🍕 Hunt Advantage Group – Pizza & Wings Sales Analytics

Engineering Architecture & Implementation Document Author: Jeremy Bishop **Date**: September 2025

📌 Project Overview

This project simulates a real-world analytics pipeline for retail food sales, focused on pizza and wings performance across multiple store locations. It is designed to demonstrate full-stack data engineering capabilities using a medallion architecture (Bronze → Silver → Gold), Power BI reporting, and strategic orchestration under licensing and platform constraints.

🧱 Architecture Summary

🥉 Bronze Layer – Raw Ingestion

- **Source**: Local CSV files from simulated POS, inventory, feedback, and store metadata
- **Location**:

  Code

  ```
  C:\source\repos\HBPizza\data_ingestion\csv_data\bronze
  ```

- **Files**:
  - `orders.csv`
  - `feedback.csv`
  - `inventory.csv`
  - `locations.csv`
  - `store_metrics.csv`

🥈 Silver Layer – Cleaned & Integrated

- **Transformations**:
  - Normalized column names and types
  - Joined store metadata with orders
  - Calculated fields: `pizza_combo`, sentiment polarity
- **Tools**: Python (Pandas), Power Query
- **Output Location**:

Code

```
C:\source\repos\HBPizza\data_ingestion\csv_data\silver
```

🥇 Gold Layer – Modeled & Aggregated

- **Star Schema**:
  - `dim_store`, `dim_product`, `dim_date`
  - `fact_orders`, `fact_feedback`
- **Aggregations**:
  - Total sales, average order value, wings mix, sentiment score
- **Tools**: Python, DuckDB, Power BI DAX
- **Output Location**:

  Code

  ```
  C:\source\repos\HBPizza\data_ingestion\csv_data\gold
  ```

📊 Reporting & Version Control

Power BI Desktop

- **Report**: `pizza_reporting.pbix`
- **Data Source**: Gold-layer CSVs
- **Measures**:
  - `Total Sales`
  - `Orders`
  - `Average Order Value`
  - `Wings Mix`
  - `Avg Sentiment`

Version Control Strategy

- **Template Format**: `.pbit` (Power BI Template)
- **Git Repo Structure**:

  Code

  ```
  /powerbi_templates/
    ├── pizza_reporting_template_v1.pbit
    ├── pizza_reporting_template_v2.pbit
  ```

```
└── CHANGELOG.md
```

- **Benefits**:
  - Lightweight versioning
  - Separation of logic from data
  - Reusability across environments

🔓 Platform Restrictions & Workarounds

Restriction	Impact	Workaround
❌ Fabric trial provisioning failed	No access to Lakehouse, Notebooks, Pipelines	Pivoted to local medallion architecture using CSVs and Python
❌ No OneDrive for Business	Blocked CSV uploads in Power BI service	Used Power BI Desktop with local files and published `.pbix` manually
❌ Dataflow Gen1 limited to Web connectors	No access to Folder or Gateway sources initially	Published `.pbix` to activate workspace, then configured gateway
❌ No Power BI Pro license	Blocked direct publishing from Desktop	Used Power BI Service “Upload” feature as workaround
❌ Large `.pbix` files not Git-friendly	Difficult to version control binary files	Saved as `.pbit` for lightweight tracking and reuse

🧠 Strategic Highlights

- Simulated Fabric-style orchestration using open-source tools
- Demonstrated medallion architecture manually for transparency and control
- Used `.pbit` and changelog discipline to showcase versioning without premium features
- Configured on-premises gateway to bridge local data with Power BI service
- Documented all constraints and workarounds to highlight adaptability

📁 Next Steps

- Finalize Gold-layer aggregations and DAX measures
- Create architecture diagram and README for GitHub repo
- Build interview-ready demo script highlighting business impact
- Explore optional deployment to Databricks or DuckDB for cloud simulation