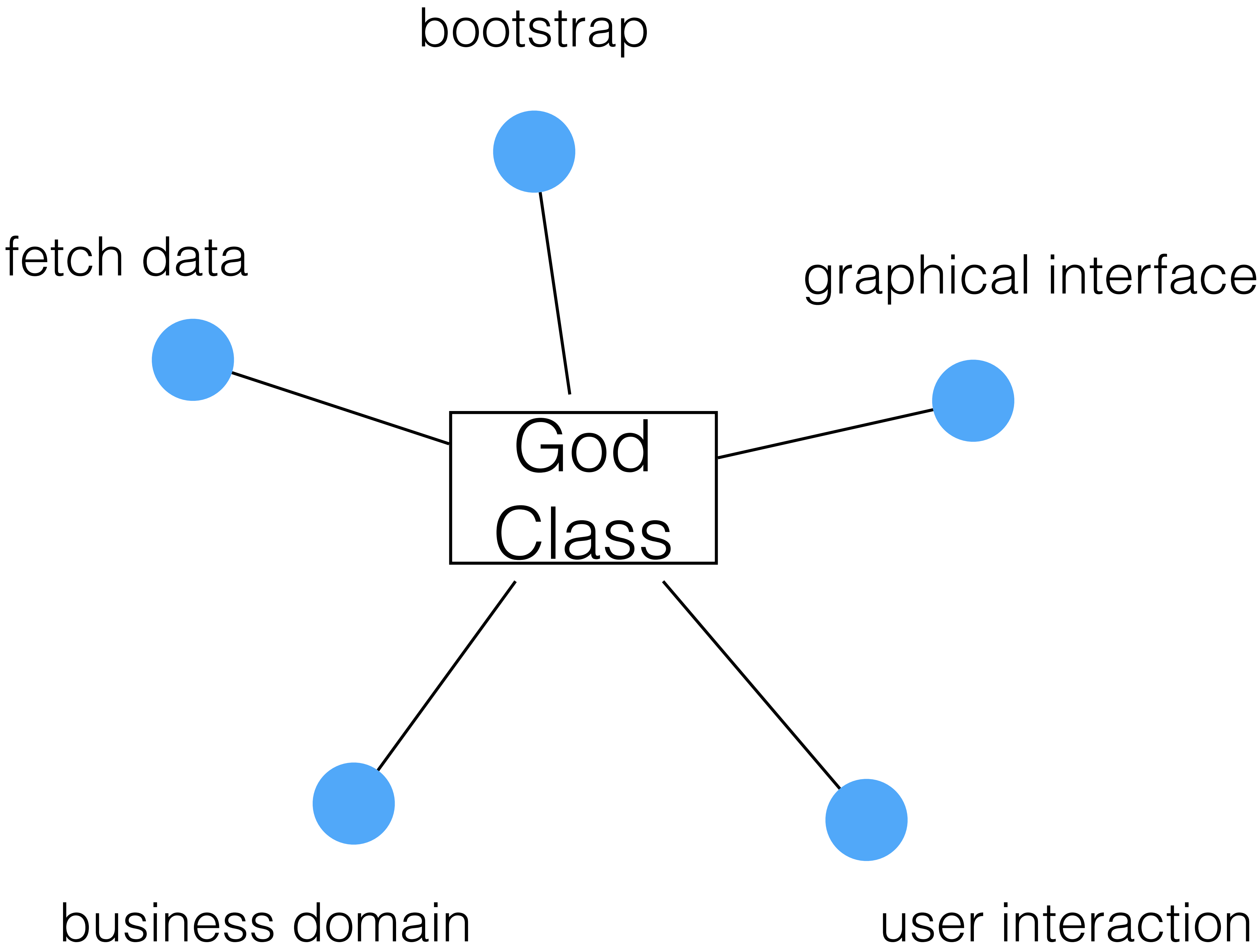
MVC

Supplementary presentation to exercises
ESE 2017, University of Bern
Slides by: Manuel Leuenberger



bootstrap

fetch data

graphical interface

God Class

business domain

user interaction

Architecture

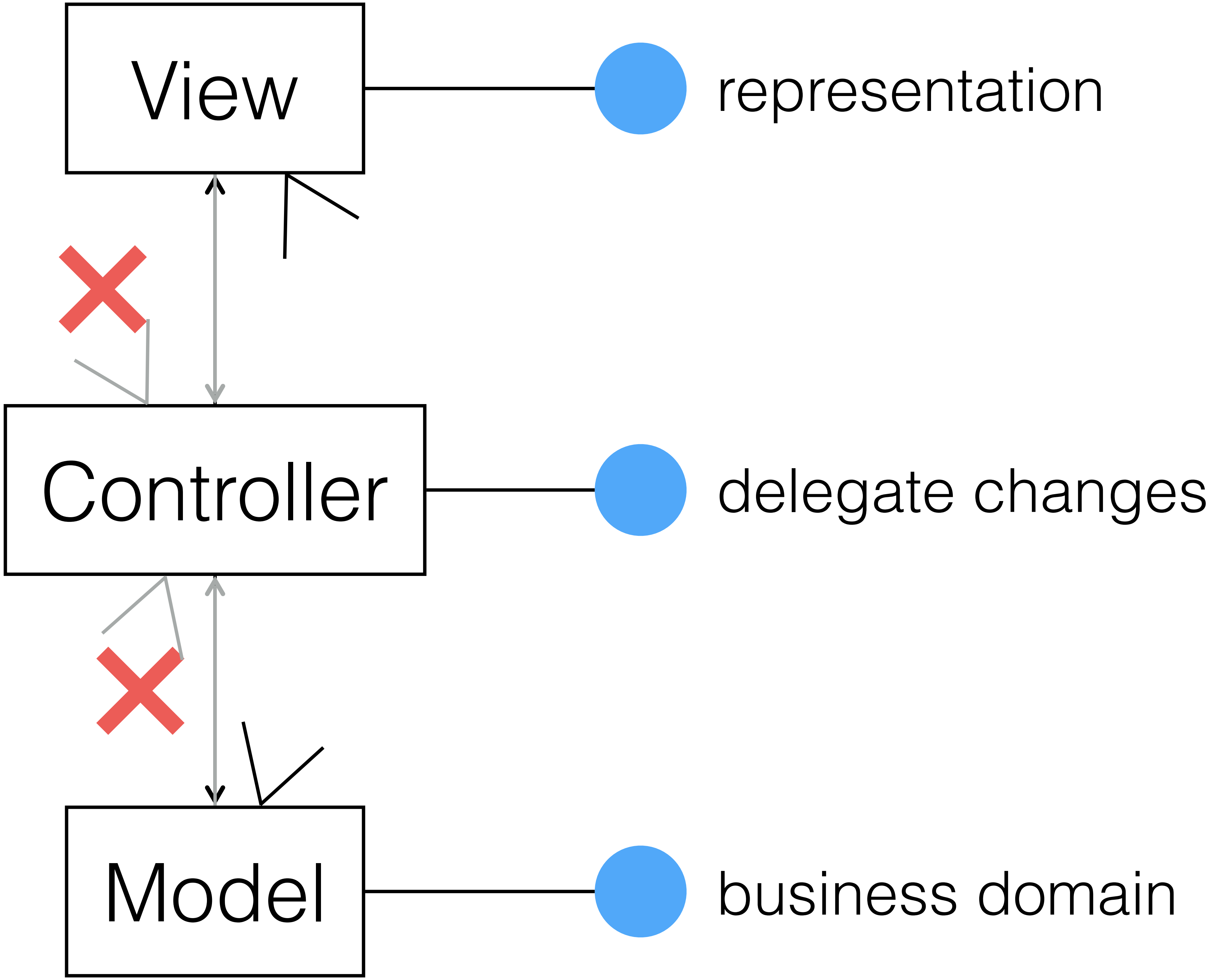
to the rescue!

manage complexity
enable reuse

Model

View

Controller



View

representation

Controller

delegate changes

Model

business domain

Spring MVC

„automagically“

Request lifecycle summary

- user navigates to URL
- request is mapped to **controller** method
- **controller** loads **model** by id
- **controller** pipes model data into **view**
- user gets rendered **view** as response

Tip

- become fluent in the web **vocabulary**
- understand concepts of used **technologies**
- HTTP: request & response, headers, body, method, status
- URL: host, path, query, hash, URL-encoding
- HTML: element, attribute, form-control
- CSS: selector, property, position, display
- JS: jQuery, JSON, DOM
- Documentation: stackoverflow, JavaDoc, Guides

Summary

- **m**odel, **v**iew, **c**ontroller
- architectural pattern, not a design pattern
- divide & conquer
  - **model** is business domain: state & behaviour
  - **view** represents **model**
  - **controller** delegates changes between **model** and **view**
- responsibilities are clearly defined without overlapping