

# Introduction to Mobile

See through your users' eyes

Zameer Fouzan  
Lead Developer Relations Engineer

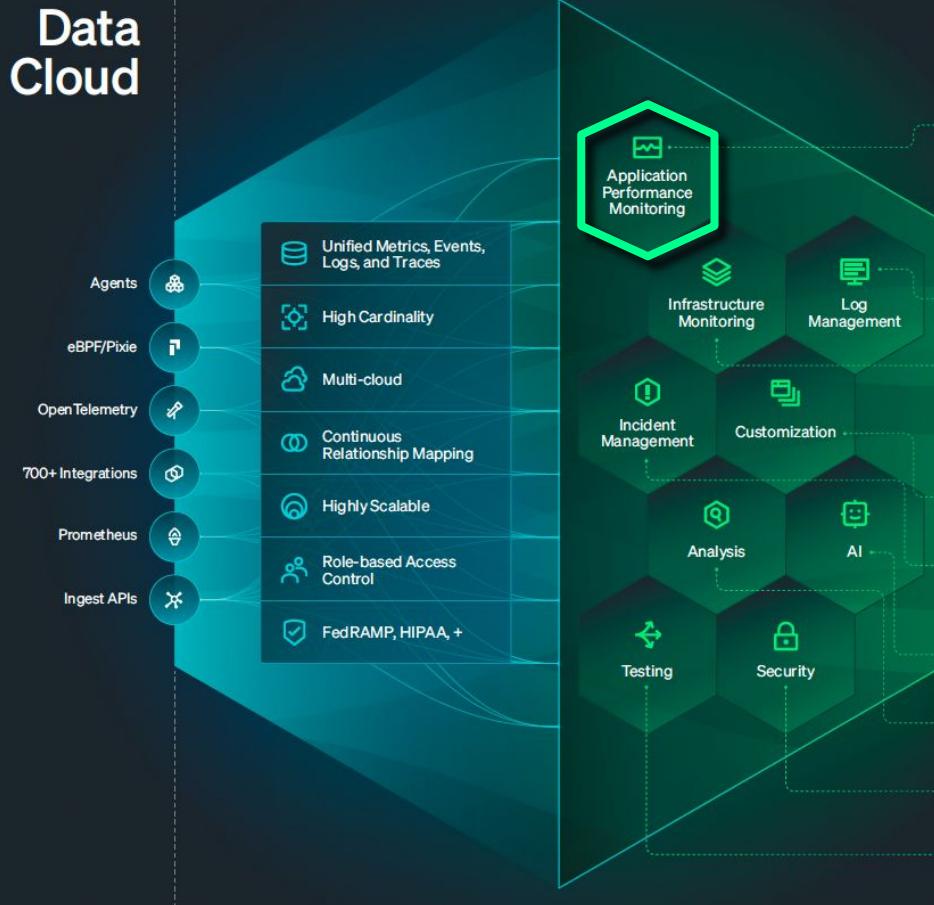
Jones Zachariah Noel  
Senior Developer Relations Engineer



# Agenda

- |    |                                    |         |
|----|------------------------------------|---------|
| 01 | Mobile Monitoring State of the Art | 10 mins |
| 02 | Features overview & Roadmap        | 15 mins |
| 03 | Platform demo                      | 20 mins |
| 04 | Hands-on Lab                       | 90 mins |
| 05 | Q&A                                | 10 mins |

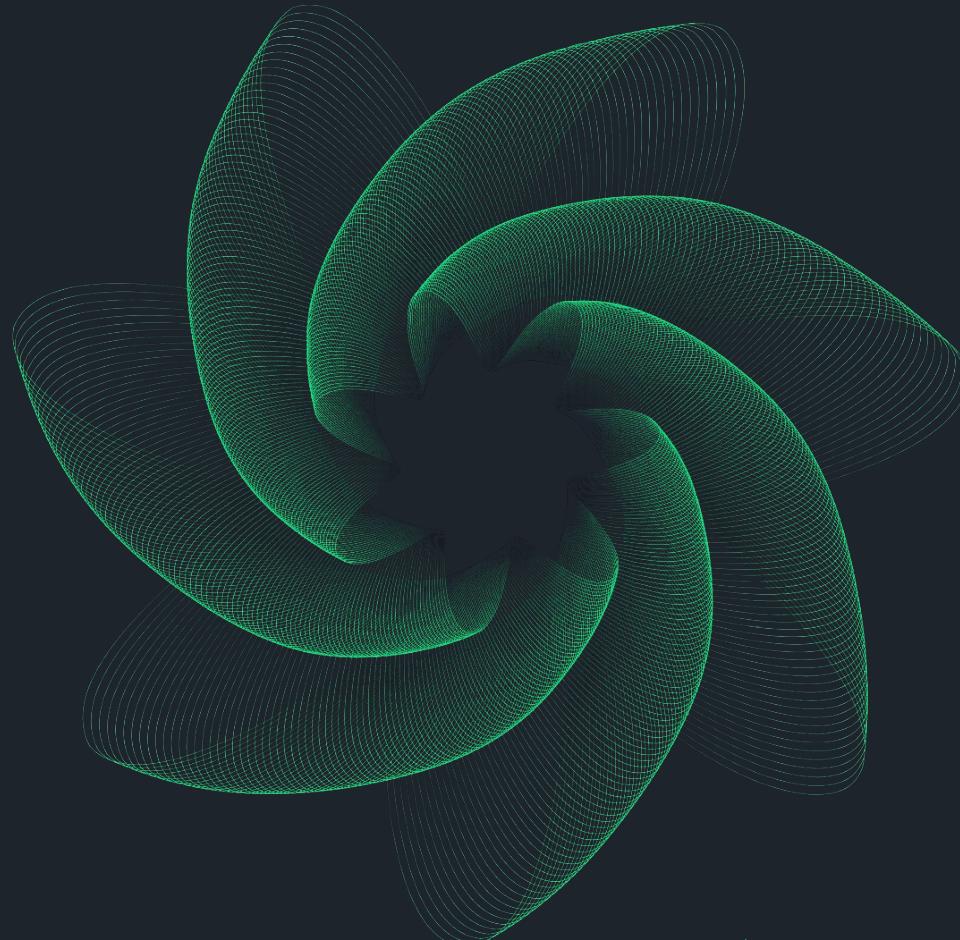
# New Relic All-in-one Observability Platform

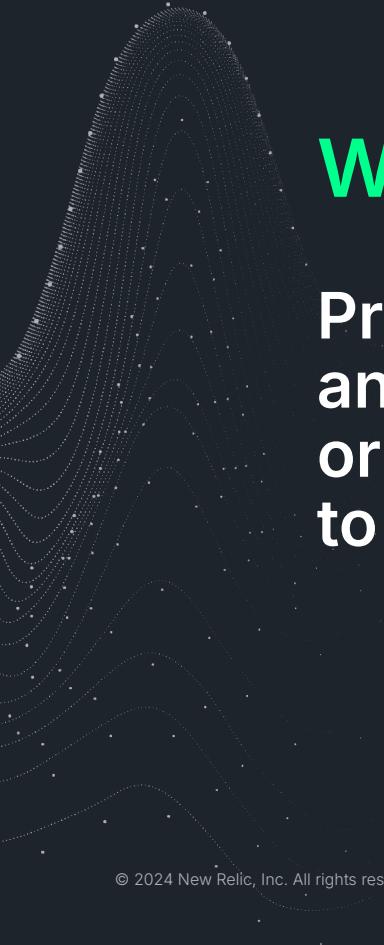


## Insight Cloud

- |                              |  |
|------------------------------|--|
| APM 360                      | Error Tracking                           |
| Distributed Tracing          | CodeStream                               |
| Session Replay               | Code Profiling                           |
| Browser Monitoring           | Service Maps                             |
| <b>Mobile Monitoring</b>     | CI/CD Monitoring                         |
| Website Performance          | Business Observability/Pathpoint         |
| SAP Monitoring               |  |
| Log Management               | Logs in Context                          |
| Live Archives                | Log Patterns                             |
| Host Monitoring              | GCP Monitoring                           |
| Container Monitoring         | Serverless Monitoring                    |
| Kubernetes Monitoring        | Network Monitoring                       |
| AWS Monitoring               | Database Monitoring                      |
| Azure Monitoring             | Kafka Monitoring                         |
| Custom Queries (NRQL & AI)   | Custom Apps                              |
| Customizable Dashboard       | APIs                                     |
| Alerts                       | Root Cause Analysis                      |
| AI Ops                       | Workflow Integrations                    |
| Change Tracking              | Notification Workflow                    |
| New Relic AI                 | OpenAI Monitoring                        |
| AI Monitoring                |  |
| Dashboards                   | Service Levels                           |
| Entity Explorer              | Service Catalog                          |
| Vulnerability Management     | Interactive Application Security Testing |
| Synthetic Browser Monitoring | Synthetic Scripted Monitoring            |
| Synthetic API Monitoring     | Continuous Automated Testing             |

# 01 Mobile Monitoring State of the Art





# What is Mobile Monitoring?

Provides deeper visibility into the performance and crash troubleshooting of your Android, iOS or hybrid mobile applications as well as gaming to improve user experience.

New App  
Downloads

**255**  
Billion

+11%

YoY Growth

iOS, Google Play, Third-Party  
Android in China

> 485,000 apps  
downloaded per  
minute in 2022

App Store  
Spend

**\$167**  
Billion

-2%

YoY Growth

iOS, Google Play, Third-Party  
Android in China

>\$318,000 spent per  
minute in 2022

Daily Time Spent  
Per User

**5**  
Hours

+3%

YoY Growth

Android Phones; Weighted  
Average Among Top 10 Mobile-  
First Markets

1/3 of daily  
waking hours

Mobile  
Ad Spend

**\$336**  
Billion

+14%

YoY Growth

Would be #44 largest  
country ranked among  
global economies

Total Hours  
Spent

**4.1**  
Trillion

+9%

YoY Growth

Android phones

11.2 Billion hours  
collectively per day  
in 2022

**data.ai (formerly AppAnnie)**  
State of Mobile 2023 Report

# New Relic Mobile Monitoring Provides Deeper Visibility so you can Quickly Understand:



**Application Performance** including app launches, crash analysis, HTTP errors, Handled exceptions, Release versions, etc.



**Correlate backend and frontend** easily connect user experience data to backend metrics with connected services when triaging issues



**Monitor iOS and Android as one**, and identify code-level errors with our hybrid and native framework support and make changes from one code base



**Version Trends** helps you understand the adoption and impact of change in your application

# Mobile Summary screen

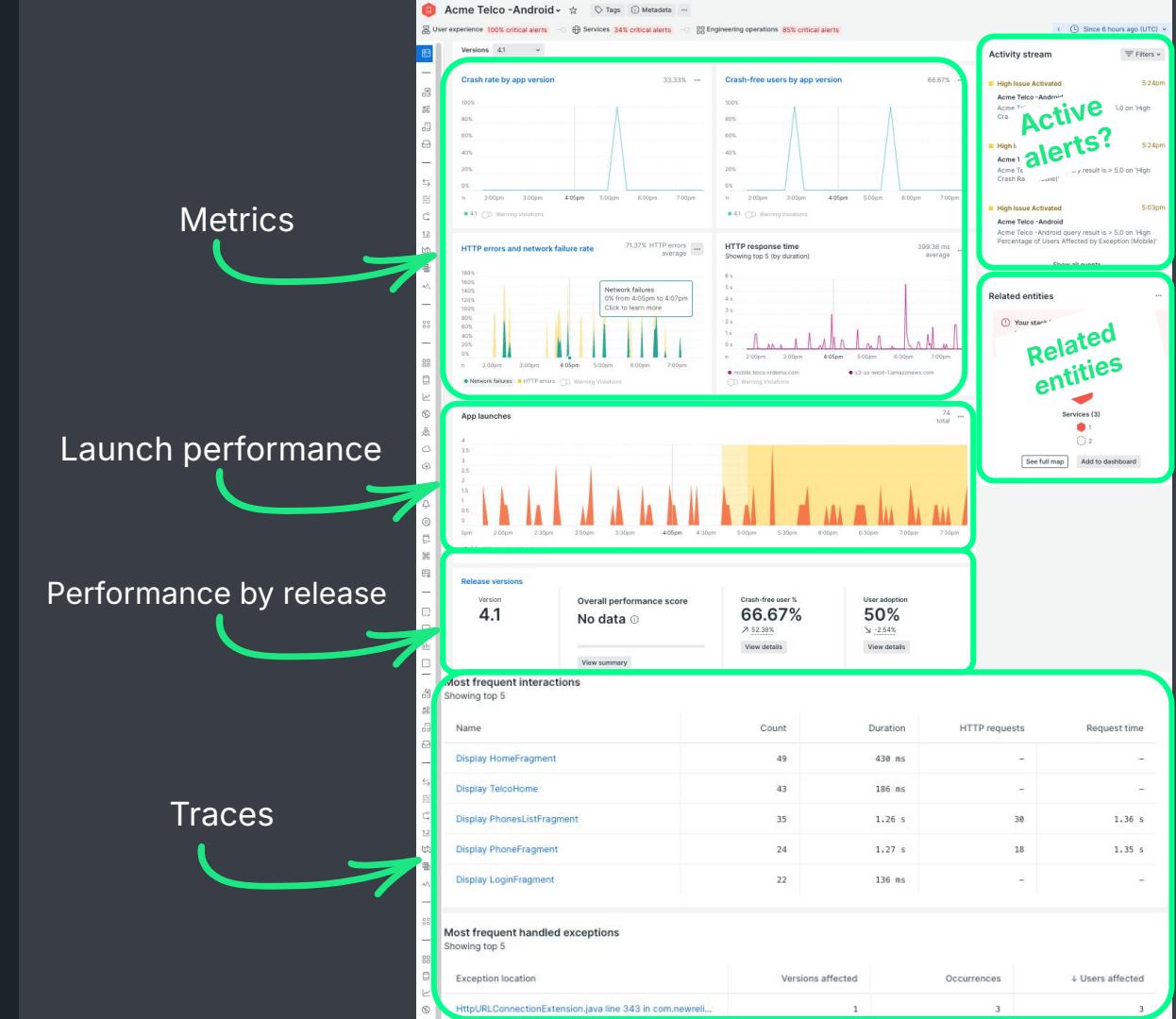
- Industry recognized metrics of performance
- Less screen swapping means reduced MTTx

Metrics

Launch performance

Performance by release

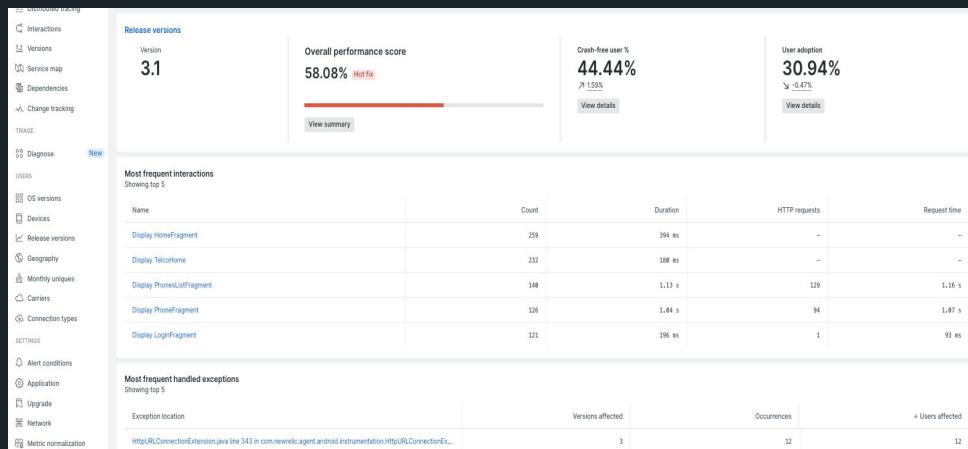
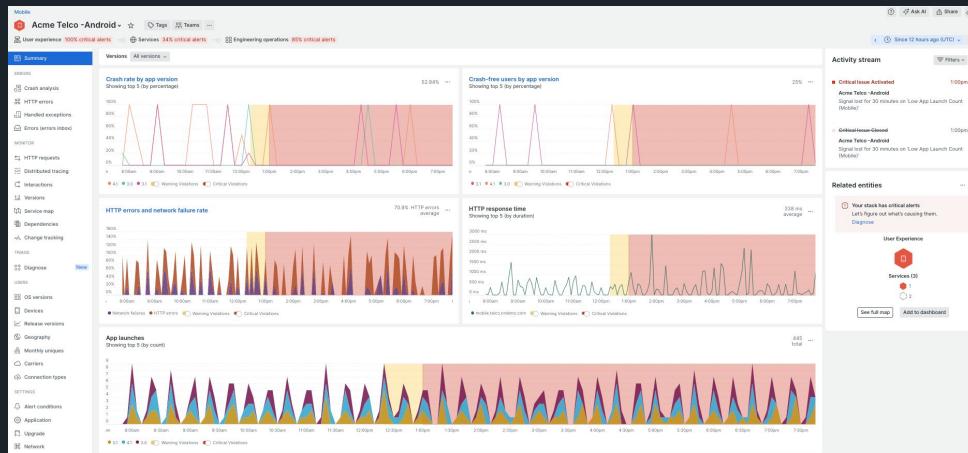
Traces



# Mobile Overview

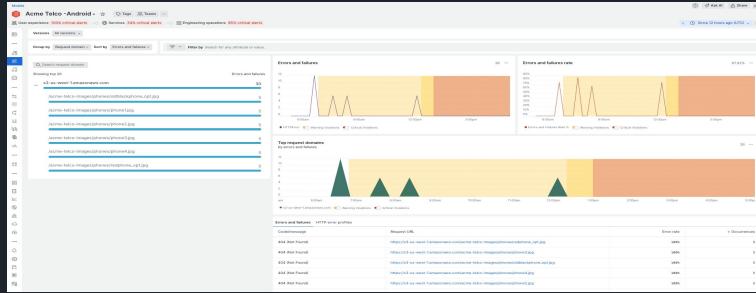
Track your entire cross-platform mobile stack in one view and correlate backend to frontend metrics to improve user experience

- View snapshot and deep-dive immediately into crashes, app launches, handled exceptions and the top 5 most frequent interactions.
- Monitor user behavior metrics and filter based on versions
- Examine HTTP and Network Performance to collaborate effectively with your backend teams.

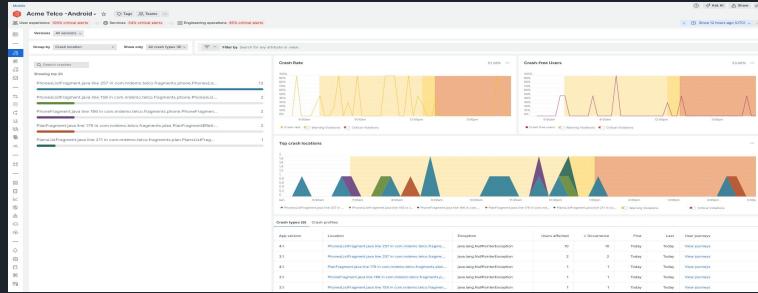


# Mobile Observability Capabilities - still one click away!

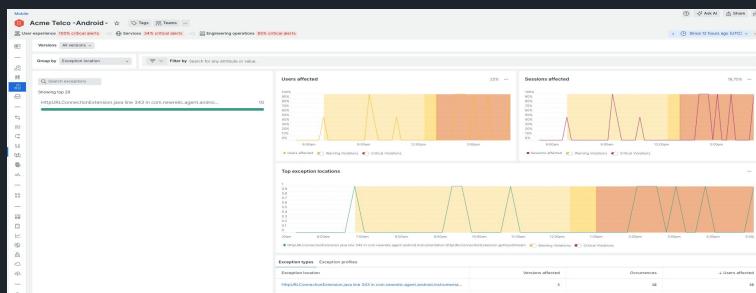
## HTTP Errors



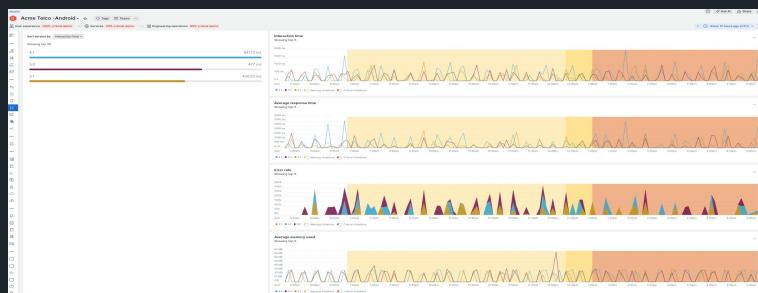
## Crash Analysis



## Handled Exceptions



## Versions

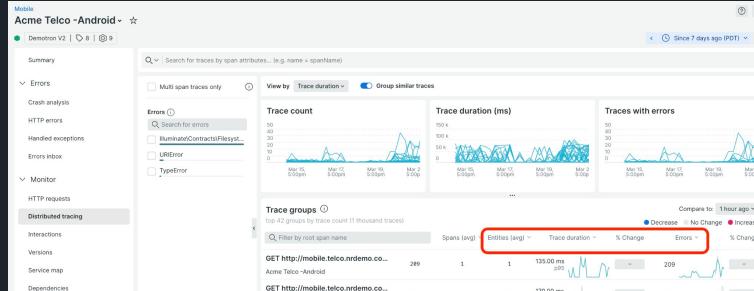


... and many more!

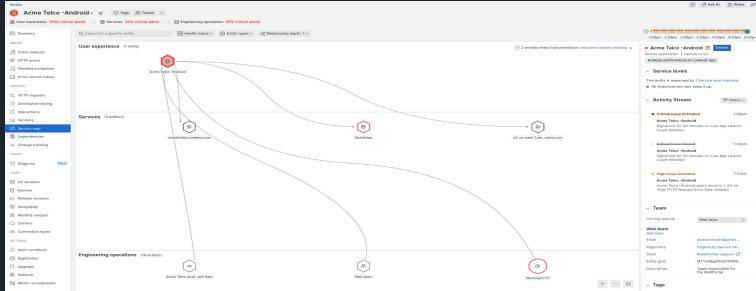
 new relic

# Mobile Observability Capabilities ... one click away!

## Distributed Tracing



## Service Maps



## Errors Inbox

Time	Service	Type	Description	Occurrences	Occurrences since 12 hours ago	Actions
Dec 1, 2023	Rolling back...	garker@...	garker@elice.ind...	-	0	0
Dec 1, 2023	HTTP Rule...	garker@...	garker@elice.ind...	-	0/1/14	100.00% (14/14)
Dec 1, 2023	Optimizing...	garker@...	garker@elice.ind...	-	0/1/14	100.00% (14/14)
Nov 30, 2023	Rolling back...	garker@...	garker@elice.ind...	-	0/1/14	100.00% (14/14)
Nov 30, 2023	HTTP Rule...	garker@...	garker@elice.ind...	-	0/1/14	100.00% (14/14)
Nov 30, 2023	Optimizing...	garker@...	garker@elice.ind...	-	0/1/14	100.00% (14/14)
Nov 29, 2023	Rolling back...	garker@...	garker@elice.ind...	-	0/1/14	100.00% (14/14)
Nov 29, 2023	HTTP Rule...	garker@...	garker@elice.ind...	-	0/1/14	100.00% (14/14)
Nov 29, 2023	Optimizing...	garker@...	garker@elice.ind...	-	0/1/14	100.00% (14/14)

## Change Tracking

Time	Version	User	Description	Changing	Type	Applies	Response to...	Throughput
Dec 1, 2023	Rolling back...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.47/min
Dec 1, 2023	HTTP Rule...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.43/min
Dec 1, 2023	Optimizing...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.37/min
Nov 30, 2023	Rolling back...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.33/min
Nov 30, 2023	HTTP Rule...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.30/min
Nov 30, 2023	Optimizing...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.27/min
Nov 29, 2023	Rolling back...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.23/min
Nov 29, 2023	HTTP Rule...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.19/min
Nov 29, 2023	Optimizing...	garker@...	garker@elice.ind...	-	-	0/1/14	100.00% (14/14)	20.15/min

## 02 Installation

# What can you report?

Crash Analysis

Handled Exceptions

Version Trends

HTTP Requests & Errors

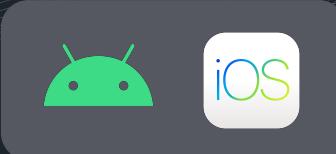
Interactions

Distributed Tracing

Log forwarding NEW!

ANR (android only)

FOR 



Native (includes  
Android TV, tvOS  
and iPadOS)



Cross-platform,  
non-JS based



Cross-platform, JS  
based

# How to install?



Gradle Dependency

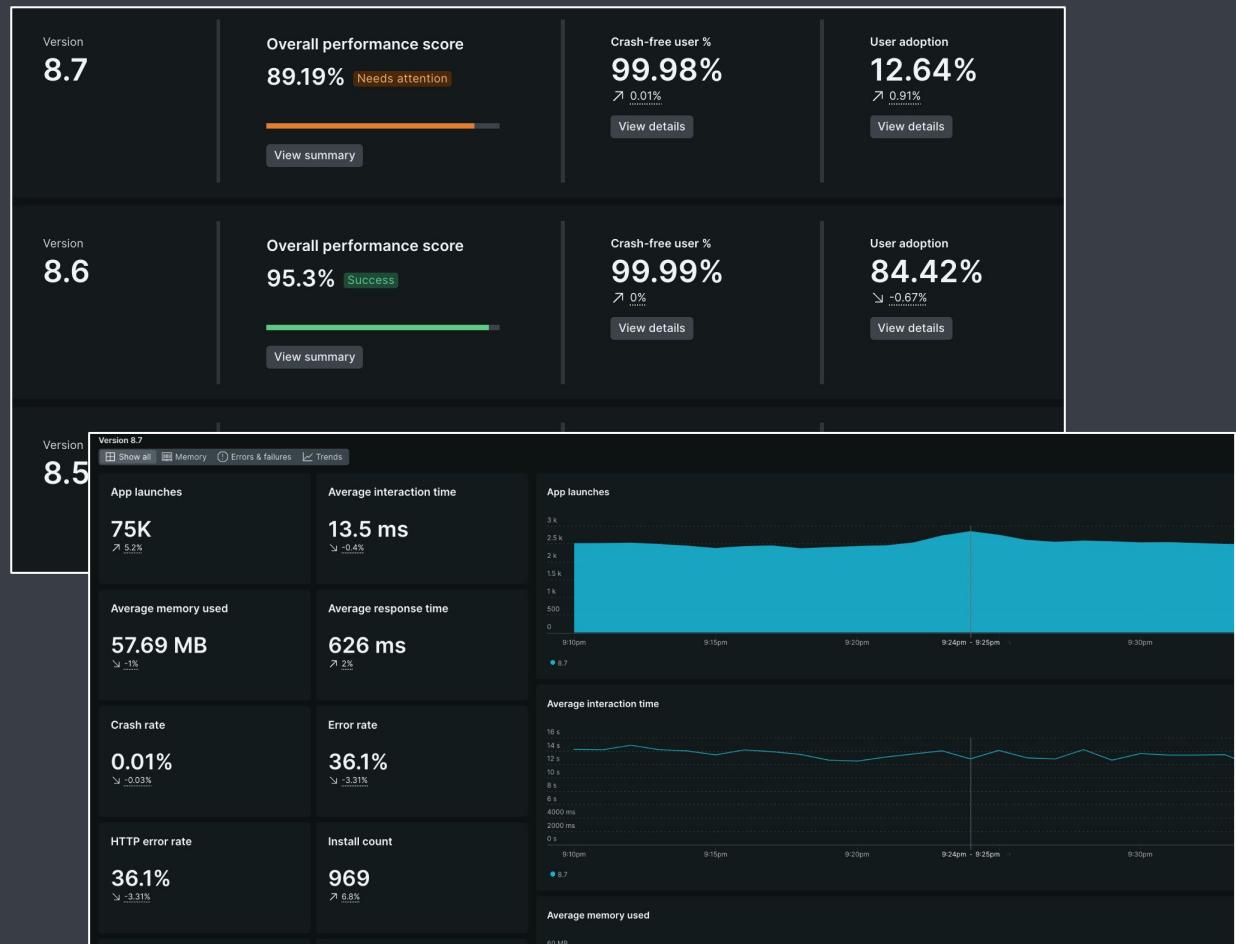


Swift Package Manager  
CocoaPods

## 03 Features overview

# Versions

View crash rates, user engagement and overall performance metrics between different release versions.



# Crashes

A unified view for all the errors that ended up crashing the app.

Get access to the stack trace, the attributes, sync with your team with our Jira integration and get an estimate of the number of impacted users.

Error groups (11) ⓘ

PlanFragment.java line 143 in com.nrdemo.telco.fragments.plan.PlanFragment\$1.onClick  
java.lang.NumberFormatException  
Acme Telco - Android | almost 2 years old - last seen about 1 hour ago

PlansListFragment.java line 211 in com.nrdemo.telco.fragments.plan.PlansListFragment\$PlansRecyclerViewAdapter.getItemCount  
java.lang.NullPointerException  
Acme Telco - Android | about 1 hour ago

PhoneFragment.java line 11 in com.nrdemo.telco.fragments.phone.PhoneFragment\$1.onClick  
java.lang.NullPointerException  
Acme Telco - Android | over 1 year ago

LoginFragment.java line 11 in com.nrdemo.telco.fragments.login.LoginFragment\$1.onClick  
java.lang.NullPointerException  
Acme Telco - Android | about 1 hour ago

PhonesListFragment.java line 11 in com.nrdemo.telco.fragments.phones.PhonesListFragment\$1.onClick  
java.lang.IllegalStateException  
Acme Telco - Android | about 1 hour ago

Start a new discussion

Error group metrics

Users impacted

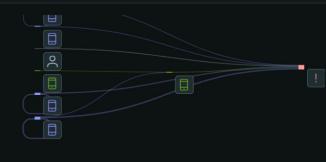
First seen almost 2 years ago in  
Unknown app version

Last seen about 1 hour ago in  
3.0

account:Demotron V2 accountid:1606862

User journeys New

View paths leading up to this crash



Occurrences 106 occurrences ⓘ



Occurrence details

Timestamp July 15, 2024, 8:33pm

Location PlanFragment.java line 143 in com.nrdemo.telco.fragments.plan.PlanFragment\$1.onClick

Exception java.lang.NumberFormatException

Message For input string: "129269472958216879119677"

App Version 3.0

App Build ID 3

Show more attributes

Stack trace

Not what you expected? If some of your frames are obfuscated, we're missing your mapping file. Upload it now. Upload missing mapping file See our docs ⓘ

Thread 2 — main

java.lang.Integer	Integer.java line 618 in java.lang.Integer.parseInt()
java.lang.Integer	Integer.java line 801 in java.lang.Integer.valueOf()
com.nrdemo.telco.fragments.plan.PlanFragment	PlanFragment.java line 143 in com.nrdemo.telco.fragments.plan.PlanFragment\$1.onClick()
android.view.View	View.java line 7259 in android.view.View.performClick()
android.view.View	View.java line 7236 in android.view.View.performClickInternal()
android.view.View	View.java line 801 in android.view.View.access\$3600()
android.view.View	View.java line 7709 in android.view.View.performClick()

Event trail

Sort by Newest

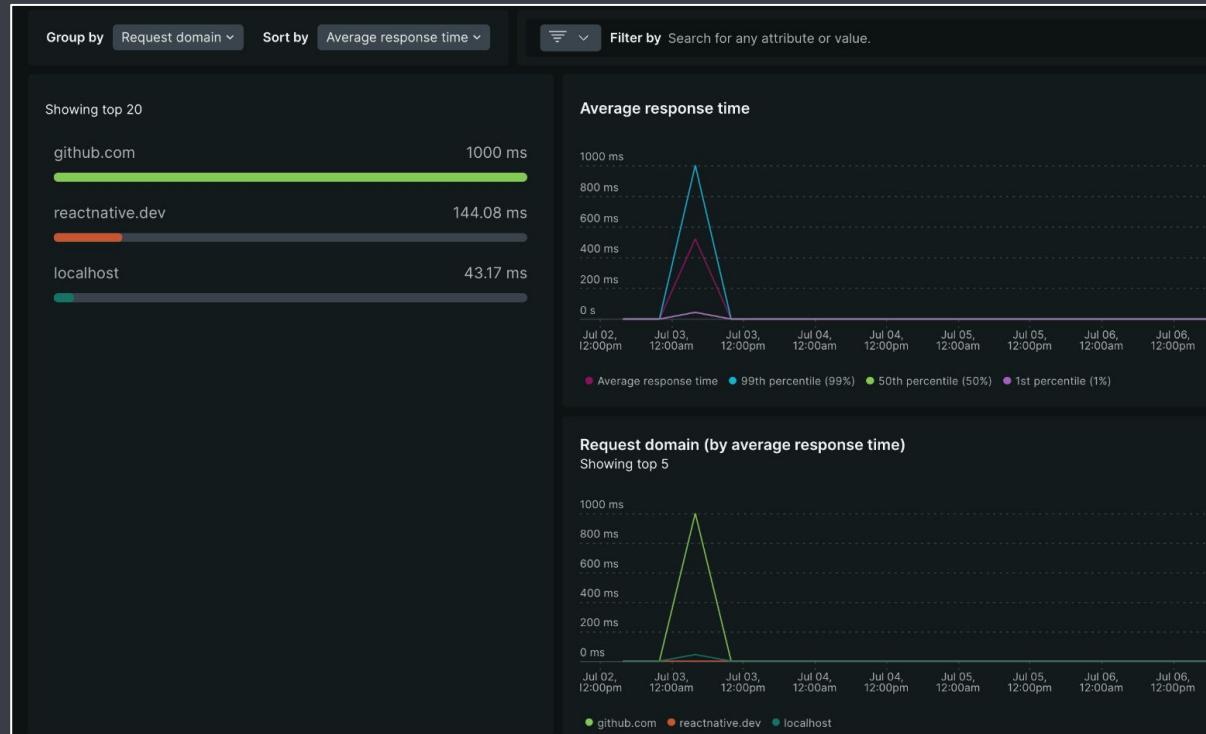
App Launch Interaction Custom Request Crash

Jul 15, 2024 20:33:27.043	Crash: For input string: "129269472958216879119677"
Jul 15, 2024 20:33:08.764	Custom: name: Atomic Tangerine Lite

# HTTP Requests

What is your app calling? Is it taking too much time?

Create customised charts to assess the footprint external services leave in your app. If you want to investigate further, distributed traces are available for you to continue your research.



# HTTP Errors

Focus on failed HTTP requests and obtain key information about what, when and why they didn't succeed.

Browse across occurrences, check the stack trace and get a glimpse of the user cohorts that got impacted the most.

Filter by  Search for any attribute or value.

### Error summary

Error	Unknown
Request URL	<a href="https://placekitten.com/200/300">https://placekitten.com/200/300</a>
Affected versions	1.0
Users affected	1
Occurrences	1

### Occurrences by version

Version	Occurrences
1.0	1

### Breakdown of error types

Type	Occurrences
1.0	1

### Occurrences

Timestamp	July 3, 2024, 6:36am
Error	Unknown
Request url	<a href="https://placekitten.com/200/300">https://placekitten.com/200/300</a>
Affection version	1.0

### Attributes

11111 Unknown responseBody	1
----------------------------	---

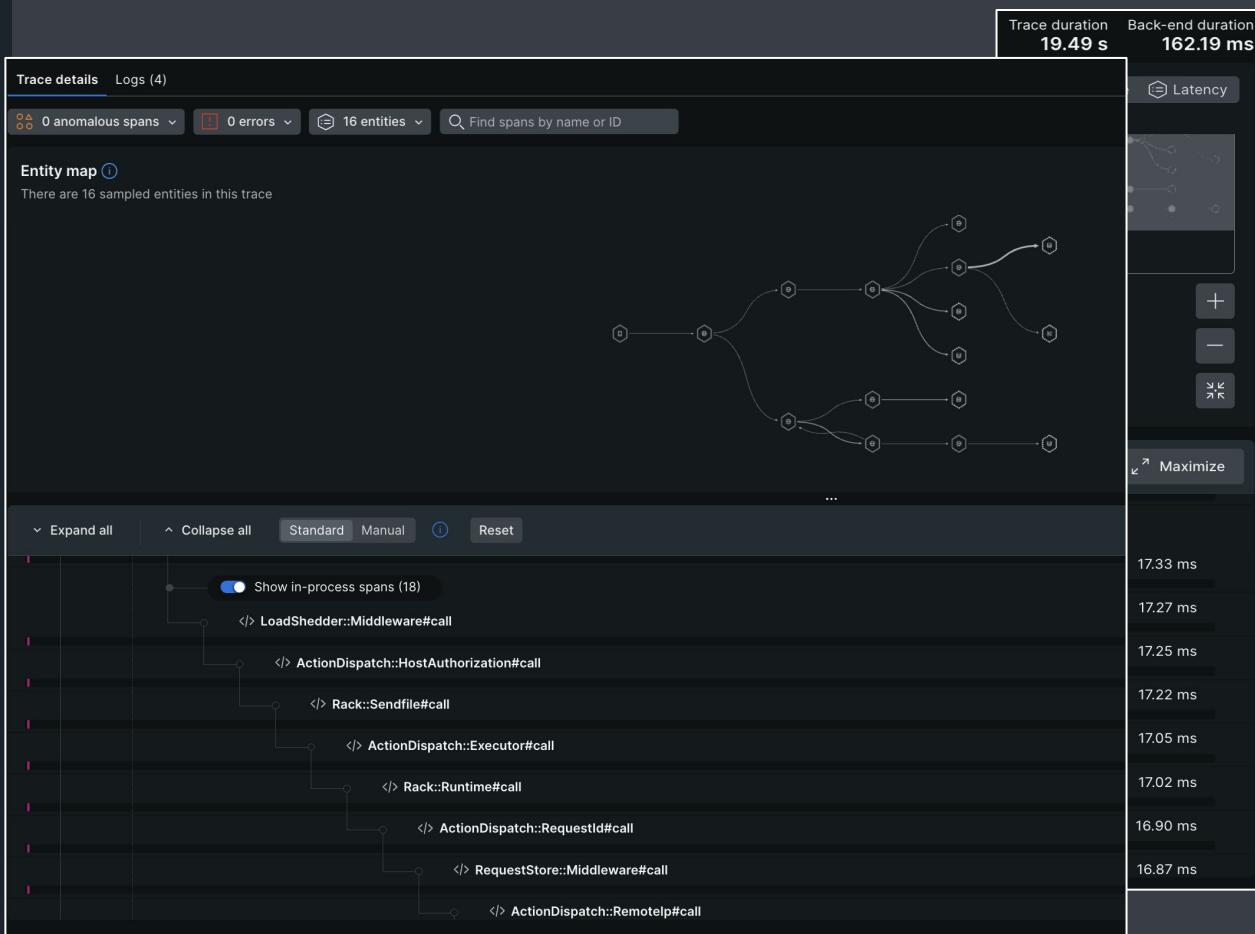
1 This response body may be truncated. There is a 4kb limit on the storage of responses.

```
1 <!DOCTYPE html>
2 <!--[if lt IE 7]>
3 <html class="no-js ie6 oldie" lang="en-US">
4 <![endif]-->
5 <!--[if IE 7]>
6 <html class="no-js ie7 oldie" lang="en-US">
7 <![endif]-->
8 <!--[if IE 8]>
9 <html class="no-js ie8 oldie" lang="en-US">
10 <![endif]-->
11 <!--[if gt IE 8]>
12 <!-->
13 <html class="no-js" lang="en-US">
14 <!--
15 <![endif]-->
16 <head>
17 <title>placekitten.com | 521: Web server is down</title>
18 <meta charset="UTF-8" />
```

# Distributed Tracing

See your app in your whole stack's context.

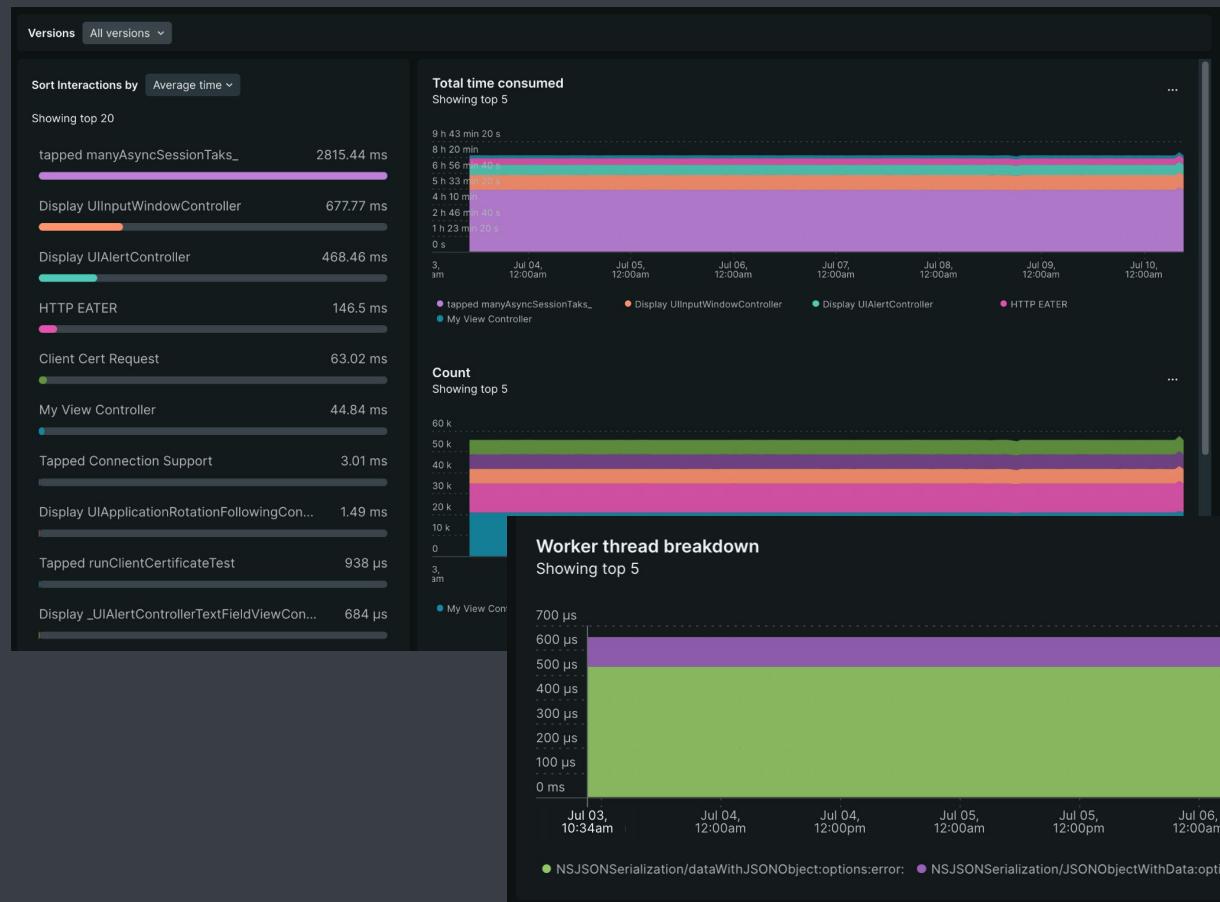
Check response times, bottlenecks and error reports to see what your customer doesn't see – but experiences.



# Interactions

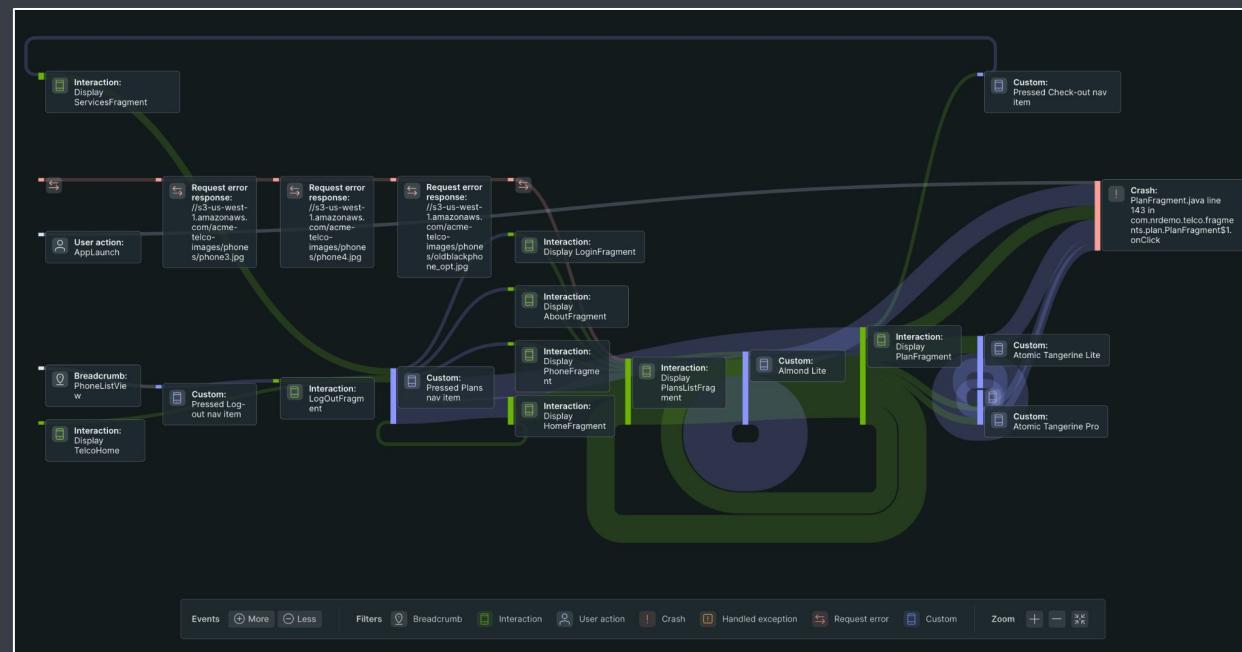
Is it the home page? Is that API call happening in the background while you're signing in?

Quickly assess where time is spent when loading resources, analyse thread usage and profile your app with real user data.



# User-Journey

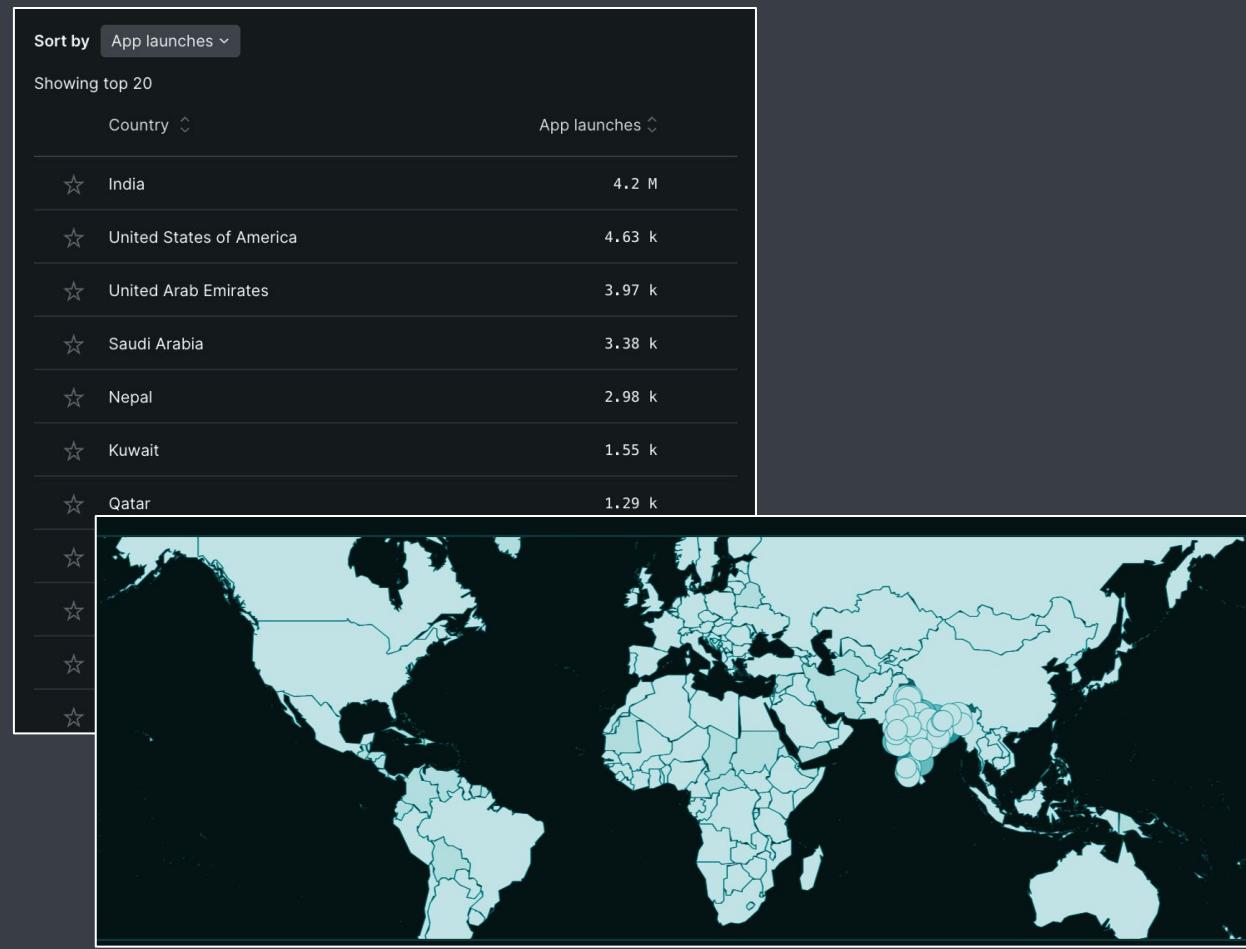
Allows you to see through the user's eyes and follow what paths lead to crashes, and how to remediate them. It aggregates the paths taken across all your users



# Geography

Shows Mobile users' experience which can be filtered by versions as a world view as well as the ability to drill down to detailed information about each country's

- App launches
- Color-coded response times
- Network Requests (calls per minute)
- Total Transfer size
- Network Failure Rates

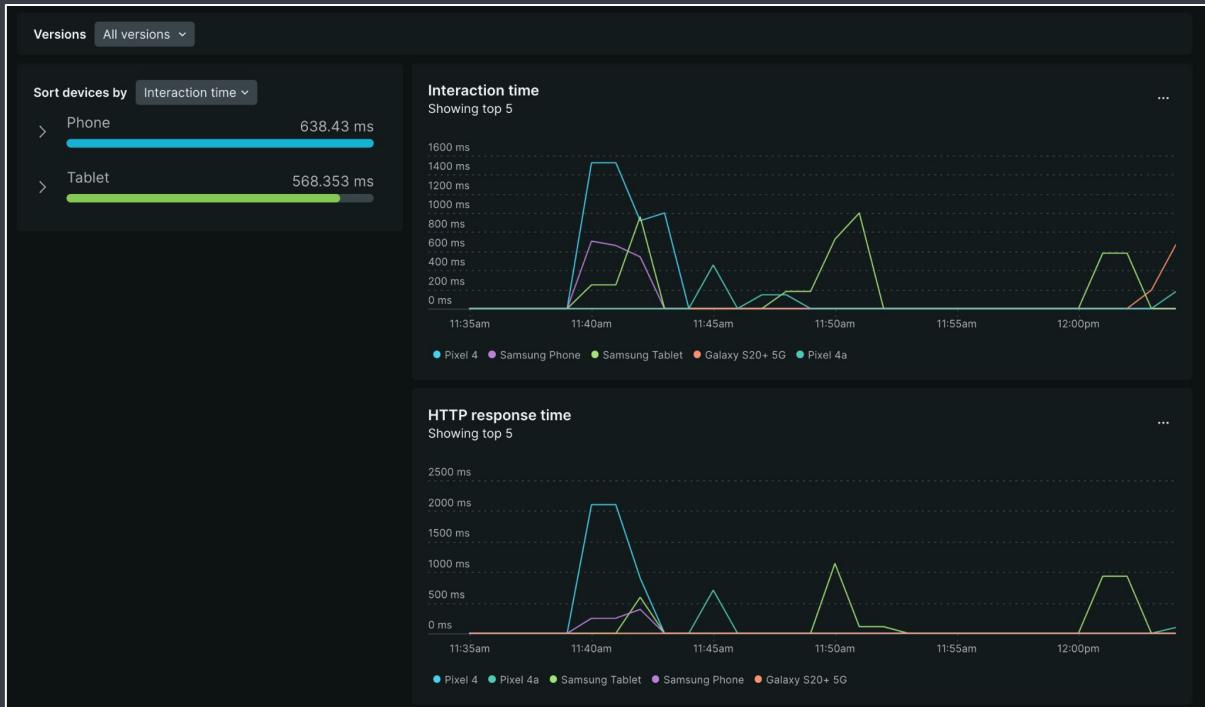


# Devices

Measure per-device adoption and troubleshoot hardware-related issues such as errors happening only on certain viewports, with lower end phones...

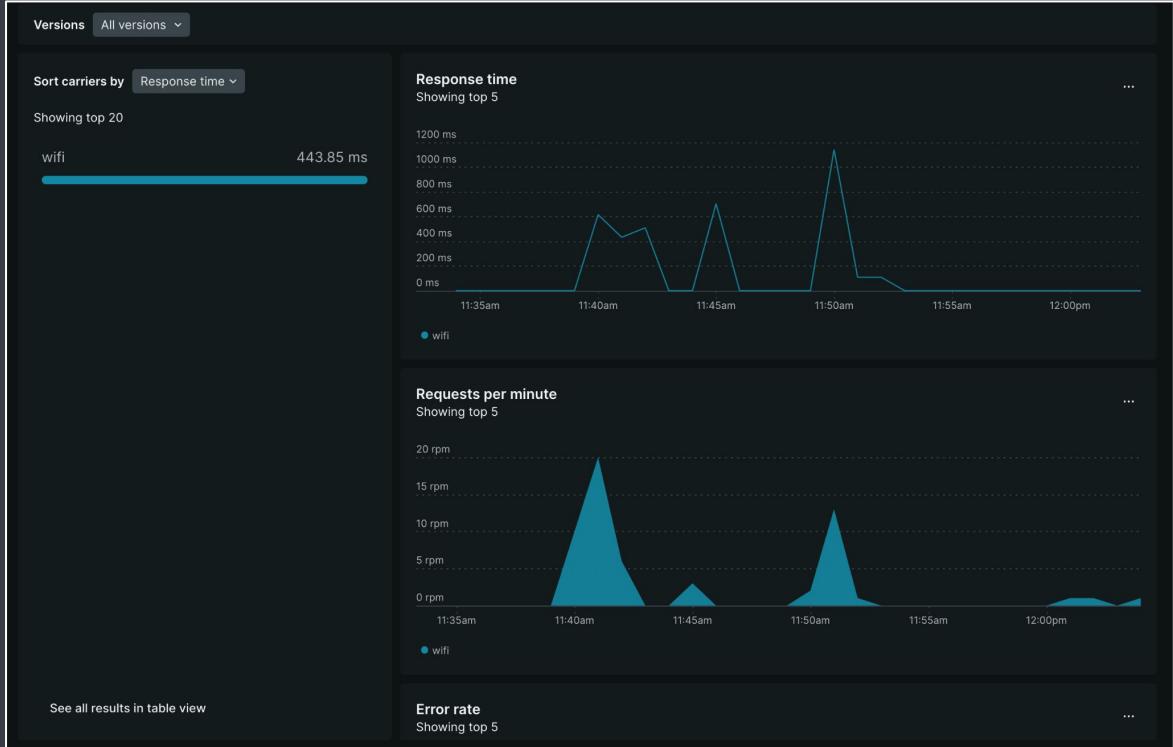
Measures available:

- Interaction time
- HTTP Response Time
- Requests per minute
- Error rate
- App launches



# Carriers

In front of an issue, confidently answer the “is it the network?” question by grouping all sorts of performance metrics by carrier or Wi-Fi.



# What is Mobile Logging?

## The problem

Client's want to collect logs during app executions, crashes, ANR, and HTTP requests to troubleshoot issues. In the past we needed multiple SDKs to complete the task. This created application bloat and sometimes undesirable side effects.

## The solution

New Relic has added Logging to the Native SDK.

- For Android there are 2 methods to collect logs:
  - android.util.Log default android logging
  - New Relic Log API
- For IOS Native logs:
  - The iOS agent collects all logs written on stdout and stderr from the app. It captures all calls to NSLog(), Swift print(), and other print-family functions. The logs collected this way are set to NRLogLevelInfo.
  - New Relic Log API

# What is An ANR?

## The problem

ANRs (Application Not Responding) are a problem because the app's main thread, which is responsible for updating the UI, can't process user input events or draw, causing frustration to the user. See this [Doc](#) from Google for more info

## The solution

The Android agent reports ANR errors as `MobileApplicationExit` events, which leverage Android's `ApplicationExitInfo` API to monitor and report on the different causes for app termination. The agent reads the `ApplicationExitInfo` API on the next app launch and asynchronously sends `MobileApplicationExit` events to New Relic for each app exit in the history.

# What is Mobile Background Reporting ?

## The problem

Prior to this update when the application was backgrounded the Mobile SDKs stopped collecting data and started a new session.

## The solution

When the application is backgrounded it will continue to collect data and send data to New Relic

# What is Offline Storage?

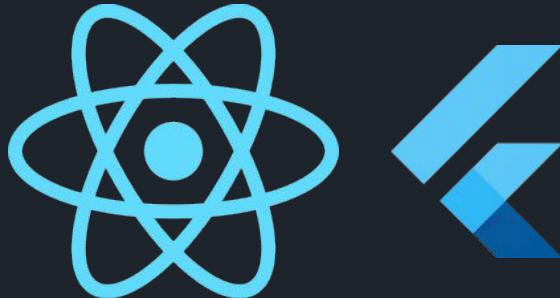
## The problem

When the New Relic Mobile SDK does not have access to the internet, The agent made 3 attempts to send the data. This lead to a loss in data to at a critical time.

## The solution

- Offline storage will store data and keep attempting to send it until it is sent.
- By default, mobile monitoring can collect a maximum of 100 megaBytes of offline storage. When a data payload fails to send because the device doesn't have an internet connection, it can be stored in the file system until an internet connection has been made. After a typical harvest payload has been successfully sent, all offline data is sent to New Relic and cleared from storage.
- The storage is configurable but realize the limitations of the devices the clients are using.

# What are Hybrid Mobile Agents?



# Flutter



# Mobile Roadmap

 = Now GA

## Recently delivered

### **Android ANRs**

Detailed data and alerting on application exits, ANRs.

### **WatchOS**

Official WatchOS support

### **Logs in Context**

Log telemetry directly from user interactions in context with performance data.

### **Mobile Crash Inbox**

Triage and debug crashes and errors with best in class Error Tracking capabilities.

## FY26 Q1

### **Mobile Session Replay LP**

See exactly what users saw while using your app. Troubleshoot issues without reproducing locally and evaluate UX opportunities.

### **ANR stack trace symbolication**

View the symbolicated stack traces (Java stack frames) of ANRs to see the code responsible for the app becoming non-responsive.

### **HTTP Errors in Errors Inbox**

Triage your HTTP errors alongside crashes and with the benefit of workflows, User Journeys, Logs, replays etc.

## FY26 Q2 and beyond

### **Mobile Views**

Analyze throughput, performance, and issue rates of each view your app

### **User Journeys - View flow**

Visually analyze how users in aggregate navigate your app. How do users get from point A to point B?

### **User Frustration Signals**

Discover when and where users are rage clicking and hitting dead clicks/taps on Browser and Mobile apps.

### **Performance Inbox**

Proactively detect and triage performance issues (e.g. N+1 queries, slow DB queries, etc).

# Mobile Roadmap

## Recently delivered

### Session Replay - Limited Preview

**Select customers get free access.** See exactly what users saw while using your app. Troubleshoot issues without reproducing locally and evaluate UX opportunities. Supported: iOS UIKit, Android XML layouts, React Native

### Logs in context

Automatic collection of logs surfaced in-context where they're most helpful. Server-side control of log level and sampling rate.

### Crash, Request Error, ANR Inbox

Triage and troubleshoot crashes, request errors, and ANRs in a unified error tracking experience with users impacted, symbolicated stack traces, session replays, logs, profiles, and user journeys.

## 2025

### Session Replay - Public Preview

**All customers get free access.**

New capabilities for PP:

- Server-side masking overrides
- Capture of SwiftUI and Jetpack Compose views
- Sessions persist if backgrounded for less than 30 minutes
- Errored-session sampling. Only collect replays of your app breaking.

## 2026

### Session Replay - GA

New capabilities for GA: Capture of WebViews, API to start/pause replays.

### Mobile Views & User Journeys

Analyze throughput, performance, and issue rates of each view your app. Visualize how users in aggregate navigate your app. How do users get from point A to point B?

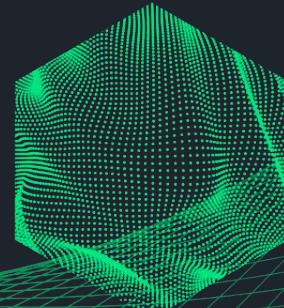
### User Frustration Signals

Discover when and where users are rage clicking and hitting dead clicks/taps on Mobile apps.

### Web sockets

Monitor your web sockets data for timeouts, lag, dropped connections, bottlenecks and more to improve users' real time experiences.

# 04 Demo



# Hands on Lab

## Prerequisites

- **Personal New Relic account.**
  - You will need to be able to access the Ingest API Key



Switch to [Desktop site on mobile](#)

# Hands on Lab

Explore the repository [\*\*github.com/zachjonesnoel/android-monitoring\*\*](https://github.com/zachjonesnoel/android-monitoring)

1. Clone the Android app
2. Test the app
3. Instrument the app with New Relic
  - a. HTTP Requests
  - b. Custom Attributes
  - c. Handled Exceptions
  - d. Custom Events
  - e. ANRs
  - f. Crash Analytics
  - g. Breadcrumbs
4. **BONUS:** Build a custom dashboard



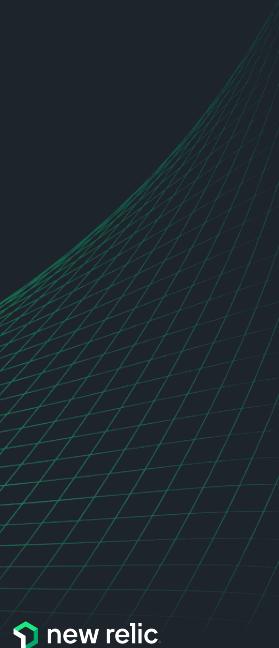
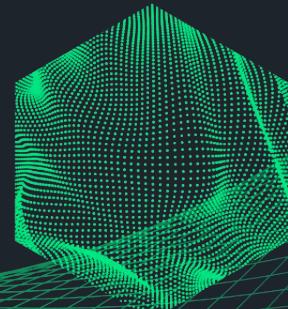
Clone the repo

# 05 Q&A

**A reminder of what we covered today:**

- Mobile capability
- Installation
- User analysis
- Demo

# 06 Next steps



# Resources

## Webinars and Q&A sessions:

- [Weekly Webinars and Q&A](#)
- [NRU Course - 2023-22-03 recording: NR Mobile & React Native](#)

## Blog posts:

- [Mobile app monitoring in minutes with observability-driven dashboards](#)
- [Mobile APM: Android and iOS monitoring](#)
- [Optimizing React Native app Performance](#)
- [Boost UX: Monitor Flutter Mobile apps efficiently](#)
- [Customer story: SkyScanner](#)
- [Monitor hybrid mobile apps in minutes with quickstart installations](#)
- [Understanding Mobile Application Security](#)
- [Building Mobile Application Performance Stories with Breadcrumbs](#)
- [Cross-platform support with React Native and Mobile Monitoring](#)

## Documentation links:

- [Introduction to Mobile Monitoring](#)
- [Docs for installing the React Native agent](#)
- [Mobile quickstart guide](#)
- [NRQL Query Examples for Mobile Monitoring](#)
- [New Relic Android Mobile Monitoring with Distributed Tracing](#)
- [Install the Android Agent](#)
- [Get started with Android Monitoring](#)
- [Install and get started with iOS Monitoring](#)
- [Mobile Best Practices](#)
- [Handled Exceptions: Analyze trends, prevent crashes, investigate occurrences](#)
- [HTTP Errors: Analyze Network Failures](#)
- [Version Analysis](#)
- [Monitor your React Native Mobile app](#)

## DataBytes/Youtube videos:

- [Monitor your mobile applications with React Native](#)
- [Mobile Distributed Tracing](#)
- [Troubleshoot mobile crashes with crash analysis](#)

**New Relic Docs**  
[docs.newrelic.com](https://docs.newrelic.com)

**Explorer's Hub**  
[forum.newrelic.com](https://forum.newrelic.com)

# Feedback

