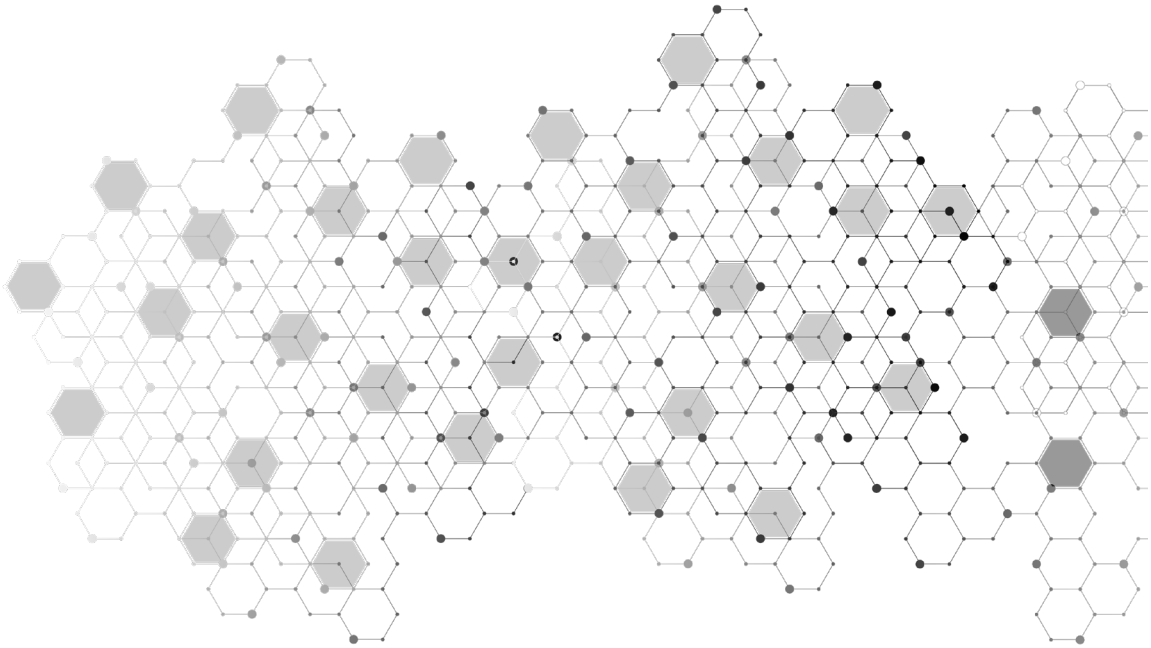
eyko®
know more



WHITE PAPER

eyko Platform Security Architecture

A Technical Whitepaper for System Architects and Security Administrators

WHITE PAPER

Table of Contents

WHITEPAPER: THE EYKO PLATFORM'S MULTI-LAYERED SECURITY ARCHITECTURE

eyko®
know more

WHITE PAPER

1.0

Introduction

A Principled Approach to Data Security

Modern data platforms demand a sophisticated, multi-layered security strategy that is both robust and flexible. As organizations increasingly rely on data for reporting, analytics, and AI, the need to enforce consistent and granular access controls becomes paramount. This whitepaper provides a comprehensive overview of the core principles and architectural layers of the eyko security model, designed for system architects, security administrators, and data governance professionals.

The eyko platform's security is built on three distinct but complementary pillars of defense. These layers work in concert to protect data throughout its lifecycle, from initial ingestion to final consumption in reports and AI queries. We will explore:

1. **Data Selection**: The foundational practice of preventing highly sensitive data from entering the platform in the first place.
2. **Source System Integration**: The seamless inheritance of existing security models from source systems like JD Edwards via the SmartSecurity feature.
3. **Native eyko Security**: The platform's powerful, fine-grained security rules that provide row and field-level control across all data sources.

This document will now examine each of these pillars in detail, followed by practical implementation patterns that demonstrate how they combine to create a comprehensive security framework.

eyko® know more

WHITE PAPER

2.0

The Three Pillars of eyko Data Security

A layered security model is a strategic imperative for any enterprise data platform. This defense-in-depth approach ensures that security is not a single point of failure but a continuous process enforced at multiple stages of the data lifecycle. From the initial connection to a source system to the final query executed by an end-user, eyko's architecture provides overlapping controls to safeguard sensitive information and ensure appropriate access.

2.1 Pillar 1: Foundational Security Through Data Selection

The first and most fundamental layer of security within the eyko architecture is selective data ingestion. This principle dictates that the most effective way to secure highly sensitive data is to prevent it from ever entering the platform.

The standard implementation pattern is to identify tables or columns containing information such as Social Security Numbers (SSNs) or other highly-regulated personal identifiers and deliberately exclude them during the ingestion process. This is achieved by either excluding entire tables from the ingestion scope or by removing specific sensitive columns at the source via a database view or custom query.

The primary benefit of this approach is its elegant simplicity and effectiveness. It confines the most sensitive data to its original, highly controlled source system or a dedicated data warehouse. This allows the eyko platform to focus exclusively on the operational and analytical data required for reporting and AI, minimizing the security footprint and reducing compliance risks.

WHITE PAPER

2.0

2.2 Pillar 2: JD Edwards Integration with SmartSecurity

For organizations using JD Edwards (JDE), eyko provides a powerful mechanism to integrate with and respect the security models already established within the source system. This capability, known as SmartSecurity, ensures consistency and leverages the significant investment already made in defining access controls within JDE.

The core components of SmartSecurity are as follows:

- **Activation**: SmartSecurity is an option that is explicitly enabled on the JDE system connection settings within eyko.
- **Function**: Once activated, this feature instructs eyko to read and apply the JDE security model for that environment. Crucially, for any given entity, it inherits security for only those fields that have explicit JDE security applied to them within the source system.
- **Mapping**: The integration works by mapping users or groups within eyko to their corresponding users or roles within JD Edwards. This mapping allows a user's access rights in eyko to be inherited directly from their established permissions in JDE.

A key constraint in this model is that each eyko user or eyko group can only be mapped to a single JDE user or role for the purpose of security resolution. If a user requires a combination of permissions from multiple JDE roles, two primary solutions are available: modeling a new, combined role directly within the JD Edwards system or augmenting the baseline JDE permissions with eyko's native security rules. This presents a clear architectural choice: security logic can be centralized and managed entirely within the JDE source system for maximum consistency, or a hybrid approach can be adopted to accommodate more flexible, report-specific permissions within eyko.

WHITE PAPER

2.0

2.3 Pillar 3: eyko Native Row and Field Level Security

The third pillar of the architecture is eyko's native security framework, which functions independently of any source system controls like JDE's SmartSecurity. This provides administrators with the flexibility to define highly granular access controls directly within the eyko platform.

These native security rules are applied within a design and are targeted at specific entities and fields. This enables precise row-level or field-level security tailored to specific business needs. For example, a rule can be created to secure the company field on the Account Master entity, restricting a user or group to seeing data for only specific company codes. This same mechanism applies equally to other common filtering dimensions like branch plant, business unit, or region.

Native rules can be assigned to either individual users or eyko groups, and they are automatically respected by all reports and AI queries built from streams that use the secured design. Over time, the strategic focus has increasingly shifted toward applying security at the design level, as these rules are respected by all streams built upon that design. This approach is more explicit, easier to reason about within the context of a data model, and—most importantly—works consistently across both JDE and non-JDE data sources.

These three pillars provide a comprehensive toolkit for data security. The following section details how these architectural components are implemented through common design patterns.

WHITE PAPER

3.0

Practical Implementation and Design Patterns

Understanding the architectural layers is essential, but it is equally important to understand the practical design patterns for implementing them effectively. This section addresses common implementation scenarios and questions faced by security administrators, providing clear guidance on how to configure and combine eyko's security features.

3.1 Configuring Native Security for Users and Groups

When not relying on JDE integration, administrators can apply native security rules directly to eyko groups. This process is straightforward and independent of any source system.

The steps to secure an entity for a specific group are as follows:

1. Navigate to the entity security panel within a design and select the field to be secured (e.g., company or branch plant).
2. In the rule assignment section, select one or more eyko groups as the target for the security rule.
3. Specify the exact data values that the members of that group are permitted to access (e.g., company codes '00001', '00070').

Once configured, all users who are members of the specified group will automatically inherit this security filter in any stream or report that utilizes the secured entity.

WHITE PAPER

3.0

3.2 The Hybrid Model: Combining SmartSecurity and Native Rules

A common and powerful implementation pattern is the use of a hybrid security model that combines the strengths of both SmartSecurity and native eyko rules. This approach provides a balance of leveraging existing security definitions while enabling flexible, model-specific controls.

In this model, the division of responsibility is typically:

- **JDE SmartSecurity**: This is used to establish a broad security baseline. It mirrors the access controls that are already defined, managed, and audited within the JDE source system, ensuring foundational consistency.
- **eyko Native Rules**: These are used to define more specific, primary row-level filters for particular reporting scenarios. This is especially valuable when combining JDE data with non-JDE sources, where a single, consistent security rule needs to be applied across the blended dataset.

3.3 The Architectural Principle of Security Propagation

A cornerstone of efficient and maintainable security architecture is the principle of defining a rule once and having it apply broadly. The eyko platform is designed around this concept via security propagation, a mechanism engineered to minimize administrative overhead, reduce the risk of configuration errors, and ensure consistency across complex data models.

A security rule applied to a core entity (e.g., a company filter on the Account Master entity) can be automatically inherited by other related entities (e.g., GL Ledger) it is joined to. For this to work, the eyko stream must be structured with relationships, such as a one-to-many join, that allow filters to propagate from the secured entity to the related entities. When a user queries a stream configured this way, the filter applied to Account Master effectively constrains the results from all downstream tables.

A security rule would only need to be repeated on another entity if that entity can be queried independently, introduces a different reporting grain, or is not connected within the eyko stream in a way that allows for filter propagation.

WHITE PAPER

3.0

3.4 Maintaining Security for Evolving Data Models

As data models evolve, administrators must ensure that security controls remain effective. When a new table is added to a design, a new security rule is not always necessary.

A new rule is only required if the new table introduces a new analytical grain or contains sensitive fields not already covered by an existing secured entity in the model. If the new table simply joins to an entity that is already properly secured (for example, via the Account Master), and it does not independently expose sensitive data, the existing security rules will likely be sufficient to protect the data.

This concludes our look at practical implementation patterns. The final section will summarize the key attributes of the eyko security framework.

eyko® know more

WHITE PAPER

Conclusion: A Flexible and Robust Security Framework

The eyko security architecture provides a multi-layered, principled framework for robust data governance. It combines three pillars: proactive data selection to minimize the security footprint; seamless integration that leverages existing JDE security investments as a foundational baseline; and powerful native controls for applying consistent, granular rules across both JDE and non-JDE sources. This defense-in-depth strategy provides organizations with the architectural flexibility and comprehensive tools necessary to secure data throughout its lifecycle, ensuring that all analytical and AI workloads are built on a foundation of trust and security.

eyko® know more