Oricato Intelligence · EARLY ACCESS

Method paper

The Oricato Intelligence method

Oricato Team · Version 1.0 · Updated Oct 16, 2026 · 14 min read

How Oricato Intelligence is designed to test send times and content, and how it will measure whether any of it changed how often people use an app. Written down before any result exists, so you can hold us to it.

Status: design, no results. This paper describes a system in development. It contains no measured effect of Oricato Intelligence. Figures are labelled as arithmetic, design or other teams' published results.

1. The question we answer

"Did Intelligence, as a whole, change how many days people use this app, compared with the same app running its normal program?" Everything else, from which time wins a slot to which text gets more opens, is a smaller question answered by experiments inside the program.

2. What Intelligence may change, and what it may not

Intelligence may change two things: the minute a message is sent inside a window the customer sets, and which approved version a person receives. It may not change whether a person gets a message, how many messages they get, the daily cap, or who is in the audience. A slot marked time-critical has no window. The fallback for any person without enough data is the slot time.

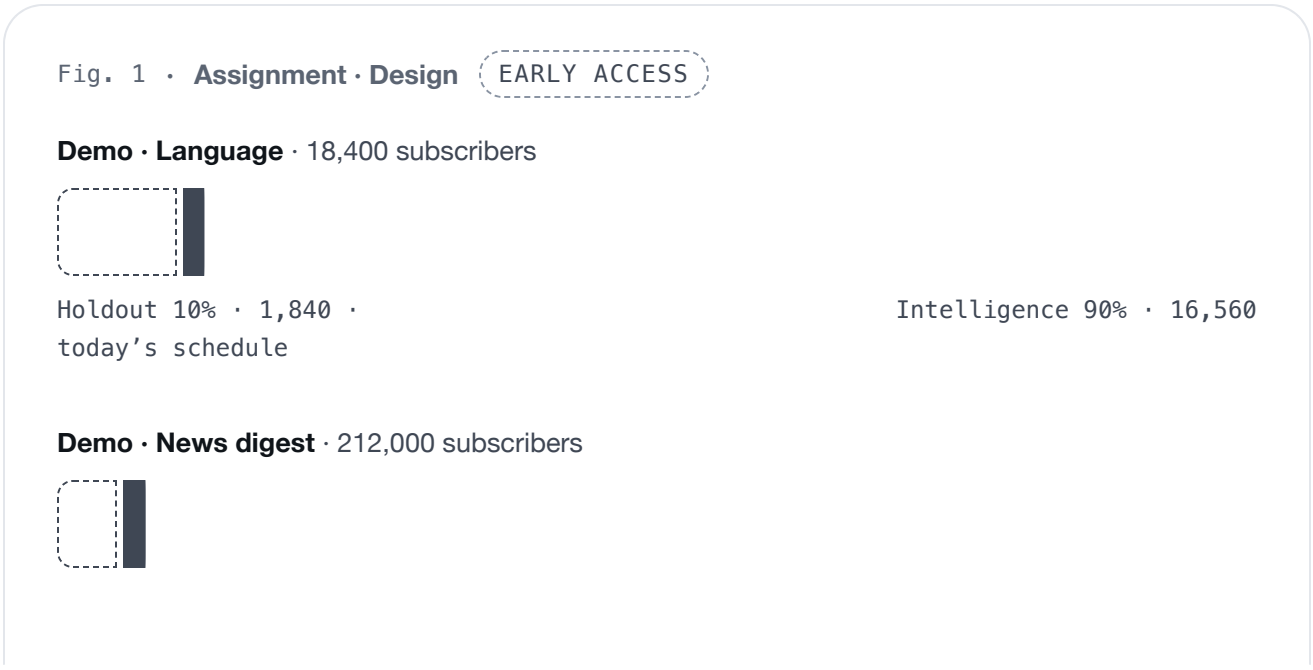
Fixing volume makes the comparison clean. A program that sends more can always show more opens; this one can't.

3. The holdout

Each subscriber is assigned once by a hash of their pseudonymous ID and a secret per app. The holdout is 10% of an app under 50,000 subscribers (20% optional) and 5% above. It is persistent and never reshuffled; a reset is an owner action that starts a 28-day wash-out and restarts every readout.

The holdout receives every message at the exact slot time, from a separate baseline rotation with its own no-repeat record. No plan entry, AI-written item or experiment can reach it. The estimand is therefore "Intelligence (timing, content, plan and AI content) versus the same program with Intelligence off." Calendar rules the customer sets reach every group, so their effect is outside this estimand and is measured with event-day experiments.

The analysis is intention-to-treat. Everyone with personalization consent who was subscribed at the window start, or joined during it, counts until they leave. Leaving is reported by group, as an outcome. People without consent are excluded from both groups. Test devices are excluded.



```
Holdout 5% · 10,600 ·                                  Intelligence 95% · 201,400
today’s schedule
```

4. The primary metric: active days per subscriber

An active day is a local day on which the app came to the foreground at least once, whether or not a notification started it. It is reported by the SDK's activity digest: for the last eight local days, 24 hourly counts of app opens not started by a notification and one bit per day for opens that were. No timestamps or content are sent.

We chose active days over opens because opens can rise without more use. Published randomized tests show the gap: per-person timing at Kuaishou raised clicks by 1.2% and daily active users by 0.1% (TIM, 2024); AI-written copy raised clicks on replaced messages by 14% and daily active users by 0.067% (PushGen, 2025).

A conversion the customer names may be a co-primary metric, chosen before the analysis starts. Opens are a diagnostic only. Guardrails are opt-outs and uninstalls per 1,000 people. If fewer than 20% of people report the digest, the app has no claimable primary metric.

5. Estimation

Lift is the relative difference in means, (treated – holdout) ÷ holdout, cumulative from the analysis start. Variance reduction with each person's pre-period activity (CUPED) is used only when the correlation measured on that app's own pre-period is at least 0.3; reported real-world reductions vary widely, so we don't assume one.

The interval is a delta-method interval for the ratio, judged against group-sequential boundaries, with a resampling estimate as a check. Each look uses

people with a full exposure window, or reports by entry week. Lift is also reported by week since the start, so novelty can be seen fading.

Two health checks gate every readout: the observed holdout share must match the planned share (a sample-ratio check), and the same estimator applied to the period before the start must show no difference (a placebo check).

```
lift = (Ȳ_treated − Ȳ_holdout) / Ȳ_holdout
Y′ = Y − θ (X − X̄),  θ = Cov(Y, X) / Var(X), used only if Corr(Y, X) ≥
```

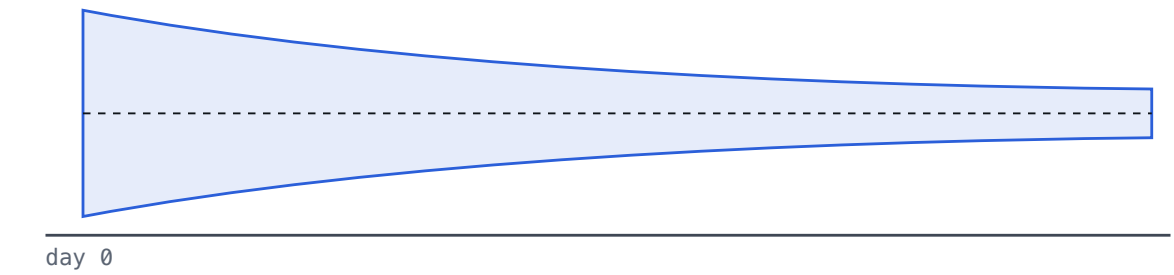
6. Looks and the claim rule

Readouts are planned every 28 days, at most six, with O'Brien–Fleming boundaries (Lan–DeMets spending) at an overall two-sided 5% level. With a known look schedule, this design is more powerful than an always-valid interval; always-valid tests are kept for guardrails that are watched continuously.

A lift may be stated only when all hold: the boundary is crossed at a scheduled look 56 days or more after the start; the most recent non-overlapping 28-day window has the same sign; the sample-ratio and placebo checks are clean; and the statement names the metric, the number of people, the period, the interval and the app's category. Statements that combine apps, or appear on our website, include every app that counts toward public statements, with a false-discovery correction or a pooled estimate of all of them.

Otherwise the readout says "no measurable difference yet," with its interval.

Fig. 2 · **Planned looks · Design** EARLY ACCESS



Day 28 · interim view

Days 56 to 168 · a result can be stated here if the bar is met

The boundary, less strict at later looks

Arithmetic

7. What an app can detect

The minimum detectable effect falls with the square root of the group sizes. The table below is illustrative (28 days, 80% power, a single look, an example metric with mean 4.2 and standard deviation 6.0 per subscriber). The setup screen computes it for each app's real metric before anything is switched on.

5,000

HOLDOUT

20%

SMALLEST DETECTABLE CHANGE

about 14%

18,000

HOLDOUT

5% · 10% · 20%

SMALLEST DETECTABLE CHANGE

about 13.7% · 10% · 7.5%

50,000

HOLDOUT

5% · 10%

SMALLEST DETECTABLE CHANGE

about 8% · 6%

100,000

HOLDOUT

5%

SMALLEST DETECTABLE CHANGE

about 5.8%

1,000,000

HOLDOUT

5%

SMALLEST DETECTABLE CHANGE

about 1.8%

3,000,000

HOLDOUT

5%

SMALLEST DETECTABLE CHANGE

about 1.1%

ARITHMETIC · NOT A MEASUREMENT

> Published randomized effects on daily activity run from 0.07% to 0.7% (Duolingo +0.5% daily active users; Kuaishou TIM +0.1%; PushGen +0.067%; LinkedIn +0.72% sessions). They are an order of magnitude below what one small app can detect. For a single app of 5,000 to 50,000 subscribers, "no measurable difference" is the expected honest result.

8. The size rule: standard mode and large mode

Every app starts in standard mode. Timing learns one offset per slot through fixed-split send-time experiments, for example −60, 0 and +60 minutes, with each person in the same arm for 4 to 8 weeks. The app's own activity curve sets the starting point. A slot's time changes only when a pre-registered experiment

crosses its boundary and a person approves. Seasons with different routines are tested separately.

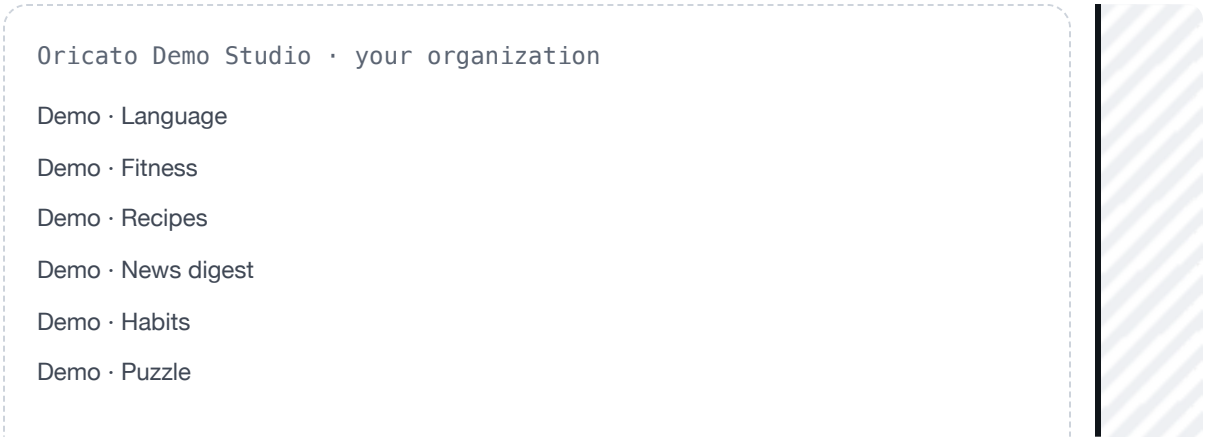
Large mode applies only to apps with at least 50,000 subscribed people, and only after a policy check. A small, logged share of sends is placed uniformly inside the window. From those sends, the value of a group-time policy and a personal-time policy is estimated by inverse-propensity weighting and compared by a paired bootstrap. Personal times switch on when the lower bound of their advantage over the group time is at least zero, and the check repeats every four weeks.

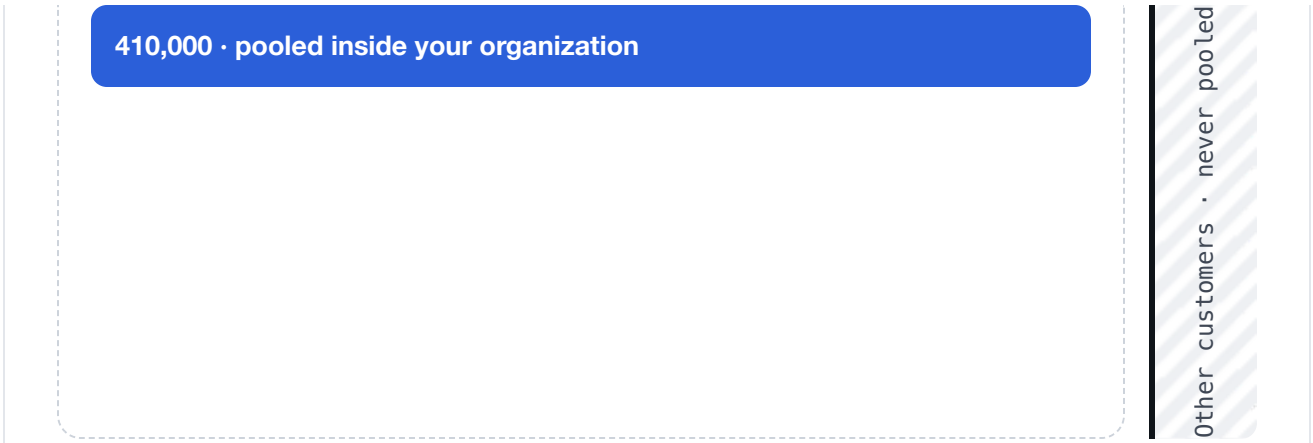
The reason for the rule: at 5,000 subscribers with 10% random exploration, separating two send offsets that differ by 10% would take on the order of 78 days per offset. Most of a personal model's machinery cannot act at that size.

9. Pooling within a studio

One instance serves one organization. Within it, small groups borrow strength from their app, then from the organization's other apps. For statements, a random-effects estimate across all of an organization's apps that count toward public statements is used, never a chosen subset. Nothing is pooled across customers.

Fig. 3 · **Pooling · Design** EARLY ACCESS





10. Experiments inside the program

Experiments run only among treated people, never in the holdout. Types form a closed list (copy angle, title, emoji, image, deep link, translation style, slot category, send offset, weekday rule) and exclude any change to volume. Lesson experiments use a fixed split with person-level assignment and group-sequential inference at pre-registered looks. Sample sizes are stated in people, with a design effect for repeated sends to the same person. Completed lessons are corrected for multiple testing every quarter (Benjamini–Hochberg, q = 0.10) and must replicate once before they inform the writer. Adaptive allocation results are reported by allocation period and labelled "optimization result, not evidence."

11. What the AI may see

Language models are optional and do language work only. Prompts may contain message texts, library items, category names and briefs, brand documents, a glossary, calendar events and aggregates over at least 20 people. They never contain subscriber IDs, tokens, per-person country, time zone or device, IP addresses or admin details. Imported messages sent to fewer than 100 recipients never enter a prompt. Numbers in the weekly AI summary are checked

against the source data by code, and causal wording is allowed only for randomized results.

The privacy page →

12. Limits and open questions

Coverage.

Active days depend on SDK version and consent. Both are balanced across groups by the hash, but low coverage means no claimable result.

Multi-device people.

By default the unit is a subscription. Where an app verifies user IDs, the verified ID keeps a person in one group across devices; readouts report the multi-device share.

Calendar effects.

Event content the customer sets reaches everyone; its effect needs event-day experiments, which have one occurrence per year per country.

Interference.

People who share content may influence each other across groups. We don't model this.

Validation.

Before real use, the code runs against simulated audiences, including worlds where timing has no effect, to check that it reports nothing when there is nothing.

References

1. Okoshi, Tsubouchi, Tokuda. Real-world product deployment of adaptive push notification scheduling on smartphones. KDD 2019. LY Corporation research

2. Yancey, Settles. A sleeping, recovering bandit algorithm for optimizing recurring notifications. KDD 2020. PDF
3. Ji et al. TIM: Temporal Interaction Model in notification system. 2024. arXiv 2406.07067
4. Bie et al. PushGen: push notifications generation with LLM. WSDM 2026. arXiv 2512.14490
5. Ocejo et al. Generative sequential notification optimization via multi-objective decision transformers. 2025. arXiv 2509.02458
6. Zeng et al. Large language models for social networks. 2024. arXiv 2401.02575
7. Kohavi, Tang, Xu. Trustworthy Online Controlled Experiments. Cambridge University Press, 2020.
8. Jennison, Turnbull. Group Sequential Methods with Applications to Clinical Trials. 2000.

Figures from these works are their authors' results, not Oricato's.

Change log

Version	Date	Change
Version 1.0	Oct 16, 2026	First publication. Corresponds to the Intelligence design, revision 3.

Every change to the method is recorded here before it applies to any readout.