

Software Components

Why - How - What

Stephan Hagemann - 2016-11-03 - RUG-B

WRITE BETTER SOFTWARE

@shageman

***WHAT* IS A COMPONENT?**

Labeled Content

+

Explicit Dependencies

Labeled Content

+

Explicit Dependencies



Labeled Content

+

Explicit Dependencies





@shageman



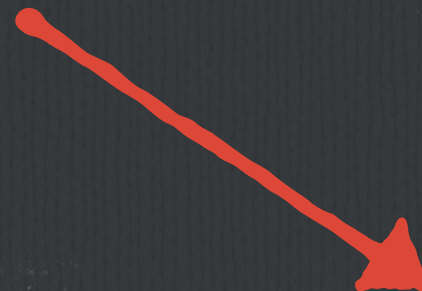


@shageman

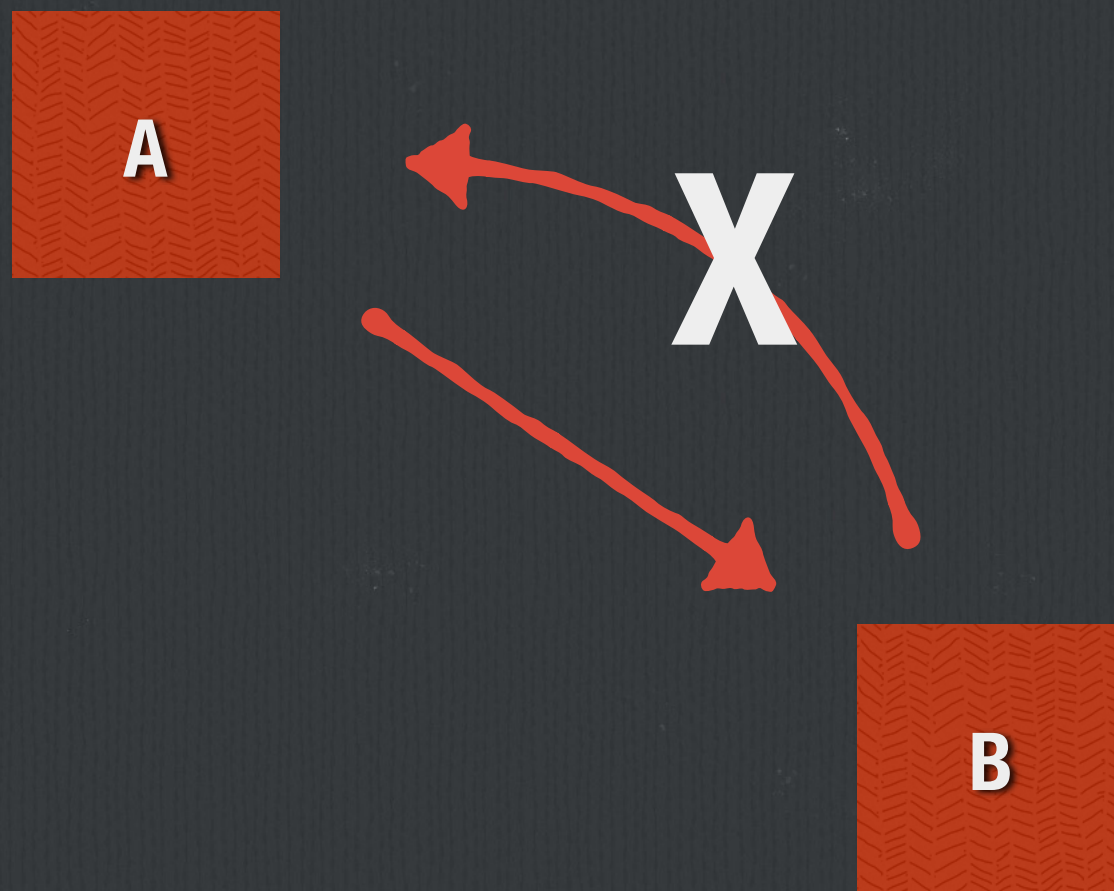


@shageman

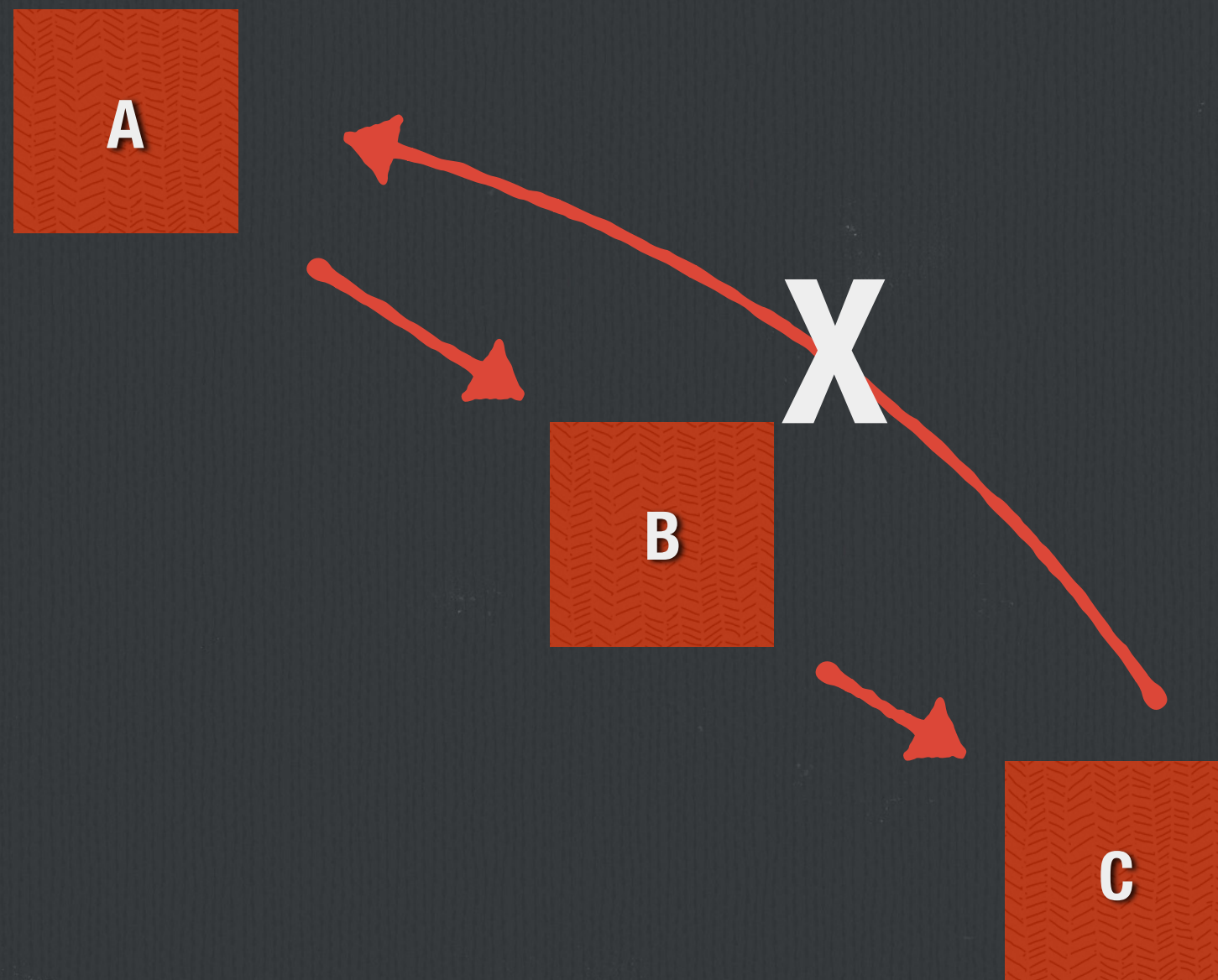
A depends on B ...



thus B can not depend on A



and this doesn't work either



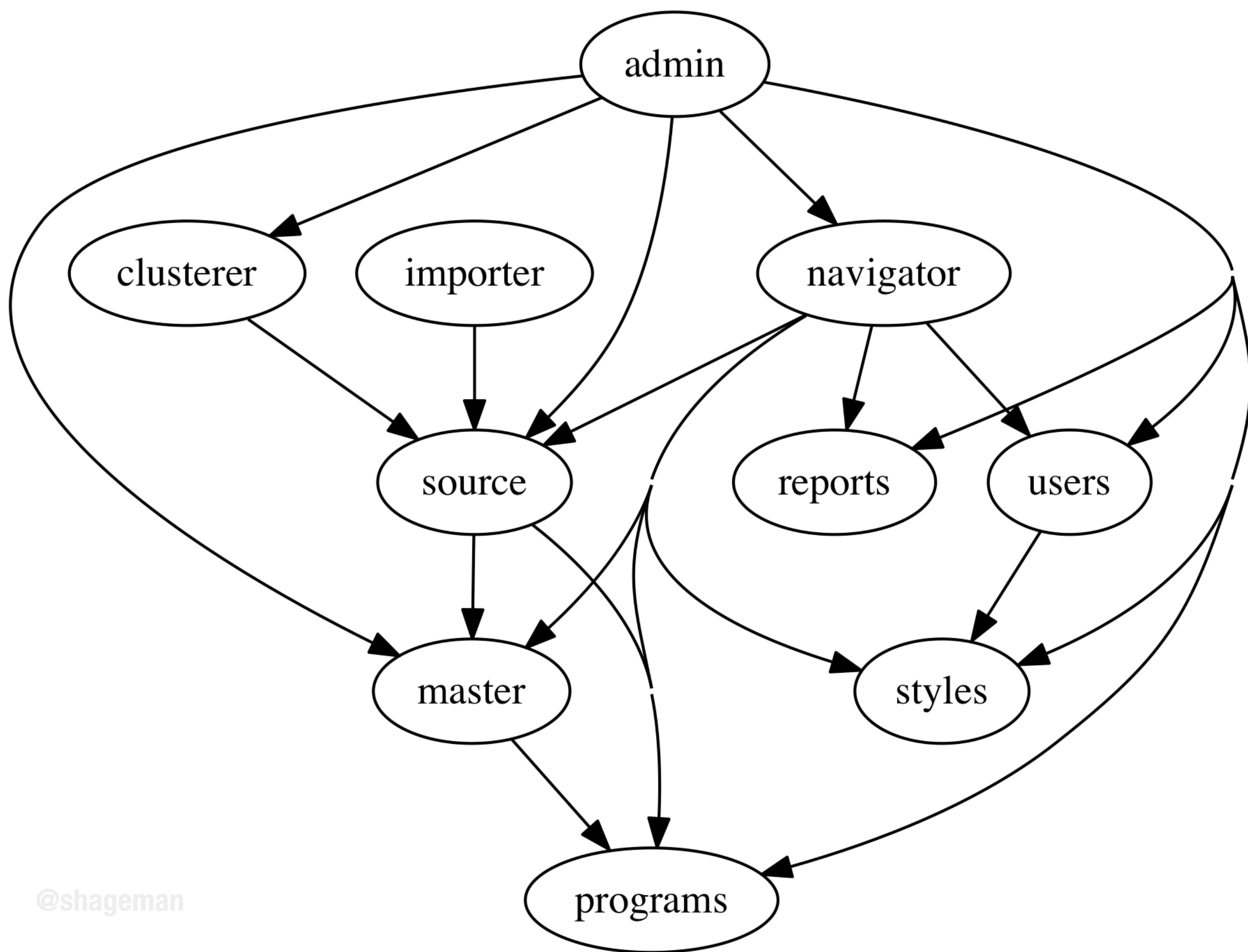
X and Y as the same thing



Uncle Bob says

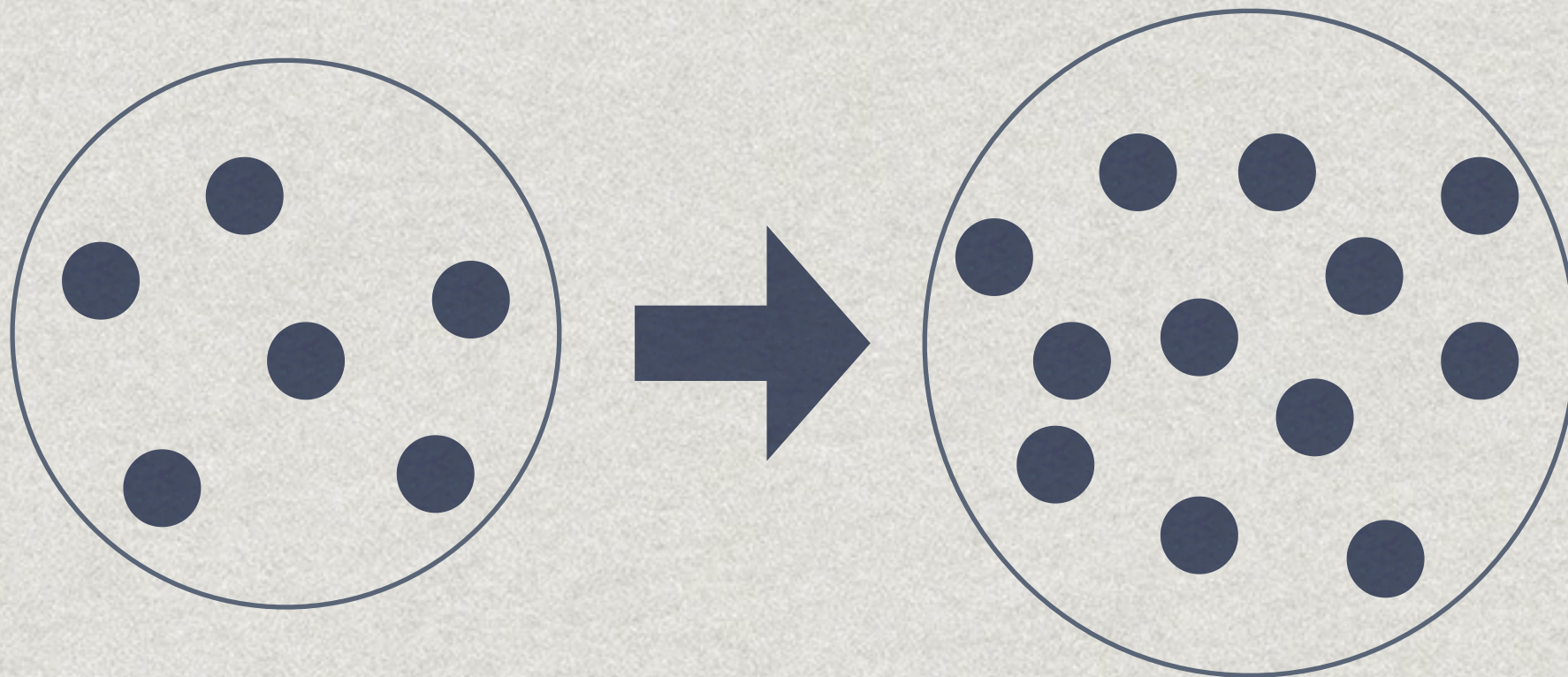
... I like to draw these lines. Architecture is the art of drawing lines. With the interesting rule that once you have drawn the lines, all the dependencies that cross that line go in the same direction...

☐ **This is why objects are typically not components...**



***WHY* COMPONENTS?**

Growing size

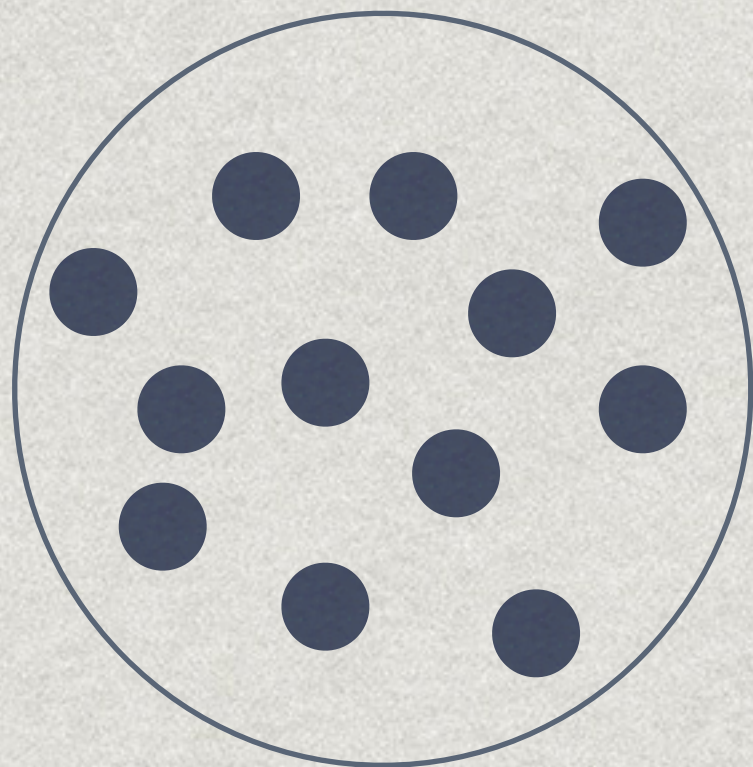


Exploding complexity

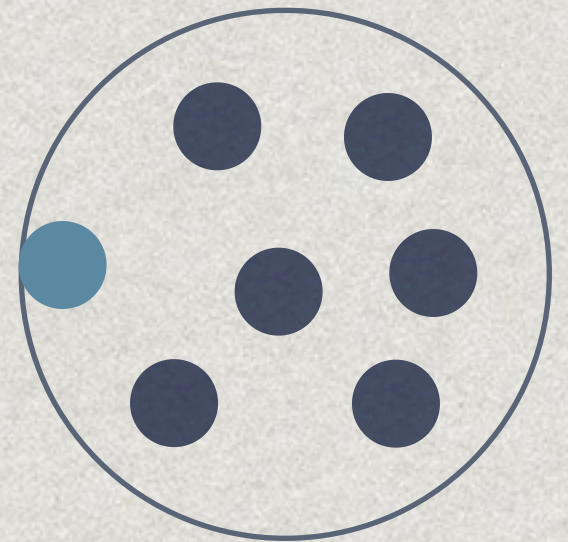
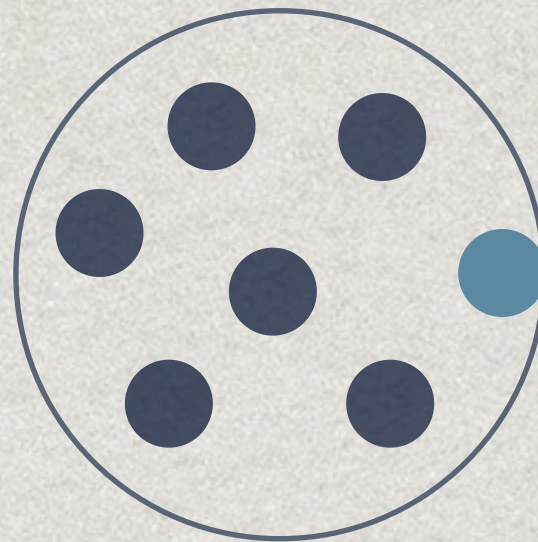
**POSSIBLE
INTERACTIONS**

SIZE

Effect of structure



vs

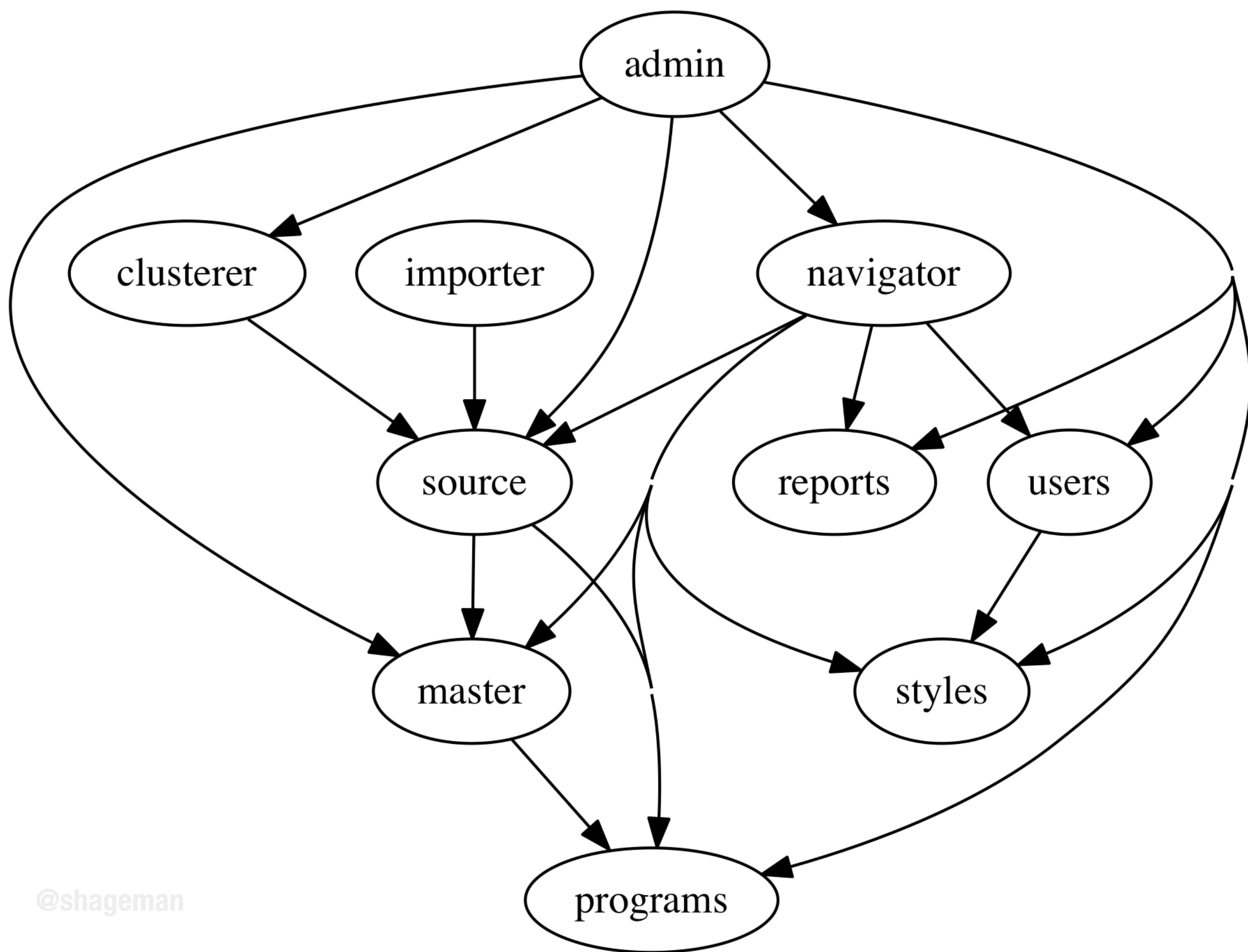


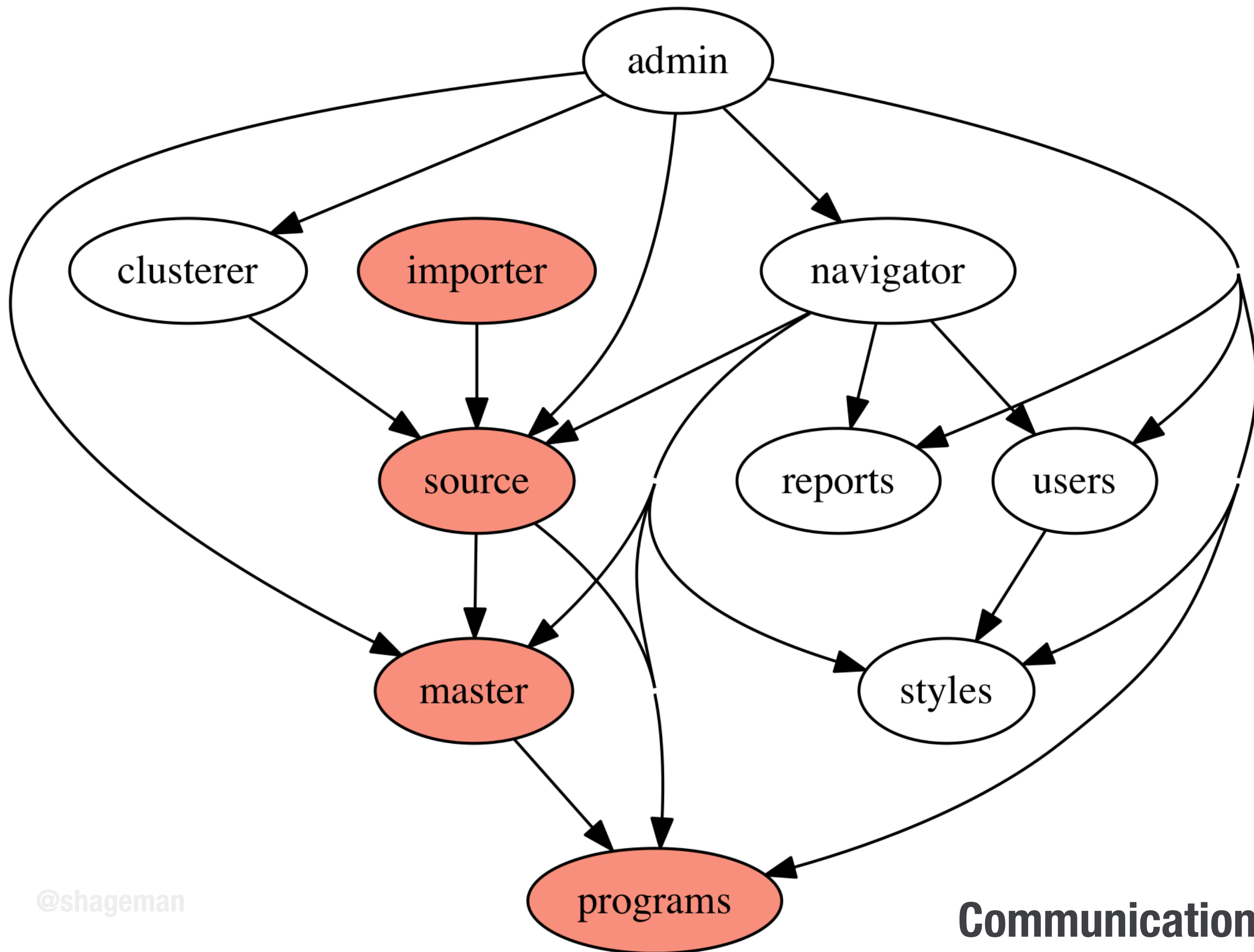
Reduced exploding complexity

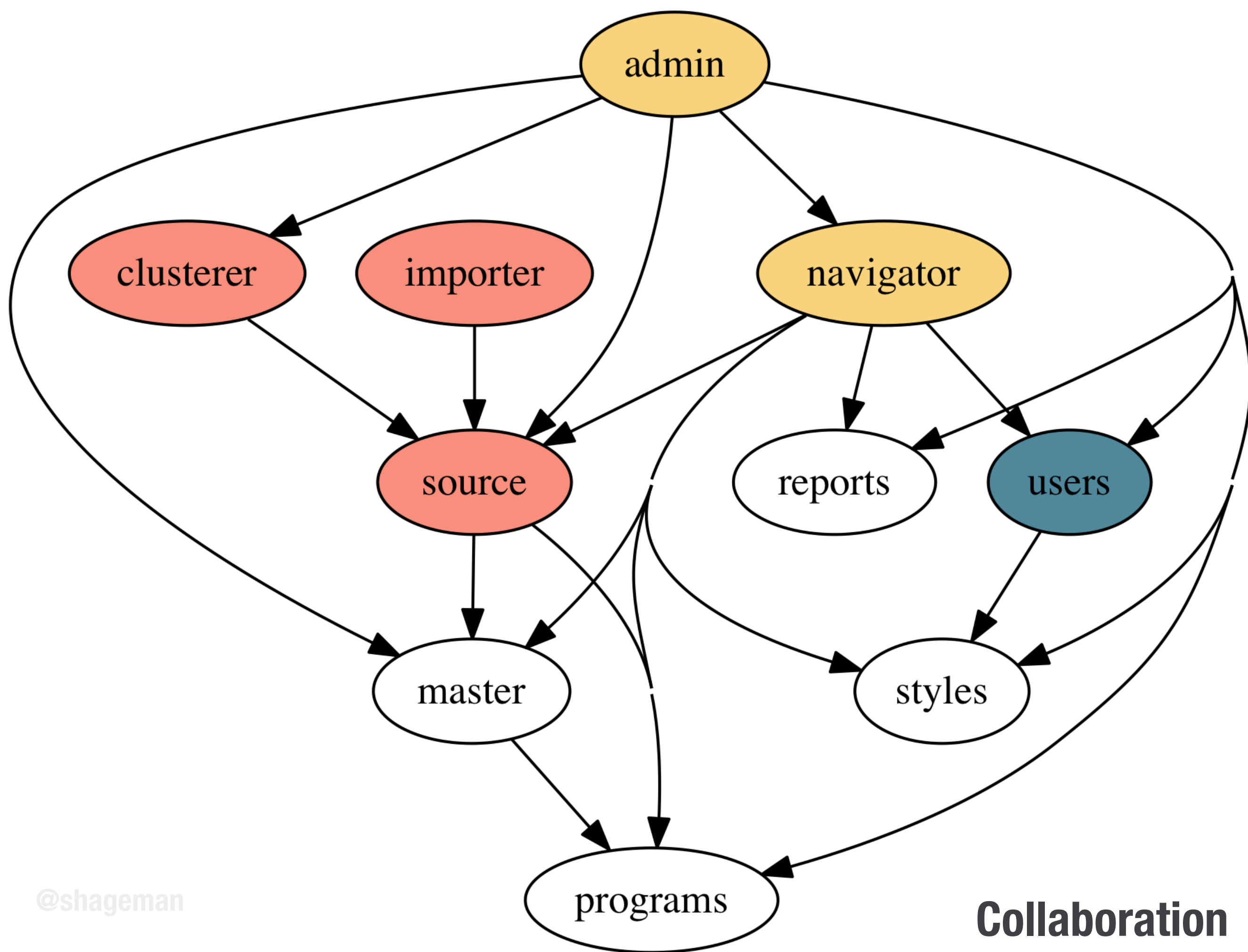
**POSSIBLE
INTERACTIONS**

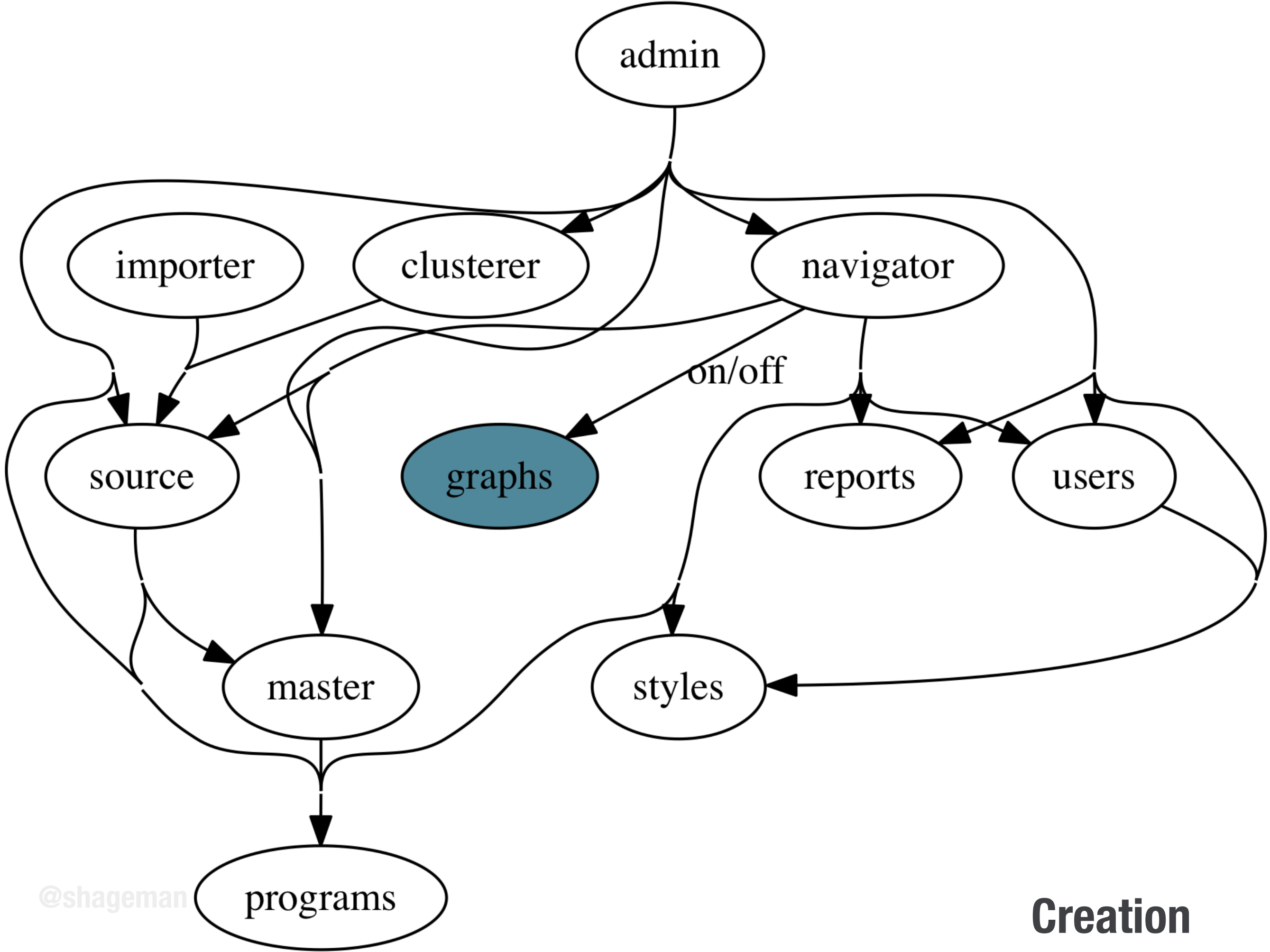


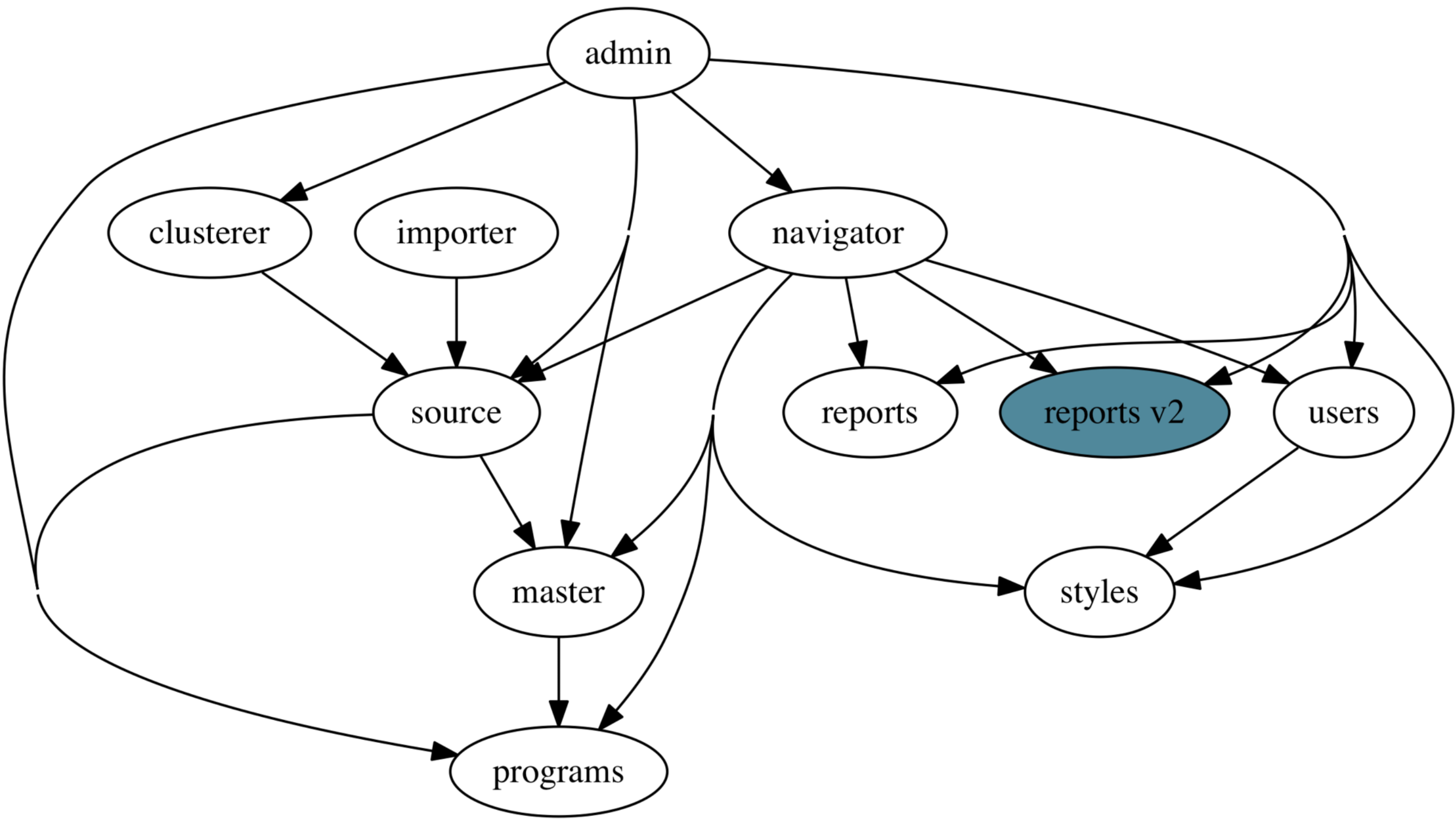
SIZE

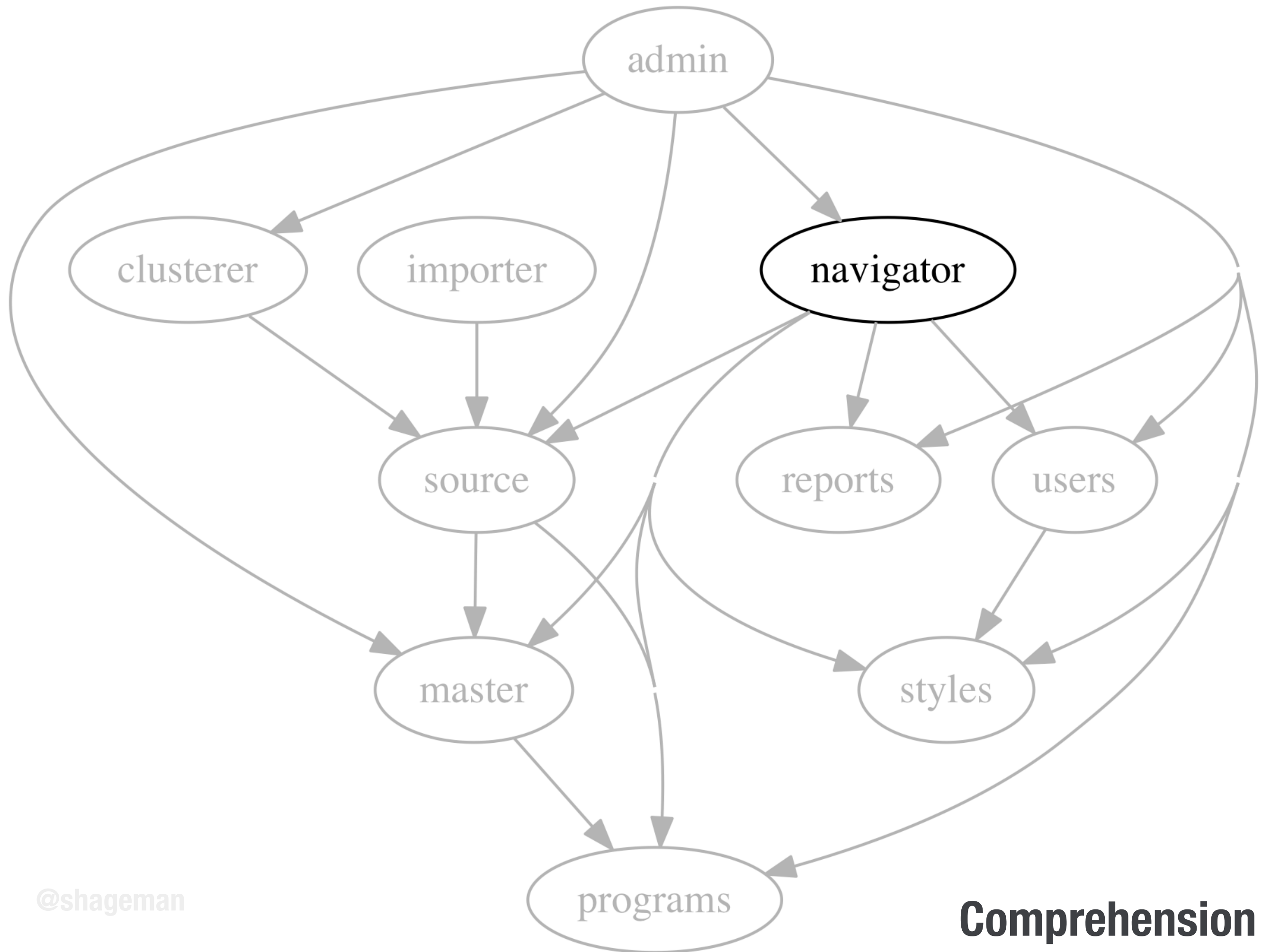






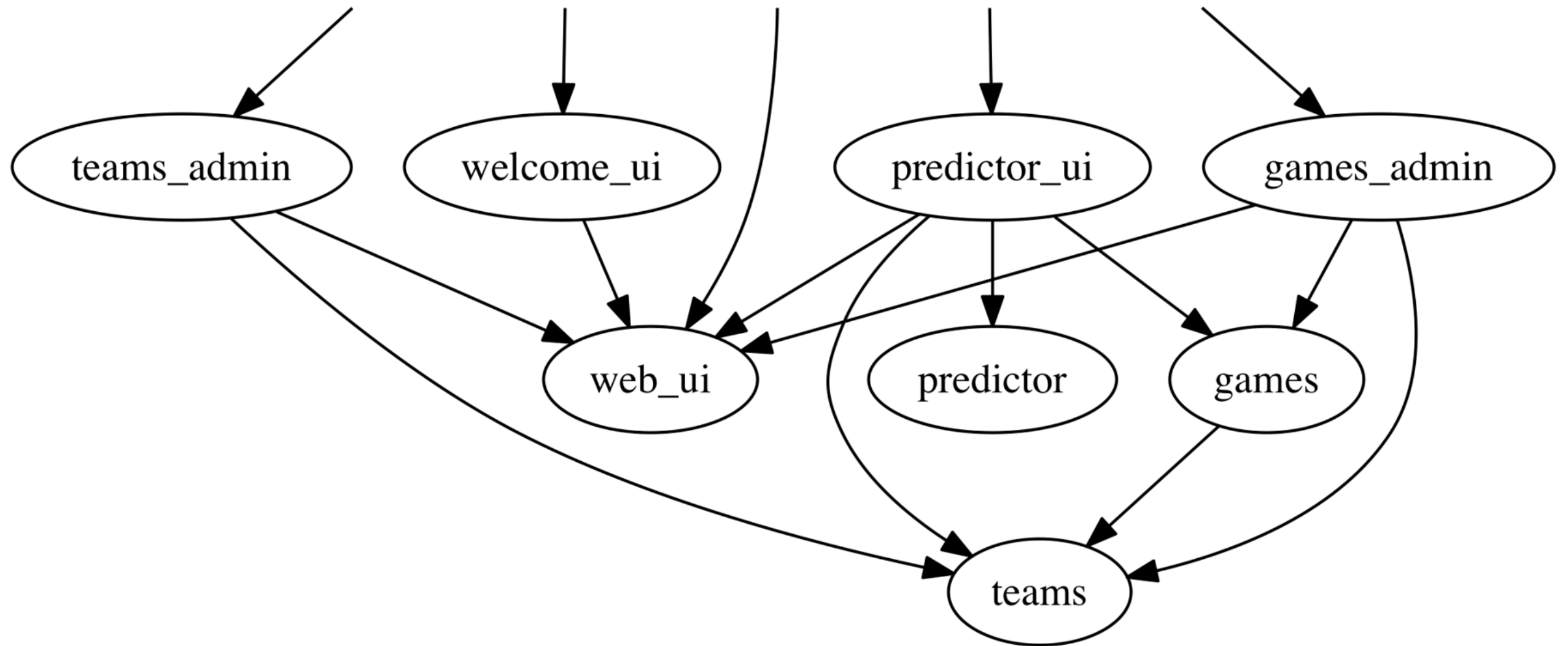






HOW TO IMPLEMENT COMPONENTS?

sportsball



Code!!

And in other languages?

- ☐ **Ruby/Rails: Gems + Engines**
- ☐ **Java: Modules**
- ☐ **Go: packages**
- ☐ **Javascript: packages (in npm lingo)**
- ☐ **C: file compiling + linking**

HOW TO DETERMINE WHAT SHOULD BE A COMPONENT?

SUMMARY

WHAT

Components =

Labeled Content

+

Explicit Dependencies

WHY

Components improve...

- ☐ **communication**
- ☐ **creation**
- ☐ **collaboration**
- ☐ **maintenance**
- ☐ **comprehension**
- ☐ **complexity**

HOW

In Ruby and Rails...

- ☐ **gems and engines can be used as components**

COMPONENT BASED RAILS APPLICATIONS

Large domains under control



STEPHAN HAGEMANN

CASE STUDIES BY RYAN PLATTE, BEN SMITH, AND ENRICO TEOTTI

295 READERS 254 PAGES 53,358 WORDS

ENGLISH PDF EPUB MOBI APP

Component-Based Rails Applications

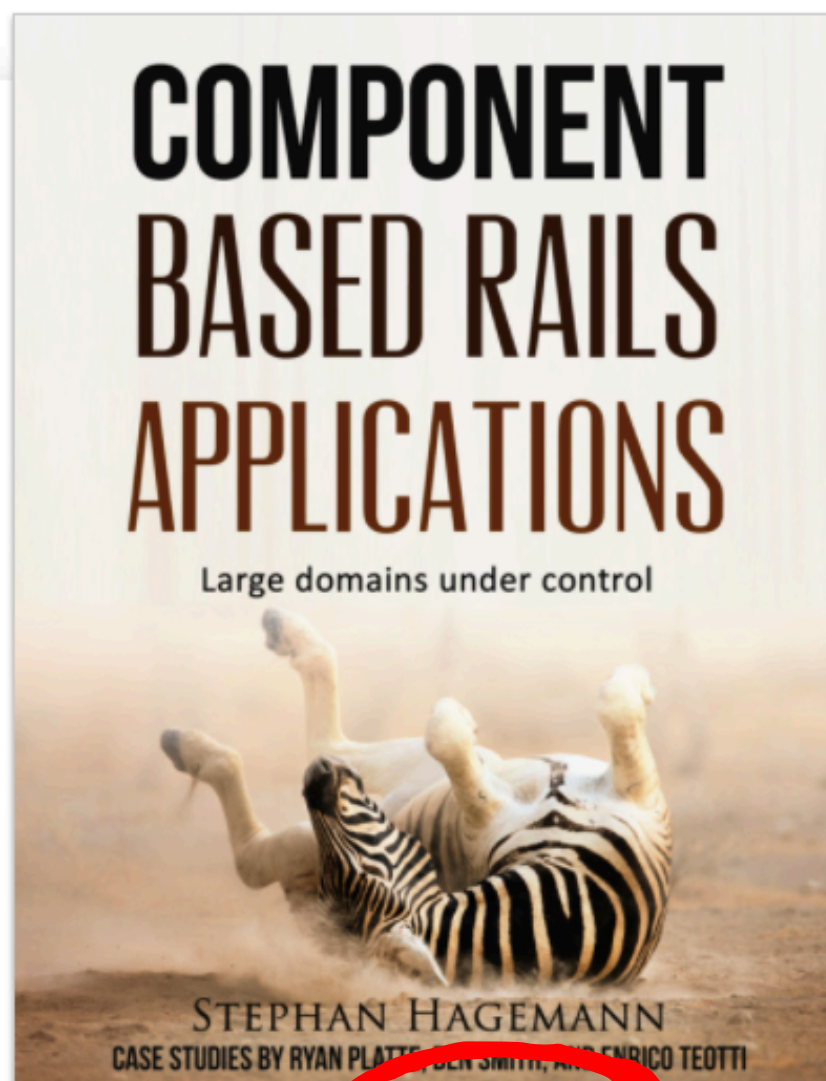
Large domains under control

Stephan Hagemann, Ryan Platte, Benjamin Smith and Enrico Teotti

Components put you in control of the destiny of your Rails application. Decompose your existing application to make it more maintainable and to set you up for a bright development future. Assemble your new application from smaller pieces that you can develop concurrently, combine reliably, and test independently.

FREE SAMPLE

Download



This book is 100% complete

UPDATED OCTOBER 30, 2016

\$19.99
MINIMUM

\$24.99
SUGGESTED

You pay (USD) ?

\$ 24.99

Authors earn

\$ 21.99

Add Ebook to Cart

@shageman

<https://leanpub.com/cbra/c/rugb>

Discount Code!

Further reading

- ☐ <https://leanpub.com/cbra>
- ☐ <http://cbra.info>
- ☐ <https://github.com/shageman>
- ☐ <http://appcontinuum.io>

THANKS!

<https://leanpub.com/cbra/c/rugb>

@shageman