

Zoology of lockfree concurrent data structures

Clément Allain

INRIA Paris

June 16, 2024

Zoology

Specimen ① : Michael-Scott queue

Specimen ② : KCAS

Zoology: what is it?

HEAPLANG (modified)

- + ADTs
- + DLS
- + exceptions
- + algebraic effects
- + relaxed memory

(planned before OSIRIS,
in case you were wondering)

Zoo

IRIS

Coq

Zoology: what is it?

```
MetaCoq Run (zoo_variant "list" [  
  "Nil" ;  
  "Cons"  
]).
```

```
Definition map : val :=  
  rec: "map" "fn" "xs" :=  
    match: "xs" with  
    | Nil =>  
      §Nil  
    | Cons "x" "xs" =>  
      let: "y" := "fn" "x" in  
      'Cons{ "y", "map" "fn" "xs" }  
  end.
```

Coq

Zoology: what is it?

HEAPLANG (modified)

- + ADTs
- + DLS
- + exceptions
- + algebraic effects
- + relaxed memory

(planned before OSIRIS,
in case you were wondering)

Zoo

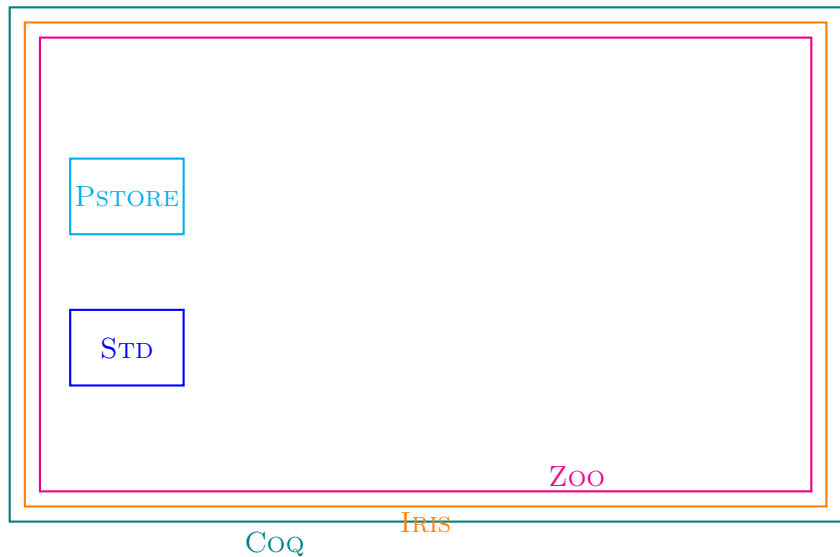
IRIS

Coq

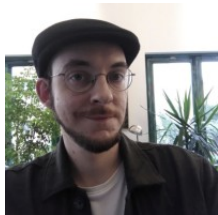
Zoology: what is it?



Zoology: what is it?



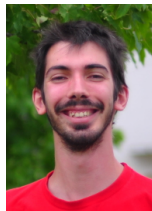
Zoology: what is it?



Basile
Clément



Alexandre
Moine



Gabriel
Scherer

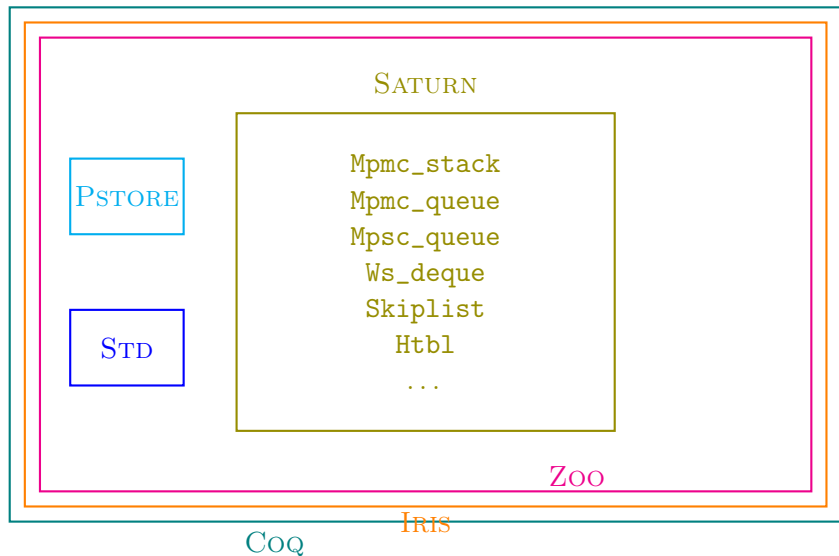
The PSTORE team!

Coq

Iris

ZOO

Zoology: what is it?



Zoology: what is it?



Vesa Karvonen



Carine Morel

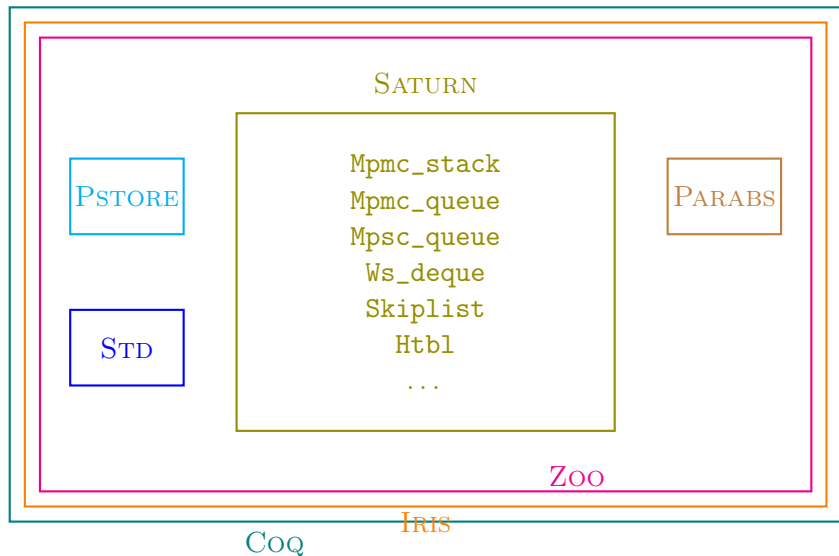
The SATURN team!

Coq

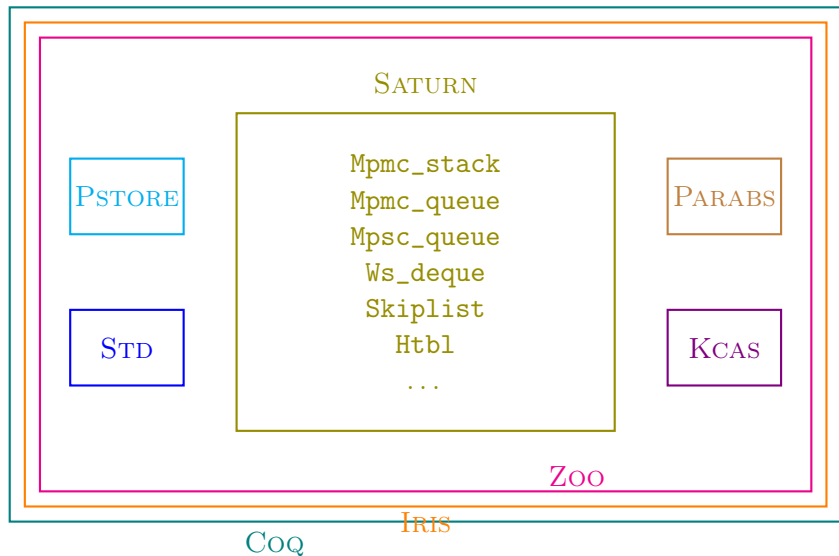
IRIS

ZOO

Zoology: what is it?



Zoology: what is it?



Zoology: what is it?



Vesa Karvonen

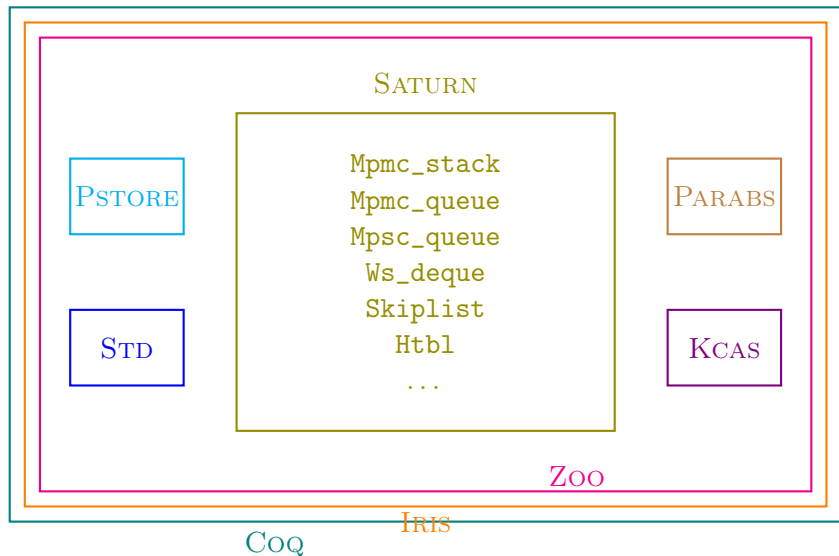
The main author of KCAS!

Coq

IRIS

ZOO

Zoology: what is it?



Zoology: why is it fun?

Lockfree algorithms typically exhibit complex behaviors:

- ▶ physical state \neq logical state,
- ▶ external linearization points,
- ▶ future-dependent linearization points.

IRIS is a good match for verifying them thanks to advanced mechanisms:

- ▶ invariants to enforce protocols,
- ▶ atomic updates to materialize linearization points,
- ▶ prophecy variables to reason on the future.

Zoology

Specimen ① : Michael-Scott queue

Specimen ② : KCAS

Michael-Scott queue

TODO

Zoology

Specimen ① : Michael-Scott queue

Specimen ② : KCAS

TODO

Thank you for your attention!