



Software Composition Analysis

Artemie Jurgenson -- Solutions Architect



100:1

developers outnumber application security

TIME TO RESPOND BEFORE EXPLOIT

Source: Adapted from IBM X-Force / Analysis by Gartner Research (September 2016)



Equifax was not alone

March 7

Apache Struts releases updated version to thwart vulnerability CVE-2017-5638

March 9

Cisco observes "a high number of exploitation events."



March 13

Okinawa Power
Japan Post



March '18

India's AADHAAR



April 13

India Post

3 Days in March

The Rest of the Story



March 8

NSA reveals Pentagon servers scanned by nation-states for vulnerable Struts instances

Struts exploit published to Exploit-DB.



March 10

Equifax



Canada Revenue Agency



Canada Statistics



GMO Payment Gateway

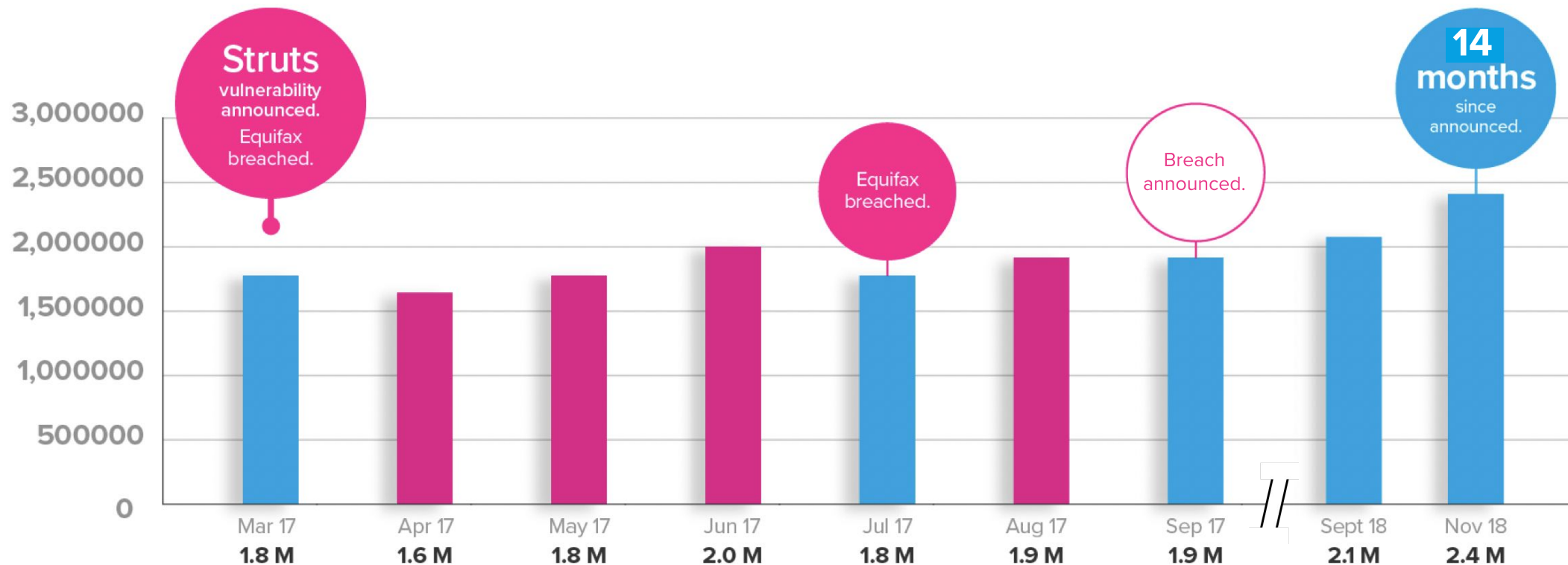
December '17

Monero Crypto Mining

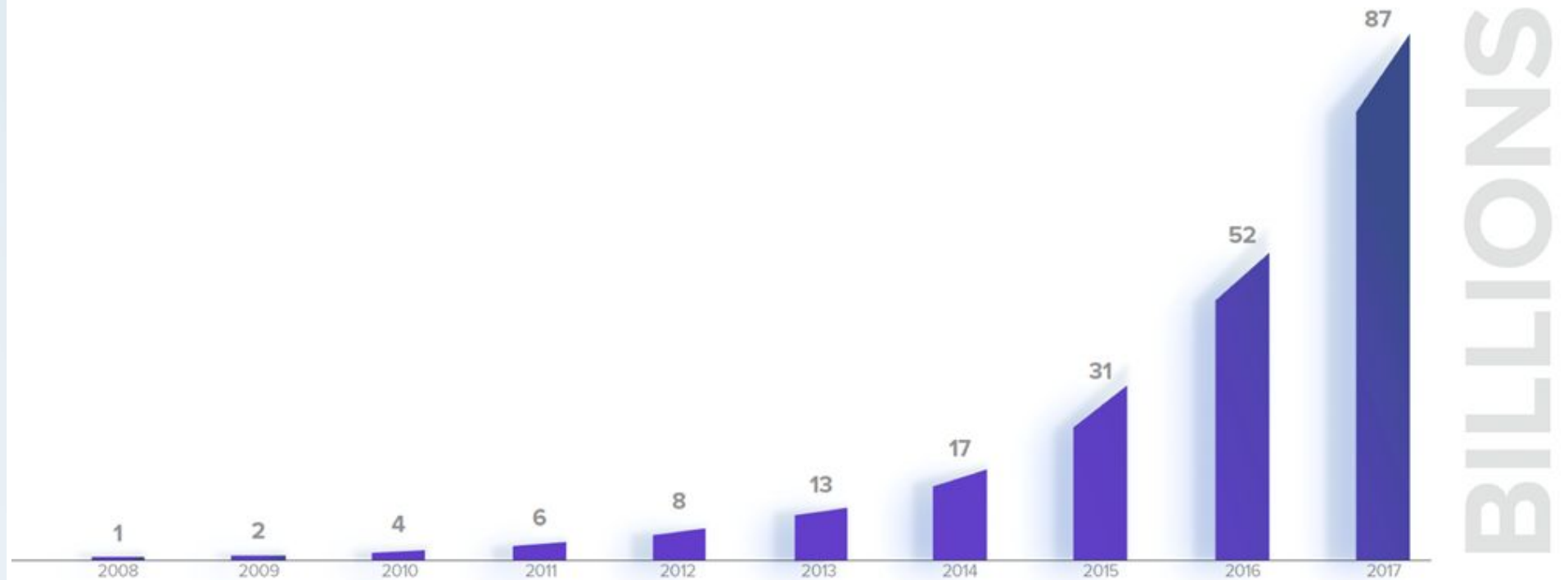
Today

65% of the Fortune 100 download vulnerable versions

18,126 organizations downloading vulnerable versions of Struts

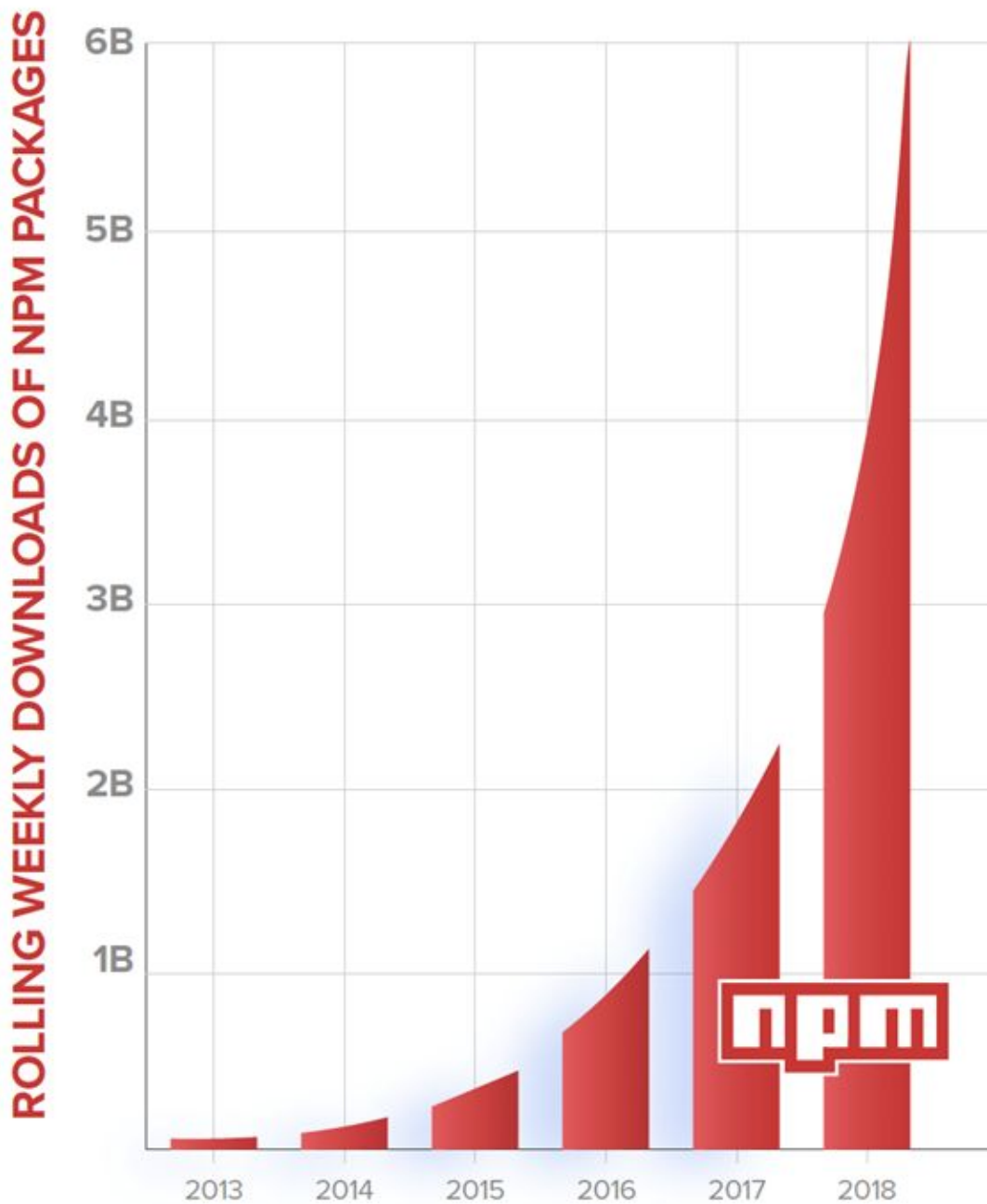


Downloads: Maven Central



Download Requests for Java Components 2008-2017 are also a proxy for the popularity of automated software development.

Downloads:
npmjs



Cryptocurrency and Cryptominers

“I have nothing of value in my application.”

Your **servers** have CPU cycles

Your users' **browsers** have CPU cycles

Your **build infrastructure** has CPU cycles

Cryptominers allow any environment to be directly monetized.

In 2016, the illicit drug trade was worth **\$435 Billion** globally.

It still is.

Cybercrime in 2016: **\$450 Billion**

Cybercrime in 2018: **~\$1.5 Trillion**

“I’m a CISO/CIO and I already have an AppSec program...”

“We’ve been doing SAST/DAST/pen testing for years!”

...

Technologies from an nascent era where most code was **proprietary**.

Would Equifax’s CISO been saved by **static code analysis**?

Who is **ultimately responsible** for risk in open source software?

Are you **fixing** things or **justifying** why you shouldn’t?

Your Developers are in Procurement. Do they know that?



85%
of your code is
sourced from
external suppliers



NET-NEW PROCUREMENT

- Block all new stuff that violates org wide policy
- Enforcement at earliest point in supply chain



BINARY & CONFIGURATION MGMT

- Store proxied and released components
- Single point of access for all languages
- Access controls enable limiting visibility
- Proxying enables replication



DEPENDENCY MGMT

- Assess Tech Debt
- Inventory Apps
- Prioritize by Tech Debt
- Reduce complexity and pick the best packages



SOURCE CONTROL



DEVELOPMENT



CI



RELEASE

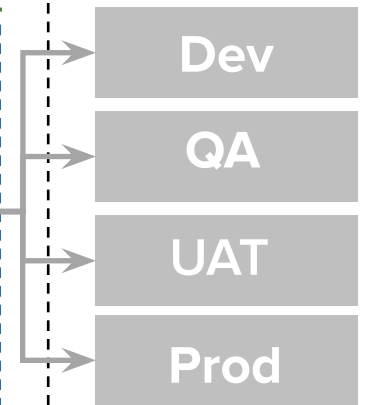


ARTIFACT REPOSITORY

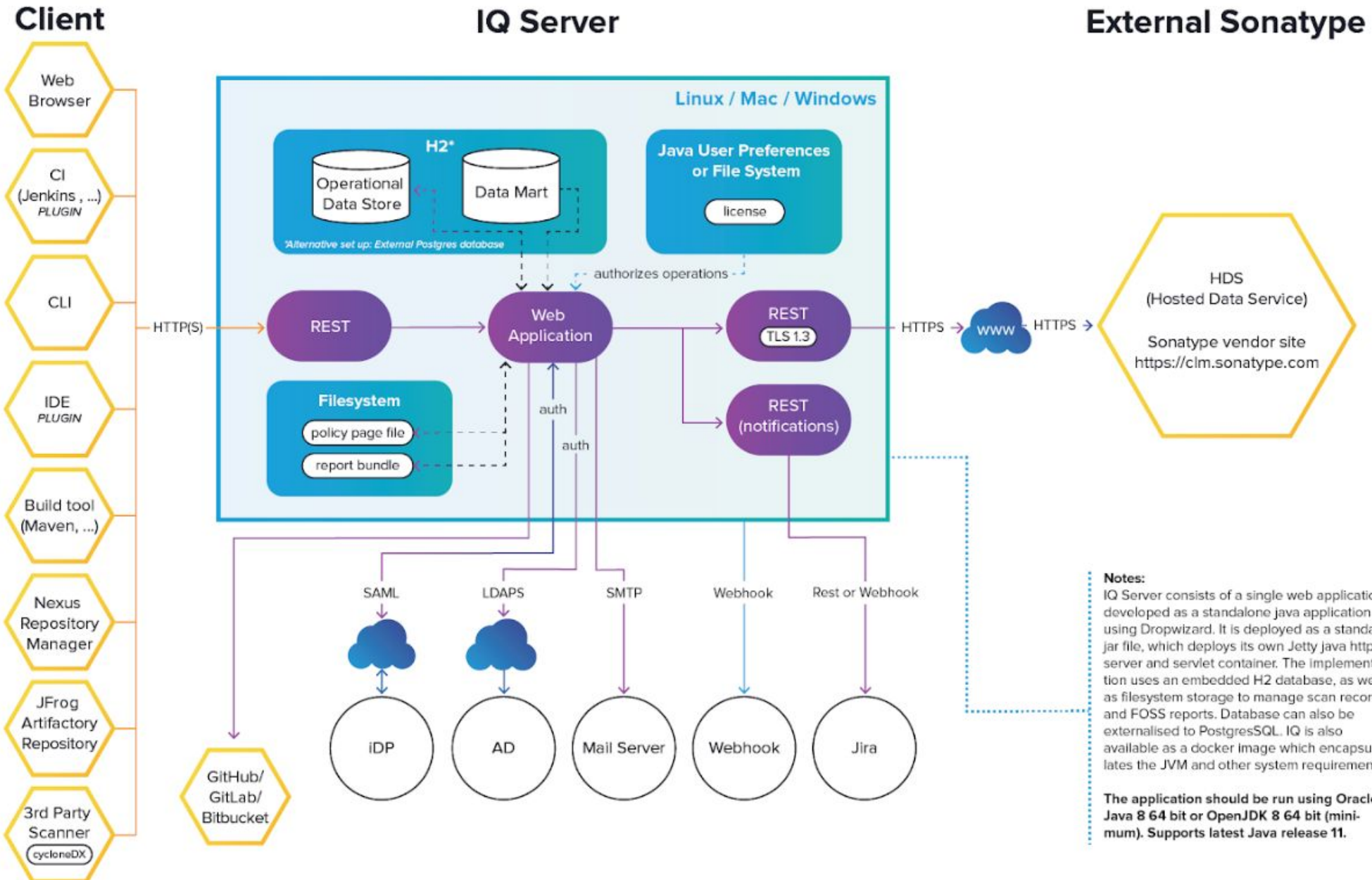


CONTINUOUS MONITORING

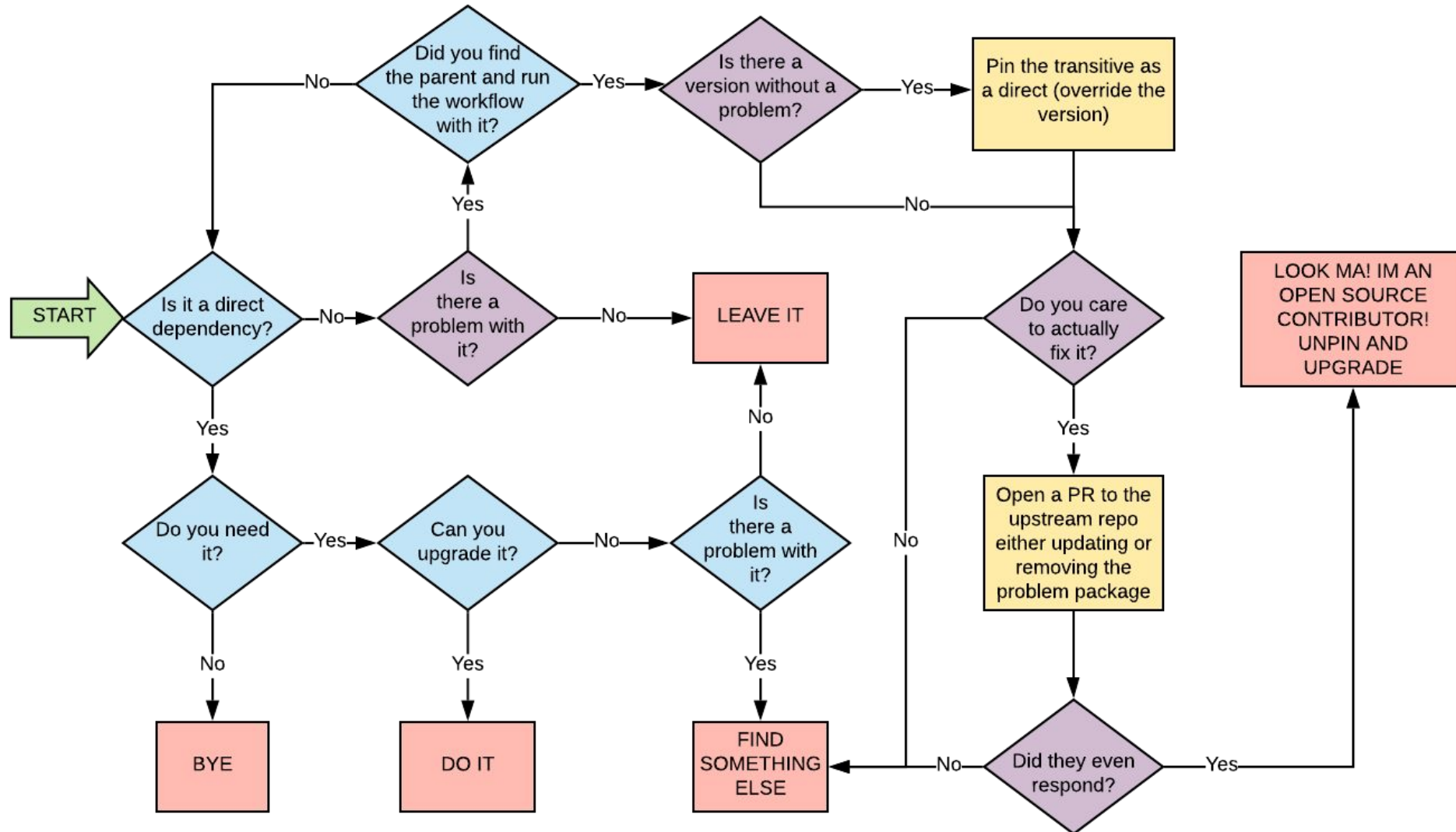
- Automatically fetch new vuln data every day
- Query historical SBOMS for known bad components




IQ Server Reference Architecture



Top-down Dependency Management





In a perfect
world

- Internal libraries are reusable packages
- Build and deploy automation are abstracted
- Inventory of entire codebase
(Software Bills of Material)
- Workflows defined for:
 - Preventing accrual of new tech debt
 - Dependency remediation by existing tech debt
 - Zero-day vulnerabilities
- Incremental improvement of component quality
- Centralized management of a complex software supply chain that crosses business units, classification levels, assets, contractors, external actors

