

Leveraging Data Warehousing for Smart Farming: A BI Architecture Designed Using the Kimball Method and the IRADAH Framework

Frédéric S. Hounkponou^{*1}, Eugène C. Ezin^{1,2}

¹*Institut de Formation et de Recherche en Informatique (IFRI), Université d'Abomey-Calavi, Bénin*

²*Institut de Mathématiques et de Sciences Physiques (IMSP), Université d'Abomey-Calavi, Bénin*

Abstract

Agriculture 4.0 leverages emerging digital technologies (IoT, AI, big data, blockchain) to optimize yield, traceability, and sustainability in agricultural practices. However, small farms and cooperative structures in developing countries struggle to harness these levers due to scattered data, lack of modular tools, and weak infrastructure. This paper proposes the application of the IRADAH (Integrated Requirement Analysis for Designing Data Warehouse) framework and the dimensional Kimball lifecycle to the agricultural sector to implement a Decision Support System (DSS). Based on a state-of-the-art review of Agriculture 4.0 and a synthesis of a Business Intelligence (BI) implementation in a faith-based nonprofit organization, we describe a methodological adaptation of IRADAH for agricultural use, present an open-source BI architecture (PostgreSQL, Talend, Metabase, Docker), and illustrate a prospective proof of concept based on simulated data. Expected outcomes include reliable data consolidation from multiple sources, enhanced digital inclusion, and strengthened decision-making capacity for agricultural actors.

Keywords

Agriculture 4.0, Business Intelligence, IRADAH, Kimball lifecycle, Data warehouse, Decision support system, Open source, BI architecture, Metadata governance

1. Introduction

Agriculture 4.0, or smart farming, relies on the integration of emerging technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), Big Data analytics, blockchain, and robotics to improve productivity, traceability, and sustainability of agricultural operations[1]. These advancements generate large volumes of heterogeneous data (sensors, satellite images, cooperative records, transactional flows) that need to be consolidated, historized, and governed over the long term to support decision-making.

The adoption of digital agriculture in developing countries such as Benin faces major structural challenges that hinder the fair and sustainable diffusion of technological innovations within agricultural farms[2].

Firstly, deficiencies in digital infrastructure are a significant barrier. In many rural areas, Internet access is limited, equipment is scarce, and the reliability of digital services remains low. These technical constraints prevent the homogeneous deployment and continuity of agro-digital tools.

Secondly, the lack of human capacity and digital literacy hampers the appropriation of technologies. The high rate of digital illiteracy among producers, the lack of appropriate training, and the low compatibility of solutions with local contexts limit their effective use. This deficit is exacerbated by a lack of technical skills among innovators, cooperatives, and support structures.

Thirdly, the fragmentation and poor governance of data hinder the consolidation of agricultural information. Data is often scattered among paper records, Excel files, and non-interoperable digital streams. This heterogeneity creates difficulties in analysis, traceability, quality, and security of data, which are essential for strategic decision-making.

CITA 2025 — Emerging Technologies and Sustainable Agriculture, 26-28 June 2025, Cotonou, Benin

^{*}Corresponding author.

✉ frederic.hounkponou@uac.bj (F. S. Hounkponou) ; eugene.ezin@uac.bj (E. C. Ezin)

ORCID 0000-0002-7850-8600 (E. C. Ezin)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

These challenges are compounded by transversal factors: low funding for research and innovation, inadequate economic models of agricultural startups (high dependence on subsidies), and a lack of structured collaboration between digital, agricultural, and public policy actors.

At the same time, our master's thesis focused on the design and implementation, for the Archdiocese of Cotonou, of a data warehouse-based decision support system founded on the Kimball lifecycle and enriched by the IRADAH (Integrated Requirement Analysis for Designing Data Warehouse)[3] framework for business requirement engineering. This approach, combining data-driven, user-driven, goal-driven, and process-driven methods, demonstrated its ability to effectively align the requirements and strategic objectives of the target organization while ensuring data quality and traceability.

Although the faith-based and agricultural sectors may seem distinct at first glance, both involve collective structures with participatory governance (dioceses, cooperatives, producer groups) facing similar challenges: multi-source information management, need for key indicators to guide activities, and scarcity of modular, lightweight, and cost-effective digital tools. Hence the central question of this article: How can the IRADAH framework and the Kimball lifecycle, proven in the context of the decision support system for the Archdiocese of Cotonou, be applied to design a robust, inclusive, and secure open-source BI architecture for agricultural farms?

To address this question, the article is structured as follows:

1. A state-of-the-art review of the challenges and best practices of Agriculture 4.0;
2. A feedback on the use of BI in a nonprofit community structure;
3. The methodological application of the IRADAH framework to the agricultural context;
4. A description of a scalable BI architecture based on open-source solutions (PostgreSQL, Talend, Metabase, Docker);
5. A prospective proof of concept illustrating the transferability of the approach.

2. State of the Art and Experience Feedback

Before presenting our methodological proposal for the agricultural sector, it is essential to establish the conceptual foundations of decision support systems (Sec. 2.1), then to present the concrete experience gained within the Archdiocese of Cotonou (Sec. 2.2). Finally, we will show how the challenges encountered and the solutions provided are directly transferable to agricultural structures (Sec. 2.2).

2.1. Definitions and Fundamental Concepts

Decision Support Systems (DSS) aim to transform raw data, often dispersed and historized, into structured information to facilitate decision-making[4]. Their typical architecture includes[5]:

- Heterogeneous data sources, including operational databases, flat files, IoT flows, or paper forms;
- An ETL (Extract-Transform-Load) pipeline responsible for ingesting, cleaning, and loading data into a staging area;
- A Data Warehouse (DWH), the core of the DSS. It is a centralized, subject-oriented, integrated, non-volatile, and historized data repository[6]. It may be organized into multidimensional data marts;
- An OLAP server offering navigation capabilities (drill-down, roll-up) and slicing/dicing;
- A reporting layer (dashboards, reports, data mining) intended for decision-makers.

The implementation of a DSS generally follows either the Top-Down or Bottom-Up approach. The project supporting this implementation is structured into a lifecycle to mitigate failure risks and ensure sustainability. Several methodologies provide frameworks to manage the design, development, and deployment phases of DSS. These include the Kimball Lifecycle, X-META, and the DWDSF Framework[7].

To enable a DSS to help end users achieve their decision-making goals, it is crucial that its design is based on requirement engineering, aiming to collect business needs in a way that ensures the relative

stability of the multidimensional models underlying data warehouses and aligns with decision-making issues.

Requirement engineering in BI combines four complementary approaches:

- Data-driven, to rely on the structure and quality of available data;
- User-driven, to collect the functional expectations of end users;
- Goal-driven, to ensure the alignment of indicators with strategic objectives;
- Hybrid, to combine the previous approaches. In this context, the IRADAH framework and hybrid approaches[8] such as GranD, CADWA can be mentioned[9].

2.2. Experience Feedback: DSS of the Archdiocese of Cotonou

As part of our master's thesis, following a project based on the Kimball lifecycle and a requirement engineering process founded on IRADAH, we designed and implemented a DSS within the Archdiocese of Cotonou[3].

Requirement engineering with IRADAH made it possible to identify five thematic domains (Diocesan governance and administration, Christian life and pastoral engagement, Financial autonomy and governance, Social and charitable impact, Risk management and compliance) covering different business areas and activities such as monitoring and evaluation of Annual Work Plans (AWPs), monitoring of charitable actions, pastoral tracking, financial management monitoring, and budget transparency.

During the design phase, each thematic domain was modeled as a multidimensional star-schema data mart (DM). A dimensional bus connected these data marts, guaranteeing the uniqueness of common facts (attendance, satisfaction, budgets) and the reuse of dimensions (time, location, and person)[3].

To provide traceability, quality, and security, a single metadata schema was created within the DSS to record indicator definitions, ETL procedures, and access permissions.

The Metabase-implemented reporting layer provided dynamic dashboards customized for each user profile, including department heads, parish priests, and executives. Stakeholder confidence in the dependability of indicators was bolstered by this implementation, which cut report generating timeframes from days to a few hours[3].

The DSS was implemented using a technology stack composed exclusively of open-source solutions to meet the financial and technical constraints of the diocese.

2.3. Functional Similarities with the Agricultural Sector

Collective structures with participatory governance (dioceses, agricultural cooperatives, producer groups) face similar challenges, summarized in four key areas:

- **Multi-source management and heterogeneity:**
 - Agricultural structures: IoT sensors, field surveys, stock registers, input invoices, and cooperative reports;
 - Faith-based organizations: membership forms, pastoral schedules, financial statements, surveys.
- **Need for impact indicators:**
 - Agricultural structures: yield per plot, adoption rate of best practices, delivery time for inputs, member satisfaction;
 - Faith-based organizations: participation level, donation trends, believer satisfaction, number of training sessions.
- **Governance, quality, and data security constraints:** Both agricultural holdings and dioceses require fine-grained traceability (origin, date, responsible party) and granular access control.
- **Requirement for affordable and flexible tools:** In the sub-Saharan region, financial resources and technology infrastructure are frequently constrained.

These similarities show that the Kimball lifecycle and the IRADAH methodological approach provide a strong basis for creating an efficient DSS in the agriculture industry that can meet the demands of digital inclusion, governance, and monitoring.

3. The IRADAH Framework Applied to Agriculture

Before detailing the technical architecture, it is essential to present the IRADAH methodological framework that guides requirement engineering, then show how this framework is transposed to agricultural actors, and finally propose a target dimensional model adapted to agricultural sectors.

3.1. Brief Presentation of IRADAH

The IRADAH framework structures requirement engineering into four complementary phases, each providing a specific perspective on the decision support solution requirements[10]:

- **User-driven:** gathering expectations and constraints of end users — farm managers, cooperative facilitators, union leaders — through interviews, workshops, and questionnaires.
- **Goal-driven:** establishing strategic goals and key performance indicators (KPIs) that reflect the vision of the organization, which includes digital inclusion, sustainability, and resilience.
- **Data-driven:** inventorying and qualifying available or deployable data sources (IoT sensors, paper records, billing systems), assessing their quality and reliability.
- **Process-driven:** identifying business events and items to historize in the data warehouse by modeling operational workflows and business processes, such as crop cycles, input logistics, and member training.

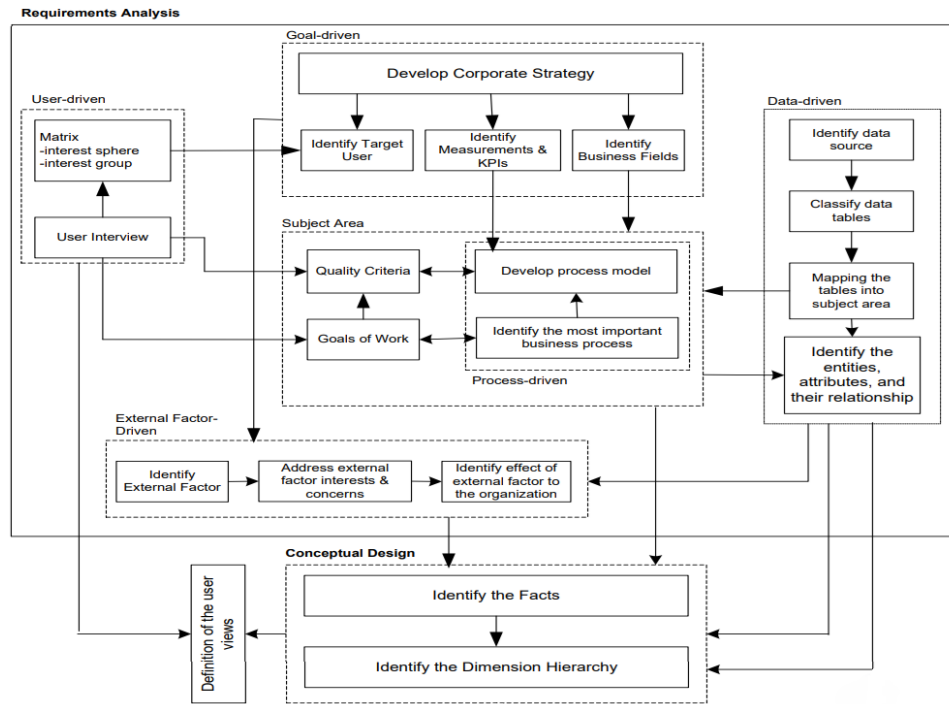


Figure 1: IRADAH framework[10].

The articulation of these axes ensures the complete alignment between the data structure, strategic goals, and operational requirements.

3.2. Applying Methodology to the Agriculture Industry the Agricultural Sector

In order to go from the ecclesiastical experience to agriculture, we first identify important key actors and map their needs:

- **Individual farmers (agro-pastoralists, smallholders):**
 - Granular monitoring of yield per plot and crop cycle;
 - Online or mobile access to weather predictions and summary information on input availability.
- **Cooperatives and producer unions:**
 - Combining aggregated data (stocks, sales, and memberships);
 - Management tools for group input negotiation and distribution planning.
- **Umbrella organizations and syndicates:**
 - Monitoring of collective performance and compliance with standards (organic certification, traceability);
 - Strategic dashboards for decision-making at regional or national level.

The needs mapping intersects these profiles with IRADAH phases (see Table 1).

Table 1
IRADAH Phase vs Agricultural Actors Mapping

IRADAH Phase	Individual Farmers	Cooperatives/Unions	Umbrella Organizations
User-driven	Simple mobile reports	Aggregated web reports	Strategic dashboards
Goal-driven	Yield/cost KPIs per cycle	Cooperation rate, economies of scale KPIs	Compliance, sustainability KPIs
Data-driven	Soil/rainfall sensors, CSV logs	Membership files, invoices, stocks	Certification databases, satellite data
Process-driven	Sowing-harvest cycles, input mgmt.	Group logistics, lot assembling	Audits, reverse traceability, regulation reporting

3.3. Target Dimensional Model

We propose a target dimensional model for an agriculture-focused data mart centered on yield, integrating specific needs of different stakeholders identified in the IRADAH approach.

Table 2
Attributes of fact tables for the Agricultural Yield data mart

Fact table (grain)	Measures / key attributes
Fact_Yield (plot × crop × harvest)	Quantity_Produced, Production_Cost, Labor_Hours, Quality_Index, <i>Gross_Margin†</i> , <i>Yield_Gap†</i> , <i>Sustainability_Score†</i> ; <i>FKs</i> : Plot_ID, Crop_ID, Harvest_Date_ID, Farmer_ID, Org_ID, Season_ID
Fact_Input (plot × input application)	Input_Type_ID, Input_Quantity, Unit_Cost, Total_Cost, Savings, Deviation_From_Recommendation; <i>FKs</i> : Plot_ID, Crop_ID, Operation_Date_ID, Farmer_ID
Fact_Operations (plot × operation event)	Operation_Type, Duration_Hours, Labor_Hours, Machine_Hours, Operation_Cost; <i>FKs</i> : Plot_ID, Crop_ID, Operation_Date_ID, Farmer_ID
Fact_Collective_Performance (organisation × season)	Adoption_Rate, Certification_Compliance, Traceability_Index; <i>FKs</i> : Org_ID, Season_ID

†Derived measures are stored only when historical traceability is required; otherwise they are computed on-the-fly in BI views.

Table 3
Attributes of dimension tables for the Agricultural Yield data mart

Dimension table	Key attributes
Dim_Plot	Plot_ID (SK), GPS_Coordinates, Area_ha, Soil_Type, Irrigation_System, Tenure_Status
Dim_Crop	Crop_ID (SK), Crop_Name, Variety, GMO_Flag, Maturity_Group
Dim_Farmer (SCD-2)	Farmer_ID (SK), Name, Age, Experience_Years, Training_Level, Credit_Access, Household_Size
Dim_Organization (SCD-2)	Org_ID (SK), Org_Type, Membership_Count, Coverage_Area, Current_Certifications
Dim_Date	Date_ID (SK), Calendar_Date, Day, Month, Quarter, Year, Season
Dim_Season	Season_ID (SK), Campaign_Label, Start_Date_ID, End_Date_ID
Dim_Input_Type	Input_Type_ID (SK), Category (Seed / Fertiliser / Pesticide), Product_Name, Active_Ingredient
Dim_Weather	Weather_ID (SK), Avg_Temp, Rainfall_mm, Wind_Speed, Weather_Event_Flag
Dim_Market	Market_ID (SK), Product, Market_Price, Currency
Dim_Subsidy	Subsidy_ID (SK), Program_Name, Amount, Eligibility_Flag
Dim_Alert	Alert_ID (SK), Alert_Type, Severity, Description

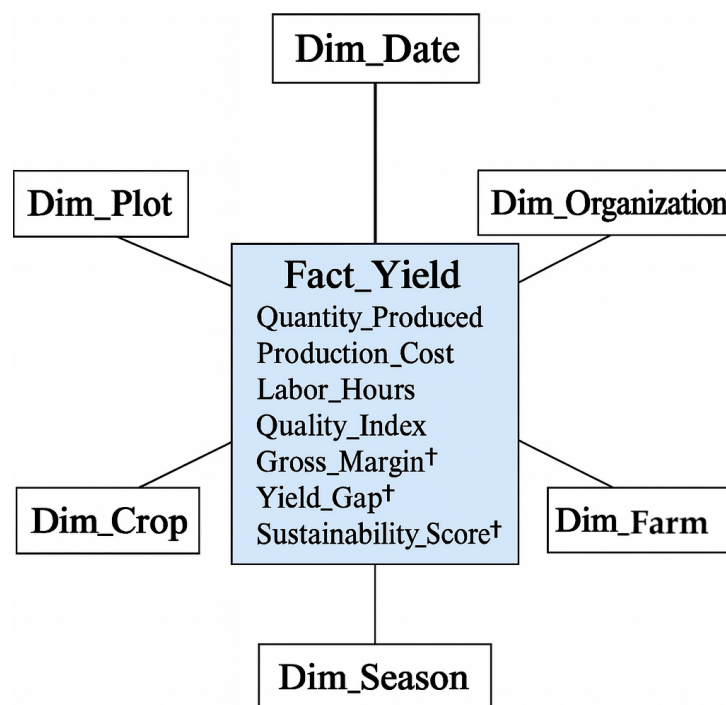


Figure 2: Illustrative star schema of the Agricultural Yield Data Mart Slice

This yield-centric star schema can be reused as the core of future data mart slices (inputs, operations, collective performance) as new analytic questions emerge.

4. Proposed Open-Source BI Architecture

In this section, we describe an end-to-end BI architecture based on proven open-source components. It is structured in three parts: the general functional architecture (Sec. 4.1), the technology stack (Sec. 4.2), and finally metadata governance and quality strategy (Sec. 4.3).

4.1. Functional Architecture

The designed architecture adheres to the traditional four-layer pipeline:

- **Data Ingestion Zone:** where heterogeneous data (weather APIs, simulated IoT streams, and CSV files exported from the field) is entered. Extracting and preliminary cleaning tasks (format standardization, duplication detection, minimum enrichment) are carried out via Talend jobs.
- **Data Warehouse (DW):** based on PostgreSQL, historized and normalized in star schema, fed by Talend ETL processes configured in incremental mode. Each fact and dimension table is based on the modeling defined in Sec. 3.3.
- **Summary Tables:** aggregated tables and materialized views arranged by subject (training, yield, and stocks). These summary tables improve analytical queries and are updated on a weekly or daily basis, depending on the user's needs.
- **Dashboard Layer:** The Metabase interface, which is installed in a Docker container, provides end users (farmers, cooperatives, and unions) with interactive reports and dashboards. Groups and profiles are used to control access privileges.

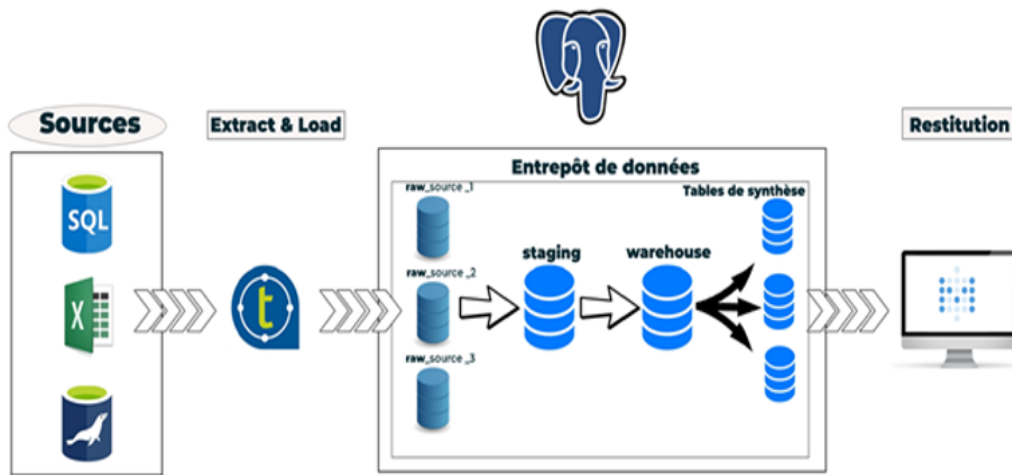


Figure 3: functional architecture of the DSS

4.2. Technology stack

To solve deployment, modularity, and cost restrictions in low-resource contexts open-source solutions are given priority in the design:

- **PostgreSQL** as the DW engine, for its reliability, indexing capabilities, and native support of geographic types (PostGIS) useful for mapping plots[11].
- **Talend Open Studio** to design ETL/ELT workflows, with its graphical components facilitating maintenance and skill development for local teams[12].
- **Metabase** for reporting, due to its intuitive web interface, native support for dynamic filters, and simple Docker deployment[13].
- **Docker** for containerization, which guarantees environment portability, service isolation, and component update simplicity.
- **DBeaver** for administration of the different databases in the DSS.

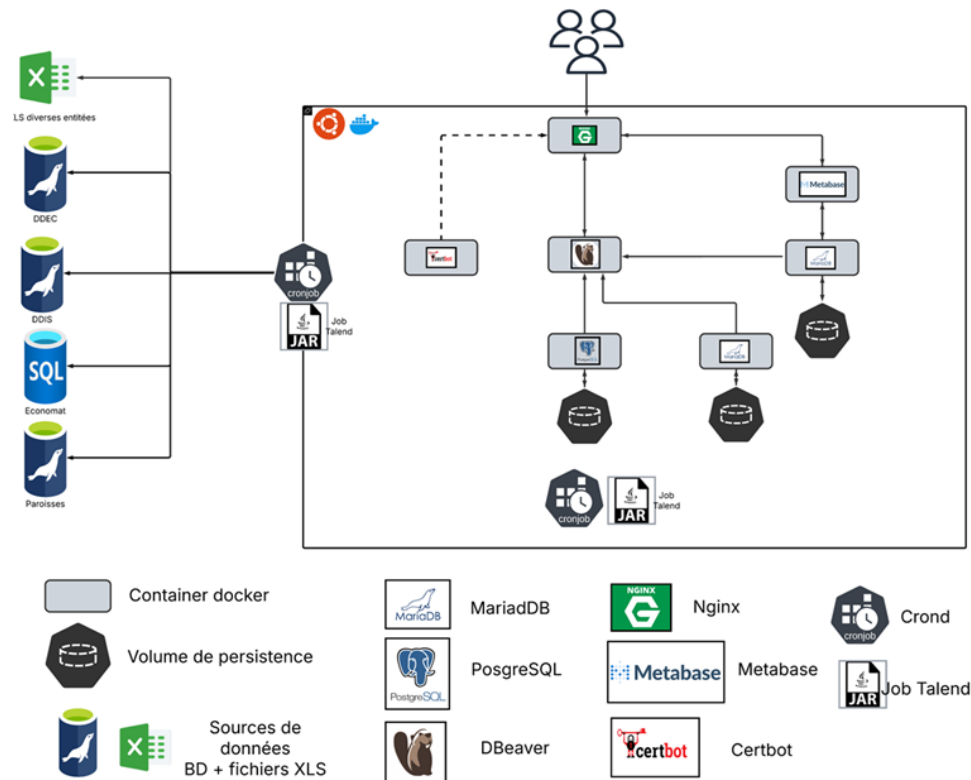


Figure 4: Containerized infrastructure and interactions between components

This component selection makes it possible to create a lightweight ecosystem that is scalable for functional expansion and simple to install on a single server or virtual machine.

4.3. Metadata Governance, Quality Integration and Stress-Tests

The robustness of a DSS depends on formalized metadata governance and an integrated quality policy:

- A centralized metadata catalog (stored in MariaDB) lists for each table, column, and indicator: description, origin, update frequency, and business owner.
- Versioned Talend logs (stored in PostgreSQL) provide the traceability of the ETL process, enabling replays or the diagnosis of irregularities in data batches.
- Referential integrity checks, outlier thresholds, and completeness are examples of automated rules that are used in a quality control approach. These rules are triggered during loading and logged in a monitoring dashboard.
- To guarantee that each profile may only access pertinent data, access permissions are controlled in Metabase (user groups) and at the PostgreSQL database level (roles and schemas).

To ensure the reliability and accuracy of each indicator, advanced validation mechanisms should be put in place: Beyond referential integrity checks and outlier thresholds, we plan to conduct a series of robustness tests (stress tests) and sensor error simulations to measure the impact of missing or corrupted data on key indicators. Automatic alerts are triggered in the event of quality drift (>5% of data outside limits), and a manual review module allows corrections to be made prior to loading into the Data Warehouse

By combining these mechanisms, the architecture ensures not only availability and performance, but also stakeholder trust in the reliability of the produced indicators.

4.4. Edge Computing and Offline Mode

To address rural connectivity constraints, the architecture will include at the end:

- **Edge Nodes** (Raspberry Pi or mini-servers) pre-processing IoT streams (aggregation, compression) and storing data locally in offline mode.
- **Delta Sync**: when reconnecting, only increments are transmitted, optimising bandwidth usage.
- **Adapted mobile interface**: Progressive Web Application (PWA) enabling offline data entry and automatic synchronisation as soon as the network is available.

5. Prospective Proof of Concept (Simulated Use Case)

To illustrate the transferability of our approach, we propose a proof of concept (PoC) based on a typical producer cooperative scenario. This simulation details the monitored key indicators, the fictitious datasets, the ETL process workflow, and the implementation of prototype dashboards, followed by an analysis of the expected benefits and limitations.

5.1. Typical Scenario: Producer Cooperative

Let us consider an agricultural cooperative gathering about fifty small farmers. The following indicators are selected to monitor activities:

- Average yield (kg/ha) per crop cycle and per plot;
- Harvest collection delay (days) between the end of harvest and delivery to the central silo;
- Input availability rate (%) — percentage of fulfilled requests at the beginning of the sowing season.

5.2. Fictitious Datasets

Two types of data are generated over a six-month period to simulate the cooperative's functioning:

- **Simulated IoT streams**: hourly sensor readings (temperature, humidity, harvest flow) formatted in JSON and injected into the staging area.
- **Cooperative records in CSV format**: monthly files containing input requests and deliveries, harvest dates, and delivered quantities.

5.3. Proof of Concept Workflow

The PoC unfolds in four steps:

- **Extraction**: Talend jobs import the CSV files and consume the JSON streams, apply format checks, and store raw data in the staging area.
- **Transformation and DWH loading**: defined transformations populate the fact tables (Yield, Stocks, Training) and dimension tables (Plot, Cycle, Input, Farmer) in the Data Warehouse.
- **Summary table loading**: materialized views and aggregations for each indicator are created and refreshed based on a configurable schedule (daily or weekly).
- **Metabase prototyping**: development of interactive dashboards exposing indicators with dynamic filters by cooperative, cycle, and plot.

5.4. Expected Benefits and Limitations

The simulation highlights several potential benefits:

- Reduction in data consolidation delays;
- Improved indicator reliability through automated checks and process traceability;
- Enhanced accessibility for farmers via lightweight web and mobile interfaces.

6. Discussion

6.1. Advantages

The IRADAH approach, coupled with the Kimball lifecycle and implemented using an open-source ecosystem, presents several advantages for the agricultural sector:

- **Modularity:** each component (ingestion, DWH, Data Marts, dashboards) can be deployed, scaled, or replaced independently, facilitating both functional and technical evolution.
- **Reduced cost:** the absence of proprietary licenses (PostgreSQL, Talend Open Studio, Metabase, Docker) reduces initial investments and maintenance costs — a critical criterion for low-budget farms.
- **Reproducibility:** the definition of a unified blueprint and the standardization of ETL processes guarantee rapid replicability in different contexts (crop types, cooperative sizes) while ensuring result consistency.
- **Digital inclusion:** lightweight and mobile interfaces (Metabase) meet the constraints of intermittent connectivity and the diverse technical profiles of users, thus fostering appropriation.

6.2. Limitations and Mitigation Strategies

Despite these benefits, some constraints must be considered:

- **Dependence on data quality:** the effectiveness of the DSS relies on the reliability of input streams; input errors, missing or incorrect data can significantly impact the indicators, hence the need for robust validation mechanisms. It's why We have integrated stress tests and automated validation workflows to reduce the impact of errors.
- **User training:** even with simple interfaces, initial support and continuous assistance are necessary; the absence of local BI expertise may slow adoption and limit advanced use of tools (custom report creation, ad hoc analyses).
- **Connectivity and infrastructure:** although the system can operate in degraded mode, data synchronization and dashboard access require a stable connection; in rural areas, network interruptions or limited bandwidth may affect responsiveness and update frequency. The mitigation strategy here is to use edge computing and PWA extension that ensure offline operation and minimise bandwidth consumption. Delta synchronisations limit network requirements.
- **Limited real-world deployment.** The PoC relies on simulated data. A pilot deployment is recommended to measure adoption, satisfaction, and performance in a real-world environment (KPIs: collection time, data completeness rate, report generation time).

7. Conclusion and Perspectives

7.1. Summary of Methodological and Architectural Contributions

This article presented the transposition of the IRADAH framework and the Kimball lifecycle, initially developed for the Archdiocese of Cotonou, to the context of Agriculture 4.0 farms. We demonstrated that:

- The IRADAH approach, articulating user-, goal-, data-, and process-driven phases, enables the formalization of varied business requirements and ensures the quality and traceability of business needs and the stability of dimensional conceptual models.
- The Kimball bottom-up method provides a reusable schema for modeling facts and dimensions, guaranteeing consistency and extensibility by integrating DMs via a dimensional bus.
- The DSS project management is firmly based on the Kimball lifecycle.
- PostgreSQL, Talend, Metabase, and Docker are examples of open-source ecosystems that provide agricultural stakeholders in sub-Saharan nations with a flexible, affordable, and easily available infrastructure.

7.2. Recommendations for a Real Pilot and Partnerships

In order to verify this proof of concept in practical settings, we advise:

- Choosing a pilot cooperative with a diverse range of farmer profiles and strong leadership commitment;
- Putting in place a plan for end users' training and ongoing support, which includes hands-on workshops and a support platform;
- Forming alliances with regional organizations (such as INRAB and agricultural associations) and IoT sensor suppliers to supply the DWH with actual field data.

7.3. Future Research Directions

Several directions could extend and deepen this work:

- **Extension to other sectors:** adapting the dimensional model and indicators to specific crops (rice, cotton, horticulture) or livestock;
- **Enhanced predictive analysis:** integrating machine learning modules for yield forecasting, phytosanitary anomaly detection, and irrigation optimization;
- **IoT-DWH automation:** deploying real-time data pipelines between sensors and the warehouse, supporting near-real-time analytics;
- **Advanced traceability:** experimenting with blockchain technology to guarantee the immutability of logistical records and strengthen market trust.

By combining these directions, it will be possible to realize a true smart farming ecosystem capable of addressing the challenges of sustainability, inclusion, and economic performance of agricultural operations in developing countries.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-o3 for the following activities: language refinement, grammar corrections, and occasional structural suggestions. After using this tool, the authors reviewed and edited all generated content as needed and take full responsibility for the publication's scientific integrity and content.

References

- [1] P. Raj, A. K. Dubey, A. Kumar, P. S. Rathore (Eds.), Blockchain, Artificial Intelligence, and the Internet of Things: Possibilities and Opportunities, EAI/Springer Innovations in Communication and Computing, Springer International Publishing, Cham, 2022. URL: <https://link.springer.com/10.1007/978-3-030-77637-4>. doi:10.1007/978-3-030-77637-4.
- [2] R. C. Gbedomon, S. Hounbo, F. Thoto, Profil de l'agriculture numérique et de l'adaptation aux changements climatiques: Cas du Bénin [Digital Agriculture and Climate-Adaptation Profile: The Case of Benin], Technical Report, Centre Africain pour le Développement Equitable, 2024. URL: <https://shorturl.at/DW1fJ>. doi:10.61647/aa84576.
- [3] F. S. Hounkponou, Modélisation et implémentation d'un système décisionnel basé sur entrepôt de données pour l'Archidiocèse de Cotonou, Master's thesis, Université d'Abomey-Calavi - Institut de Formation et de Recherche en Informatique (IFRI), Calavi, Bénin, 2025.
- [4] R. Kimball, M. Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, Wiley, 2002.
- [5] A. Vaisman, E. Zimányi, Data Warehouse Systems: Design and Implementation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. URL: <https://link.springer.com/10.1007/978-3-642-54655-6>. doi:10.1007/978-3-642-54655-6.

- [6] W. H. Inmon, Building the Data Warehouse, 3ème édition ed., Wiley, 2002.
- [7] D. Asrani, R. Jain, U. Saxena, Data warehouse development standardization framework (dwdsf): A way to handle data warehouse failure, IOSR Journal of Computer Engineering 19 (2017) 29. URL: https://www.academia.edu/33745942/Data_Warehouse_Development_Standardization_Framework_DWDSF_A_Way_to_Handle_Data_Warehouse_Failure.
- [8] D. S. R. P. Pooja D. Kavishwar, Hybrid data warehouse development method, International Journal of Scientific Research in Computer Science, Engineering and Information Technology (2021). URL: https://www.academia.edu/62211821/Hybrid_Data_Warehouse_Development_Method.
- [9] N. H. Z. Abai, J. H. Yahaya, A. Deraman, User Requirement Analysis in Data Warehouse Design: A Review, Procedia Technology 11 (2013) 801–806. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2212017313004155>. doi:10.1016/j.protcy.2013.12.261.
- [10] Munawar, N. Salim, R. Ibrahim, Quality-based framework for requirement analysis in data warehouse, in: 2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA), Bandung, Indonesia, 2014, pp. 152–158. URL: <http://ieeexplore.ieee.org/document/7005932/>. doi:10.1109/ICAICTA.2014.7005932.
- [11] M. Stonebraker, L. A. Rowe, The design of the postgres storage system, Proceedings of the 13th International Conference on Very Large Data Bases (VLDB) (1987) 289–300. Also foundational work for PostgreSQL design.
- [12] Talend Inc., Start | qlik talend help, <https://help.qlik.com/talend/>, 2025. Accessed 12 June 2025.
- [13] Metabase Team, Metabase documentation, <https://www.metabase.com/docs/latest/>, 2025. Accessed 12 June 2025.