White Paper

Confidential Computing

Hardware and Software Components of Confidential Computing Architectures

Confidential computing protects data while in use, using a hardware and software stack based on a CPU root of trust to enable more secure enclaves that isolate sensitive data and code. Fortanix Runtime Encryption provides confidential computing services based on Intel® Software Guard Extensions (Intel® SGX) that enable applications to take advantage of more secure enclaves without requiring any changes to code.

Table of Contents

1 Executive Summary

Protecting data while it is in use and held in active system memory has always been more challenging than protecting it while in storage or transit. The act of decrypting data to access and perform operations on it creates potential security exposure from the presence of other processes running in the same shared memory space, including those related to system software compromise or insider threats. Moreover, in a cloud-native world, edge and IoT assets exist outside any security perimeter, and potential attackers may even have physical access to them. Such exposures are of particular concern with regard to the privacy of financial or healthcare information.

> Organizations that handle sensitive data such as Personally Identifiable Information (PII), financial data, or health information need to mitigate threats that target the confidentiality and integrity of either the application or the data in system memory.
>
> – Confidential Computing Consortium[1]

Confidential computing is an approach to protect data while in use by providing a trusted execution environment (TEE) where trusted code can operate on trusted data in isolation from unauthorized entities. The TEE protects the data and code as well as the integrity of the outcomes of operations performed on it. Because a TEE must remain safe from unauthorized access even from highly privileged system software or human operators with root access, it requires a low-level hardware root of trust. Anchoring the trusted computing environment in low-level hardware eliminates underlying dependencies and their accompanying attack surfaces.

Authors

Fortanix: Jattin Dudakia, Pawan Khandavilli, David Greene
Intel: Kapil Sood, Raghu K. Moorthy, Vinodh Raghunathan

Intel® SGX implements TEE functionality by providing more *secure enclaves*, which are dedicated private memory address spaces, protected from external access even if the compute platform is compromised. This silicon-based functionality, available on the 3rd generation Intel Xeon processor, is controlled by a processor instruction set. Data and code held within an enclave are protected by a hardware root of trust against being read or written from outside the enclave, regardless of the outside entity's privilege level. Intel SGX requires that application code be divided into trusted and untrusted portions, with trusted portions operating in more secure enclaves. Such re-architecting of software can be a blocking factor for some organizations.

Fortanix Runtime Encryption is a set of software technologies that streamline implementation of Intel SGX, overcoming the need to alter code by allowing unmodified applications to run in more secure enclaves. Fortanix Enclave OS is a runtime for code inside enclaves, operating a CPU root of trust. Fortanix Confidential Computing Manager is a cloud-native service for managing and orchestrating enclaves, including policy enforcement and secure attestation. The Fortanix Enclave Development Platform (EDP) is an open source development environment for creating Intel SGX enclaves in the Rust programming language, with built-in code-safety tools and a dedicated compiler.

This white paper reviews each of these elements and how they form the basis of architectures to protect data privacy while it is in use, even on unsecured or compromised platforms. This ability is increasingly critical as cloud computing models offer increasing benefits from data sharing to power large-scale analytics, machine learning, and other data-driven applications. The discussion also introduces a number of use cases to illustrate the value of confidential computing in real-world deployments and introduces a new deployment mechanism for Fortanix Runtime Encryption Technology on container topologies based on Red Hat® OpenShift® Container Platform.

2 Intel® SGX: The Hardware Foundation for Confidential Computing

Within hierarchical system security, workloads inherit threat exposures from more privileged layers further down the stack. An application and its data are vulnerable to processes running with higher privileges, such as those of the OS, hypervisor, pre-boot partitions, or firmware. For example, if the OS is compromised, the applications that run on it are necessarily compromised as well, effectively extending the application's attack surface to the entire OS, as shown in the "Without Intel SGX" pane of Figure 1.

2.1 Application Architecture for Intel SGX

Intel SGX stores an application's secrets—such as encryption keys, passwords, and financial or healthcare data—within more secure enclaves. All operations that require access to that information must be run in the enclave's restricted memory address space as well. The contents of an enclave are never exposed outside the enclave to any user, process, or application, regardless of privilege level. Memory within the enclave is encrypted using hardware secret keys that are inaccessible to software. This architecture limits a secret's trust boundary to the relevant Intel SGX enclave itself, which dramatically reduces the attack surface, as shown in the "With Intel SGX" pane of Figure 1.

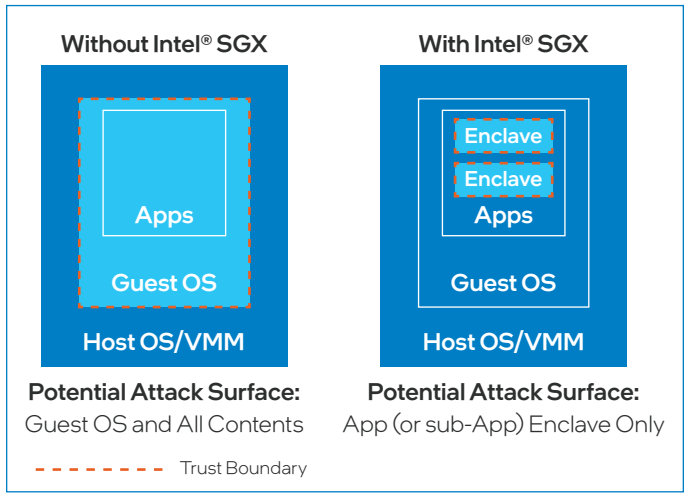

Figure 1. Reduced trust boundary and attack surface with Intel® SGX.

The code that operates within an enclave is referred to as an application's "trusted component," which has access to the application secrets within the corresponding enclave. (One application may have multiple enclaves, with a trusted code component for each.) The remaining parts of the application and all its dependencies make up the "untrusted component," which do not have access to enclave contents. Interactions between trusted and untrusted components are accomplished using Intel SGX instructions, as shown in Figure 2.

From the viewpoint of code or data inside an enclave, everything outside is regarded as untrusted, which includes system software, BIOS, firmware, etc. Developers can use the Intel SGX SDK to define trusted and untrusted components of applications and create enclaves for trusted code and data. This implementation designates enclaves as shared libraries, which can be called by the untrusted parts of the application. Best practices call for interactions between trusted and untrusted components of the application to be as limited as possible.

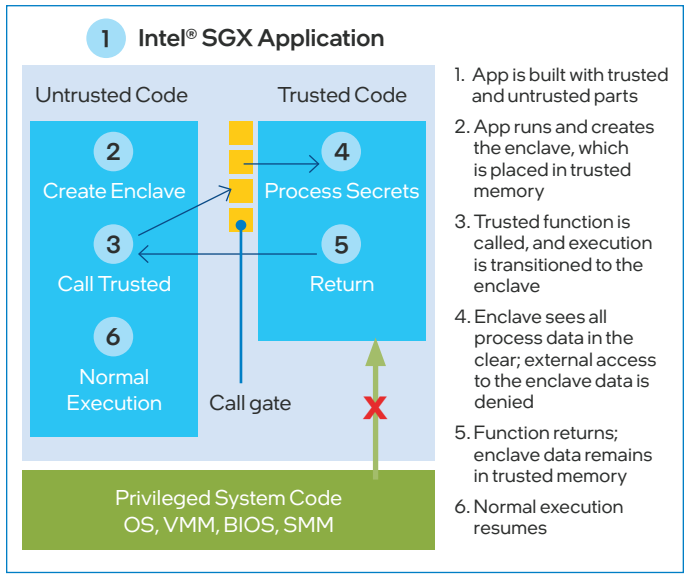

Figure 2. Intel® SGX interaction between trusted and untrusted application components.

2.2 Sealing for Data Persistence

Because enclaves function by means of shared function calls, the internal state of the enclave is not persisted after the host application unloads the library. To satisfy cases where developers need to persist data that is internal to enclave processes beyond the point where the enclave is loaded into system memory, Intel SGX provides a mechanism called sealing that allows this secret data to be stored more securely as an encrypted blob on untrusted media, secured by a CPU-generated key. The key can be tied to the enclave's fingerprint, making the data only available to a specific version of a specific enclave, or it can be tied to the enclave's sealing authority, in which case multiple enclaves from that authority can seal and unseal each other's data.

2.3 Attestation for Protected Enclave Interactions

To support cases where enclaves need to communicate or interact with one another, Intel SGX provides attestation services that enable those enclaves to each cryptographically verify the trusted status of the other's secured execution environment. Those assurances include the following factors:

- **The code is running as-built** in a genuine enclave
- **The hardware is a more secure Intel SGX-capable platform** with all needed microcode updates applied
- **All necessary Intel SGX hardware and software configurations** are made correctly

The enclaves involved in attestation may or may not be hosted on the same platform. "Local attestation" refers to interactions between enclaves on the same platform, so multiple enclaves in a single application can work together on common tasks or so separate applications can communicate data between enclaves. "Remote attestation" refers to similar verification for enclaves on separate hosts, as in the case where client and server applications must prove their integrity to one another. Both local and remote attestation enable trusted communication with integrity and confidentiality assurances to occur over untrusted channels.

3 Fortanix Runtime Encryption Technology: Enablement for Intel SGX

To enable enterprises to protect data while it is in use, Fortanix Runtime Encryption implements deterministic security such that computation can be carried out on encrypted data without ever exposing it outside enclaves in the clear. The platform supports application binaries in their existing forms, without code modifications, so that time to benefit is accelerated, and no burden or learning curve is imposed on developers. It also integrates easily with existing orchestration tools and development workflows.

The components of Fortanix Runtime Encryption Technology—Enclave OS, Confidential Computing Manager, and Enclave Development Platform—are summarized in Figure 3, and each is discussed in more detail in the remainder of this section.

Platform Capabilities that Complement Intel® SGX

3rd Gen Intel® Xeon® Scalable processors incorporate multiple hardware-resident security features that work in conjunction with Intel® SGX. The following features are of particular interest to CoSPs as they deploy 5G network functions:

- Built-in crypto acceleration. To help CoSPs handle the performance impact of pervasive encryption in 5G, platform results include up to 4.2x higher TLS-encrypted connections per second.[2]
- Intel® Platform Resilience. To protect fundamental platform firmware components, this Intel® FPGA-based solution establishes a chain of trust and verifies firmware images before execution.

Fortanix® ENCLAVE OS

Runtime for code inside Intel® SGX enclaves:

- No changes to application binaries required
- Operates a CPU-based root of trust

Fortanix® Confidential Computing Manager

Manages enclaves and confidential computing nodes:

- Single pane of glass; cloud-native SaaS
- Provides policy enforcement and attestation

Fortanix® EDP

Environment for creating enclaves from scratch:

- Open source; based on Rust programming language
- High performance and built-in security measures

Figure 3. Fortanix Runtime Encryption Technology.

3.1 Enclave OS: The Runtime Inside Enclaves

Fortanix Enclave OS provides runtime functionality for code inside enclaves so that applications can run unmodified with the confidential computing benefits of Intel SGX. In preparation for deployment, the application, its dependencies such as libraries or managed runtimes, and Enclave OS are packaged together. The CPU itself derives the encryption key for this memory space on the fly using secrets provisioned in silicon without involvement from or exposure to privileged system software. Enclave OS provides encryption mechanisms for data written outside of the enclave:

- **Data written to system memory** is protected by an encryption key generated at boot time that is usable only by hardware-based processes.
- **Data written to storage** is encrypted with a data seal key derived at runtime using a hardware secret, the identity of the application, and the identity of the signer.
- **Data written to the network** is protected by encryption keys (often TLS) generated by the application, which are isolated within enclaves while in use and encrypted using a data seal key while in storage.

3.2 Confidential Computing Manager: Enclave Management, Orchestration, and Attestation

Fortanix Confidential Computing Manager is a single pane of glass to control the overall enclave lifecycle, including creation, deployment, monitoring, and auditing. Teams use it to provision, orchestrate, and manage systems and applications for confidential computing based on Intel SGX, including to create and manage the trust relationships between them. As a cloud-native SaaS platform, Confidential Computing Manager provides future-ready support for emerging enterprise infrastructures and topologies. It also manages all attestation activities for the Fortanix Runtime Encryption platform.

Each compute node sends a one-time attestation of its status as a genuine Intel SGX-capable system to Confidential Computing Manager for purposes of enrollment as an Application Node for enclave applications. After verifying that status with the Intel Attestation Service, Confidential Computing Manager provisions a secret with the Fortanix Quoting and Provisioning Enclave on the compute node that enables it to attest to its genuine status on its own. Application Nodes may consist of any hardware that supports Intel SGX, whether located on-prem, in a hosted environment, or in a public cloud. Confidential Computing Manager also provides policy enforcement, such as whitelisting containers that have been modified to operate with Fortanix Runtime Encryption and geofencing them to govern where they can execute, to comply with regulations such as the EU General Data Protection Regulation (GDPR).

3.3 Enclave Development Platform: Enclaves from Scratch

Fortanix Enclave Development Platform (EDP) is an open-source environment specifically for writing Intel SGX enclaves from scratch using the Rust programming language. Product engineering teams at Fortanix originally developed the platform for internal use, to simplify access to Intel SGX features and functionality. The EDP makes it easy for developers to enable software for confidential computing.

Developers code using standard techniques, without needing to partition applications into trusted and untrusted components. They simply compile for Intel SGX using the built-in Rust compiler, which also provides advanced static code analysis to automatically help improve software security. Rust provides high compute performance, as well as portability to build code once and then run it across a range of OSs. The Rust project also makes assurances that existing code will continue to be compatible with all future compiler updates.

4 Confidential Computing Use Cases: Finance and Healthcare

4.1 Use Case 1: Money Laundering Detection

The United Nations estimates that two to five percent of the global GDP is processed each year by money laundering operations to disguise illicit sources.[3] Money laundering is a critical process for criminal parties to be able to make use of their illegal profits, with much of this money generated through illicit drugs, weapons, and human exploitation. Denying such resources to criminals is of particular importance, as money can be funneled into potential uses such as funding terrorism or financing the proliferation of nuclear, chemical, or biological weapons.

Fortanix® Confidential Computing technology and Intel® SGX lie at the heart of the federated learning arrangement illustrated in Figure 4, where encrypted customer account and network telemetry data from multiple banks is aggregated in a more secure enclave. That collective data set supports far more sophisticated analytics than any individual bank's data alone, enabling detection of transactions and patterns that signal money laundering. The larger data set also helps accelerate learning for deep learning models used in detection. The Confidential Computing arrangement represented in Figure 4 enables these calculations to be performed without exposing sensitive information, with auditable privacy protections throughout the data lifecycle.

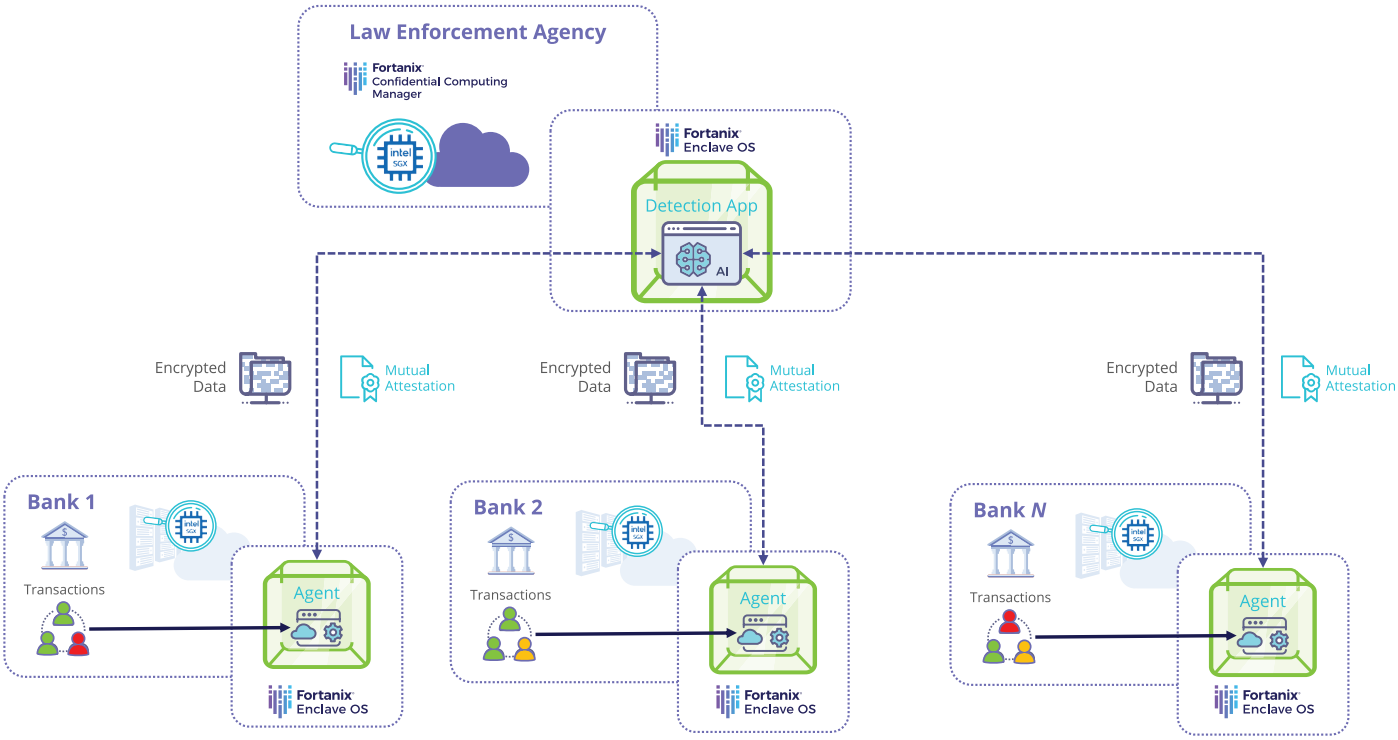

Figure 4. Federated learning to detect money laundering.

4.2 Use Case 2: Electronic Health Records Implementation

The German Social Security Code (§ 291a SGB V) sets out standards for what health data must be stored in electronic health records (eHR). The dataset includes "data on findings, diagnoses, therapeutic measures, treatment reports, and vaccinations for cross-case and multi-patient documentation about the patient" for some 75 million individuals with private healthcare insurance. As the controllers of their own health data, patients must have confidence in the data's safety as well as the ability to control access to it by medical personnel and others.

The eHR implementation, represented in Figure 5, deploys Fortanix® Confidential Computing technology on the IBM cloud to provide security enclaves using Intel® SGX for isolated machine learning-based processing of personal heath data. Because the data is processed in unencrypted form only within more secure enclaves, it remains unavailable to unauthorized parties, even those with root access to the server the data is being processed on. Data is accessible only by means of more secure enclaves, with data access subject to approval by the patient. The subsequent e-prescription project in Germany specifies the use of TEE architecture for processing about 800 million transactions per year, demonstrating the scalability of this type of privacy preserving solution.

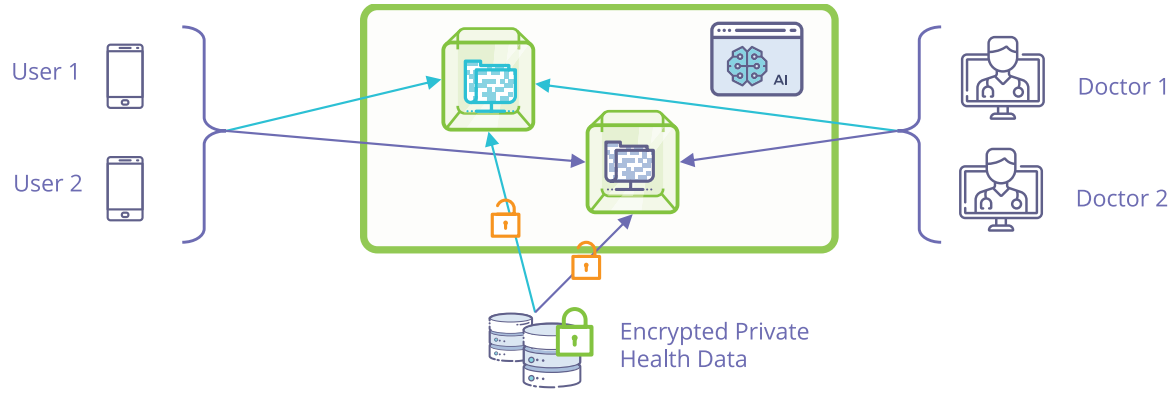

Figure 5. Protected access to electronic health records.

4.3 Use Case 3: Protected Real-World Evidence for Clinical Research

The ability to use real-world evidence—as distinguished from data collected in controlled trials—provides cost-efficient access to large data sets for clinical research. Examples could include real-time analysis of epidemiological or clinical data, reviews of medical and surgical outcomes, or investigations of potential new uses for approved medications. For these usages, privacy regulations require the data itself to be cloaked from the researchers, even as they make secure queries against data sourced from large numbers of healthcare data providers.

The topology illustrated in Figure 6 implements Fortanix® Confidential Computing Manager™ to support access and querying within Intel® SGX secure enclaves against native personal health data housed in eHR systems. Patient data is protected using integrated policy controls at the federal, state, and local levels, and enclave-protected encryption keys protect audit logging that assists with regulatory compliance. The ability to use real-world evidence in a protected framework could enable research that would otherwise be impossible.

4.4 Use Case 4: Chest X-Ray Interpretation

Chest X-ray images provide vital information for early detection and treatment of pneumonia and other respiratory diseases. Technological advances are allowing for improvements in diagnostic capabilities that can assist caregivers with information about effects on different regions of the chest. Deep learning models are adept at analysis across large numbers of radiological images, subject to privacy requirements for the personal health data, and secure training of those models benefits from access to the largest amount of data possible.

Fortanix® Confidential Computing Manager™ enables protected programmatic access to chest X-ray images within an Intel® SGX secure enclave, as shown in Figure 7. A convolutional neural network (CNN) classifies the images according to their indications for pneumonia, other conditions, or no findings. Because the neural network can easily be deployed from the cloud, the model can also help compensate for shortages of radiological specialists and computing infrastructure, especially in remote areas.

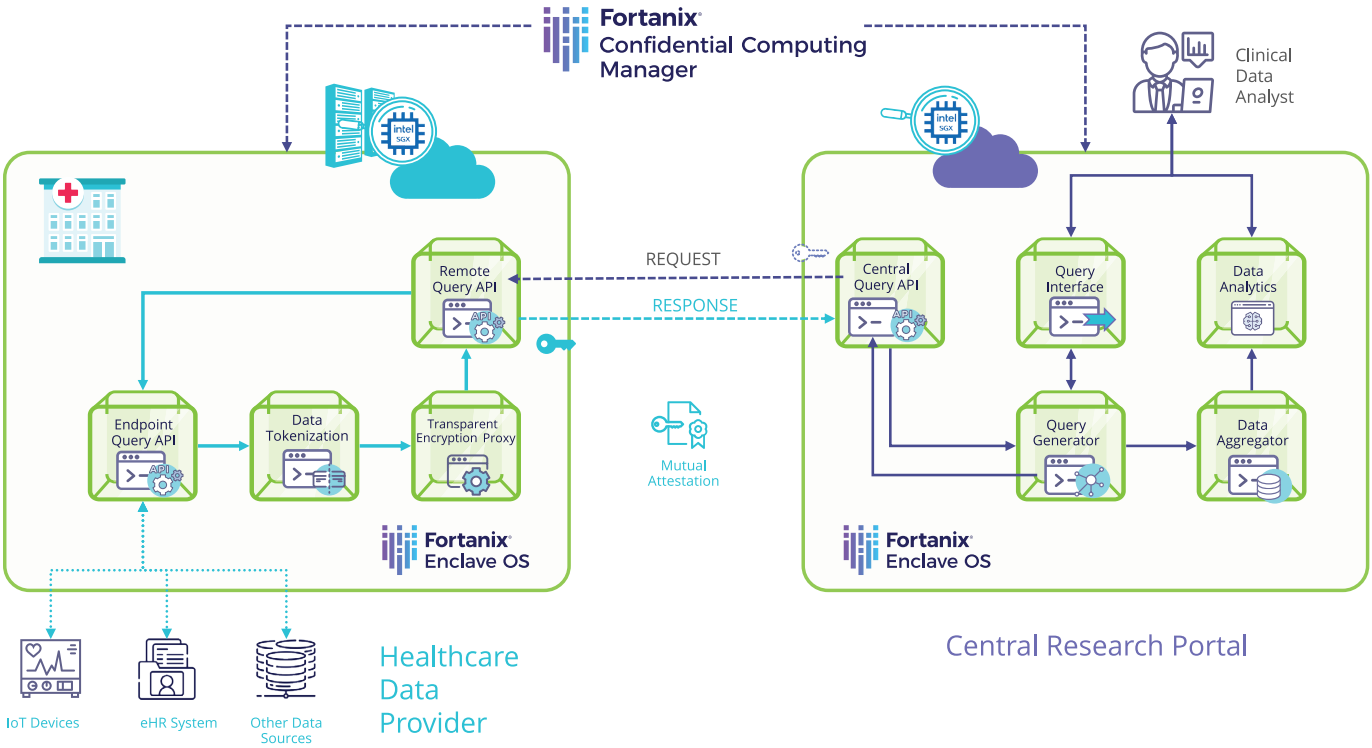

Figure 6. Research and analysis with real-world clinical data.

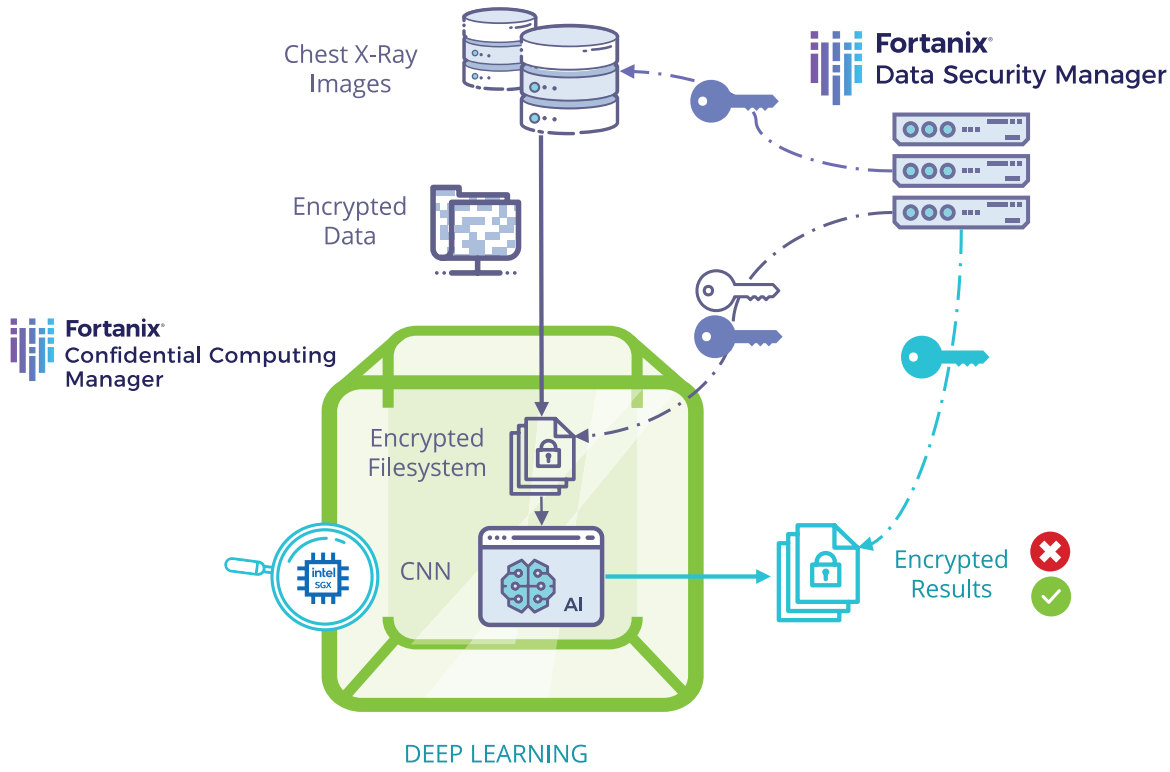

Figure 7. Automated, more secure diagnosis using chest X-ray images.

5 Fortanix Integration with Red Hat OpenShift

Fortanix Node Agent is a software element that is deployed on compute nodes to enable confidential computing, including allowing those nodes to register with Fortanix Confidential Computing Manager. The node agent assists with verification of compute-node hardware and system software, making it instrumental in setting up trusted compute pools. It also enables management of nodes and applications running in more secure enclaves. To enable integration between Red Hat OpenShift and Fortanix Runtime Encryption, Fortanix engineers have implemented the Node Agent as a Red Hat OpenShift Operator for use across on-premises, hosted, and public cloud compute nodes based on Intel SGX-capable systems, as illustrated in Figure 8.

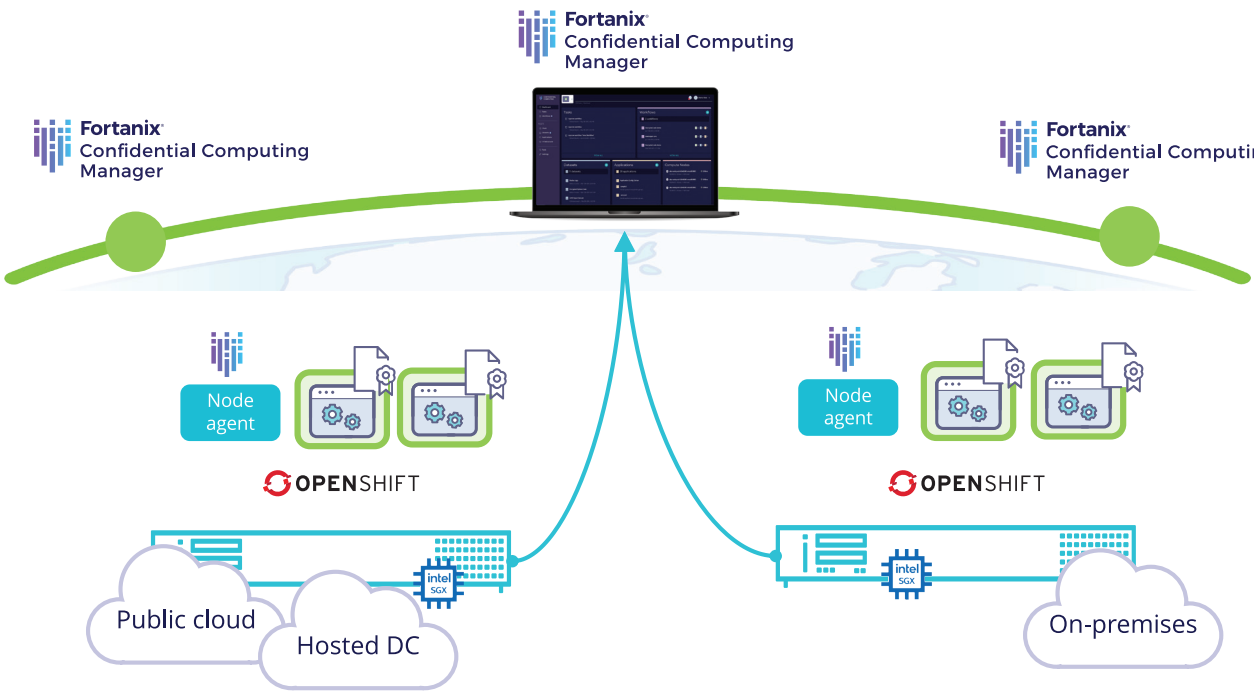

Figure 8. Confidential computing with Fortanix Runtime Encryption, Intel® SGX, and Red Hat® OpenShift®.

Red Hat OpenShift Operators are encapsulated software representations of discrete sets of capabilities that Red Hat OpenShift uses to provide services such as automating management or configuration tasks. Red Hat tests and validates Operators for functionality and soundness, creating an ecosystem of Red Hat OpenShift Certified Operators, which it offers on a SaaS basis. This certification process complements Red Hat tooling and other support, including the Operator SDK, which streamlines development by abstracting away a layer of complexity when working with Kubernetes APIs. Red Hat also provides the Operator Lifecycle Manager, which oversees the lifecycles of all Operators on a Kubernetes cluster, including installation, configuration, and updates.

The Fortanix Confidential Computing Manager Node Agent Operator is available to developers and cluster administrators through the Red Hat Embedded Operator Hub, which is included in Red Hat OpenShift. This simple deployment path allows for quick provisioning and streamlined maintenance of Fortanix Runtime Encryption capabilities on compute nodes as organizations stand up enterprise-grade Kubernetes services with Red Hat OpenShift. This combination of technologies provides benefits across industry verticals as enterprises transform their digital operations for multi-cloud infrastructure.

6 Conclusion

Confidential computing fills an important gap in enterprise security by protecting data while it is in use. Even sensitive types of data such as passwords and encryption keys are traditionally held in clear text within active system memory while computations are carried out on them, making them potentially vulnerable to interception. In particular, applications have been unable to shield data effectively from privileged processes such as OS services, dramatically reducing the effectiveness of encryption against compromised system software or insider threats.

Intel SGX enables developers to partition a region of memory as a more secure enclave that protects code and data while in use with encryption based on a hardware root of trust. Developers designate a portion of the application as "trusted," which runs in an enclave, shielded from all external processes and users, regardless of their privilege level. Fortanix Runtime Encryption streamlines the use of Intel SGX by allowing applications to take advantage of more secure enclaves without being modified and is included in Red Hat OpenShift by means of a Red Hat OpenShift Certified Operator.

The combined hardware and software stack composed of Intel SGX, Fortanix Runtime Encryption, and Red Hat OpenShift supports enterprise strategic imperatives to create more secure, highly automated, cloud-native environments for the future.

7 More Information

Intel® SGX:
intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions.html

Fortanix® Confidential Computing:
fortanix.com/solutions/use-case/confidential-computing/

Red Hat® OpenShift®:
redhat.com/en/technologies/cloud-computing/openshift

[1] Confidential Computing Consortium. https://confidentialcomputing.io/.

[2] See [70], [90], [71], and [69] at 3rd Generation Intel® Xeon® Scalable Processors - 1 - ID:615781 |Performance Index. Testing by Intel as of August 4, 2020. Performance comparisons relative to 2nd Gen Intel® Xeon® Scalable processors using a single buffer algorithm versus multi-buffer algorithms for 3rd Gen Intel Xeon Scalable processors. Results have been estimated based on pre-production tests at iso core count and frequency as of August 2020. Performance gains are shown for individual cryptographic algorithms.

[3] United Nations Office on Drugs and Crime, "Money Laundering." https://www.unodc.org/unodc/en/money-laundering/overview.html.

0222/RKM/MESH/346427-001US