

# Using S/PAY<sup>tm</sup>

Dr. Robert W. Baldwin  
RSA Data Security, Inc.  
[baldwin@rsa.com](mailto:baldwin@rsa.com)

RSA Data Security Conference  
January 30, 1997



# Topics

---

- Introduction to SET
- What is S/PAY?
- Using S/PAY
- Customizing S/PAY



# Goals of SET

- The Internet Payment Card Protocol
- World-Wide Deployment
- Authentication & Privacy
- Reduce Merchant Fraud



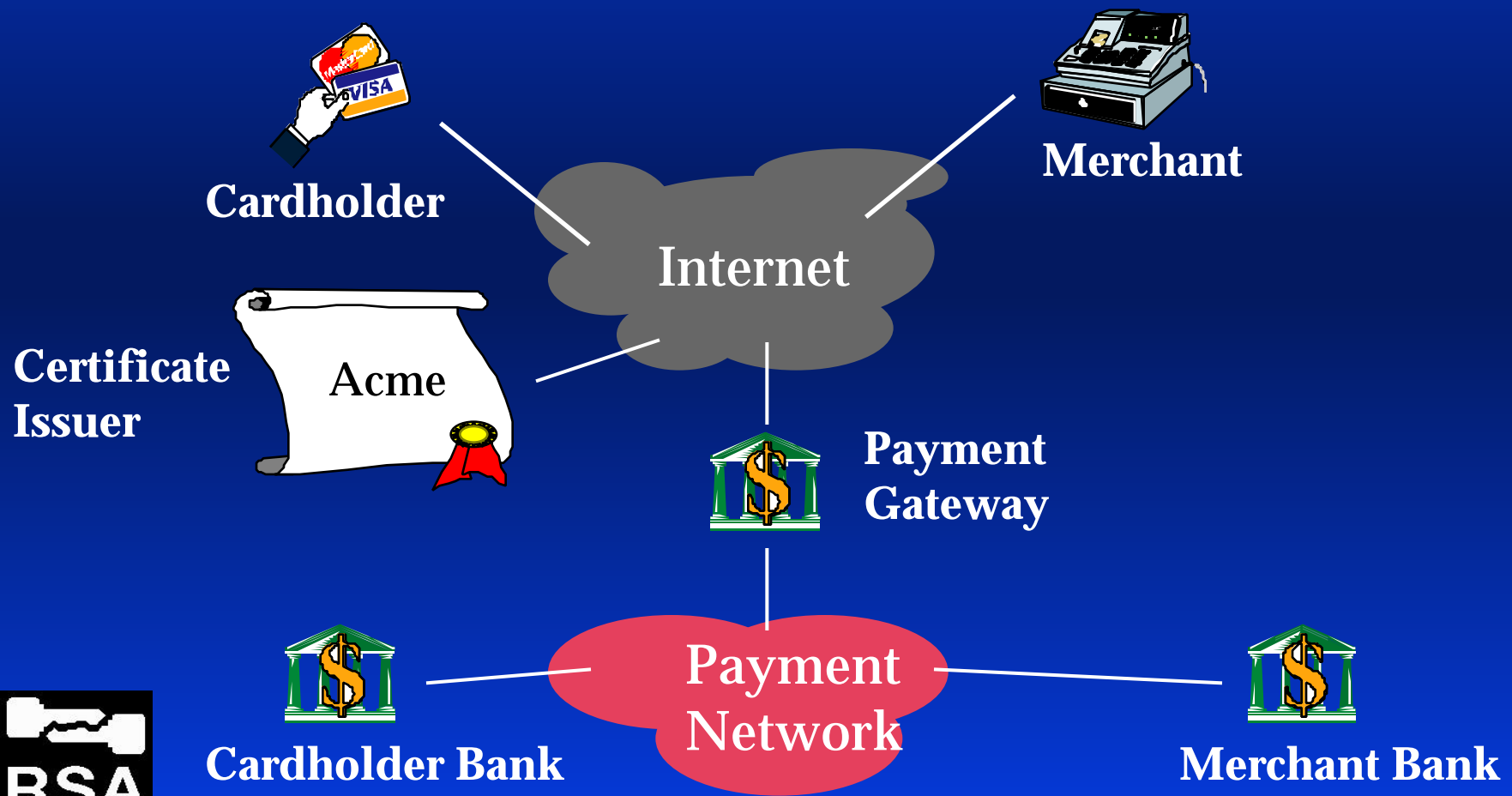
# SET Does Not Cover

---

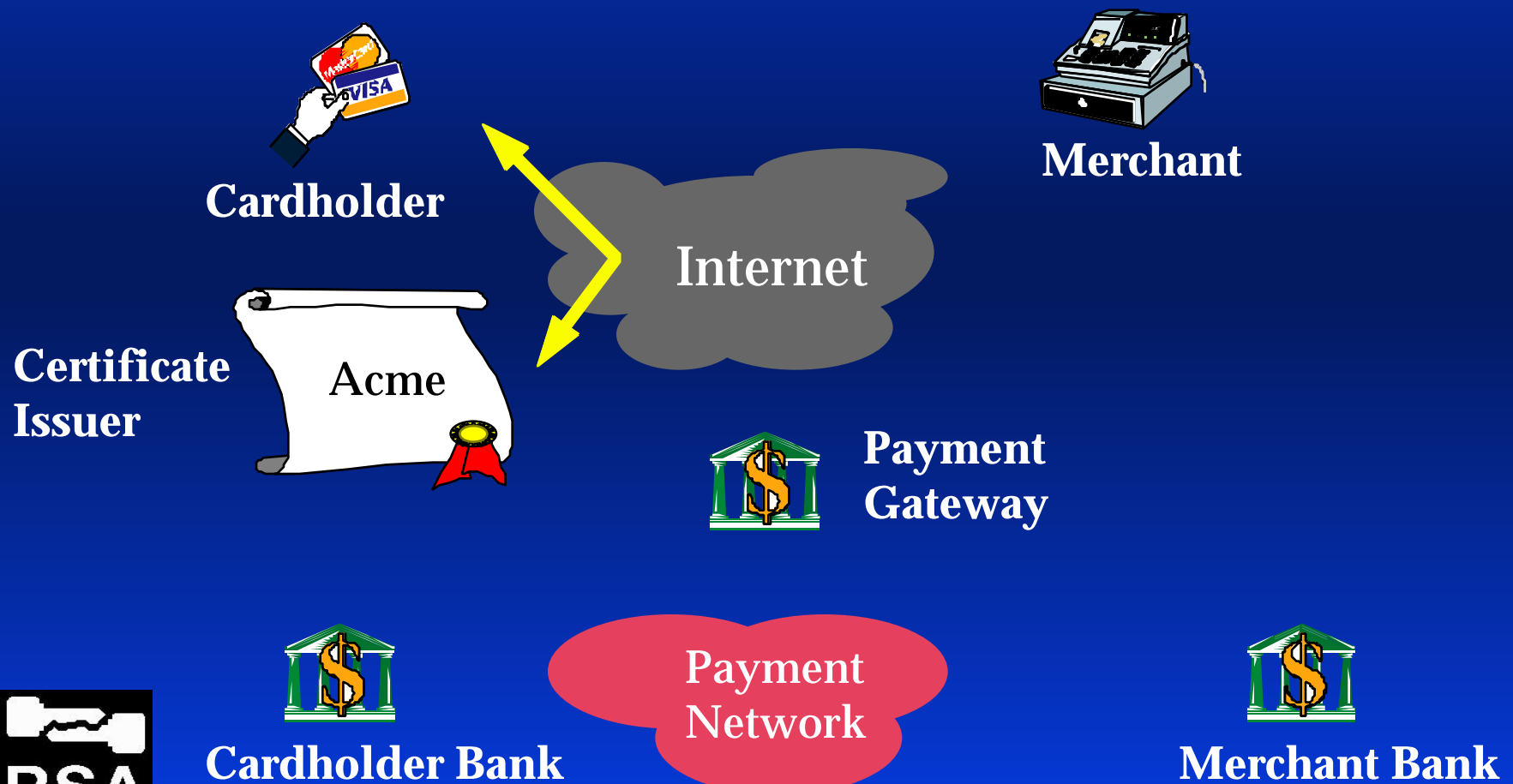
- Shopping Protocol
- Merchant Financial Settlement
- Home banking
- Electronic Checks



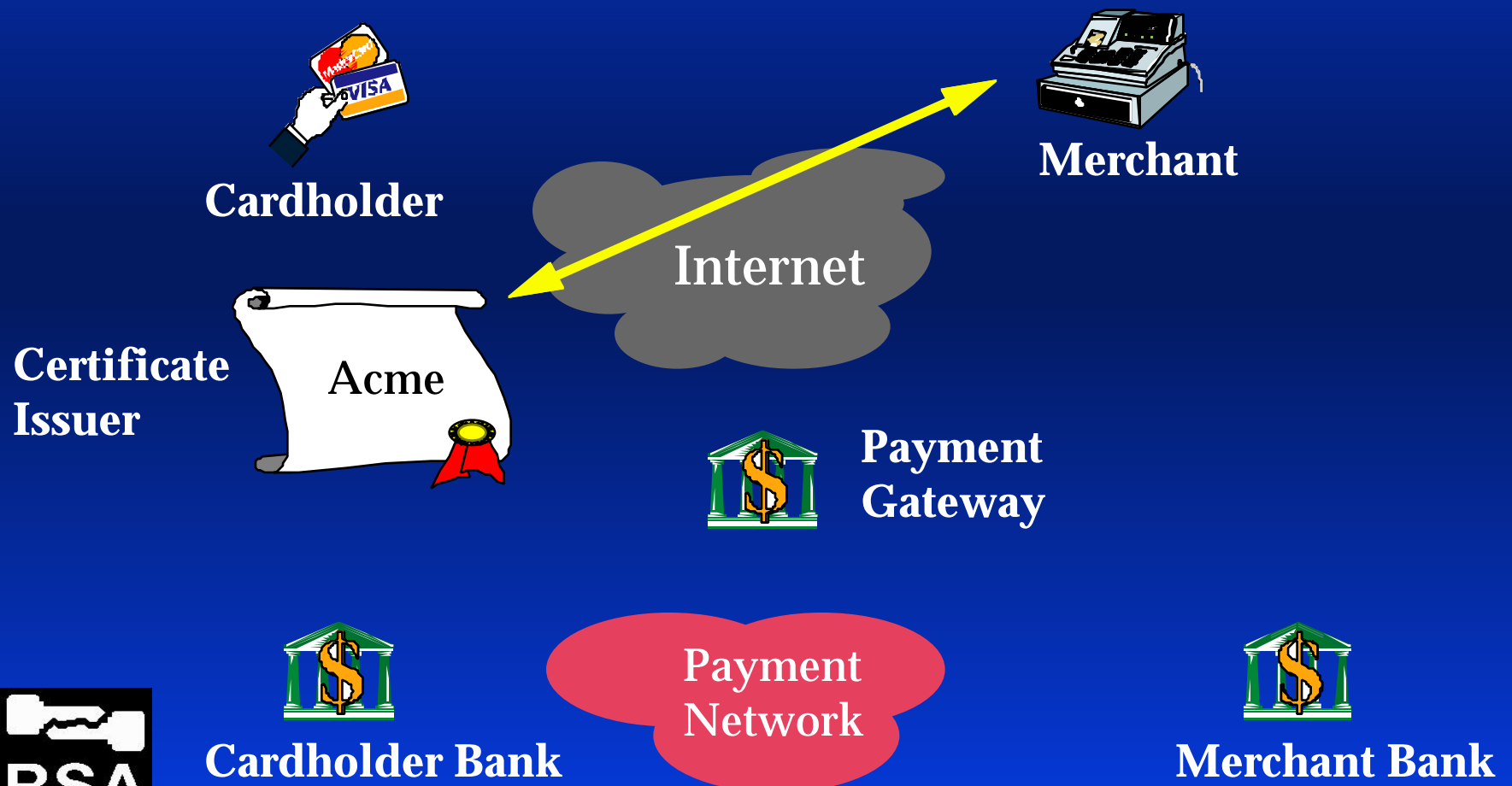
# SET Entities



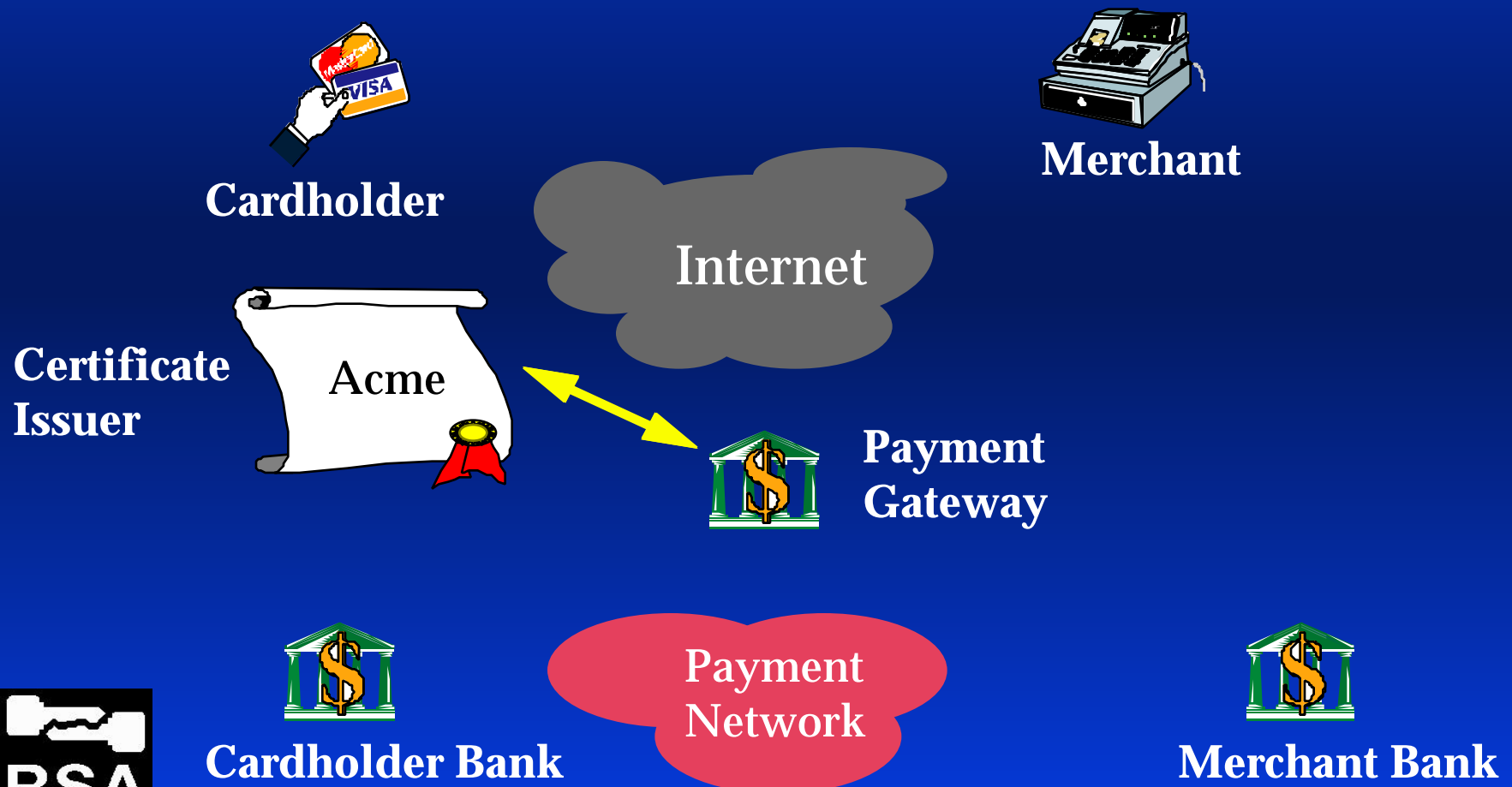
# SET Certificate Issuing



# SET Certificate Issuing



# SET Certificate Issuing





# SET Certificates Done



**Payment  
Gateway**



**Cardholder Bank**

**Payment  
Network**

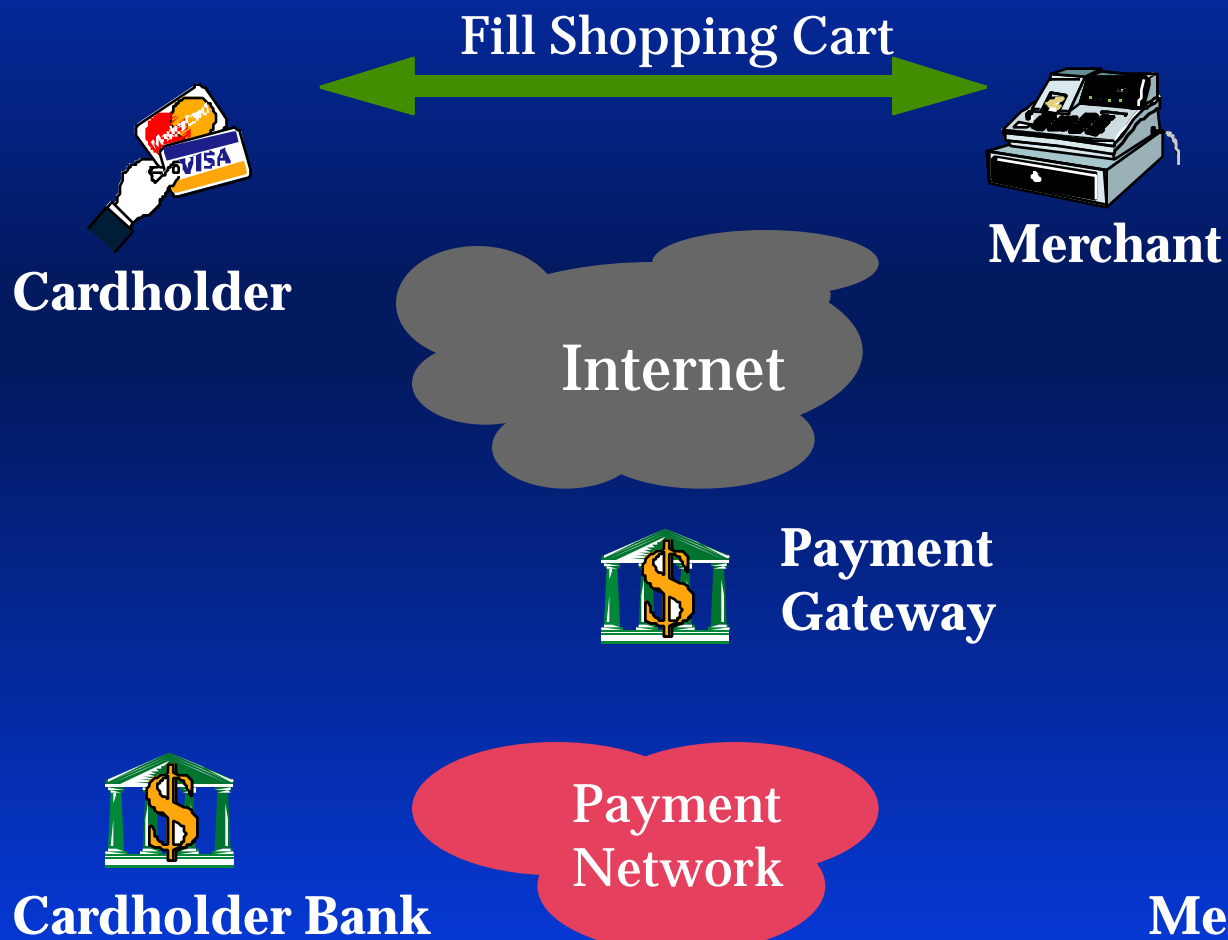


**Merchant Bank**

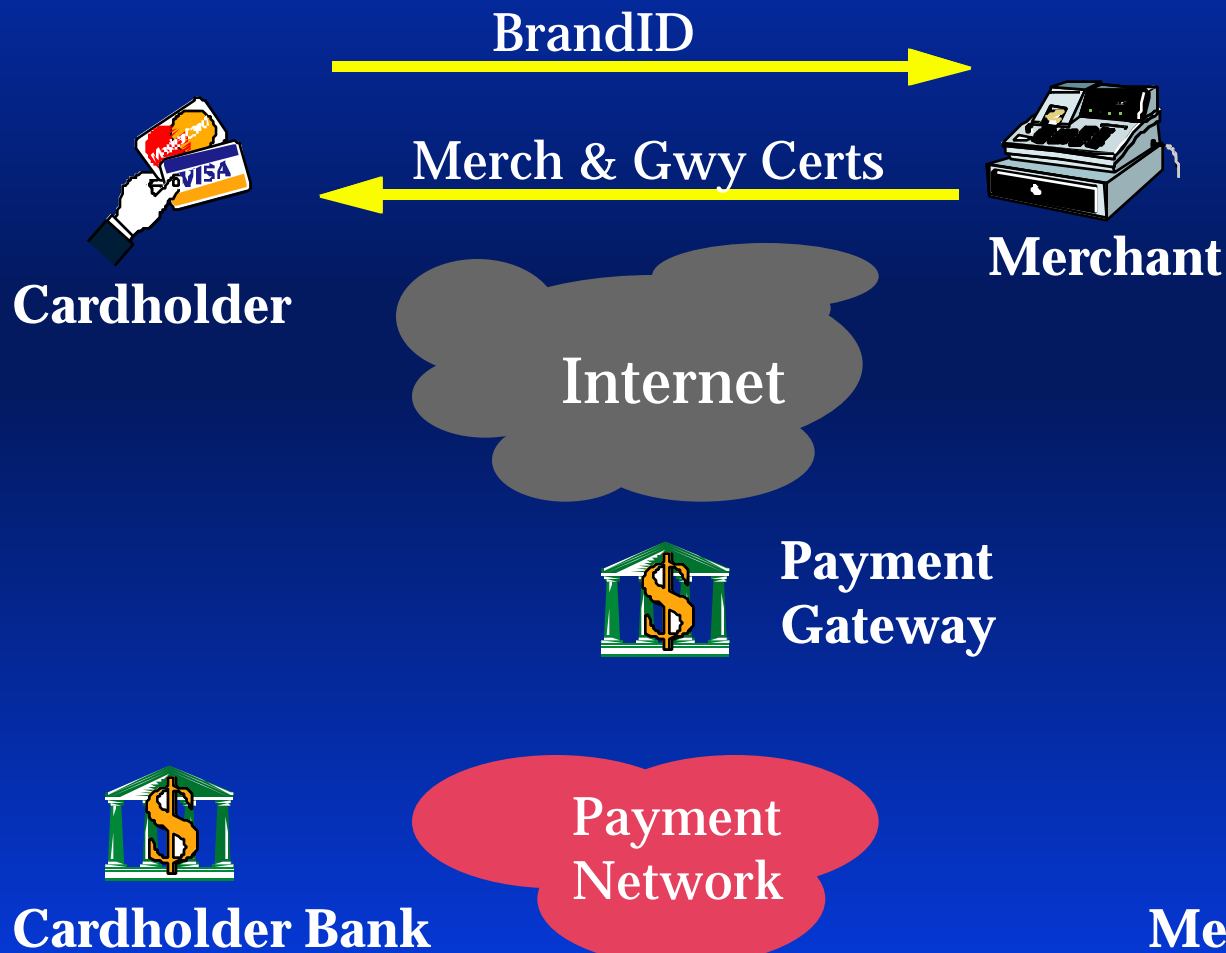


Copyright 1996, 1997 RSA Data Security, Inc. All rights reserved.

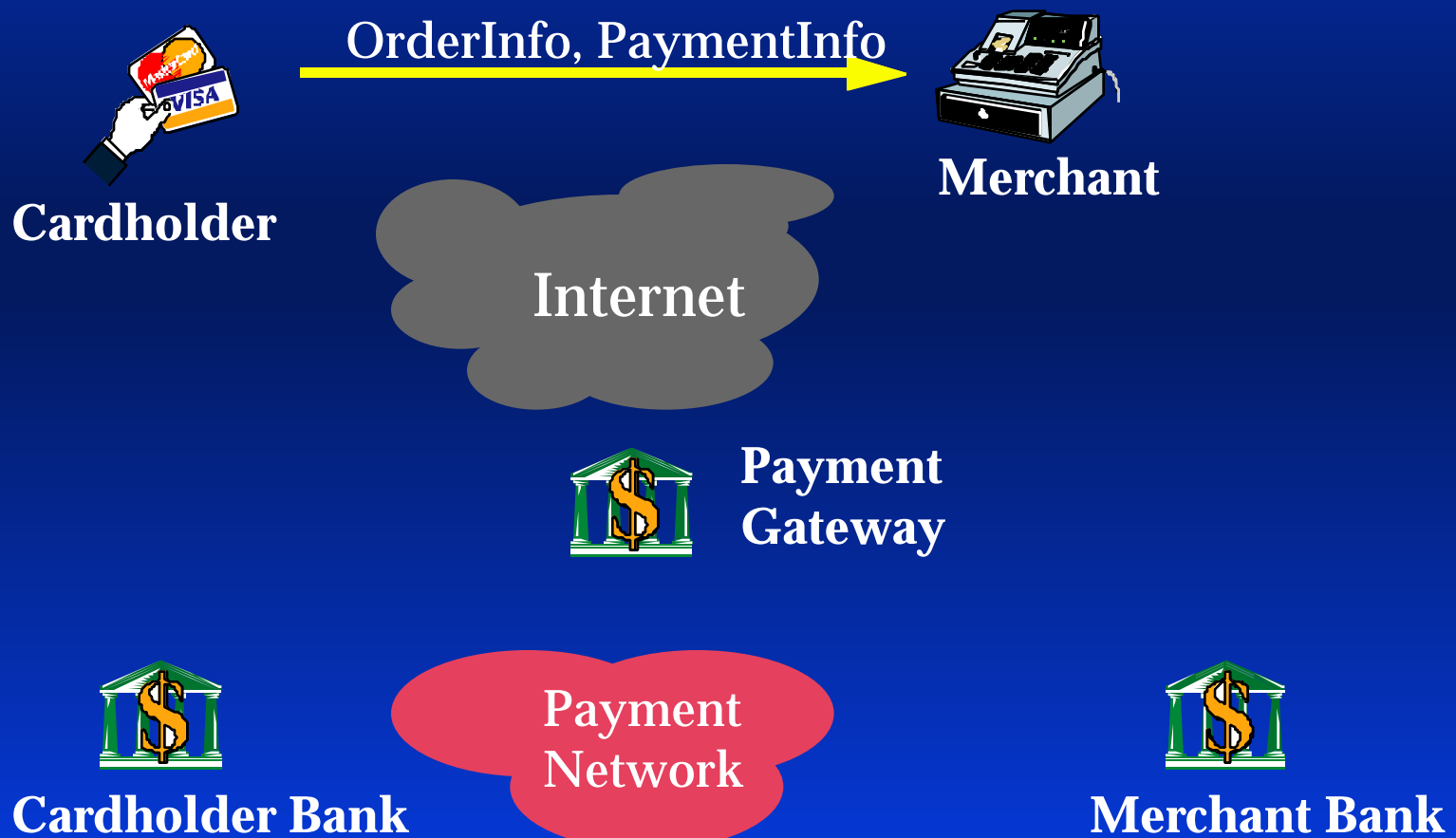
# SET Shopping



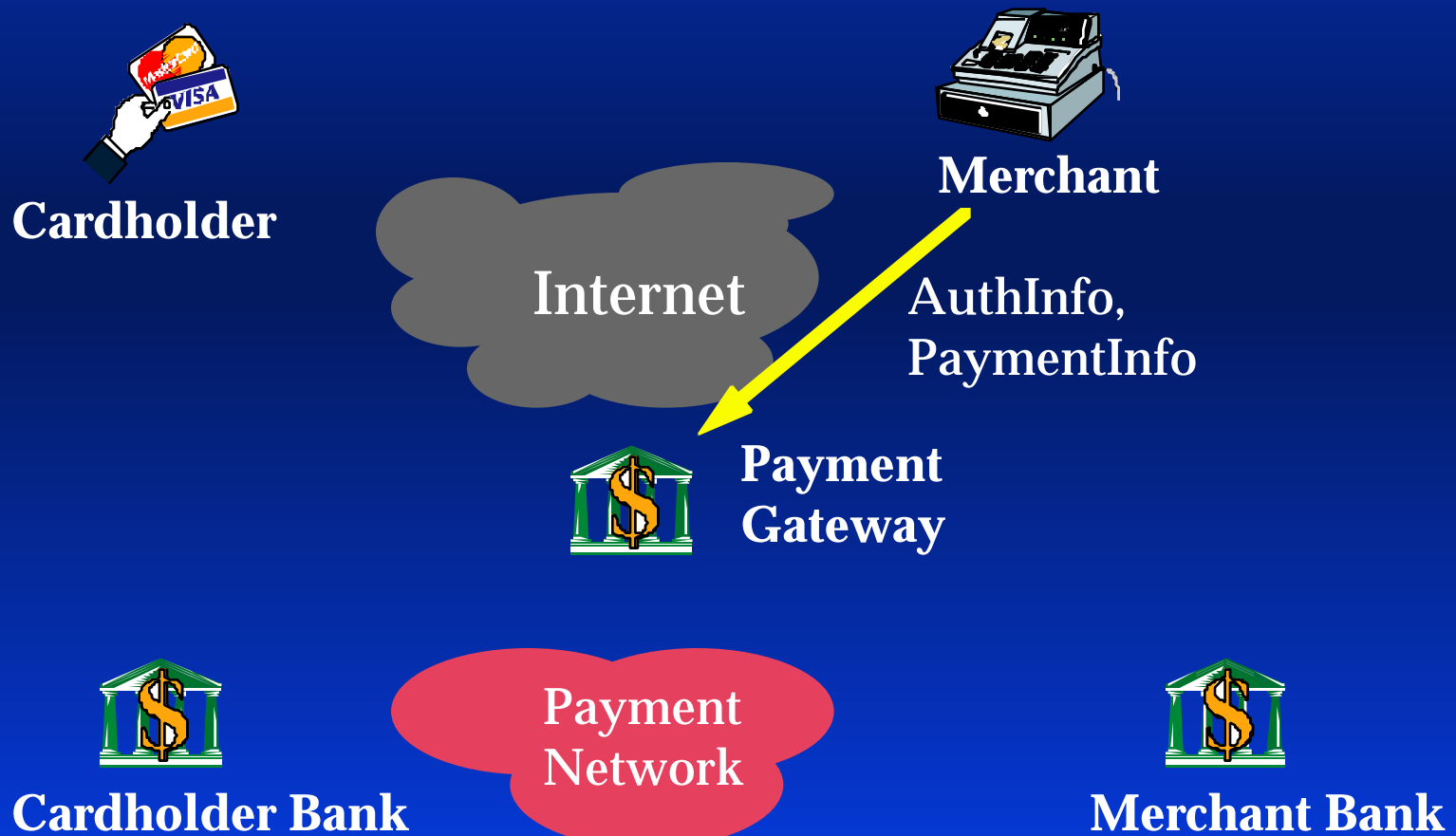
# SET Purchase Init



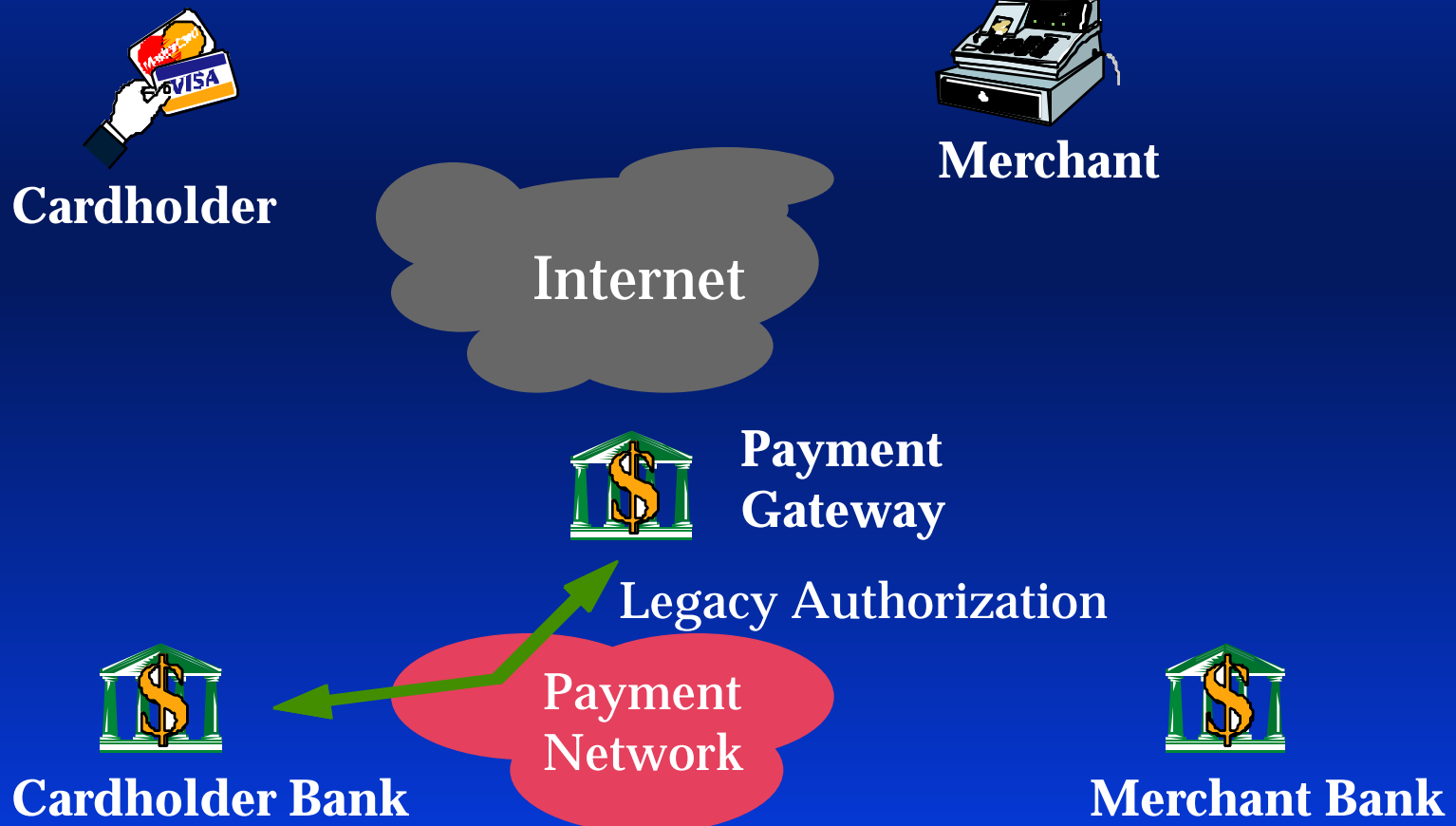
# SET Purchase Request



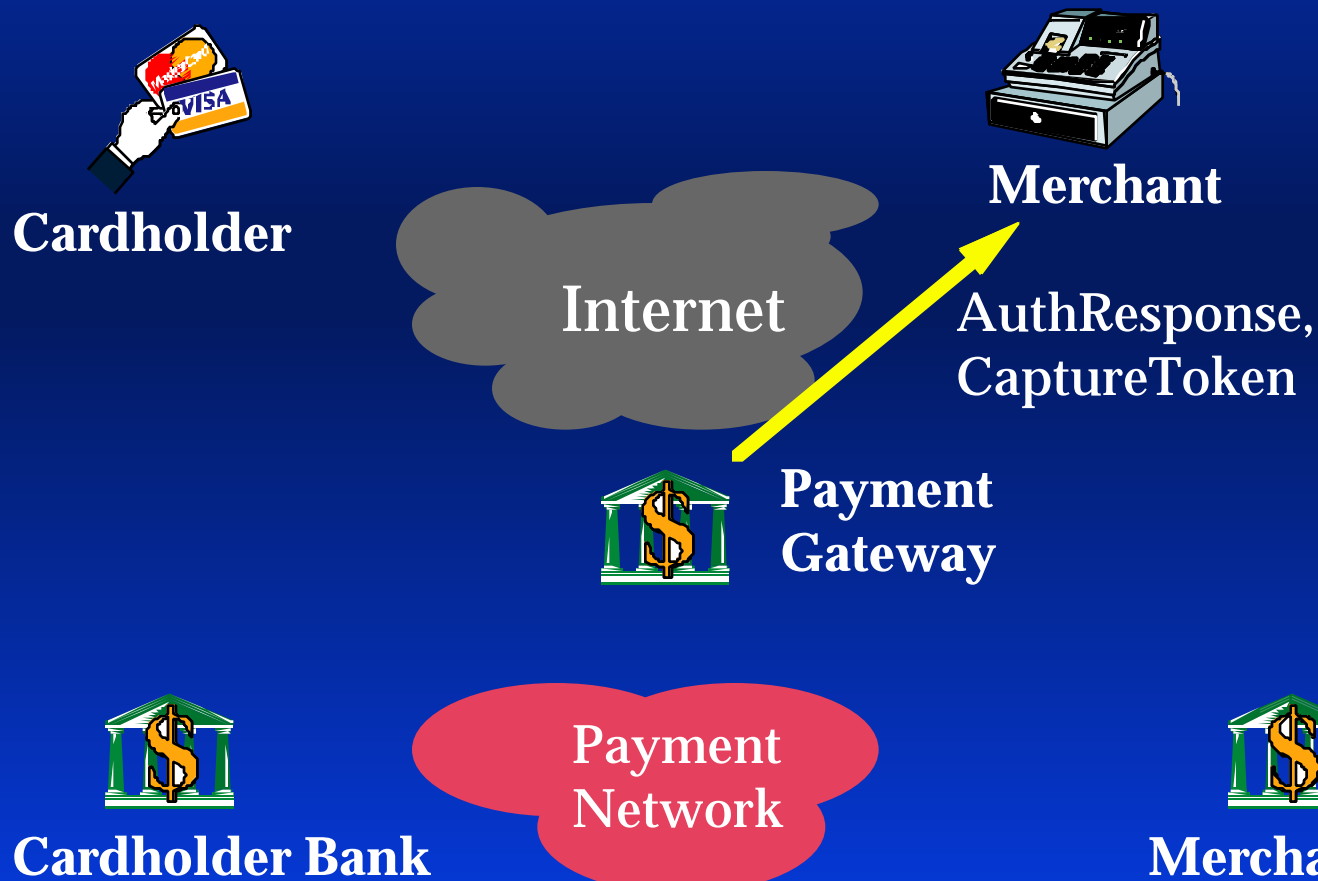
# SET Authorization Request



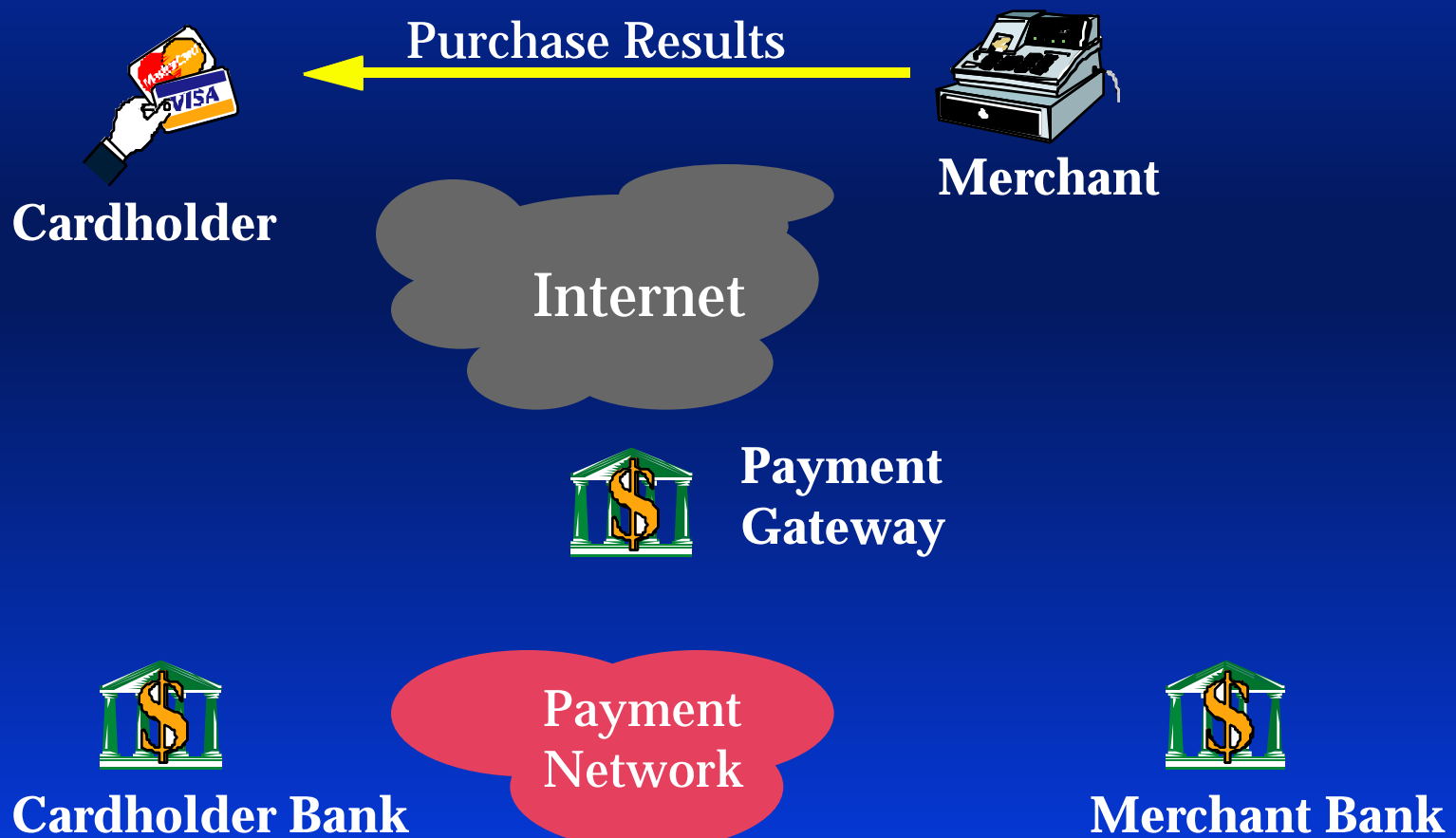
# SET Legacy Auth



# SET Authorization Response

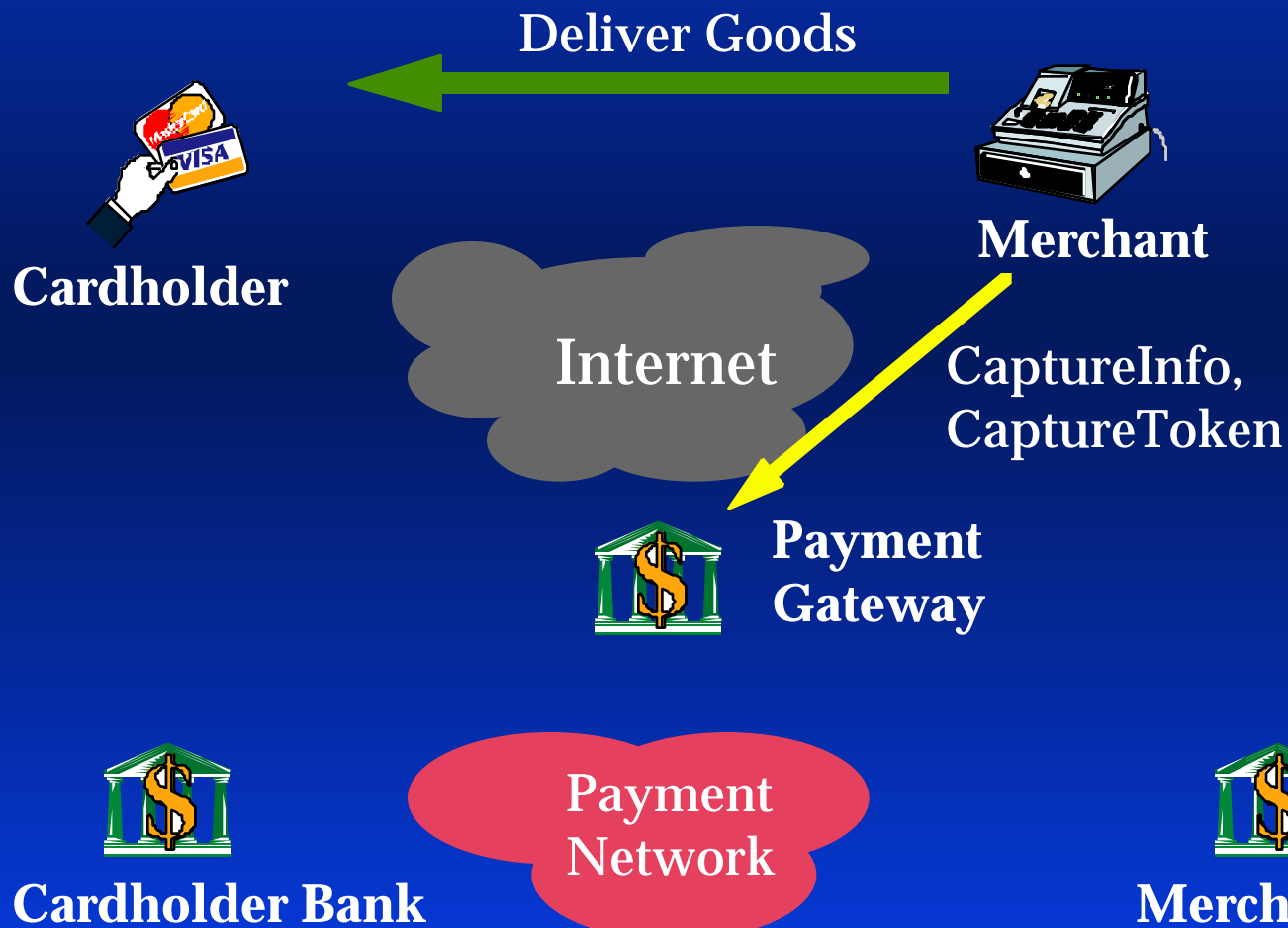


# SET Purchase Response

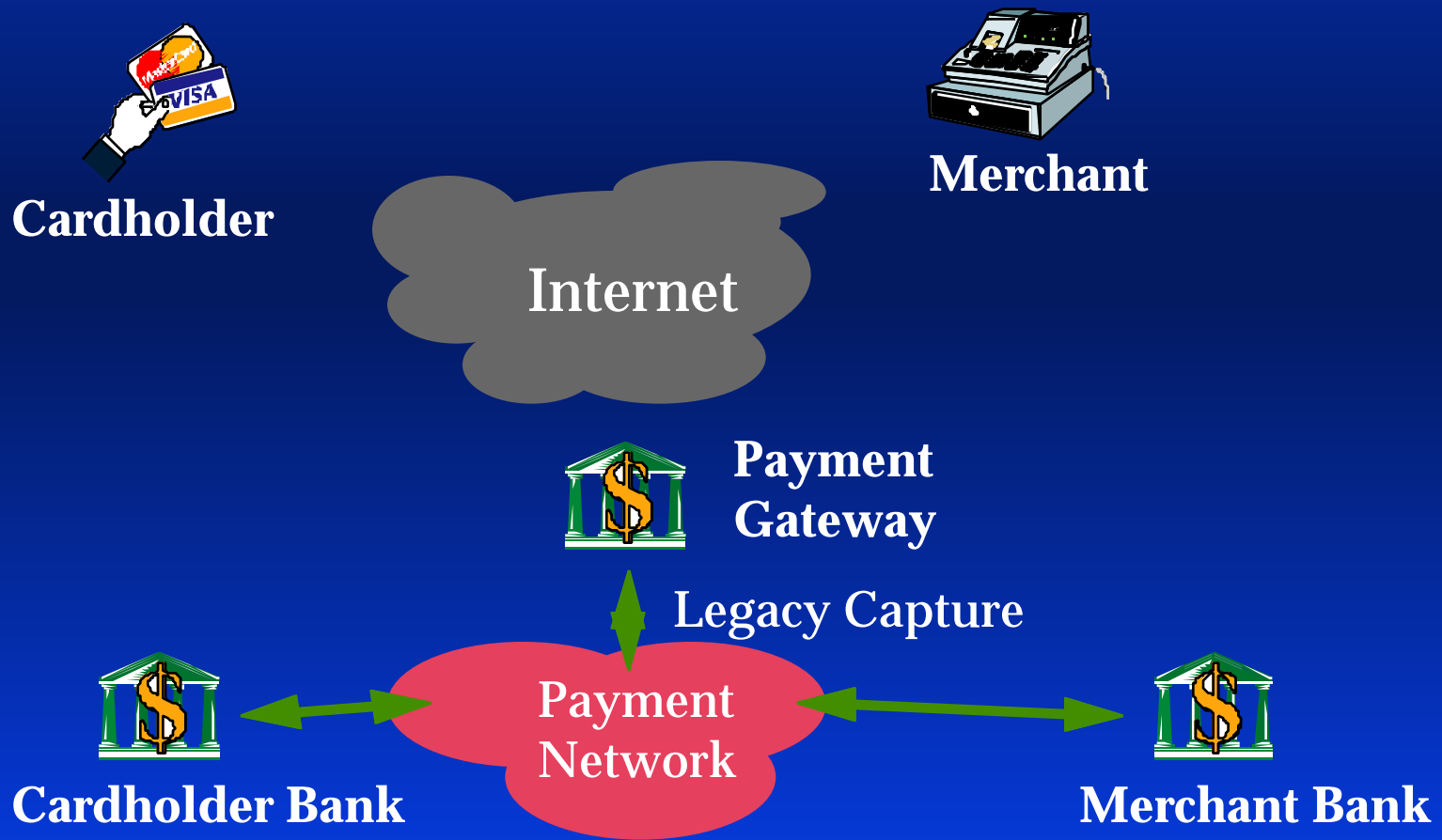




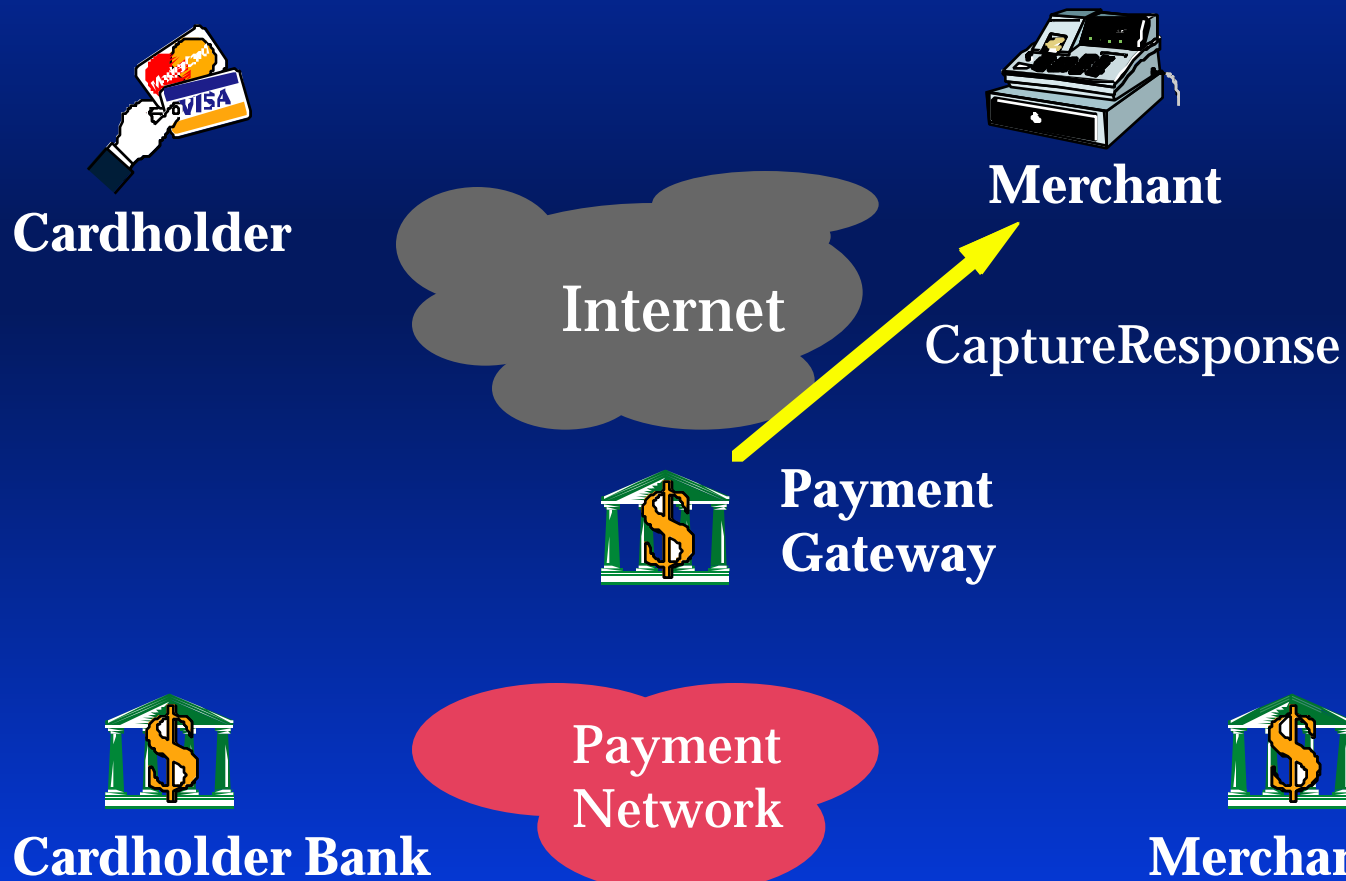
# SET Capture Request



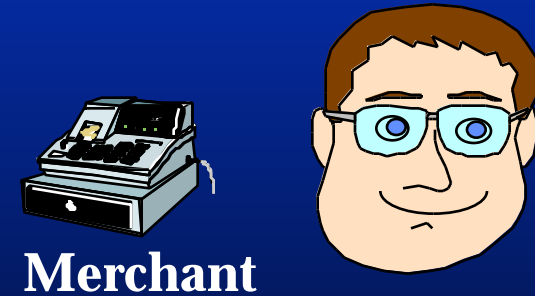
# SET Legacy Capture



# SET Capture Response



# SET Purchase Done



Payment  
Gateway



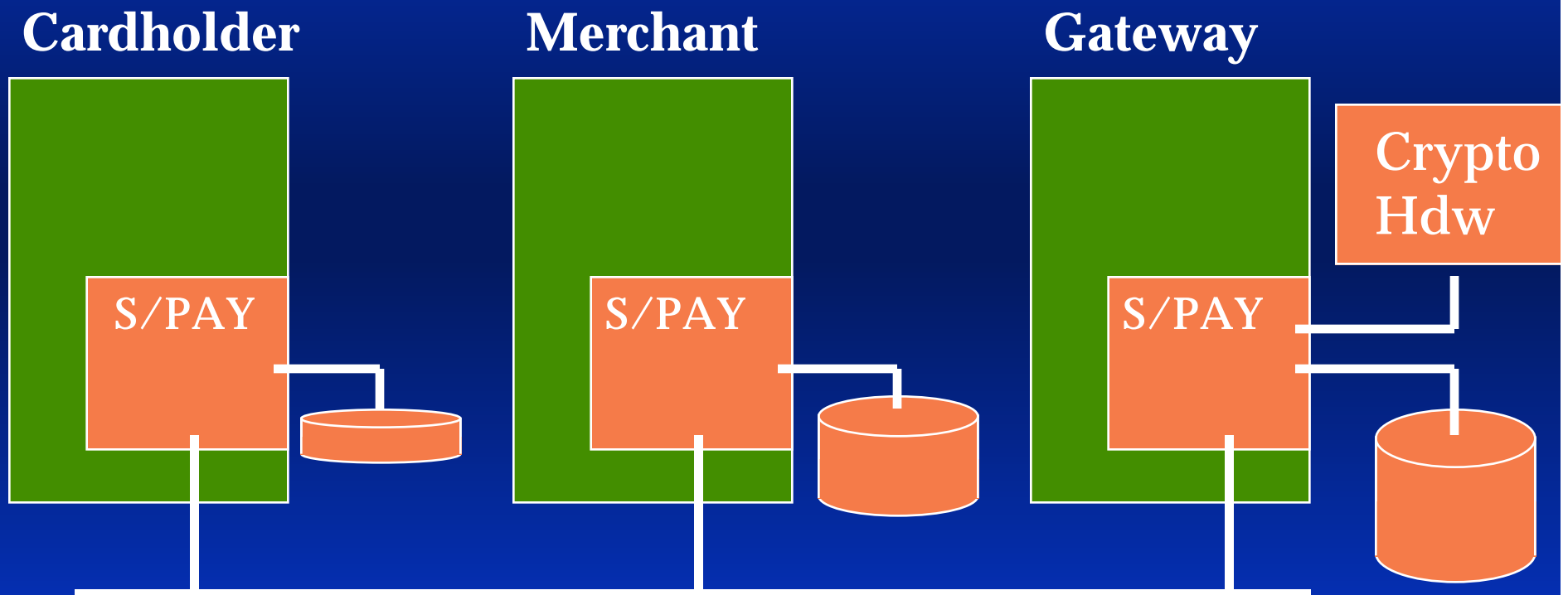
Payment  
Network

# S/PAY SET Engines

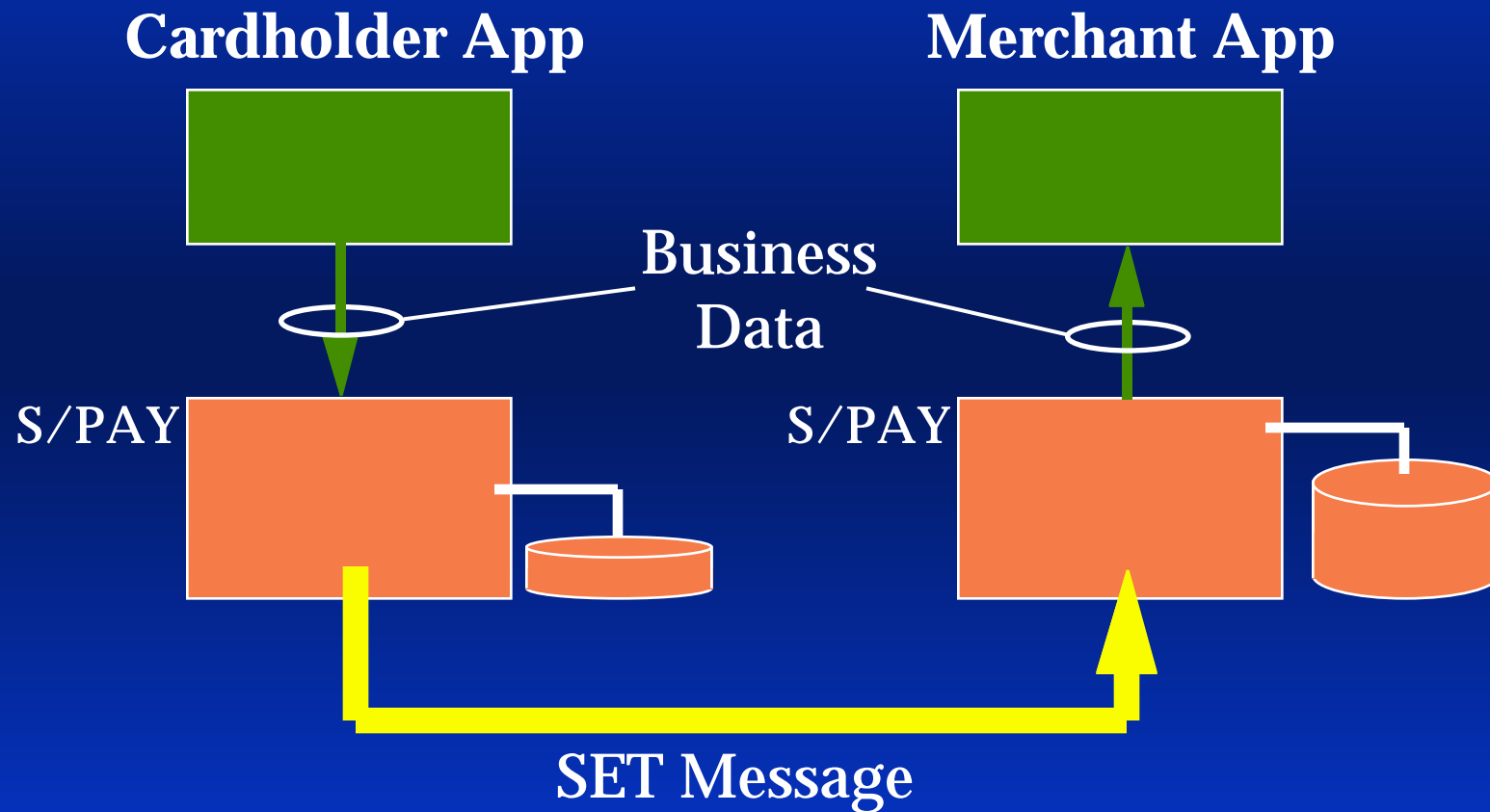
- Complete protocol engines
- SET, the whole SET, nothing but the SET
- From Set-Top Box to Mainframe



# S/PAY<sup>tm</sup> Enabled Applications



# S/PAY As Protocol Stack



# S/PAY SET Engines

---

- Flexible and Easy to Use
- Reduces time to market
- Quickly track SET changes





# S/PAY Supports:

---

- Multiple wallets on the cardholder
- Multiple merchants per server
- Multiple gateways & acquirers per server
- CD-ROM & Internet shopping



# S/PAY Supports:

---

- C and C++ interface
- Easy to glue to JAVA or ActiveX
- Multi-threading & multi-processing



# S/PAY Supports:

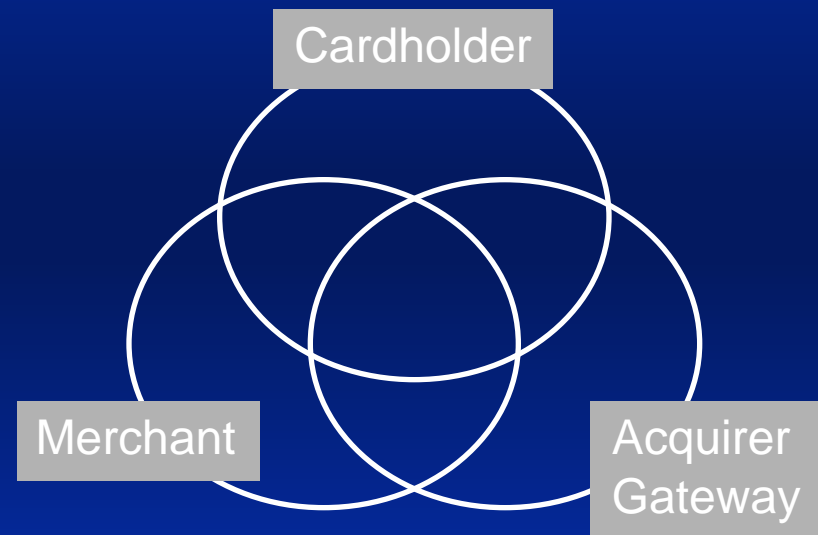
---

- Wide range of platforms
- Replaceable modules for DB & Comm.
- Callbacks for business processing



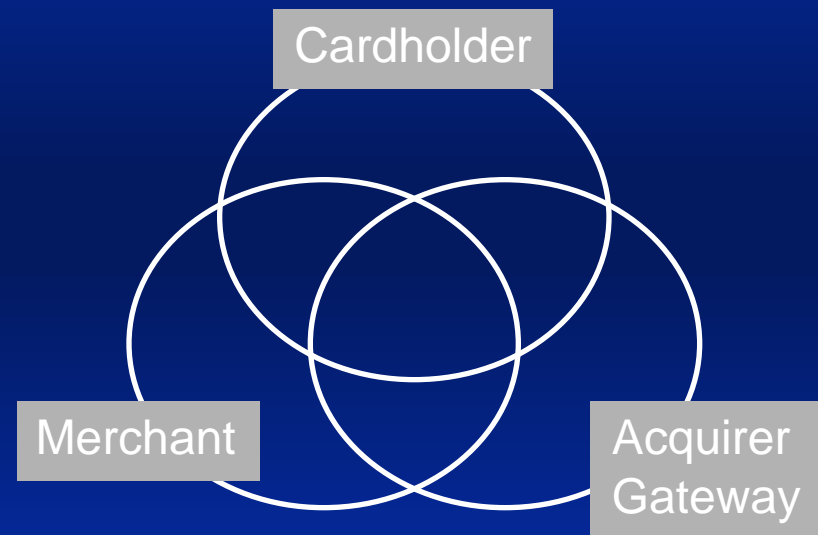
# Three Engines

- Cardholder
  - Merchant
  - Acquirer/Gateway
- 
- Includes Certificate Requesting, but
  - Not Issuing them
    - » See VeriSign



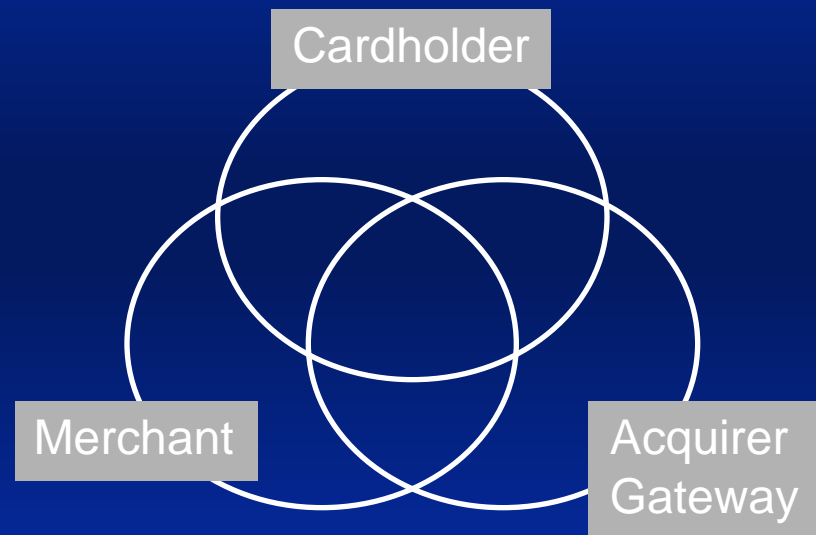
# S/PAY Includes

- Core object library
- Source for Demo Applications
- Source for ODBC database interface (Unix, Win32, Mac)
- Source for other replaceable modules



# S/PAY Includes

- Test tools
- Test scripts
- Loader utilities
- Test certificates & KeySets



# Architecture

Source

Comm.

Business Callbacks

Object

Admin	SETMsgs	CRL/BCI	CertChains
TransMgr	EncBX	PKCS7	SETCerts
SETrpc	ASN1	Crypto	X.509

Source

Database Routines

Platform  
Routines



# Cardholder Top API

SC\_StartPurchase



Comm.

Business  
Callbacks

S/PAY Core Library

Database Routines

Platform  
Routines



Copyright 1996, 1997 RSA Data Security, Inc. All rights reserved.



# Cardholder Bottom API

SX\_GetConfigRecord  
SX\_GetKeySetEncryptRecord  
SX\_GetMerchBrandRecord  
SXC\_PutCardPayTransRecord

Comm.

Business  
Callbacks

S/PAY Core Library

Database Routines

Platform  
Routines



# Cardholder Bottom API

SX\_MutexStart & End  
T\_malloc & T\_free  
T\_time

Comm.

Business  
Callbacks

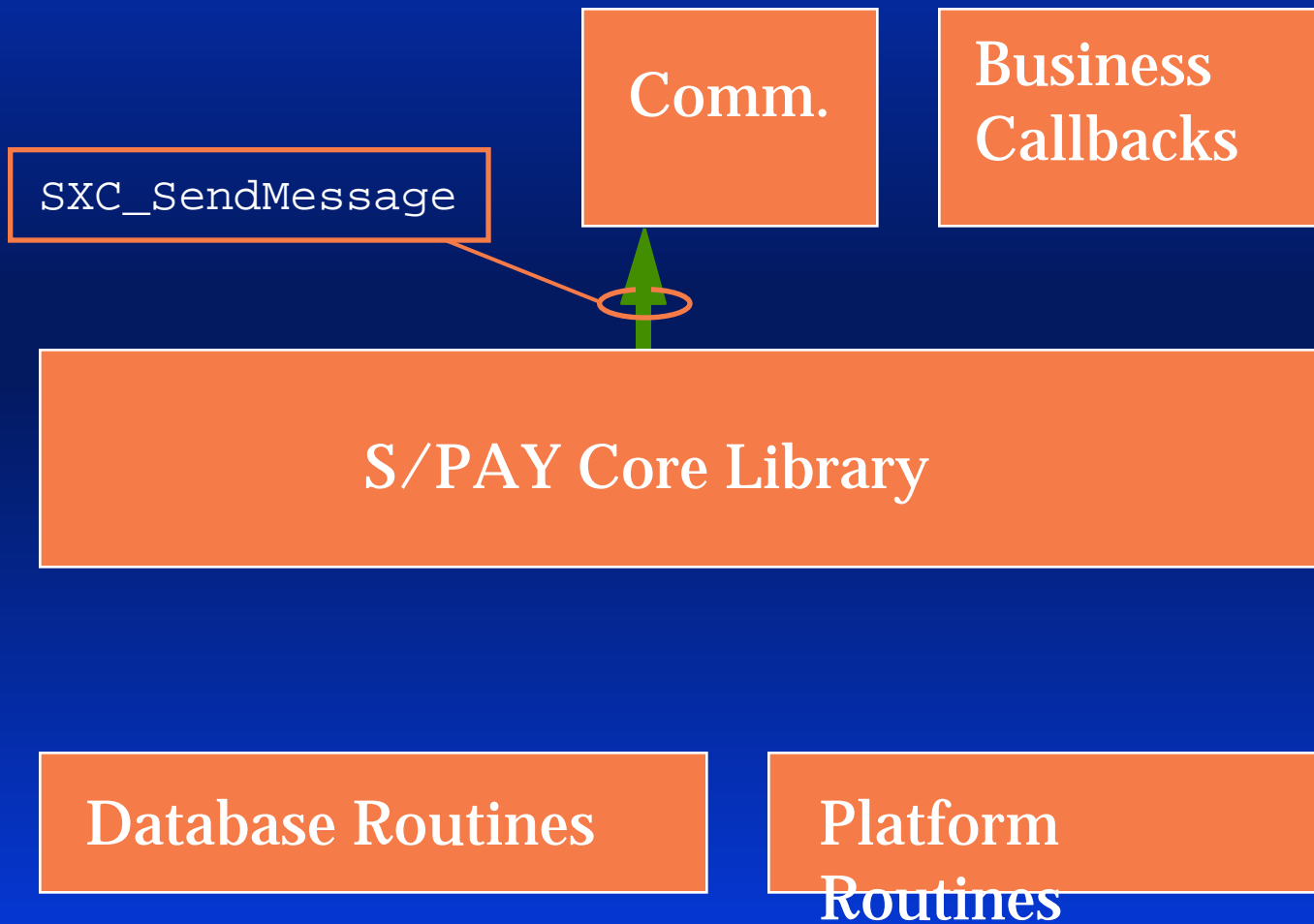
S/PAY Core Library

Database Routines

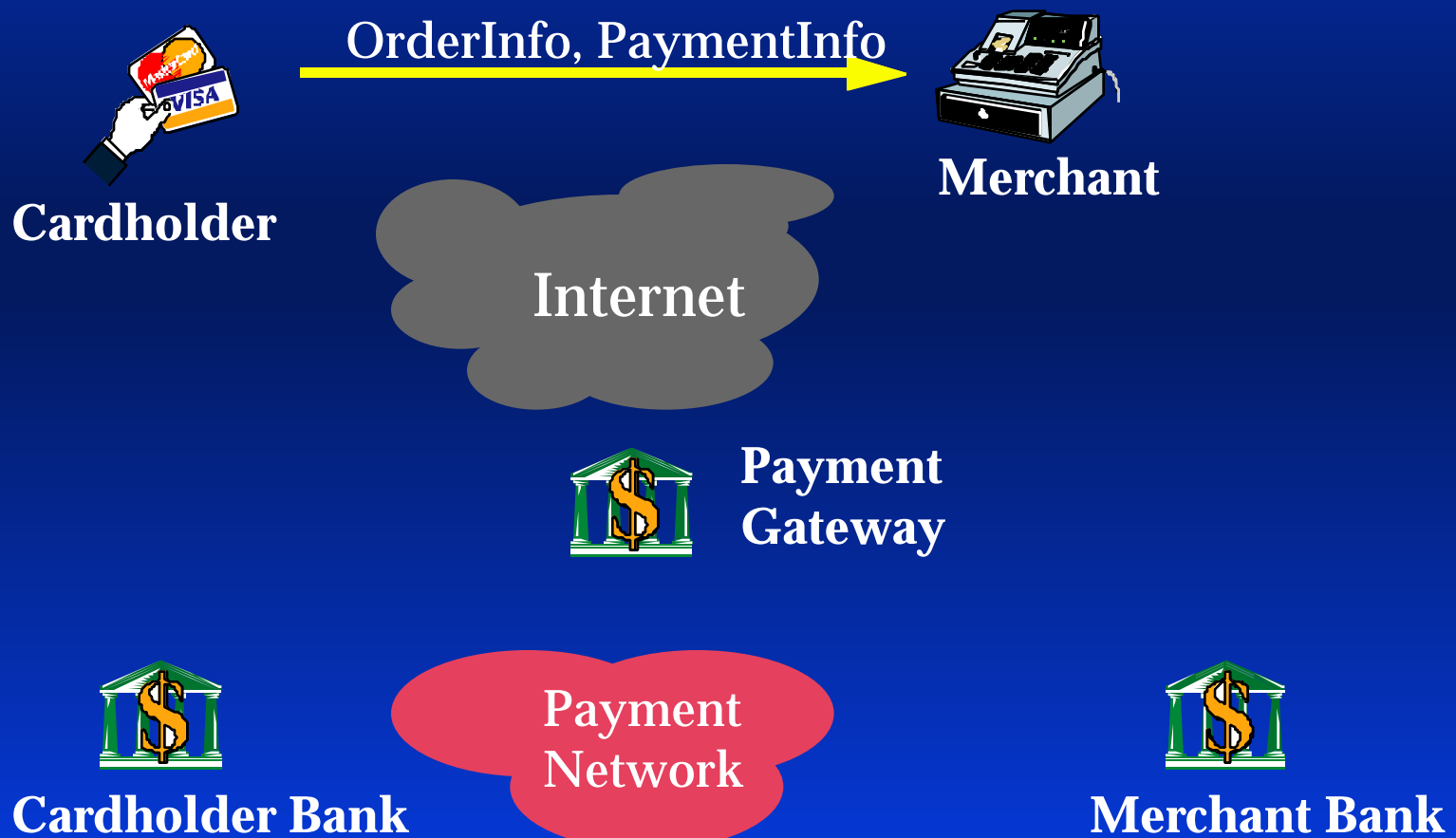
Platform  
Routines



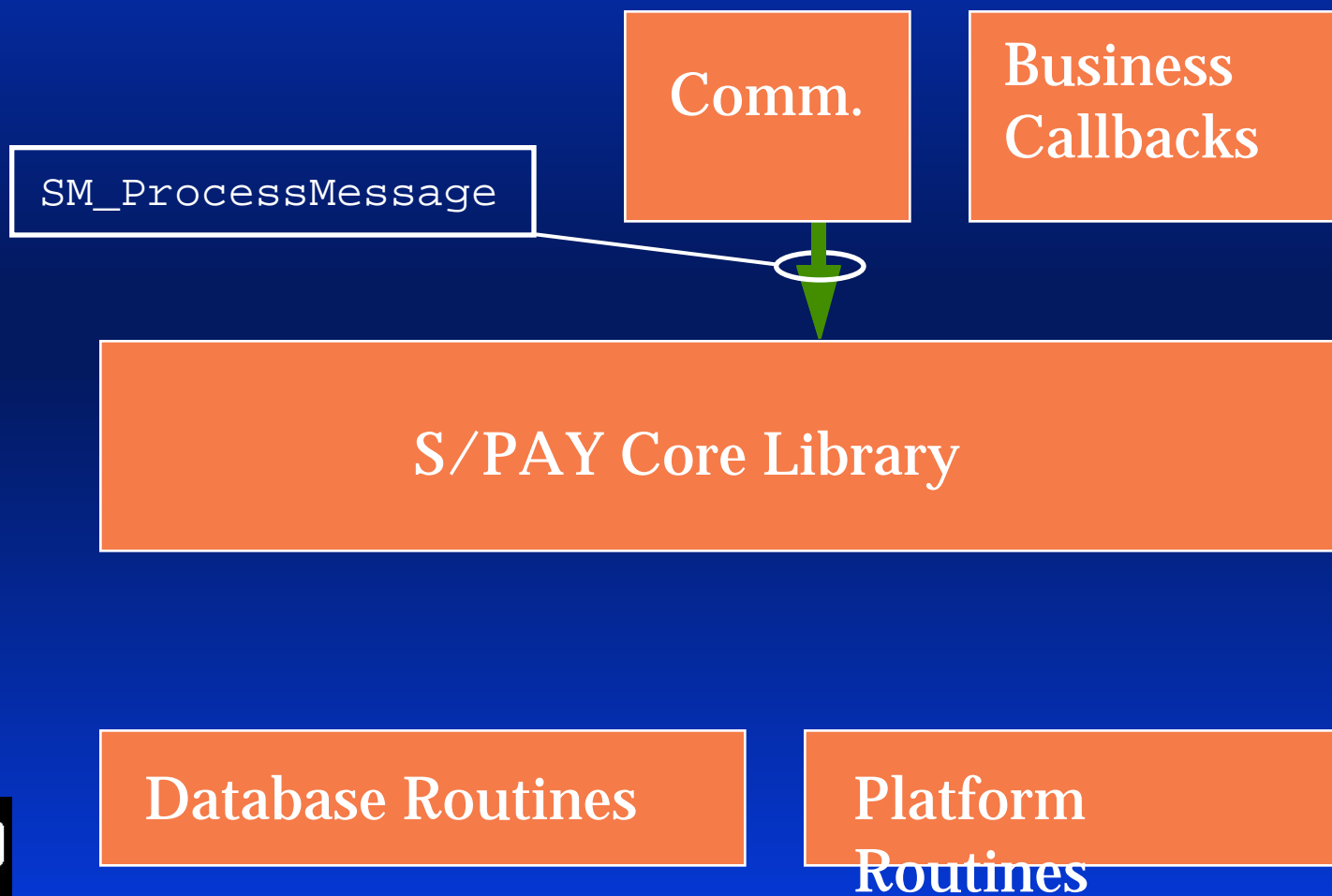
# Cardholder Comm API



# SET Purchase Request



# Merchant Top API



# Merchant Bottom API

SX\_GetCertRecord  
SX\_GetCRLRecord  
SX\_GetMerchBrandRecord  
SXM\_PutMerchPayTransRecord

Comm.

Business  
Callbacks

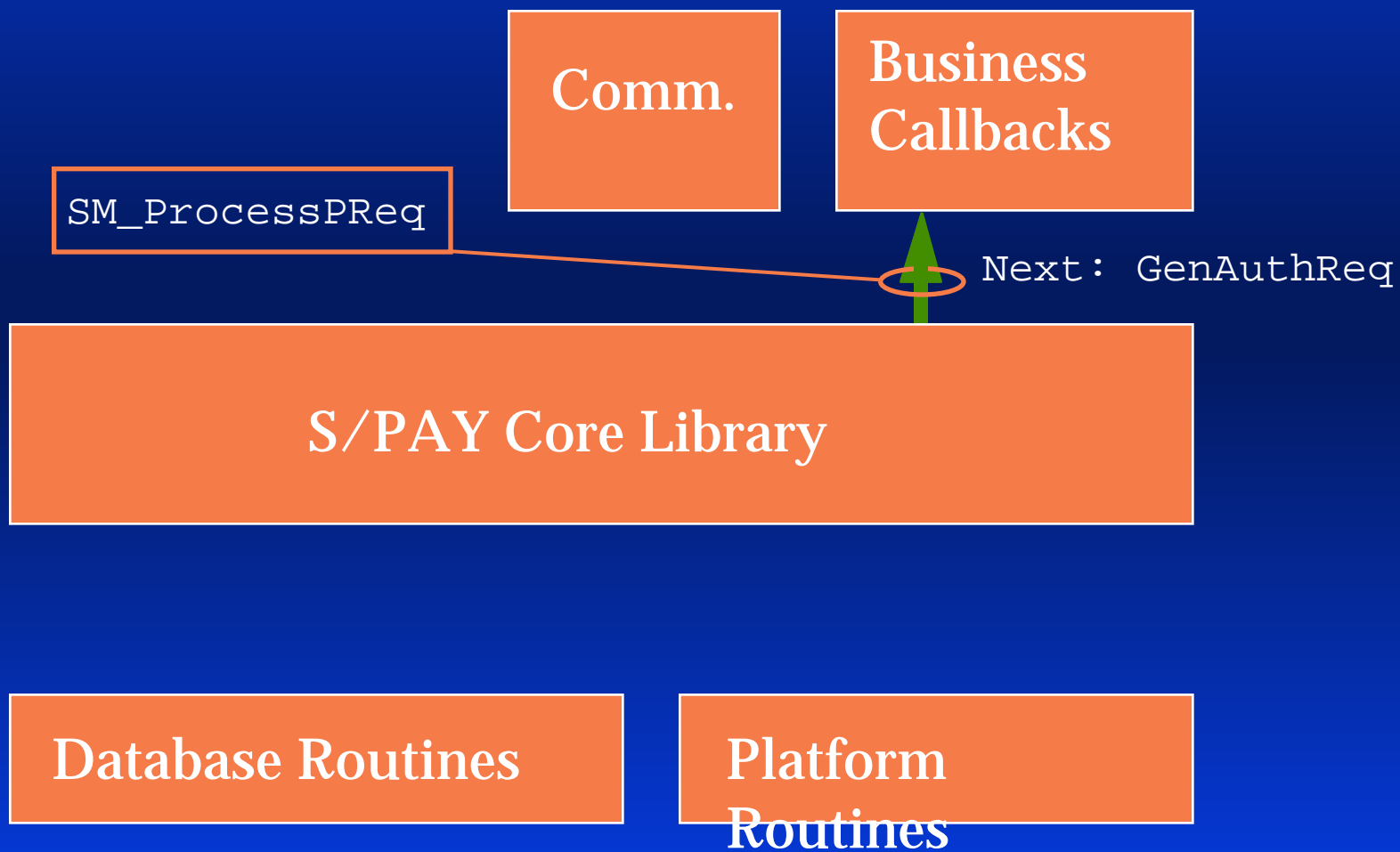
S/PAY Core Library

Database Routines

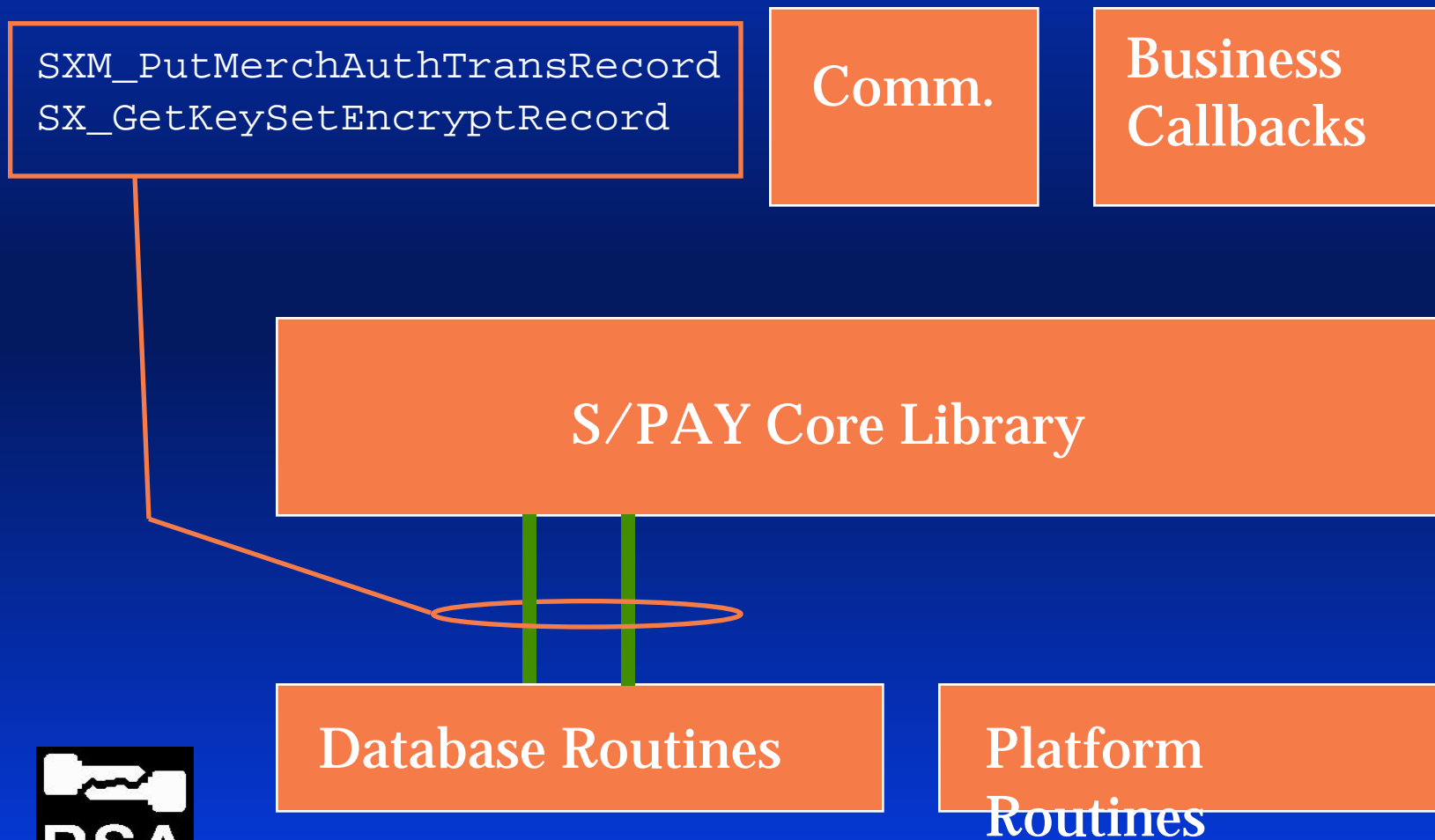
Platform  
Routines



# Merchant Callback API

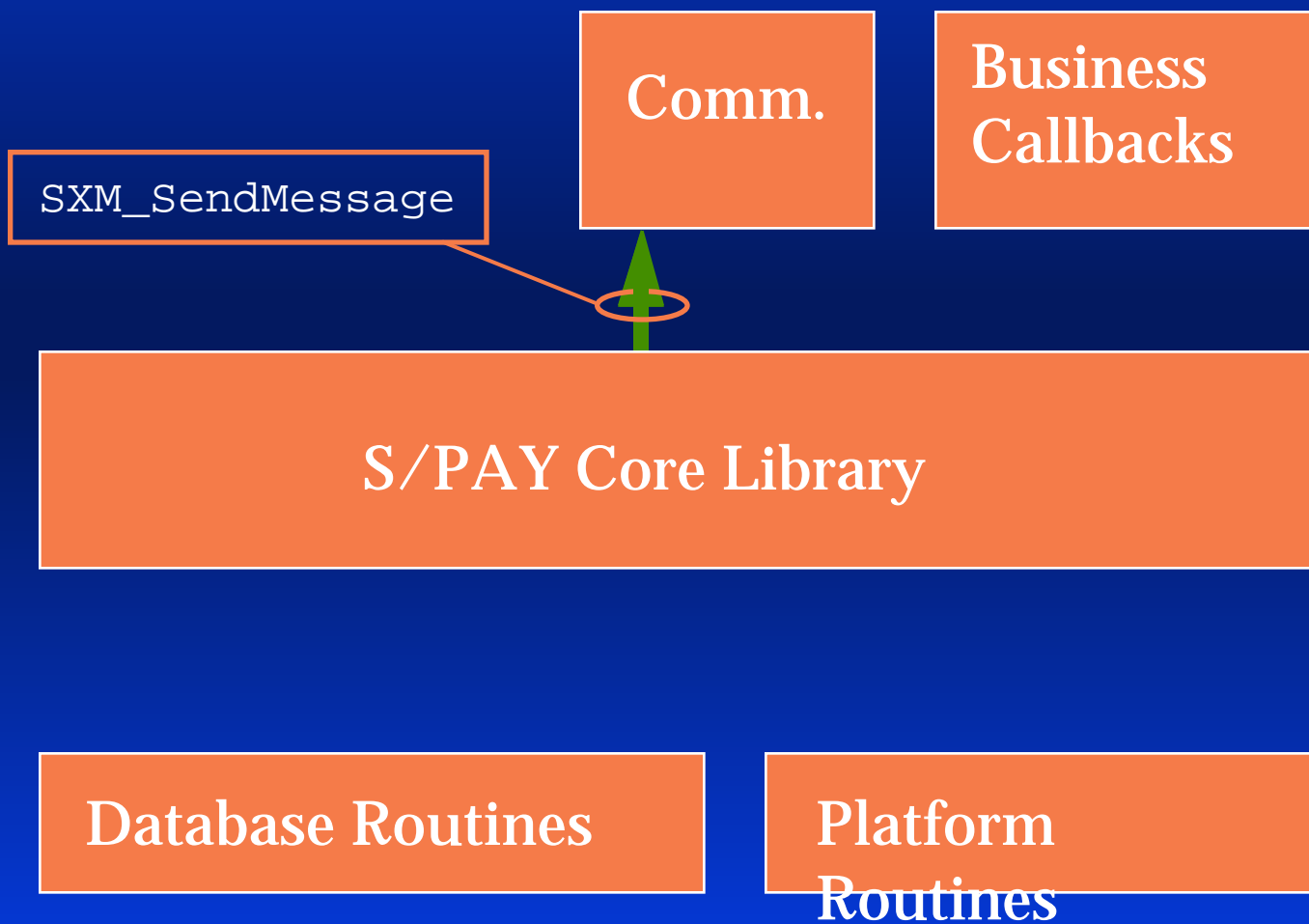


# Merchant Bottom API

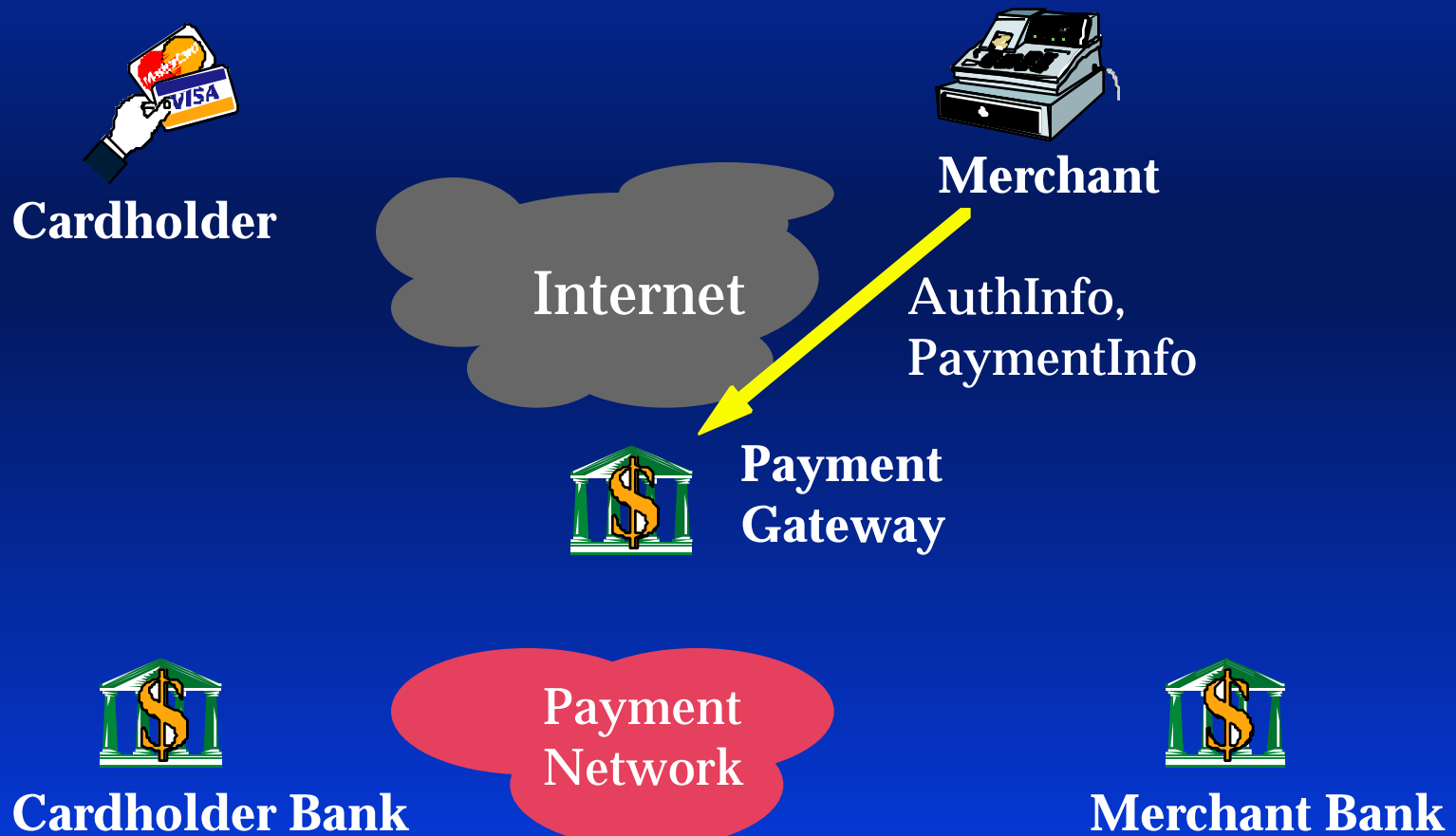




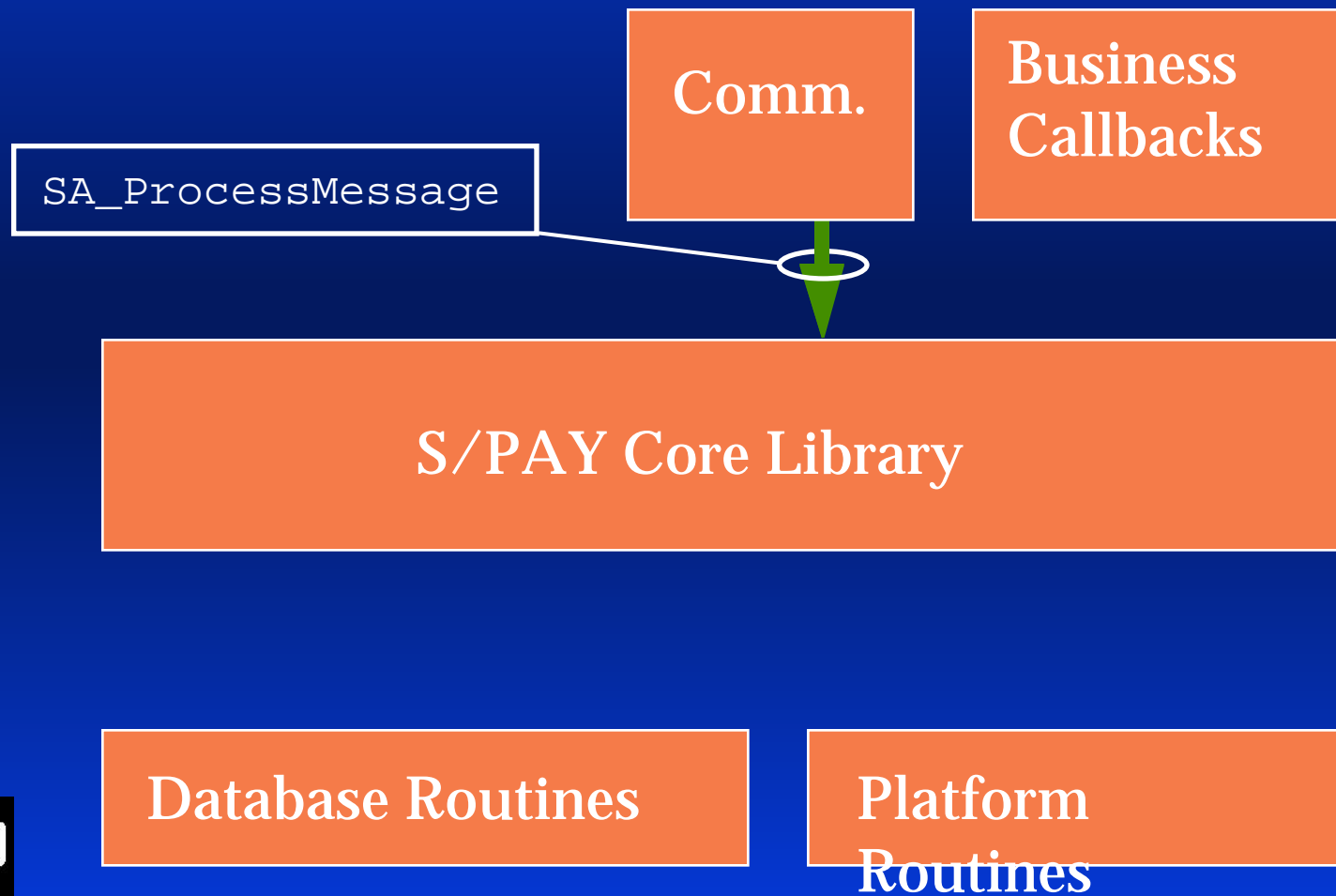
# Merchant Comm API



# SET Authorization Request



# Gateway Top API



# Gateway Bottom API

SX\_GetCertRecord & CRL  
SX\_GetMerchBrandRecord  
SX\_GetKeySetEncryptRecord  
SXA\_PutAcqPayTransRecord  
SXA\_PutAcqAuthTransRecord

Comm.

Business  
Callbacks

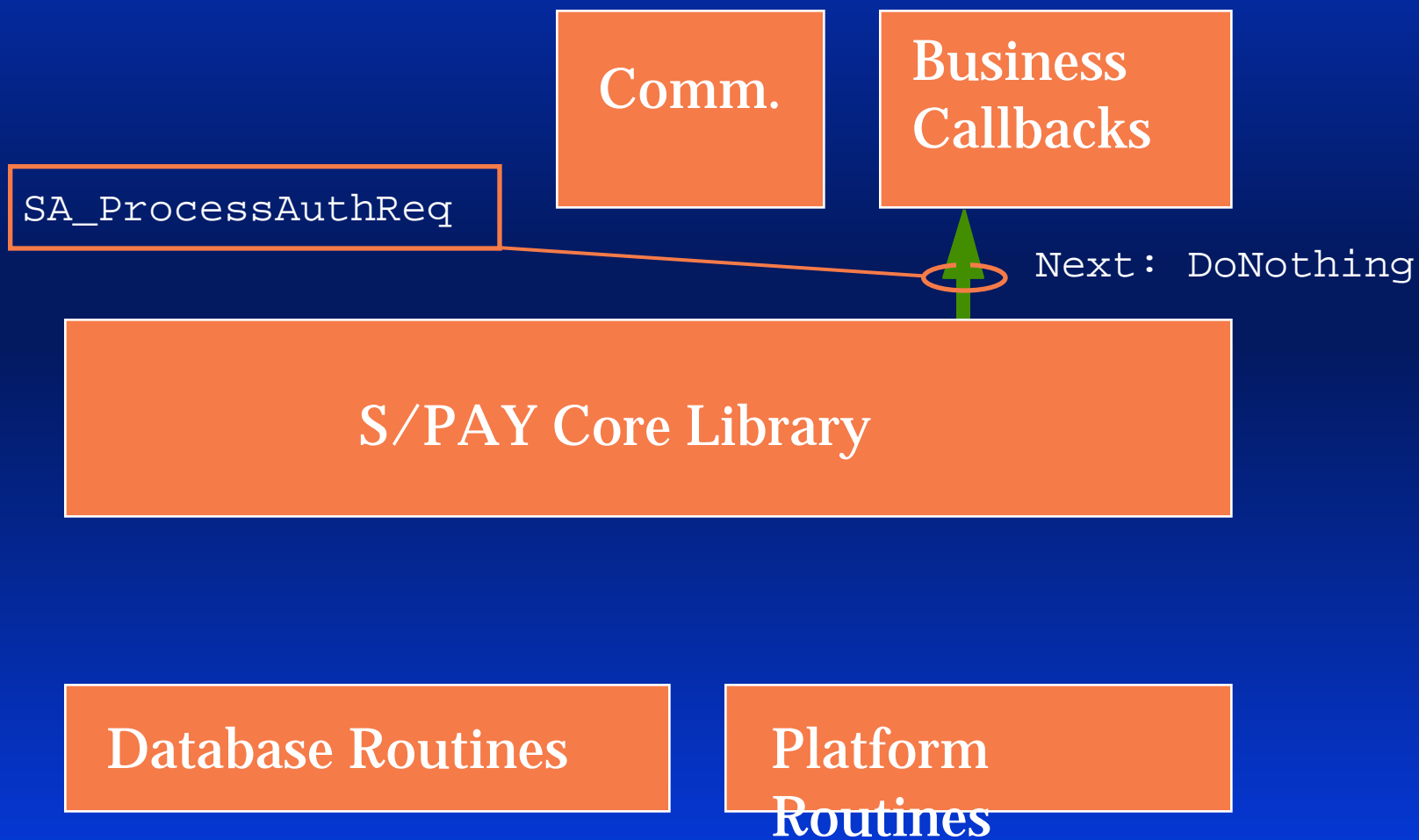
S/PAY Core Library

Database Routines

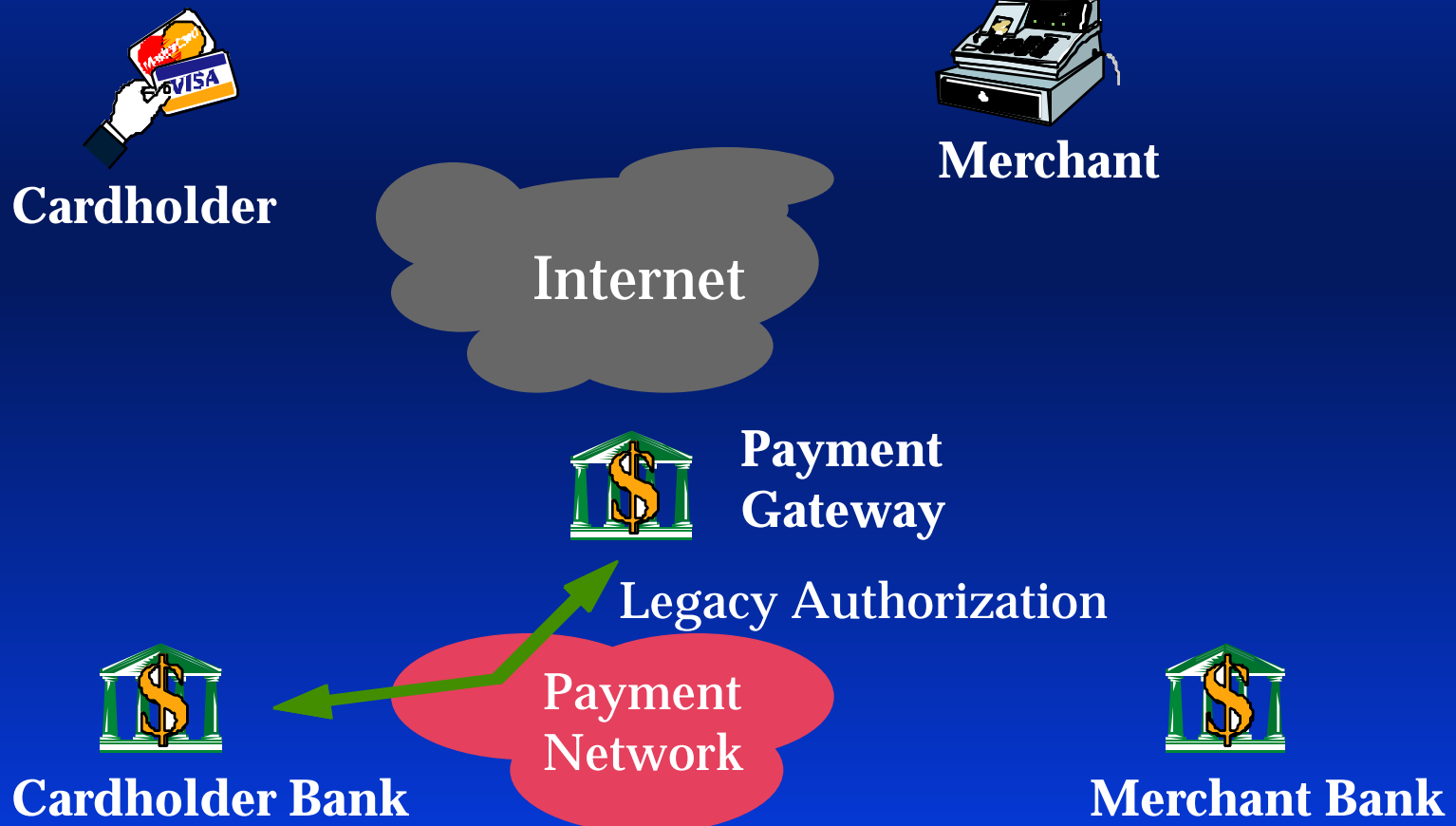
Platform  
Routines



# Gateway Callback API



# SET Legacy Auth



# Gateway Top API

SA\_CompleteAuthorization



Comm.

Business  
Callbacks

S/PAY Core Library

Database Routines

Platform  
Routines



Copyright 1996, 1997 RSA Data Security, Inc. All rights reserved.

# Gateway Bottom API

SX\_GetCertRecord & CRL  
SX\_GetMerchBrandRecord  
SX\_GetKeySetEncryptRecord  
SXA\_PutAcqAuthTransRecord

Comm.

Business  
Callbacks

S/PAY Core Library

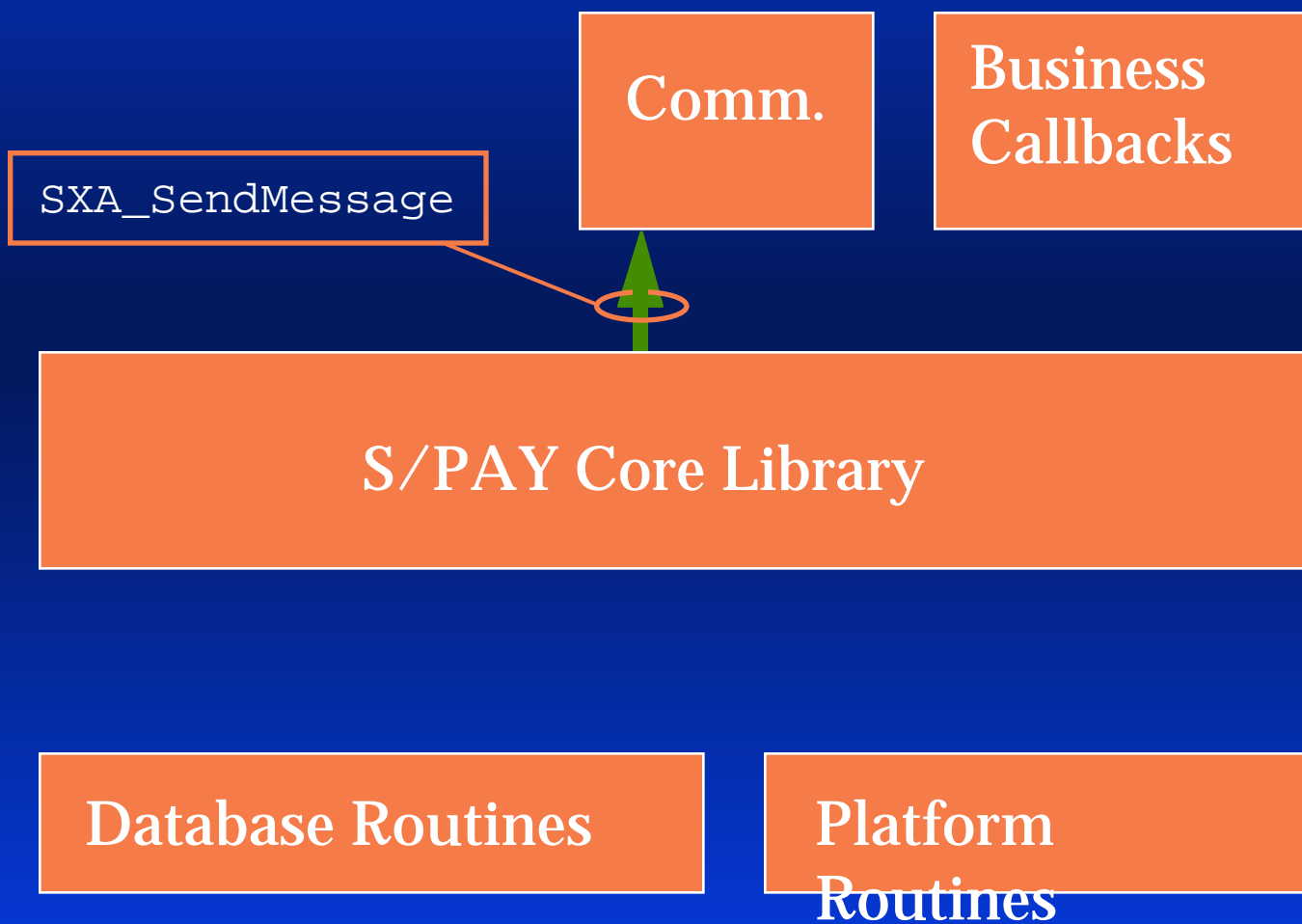
Database Routines

Platform  
Routines

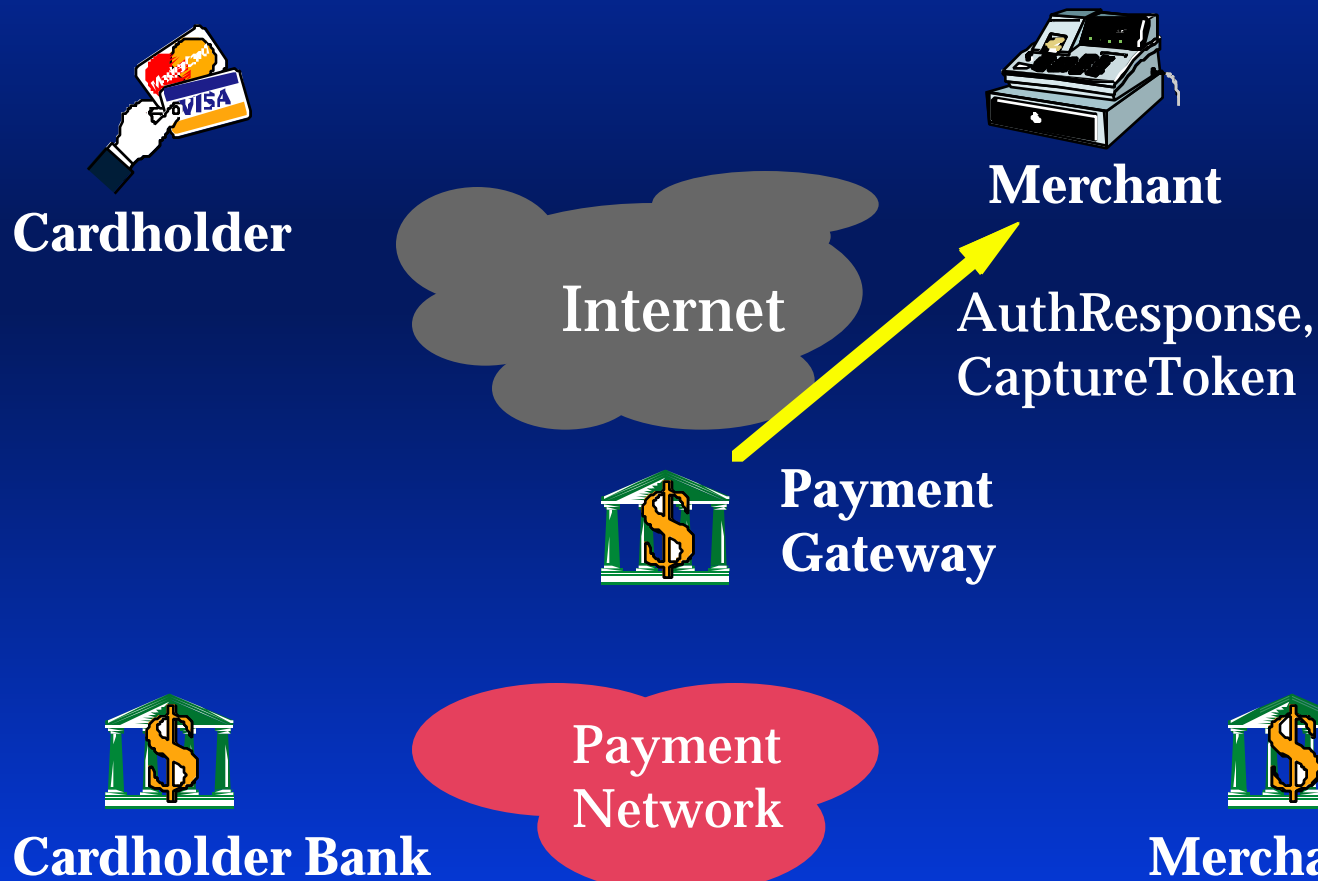




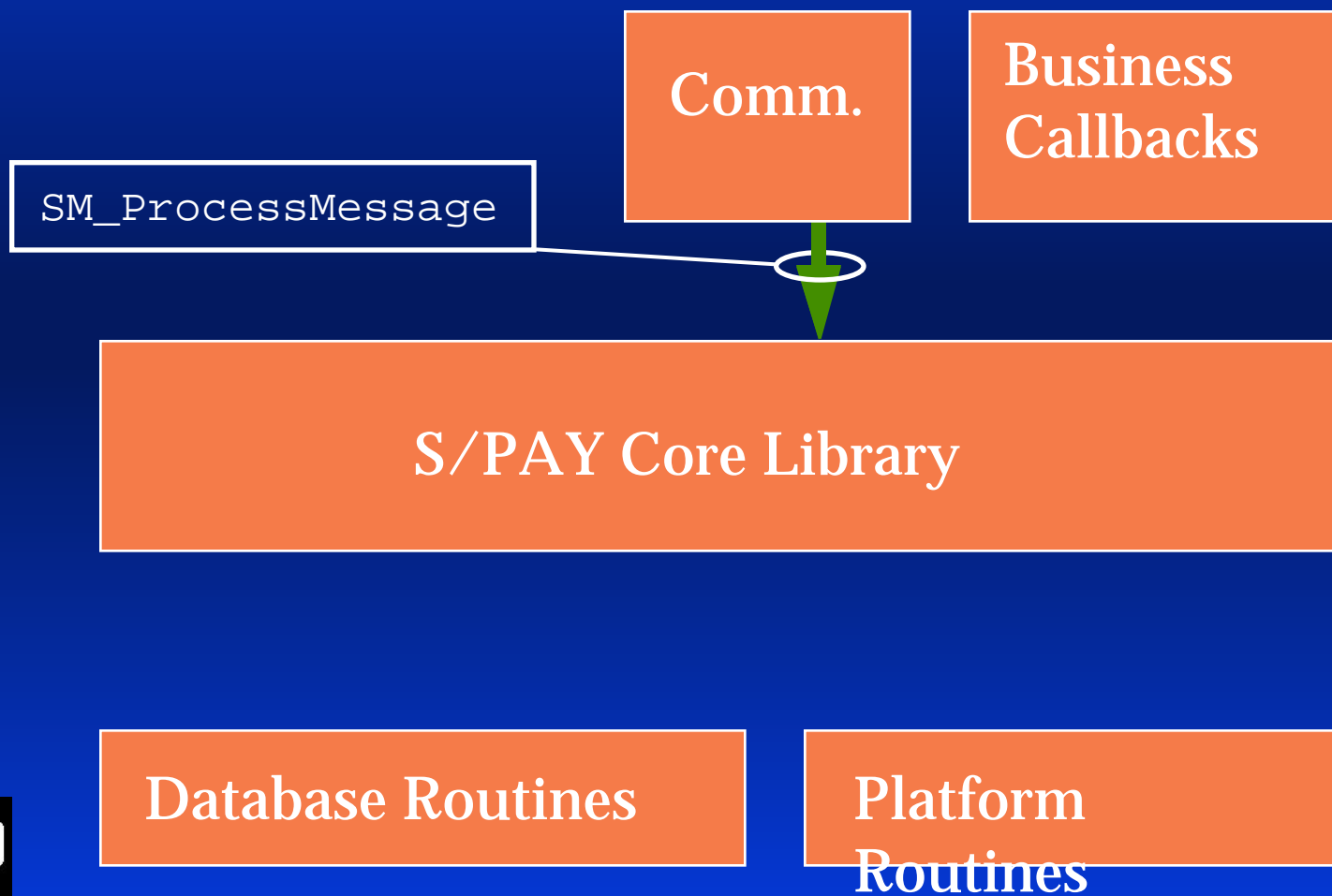
# Gateway Comm API



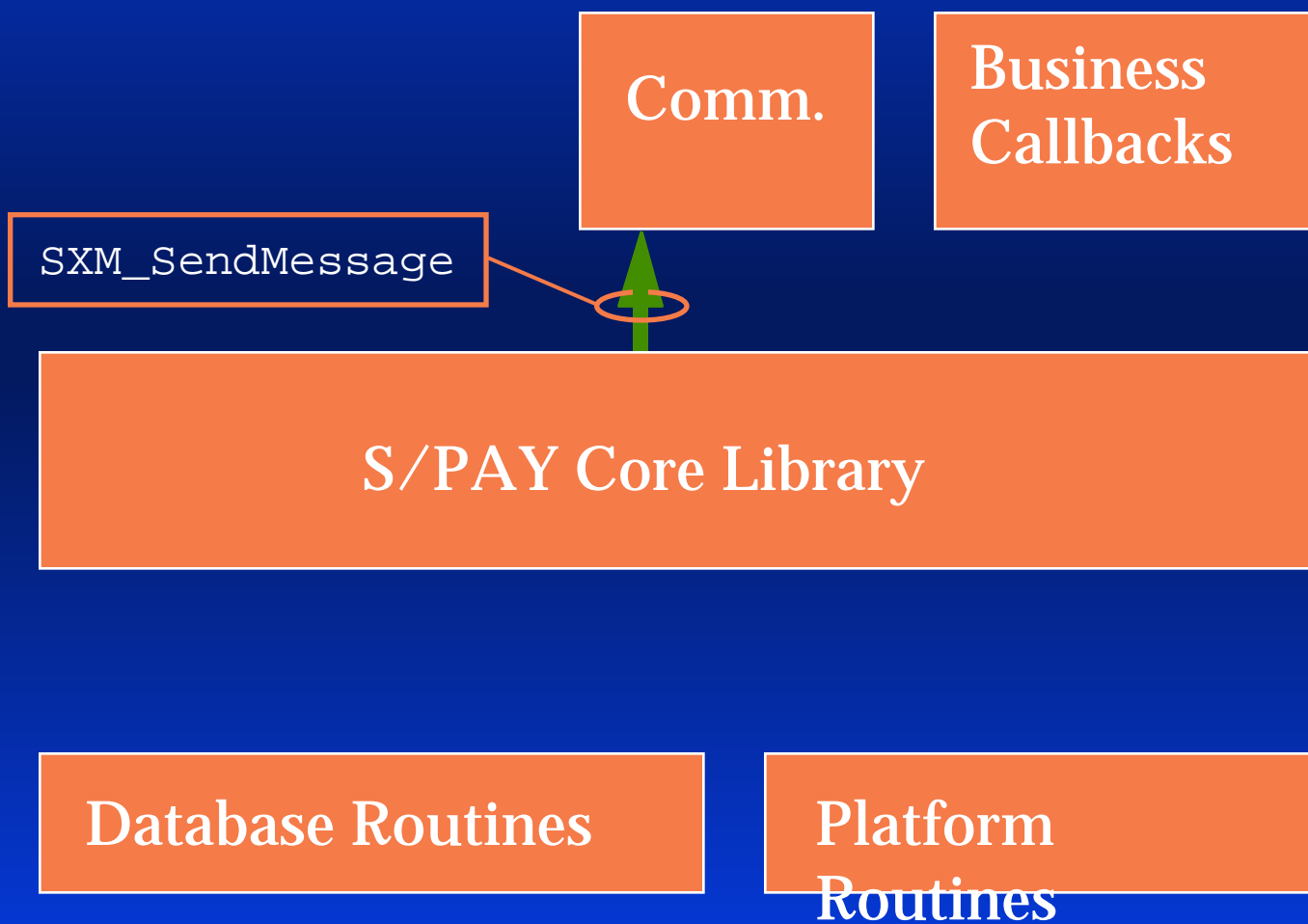
# SET Authorization Response



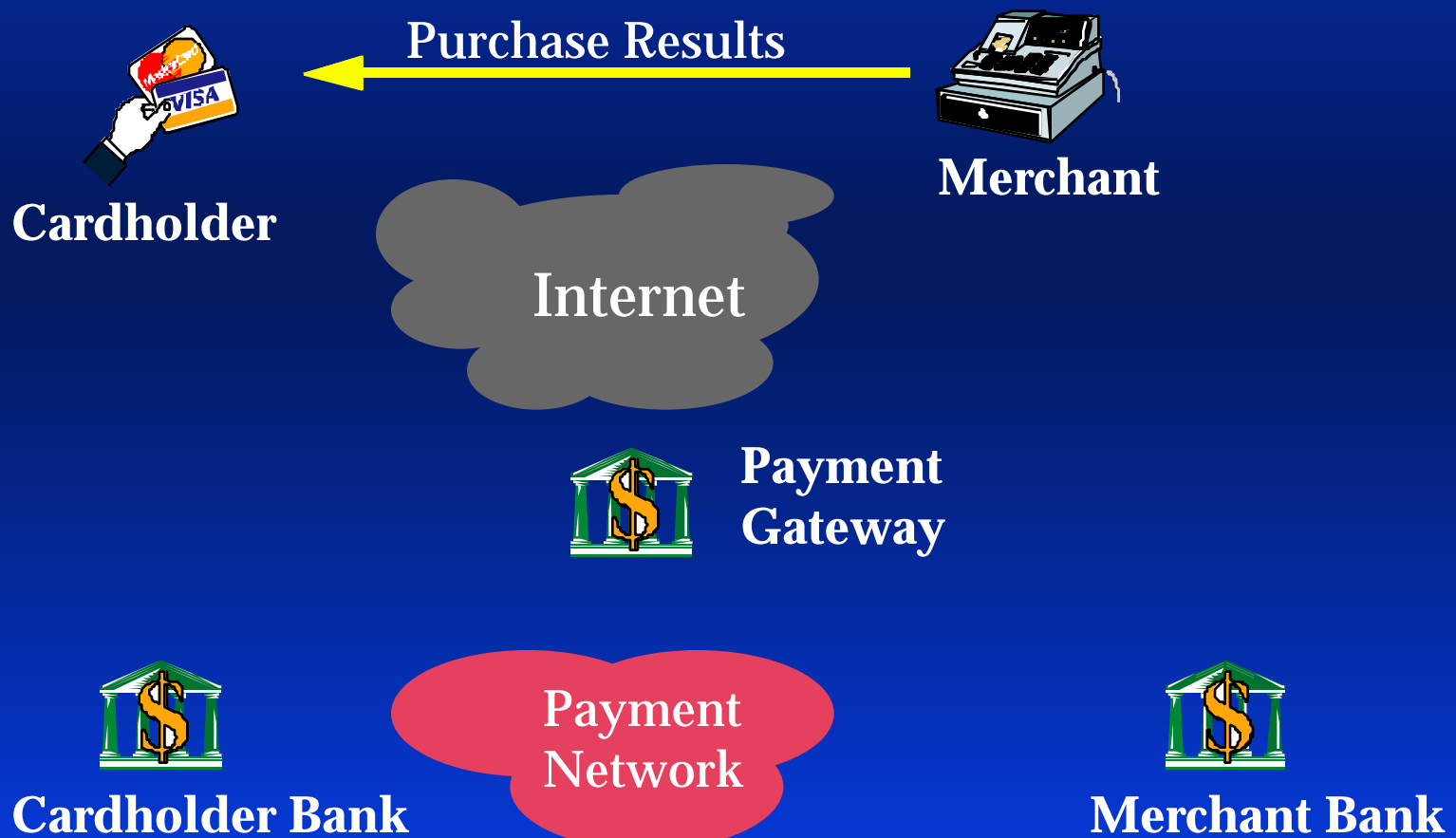
# Merchant Top API



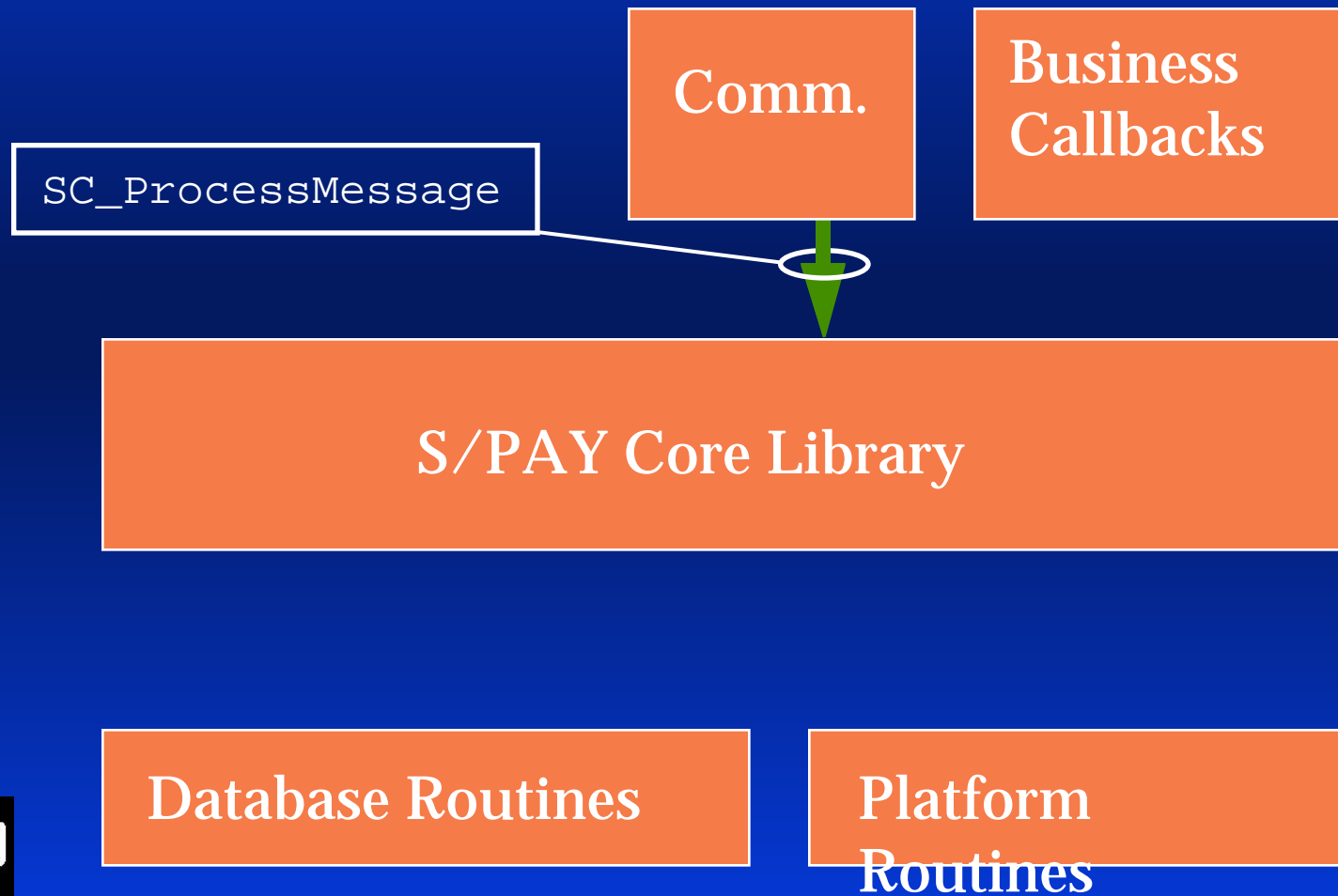
# Merchant Comm API



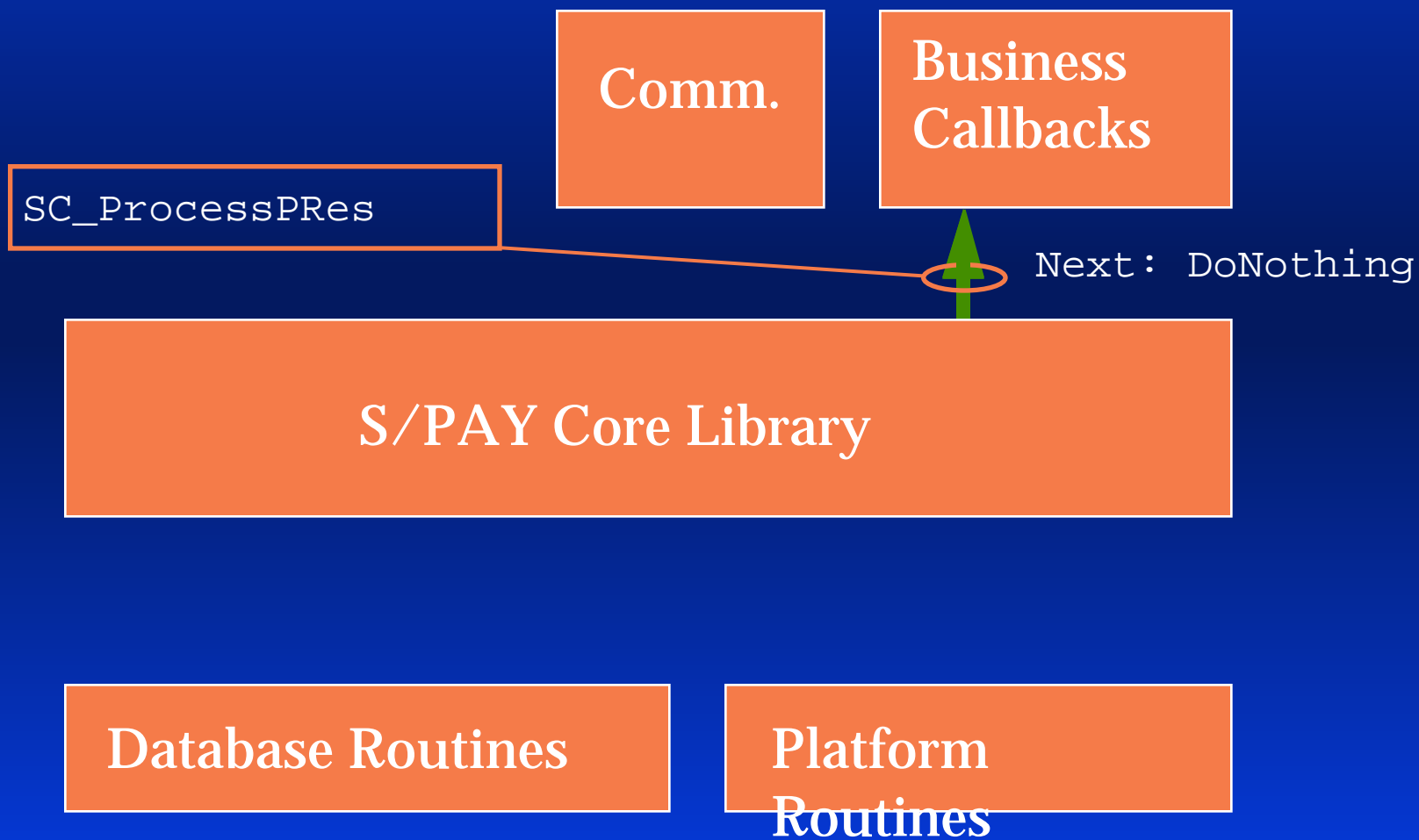
# SET Purchase Response



# Cardholder Top API



# Cardholder Callback API



# SET Purchase Done



Payment  
Gateway



Cardholder Bank



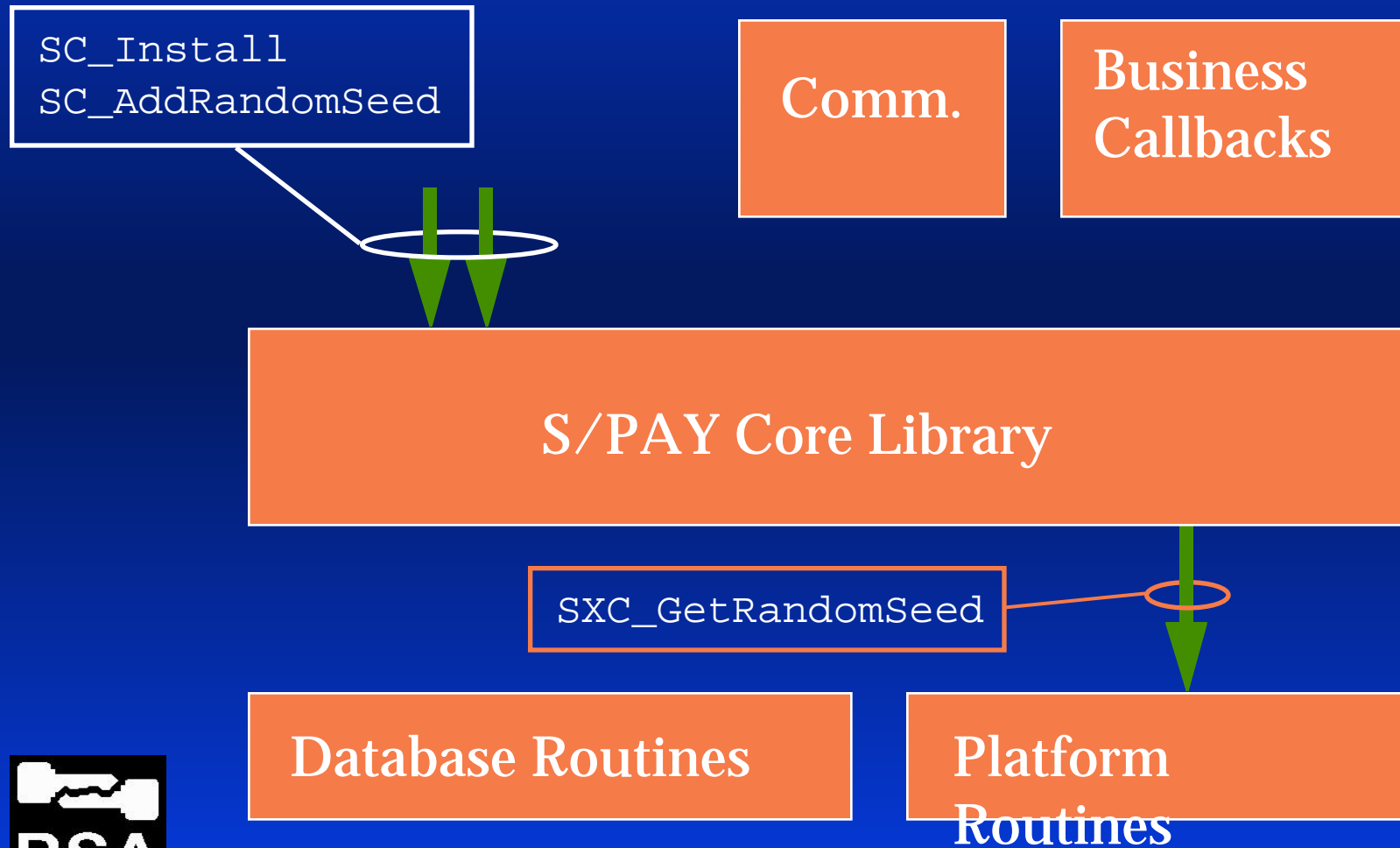
Merchant Bank



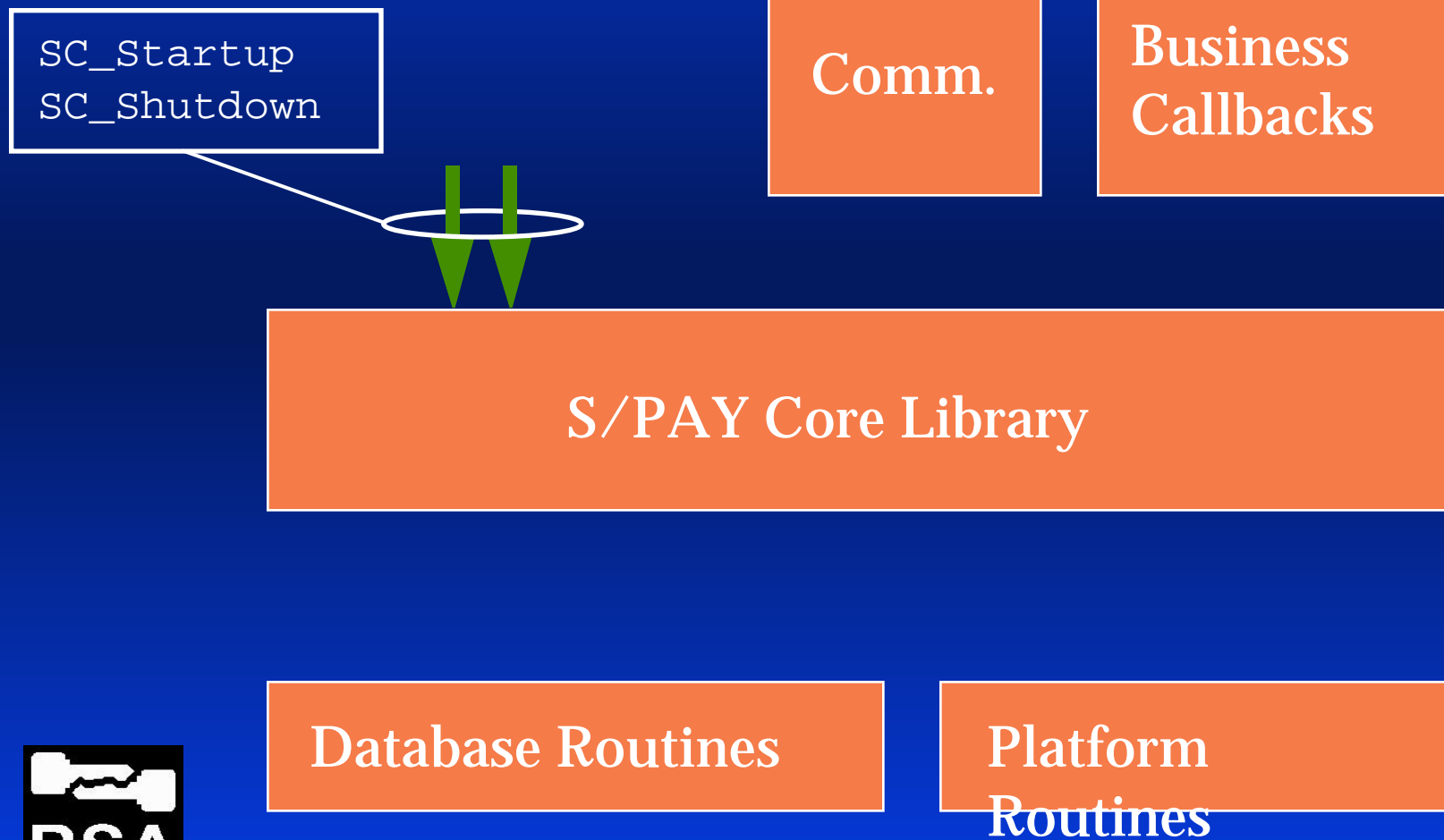
Copyright 1996, 1997 RSA Data Security, Inc. All rights reserved.



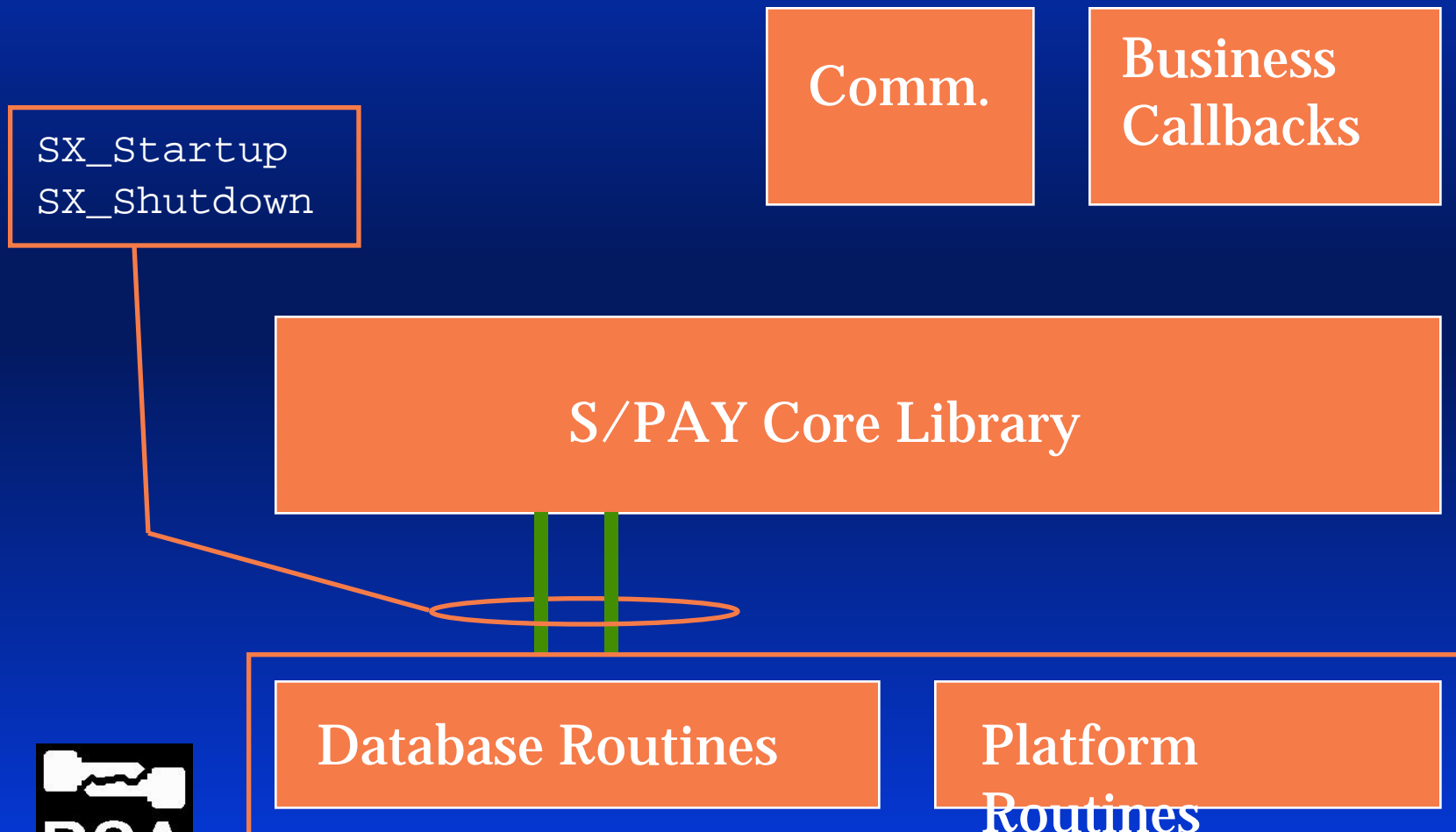
# Admin API



# Admin API



# Admin Bottom API



# Loader API

SC\_LoadCard  
SC\_LoadCerts  
SC\_LoadMerchBrand

Comm.

Business  
Callbacks

S/PAY Core Library

Database Routines

Platform  
Routines




# Handles

- SC\_ProcessMessage (sHandle, appHandle, msgBuffer, ... )
- sHandle created by SC\_Startup (...).
- appHandle created by application



# Handles

- Both handles passed to callback & database routines
  - SC\_ProcessMessage (sHandle, appHandle, msgBuffer, ... )
  - SXC\_GetCertByIS (sHandle, appHandle, issuerName, serialNumber, ... )
- 
- The diagram consists of two orange arrows pointing downwards. The first arrow originates from the 'sHandle' parameter in the SC\_ProcessMessage function signature and points to the 'sHandle' parameter in the SXC\_GetCertByIS function signature. The second arrow originates from the 'appHandle' parameter in the SC\_ProcessMessage function signature and points to the 'appHandle' parameter in the SXC\_GetCertByIS function signature.



# AppHandle

- Passes state from top API to callbacks.
- Ex: Pass GUI info to business callbacks so they can display progress.
- Ex: Hold onto Oracle database context.
- Ex: Pass thread specific state to platform routines like SXC\_MutexStart.



# AppHandle

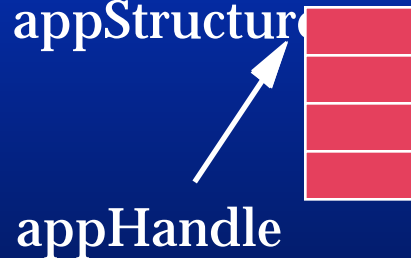
- Return info from callbacks to top API.
- Ex: SXC\_ProcessPRes puts authorization amount into struct identified by appHandle.
- Ex: SXC\_SendMessage could copy message buffer into appHandle to let top level code do HTTP post.





# AppHandle Example

1. Create appStructure



```
SPAY_Wrapper.DoProcessMessage()  
/* Enable application code to  
* send the SET message, instead  
* of the "comm" routine. */
```

Comm.

Business  
Callbacks

S/PAY Core Library

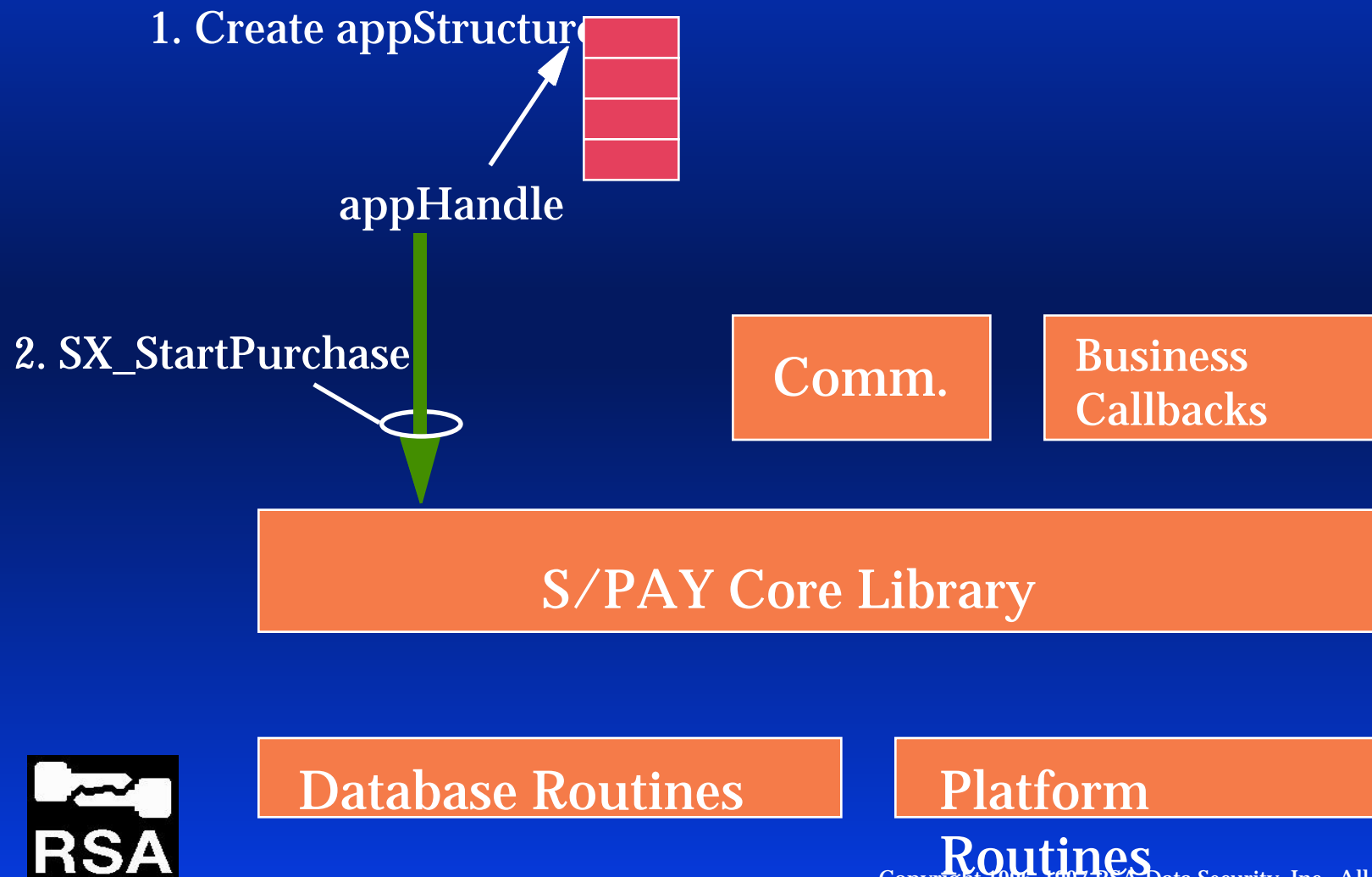
Database Routines

Platform  
Routines

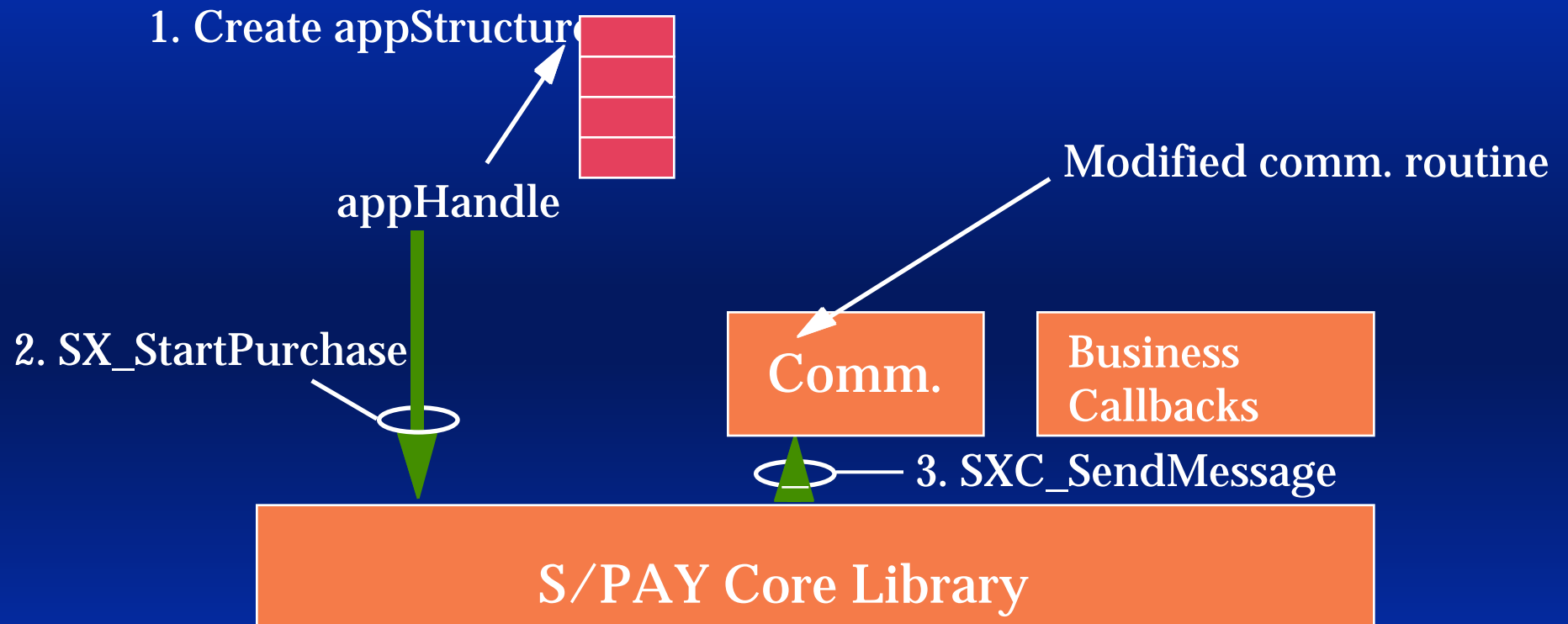


Copyright 1996, 1997 RSA Data Security, Inc. All rights reserved.

# AppHandle Example



# AppHandle Example

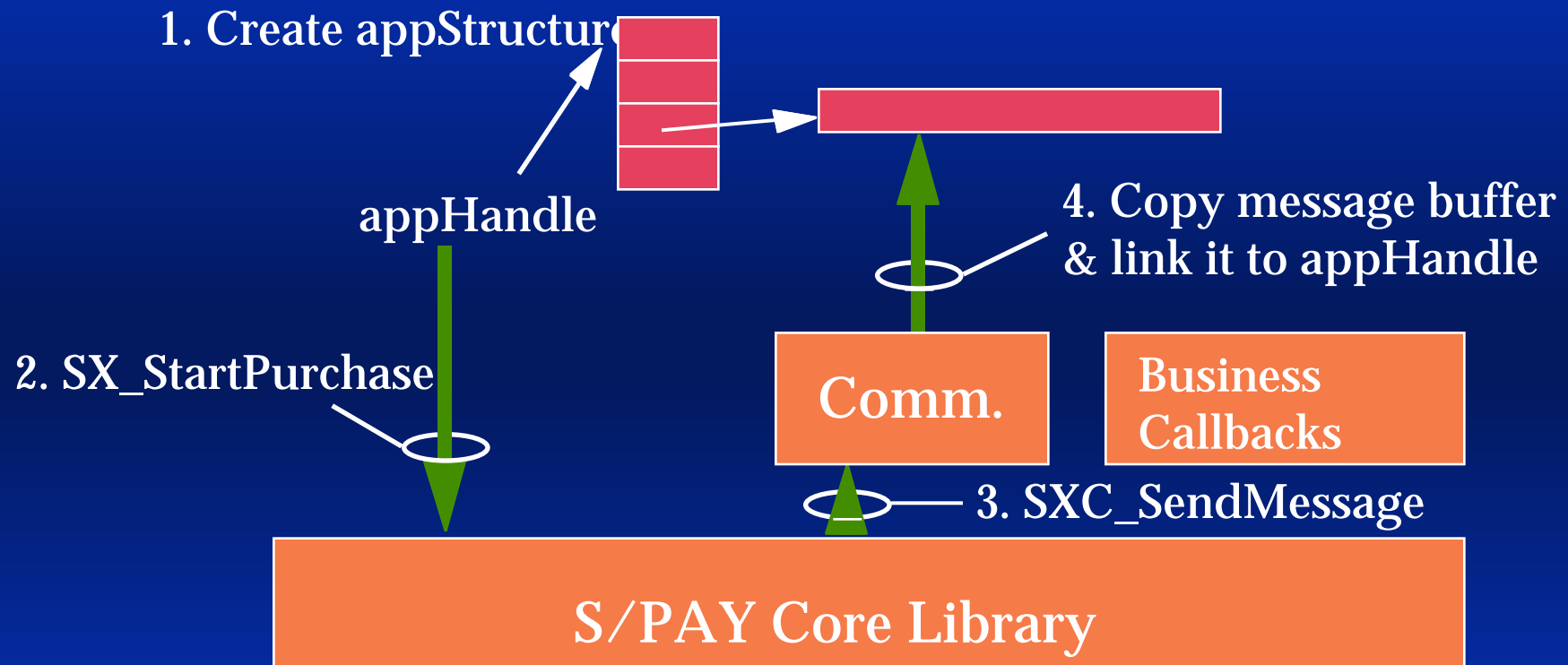


Database Routines

Platform  
Routines

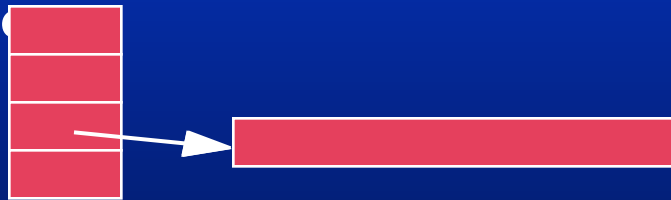
Copyright 1996, 1997 RSA Data Security, Inc. All rights reserved.

# AppHandle Example



# AppHandle Example

1. Create appStructure



5. SC\_StartPurchase  
returns OK



Comm.

Business  
Callbacks

S/PAY Core Library

Database Routines

Platform

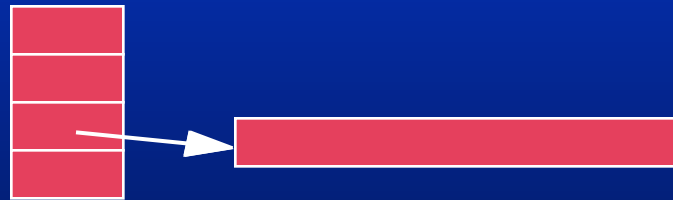
Routines



Copyright 1996, 1997 RSA Data Security, Inc. All rights reserved.

# AppHandle Example

6. Application calls  
http Post routine  
with buffer.
7. Free buffer.



Comm.

Business  
Callbacks

S/PAY Core Library

Database Routines

Platform  
Routines



Copyright 1996, 1997 RSA Data Security, Inc. All rights reserved.

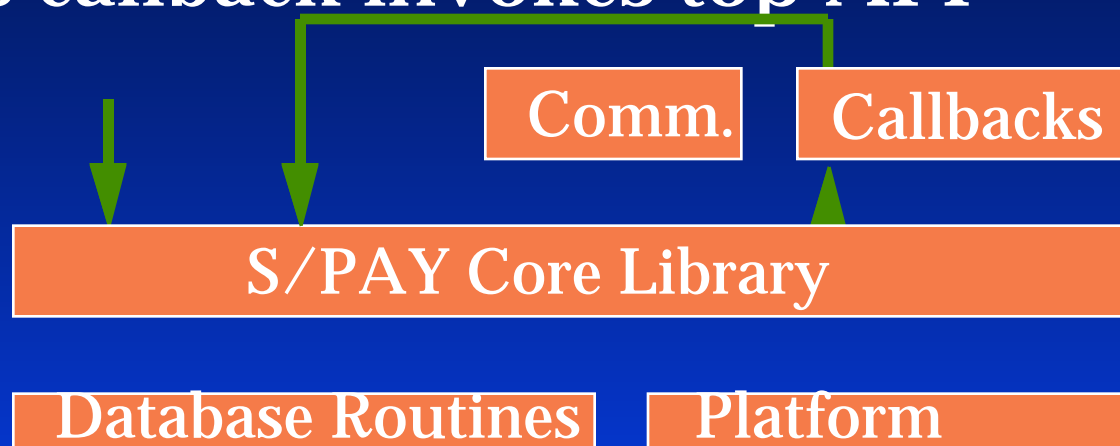
# Java Example

```
try {  
    receiver = new Receiver(commParameters);  
    sender = new Sender(commParameters);  
  
    request = receiver.nextMessage;  
    response = SPAY_Wrapper.DoProcessMessage(request);  
    sender.post(response);  
  
} catch( Exception e ) {  
    System.out.println("Gateway: "+e);  
}
```



# Sample Gateway Code

- Single process, single thread
- Synchronous interface to payment network
- Business callback invokes top API





# Gateway Include Files

```
#include "acqapi.h"
```

```
typedef struct appStruct {  
    void *      dbHandle;  
} AppStruct;
```

```
AppStruct    appStruct;  
AppStruct *  appHandle = &appStruct;
```



# Gateway Main

```
int main (int argc, char* argv[])
{
    ITEM      passPhrase;          /* Password */

    passPhrase.data = (POINTER) argv[1];
    passPhrase.len = strlen(argv[1]);
```



# Gateway Startup

```
int          status;  
SHandle      sHandle;  
  
status = SA_Startup(appHandle,  
                    &passPhrase, &sHandle);  
T_memset (passPhrase.data,  
          0, passPhrase.len);  
if (status != 0) return status;
```



# Gateway Process Loop

```
ITEM  setMessage;          /* Message received */
ITEM  replyCommName;       /* Who it was from */

while (WaitForNextSetMessage()) {
    status = GetSetMessage (&setMessage,
                           &replyCommName);

    if (status != 0) break;
    status = SA_ProcessMessage(appHandle,
                              sHandle,
                              &setMessage,
                              &replyCommName);

    if (status != 0) break;
}
```



# Gateway Shutdown

```
status = SA_Shutdown(appHandle, &sHandle);  
return status;  
}    /* End of main */
```



# Gateway Callback

```
int  SXA_ProcessAuthReq ( /* Valid AuthReq */
    AppHandle  appHandle,          /* in */
    SHandle    sHandle,            /* in */
    RRPID *    rrpId,              /* in */
    MerchantInfo * merInfo,        /* in */
    Date *     currentDate,        /* in */
    MerTermIDs * merTermIDs,       /* in */
    AuthRetNum * authRetNum,       /* in */
    CardInfo *  cardInfo,          /* in */
    PurchaseInfo * purchaseInfo,   /* in */
    AuthTokenArgs * authTokenArgs, /* in */
    AuthReqArgs * authReqArgs,     /* in */
    Bool        captureNow,        /* in */
    SaleDetail * saleDetail)       /* in */
```

```
{
```



# Gateway Business Check

```
int          status = 0;
CapTokenArgs * capTokenArgs = NULL;
AcqCardMsg *  acqCardMsg = NULL;
ITEM         authTokenOpaque;
AuthResArgs   authResArgs;
LegacyResult  legacyRes;

if (captureNow) return SE_Unsupported;
```



# Gateway Legacy Call

```
status = DoLegacyAuthorization ( ...,  
                                &legacyRes );  
if (status != 0) return status;
```





# Gateway Prepare Resp. Data

```
memset(&authResArgs, 0, sizeof(authResArgs));
authResArgs.respCode.data =
    (POINTER) legacyRes.responseCode;
authResArgs.respCode.len =
    strlen(legacyRes.responseCode);

authResArgs.authAmount.currency = 840; /* US Dollars */
authResArgs.authAmount.amount = legacyRes.authAmount;
authResArgs.authAmount.amtExp10 = -2; /* Pennies to $ */

authResArgs.present |= AUTHRESARGS_PAYSYSID;
authResArgs.paySysID.data = (POINTER) legacyRes.paySysID;
authResArgs.paySysID.len = strlen(legacyRes.paySysID);
```



# Gateway Send Response

```
authTokenOpaque.len = 0;

status = SA_CompleteAuthorization (
    appHandle,      /* in */
    sHandle,        /* in */
    rrpId,          /* in */
    &authResArgs,    /* in */
    &authTokenOpaque, /* in, optional */
    capTokenArgs,    /* in, optional */
    acqCardMsg);    /* in, optional */

return status;
} /* End of SXA_ProcessAuthReq */
```



# Customizing S/PAY

- Simultaneous authorize & capture
- Using Oracle
- Multiple processes running S/PAY
- Multiplexed Comm to Gateway
- Adding new SET-like messages



# Auth & Capture

- Change business two rule routines:
- SXM\_ProcessPReq
- SXA\_ProcessAuthReq



# Using Oracle

- Purchase Oracle ODBC driver
- Relink your application with Oracle

## Database Module - Source Code

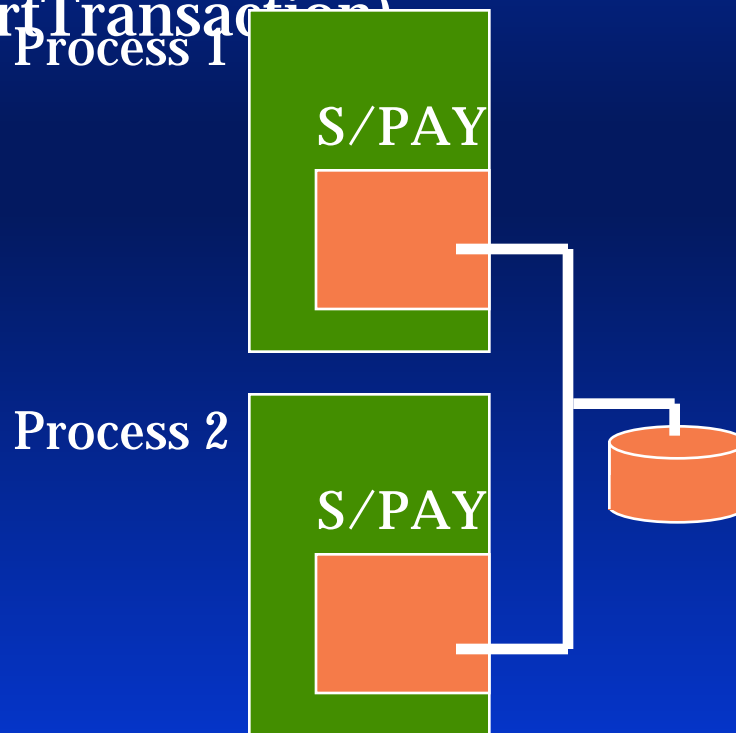
SET DB Interface Routines

ODBC Driver (Informix, Sybase,  
Oracle, DB2, MS Access)



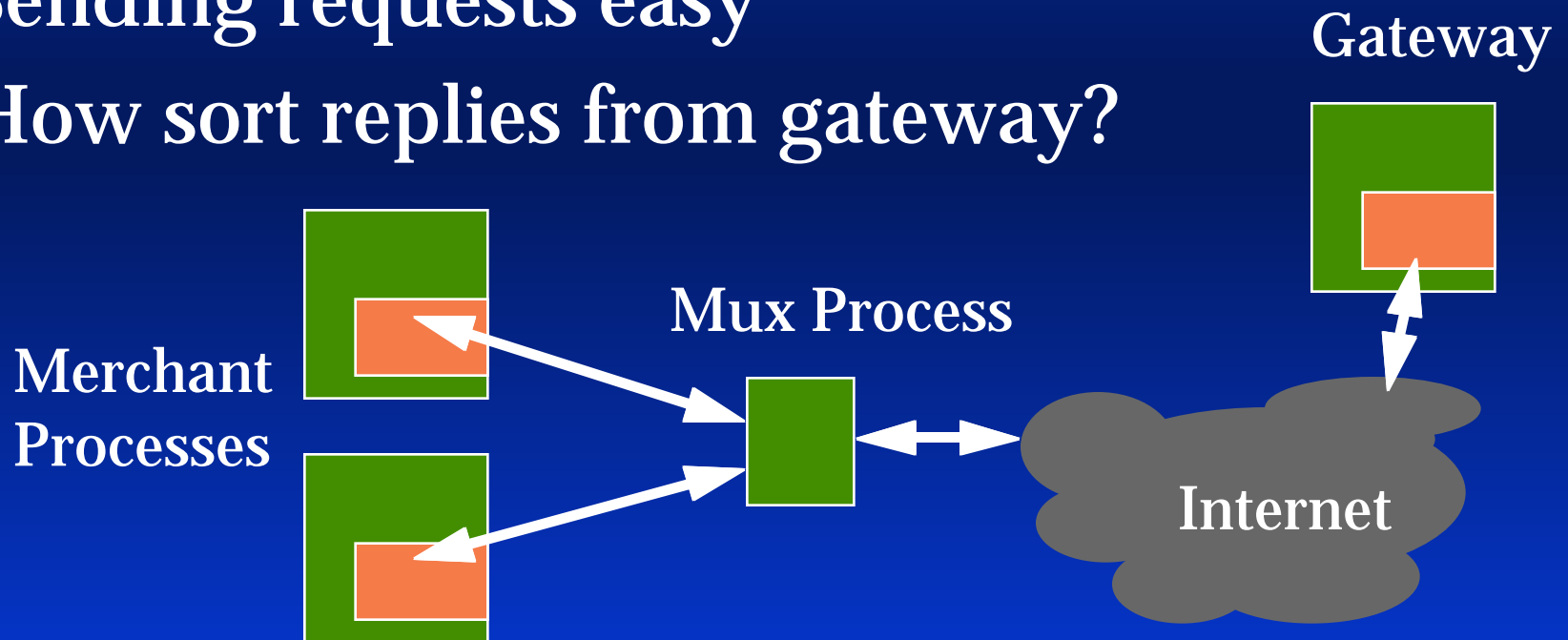
# Multiple S/PAY Processes

- Works if database handles it  
(SX\_StartTransaction)



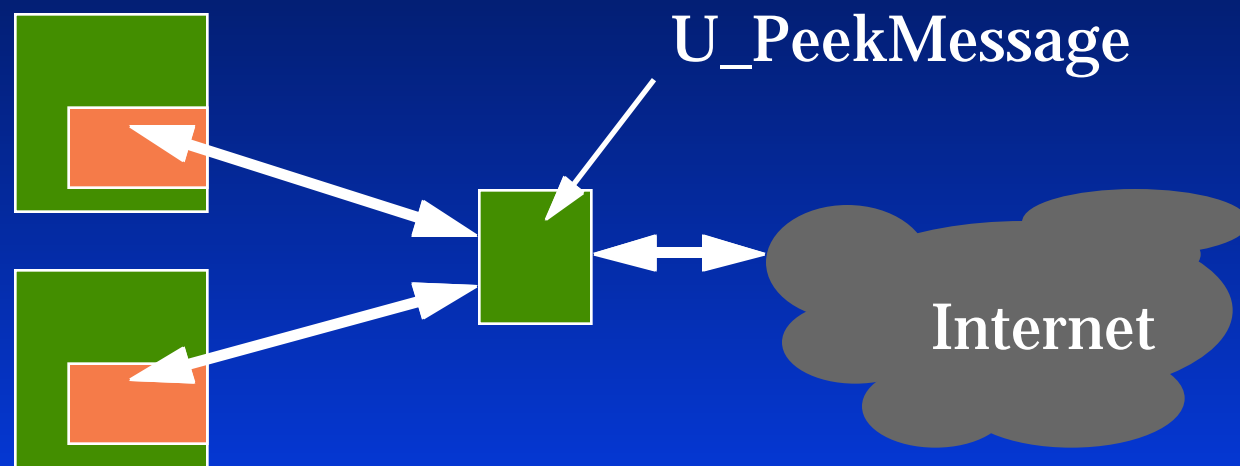
# Multiplexed Communication

- S/PAYs share comm link to Gateway
- Sending requests easy
- How sort replies from gateway?



# Multiplexed Communication

- U\_PeekMessage
  - » LIDs, RRPIDs, MsgType, etc.
- Option 1: LID\_M includes process ID
- Option 2: Track RRPIDs





# New SET-like Message

- Not allowed
- “Set, the whole SET and nothing but the SET”
- Contact RSA
- Ex: Japanese Bank Xfer



# Conclusions

---

- S/PAY, RSA's SET Engine
- Complete, High Quality, Easy to Use
- Flexible (Set-Top to Mainframe)
- Fast Access to World Wide Market
- Quickly Track Evolving SET Standard



# Using S/PAY

Dr. Robert W. Baldwin  
RSA Data Security, Inc.  
[baldwin@rsa.com](mailto:baldwin@rsa.com)

RSA Data Security Conference  
January 30, 1997

