Phase 5: Deployment, Documentation & Final Presentation

Document Technical Blueprint

Overview

This technical blueprint documents the internal architecture, automation logic, and data flow implemented for the **Automated Network Request Management in ServiceNow** solution. It provides a detailed breakdown of Flow Designer workflows, catalog variable mappings, custom table structures, approval logic, and portal-level configurations, ensuring the solution is transparent, maintainable, and reproducible.

System Architecture Overview

The solution is built using native ServiceNow components and follows a low-code, configuration-driven approach:

- Service Catalog for user request intake
- Flow Designer for automation and approvals
- Custom tables for structured data storage
- UI Policies and Variable Sets for form behavior
- Service Portal for end-user interaction

Flow Designer Workflow

Flow Name

Network Request

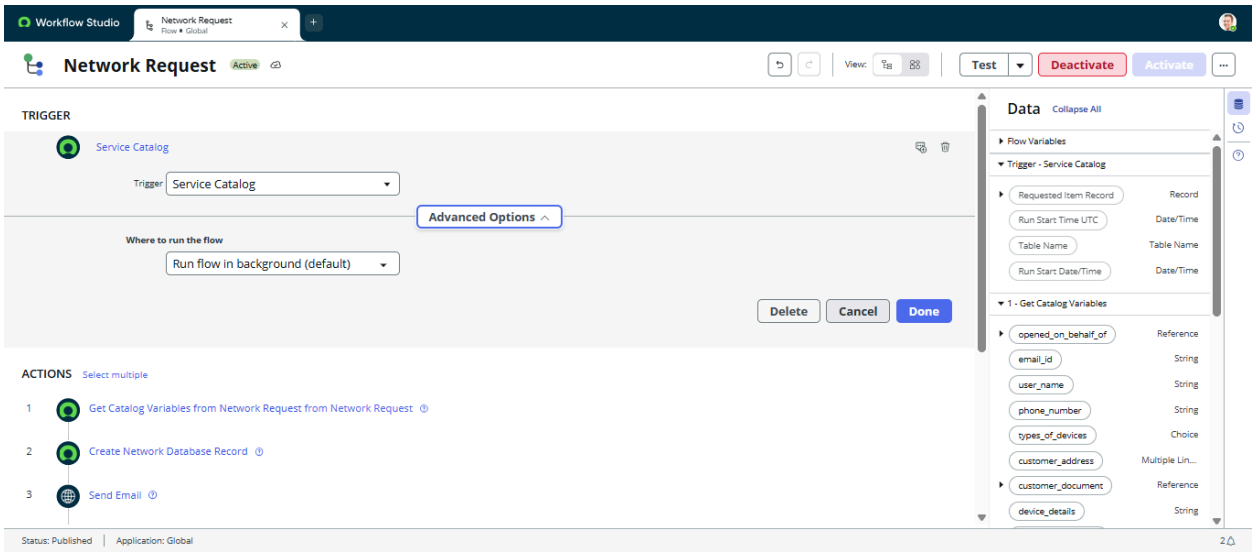
Trigger Configuration

- **Trigger Type:** Service Catalog
- **Trigger Condition:** Requested Item Submitted
- **Catalog Item:** Network Request

This trigger initiates automation immediately after a user submits a network request from the Service Portal.

📸 Screenshot (Recommended)

Flow trigger configuration showing Service Catalog ⟶ Requested Item



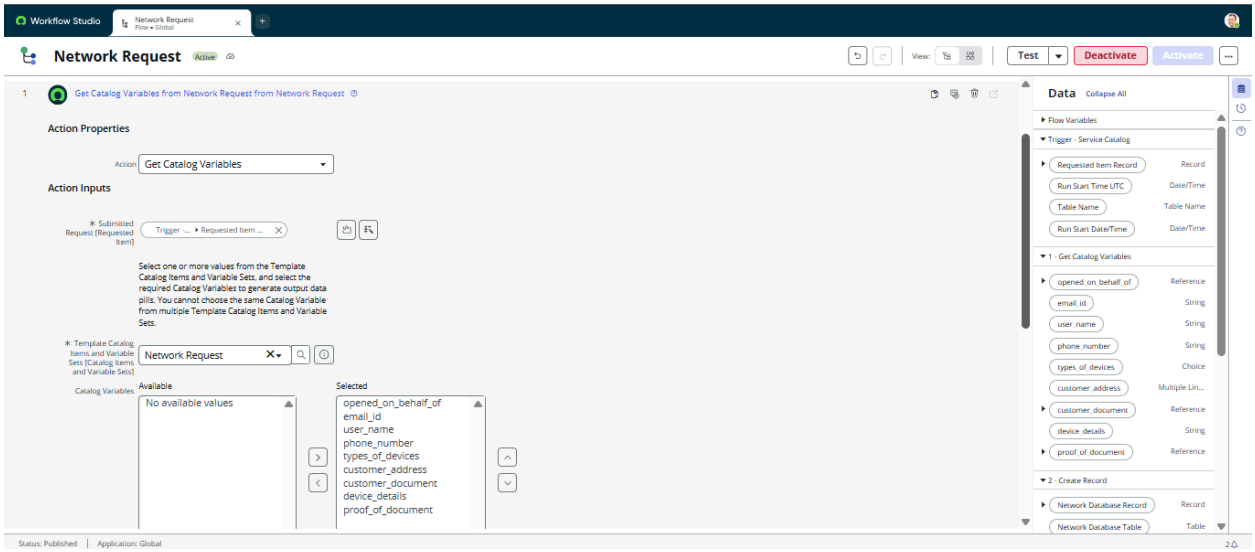
Flow Actions Overview

1. Get Catalog Variables

- Retrieves all submitted catalog variables from the request
- Variables selected explicitly to avoid data loss
- Acts as the primary data source for downstream actions

📸 Screenshot (Recommended)

Get Catalog Variables action with selected variables



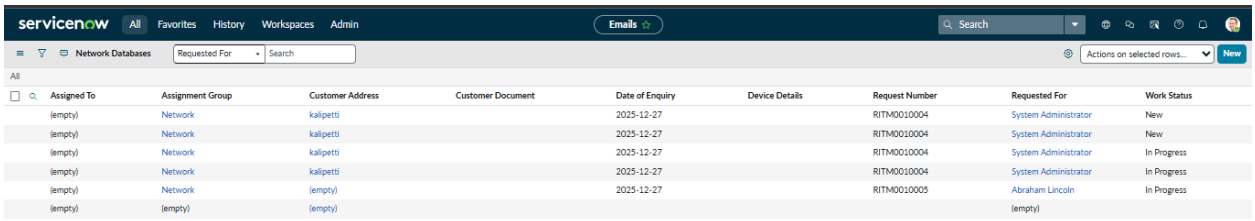
---

2. Create Record – Network Database

- Creates a new record in the custom table **u_network_database**
- Stores request data in a structured format for tracking and reporting

📸 Screenshot (Recommended)

Create Record action with table set to u_network_database



Variable-to-Field Mapping Logic

Catalog variables are mapped to corresponding fields in the custom table using the **Add Fields (+)** option in Flow Designer.

Example Mapping

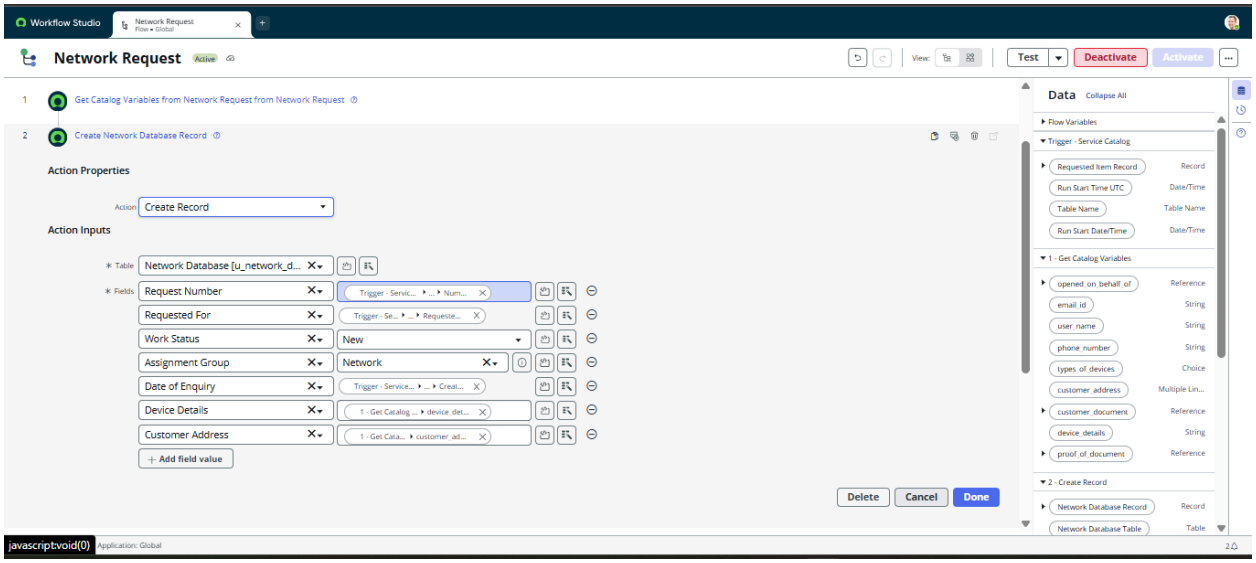
Catalog Variable	Custom Table Field
Request Number	u_request_number
Requested For / Opened on behalf of	u_requested_for
Email ID	u_email
Phone Number	u_phone
Device Type	u_device_type
Device Details	u_device_details
Customer Address	u_customer_address
Assignment Group	Network

Work Status	New
Date of Enquiry	Request Created Date

This mapping ensures:

- No manual data entry
- Consistent data persistence
- Full traceability from request to record

📸 **Screenshot (Recommended)**
Field mapping section showing multiple variable mappings



Approval Logic Configuration

Ask for Approval Action

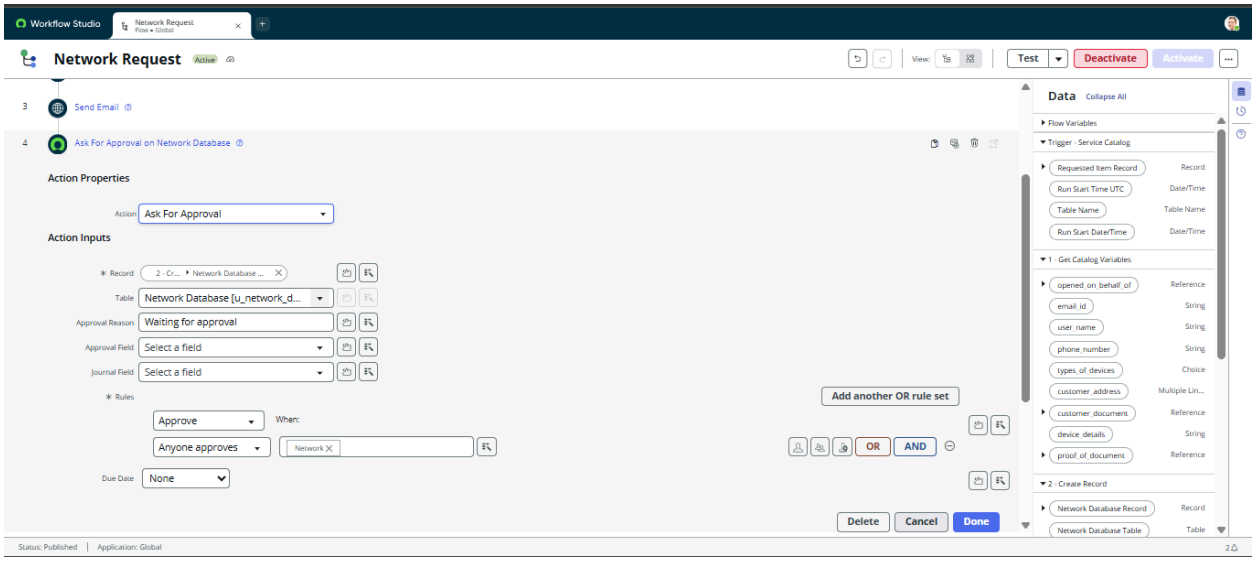
- Approval requested for the newly created **u_network_database** record
- Approval reason: *Waiting for approval*
- Approver assigned dynamically (user or group)

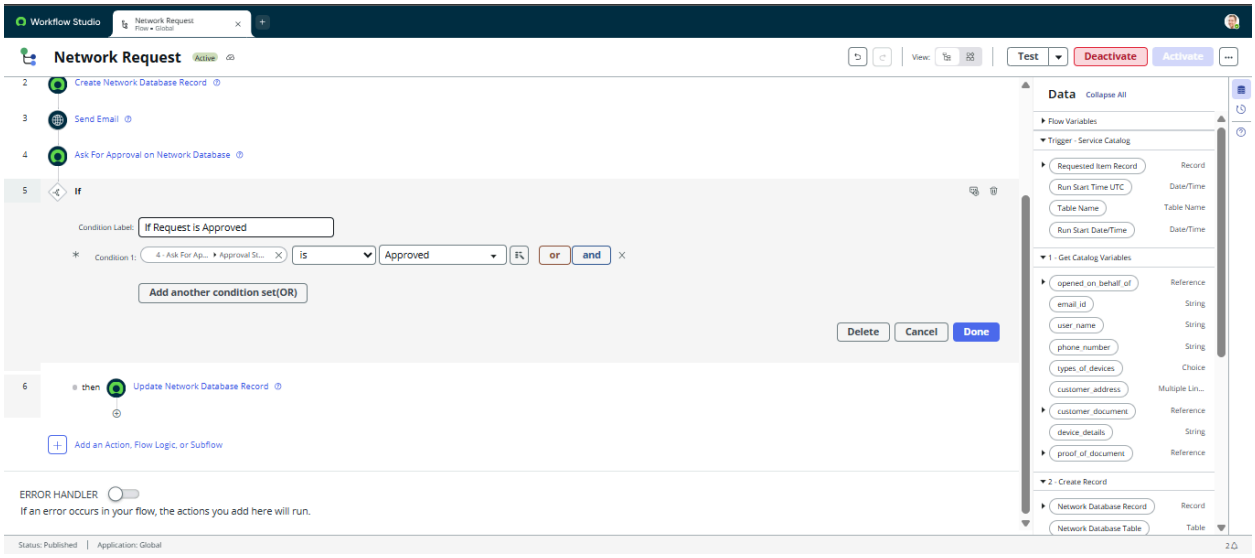
Flow Logic – If Condition

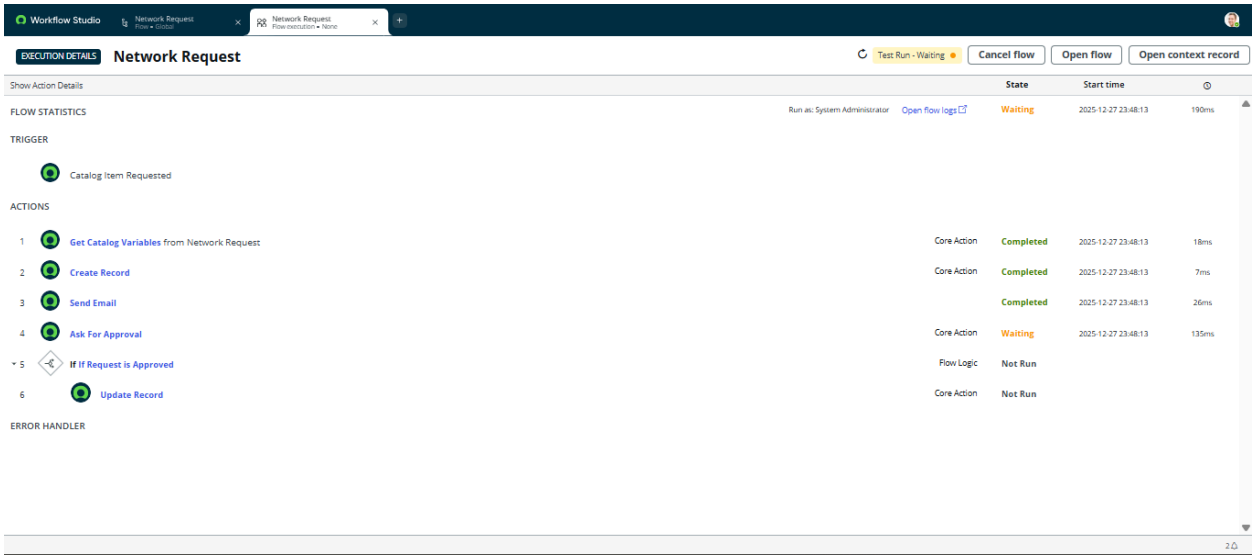
- Approval state is evaluated using Flow Logic
- Conditions:
    - **Approved:** Continue workflow and update record
    - **Rejected:** Stop or mark request accordingly

📸 Screenshot (Recommended)

Ask for Approval action and If condition configuration







Post-Approval Record Update

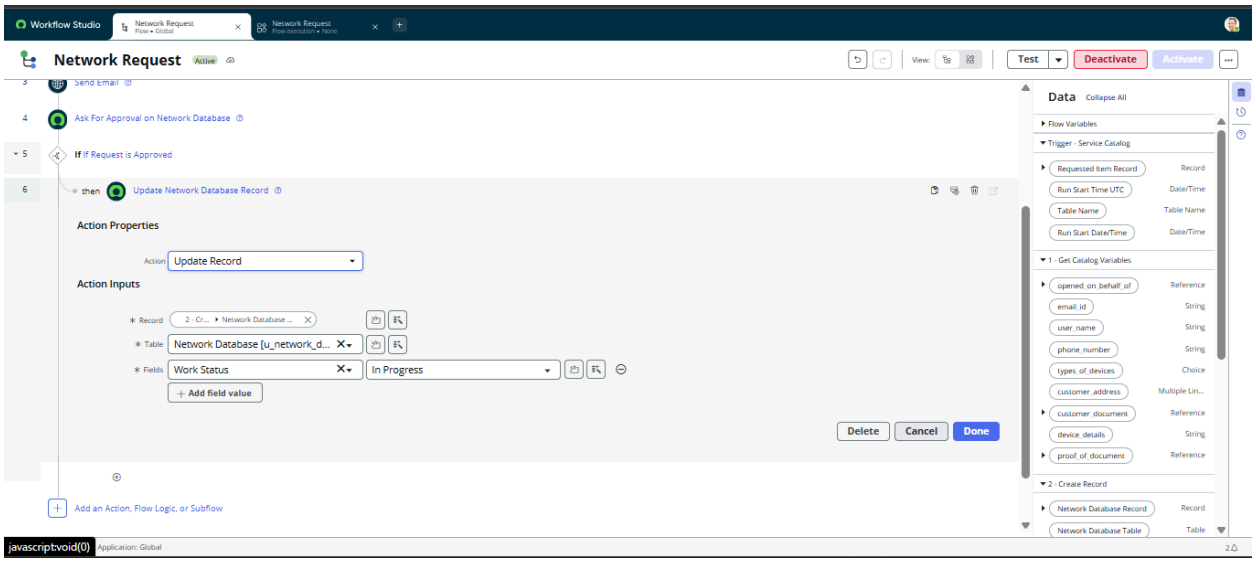
Once approval is completed:

- Network Database record is updated automatically

- Work Status or approval-related fields are modified
- Ensures request lifecycle consistency

📸 Screenshot (Optional)

Update Record action configuration



Custom Table Schema

u_network_database

This table stores all submitted network requests in a structured format.

Key fields include:

- u_request_number
- u_requested_for (Reference → sys_user)
- u_email
- u_phone

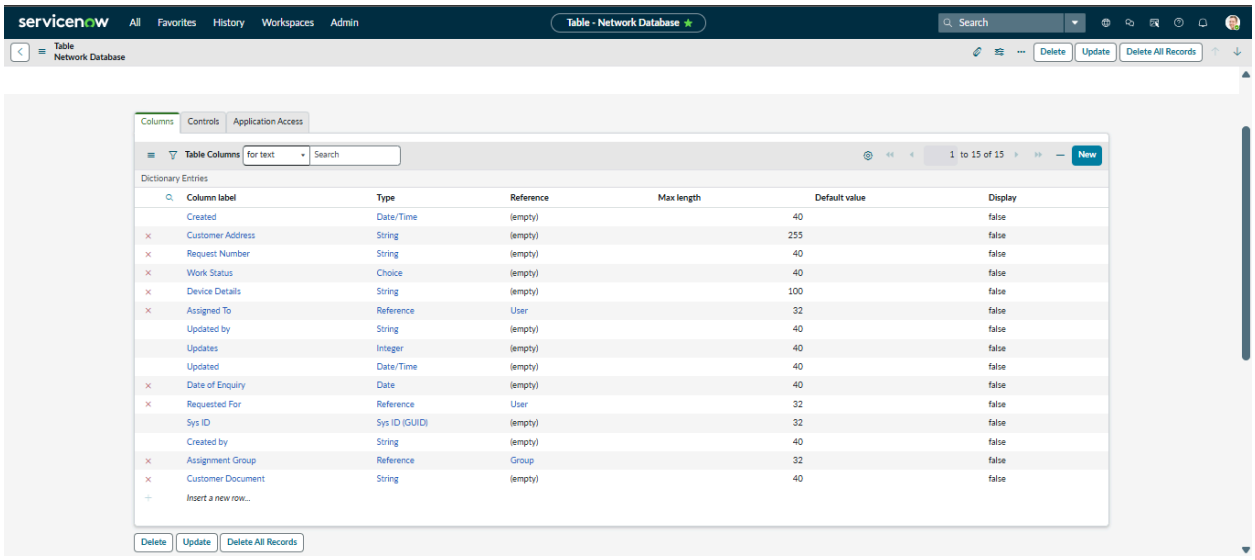
- u_device_type
- u_device_details
- u_customer_address
- u_assignment_group
- u_work_status
- u_date_of_enquiry

u_network_task (if applicable)

- Used for future task tracking or extension
- Supports scalability of the solution

📸 **Screenshot (Optional)**

Table schema showing fields and data types



Service Portal Configuration

- Network Request catalog item is exposed via Service Portal
- Variable Sets used to group user information
- UI Policies control conditional visibility
- No custom portal widgets were required (native portal features used)

📌 **Note:**
If no custom widgets were developed, this is explicitly documented for clarity.

Technical Benefits

- Low-code and maintainable architecture
- Clear separation of UI, logic, and data layers
- Dynamic approvals without hard-coding
- Ready for extension and reporting

Outcome

The technical blueprint confirms that the Automated Network Request Management solution is:

- Architecturally sound
- Secure and scalable
- Fully automated
- Easy to maintain and enhance