

Today's Plan

Abstract data types

- Bags
- Lists
- Stacks
- Queues

Java

- Generics
- Inheritance
- Polymorphism

Object Oriented Paradigm

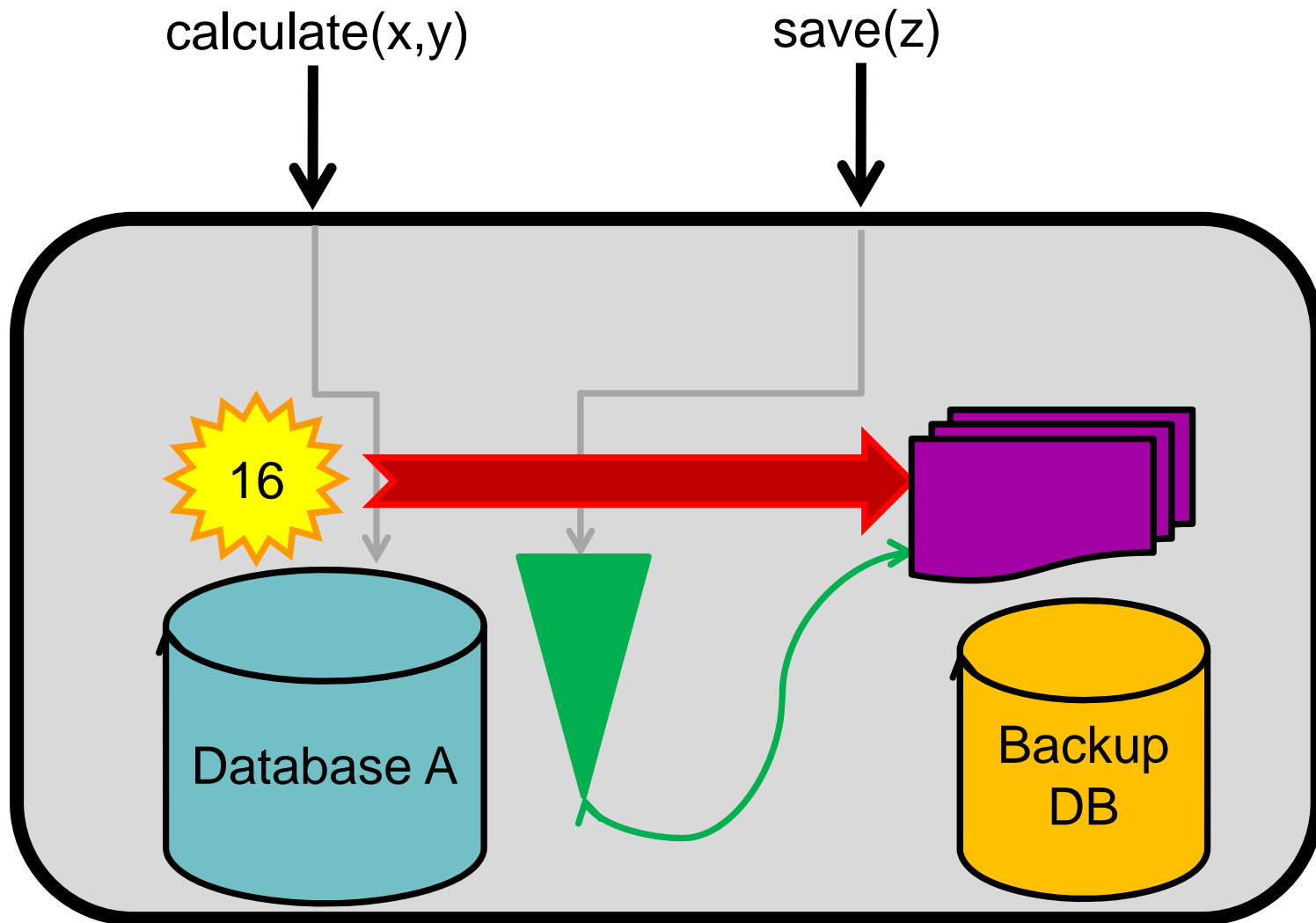
Encapsulation

Inheritance

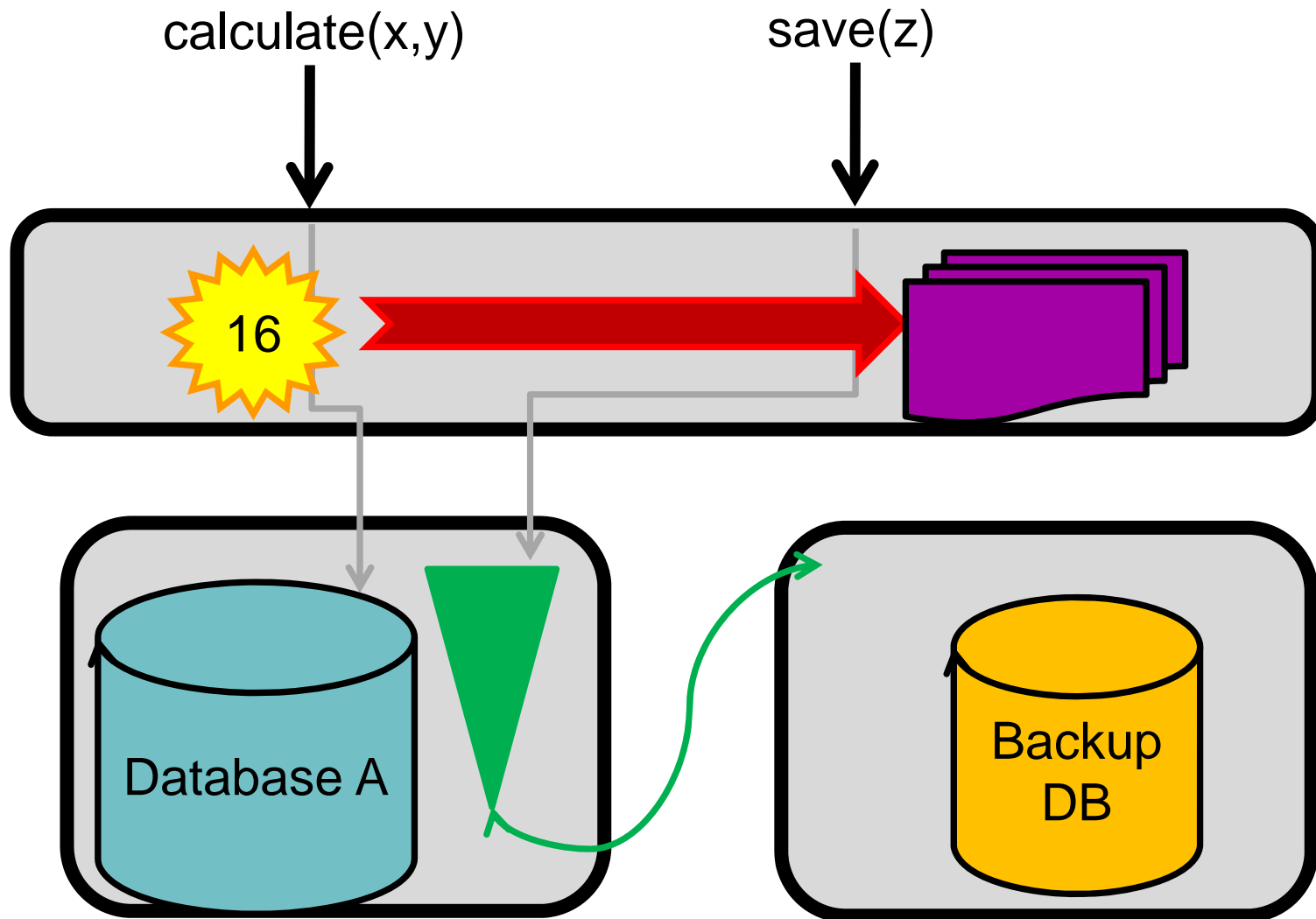
Polymorphism

Abstraction

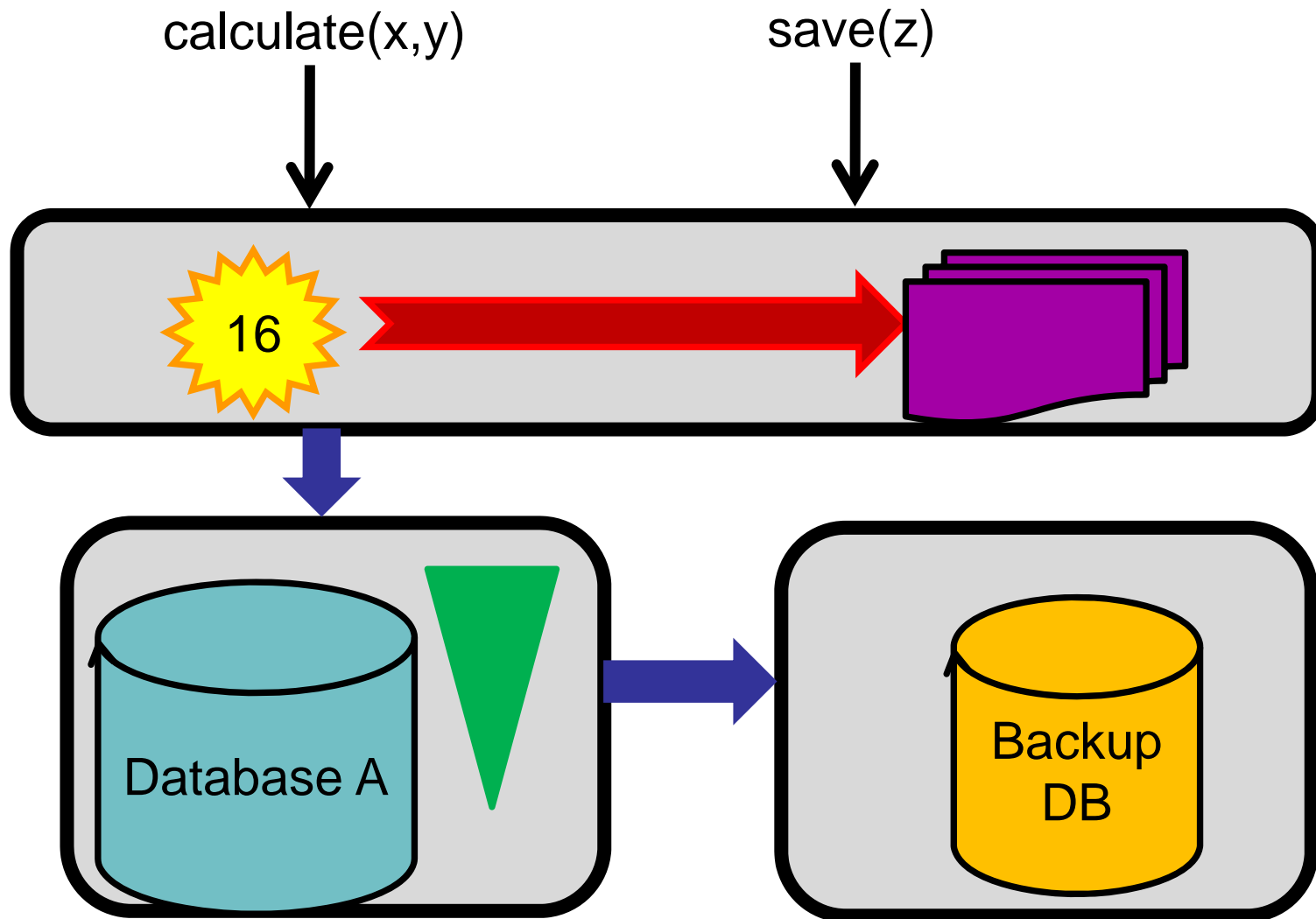
Abstraction



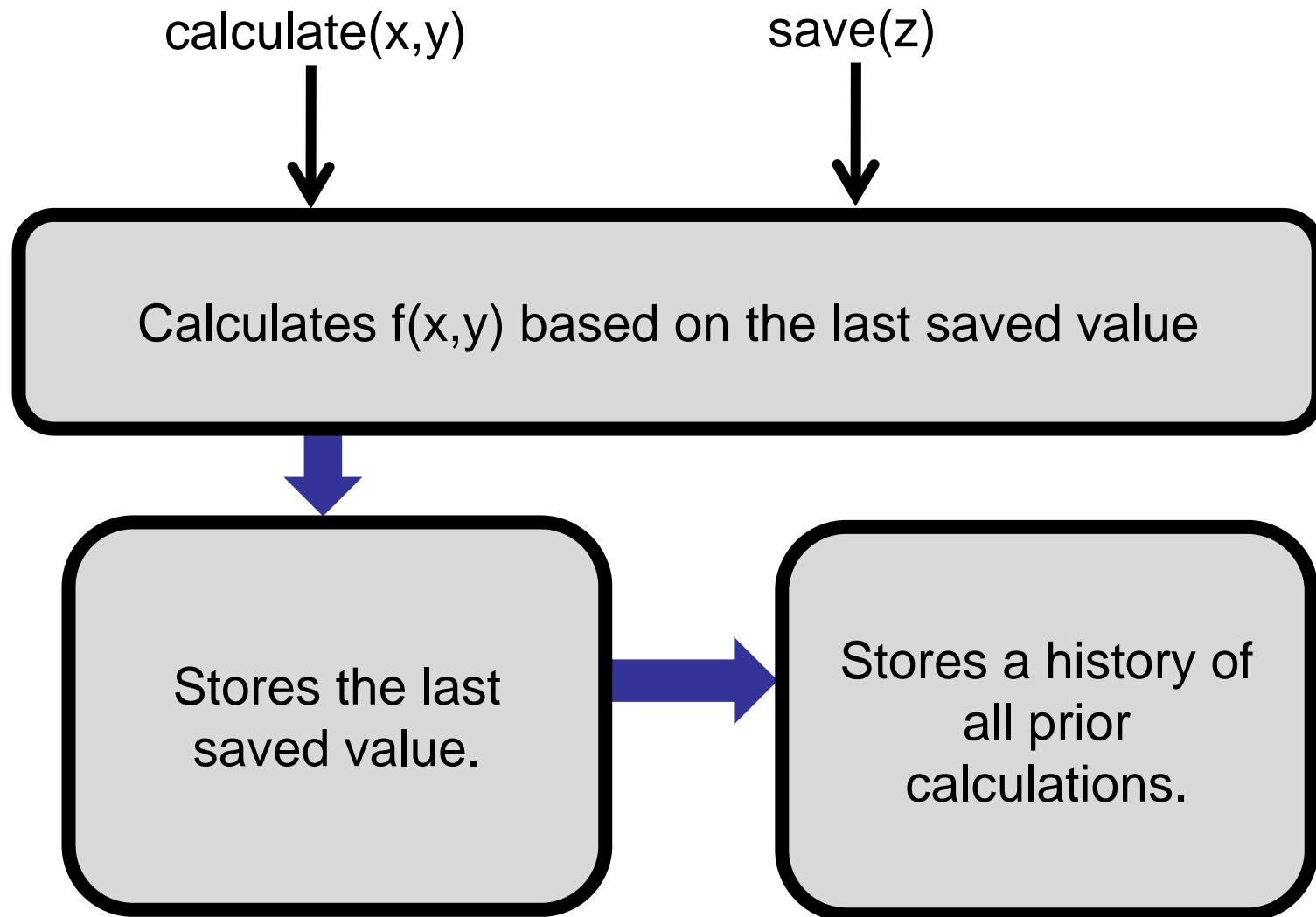
Abstraction



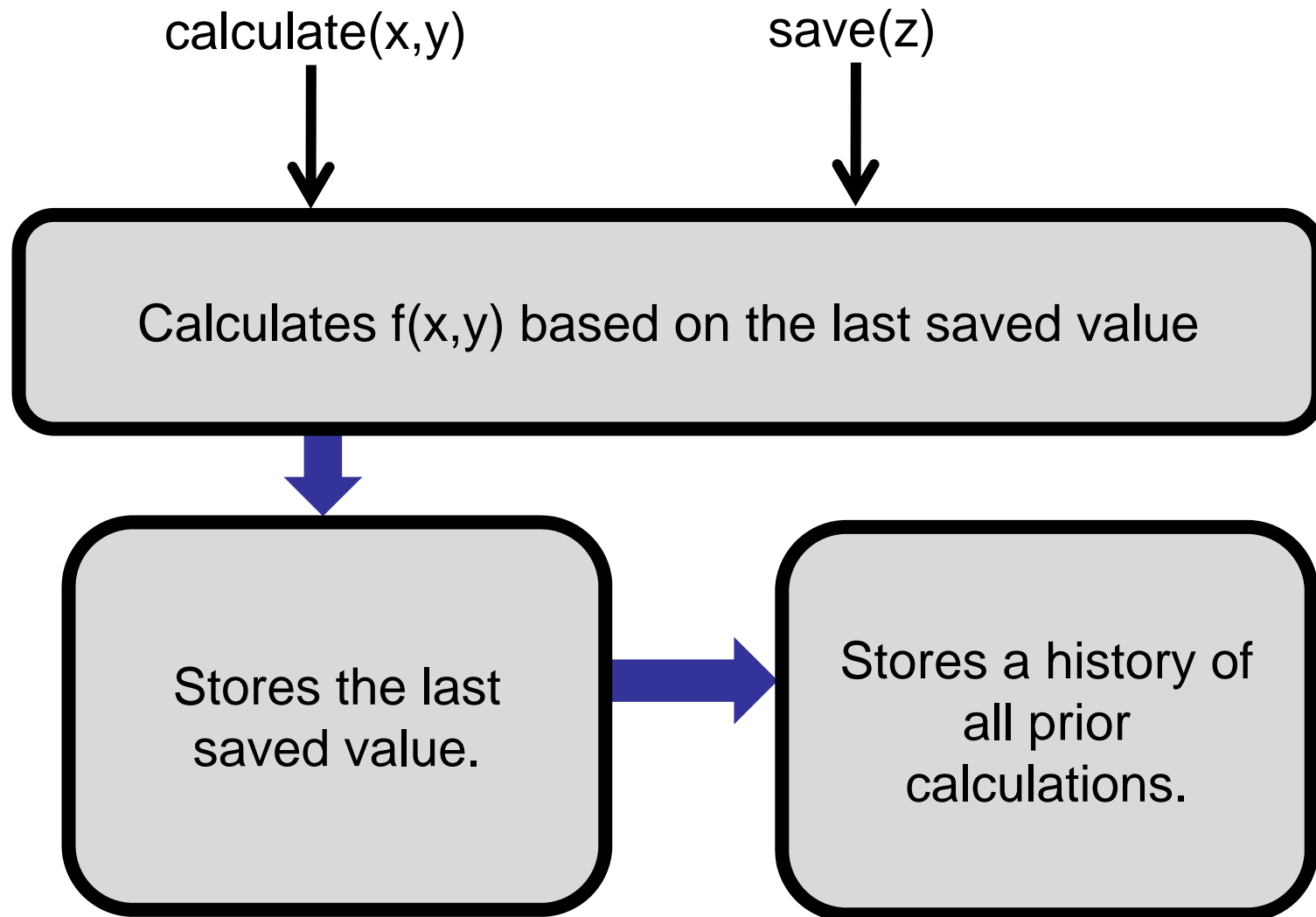
Abstraction



Abstraction



Top Down Design



Abstraction

Software engineering

- Divide problem into components.
 - Define *interface* between components.
 - Assign one team to build each component.
 - (Recurse.)
-
- Top down design: get the big idea first, then figure out how to implement it.

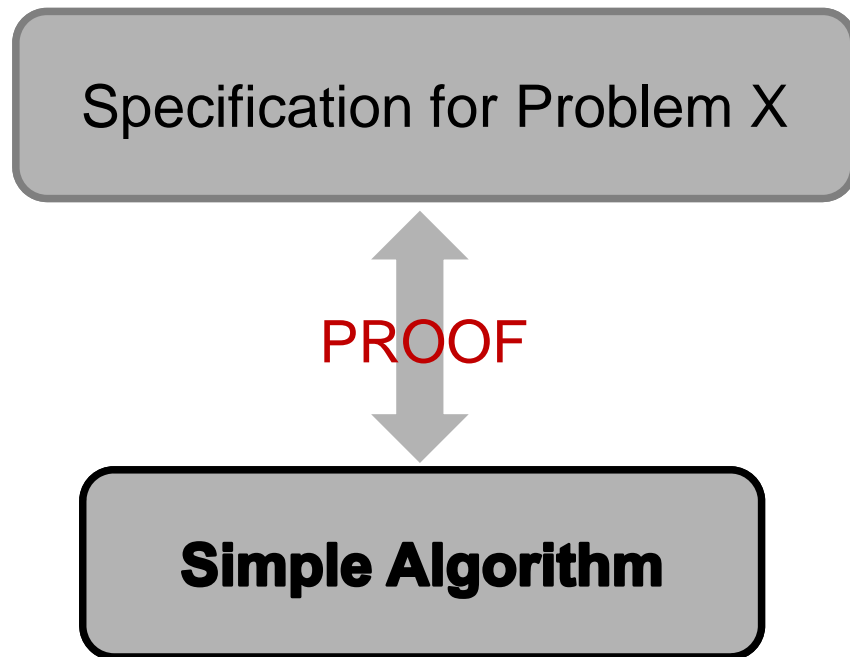
Abstraction

Algorithm design

- Divide problem into components.
- Define *interface* between components.
- Solve each problem separately.
- (Recurse.)
- Combine solutions.

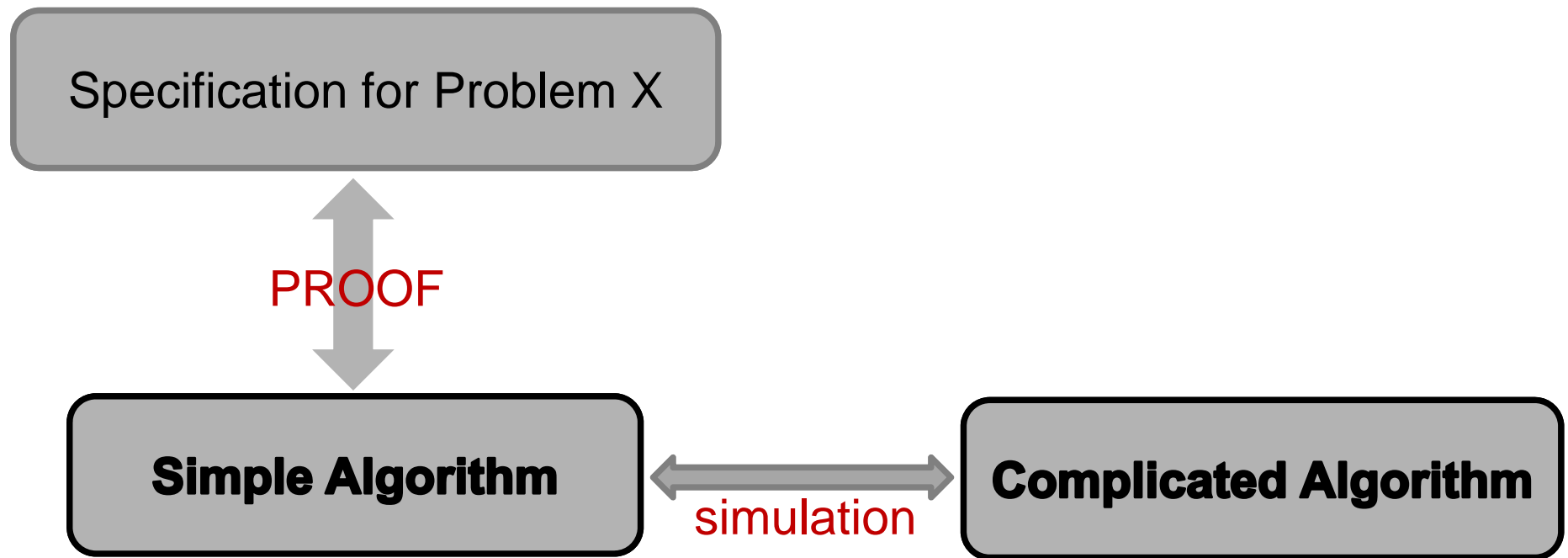
Abstraction

Proving an algorithm correct



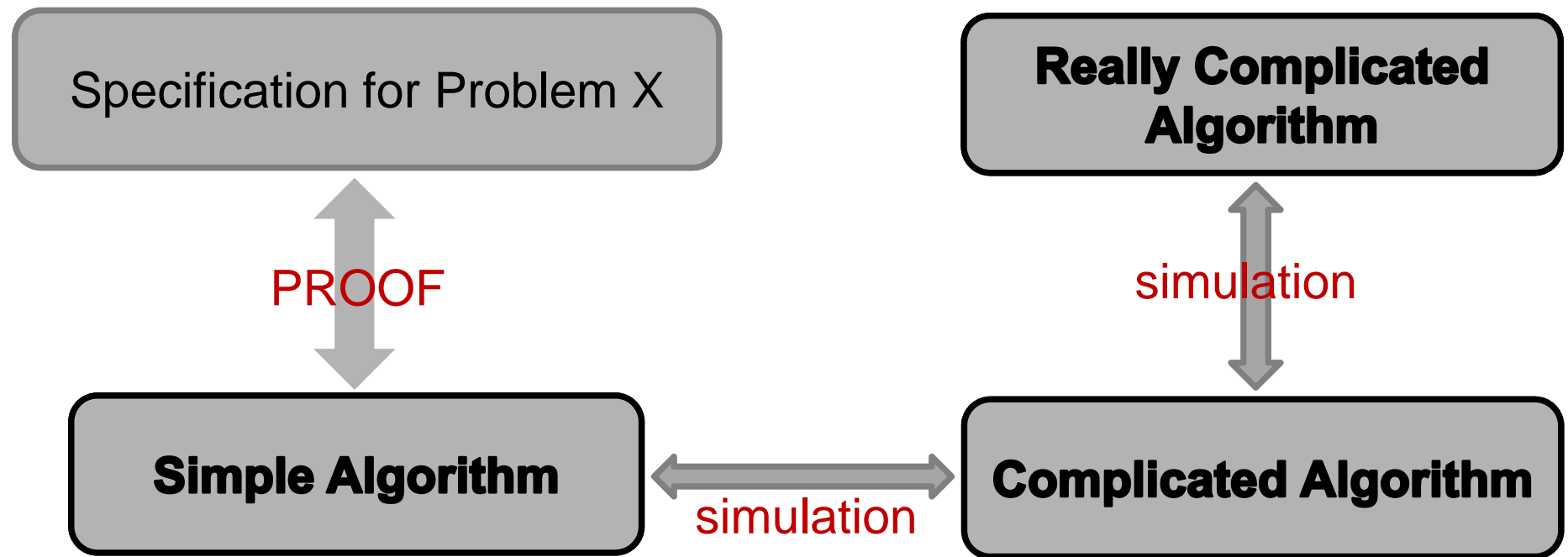
Abstraction

Proving an algorithm correct



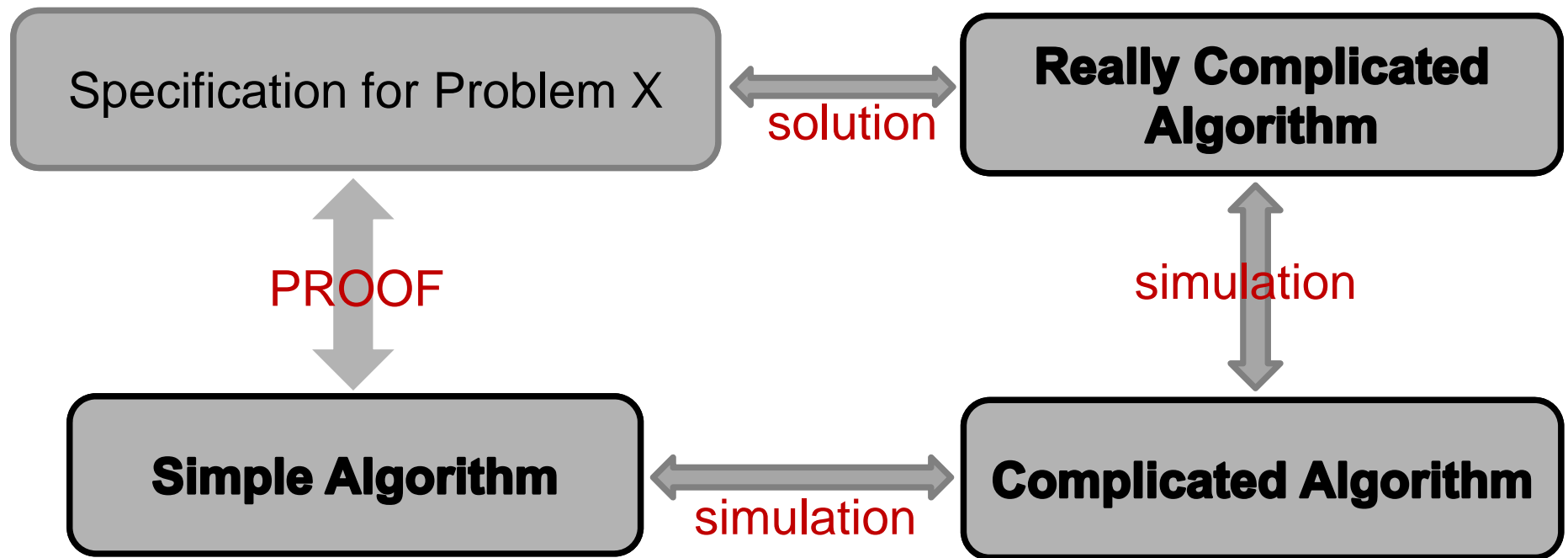
Abstraction

Proving an algorithm correct



Abstraction

Proving an algorithm correct



Abstraction

Key ideas

- Separate interface and implementation
- Hide implementation details
- Modularity: implement/analyze components separately