

# rapture

## mainly modes



C O D A C Y

Johann Egger

- ▶ a (beautiful) way to abstract failures
- ▶ different types of monadic types that we use everywhere

- ▶ classes from `scala.reflect` (the thing that meta wants to replace)
- ▶ quasiquotes (extractors & constructors)
- ▶ the documentation

## def-macros

- ▶ implementation in separate (sbt-)project
- ▶ rhs: keyword 'macro' + call to implementation
- ▶ implementation-params: context + [expression] + [typeTag]

ex0: materializer

## **example: materializer 0|x**

- ▶ sbt structure
- ▶ common shared between in macro code and non-macro code

## example: materializer 1|x

- ▶ problem: no (deep) recursion
- ▶ problem: duplicated in every companion
- ▶ solution: resolve it implicitly (using generic type)

ex1: static validator



## example: static validator 1|x

- ▶ email address: parser returns Try
- ▶ sometimes values can be verified @ compile-time :)
- ▶ string-context + macro

ex2: annotation

## example: annotation 1|x

- ▶ needs paradise added to implementation & user code
- ▶ allows type modifications (type providers)
- ▶ example in core: class based on wrapped type

# thank you!



C O D A C Y

[johann@codacy.com](mailto:johann@codacy.com)