Personalization Data Pipeline Readiness Checklist — One Page

Fora Soft Learn

Recommendations, search, and merchandising are only as good as the pipeline feeding them. Get four things right: collect the events from every screen, choose real-time versus batch deliberately, run a feature store that serves the same numbers in training and production, and enforce a privacy boundary on viewing data. The viewing-data law (VPPA) is fast-moving - this is engineering guidance, not legal advice; confirm specifics with counsel.

1 · COLLECT THE EVENTS (the raw material)

- ☐ **Three event families** - playback (play, pause, seek, complete), navigation/clickstream (impression, click, scroll), and search.
- ☐ **Keep the heartbeat** - the player's 'still watching, at this position' ping every few seconds is how watch time is reconstructed.
- ☐ **Log impressions, not just clicks** - a model must learn from what was shown and ignored, not only from what was played.
- ☐ **Size for the peak** - 1M concurrent at ~10 events/min is ~167k events/sec; design for the spike, then make steady state cheap.

2 · REAL-TIME VS BATCH (the two clocks)

- ☐ **Stream when freshness changes the answer** - last hour watched, trending now, continue-watching; computed in seconds (e.g. Flink).
- ☐ **Batch for slow, large signals** - 90-day taste, lifetime hours, embeddings; recomputed on a schedule (e.g. Spark), far cheaper.
- ☐ **Don't reach for real-time everywhere** - streaming infra costs more and is harder to run; most signals don't need seconds-fresh.
- ☐ **Lambda (two layers) vs Kappa (one)** - prefer one code path over a replayable log unless you truly need a separate batch layer.

3 · FEATURE STORE & SKEW (correctness)

- ☐ **One feature definition, served to both** - training and production read the same logic; this is what kills training-serving skew.
- ☐ **Offline + online store** - offline for training over full history, online for low-latency serving (~10 ms) inside a live request.
- ☐ **Log served features, train on them** - if you train on exactly what you served, the two code paths cannot drift apart.
- ☐ **Point-in-time joins** - a training row must see only data that existed before its timestamp; otherwise leakage fakes the accuracy.

4 · THE PRIVACY BOUNDARY (viewing data)

- ☐ **Never leak who-watched-what to third-party tags** - the VPPA exposure is $2,500 per person (18 U.S.C. § 2710); the #1 mistake.
- ☐ **Specific, separate consent** - VPPA consent is not buried in the terms of service; it can be advance, with a clear way to withdraw.
- ☐ **Retention limit** - destroy viewing PII within a year of when it is no longer needed (VPPA § 2710(e); GDPR storage limitation).
- ☐ **Pseudonymize and minimize** - work on an opaque key, collect only what you use, carry consent state with the data (GDPR, CCPA).

THE ORDER OF OPERATIONS — COLLECT, MOVE, STORE, PROTECT — AND THE PIPELINE IS THE PRODUCT

Everything in discovery - recommendations, search, merchandising, and the experiments that tune them - runs on the personalization data pipeline, so the pipeline, not the model, is usually where a streaming platform wins or loses. Build it in order. First, collect events from every screen: playback events (the heartbeat reconstructs watch time), navigation/clickstream (impressions matter as much as clicks), and search. Second, move them through a replayable log and choose the clock per feature - stream processing in seconds where freshness changes the answer, scheduled batch where the signal is slow and large; prefer one code path (Kappa over a replayable log) to maintaining the same logic twice (Lambda). Third, store features in a feature store that serves one shared definition to both training and production - this is what eliminates training-serving skew, the silent bug where a feature computed one way in training and another at serving quietly wrecks predictions - and assemble training data with point-in-time joins so a row never sees data from after the moment being predicted. Fourth, enforce the privacy boundary: viewing data is legally sensitive, so never disclose who watched what to a third-party advertising or analytics tag (the pattern behind the wave of Video Privacy Protection Act class actions, at $2,500 per person), gate it on specific consent, minimize and pseudonymize it, and delete it on a retention clock. The convenient shortcut - a quick third-party pixel, or a second feature query that is faster to write than to share - is the expensive bug or the expensive lawsuit. Treat the pipeline as the product. This is engineering guidance, not legal advice - confirm specifics with counsel.