

# Visualizing Complex Graph Statistics

with d3 Chord Diagrams

# Who are you exactly?

And why should I pay attention?

*Founder & CEO at*



**nodejitsu**

*I'm on the Internet!*

**@indexzero**

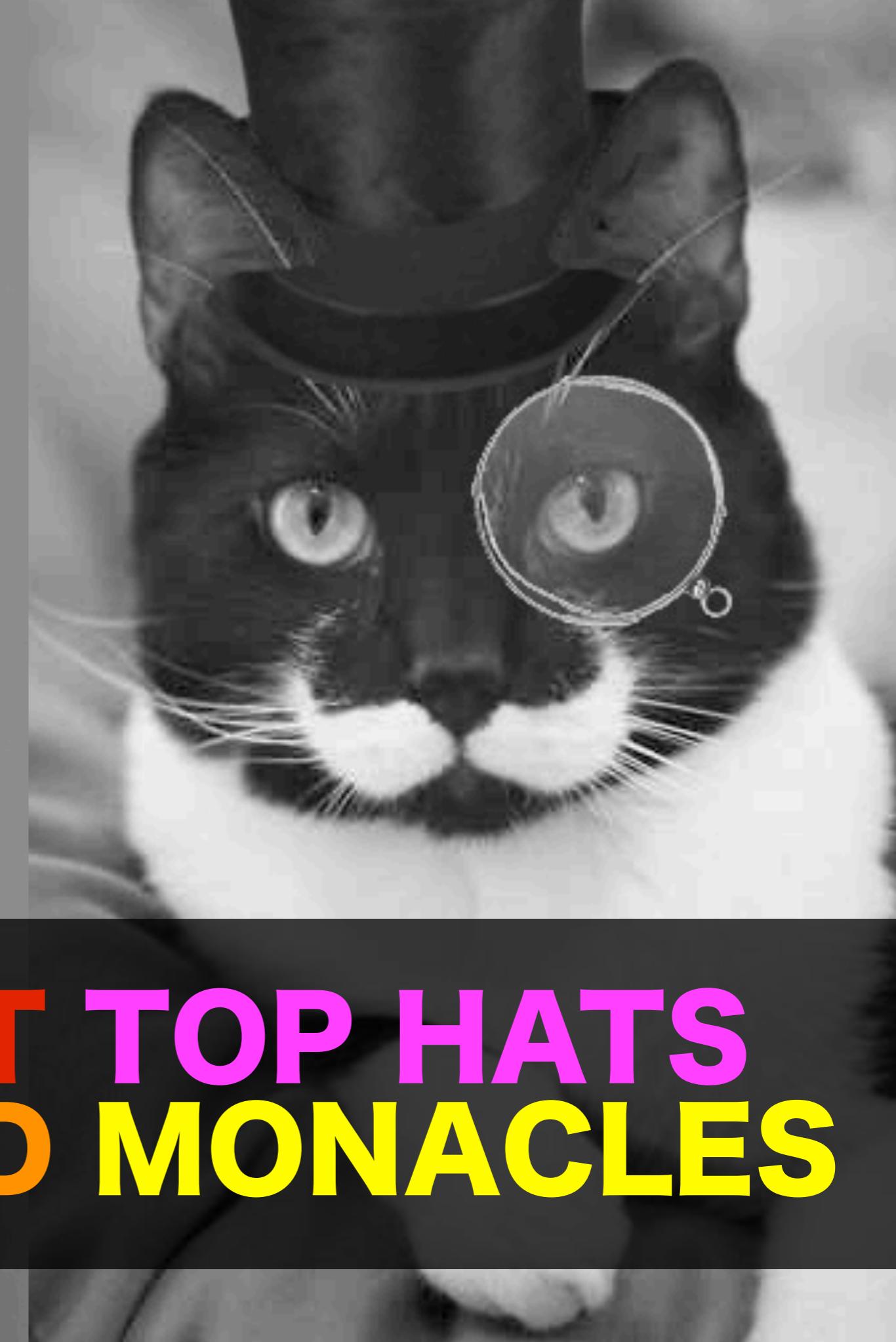


# **WARNING: shameless plug**

for  **nodejitsu**

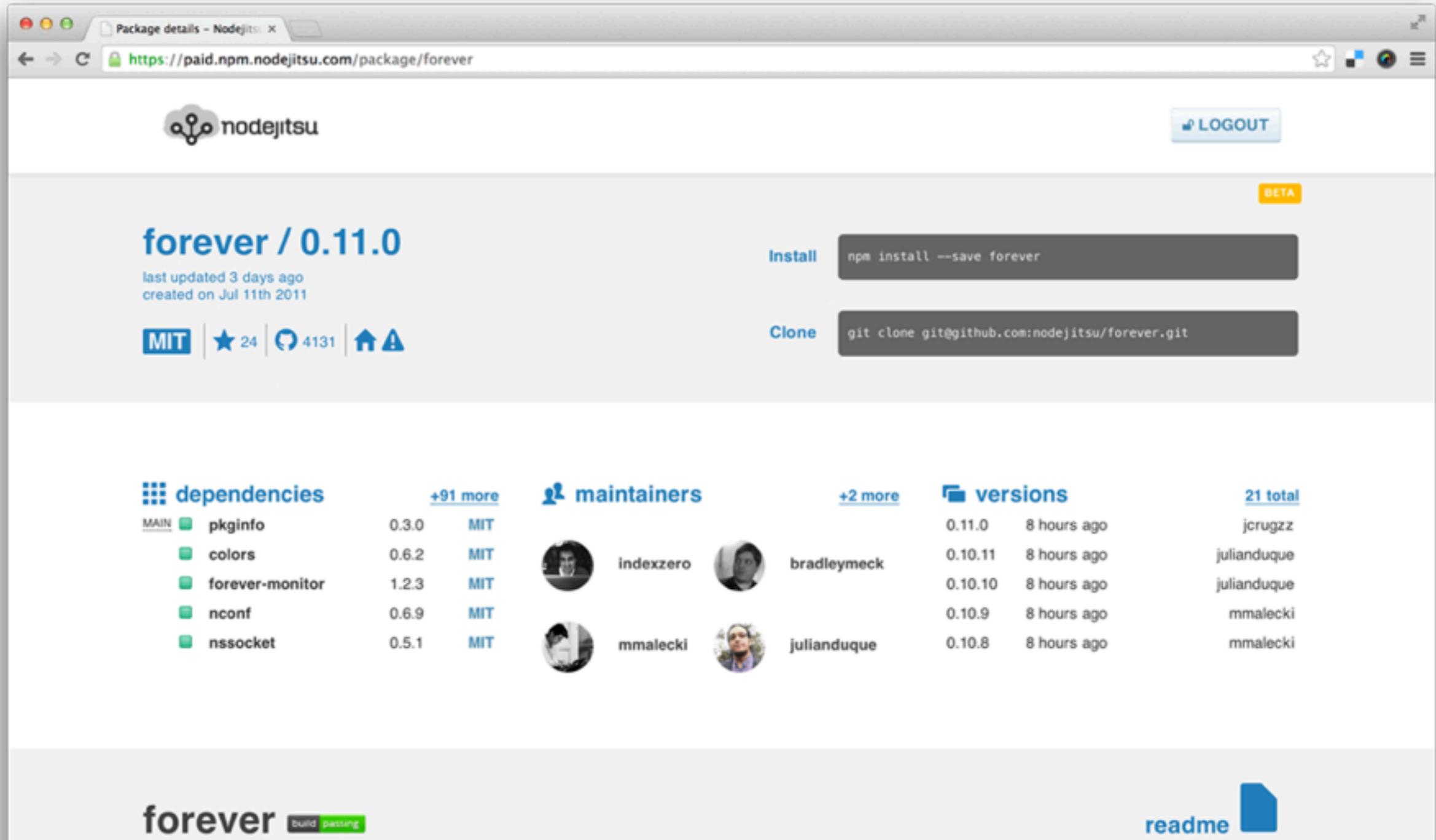
---

I will be brief. I promise.



**NOT TOP HATS  
AND MONACLES**

# Github for production Javascript



The screenshot shows a web browser displaying the Nodejitsu package details page for the 'forever' package. The URL in the address bar is <https://paid.npm.nodejitsu.com/package/forever>. The page features the Nodejitsu logo at the top right. The main content area shows the package name 'forever / 0.11.0' in large blue text, with a 'last updated 3 days ago' and 'created on Jul 11th 2011' message below it. To the right, there are 'Install' and 'Clone' buttons with their respective commands: 'npm install --save forever' and 'git clone git@github.com:nodejitsu/forever.git'. A 'BETA' label is visible in the top right corner. Below the package name, there are links for 'MIT' license, '24' stars, '4131' forks, and two other icons. The page also displays sections for 'dependencies', 'maintainers', and 'versions'.

**forever / 0.11.0**

last updated 3 days ago  
created on Jul 11th 2011

MIT | ★ 24 | 4131 |  

**dependencies**

MAIN	pkinfo	0.3.0	MIT
	colors	0.6.2	MIT
	forever-monitor	1.2.3	MIT
	nconf	0.6.9	MIT
	nssocket	0.5.1	MIT

[+91 more](#)

**maintainers**

	indexzero	bradleymeck
		
		
	mmalecki	
		julianduque

[+2 more](#)

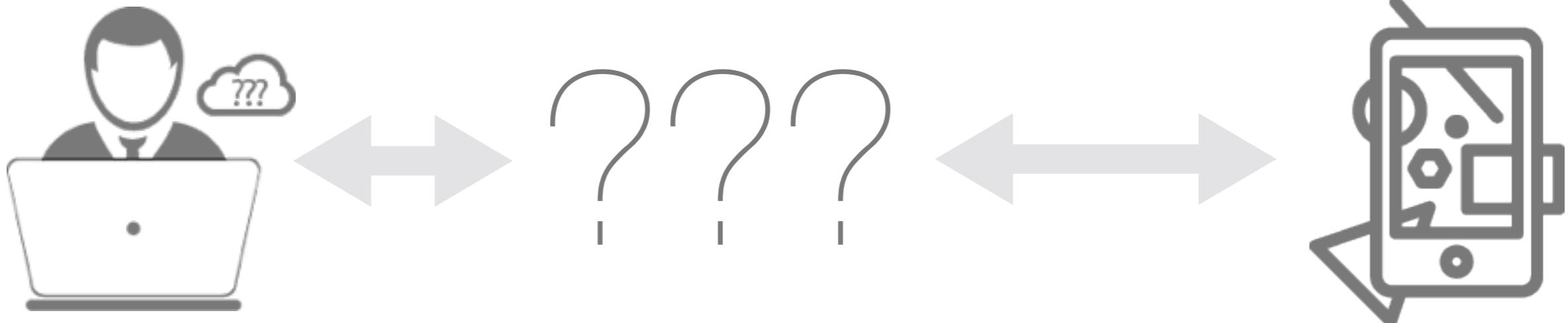
**versions**

21 total	0.11.0	8 hours ago	jcrugzz
	0.10.11	8 hours ago	julianduque
	0.10.10	8 hours ago	julianduque
	0.10.9	8 hours ago	mmalecki
	0.10.8	8 hours ago	mmalecki

**forever** 

**readme** 

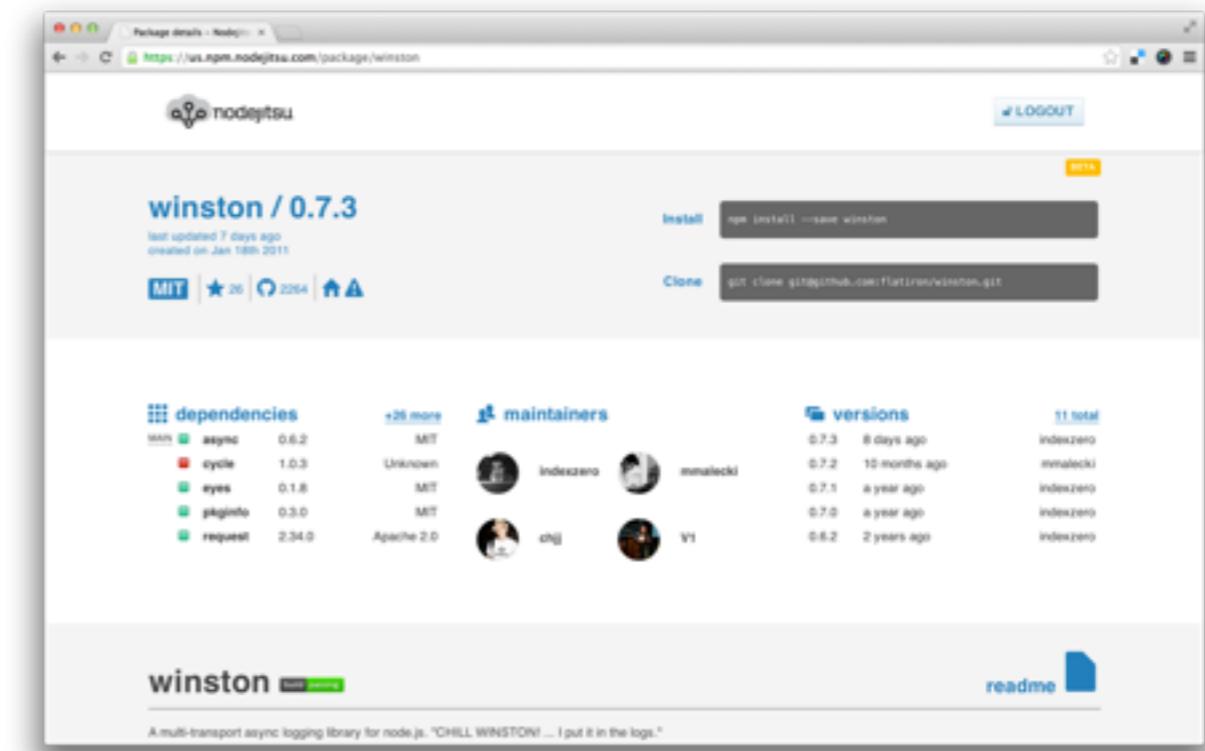
# What problem do we solve for **npm**?

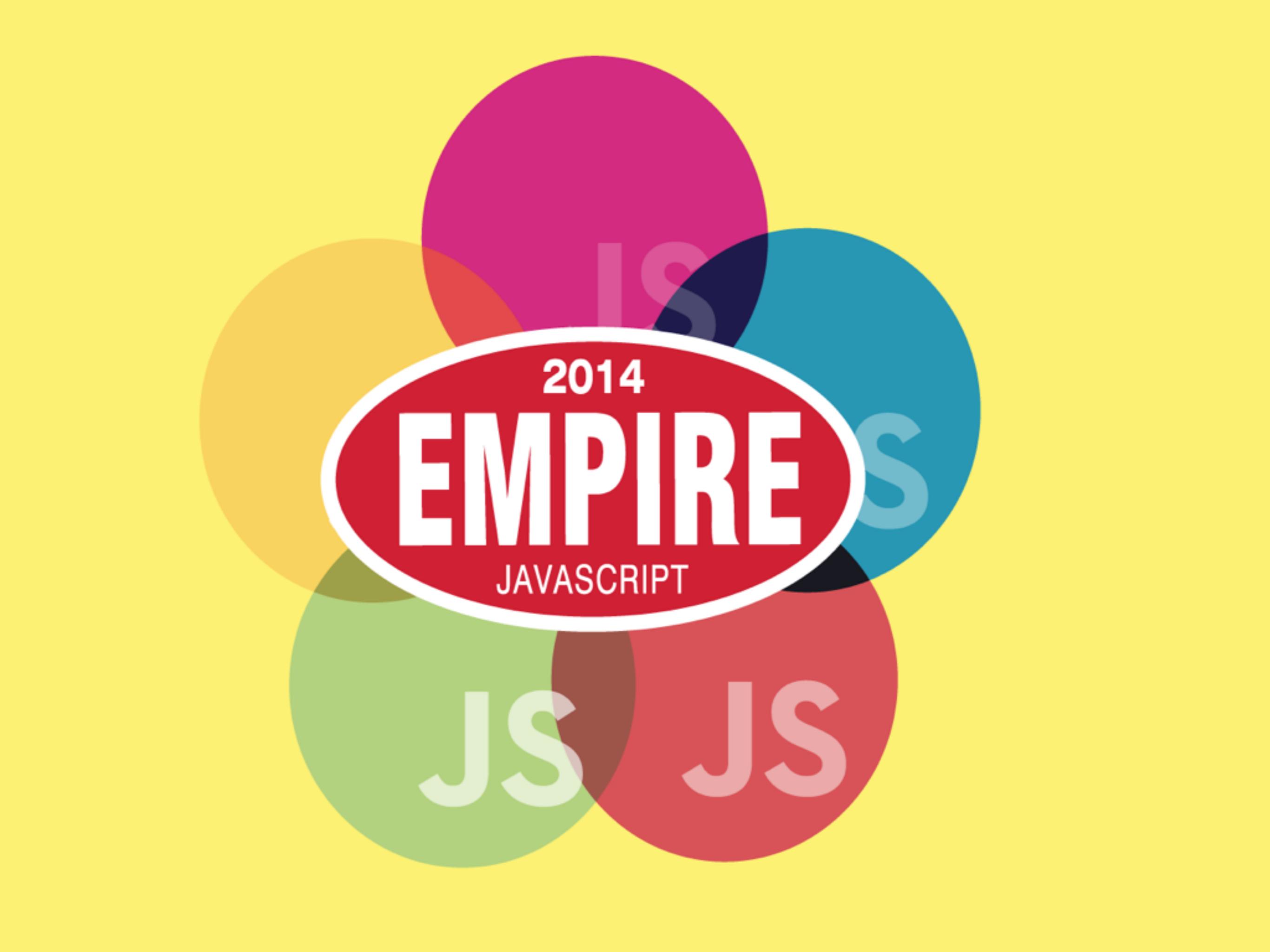


**YOU** have **QUESTIONS** about your **MODULES**

Stop wasting time and answer questions

- *Where do I store my private code?*
- *Is my Node.js application **secure**? Against what vulnerabilities?*
- *How do I **deploy** my application?*
- *Am I at **legal risk** because of Open Source licenses?*
- *What module should I use to do <feature> in my app?*
- *Is this Open Source module **good**?*





2014  
**EMPIRE**  
JAVASCRIPT

JS

JS

# EMPIRE NODE

**One Day. One Track. All Node.js**

**Tickets on Sale!**

**October 27th, 2014**

**<http://empirenodenode.org>**



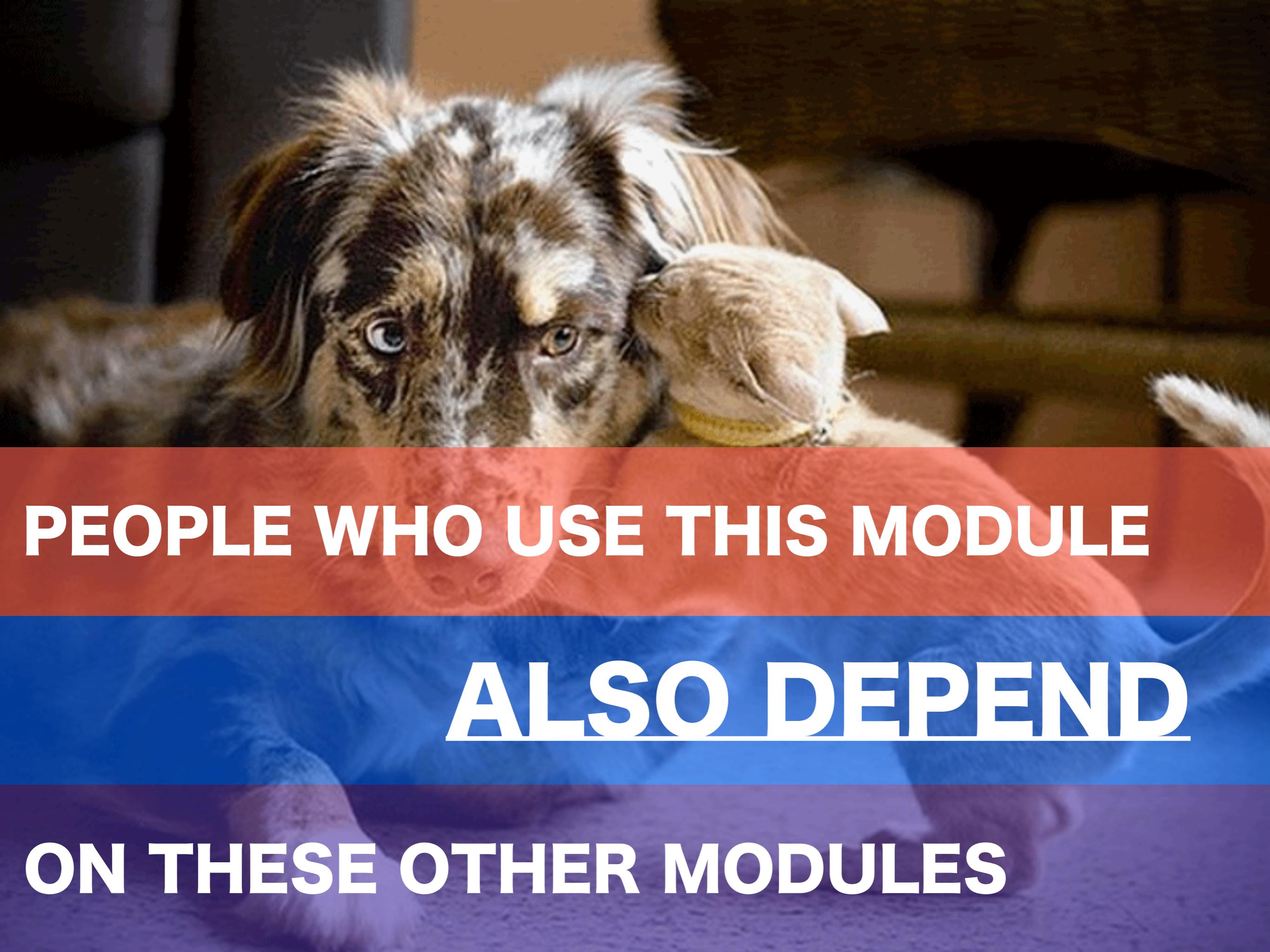
MODULE AUTHORS  
AND DEVELOPERS  
HAVE QUESTIONS  
ABOUT MODULES  
BEING DEVELOPED



SCIENCE CAT MUST KNOW

WHAT QUESTION ARE  
WE ANSWERING?





**PEOPLE WHO USE THIS MODULE  
ALSO DEPEND  
ON THESE OTHER MODULES**

**GRAPHS**

**RULE**

**EVERYTHING**

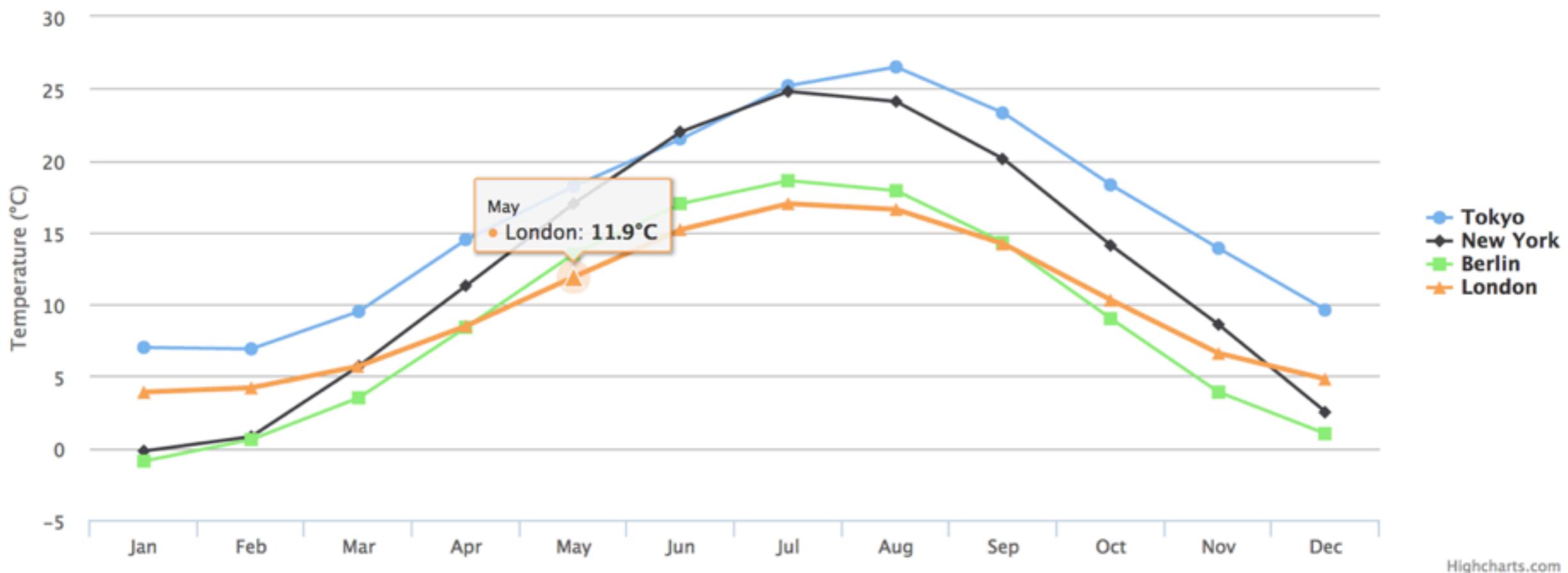
**AROUND**

**ME**

# NOT THIS KIND

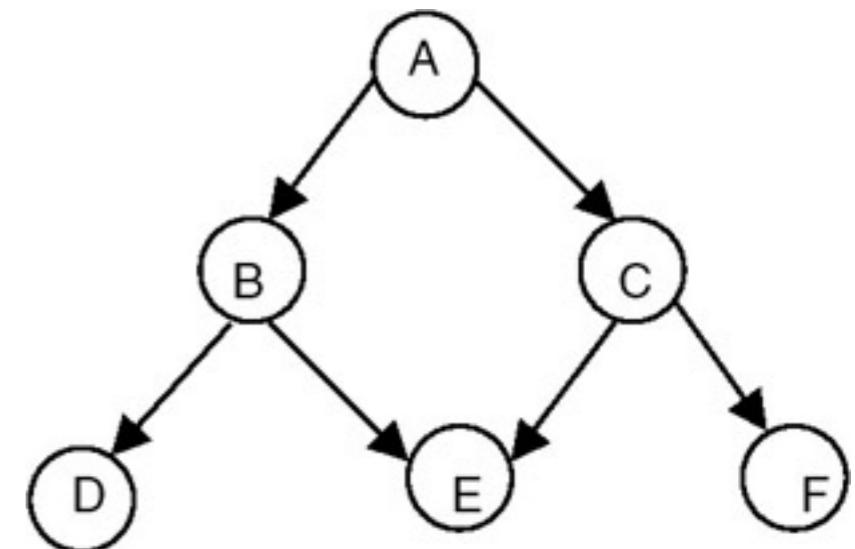
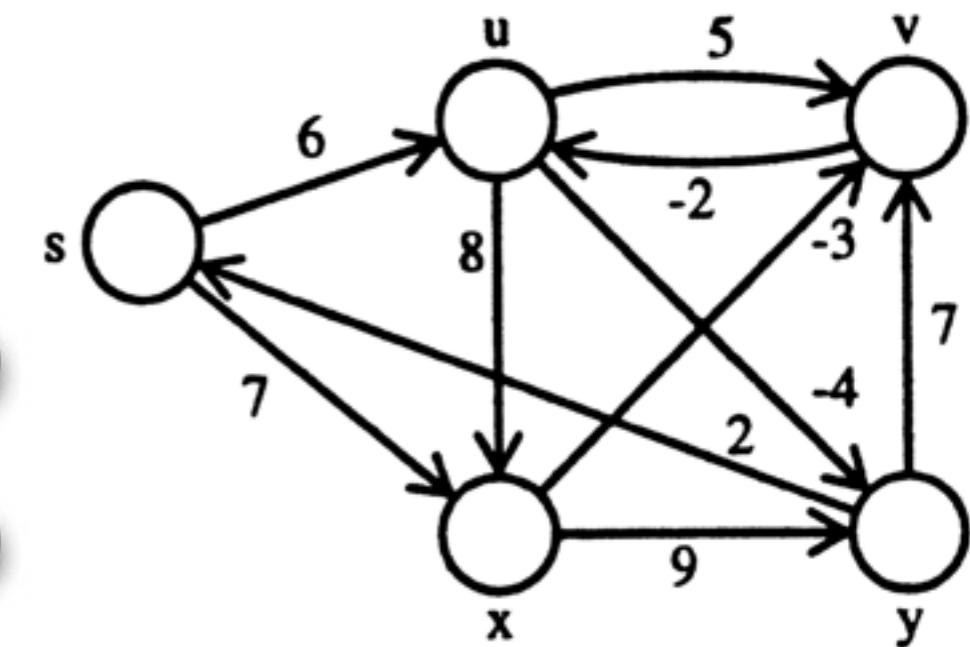
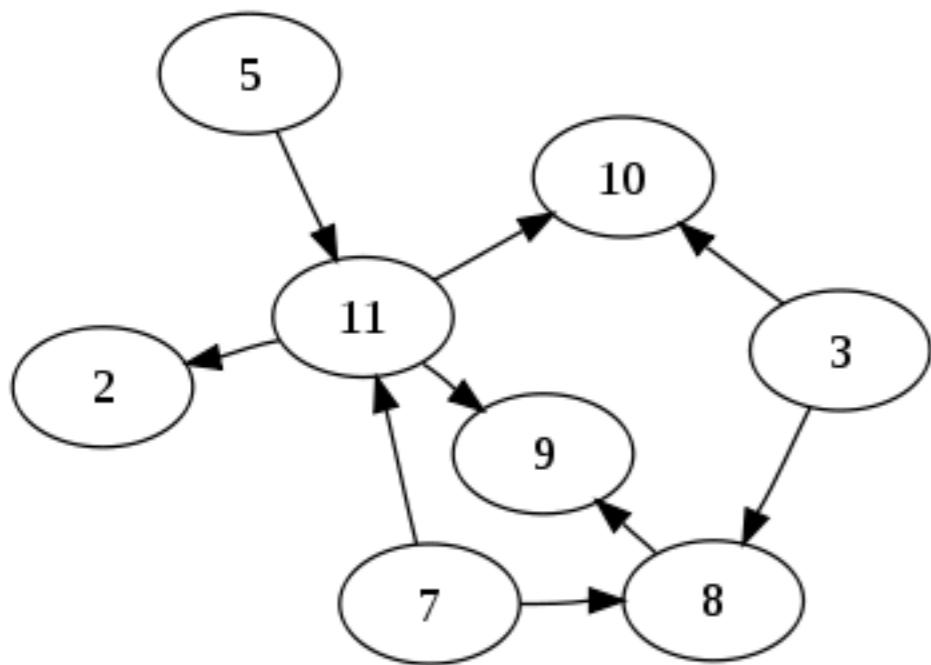
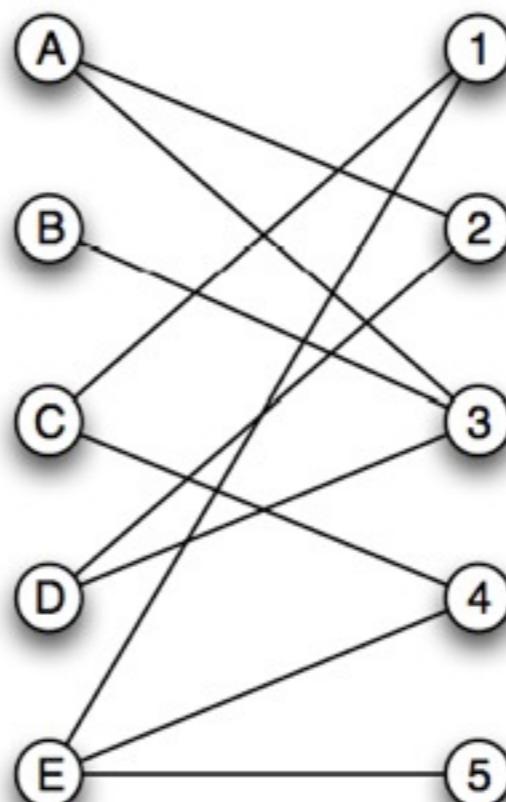
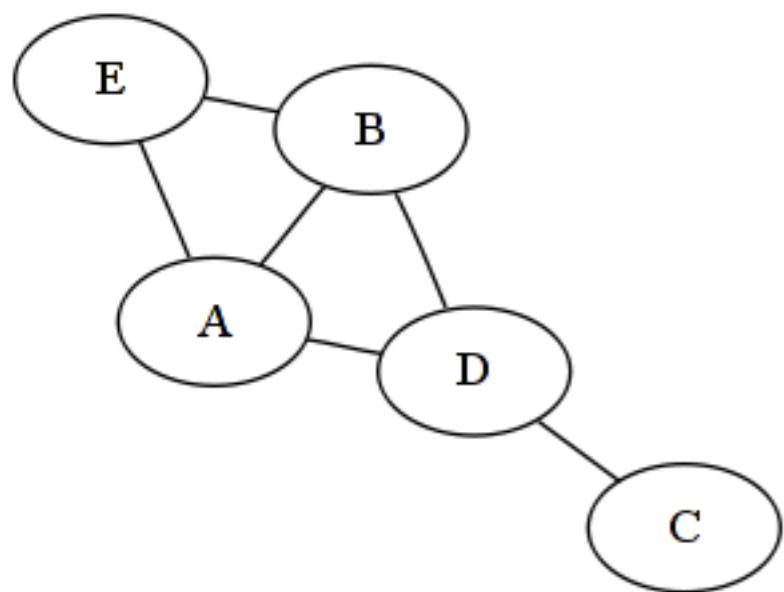
Monthly Average Temperature

Source: WorldClimate.com



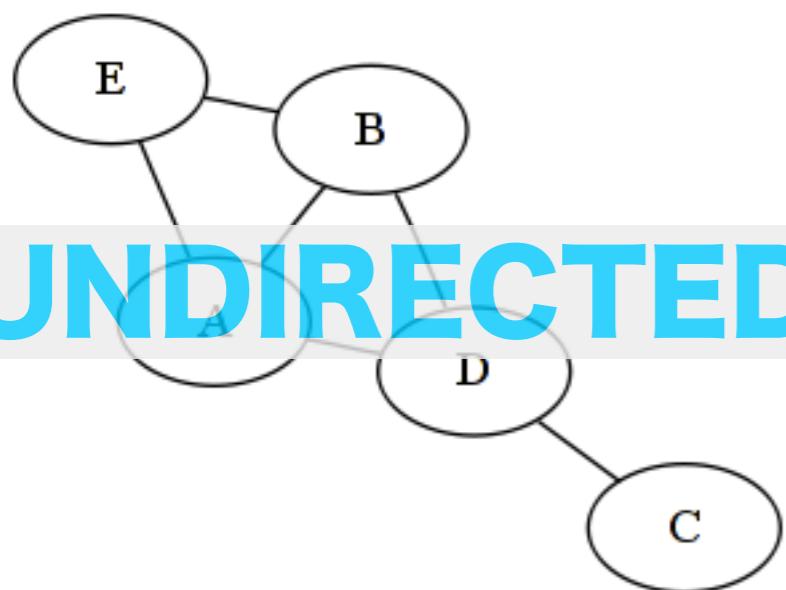
# OF GRAPH

# THESE KINDS

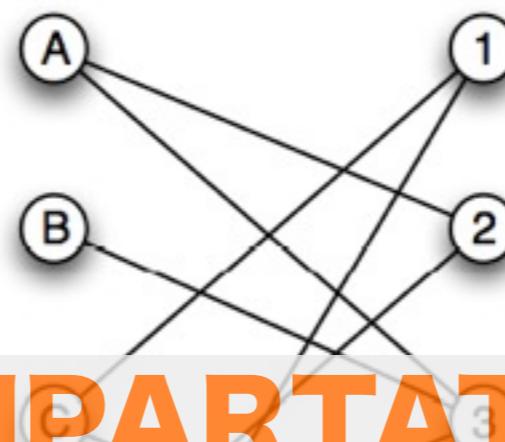


# OF GRAPHS

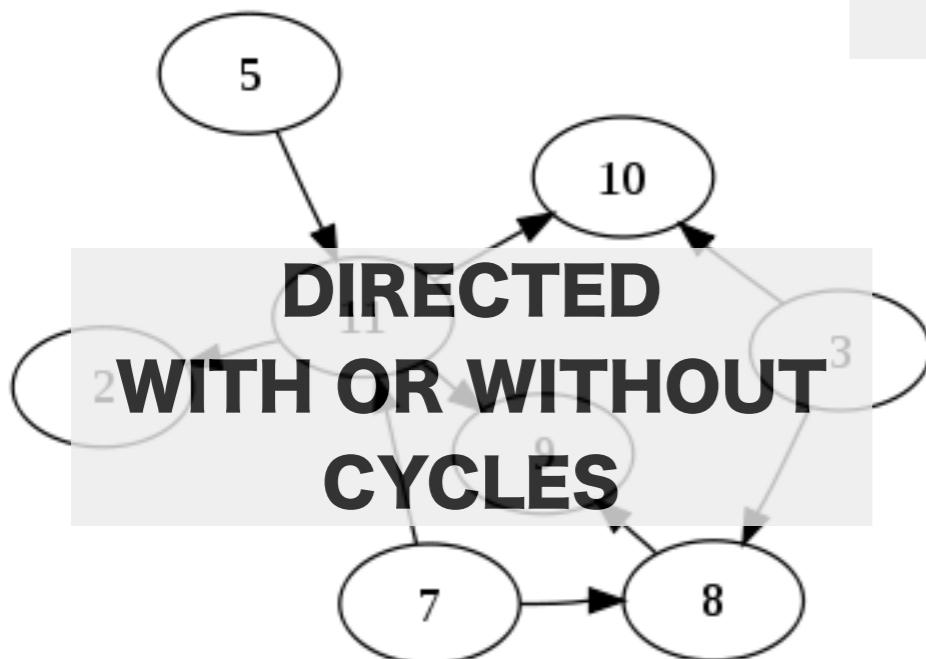
# ALL OF THESE GRAPHS



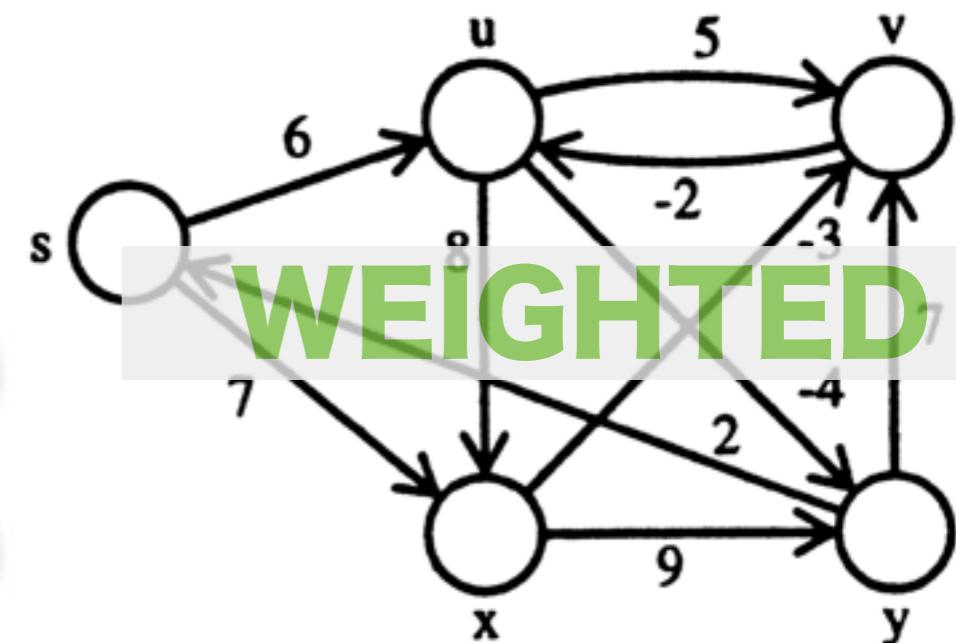
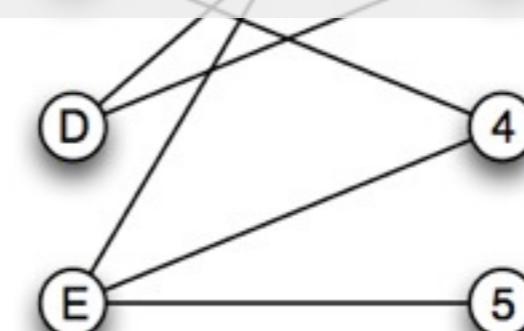
**UNDIRECTED**



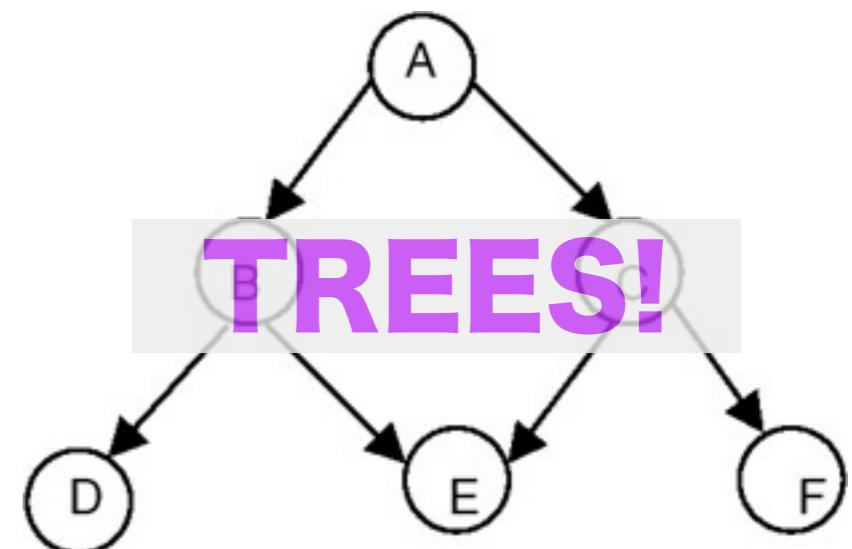
**BIPARTITE**



**DIRECTED  
WITH OR WITHOUT  
CYCLES**



**WEIGHTED**



**TREES!**

# ARE DIFFERENT

**TRANSPORTATION  
NETWORKS**

**INFORMATION NETWORKS**

**MOLECULAR CHEMISTRY**

**WIRELESS NETWORKS**

**MAJOR LEAGUE BASEBALL**

**DEPENDENCY MANAGEMENT**

A massive, bright orange and yellow explosion dominates the center of the image, set against a backdrop of a clear blue sky and a green, hilly landscape. The explosion is highly detailed, with intense fire and smoke billowing upwards and outwards. The text 'MIND. EXPLOSION.' is overlaid in large, white, sans-serif capital letters.

MIND. EXPLOSION.

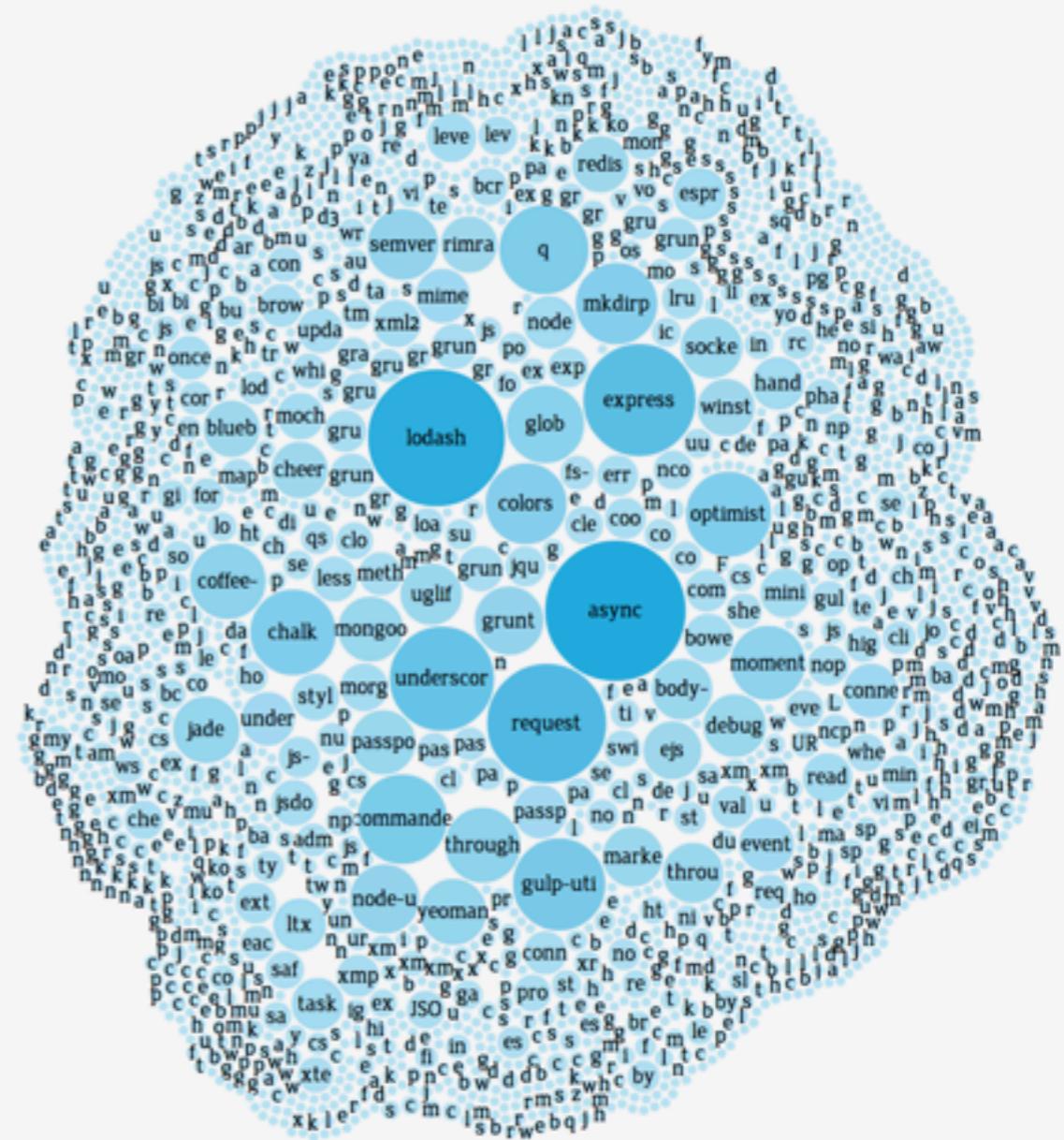
Actually.  
It's not so bad.



# Dependency graphs

## The $G = (V, E)$ kind of graph

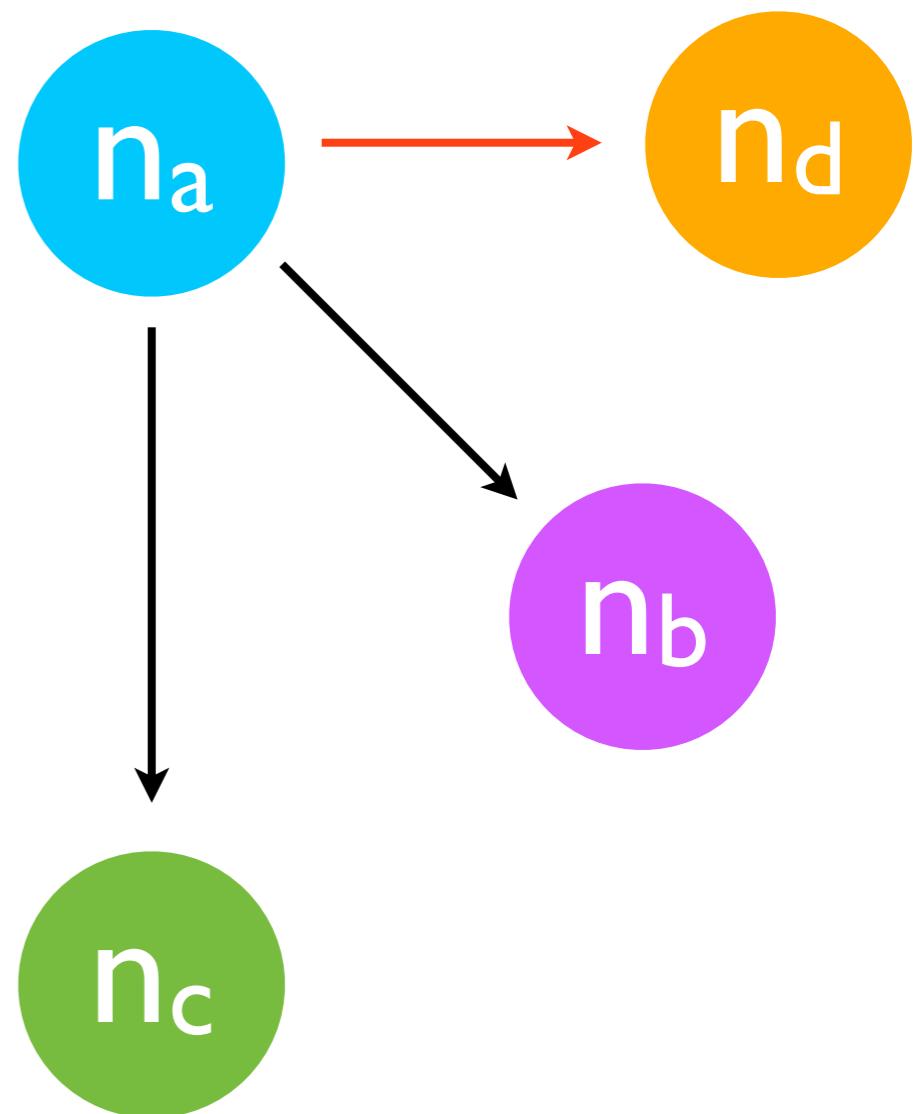
- Basic graph representation is not extremely helpful for visualizing package relationships.
  - But it does provide a basic structure for a graph search problem.



# Dependency graphs

The  $G = (V, E)$  kind of graph

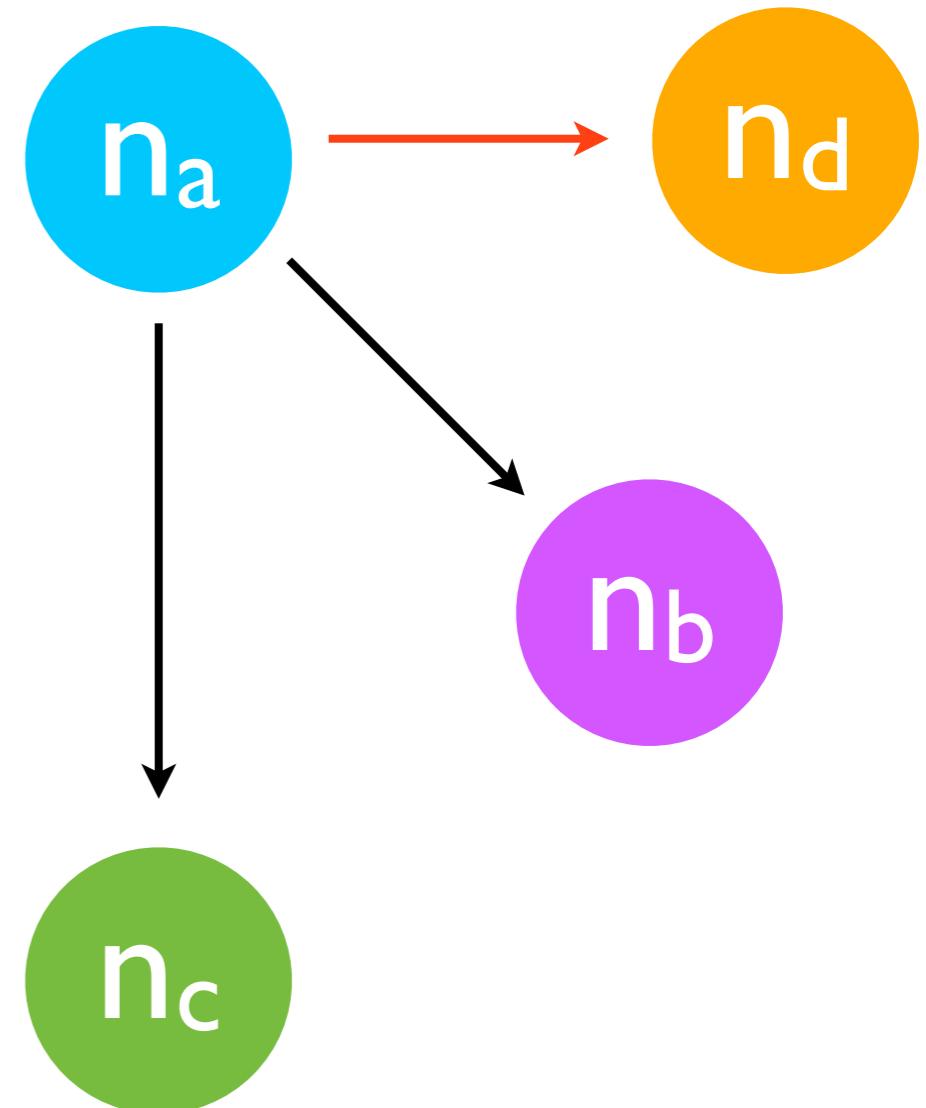
- To build our graph,  $G$ , we add a node (or vertex) for every package.
- We then add a colored edge from any node  $n_A$  to  $n_B$  if package A depends on package B.
- Edges are colored by dependency type:  
**dependencies** or  
**devDependencies**



# Dependency graphs

The  $G = (V, E)$  kind of graph

```
{  
  "name": "package-a",  
  "dependencies": {  
    "package-b": "~1.0.4",  
    "package-c": "~2.1.3"  
  },  
  "devDependencies": {  
    "package-d": "~3.1.2"  
  },  
  "main": "./index.js"  
}
```



Now imagine this for 100,000+ packages!

A massive, bright orange and yellow explosion dominates the center of the image, set against a backdrop of a clear blue sky and a hilly, grassy landscape. The explosion is highly detailed, showing intense fire and smoke. The text 'ANOTHER. MIND. EXPLOSION.' is overlaid in large, white, sans-serif capital letters.

ANOTHER. MIND. EXPLOSION.

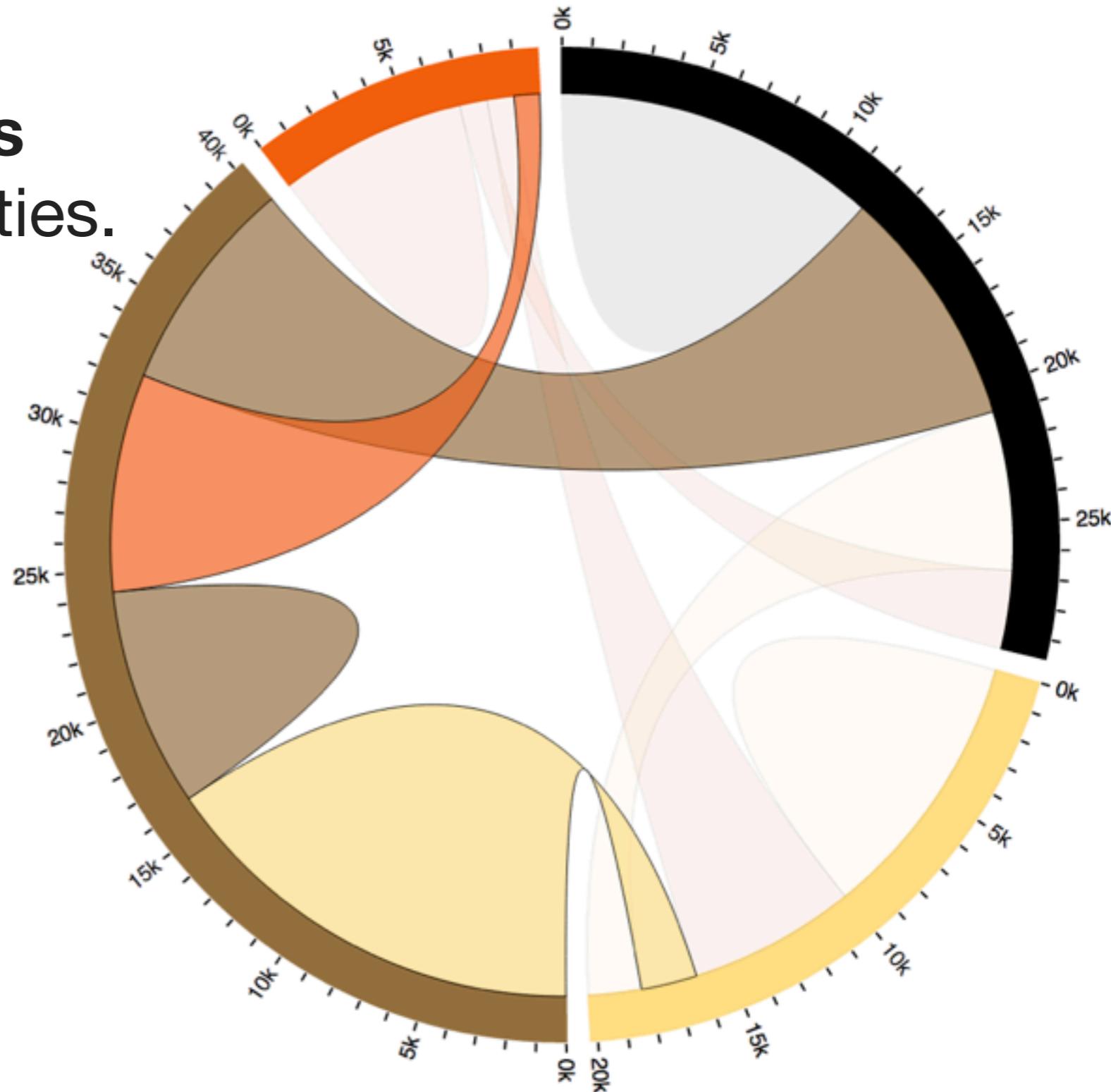


**I HEARD YOU'RE HAVING PROBLEMS  
D3.JS IS HERE TO HELP.**

# Graphs and Chords

What is a d3.js chord diagram?

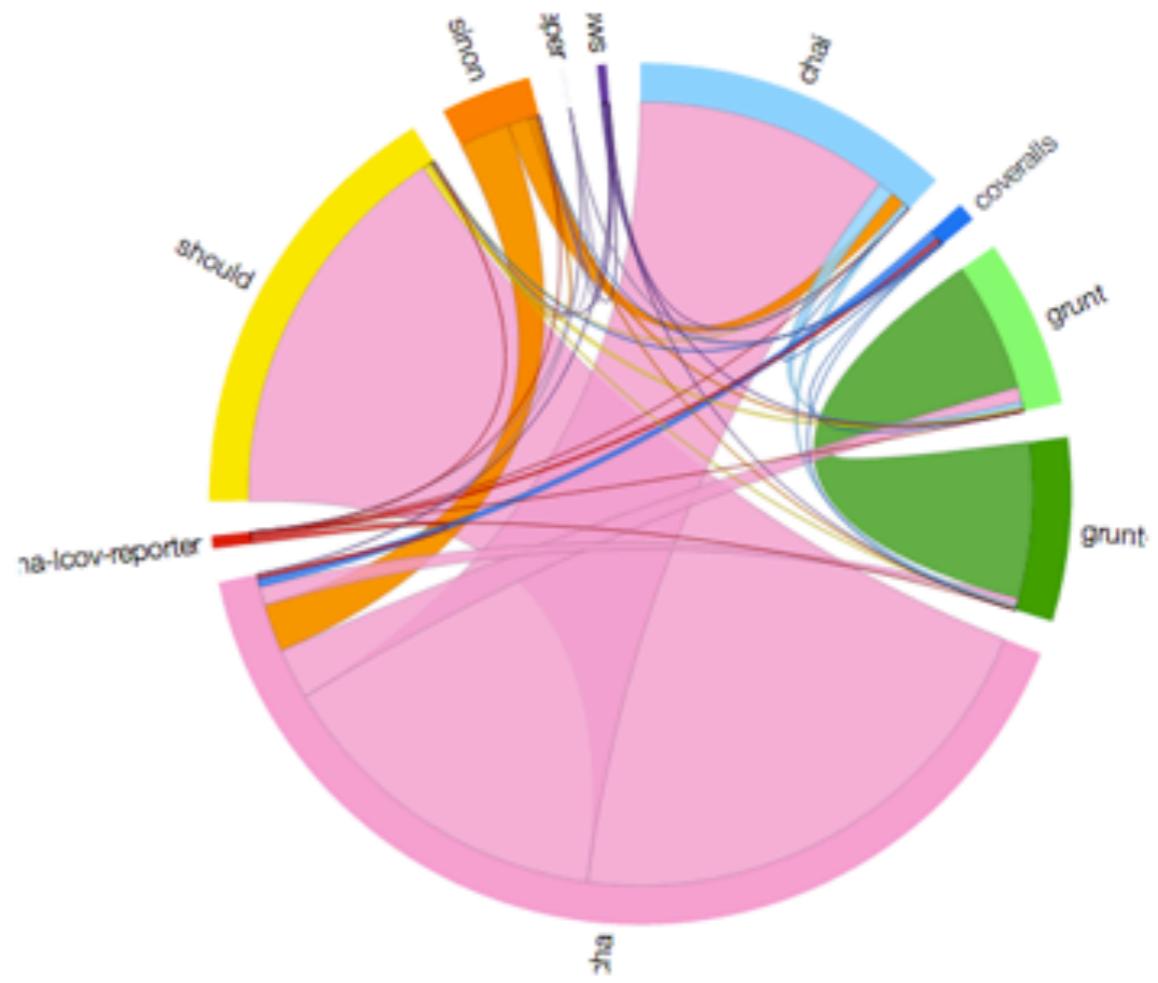
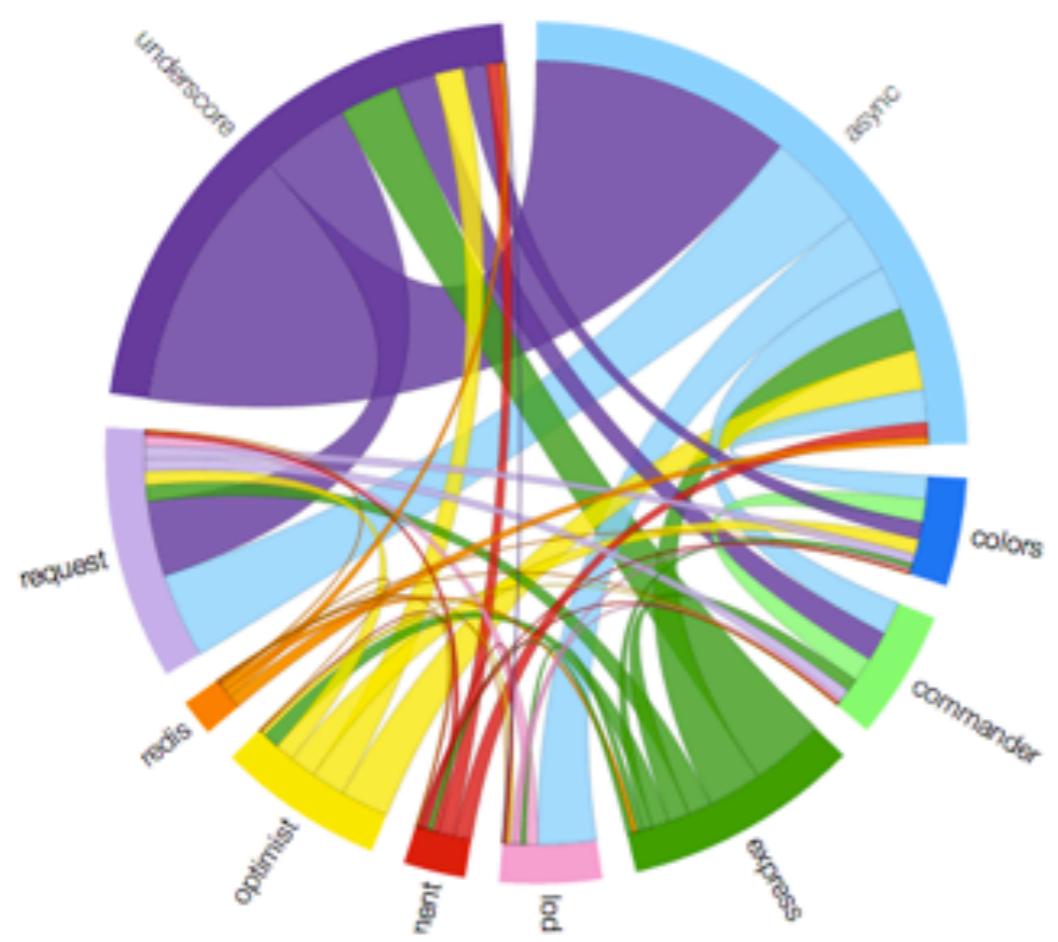
- Chord diagrams show **directed relationships** among a group of entities.
- They are built from a **matrix of values** that represent that relationship.
- That matrix is combined to calculate **arc size** and **chord size**.



# Graphs and Chords

How does a dependency relate to a chord diagram?

- There are tons of useful applications of dependency graphs.
- **Lets one: codependencies between modules.**



# AND THUS MEANINGFUL GRAPHS...





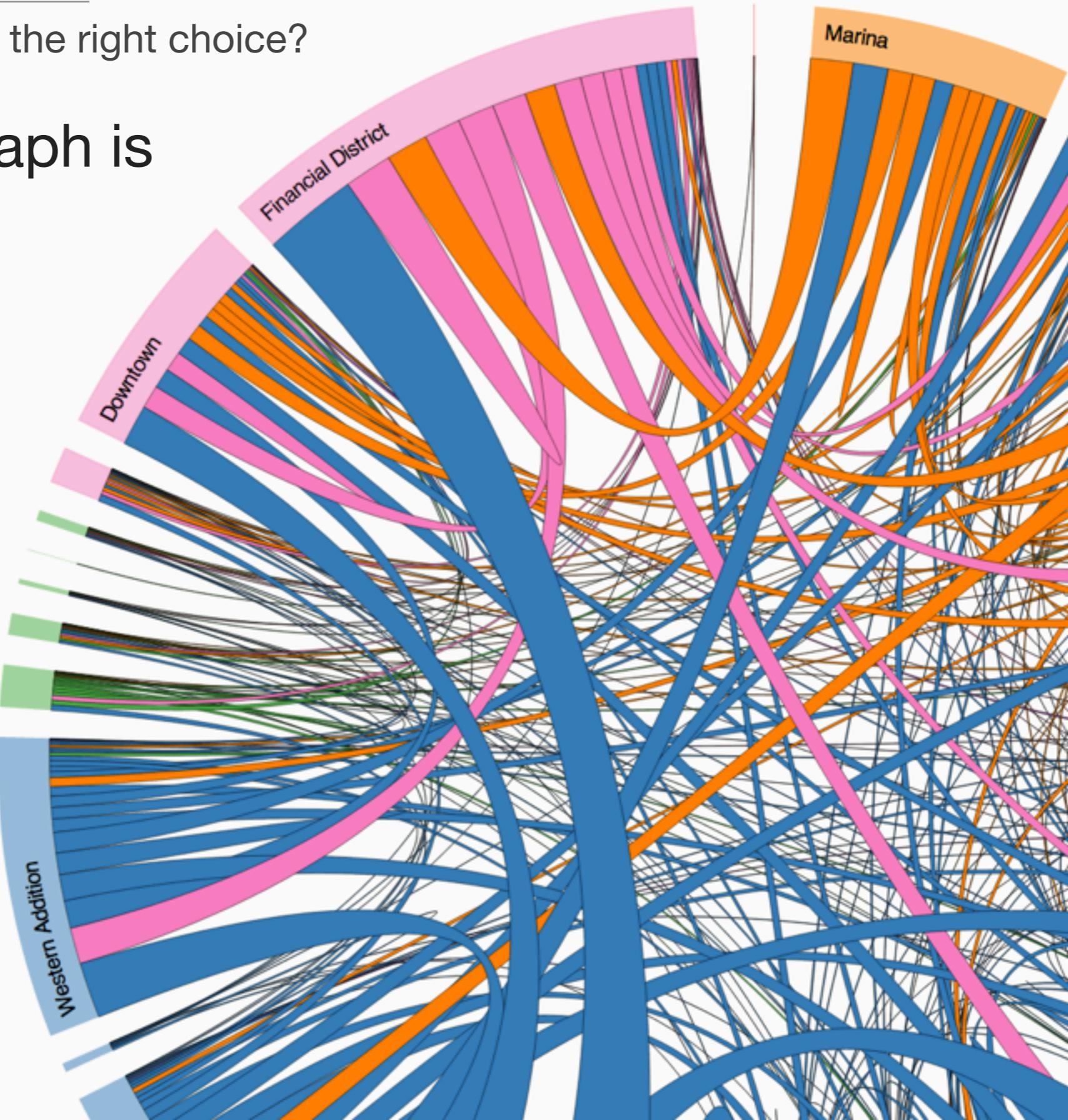
Well ... maybe

The **how** matters just as much as the **what**

# Graphs and Chords

Why was the d3.js chord diagram the right choice?

- The dependency graph is a set of **directed relationships**.
- Using npm and CouchDB we can **build a matrix** of that relationship.
- Semantically we can **assign value** to arc size, **chord size**, and **chord color**.



# Codependencies?

Well, technically co(\*)dependencies.

- Remember codependencies answer the question  
*“people who depend on package A also depend on ...”*

[ "package", "codep", "thru" ]

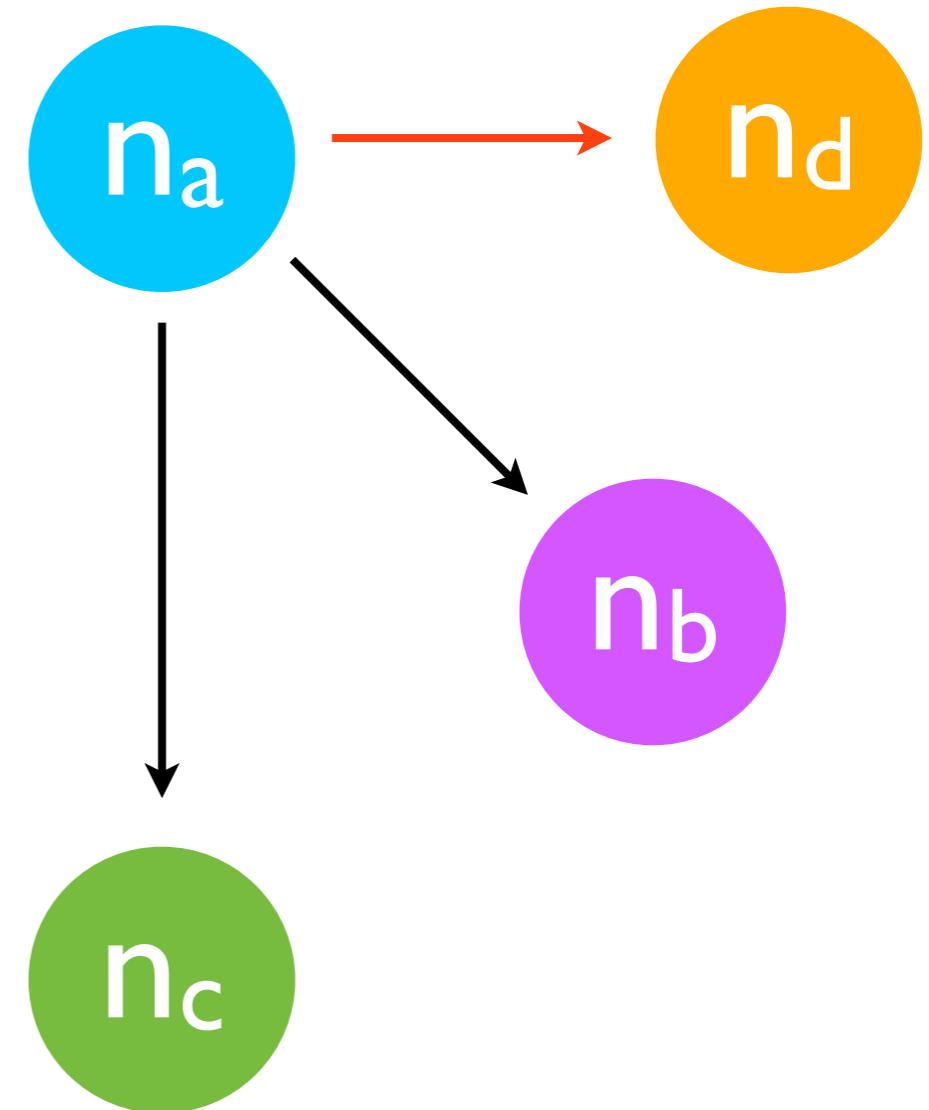


<3 CouchDB!

# Dependency graphs

The  $G = (V, E)$  kind of graph

```
{  
  "name": "package-a",  
  "dependencies": {  
    "package-b": "~1.0.4",  
    "package-c": "~2.1.3"  
  },  
  "devDependencies": {  
    "package-d": "~3.1.2"  
  },  
  "main": "./index.js"  
}
```



Here we would say that package-b and package-c have a codependency relationship thru package-a

# Codependencies?

Thanks CouchDB!

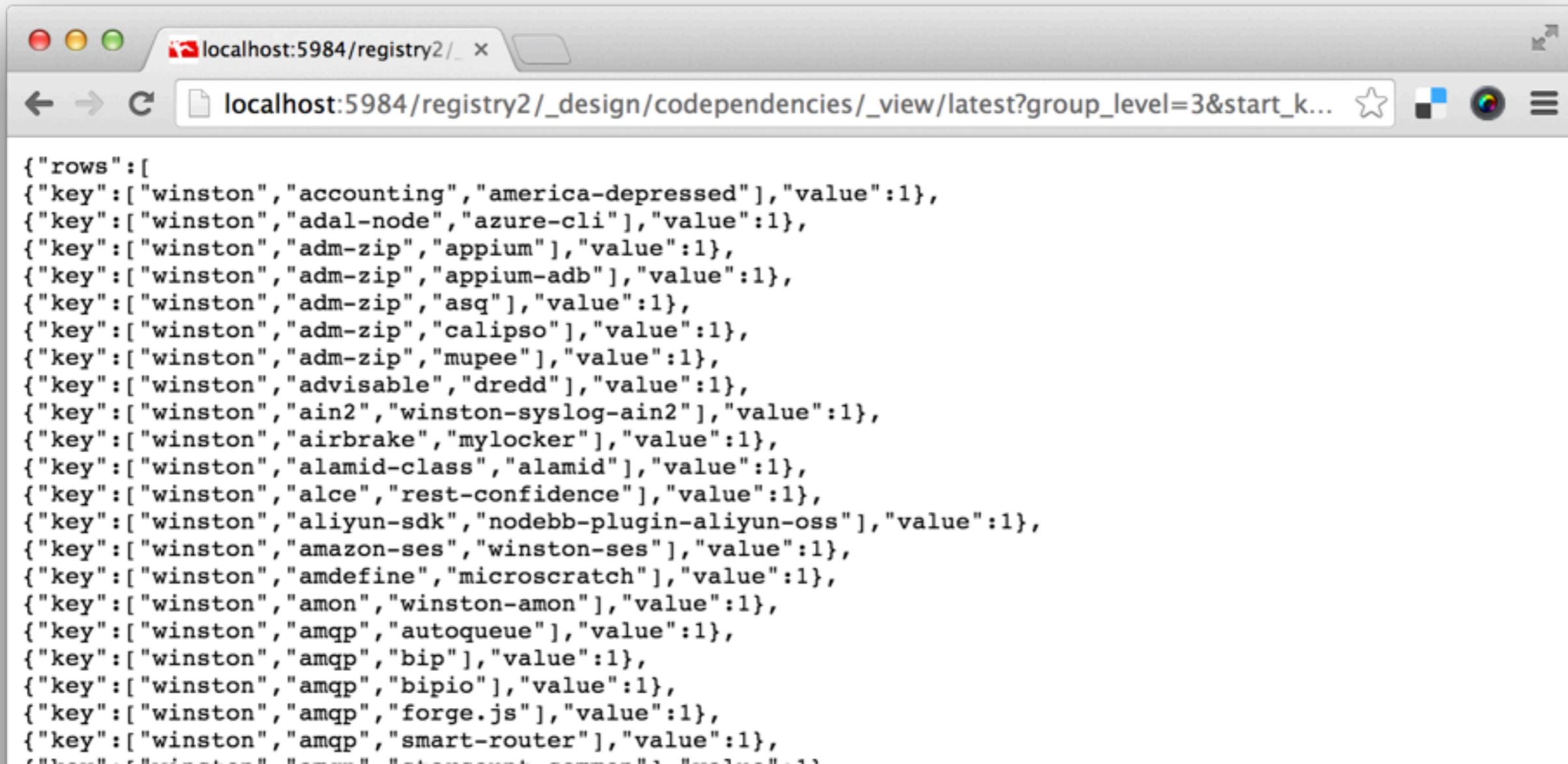
- We can get all of this from a **simple CouchDB view.**

```
for (var dep in d) {  
  dep = dep.trim();  
  for (var codep in d) {  
    codep = codep.trim();  
    if (dep !== codep) {  
      emit([dep, codep, doc._id], 1)  
    }  
  }  
}
```

# Codependencies

Thanks CouchDB!

```
group_level=3
&start_key=["winston"]
&end_key=["winston",{}]
```



```
{"rows": [
  {"key": ["winston", "accounting", "america-depressed"], "value": 1},
  {"key": ["winston", "adal-node", "azure-cli"], "value": 1},
  {"key": ["winston", "adm-zip", "appium"], "value": 1},
  {"key": ["winston", "adm-zip", "appium-adb"], "value": 1},
  {"key": ["winston", "adm-zip", "asq"], "value": 1},
  {"key": ["winston", "adm-zip", "calipso"], "value": 1},
  {"key": ["winston", "adm-zip", "mupee"], "value": 1},
  {"key": ["winston", "advisable", "dredd"], "value": 1},
  {"key": ["winston", "ain2", "winston-syslog-ain2"], "value": 1},
  {"key": ["winston", "airbrake", "mylocker"], "value": 1},
  {"key": ["winston", "alamid-class", "alamid"], "value": 1},
  {"key": ["winston", "alce", "rest-confidence"], "value": 1},
  {"key": ["winston", "aliyun-sdk", "nodebb-plugin-aliyun-oss"], "value": 1},
  {"key": ["winston", "amazon-ses", "winston-ses"], "value": 1},
  {"key": ["winston", "amdefine", "microscratch"], "value": 1},
  {"key": ["winston", "amon", "winston-amon"], "value": 1},
  {"key": ["winston", "amqp", "autoqueue"], "value": 1},
  {"key": ["winston", "amqp", "bip"], "value": 1},
  {"key": ["winston", "amqp", "bipio"], "value": 1},
  {"key": ["winston", "amqp", "forge.js"], "value": 1},
  {"key": ["winston", "amqp", "smart-router"], "value": 1},
  {"key": ["winston", "amon", "winston-amon"], "value": 1}
]}
```

# Codependencies

---

The meat of the analysis

- So this is all well and good, but what the heck are you doing?!?!

For module **NAME** generate a matrix by:

- **Rank** codependencies based on number of times they appear
- For each codependency **C** in the **SET** of the top **N**:  
**Rank** **SET** – **{C}** by number of times they appear to create **Row[C]**

# Codependencies

Understanding codependencies through winston

- This last step yields a matrix for the codependency relationship:

	asyn...	expr...	opti...	requ...	unde...
asyn...	<b>0.0000</b>	<b>553.0000</b>	<b>534.0000</b>	<b>837.0000</b>	<b>1359.0000</b>
expr...	<b>553.0000</b>	<b>0.0000</b>	<b>314.0000</b>	<b>365.0000</b>	<b>648.0000</b>
opti...	<b>534.0000</b>	<b>314.0000</b>	<b>0.0000</b>	<b>335.0000</b>	<b>448.0000</b>
requ...	<b>837.0000</b>	<b>365.0000</b>	<b>335.0000</b>	<b>0.0000</b>	<b>786.0000</b>
unde...	<b>1359.0000</b>	<b>648.0000</b>	<b>448.0000</b>	<b>786.0000</b>	<b>0.0000</b>

# Codependencies

Understanding codependencies through winston

- Now we need to weight the matrix based on the overall appearance of these codependencies

240		async		0.2761
207		underscore		0.2382
163		express		0.1875
133		request		0.1530
126		optimist		0.1449
869	<b>total</b>			

# Codependencies

Understanding codependencies through winston

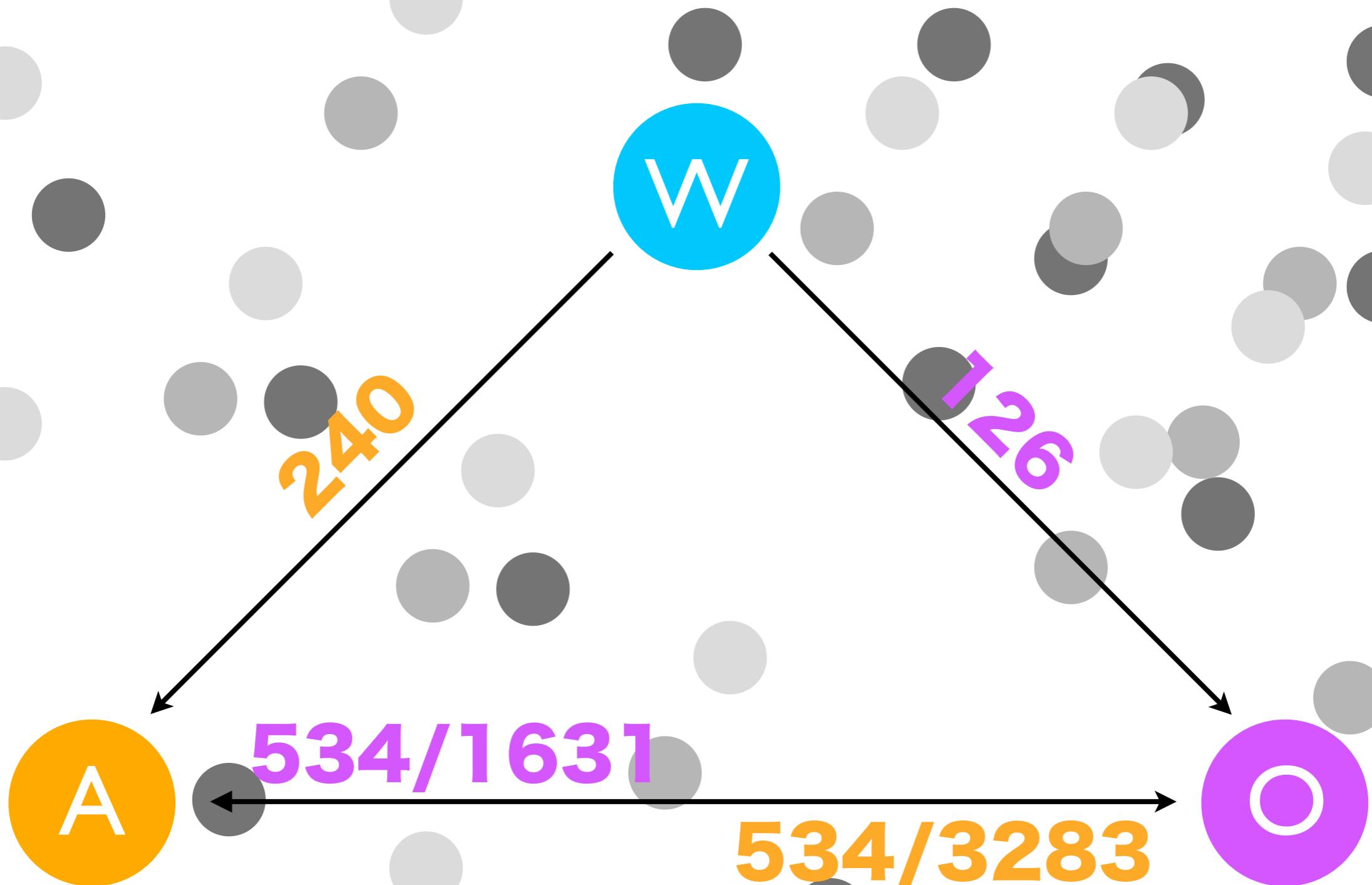
- To do this we go and calculate the codependencies for each of the members

## ASYNC

1359		underscore	0.4139
837		request	0.2549
553		express	0.1684
534		optimist	0.1626
3283	<b>total</b>		



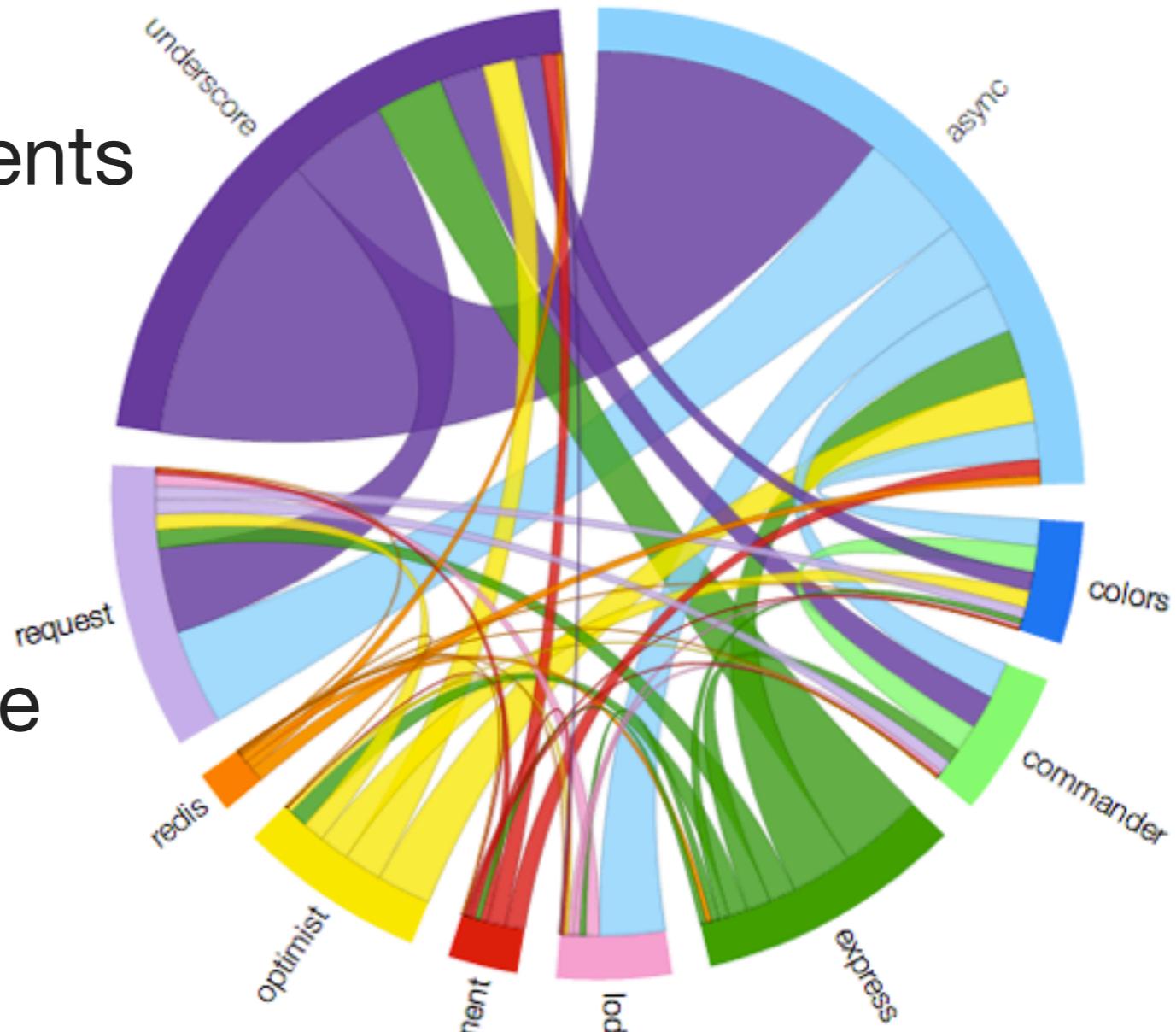
WAT?



# Codependencies

Reading the tea leaves of dense data visualization

- **The size of the arc** represents the degree of the codependency relationship with the parent module.
- **The size of the chord** represents the degree of the codependency relationship between each pair.
- **The color of the chord** represents the “dominant” module between the pair.



winston



PAUSE FOR  
DEMO

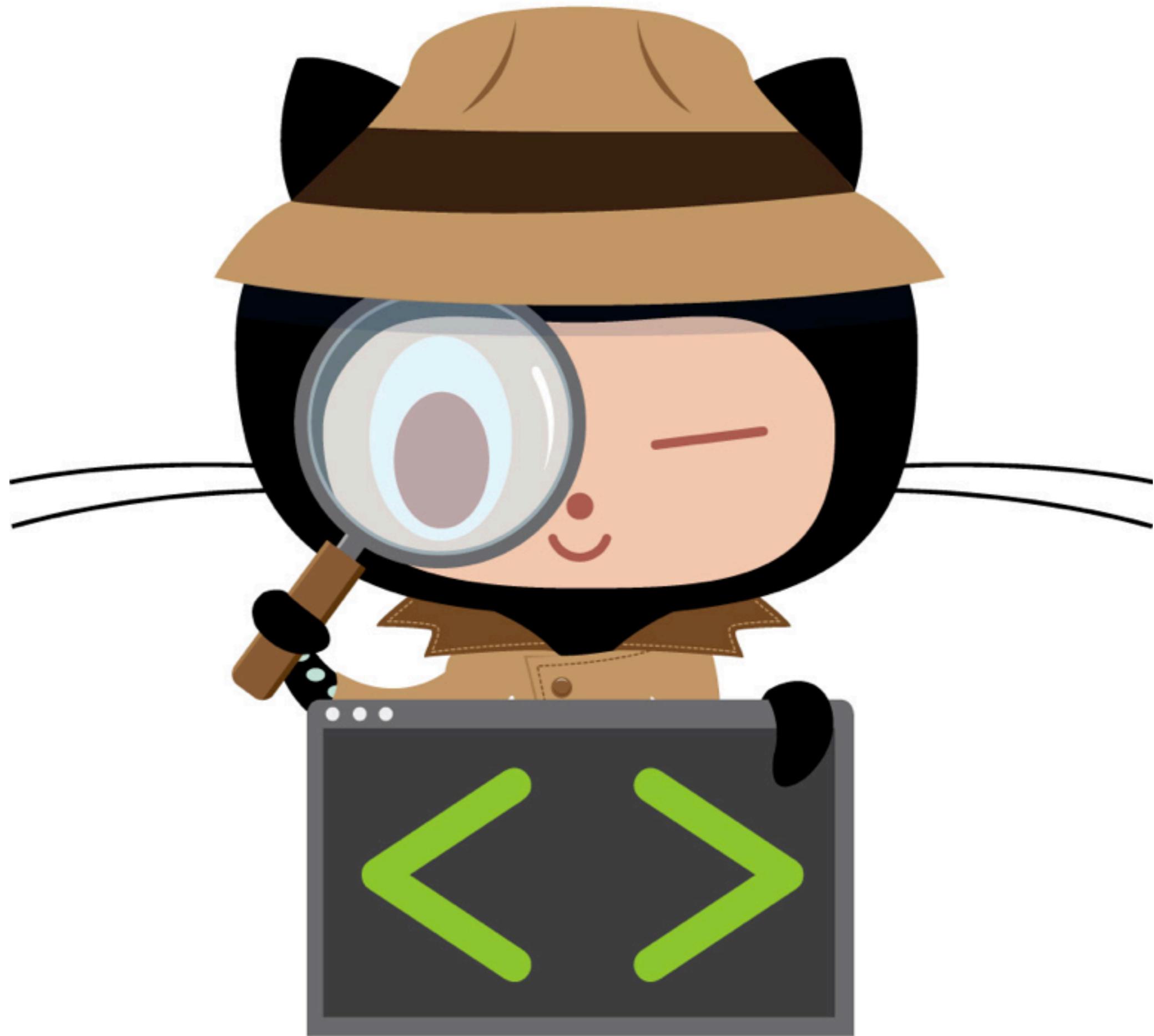
COMEDY C

# ALL OPEN SOURCE

**Seriously... what else do people do?**

**indexzero/npm-codependencies**

**indexzero/npm-comp-stat-www**









**nodejitsu**

**FIN**