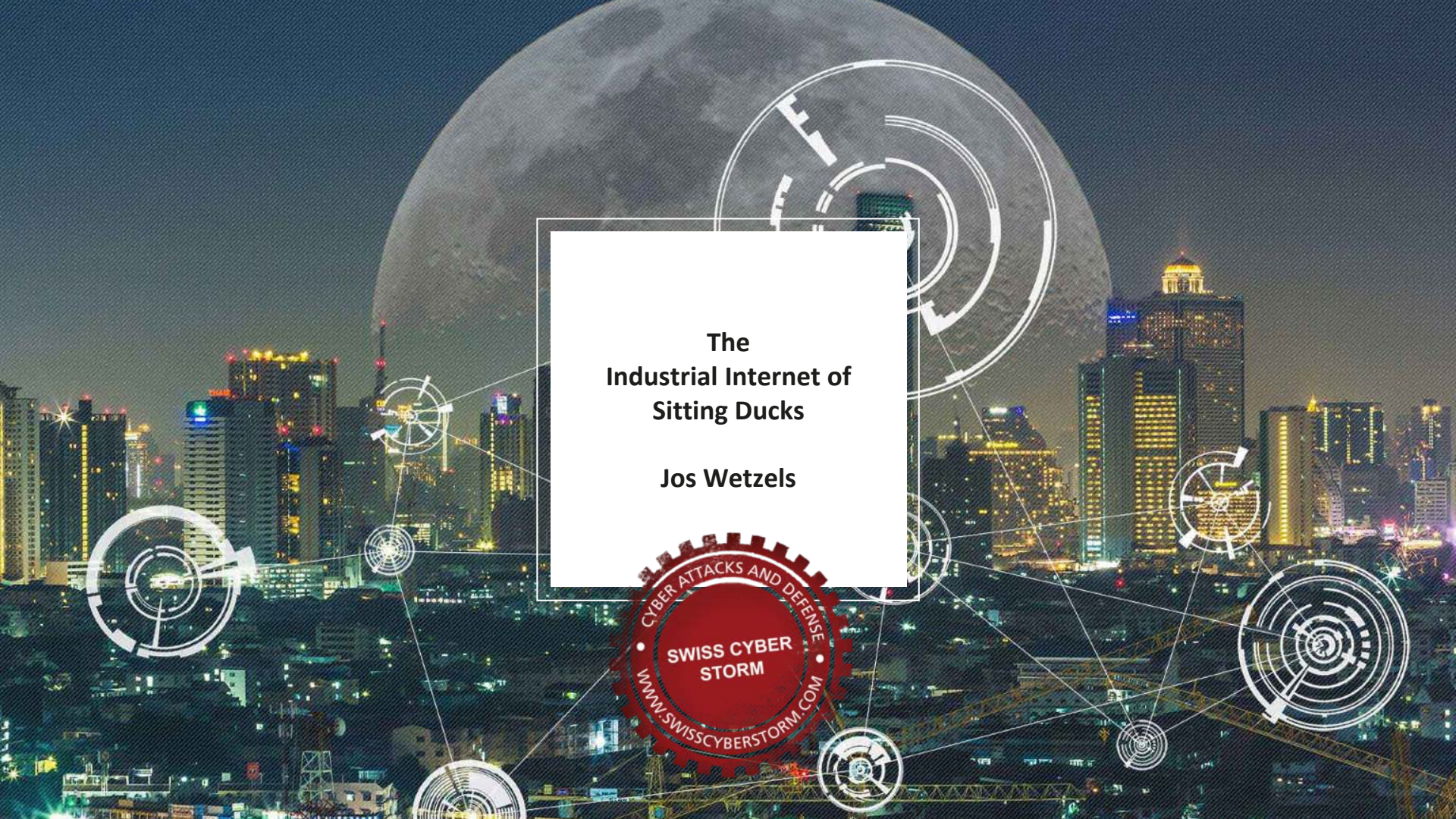
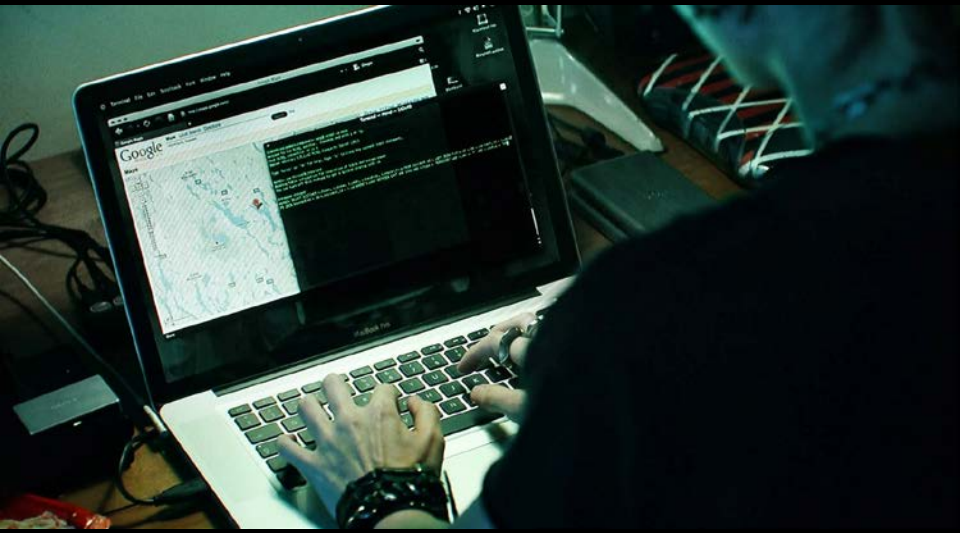

The
Industrial Internet of
Sitting Ducks
Jos Wetzels
CYBER ATTACKS AND DEFENSE
SWISS CYBER
STORM
WWW.SWISSCYBERSTORM.COM



Jos Wetzels

Midnight Blue

Independent Security Researcher

ICS
IoT
Automotive
Medical
Access Controls
Networking & Firewalls

UNIVERSITY OF TWENTE.

(Former) Security Researcher

Critical Infrastructure
Embedded BinSec
HIDS / NIDS

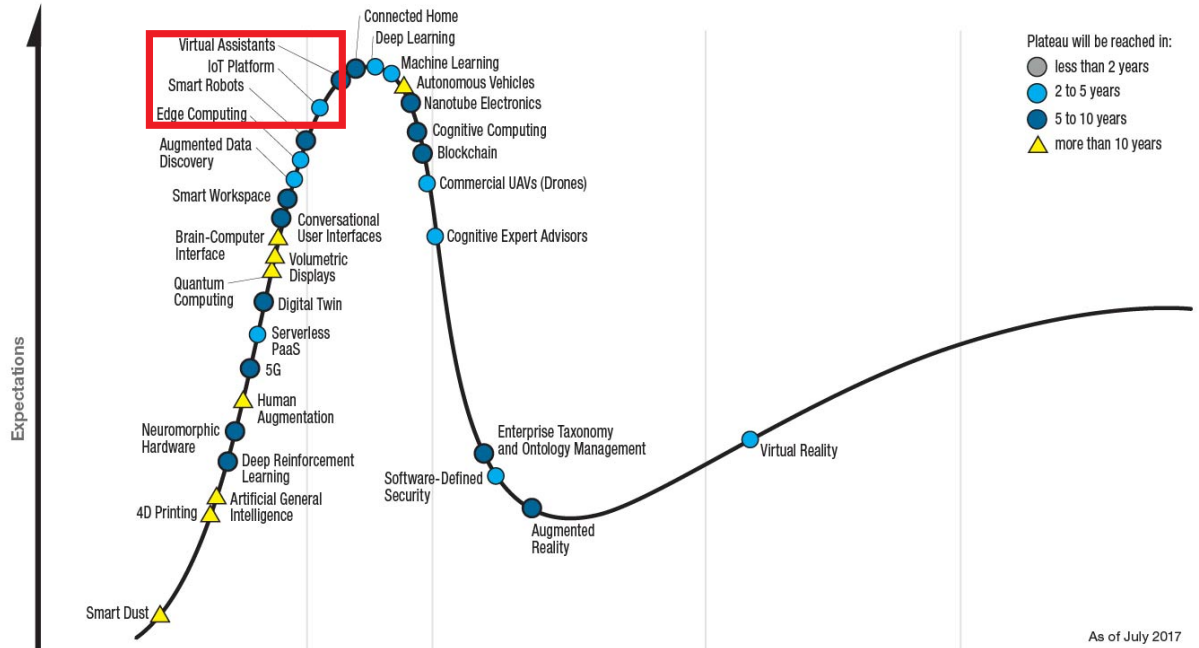
@s4mvartaka

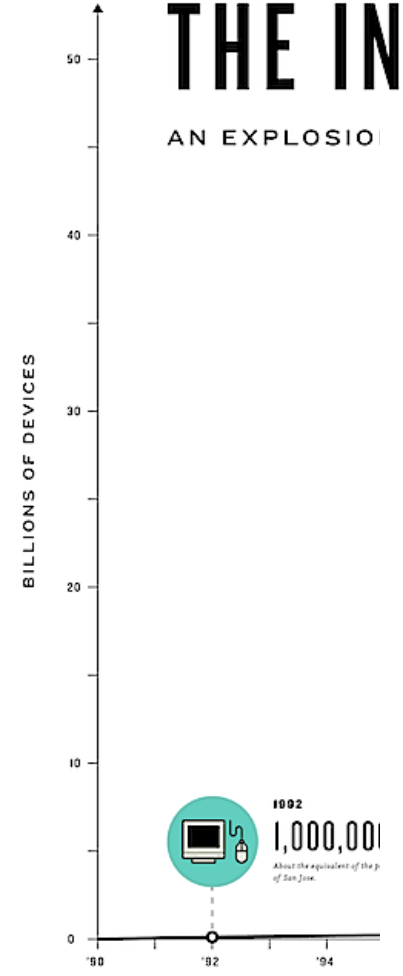
www.midnightbluelabs.com
samvartaka.github.io

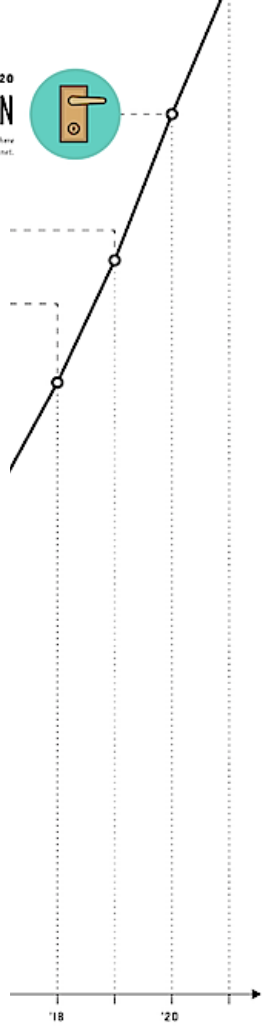
THE INTERNET OF THINGS

AN EXPLOSIO

Gartner Hype Cycle for Emerging Technologies, 2017







YEAR

“

We are seeing a market failure for cybersecurity and privacy. (…) Currently there is no basic level, no level zero defined for the security and privacy of connected and smart devices.

- ENISA / Infineon / NXP / STM

Pizza Hut made shoes that will order pizza

Parents warned over exploding fidget spinners powered by Bluetooth

You Can Only Wash Google And Levi's New $350 'Connected' Jacket Ten Times

Moxie: Showerhead with wireless speaker

★☆☆☆☆ **Scdragon** · 8 days ago

Wifi does not connect

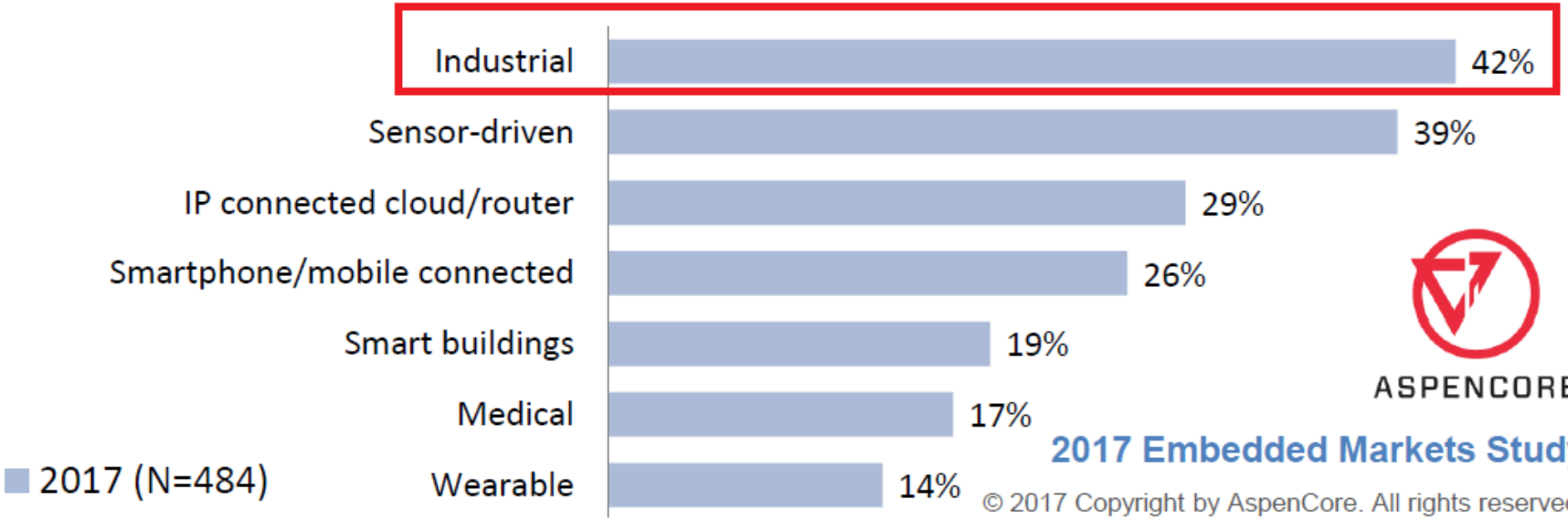
Wifi connectivity does not work. Stuck in setup mode.

Maid: Smart Microwave Oven

INDUSTRIAL
Internet of Things

CONSUMER
Internet of Things

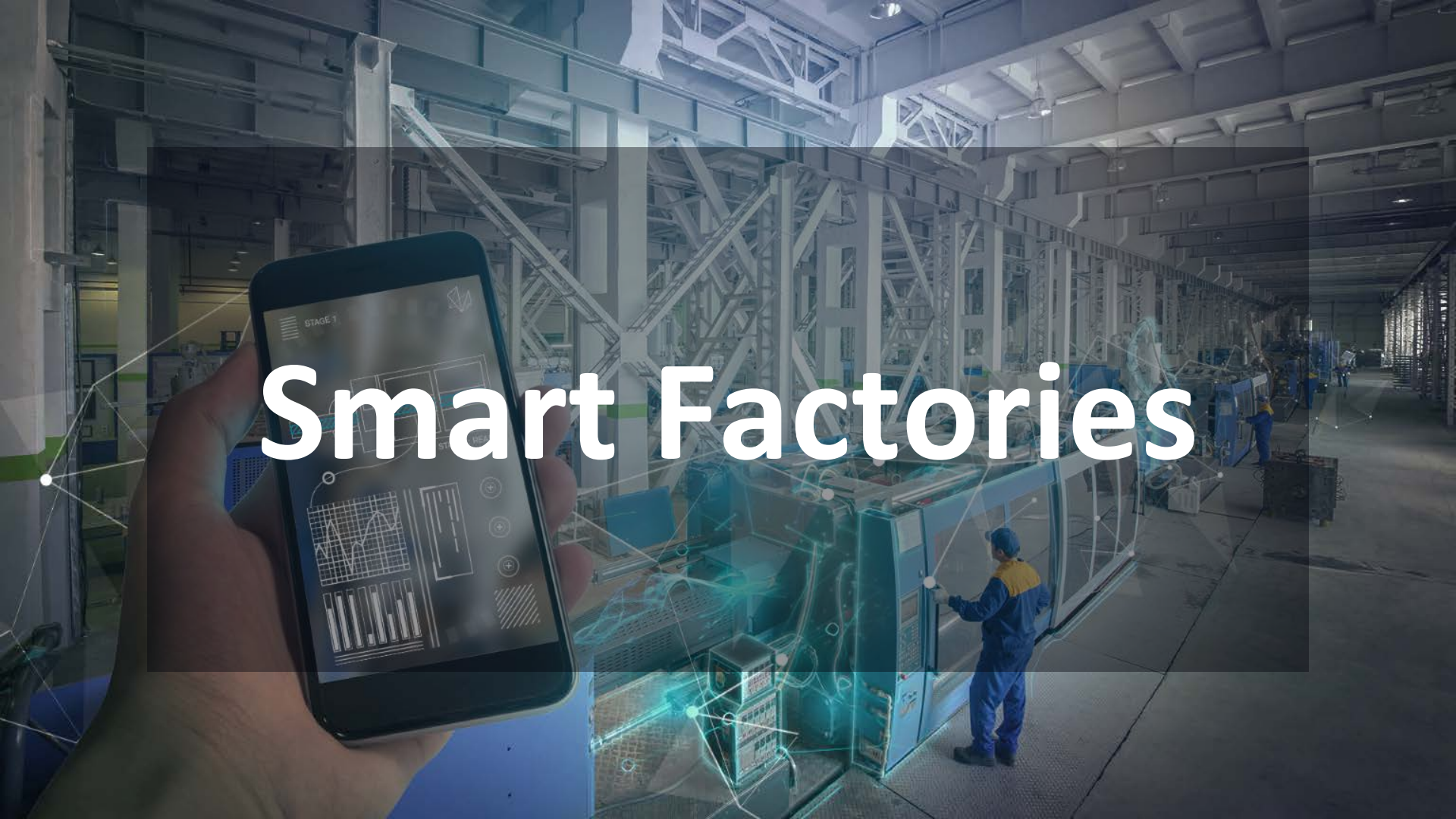
If you are developing Internet of Things (IoT) applications, please indicate the type of application.



ASPENCORE

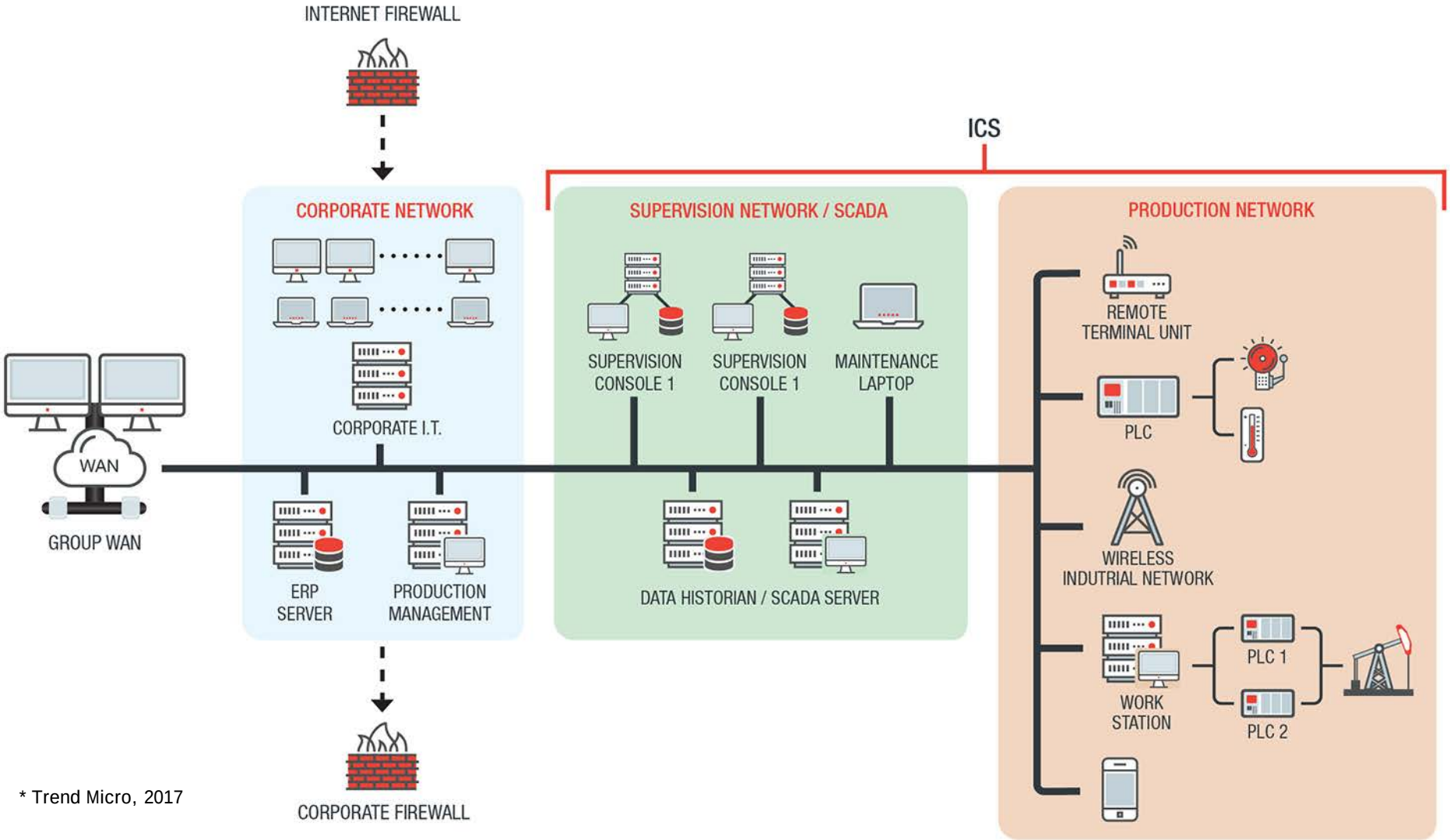
2017 Embedded Markets Study

NATIONAL INSTRUMENTS



Smart Factories


Critical
Infrastructure



* Trend Micro, 2017

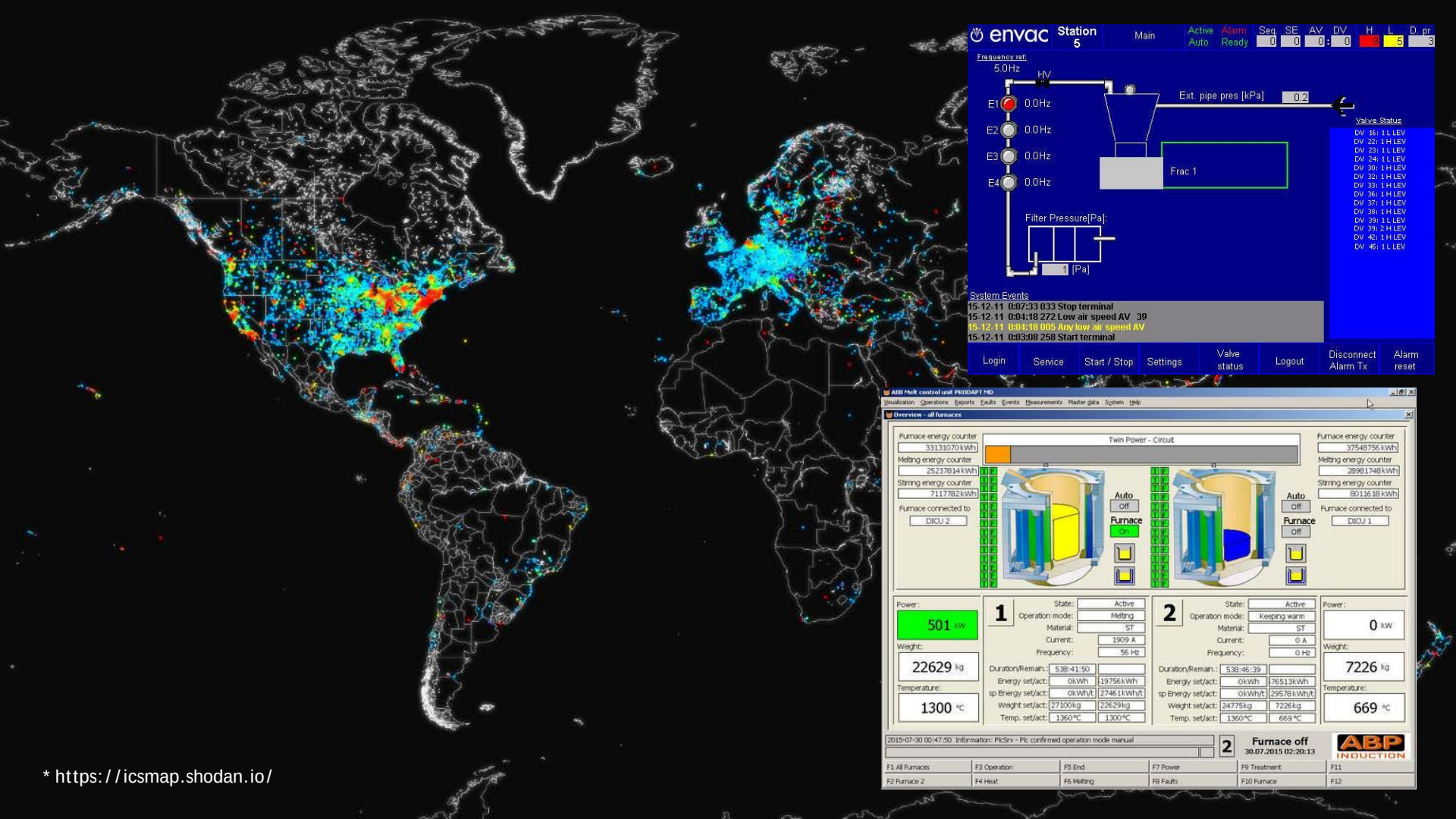
PLCs

RTUs

Gateways, Modems

HMIs

Sensors, Actuators



* https://icsmap.shodan.io/


Stuxnet

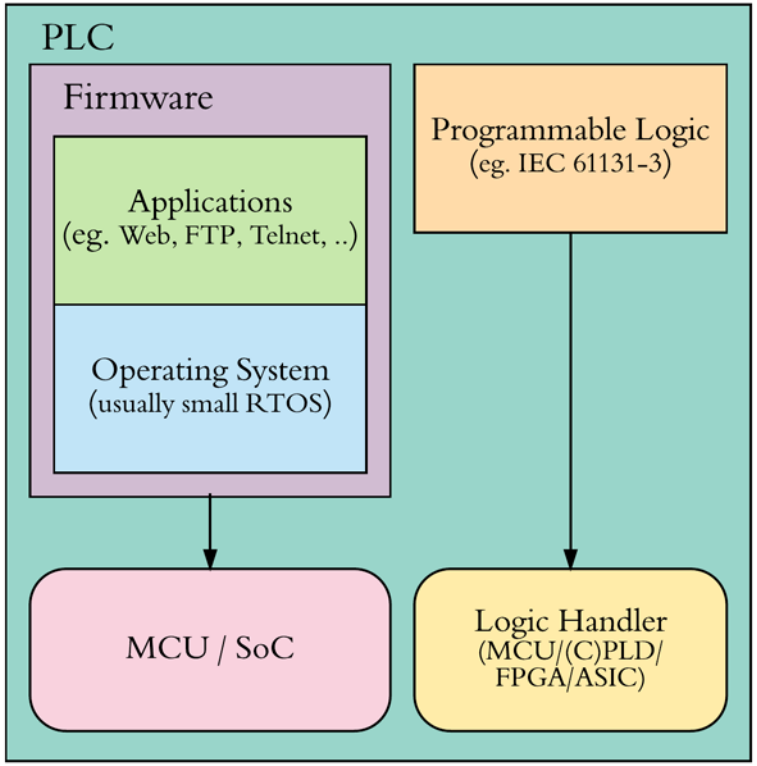

Sandworm

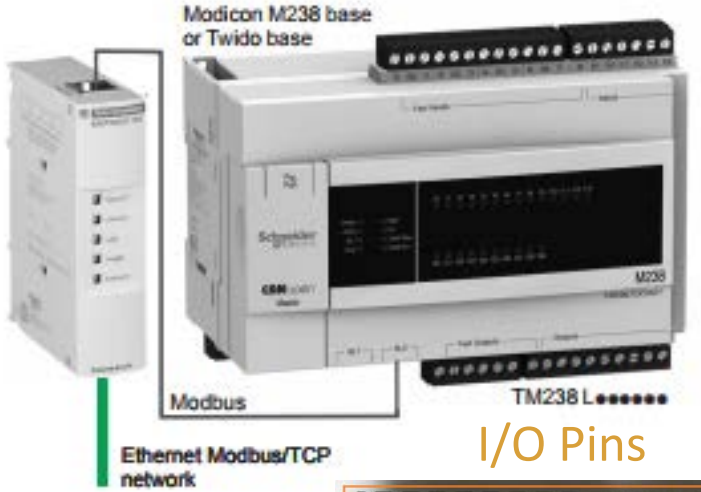
The Economics of Ransomware: How SCADA/ICS Changes the Equation

Researchers Create PoC Ransomware That Targets ICS/SCADA Systems

Take Down: Hackers Looking to Shut Down Factories for Pay

* Stephen A. Ridley, Senrio Inc., 2016



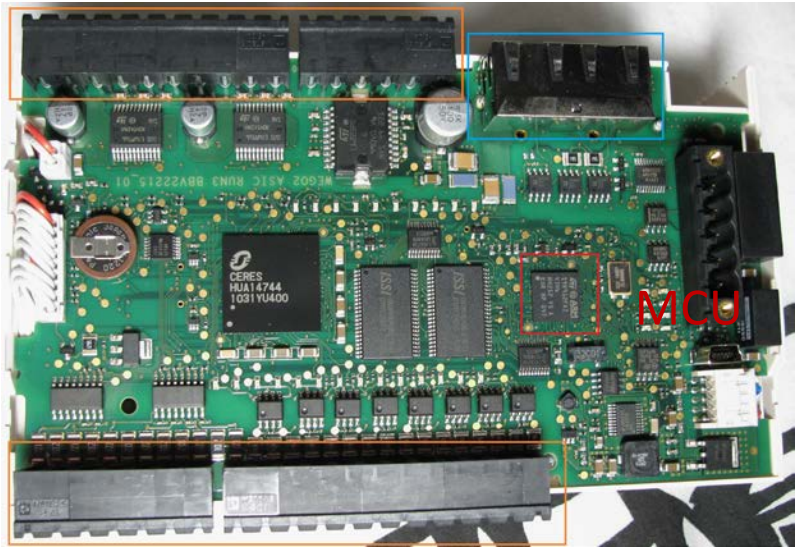


Schneider Electric

* https://www.eevblog.com, 2014

I/O Pins

Serial Link



* David Barksdale, Jeremy Brown, 2016

Typical ICS Security #1: Opto 22 OPTEMU-SNR-DR2*

- **Energy Monitoring & Control Device**
- **Managed over Ethernet**
  FTP, SNMP, OptoMMP (unauthenticated), PAC Control (unauthenticated)
- **Use OptoMMP to**
  disable IP filtering, enable FTP, get FTP credentials
- **Upload firmware & reflash over FTP**
  (no firmware signing)

* K. Reid Wightman, Rubén Santamarta, 2011-2012

Typical ICS Security #2: Modicon Quantum PLC*

- **Large PLC for process applications**
- **FTP with hardcoded backdoor**
  Read/Write Access to configuration, firmware, passwords, etc.
- **Telnet with hardcoded backdoor**
  Is actually a C interpreter…
- **Unauthenticated Modbus Extension**
  Start/Stop PLC
  Overwrite programmable logic
  Etc.

Open Secret:
ICS Security Sucks

- Unauthenticated Plaintext Protocols
- Unauthenticated firmware & logic uploads
- Default Backdoor Passwords
- Absent or Infrequent Patching
- Etc.

(Some) Reasons Why

- **Insecure by Design**
  Not designed for ‘open’ networks
  Security not considered

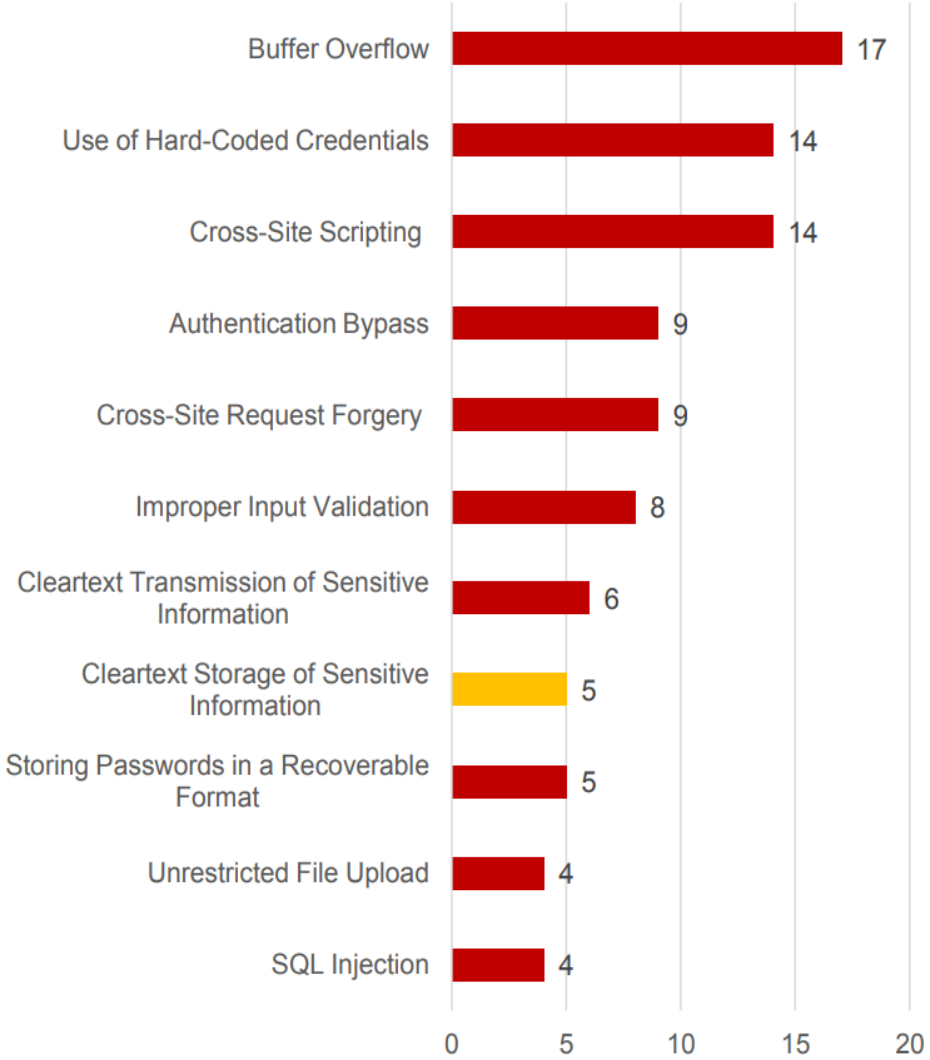
- **Device Lifespan**
  10+ years

  Age-old designs

  Ceased vendor support

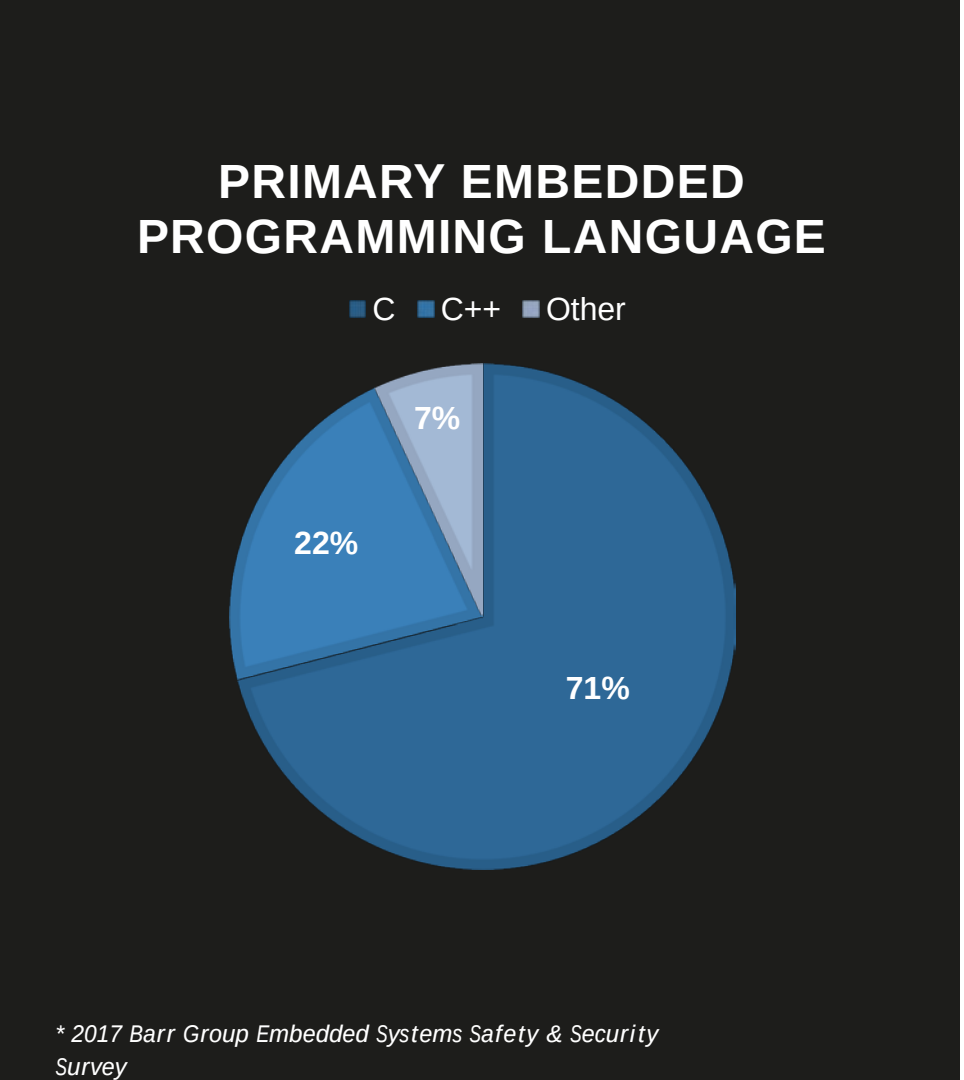
- **Legacy / Backwards Compatibility**
  Adhere to old, insecure standards


Let’s say we fix all this…
What’s next?



Memory Corruption is a Big Deal™ in Embedded

** 2016 ICS Vulnerabilities Statistics, Kaspersky*



** 2017 Barr Group Embedded Systems Safety & Security Survey*

Unsafe Languages Are Here To Stay

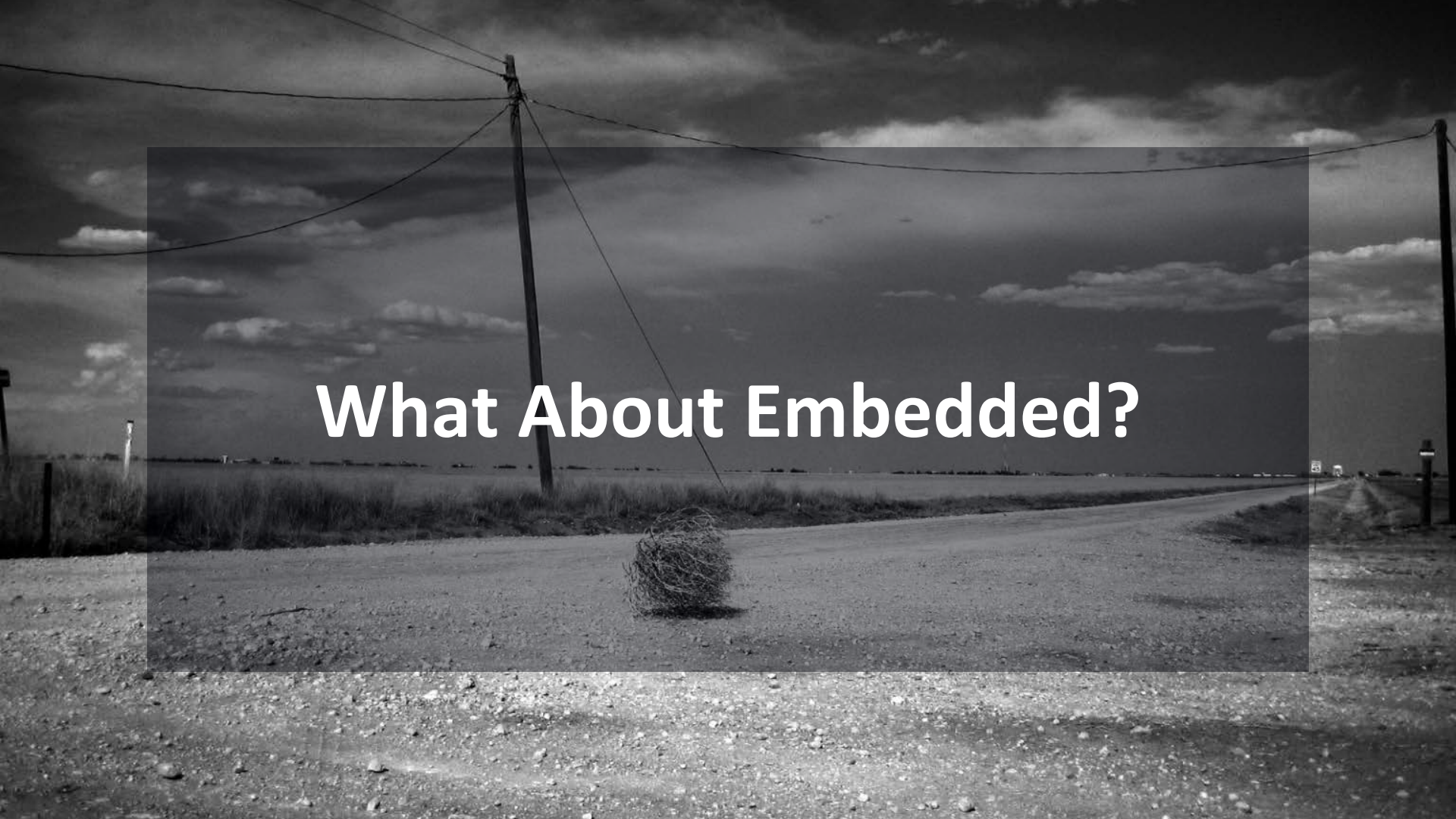
- **Ideally use safe languages**
  Java, Go, Erlang, Rust
- **But unsafe continues to dominate**
  C, C++


Exploit Mitigations

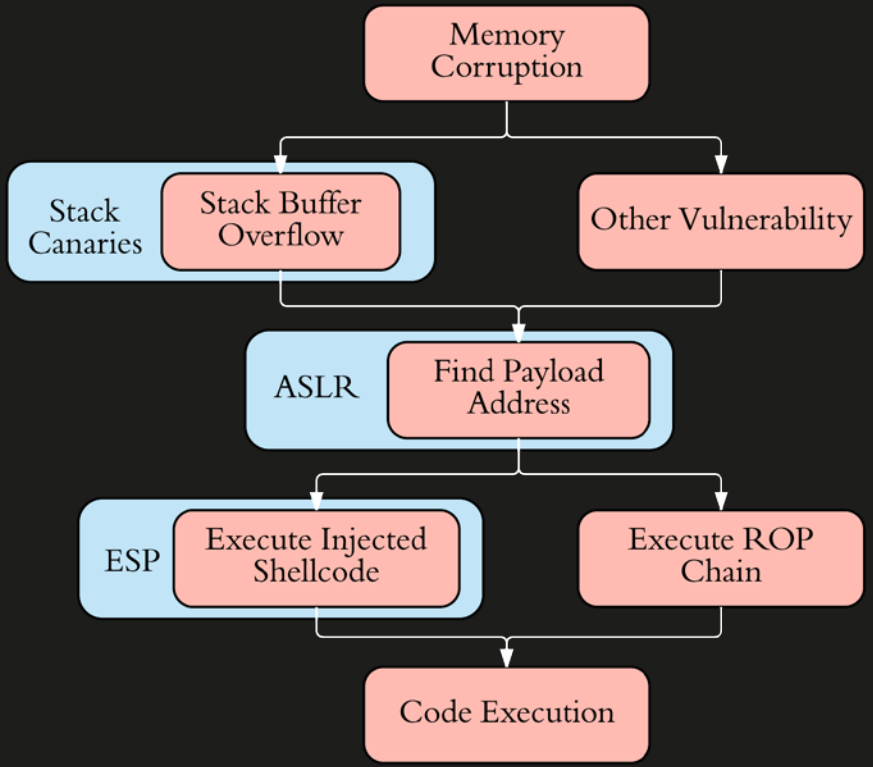
General Purpose Exploitation Has Been Getting Harder

2010

2016


What About Embedded?

Quantitative Analysis



Minimum Mitigation Baseline

- **ESP / DEP / NX / W^X**
  Non-exec. data memory
- **ASLR**
  Address Space Layout Randomization
- **Stack Canaries / SSP**
  Stack buffer overflow protection

Operating System Selection

- **Selected 45 Popular Embedded Oses**
- **High-end, Low-end, Linux/Windows/BSD-based, proprietary, etc.**
- **Evaluated Support For Mitigation Baseline**

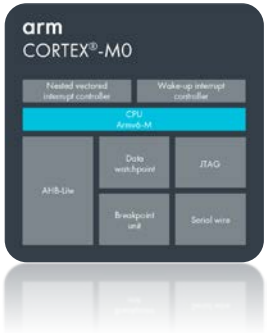
Optimistic Assesment


Mitigation Support
ESP
ASLR
Stack Canaries
100
90
80
70
60
50
40
30
20
10
0
All Oses
Non-Mobile
Non-Win/Lin/BSD
Deeply Embedded
NOW
THAT'S WHAT I CALL THE
90s

What’s Going On Here?

Usual Embedded Suspects

Resource Constraints



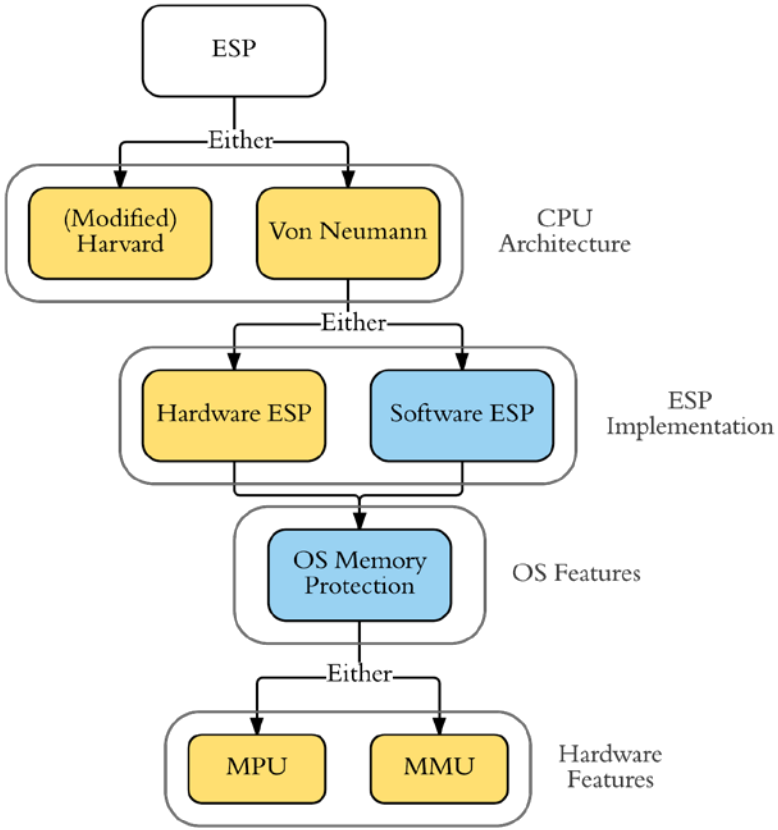
Hardware Limitations

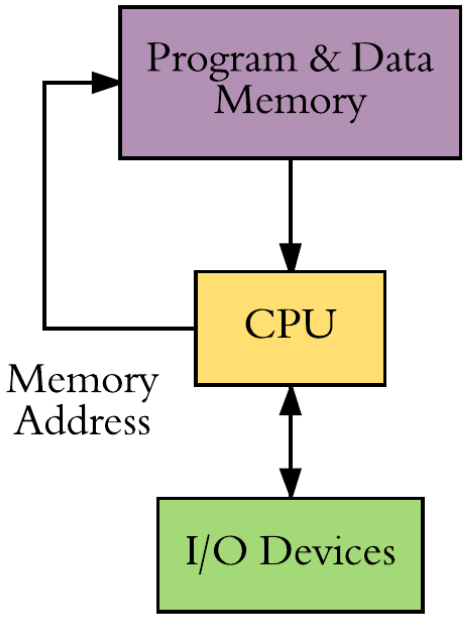
Cost Sensitivity

Random Number Generator Issues*

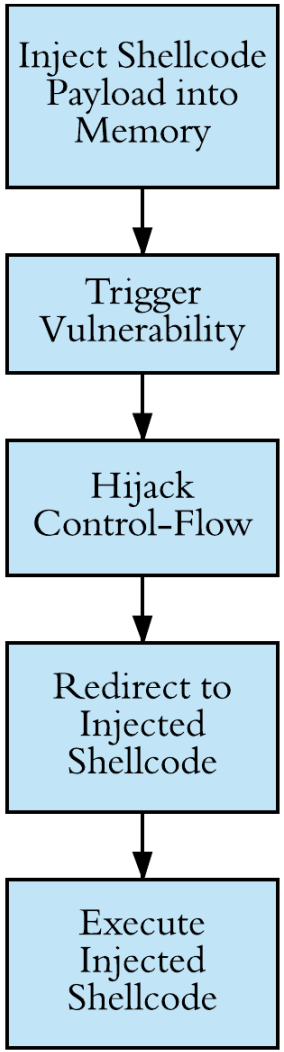
* '*Wheel of Fortune*' – Jos Wetzels, 33C3, 2016

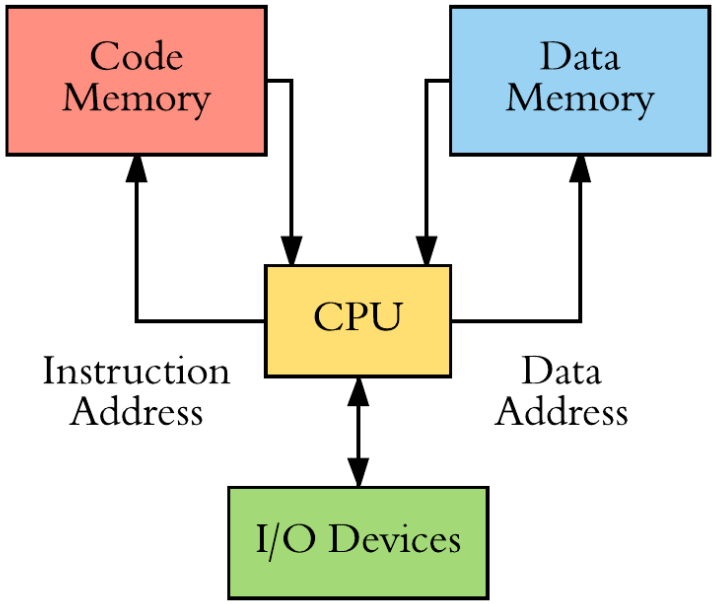
Hardware & Software Dependencies



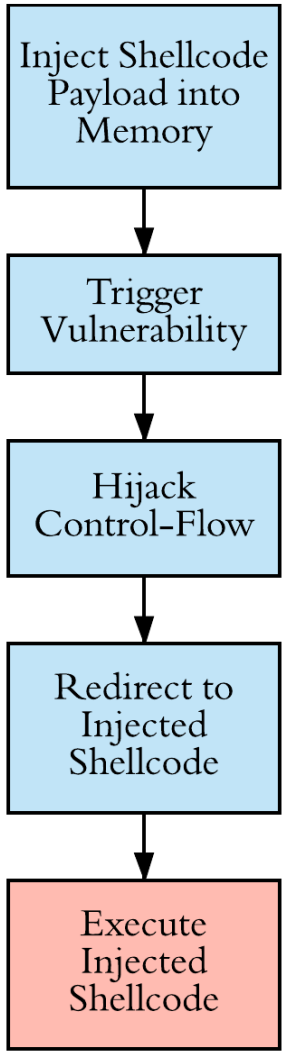


Von Neumann Architecture (VNA)

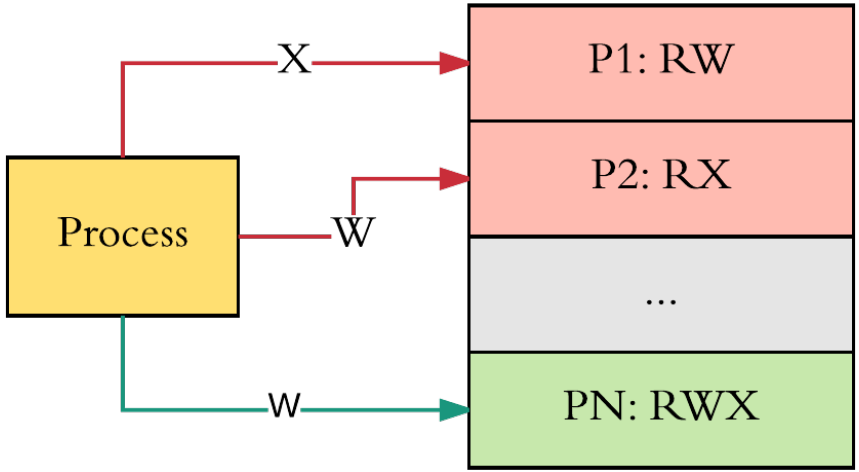


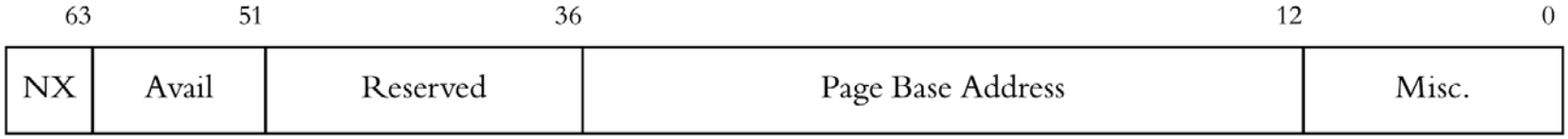


Harvard Architecture (HA)

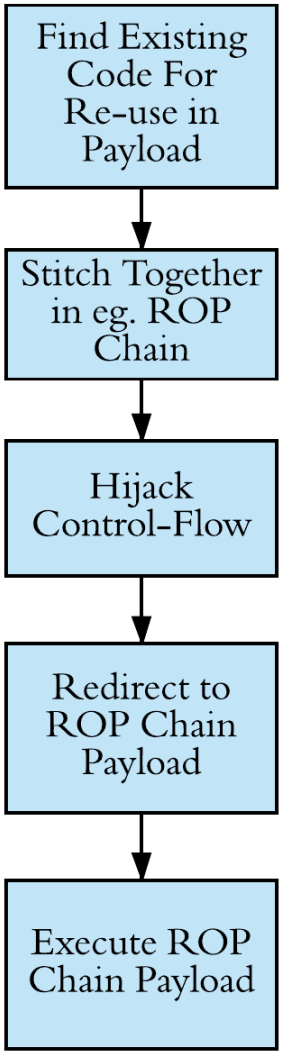


MPUs/MMUs & Hardware ESP Support

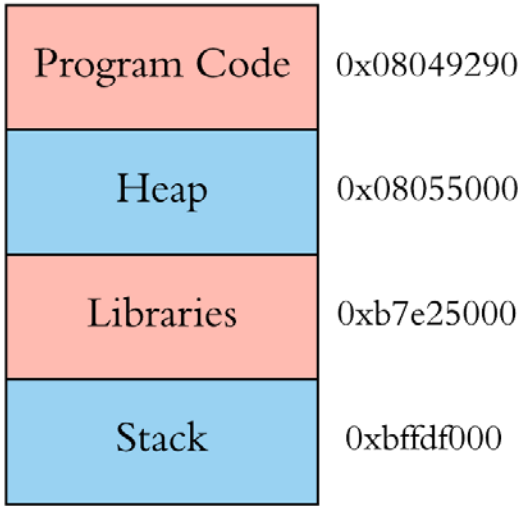


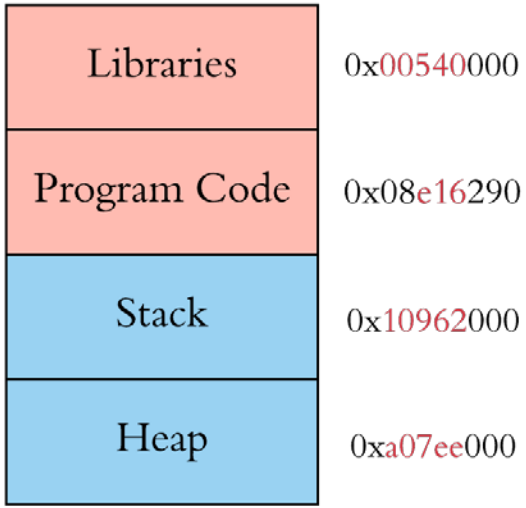


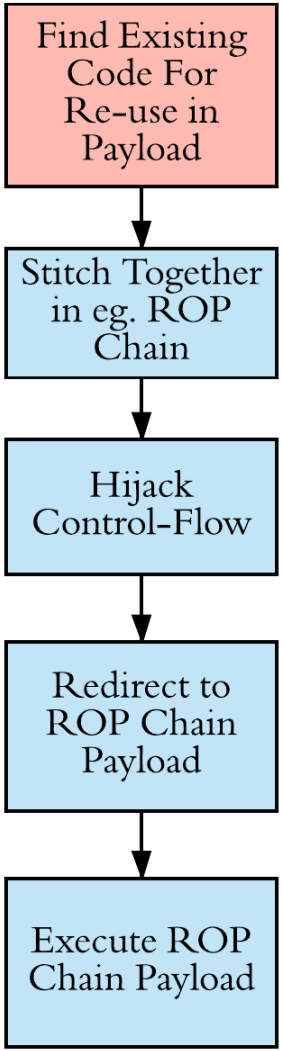
64-bit PTE (PAE mode)

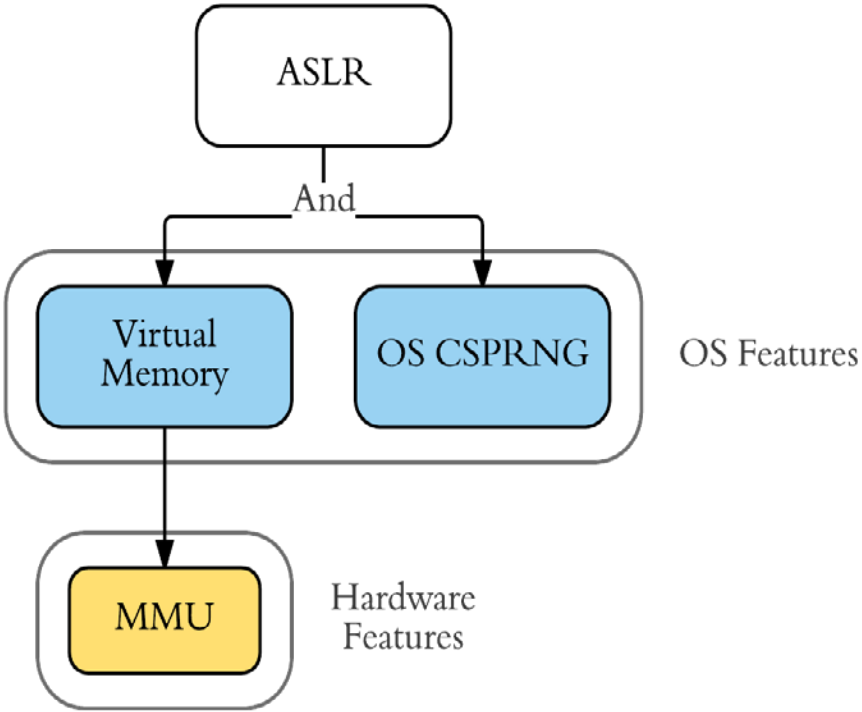


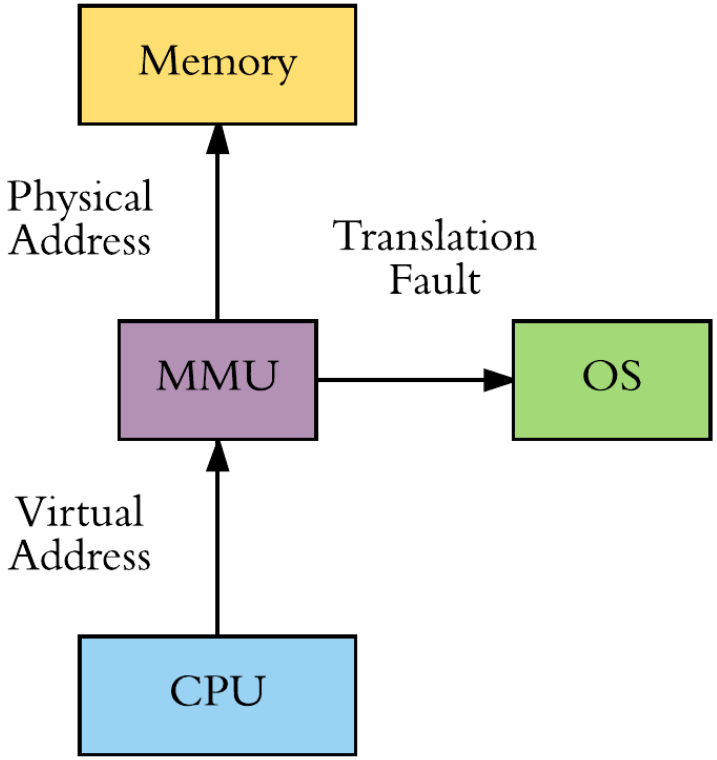
return oriented programming

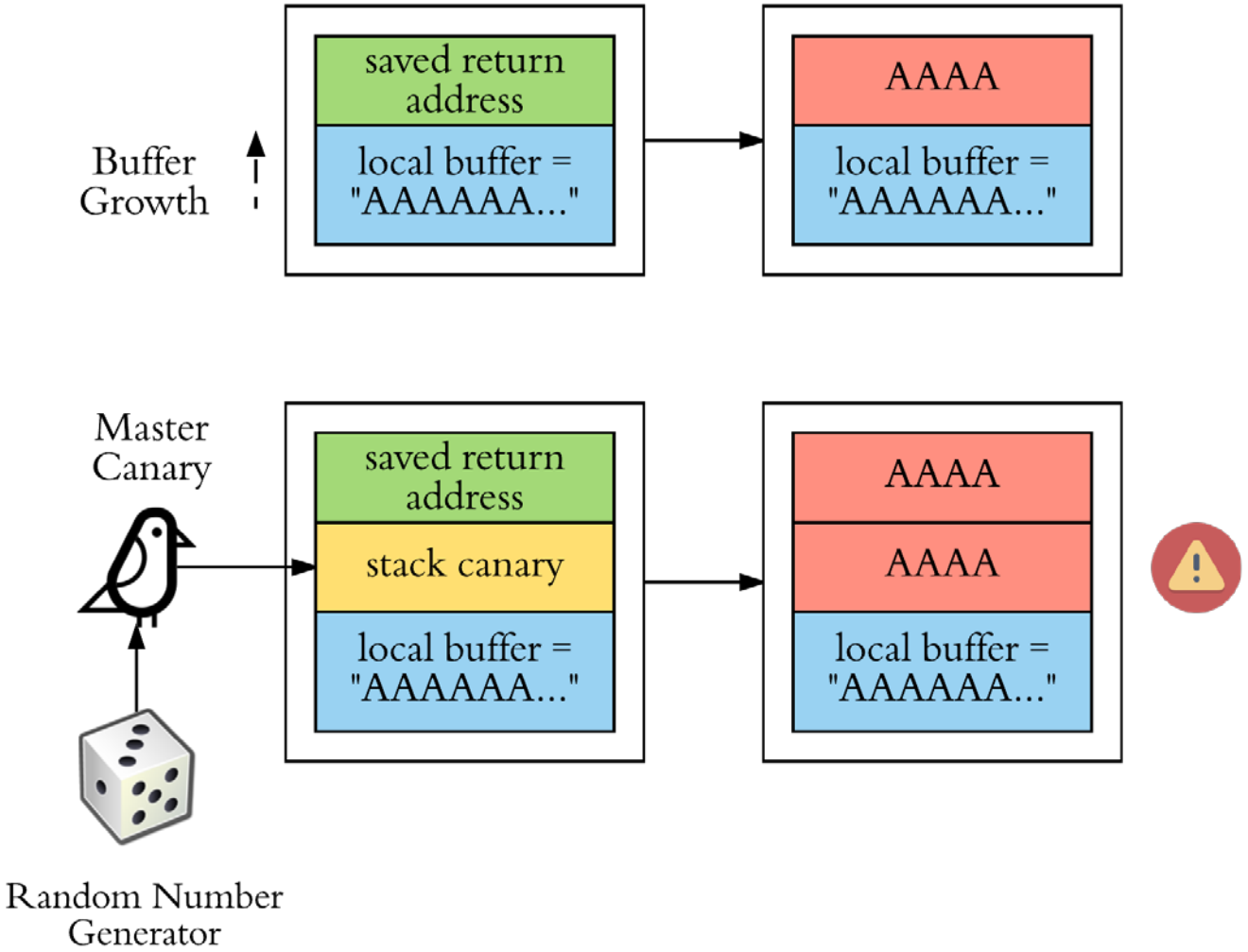


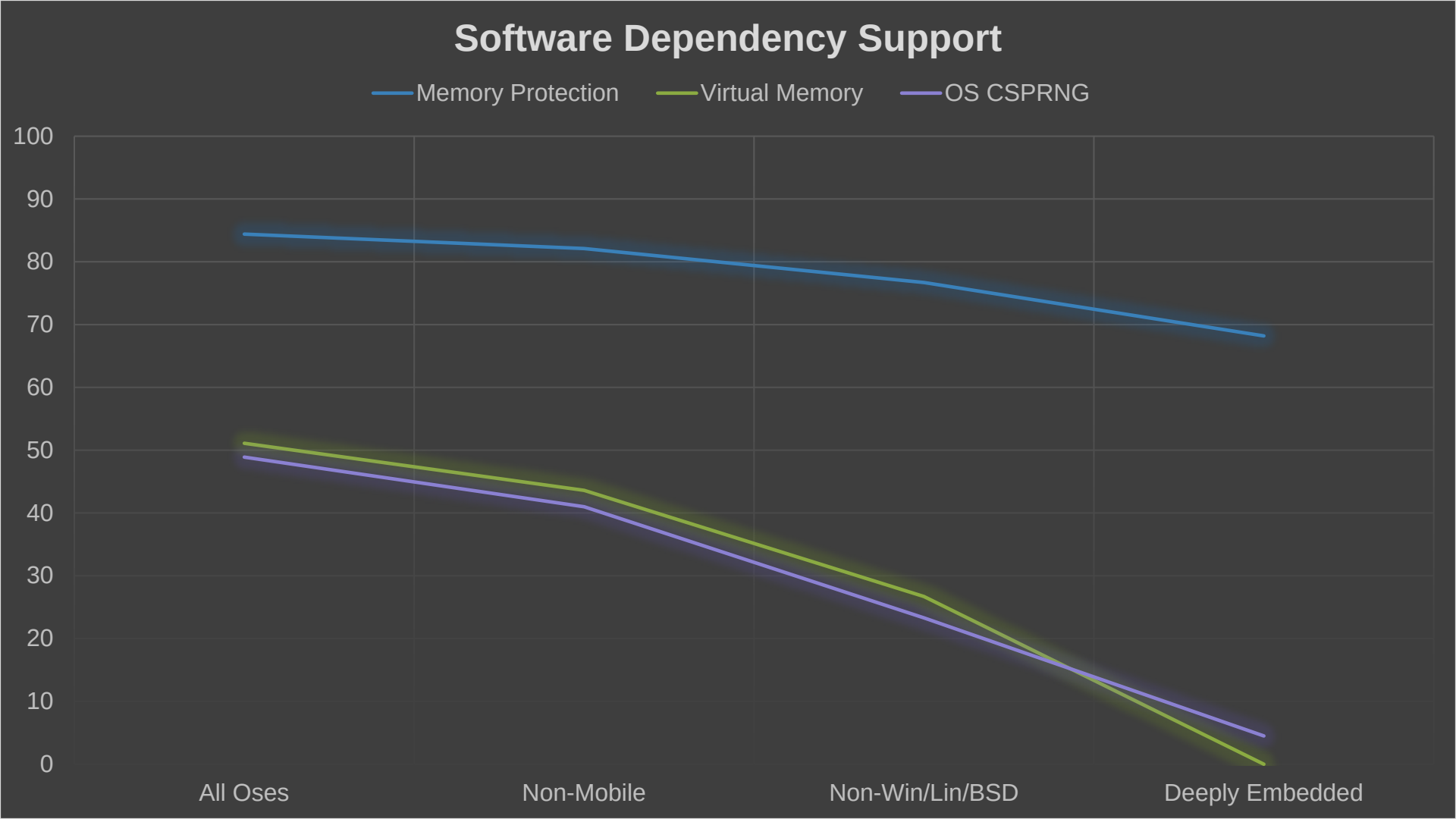





ASLR
And
Virtual Memory
OS CSPRNG
OS Features
MMU
Hardware Features

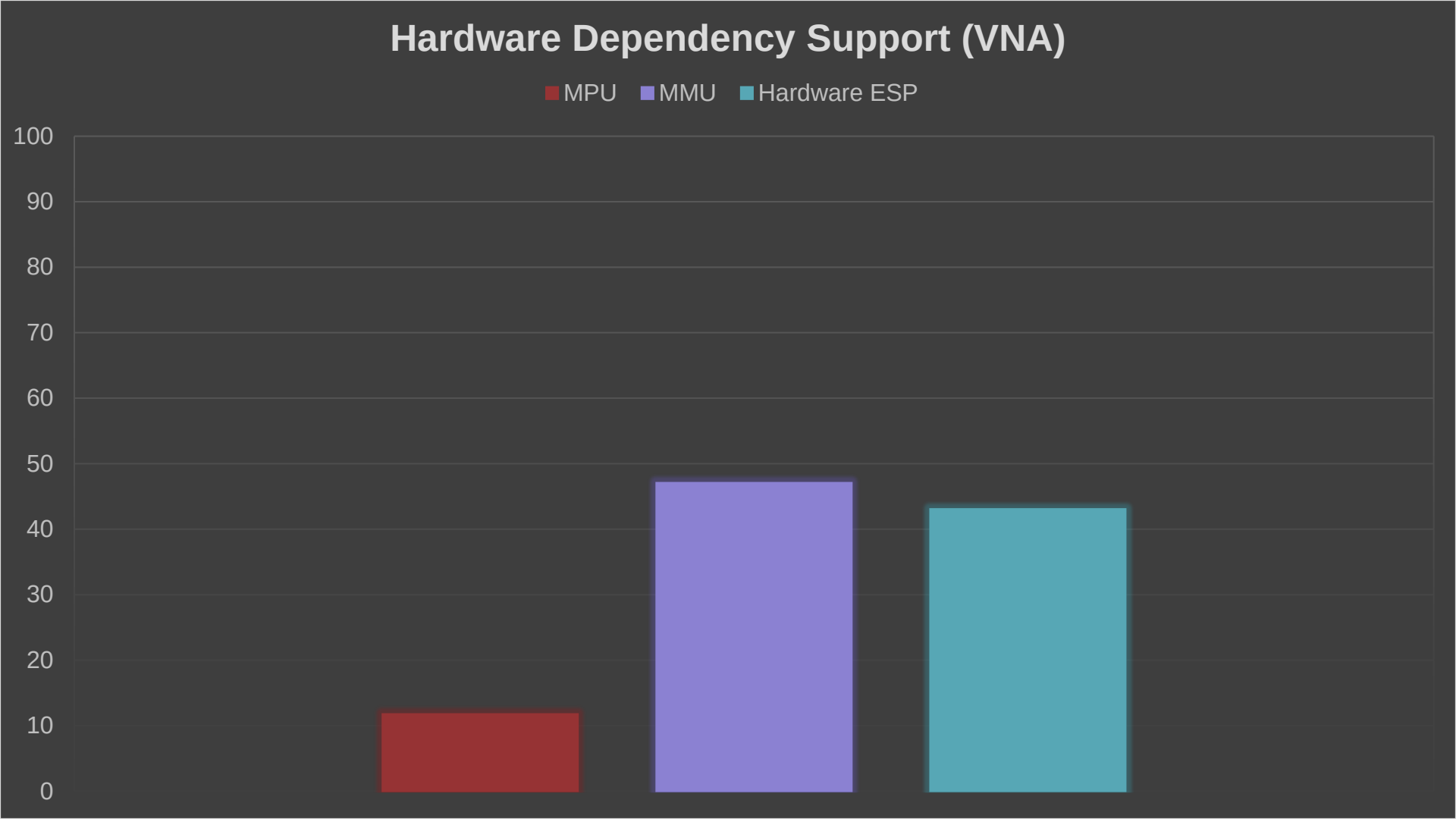

Memory
Physical Address
Translation Fault
MMU
OS
Virtual Address
CPU


Buffer Growth
saved return address
local buffer = "AAAAAA..."
AAAA
local buffer = "AAAAAA..."
Master Canary
saved return address
stack canary
local buffer = "AAAAAA..."
AAAA
AAAA
local buffer = "AAAAAA..."
Random Number Generator


Software Dependency Support
Memory Protection
Virtual Memory
OS CSPRNG
100
90
80
70
60
50
40
30
20
10
0
All Oses
Non-Mobile
Non-Win/Lin/BSD
Deeply Embedded

Hardware Selection

- Selected 78 Popular Embedded '*Core Families*'
- Evaluated for Hardware Dependency Support


Hardware Dependency Support (VNA)
MPU
MMU
Hardware ESP
100
90
80
70
60
50
40
30
20
10
0


Practical (I)IoT Examples

Schneider Electric

ARM

VxWorks

Typical (Entry-Level) PLC: Modicon Momentum Unity

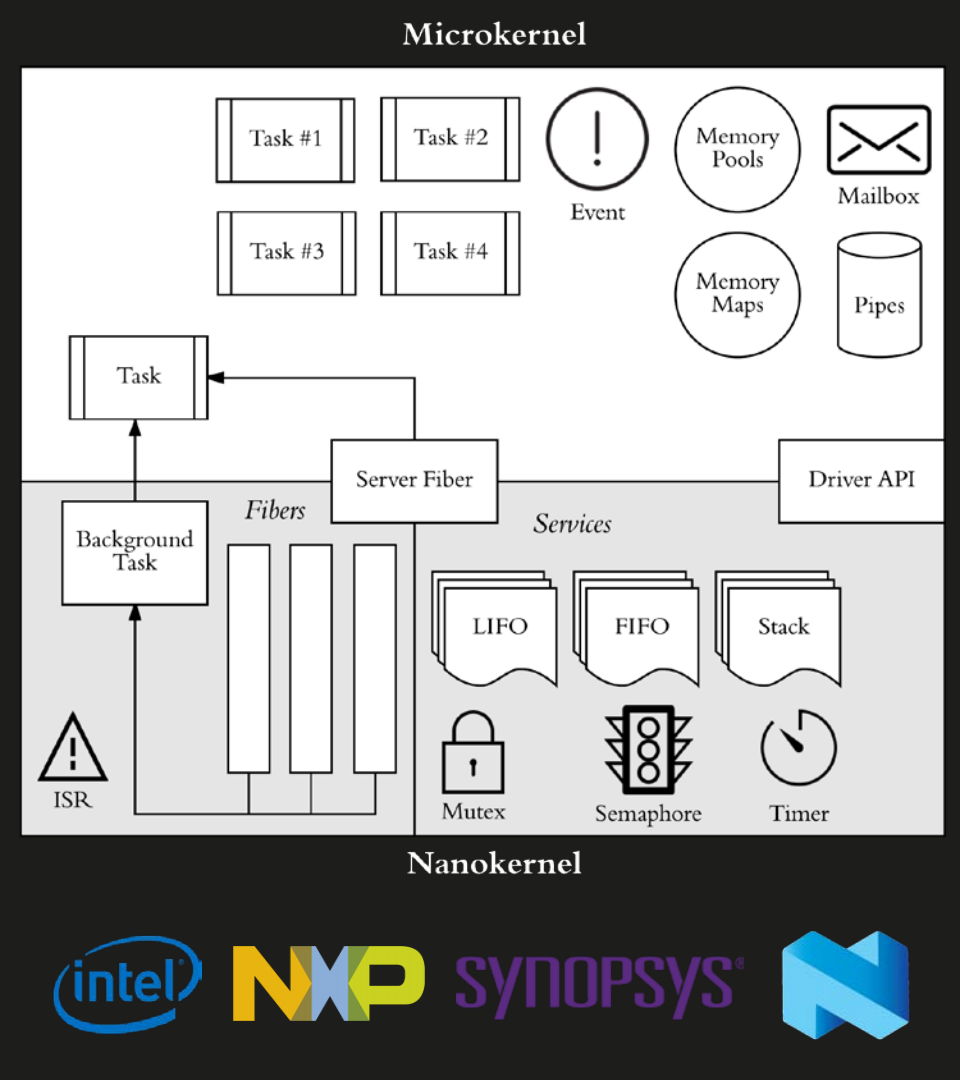
- **ST SPEAr 320s (ARM926EJ-S)**
  - Von Neumann
  - MMU (No XN Bit)
  - No TRNG
- **VxWorks RTOS**
  - No Mitigations
  - Memory Protection
  - Virtual Memory
  - No OS CSPRNG

ADVANTECH

Typical Wireless Sensor Node: Advantech WISE-1021

Texas Instruments

- **MSP430F5419A**
  - Von Neumann
  - No MPU / MMU
  - No TRNG
- **TI-RTOS**
  - No Mitigations
  - Memory Protection
  - No Virtual Memory
  - No OS CSPRNG



Zephyr™

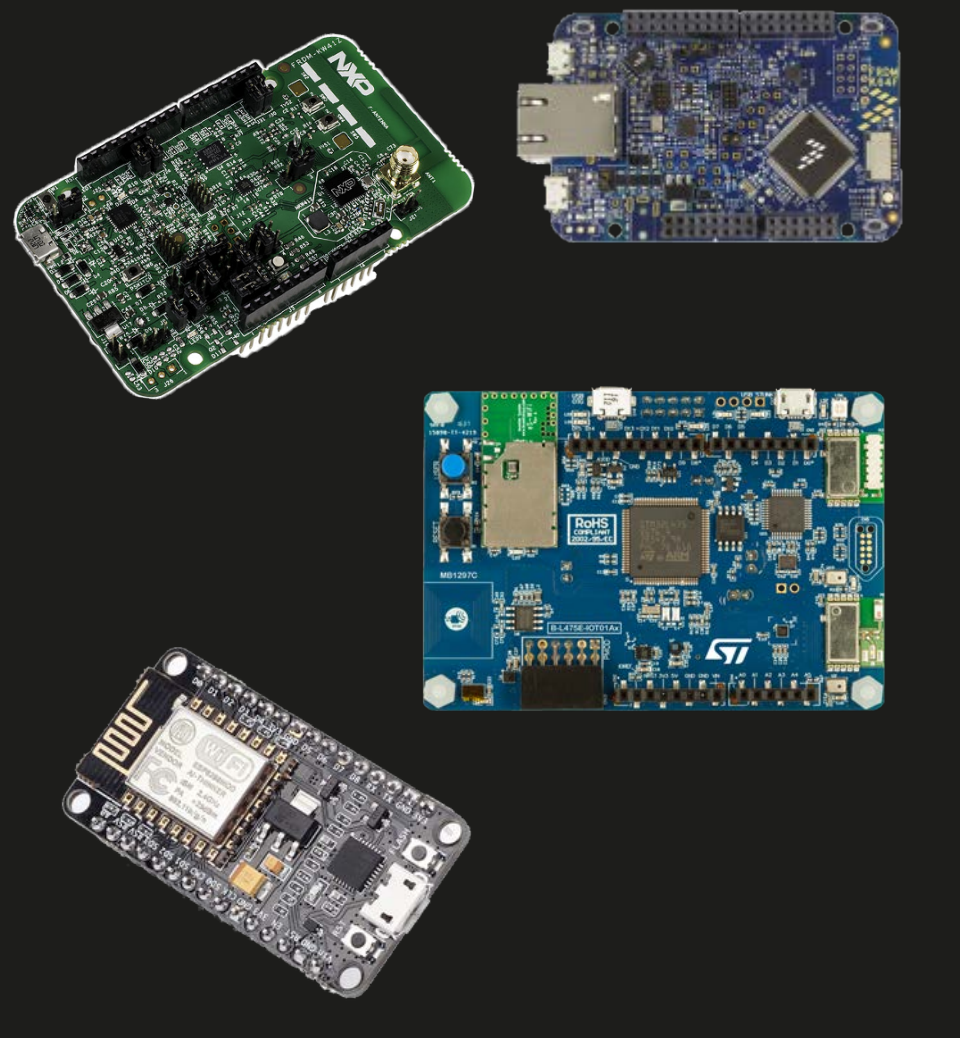
- **Library based RTOS**
  Based on Wind River Rocket
- **OS Linux Foundation Project**
  Aimed at resource-constrained IoT
- **Young (2016) but promising**
  Input from major chipmakers

Explicit focus on security

```
349 FUNC_NORETURN void _Cstart(void)
350 {
351 #ifdef CONFIG_ARCH_HAS_CUSTOM_SWAP_TO_MAIN
352         struct k_thread *dummy_thread = NULL;
353 #else
354         struct k_thread dummy_thread_memory;
355         struct k_thread *dummy_thread = &dummy_thread_memory;
356 #endif
357
358         /*
359          * Initialize kernel data structures. This step includes
360          * initializing the interrupt subsystem, which must be performed
361          * before the hardware initialization phase.
362          */
363
364         prepare_multithreading(dummy_thread);
365
366         /* perform basic hardware initialization */
367         _sys_device_do_config_level(_SYS_INIT_LEVEL_PRE_KERNEL_1);
368         _sys_device_do_config_level(_SYS_INIT_LEVEL_PRE_KERNEL_2);
369
370         /* initialize stack canaries */
371 #ifdef CONFIG_STACK_CANARIES
372         __stack_chk_guard = (void *)sys_rand32_get();
373 #endif
374
375         /* display boot banner */
376
377         switch_to_main_thread();
378
379         /*
380          * Compiler can't tell that the above routines won't return and issues
381          * a warning unless we explicitly tell it that control never gets this
382          * far.
383          */
384
385         CODE_UNREACHABLE;
386 }
```

Zephyr
Stack Canaries

- **Based on Clang/GCC SSP**
- **One master canary for entire address space**
  Generated once at system boot
- **Generated using RNG API**
  Implementation depends on chosen *random* driver

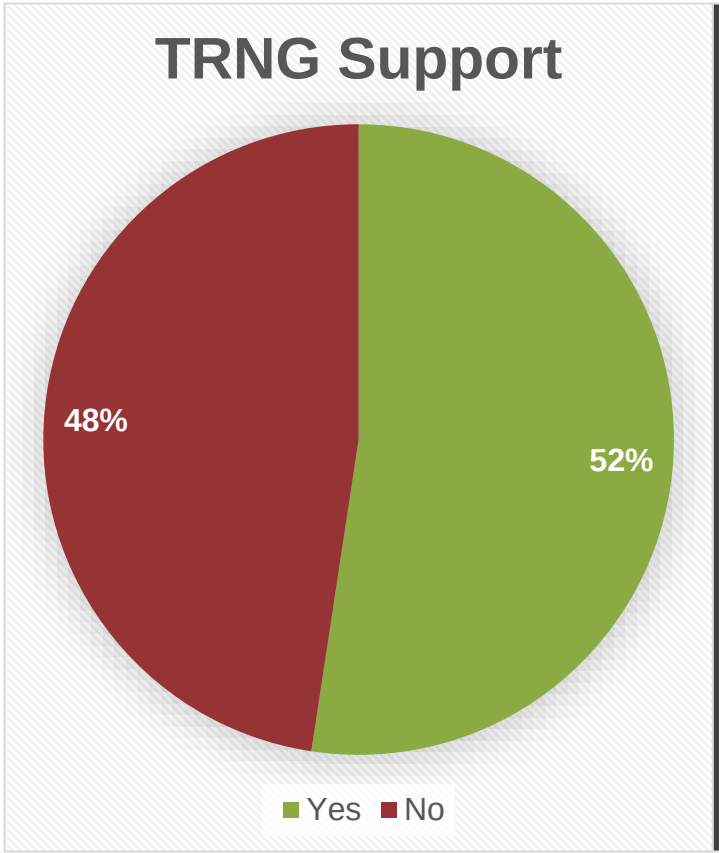


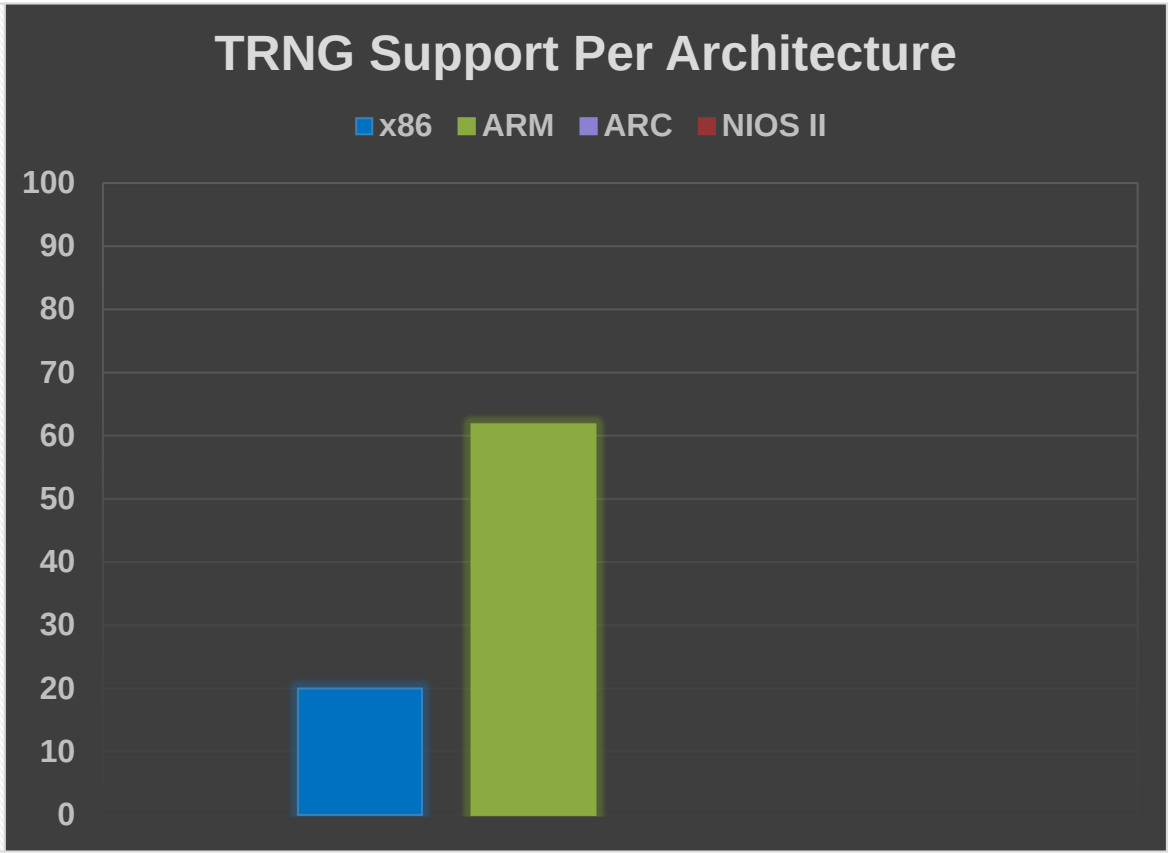
Zephyr Random API

RANDOM_HAS_DRIVER (TRNG)

- **RANDOM_MCUX_RNGA**
  NXP Kinetis K64F
- **RANDOM_MCUX_TRNG**
  NXP Kinetis KW40Z & KW41Z
- **RANDOM_STM32_RNG**
  STM32 Boards
- **RANDOM_ESP32_RNG**
  ESP32 Boards
  (requires Wi-Fi & Bluetooth enabled)

TRNGs Among Zephyr 1.8 Supported Boards





Zephyr Random API

TEST_RANDOM_DRIVER (PRNG)

- **X86_TSC_RANDOM_GENERATOR / TIMER_RANDOM_GENERATOR**
  Uses timestamp (eg. x86 RDTSC)

  Directly, not pulled through PRNG

  Canary 1st value drawn from API -> little difference between bootruns

  Infoleaks everywhere

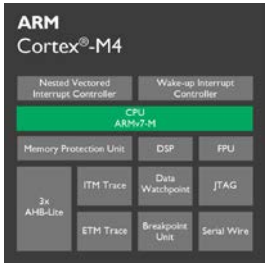
- **Had contact with Zephyr team, plans to integrate OS CSPRNG**


Defense

(Some) IoT Binary Security Tips

Harvard CPU

MCU with TRNG



VNA CPU
with MPU/MMU
& XN

Security-Oriented (RT)OS
(Warning: Still early days)

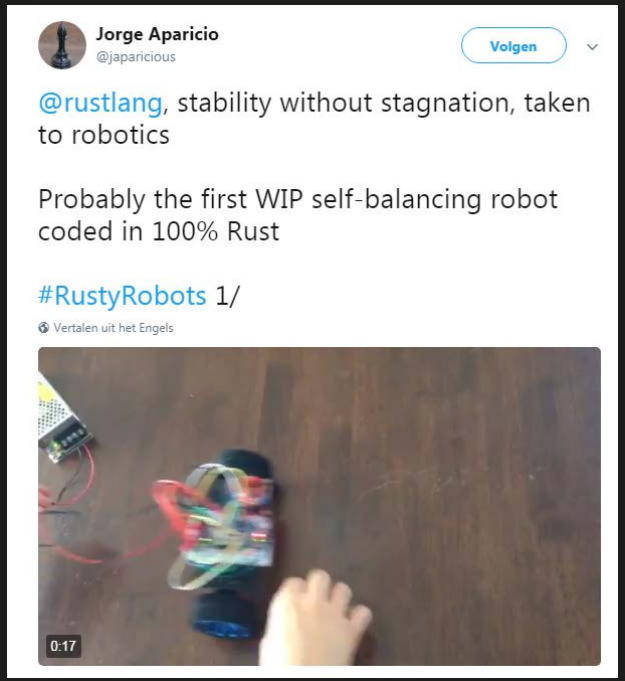
But: Reality Strikes (& Sucks)

- **Lifespan & Legacy**
  What we buy *today* was designed years ago

  What we design *today* might still be used in *10+ years*

- **Mitigations are a stop-gap**
  Need to start work on long-term now

Tock



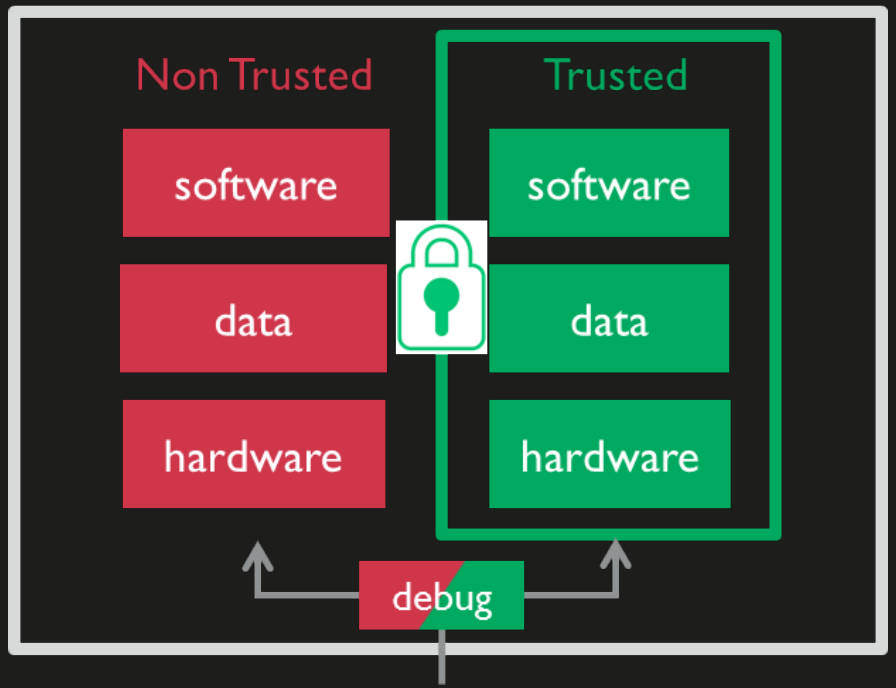
Safe Languages

- Can’t emphasize this enough
- Move to safe embedded development as soon as possible

* '*The Seven Properties of Highly Secure Devices*' – Galen Hunt, Microsoft, 2017

Renewable Security & Defense in Depth*

- Make your device *patchable,* ensure an *update infrastructure* is in place
- Layer your defenses, avoid single points of failure



* '*The Seven Properties of Highly Secure Devices*' – Galen Hunt, Microsoft, 2017

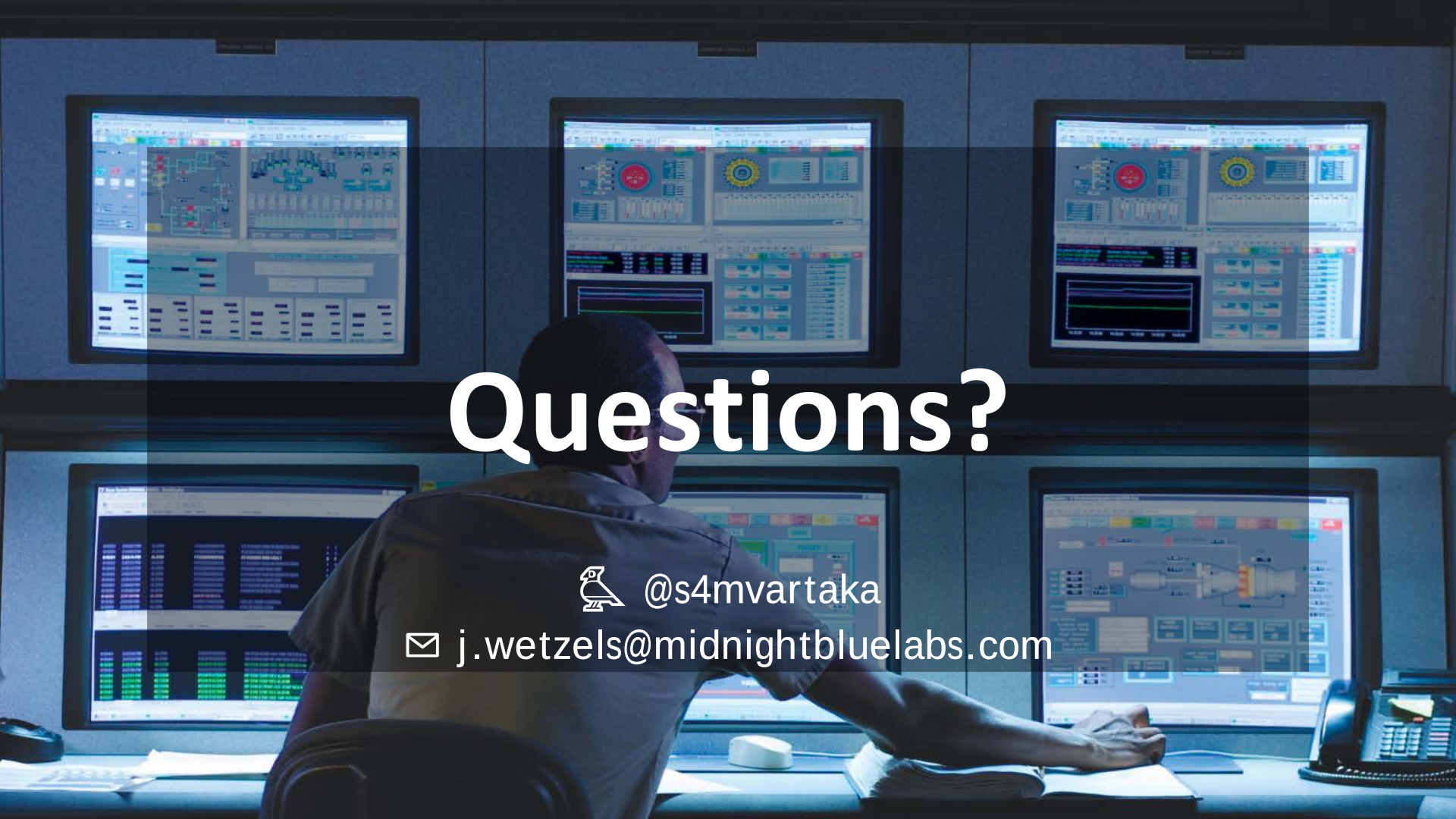
Compartmentalization & Small TCB*

- **Use hardware to enforce barriers between software components**
  (MMU, TrustZone)
- **Principle of Least Privilege**
  Keep your TCB Small

Failure Reporting*

- **If possible, ensure failures (eg. segfaults) are reported to cloud-based backend for later analysis**
- **Windows Error Reporting**
  The Inside Story Behind MS08-067, John Lambert, 2015

* '*The Seven Properties of Highly Secure Devices*' – Galen Hunt, Microsoft, 2017


Questions?
@s4mvartaka
j.wetzels@midnightbluelabs.com