Large scale deployment of libTooling derived tools

Bloomberg

Engineering

2023 US LLVM Developers' Meeting
October 11, 2023

Vaibhav Yenamandra & Konstantin Romanov
Software Engineers
Static Analysis and Automated Refactoring team

TechAtBloomberg.com

Hello LLVM!

- We are Vaibhav Yenamandra and Konstantin Romanov
- Work in the Static Analysis and Automated Refactoring team
- Part of Developer Experience (DevX) @ Bloomberg

TechAtBloomberg.com

Bloomberg

Engineering

What is the plan for today?

- Motivation
  - Why static analysis pipelines?
  - Source code at Bloomberg
  - What kinds of tools do we want?
- Example workflows
  - What was needed
  - Our solution
  - Challenges
- Questions

TechAtBloomberg.com

Bloomberg

Engineering

Source Code at Bloomberg

- Bloomberg has hundreds of millions of lines of C, C++ source code
- Our C++ code changes constantly
  - **15K+** PRs merged weekly
  - **4K+** source packages published weekly
  - **1.7M+** packages built weekly (includes dependency rebuilds)
- Most of our modern code lives in Git
  - Leverages one or more CI/CD solutions
  - CMake
  - Makefile
- Code is built together in a consistent “distribution”
- Statically-linked targets where it matters
- Internal libraries and middleware

TechAtBloomberg.com

Bloomberg

Engineering

Why Static Analysis Pipelines

- Need a structured way to introduce tooling
- Failed PRs are better than runtime errors
- Centrally understand “the codebase”

What kinds of tools do we use?

- Emit diagnostics
  - `clang++` , `g++ -Werror` for compiler upgrades
  - `clang-tidy`
- Emit patches
  - `clang-tidy`
  - Automatic feature toggle retirement
  - Automatic `#include` fixer
- Offline analysis
  - Capturing call-graphs & dependencies between functions
  - Infer RW-ness of pointer params
  - Static application security analysis

Bloomberg

Engineering

Example: Gating Package Releases

TechAtBloomberg.com

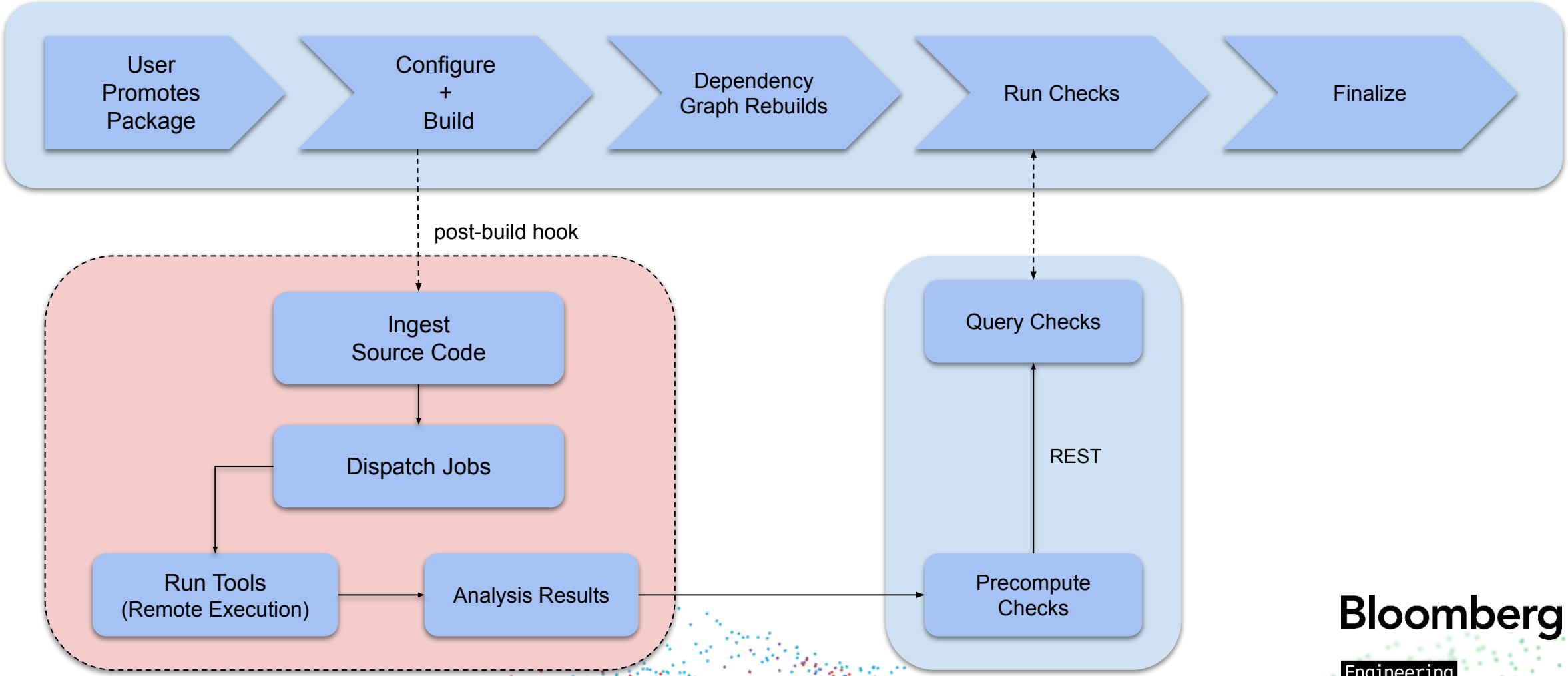
Bloomberg

Engineering

Gating Package Releases

- Code is built together in a consistent “distribution”
- All published packages are checked
  - Contents inspected
  - Check for potential regressions
- Releases gated on check outcome
- Need real-time, parallel checks
  - Missing deadlines holds up package release
  - Might have custom compute for checks
  - Publish pipeline needs to be light

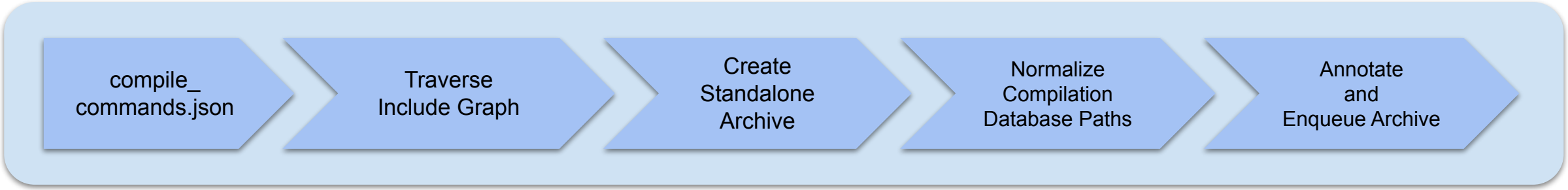
Package Release Workflow



Bloomberg

Engineering

Ingesting Source Code

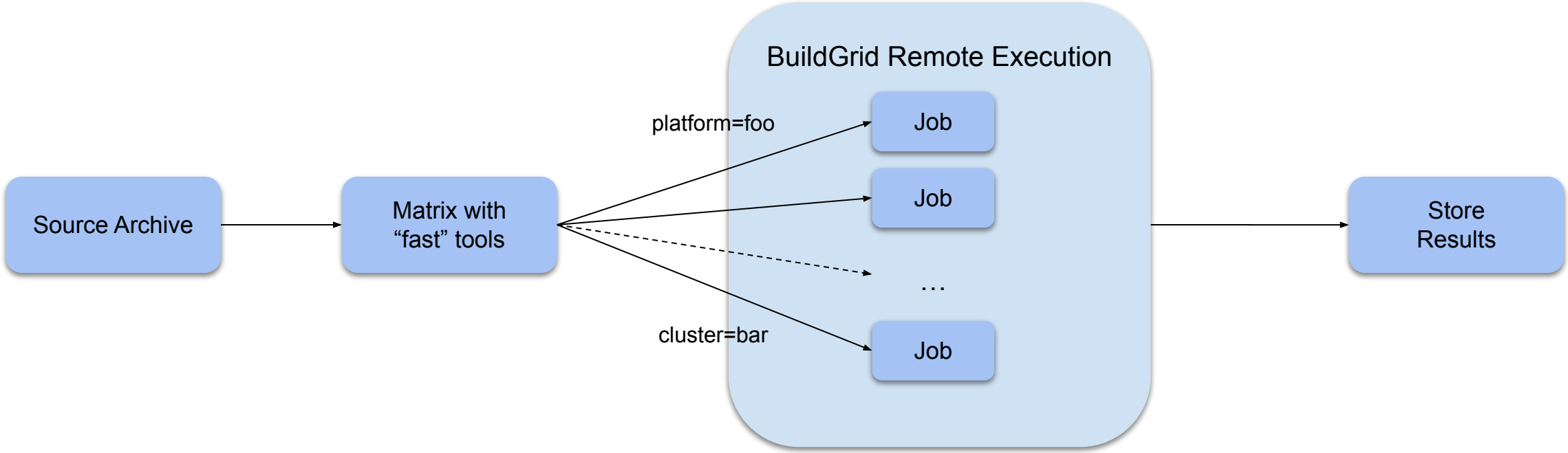


TechAtBloomberg.com

Bloomberg

Engineering

Dispatch and Analysis



TechAtBloomberg.com

Bloomberg

Engineering

Challenges

- Parsing effort is duplicated
- Regression metrics: Diagnostics are hard to trace
- Consistency of distributed source code and diagnostics

Example: Feature Toggle Retirement

TechAtBloomberg.com

Bloomberg

Engineering

Feature Toggle Retirement

- Code, feature sets advance rapidly
- Desirable to control new features
  - Granular on-switches
- Centrally managed
  - Each switch must be registered with a end-date
  - Scope of deployment set on registration
  - Can be retired early by developers
- Feature switches are perfect for automatic removal
  - Values are runtime boolean constants
  - Usually have well known end of life
  - In C++ code each switch has one distinct accessor

Example Code Transformation

Before removal

```
#include <feature_1234.h>

// void process_workflow_foo() {
if (have_feature_1234()) {
   // Code that uses this feature ...
}
```

After removal of feature_1234 == true

```
// void process_workflow_foo() {
{
   // Code that uses this feature ...
}
```

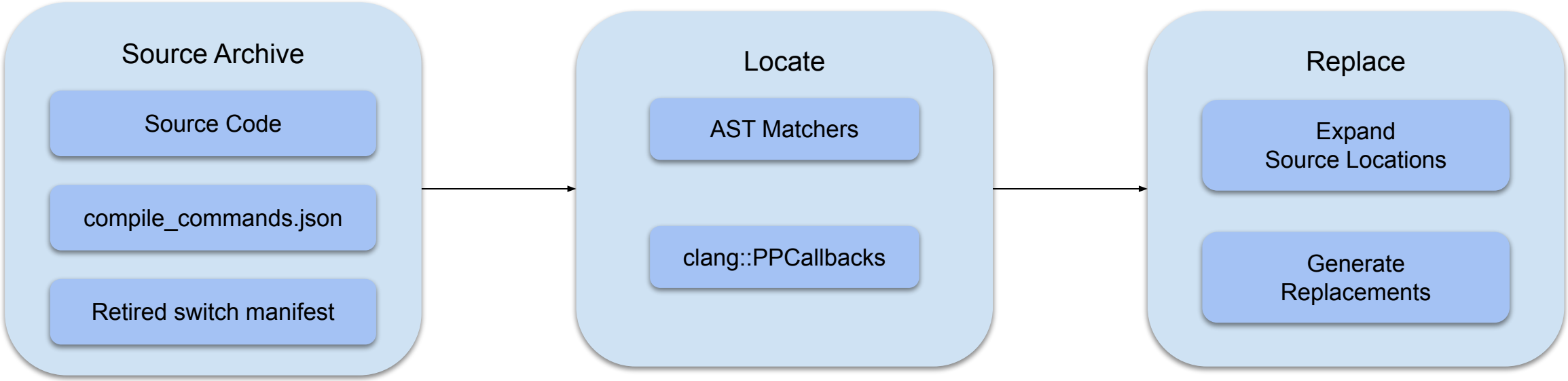
```
#include <feature_1234.h>
#include <feature_5678.h>

// void process_workflow_bar() { ...
if (have_feature_1234() &&
    have_feature_5678()) {
   // Extra work
}
```

```
#include <feature_5678.h>

// void process_workflow_bar() { ...
if (have_feature_5678()) {
   // Extra work
}
```

The Tool



TechAtBloomberg.com

Bloomberg

Engineering

Recursive AST Matchers for Control Structures

```
ignoringParenImpCasts(
  callExpr(callee(isBoolFeatureFn()),
    hasArgument(
      0,
      ignoringParenImpCasts(
        unaryOperator(
          hasOperatorName("&"),
          hasUnaryOperand(
            declRefExpr(to(varDecl().bind("featureVarDecl")))
          )
))));
```

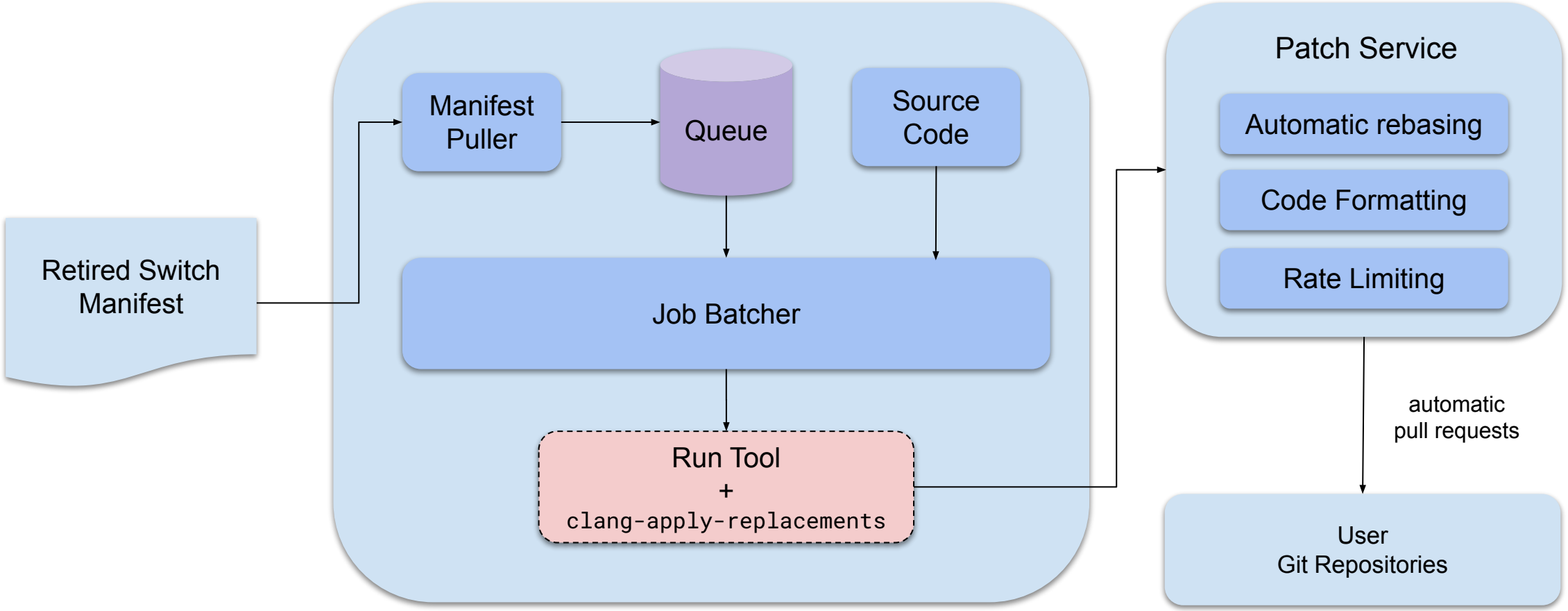
Matches

```
if(have_feature_7())
if(!have_feature_7())

have_feature_7() ? foo() : bar()

have_feature_7() && something
have_feature_7() || something
```

Design – Feature Toggle Retirement



TechAtBloomberg.com

Challenges: Replacing Test Code

```
#include <test/framework.h>
#include <feature_1234_override.h>

TEST_CASE(Suite, TestFeature1234Works) {
   // Check positive case
   override_feature_1234(true)          // Default is true => this line is
safe to remove.
   run_some_code();

   // Check negative case
   override_feature_1234(false);
   run_some_other_code();               // Do we need to remove this line?
}
```

Challenges: Corner Case 1

```
#include <feature_7.h>

void foo( … ) {
   // Cache expensive calls
   // before tight-loop code
   bool feat_7 = have_feature_7();

   for (const auto &item: sequence) {
      if (feat_7) {
          // Optional extra processing
      }
      // Regular work
   }
}
```

Challenges: Corner Case 2

```
#include <feature_7.h>

void foo(int x) {
  // Chained ifs
  if(x > 30) {
     // Regular work
  } else if(!have_feature_7()) {
     // Deprecated processing
  }
}
```

Challenges: Corner Case 3

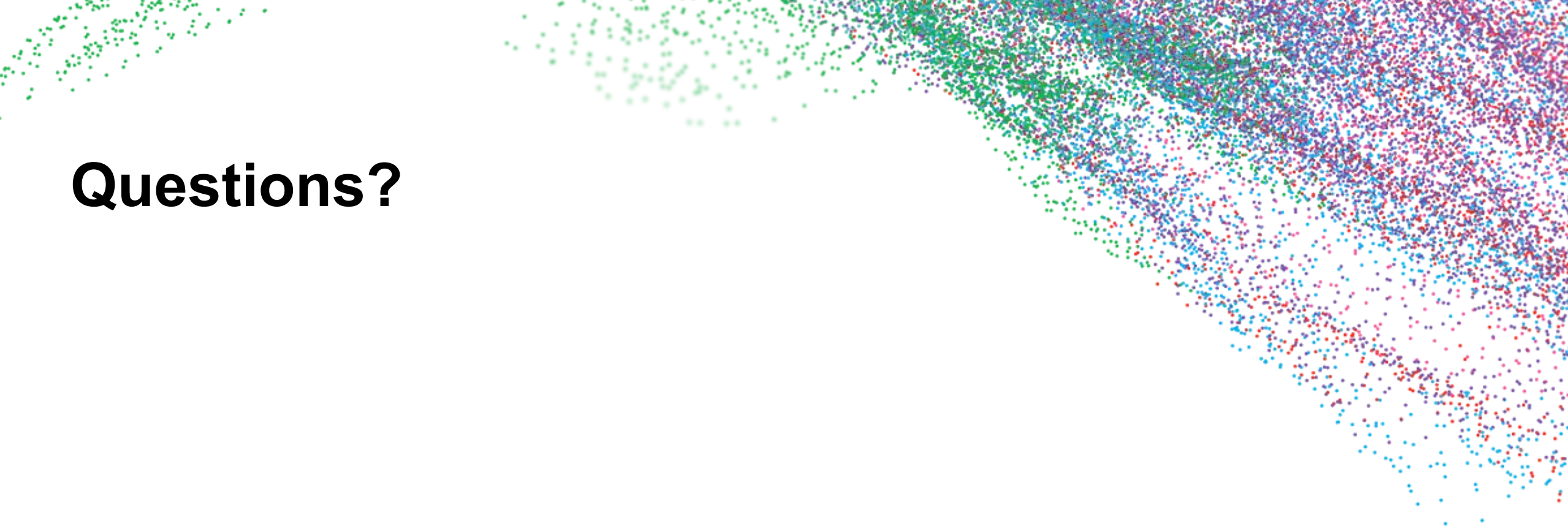
```
#include <feature_7.h>

void foo() {
#if defined(__linux) || defined(__gcc)
   bool feat_7 = ...;
#else
   bool feat_7 = have_feature_7();
#endif
}
```

Closing Thoughts / Considerations

- Wishlist for `clang::`
  - Matchers for macros!
  - Easier to see preprocessor's view of source code
  - Access to constant propagation / folding
  - Multi-pass analysis that saturates
- Wishlist for `clang-apply-replacements`
  - Stabilization of the input interface
  - Would like to use it for languages other than C/C++
  - Better handling of conflicting insertions
    - Merge strategies?
    - No abnormal terminations

Questions?



TechAtBloomberg.com

Bloomberg

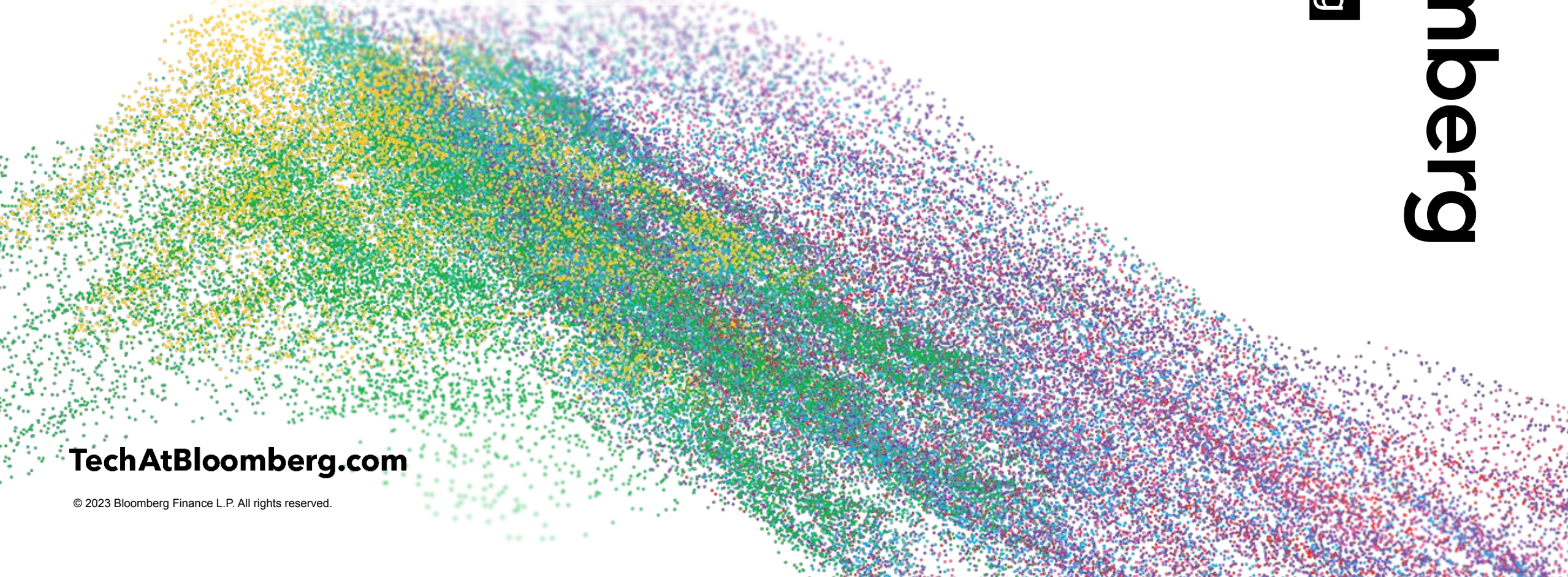
Engineering

Thank you!

We are hiring: bloomberg.com/engineering

Bloomberg

Engineering



TechAtBloomberg.com