practical decision tree implementations

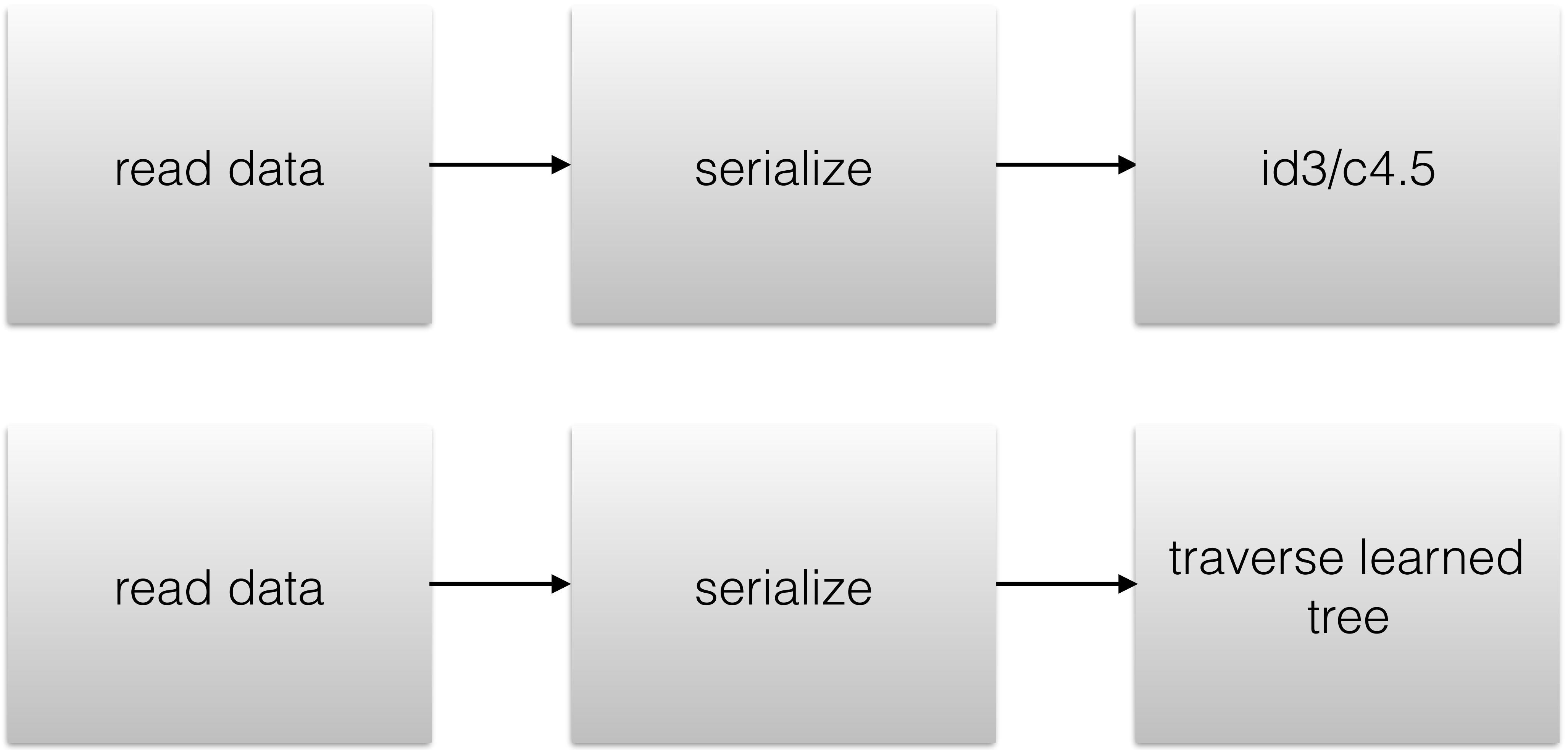
repository

- https://github.com/madmax88/decision-tree
- Works with SBCL — will *not* work with other implementations

obstacles to production applications

- implementation
    - i.e. difficulty in implementing correctly and efficiently
- accountability
    - i.e. *knowing* what is being deployed

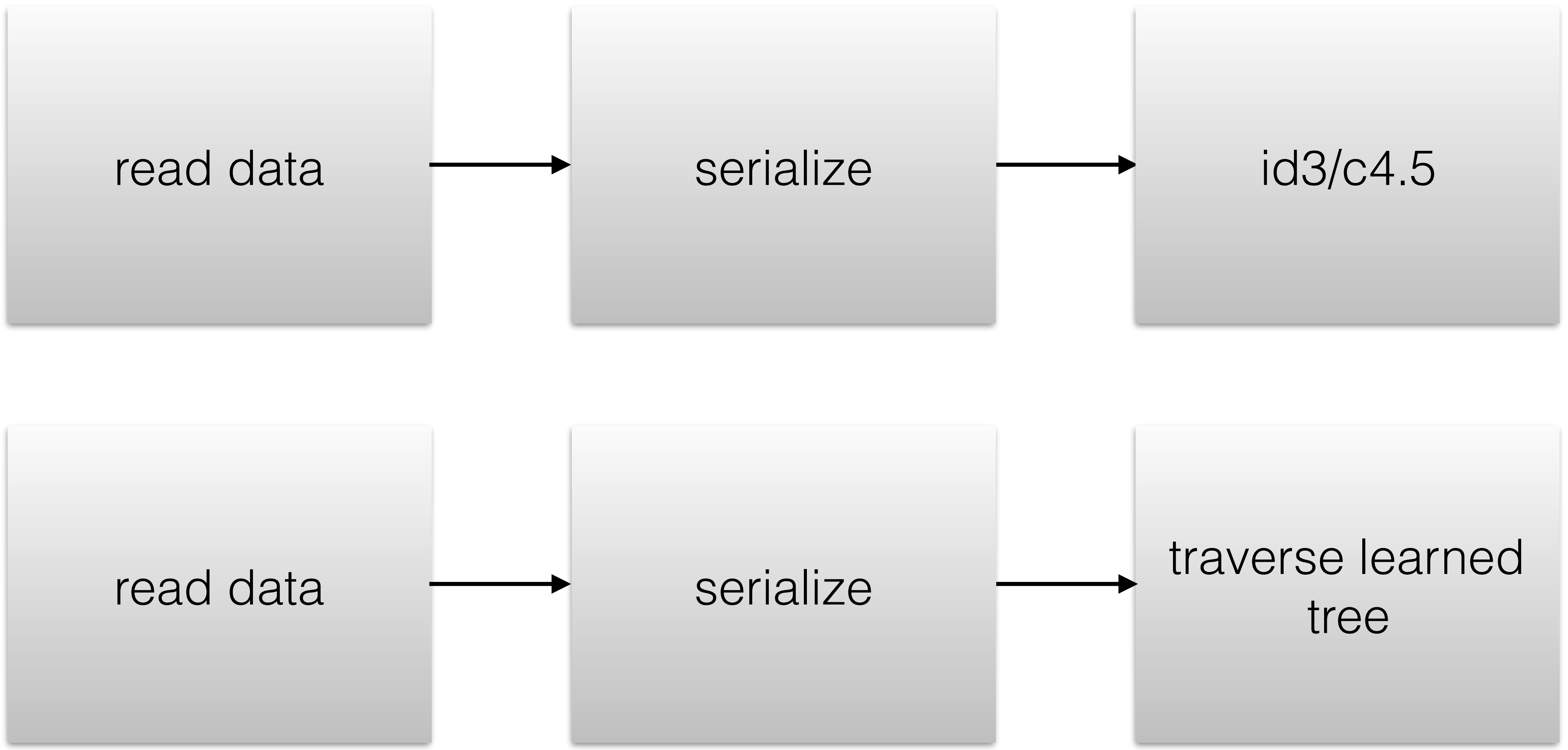
problems with implementations



the problem

- we are building primitives (read, serialize) that already *exist*
- the decision tree itself is basically a function; i.e. DecisionTree :: A -> Boolean
  - to actually use the decision tree, we need to write a evaluator

lisp implementation



- lisp allows users to manipulate its abstract syntax tree
    - exposed as common data structures
- reader is exposed and can be programmatically modified
    - i.e. a arbitrary data format like JSON can be read and serialized by the language
- functions are first-class objects -> we can use the language to evaluate formed decision trees

strategy

- apply id3 to training data
  - create test(s) for the best attribute and write it as a part of the AST
  - create a function whose body is the new AST

further reading

- Paul Graham's *ANSI Common Lisp* and *Let Over Lambda*
- *The Structure and Interpretation of Computer Programs — free*
- *Practical Common Lisp — free* http://www.gigamonkeys.com/book/