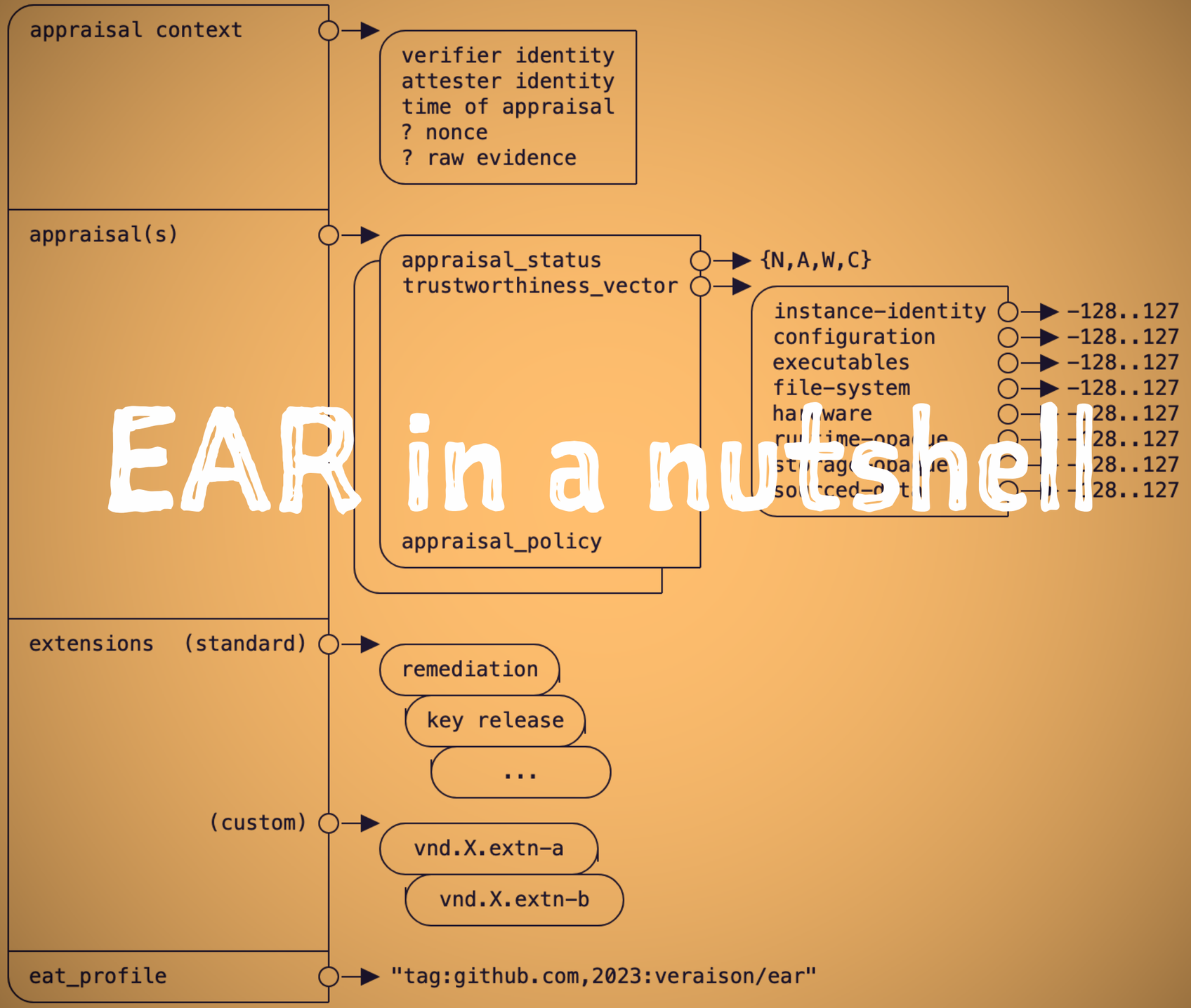
EAT AR4SI

draft-fv-rats-ear-00

RATS WG, IETF 116, Yokohama

Where were we: IETF 115 "Next Steps"

- Continue with the experimentation ✔
- Talk to adjacent communities ✔
- Write up the proposal in an I-D ✔
- Present progress update at IETF 116 ✔



EAR in a nutshell

Prototyping and impact on design

We spent the last few months prototyping

Two non-trivial use cases we dealt with:

- Composite attesters
- Split appraisal: e.g., platform vs workload verifier (typical CC use case)

↪ Adjust initial design to allow multiple separate (and possibly incomplete) appraisals to co-exist within the same EAR

Running code

Golang package and CLI

github.com/veraison/ear
github.com/veraison/ear/arc

- version: 1.0.1
- license: Apache 2.0
- roles: Verifier / RP
- serialisation: JWT
- docs: pkg.go.dev/github.com/veraison/ear

C library

github.com/veraison/c-ear

- version: 0.0.1
- license: Apache 2.0
- roles: (minimalist) RP
- serialisation: JWT
- docs: ear.h

EAR ∩ RATS ≠ ∅

- EAR = EAT + AR4SI
  - Looks like a good fit for a WG item (A)
  - Could as well be *just* a EAT profile (B)
- The charter supports both outcomes
- (We'd be happy either way)

Q: A or B?

Pointers

- I-D GitHub repo & Issue tracker
- Go Package
- Go CLI
- C Library