

Evolve a Legacy Game Engine with Open Source Tools

Kai Zhang
NetEase

Agenda

Evolve a legacy game engine with open source tools

Bazel and related open source tools

Clang-tidy

Include-What-You-Use

Experience working with the open source community

Background



EVE Echoes



NeoX

NeoX



LifeAfter

NeoX



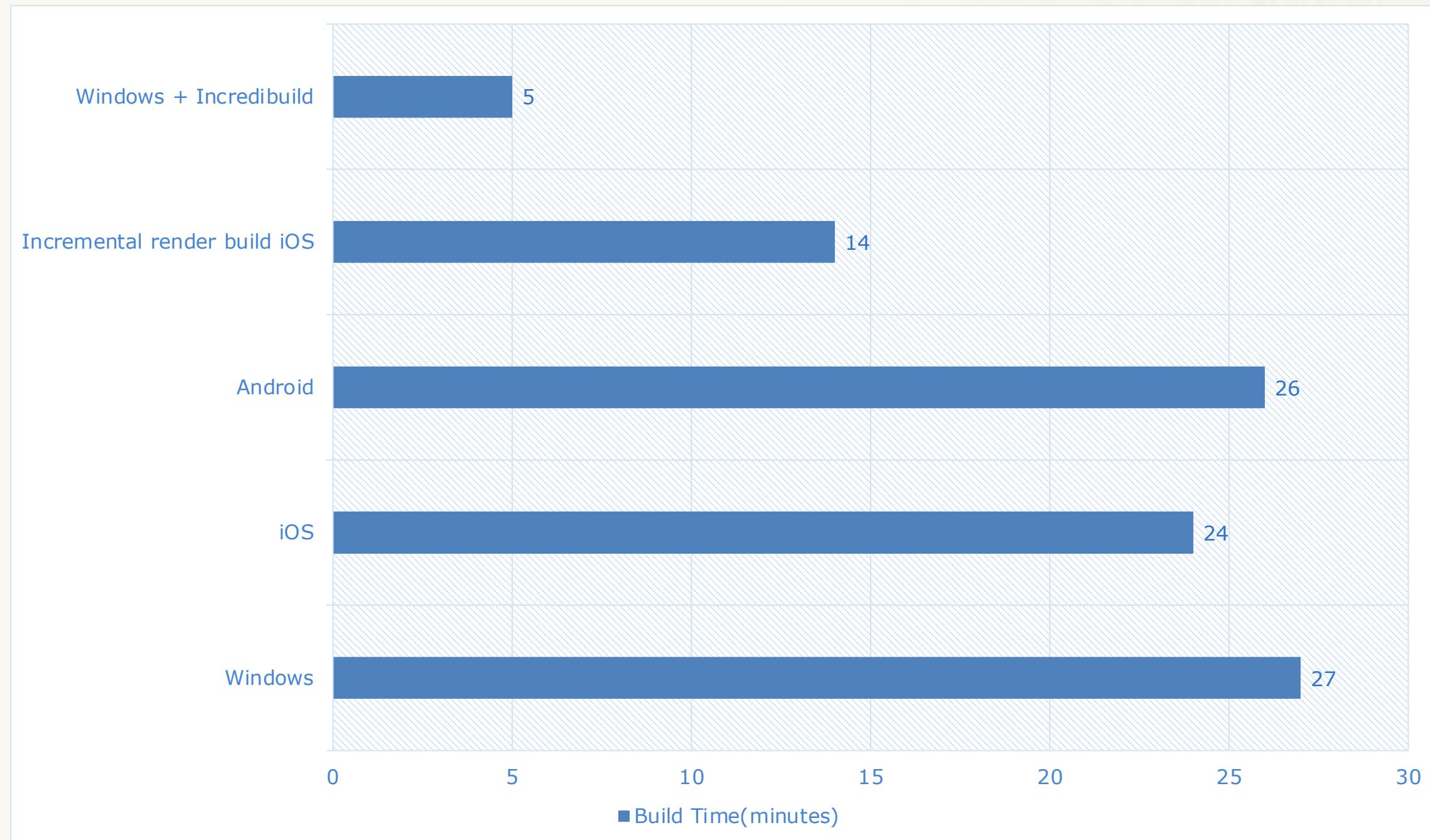
Identity V

NeoX

Pet Kingdom Online - 2009



Build Time

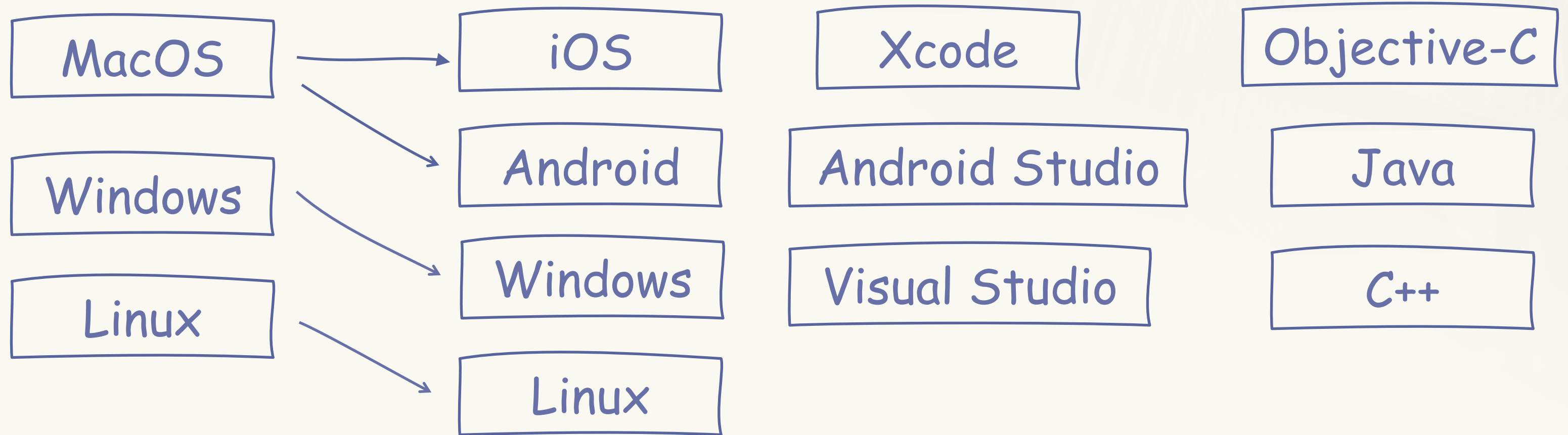


Team Size

- 20 developers
- 4 working on engine fulltime



Challenge



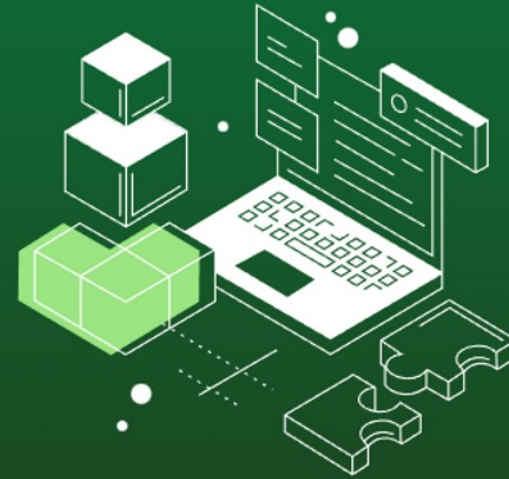
Remote cache and execution

Bazel

{ Fast, Correct } — Choose two

From startup to enterprise, choose the Bazel open source project to build and test your multi-language, multi-platform projects.

Install Bazel



Build better

Rebuild only what is necessary. Get fast, incremental builds with Bazel's advanced local and distributed caching, optimized dependency analysis, and parallel execution.



Multilingual magic

Build and test using Java, C++, Go, Android, iOS and many other languages and platforms. Bazel runs on Windows, macOS, and Linux.



Simply scalable

Scale your organization, codebase, and Continuous Integration systems. Bazel handles codebases of any size, whether in multiple repositories or a huge monorepo.

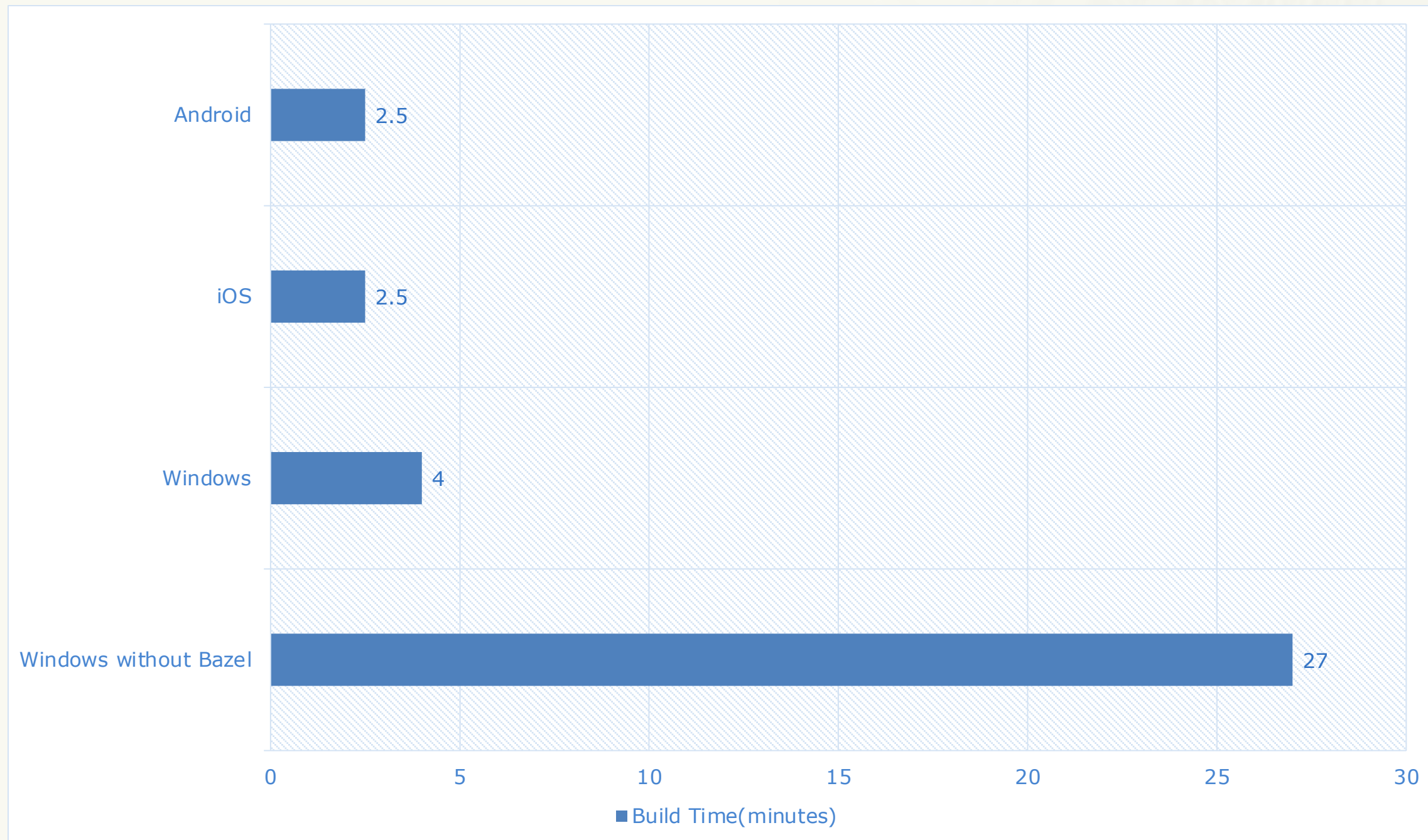


Endlessly extensible

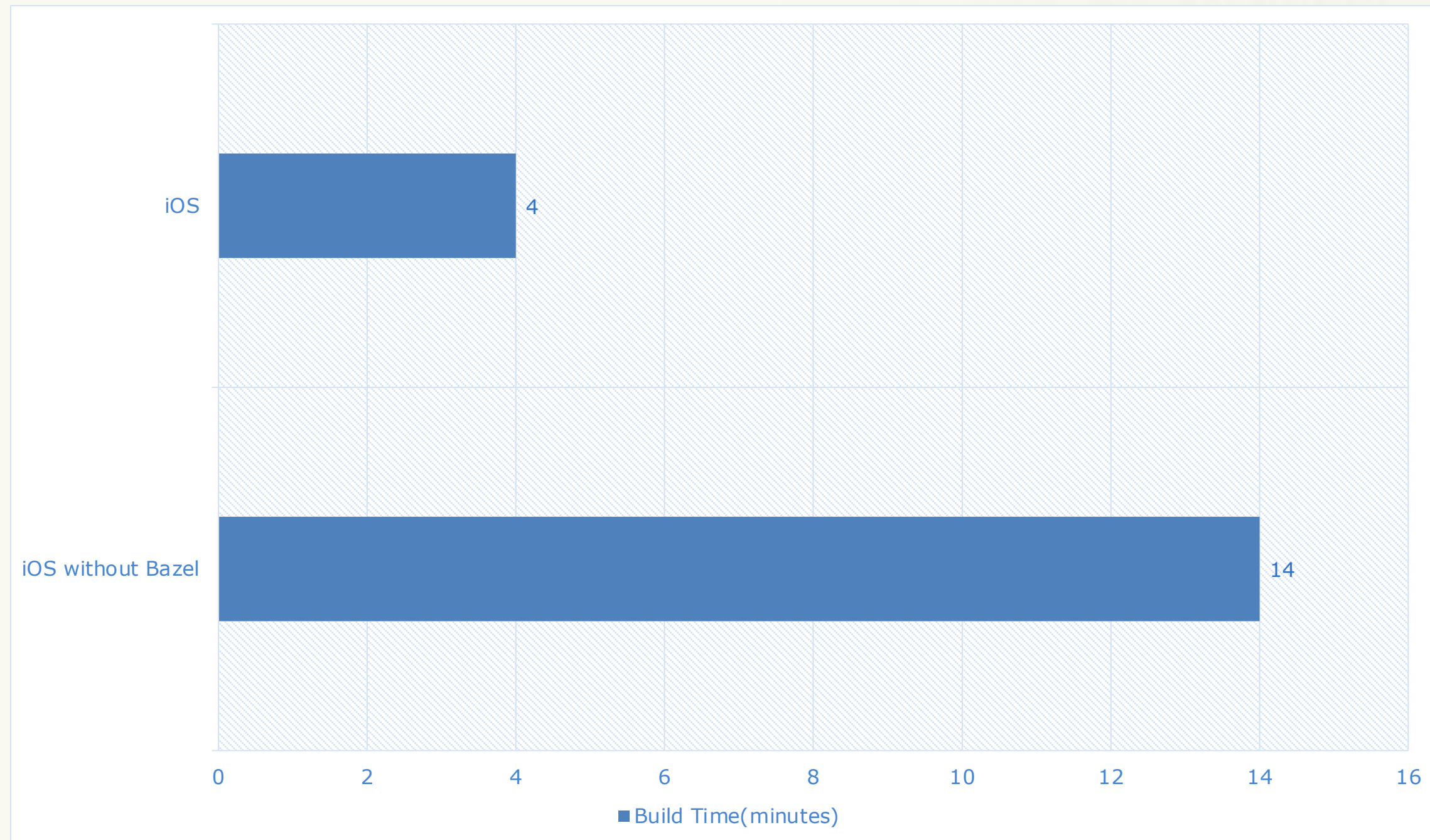
Add support for new languages and platforms with Bazel's extension language. Share and re-use language rules written by the growing Bazel community.

<https://bazel.build/?hl=en>

Result – full build with cache



Result – Incremental Render Build



BUILD.bazel

```
1 cc_library(  
2     name = "math",  
3     hdrs = [  
4         "vector.h",  
5         "matrix.h",  
6     ],  
7     srcs = [  
8         "vector.cpp",  
9         "matrix.cpp",  
10        "_vector_imp.h",  
11        "_matrix_imp.h",  
12    ],  
13     deps = [  
14         "portable"  
15    ]  
16 )
```

```
1 project("math")  
2  
3 add_library(${PROJECT_NAME}  
4     vector.cpp  
5     matrix.cpp  
6     _vector_imp.h  
7     _matrix_imp.h  
8 )  
9  
10 set(public_headers vector.h matrix.h)  
11  
12 set_target_properties(${PROJECT_NAME}  
13     PROPERTIES  
14     PUBLIC_HEADER  
15     "${public_headers}")  
16 )  
17  
18 target_link_libraries(${PROJECT_NAME} PUBLIC portable)
```

Starlark

- <https://github.com/bazelbuild/starlark>
- Python dialect
- Purely declarative (in BUILD.bazel)
- Same input always get same output

```
[_ NeoX [demo_for_gdc] ⚡ date "+%Y-%m-%d %H:%M:%S"  
2023-03-08 18:18:48  
[_ NeoX [demo_for_gdc] ⚡ date "+%Y-%m-%d %H:%M:%S"  
2023-03-08 18:18:56
```

Bazel Sandbox

NeoX

- container
- hashmap.h
- hashmap.cpp
- set.h
- set.cpp

```
1  cc_library(  
2      name = "hashmap",  
3      hdrs = ["hashmap.h"],  
4      srcs = ["hashmap.cpp"]  
5  )  
6  
7  cc_library(  
8      name = "set",  
9      hdrs = ["set.h"],  
10     srcs = ["set.cpp"]  
11 )
```


Bazel Sandbox

NeoX

- container
- hashmap.h
- hashmap.cpp
- set.h
- set.cpp

sandbox_dir

- container
- hashmap.h
- hashmap.cpp

----->

----->



```
clang -c ./container/hashmap.cpp -I./container
```

Move to Bazel

- Build with Bazel
- Integrate with IDEs
- Remote cache and execution

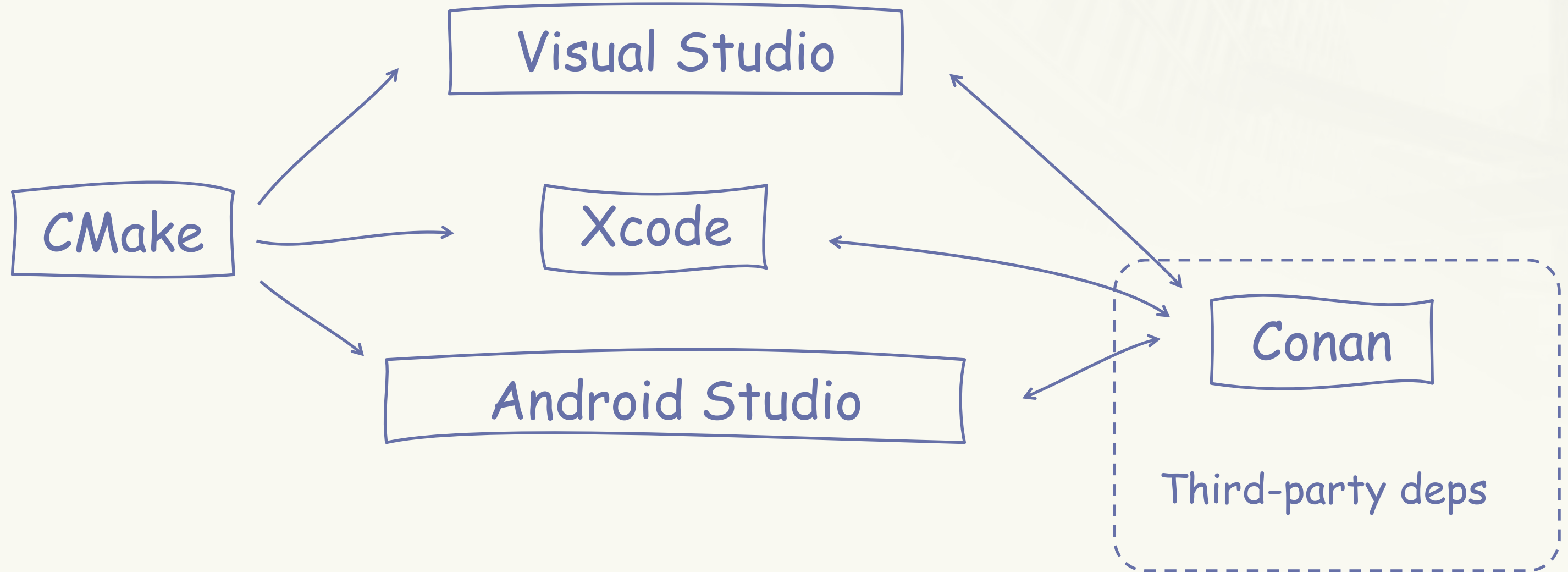


Move to Bazel

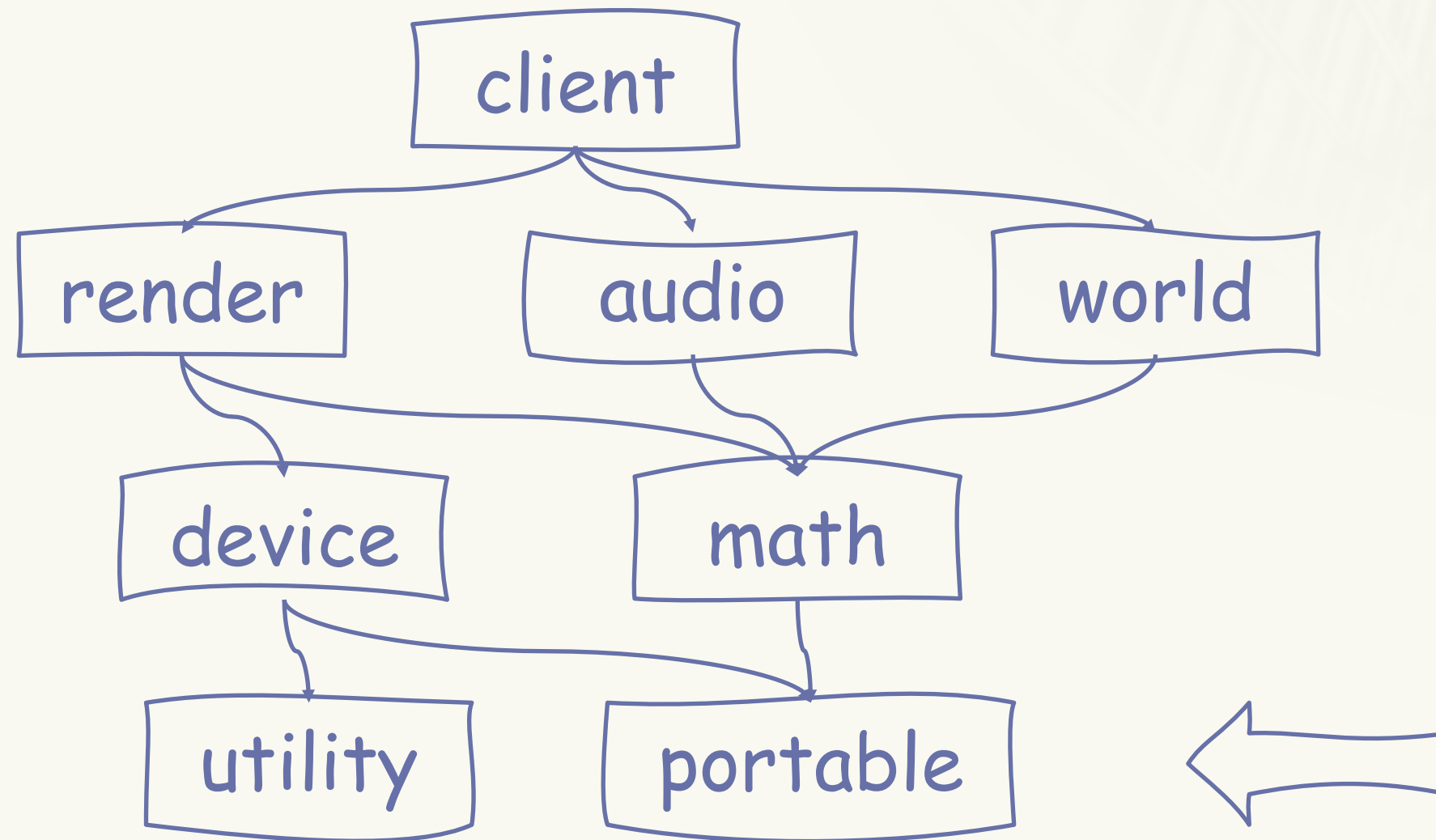
- Build with Bazel
- Integrate with IDEs
- Remote cache and execution



Migration



Migration



Migration



```
1 cc_library(  
2     name = "portable",  
3     hdrs = ["portable.h"],  
4     srcs = ["portable.cpp"],  
5     visibility = ["//visibility:public"],  
6 )
```

```
[NeoX [demo_for_gdc] ⚡ bazel build //:portable  
INFO: Analyzed target //:portable (0 packages loaded, 0 targets configured).  
INFO: Found 1 target...  
Target //:portable up-to-date:  
  bazel-bin/libportable.a  
INFO: Elapsed time: 8.299s, Critical Path: 7.92s  
INFO: 3 processes: 1 internal, 2 darwin-sandbox.  
INFO: Build completed successfully, 3 total actions
```


Migration

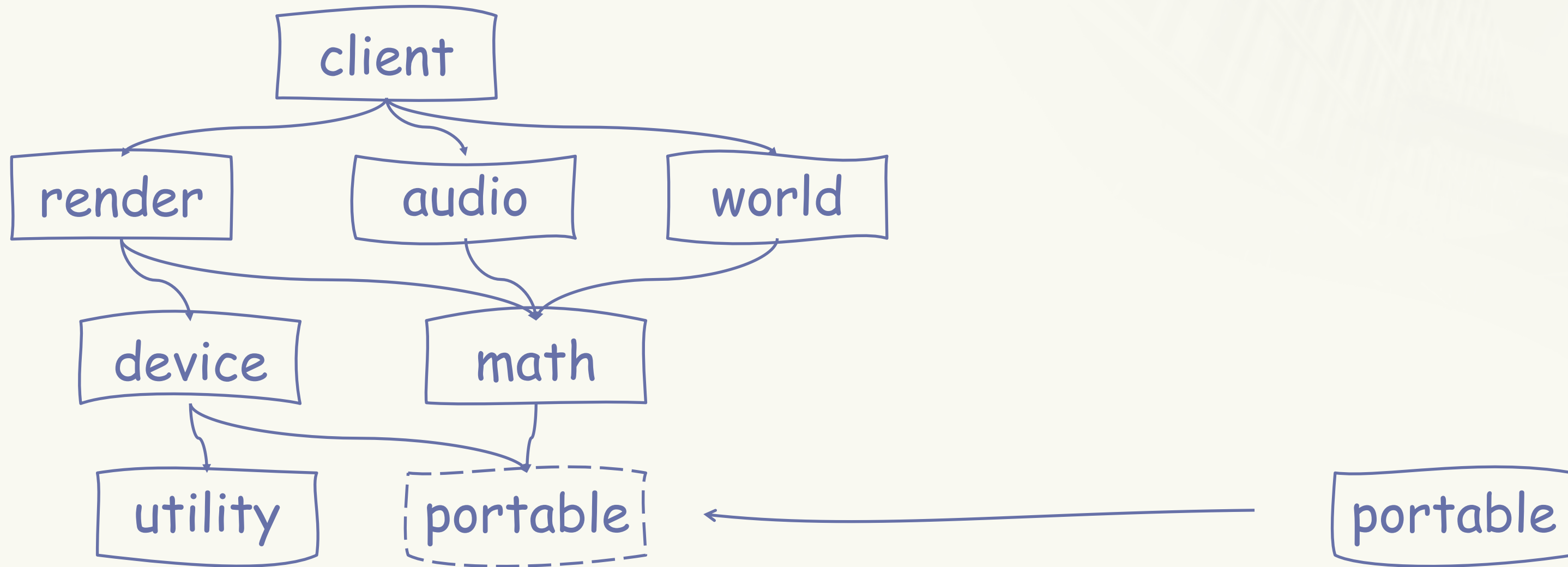


```
1 add_library(portable STATIC IMPORTED GLOBAL)
2 set_property(TARGET ${TARGET} PROPERTY IMPORTED_LOCATION ${BAZEL_OUTPUT_PATH})
```



```
1 add_custom_target(
2     portable_bazel_build
3     COMMAND bazel build //:portable
4     DEPENDS install_conan_deps
5     BYPRODUCTS ${BAZEL_OUTPUT_PATH}
6     SOURCES ${HEADER_FILES} ${INLINE_FILES} ${SOURCE_FILES}
7     WORKING_DIRECTORY ${NEOX_ROOT_DIR})
8 add_dependencies(portable portable_bazel_build)
```

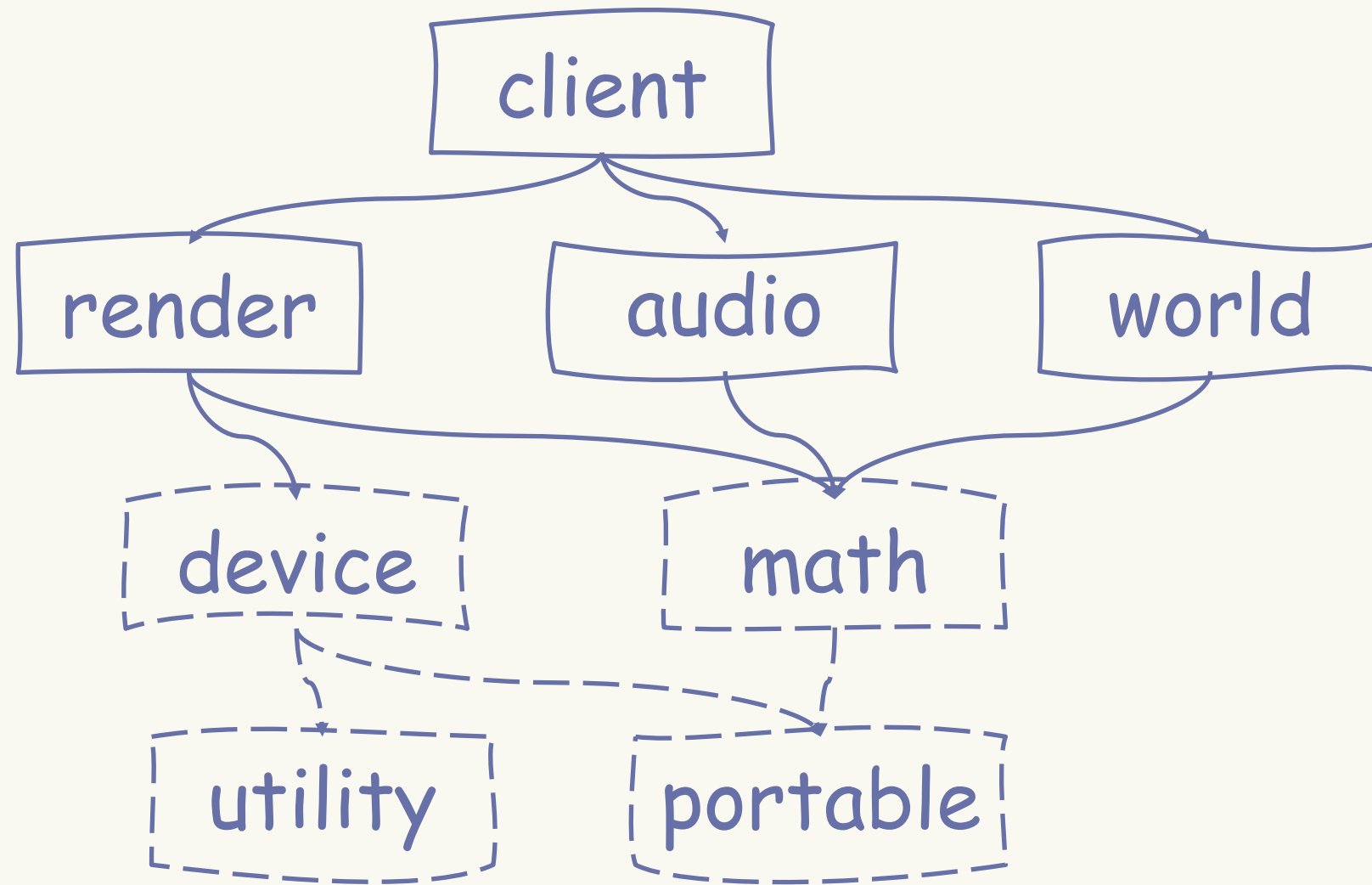
Migration



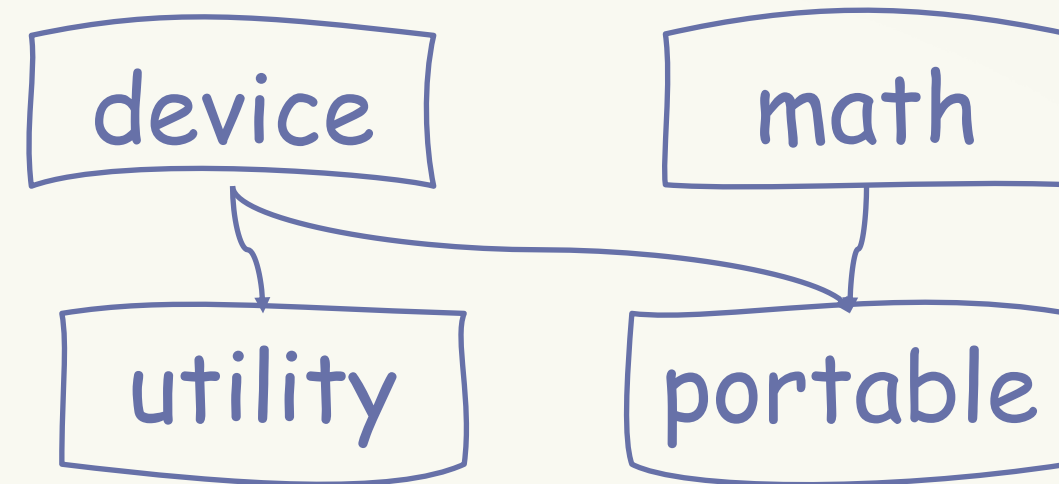
CMake based

Bazel

Migration

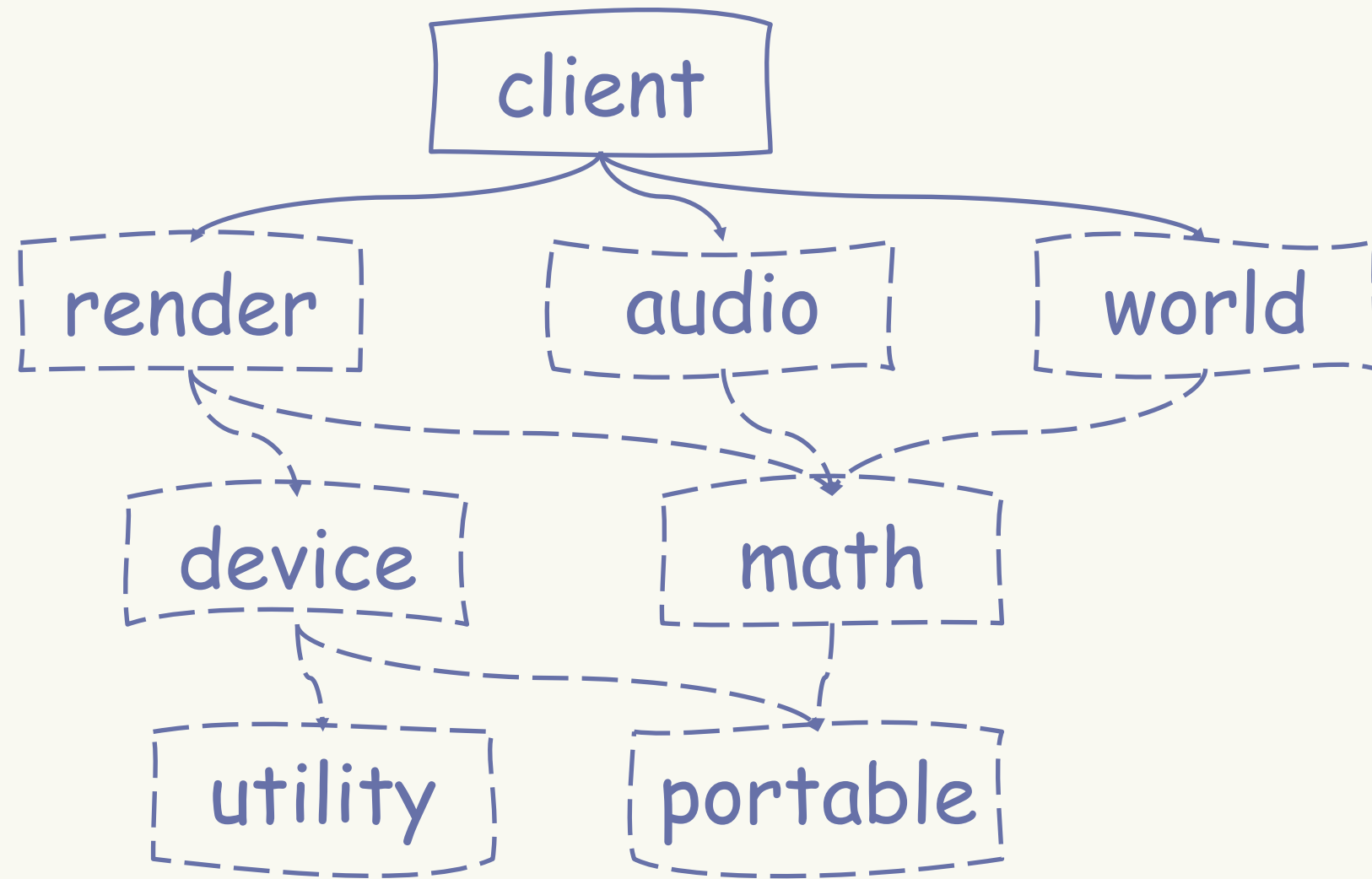


CMake based

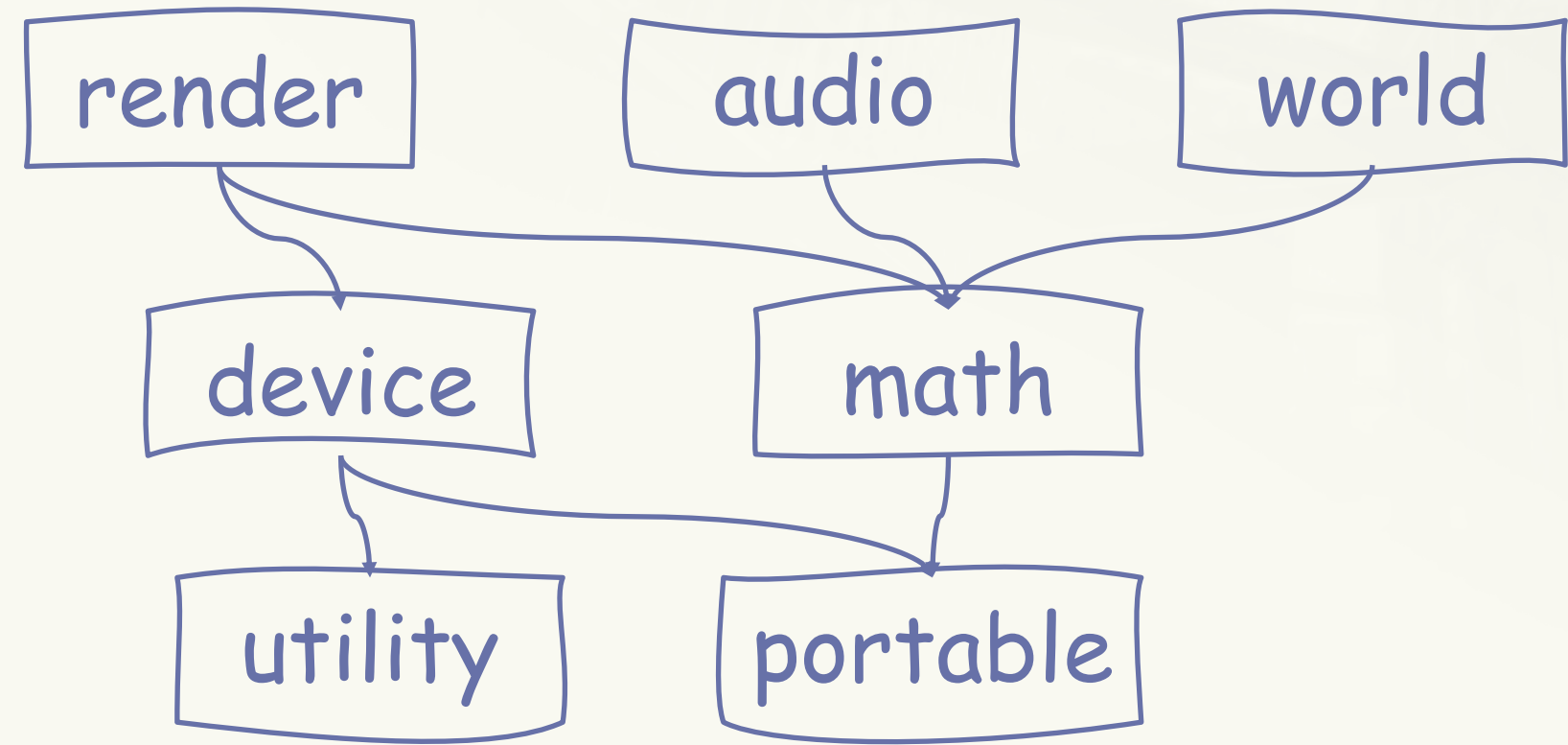


Bazel

Migration

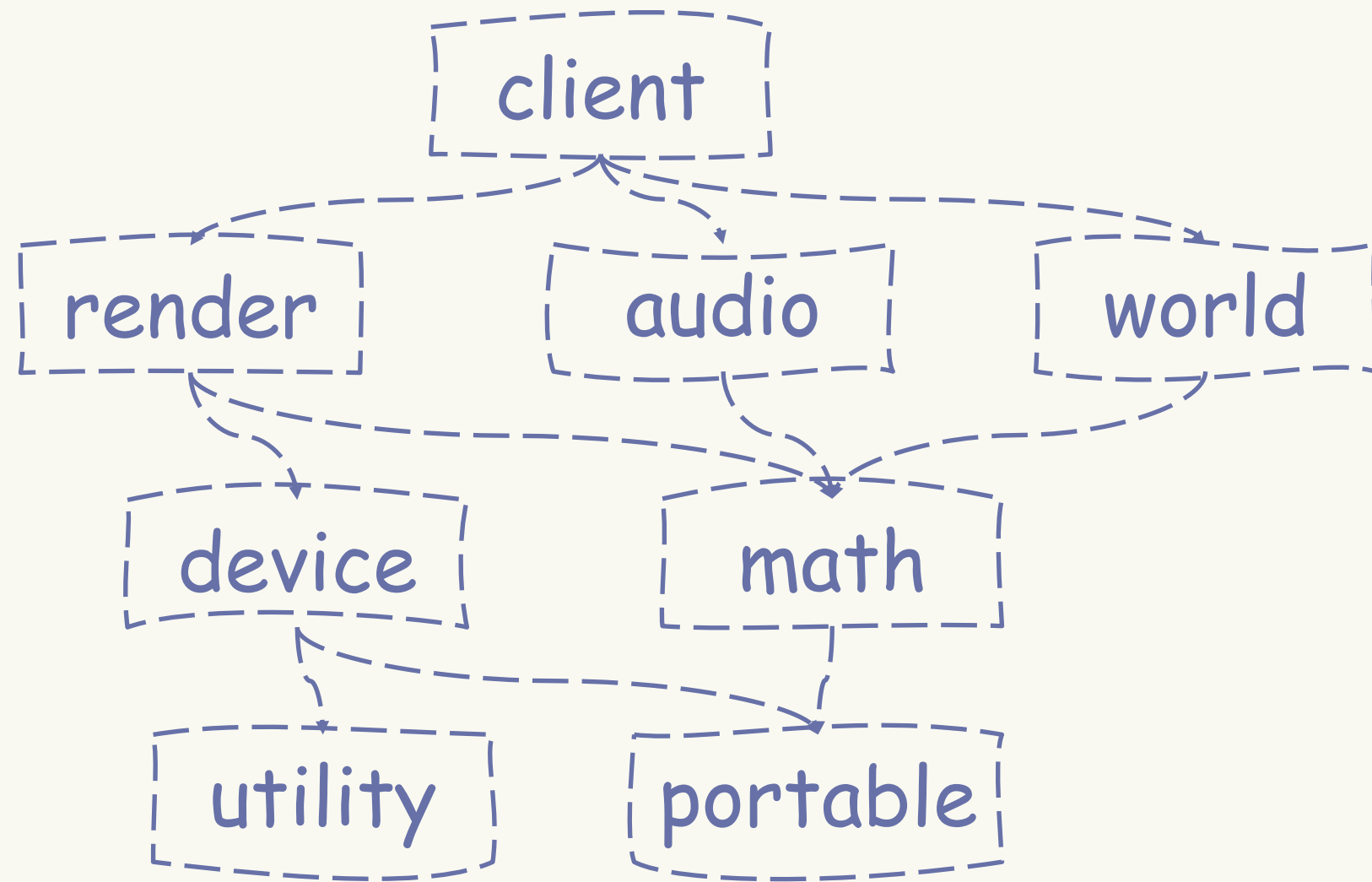


CMake based

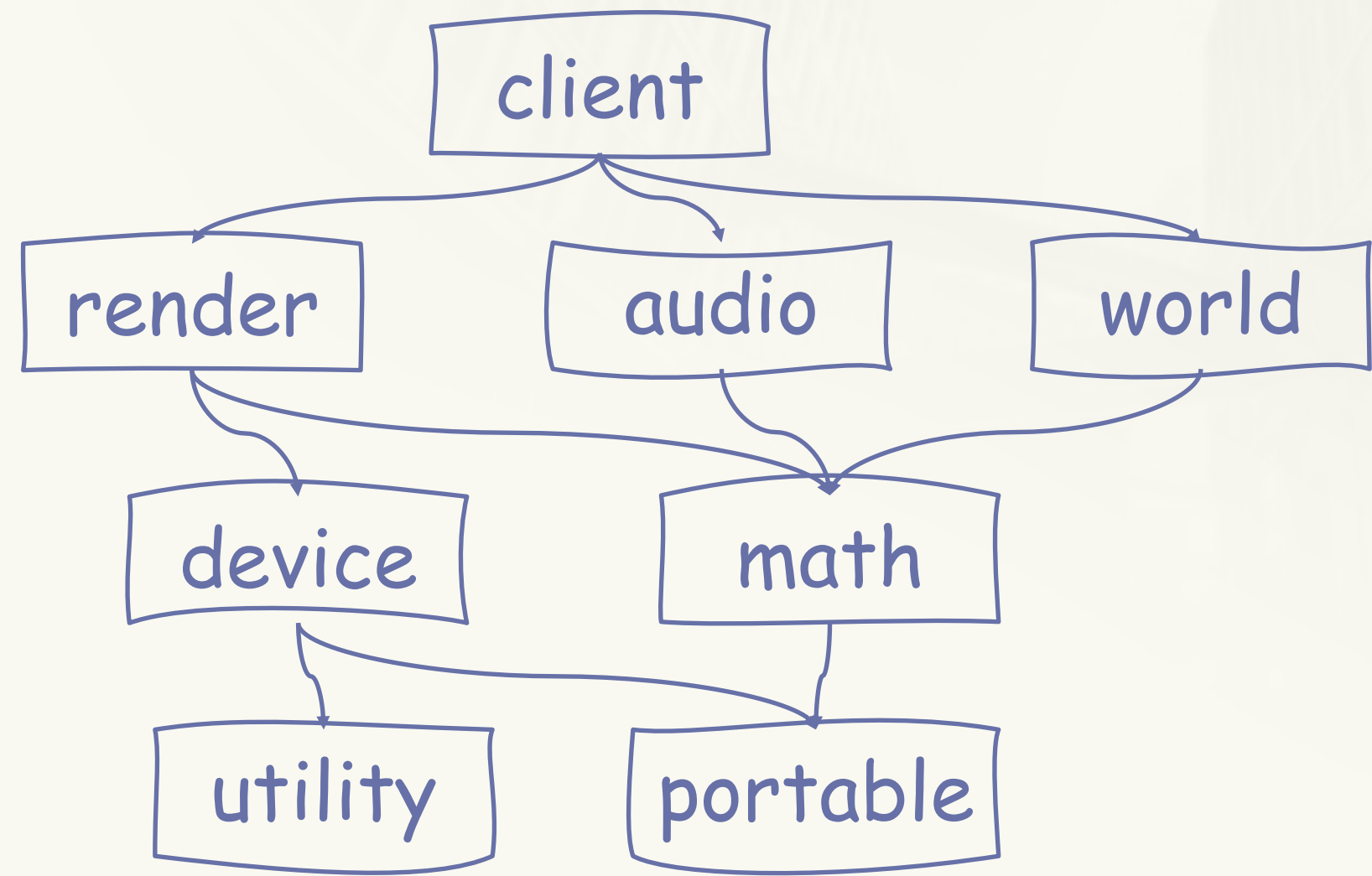


Bazel

Migration

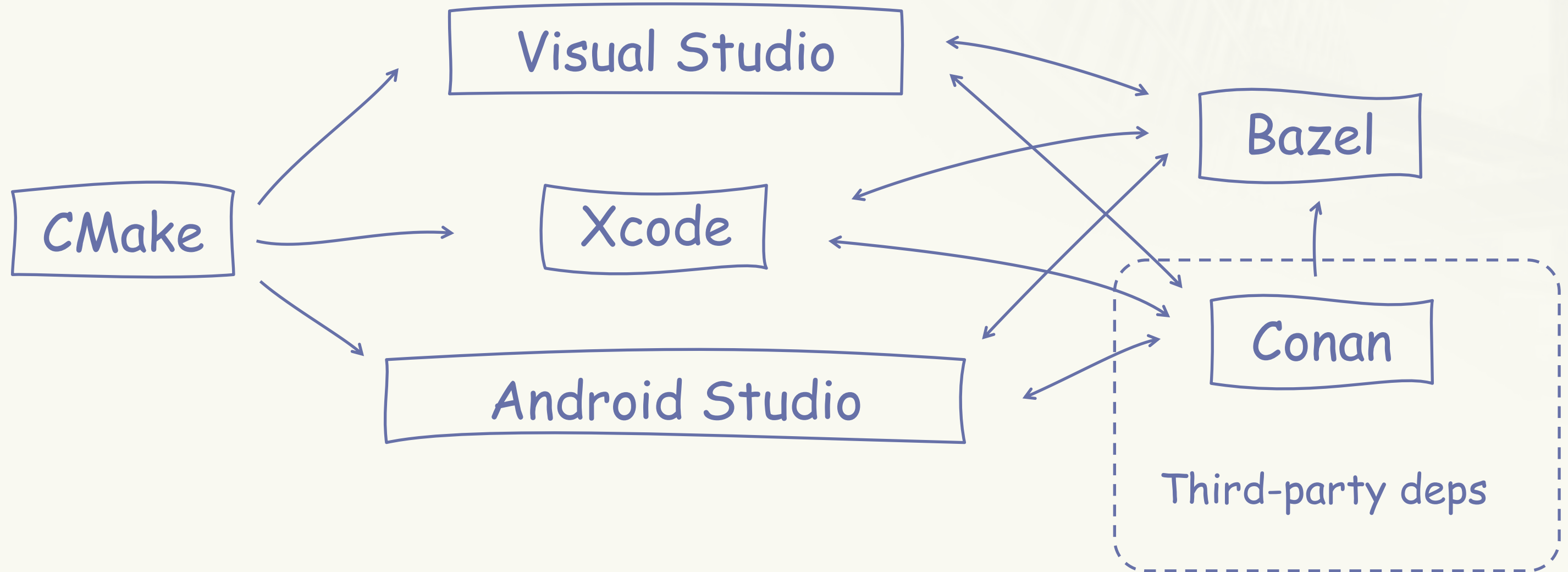


CMake based



Bazel

Migration



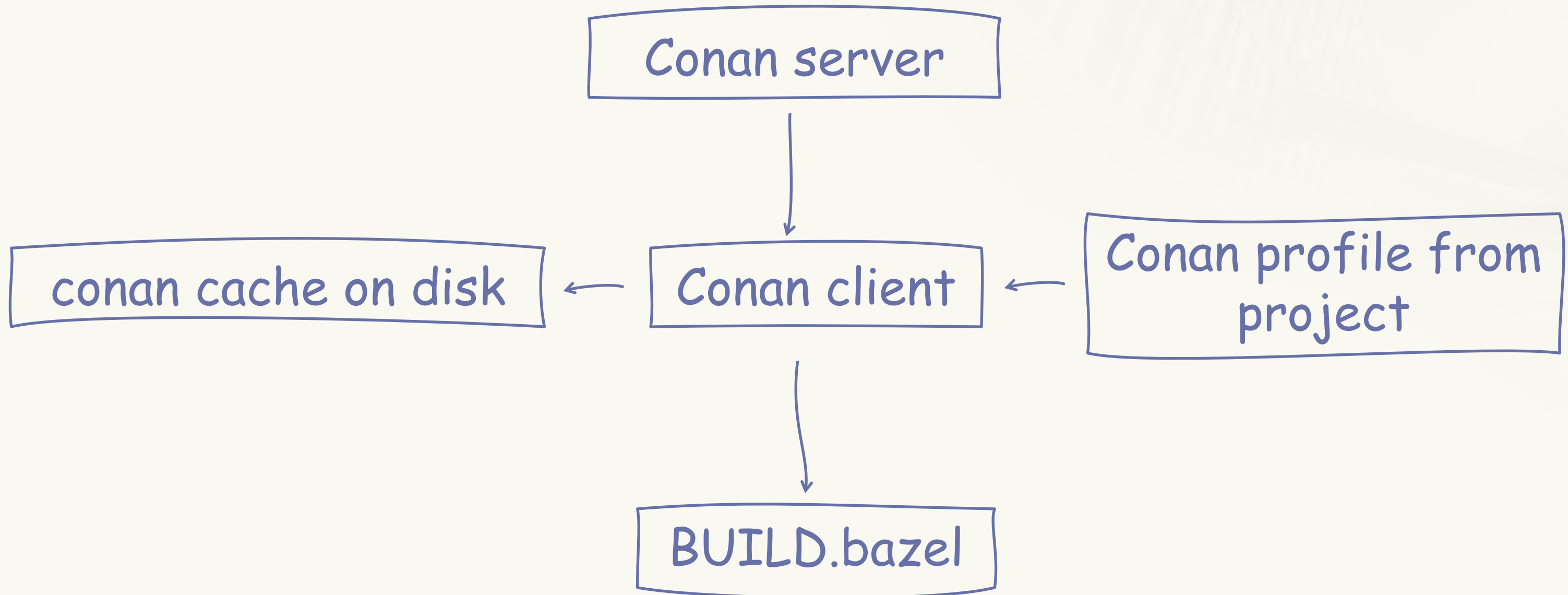
Conan Bazel Generator

https://docs.conan.io/1/integrations/build_system/bazel.html



```
1  cc_import(  
2      name = "libfoo",  
3      hdrs = ["libfoo.h"],  
4      static_library = "libfoo.a",  
5      shared_library = "libfoo.so",  
6  )
```


Conan Bazel Generator




Conan Bazel Generator

- Experimental
- Not work with some deps we have



Conan Bazel Generator



```
1  cc_import(  
2      name = "libfoo",  
3      hdrs = ["libfoo.h"],  
4      static_library = "liblibfoo.lib",  
5      interface_library = "libfoo.lib",  
6      shared_library = "libfoo.dll",  
7  )
```

<https://github.com/conan-io/conan/pull/11355>

<https://github.com/conan-io/conan/pull/11343>

Move to Bazel

- Build with Bazel
- Integrate with IDEs
- Remote cache and execution



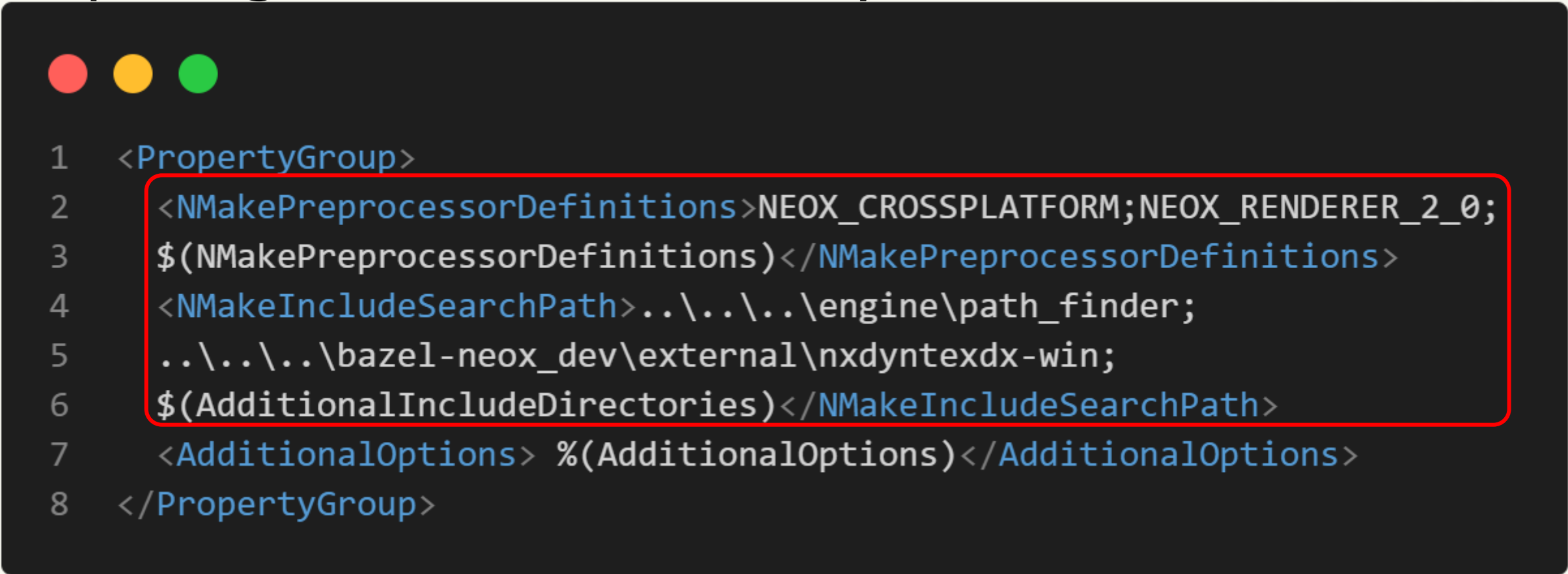
Visual Studio

- <https://github.com/tmandry/lavender>

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3   <Target Name="Build">
4     <Message Text="NMakePreprocessorDefinitions = $(NMakePreprocessorDefinitions)" />
5     <Exec Command="C:\bin\bazel.exe build $(BazelCfgOpts) //engine/client:client"
6           Outputs="client.exe"
7           WorkingDirectory="..\..\.." />
8   </Target>
9
10  <Target Name="Clean">
11    <Exec Command="C:\bin\bazel.exe clean $(BazelCfgOpts)"
12          WorkingDirectory="..\..\.." />
13  </Target>
14
15  <!-- Override PrepareForBuild so it doesn't try to create OutDir and interfere with bazel. -->
16  <Target Name="PrepareForBuild" DependsOnTargets="$(PrepareForBuildDependsOn)"></Target>
17 </Project>
```

Visual Studio

- <https://github.com/tmandry/lavender>



A screenshot of a Visual Studio code editor window. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The code is XML, and a red rectangular box highlights a section of it. The code is as follows:

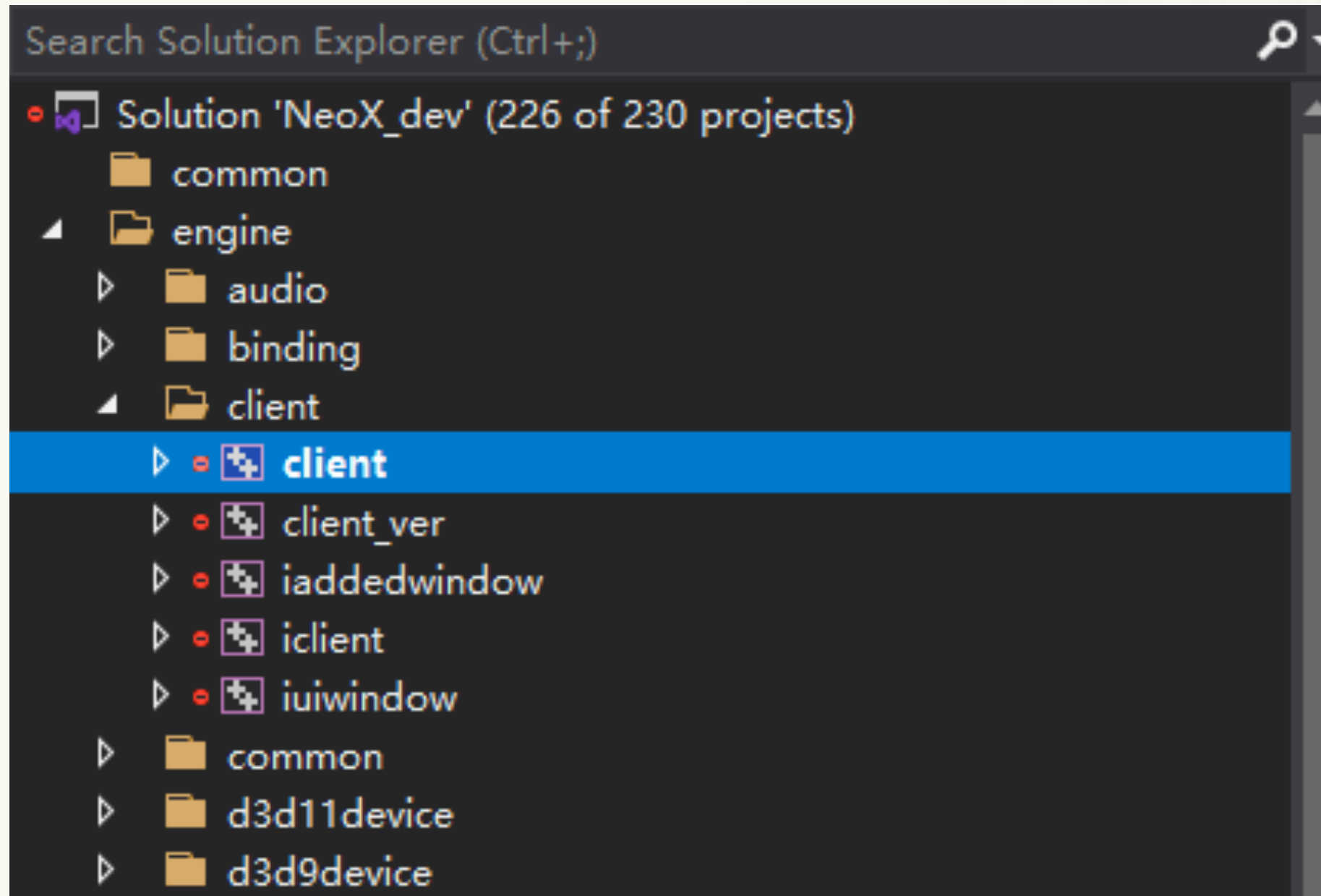
```
1 <PropertyGroup>
2   <NMakePreprocessorDefinitions>NEOX_CROSSPLATFORM;NEOX_RENDERER_2_0;
3   $(NMakePreprocessorDefinitions)</NMakePreprocessorDefinitions>
4   <NMakeIncludeSearchPath>..\..\..\engine\path_finder;
5   ..\..\..\bazel-neox_dev\external\nxdyntexdx-win;
6   $(AdditionalIncludeDirectories)</NMakeIncludeSearchPath>
7   <AdditionalOptions> %(AdditionalOptions)</AdditionalOptions>
8 </PropertyGroup>
```

Lavender

- No longer maintained
- Code intelligence broken in some cases
- <https://github.com/tmandry/lavender/issues/5>
- <https://github.com/kkpattern/lavender>



Lavender



Xcode

Tulsi

- <https://github.com/bazelbuild/tulsi>
- https://github.com/buildbuddy-io/rules_xcodeproj



Android Studio

Bazel Plugin

- <https://github.com/bazelbuild/intellij>



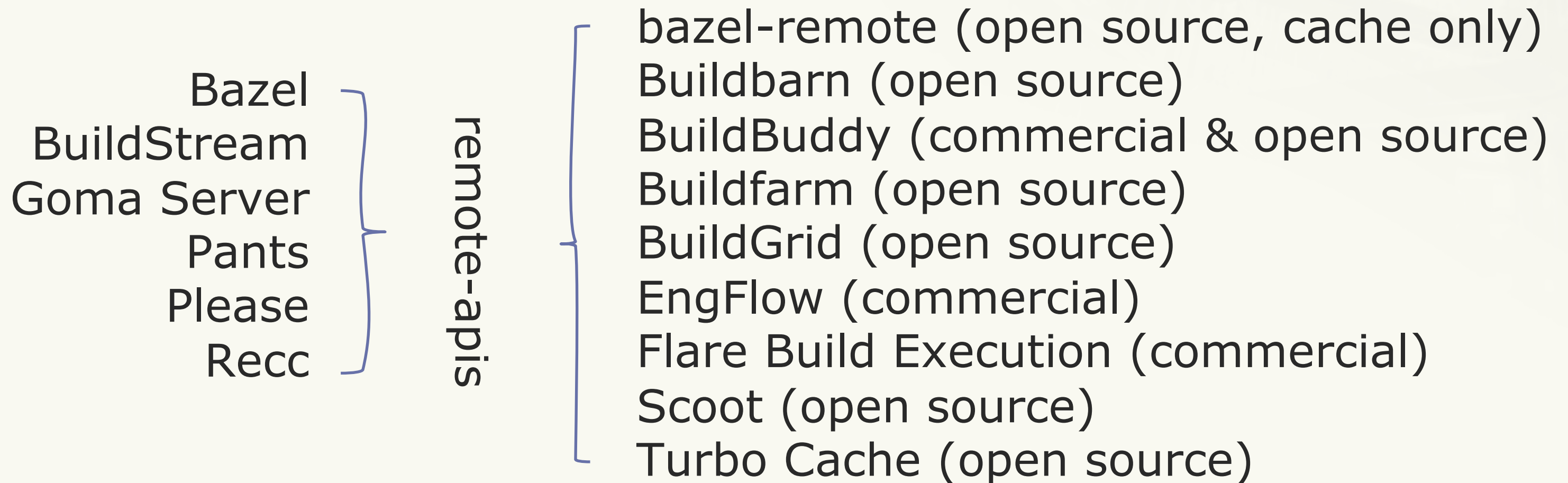
Move to Bazel

- Build with Bazel
- Integrate with IDEs
- Remote cache and execution



Remote API

- <https://github.com/bazelbuild/remote-apis>



Remote API



```
1 // The Remote Execution API is used to execute an
2 // [Action][build.bazel.remote.execution.v2.Action] on the remote
3 // workers.
4 //
5 // As with other services in the Remote Execution API, any call may return an
6 // error with a [RetryInfo][google.rpc.RetryInfo] error detail providing
7 // information about when the client should retry the request; clients SHOULD
8 // respect the information provided.
9 service Execution {
10     ...
11     rpc Execute(ExecuteRequest) returns (stream google.longrunning.Operation) {
12         option (google.api.http) = { post: "/v2/{instance_name=**}/actions:execute" body: "*" };
13     }
14
15     ...
16     rpc WaitExecution(WaitExecutionRequest) returns (stream google.longrunning.Operation) {
17         option (google.api.http) = { post: "/v2/{name=operations/**}:waitExecution" body: "*" };
18     }
19 }
```



Remote API



```
1 // The action cache API is used to query whether a given action has already been
2 // performed and, if so, retrieve its result. Unlike the
3 // [ContentAddressableStorage][build.bazel.remote.execution.v2.ContentAddressableStorage],
4 // which addresses blobs by their own content, the action cache addresses the
5 // [ActionResult][build.bazel.remote.execution.v2.ActionResult] by a
6 // digest of the encoded [Action][build.bazel.remote.execution.v2.Action]
7 // which produced them.
8 //
9 // The lifetime of entries in the action cache is implementation-specific, but
10 // the server SHOULD assume that more recently used entries are more likely to
11 // be used again.
12 //
13 // As with other services in the Remote Execution API, any call may return an
14 // error with a [RetryInfo][google.rpc.RetryInfo] error detail providing
15 // information about when the client should retry the request; clients SHOULD
16 // respect the information provided.
17 service ActionCache {
18     ...
19     rpc GetActionResult(GetActionResultRequest) returns (ActionResult) {
20         option (google.api.http) = { get: "/v2/{instance_name=**}/actionResults/{action_digest.hash}/{action_digest.size_bytes}" };
21     }
22     ...
23     rpc UpdateActionResult(UpdateActionResultRequest) returns (ActionResult) {
24         option (google.api.http) = { put: "/v2/{instance_name=**}/actionResults/{action_digest.hash}/{action_digest.size_bytes}" body: "action_result" };
25     }
26 }
27 }
```



Remote API



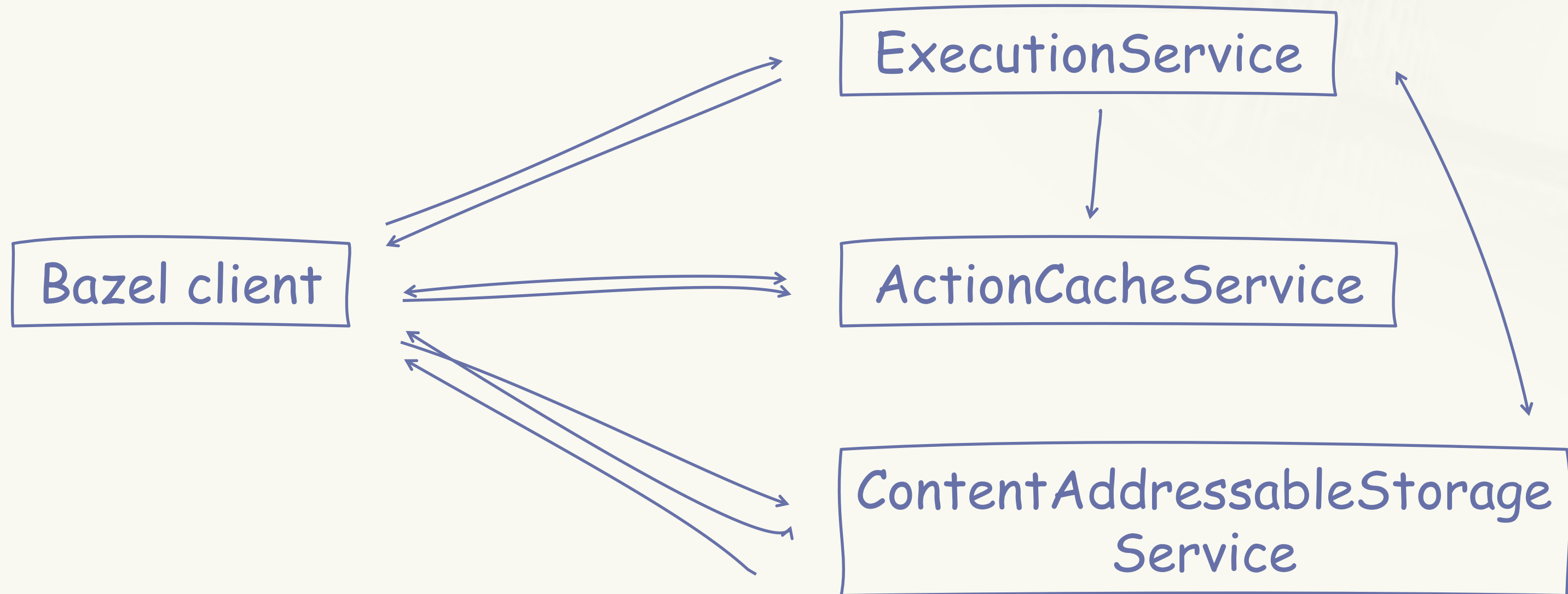
```
1 // uploaded under the hash as with any other piece of content. In practice, the
2 // input root directory is likely to refer to other Directories in its
3 // hierarchy, which must also each be uploaded on their own.
4 //
5 // ...
6 service ContentAddressableStorage {
7     ...
8     rpc FindMissingBlobs(FindMissingBlobsRequest) returns (FindMissingBlobsResponse) {
9         option (google.api.http) = { post: "/v2/{instance_name=**}/blobs:findMissing" body: "*" };
10    }
11
12    ...
13    rpc BatchUpdateBlobs(BatchUpdateBlobsRequest) returns (BatchUpdateBlobsResponse) {
14        option (google.api.http) = { post: "/v2/{instance_name=**}/blobs:batchUpdate" body: "*" };
15    }
16
17    ...
18    rpc BatchReadBlobs(BatchReadBlobsRequest) returns (BatchReadBlobsResponse) {
19        option (google.api.http) = { post: "/v2/{instance_name=**}/blobs:batchRead" body: "*" };
20    }
21
22    ...
23    rpc GetTree(GetTreeRequest) returns (stream GetTreeResponse) {
24        option (google.api.http) = { get: "/v2/{instance_name=**}/blobs/{root_digest.hash}/{root_digest.size_bytes}:getTree" };
25    }
26 }
```

Remote API



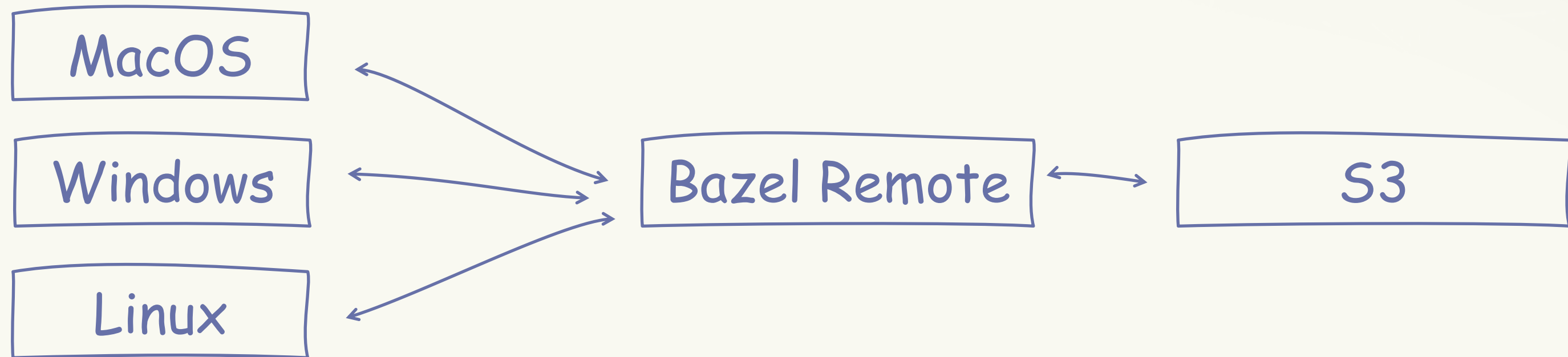
```
1 message Digest {  
2     // The hash. In the case of SHA-256, it will always be a lowercase hex string  
3     // exactly 64 characters long.  
4     string hash = 1;  
5  
6     // The size of the blob, in bytes.  
7     int64 size_bytes = 2;  
8 }
```


Remote API

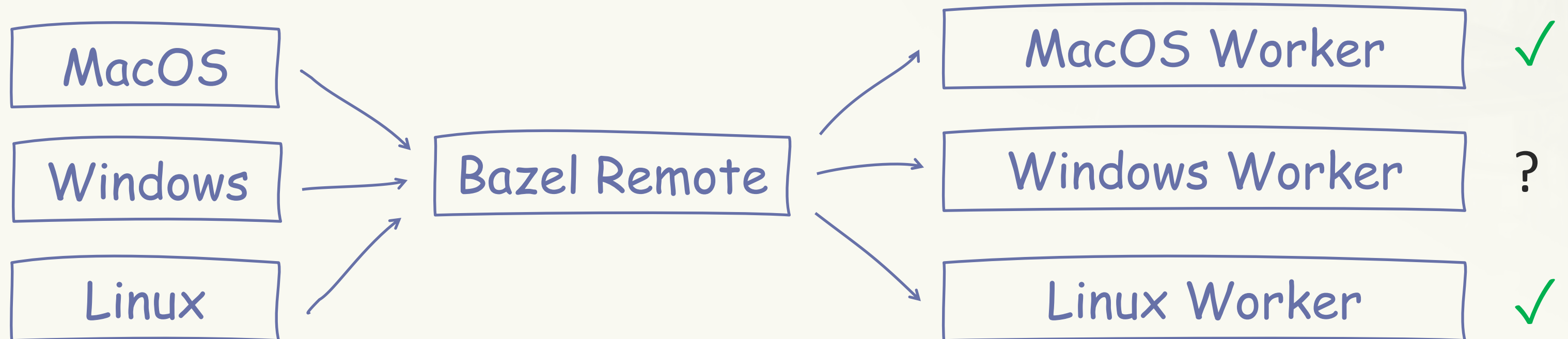


Bazel-Remote

- <https://github.com/buchgr/bazel-remote>
- Full clang-tidy run: 40 minutes -> 7 minutes



Remote Execution



Buildfarm

<https://github.com/bazelbuild/bazel-buildfarm>



Buildbarn

<https://github.com/buildbarn>

Virtual filesystem

Linux: FUSE

MacOS: NFSv4

Buildbarn

math

- matrix.h
- utility.h

renderer

- render.cpp



Buildbarn

math

- matrix.h
- utility.h

renderer

- render.cpp



Buildbarn

math

- matrix.h
- utility.h

renderer

- render.cpp



Buildbarn

math

- matrix.h
- utility.h

renderer

- render.cpp



Buildbarn

<https://github.com/buildbarn>

Virtual filesystem

Linux: FUSE

MacOS: NFSv4

Windows: ProjFS ❌

Buildbarn Worker

210 lines of protobuf code

1 service

1 rpc call

4 messages

```
1 message SynchronizeResponse {
2     // The timestamp at which the worker is supposed to synchronize again.
3     // The worker is permitted to contact the scheduler earlier if any
4     // notable events occur (e.g., execution of an action completes).
5     //
6     // If the worker does not synchronize with the scheduler at around
7     // this time, the scheduler may assume the worker has gone offline.
8     // To compensate for clock inaccuracies, network latency and transient
9     // network failures, the scheduler should tolerate a certain amount of
10    // skew. Schedulers may vary the interval returned to tune the amount
11    // of load received.
12    google.protobuf.Timestamp next_synchronization_at = 1;
13
14    // The state to which the scheduler desires the worker to transition.
15    // This value must be left unset in case the scheduler desires the
16    // worker to remain in its current state. For example, when it wants
17    // the worker to continue executing the currently running build
18    // action.
19    DesiredState desired_state = 2;
20 }
```

Our Buildbarn Worker

Implemented in Python

Shared input dirs

2480 lines of code

<https://github.com/kkpattern/bb-remote-execution-py>

Debug

```
1  bazel-bin/engine/common/libc_common.a(NXDispatcher.o):  file format Mach-O arm64
2
3  .debug_info contents:
4  0x00000000: Compile Unit: length = 0x000239fa, format = DWARF32, version = 0x0004, abbr_offset = 0x0000, addr_size = 0x08 (next unit at 0x000239fe)
5
6  0x00000032:  DW_TAG_class_type
7                DW_AT_calling_convention      (DW_CC_pass_by_reference)
8                DW_AT_name                    ("NXDispatcher")
9                DW_AT_byte_size               (0x0240)
10               DW_AT_decl_file                ("/private/var/tmp/.../sandbox/darwin-sandbox/935/execroot/neox/./engine/common/NXDispatcher.h")
11               DW_AT_decl_line                (333)
```

Debug

- fdebug-prefix-map=\${EXECROOT} = .

```
1  bazel-bin/engine/common/libc_common.a(NXDispatcher.o):  file format Mach-O arm64
2
3  .debug_info contents:
4  0x00000000: Compile Unit: length = 0x000239fa, format = DWARF32, version = 0x0004, abbr_offset = 0x0000, addr_size = 0x08 (next unit at 0x000239fe)
5
6  0x00000032:   DW_TAG_class_type
7                DW_AT_calling_convention      (DW_CC_pass_by_reference)
8                DW_AT_name                    ("NXDispatcher")
9                DW_AT_byte_size               (0x0240)
10               DW_AT_decl_file                ("../engine/common/NXDispatcher.h")
11               DW_AT_decl_line                (333)
```

Clang-Tidy

“clang-tidy is a clang-based C++ "linter" tool. Its purpose is to provide an extensible framework for diagnosing and fixing typical programming errors, like style violations, interface misuse, or bugs that can be deduced via static analysis. ”

```
1  ---
2  MainSourceFile: 'engine/render/postprocess.cpp'
3  Diagnostics:
4    - DiagnosticName: clang-analyzer-deadcode.DeadStores
5      DiagnosticMessage:
6        Message: 'Value stored to 'pass_count' during its initialization is never read'
7        FilePath: 'engine/render/postprocess.cpp'
8        FileOffset: 123681
9        Replacements: []
10       Ranges:
11         - FilePath: 'engine/render/postprocess.cpp'
12           FileOffset: 123681
13           Length: 10
14         - FilePath: 'engine/render/postprocess.cpp'
15           FileOffset: 123694
16           Length: 27
17       Notes:
18         - Message: 'Value stored to 'pass_count' during its initialization is never read'
19           FilePath: 'engine/render/postprocess.cpp'
20           FileOffset: 123681
21           Replacements: []
22           Ranges:
23             - FilePath: 'engine/render/postprocess.cpp'
24               FileOffset: 123681
25               Length: 10
26             - FilePath: 'engine/render/postprocess.cpp'
27               FileOffset: 123694
28               Length: 27
29       Level: Warning
30       BuildDirectory: ''
```




```
1 /NeoX/engine/metaldevice/metalrenderpass.mm:392:27: warning: Called C++ object pointer is null [clang-analyzer-core.CallAndMessage]
2     texture = depth->GetNative();
3             ^
4 ...
5 /NeoX/engine/metaldevice/metalrenderpass.mm:343:13: note: Assuming 'depth' is equal to NULL
6     if (depth != NULL)
7         ^
8 /NeoX/engine/metaldevice/metalrenderpass.mm:343:9: note: Taking false branch
9     if (depth != NULL)
10        ^
11 /NeoX/engine/metaldevice/metalrenderpass.mm:374:13: note: Assuming 'stencil' is not equal to NULL
12     if (stencil != NULL)
13         ^
14 /NeoX/engine/metaldevice/metalrenderpass.mm:374:9: note: Taking true branch
15     if (stencil != NULL)
16         ^
17 ...
18 /NeoX/engine/metaldevice/metalrenderpass.mm:392:27: note: Called C++ object pointer is null
19     texture = depth->GetNative();
20             ^
21 Suppressed 1 warnings (1 in non-user code).
22 Use -header-filter=.* to display errors from all non-system headers. Use -system-headers to display errors from system headers as well.
```



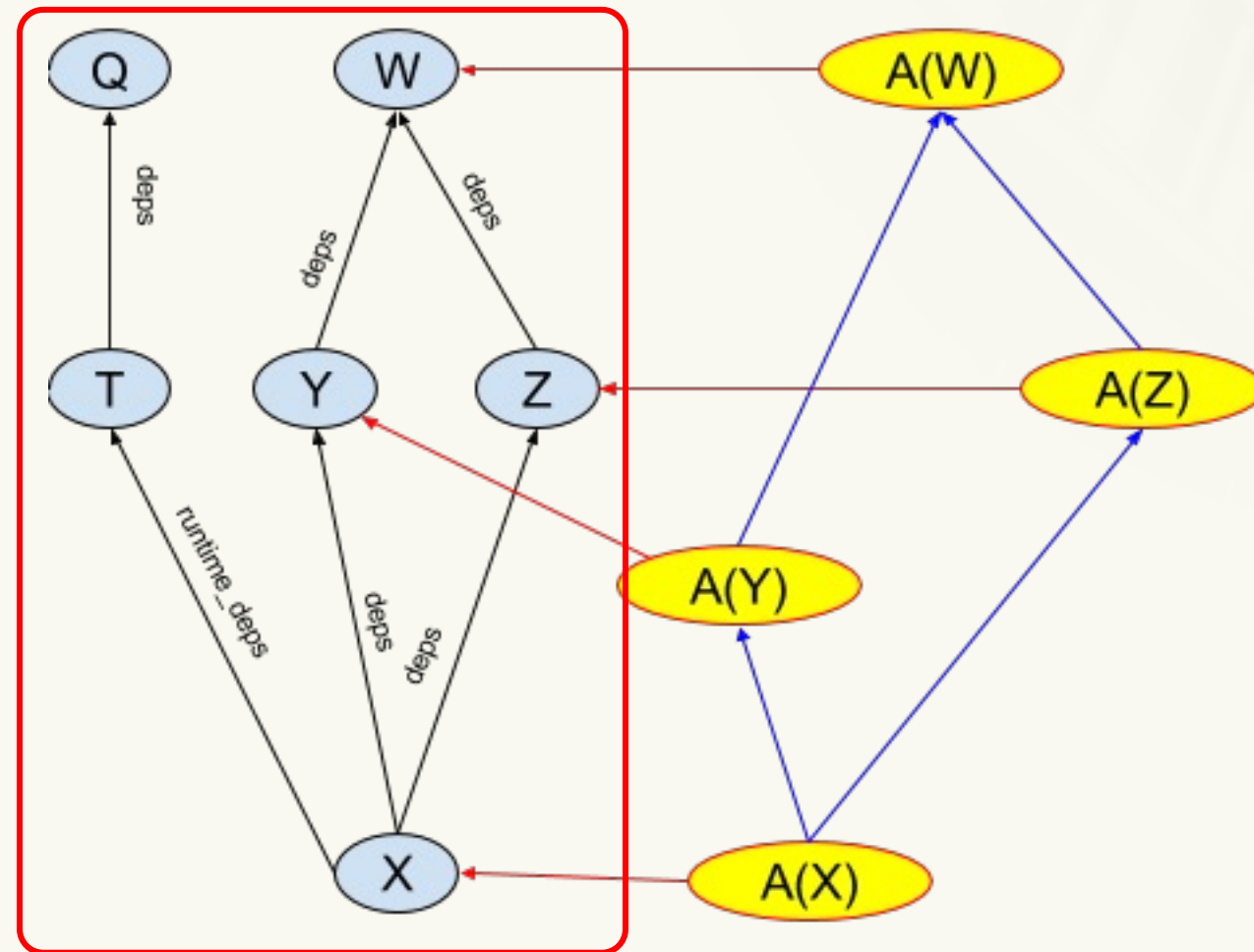
```
1  if (depth != NULL)
2  {
3      ...
4  }
5
6  if (stencil != NULL)
7  {
8      id<MTLTexture> texture = nil;
9      id<MTLTexture> textureResolve = nil;
10
11     if (sampleCount > 1)
12     {
13         ...
14     }
15     else
16     {
17         texture = depth->GetNative();
18         ...
19     }
20 }
```

Clang-Tidy

40 minutes on NeoX code base
No incremental checking



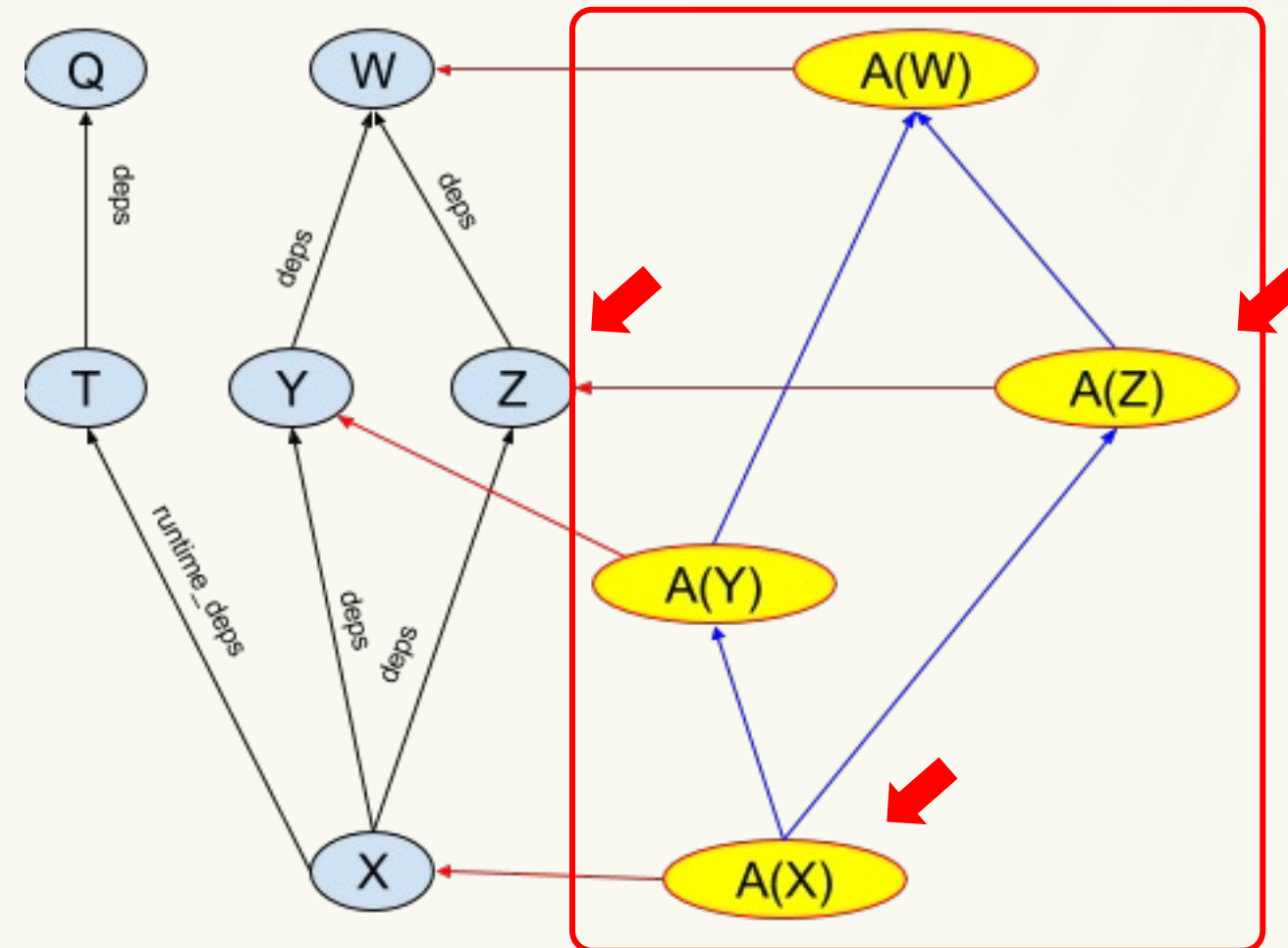
Bazel Aspects



From <https://bazel.build/extending/aspects?hl=en>

`bazel build //:X`

Bazel Aspects



From <https://bazel.build/extending/aspects?hl=en>

```
bazel build //:X --aspects clang_tidy.bzl%tidy_aspect
```


Bazel Clang Tidy

```
1 def tidy_aspects(  
2     ctx,  
3     infile,  
4     discriminator):  
5     inputs = get_inputs(ctx)  
6  
7     outfile = ctx.actions.declare_file("bazel_clang_tidy_" + infile.path + "." + discriminator + ".clang-tidy.yaml",)  
8  
9     args = ctx.actions.args()  
10  
11     args.add(outfile.path)  
12     # add source to check  
13     args.add(infile.path)  
14     # start args passed to the compiler  
15     args.add("--")  
16     # add all compile flags  
17     args.add_all(get_compile_flags(ctx))  
18  
19     ctx.actions.run(inputs = inputs,  
20                     outputs = [outfile],  
21                     executable = "clang-tidy",  
22                     arguments = [args])  
23     return outfile
```

Bazel Clang Tidy

https://github.com/erenon/bazel_clang_tidy

https://github.com/kkpattern/bazel_clang_tidy



```
1 /NeoX/engine/metaldevice/metalrenderpass.mm:392:27: warning: Called C++ object pointer is null [clang-analyzer-core.CallAndMessage]
2     texture = depth->GetNative();
3             ^
4 ...
5 /NeoX/engine/metaldevice/metalrenderpass.mm:343:13: note: Assuming 'depth' is equal to NULL
6     if (depth != NULL)
7         ^
8 /NeoX/engine/metaldevice/metalrenderpass.mm:343:9: note: Taking false branch
9     if (depth != NULL)
10        ^
11 /NeoX/engine/metaldevice/metalrenderpass.mm:374:13: note: Assuming 'stencil' is not equal to NULL
12     if (stencil != NULL)
13         ^
14 /NeoX/engine/metaldevice/metalrenderpass.mm:374:9: note: Taking true branch
15     if (stencil != NULL)
16         ^
17 ...
18 /NeoX/engine/metaldevice/metalrenderpass.mm:392:27: note: Called C++ object pointer is null
19     texture = depth->GetNative();
20             ^
21 Suppressed 1 warnings (1 in non-user code).
22 Use -header-filter=.* to display errors from all non-system headers. Use -system-headers to display errors from system headers as well.
```

```
1 diff --git a/engine/metaldevice/metalrenderpass.mm b/engine/metaldevice/metalrenderpass.mm
2 index a1d991f39c..e950270a64 100644
3 --- a/engine/metaldevice/metalrenderpass.mm
4 +++ b/engine/metaldevice/metalrenderpass.mm
5 @@ -466,5 +466,24 @@ namespace device
6
7         return true;
8     }
9 +
10 + void MetalRenderPass::CheckSize()
11 + {
12 +     id<MTLTexture> tex = nil;
13 +     tex = mMTLPassDesc.colorAttachments[0].texture;
14 +     if (tex == nil)
15 +         tex = mMTLPassDesc.depthAttachment.texture;
16 +     if (tex == nil)
17 +         tex = mMTLPassDesc.stencilAttachment.texture;
18 +     if (tex != nil)
19 +     {
20 +         mWidth = tex.width;
21 +         mHeight = tex.height;
22 +     }
23 +     else
24 +     {
25 +         mWidth = mHeight = 0;
26 +     }
27 + }
28 }
29 }
```



engine/metaldevice/metalrenderpass.cpp

Called C++ object pointer is null

Line:343

Line:374

Line:392

diff engine/metaldevice/metalrenderpass.cpp

Added lines: 470 - 487

engine/metaldevice/metalrenderpass.cpp ✓

Whatthepatch

<https://github.com/cscorley/whatthepatch>



Break Large Library

libworld.a

- 451 header files
- 375 source files
- 4 minutes to build locally
- Many unused includes

Include-What-You-Use

<https://github.com/include-what-you-use/include-what-you-use>

Remove unused includes

Add missing includes

Only check non-legacy files in CI

Compilation Database

```
1  [
2  {
3    "file": "engine/client/crossplatform/client2listener.cpp",
4    "arguments": [
5      "/Applications/Xcode-14.1.0.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/clang",
6      "-O3",
7      "-std=c++14",
8      "-DNEOX_STATIC_LIB",
9      "-isysroot",
10     "/Applications/Xcode-14.1.0.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS.sdk",
11     "-target",
12     "arm64-apple-ios",
13     "-c",
14     "engine/client/crossplatform/client2listener.cpp",
15     "-o",
16     "bazel-out/ios_arm64-opt/bin/engine/client/_objs/client_ios/arc/client2listener.o",
17     "-Werror",
18     ...
19   ],
20   "directory": "/path/to/NeoX"
21 }
22 ...
23 ]
```

<https://github.com/hedronvision/bazel-compile-commands-extractor>

After

- 1 target to 25 targets
- Local build time: 4 minutes -> <1 ~ 3 minutes



More Tools

<https://github.com/llvm/llvm-project>

<https://github.com/kubernetes/kubernetes>

<https://github.com/jenkinsci/jenkins>

<https://github.com/tiangolo/fastapi>

Working with community

- Ask questions
- Question will be answered later
- Not to skip a tool if it's not perfect
- Always contribute back

Main Takeaways

- Open source tools are powerful
- Sometimes essential
- Involve with the community

Thanks

EVE Echoes team



Thanks

EVE Echoes team

NeoX team

Open source community



Q & A

 @kkpattern

 kylerzhang11@gmail.com

 <https://github.com/kkpattern/lavender>

 https://github.com/kkpattern/bazel_clang_tidy

 <https://github.com/kkpattern/bb-remote-execution-py>