ADVANCED COMMUNICATIONS
CONCEPTS INCORPORATED

Tactically Unbreakable COMSEC
(TUC)

The Use of Reconfigurable Logic to Secure Data in Motion & Data at Rest
System Hardware & Software Integrity and more...

What TUC is

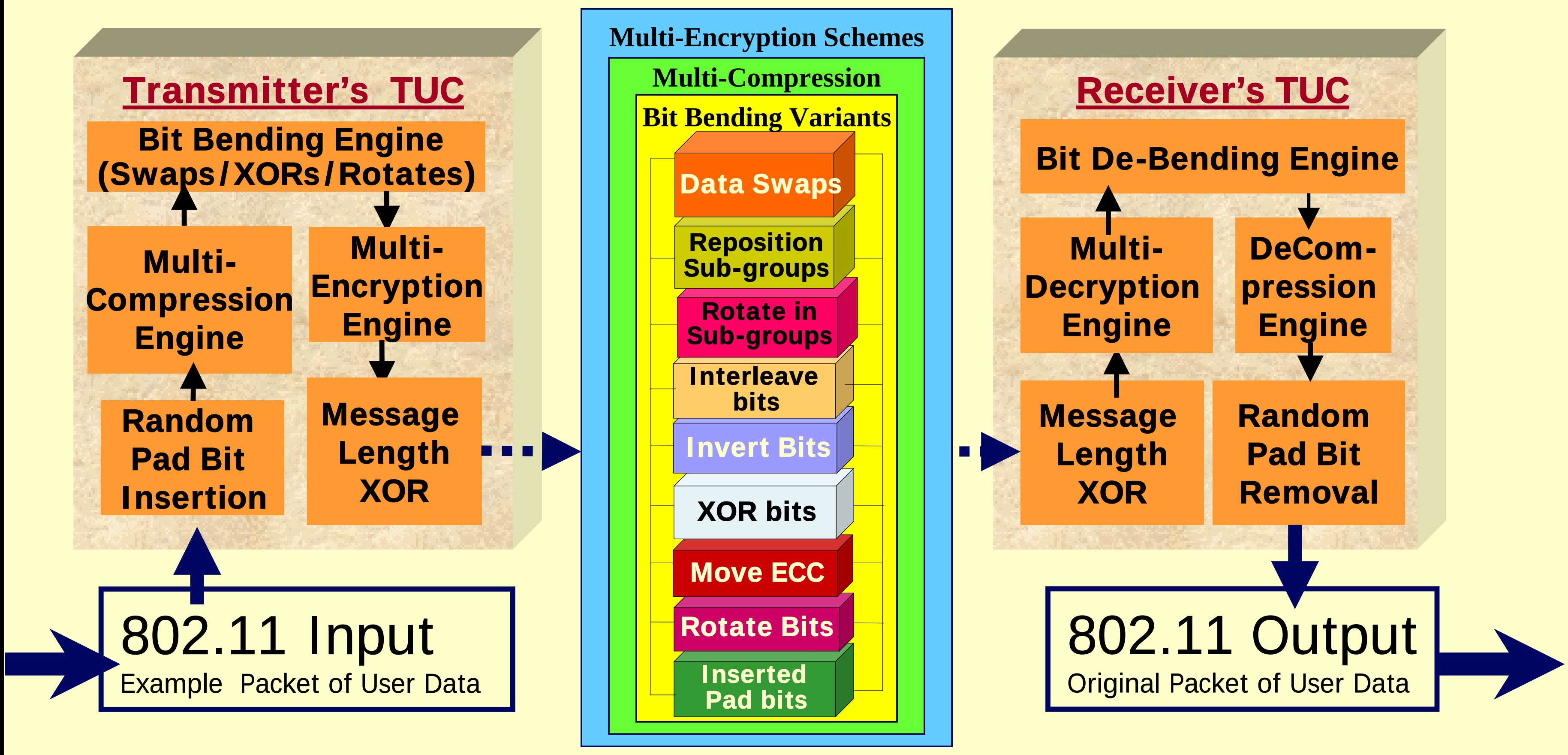
Tactically Unbreakable COMSEC

- What TUC is :
  - A Reconfigurable Logic (RCL) based Protocol Hopping security structure
    - RCL is hardware circuits than can be “reprogrammed” on-the-fly
    - Protocol Hopping is a super-set of spread spectrum’s Frequency Hopping
  - Single product solution for :
    - Data-at-Rest (FISMA, DoD 8500 series)
    - Data-in-Motion (HIPAA, ….)
    - Provider Remote Access (Customer Demand, HIPAA)
  - Operates at OSI Level 2, transparent to Applications & Physical layers
    - Application independent – no program needs to be modified to use TUC
    - User independent – no operator action (or awareness) needed to use
  - Layered security protection (& QoS) for wired & wireless comms
  - A powerful enhancement to traditional software-only implementations of Multi Level Security schemes
  - Cheap & easy enough to put in EVERY Device/Form Factor, be it laptops, computer nets, PDAs, WLANs, etc.

How TUC works

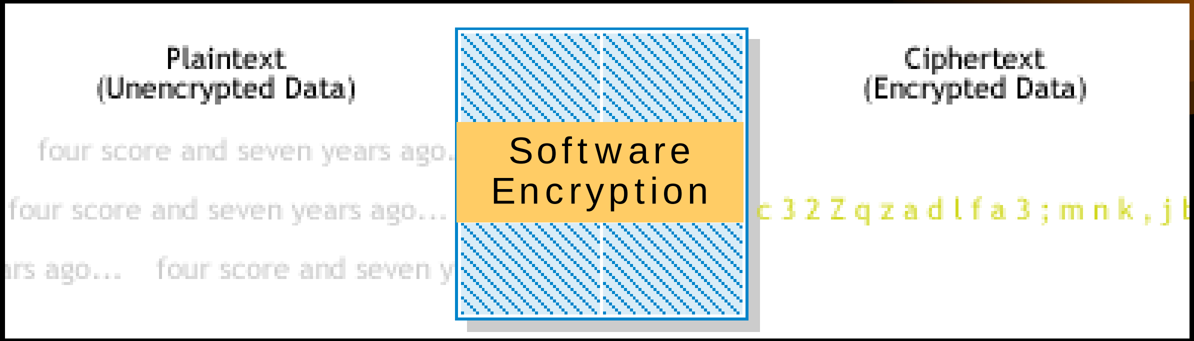
Tactically Unbreakable COMSEC (TUC)

Complexity Layering on a per packet basis

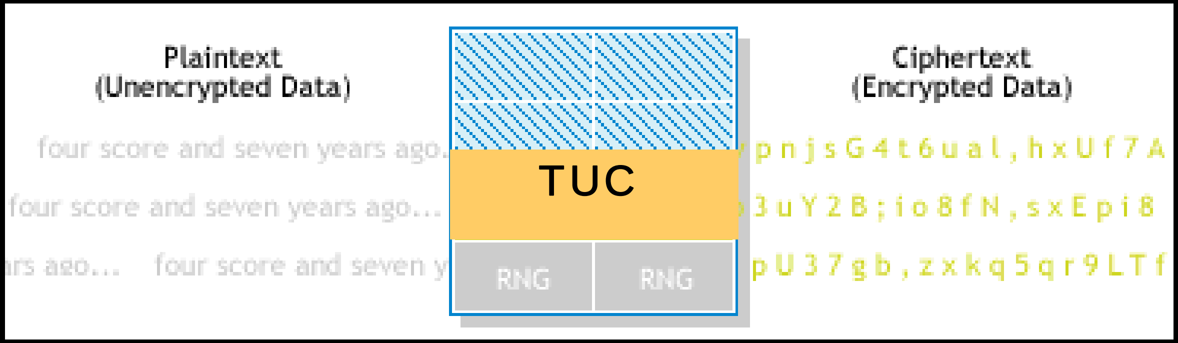


Hardware vs. Software Execution Speeds

Software implementations bog down the host processors and severely limit the speed of the channel that can be protected

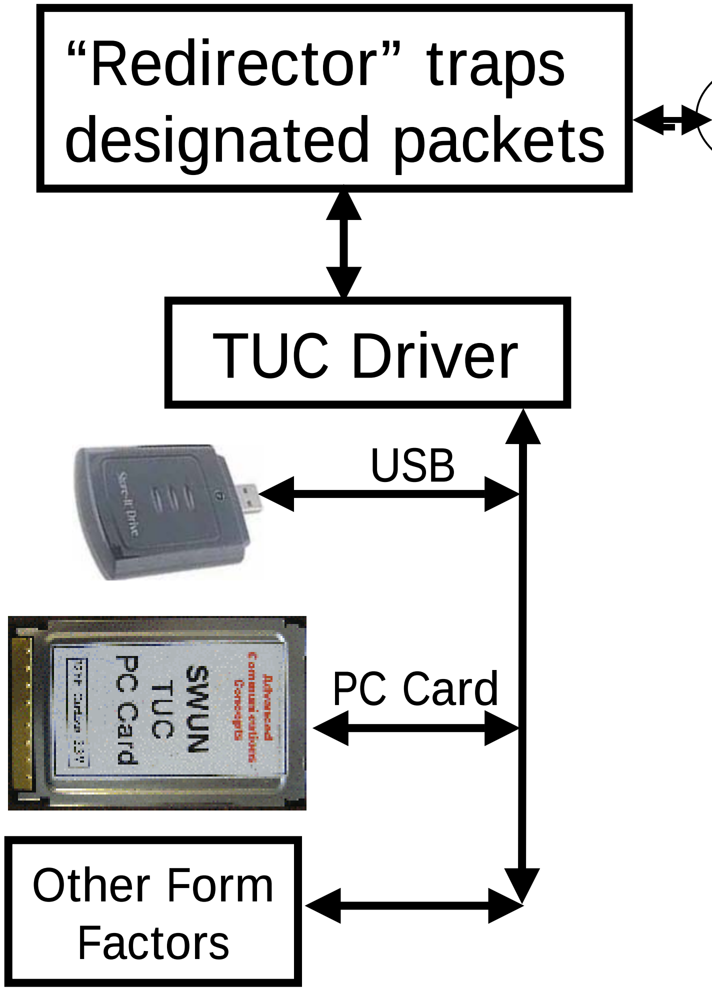


Anybody think software will keep up with a 100 Megabit Ethernet?



TUC operates at full hardware circuit speeds -- because it IS hardware

How TUC protects "Data in Motion"



Windows Comms Example

One Networking module:

NDIS

All Application data flows to/from device drivers through NDIS

- Table Driven Packet designation by:
  - I/O Device (802.11 / Ethernet)
  - Packet Source/Destination Address
  - "Port type" (ie 80=HTTP, etc)
- All networking comms types covered
  - Wireless
    - 802.11/Bluetooth/UWB/etc.
  - Wired
    - 802.3 / Token Ring / others
  - Protocol insensitive
    - IP - TCP/UDP/etc
    - ATM/SONET/etc.

OSI Model – where TUC fits

OSI Layer	Function	Example
7 - Application	Provide network access to user and system applications	HTTP, SMTP, Telnet
6 - Presentation	Data Transforms for Conversion, Compression and Encryption	MIME, Secure Socket Layer, XML. EBCDIC<->ASCII, MPEG
5 - Session	Maintaining Connections, "Sessions"	NetBIOS, Port Numbering , X.225
4 - Transport	Insure data delivery end-to-end	TCP, UDP, SPX and NetBEUI
3 - Network	Data frame routing to logical address	IP, IPX, Routers
2 - Data Link	Physical addressing of transfer data units (frames) and error checking	X.25, AppleTalk, SDLC, PPP & SLIP, bridges
1 - Physical	Connection, contention & flow control	Cables, hubs & network adapters, RS-232, T1, 10BASE-T

Increased Security

TUC operates at the Data Link level with hooks into the Network Level

WHAT THIS MEANS :

Applications are UNAWARE of TUC – no special coding needed !

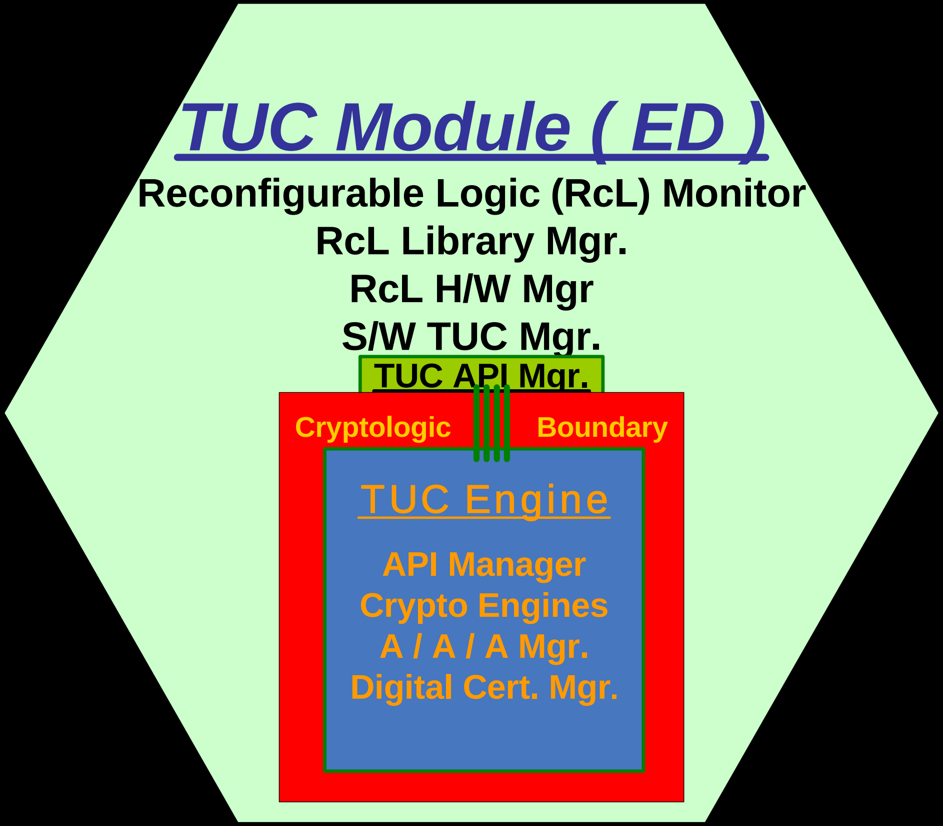
Applications & Utilities can't by-pass TUC – can't avoid security!

TUCNet Data-in-Motion Demo

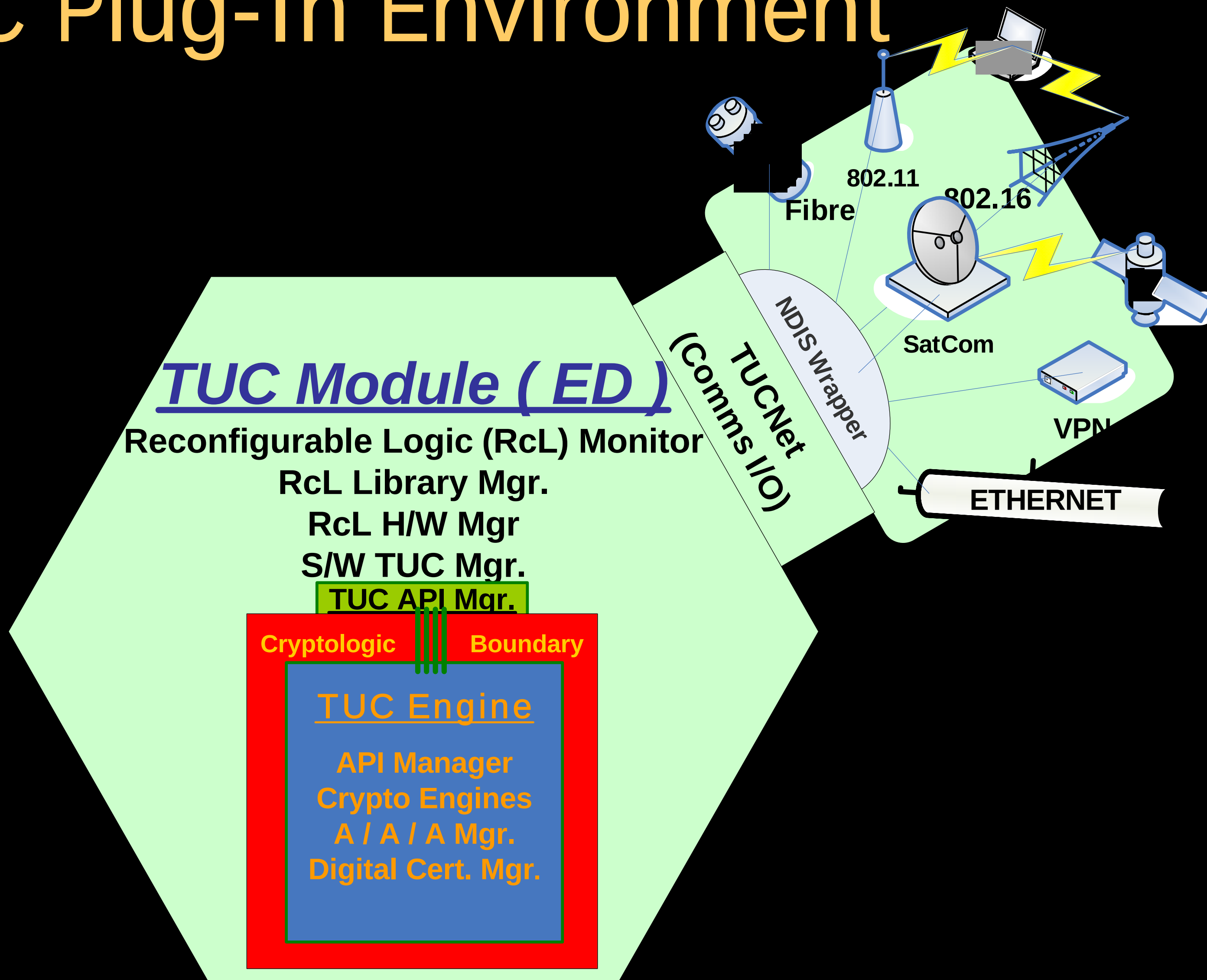
TUC Plug-In Environment

All TUC applications include the TUC Core Module

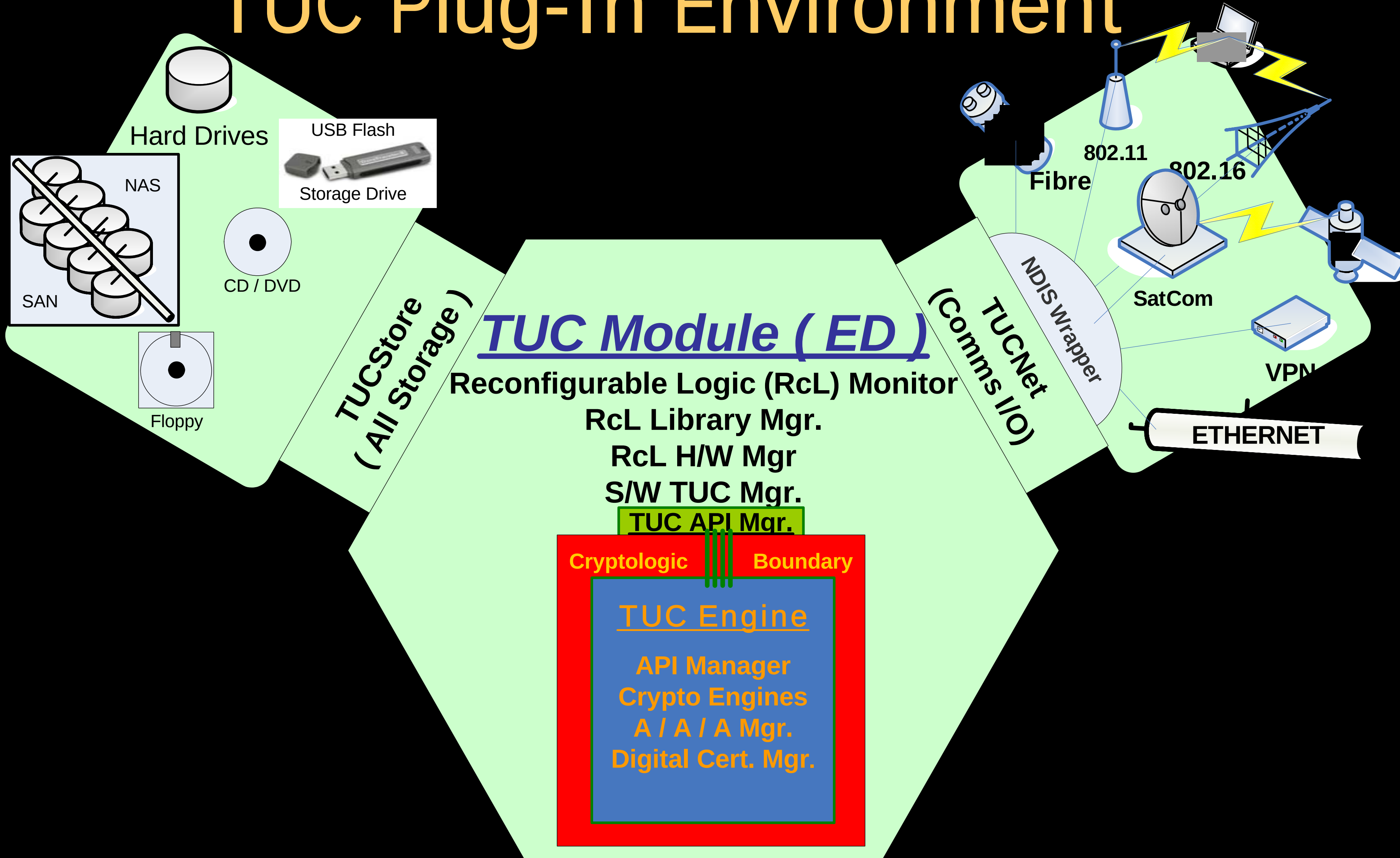
(each installer checks to see if Core already installed)



TUC Plug-In Environment



TUC Plug-In Environment



How TUC protects "Data at Rest"

Trap all Storage Read/Write Calls

TUC Driver

USB

PC Card

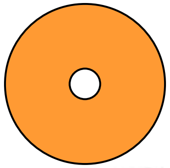
Other Form Factors

LINUX Storage Example

Two Main Entry Points:

- Floppy (just floppies)
- SCSI (everything else)

- SCSI/FC drives
- ATALib
  - ATA (IDE) Drives
  - ATAPI devices
    - CD/DVDs
    - USB



TUCStore Data-at-Rest Demo

Key Attributes of Data@Rest protection of Software and Data Integrity

1. Encryption of storage files is the minimum requirement
   – but not nearly enough for true Data-at-Rest Security
2. TUC has an independent 3rd Party "Trusted Processor" that can not be corrupted or accessed by hackers
   1. On-FPGA PowerPC, physically and programmatically isolated
   2. All "Trusted Processor" programming, data & buffers isolated
   3. Digital Certificates and Public/Private keys are stored on Trusted Processor TUC module – totally physically isolated
   4. Hackers can not access or corrupt Digital Certificates or Keys
      – Used to validate/certify all key Host software and data files
      – Used to validate, encrypt & digitally sign all protected files
3. Legal "Chain of Certification" can be tested & proven
   1. Certification for system software, data files and downloads
   2. Every item Digitally Signed, encrypted, & run time verified

How TUC could protect software Programming Files on each power-up

On System Power Up :

TUC

1. TUC Hashs contents of 'Verification File' on Drive
2. TUC reads "encrypted Hash value" from file
3. TUC decrypts "encrypted Hash value"
4. TUC compares Hash values and verifies or rejects File Integrity
   If Rejected, report error & STOP SYSTEM
5. Start 'Verification File' program
6. For each program in 'Verification File' list (all Vendor files):
   - Compute Hash on file
   - Compare computed Hash to Hash stored in 'Verification File' table
7. If all Vendor files verified –> start Vendor software
8. If any file corrupted -> report error & STOP SYSTEM

TUC certifies Verification File – then file certifies rest of software. Chain of certification legally traceable to TUC's Digital Certificate

How TUC could protect Data File Updates

On Running "Update Data Files" Routine:

TUC

1. TUC's Update Data File program invoked by file update software
2. TUC verifies Digital Certificate signature from file
3. TUC reads Digital Certificate signed "encrypted Hash value" from file
4. TUC decrypts "encrypted Hash value"
5. TUC compares Hash values and verifies or rejects File Integrity
   If Rejected, report error & STOP SYSTEM
6. If Hash & Digital Certificate verified, start 'Verify Data File' program
7. For each program in 'Verify Data File' internal list:
   - Compute Hash on file
   - Compare computed Hash to Hash stored in 'Data File' internal table
8. If all Vendor files verified – Update new Verify Program HASH value and allow data file Update to disk
9. If any file corrupted - report error & STOP SYSTEM

TUC certifies Data Update files – then files loaded to disk.
Chain of certification legally traceable to TUC's Digital Certificate

How TUC could protect software Updates

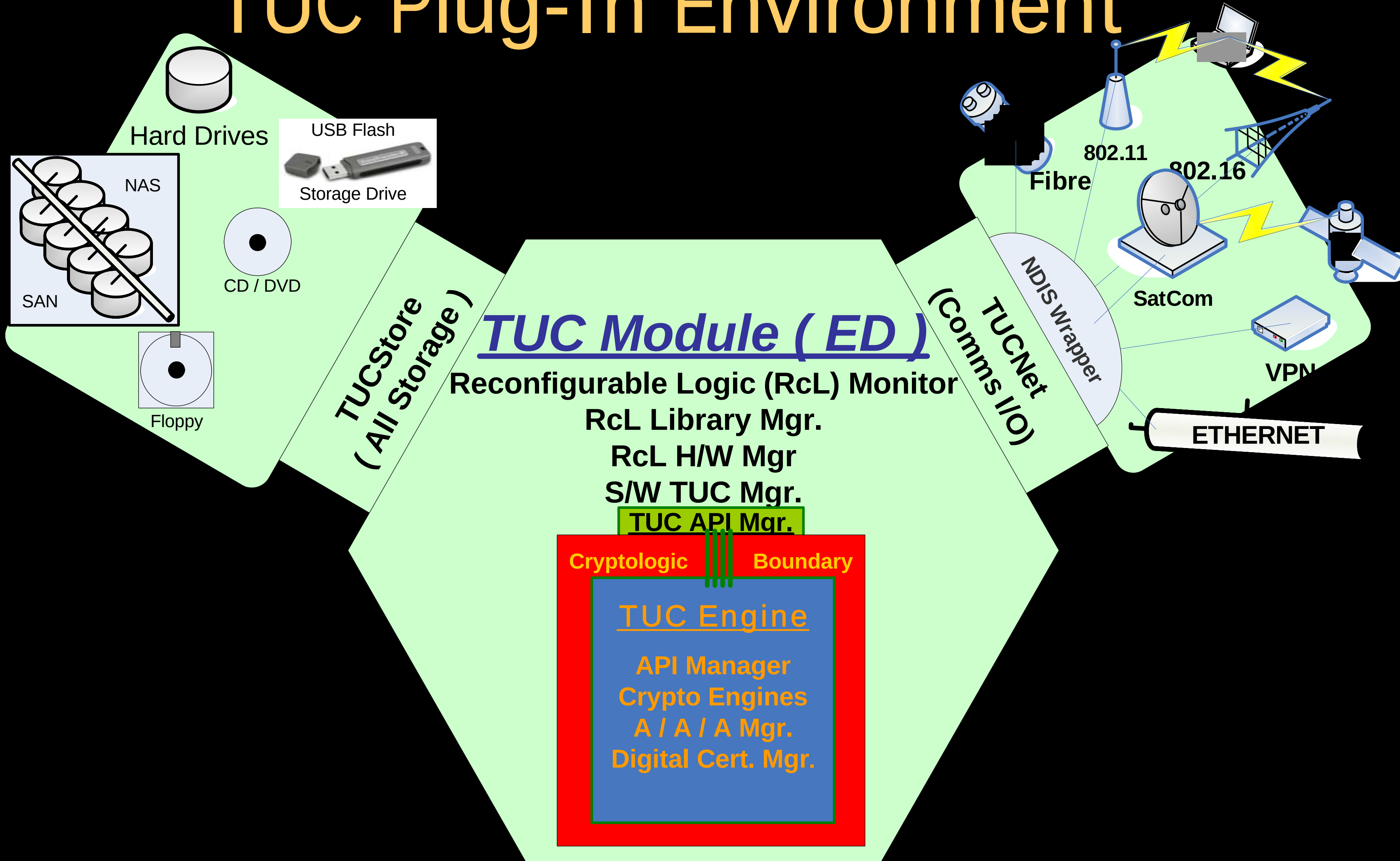
On Running "Update Software" Routine:

TUC

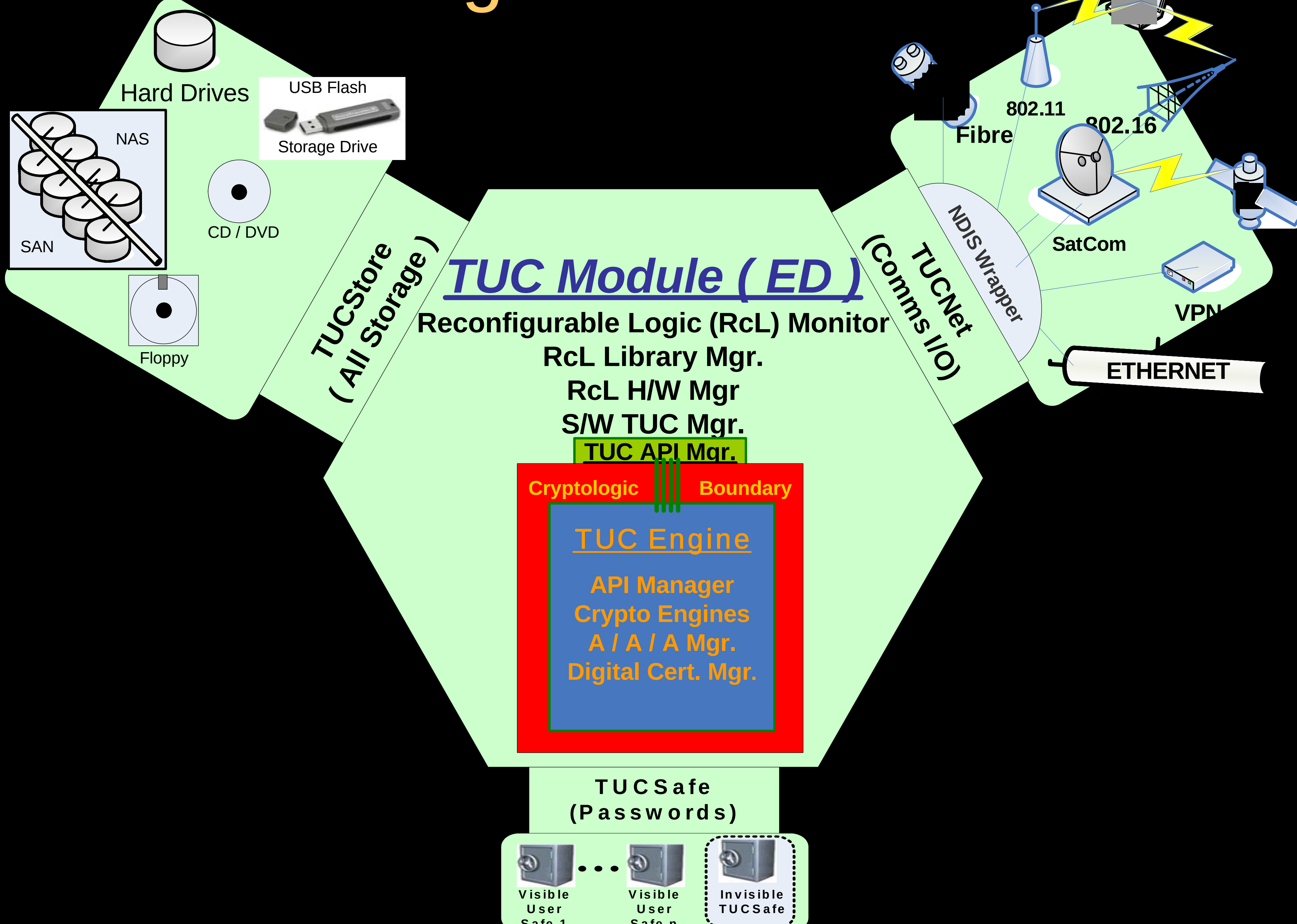
1. TUC's Update Software program invoked by install software
2. TUC verifies Digital Certificate signature from file
3. TUC reads Digital Certificate signed "encrypted Hash value" from file
4. TUC decrypts "encrypted Hash value"
5. TUC compares Hash values and verifies or rejects File Integrity.
   If Rejected, report error & STOP SYSTEM
6. If Hash & Digital Certificate verified, start 'Verify Update File' program
7. For each program in 'Verify Update File' internal list:
   - Compute Hash on file
   - Compare computed Hash to Hash stored in 'Verification File' internal table
8. If all Vendor files verified – Update new Verify Program HASH value and allow file Update to disk
9. If any file corrupted - report error & STOP SYSTEM

TUC certifies Update files – then files loaded to disk.
Chain of certification legally traceable to TUC's Digital Certificate

TUC Plug-In Environment



TUC Plug-In Environment

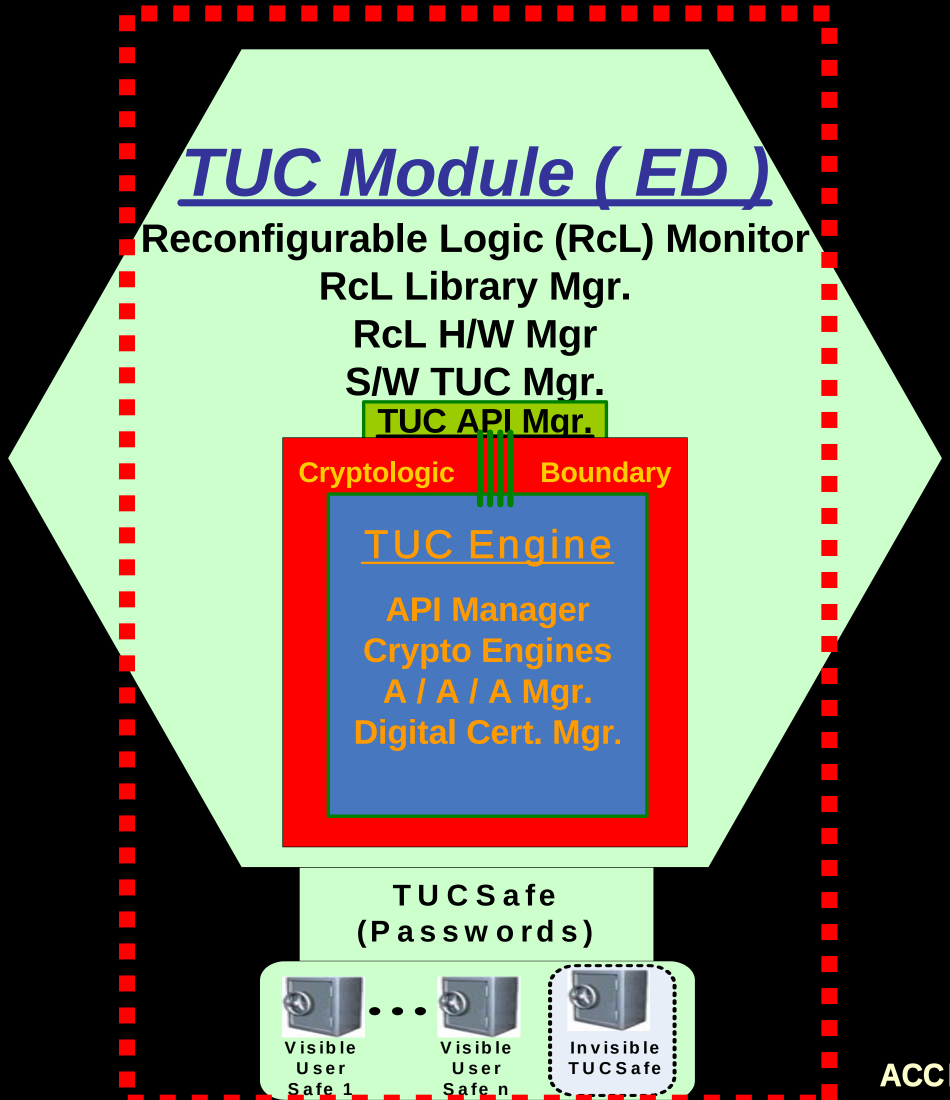


TUCSafe Password Protection Demo

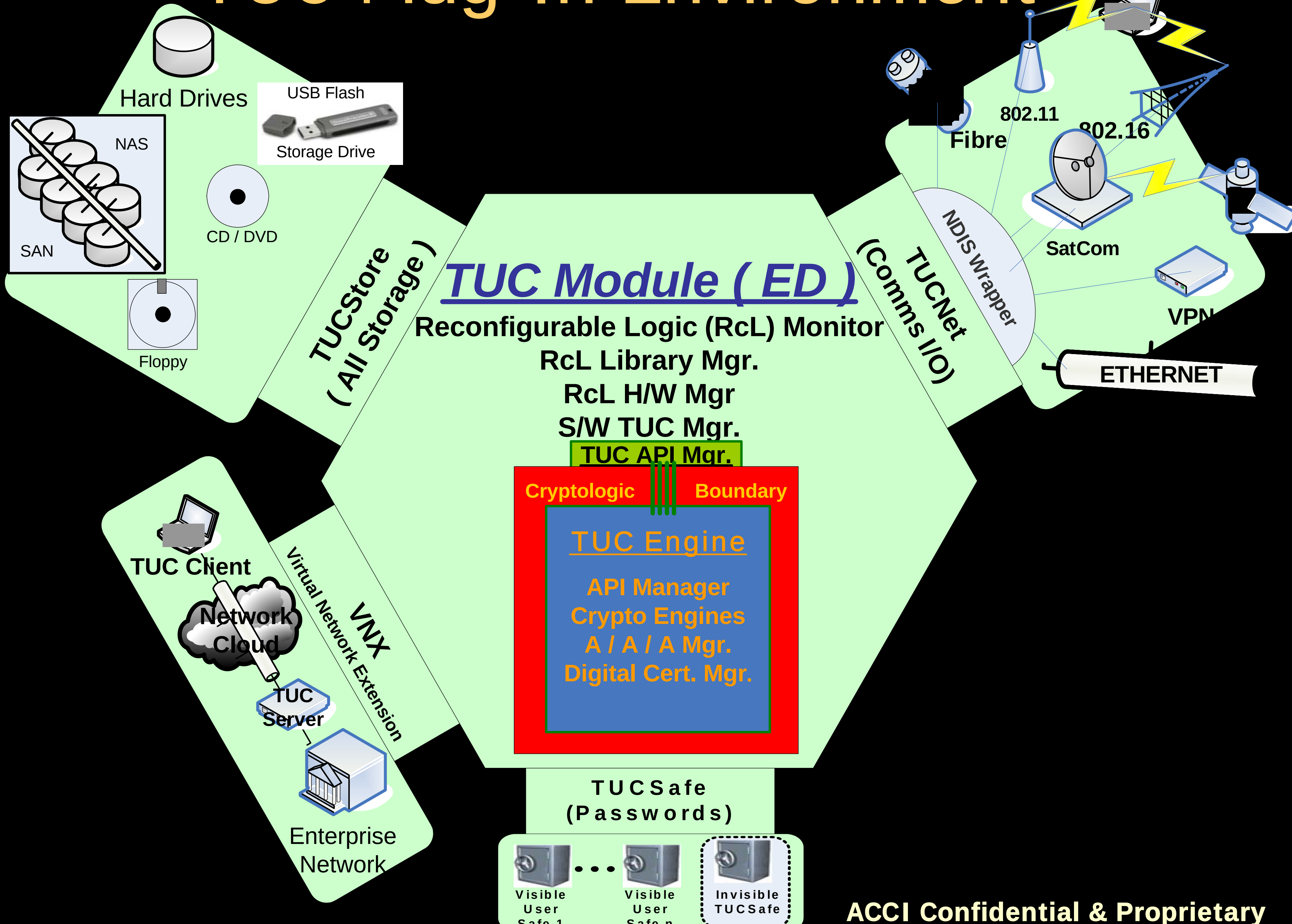
TUC Single Security Sign On

Any TUC application checks to see if TUCSafe installed :

If Yes, it uses it ; If No, it loads a minimal version & uses it

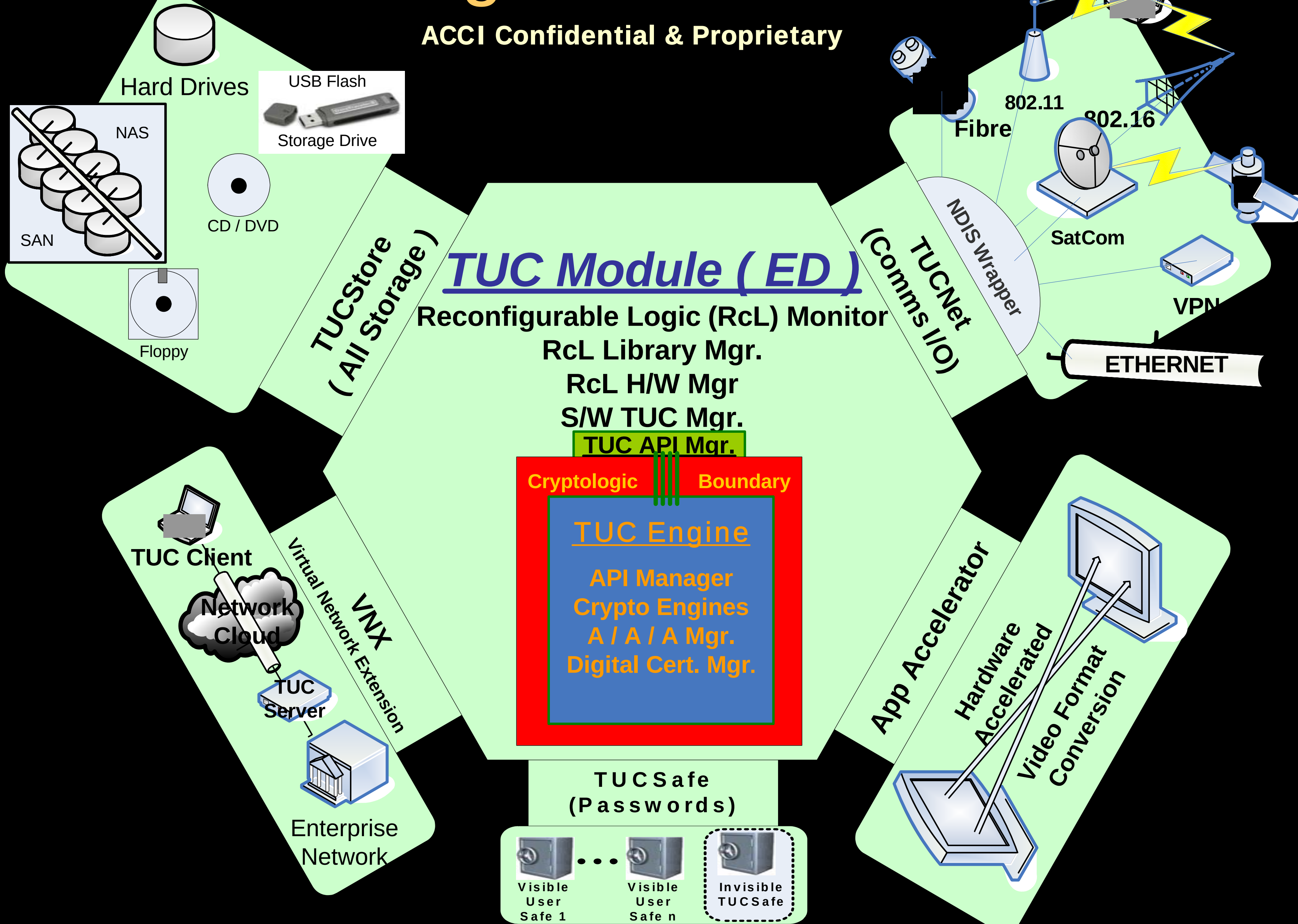


TUC Plug-In Environment



TUC Plug-In Environment

ACCI Confidential & Proprietary



TUC Benefits

- Can use CAC, or any other client Authentication scheme, as TUC Authentication/Authorization scheme
- Single Security Sign-on – all TUC apps use TUCSafe
- Single TUC Core module protects:
  - Wired and Wireless communications
  - All data storage devices – disks/ USB flash drives/ external drives/ burnable CDs & DVDs / tapes/ floppies
  - All passwords / keys / account information / secure URLs
- Secure wireless up to Top Secret (and SCI & above)
- Wide Area secure wireless for :
  - Multi mile coverage from single access point
  - Wireless VoIP phone service
  - Move personnel without IT/security burden
  - Secure laptop/remote log in from anywhere in the world
  - Disaster recovery/rapid mobility for entire command

Other TUC Advantages

- Future Proofing – can be as flexible as the threat
  - New/Updated encryption protocols can be added on-the-fly
  - Protocols enabled/disabled/restricted individually (in case of protocol "break" or updated certification/decertification)
- "Trusted 3rd Party" processor to provide program/file integrity verification
- NSA Certifiable Red/Black separation on single FPGA
- Instant "Zeroability" of encryption libraries, no "reverse-engineering" of security chips possible
- On-chip programming decryption, TUC configuration programming can be stored as encrypted files
- Complete interoperability of all TUC form factors – PC-Cards, USB, PCI boards, software implementation

Any Questions ?

Contact information:

Jonathan W. Ellis

Advanced Communications Concepts, Inc.

8834 N. Capital of Texas Hwy, Suite 212

Austin, Texas 78750-6396

(512) 275-6238 – office phone

(512) 275-6075 – office fax

Jonathan_ellis@advcommcon.com

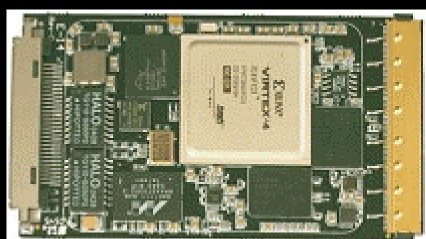
www.advcommcon.com

Back Up Slides

One Basic Circuit Design

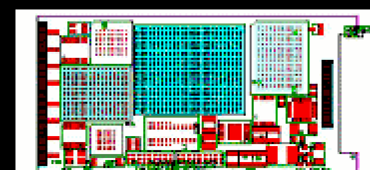
Multiple Busses Supported; Common Software & Tools

PC Card

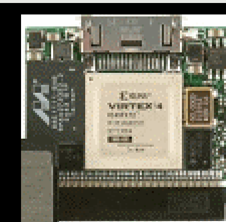


USB

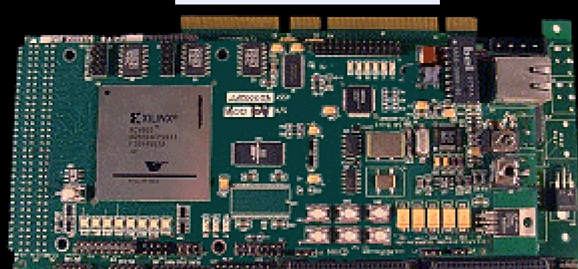
PCI Express

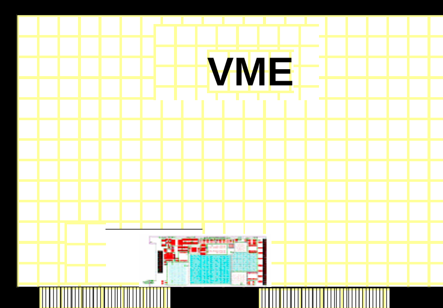


Compact Flash

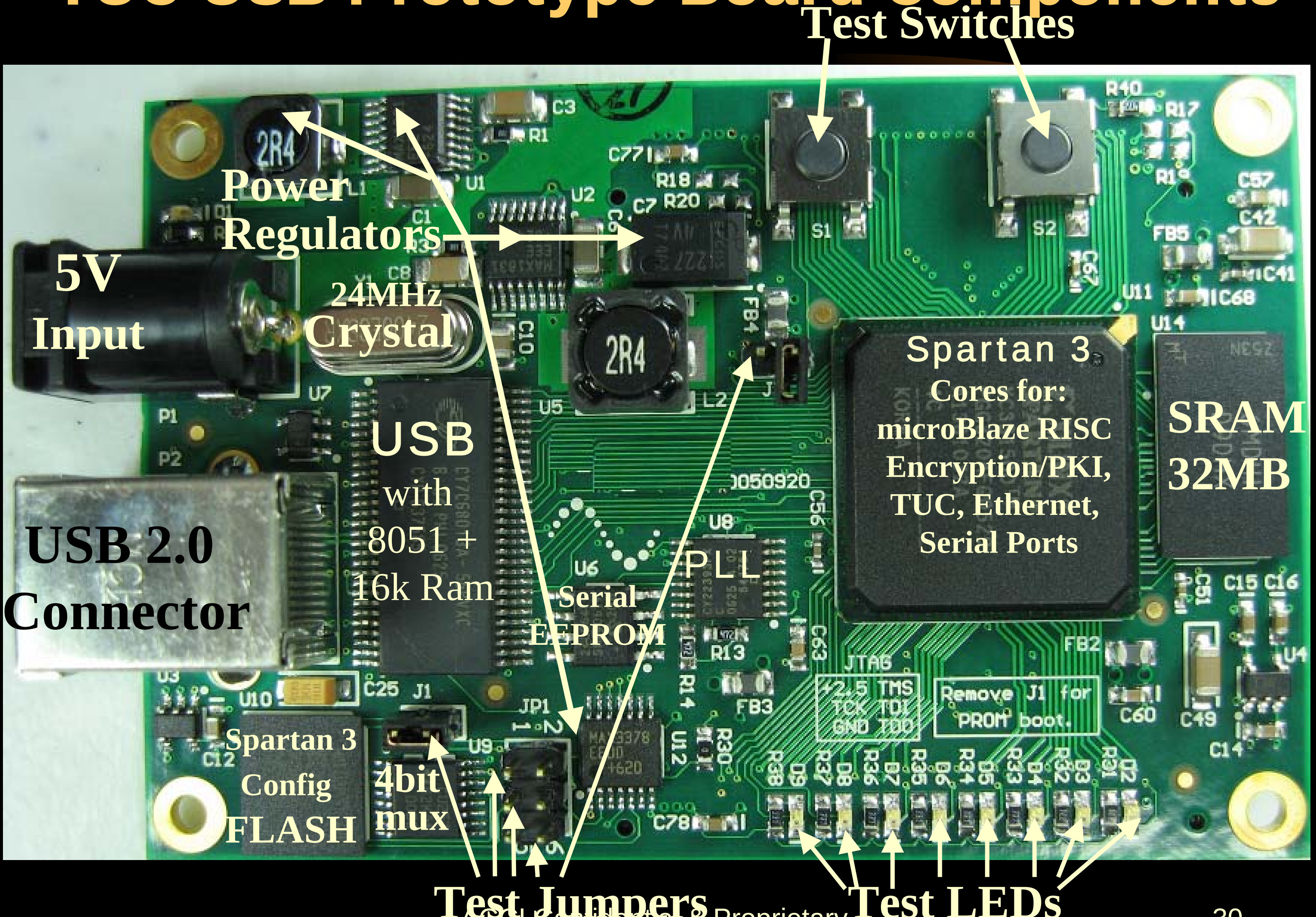


PCI Card

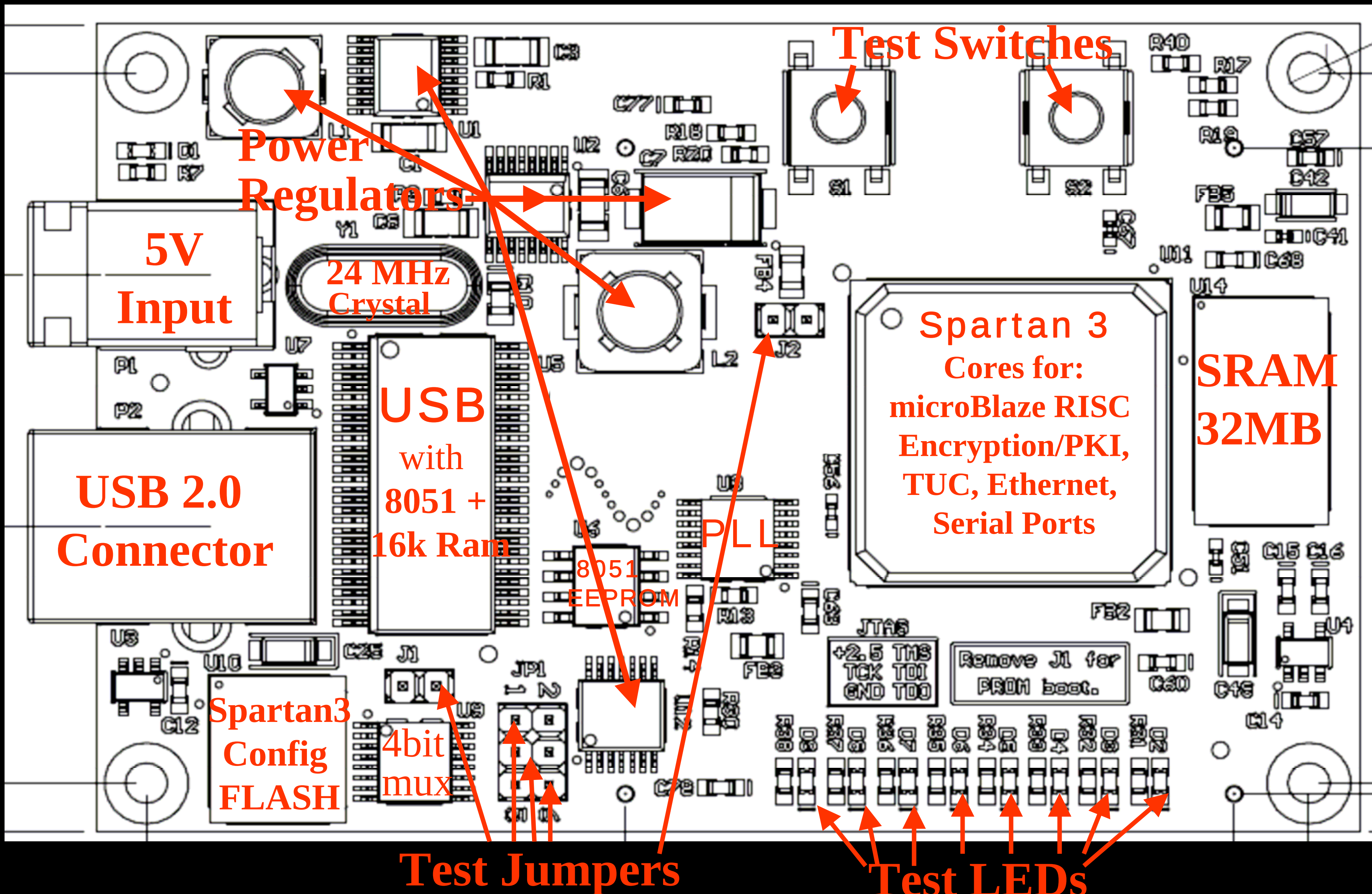




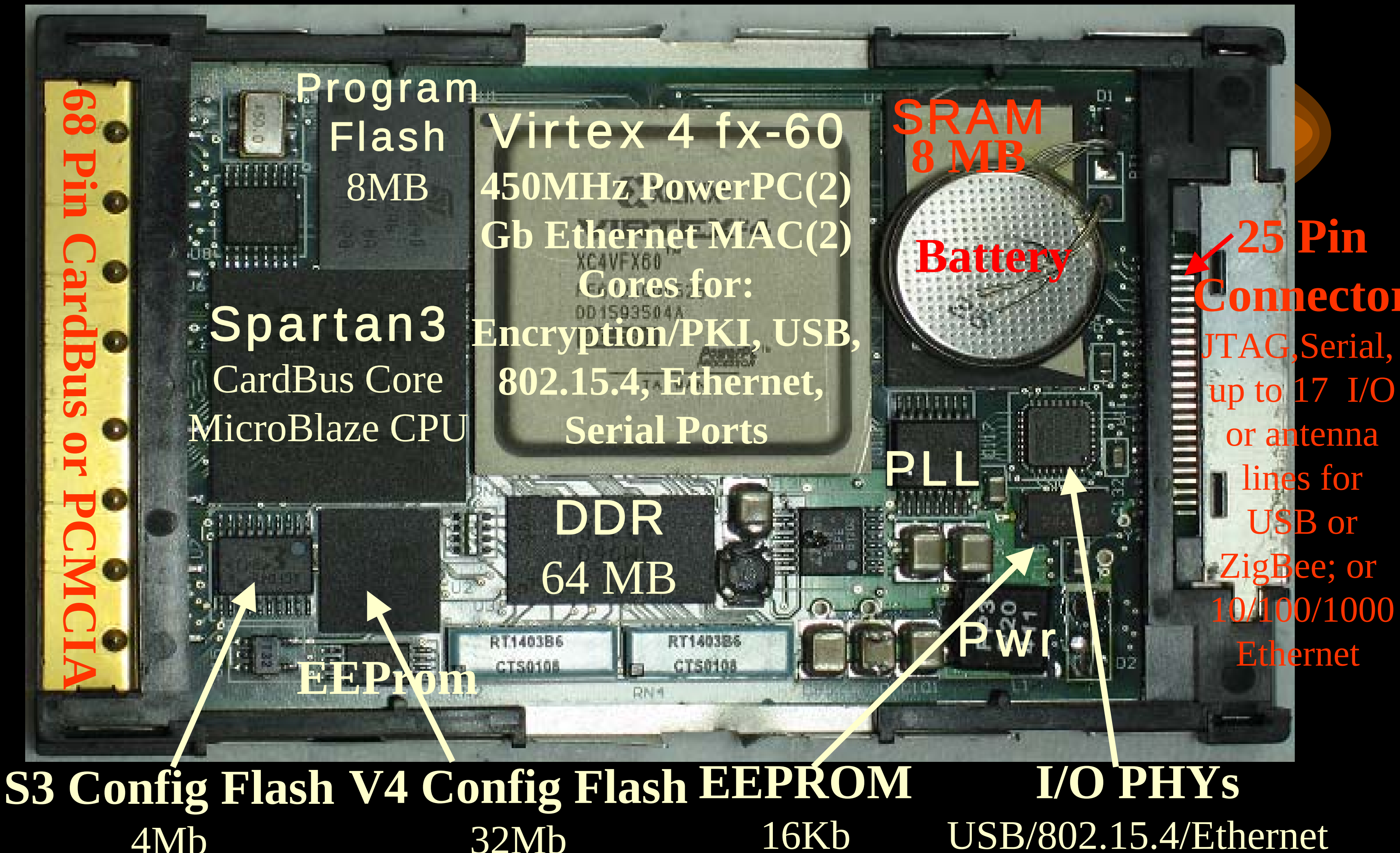
TUC USB Prototype Board Components



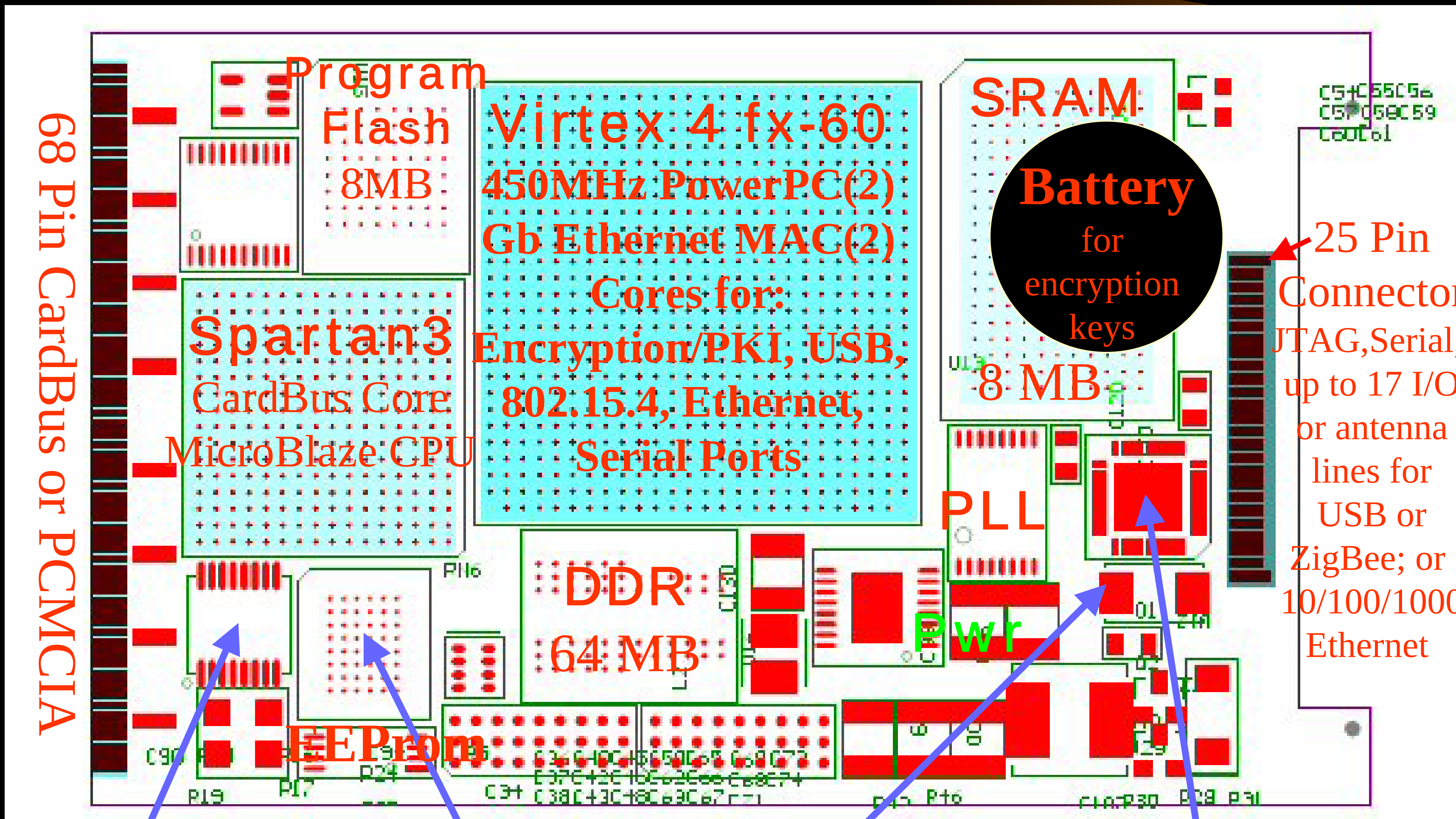
TUC USB Prototype Board Components



TUC PC Card Components



TUC PC Card Components



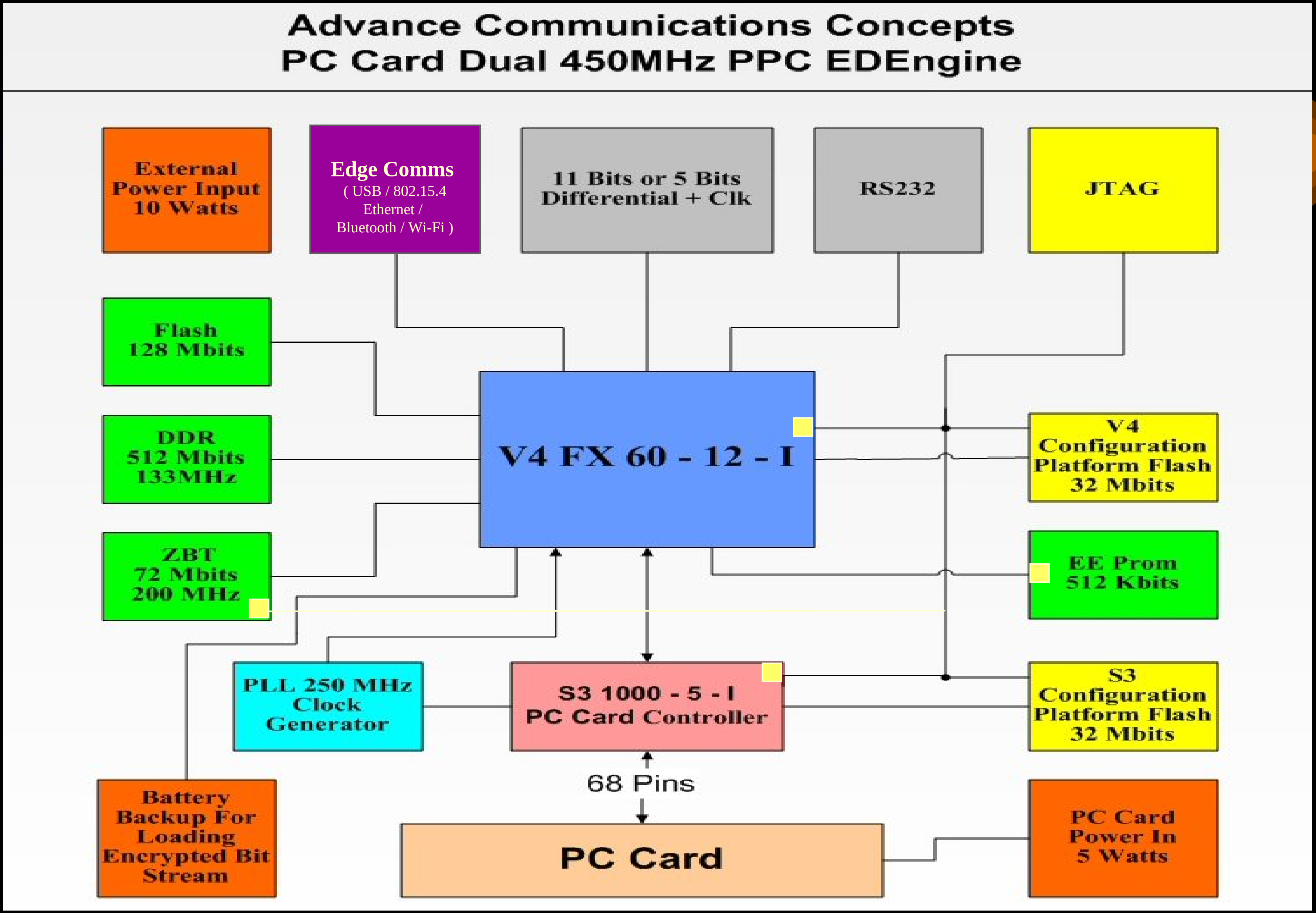
S3 Config Flash 4Mb

V4 Config Flash 32Mb

EEPROM 16Kb

I/O PHYs USB/802.15.4/Ethernet

TUC PCcard H/W Block Diagram



TUC vs. Standard Attack Schemes

- **Cipher Text only attack**
  - Computational Complexity of hopping sequences/per packet rekeying
- **Chosen Plaintext (w/compromised equipment)**
  - Varying text treatments with changing protocols/compression/encryption
    - **Send same message a million times – get a million different outputs**
- **Correlation/linear/differential attacks**
  - Varying text treatments with changing protocols/compression/encryption
- **Man-in-the-Middle attack w/compromised equipment**
  - Session Initialization Digital Signature/nonce and Client location authentication
- **Micro Power/Radiation/Timing Analysis**
  - Multiple concurrent chip operations (protocol/compression/encrypt/etc)
- **Introduced fault cryptanalysis**
  - Single chip operation of encrypted logic-mask based algorithms
- **Captured Equipment protocol library monitoring**
  - Multiple redundant, padded, Encrypted library entries per protocol