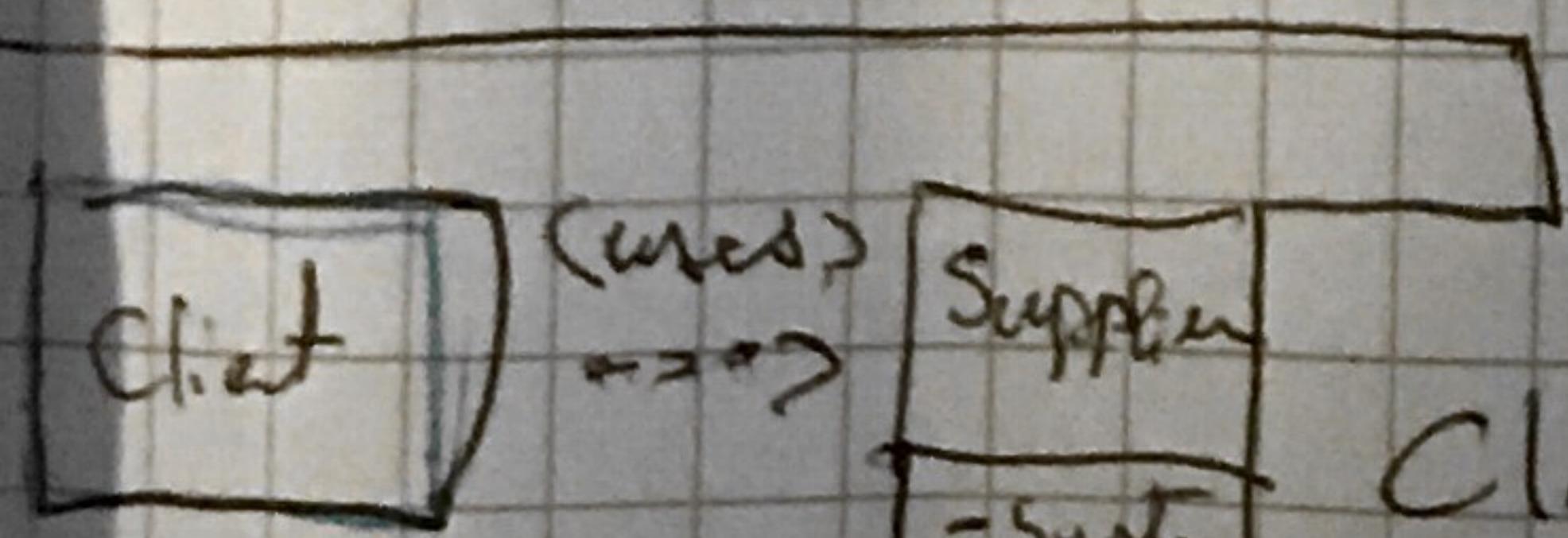
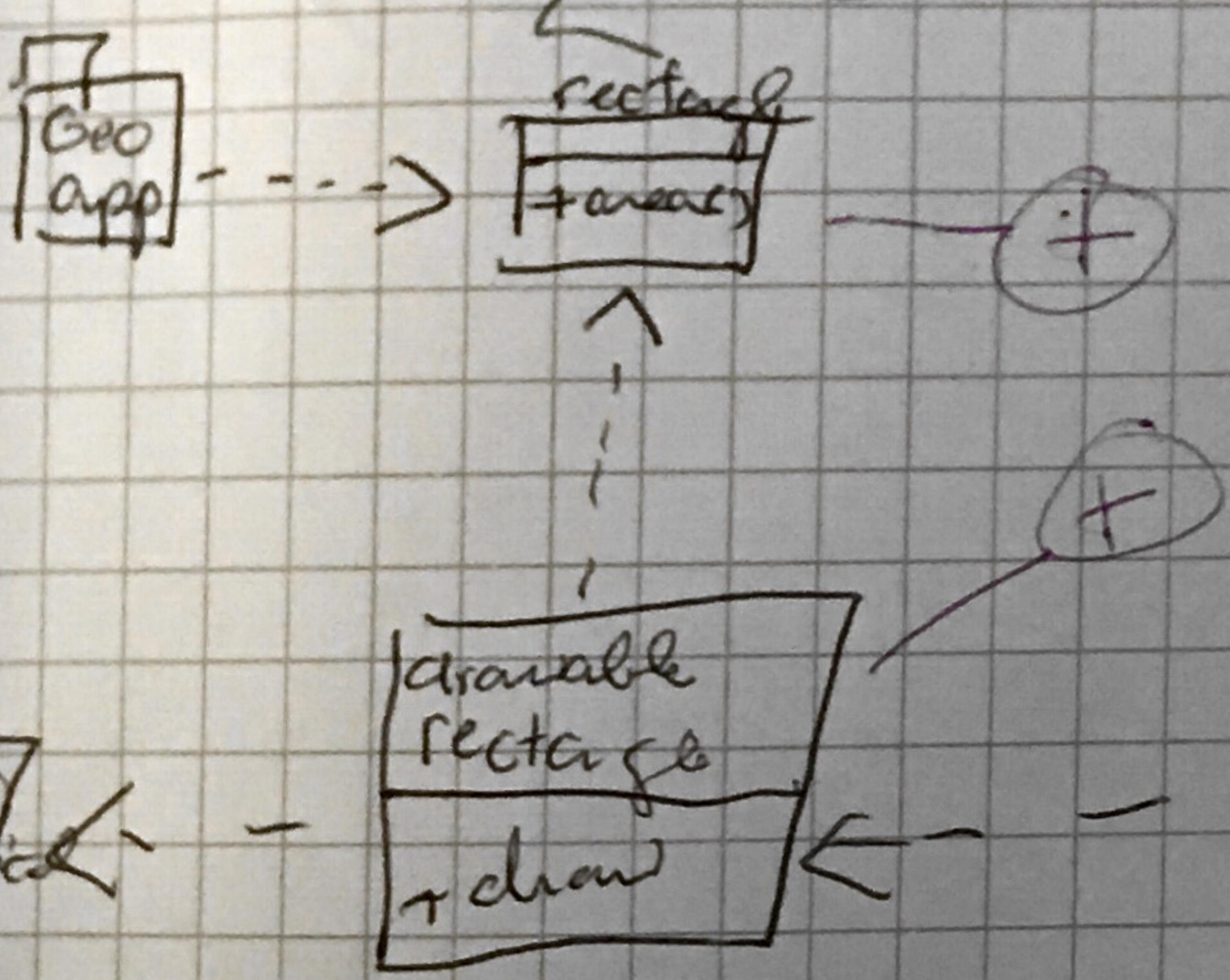
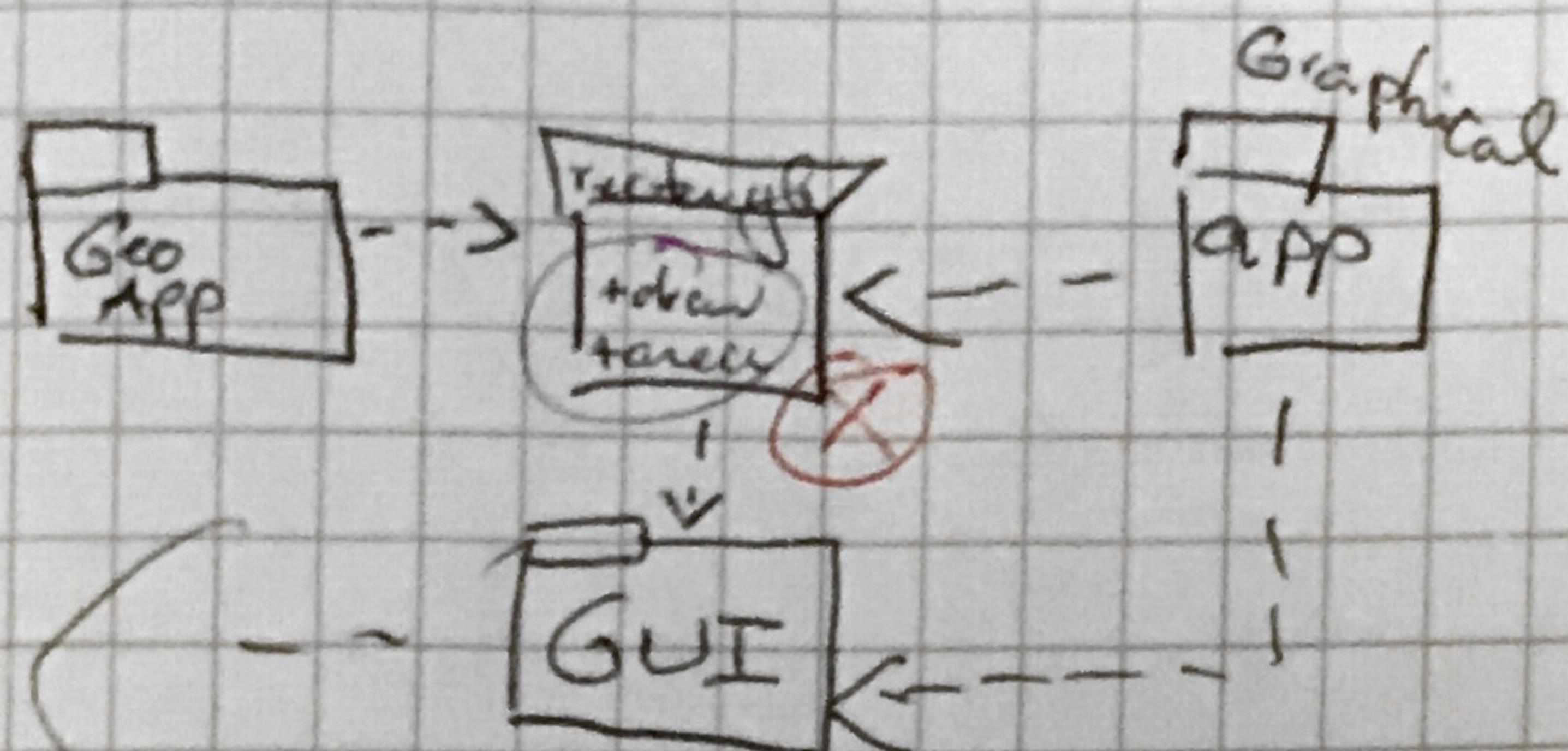


Content - Sys  
Single responsibility principle

- every class should have a responsibility
- every responsibility should be entirely in a class

Violating SRP



Class Person E violates SRP

- closed if not due to change
- open "available for extension"

private String PPS;

does != >= ?  
have

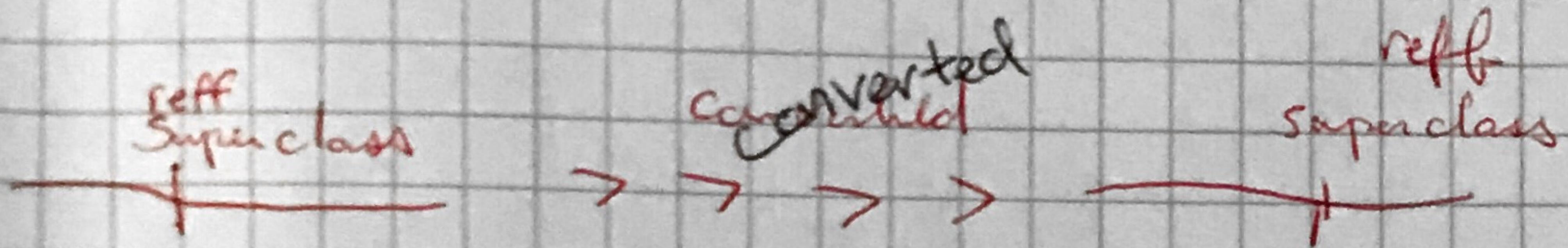
leads to ~~duplicated~~ code

OCP  
Open  
Close  
Change

## Liskov Substitution Principle

important

States the condition under which reference to Superclasses can safely be converted to reference to subclasses



- includes interface

→ any context that expects a member of a generalized class must work correctly when provided with an appropriate object

Stranger = demand more, promises less

- an overriding operation must have a weaker or  $\leq$  precondition and a stronger or equal post condition than the one it overrides

"be concise"

Java needs to be type safe

Contravariance? The type of ...

- demanding an argument type in an overriding method → must be some type

Covariance: permits runtime type errors to occur

method overloads for it to be true

→ - explains what Substitution means in the context of inheritance

1. Single Responsibility (P)
2. Open Closed (C)
3. Liskov Substitution (P)
4. Interface Segregation (P)

## Interface Segregation principle (ISP)

- tailor interfaces to the clients specific req

