

VELUX Take-Home Case: Cloud/Data Engineer

by: Iurii Ivanov

04/06/2025

Agenda

- **The VELUX Challenge:** Project Goals
- **Infrastructure:** Terraform
- **Architectural Approach:** Data Mesh
- **Data Sources & Technical Stack**
- **Data Modelling with dbt**
- **Governance and Lineage**
- **What was difficult**
- **Future Vision**

The VELUX Challenge: Project Goals

1

Design a data model, transform with dbt according to the design.

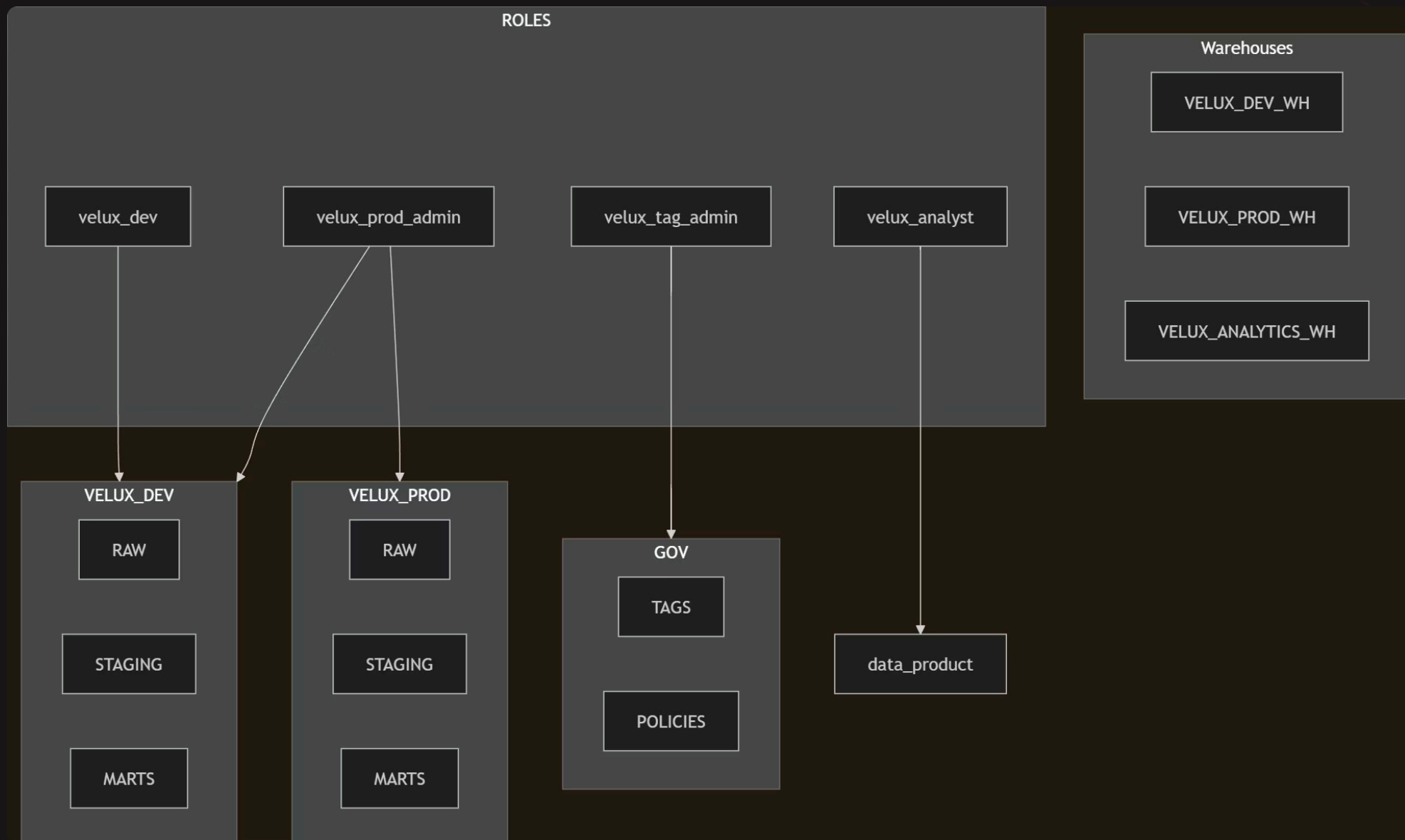
2

Build a scalable and robust architecture for large-scale data processing.

3

Organize the project with models, tests and documentation.

Infrastructure (Terraform)



- RBAC
- Warehouse Policies
- Resource Monitors
- Database and Schemas

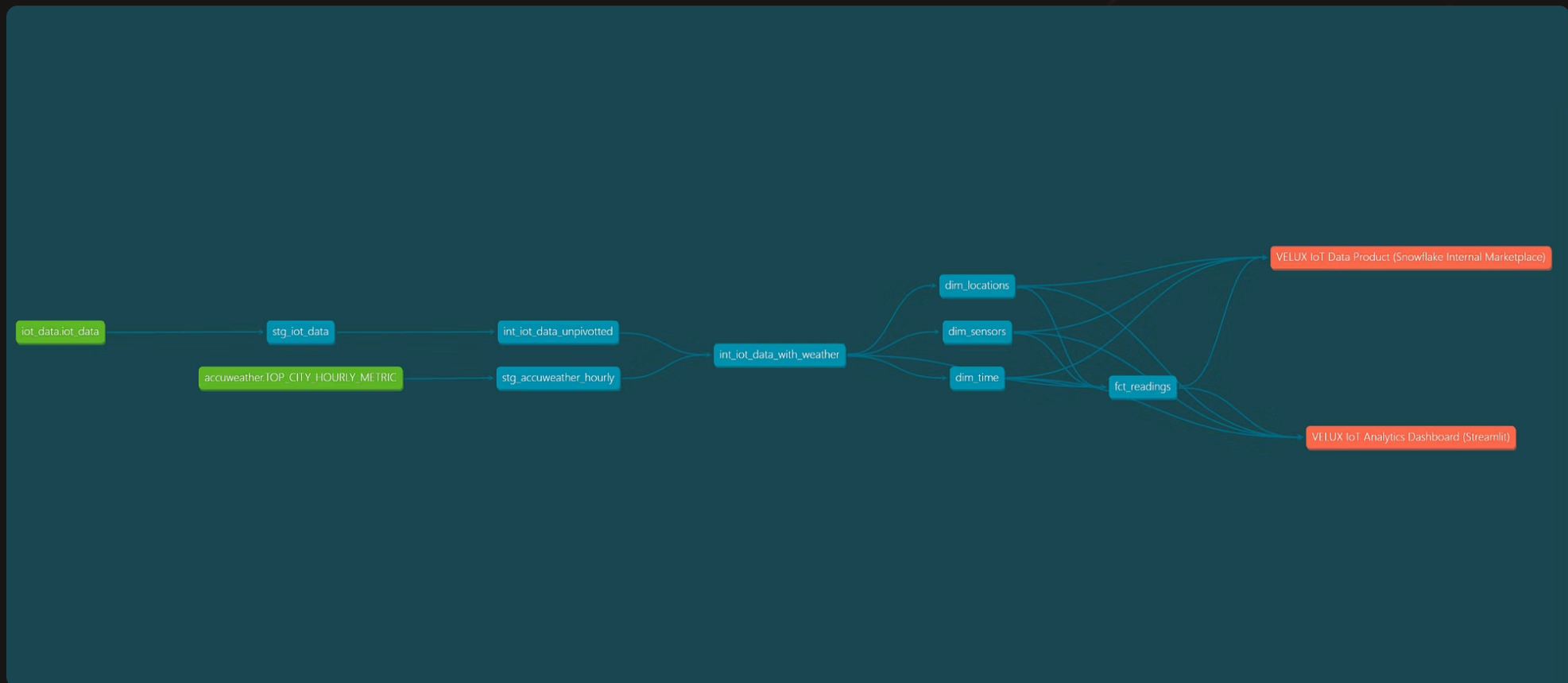
Architectural Approach: Data Mesh

- Why Data Mesh?
- Domain: Product R&D
- Self-Serve Data Platform (Snowflake + dbt Cloud)
- Federated Computational Governance (central policies + domain-level governance)
- Data as a Product

Data Sources & Technical Stack

- **Data Sources:**
 - **Accuweather:** Snowflake Marketplace data.
 - **IoT Sensor Data:** Manually uploaded CSV.
- **Ingestion:**
 - **Accuweather:** Through "IMPORTED PRIVILEGES".
 - **IoT Sensor Data:** Manual upload into the RAW stage.
- **Platform:** Snowflake (Trial Account).
- **Transformations:** dbt Core and Cloud (Trial Account, Snowflake Partner Connect).
- **Infrastructure as Code:** Terraform for managing Snowflake resources.
- **Version Control:** GitHub.

Data Modelling with dbt



- Layers
- Materialization Strategy
- Macros
- Tests
- CI/CD
- Idempotency
- Documentation

Governance and Lineage

Federated Computational Governance

- **Central Platform Governance:**
 - **RBAC and Terraform:**
 - Clearly defined roles
 - Managed as code
 - **Cost Control & Predictability:**
 - Resource Monitors
 - **Security:**
 - Network Policies
- **Domain-Level Governance:**
 - **Data Classification, Tagging & Automated Policy Enforcement**
 - Tags
 - Masking Policies
 - Compliance
 - **Proactive Data Quality Assurance:**
 - Comprehensive tests
 - All tests are integrated into CI/CD pipeline
 - **Schema Contracts & Stability:**
 - Models have defined schema contracts (columns, data types).
 - **Code & Project Standards:**
 - SQL linting with SQLFluff
 - Standardized dbt project structure
 - **Transparent Documentation:**
 - dbt docs
 - dbt catalog

Data Lineage :

- **Automated by dbt:**
 - dbt inherently understands and tracks dependencies between all sources, models, seeds, snapshots, and exposures.
 - dbt cloud provides catalog and cross-project lineage
 - The lineage is automatically generated and visualized
- **Critical Benefits:**
 - Impact Analysis
 - Root Cause Analysis
 - Enhanced Trust & Discoverability
 - Auditability

What was difficult

- Managing Grants
- Transitioning from dbt Core to dbt Cloud

Future Ideas

- Automated Policy Enforcement via dbt posthooks
- Model Versioning Strategy
- Anomaly Detection & Alerting
- SLA/Freshness/DQ Dashboard
- SCD 2/6 logic
- Formalized Semantic Layer
- Automated Marketplace Listing Management via dbt posthooks