

CASE STUDY



How PopClub turned
Datadog alerts into
incidents that reach the
right engineer within
seconds

PopClub's numbers using Xurrent IMR

< 1 min

Monitoring alerts to
incident, often seconds

10-15 min

current P0 acknowledgment

MTTD 10

Time to detect the affected
services/pod

**30-60 min
MTTR (P0)**

typical P0 recovery

**10-15 min
MTTR (P1-P2)**

recovery on a known revert

< 2 hrs (P0)

average recovery, harder
cases

What PopClub runs, and why good monitoring still left a gap

PopClub is a rewards UPI app backed by Razorpay. Mohan Kumar Pandian leads the engineering team, with Ankur and Shivam running the backend pods under him. They have run Datadog for observability for a while, and the monitors catch what they're supposed to.

Detection was solid. What broke down was what happened to an alert after it fired.

For a year, Datadog sent every alert into Slack as a message across three states: warning, critical, and recovery. The team had full visibility. What a Slack message cannot do is interrupt you. As Mohan put it, a message does not warrant the same attention as an incident. When a payments-down Po lands in the same channel as a routine warning, it reads like any other line until someone scrolls to it, and on a payments app, the minutes spent scrolling are minutes customers feel.

This is the coordination problem every fast-moving SRE team meets sooner or later. Monitoring generates plenty of signal, most of it noise.

Deciding which signals deserve a human, and getting each one to the person who can actually act, is a separate discipline that a monitoring tool was never built to run.

Xurrent IMR became the layer that sorts each alert by impact and sends it to the engineer who owns the fix

PopClub added Xurrent IMR on top of Datadog and used it to make three decisions that Datadog alone was not going to make for them.

✦ It decides which alerts interrupt a human.

Monitors are sorted P0, P1, and P2 by business impact. Everything still flows to Slack for visibility, so nothing is lost. Only a critical alert crosses the threshold into an alert, on a five-minute P0 window. The noise stays in Slack where it belongs, and the alert means something again.

“

Whenever an alert fires from Datadog, the webhook triggers and whoever is on call gets the alert. That is the basic rule we follow.

Mohan Kumar Pandian, PopClub

✦ It decides who gets the alerts.

PopClub's pods each own several services, and every service has an owner who holds the context. The team split on-call along those lines. Each owner now has their own rotation and their own escalation policy, so an alert for a payments service pages the person who built it.

The old pattern, where whoever held the pod pager could get an alert for a service they had never touched, acknowledge it, and then hand it to someone with context, is gone. This works because service ownership at PopClub is real and current. A team without that would need to fix ownership first, or fine-grained rotations become upkeep.

“

Now I can assign a service or cluster-level on-call, and that person gets the alerts and can solve it very fast. We tried it recently and it is working pretty fine for us

Mohan Kumar Pandian, PopClub

✦ It decides how hard to escalate.

A single escalation policy for every monitor is a decision made by not making one. PopClub weighs escalation by impact. Low-priority monitors escalate gently.

High-impact ones escalate hard, so if the first responder misses it, the next engineer is minutes away and a serious incident cannot sit unacknowledged while it spreads. Mohan stays out of the rotation and never carries a pager, but he sees every incident through Slack, so he gets oversight without the 3am noise.

The handoff from a Datadog alert to an incident now takes under a minute

Detection (MTTD) is the number PopClub watches most closely, and it is where Xurrent IMR earns its place. Datadog already burns a confirmation window before it fires, three to five minutes on some monitors. Anything the handoff adds lands on top of a clock that is already running, so the team guards it.

“

There are no lags. If Datadog triggers an alert, within a second, we are getting the alert from Xurrent IMR and that saves us a lot of time during the critical first minutes of an incident.

Mohan Kumar Pandian, PopClub

Detection and response lives in Xurrent IMR

PopClub keeps a clean line between the two tools. Xurrent IMR handles detection and routing. The incident details carry just enough signal to make a first call: the affected API groups or service, the latency or error-rate impact, and the service name. From there the engineer opens Datadog and traces it. For anything with a wide blast radius, the team spins up a Slack channel with a Meet link and pulls responders in by hand.

“

All the data needed for the judgment is what we try to pass in the alert. The developer then goes to Datadog, because that is where the entire implementation after root cause analysis happens.

Mohan Kumar Pandian, PopClub

Before Xurrent IMR	After Xurrent IMR
Datadog alerts arrived as Slack messages	Critical alerts interrupt as an alert within a second of Datadog firing
One shared channel, where a Po looked like a warning	Severity decides incident versus alert , on a five-minute Po window
Pod-wide on-call, so alerts could miss their owner	Service-level rotations alert the engineer who owns the fix
One escalation policy for everything	Escalation weighted by business impact
The lead tracked incidents from a console	The lead gets Slack visibility and carries no pager

What the team chose to keep after a year of real on-call with Xurrent IMR

These are the features that survived real on-call for PopUPI.

Configuration down to the service owner is the one Mohan names first. Who gets paged, which SLA applies, and how escalation runs are all set per service, and that granularity is what made the whole routing model possible.

“

I can configure it down to the service owner. The SLAs, the escalation policies, the on-call, all set per service.

Mohan Kumar Pandian, PopClub

Integrations that stay out of the way come next. Slack and Datadog plug in cleanly, and as a lead, Mohan does not need to live on the platform. The work happens on it and reaches him off it.

“

As a leader I do not need to be on the platform. Things work on and off the platform, and that fungibility is something I like.

And a small one that turned out to matter more than expected. The team set up deliberately funny page tones, so the sound itself tells the floor an incident is live before anyone reads a word.

“

We set up a lot of funny alert sounds. Now the sound itself tells the floor something is live, where earlier it was one flat tone for everything.

Mohan Kumar Pandian, PopClub

Where PopClub is taking this next

The backend team is heavily on the platform now, at 25 engineers. Front-end is next: PopClub is wiring its RUM (Real User Monitoring) signal so that a crash-rate spike or an unresponsive app maps to the right pod and raises a page, the same way backend does today.

When that lands, 40 to 45 developers will be on the platform, with another 10 to 15 covered through escalation, for roughly a 60-person presence.

The targets tighten with scale. PopClub is driving MTTD toward 10 minutes for P0 issues and recovery toward sub-13 minutes on known reverts, ahead of a sale coming in October.

For a consumer-facing payments app, every minute shaved off detection is a minute fewer customers spend unable to pay.

The takeaway for teams with good monitoring and clean ownership

PopClub did not add monitors or headcount. They changed where alerts go. A page instead of a message for anything critical.

The service owner instead of the pod for anything routed. Escalation sized to what an incident can actually cost the business. If your monitoring is already solid and your service ownership is clean, that is the place the fast wins are.

If this is something you're trying to solve, Talk to us!

[Book a demo](#)

[Read more customer stories](#)

