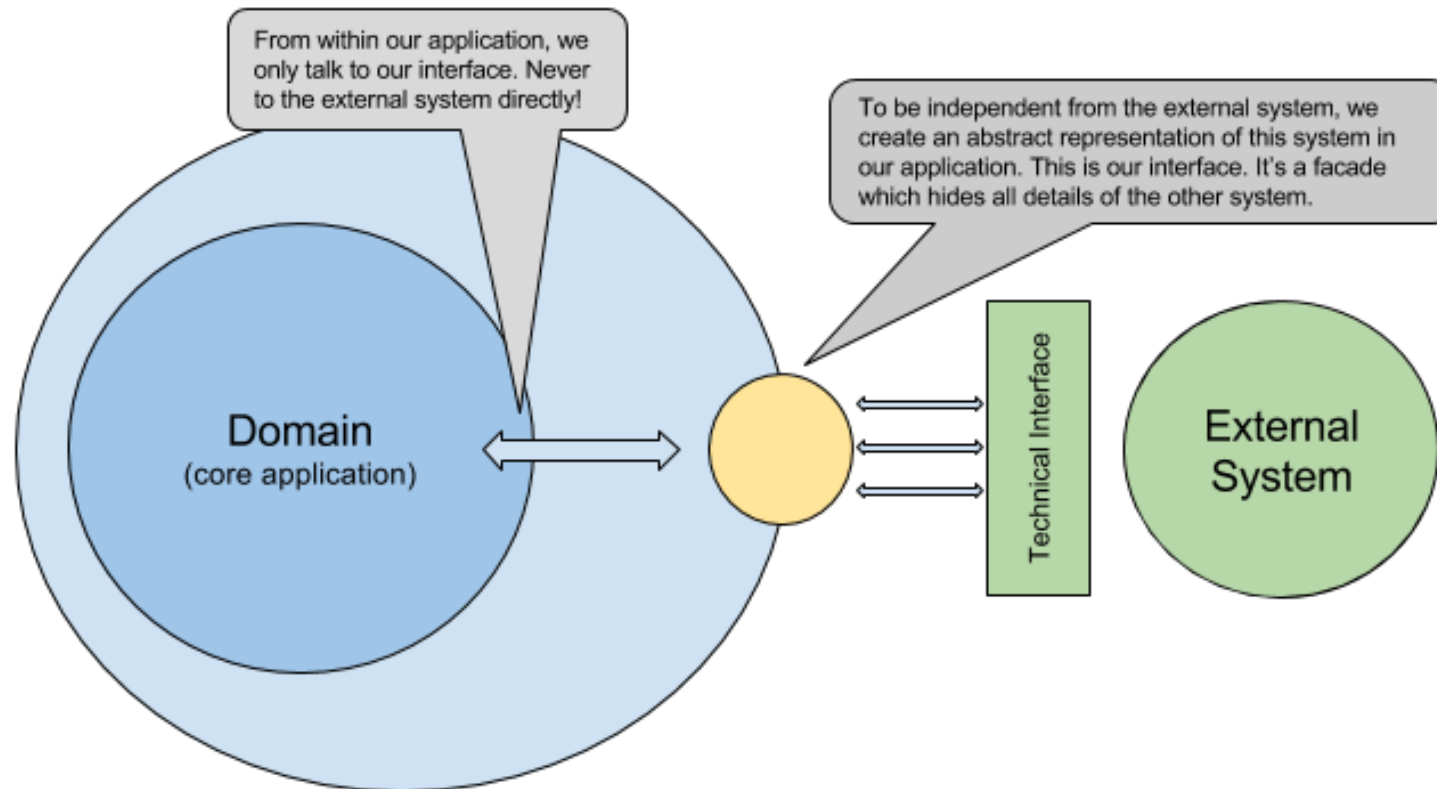# Programming to an interface

(not an implementation)

The design principle

Separation between **domain** and **external systems**
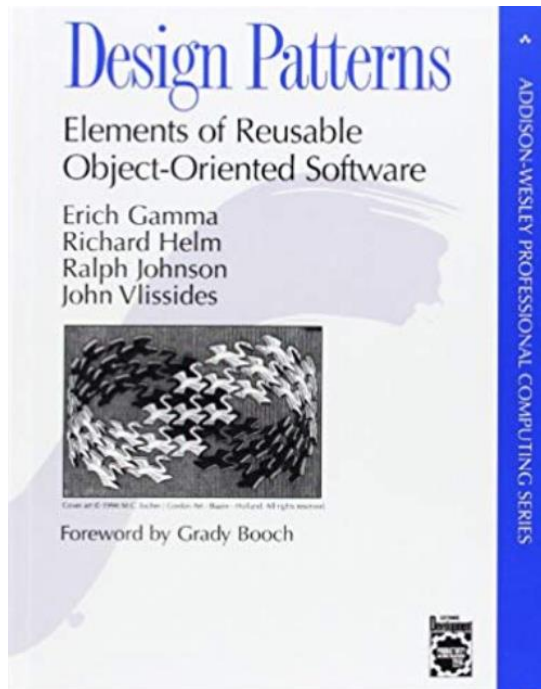
- Focus the design on **what** the code is doing, not **how** it does it

interface   implementation

From "Design Patterns" (GoF)

The authors claim the following as advantages of interfaces over implementation:

- *Clients remain unaware of the specific types of objects they use, as long as the objects adhere to the interface that clients expect.*

- *Clients remain unaware of the classes that implement these objects. Clients only know about the abstract class(es) defining the interface.*

NGramDAO
+ metric: Metric
+ findMatches(...): Map<Tu, Double>

<<Interface>> Metric
+ compute(...): double

FuzzyWuzzy
+ compute(...): double

SmoothBleu
+ compute(...): double