	an interpreter. This allows programs to operate independent of processor. With the newer technique of also creating virtual peripherals then the whole is referred to as a 'virtual machine'." *The '683 patent*, col. 3, ll. 48-54. **OTA** "The hardware environments include . . . a target which is some form of payment terminal." *OTA*, p. 73, ¶3. Such payment terminals are used for communication: "In the embedded systems for which OTA is targeted, system functions cover not only OTA functions such as communications . . . ." *Id*. at p. 75, ¶8. A virtual machine is used on OTA terminals by implementing code that controls the device**:** **"**OTA terminal code is based on a single virtual machine which is emulated on the actual devices." *Id*., at p. 74, ¶3. "The software in every OTA terminal is written in terms of a common virtual machine." Id., at p. 74, ¶4.

Element A. The first element of the '683 patent is directed to a virtual function processor that includes instructions for controlling operating of the communication device. The concept of using a virtual machine that includes instructions for controlling operation of a device is taught by EMV '96. EMV'96 describes "a theoretical microprocessor" which is the claimed "virtual function processor." A "virtual processor" is actually redundant with a "virtual machine," since the processor is what necessarily runs the machine. In EMV '96, a virtual processor interprets instructions and allows a point of service terminal to be controlled based

on such instructions. Further, OTA indicates that a virtual machine may be implemented in the form of software on an OTA terminal.

[A] a virtual function processor and function processor instructions for controlling operation of the device, and	**EMV '96** "In the case of an interpreter capability, these modules will be code, written in a virtual machine instruction set implemented within the terminal, to be interpreted by the terminal control program." *EMV '96*, at §1.2, p. II-2 (emphasis added). "An interpreter implementation defines a single software kernel, common across multiple terminal types. This kernel creates a virtual machine that may be implemented on each CPU type and that provides drivers for the terminal's input/output (I/O) and all low-level CPU-specific logical and arithmetic functions. High-level libraries, terminal programs and payment applications using standard kernel functions may be developed and certified once; thereafter, they will run on any conforming terminal implementing the same virtual machine without change. Therefore, a significant consequence of an interpreter is a simplified and uniform set of test and certification procedures for all terminal functions." *Id.*, at §1.4.1, pp. II-3–II-4. "The application software in every terminal using the interpreter approach is written in terms of a common virtual machine. The virtual machine is a theoretical microprocessor with standard characteristics that define such things as addressing mode, registers, address space, etc." *Id.*, at §1.4.2, p. II-4 (emphasis added). "Virtual machine emulation may be accomplished by one of three methods: interpreting virtual machine instructions, translating the virtual machine language into a directly executable 'threaded code' form, or translating it into actual code for the target CPU." *Id.*,

	at §1.4.4, p. II-5. "Programs may be converted to an intermediate language, between the high level source language used by the programmer and the low-level machine code required by the microprocessor, and subsequently transported to the target terminal to be processed by the terminal into an executable form." *Id*., at §1.4.4, p. II-5. **OTA** "The software in every OTA terminal is written in terms of a common virtual machine. This is a theoretical 32-bit microprocessor with standard characteristics defining addressing modes, stack usage, register usage, address space, etc. The kernel for each particular CPU type is written to make that processor emulate the virtual machine." *Id*., at p. 74, ¶4. "Virtual machine emulation may be accomplished by one of three methods: interpreting token representing VM instructions (like Java), translating these token into a directly executable 'threaded code' form (like Open Firmware), or translating them into actual code for the target CPU. The latter two methods offer improved performance at a modest cost in added target complexity." *Id*., at p. 74, ¶5.

Element B. This element of the '683 patent is directed to having stored instructions on the communication device that describe message data. POS devices have the ability to interpret received messages and compose messages for transmission. Therefore, POS devices necessarily have stored instructions that describe the format of such messages such that received messages can be understood and messages can be sent that will be understandable to the recipient

device. On point, the POS device of EMV '96 creates messages to be sent (e.g., to a host system) and interprets received messages (e.g., from the host system). EMV '96 explicitly states that the ISO 8583 standard, which defines how messages for financial card transactions are composed, is used by the device of EMV '96.

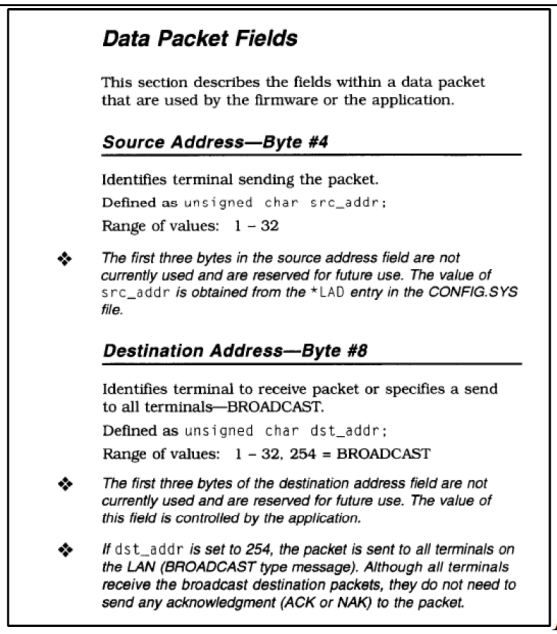
Additionally, it would be obvious to combine EMV '96 with the OMNI 300 Series Terminal. Both EMV '96 and OMNI 300 are directed to forms of payment terminals, such as point-of-service (POS) devices that are used in conducting financial transactions. One of skill in the art looking at updating the OMNI 300 in 1996 would want to make sure it complied with EMV '96, the de facto industry standard. OMNI 300 explicitly defines how data packets that are exchanged as messages on a network are defined. Since these message definitions are used by the device, the device necessarily has a stored description of such messages.

OTA indicates that the I/O of the terminal may be tested. This testing involves a download to the target terminal from another system, such as a PC host. Such a download would involve the reception of messages by the terminal. In order to receive and interpret such messages, a description of at least the formatting of such messages would need to be stored by the terminal. Further, I/O of a payment terminal necessarily includes messages being transmitted because payment

terminals rely on the ability to communicate with a host system to conduct transactions.

[B] message instruction means including a set of descriptions of message data;	**EMV '96** "Messages typically flow from the terminal to the acquirer and from the acquirer to the issuer." *Id.*, at §2.1, III-6. "The data transmitted in messages as defined in ISO 8583:1987 and bit 55 from ISO 8583:1993 . . . ." *EMV '96*, at p. D-1. **OMNI 300** The arrangement of data within packets for transmission via a network is defined: LAN messages are exchanged on the LAN using a data packet, which consists of an 18-byte header, 0 to 1500 bytes of binary data and a 2-byte checksum. The packet header is defined as `struct packet_header` in `<lan.h>`. *OMNI 300*, p. 10-4.

Control	Name	Label (<lan.h>)	Range	Size (bytes)
ignored	Source address	src_reserved	—	1
ignored		src_net	—	1
ignored		src_socket	—	1
firmware		src_addr	1 – 32	1
ignored	Destination address	dst_reserved	—	1
ignored		dst_net	—	1
ignored		dst_socket	—	1
application		dst_addr	1 – 32, 254	1
firmware/ application	Sequence number	tx_seq	0 – 0xFFFFFFFF	4
ignored	Protocol	protocol	—	2
application	Application message type	app_type	app.-defined	1
firmware/ application	Message type	type	0x0000–0x0082	1
firmware	Data length	dg_length	0 – 1500	2
application	Data	—	binary data	< 1500
firmware	CRC-16	—	0 – 0xFFFF	2



Data Packet Fields

This section describes the fields within a data packet that are used by the firmware or the application.

Source Address—Byte #4

Identifies terminal sending the packet.

Defined as `unsigned char src_addr;`

Range of values: 1 – 32

- *The first three bytes in the source address field are not currently used and are reserved for future use. The value of `src_addr` is obtained from the `*LAD` entry in the CONFIG.SYS file.*

Destination Address—Byte #8

Identifies terminal to receive packet or specifies a send to all terminals—BROADCAST.

Defined as `unsigned char dst_addr;`

Range of values: 1 – 32, 254 = BROADCAST

- *The first three bytes of the destination address field are not currently used and are reserved for future use. The value of this field is controlled by the application.*
- *If `dst_addr` is set to 254, the packet is sent to all terminals on the LAN (BROADCAST type message). Although all terminals receive the broadcast destination packets, they do not need to send any acknowledgment (ACK or NAK) to the packet.*

Id., at p. 10-5 (excerpt of packet definition).

OTA

"For final testing using the terminal's own kernel and I/O, these programs would be tokenized (see discussion below) and downloaded to the target terminal." *OTA*, ¶74.

Element C. This element of the '683 patent is directed to handling the assembly, disassembly, and comparison of messages in accordance with the message definitions of element **[B]**. These functions are performed by a software module that is called by the software module of element **[A]**. Assembly of a message can be understood as creating a message for transmission to another device. Disassembly of a message can be understood as extracting data from a message received from another device. Comparison of a message can be understood as analyzing an "incoming message against the description specified by the message

instruction means . . . ." *Ogilvy '683 patent*, col. 17, ll. 47-49. Therefore, a received message is checked against the descriptions of element **[B]**.

EMV '96 discloses such reception, transmission, and comparison of messages. Such handling of messages would be invoked by instructions executed by the virtual machine of the terminal. For instance, EMV '96 notes that a transaction message may be used when online data capture is being used. The terminal is further configured to receive a message that is a transaction response and transmit results in a message. Therefore, for such transmission and reception of messages, assembly and disassembly of messages is necessary. Further, assembly or disassembly of messages would be performed in accordance with how such messages are defined as noted in element **[B]**.

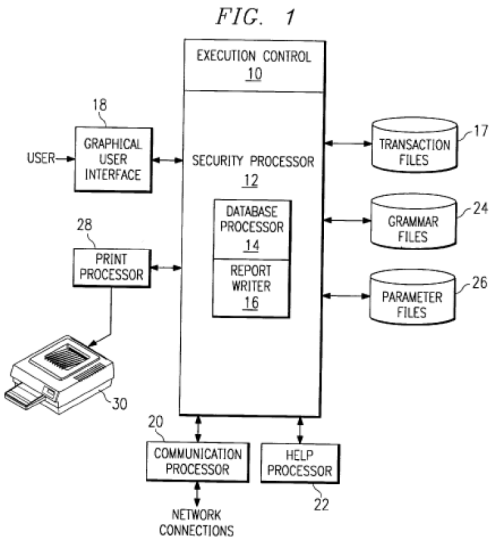
Regarding a comparison being performed, EMV '96 discloses that an error can be detected during communication involving the system or the terminal, which would require some form of comparison to a stored definition of how an acceptable message should be formatted.

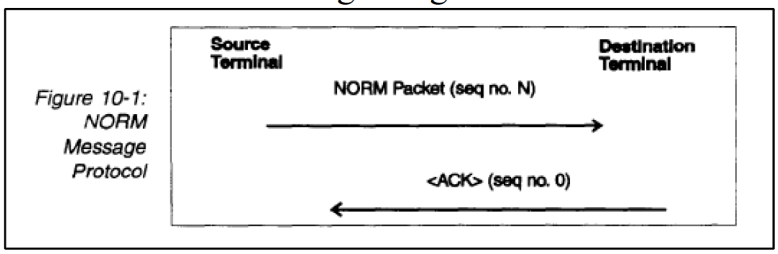
In addition, it would be obvious to combine EMV '96 with OMNI 300, (Custy et. al, US Pat. No. 5,774,879 assigned to First Data – hereinafter "First Data '879 Patent"), or both. One of skill in the art looking at updating OMNI 300 in 1996

would want to make sure it complied with EMV '96, the de facto industry standard. The First Data '879 Patent also deals with a POS device, and would be reviewed for the same reason. EMV '96 and the First Data '879 Patent disclose financial processing systems that are intended to be portable among various software and hardware platforms, and thus are not just related POS subject matter, but are both directed to the same emulatable concept. The First Data '879 Patent teaches the use of a distinct communication software module, referred to as a "communication processor" [the claimed "virtual message processor"] handling messaging functions with an "execution control processor" software module [the claimed "virtual function processor"]. The First Data '879 Patent explicitly teaches how distinct software modules can be used to handle communication processing separately from other processing.

OMNI 300 discloses that a service call invoked by an application can initiate a data transfer. Regarding a message comparison, OMNI 300 performs a check for corrupt data received in messages by comparing a received cyclic redundancy check (CRC) value with calculated CRC value to determine if a match is present. Two bytes in the header of messages exchanged via a network are used for the comparison of CRC values, as indicated in the first table of element **[B]**.

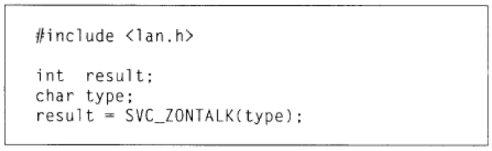
[C] a virtual message	**EMV '96**

processor, which is arranged to be called by the function processor and which is arranged to carry out the message handling tasks of assembling the messages, disassembling messages and comparing the messages under the direction of the message instruction means that is arranged to provide directions for operation of the virtual message processor,	"An authorisation message shall be used when transactions are batch data captured. A financial transaction message shall be used when online data capture is performed by the acquirer." *EMV '96*, at §2.1, p. III-6. "The terminal shall be able to support at least one or more Issuer Scripts in each authorization or financial transaction response it receives . . . ." *Id.*, at §2.2.9, p. I-10. Further, "The terminal shall transmit the Issuer Script Results in the batch data capture message . . . ." *Id.* " '0F' - PROCESSING ERROR - Displayed to the cardholder or attendant when the card is removed before the processing of a transaction is complete or when the transaction is aborted because of a power failure, or the system or terminal has malfunctioned, such as communication errors or time-outs." *Id.*, at pp. III-3, III-4 **<u>The First Data'879 Patent</u>**  FIG. 1 EXECUTION CONTROL 10 SECURITY PROCESSOR 12 DATABASE PROCESSOR 14 REPORT WRITER 16 GRAPHICAL USER INTERFACE 18 USER TRANSACTION FILES 17 GRAMMAR FILES 24 PARAMETER FILES 26 PRINT PROCESSOR 28 30 COMMUNICATION PROCESSOR 20 HELP PROCESSOR 22 NETWORK CONNECTIONS *First Data*, Fig. 1. "It should be understood that the term processor used herein refers to a software module operating to perform a particular task or group of tasks. A single such module may actually be running on a variety of hardware architectures which could include single or multiple hardware processors." First Data '879 Patent, at col. 2, ll. 58-63. "The execution control processor 10 is also coupled to

	a communications processor 20." *Id.*, at col. 4, ll. 5, 6. **OMNI 300** An application causes communication to occur: "Terminal applications may be programmed to initiated ZONTALK 2000 downloads (full or partial) by using the following service call: result – SVC_ZONTALK(x) . . . ." *OMNI 300*, p. 2-4. Messages are both assembled for transmission and disassembled following being received:  *Figure 10-1: NORM Message Protocol* *Id.*, at p. 10-8. Such messages are constructed and deconstructed in accordance with "struct packet_header" as detailed in element **[B]**. "Corrupt Data – If the destination terminal receives a data packet whose CRC-16 entry does not match the terminal's CRC calculation, it sends a NAK to the sending terminal." *Id.*, at p. 10-10.

Element D. This element of the '683 patent is directed to calling a message processor to handle a message. Each of EMV '96 and the first data '879 patent disclose a software module being called to handle a message. Further, OMNI 300 specifies how an application can call on a communication software module to handle a message exchange such that a download can occur. In OMNI 300, an application (which could be executing as a first software module) can make a call to start a download (which is handled by a separate software module).

[D] whereby when a message is required to be handled by the communications device the message processor is called to carry out the message handling task,	**EMV '96** "If the card indicates to process online, the terminal shall transmit an authorization or financial transaction request message, if capable." *EMV '96*, at §2.2.7, p. I-10. "The terminal shall be able to recognize the tag for the Issuer Script transmitted in the response message. If the tag is '71', the terminal shall process the script before issuing the second GENERATE AC command." *EMV '96*, at §2.2.9, p. I-11. **The First Data '879 Patent** "The Execution control processor 10 is also coupled to a communications processor 20. The communications processor 20 allows the integrated system to communicate with other system through network connections. According to one embodiment of the present invention, the communications processor 20 allows for communication with an integrated communications platform system to allow for session-based communications with a host computer." *First Data*, col. 4, ll. 5-8. "The communications processor 20 acts as an interface between the integrated systems and whatever communication facilities are available via network connections." *Id.*, at col. 4, ll. 32-35. "The system of the present invention then uses this serial number to access the data files of the host computer through the communications processor 20 and network connections as described in step 116." *Id.*, at col. 14, ll. 4-7. **OMNI 300** The application may initiate a LAN download: "OMNI 300 Series LAN terminals support application download via the LAN port. Prior to beginning the download, each terminal must have a particular set of parameters present in its CONFIG.SYS file." *OMNI 300*, p. 2-4. "Terminal applciations may be

	programmed to initiate ZONTALK 200 downloads (full or partial) by using the following service call: result=SVC_ZONTALK(x); . . .” *Id*  *OMNI 300*, p. 10-38.

Element E. This element of the ‘683 patent requires that the implemented virtual machine be emulatable on different computers (e.g., different POS devices) having different, incompatible hardware platforms or different, incompatible operating systems. EMV ’96 and the First Data ‘879 Patent disclose such an ability to be emulatable on different hardware platforms and/or different operating systems. Further, OMNI 300 indicates how different dialects of the C programming language can be used to allow for compatibility across varying systems. OTA indicates how application programs for OTA terminals can be “completely platform independent.” Since all teach emulatable software, it would be obvious to combine.

[E] wherein the virtual machine means is emulatable in different computers having incompatible hardwares or operating systems,	**EMV ‘96** “The kernel for each particular CPU type is written to make that processor emulate the virtual machine. The virtual machine concept makes a high degree of standardisation possible across widely varying CPU types and simplifies program portability, testing, and certification issues.” *EMV ‘96*, at ¶1.4.4, p. II-5.

	Also, see element **[A]** re: *EMV '96*, §1.4.1, pp. II-3–II-4. **<u>The First Data '879 Patent</u>** "The financial instrument processing system of the present invention comprises an object-oriented software system that is highly portable between various hardware platforms. The architecture of the integrated software system is constructed such that the system can be easily and conveniently ported to a variety of operating system such as MS DOS, Windows, OS2, or UNIX." *First Data '879 Patent*, col. 2, ll. 43-49. **<u>OMNI 300</u>** "VeriFone supports the Standard ANSI C and a UNIX-V7 compatible dialect of the C language (non-ANSI) for TXO application development." *OMNI 300*, p. 3-1. "VeriFone has maintained programming compatibility to enable you to port application source code from one terminal platform to another." *Id.*, at p. F-1, Table F-1, p. F-2 and p. F-7. **<u>OTA</u>** "Using the 'Open Terminal Architecture' (OTA) it will be possible for credit card issuers and acquirers to write application programs that will be completely platform independent, and run on all OTA-compliant kernels." *OTA*, p. 73, ¶1.

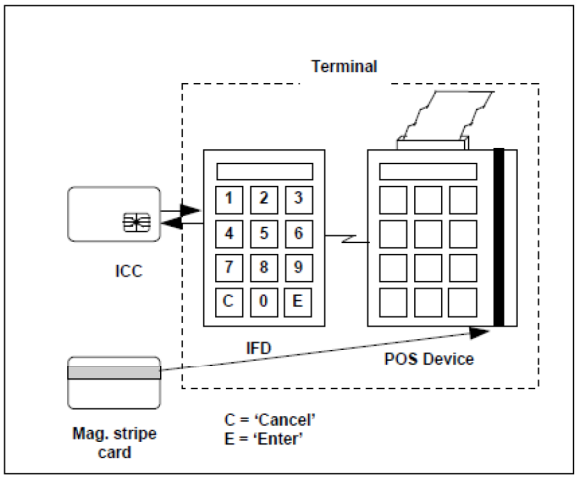
<u>Element F.</u> This element of '683 requires: first, the device be a payment terminal device; and second, that the virtual message processor communication with a peripheral units associated with said device. As examples of peripheral units, Ogilvy '683 mentions the examples of a "card reader, display, printer,

communications interface, etc." *'683 Patent*, col. 3, ll. 9-11. Such a communications interface is used "for communication with an account acquirer." *Id*., at col. 9, ll. 50-54.

Regarding the first requirement of element **[F]**, EMV '96 is explicitly directed to point of service terminals. Similarly, each of the other references of The First Data '879 Patent, OMNI 300, and OTA are also directed to financial transaction processing and, more specifically, payment terminals.

Regarding the second requirement of element **[F]**, EMV '96 shows that multiple devices, such as a magnetic stripe reader and a PIN pad can be connected to the terminal as peripherals. These peripherals communicate with the virtual machine implemented on the terminal, which is detailed in relation to element **[A]** of claim 1. As previously detailed in relation to element **[C]** of claim 1, EMV '96 discloses reception, transmission, and comparison of messages. Such handling of messages could be applied to one or more connected peripheral devices. Further, the First Data '879 Patent discloses that the communication processor (which is a software module) handles communication with network connections. Such network connections are a form of communications interface, as defined by Ogilvy '683. The First Data '879 Patent also discloses communication with a printer and GUI (via a security processor). It can be understood as a design choice to have the

printer communicate directly with the execution control processor rather than with the communication processor and to have the GUI interact with a security processor rather than via the communication processor. Referring to OMNI 300, communication with the peripherals of a keyboard and a display device are detailed. The use of such peripheral devices necessarily involves data being communicated between the device and peripheral. Whether a separate virtual message processor is used for such communication would be a design choice.

[F] wherein said device is a payment terminal device and wherein said virtual message processor is used to communicate with peripheral units associated with said device.	**EMV '96** "Terminals include but are not limited to automated teller machines (ATMs) . . . and point of service (POS) terminals." *EMV '96*, §1, p. vii. Figure I-1 illustrates an example of an attended terminal where the integrated circuit (IC) interface device (IFD) and PIN pad are integrated but separate from the POS device (such as for an electronic fund transfer terminal or an electronic cash register).  Figure I-1 - Example of an Attended Terminal *Id.*, at I-3. "If the terminal does not have a combined IC and magnetic stripe reader, when the magnetic stripe of

the card is read and the service code begins with a '2' or a '6' indicating that an IC is present, the terminal shall prompt for the card to be inserted into the IC reader such as by displaying the 'Use Chip Reader' message." *Id.*, at I-15.

"The terminal should be designed and constructed to facilitate the addition of a PIN pad, if not already present, such as having a serial port." *Id.*, at I-18.

Characteristics	Example 1
Physical:	
Key pad	Attendant key pad (numeric and function keys) + PIN pad
Display	One for attendant One for cardholder
Printer	Yes for attendant
Magnetic stripe reader	Yes
IC reader	Yes
Functional:	
Language selection	Supports part 1 of ISO 8859
Transaction type	Goods, cashback
Static data authentication	Yes
Cardholder verification	Offline PIN, signature
Card capture	No
Online capable	Yes
Offline capable	Yes

Table F-1 - Example of POS Terminal or Electronic Cash Register

EMV '96, at p. F-1.

The First Data '879 Patent

The First Data '879 Patent is directed to a "financial instrument processing system . . . ." *The First Data '879 Patent*, col. 2, ll. 42-45.

"The communications processor 20 allows the integrated system to communicate with other systems through network connections." *The First Data '879 Patent*, col. 4, ll. 6-8. "the communications processor 20 allows for communication with an integrated communications platform system to allow for session-based

	communications with a host computer." *Id*., at col. 4, ll. 9-12. "The communications processor 20 acts as an interface between the integrated system and whatever communication facilities are available via network connections." *Id*., at col. 4, ll. 32-35. "The security processor 12 interfaces directly with a graphical user interface 18 to monitor attempted accesses to various objects within the integrated system made by users of the graphical user interface 18." *Id*., at col. 3, ll. 50-53. Regarding other peripherals connected with the system: "a financial instrument processing system is provided that comprises a graphical user interface that is operable to communicate with a data base processing system." *Id*., at col. 1, ll. 50-53. "A printer is coupled to the print processing system and is operable to print financial instruments using information sent from the execution control processor." *Id*., at col. 1, ll. 57-59. "The printer 30 is capable of bidirectional communication through the print processor 28 to the execution control processor 10." *Id*., at col. 6, ll. 24-26. "The communication between the execution control processor 10 and the printer 30 through print processor 28 is a secure communications path." *Id*., at col. 10, ll. 30-32. **OMNI 300** "OMNI 300 Series terminals are 8-bit machines running the VeriFone® TXQ® operating system. These terminals are ideal for a multitude of applications, including: Point of Sale/Service (POS)" *OMNI 300*, p. 1-1. " System Devices – This chapter describes each of the system devices: keyboard[;] display[;] beeper[;] magnetic card reader[; and] real time clock/calendar[.] System devices are access in the

	same manner as files, using the same basic set of function calls . . . .” *OMNI 300*, p. 8-1. “When the user presses a function key, the system simply passes the key’s code to the application, which acts upon the key press according to the application design.” *OMNI 300*, p. 8-5. “All data written to the display is stored in a 32-character buffer. The display shows only 16 characters, alpha or numeric, at one time. . . . . Any character which the display device is unable to properly display will be shown as an underscore (ASCII Ox5F). To clear the display, send the form feed character (‘ \ f’ or 0x0C).” *Id*., at p. 8-11. **OTA** “The purpose of an OTA system is to provide software to run in terminals used in payment applications.” *OTA*, p. 73, ¶3.

Claim 2. Claim 2 of the ‘683 patent discloses that the terminal must be a payment terminal device. Such recitations are taught by the references as noted in reference to claim 1, element **[F]**. Claim 2 further requires that the virtual message processor be used for communication with an application associated with the device. As noted under the claim construction section above, this is construed under the ordinary meaning of the words to mean any application which runs on the device.

EMV ’96 discloses that code modules, which can be complete applications, can be maintained in an application library and are implemented via a virtual machine of

the terminal. The First Data '879 Patent specifies the concept of having a communication processor software module that is distinct from another processing module of the device. Further, the First Data '879 Patent discloses a user can interact with a data base processor, which is a piece of code being executed. Via the communication processor, which is a software module, transactions are uploaded from the data base processor. OMNI 300 discloses that communication buffers are used to pass information to and receive information from applications. Regarding OTA, the concept of an application executing on the device, the application being involved in the performance of a transaction is disclosed.

The '683 Patent	
2. A device in accordance with claim 1 wherein said device is a payment terminal device and wherein said virtual message processor is used to communication [sic – communicate] with an application associate [sic – associated] with said device.	**EMV '96** With either the API or the interpreter approach, the terminal should have the ability to maintain an application library of modules or routines that may be dynamically incorporated into the processing of a given transaction. Modules in the application library may be complete application programs, or they may be subroutines to be called upon at the direction of data within the terminal or the ICC. In the case of an interpreter capability, these modules will be code, written in a virtual machine instruction set implemented within the terminal, to be interpreted by the terminal control program. *EMV '96*, p. II-2. **The First Data '879 Patent** "The integrated system of the present invention allows the user to perform a variety of transactions which are logged in the data base of information managed by data base processor 14. Periodically, these transactions may

	be uploaded to a host computer or super server through the communications processor 20." *Id.*, at col. 4, ll. 19-24. **OMNI 300** "Communications buffers are used by the data communications device drivers (MODEM, RS232, PIN PAD). Each buffer is 256 bytes in length and can contain up to 254 bytes of data; two bytes are reserved-one for the count and one is unused. The maximum number of allocated buffers is 32, and the minimum number is 4. The environment variable *8 in CONFIG.SYS contains the number of buffers which will be allocated by the system. If *B does not exist, the minimum value of 4 will be used. Thus, the maximum amount of RAM which can be allocated for the communications buffer pool is 8k, with the minimum being l k." *OMNI 300*, p. 5-6. **OTA** "The software includes development software, which runs on the PC; kernels, which include all platform specific software in a terminal and other mandatory standard functions; libraries, which provide general functions to support terminal programs. and payment applications; applications, which are the functions specific to a particular payment product, and terminal programs, which perform general non-payment terminal functions and include high-level mechanisms for selecting and executing transactions and associated applications." *OTA*, p. 73 "An OTA development system is used to develop terminal software, either low-level kernel software or high-level library or application software." *Id.*, at p. 73 "A terminal transaction will select an application as part of its processing flow. Applications fall into three general areas: cashless 'purses' (Pay Before), debit cards (Pay Now) and credit cards (Pay Later);

	applications will generally vary in their method of processing a given transaction. Versions of these applications may be provided by different payment systems and further customized by individual issuers or acquirers. Applications are supplied in token form via the communications path, and (if security considerations permit) may be enhanced by token programs on an ICC." *Id.*, at p.75.

Claim 3. Claim 3 of the '683 patent requires that the communication means that implements the virtual message processor also implements "cryptographic series." As described under the claim construction section above, this is construed to mean that data transmitted to or from the terminal is encrypted.

EMV '96 indicates that unused cryptographic keys are stored, messages that are transmitted can involve a cryptogram request and that in a response message a cryptogram may be received. OTA discloses that the terminal can interact with an ICC having encrypted data, which can include the use of a cryptographic series, that can be used during validation of a transaction.

The '683 Patent	
3. A device in accordance with claim 1 wherein said wherein said virtual message processor is used to communicate means associated with the device for implementing cryptographic series.	**EMV '96** "a tamper-evident device, when not in use, shall contain no sensitive information except unused cryptographic keys." *EMV '96*, p. I-21. "If the transaction is forced online (by the terminal or the attendant), the

terminal shall not set the Authorisation Response Code and shall transmit an authorization or financial transaction request message using the Application Authorisation Referral (AAR) as an Authorisation Request Cryptogram (ARQC)." *Id.*, at p. I-13.

"If an Authorisation Response Cryptogram (ARPC) is present in the authorisation response message, the terminal may issue the EXTERNAL AUTHENTICATE command either before or after the referral data is manually entered." *Id.*, at p. I-14.

"Cryptogram – Result of a cryptographic operation." *Id.*, at p. x.

Data Element	Condition
Application Interchange Profile *	
Application Transaction Counter *	
ARQC *	
Cryptogram Information Data	
CVM Results	
IFD Serial Number	Present if Terminal Identifier does not implicitly refer to IFD Serial Number
Issuer Application Data *	Present if provided by ICC in GENERATE AC command response
Terminal Capabilities	
Terminal Type	
Terminal Verification Results *	
Unpredictable Number*	Present if input to application cryptogram calculation

Id., at III-7.

OTA

"One function of ICCs is to improve transaction security by incorporating and managing encrypted data and participating actively in the transaction

	validation process." *OTA*, at p. 75.

Claim 4. This claim of the '683 patent requires that the device be both personal and mobile. As described in the claim construction section above, "personal mobile" is construed under the ordinary meaning of the words to simply mean that the device can be moved by a person.

EMV '96 discloses that a terminal can be unattended, online or offline, and under operational control of the cardholder. The First Data '879 Patent discloses that the system can be a stand-alone unit incorporated into a personal computer. OMNI 300 discloses that the device is compact. Finally, OTA indicates that the device can be small and handheld, thus being both personal and mobile.

The '683 Patent	
4. A new device in accordance with claim 1 wherein said device is a personal mobile device.	**EMV '96** "as described in the scope, this specification addresses a broad spectrum of terminals. For the purpose of this specification, terminals are categorized by the following:" • Environment: Attended or unattended • Communication: Online or offline • Operational control: Financial institution, merchant, or cardholder *EMV '96*, p. I-1, §1.1. **The First Data '879 Patent** "The integrated system described with reference to FIG. 1 may be constructed as a stand-alone unit with all components of the system residing, for example, in a personal computer coupled to the printer 30. In the alternative, the system of the

	present invention may be implemented in a network environment. In this environment, various processors described may be physically located in dedicated servers. For example, the graphical user interface 18 might be located in a user node While the execution control processor 10, the grammar files 24, the parameter files 26 and the help processor 22 might be resident in a file server coupled to the user node through the network. Under this implementation, the print processor 28 and the communication processor 20 may be located in dedicated servers as Well. The system of the present invention is constructed as an object-oriented integrated system to allow for portability between a variety of both stand-alone and network-based platforms." *The First Data '879 Patent*, col. 7, ll. 6-22. **OMNI 300** "The OMNI™ 300 Series is a family of dial-type and LANtype transaction automation systems that are compact and efficient microcomputers capable of gathering and transferring information at high speed." *OMNI 300*, p. 1-1. **OTA** "The purpose of an OTA system is to provide software to run in terminals used in payment applications." *OTA*, p. 73. "The target system is anyone of a large variety of payment terminals. Actual products range from small, hand-held devices with simple 8-bit microprocessors such as the 8031/51 family to 32-bit computers running operating systems such as Windows-NT or Unix. In order to simplify the production, certification and maintenance of software on such a wide variety of targets, OTA terminal code is based on a single virtual machine which is emulated on the actual devices. Prototype coding and testing has shown that this approach is feasible and provides good run-time performance, even on an 8051 CPU." *Id.*, at p. 74.

Claim 5. This claim of the '683 patent requires that the virtual message processor implement secure communications. As described under the claim construction section above, this element is construed to simply mean data transmitted to or from the terminal is encrypted.

As detailed in relation to claim 3, EMV '96 discloses the transmission and receipt of cryptograms. The exchange of such cryptographs can be understood as a form of secure communications. Further, the terminal of OTA discloses that encrypted data involving an ICC (integrated circuit card) of a credit card exchanging encrypted data and participating in the transaction validation process with a host system. As such, by transmitting encrypted data, secure communications would be implemented.

The '683 Patent	
5. A device in accordance with claim 2 wherein said virtual message processor implements secure communication services.	See claim 3.