eCTF @ 10

Lessons Learned from a Decade of Embedded Security Competitions

Ben Janis

June 2, 2025

eCTF 10

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

MITRE

eCTF 10

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

MITRE

About MITRE

Through our public-private partnerships and federally funded R&D centers, we work across government and in partnership with industry to tackle challenges to the safety, stability, and well-being of our nation.

Capture the Flags

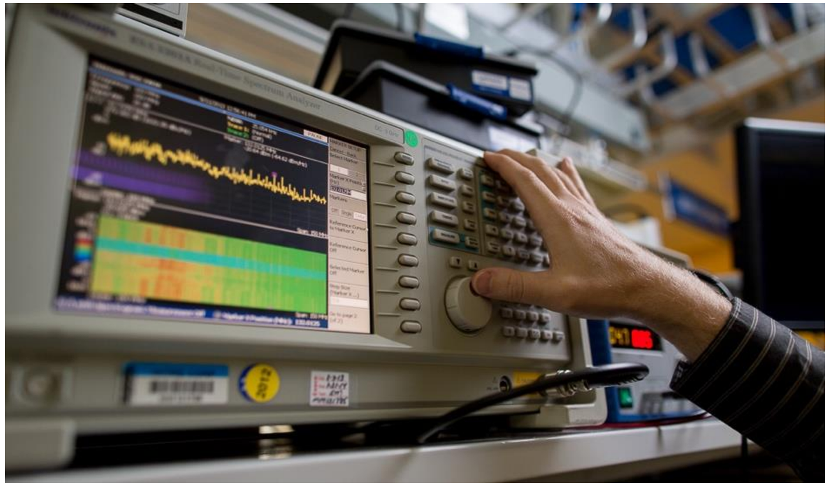
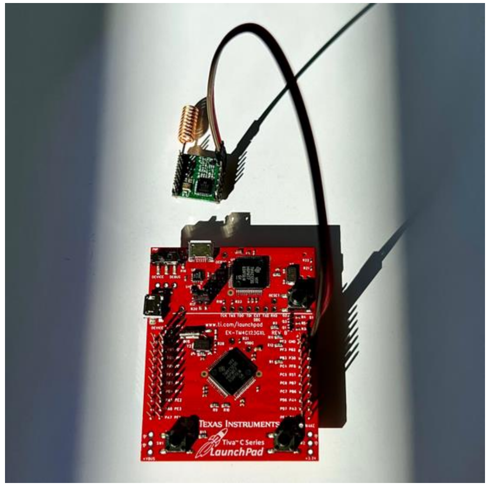
Cybersecurity Club @ Florida State University

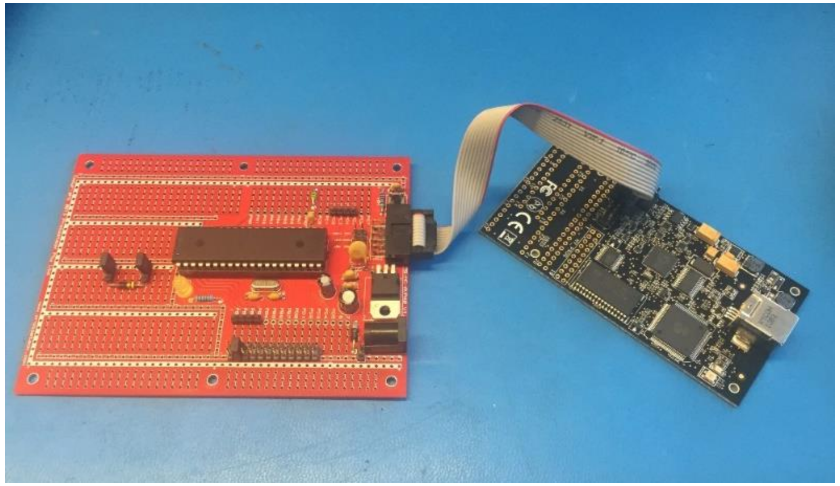
Photo: http://cybersecurity.cci.fsu.edu/

What Makes the eCTF Different?

Embedded Systems

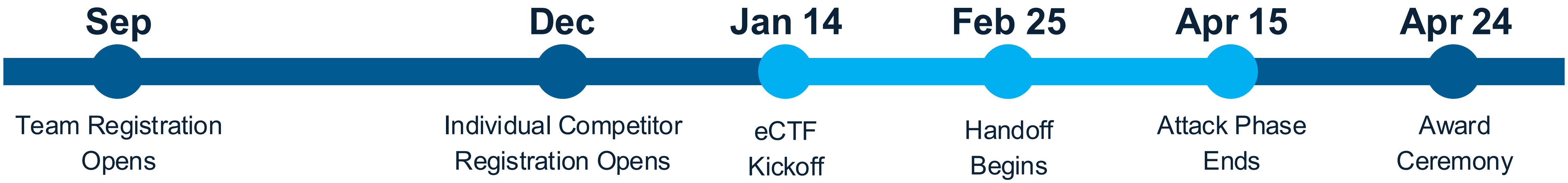
eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



MITRE

Extended Time



Sep – Team Registration Opens

Dec – Individual Competitor Registration Opens

Jan 14 – eCTF Kickoff

Feb 25 – Handoff Begins

Apr 15 – Attack Phase Ends

Apr 24 – Award Ceremony

Real-World Scenarios

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

- Smart Door Lock
- ATM Machine
- Self-Driving Car
- Drone Delivery
- Video Game Player
- Avionics

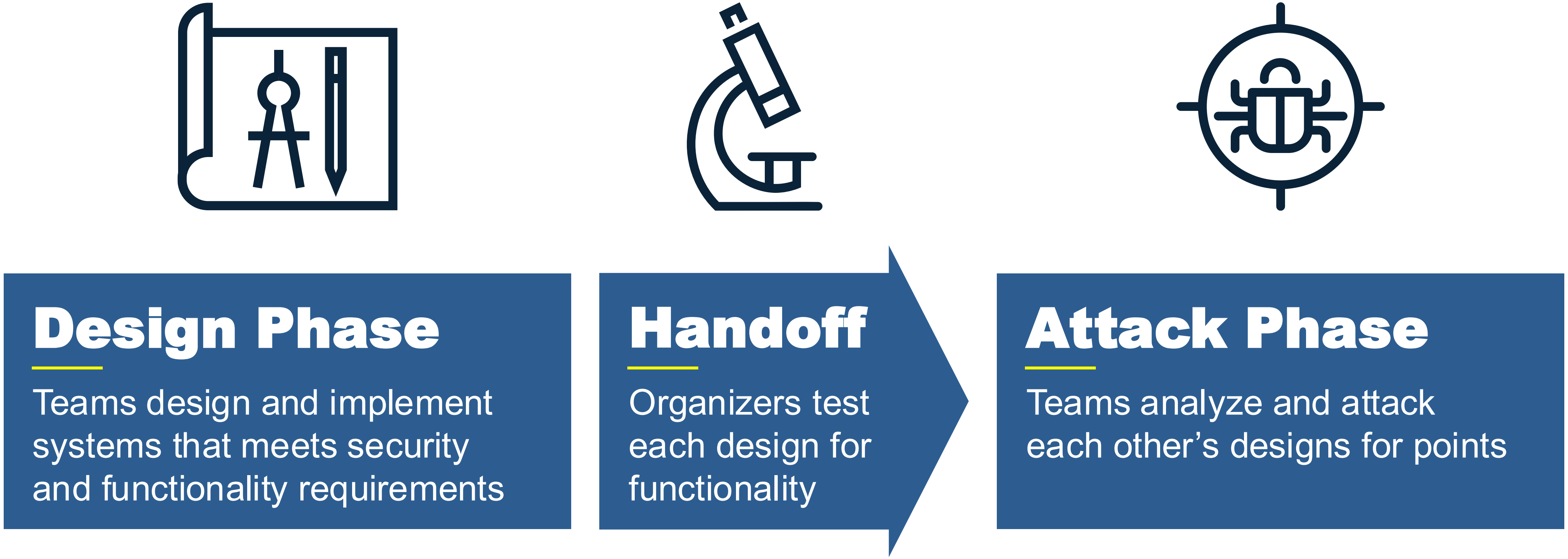
MITRE

What Students are Given

- Functional Requirements
- Hardware
- Deadlines
- Security Requirements
- Example Software
- Organizer Support

Competition Phases

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



Design Phase

Teams design and implement systems that meets security and functionality requirements

Handoff

Organizers test each design for functionality

Attack Phase

Teams analyze and attack each other’s designs for points

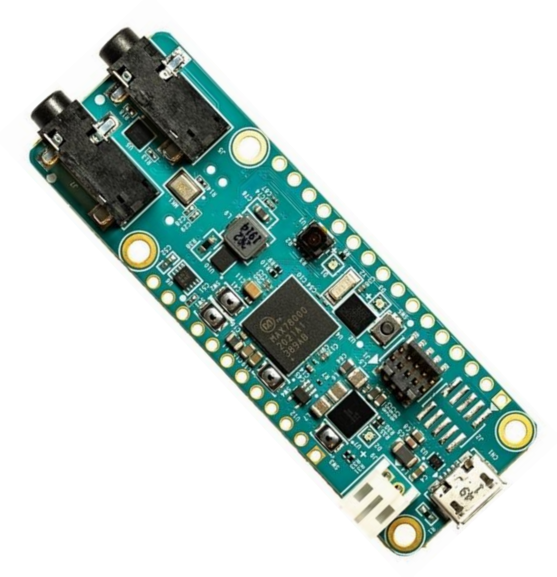
MITRE

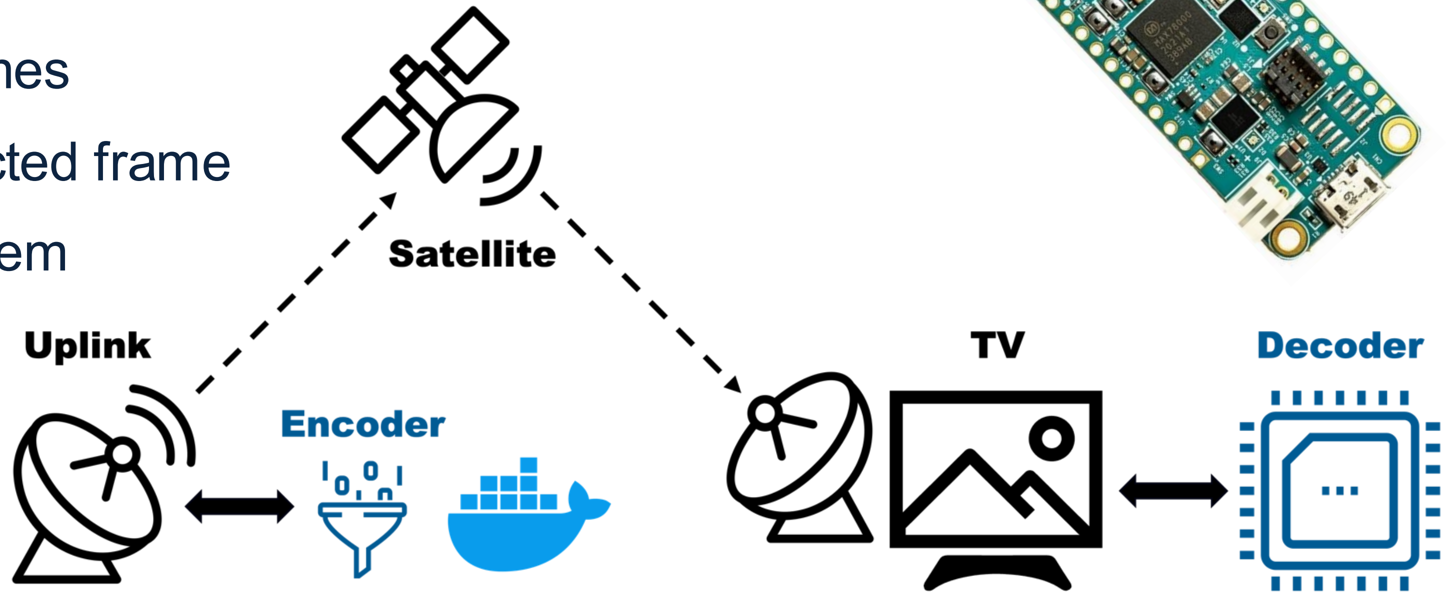
Modern eCTF

2025 Challenge Overview

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

- Teams were tasked with designing a secure implementation for a satellite TV
- MAX78000FTHR boards with ARM microcontroller
- Teams delivered:
  - Encoder to generate protected frames
  - Decoder firmware to decode protected frame
  - Tools to build and manage the system





MITRE

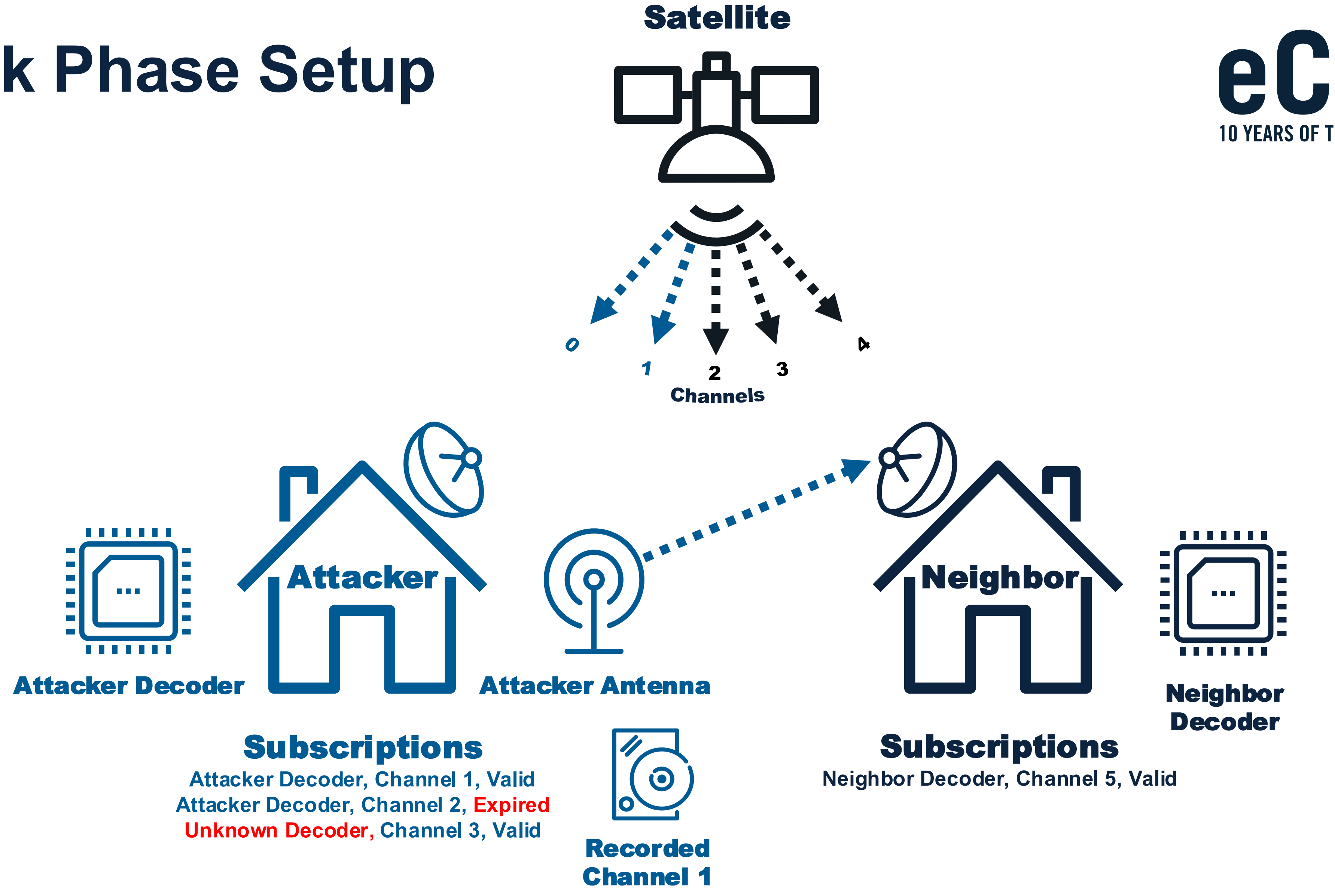
Security Requirements

1. An attacker should not be able to decode TV frames without a Decoder that has a valid, active subscription to that channel

2. The decoder should only decode valid TV frames generated by the Satellite System the Decoder was provisioned for

3. The Decoder should only decode frames with increasing timestamps

Attack Phase Setup



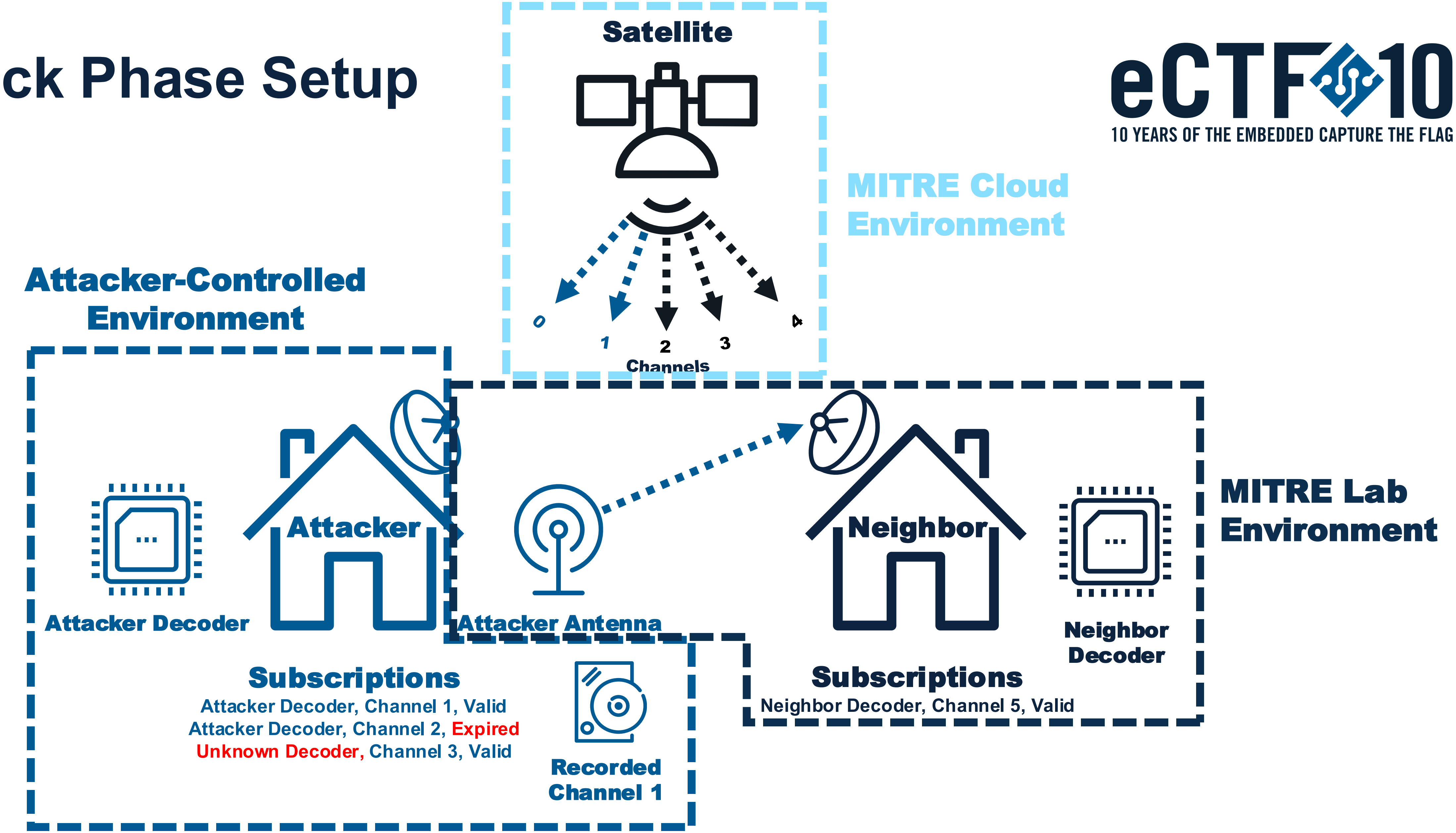
Subscriptions

Attacker Decoder, Channel 1, Valid
Attacker Decoder, Channel 2, Expired
Unknown Decoder, Channel 3, Valid

Subscriptions

Neighbor Decoder, Channel 5, Valid

Attack Phase Setup



Subscriptions

Attacker Decoder, Channel 1, Valid
Attacker Decoder, Channel 2, Expired
Unknown Decoder, Channel 3, Valid

Subscriptions

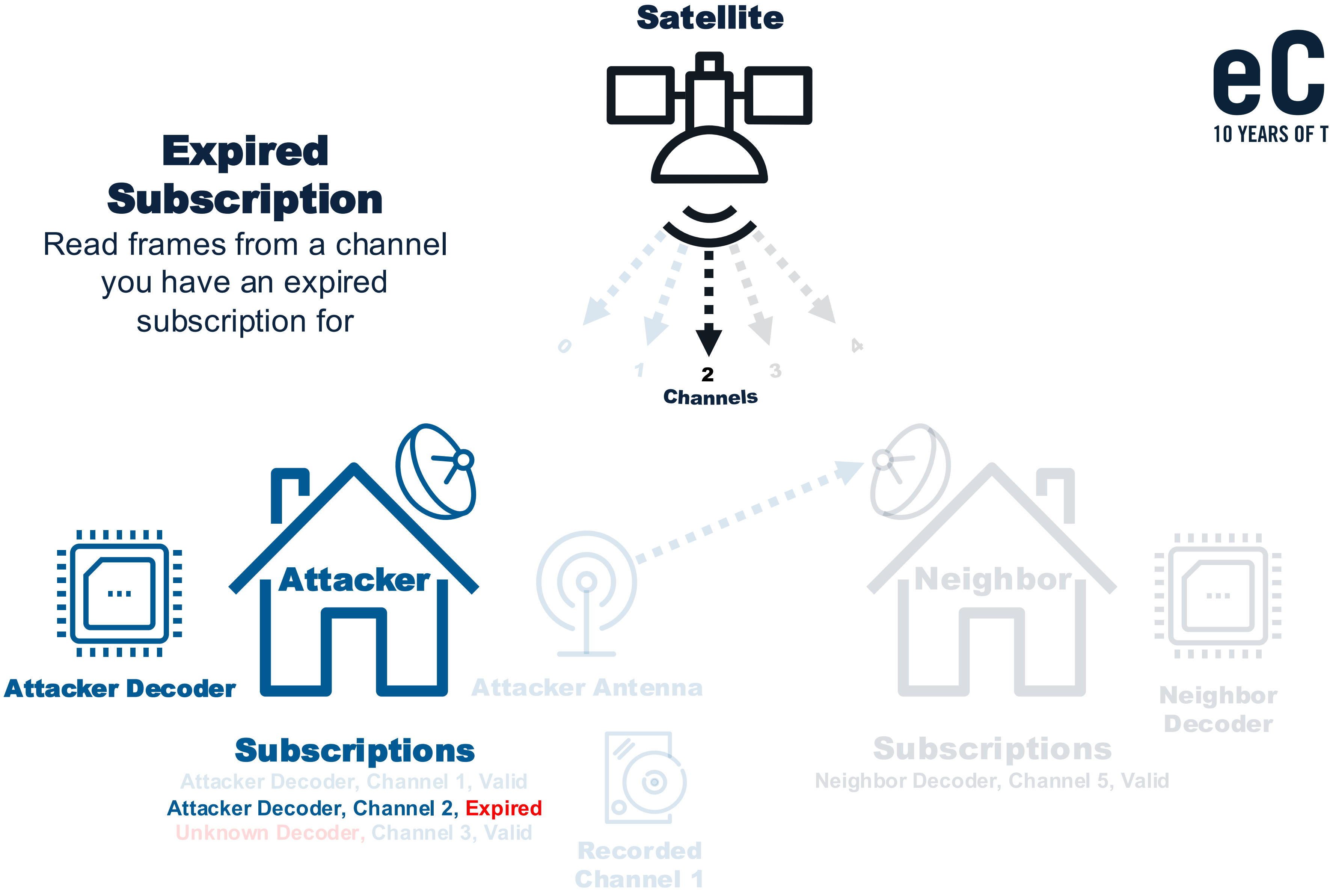
Neighbor Decoder, Channel 5, Valid

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

MITRE

Expired Subscription

Read frames from a channel you have an expired subscription for



Subscriptions

Attacker Decoder, Channel 1, Valid

Attacker Decoder, Channel 2, Expired

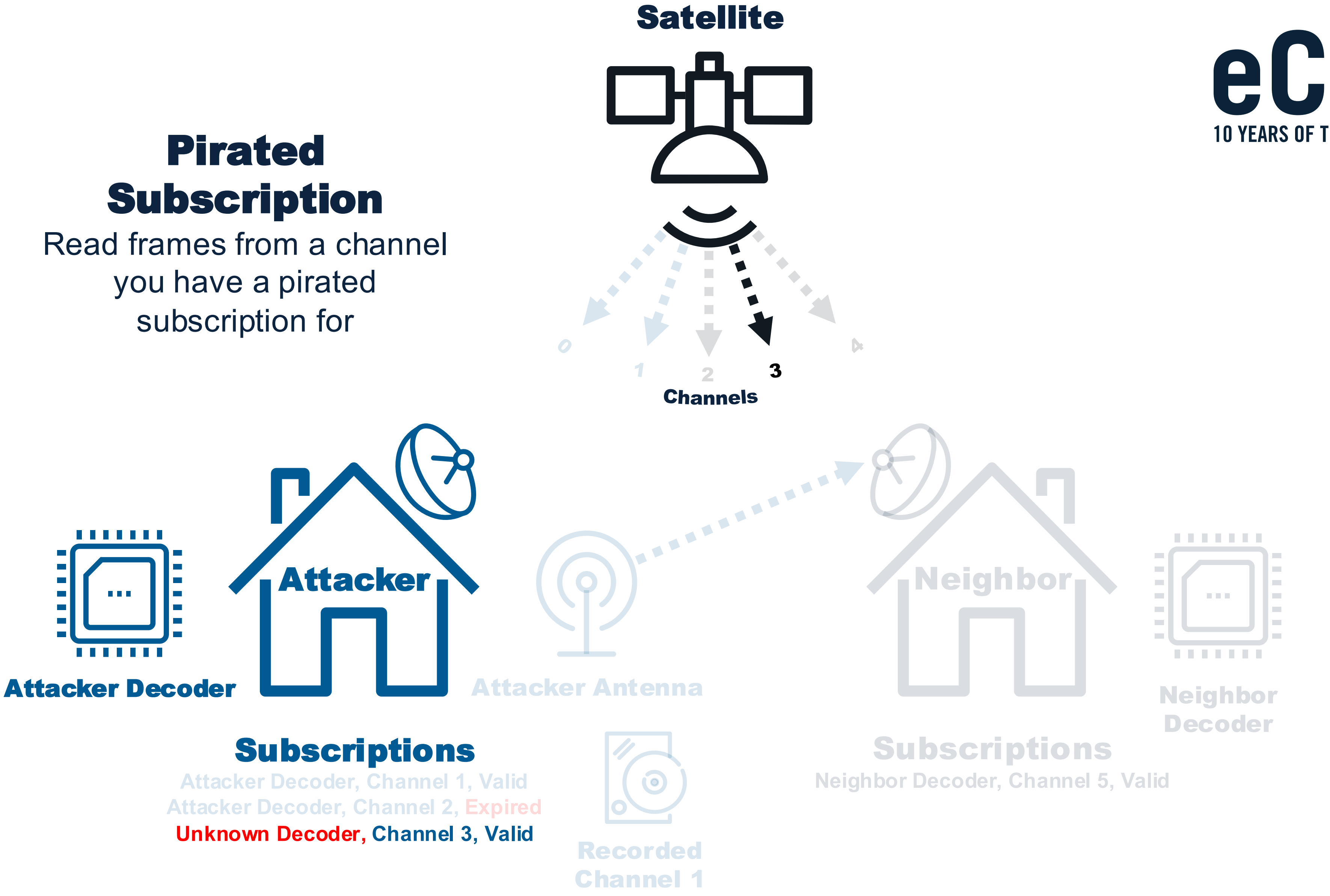
Unknown Decoder, Channel 3, Valid

Subscriptions

Neighbor Decoder, Channel 5, Valid

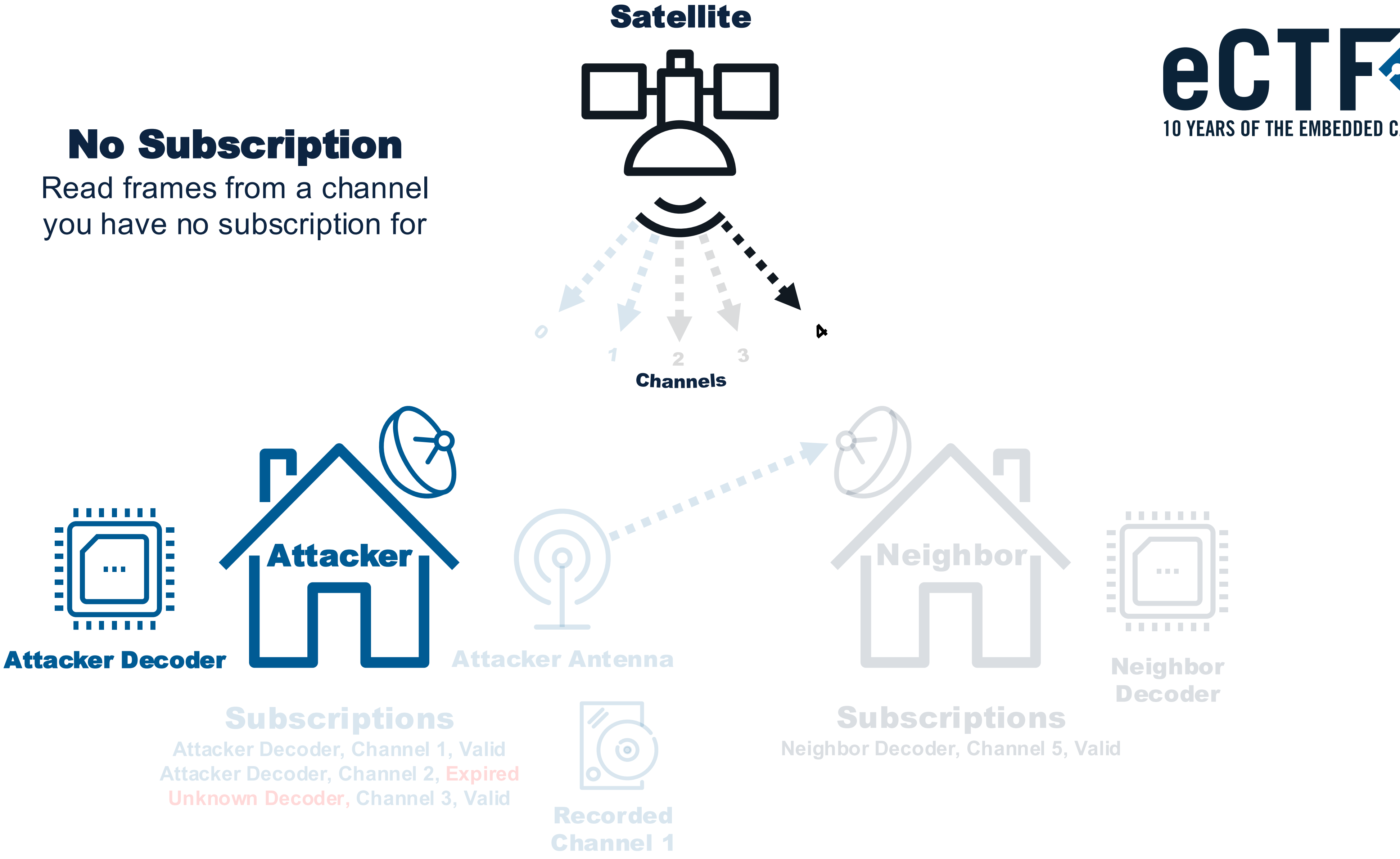
Pirated Subscription

Read frames from a channel you have a pirated subscription for


Satellite
0
1
2
3
4
Channels
Attacker
Attacker Decoder
Attacker Antenna
Neighbor
Neighbor Decoder
Recorded Channel 1
Subscriptions
Attacker Decoder, Channel 1, Valid
Attacker Decoder, Channel 2, Expired
Unknown Decoder, Channel 3, Valid
Subscriptions
Neighbor Decoder, Channel 5, Valid

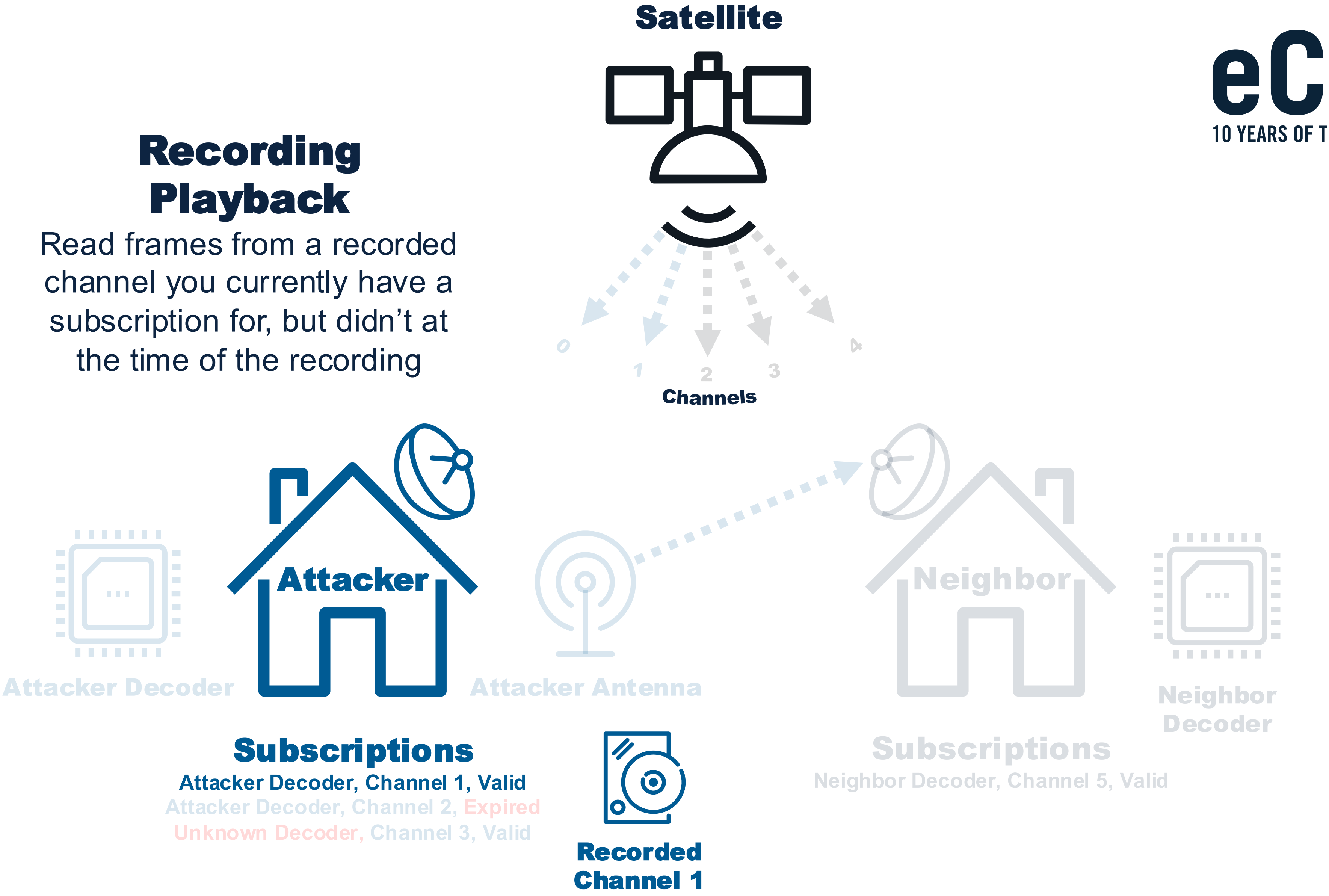
No Subscription

Read frames from a channel you have no subscription for


Satellite
0
1
2
3
4
Channels
Attacker Decoder
Attacker
Attacker Antenna
Neighbor
Neighbor Decoder
Subscriptions
Attacker Decoder, Channel 1, Valid
Attacker Decoder, Channel 2, Expired
Unknown Decoder, Channel 3, Valid
Recorded Channel 1
Subscriptions
Neighbor Decoder, Channel 5, Valid

Recording Playback

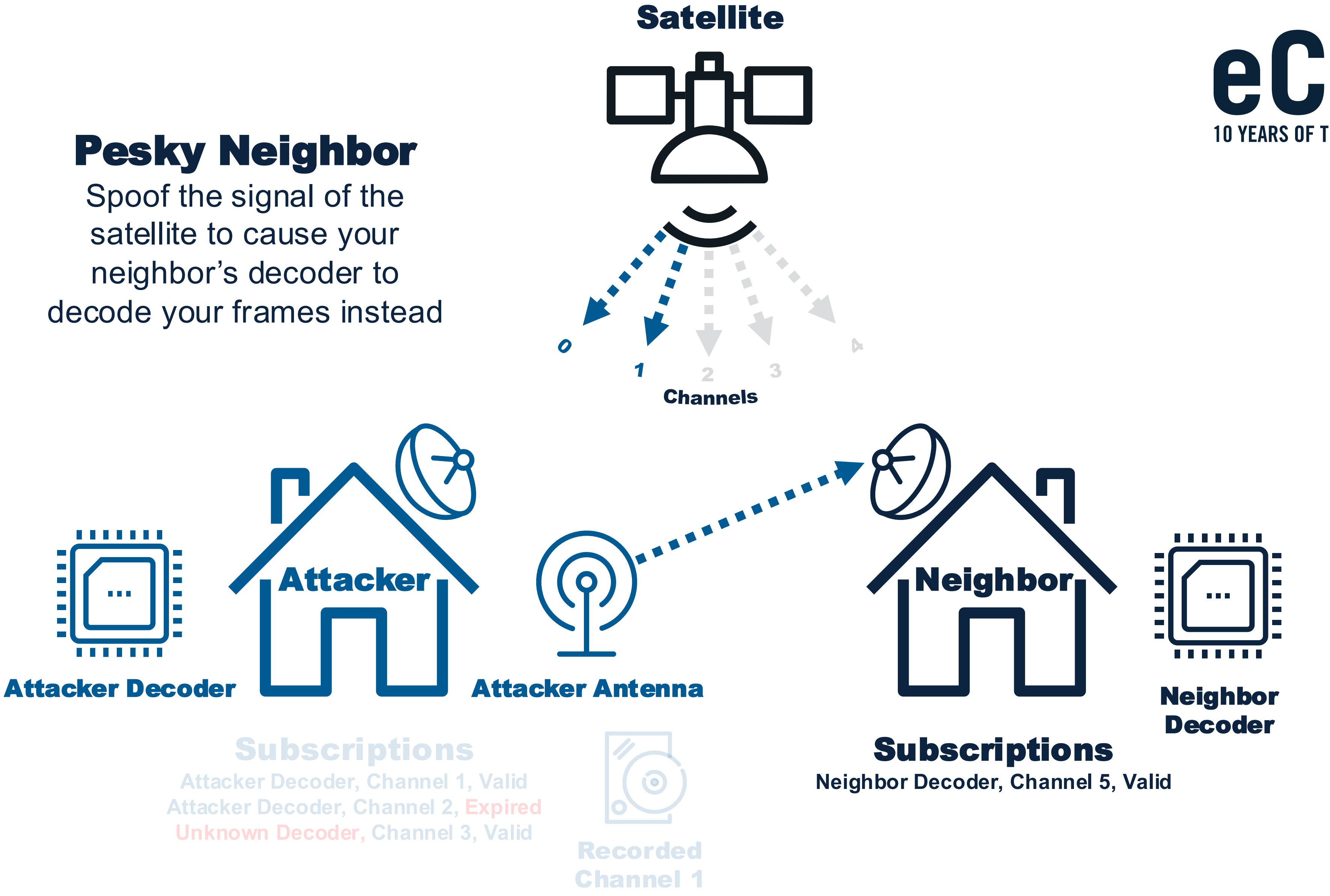
Read frames from a recorded channel you currently have a subscription for, but didn't at the time of the recording


Satellite
0
1
2
3
4
Channels
Attacker Decoder
Attacker
Attacker Antenna
Neighbor
Neighbor Decoder
Subscriptions
Attacker Decoder, Channel 1, Valid
Attacker Decoder, Channel 2, Expired
Unknown Decoder, Channel 3, Valid
Recorded Channel 1
Subscriptions
Neighbor Decoder, Channel 5, Valid

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

Pesky Neighbor

Spoof the signal of the satellite to cause your neighbor's decoder to decode your frames instead

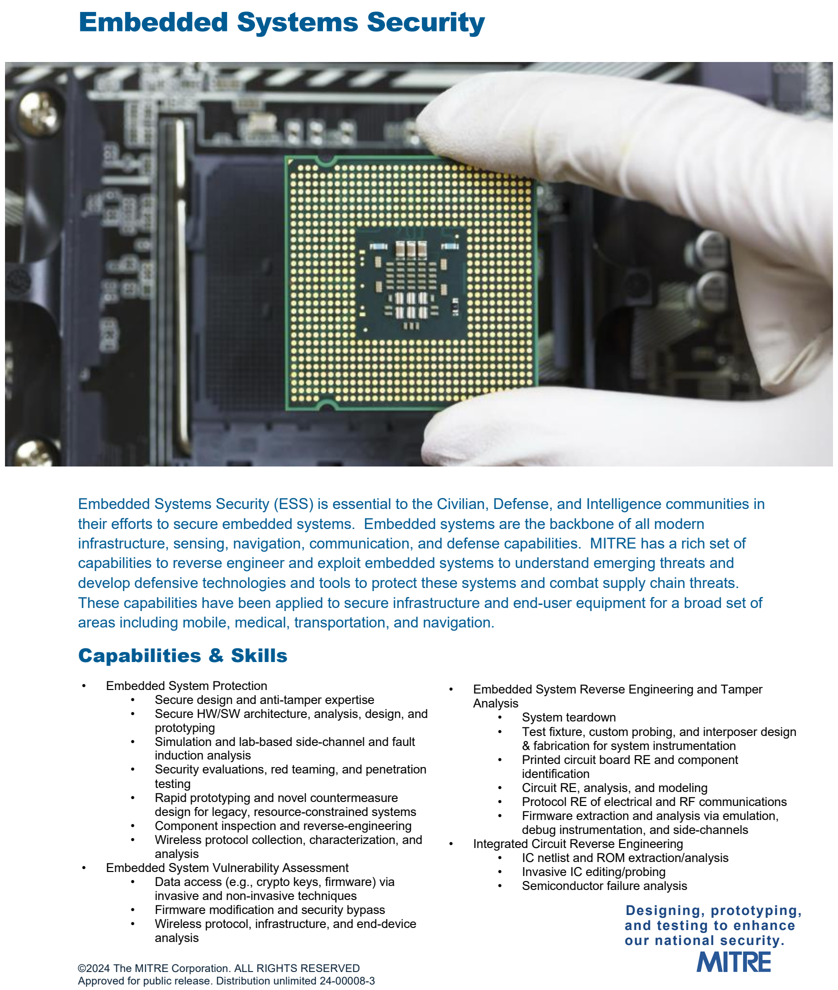


Satellite
0
1
2
3
4
Channels
Attacker
Attacker Decoder
Attacker Antenna
Neighbor
Neighbor Decoder
Subscriptions
Attacker Decoder, Channel 1, Valid
Attacker Decoder, Channel 2, Expired
Unknown Decoder, Channel 3, Valid
Recorded Channel 1
Subscriptions
Neighbor Decoder, Channel 5, Valid

MITRE

eCTF Origins

Embedded Systems Security



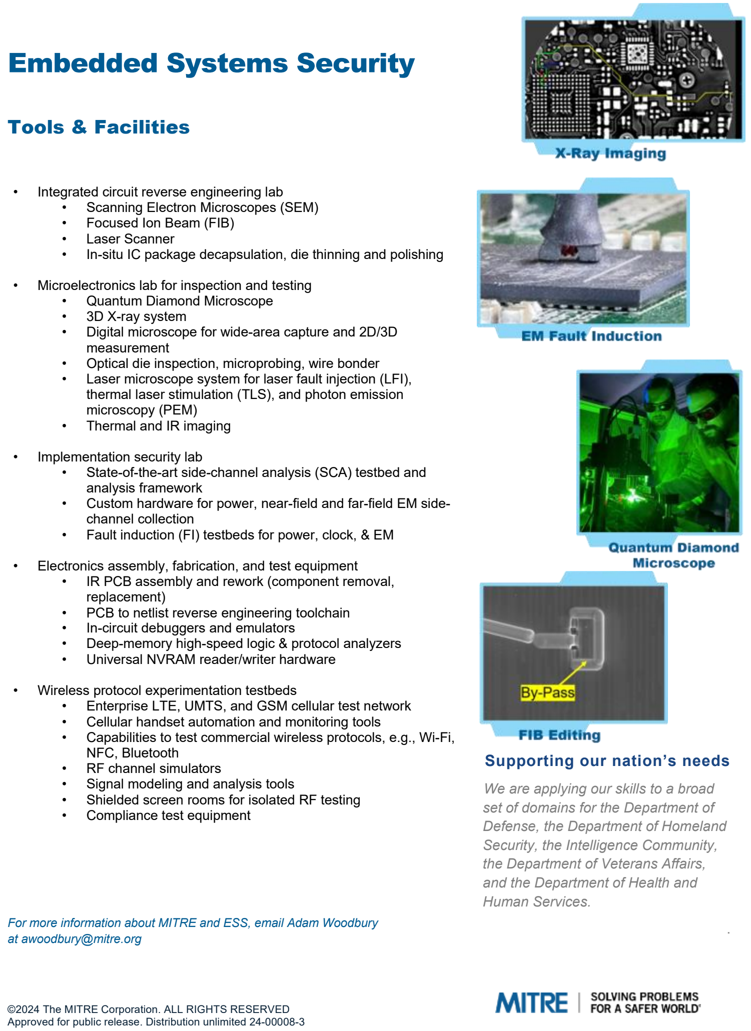
Embedded Systems Security

Embedded Systems Security (ESS) is essential to the Civilian, Defense, and Intelligence communities in their efforts to secure embedded systems. Embedded systems are the backbone of all modern infrastructure, sensing, navigation, communication, and defense capabilities. MITRE has a rich set of capabilities to reverse engineer and exploit embedded systems to understand emerging threats and develop defensive technologies and tools to protect these systems and combat supply chain threats. These capabilities have been applied to secure infrastructure and end-user equipment for a broad set of areas including mobile, medical, transportation, and navigation.

Capabilities & Skills

- Embedded System Protection
  - Secure design and anti-tamper expertise
  - Secure HW/SW architecture, analysis, design, and prototyping
  - Simulation and lab-based side-channel and fault induction analysis
  - Security evaluations, red teaming, and penetration testing
  - Rapid prototyping and novel countermeasure design for legacy, resource-constrained systems
  - Component inspection and reverse-engineering
  - Wireless protocol collection, characterization, and analysis
- Embedded System Vulnerability Assessment
  - Data access (e.g., crypto keys, firmware) via invasive and non-invasive techniques
  - Firmware modification and security bypass
  - Wireless protocol, infrastructure, and end-device analysis
- Embedded System Reverse Engineering and Tamper Analysis
  - System teardown
  - Test fixture, custom probing, and interposer design & fabrication for system instrumentation
  - Printed circuit board RE and component identification
  - Circuit RE, analysis, and modeling
  - Protocol RE of electrical and RF communications
  - Firmware extraction and analysis via emulation, debug instrumentation, and side-channels
- Integrated Circuit Reverse Engineering
  - IC netlist and ROM extraction/analysis
  - Invasive IC editing/probing
  - Semiconductor failure analysis

Designing, prototyping, and testing to enhance our national security.
MITRE



Embedded Systems Security

Tools & Facilities

- Integrated circuit reverse engineering lab
  - Scanning Electron Microscopes (SEM)
  - Focused Ion Beam (FIB)
  - Laser Scanner
  - In-situ IC package decapsulation, die thinning and polishing
- Microelectronics lab for inspection and testing
  - Quantum Diamond Microscope
  - 3D X-ray system
  - Digital microscope for wide-area capture and 2D/3D measurement
  - Optical die inspection, microprobing, wire bonder
  - Laser microscope system for laser fault injection (LFI), thermal laser stimulation (TLS), and photon emission microscopy (PEM)
  - Thermal and IR imaging
- Implementation security lab
  - State-of-the-art side-channel analysis (SCA) testbed and analysis framework
  - Custom hardware for power, near-field and far-field EM side-channel collection
  - Fault induction (FI) testbeds for power, clock, & EM
- Electronics assembly, fabrication, and test equipment
  - IR PCB assembly and rework (component removal, replacement)
  - PCB to netlist reverse engineering toolchain
  - In-circuit debuggers and emulators
  - Deep-memory high-speed logic & protocol analyzers
  - Universal NVRAM reader/writer hardware
- Wireless protocol experimentation testbeds
  - Enterprise LTE, UMTS, and GSM cellular test network
  - Cellular handset automation and monitoring tools
  - Capabilities to test commercial wireless protocols, e.g., Wi-Fi, NFC, Bluetooth
  - RF channel simulators
  - Signal modeling and analysis tools
  - Shielded screen rooms for isolated RF testing
  - Compliance test equipment

X-Ray Imaging

EM Fault Induction

Quantum Diamond Microscope

FIB Editing

Supporting our nation's needs

We are applying our skills to a broad set of domains for the Department of Defense, the Department of Homeland Security, the Intelligence Community, the Department of Veterans Affairs, and the Department of Health and Human Services.

For more information about MITRE and ESS, email Adam Woodbury at awoodbury@mitre.org

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD

eCTF Origins: Summer 2015

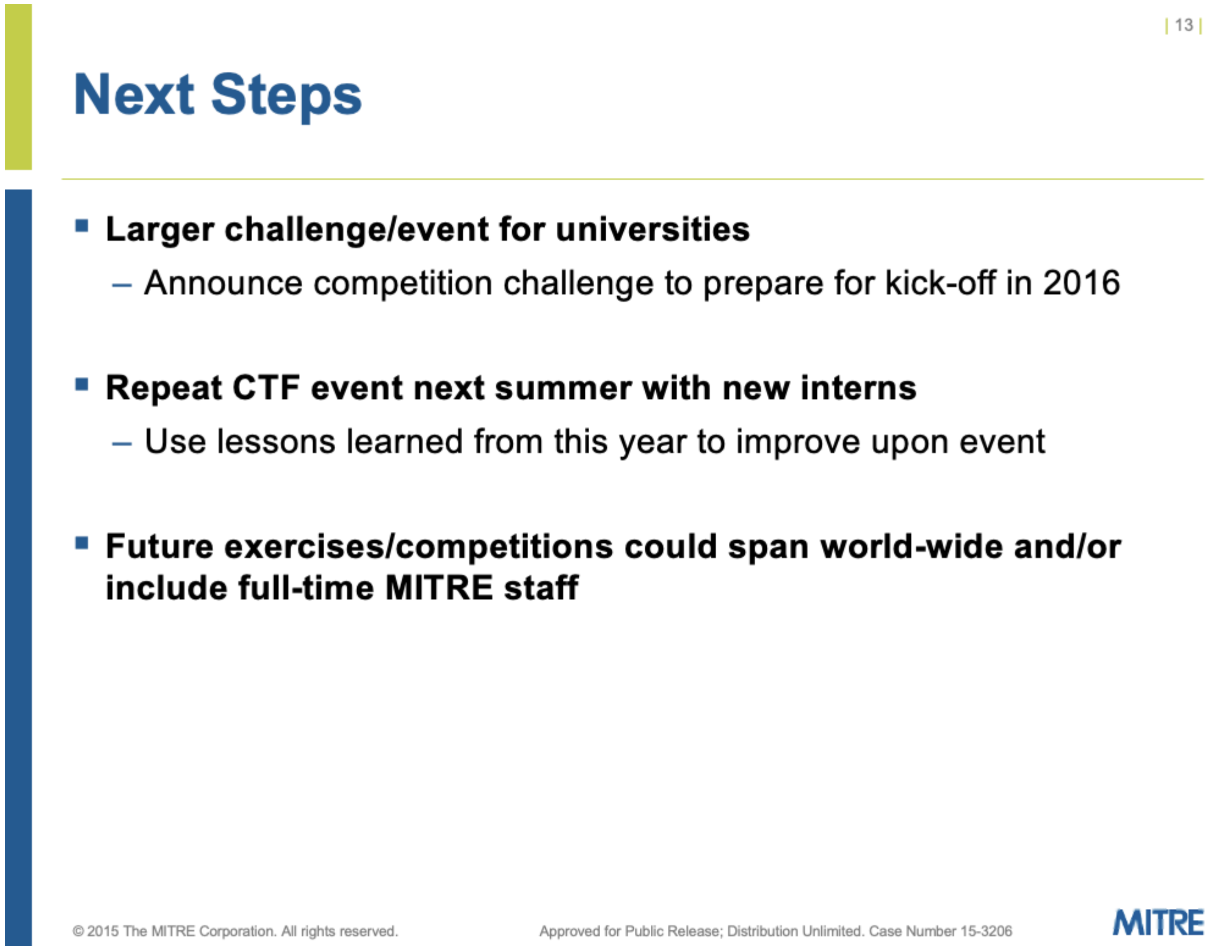
MITRE | Our Work

September 2015

MITRE's Summer Cyber Competition Teaches Interns to Think Like Adversaries

Even the best schools have a hard time keeping up with the rapidly changing world of cybersecurity. When MITRE staff detected a gap in cyber training for embedded systems, they challenged a group of interns to learn new skills in a head-to-head contest.

© 2015 The MITRE Corporation. All Rights Reserved.
Approved for Public Release; Distribution Unlimited, Case Number 15-2636 8/20/2015



| 13 |

Next Steps

- **Larger challenge/event for universities**
  - Announce competition challenge to prepare for kick-off in 2016
- **Repeat CTF event next summer with new interns**
  - Use lessons learned from this year to improve upon event
- **Future exercises/competitions could span world-wide and/or include full-time MITRE staff**

© 2015 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case Number 15-3206 MITRE

eCTF Origins: Collegiate 2016

LVX VERITAS VIRTVS
Northeastern University

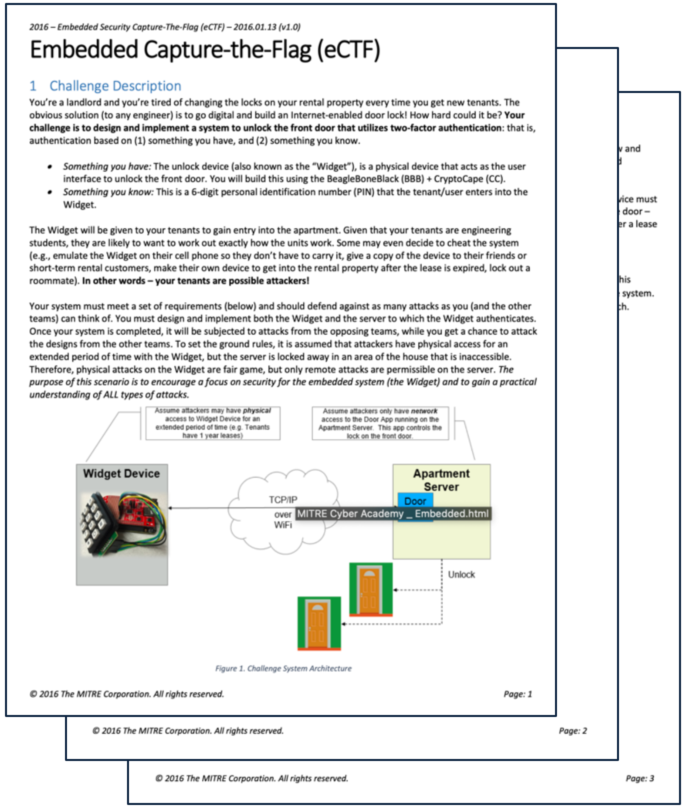
Tufts UNIVERSITY

UMass

WPI

eCTF Origins: Collegiate 2016

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



2016 – Embedded Security Capture-The-Flag (eCTF) – 2016.01.13 (v1.0)

Embedded Capture-the-Flag (eCTF)

1 Challenge Description

You're a landlord and you're tired of changing the locks on your rental property every time you get new tenants. The obvious solution (to any engineer) is to go digital and build an Internet-enabled door lock! How hard could it be? **Your challenge is to design and implement a system to unlock the front door that utilizes two-factor authentication**: that is, authentication based on (1) something you have, and (2) something you know.

- *Something you have:* The unlock device (also known as the "Widget"), is a physical device that acts as the user interface to unlock the front door. You will build this using the BeagleBoneBlack (BBB) + CryptoCape (CC).
- *Something you know:* This is a 6-digit personal identification number (PIN) that the tenant/user enters into the Widget.

The Widget will be given to your tenants to gain entry into the apartment. Given that your tenants are engineering students, they are likely to want to work out exactly how the units work. Some may even decide to cheat the system (e.g., emulate the Widget on their cell phone so they don't have to carry it, give a copy of the device to their friends or short-term rental customers, make their own device to get into the rental property after the lease is expired, lock out a roommate). **In other words – your tenants are possible attackers!**

Your system must meet a set of requirements (below) and should defend against as many attacks as you (and the other teams) can think of. You must design and implement both the Widget and the server to which the Widget authenticates. Once your system is completed, it will be subjected to attacks from the opposing teams, while you get a chance to attack the designs from the other teams. To set the ground rules, it is assumed that attackers have physical access for an extended period of time with the Widget, but the server is locked away in an area of the house that is inaccessible. Therefore, physical attacks on the Widget are fair game, but only remote attacks are permissible on the server. *The purpose of this scenario is to encourage a focus on security for the embedded system (the Widget) and to gain a practical understanding of ALL types of attacks.*

Assume attackers may have *physical* access to Widget Device for an extended period of time (e.g. Tenants have 1 year leases)
Assume attackers only have *network* access to the Door App running on the Apartment Server. This app controls the lock on the front door.
Widget Device
TCP/IP over WiFi
Apartment Server
Door
MITRE Cyber Academy _ Embedded.html
Unlock

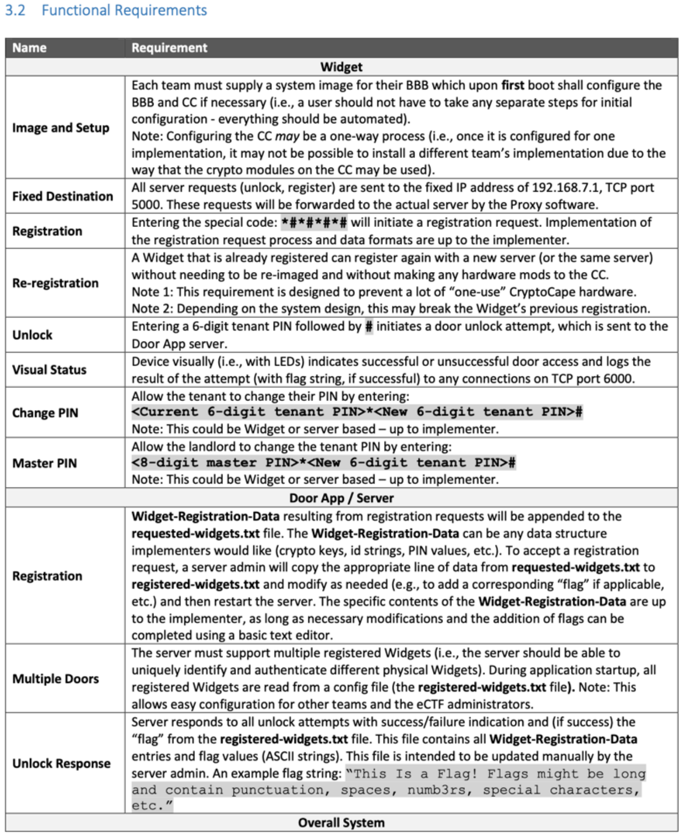
Figure 1. Challenge System Architecture

v and
ice must
door –
r a lease
his
system.
h.

© 2016 The MITRE Corporation. All rights reserved. Page: 1

© 2016 The MITRE Corporation. All rights reserved. Page: 2

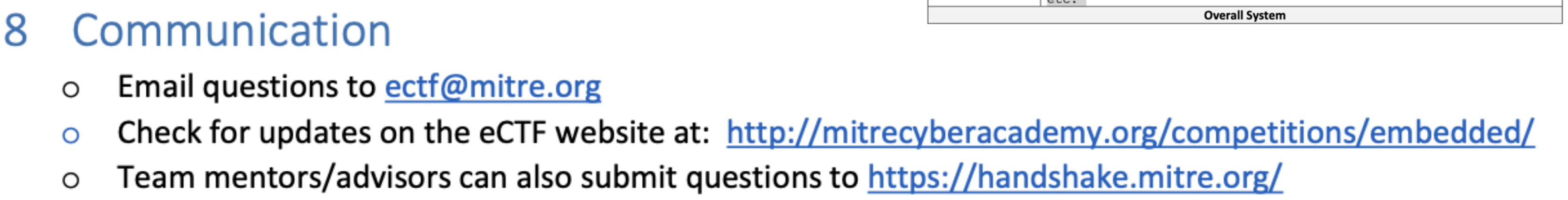
© 2016 The MITRE Corporation. All rights reserved. Page: 3



3.2 Functional Requirements

Name	Requirement
	Widget
Image and Setup	Each team must supply a system image for their BBB which upon **first** boot shall configure the BBB and CC if necessary (i.e., a user should not have to take any separate steps for initial configuration - everything should be automated). Note: Configuring the CC *may* be a one-way process (i.e., once it is configured for one implementation, it may not be possible to install a different team's implementation due to the way that the crypto modules on the CC may be used).
Fixed Destination	All server requests (unlock, register) are sent to the fixed IP address of 192.168.7.1, TCP port 5000. These requests will be forwarded to the actual server by the Proxy software.
Registration	Entering the special code: `*#*#*#*#` will initiate a registration request. Implementation of the registration request process and data formats are up to the implementer.
Re-registration	A Widget that is already registered can register again with a new server (or the same server) without needing to be re-imaged and without making any hardware mods to the CC. Note 1: This requirement is designed to prevent a lot of "one-use" CryptoCape hardware. Note 2: Depending on the system design, this may break the Widget's previous registration.
Unlock	Entering a 6-digit tenant PIN followed by `#` initiates a door unlock attempt, which is sent to the Door App server.
Visual Status	Device visually (i.e., with LEDs) indicates successful or unsuccessful door access and logs the result of the attempt (with flag string, if successful) to any connections on TCP port 6000.
Change PIN	Allow the tenant to change their PIN by entering: **`<Current 6-digit tenant PIN>*<New 6-digit tenant PIN>#`** Note: This could be Widget or server based – up to implementer.
Master PIN	Allow the landlord to change the tenant PIN by entering: **`<8-digit master PIN>*<New 6-digit tenant PIN>#`** Note: This could be Widget or server based – up to implementer.
	Door App / Server
Registration	**Widget-Registration-Data** resulting from registration requests will be appended to the **requested-widgets.txt** file. The **Widget-Registration-Data** can be any data structure implementers would like (crypto keys, id strings, PIN values, etc.). To accept a registration request, a server admin will copy the appropriate line of data from **requested-widgets.txt** to **registered-widgets.txt** and modify as needed (e.g., to add a corresponding "flag" if applicable, etc.) and then restart the server. The specific contents of the **Widget-Registration-Data** are up to the implementer, as long as necessary modifications and the addition of flags can be completed using a basic text editor.
Multiple Doors	The server must support multiple registered Widgets (i.e., the server should be able to uniquely identify and authenticate different physical Widgets). During application startup, all registered Widgets are read from a config file (the **registered-widgets.txt** file**).** Note: This allows easy configuration for other teams and the eCTF administrators.
Unlock Response	Server responds to all unlock attempts with success/failure indication and (if success) the "flag" from the **registered-widgets.txt** file. This file contains all **Widget-Registration-Data** entries and flag values (ASCII strings). This file is intended to be updated manually by the server admin. An example flag string: `"This Is a Flag! Flags might be long and contain punctuation, spaces, numb3rs, special characters, etc."`
	Overall System

Name	Point Value
Master PIN	300 / 150
Shoulder Surfing	350
New Neighbor	450
Stolen Widget	250
Cloning	200
Permanent Access	300



8 Communication

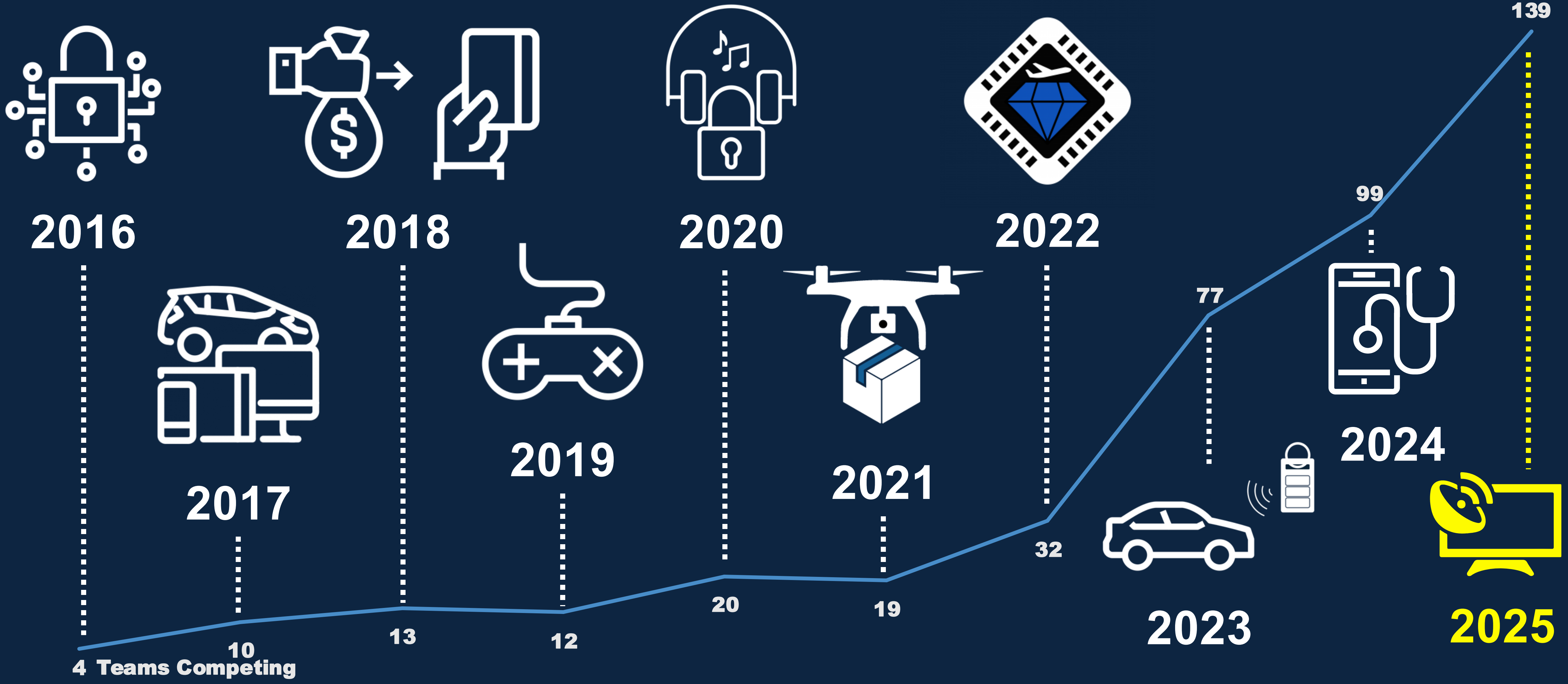
- Email questions to ectf@mitre.org
- Check for updates on the eCTF website at: http://mitrecyberacademy.org/competitions/embedded/
- Team mentors/advisors can also submit questions to https://handshake.mitre.org/

MITRE

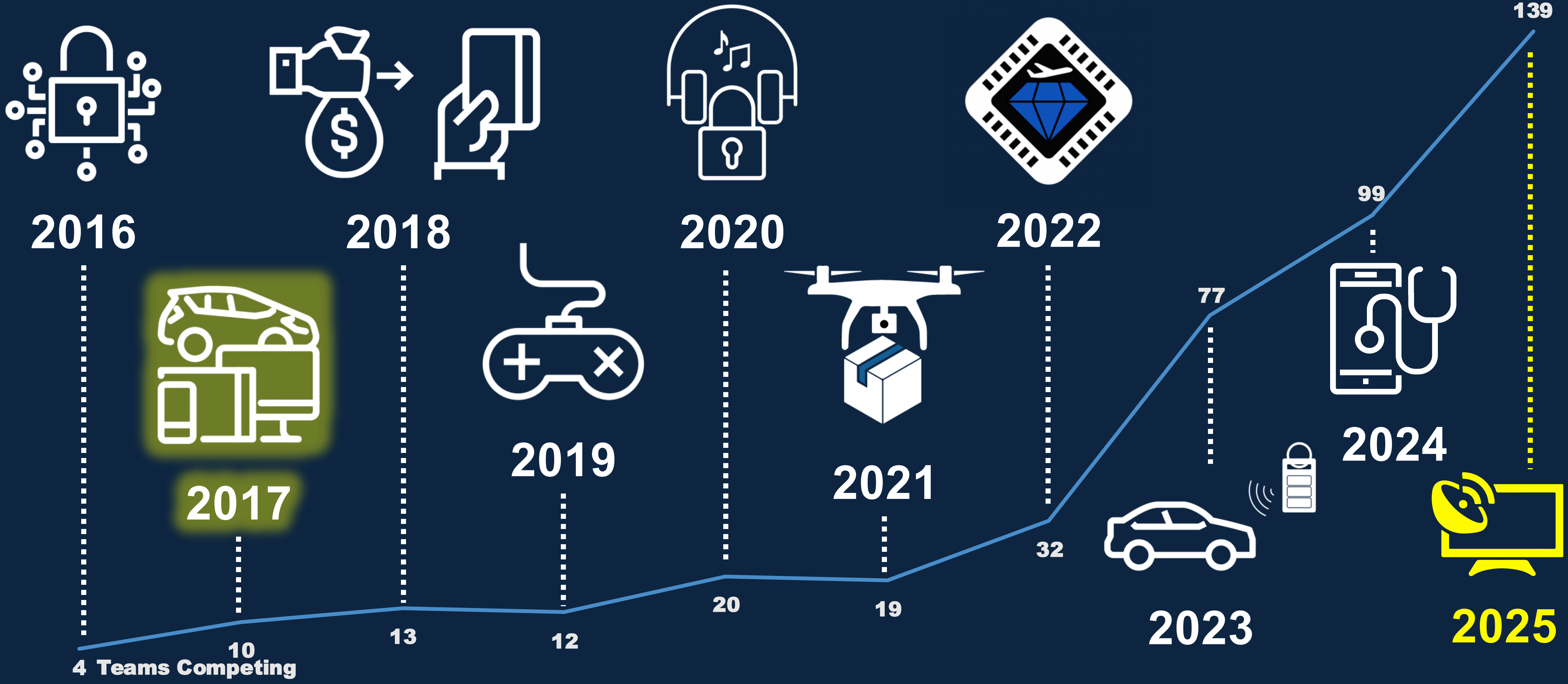
eCTF Impact

MITRE

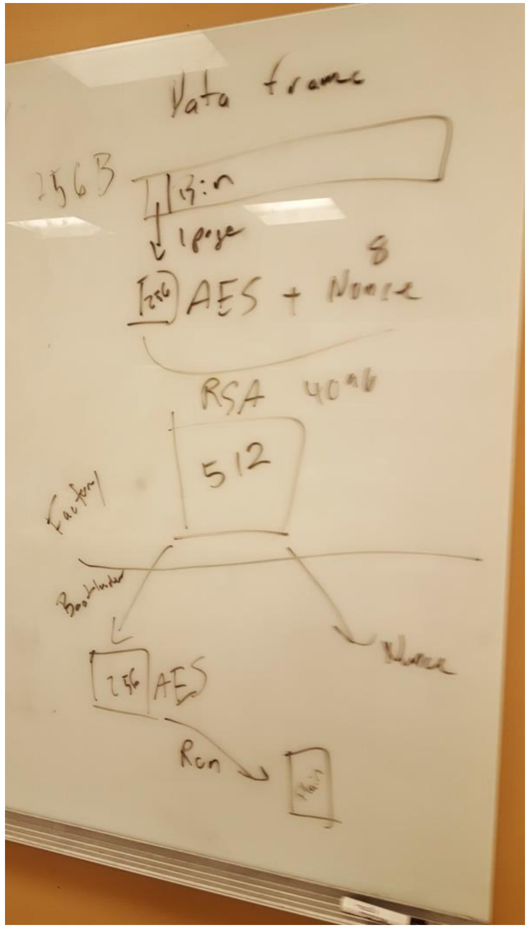
Growth of the eCTF

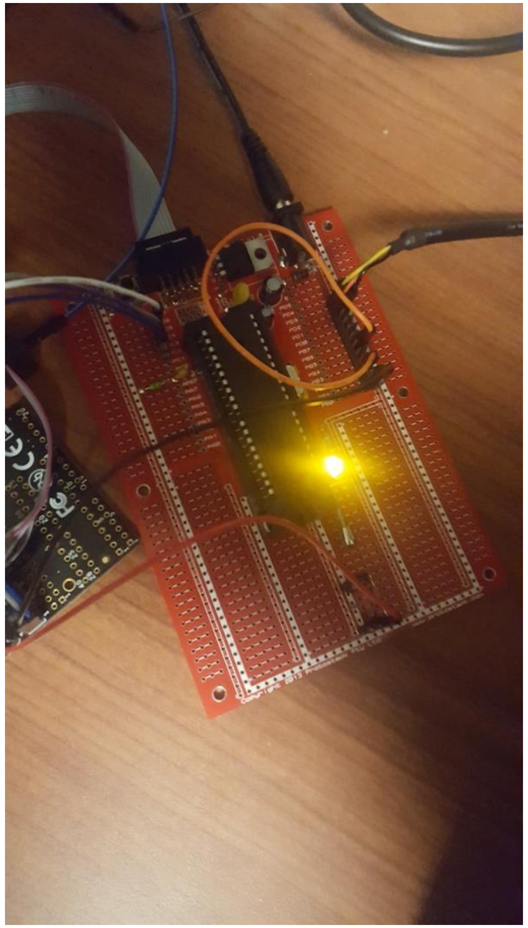


Growth of the eCTF



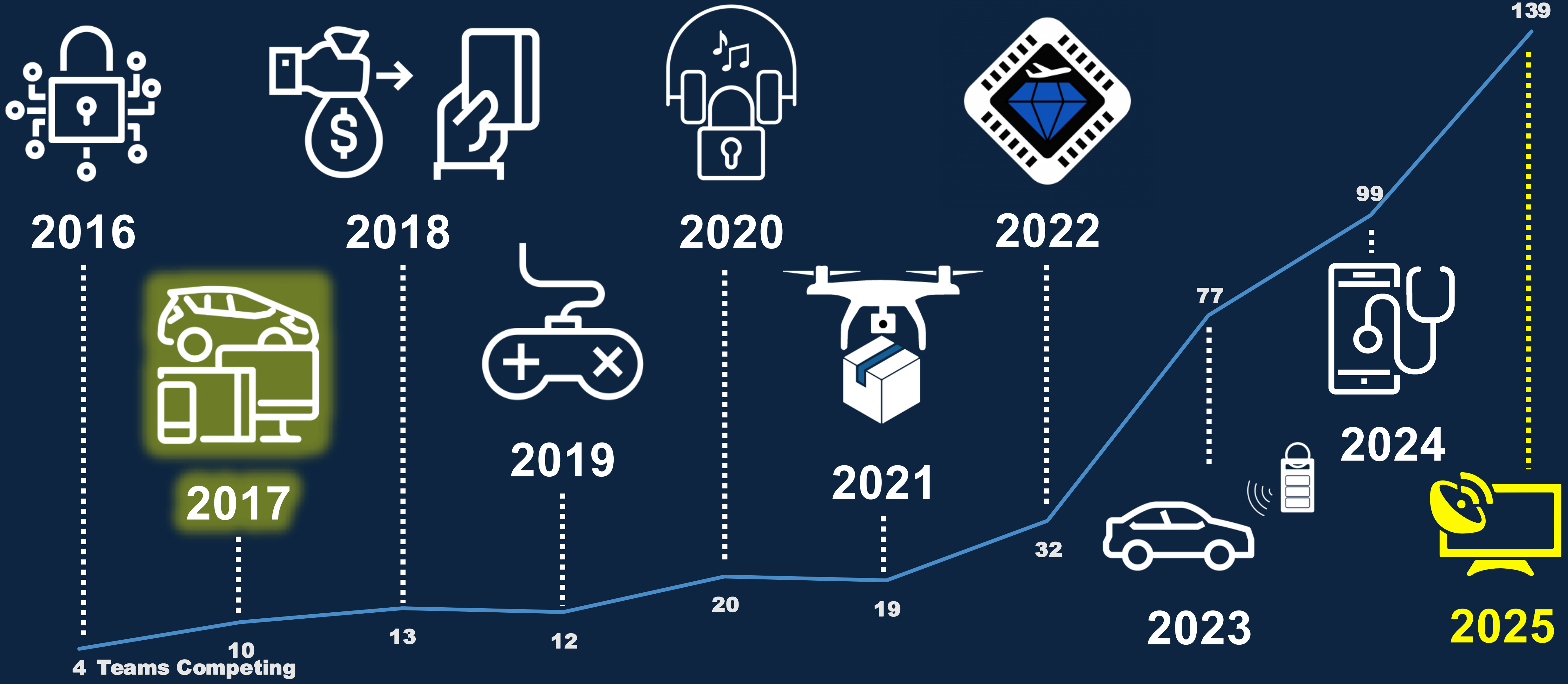
2017 eCTF





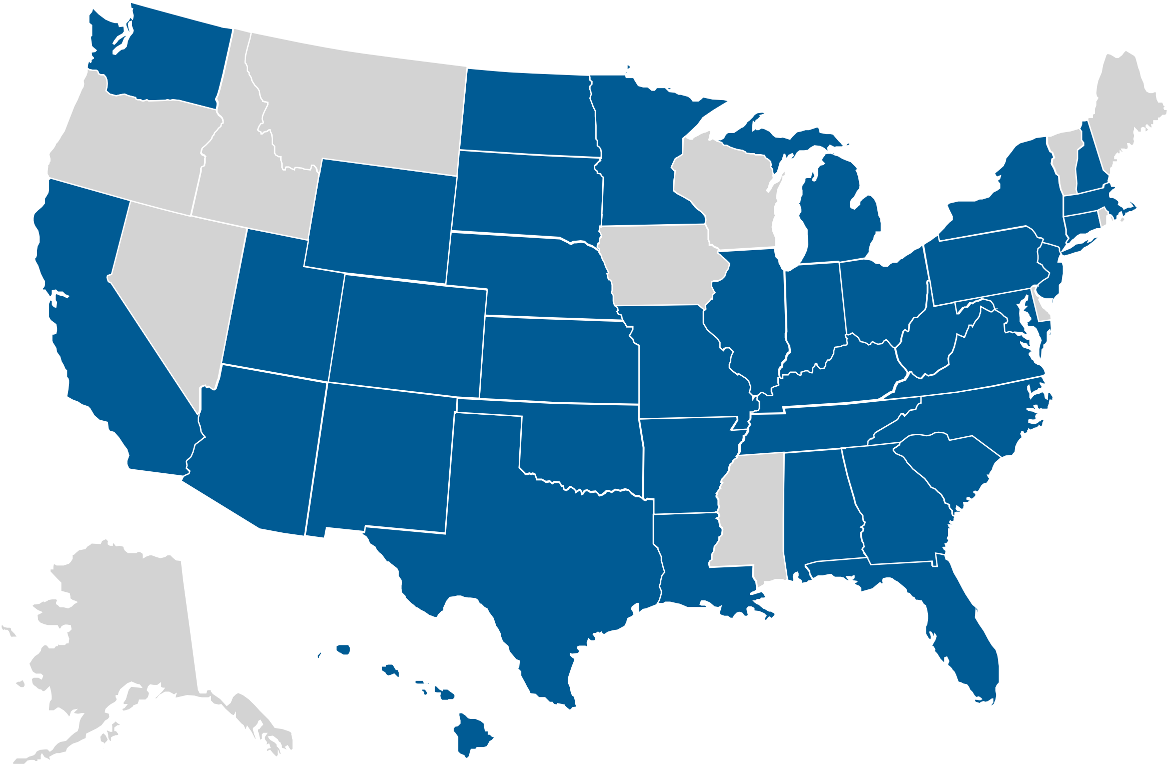
Tufts bootloader has been compiled and packaged as intel hex

Growth of the eCTF



MITRE

Nationwide Competition



Over 10 years we've reached:

38 States + DC

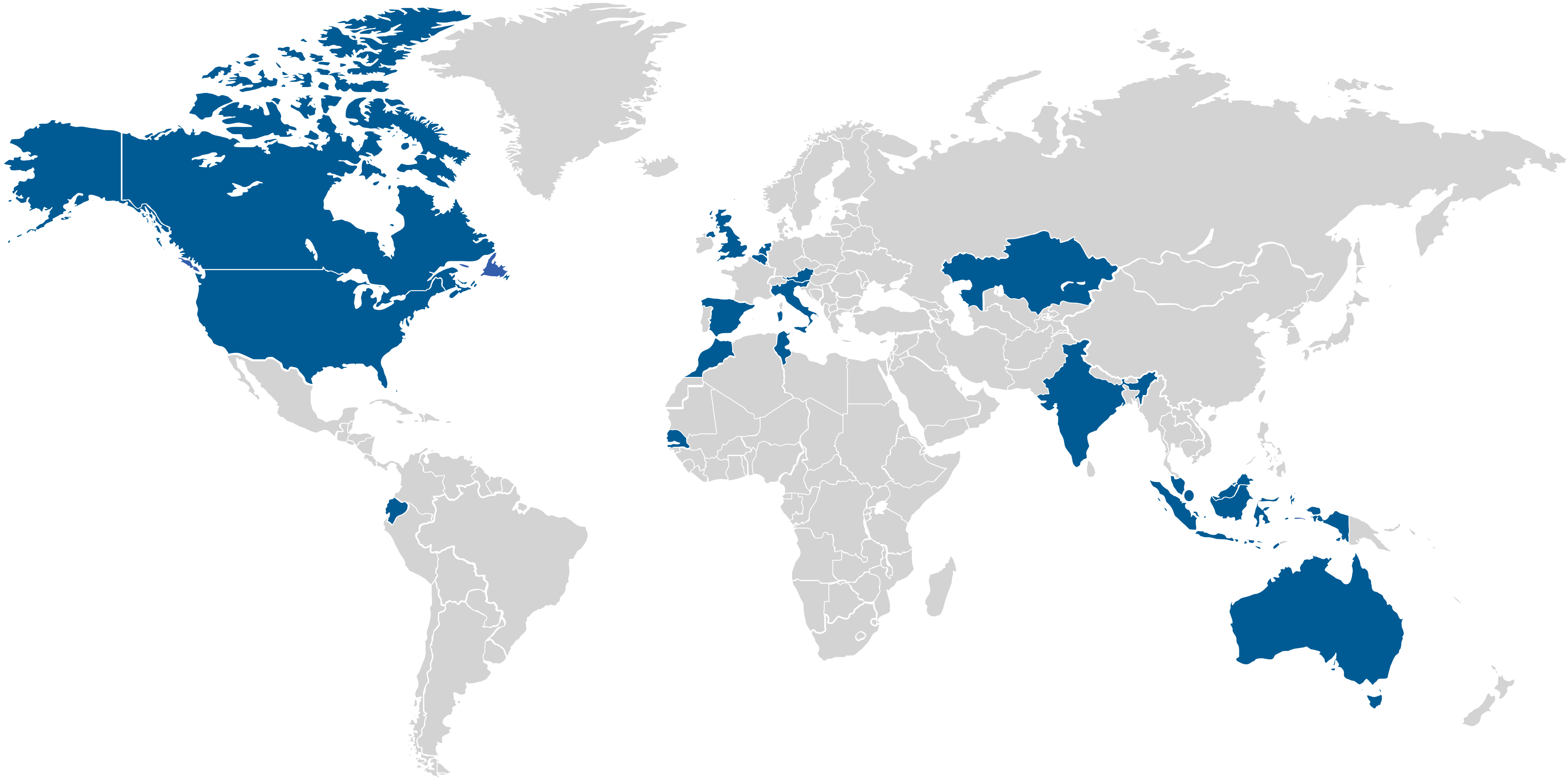
180 Schools

3000+ Students

And counting…

MITRE

International Representation



"This CTF… motivated me to dive in deeper and work that much harder to get better as an engineer. The MITRE staff was AMAZING! Thank you for this opportunity."

100%

Learned more about embedded system security

93%

Enjoyed participating in the eCTF

"This competition exposed an entirely new side of cybersecurity to me as a Computer Science major… [It] was a great learning experience and got me interested in lower-level security"

Lessons Learned

Competition Motivation

Attract students to embedded security

Increase awareness of domain

Teach mistakes before they become CVEs

Structural Goals

Quality

Best educational outcomes

Enjoyable experience

Scalability

Maximize participation

Decrease resource requirement

Accessibility

Lower barrier to entry

Increase success rate

Lessons Learned Overview

Organizational

Scoring

Hardware

Testing

Team Support

Financial

Organizational Design

Institutional Support

Leadership Chain

Deliver value for their portfolio

IR&D Program

Develop novel programs and data

Recruitment

Identify and attract top talent

University Relations

Connect with university labs

Sponsor-Facing Programs

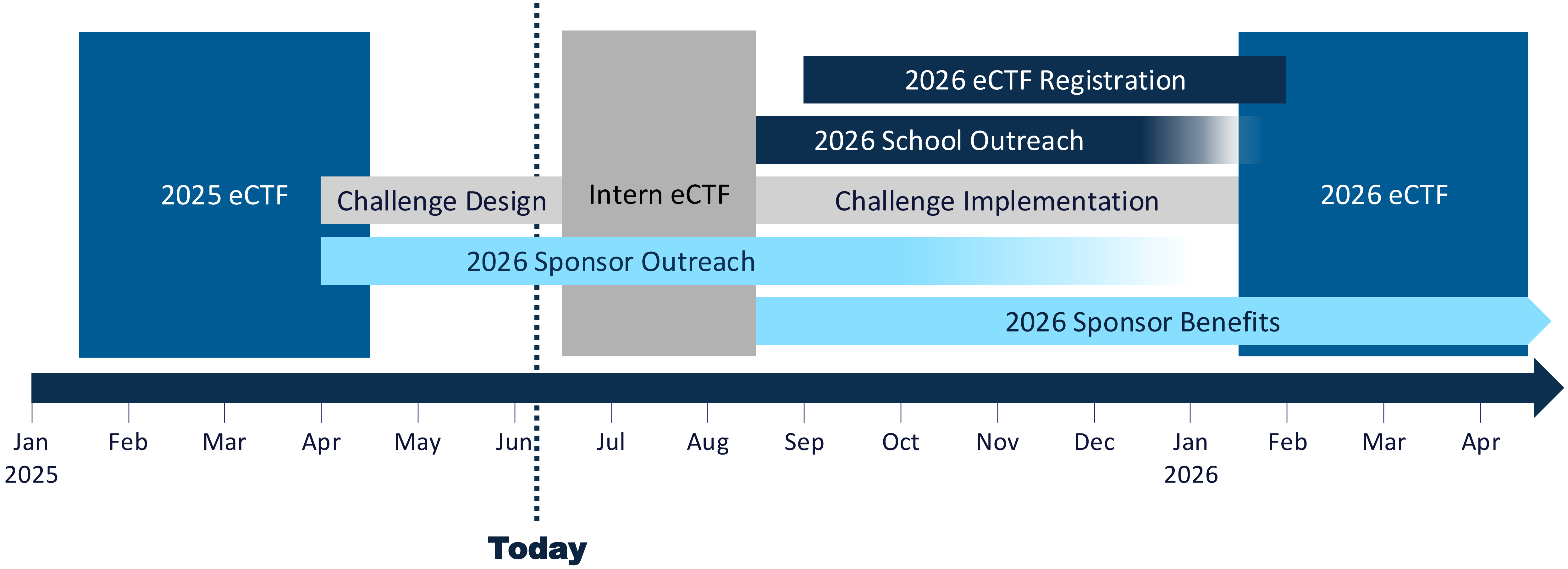
Battle-test new concepts

STEM Outreach

Train the next generation of engineers

eCTF Timeline

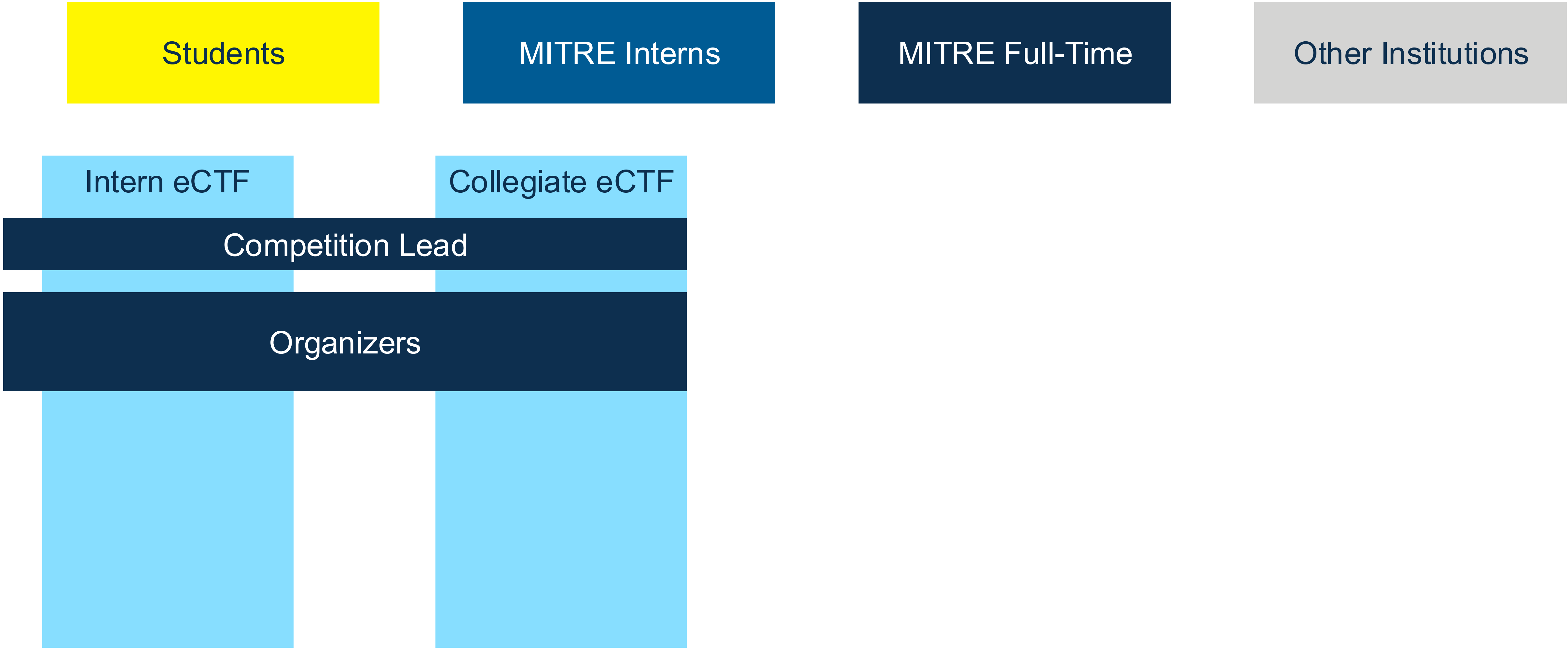
eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



MITRE

eCTF Pipeline

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



Students

MITRE Interns

MITRE Full-Time

Other Institutions

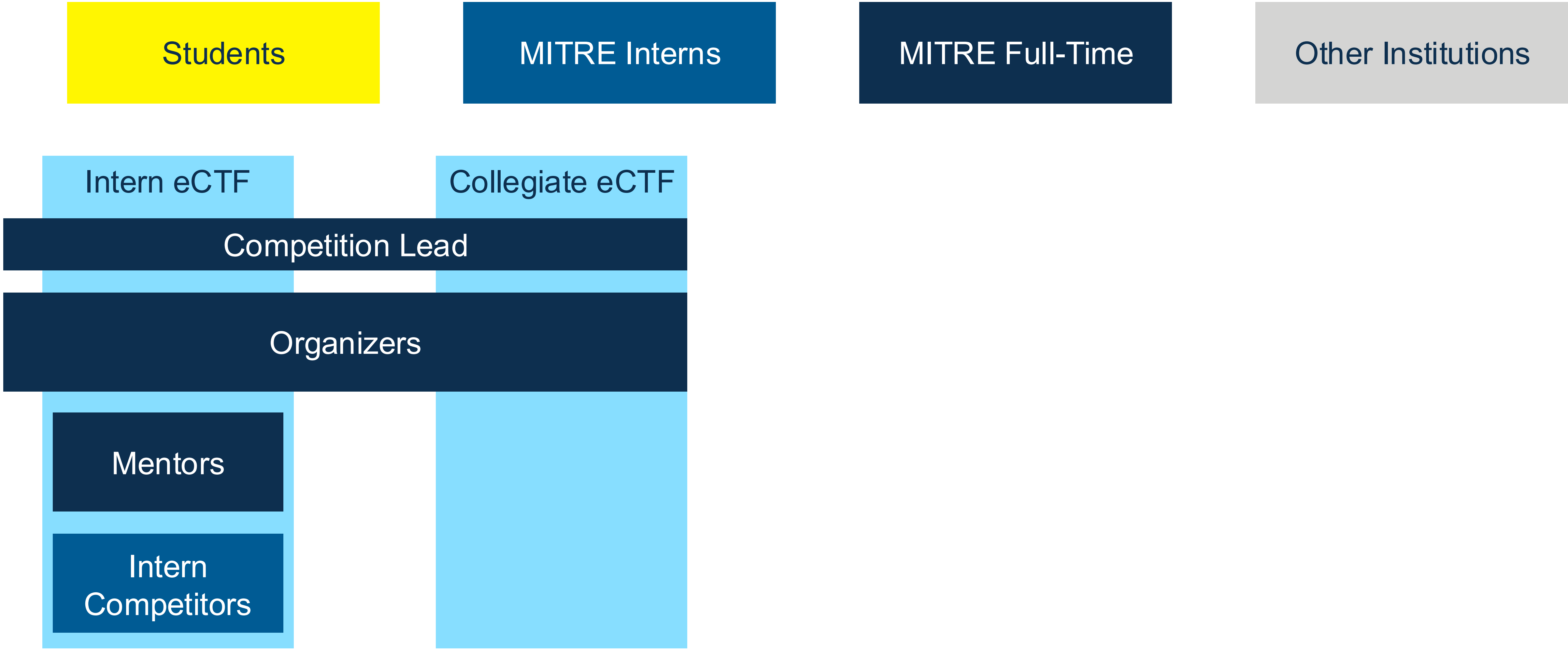
Intern eCTF

Collegiate eCTF

Competition Lead

Organizers

eCTF Pipeline



Students

MITRE Interns

MITRE Full-Time

Other Institutions

Intern eCTF

Collegiate eCTF

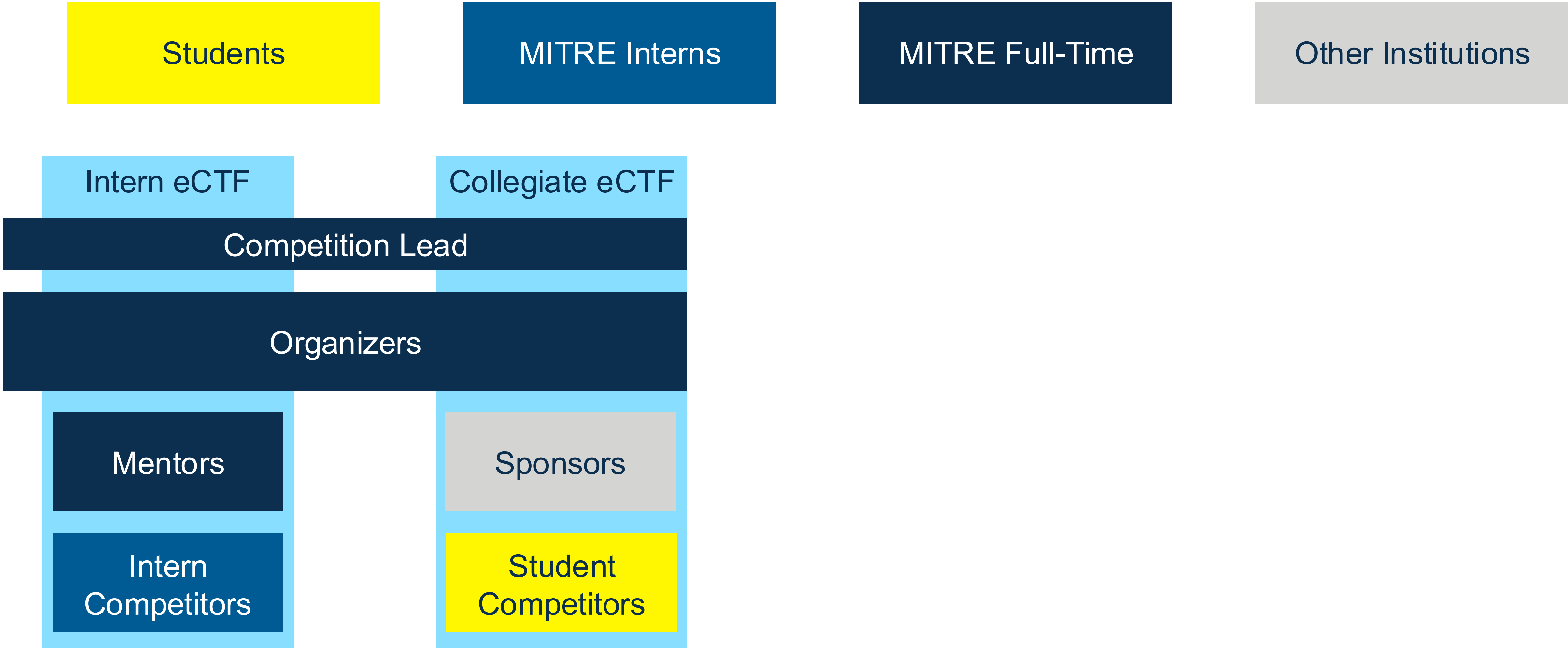
Competition Lead

Organizers

Mentors

Intern Competitors

eCTF Pipeline



Students

MITRE Interns

MITRE Full-Time

Other Institutions

Intern eCTF

Collegiate eCTF

Competition Lead

Organizers

Mentors

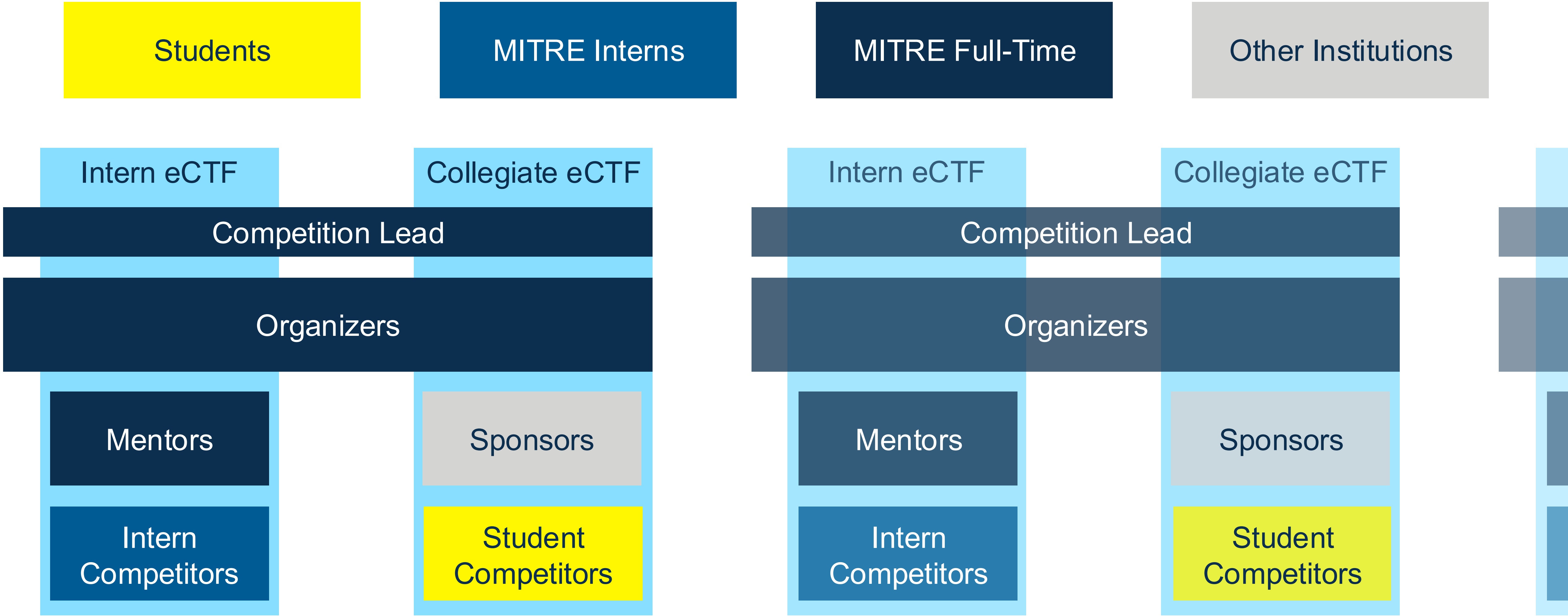
Sponsors

Intern Competitors

Student Competitors

eCTF Pipeline

eCTF10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



Students

MITRE Interns

MITRE Full-Time

Other Institutions

Intern eCTF

Collegiate eCTF

Competition Lead

Organizers

Mentors

Sponsors

Intern Competitors

Student Competitors

Intern eCTF

Collegiate eCTF

Competition Lead

Organizers

Mentors

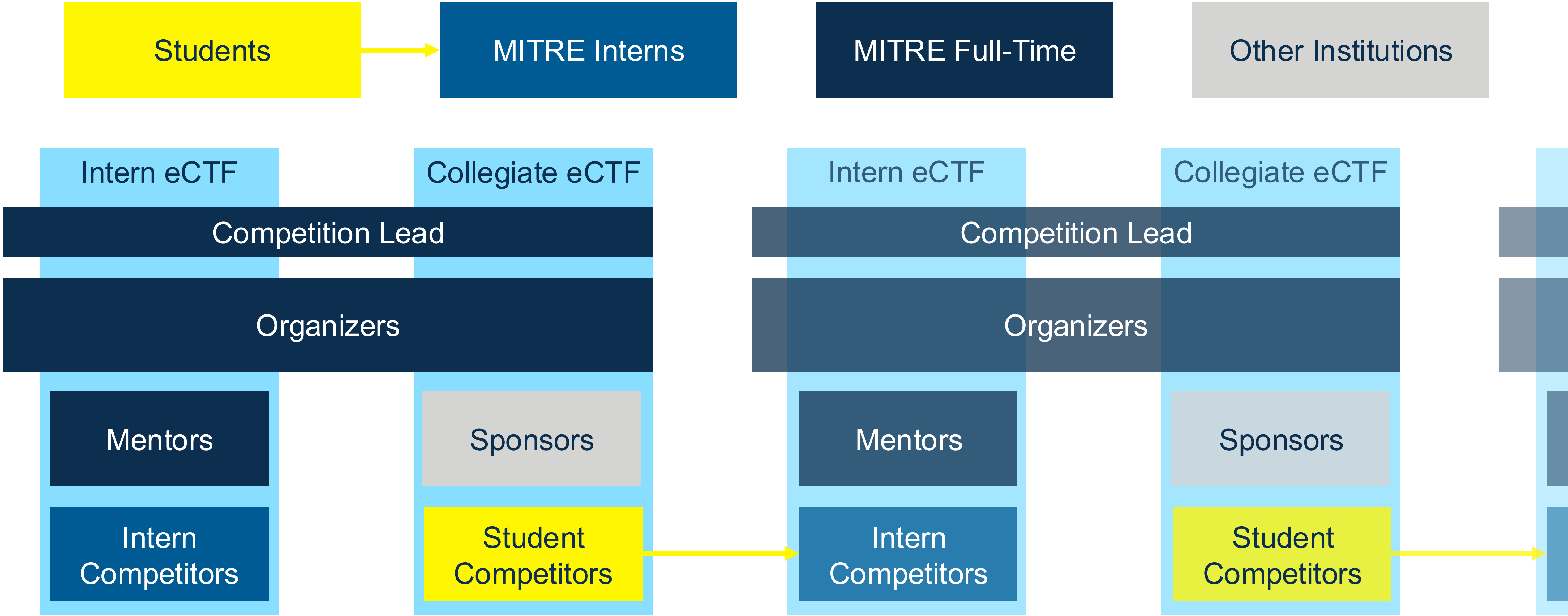
Sponsors

Intern Competitors

Student Competitors

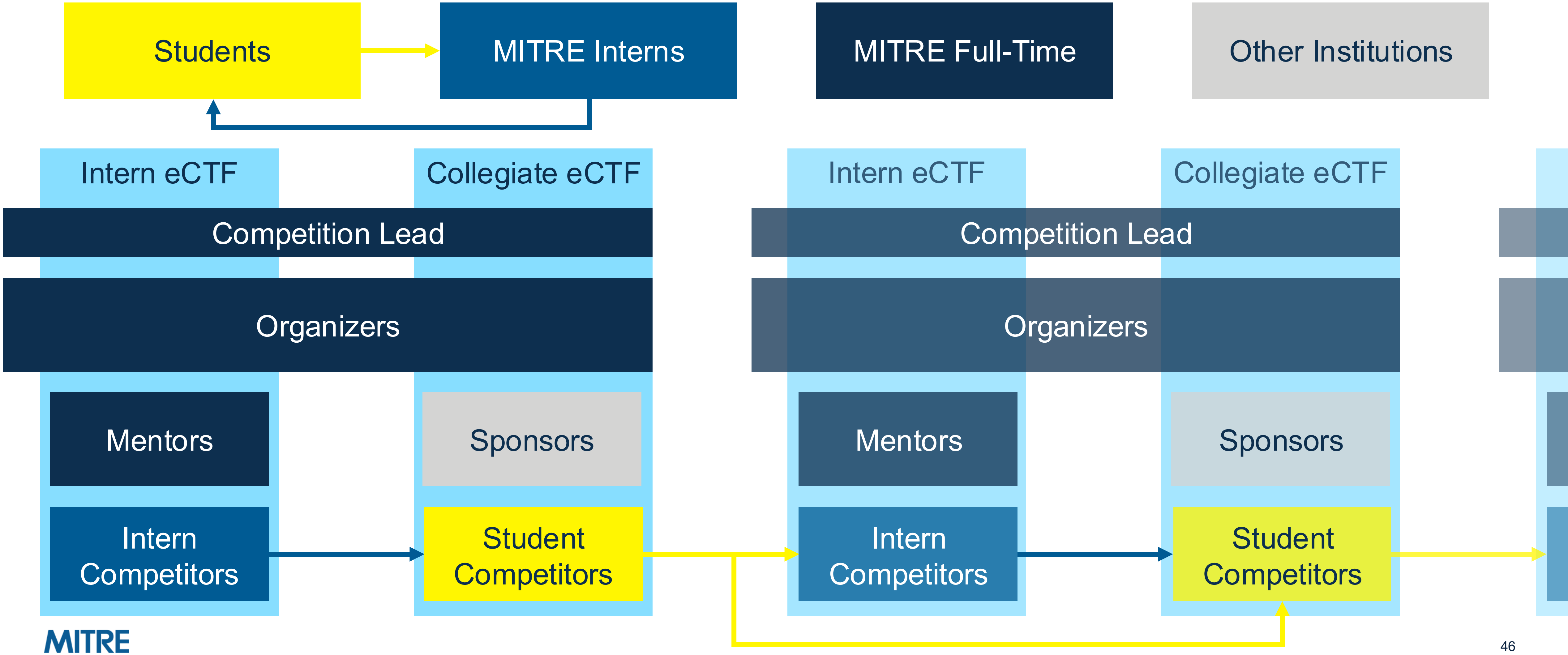
MITRE

Competitors Hired as Interns



Interns Bring eCTF Back to School

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



Students
MITRE Interns
MITRE Full-Time
Other Institutions

Intern eCTF
Collegiate eCTF
Intern eCTF
Collegiate eCTF

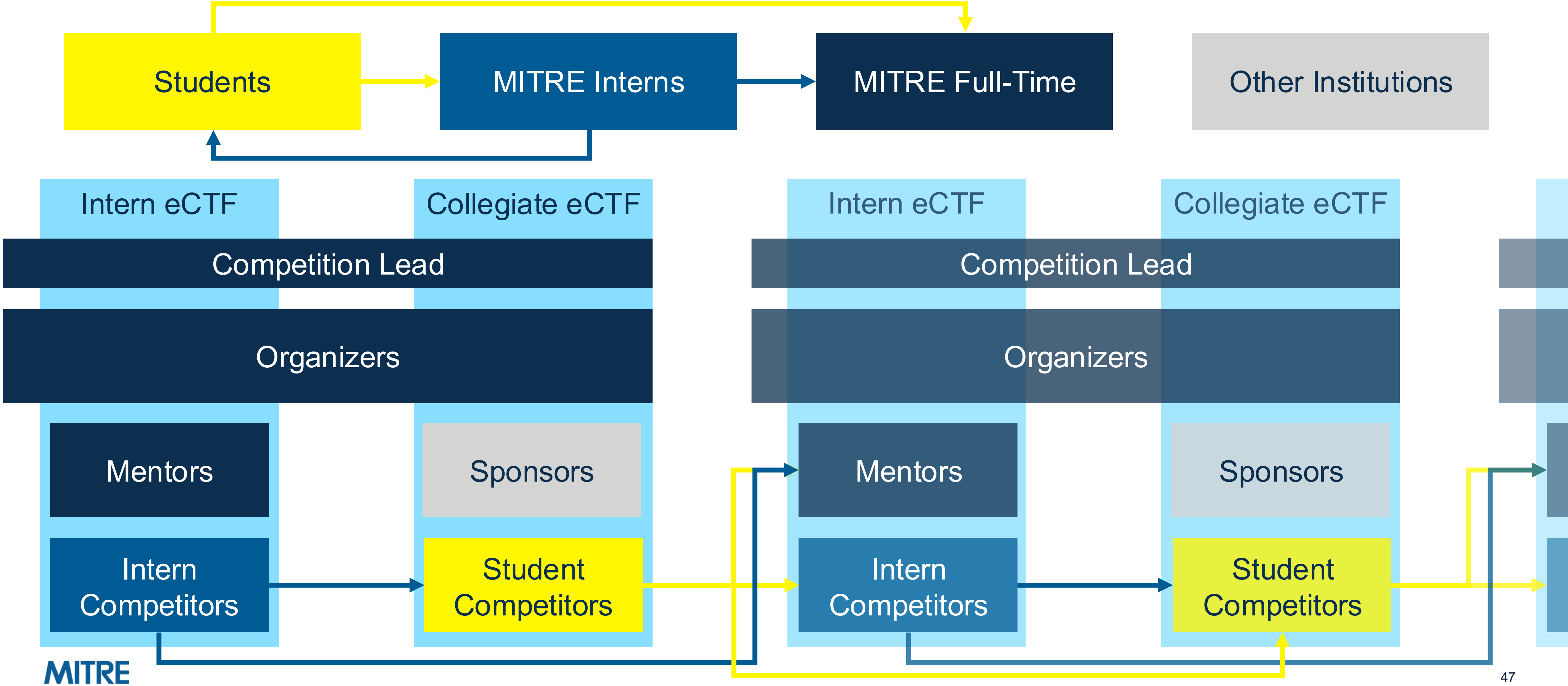
Competition Lead
Organizers
Mentors
Sponsors
Intern Competitors
Student Competitors

Competition Lead
Organizers
Mentors
Sponsors
Intern Competitors
Student Competitors

MITRE

Competitors and Interns Hired Full-Time

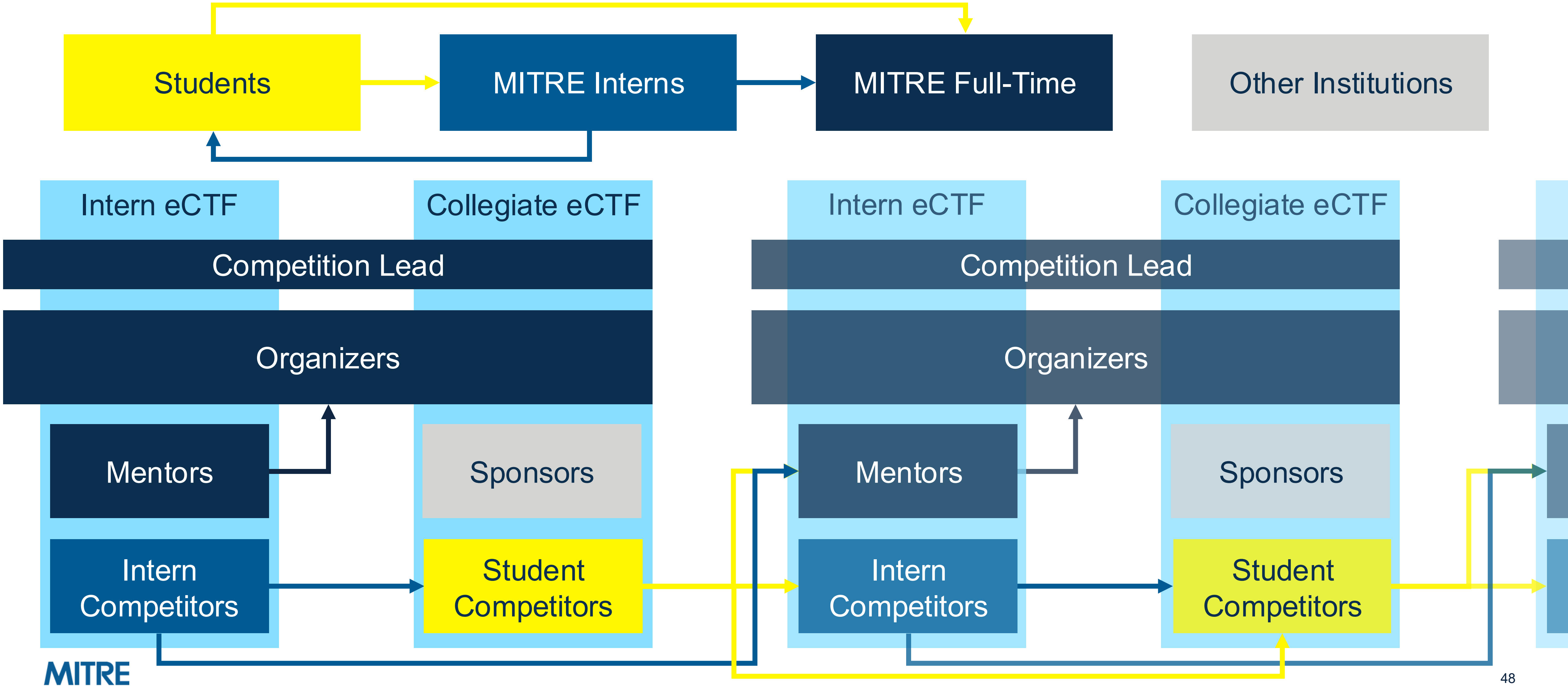
eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



MITRE

Mentors Become Organizers

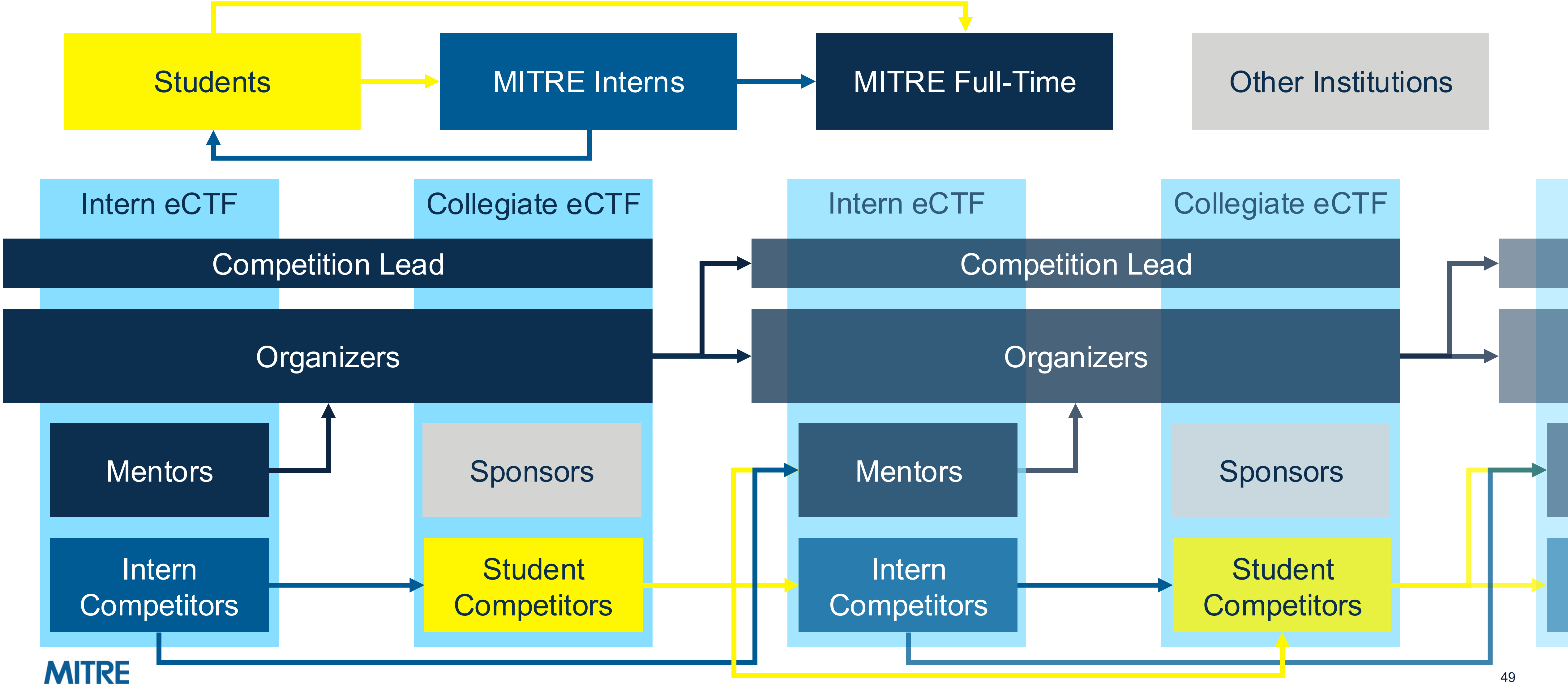
eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



MITRE

Organizers Become Competition Lead

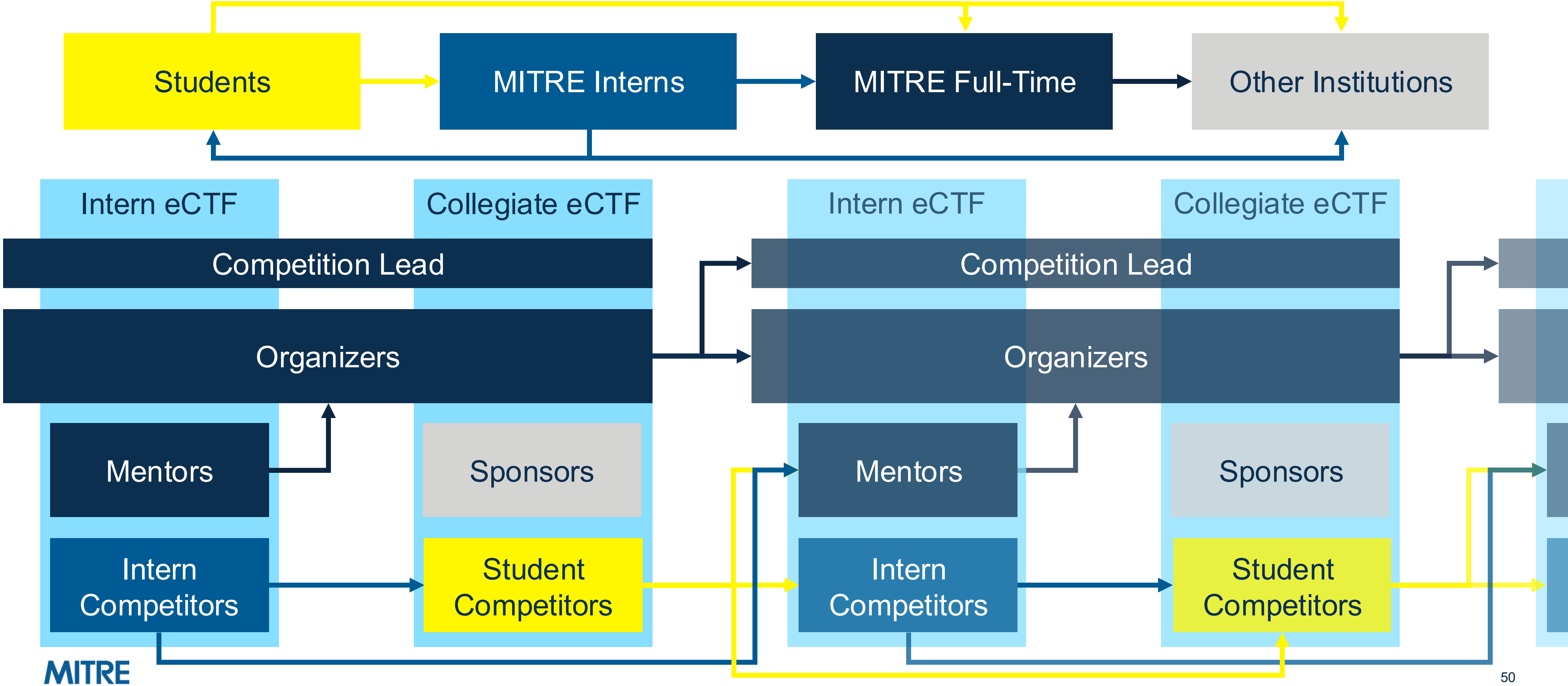
eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



MITRE

eCTF Participants Join Other Institutions

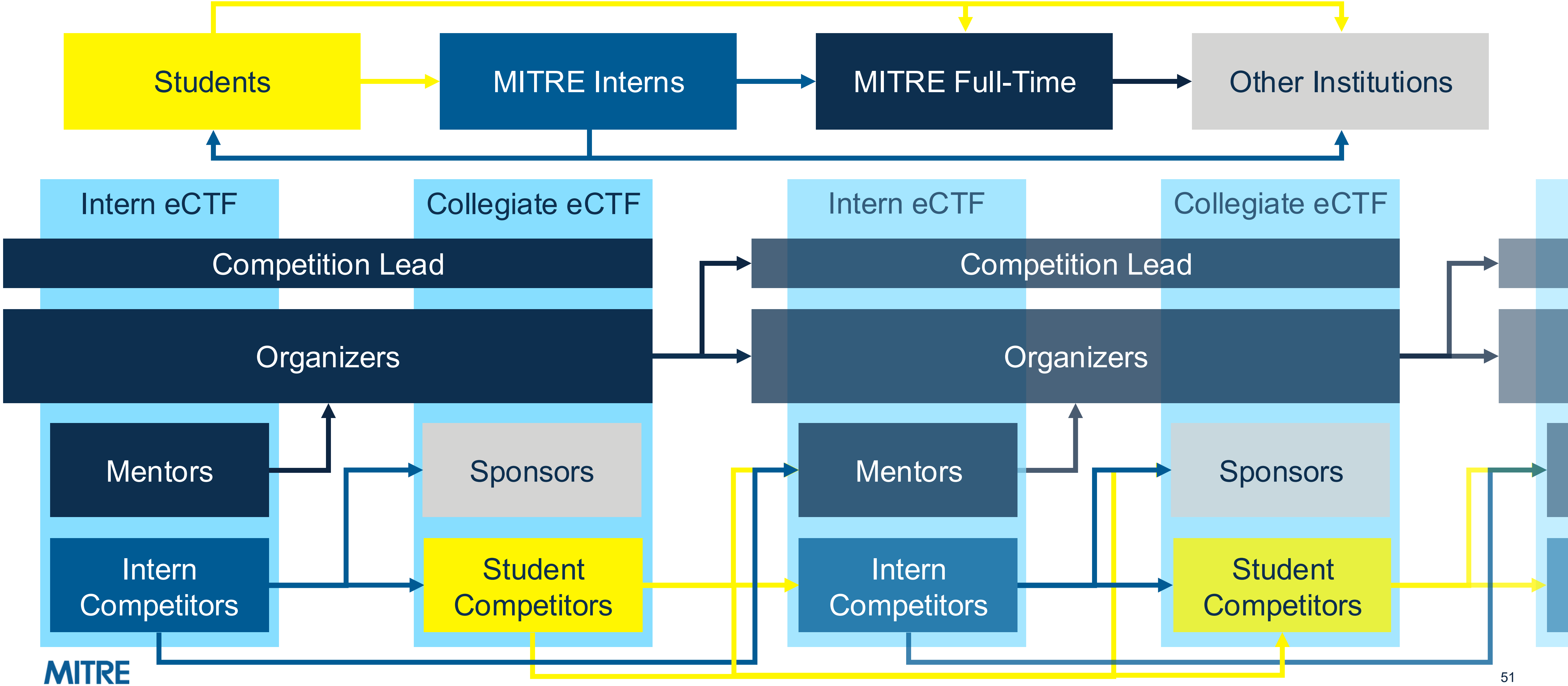
eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



MITRE

Former Participants Return as Sponsors

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



MITRE

Scoring System

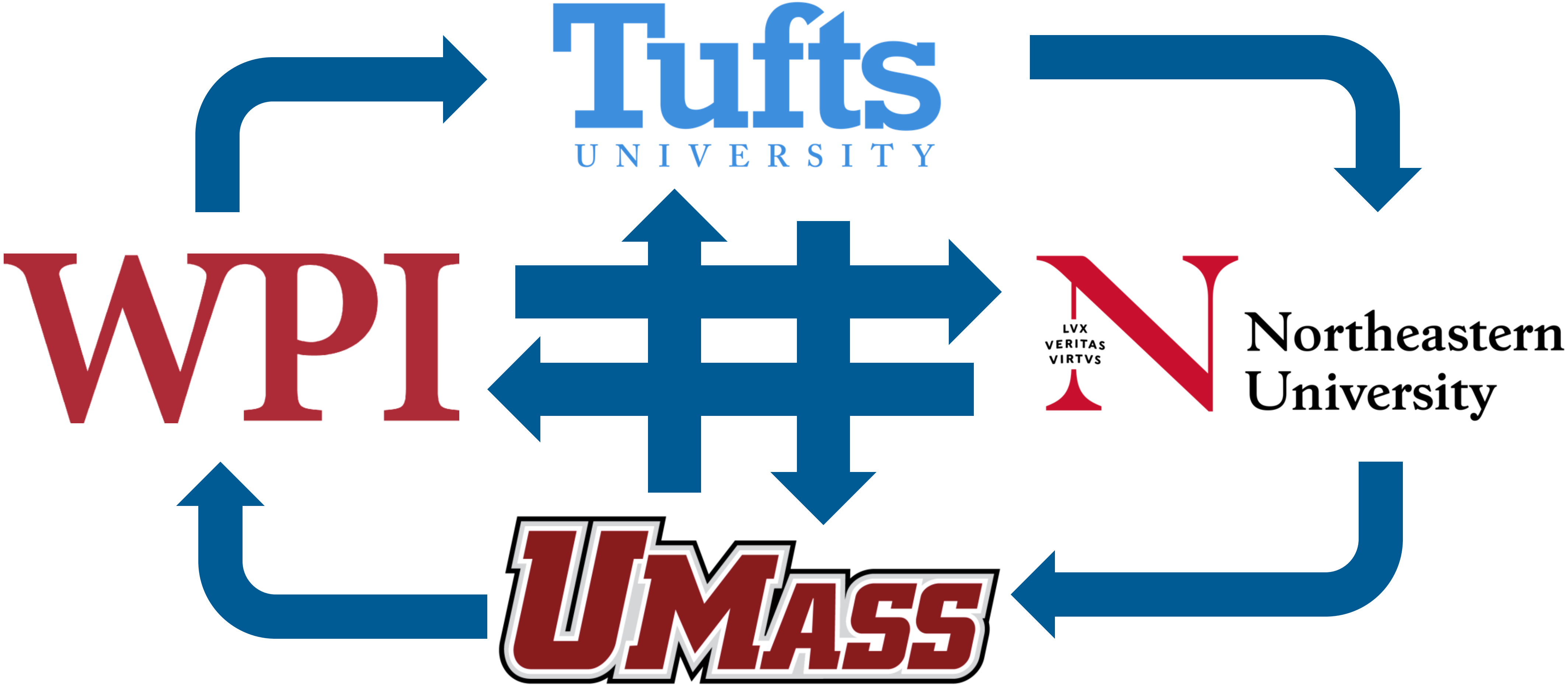
Scoring System

Reward strong designs

Encourage difficult attacks

No manual review by organizers

Year 1: Round Robin



Name	Point Value
Master PIN	300 / 150
Shoulder Surfing	350
New Neighbor	450
Stolen Widget	250
Cloning	200
Permanent Access	300

Scoring System

Reward strong designs

Encourage difficult attacks

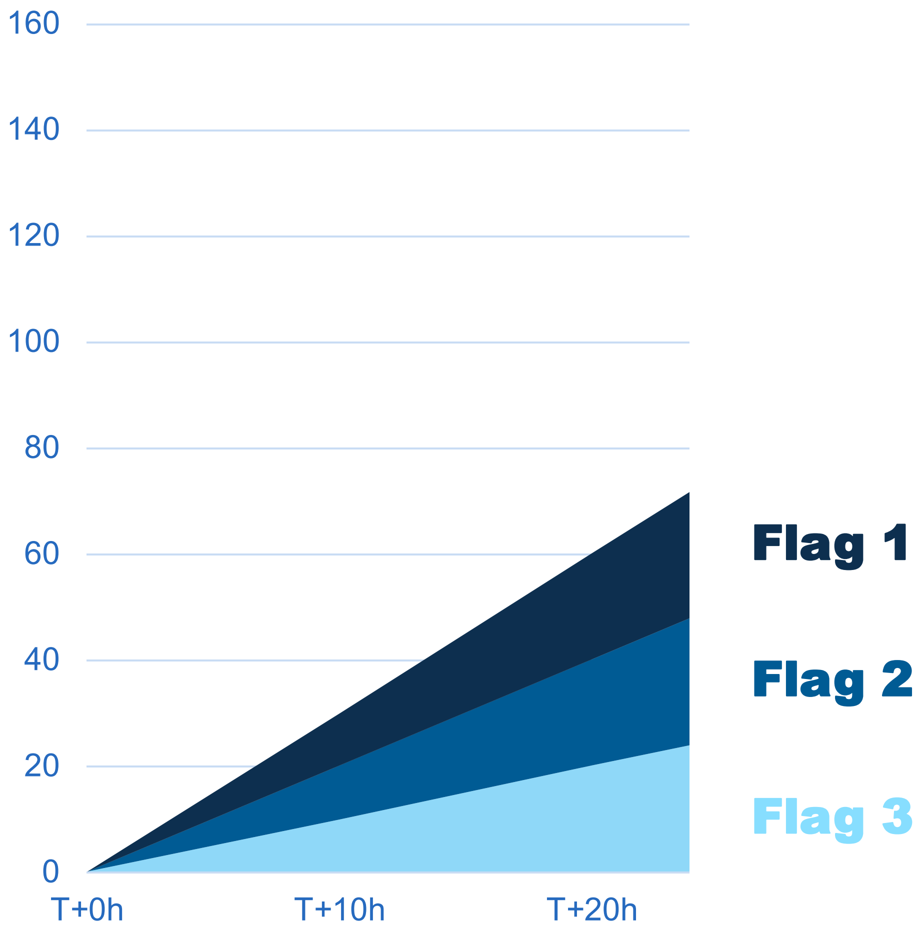
No manual review by organizers

Modern eCTF Scoring System

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

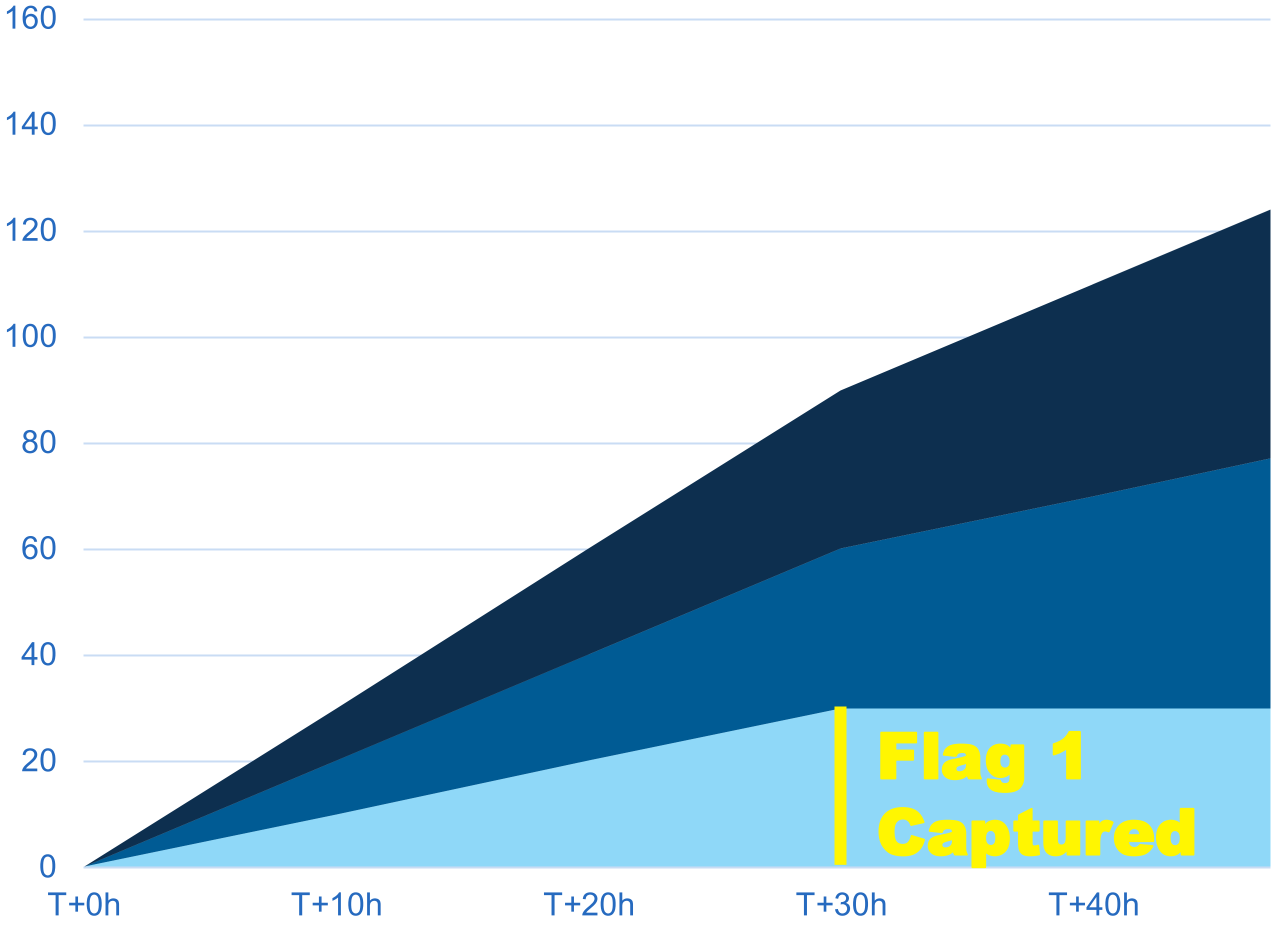
Score = Defensive + Offensive + Bonus

eCTF Scoring System: Defensive Points



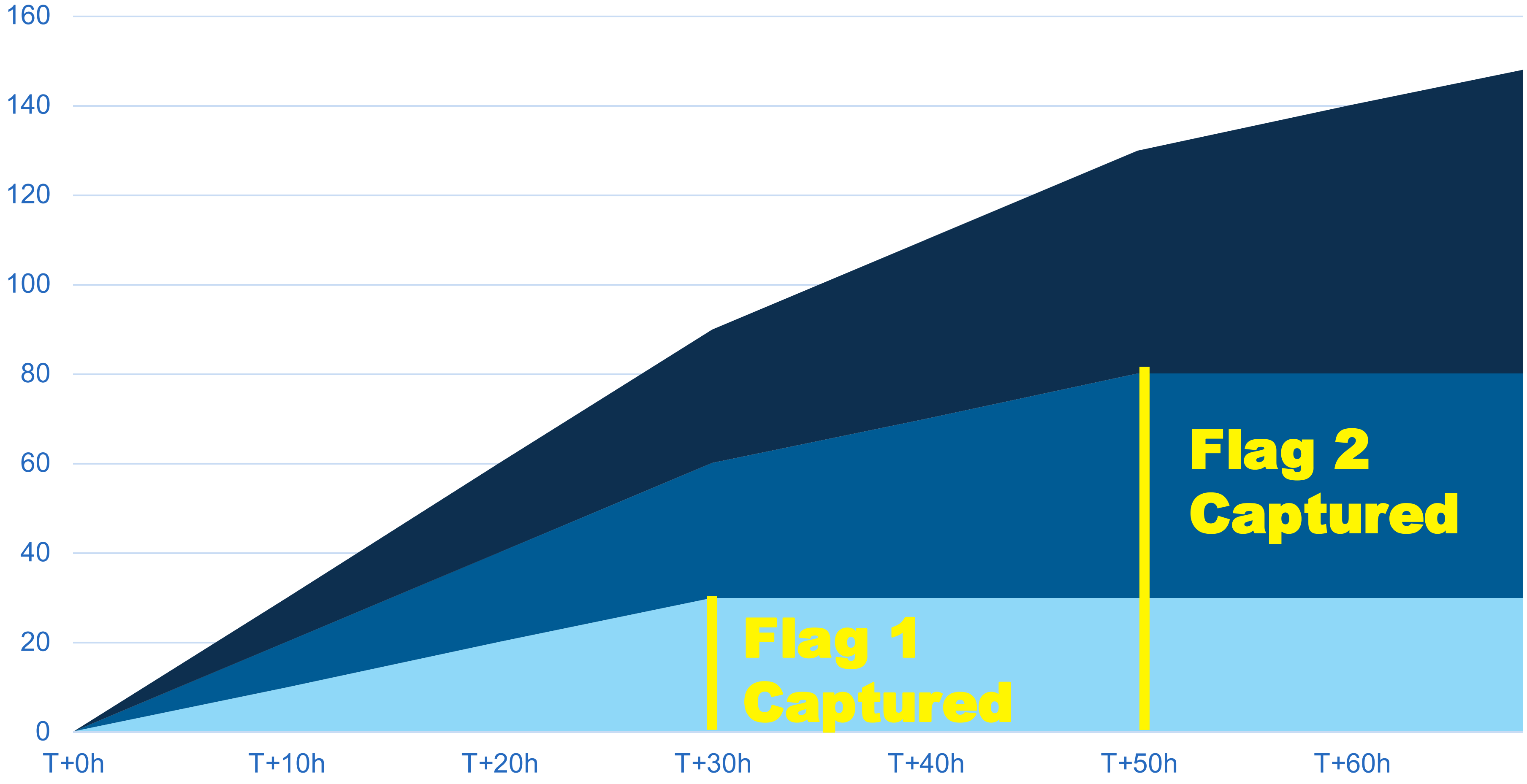
Each flag accrues points per hour until captured

eCTF Scoring System: Defensive Points



eCTF Scoring System: Defensive Points

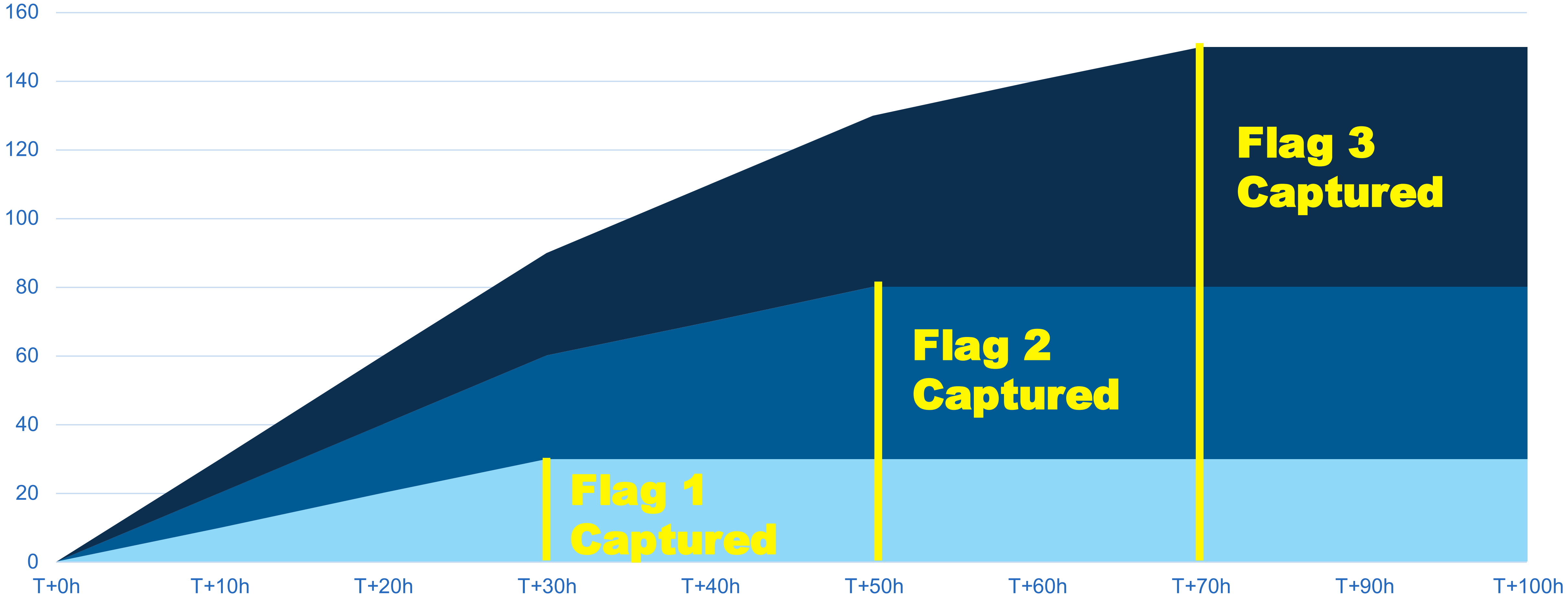
eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



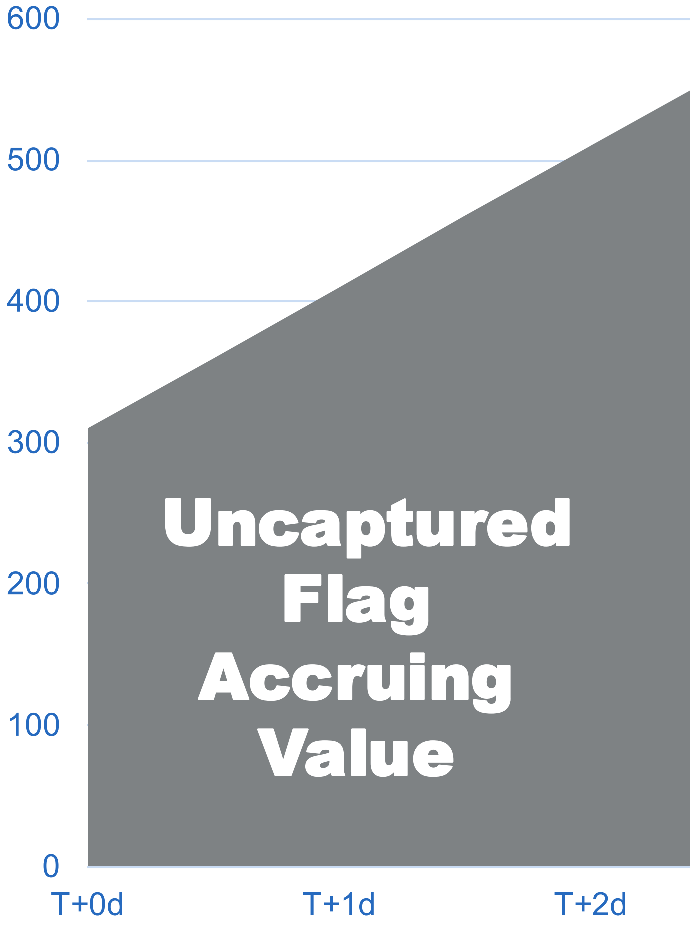
MITRE

eCTF Scoring System: Defensive Points

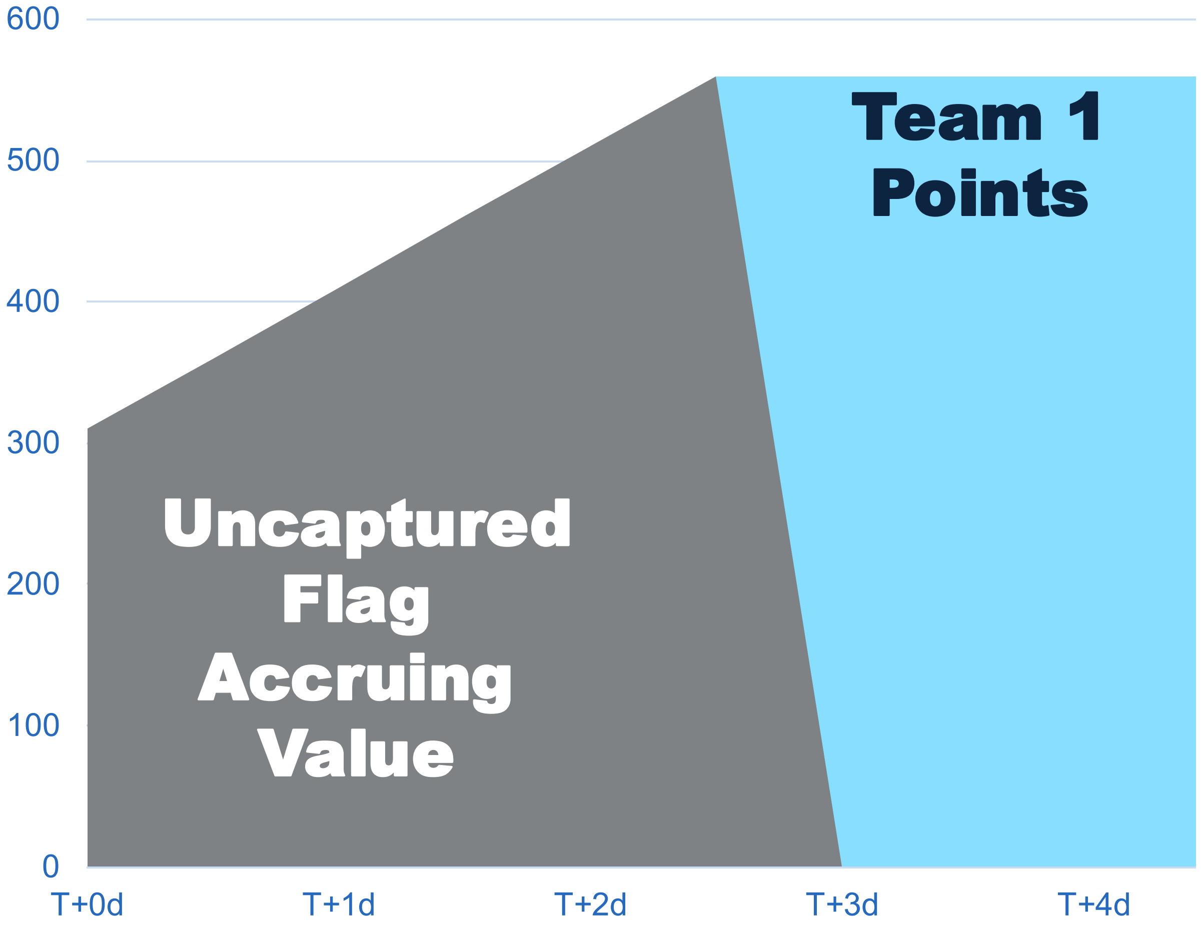
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



eCTF Scoring System: Offensive Points



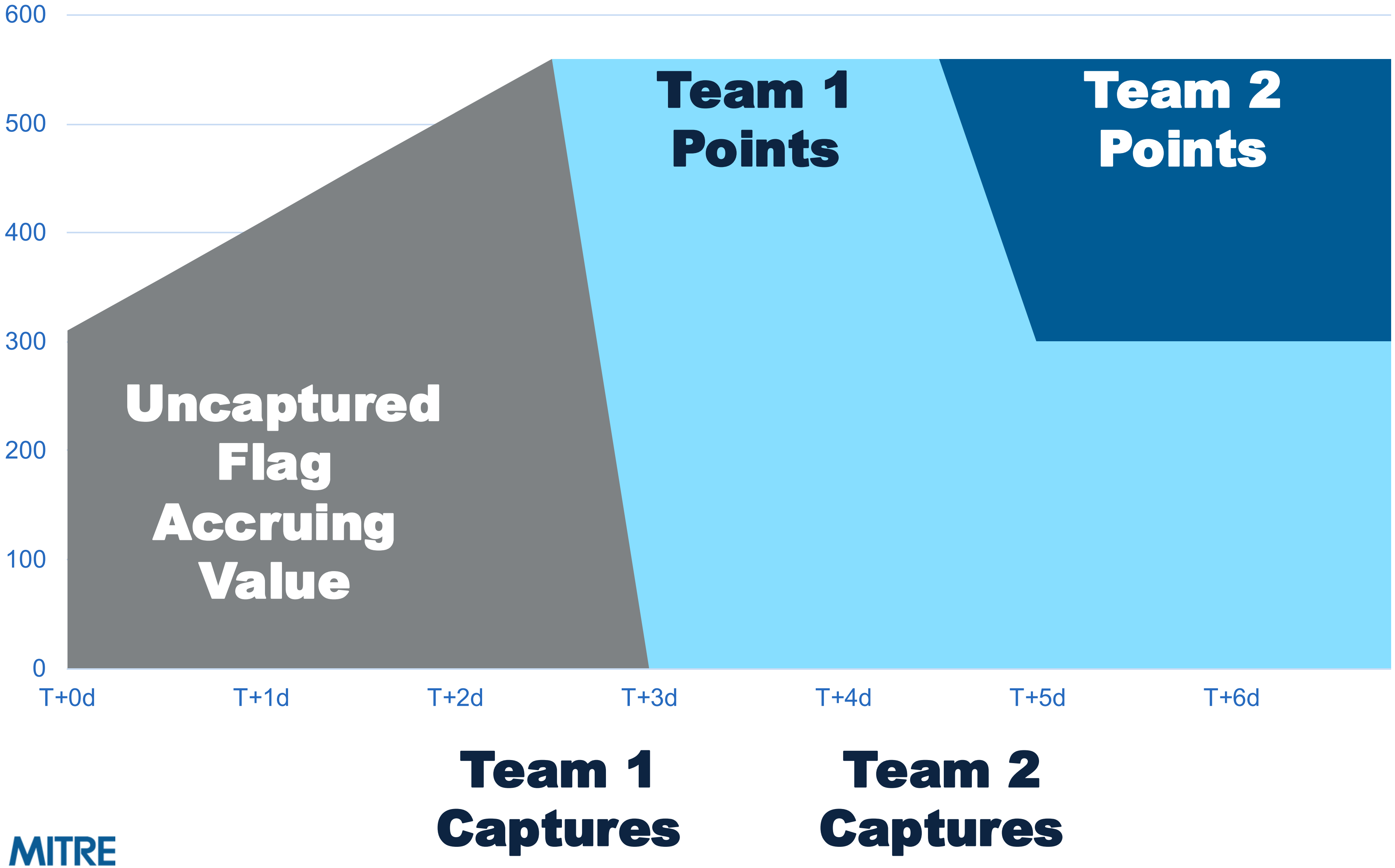
eCTF Scoring System: Offensive Points



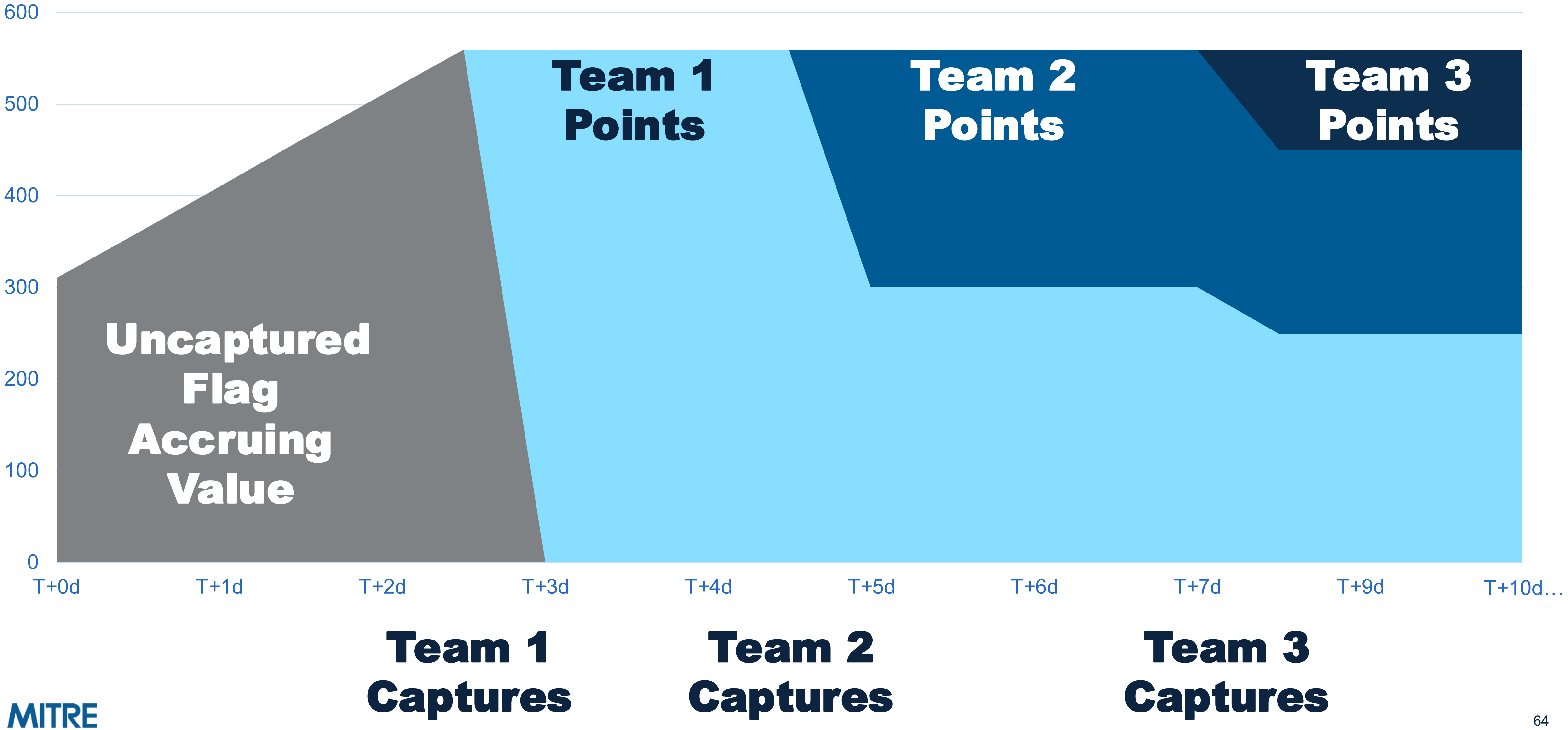
Team 1 Captures

eCTF Scoring System: Offensive Points

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

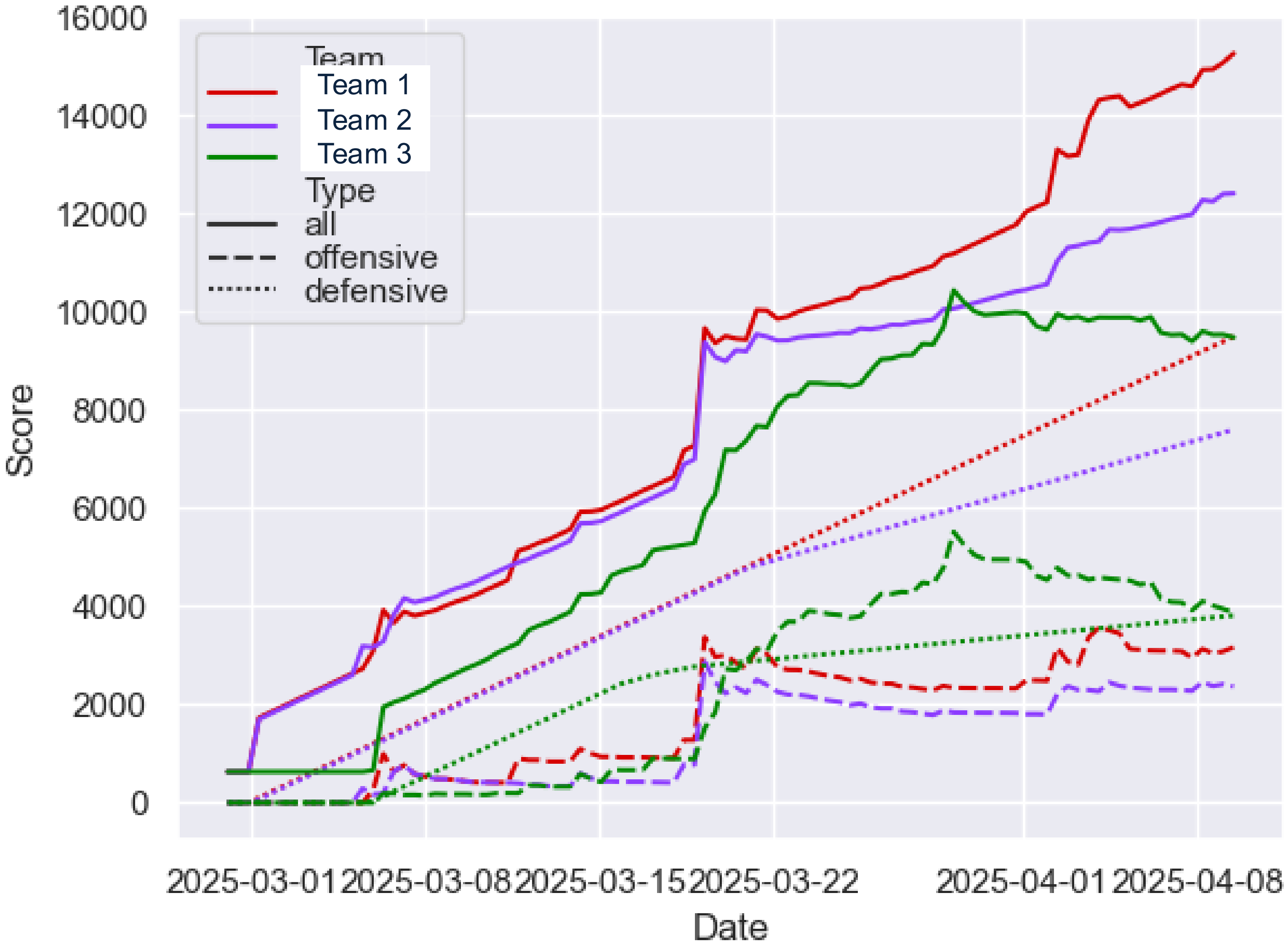


eCTF Scoring System: Offensive Points



Example From 2025 Attack Phase

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



MITRE

Scoring System

Reward strong designs

Encourage difficult attacks

No manual review by organizers

Hardware Challenges

Hardware Challenges

2016

Manual Verification

Hardware Challenges

2016

Manual Verification

2017

N² Complexity

Hardware Challenges

2016

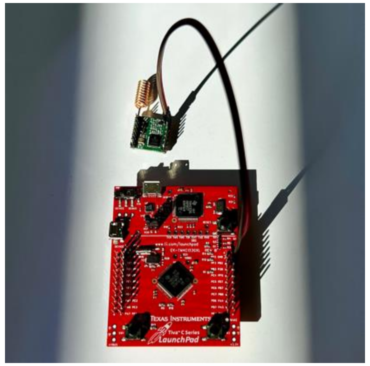
Manual Verification

2017

N² Complexity

2018

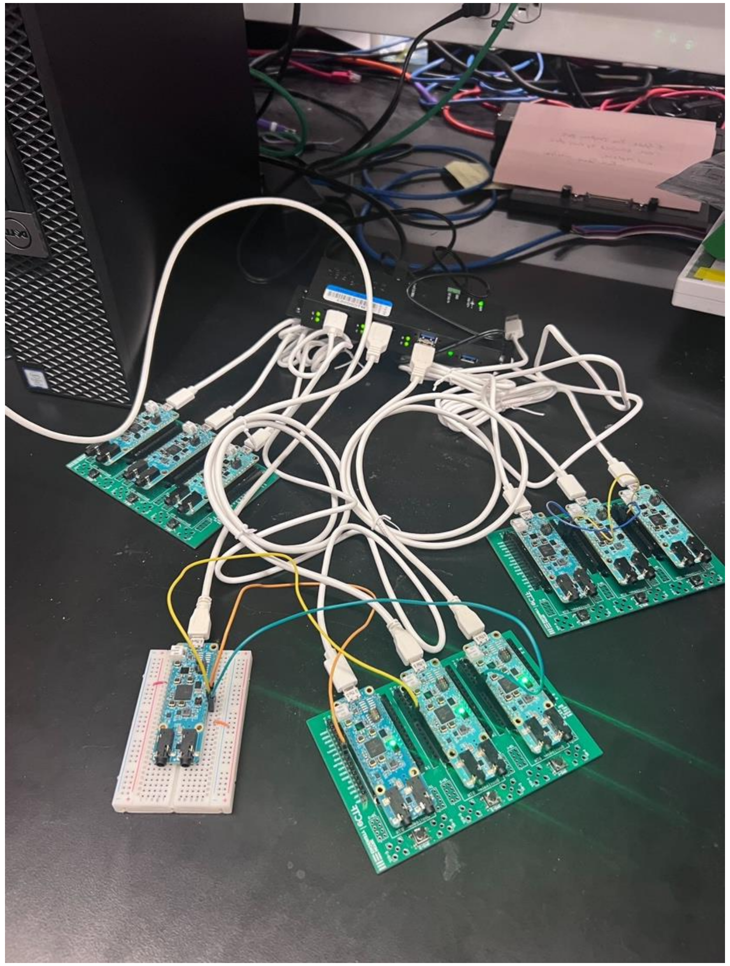
Secure Bootloader



Other Hardware Lessons

Invest in good USB hubs

Bus protocols can be tricky



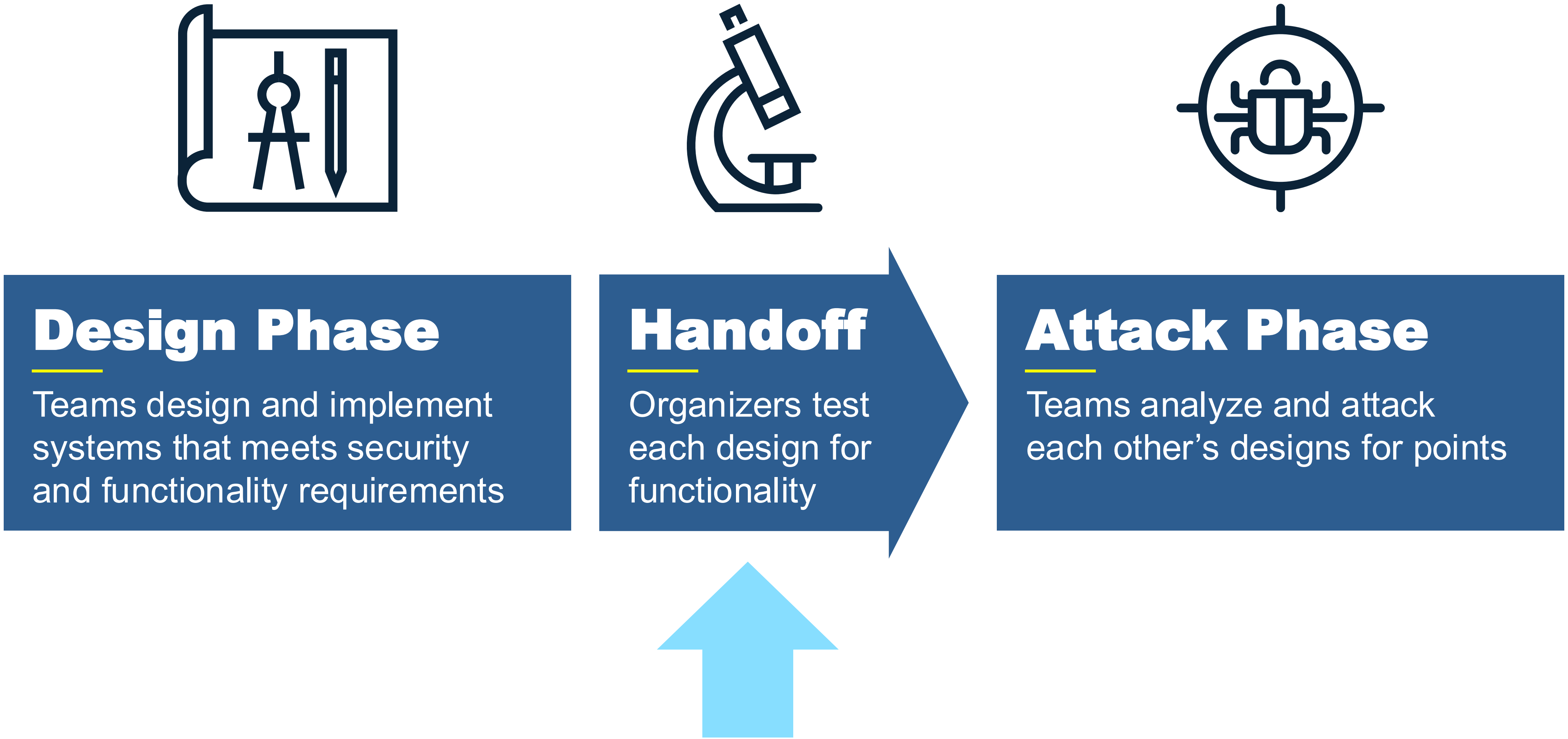
Redundancy
Redundancy
Redundancy

Verify your root of trust

Testing Framework

Testing Infrastructure

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



Design Phase

Teams design and implement systems that meets security and functionality requirements

Handoff

Organizers test each design for functionality

Attack Phase

Teams analyze and attack each other’s designs for points

Testing Process

eCTF 10

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

Clone team repository

ERROR: Repository not found. Please make sure you have the correct access rights

Compile firmware

“Well, it works on my computer”

Flash firmware onto hardware

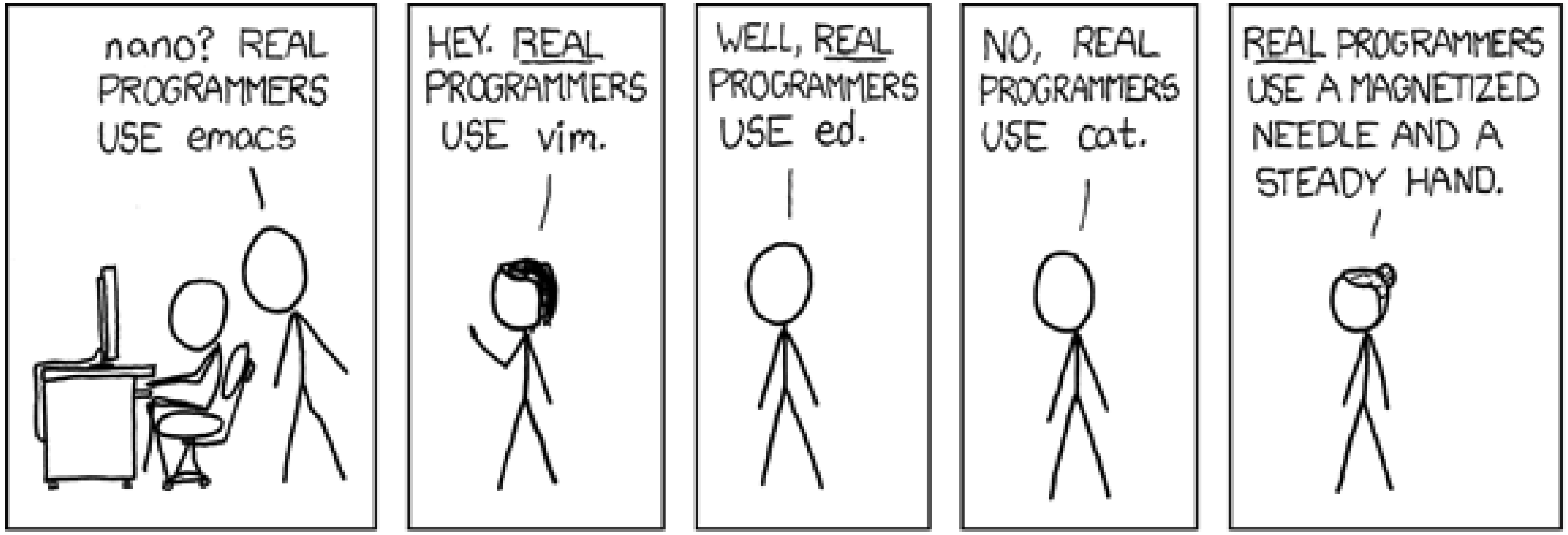
“Where did the serial device go?”

Run tests on hardware

No test survives first contact

MITRE

Testing Service v0.0



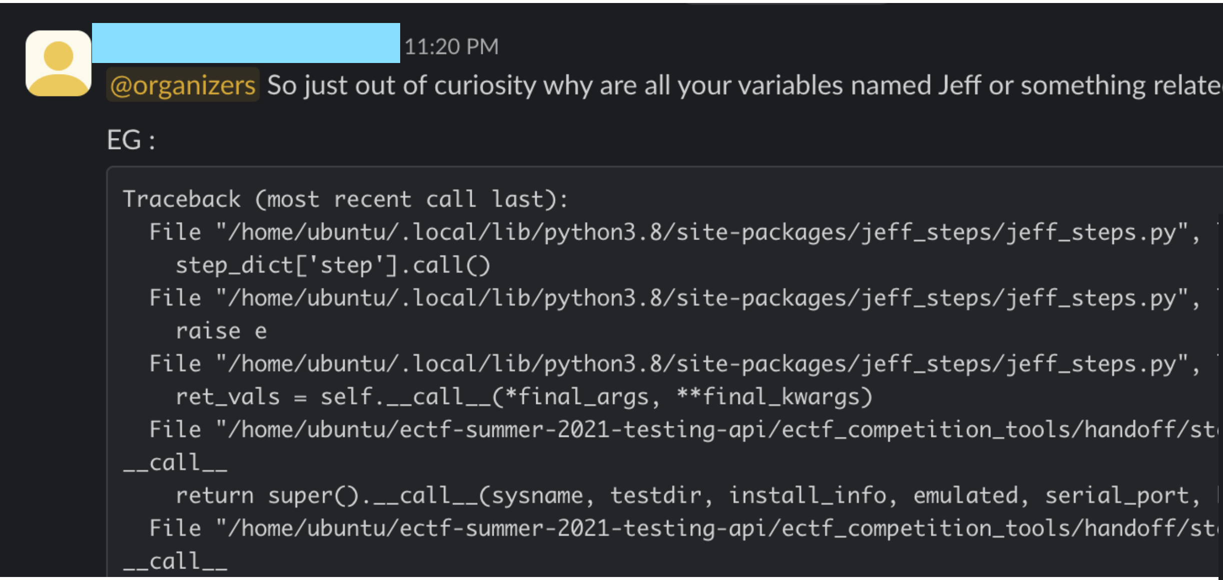
nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.

https://xkcd.com/378/

Jeff

Testing Service v1.0: JeffSteps

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

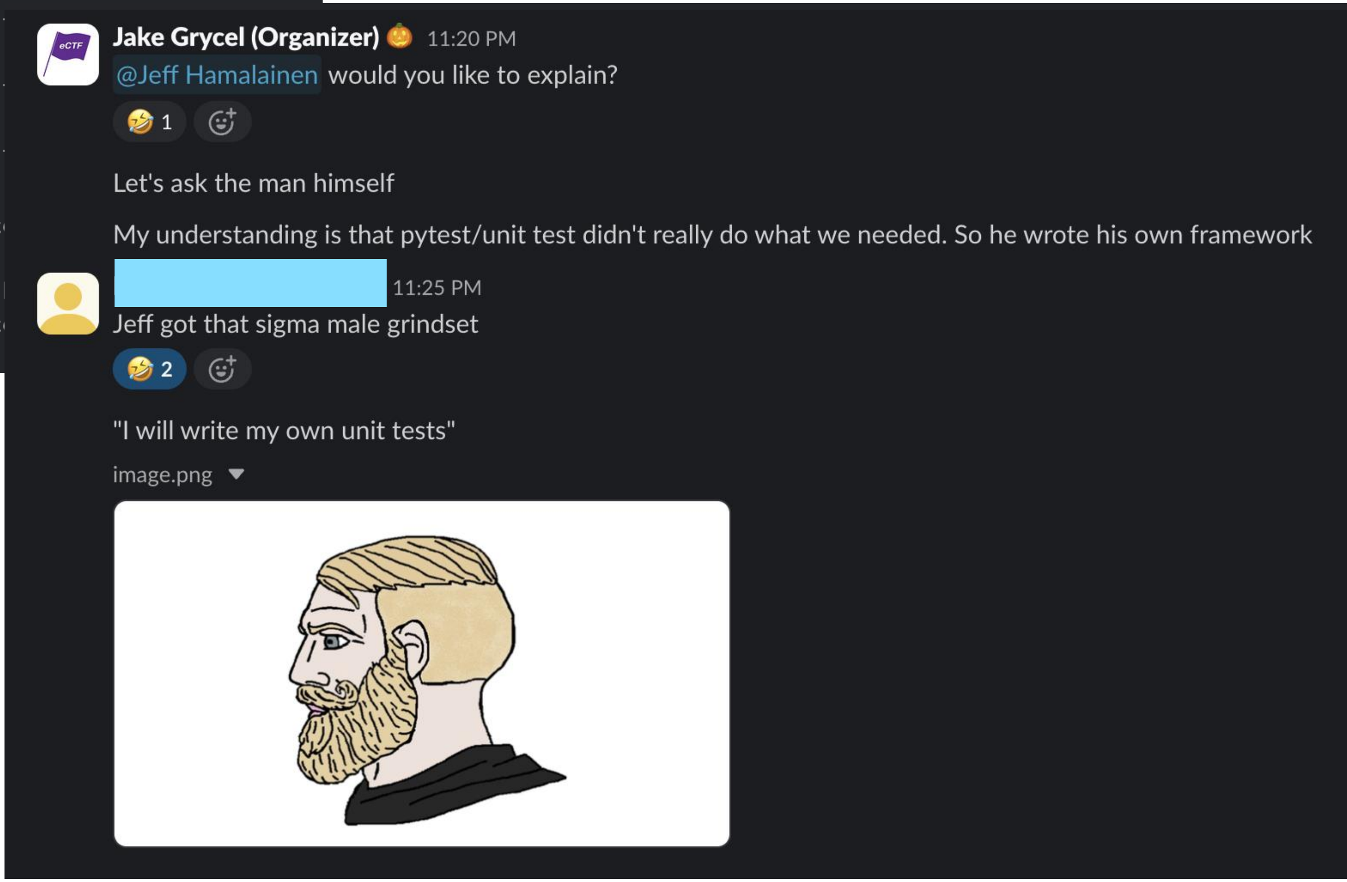


[redacted] 11:20 PM

@organizers So just out of curiosity why are all your variables named Jeff or something related to jeff

EG :

```
Traceback (most recent call last):
  File "/home/ubuntu/.local/lib/python3.8/site-packages/jeff_steps/jeff_steps.py",
    step_dict['step'].call()
  File "/home/ubuntu/.local/lib/python3.8/site-packages/jeff_steps/jeff_steps.py",
    raise e
  File "/home/ubuntu/.local/lib/python3.8/site-packages/jeff_steps/jeff_steps.py",
    ret_vals = self.__call__(*final_args, **final_kwargs)
  File "/home/ubuntu/ectf-summer-2021-testing-api/ectf_competition_tools/handoff/st
__call__
    return super().__call__(sysname, testdir, install_info, emulated, serial_port,
  File "/home/ubuntu/ectf-summer-2021-testing-api/ectf_competition_tools/handoff/st
__call__
```



Jake Grycel (Organizer) 11:20 PM

@Jeff Hamalainen would you like to explain?

🤣 1

Let's ask the man himself

My understanding is that pytest/unit test didn't really do what we needed. So he wrote his own framework

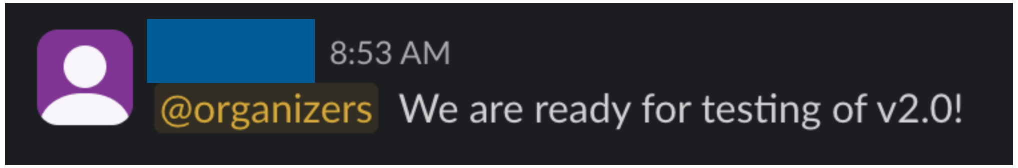
[redacted] 11:25 PM

Jeff got that sigma male grindset

🤣 2

"I will write my own unit tests"

image.png ▾

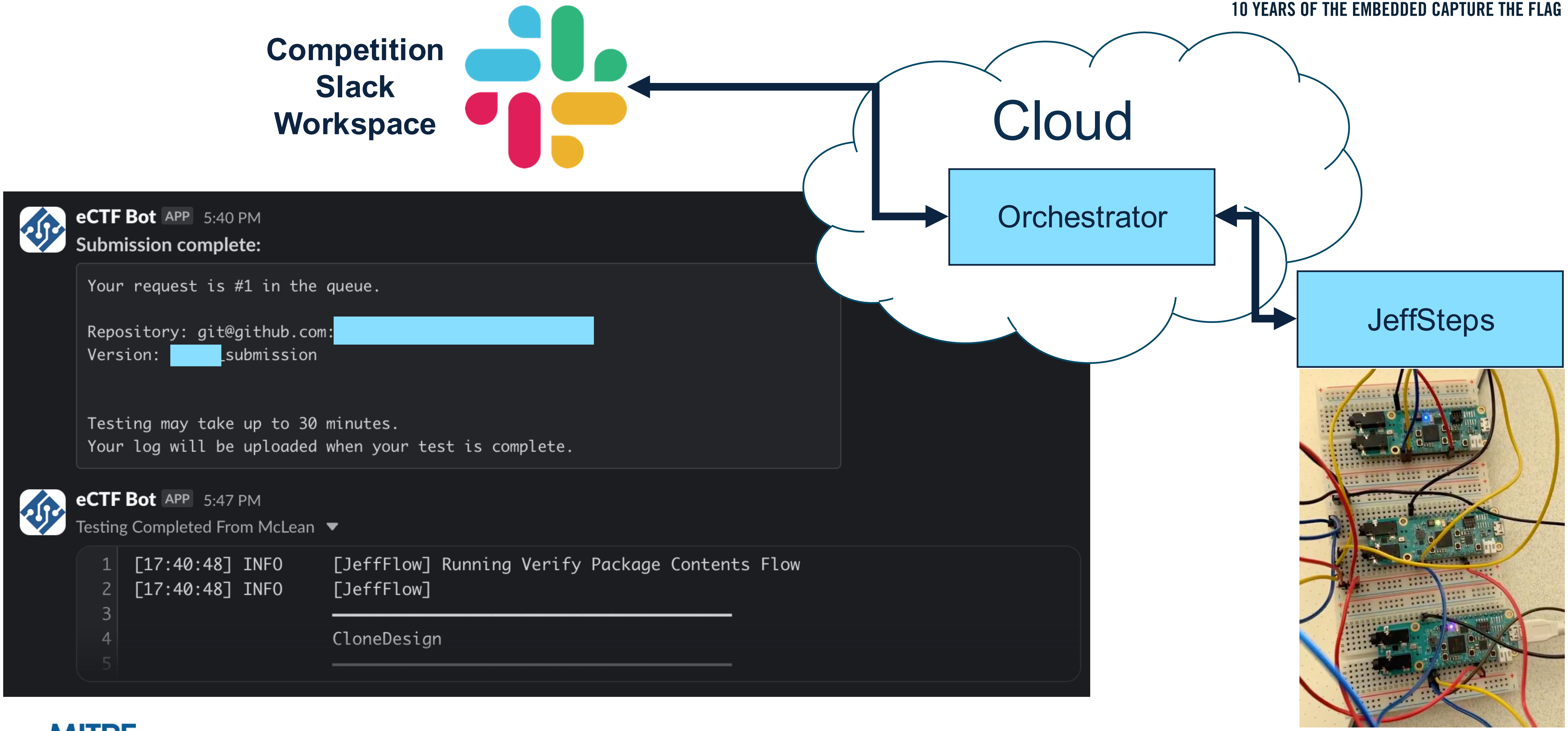


[redacted] 8:53 AM

@organizers We are ready for testing of v2.0!

MITRE

Testing Service v2.0: Slackbot



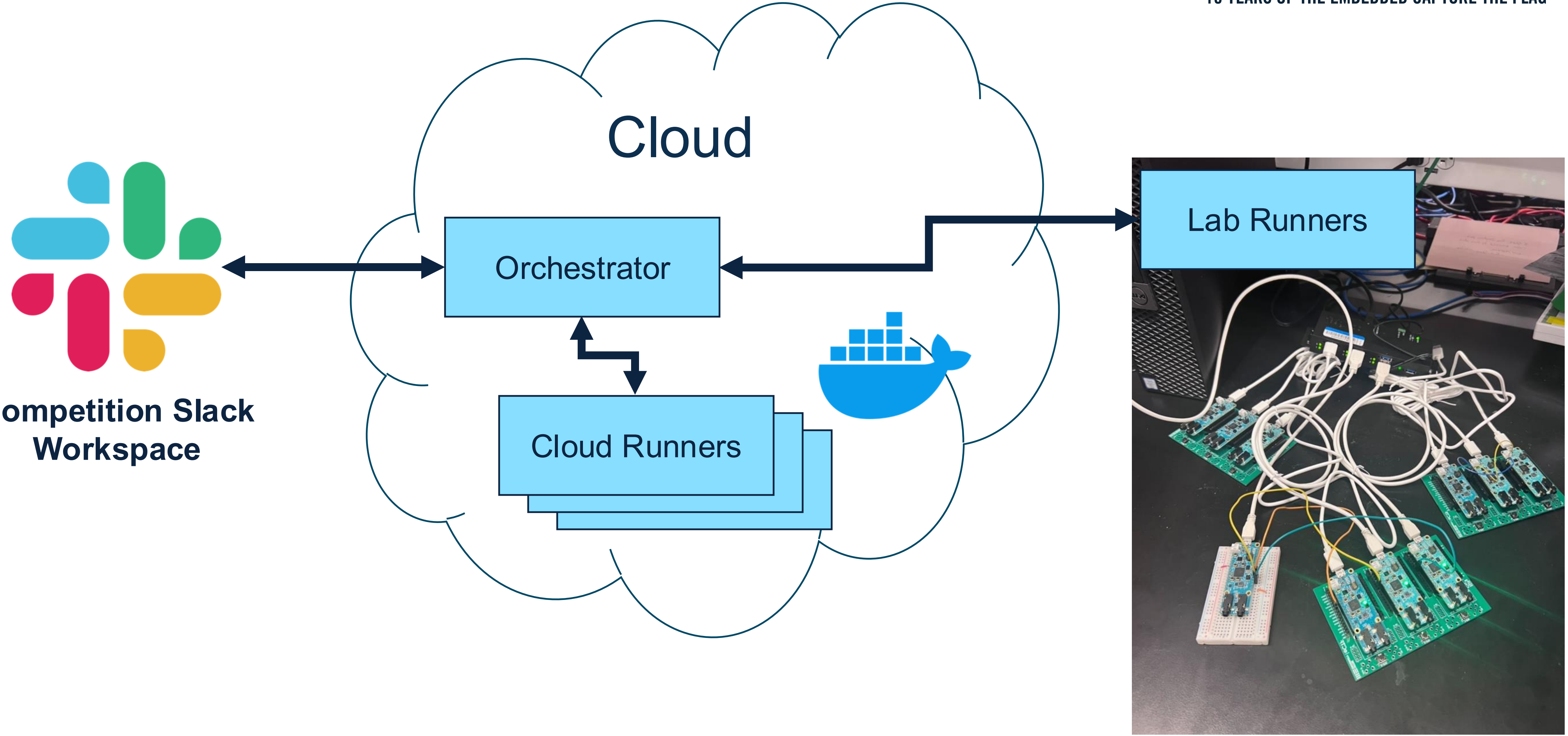
eCTF Bot APP 5:40 PM
Submission complete:

```
Your request is #1 in the queue.

Repository: git@github.com:[redacted]
Version: [redacted]_submission


Testing may take up to 30 minutes.
Your log will be uploaded when your test is complete.
```

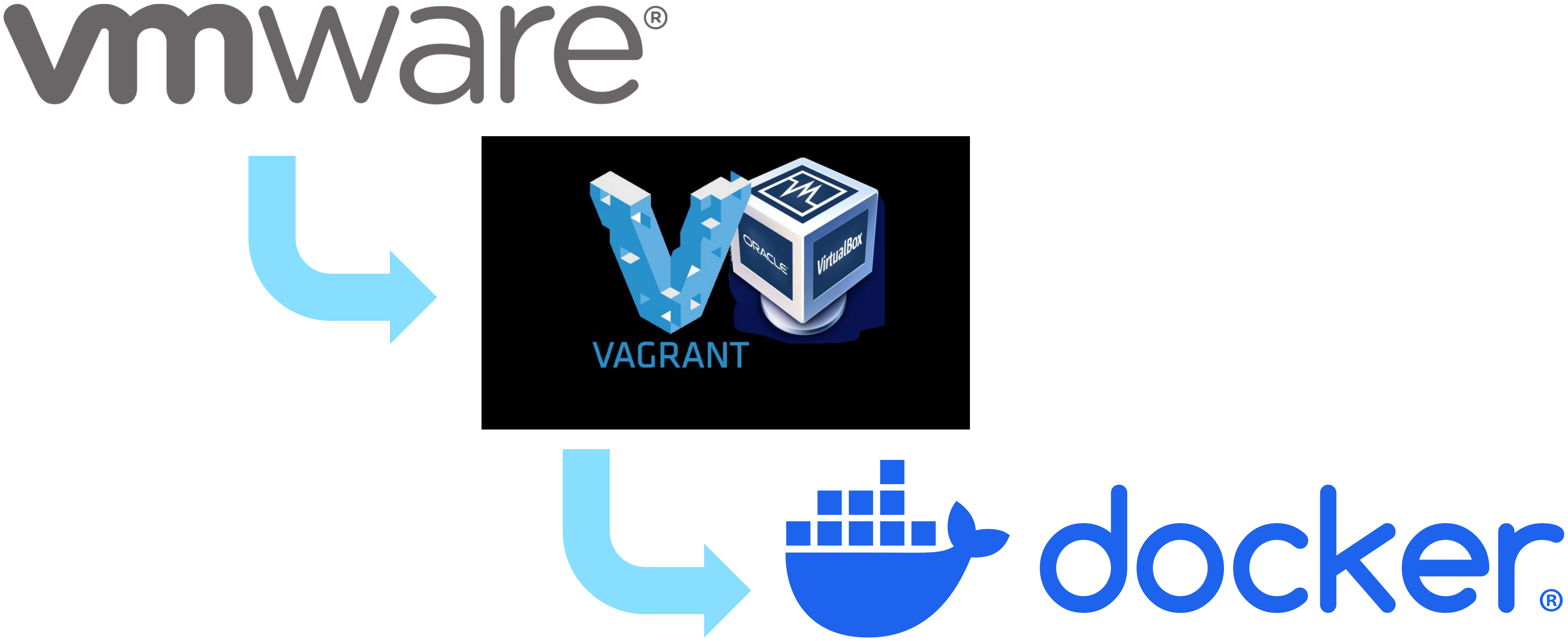
eCTF Bot APP 5:47 PM
Testing Completed From McLean ▾

```
[17:40:48] INFO     [JeffFlow] Running Verify Package Contents Flow
[17:40:48] INFO     [JeffFlow]
                    ──────────────────────────────────────
                    CloneDesign
                    ──────────────────────────────────────
```

Testing Service v3.0: eCTF Runners



Cross-Platform Repeatability



Competitor Support

Support Resources

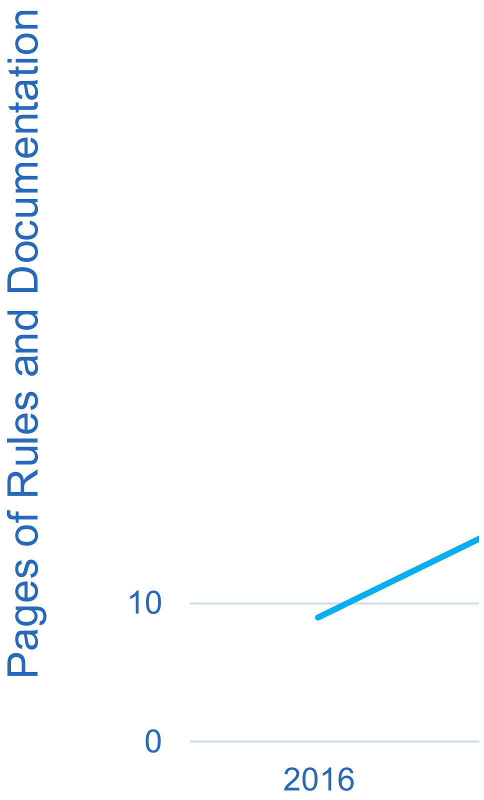
Documentation

Design Phase Flags

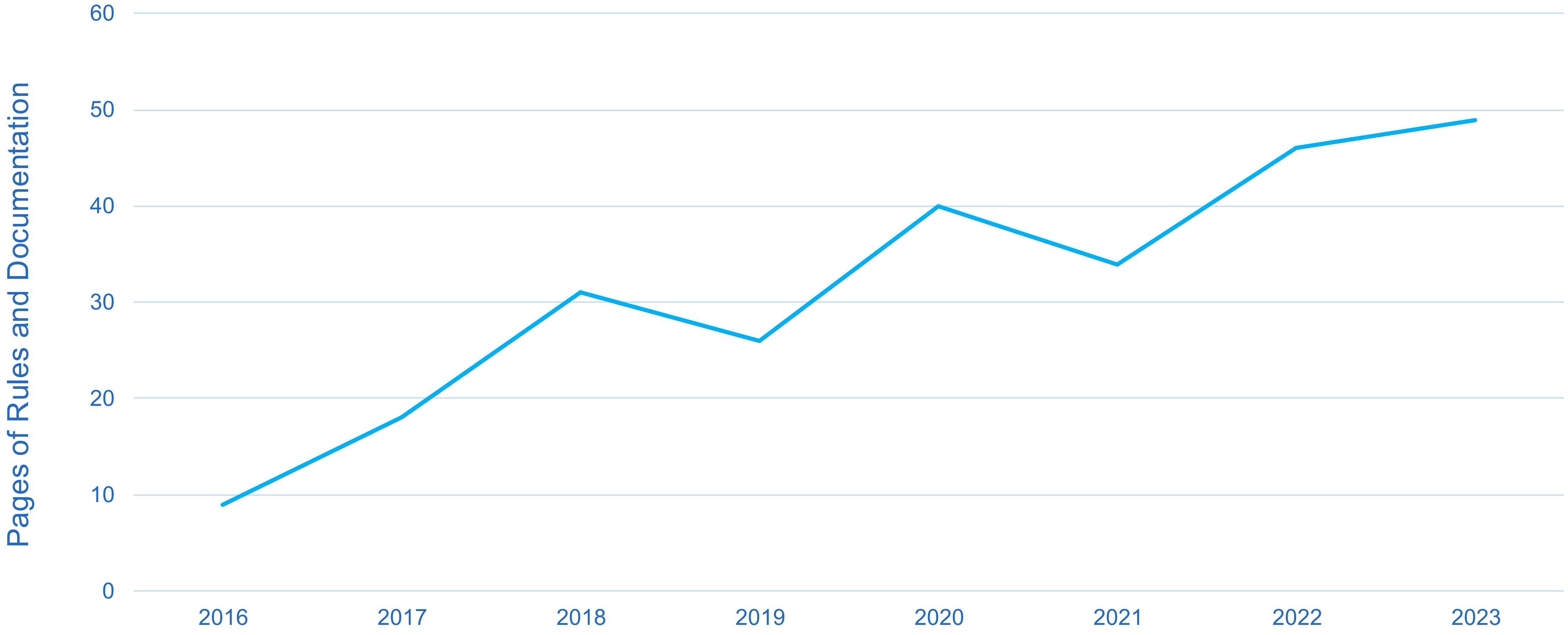
Organizer Support

Team Structure

Documentation Explosion



Documentation Explosion



2024- Rules Site

https://rules.ectf.mitre.org

Generated by Sphinx

Hosted on GitLab Pages

eCTF 10

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

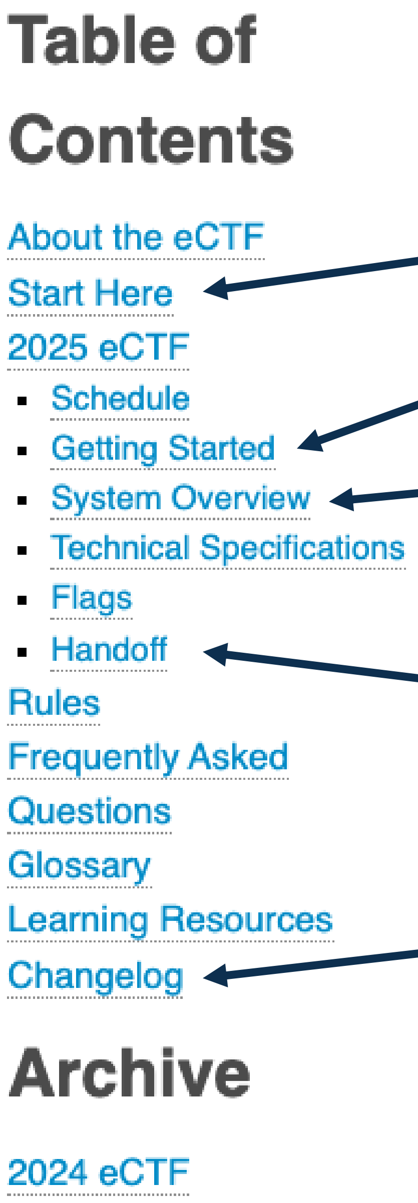


Table of Contents

- About the eCTF
- Start Here
- 2025 eCTF
  - Schedule
  - Getting Started
  - System Overview
  - Technical Specifications
  - Flags
  - Handoff
- Rules
- Frequently Asked Questions
- Glossary
- Learning Resources
- Changelog

Archive

- 2024 eCTF

Info for new teams → Start Here

Year-specific setup → Getting Started

High-level overview → System Overview

Technical details → Technical Specifications

How to score points → Flags

How to pass testing → Handoff

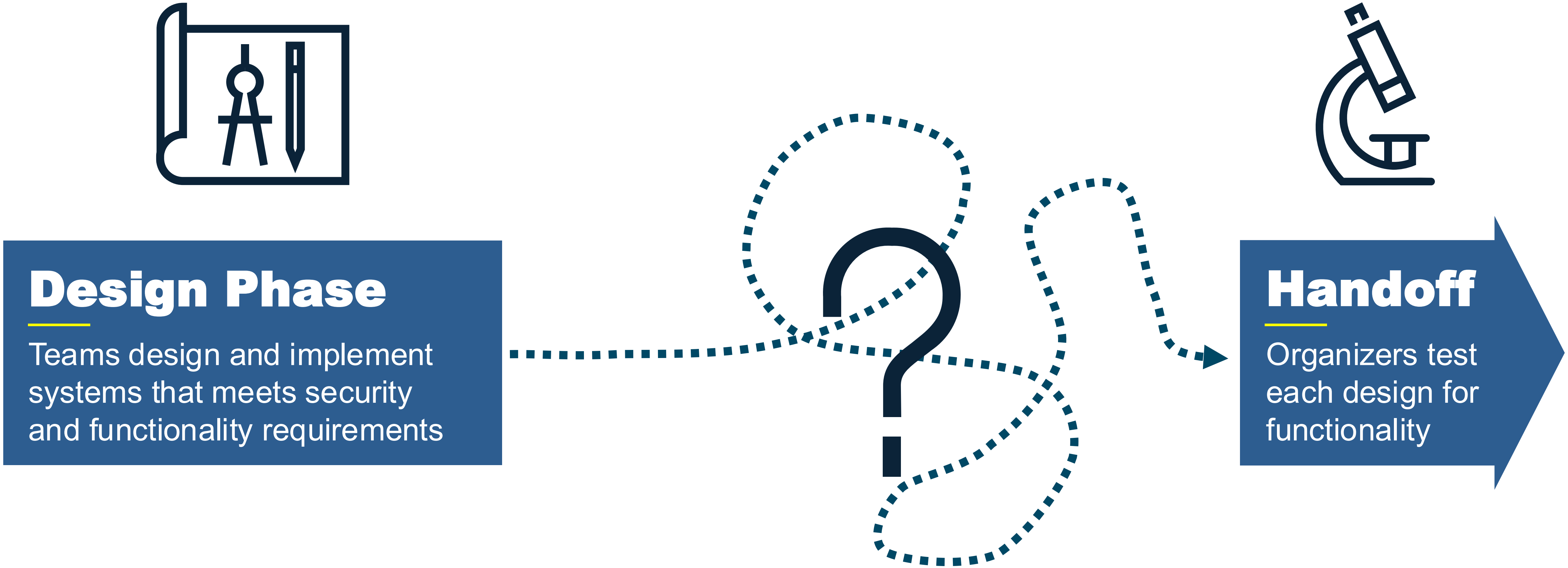
Competition rules → Rules

Helpful information → Frequently Asked Questions, Glossary, Learning Resources

Log of rule changes → Changelog

Past years → Archive

Design Phase Confusion



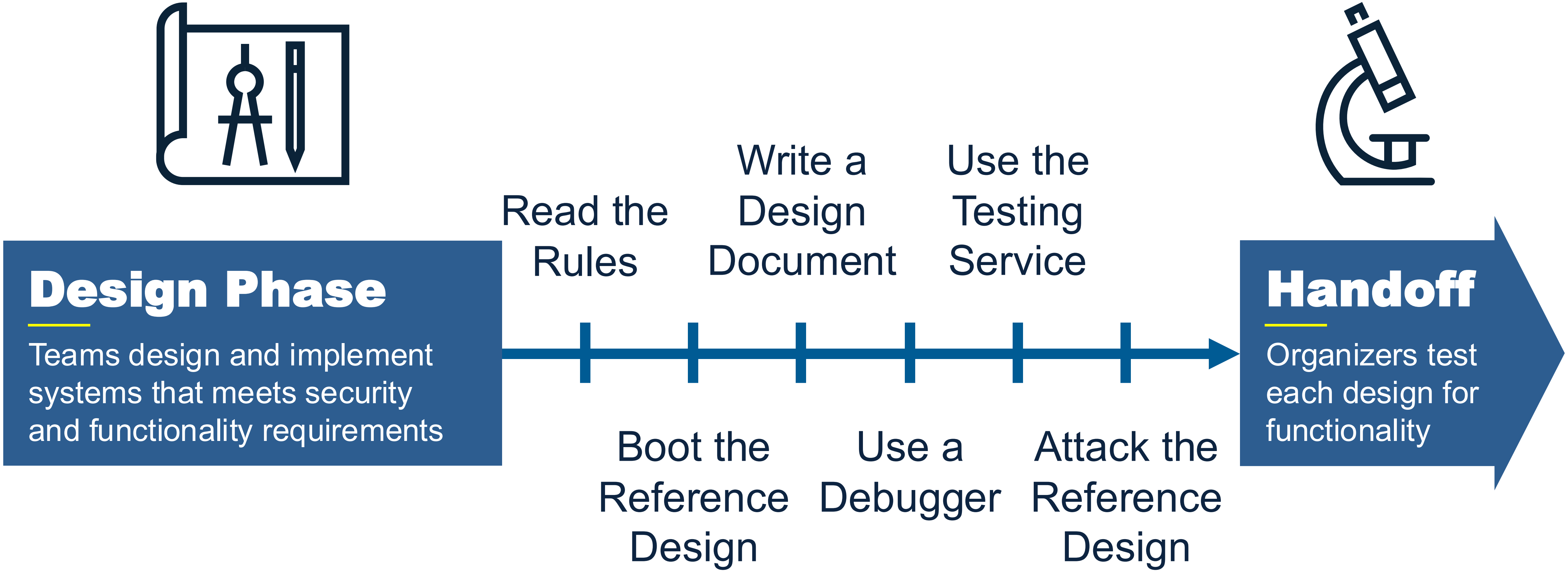
Design Phase

Teams design and implement systems that meets security and functionality requirements

Handoff

Organizers test each design for functionality

Design Phase Flags



Design Phase

Teams design and implement systems that meets security and functionality requirements

Read the Rules

Boot the Reference Design

Write a Design Document

Use a Debugger

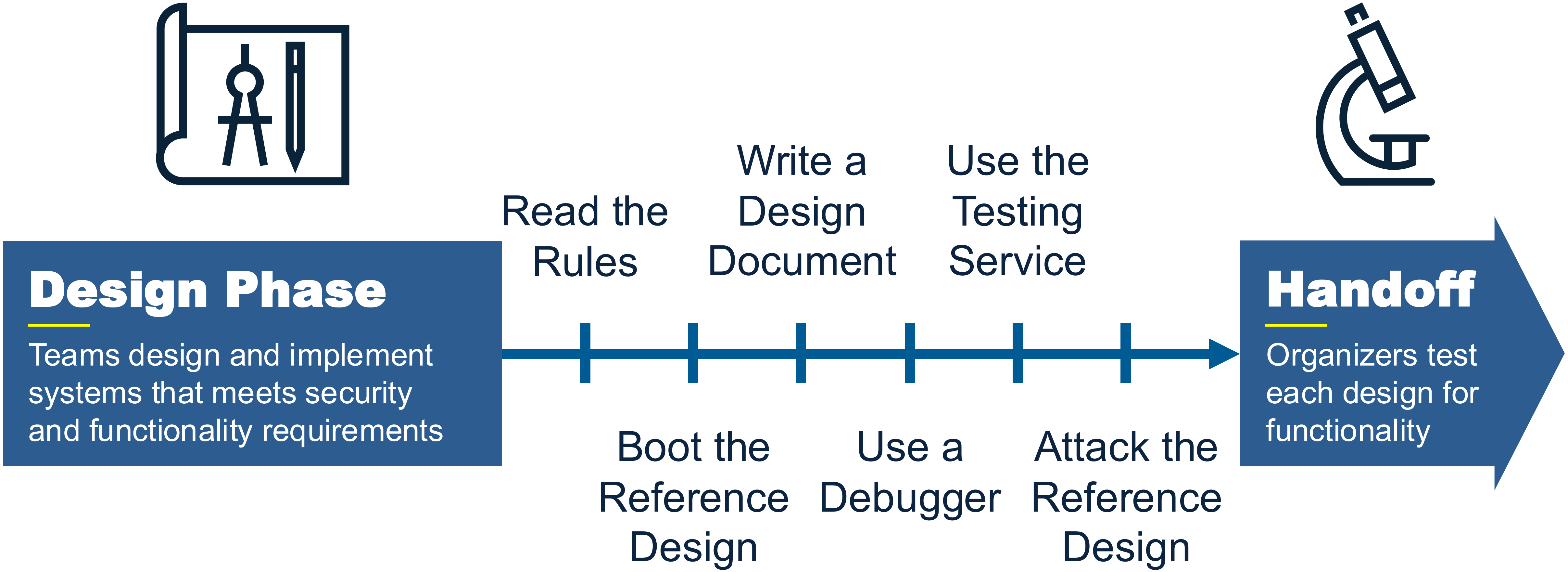
Use the Testing Service

Attack the Reference Design

Handoff

Organizers test each design for functionality

Office Hours



Design Phase

Teams design and implement systems that meets security and functionality requirements

Read the Rules

Boot the Reference Design

Write a Design Document

Use a Debugger

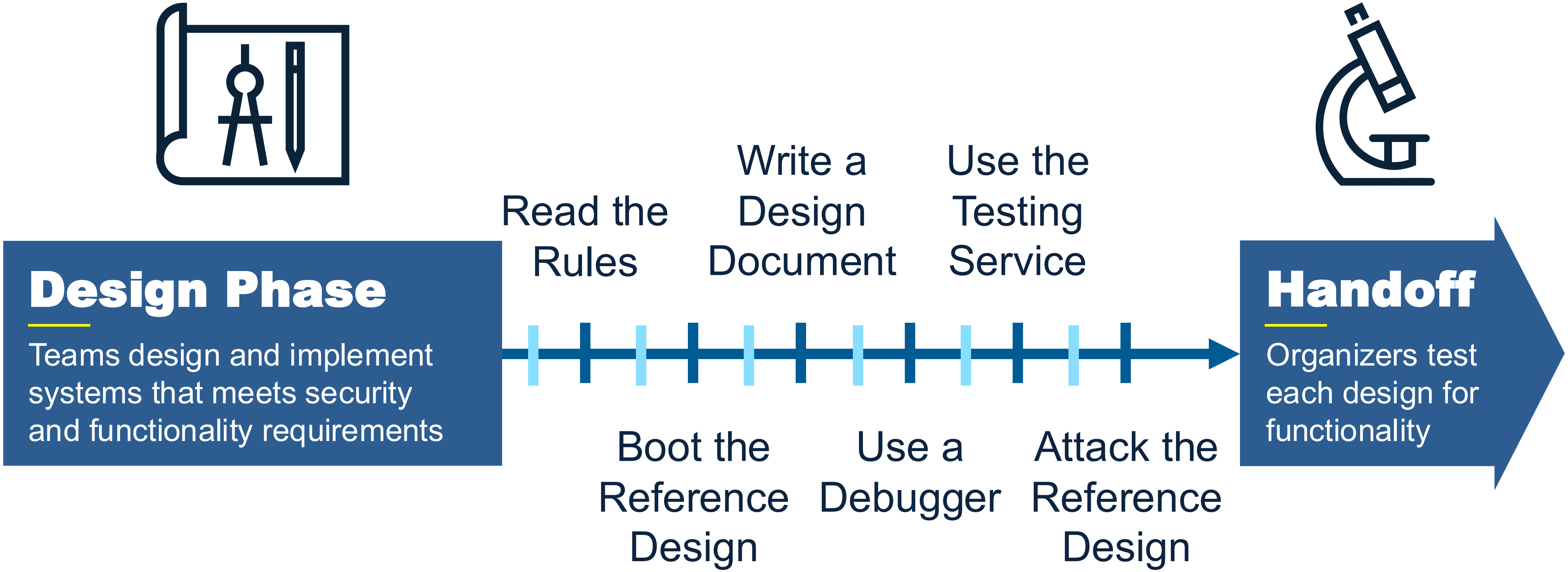
Use the Testing Service

Attack the Reference Design

Handoff

Organizers test each design for functionality

Workshops



Design Phase

Teams design and implement systems that meets security and functionality requirements

Read the Rules

Boot the Reference Design

Write a Design Document

Use a Debugger

Use the Testing Service

Attack the Reference Design

Handoff

Organizers test each design for functionality

Workshop Tips

Focused

Regular

Recorded

Team Structure

Project Management

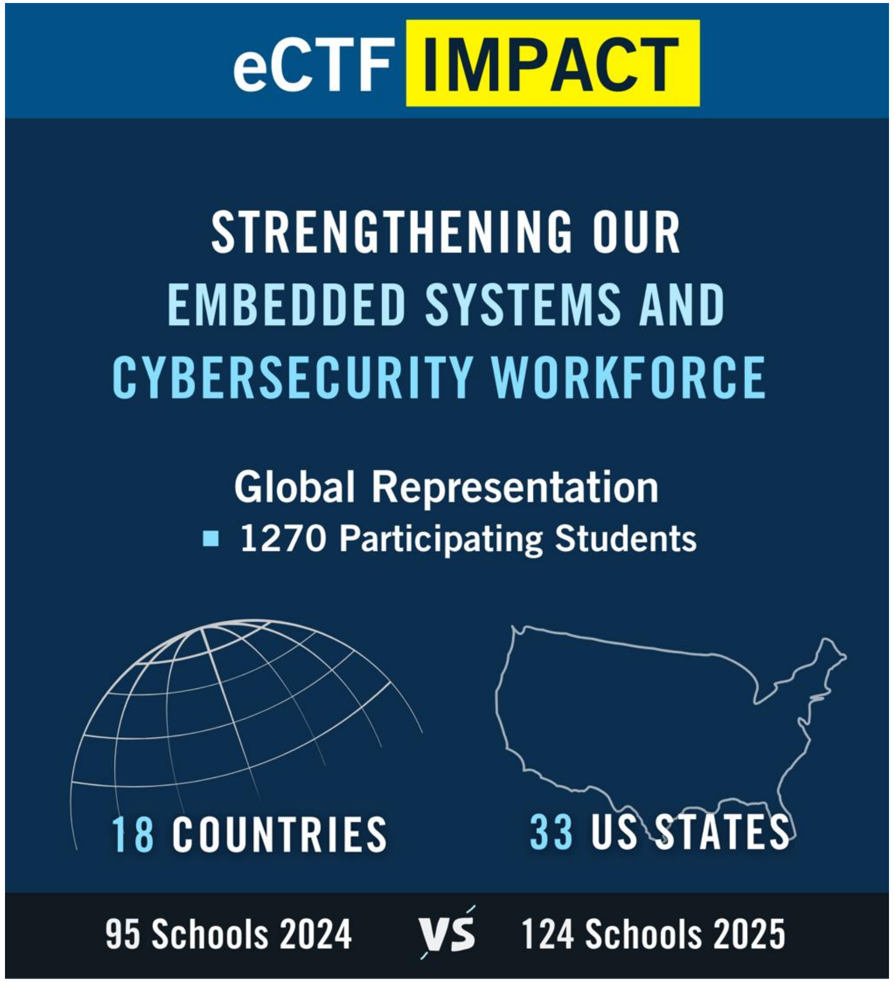
eCTF for Credit

Mentorship

Financial Sustainability

Workforce Development

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



eCTF **IMPACT**

STRENGTHENING OUR EMBEDDED SYSTEMS AND CYBERSECURITY WORKFORCE

Global Representation

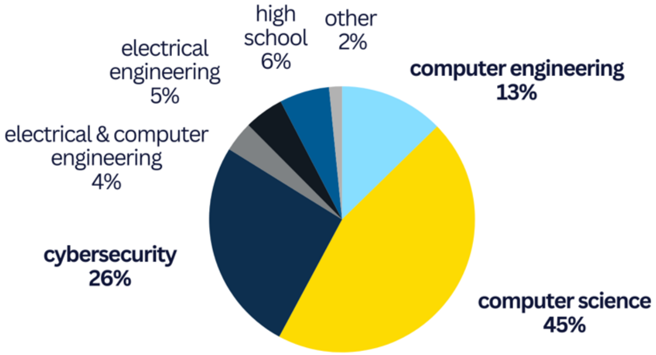
- 1270 Participating Students

18 COUNTRIES

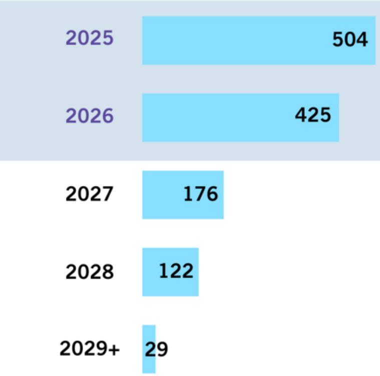
33 US STATES

95 Schools 2024 VS 124 Schools 2025

2025 MAJORS
1123 COLLEGIATE STUDENTS



GRADUATION YEAR



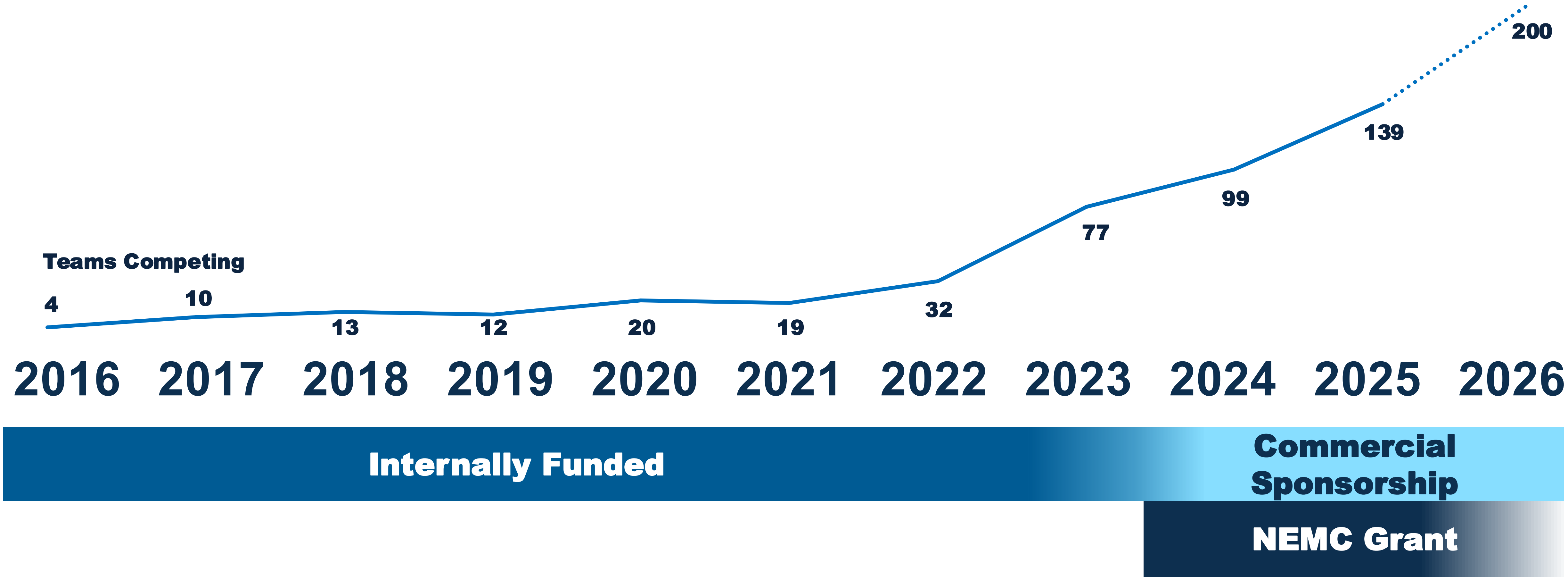
929
young professionals entering the workforce

- 🟩 77 high school students
- 🟦 862 undergraduate students
- 🟥 216 graduate students
- 🟦 45 PhD candidates

MITRE

eCTF History

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG



Win-Win of Sponsorship Program

Students get direct connection to employers

Employers can see evidence of student experience

Students gain experience on real tools and hardware

Thank you to our sponsors!

NEMC HUB

FORTINET®

CROWDSTRIKE

ROLLS
RR
ROYCE

ANALOG DEVICES

sysdig SECURE EVERY SECOND.

BCI

Rambus

IDAHO

Future of the eCTF

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

Continued growth

Professional development

Leveraging eCTF materials

Call to Action

eCTF 10
10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

Compete

High school through grad school

Professional Division

Connect

Programs and resources that could benefit from the eCTF

New applications of concept

Support

Competition sponsorship

In-kind support and partnership

eCTF 10

10 YEARS OF THE EMBEDDED CAPTURE THE FLAG

Ben Janis

btjanis@mitre.org

ectf.mitre.org

www.linkedin.com/in/benjanis/

Thank you!

Questions?

MITRE