Space Invaders

An Experimental Security Analysis of
LEO Satellites

Johannes Willbold


@jwillbold


/jwillbold


jwillbold

$whoami

- Satellite & Space Systems Security
- PhD Student
  - Ruhr University Bochum, DE
- Co-Founder of SpaceSec
- Visiting Researcher
  - Cyber-Defence Campus, CH
- 2022 CySat Speaker

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

***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 This lack likely stems from by now outdated assumptions on achieving security by obscurity, effectively preventing meaningful research on satellite firmware.**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].

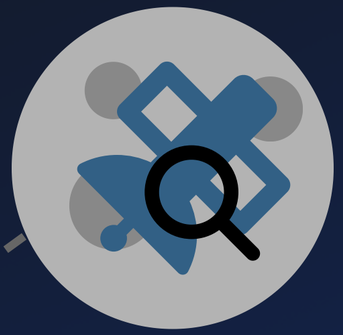
44th IEEE Symposium on Security and Privacy (S&P)

Our Journey ...

Firmware Attacks

Our Journey ...

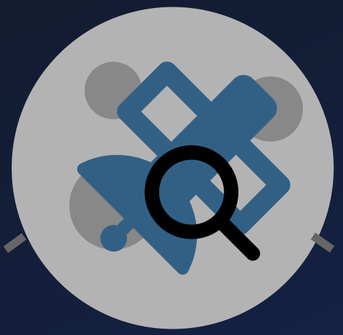
Security Analysis



Firmware Attacks

Our Journey ...

Security Analysis

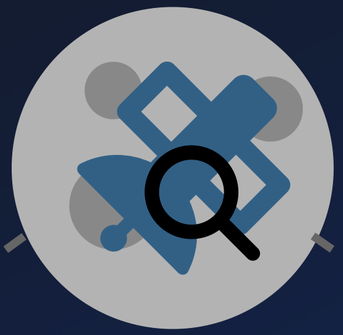


Firmware Attacks

Live Demo

Our Journey ...

Security Analysis

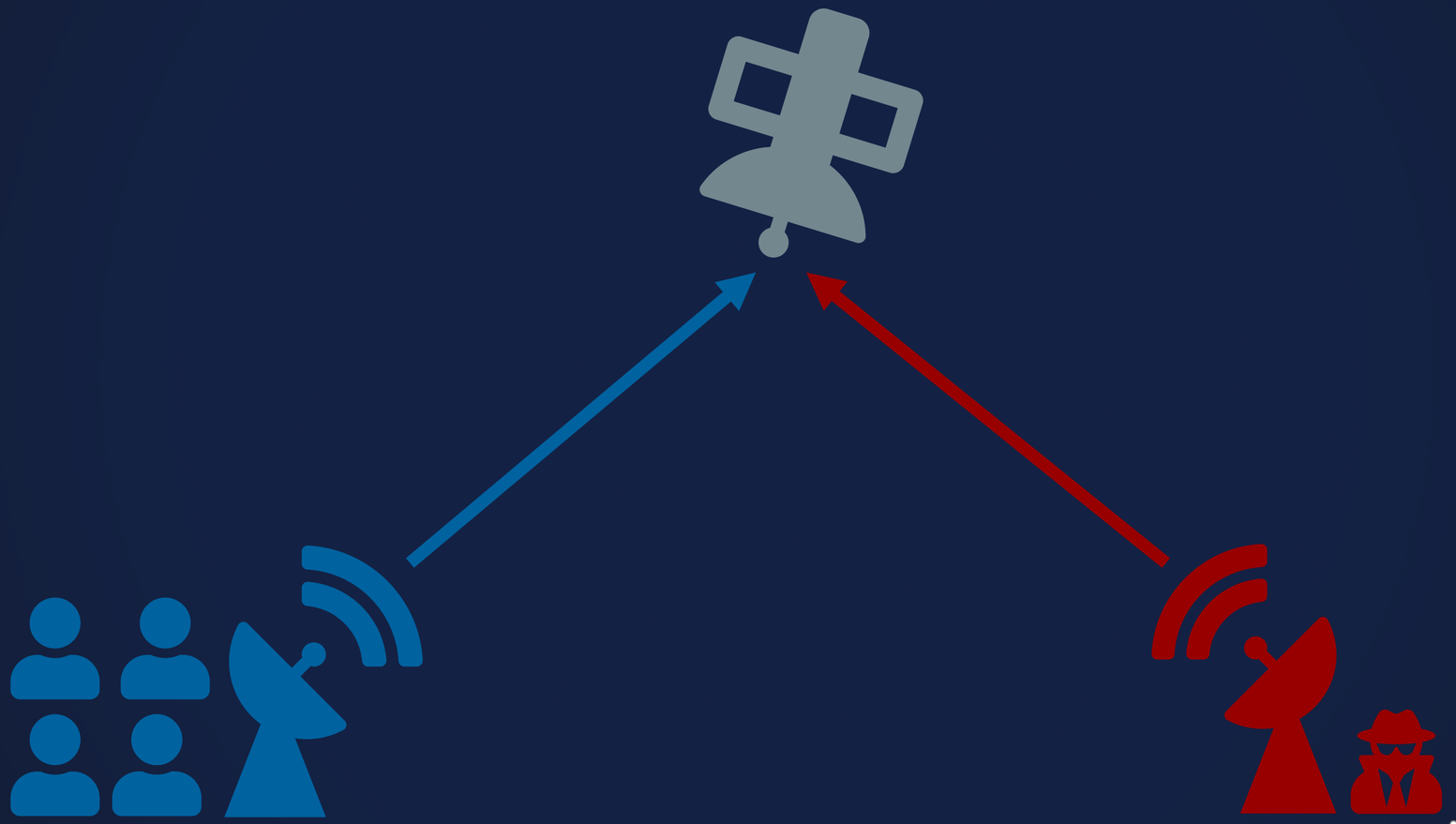


Lessons Learnt

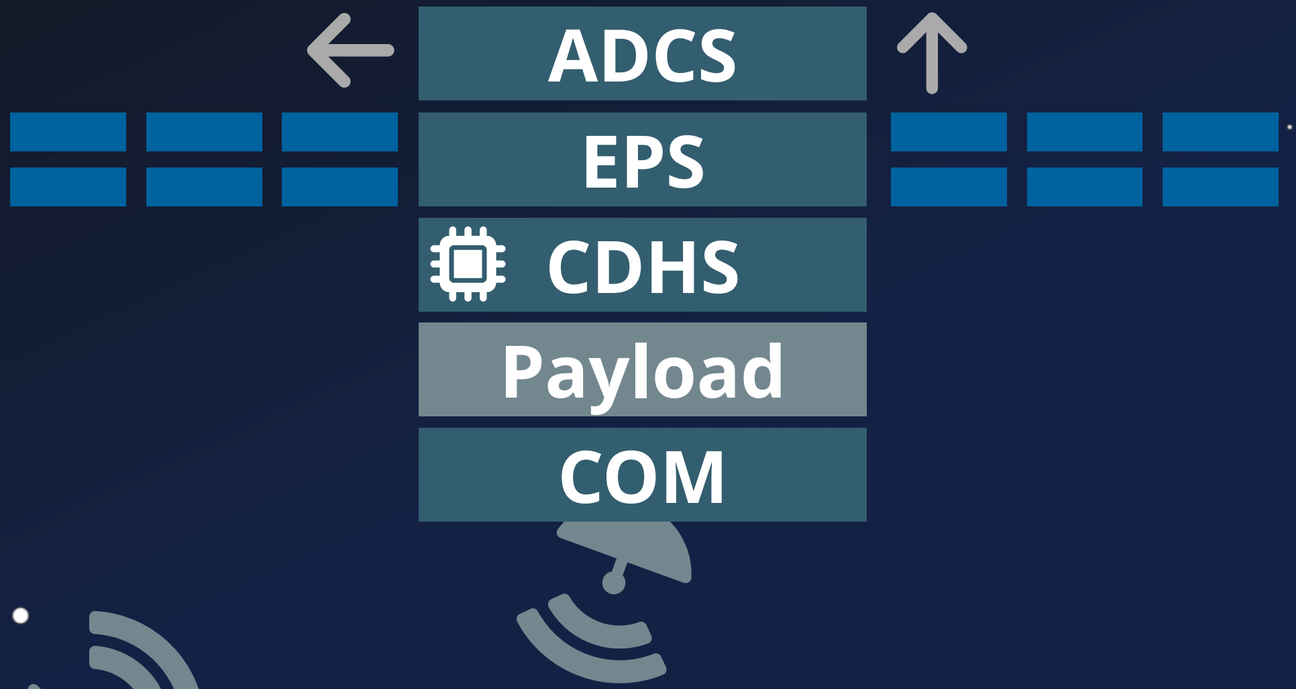
Firmware Attacks

Live Demo

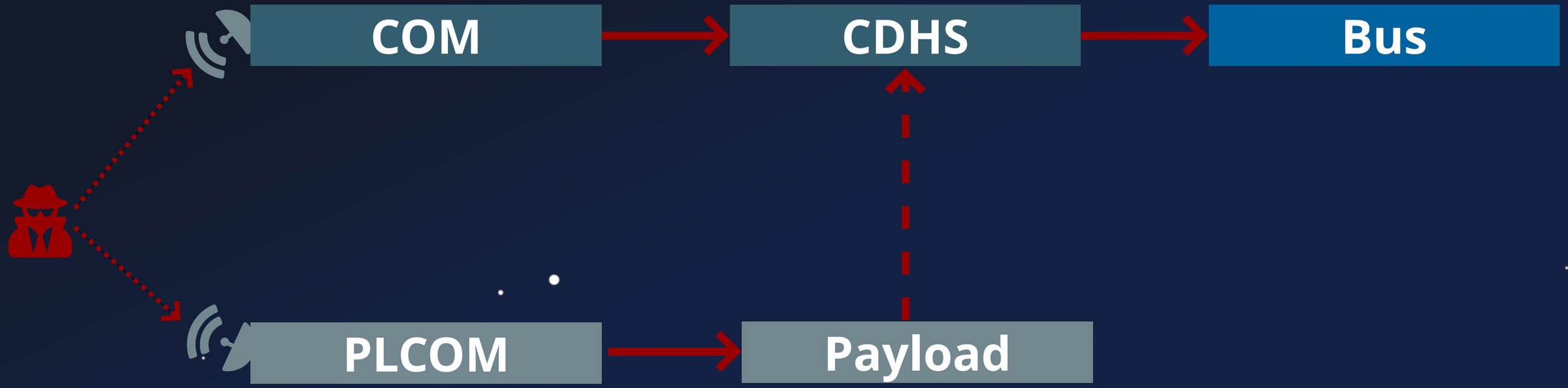
Firmware Attacks



Components



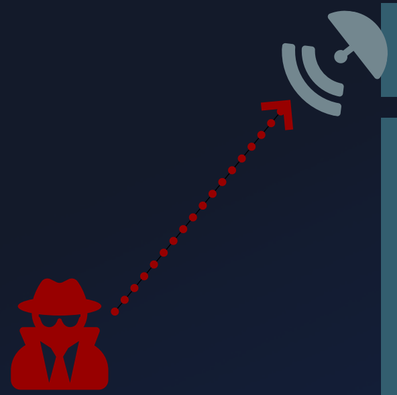
Components



Components

- Hack CySat 2022
- *How I hacked an ESA's experimental satellite*
  - Maurice-Michel Didelot
- Tomorrow
  - Matteo Calabrese
- Hack CySat 2023

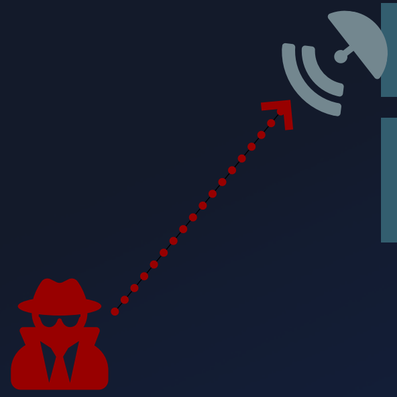
Components



COM → **CDHS** → **Bus**

- Bypass COM Protection
  - Missing AC
  - Insecure Protocol
  - Outdated Crypto
  - Timing Side Channels
  - Leaked Keys
  - Timed Backdoor
  - ...

Components



COM

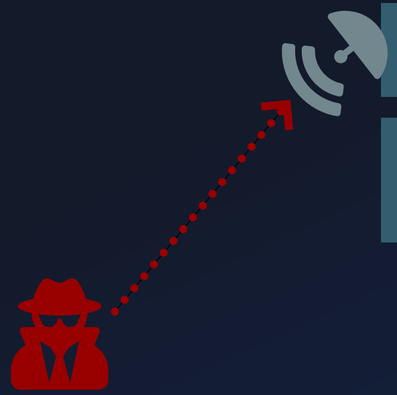
- Bypass COM Protection
  - [...]

CDHS

- Deploy Attacker Payload
  - Firmware Update
    - Signed Image
    - Slow Upload
    - Comples System

Bus

Components



COM

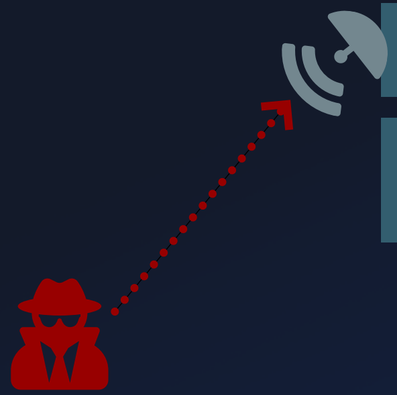
- Bypass COM Protection
  - [...]

CDHS

- Deploy Attacker Payload
  - ~~Firmware Update~~
  - Dangerous TC
  - Vulnerbale TC

Bus

Components



COM

- Bypass COM Protection
  - [...]

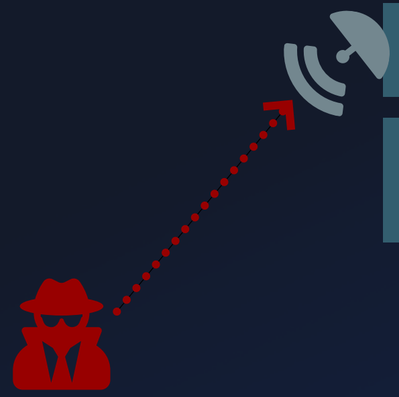
CDHS

- Deploy Attacker Payload
  - ~~Firmware Update~~
  - Dangerous TC
  - Vulnerbale TC

Bus

- Hijack Bus Control Flow
- Full Bus Privileges

Components



COM

- Bypass COM Protection
  - [...]

CDHS

- Deploy Attacker Payload
  - ~~Firmware Update~~
  - Dangerous TC
  - Vulnerbale TC

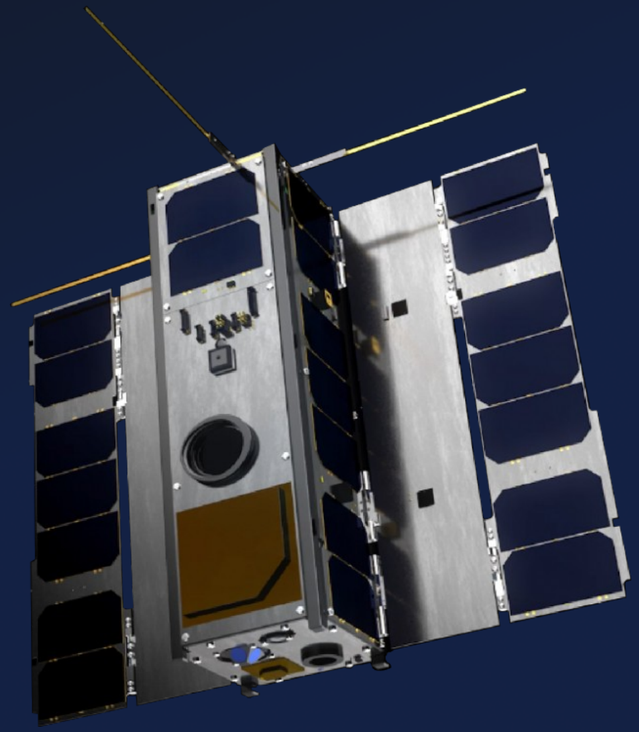
Bus

- Hijack Bus Control Flow
- Full Bus Privileges

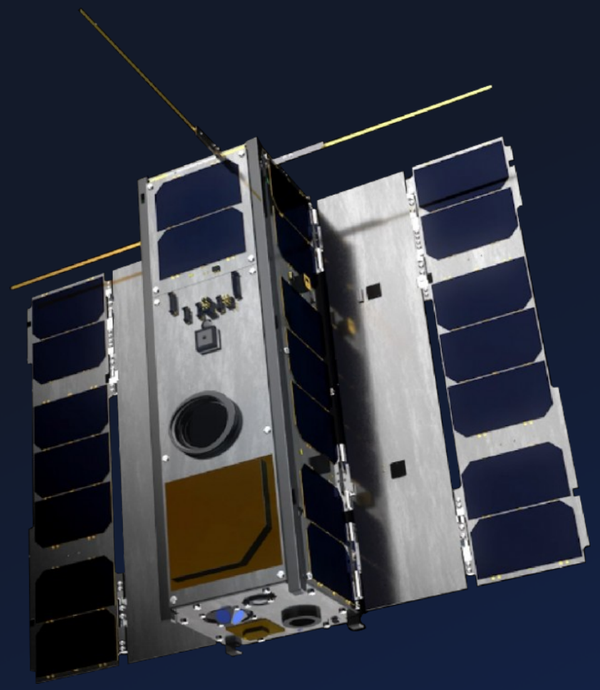
Objectives

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

Security Analysis



Awesome Sat



OPS-Sat

Operated by ESA

Open for Research

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

Peripherals

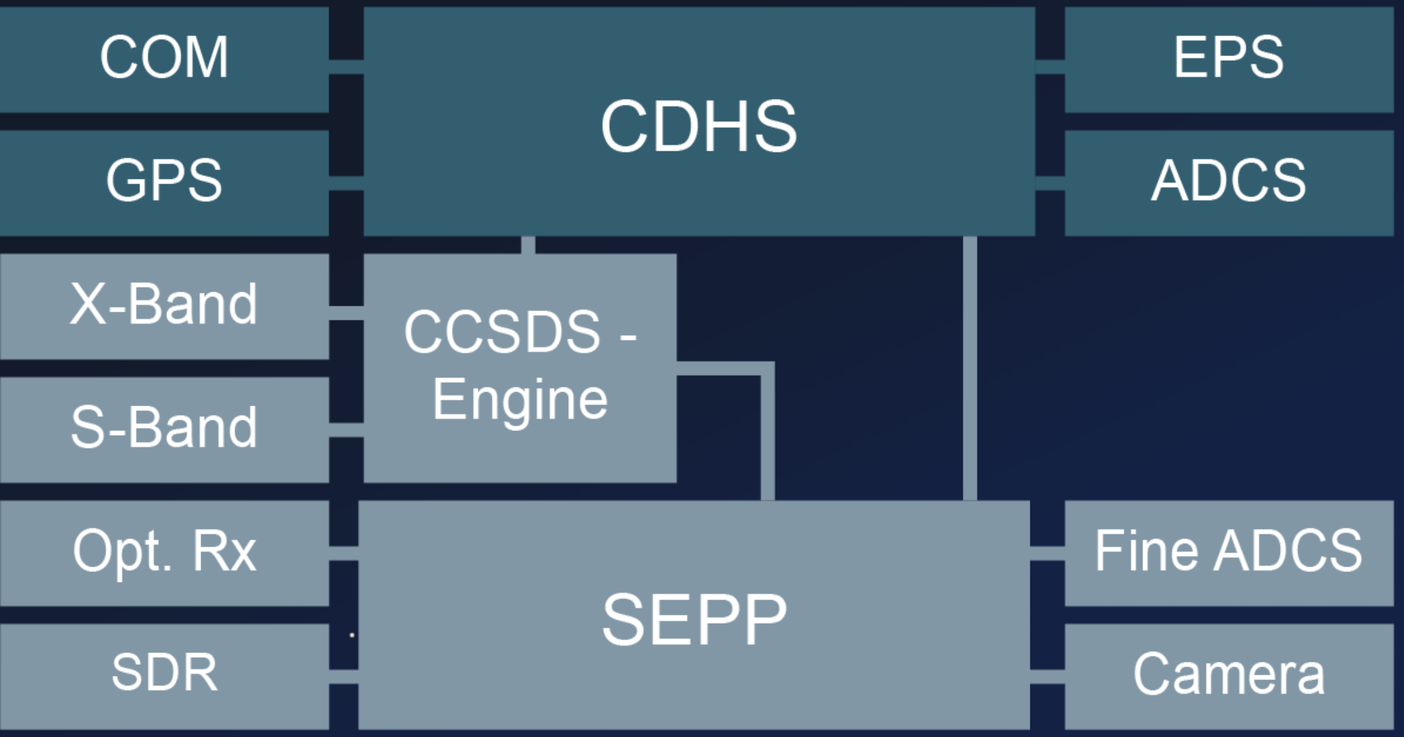
ARM-Based Linux + FPGA

Payload Plattform

December 2019

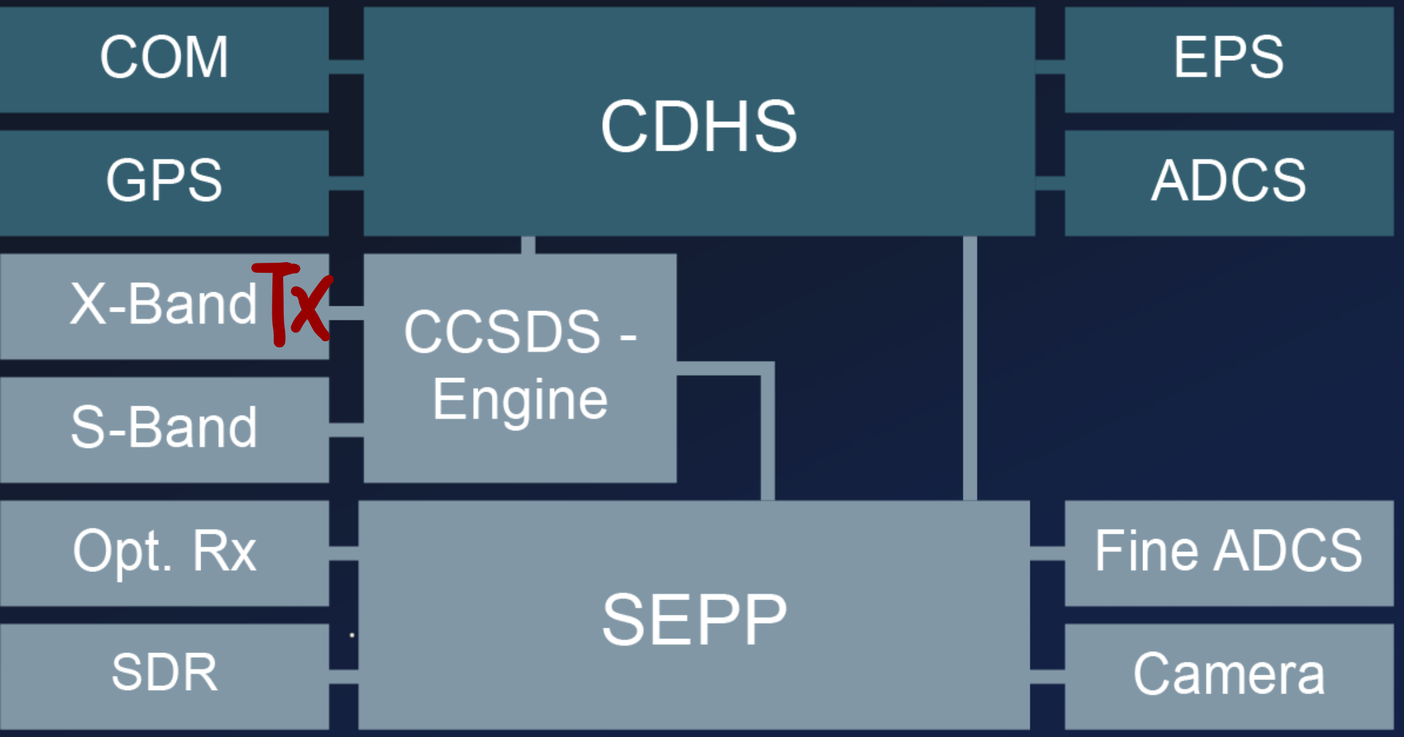
Launched

System Chart



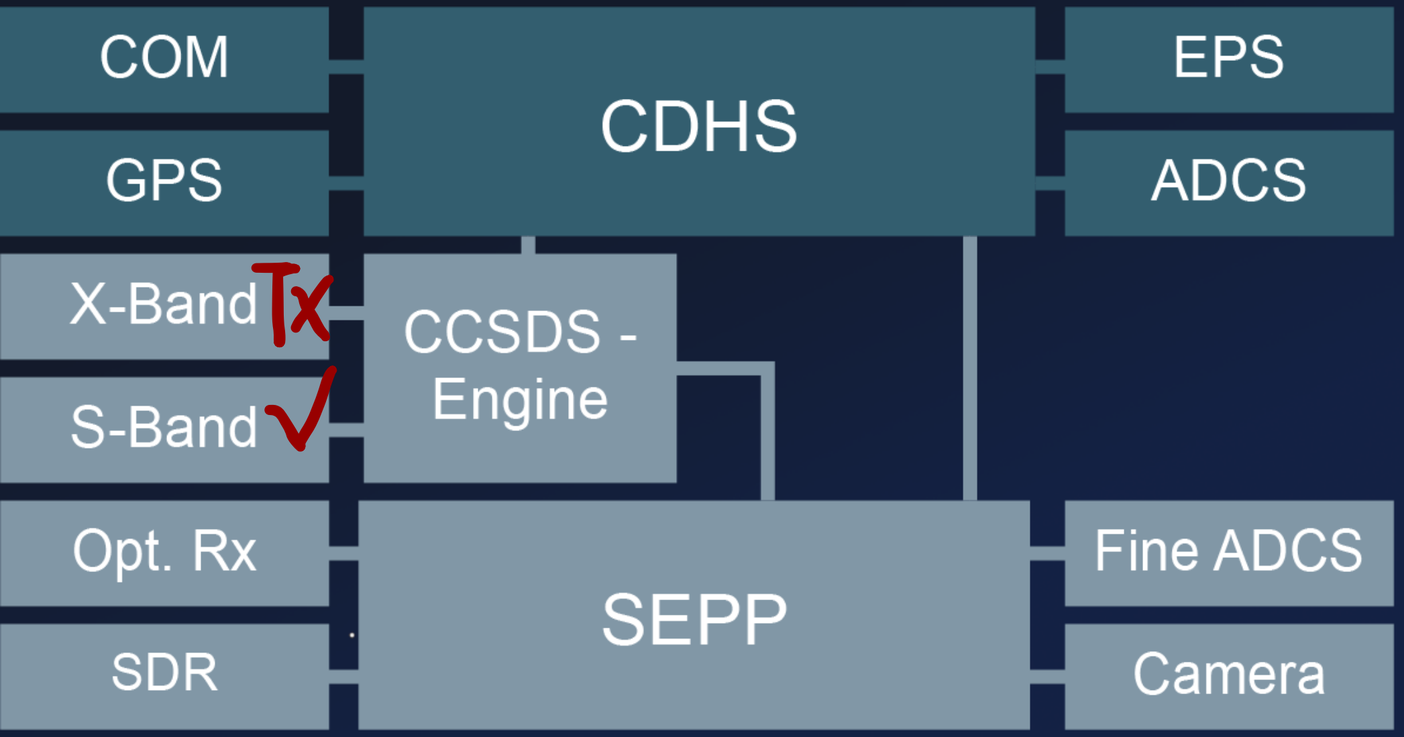
1. Bypass COM Protection

System Chart



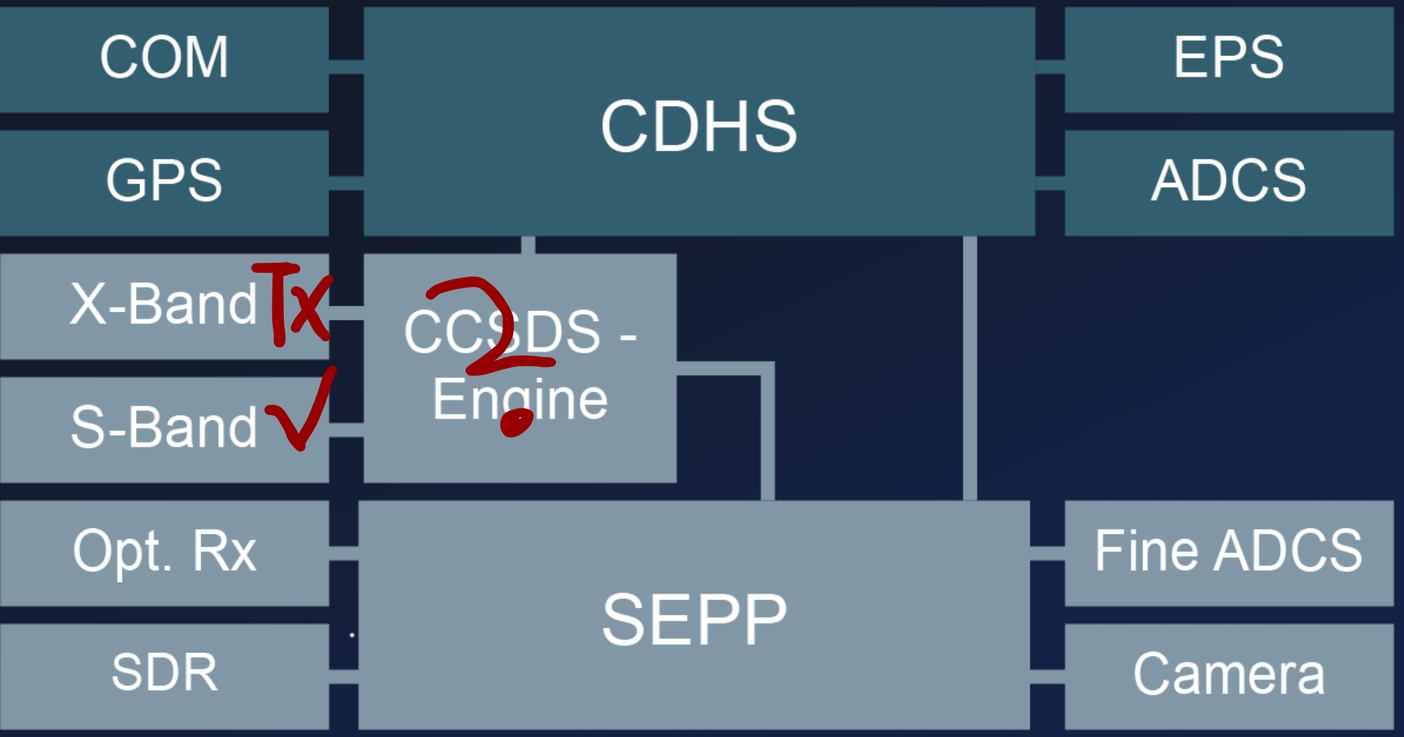
1. Bypass COM Protection

System Chart



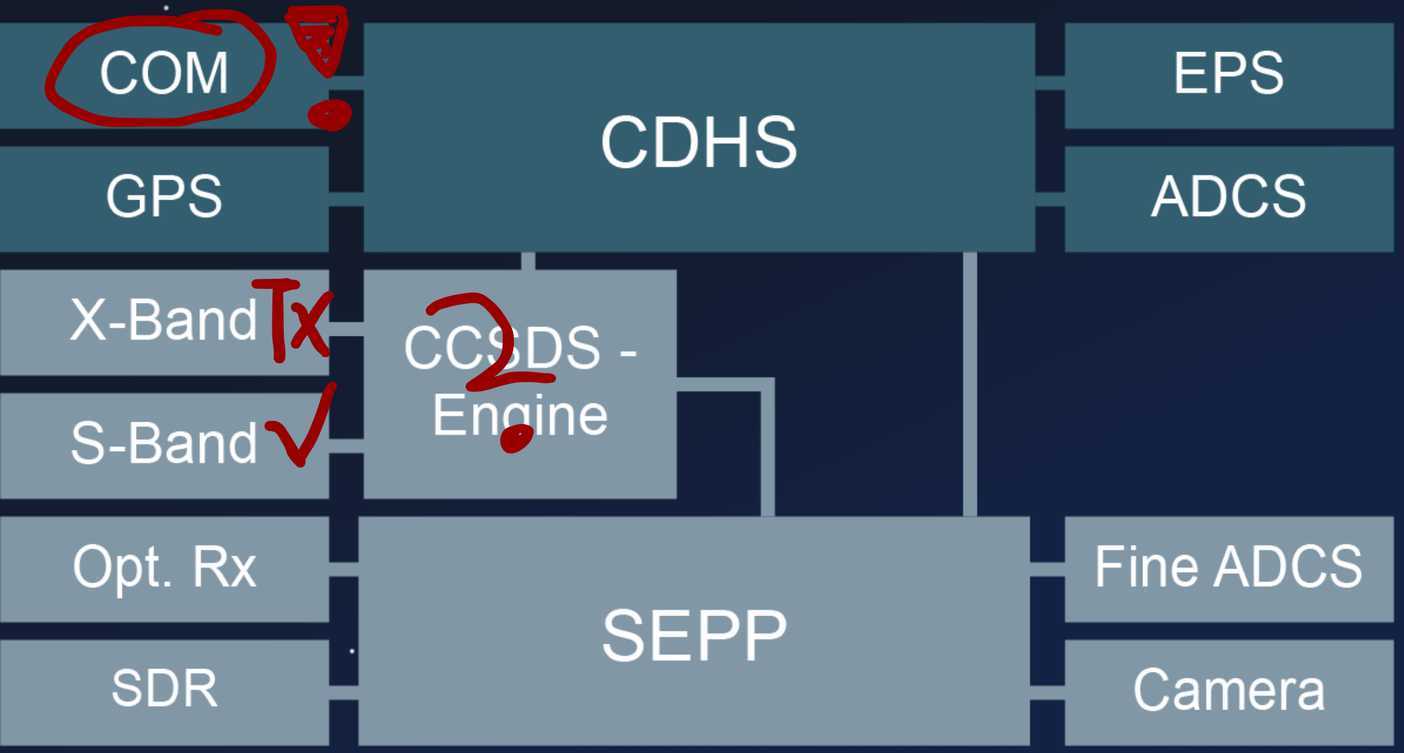
1. Bypass COM Protection

System Chart



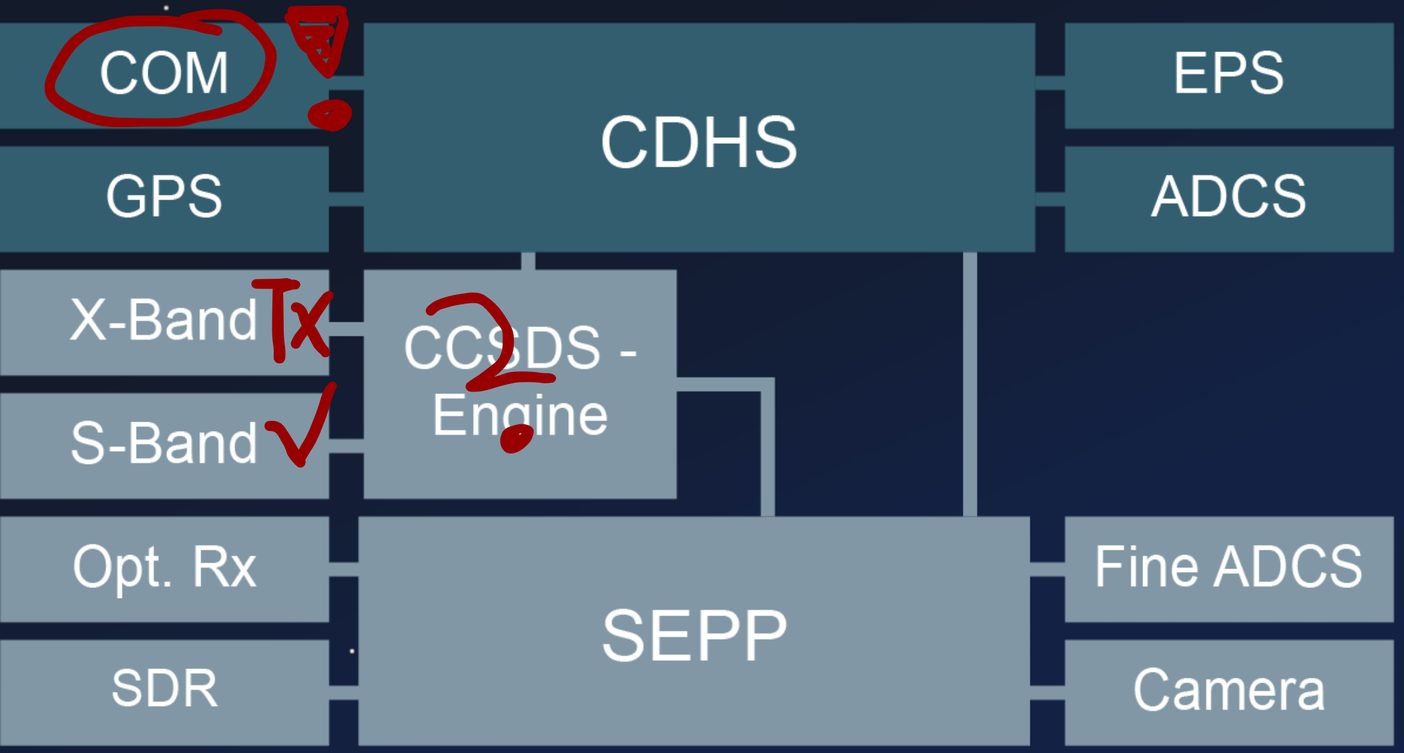
1. Bypass COM Protection

System Chart



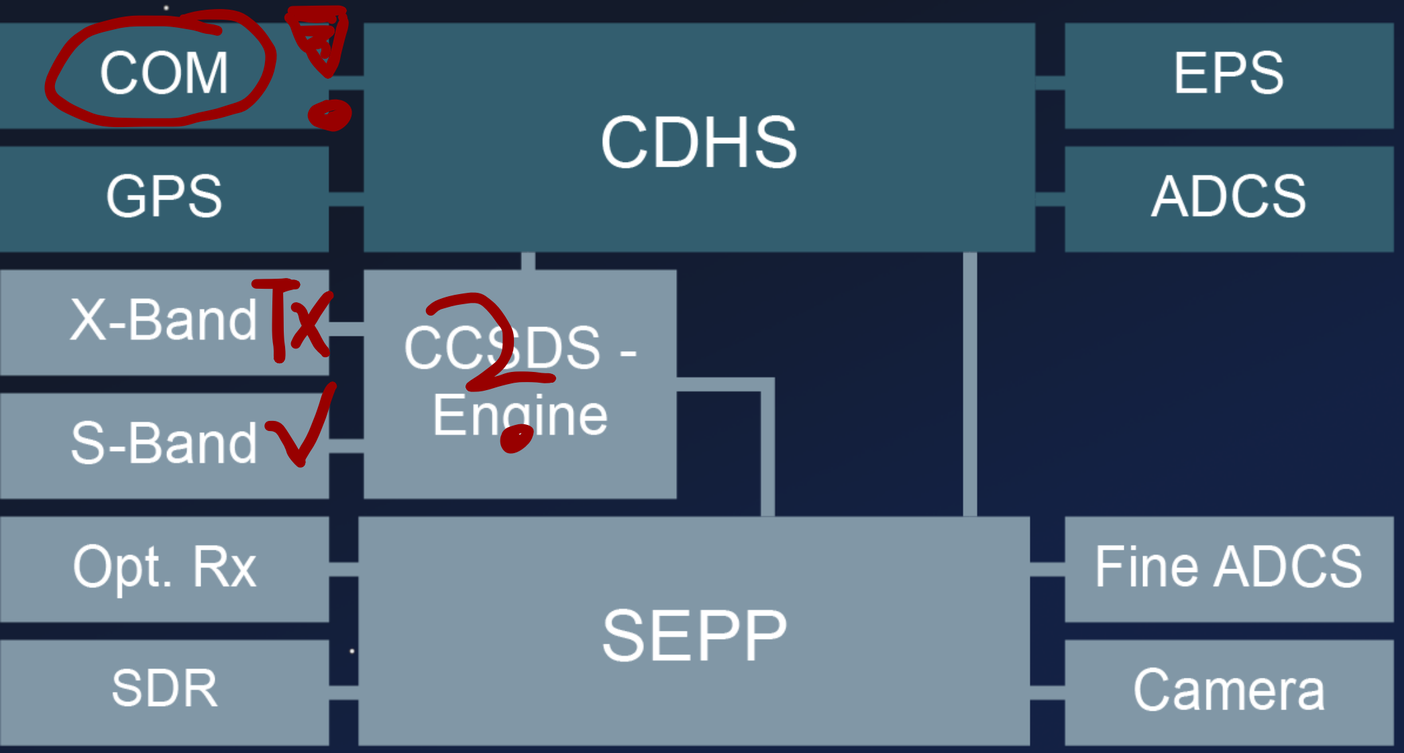
1. Bypass COM Protection

System Chart



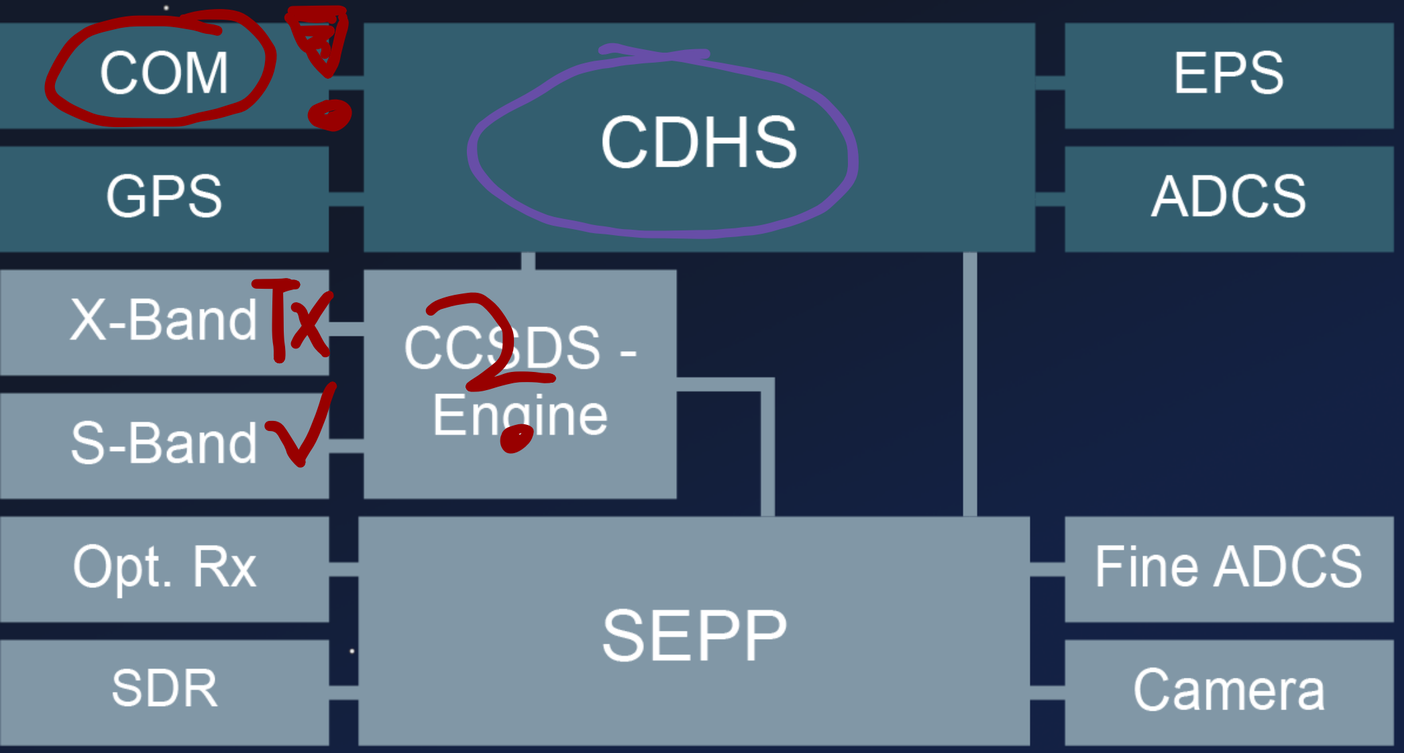
1. Bypass COM Protection

System Chart



1. Bypass COM Protection
2. Dangerous / Vulnerable TC

System Chart



1. Bypass COM Protection
2. Dangerous / Vulnerable TC

System Chart

```c
int sch_handler_set_raw_memory(scheduler_cmd_t* pCmd) {
  raw_mem_access_cmd_t* pAddr = pCmd->pCmdArgs;
  char* pWriteData;

  if (pAddr) {
    if (g_sch_exec_mode != 1 ) {
      /* exception and return */
    }
    char* pWriteData = &pAddr->start_of_data_buf;
    if (pAddr->filesystem_target) {
      // [...]
    } else {
      memcpy(pAddr->targetAddr,
             &pAddr->start_of_data_buf,
             pAddr->writeLength);
    }
  }
  // ...
}
```

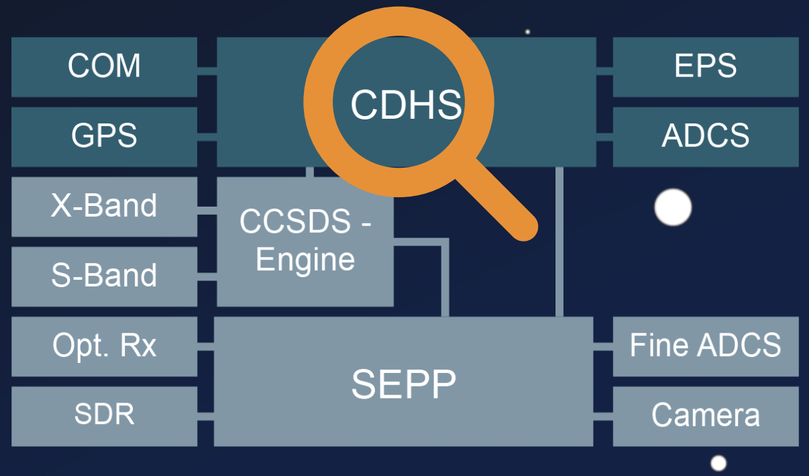
- *memcpy* as TC
  - Config Changes
  - Quick/Hot Patching
  - Debugging

System Chart

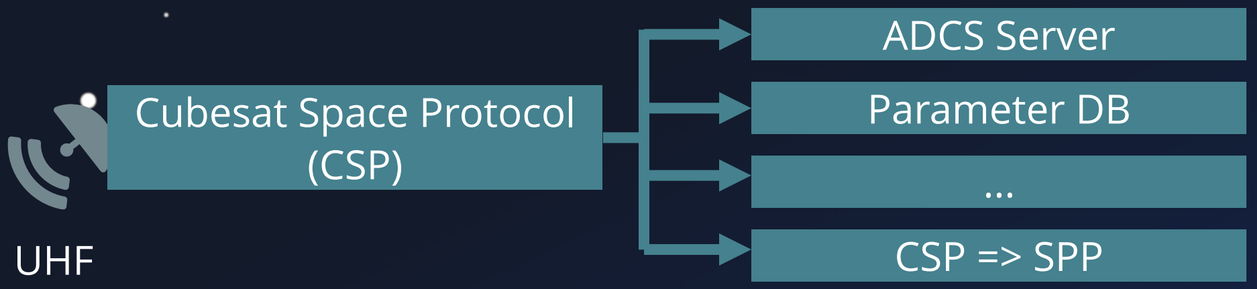
```c
int sch_handler_set_raw_memory(scheduler_cmd_t* pCmd) {
  raw_mem_access_cmd_t* pAddr = pCmd->pCmdArgs;
  char* pWriteData;

  if (pAddr) {
    if (g_sch_exec_mode != 1 ) {
      /* exception and return */
    }
    char* pWriteData = &pAddr->start_of_data_buf;
    if (pAddr->filesystem_target) {
      // [...]
    } else {
      memcpy(pAddr->targetAddr,
             &pAddr->start_of_data_buf,
             pAddr->writeLength);
    }
  }
  // ...
}
```

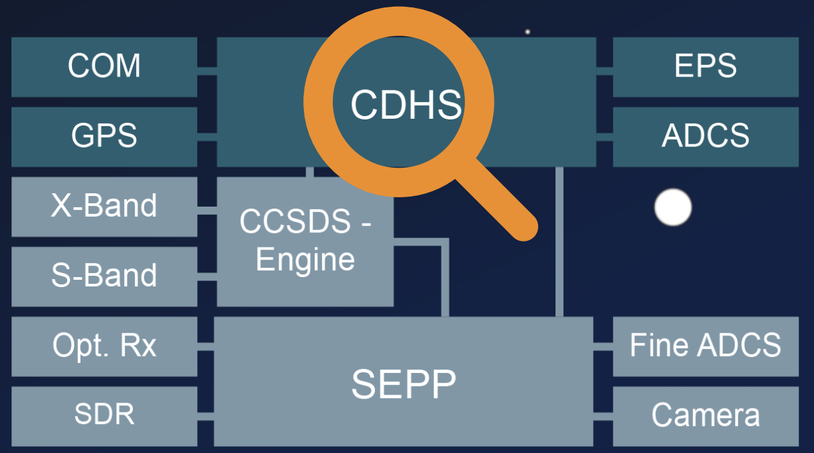
- *memcpy* as TC
  - Config Changes
  - Quick/Hot Patching
  - Debugging

System Chart

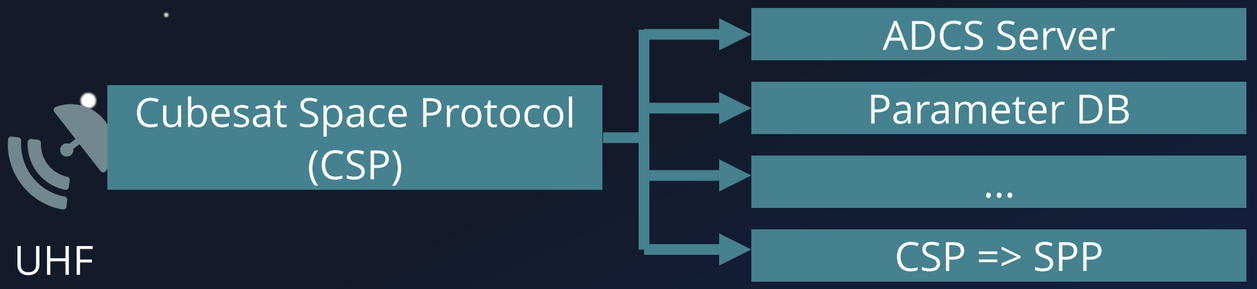


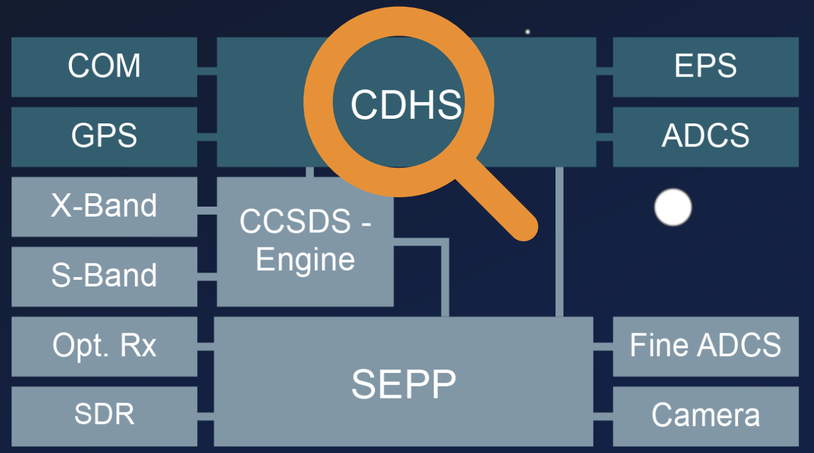
System Chart

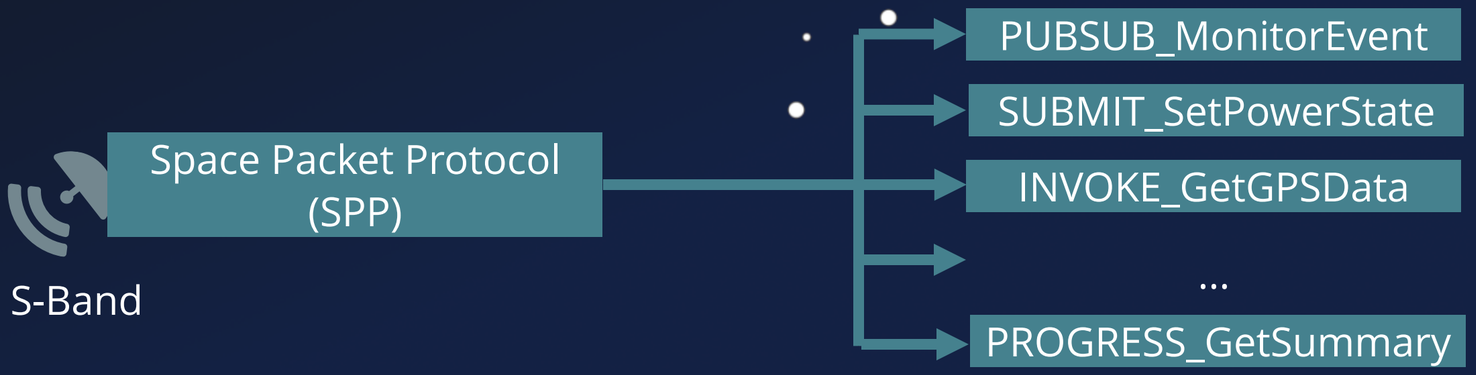




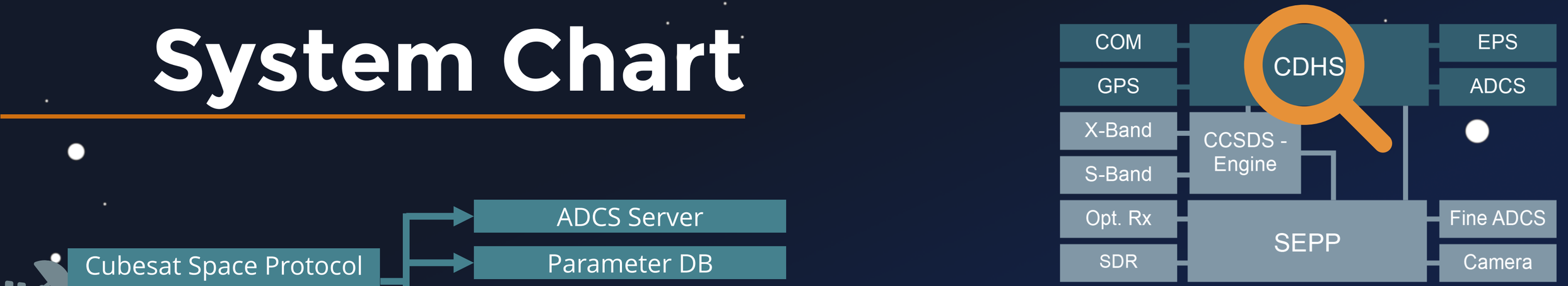
System Chart

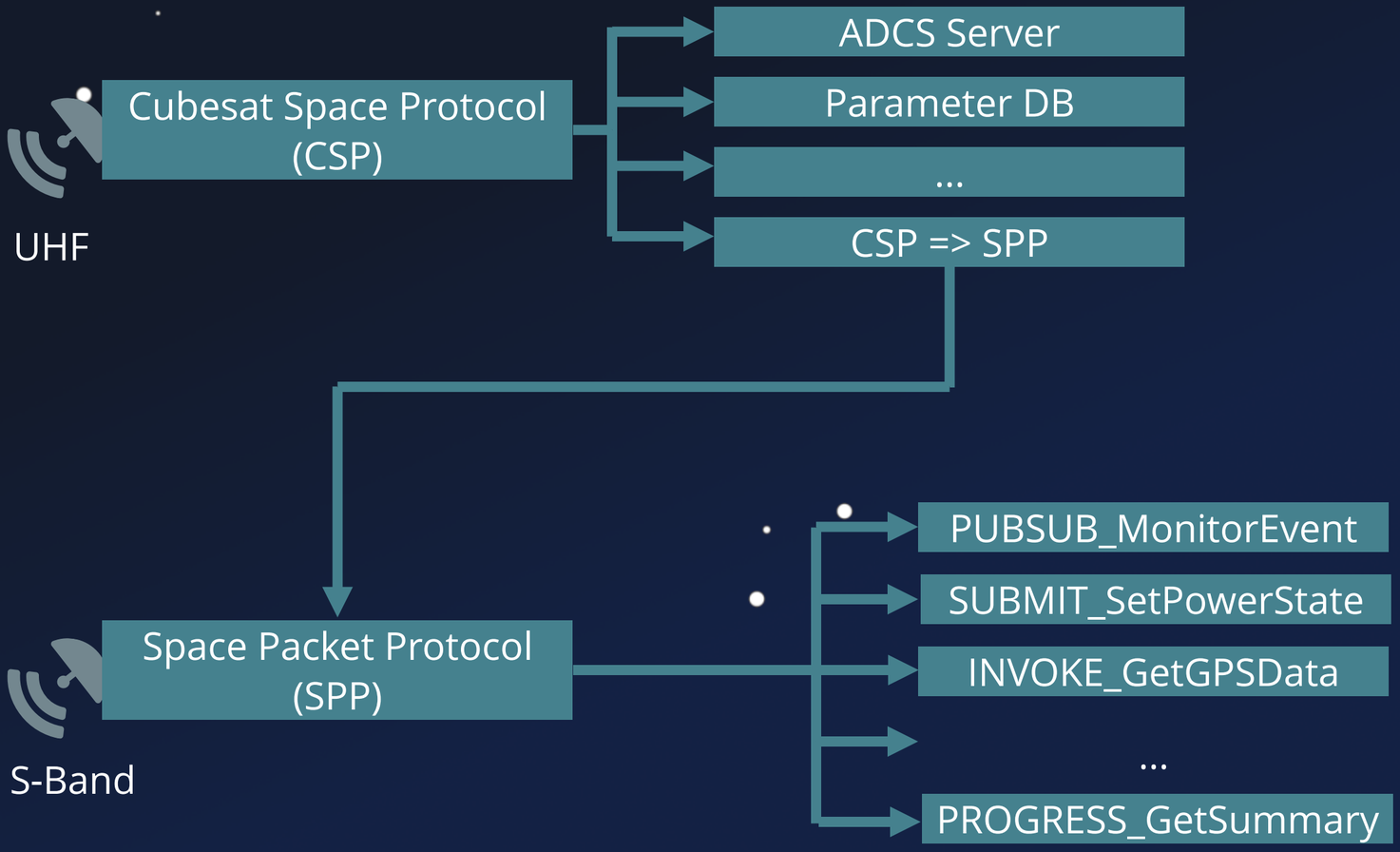




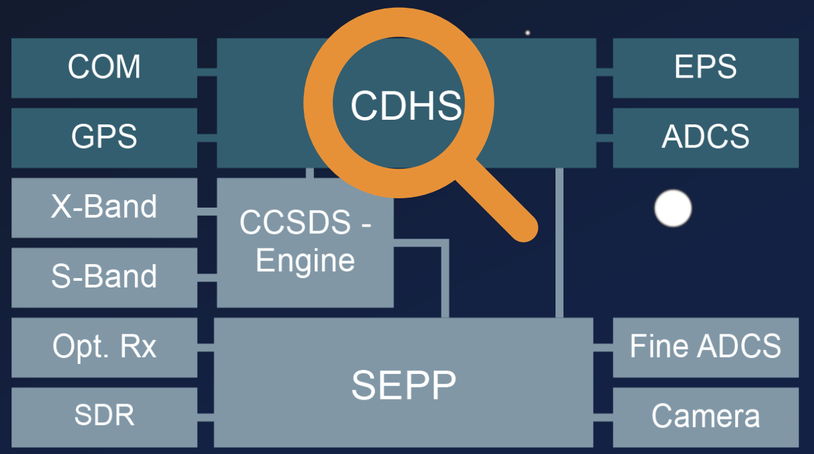


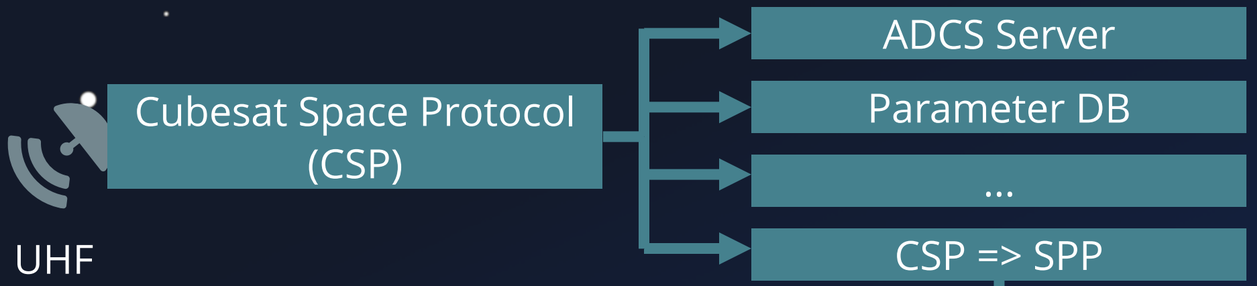
System Chart

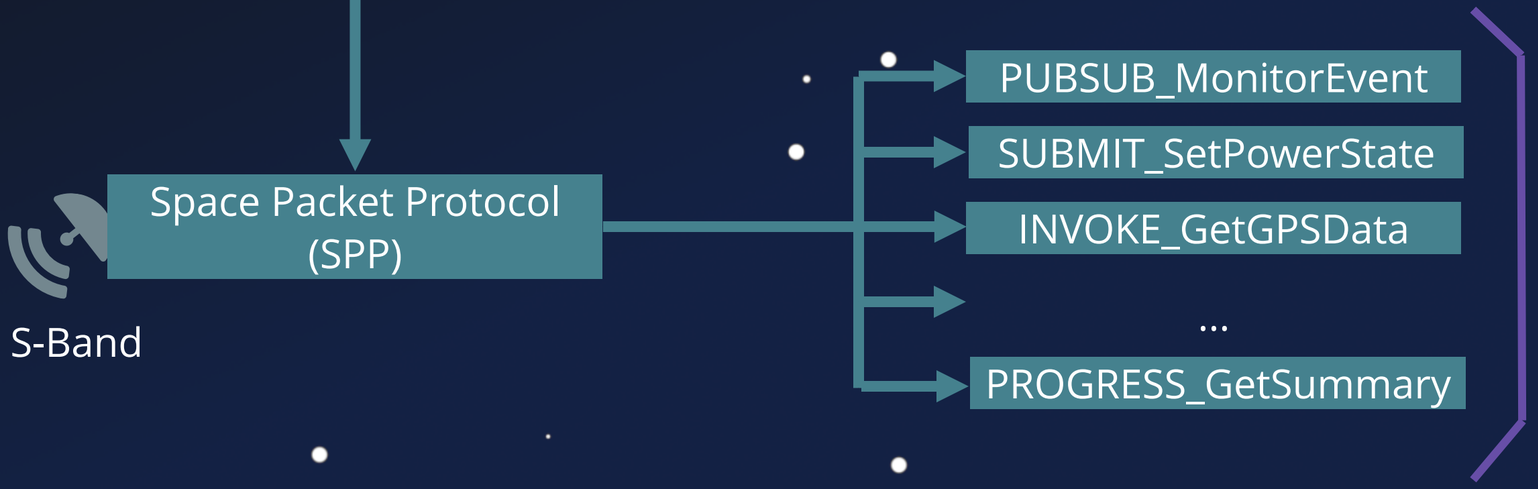




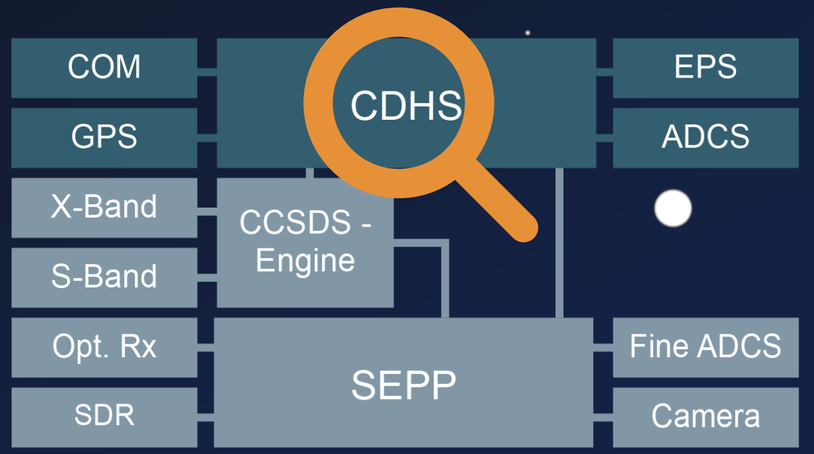
System Chart

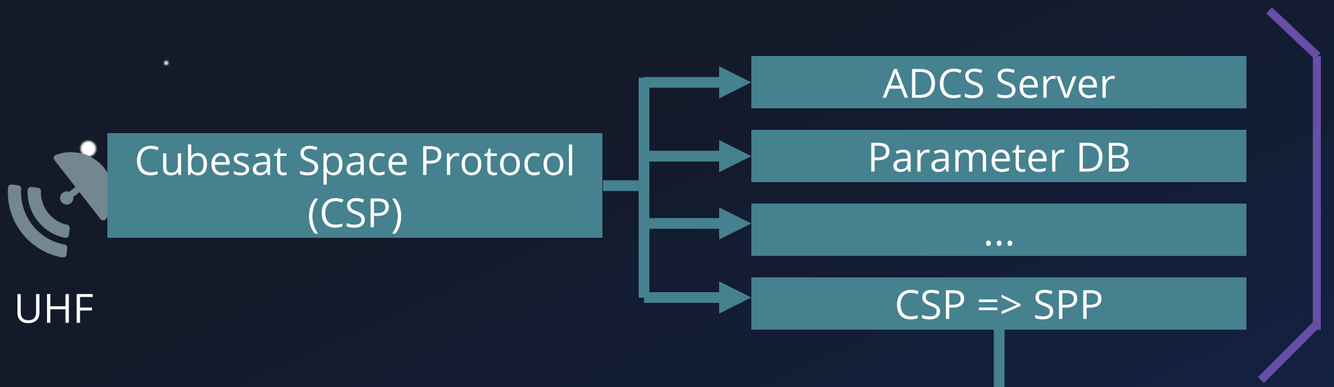


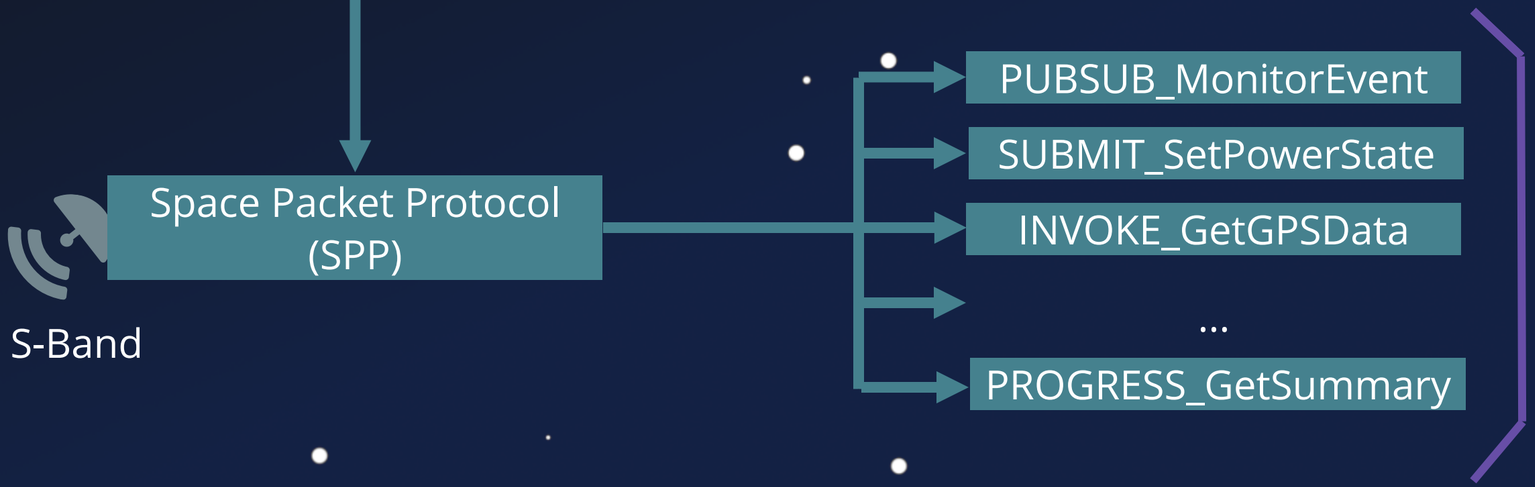




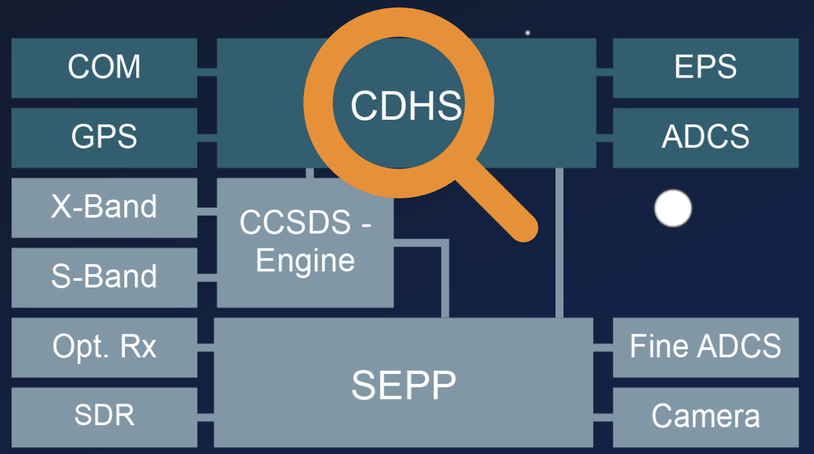
System Chart







System Chart



UHF

Cubesat Space Protocol (CSP)

- ADCS Server
- Parameter DB
- ...
- CSP => SPP

Custom Byte Parsing

S-Band

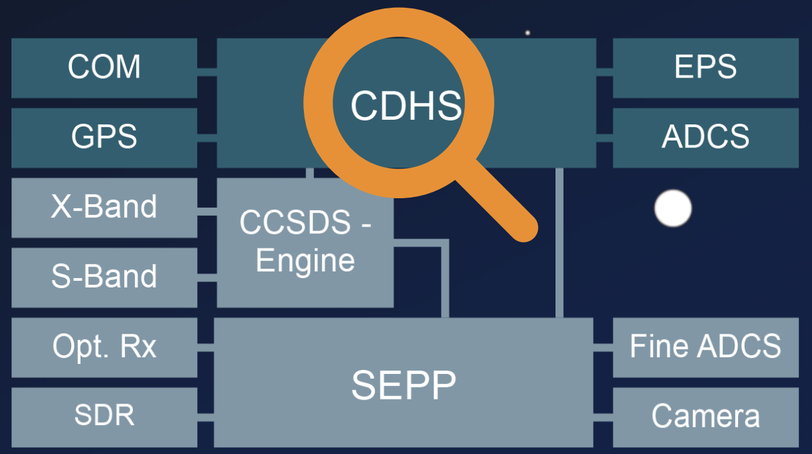
Space Packet Protocol (SPP)

- PUBSUB_MonitorEvent
- SUBMIT_SetPowerState
- INVOKE_GetGPSData
- ...
- PROGRESS_GetSummary

Message Abstraction Layer (MAL)

System Chart

Cubesat Space Protocol (CSP) → ADCS Server

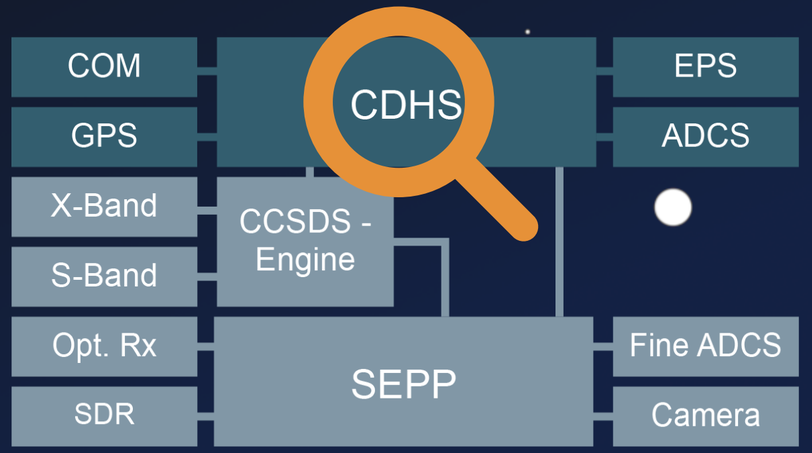



```c
void task_adcs_servr() {
  char log_file_name [32];

  csp_listen(socket, 10);
  csp_bind(socket, port);

  do {
    do {
      conn = csp_accept(socket, 0xff);
    } while (do_wait_for_conn);

    packet = csp_read(conn, 10);
    if (packet) {
      packet_data = packet->data;
      switch(*packet_data) {
        // [...]
        case SET_LOGFILE: {
          packet_data = packet->data + 0xf;
          log_file_name[0] = '\0';
          strcat(log_file_name,packet_data);
          // ...
        }
      }
    }
  }
```

System Chart

Cubesat Space Protocol (CSP) → ADCS Server

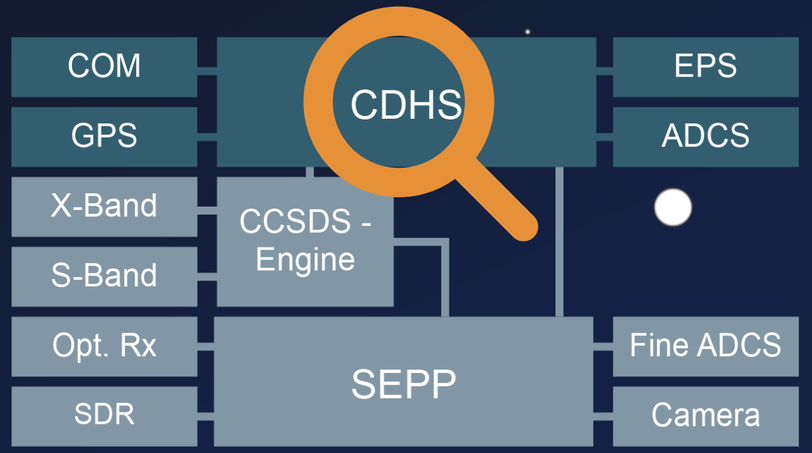
```c
void task_adcs_servr() {
  char log_file_name [32];

  csp_listen(socket, 10);
  csp_bind(socket, port);

  do {
    do {
      conn = csp_accept(socket, 0xff);
    } while (do_wait_for_conn);

    packet = csp_read(conn, 10);
    if (packet) {
      packet_data = packet->data;
      switch(*packet_data) {
        // [...]
        case SET_LOGFILE: {
          packet_data = packet->data + 0xf;
          log_file_name[0] = '\0';
          strcat(log_file_name,packet_data);
          // ...
        }
      }
    }
  }
```



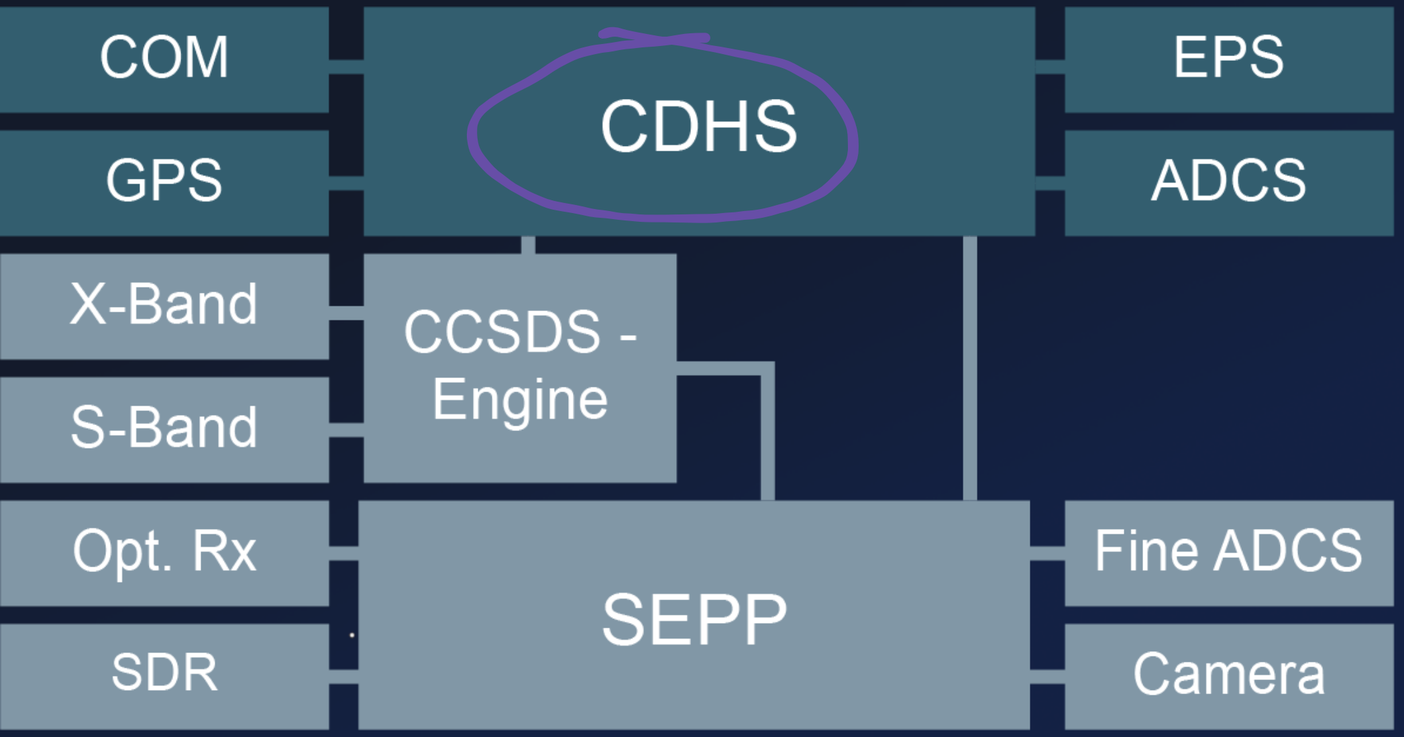
System Chart

Cubesat Space Protocol (CSP) → ADCS Server

```c
void task_adcs_servr() {
  char log_file_name [32];

  csp_listen(socket, 10);
  csp_bind(socket, port);

  do {
    do {
      conn = csp_accept(socket, 0xff);
    } while (do_wait_for_conn);

    packet = csp_read(conn, 10);
    if (packet) {
      packet_data = packet->data;
      switch(*packet_data) {
        // [...]
        case SET_LOGFILE: {
          packet_data = packet->data + 0xf;
          log_file_name[0] = '\0';
          strcat(log_file_name,packet_data);
          // ...
        }
      }
    }
  }
```

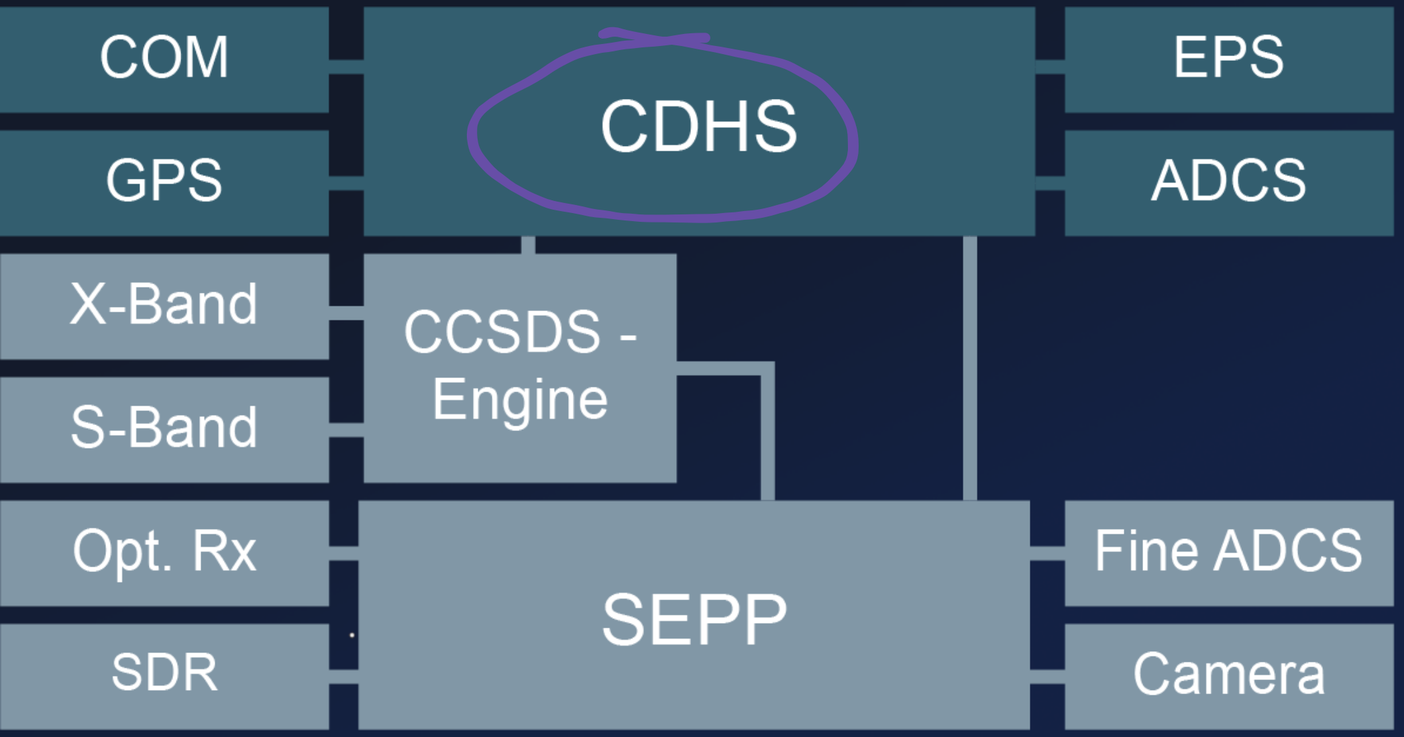


System Chart



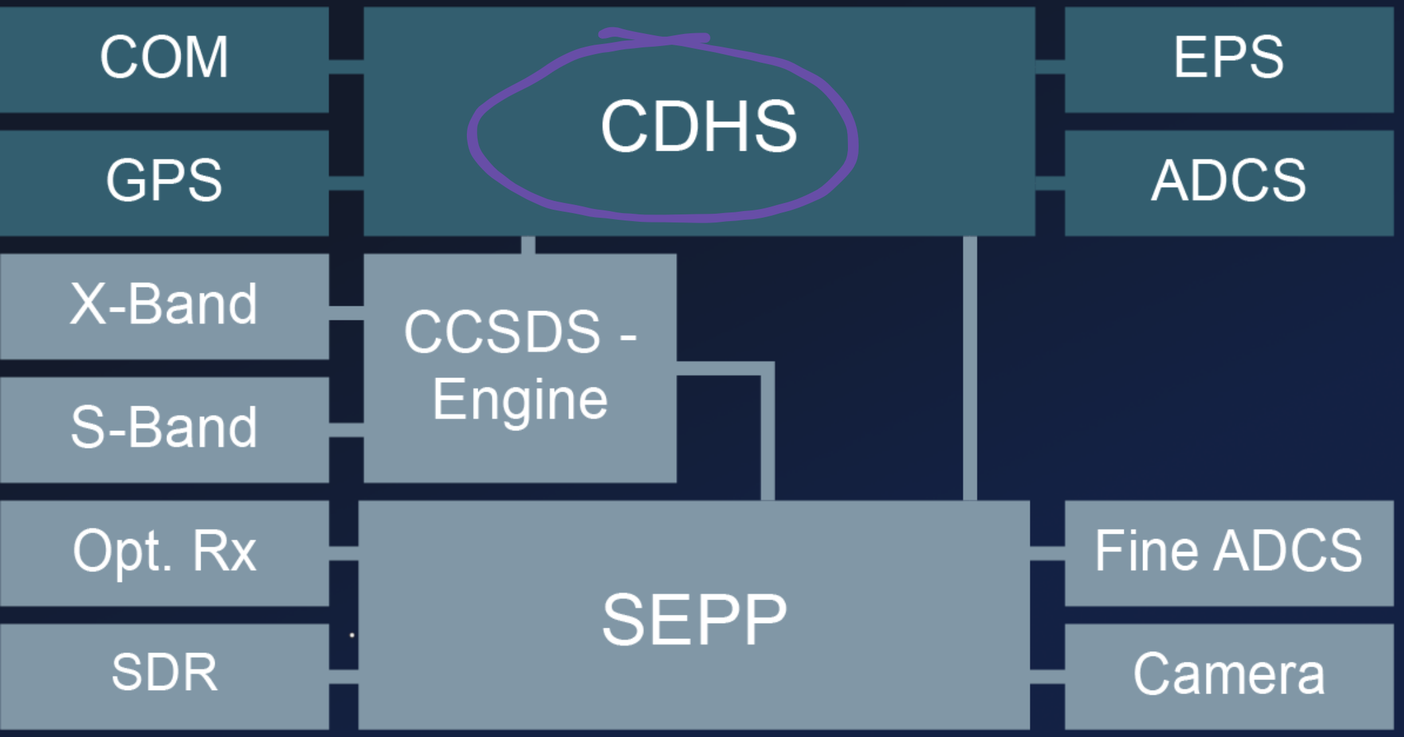
1. Bypass COM Protection
2. Dangerous / Vulnerable TC

System Chart



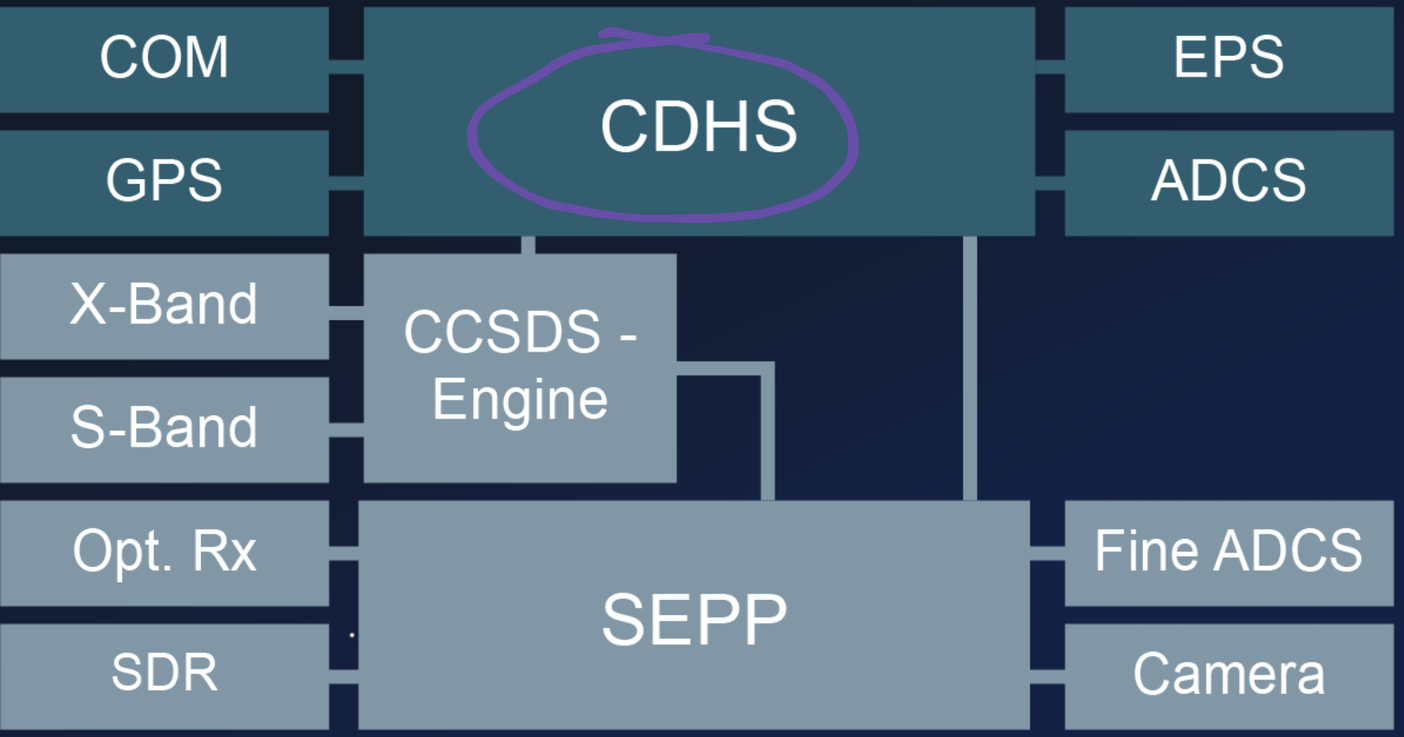
1. Bypass COM Protection
2. Dangerous / Vulnerable TC

System Chart



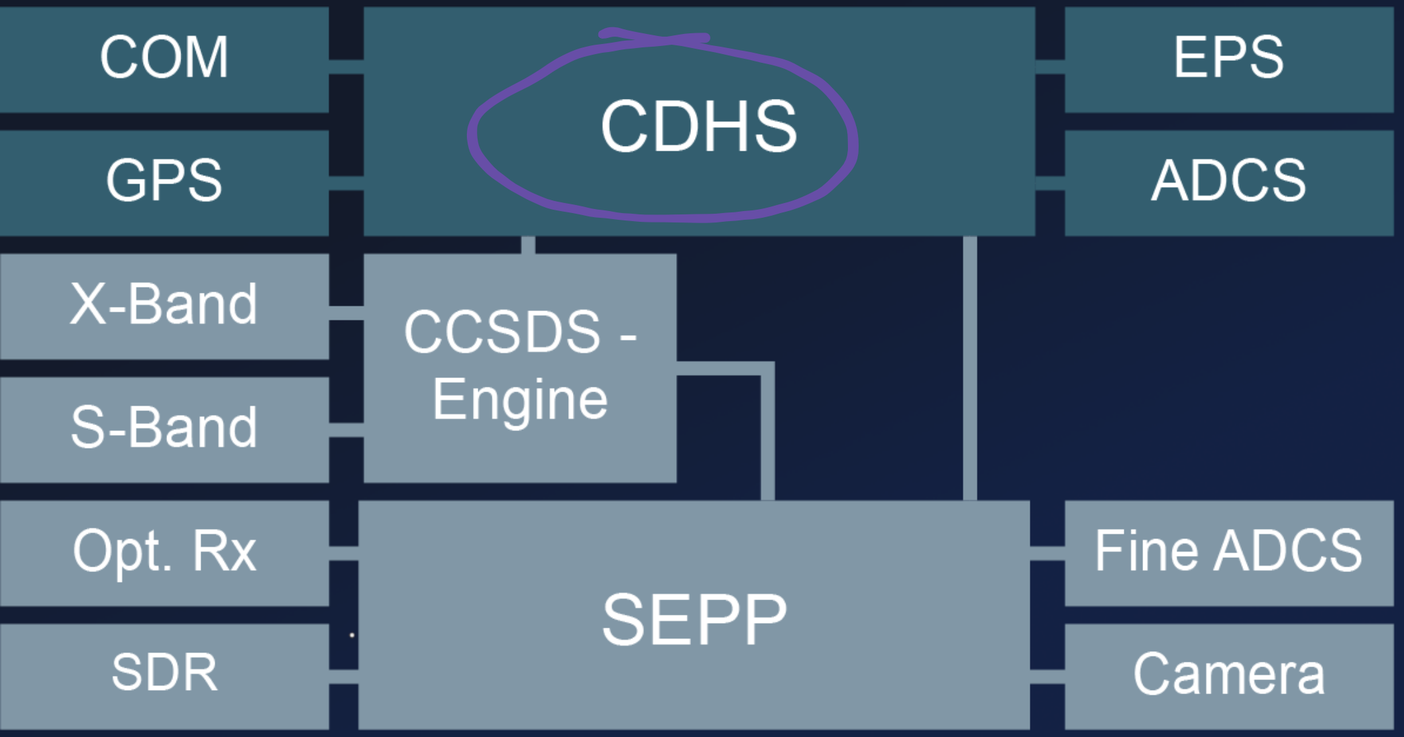
1. Bypass COM Protection
2. Dangerous / Vulnerable TC
3. Hijack Bus Control Flow
   - No OS-Defenses
     - ASLR
     - NX Stack
   - No SW-Defenses
     - Stack Cookies

System Chart



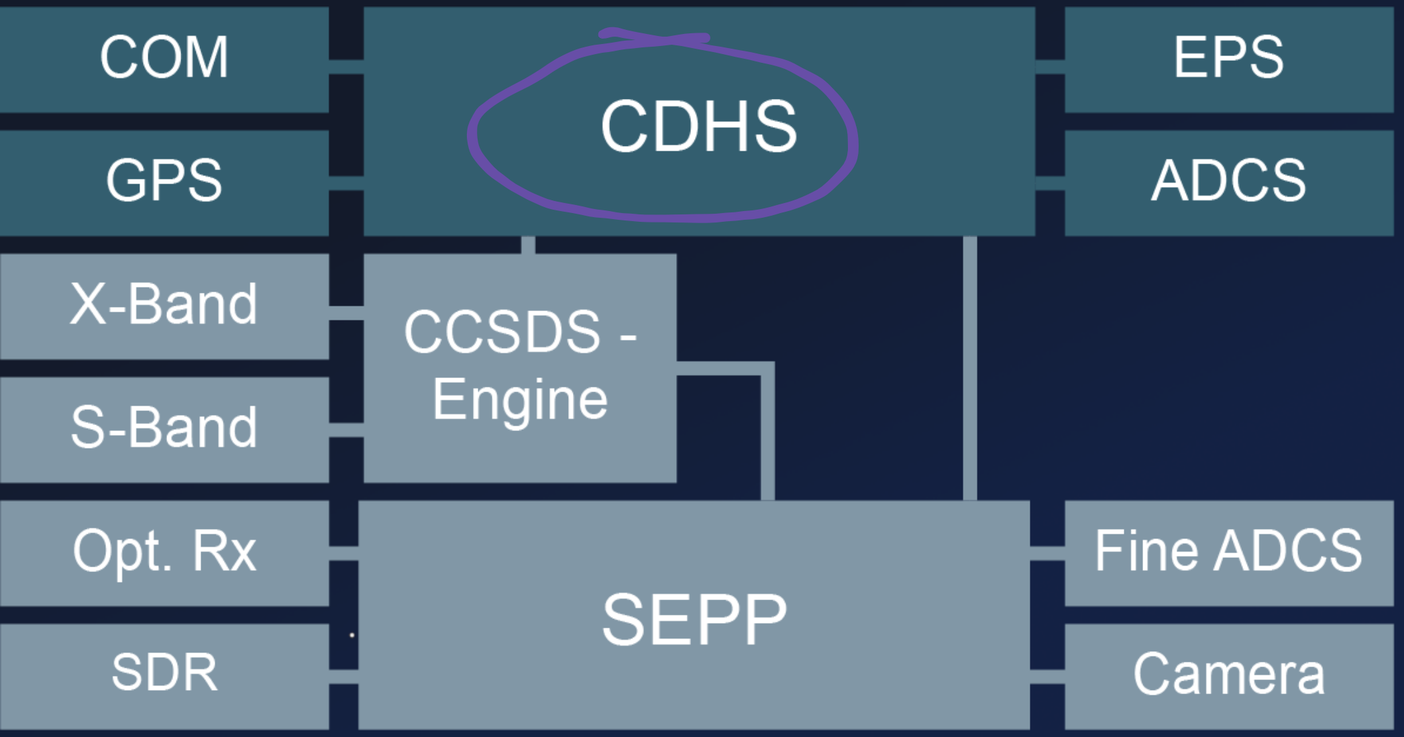
1. Bypass COM Protection
2. Dangerous / Vulnerable TC
3. Hijack Bus Control Flow
    - No OS-Defenses
        - ASLR
        - NX Stack
    - No SW-Defenses
        - Stack Cookies

System Chart



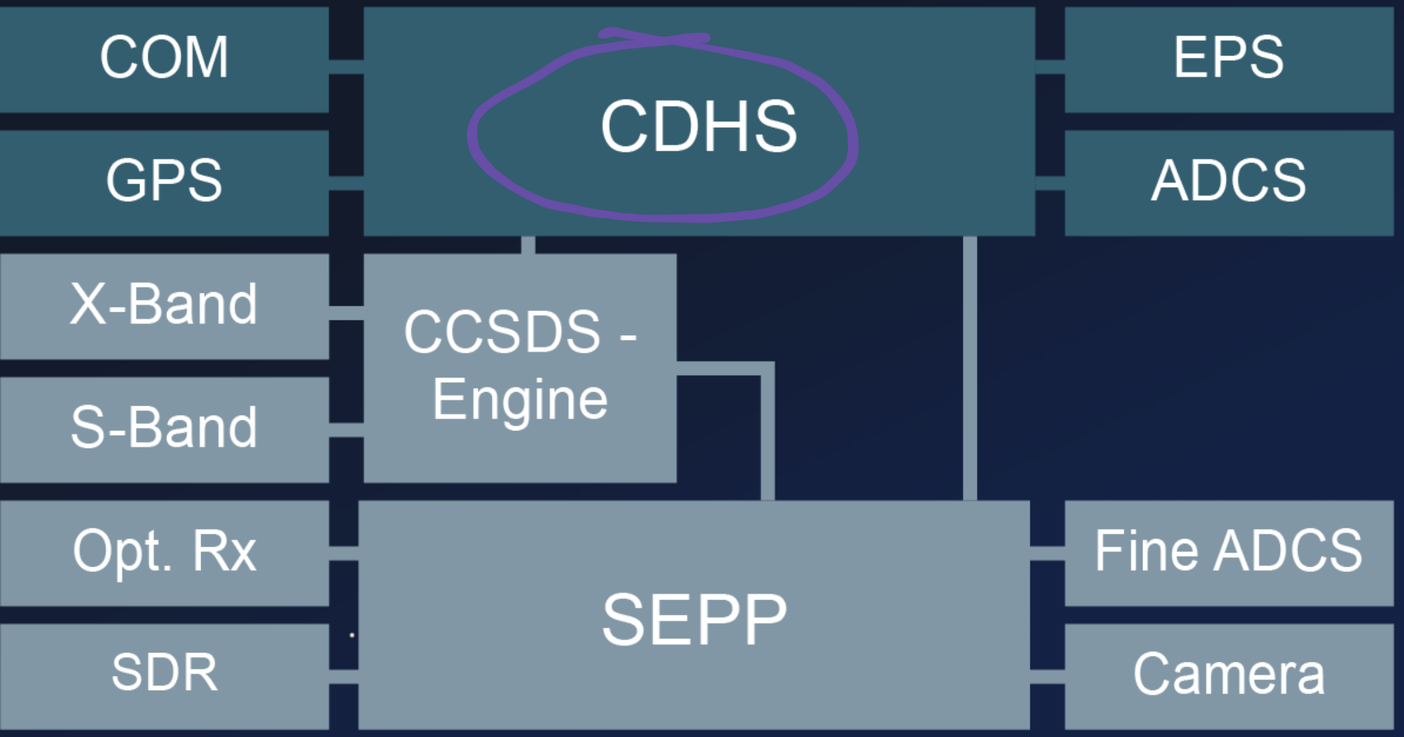
1. Bypass COM Protection
2. Dangerous / Vulnerable TC
3. Hijack Bus Control Flow
4. Full Bus Privileges

System Chart



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

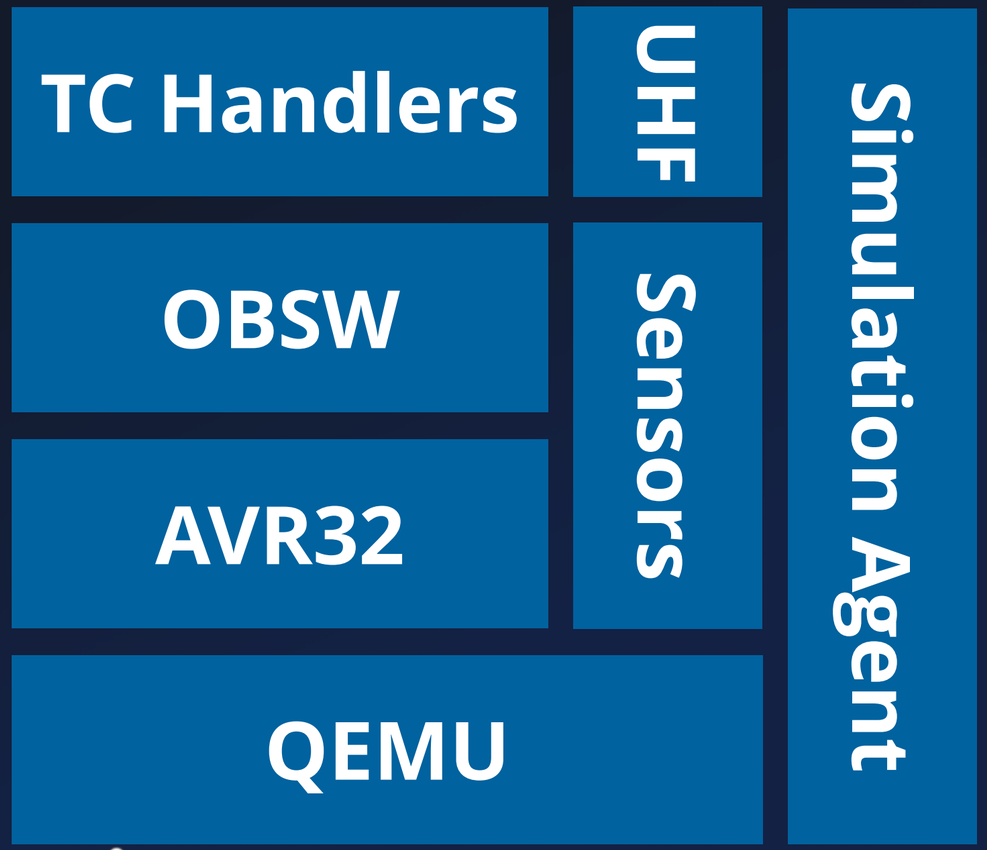
System Chart



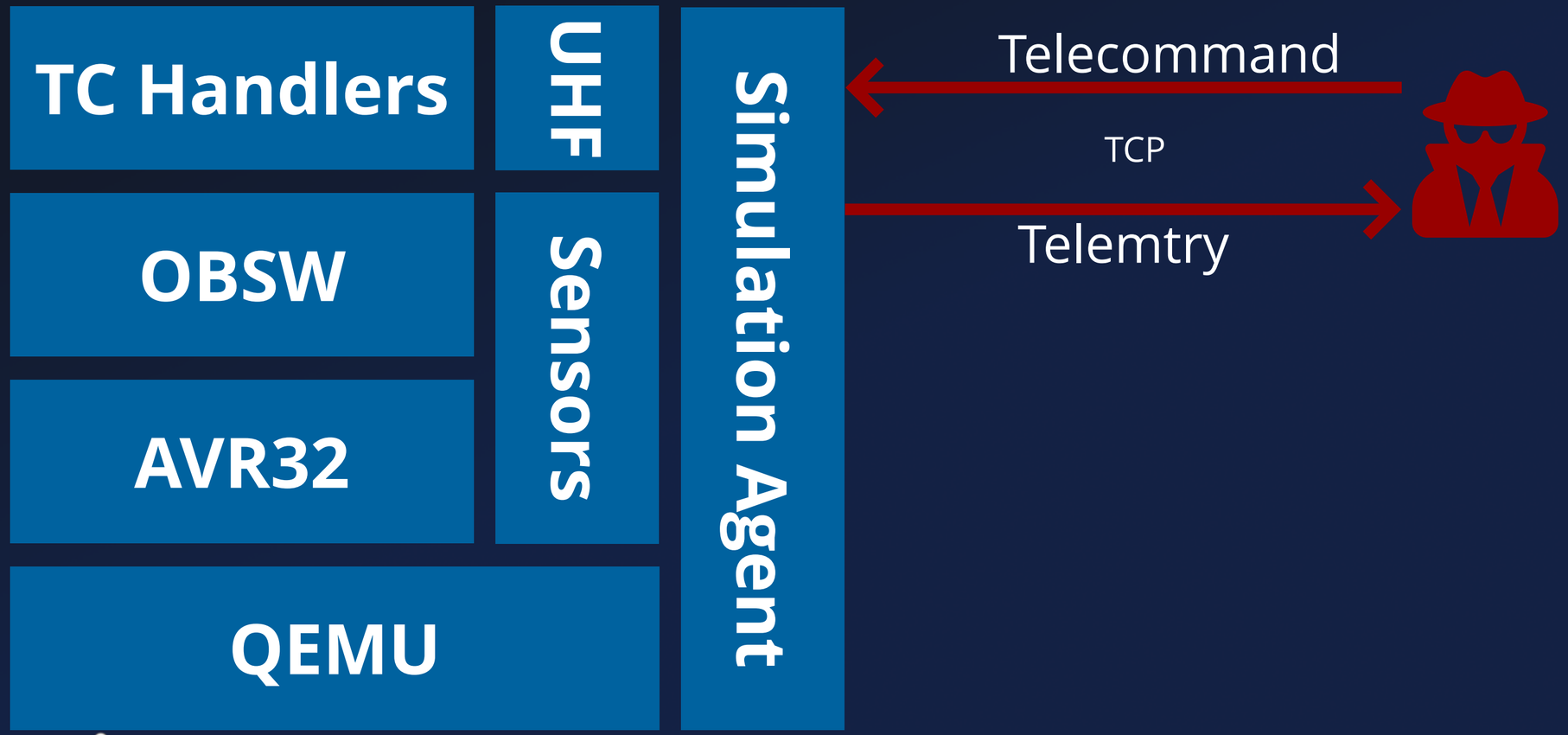
1. Bypass COM Protection
2. Dangerous / Vulnerable TC
3. Hijack Bus Control Flow
4. Full Bus Privileges
    - Privilege-free RTOS

Demo Setup

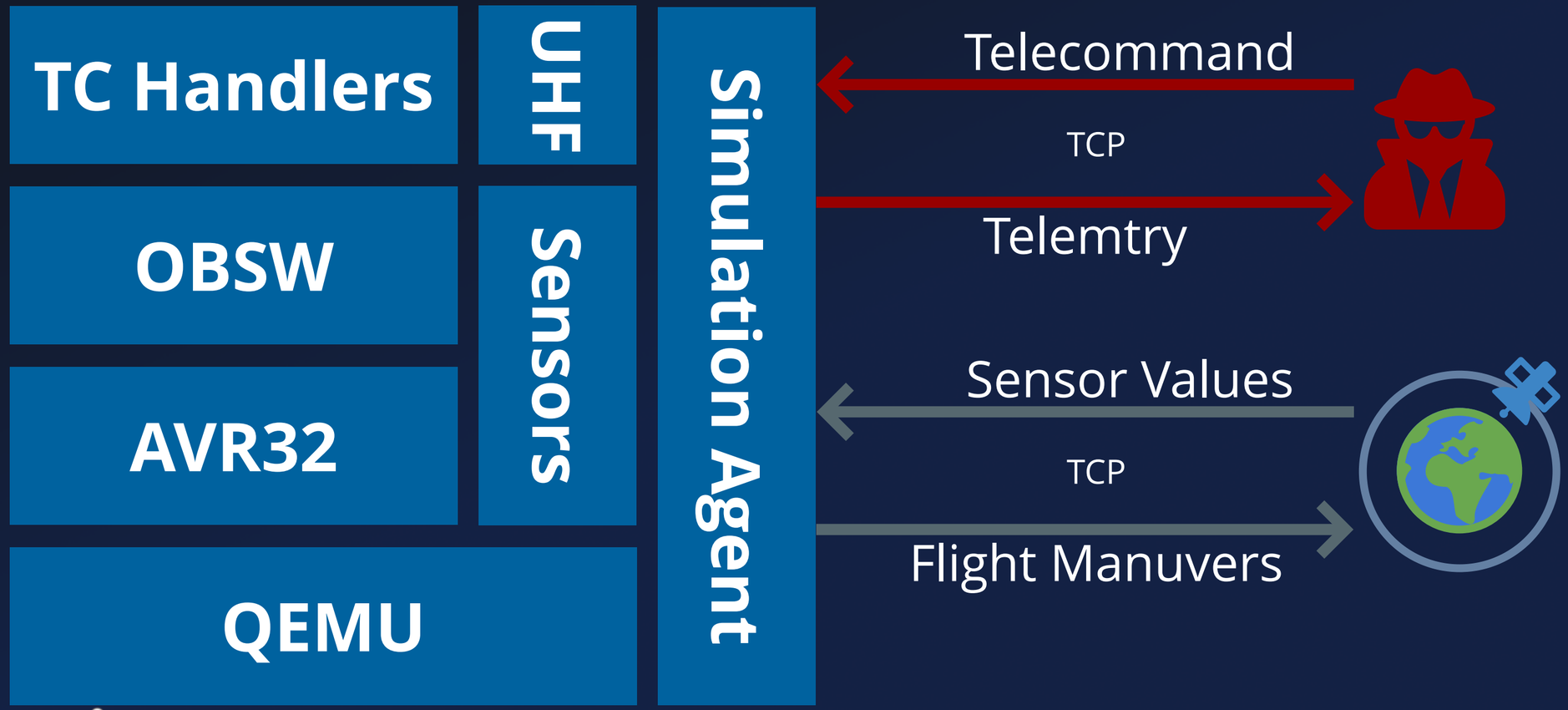
Emulation Overview



Emulation Overview



Emulation Overview



Live Demo

```
$> ./access-satellite.
[*] Uploading TC ...
[*] Deploying payload ...
[*] Payload written to flash ...
[*] Rebooting ...
[*] $$$
```

Lessons Learnt

Firmware Attacks on Satellite are a thing

Lessons Learnt

Firmware Attacks on Satellite are a thing

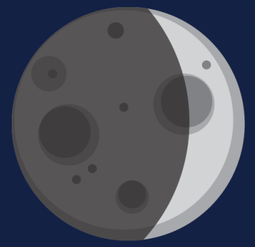
Just TC Execution is not Enough

Lessons Learnt

Firmware Attacks on Satellite are a thing

Just TC Execution is not Enough

Missing State-of-the-Art Defense

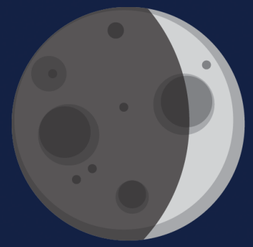


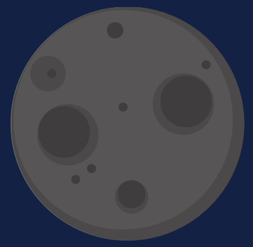
Lessons Learnt

Firmware Attacks on Satellite are a thing

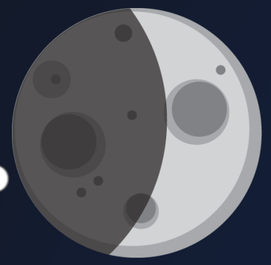
Just TC Execution is not Enough

Missing State-of-the-Art Defense





90s-style Buffer Overflows in Space Systems



Thanks!

- Firmware Attacks on Satellite
  - External Attacker → COM → CDHS → $$$
- Satellite Exploitation Objectives
- Vulnerable & Dangerous TCs
- Missing OS & SW-Defenses

Also visit my Talk @ REcon, Montreal, Canada


@jwillbold


/jwillbold


@jwillbold

Johannes Willbold - johannes.willbold@rub.de