

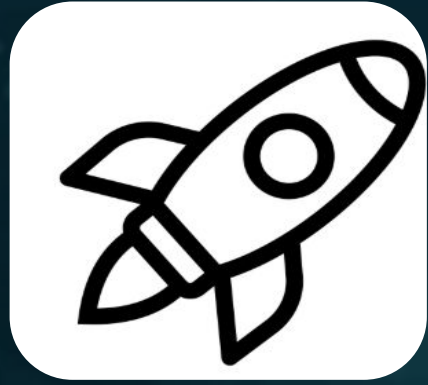
Securing Space

Threat Assessment and Remediation
Analysis for Nanosatellite On Board
Computers

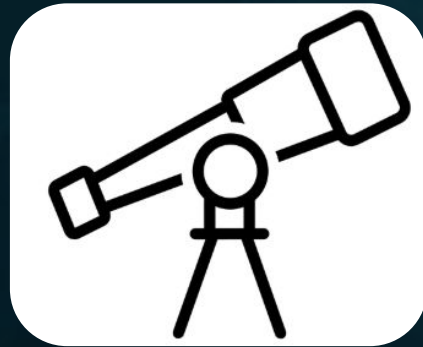
By: Kinryū Labs



Agenda



Introduction to cybersecurity in space: State of affairs in NanoSat space; introduction to nanosat OBCs



Threat modeling frameworks
- MITRE EMB3D and TARA



Combining threat modeling frameworks to deploy NanoSats securely by design.

Introduction

Introduction

Who are we?

- Kinryū Labs
- Website: kinryu.sh

Research

- Advanced threat intelligence and Cybersecurity in Space Systems



Members



Klara Kim



Donavan Cheah



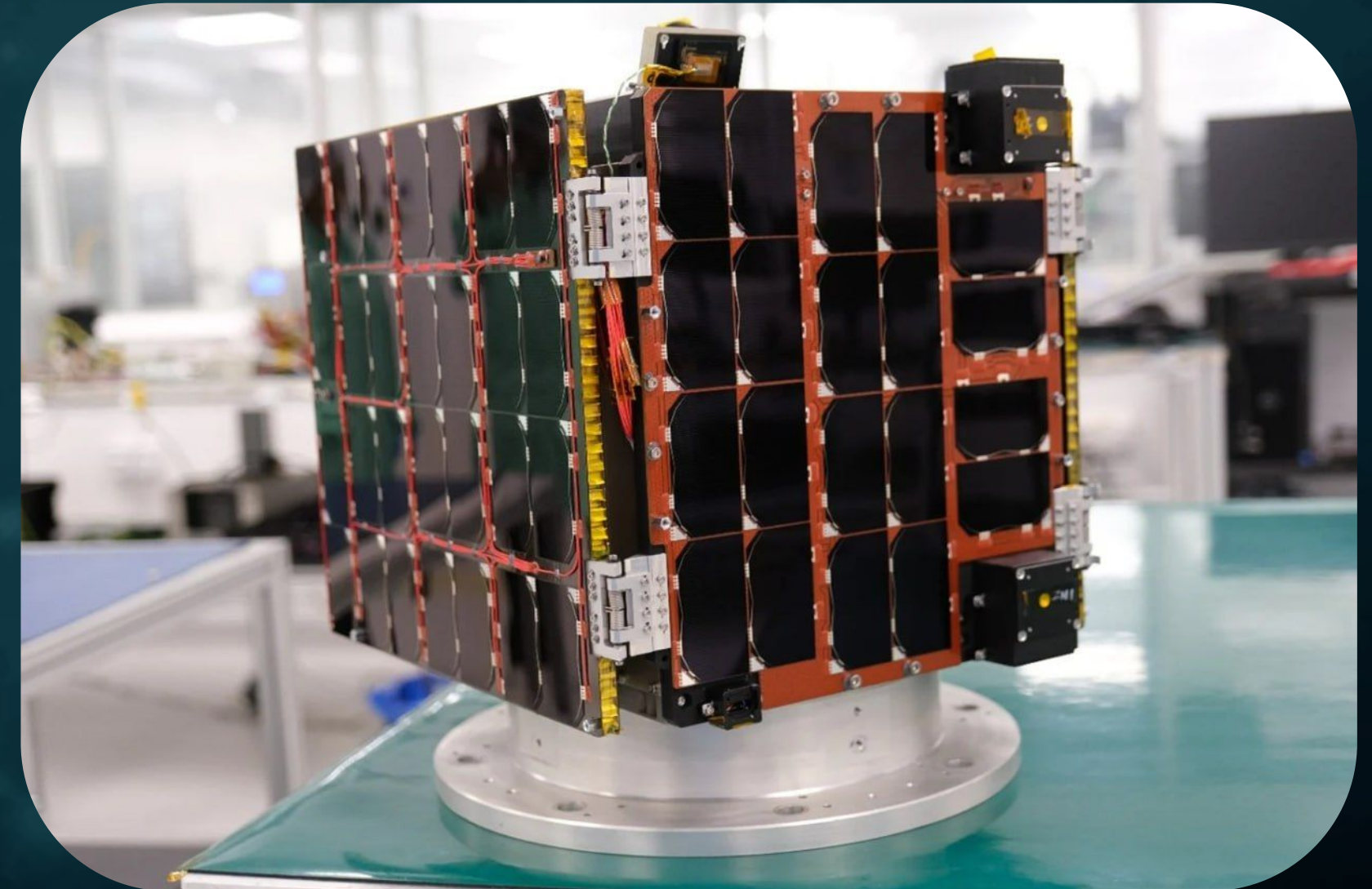
Davis Zheng



Yi-Ting Shen

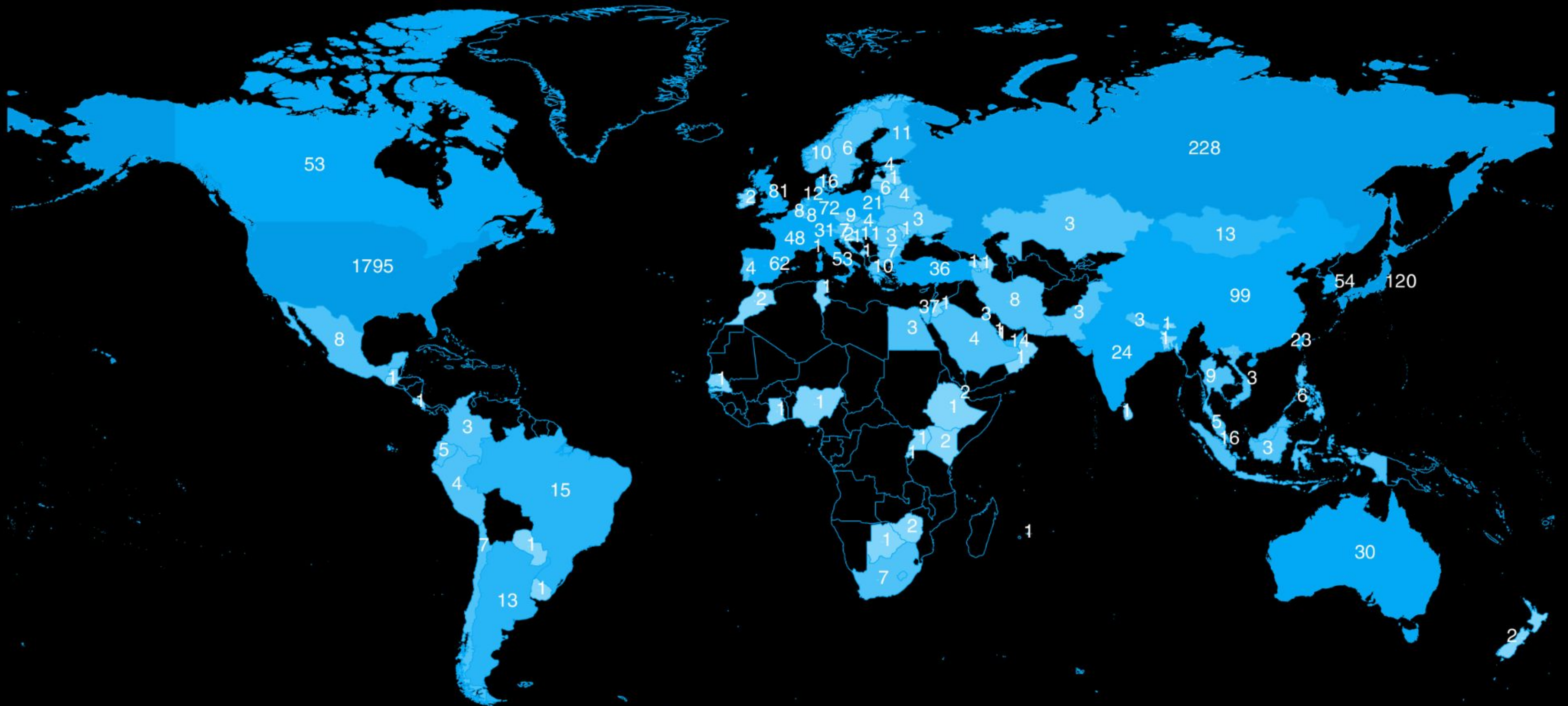
What are NanoSats?

- Miniature satellites
 - Typically developed by universities, governments and companies
- Details
 - 1-3U (1U = 10CM)
 - \$50,000 - \$150,000
 - SpaceX and ISRO has made it cheaper to launch satellites than ever before
 - Over 3200 NanoSats launched as of 2026



Velox AM by NTU SARC

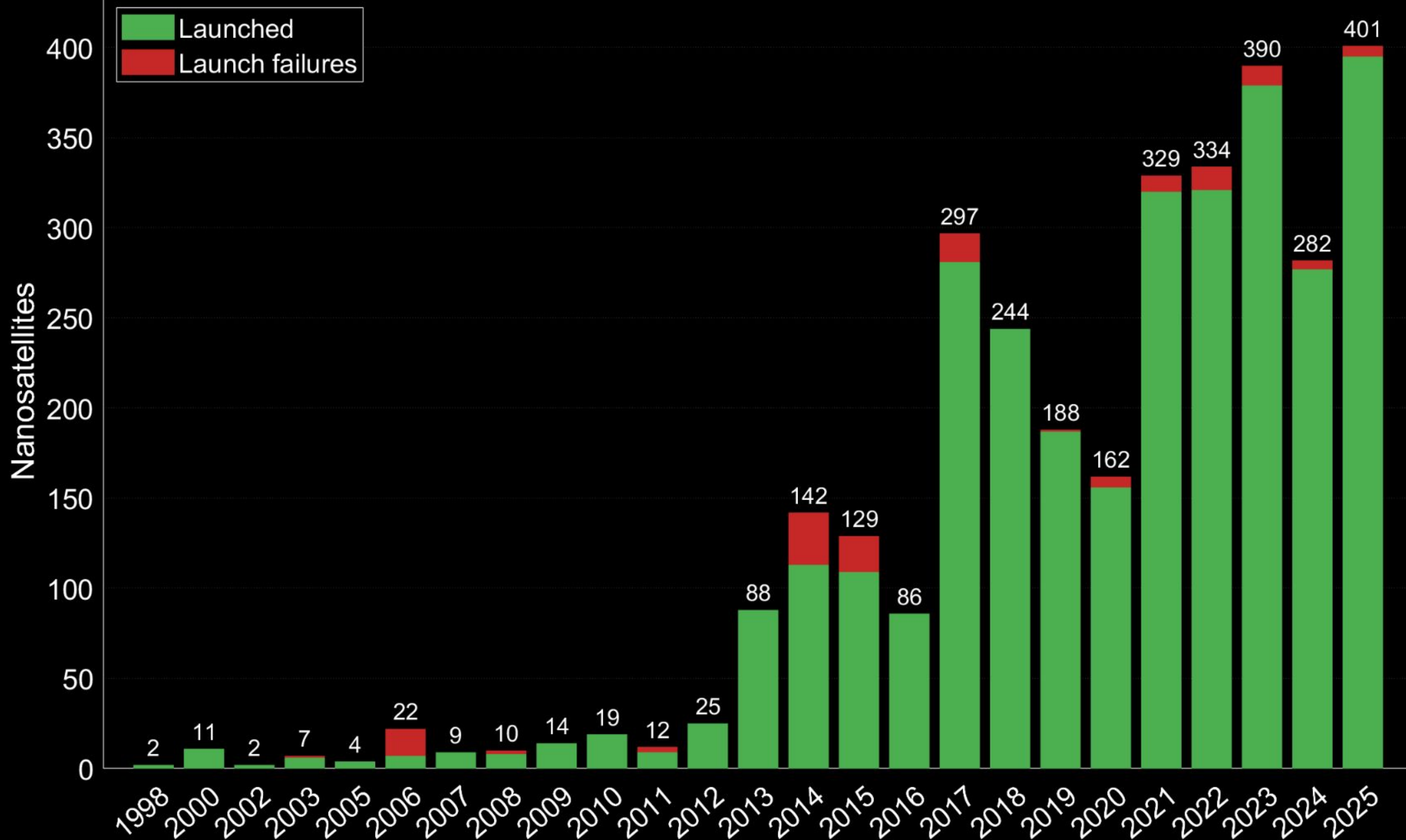
Launched nanosatellites



Nanosatellites launched

2025/12/31

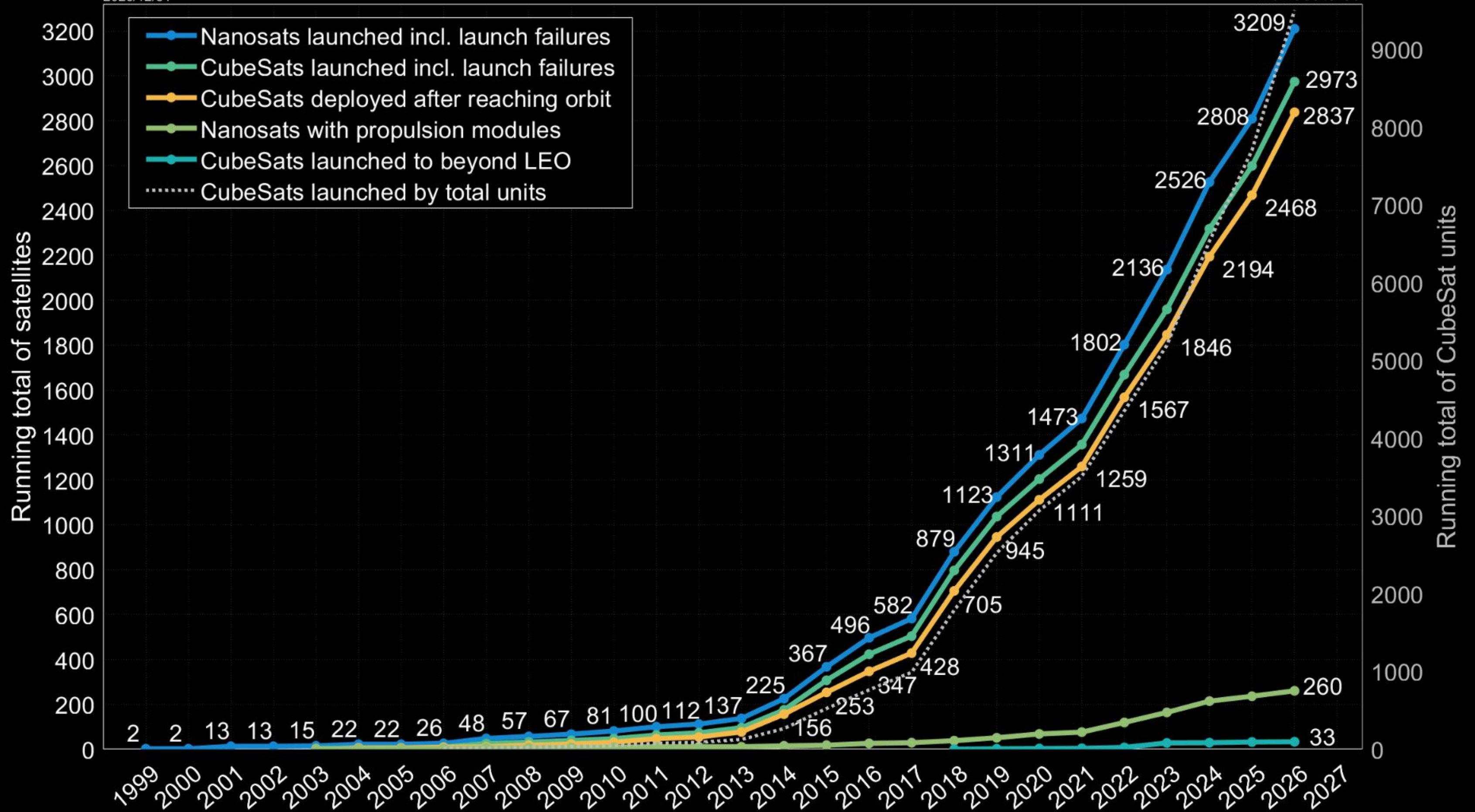
nanosats.eu



Total nanosatellites and CubeSats launched

2025/12/31

nanosats.eu



Cost-Benefit Profile

Mission Functionality > Security

- Benefits
 - Affordable to build
 - NanoSat (\$3,500++) vs Commercial Sat (\$10,000,000++)
 - Rapid prototyping
- Constraints
 - Tight Budget and timelines
 - Small size
 - Limited capabilities and processing power
 - Use of commercial off the shelf parts



Commercial Sat Teleos 2 by
STEngg

Rising Cyber Risks

- Nanosats often lack proper cybersecurity design
- Satellites are dual-use
- Tight schedules → no formal security testing
- Many engineers omit cybersecurity due to constraints and negligence
- As use cases and no. of satellites increase (e.g., communication, Earth observation), attack surfaces grows

```
if(bytes_to_read > 256) {
    send_nack(cmd);
    return;
}
if(!valid_read_region(read_address, bytes_to_read)) {
    send_nack(cmd);
    return;
}
// TODO: AN3154 says nack if ReadOutProtection is enabled and I think this
// is something in the option bytes? But we don't use it and I'm too lazy to
// check right now.
send_ack(cmd);

while(bytes_to_read > 0) {
    CANTxFrame reply = {
        .SID = cmd->SID,
        .DLC = MIN(bytes_to_read, 8),
    };
    memcpy(reply.data8, read_address, reply.DLC);
    bytes_to_read -= reply.DLC;
    read_address += reply.DLC;
}
```

<https://github.com/oresat/oresat-firmware>

Satellite Cyberattacks: A 25-Year History of Escalating Threats

1999

NASA Hack

15-year-old breached NASA systems; 21-day shutdown; \$1.7M data stolen



2019

NASA Mars Breach

Sensitive Mars mission data stolen; satellite dish network compromised



2022

Viasat KA-SAT

AcidRain wiper malware; tens of thousands modems disabled; Europe-wide outage



2024

Intelsat-33e

Sudden power loss; possible cyber-induced failure; satellite lost



2024

Jamming & Commands

Documented attempts at unauthorized commands; jamming attacks ongoing



Ground Infrastructure Focus

Most attacks target ground systems and management networks, not satellites in orbit

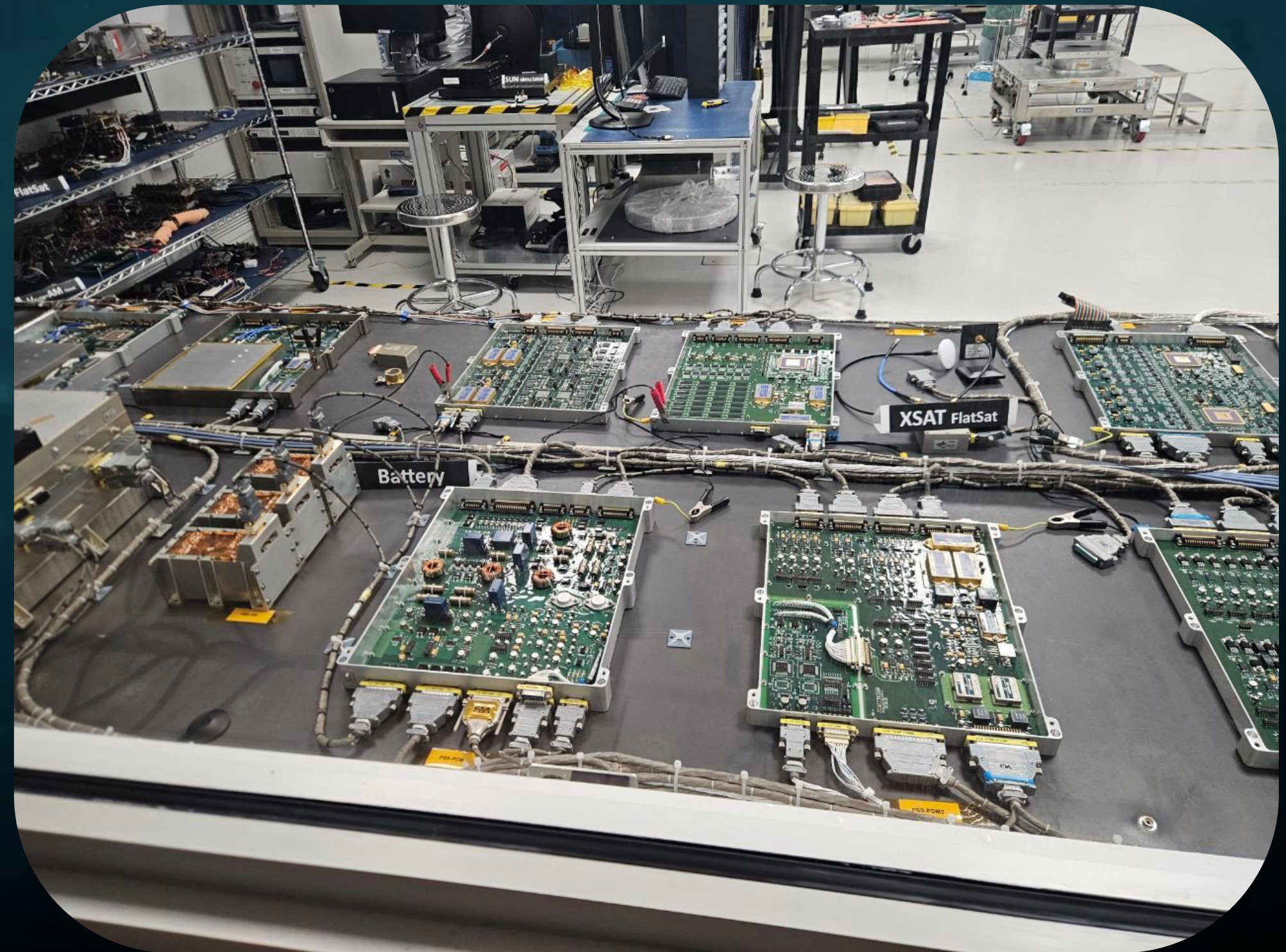
Escalating Sophistication

From data theft to destructive wiper malware to possible cyber-induced hardware failure

Stealthy Attribution

Cyber incidents may masquerade as hardware failures, complicating detection and response

Nanosat System Architecture and Flatsat



NTU XSAT Flatsat and Architecture

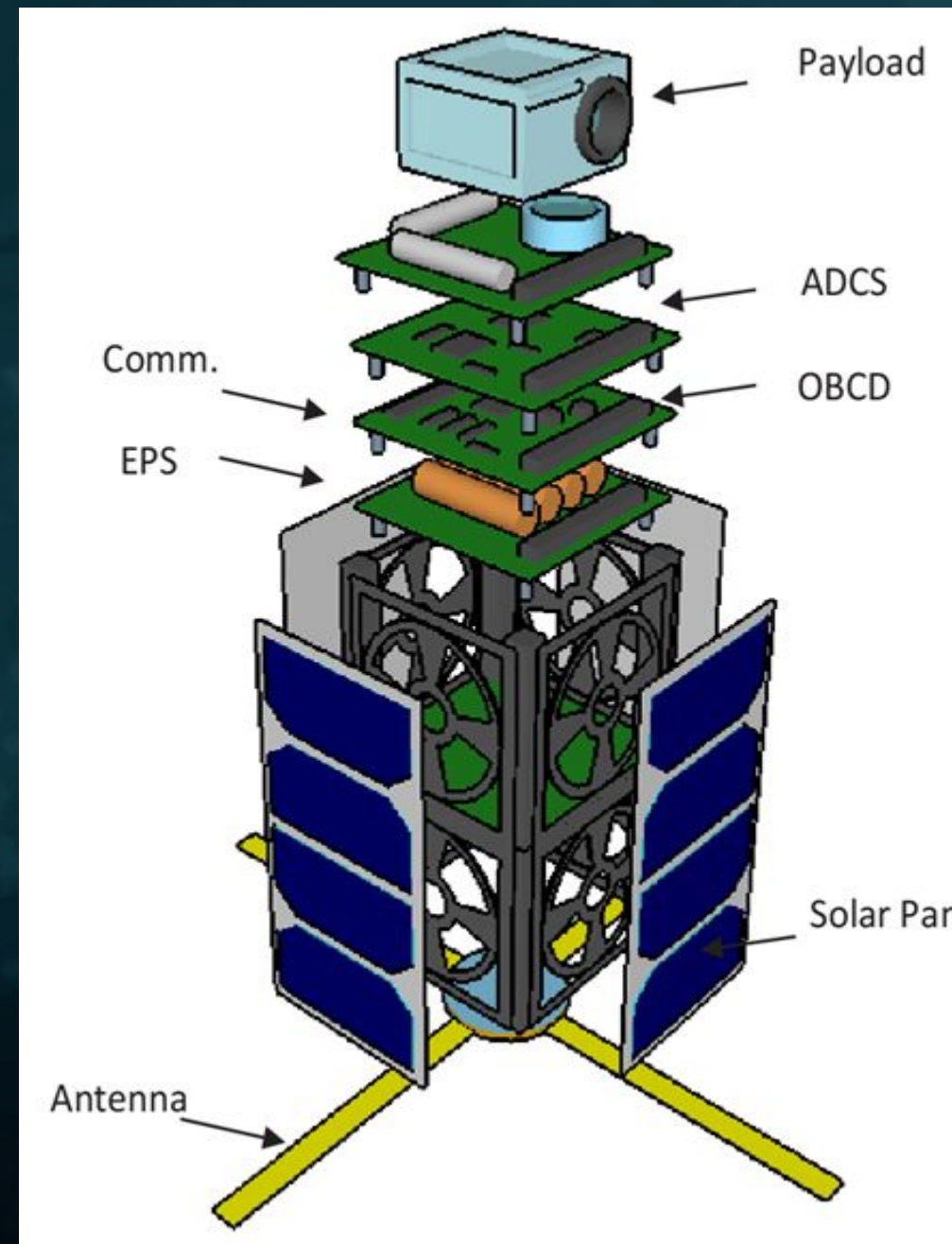
Core Subsystems of NanoSat and Security Implications

OBC (On-Board Computer)

- Central control and subsystem management
- **Impact**
 - a. **Full mission paralysis**
 - b. *Potential Weaponization of Satellite*

Comm (Communication Subsystem)

- Uplink/downlink data exchange with ground stations
- Impact
 - a. *Command spoofing or data exfiltration*



Typical Nanosat Architecture

Payload

- Mission-specific instrument or sensor package
- *Impact*
 - a. *loss or corruption of mission data*

ADCS (Attitude Determination and Control System)

- Satellite orientation and attitude control
- *Impact*
 - a. *Orbital collision risk or observation failure*

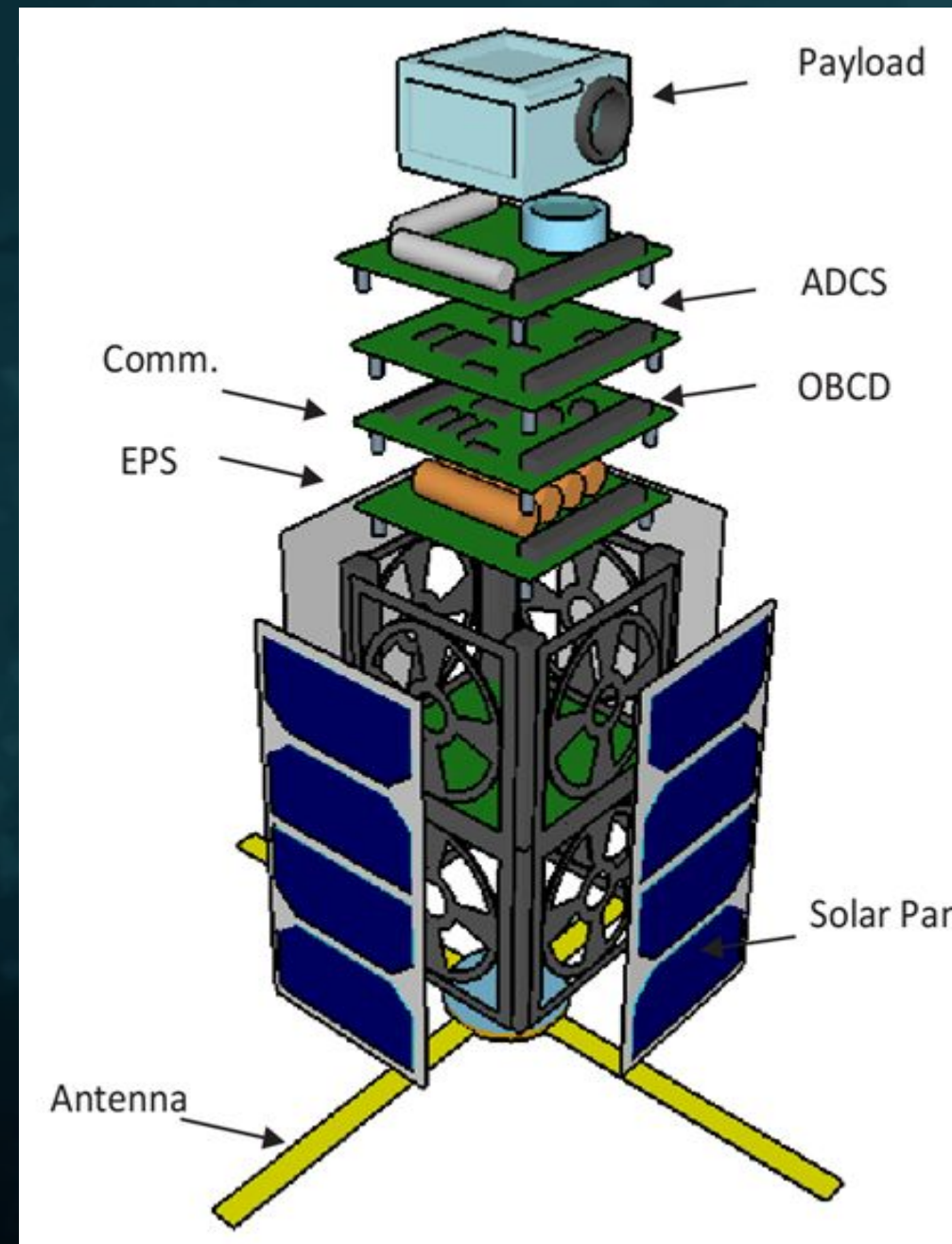
Core Subsystems of NanoSat and Security Implications

EPS (Electrical Power System)

- Solar array and battery power management
- Impact
 - a. Satellite shutdown

Solar Panels

- Power generation via sunlight
- Impact
 - a. Power degradation or denial of service



Typical Nanosat Architecture

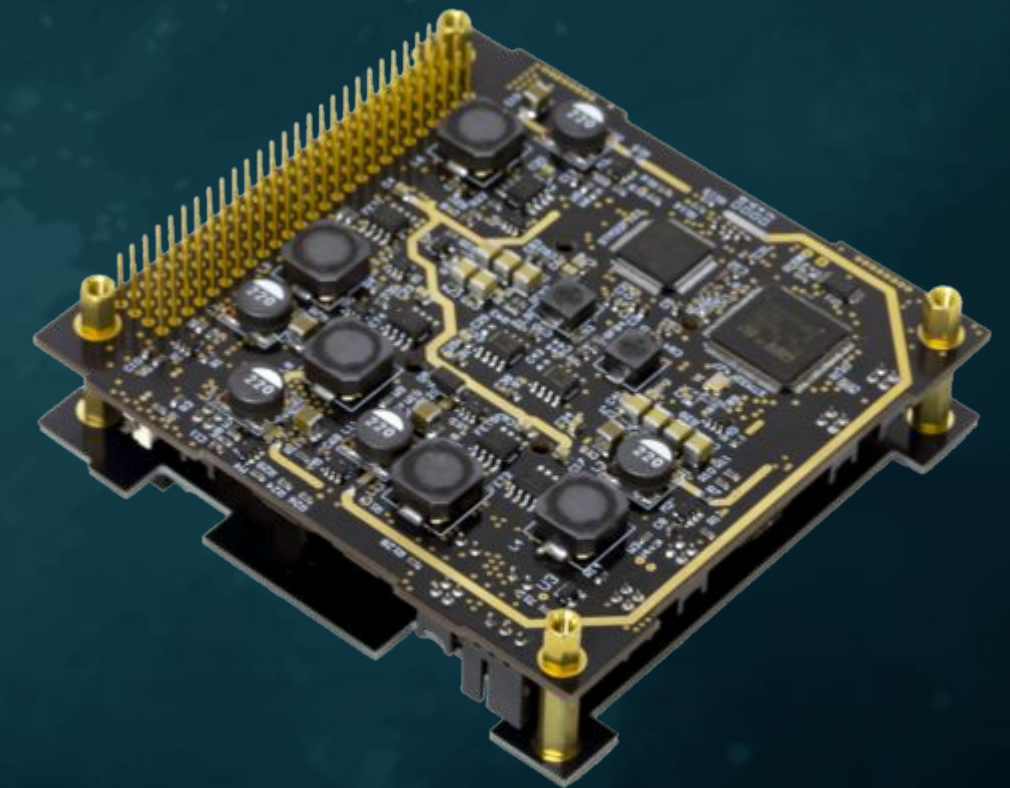
Antenna

- Transmission and reception of RF signals
- Impact
 - a. Jamming or loss of communication link

And other components like TT&C depending on the mission

Nanosat OBC Basics

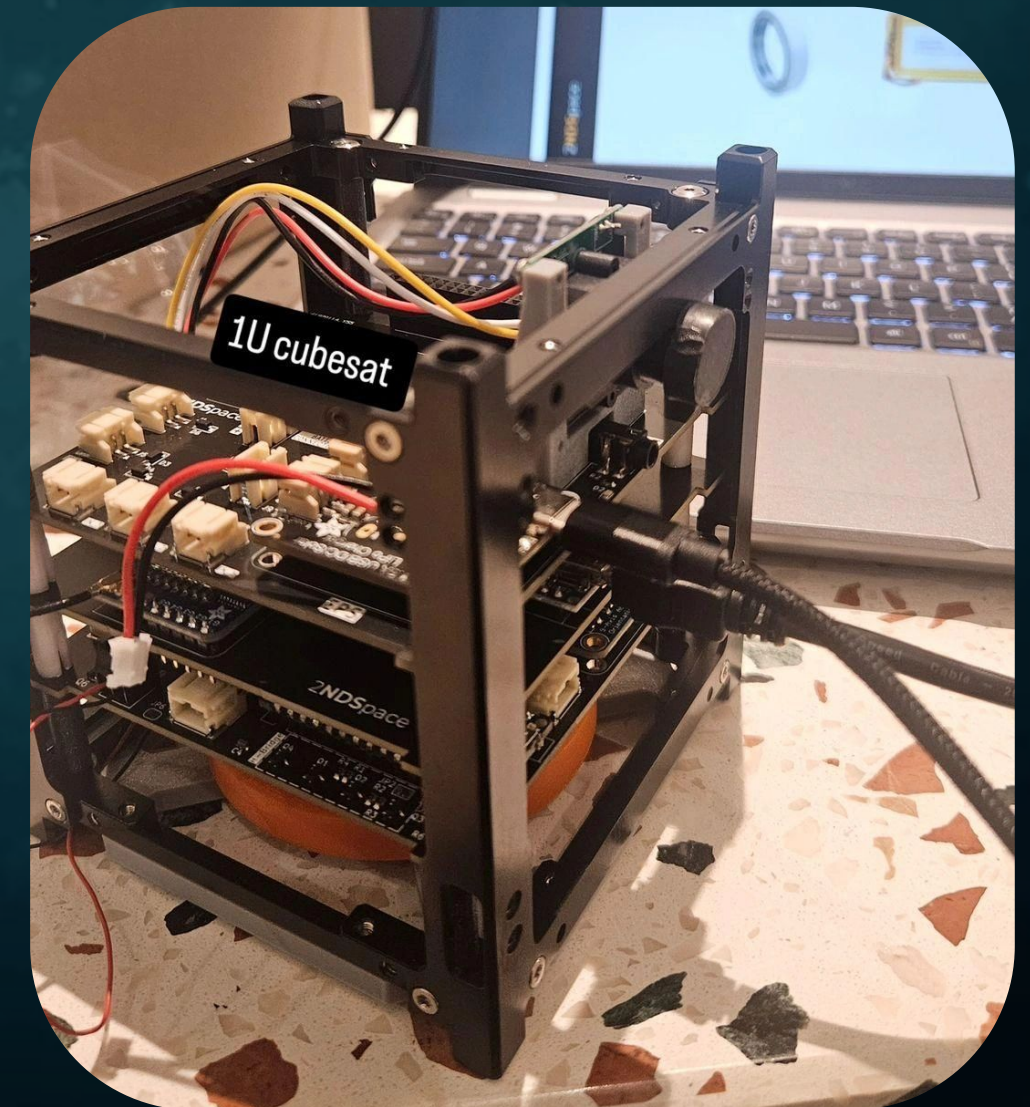
- OBCs are typically low-powered microcontrollers (e.g. ARM Cortex-M or similar)
 - Often runs simple real-time operating system like FreeRTOS¹.
- Manages critical functions:
 - processing telecommands
 - collecting sensor data
 - controlling subsystems
 - communicating with the ground
- Minimal memory and no hardware memory management unit (MMU) in many cases – meaning **no process isolation or user/kernel separation**.



Aphelion Cubesat Bus Module
and Integrated OBC

Nanosat OBC Basics

- NanoSat Teams prioritize
 - Functionality → Communication → Power → Security (last)
- Reasons
 - Mission constraints
 - i. Not enough processing power
 - ii. Mission critical processing takes precedence
 - iii. No budget to hire a cyber personnel
 - No or little impact
 - i. Not many systems to attack and the end result is a DoS
 - ii. There are no peripherals to hack
 - Lack of cybersecurity training or understanding
 - i. Aerospace engineers lack cybersecurity understanding
 - ii. Small Team Sizes



CubeSat frame from 2ndSpace

What is Threat Modeling?

A systematic approach to identifying, analyzing, and prioritizing system threats.

Threat Modeling Manifesto:

1

What are we working on?

Scope of the threat model. In our case, we are working on the **full** stack of nanosat systems.

2

What can go wrong?

SATCR4CK has decided to use MITRE EMB3D and TARA to enumerate threats.

3

What are we going to do about it?

Nanosats present limitations requiring lightweight and risk-prioritised mitigations.

4

Did we Do a Good Enough Job?

Conclude if threat model adequately addresses the need for Nanosat security.

Threat Modeling in NanoSat Systems

Limited Resources

Nanosats have constraints in power, computation, and memory
→ can't apply every security control.

Need to **prioritize the most critical threats** systematically.

Complex Attack Surface

Hardware, firmware, and networks all create multiple attack paths.

Threat modeling helps to provide a system-level view of attack paths.

Patch Limitations

Ground systems can be updated, but CubeSats in orbit have limited patchability due to available uplink/downlink time.

Standardized Communication Tool

Acts as a **shared framework** between developers, security teams, and operators for consistent understanding through data flow diagrams (data store, process, data flow, external entity)

Threat Modeling Frameworks

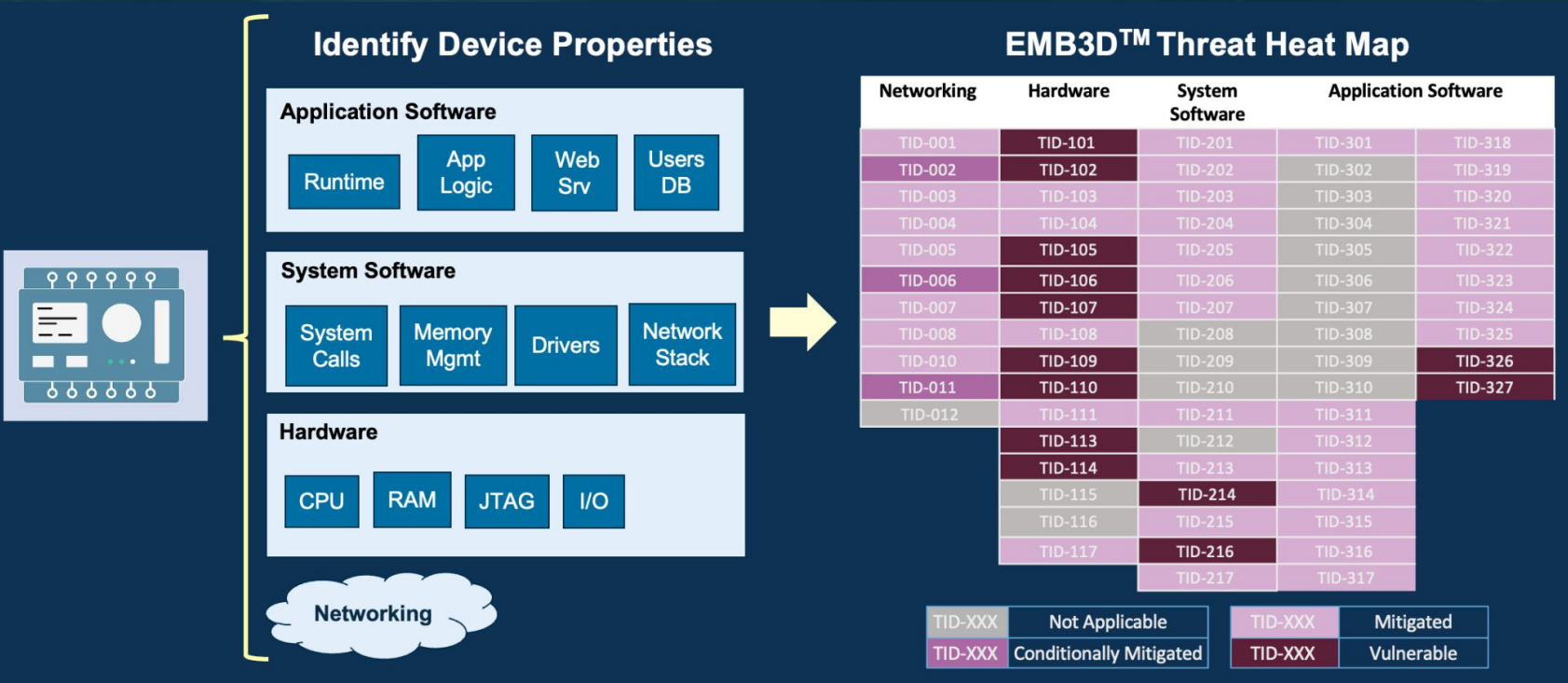
Model	Main Target	Core Concept / Method	Strengths	Limitations
STRIDE (Microsoft)	Software systems	6 threat categories: Spoofing, Tampering, Repudiation, Information Disclosure, DoS, Elevation of Privilege	Intuitive, simple, suitable for software threats	May oversimplify; not ideal for complex system threats
LINDDUN	Privacy-sensitive systems	7 Privacy threat categories: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure, Unawareness, Non-compliance	Strong privacy protection focus	Limited for general threat modeling (non-privacy systems)
PASTA (Process for Attack Simulation & Threat Analysis)	Business & technical systems	Risk-based, 7-stage threat modeling framework	Quantitative risk analysis possible	Complex and time-consuming to execute
OCTAVE (Carnegie Mellon)	Organizational information systems	Asset-driven risk assessment	Good for organizational security posture	Limited technical depth
VAST (Visual, Agile & Simple Threat)	DevOps / Agile environments	CI/CD pipeline-integrated visual threat modeling	High speed, DevSecOps friendly	Limited standardization depth
MITRE EMB3D	Embedded systems / IoT	Property (PID) – Threat (TID) – Mitigation (MID)	Excellent for embedded and IoT security	Lower applicability to general IT systems
TARA (Threat Analysis & Risk Assessment, ISO/SAE 21434)	Automotive & shared CPS	Threat scenarios + risk level mapping	Focused on automotive and CPS domains	Requires risk level definition (baseline or reference needed)

➡ We selected **EMB3D + TARA** as they fit embedded CPS like satellites.

MITRE EMB3D: An Introduction

What is EMB3D?

Threat model for **embedded devices** developed by MITRE. (Sep. 2024)
Applicable across **IoT, robotics, satellites, and cyber-physical systems (CPS)**.



Features

Structured as **PID-TID-MID** relationships linking.

- **Properties (P) → Threats (T) → Mitigations (M).**

Based on **STIX 2.1** format, enabling **automated data analysis and mapping**.

Uses a heat map to enumerate threat exposure.

System Software:

- ☐ PID-21 - Device includes a bootloader
- ☒ PID-22 - Device includes a debugging capabilities
- ☒ PID-23 - Device includes OS/kernel
 - ☐ PID-231 - Device includes an operating system that uses drivers/modules that can be loaded
 - ☐ PID-232 - Device includes separate users/processes with access to different OS data or functions
- ☐ PID-24 - Device includes virtualization and containers
- ☒ PID-25 - Device includes software/hardware root of trust
 - ☒ PID-251 - Root of Trust is physically accessible or is not immutable
 - ☐ PID-252 - Root of Trust is immutable
- ☐ PID-26 - Device lacks firmware/software update support
- ☐ PID-27 - Device includes support for firmware/software updates
- ☒ PID-28 - Device stores logs of system events and information

Applicable Threats:

[Download CSV](#)

- PID-22**
 - TID-224 - Excessive Access via Software Diagnostic Features
- PID-23**
 - TID-202 - Exploitable System Network Stack Component
 - TID-218 - Operating System Susceptible to Rootkit
- PID-251**
 - TID-214 - Secrets Extracted from Device Root of Trust
- PID-28**
 - TID-225 - Logs can be manipulated on the device
 - TID-226 - Device leaks security information in logs

MITRE EMB3D Process

Three-Step Process

- 1.  Identify Device Properties
 - a. Hardware/Software components and interfaces (PIDs)
- 1.  Map to Threats
 - b. Match each property to relevant Threat IDs (TIDs) from EMB3D's threat catalog
- 1.  Identify Mitigations
 - b. Apply Mitigation IDs (MIDs) organized by robustness tiers

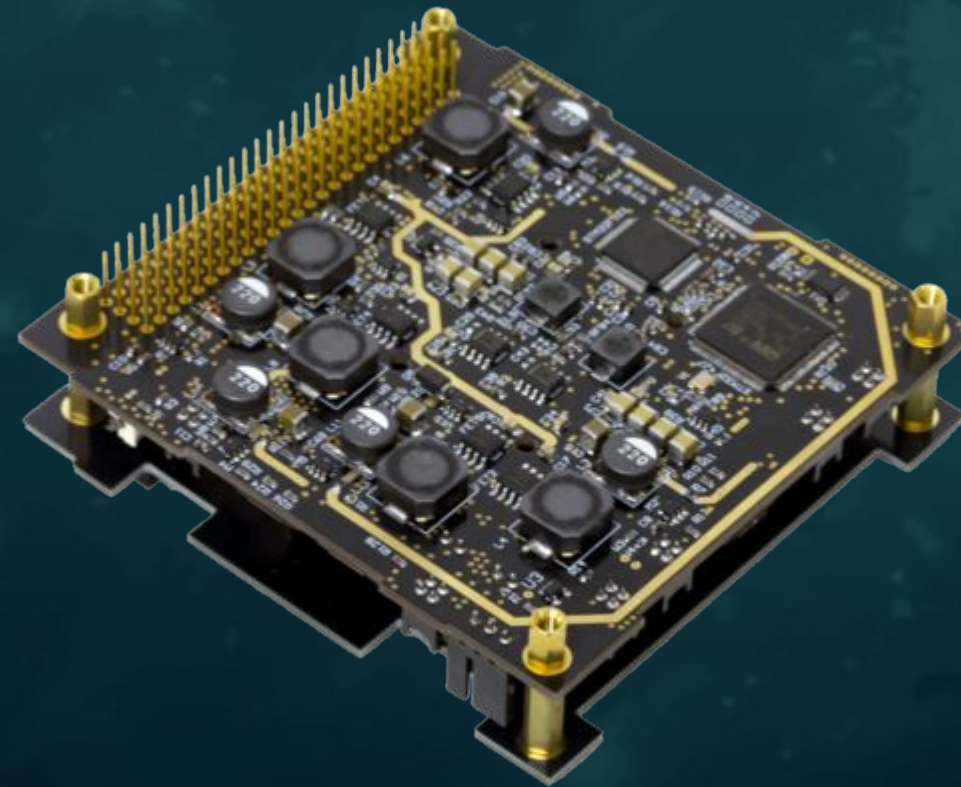
Hardware			
Device Properties		Threats	
PID-11	Device includes a microprocessor	TID-101	Power Consumption Analysis Side Channel
		TID-102	Electromagnetic Analysis Side Channel
		TID-103	Microarchitectural Side Channels
		TID-105	Hardware Fault Injection – Control Flow Modification
PID-12	Device includes Memory/Storage (external to CPU)		
PID-121	Device includes buses for external memory/storage	TID-106	Data Bus Interception
PID-122	Device includes discrete chips/devices that have access to the same physical memory	TID-107	Unauthorized Direct Memory Access (DMA) 

Why MITRE EMB3D for NanoSat Systems?

- NanoSat's OBC's: Classic embedded systems in orbit

- Unique constraints
 - Real-time tasks
 - Limited compute
 - Exposed interfaces

- EMB3D helps map
 - real design features
 - potential threats
 - mitigations



Aphelion NanoSat OBC VS an embedded system

- Enables “security by design” engineering from early mission stages

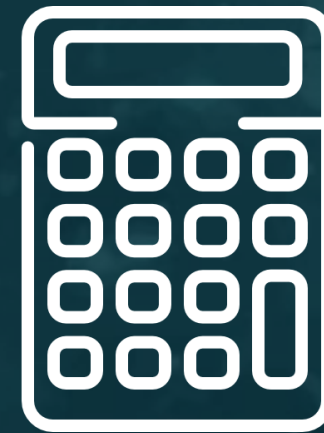
What is TARA?

Threat Assessment and Remediation Analysis

- Developed by MITRE and used widely in automotive and embedded security
- Provides a structured method to prioritize and manage threats
- Converts qualitative threats into quantitative risk values
- Complements EMB3D by evaluating likelihood, impact, and mitigation effectiveness.

What is TARA?

4 Steps in the TARA Process



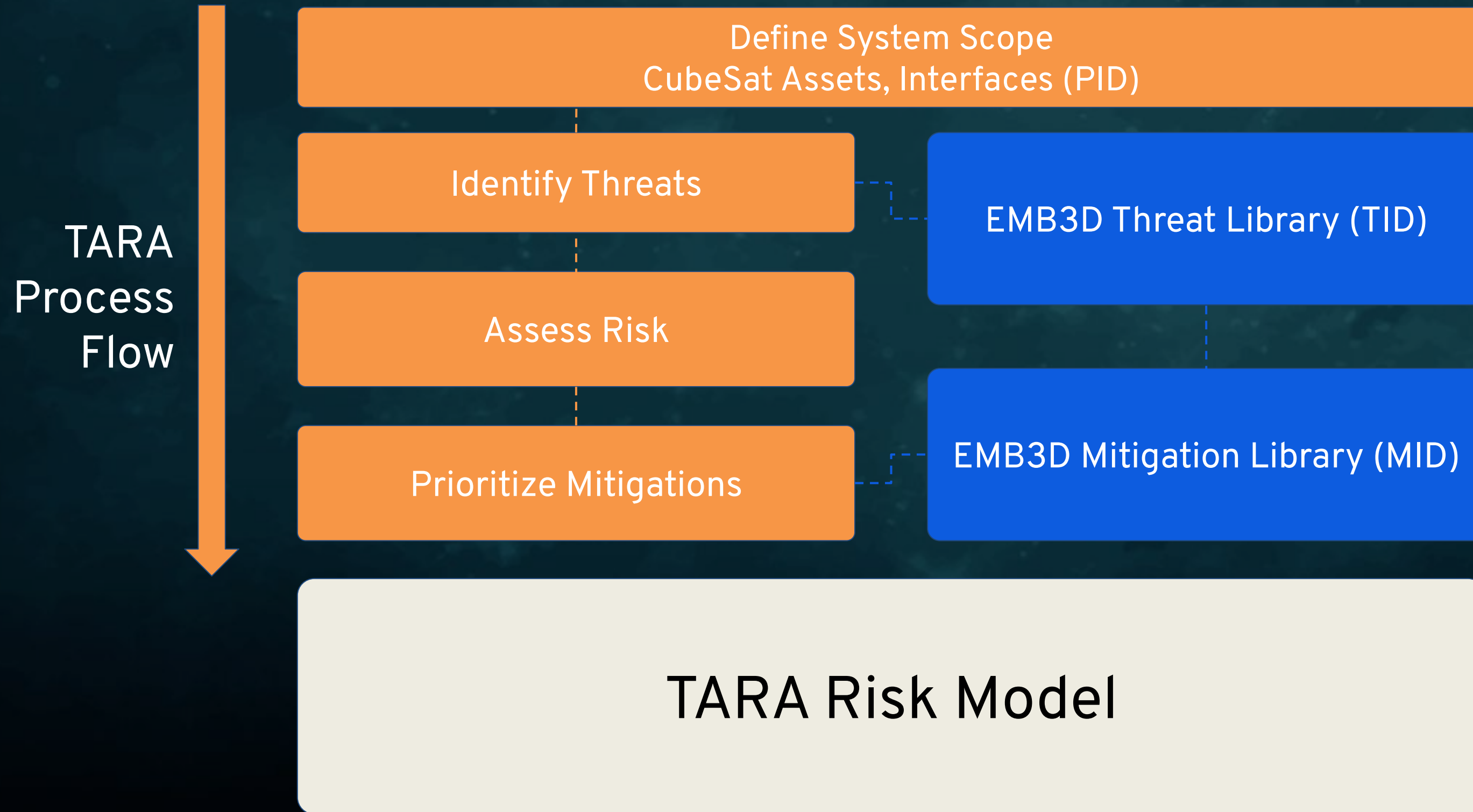
1. Identify Assets and Threats

2. Assess Exposure & Feasibility

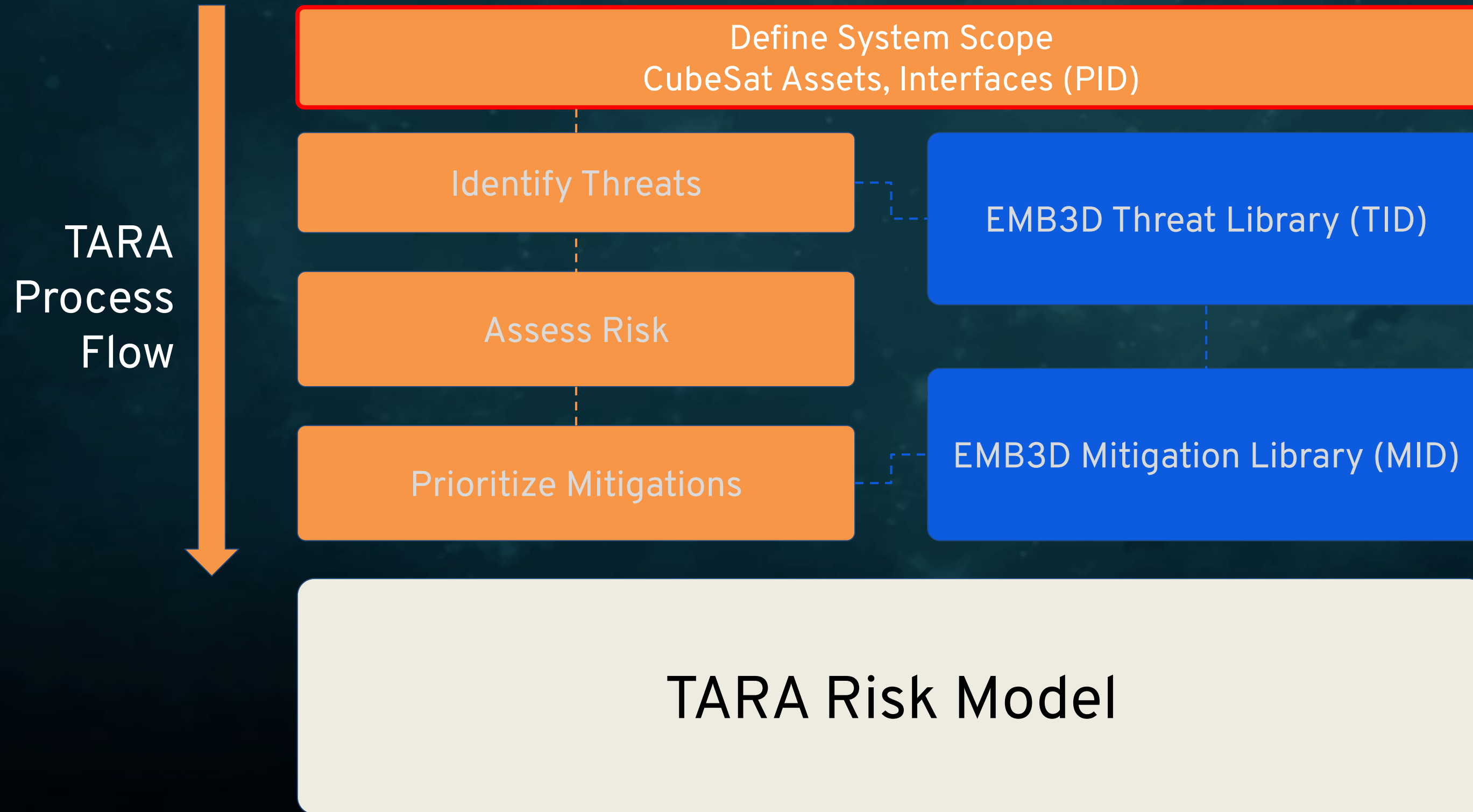
3. Estimate Impact

4. Determine Risk & Prioritize Mitigations

Integration of EMB3D Framework into the TARA Process



Integration of EMB3D Framework into the TARA Process



Identify and Define System Scope

1) Identify and Define System Scope by ...

- Extract system properties from the OBC datasheet
 - CPU, memory, interfaces, firmware, sensors
- Classify each asset (hardware, firmware, software, network)
- Match to EMB3D PIDs using the MITRE taxonomy (e.g., PID-11 Processor, PID-13 Buses, PID-23 OS)
- Validate with datasheet evidence for accuracy

Technical Specifications

General	
Processor	Dual-Core LEON3-FT Sparc V8
System Clock	40 MHz (up to 80 MHz)
FPU	IEEE-754 compliant, double precision
Boot Memory	2 MB
Non-Volatile Memory	8 MB MRAM
Operational Memory	32 MB SRAM
Operating Temperature	- 30°C to +70°C
Timing Synchronization	500 us with PPS
Radiation	- TID: > 20 kRAD - SEE: > 43 MeV·cm ² /mg
Expected Lifespan	> 5 years in LEO
EMC Compliance	MIL-STD-461

Interfaces	
CAN	2x CAN 2.0, 1 Mbps
SpaceWire	4x SpW, 40 Mbps
MIL-STD-1553	2x MIL-STD-1553B, dual bus with Bus Controller capability
UART	6x RS422, 115.2 Kbps
TC/TM	2x LVDS, 250 Kbps, CCSDS compliant
PPS	2x RS422/LVDS
Development & Debugging	JTAG & Ethernet

Size, Weight & Power	
Volume	215 x 175 x 30 mm ³
Mass	0.95 Kg
Input Voltage	16 V to 40 V
Nominal Power Consumption	4.8 W

Identify and Define System Scope

1-1) Identify NanoSat OBC Properties (Using certain COT OBC and university specs)

Category	Property	Details
Processor	ARM Cortex M4/M7 (180–216 MHz) / Dual-Core LEON3-FT (80 MHz)	Common CubeSat and radiation-tolerant options
Memory	2 MB Flash / 256 kB–2 MB RAM / optional 64 Mbit SRAM / 8 MB MRAM	Mix of volatile and non-volatile memory
Operating System	RTEMS (RTOS) or bare-metal firmware or FreeRTOS	Depends on variant
Interfaces	CAN 2.0 (1 Mbps), SpaceWire (40 Mbps), RS422 UART, SPI, I ² C, USB (VCP), PPS, JTAG	Standard CubeSat bus and debug support
Storage	MicroSD slot (256 Mbit Flash / 1 Gbit NOR)	External data logging / payload buffer
Sensors	Dual 3-axis accelerometers $\pm 2/4/8$ g; dual 3-axis digital compasses ± 16 gauss; sun and temperature sensors	For attitude determination and thermal monitoring
Actuators	3 $\times$ PWM outputs for magnetorquers (up to 3 A @ 3.3 V)	Attitude control
Encryption & Security	Companion FPGA for AES-256 encryption of TTC link	Hardware-based crypto support
Fault Tolerance	FDIR (Fault Detection, Isolation and Recovery); SEE tolerant; dual-redundant configuration	Space-grade reliability
Synchronization	PPS timing ≤ 500 μ s accuracy	Time-sync for payload data
Electrical	3.3 V logic; input 16–40 V; typical power ≈ 4.8 W	Compatible with CubeSat EPS
Environmental	Operating -30 $^{\circ}$ C to $+70$ $^{\circ}$ C (LEON3) / -30 $^{\circ}$ C to $+85$ $^{\circ}$ C (Type I); TID > 20 krad	LEO radiation-tolerant
Physical	215 $\times$ 175 $\times$ 30 mm, 0.95 kg (LEON3) / 58 g (Type I)	Varies by OBC generation
Qualification	ESA ECSS, GEVS thermal / vibration tested	Flight-ready CubeSat component

Identify and Define System Scope

2) Property selection can be easily done with MITRE EMB3D properties mapper

MITRE | **EMB3D**

MITRE EMB3D™
*A knowledge base of cyber threats
and associated mitigations for
embedded devices*

[HOME](#)
[BLOG](#)
[BACKGROUND](#)
[PROPERTIES LIST](#)
[PROPERTIES MAPPER](#)

Device Properties:

Hardware:

☒ PID-11 - Device includes a microprocessor

☐ PID-12 - Device includes Memory/Storage (external to CPU)

☐ PID-13 - Device includes peripheral chips and integrated data buses

☐ PID-14 - Device includes external peripheral interconnects (e.g., USB, Serial)

☐ PID-15 - Device includes a hardware access port (e.g., UART, JTAG)

Applicable Threats:

PID-11

- [TID-101](#) - Power Consumption Analysis Side Channel
- [TID-102](#) - Electromagnetic Analysis Side Channel
- [TID-103](#) - Microarchitectural Side Channels
- [TID-105](#) - Hardware Fault Injection – Control Flow Modification

Download CSV

Identify and Define System Scope

2) Identify Assets (MITRE EMB3D) & Select matching PIDs

- PID-11 - Device includes a microprocessor
 - TID-101 - Power Consumption Analysis Side Channel
 - TID-102 - Electromagnetic Analysis Side Channel
 - TID-103 - Microarchitectural Side Channels
 - TID-105 - Hardware Fault Injection – Control Flow Modification
- PID-15 Device includes a hardware access port (e.g., UART, JTAG)
 - TID-115 - Firmware/Data Extraction via Hardware Interface
 - TID-116 - Latent Privileged Access Port
 - TID-119 - Latent Hardware Debug Port Allows Memory/Code Manipulation
- ETC ...

Identify and Define System Scope

2-1) Map Properties to PID

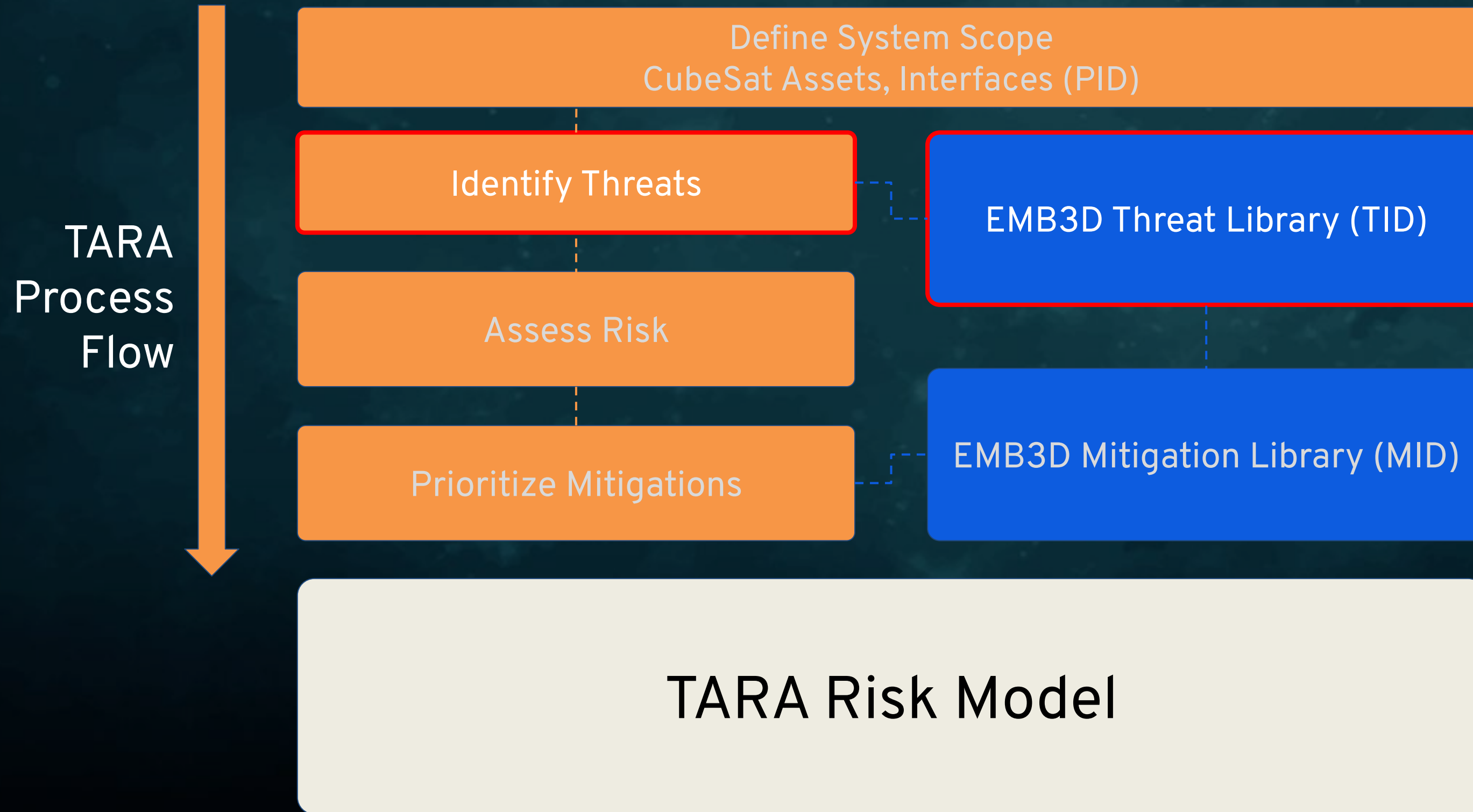
Device	PID	Property / Component	Description / Evidence
hardware	PID-11	Microprocessor	ARM Cortex-M4/M7 (180–216 MHz) or Dual-Core LEON3-FT (80 MHz); executes flight and control software
	PID-12	Memory Subsystem	Non-volatile Flash (2 MB – 1 Gbit), MRAM (8 MB), SRAM (32 MB), optional 64 Mbit RAM
	PID-121	Memory Bus	Internal/external memory data buses connecting CPU–SRAM–Flash
	PID-123	Removable / External Storage	MicroSD slot and serial NOR Flash for payload logging
	PID-124	Volatile Memory	Working SRAM / RAM used during operations
	PID-13	Peripheral Subsystem & Buses	CAN 2.0, I ² C, SPI, SpaceWire, UART, RS-422, PPS time sync interfaces
	PID-14	External Ports / Interfaces	USB (VCP), PC/104, ProtoBoard connectors (JP2/JP14), external payload links
	PID-15	Debug Access	JTAG port, UART console (Sys Tx/Rx), debug Ethernet
System Software	PID-21	Bootloader	Built-in boot sequence in ROM for firmware loading
	PID-22	Debugging Capabilities	Diagnostic access via JTAG, USB, UART during integration and test

Identify and Define System Scope

2-1) Map Properties to PID

Device	PID	Property / Component	Description / Evidence
System Software	PID-23	Operating System / Firmware	RTEMS RTOS (or bare-metal firmware) controlling satellite operations
	PID-231	Loadable Modules	RTOS-based or custom loadable flight applications
	PID-27	Firmware Update Capability	Supports firmware re-flashing over debug / bus interface
	PID-271 / PID-272	Update Integrity Check	Integrity validation (PID-272) if signed updates; none if PID-271
	PID-28	System Logging	System logs stored on SD / Flash for telemetry
Application Software	PID-31	Application Software	Flight or payload control logic executed by OBC
	PID-3122	Programming Language (C/C++)	Bare-metal C/C++ firmware controlling OBC logic
	PID-32	Custom App Deployment	Payload modules or research scripts deployable to OBC
	PID-321 / PID-3231	Execution Method	Custom payload binaries executed without confinement
	PID-34	Application Logging	Payload data / sensor logs (e.g., ADCS telemetry)
Networking	PID-41	Network Interfaces / Services	Telecommand & Telemetry (TTC) link and data buses
	PID-4113	Cryptographic Services	AES-256 FPGA module for TTC link encryption
Etc.	PID-13 + PID-14	Sensors / Actuators	Accelerometers, magnetometers, sun & temperature sensors; PWM magnetorquers (3 A @ 3.3 V)
	PID-28 + PID-15	Time Synchronization	PPS input ($\leq 500 \mu\text{s}$ accuracy) for logging and payload alignment
	PID-FDIR (Custom)	Fault Tolerance Mechanism	On-board Fault Detection, Isolation & Recovery (FDIR); dual-redundant configuration

Integration of EMB3D Framework into the TARA Process



Identify Threats

3) Map to Threats (TIDs)

- Properties mapping allows for threat taxonomy.
- E.g. PID-15 suggests hardware contains hardware access port.
 - Leads to threats such as data exfiltration through hardware interface (TID-115).

Properties Mapper allows for customizable list of PIDs that map to TIDs.

The screenshot displays the MITRE EMB3D Properties Mapper web application. The interface is divided into three main sections: a left sidebar, a central 'Device Properties' area, and a right 'Applicable Threats' area.

Left Sidebar: Contains the MITRE EMB3D logo and a navigation menu with the following items: HOME, BLOG, BACKGROUND, PROPERTIES LIST, **PROPERTIES MAPPER** (highlighted with a red box), THREATS, HARDWARE, SYSTEM SOFTWARE, APPLICATION SOFTWARE, NETWORKING, MITIGATIONS, and FOUNDATIONAL.

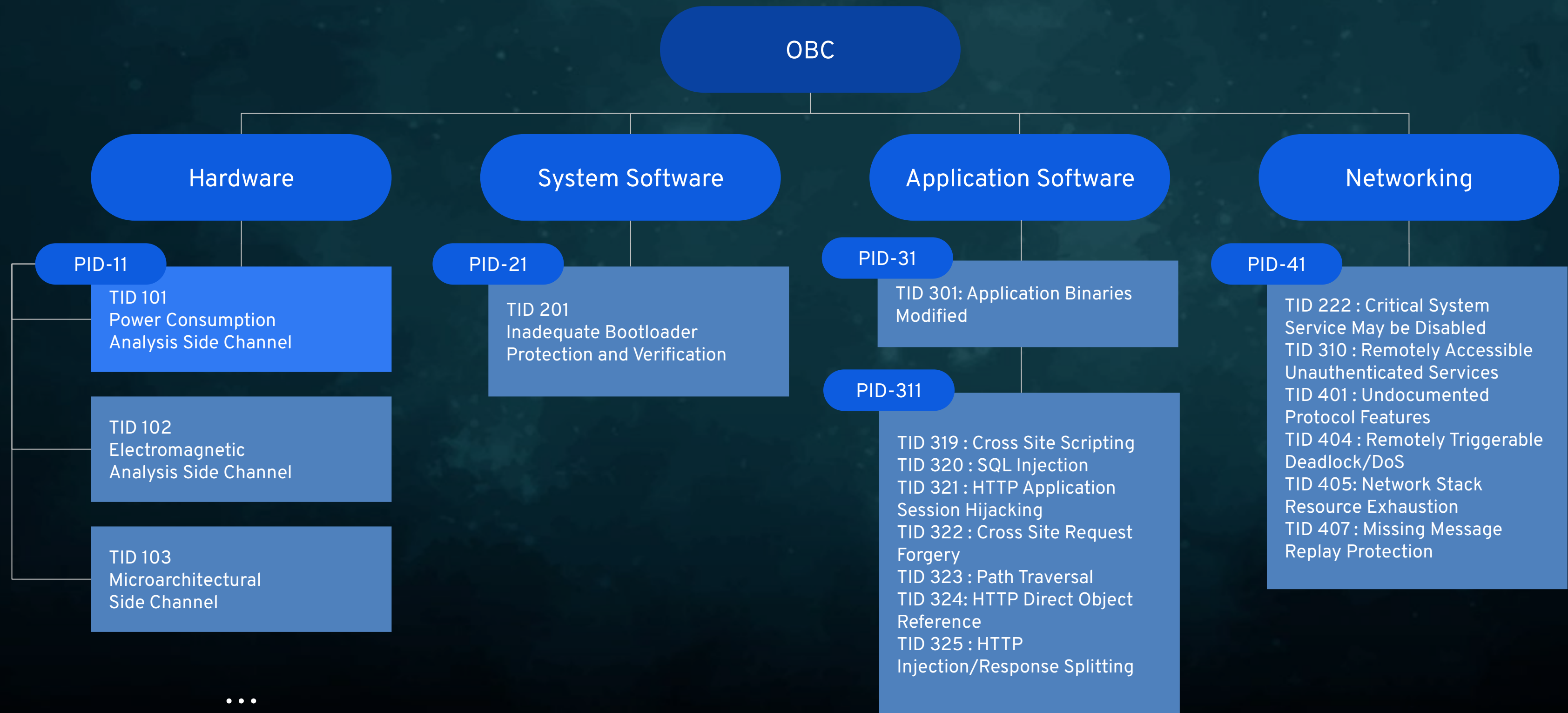
Device Properties: This section is divided into two sub-sections:

- Hardware:** Lists various hardware properties with checkboxes. PID-15 is checked, indicating 'Device includes a hardware access port (e.g., UART, JTAG)'. Other listed properties include microprocessors, memory/storage, peripheral chips, and external interconnects.
- System Software:** Lists software-related properties. PID-23 is checked, indicating 'Device includes OS/kernel'. Other listed properties include bootloaders, debugging capabilities, and operating systems.

Applicable Threats: This section displays a list of threats associated with the selected properties. A 'Download CSV' button is present. The threats listed are:

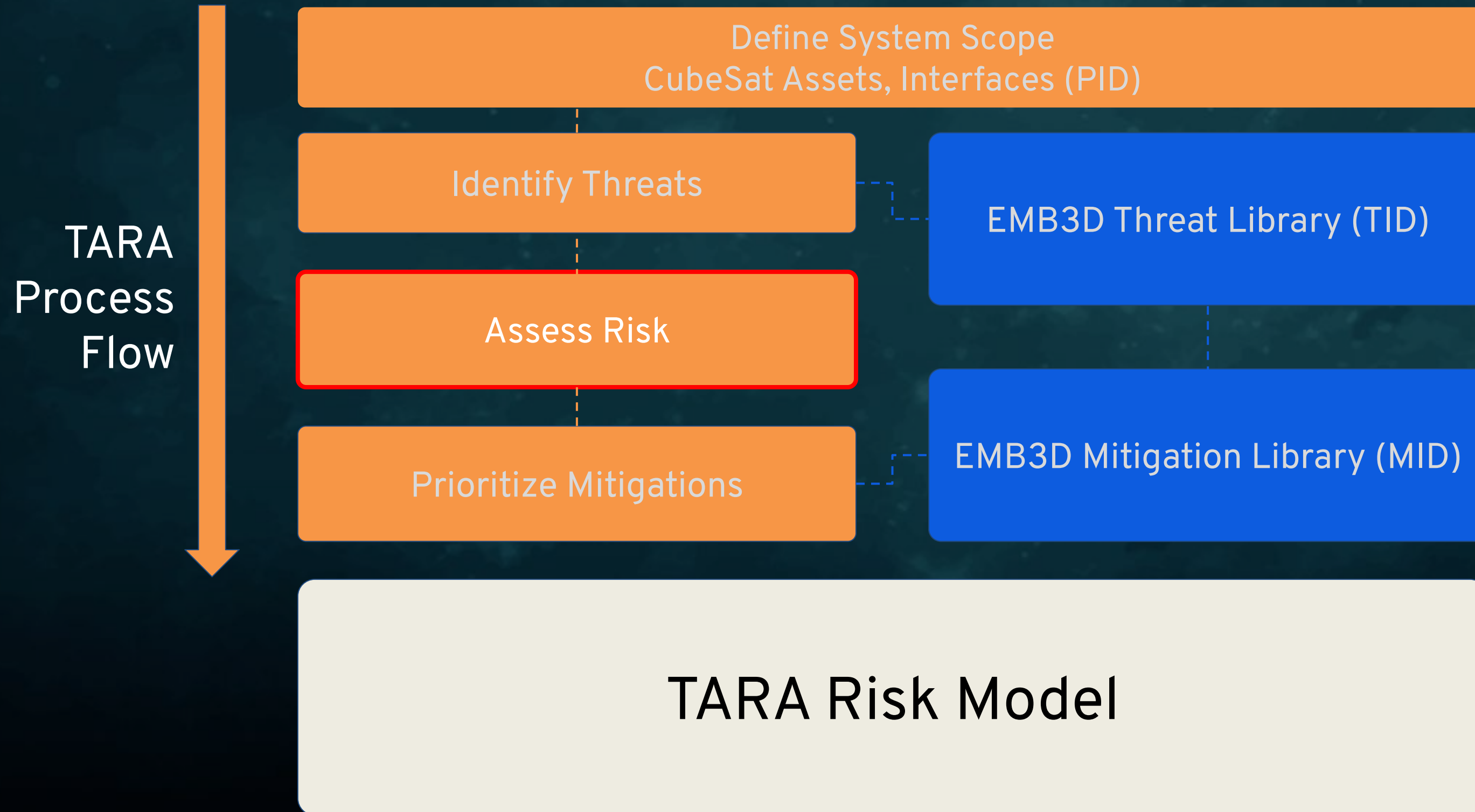
- PID-15:** TID-115 - Firmware/Data Extraction via Hardware Interface, TID-116 - Latent Privileged Access Port, TID-119 - Latent Hardware Debug Port Allows Memory/Code Manipulation.
- PID-23:** TID-202 - Exploitable System Network Stack Component, TID-218 - Operating System Susceptible to Rootkit.
- PID-23221:** TID-205 - Existing OS Tools Maliciously Used for Device Manipulation, TID-219 - OS/Kernel Privilege Escalation.
- PID-23222:** TID-206 - Memory Management Protections Subverted, TID-223 - System Susceptible to RAM Scraping.

Identify Threats



(Not an exhaustive list of all threats on a nanosat OBC)

Integration of EMB3D Framework into the TARA Process



Assess Risk

4) Assess Exposure & Feasibility

Metrics

- Accessibility (Remote, Proximal, Physical)
- Complexity (Low vs. High skill required)
- Required Resources (Time, tools, expertise)

Examples

- Debug port → Physical access required → Moderate exposure
- RF interface → Remote access → High exposure

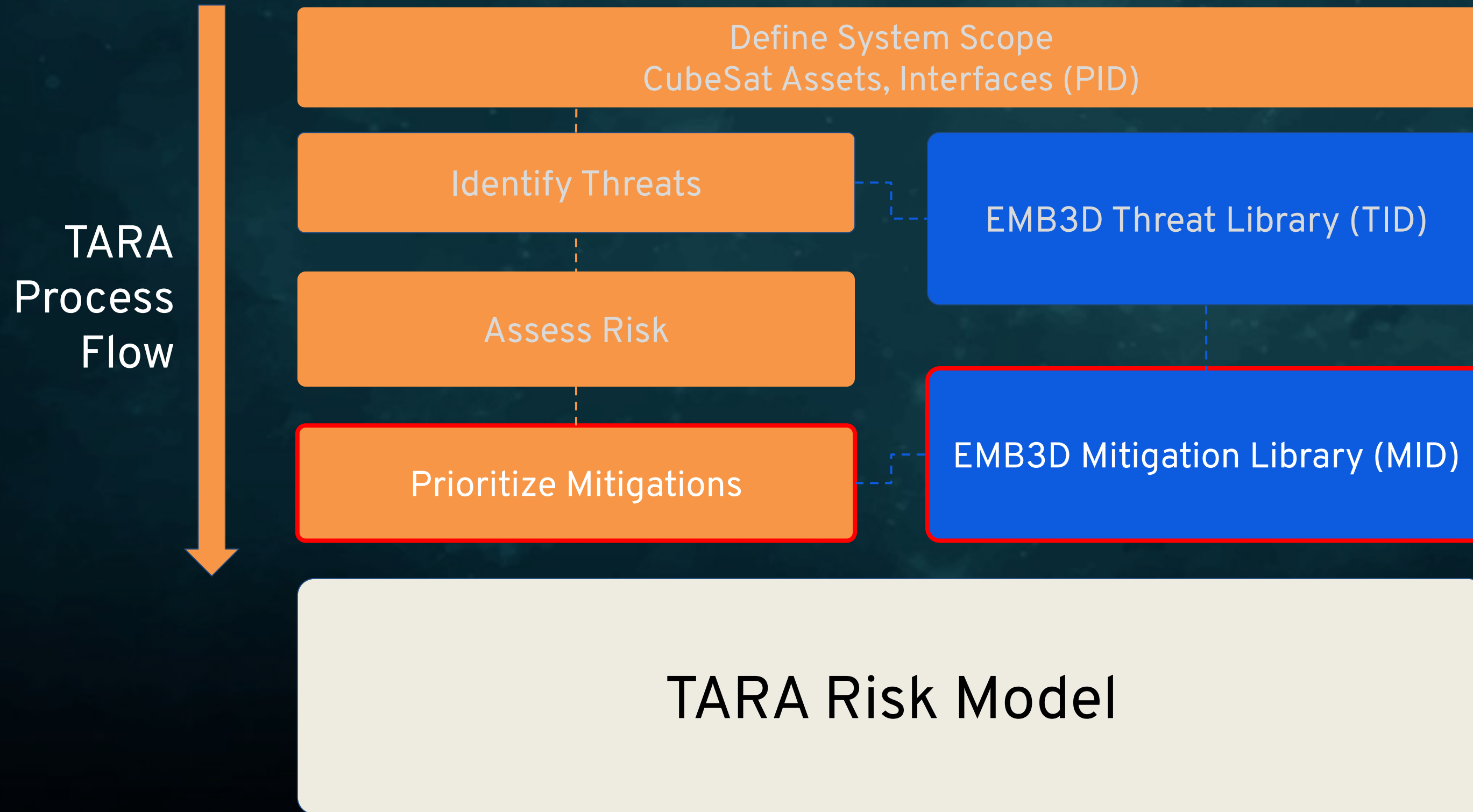
Assess Risk

4-1) Assess Impact and Risk (Risk Scoring)

$$Risk\ Score = \sum ((risk\ factor\ value_i \times factor\ weighting_i)) \qquad (1)$$

Factor Range	Low = 1	Medium = 2	High = 3	Factor Weight	TID-119	TID-115	TID-XXX
Locality: How localized are the effects posed by this TTP?	isolated to single unit	external networks potentially impacted	all units globally and associated infrastructure	0.2	3	2	X
Impact (Confidentiality): How serious an impact is loss of data confidentiality resulting from successful application of this TTP?	no impact from TTP	limited impact requiring some remediation	data spills routinely exercised	0.2	3	3	X
Impact (Availability): How serious an impact is loss of system availability resulting from successful application of this TTP?	no impact from TTP	limited impact requiring some remediation	simulated system outages routinely exercised	0.2	3	2	X
Prior Use: Evidence this TTP has been used before?	no evidence of TTP use	confirmed evidence of TTP use	widespread use of TTP reported	0.3	3	2	X
Stealth: How detectable is this TTP when it is applied?	TTP obvious without monitoring	detection likely with routine monitoring	undetectable	0.1	3	3	X
Score				1.0	3.0	2.3	X

Integration of EMB3D Framework into the TARA Process



Prioritize Mitigations

5) Determine Risk & Prioritize Mitigations

Threat Susceptibility Matrix

ID	Attack Vector Name	Risk Score	Evaluation Targets						
			OS/Firmware	Ports & Debug	Buses	CPU	Boot/Update	Memory/Storage	Cryptography
TID-119	Latent Hardware Debug Port Allows Memory/Code Manipulation	3.0	X	X					
TID-115	Firmware/Data Extraction via Hardware Interface	2.3		X			X		
TID-105	Hardware Fault Injection – Control Flow Modification	X.X				X	X		
TID-101	Power Consumption Analysis Side Channel	X.X				X			
TID-102	Electromagnetic Analysis Side Channel	X.X				X			
TID-103	Microarchitectural Side Channels	X.X	X			X			
TID-XXX	xxxxxxxxxxxxxxxxxxxxxxxxxxxx	X.X	-	-	-	-	-	-	-

TID-119 Related Threats:
TID-115, TID-116

Prioritize Mitigations

6) Identify EMB3D Mitigations (MIDs)

PID 15 ↔ TID 119 ↔ MID 057, MID 058

Example Threat: **TID 119**- Latent Hardware Debug Port Allows Memory/Code Manipulation

Threat Description:

Hardware debugging ports (e.g. JTAG, SWD, etc.) oftentimes have high privileges or director access to the running device's memory and integrated hardware. By leveraging one of these hardware ports, an adversary may be able to read memory values from the device, change the value of a section of memory during runtime, or control the execution of code on the processor. This can give threat actors increased privileges on the device or bypass other security protections.

CWE-1191: On-Chip Debug and Test Interface With Improper Access Control

Suggested Mitigations:

- MID-057: Disable Physical Development and Debugging Ports (Foundational)
- MID-058: Engage hardware Readout Protection Mechanisms (Foundational)

Mitigation Tiers

- ❑ **Foundational: Basic software defenses**
- ❑ Intermediate: Mixed HW/SW protections
- ❑ Leading: Hardware-enforced, cryptographically strong measures

IEC 62443 4-2 Mappings

- EDR / HDR / NDR 2.13 – Use of physical diagnostic and test interfaces
- EDR / HDR / NDR 3.11 – Physical tamper resistance and detection

Prioritize Mitigations

6) Identify EMB3D Mitigations (MIDs)

- Corresponding Mitigations for Evaluations can be found via TID Mitigations

MITRE | **EMB3D**

MITRE EMB3D™
A knowledge base of cyber threats and associated mitigations for embedded devices

TID-118

TID-119

→

SYSTEM SOFTWARE

↓

→

APPLICATION SOFTWARE

↓

→

NETWORKING

↓

CVE

- None referenced

Related Threats:
[TID-115](#), [TID-116](#)

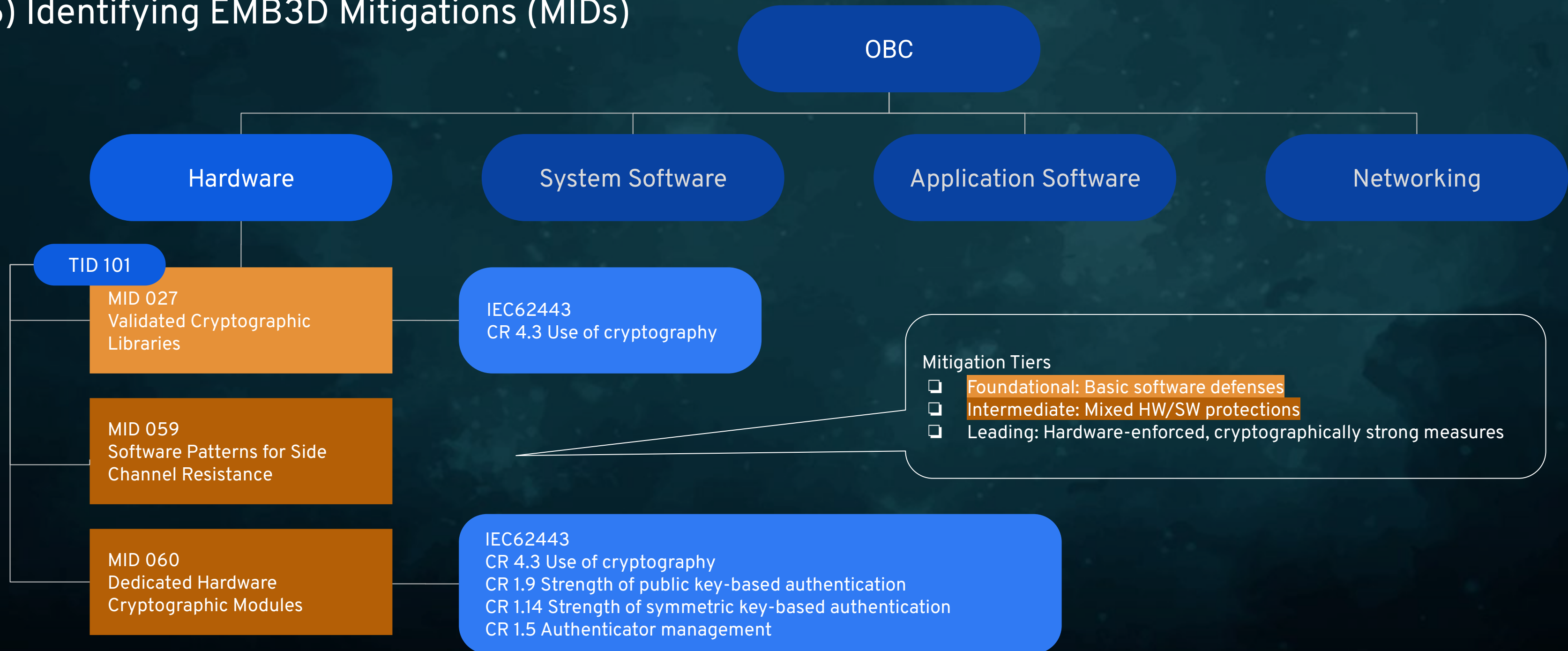
Device Properties:
[PID-15](#) - Device includes a hardware access port (e.g., UART, JTAG)

Mitigations:

Foundational	Intermediate	Leading
MID-057 - Disable Physical Development and Debugging Ports MID-058 - Engage Hardware Readout Protection Mechanisms		

Prioritize Mitigations

6) Identifying EMB3D Mitigations (MIDs)



Prioritize Mitigations

6) Identifying EMB3D Mitigations (MIDs)

- Which mitigation comes first?

PID-15	IID-119	proof of concept	MID-057	Foundational
PID-15	IID-119	proof of concept	MID-058	Foundational
PID-21	IID-201	observed adversarial technique	MID-001	Foundational
PID-21	IID-201	observed adversarial technique	MID-002	Intermediate
PID-21	IID-201	observed adversarial technique	MID-029	Intermediate
PID-21	IID-201	observed adversarial technique	MID-003	Leading
PID-22	IID-224	observed adversarial technique	MID-018	Foundational
PID-22	IID-224	observed adversarial technique	MID-039	Foundational
PID-22	IID-224	observed adversarial technique	MID-039	Foundational
PID-22	IID-224	observed adversarial technique	MID-039	Foundational
PID-22	IID-224	observed adversarial technique	MID-039	Foundational
PID-22	IID-224	observed adversarial technique	MID-039	Foundational
PID-23	IID-202	known exploitable weakness	MID-004	Foundational
PID-23	IID-202	known exploitable weakness	MID-005	Intermediate
PID-23	IID-202	known exploitable weakness	MID-007	Intermediate
PID-23	IID-202	known exploitable weakness	MID-008	Intermediate
PID-23	IID-202	known exploitable weakness	MID-014	Intermediate
PID-23	IID-202	known exploitable weakness	MID-006	Leading
PID-23	IID-202	known exploitable weakness	MID-086	Leading
PID-23	IID-202	known exploitable weakness	MID-088	Leading
PID-23	IID-202	known exploitable weakness	MID-089	Leading
PID-23	IID-218	known exploitable weakness	MID-001	Foundational
PID-23	IID-218	known exploitable weakness	MID-002	Intermediate
PID-23	IID-218	known exploitable weakness	MID-009	Intermediate
PID-23	IID-218	known exploitable weakness	MID-003	Leading
PID-231	IID-203	observed adversarial technique	MID-001	Foundational
PID-231	IID-203	observed adversarial technique	MID-010	Foundational
PID-231	IID-203	observed adversarial technique	MID-011	Foundational
PID-231	IID-203	observed adversarial technique	MID-002	Intermediate
PID-231	IID-203	observed adversarial technique	MID-009	Intermediate
PID-231	IID-203	observed adversarial technique	MID-003	Leading
PID-271	IID-211	observed adversarial technique	MID-001	Foundational
PID-271	IID-211	observed adversarial technique	MID-026	Foundational
PID-271	IID-211	observed adversarial technique	MID-002	Intermediate

Threat maturity score:

- hypothetical=1
- PoC=2
- observed=3
- known exploitable=4

Mitigation maturity score:

- foundational=3
- intermediate=2
- leading=1

(Threat Maturity Score) x (Mitigation Maturity Score)
- The higher, the more significant to mitigate

Prioritize Mitigations

6-1) Develop Mitigations

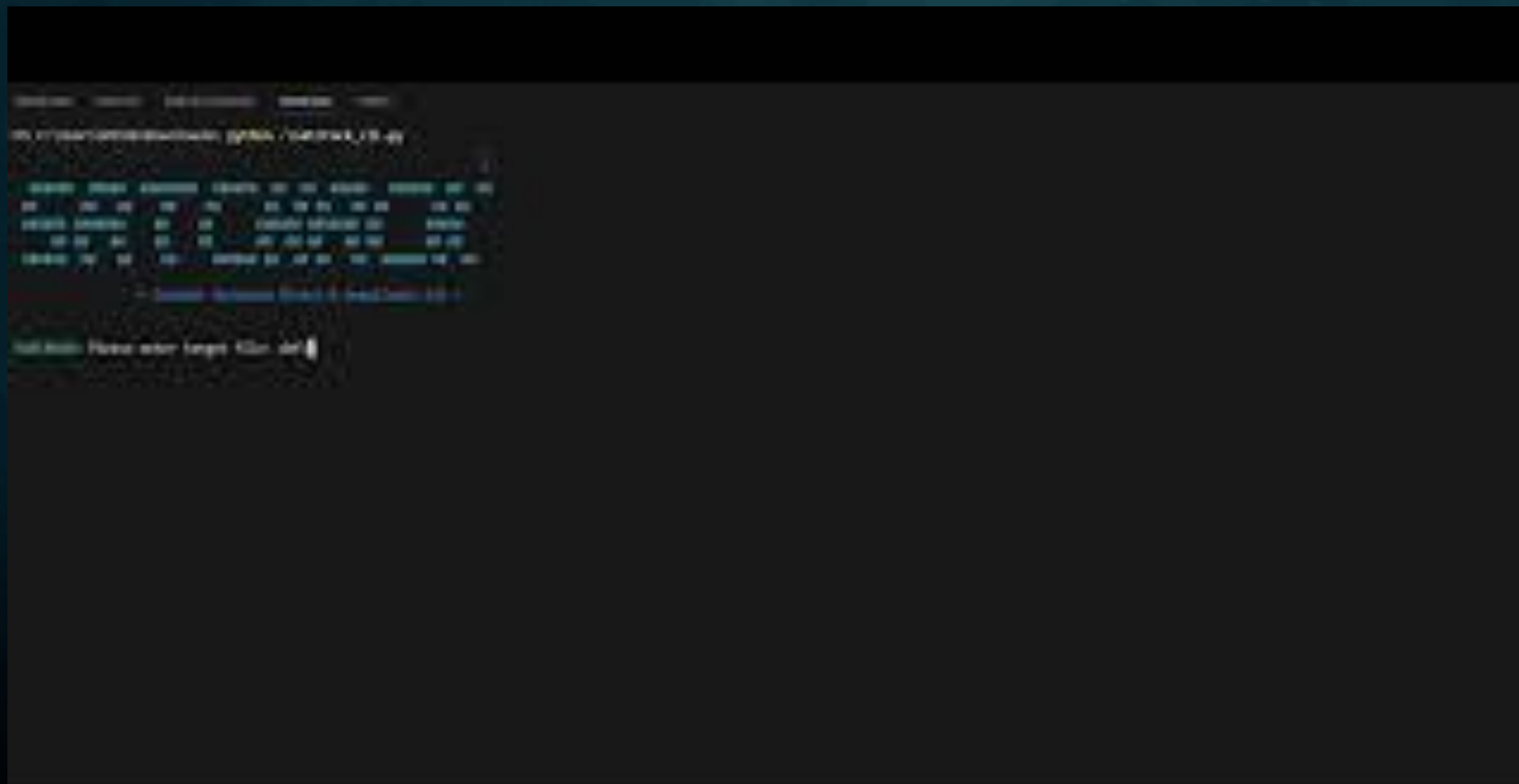
Combine AV/CM (Countermeasure) table with MID (mitigation ID)

Legend
P - Preventive
M - Mitigative
Cost Index - Very High to Very Low
(Depends on your organisation)

Countermeasure		Mitigation Effectiveness			
MID	Name	Cost Index	TID-119	TID-105	TID-101
MID-057	Disable Physical Development and Debugging Ports	Medium	M		
MID-058	Engage Hardware Readout Protection Mechanisms	High	M		
MID-063	Software Mitigations for Fault Injection	Low		M	
MID-066	Implement Redundant Processing and Memory	High		P	
MID-062	Hardware Mitigations for Fault Injection	Very high		M	
MID-027	Validated Cryptographic Libraries	Low			P
MID-059	Software Patterns for Side Channel Resistance	Medium			P
MID-060	Dedicated Hardware Cryptographic Modules	Very High			P

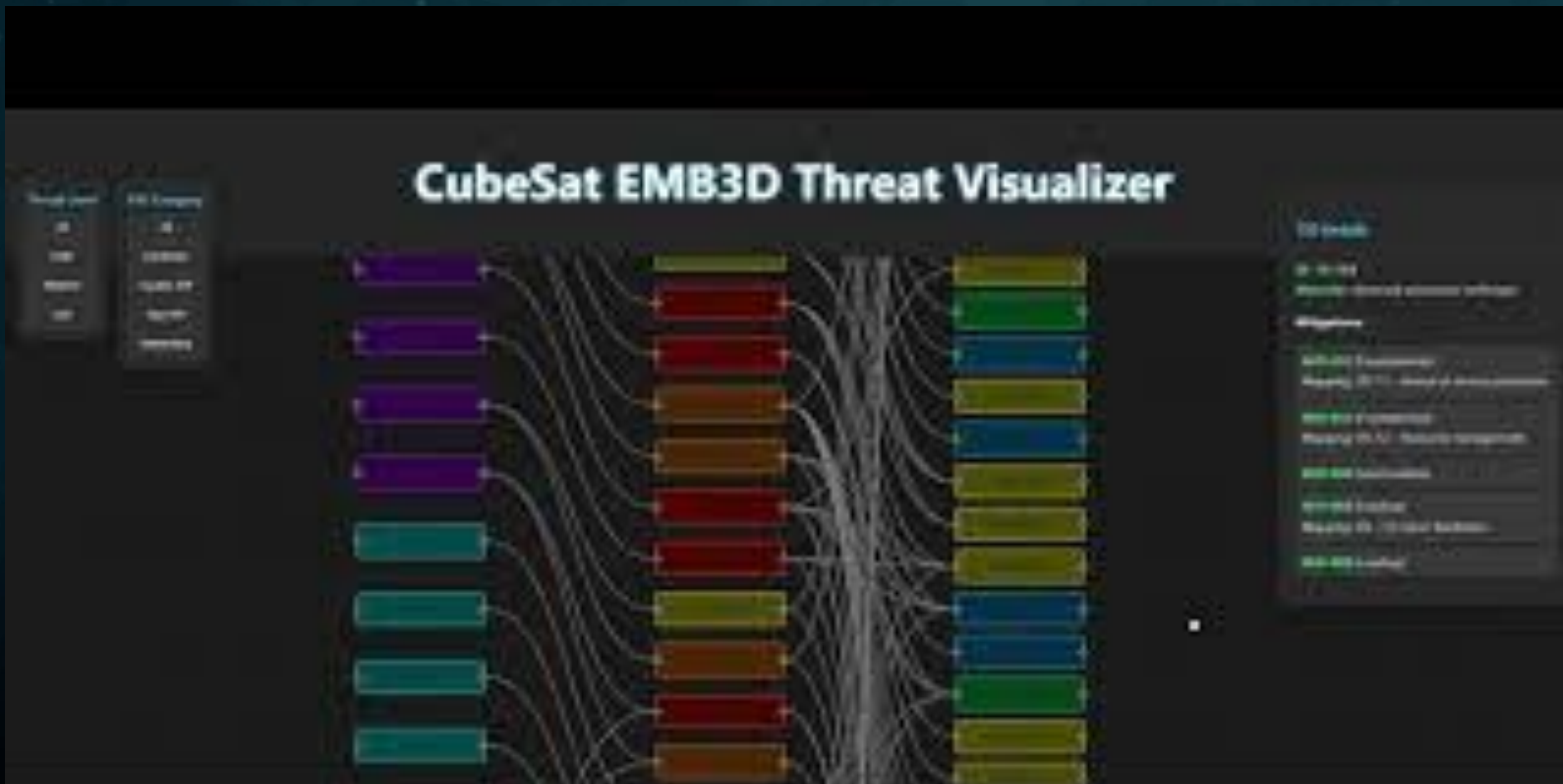
Prioritize Mitigations

6) Identifying EMB3D Mitigations (MIDs)



Prioritize Mitigations

6) Identifying EMB3D Mitigations (MIDs)



Conclusion

Using the TARA Model integrated with EMB3D

- Map out inventory of possible surface areas
- Identify and prioritise risk that these pieces bring
- Identify mitigation and preventive measures for all the risks identified along with the realistic cost of implementation
- Enabling a realistic response to potential cyber security risks
- Improving overall security posture while doing due diligence
- Conscientious of time and resources, enabling effective secure by design development

Concluding Remarks



NanoSats are Fundamentally Vulnerable

- Nanosats have constraints in power, computation, and memory. Security is usually a satellite team's last priority.



EMB3D and TARA are complementary

- MITRE EMB3D contains a formal approach (PID-TID-MID) that combines well with TARA in prioritizing cybersecurity risk.



Secure-by-Design isn't a luxury, it's launch-critical.

- It is possible to deploy secure-by-design satellite solutions while still fulfilling operational requirements.



<https://kinryu.sh/>

<https://www.linkedin.com/company/kinryulabs/>

Q&A



Klara Kim



Donavan Cheah



Davis Zheng



Yi-Ting Shen