

# Open Source

17-313 Spring 2024

Foundations of Software Engineering

<https://cmu-313.github.io>

**Michael Hilton** and Eduardo Feo Flushing

# Administrivia

- P4 due tonight
- Midterm 2 review session in recitation 4/15
- Final Exam attendance Mandatory:
  - Monday, April 29, 2024 05:30pm-08:30pm
  - If you will be celebrating Passover, let us know ASAP to support alternatives.
  - Conflicts come talk to us as well
- Monday April 8<sup>th</sup> eclipse

# Early Course Feedback

- Start Doing:
  - more descriptive writeups x6
  - JS/TS review x4
  - more lectures on how to navigate code bases
  - more recitation group activities
  - more check-ins with TAs
  - explain project in a lecture or recitation
  - more diverse candy
  - more TA office hours, OHQ
  - make teams more fair
  - show solutions to technical challenges

# Early Course Feedback

- Stop Doing:
  - slack
  - unclear project instructions x4
  - more Businessy lectures.
  - in-class attendance
  - JS/TS without teaching in class x2
  - lecture without breaks x2
  - short checkpoints
  - no work on weekends

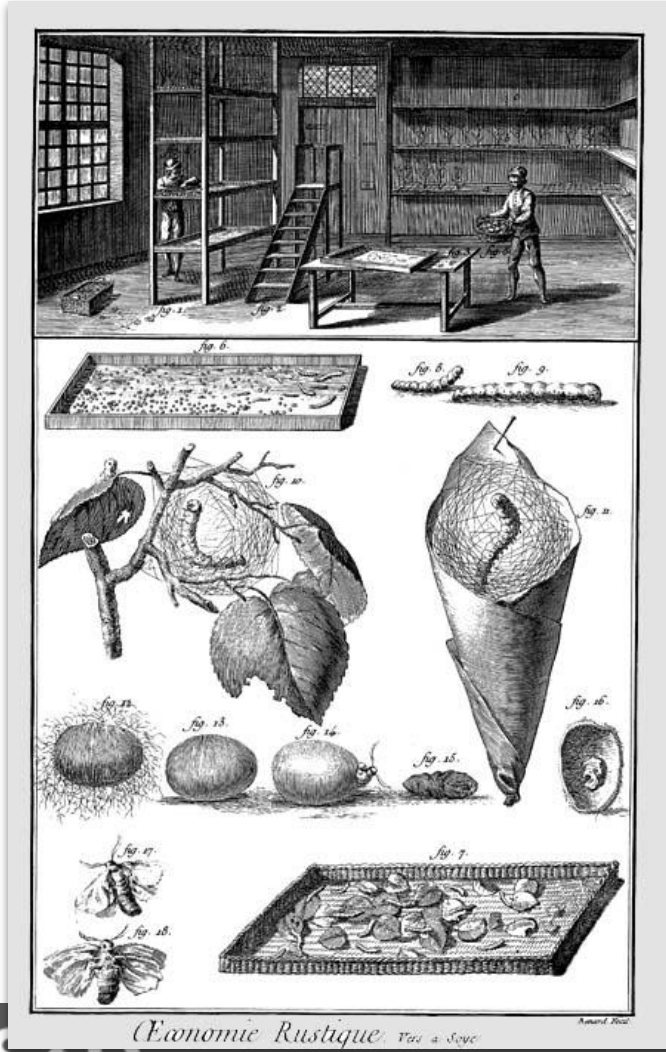
# Early Course Feedback

- Keep Doing:
  - candy x6
  - reciations
  - project based x3
  - lecture topics
  - interactive in-class activites x5
  - laptop policy x3
  - slack

# Software Patents

# Software Patents: The Good, The Bad, and The Ugly

# Venice, 1474



*Economie Rustique. Vrai à Savoir*

Mcccclxxiiii. die xxiiii Martij. .32.

Sap Consily. vi.  
 p. Marcus Coz mil.  
 p. Ludovicus fustis tota.  
 p. Paulus Manzocina.  
 p. Bernardus Justin mil.  
 p. Vital' Lando doc. 7 mil.  
 Sap. tre firme.  
 p. Anton' de puolis.  
 p. Lodovicus filitio.  
 p. Zachar' Ba.  
 p. Benedictus t.

Sono i questa Cita / et anche ala zornada p la grandeza et bona sda  
 Concorra homeni da dmerzse bande / et acutissimi frugeri / apti ad excogitar  
 et trouar vazij Ingegnerosi artificij . Et sel fosse promisto / che le opere et artificij  
 trouade da loro . altri viste che le hanessero / no podesseno farle se tuoz honoz  
 suo / Simel homeni exercitaziano linguerno / troueziano / et faziano & le chosse /  
 che sauzano de no picola vtilita et beneficio al stato nro . ppero L andara parte  
 che p auctorita de questo ofero / chadany che fara i questa Cita alqun nuouo  
 modo artificio / no facto panti nel dno nro / Reducto chel fara a  
 darlo i nota al officio  
 un altro i alguna terra  
 vltitudine & quello senza  
 tamen se alqun el fesse /  
 achadany officio de  
 sia astreto apatarli  
 liberta de la nra signor /  
 di diti artificij / et  
 auctori no li possi excitar .





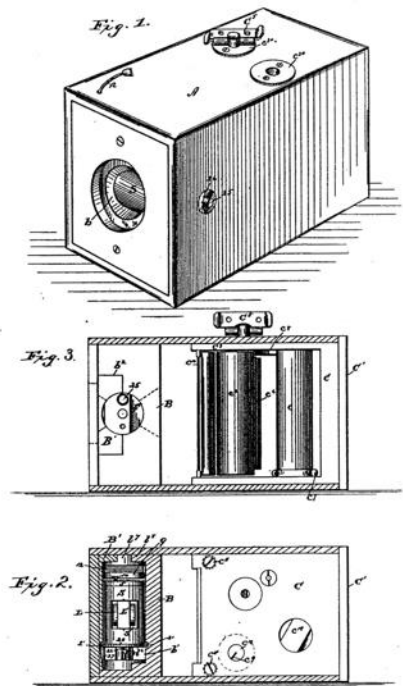
# England, 1566





# Today: USA

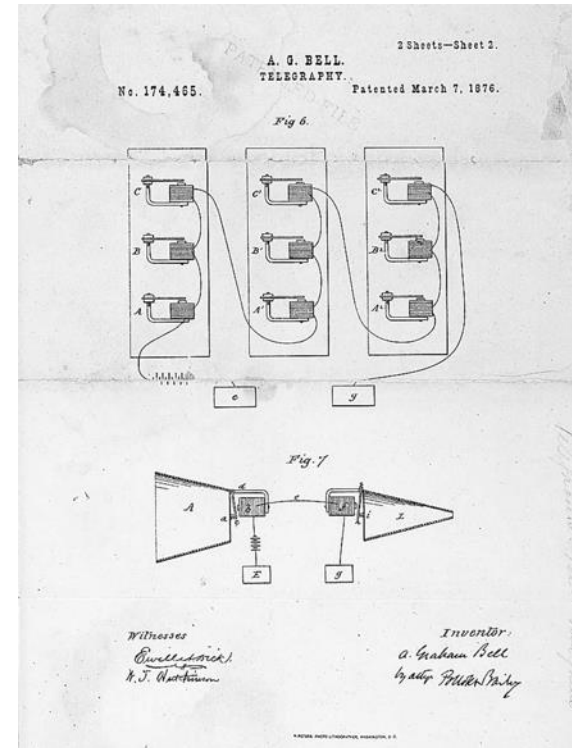
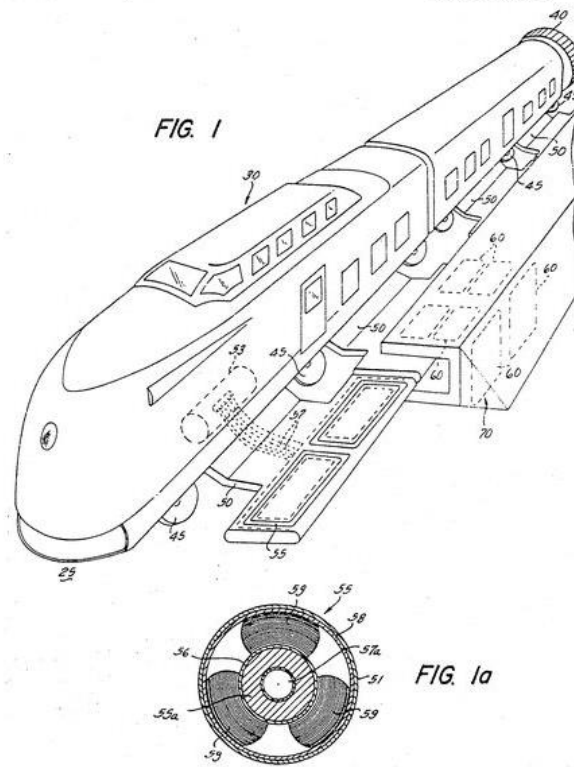
(No Model.)  
 G. EASTMAN.  
 CAMERA.  
 No. 388,850. Patented Sept. 4, 1888.  
 3 Sheets—Sheet 1.



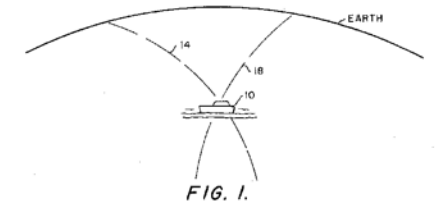
Witnesses:  
 Chas. R. Bur.  
 A. J. Stewart

Inventor:  
 George Eastman.  
 by *Charles H. Clark*  
 his Attorneys.

Oct. 7, 1969 J. R. POWELL, JR. ET AL 3,470,828  
 ELECTROMAGNETIC INDUCTIVE SUSPENSION AND STABILIZATION  
 SYSTEM FOR A GROUND VEHICLE  
 Filed Nov. 21, 1967 8 Sheets—Sheet 1



PATENTED JAN 29 1874 3,789,409  
 SHEET 1 OF 2  
 O-12 SATELLITE O-16 SATELLITE



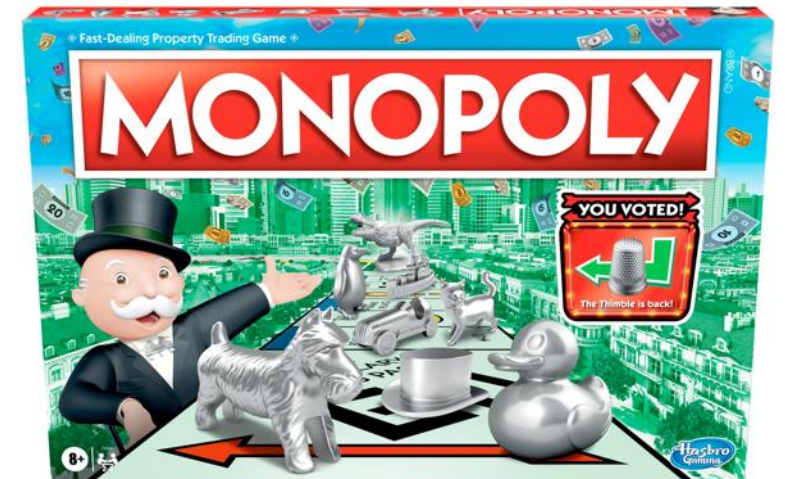
What is a patent? New. Useful. Non-obvious.

“A patent is an exclusive right granted for an invention, which is a product or a process that provides, in general, **a new way of doing something**, or offers a **new technical solution to a problem**. To get a patent, technical information about the invention must be disclosed to the public in a patent application.”



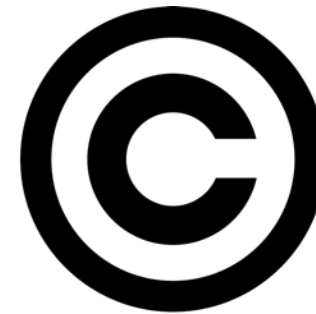
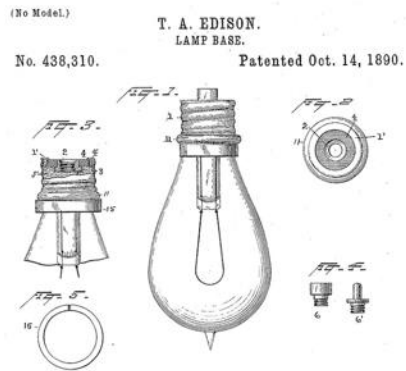
# What rights do patents grant?

- Patents **don't** give you the right to make, use, or sell an invention.
- Patents **do** give you the right to **exclude others** from making, using, and selling an invention for the term of a patent (20 years)
  - stop or sue others
  - licensing and royalties



## What's the difference? Patents vs. Copyright

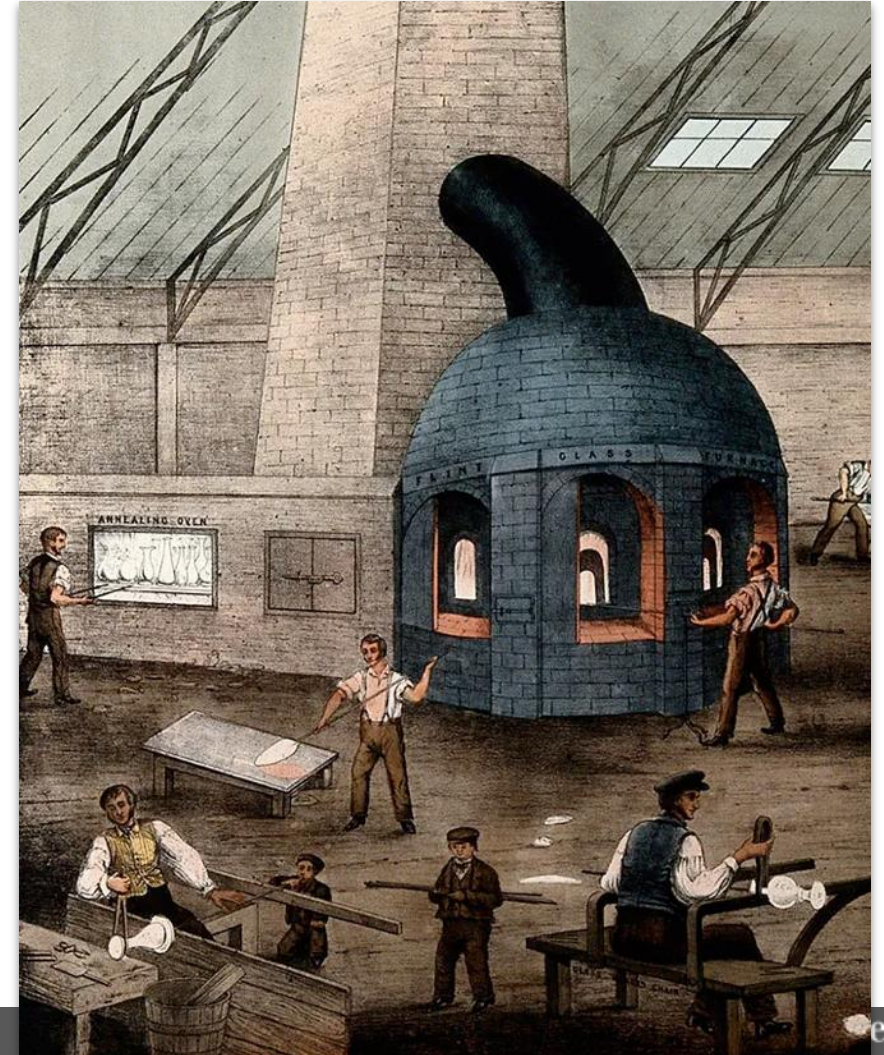
- Copyrights cover the details of expression of a work
- Copyrights don't cover any ideas  
Patents only cover ideas and the use of ideas
- Copyrights happen automatically.  
Patents are issued by a patent office in response to an application.





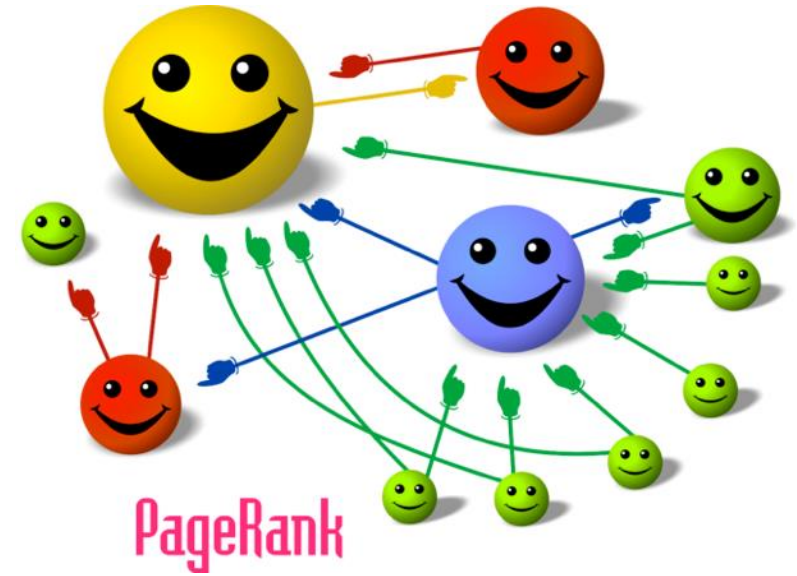
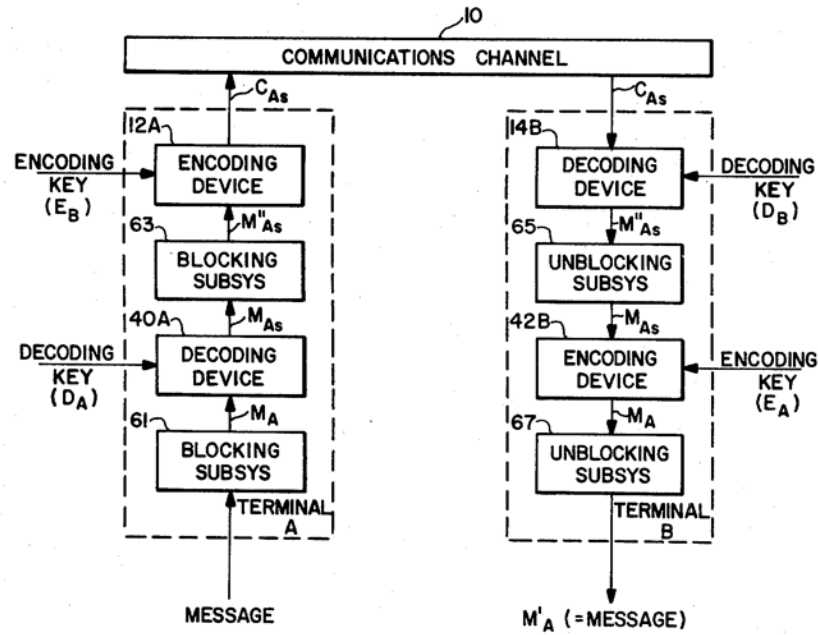
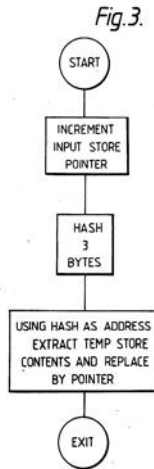
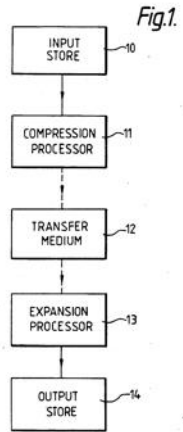
# Why do patents exist?

- Encourage disclosure of inventions
- Reward invention and creativity
- Protect investment of capital into R&D
- Encourage the market to “design around”
- Protect small companies from large ones



# Software Patents

U.S. Patent Oct. 20, 1987 Sheet 1 of 6 4,701,745



Patent or not?




## Patent or not?

1. Running bingo on a computer
2. Using a computer to help users plan meals while achieving diet goals
3. Using a computer to order a pizza with customized toppings
4. Prompting a user before establishing a new network connection
5. Automatically notifying users when an item is picked up or delivered
6. Using a computer network to ask people to complete tasks and then wait for them to do them
7. Using SMS to perform tasks (e.g., checking bank balance)
8. Selecting ALL images in a CAPTCHA that match a given text

The software patent system is broken!

# Alice vs. CLS Bank (2014)

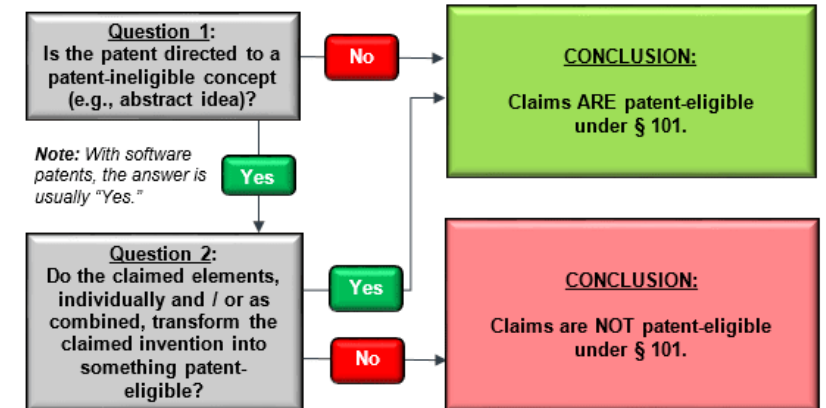
Case	Claimed Invention		Result
<i>Alice Corp. v. CLS Bank</i> (June 19, 2014)	Method of computerized risk mitigation in financial settlements	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     ✗ Step 1 ✗ Step 2                 </div>	<b>NOT Patent Eligible</b> Why? Risk mitigation is a long-standing “fundamental economic practice” (step 1) and the claims merely required generic computer implementation (step 2)
<i>Digitech</i> (July 11, 2014) 	Method of digital image processing; used “device profiles” to organize devices’ spatial and color properties	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     ✗ Step 1 ✗ Step 2                 </div>	<b>NOT Patent Eligible</b> Why? Claimed “device profile” was intangible; method claims covered organization of information untethered to specific structure.
<i>buySAFE v. Google</i> (Sep. 3, 2014)	Online transaction performance guarantee	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     ✗ Step 1 ✗ Step 2                 </div>	<b>NOT Patent Eligible</b> Why? The claims are about creating a contractual relationship that is performed by any general purpose computer.
<i>Ultramerical v. Hulu</i> (Nov. 14, 2014)	Internet-distribution of copyright material	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     ✗ Step 1 ✗ Step 2                 </div>	<b>NOT Patent Eligible</b> Why? Offering media in exchange for viewing an advertisement is an abstract idea. Implementing it on the internet does not transform it into patent eligible.

**ars** TECHNICA
SUBSCRIBE
🔍 ☰ SIGN IN

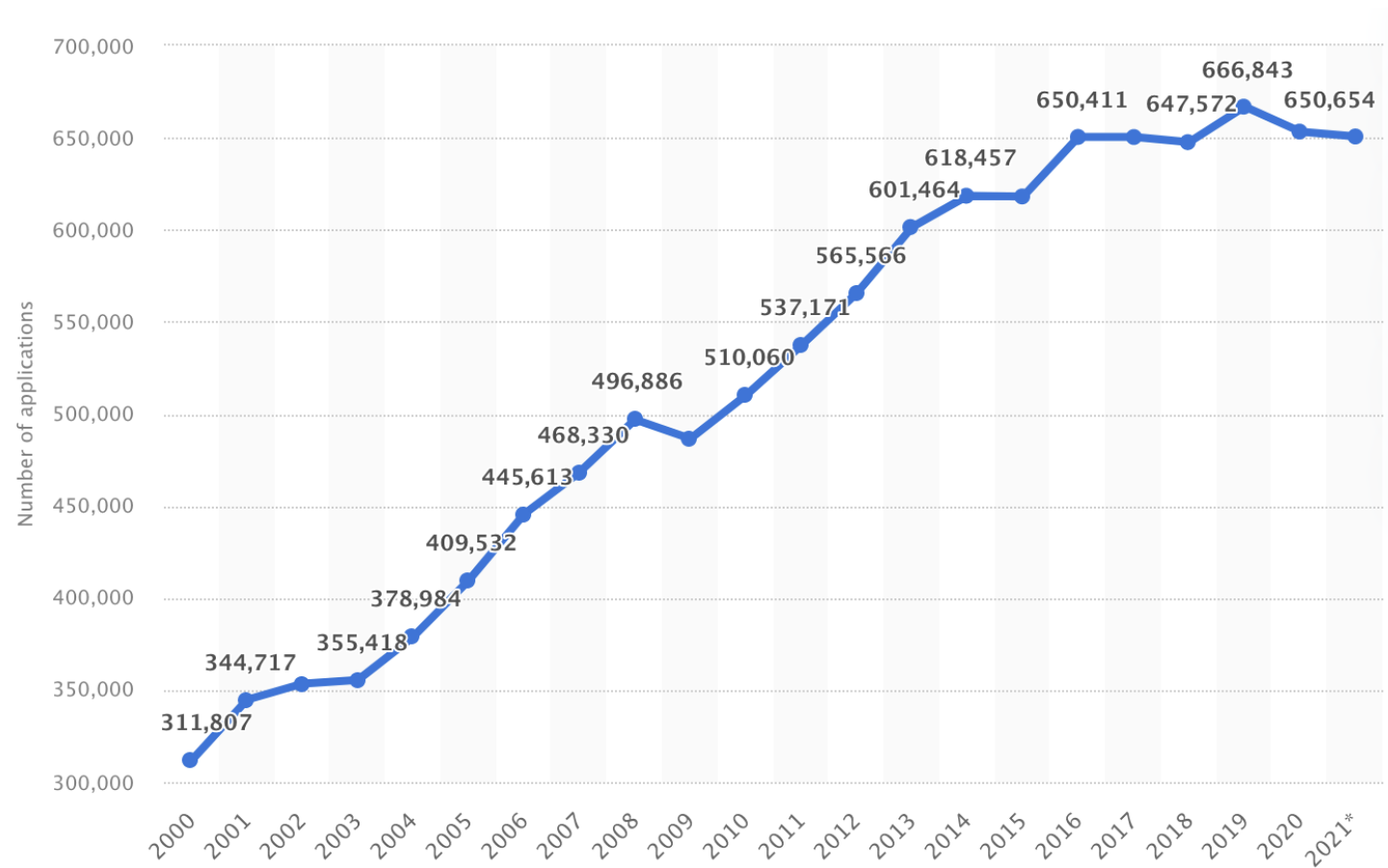
POLICY —  
**Supreme Court smashes “do it on a computer” patents in 9-0 opinion**

Court declines to stop software patents altogether.

JOE MULLIN - 6/19/2014, 12:08 PM



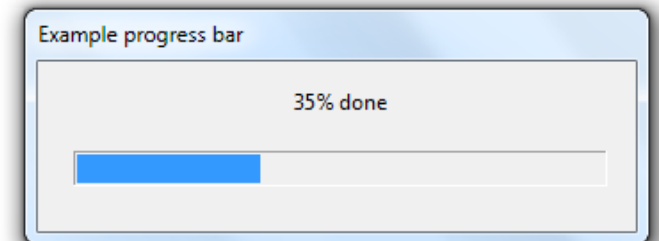
# Problem: Inventive step and non-obviousness



or 1-Click Checkout



[US5960411A](#)



[US5301348A](#)

# Problem: Long patent pendencies and terms

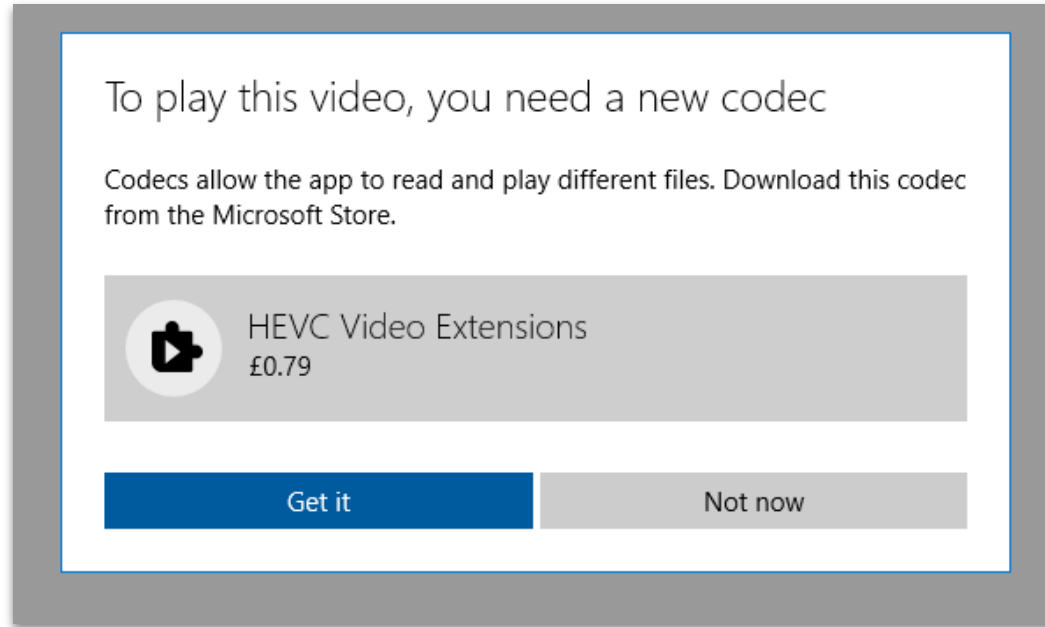
TABLE 4: **PATENT PENDENCY STATISTICS (FY 2021)**

Utility, Plant, Reissue Pendency Statistics by Technology Center (in months)	Average First Action Pendency	Total Average Pendency
<b>Total Utility, Plant, and Reissue Pendency</b>	<b>16.9</b>	<b>23.3</b>
Tech Center 1600—Biotechnology and Organic Chemistry	17.0	24.0
Tech Center 1700—Chemical and Materials Engineering	18.8	26.7
Tech Center 2100—Computer Architecture, Software, and Information Security	17.5	25.6
Tech Center 2400—Networks, Multiplexing, Cable, and Security	15.7	22.9
Tech Center 2600—Communications	13.5	19.9
Tech Center 2800—Semiconductor, Electrical, Optical Systems, and Components	15.7	22.3
Tech Center 3600—Transportation, Construction, Agriculture, and Electronic Commerce	18.1	25.9
Tech Center 3700—Mechanical Engineering, Manufacturing, and Products	18.6	26.7



# Problem: Incompatibility

- PNG was invented to avoid GIF patent issues
- Opus is a patent-free MP3 alternative
- AV1 vs H265



Problem: Independent discovery doesn't matter!

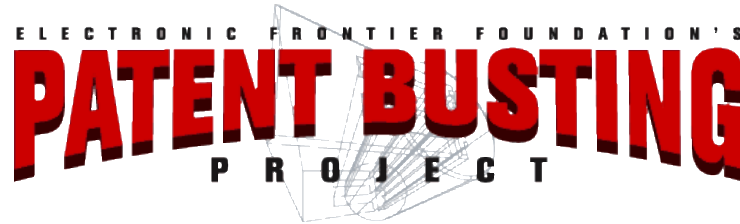
"The idea that I can be presented with a problem, set out to logically solve it with the tools at hand, and wind up with a program that could not be legally used because someone else followed the same logical steps some years ago and filed for a patent on it is horrifying."

*John Carmack*



## Problem: Only large organizations benefit

- **The patent system relies on people to challenge bad patents**
  - requires considerable time, money, and legal expertise
  - the US legal system requires both parties to pay legal fees (c.f., losers pay costs in Europe) \*
- US software patents cost between **\$15,000 to \$45,000!**
  - that's before you even apply for international patents!



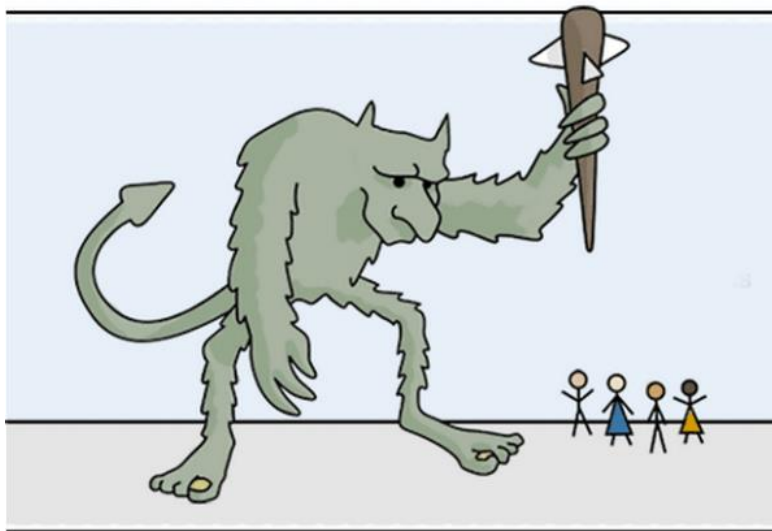
<https://www.patenttrademarkblog.com/how-much-patent-costs>

<https://www.eff.org/issues/patent-busting-project>



# Problem: Non-Practicing Entities (Patent Trolls)

PATENT TROLLS ARE A PROBLEM IN THE U.S.



**Patent trolls hijack ideas and extort money from those who do the real work.**

Today the Administration is taking action to protect innovators and ensure the highest-quality patents in our system.

WH.GOV/PATENTTROLLS

JUNE 4, 2013



BBC Home News Sport Reel Worklife

## NEWS

Home | War in Ukraine | Coronavirus | Climate | Video | World | US & Canada | UK

Tech

### 'Patent trolls' cost other US bodies \$29bn last year, says study

© 29 June 2012

**Infringement of patents**  
of \_\_\_\_\_ which are alleged to be infrin  
oss by reason of purchasing and selling \_\_\_\_  
erson or persons purchasing and using su  
parties that A will sell B \_\_\_\_\_

THINKSTOCK

Patent portfolio owners say their actions help incentivise inventors to carry out research

## Problem: Innovation is Stifled

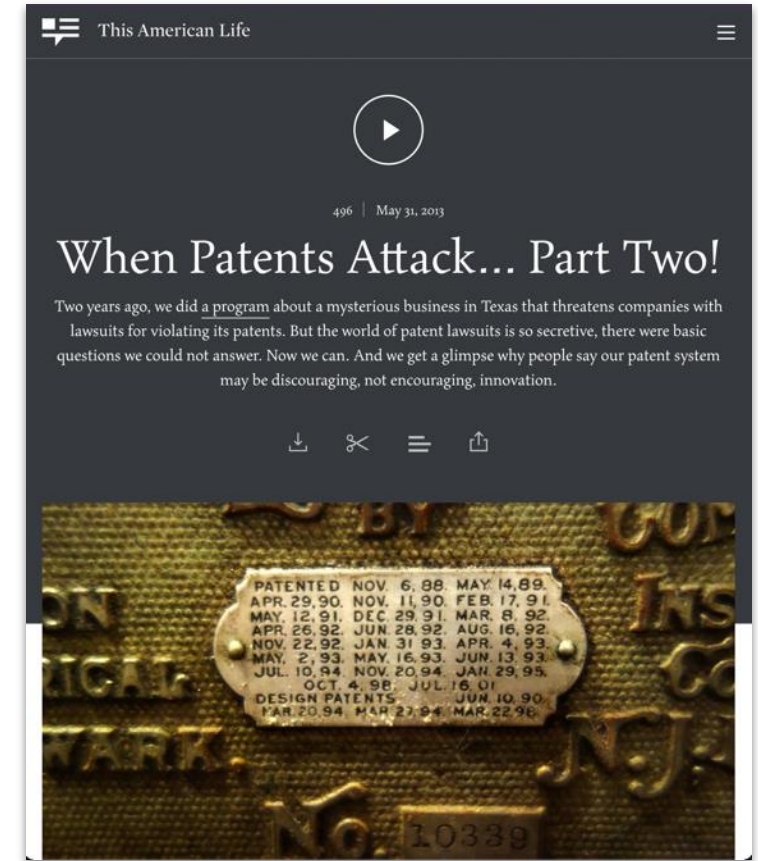
“As a developer for a small startup, absurd software patents are a constant worry. Stories abound of people like us getting pressured out of existence over the use of incredibly vague, basic interface elements and system components.”

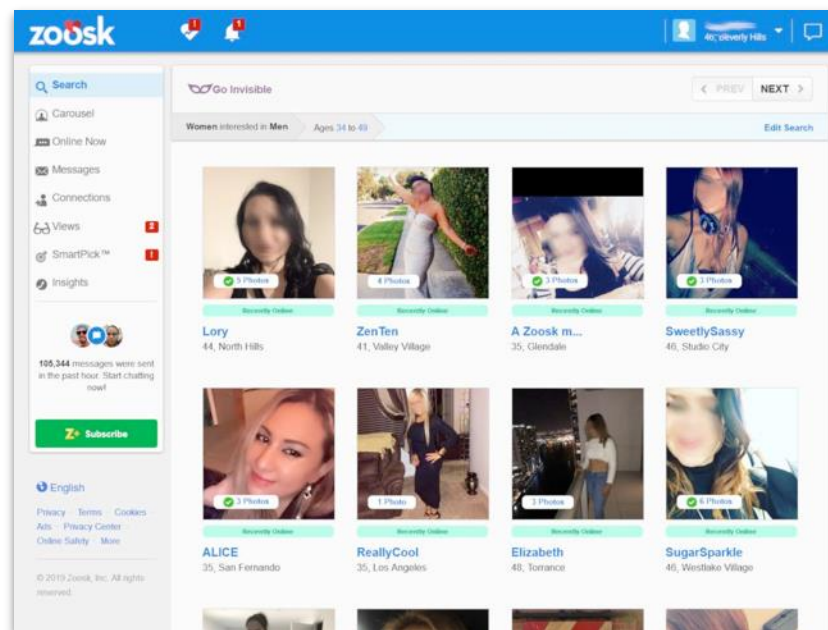
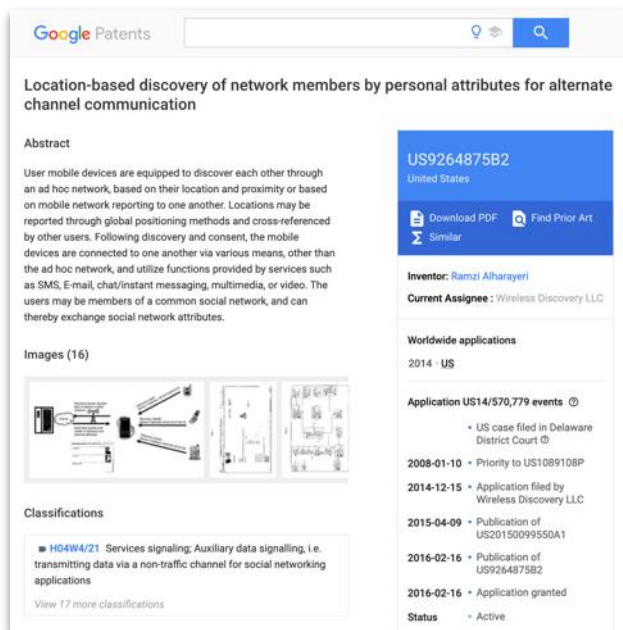
“Software patents are generally written in vague and nontechnical legal language, which obfuscates the patent in question . . . and also makes it easy to dramatically extend the patent to elements not considered at all when the patent was originally filed.”



# This American Life: When Patents Attack!

- Innovatio sued libraries and coffee shops for providing WiFi in a public space
- Boadin has sued various media outlets, claiming that its patents are infringed whenever a word or phrase on your computer autocompletes
- NPHJ claims they hold a patent on “scanning and emailing documents”. They tried to sue non-profits for \$1000 per employee in damages.





- Zoosk has a website that mobile devices can connect to
- Zoosk's server collects information from the mobile devices, including location and unique device identifiers
- Zoosk users can send and accept invitations to connect with and send messages to each other.
- Zoosk shares profile information of connected users, who are "members of a same social network" (i.e., they're on Zoosk)
- Zoosk can connect users who are in the immediate vicinity of each other, or a particular distance away



# Problem: Open Source is under attack, too!

5 MIN READ

## Ensuring Patents Foster Innovation in Open Source

DAN WHITING | 23 JUNE 2022

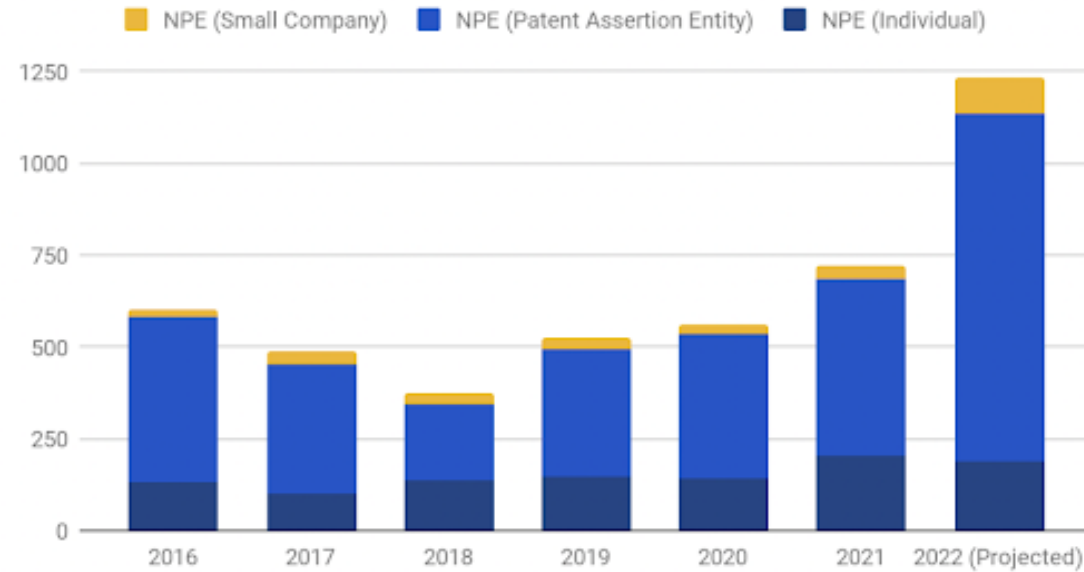
So, I am old enough to remember when the U.S. Congress temporarily intervened in a patent dispute over the technology that powered BlackBerries. A U.S. Federal judge ordered the BlackBerry service to shutdown until the matter was resolved, and Congress determined that BlackBerry service was too integral to commerce to be allowed to be turned off. Eventually, RIM settled the patent dispute and the BlackBerry rode off into technology oblivion.

I am not here to argue the merits of this nearly 20-year-old case (in fact, I coincidentally had friends on both legal teams), but it was when I was introduced to the idea of companies that purchase patents with the goal of using this purchased right to extract money from other companies.

Patents are an important legal protection to foster innovation, but, like all systems, it isn't perfect.

At this week's Open Source Summit North America, we heard from Kevin Jakes with Unified Patents. Kevin is a patent attorney who saw damage being done to innovation by patent trolls - more kindly known as non-practicing entities (NPEs).

## Litigation Targeting Open Source Technologies



\*\*\*Data collected through June 6, 2022\*\*\*

Home / Business / Enterprise Software

## Patent troll attacks against open source projects are up 100% since last year. Here's why

In recent years, patent trolls have started attacking open-source developers and companies. But, the open-source community is fighting back.

Written by Steven Vaughan-Nichols, Senior Contributing Editor on Sept. 12, 2022

## What next?

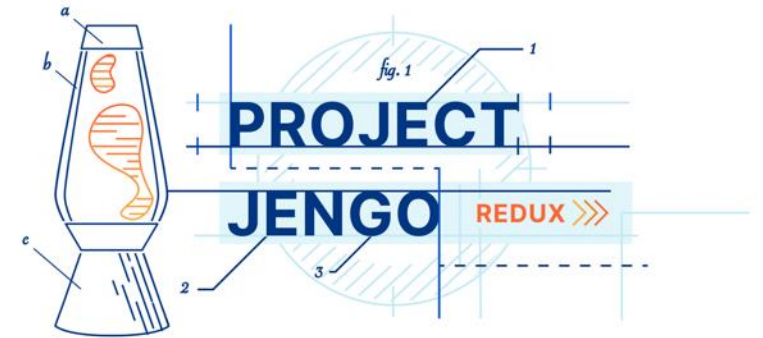
- Alternative licensing models
  - The Defensive Patent License (DPL)
  - The Open Invention Network (OIN)
  - License on Transfer (LOT)
- Bogus patent bounties
- [Unified Patents](#)
- Commonsense reform
- **Abolish software patents?**

### Project Jengo Redux: Cloudflare's Prior Art Search Bounty Returns

04/26/2021



Doug Kramer



# Dependency Management

# Left-pad (March 22, 2016)

OBSSESSIONS

QUARTZ

NPM ERR!

THE VERGE

TECH

REVIEWS

## How one programmer broke the internet by deleting a tiny piece of code

REPORT TECH

### How an irate developer briefly broke JavaScript

*Unpublishing 11 lines of code brought down an open source house of cards*

By Paul Miller | @futurepaul | Mar 24, 2016, 4:29pm EDT



SIGN IN

The Register

{\* SOFTWARE \*}


### How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript

Code pulled from NPM – which everyone was using



# Left-pad (March 22, 2016)

npmjs.org tells me that left-pad is not available (404 page) #4

 Closed silkenrance opened this issue on Mar 22, 2016 · 193 comments



silkenrance commented on Mar 22, 2016

When building projects on travis, or when searching for left-pad on npmjs.com, both will report that the package cannot be found.

Here is an excerpt from the travis build log

```
npm ERR! Linux 3.13.0-40-generic
npm ERR! argv "/home/travis/.nvm/versions/node/v4.2.2/bin/node" "/home/travis/.nvm/versions/node/v4.2.2/bin/npm"
npm ERR! node v4.2.2
npm ERR! npm v2.14.7
npm ERR! code E404
npm ERR! 404 Registry returned 404 for GET on https://registry.npmjs.org/left-pad
npm ERR! 404
npm ERR! 404 'left-pad' is not in the npm registry.
npm ERR! 404 You should bug the author to publish it (or use the name yourself!)
npm ERR! 404 It was specified as a dependency of 'line-numbers'
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.
npm ERR! Please include the following file with any support request:
npm ERR!   /home/travis/build/coldrye-es/pingo/npm-debug.log
make: *** [deps] Error 1
```

And here is the standard npmjs.com error page <https://www.npmjs.com/package/left-pad>

However, if I remove left-pad from my local npm cache and then reinstall it using npm it will happily install left-pad@0.0.4.

 88  3

# Left-pad (Docs)

## left-pad

String left pad

build unknown

## Install

```
$ npm install left-pad
```

## Usage

```
const leftPad = require('left-pad')

leftPad('foo', 5)
// => "  foo"

leftPad('foobar', 6)
// => "foobar"

leftPad(1, 2, '0')
// => "01"

leftPad(17, 5, 0)
// => "00017"
```

## Install

```
> npm i left-pad
```

## Repository

[github.com/stevemao/left-pad](https://github.com/stevemao/left-pad)

## Homepage

[github.com/stevemao/left-pad#readme](https://github.com/stevemao/left-pad#readme)

## Weekly Downloads

2,962,641



## Version

1.3.0

## License

WTFPL

## Unpacked Size

9.75 kB

## Total Files

10

## Issues

3

## Pull Requests

7

## Last publish

4 years ago

# Left-pad (Source Code)

17 lines (11 sloc) | 222 Bytes

```
1  module.exports = leftpad;
2
3  function leftpad (str, len, ch) {
4    str = String(str);
5
6    var i = -1;
7
8    if (!ch && ch !== 0) ch = ' ';
9
10   len = len - str.length;
11
12   while (++i < len) {
13     str = ch + str;
14   }
15
16   return str;
17 }
```

# See also: isArray

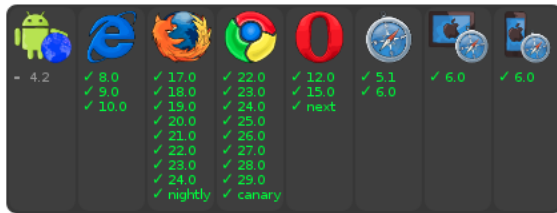
5 lines (4 sloc) | 133 Bytes

```
1 var toString = {}.toString;
2
3 module.exports = Array.isArray || function (arr) {
4   return toString.call(arr) === '[object Array]';
5 };
```

## isarray

Array#isArray for older browsers and deprecated Node.js versions.

build passing downloads 227M/month



Just use `Array.isArray` directly, unless you need to support those older versions.

## Usage

```
var isArray = require('isarray');

console.log(isArray([])); // => true
console.log(isArray({})); // => false
```

## Install

```
> npm i isarray
```

## Repository

[github.com/juliangruber/isarray](https://github.com/juliangruber/isarray)

## Homepage

[github.com/juliangruber/isarray](https://github.com/juliangruber/isarray)

## Weekly Downloads

50,913,317

Version	License
2.0.5	MIT

Unpacked Size	Total Files
3.43 kB	4

Issues	Pull Requests
4	3



**larissa tyagi** 11:10 AM

Hey! I'm getting this error in my LLM file in the Test robustness question. Would this likely mean that something went wrong with my vertex ai installation?

Screenshot 2024-03-27 at 2.09.41 pm.png ▾



**Alexis Axon** 7 days ago



**Deon Kouatchou** 7 days ago

I found a temporary solution where you import from the `vertexai.language_models` module instead of `vertexai.preview.language_models`. I haven't gotten to this part of the project so I don't know if the decorator will work and if the modules inherit the same functionalities.



**Kevin He** 3 days ago

You can also try using a later version and running the following code:

```
!pip uninstall bigframes
```

```
!pip install bigframes==0.26.0
```



d the  
ne error  
as a bug  
dated  
e tests  
sert

# Dependency Management

- It's hard
- It's mostly a mess (everywhere)
- But it's critical to modern software development

# What is a Dependency?

- Core of what most build systems do
  - “Compile” and “Run Tests” is just a fraction of their job
- Examples: Maven, Gradle, NPM, Bazel, ...
- **Foo->Bar**: To build Foo, you may need to have a built version of Bar
- Dependency Scopes:
  - **Compile**: Foo uses classes, functions, etc. defined by Bar
  - **Runtime**: Foo uses an abstract API whose implementation is provided by Bar (e.g. logging, database, network or other I/O)
  - **Test**: Foo needs Bar only for tests (e.g. JUnit, mocks)
- Internal vs. External Dependencies
  - Is Bar also built/maintained by your org or is it pulled from elsewhere using a package manager?

# Dependencies: Example

```
github.com/CMU-313/Teedy/blob/main/pom.xml
152 <dependencyManagement>
153 <dependencies>
154 <dependency>
155 <groupId>com.sismics.docs</groupId>
156 <artifactId>docs-core</artifactId>
157 <version>${project.version}</version>
158 </dependency>
159
160 <dependency>
161 <groupId>com.sismics.docs</groupId>
162 <artifactId>docs-web-common</artifactId>
163 <version>${project.version}</version>
164 </dependency>
165
166 <dependency>
167 <groupId>com.sismics.docs</groupId>
168 <artifactId>docs-web-common</artifactId>
169 <type>test-jar</type>
170 <version>${project.version}</version>
171 </dependency>
172
173 <dependency>
174 <groupId>com.sismics.docs</groupId>
175 <artifactId>docs-web</artifactId>
176 <version>${project.version}</version>
177 </dependency>
178
179 <dependency>
180 <groupId>org.eclipse.jetty</groupId>
181 <artifactId>jetty-server</artifactId>
182 <version>${org.eclipse.jetty.jetty-server.version}</version>
183 </dependency>
184
185 <dependency>
186 <groupId>org.eclipse.jetty</groupId>
187 <artifactId>jetty-webapp</artifactId>
188 <version>${org.eclipse.jetty.jetty-webapp.version}</version>
189 </dependency>
```

Package: git (1:2.17.1-1ubuntu0.9 :

fast, scalable, distributed revision control system

Other Packages Related to git

- depends
  - ◆ recommends
  - suggests
  - enhances
- **git-man** (<< 1:2.17.0-.) [not amd64, i386]  
fast, scalable, distributed revision control system (manual pages)
  - **git-man** (<< 1:2.17.1-.) [amd64, i386]
  - **git-man** (>> 1:2.17.0) [not amd64, i386]
  - **git-man** (>> 1:2.17.1) [amd64, i386]
  - **libc6** (>= 2.16) [not arm64, ppc64el]  
GNU C Library: Shared libraries  
also a virtual package provided by **libc6-udeb**
  - **libc6** (>= 2.17) [arm64, ppc64el]
  - **libcurl3-gnutls** (>= 7.16.2)  
easy-to-use client-side URL transfer library (GnuTLS flavour)
  - **liberror-perl**  
Perl module for error/exception handling in an OO-ish way
  - **libexpat1** (>= 2.0.1)  
XML parsing C library - runtime library
  - **libpcre3**  
Old Perl 5 Compatible Regular Expression Library - runtime files
  - **perl**  
Larry Wall's Practical Extraction and Report Language
  - **zlib1g** (>= 1:1.2.0)  
compression library - runtime
  - ◆ **less**  
pager program similar to more
  - ◆ **patch**  
Apply a diff file to an original
  - ◆ **ssh-client**  
virtual package provided by **openssh-client**



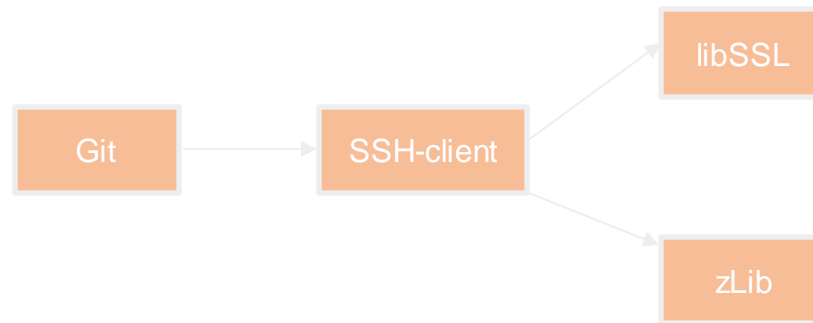
# Where are the dependencies hosted?

- Typically downloaded from dependency servers:
  - Maven Central (<https://repo.maven.apache.org/maven2/>)
  - Ubuntu Packages for Apt (<https://packages.ubuntu.com/>)
  - Python Package Index (<https://pypi.org/>) ]
  - NPM Public Registry (<https://registry.npmjs.org/>)
- Packages need a unique identifier
  - Typically a package name (sometimes owner name) and version
- Custom repositories allowed by most package managers
  - Often used for company-internal packages or cache mirroring
  - Note problems with duplicates (same package name in different repositories; some priority order is needed)
- Somebody needs to manage repositories
  - Availability: Repository needs to be running
  - Access Control: Packages should only be published by owners
  - Integrity: Packages should be signed or otherwise verifiable
  - Uniqueness and archival: Only one artifact per version
  - Traceability: Packages can have metadata pointing to source or tests
  - Security: ???



# Transitive Dependencies

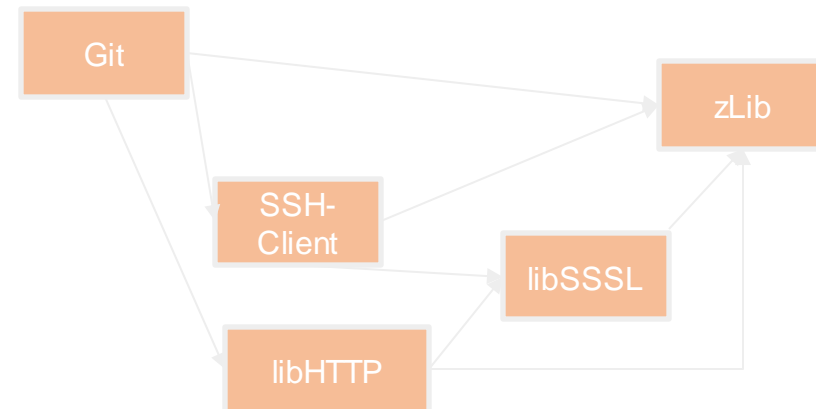
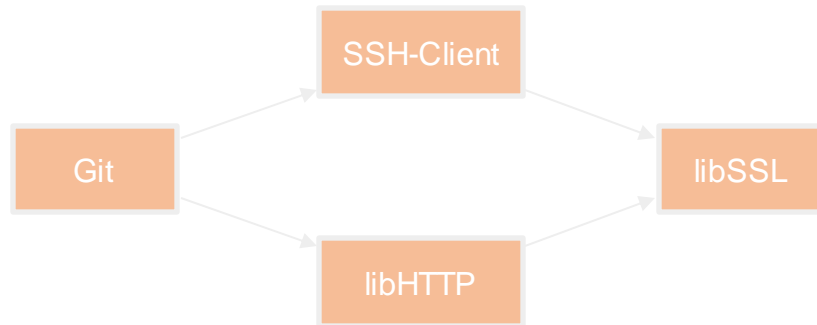
Packages can depend on other packages



Q: Should Git be able to use exports of libSSL (e.g. certificate management) or zLib (e.g. gzip compression)?

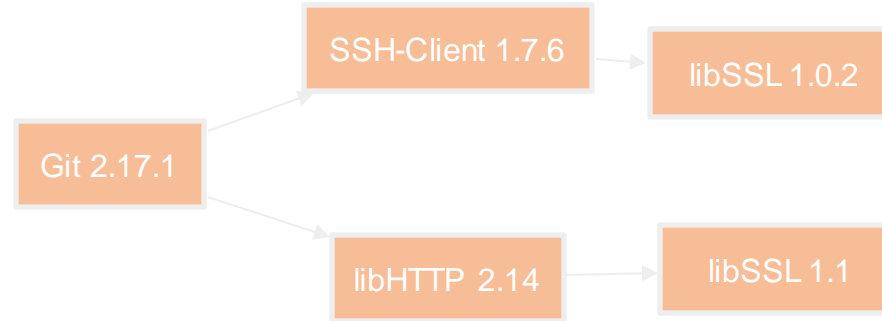
# Diamond Dependencies

What are some problems when multiple intermediate dependencies have the same transitive dependency?



# Diamond Dependencies

What are some problems when multiple intermediate dependencies have the same transitive dependency?



# Resolutions to the Diamond Problem

1. Duplicate it!
  - Doesn't work with static linking (e.g. C/C++), but may be doable with Java (e.g. using ClassLoader hacking or package renaming)
  - Values of types defined by duplicated libraries cannot be exchanged across
2. Ban transitive dependencies; just use a global list with one version for each
  - Challenge: Keeping things in sync with latest
  - Challenge: Deciding which version of transitive deps to keep
3. Newest version (keep everything at latest)
  - Requires ordering semantics
  - Intermediate dependency may break with update to transitive
4. Oldest version (lowest denominator)
  - Also requires ordering semantics
  - Sacrifices new functionality
5. Oldest non-breaking version / Newest non-breaking version
  - Requires faith in tests or semantic versioning contract

# Semantic Versioning

- Widely used convention for versioning releases
  - E.g. 1.2.1, 3.1.0-alpha-1, 3.1.0-alpha-2, 3.1.0-beta-1, 3.1.0-rc1
- Format: {MAJOR} . {MINOR} . {PATCH}
- Each component is ordered (numerically, then lexicographically; release-aware)
  - $1.2.1 < 1.10.1$
  - $3.1.0\text{-alpha-1} < 3.1.0\text{-alpha-2} < 3.1.0\text{-beta-1} < 3.1.0\text{-rc1} < 3.1.0$
- Contracts:
  - MAJOR updated to indicate breaking changes
    - Same MAJOR version => backward compatibility
  - MINOR updated for additive changes
    - Same MINOR version => API compatibility (important for linking)
  - PATCH updates functionality without new API
    - Ninja edit; usually for bug fixes



<https://semver.org/>

[2.0.0](#) [2.0.0-rc.2](#) [2.0.0-rc.1](#) [1.0.0](#) [1.0.0-beta](#)

# Semantic Versioning 2.0.0

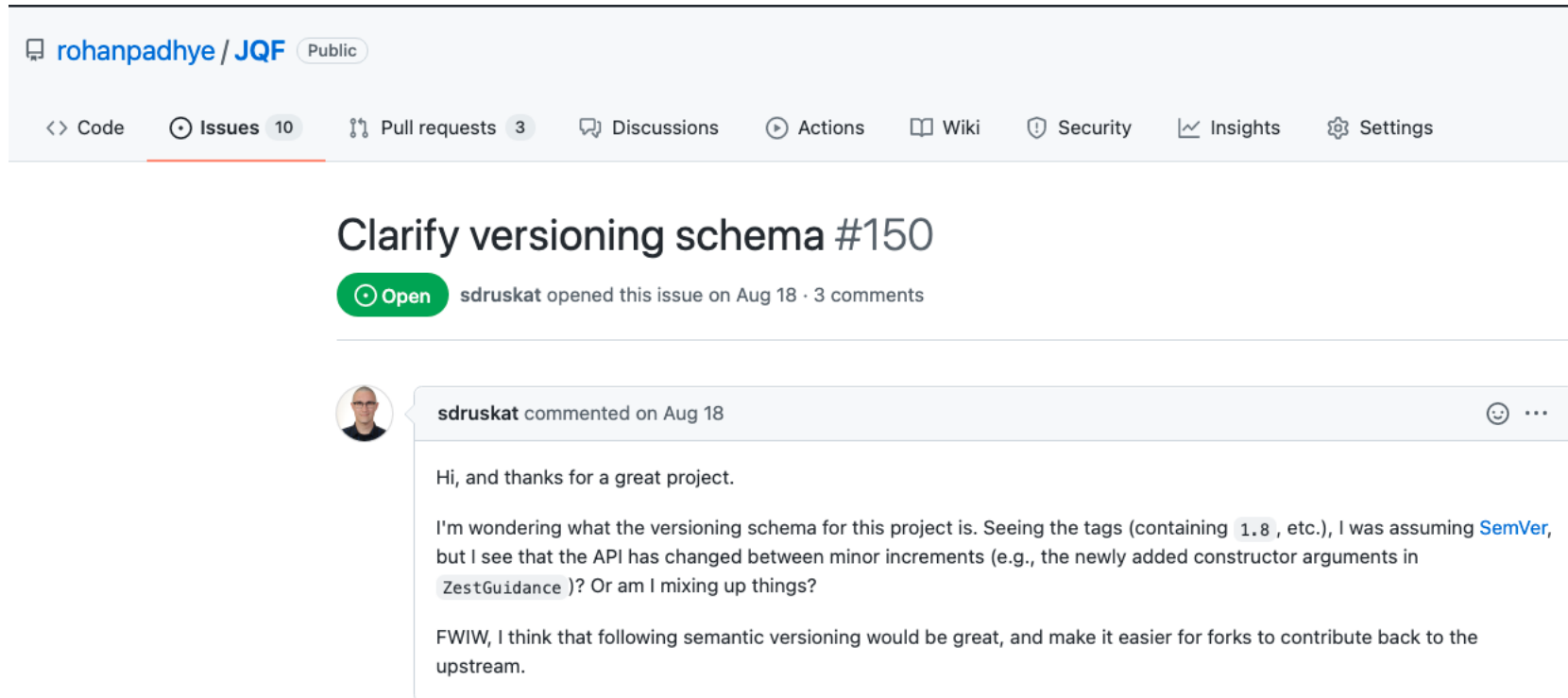
## Summary

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards compatible manner, and
3. PATCH version when you make backwards compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

# People rely on SemVer contracts



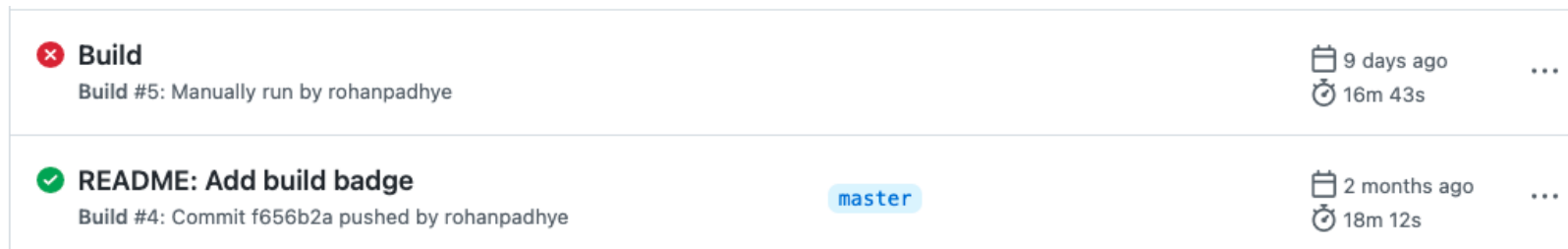
The screenshot shows a GitHub repository page for 'rohanpadhye / JQF' with a 'Public' label. The navigation bar includes links for Code, Issues (10), Pull requests (3), Discussions, Actions, Wiki, Security, Insights, and Settings. The main content area displays an issue titled 'Clarify versioning schema #150' which is 'Open'. It was opened by 'sdruskat' on August 18 and has 3 comments. A comment from 'sdruskat' is visible, dated August 18. The comment text is: 'Hi, and thanks for a great project. I'm wondering what the versioning schema for this project is. Seeing the tags (containing 1.8, etc.), I was assuming SemVer, but I see that the API has changed between minor increments (e.g., the newly added constructor arguments in ZestGuidance)? Or am I mixing up things? FWIW, I think that following semantic versioning would be great, and make it easier for forks to contribute back to the upstream.'

# Dependency Constraints

- E.g. Declare dependency on "Bar > 2.1"
  - Bar 2.1.0, 2.1.1, 2.2.0, 2.9.0, etc. all match
  - 2.0.x does NOT match
  - 3.0.x does NOT match
- Diamond dependency problem can be resolved using SAT solvers
  - E.g. Foo 1.0.0 depends on "Bar >= 2.1" and "Baz 1.8.x"
    - Bar 2.1.0 depends on "Qux [1.6, 1.7]"
    - Bar 2.1.1 depends on "Qux 1.7.0"
    - Baz 1.8.0 depends on "Qux 1.5.x"
    - Baz 1.8.1 depends on "Qux 1.6.x"
  - Find an assignment such that all dependencies are satisfied
    - Solution: Use Bar 2.1.0, Baz 1.8.1, and Qux 1.6.{latest}

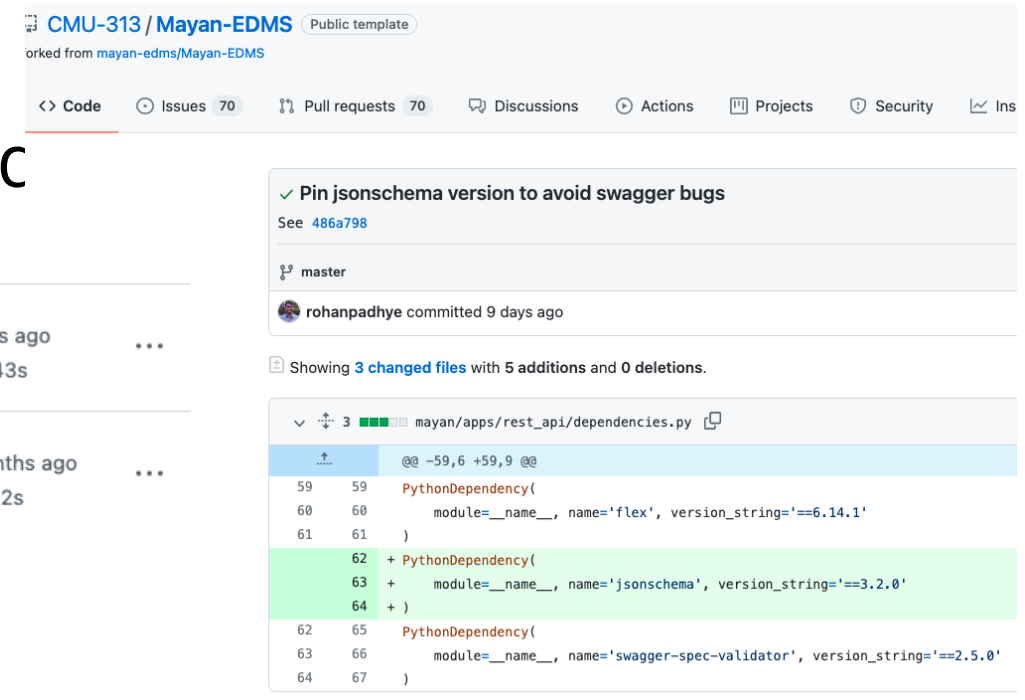
# Semantic Versioning Contracts

- Largely trusting developers to maintain them
- Constrained/range dependencies can cause unexpected build failures
- Automatic validation of SemVer is hard



A screenshot of the GitHub Actions workflow runs page. It shows two workflow runs:

- Build** (failed): Build #5: Manually run by rohanpadhye. Status: Failed (red X). Duration: 16m 43s. Triggered 9 days ago.
- README: Add build badge** (succeeded): Build #4: Commit f656b2a pushed by rohanpadhye. Status: Succeeded (green checkmark). Duration: 18m 12s. Triggered 2 months ago. Branch: master.



A screenshot of a GitHub pull request for the repository CMU-313 / Mayan-EDMS. The pull request is titled "Pin jsonschema version to avoid swagger bugs" and is on the master branch. It was committed by rohanpadhye 9 days ago. The pull request shows 3 changed files with 5 additions and 0 deletions. The code diff for the file `mayan/apps/rest_api/dependencies.py` is shown, highlighting the addition of a `PythonDependency` for `jsonschema` with version `==3.2.0`.

```
@@ -59,6 +59,9 @@
59 59 PythonDependency(
60 60     module=__name__, name='flex', version_string=='6.14.1'
61 61 )
62 + PythonDependency(
63 +     module=__name__, name='jsonschema', version_string=='3.2.0'
64 + )
62 65 PythonDependency(
63 66     module=__name__, name='swagger-spec-validator', version_string=='2.5.0'
64 67 )
```

# Cyclic Dependencies

- A very bad thing
- Avoid at all costs
- Sometimes unavoidable or intentional
  - E.g. GCC is written in C (needs a C compiler)
  - E.g. Apache Maven uses the Maven build system
  - E.g. JDK tested using JUnit, which requires the JDK to compile



# Cyclic Dependencies

- Bootstrapping: Break cycles over time
- Assume older version exists in binary (pre-built form)
- Step 1: Build A using an older version of B
- Step 2: Build B using new (just built) version of A
- Step 3: Rebuild A using new (just built) version of B
- Now, both A and B have been built with new versions of their dependencies
- Doesn't work if both A and B need new features of each other at the same time (otherwise Step 1 won't work)
  - Assumes incremental dependence on new features
- How was the old version built in the first place? (it's turtles all the way down)
  - Assumption: cycles did not exist in the past
  - Successfully applied in compilers (e.g. GCC is written in C)



# Dependency Security

- Will you let strangers execute arbitrary code on your laptop?
  - Think about this every time you do “pip install” or “npm install” or “apt-get upgrade” or “brew upgrade” or whatever (esp. with sudo)
  - Scary, right? Who are you trusting? Why?
- Typo squatting (“pip install numpi”)
- Outright malice (remember the *event-stream* incident?)
- Genuine security vulnerabilities due to software bugs

Dependabot alerts / #74

## Deserialization of Untrusted Data in Apache Log4j #74

Dismiss alert ▾

 Open Opened 3 days ago on log4j:log4j (Maven) · pom.xml

Package	Affected versions	Patched version
 log4j:log4j (Maven)	<= 1.2.17	None

[CVE-2020-9493](#) identified a deserialization issue that was present in Apache Chainsaw. Prior to Chainsaw V2.0 Chainsaw was a component of Apache Log4j 1.2.x where the same issue exists.

Users are advised to migrate from `log4j:log4j` to `org.apache.logging.log4j:log4j` for an updated version of the library.

 dependabot bot opened this 3 days ago

Severity

Critical 9.8 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Weaknesses

CWE-502

# Takeaways

- Dependency management is hard.