**DART Nework functionality**:

**DART Overview:**

1. **Data Sources:**   
   The architecture integrates data from various sources such as Siebel, Caplogix, Single View, Data Quality Management, Global Network Console, and External Systems (Jarvis).
2. **Data Flow and Transformation**:   
   The data ingestion process occurs every 15 minutes, where data from the sources is ingested and transformed. The transformed data is then stored in an Oracle Database. Additionally, an Elastic DB is used for caching purposes to ensure faster response times.
3. **Core Components:**   
   The architecture consists of a backend component responsible for data processing, validation, and synchronization. The frontend component provides user interfaces for different DART applications, including DART Pro, DART Network, DART Mobile, and more.
4. **Middleware:**  
    Kafka acts as a message broker, facilitating communication between DART and other systems like Quadient, Siebel, and TIBCO. TIBCO is used for integration with systems such as Quadient and Siebel.
5. **Notification System:**   
   A Notification Service is implemented to send notifications to users upon completion of Data Fix Requests (DFRs).
6. **DART Applications:**  
    The architecture includes various DART applications tailored for specific teams and purposes. These applications include DART Pro for Global Ops Enablement and Regional Ops teams, DART Network for Network Operations, DART Mobile for IBX Ops teams, DART Staging for M&A teams and Data Quality Management, and DART MS for the Brazil Managed Services team.
7. **DFR Process Flow:**   
   The Data Fix Request (DFR) process flow involves initiation, review and validation, submission, data synchronization across targeted applications, and completion. DFRs are initiated by the respective teams, reviewed, validated, and then submitted. The data is synchronized across systems, and upon successful synchronization, the DFR status is updated to 'Completed'.
8. **User Interfaces:**   
   Each DART application has its own user interface, providing access to specific functionalities. For example, DART Pro UI includes features like Home, Create DFR, Create Asset, DART App, My Reports, and Old Home Screen. Similarly, DART Network UI and DART Mobile UI have their own sets of features.
9. **Validation and Error Handling:**   
   The architecture incorporates validation checks and error handling mechanisms. Soft stops are used for warnings that allow the process to continue, while hard stops are implemented for errors that must be resolved before proceeding.
10. **DFR Lifecycle:**   
    The DFR lifecycle includes initiation, in-progress processing, validation, submission, data synchronization, and completion. DFRs go through these stages to ensure proper handling and synchronization of data.

The diagram visually represents the flow of data and the interaction between different components in the DART Application Architecture.

DART Network :  
  
DFR Lifecycle:

1. **Initiation:**The DFR is created and initiated by the respective team, such as DART Network.
2. **In-Progress Processing:**The DFR enters the in-progress stage, where it undergoes processing and necessary actions are taken to address the data fix request.
3. **Validation**:  
    The DFR is subjected to validation checks to ensure the accuracy and integrity of the data fix. This step helps identify any potential errors or issues.
4. **Submission:** Once the DFR has been validated, it is submitted for further processing and synchronization.
5. **Data Synchronization:** The data fix requested in the DFR is synchronized across all targeted applications, including Quadient, Siebel, Single View, Global Network Console, Caplogix, DCIM, and external systems.
6. **Completion:** Upon successful synchronization and completion of the data fix, the DFR status is updated to 'Completed'.

These steps ensure a systematic and controlled process for handling data fix requests, from initiation to completion, while ensuring proper validation, synchronization, and accuracy of the data fixes.

**DART Network Migration Process:**

DART Network now supports the network Port Migrations as a part of the new updates and enhancements to the existing process.

The requirement was to move from the legacy excel based solution to streamline this through DART application .

**Overview**  
  
The Dart Migration Process is an initiative to streamline migration-related activities from a legacy Excel-based system to the Dart Network Application. It involves creating and managing migration requests, Scope of Work (SOW), Operations Action Requests (OAR), and Migration Requests (MR) while ensuring seamless integration with other systems like Siebel and GNC.  
  
**Key Steps in the Dart Migration Process**  
  
**1. User Profiles and Roles**  
  
1. **User Profiles:**  
• **Global Network Data Users**: Access to DFRs, port migration functionalities, and full read/write permissions.  
• **Dart Network Port Migration Users**: Limited to port migration functionalities based on their roles.  
2. **Role-based Field Access:**  
• **IBX Ops Teams**: Editable fields specific to their roles.  
• **Network Engineers**: Editable fields like NSE comments.  
• **Global Network Data Team**: Read/write access to most functionalities and fields.  
  
**2. Scope of Work (SOW)**  
  
1. **Creation of SOW:**  
• Initiated from the Dart Network homepage via the “Create SOW” button.  
• Users specify switches/ports to be migrated within a particular migration window.  
2. **Field Management:**  
• 46+ predefined fields capture port-related information.  
• Data auto-fetched from systems like GNC and Siebel, avoiding manual intervention.  
3. **User Actions:**  
• Audit conducted by the IBX Ops team with editable fields.  
• Errors routed back to relevant teams for correction via DFRs.

4. **Dynamic Updates:**  
• Users can append new switches/ports even after SOW creation.  
• 15-minute sync ensures updates from other systems are reflected.  
  
**3. Order Activities Requests (OAR)**  
  
1. **Initiation:**  
• Ports marked with the sub-status “Needs Migrating.”  
• Users create OARs for selected ports under a switch or multiple IBXs.

2. **Sequential OAR Workflow:**  
• **Physical Audit OAR** → **Cabling Installation OAR** → **Loopback Testing OAR** → **Migration Window Scheduling.**

3. **Order Number Dependency:**  
• Next OAR cannot be initiated until the previous one is completed and a one-dash order number is generated.

4. **Status Tracking:**  
• Progress of SOW (e.g., “Partially Completed,” “Completed”) is monitored within the Dart application.  
  
**4. Migration Request (MR)**  
  
1. **Creation:**  
• Button to “Initiate MR” generates MR in Siebel.  
• Install and deinstall activities are created and routed to IBX Ops queues.

2. **Pre-requisites:**  
• Fields like Migration Date must be filled before raising an MR.  
• Validations ensure no errors exist in pre-defined fields.

3. **Execution:**  
• Migration status updates as “Successfully Completed” once all activities are finished.  
• Sync jobs update relevant systems like GNC and Siebel with the new details.  
  
**5. Validation and Error Handling**  
  
1. **Pre-Validation and Post-Validation:**  
• Checks for port compatibility, speed, usage, and other parameters across systems (Dart, GNC, Siebel).

2. **Duplicate Port Restrictions:**  
• Ports already part of an SOW cannot be added to another SOW.

3. **Error Management:**  
• DFRs created for data mismatches or updates required before/after migration.  
  
**6. Customization and User Functionality**  
  
1. **Customized Views:**  
• Users can sort, filter, and save preferred column views.  
• Option to restore default views.

2. **Real-Time Monitoring:**  
• Real-time refresh of SOW status with a sync interval of 15 minutes.

3. **Permissions:**  
• Only Global Network Data Users can create DFRs.  
• Port Migration Users have restricted access.  
  
**7. Completion and Submission**  
  
1. **SOW Finalization:**  
• All ports in the SOW are marked as migrated once the migration process is complete.  
• Status updated as “Completed.”

2. **Data Sync:**  
• Clicking the “Submit Data Sync” button ensures details are updated across systems.

3. **Handling Unmigrated Ports:**  
• Remaining ports can be reassigned to other SOWs or deleted.