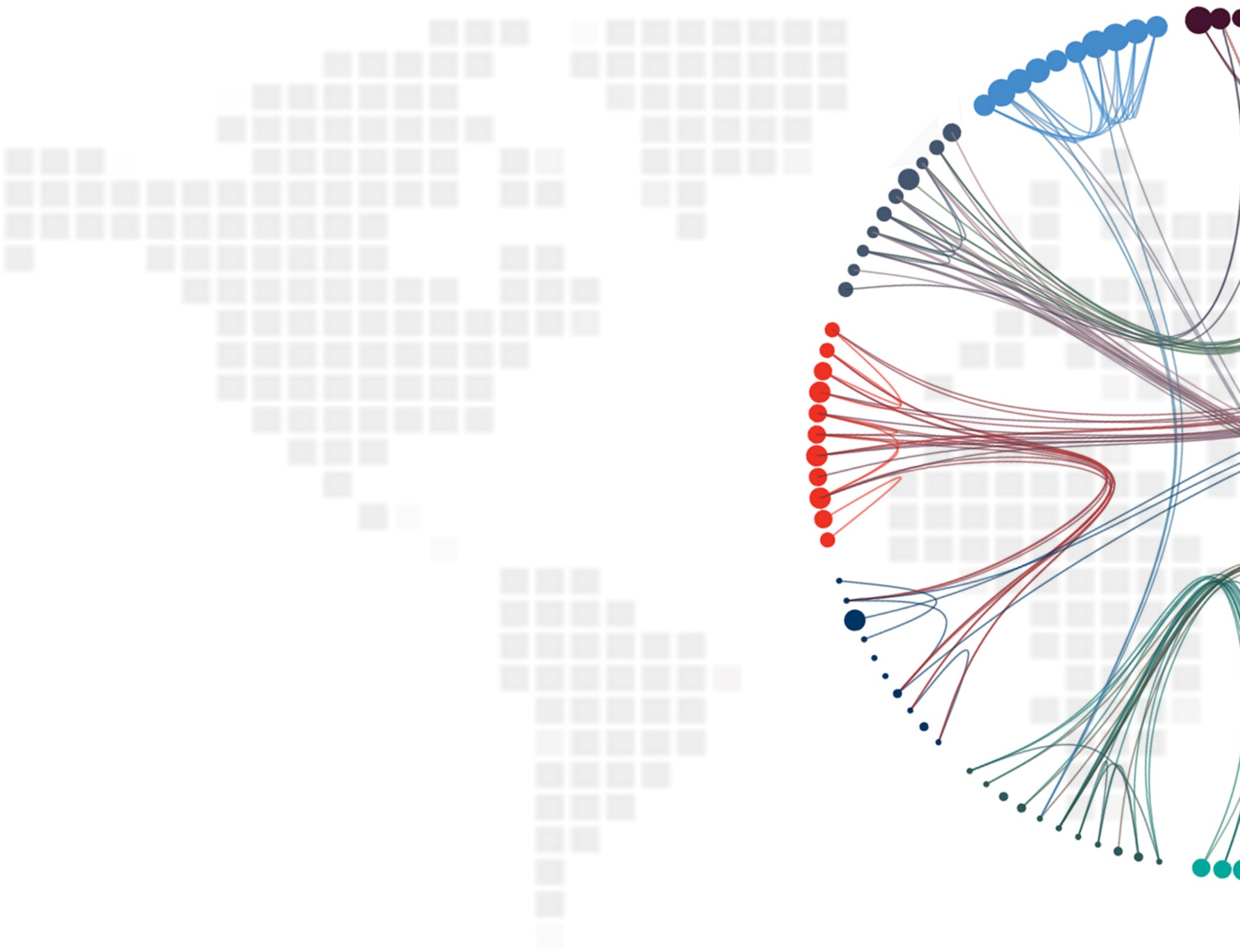


Evaluating Data-centric Protection Solutions

A Guide for Enterprise Architects and CISOs

Henning Horst, Mark Bower

Contents

About this Document

This document is a guide for Enterprise Security Architects, Security Analysts, and CISOs evaluating and comparing tokenization solutions. Tokenization is an architecture model, not just a technology, nor simply an API. Successful tokenization implementations come from evaluating critical areas of concern across data security, architectural compatibility, scale, performance, operation, monitoring, compliance audit and integration. The business value of tokenization is high when it is successful, but as a business critical foundation technology, success will be short lived without thorough assessment up front beyond the commonly evaluated application interfaces and token format policies.

Keywords: data-centric protection, data security, tokenization

Introduction

Enterprise wide data-centric protection has become the preferred strategic way to protect sensitive data in leading enterprises. By ensuring individual data elements are protected in all states across their lifecycle—at rest, in transit, and in use—data can be de-risked from theft and abuse, security and privacy compliance can be simplified, and business agility enhanced. By removing inevitable security and compliance concerns over live data processing, new technologies that underpin innovation, agility, and growth can also be embraced more rapidly. Data-centric security works by using state-of-the art protection methods directly applied at the data elements from capture or creation. Methods include traditional encryption, data tokenization, format preserving encryption (FPE), and masking.

A data-centric model operates on the principle of zero trust, converting sensitive data to a non-sensitive form at all times, and restricting sensitive live data exposure to a small set of readily monitored, managed and defended trusted processes or nodes. The data-centric model inverts the traditional model of protecting the boundary around the data which is increasingly indefensible: the protected data in effect becomes the persistent protection boundary itself, wherever it goes. When data is secured in this fashion, it can flow into low or zero-trust processing environments more freely without risk, including cloud platforms, third party services, machine learning pipelines, file systems, transaction systems, data stores, and data lakes.

Contemporary data tokenization is often favored as it can protect a wide variety of data without constraints or security limitations. It can preserve the meaning, value, and intent of the original sensitive data. With a broad data-centric strategy, such sensitive data can be properly protected and de-identified to neutralize it against data breaches and for streamlining compliance to the regulations like PCI DSS, HIPAA, GDPR, PIPEDA, Privacy Act, or CCPA.

With data being the pervasive lifeblood of every organization, it is critical that any data-centric protection system and various integration points inside the enterprise work 100% of the time. An error or outage of the protection system or integration components in the enterprise could easily bring the business as a whole to a standstill. For this reason, it is very important to ensure that any data-centric protection system is evaluated for both security and critical enterprise capabilities required to deliver a truly agile, yet mission critical tokenization service to the whole organization.

Evaluation Categories

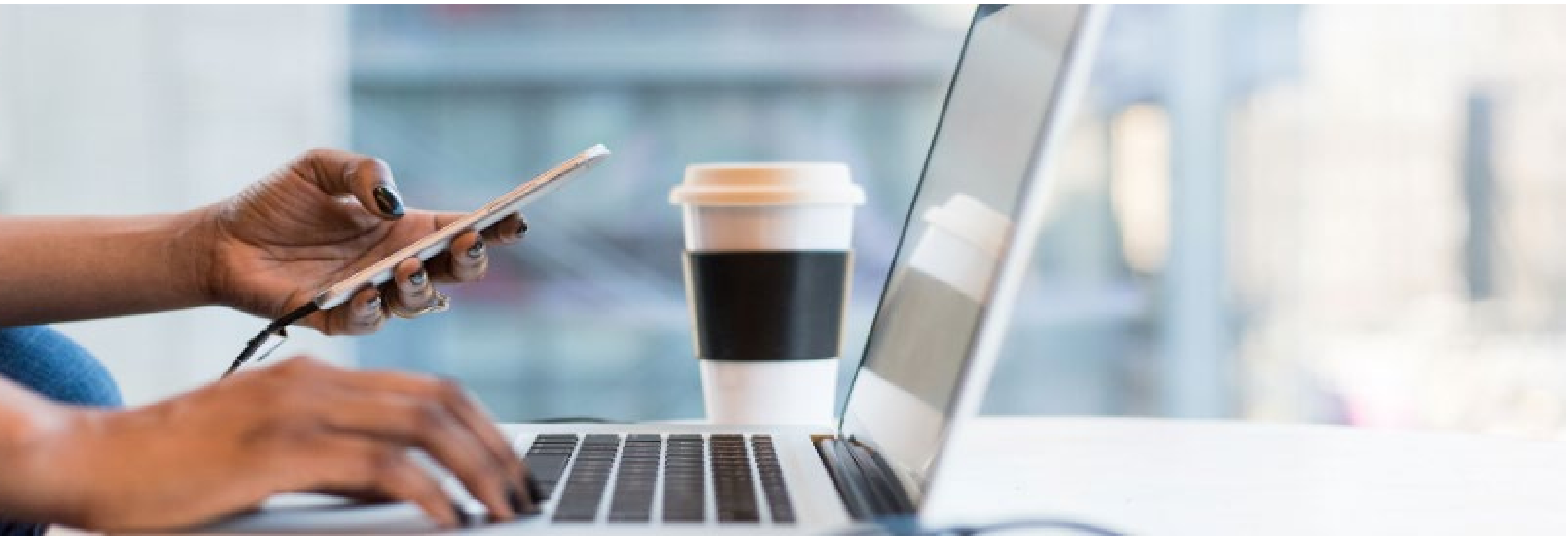
A comprehensive tokenization solution evaluation process covers the following 5 critical categories:

- Security
- Integration vs Configuration
- Architecture, Operations, Service-level Management, and Maintenance
- Critical Success Factors and Common Traps
- Future proofing

Each of these areas are covered as open questions and reasoning behind them in the following sections.

Security

The total security of a data-centric protection solution consists of the security of the algorithms used and supported, the security of the environmental infrastructure, the implementation strategy, and the supported workflows and capabilities enabling secure operation. The following capabilities should be considered when evaluating any data-centric security system:



Do the protection capabilities of the solution provide sufficient security and flexibility for the protection of various data types, length, and formats without known limitations that can lead to data breach notification and regulatory compliance roadblocks?

- Apart from the basic protection capabilities for common data elements like Primary Account Numbers (PANs), and Tax ID's, can data protection of common short or long data elements be performed in a secure and efficient way without regulatory compliance violations?

- Are there specific and published limitations with common data elements, like dates, postal codes or partial fields due to algorithm compromise? For example, NIST has recently warned that 800-38G Standard FPE algorithms as published can only be securely used for data elements over 6 characters long, and recent cryptographic attacks on in-use tokenization approaches have demonstrated practical sensitive data recovery from small samples of tokenized data with more than 90% success.

- Further, are there mechanisms which prevent information leakage based on uneven distribution of data element lengths within a particular set of data to be tokenized? A typical use case where such a mechanism is required for achieving proper security is protection of people's names. For example, very short names such as 'Ng' or 'Doe' can be guessed at a high probability if no additional protection methods like padding, randomization, or tweaking are applied to the protection method by the solution.

- Can the protection methods sustain full and detailed cryptographic review by well-established and truly vendor independent cryptographers without vested interests, patent and other connections to the vendor?

- Have the vendor’s method of data protection ever been identified by cryptanalysis as having a weakness or limitation? If so, what was the specific impact to data, business continuity and compliance

- Which technique to utilize for reversible, format preserving protection, and static table based tokenization or Format Preserving Encryption (FPE)?

How is the overall data-centric protection system secured and isolated?

The most secure algorithm does not protect the overall system if the tokenization data mapping secret or vault process are not secured properly. This could be due to the system not providing sufficient isolation, exposure of authentication credentials, a high attack surface from vulnerable protocols, or lack of advanced security measures and hardening. Any data-centric protection system must:

- Only provide the minimum attack surface necessary (what are the exposed technologies and services? How often have these been subject to vulnerabilities in the past?).

- Never hand out the protection secrets to outside entities, agents or APIs in violation of tokenization isolation principles.

- Provide advanced security features for secure operation, control, and monitoring and to restrict sensitive data exposure during protection and translation operations.

- Elegantly integrate into existing IAM infrastructure with fine-grained access control, audit, and monitoring. This must include options for direct Kerberos support for granular and transparent user-level access to tokenized and detokenized data without cumbersome duplication or syncing of identities from central existing IAM systems, diluting central IAM principles, and best practice.

When evaluating a data-centric protection system, a closer look at the specific security properties will quickly reveal big differences in the security capabilities and concepts implemented in the different data-centric protection solutions. This ranges from a micro-service that encapsulates the protection algorithms and process inside an off-the-shelf docker container, to a highly isolated, Hardware Security Module (HSM) like SoftHSM technology important in cloud deployments, or supporting traditional HSMs.

Integration vs. Configuration

Tokenization must be applied to the point of data capture to have the most value, but implementation costs for first generation solutions can be up to 10x of initial solution cost or more, depending on the capabilities (or lack thereof) of the particular solution. Integration at the earliest point in the data lifecycle requires disintermediation of the data flow – either at the data entry user interface, file capture, on-the-fly, or at the application. Many enterprise applications, particularly in financial processes, operate on vast arrays of file services to share data. Thus change impact on data processing and application code can be massive and disruptive without transparent integration options. This is particularly important to meet CISO goals to prove the value of the investment in short order without major changes, and to resolve data exposure risks quickly and efficiently, for example in a new data-led initiative or after a data compromise.

Another often overlooked but important consideration for a project's overall production delivery effort, cost and agility is its ongoing maintenance. It is important to determine what is required to take the various integration points from an initial functional test towards a fully fault tolerant, scalable and high performance integration.

Strong focus thus needs to be put on how the solution integrates into processes, applications, cloud-native systems, SaaS, and third party systems, in particular:

- Are the solution integration options ubiquitous across the processing environment, including mission critical platforms such as HPE NonStop, IBM z/OS, cloud-native, SaaS, enterprise applications, file systems, data streaming, and analytics/big data systems?

- Does the solution provide transparent integration for file processes and batch processes and include virtual file system technology for ‘tokenized file’ handling automatically by configuration over integration?

- Can the solution integrate without code changes into core processing platforms, for example Base24, Connex, or other complex financial applications?

- Can the solution handle complex payment protocols transparently, like ISO-8583 to tokenize streams of payment data without coding?

- Does the solution offer cloud access security broker capability (CASB) for enterprise web applications or SaaS applications as an option to avoid app code integration?

- Are developers required to invest in building and maintaining a fault tolerant, scalable and high performance integration layer on top of basic APIs in every application utilizing protection services in order to achieve the required availability and performance levels? Or does the solution offer these capabilities out of the box so the application teams can focus purely on its business logic providing direct value to the organization?

- Does the solution support modern micro-service architectures for applications running in modern cloud environments, container workload ecosystems, or private cloud/Kubernetes platforms?

- Does the solution feature capabilities to allow migration from live sensitive fields to tokenized data in a progressive way without a ‘big bang’ integration beforehand? Does the solution enable both tokenized and live data to be present simultaneously without application failure in critical processes that cannot have any downtime?

- Does the solution enable and support a modern and intelligent comprehensive data discovery strategy to identify where sensitive data is being processed and stored, and thus where the optimal integration points for the data-centric protection solution are?

- How are applications changed – does the platform limit agility and strategies for DevOps-centric application delivery?

- How does the solution integrate without code into a modern data streaming or ingestion service, such as Kafka? Does the solution support streaming native tokenization or requires to explicitly micro batch?

- Can the solution support modern languages covering enterprise, data science and machine learning from various languages and frameworks like go, node.js, python, R, Rust, or traditional C/C++, .NET, and Java?

Architecture, Operations, Service-Level Management and Maintenance

Enterprises typically deliver tokenization as a core service within the business with extremely high service levels internally (6 9's or more), and to support aggressive service levels with the enterprises partners and 3rd party data processors. Downtime is not an option, nor is the inability to scale to emerging market requirements on a rapid, agile automated basis. Similarly, traditional tokenization requires a ‘peak load’ architecture that suffers from overprovisioned resources, costs and reserved compute. More modern architectures permit cost and performance scaling dynamically in real time, with robotic and increasingly intelligent automation strategies.

Therefore, another key criterion in the selection process of a data-centric protection solution is how the architecture supports delivering stable, performant, and reliable IT to the entire global business, with key questions being:

- Does the architecture follow modern Infrastructure as Code models or does the solution require human interfaces for configuration? A modern fault-tolerant, cloud-ready architecture will allow process automation, robotic management, and machine readable input configuration and outputs.

- Is the policy mechanism based on a Software Defined approach? Can the configuration integrate with existing IT operations, DevOps, or DevSecOps strategies easily using accepted methods, like YAML?

- Does the solution provide for a full, fault tolerant architecture for continuous operation without downtime?

- Does the solution require additional tooling or complex processes for backup, restore, roll-back, or back out, or is this problem eliminated by the architecture in a fundamental way?

- Are the core components of the solution delivered in a self-contained package, or built on top of general purpose, fully accessible operating system? Who owns the majority of tasks for keeping the core protection system secure, the organization or the solution provider? What is the resulting need for pure personnel cost for ongoing security and general maintenance of the system and the required training to perform those tasks?

During error processing conditions, does the system continue to operate without failure, or does the platform require intervention to reconcile data synchronization issues or API errors?

- What architecture is required for a 99.999% SLA, or 100% uptime?

- How does the architecture scale to 10,000 tps, to 1M tps? 10M tps? How fast can the system scale to a new, arbitrary production scale requirements or data transformation requirement for scaled AI or machine learning data sets, emerging geographic requirements for local regulatory compliant operation, or for dynamic application testing and development needs? Markets change fast, and IT agility must not be the bottleneck to market adaption success. While initial uses today may only require modest performance, leading enterprises demand fast results from large data sets for analytic advantage. Tokenization cannot be a limiter, and must be an enabler of such initiatives within the time-value of the data itself.

- Can the architecture enable edge-computing strategies with sensitive data handling? How?

- Does the core functionality permit protection in future, secure, edge, or embedded computer systems in support of IoT initiatives at a device level, in addition to the back-end analytics level? How?

Critical Success Factors and Common Traps

Besides technology, common areas that organizations struggle with tokenization solutions span areas not covered by technical evaluations. Once implemented, tokenization becomes intertwined with business growth – punitive license models and limited vendor domain knowledge can dramatically limit enterprise trajectory, enhancement delivery, continued innovation, and differentiation.

Another key question in the selection process is - does the solution's economic model, vendor support model, and innovation strategy align or impede success and growth?

- What is the expertise and staffing required to 1) deploy and, more critically 2) integrate and deploy at scale in production? How does the vendor support the success?

- Does the license model scale with success, or impede it? License models that penalize use and consumption force architectural compromise, limit agility, or and require technical workarounds versus maximizing the utility, efficiency and return on tokenization investment. Tokenization must be ubiquitous without license constraints for achieving the success as proven in leading enterprises powered by comforte's tokenization expertise.

- Has the vendor proven an ability to partner and rapidly deliver new capabilities with the agility of the enterprise itself? How is this evidenced?

- Does the vendor have in-house cryptographic expertise with a track record of publication, peer review, and tokenization standards development? Are up-to-date external cryptanalysis efforts in place as a best practice to underpin continued 'ahead of the game' security against new and evolving attacks and exposures? Is this evidenced?

- Does the vendor have deep domain experience, e.g. financial services applications, transaction processing, data models, data flows, and a proven track record?

- Is the vendor already a trusted supplier? What is the track record to date for outages, support resolution, remediation of issues, and response to product enhancements?

Future Proofing

The application ecosystem is in rapid change. In the last few years, the rise of new, converged and hyper-scaled cloud ecosystems has transformed the way code is developed, operated, and delivered to the most agile route possible. Tokenization has emerged over the last decade, but architectures must be forward looking to sustain the pace of change required over the data and application lifecycle. Solutions built to older monolithic non-DevOps models inhibit agility and are incompatible with modern cloud, container, and micro service orchestration architectures. Continuous innovation, investment, and forward looking developments are key criteria to ensure long lasting value of the data-centric protection solution to the organization. Key questions for the selection therefore include:

- How does the vendor support containers, Kubernetes, server-less architectures, and contemporary secure computing technology to keep pace with enterprise demands and ecosystem attack risks?

- Does the vendor have tokenization as a primary business function, or is it now mostly a maintenance stream? What evidence of updates is available from the solution?

- Is the architecture designed for true cloud-native ecosystems, with the "herd" approach to utility? Or, is the solution a legacy product with a "cloud option" bolted on, still taking a monolithic "pet" approach to software delivery?

- Is the vendor's strategy focused on audits and maintenance renewals or innovation and long term customer partnerships? With market consolidation, well-established vendors have found themselves within larger software providers focused less on enhancing customer value, and instead driven by cost cutting and reducing capacity to lead and innovate. Tokenization is a multi-year commitment.

- Does the architecture permit use of emerging enclave technology, to allow distributed and trusted tokenization? Can the tokenization core service operate on least-resource compute models in trusted platforms and achieve acceptable performance and scale?

Conclusion and Recommendations

Decisions on tokenization technology can be daunting. On the surface, many solutions may seem attractive and comparable on the basis of narrow technical capabilities. However, as illustrated, evaluations must look carefully across the different dimension related to total investment and operational compatibility. Lastly, a meaningful proof of concept to contrast against a real business problem should be strongly considered that also looks to the future to ensure long-haul success for all involved.

comforte

Secure Your Growth