Data Protection by Design in Healthcare for Smart Cities: A Hardware-based Pilot using ARM Morello Board

Regis Schuch, Mailson Teles-Borges, Diego Mendes Rodrigues, Sandro Sawicki, Fabricia Roos-Frantz, Rafael Z. Frantz

Unijuí University – Ijuí, RS – Brazil

{regis.schuch, sawicki, frfrantz, rzfrantz}@unijui.edu.br

mailson.borges@sou.unijui.edu.br

Fundação Municipal de Saúde de Santa Rosa (FUMSSAR) – Santa Rosa, RS – Brazil

diegorodrigues@fumssar.com.br

Abstract. *Smart and sustainable cities increasingly depend on integration processes to exchange data across their digital services, such as primary healthcare services. In conventional execution environments, the sensitive data handled by these processes remains exposed in memory, creating opportunities for unauthorised access, data exfiltration, and loss of institutional control over data processing. This paper reports the development and execution of the integration of digital services in a pilot project of a hardware-based approach using the ARM Morello Board. Our solution executes the integration process in a CHERI-based trusted execution environment and uses execution-environment attestation so that digital services can verify execution conditions before authorising them.*

1. Introduction

Smart and sustainable cities increasingly rely on the integration of multiple digital services to enhance citizens' quality of life and optimise public service delivery. The digital government transformation has boosted the use of these digital services that provide essential functionalities, such as urban mobility, health, security, and social assistance [Harrison et al. 2010, Bibri and Krogstie 2017]. These services rely on personal data, which are frequently sensitive and must be shared and processed through integration processes. Such processes are responsible for coordinating a set of digital services and producing new outcomes. However, during the process execution, sensitive data stored in memory may be accessed or exfiltrated by unauthorised entities. This compromises the privacy of the citizens and undermines the reliability of the digital government ecosystem [Hamid and Huda 2025, Gupta et al. 2025].

Such a scenario evidences the need to implement data protection mechanisms *by design*, whereby privacy is supported by the hardware architecture and by the execution infrastructure, rather than relying solely on organisational policies, external controls, or software-based safeguards. Data protection *by design* is particularly relevant in the context of digital government, where regulations such as the General Data Protection Law (LGPD) in Brazil and the General Data Protection Regulation (GDPR) in Europe require the adoption of technical and administrative measures to protect personal data [Brazilian Government 2018, European 2016]. Although these regulations establish

an essential legal framework for accountability and compliance, they do not, by themselves, prevent violations such as unauthorised access or data exfiltration during processing, particularly in scenarios where municipalities rely on third-party infrastructure to host or execute digital integration processes. From a governance perspective, this creates the need for mechanisms that not only define who is allowed to process data, but also provide verifiable evidence about where, how, and under which execution conditions such processing occurs. For this reason, a protection strategy *by design* plays an important role, as it provides technical guarantees that complement the legal and organisational measures required by regulation. In this regulatory and organisational environment, hardware-based security mechanisms – such as Trusted Execution Environments (TEEs) – emerge as a promising means of enforcing confidentiality and integrity guarantees during data processing [Sabt et al. 2015].

TEEs have been proposed as a mechanism to protect code and sensitive data during software execution through hardware-enforced isolation [Sabt et al. 2015]. Major technology companies have introduced hardware-based TEE solutions, including Intel TDX [Intel 2026], AMD SEV [Advanced Micro Devices, Inc. 2020], and, more recently, the ARM Morello Board [ARM 2024]. AMD SEV and Intel TDX are primarily designed to protect virtual machines (VMs) from a potentially malicious or compromised host or hypervisor by encrypting and isolating guest memory. In contrast, the ARM Morello Board implements the Capability Hardware Enhanced RISC Instructions (CHERI) architecture, which introduces a capability-based memory protection model that enforces fine-grained access control and spatial memory safety at the architectural level [Watson et al. 2015].

VM-based TEEs mitigate the risks associated with untrusted infrastructure, such as cloud provider or host compromise, whereas CHERI focuses on preventing memory violations, thus limiting the impact of software vulnerabilities within applications. Rather than competing, these technologies are complementary and can provide layered protection when combined. While VM-based TEEs have been extensively explored in the literature [Zheng et al. 2021, Kaplan et al. 2016] and are widely available in commercial cloud environments, CHERI-based platforms remain experimental and are primarily used for research and evaluation. Nevertheless, the capability-based protection model enables fine-grained memory compartmentalisation, which reduces the attack surface by enforcing strict isolation between memory regions and limiting the impact of memory violations. This capability-based protection model is particularly relevant for integration processes in digital government ecosystems because it helps to prevent sensitive citizen data from becoming exposed in execution time and reduces the risk of external entities access data that should remain under institutional or citizen control.

In this paper, we report our experience in the development of a pilot integration project involving digital services in a smart city environment. The proposed architecture was applied to an integration between a private hospital and primary healthcare units (*Unidades Básicas de Saúde* – UBS) in the city of Santa Rosa, Brazil. This pilot project allowed us to evaluate the use of a hardware-assisted security approach within the software ecosystem of the municipality of Santa Rosa. The novelty of this work lies in the design of an integration process that leverages the CHERI architecture through a server based on the ARM Morello Board. The integration workflow mirrors the one between the private hospital and the UBSs. In this scenario, the integration process handles sensitive

patient data exchanged amongst the digital services. To protect such data during execution, we developed a TEE-based solution deployed on the ARM Morello Board platform. The integration component executes within a hardware-isolated environment, and digital services interact with it through Application Programming Interfaces (APIs) that support secure data exchange. Certificates attest to the integrity of the execution environment and of the deployed integration process. This design adheres to a data protection *by design* principle by restricting data access to applications executing within trusted environments and by leveraging CHERI's capability-based memory isolation to limit the impact of potential software vulnerabilities.

This paper makes the following contributions: (i) the design and implementation of a secure smart city integration process on the ARM Morello Board; (ii) the architectural mapping between data protection *by design* principles and CHERI capability mechanisms; and (iii) the practical validation of TEE-based execution for digital government data sharing and integration. The remainder of this paper is organised as follows: Section 2 describes the healthcare integration scenario; Section 3 illustrates the proposed architecture for implementing the integration process in the CHERI environment; and, finally, Section 4 discusses the results, limitations, and lessons learned.

2. Integration Problem

This section presents the healthcare integration scenario used to evaluate the proposed architecture. The scenario is based on a real integration process between primary healthcare units and a hospital information system in the city of Santa Rosa. In this context, primary healthcare services are managed by the *Fundação Municipal de Saúde de Santa Rosa* (FUMSSAR) [FUMSSAR 2026], a public foundation equivalent to the Municipal Health Secretariat, which maintains the municipal primary healthcare units and related health services. The hospital involved in the pilot is managed by a different organisation. This institutional arrangement reinforces the inter-organisational nature of the integration problem addressed in this work, since patient data are produced, stored, and updated across systems that belong to distinct organisations.

In the regular flow, patients who require hospital assistance must first be attended by physician at a *Unidade Básica de Saúde* (UBS), which is a primary healthcare unit responsible for initial medical assessment. During this appointment, the UBS staff collect the patient's personal data, such as name, identification document, and residential address. They also record clinical data, including symptoms, preliminary diagnosis, and the prescribed procedures. All these data are stored in the UBS software system. When the patient requires specialised treatment or further examination, they are referred to a hospital. However, the hospital software system does not have direct access to the data stored in the UBS system. As a result, the patient must carry physical documents containing the previously recorded data. Upon arrival at the hospital the medical staff manually re-enter these data into the hospital information system. This manual transcription process is inefficient and prone to errors, potentially leading to incomplete records, insistences, and delays in patient care. This scenario highlights the need for coordinated interaction and data exchange between digital services to support services delivered to citizens.

To address this problem, we developed an integration process capable of exchanging patient data between the UBS and the hospital information systems. This process supports bidirectional communication: it transfers patient data from the UBS to the hos-

pital when the patient is referred for specialised care, and it also allows the hospital to update the UBS with data generated during treatment, since patient may require future assistance at the UBS. Figure 1 illustrates the integration workflow between these services. When the patients arrives at the hospital, the hospital staff use the patient's documents to search for their records in the `Hospital Service`. If the patient data is outdated or unavailable, the `Hospital Service` requests the missing data from the `Integration Process`. The `Integration Process` then queries the `UBS Service` to retrieve the required data. After receiving the response from the `UBS Service`, the `Integration Process` updates the `Hospital Service` with the retrieved data, allowing the hospital staff to continue the patient's treatment.

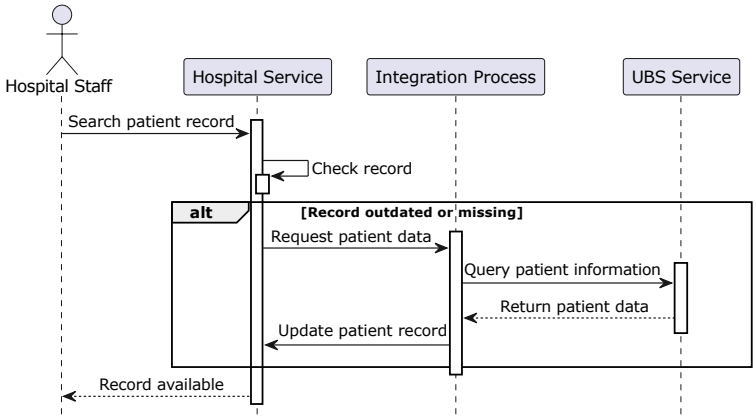

Figure 1. Conceptual Representation of the Integration of Digital Services.

It is important to note that the data exchanged between the digital services and the integration process are transmitted over the network and may be protected by well-established technologies, such as HTTPS encryption or Virtual Private Networks (VPNs), which ensure confidentiality and integrity during communication. However, the implementation and execution of integration processes are often outsourced to third-party companies responsible for enabling interoperability between systems maintained by different organisations, which motivates the use of hardware-assisted isolation mechanisms capable of protecting sensitive data during processing, not only during transmission.

3. Integration Solution

This section presents the architecture used to implement the integration process using CHERI-based trusted execution environments. The proposed architecture enables digital services from the healthcare integration scenario to interact with an integration process deployed within a trusted execution environment.

3.1. Architectural Proposal

The healthcare integration scenario is organised into two clearly defined domains, as illustrated in Figure 2. Figure 2 (a) represents the `Digital Government Infrastructure`, where the digital services operate in a conventional execution environment. Each digital service maintains a local repository and exposes APIs for secure communication. Requests sent by the integration process to the digital services are accompanied by an attestation certificate, `signedCert`, containing attributes of the execution environment. Therefore, each digital service implements a `verifyCertificate()`

operation to validate incoming requests based on the certificate associated with the integration process. In turn, Figure 2 (b) represents the integration process running inside a `Trusted Execution Environment` deployed on an ARM Morello Board. In this environment, we introduce a component called `Launcher`. The `Launcher` acts as a mediator between the digital services and the integration process. Because the integration process binaries must be compiled for the CHERI architecture, the `Launcher` encapsulates the steps required to compile the integration process source code for this environment. Furthermore, the `Launcher` generates certificates containing verifiable attributes, such as hardware properties, the operating system version, and the cryptographic identification of the installed binary. These certificates are signed by a Certificate Authority and attached to requests sent to the digital services. In organisational terms, they provide technical evidence for governance and compliance, since they allow digital services to verify whether the integration process is executing in an expected and authorised environment before exchanging sensitive data.

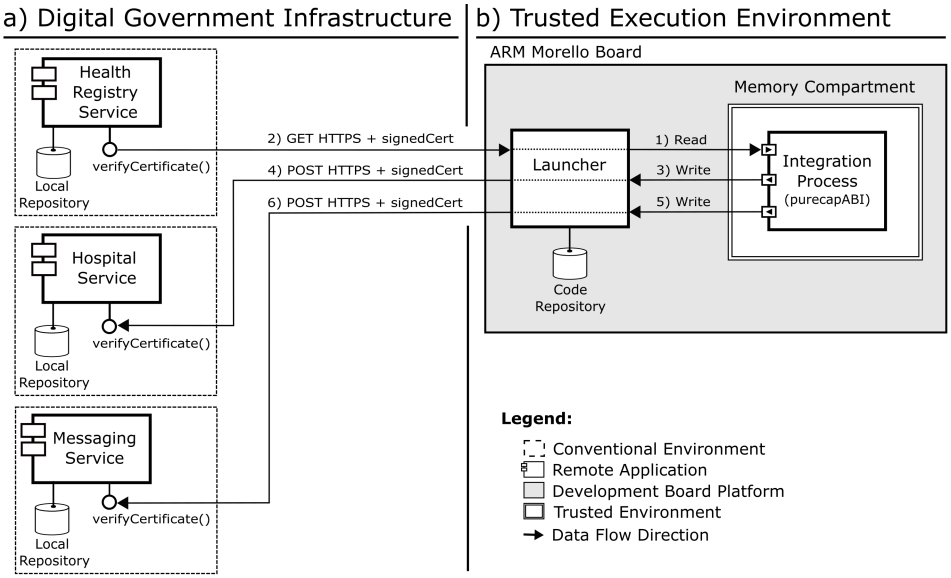

Figure 2. Digital service interaction with the CHERI-based integration process.

3.2. Binary Generation and Execution

This section describes the environment used to compile and run the `Integration Process` within the CHERI architecture supported by the ARM Morello Board. In this environment, compilation and execution determine how capability-based memory protection mechanisms are enforced [Watson et al. 2015].

CHERI supports two compilation modes: hybrid Application Binary Interface (ABI), in which developers annotate capability pointers with `__capability`, and pure-capability ABI (PureCap ABI), in which all pointers are represented as capabilities by default. A capability extends a conventional pointer with bounds and access-permission metadata, allowing hardware to constrain memory accesses. For this pilot, we adopted the PureCap ABI because it enables the integration process to benefit from CHERI's protection model with limited source-code changes. The process runs on CheriBSD, a FreeBSD-derived operating system adapted to support CHERI mechanisms [Watson et al. 2015].

The integration process is compiled from C source code using the `clang-morello` compiler, which provides support for CHERI extensions [Watson et al. 2026]. Because CHERI introduces capability-aware instructions and pointer representations, applications must be compiled with a CHERI-enabled toolchain to generate compatible binaries. Listing 1 presents the commands used to compile and execute the integration process. The first command generates a binary for the Morello capability model using the PureCap ABI, while the second launches the program with CHERI C runtime support enabled. Consequently, the integration process runs with hardware-enforced capability semantics provided by the CHERI architecture.

```
(a) Compilation
$ clang-morello -march=morello+c64 -mabi=purecap -o integration_process
    ↪ integration_process.c

(b) Execution
$ proccontrol -m cheric18n -s enable ./integration_process
```

Listing 1. Commands for compiling and executing the integration process.

The deployment of the integration process was performed on an ARM Morello Board hosted at TODAQ[1], Toronto, Canada, in the context of the Cloud Attestables on Morello Boards (CAMB) project[2]. The server was remotely accessed via SSH. The integration process code is publicly available on GitHub[3].

4. Discussion, Limitations, and Lessons Learned

In this paper, we presented the implementation of a pilot integration process developed to address the need to interconnect systems used in the public healthcare system of the municipality of Santa Rosa. In the observed scenario, integration processes between public systems are frequently developed or operated by third parties, or executed on infrastructures that do not belong to the public institutions themselves. This dependency may represent risks to citizens' privacy and data protection, particularly when sensitive data pass through environments that are not under the direct control of the organisations responsible for the public service. Although data protection legislation has been established in several countries, such instruments alone do not enforce data security. In inter-organisational settings, compliance also depends on governance arrangements that define responsibilities for operating protected execution environments, authorising data exchanges, auditing execution evidence, and responding to failures or suspected misuse. In this context, approaches based on data protection *by design* emerge as a paradigm that seeks to incorporate protection mechanisms directly into the conception and implementation of hardware-based systems. Unlike approaches that rely predominantly on policies or external controls, this paradigm aims to ensure data protection through the system's technical properties, including the software, the execution environment, and the underlying hardware architecture.

The pilot developed in this work explored the implementation of the integration process in an environment based on secure execution technologies, using memory protection mechanisms provided by the CHERI architecture. Our experience conducting this

[1]`https://engineering.todaq.net/`
[2]`https://www.cl.cam.ac.uk/research/srg/projects/camb/`
[3]`https://github.com/gca-research-group/lasdigov-2026`

pilot indicates that the proposed solution is technically feasible and can be applied in real-world system integration scenarios. Furthermore, although CHERI is still an experimental technology, other secure hardware approaches and TEEs exist that can support solution models aimed at data protection. Based on these technical and organisational considerations, the following lessons learned also summarise the main limitations, threats to validity, and mitigation actions observed during the pilot:

- **L1: The role of hardware in ensuring security and privacy:** the use of the CHERI architecture increased the granularity in controlling memory access during the execution of the integration process, allowing the system to enforce stronger isolation amongst components that manipulate sensitive data.
- **L2: Legacy software can be adapted with limited modifications:** porting existing software components to capability-based environments proved feasible. Execution modes such as PureCap ABI enable existing source code to be reused with relatively limited changes. This suggests that integrating hardware-based protection mechanisms does not require rewriting the entire integration process, which may facilitate gradual adoption.
- **L3: Data volume as a limitation to performance validity:** although the pilot project demonstrated the technical feasibility of the integration process, it did not evaluate the synchronisation of large data volumes. This limitation is important in healthcare scenarios involving high-volume clinical artefacts, such as medical images. To mitigate overgeneralisation, our conclusions are restricted to the implemented integration workflow, and future evaluations should consider larger datasets and different clinical data types.
- **L4: Technological maturity as a threat to replication and adoption:** despite the observed advantages, one of the identified challenges concerns the maturity of the technologies involved. In the specific case of CHERI, commercial availability is still limited, the existing documentation remains restricted, and the tooling ecosystem is still in its early stages.
- **L5: Professional training as an organisational limitation:** regarding data protection *by design*, professional training may represent a limiting factor. Secure execution technologies and security-oriented hardware architectures are still rarely addressed in undergraduate computer science programmes, hindering the dissemination of this development model and increasing the effort required to train technical teams. This threat can be mitigated through documentation, training materials, and gradual adoption strategies involving developers, infrastructure teams, and public-sector stakeholders.

Acknowledgment

This research was partially funded by the Co-ordination for the Brazilian Improvement of Higher Education Personnel (CAPES) and the Brazilian National Council for Scientific and Technological Development (CNPq), under grants 309425/2023-9, 402915/2023-2, and 311011/2022-5. We thank Dr Carlos Molina-Jiménez, Computer Lab, University of Cambridge, for valuable discussions; Erick Derohanian for providing and maintaining the ARM Morello Board operational environment at Todaq, Toronto; and Kananda Raquel Mayer and Luis Henrique dos Santos Carvalho, from IPM Sistemas, Brazil.

References

Advanced Micro Devices, Inc. (2020). SEV-SNP: Strengthening VM Isolation with Integrity Protection and More. Technical Report 56860, Advanced Micro Devices.

ARM (2024). ARM Morello Program. `https://arm.com/architecture/cpu/morello`. Acc: 2026-01-05.

Bibri, S. E. and Krogstie, J. (2017). Smart sustainable cities of the future: An extensive interdisciplinary literature review. *Sustainable Cities and Society*, 31:183–212.

Brazilian Government (2018). General Data Protection Law (LGPD). `https://planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm`. Acc: 2026-01-05.

European (2016). General Data Protection Regulation (GDPR). `https://gdpr-info.eu/`. Acc: 2025-09-12.

FUMSSAR (2026). FUMSSAR – Fundação Municipal de Saúde de Santa Rosa. `https://www.fumssar.com.br/`. Acc: 2026-03-11.

Gupta, P., Hooda, A., Jeyaraj, A., Seddon, J. J., and Dwivedi, Y. K. (2025). Trust, risk, privacy and security in e-government use: insights from a MASEM analysis. *Information Systems Frontiers*, 27(3):1089–1105.

Hamid, S. and Huda, M. N. (2025). Mapping the landscape of government data breaches: A bibliometric analysis of literature from 2006 to 2023. *Social Sciences & Humanities Open*, 11:101234.

Harrison, C., Eckman, B., Hamilton, R., Hartswick, P., Kalagnanam, J., Paraszczak, J., and Williams, P. (2010). Foundations for Smarter Cities. *IBM Journal of Research and Development*, 54(4):1–16.

Intel (2026). Intel Trust Domain Extensions (TDX). `https://intel.com/content/www/us/en/products/docs/accelerator-engines/trust-domain-extensions.html`. Acc: 2026-03-05.

Kaplan, D., Powell, J., and Woller, T. (2016). AMD Memory Encryption. In *International Conference on Hardware/Software Codesign and System Synthesis*, pages 1–6.

Sabt, M., Achemlal, M., and Bouabdallah, A. (2015). Trusted Execution Environment: what it is, and what it is not. In *IEEE Trustcom/BigDataSE/Ispa*, volume 1, pages 57–64.

Watson, R. N. M., Clarke, J., Davis, B., Witaszczyk, K., and Neville-Neil, G. V. (2026). Getting Started with CheriBSD 25.03: User level software compartmentalization (experimental). `https://ctsrd-cheri.github.io/cheribsd-getting-started/features/c18n.html`. Acc: 2026-03-05.

Watson, R. N. M., Woodruff, J., Neumann, P. G., Moore, S. W., Anderson, J., Chisnall, D., Dave, N., Davis, B., Gudka, K., Laurie, B., Murdoch, S. J., Norton, R., Roe, M., Son, S., and Vadera, M. (2015). CHERI: A Hybrid Capability-System Architecture for Scalable Software Compartmentalization. In *IEEE Symposium on Security and Privacy*, pages 20–37.

Zheng, W., Wu, Y., Wu, X., Feng, C., Sui, Y., Lu, X., and Zhou, Y. (2021). A Survey of Intel SGX and Its Applications. *Frontiers of Computer Science*, 15(3):153808.