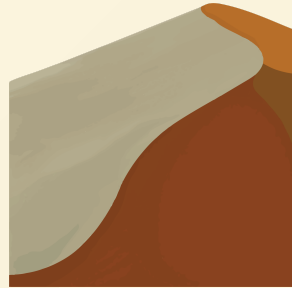


# Dune

## The OCaml build system



**Jeremie Dimino, Jane Street**

 **@diml**

 **@dimenix**

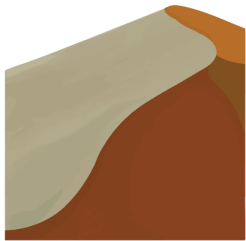


**Jane  
Street**



Jane Street

# OPEN SOURCE



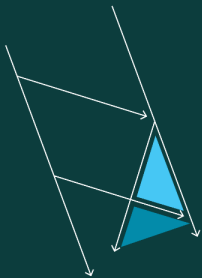
**DUNE**



**BASE**



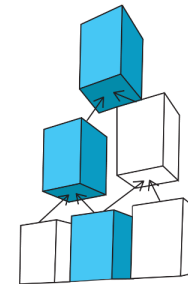
**CORE**



**ASYNC**

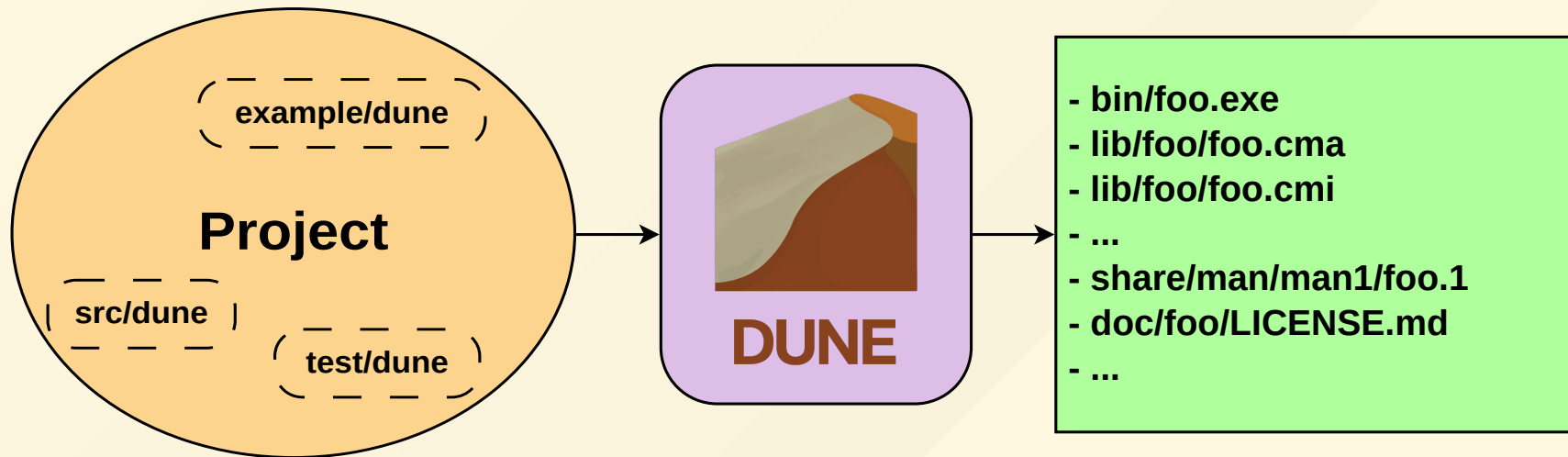


**INCR\_DOM**



**INCREMENTAL**

**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 solid green rounded rectangle labeled 'Dune language'. A black arrow points down from its bottom center to a yellow rounded rectangle with a dashed black border, labeled 'OCaml Build Logic'. Another black arrow points down from the bottom center of the yellow rectangle to an orange rounded rectangle with a dashed black border, labeled 'Build system'.

**OCaml Build Logic**

**Build system**

# Interesting Dune features

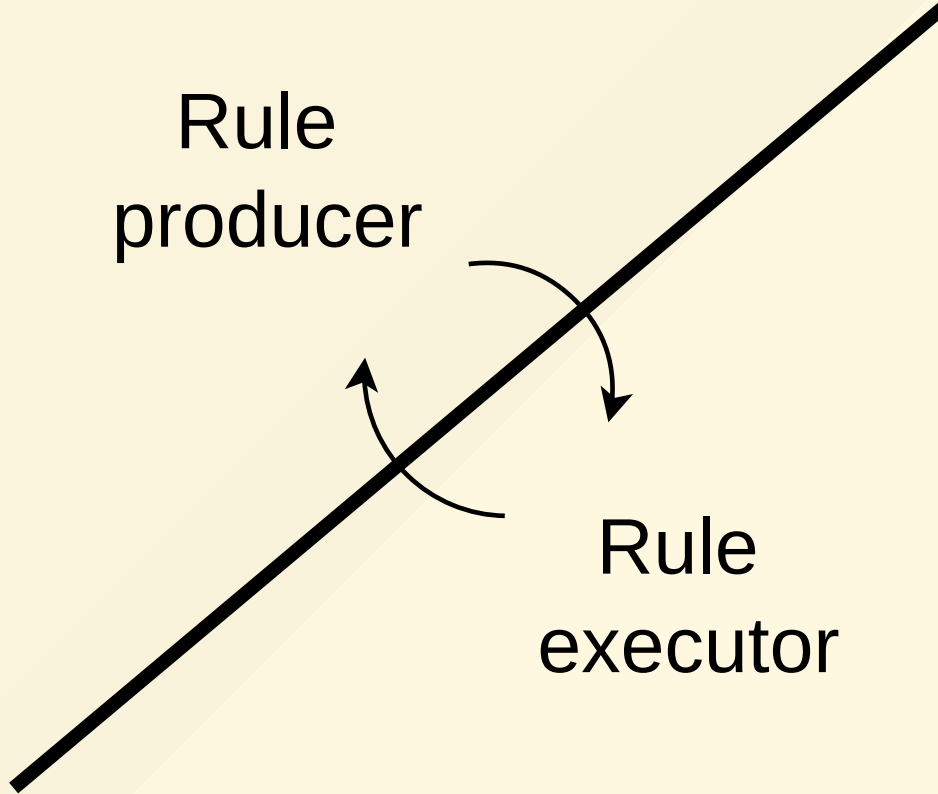
- Composability
- Multiple build contexts

**How does it work?**

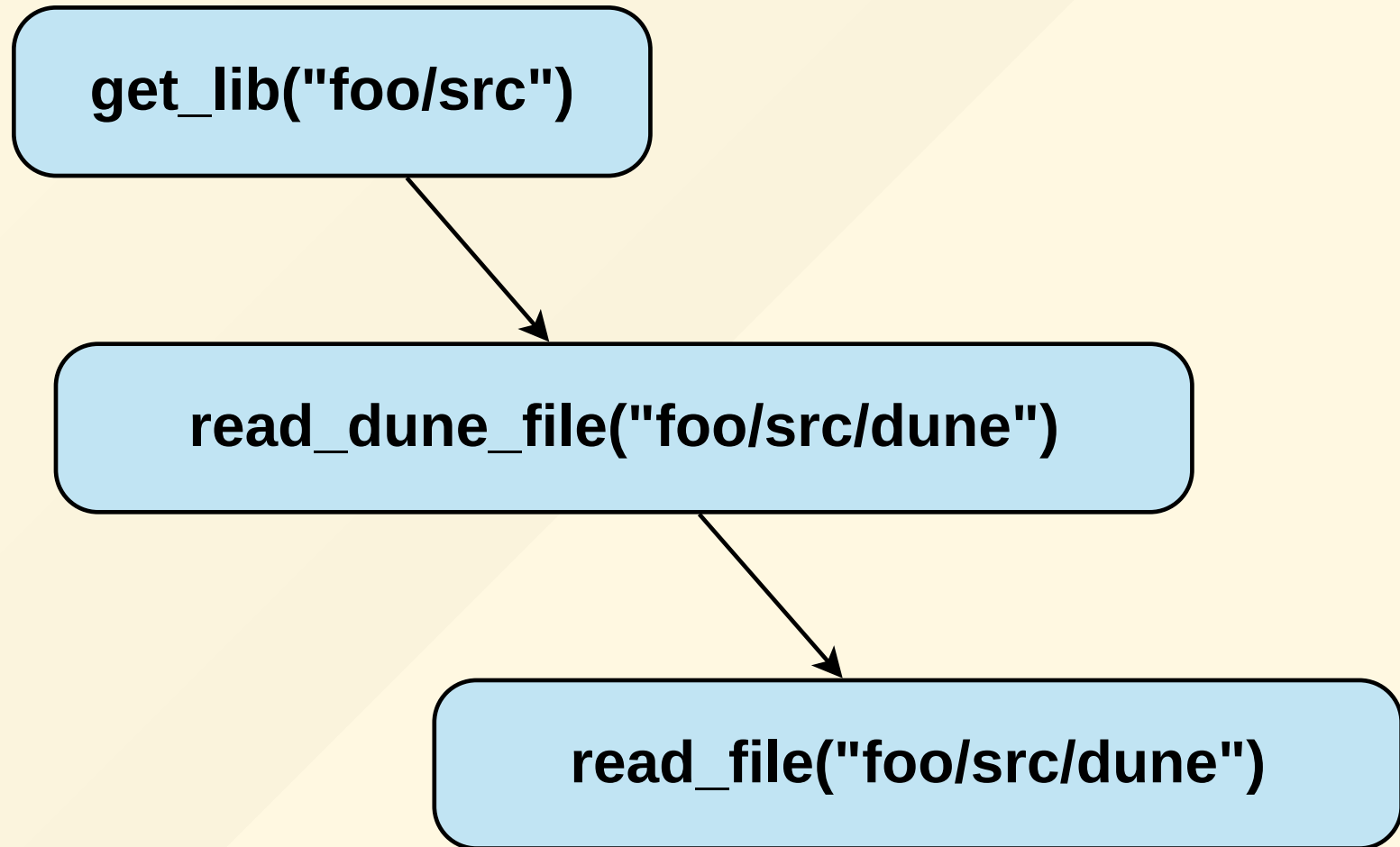


Rule  
producer

Rule  
executor



# Memoisation framework



# The end



[dune.build](https://dune.build)



[discuss.ocaml.org](https://discuss.ocaml.org)



[opensource.janestreet.com](https://opensource.janestreet.com)