Terms of Reference

Reusable Component: Audit & Activity History Framework

Mendix Reusable Components Initiative - ISAA

Version	Status
0.0.1	Draft

1. Background

This Terms of Reference document is issued as part of ISAA's (Information Systems Agency of Armenia) initiative to build a standard, reusable foundation of Mendix components for public digital services. Full context, goals, and target users for the overall initiative are described in the accompanying **Concept Note**.

Mendix's native Audit Trail module captures entity/attribute-level changes (old value, new value, who, when) but presents them as raw, technical, per-attribute records. Each project that needs to show users a meaningful history of what happened to an application or case currently has to build its own logic to interpret, group, and secure that data. This component builds on Mendix's native Audit Trail as the underlying capture mechanism, and concentrates reusable functionality on correlation, presentation, configuration, and access control.

2. Objective

Develop a configurable, reusable **Audit & Activity History Framework** for Mendix applications that provides a standardized approach to recording, correlating, and presenting audit and activity history across government digital services, built on top of Mendix's native Audit Trail capabilities.

3. Scope of Work

3.1 Functional Requirements

Standardized Audit Model

Provide a standardized audit model that offers a consistent representation of audit information across different application domain models, while leveraging Mendix's native Audit Trail capabilities as the underlying capture mechanism.

Configuration-Based Adoption

The framework should be configurable for different application domain models, allowing new

applications to adopt audit capabilities with minimal implementation effort.

Business Activity Correlation

The framework should support correlating multiple technical audit events into meaningful business activities, giving users a business-oriented view of application/case history rather than a raw list of field-level changes.

Vendors should describe their proposed correlation mechanism, including:

- How audit entries belonging to the same real-world action (e.g., all field changes made during a single "submit" transaction) are grouped together reliably - e.g., via a correlation/request ID stamped at the transaction boundary - rather than inferred from timing alone;
- How a correlated group of changes is given a human-readable activity label (e.g., "Application Submitted"), whether via explicit logging calls, configurable labeling rules, or both;
- What happens to changes that are correlated but have no configured label (expected fallback behavior, e.g., a generic grouped entry with drill-down to raw field changes).

Non-CRUD Business Events

The framework should provide a reusable mechanism (e.g., a microflow or Java action) for applications to explicitly log business events that are not simple entity attribute changes - for example, a notification being sent, an external API/X-Road call succeeding or failing, or a payment being received - so that these events appear in the same activity history alongside entity-change-derived activities.

Integration with Other Reusable Components

The framework should be usable by, and integrate with, other components (if applicable, for example, the Application & Case Management Framework) providing custom visualization and traceability. The specific integration points and mechanism should be reviewed and defined together with each relevant component's vendor.

Authorization/Access

Provide a reusable authorization model for audit capabilities that integrates with the application's security model. The component should support configurable access to:

- Summary activity history;
- Detailed field-level changes;

without requiring application-specific authorization logic.

Sensitive Data Redaction

The framework should support configurable redaction/masking of sensitive attribute values (e.g., national ID, other personal data) in the activity view and in exports, even where the underlying value is captured by the native Audit Trail.

Export & Reporting

The framework should support filtered export of audit/activity history (e.g., by date range, business

object, user, activity type) for compliance or investigation purposes.

Search & Filtering

The framework should support searching and filtering activity history by user, date range, business object, and activity type.

Retention & Archiving

The framework should support configurable retention periods for audit/activity data, with an archive-then-purge mechanism to prevent unbounded growth of underlying audit tables.

3.2 Non-Functional Requirements

- **Performance**: Correlation and presentation logic should not materially degrade transactional performance of the underlying business microflows; heavy operations (e.g., export, archiving) should run asynchronously.
- **Scalability**: The activity history view should remain performant (e.g., via pagination/indexing) for business objects with a long history.
- **Security**: Redaction and authorization rules must be enforced consistently regardless of access path (UI, export, API).
- **Compatibility**: Must run on *Mendix version > 11.10.0* and be upgradable across supported Mendix versions.

4. Deliverables

Vendors are expected to provide the following, in addition to the specific requirements above:

Standard Implementation Deliverables

#	Deliverable	Description
1	Mendix Source	Mendix application source (Team Server/Git) and deployment package.
2	Technical Documentation	Module purpose, architecture, dependencies, exposed entities, microflows, Java actions, and configuration.
3	Architecture Documentation	Architecture overview and key design decisions.
4	Security Configuration	Module roles, user role mappings, authentication and authorization configuration.
5	API Documentation	Interface specifications, request/response schemas, sample payloads, and error handling.
6	Configuration Guide	Environment-specific constants and configuration settings.

7	Third-Party Component Inventory	Marketplace modules and external dependencies used.
8	Code Quality Report	Static code analysis and Mendix quality metrics.
9	Test Plan	Unit, integration, system, UAT, regression, and non-functional testing approach.
10	Test Results	Test cases and execution evidence.
11	Security Test Report	Security testing and vulnerability assessment results (where applicable).
12	Performance Test Report	Performance and load testing results (where applicable).
13	User Documentation	User and administrator guides.
14	Training Materials	Training guides and supporting materials (where applicable).

Additional Deliverables for Reusable Components

#	Deliverable	Description
1	Module README	Purpose, prerequisites, installation and upgrade instructions.
2	Interface Contract	Public microflows, entities, Java actions, parameters, and expected outputs.
3	Configuration Guide	Post-import configuration instructions.
4	Sample Application	Reference application demonstrating component implementation, including at least one worked example showing correlated business activities, redaction, and export.

5. Vendor Response Requirements

Vendor proposals should address:

1. **Technical approach** - proposed architecture, how it maps to the functional requirements in Section 3, and how it builds on (rather than replaces) Mendix's native Audit Trail.
2. **Configuration model** - concretely how a new application/domain model adopts audit capabilities, and what is/isn't possible without custom development.
3. **Correlation approach** - the mechanism used to group and label business activities (see Section 3.1).
4. **Security & governance approach** - authorization model, redaction mechanism, and export controls.

5. **Reuse & extensibility** - how the framework will be consumed by other components in this initiative, and how cross-component identifiers are carried through.
6. **Team & experience** - Experience with Mendix (certificates, if available) and completed relevant projects.
7. **Timeline** - proposed delivery schedule against the deliverables in Section 4.
8. **Indicative Cost** - cost breakdown by phase/deliverable.
9. **Assumptions & risks** - any assumptions made and risks identified.