- Upside
  - No circular dependencies
  - Clear naming
  - Clear domain
  - Easy to read
  - Easy to maintain
  - Easy to evolve

components
accounts
allocations
backlog
jdbc-support
projects
rest-support
timesheets
users
databases

INTRODUCE APPLICATIONS, COMPONENTS, AND SERVICES. A DISTRIBUTED SYSTEM

a

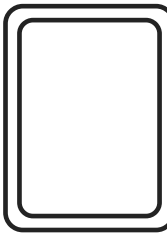
C

t

t

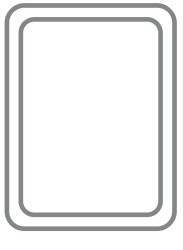
C

j

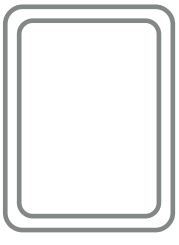


MULTIPLE APPLICATIONS, COMPONENTS AND SERVICES

application
src
main
java
App.java



App.java



App.java

Easy to tune/ scale

Easy to introduce new people/ scale teams

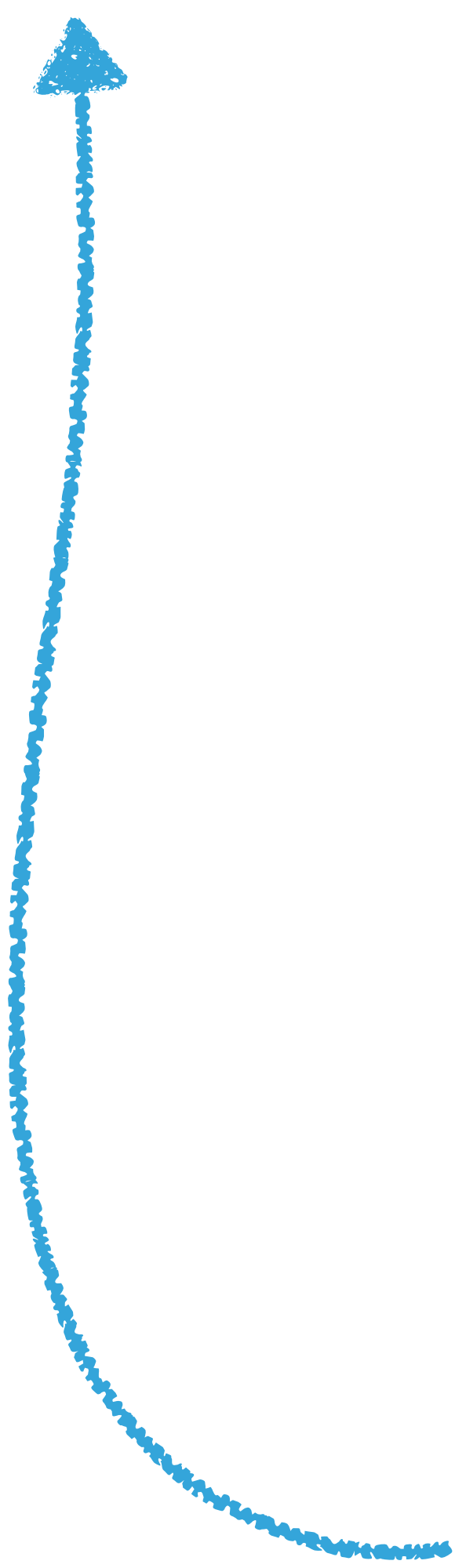
A

C

fi

C

y



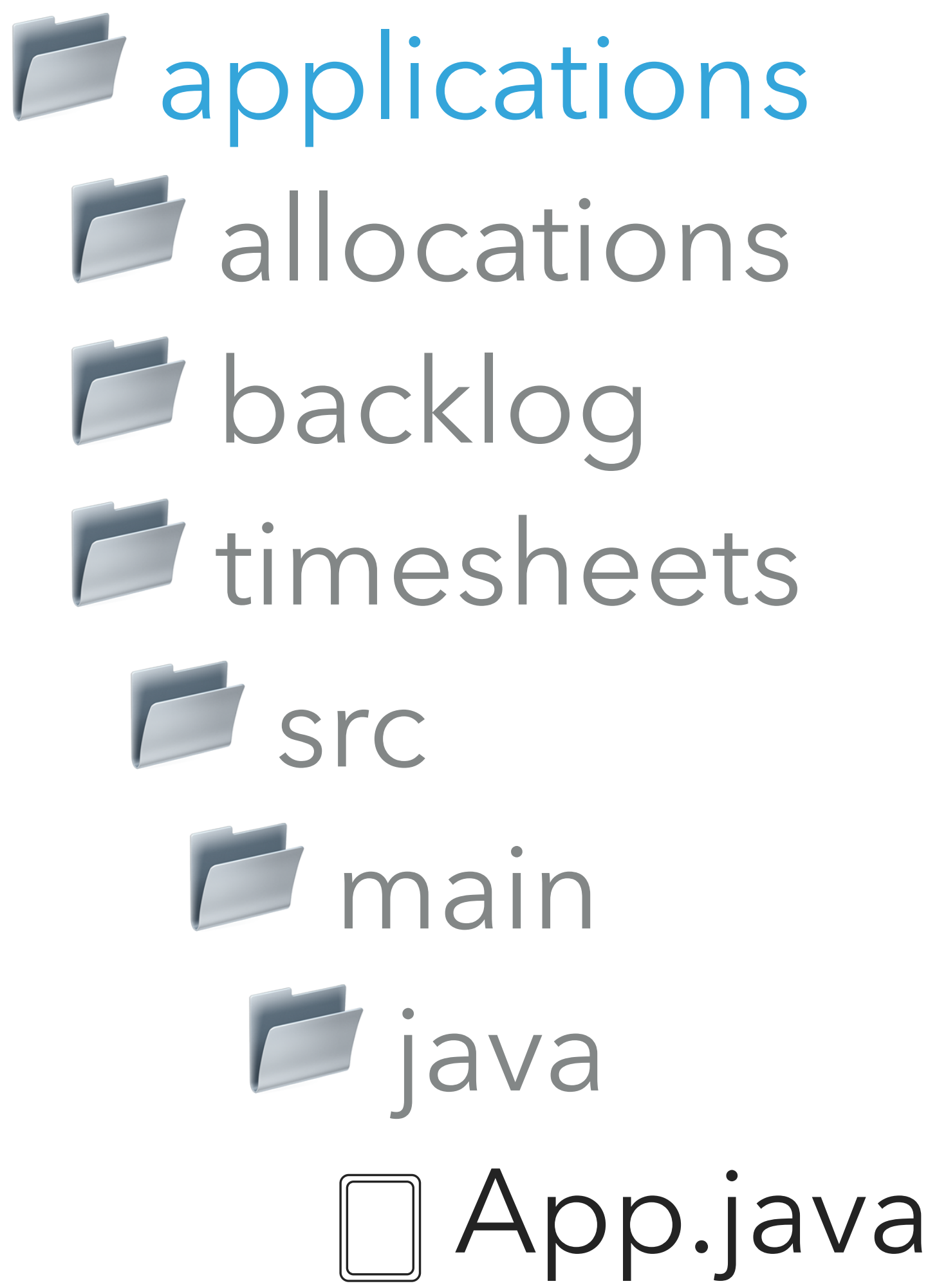
- Downside

New circular dependencies

Secure Communications

Distributed Transactions

API Versioning


applications
allocations
backlog
timesheets
src
main
java
App.java

Apps contain configuration and dependency information

MULTIPLE APPLICATIONS, COMPONENTS AND SERVICES

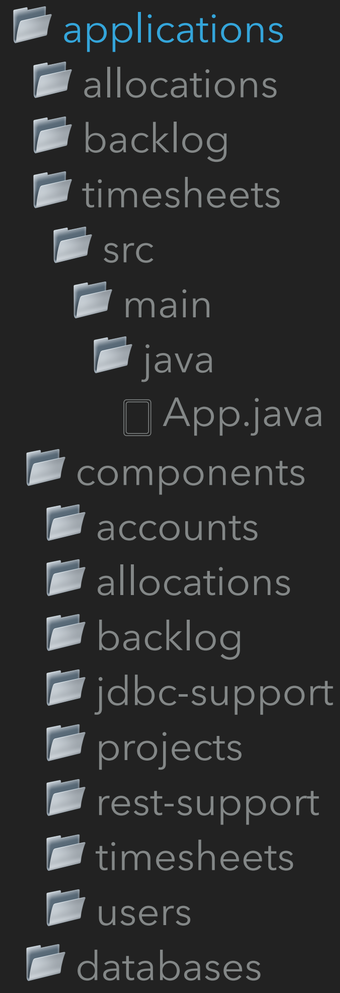
- Downside

  New circular dependencies

  Secure Communications

  Distributed Transactions

  API Versioning



THAT'S A WRAP. NOW ALL YOU NEED IS PRACTICE