**integration and dependencies** between the **System Modules & Apps** in the **Kalemie Agro Poultry Management System**:

**🔗 System Modules & Their Interdependencies**

**1. Infrastructure & Control**

* **Core Role**: Provides the foundational structure (farms, houses, flocks, sectors, user accounts).
* **Dependencies**:
  + Used by **Operations** to associate activities with specific flocks or houses.
  + Referenced by **HR & Finance** to assign workers to sectors or houses.
  + Linked with **RBACApp** in **Core System Components** for access control.

**2. Operations**

* **Core Role**: Manages daily farm activities (feeding, watering, cleaning, health, egg collection, slaughter).
* **Dependencies**:
  + Uses **Infrastructure** data to locate flocks and houses.
  + Triggers **InventoryApp** to deduct feed, water, and supplies.
  + Logs labor activities for **HRApp** and **PayrollApp**.
  + Sends data to **Analytics & Reporting** for performance tracking.

**3. Commerce & Inventory**

* **Core Role**: Manages products, stock, suppliers, customers, orders, and sales.
* **Dependencies**:
  + Uses **CatalogApp** to define products (e.g., eggs, meat, equipment).
  + Interacts with **InventoryApp** to track stock movements.
  + Pulls supplier data from **SupplierApp** and customer data from **CustomerApp**.
  + Sends financial data to **FinanceApp** for revenue and cost tracking.
  + Supports **Web App**, **Mobile App**, and **Public Website** for order processing.

**4. HR & Finance**

* **Core Role**: Manages employees, payroll, expenses, and financial records.
* **Dependencies**:
  + Receives task logs from **Operations** for performance and payroll.
  + Tracks product costs and sales from **Commerce & Inventory**.
  + Uses **AccountApp** for user-role mapping and permissions.
  + Sends data to **Analytics & Reporting** for financial dashboards.

**5. Analytics & Reporting**

* **Core Role**: Provides dashboards and reports for decision-making.
* **Dependencies**:
  + Aggregates data from **Operations**, **Commerce**, **HR**, and **Finance**.
  + Visualizes KPIs like egg production, mortality rate, sales, and expenses.
  + Used by **Admin Console** and **Web App** for management insights.

**6. IoT Sensors & Integration**

* **Core Role**: Collects real-time data from sensors (temperature, humidity, feed levels).
* **Dependencies**:
  + Sends alerts to **NotificationApp** in **Admin Console**.
  + Feeds data into **Analytics** for environmental monitoring.
  + Can trigger actions in **Operations** (e.g., auto-refill water).

**7. Mobile App**

* **Core Role**: Enables field workers to interact with the system on the go.
* **Dependencies**:
  + Uses **MobileAPIApp** to access backend services.
  + Interacts with **Operations**, **Inventory**, and **HR** modules.
  + Supports offline mode and syncs with **Core System Components**.

**8. Admin Console**

* **Core Role**: Centralized configuration and system monitoring.
* **Dependencies**:
  + Manages settings for all modules.
  + Uses **AuditLogApp** to track user actions.
  + Sends alerts via **NotificationApp** based on sensor or system events.

**9. API Layer**

* **Core Role**: Provides secure access to system data for mobile and third-party apps.
* **Dependencies**:
  + Interfaces with all modules via **APIGatewayApp**.
  + Uses **AuthAPIApp** for authentication and RBAC enforcement.

**10. Core System Components**

* **Core Role**: Provides middleware, RBAC, and shared logic.
* **Dependencies**:
  + Used by all modules for user authentication, permissions, and session management.

**11. Web App**

* **Core Role**: Main interface for managers and admins.
* **Dependencies**:
  + Accesses all modules via the API Layer.
  + Displays dashboards, reports, and operational tools.

**12. Public Website**

* **Core Role**: Marketing and customer engagement.
* **Dependencies**:
  + Pulls product data from **CatalogApp**.
  + Uses **OrderApp** and **CustomerApp** for online sales.
  + Integrates with **BlogApp** and **ContactApp** for outreach.

**🔄 Example Integration Flow: Egg Collection to Sale**

1. **Operations** logs egg collection →
2. **InventoryApp** updates egg stock →
3. **SalesApp** processes customer order →
4. **FinanceApp** records revenue →
5. **HRApp** logs worker performance →
6. **DashboardApp** visualizes production vs. sales.

Would you like this integration map visualized as a **flowchart or system diagram**? I can also generate a **PDF or slide deck** version for presentations.