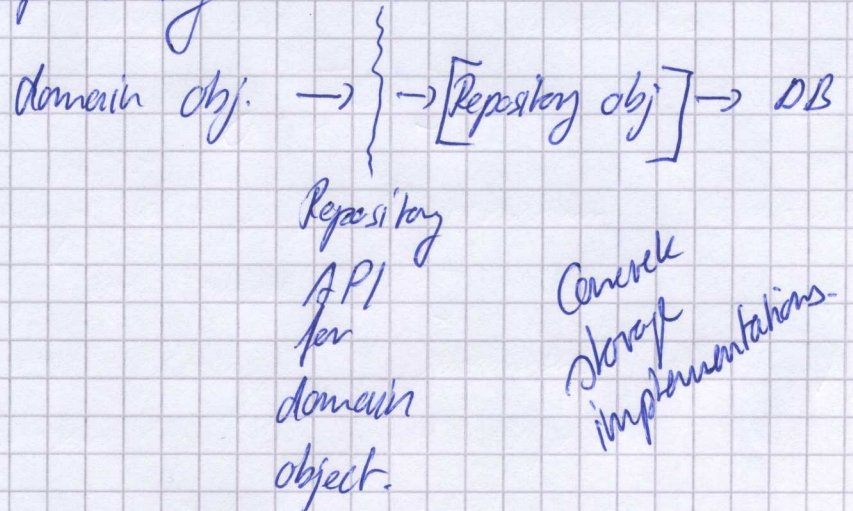Service: Operations that do not belong to
the domain knowledge of an
object.

Repository: Retrieves domain object (value, aggregate
or Entity) and provides methods
for storing

domain obj. $\longrightarrow$ } $\rightarrow$ [Repository obj] $\rightarrow$ DB

Repository
API
for
domain
object.

Concrete
storage
implementations.

Factory: Methods for creating domain
objects should delegate to a specialized
factory object to make domain
object alternative implementations
possible and interchangeable

[DDD book]: „Model as distilled knowledge"
„bundeling data and behavior."
Encapsulate the behaviour in common
types

All in all:
- compiler → type safety
- encapsulation → with possible operations
- readability: → clear where to find

cyclic
in Rule?