Verifiable Claims Test Suite

presented by Ganesh Annan

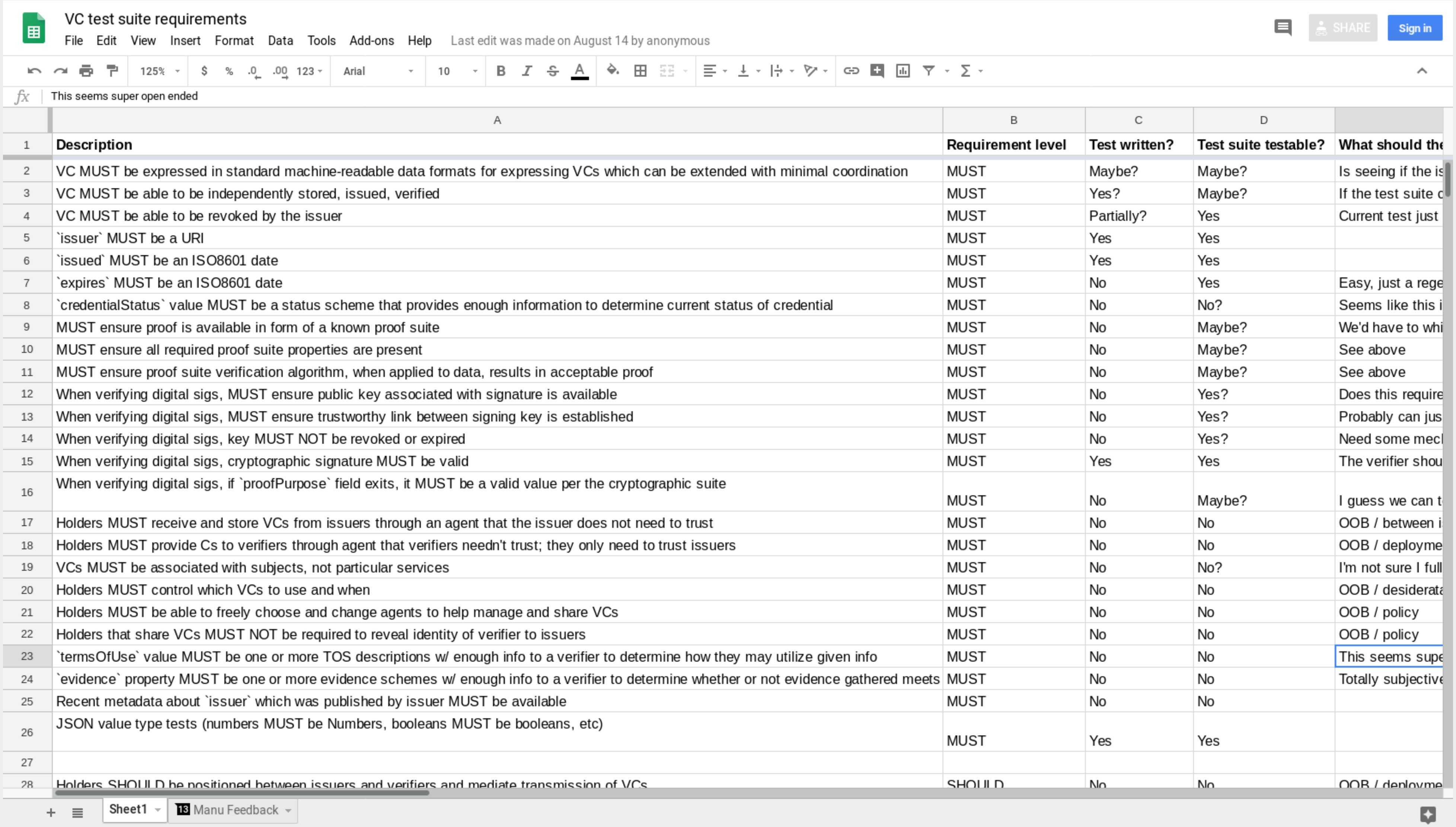
proxying for Chris Webber

DISCLAIMER

Anything that is bad is Chris Webber's fault

Including these slides

Plenty of requirements to test...

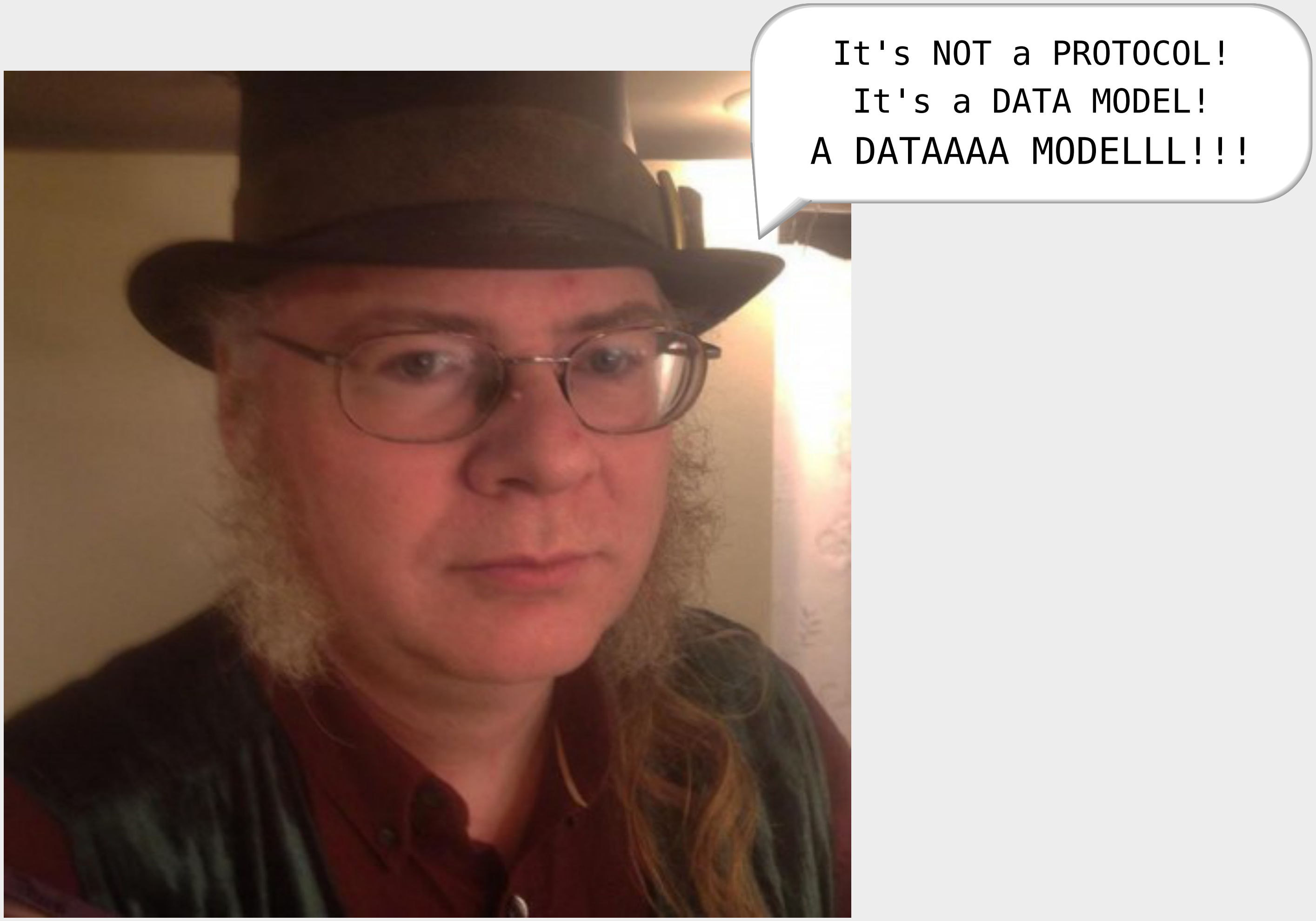

VC test suite requirements
File Edit View Insert Format Data Tools Add-ons Help
Last edit was made on August 14 by anonymous
SHARE
Sign in
125%
$ % .0 .00 123
Arial
10
B I S A
fx
Sheet1
Manu Feedback

This seems super open ended

	A	B	C	D	
1	**Description**	**Requirement level**	**Test written?**	**Test suite testable?**	**What should the**
2	VC MUST be expressed in standard machine-readable data formats for expressing VCs which can be extended with minimal coordination	MUST	Maybe?	Maybe?	Is seeing if the is
3	VC MUST be able to be independently stored, issued, verified	MUST	Yes?	Maybe?	If the test suite c
4	VC MUST be able to be revoked by the issuer	MUST	Partially?	Yes	Current test just
5	\`issuer\` MUST be a URI	MUST	Yes	Yes	
6	\`issued\` MUST be an ISO8601 date	MUST	Yes	Yes	
7	\`expires\` MUST be an ISO8601 date	MUST	No	Yes	Easy, just a rege
8	\`credentialStatus\` value MUST be a status scheme that provides enough information to determine current status of credential	MUST	No	No?	Seems like this i
9	MUST ensure proof is available in form of a known proof suite	MUST	No	Maybe?	We'd have to whi
10	MUST ensure all required proof suite properties are present	MUST	No	Maybe?	See above
11	MUST ensure proof suite verification algorithm, when applied to data, results in acceptable proof	MUST	No	Maybe?	See above
12	When verifying digital sigs, MUST ensure public key associated with signature is available	MUST	No	Yes?	Does this require
13	When verifying digital sigs, MUST ensure trustworthy link between signing key is established	MUST	No	Yes?	Probably can jus
14	When verifying digital sigs, key MUST NOT be revoked or expired	MUST	No	Yes?	Need some mecl
15	When verifying digital sigs, cryptographic signature MUST be valid	MUST	Yes	Yes	The verifier shou
16	When verifying digital sigs, if \`proofPurpose\` field exits, it MUST be a valid value per the cryptographic suite	MUST	No	Maybe?	I guess we can t
17	Holders MUST receive and store VCs from issuers through an agent that the issuer does not need to trust	MUST	No	No	OOB / between i
18	Holders MUST provide Cs to verifiers through agent that verifiers needn't trust; they only need to trust issuers	MUST	No	No	OOB / deployme
19	VCs MUST be associated with subjects, not particular services	MUST	No	No?	I'm not sure I full
20	Holders MUST control which VCs to use and when	MUST	No	No	OOB / desiderata
21	Holders MUST be able to freely choose and change agents to help manage and share VCs	MUST	No	No	OOB / policy
22	Holders that share VCs MUST NOT be required to reveal identity of verifier to issuers	MUST	No	No	OOB / policy
23	\`termsOfUse\` value MUST be one or more TOS descriptions w/ enough info to a verifier to determine how they may utilize given info	MUST	No	No	This seems supe
24	\`evidence\` property MUST be one or more evidence schemes w/ enough info to a verifier to determine whether or not evidence gathered meets	MUST	No	No	Totally subjective
25	Recent metadata about \`issuer\` which was published by issuer MUST be available	MUST	No	No	
26	JSON value type tests (numbers MUST be Numbers, booleans MUST be booleans, etc)	MUST	Yes	Yes	
27					
28	Holders SHOULD be positioned between issuers and verifiers and mediate transmission of VCs	SHOULD	No	No	OOB / deployme

some of these were removed, but still...

And yet there’s this complication...



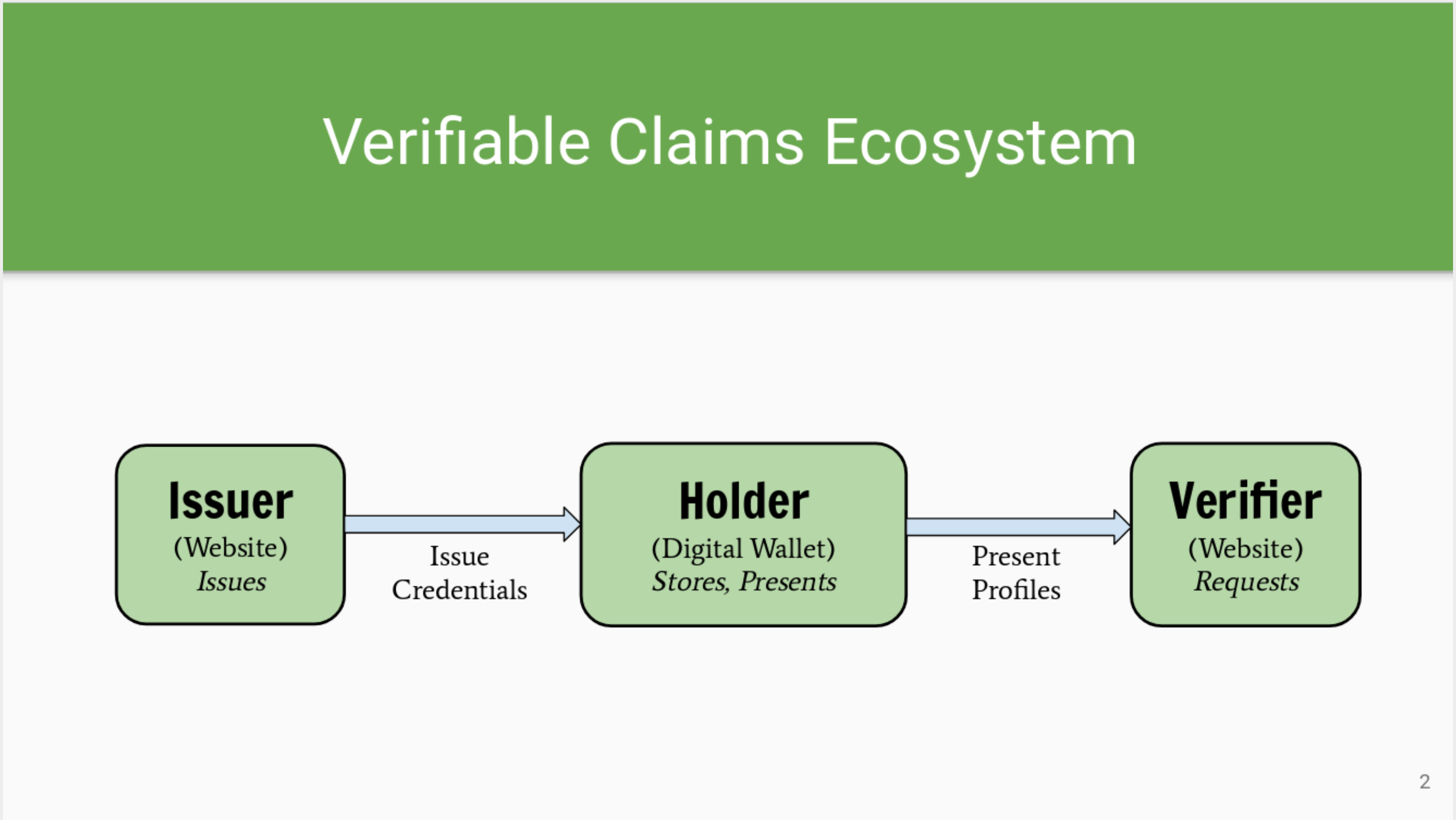
tallted is right, but how do we test a data model?

Also...

Verification mechanism is flexible

Different VC implementations understand different types of proofs

Last year's ecosystem overview...



Verifiable Claims Ecosystem

Issuer
(Website)
Issues

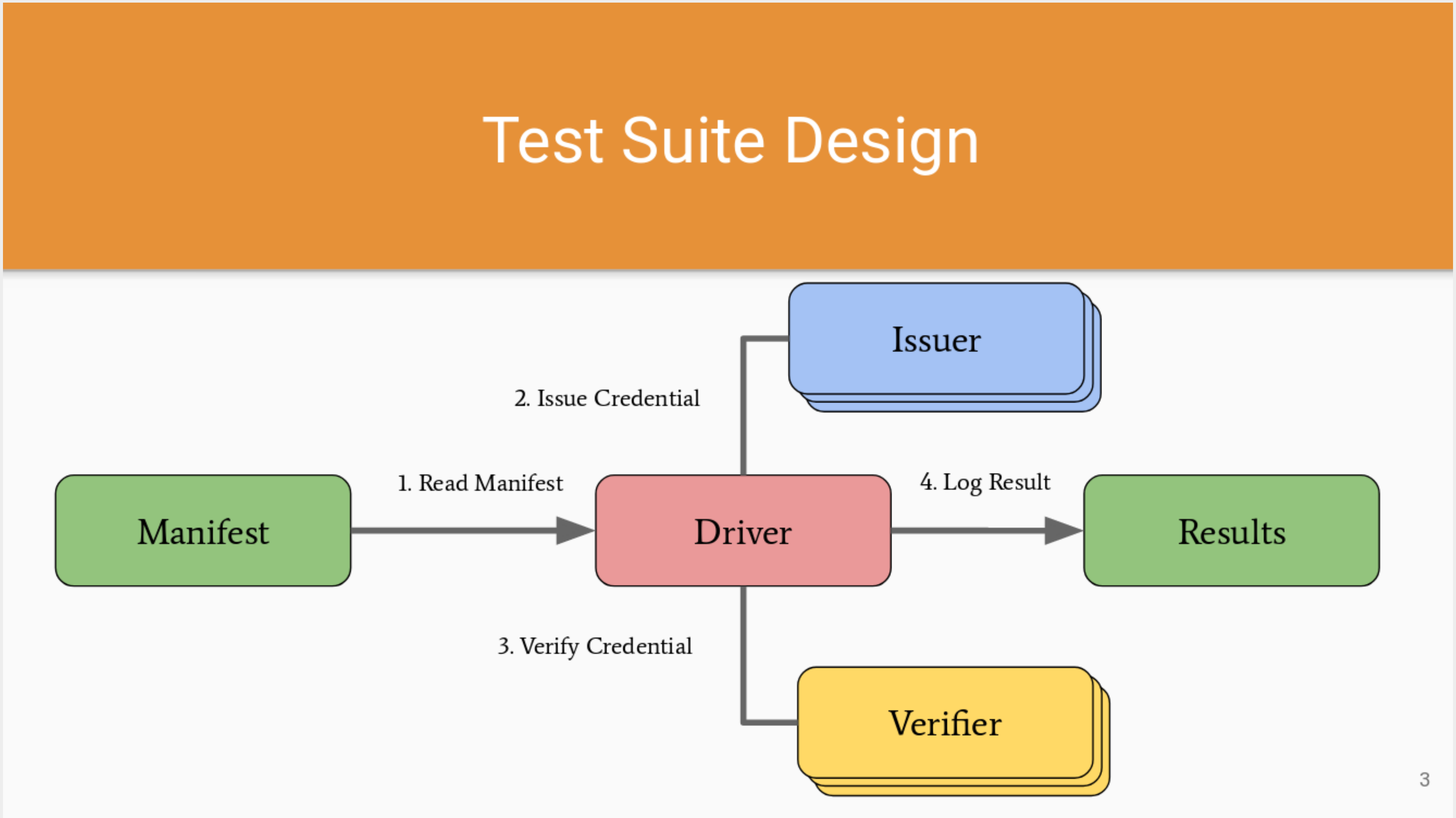
Issue Credentials

Holder
(Digital Wallet)
Stores, Presents

Present Profiles

Verifier
(Website)
Requests

BYO issuer/verifier, we’ll supply the driver!



Test Suite Design

Manifest

1. Read Manifest

Driver

2. Issue Credential

Issuer

3. Verify Credential

Verifier

4. Log Result

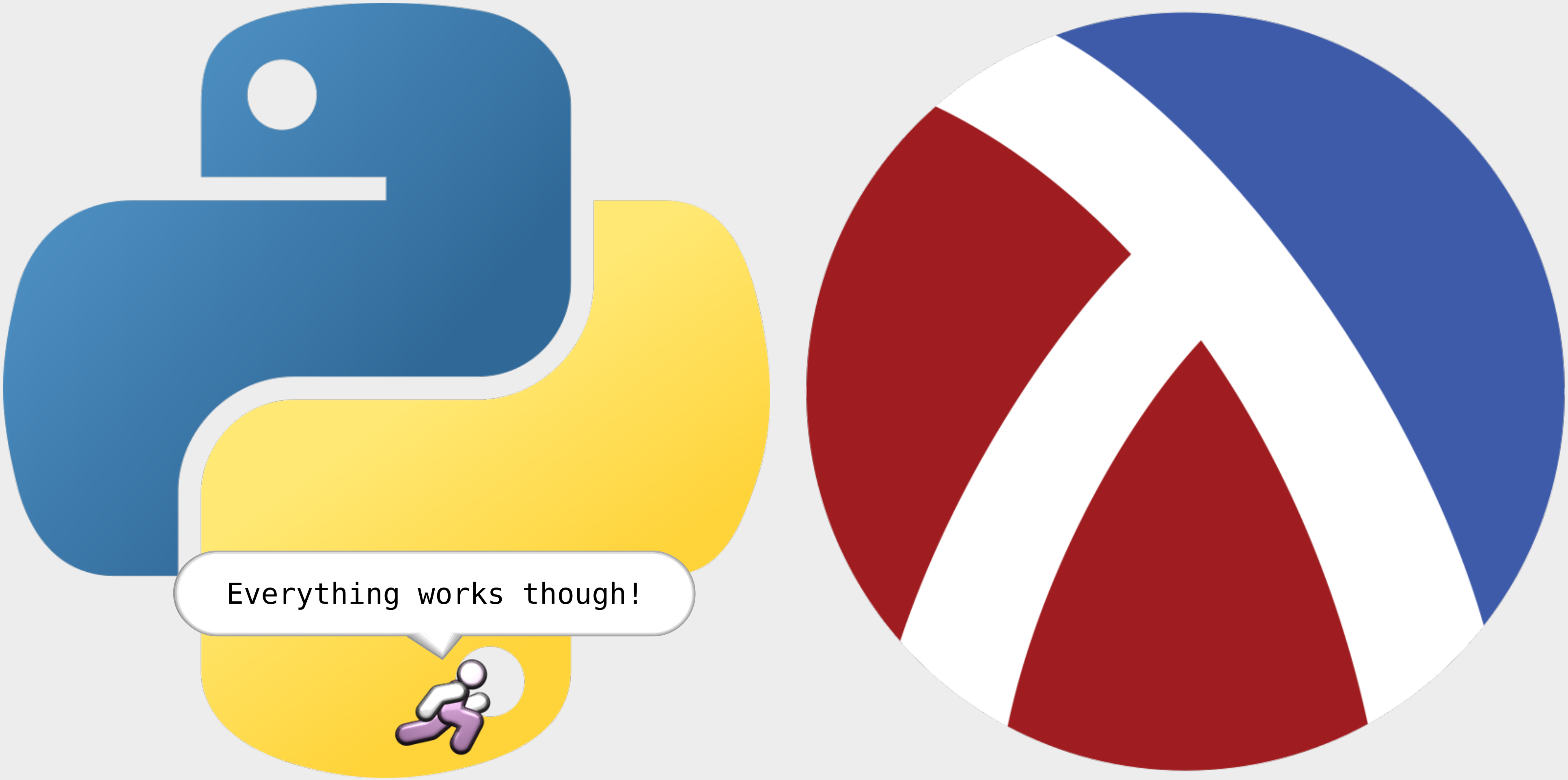
Results

```
       _____
    .-'  ~~~'.
   |  O  .-- --.
   ' o   .--..--
   _\__-||  ||  |   GOOD NEWS, EVERYONE!
   ('   ''--''\-'
    \\,  - (_  \'
   _ /    /  '-'.
 _/ \//  .'----'
/ '  \/  `_.'
   \_ \-_.'
     \ \_/ |
```

- The test suite works!
- 28 of 31 tests implemented (or more, some tests include sub-tests)
- DB’s Javascript implementation is passing all but 3 of those tests

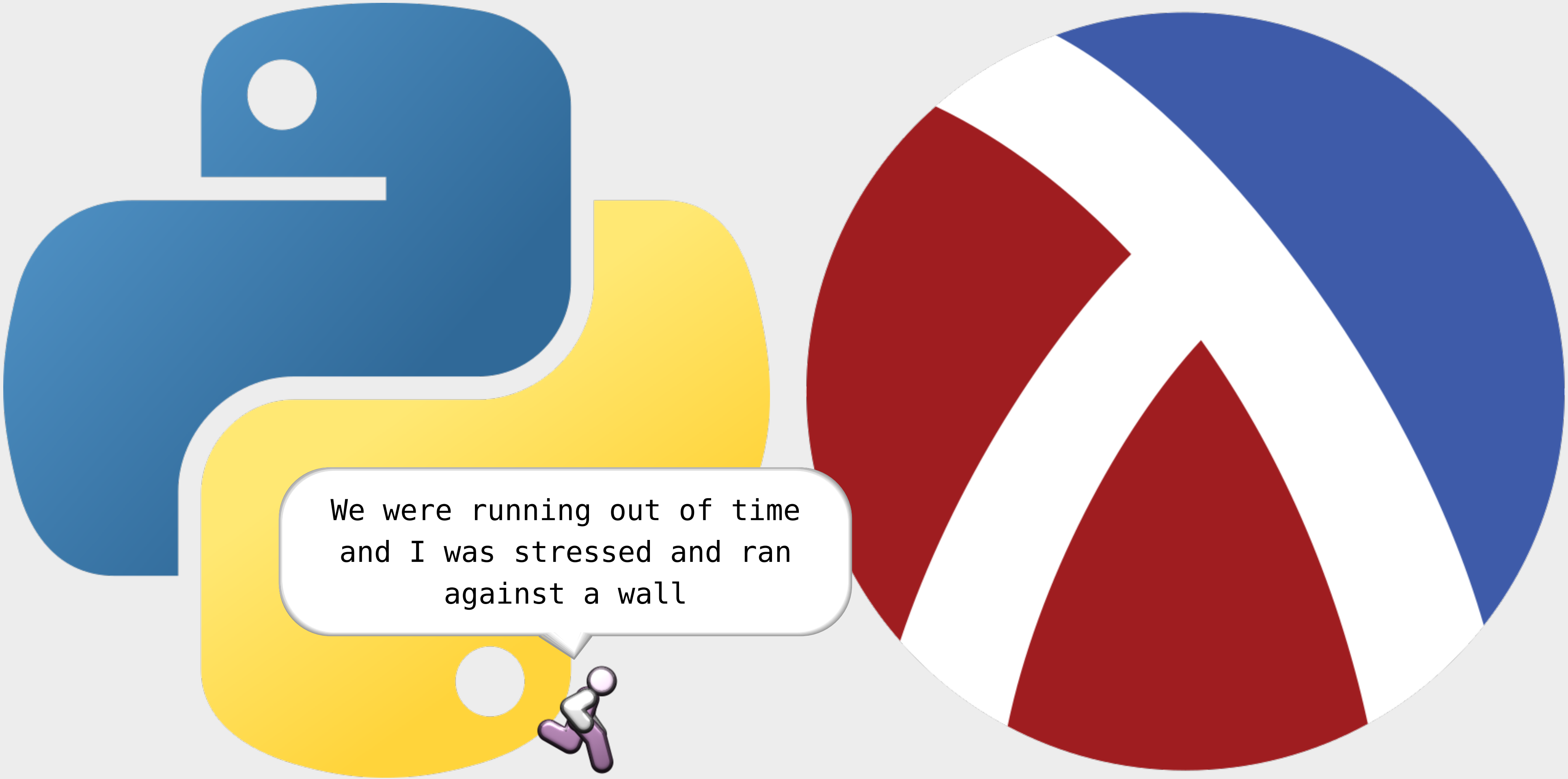
Bad news, everyone???

Chris reimplemented the test suite from Python to Racket while nobody was looking



Bad news, everyone???

Chris reimplemented the test suite from Python to Racket while nobody was looking


We were running out of time and I was stressed and ran against a wall

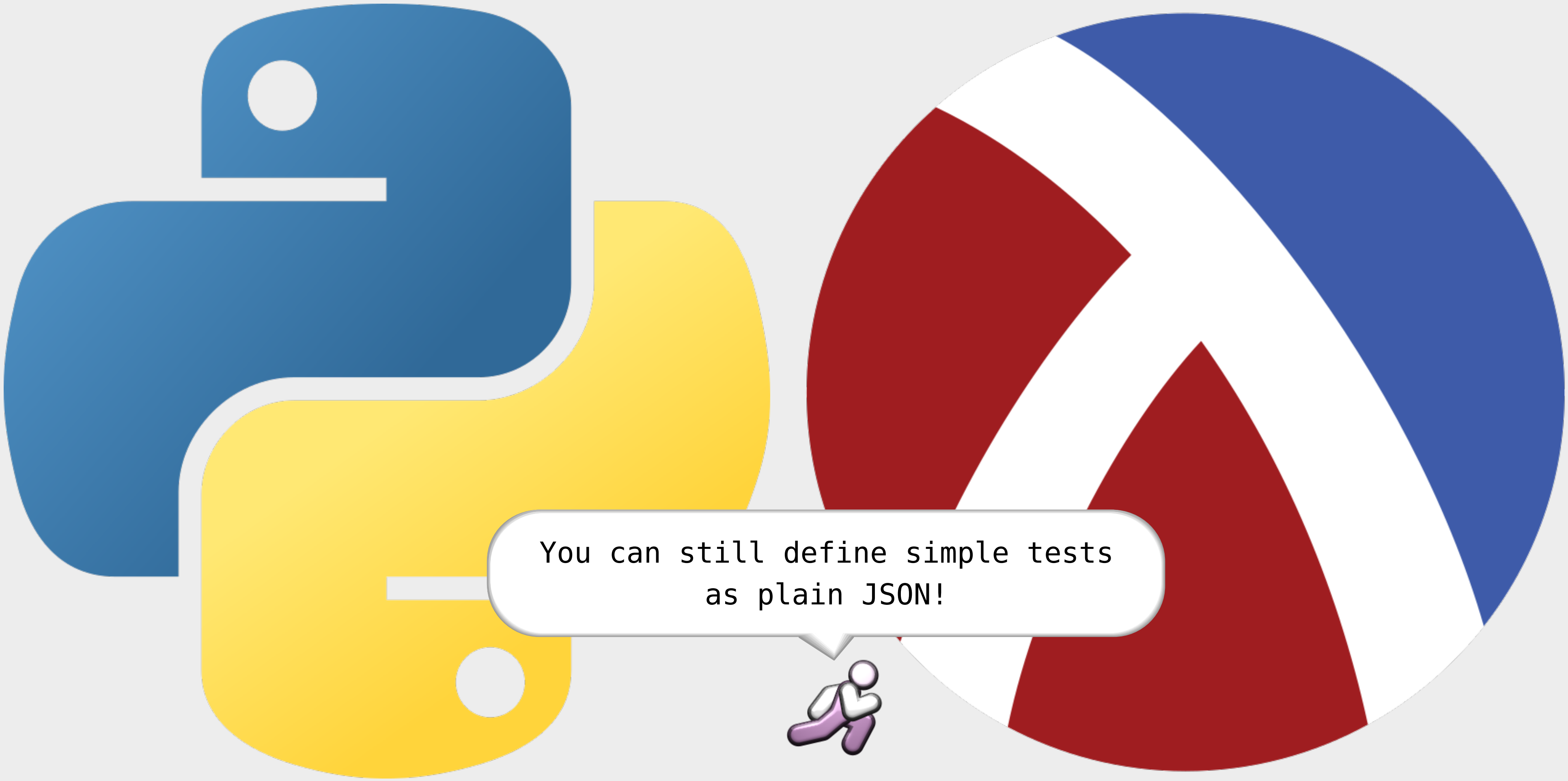
Bad news, everyone???

Chris reimplemented the test suite from Python to Racket while nobody was looking

Some tests needed custom checks inexpressable as just JSON! I needed a way to combine code and data!

Bad news, everyone???

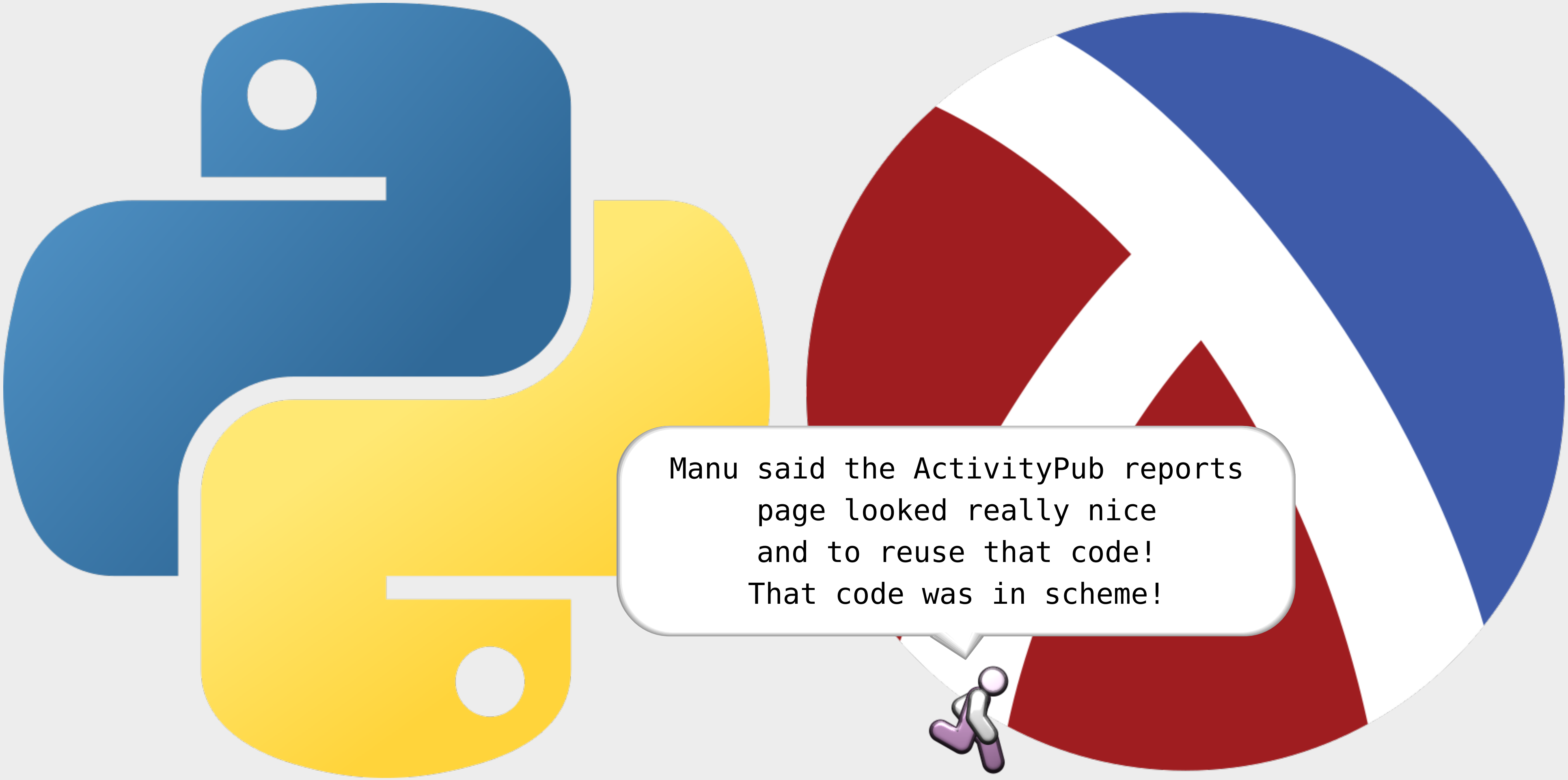
Chris reimplemented the test suite from Python to Racket while nobody was looking



You can still define simple tests as plain JSON!

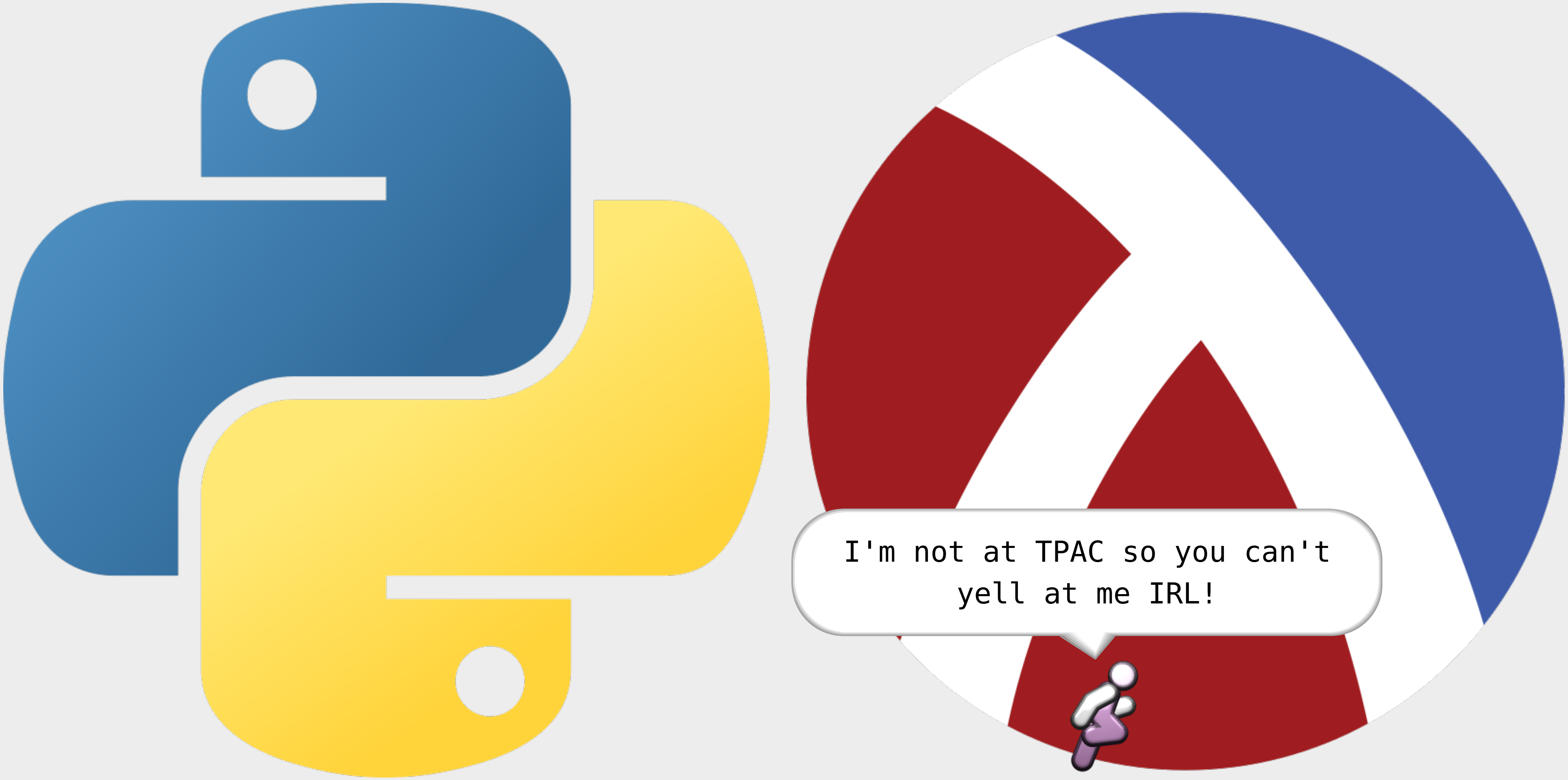
Bad news, everyone???

Chris reimplemented the test suite from Python to Racket while nobody was looking


Manu said the ActivityPub reports page looked really nice and to reuse that code! That code was in scheme!

Bad news, everyone???

Chris reimplemented the test suite from Python to Racket while nobody was looking

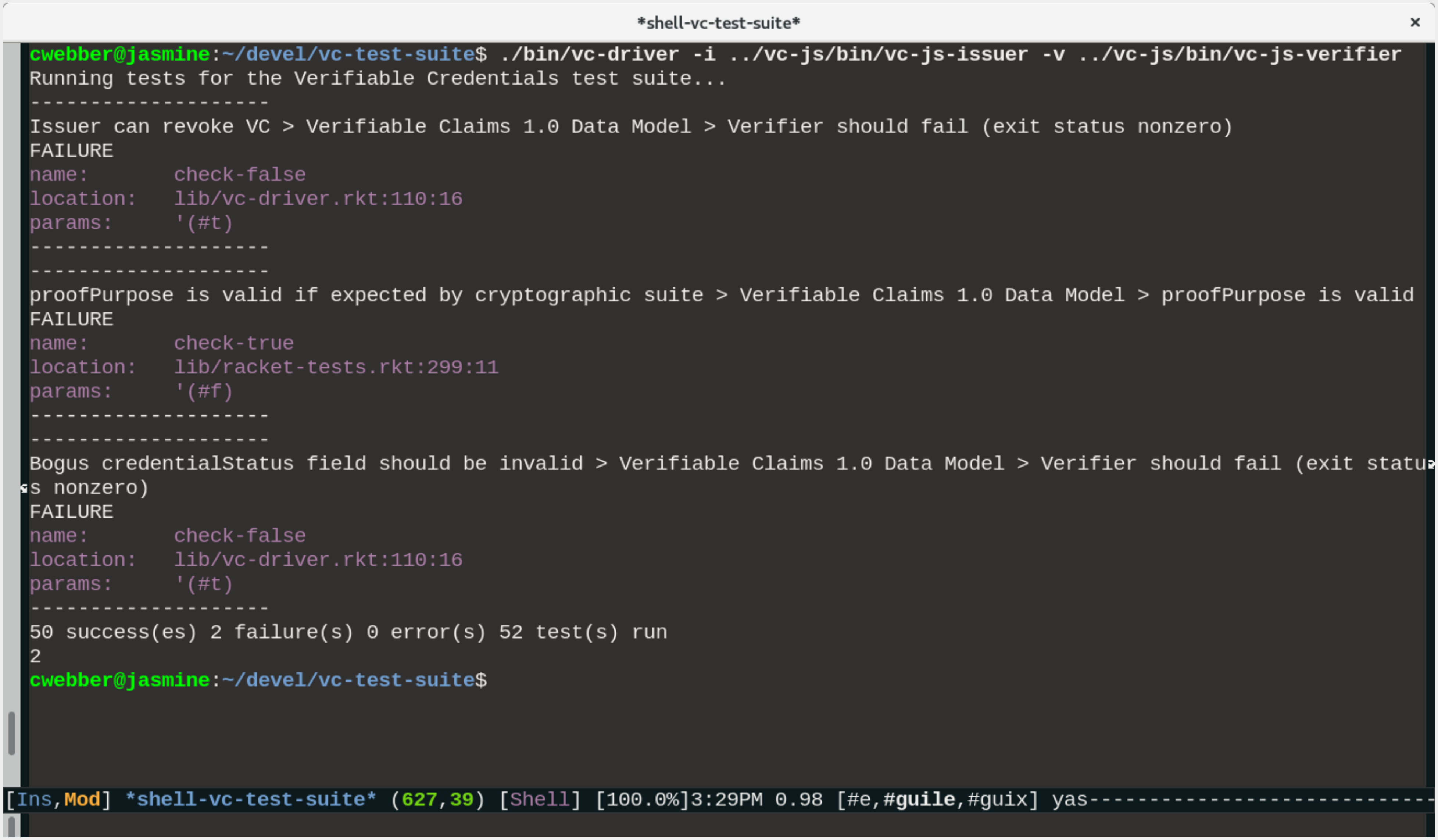


Enough tomfoolery, let’s see it in action!

Yeah! Okay we're gonna need:

- Racket (damnit, Chris)
- Do a git clone of https://github.com/w3c/vc-data-model/
- Have a VC implementation handy! It’ll need two binaries:
    - One to issue VCs
    - One to verify VCs
- If you have two different implementations, you can test them against each other!

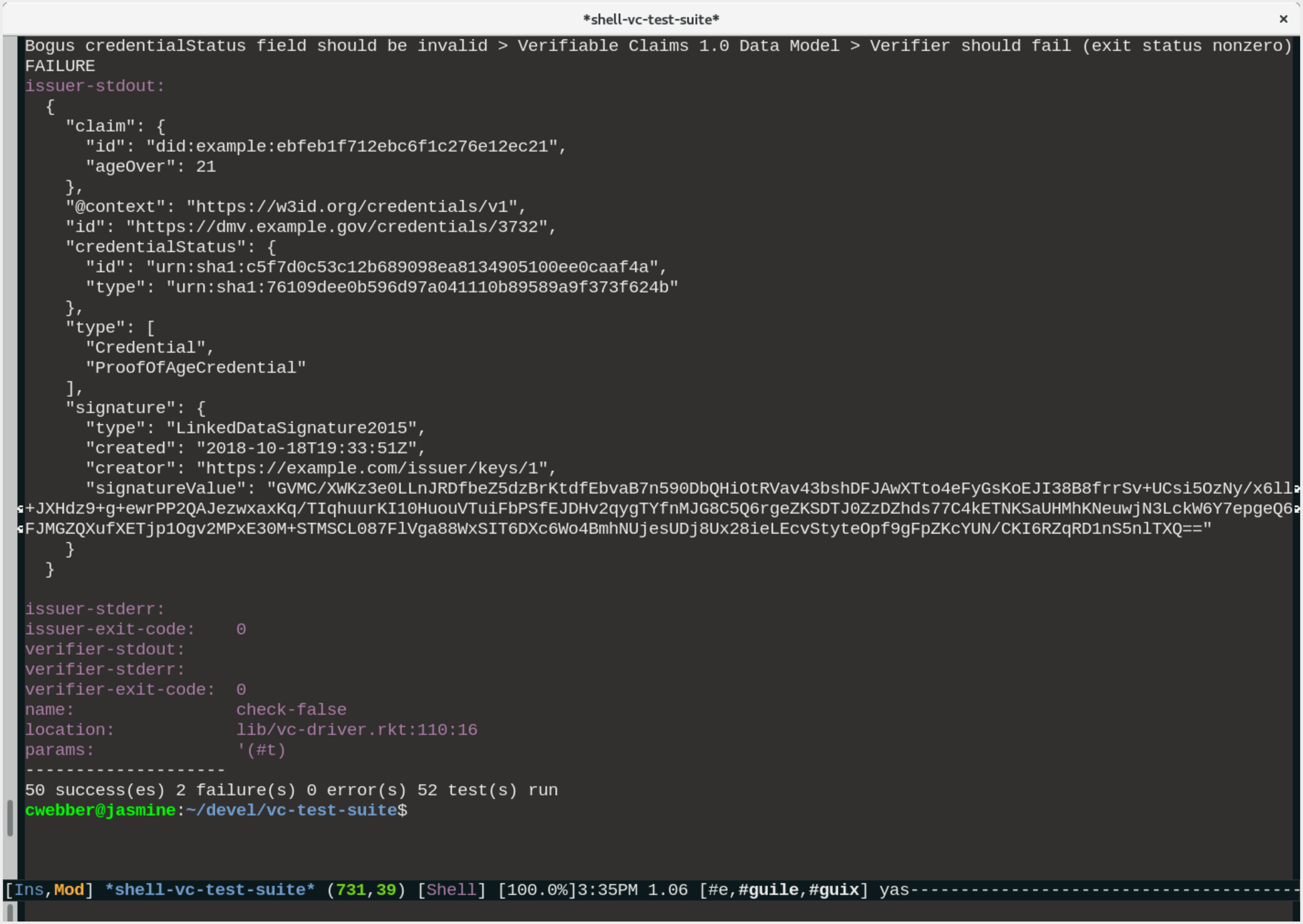
```
$ ./bin/vc-driver -i ../vc-js/bin/vc-js-issuer \
                  -v ../vc-js/bin/vc-js-verifier
```



```
cwebber@jasmine:~/devel/vc-test-suite$ ./bin/vc-driver -i ../vc-js/bin/vc-js-issuer -v ../vc-js/bin/vc-js-verifier
Running tests for the Verifiable Credentials test suite...
--------------------
Issuer can revoke VC > Verifiable Claims 1.0 Data Model > Verifier should fail (exit status nonzero)
FAILURE
name:       check-false
location:   lib/vc-driver.rkt:110:16
params:     '(#t)
--------------------
--------------------
proofPurpose is valid if expected by cryptographic suite > Verifiable Claims 1.0 Data Model > proofPurpose is valid
FAILURE
name:       check-true
location:   lib/racket-tests.rkt:299:11
params:     '(#f)
--------------------
--------------------
Bogus credentialStatus field should be invalid > Verifiable Claims 1.0 Data Model > Verifier should fail (exit statu
s nonzero)
FAILURE
name:       check-false
location:   lib/vc-driver.rkt:110:16
params:     '(#t)
--------------------
50 success(es) 2 failure(s) 0 error(s) 52 test(s) run
2
cwebber@jasmine:~/devel/vc-test-suite$
```

```
[Ins,Mod] *shell-vc-test-suite* (627,39) [Shell] [100.0%]3:29PM 0.98 [#e,#guile,#guix] yas---------------------------
```

--verbose

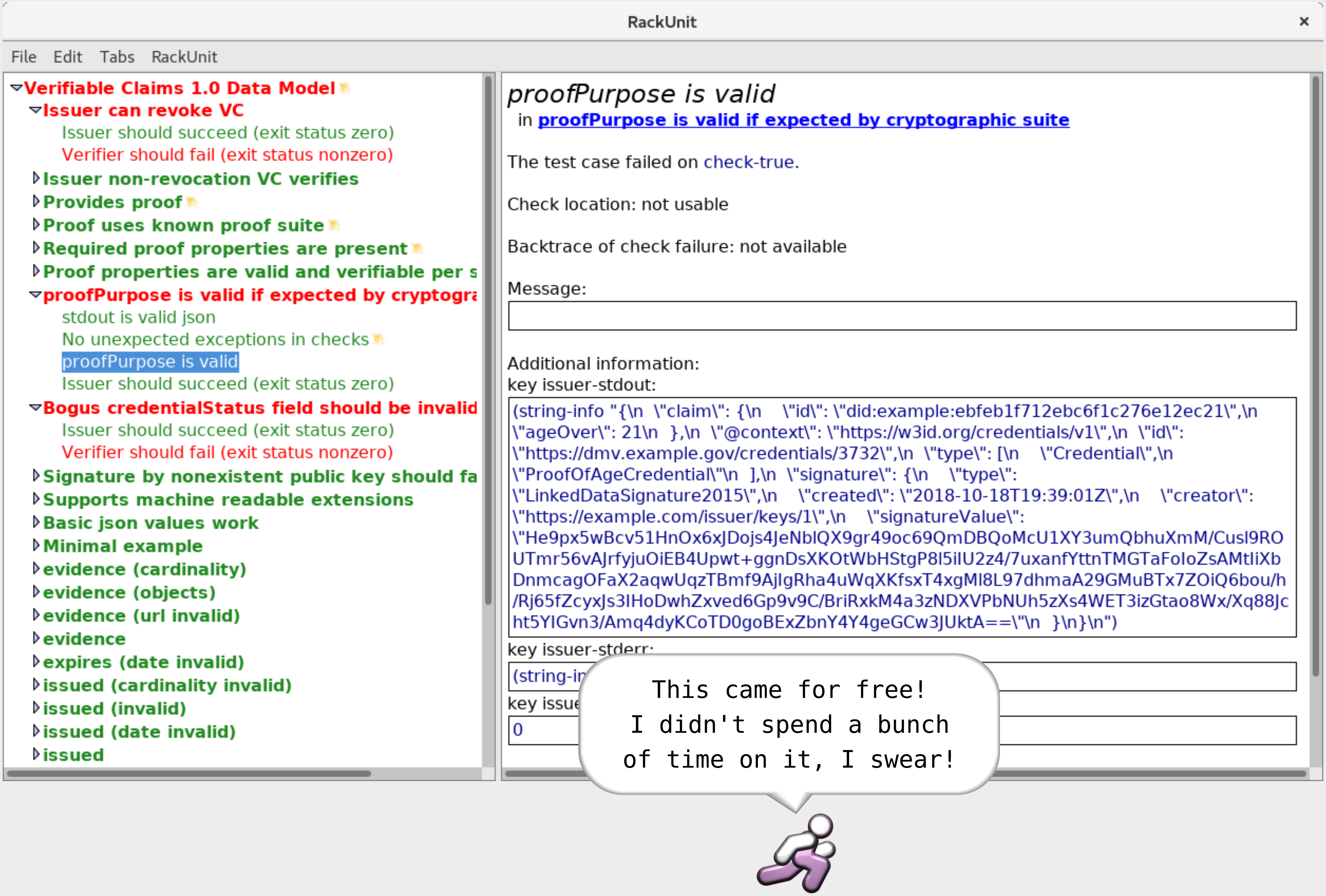


```
Bogus credentialStatus field should be invalid > Verifiable Claims 1.0 Data Model > Verifier should fail (exit status nonzero)
FAILURE
issuer-stdout:
  {
    "claim": {
      "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
      "ageOver": 21
    },
    "@context": "https://w3id.org/credentials/v1",
    "id": "https://dmv.example.gov/credentials/3732",
    "credentialStatus": {
      "id": "urn:sha1:c5f7d0c53c12b689098ea8134905100ee0caaf4a",
      "type": "urn:sha1:76109dee0b596d97a041110b89589a9f373f624b"
    },
    "type": [
      "Credential",
      "ProofOfAgeCredential"
    ],
    "signature": {
      "type": "LinkedDataSignature2015",
      "created": "2018-10-18T19:33:51Z",
      "creator": "https://example.com/issuer/keys/1",
      "signatureValue": "GVMC/XWKz3e0LLnJRDfbeZ5dzBrKtdfEbvaB7n590DbQHiOtRVav43bshDFJAwXTto4eFyGsKoEJI38B8frrSv+UCsi5OzNy/x6ll
+JXHdz9+g+ewrPP2QAJezwxaxKq/TIqhuurKI10HuouVTuiFbPSfEJDHv2qygTYfnMJG8C5Q6rgeZKSDTJ0ZzDZhds77C4kETNKSaUHMhKNeuwjN3LckW6Y7epgeQ6
FJMGZQXufXETjp1Ogv2MPxE30M+STMSCL087FlVga88WxSIT6DXc6Wo4BmhNUjesUDj8Ux28ieLEcvStyteOpf9gFpZKcYUN/CKI6RZqRD1nS5nlTXQ=="
    }
  }

issuer-stderr:
issuer-exit-code:    0
verifier-stdout:
verifier-stderr:
verifier-exit-code:  0
name:                check-false
location:            lib/vc-driver.rkt:110:16
params:              '(#t)
--------------------
50 success(es) 2 failure(s) 0 error(s) 52 test(s) run
cwebber@jasmine:~/devel/vc-test-suite$

[Ins,Mod] *shell-vc-test-suite* (731,39) [Shell] [100.0%]3:35PM 1.06 [#e,#guile,#guix] yas--------------------------------------
```

--verbose --mode gui



- Verifiable Claims 1.0 Data Model
  - Issuer can revoke VC
    - Issuer should succeed (exit status zero)
    - Verifier should fail (exit status nonzero)
  - Issuer non-revocation VC verifies
  - Provides proof
  - Proof uses known proof suite
  - Required proof properties are present
  - Proof properties are valid and verifiable per s
  - proofPurpose is valid if expected by cryptogra
    - stdout is valid json
    - No unexpected exceptions in checks
    - proofPurpose is valid
    - Issuer should succeed (exit status zero)
  - Bogus credentialStatus field should be invalid
    - Issuer should succeed (exit status zero)
    - Verifier should fail (exit status nonzero)
  - Signature by nonexistent public key should fa
  - Supports machine readable extensions
  - Basic json values work
  - Minimal example
  - evidence (cardinality)
  - evidence (objects)
  - evidence (url invalid)
  - evidence
  - expires (date invalid)
  - issued (cardinality invalid)
  - issued (invalid)
  - issued (date invalid)
  - issued

proofPurpose is valid
in proofPurpose is valid if expected by cryptographic suite

The test case failed on check-true.

Check location: not usable

Backtrace of check failure: not available

Message:

Additional information:
key issuer-stdout:

```
(string-info "{\n  \"claim\": {\n    \"id\": \"did:example:ebfeb1f712ebc6f1c276e12ec21\",\n
\"ageOver\": 21\n  },\n  \"@context\": \"https://w3id.org/credentials/v1\",\n  \"id\":
\"https://dmv.example.gov/credentials/3732\",\n  \"type\": [\n    \"Credential\",\n
\"ProofOfAgeCredential\"\n  ],\n  \"signature\": {\n    \"type\":
\"LinkedDataSignature2015\",\n    \"created\": \"2018-10-18T19:39:01Z\",\n    \"creator\":
\"https://example.com/issuer/keys/1\",\n    \"signatureValue\":
\"He9px5wBcv51HnOx6xJDojs4JeNblQX9gr49oc69QmDBQoMcU1XY3umQbhuXmM/Cusl9RO
UTmr56vAJrfyjuOiEB4Upwt+ggnDsXKOtWbHStgP8l5ilU2z4/7uxanfYttnTMGTaFoloZsAMtliXb
DnmcagOFaX2aqwUqzTBmf9AjlgRha4uWqXKfsxT4xgMl8L97dhmaA29GMuBTx7ZOiQ6bou/h
/Rj65fZcyxJs3lHoDwhZxved6Gp9v9C/BriRxkM4a3zNDXVPbNUh5zXs4WET3izGtao8Wx/Xq88Jc
ht5YlGvn3/Amq4dyKCoTD0goBExZbnY4Y4geGCw3JUktA==\"\n  }\n}\n")
```

key issuer-stderr:

(string-in

key issue

0

This came for free!
I didn't spend a bunch
of time on it, I swear!

Finally, to export a report

```
--mode report
```

Not much to say; exports a json document with your test reports to stdout. Save it, change your project’s name, submit it.

Gregg Kellogg demands more EARL output

Coming soon, Gregg!

Writing a simple test, using JSON:

First add to the manifest:

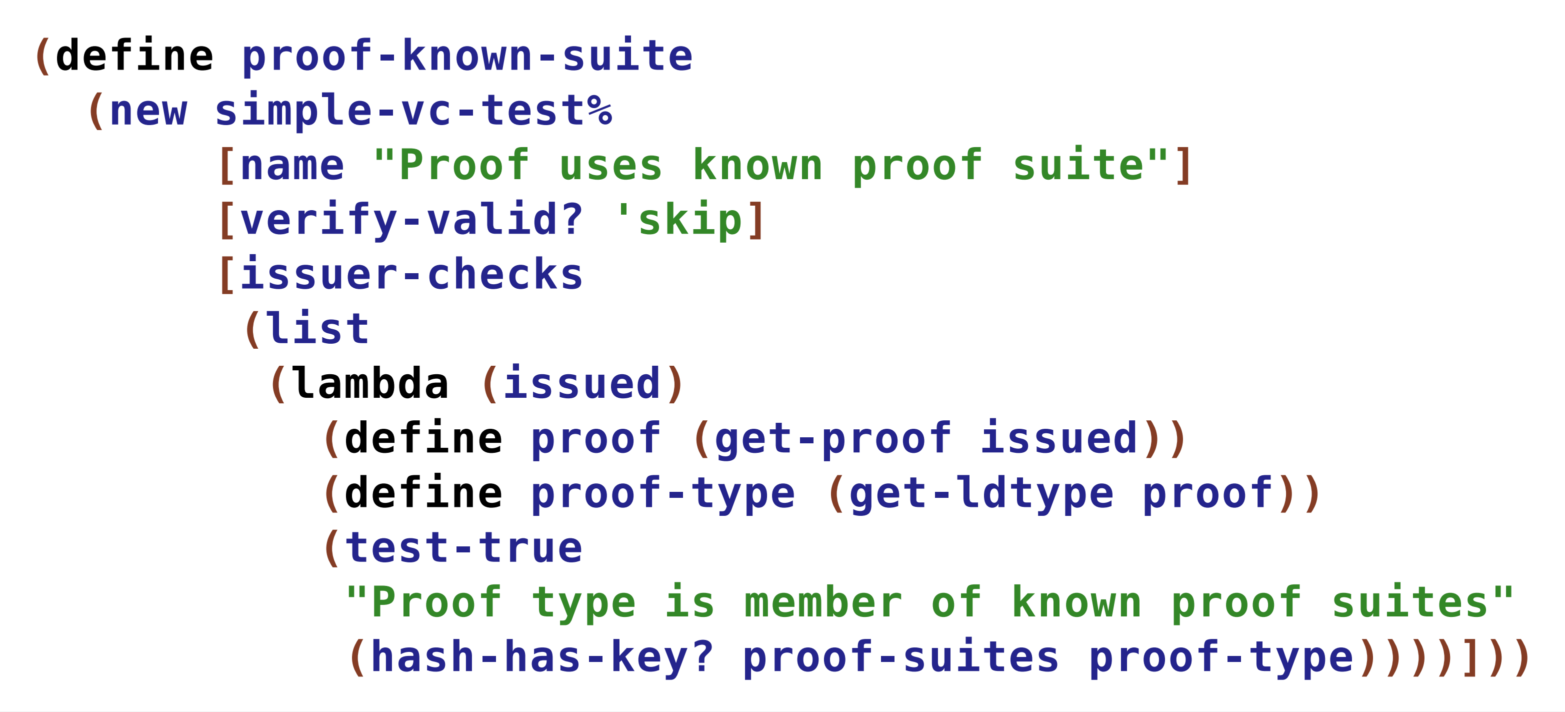
```
{
  "test": [
    // ...
    {
      "name": "evidence (url invalid)",
      "extend": "default",
      "file": "tests-1.0/evidence-url-invalid.jsonld",
      "issueValid": false
    }
   // ...
  ]
}
```

Writing a simple test, using JSON:

Next add the test file:

```json
{
  "@context": "https://w3id.org/credentials/v1",
  "id": "https://dmv.example.gov/credentials/3732",
  "type": ["Credential", "ProofOfAgeCredential"],
  "claim": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "ageOver": 21
  },
  "evidence": "INVALID_URL"
}
```

Writing a Racket-based test for custom checks



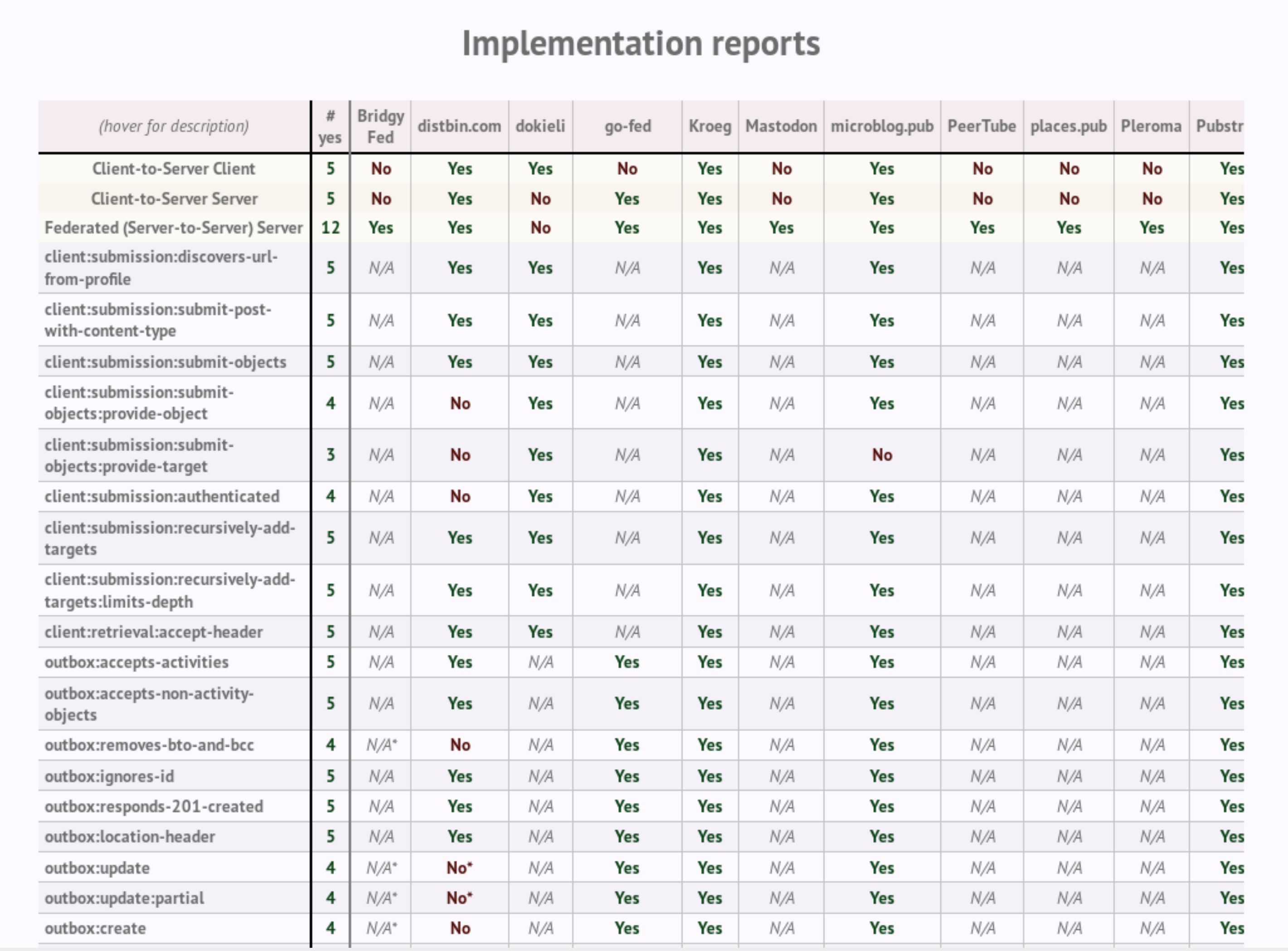
```
(define proof-known-suite
  (new simple-vc-test%
       [name "Proof uses known proof suite"]
       [verify-valid? 'skip]
       [issuer-checks
        (list
         (lambda (issued)
           (define proof (get-proof issued))
           (define proof-type (get-ldtype proof))
           (test-true
            "Proof type is member of known proof suites"
            (hash-has-key? proof-suites proof-type))))]))
```

Damnit, Chris

- Many tests can be written as just JSON
- Need to write custom tests and you don’t want to write Racket?
  - Chris has some code where you can run checks from a Python script
  - It’s a lot more boilerplate though

Implementation reports page?

Well we don’t have reports yet, but here’s what ActivityPub’s implementation reports page looks like:



Implementation reports

(hover for description)	# yes	Bridgy Fed	distbin.com	dokieli	go-fed	Kroeg	Mastodon	microblog.pub	PeerTube	places.pub	Pleroma	Pubstr
Client-to-Server Client	5	No	Yes	Yes	No	Yes	No	Yes	No	No	No	Yes
Client-to-Server Server	5	No	Yes	No	Yes	Yes	No	Yes	No	No	No	Yes
Federated (Server-to-Server) Server	12	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
client:submission:discovers-url-from-profile	5	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
client:submission:submit-post-with-content-type	5	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
client:submission:submit-objects	5	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
client:submission:submit-objects:provide-object	4	*N/A*	No	Yes	*N/A*	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
client:submission:submit-objects:provide-target	3	*N/A*	No	Yes	*N/A*	Yes	*N/A*	No	*N/A*	*N/A*	*N/A*	Yes
client:submission:authenticated	4	*N/A*	No	Yes	*N/A*	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
client:submission:recursively-add-targets	5	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
client:submission:recursively-add-targets:limits-depth	5	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
client:retrieval:accept-header	5	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:accepts-activities	5	*N/A*	Yes	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:accepts-non-activity-objects	5	*N/A*	Yes	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:removes-bto-and-bcc	4	*N/A**	No	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:ignores-id	5	*N/A*	Yes	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:responds-201-created	5	*N/A*	Yes	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:location-header	5	*N/A*	Yes	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:update	4	*N/A**	No*	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:update:partial	4	*N/A**	No*	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes
outbox:create	4	*N/A**	No	*N/A*	Yes	Yes	*N/A*	Yes	*N/A*	*N/A*	*N/A*	Yes

Image credits

- "Sir Not Appearing In this Film" from Monty Python's Quest for the Holy Grail
- Images from last year's presentation courtesy Manu

Questions?

```
           ,.  ,-.          _--------_
         .'.-.-.-.-       .' Hope you '.
       ..-..-,.,.-..--   |   Have fun   |
      ,.-.    '   ' '    '.  at TPAC!  .'
     '.,,- __---   --      '--.   .---'
      .-(\\  [ o]-[ o]         ) )
       ', ||     _) |       --=-'
         | ||   ___ /
     ___-/    _____/
   ./   |\      |
  /     | \ _   .\\
 |   ,  |__| -'' |\\
|    |     |      \ |
|    |      |      ||
|    |      |      ||
```