

Integration Challenges in Static Analysis and Verification

HCSS 2020

Stephen Magill | CEO, MuseDev

INTEGRATION STORY



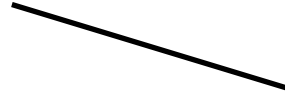
The ROFL (Report Only Failure List) Assumption: All an analysis needs to do is report only a failure list, with low false positives, in order to be effective.

“Soon after the ROFL episode we switched Infer on at diff time. The response of engineers was just as stunning: the fix rate rocketed to over 70%. The same program analysis, with same false positive rate, had much greater impact when deployed at diff time.”

From: *Peter W. O'Hearn. 2018. Continuous Reasoning: Scaling the impact of formal methods. In ACM/IEEE Symposium on Logic in Computer Science (LICS '18).*



Code
Review





Code
Review



Infer

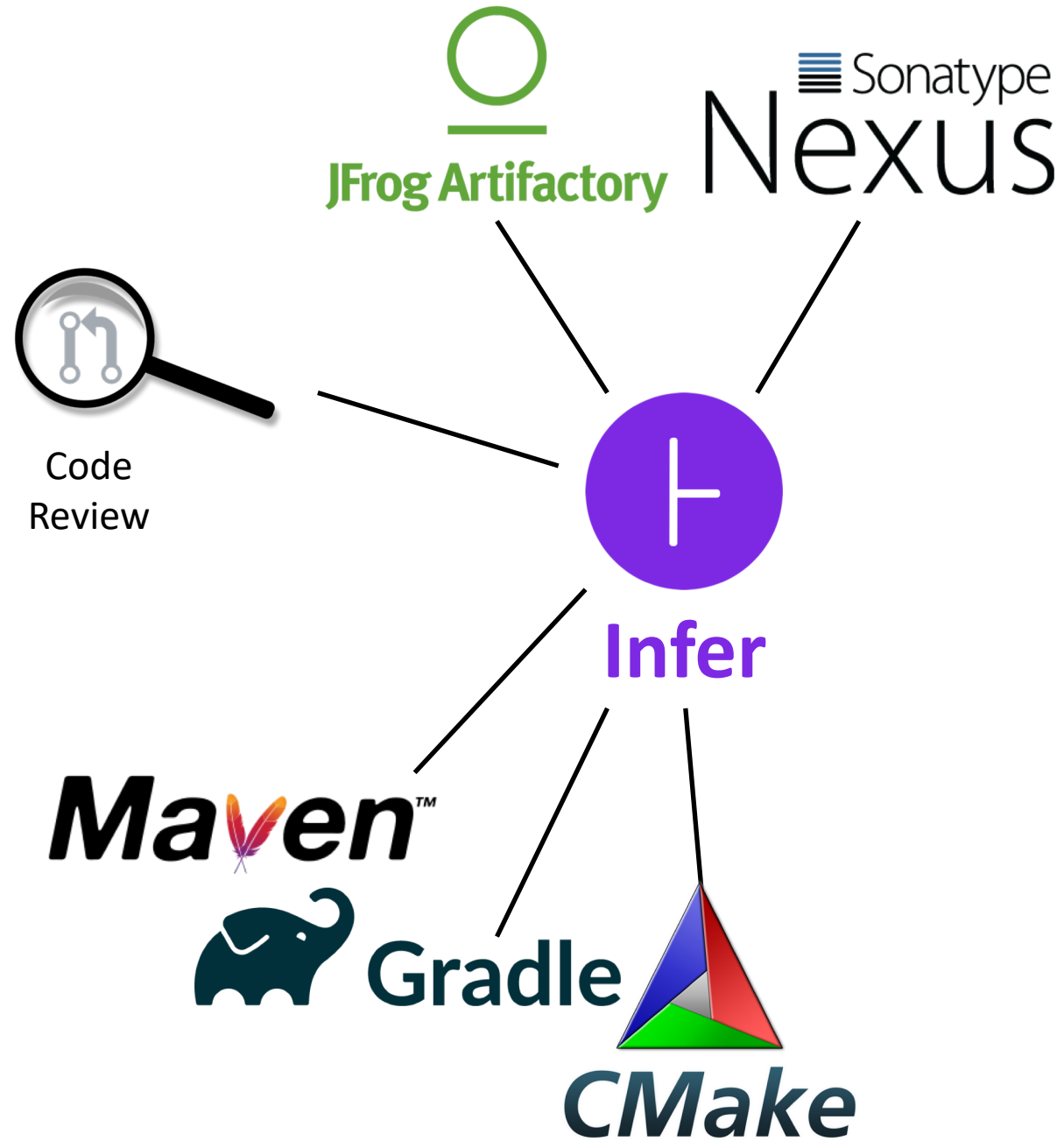
Maven[™]

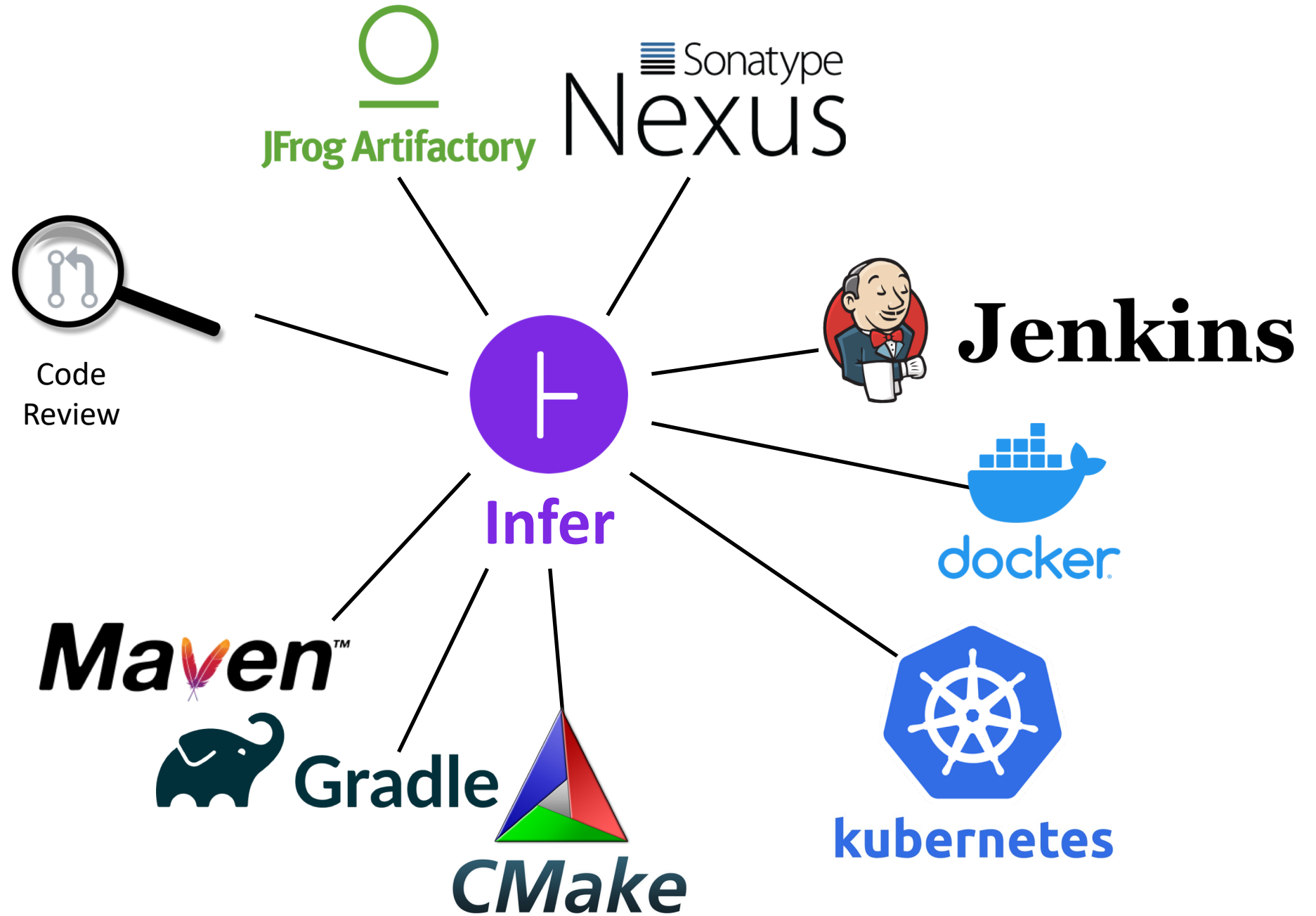


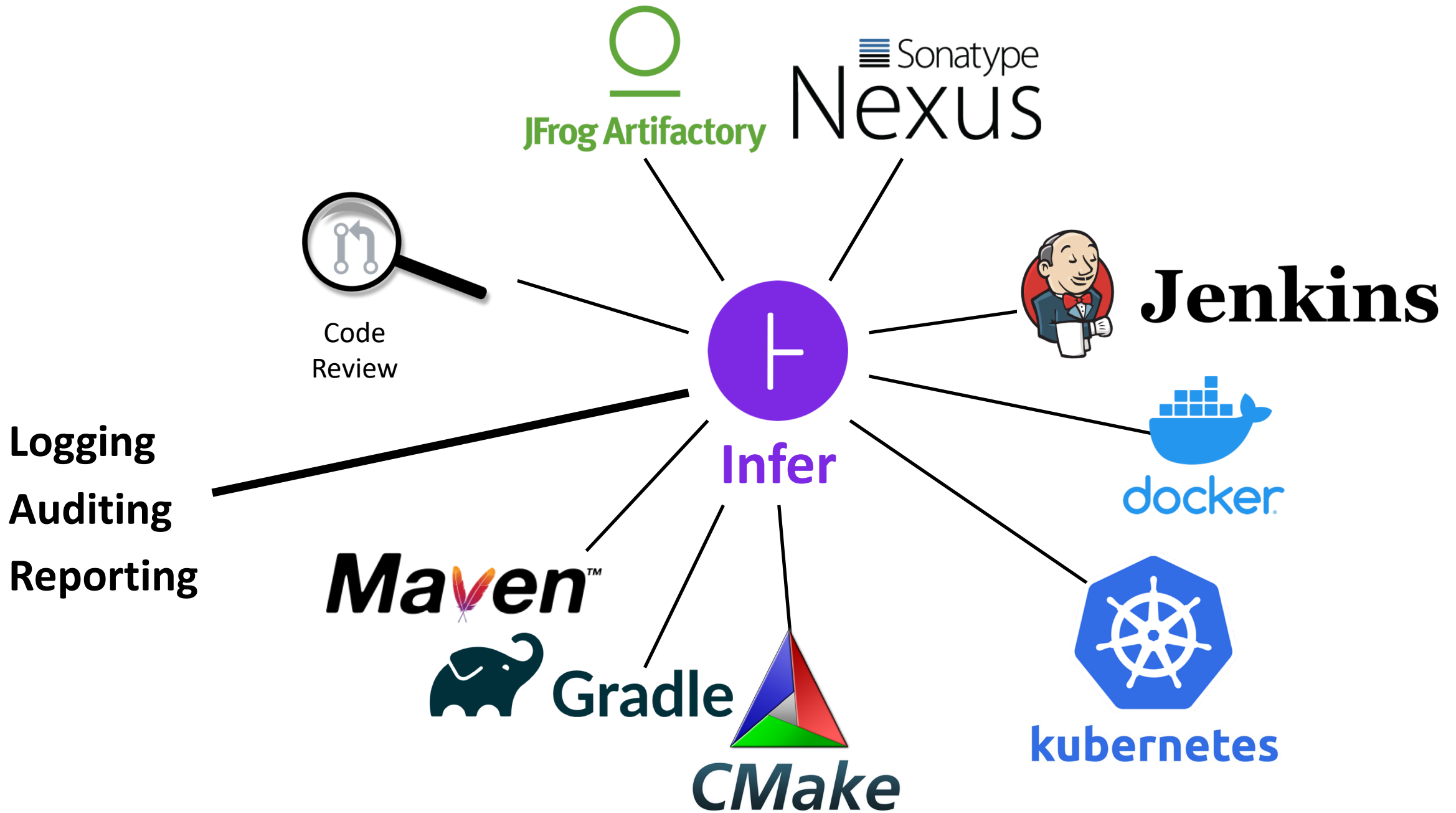
Gradle

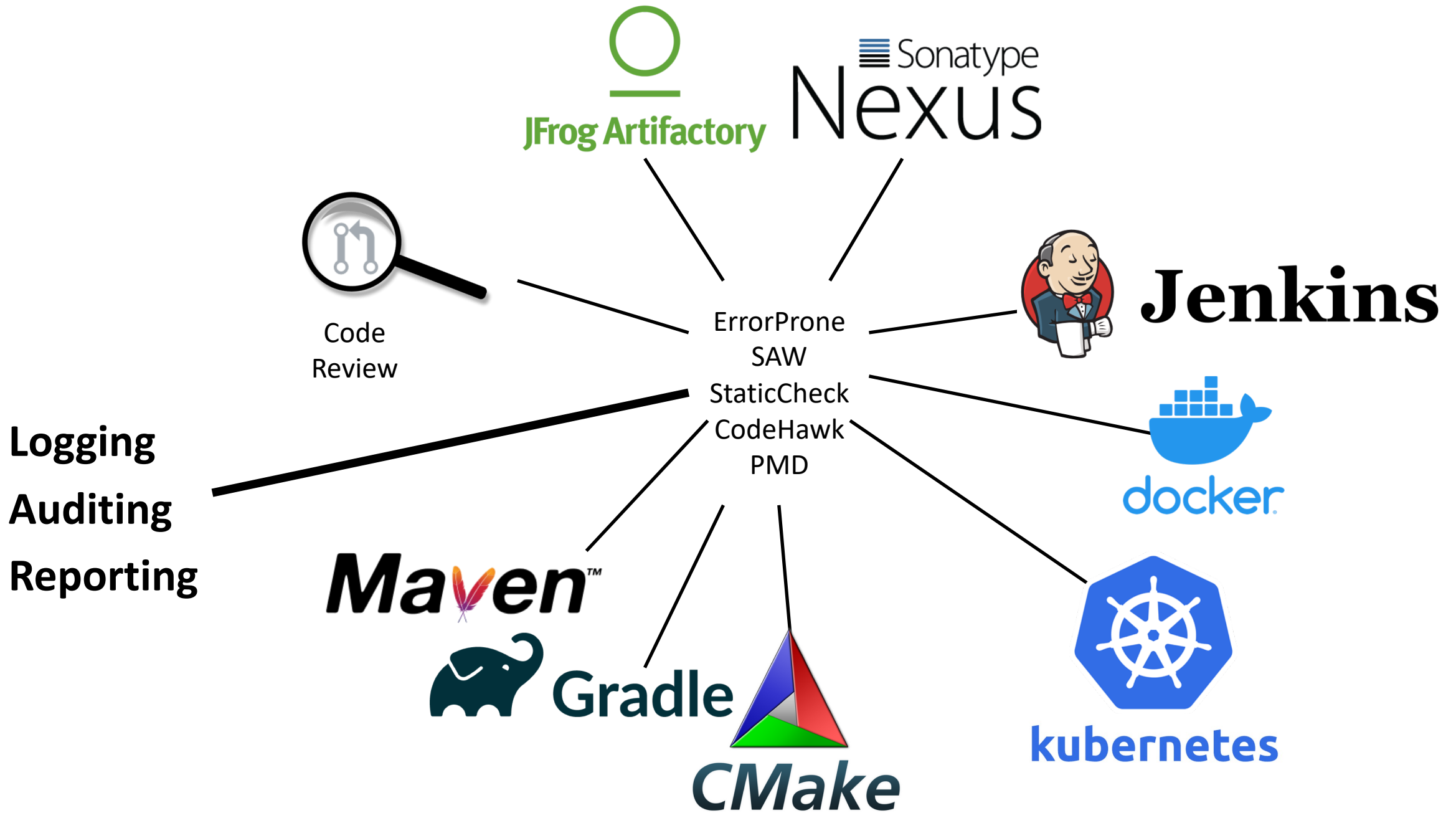


CMake









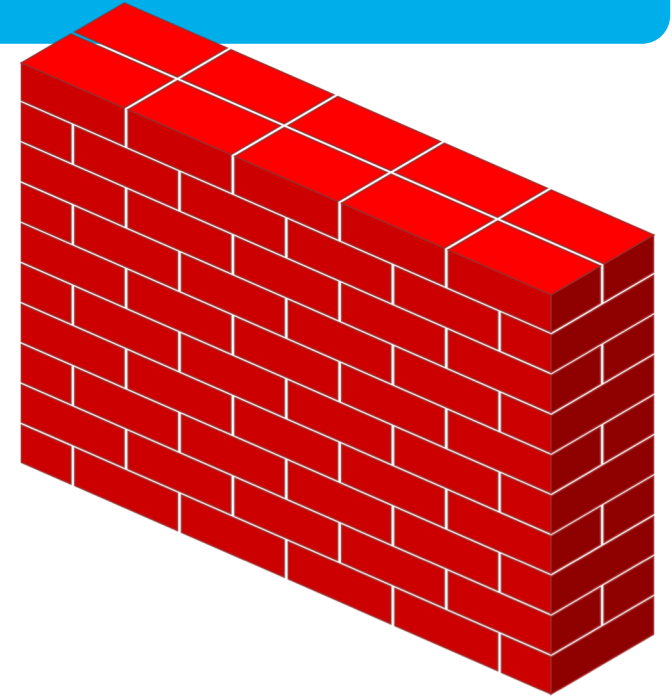
BACK TO THE FUTURE OF STATIC ANALYSIS



Coverity noted these challenges years ago

WHY?

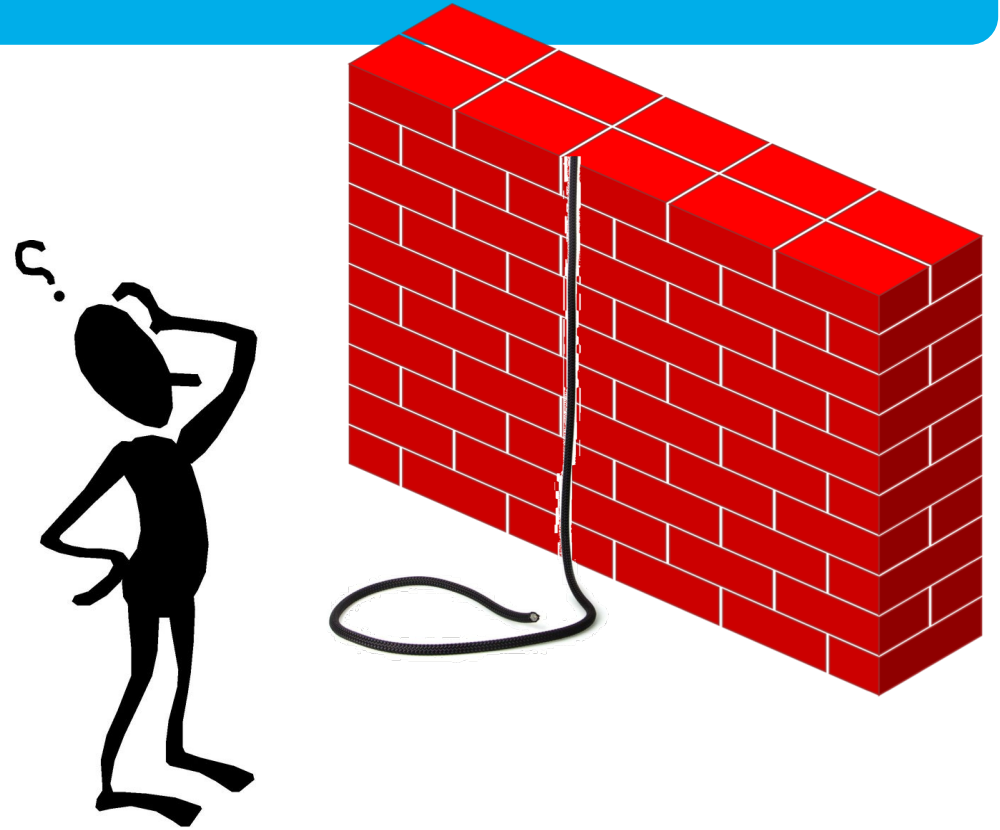
I like a challenge



WHY?

I like a challenge

There is hope

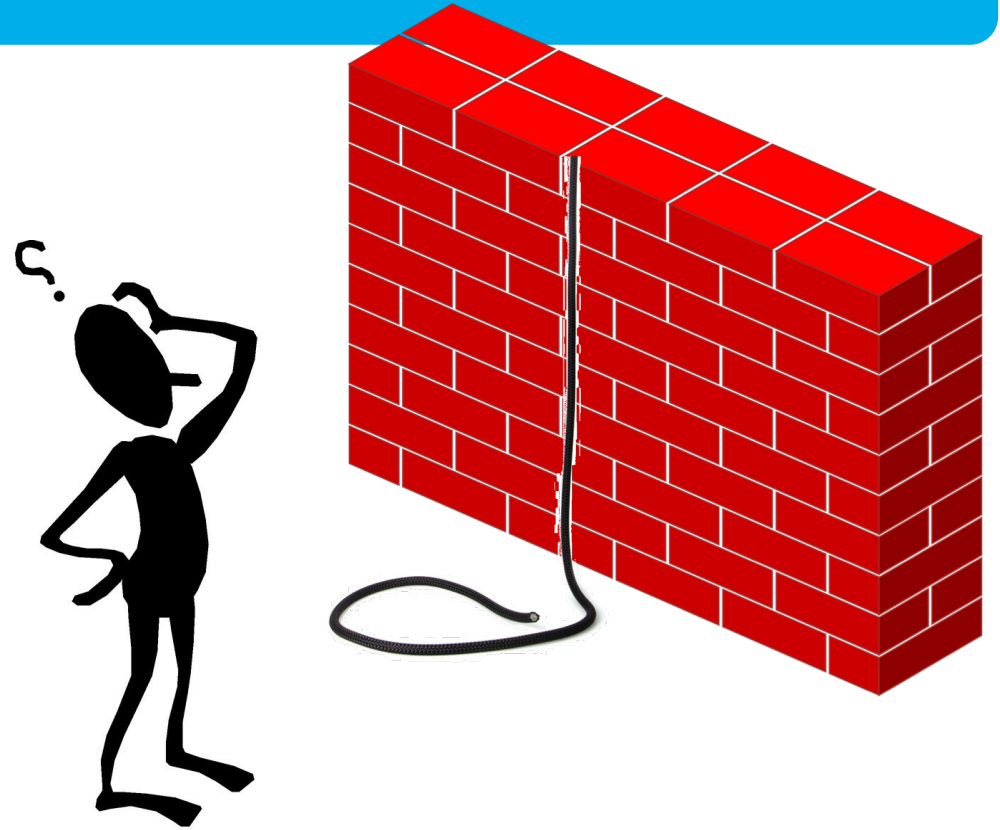


WHY?

I like a challenge

There is hope

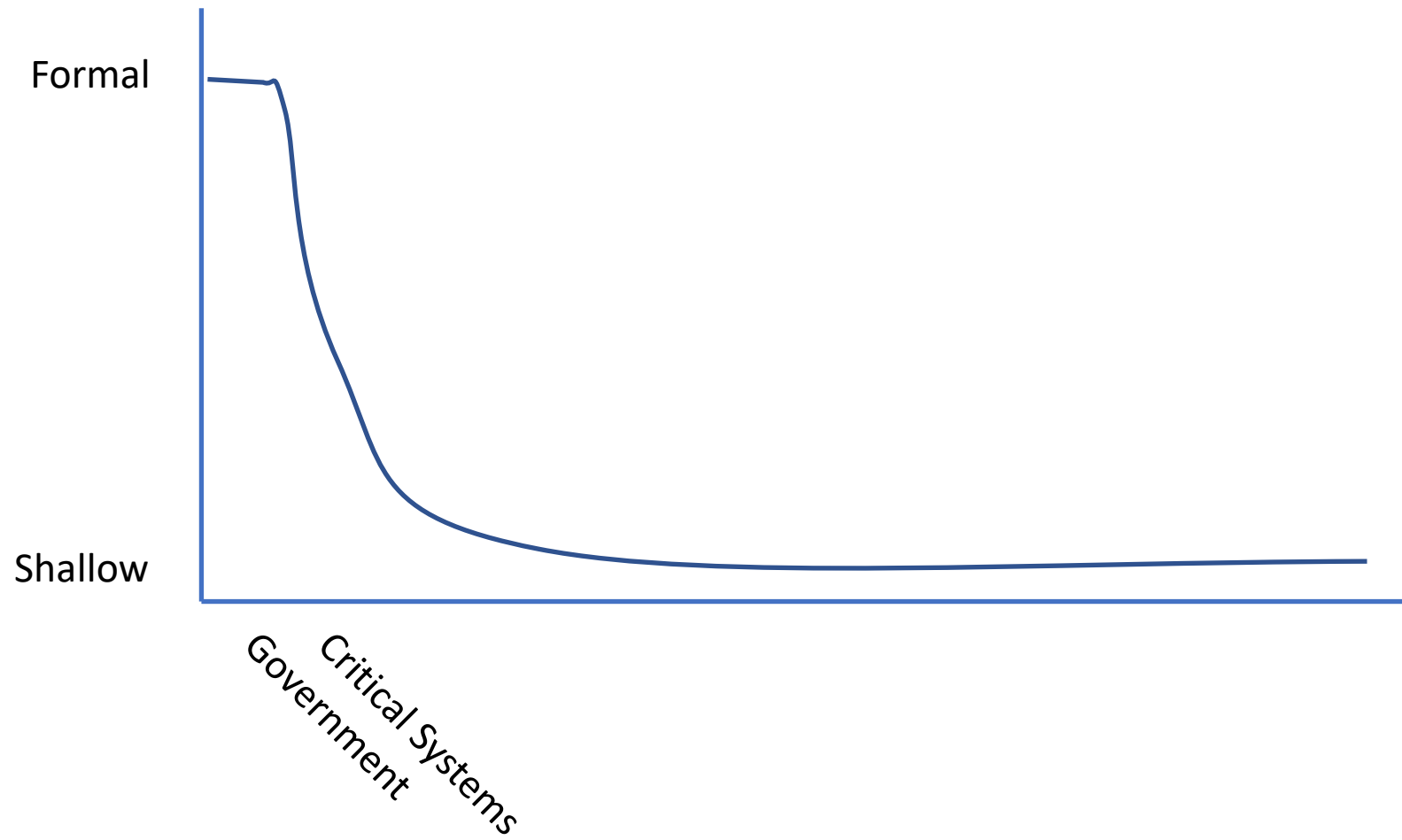
It's necessary



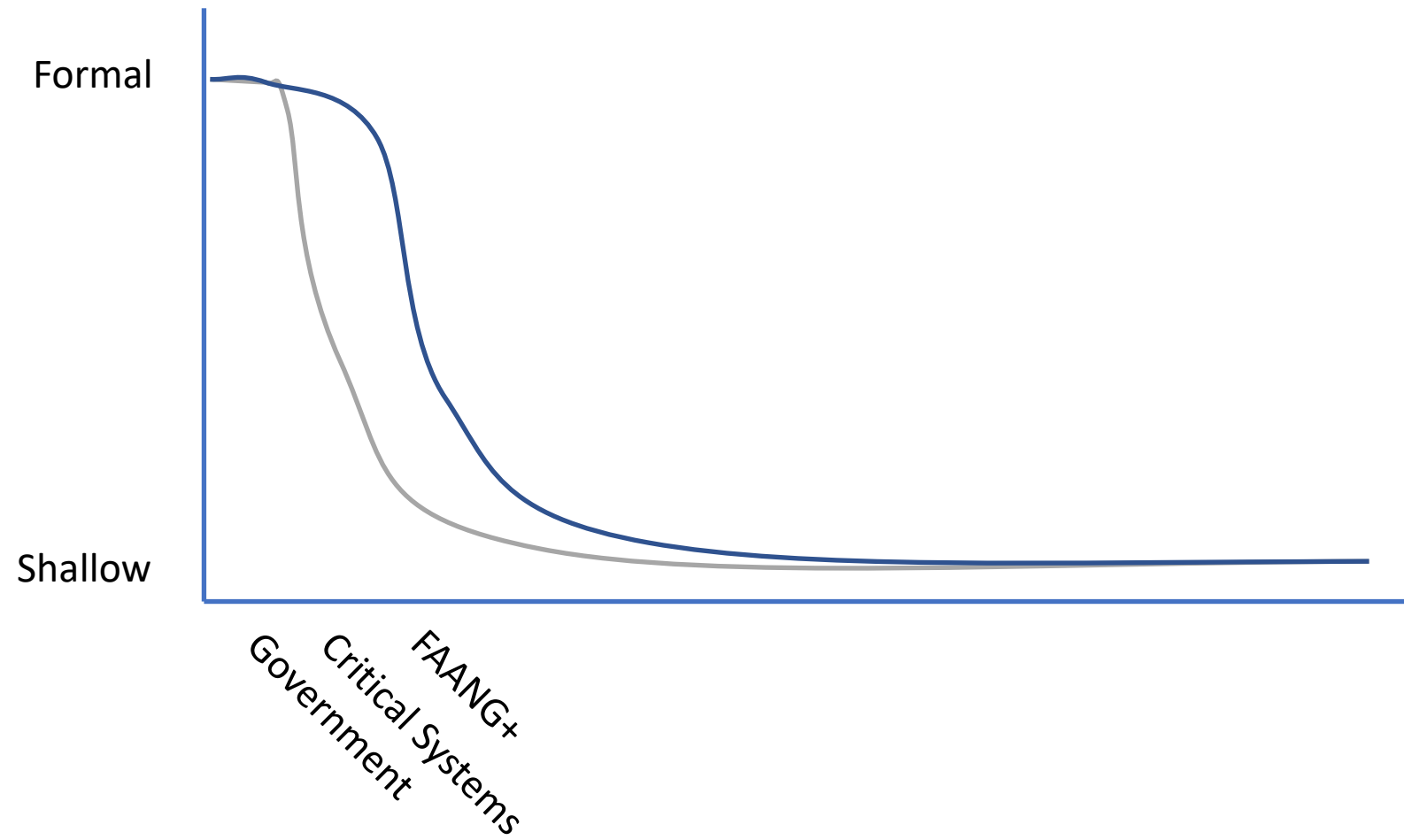
Positive Signs

Success at scale at Facebook, Google, Amazon, Uber

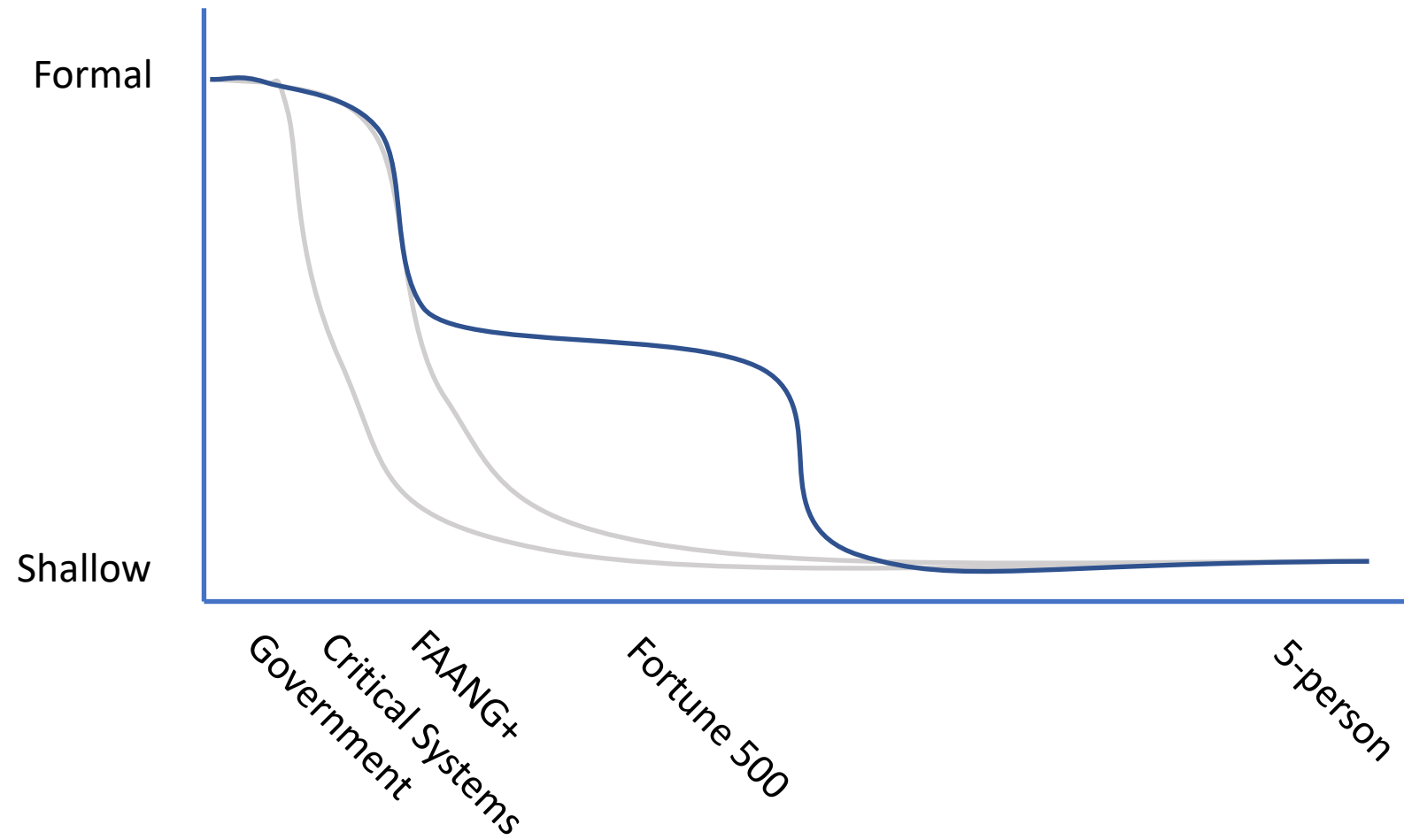
Goals



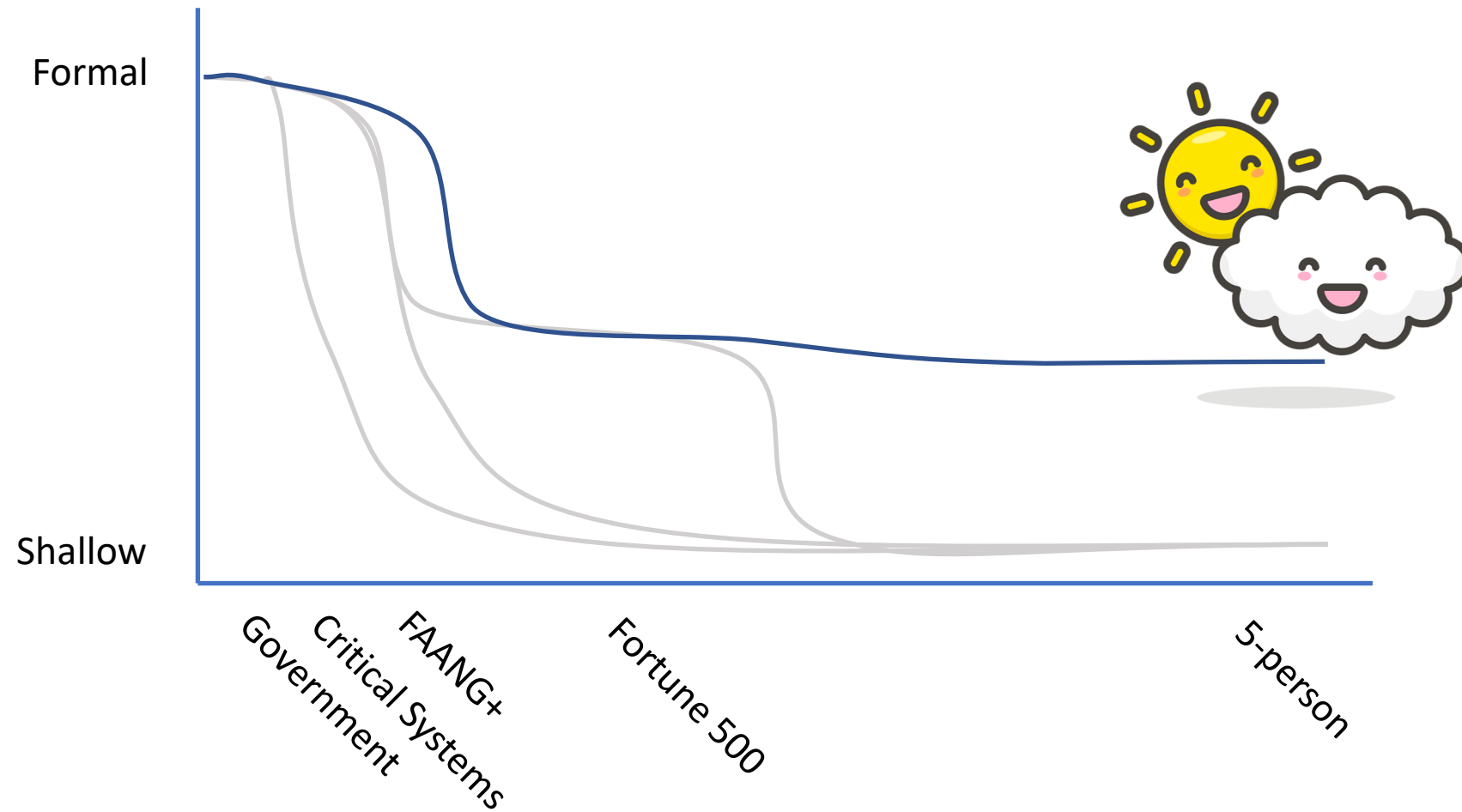
Goals



Goals



Goals



This Talk

- What can a code analysis platform do to
 - Help analysis technology transition
 - Help improve program analysis
- What challenges remain?

MUSEDEV

- Focused on bringing effective program analysis to every company that wants to prioritize code quality.
- Help new program analysis capabilities transition.
- A platform for program analysis experimentation at scale.
- Help foster tighter relationship between academia, government, and industry.

MUSE (PRODUCT)

- A suite of advanced code analysis tools integrated into development
- New bugs flagged in code review
- Extensible platform (easy to add new analyzers)
- Data-driven: Bug reports, bug fixes, and developer feedback tracked over time

Refactored MatrixApiActivity and XML #1187

🔗 Open

langsmith wants to merge 2 commits into `master` from `ls-matrixapi-refactor` 📄

💬 Conversation 4

🔗 Commits 2

📄 Checks 0

📄 Files changed 3



langsmith commented on Aug 26, 2019

Contributor



`MatrixApiActivity` was using a bunch of deprecated Maps SDK classes and methods, such as `addMarker()`, `setOnMarkerClickListener()`, etc.

This pr brings this example up-to-date with `SymbolLayer` icons, `onMapClick()` logic, adding null checks, etc. This pr gets rid of the final usage of `addMarker()` 🐛🔨



langsmith added

✓ ready for review 👍

refactor 🔄

labels on Aug 26, 2019



langsmith force-pushed the `ls-matrixapi-refactor` branch from `4be4ff9` to `0441d3a`
on Oct 5, 2019



muse-dev bot reviewed on Oct 5, 2019

[View changes](#)

```
...Demo/src/main/java/com/mapbox/mapboxandroiddemo/examples/javaservices/MatrixApiActivity.java
```

Outdated

```
135 +     if (pointOfSelectedStation != null) {
136 +         String selectedBoltFeatureName = renderedStationFeatures.get(
137 +         List<Feature> featureList = featureCollection.features();
138 +         for (int i = 0; i < featureList.size(); i++) {
```



muse-dev bot on Oct 5, 2019



NULL_DEREFERENCE: object `featureList` last assigned on line 137 could be null and is dereferenced at line 138.



Reply...



New changes since you last viewed

[View changes](#)



Application

Muse-Dev

ⓘ You've already granted this app access to GitHub outside of GitHub Marketplace.

[Set up a plan](#)

[⚙️ Configure access](#)

? Not verified by GitHub

Categories

[Code quality](#)

[Code review](#)

[Free](#)

Supported languages

C, C++, Java
and [2 other languages supported](#)

Continuous Assurance, Delivered.

[Muse](#) helps you find and fix your most elusive bugs so you can spend time writing great code, not debugging it. Muse looks for a broad range of performance, security, and reliability errors, making it an ideal all-in-one bug catcher for your entire company. Running at each pull request, Muse delivers results as code review comments so you can fix bugs in minutes. (Yes really)

[Read more...](#)

Integration Challenge #1: Feedback

- Get feedback on
 - Presence of bugs in existing code bases
 - Usefulness of bug reports to developers
- Aggregate this information
- Use this to adjust configurations & improve tools

Feedback Loop (Testing)

```
In [415]: dataset['jobid'].apply(lambda j: show_bug_types(static_check_results(j)))
```

```
Out[415]: 0          {SA4006, S1005, S1002, S1028}
1          {}
2          {}
3          {S1005}
4          {S1007, ST1005, U1000, SA4006}
5          {S1008, SA5001}
6          {compile, ST1005, S1003, S1023, SA4006}
7          {S1019, ST1005, S1004, S1023, S1034}
8          {}
9          {S1008, ST1005, S1034, S1004}
10         {S1008, ST1005, S1001, U1000, SA1016, ST1006}
11         {SA1029, S1009, U1000, SA6002, S1010, S1034, S...
12         {S1023, compile, S1021}
13         {compile}
14         {}
15         {U1000, SA4011, S1006, S1003, S1028}
```


```
dataset[bug_types].sum()
```


simplifications	224
concurrency	0
testing	2
useless code	47
correctness	8
performance	4
dubious constructs	6
style issues	225
dtype:	int64

Feedback Loop (Production)

```
...droidDemo/src/global/java/com/mapbox/mapboxandroidide
mo/MainActivity.java Outdated
```


```
... @@ -202,6 +206,8 @@
202 206 private boolean loggedIn;
203 207 private int currentCategory = R.id.nav_basics;
204 208 private boolean showJavaExamples = true;
209 + private long clickTimeOfLastSelectedExample = 0;
```


 **muse-dev** (bot) on Oct 4, 2019
UnusedVariable: The field 'clickTimeOfLastSelectedExample' is never read.

 **langsmith** on Oct 4, 2019 Author Contributor
I've removed it.


```
src/acvp_transport.c Outdated
```

```
1131 - int max_url = ACVP_ATTR_URL_MAX + 1;
1132 - int rem_space = max_url - 1;
1131 + int max_url = ACVP_ATTR_URL_MAX;
1132 + int rem_space = max_url;
```

 **muse-dev** (bot) on Mar 23
DEAD_STORE: The value written to &rem_space (type int) is never used.

 Fixed bot issue 51a997e

address a couple of the muse errors #254

 **Merged** bfussell merged 4 commits into [cisco:master](#) from [bfussell:master](#)  on Apr 12, 2019

Remaining Challenge: Feedback for non-open-source-code

- Major IP concerns
 - Detailed bug results could leak proprietary information
 - Even existence of certain bug types could be sensitive
- Connectivity issues
 - Is there even a way to communicate information back?
- Utility concerns
 - Is failure data useful when separated from the code?
- May remain manual / custom

Integration Challenge #2: Environments

I thought:

“Docker solves everything!”

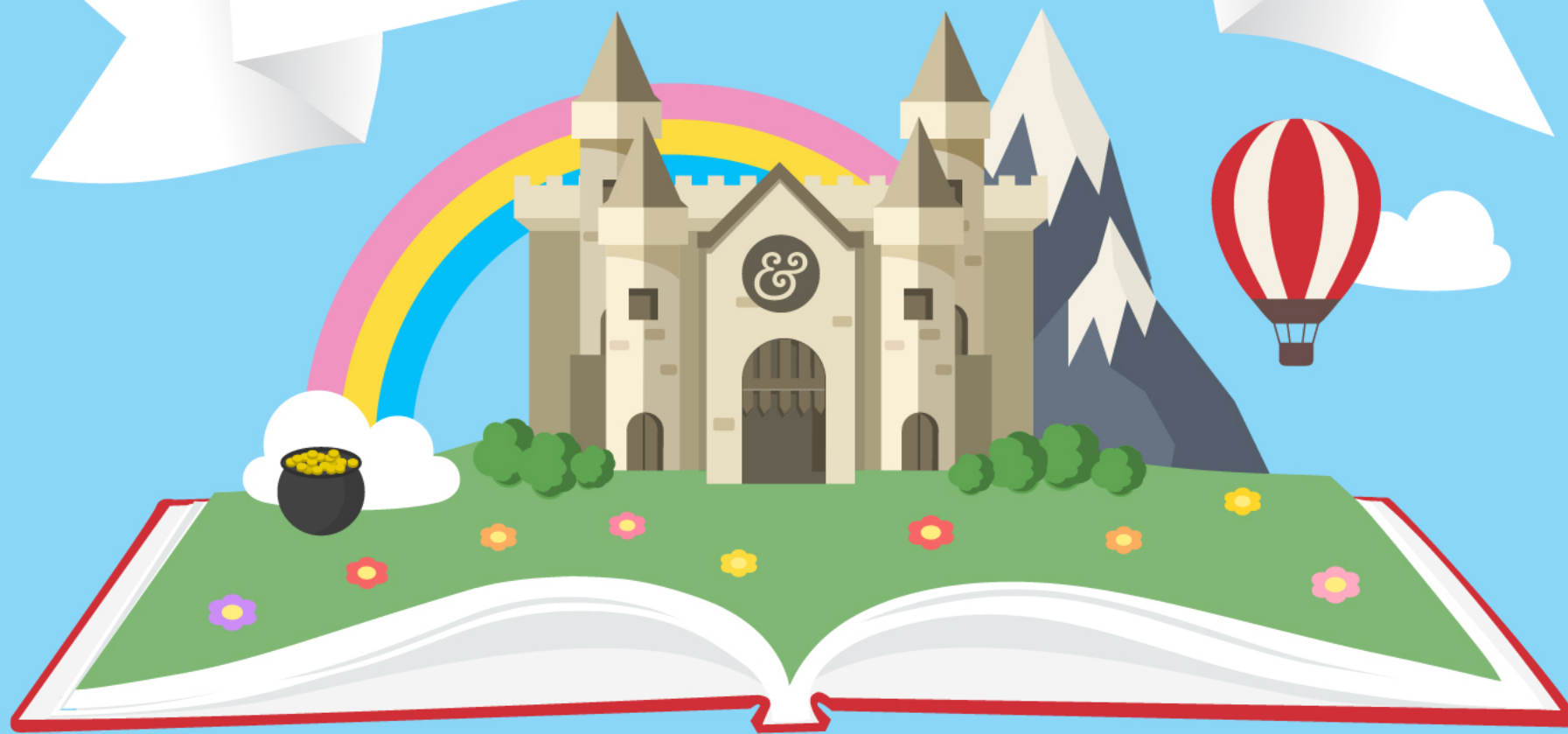
Integration Challenge #2: Environments

I thought:

“Docker solves everything!”



Story Time



Solutions

- Configurability
 - Deployment options (containers or native)
 - Build information
 - Artifact servers
- Autodetection
 - Keep common cases simple
- Simplified tool interface
 - Pull dependencies so tools don't have to
 - Compilation databases so build tooling is off the critical path

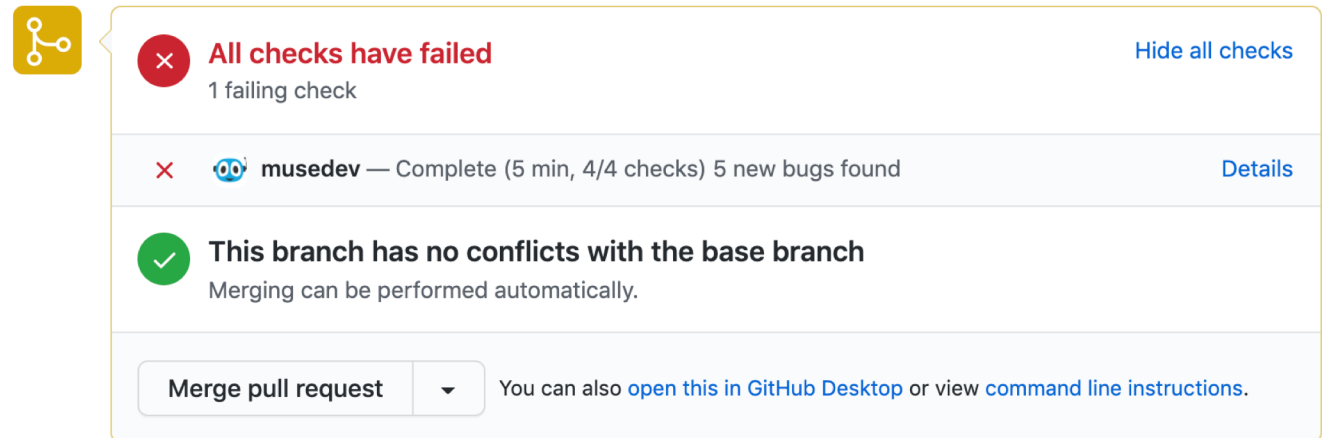
Open Question:

Just how smooth can we make this?

- Right balance of supportability and breadth of coverage
- Seamless update experience
- Supportability

Integration Challenge #3: Minimizing Developer Friction

- Should errors block the build? Turn the “badge” red?
- What about errors in generated code? Test code?
- Need configurability
- Nondeterministic analyses?
- Tracking bug identity over time
- **Also: Actionable bug reports**



The screenshot shows a GitHub pull request status bar. It features a yellow icon of a branching diagram on the left. The main content is a list of checks:

- A red circle with a white 'x' icon, followed by the text "All checks have failed" in red, and "1 failing check" below it. A "Hide all checks" link is on the right.
- A red circle with a white 'x' icon, followed by the GitHub logo, the username "musedev", and the text "Complete (5 min, 4/4 checks) 5 new bugs found". A "Details" link is on the right.
- A green circle with a white checkmark icon, followed by the text "This branch has no conflicts with the base branch" and "Merging can be performed automatically." below it.

At the bottom, there is a button labeled "Merge pull request" with a dropdown arrow, and a note: "You can also [open this in GitHub Desktop](#) or view [command line instructions](#)."

Challenges

1. Providing Feedback
2. Complex Environments
3. Minimizing Friction

These Challenges are Addressable

- We can provide tool agnostic support for
 - Multiple tools
 - Aggregation of results
 - Statistics and reporting
 - Providing a common UI
 - Common configuration
- These are all high value for both users and tool authors

NEXT STEPS



We are looking for feedback in three key areas:

USERS |

Want to experiment with Muse? Reach out!

FEATURE REQUESTS |

What would you most like to see?

INTROS |

Who should we talk to?

THANK YOU

Stephen Magill | CEO, MuseDev

@stephenmagill
stephen@muse.dev

MuseDev

<https://muse.dev/>