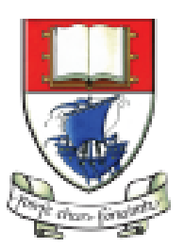
Assignment 1 (Structure 101 Analysis)

Produced by: Eamonn de Leastar (edeleastar@wit.ie)



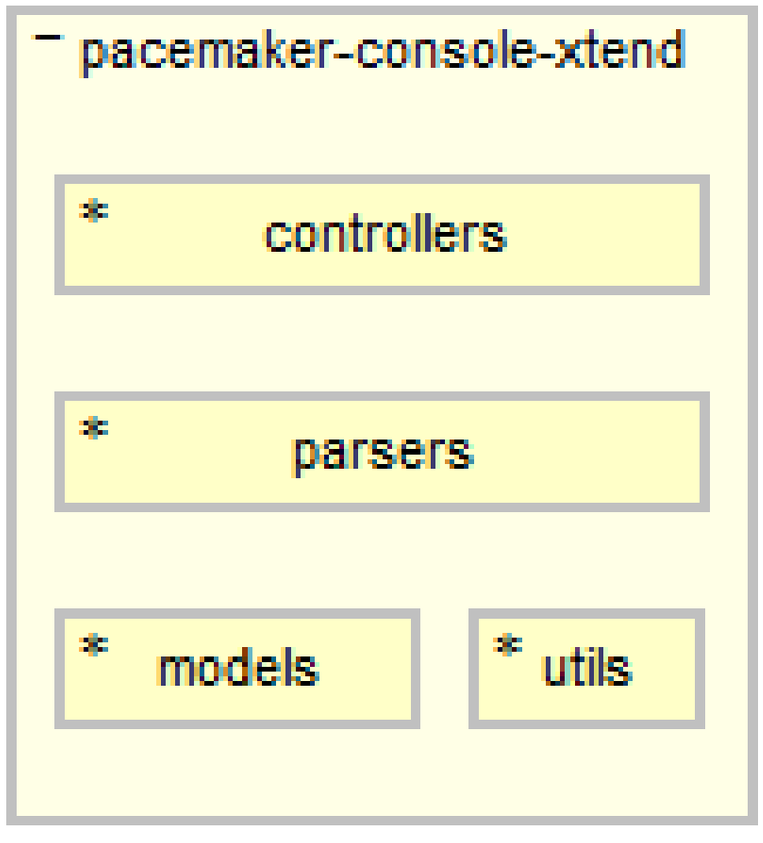
Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

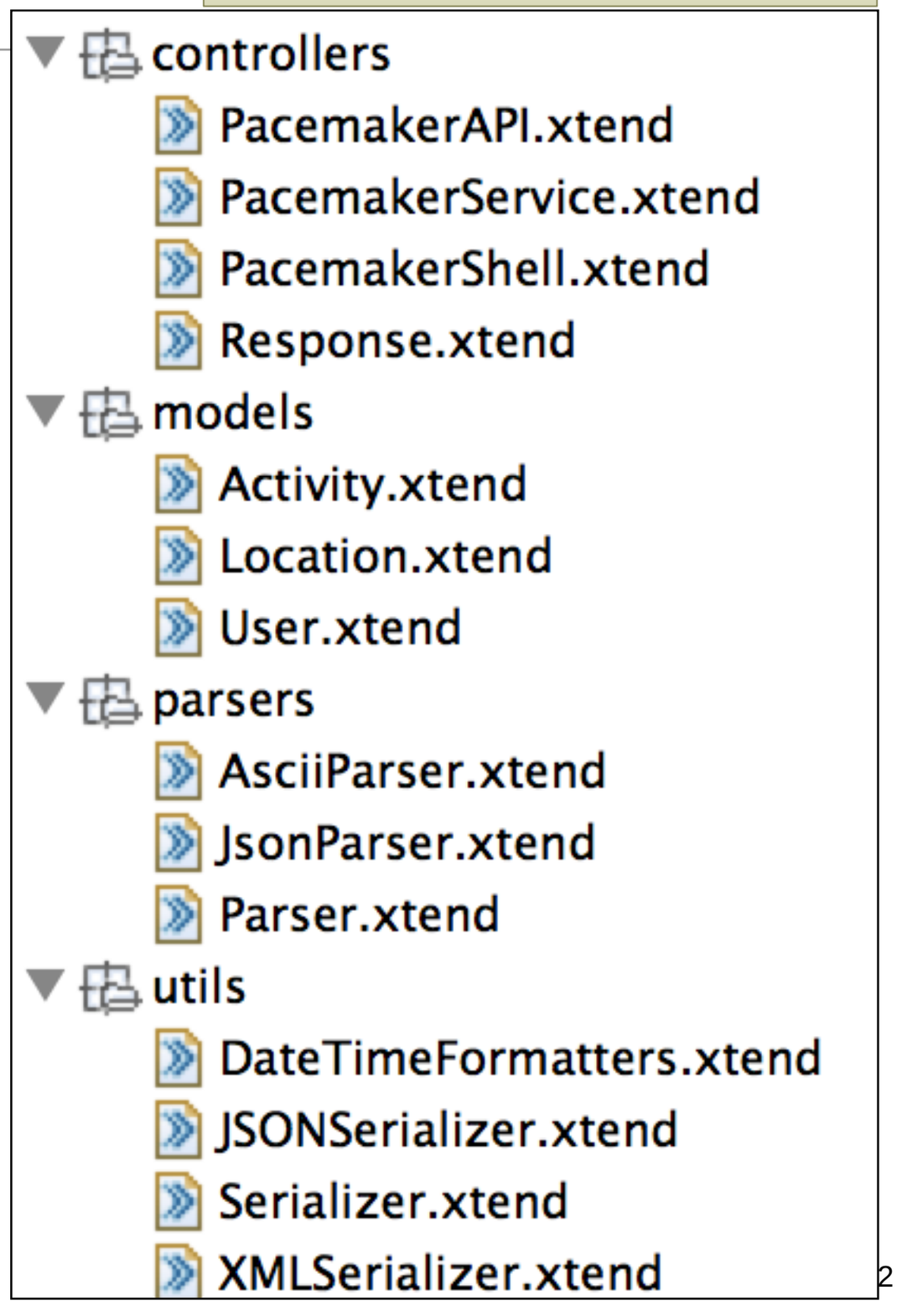
pacemaker-console-xtend

The assignment solution has:

- a number of source files.
- interdependencies between these files.

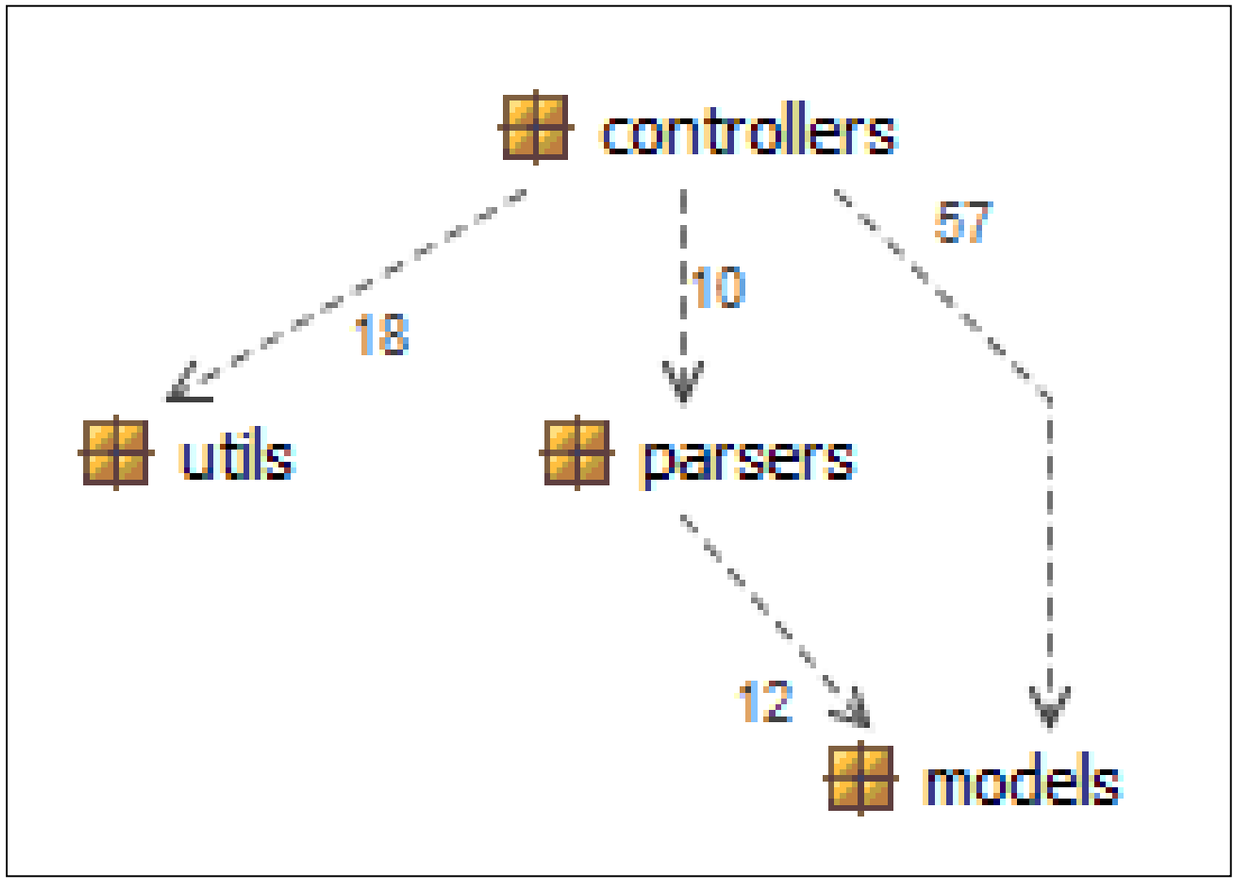


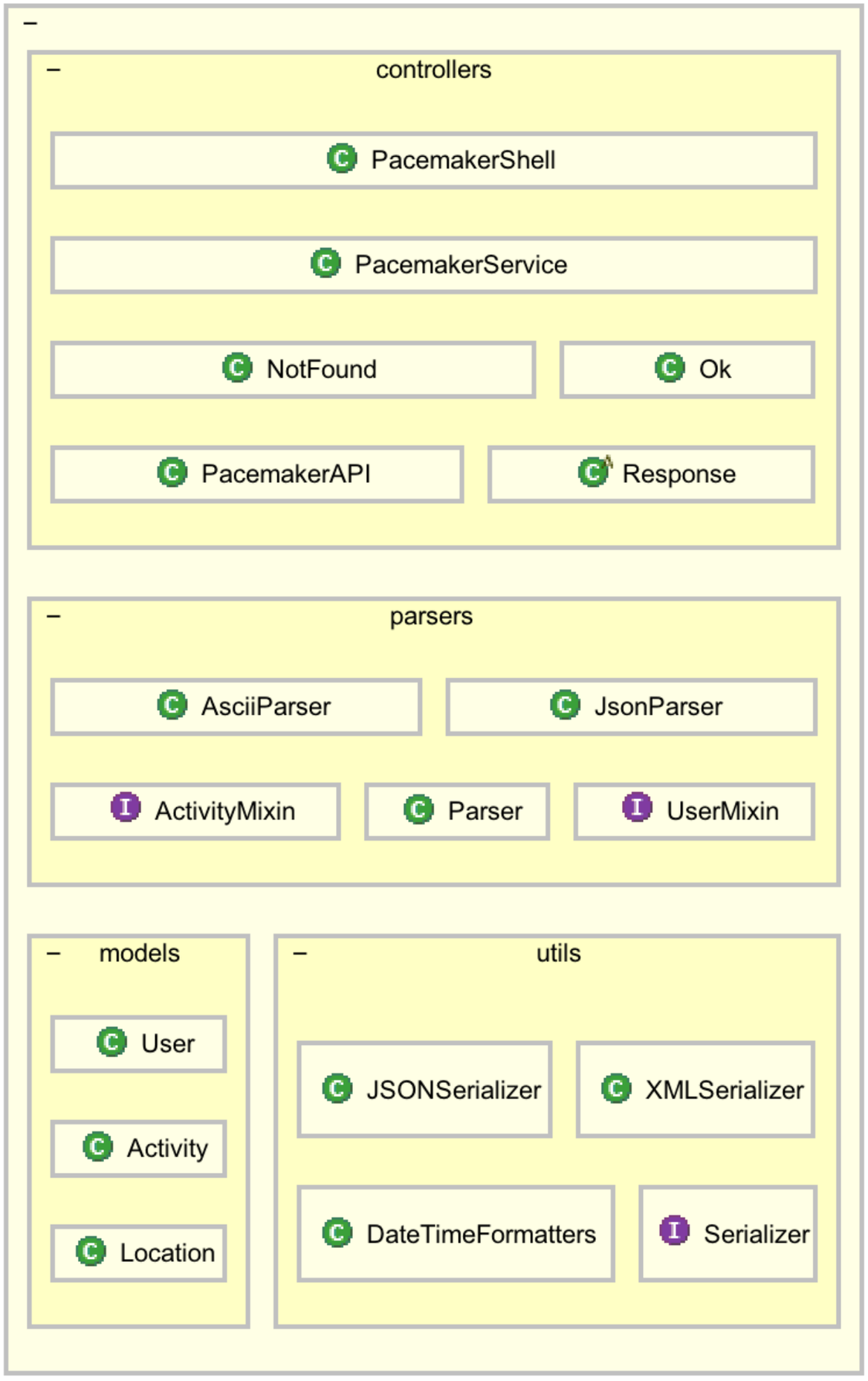
Physical Organisation



Containment model (dependency graph)

- Use containers (packages) to manage complexity.
- Containment creates dependencies between containers.
- How we organise our containers is key to controlling dependencies between the containers.



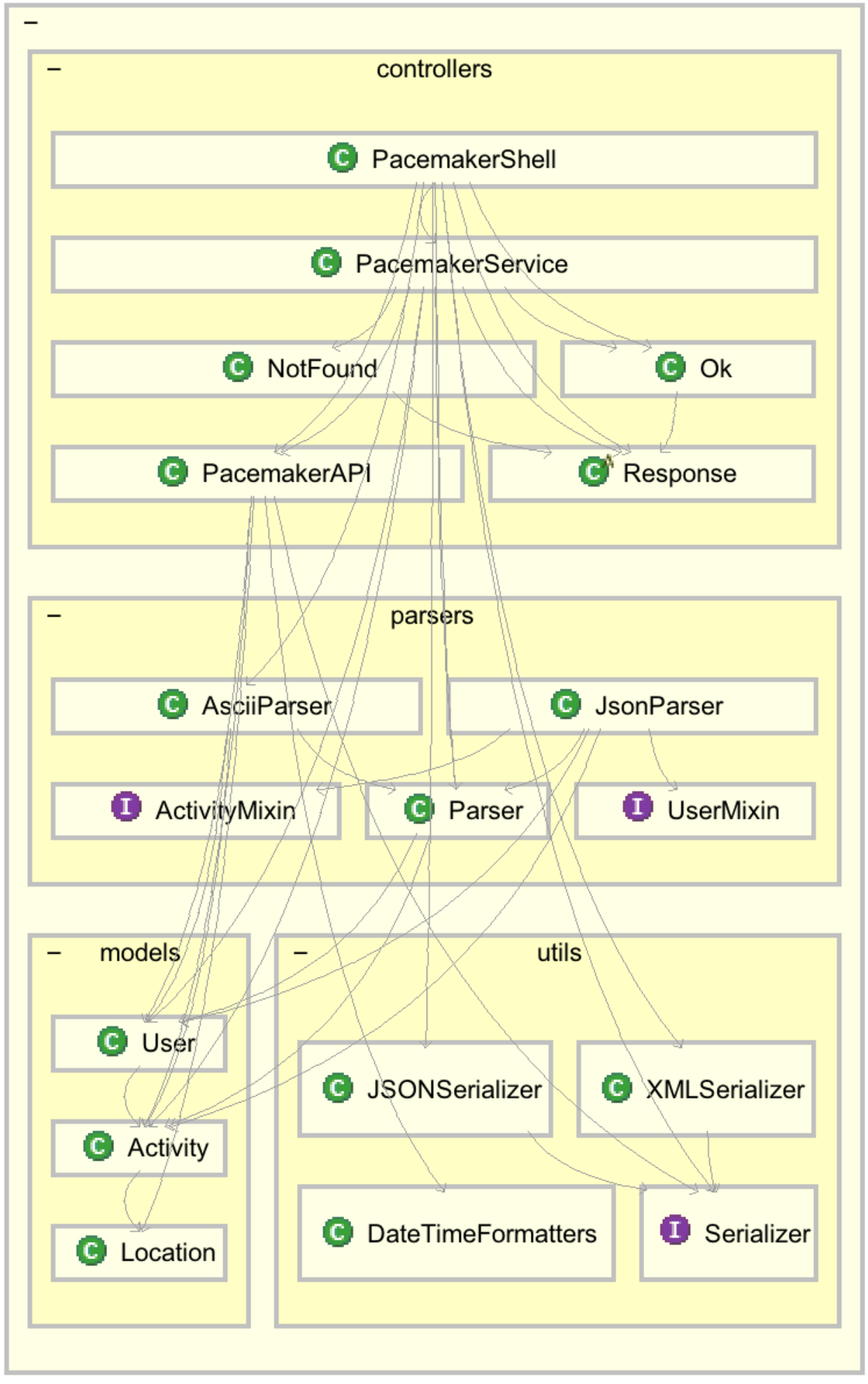

controllers
PacemakerShell
PacemakerService
NotFound
Ok
PacemakerAPI
Response
parsers
AsciiParser
JsonParser
ActivityMixin
Parser
UserMixin
models
User
Activity
Location
utils
JSONSerializer
XMLSerializer
DateTimeFormatters
Serializer

controllers
NotFound
Ok
PacemakerAPI
PacemakerService
PacemakerShell
Response
models
Activity
Location
User
parsers
ActivityMixin
AsciiParser
JsonParser
Parser
UserMixin
utils
DateTimeFormatters
JSONSerializer
Serializer
XMLSerializer

Dependencies

- When dependencies all point in the same direction (e.g. downwards), the cumulative dependency is smallest.

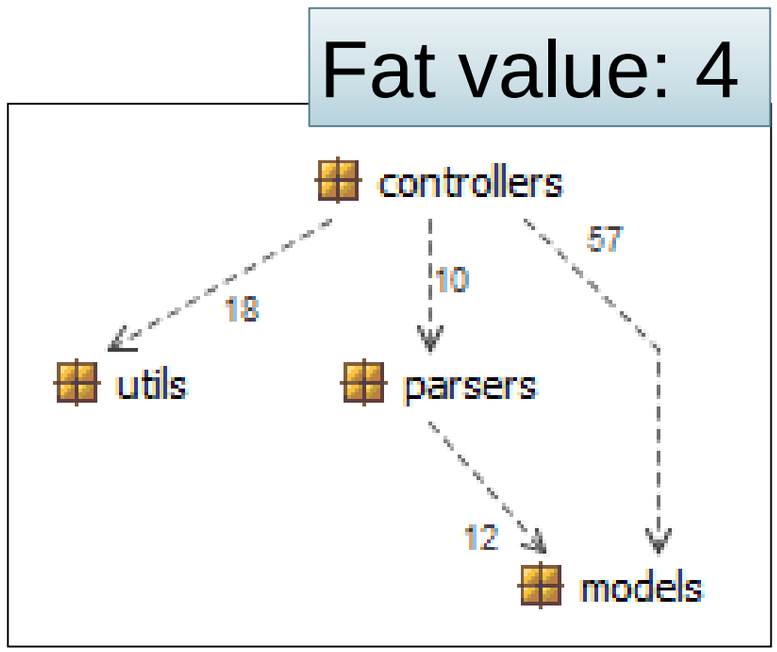
- When you have cyclical dependencies (i.e. dependencies pointing upwards) the cumulative dependency becomes large and you have what is called a **Design Tangle**.



Measuring complexity in a code base.

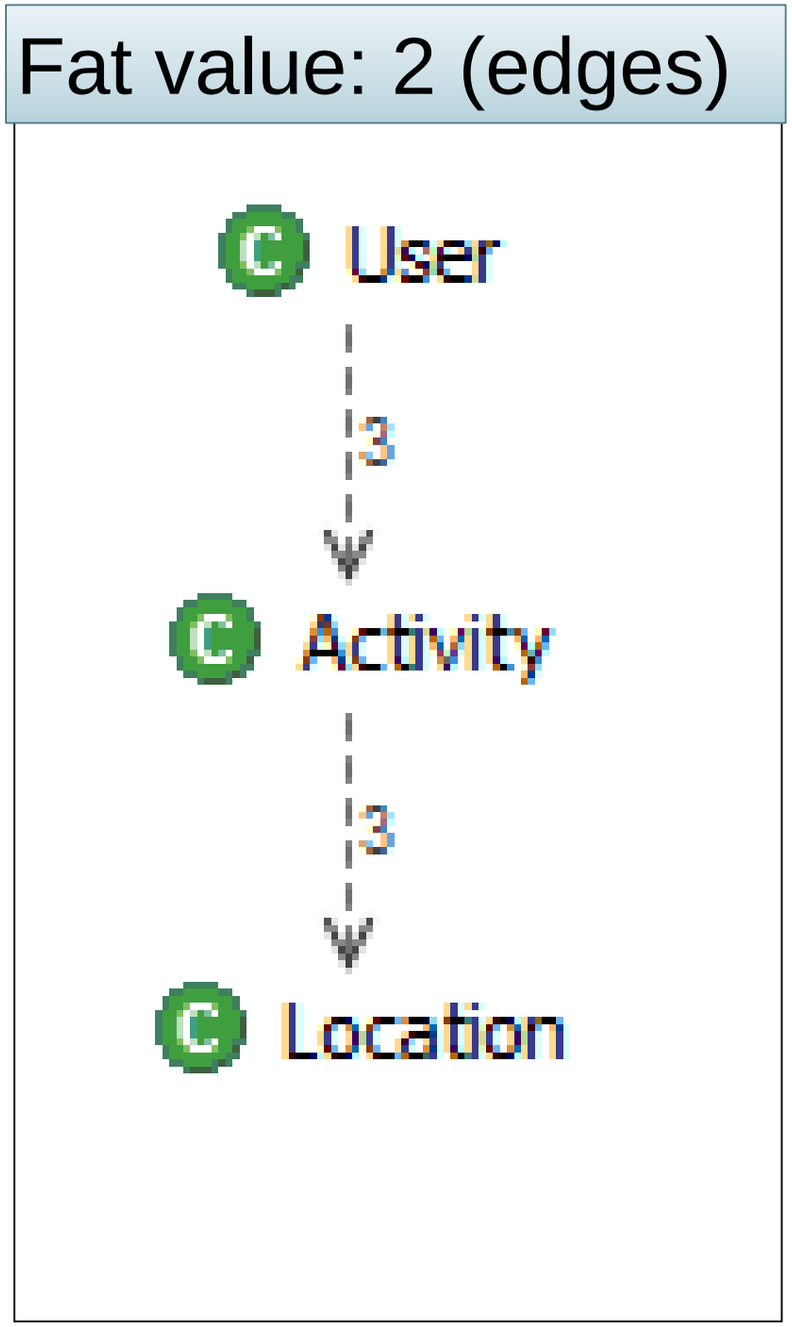
Excess complexity is measured in terms of 2 metrics:

- Fat is a measure of the compositional (breakout) complexity at ***every level*** e.g. number of edges in a dependency graph. Default threshold in Structure101 is 15 (methods), 120 (design, package, class level).
- Design Tangles are cyclic dependencies between packages, measured at the breakout of high-level packages ("design level"). Default threshold in Structure101 is 0% - always undesirable!

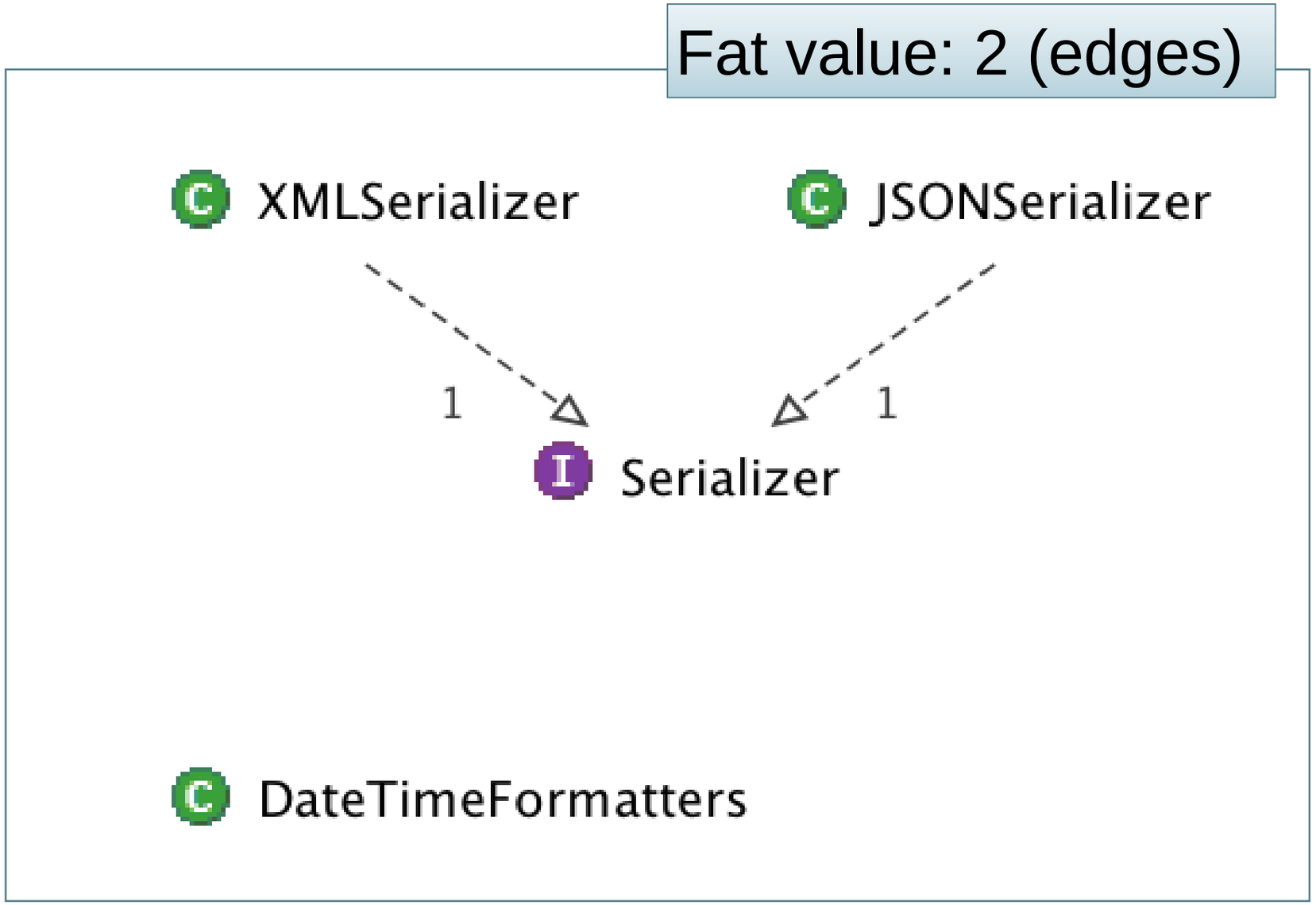


https://structure101.com/help/java/structure101/Content/xs/measuring-complexity.html

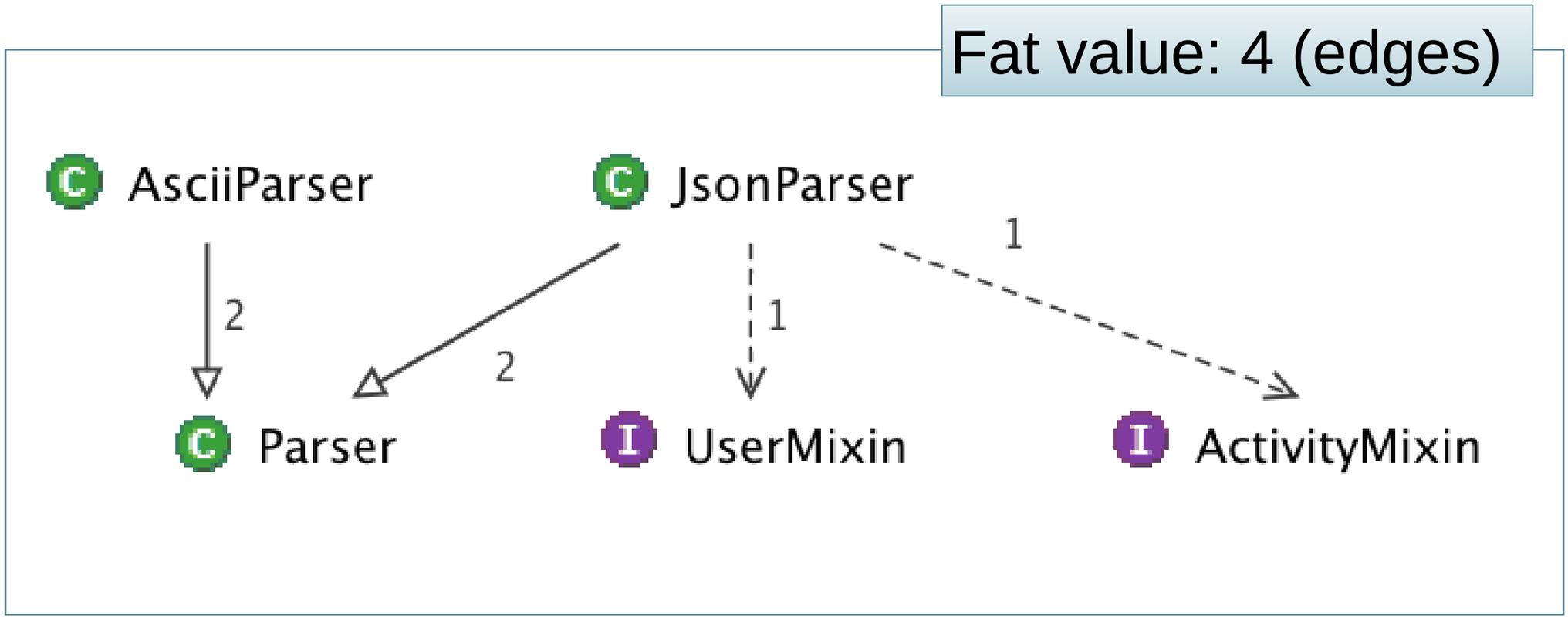
Dependency graph (models)



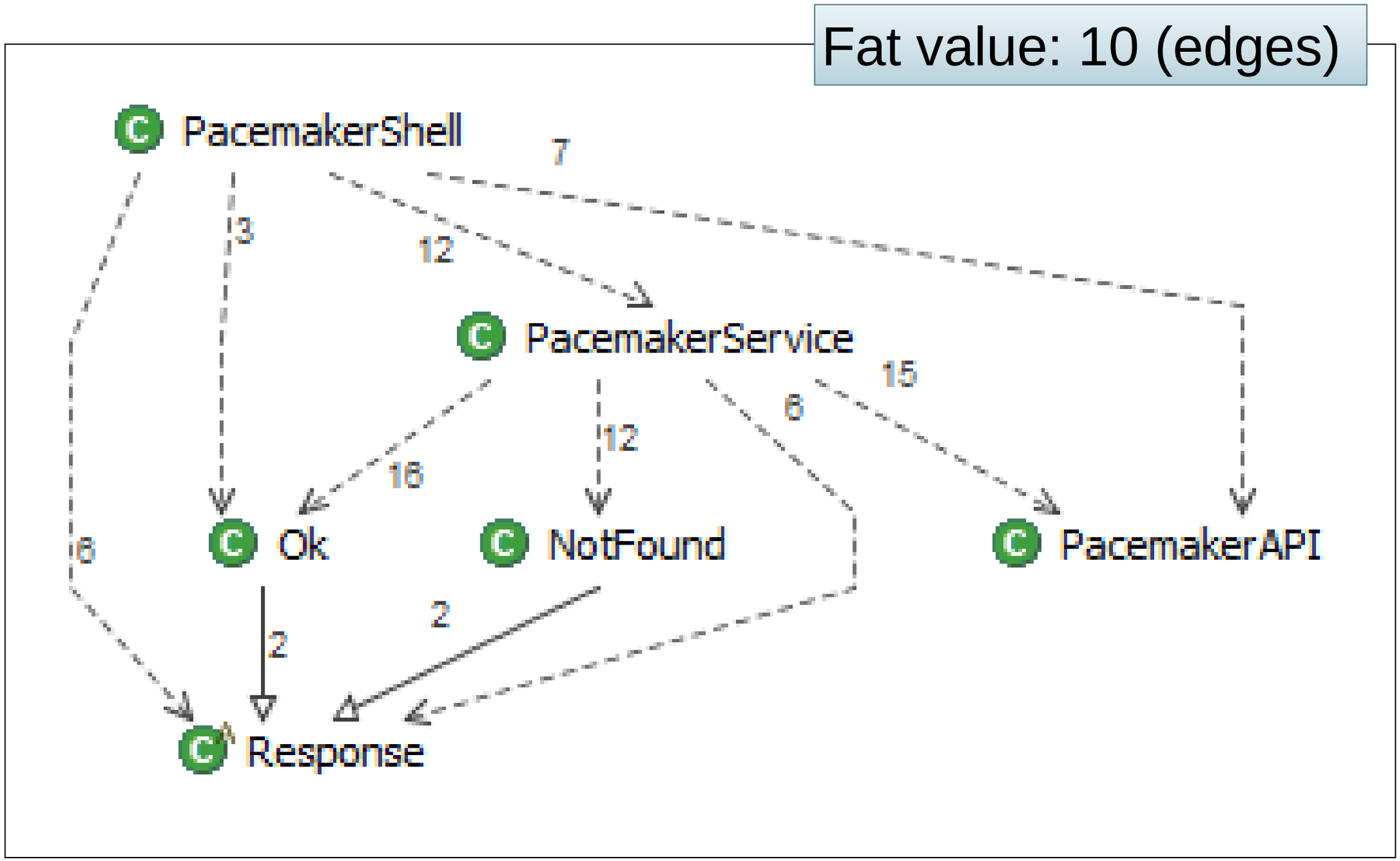
Dependency graph (utils)



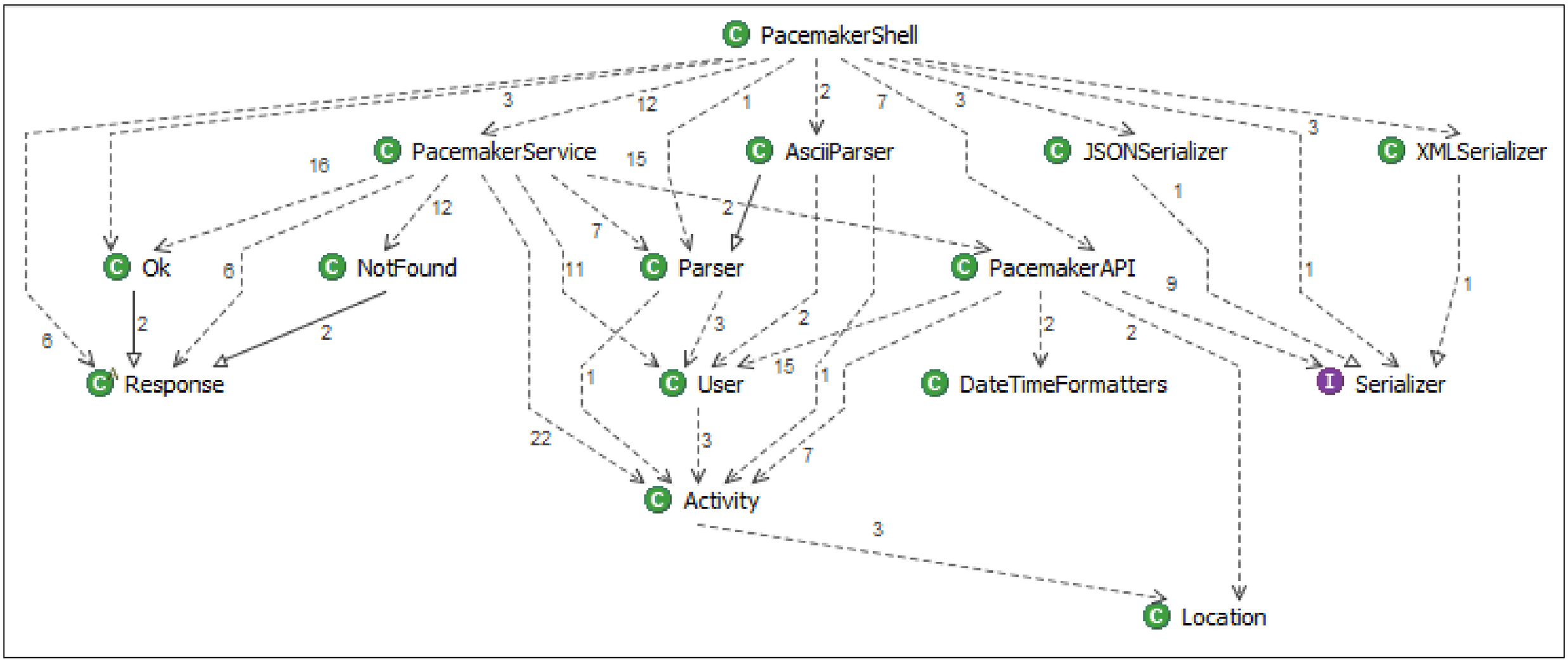
Dependency graph (parsers)



Dependency graph (controllers)



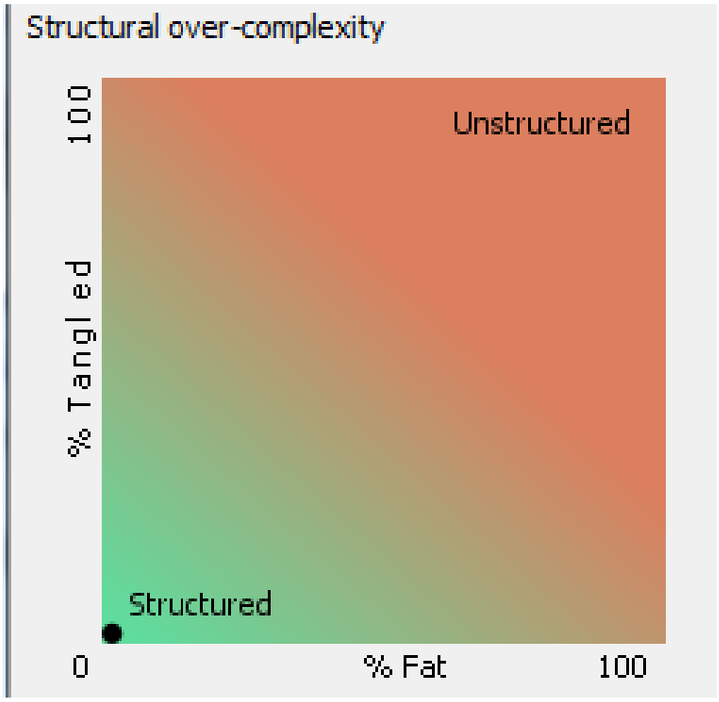
Dependency graph (no externals)



Summary of pacemaker-console-xtend

Flat Tangles

Level	#Items	#Tangles	#Tangled items	Biggest	Degree
Leaf package	4	0	0	0	0%
Jar	1	n/a	n/a	n/a	n/a
Outer class	18	0	0	0	0%



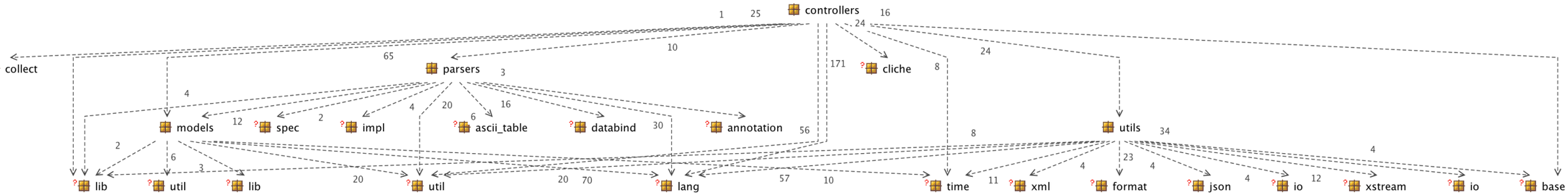
XS breakout by metric (and scope)

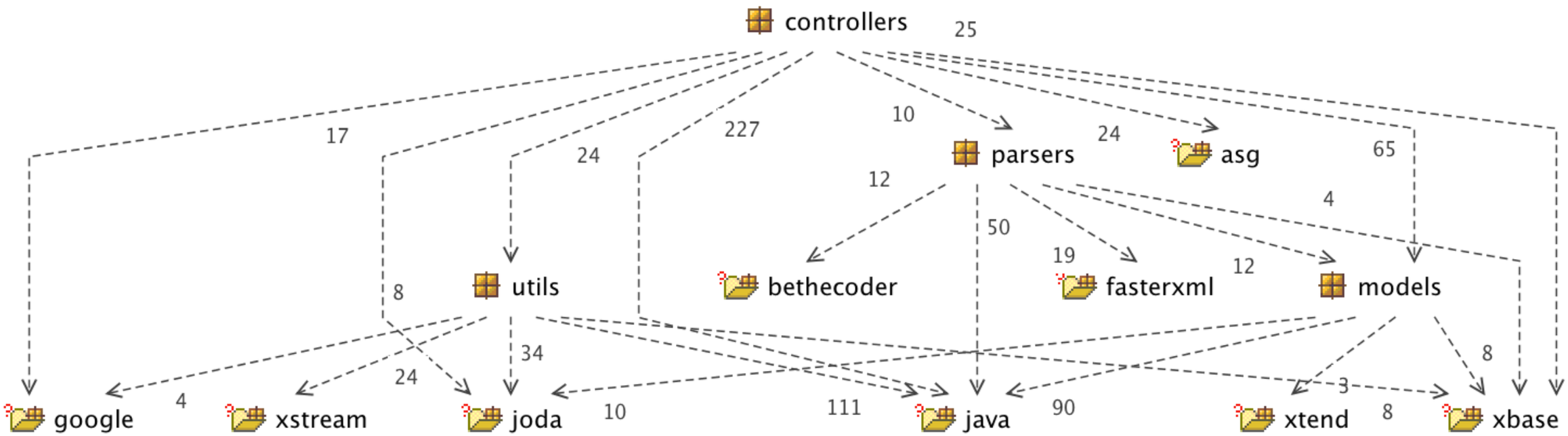
Metric (and scope)	Threshold	#Offenders	Offenses (%)	XS contribution
Tangled (design)	0	0 of 1	0%	0%
Fat (design)	120	0 of 1	0%	0%
Fat (leaf package)	120	0 of 4	0%	0%
Fat (class)	120	0 of 23	0%	0%
Fat (method)	15	2 of 123	2%	100%
Total				**100%**

Items with highest XS: 2

Item	Tangled	Fat	Size	XS
pacemaker-console-xtend.models.Activity.equals(java.lang.Object):boolean		17	128	15
pacemaker-console-xtend.models.User.equals(java.lang.Object):boolean		16	118	7

Dependency graph (with externals)




controllers
25
17
24
227
10
24
65
parsers
asg
12
4
50
19
12
bethecoder
fasterxml
models
utils
8
34
24
4
8
3
8
10
111
90
google
xstream
joda
java
xtend
xbase

Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see http://creativecommons.org/licenses/by-nc/3.0/