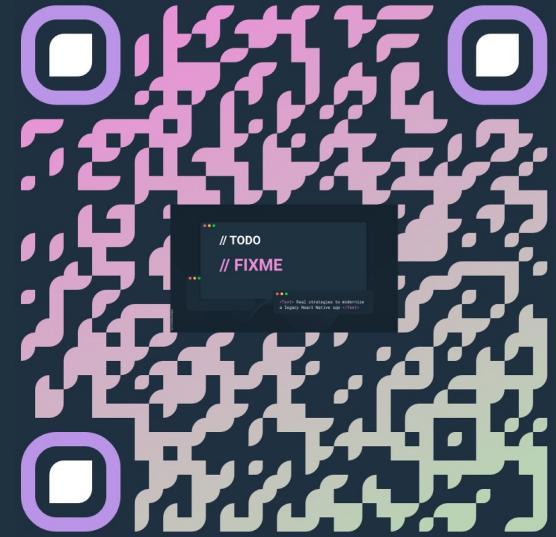


// TODO
// FIXME



<Text> Real strategies to modernize
a legacy React Native app </Text>

HELLO! I'm Erik.

I'm an engineering leader at Cisco
and that's my mom on the right.







01 LET'S SET A GOAL

<> Help at least 1 person launch a refactor project somewhere to modernize some piece of painful legacy code. </>

TABLE OF CONTENTS.



01

Let's set a goal.



02

Then set an outline.



03

Level-set with a reality check.



04

Get serious about legacy systems.



05

Winning management strategies for modernization.



06

Finish with hope & positivity.



This is EASY.

1. rm rf LegacyProject
2. react-native init GreenfieldProject
3. Profit.



End.



Questions?



This is EASY. 😅

1. `rm rf LegacyProject`
2. `npx create-react-native-app GreenfieldProject`
3. Profit.



This is EASY. 😅😅😅

1. `rm rf LegacyProject`
2. `npm uninstall -g react-native-cli @react-native-community/cli && npx @react-native-community/cli@latest init GreenfieldProject`
3. Profit.



This is EASY.

1. `rm rf LegacyProject`
2. `npx create-expo-app@latest GreenfieldProject`
3. Profit.



03 REALITY CHECK

<> What is software engineering actually like? </>

Let's get a **REALITY CHECK**



Ideal Software Engineering:

1. Greenfield
2. Latest Tech Stack
3. Amazing Tooling
4. Simple Data Models
5. Everything is Sensible
6. Like Your Side Projects...



Let's get a **REALITY CHECK**



Ideal Software Engineering:

1. Greenfield
2. Latest Tech Stack
3. Amazing Tooling
4. Simple Data Models
5. Everything is Sensible
6. Like Your Side Projects...



Reality:

1. OMG Legacy PAINZZZ ugghh
2. Multiple overlapping systems that refuse to go to couples therapy
3. Normalized broken windows
4. Other Priorities!!!!
5. Other People! Ah!!
6. OPINIONS! 🙌 OPINIONS! 🙌 OPINIONS! 🙌
7. Natural bias toward inaction (status quo)
8. TECH DEBT
9. TECH DEBT
- 10.



“

Hi, YOU ARE GOING TO BE
DEALING WITH TECH DEBT
AND IT'S GOING TO SUCK. YOU
MIGHT START QUESTIONING
REALITY.



– Me (Now)



“

THERE IS HOPE.

THERE ARE WAYS TO MAKE
REALITY BETTER.



– Me (Now)



04 GETTING SERIOUS ABOUT LEGACY

<> What are we likely to encounter? How can we make things better for ourselves? </>

Best Of...React Native Project Modernizations



Old React Native

The fundamentals
change

Repo Architecture

Mono. Micro. Mono.
Wait, did I already
say “Mono”?

State Management

Which library is the
best? Which are the
not best?

Design System

Watch Hinge Health’s
talk from yesterday.

Typed JavaScript

JS => TS

Testing

Simple principles will
take you far.



React Native turns 10 next year.





“

REACT NATIVE CAN
SIMULTANEOUSLY BE HYPER-
MODERN & TEDIOUSLY LEGACY.



– Me (Now)



Some *REACT NATIVE* Specifics...



When a React Web Dev complains about Navigation...





YOU ADAPTED TO THE DARK

I WAS BORN IN IT



Some *REACT NATIVE* Specifics...

Navigation



Some *REACT NATIVE* Specifics...

Navigation

Animations



Some *REACT NATIVE* Specifics...

Navigation

Animations

Gestures



Some *REACT NATIVE* Specifics...

Navigation

Animations

Gestures

Using native dependencies



Some *REACT NATIVE* Specifics...

Navigation

Animations

Gestures

Using native dependencies

ANDROID IS ALSO A MOBILE PLATFORM?





Some *REACT* Specifics...

Lifecycle methods vs. Hooks (a.k.a. Class vs. Function Components)

HoCs

Render props

Mixins

Class to Class Components



If you don't use ES6 yet, you may use the `create-react-class` module instead:

```
var createReactClass = require('create-react-class');
var Greeting = createReactClass({
  render: function() {
    return <h1>Hello, {this.props.name}</h1>;
  }
});
```





WE NEED TO GO

DEEPER



Some **JAVASCRIPT** Specifics...

ES6. ES6. ES6.

Classes

Imports

Async (Promises, Generators, Async/Await)

Array Methods (.map, .reduce, .filter)

Arrow Functions, Destructuring, ...Spread & Rest...



66

CHANGE IS GOOD.



– Me (Now)



“

CHANGE IS GOOD.

CHANGE IS INEVITABLE.



– Me (Now)



“

CHANGE IS GOOD.

CHANGE IS INEVITABLE.

BET ON CHANGE.



– Me (Now)

PITCH IT.



DEFINE



START



PROCESS

PITCH IT.



DEFINE

Best practices

Easy & Brief



START



PROCESS

PITCH IT.



DEFINE

Best practices

Easy & Brief



START

"As you're in
there."

Small wins



PROCESS

PITCH IT.



DEFINE

Best practices

Easy & Brief



START

"As you're in
there."

Small wins



PROCESS

Tooling - Friendly
& Helpful (e.g.
Linters)

"Jank Days"

Measurement



STATE MANAGEMENT



NOT ENOUGH TIME FOR
STATE
MANAGEMENT
IN THIS TALK.

PITCH IT.



DEFINE

Where, When, What,
How, and Why?

Pick one library
for each concern

Analyze the state
of your state
management



START

Consolidate in
waves and
iterations (e.g.,
by feature)

...OR all at once in
some cases



PROCESS

Focus on the Dev
Experience

Linters

Minimize
boilerplate
surface



TYPED JavaScript



66

tsconfig.json IS YOUR FRIEND.



– Me (Now)



“

tsconfig.json IS YOUR FRIEND.

TYPES FOR DEV EXPERIENCE.



– Me (Now)



“

tsconfig.json IS YOUR FRIEND.

TYPES FOR DEV EXPERIENCE.

TYPES FOR BOUNDARIES.



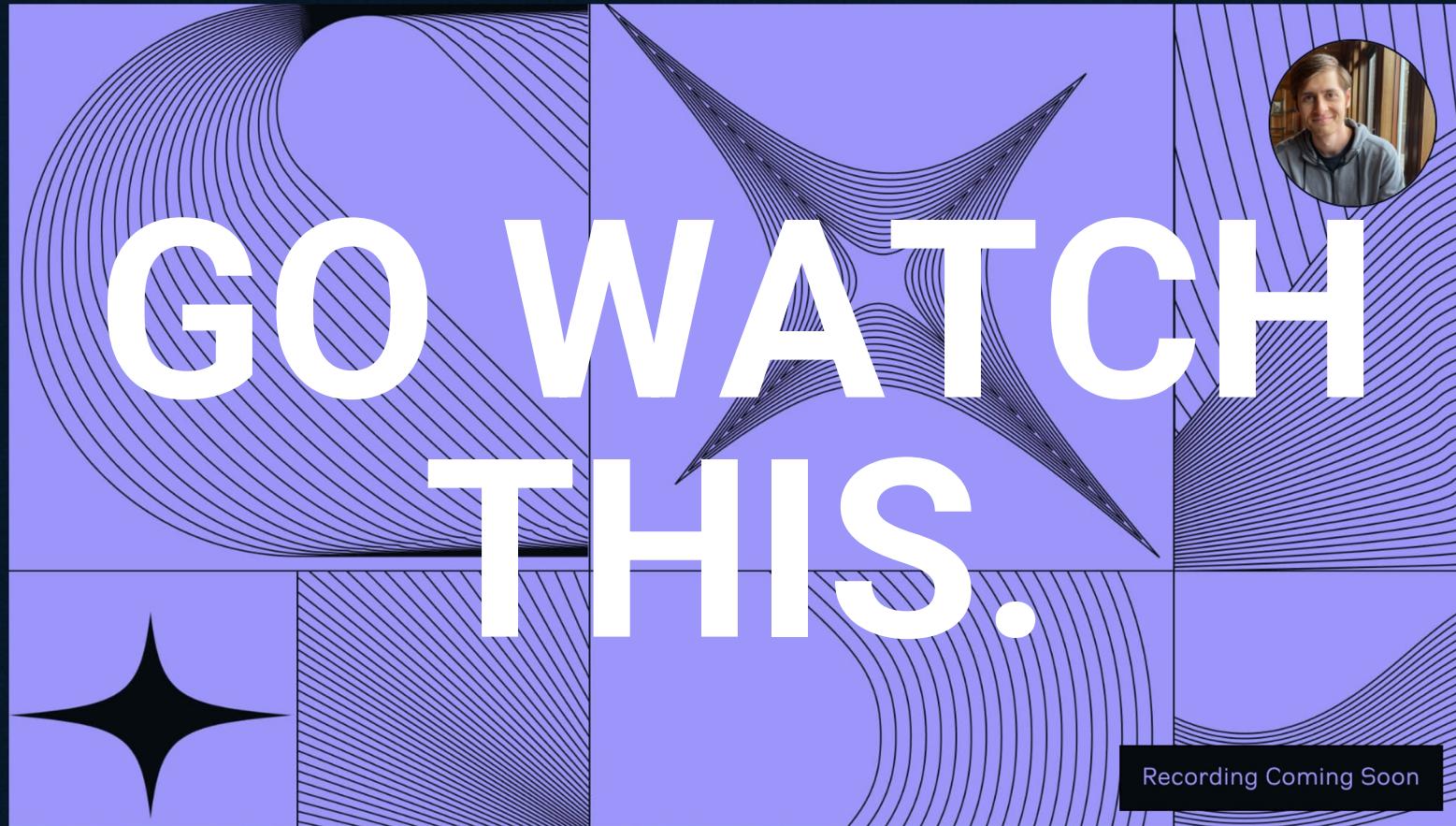
– Me (Now)

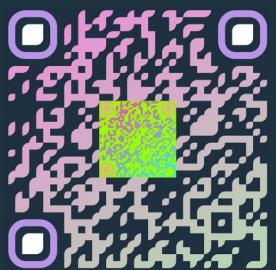
Optimizing With Static Hermes

30 min talk

Talk by: [Neil Dhar](#)

A walk through the Static Hermes optimizer, describing how it takes advantage of static type information to improve performance.





Game Plan: TS-ify your JS



Business Types

Start with your most important data types

Measure

Measure all `any` types and conversions.

"noImplicitAny": true

Explicitly type everything, even your any types

Improve

Use tooling to prevent falling back

tsconfig.json

Your best friend.

propTypes ..byeeeeee

These are usually better to just delete instead of using primitives at the component interface level.

PITCH IT.



DEFINE VALUE

Catch bugs at dev time.

Documentation at fingertips.

Performance



START

Gameplan.

Small wins.

Get to “last mile” with momentum.



PROCESS

Tooling enforcement

“Type Jank Days”

Measurement

Celebrate wins!



Repo ARCHITECTURE

Repo ARCHITECTURE ?



<> The monorepo movement is one of the biggest changes right now </>

Repo ARCHITECTURE ?



- <> The monorepo movement is one of the biggest changes right now </>
- <> We left monoliths alone for so long... </>

Repo ARCHITECTURE ?



- <> The monorepo movement is one of the biggest changes right now </>
- <> We left monoliths alone for so long... </>
- <> We thought splitting microservices/apps was better </>

Repo ARCHITECTURE ?



- <> The monorepo movement is one of the biggest changes right now </>
- <> We left monoliths alone for so long... </>
- <> We thought splitting microservices/apps was better </>
- <> Turns out that can be chaotic and inefficient </>

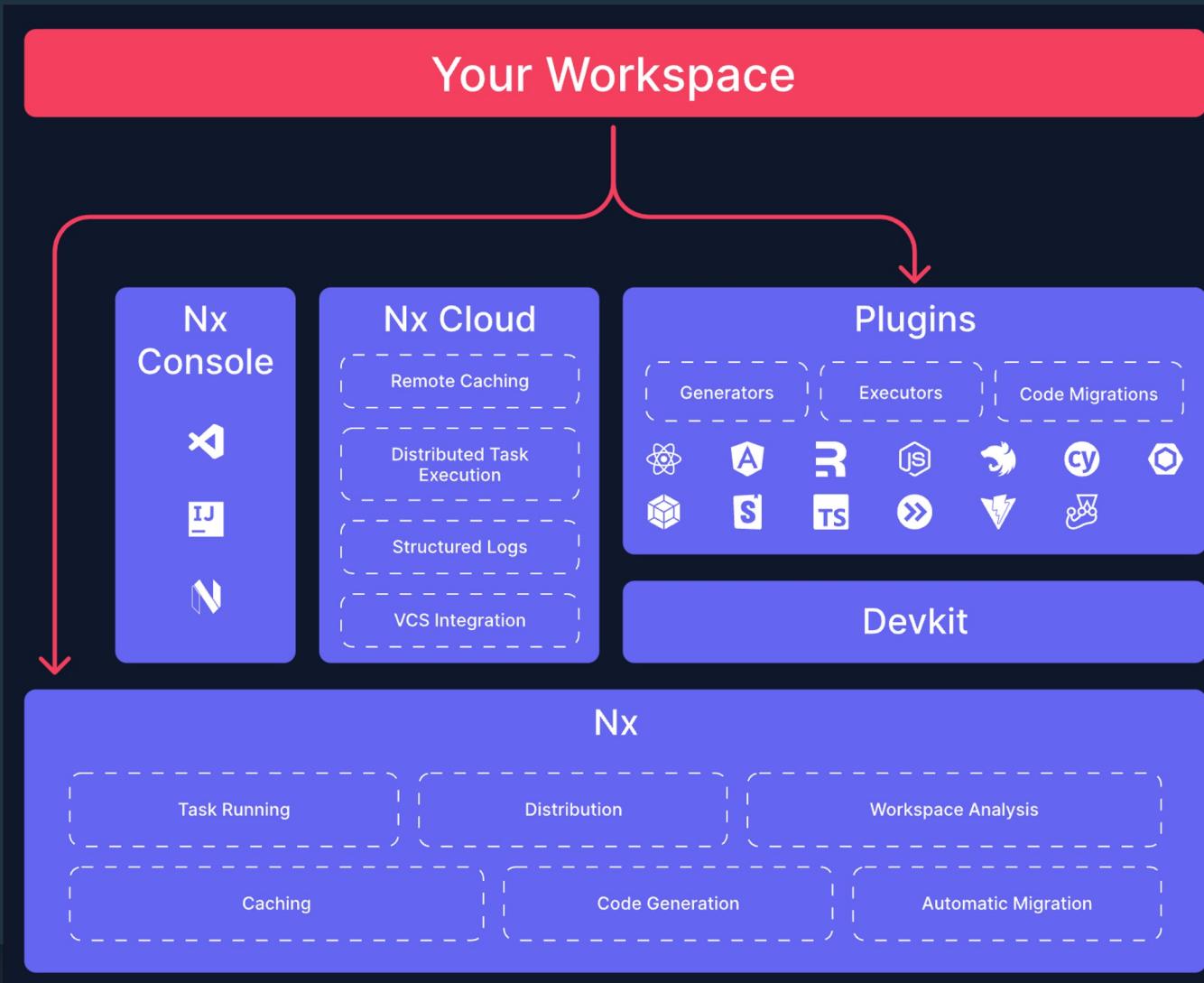


Nx.dev Monorepo Tooling FTW

<> “Monorepo” not “monolith” </>



How Does Nx Work?



Managed Repo

Nx knows the relationships

Magic

- `:affected` flag
- Workspace Analysis
- Generators
- Caching

PITCH IT.



VALUE

Godly Visibility

“Affected” code &
CI jobs

Distributed
caching



START



PROCESS

PITCH IT.



VALUE

Godly Visibility

“Affected” code &
CI jobs

Distributed
caching



START

Read the docs

Move everything
into a
‘deprecated’
folder

Split things out
into libs and apps



PROCESS

PITCH IT.



VALUE

Godly Visibility

“Affected” code &
CI jobs

Distributed
caching



START

Read the docs

Move everything
into a
‘deprecated’
folder

Split things out
into libs and apps



PROCESS

Nx best practices

Measure
improvements

Use CODEOWNERS
tools

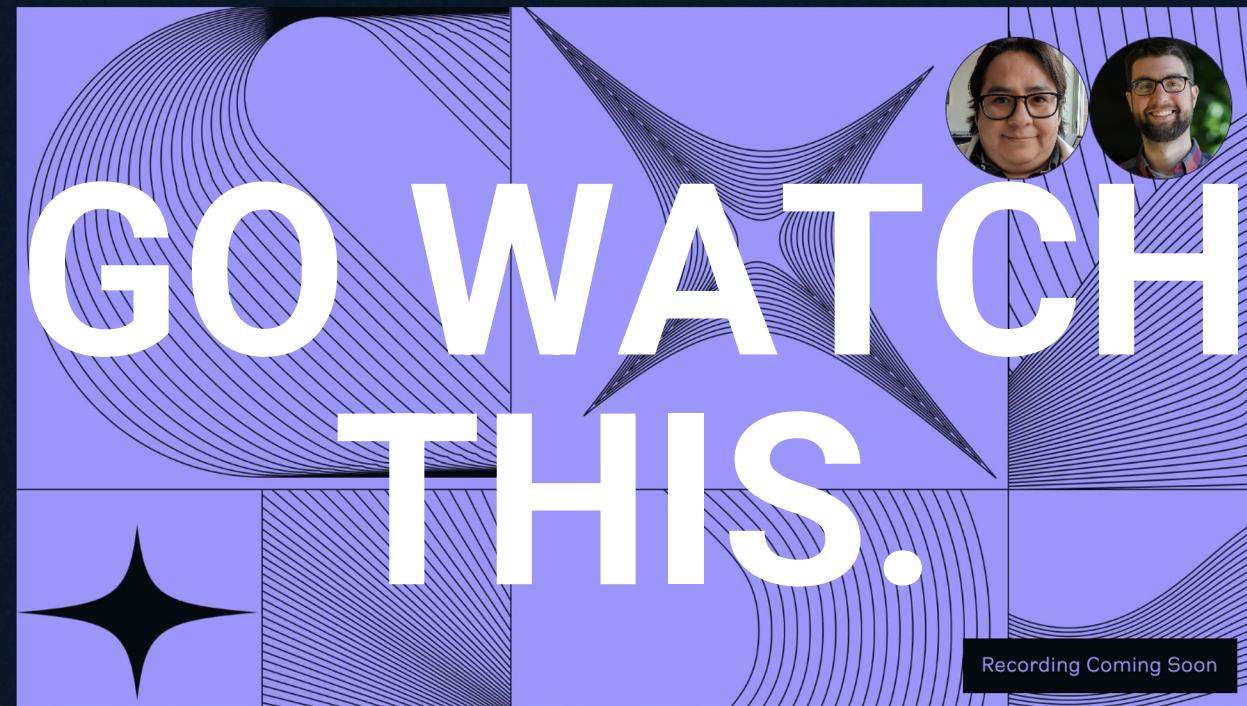


DESIGN SYSTEMS

The Efficient Path To Design System Implementation in Large React Native Applications

Talk by: [Rafael Mendiola, Kyle Rosenberg](#)

Hinge Health developed a design system, Heal, and transformed its very large React Native codebase that is developed by over 90 individual contributors. The implementation involved creating a separate component library using react-native-web and storybook while ensuring compatibility with our mobile binaries, assessing component usage with a custom analyzer tool, and extensive code refactoring to replace legacy components. This advanced-level presentation will outline our efficient strategy for managing large-scale design system integration.





Useful TESTING

PITCH IT.



VALUE

Empower Change

Empower *Radical*
Change

Embedded Expertise



START

Audit your *real*
coverage

Audit your tools
and libraries

Identify the
critical paths



PROCESS

Make it easy.

Make it clear.

Make it useful.

Make it
trustworthy.



05 WINNING MANAGEMENT STRATEGIES

<> Managing people, expectations, and your ambitious roadmap to modernity </>



“

WHEN EXECUTED,
PROFESSIONALLY WRITTEN
SOFTWARE IS MEANT TO DO
SOMETHING VALUABLE FOR
SOMEONE, SOMEWHERE.



– Me (Now)

(Some) Business Metrics



VELOCITY

How quickly can devs get things done?

PERFORMANCE

Is the application sluggish?

COST OF INACTION

What happens if we just don't though?

SURVEYS

Dev Experience,
Anecdotes, Wishbox

CODE CHURN

How much code has to be rewritten to make changes?

RECRUITING

Will candidates be excited to work in your stack?



Explora the D.O.R.A.



Deployment Frequency

How often an organization successfully releases to production

Change Failure Rate

The percentage of deployments causing a failure in production

Lead Time for Changes

The amount of time it takes a commit to get into production

Time to Restore

How long it takes an organization to recover from a failure in production

Industry standard

"4 Key Metrics to assess the performance of a software development team."

Originally from Google Cloud DevOps group

Measure What Matters

Drill down.

Customize for your use cases and system.

Bonus: Code Review & Code Complexity metrics are my personal favorites

Reasonable Expectations



NEVER OVERSELL

Keep it matter-of-fact

NO MIRACLES

And no heroes either!

MEASURE SUCCESS

Metrics & Wins!

START SMALL

With low-hanging fruit
and high ROI!

MARATHON

Keep it up as a
regular habit.

FIND ALLIES

Who else will help
you?



“

CHANGE IS GOOD.

CHANGE IS INEVITABLE.

BET ON CHANGE.



– Me (Now)



“

OUR INDUSTRY EVOLVES AT A
RAPID PACE.

AMAZING PEOPLE RELENTLESSLY
MAKE BUILDING SOFTWARE
BETTER.



– Me (Now)



“

THAT'S A GREAT THING.



– Me (Now)



“

TO ALL YOU BRAVE ADVENTURERS
PUSHING THE FRONTIERS OF
BETTER SOFTWARE ENGINEERING,
YOU LIFT US ALL UP.



– Me (Now)



66

BUILD *FOR* CHANGE.



– Me (Now)



66

BUILD **TECH WEALTH.**



– Me (Now)

THANK
YOU!



Erik Thorelli
@esthor