PC/SC Workgroup

- Formed in May '96
- Founding Members:
  - Microsoft
  - Bull CP8 Transac
  - Hewlett-Packard
  - Schlumberger
  - Siemens Nixdorf Information Systems

Goals

- Address need for PC-ICC interoperability
  - Interfaces to IFDs
  - Common programming interfaces and control mechanisms
  - Compatibility with existing devices
- Develop solutions meeting broad industry needs

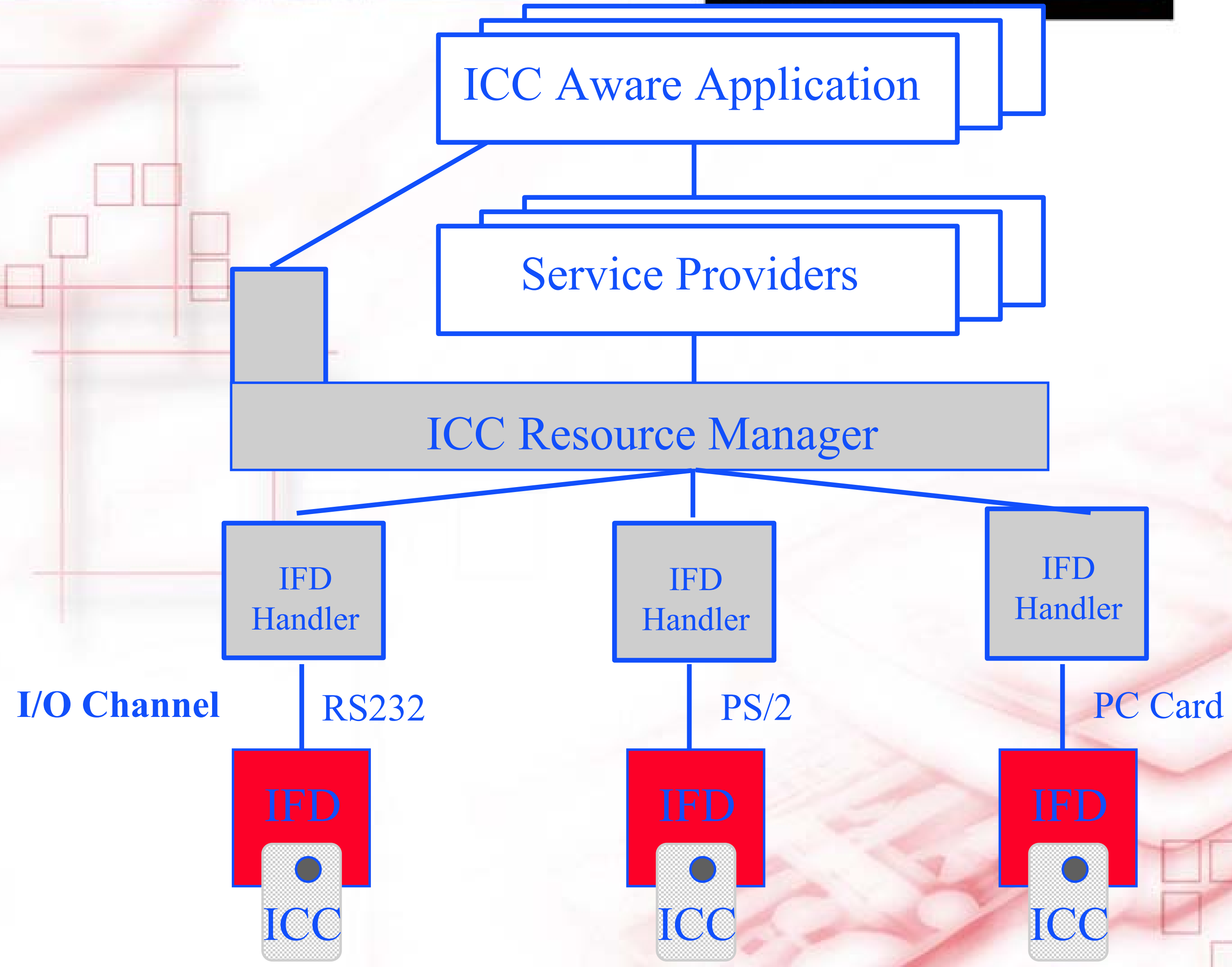
Workgroup Objectives

- Define comprehensive solution
  - Device compatibility requirements
  - Standard IFD interfaces
  - High level interface abstracts card services
  - Proposal for crypto and storage services
- Application and vendor neutral
- Deliver as proposed standards

Architecture

- ICC devices are accessed by PC-based applications through an IFD
- Can have multiple IFDs and a varieties of I/O channels e.g. RS-232C, USB, PS/2, PCMCIA
- Associated IFD is an IFD Handler (device driver)
- ICC Resource Manager provide system level service
  - Manages the ICC and IFD resources
  - Controls shared access to these devices
  - Supports transaction management primitives

Architecture



Prior To PC/SC Standard

- No standard to PC-reader communication protocol
- No standard to reader vendor API
- Application is locked to a particular reader vendor
- Cannot switch reader vendor without application software modification
- Reader is very expensive

End Result : Application cannot take-off

PC/SC Standard

- ◆Standard model for interfacing smart card readers and cards with PCs
- ◆Microsoft Smart Card Component req'd in Windows 95, 98, standard in Windows 2000
- ◆PC/SC reader vendor supply PC/SC driver, interfaced to operating system
- ◆Application access smart card and reader via reader vendor independent API

Implication Of PC/SC Standard

- ◆New PC will be equipped with PC/SC reader as a standard option
  - ☞Floppy mount smart card reader
  - ☞Keyboard smart card reader
- ◆Existing PC can be equip with external smart card reader
- ◆Low cost reader
- ◆Wide-spread smart card applications
  - ☞PC access control
  - ☞Electronic ID / Electronic Commerce
  - ☞Software Intellectual Proprietary Protection

PC/SC API defines.h

BYTE unsigned char
USHORT unsigned short
ULONG unsigned long
BOOL short
DWORD unsigned long
WORD unsigned long
LONG long
RESPONSECODE long
LPCSTR const char *
LPSTR char *
LPCWSTR char *

SCARDCONTEXT unsigned long
PSCARDCONTEXT unsigned long *
LPSCARDCONTEXT unsigned long *
SCARDHANDLE unsigned long
PSCARDHANDLE unsigned long *
LPSCARDHANDLE unsigned long *
LPCVOID const void *
LPVOID void *
LPCBYTE const unsigned char *
LPBYTE unsigned char *
LPDWORD unsigned long *

PC/SC - error messages

SCARD_E_NOTIMPL
SCARD_E_INVALID_HANDLE
SCARD_E_INVALID_TARGET
SCARD_F_COMM_ERROR
SCARD_E_UNKNOWN_CARD
SCARD_W_REMOVED_CARD
SCARD_E_NO_SMARTCARD
SCARD_E_PROTO_MISMATCH
SCARD_E_PCI_TOO_SMALL
SCARD_E_NO_SERVICE

SCARD_E_UNSUPPORTED_INTERFACE
SCARD_E_INSUFFICIENT_BUFFER
SCARD_E_UNKNOWN_READER
SCARD_E_SHARING_VIOLATION
SCARD_E_SYSTEM_CANCELLED
SCARD_E_READER_UNAVAILABLE
SCARD_W_UNSUPPORTED_CARD
SCARD_W_UNPOWERED_CARD
SCARD_E_UNKNOWN_READER
SCARD_E_DUPLICATE_READER

PC/SC - error messages

SCARD_E_INVALID_ATR
SCARD_E_INVALID_VALUE
SCARD_F_INTERNAL_ERROR
SCARD_E_NO_SMARTCARD
SCARD_E_NOT_READY
SCARD_W_RESET_CARD
SCARD_W_INSERTED_CARD
SCARD_E_UNKNOWN_CARD
SCARD_E_TIMEOUT

SCARD_E_UNSUPPORTED_FEATURE
SCARD_E_UNSUPPORTED_FUNCTION
SCARD_E_INVALID_PARAMETER
SCARD_E_NOT_TRANSACTED
SCARD_F_UNKNOWN_ERROR
SCARD_W_UNRESPONSIVE_CARD
SCARD_E_SYSTEM_CANCELLED
SCARD_E_READER_UNSUPPORTED
SCARD_E_CARD_UNSUPPORTED
SCARD_E_SERVICE_STOPPED

PC/SC API - SCardEstablishContext

SCardEstablishContext(DWORD dwScope, LPCVOID pvReserved1, LPCVOID pvReserved2, LPSCARDCONTEXT phContext)

- Creates a communication context to the PC/SC Resource Manager
- Must be first function called

PC/SC API - SCardReleaseContext

```
rv = SCardReleaseContext(hContext);
```

- Destroy a communication context to the PC/SC Resource Manager
- Must be the last function called

PC/SC API - SCardListReaders

LONG SCardListReaders(SCARDCONTEXThContext, LPCSTR szGroups, LPSTR mszReaders, LPDWORD pcchReaders);

- Returns a list of currently available readers mszReaders is a pointer to a character string
- If the application sends mszGroups and mszReaders as NULL then this function will return the size of the buffer needed to allocate in pcchReaders.
- The reader names will be a multi-string and separated by a NULL character and ended by a double NULL e.g. “ReaderA\0ReaderB\0\0”

PC/SC API - SCardConnect

```
LONG SCardConnect( SCARDCONTEXT hContext,
LPCSTR szReader, DWORD dwShareMode, DWORD
dwPreferredProtocols, LPSCARDHANDLE phCard,
LPDWORD pdwActiveProtocol );
```

- This function establishes a connection to the reader name specified in szReader
- The first connection will power up and perform a reset on the card

PC/SC API - SCardDisconnect

```
LONG SCardDisconnect( SCARDHANDLE hCard,
DWORD dwDisposition );
```

- This function terminates a connection to the connection made through SCardConnect

PC/SC API - SCardBeginTransaction

LONG SCardBeginTransaction(SCARDHANDLE hCard);

- Establishes a temporary exclusive access mode for doing a series of commands or transaction
- Can be used when selecting a few files and then writing a large file to ensure another application will not change the current file
- If another application has a lock on this reader or this application is in SCARD_SHARE_EXCLUSIVE there will be no action taken.

PC/SC API - SCardEndTransaction

LONG SCardEndTransaction(SCARDHANDLE hCard, DWORD dwDisposition);

- This function ends a previously begun transaction
- The calling application must be the owner of the previously begun transaction or an error will occur

PC/SC API - ScardTransmit

LONG SCardTransmit(SCARDHANDLE hCard, LPCSCARD_IO_REQUEST pioSendPci,
LPCBYTE pbSendBuffer, DWORD cbSendLength,
LPSCARD_IO_REQUEST pioRecvPci,
LPBYTE pbRecvBuffer, LPDWORD pcbRecvLength);

- Sends an APDU to the smartcard
- Responds from the APDU stores in pbRecvBuffer
- Length of response in pcbRecvLength
- SendPci and RecvPci are structures :

```
typedef struct {
DWORD dwProtocol; /* SCARD_PROTOCOL_T0 or
SCARD_PROTOCOL_T1 */
DWORD cbPciLength; /* Length of this structure – not used */
} SCARD_IO_REQUEST;
```

PC/SC API : SCardStatus

LONG SCardStatus(SCARDHANDLE hCard, LPSTR szReaderName, LPDWORD pcchReaderLen, LPDWORD pdwState, LPDWORD pdwProtocol, LPBYTE pbAtr, LPDWORD pcbAtrLen);

- Returns the current status of the reader
  - Reader Name stored in szReaderName
  - pcchReaderLen - size of buffer for szReaderName
  - pdwState - current state
  - pdwProtocol – protocol

PC/SC API - SCardGetStatusChange

LONG SCardGetStatusChange(SCARDCONTEXT Context, DWORD dwTimeout, PSCARD_READERSTATE rgReaderStates, DWORD cReaders);

- This function blocks for a change in state to occur on any of the OR ‘d values contained in dwCurrentState for a maximum blocking time of dwTimeout or forever for a specified reader
- The new event state will be contained in dwEventState
- A status change might be a card insertion or removal event, a change in ATR, etc.

PC/SC API - SCardCancel

```
LONG SCardCancel( SCARDCONTEXT hContext );
```

- This function cancels all pending blocking requests on the GetStatusChange function.