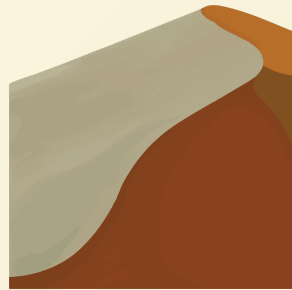


Dune

The OCaml build system



Jeremie Dimino, Jane Street

 @diml

 @dimenix

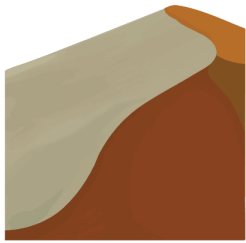


**Jane
Street**



Jane Street

OPEN SOURCE



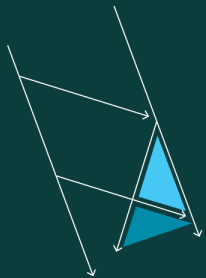
DUNE



BASE



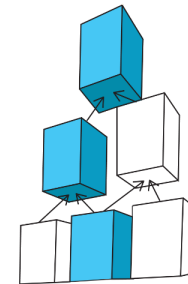
CORE



ASYNC



INCR_DOM



INCREMENTAL

The history of Dune

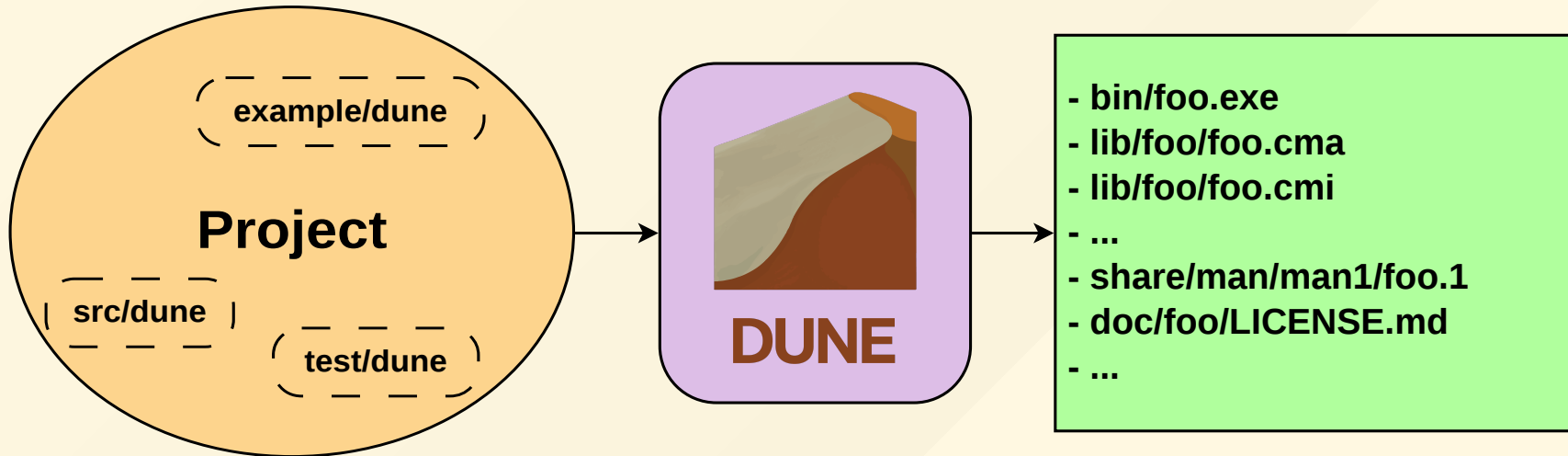
At the beginning, there was omake, ocamlbuild, ...

... then Jane Street did Jenga and the Jenga rules...

... then a domain specific tool called jbuilder...

... and finally Dune.

What is Dune?



```
(library
  (public_name mylib)
  (libraries base re lwt))

; Custom build rule
(rule (with-stdout-to m.ml (run gen/gen.exe)))
```

Dune language

```
graph TD; A[Dune language] --> B[OCaml Build Logic]; B --> C[Build system];
```

The diagram illustrates a three-step process. At the top is a light green rounded rectangle with a black border containing the text "Dune language". A black arrow points downwards from the bottom center of this box to the top center of a larger yellow rounded rectangle with a dashed black border, containing the text "OCaml Build Logic". Another black arrow points downwards from the bottom center of the yellow box to the top center of an orange rounded rectangle with a dashed black border, containing the text "Build system".

OCaml Build Logic

Build system

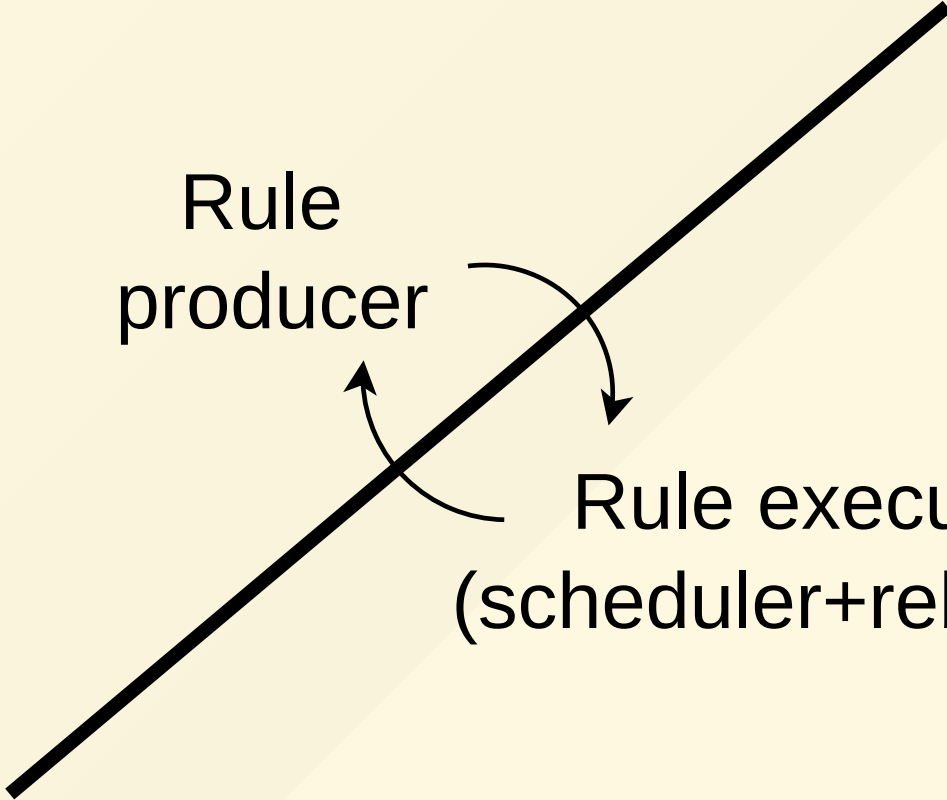
Interesting Dune features

- Composability
- Multiple build contexts

How does it work?

Rule
producer

Rule executor
(scheduler+rebuilder)



```
let read_dune_file fname =  
  let contents = read_file fname in  
  parse_dune_file fname  
  
let get_lib dir =  
  let stanzas =  
    read_dune_file (Path.relative dir "dune")  
  in  
  List.find_map stanzas ~f:(function  
    | Library lib -> Some lib.name  
    | _ -> None)
```

Memoisation framework

`get_lib("foo/src")`



```
graph TD; A[get_lib("foo/src")] --> B[read_dune_file("foo/src/dune")]; B --> C[read_file("foo/src/dune")];
```

`read_dune_file("foo/src/dune")`

`read_file("foo/src/dune")`

The end



dune.build



discuss.ocaml.org



opensource.janestreet.com