 5: Lambda thinking in the KNN.

Problem

Design

Implementation

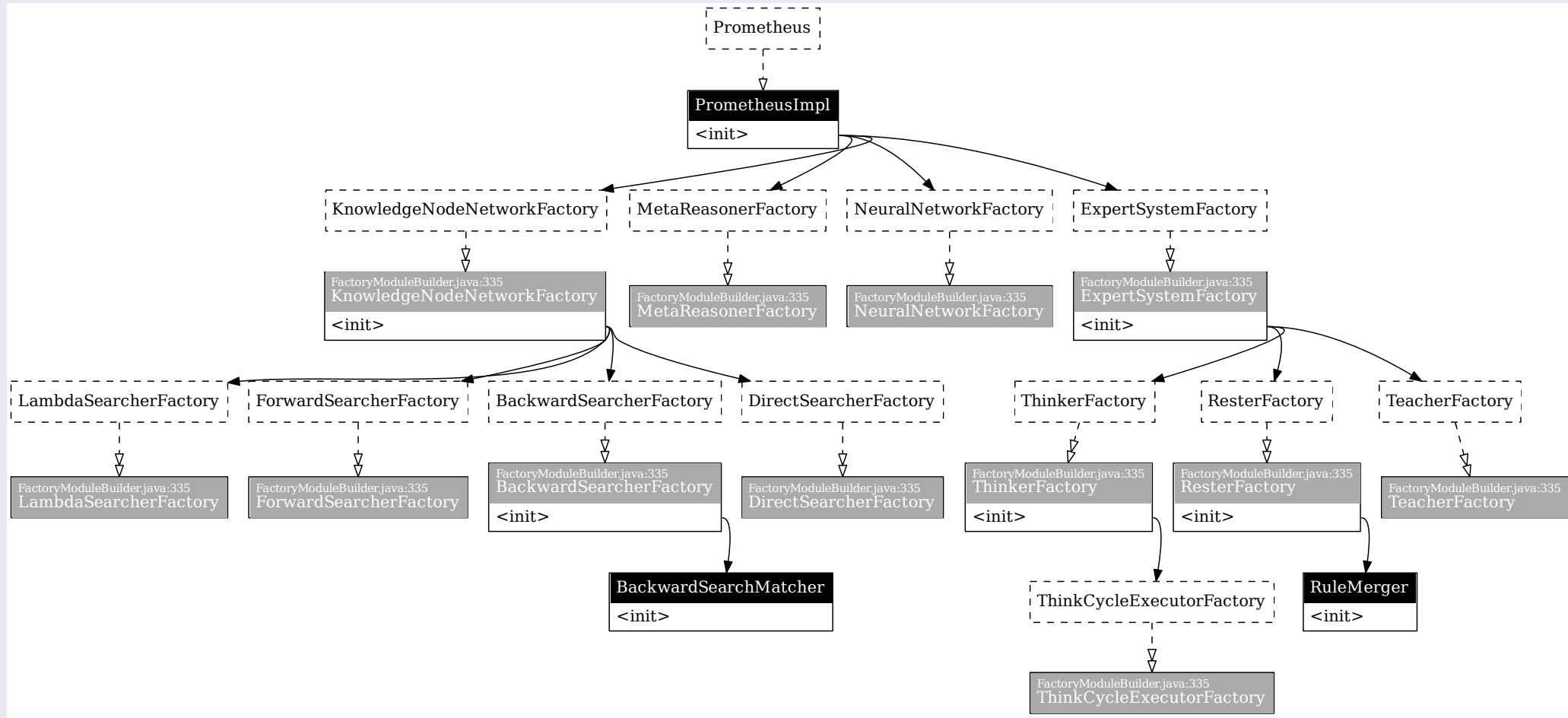


Figure 6: Guice dependency graph.

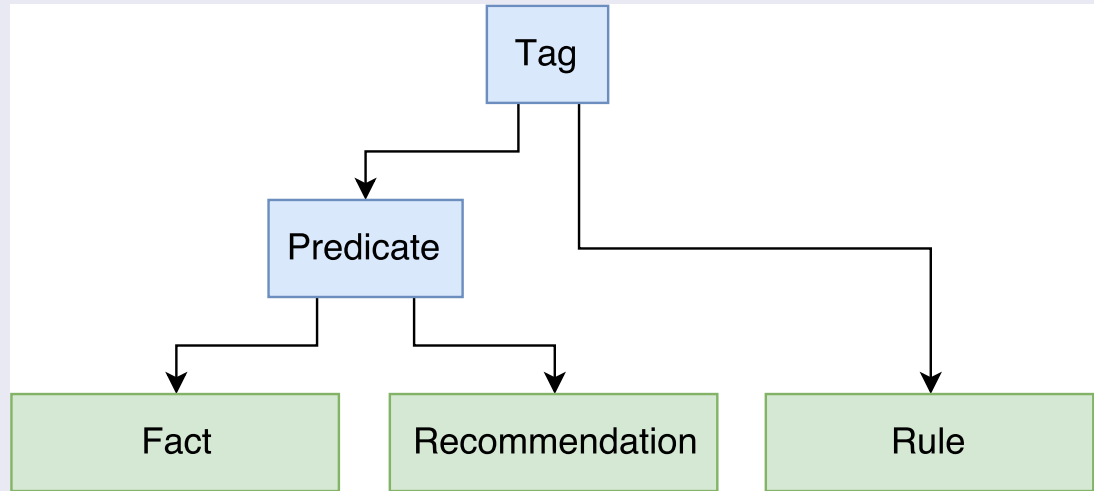
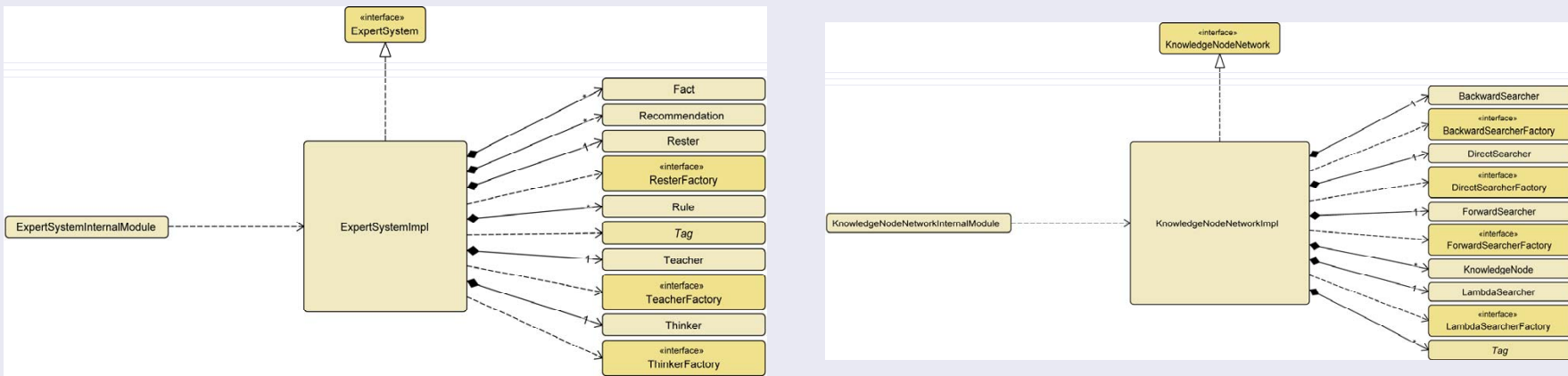


Figure 7: Tag inheritance graph.



(a) ES package.

(b) KNN package.

Figure 8: UML diagrams for the major modules.

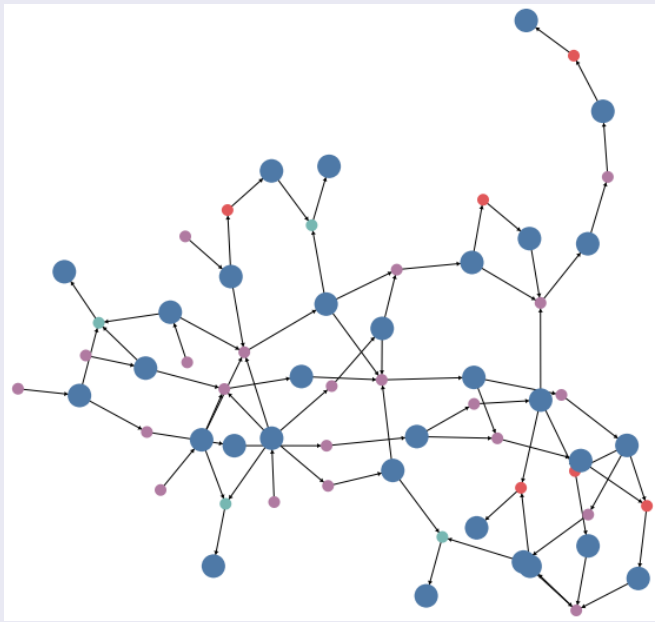
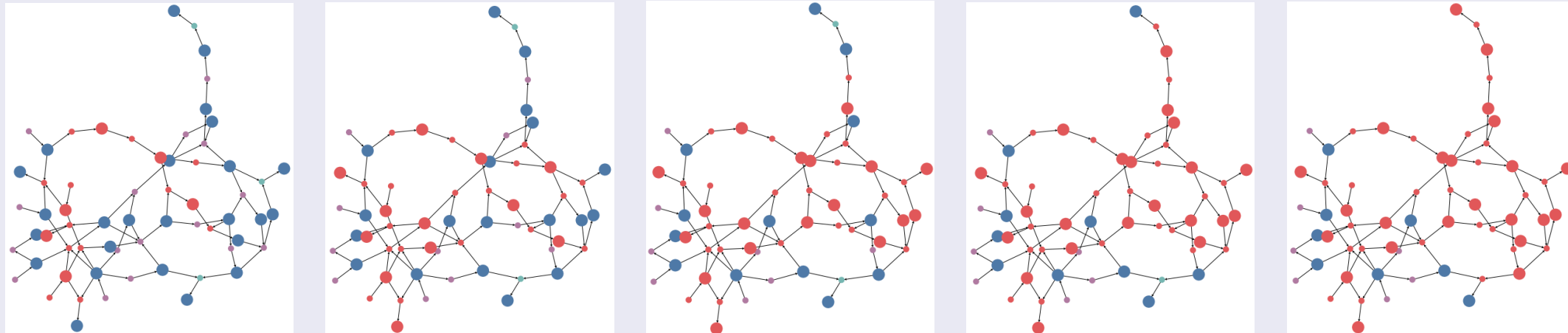


Figure 9: Visualization of the KNN. Small nodes are Tags, big nodes are Knowledge Nodes (KNs). Red nodes represent active Tags or fired KNs. For non-active Tags, Facts are purple, Rules are blue and Recommendations are orange.



(a) Ply 1.

(b) Ply 2.

(c) Ply 3.

(d) Ply 4.

(e) Ply 5.

Figure 10: Forward thinking visualization in the KNN.

Results & Tests

- Unit tests
- Integration tests
- TestNG and TravisCI

State	Ready Rules	Active Rules	Active Facts	Active Recommendations
Initial	$(A)(B) \rightarrow (D)$ $(D)(B) \rightarrow (E)$ $(D)(E) \rightarrow (F)$ $(G)(A) \rightarrow (H)$ $(E)(F) \rightarrow (\#Z)$		$(A), (B)$	$(\#X), (\#Y)$
⋮	⋮	⋮	⋮	⋮
Final	$(G)(A) \rightarrow (H)$	$(A)(B) \rightarrow (D)$ $(D)(B) \rightarrow (E)$ $(D)(E) \rightarrow (F)$ $(E)(F) \rightarrow (\#Z)$	$(A), (B),$ $(D), (E)$ $(F)$	$(\#X), (\#Y), (\#Z)$

Figure 12: Simple ES test setup.

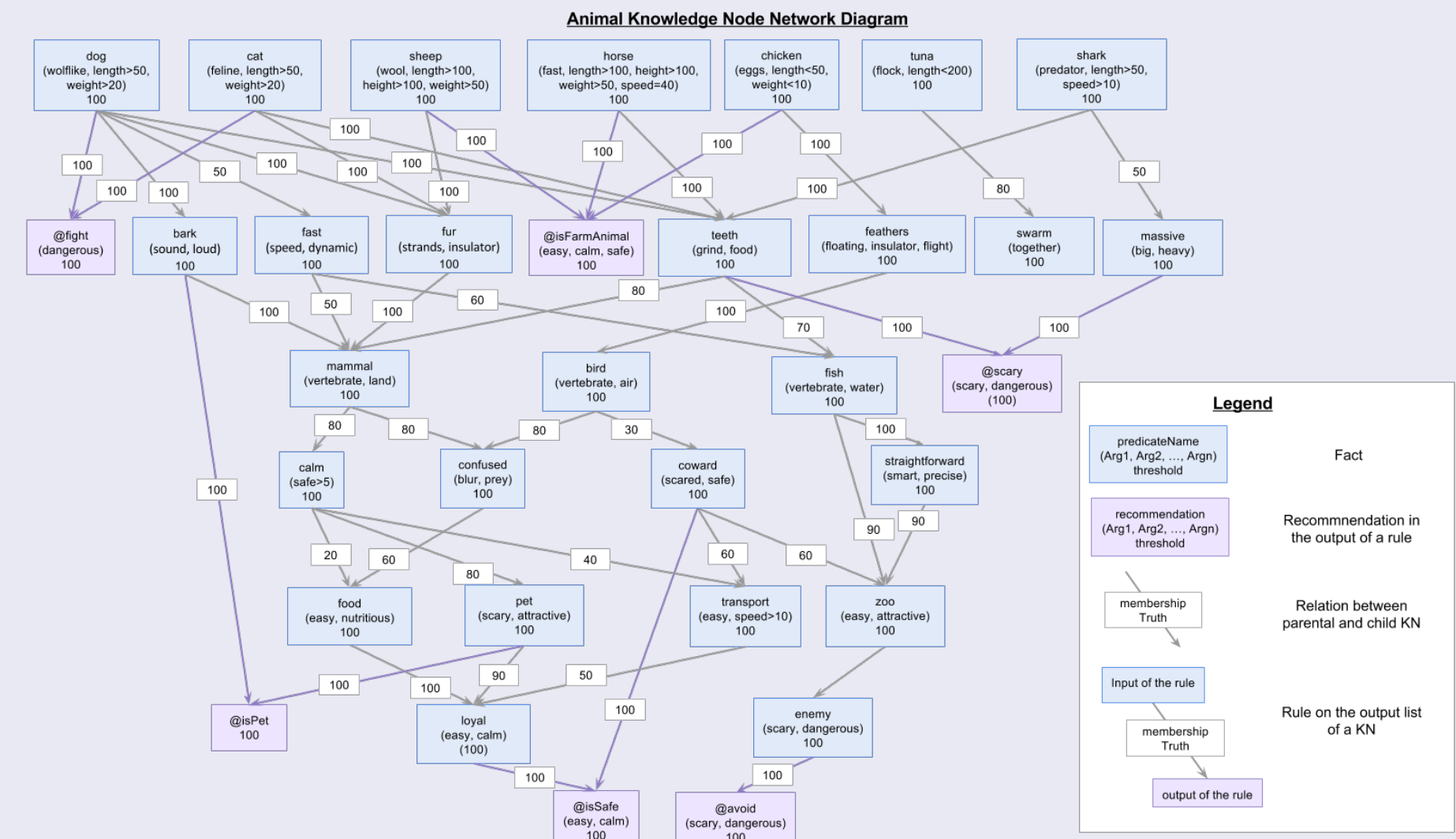


Figure 13: Elaborate test KNN network.

Module	Time
prometheus-ai	2.07 s
BackwardSearcherTest	559 ms
BackwardSearchMatcherTest	2 ms
DirectSearcherTest	0 ms
ExpertSystemImplTest	194 ms
ExpertSystemTest	199 ms
ForwardSearcherTest	17 ms
KnnAndEsTest	28 ms
KnowledgeNodeNetworkAnimalTest	62 ms
KnowledgeNodeNetworkImplTest	114 ms
KnowledgeNodeNetworkPetTest	6 ms
KnowledgeNodeNetworkTest	3 ms
LambdaSearcherTest	1 ms
ResterTest	14 ms
RuleMergerTest	2 ms

Figure 14: Test results for various modules.

Conclusion