

The Network Times

Ian Turner

June 21, 2024

1 Full Stack Rust Rewrite Loading

i love leptos so this app will be my way to become cracked at rust. it's in different [git repo](#) btw (too lazy to update readme in other repo)

2 Bounty Submission for Balajis on Farcaster

Balaji put out a **cast** on Wednesday the 6th challenging anyone to create an AI NYT. He provided a nice example of something working **18 months ago** with inferior LLM models. I've been looking for a reason to stay up all night coding, so I figured I'd give it a shot.

Why \LaTeX Ian?

Great question anon! I write these notes everyday at work and on other projects since pmarca gave me the idea with his ANTI-TODO list concept. Blog is no longer on the internet or else I would link it here. This may be completely useless to people, but I don't really but much detail in my git commits (I should start) so maybe this will help if you end up contributing.

3 Week 1 - Ship Website ASAP

Friday, 3/8/2024

- Recruit 10x designer, Michael Raisch.
- Set ship or die goal to Sunday (Didn't know the deadline at the time).
- Install hubble and run node.
- Get basic views built.
- Rust server which makes API calls to various LLM models.

Saturday, 3/9/2024

- Connect rust server api calls to client components.
- Struggle with hosting server.

Sunday, 3/10/2024

- Launch at 6AM rofl.

4 Week 2 - Feature Maxing

Monday, 3/11/2024

- Settle on OVH server to host and migrate away from AWS.

Tuesday, 3/12/2024

- Get Claude 3 Opus API working for article gen on photo click.

Wednesday, 3/13/2024

- Redesign by another 10X designer, Elvia Franco.

Thursday, 3/14/2024

- Skim through [1] for the 4th time.
- Initialize this very document (meta).
- Start to implement redesign.
- redesign nearly complete, need to style tweets still.

Friday, 3/15/2024

- Adjust system prompt, use haiku to save money rofl.
- Add L^AT_EX pdf for change log button.

Saturday, 3/16/2024

- See Figure 1.



Figure 1: *ahhh*

Sunday, 3/16/2024

- See Figure 1.

5 Week 3 - AHHHHH

Monday, 3/11/2024

- See Figure 2.



Figure 2: *AHHHH*

Tuesday, 3/12/2024

- See Figure 3.

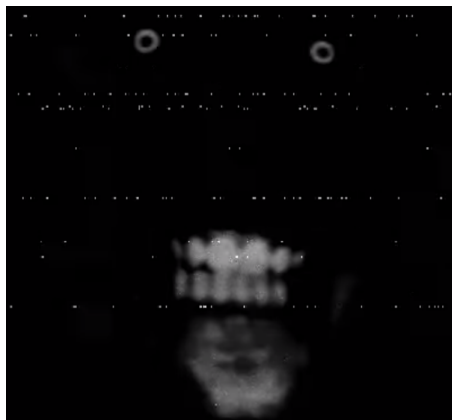


Figure 3: *AHHHHHHHHHHHHHHH*

Wednesday, 3/13/2024

- Store articles in local storage to prevent multiple API calls per channel per session. Still very clunky behavior with NavBar versus logos on ArticleList.

6 Week 4 SEE WEEK 5

7 Week 5

I can't be asked

- I am too lazy to summarize my shitty git commits from the 13 til now (the 31st).
- Enjoy Mr. Incredible Uncanny 4 instead:



Figure 4: *oooooooooooooooooooooooo*

- Right, now I've added a few things:
 - custom link style and formatting in casts
 - render images in casts
 - sign in with farcaster (doesn't do much else yet)
 - link to view cast on warpcast
- I suppose I should add reaction data and stuff now.
- I'm avoiding the actual hard tech of having the LLM not hallucinate. I'm thinking of having a writers room where my proprietary spaghetti code strips out the things that don't make logical sense from the summary from the LLM.

I really can't be asked

- I suck at updating this, but I'm focused on a native farcon app at the moment. you can check it out [here](#).
- I'm trying to get messaging working by handcrafting endpoints in rust (this will be my downfall, I should just use Neynar) so I'll use them in both apps anyways.
- Figure 5 is a drawing (from farcaster YT tutorial) which shows how to create a message with a custom client application.

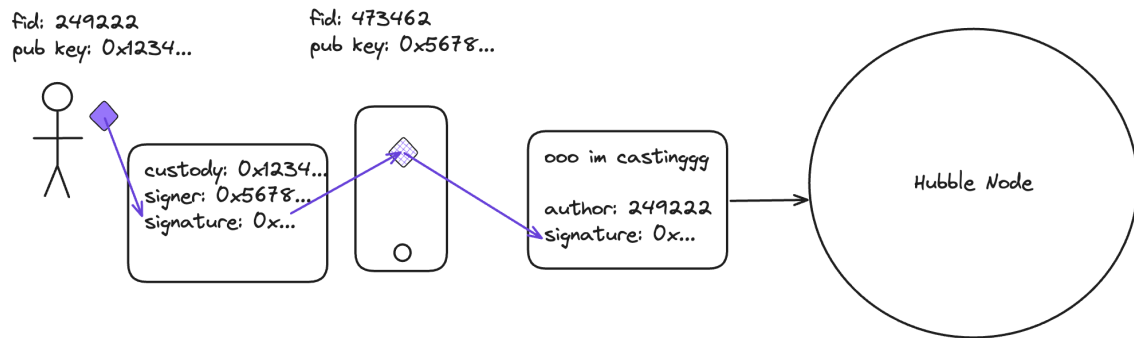


Figure 5: *Farcaster Signer Concept*

- pinata, farcasterindex, things to thing about using recommended by a friend.

8 Week 7?

Saturday, 04/20/2024

- Q is helping me with ai nyt server.
- I am hard stuck on messages so perhaps we will crack it together.

Sunday, 04/21/2024

- Got diesel initialized in the project for storing articles, have not yet tested the ORM yet, I'll do this soon.
- I was distracted by the reactions feature for the network times. This was really easy to add, and I styled it indigo-300 or something like this. I've never seen a social site have this color like, so it's interesting at least.
- You can't yet press the like, this will work once I finally figure out how to make the app a valid signer for users.
- I added a profile page as shown in Figure 6.

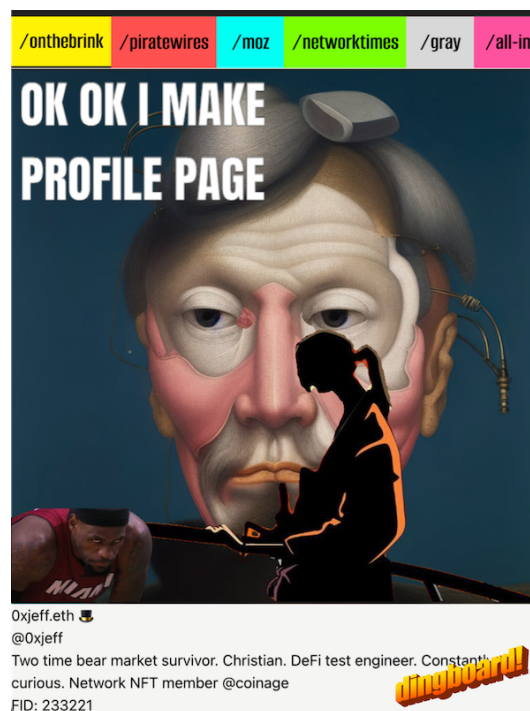


Figure 6: Profile Page

- I added hover effects (you can't see it here but matt's pfp background has a slight black circle with 10% opacity applied here (Figure 7)).

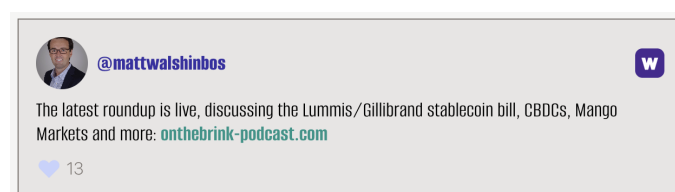


Figure 7: Pfp Hover Effect

9 Week 8?

Sunday, 06/10/2024

- we might be back (leptos rewrite). notes from testing existing hubble node routes with leptos components.
- this is not live yet since it has very limited functionality.
- `fetch_username` checks if `lead_usernames` already has the `fid`. if not, it fetches the username and updates the state.
- added `ongoing_requests` to track fetches and avoid multiple requests for the same `fid`.
- Used `create_effect` to manage side effects, ensuring usernames are fetched only once.
- Used `spawn_local` for async tasks to keep the main thread non-blocking.
- `Signal` and `set_signal` handle reactive state in the `Channels` component, making sure the UI updates when data changes.
- `HashSet` tracks `ongoing_requests`, preventing duplicate fetches and redundant state updates.
- The component dynamically displays usernames from the updated state.
- asdfasdf

Monday, 06/11/2024

- removed the `lead_username` variable and its assignment using `unwrap_or_else` from the `view!` macro.
- added closure inside the `view!` macro that matches on `lead_usernames.get().get(<ref>fid)`.
- if the lead username is available show it, else show "chill" in div where username would be
- closure is defined using `move ||` to capture the `lead_usernames` and `fid` variables.

Tuesday, 06/12/2024

- begin rewrite of cast and cast list pages, need to think about using `create_effect` in the complicated way that i did above.
- using this `create_effect` in the way that i am to sync the reactive system is *officially* (Figure 8) discouraged since you might shoot off your foot with an infinite loop if you don't create something to keep track of the ongoing requests. i like the chill message though and how it could show up at different times for different channels in some cases (probably won't be used, but we will see).
- i'll do the cast stuff without `create_effect` to see if it's less complicated and achieves virtually the same thing.

```
let (a, set_a) = create_signal(0);
let (b, set_b) = create_signal(0);

// ✓ use effects to interact between reactive state and the outside world
create_effect(move |_| {
  // immediately prints "Value: 0" and subscribes to `a`
  log::debug!("Value: {}", a.get());
});

set_a.set(1);
// ✓ because it's subscribed to `a`, the effect reruns and prints "Value: 1"

// ✗ don't use effects to synchronize state within the reactive system
create_effect(move |_| {
  // this technically works but can cause unnecessary re-renders
  // and easily lead to problems like infinite loops
  set_b.set(a.get() + 1);
});
```

Figure 8: *create effect bad*

Monday, 07/17/2024

Monday, 07/17/2024

- Tuesday, 07/18/2024

- Wednesday, 07/19/2024

- # deadpool-diesel

Dead simple async pool for diesel

by Alice Rychl, Michael P. Jung

<https://lib.rs Rust package>

```

Blocking waiting for file lock on package cache
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.36s
Compiling nwttr v0.1.0 (/home/eze/dev/nwttr)
  Cargo finished cargo build --package=nwttr --lib --target-dir=/home/eze/dev/nwttr/target
v/front --target=x86_32-unknown-unknown --no-default-features --features=hydrate
Front compiling WASM
Tailwind finished tailwind --input style/tailwind.css --config tailwind.config.js --output
et /home/eze/dev/nwttr/target/tmp/tailwind.css
Finished `dev` profile [unoptimized + debuginfo] target(s) in 1.65s
Cargo finished cargo build --package=nwttr --bin=nwttr --no-default-features --features
ssr

"active_model": "gpt-3.5-turbo-0125",
"active_lab": "openai"
},
failed to parse the request body as JSON: active_lab: expected value at line 6 column 26:ze
- $ curl -X POST http://localhost:3000/api/create_message -H "Content-Type: applicatio
n/json"
  {
    "thread_id": "63429007",
    "content": "ooooo in deadpoolingggg",
    "role": "user",
    "active_model": "gpt-3.5-turbo-0125",
    "active_lab": "openai"
  }
}

eze ~ $
[0] 1:notes 2:dev: 3:nvmin

```

fearless concurrency! :O

```

use cfg_if::cfg_if;
12 use cfg_if::cfg_if;
13
14 if cfg!(feature = "ssr") {
15     use diesel::prelude::*;
16     use diesel::PgConnection;
17     use deadpool_diesel::(Manager, Pool, Runtime);
18     use crate::models::conversations::(<NewMessage, Th
19     use crate::schema::(<threads, messages);
20
21     pub type DbPool = Pool<Manager<PgConnection>;>;
22
23     pub fn establish_connection(database_url: &str) -> DbPool {
24         let manager = Manager::new(database_url, Runtime::Tokio1);
25         Pool::builder(manager)
26             .max_size(8)
27             .build()
28             .expect("Failed to create pool.")
29     }
30
31     pub fn create_thread(conn: &mut PgConnection, new_thread: &Thread) -> Q
32         diesel::insert_into(<threads::table)
33             .values(<new_thread)
34             .execute(conn)
35     }
36
37     pub fn add_message(conn: &mut PgConnection, new_message: &NewMessage) ->
38         diesel::insert_into(<messages::table)
39             .values(<new_message)
40             .execute(conn)
41     }
42 }
43
44 src/database/db.rs
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Figure 9: *fearless concurrency !*

¹this will destroy me inside

Thursday, 07/20/2024

- ...

References

- [1] C. Huyen, Designing Machine Learning Systems. USA: O'Reilly, 2022.