Coq Implementors Workshop

Maxime Dénès & Enrico Tassi – 30/5/2016

Schedule

	Monday 30	*Tuesday 31*	*Wednesday 1*	*Thursday 2*	*Friday 3*
8 AM		**Room (6)**	**Room (6)**	**Room (6)**	**Room (6)**
9 AM					
10 AM	**Intro (1) (Enrico Tassi, Maxime Dénès)**	**Ltac internals (Pierre-Marie Pédrot)**	**Notations (Hugo Herbelin)**	**Parallelism (Enrico Tassi)**	**Universes (Matthieu Sozeau)**
11 AM	**Round table (2)**	**Code (4)**	**Code (4)**	**Code (4)**	**Code (4)**
12 PM	**Lunch**				
1 PM					
2 PM	**Code extraction (Pierre Letouzey)**	**Code (4)**	**Code (4)**	**Code (4)**	**Debriefing (5)**
3 PM	**Code (4)**				**Code (4)**
4 PM		**Roadmap 8.6 (7)**	**Roadmap 8.6 (7)**	**Roadmap 8.6 (7)**	
5 PM	**Pub (3)**				
6 PM		**Room (6)**	**Room (6)**	**Room (6)**	**Room (6)**
7 PM					
Building (room):	**Kahn**	**Galois (Coriolis)**	**Kahn**	**Kahn**	**Kahn**

Food: 2 options

1. Here at Inria, nothing fancy but good price: 7.50€
2. St. Philippe (700m downhill) offers more choice but at higher prices. Eg. chees burger with fries or dish of the day at ~ 15€.

Contributions: where/how

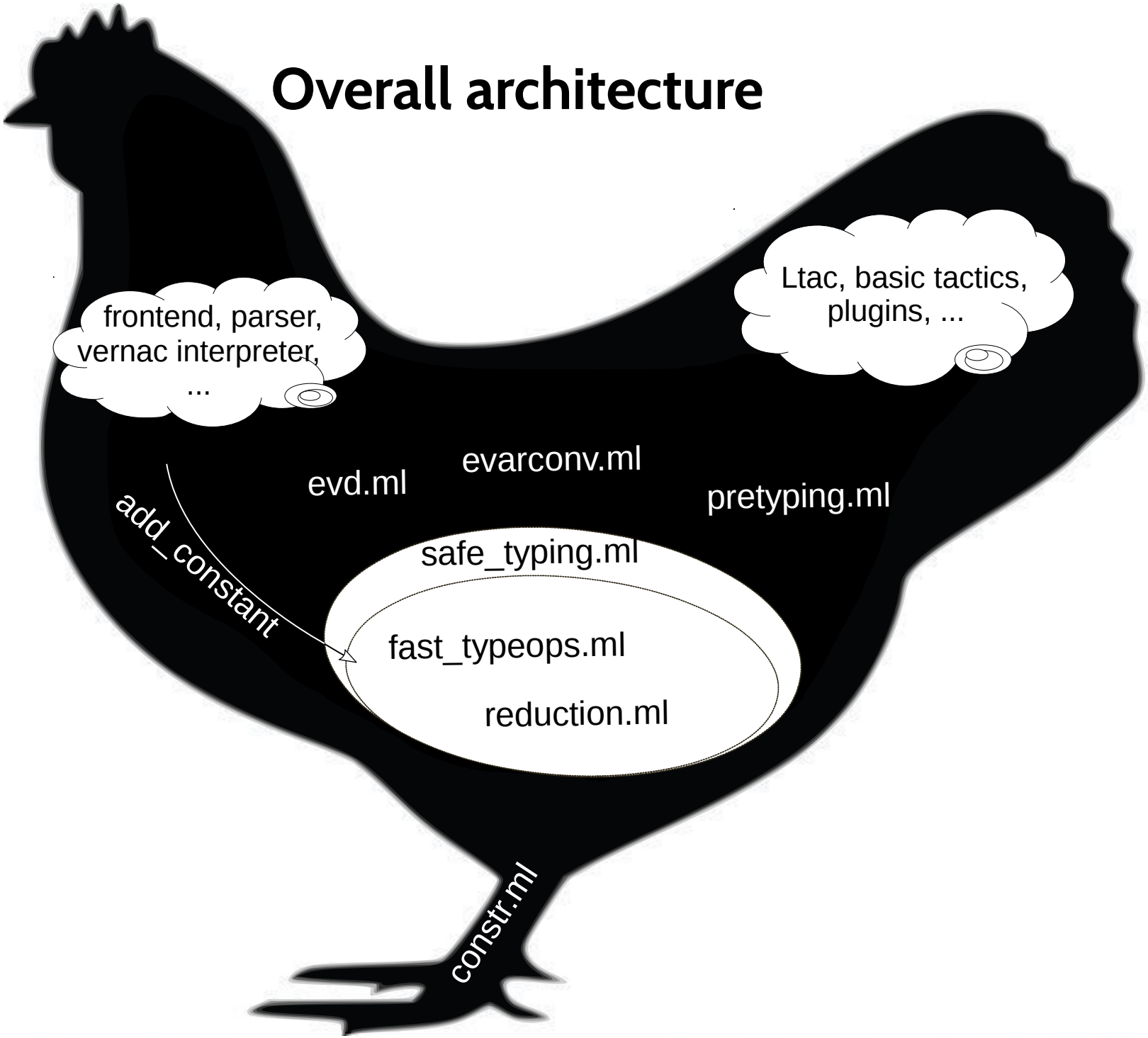
1. Plugin: put your code on github
2. Patches: pull requests to coq/coq
3. Bugfix: coordinate using the bugtracker

In all cases, please log your activity in the dedicated page

2

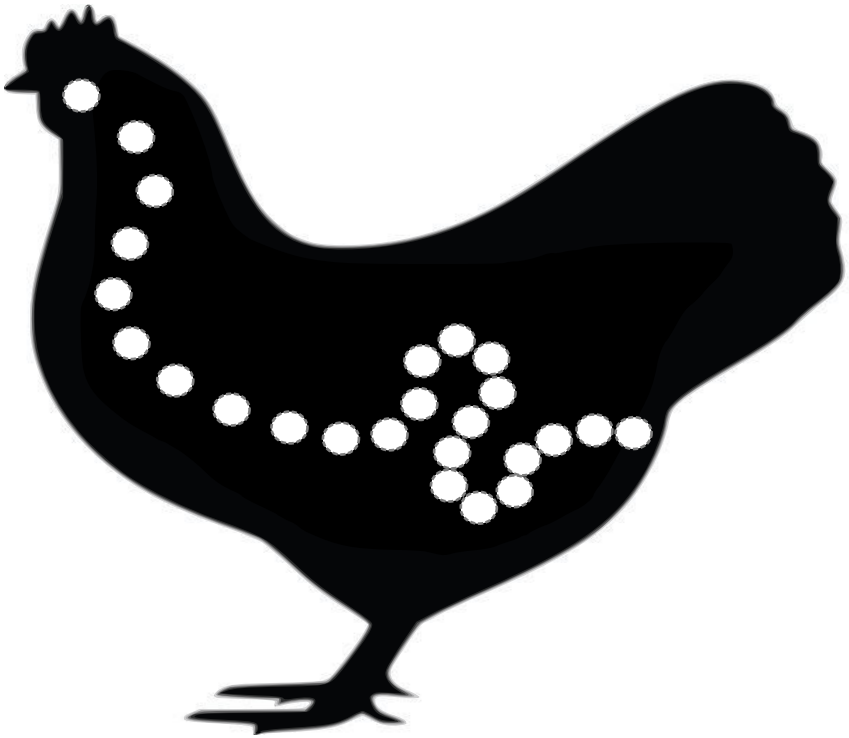
Bird eye view of Coq's internals

Overall architecture



Guided tour

```
Definition foo :=
  fun x => x = 3.
Print foo.
```



```
foo : fun x : nat => x = 3
```

Data types and transformations

string

```
Definition foo :=
  fun x => x = 3.
Print foo.
```

parsing

constr_expr (AST)

```
VernacDefinition( “foo”,
  DefinedBody(
    CLambdaN([ “x”, CHole],
      CNotation( “_ = _”,
        [ CRef “x”; CPrim 3 ])))
VernacPrint (PrintName “foo”)
```

internalization
(notations, globals, implicit args)

glob_constr (untyped)

```
GLambda( “x”, GHole,
  GApp( GRef “Coq.Init.Logic.eq”,
    [ GHole;
      GVar “x”;
      GApp( GRef “Coq.Init.Datatypes.S”,
        […GRef “Coq.Init.Datatypes.O”...])]))
```

pretyping
(De Bruijn idxs,coercions,…)

constr (typed)

```
Lambda( “x”, Ind “Coq.Init.Datatypes.nat”,
  App( Ind “Coq.Init.Logic.eq”,
    [ Ind “Coq.Init.Datatypes.nat”;
      Rel 1;
      App(Construct “Coq.Init.Datatypes.S”,
        […Construct “Coq.Init.Datatypes.O”...])]))
```

Data types involved

string

```
fun x : nat => x = 3.
```

printing

constr_expr

```
CLambdaN([ “x”, CRef “nat” ],
  CNotation( “_ = _”,
    [ CRef “x”; CPrim 3 ])))
```

externalization

glob_constr

```
GLambda( “x”, GRef “Coq.Init.Datatype.nat”,
  GApp( GRef “Coq.Init.Logic.eq”,
     [ GRef “Coq.Init.Datatypes.nat”;
       GVar “x”;
       GApp( GRef “Coq.Init.Datatypes.S”,
         […GRef “Coq.Init.Datatypes.O”...])]))
```

detyping

constr

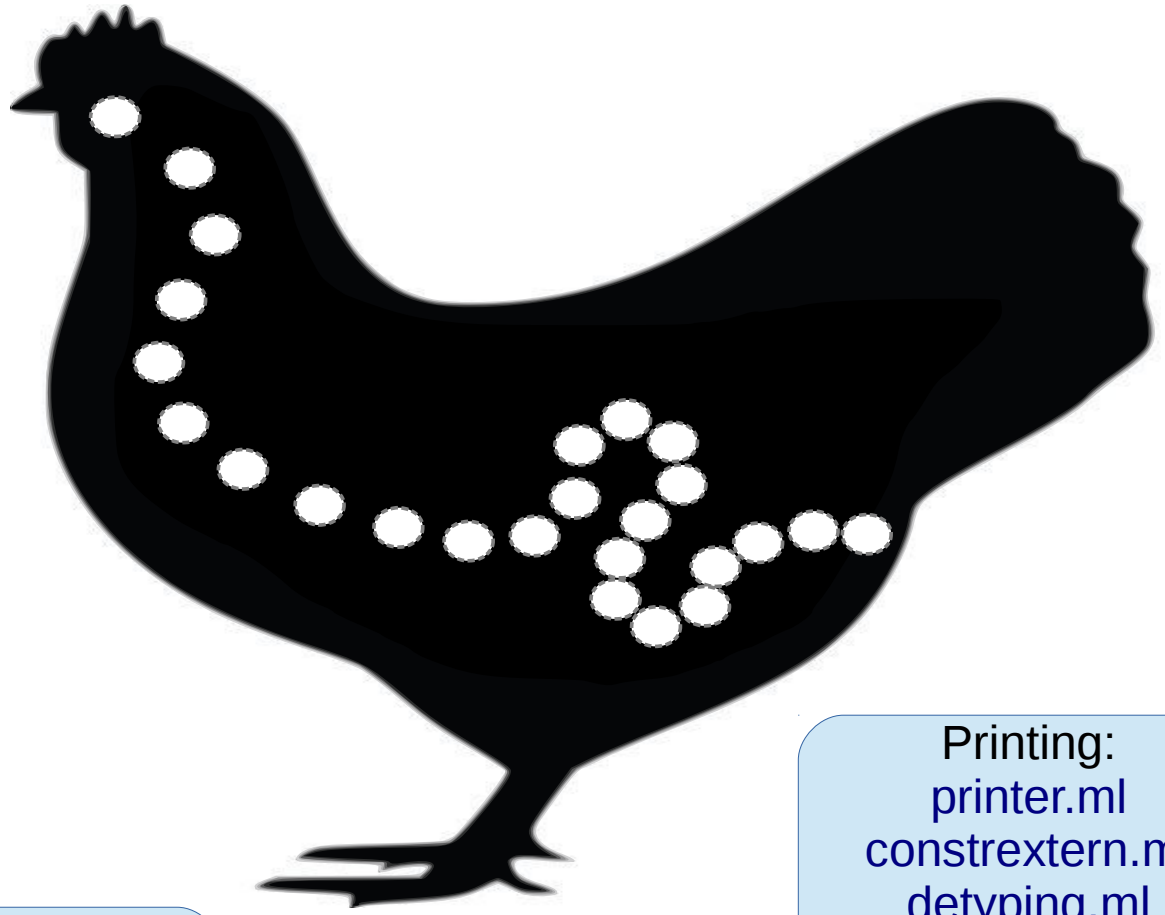
```
Lambda( “x”, Ind “Coq.Init.Datatypes.nat”,
  App( Ind “Coq.Init.Logic.eq”,
     [ Ind “Coq.Init.Datatypes.nat”;
       Rel 1;
       App( Construct “Coq.Init.Datatypes.S”,
         […Construct “Coq.Init.Datatypes.O”...])]))
```

Where's the code?

Frontend:
vernac.ml

Parsing:
g_vernac.ml4 g_constr.ml4
vernac_expr constr_expr

Interpreter:
vernacentries.ml
(dumbglob.ml)



Printing:
printer.ml
constrextern.ml
detyping.ml

Term internalization:
constrintern.ml
notation.ml
glob_constr

Type inference:
pretyping.ml
constr

3

Demo

Demo

1. Create a feature branch
2. Test it on ci.inria.fr/coq

4

Next

Roundtable

1. Everybody with a project in mind talks about it (5' max)
2. So that we know what you are going to do
3. So that you can group with others working on similar projects