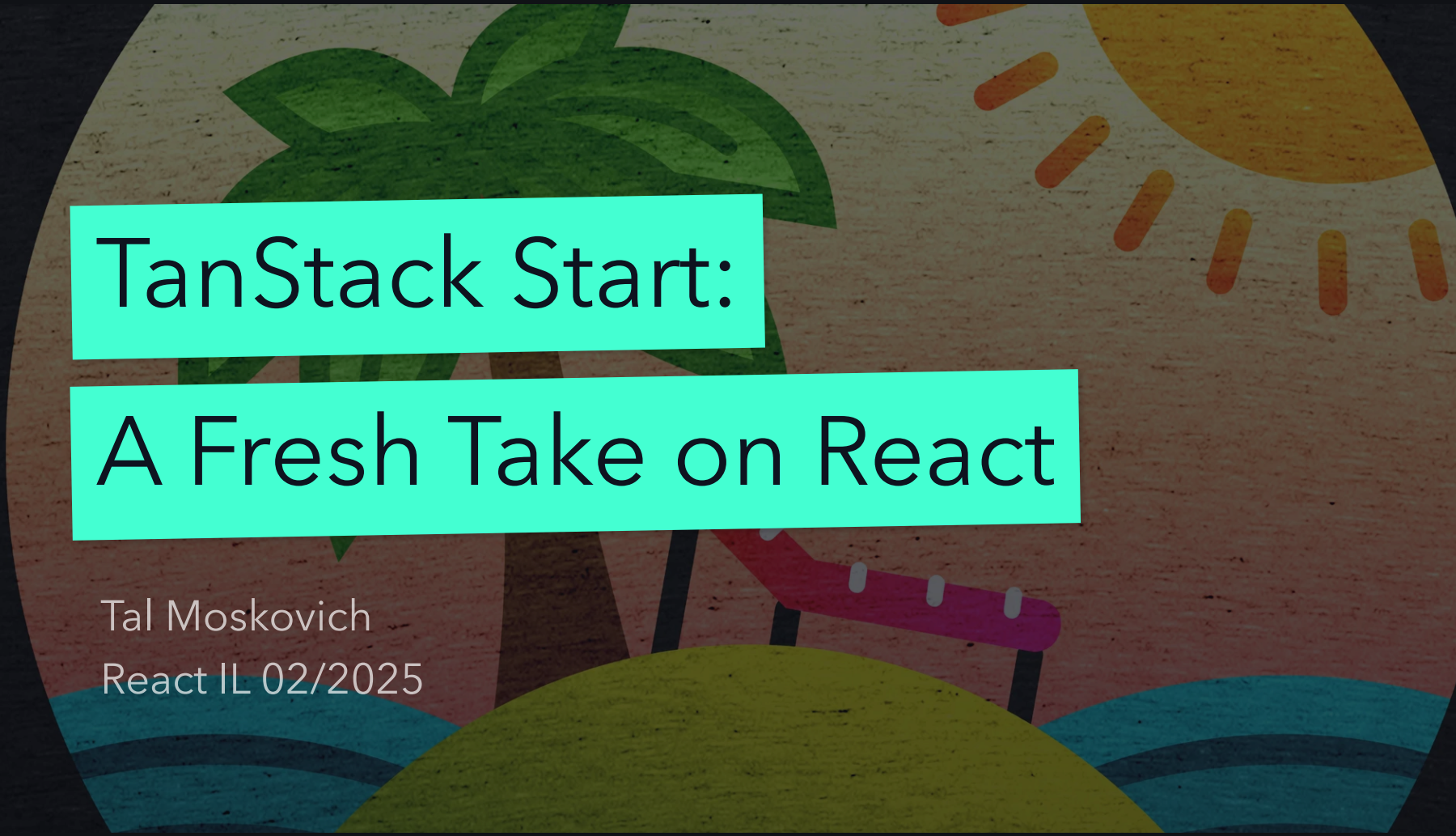

TanStack Start:
A Fresh Take on React
Tal Moskovich
React IL 02/2025

Hello!

Tal Moskovich

Committing Code & Pushing Personal Boundaries

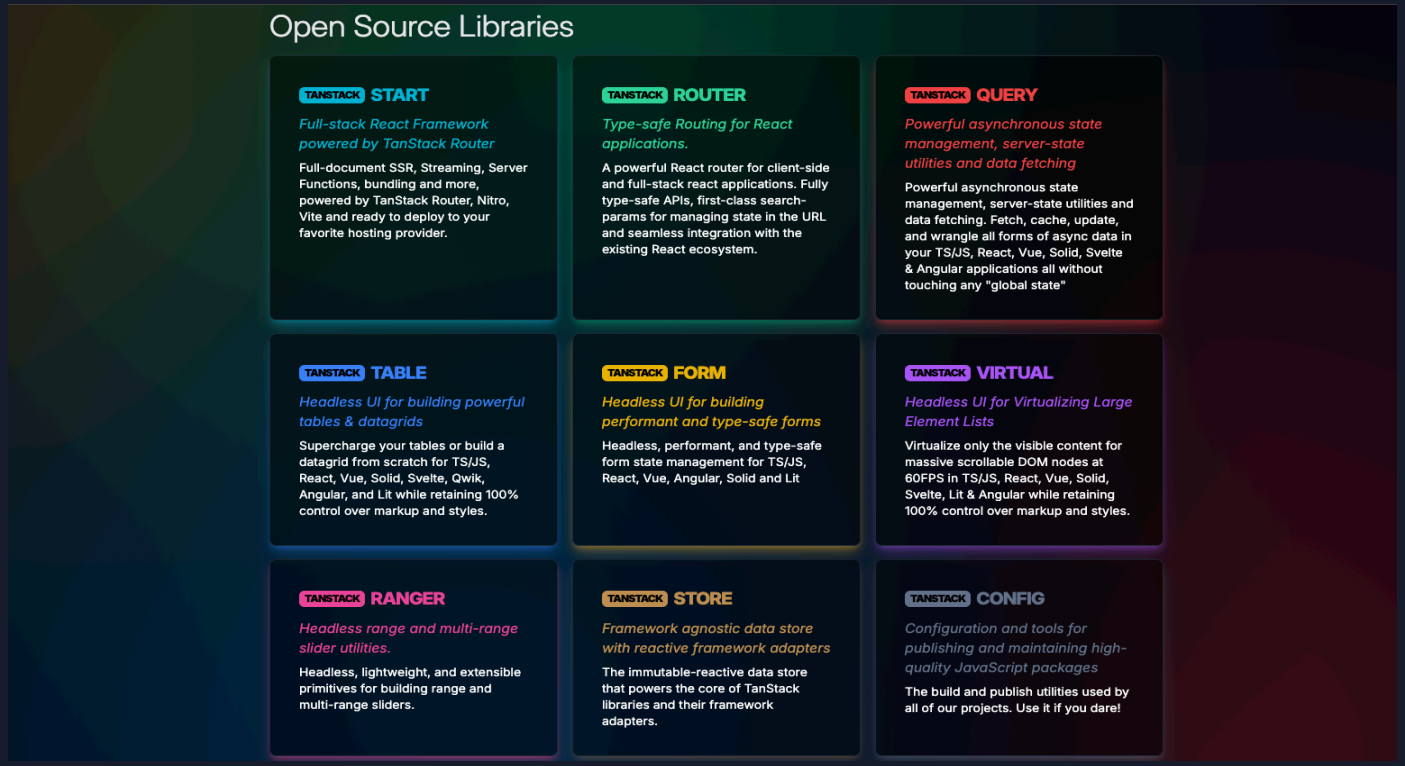
👨‍💻 Frontend Engineer @ ImagenAI

🎧 Podcaster @ Lo-Techi

https://lotechni.dev

https://www.linkedin.com/in/tmosko

Meet TanStack



The Magic Behind TanStack Start

The Magic Behind TanStack Start

- TanStack Router

The Magic Behind TanStack Start

- TanStack Router
- TanStack Query

The Magic Behind TanStack Start

- TanStack Router
- TanStack Query
- Vinxi -> Nitro -> Vite

The Magic Behind TanStack Start

- TanStack Router
- TanStack Query
- Vinxi -> Nitro -> Vite

Tanner Linsley

Why Do We Need Another Meta-Framework?

Why Do We Need Another Meta-Framework?

Client-First Philosophy 🎯

- Natural React development experience
- No "use client" directive chaos
- Opt-in to server features, not opt-out

Why Do We Need Another Meta-Framework?

Client-First Philosophy 🎯

- Natural React development experience
- No "use client" directive chaos
- Opt-in to server features, not opt-out

```
// Natural React code without directives
export function Component() {
  return <div>Client-side by default</div>
}
```

Why Do We Need Another Meta-Framework?

Cache = TanStack Query 🔄

- No new caching system to learn
- Same patterns you already use in production

Why Do We Need Another Meta-Framework?

Cache = TanStack Query 🔄

- No new caching system to learn
- Same patterns you already use in production

```
// Works seamlessly with loaders
export const route = createFileRoute('/users')({
  loader: async () => {
    // Same patterns, same caching, zero learning curve
    return queryClient.ensureQueryData({
      queryKey: ['users'],
      queryFn: getUsers
    });
  }
});
```

Why Do We Need Another Meta-Framework?

Smart Data Handling - No More Full Page Refreshes 🚀

- No more "all or nothing" data fetching approach
- No more waiting for slow data to show fast data
- Only refetch what actually changed

Why Do We Need Another Meta-Framework?

Smart Data Handling - No More Full Page Refreshes 🚀

- No more "all or nothing" data fetching approach
- No more waiting for slow data to show fast data
- Only refetch what actually changed

```
// Update user profile without refreshing entire page
export const updateProfile = createServerFunction('post')(
  async (data) => {
    await saveProfile(data);
    // Only invalidate profile data, everything else stays cached
    queryClient.invalidateQueries({
        queryKey: ['profile', userId]
      });  }
);
```

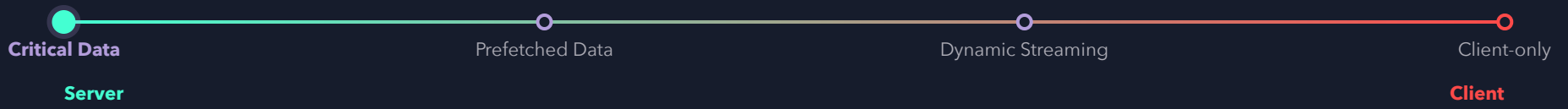
Why Do We Need Another Meta-Framework?

Incremental Adoption 📈

- Easy migration path from existing apps
- Start with client-side features
- Add server capabilities gradually
- No need for full rewrites

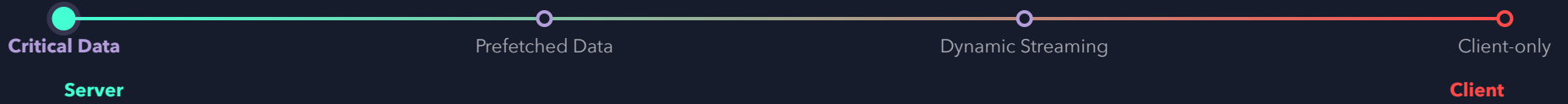
Demo

Data Fetching



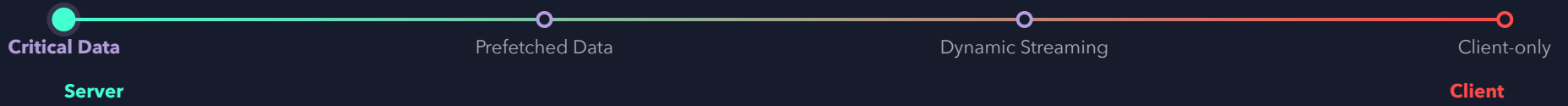
Data Fetching

```tsx
// Fetching critical data on server-side
export const route = createRoute({
  loader: async () => {
    await queryClient.ensureQueryData(postsQuery)
  },
  component: () => {
    const query = useQuery(postsQuery)
    // Data is already available from loader
    return <PostsList posts={query.data} />
  }
});
```



Data Fetching

```tsx
// We can use `fetch` directly in the loader instead of `TanStack Query`
export const route = createRoute({
  loader: async () => ({
    posts: await fetchPosts()
  }),
  component: () => {
    const { posts } = useLoaderData()
    return <PostsList posts={posts} />
  }
});
```



Data Fetching

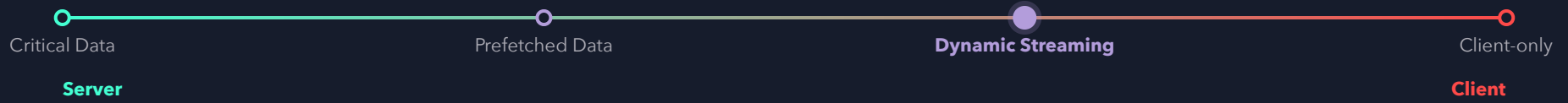
```
// Prefetch data with deferred loading using `SuspenseQuery`
export const route = createRoute({
  loader: async () => {
    queryClient.prefetchQuery(postsQuery)
  },
  component: () => {
    const { data } = useSuspenseQuery(postsQuery)
    return <PostsList posts={data} />
  }
});
```

Data Fetching

```tsx
// Deferred loading with `Await` component
export const route = createRoute({
  loader: async () => ({
    postsPromise: fetchPosts()
  }),
  component: () => {
    const { postsPromise } = useLoaderData()
    return (
      <Await
        promise={postsPromise}
        children={posts => {
          return <PostsList posts={posts} />
        }}
      />
    )
  }
});
```

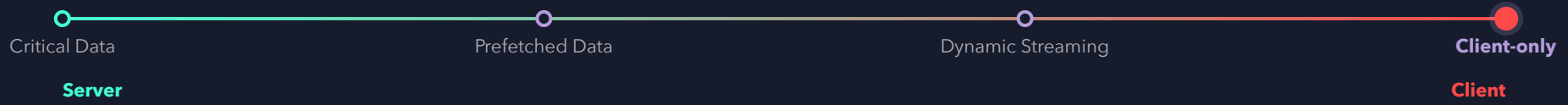
Data Fetching

```
// Dynamic streaming

const { data } = useSuspenseQuery(postsQuery)
```



Data Fetching

```
// Client-only data fetching

const { data } = useQuery(postsQuery)
```



Demo + Slides

Thank You!

Tal Moskovich

lotechni.dev