TWEAG
by Modus Create

EXTENDING BUCK2

Andreas Herrmann

London Build Meetup — May 22nd, 2025

USE CASE

USE CASE

- Haskell & TypeScript

USE CASE

- Haskell & TypeScript
- Linux & MacOS

USE CASE

- Haskell & TypeScript
- Linux & MacOS
- >1M LoC Haskell

USE CASE

- Haskell & TypeScript
- Linux & MacOS
- >1M LoC Haskell
- >10k modules

USE CASE

- Haskell & TypeScript
- Linux & MacOS
- >1M LoC Haskell
- >10k modules
- Nix integration

MOTIVATION

HASKELL RULES

MercuryTechnologies/buck2-prelude

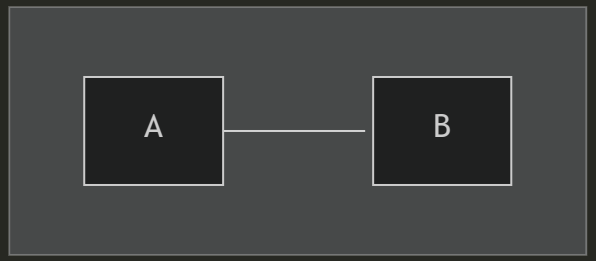
DYNAMIC DEPENDENCIES

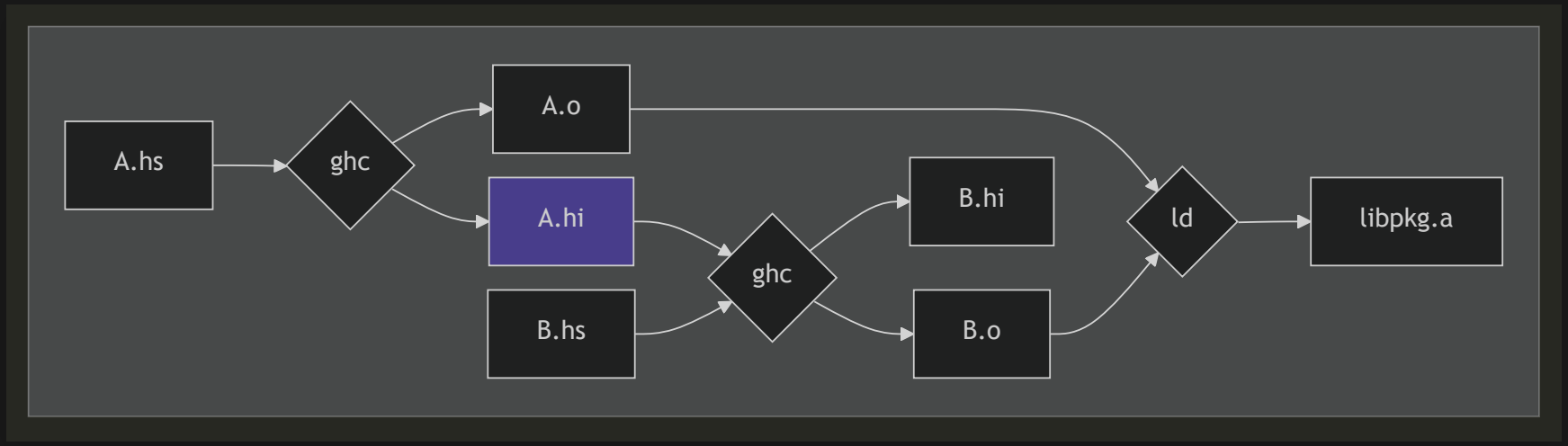
DYNAMIC DEPENDENCIES

```haskell
-- B.hs
import A
```

DYNAMIC DEPENDENCIES

```haskell
-- B.hs
import A
```

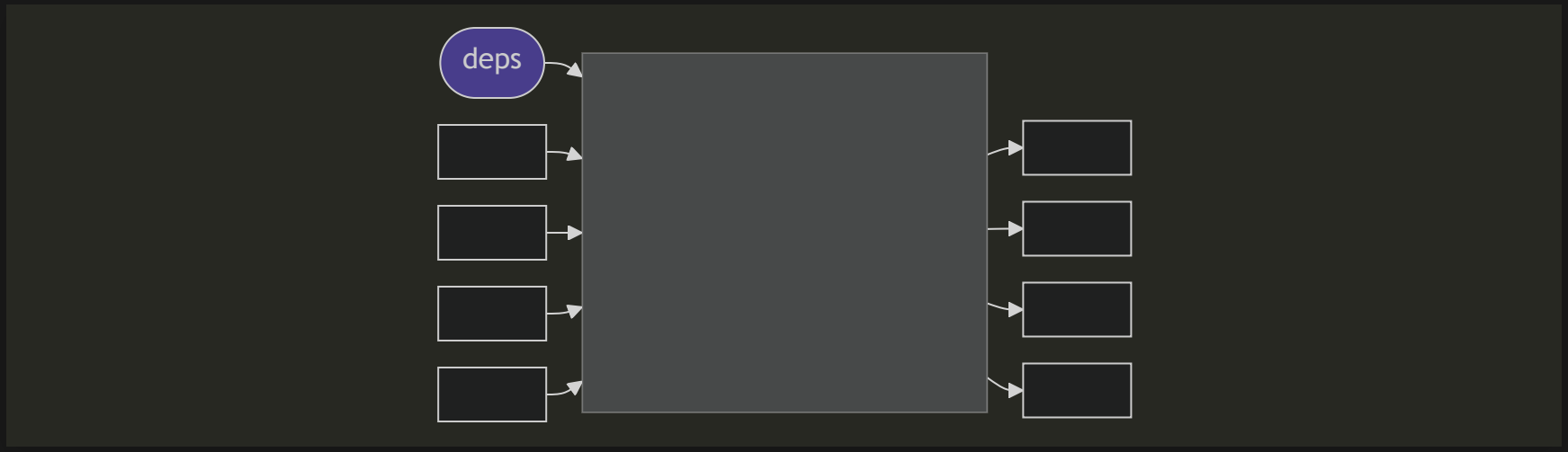


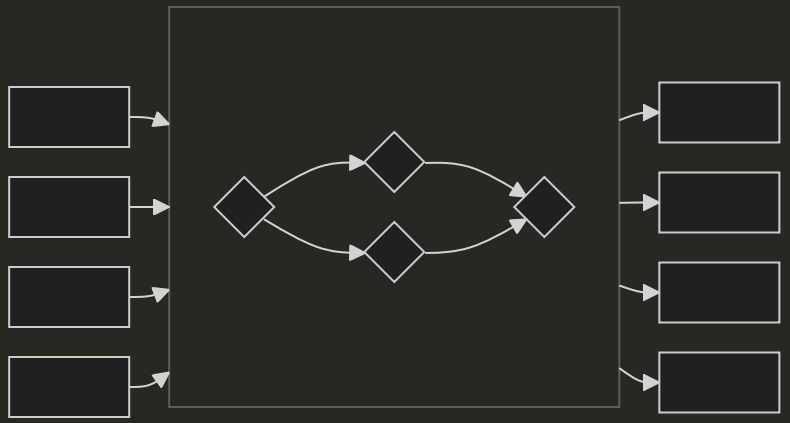

A.hs
ghc
A.o
A.hi
B.hs
ghc
B.hi
B.o
ld
libpkg.a

```
hs_module(name = "A", src = "A.hs")
hs_module(name = "B", src = "B.hs", deps = [":A"])
hs_library(name = "pkg", modules = [":A", ":B"])
```

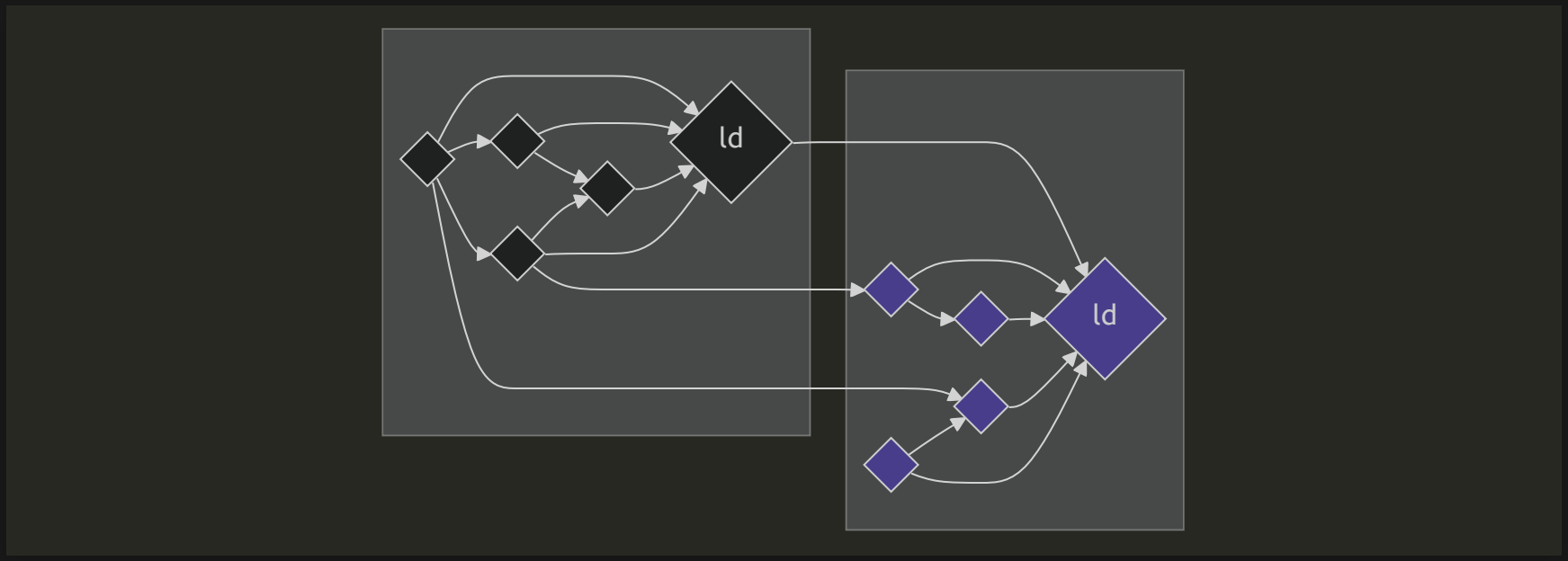
```
hs_library(
  name = "pkg",
  srcs = ["A.hs", "B.hs"],
)
```

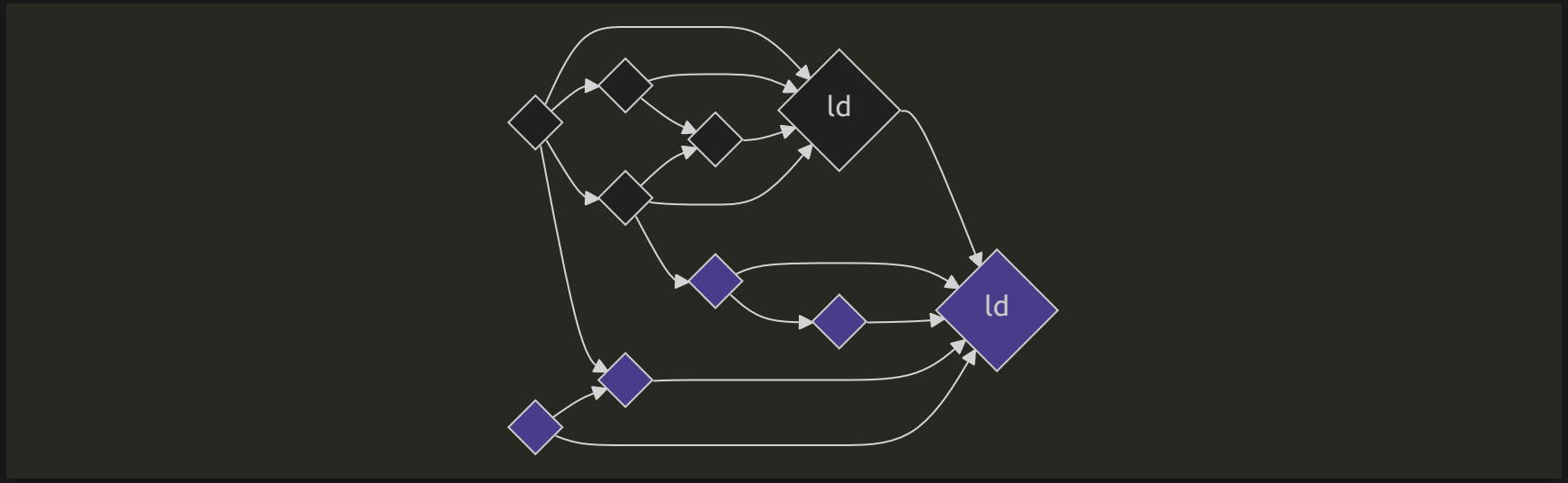
DYNAMIC ACTIONS


deps



PACKAGE BOUNDARIES


ld
ld


ld
ld

DYNAMIC VALUES

Pass data between dynamic actions Buck2#619

```
compile = dynamic_actions(impl = dyn_impl, attrs = {
  "dynval": dynattrs.dynamic_value(),
  # ...
})

def dyn_impl(actions, dynval):
  info = dynval.providers[DynDepInfo]
  # ...

def rule_impl(ctx):
  # ...
  ctx.actions.dynamic_output_new(compile(...))
  # ...
```

DYNAMIC VALUES

Pass data between dynamic actions Buck2#619

```
compile = dynamic_actions(impl = dyn_impl, attrs = {
  "dynval": dynattrs.dynamic_value(),
  # ...
})

def dyn_impl(actions, dynval):
  info = dynval.providers[DynDepInfo]
  # ...

def rule_impl(ctx):
  # ...
  ctx.actions.dynamic_output_new(compile(...))
  # ...
```

DYNAMIC VALUES

Pass data between dynamic actions Buck2#619

```
compile = dynamic_actions(impl = dyn_impl, attrs = {
  "dynval": dynattrs.dynamic_value(),
  # ...
})

def dyn_impl(actions, dynval):
  info = dynval.providers[DynDepInfo]
  # ...

def rule_impl(ctx):
  # ...
  ctx.actions.dynamic_output_new(compile(...))
  # ...
```

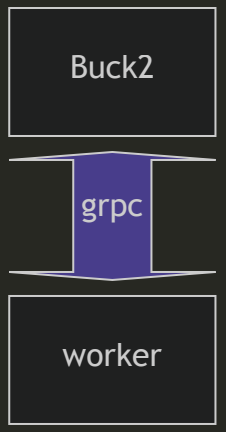
DYNAMIC VALUES

Pass data between dynamic actions Buck2#619

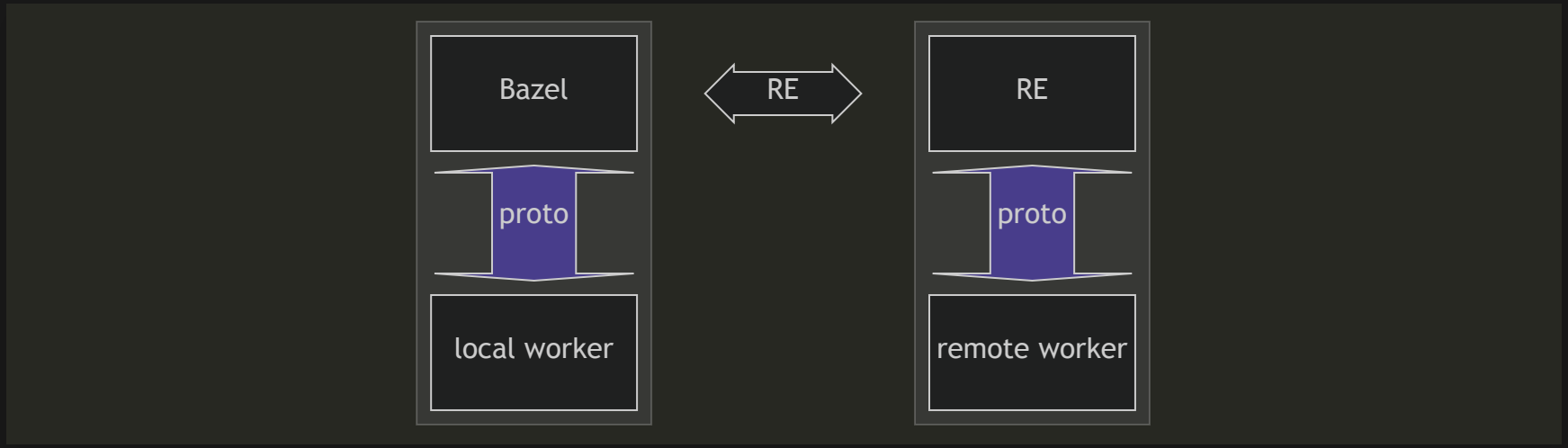
```
compile = dynamic_actions(impl = dyn_impl, attrs = {
  "dynval": dynattrs.dynamic_value(),
  # ...
})

def dyn_impl(actions, dynval):
  info = dynval.providers[DynDepInfo]
  # ...

def rule_impl(ctx):
  # ...
  ctx.actions.dynamic_output_new(compile(...))
  # ...
```

REMOTE PERSISTENT WORKER

BUCK2 PERSISTENT WORKER

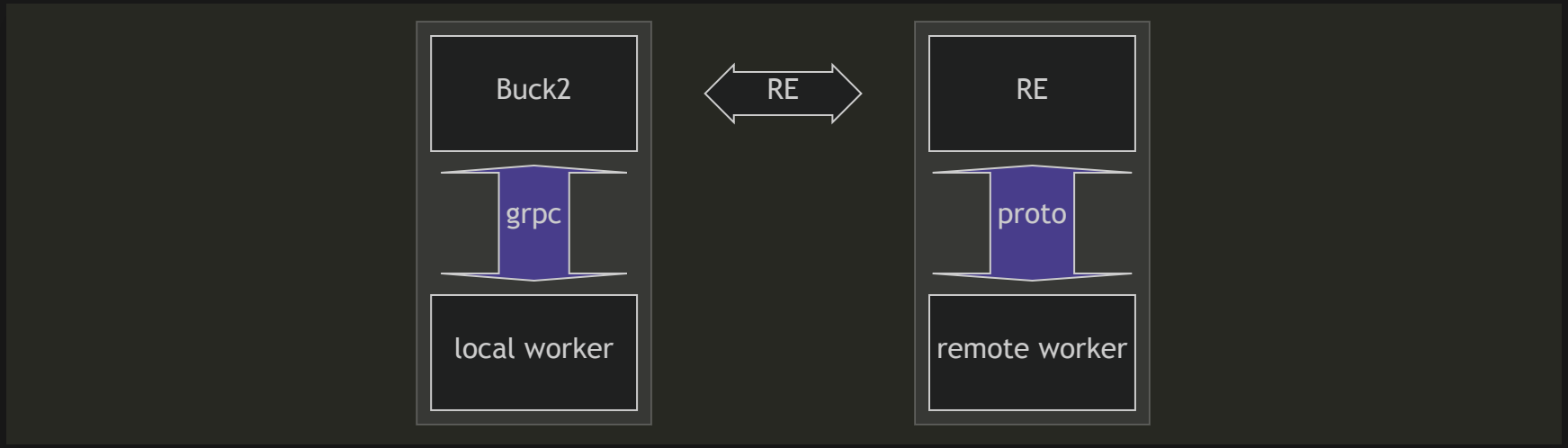


BAZEL PERSISTENT WORKER



Bazel proposal 2021-03-06-remote-persistent-workers

REMOTE PERSISTENT WORKER



Buck2 implementation Buck2#787

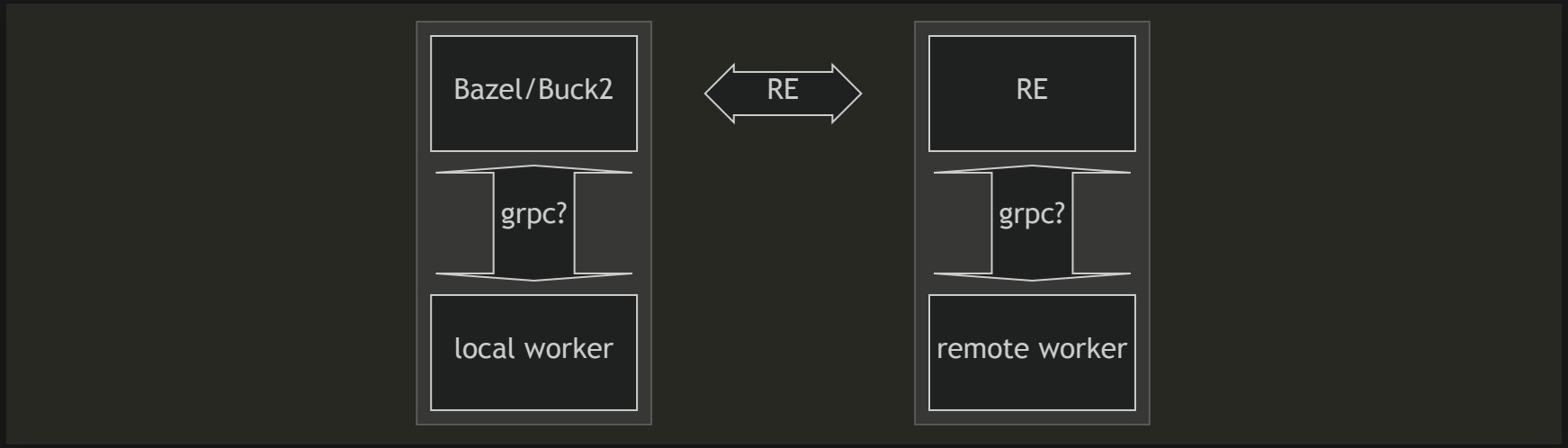
```
def _worker_impl(ctx):
  return [..., WorkerInfo(
    ...
    supports_bazel_remote_persistent_worker_protocol = True,
  )]
```

```
def _worker_impl(ctx):
  return [..., WorkerInfo(
    ...
    supports_bazel_remote_persistent_worker_protocol = True,
  )]
```

```
def main():
  if socket_path := os.getenv("WORKER_SOCKET"):
    # Buck2 worker (grpc)
  elif "--persistent_worker" in sys.arv:
    # Bazel worker (proto stdin/out)
  else:
    # local command
```

Full example in Buck2#787

FUTURE?



Discussion at Buck2#776

BUILD EVENT SERVICE

BAZEL

BAZEL

- Build Event Protocol (BEP) - protobuf

BAZEL

- Build Event Protocol (BEP) - protobuf
- Build Event Service (BES) - gRPC

BUCK2 (META)

BUCK2 (META)

- Buck Event

BUCK2 (META)

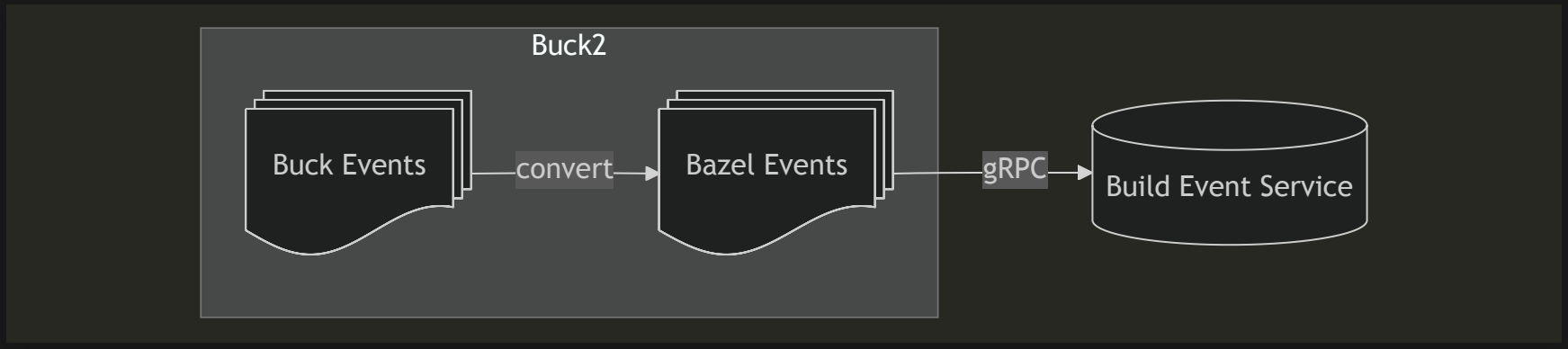
- Buck Event
- Remote Event Sink (Scribe / Thrift)

BUCK2 (OPEN SOURCE)

BUCK2 (OPEN SOURCE)

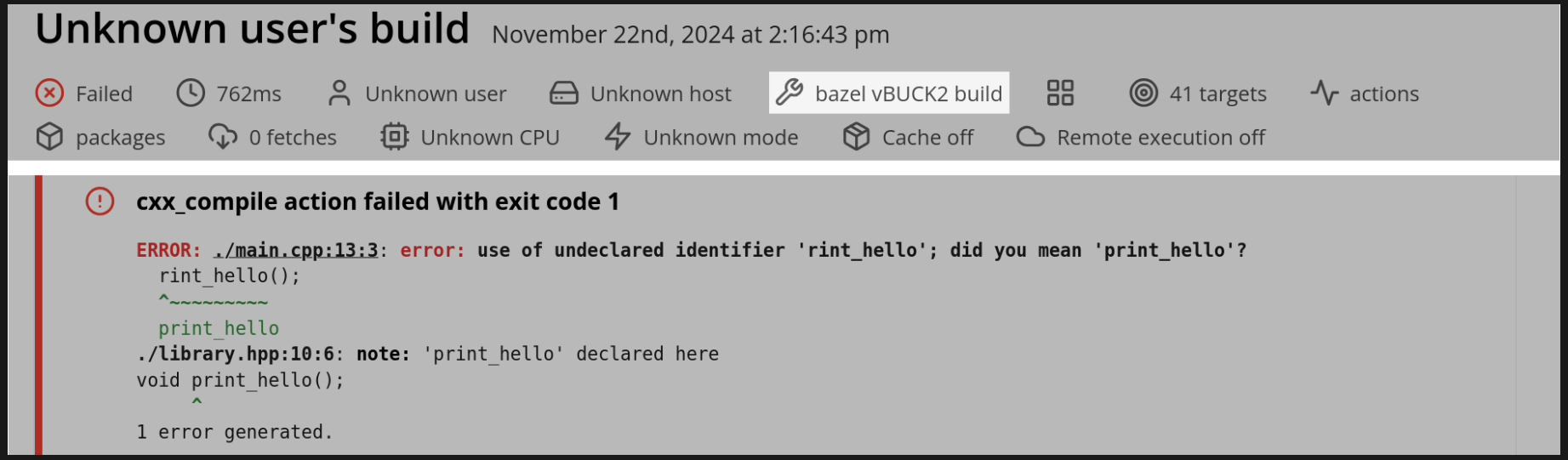
- ???

BUCK2 BUILD EVENT SERVICE



WIP implementation Buck2#811

BUCK2 BUILD EVENT SERVICE



IMPEDANCE MISMATCH

IMPEDANCE MISMATCH

- BES is Bazel specific

IMPEDANCE MISMATCH

- BES is Bazel specific
- new protocol? Buck2#226

NIX INTEGRATION

NIX INTEGRATION

- developer environment

NIX INTEGRATION

- developer environment
- toolchains

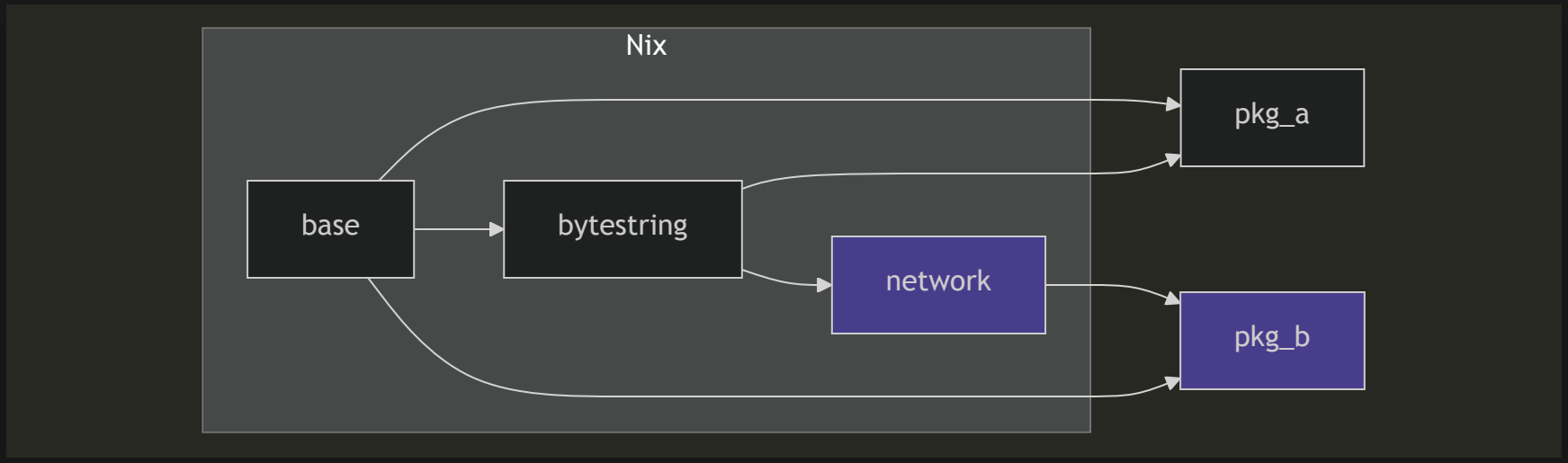
NIX INTEGRATION

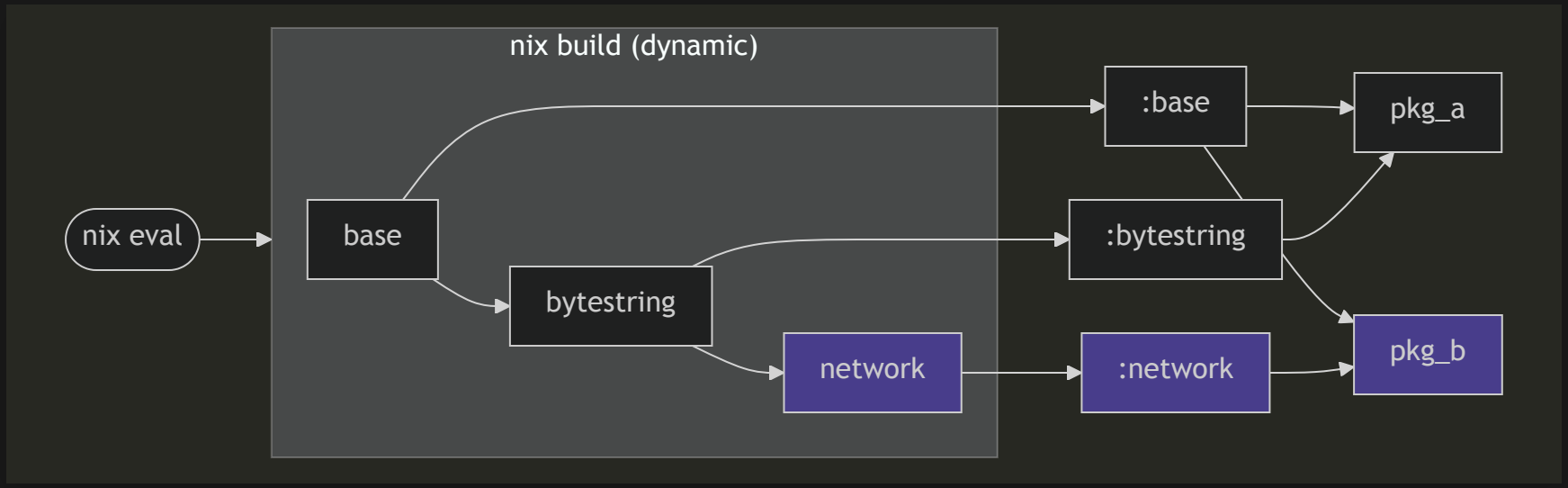
- developer environment
- toolchains
- Haskell packages

HASKELL PACKAGES

```
haskell_library(
  name = "pkg_a",
  deps = [
    "//haskell:base",
    "//haskell:bytestring",
  ],
)
```

```
haskell_library(
  name = "pkg_b",
  deps = [
    "//haskell:base",
    "//haskell:network",
  ],
)
```


Nix
base
bytestring
network
pkg_a
pkg_b


nix build (dynamic)
nix eval
base
bytestring
network
:base
:bytestring
:network
pkg_a
pkg_b

```
$ buck2 bxl haskell/toolchain.bxl:libs
```

BUCK2.NIX

BUCK2.NIX

- github.com/tweag/buck2.nix

BUCK2.NIX

- github.com/tweag/buck2.nix
- NixCon 2024 — Claudio Bley

BUCK2.NIX

- github.com/tweag/buck2.nix
- NixCon 2024 — Claudio Bley
- FOSDEM 2025 — Claudio Bley

BUCK2.NIX

- github.com/tweag/buck2.nix
- NixCon 2024 — Claudio Bley
- FOSDEM 2025 — Claudio Bley
- tweag.io/blog soon...

SUMMARY

SUMMARY

- Haskell rules

SUMMARY

- Haskell rules
- Dynamic values

SUMMARY

- Haskell rules
- Dynamic values
- Remote persistent worker

SUMMARY

- Haskell rules
- Dynamic values
- Remote persistent worker
- Build event stream

SUMMARY

- Haskell rules
- Dynamic values
- Remote persistent worker
- Build event stream
- buck2.nix

TWEAG
by Modus Create

THANK YOU