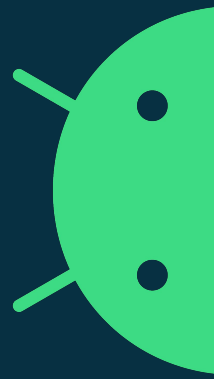
Android Engineering Productivity

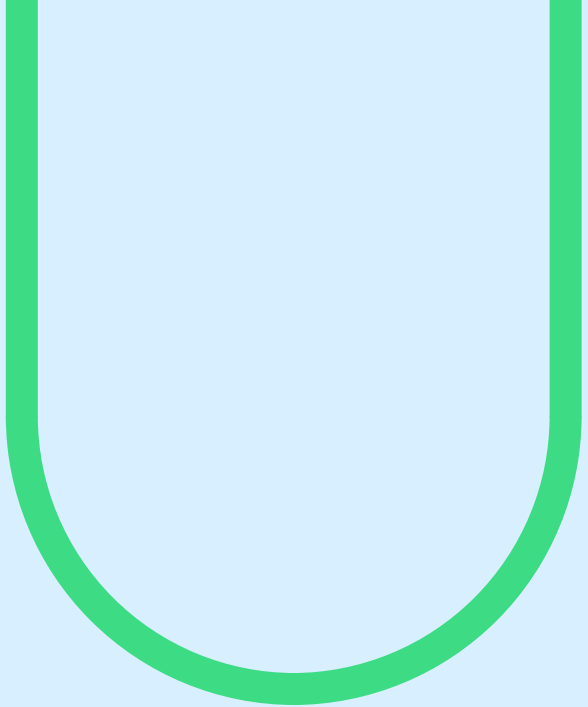


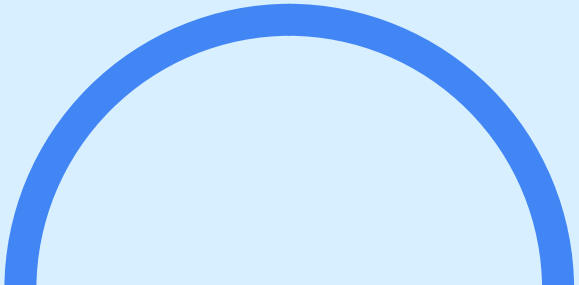
Android Continuous Integration Infrastructure
tsu@

Agenda

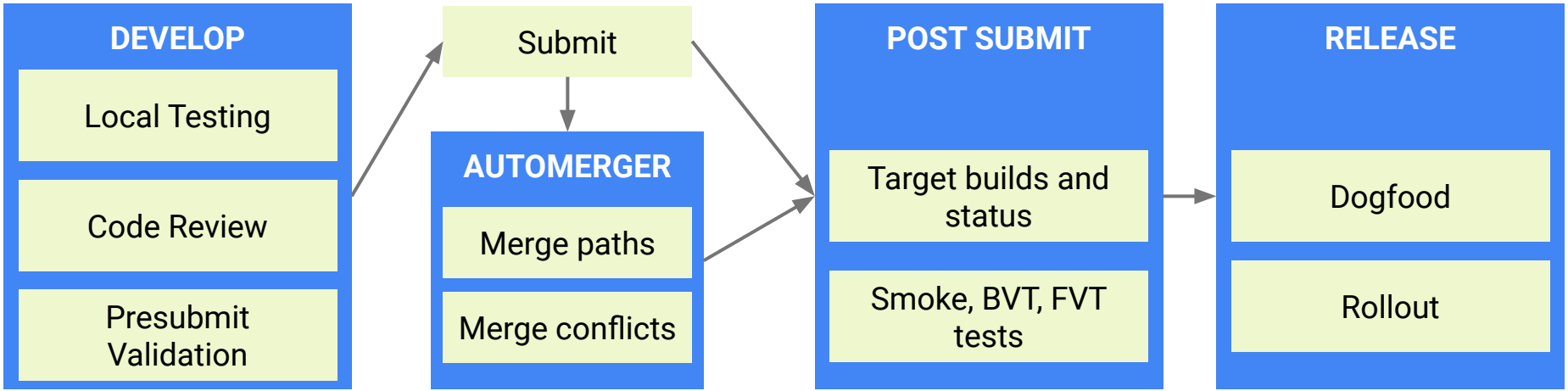
- **Lifecycle of a Changelist**
  - Develop
  - Submit
  - Post Submit
  - Release
- **Infrastructure**
  - Overview
  - Build
  - Test
- **Q & A**

Lifecycle of a CL



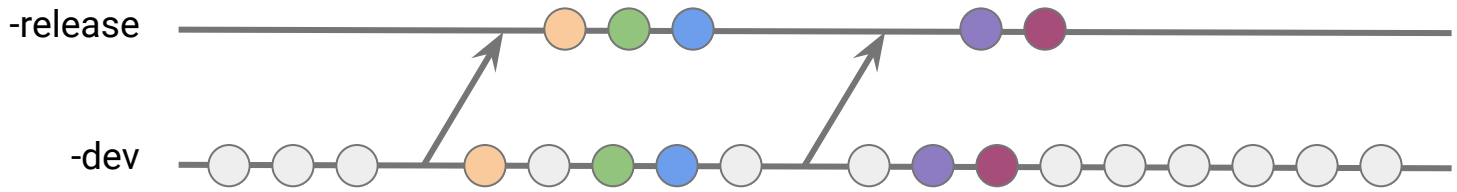


android



android

Platform branch strategy



- Typically two branches, -dev and -release, for example, qt-dev and qt-release.
- All CLs are committed to the dev branch.
- Release branch is regularly snapped from the dev branch.
- Release branch is stabilized by cherry-picking from dev branch.

Development tools

- Git is designed to handle large projects that are distributed over multiple repositories. Android uses Git for local operations such as local branching, commits, diffs, and edits.
- Repo unifies Git repositories when necessary, performs uploads to the Gerrit revision control system, and automates parts of the Android development workflow.
- Gerrit is the web interface for Android and Google Git repositories that enables code reviews and TreeHugger presubmit testing.
- Android Build Dashboard - The place to see what builds are available.
- Treehugger - Tooling and distributed services which aims to provide the necessary infrastructure such that all core development branches can always be in a nearly shippable state.

Local Testing

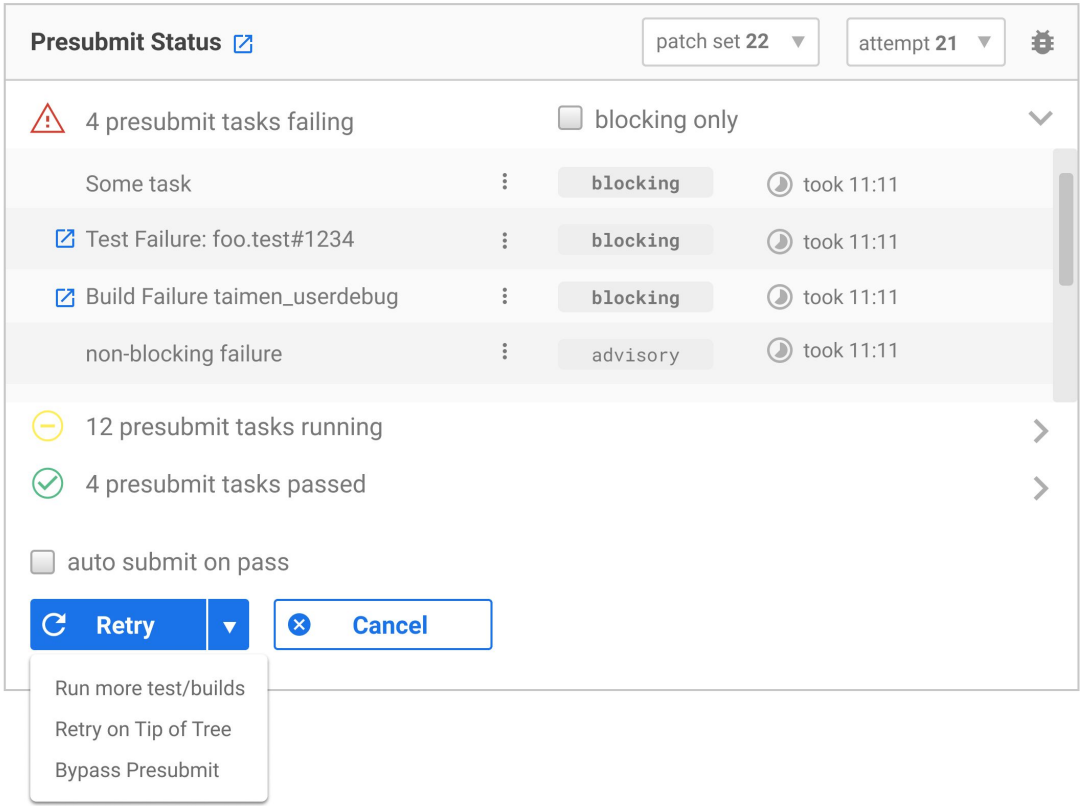
- Manually
  - Push the binary or apk to the device and execute or instrument (pm instrument) it.

- ATest
  - A command line tool that allows users to build, install, and run Android tests locally, greatly speeding test re-runs.
  - Doesn't require knowledge of Trade Federation test harness command line options
  - ATest commands take the following form:

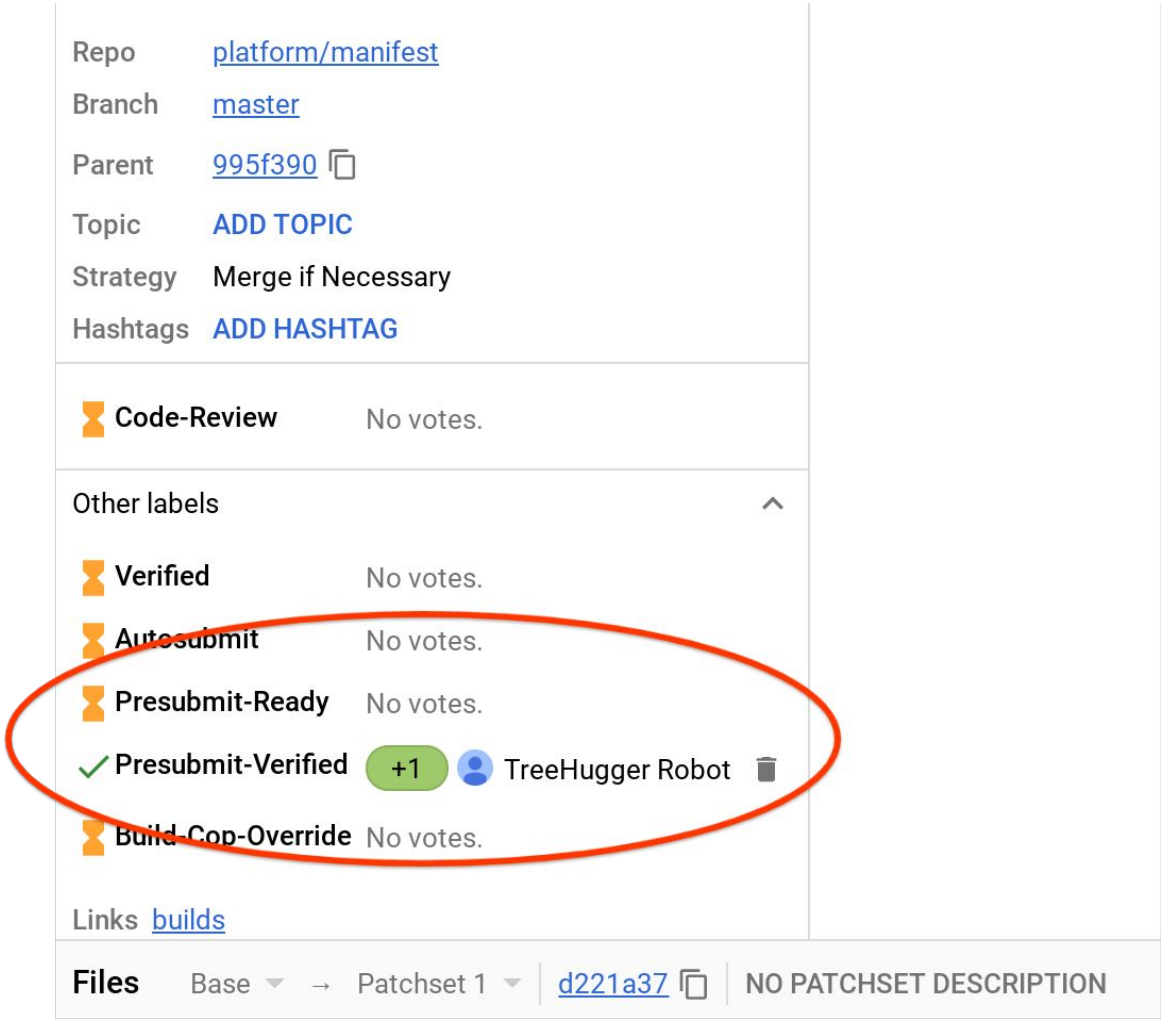
    ```
    atest [optional-arguments] test-to-run
    ```

  - Run `atest -h` (or `--help`) to see all available options

Presubmit Validation



android

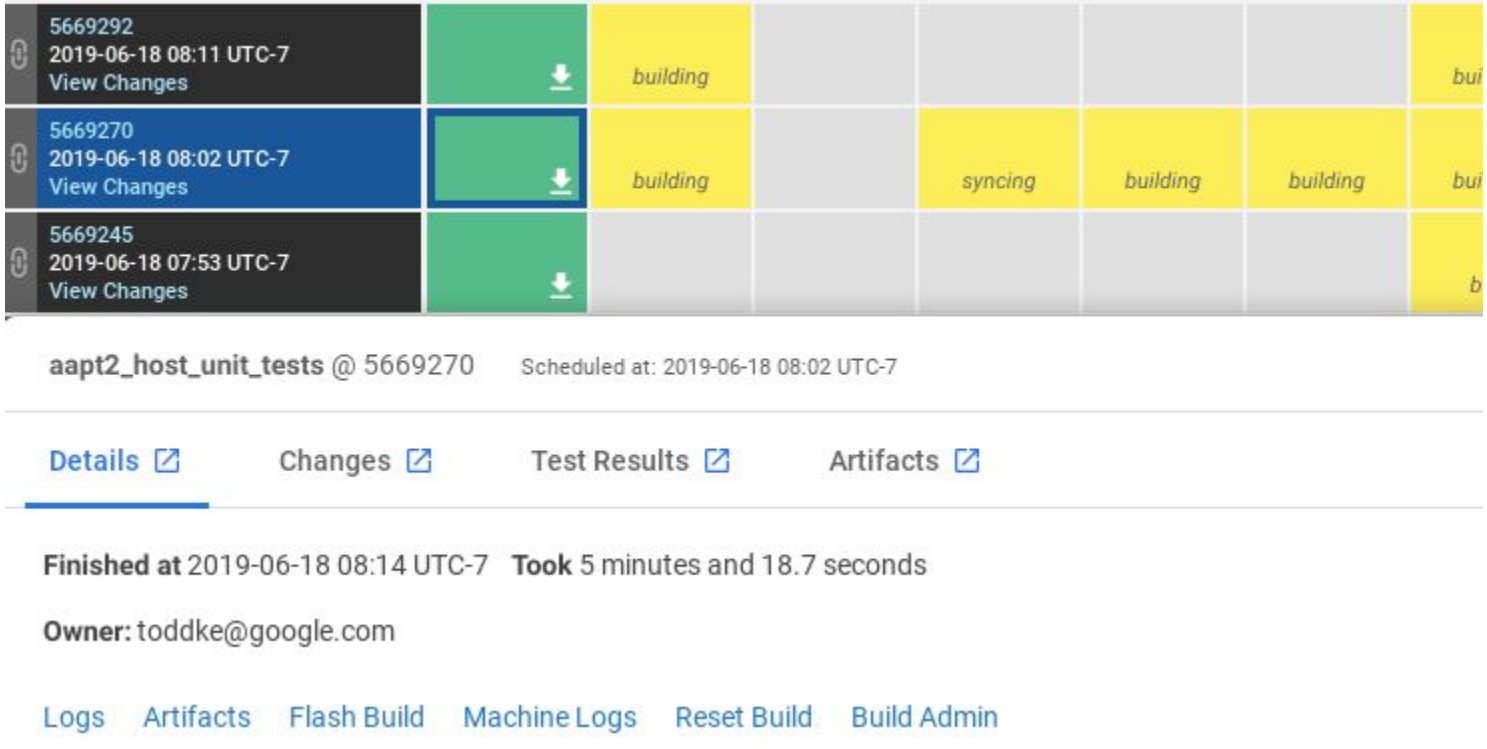


android

Automerger

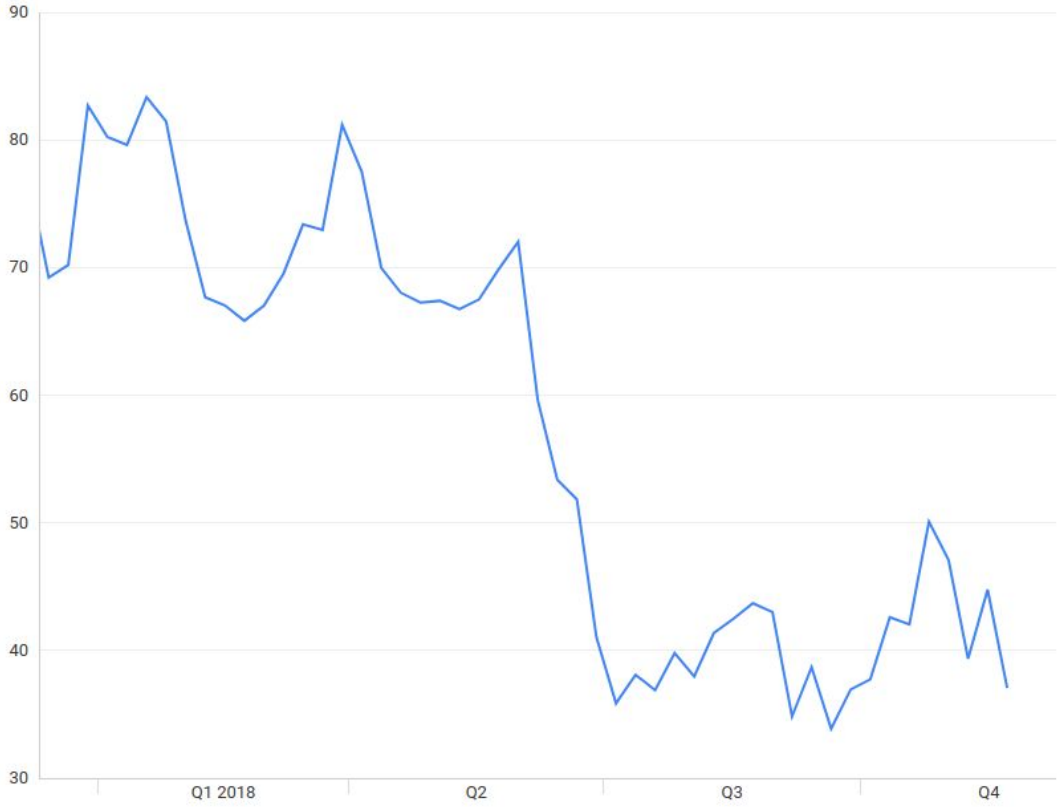
- When a CL is committed, the automerger propagates the patch across branches automatically
- The automerger will email the CL author if there is a merge conflict.
  - The author then resolves the merge conflict
  - Dedicated merge resolution dashboard
- TreeHugger (Presubmit) also checks for merge conflict for major branches

Postsubmit - Integration Builds



android

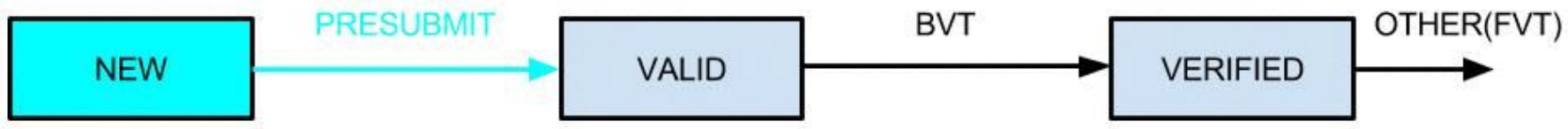
Incremental Builds



- Implementation notes:
  - Over 50% decrease
  - GCE Disk Sharing
  - Smart Scheduling

android

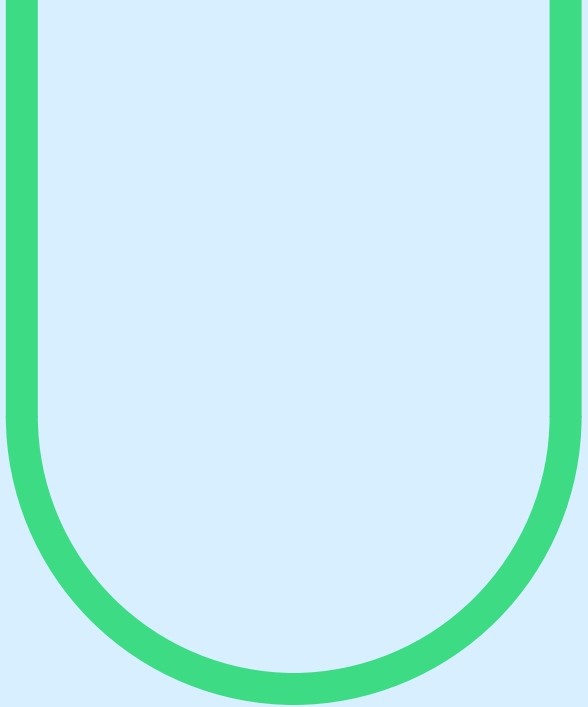
Postsubmit - Running Tests

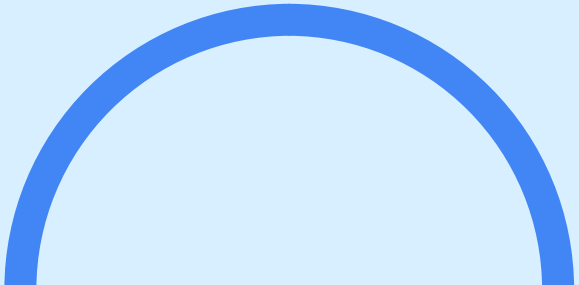


Build is in NEW stage, running PRESUBMIT tests

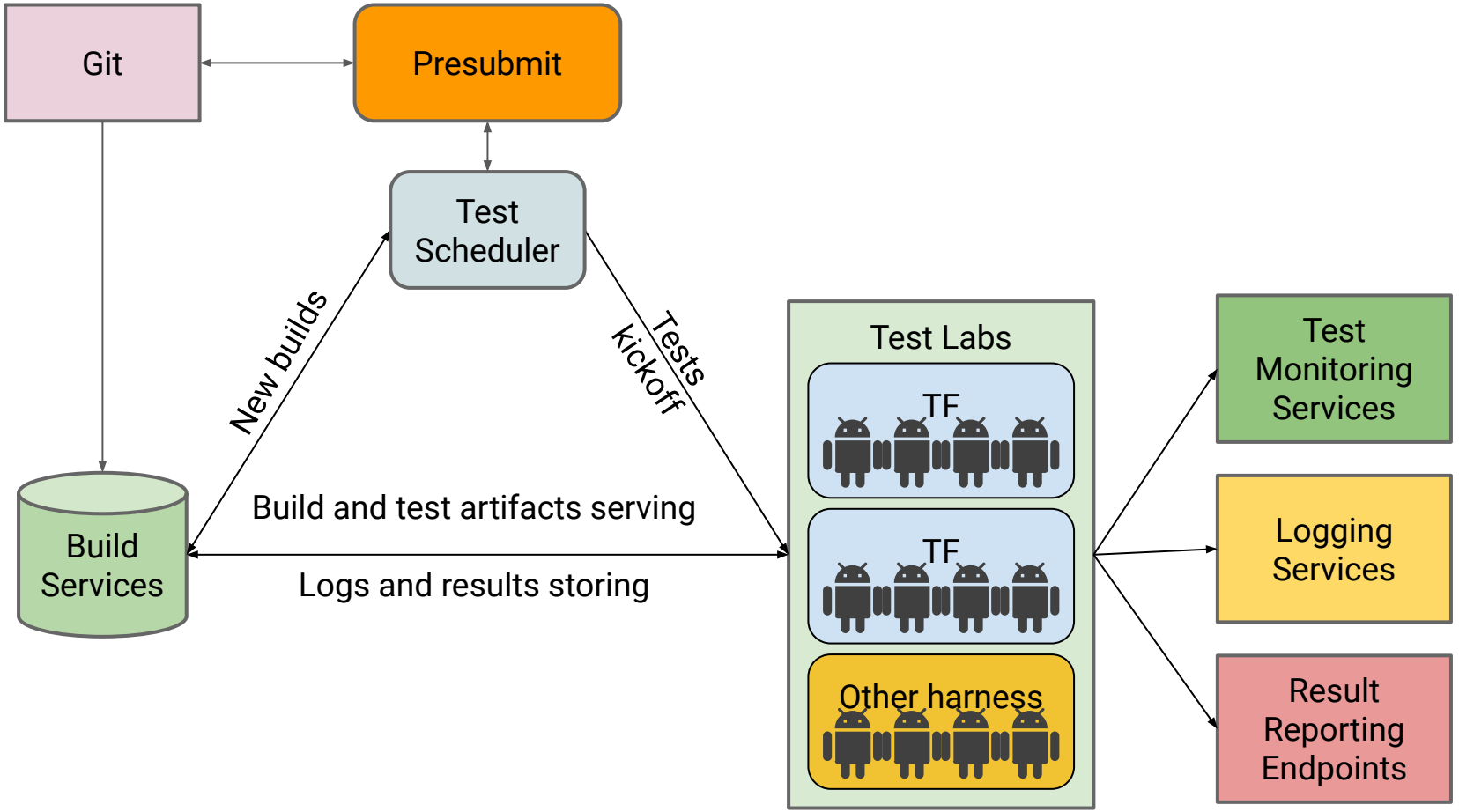
- **Presubmit:** (aka Smoke Tests) Tests that are also executed prior to CL submission by TreeHugger (Presubmit Pipeline)
  - e.g.: Boot test
- **BVT (Build Validation Tests):** Ensure build is valid for additional testing.
  - e.g.: Wifi connectivity test
- **FVT (Functional Validation Tests):** Ensure quality of the build prior to release.
  - e.g.: Stability tests

Software Infrastructure





android



android

Build Infrastructure

- Manages and creates all artifacts required for every step of the development cycle
- In presubmit, it will build the necessary targets to validate the health of a CL.
- In postsubmit, it kicks off builds continuously as CLs are committed to -dev branches.
- In postsubmit, there are also periodic snaps to the -release branch, which are also built
    - There are also high priorities patches, which are cherry picked into -release branch, and they will also trigger builds

Centralized Test Scheduler

- Schedules all tests to be run in distributed labs
  - Both for presubmit and postsubmit
  - Supports both continuous/periodic (automatic) as well as on-demand (user triggered) tests

- Provides dashboard and alerts to monitor labs' health
  - such as devices and hosts status

- Manages build promotion (as mentioned in prior slide) for post submit builds

- Has safeguards in place
  - Auto test suspension
  - Build blacklisting
  - Throttling
  - Checks for SLO

- Provides APIs upon which other services can be built

Distributed Test Laboratories

- Various physical and virtual labs that execute tests
  - Virtual labs host virtual devices which allows us to scale our testing

- “High-touch” vs “Low-touch” labs
  - “High-Touch” lab include custom setups that target specific test verticals
    - Thermal chambers to validate performance at various temperatures
  - “Low-Touch” lab are generic, uniform setup that allows for redundancy and scalability
    - Virtual labs are an example of low touch lab

- Host a variety of test harnesses, test rigs, test devices
  - Centralized test scheduler is agnostic of such setup, which allows for decoupling of test scheduling and execution

Trade Federation (TF)

- Test harness built/maintained by the Android EngProd Infra Team for Android Platform Testing
  - Used locally by Dev's at their workstation as part of their iteration
  - Powers our testing infrastructure as the building block for our CI
- Open Sourced (AOSP): code & documentation
- Extensible - can be used to build end-to-end integration tests
  - Serve as base for *TS, Test Mapping, ATest
- Written in Java, and configured in XML
- There are pre-defined templates, examples for the common case
  - New end-to-end integrations can be written

TreeHugger

- Presubmit infrastructure that runs on changes prior to submission
  - There are static checks for pending CLs as well, such as lint checks

- Monitors incoming changes in specific repos and schedules builds and tests in distributed services (Build and Test Infrastructure)

- Upon completion of builds and tests provides a signal to the user to indicate health of change
  - The status of tests and builds appear in the changelist comments and in the gerrit UI
  - Strong signals from tests and builds will block bad CLs from submission

Next Steps

- Explore interest and understand needs from partners with regards to CI
- Sharing lessons learnt at Google while scaling infrastructure
- Opensourcing infrastructure for the ecosystem

* Disclaimer: No commitment, all ideas/project mentioned in this session and projects are subject to change

Thank you for your time and attention.

Q & A