Fast, Not Furious Developer Experience

Gautam Korlam

Senior Staff Engineer

Fast, not Furious Developer Experience

Gautam Korlam
Java Platform Tech Lead

Uber

@kageiit

Agenda

01 Uber’s Scale
02 Fast, Modern Tools and Frameworks
03 Safety nets
04 Standardization
05 Results and Summary


Uber’s Scale

Development at Uber Scale

Thousands
Microservices

Hundreds of Thousands
Deployments per month

Millions
Config changes per month

Tens
Mobile apps

4K+
Engineers

7
Programming languages

Tens of Thousands
commits per month

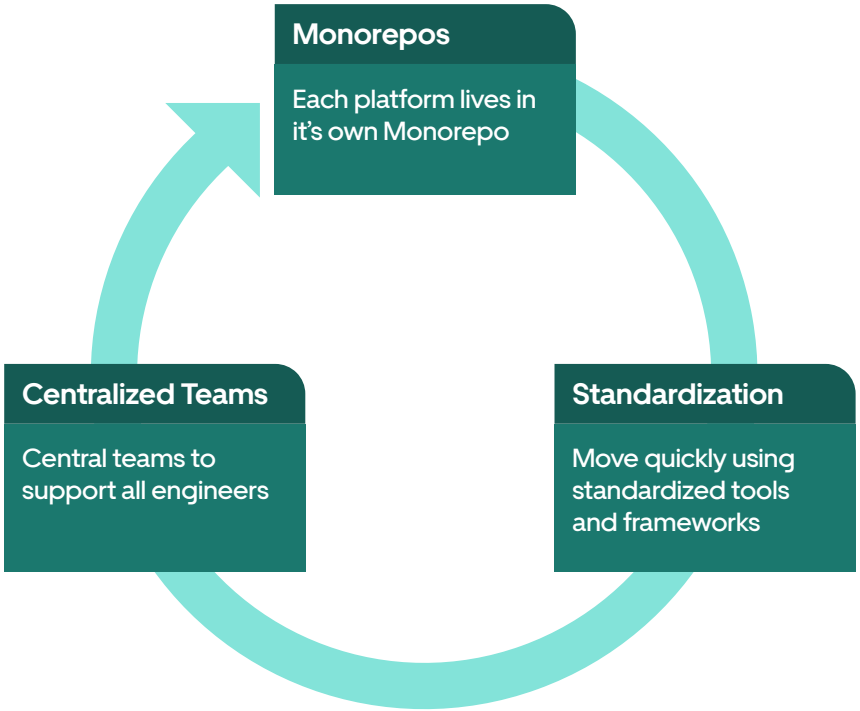
Hundreds of Millions
Monorepo LOC

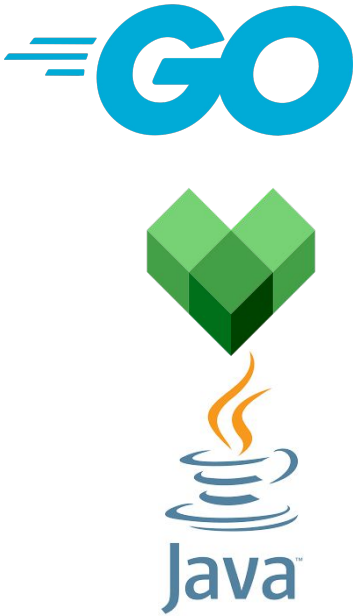
5
Platform Monorepos

Trillions
Mobile analytics events per month

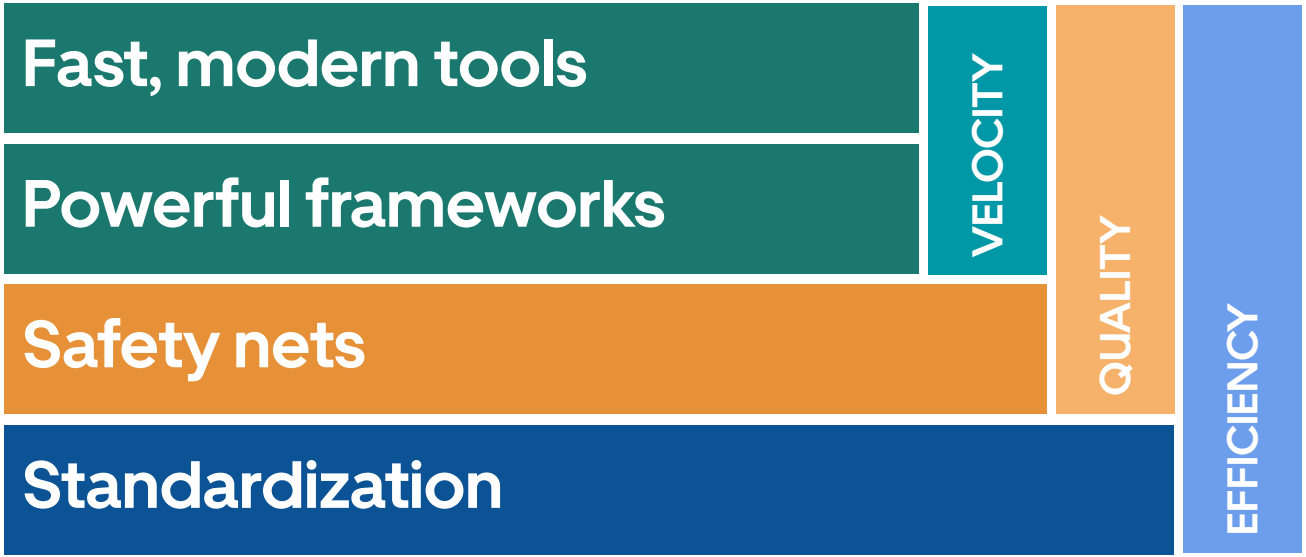
Billions
User sessions per month

Development Stack





Strategy

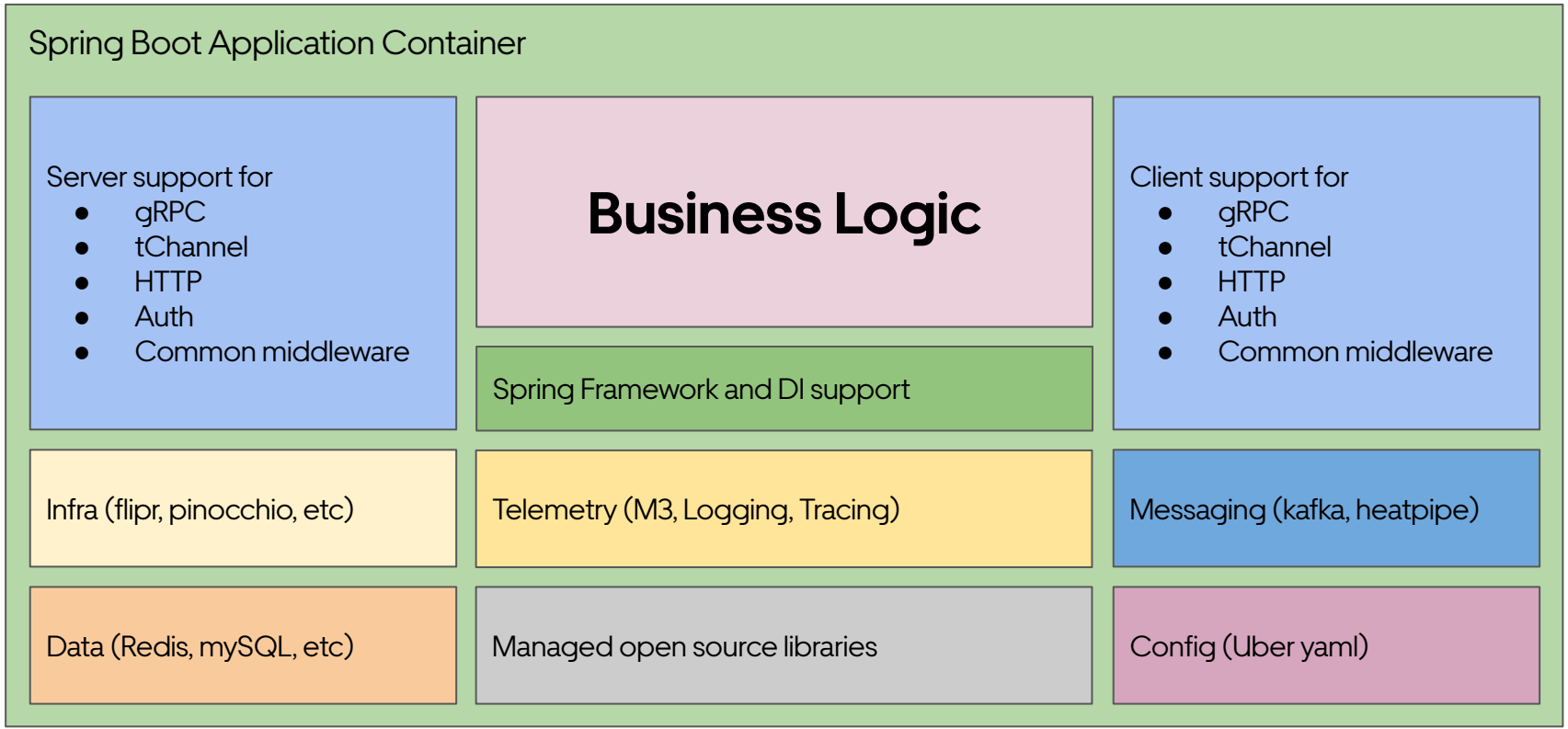

Fast, modern tools
Powerful frameworks
Safety nets
Standardization
VELOCITY
QUALITY
EFFICIENCY


Fast, Modern Tools
and Frameworks

Modern Tools

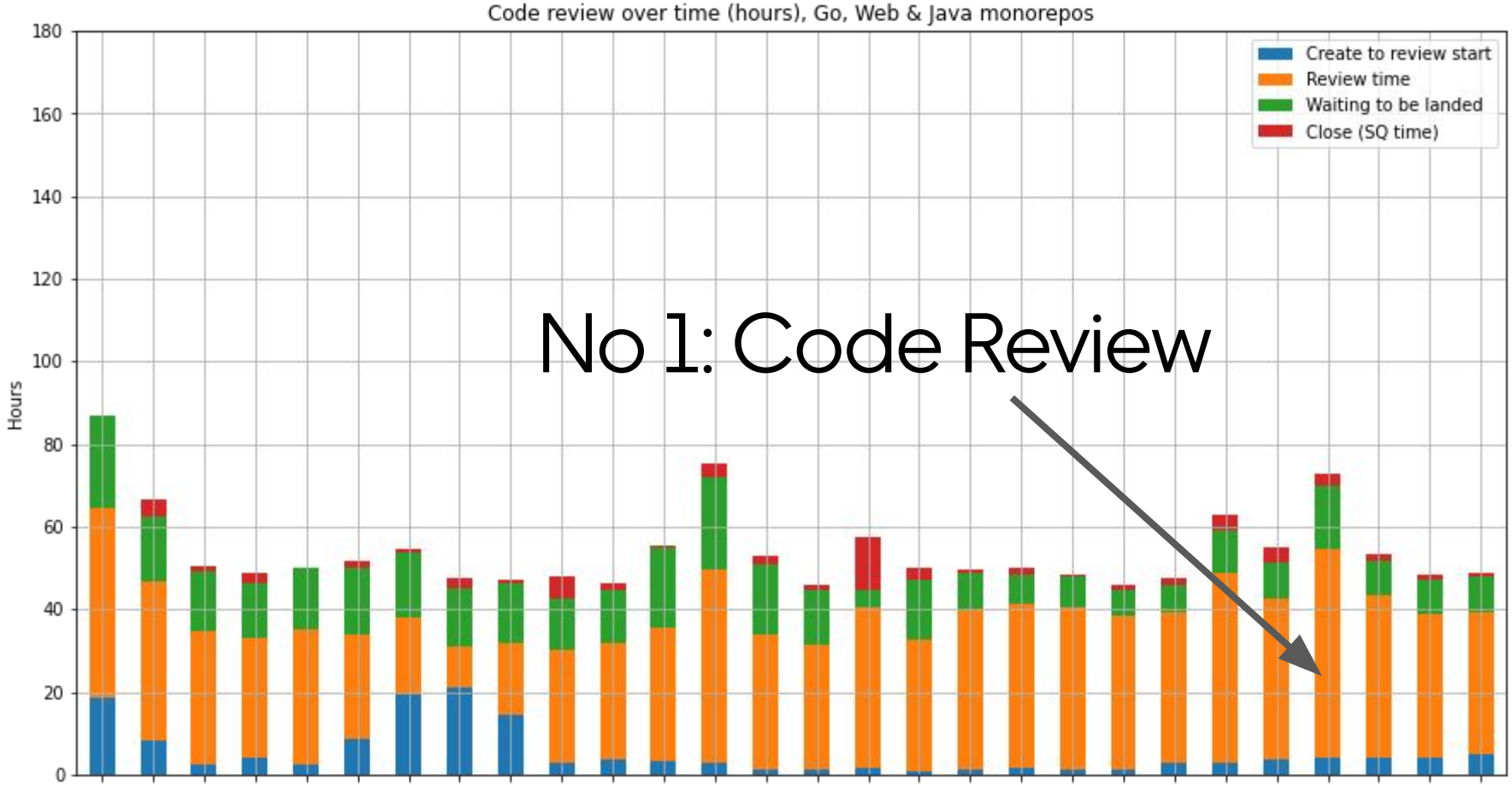
Laptops	Devpods + M1
Phabricator	Github
Jenkins	Buildkite
Intellij 2021	Intellij 2022
Buck	Bazel
Java 8	Java 11 → Java 17

JFx - Powerful Java Framework




Need for Speed


What slows down the rate of commits?

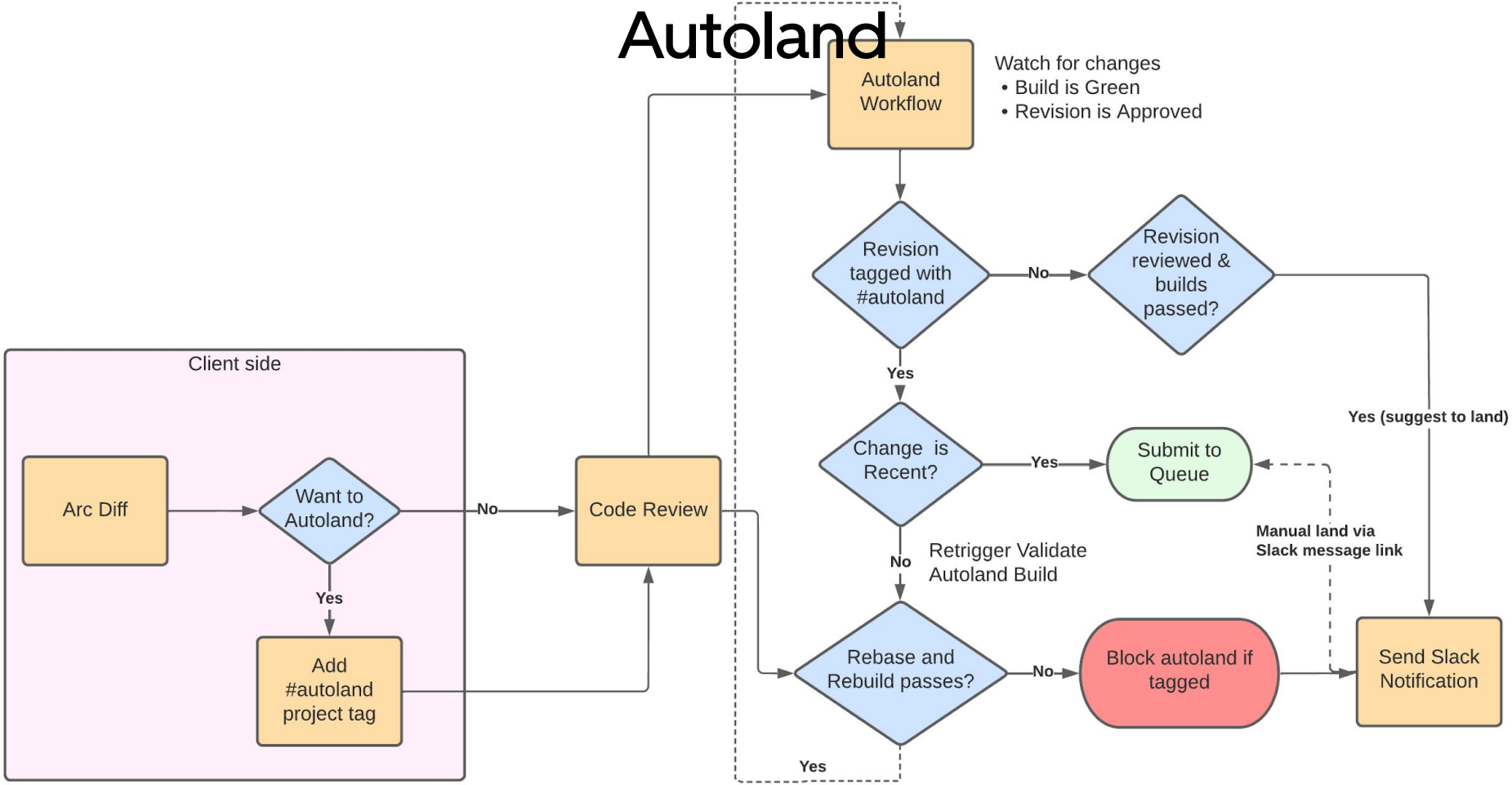

Code review over time (hours), Go, Web & Java monorepos
Hours
180
160
140
120
100
80
60
40
20
0
Create to review start
Review time
Waiting to be landed
Close (SQ time)
No 1: Code Review


Code review over time (hours), Go, Web & Java monorepos
Hours
180
160
140
120
100
80
60
40
20
0
Create to review start
Review time
Waiting to be landed
Close (SQ time)
No 2: Wait before Merge

When Changes Wait

- More likely to conflict
- Build caches out of date
- Overhead in Merge Queues
- Context switches for developers

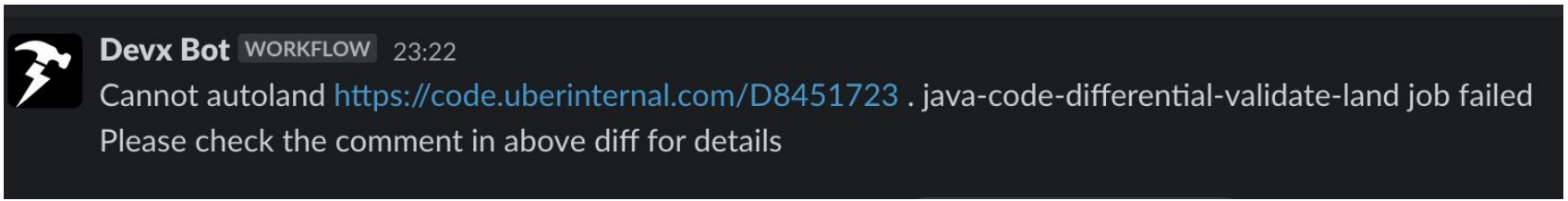
Autoland

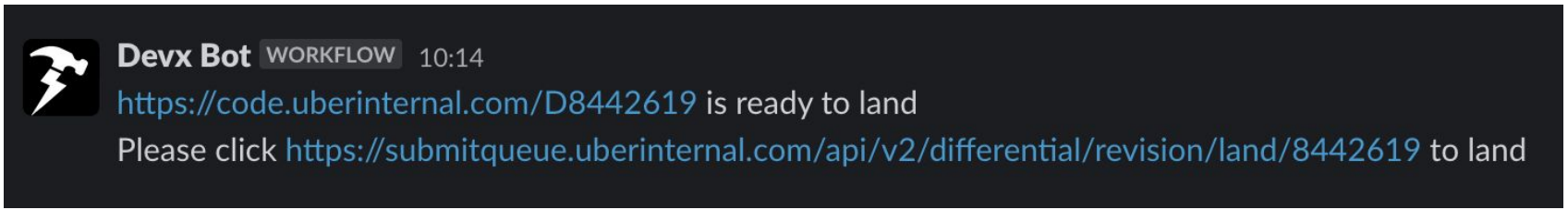

Autoland Workflow
Watch for changes
• Build is Green
• Revision is Approved
Revision tagged with #autoland
No
Revision reviewed & builds passed?
Yes
Yes (suggest to land)
Client side
Arc Diff
Want to Autoland?
No
Code Review
Change is Recent?
Yes
Submit to Queue
Yes
Manual land via Slack message link
No
Retrigger Validate Autoland Build
Rebase and Rebuild passes?
No
Block autoland if tagged
Send Slack Notification
Add #autoland project tag
Yes

Autoland UX

```
❯ arc diff --create

    Autoland after builds pass and reviewers approve (Default Y)? [Y/n]
```





Autoland Impact

Merge Queue Time Improved

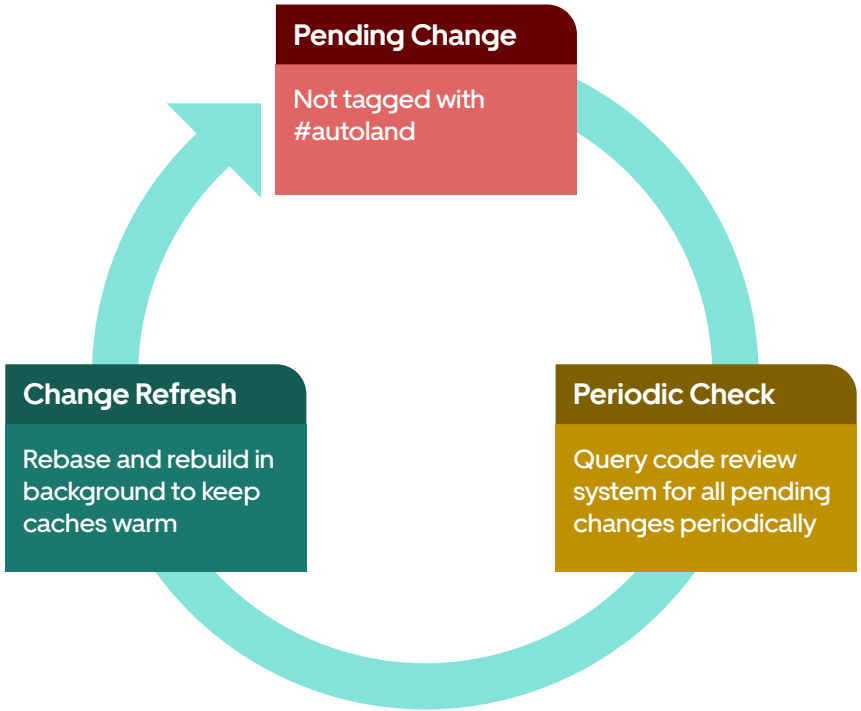
Merge Conflicts (3% → 1%)

~ 60% of changes are Autolanded

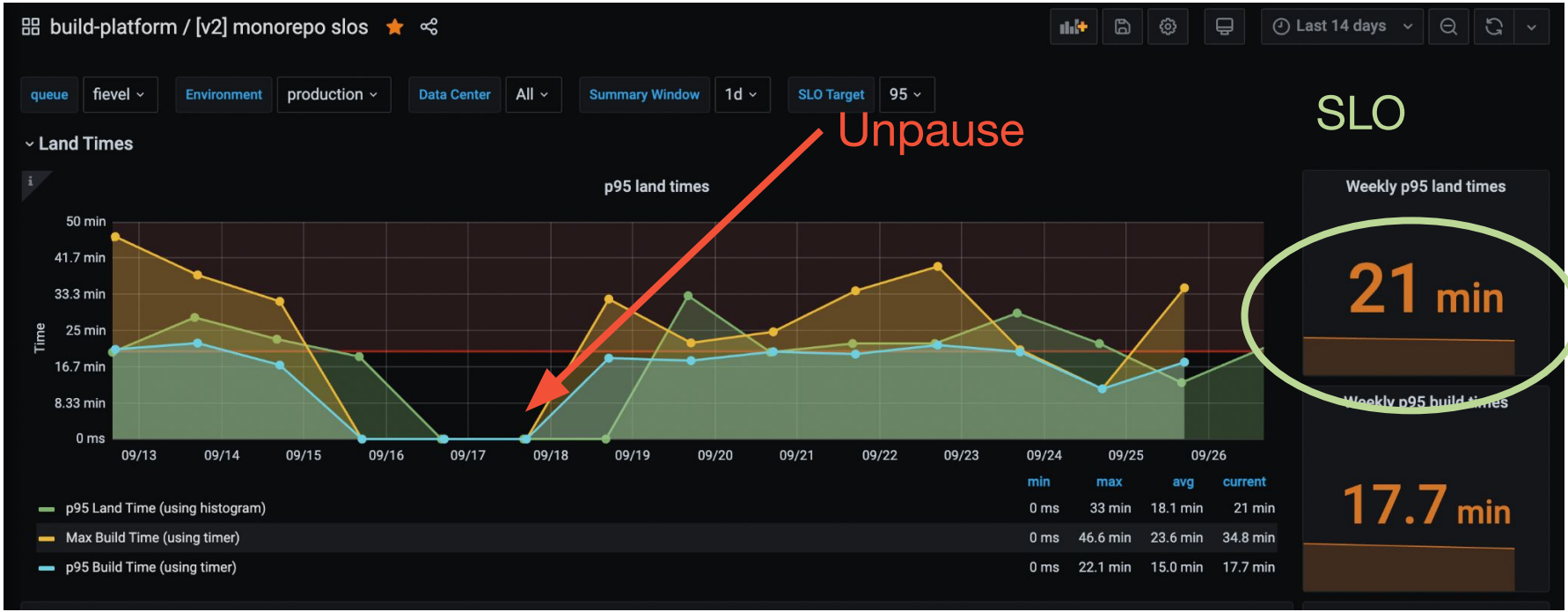
Not All Changes Autoland

- Even after approvals, some changes take time to merge
- Wait for more reviewers, manual testing etc.
- Cause spikes in p95 build time metrics

Diff Refresh



Diff Refresh Impact



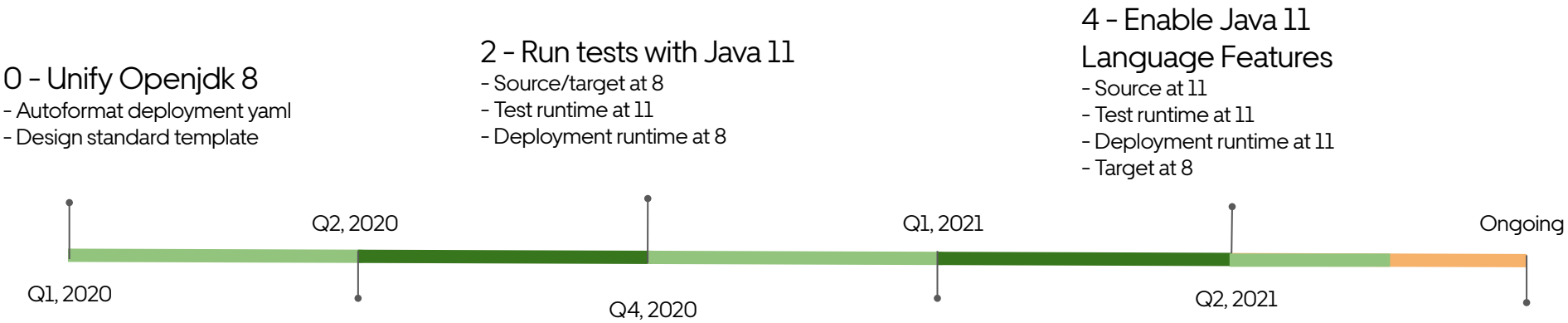
Platform Freshness

Platform Freshness

Platform	Done	Planned
	Android 12 Kotlin 1.7.10 IntelliJ 2022.2	Kotlin 1.7.20, 1.8.x+ Intellij 22.3, 23.1 Android 13
	iOS 16, Swift 5.7, Xcode 14, Bazel 5.2	Swift 6, RxSwift 6.5, Bazel 6 Xcode & iOS minor updates (14.1, 14.2, 16.1, etc)
	Go 1.19 Goland 2022.2 Bazel 5.3.1	Go 1.20 Bazel 6.0.0 Goland 22.3, 23.1
	Java 11 IntelliJ 2022.2	Spring Boot 2.7.2 Java 17 Runtime Intellij 22.3, 23.1
	Node 16 React 18	Node 18, Bazel 6, Jest 28, Apollo 3


Cost of
Migrations

Phases of Java 8 to 11 Migration




Migrations at Scale

GodWit - Automating Code Migrations and Cleanups

- 5 monorepos spanning 5 languages
- Multiple code cleanup workflows
- Multi-language projects

Polyglot Piranha

Language agnostic structural replace with deep cleanup

- Splitting the migration into multiple diffs
- Systematically tracking the migrations via JIRA

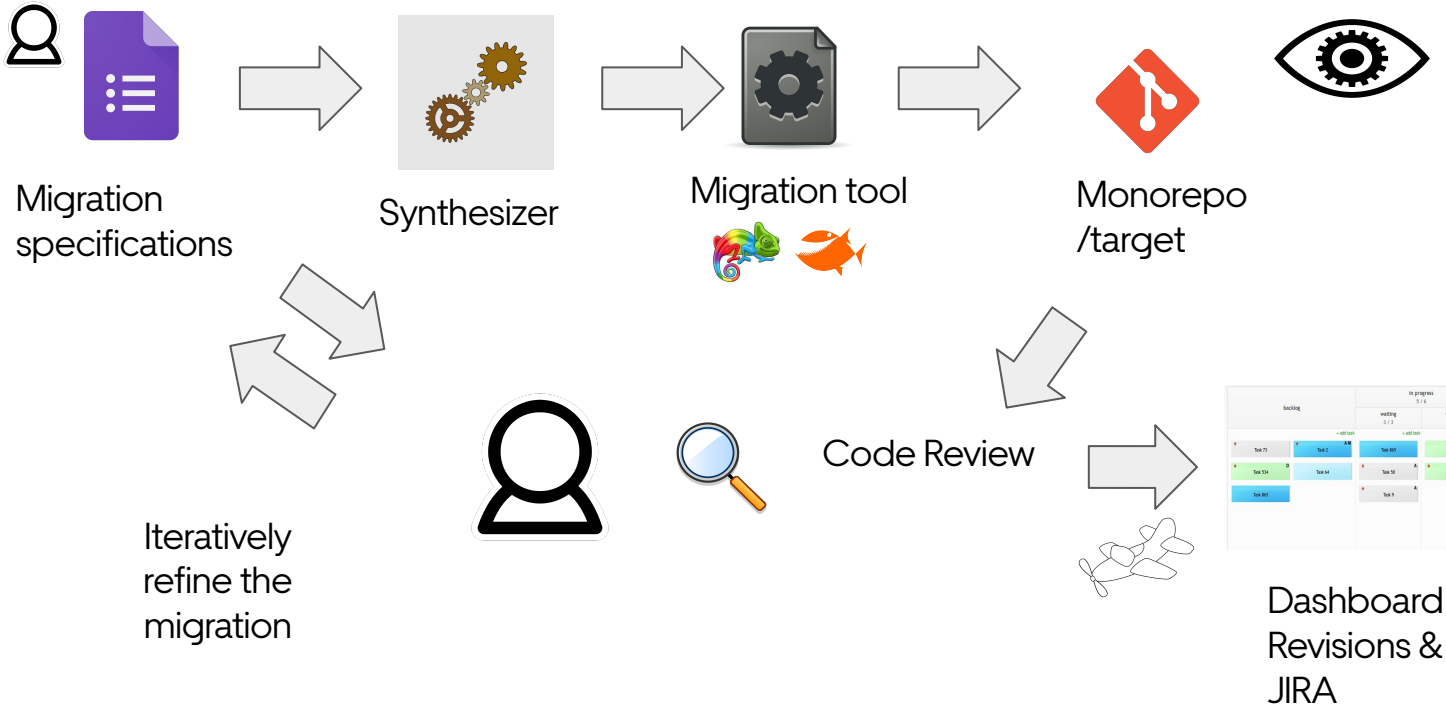
Kameleon

Automated diff and JIRA tracker

Assistant for developing code migration tools

https://github.com/uber/piranha/tree/master/polyglot/piranha

GodWit overview



Use Cases

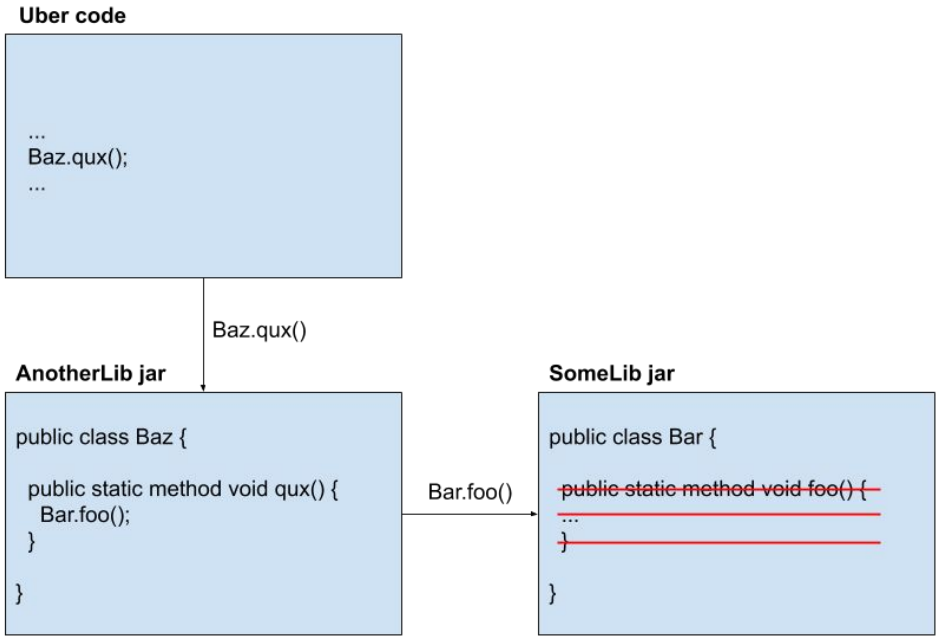
- Stale feature flag cleanup in mobile
- Stale string cleanup in iOS
- Kotlin Symbol Processing (KSP) Migration
- Powermockito to Mockito Migration

Safety Nets

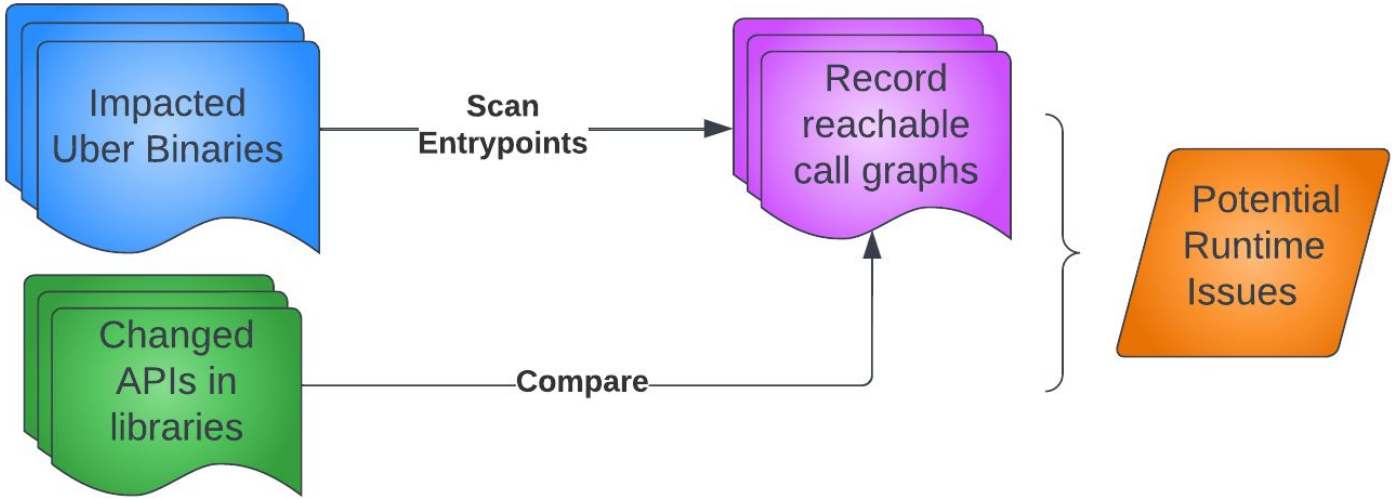
Dependency Management

Dependency Upgrade Issues

- Transitive dependency incompatibility
- Typically detected at runtime
- Hard to debug



Runtime Compatibility Checker



Build has FAILED

Found the following runtime issues:

```
MISSING CALL PATHS FOUND FOR //geosearch/geosearch-modeling/geosearch-modeling-misspellings:bin_main_unshaded:
        io/grpc/netty/shaded/io/netty/util/internal/logging/Log4J2Logger.isEnabled()
```

Please see http://t.uber.com/dep-validator for more info


Standardization

Cost of not doing

Cognitive overhead, hard to move around

Reduce cost to learn

Migrate in place - how we got good at it

Buck to Bazel Example

Seamless Switching

Buck to Bazel Example 1

Seamless Switching

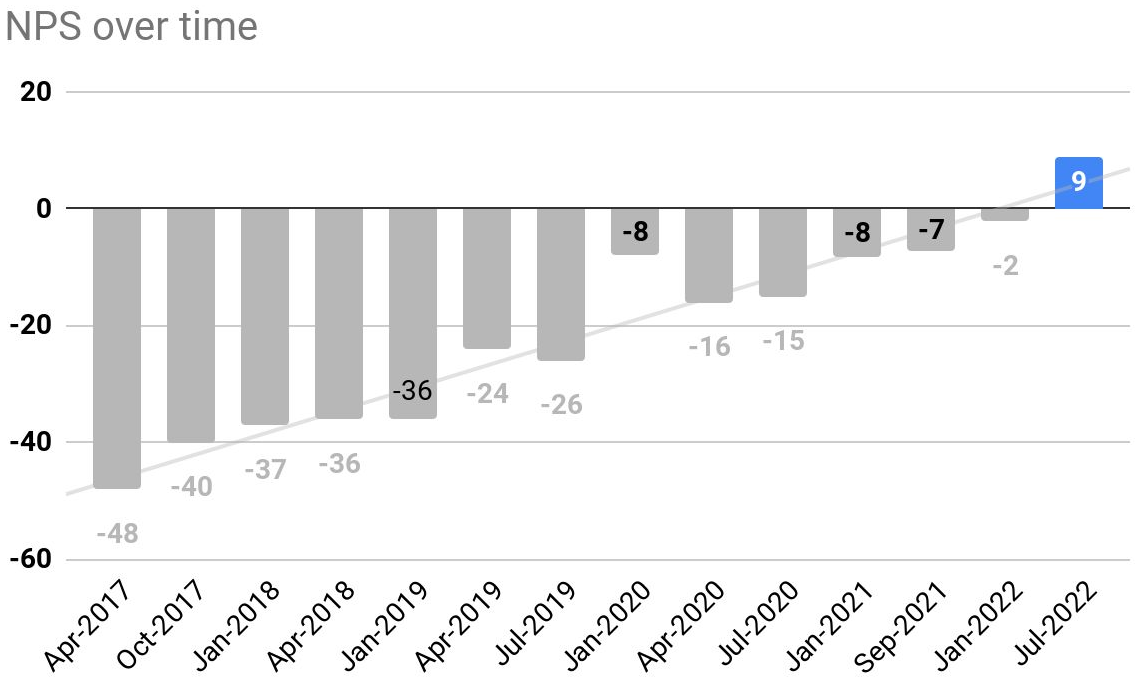
Buck to Bazel Example 2

Seamless Switching

Buck to Bazel Example 3

Seamless Switching

Results - Developer Satisfaction (NPS)



Summary

- Speed alone is not sufficient for great developer experience
- Invest in tooling to reduce developer tax to keep platforms updated at scale.
- Focus time is very valuable. Minimize learning curves, context switches and overhead for developers.

Fast, not Furious Developer Experience

Gautam Korlam
Java Platform Tech Lead

Uber

@kageiit