Advanced Security

DATA SHEET

Progress® MarkLogic® Server was built to offer enterprise-grade, fine-grained security controls organizations require to implement their information security and data access policies. Out of the box, MarkLogic provides Document Level Security, Element Level Security, Auditing, Support for External Authentication (LDAP and Kerberos), Compliance Archives, Encryption and additional security features.

For certain use cases, there is an Advanced Security option, which includes three additional capabilities:

- **External Key Management** – Provides support for external, third-party key management systems that are KMIP 1.2 compliant.
- **Redaction** – Helps prevent leakage of sensitive information to unauthorized users when importing, exporting or copying data into and out of MarkLogic.
- **Compartment Security** – Additional security control to specify that a user must have all the right roles to interact with a document rather than just one of the right roles.

© 2024 Progress. All Rights Reserved.

Overview of MarkLogic Security

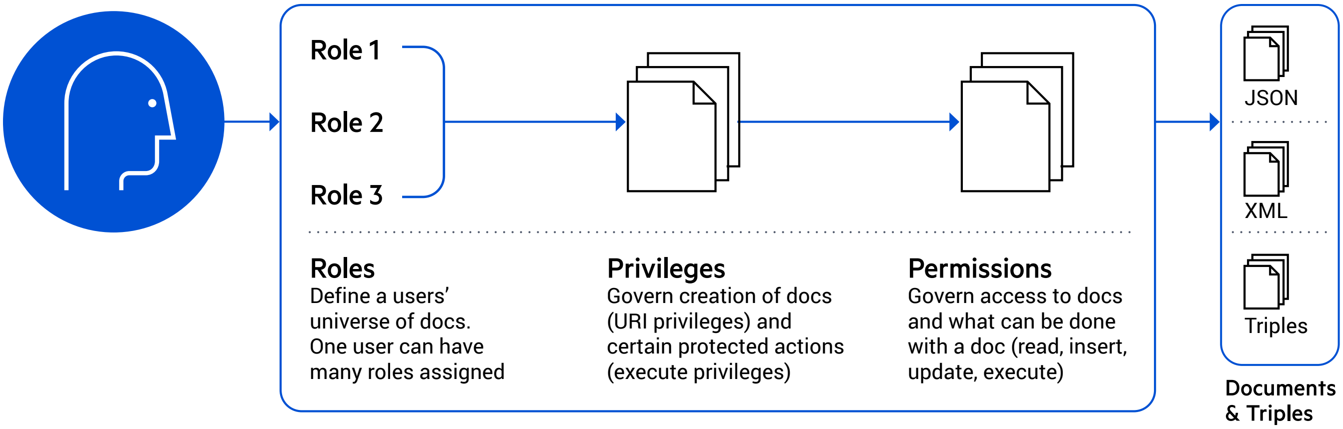
MarkLogic has been in the business of helping organizations protect and secure data for over a decade in both commercial and government settings. MarkLogic was the first non-relational database to be Common Criteria-certified. The Common Criteria for Information Technology Security Evaluation is the driving force for the widest available mutual recognition of secure IT products worldwide.

MarkLogic is installed and operational on systems that require databases to meet extremely rigorous requirements. These requirements include stringent measures for access, authentication, management, audits, role separation and system assurance.

It is for this reason that MarkLogic is chosen to run the most demanding, mission-critical applications at the heart of large investment banks, major healthcare organizations and classified government systems. Our customers have received Authority to Operate (ATO) for information systems utilizing MarkLogic that involve nearly all major systems security standards, including HIPAA, FedRamp, FDA and STIG.

By default, MarkLogic uses a role-based access control (RBAC) security model in which each user is assigned any number of roles, and these roles are associated with any number of privileges and permissions. Privileges govern the creation of documents and execution of functions (URI and execute privileges), while permissions govern what can be done with a document (read, insert, update, execute).

Role-Based Access Control At The Document Level



Role 1
Role 2
Role 3
Roles
Define a users' universe of docs. One user can have many roles assigned
Privileges
Govern creation of docs (URI privileges) and certain protected actions (execute privileges)
Permissions
Govern access to docs and what can be done with a doc (read, insert, update, execute)
JSON
XML
Triples
Documents & Triples

Figure 1: Each specific role contains both privileges (write and execute) and permissions (read and modify)

© 2024 Progress. All Rights Reserved.

External Key Management

Encryption at rest provides transparent encryption of databases, logs, configuration files and backup, with separate and powerful key management through a local Key Management System (KMS) or external KMS. MarkLogic supports both local and external KMS. A KMS, or “keystore,” is a secure location where the enveloped encryption keys used to encrypt data are stored. Key management in MarkLogic includes:

- **End-to-end encryption support** – MarkLogic provides encryption at rest. This enables transparent and selective encryption of data residing on disk (locally or in the cloud), which helps maintain confidentiality and prevent information tampering.

- **Separation of duties** – Encryption at rest significantly enhances data security controls by helping to enforce separation of duties. The system administrator with access to the host is not the same person who has control over the encryption keys and the encryption key lifecycle. This helps reduce the potential threat from insiders and hostile entities residing in the network, such as Advanced Persistent Threats (APTs).

The best practice for encryption at rest is to use an external, third-party KMS that is deployed and managed separately from the application servers. The external KMS securely stores authentication or encryption keys entrusted to it and provides them on demand to authorized systems. This adds another level of security by storing authentication keys separate from the storage system. The keys are never displayed in clear text. An external KMS enabled by the MarkLogic Server Advanced Security option provides:

- **Additional security** – An external KMS offers additional security for encryption keys, along with key management capabilities like automatic key rotation, key revocation and key deletion.

- **Additional separation of duties** – If an external KMS is used, then neither an unauthorized database admin, system admin, nor the storage admin can access the database files. The external KMS admin controls access to the encryption keys.

- **KMIP compliance** – Key Management Interoperability Protocol (KMIP) is a communication protocol standard that defines message formats for the manipulation of cryptographic keys on a key management server. MarkLogic interoperates with third-party external KMS systems that are KMIP 1.2 compliant, including Amazon Web Services (AWS) Key Management Service (KMS) and Microsoft Azure Key Vault.

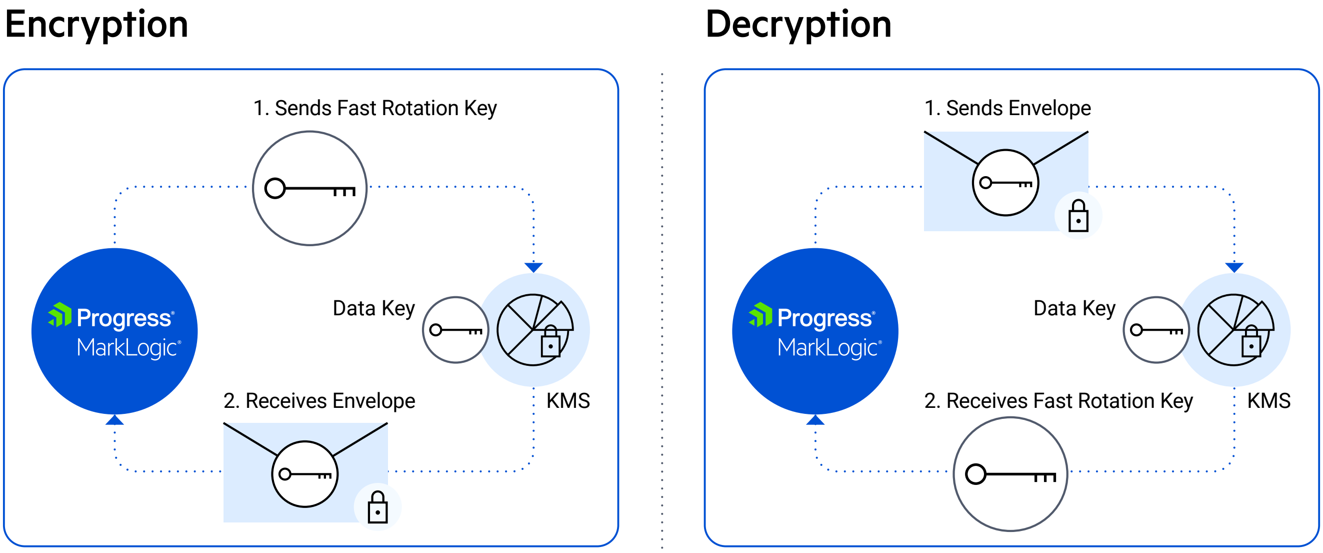


Figure 2: With Encryption at Rest, to access low level keys and read files, MarkLogic sends an envelope to the KMS, which then sends back the unencrypted key. If using an external KMS, MarkLogic has no access to envelope keys, which means no access to files, no ingestion and no compromises.

© 2024 Progress. All Rights Reserved.

Redaction

Redaction helps prevent leakage of sensitive information to unauthorized users when importing, exporting or copying data into and outside of MarkLogic. For example, redaction is often required when providing data for analysis by data scientists, or when a developer needs production data but should not have access to real credit card data or personally identifiable information.

Here are some key characteristics of redaction:

- **Based on Rules and Policies** – To implement, a MarkLogic security administrator creates redaction policies that contain rules, defining which sensitive information should be redacted, then chooses which policy to apply when running an export. Administrators can combine built-in or custom rules into policies to match different target needs.

- **Utilizes Built-In Functions** – Includes built-in functions for different types of redaction:
  - Concealing: Hide elements and/or their values (or properties and/or their values in the case of JSON).
  - Masking: Change the data using random masking (the value varies with each instance), deterministic masking (the same value is applied every time) or dictionary masking (the value is applied from a specified dictionary).
  - Patterns: Change the data using a pattern such as Social Security Number, US phone number, email, IPv4 or Regex.
  - Custom: Use server-side JavaScript or XQuery functions to apply unique rules (e.g., redact the name if the person is under 18 years old).

- **Fully Auditable** – All rules and actions taken by users are logged, which allows the export activity to be audited in the future.

- **Performs Batch at Scale** – Redaction is designed to be used when running large bulk exports. By utilizing the MarkLogic Content Pump (MLCP), it is faster and more secure than solutions implemented at the application layer.

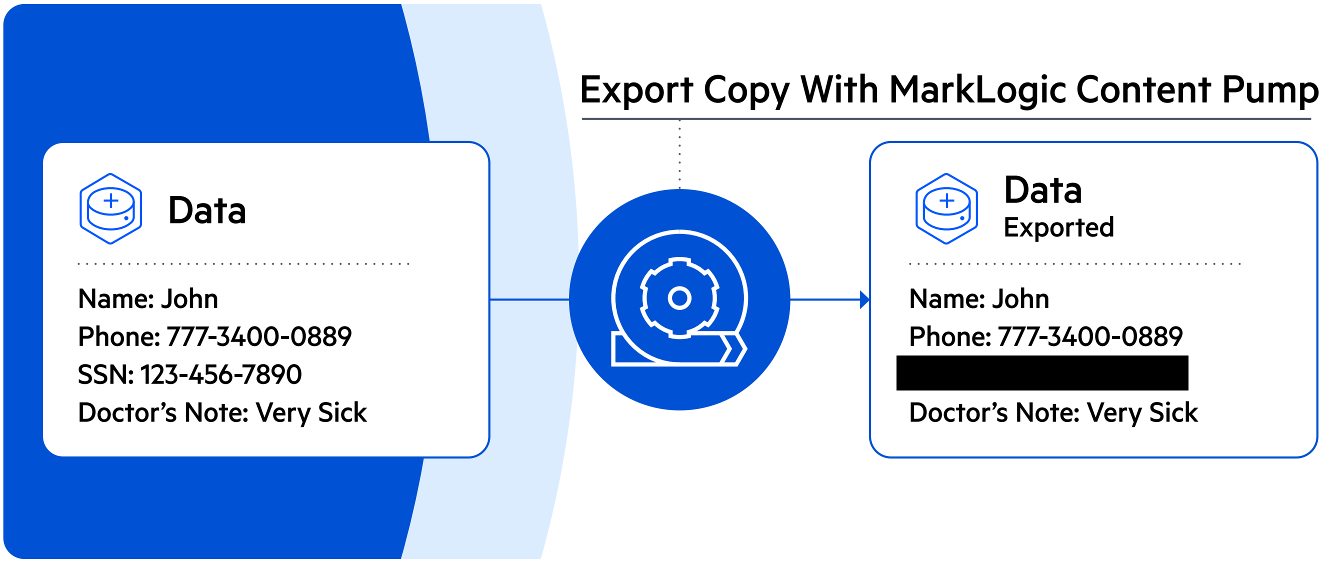


Figure 3: With Redaction, key information can be removed or masked

Compartment Security

With Compartment Security, more complex role-based security rules for data access and updates can be applied. It is possible to specify that a user can be assigned more than one role to access or create a document, offering tighter, more restricted permissions than role-based controls. When a role is compartmented, all privileges associated with a resource must be valid at the same time (AND semantics). However, when roles are not compartmented, satisfying any privilege authorization condition will be sufficient (OR semantics).

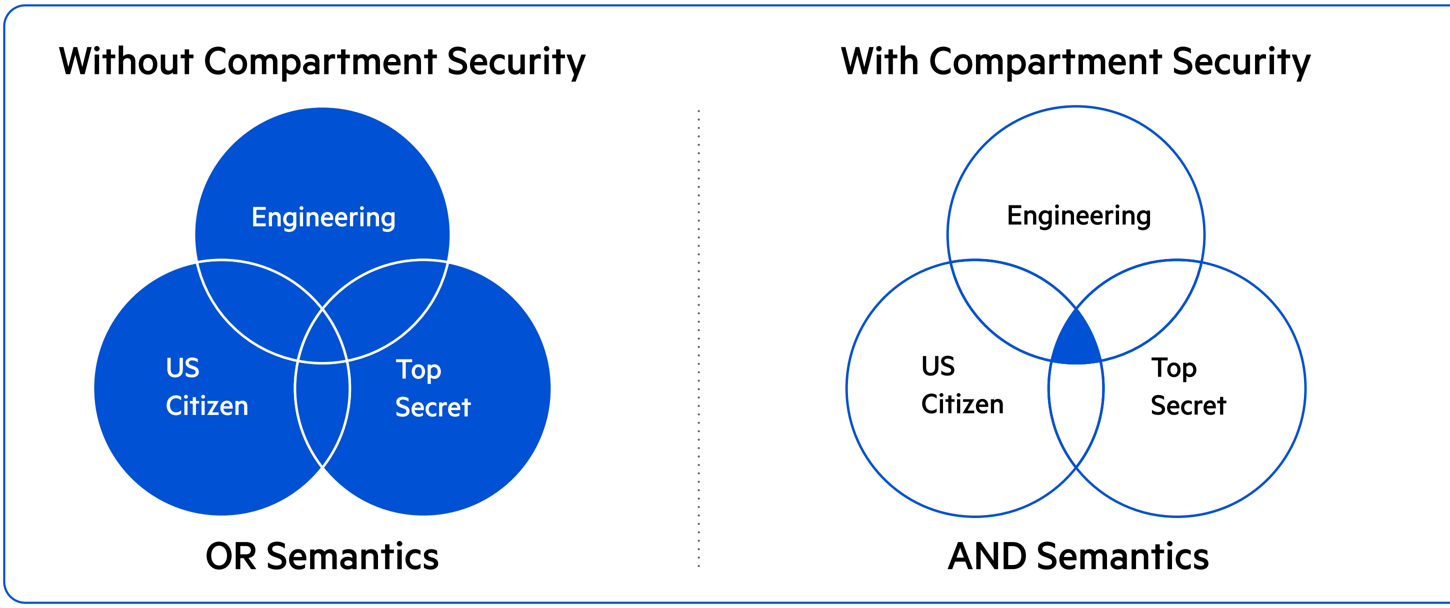


Figure 4: With Compartment Security it can be specified that only those with the combined roles of "Top Secret," "US Citizen" and "Engineering" can access the data. Without Compartment Security, anyone with at least one of the roles can access the data.

For example, if a document has read permission for role1 and read permission for role2, a user who possesses either role1 or role2 can read that document. If those roles have different compartments associated with them (for example, compartment1 and compartment2, respectively), then the permissions are checked using AND semantics for each compartment, as well as OR semantics for each non-compartmented role. To access the document, if role1 and role2 are in different compartments, a user must possess both role1 and role2 to access the document, as well as a non-compartmented role that has a corresponding permission on the document.

Another example would be when a government document is classified at the "Top Secret" level with an additional security marking of "NOFORN" ("no foreign nationals"), it cannot be read unless the user has both a "Top Secret" role and a role that describes the individual as a citizen of that country.

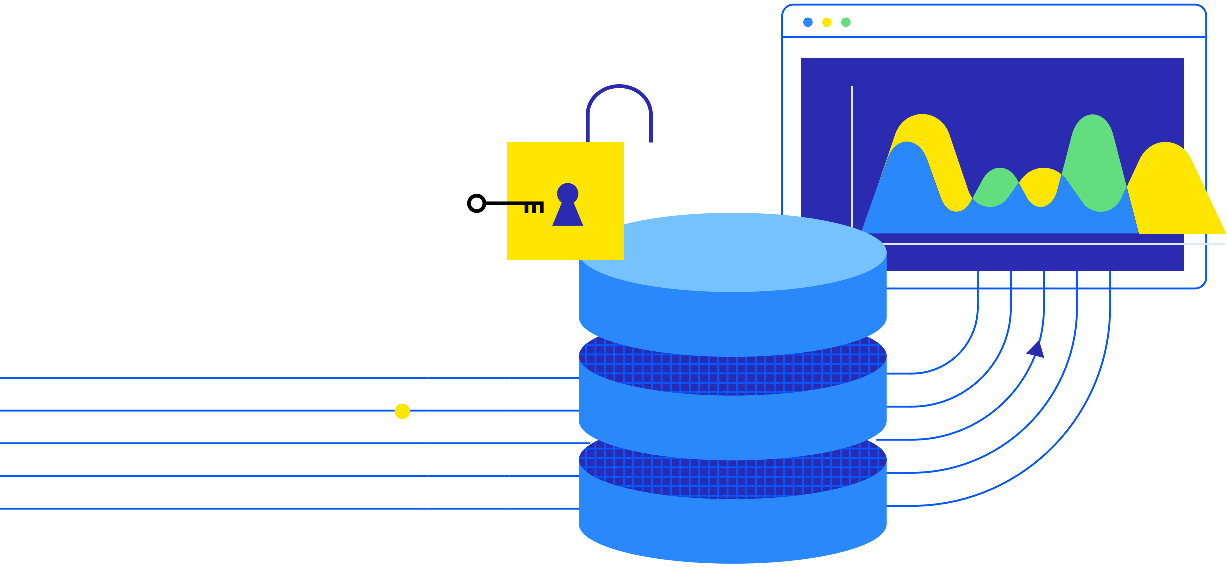
© 2024 Progress. All Rights Reserved.

When to Use

External Key Management: Provides additional separation of concerns and ease of management for storing encryption keys. This option is helpful when you want to leverage an external Key Management System (KMS) that is already in use.

Redaction: Enables you to remove or obscure pieces of your data when exporting data for sharing. This feature can help you meet compliance guidelines like HIPAA, SEC17a-4, FINRA and GDPR.

Compartment Security: Further restricts data access by requiring users to have more than one role to view data, not just one of the right roles. It's often used to protect classified material in government systems.



Lean More at progress.com/marklogic

About Progress

Progress (Nasdaq: PRGS) provides software that enables organizations to develop and deploy their mission-critical applications and experiences, as well as effectively manage their data platforms, cloud and IT infrastructure. As an experienced, trusted provider, we make the lives of technology professionals easier. Over 4 million developers and technologists at hundreds of thousands of enterprises depend on Progress. Learn more at www.progress.com

© 2024 Progress Software Corporation and/or its subsidiaries or affiliates.
All rights reserved. Rev 2024/02 | RITM0231292

f /progresssw
X /progresssw
▶ /progresssw
in /progress-software
◎ /progress_sw_

Progress