# A security pattern for Zone isolation using Virtual Processors in mobile and embedded systems

EDUARDO B. FERNANDEZ, Florida Atlantic University
JORGE T. FORNERON, National University of Pilar, Paraguay

Employees in companies want to use their smart phones or other devices in their daily work. However, we do not want them to mix their private activities which do not require high security with work-related functions that need to be much more secure. In addition to security there may be also different requirements for the two types of activities. For this reason, we need to have two isolated zones to separate these activities. We present here a pattern that describes a way to perform this separation by using two virtual processors.

Categories and Subject Descriptors: D.2.11 **[Software Engineering]** Software Architectures–Patterns;

General Terms: Design

Additional Key Words and Phrases: Security patterns, smart phones, secure software, software architecture, virtual machines, virtual processors

**ACM Reference Format:**

Fernandez, Eduardo B., Forneron, Jorge T., 2018. A security pattern for zone isolation using virtual processors in mobile and embedded systems. Procs. of the 12th Sugarloaf PLoP, November 20-23, Valparaiso, Chile. 6 pages.

## 1. INTRODUCTION

People use smartphones and tablets for all kind of tasks. This reliance in their devices makes them want to use these devices also for their work functions A basic use of the phone is for social activities, which involve interacting with family and friends, as well as other private interactions. However, we do not want individuals to mix their private activities which may not require high security with work-related functions that need to be more secure. Also, private activities include phone calls and cameras which require real-time operation. The secure or trusted zone can be used for executing trusted operating systems, transaction processing, and certification activities. We present here a pattern that describes this separation, implemented using two virtual processors:

***Zone Isolation using virtual processors***--Provide two execution environments: one is intended to support the standard functions of phones: make phone calls, take pictures, or store lists of contacts. The other is intended for secure business functions such as accessing databases, using development environments, and employee email. The separation is performed by creating two types of strictly separated virtual processors.

This pattern is used in smart phones, tablets, game platforms, and some other embedded devices. Our audience includes mobile and embedded system designers, as well as security administrators of institutions which store sensitive data. We describe the pattern using a slightly-modified POSA template (Buschmann et al. 1996).

Authors' addresses: Eduardo B. Fernandez (corresponding author), Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: ed@cse.fau.edu; Jorge T. Forneron, National University of Pilar, Paraguay, email: jforneron@gmail.com.

## 2. ZONE ISOLATION USING VIRTUAL PROCESSORS

### 2.1 Intent

Provide two execution environments: one is intended to support the standard functions of phones: make phone calls, take pictures, or store lists of contacts. The other is intended for secure business functions such as accessing databases, using development environments, and employee email. The separation is performed by creating two types of strictly separated virtual processors.

### 2.2 Also known as

Virtual zones, virtual worlds, TrustZone

### 2.3 Context

Corporate databases contain data that may be valuable because they include information about business plans, customer medical records, and similar. We want our employees to have secure access to these resources according to their business functions. People want to use their smart phones and tablets for all kinds of applications, e.g. digital payments as well as all their private functions such as phone calls, interactions with friends, and similar. When using mobile devices to perform work functions we require a highly secure execution environment; for example, to store access rights, cryptographic keys, and trusted processes (Winter 2008).

### 2.4 Problem

Mixing employees' private functions with company information may leak valuable company information. A malicious user who has compromised the private zone could get data that can help her attack the work zone. A malicious application run by an unsuspecting user could also collect similar information. We need to keep these two environments strictly separated with controlled interactions.

### 2.5 Forces

The following forces apply to the possible solution:

- *Function requirements.* Different zones may be used differently which may impose different requirements on them; for example, real-time requirements for phone calls or cameras. We should be able to provide different types of zones to accommodate the needs of different applications.
- *Isolation.* We want to make sure that the different zones are strongly isolated. When an operating system in a zone crashes or it is penetrated by a hacker, the effects of this situation should not propagate to other operating systems running on different zones. For example, an attacker should not be able to access data in another zone.
- *Flexibility.* If there are significant changes in the threats or the type of applications or system software, we may need to modify the environment to implement different defense mechanisms.
- *Performance overhead*. We would like to have minimum performance overhead to perform this isolation.
- *Trust*. When the devices are part of some ecosystem we need to create trusted environments. It would be valuable to use the secure zone as a source of trust attestation. System activities such as bootstrapping also require a trusted root.
- *Controlled interzone communications*. Both zones need to communicate and there must be a way to perform this communication in a secure way.

### 2.6 Threats

Zone isolation can be attacked in the following ways:

- T1. Compromising the private zone may result in finding information that can be used to attack the secure side; for example, passwords to use the corporate database.
- T2. If the separation mechanism is complex, it is more likely to have vulnerabilities, making it easier to be compromised by an attacker.

- T3. Lack of security defenses in the work zone could allow an attacker to access enterprise resources.
- T4. Specific implementations of the zone communication mechanism may have vulnerabilities that can be exploited by attackers.

2.7 Solution

Use specialized virtual processors created by a virtualization environment (VE). This VE only needs to create a few virtual processors of predefined types. The virtualization environment can be purely software or supported by hardware.

2.7.1 *Structure*

Figure 1 shows the class model of this pattern. The Virtualization Environment (VE) creates Zones of two or more types: Private and Work. As in any VE, the VE controls the real Hardware. Each Zone can run a Guest Operating System supported by the VE.

2.7.2 *Dynamics*

Possible use cases include "Make a phone call", "Access a company database", "Update Subject Information", and Work Zone accessing Private Zone data.

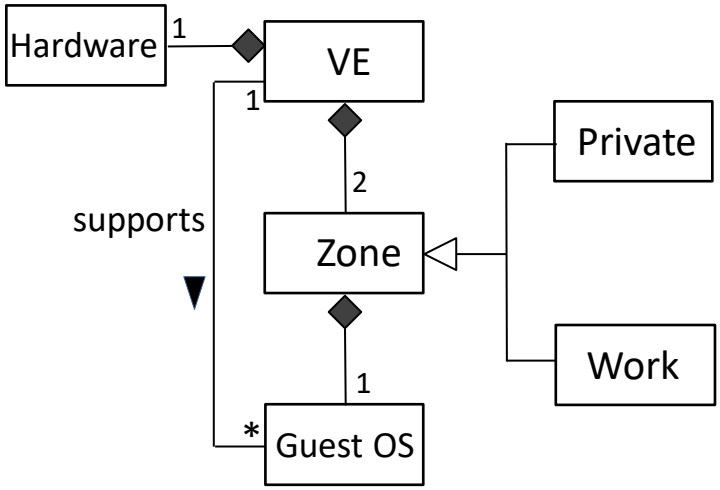


Fig. 1. Class diagram of the Zone Isolation pattern

2.8 Implementation

The early versions of this pattern used only software processors (Heiser and Leslie 2010). Later implementations use hardware support where a trusted zone can separate trusted process execution (Winter 2008). Almost all current phones use some variety of ARM with virtualization extensions (ARM, J. Guilbon 2018, Varanasi and Heiser 2011).

ARM processors are RISC processors with a TrustZone and implement architectural Security Extensions in which each of the physical processors provides two virtual processors, one being considered non-secure (Non Secure World), the other being considered Secure (Secure World), and a mechanism to switch between the two, known as Monitor mode (ARM, Wikipedia, Ngabonziza et al., 2016). As illustrated in figure 2, TrustZone consists of a monitor, an optional operating system (OS) and optional applications. Typically, a rich operating system is run in the less trusted world, with smaller security-specialized code in the more trusted world, aiming to reduce the attack surface. A Trustzone implementation could be all those components like on the Qualcomm or Trustonic implementations, or only a Monitor as done in the Nintendo Switch. TrustZone is used for many purposes, including DRM, accessing platform hardware

features such as stored RSA public key hash in eFuse, Hardware Credential Storage, Secure Boot, Secure Element Emulation, and others.

Another implementation of trusted zones is Intel Software Guard Extensions (SGX), a CPU instruction set that allows user-level code to allocate private regions of memory, called enclaves, that are protected from processes running at higher privilege levels. Intel designed SGX to be useful for implementing secure remote computation, secure web browsing, and digital rights management (DRM).

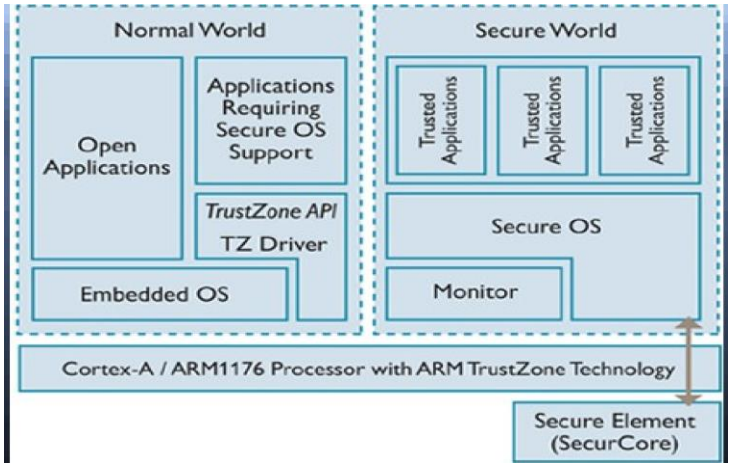


Fig. 2. A possible separation architecture (ARM)

.

2.9 Known uses

- The OKL4 Microvisor implements a combination virtual machine monitor/microkernel (Heiser and Leslie 2010) that creates a few types of VMs. The OKL4 Microvisor supports ARM hardware virtualization extensions, as introduced in the Cortex-A15 processor. The use of hardware virtualization greatly reduces the changes required to a guest OS.

- Apple phones have a Secure Enclave that runs a customized version of the OKL4 microvisor (Apple 2018). The Secure Enclave is an Apple coprocessor that has encrypted memory and a random number generator. Its function is to keep data integrity. Communication with the application processor uses an interrupt-driven mailbox.

- Samsung phones' Knox security functions use virtual machines for this separation (Samsung). Its workspace also uses two-factor authentication. ARM TrustZone Secure World is isolated by hardware and intended for sensitive software. The Secure World has a higher privilege and can access Secure and Normal World resources. TrustZone is used also for purposes such as detecting modifications to the kernel.

- Huawei phones use hardware-supported virtual processors using the Kylin CPU, based on the ARM architecture (Huawei 2017). One of the virtual processors is a Trusted Execution Environment.

- Motorola phones used the OKL4 (Geiser2009).

2.10 Consequences

This pattern provides the following advantages:

- *Function requirements.* We can provide different types of zones, tailored to different types of applications by installing differen types of operating system in a zone.
- *Isolation*. The separation code can be proven correct, thus assuring isolation (Klein et al., 2009). Even if not proven correct, since it is a small system it can be carefully examined; for example a prototype L4 uses only about 6K lines of code (Varanasi and Heiser 2011).
- *Flexibility*. We can add easily more types of virtual machines or more instances of a given type of VMs.
- *Performance overhead*. Experimental studies have shown that the overhead is minimum when using hardware support (Winter 2008).
- *Trust*. The secure zone can be used as a trust zone, useful for creating trusted computing environments {Eriksson, Pourzandi, Smeets, 2014).
- *Controlled interzone communications*. Both zones need to communicate and there must be a way to perform this communication in a secure way.

Liabilities include :

- Software has a larger attack surface than hardware, which would make software virtualization easier to attack than the isolation based on hardware-defined zones. Because of this all modern phones use hardware trusted zones.
- There is a performance overhead in the use of virtual machines if no hardware support is used.
- Use of specialized hardware reduces flexibility for migrating to better future processors.

2.11 Countermeasures

T1 requires providing simple security facilities such that the users can set up authentication and authorization. T2 requires using a secure hypervisor such as the microvisor in (Heiser and Leslie 2010), which was proven secure (Klein et al. 2009). T3 can be handled if the secure zone has appropriate security defenses. T4: Several vulnerabilities have been found in the ARM processors which have been patched in recent versions (J. Guilbon 2018).

2.12 Related patterns

- This pattern is based on the Virtual Machine Operating System pattern presented in (Fernandez 2013), specialized for a specific use.
- The Microkernel Operating System pattern describes how to move as much of the operating system functionality as possible from the kernel into specialized servers, coordinated by a microkernel (Buschmann et al. 1996). The microkernel itself has a very basic set of functions. Operating system components and services are implemented as external and internal servers. A microkernel is the basis for the microvisor of (Heiser and Leslie 2010).
- It is also possible to separate environments through cryptography, although that approach is not convenient for this type of functions. Another type os separation is based on multilevel architectures (Fernandez and La Red 2008).
- The Virtual Machine Environment (Syed and Fernandez 2016) describes the creation of virtual machines in clouds.
- Further protection of application data can be provided by using sandboxes withing zones, examples of the Controlled Virtual Address Space in (Fernandez 2013).

## 3. CONCLUSIONS

This pattern is used in a variety of modern portable devices such as smartphones, tablets, game playing devices, and similar, which makes it of practical importance. Another possible related pattern could be the hardware architecture of trusted zone processors and TPMs, which are in our list of future work.

ACKNOWLEDGEMENTS

We thank our shepherd Gunther Pernul for his valuable comments which significantly improved this paper.

## REFERENCES

Apple inc., iOS security: iOS 11.4, August 2018, https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf

ARM, https://www.arm.com/files/pdf/AT-Exploring_the_Design_of_the_Cortex-A15.pdf

F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, and M. Stal., Pattern- oriented software architecture, Wiley 1996.

M. Eriksson, M. Pourzandi, B. Smeets, “Trusted computing for infrastructure”, Ericsson Review, October 24, 2014, 2-8.

E.B.Fernandez and D. LaRed M., "Patterns for the secure and reliable execution of processes". *Procs. of the 15th Int.Conference on Pattern Languages of Programs (PLoP 2008)*, Nashville, TN, Oct. 2008.

E. B. Fernandez, Security patterns in practice: Building secure architectures using software patterns, Wiley, April 2013.

J. Guilbon, Quarkslab’s blog, “Introduction to Trusted Execution Environment: ARM’s TrustZone”, *19 June 2018, https://blog.quarkslab.com/introduction-to-trusted-execution-environment-arms-trustzone.html*

G. Heiser, “The Motorola Evoke QA4: A Case Study in Mobile Virtualization”, 2009.
*https://www.researchgate.net/publication/242743911_The_Motorola_Evoke_QA4_A_Case_Study_in_Mobile_Virtualization*

Gernot Heiser and Ben Leslie. “The OKL4 Microvisor: Convergence point of microkernels and hypervisors”. Proceedings of the 1st Asia-Pacific Workshop on Systems (APSys), pp. 19–24, New Delhi, India, August 2010.

Huawei, EMUI 8.0 Security Technical White Paper, October 31, 2017, https://consumer-img.huawei.com/content/dam/huawei-cbg-site/en/mkt/legal/privacy-policy/EMUI%208.0%20Security%20Technology%20White%20Paper.pdf

G. Klein et al., *“seL4: Formal verification of an OS kernel”, ACMTrans. Comp.Syst.,* 32 (1). 2:1-2:70, 2014

B.Ngabonziza, D.Martin, A. Bailey, H. Cho, S. Martin, “TrustZone explained: Architectural features and use cases*”, 2016 IEEE 2nd Int. Conf. on Collaboration and Internet Computing,* 445-451.

Samsung, *Samsung Knox security solution (white paper), https://www.samsungknox.com/docs/SamsungKnoxSecuritySolution.pdf*

M.H.Syed and E.B. Fernandez, *''A pattern for a virtual machine environment'', Pattern Languages of Programs Conference (PLoP 2016), October 24-26, 2016*

P. Varanasi and G. Heiser, “Hardware-supported virtualization on ARM”, *APSys’11,* July 11-12, 2011, Shanghai, China.

Wikipedia, Trust Zone, https://en.wikipedia.org/wiki/ARM_architecture#Security_extensions_(TrustZone)

J. Winter, “Trusted computing building blocks for embedded Linux-based ARM TrustZone platforms”, STC’08, Oct. 31, 2008, Fairfax, VA, USA. ACM, 21-30.