FME eBonding with FDT ServiceNow: Technical design and HLD document

Article ID: KB0038073
Created: 2024-11-05 10:29:08
Updated: 2024-11-05 10:29:38
Author: {'link': 'https://fmcnadev.service-now.com/api/now/table/sys_user/9a6b714987495adc2836fc88cebb3519', 'value': '9a6b714987495adc2836fc88cebb3519'}
Category: {'link': 'https://fmcnadev.service-now.com/api/now/table/kb_category/2e50a3d7976dd214e977bc67f053af3c', 'value': '2e50a3d7976dd214e977bc67f053af3c'}

1. Business Summary and End State

This section provides an executive-level summary of the capabilities of the solution, where it will be implemented, what will be included in the system, who will be using it, and any constraints or dependencies. Changes made during the project may require updates to this section.

Project Overview:This project focuses on the ebonding of incidents assigned to the SAP Assignment group between FME ServiceNow and FDT ServiceNow. The primary goal is to ensure that whenever an incident is assigned to the SAP Dispatch assignment group in FME ServiceNow, it is automatically creating incident in FDT ServiceNow instance. This integration will facilitate real-time updates between both systems, enhancing collaboration and incident management efficiency.

Purpose of Document:The purpose of this High-Level Design (HLD) document is to outline the design specifications for the ebonding of incidents between the two ServiceNow instances. It highlights the scope of the project, the functionalities to be implemented, and the expected outcomes.

Scope:The scope of this design includes the following:

Implementation of incident ebonding between FME ServiceNow and FDT ServiceNow for incidents assigned to the SAP Dispatch assignment group through ONEiO.

Synchronization of updates made to incidents on both sides.

This document will not cover catalog item-related functionalities, which will be discussed in a separate document.

And below is the ONEiO Configurations which we must complete to Integrate FME ServiceNow with FDT ServiceNow.

ONEiO Configurations

We have decided to use the ONEiO interface to integrate FME ServiceNow with the FDT ServiceNow. ONEiO is an integration platform that helps solve integration challenges in a more efficient, scalable, and cost-effective way.

In the ONEiO environment, we can maintain two distinct environments: QA and PROD.

We can use the ONEiO QA environment to integrate the FME NON-PROD instance with the FDT NON-PROD instance.We can use the ONEiO PROD environment to integrate the FME PROD instance with the FDT PROD instance.

If we choose FME DEV and FDT DEV instances to integrate, we must complete the following configurations:

Configure FMC DEV Endpoint in ONEiO:

Create a technical user named for example 'oneiofmena' and assign ITIL roles in the FME instance.In the ONEiO platform, configure the FMCDEV endpoint with the following details:

Status: ActiveServiceNow Root URL: https://fmcnadev.service-now.comTimezone: [Specify Timezone]Select Basic authentication to establish connectivity between ONEiO and FMCDEV. Use the ServiceNow technical user 'oneiofmena' to authenticate and complete the endpoint configuration.

ONEiO will generate unique username and password combinations for each endpoint. These credentials should be securely saved to establish connectivity from the FMCDEV instance to ONEiO.

Configure FNC DEV Endpoint in ONEiO:

Create a technical user named 'oneiofnc' and assign ITIL roles in the FNC instance.In the ONEiO platform, configure the FNCDEV endpoint with the following details:

Status: ActiveServiceNow Root URL: https://fncdev.service-now.comTimezone: [Specify Timezone]Select Basic authentication to establish connectivity between ONEiO and FNCDEV. Use the ServiceNow technical user 'oneiofnc' to authenticate and complete the endpoint configuration.

ONEiO will generate unique username and password combinations for each endpoint. These credentials should be securely saved to establish connectivity from the FNCDEV instance to ONEiO.

Integration will be triggered from FME through ONEiO to FDT only when an incident is created or updated in the SAP Dispatch group.Integration will be triggered from FDT through ONEiO to FME only when an FME ebonded incident is updated in SAP Dispatch group.

Create REST Message in FMEDEV

In FMEDEV, create a Basic Auth Configuration named 'ONEiO QA' using the username and password generated in ONEiO during the FMCDEV endpoint configuration.Create a REST Message named ‘ONEiOFME’ and set the endpoint to the ONEiO QA endpoint: https://rest-receiver-test.service-flow.com/api.Select Authentication type as Basic and use the Basic auth profile ‘ONEiO QA’ created in step 1.Create one POST method and select authentication type as inherit from the parent.

Create REST Message in FNC DEV

In FNCDEV, create a Basic Auth Configuration named 'ONEiO QA' using the username and password generated in ONEiO during the FMCDEV endpoint configuration.Create a REST Message named ‘ONEiOFNC’ and set the endpoint to the ONEiO QA endpoint: https://rest-receiver-test.service-flow.com/api.Select Authentication type as Basic and use the Basic auth profile ‘ONEiO QA’ created in step 1.Create one POST method and select authentication type as inherit from the parent.

Business Rules:

Once the REST message is created from both the FME and FNC ServiceNow instances, and a successful connection with ONEiO is established, we can send the incident payload as JSON to ONEiO through the business rule.We can create an "after insert" or "after update" business rule with the condition that the assignment group is "SAP Assignment Group." This will trigger the ONEiO REST message and send the JSON payload to ONEiO.Additional business rules can be created to address different conditions or use cases as needed.Once the conditions are met and the business rules are triggered, ONEiO will receive a sample payload as below.

"ServiceNow-table": "incident",

"active": "1",
"active_name": "true",
"activity_due": "",
"activity_due_name": "UNKNOWN",
"approval": "not requested",
"approval_name": "Not Yet Requested",
"assignment_group": "a93c9243db344ad09b815117f396191f",
"assignment_group_name": "SAP_Dispatch_group",
"caller_id": "801290791b8295d0d65632a99b4bcbd0",
"caller_id_email": "Rajeev.Ponugumati@ext.fresenius.com",
"caller_id_name": "Rajeev Ponugumati",
"category": "service",
"category_name": "Request",
"child_incidents": "0",
"cmdb_ci": "769ca912471f4ad428ef4e3a516d43b0",
"cmdb_ci_name": "fresenius-fkrobrv1-39345249e002",
"company": "f2be979a6f28b1005b09c145eb3ee494",
"company_name": "Fresenius Digital Technology GmbH",
"contact_type": "phone",
"contact_type_name": "Phone",
"description": "TEST",
"escalation": "0",
"escalation_name": "Normal",
"hierarchical_variables": "variable_pool",
"impact": "3",
"impact_name": "3 - Low",
"incident_state": "9",
"incident_state_name": "Assigned",
"knowledge": "0",
"knowledge_name": "false",
"location": "f467c8eb6fa0f1005b09c145eb3ee4dd",
"location_name": "Else-Kröner-Straße 1, Bad Homburg, Germany",
"made_sla": "1",
"made_sla_name": "true",
"notify": "1",

"notify_name": "Do Not Notify",
"number": "INC2015113",
"opened_at": "2024-10-17 04:20:02",
"opened_at_name": "2024-10-17 06:20:02",
"opened_by": "8d76c5ba1bda5dd0102a411acd4bcb6b",
"opened_by_email": "rajeev.ponugumati@capgemini.com",
"opened_by_name": "Rajeev Ponugumati (A)",
"priority": "4",
"priority_name": "4 - Normal",
"reassignment_count": "0",
"reopen_count": "0",
"service_offering": "9c6a72f8db27f34001f4fba51d9619b9",
"service_offering_name": "Network Services - WAN",
"severity": "3",
"severity_name": "3 - Low",
"short_description": "TEST",
"sla_due": "",
"sla_due_name": "UNKNOWN",
"state": "9",
"state_name": "Assigned",
"sys_created_on": "2024-10-17 04:20:38",
"sys_created_on_name": "2024-10-17 06:20:38",
"sys_domain": "global",
"sys_id": "95253d693b911e90cb04952a85e45a78",
"sys_mod_count": "0",
"sys_updated_by": "admin.rajeev.ponugumati",
"sys_updated_on": "2024-10-17 04:20:38",
"sys_updated_on_name": "2024-10-17 06:20:38",
"task_effective_number": "INC2015113",
"time_worked": "1970-01-01 00:00:00",
"time_worked_name": "0 Seconds",
"
"upon_approval": "proceed",
"upon_approval_name": "Proceed to Next Task",
"upon_reject": "cancel",

"upon_reject_name": "Cancel all future Tasks",

"urgency": "2",

"urgency_name": "2 - Medium",

}

Routing rules and mappings in ONEiO:

All field mappings and routing can be managed within ONEiO itself.When ONEiO receives a payload from FME, the goal is to create an incident in FNC ServiceNow.To achieve this, a routing rule must be created, including the necessary field mappings. (Caller, Priority, short description, description, category, CI,Service Offering etc..)The payload will then be routed to FDT, where the incident will be created in FDT ServiceNow.

Below is the sample routing rule for incident creation from FME to FDT

Once the incident is created in FDT, it sends a response back to ONEiO, which in turn sends the response back to FME ServiceNow.

Below is a sample response payload:

{

"Status": "OK",

"ID": "INC46429778",

"CustomerReference": "INC2027748",

"ActionTaken": "Create",

"CorrelationId": "",

"SysID": "6dc78d2b47111e10eca2d24c416d4386",

"TransactionID": "ed9dcf18-1535-4807-a8cf-1521298e8693",

"ResponseId": ""

}

To send the response back to FME, another routing rule must be created, allowing for additional field mappings to be defined. After the incident is created in FNC, the FNC incident number will be sent back to ONEiO. Through the routing rule, this FNC incident number will be mapped to the FME Correlation ID. The FNC incident number will also be updated as the FME Correlation ID, and work notes will be added, for example: “Incident ${number} created in FNC ServiceNow successfully.”

Below is the sample update routing rule from FDT to FME

Multiple routing rules can be created based on different requirements, conditions, and use cases.

Important Notice:

Please do not delete or omit any of the sections listed below. If a section is not applicable to your situation, kindly indicate this by writing "Not Applicable" in the respective section.

1.1. Business Summary

This project focuses on the ebonding of incidents assigned to the SAP Assignment group between FME ServiceNow and FDT ServiceNow. The primary goal is to ensure that whenever an incident is assigned to the SAP Dispatch assignment group in FME ServiceNow, it is automatically ebonded with the corresponding incident in the FDT ServiceNow instance. This integration will facilitate real-time updates between both systems, enhancing collaboration and incident management efficiency.

1.2. Business End State

The end state of this project will see a fully functional ebonding mechanism between the FME and FDT ServiceNow instances. Incidents assigned to the SAP Dispatch group will be seamlessly synchronized, ensuring that updates are reflected in both systems. This will lead to improved incident resolution times and a more streamlined workflow for users.

1.3. Business Usability

The solution will be designed with user-friendliness in mind, enabling users to efficiently manage and track incidents across both ServiceNow instances. Users will benefit from real-time updates and a cohesive interface that reduces the need for manual intervention, ultimately enhancing overall productivity.

1.4. Geographical Consumption

Region

Country

City

Estimated total users at Go-Live

Post Go Live User Growth

(per month/year)

System Architecture Overview

This section provides an overview of the solution architecture and design proposed to meet the required business outcomes. If any changes are made during the project this section may need to be updated.

Important Notice:

Please do not delete or omit any of the sections listed below. If a section is not applicable to your situation, kindly indicate this by writing "Not Applicable" in the respective section.

2.1. Requirements (Key Business and Non-Functional)

2.1.1 Business and Functional Requirements

Core Functionalities:

Incident ebonding between FME ServiceNow and FDT ServiceNow for incidents assigned to the SAP Dispatch group.Automatic triggering of inbound actions when an incident is created and assigned to the SAP Dispatch group.Synchronization of updates to incidents through the ONEiO interface.

Use Cases:

When an incident is created in FME ServiceNow and assigned to the SAP Dispatch group, an inbound action triggers an ebonding to FDT ServiceNow.

Any updates made to the incident in either system is automatically reflected in the other through the ONEiO integration.

2.1.2 Systems (non-Functional) Requirements

Performance: Expected system performance metrics (response time, throughput).Scalability: Define scalability expectations (vertical, horizontal).Security: Implement strong authentication methods (e.g., Basic Auth) and ensure data is encrypted in transit.Availability: Expected uptime, failover strategies, and disaster recovery plans.Compliance: Compliance with GDPR and other relevant regulatory standards.Maintainability: Expected ease of maintenance and upgrade strategies.Usability: The user

interface will follow best practices for UI/UX, ensuring ease of use for incident management tasks.

2.1.3 IT Security Requirements

Additional security measures include using ONEiO's secure credential management for API connections and ensuring role-based access controls (RBAC) are in place.

2.1.4 Constraints

Constraints

Description

Data restrictions

Sensitive data must be handled according to compliance standards.

Security Clearance and Vetting

Personnel involved in handling sensitive data must undergo background checks.

Foreign Ownership, Control or Influence (FOCI)

Compliance with policies governing foreign ownership in sensitive systems.

2.2. High Level Architecture

2.2.1 System Architecture Diagram

2.2.2 Core Architectural Components

ServiceNow Platform Overview: The key platform capabilities leveraged include automation and integrations that facilitate incident management and ebonding processes.

2.2.3 Interaction with External Systems

External Integrations: The integration strategy with external systems will utilize REST APIs via the ONEiO interface.

Data Flows: Seamless data flow between ServiceNow instances through ONEiOFunctional Architecture

High-Level Functional Flow The architecture supports interaction between Incident Management and Request Fulfillment processes, enhancing incident resolution and user experience.Workflows and Automation: Not applicableUser Roles and Access Control: high-level overview of Role-Based Access Control (RBAC) will be implemented to manage user permissions and access rights across both ServiceNow instances, ensuring secure operations.

2.2.6 Non-Functional Architecture

Performance and SLAs:

Response Time

Definition: The time taken by the system to respond to a user request. This includes the time from when a user submits a request until they receive a response.Metric Example:

Average Response Time: Measured in milliseconds (ms), aiming for responses under 200 ms for most user interactions.Maximum Response Time: A threshold that should not be exceeded (e.g., 500 ms).

Throughput

Definition: The number of transactions or requests the system can process in a given time period, often measured in requests per second (RPS).Metric Example:

Average Throughput: Targeting a minimum of X requests per second during peak loads, depending on system capacity and usage patterns.

Error Rate

Definition: The percentage of requests that result in an error or failure. This is critical for assessing system reliability.Metric Example:

Error Rate: Aiming for less than 1% of total requests resulting in errors.

System Availability

Definition: The percentage of time the system is operational and accessible to users.Metric Example:

Uptime Percentage: Targeting 99.9% uptime, translating to roughly 8.76 hours of downtime per year.

User Satisfaction

Definition: Measures how satisfied users are with the system performance and their overall experience.Metric Example:

Net Promoter Score (NPS): Collecting user feedback to gauge satisfaction levels, aiming for a score above a certain threshold (e.g., 70).

Establishing Service Level Agreements (SLAs)

Define SLA Objectives

Identify specific performance targets based on the metrics above. Clearly state the expected levels of performance, such as response times, availability, and error rates.

SLA Components

Service Description: Outline what services are covered under the SLA.Performance Metrics: List the key performance metrics (e.g., response time, availability) along with their defined targets.Measurement Method: Specify how performance will be measured (e.g., monitoring tools, user feedback).Reporting Frequency: Indicate how often performance reports will be generated (e.g., monthly, quarterly).

Consequences of SLA Violations

Define the implications of not meeting SLA targets, which can include:

Compensation: Offering credits or discounts to users for service outages or failures to meet response times.Remediation Plans: Implementing improvement plans to address recurring performance issues.

Review and Update Procedures

Establish regular review periods (e.g., quarterly) to assess SLA performance and make necessary adjustments based on changing business needs or system capabilities.

Communication Plan

Define how and when SLA performance will be communicated to stakeholders, ensuring transparency and accountability.

Scalability: Outline a comprehensive scaling strategy for the ServiceNow platform to accommodate future growth in users, incidents, and data.Security Architecture: Describe high-level security architecture, including authentication methods (e.g., SSO, MFA), encryption, and access control strategies.Compliance: we will implement a comprehensive scaling strategy, the ServiceNow platform can effectively accommodate future growth in users, incidents, and data.

2.2.7 Security Architecture

Authentication and Authorization: Basic authentication will be used to connect both the ServiceNow instances to ONEiO interface.Data Security: ONEiO encrypts the message that it has received and stores it to its database. ONEiO servers have an encryption key used in the AES-256 encryption. This key is generated by ONEiO and known only by ONEiO, so even if someone could get access to the DB, they cannot read the message payloads. The database stores the data and backups using AES-256 encryption as well. Vulnerability Management: Outline how the system will handle vulnerabilities and security patches.

2.2.8 Deployment and Instance Strategy

Instance Strategy:

Development (DEV):

This is the initial environment where all development work is carried out. It is used for coding, configuration, and unit testing by the development team.

Testing (TEST):

Once development is complete, changes are promoted to the TEST instance. This environment is used for system testing, including integration testing and user acceptance testing (UAT), to ensure the functionality meets business requirements.

Production (PROD):

Finally, once all testing is completed and approved, changes are deployed to the PROD environment, where the live application operates. This instance is critical for end-users and must be stable and reliable.

Update Set Management:

1. Understanding Update Sets

Update sets in ServiceNow are packages that contain customizations and configurations. They allow for the migration of changes between different instances (e.g., DEV, TEST and PROD). Managing update sets effectively is crucial for ensuring that changes are deployed smoothly and without disruption.

2. Creating Update Sets

Create Update Set: Whenever a new feature or change is developed, a new update set should be created in the DEV instance. This update set will capture all changes made during the development process.Naming Convention: Use a clear naming convention for update sets that includes the project name, purpose, and story number

3. Capturing Changes

Automatic Capture: Ensure that all changes made in the DEV instance are automatically captured in the update set. This includes scripts, forms, workflows, and any other customizations.Manual Inclusion: For any changes not automatically captured, manually include them in the update set to ensure completeness.

4.Move to TEST Instance

Once development is complete, the story is marked as "Ready for Code Review."Once the code review is passed, the story will be moved to "Ready for QA." Testing will be performed in the Test environment, and if it passes, the story will then be moved for Product Owner approval.Validate Functionality: Perform thorough testing in the TEST instance to validate that all changes work as intended and do not introduce any issues.

5. Deploying to Production

Final Review and Approval: Once testing in the Test instance is complete, the story will be moved for Product Owner approval.Export to PROD: Once the Product Owner has approved, export the update set from the Test instance to the PROD instance through a Change Request.Post-Deployment Monitoring: After deployment to PROD, closely monitor the system for any issues and gather user feedback.

6. Managing Update Set Conflicts

Conflict Resolution: If there are conflicts when importing an update set (e.g., due to changes in the target instance), resolve these conflicts based on business priorities and testing outcomes.Documentation: Document any conflicts and resolutions for future reference.

Post deployment Support: The post-go-live support plan (hyper care) is designed to provide structured assistance to users, customers, and employees after the system implementation. This plan ensures that issues are resolved efficiently and effectively, promoting a smooth transition and high user satisfaction.

3. Technology Stack

Frontend Technologies: The integration does not specifically involve custom frontend technologies, as ServiceNow instances utilize the native user interface provided by ServiceNow for user interactions.Backend Technologies: Both ServiceNow instances are built on the ServiceNow platform, which utilizes JavaScript for server-side scripting and integration processes. The ONEiO interface serves as the integration layer for communication between the two instances.Databases: ServiceNow utilizes a proprietary database system that is built on a relational database model, optimized for its applications. The specifics of the underlying database technology are abstracted from the user.Middleware: ONEiO serves as the middleware platform, facilitating integration between the two ServiceNow instances. It provides an interface for managing data flow and interactions between FME ServiceNow and FDT ServiceNowAPIs/Integrations: The integration leverages REST APIs provided by ServiceNow and the ONEiO platform for data exchange. This includes creating, updating, and querying incidents in both ServiceNow instances through the established endpoints in the ONEiO interface.

4. Assumptions

Both FME and FDT ServiceNow instances are properly configured and operational.Users have the necessary roles and permissions for incident management.ONEiO configurations will be established before the integration process.

5. Risks and Mitigation

The embedded Risk Log below provides a detailed view of all security risks associated with the solution, and the associated treatment/mitigation, if any. All the below risks must be signed off by the respective business owner.

If…

Then…

Impact

Mitigations

Lack of Readiness from FME Side (Catalog Items)

The catalog items that need to be ebonded have not yet been confirmed by FME, which may delay the implementation process and impact overall project timelines.

Delay in project deliverables and potential disruption to project timelines.

Conduct a detailed review of required catalog items with FME to confirm their readiness.

6. Conclusion

Project Overview: This project aims to implement ebonding between FME ServiceNow and FDT ServiceNow for incidents assigned to the SAP Dispatch group. This integration will enhance real-time collaboration and incident management efficiency.System Architecture: The integration will utilize the ONEiO interface for seamless data flow between the two ServiceNow instances, leveraging REST APIs for real-time updates.Security Measures: Strong authentication protocols will be implemented, including role-based access control and data encryption during transmission. A vulnerability management process will be established to handle security patches efficiently. Performance Metrics: Key performance metrics will be defined, including response times and throughput, with established Service Level Agreements (SLAs) to ensure reliability and user satisfaction.Scaling Strategy: A comprehensive scaling strategy will be put in place to accommodate future growth in users, incidents, and data, ensuring the platform can adapt to increased demands.Deployment Strategy: A structured instance hierarchy (Development, Test, Production) will facilitate the promotion of changes through update sets, ensuring a controlled and systematic deployment process.Post-Go-Live Support: A hyper-care support plan will be established to assist users post-deployment, ensuring prompt resolution of issues and ongoing assistance.Monitoring Practices: System health and performance will be monitored using ServiceNow Event Management and external tools, ensuring proactive identification and resolution of issues.

7. Appendix

7.1. Appendix A

Glossary of Terms

Term

Definition

ServiceNow

A cloud-based platform providing enterprise service management software, primarily used for IT services, business operations, and customer service management.

ONEiO

An integration platform that connects various IT service management tools, allowing seamless data flow and communication across different service platforms.

Incident

An unplanned interruption to an IT service or a reduction in the quality of an IT service. Incidents in ServiceNow are recorded and tracked for resolution and management.

Script Include

A reusable JavaScript server-side script in ServiceNow used to hold commonly used functions, logic, or data and accessed by other scripts or business rules.

Business Rule

A server-side script in ServiceNow that executes whenever a record is inserted, updated, deleted, or queried, used to enforce policies or data consistency.

7.2. Appendix B

References and Resources

ServiceNow Documentation

ServiceNow official documentation covering business rules, UI actions, script includes, and integrations. https://docs.servicenow.com

ONEiO Integration Platform Documentation

Resources covering setup, use, and configurations within the ONEiO platform for system integrations. https://support.oneio.cloud/hc/en-us/categories/360002591599-Technical-Documentation\

ServiceNow Developer Community

Community forums and resources offering troubleshooting and coding best practices for ServiceNow. https://developer.servicenow.com

Best Practices in Incident Management

Industry white papers and guides outlining best practices and standards for ITIL-based incident management.

7.3. Appendix C

1) FDT (Fresenius Digital Technologies) Refers to Fresenius Digital Technologies, representing the company. Any mention of "FDT" is in reference to this company entity.

2) FNC Represents all ServiceNow instances where the URL contains "FNC," including examples such as FNCDEV and FNCTEST. This term helps distinguish instances associated with "FNC."

3) FME (Fresenius Medical Care AG & Co. KGaA) Refers to the company Fresenius Medical Care AG & Co. KGaA, which is listed on the Frankfurt Stock Exchange under the ticker symbol "FME." The term "FME" indicates this specific company.

4) FMC (Fresenius Medical) Refers to Fresenius Medical in the context of ServiceNow instances, where URLs contain "FMC" (e.g., FMCNADEV and FMCNATEST). This designation helps differentiate instances specific to "FMC."

7.4. Appendix D

Key Architecture Principles

Modularity and Reusability

Ensure that each component, such as script includes and business rules, is designed for reuse across different incident types or service management scenarios.

Data Integrity and Security

Prioritize secure data transmission, particularly when integrating with ONEiO, to ensure sensitive information is safeguarded during cross-platform data flows.

Automation and Efficiency

Maximize automation within ServiceNow workflows to reduce manual intervention, leveraging ONEiO integration for streamlined incident processing.

Scalability

Architect the solution to handle potential increases in incident volume and new requirements without substantial redesign.

7.5. Appendix E

Technical Assumptions/Dependencies

ServiceNow and ONEiO Integration Availability

It is assumed that both the ServiceNow instance and ONEiO integration are stable, available, and have the necessary permissions for data transfer.

Compatibility with ServiceNow Updates

Any platform updates to ServiceNow will need testing to ensure the custom configurations (UI actions, business rules) continue to function as intended with ONEiO.

Access Control

Proper access rights are assumed for team members working with incident configurations in ServiceNow, ensuring no unauthorized access to incident data or scripts.

S/No

Assumptions/Dependencies

Owner

1

The catalog items that need to be e-bonded should be confirmed

FME

2

Child/Parent Record.

This aspect has not yet been discussed. No agreed approach exists for handling relationships between records during the eBonding process or resolution.

FME and FDT