

# Designing Real-time Web Apps

Tim Mecklem

VP of Engineering & Delivery



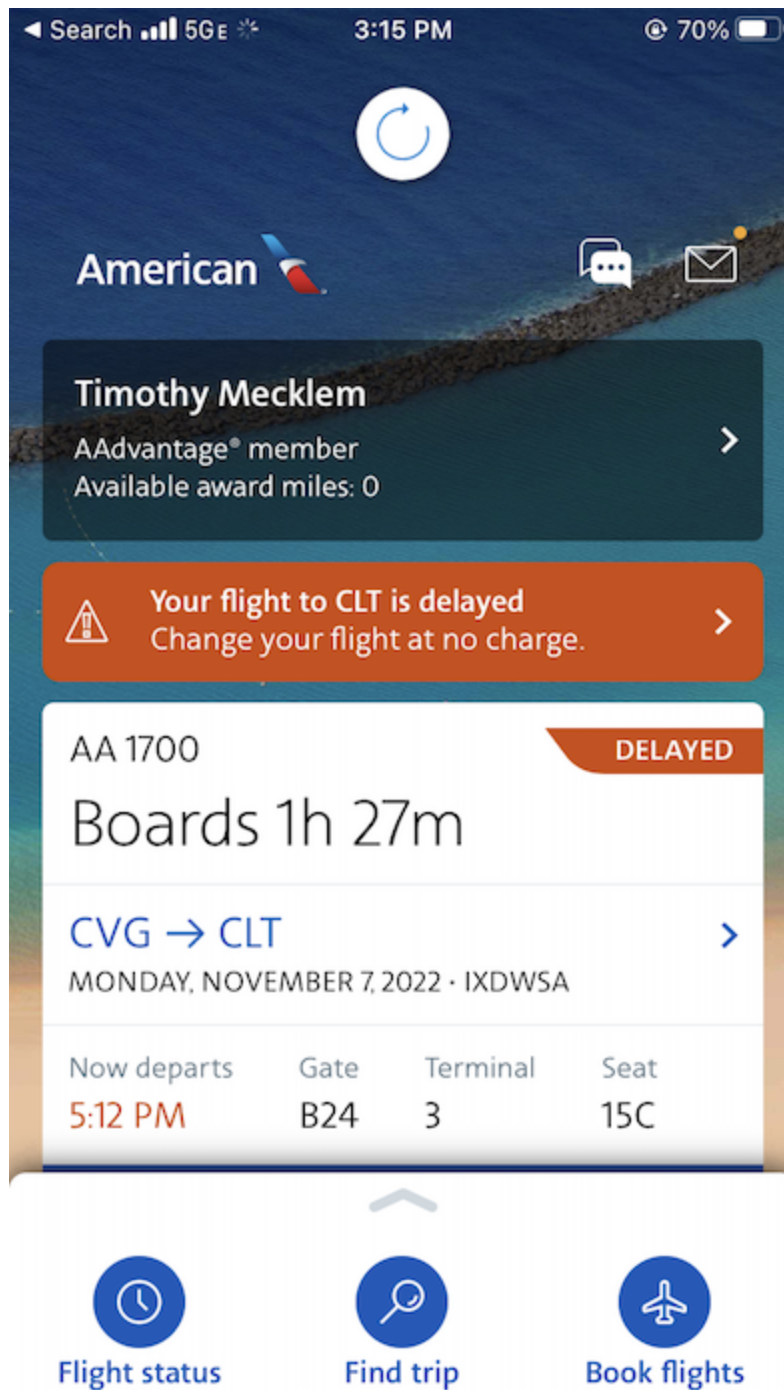
slido

Join at  
**slido.com**  
**#3097 839**



**Do today's web apps support real-time decisions better than the technologies they replaced?**

**Information isn't static**



**We're just catching up to the  
capability of our tools**

Tim Mecklem

VP of Engineering & Delivery

[launchscout.com](https://launchscout.com)

[@tim\\_mecklem](https://twitter.com/tim_mecklem) (Twitter)

[timothymecklem](https://www.linkedin.com/in/timothymecklem) (LinkedIn)

[@tim\\_mecklem@ruby.social](https://ruby.social/@tim_mecklem) (Mastodon)

## THE TAKEAWAY

**Real-time user experiences require  
*full-team, full-stack* solutions.**

Corollary:

This is not *just* a frontend dev or UX designer's problem.

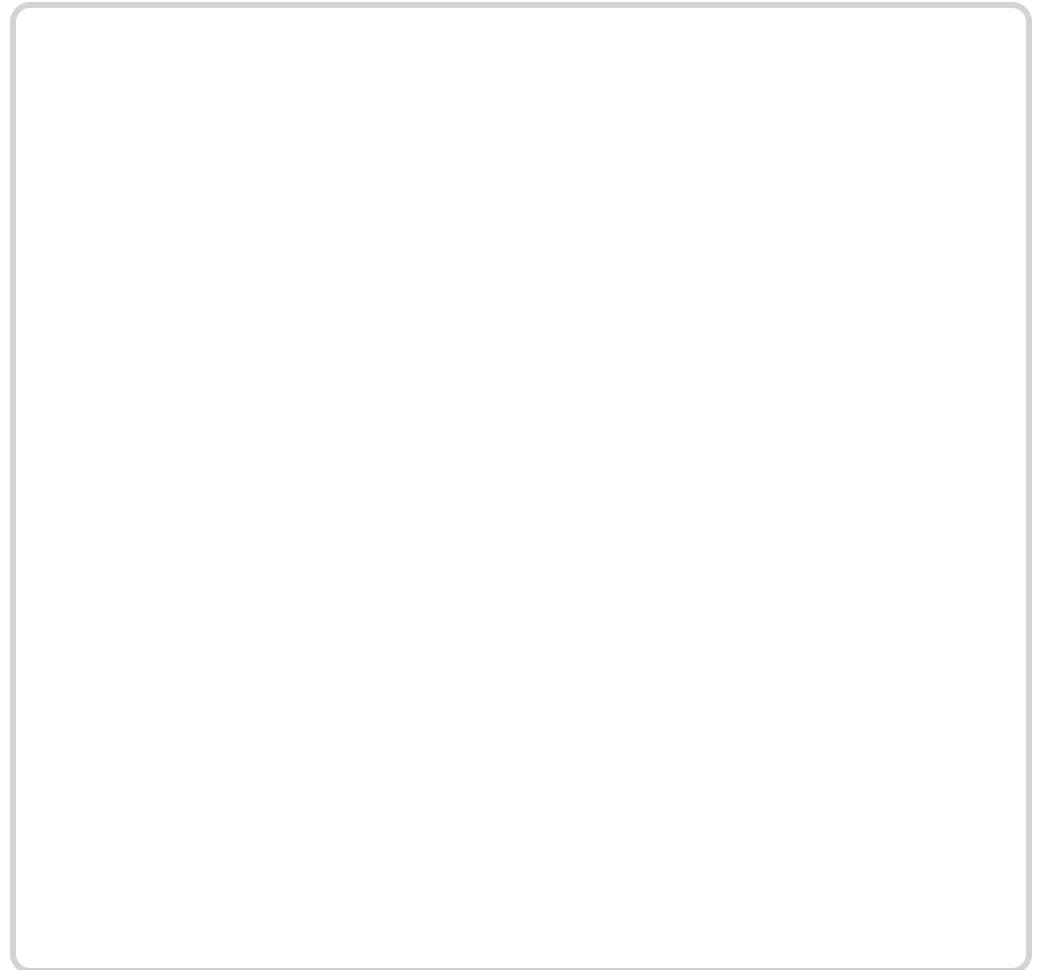
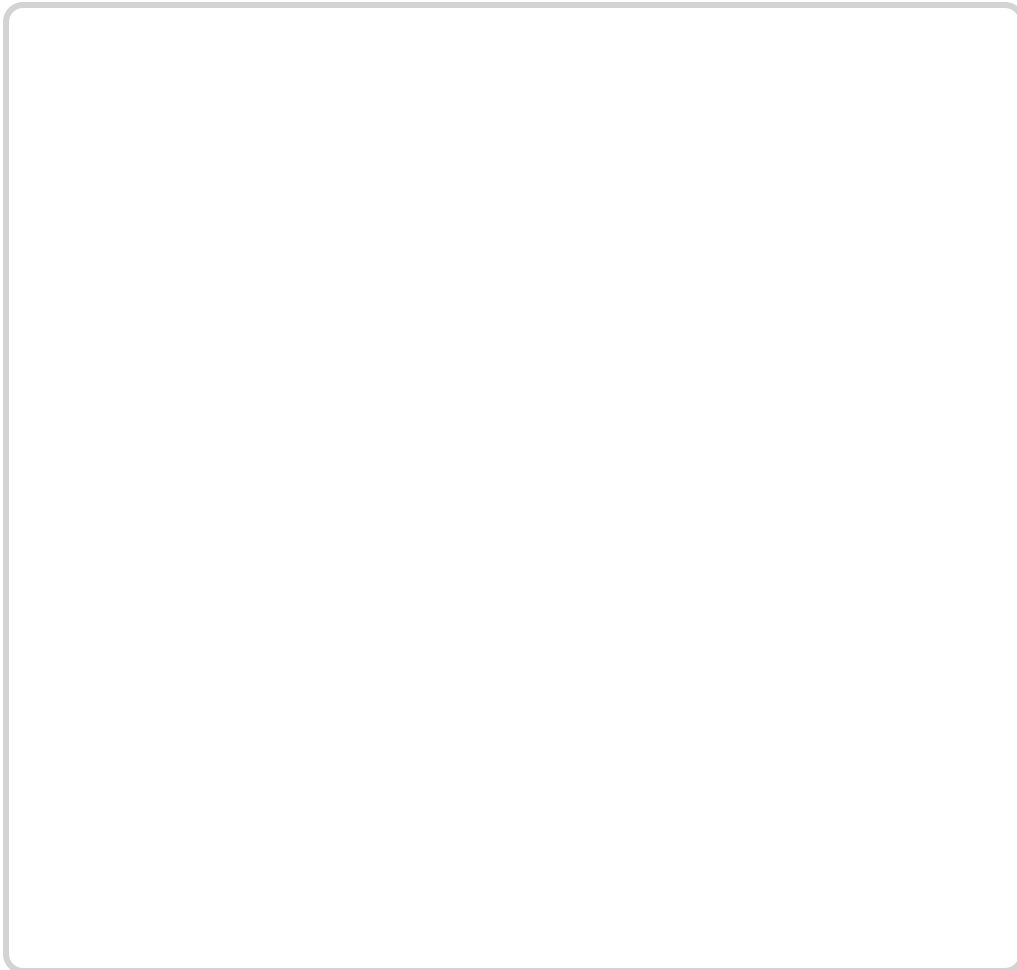


# The state of real-time web applications

**PROBLEM:**

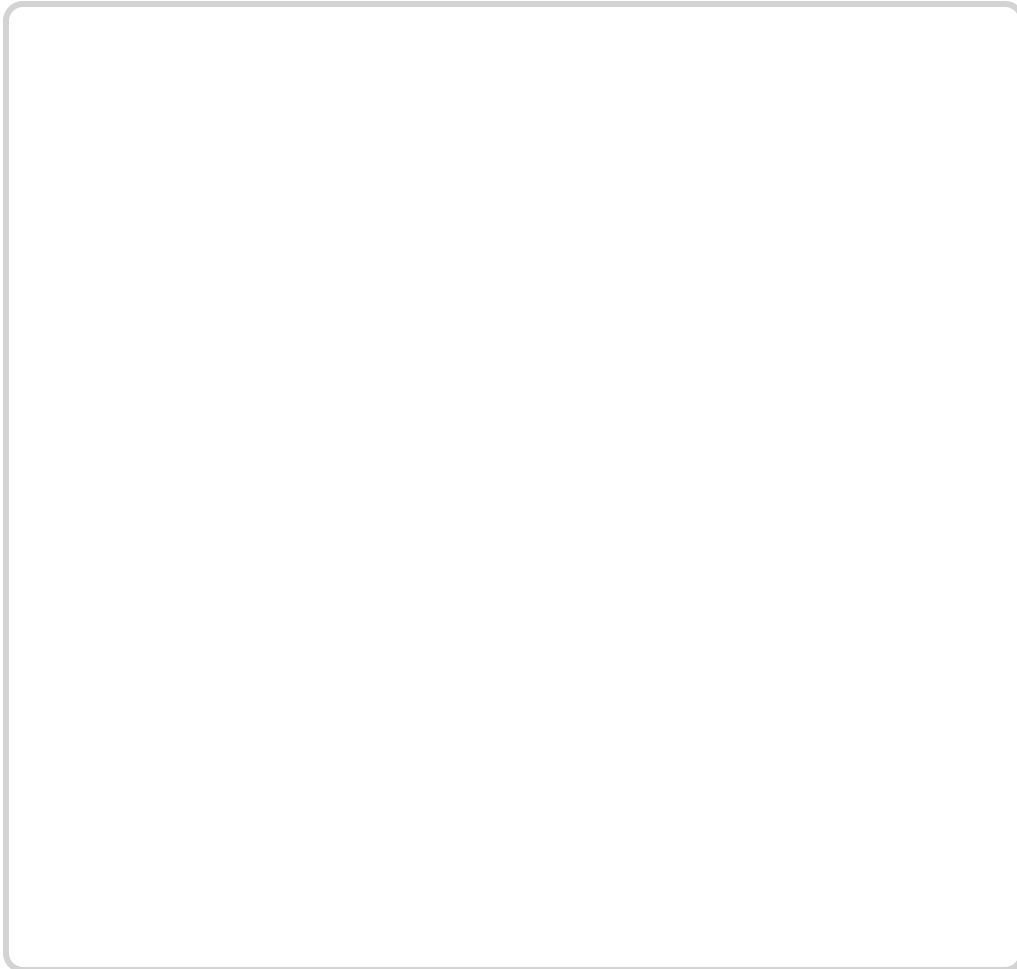
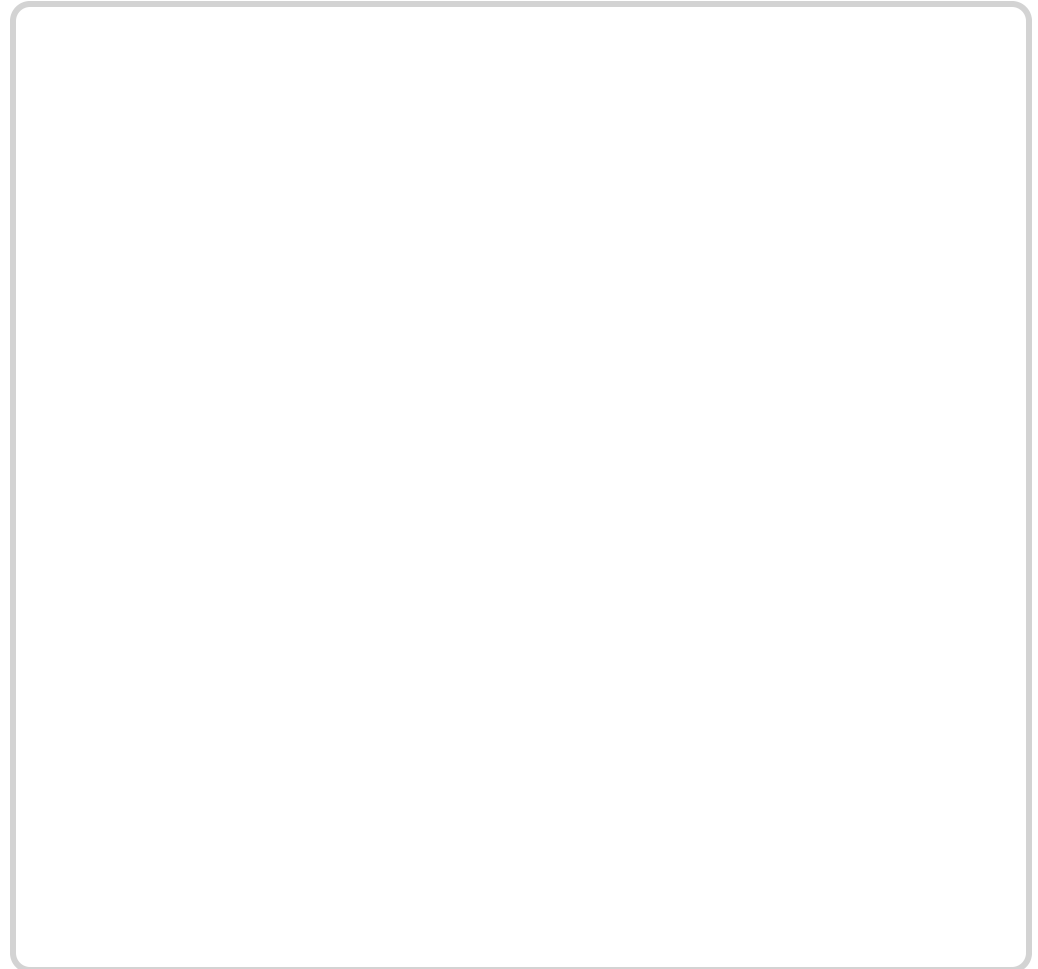
# Scarce Resources

## Example - low inventory on a commerce page



**What are the problems here?**

## One potential solution

A large, empty rectangular box with a thin gray border, intended for writing notes or details about the first potential solution.A large, empty rectangular box with a thin gray border, intended for writing notes or details about the second potential solution.

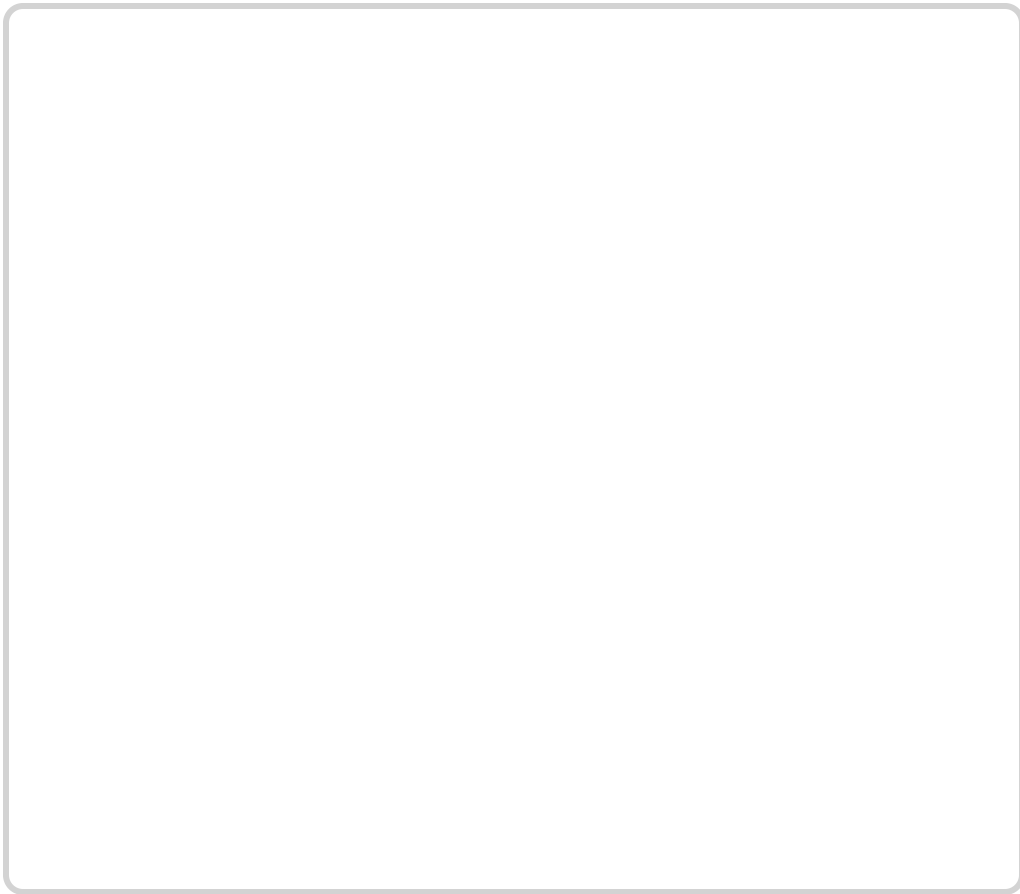
**PRINCIPLE**

**Value and respect the user's agency**

**PROBLEM:**

**Rapid influx of data**

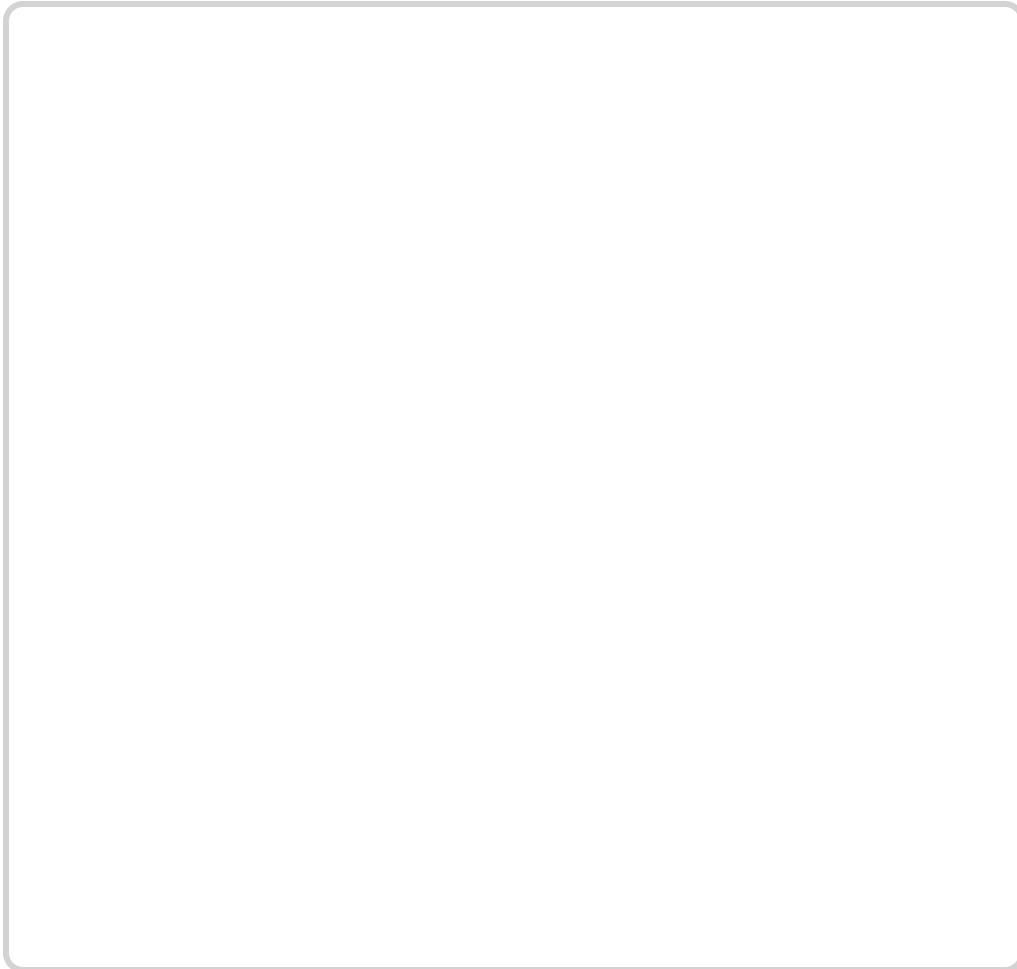
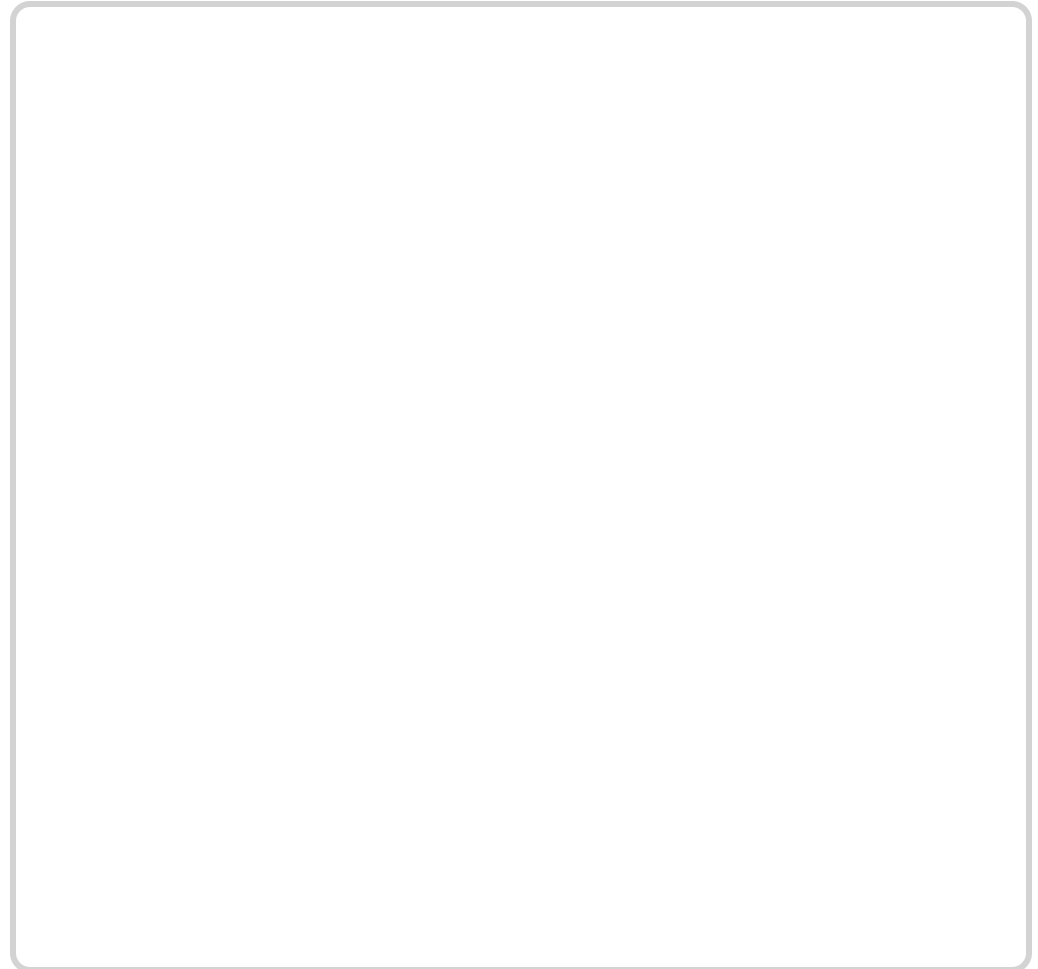
# Example





**What are the problems here?**

## One potential solution

A large, empty rectangular box with a thin gray border, intended for writing notes or details about the first potential solution.A large, empty rectangular box with a thin gray border, intended for writing notes or details about the second potential solution.

PRINCIPLE

**Give the user quality over quantity**

# But wait, there's more!

- Seamless Session Handoff Between Devices
- Collaborative Editing/Drawing
- Handling Disconnects Gracefully
- Managing Notifications Across Multiple Connected Devices
- Dynamic Maps and Location Based Services
- Paging Data
- Integrating IoT and Other Non-web Events

slido

Join at  
**slido.com**  
**#3097 839**



# Web technologies, revisited

# The case for something different

- Elixir, Phoenix, and LiveView
- Ruby, Rails, and Hotwire
- PHP, Laravel, and Livewire
- next.js and SSR React
- .NET and Blazor

# Keys to the technology decision



ONE FINAL CONSIDERATION

# The Real-time Uncanny Valley

## Quick setup — if you've done this kind of thing before



Set up in Desktop

or

HTTPS

SSH

git@github.com:tmecklem/my-game-changing-new-project.git



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

## ...or create a new repository on the command line

```
echo "# my-game-changing-new-project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:tmecklem/my-game-changing-new-project.git
git push -u origin main
```



## ...or push an existing repository from the command line

```
git remote add origin git@github.com:tmecklem/my-game-changing-new-project.git
git branch -M main
git push -u origin main
```



## ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

OUR ADVICE:

**Avoid the Uncanny Valley**

slido

Join at  
**slido.com**  
**#3097 839**



# Content for this talk

[github.com/tmecklem/realtime-web](https://github.com/tmecklem/realtime-web)

Slides: MARP ([marp.app](https://marp.app))

Examples: Phoenix and LiveView ([phoenixframework.org](https://phoenixframework.org))

Running app: <https://realtime-web.fly.dev>

# Questions?

LAUNCHSCOUT.COM



# Frontend Code!

Called on SSR and then as client connects via websocket

```
def handle_params(%{"sku" => sku}, socket) do
  # lookup information from database, etc
  user = fetch_user_from_database()
  product = get_product!(sku)

  if connected?(socket), do: InventoryEvents.listen_for_events(product)

  {:noreply,
   socket
   |> assign(:user, user)
   |> assign(:product, product)
   |> assign(:button_state, button_state(user, product))}
end
```

# Template

```
Welcome, <%= @user.name %>
<span style="float: right;">
  <%= link "View Cart", to: Routes.better_cart_show_path(...) %>
</span>

<h1>Products</h1>
<h2><%= @product.name %></h2>
<%= @product.stock_level %> in stock
<p>
  <button style="margin-top: 20px;" phx-click="add_to_cart"
    disabled={elem(@button_state, 0)}><%=elem(@button_state, 1)%> />
</p>
```



# Handling events

Events are handled similarly regardless of origin

```
# browser event
def handle_event("add_to_cart", _value, socket) do
  if BetterCommerce.exclusive_add_to_cart(...) do
    {:noreply,
     socket
     |> push_navigate(to: Routes.better_cart_show_path(...))
    }
  end
end

# backend event
def handle_info({:inventory_changed, %Product{sku: sku} = product}, socket) do
  {:noreply,
   socket
   |> assign(:button_state, button_state(socket.assigns.user, product))
   |> assign(product: product)
  }
end
```

# Backend Code!

```
updated_product = %{product | stock_level: stock_level - 1}

Repo.update!(updated_product)
InventoryEvents.notify(sku, :inventory_changed, updated_product)
```

# How do I get started with this?

Talk with me!

-- or --

*slightly dated*

Build a real-time Twitter clone in 15 minutes with LiveView and Phoenix 1.5

<https://www.youtube.com/watch?v=MZvmYaFkNJI>

# But will it scale?

Yes.

- <https://phoenixframework.org/blog/the-road-to-2-million-websocket-connections>
- <https://2022.elixirconf.com/speakers/zack-kayser>
- <https://davelucia.com/blog/observing-elixir-with-lightstep>

# Thank you!

Tim Mecklem

VP of Engineering & Delivery

[launchscout.com](https://launchscout.com)

[@tim\\_mecklem](https://twitter.com/tim_mecklem) (Twitter)

[timothymecklem](https://www.linkedin.com/in/timothymecklem) (LinkedIn)

[@tim\\_mecklem@ruby.social](https://ruby.social/@tim_mecklem) (Mastodon)

