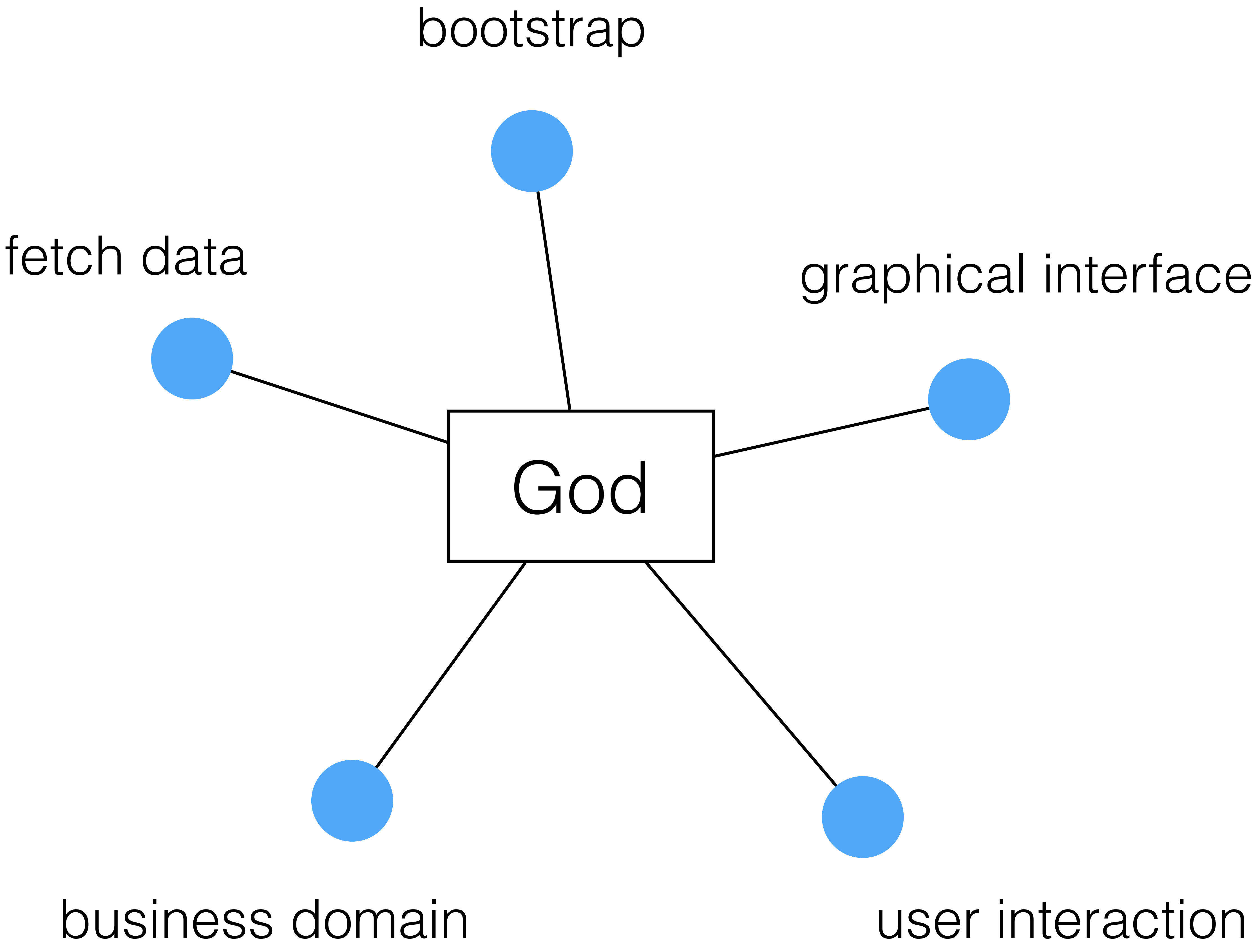
MVC

Supplementary presentation to exercises
ESE 2016, University of Bern
Manuel Leuenberger

God’s Tinder



bootstrap

fetch data

graphical interface

God

business domain

user interaction

Architecture

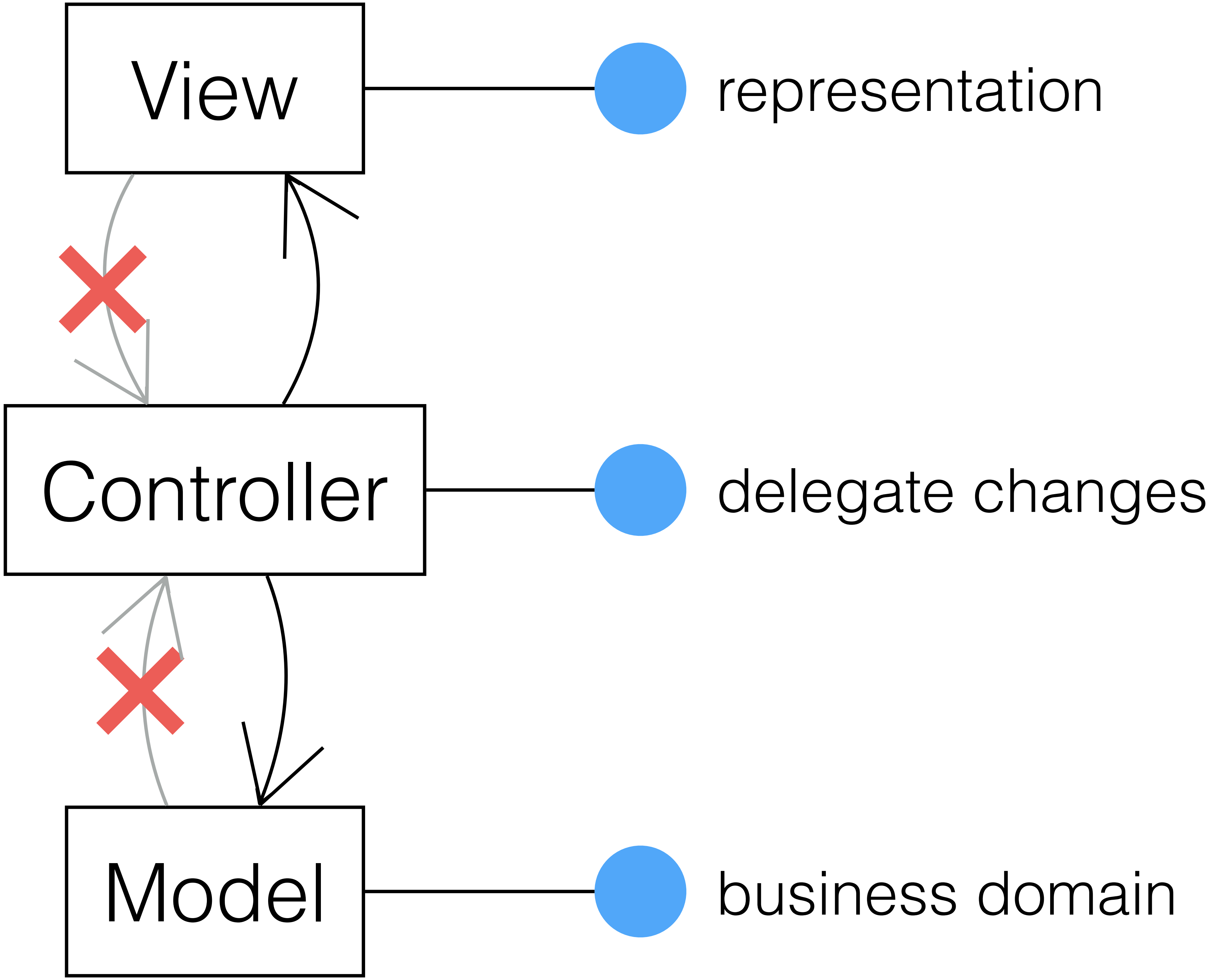
to the rescue!

manage complexity
enable reuse

Model

View

Controller



View

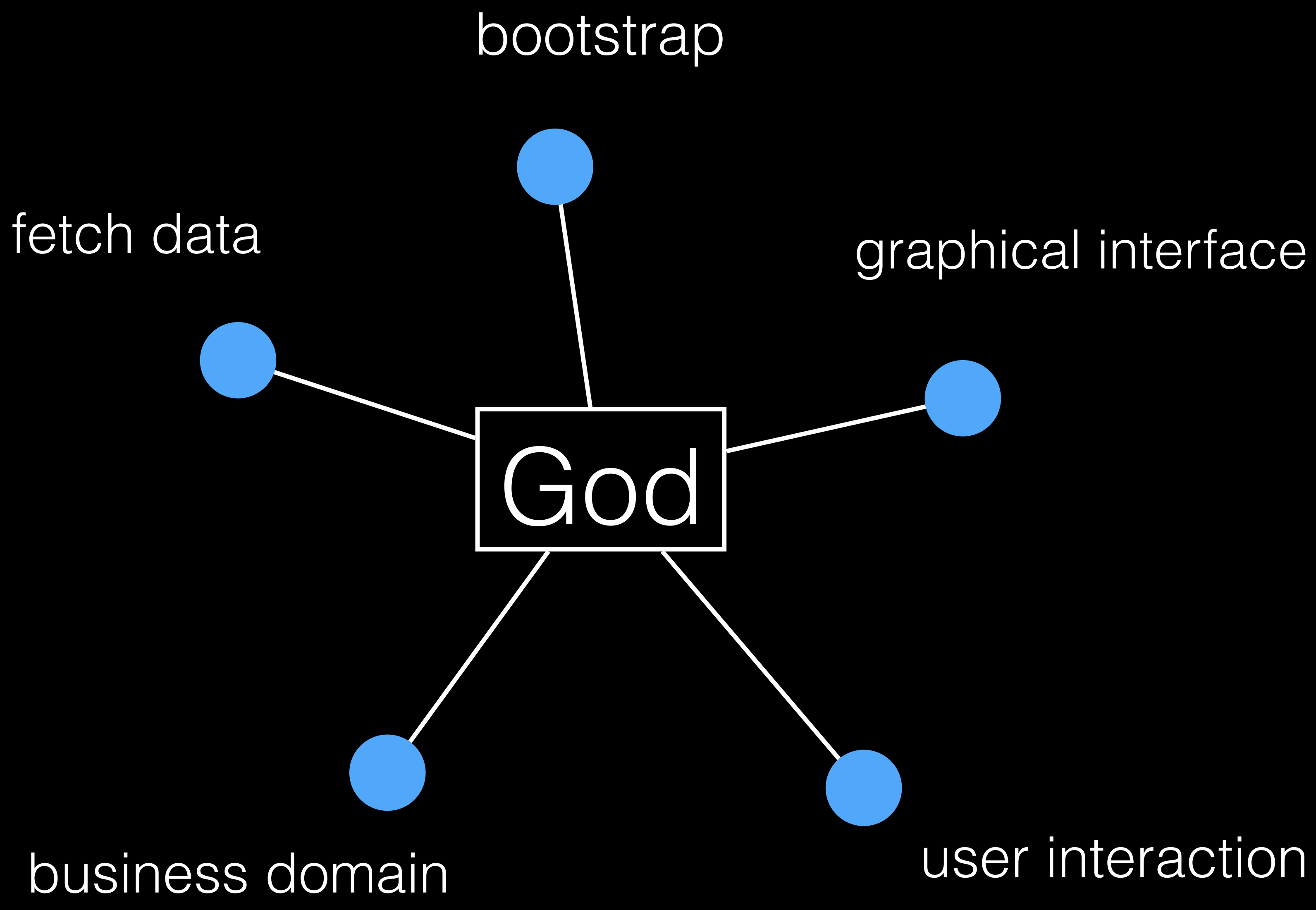
representation

Controller

delegate changes

Model

business domain



bootstrap
fetch data
graphical interface
God
business domain
user interaction

Spring MVC

Request lifecycle

- 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: RFC, MDN, stackoverflow, W3C, JavaDoc, Guides

Summary

- **m**odel, **v**iew, **c**ontroller
- architectural style, 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