

Working With Alternative Build Systems

Gautam Korlam & Kurt Nelson 😊
Developer Platform at Uber



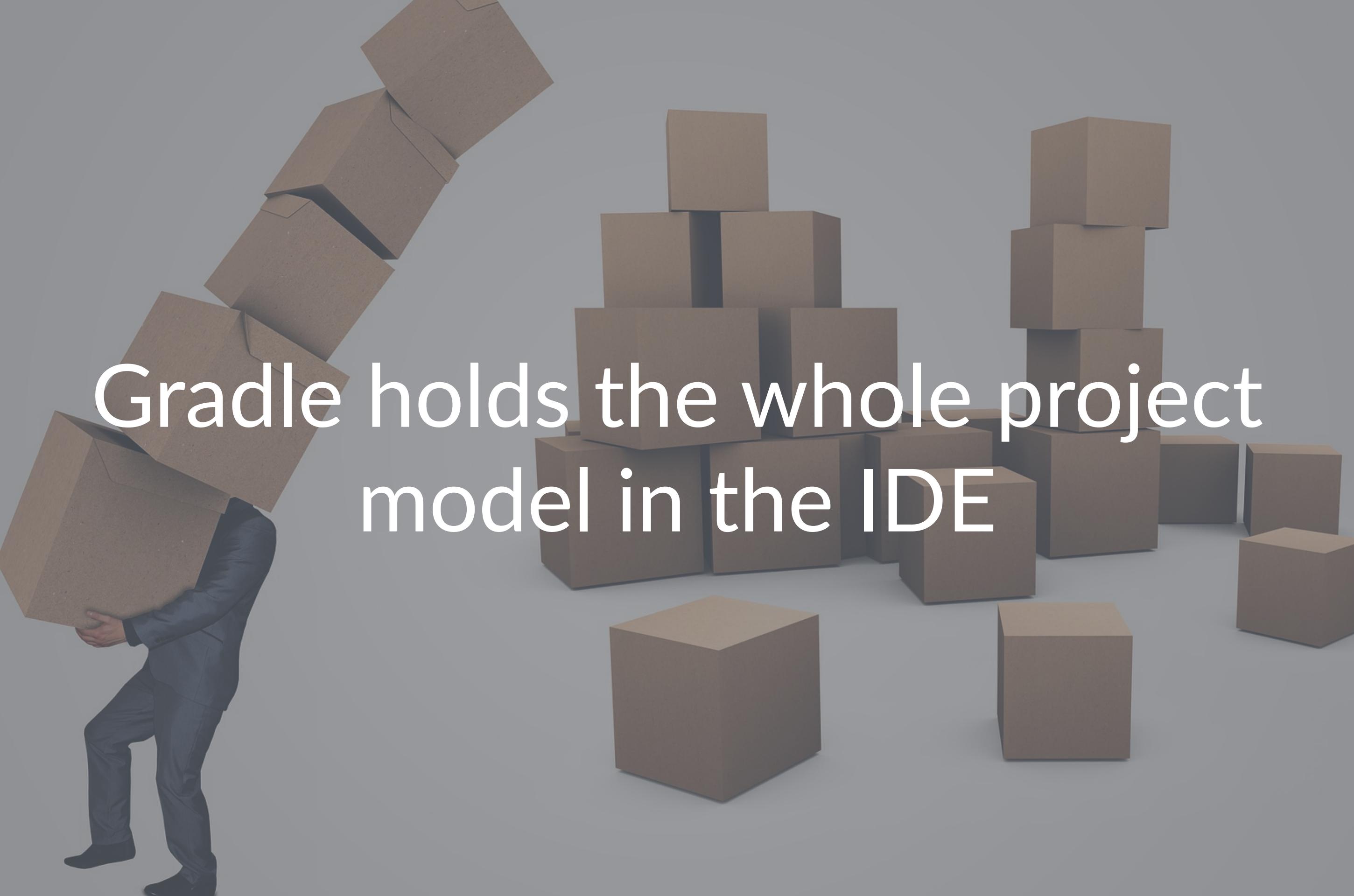
Gradle at scale

A close-up, low-angle shot of a single sea turtle hatchling crawling across a light-colored, sandy beach. The hatchling's dark, patterned shell and flippers are clearly visible as it moves towards the right side of the frame. In the background, the ocean is visible with gentle waves under a sky filled with scattered, soft clouds.

The configuration phase becomes slow

A formation of planes leaving colored smoke trails in the sky.

Groovy and Kotlin are full programming
languages, not configuration languages



Gradle holds the whole project
model in the IDE



Performance tuning is hard

Tuning 3rdparty plugins is harder

A close-up photograph of a row of colorful clothespins hanging from a thin, light-colored string. The clothespins are arranged in a staggered pattern, showing various colors including purple, blue, green, red, and orange. They are held in place by a metal clip at the top of the string. The background is a plain, light gray.

Working with multiple languages is
not easy



Reproducibility is key

The background image shows a dense urban area with numerous buildings packed closely together. The roofs are primarily made of red tiles, and the buildings vary in height and color, including shades of brown, tan, and white. Some have satellite dishes or air conditioning units visible. A few small flags, possibly the Catalan flag, are seen hanging from building facades. The overall scene is a typical view of a European city's residential district.

Dependency management



Extensibility with plugins



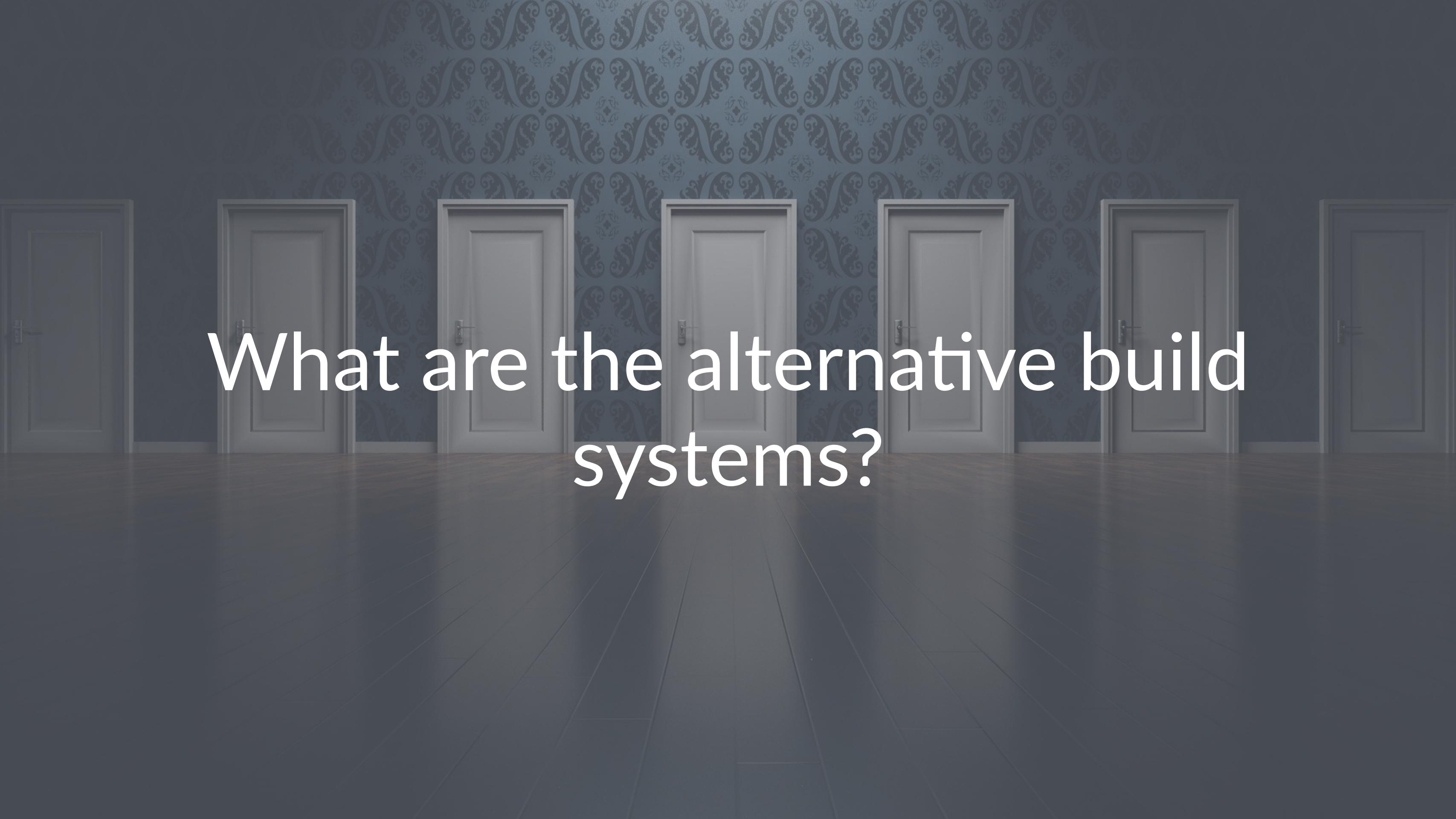
Enterprise feature set



How do we fix these issues?

A language built for builds - Starlark

Deterministic, Hermetic, immutable, performant, simple and built
with tooling in mind

The background of the slide features a dark, atmospheric scene. In the center, there is a row of several closed doors, all identical in style, set against a wall covered in a repeating, ornate damask pattern. The floor is made of dark wood planks. The lighting is dramatic, coming from the front, which creates a bright path on the floor and highlights the edges of the doors, while the rest of the room remains in deep shadow.

What are the alternative build
systems?



Bazel



Buck

Bazel and Buck are language and
toolchain agnostic build systems
that use starlark and are tailored
towards large modular codebases



Bazel

Built by Google

Used by Dropbox, Google, Lyft,
Spotify, Square, Uber

Buck

Built by Facebook

Used by Airbnb, Facebook, Lyft,
Square, Uber



Okbuck

Built by Uber

Hybrid inter-op between Gradle and Buck

Example configurations using Starlark and Bazel

Example android kotlin app¹

```
load("@io_bazel_rules_kotlin//kotlin:kotlin.bzl", "kt_android_library")

kt_android_library(
    name = "lib",
    srcs = glob(["java/com/example/bazel/*.kt"]),
    custom_package = "com.example.bazel",
    manifest = "AndroidManifest.xml",
    resource_files = glob(["res/**"]),
    visibility = ["//:_pkg_:"],
    deps = [
        "@maven//:androidx_appcompat_appcompat",
        "@maven//:androidx_core_core",
        "@maven//:androidx_core_core_ktx",
        "@maven//:androidx_drawerlayout_drawerlayout",
        "@maven//:androidx_fragment_fragment",
        "@maven//:androidx_lifecycle_lifecycle_common",
        "@maven//:androidx_lifecycle_lifecycle_viewmodel",
    ],
)
```

¹ For more examples, checkout [github](#)

Example android kotlin app¹

```
android_binary(  
    name = "app",  
    custom_package = "com.example.bazel",  
    manifest = "AndroidManifest.xml",  
    deps = [  
        ":lib",  
    ],  
)
```

¹ For more examples, checkout [github](#)

Why use an alternative build system?



Scale better as  your codebase grows

Design differences

Gradle

Task Graph

Lazy if configured by task

Combining different languages
in one build is difficult

Bazel

Action Graph

All targets lazy by default

Fully native graph of different
languages possible

Combining Languages

```
cc_library(  
    name = "main-jni-lib",  
    srcs = [  
        "@local_jdk//:jni_header",  
        "@local_jdk//:jni_md_header-linux",  
        "Main.cc"  
    ],  
    hdrs = [ "Main.h" ],  
    includes = [ "external/local_jdk/include", "external/local_jdk/include/linux" ],  
)  
  
cc_binary(  
    name = "libmain-jni.so",  
    deps = [ ":main-jni-lib" ],  
    linkshared = 1,  
)
```

Combining Languages

```
java_binary(  
    name = "Main",  
    srcs = [ "Main.java" ],  
    main_class = "Main",  
    data = [ ":libmain-jni.so" ],  
    jvm_flags = [ "-Djava.library.path=." ],  
)
```

Fool proof extensibility

Gradle

Caching is opt-in

Easy to have side effects

Need to understand plugin api and
internals

Modifying existing behavior is not
straight forward

Bazel

Caching is built in

Impossible to have side effects

Standard Starlark

All rules can be wrapped in a macro

Extending using macros²

```
def kt_android_library(name, exports = [], visibility = None, **kwargs):
    """Creates an Android sandwich library. `srcs`, `deps`, `plugins` are routed to `kt_jvm_library` the other android related attributes are handled by the native `android_library` rule.
    """
    native.android_library(
        name = name,
        exports = exports + _kt_android_artifact(name, **kwargs),
        visibility = visibility,
        testonly = kwargs.get("testonly", default=0),
    )
```

² For more details on bazel kotlin rules, checkout [github](#)

Extending using macros²

```
def _kt_android_artifact(name, srcs = [], deps = [], plugins = [], **kwargs):
    """Delegates Android related build attributes to the native rules but uses the Kotlin builder to compile Java and
    Kotlin srcs. Returns a sequence of labels that wrapping macro should export.
    """
    base_name = name + "_base"
    kt_name = name + "_kt"

    base_deps = deps + ["@io_bazel_rules_kotlin//kotlin/internal:jvm:android_sdk"]

    native.android_library(
        name = base_name,
        visibility = ["//visibility:private"],
        exports = base_deps,
        **kwargs
    )
    _kt_jvm_library(
        name = kt_name,
        srcs = srcs,
        deps = base_deps + [base_name],
        plugins = plugins,
        testonly = kwargs.get("testonly", default=0),
        visibility = ["//visibility:private"],
    )
    return [base_name, kt_name]
```

² For more details on bazel kotlin rules, checkout [github](#)

Organizational

Gradle

Popular standard for Java/Kotlin

Android developers know it already

Follow Android plugin updates

Supported by Google and usable out of the box

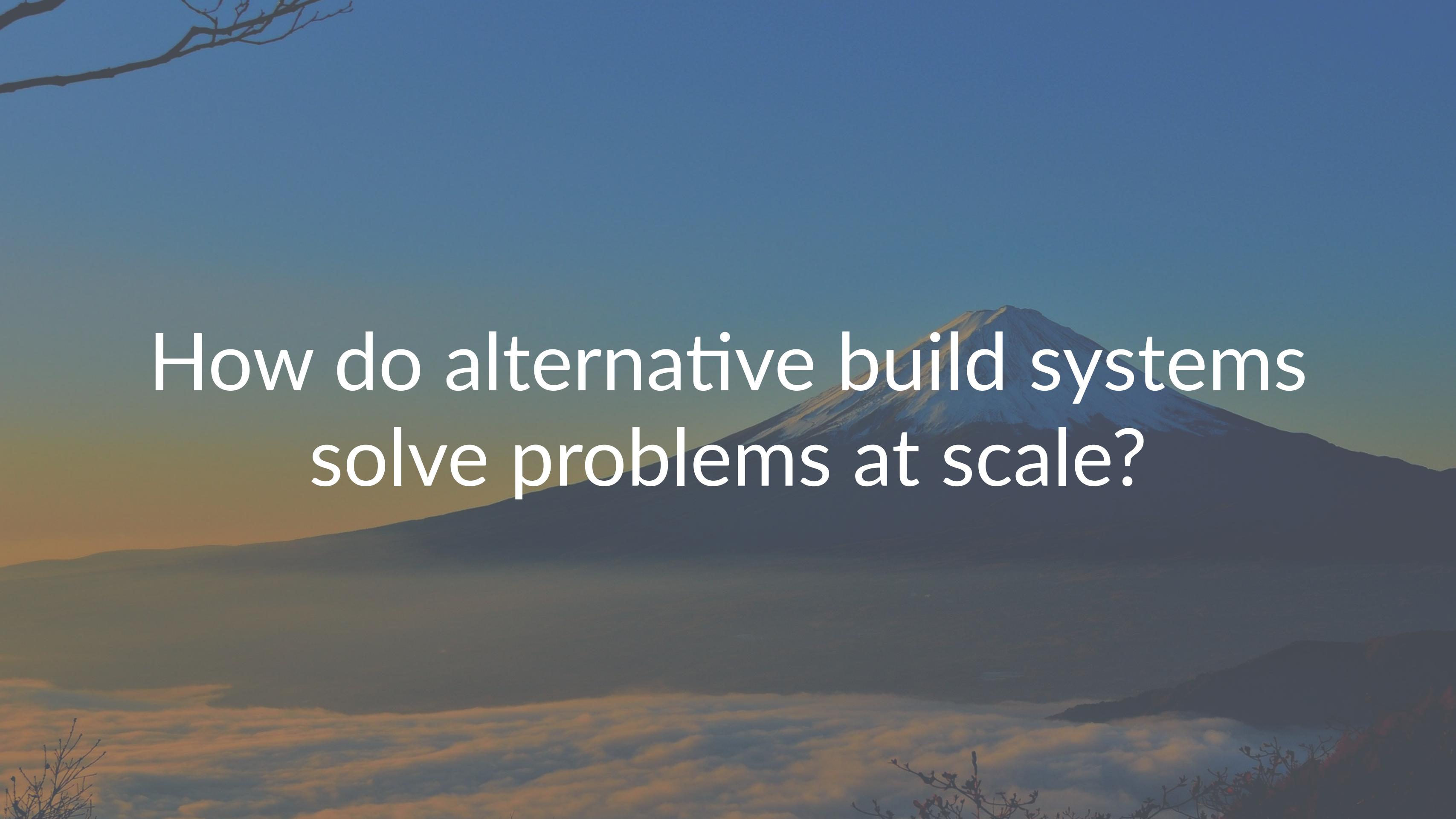
Bazel

Any Language or platform, including iOS

Android developers can build iOS or backend

Local control over when updates happen

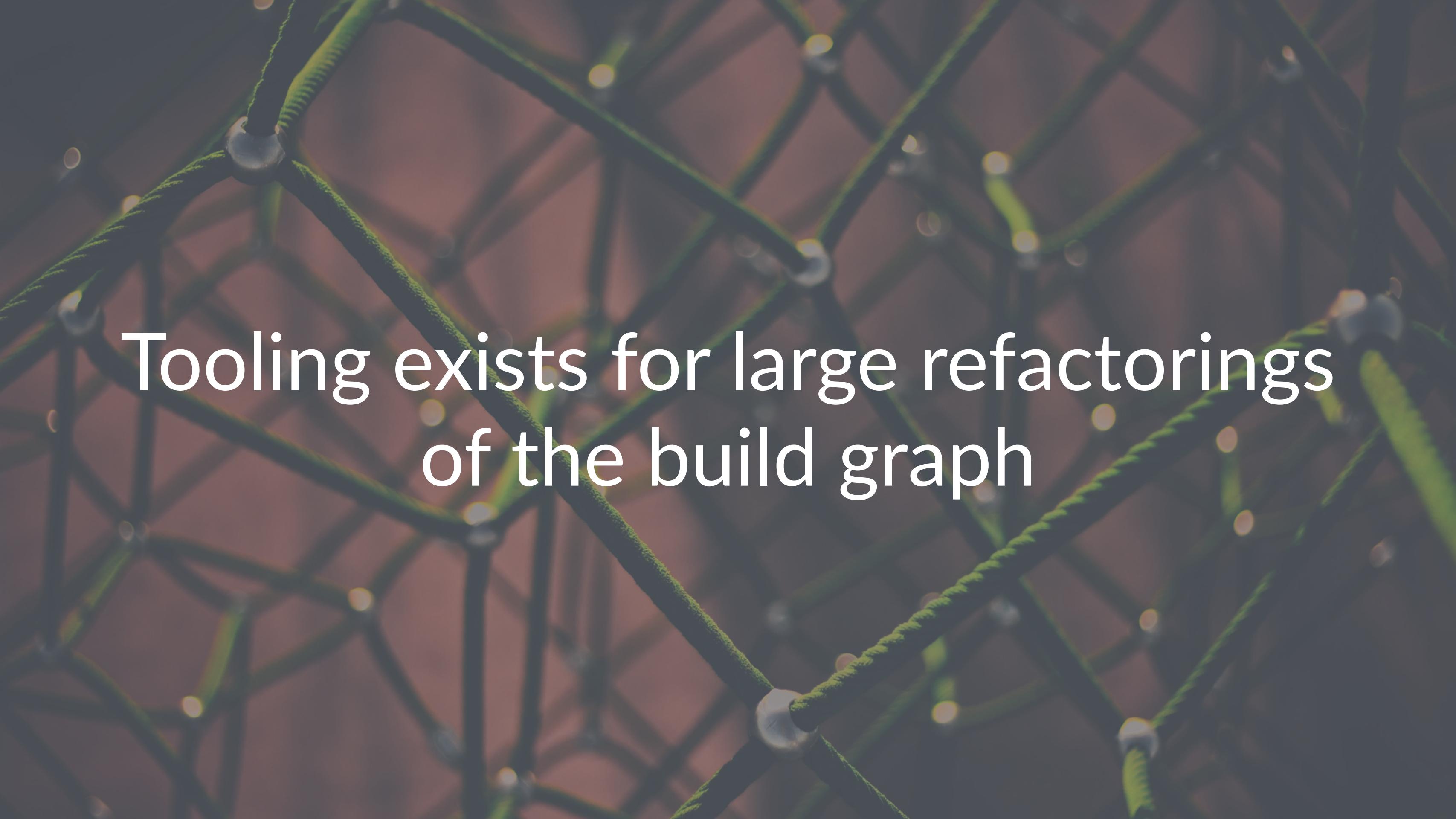
Requires resourcing



How do alternative build systems
solve problems at scale?



Hermetic and reproducible builds
out of the box

The background of the slide features a close-up, low-angle shot of a climbing rope and metal carabiners against a dark background. The rope is a vibrant green color, and the carabiners are silver. The lighting creates bright highlights on the metallic parts and deep shadows in the crevices of the rope, giving it a textured, almost organic appearance.

Tooling exists for large refactorings
of the build graph

Shared infrastructure for all

All languages are on equal footing

All targets can be remotely built and cached

All platforms can share common code like models

A space shuttle is shown launching from a launch pad. The shuttle is white with a dark blue thermal protection system on its nose and wings. It is mounted on top of a large white External Tank. Two solid rocket boosters are attached to the sides of the External Tank. The shuttle is angled upwards, and a massive plume of white smoke and orange fire is billowing out from the base of the External Tank. The background is a clear blue sky.

Changes the developer experience

A photograph of the Great Pyramids of Giza at sunset or sunrise. The pyramids are silhouetted against a vibrant sky filled with orange, yellow, and pink hues. The foreground shows the sandy desert ground.

Flakiness is a thing of the past!

A detailed illustration of a steampunk-style pocket watch. The watch has a gold-colored case with a textured surface. The dial is white with black Roman numerals at 12, 3, 6, and 9 o'clock. There are also smaller numbers and dots between the main numerals. The hands are black with ornate gold-colored crowns at the ends. Behind the dial, several gold-colored gears of different sizes are visible, some with small red lights or glowing points. The watch is attached to a dark brown leather strap with gold-colored metal links.

Take updates on your own cadence



Adding new targets is simple

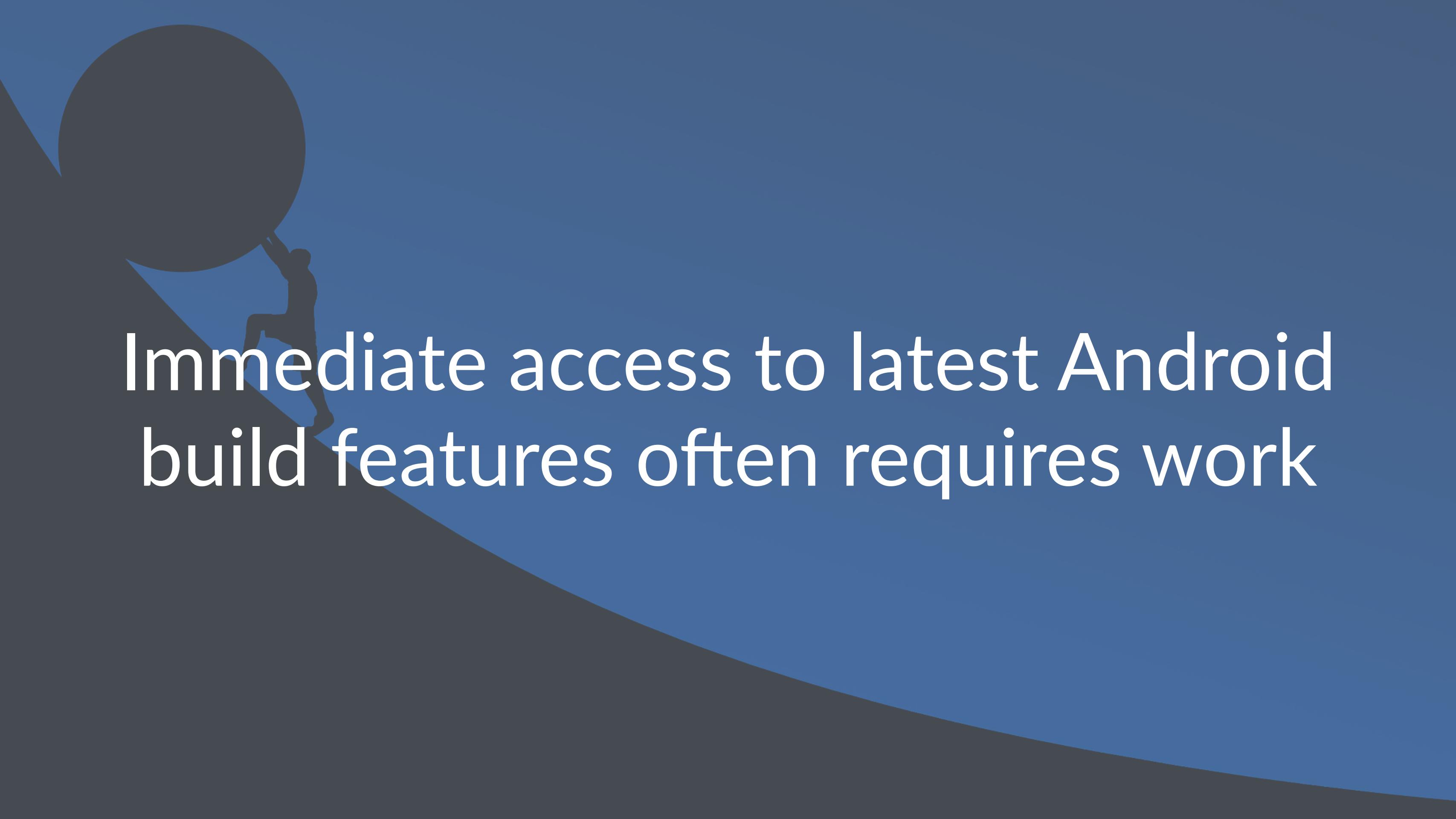
Developers naturally build smaller targets

An aerial photograph of a city street scene. In the foreground, there's a multi-lane road with several cars and a bus. To the left, a large building with a grey facade and a flat roof is visible. In the center, a tall residential or office building with many windows and a light-colored facade stands prominently. To the right, another large building with a blue-tinted glass facade is partially visible. The background shows more buildings and a mix of paved roads and green areas with trees.

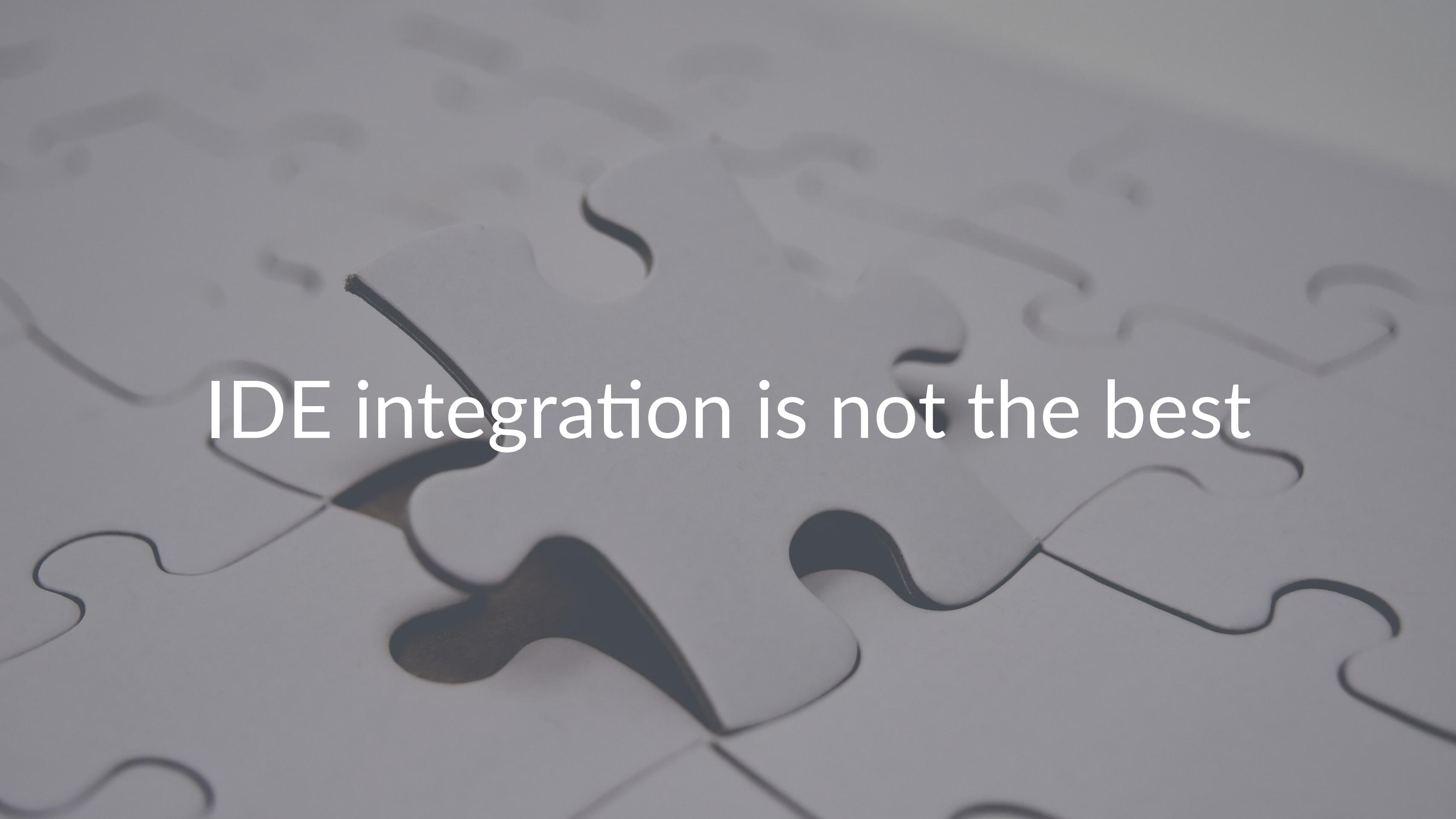
All developers are empowered to
extend the build system



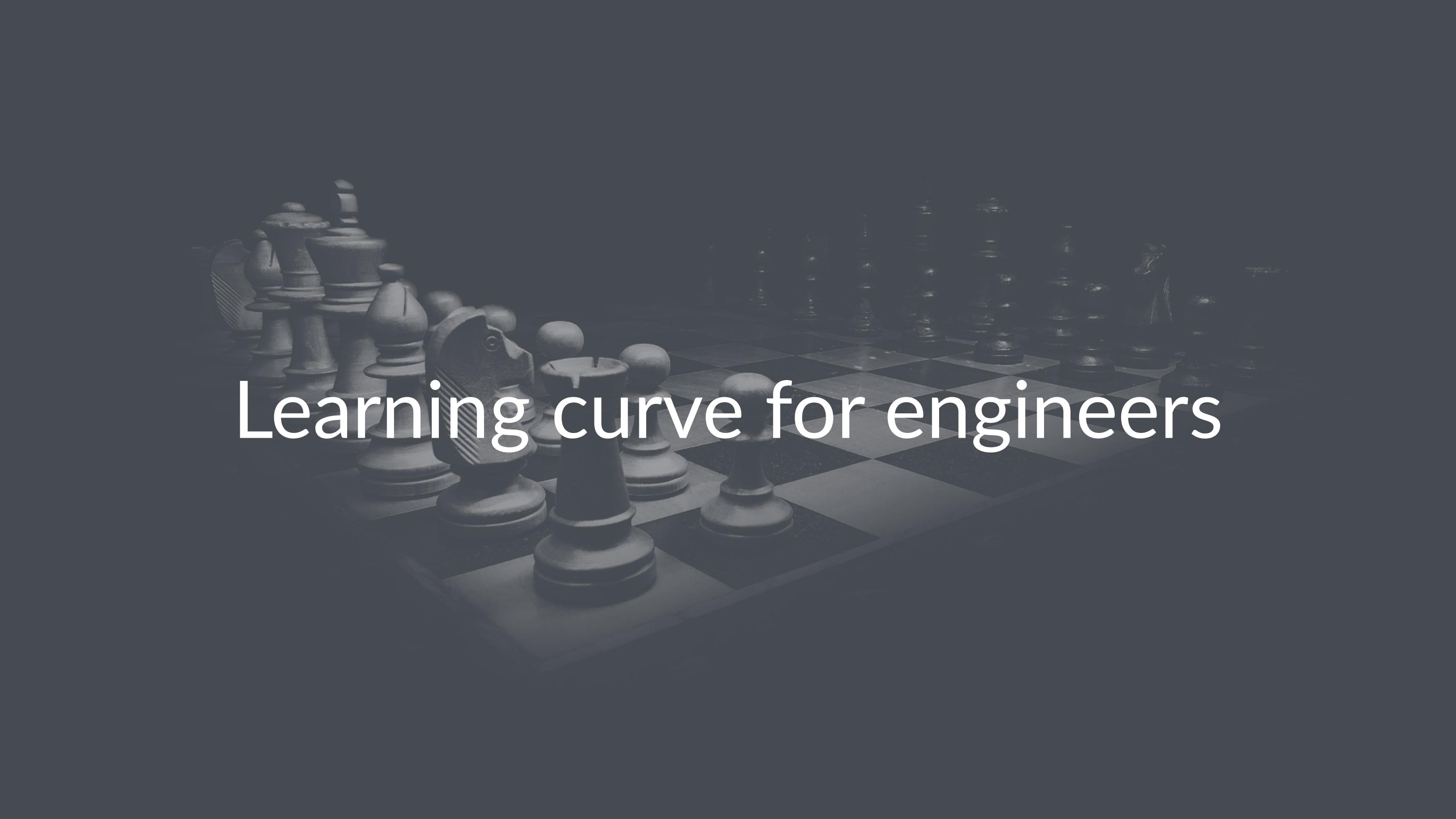
Downsides



Immediate access to latest Android
build features often requires work



IDE integration is not the best

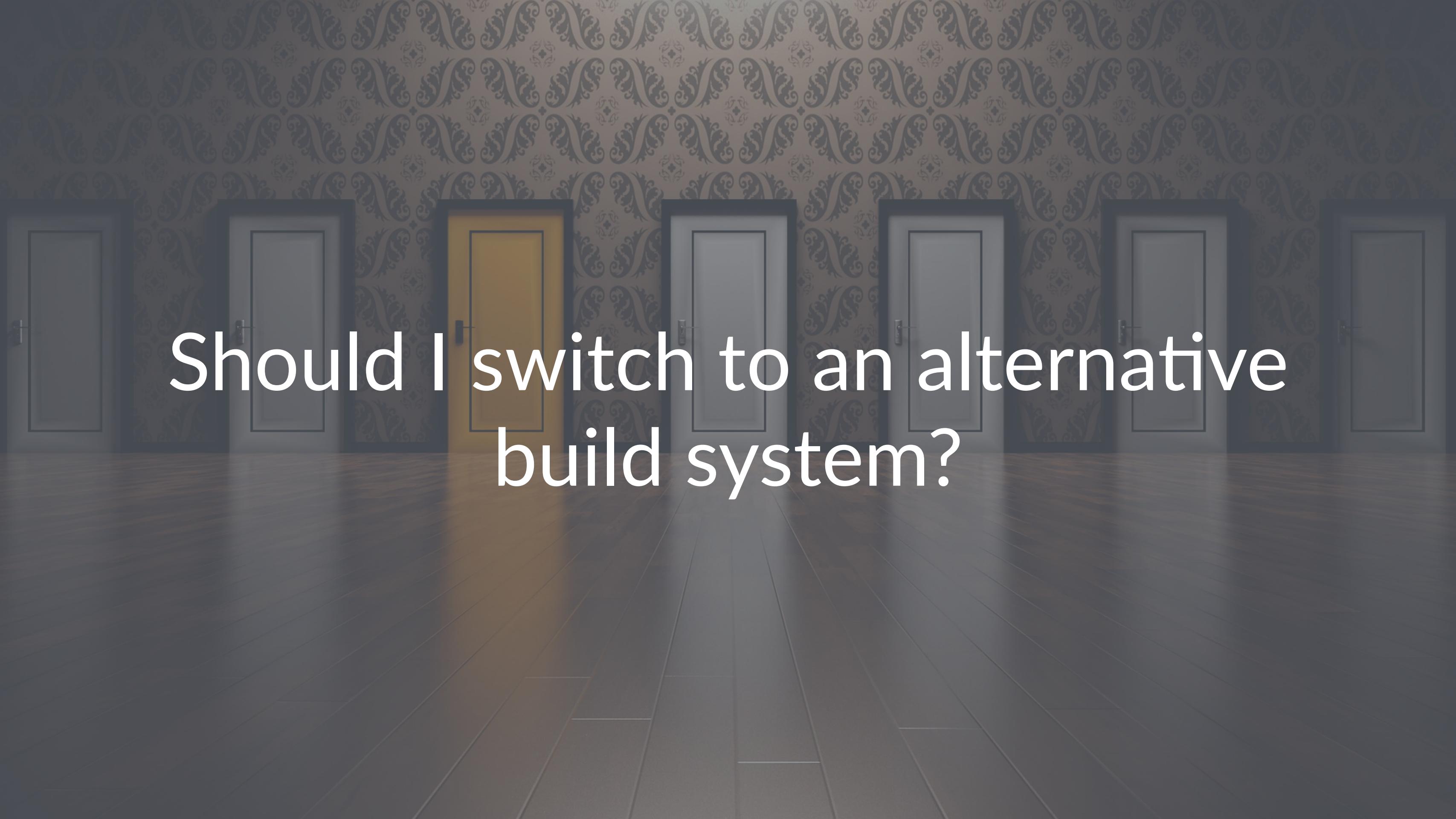


Learning curve for engineers

The background of the slide features a dramatic sunset or sunrise scene with a bright yellow-orange glow in the center, surrounded by dark, textured clouds. In the foreground, the dark silhouette of a person is climbing a large, gnarled tree. The person's arms and legs are visible as they grip the branches. The tree's trunk and branches are silhouetted against the bright background.

Community and resources³

³ More info about [BazelCon](#)



Should I switch to an alternative
build system?

A photograph of a woman with blonde hair, seen from behind, standing on a rocky outcrop. She is wearing a dark blue jacket, dark pants, and red boots. She is looking out over a vast landscape that includes a city built on hills, a large body of water, and distant mountains under a hazy sky.

When should I consider it?

A photograph of a railway track that branches into two separate paths, curving away from each other. The tracks are made of dark metal rails and are set on a bed of light-colored gravel. In the background, there are several tall, grey metal utility poles with multiple cross-arms holding up power or signal cables. The surrounding area is lush with green trees and bushes under a clear blue sky.

How do I go about switching to an
alternative?

Thanks

Questions?