

# **Making life better with custom DevTools**

**Stories from Facebook and Flipper**

**Timur Valiev & Michel Weststrate**

# Next 40-ish minutes of your life

Should I consider building custom dev tools and how to approach it?

- I'll start with my story
- What are dev tools & What do I mean by \*custom\*
- Flipper - what's included and how to extend
  - “Starter” ideas and more advanced stuff - threading & Server Driven UI
- How to measure success and identify opportunities
- Do's and Dont's
- I hope we'll have fun!

# How I ended up with building DevTools

## Story of long onboarding and steep ramping up curve

- I took me a few months to start being productive
- A lot of in-house built custom tools:
  - ~~Git~~ Mercurial
  - ~~Maven/Gradle~~ Buck
  - Monorepos, custom protocols, own programming languages
- Huge, complex projects with 100s and 1000s of people working on them
- I felt that our tools should be better to help us

# Why **\*custom\*** Dev Tools?

## Symptoms and patterns

- In house built frameworks/libs/communication protocols etc.
- [Very] complex architecture and business logic
- Tough and long onboarding/ramp up
- Feeling of 'slow' development
- Blocked people - waiting for expert's help

# What is Dev Tools?

**Virtually any tool that help you to develop your stuff**

- Chrome Dev Tools
- Your IDE's debugger
- That script your wrote 3 years ago to prettify logs
  - it has become an essential part of your day-to-day work since then!
- You name it

# What are *\*custom\** Dev Tools?

## What are we talking about?

- Aren't they all custom to some extent?
- By *\*custom\** we mean that you extend existing/create new tools specifically for your needs


# Why *\*custom\** Dev Tools?

How can they help? And how to measure success?

- Dev Velocity (aka how to make devs faster)
- Self Sufficiency (aka how to unblock devs)
  
- So, should we invest in it?
- It depends. Sorry, but it's the best answer I can come up with

# What are \*custom\* Dev Tools?

## Examples

- CLI tools
- IDE extensions
- Linters/Code analysis
- Browser's Dev Tools extensions (think of React Dev Tools)
- Flipper - extensible mobile apps debugger
  - Docs: <https://fbflipper.com>
  -  [@flipper\\_fb](https://twitter.com/flipper_fb)



# Flipper

## Custom Plugins - Simple ideas

- [Custom] communication protocols
- Logging/telemetry validation
- Configuration helpers & other debug utils

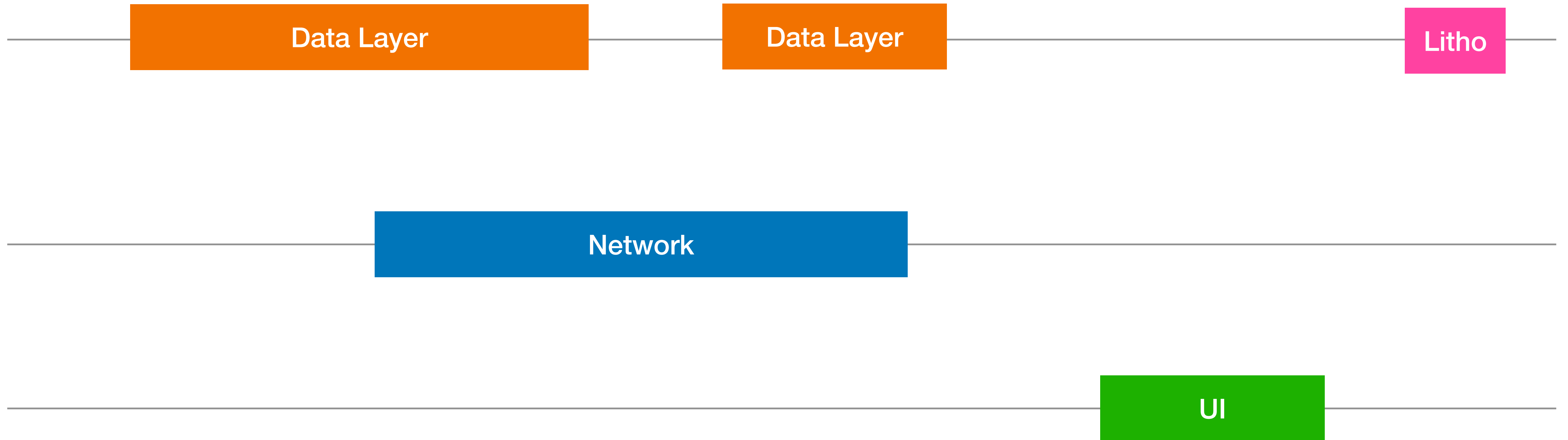
# Flipper

## Custom Plugins - Advanced stuff

- Causality and execution flows in complex scheduled environments
- Inspecting and Debugging Server driven UI
- State monitoring

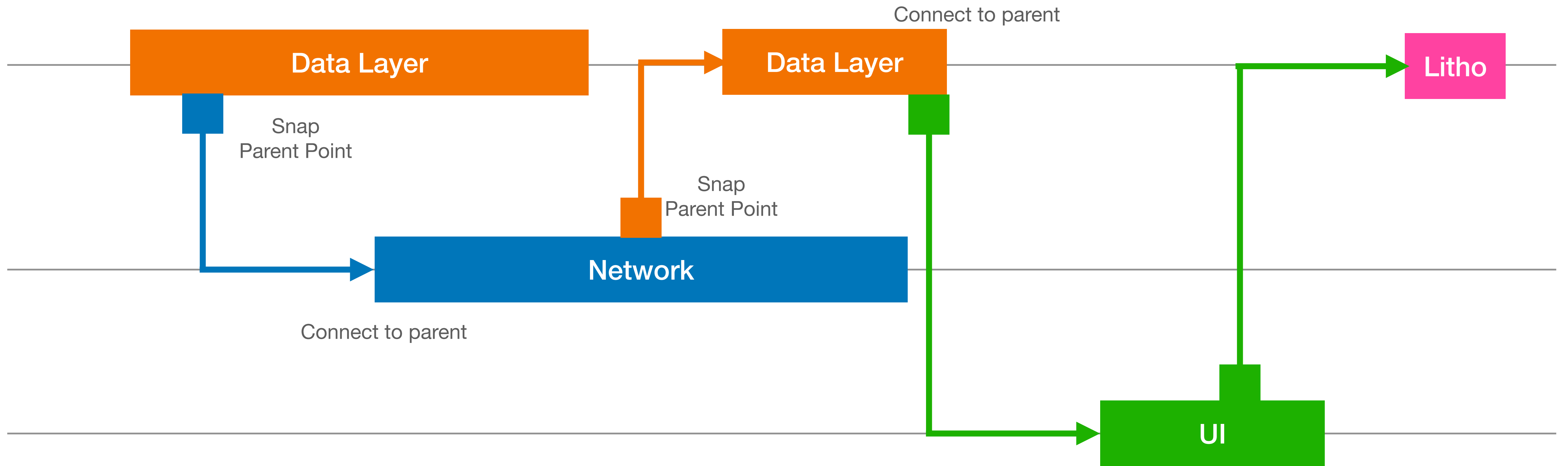
# Causality Flows

Connect frameworks across threads and units of work so **devs can spot overall flow inefficiencies**



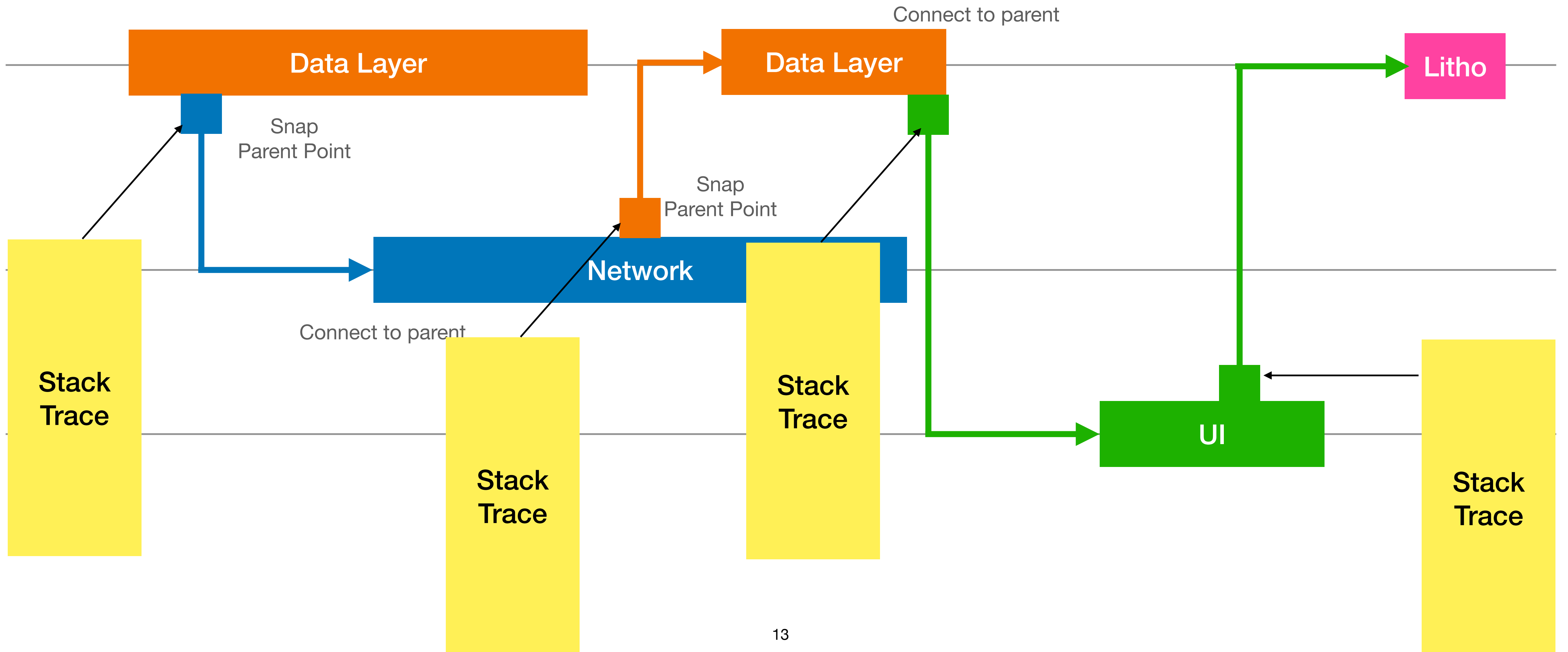
# Causality Flows

Connect frameworks across threads and units of work so devs can spot overall flow inefficiencies



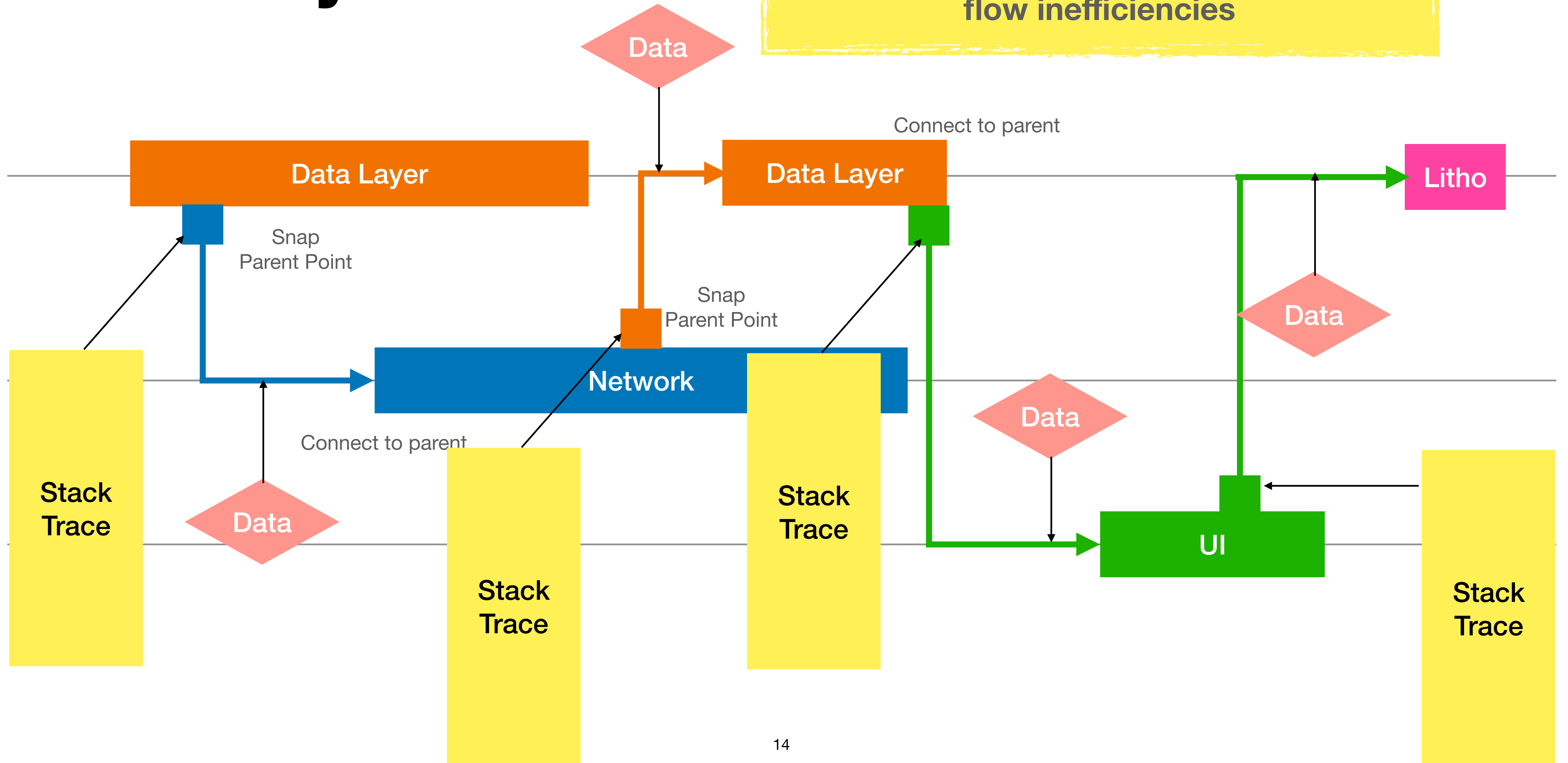
# Causality Flows

Connect frameworks across threads and units of work so devs can spot overall flow inefficiencies



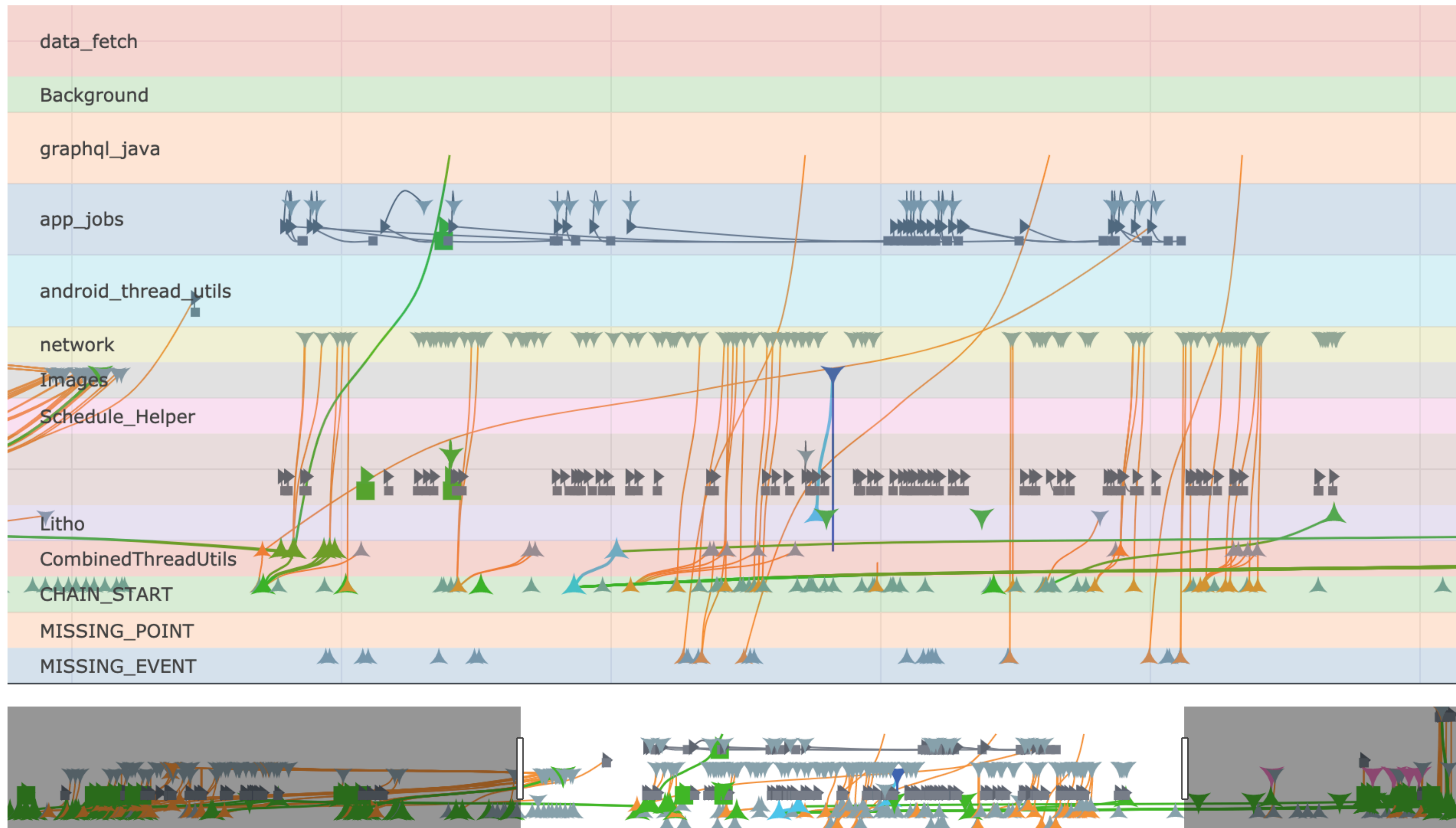
# Causality Flows

Connect frameworks across threads and units of work so devs can spot overall flow inefficiencies





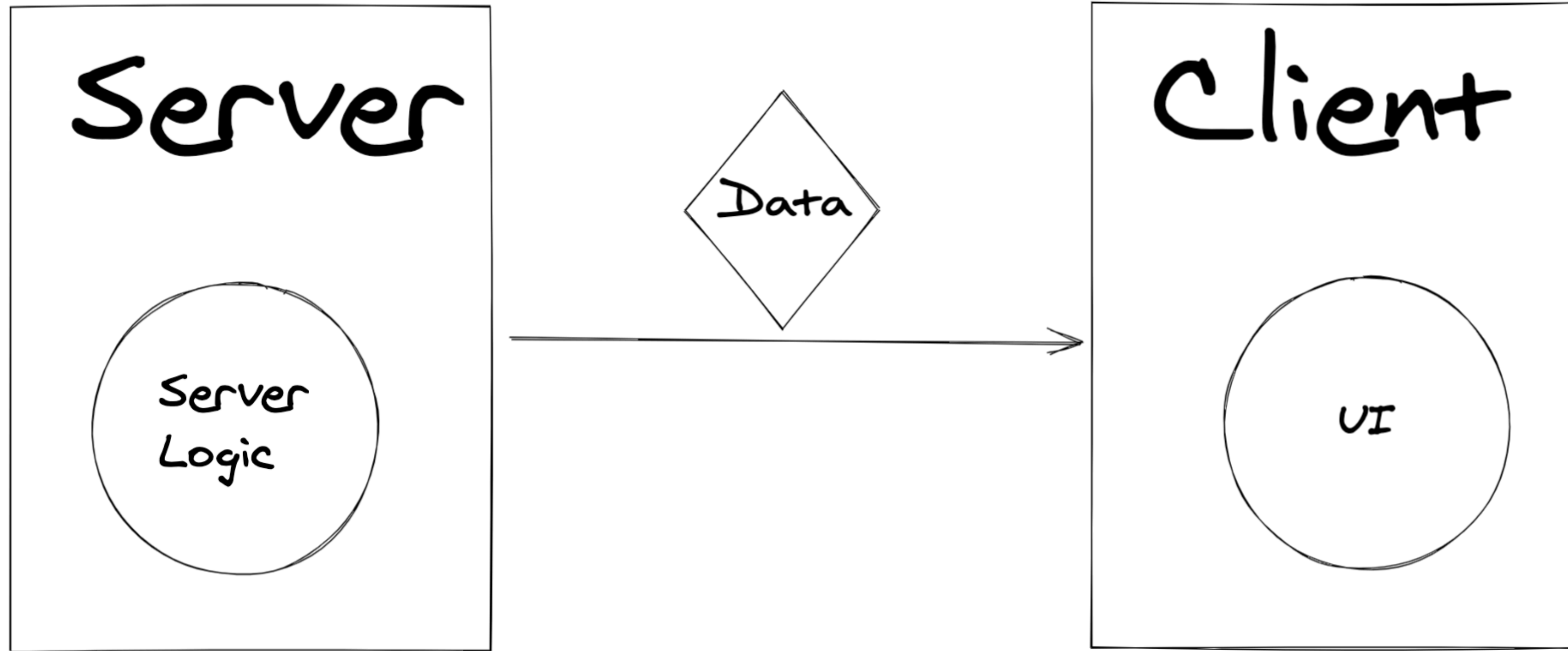
# Causality Flows - Timelines





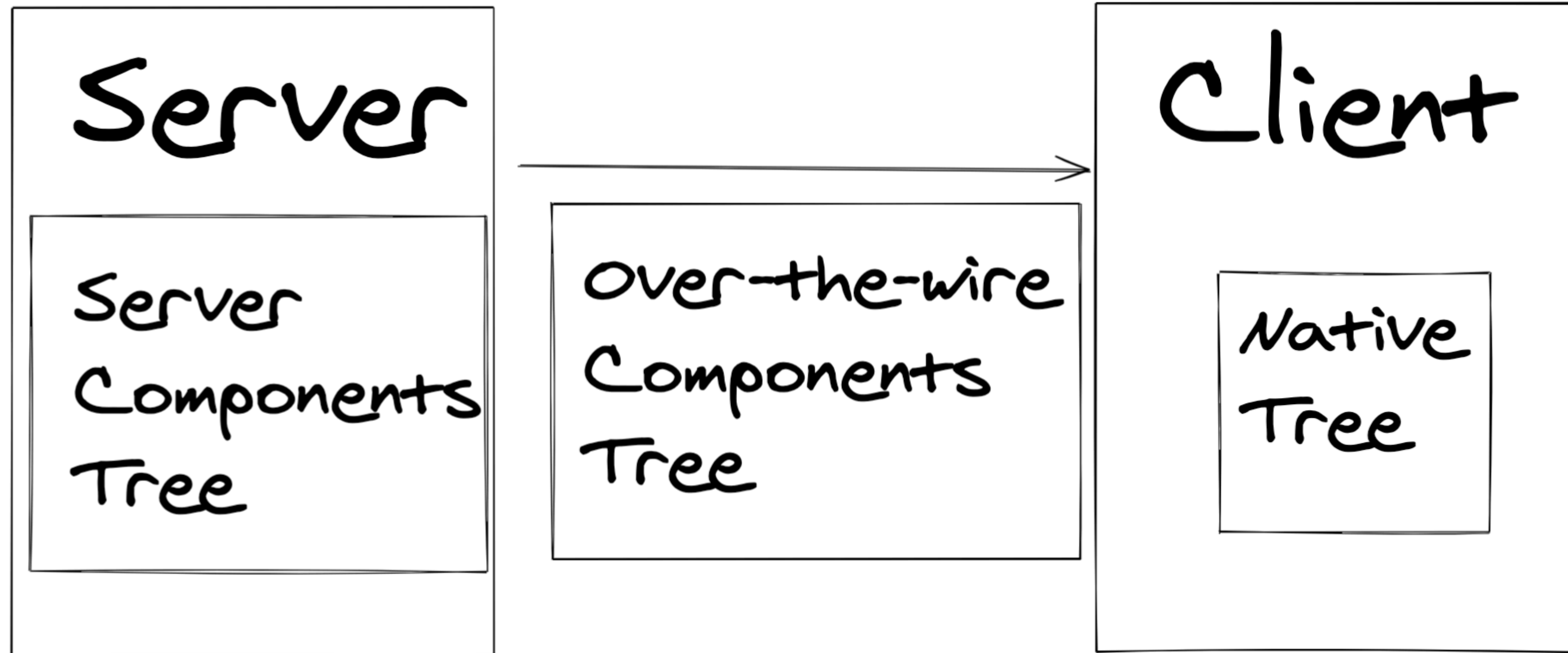
# Server Driven UI

Connecting server & native worlds



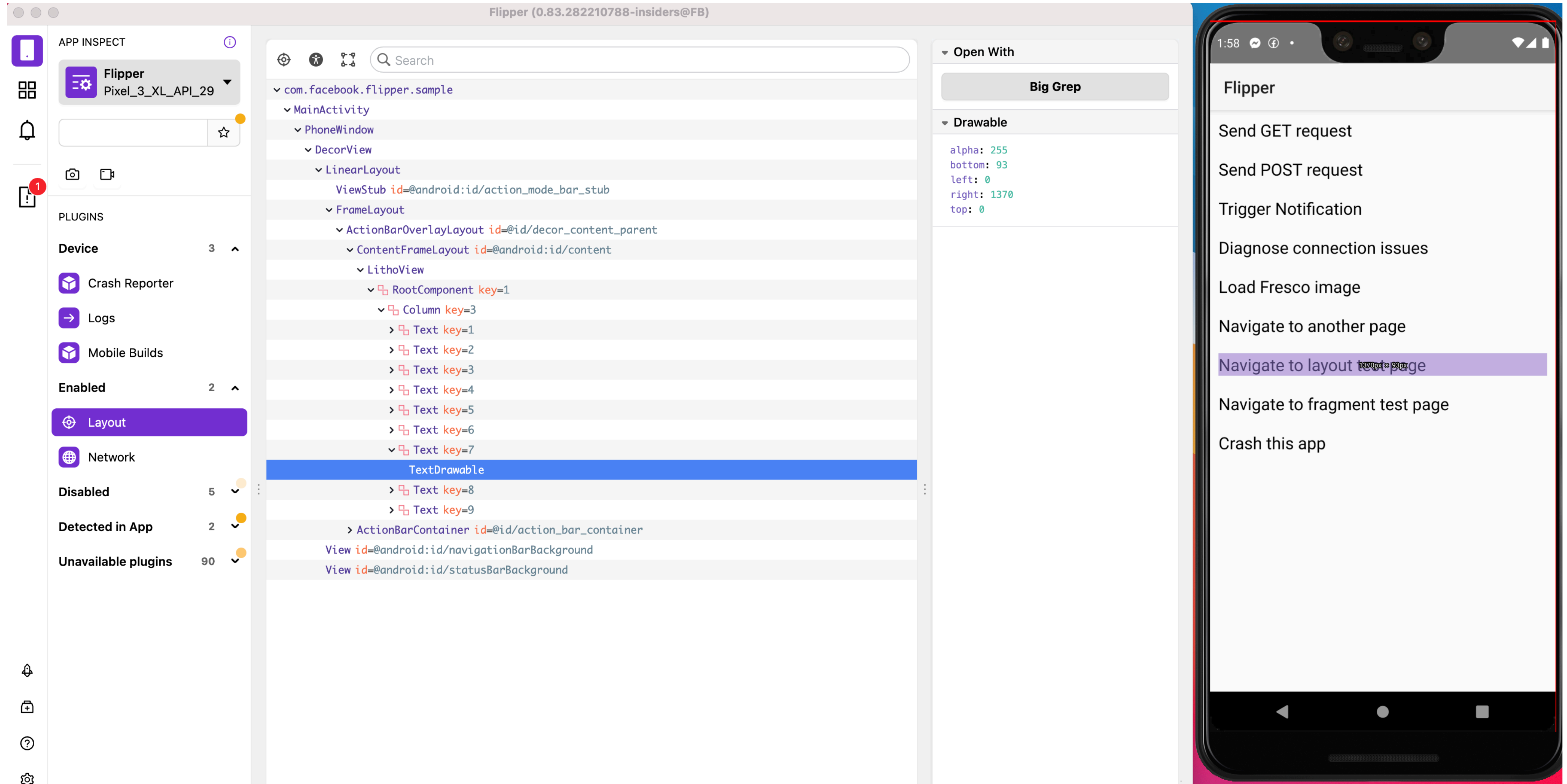
# Server Driven UI

Connecting server & native worlds



# Server Driven UI

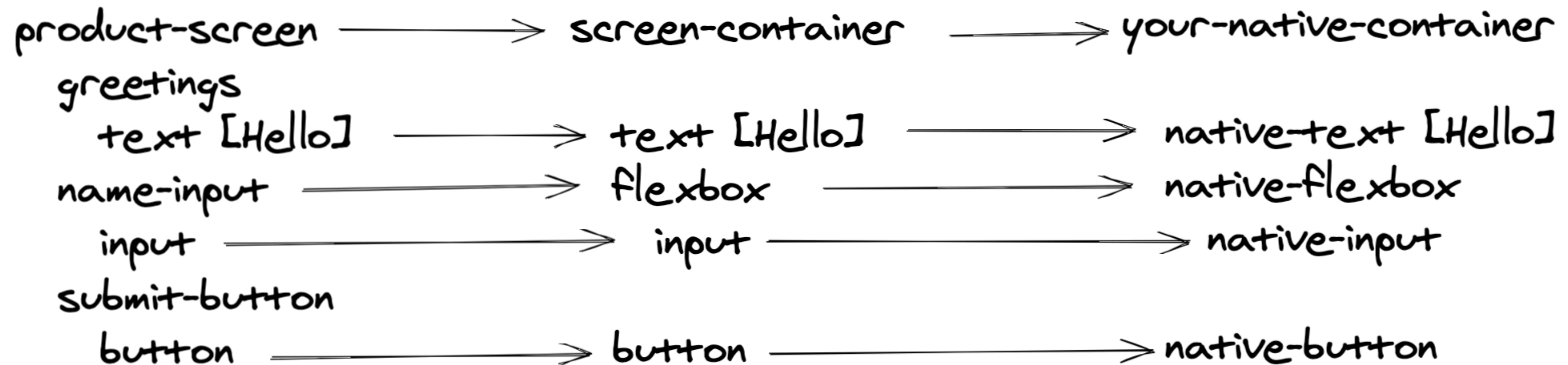
Connecting server & native worlds



# Server Driven UI

Connecting server & native worlds

Server over-the-wire Client



# Server Driven UI

Connecting server & native worlds

The image shows the Flipper (12.0.1-dev@FB) interface. The top section displays a table of created elements:

Created	Root Element	Product	Oncall	Elements	Bundle Size
12:45:14	:screen:wrapper			1	20.0 KB

Below this is a detailed component tree. The selected element is `components.Text`. The right-hand pane shows the details for this element:

**SELECTED ELEMENT**

- Primitive Element Info**
  - Element Style: `components.Text`
  - Subtree Elements: 1
  - Subtree Size: 203 B
- Primitive Element Attributes**
  - `style: bk.components.Text`
  - `text: "Hello, World!"`
  - `text_size: "16sp"`
  - `text_style: "normal"`
- Bindings & Scripts**
- COMMITTED VARIABLES**
  - `{}`
- Reduction Trace**
  - `:bk:screen:wrapper`
  - `:bk:flexbox`
  - `:bk:flexbox`
  - `:bk:shell:home:collection`
  - `:bk:shell:home:collection:container`
  - `:bk:collection`
  - `:bk:shell:home:section`
  - `:bk:flexbox`
  - `:bk:flexbox`
  - `:bk:shell:common:text`
  - `:bk:text`

On the right, a mobile app preview shows a screen with a list of items. The first item is "Hello, World!" with a subtitle "I'm a Mobius demo". Other items include "Examples", "Playground", and "Tooling".

# Flipper

## Do's and don'ts, pros and cons

- Don't ship it to prod! It's not just make your app heavier, it exposes a lot of stuff you don't want to share. Dummy instrumentation for prod builds might be your friend
- No more breakpoints and ad-hoc logging and long rebuilds!
- It might affect debug build's performance though
- Maintenance costs - who should own the tool?

# Building Dev Tools - How to approach?

## Identifying possible areas/targets

- Talk with your customers (other devs), find their pains
- Monitor support chats/groups
- Run user researches
- Keep in touch with infra builders




# Building Dev Tools - How to approach?

How to make sure that we are doing right things?

- Use data (if you can). Spoiler: it's hard.
- Human feedback
- User research: surveys, interviews



# Thanks!

- Docs: <https://fbflipper.com>
-  [@flipper\\_fb](https://twitter.com/flipper_fb)
-  [@mweststrate](https://twitter.com/mweststrate)
-  [@ruggerprogramm](https://twitter.com/ruggerprogramm)