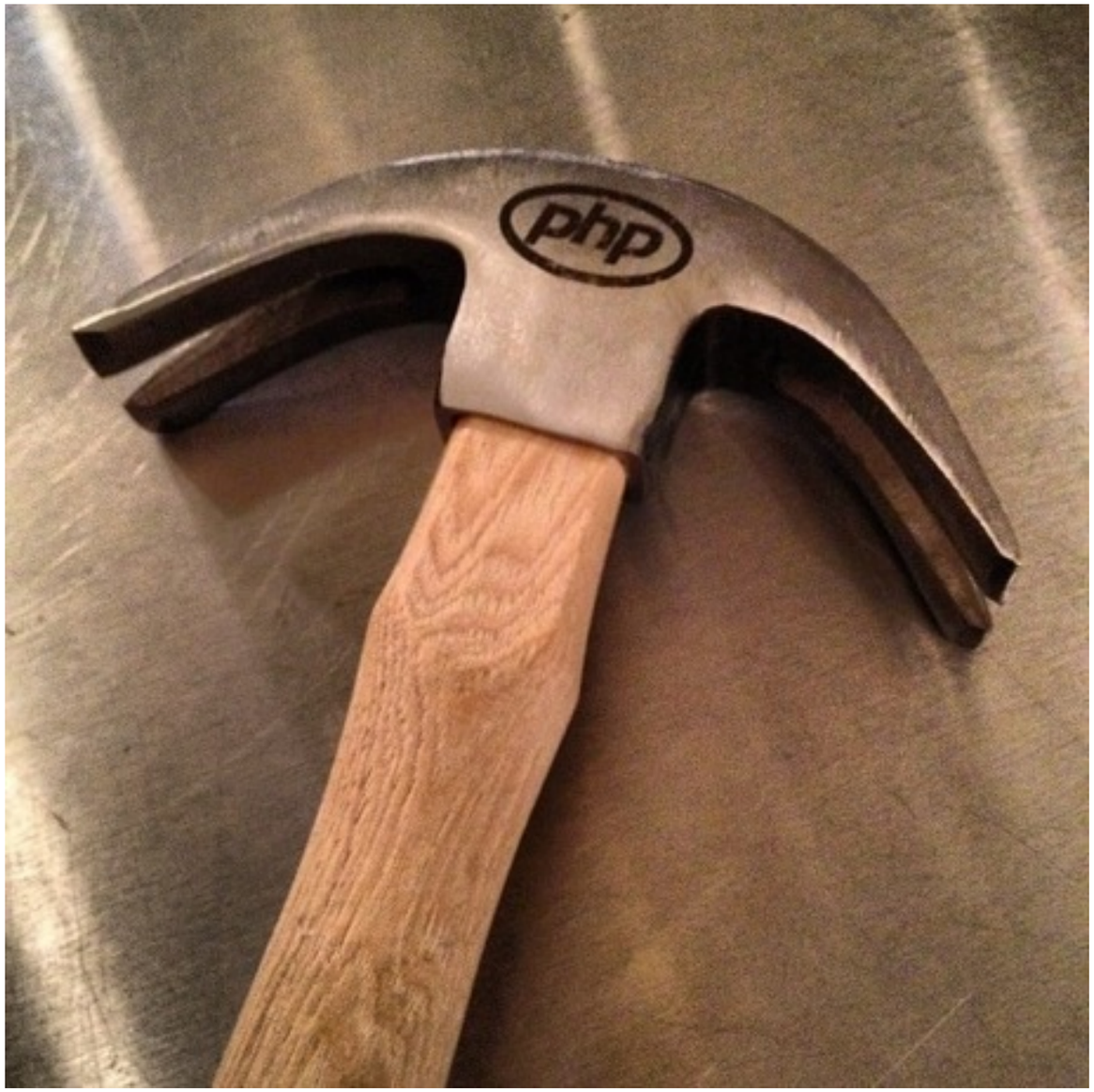
Mobile Authentication

One Rack-App At A Time

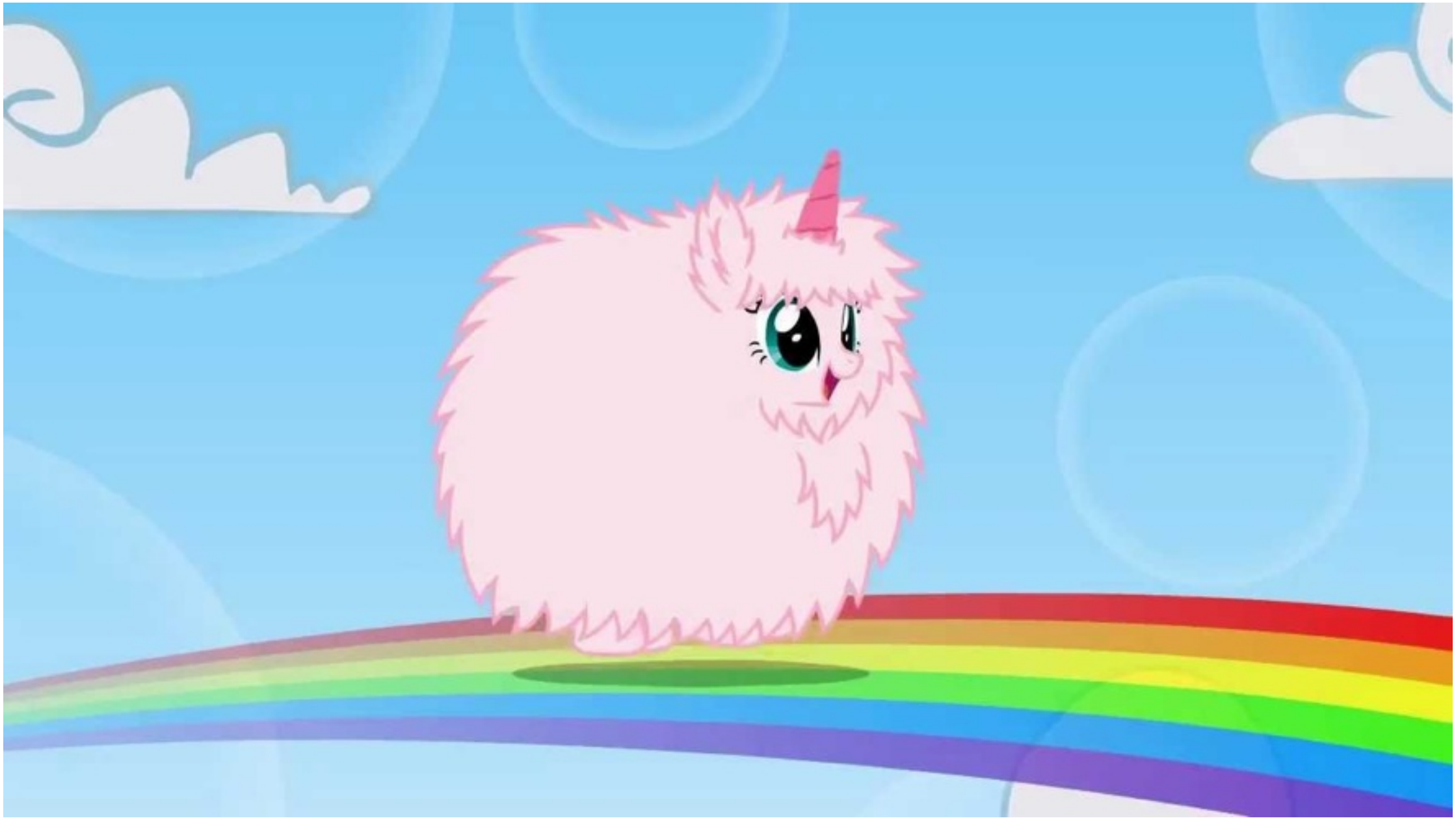
DONAL ELLIS

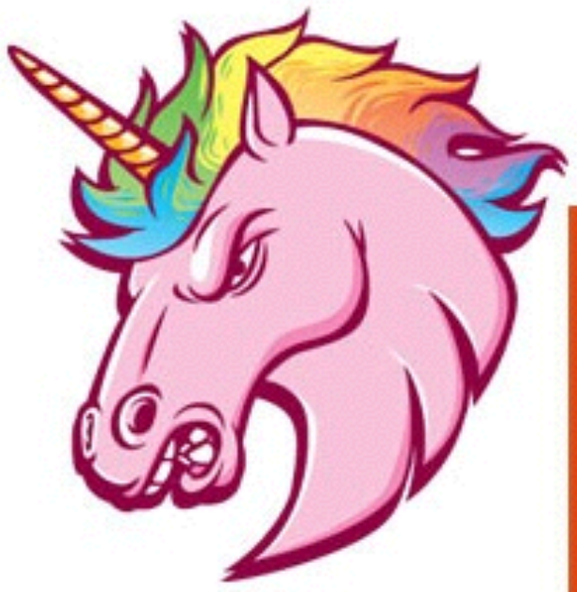
twitter?
facebook?
donal@getperx.com

From…



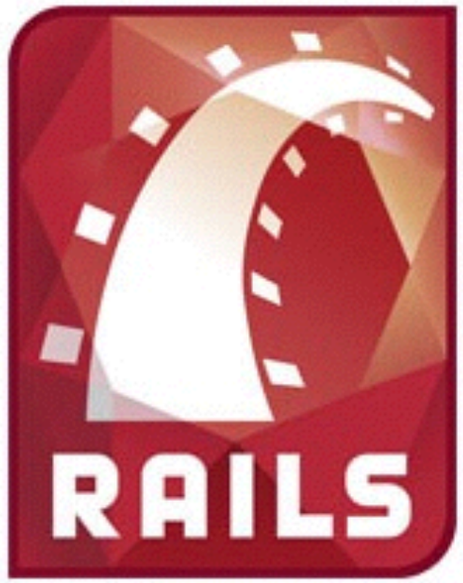
To…





NGINX

ubuntu



Web App for Mobiles

- API only (JSON)
- No Sessions
    - Every request must authenticate itself

Authentication

- Identify:
  - User (`consumer_key`)
  - Device (`udid`)

Secure Against?

- Very smart people
  - Write a script to mimic requests
    - GIVE ME ALL THE CHOPS
- Man-in-the-middle attacks
  - Really?

OAuth1.0a

- http://oauth.net/core/1.0a/
- Nonce
- Timestamp
- Signature
- `consumer_key`
- (No `udid`)

Where’s the OAuth?

- Users can’t grant/remove permission

OAuth1.0a vs. 2.0

- 1.0a
  - requires signatures
- 2.0
  - requires SSL
- Does SSL solve our security requirements?
  - No client SSL authentication

API: login/register

- Send your credentials
- Send OAuth data in **`Accept`** header:
  - **`nonce, timestamp, udid`**
- (You could send **`udid`** in **`User-Agent`** header)
- Get back **`consumer_key`**

API: all other endpoints

- Send OAuth data:
  - `nonce, timestamp, udid, app_key`
- Plus send `consumer_key`
- Get back (whatever you ask for)

Client Side: Signature

- `secret_key` ships with phone app
- Build parameter string (see OAuth1.0a protocol) from outgoing data
  - Both `Authorization` header and request params
- Use `secret_key` as input to HMAC-SHA1 to encrypt parameter string
- Include signature in `Authorization` header

Server-Side: Signature

- Build parameter string (see OAuth1.0a protocol) from incoming data
  - Both **`Authorization`** header and request params
- Use **`secret_key`** (stored on server) as input to HMAC-SHA1 to encrypt parameter string
- Compare signatures

Server-Side: Timestamp

- “The timestamp value MUST be a positive integer and MUST be equal or greater than the timestamp used in previous requests.”

- Use Redis GET/SET

  - `request_timestamp:consumer_key:<123>=<timestamp>`

Server-Side: Nonce

- “…a Nonce value that is unique for all requests with that timestamp”
- Use Redis SADD (a set with fixed duration)
  - `nonces:consumer_key:<123>:timestamp:<timestamp>=<nonce>`

Rack

- A protocol for an interface
  - Between Ruby web servers and applications
- https://github.com/rack/rack
  - There is code…but that's not important right now.

Rack.call(env)

- Must return array of:
  - Status
  - Hash of HTTP headers
  - Object that responds to **`#each`** – the response body

Rails Rack

- http://guides.rubyonrails.org/rails_on_rack.html

A Middleware For Everyone

- Check `Authorization` Token
- Check Nonce
- Check Timestamp
- Check username/password (Register vs. Login)
- Check Signature

Warden

- Proxy injected into request
  - `env['warden']`
- Available to downstream middleware/apps
- Strategies
- Is it worth it?
  - Flexible, structure vs. early response

DEMO

1. Rack App
2. Rack Middleware
3. Security Middleware/Chain
4. Checking Routes
5. Warden
6. Rails and Rack

Conclusion

- Rack is widely used in Ruby
- Rack is an integral part of Rails (and Sinatra)
- Devise is built on Warden (built on Rack)
- Mobiles complicate web development
  - But also simplify!
- Know your technologies, know your protocols!

Extras

- https://github.com/superfeedr/rack-superfeedr/pull/2