

# ***Phoenix and Rails Authentication***

Introducing Phoenix Auth APIs to a diverse ecosystem.

***Hi! I'm Andrew.***

I work here.

# ***I'm a bike commuter***

A while ago I built a bunch of little tools to track where I was going  
on my bike.

# *Sooo many different things*

It's kind of a mess:

- GPS track ingestion in Node and JS, Mongo
- Visualization in Ruby on Rails
- Storage in PostgreSQL
- Authentication & identity in... TBD

# What the app is



# *I know, I'll use Elixir!*

Idea: What if I introduced Elixir into my project as an identity service?

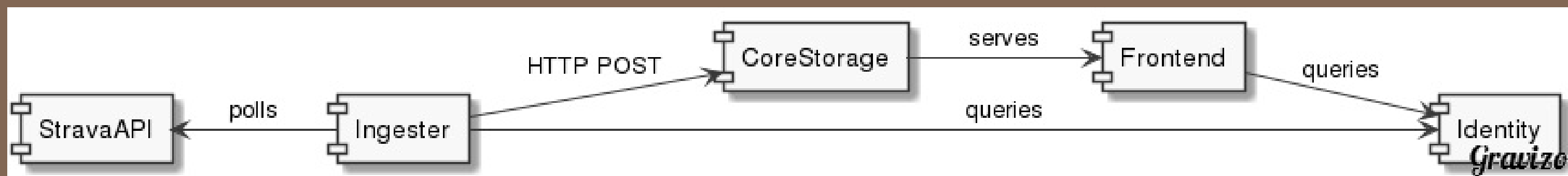
Responsibilities:

- Authentication
- Authorization (TBD)

***Fell in love!***

# Desired architecture

Introduce an identity system, which will store the list of users and their tokens - and manage sessions, too!





# Step 1: Phoenix app from scratch

Played with Ueberauth

- Wrote a plugin: ueberauth\_strava
- Wrote it inside my Elixir app, then extracted into its own hex package.
- Ueberauth is kind of like OmniAuth

# *Demo*

See: Ueberauth code

# *Step 1, done:*

At this point, the app can log you in (SSO) with Strava, and find (or create) a user account. It also stores a token.

# *Step 2: Research authentication*

Ueberauth is closely aligned with Guardian, which pushes you to use JWT (JSON Web Tokens) as an auth and session mechanism.

# *JWT, briefly.*

[www.jwt.io](http://www.jwt.io)

JSON object that stores:

- Claims (authorizations, permissions)
- Signatures, tokens
- Expiry times

Store it in:

Cookie? Local Storage?

# Step 2, findings:

Hm, that might not be for me. Why not?

- Session expirations complicated
- Complex implementation
- Overkill - this is just a side project!

"Stop Using JWT For Sessions"

# *Step 3: Rails and Phoenix session sharing!*

Rails and Phoenix share parallel implementations of the Rails session serialization and deserialization code. Stored in a cookie.

# *Step 3: Rails and Phoenix session sharing!*

Rails and Phoenix share parallel implementations of the Rails session serialization and deserialization code. Stored in a cookie.

I wrote a blog post on this: [Rails, Meet Phoenix](#)



# *How to do this:*

Set up Phoenix and Rails with the same:

- SECRET\_KEY
- cookie name prefix
- cookie salt (encrypted, and signing salt)

Then add a plug library PlugRailsCookieSessionStore

***Tada!***

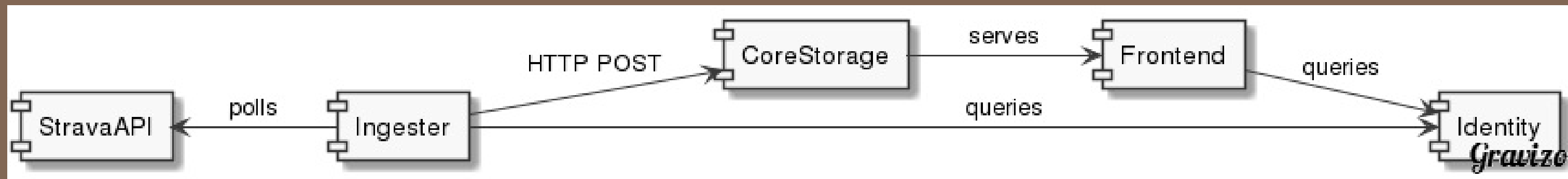
# ***Finally: open a Users API***

Internal apps can access it to get a list of users and their tokens.

GET /users

Simple Bearer-Token auth, protected over SSL.

# Sooooo...



***Which brings us  
here...***

[cyclecity.io](https://cyclecity.io)

# Takeaways

- Get started with Elixir however you can.
- Just because it's shiny.. doesn't mean you have to use it!

# *Thanks!*

Track your rides! [cyclecity.io](https://cyclecity.io)

Me:

[andrew@carbonfive.com](mailto:andrew@carbonfive.com)

[twitter.com/@andrewhao](https://twitter.com/@andrewhao)

[github.com/andrewhao](https://github.com/andrewhao)