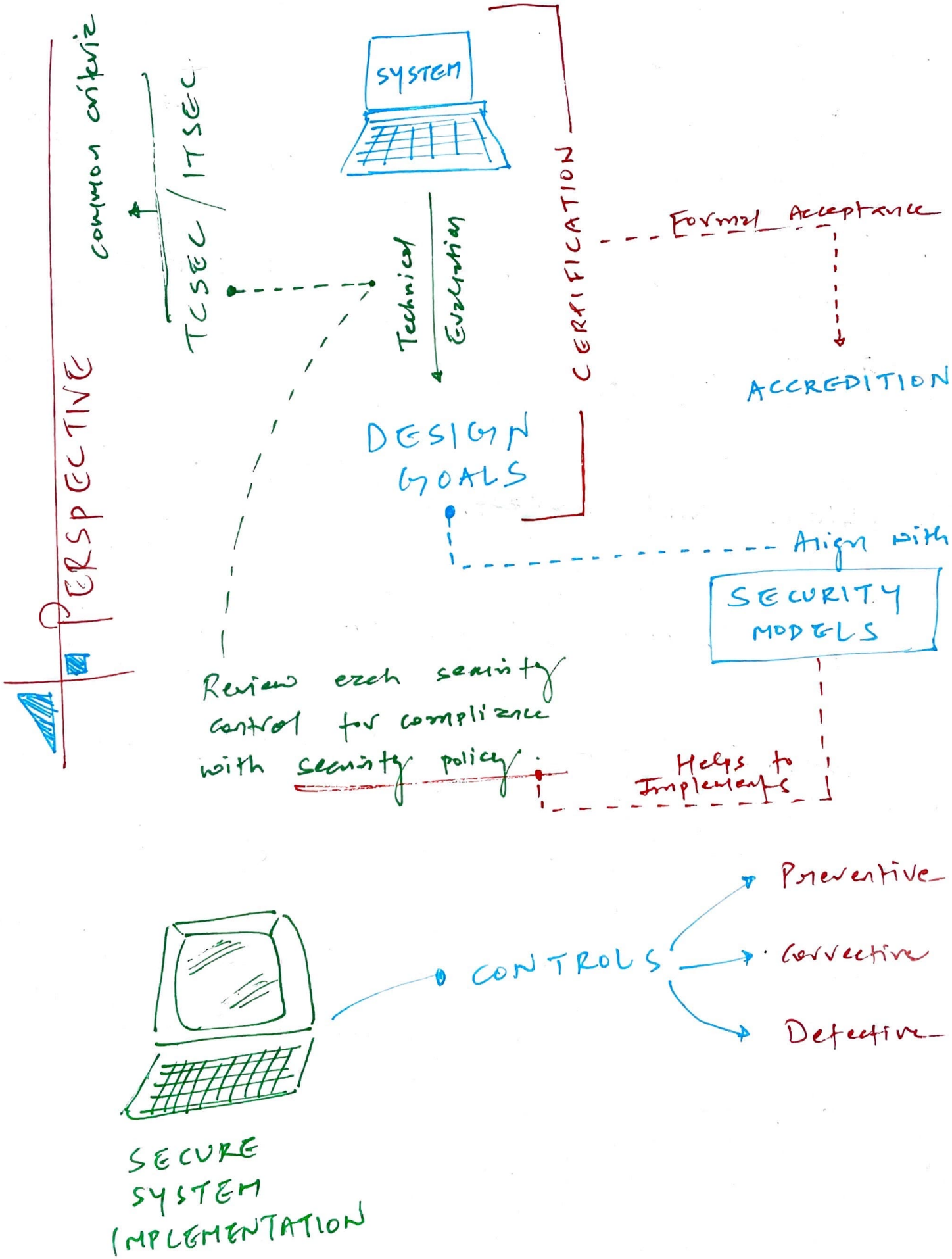
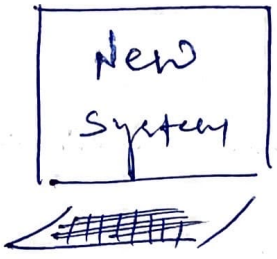
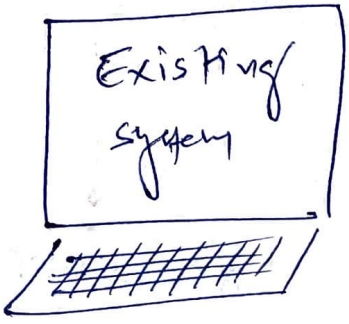
CH: 8 PRINCIPLES OF SECURITY MODELS, DESIGN, AND CAPABILITIES



IMPLEMENT & MANAGE ENGINEERING PROCESS USING SECURE DESIGN PRINCIPLES

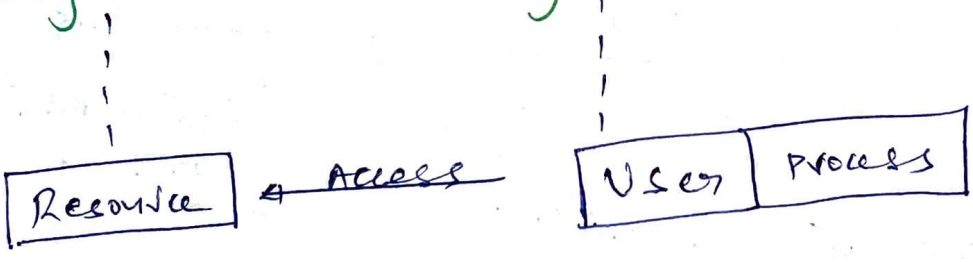


← Build security : EASY

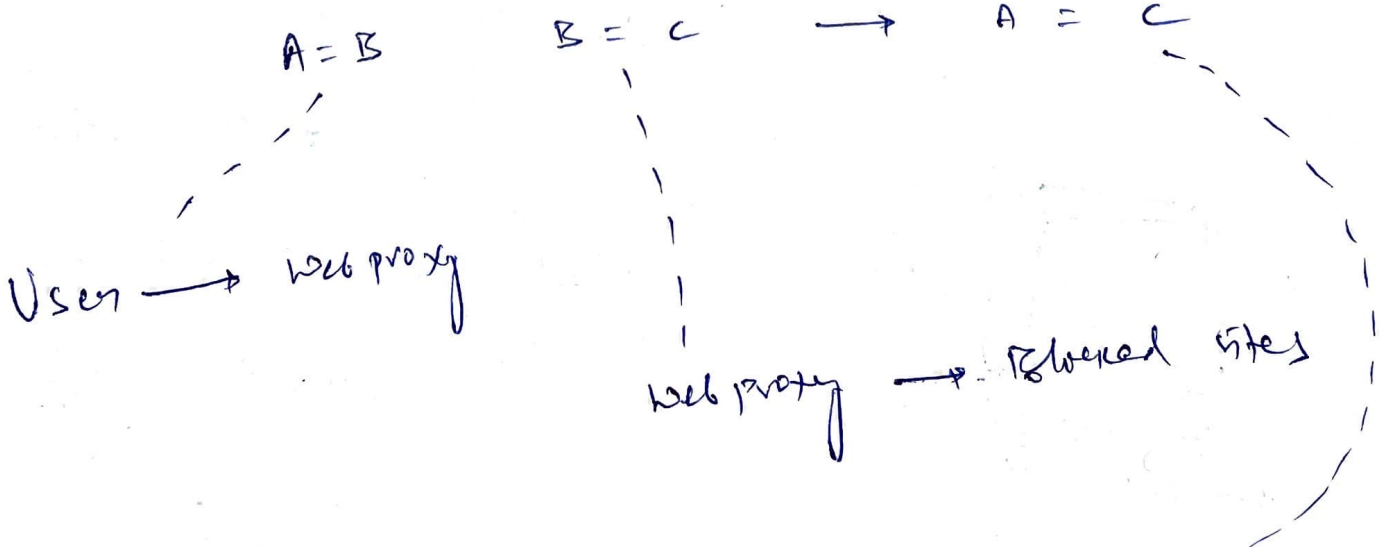


← Add security : DIFFICULT

* objects and subjects



Transitive trust = serious security issue



* closed and opened Systems

- Proprietary + same manif.
  + Not easy to integrate
  - Secure

- Industry standard + Easy to integrate
  - Not secure

Open Design Methodology — refers to Kerchoff's principle — Security of design shouldn't be secret like cryptosystem — this approach of open design provides independent security confirmation

* open-source vs. closed-source

- open-source: Internal logic + source code open to public (LINUX)
- closed-source: Hidden from public (WINDOWS)

Both can be either open or closed system.

* Techniques for ensuring CIA

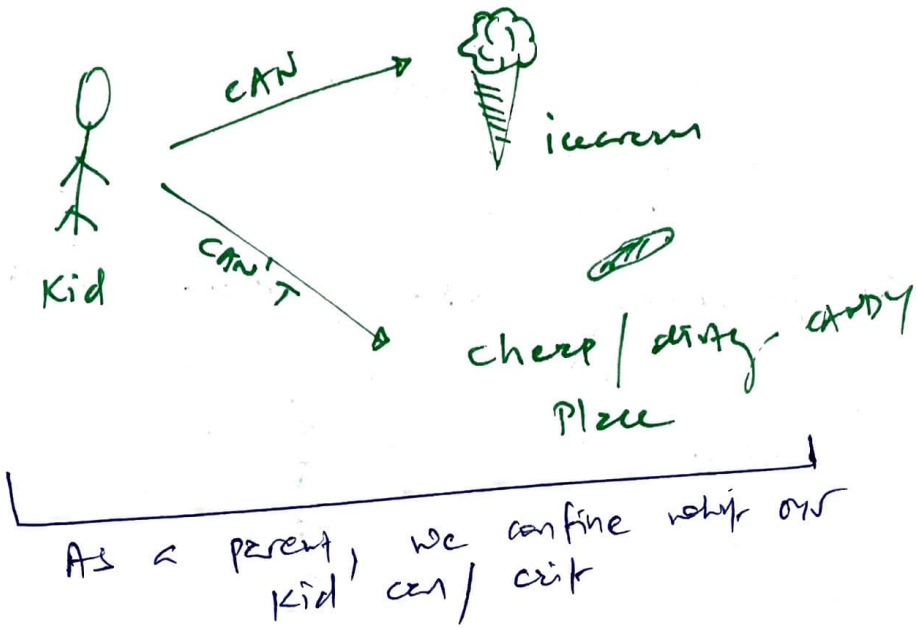
(from s/w designer/ perspective but applies to all areas of security)

- Confinement
- Bounds
- Isolation
- Controls
- Trust & Assurance

CONFINEMENT --- Sandboxing

- Restricts a process or software program to reading from or to certain memory locations.

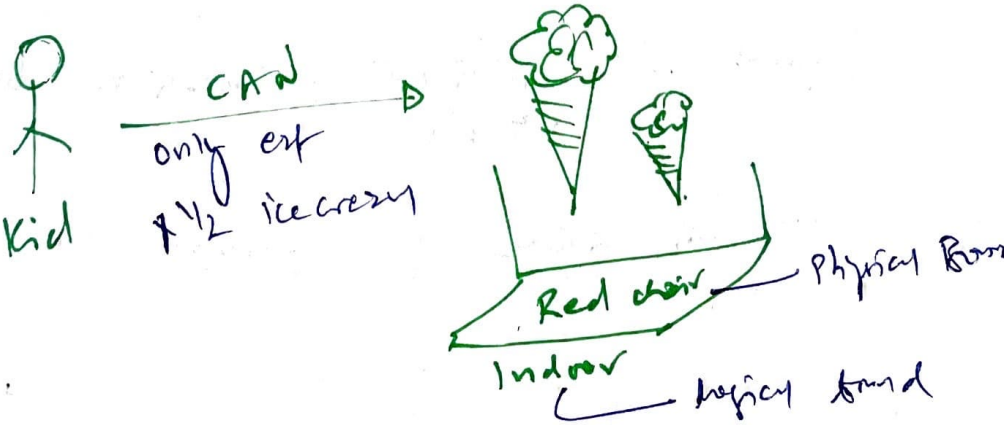
Confinement in s/w development is concept of isolation to ensure that running process / Application cannot interact with other entities unless allowed to.



BOUNDS --- physical
--- logical

- Bounds are the limits of memory a process cannot exceed when reading or writing.

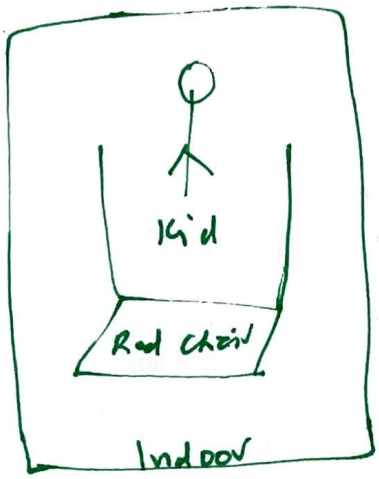
Bounding checking restricts process to specific areas of memory but doesn't prevent external inputs from interaction.



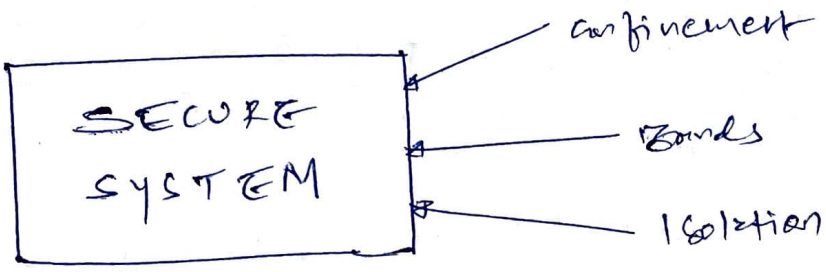
- Bound is that area where process (kid) is already confined.

ISOLATION ---- mode

- where process is confined with memory bound.



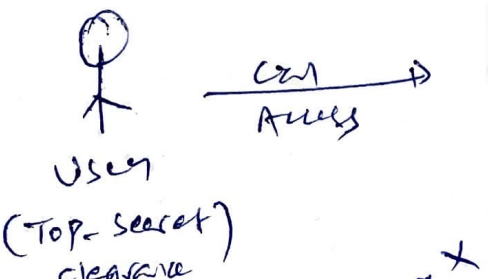
<---- isolated / safe from external toxic environment

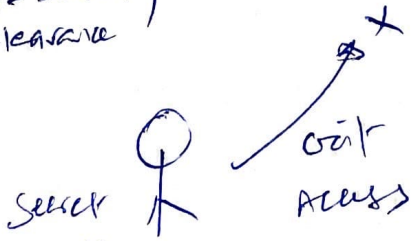


CONTROLS ---- MORE Ch: 14, 629-633

MAC - (p.1.0 End)

- Subjects & objects have labels / classification





DAC

- check Access control matrix model for example

- All objects have owners
- owner have full permission
- owner can also delegate to data custodian for day-to-day operations.

GOAL

- To limit access to object & subject
- To protect confidentiality & Integrity of data

TRUST & ASSURANCE

SECURITY → ✗ Not an afterthought

↓ ✓

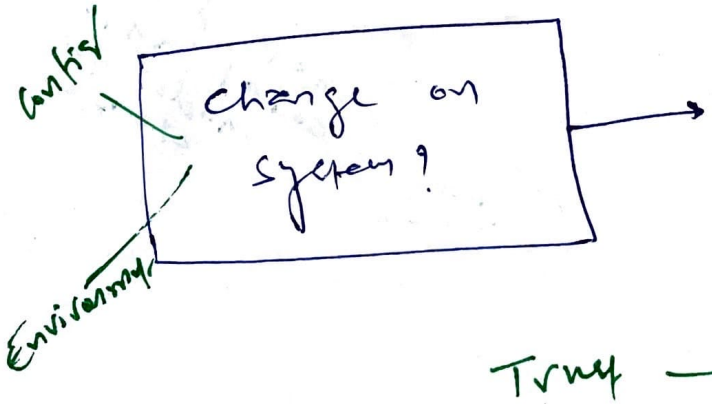
Building (once security integrated to design)

IT MUST BE

Design → Engineered → Implemented → Tested → Audited → Evaluated → Certified → Accredited

(so far I did this)

now



Trust — Assurance

Trust: Implementing specific security features for the system

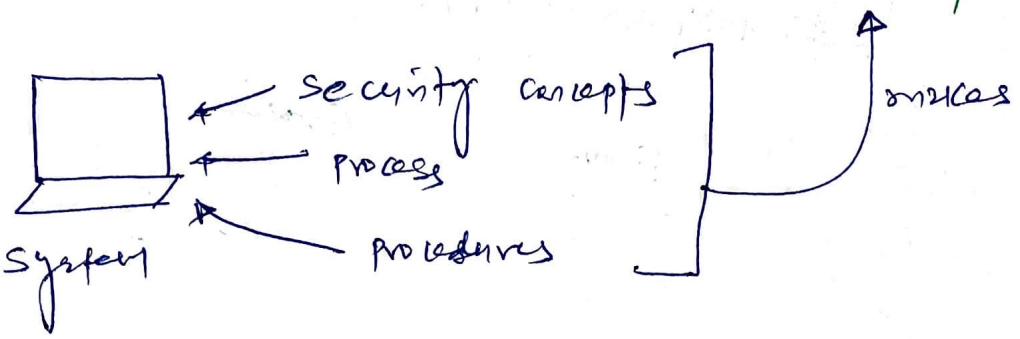
Assurance: Assessment of reliability & usability of security feature in real world situations.

TRUST BUT VERIFY.

Never trust. Always verify. ZERO TRUST

FUDAMENTAL CONCEPSTS OF SECURITY MODELS.

Provide a way to formalize SECURITY POLICIES.



These models offers deep understanding on how OS should be designed & developed to support a specific security policy.

Tokens → Every object is seperate token associated with resource. Token has security information about object prior accessing access to actual object.

Capabilities → ~~Table, that which subject can access which objects.~~ Row that maintains ~~security attributes~~ for controlled object

Security Labels → Permanent label attached to object, can never altered.

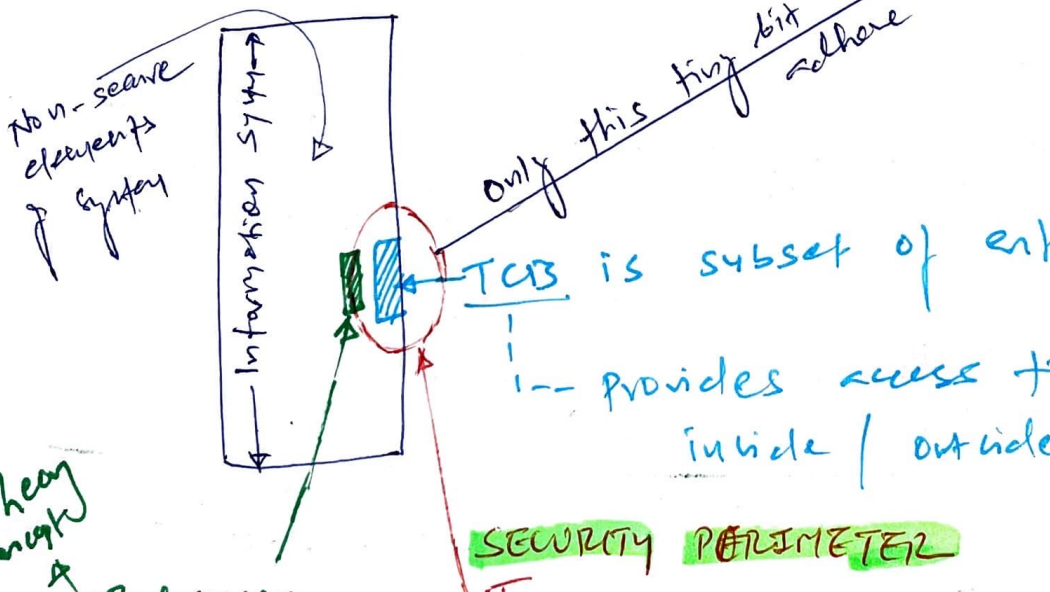
→ Provides safeguard against tampering that token & capabilities never provide.

* TRUSTED COMPUTING BASE (TCB)

orange book / TCSEC
(Trusted computer system Evaluation Criteria)

P.T.O END for REF. MONITOR CONCEPT

Software + Hardware + controls ⟹ forms TCB ⟹ Enforce Security Policy



1. STATE MACHINE MODEL

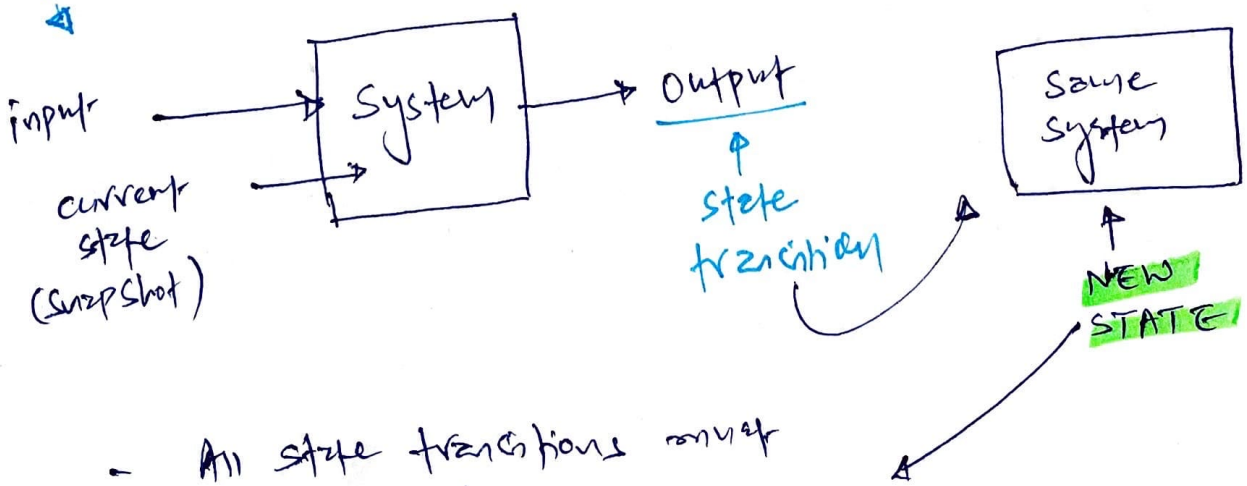
Always boot in secure state

All instances of subjects → MUST BE SECURE → objects

- Based on FSM (Finite State Machine)
  - system is always secure, no matter which state is in

Snapshot of System = Requirements of security policy

Secure System



- All state transitions must be evaluated.
- New state will be checked against security policy, if it's compliant = secure state.

2. INFORMATION FLOW MODEL

Biba
Bell-La Padula

Prevents → Unauthorised
Inseecure
Restricted

Information Flow. b/w diff. level of security

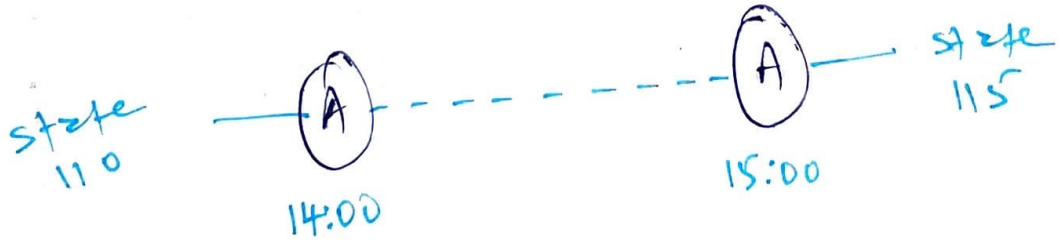
Model Focus → Information flow
Type of flow.

Address COVERT CHANNEL by excluding nondefined flow pathways.

Allows authorised information flow b/w subject & object within same classification or different classification level.

Interesting

Establish relationship b/w two ~~object~~ versions of same object at different point of time.



→ Biba + Bell-La padula = MAC Models

* 3. NON-INTERFERENCE MODEL.

Action of Subject A : Higher classification → should not affect X

Action of Subject B : Lower classification

Provides protection

← if this happens, Subject B will be placed into insecure state

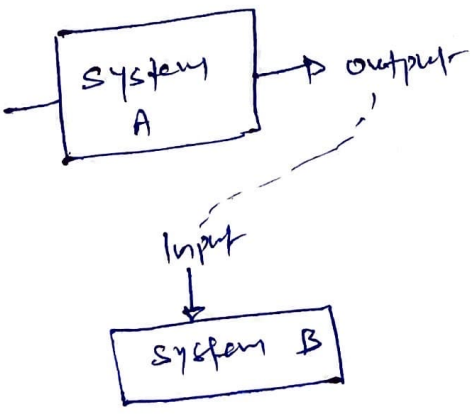
→ This type of Information leakage creates COVERT CHANNEL

---

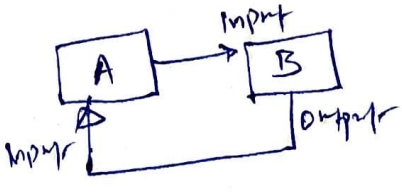
* Composition Theories

- Kind of Information flow model
- Info. flows b/w systems rather than individual system

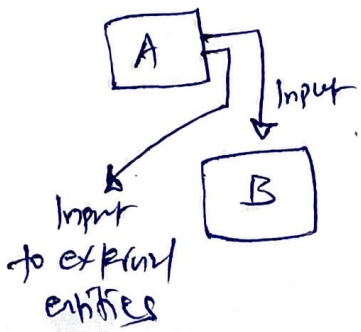
Cascading



Feedback



Hookup



4. TAKE-GRANT MODEL

Focus: Dictates how rights are passed from one subject to another or from subject to object.

2 Rules

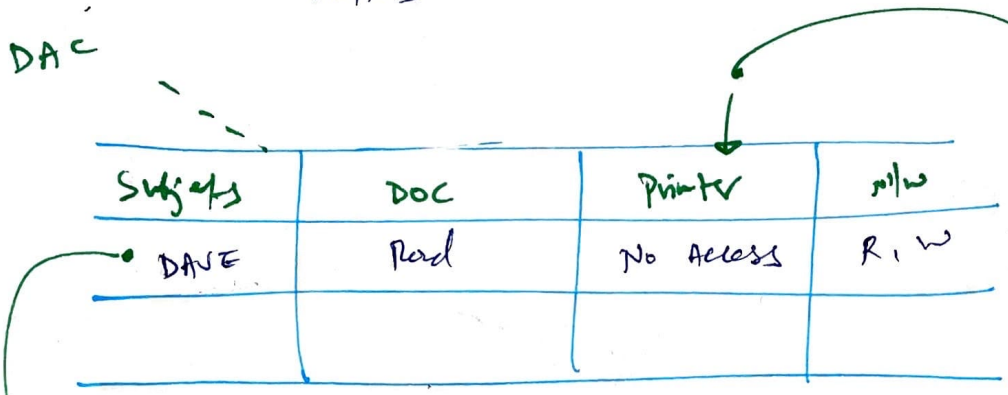
- Take Right
  - from subject to other subject / object
- Grant Right
  - from subject to other subject / object

subject → subject
subject → object

5. ACCESS CONTROL MATRIX

Just a table, lists all subjects & objects. What subject can perform on objects

actions



Row = Capabilities list

Tied with subject, list actions it can perform on object.

Column = ACL (Access control list)

Tied with object, list valid actions each subject can perform.

Need both for ease of management

Column → Object

Row → Subject

FIRST REMEMBER THIS

Addresses only confidentiality of Data

6. BELL-LAPADULA MODEL

Info. Flow model
state machine concept

for Federal

→ To protect classified information

- Prevents information flow from low to high security level.

- Prevents information leak from classified level to less secure clearance levels.

HOW!

Higher. classified objects

BLOCKS

lower. classified Subjects

bell-lapadula (contd.)

Employs STATE MACHINE Concept

LATTICE CONCEPT

3. properties
To maintain confidentiality

1. Simple security Property — can read at your level and below but not up
   NO READ-UP
2. * (star) security property (confinement property)
   NO WRITE-DOWN — can write at your level and up but not down

Prevent information leakage from top to bottom

3. Discretionary Security Property (strong *) — only read and write into into your own security level - no up or down
   Enforce DAC

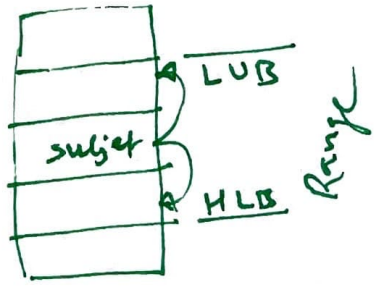
Enforce subject need to know basis in order to access object based on access matrix

- Only supports "C", no "I" & "A"
- Assumes security transitions b/w layers
- No support for covert channel.

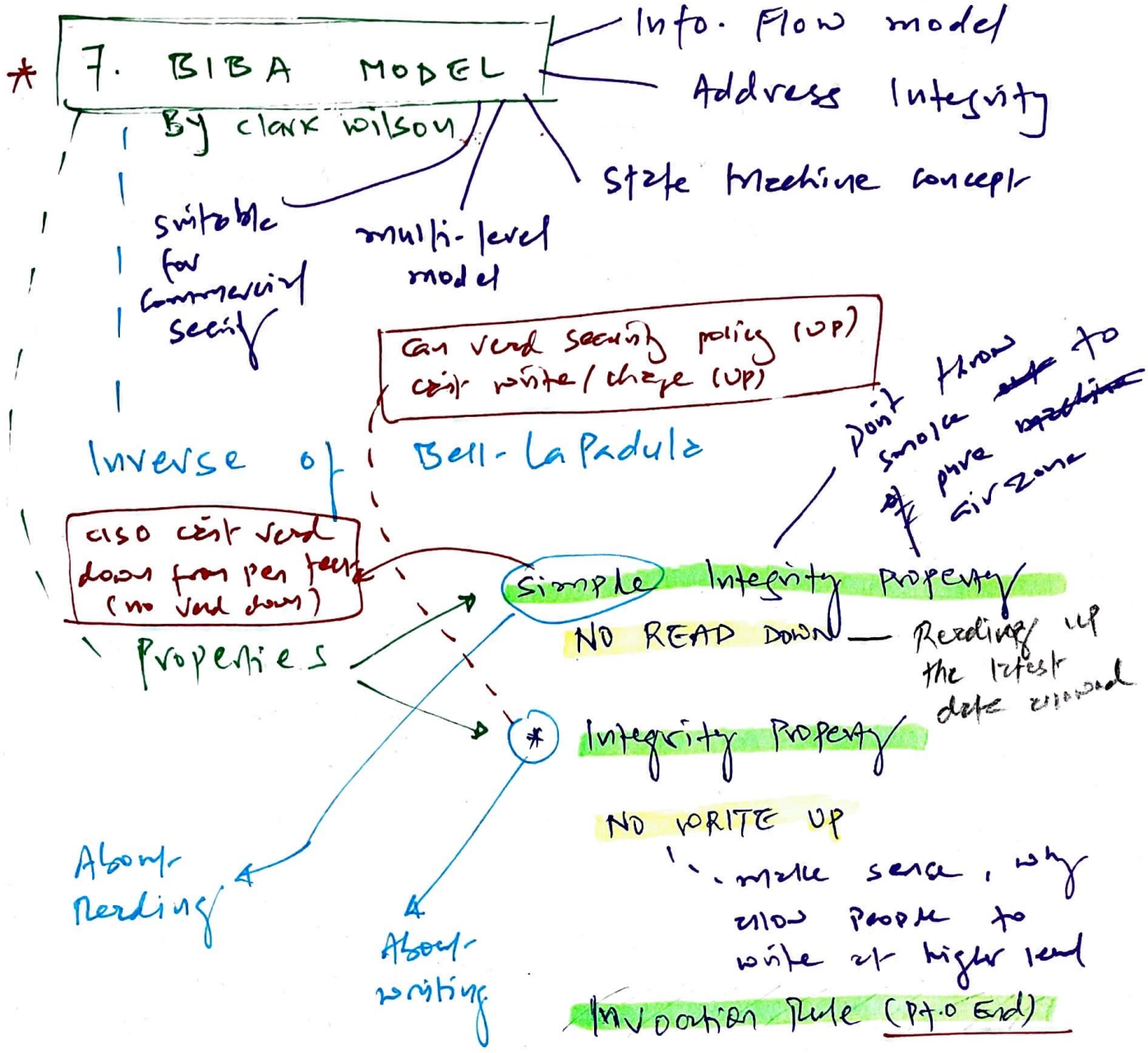
Lattice-Based Access control

- Subjects are assigned with lattice.

- subject can only access objects that fall into range b/w
  Least upper bound & Highest low bound.



spy wants to convey message to previously he can write up



* Biba address three integrity issues:
(3 rules of Integrity)

(1) Prevents unauthorised subjects modifying objects

(2) Prevents authorised subjects modifying unauthorised objects

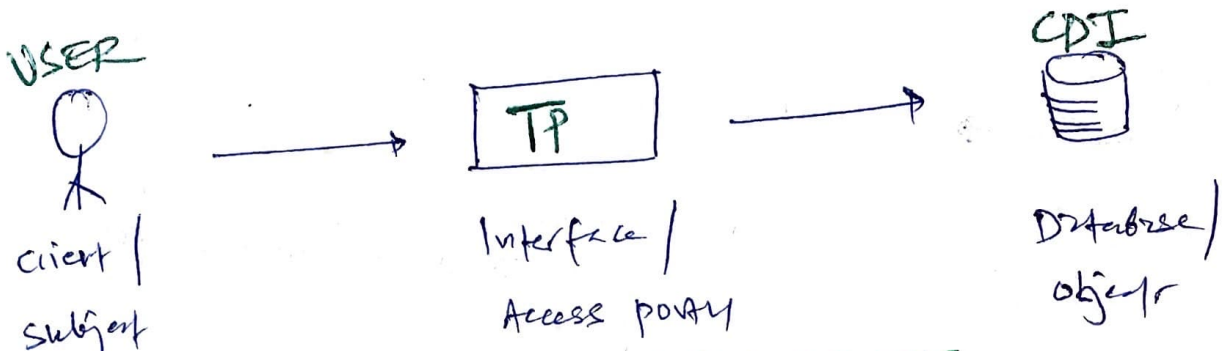
(3) Protect internal & external object consistency.

* 6. CLARK-WILSON MODEL — Data Integrity

For commercial Apps.

- No state machine or lattice structure

Instead use three-part-program (triple)



Access-Triple - subject - program - object

(P.T.O End for revised diagram concept)

Enforced Separation of Duties

* Restricted interface model

- Uses security labels as classifications

Well-formed transactions

- Each subject can access object via portal (TP)

- like a program

- Each program has limitations on what it can / it cannot do to object, Hence, limiting subject's capabilities.

- One subject (Confidential)

- separation

- one set of Data + Function

- second subject (Restricted)

- Another set of data + Function

9. BREWER & NASH MODEL

Integrity model

Chinese wall

Dave → Syber → ANZ

Dave — Access to proprietary Data — Employment

should not have access to competitor

× Commonwealth

× NAB

× Westpac

Principles

other perspective

Data Isolation

- Keeps users (Dave) out of potential conflict of interest

Admin ← Full control

if conflict with Data they

Admin ← Access blocked temporarily only for conflicted data

when resolved

Admin ← Full Access

10. GOGUEN - MESEGUER MODEL.

Integrity

Foundation of Noninterference model

Based on

Automation Theory

- predefined subjects
allows predefined actions
against predefined objects

Domain seperation

- similar subjects grouped = Domain 1

× NO access to other domain

11. SUTHERLAND MODEL

Integrity

Based on state machine & Info. flow model

Prevents interference

Integrity maintained through PREDEFINED SECURE STATES & interference is prohibited.

EXAMPLE

Prevents COVERT CHANNEL

OSU pg 385

- Path / channel normally not used for communication = exploit, violate, bypass security policy

OVERT CHANNEL = known, Expected, Authorised, Designed, monitored, controlled

12. GRAHAM - DENNIG MODEL

Secure creation & deletion of objects and subjects using eight primary protection rules or action.

SELECT

SELECT CONTROLS BASED ON SYSTEM SECURITY REQUIREMENTS

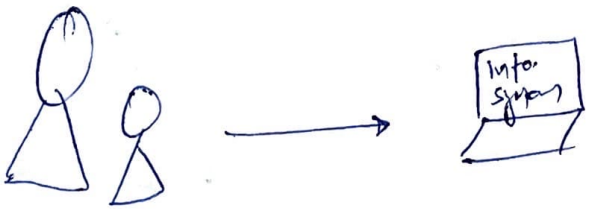
Buying : 2 step evaluation process

System testing + technical evaluation, meets security capabilities for intended use

Compare system with secure design, individual response, performance

↓

Those changes = ReTest criteria



Management's responsibility to accept RISK when system is approved & Deployed.

3 main product evaluation models

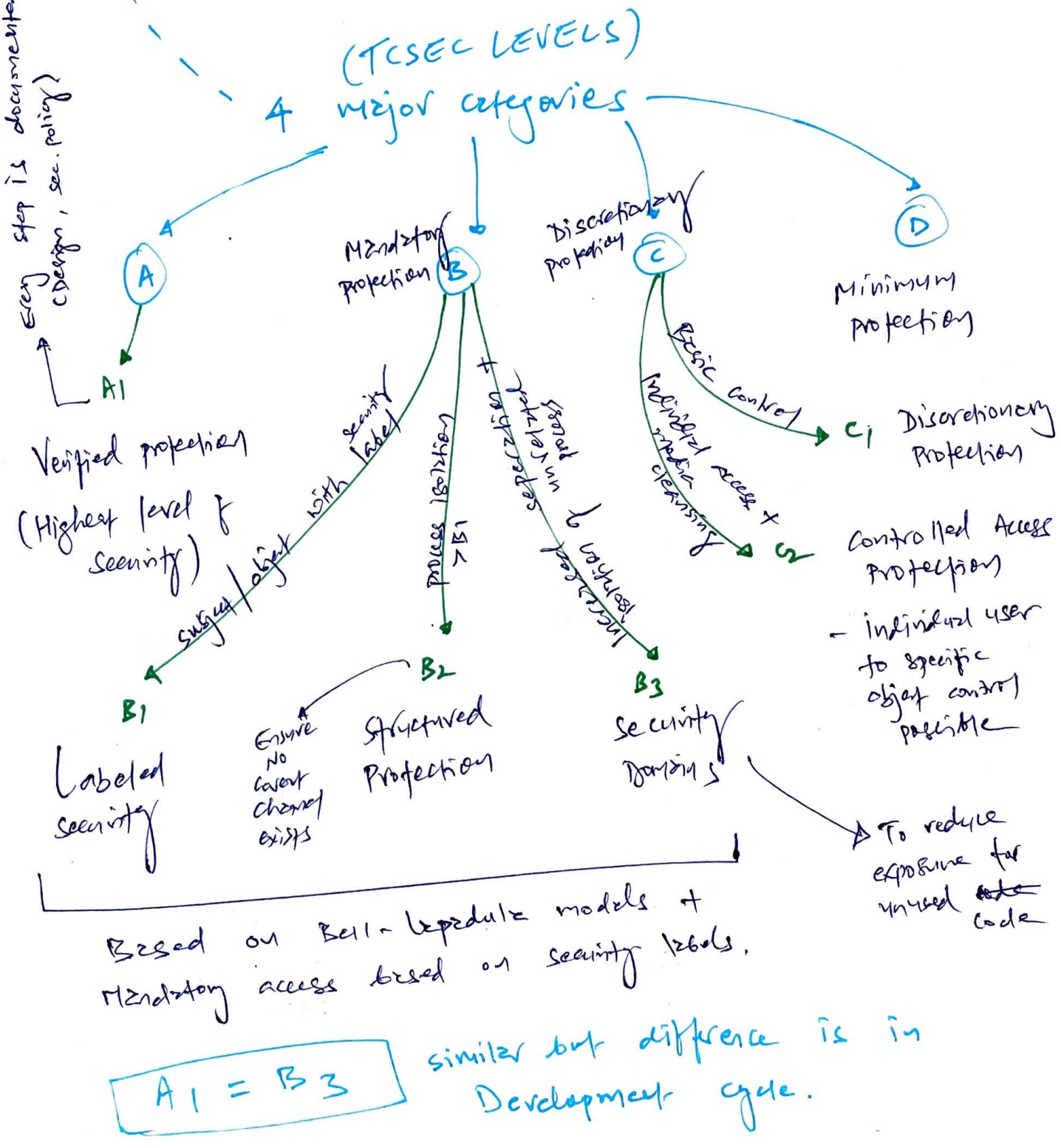
- TCSEC (1980s)
- ITSEC 1990-1998 : European model
- Common Criteria 1998 - replaced TCSEC + ITSEC

2005

1. TCSEC classes & Required Functionality.

TCSEC only Address confidentiality.

Trusted computer System Evaluation Criteria (orange book)



A1 = B3 similar but difference is in Development cycle.

* Rainbow series

Orange book : Applies to stand-alone computers, not attached to a network.

Red book : Interprete TCSEC in networking context

Green Book : Provides password creation & management guidelines.

* Problems with TCSEC → Replaced with common criteria in 2005

- Focus on C, Not I
- Doesn't address personal, physical & procedural policy to implement full security policy.

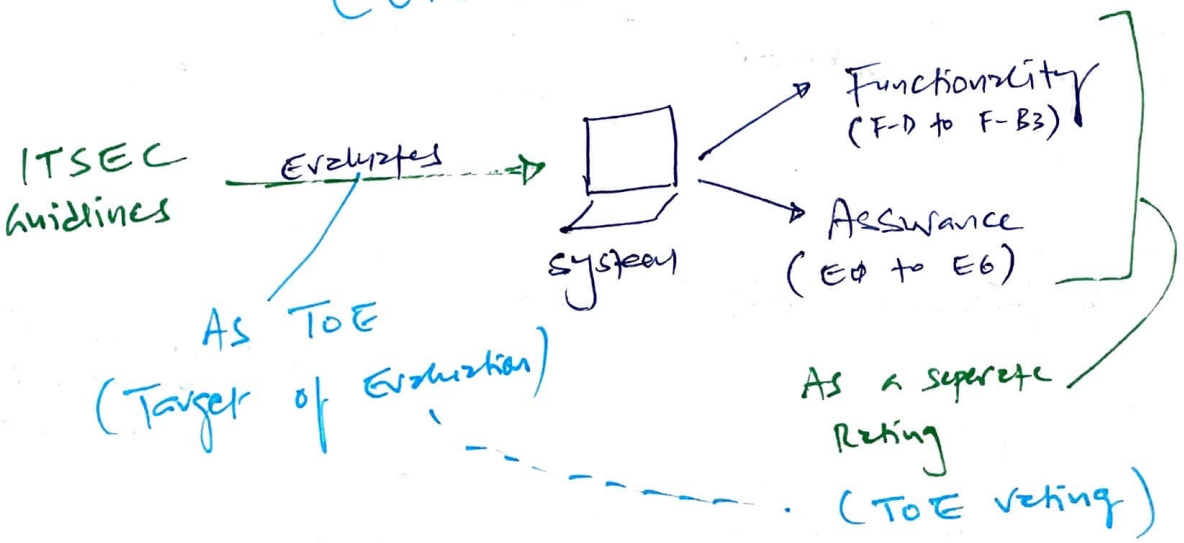
side notes, not bad.

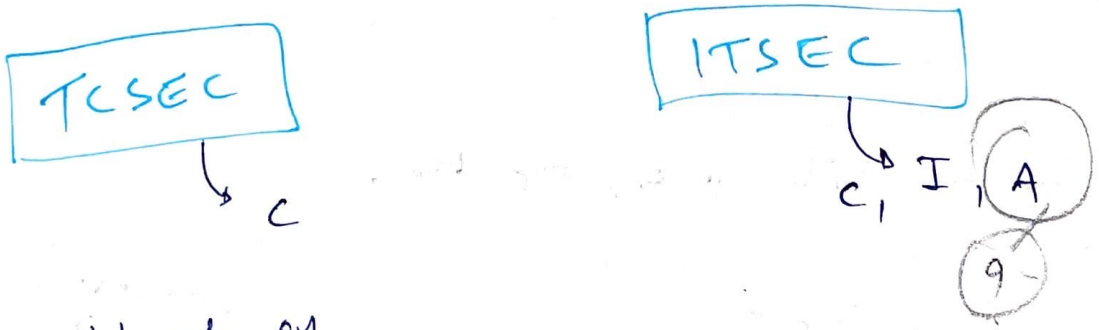
TCSEC	Common Criteria (CC)
- Address only confidentiality	- Address CIA
- US Domestic	- International
- Narrow scope	- Broader Scope
- Rigid model	- Allows flexibility
-	

2. ITSEC classes & Required Assurance & Functionality.

Information Technology Security Evaluation Criteria

- Initiated for evaluating Security criteria in EUROPE





- Doesn't rely on TCB (system components doesn't need to isolate within TCB)
- changed system requires to be Reevaluated → don't need it.

FIRST PTO for 10,000 feet AT END

3. COMMON CRITERIA (CC)

Helps customer buy with confidence

→ Designed as a PRODUCT EVALUATION MODEL

> High CC rating ≠ secure system, free from VULS.

→ official standard: ISO 15408
(Evaluation criteria for Info. tech. security)

— CC process based on two elements:

Protection Profiles (PP)

- For product evaluation that meets security requirements & protections

↓

SECURITY DESIRE
("I want" from customer)

Security Targets (ST)

- Implemented security measure from vendor built into target of evaluation (TOE)

↓

"It will provided" from vendor for specific system such as Cisco ASA

Customer PP — Vendor 1 ST (TOE)
— Vendor 2 ST (TOE)
— Vendor 3 ST (TOE)

↓

closest / best match is client purchase