UNIVERSITÄT PASSAU

Fakultät für Informatik und Mathematik

6090: Security of Computer and Embedded Systems

Week 1: Introduction; Computer and Embedded Systems; Their Need for Security

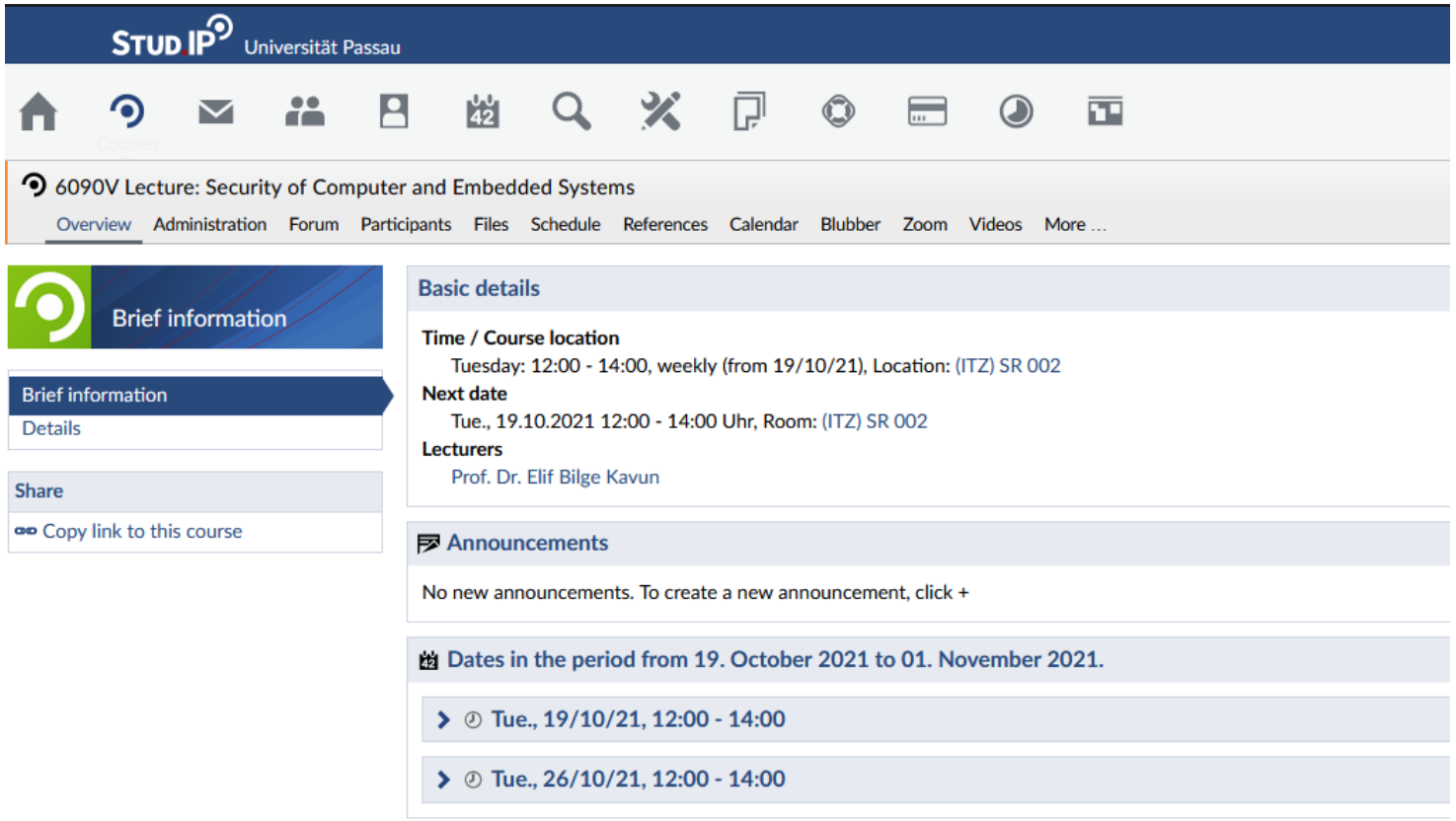
Elif Bilge Kavun

elif.kavun@uni-passau.de

October 19, 2021

- Stud.IP

Main communication point for the course!



Lecture Organization

- *Week 1:* Introduction; Computer and Embedded Systems; Their Need for Security
- *Week 2:* Computer and Embedded Systems Security Fundamentals; Access Control
- *Week 3:* Secure Software Development Lifecycle (SSDL); Threat Modelling
- *Week 4:* Software Vulnerabilities; Common Vulnerability Scoring System (CVSS); Secure Programming
- *Week 5:* Security Testing: Overview, Fuzzing, Static Analysis; Security of Third-Party Components
- *Week 6:* Cryptographic Foundations 1
- *Week 7:* Cryptographic Foundations 2
- *Week 8:* Public Key Infrastructures (PKIs)
- *Week 9:* Security Protocols; Network Security (Protocols); Formal Analysis of Security Protocols
- *Week 10:* Free
- 2 Weeks Xmas Break
- *Week 11:* Attacking Crypto and Protocols (theoretical and practical); Evaluation Methods and Tools
- *Week 12:* RFID Security; IC Security (Hardware Fingerprinting)
- *Week 13:* Flexible Electronics and Their Security; Other Novel Applications and Their Security
- *Week 14:* Free
- *Week 15:* Overview of Topics; Repetition before Exam

Course Schedule
(also available on Stud.IP)

- Tuesdays (6090V)
  - Lectures
    - Presenting new content, i.e., extending your knowledge
    - Slides will be available before/after the lecture
  - 12:00-14:00, (ITZ) SR 002 [Zoom stream in parallel]
  - Will be recorded and later shared on Stud.IP

- Tuesdays (6090UE)
  - Tutorials/Exercise sessions (covering problem sheets)
    - Deepen your knowledge by discussing problems from the problem sheets
  - 14:00-15:00 , (ITZ) SR 002 [Zoom stream in parallel]
  - Will be recorded and later shared on Stud.IP
  - Problem sheets generally available before the lecture on Stud.IP
    - Not assessed
    - Study basis for the final assessment
    - Solutions available after the lecture

Problem Sheets

- Problem sheets serve several purposes
  - Extend and deepen your knowledge of the subject
  - Provide detailed references (reading list) on a per-chapter level (mostly referring to freely available material/books)
  - Help you to catch up on preliminaries
- Good for repetition
  - Exercises discuss also preliminaries
  - You might know these already – if not, these exercises will help you to catch up
  - They usually contain practical exercises that you can do by hand or on your computer

Assessment

- Formal Written / Oral Exam
  - 60min written (or 20min oral, if required due to pandemic)
  - Open questions
    - You need to show that you can apply your knowledge to new problems

Contact for Questions

- During the lecture/tutorials
  - Raise your hand or write in chat box during the lecture
- During the week
  - Please use *Stud.IP Forum* for questions!
    - So that your classmates can also learn from the discussions
    - Check the forum first, the question might already be answered
    - If not, post a new question
    - I will check at least every other day
- If you feel your question is personal/confidential
  - Send an email if the question is NOT of general interest

Personal Background

- BScEE degree and MSc degree in Cryptography from Turkey
- PhD degree in Embedded Security from Ruhr University Bochum, Germany
  - Thesis on “Resource-constrained (lightweight) cryptography”
- Six years of industry and consultancy experience
  - Internship and consultancy for hardware security projects of different companies (Turkey, USA)
  - Staff Engineer – Crypto Cores Design, Infineon Technologies (Germany)
    - Leading company in semiconductor sector
    - Responsible for design and evaluation of secure symmetric crypto primitives
  - Lecturer in Cybersecurity, The University of Sheffield (UK)
- Since 10/2020
  - Assistant Professor in Secure Intelligent Systems, University of Passau
- Main work areas/research interests
  - Hardware security
  - Design and implementation of cryptographic primitives
  - Lightweight cryptography
  - Side-channel attacks and countermeasures
  - Security of intelligent systems
  - AI for security applications

Course Content

- You will need to work with
  - Set theory and logic
    - $x = \{x \mid x \in Y \ \wedge \ \exists z > x \cdot z \bmod 3 = 0\}$
    - $x \oplus y = x \ \wedge \ \neg \mathrm{y} \ \vee \ \neg x \ \wedge \ \mathrm{y}$
  - Basic algebra
    - $a^{n+m} = a^n \cdot a^m$
- Solid knowledge of at least one programming language and/or program pseudocode
  - Python, C, etc.

Examples of the Most Important Preliminaries

- You should already
  - know and be able to apply the algebraic properties of exponentiation
    - $a^{n+m} = a^n \cdot a^m$
  - know and be able to apply the algebraic properties of modular arithmetic
    - $(a \cdot b \cdot c) \bmod n = \Big(\big((a \cdot b) \bmod n\big) \cdot c\Big) \bmod n$
  - use the algebraic properties of exponentiation and modular arithmetic to compute $39^{17} \bmod 11$ using a standard calculator
  - know and be able to apply the algebraic properties of Boolean algebra
    - $\big((a \oplus b) \oplus b\big) = a$, where $a \oplus b = a \wedge \neg b \vee \neg a \wedge b$
  - know basic set notation
    - $x \in X$ (set membership)
    - $\{x | x > 5\}$ (set comprehension)

How Can We Build "Secure" Computer and Embedded Systems?

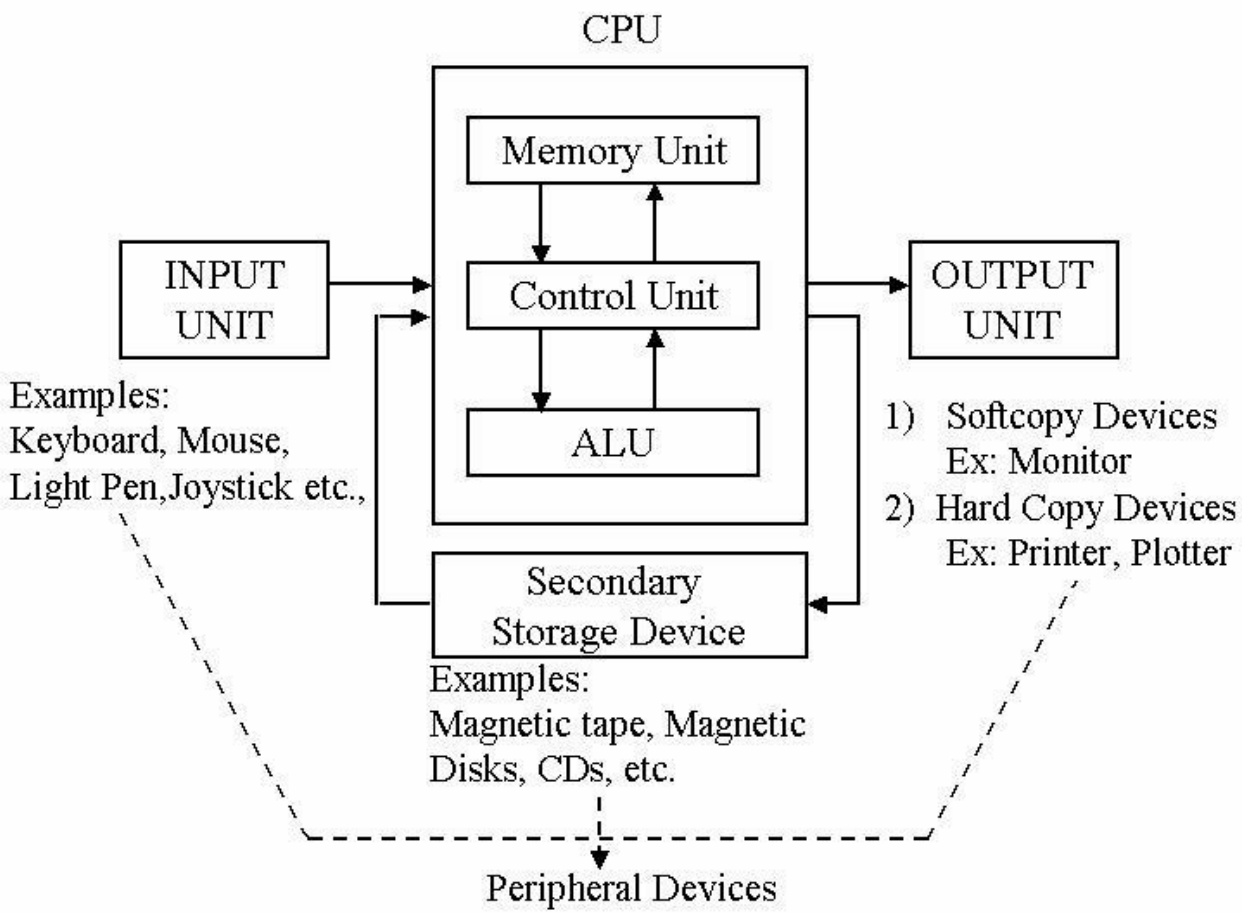
- In this course you will
  - Develop a general understanding of computer and embedded systems' security
  - Learn various security technologies
  - Learn the nature of security problems in such systems
  - Learn the challenges managing and discussing security issues
  - Learn how to develop secure systems (defensive)
  - Learn how to test the security of systems (offensive)
  - Develop a general understanding of information/computer security

“What is a Computer System” ?

“What is a Embedded System” ?

Let’s brainstorm…

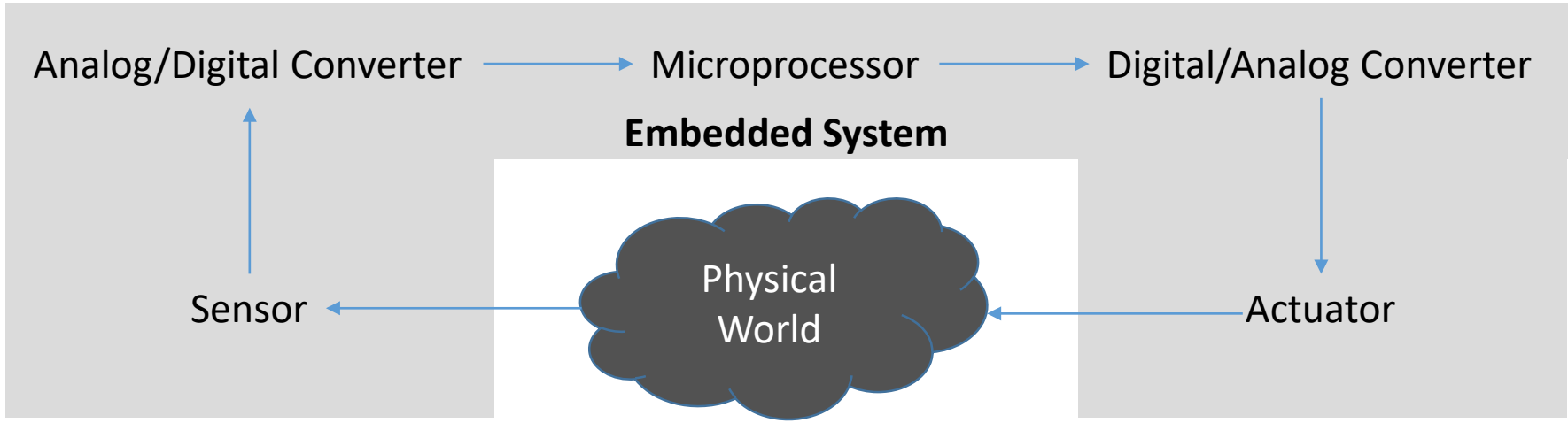
What Is A Computer System?



Source: peda.net

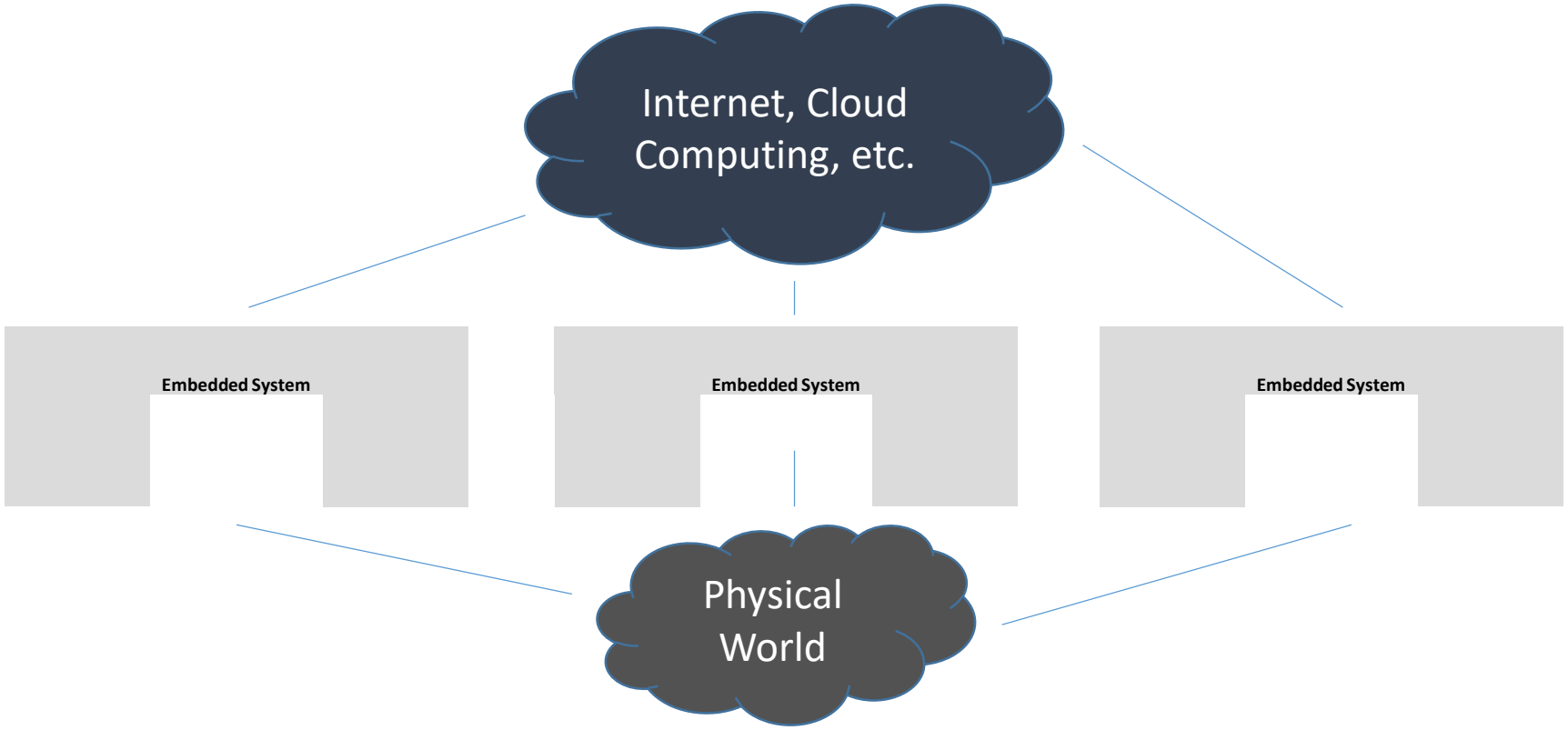
- A set of integrated devices that input, output, process, and store data and information
- Built around at least one digital processing device
- Five main hardware components
  - Input, Processing, Storage, Output, and Communication devices.

What Is An Embedded System?



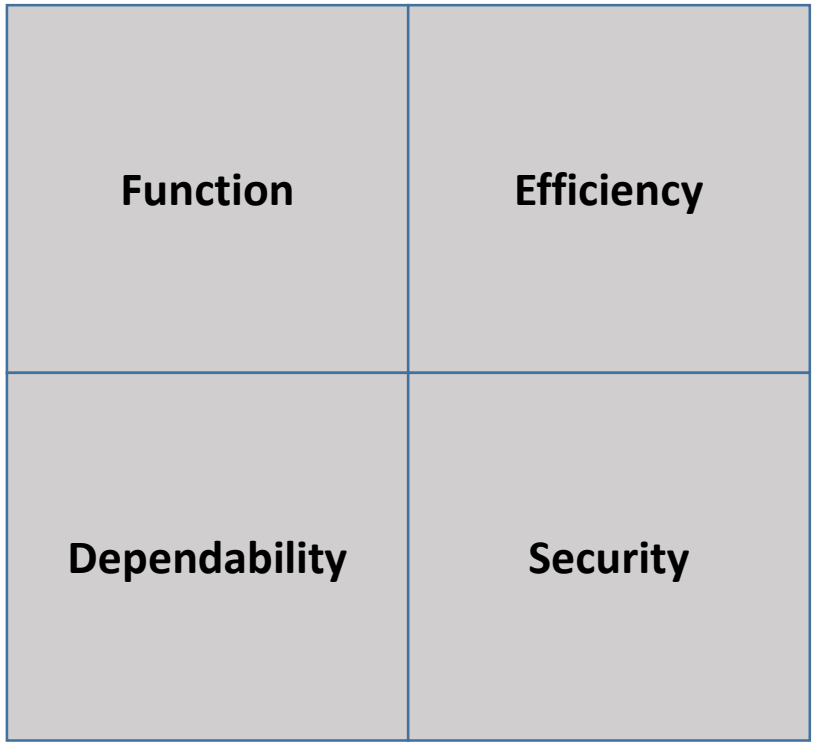
- Software/Firmware
  - Specific application-specific hardware-related software
  - Operating systems with special applications
- Hardware
  - Integrated ASICs (Application-specific Integrated Circuits)
- Systems that process information which are embedded into a larger product

Cyber-physical System (CPS)

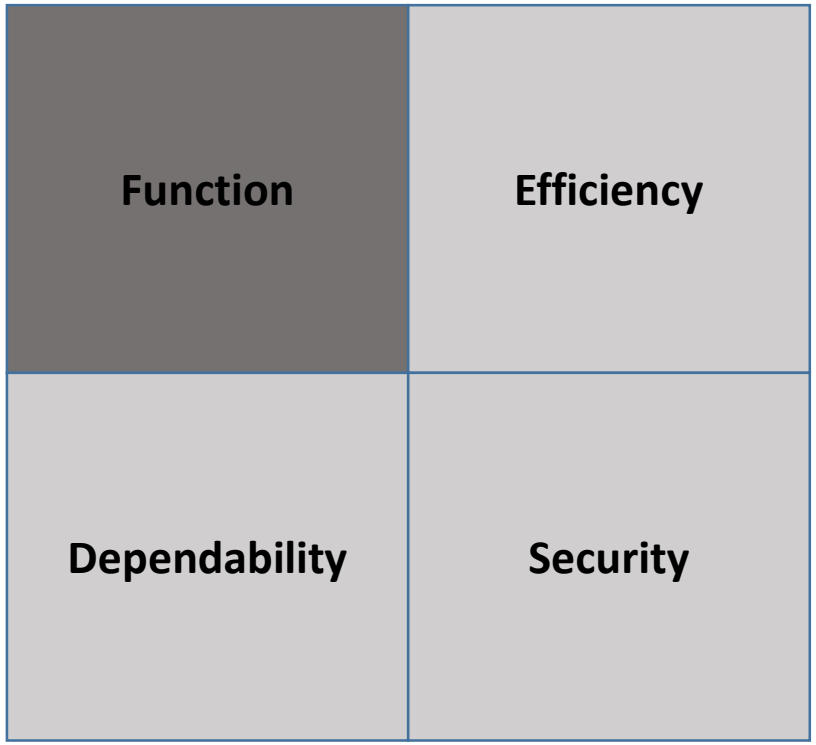


- Integration of internet
  - Communicating embedded systems
- Integration of computation with physical processes
  - *CPS = Embedded System + Physical environment + Internet*

Embedded System Requirements

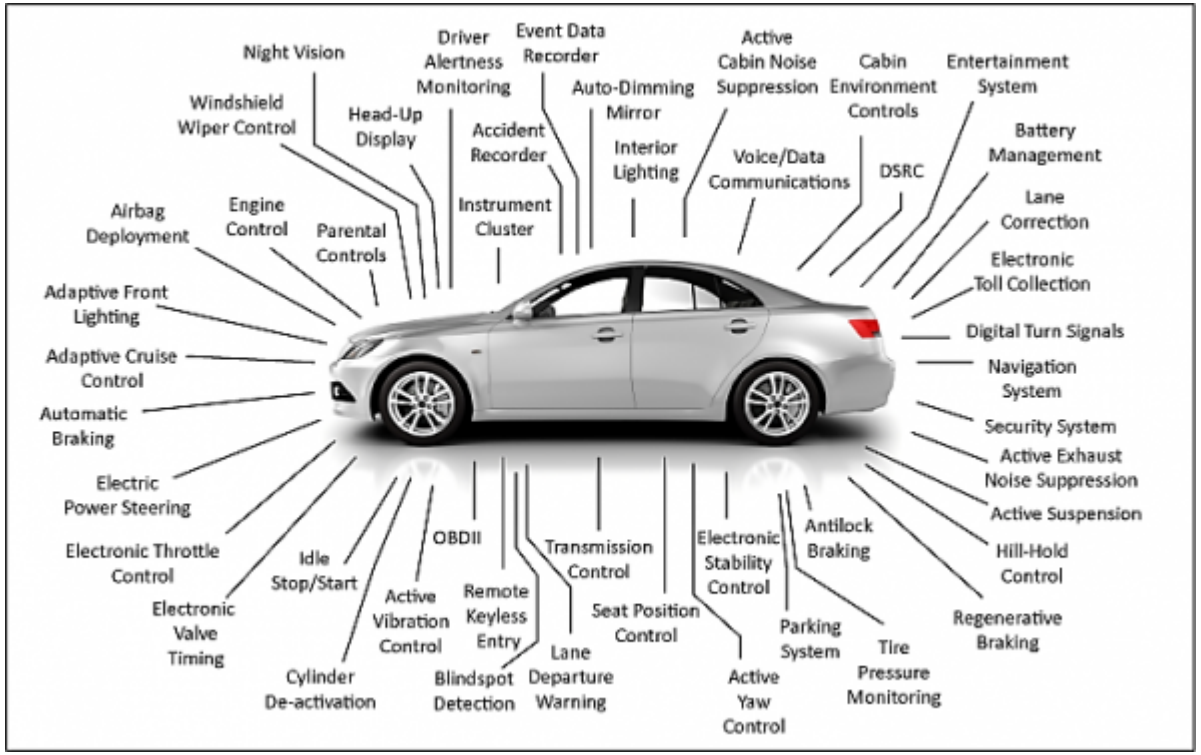


Embedded System Requirements



Automotive

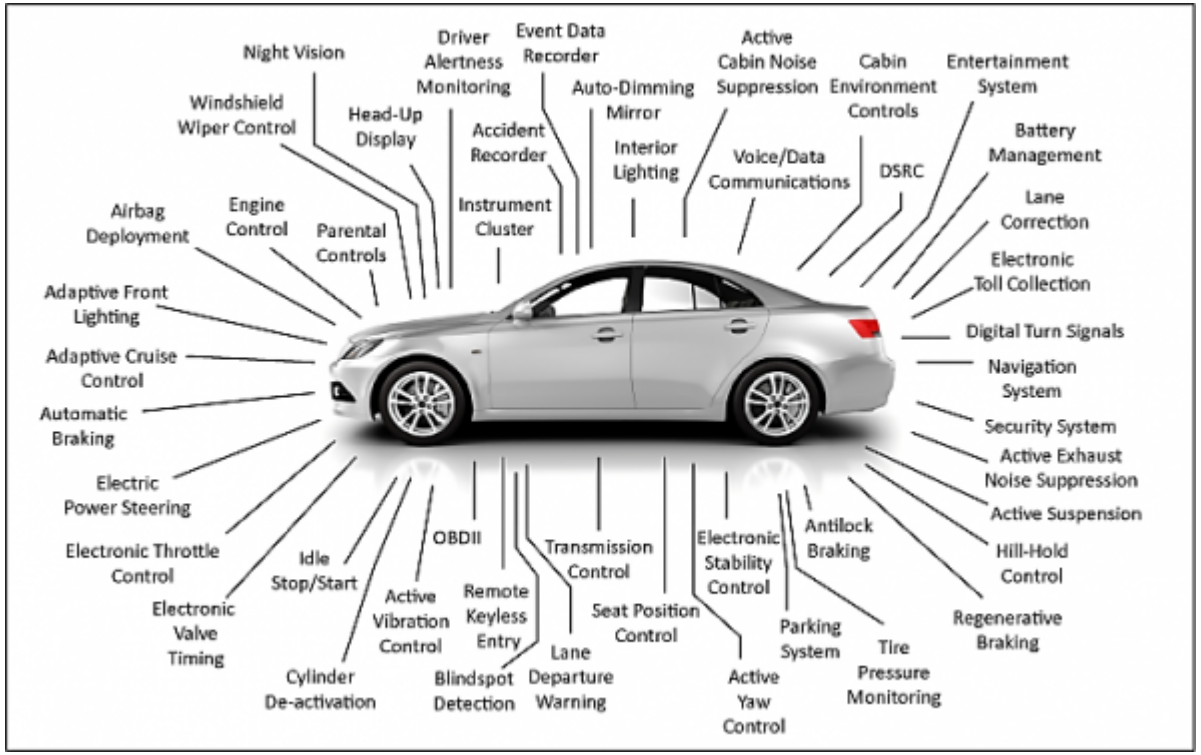
- Modern cars have many microcontrollers
  - Around 60-70



*Picture taken from:
https://www.smart2zero.com/design-center/automotive-service-era-electronic-car-0

Automotive

- Modern cars have many microcontrollers
  - Around 60-70
- Applications
  - Engine control
  - ABS (Anti-lock braking systems)
  - ESP (Electronic stability control)
  - Automatic gearboxes
  - Airbags
  - Theft prevention
  - Smart keys
  - Blind-angle/drive alert systems
  - Battery charging for electric/plug-in hybrid cars
  - Autonomous drive
  - Comfort systems



*Picture taken from:
https://www.smart2zero.com/design-center/automotive-service-era-electronic-car-0

Industrial Automation

- Industry 4.0 applications
- Smart factories
- Smart grid
  - Smart meters
  - Smart appliances

*Picture taken from:
https://www.industr.com/en/new-industrial-automation-system-topologies-accomplished-by-iiot-2385176

Industrial Automation

- Industry 4.0 applications
- Smart factories
- Smart grid
  - Smart meters
  - Smart appliances
- Applications
  - Robots
  - PLC (Programmable Logic Controller)
  - Logistics: Tracking systems via RFID Tags

*Picture taken from:
https://www.industr.com/en/new-industrial-automation-system-topologies-accomplished-by-iiot-2385176

Communication Systems

*Pictures taken from:

https://en.wikipedia.org/wiki/Mobile_phone
https://ethw.org/Cellular_Base_Stations
https://en.wikipedia.org/wiki/SIM_card
https://www.govtech.com/blogs/lohrmann-on-cybersecurity/the-biggest-problem-with-smartphones.html

Communication Systems

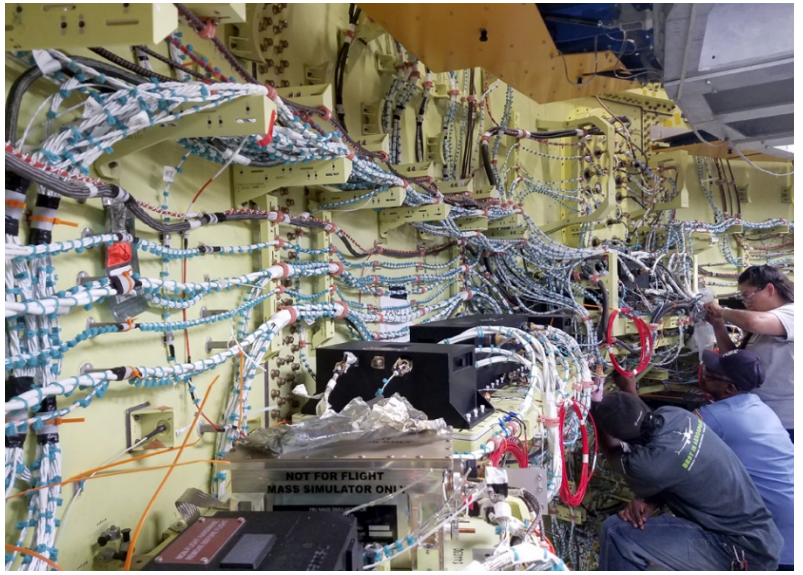
- Applications
  - Mobile phones
  - Smartphones
  - SIM cards
  - Routers
  - Base stations

*Pictures taken from:

https://en.wikipedia.org/wiki/Mobile_phone
https://ethw.org/Cellular_Base_Stations
https://en.wikipedia.org/wiki/SIM_card
https://www.govtech.com/blogs/lohrmann-on-cybersecurity/the-biggest-problem-with-smartphones.html

Avionics

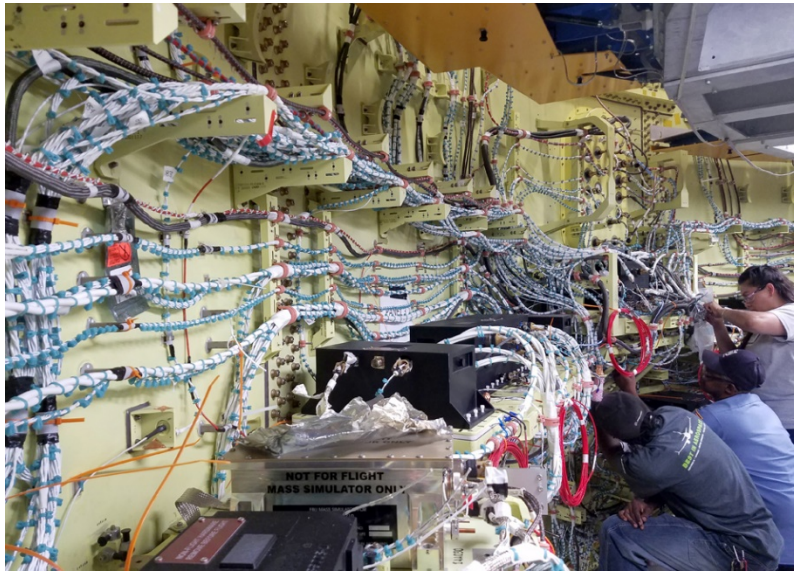
- Airplanes have a huge amount of electronics & embedded systems
  - Avionics = Aviation + Electronics



*Pictures taken from:
https://glance-efis.com/electronic-flight-instrument-system-overview/
http://spaceref.com/sls/a-closer-look-at-sls-avionics.html

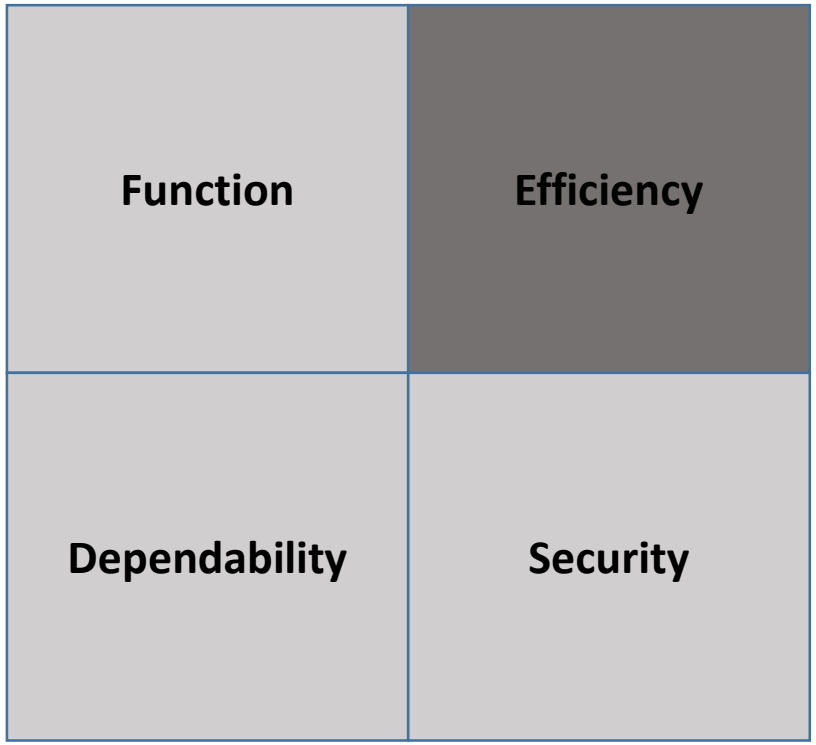
Avionics

- Airplanes have a huge amount of electronics & embedded systems
  - Avionics = Aviation + Electronics
- Applications
  - Flight control
  - Anti-collision
  - Pilot information
  - Flap control
  - Power supply
  - Entertainment systems



*Pictures taken from:
https://glance-efis.com/electronic-flight-instrument-system-overview/
http://spaceref.com/sls/a-closer-look-at-sls-avionics.html

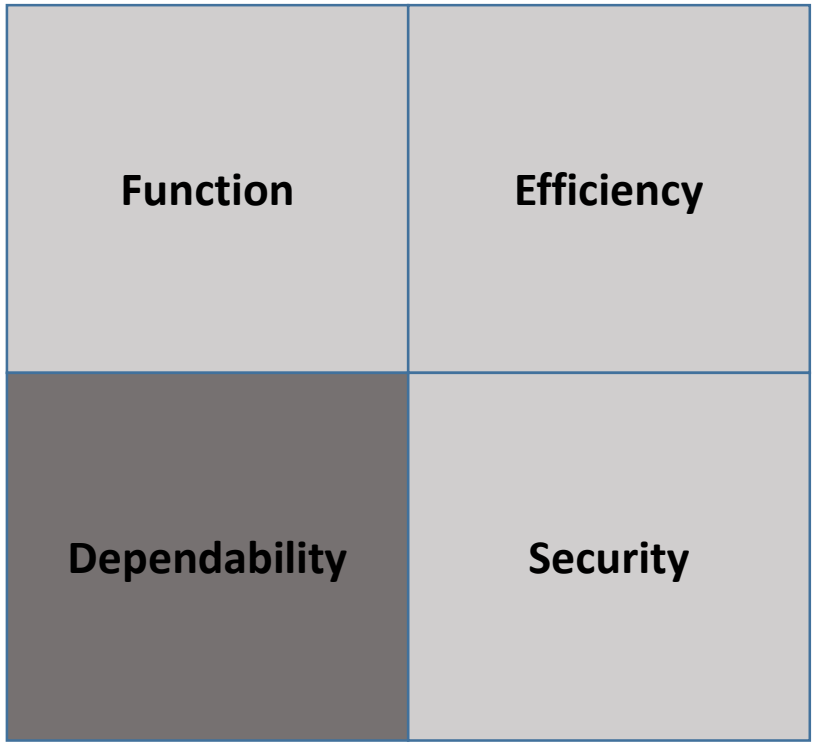
Embedded System Requirements



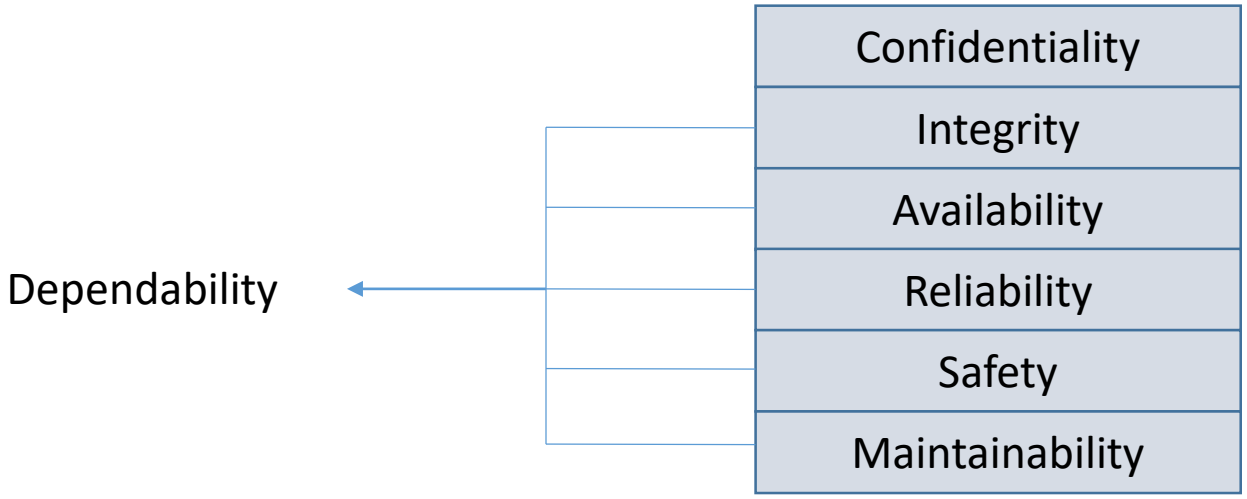
Efficiency

- Run-time & Performance
- Code-size
- Silicon area
- Cost
- Power & Energy
- Weight & Space

Embedded System Requirements



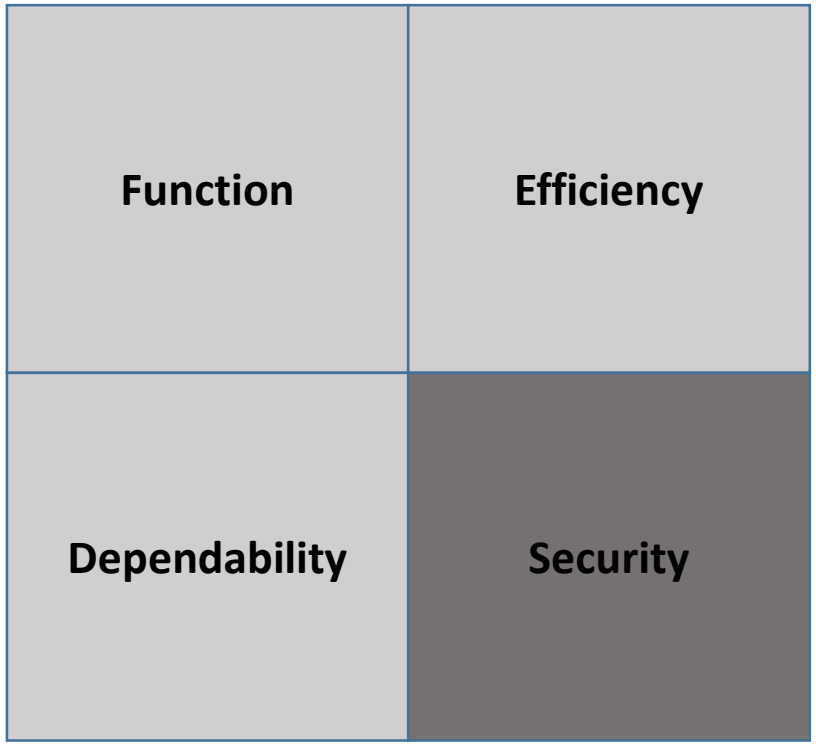
Dependability



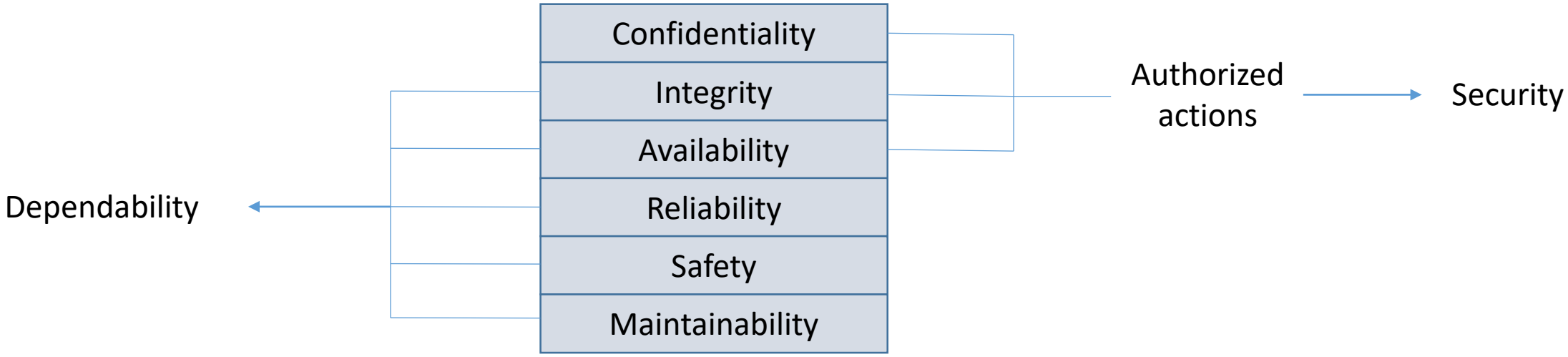
- Important for avionics!

***Source:** Avizzienis et al.; Basic Concepts and Taxonomy of Dependable and Secure Computing; IEEE Transactions on Dependable and Secure Computing; Vol. 1, No. 1; 2004.

Embedded System Requirements

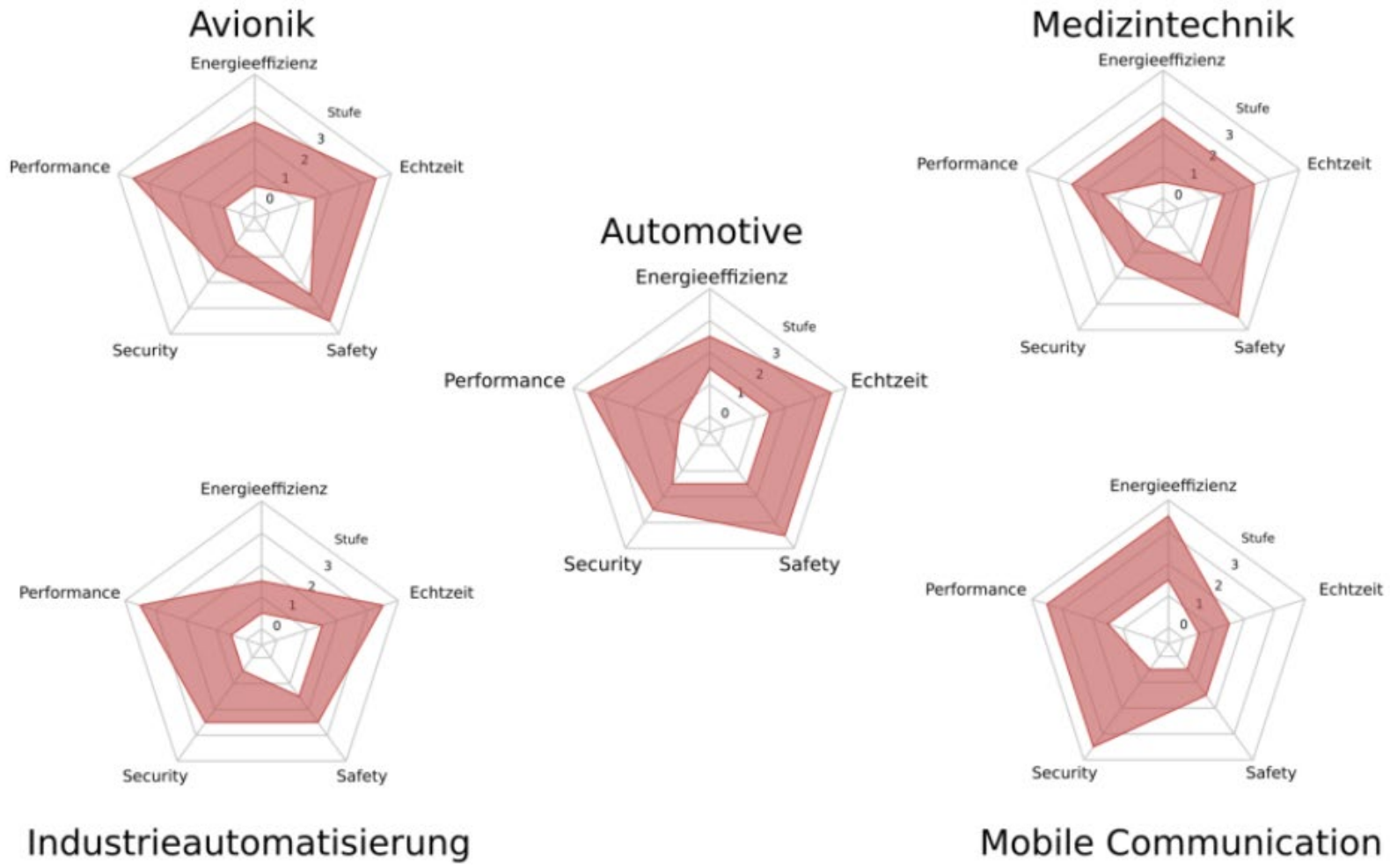


Relationship between Dependability and Security



***Source:** Avizzienis et al.; Basic Concepts and Taxonomy of Dependable and Secure Computing; IEEE Transactions on Dependable and Secure Computing; Vol. 1, No. 1; 2004.

Importance of Requirements in Different Application Domains

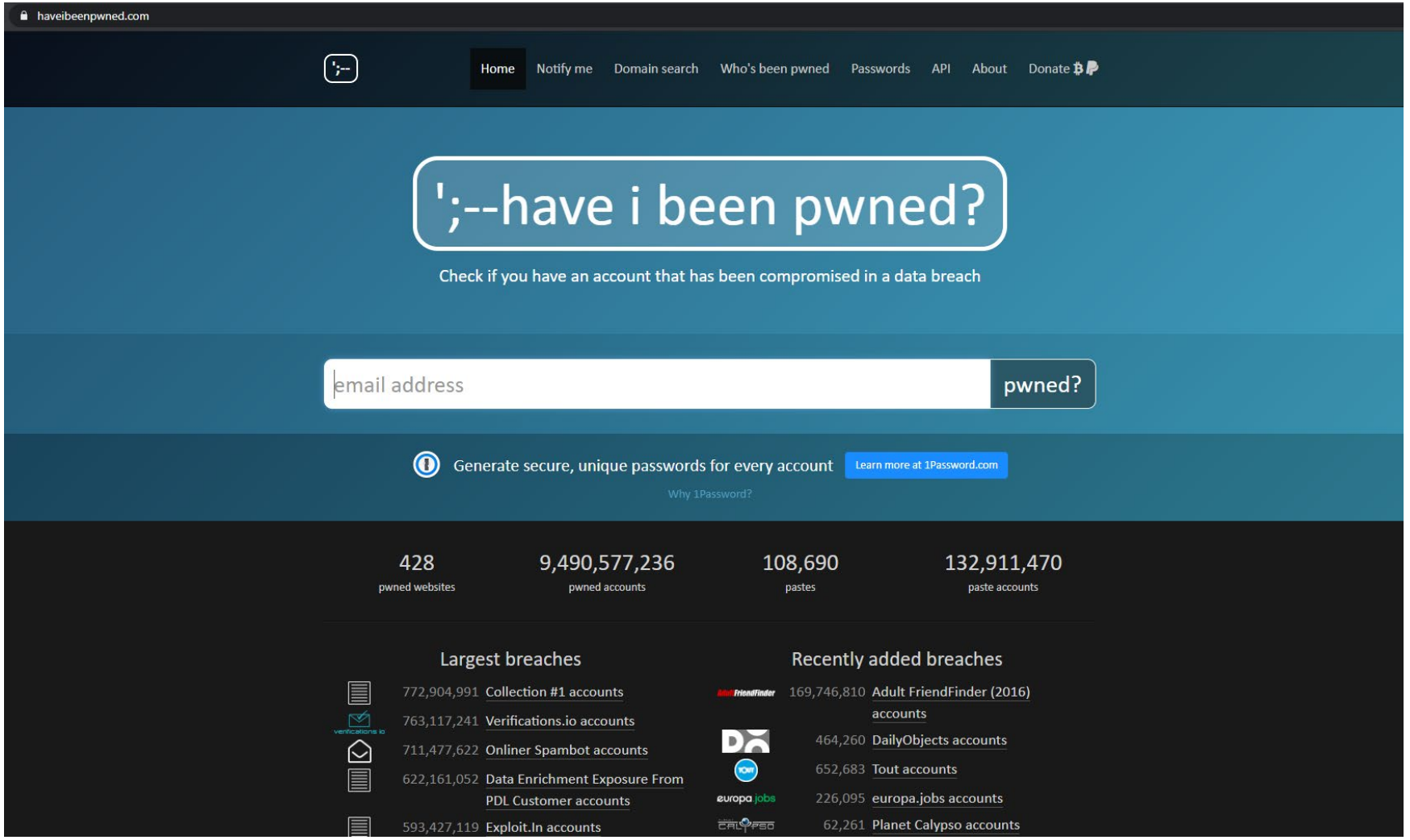


***Source:** Herkersdorf et al.; Potentials and Challenges for Multi-Core Processors in Robotic Applications; Workshop “Robotor-Kontrollarchitekturen” INFORMATIK13; 2013.

Why does Security of Computer and Embedded Systems get more and more important?

Computer Security

(https://haveibeenpwned.com/)



Computer Security

Information Leak Examples

- LinkedIn, May 2016
  - 164 million email addresses and passwords
  - From an attack in 2012, offered for sale May 2016
  - Compromised data
    - Email addresses
    - Passwords
  - Again in June 2021: 700 million users
- Facebook, April 2019
  - 533 million users
  - Phone numbers, account names, FB IDs revealed

Information Leak Examples

- Yahoo, August 2013 and also 2014
  - 3 billion accounts
- Alibaba, November 2019
  - 1,1 billion pieces of user data
- Marriott International, September 2018
  - 500 million customers
  - Guests' names, mailing addresses, phone numbers, email addresses, passport numbers, guest account information, dates of birth, gender, arrival and departure information, reservation dates, and communication preferences
  - For some, the information also included payment card numbers and expiration dates (though encrypted)
- My Fitness Pal, February 2018
  - 150 million user accounts

Information Leak Examples

- Ashley Madison, July 2015
  - More than 30 million email addresses and much more
  - Compromised data
    - Dates of birth
    - Email addresses
    - Ethnicities
    - Genders
    - Sexual preferences
    - Home addresses
    - Phone numbers
    - Payment histories
    - Passwords, Usernames, security questions and answers
    - Website activity
  - Similar Leak: Mate1 in February 2016
    - 27 million records with even more personal details (e.g., drinking/drug habits, parenting plans, political views)

Costs of Data Breaches

- *“A hack not only costs a company money, but also its reputation and the trust of its customers. It can take years and millions of dollars to repair the damage that a single computer hack inflicts.”*

  (http://financialedge.investopedia.com/financial-edge/0711/Most-Costly-Computer-Hacks-Of-All-Time.aspx)

- TJX Company, Inc. (2007): $250 million
- Sony (2011): $170 million
- Marriott (2018): £18.4 million (fined)
- Heartland Payment Systems (2009): $41 million

- Note:
  - Publicly known incidents are usually "Business-to-Customer (B2C)"
  - Business-to-Business (B2B) incidents are often not publicly known

Importance of Embedded Security

- “Formerly” isolated systems can be attacked via network
  - Industry automation systems can be manipulated: See “Stuxnet”
  - Smart metering systems can be manipulated
  - Cars can be attacked via network interface
  - Health control systems can be intercepted
  - Relation with “Safety”: External attackers have access to safety-critical parts
- The background system can be attacked via a hacked embedded system
  - Smart metering systems can be used to get server access
  - Cars may communicate wrong data in a Car2X environment

What’s the Problem?

Authenticate without a password using "SQL Injection"

“Hostile data is sent to an interpreter as command or query to trick the interpreter into executing unintended commands or accessing data without proper authorization.”

In a database, **users_list** table

id	username
1	Alice
2	Bob
3	Eve

What’s the Problem?

Authenticate without a password using "SQL Injection"

username=INPUT

What’s the Problem?

Authenticate without a password using "SQL Injection"

username=**Alice**

What's the Problem?

Authenticate without a password using "SQL Injection"

username=**Alice**

id	username	
1	Alice	**TRUE**
2	Bob	**FALSE**
3	Eve	**FALSE**

1, Alice

What’s the Problem?

Authenticate without a password using "SQL Injection"

- Now try apostrophe as input

```
username='
```

What’s the Problem?

Authenticate without a password using "SQL Injection"

username='

Syntax error message

username = '''; ⇨ *Reveals the SQL statement structure!*

What's the Problem?

Authenticate without a password using "SQL Injection"

username='

Syntax error message

username = '"; ⇨ *Reveals the SQL statement structure!*

SELECT * FROM **users_list** WHERE username = '**USERNAME**';

statement = "SELECT * FROM **users_list** WHERE username = '" + **USERNAME** + "';"

What’s the Problem?

Authenticate without a password using "SQL Injection"

username=' **OR '1'='1**

What's the Problem?

Authenticate without a password using "SQL Injection"

username=**' OR '1'='1**

SELECT * FROM **users_list** WHERE username = **'' OR '1'='1**';

id	username	
1	Alice	**FALSE OR TRUE = TRUE**
2	Bob	**FALSE OR TRUE = TRUE**
3	Eve	**FALSE OR TRUE = TRUE**

1, Alice
2, Bob
3, Eve

What’s the Problem?

Authenticate without a password using "SQL Injection"

username=**' OR 1=1 --**

What's the Problem?

Authenticate without a password using "SQL Injection"

username=**' OR 1=1 --**

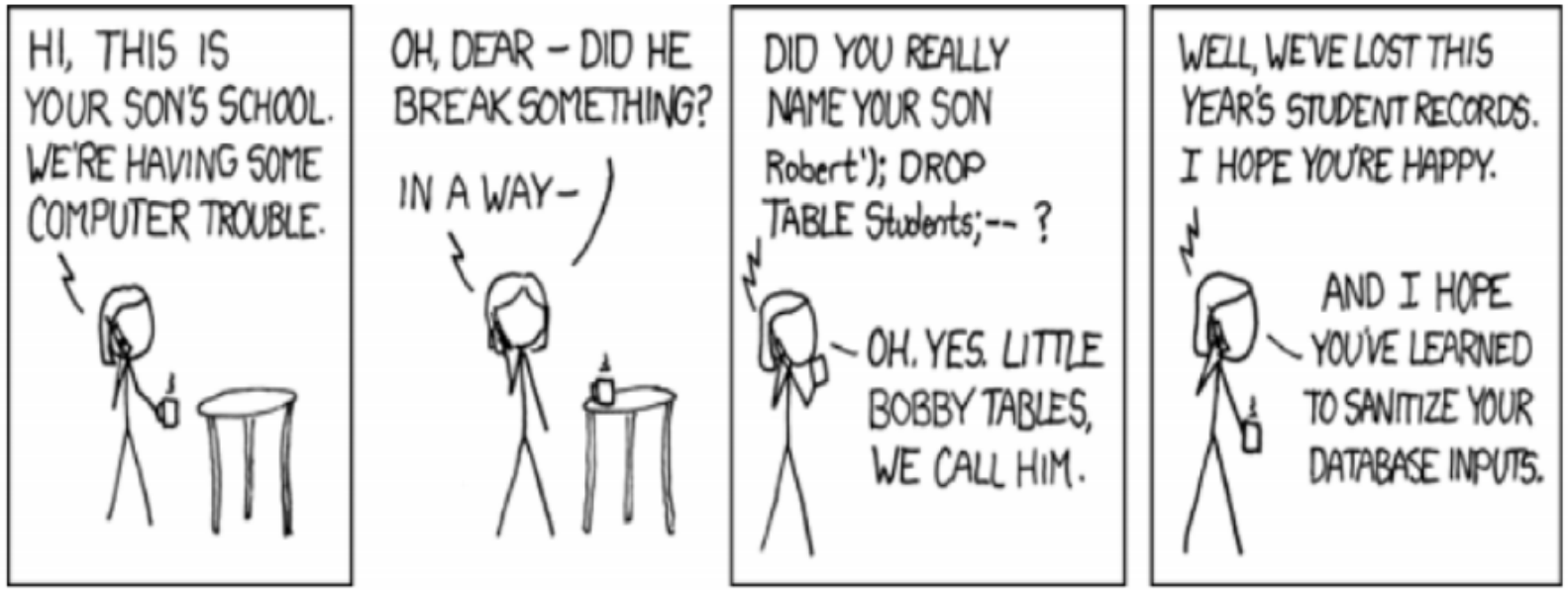
SELECT * FROM **users_list** WHERE username = **'' OR 1=1 --**';

id	username	
1	Alice	**FALSE OR TRUE = TRUE**
2	Bob	**FALSE OR TRUE = TRUE**
3	Eve	**FALSE OR TRUE = TRUE**

1, Alice
2, Bob
3, Eve

What's the Problem?

Authenticate without a password using "SQL Injection"



xkcd webcomic, https://xkcd.com/327/

What’s the Problem?

Authenticate without a password using "SQL Injection"

```
database.execute("INSERT INTO students (name) VALUES ('" + student_name + "');");
INSERT INTO students (name) VALUES ('student_name');
```

What’s the Problem?

Authenticate without a password using "SQL Injection"

database.execute("INSERT INTO students (name) VALUES ('" + student_name + "');");
INSERT INTO students (name) VALUES ('student_name');

student_name=**Alice**

INSERT INTO **students** (name) VALUES ('**Alice**'); ⇨**Alice** is inserted in **students** table

What's the Problem?

Authenticate without a password using "SQL Injection"

```
database.execute("INSERT INTO students (name) VALUES ('" + student_name + "');");
INSERT INTO students (name) VALUES ('student_name');
```

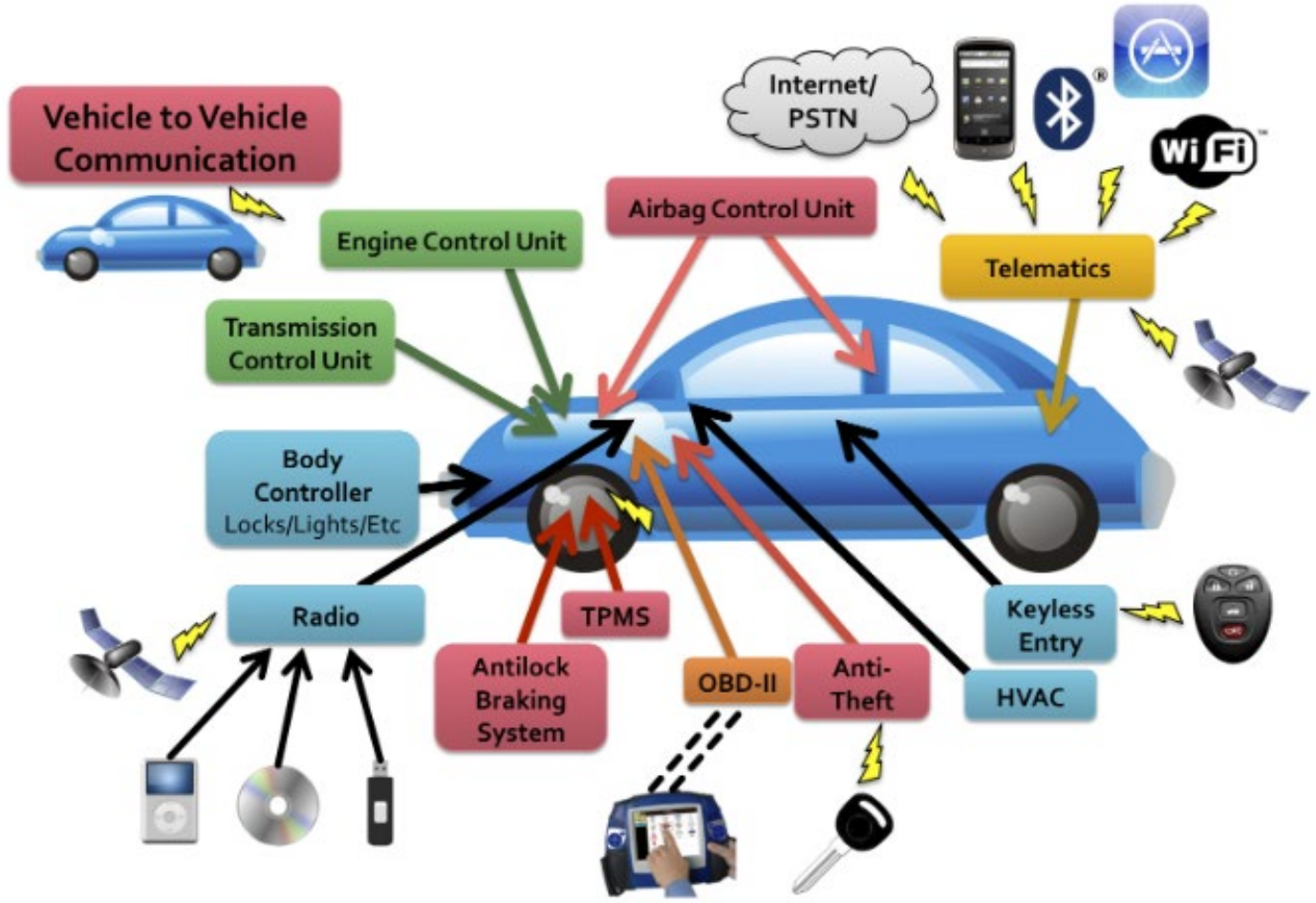
```
student_name=Robert'); DROP TABLE students;--

INSERT INTO students (name) VALUES ('Robert'); DROP TABLE students;-- ');
                                   ⇩
INSERT INTO students (name) VALUES ('Robert');
DROP TABLE students;
--');
```

⇩

Robert is inserted in **students** table, and then **students** table is removed

Example: Attack vectors in a modern car



***Source:** Checkoway et al.; Comprehensive Experimental Analyses of Automotive Attack Surfaces; USENIX Security Conference; 2011.

Embedded Systems in Security Applications

- Smart Cards
  - Banking application: Payment, etc.
  - Access control (also via smart keys)
  - Government ID: IDs, Passports, etc. (Biometric)
  - SIM cards
- Trusted Computing: Trusted Platform Module (TPM)
- Hardware Security Module (HSM)
- Secure Sensing: Tachograph, etc.

Secure Design Flow

- The course roughly follows secure design flow (secure software lifecycle)
  - Foundations and Security Technologies
    - Access Control
    - Cryptography
    - Security Protocols
  - Building Secure Systems
    - Risk Identification, Analyzing Systems
    - Analyzing Security Protocols
    - Application Security & Secure Programming
    - Security Testing

Bibliography

- Frank Vahid, Tony Givargis. Embedded System Design: A Unified Hardware/Software Introduction. John Wiley & Sons, 2002.
  - Book slides are available at: http://esd.cs.ucr.edu/
- Ross J. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2001.
  - The complete book is available at: http://www.cl.cam.ac.uk/~rja14/book.html
- Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. Handbook of Applied Cryptography. CRC Press, Inc., Boca Raton, FL, USA, 5th edition, 2001.
  - The complete book is available at: http://cacr.uwaterloo.ca/hac/
- D. Elliott Bell and Leonard J. LaPadula. Secure Computer Systems: A Mathematical Model, volume II. In Journal of Computer Security 4, pages 229–263, 1996. An Electronic Reconstruction of Secure Computer Systems: Mathematical Foundations, 1973.
- Roger M. Needham and Michael D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. Commun. ACM, 21:993–999, December 1978.
- M Golla, M Wei, J Hainline, L Filipe, M Dürmuth. What was that site doing with my Facebook password? Designing Password-Reuse Notifications. Proceedings of the 2018 ACM SIGSAC Conference, 2018.
- Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D Ullman. Protection in Operating Systems. Communications of the ACM, 19(8):461–471, 1976.

Thanks for your attention!

- Any questions or remarks?