# Web componenten

* Een essentiele onderdeel: de Spring container. Het is een verzameling van objecten.

Verzameling van objecten

* **High cohesion** : alles wat te maken heeft met een reken machine zal in de reken machine blijven en niet buiten. En alles die niets te make heeft met een reken machine gaat niet in de reken machine.
* Slecht voorbeeld with service en doGet methodes : een servlet dit alles doet: connection with DB, write stuff in the DB, more interne calculation, writing an output to the response.

## Loose coupeling

MyServletcontroller

PersonDAO

Connection

* Service()
* doGet()

PersonService (interface)

* calculateAge()

BadImplementation

* calculateAge()

causes memory leak

GoodImplementation

* calculateAge()

works great

* De beans worden in de container gestoken met een naam.
* We zullen independency injection gebruiken.

# Application Context

* De manier om contact te hebben met Spring is via de ApplicationContext.
* Darmee kunnen we objecten vragen die in de container is. We zullen rechtstreeks de objecten van to container ophallen.
* Een bean is hetzelfde als een Java object.

# Application Context

Spring begint vanaf het moment als we een niew ApplicationContext(AppConfig.class) object vragen.

De parameter (AppConfig.class )is een configuratie bestand.

Wij babellen met onze appplicatie context.

De container is een lijst met beans.

LIbraries

Container with beans

SPRING

## Bean factory

Een bean factory is een object die beans maakt.