A Flexible Invocation Framework
for Java Card

Michael Montgomery and Ksheerabdhi Krishna

Austin Product Center, Schlumberger, 8311 North FM 620 Rd, Austin, TX 78726, USA
mmontgomery@slb.com, kkrishna@slb.com

Abstract. The Java Card Specification specifies a framework suitable for most tasks that interface with ISO-7816 protocols. However, this framework limits Java Card applets to only handling tasks that are supported by this protocol. Tasks such as physical security using the standard Wiegand protocol, cards which test ISO-7816 card readers by pushing the limits of the protocol, and applications that depend on interfacing to different communication mechanisms, such as full duplex serial, USB, TCP/IP, or Bluetooth are among the many applications that cannot be handled within the Java Card Framework. In this paper we describe an approach that maintains backwards compatibility with the current Java Card framework, while enabling applications a means to escape from some of the constraints of the framework when necessary.

1 Introduction

The Java Card Specification [1] presents an architecture for interoperable Java Cards. The specification consists of three parts. The Java Card Virtual Machine Specification outlines the characteristics of the interpretation engine accompanied with file format layouts for execution and interfaces. The Java Card Runtime Environment (JCRE) Specification describes the features of the card runtime such as communications, method sharing, transactions, installation, and invocation mechanisms of Java Card applets. The Java Card API specification describes the on-card API used by the Java Card applet programmer to write Java Card applets.

To enforce interoperability, the JCRE lays out certain firm rules to describe runtime behaviour. This necessarily limits the kinds of applets that can be run within this framework. Furthermore, there is no way specified to escape the framework constraints when necessary to do so to implement specific applications, thus rendering those applications unimplementable with the current Java Cards.

In this paper, applet always implies a card application, and often will be used without qualification for brevity.
2 Java Card Runtime Environment

The Java Card assumes that the underlying application level protocol between an on-card applet and the host application is performed by exchanging data packets known as APDUs (Application Protocol Data Units). This protocol for communication between a smart card and a host application is specified in ISO 7816-4 [2] and comprises two primary structures, one to send commands to the card (C-APDU) and another to send a response back from the card (R-APDU).

![Diagram of communication between host application and JCRE](image)

**Figure 1.** Applet Communication with the JCRE

Data sent to the card is first received by the JCRE. As shown in Figure 1, the JCRE routes the APDUs targeted to the card to the appropriate applet [3]. Hence, the JCRE assumes a particular structure for the incoming data to parse it appropriately. When the host application wants to select an applet to run, it sends an APDU that specifies the SELECT command and the AID of the requested applet. The JCRE searches its internal table to find the AID that matches the one specified in the APDU and if found, selects the corresponding applet to run. All subsequent APDUs are forwarded to the selected applet until a new applet is selected. Access to the contents of the APDU are made available via an on card API as shown in Figure 2.

![Diagram of ISO-7816 communication framework](image)

**Figure 2.** Typical ISO-7816 Communication Framework