

SDMT cost UI surfaces and data-lineage impact

Finanzas SD – Architecture, Flows & SOPs

Arquitectura, Flujos y Procedimientos

1 SDMT cost UI surfaces and data-lineage impact

This note answers whether recent SDMT cost UI files impact data lineage/streaming of cost data.

- `src/features/sdmt/cost/Catalog/SDMTCatalog.tsx`: Initiates writes by calling the Finanzas API (e.g., `addProjectRubro`) and reads project line items via `useProjectLineItems`. It is the only listed file that mutates cost data by posting catalog changes to the backend.
- `src/features/sdmt/cost/Changes/SDMTChanges.tsx` and `src/features/sdmt/cost/Recon`: Read-only for cost data. They consume `useProjectLineItems` to populate selectors; they do not push cost amounts upstream.
- `src/features/sdmt/cost/Forecast/SDMTForecast.tsx` and `src/features/sdmt/cost/Cashflow`: Read-only visualizations. They request forecast/cashflow data from the backend and render charts/tables without mutating cost records.
- `src/features/sdmt/cost/Reconciliation/SDMTRecon.tsx` (Spark-generated screen): Pure UI shell; it does not participate in cost persistence.

Implication: For cost data lineage/stream integrity, Catalog actions are the only UI surface in this list that writes cost information; the others read from backend endpoints and do not alter cost data. Backend handlers and Dynamo tables remain the sources of truth for cost persistence and aggregation.