WHIS logo

Mock WHIS Implementation (Start to Finish)

1. Planning and Kickoff

The first step is defining the business goals. In this mock project, the company wants to connect its online store to WHIS so that all orders flow into their ERP for invoicing, their CRM for customer tracking, the warehouse for fulfilment, and their data warehouse for reporting.

During planning, both teams agree on the following:

- Which systems need to be connected.
- What the “source of truth” data format will look like.
- Security and access requirements.
- A timeline and roles for each stage of the project.

2. Access and Security Setup

Next, WHIS creates separate environments (development, staging, production). The client’s IT team configures single sign-on, user roles, and access permissions. Service accounts are created for system-to-system connections, and secrets (like API keys) are stored securely.

At this stage, IP allow-lists, encryption rules, and compliance checks (GDPR, CCPA) are also confirmed.

3. Data Mapping

Before data starts flowing, both sides agree on a canonical schema — a standardised format for orders. For example, the schema might require fields such as order ID, creation time, customer details, amount, items, and shipping address.

Each field from the company’s store is mapped into this schema. For instance, “buyer.email” from the source becomes “customer.email” in WHIS. Transformations like converting cents to dollars, standardising currency codes, and cleaning up name fields are also agreed.

4. Ingestion Setup

There are two ways data enters WHIS in this mock project:

- Real-time batches via API: the store sends orders directly into WHIS.
- Bulk CSV uploads: for large backfills, the operations team can drop spreadsheets into the Admin Console.

Idempotency is introduced at this stage — this ensures that if a batch is accidentally sent twice, WHIS will recognise it and not duplicate the data.

5. Validation and Error Handling

As orders arrive, WHIS applies validation rules. These rules check things like:

- Required fields (order ID, customer, items) are present.
- Amounts match the sum of items.
- Currency is in an allowed set (AUD, USD, EUR).
- Email addresses and postcodes are in the correct format.

If records fail validation, they are flagged and moved into an error queue. Staff can review and fix them in the Admin Console, download a rejection report, and re-upload corrected data.

6. Transformation, Enrichment, and Routing

Valid orders are then enriched (for example, tagging VIP customers or calculating tax) and routed to downstream systems. Rules might say:

- Send all paid or fulfilled orders to the ERP.
- Always update the CRM with customer details.

- Forward pick tickets to the warehouse system.
- Copy all orders into the analytics data warehouse.

Routing is reliable and includes retries if a destination system is temporarily unavailable.

7. Webhooks and Notifications

Whenever a job is completed, WHIS sends a webhook notification back to the client's systems. This message confirms whether the batch succeeded, how many records were inserted, and whether any were rejected.

These notifications allow the client's IT team to update dashboards, trigger follow-up workflows, or alert staff in case of problems.

8. Monitoring and Observability

WHIS provides dashboards that show:

- Records processed per minute.
- Success vs. rejection rates.
- Processing times (latency).
- Errors or retry counts.

Alerts are also set up so the team is notified if too many records fail, if retries build up, or if throughput drops below expectations.

9. Testing and Pilot

Before going live, the integration is thoroughly tested:

- Unit tests confirm that schema validation works.
- Integration tests push both valid and invalid data through staging.
- A user acceptance test simulates real business cases, such as cancelled orders or refunds.

After testing, a pilot run is performed where real data flows through staging, but with routes pointing to shadow systems so production systems aren't impacted.

10. Go-Live

On launch day, a runbook is followed. This includes:

- Freezing non-essential changes.
- Rotating production API keys.
- Switching the source store to send data to WHIS production.
- Running a “smoke test” with a handful of orders.
- Watching dashboards and alerts closely for the first few hours.

If problems occur, a rollback plan is ready: the store can be pointed back to its previous pipeline, and WHIS can pause routing until issues are resolved.

11. Post-Launch Optimisation

Once the system is stable, tuning begins:

- Adjusting batch sizes and retry intervals.
- Adding new fields to the schema as business needs change.
- Expanding integrations to include other systems (for example, fraud detection or returns processing).

Regular reviews are held to check performance, costs, and whether the pricing tier still fits.

12. Training and Operations

Operations staff are trained on how to:

- Use the Admin Console to monitor jobs.
- Handle rejected records.
- Run backfills with historical data.
- Interpret dashboards and respond to alerts.

Runbooks are provided for common scenarios like resubmitting rejects, performing bulk backfills, or purging test data.

13. Service Levels

The company runs under WHIS's service targets:

- 99.9% uptime.
- Fast processing times for ingested batches.
- Real-time notifications for errors.

Depending on the support tier, the business has access to business-hours help or 24/7 escalation lines.

End Result

At the end of this mock implementation, the company has:

- Automated the flow of orders from their store to ERP, CRM, warehouse, and analytics.
- Eliminated manual reconciliation of files.
- Gained real-time visibility into data health and performance.
- Reduced operational risk with strong validation, monitoring, and rollback procedures.