

Houston, We Have a Problem

Analyzing the Security of Low Earth Orbit Satellites

Johannes Willbold



@jwillbold



/jwillbold

\$whoami



- Satellite & Space Systems Security
- Doctoral Student
 - Ruhr University Bochum, DE
- Visiting Researcher
 - Cyber-Defence Campus, CH
- Co-Founder of the SpaceSec Workshop

Space Odyssey

Space Odyssey: An Experimental Software Security Analysis of Satellites

Johannes Willbold*, Moritz Schloegel*[‡], Manuel Vögele*, Maximilian Gerhardt*,
Thorsten Holz[‡], Ali Abbasi[‡]

*Ruhr University Bochum, *firstname.lastname@rub.de*

[‡]CISPA Helmholtz Center for Information Security, *lastname@cispa.de*



**Distinguished
Paper Award**

Abstract—Satellites are an essential aspect of our modern society and have contributed significantly to the way we live today, most notable through modern telecommunications, global positioning, and Earth observation. In recent years, and especially in the wake of the *New Space Era*, the number of satellite deployments has seen explosive growth. Despite its critical importance, little academic research has been conducted on satellite security and, in particular, on the security of onboard firmware. This lack likely stems from by now outdated assumptions on achieving security by obscurity, effectively preventing meaningful research on satellite firmware.

In this paper, we first provide a taxonomy of threats

in 2022 [2]. The vast majority of these satellites form mega-constellations like *Starlink*, which plans to launch more than 40,000 satellites in the coming years [3].

Small satellites [4] are at the heart of this *New Space Era* as their size and the widespread use of Commercial off-the-shelf (COTS) components makes them affordable even for small institutions. Furthermore, they cover a broad spectrum of use cases ranging from commercial applications (like Earth observation, machine-to-machine communication, and Internet services) to research applications, such as technology testing, weather and earthquake forecasting, and even interplanetary missions [5]–[8].

44th IEEE Symposium on Security and Privacy (S&P)

Applications



Telecommunications



Global Positioning



Earth Observation



Research

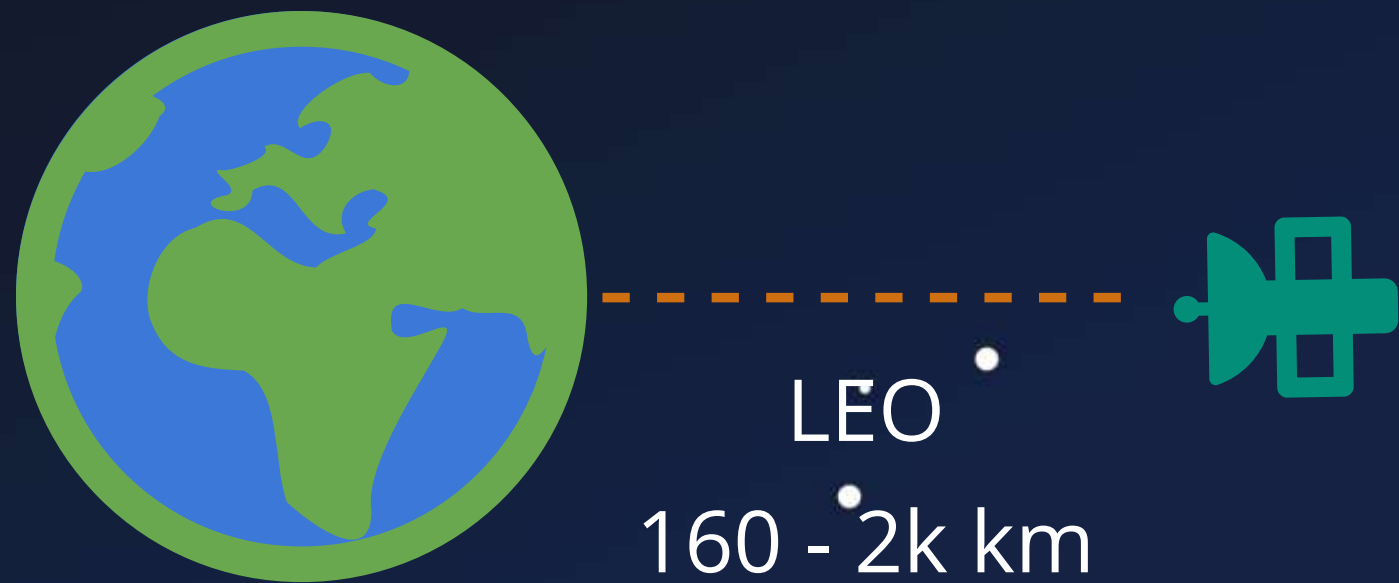


Technology Testing

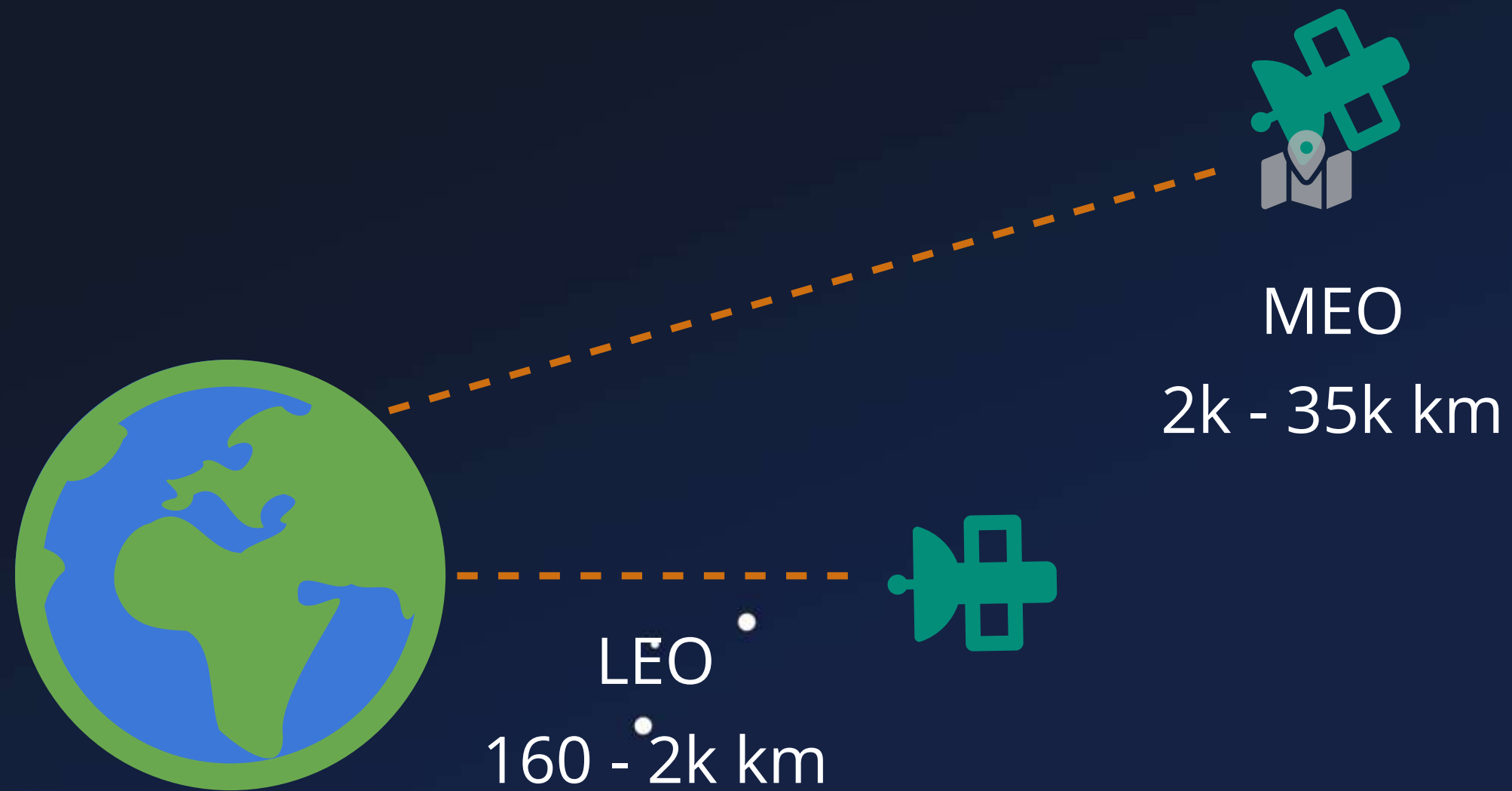
Satellite Orbits



Satellite Orbits



Satellite Orbits



Satellite Orbits



Satellite Orbits



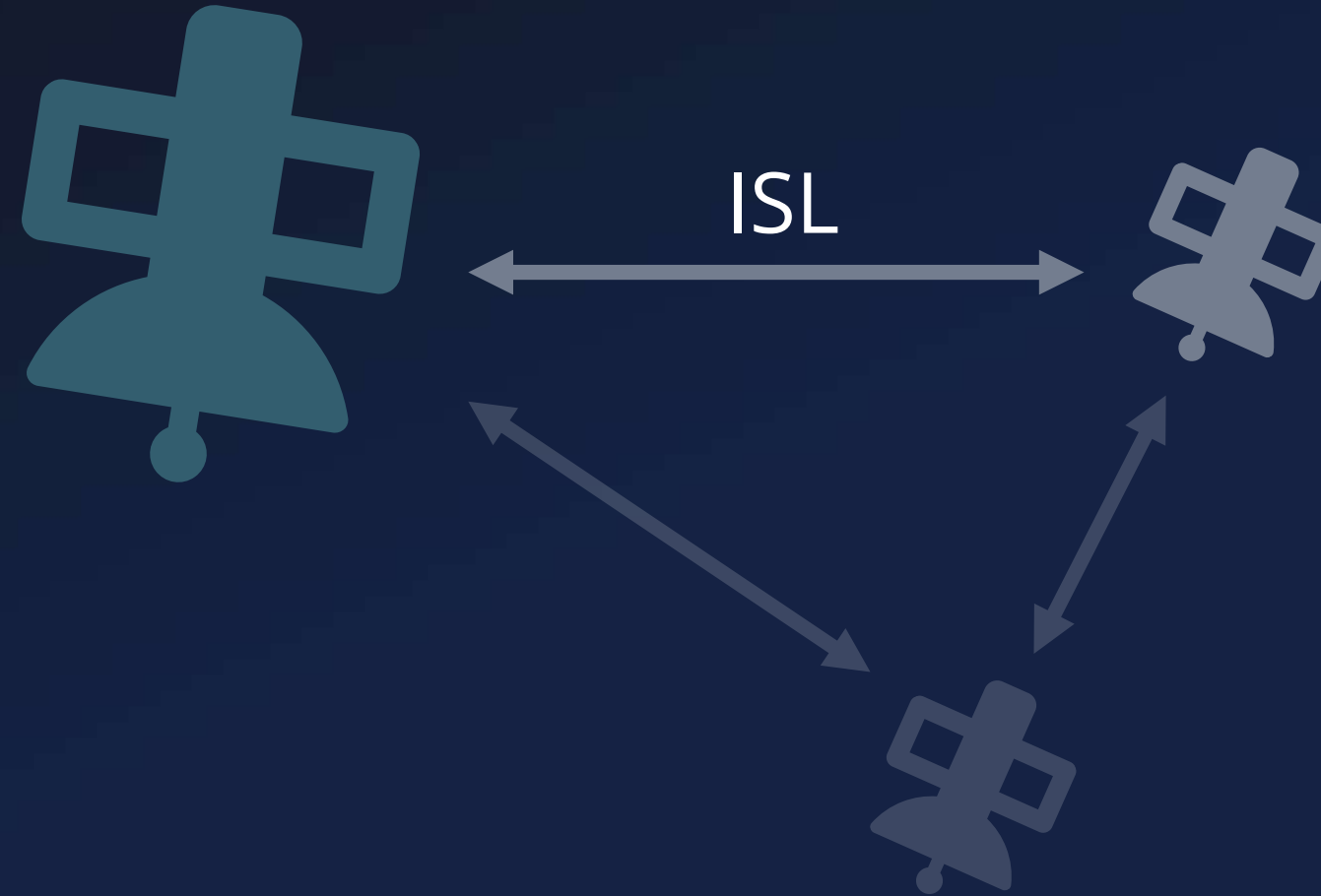
Context

Space Segment

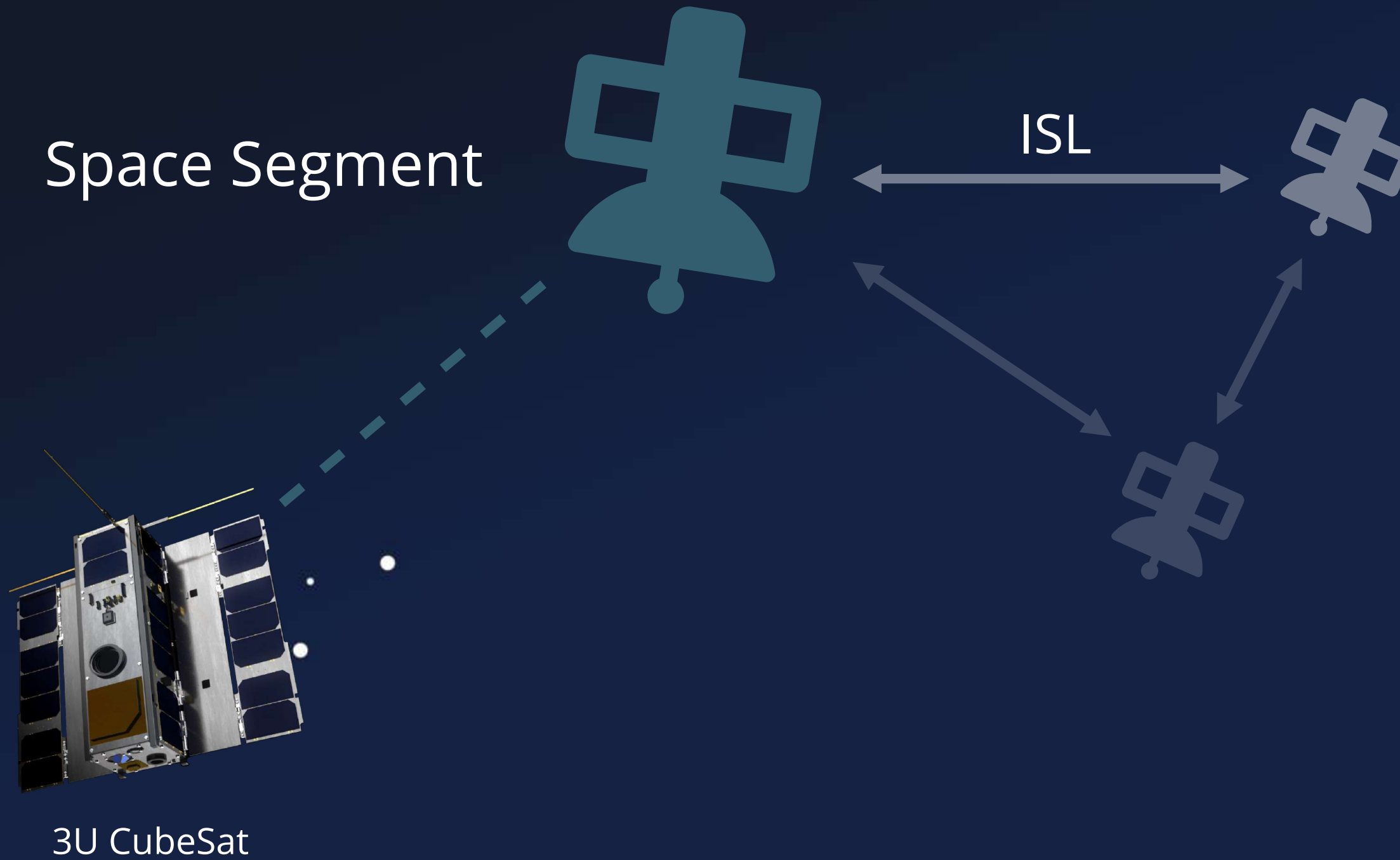


Context

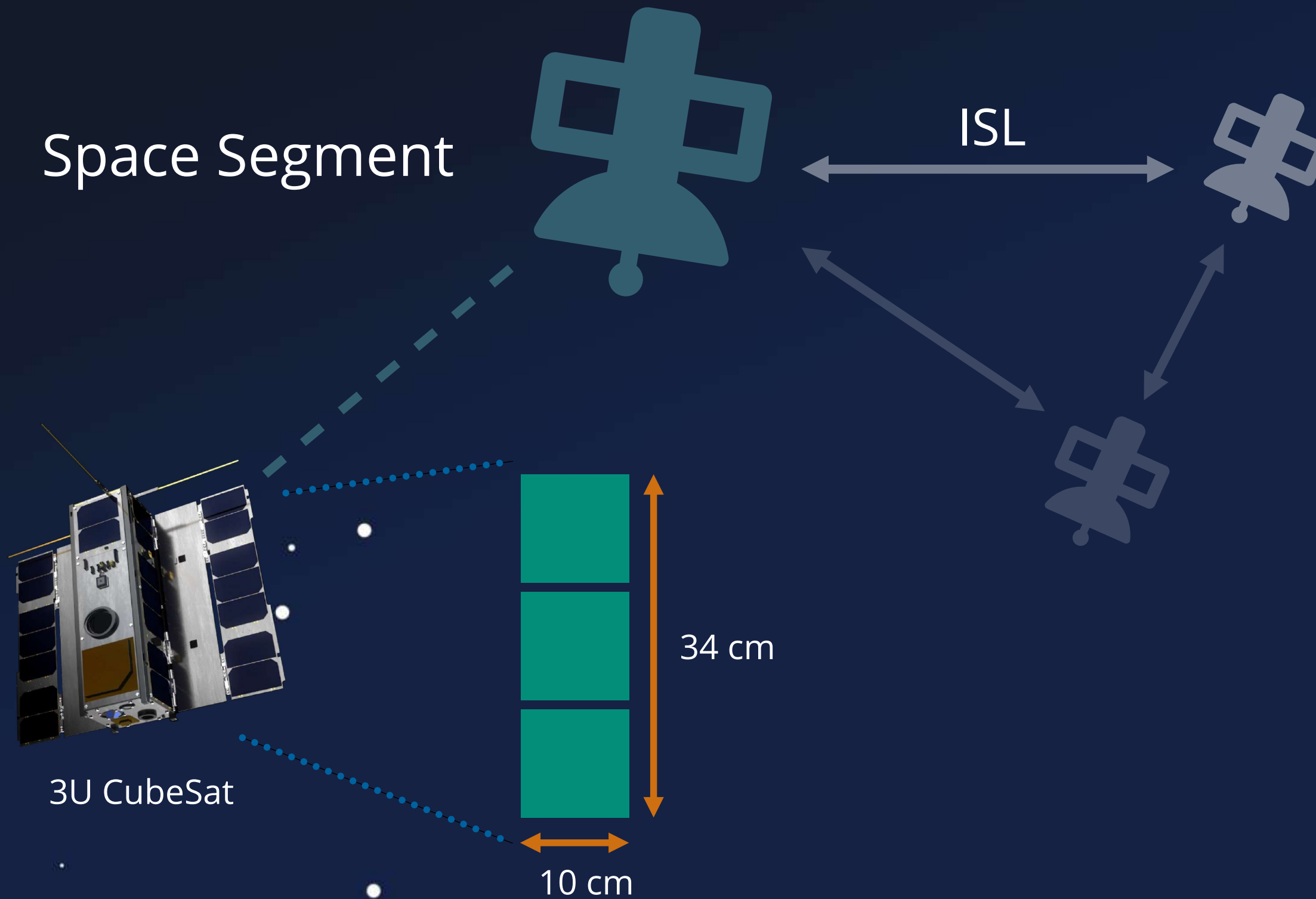
Space Segment



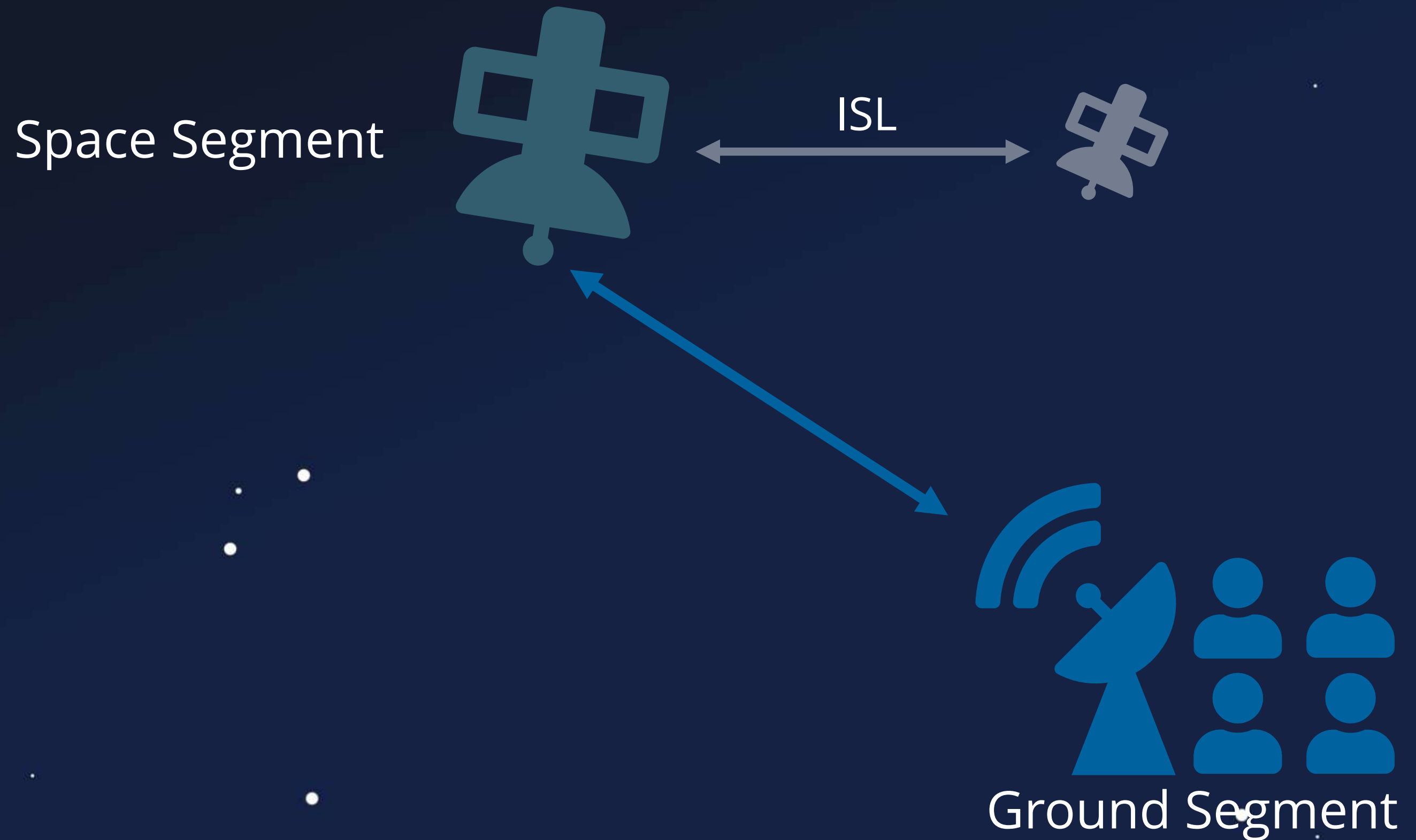
Context



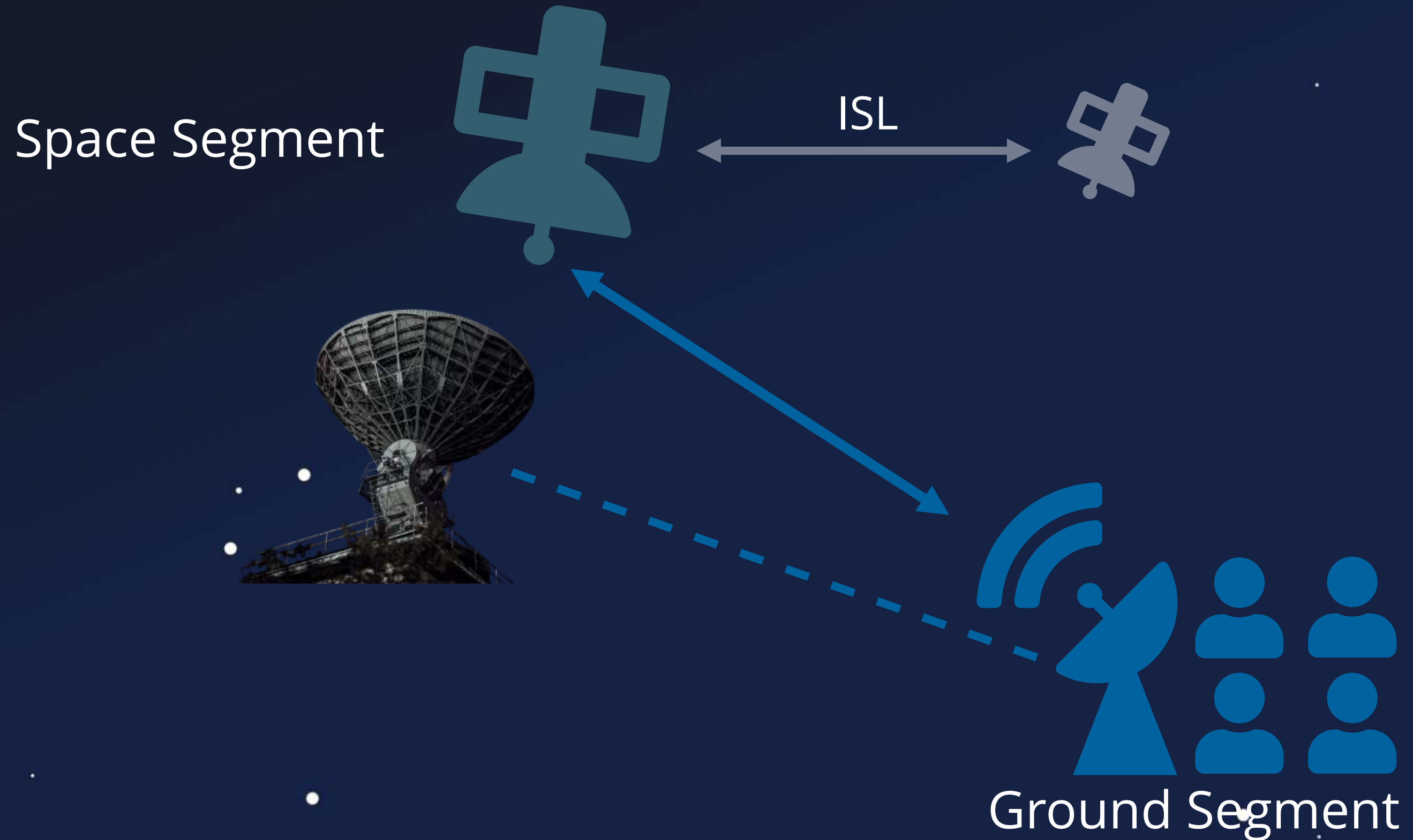
Context



Context



Context



Context

Space Segment

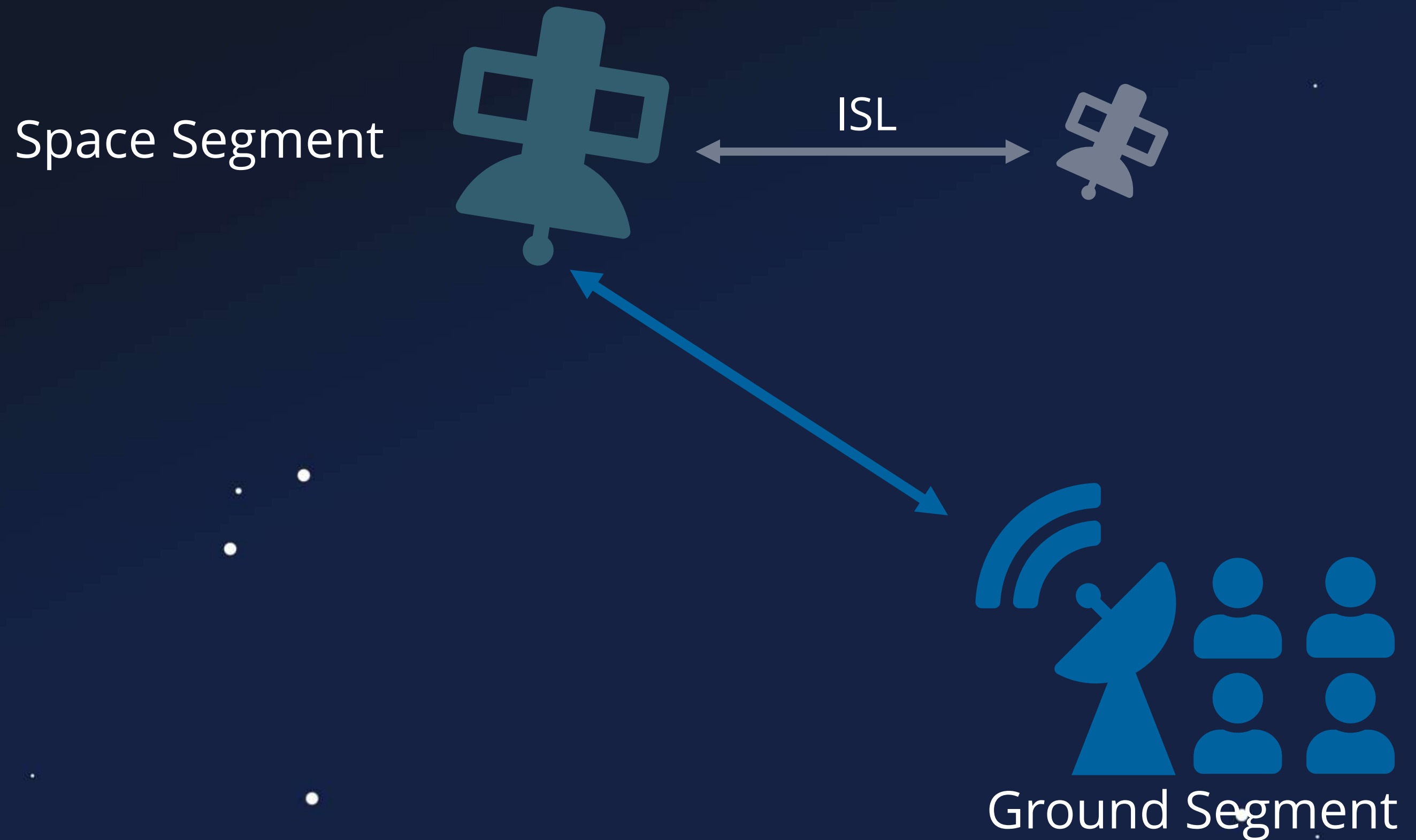


ISL

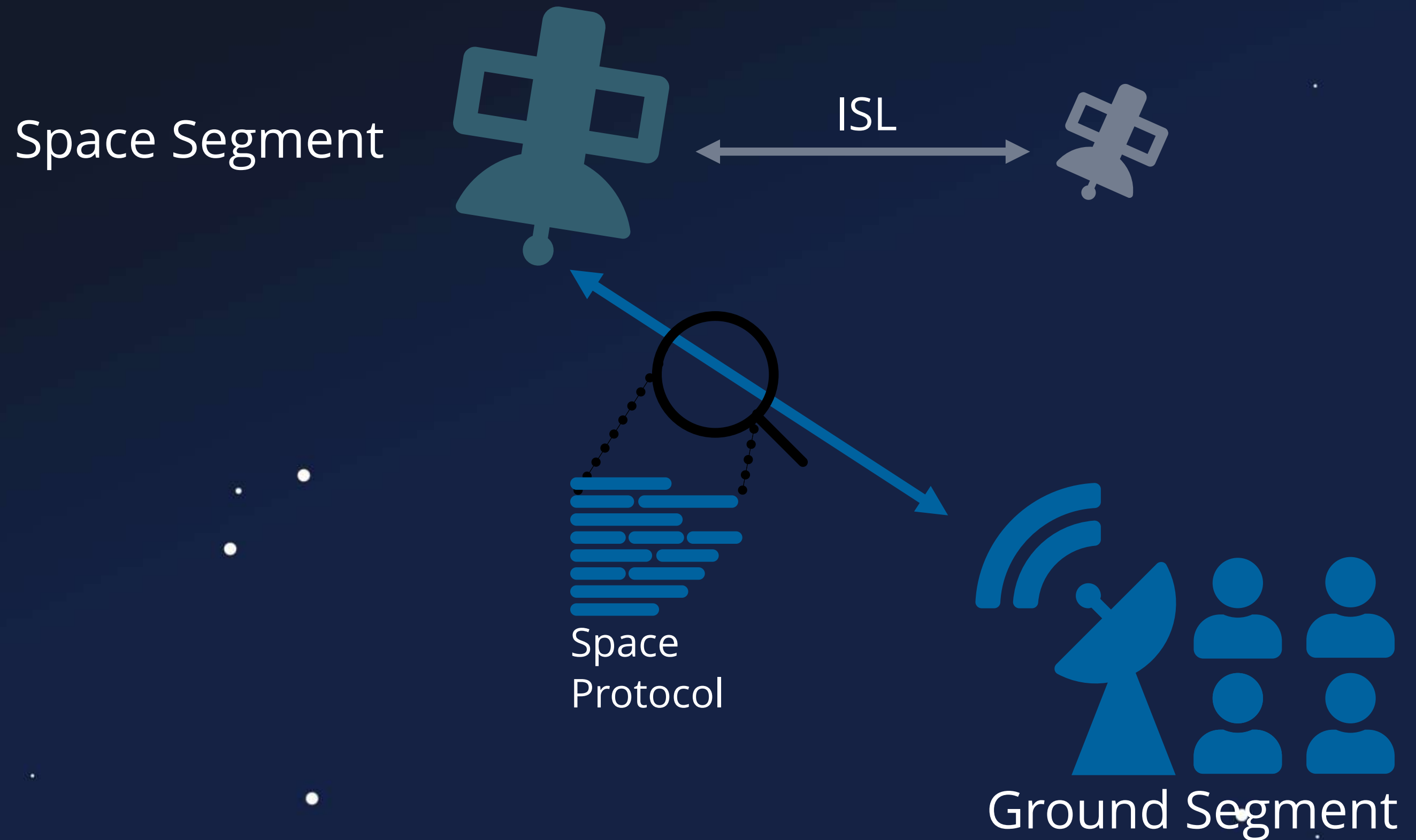


Ground Segment

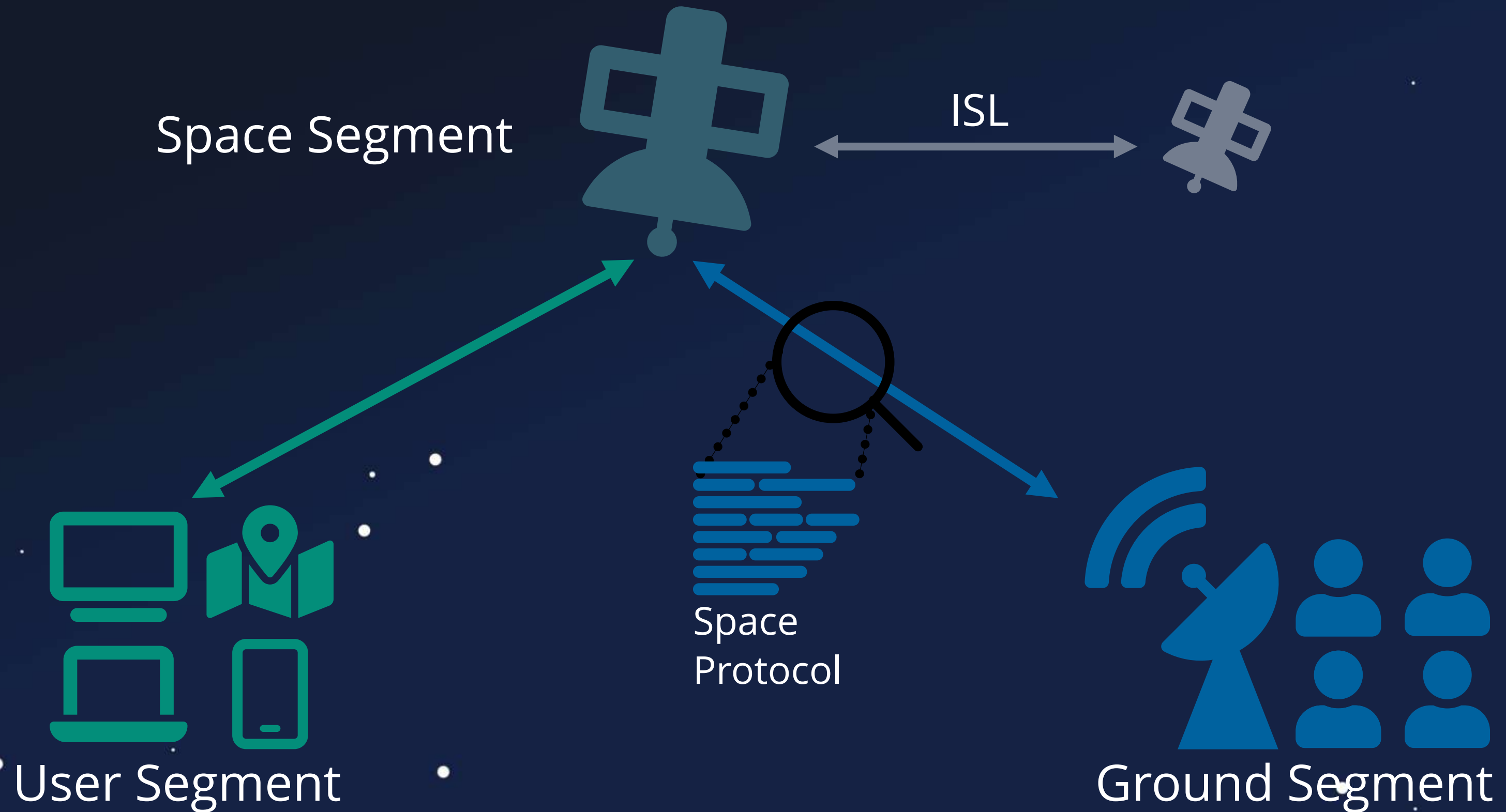
Context



Context



Context



Our Journey ...



Firmware Attacks

Our Journey ...



Firmware Attacks

Our Journey ...

System Analysis



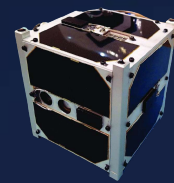
Firmware Attacks

Our Journey ...

System Analysis

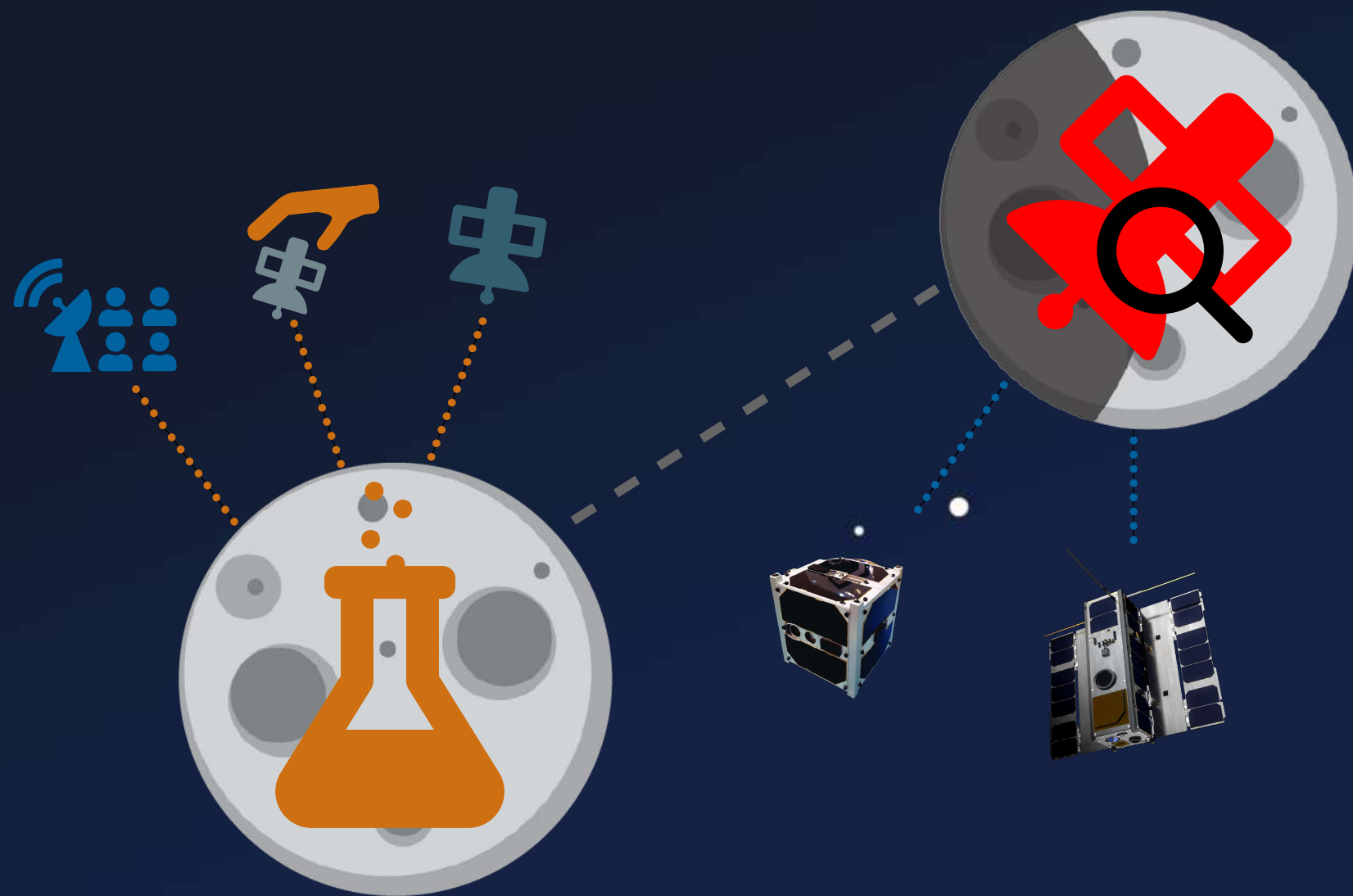


Firmware Attacks



Our Journey ...

System Analysis



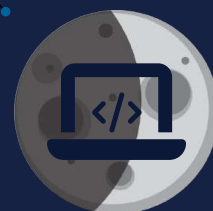
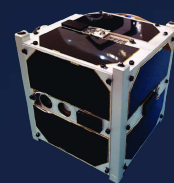
Firmware Attacks

Our Journey ...

System Analysis



Firmware Attacks



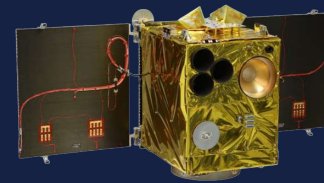
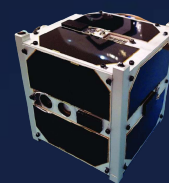
Live Demo

Our Journey ...

System Analysis



Firmware Attacks



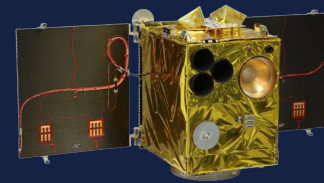
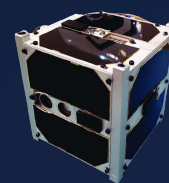
Live Demo

Our Journey ...

System Analysis



Firmware Attacks



Live Demo



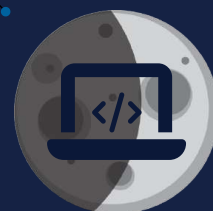
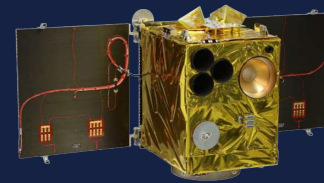
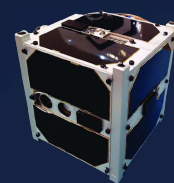
Survey

Our Journey ...

System Analysis



Firmware Attacks



Live Demo



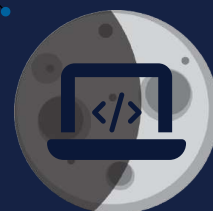
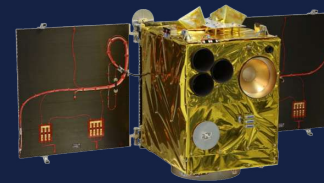
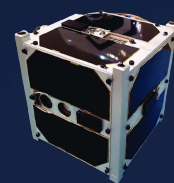
Survey

Our Journey ...

System Analysis



Firmware Attacks



Live Demo

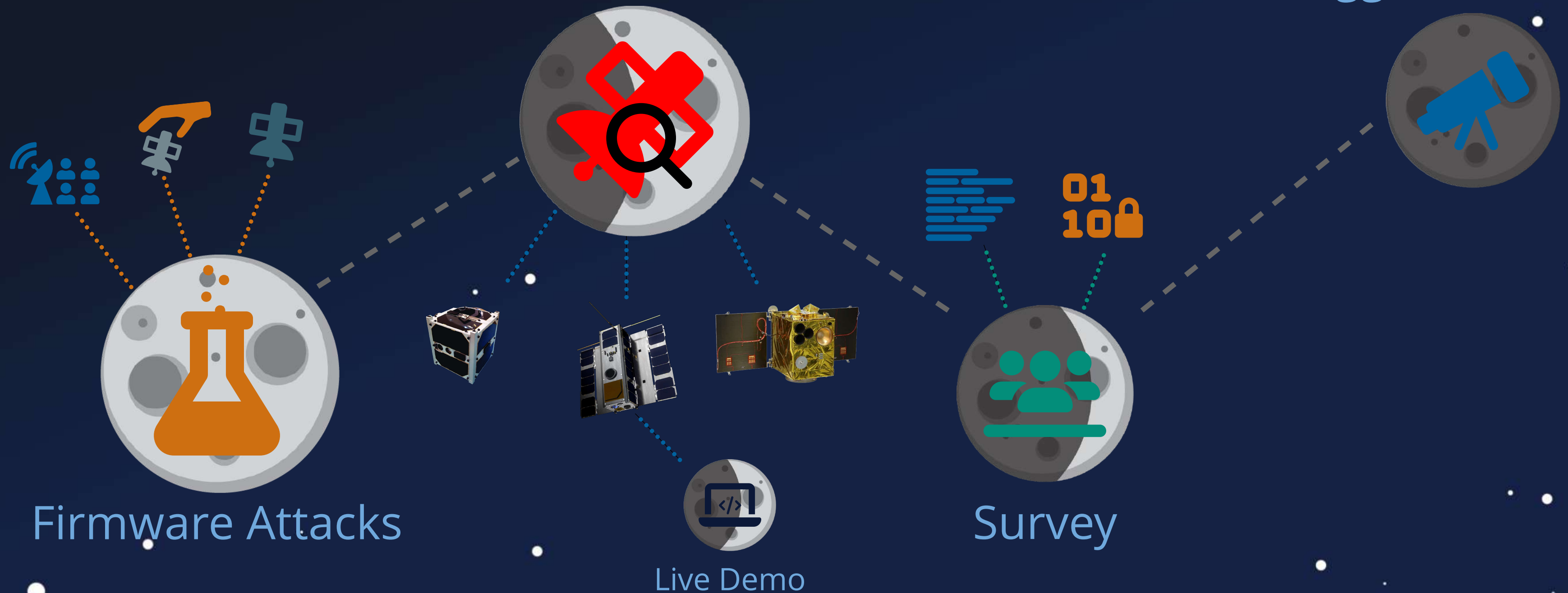


Survey

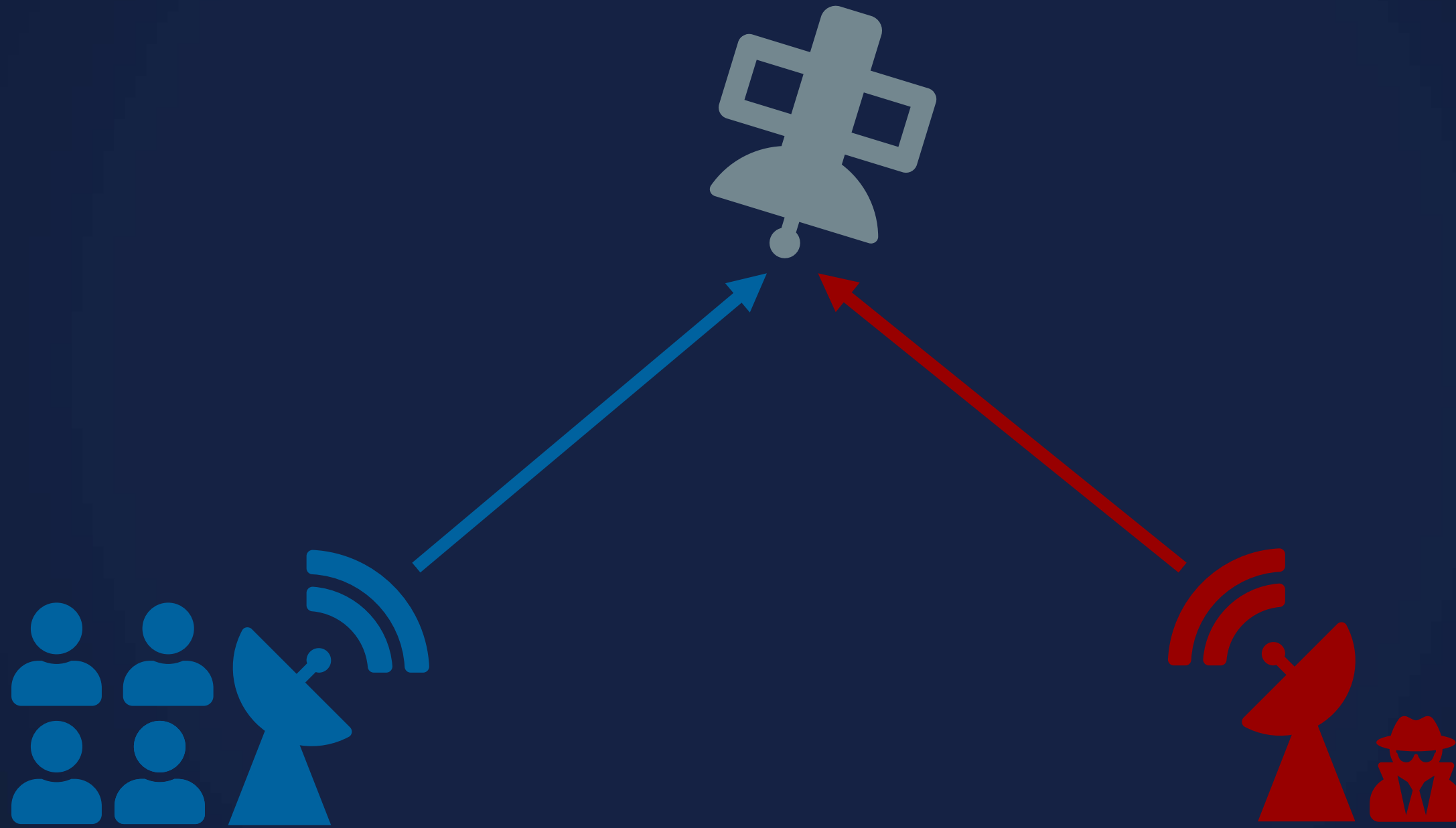
Our Journey ...

System Analysis

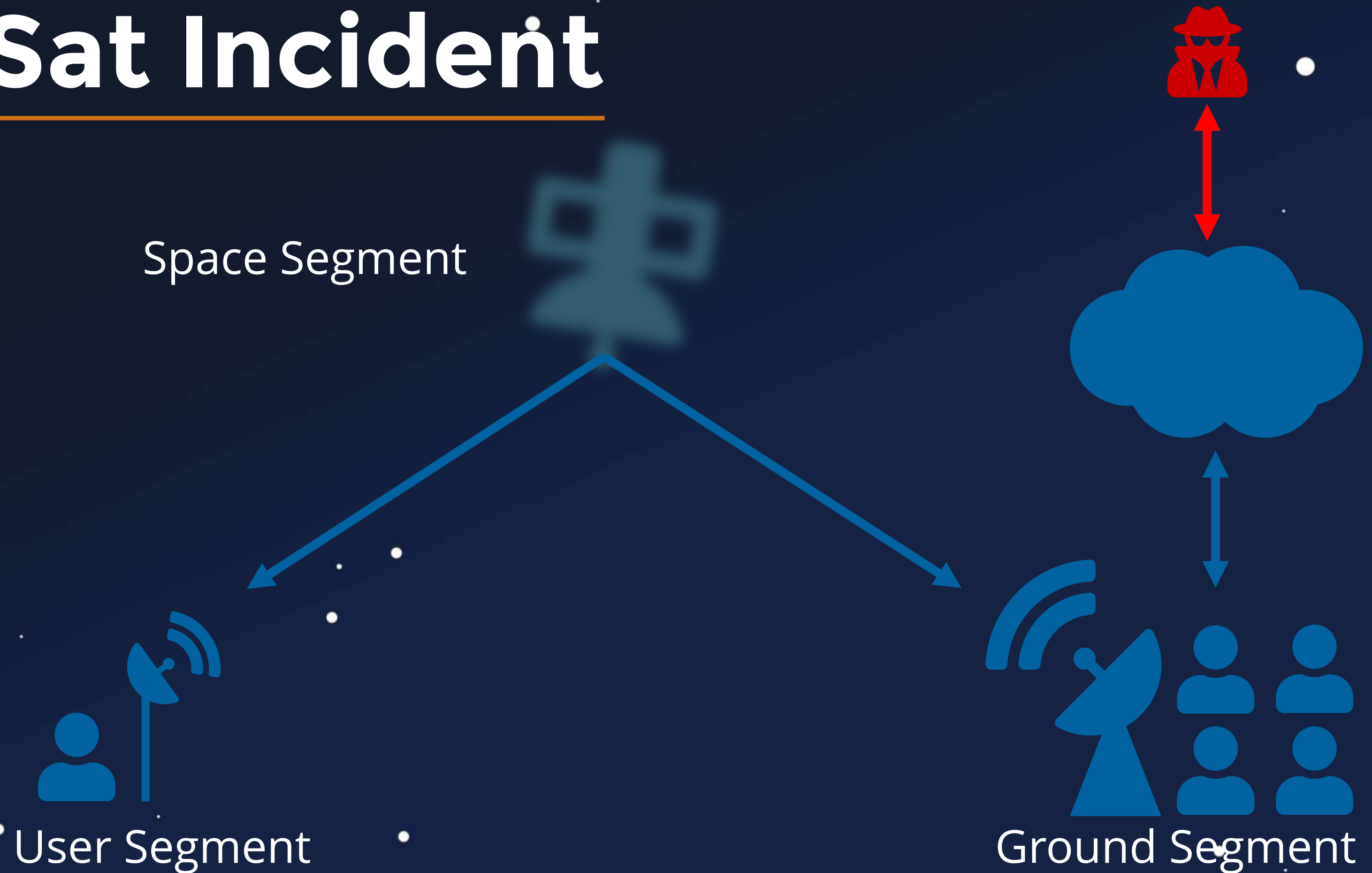
Bigger Picture



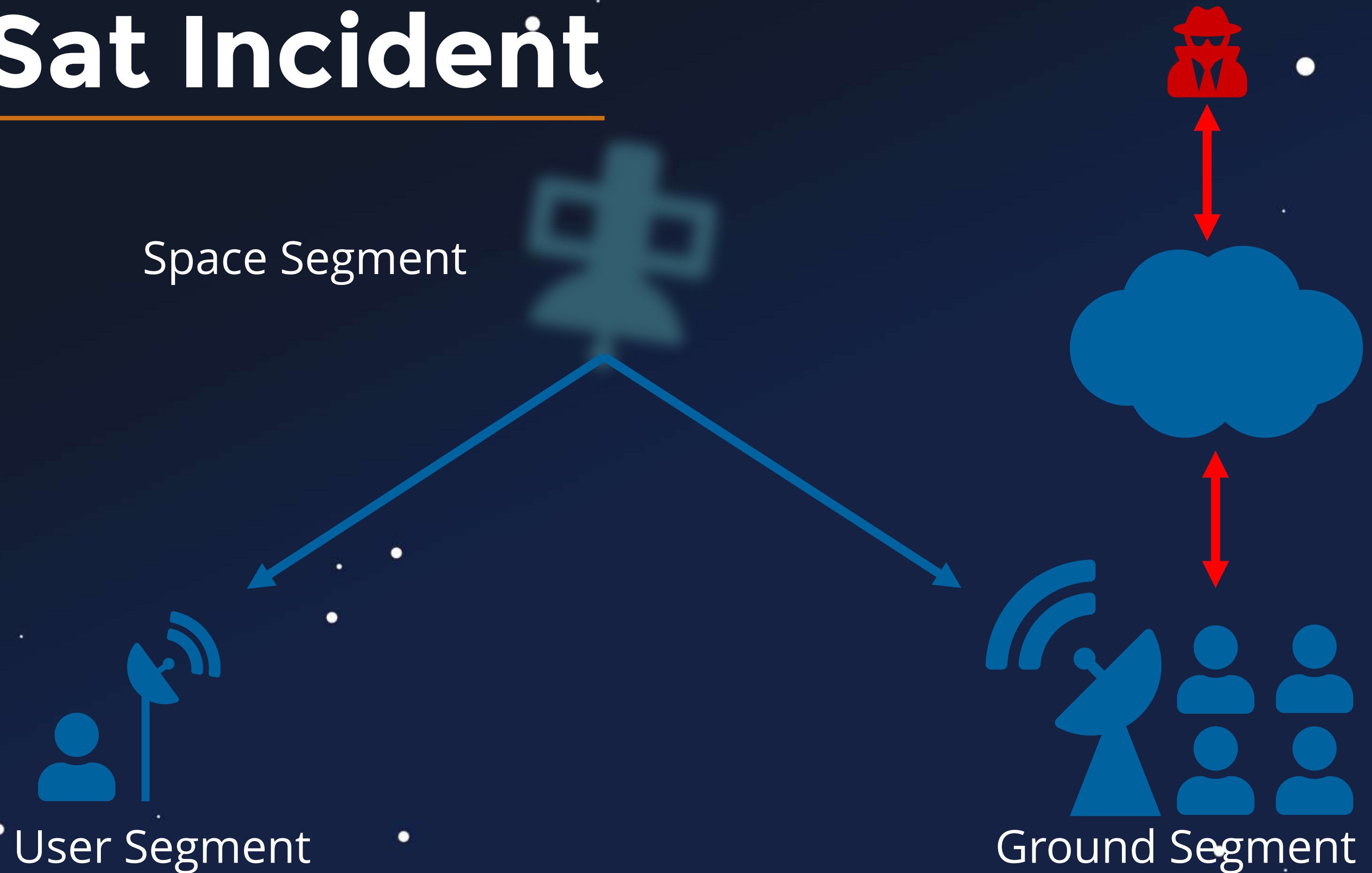
Firmware Attacks



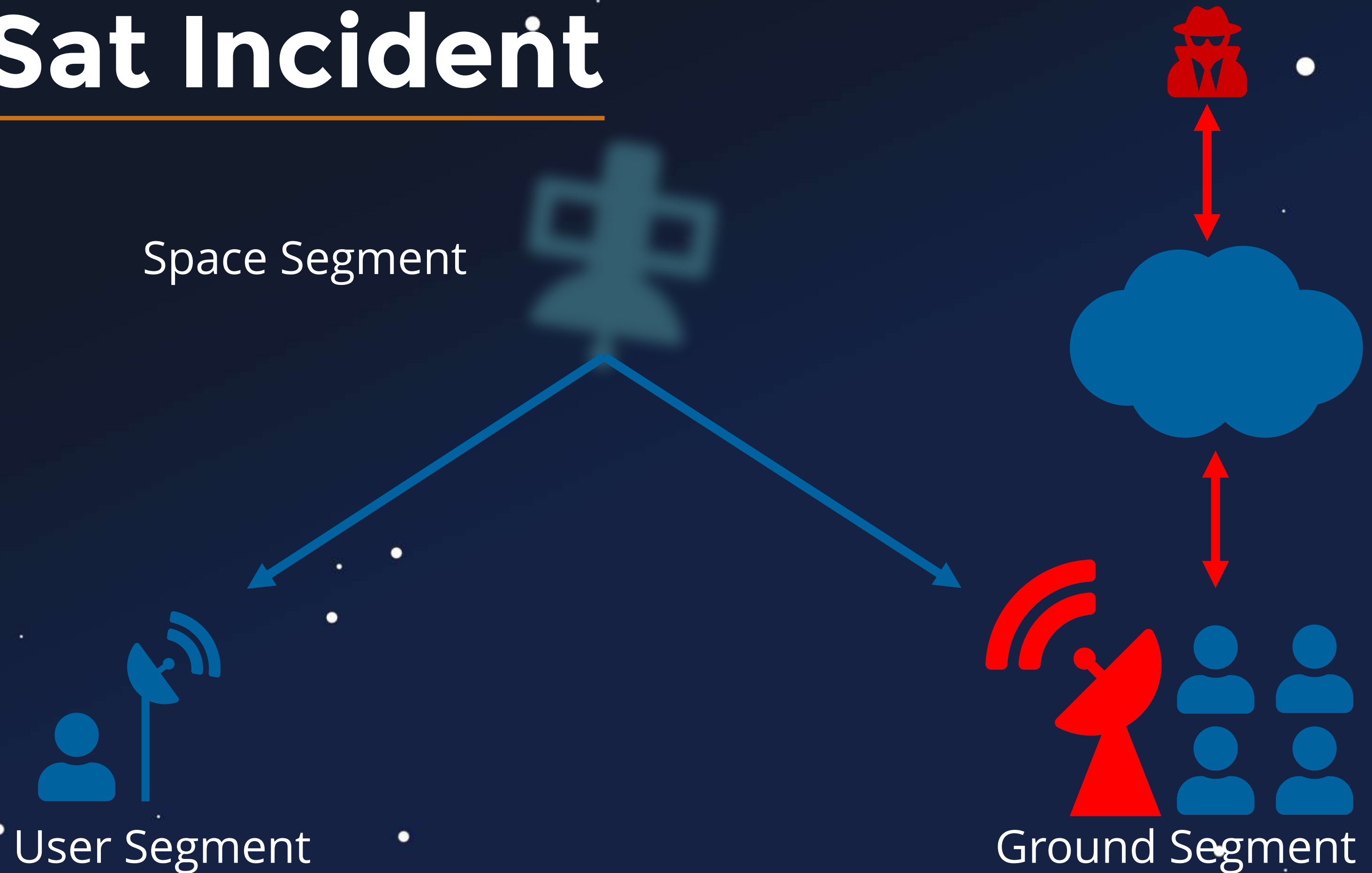
ViaSat Incident



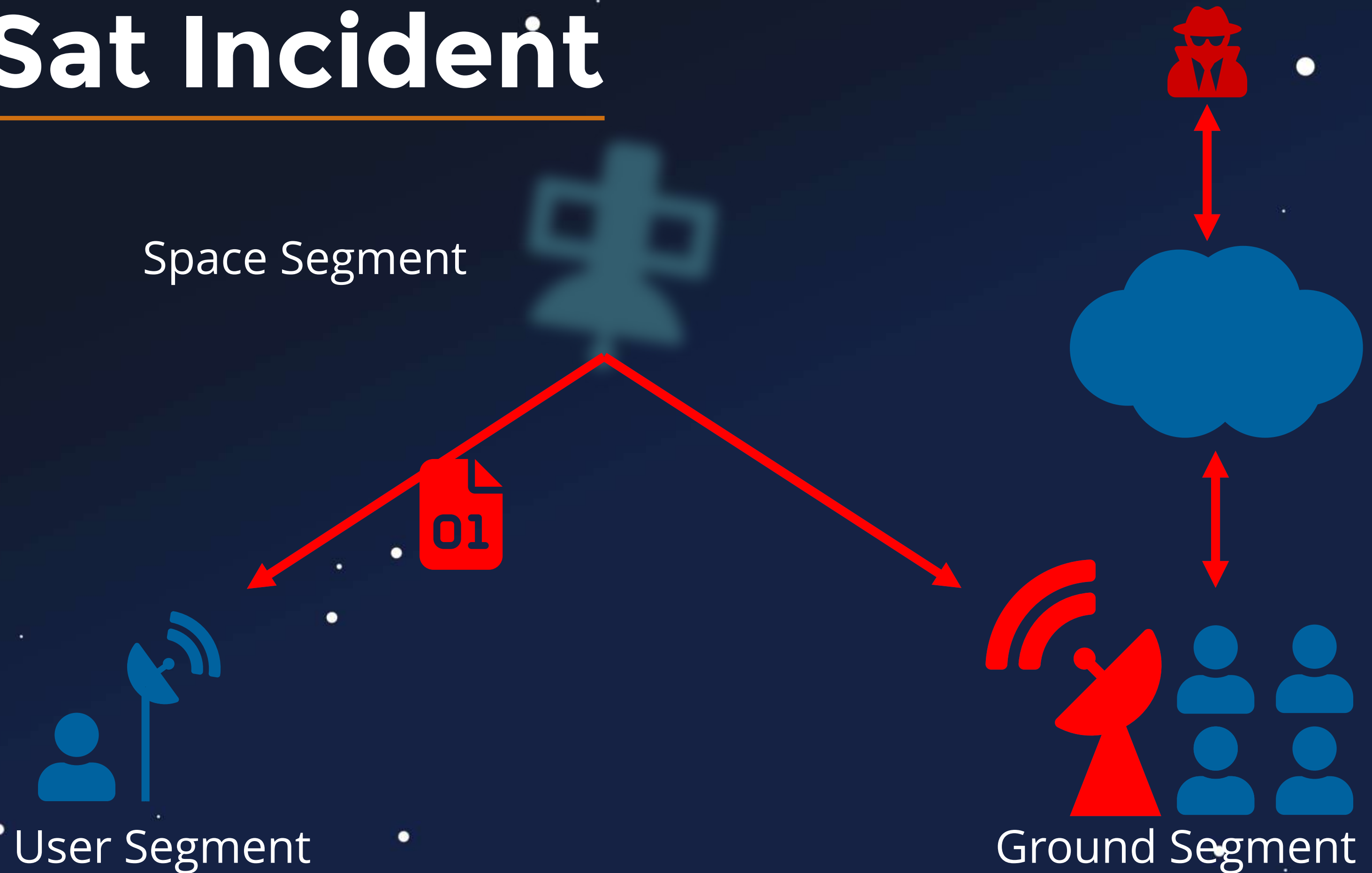
ViaSat Incident



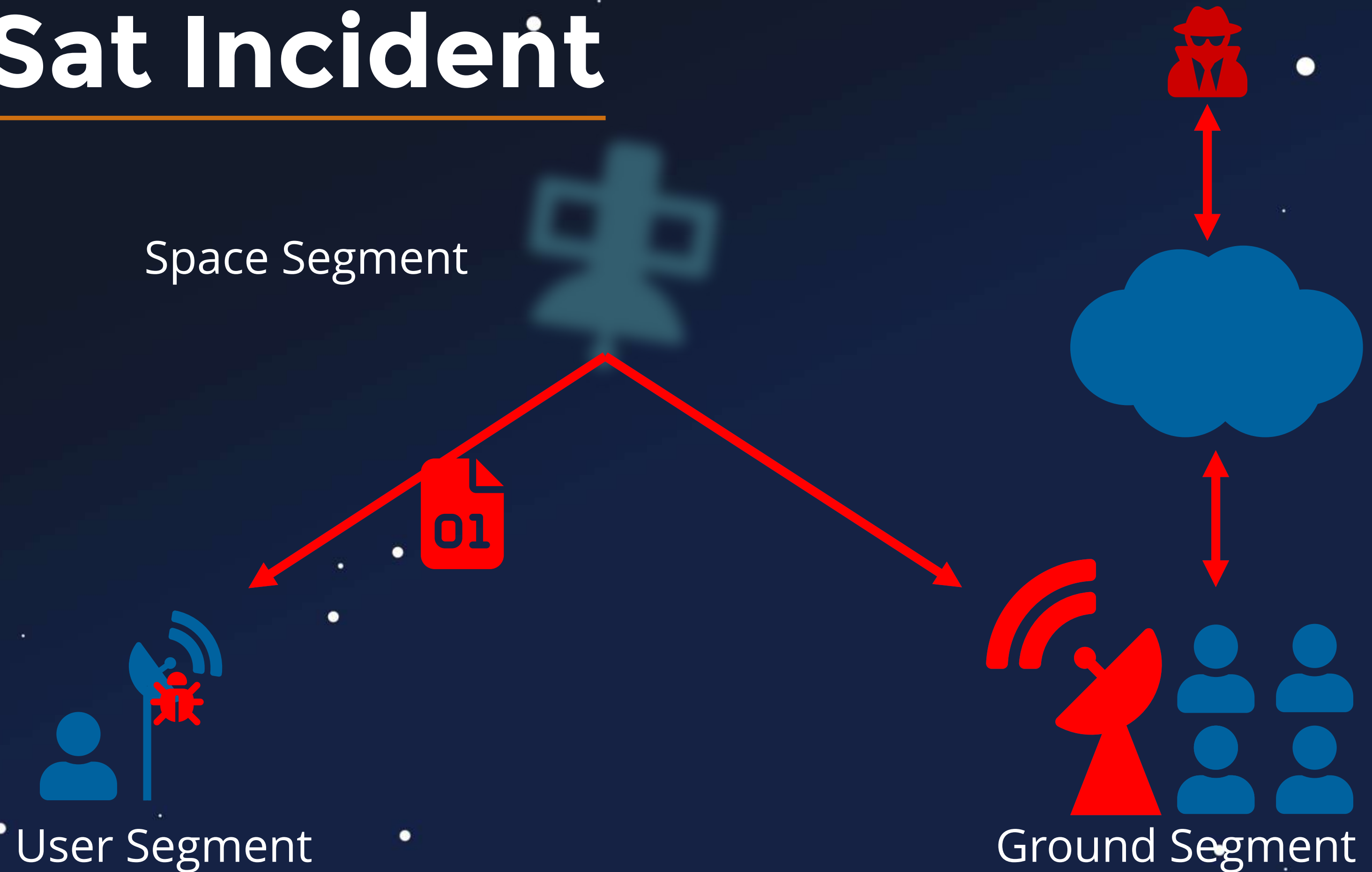
ViaSat Incident



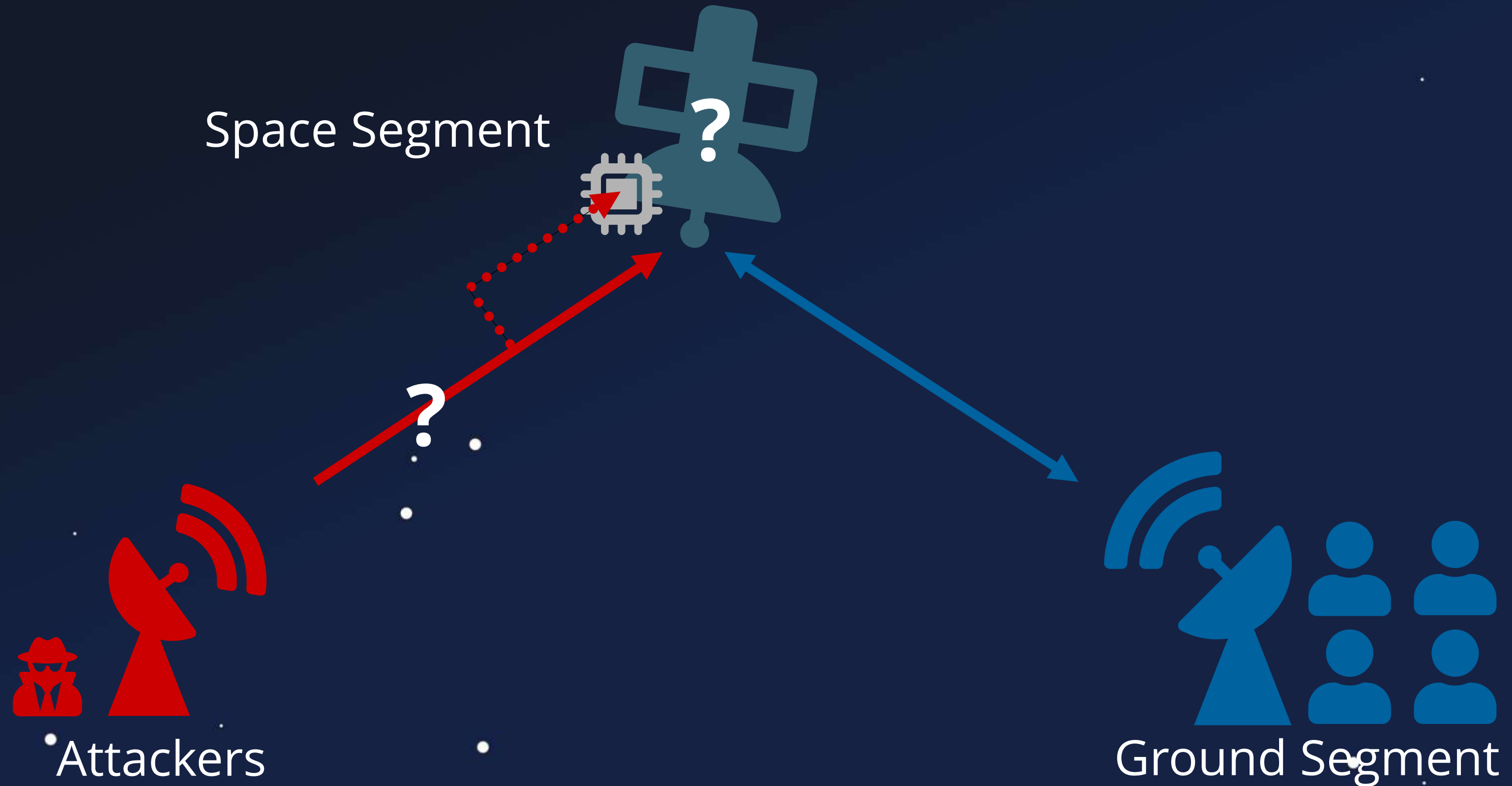
ViaSat Incident



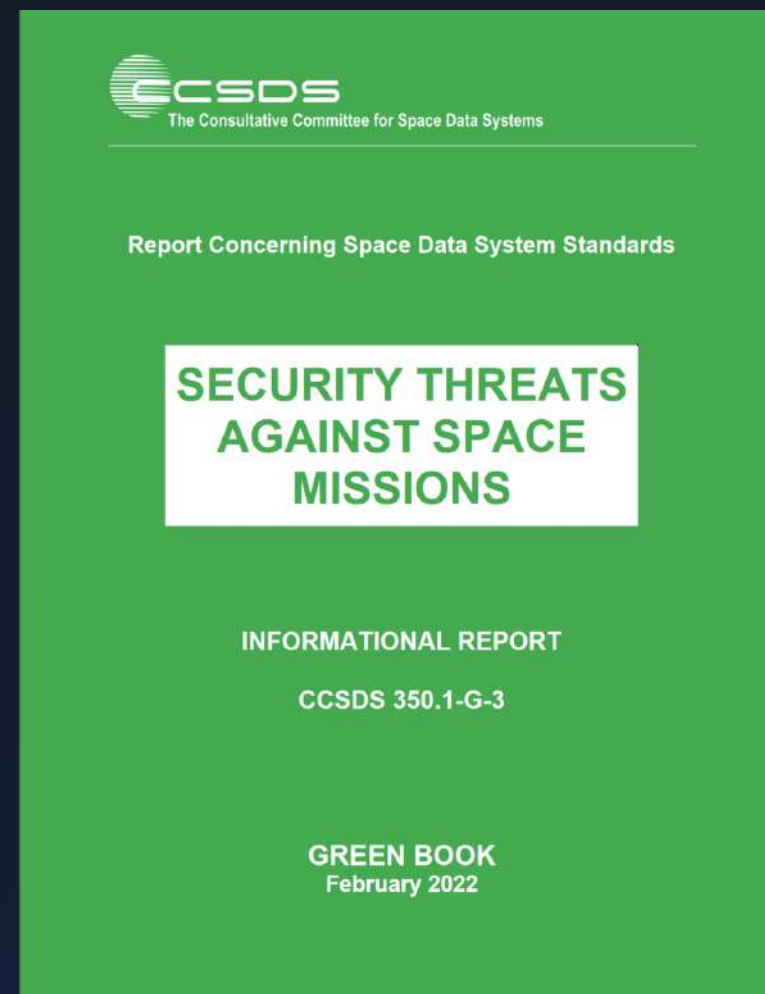
ViaSat Incident



Firmware Attacks



Not so Novel



Not so Novel



Rep

3.4.8 REPLAY

Applicable to: Space Segment, Ground Segment, Space-Link Communication.

Description: Transmissions to or from a spacecraft or between ground system computers can be intercepted, recorded, and played back at a later time.

Possible Mission Impact: If the recorded data were a command set from the ground to the spacecraft and they are re-transmitted to the spacecraft, they might be executed, potentially at a later time. If the replayed commands are not rejected, they could result in duplicate spacecraft operations, such as a maneuver of a spacecraft re-orientation, with the result that a spacecraft is in an unintended orientation (e.g., tumbling, antenna pointed in the wrong direction, solar arrays pointed away from the sun, the reset of critical onboard parameters).

3.4.9 SOFTWARE THREATS

Applicable to: Space Segment, Ground Segment.

Description: Users, system operators, and programmers often make mistakes that can result in security problems. Users or administrators can install unauthorized or unvetted software that might contain bugs, viruses, or spyware, which could result in system instability. System operators might misconfigure a system resulting in security weaknesses. Programmers may introduce logic or implementation errors that could result in system vulnerabilities, or instability/reliability. Weaknesses may be discovered after a mission is operational, which external threat agents might attempt to exploit to inject instructions, software, or configuration changes.

Possible Mission Impact: Software threats could result in loss of data and safety issues, loss of spacecraft control, unauthorized spacecraft control, or loss of mission.

3.4.10 UNAUTHORIZED ACCESS

Applicable to: Space Segment, Ground Segment.

Description: Access control policies based on strong authentication provide a means by which only authorized entities are allowed to perform system actions, while all others are prohibited.

Possible Mission Impact: An access control breach would allow an unauthorized entity to take control of a ground system or a ground system network, shut down a ground system, upload unauthorized commands to a spacecraft, execute unauthorized commands aboard a crewed mission, obtain unauthorized data, contaminate archived data, or completely shut down a mission. If weak access controls are in place, unauthorized access might be obtained. Interception of data might result in unauthorized access because identities, identifiers, or passwords might be obtained. Social engineering could be employed to obtain identities, identifiers, passwords, or other technical details permitting unauthorized access.

Not so Novel



Rep

CCSDS REPORT CONCERNING SECURITY THREATS AGAINST SPACE MISSIONS

3.4.8 REPLAY

Applicable to: Space Segment, Ground Segment, Space-Link Communication.

Description: Transmissions to or from a spacecraft or between ground system computers can be intercepted, recorded, and played back at a later time.

Possible Mission Impact: If the recorded data were a command set from the ground to the spacecraft and they are re-transmitted to the spacecraft, they might be executed, potentially at a later time. If the replayed commands are not rejected, they could result in duplicate spacecraft operations, such as a maneuver or a spacecraft re-orientation, with the result that a spacecraft is in an unintended orientation (e.g., tumbling, antennas pointed in the wrong direction, solar arrays pointed away from the sun, the reset of critical onboard parameters).

3.4.9 SOFTWARE THREATS

Applicable to: Space Segment, Ground Segment.

Description: Users, system operators, and programmers often make mistakes that can result in security problems. Users or administrators can install unauthorized or unvetted software that might contain bugs, viruses, or spyware, which could result in system instability. System operators might misconfigure a system resulting in security weaknesses. Programmers may introduce logic or implementation errors that could result in system vulnerabilities, or instability/reliability. Weaknesses may be discovered after a mission is operational, which external threat agents might attempt to exploit to inject instructions, software, or configuration changes.

Possible Mission Impact: Software threats could result in loss of data and safety issues, loss of spacecraft control, unauthorized spacecraft control, or loss of mission.

3.4.10 UNAUTHORIZED ACCESS

Applicable to: Space Segment, Ground Segment.

Description: Access control policies based on strong authentication provide a means by which only authorized entities are allowed to perform system actions, while all others are prohibited.

Possible Mission Impact: An access control breach would allow an unauthorized entity to take control of a ground system or a ground system network, shut down a ground system, upload unauthorized commands to a spacecraft, execute unauthorized commands aboard a crewed mission, obtain unauthorized data, contaminate archived data, or completely shut down a mission. If weak access controls are in place, unauthorized access might be obtained. Interception of data might result in unauthorized access because identities, identifiers, or passwords might be obtained. Social engineering could be employed to obtain identities, identifiers, passwords, or other technical details permitting unauthorized access.

CCSDS 350.1-G-3

Page 3-8

February 2022

MARCH 2020

A REPORT OF
THE CSIS
AEROSPACE
SECURITY
PROJECT

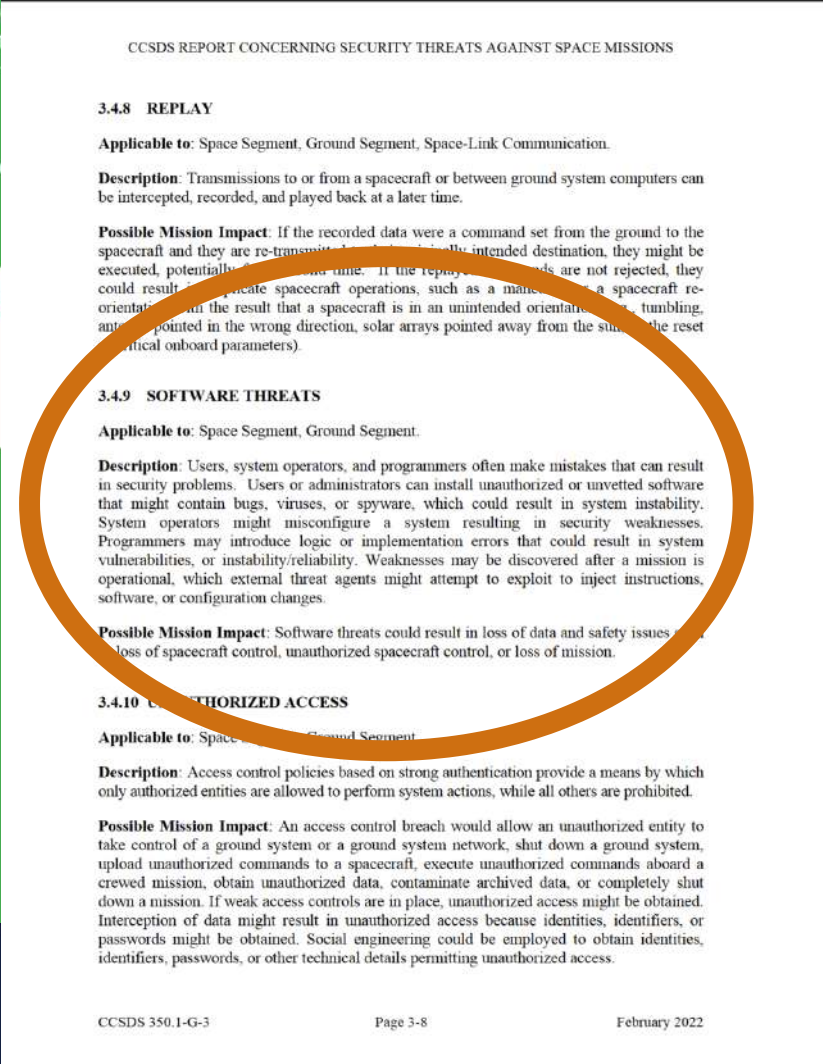
SPACE THREAT ASSESSMENT 2020

Authors
TODD HARRISON
KAITLYN JOHNSON
THOMAS G. ROBERTS
TYLER WAY
MAKENA YOUNG

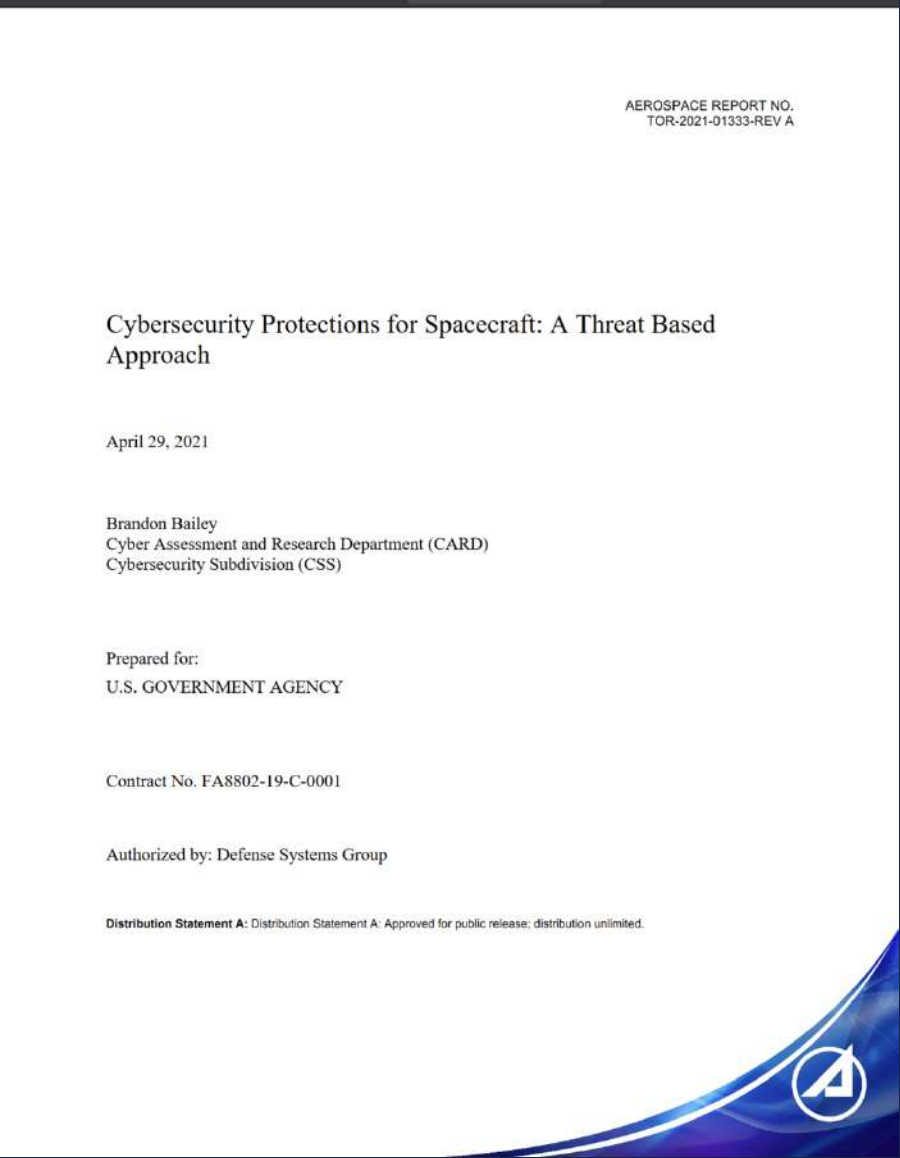
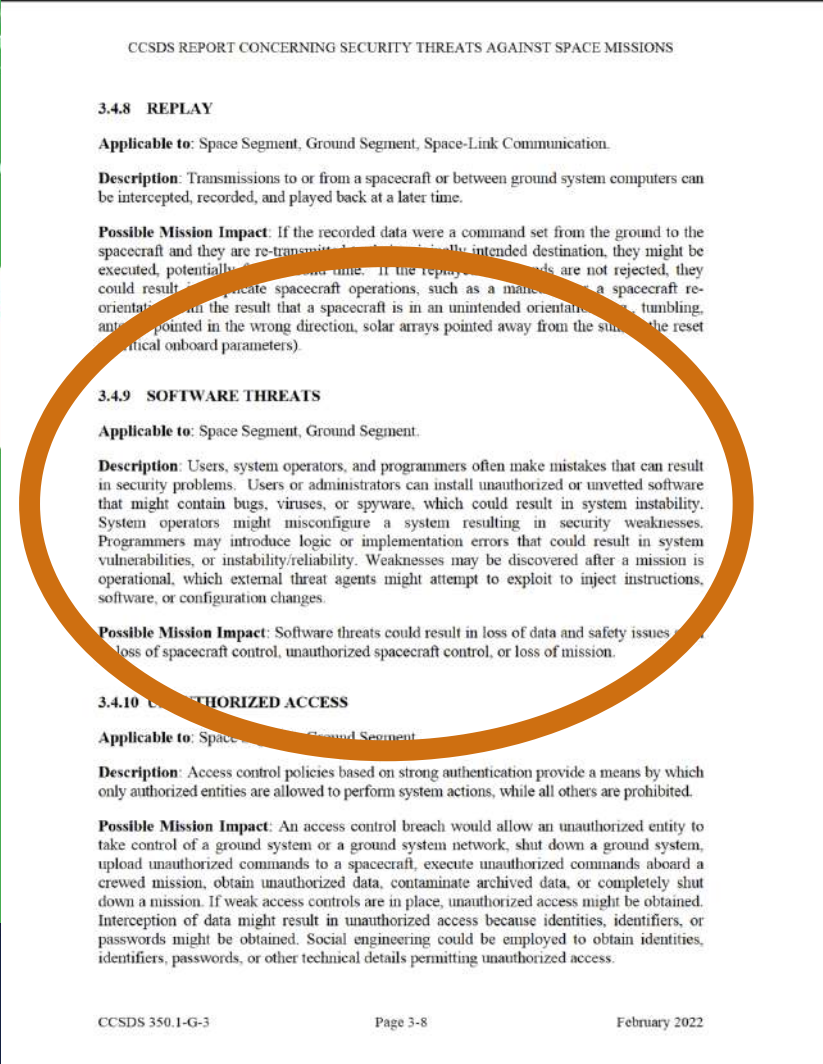
Foreword
MARTIN C. FAGA

CSIS | CENTER FOR STRATEGIC
INTERNATIONAL STUDIES

Not so Novel



Not so Novel



Outdated Assumptions



Myth of Inaccessibility



\$\$\$ → \$

Affordable
Ground Stations

Myth of Inaccessibility



\$\$\$ → \$

Affordable
Ground Stations



Ground Station as a Service
GSaaS

Myth of Inaccessibility

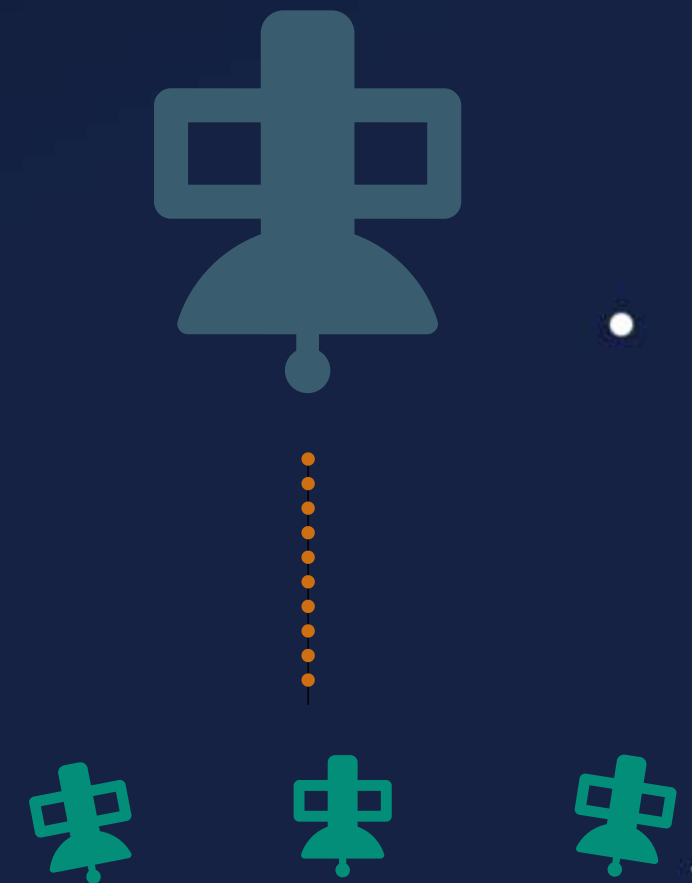


\$\$\$ → \$

Affordable
Ground Stations



Ground Station as a Service
GSaaS



More Satellites
GEO → LEO

Security by Obscurity

// *No Insights \Leftrightarrow No Attacker*

Security by Obscurity

// ~~No Insights \leftrightarrow No Attacker~~

Security by Obscurity

// ~~No Insights \leftrightarrow No Attacker~~



More Developers
More People Involved

Security by Obscurity

// ~~No Insights \leftrightarrow No Attacker~~



More Developers
More People Involved



Commercial off-the-Shelf
(COTS)
Components

Security by Obscurity

// ~~No Insights~~ \leftrightarrow ~~No Attacker~~



More Developers
More People Involved



Commercial off-the-Shelf
(COTS)
Components



Higher Stakes
Critical Infrastructure

Attacker Goals



Denial of Service

Attacker Goals



Denial of Service



Malicious Data
Interaction

Attacker Goals



Denial of Service



Seizure of Control



Malicious Data
Interaction

Attacker Goals



Denial of Service



Seizure of Control



Malicious Data Interaction

Attacker Goals



Seizure of Control

Attacker Goals

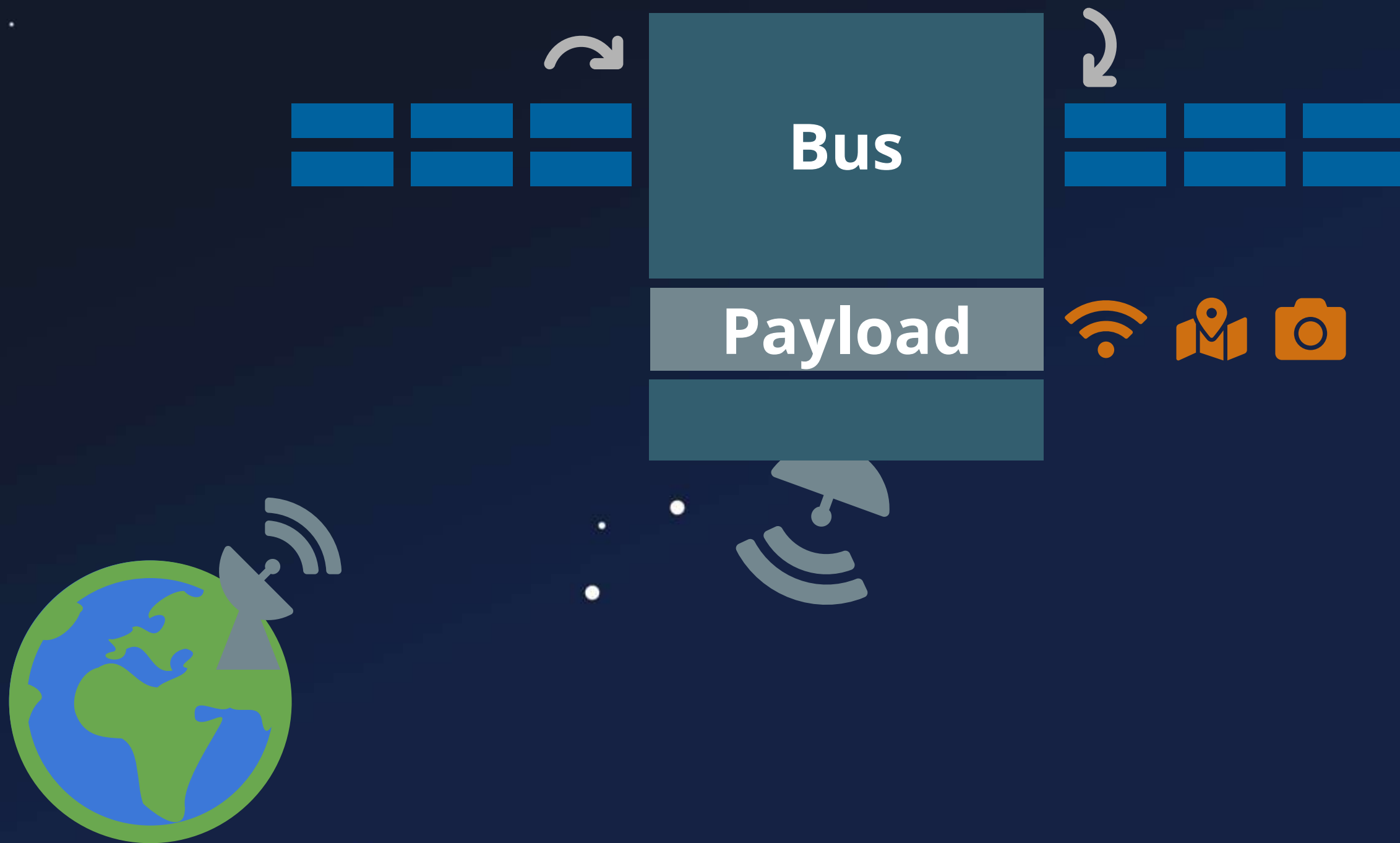


Seizure of Control

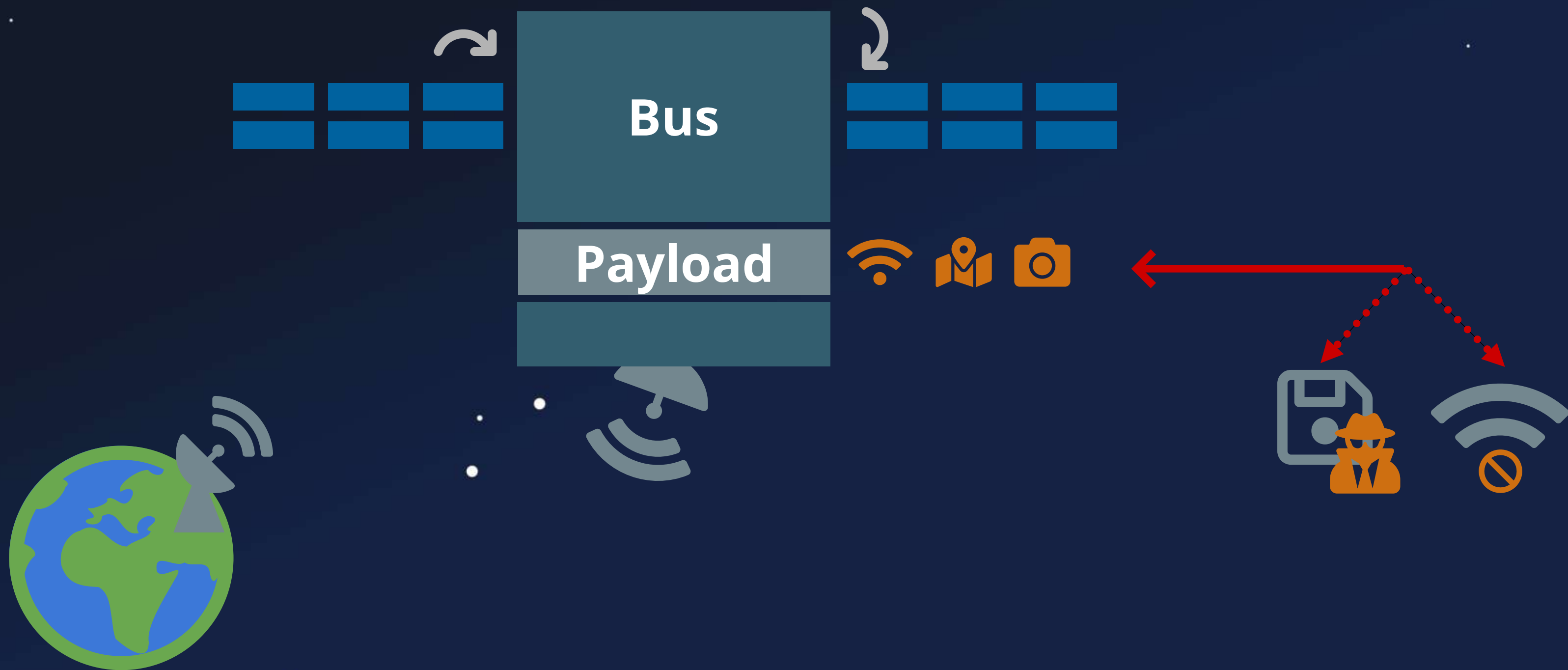
Components



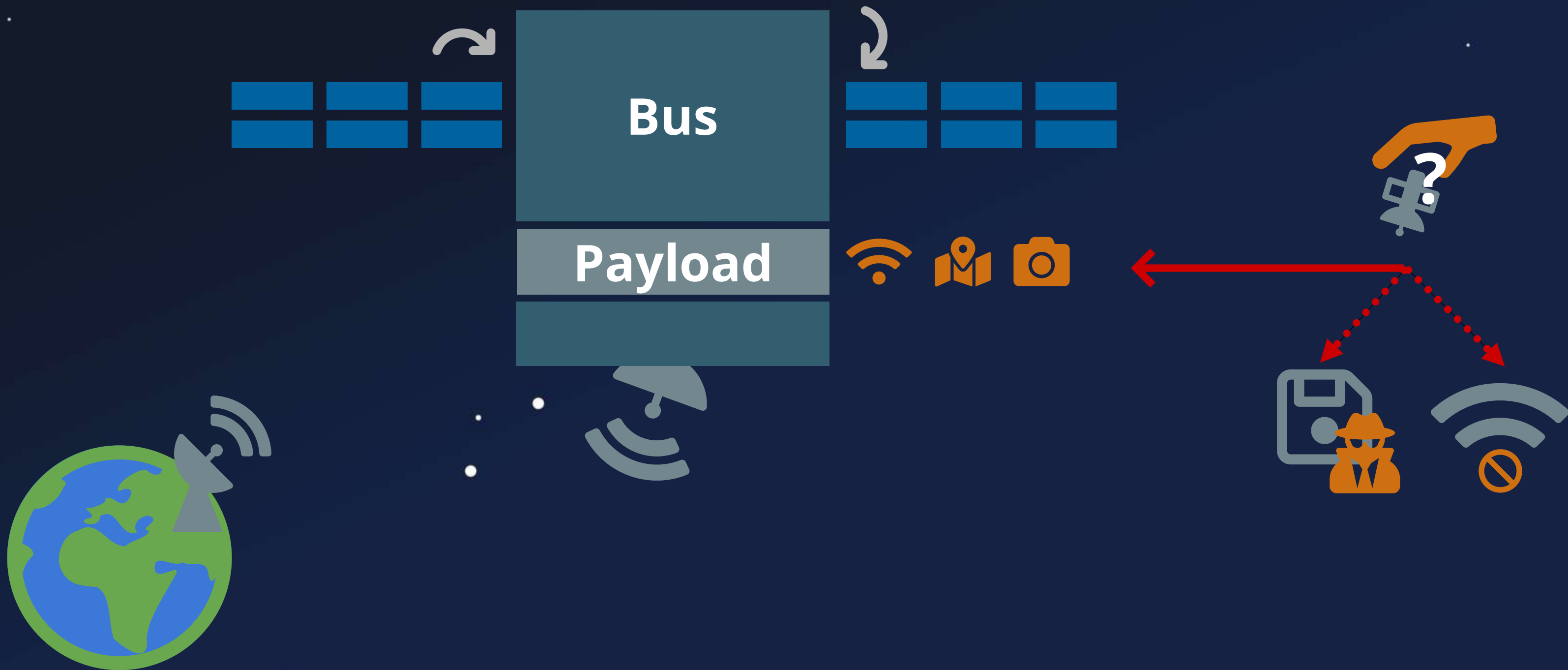
Components



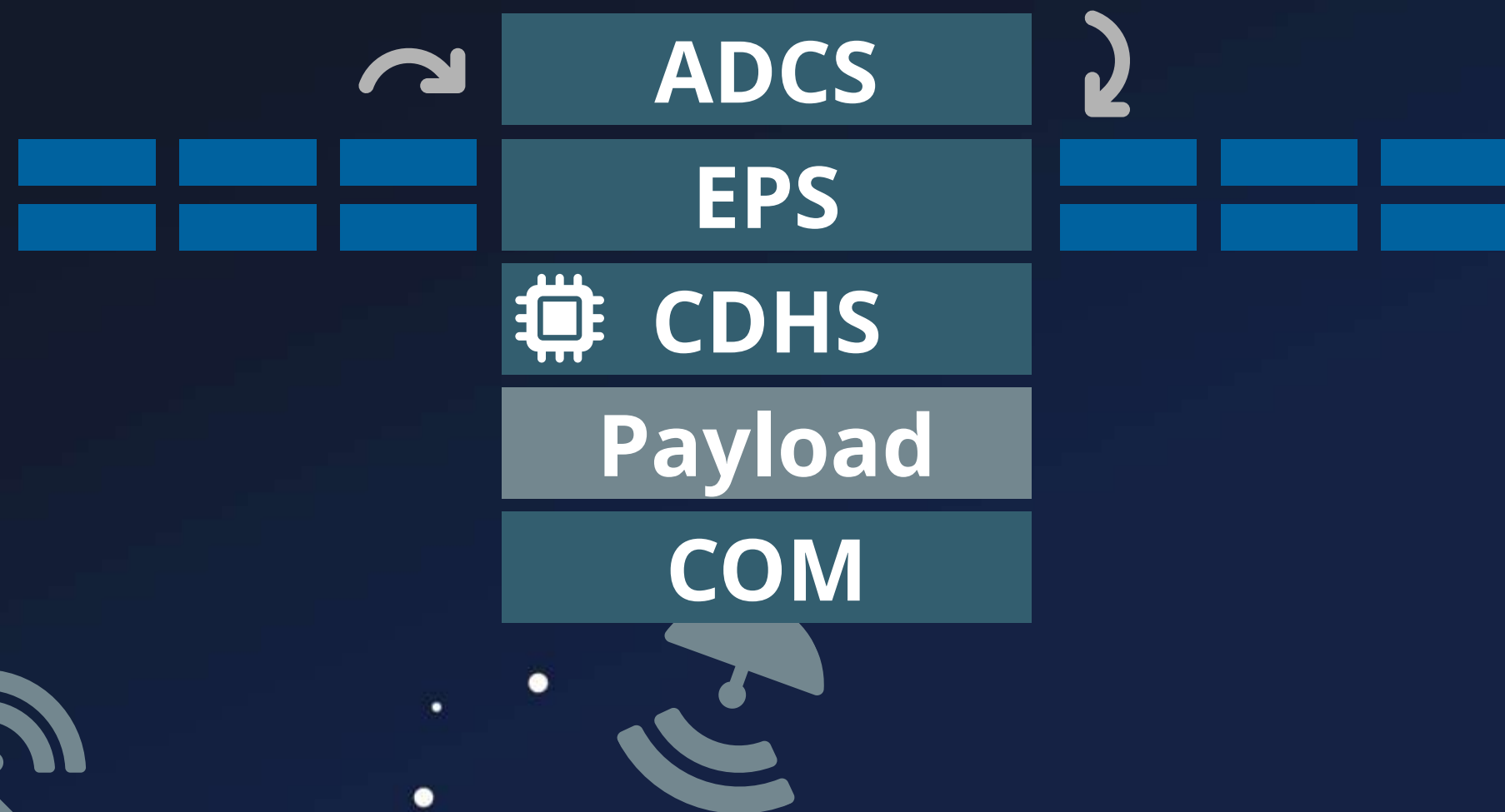
Components



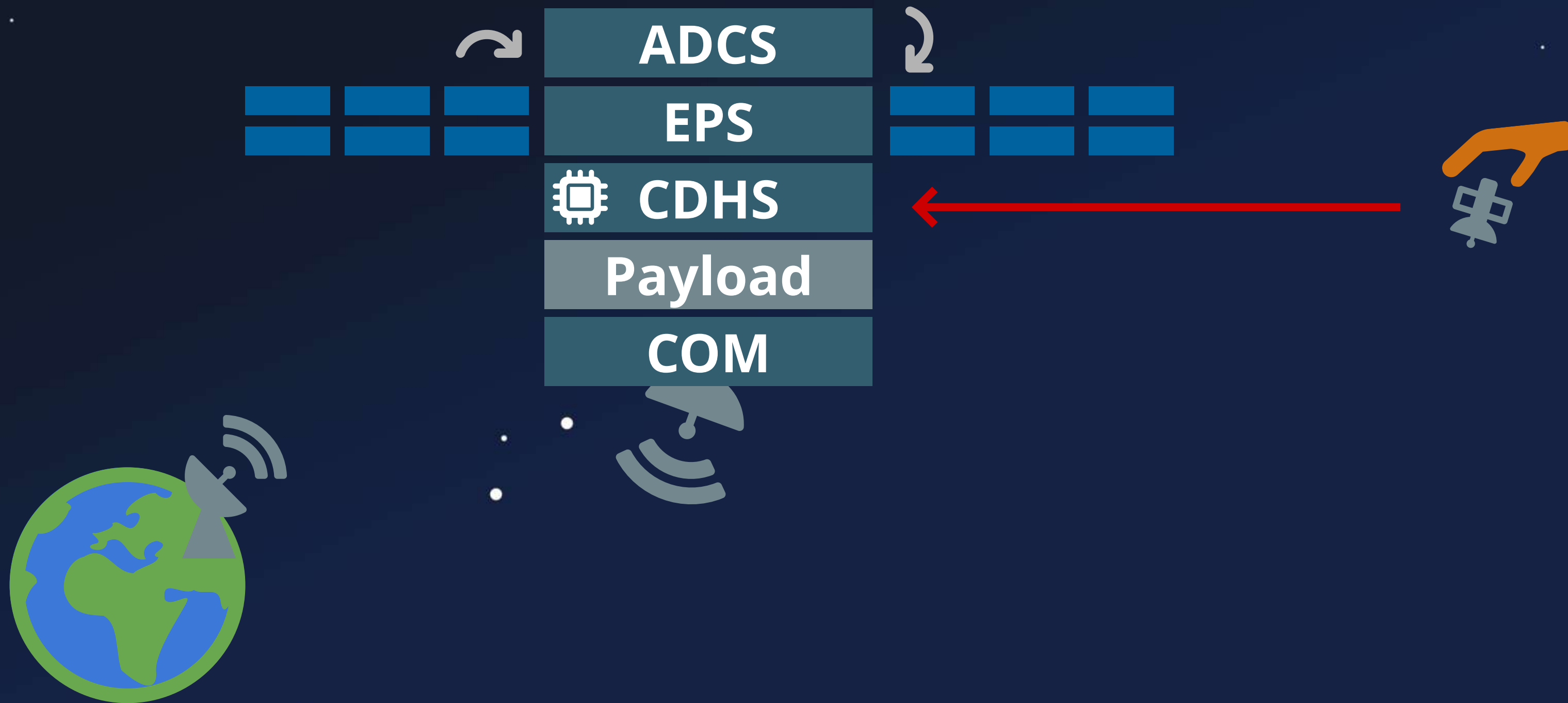
Components



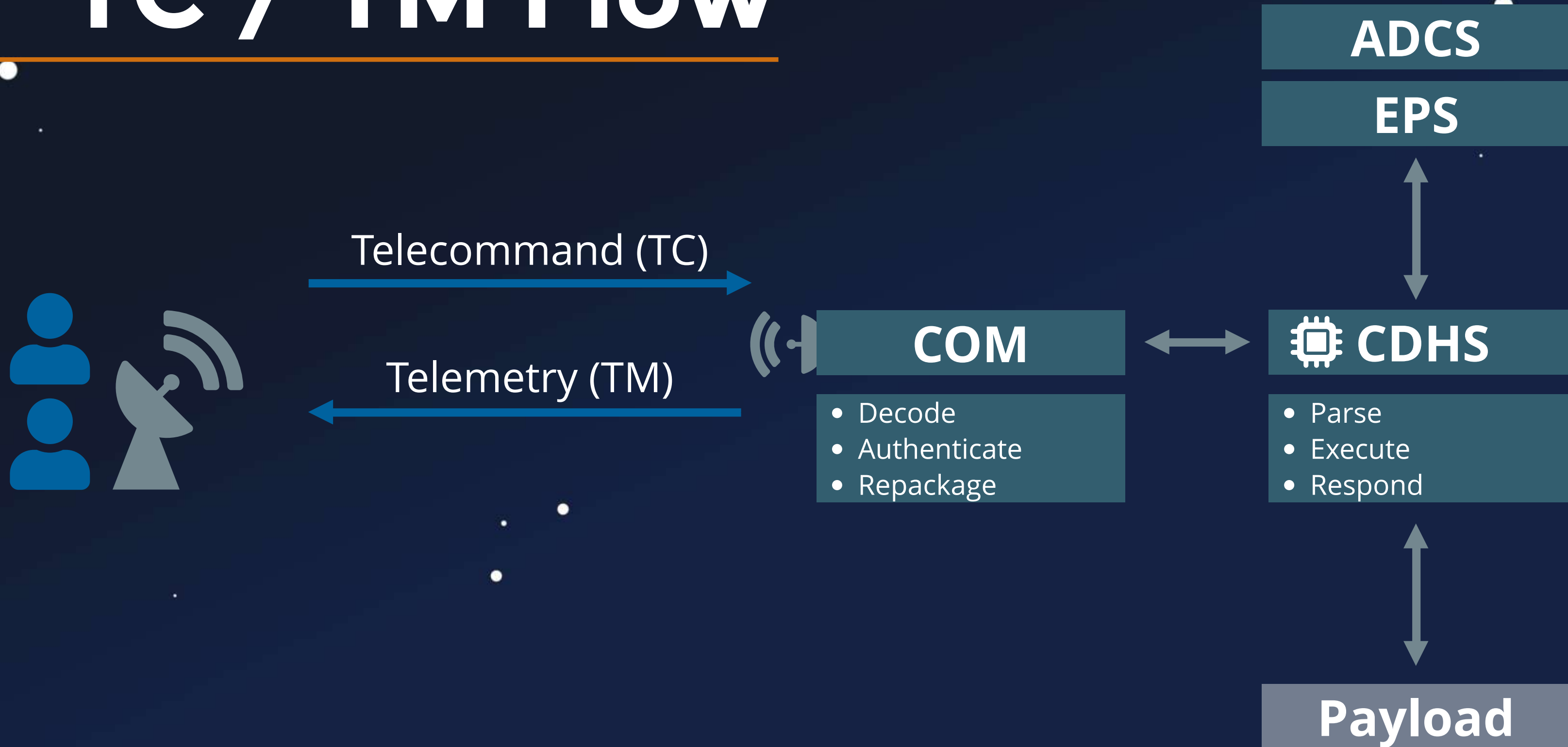
Components



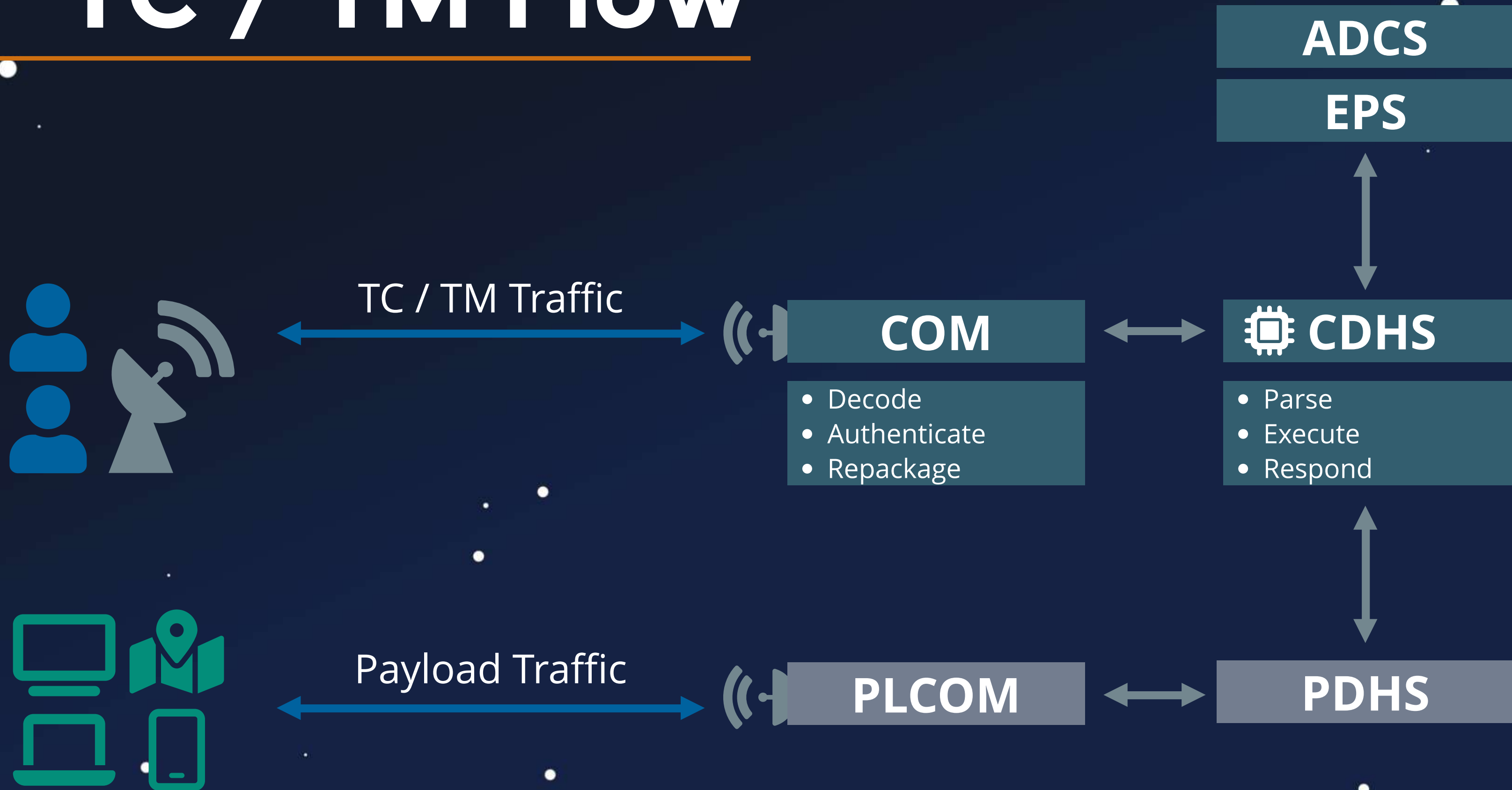
Components



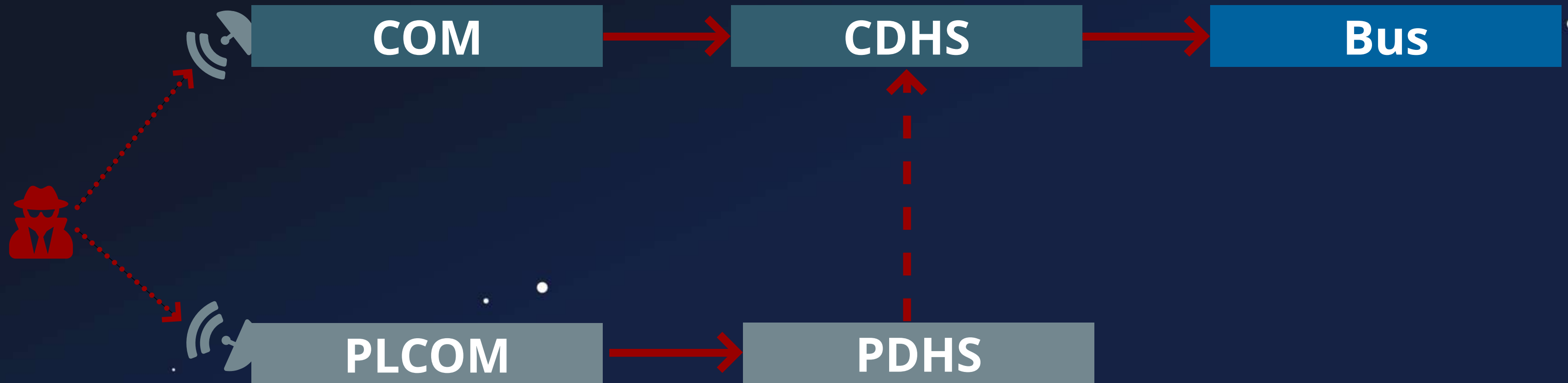
TC / TM Flow



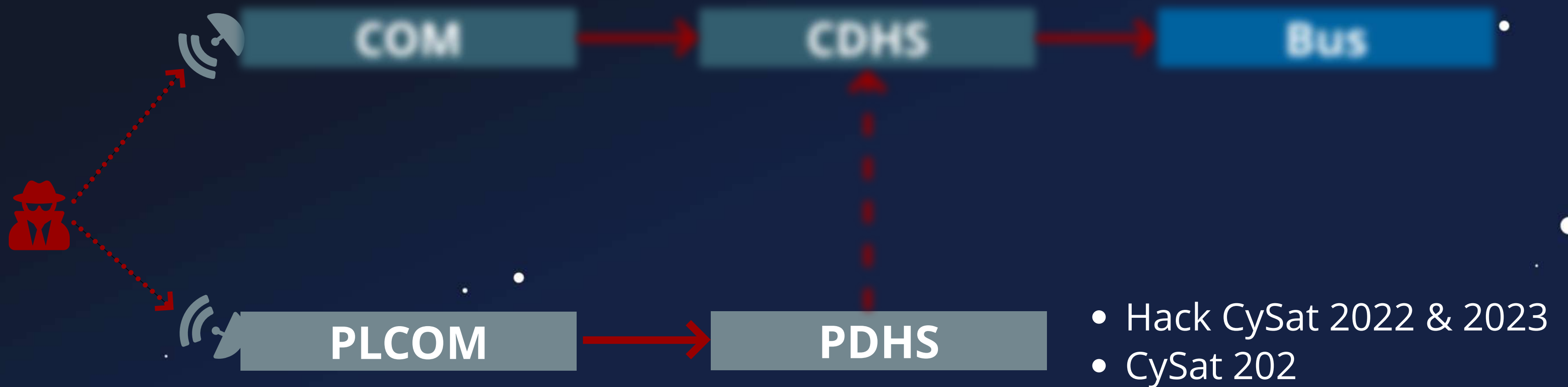
TC / TM Flow



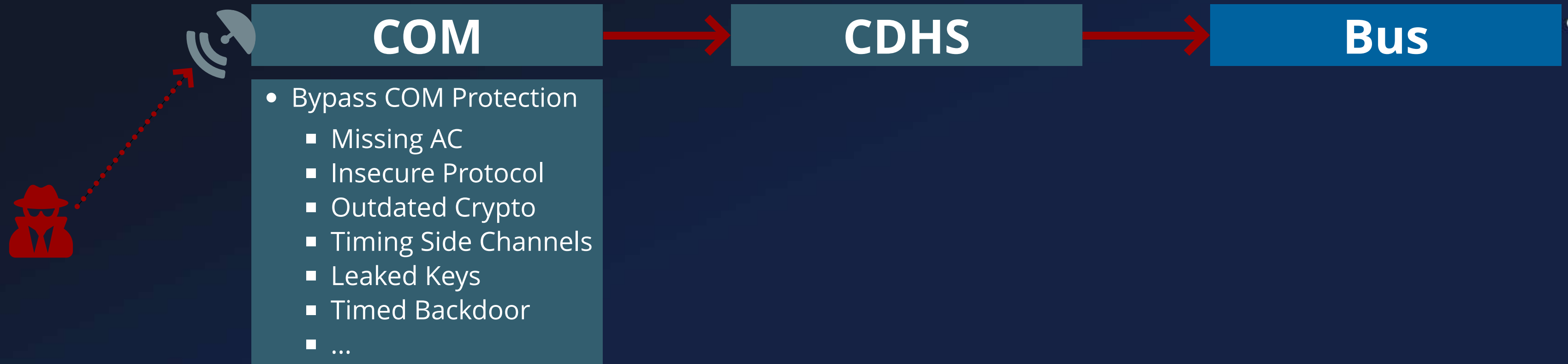
Attack Path



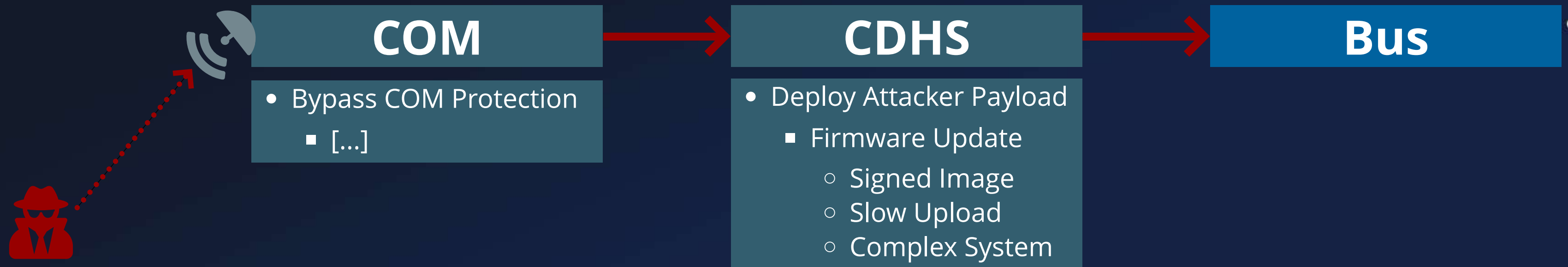
Attack Path



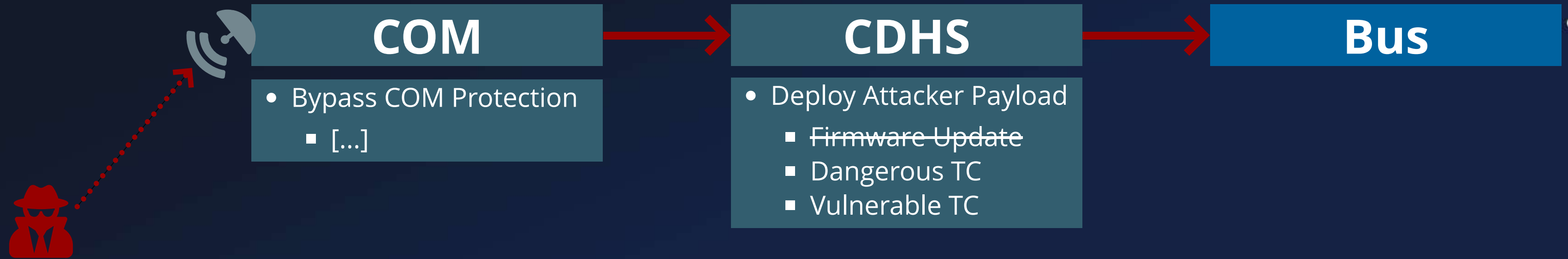
Attack Path



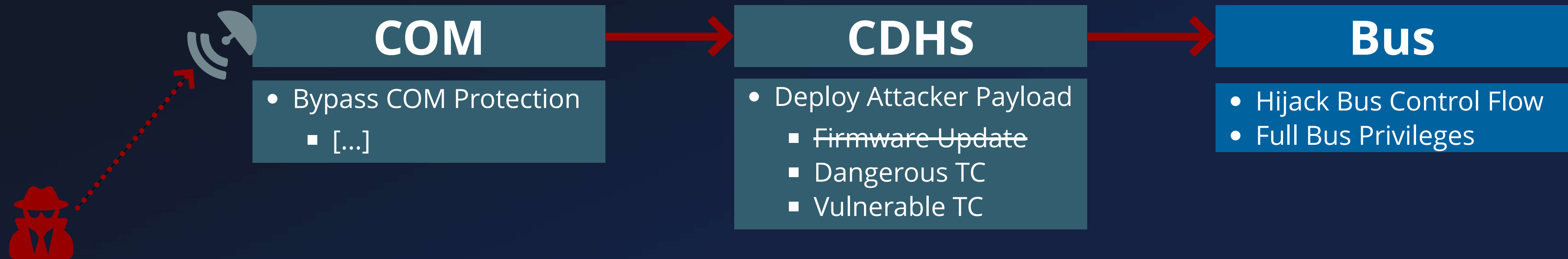
Attack Path



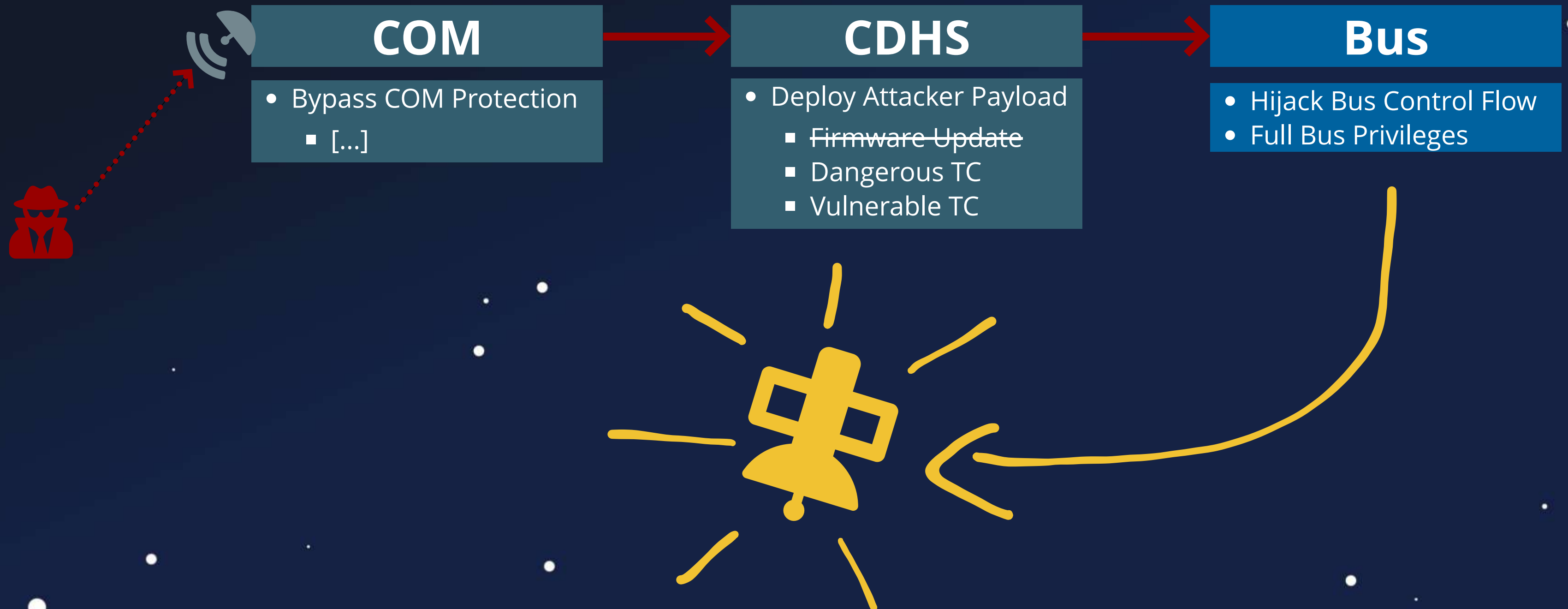
Attack Path



Attack Path



Attack Path

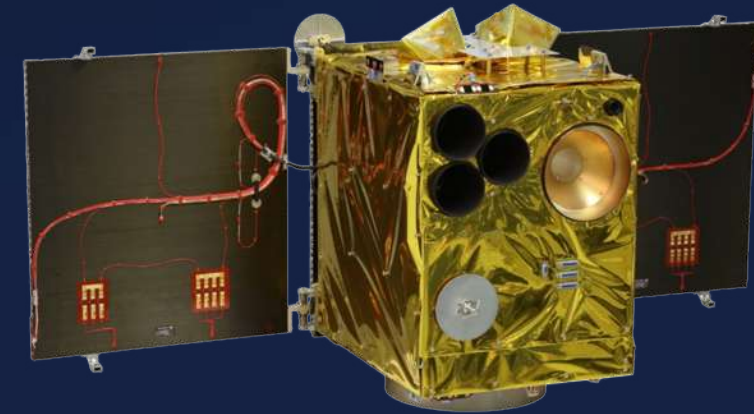
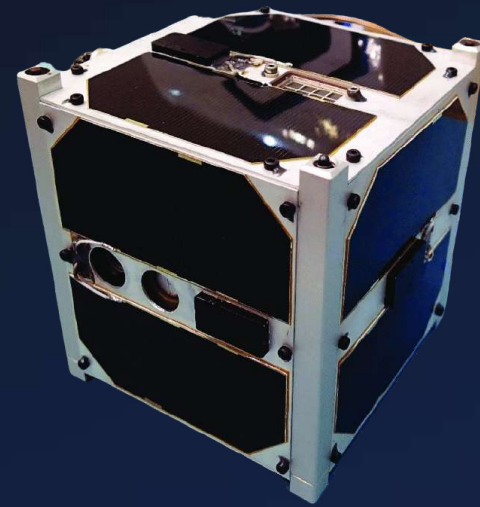


Objectives



- ① Bypass COM Protection
- ② Dangerous / Vulnerable TC
- ③ Hijack Bus Control Flow
- ④ Full Bus Privileges

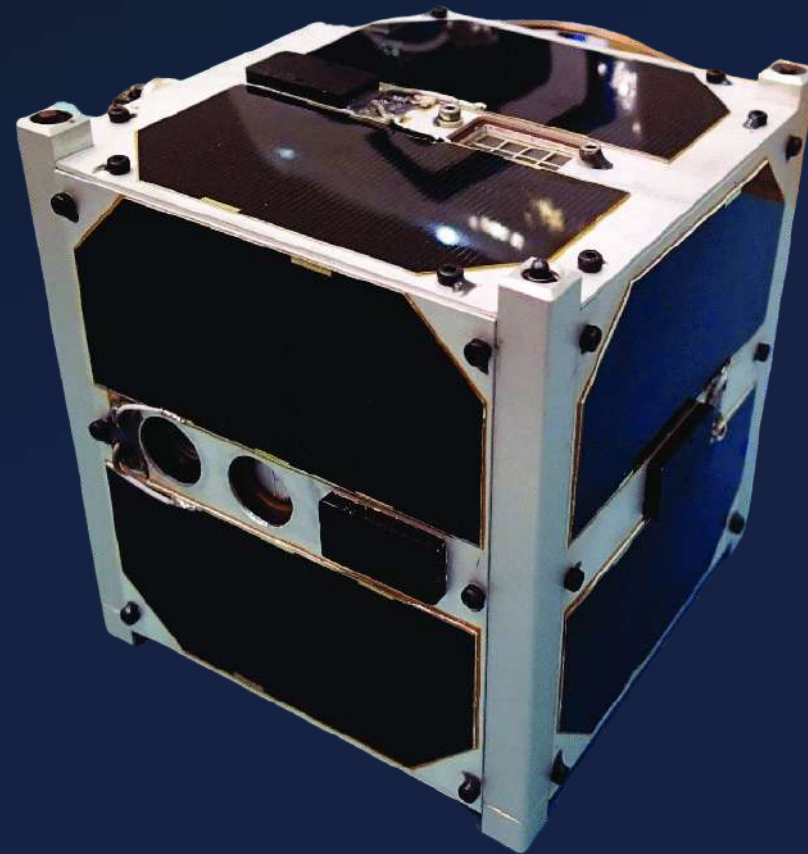
Satellite Case Studies



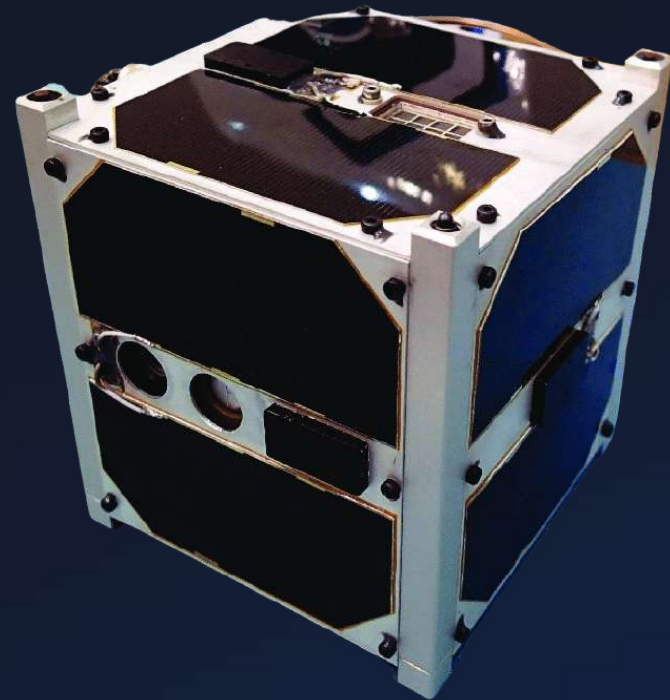
Approach



ESTCube-1

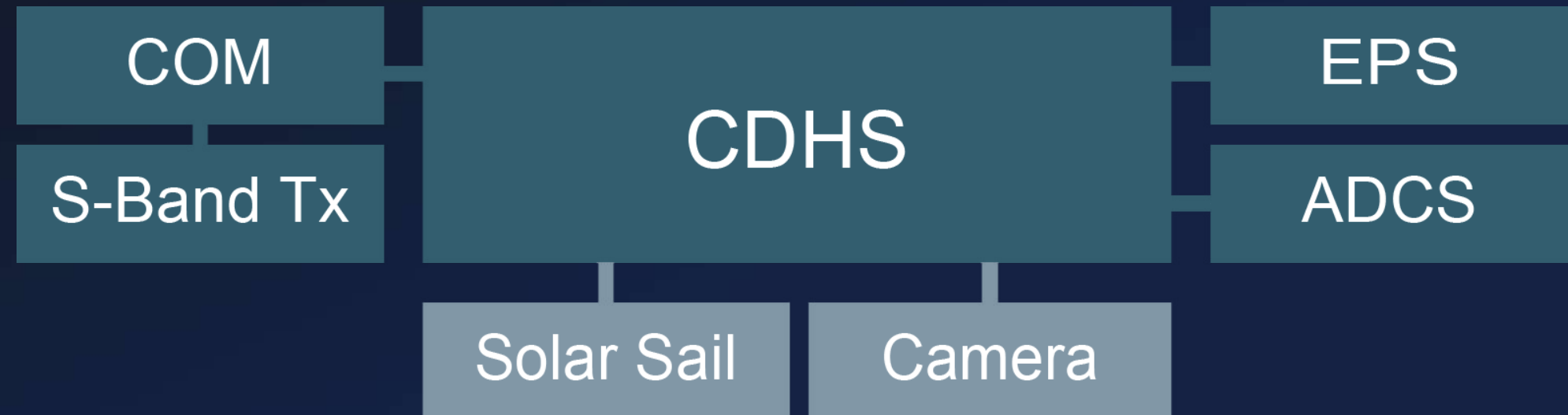


ESTCube-1



ESTCube-1 ^[1]

Developed by
University of Tartu



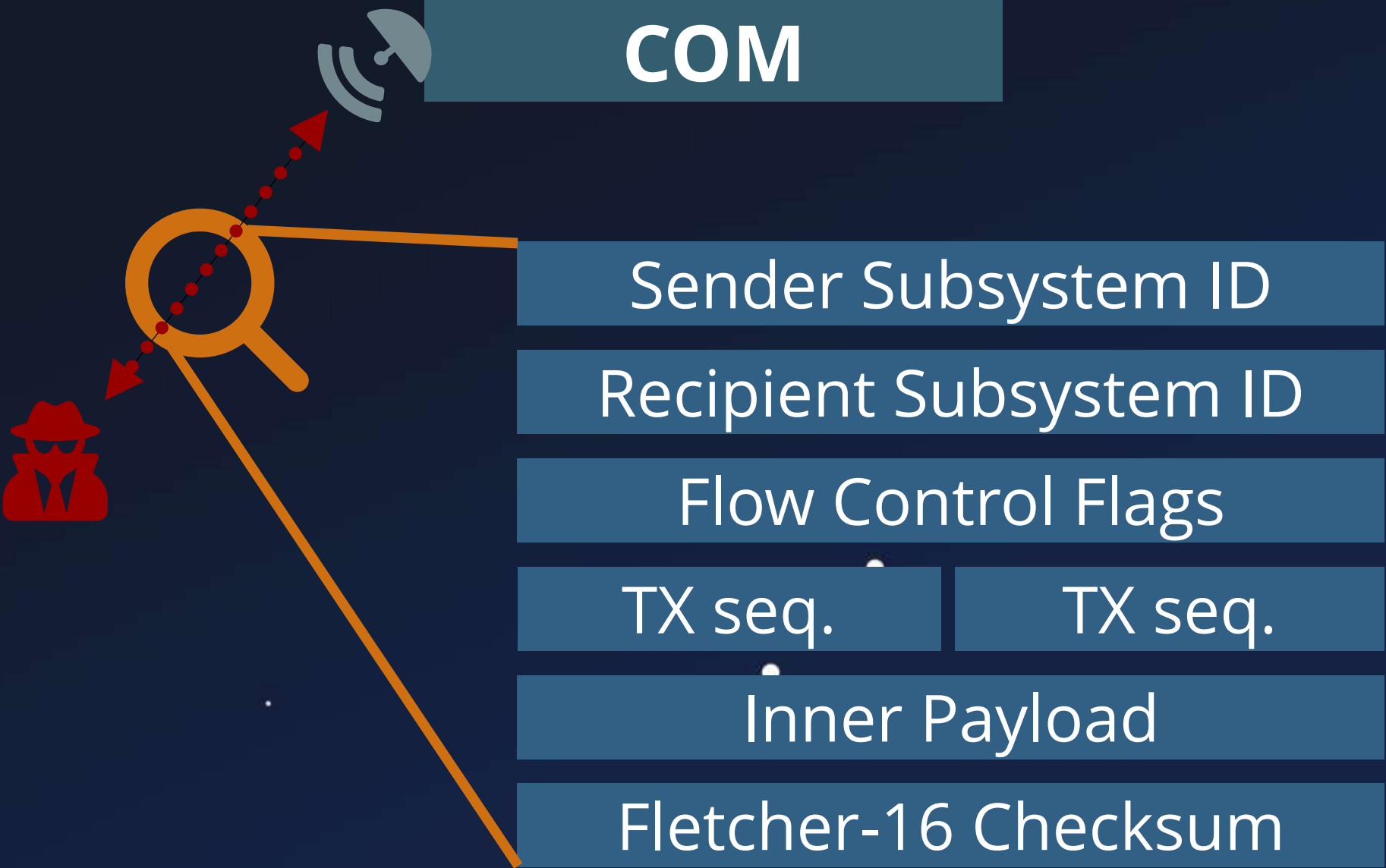
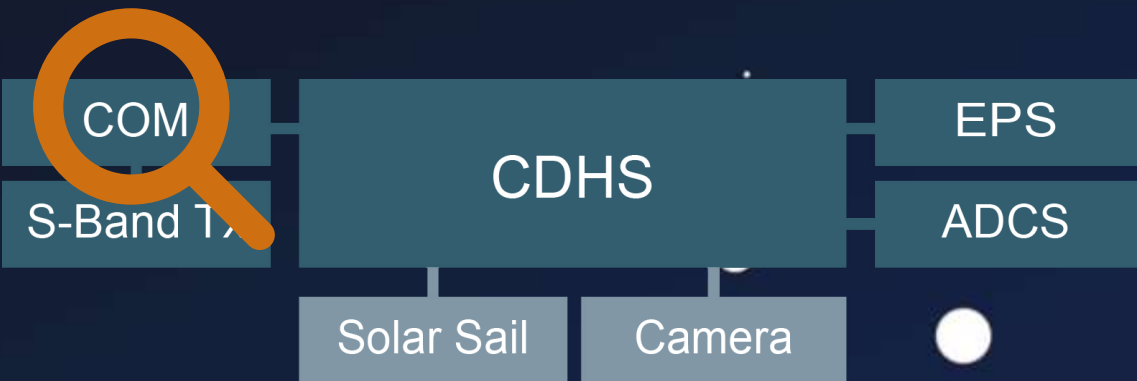
E-Sail (Electric Solar Wind Sail) Propulsion

Peripherals

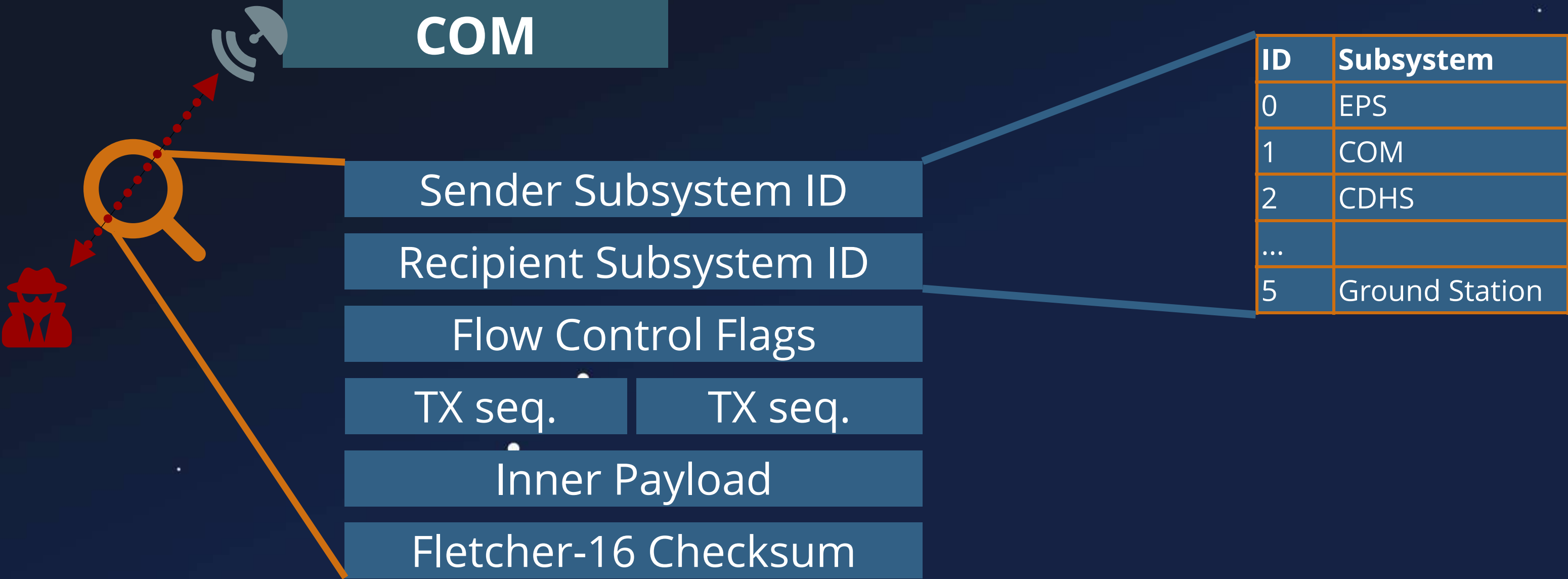
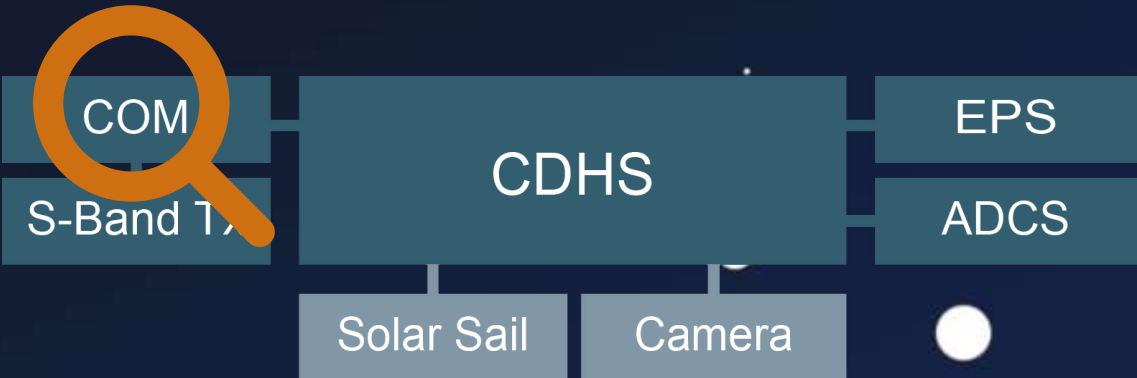
ARM STM32

Payload Platform

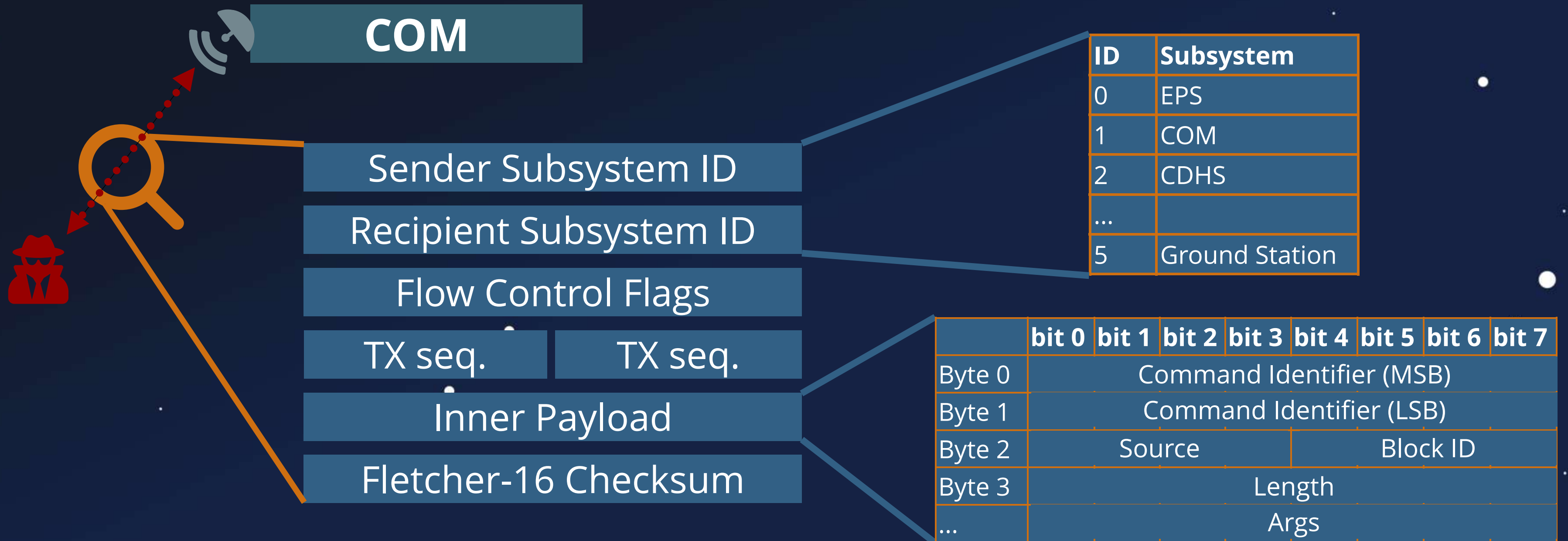
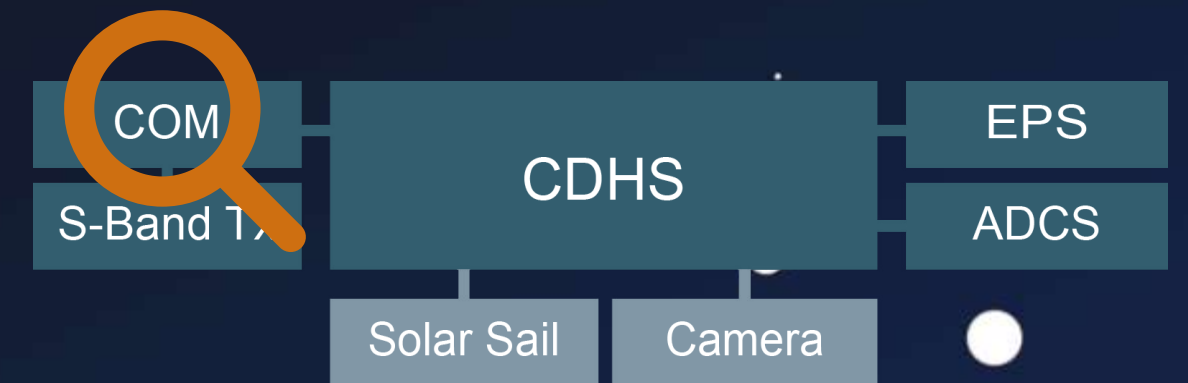
Custom Protocol



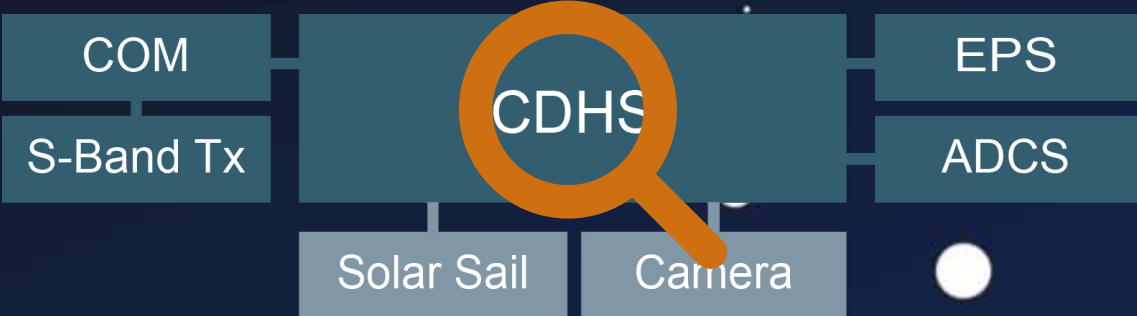
Custom Protocol



Custom Protocol



Security Analysis



	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
Byte 0	Command Identifier (MSB)							
Byte 1	Command Identifier (LSB)							
Byte 2	Source				Block ID			
Byte 3	Length							
...	Args							

```
1 int sch_handle_command(scheduler_packed_cmd_t *pCmd) {
2     // ! simplified !
3     sch_unpack_command(&g_command, pCmd);
4     // ...
5     handler_func = &handler_table[g_command.handler_func_index] ;
6     // ...
7     retval = (*handler_func) (&g_command) ;
8 }
```

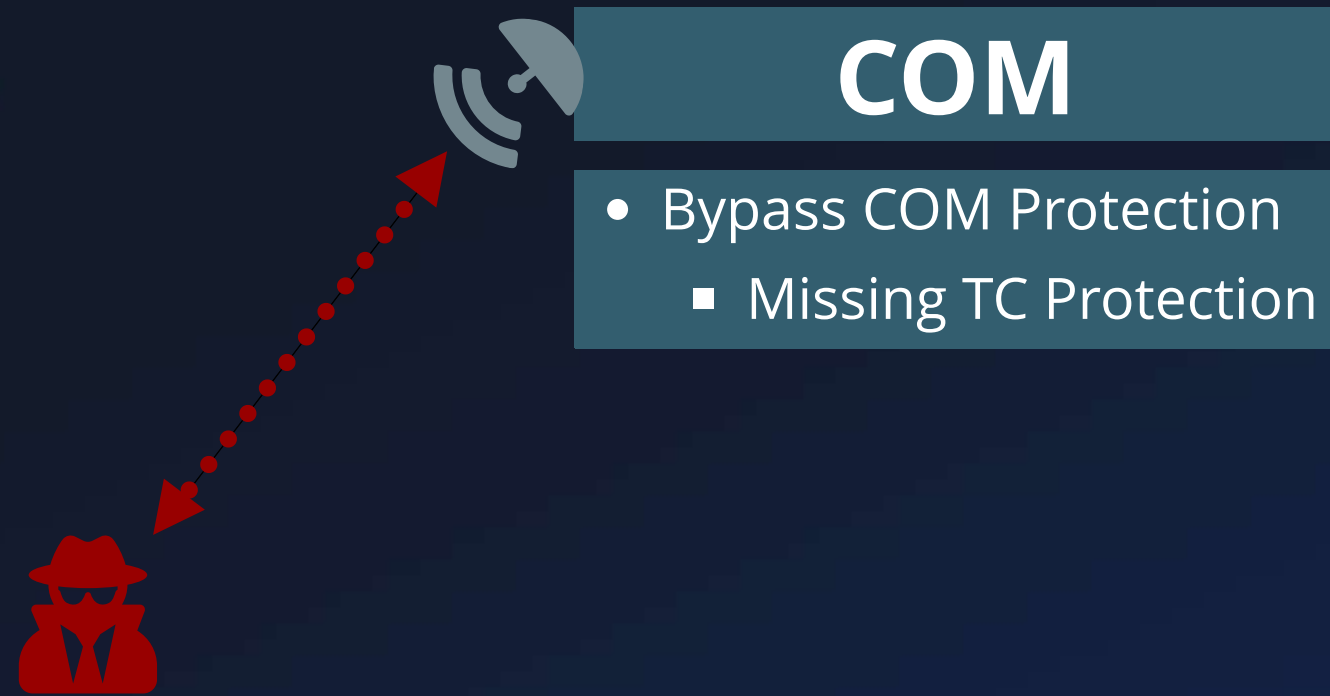
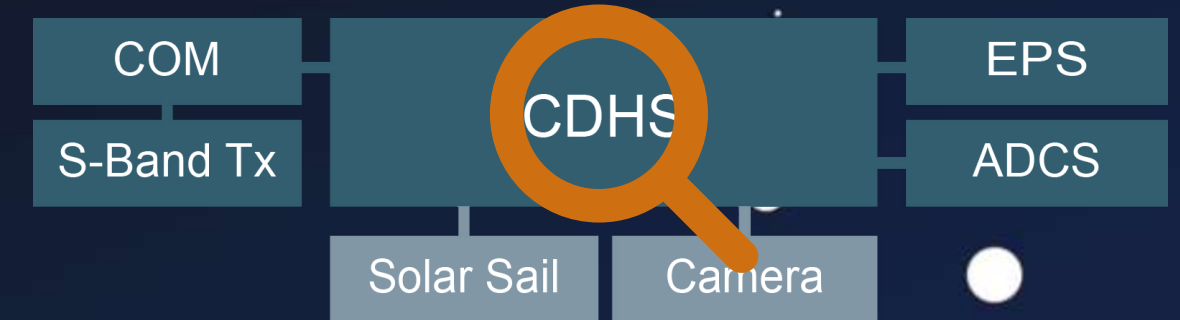
Security Analysis



	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
Byte 0	Command Identifier (MSB)							
Byte 1	Command Identifier (LSB)							
Byte 2	Source				Block ID			
Byte 3	Length							
...	Args							

```
1 int sch_handle_command(scheduler_packed_cmd_t *pCmd) {
2     // ! simplified !
3     sch_unpack_command(&g_command, pCmd);
4     // ...
5     handler_func = &handler_table[g_command.handler_func_index] ;
6     // ...
7     retval = (*handler_func) (&g_command) ;
8 }
```

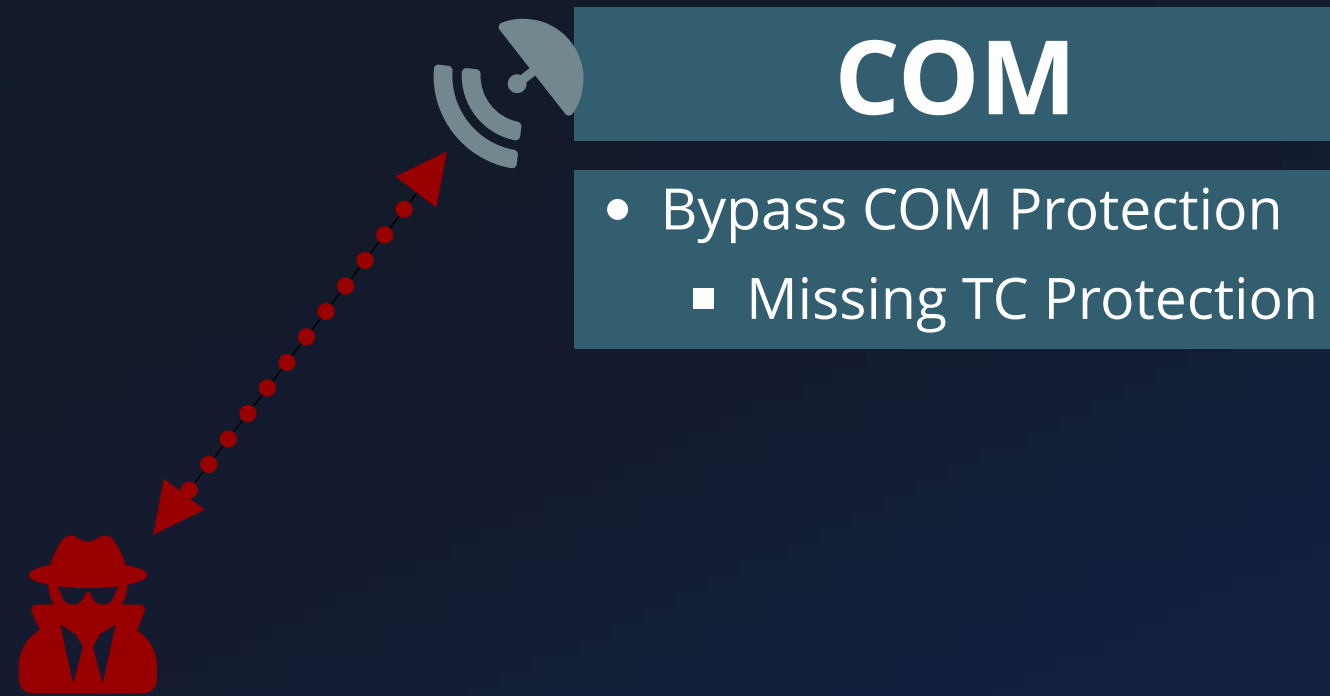
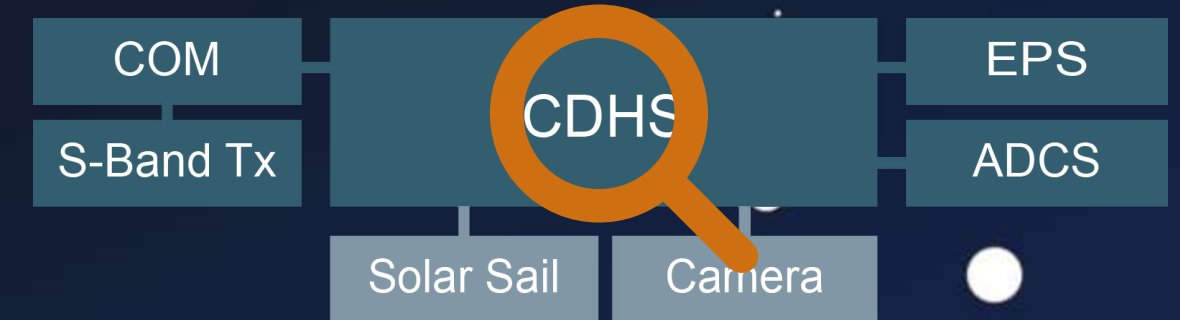
Security Analysis



CDHS

```
1  int sch_handler_set_raw_memory(scheduler_cmd_t* pCmd) {
2      raw_mem_access_cmd_t* pAddr = pCmd->pCmdArgs;
3      char* pWriteData;
4
5      if (pAddr) {
6          if (g_sch_exec_mode != 1 ) {
7              /* exception and return */
8          }
9          char* pWriteData = &pAddr->start_of_data_buf;
10         if (pAddr->filesystem_target) {
11             // [...]
12         } else {
13             memcpy(pAddr->targetAddr,
14                   &pAddr->start_of_data_buf,
15                   pAddr->writeLength);
16         }
17     }
18     // ...
19 }
```

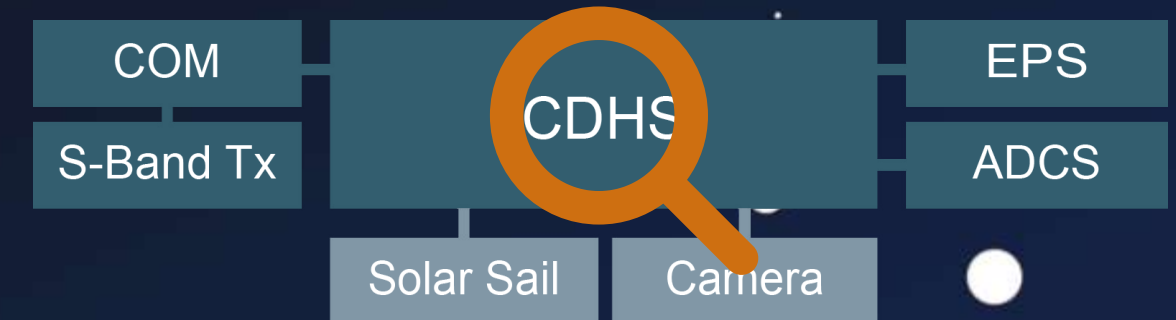
Security Analysis



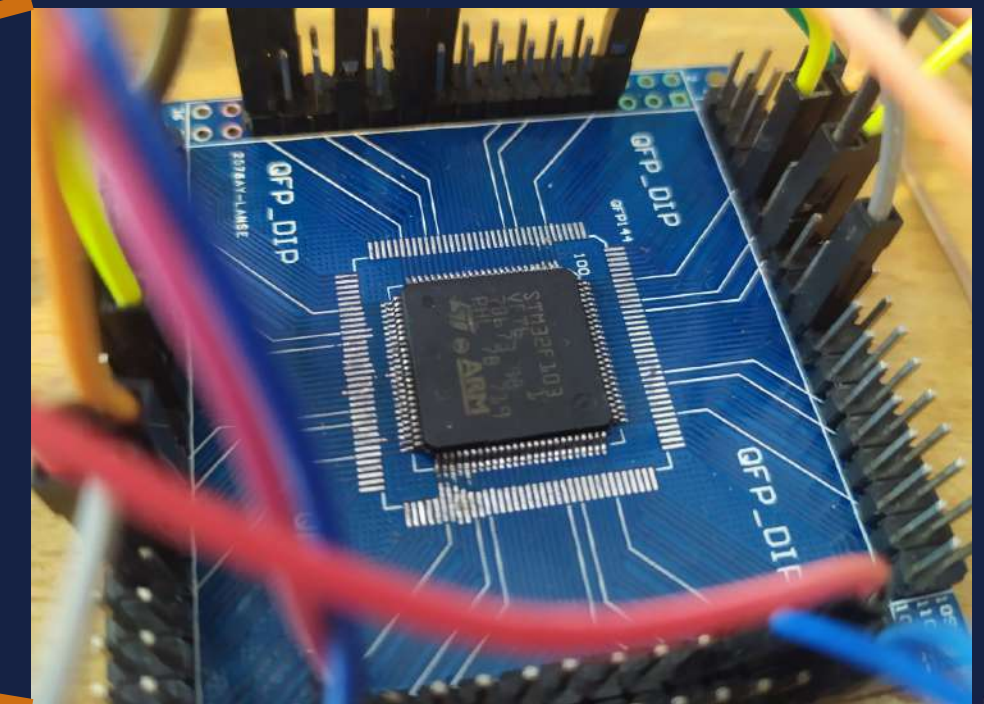
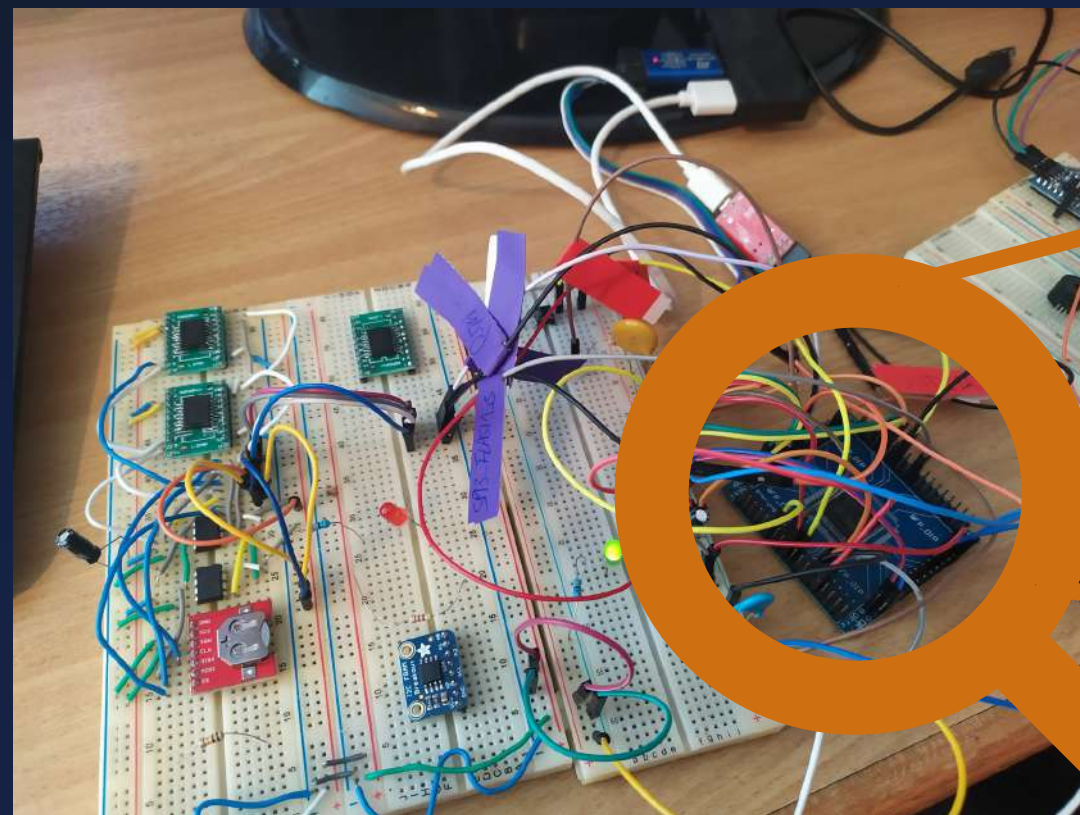
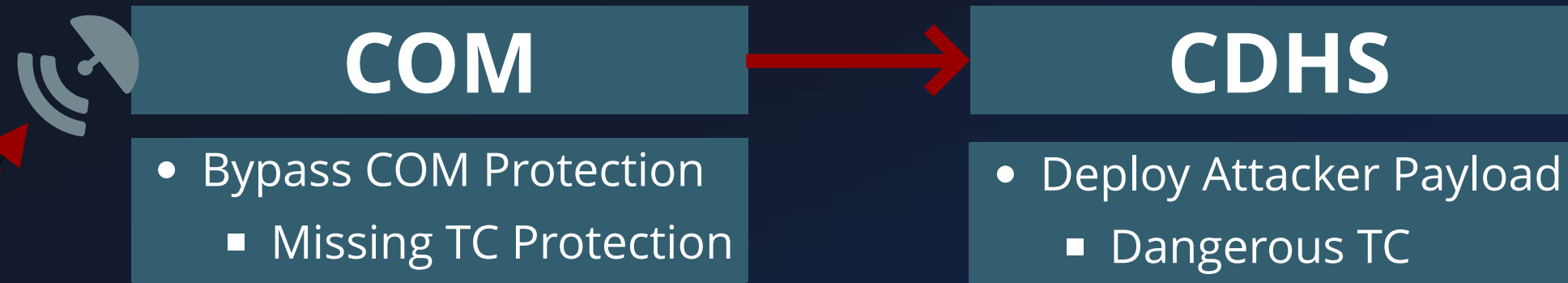
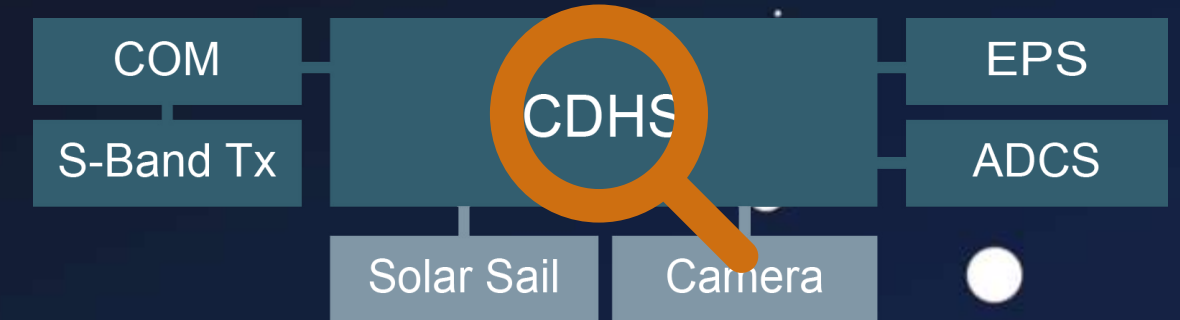
CDHS

```
1 int sch_handler_set_raw_memory(scheduler_cmd_t* pCmd) {
2     raw_mem_access_cmd_t* pAddr = pCmd->pCmdArgs;
3     char* pWriteData;
4
5     if (pAddr) {
6         if (g_sch_exec_mode != 1 ) {
7             /* exception and return */
8         }
9         char* pWriteData = &pAddr->start_of_data_buf;
10        if (pAddr->filesystem_target) {
11            // [...]
12        } else {
13            memcpy(pAddr->targetAddr,
14                  &pAddr->start_of_data_buf,
15                  pAddr->writeLength);
16        }
17    }
18    // ...
19 }
```

Real-World Test



Real-World Test



OPS-Sat

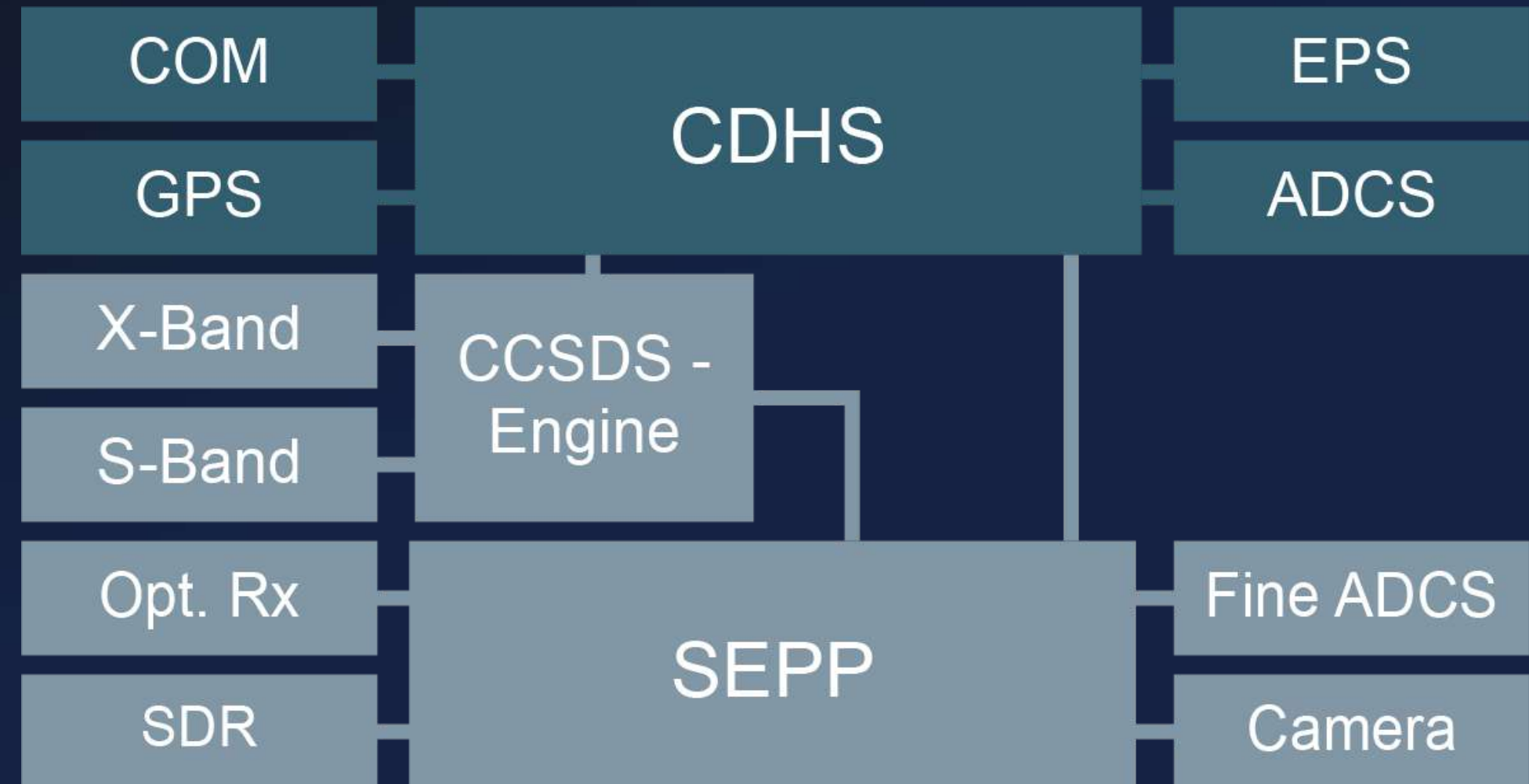


System Chart



Experimenter

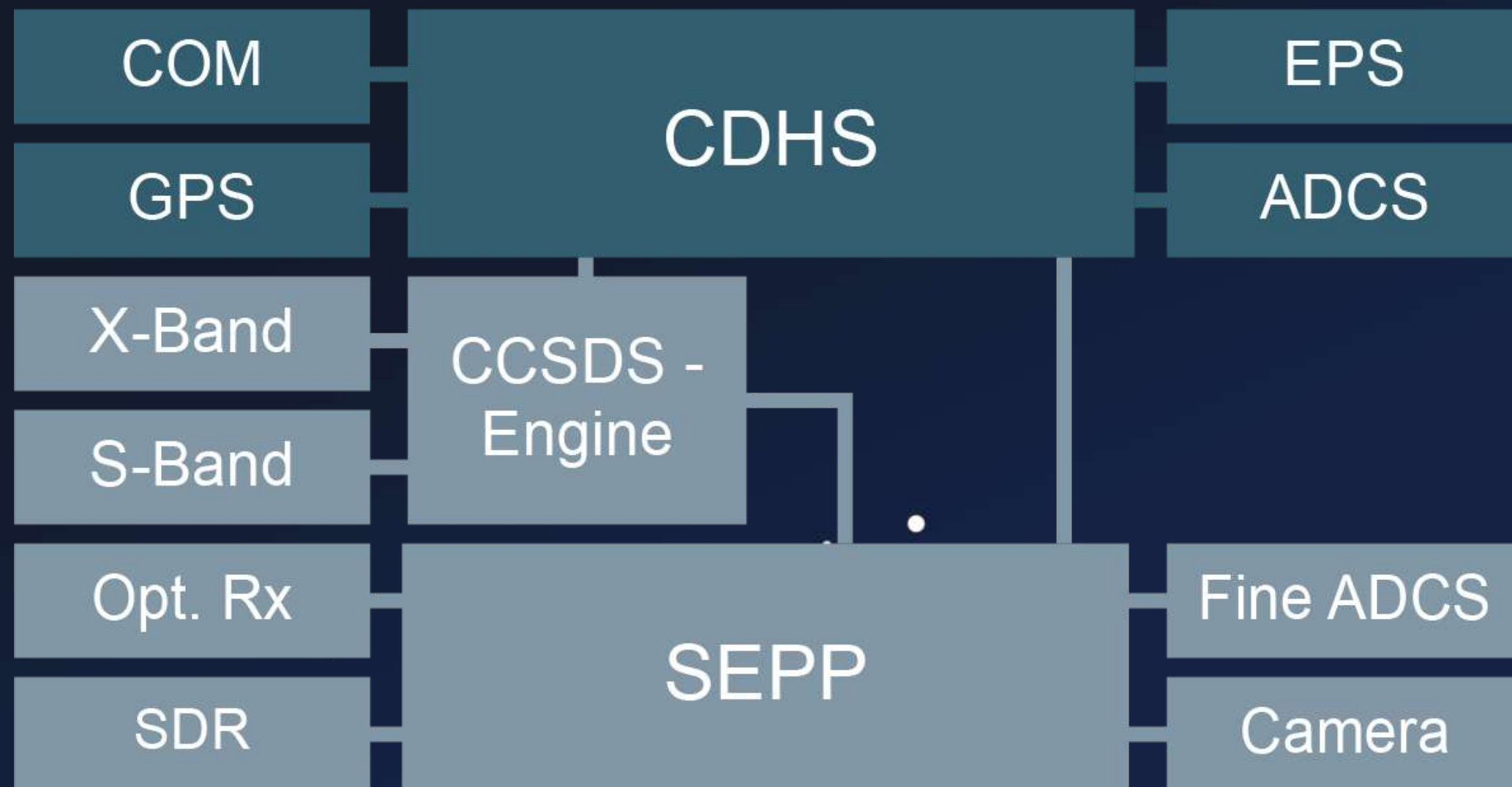
Operated by ESA
Open for Research



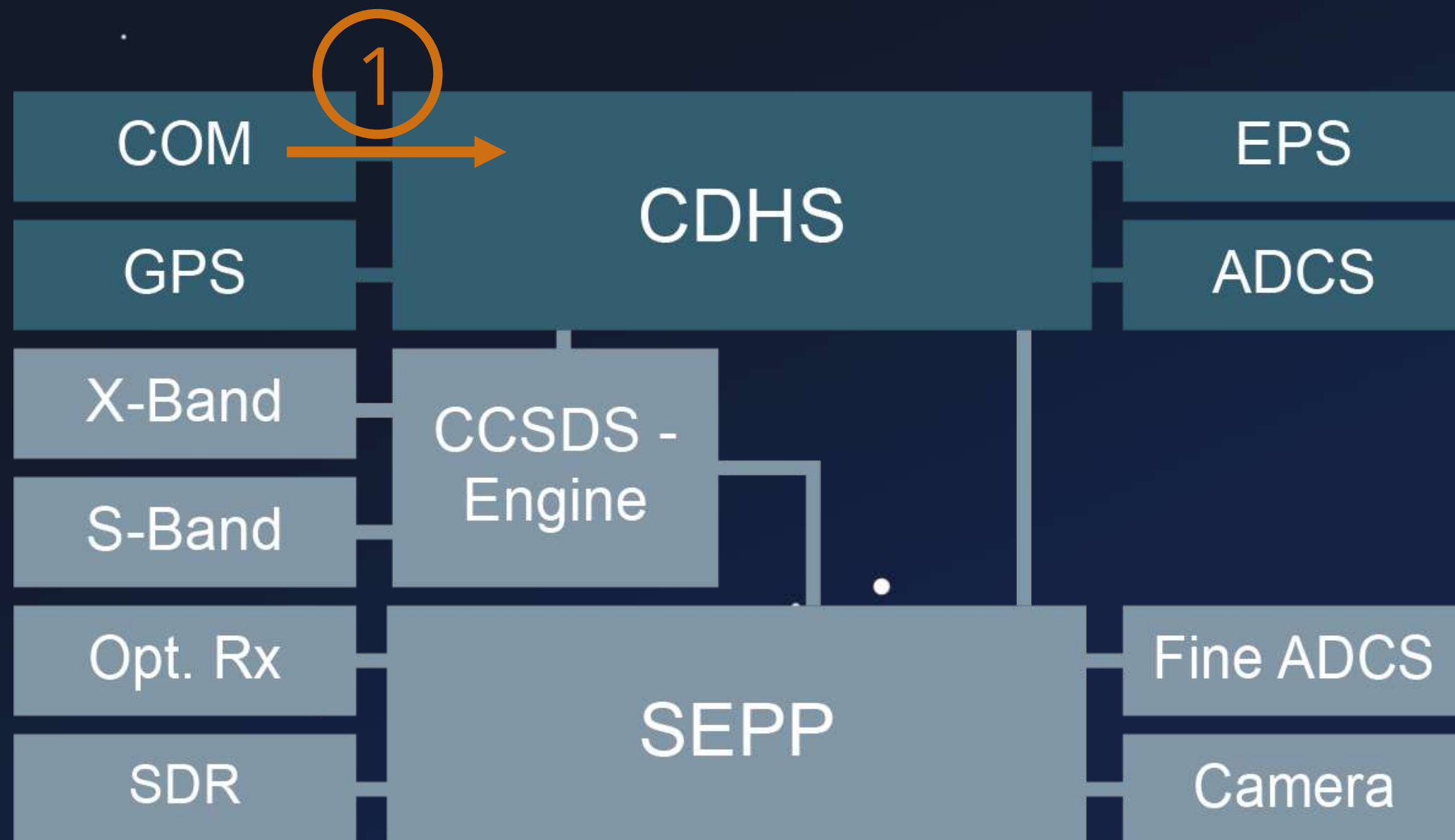
S-/X-Band, SDR, Optical Rx., Camera, ...

Peripherals

System Chart

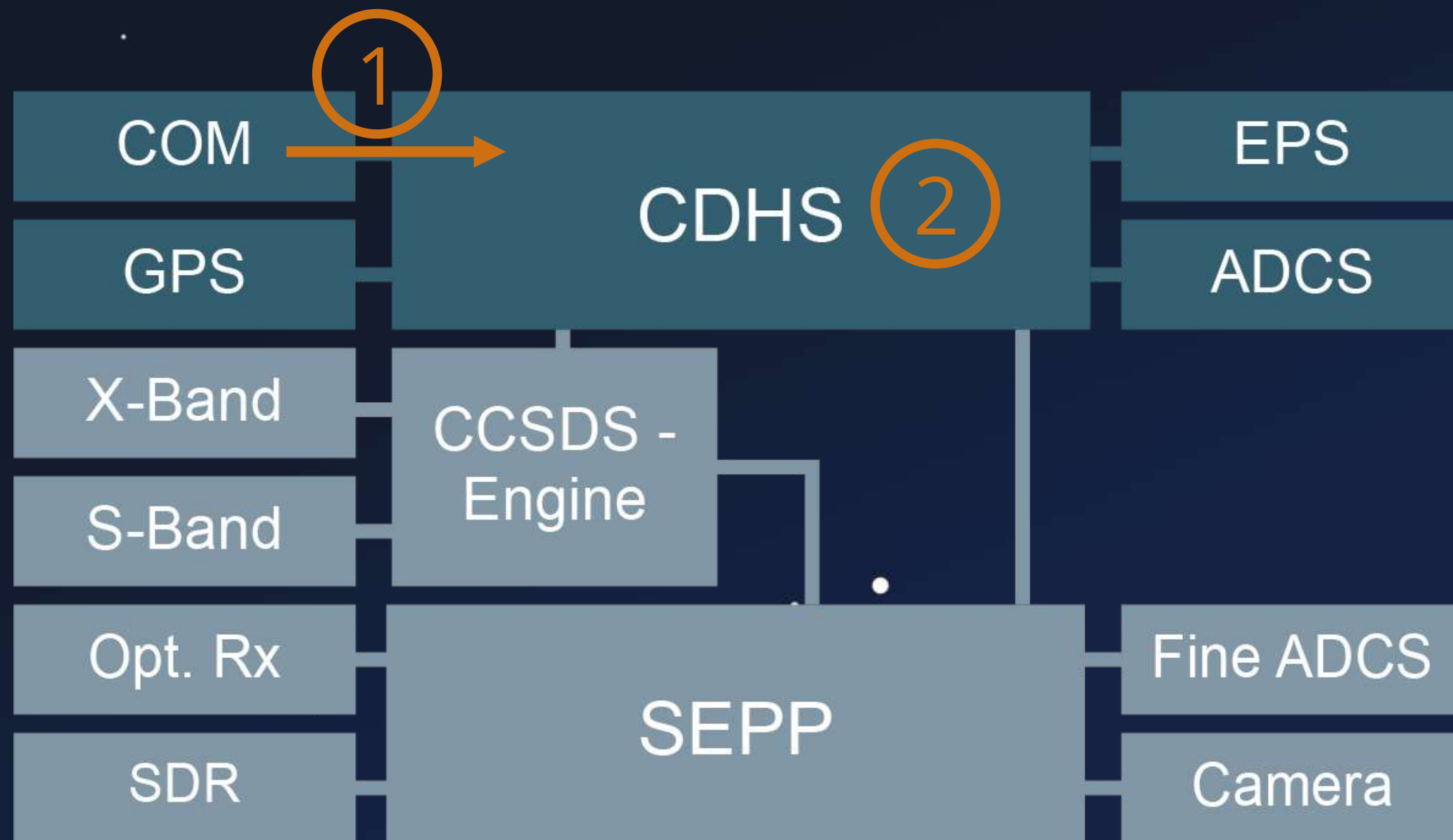


System Chart



1 Cubesat Space Protocol (CSP)

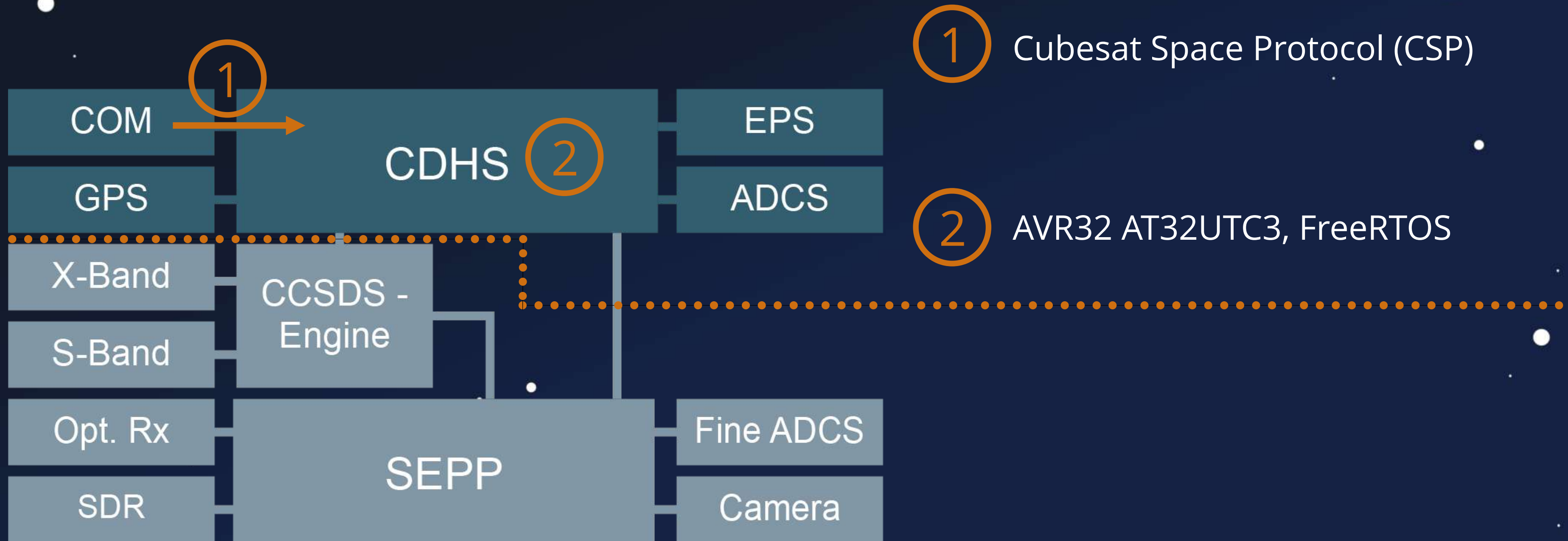
System Chart



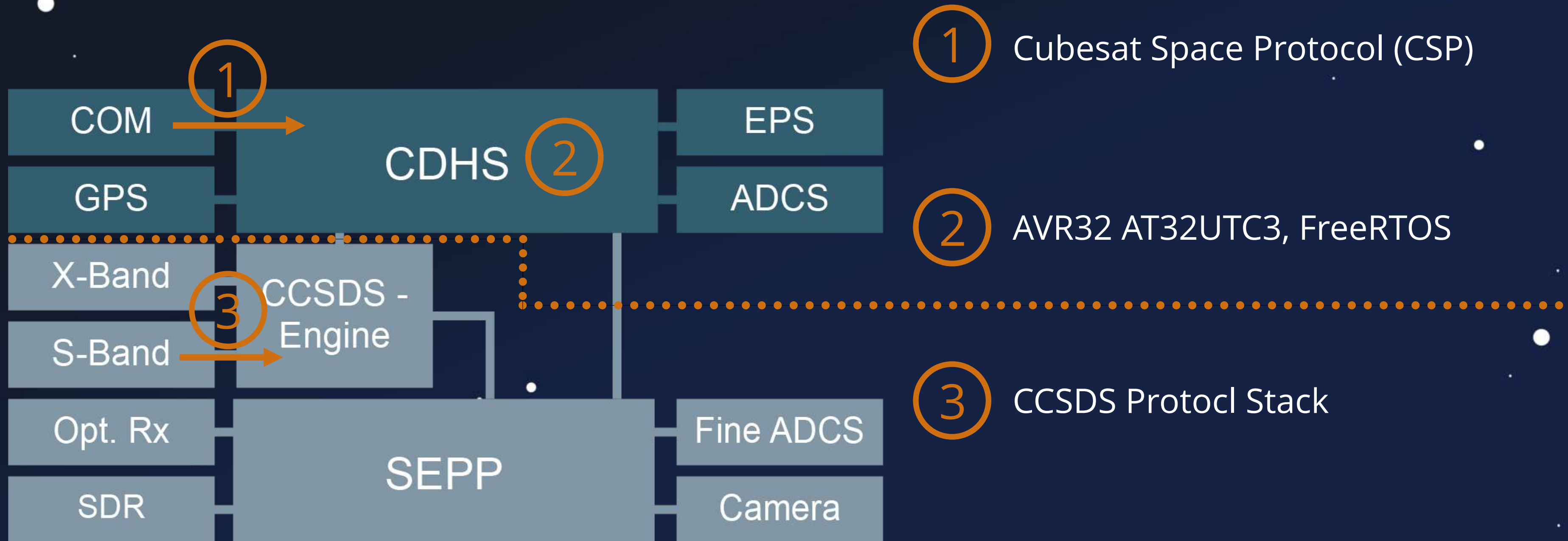
① Cubesat Space Protocol (CSP)

② AVR32 AT32UTC3, FreeRTOS

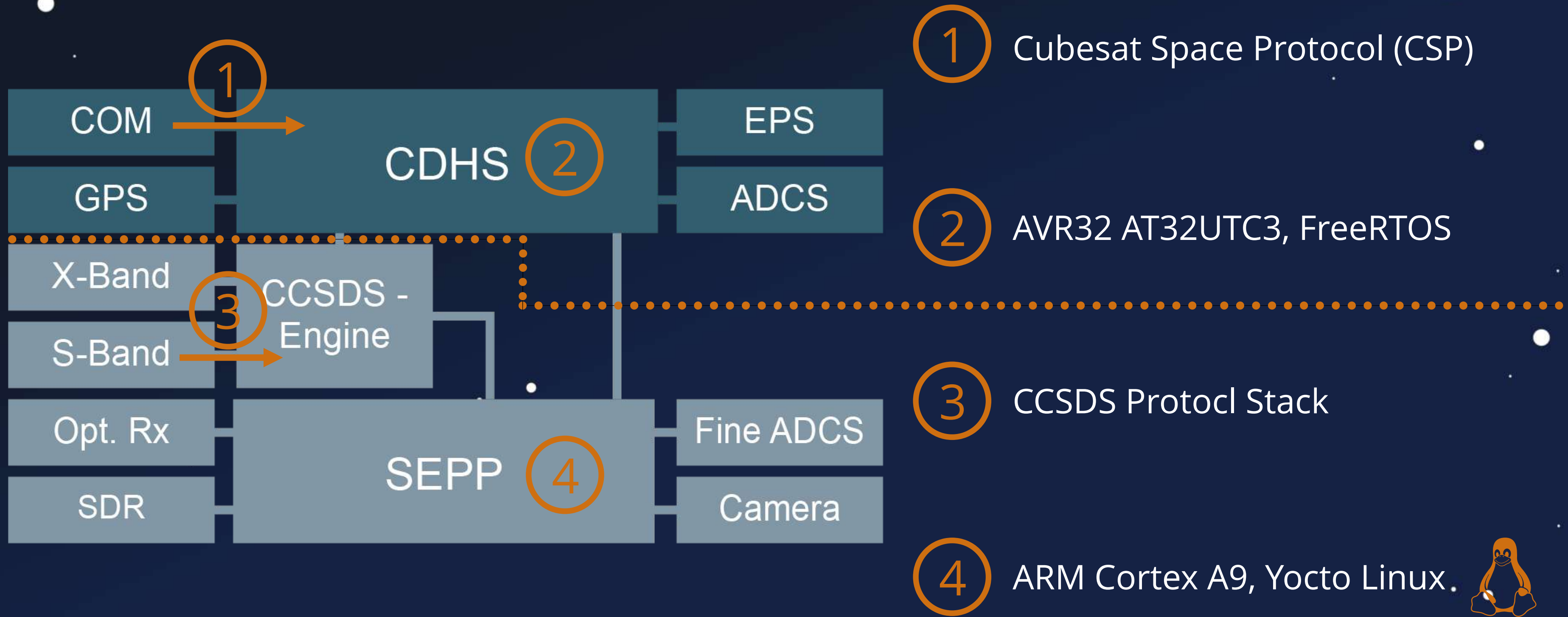
System Chart



System Chart



System Chart

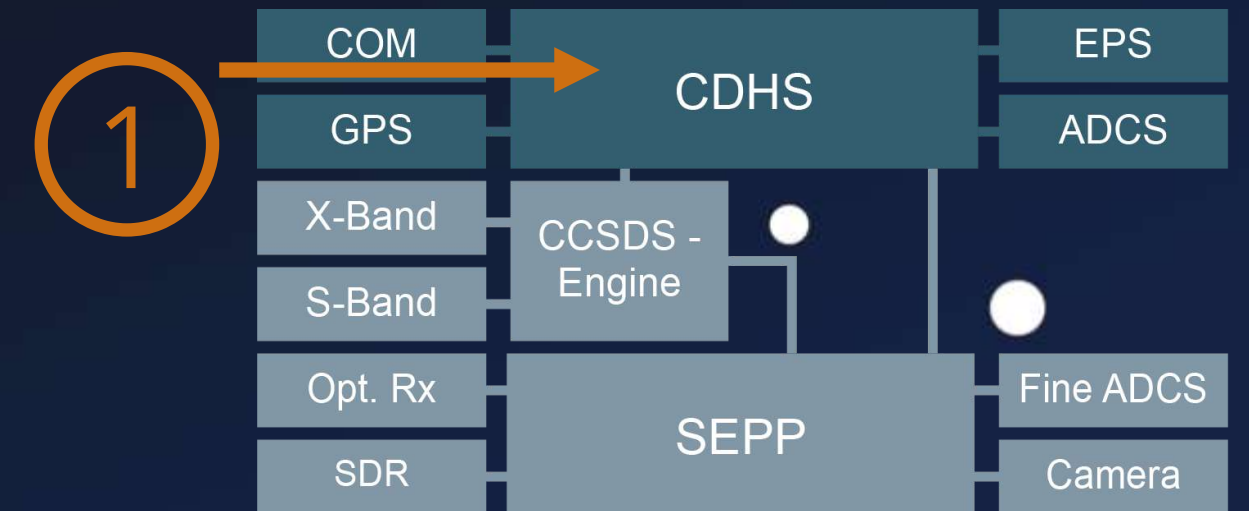


UHF-Stack

Cubesat Space Protocol (CSP) v1



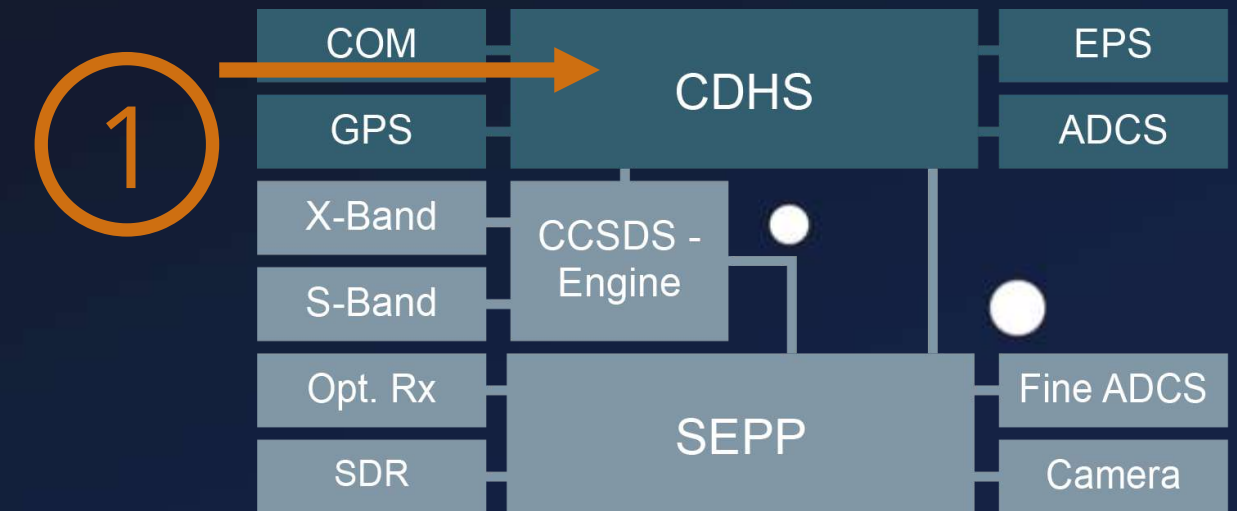
TCP/IP Oriented Design



CSP Header 1.x																																
Bit offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Priority		Source				Destination				Destination Port				Source Port				Reserved				H M A C	X T E A	R D P	C R C						
32	Data (0 – 65,535 bytes)																															

Source: https://en.wikipedia.org/wiki/Cubesat_Space_Protocol

UHF-Stack

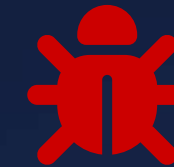


Cubesat Space Protocol (CSP) v1



Security Features

- HMAC-SHA1 Authentication
- XTEA Encryption Support

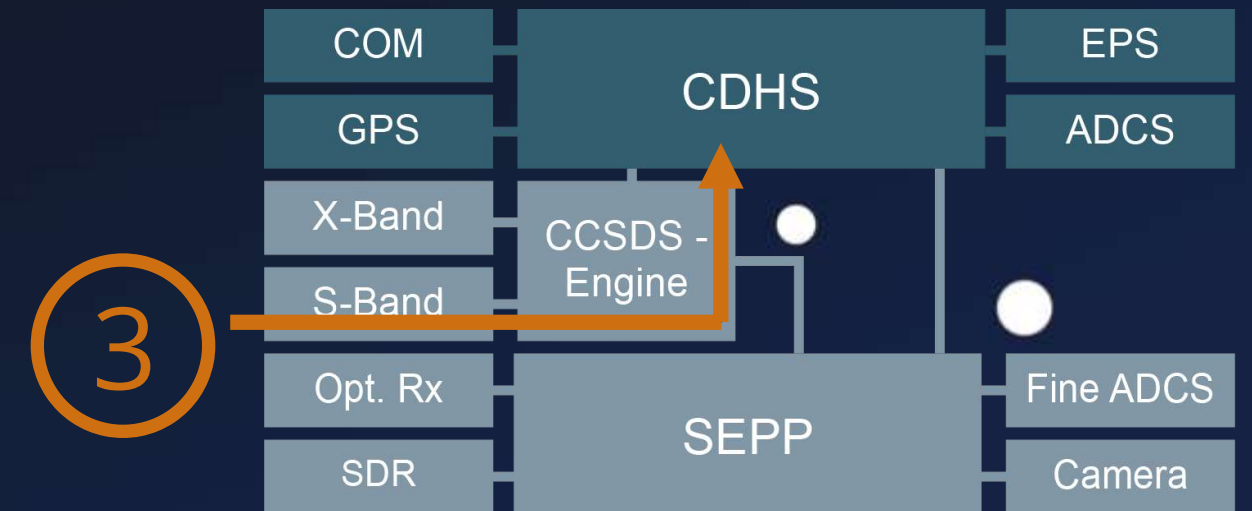


Security Issues

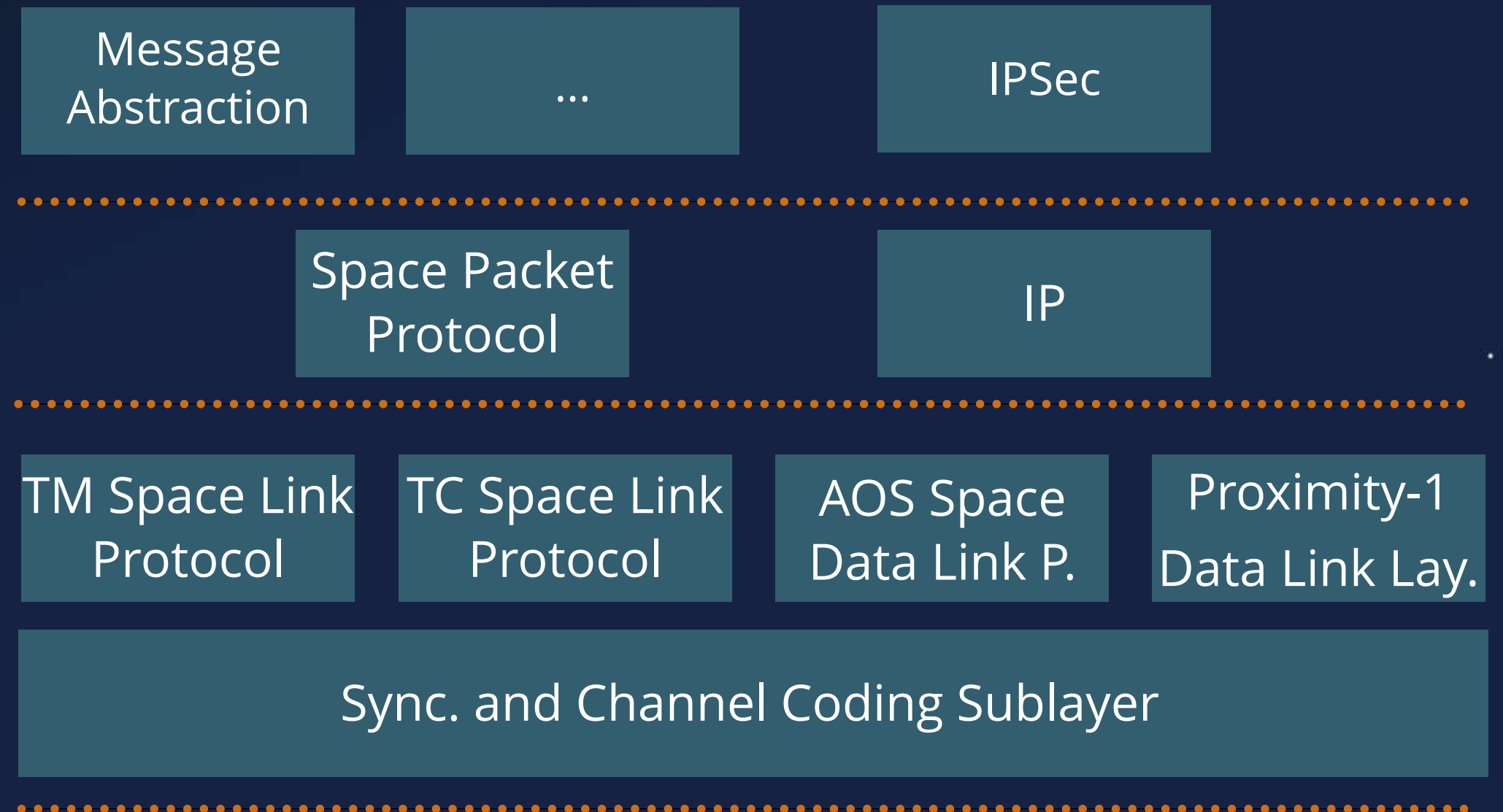
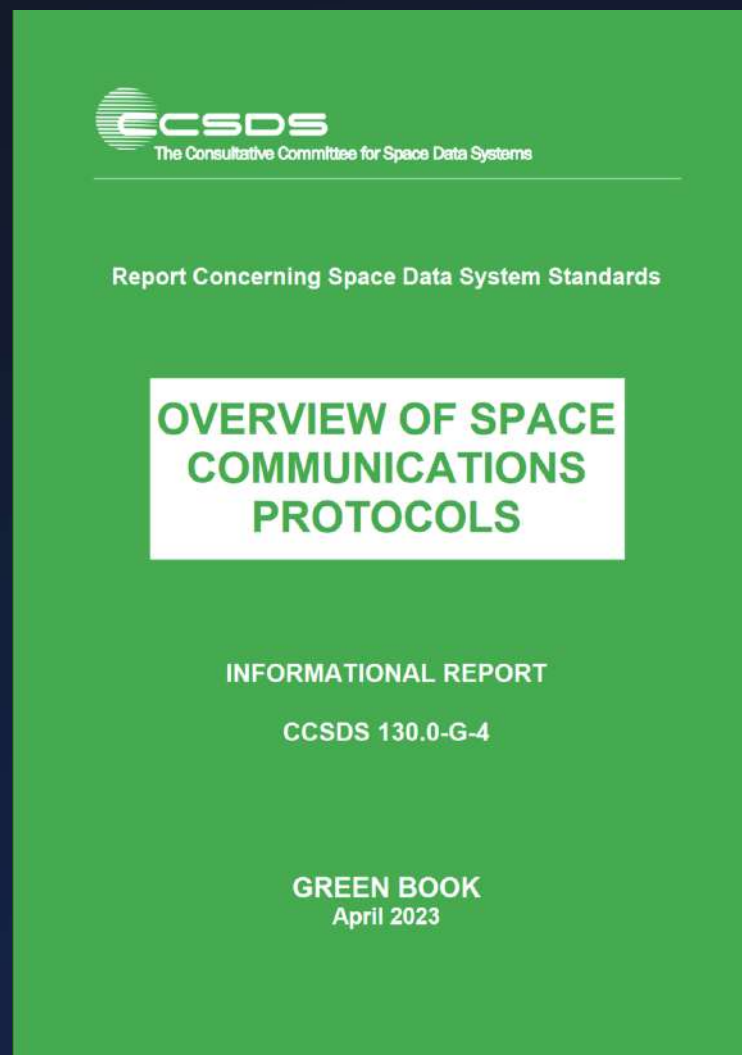
1. MAC comparison leaks timing data #44
 - memcmp to compare the digest
2. HMAC doesn't protect headers #45
 - Same problem for the CRC checks
3. XTEA encrypt packet nonce too predictable #162
 - `const uint32_t nonce = (uint32_t)rand();`

Authors: Issues fixed in libcsp v2

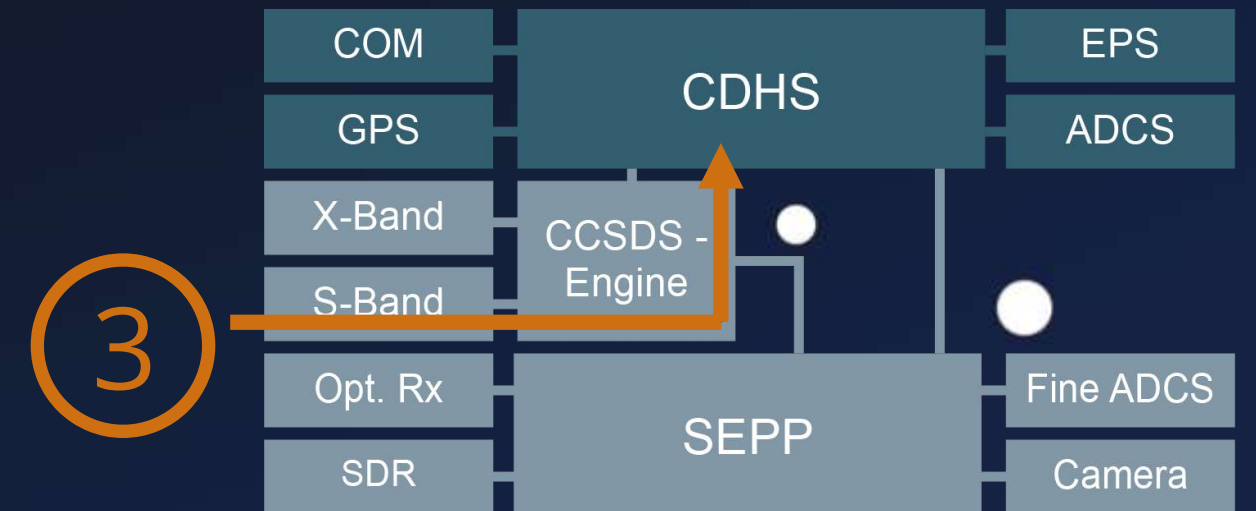
S-Band Stack



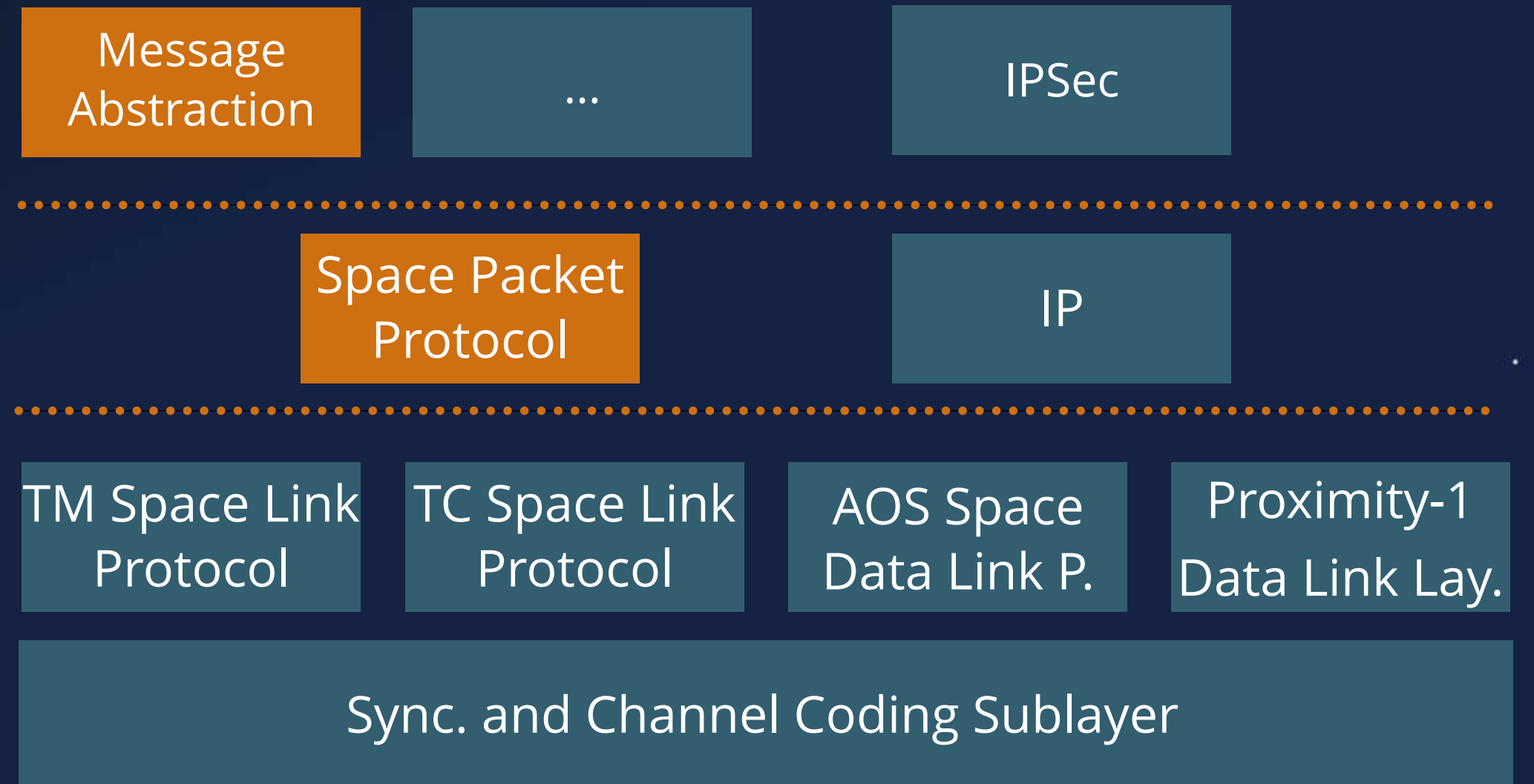
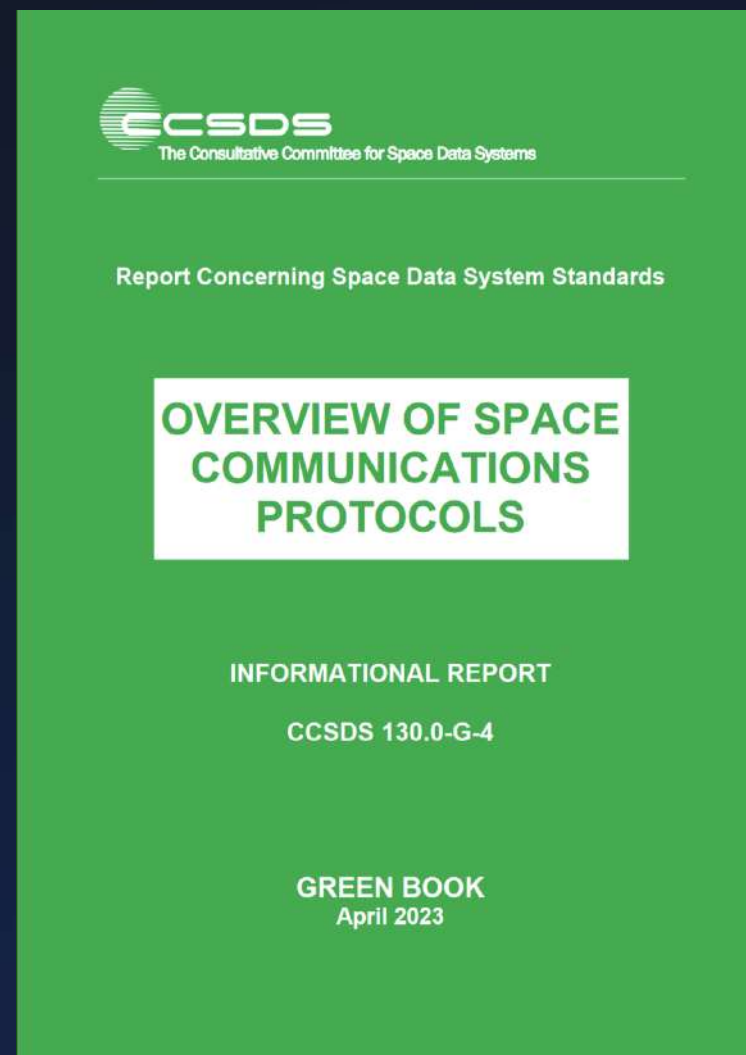
CCSDS - Protocol Stack



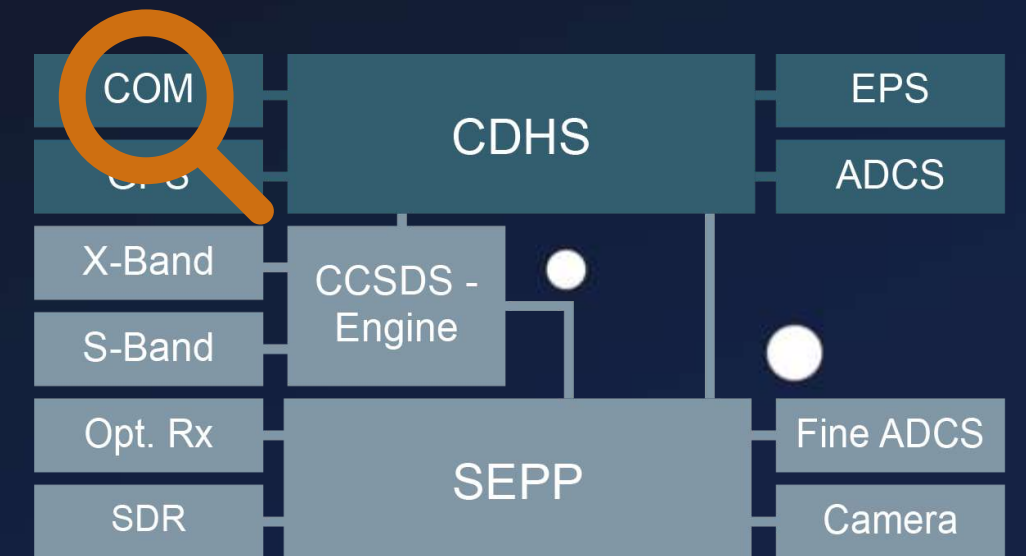
S-Band Stack



CCSDS - Protocol Stack

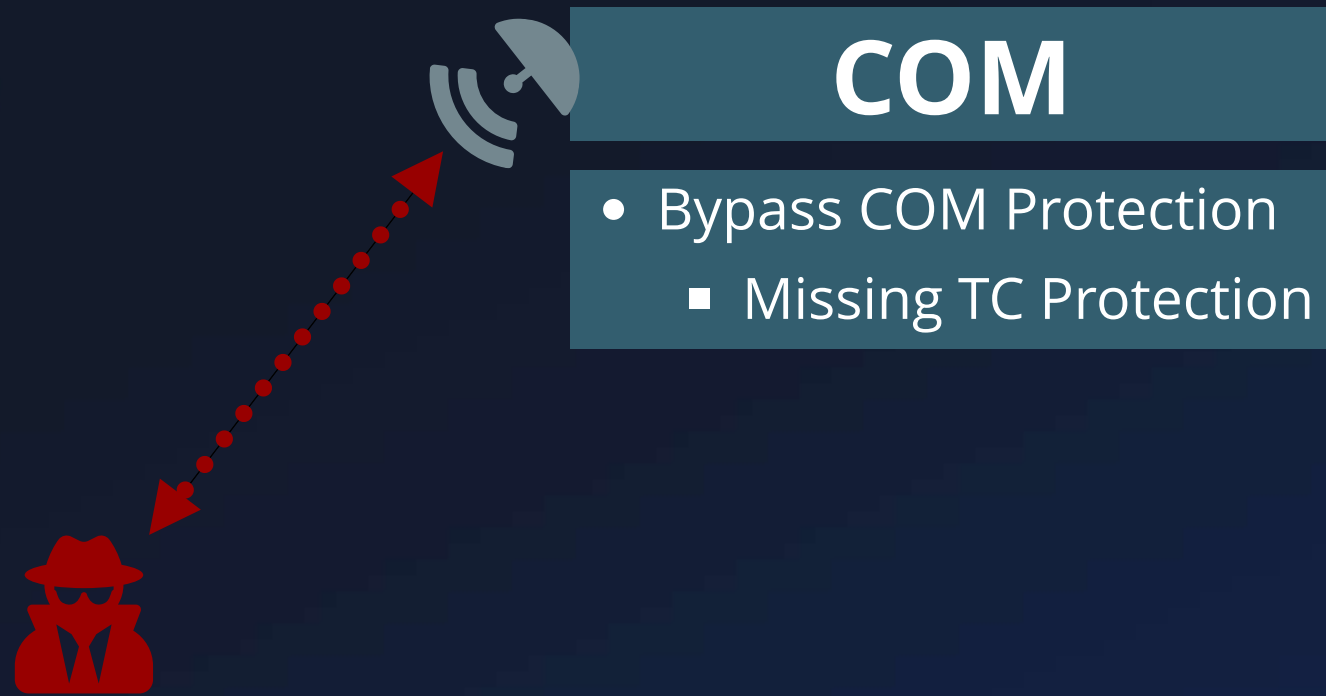


Unprotected TCs



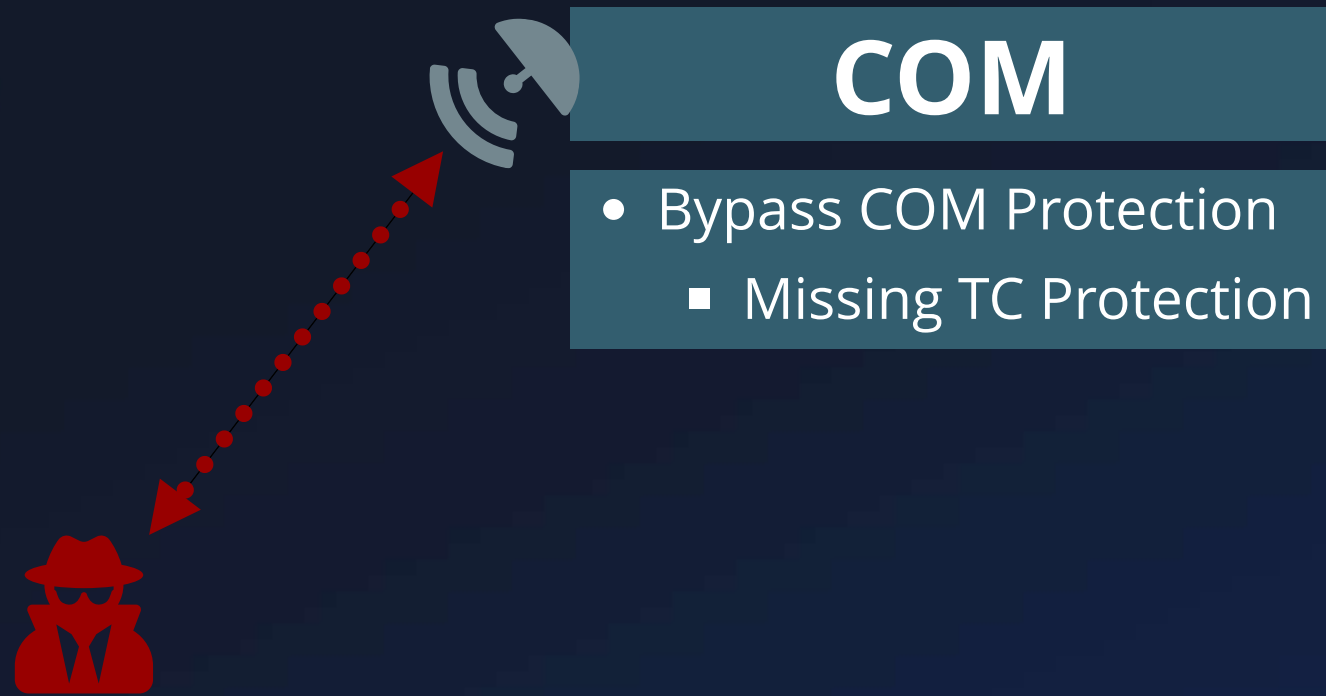
```
1 int csp_route_security_chek(...) {
2     if (packet->id.flags & CSP_FXTEA) {
3         csp_log_error("Received XTEA encrypted packet, but CSP
was compiled without XTEA support. Discarding packet");
4     }
5
6     // ...
7
8     if (packet->id.flags & CSP_FHMAC) {
9         csp_log_error("Received packet with HMAC, but CSP was
compiled without HMAC support. Discarding packet");
10    }
11
12    // ...
13 }
```

Unprotected TCs



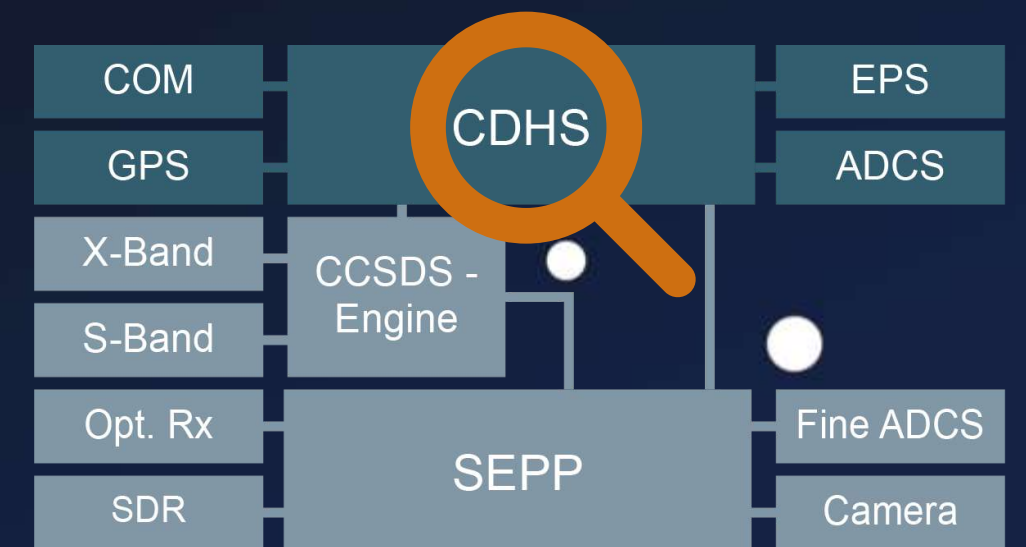
```
1 int sch_handler_set_raw_memory(scheduler_cmd_t* pCmd) {
2     raw_mem_access_cmd_t* pAddr = pCmd->pCmdArgs;
3     char* pWriteData;
4
5     if (pAddr) {
6         if (g_sch_exec_mode != 1 ) {
7             /* exception and return */
8         }
9         char* pWriteData = &pAddr->start_of_data_buf;
10        if (pAddr->filesystem_target) {
11            // [...]
12        } else {
13            memcpy(pAddr->targetAddr,
14                  &pAddr->start_of_data_buf,
15                  pAddr->writeLength);
16        }
17    }
18    // ...
19 }
```

Unprotected TCs

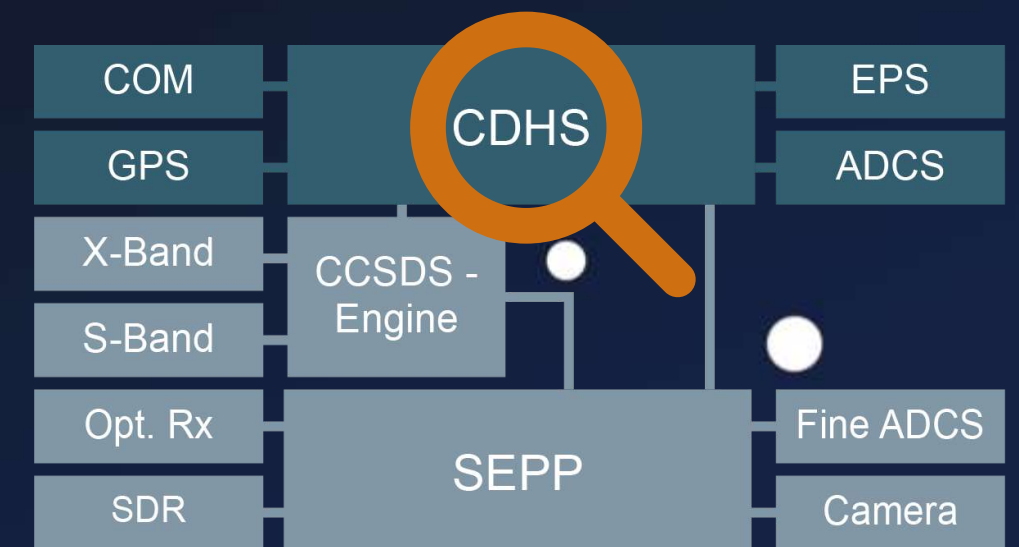
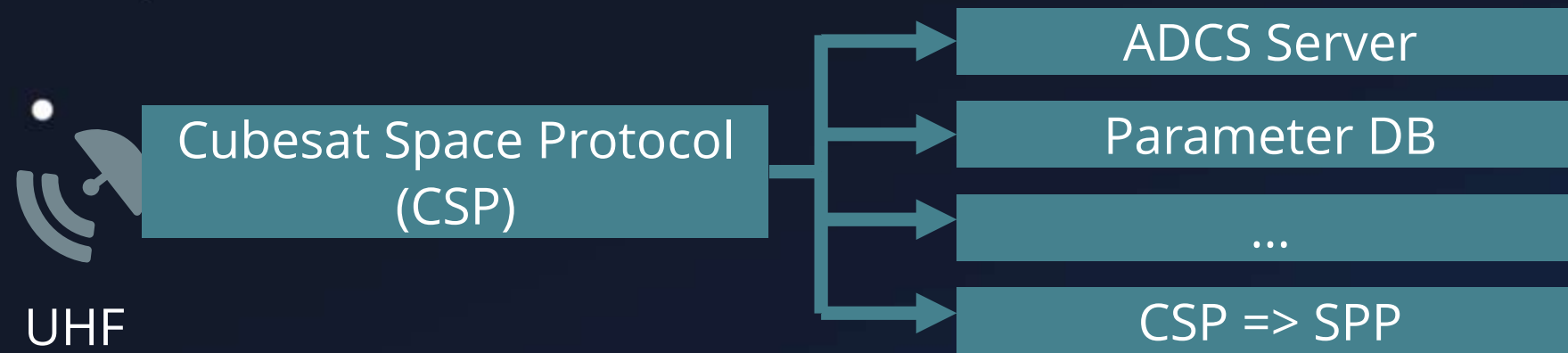


```
1 int sch_handler_set_raw_memory(scheduler_cmd_t* pCmd) {
2     raw_mem_access_cmd_t* pAddr = pCmd->pCmdArgs;
3     char* pWriteData;
4
5     if (pAddr) {
6         if (g_sch_exec_mode != 1 ) {
7             /* exception and return */
8         }
9         char* pWriteData = &pAddr->start_of_data_buf;
10        if (pAddr->filesystem_target) {
11            // [...]
12        } else {
13            memcpy(pAddr->targetAddr,
14                  &pAddr->start_of_data_buf,
15                  pAddr->writeLength);
16        }
17    }
18    // ...
19 }
```

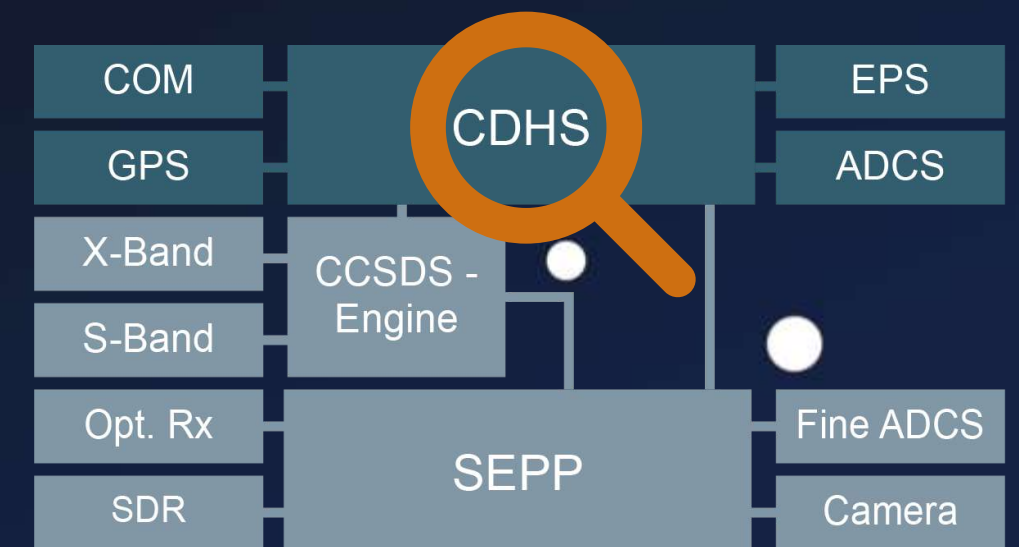
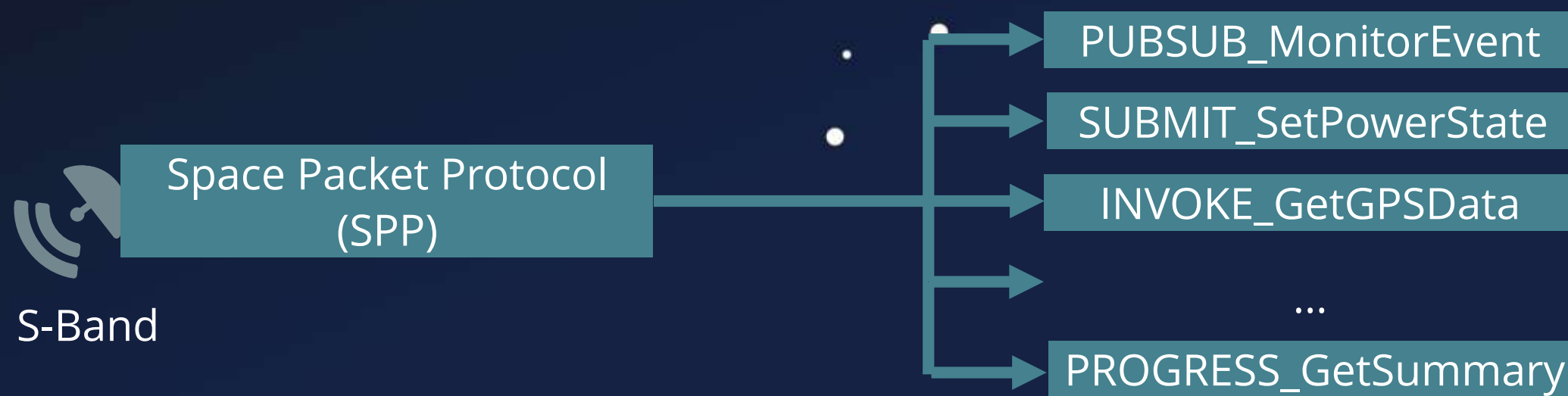
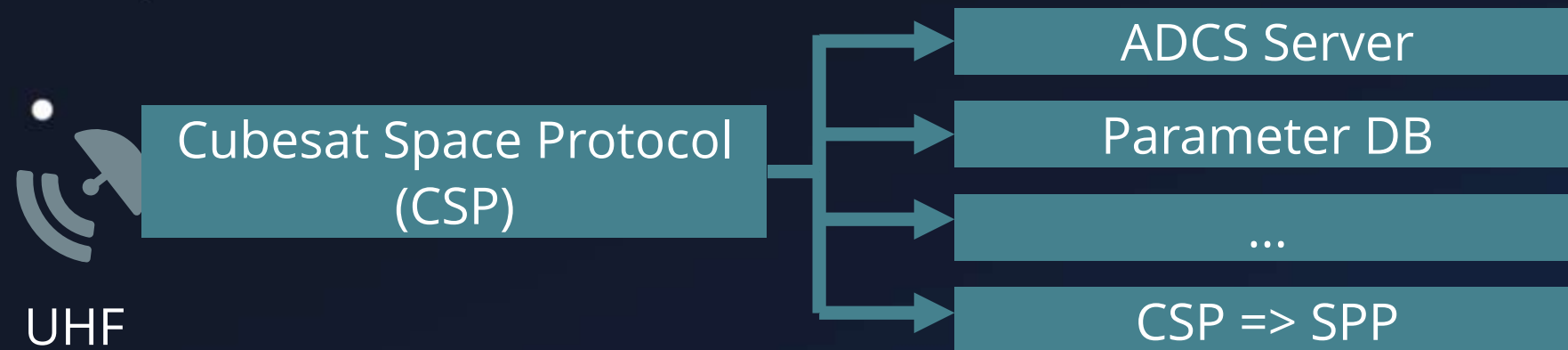

Vulnerable TC



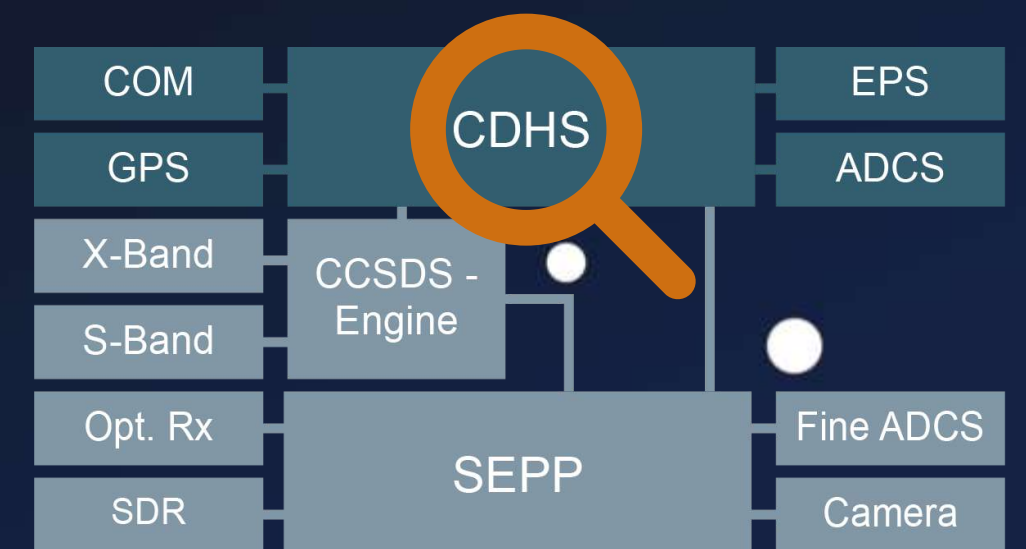
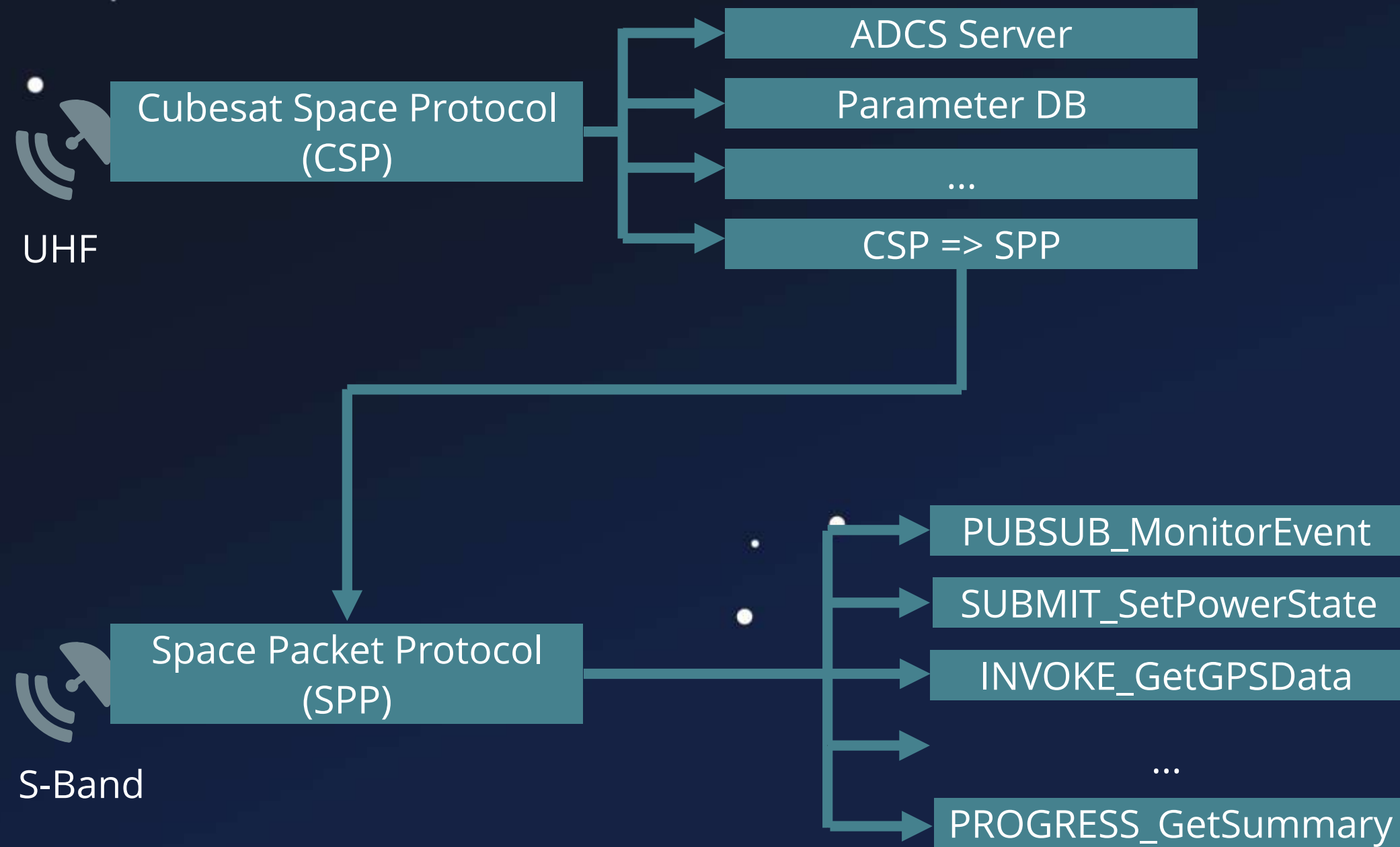
Vulnerable TC



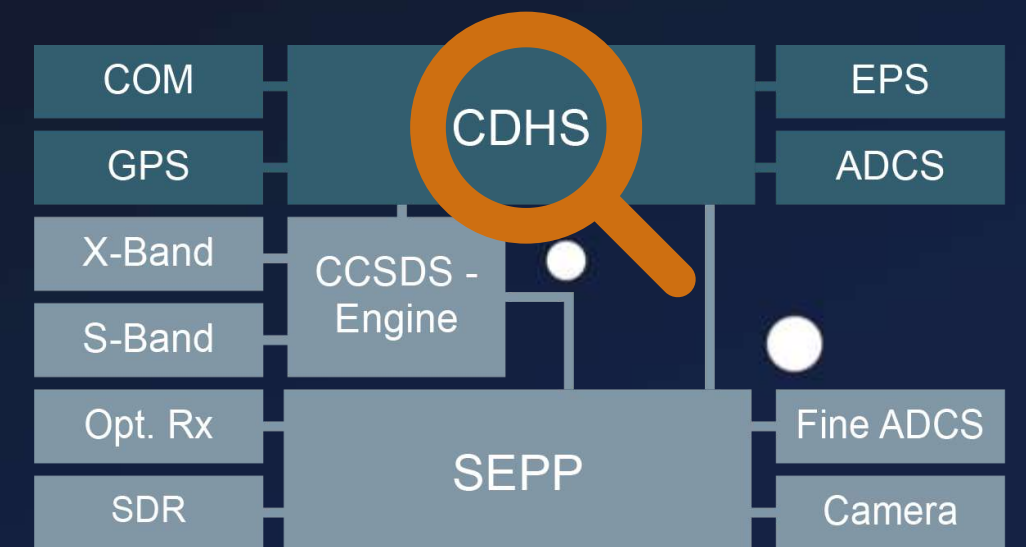
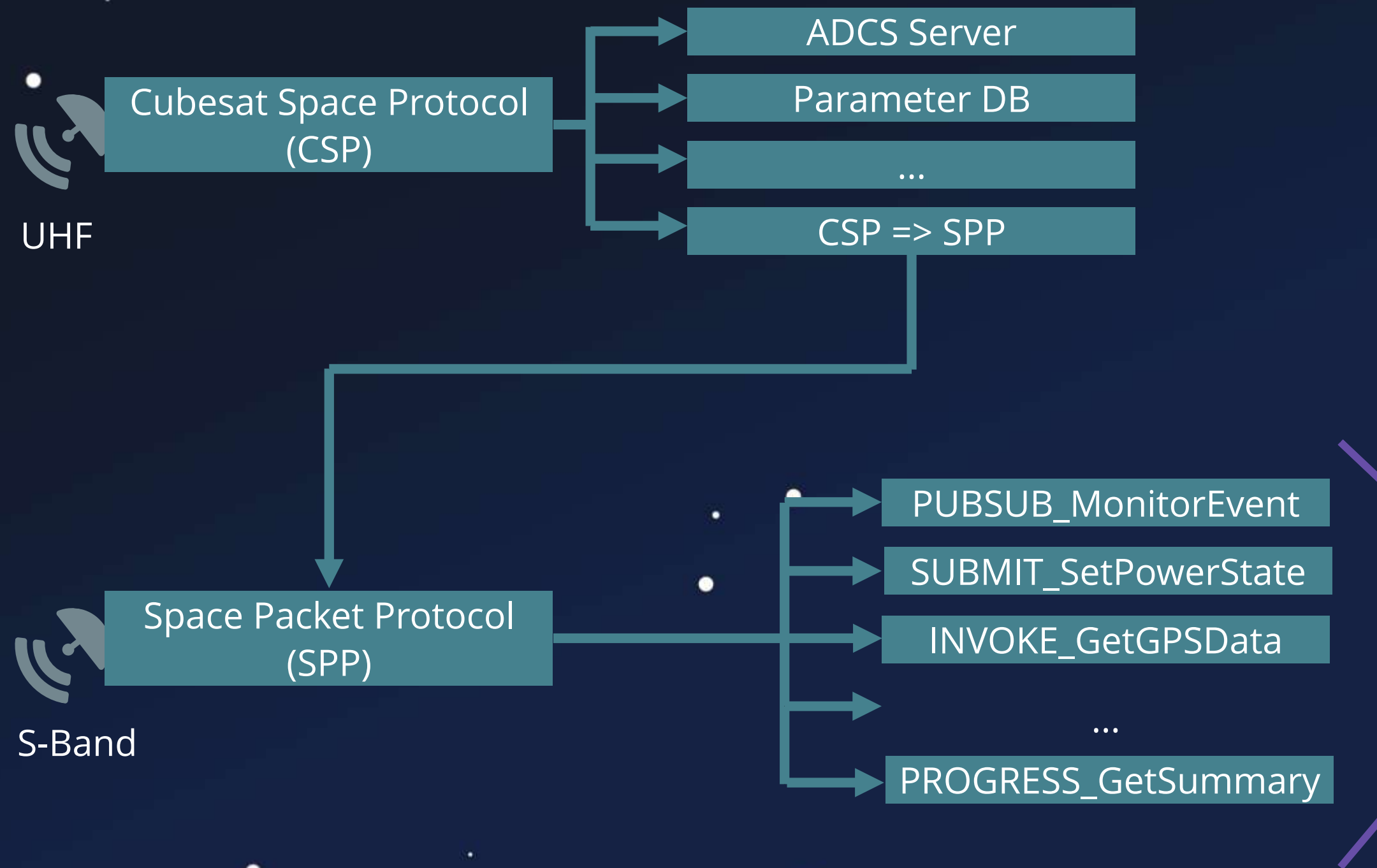
Vulnerable TC



Vulnerable TC

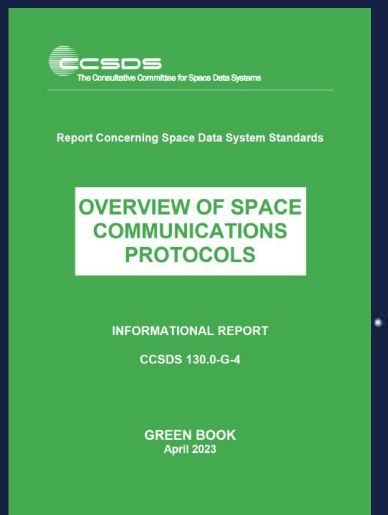


Vulnerable TC

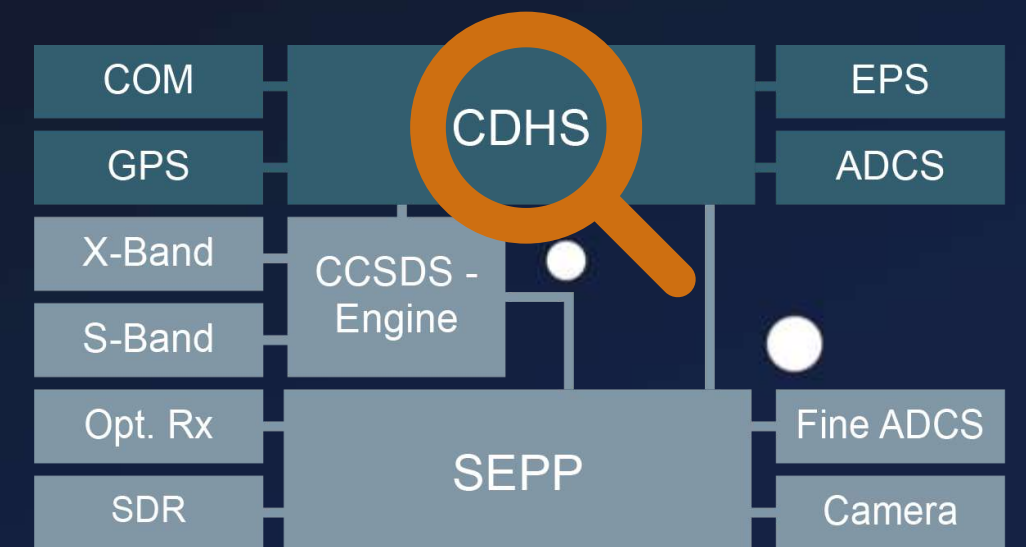
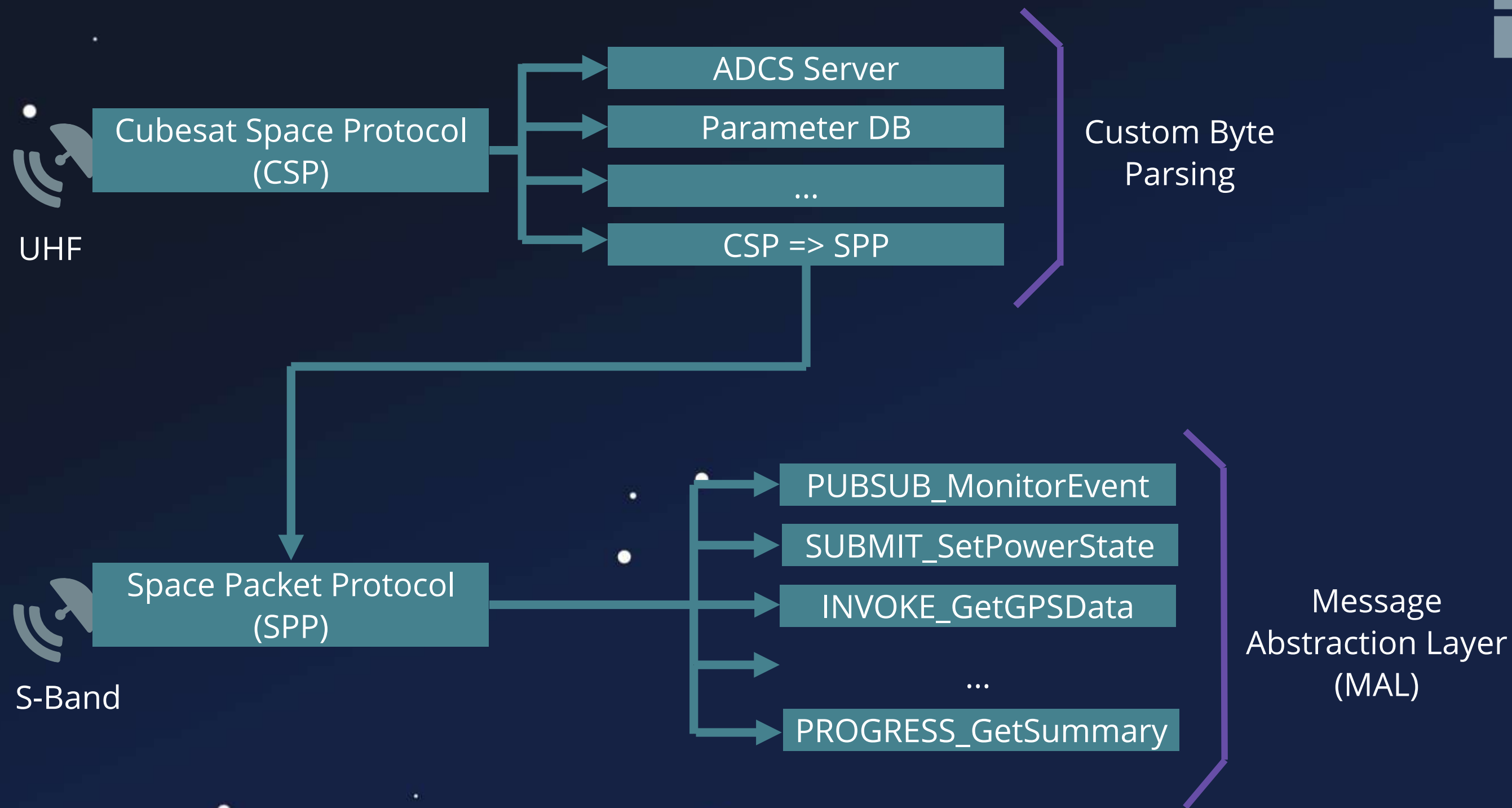


Message
Abstraction

Space Packet
Protocol

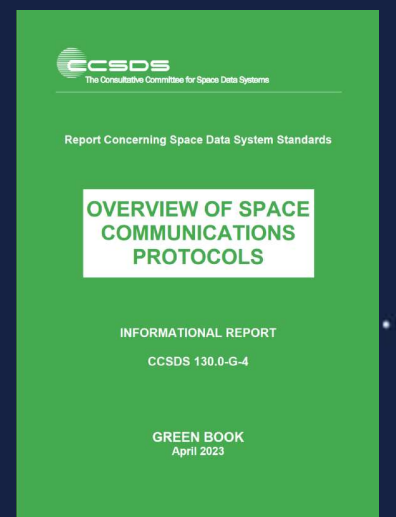


Vulnerable TC

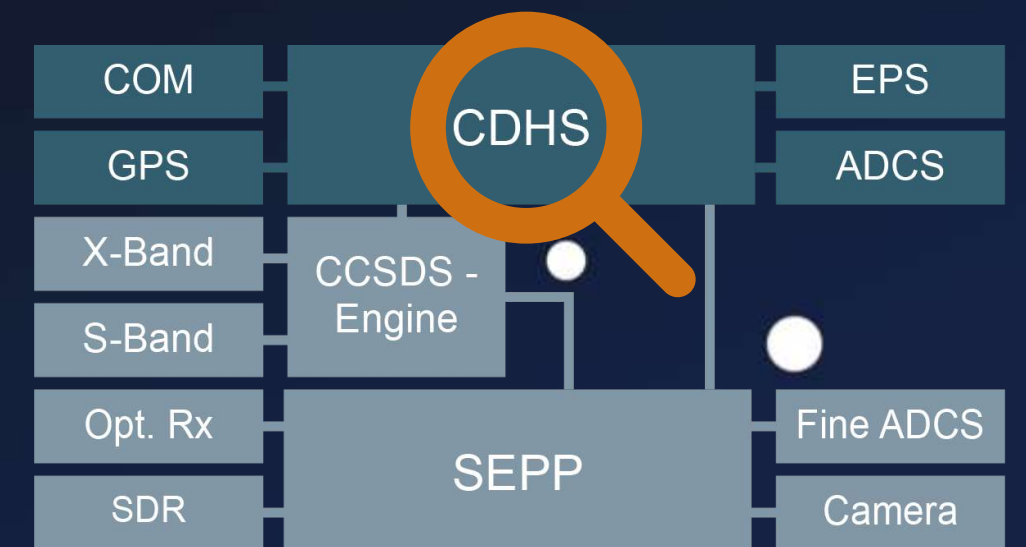
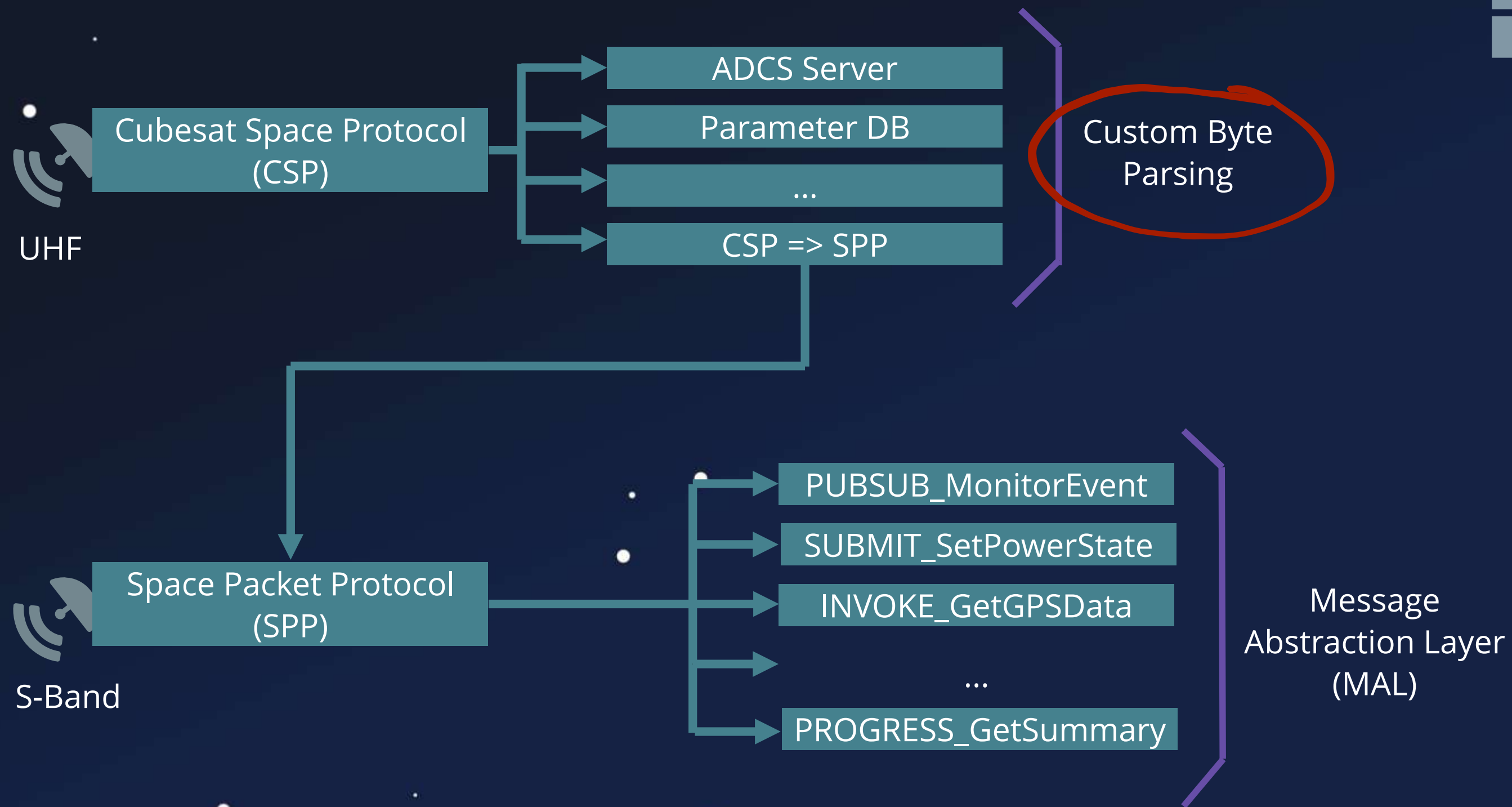


Message Abstraction

Space Packet Protocol



Vulnerable TC



Message
Abstraction

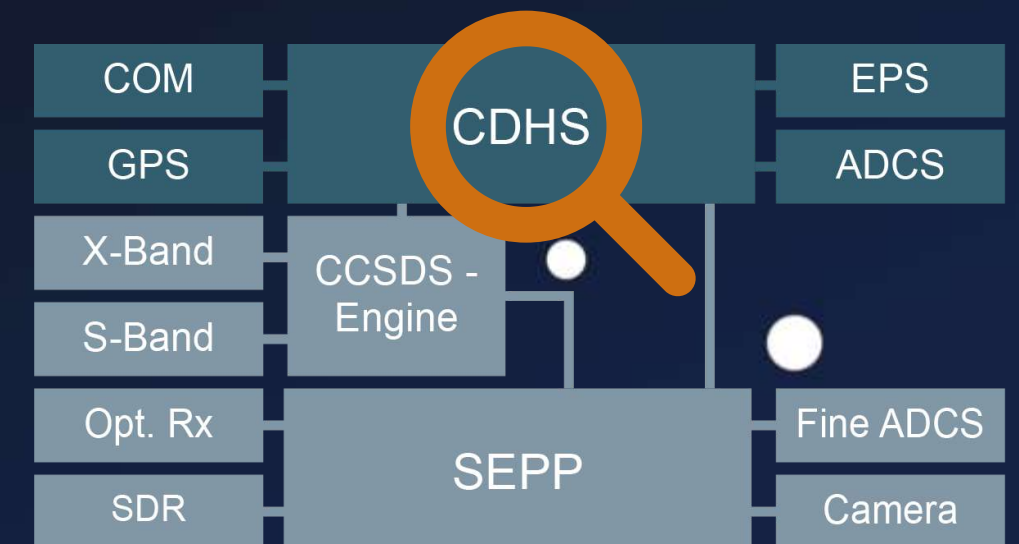
Space Packet
Protocol



Vulnerable TC

Cubesat Space Protocol (CSP) → ADCS Server

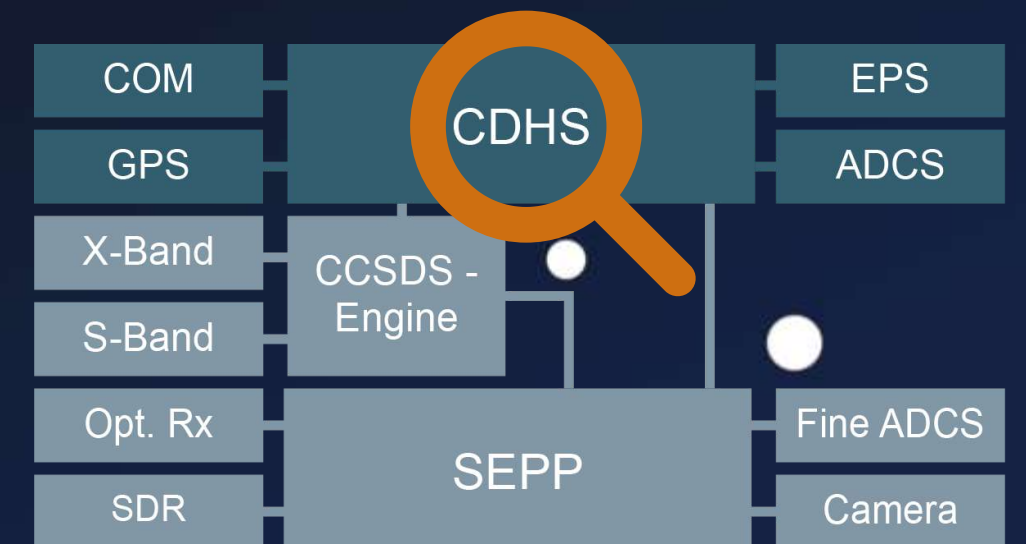
```
1 void task_adcs_servr() {
2     char log_file_name [32];
3
4     csp_listen(socket, 10);
5     csp_bind(socket, port);
6
7     do {
8         do {
9             conn = csp_accept(socket, 0xff);
10        } while (do_wait_for_conn);
11
12        packet = csp_read(conn, 10);
13        if (packet) {
14            packet_data = packet->data;
15            switch(*packet_data) {
16                // [...]
17                case SET_LOGFILE: {
18                    packet_data = packet->data + 0xf;
19                    log_file_name[0] = '\0';
20                    strcat(log_file_name, packet_data);
21                    // ...
22                }
23            }
24        }
25    }
```



Vulnerable TC

Cubesat Space Protocol (CSP) → ADCS Server

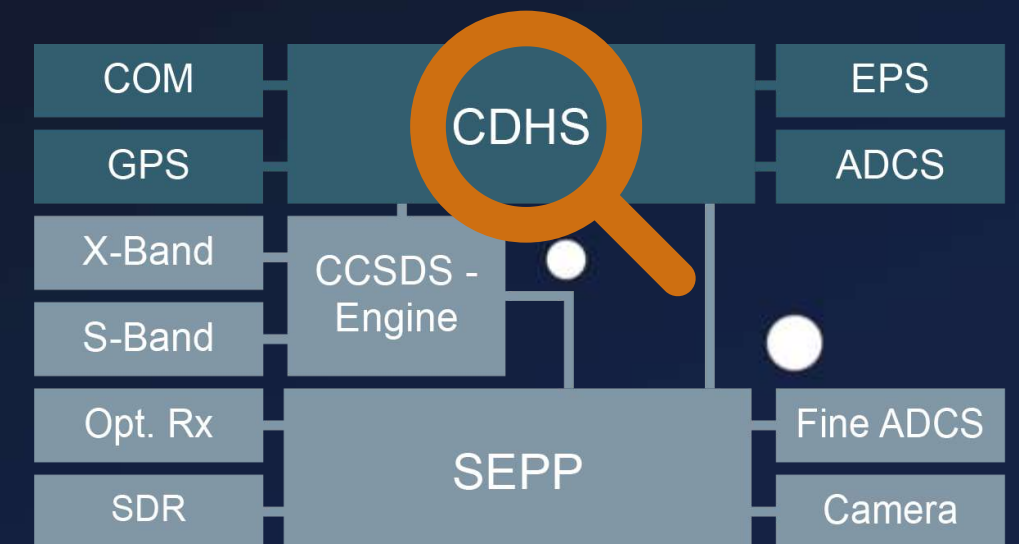
```
1 void task_adcs_servr() {
2     char log_file_name [32];
3
4     csp_listen(socket, 10);
5     csp_bind(socket, port);
6
7     do {
8         do {
9             conn = csp_accept(socket, 0xff);
10        } while (do_wait_for_conn);
11
12        packet = csp_read(conn, 10);
13        if (packet) {
14            packet_data = packet->data;
15            switch(*packet_data) {
16                // [...]
17                case SET_LOGFILE: {
18                    packet_data = packet->data + 0xf;
19                    log_file_name[0] = '\0';
20                    strcat(log_file_name, packet_data);
21                    // ...
22                }
23            }
24        }
25    }
```



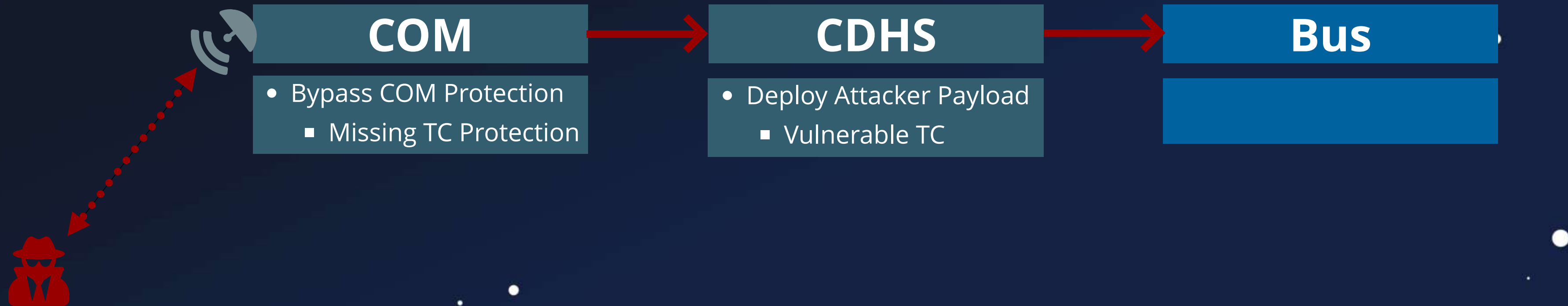
Vulnerable TC

Cubesat Space Protocol (CSP) → ADCS Server

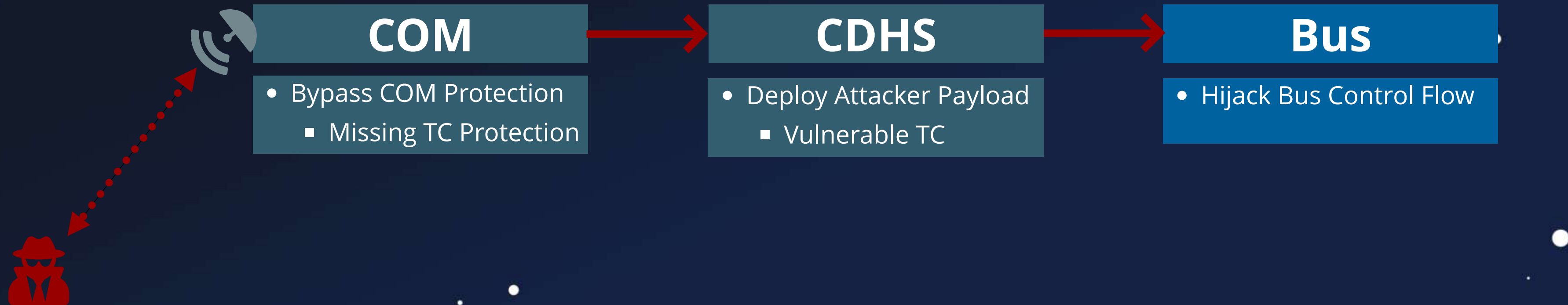
```
1 void task_adcs_servr() {
2     char log_file_name [32];
3
4     csp_listen(socket, 10);
5     csp_bind(socket, port);
6
7     do {
8         do {
9             conn = csp_accept(socket, 0xff);
10        } while (do_wait_for_conn);
11
12        packet = csp_read(conn, 10);
13        if (packet) {
14            packet_data = packet->data;
15            switch(*packet_data) {
16                // [...]
17                case SET_LOGFILE: {
18                    packet_data = packet->data + 0xf;
19                    log_file_name[0] = '\0';
20                    strcat(log_file_name, packet_data);
21                    // ...
22                }
23            }
24        }
25    }
```



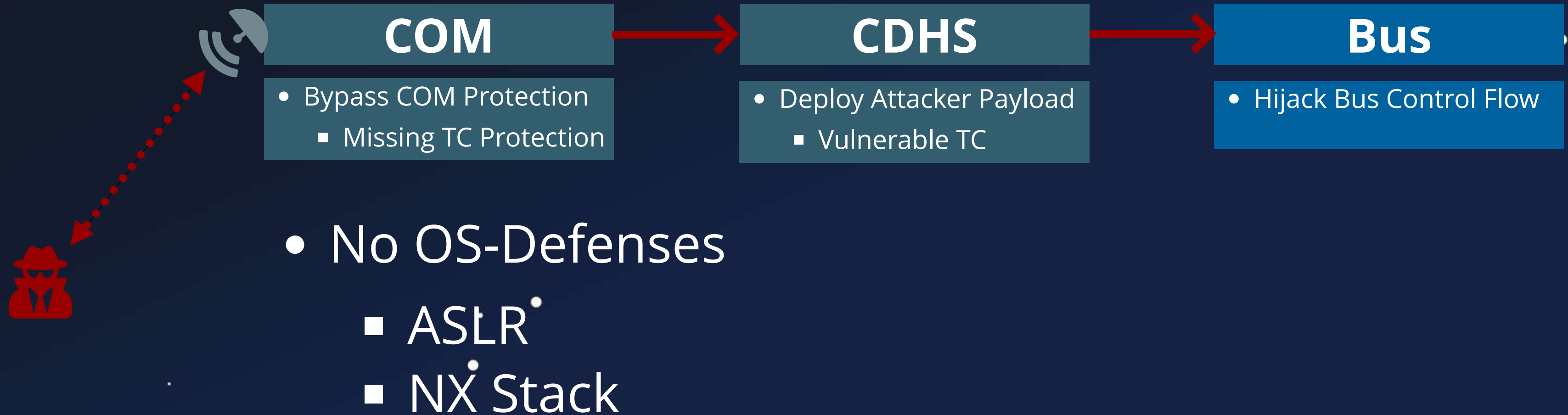
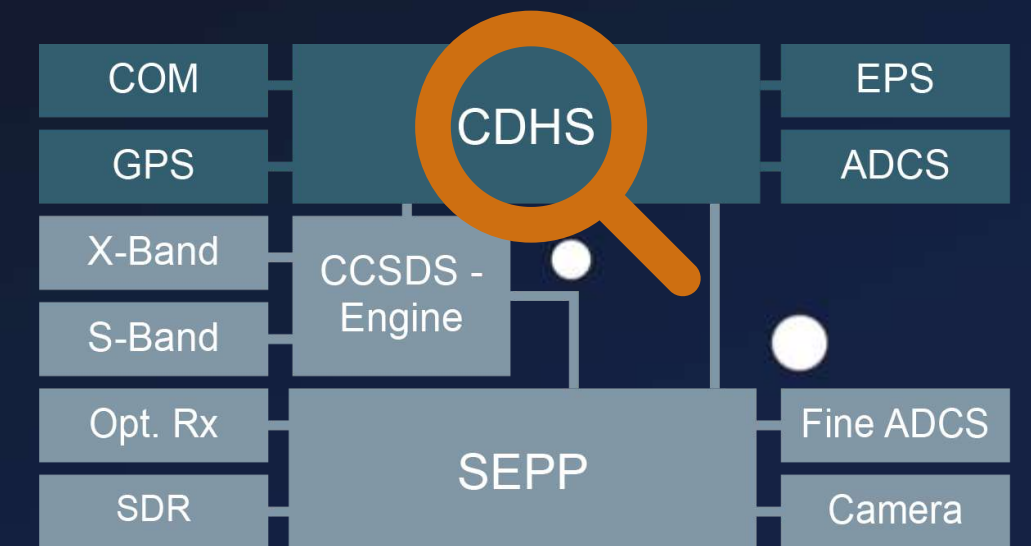
Defenses - 404?



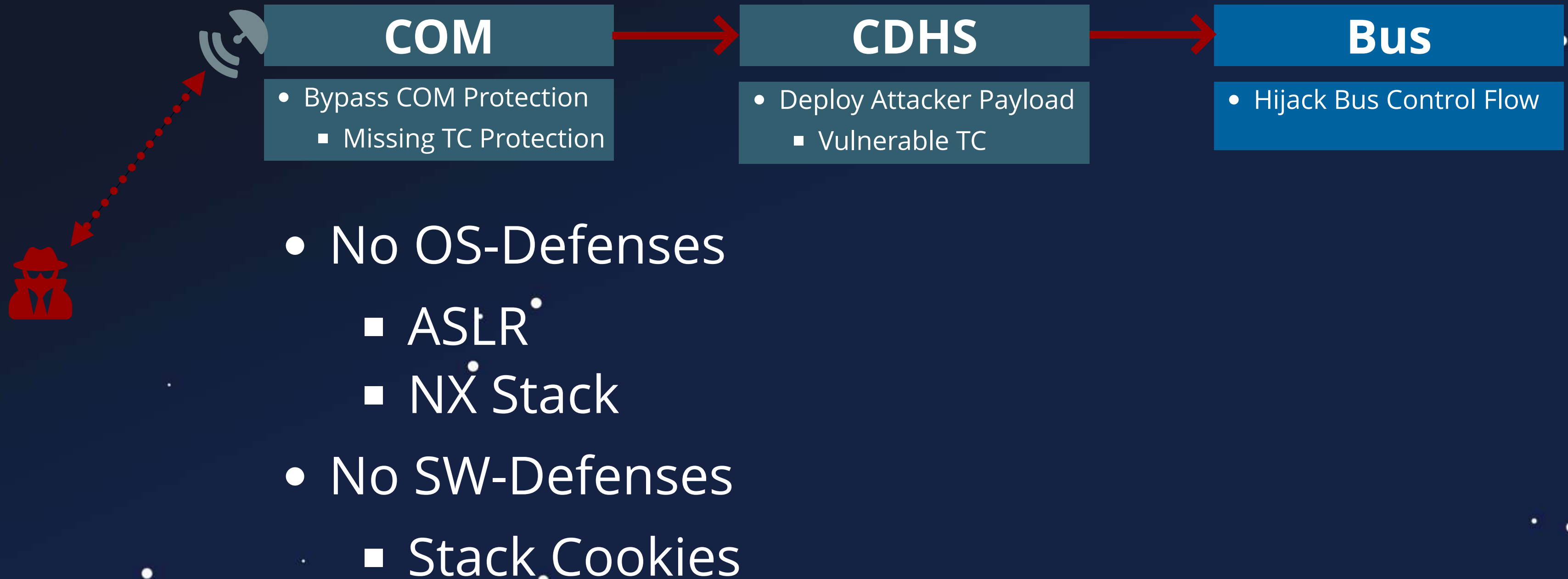
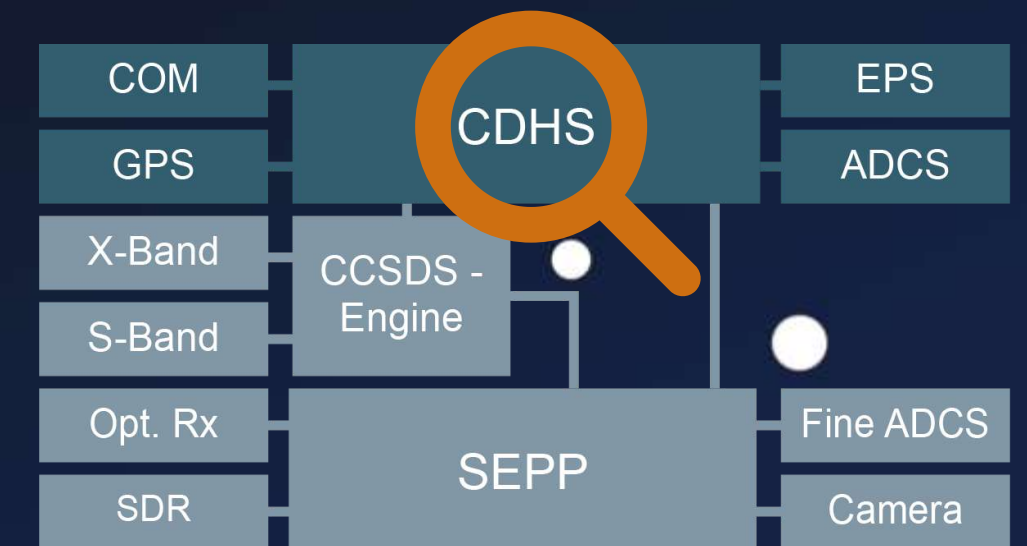
Defenses - 404?



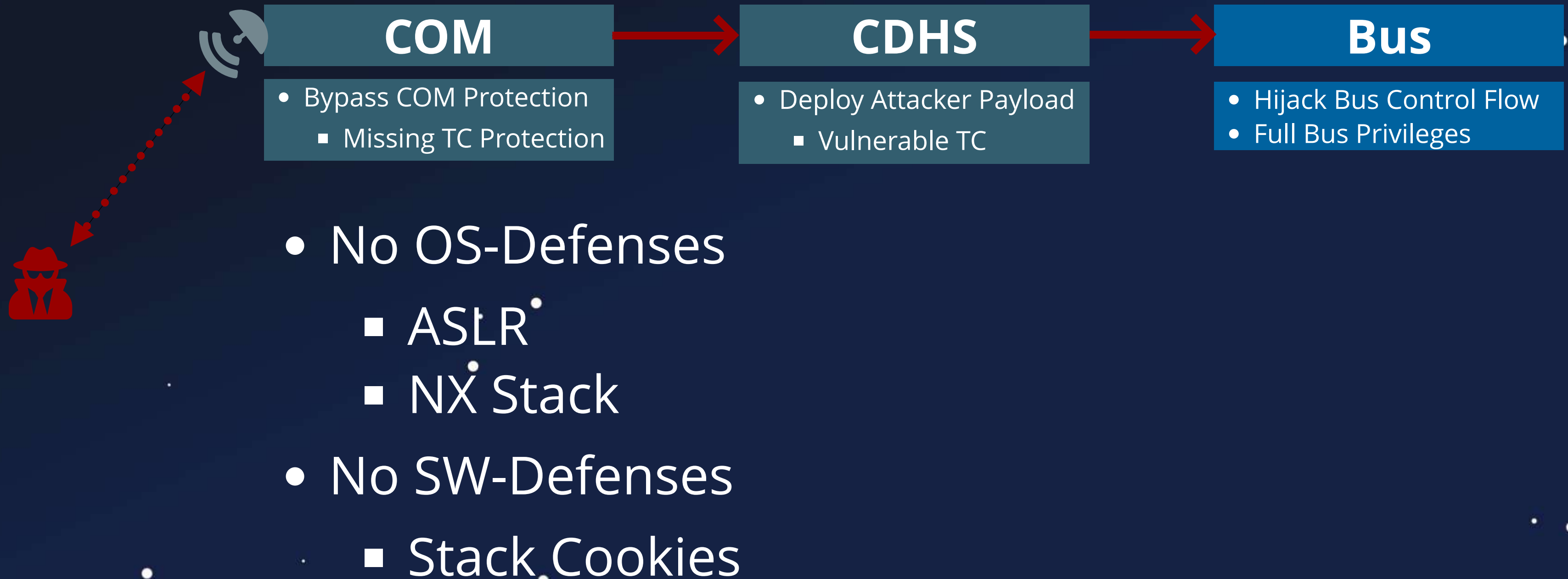
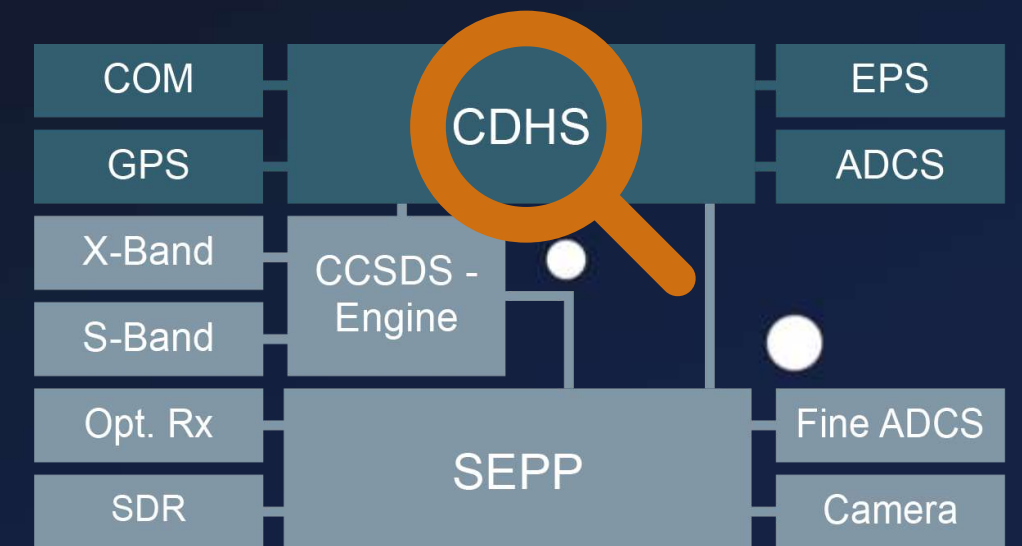
Defenses - 404?



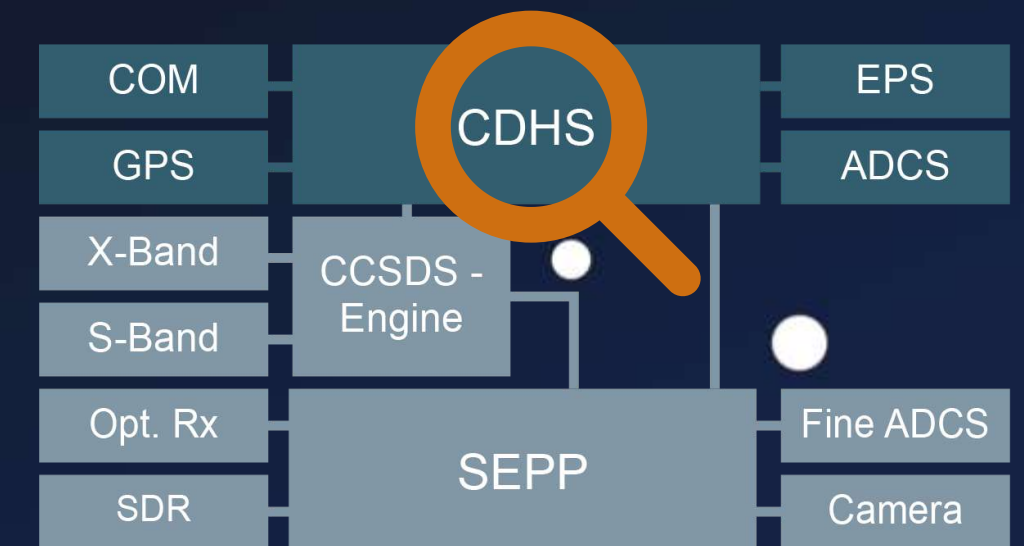
Defenses - 404?



Defenses - 404?



Defenses - 404?



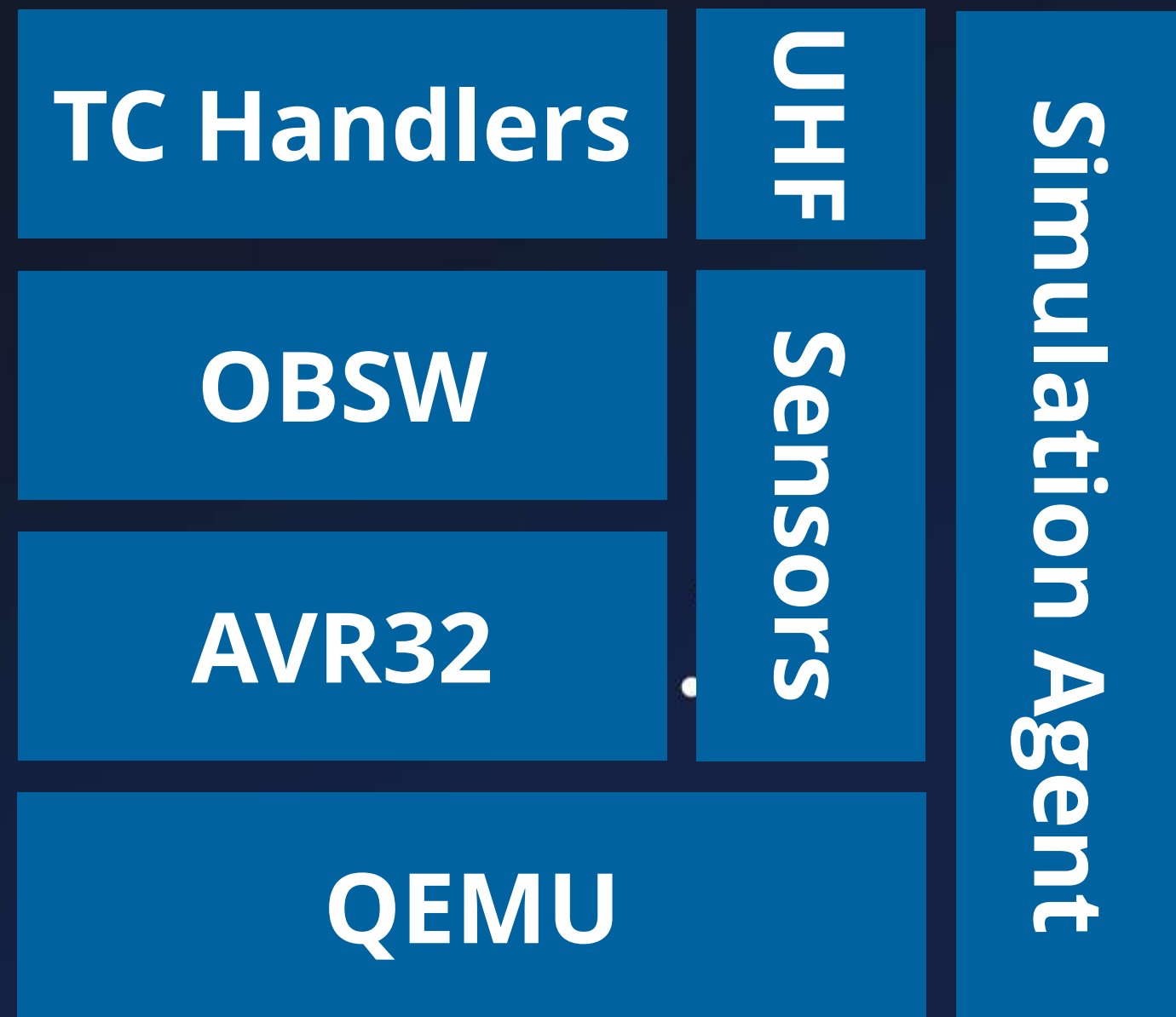
- No OS-Defenses
 - ASLR
 - NX Stack
- No SW-Defenses
 - Stack Cookies

- Privilege-free RTOS

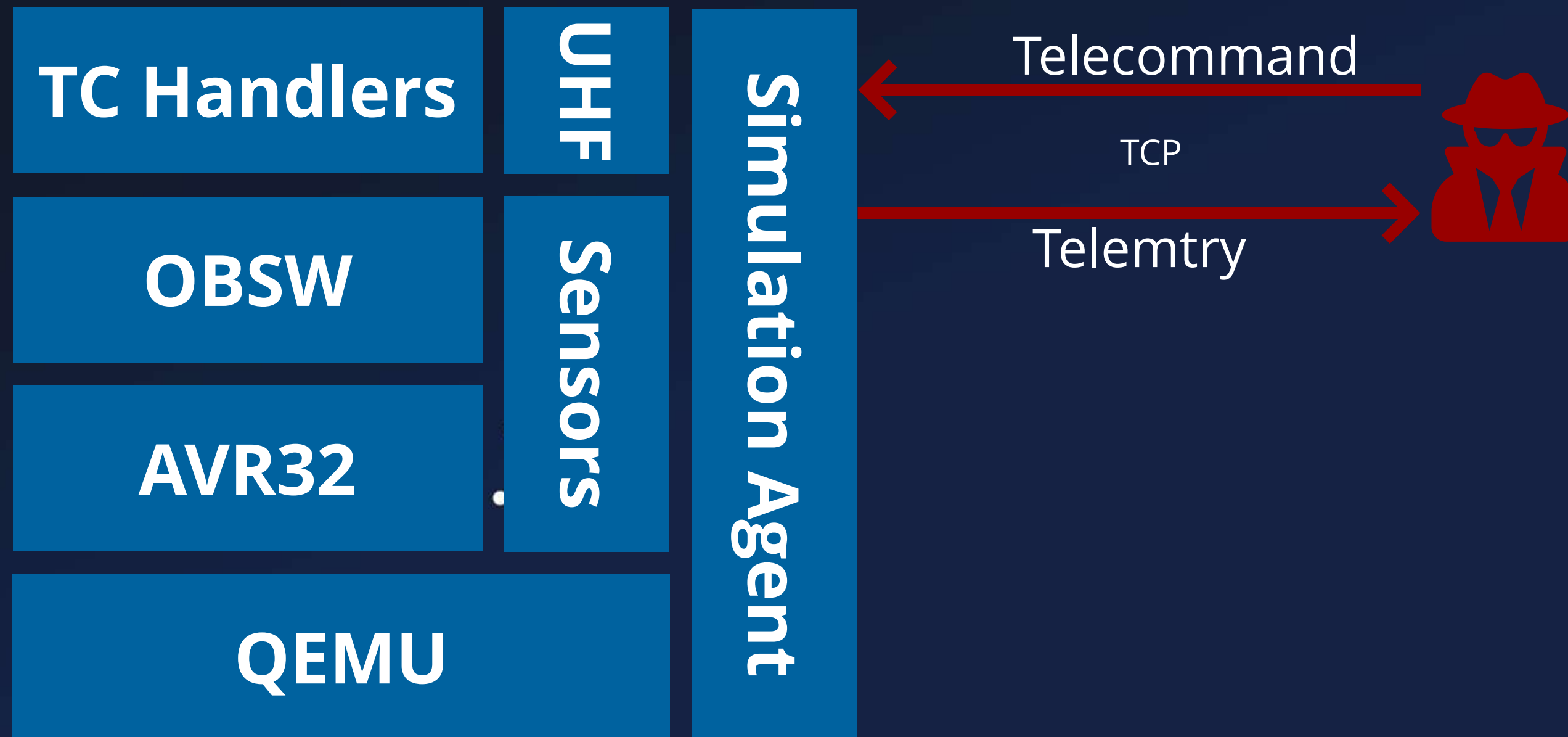
Demo Setup



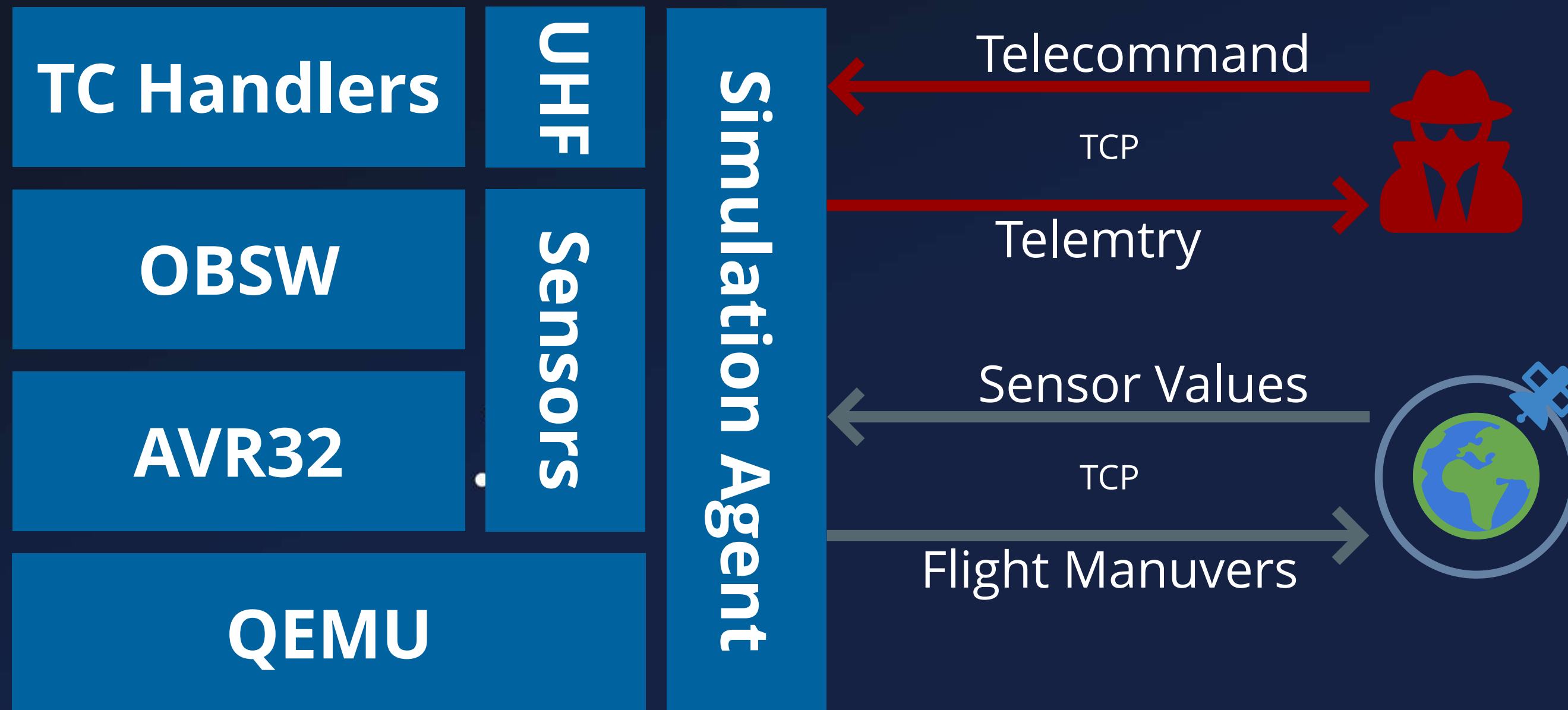
Emulation Overview



Emulation Overview



Emulation Overview



AVR32-QEMU

404 - AVR32 Not Found

AVR32

QEMU

AVR32-QEMU

404 - AVR32 Not Found

AVR32

QEMU

RUHR
UNIVERSITÄT
BOCHUM

RUB

RUHR-UNIVERSITÄT BOCHUM

Hacking the Stars: A Fuzzing Based Security
Assessment of CubeSat Firmware

Florian Göhler

Master's Thesis – December 22, 2022,
Chair for System Security.

1st Supervisor: Prof. Dr. Thorsten Holz
2nd Supervisor: M.Sc. Johannes Willbold

hg:SYSSEC



AVR32-QEMU

404 - AVR32 Not Found

AVR32

QEMU



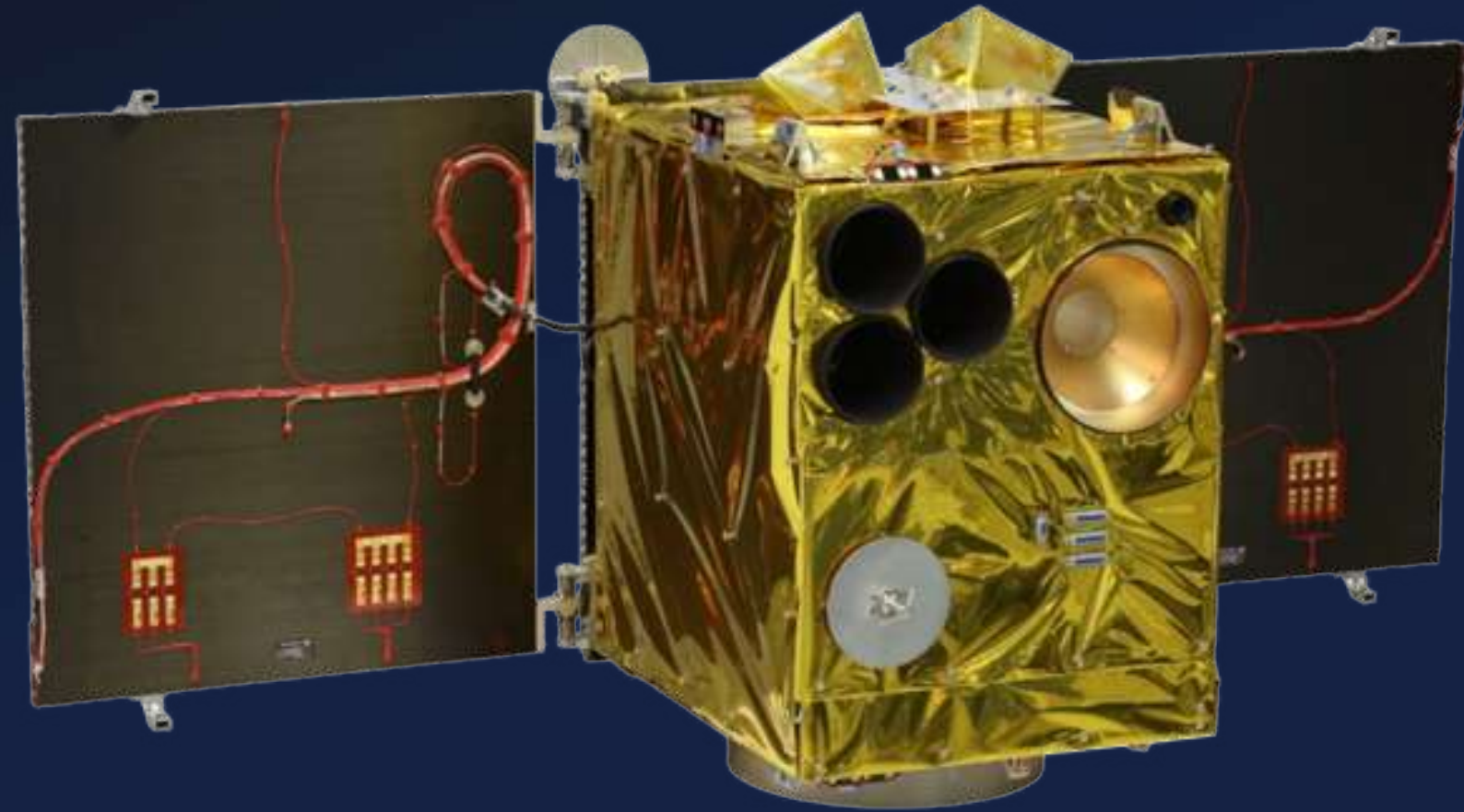
- Florian Göhler
- AVR32 in QEMU from Scratch
- Incl. I2C, SPI, PDCA, etc.
- Blog:
 - *How to add a new architecture to QEMU - Part 1-4*

Live Demo

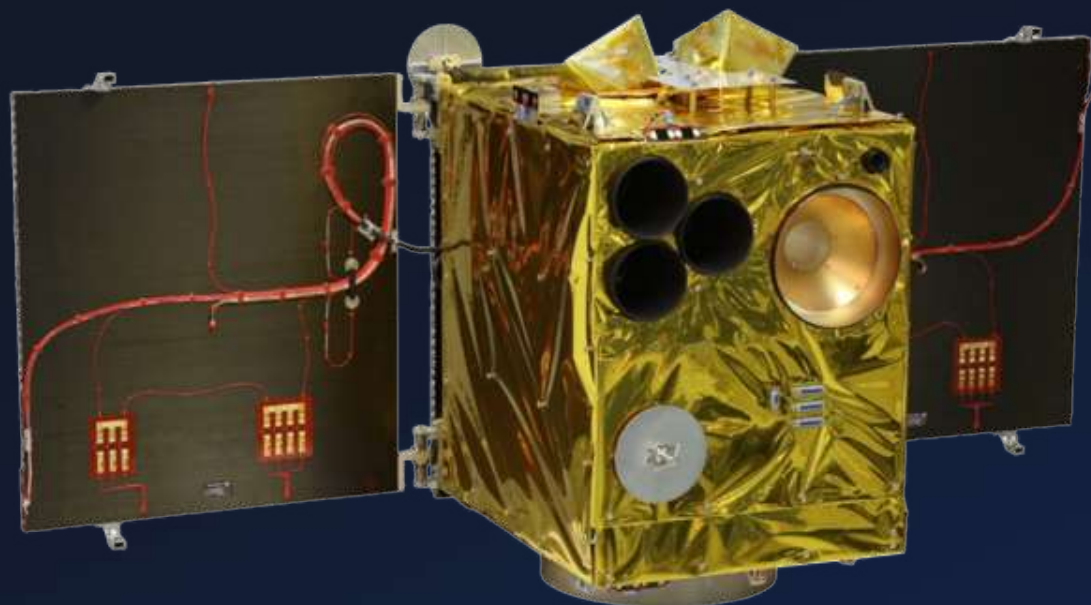


```
1 $> ./access-satellite.  
2 [*] Uploading TC ...  
3 [*] Deploying payload ...  
4 [*] Payload written to flash ...  
5 [*] Rebooting ...  
6 [*] $$$
```

Flying Laptop



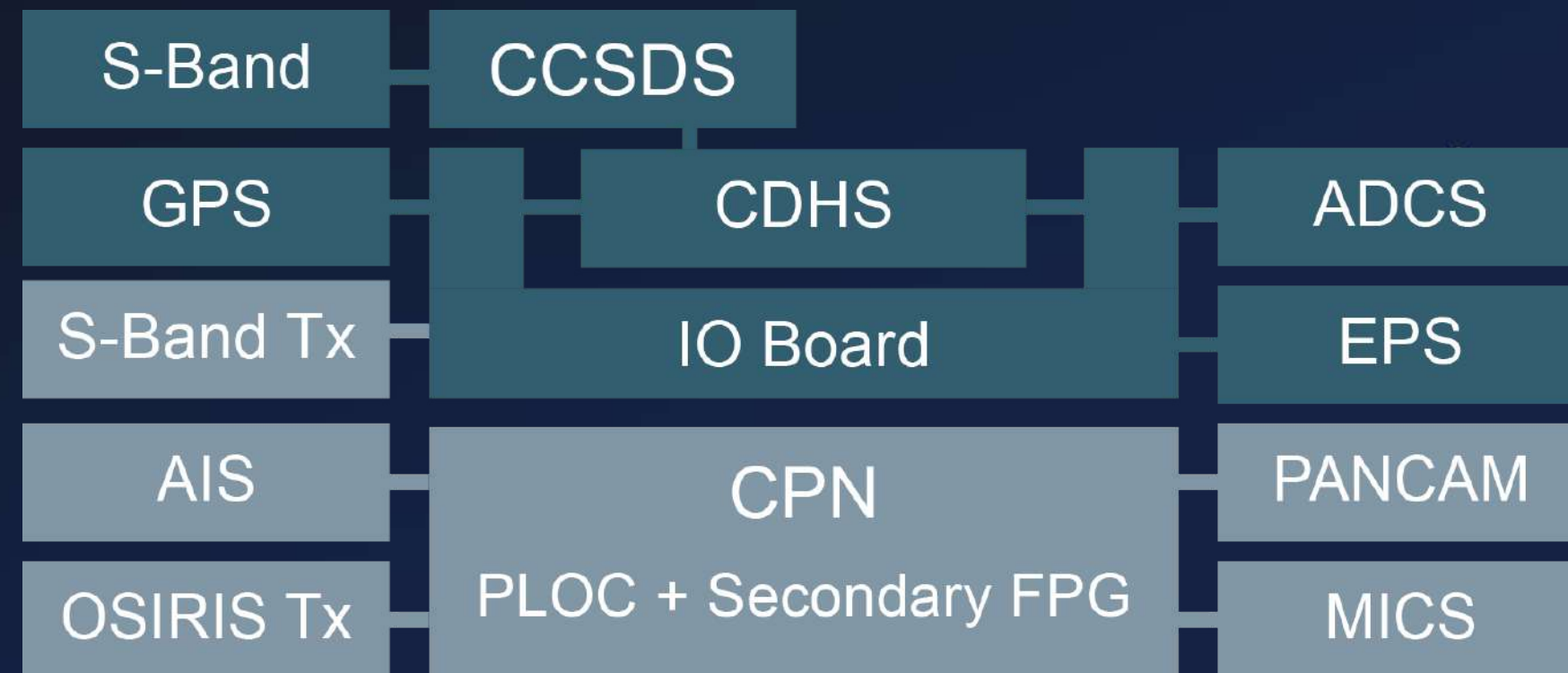
Flying Laptop



[3]

Technology Tester

Co-Developed by
Airbus Space & Defense



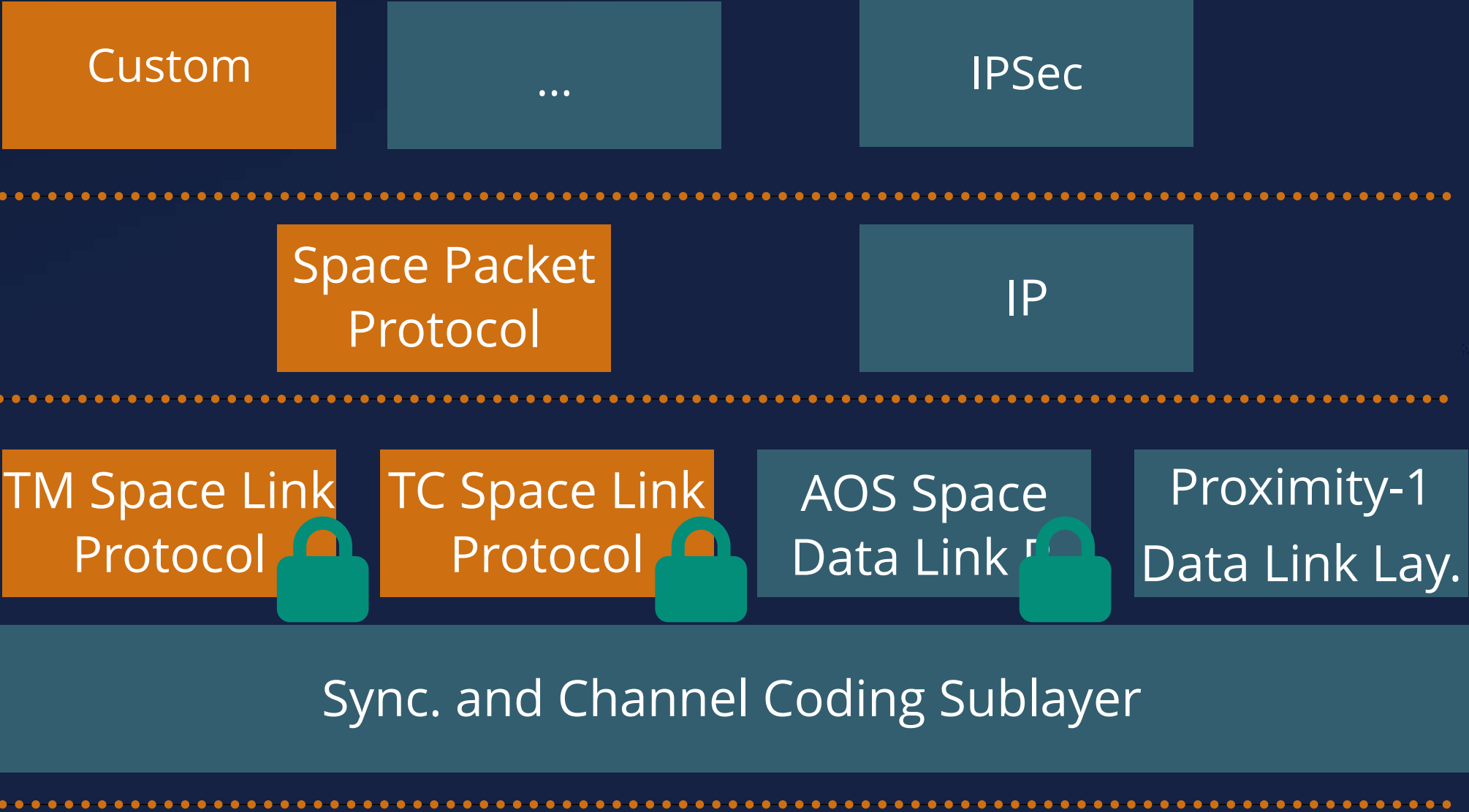
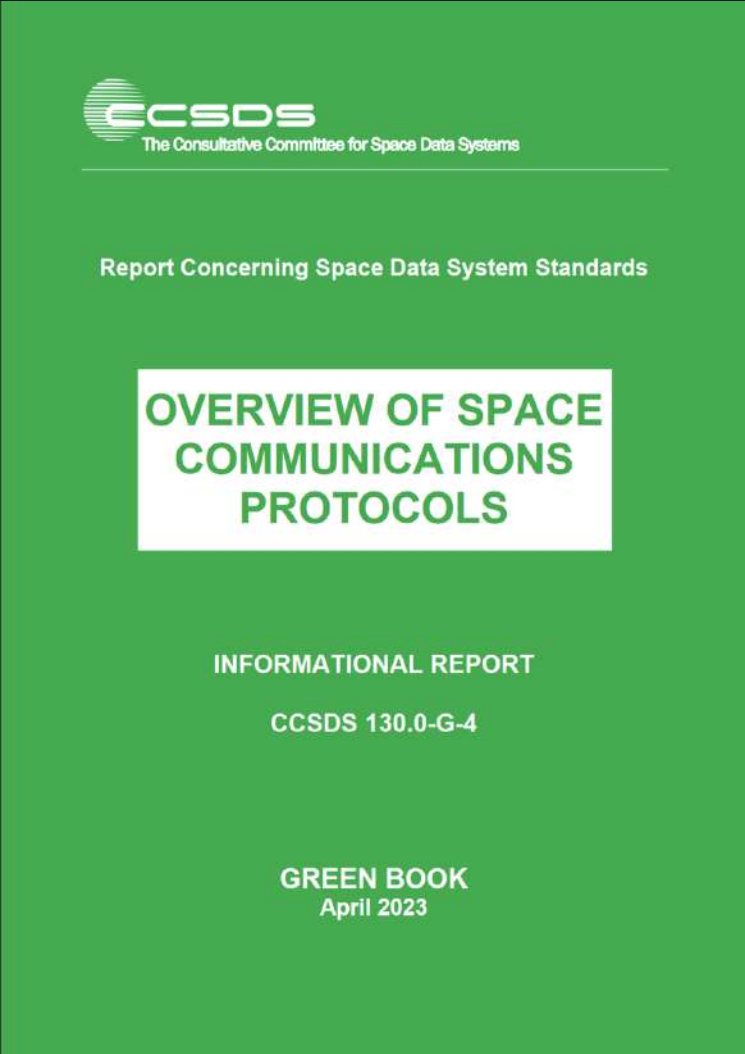
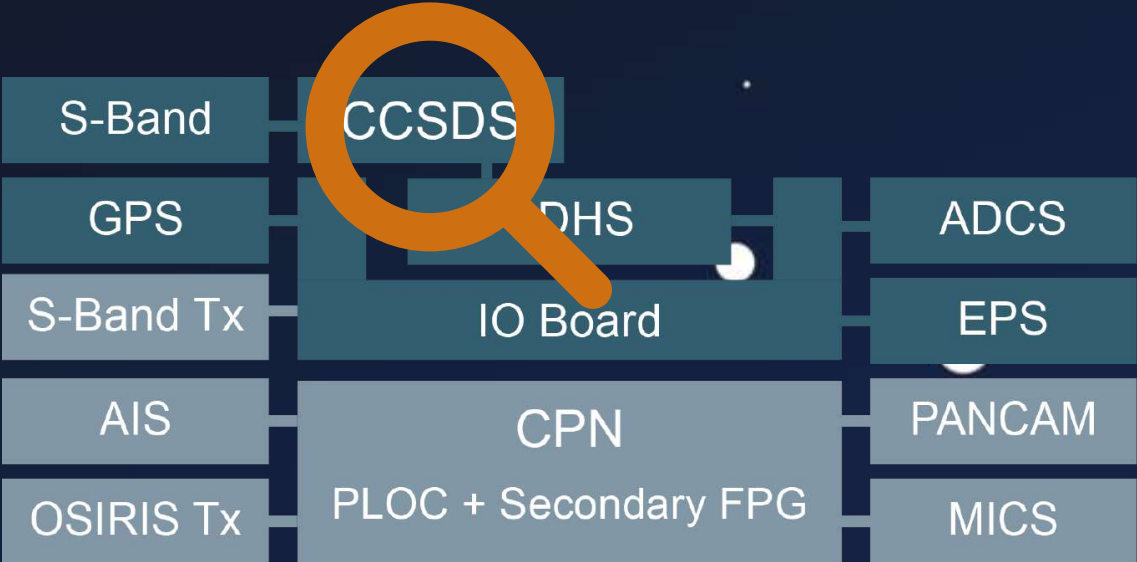
De-orbit mechanism, AIS, Camera, etc...

Peripherals

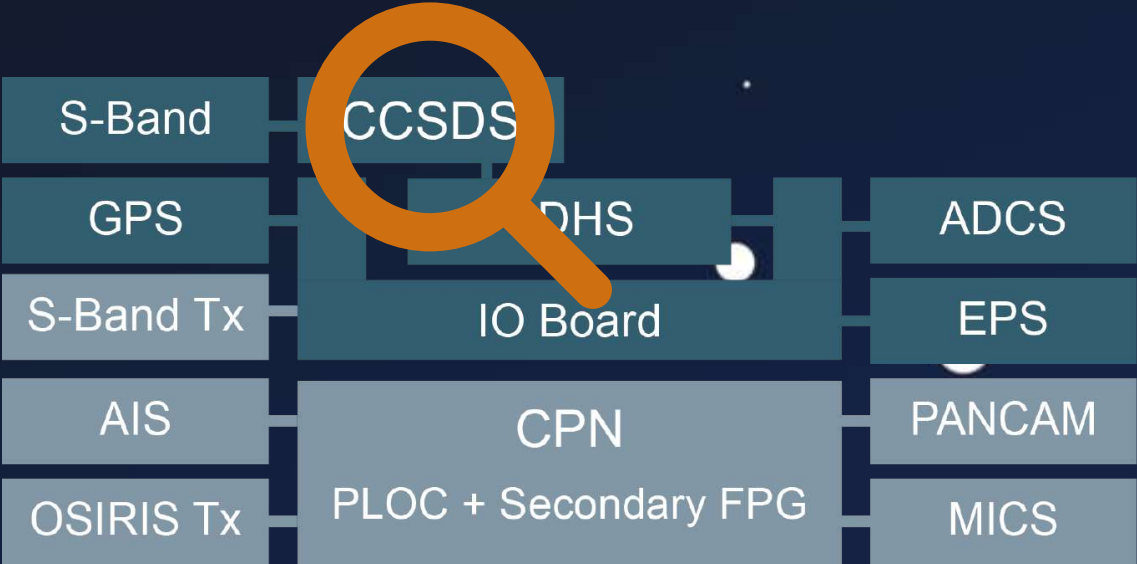
SPARC LEON 3 - OBC from Airbus S&D

Bus Platform

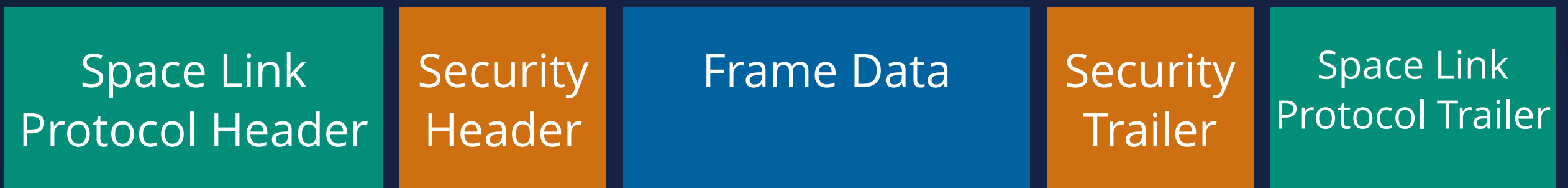
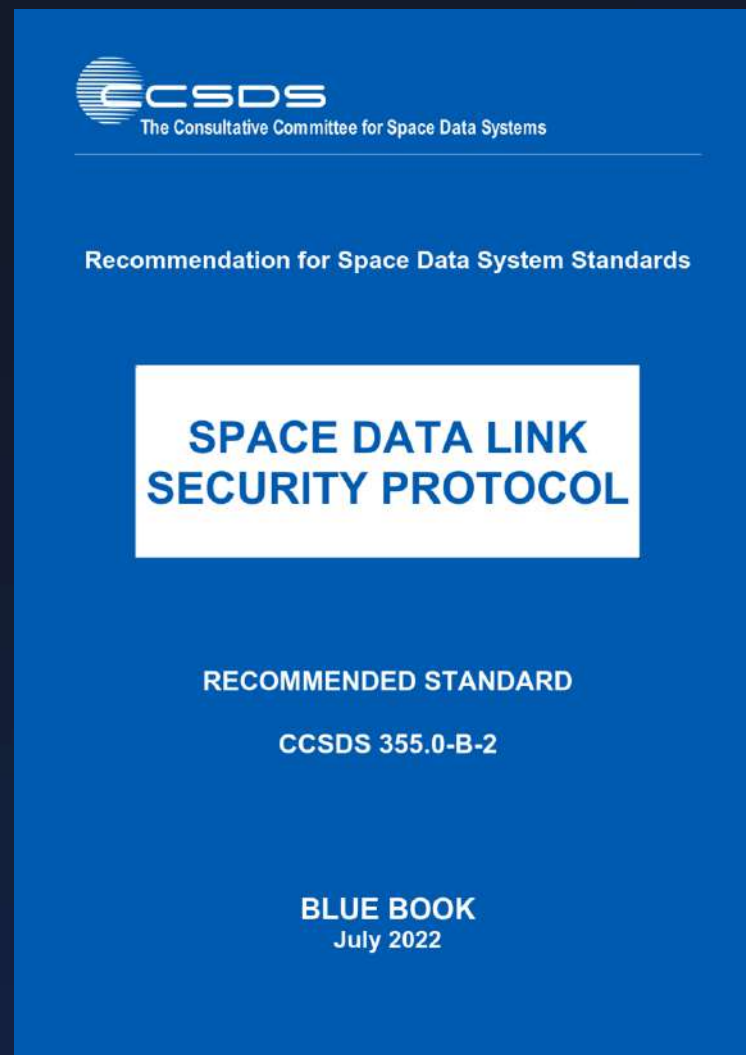
CCSDS



CCSDS - SDLP



CCSDS - SDLS



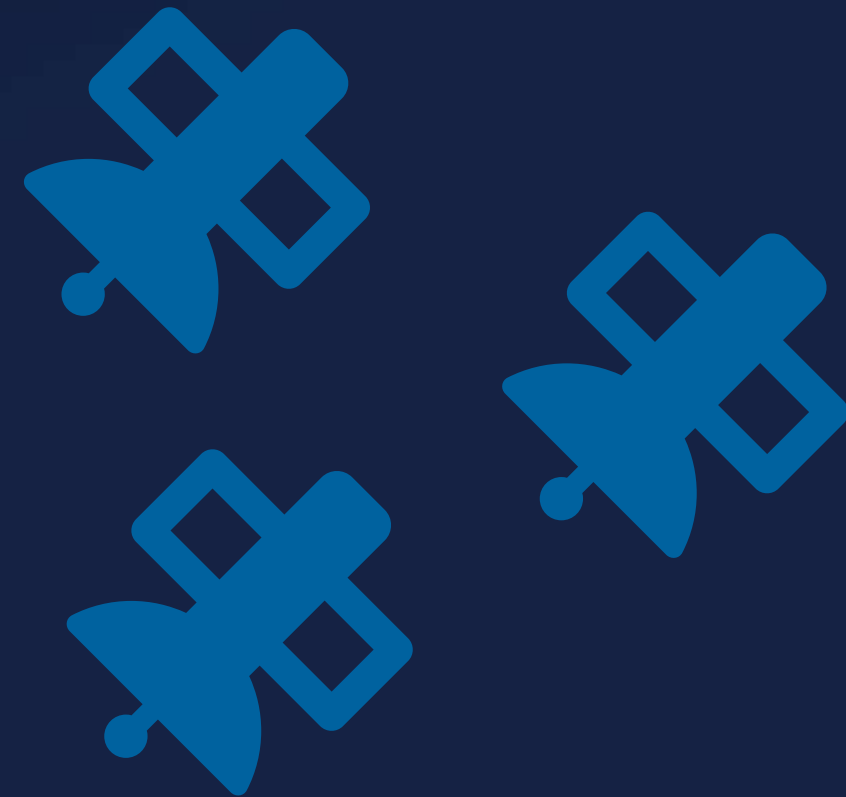
Bigger Picture



***"But it's different for
[...] satellites."***

***“ But it's different for
[...] satellites,
.... right?”***

Developer Survey



TC Protocols



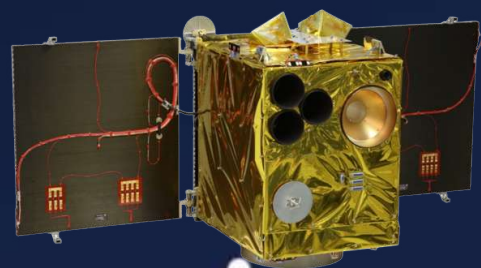
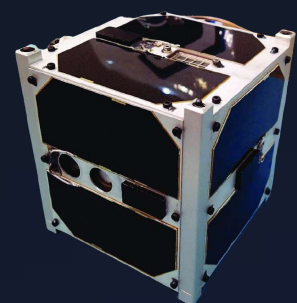
	Custom	Standard	Weight
	✓	✗	~ 1.3 kg
	~	✓	~ 5.4 kg
	✗	✓	~ 120 kg

Weight \approx Money

TC Protocols



Custom /
Standard



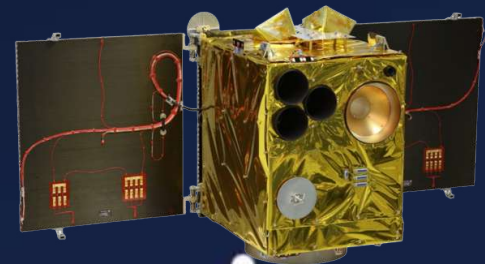
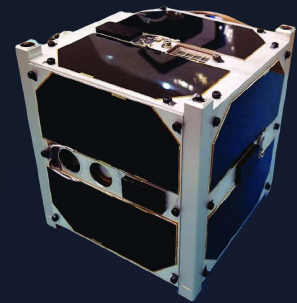
	1-50 kg	50-100 kg	> 100 kg
Standard	1	1	4
Custom	6	1	0
Abstains	3	0	1
Σ	10	2	5

Weight \approx Money

TC Protocols



Custom /
Standard



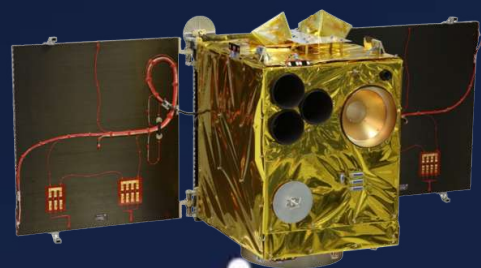
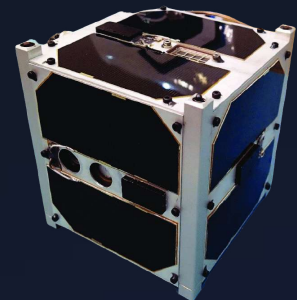
	1-50 kg	50-100 kg	> 100 kg
Standard	1	1	4
Custom	6	1	0
Abstains	3	0	1
Σ	10	2	5

Weight \approx Money

TC Protocols



Custom /
Standard



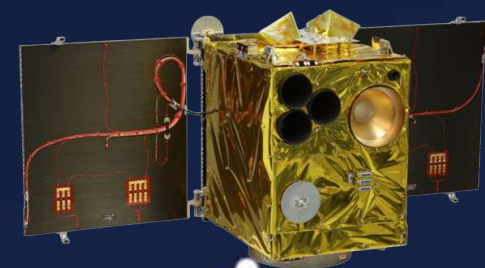
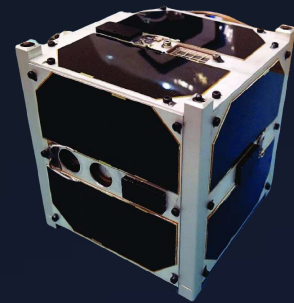
	1-50 kg	50-100 kg	> 100 kg
Standard	1	1	4
Custom	6	1	0
Abstains	3	0	1
Σ	10	2	5

Weight \approx Money

TC Protocols



Custom /
Standard



	1-50 kg	50-100 kg	> 100 kg
Standard	1	1	4
Custom	6	1	0
Abstains	3	0	1
Σ	10	2	5

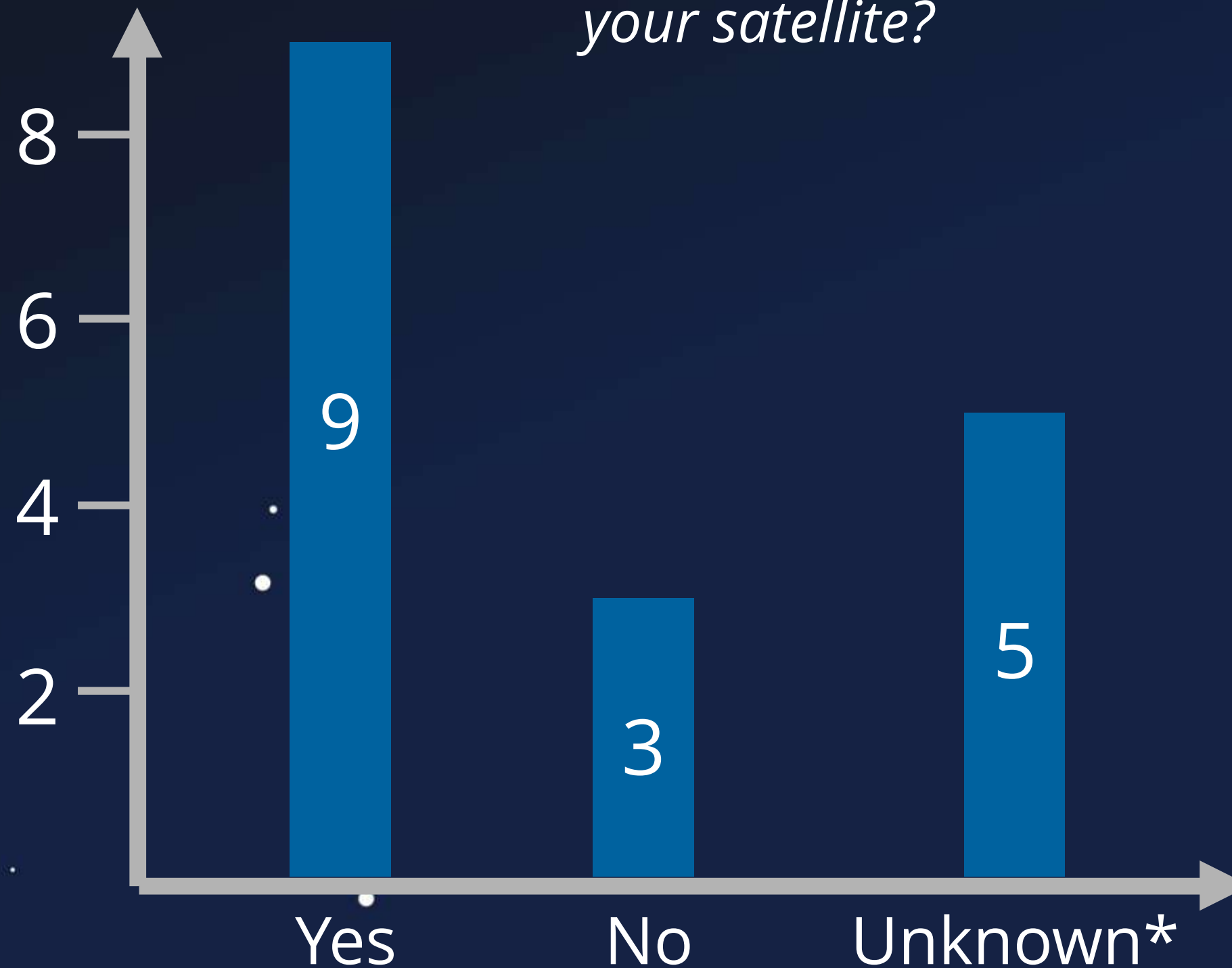
Weight \approx Money

=> Inaccessible Standard

TC Protection



Question: Are ***any measures deployed*** to prevent 3rd parties from controlling your satellite?

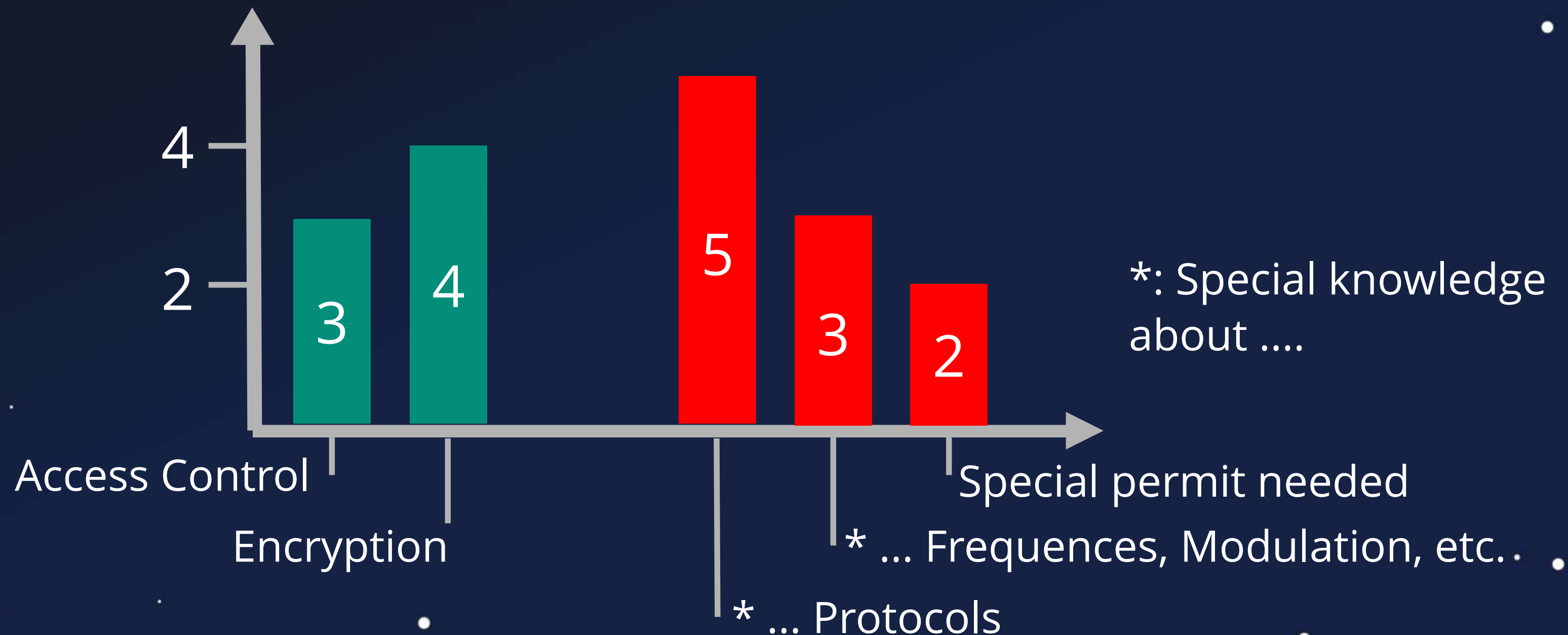


Unknown*:
Prefer not to say /
Don't know

TC Obscurity



Question: ***What measures** are deployed to prevent 3rd parties from controlling your satellite? (Multiple Answers)*



***" But it's different for
my satellite***

Impact



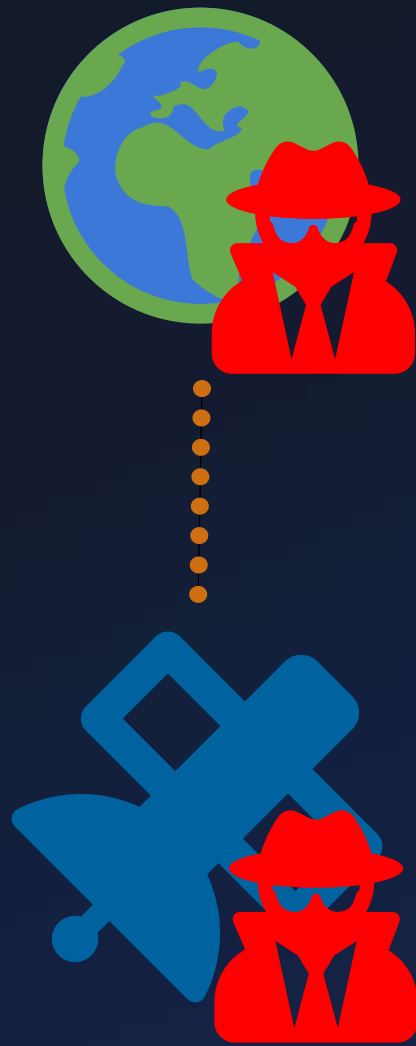
1. Hack a Satellite



???



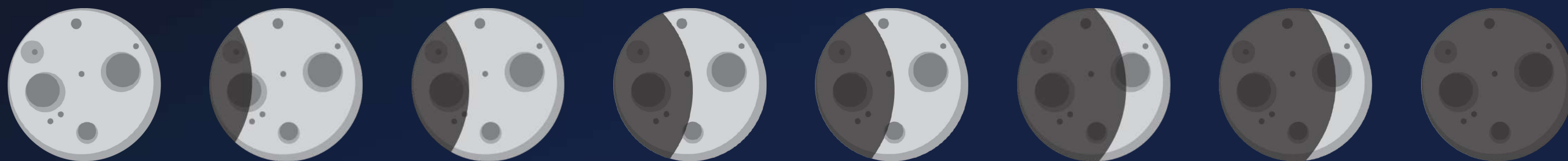
Scenarios



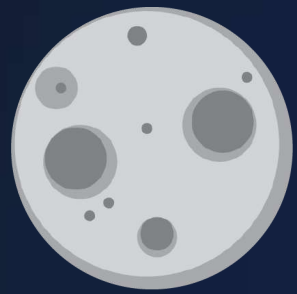
Orbital Access

- ① Attacking Inter-Sat Links
- ② Orbital Traffic Interception
- ③ Kessler Syndrome

Lesson Learnt



Lessons Learnt



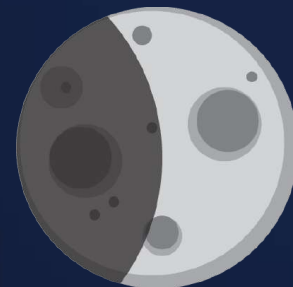
Firmware Attacks on Satellites are a thing



ViaSat Incident != Satellite Firmware Attack



Common Sat Protocols lack Security



Security by Obscurity

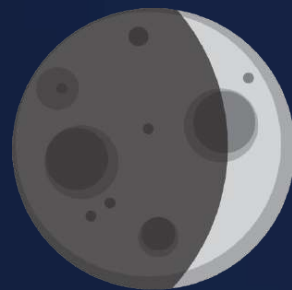
Lessons Learnt



Missing TC Protection



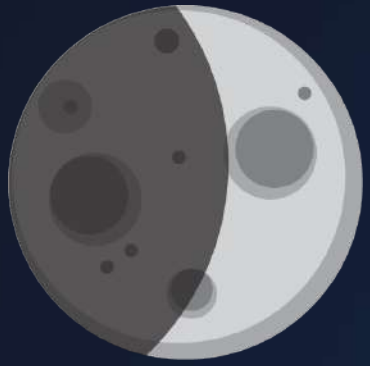
Missing State-of-the-Art Defenses



Attacker Access to Orbit as Staging Ground



Unknown Consequences



Thanks!



- Firmware Attacks on Satellite
- Satellite Exploitation Objectives
- Three Satellite Case Studies
- Satellite Developer Survey
- Impact beyond Vulnerable Satellites

 @jwillbold

 /jwillbold

Johannes Willbold - johannes.willbold@rub.de

[1] ESTCube-1 Image: <https://www.eoportal.org/satellite-missions/estcube-1>

[2] OPS-Sat Image: https://www.esa.int/ESA_Multimedia/Videos/2019/12/OPS-SAT_ESA_s_flying_lab_open_to_all

[3] Flying Laptop Image: <https://www.irs.uni-stuttgart.de/en/research/satellitetechnology-and-instruments/smallsatelliteprogram/flying-laptop/>