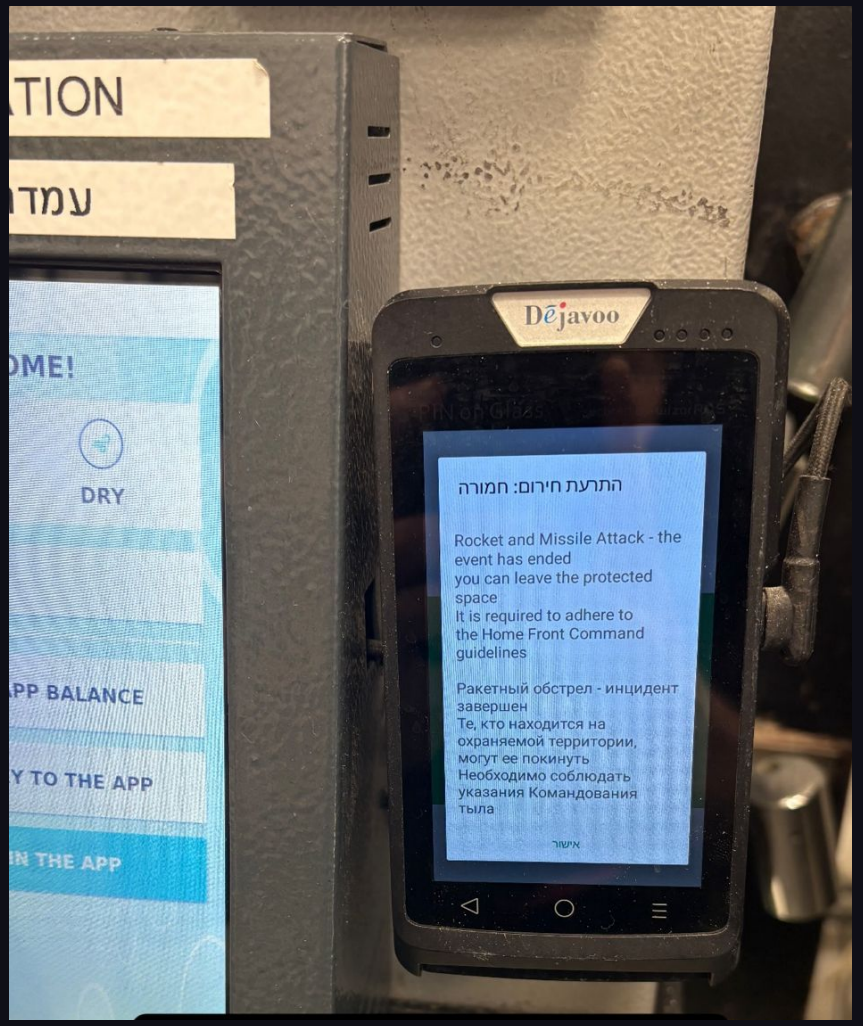
Acceleration × AOSP

Slashing build times without migration


10:17
POWERED BY
android
CAREFUL HOT
DO NOT TOUCH


vodafone.de
Thu 12 Mar · 23 Adar 5786
11:00
Extreme Danger
now
TEST ALERT for Bayern Thu 2026/03/12, 11:00 am - Test alert - for Bayern - There is no danger. - Further information: https://warnung.bund.de/m/-8P4f9wk1Jj_ - Published by: Bayerisches Staatsministerium des Innern, für Sport und Integration - Katastrophenschutz


TION
DRY
APP BALANCE
Y TO THE APP
N THE APP
Dējavoo
התרעת חירום: חמורה
Rocket and Missile Attack - the event has ended
you can leave the protected space
It is required to adhere to the Home Front Command guidelines
Ракетный обстрел - инцидент завершен
Те, кто находится на охраняемой территории, могут ее покинуть
Необходимо соблюдать указания Командования тыла

AI velocity gap

INCREDIBUILD

Founded 25 years ago

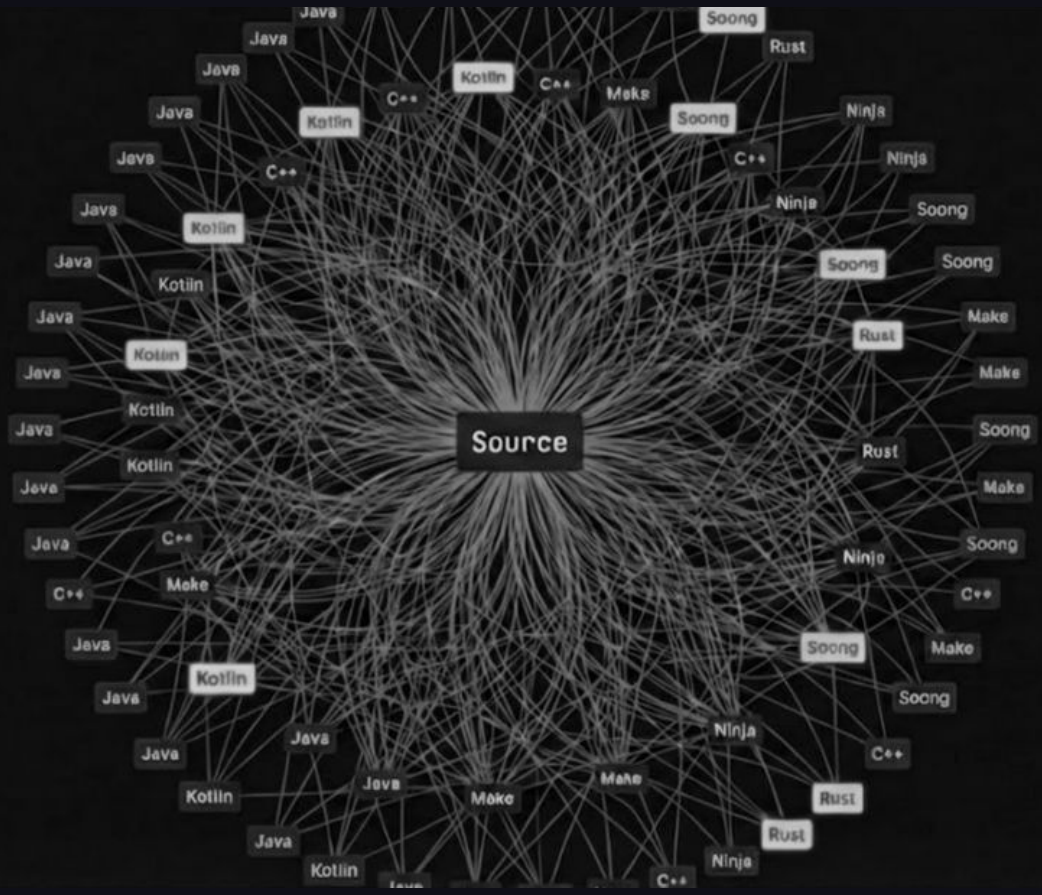
Universal acceleration platform for secured CI/CD pipelines

Zero changes to tools and build scripts

What’s changing in AOSP builds?

The AOSP Complexity Beast

- Variant Explosion - Targets × Products × SoCs
- Multi-Language Pipeline
- Full-Build Penalty
- AI Velocity Shock → CI overload



Slow builds - who cares?

The Real Cost of Slow AOSP Builds

Developer

- Slower iteration cycles
- AI feedback bottlenecks

Platform / CI

- Long build queues
- Repeated full rebuilds

Business

- Escalating cloud spend
- Release friction

What would an optimal accelerator look like?

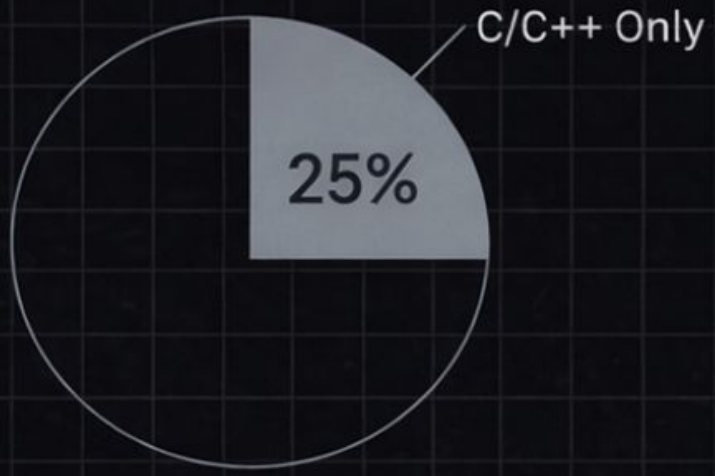
AOSP Acceleration Core Principles

- Dramatic build time reduction
- Zero changes to build tools and scripts
- Low maintenance
- Work across versions and other solutions
- Compute efficiency, cache-first
- Build analysis and visualization

What are my options?

Where Existing Approaches Fall Short

ccache



C/C++ Only

- ✗ **Optimizes C/C++ only**
- ✗ No coverage for Rust, Java, Kotlin
- ✗ **Limited build reuse**

Bazel Migration

- ✗ Rule conversion required across AOSP
- ✗ Ongoing maintenance and migration cost
- ✗ High risk before acceleration payoff

What is Incredibuild’s solution for AOSP acceleration?

Incredibuild for AOSP

- Universal, multi-tool shared caching
- Distributed execution for cache misses
- Zero changes to build tools and scripts
- Works seamlessly across AOSP versions
- Accelerate any build (QNX, Yocto, Kernel, etc)
- Full build visibility

Incredibuild for AOSP

Faster Single-Build

Optimized Throughput & Cloud Costs

Shared Cache & Distribution. Extendable

Zero Changes to Tools and Scripts

CI & Developers. Build Agnostic.

Build Visibility

Data, Auditing, Compliance and Security

How Incredibuild Accelerates AOSP

1 Process Interception

2 Shared Cache

3 Distributed Execution

Process Interception

We observe tasks at execution time (compilers, linkers, tools)

Shared Cache

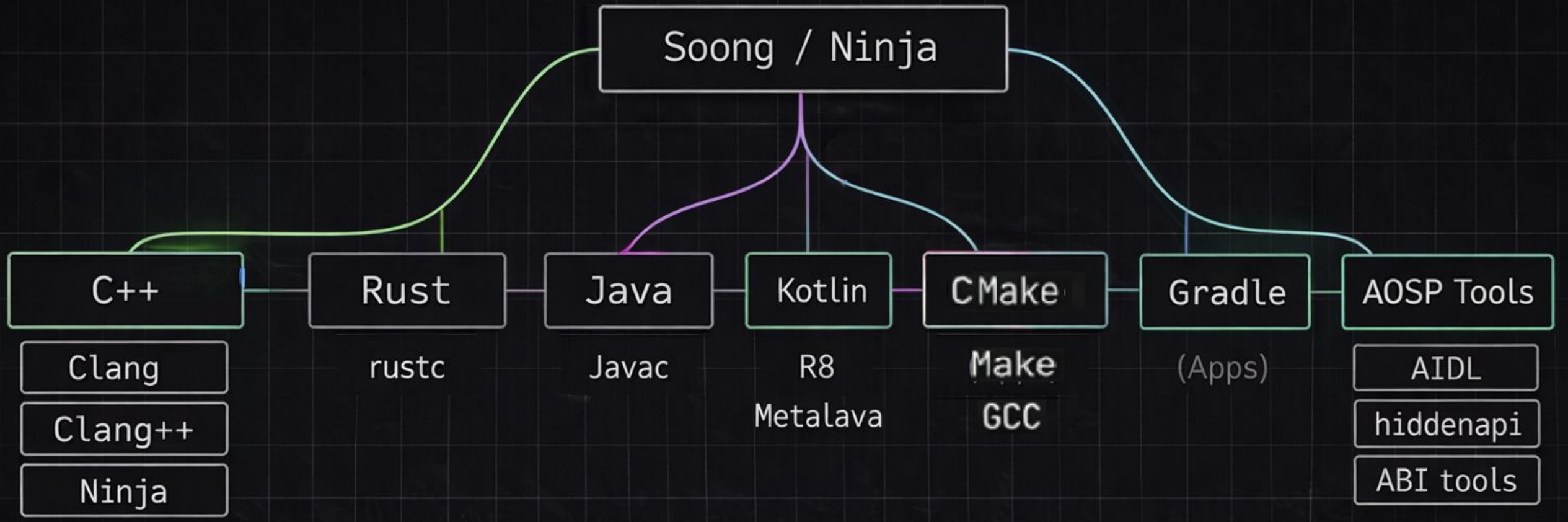
Hash inputs → reuse outputs

Distributed Execution

Cache miss → run remotely at scale

Cache what can be reused. Distribute what must run.

Accelerating the Entire AOSP Execution Graph



We accelerate every process in your AOSP build.

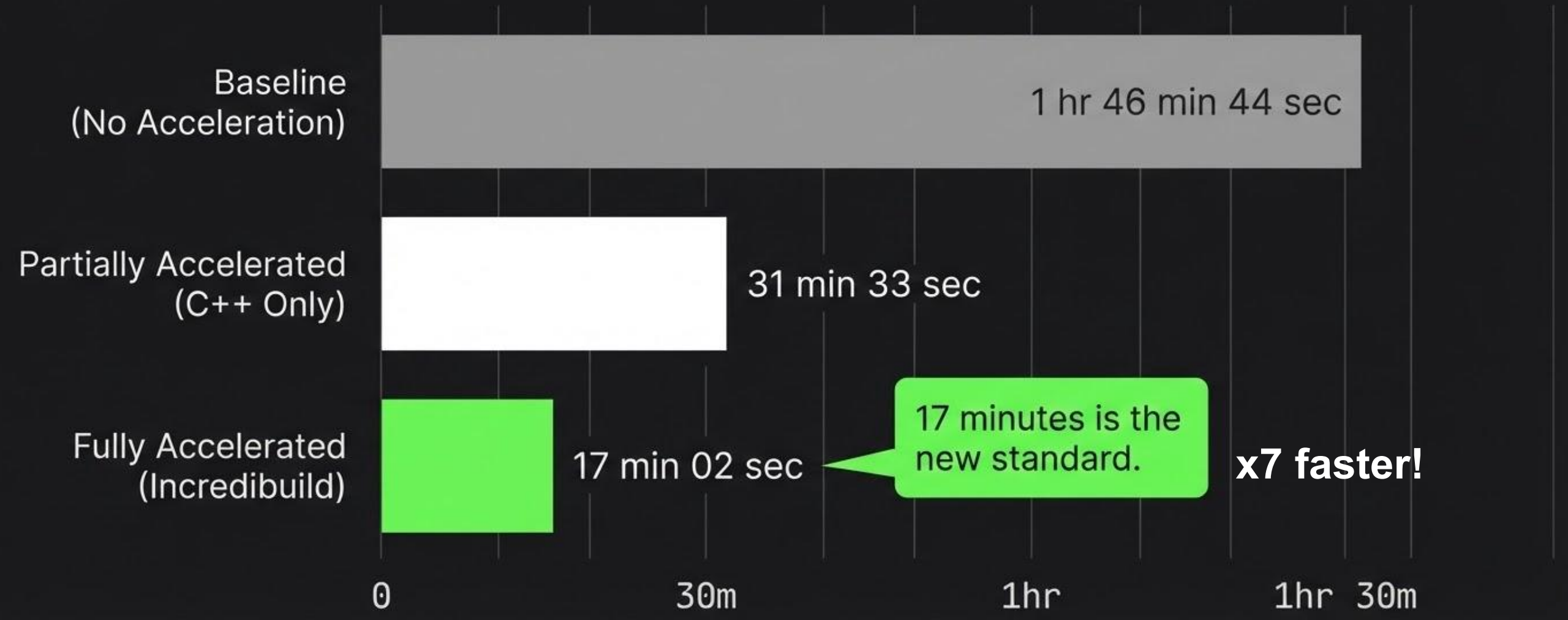
Without touching a single build file.

Benchmarks

From 1h 46m to 17m

Same Hardware. No Script Changes.

AOSP 16 full clean build on a 32-core machine.



Proven Results in the Field

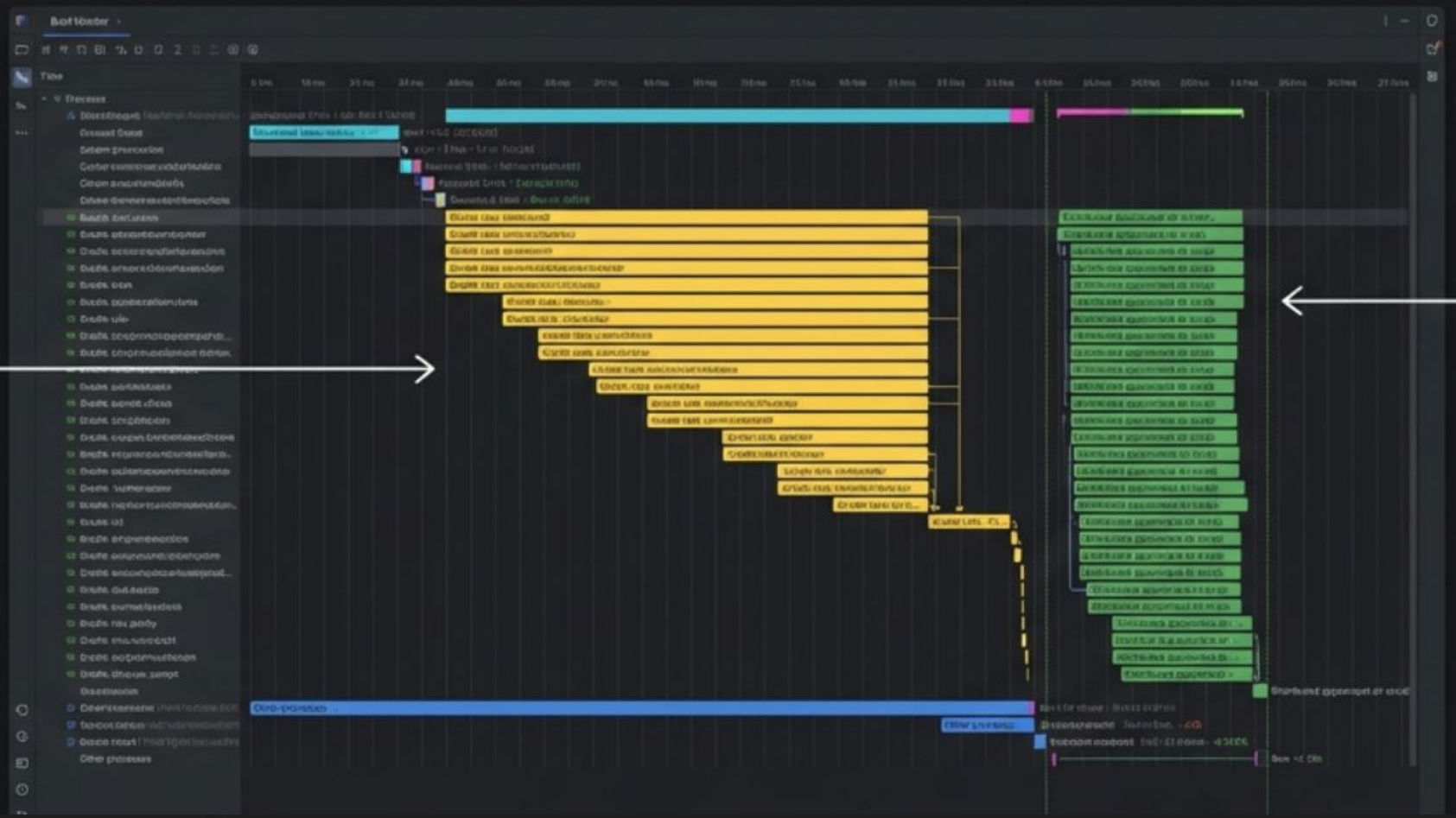
Project	Time Reduction	Impact
AAOS 15 (Major OEM)	110 min → 33 min	3x Faster
QNX ADAS AV	50 min → 18 min	3x Faster
ADAS Perception (EV OEM)	32 min → 7 min	4x Faster
Yocto (AGL 16.4)	192 min → 35 min	5x Faster

"We have aggressive goals within GM to accelerate production... Incredibuild plays an important role." — Ziv Haziz, DevOps & Tools Manager, GM.

X-Ray Vision for Your Build

Visualize bottlenecks and execution flow in real-time.

Gradle (Java apps) total ~16 minutes.



Distributed appcompat.sh script.

What’s Next

Bulk Caching: Stop Re-Executing the Graph

Leaf Caching

Bulk Caching

Leaf Caching

Caches individual tasks.
Graph still runs.

Bulk Caching

Skip entire execution phases.
Restore outputs directly.

17 min → 7 min

When nothing meaningful changed

Build Runners

Accelerated Github actions runners with prewarmed cache

Who is this relevant for?

Don’t Guess. Diagnose Your AOSP Build.

1 RUN IT

- 3 steps
- No installation
- Wrap your existing build

2 WE ANALYZE

- Acceleration potential
- Tool bottlenecks
- CPU / IO / RAM profile

3 YOU DECIDE

- Clear forecast
- Deployment recommendations
- No commitment

We show you exactly how much faster your AOSP build can run.

Build at the speed of your imagination

INCREDIBUILD

Thank you!

Free pipeline assessment

Peleg.rand@incredibuild.com