



**Opening the NFC  
stack to Java  
and native  
applications**

**driving trust** **inside**  
SECURE

[www.insidesecond.com](http://www.insidesecond.com)

## OVERVIEW

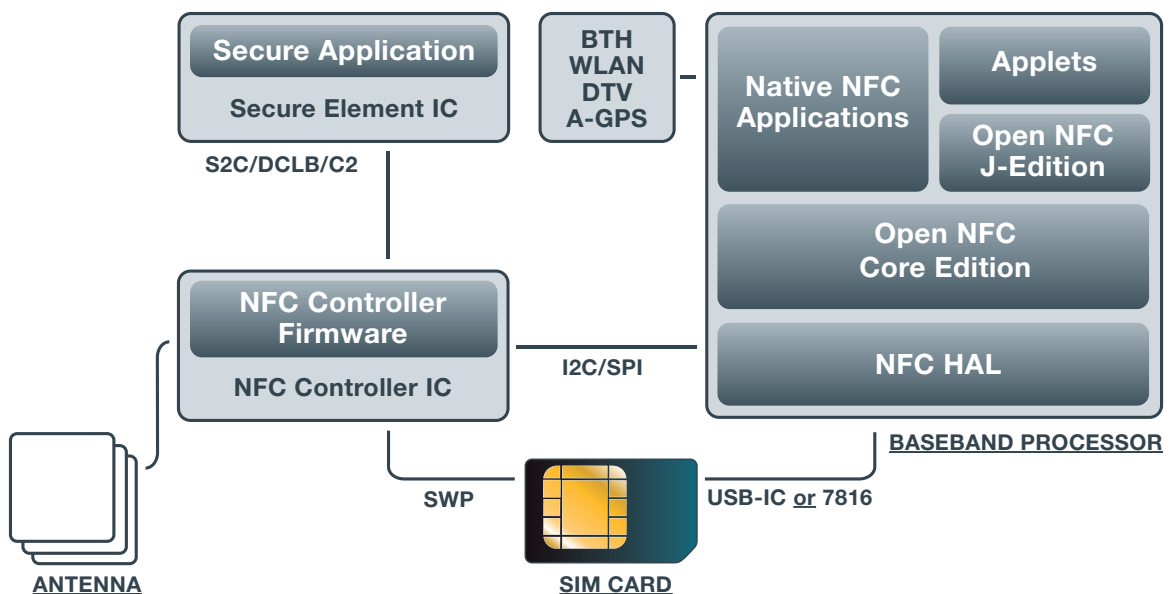
Open NFC™- Core Edition is a software stack under open-source license, which implements NFC protocols and functionalities on top of a dedicated NFC controller IC. INSIDE Secure has proposed Open NFC as a standard control-plane stack, and delivers this stack to handset integrators and partners, under an Apache v2.0 open-source license. This paper will show how the functionality provided by such an open-source approach meets the needs of mobile integrators for an Application Programming Interface (API) that will dramatically reduce NFC application development cycles.

As a pioneer in the field of NFC controllers and contactless integrated circuit cards, INSIDE has been offering a NFC driver stack for the last two years. The company is a leader in real hardware NFC IC integrations and Universal Integrated Circuit Card (UICC) interoperability testing on a broad range of product architectures. Opening the source of the API, already in use in several handset designs, represents the next step in bringing open integration to the entire mobile industry.

The most critical factor of the Open NFC - Core Edition is the added security realized through INSIDE's strong control of source-code-level security. Open NFC was designed to be both hardware-independent, and distributed in source-code format. INSIDE's years of experience has allowed the company to integrate countermeasures to prevent attacks based on knowledge of source code. Experience with attacks on Mifare and other codes demonstrates that a reliance on algorithm-level security is not enough to guarantee a secure design. In the same way, a reliance on secrecy of implementation is inadequate. Because of the protection codes INSIDE offers as countermeasures, the knowledge of the source files does not provide an avenue for attack for the potential hacker.

Open NFC - Core Edition provides several APIs with different levels of functionalities, from low-level contactless RF control to high-level NFC Forum message-handling (NFC Data Exchange Format, or NDEF messages). Figure 1 below represents a NFC handset system overview. We can see that typical designs assume an interface between common wireless and media interfaces such as Wi-Fi, television or GPS, and NFC NFC services, either through native applications and Java applets. Besides the contactless payment or transport applications that are hosted in the USIM card or UICC, provision of an interface from common handset functionalities to the NFC will leverage the added value for each of the NFC smart phone features.

**Figure 1 - NFC handset for Windows CE or Android architecture**

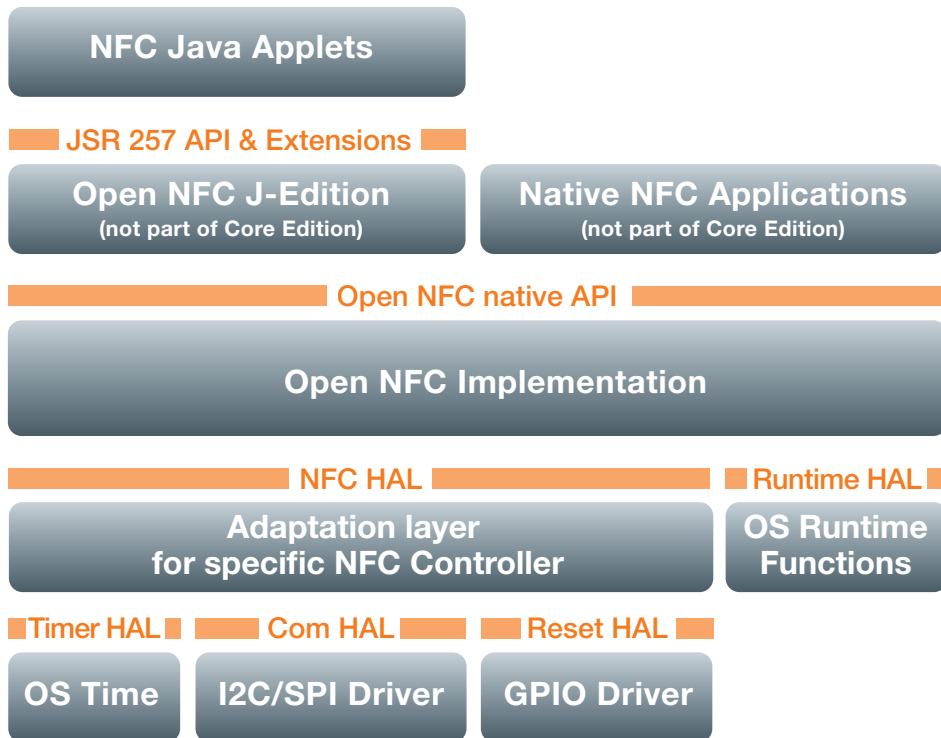


On the top functional layer, Open NFC – Core Edition supports NFC applications native to the OS. Open NFC – Core Edition can connect to a Java Specification Request 257 (JSR 257) implementation and access the portfolio of Java-based applications that will make host NFC capabilities more profitable.

JSR 257 defines the contactless API for the Java 2 Micro Edition (J2ME) mobile framework. In order to accelerate JSR 257 interfacing and validation, INSIDE separately distributes the license Apache V2 Open NFC J-Edition, which guarantees success in using the Test Compliance Kit (TCK), as compared to an extensive range of contactless targets. Versions of the Core Edition are available for both Windows CE and Android.

To access the NFC hardware, Open NFC – Core Edition relies on the NFC controller hardware abstraction layer (NFC HAL); the NFC HAL is a well-defined interface abstracting the basic functionalities of a NFC Controller IC. In the case of INSIDE’s MicroRead NFC controller, the HAL relies upon GPIO and communication interfaces on the hardware side, and also requires a small set of OS runtime functions and one Timer as seen in Figure 2.

## Figure 2 - Software Architecture



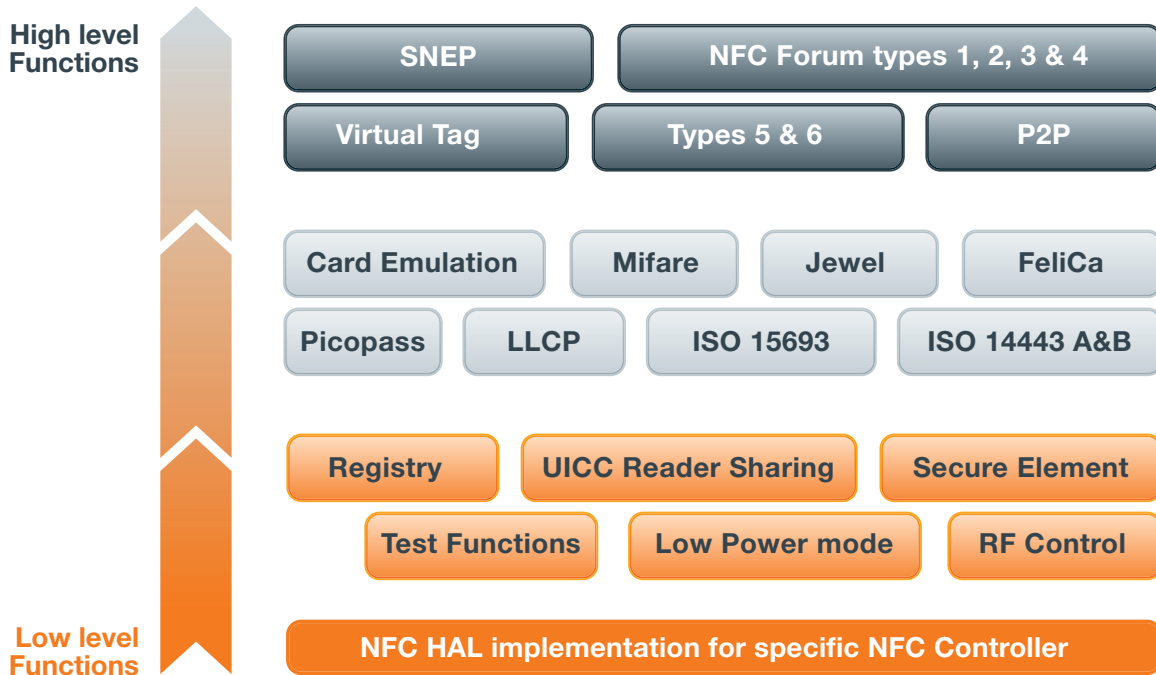
As NFC relies upon an extensive existing contactless ecosystem, one of the major roles of the API is to make an abstraction of a large number of contactless chip technologies. These constitute more than 15 product lines using four RF standard families, split over a dozen different RF protocol-layer entry points. Given this, it is apparent that JSR TCK and NFC interoperability testing may represent a potential hazard for project completion, as long as the NFC APIs remain hardware- and vendor-dependent black boxes.

Using an open-source NFC stack guarantees fast reuse and maintenance in the event of corrections or added features that may be required during NFC project development., Nevertheless, use of the open stack preserves hardware independence.

From high level down to HAL layer, Open NFC – Core Edition is actually built over an infrastructure that is meant to connect to NFC system-critical aspects, such as low power management, RF control and NFC functionality shared with UICC and Secure Elements. Having an open source implementation of this API layer ensures the fastest porting to any physical NFC front end component. Furthermore, it allows handset integrators to implement proprietary management that may provide a strong design differentiator.

When accessing test functions, the optimization and coverage of the testing process, either on the production line or in post-sale environments, paves the way to numerous cost savings during the product life cycle. This is particularly true when implementing a new technology within a highly integrated system. As NFC integration represents a new and unfamiliar knowledge realm for many handset vendors and manufacturing subcontractors, the experience in NFC testing and prototyping will translate into a need for a straightforward implementation of new testing functionalities. Using an open source NFC stack will remove many potential design bottlenecks.

**Figure 3 - Open NFC functionalities**



The API provides the entire function set that is required for the building of new records and messages, the parsing of records and messages, and the tag-based reading or writing of NDEF messages. Helpers are provided to handle NFC Forum text and URI handling.

Apart from the effortless ability to format of NFC Forum standardized tags, the NDEF capabilities will be appreciated when using the Virtual Tag feature. A Virtual Tag-based application can simulate a tag with the handset. The API provides the ability to write a NDEF message into the virtual tag. Then, the MicroRead function simulates the tag with the message content, under the driver stack control.

Apart from NDEF applications, the ISO 7816-4 support is intended to interact with mart card applications. This protocol was originally designed for contact smart cards, but it is also used on top of ISO 14443-4 A/B contactless protocol to communicate with NFC cards. The Open NFC framework implements ISO 7816-4 to communicate with the contactless cards of type 14443-4

As far as the UICC is concerned, Open NFC is compatible with the "connectivity" protocol defined by ETSI Host Controller Interface (HCI) that is used between the NFC and the UICC over the Single Wire Protocol (SWP) interface. The connectivity interface allows the reception of payload-carrying events from the USIM - to the handset. It may be used to display the results of a transport or payment transaction . It may also trigger a communication with the UICC on the ISO 7816 interface.

Preserve Proprietary Values in Designs.

INSIDE licenses Open NFC to its customers and partners under the Apache V2.0 software license. The Apache license requires preservation of the copyright notice and disclaimer, and allows the use of the source code for the development of proprietary software as well as free and open source software.

Such a software license allows IC vendors, handset integrators or platform vendors to develop any NFC IC hardware layer porting, and license it under any conditions. This is also the case on the Java side of the stack, where a JSR implementation is separately distributed.

Open NFC – Core Edition reference implementation is available under Linux (TI OMAP2430 and generic ARM9), Windows CE 6.0, and Windows Mobile 7 and Android. It is currently being ported on more than five mobile device integrators, including several top OEMs. Ongoing developments include Nucleus, REX and other Real time OSs. It is offered as a compatible from the legacy “MSF” driver stack for INSIDE’s MicroRead within the roadmap shown in Figure 4 .

**Figure 4 - Open NFC Roadmap**

