Data Model and Storage Solution Using Data Vault 2.0 Methodology

**1. Technology Stack** To handle the scale, complexity, and real-time requirements of the proposed architecture, I recommend using the following technologies:

**Data Storage:**

Azure Data Lake Storage (ADLS) for storing raw and processed data.

Synapse Analytics (or Azure SQL Database) for structured data storage.

**Data Processing:**

Azure Synapse Pipelines and Azure Databricks for ETL/ELT processes.

**Data Ingestion and Real-Time Processing:**

Azure Event Hubs or Kafka for event streaming, followed by Stream Analytics

or Azure Databricks for real-time data transformation.

**Data Modeling:**

Implement Data Vault 2.0 using Synapse SQL or Databricks Delta Lake tables.

**2. Data Vault 2.0 Table Structure**

Data Vault 2.0 methodology splits data into three primary components:

**Hubs, Links and Satellites.**

***a. Hubs***

Hubs are the core business entities identified by unique keys. Each Hub contains:

- *Primary Key* (Business Key): Unique identifier for the entity ( *\*PII data will be hashed)*

- *Load Date*: Timestamp when the data was loaded.

- *Record Source*: Source of the data.

We create Hubs for Player, Game, and Event

**Hub\_Player**

Player\_ID: INT --- The unique identifier for the player (mapped from uid in events).

Load\_Date: DATETIME --- The timestamp when the data was loaded into the Hub.

Record\_Source: VARCHAR --- The source of the data, Ex:"auth\_event", "spin\_event".

**Hub\_Game**

Game\_ID: VARCHAR --- The unique identifier for the game/application (ex: "app\_3").

Load\_Date: DATETIME --- The timestamp when the data was loaded into the Hub.

Record\_Source: VARCHAR --- The source of the data.

**Hub\_Event**

Event\_ID: INT --- The unique identifier for the event (msg\_id from each event).

Event\_Type: VARCHAR --- The type of event, Ex:"auth\_event", "spin\_event", "purchase\_event".

Load\_Date: DATETIME --- The timestamp when the event was loaded into the Hub.

Record\_Source: VARCHAR --- The source of the data.

***b. Links***

Links establish relationships between hubs. We define Links to establish connections between players, games, and events. Each link table contains:

*- Primary Key*: Composite key made up of the foreign keys from the linked hubs.

*- Load Date*

*- Record Source*

**Link\_Player\_Game**

Player\_ID: INT --- Foreign key to Hub\_Player.

Game\_ID: VARCHAR--- Foreign key to Hub\_Game.

Event\_ID: INT --- Foreign key to Hub\_Event (for linking specific events to this relationship).

Load\_Date: DATETIME --- The timestamp when the link was created.

Record\_Source: VARCHAR--- The source of the data.

**Link\_Player\_Event**

Player\_ID: INT --- Foreign key to Hub\_Player.

Event\_ID: INT --- Foreign key to Hub\_Event.

Load\_Date: DATETIME --- The timestamp when the link was created.

Record\_Source: VARCHAR--- The source of the data.

**Link\_Game\_Event**

Game\_ID: VARCHAR--- Foreign key to Hub\_Game.

Event\_ID: INT --- Foreign key to Hub\_Event.

Load\_Date: DATETIME --- The timestamp when the link was created.

Record\_Source: VARCHAR--- The source of the data.

***c. Satellites***

Satellites store descriptive information related to hubs or links and they may include SCD type 2 (attributes that may change over time)

Examples:

**Satellite\_Player\_Profile**

Player\_ID: INT --- Foreign key to Hub\_Player.

Email: VARCHAR--- The player’s email address (PII, encrypted or hashed).

Phone: VARCHAR--- The player’s phone number (PII, encrypted or hashed, or null if unavailable).

Load\_Date: DATETIME --- The timestamp when the data was loaded.

Record\_Source: VARCHAR--- The source of the data (Ex:"auth\_event").

**Satellite\_Event\_Details**

Event\_ID: INT --- Foreign key to Hub\_Event.

Publish\_Timestamp: DATETIME --- The timestamp when the event was published (publish\_ts from the event).

Event\_Type: VARCHAR--- The type of event, Ex:"auth\_event", "spin\_event", "purchase\_event".

Load\_Date: DATETIME --- The timestamp when the data was loaded.

Record\_Source: VARCHAR--- The source of the data.

**Satellite\_Auth\_Event**

Event\_ID: INT --- Foreign key to Hub\_Event.

UID: INT --- User identifier for linking purposes (uid from auth\_msg).

App: VARCHAR--- The app associated with the event (Ex:"app\_3").

Email: VARCHAR--- The email address, encrypted or hashed.

Phone: VARCHAR--- The phone number, encrypted or hashed, or null if unavailable.

Load\_Date: DATETIME --- The timestamp when the data was loaded.

Record\_Source: VARCHAR--- The source of the data.

**Satellite\_Spin\_Event**

Event\_ID: INT --- Foreign key to Hub\_Event.

UID: INT --- User identifier (uid from spins\_msg).

Spin\_Amount: INT --- The amount or number of spins (spin from spins\_msg).

App: VARCHAR--- The app where the spin occurred (Ex:"app\_3").

Load\_Date: DATETIME --- The timestamp when the data was loaded.

Record\_Source: VARCHAR--- The source of the data.

**Satellite\_Purchase\_Event**

Event\_ID: INT --- Foreign key to Hub\_Event.

UID: INT --- User identifier (uid from purchase\_msg).

Amount: DECIMAL(10, 2) --- The purchase amount (amount from purchase\_msg).

App: VARCHAR--- The app where the purchase occurred (Ex:"app\_3").

Load\_Date: DATETIME --- The timestamp when the data was loaded.

Record\_Source: VARCHAR--- The source of the data

**Event Mapping to Data Vault Model**

****

**auth\_msg:**

msg\_id: 124 → Event\_ID in Hub\_Event.

publish\_ts: "2024-10-12T14:00:00" → Publish\_Timestamp in Satellite\_Event\_Details.

type: "auth\_event" → Event\_Type in Satellite\_Event\_Details.

uid: 453135 → Player\_ID in Hub\_Player.

email: "SomeEmail@test.com" → Email in Satellite\_Player\_Profile.

phone: null → Phone in Satellite\_Player\_Profile.

app: "app\_3" → App in various tables.

**spins\_msg:**

msg\_id: 1275 → Event\_ID in Hub\_Event.

publish\_ts: "2024-10-12T14:02:00" → Publish\_Timestamp in Satellite\_Event\_Details.

type: "spin\_event" → Event\_Type in Satellite\_Event\_Details.

uid: 125331 → Player\_ID in Hub\_Player.

spin: 1400 → Spin\_Amount in Satellite\_Spin\_Event.

app: "app\_3" → App in various tables.

**purchase\_msg:**

msg\_id: 2112 → Event\_ID in Hub\_Event.

publish\_ts: "2024-10-12T17:09:00" → Publish\_Timestamp in Satellite\_Event\_Details.

type: "purchase\_event" → Event\_Type in Satellite\_Event\_Details.

uid: 124442 → Player\_ID in Hub\_Player.

amount: 1499 → Amount in Satellite\_Purchase\_Event.

app: "app\_3" → App in various tables.

**3. Additional Components**

Data Ingestion Framework:

Implement a framework using Azure Event Hubs or Kafka for ingesting JSON events into the data lake.

Data Processing and Transformation:

Azure Synapse Pipelines or Azure Dtabricks to handle ETL/ELT, batch processing, and stream processing.

Data Governance and Security:

Use Azure Purview for data cataloging and governance. Implement encryption for PII data using Azure Key Vault.

Reporting and Analytics: Use Power BI to build dashboards and reports.

Data Monitoring & Alerts: Set up monitoring and alerting using Azure Monitor or Application Insights for real-time health checks and performance tracking.

**4. Storage Format**

Raw Data Layer: Store raw JSON events as they arrive in Azure Data Lake Storage

Structured Data Layer:

Use Delta Lake or Parquet format for processed data in Azure Synapse for optimized querying.

PII Management: Store encrypted or hashed PII data and manage encryption keys using Azure Key Vault.