

Seeing Failures Before They Reached the Data Center

How we built an early large-scale mobile APM platform at Ele.me to make client-side failures and performance visible - before a request ever reached the IDC.

OBSERVABILITY GAP

Traditional server monitoring started too late

DESIGN CONSTRAINT

Rich telemetry had to stay cheap on-device

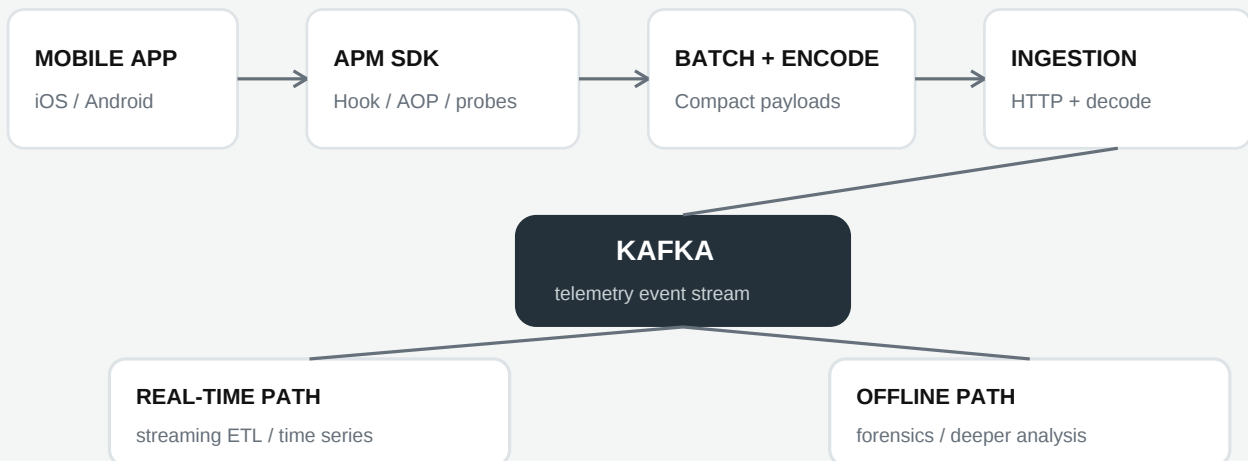
CONTEXT

Server-side monitoring could explain what happened after traffic reached the data center. It could not explain a crash, interaction failure, network problem, or performance degradation that happened on the phone before that boundary.

MY SCOPE

I led mobile architecture and infrastructure work that included APM, instrumentation and the client-side monitoring foundation. The goal was not only to count failures, but to capture enough context to understand what users actually experienced.

SIMPLIFIED TELEMETRY ARCHITECTURE



ARCHITECTURE LESSON

If failure can happen outside the server boundary, observability has to start outside it too.

The hard part was the trade-offs across the whole pipeline

The platform evolved over time. The architecture below focuses on the decisions that mattered: capture enough client context, move it efficiently, process it at scale, and preserve both real-time visibility and offline depth.

01 EDGE INSTRUMENTATION

Instrument iOS and Android with hooks / AOP-style probes to capture operations, failures and performance signals with minimal changes to product code.

TRADE-OFF

Coverage vs runtime overhead

02 ENCODING + TRANSPORT

Batch telemetry, use compact serialization such as Protobuf, and tune compression around a three-way trade-off: bandwidth, mobile CPU cost, and server-side decode throughput.

TRADE-OFF

Compression ratio vs compute cost

03 STREAM PROCESSING

Decode incoming batches into events, push them through Kafka, and move heavy aggregation / ETL into a streaming layer rather than forcing storage or the web edge to do the work.

TRADE-OFF

Edge simplicity vs processing flexibility

04 REAL-TIME + OFFLINE

Feed time-series / search-oriented stores and dashboards for immediate visibility while retaining an offline path for replay, forensic analysis and deeper reports.

TRADE-OFF

Immediate signals vs forensic depth

PUBLIC VALIDATION

Conference records from 2016 describe my responsibility for Ele.me mobile middleware and infrastructure, including APM. A 2017 InfoQ architecture talk later described Grand as an end-to-end mobile performance platform using client hooks, streaming ETL, Spark Streaming and time-series / search-oriented backends.

WHY THIS STILL MATTERS

Production readiness is not only about making the backend reliable. It is about defining the real failure boundary, instrumenting it, and building a feedback system that tells the team what is failing before customers have to.