ΕΦΑΡΜΟΓΕΣ ΑΣΦΑΛΕΙΑΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ

Επίκ. Καθ. Αθανάσιος Παπαδημητρίου,

Contents

- GOOD PRACTICES FOR SECURITY OF IOT
ENISA REPORT

Ως προς την εργασία

- Να αναφέρετε και να αναλύσετε Assets και Threats αποκλειστικά του IoT κόμβου και της εφαρμογής σας
- Να προσθέσετε τουλάχιστον 5 μηχανισμούς ασφάλειας και να εντοπίσετε από ποιες απειλές προστατεύουν. Επιπλέον τεχνικές θα δώσουν έξτρα βαθμολογία.
- Αναλύστε τα ευάλωτα σημεία της IoT εφαρμογής που σχεδιάσατε σχηματίζοντας ένα σχηματικό διάγραμμα

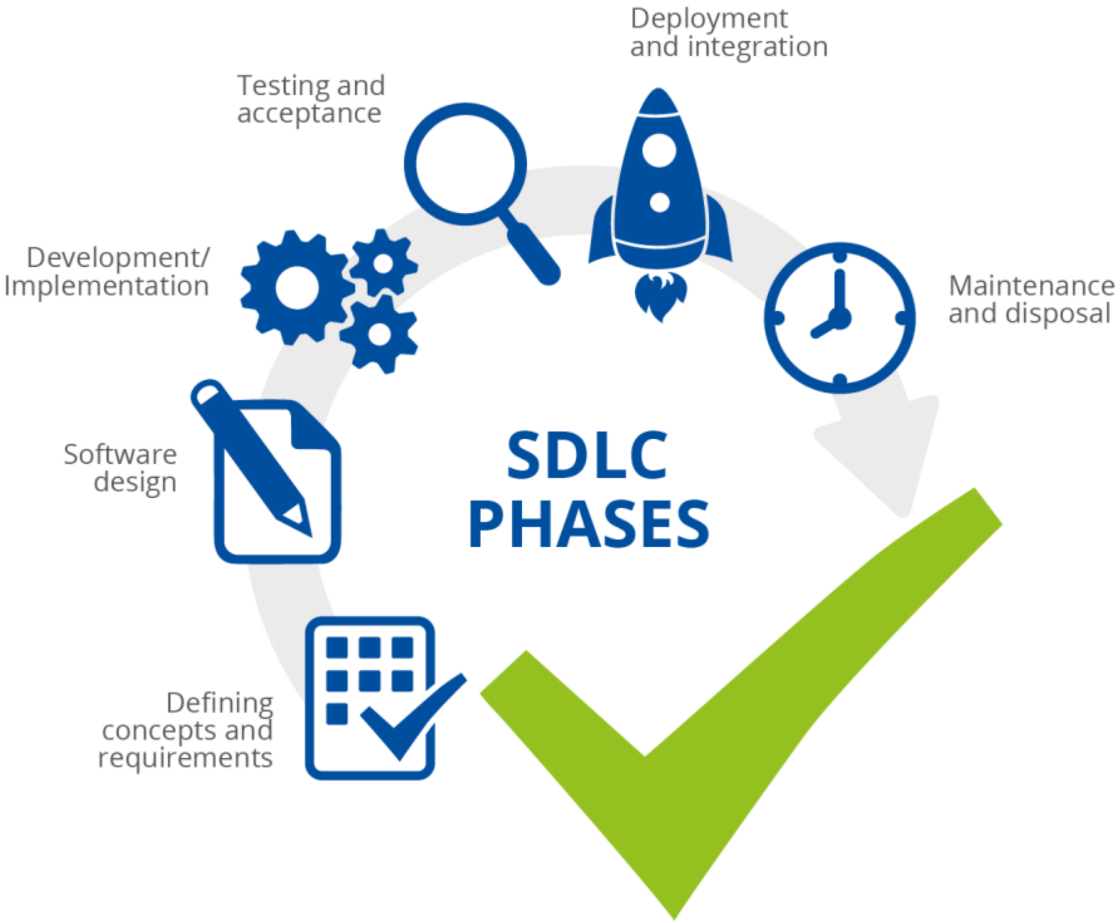
Securing the IoT

- Security and privacy by design and by default
- secure Software Development Life Cycle (sSDLC) principles
- Developers trained in secure coding

secure Software Development Life Cycle (sSDLC)

- Security by design
- Checking for security vulnerabilities
- Secure deployment
- Ensuring continuity of secure development in cases of integrators

SDLC Phases


Deployment and integration
Testing and acceptance
Development/ Implementation
Maintenance and disposal
Software design
SDLC PHASES
Defining concepts and requirements

Software & Hardware

- Hardware & software are interconnected concerning security
- Software is always executed on hardware!
- Software security techniques are not enough
- Hardware can be used to protect against hardware and software attacks
  - Trusted Platform Module (TPM)
  - Crypto engines

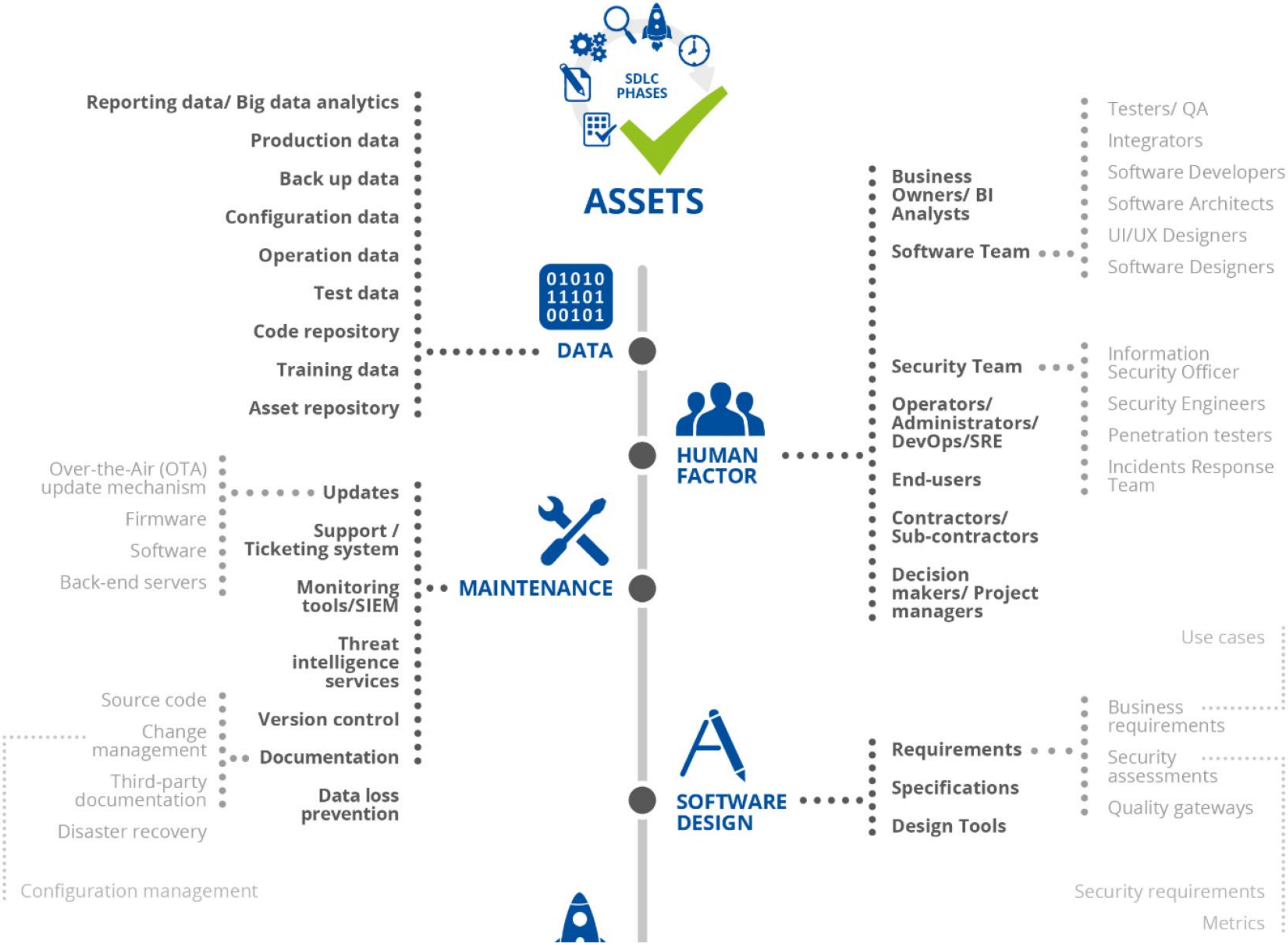
sSDLC

- Requirements
  - E.g. user passwords
- Specifications (must meet requirements)
  - Password change cycles
  - Minimum password length
  - Special symbols
- Threat modelling
- Attack surface analysis
  - Attacker’s potential motivations
  - Intentions and capabilities
  - Attack paths

Early security analysis

- IMPORTANT: Identification and mitigation of security threats early in the design phase
- IoT relies on the CIA triad
  - Confidentiality(Εμπιστευτικότητα)
  - Integrity(Ακεραιότητα)
  - Availability (Διαθεσιμότητα)
- Safety

Asset Taxonomy



SDLC PHASES

ASSETS

DATA

- Reporting data/ Big data analytics
- Production data
- Back up data
- Configuration data
- Operation data
- Test data
- Code repository
- Training data
- Asset repository

HUMAN FACTOR

- Business Owners/ BI Analysts
- Software Team
  - Testers/ QA
  - Integrators
  - Software Developers
  - Software Architects
  - UI/UX Designers
  - Software Designers
- Security Team
  - Information Security Officer
  - Security Engineers
  - Penetration testers
  - Incidents Response Team
- Operators/ Administrators/ DevOps/SRE
- End-users
- Contractors/ Sub-contractors
- Decision makers/ Project managers

MAINTENANCE

- Updates
  - Over-the-Air (OTA) update mechanism
  - Firmware
  - Software
  - Back-end servers
- Support / Ticketing system
- Monitoring tools/SIEM
- Threat intelligence services
- Version control
- Documentation
  - Source code
  - Change management
  - Third-party documentation
  - Disaster recovery
- Data loss prevention

Configuration management

SOFTWARE DESIGN

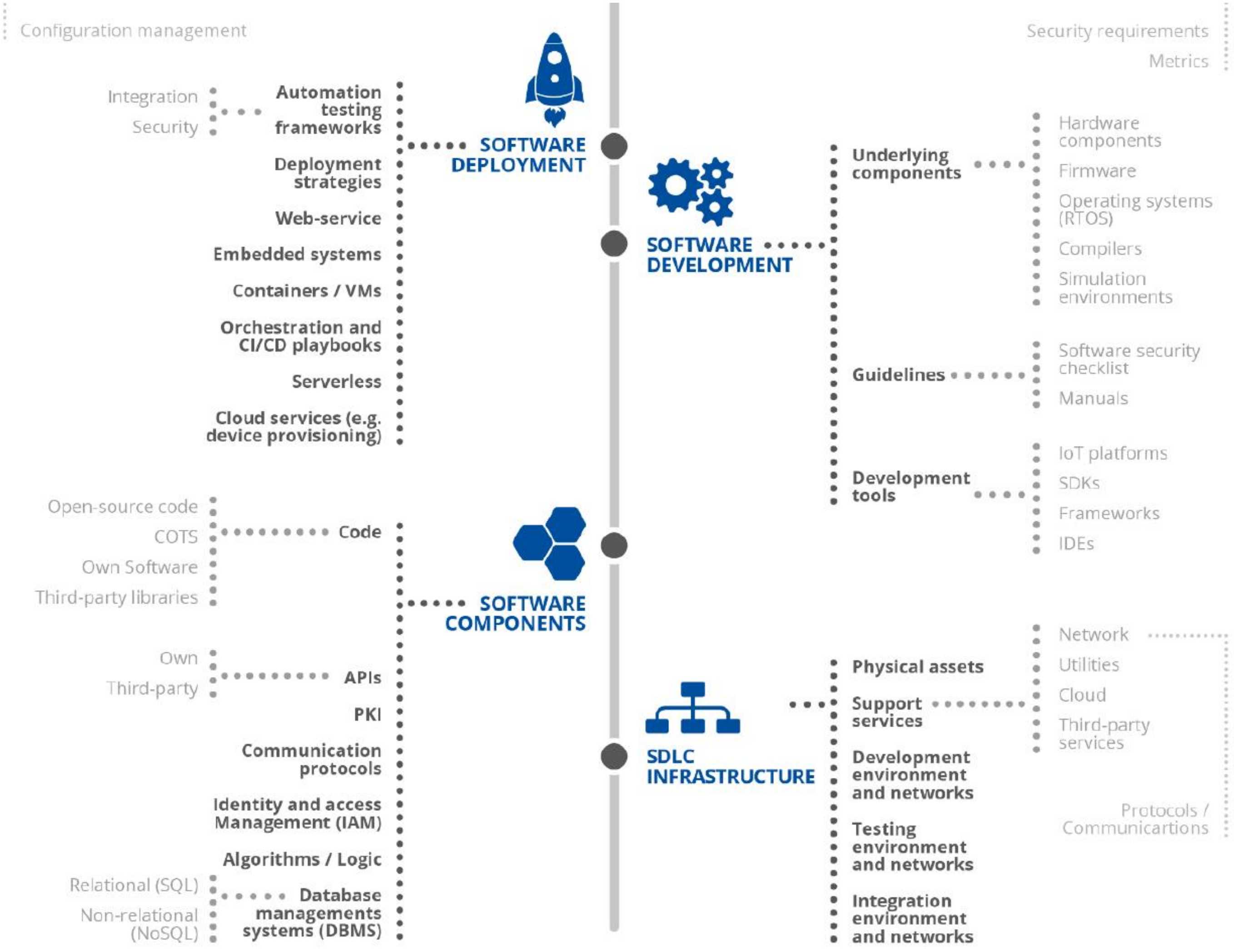
- Requirements
  - Business requirements
  - Security assessments
  - Quality gateways
- Specifications
- Design Tools

Use cases

Security requirements

Metrics

Asset Taxonomy



Configuration management

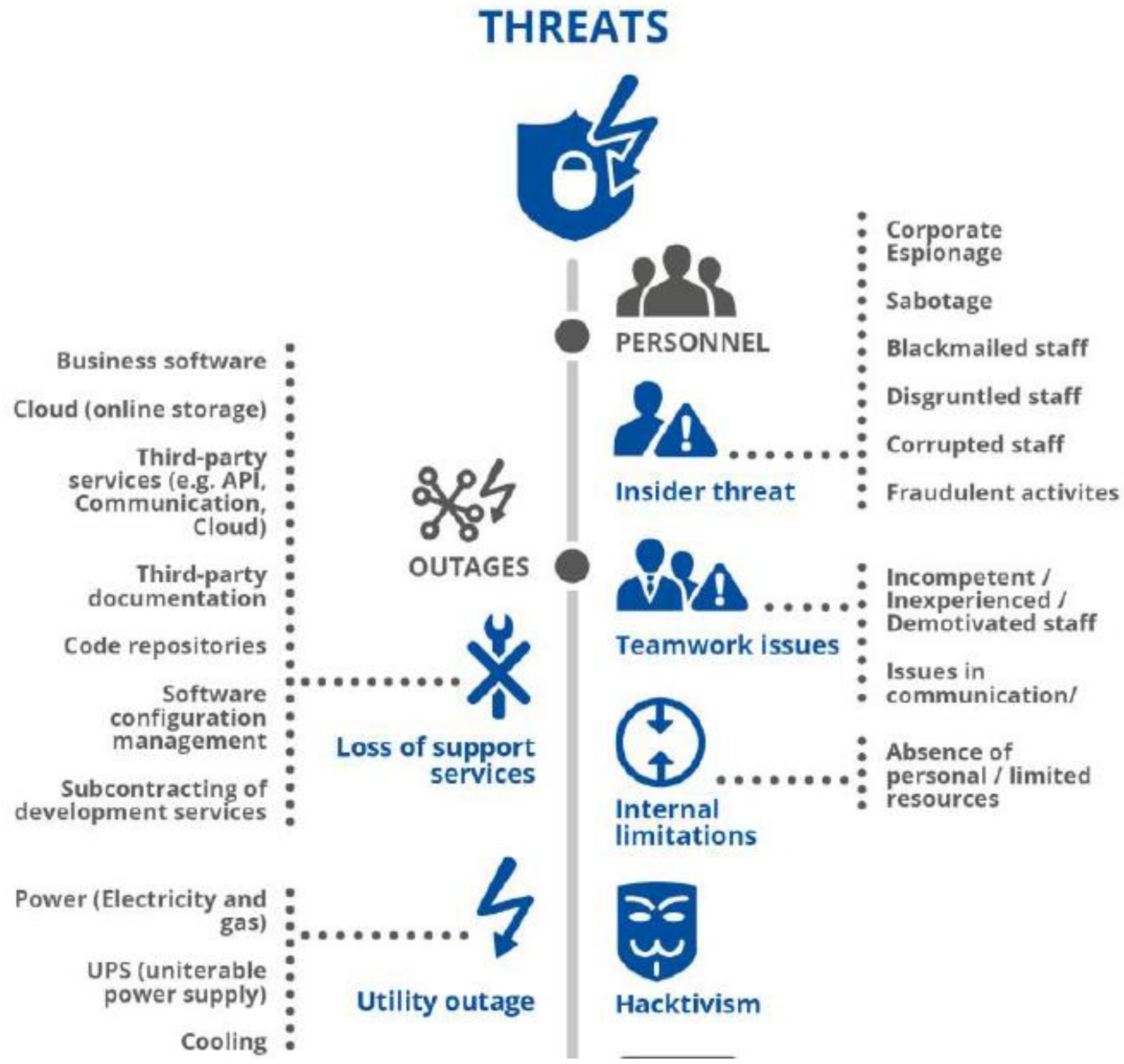
- SOFTWARE DEPLOYMENT
  - Automation testing frameworks
    - Integration
    - Security
  - Deployment strategies
  - Web-service
  - Embedded systems
  - Containers / VMs
  - Orchestration and CI/CD playbooks
  - Serverless
  - Cloud services (e.g. device provisioning)
- SOFTWARE COMPONENTS
  - Code
    - Open-source code
    - COTS
    - Own Software
    - Third-party libraries
  - APIs
    - Own
    - Third-party
  - PKI
  - Communication protocols
  - Identity and access Management (IAM)
  - Algorithms / Logic
  - Database managements systems (DBMS)
    - Relational (SQL)
    - Non-relational (NoSQL)

Security requirements
Metrics

- SOFTWARE DEVELOPMENT
  - Underlying components
    - Hardware components
    - Firmware
    - Operating systems (RTOS)
    - Compilers
    - Simulation environments
  - Guidelines
    - Software security checklist
    - Manuals
  - Development tools
    - IoT platforms
    - SDKs
    - Frameworks
    - IDEs
- SDLC INFRASTRUCTURE
  - Physical assets
  - Support services
    - Network
    - Utilities
    - Cloud
    - Third-party services
  - Development environment and networks
  - Testing environment and networks
  - Integration environment and networks

Protocols / Communicartions

Threat Taxonomy



THREATS

PERSONNEL

Insider threat

- Corporate Espionage
- Sabotage
- Blackmailed staff
- Disgruntled staff
- Corrupted staff
- Fraudulent activites

Teamwork issues

- Incompetent / Inexperienced / Demotivated staff
- Issues in communication/

Internal limitations

- Absence of personal / limited resources

Hacktivism

OUTAGES

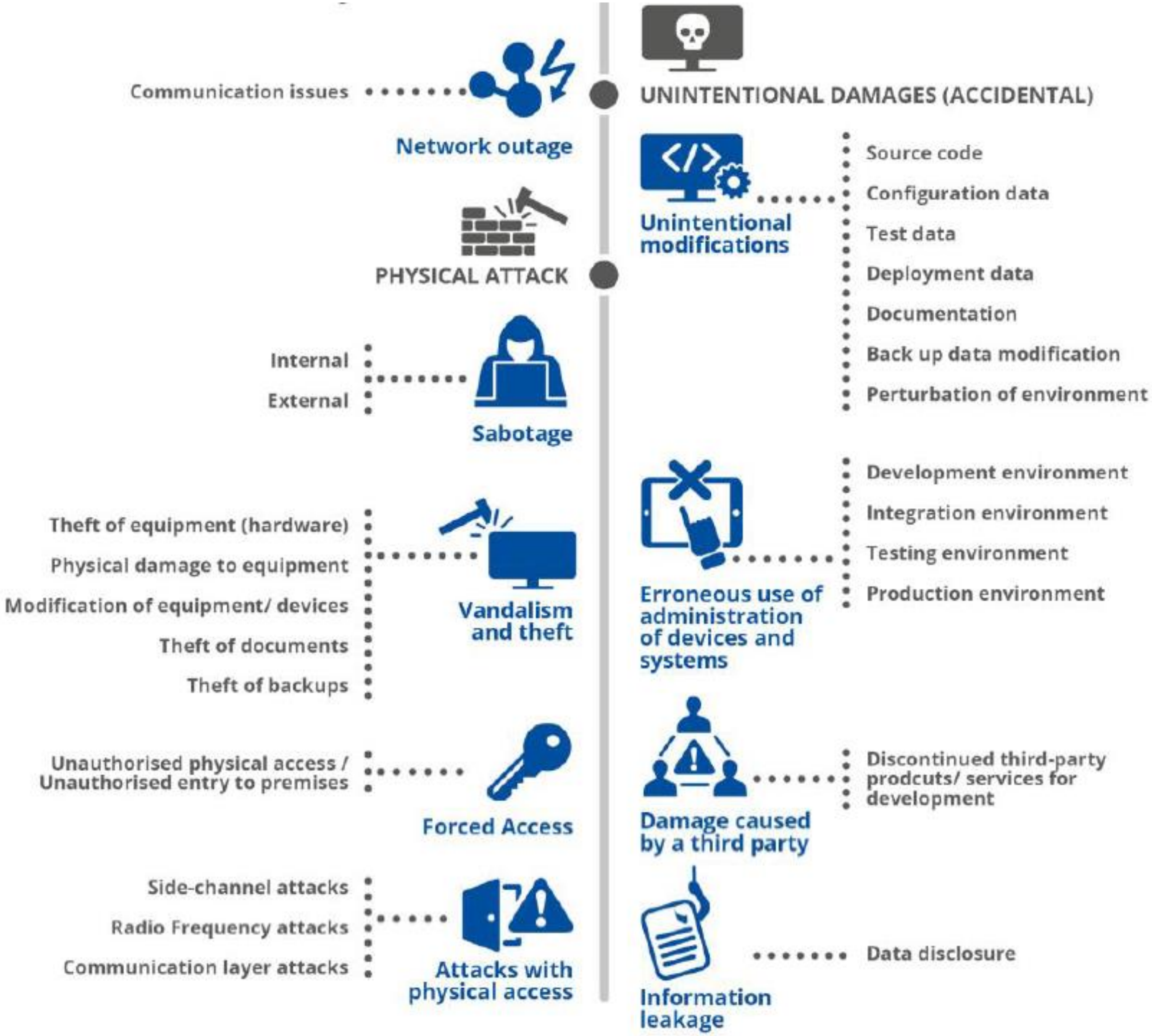
Loss of support services

- Business software
- Cloud (online storage)
- Third-party services (e.g. API, Communication, Cloud)
- Third-party documentation
- Code repositories
- Software configuration management
- Subcontracting of development services

Utility outage

- Power (Electricity and gas)
- UPS (uniterable power supply)
- Cooling

Threat Taxonomy



Communication issues

Network outage

PHYSICAL ATTACK

Internal

External

Sabotage

Theft of equipment (hardware)

Physical damage to equipment

Modification of equipment/ devices

Theft of documents

Theft of backups

Vandalism and theft

Unauthorised physical access / Unauthorised entry to premises

Forced Access

Side-channel attacks

Radio Frequency attacks

Communication layer attacks

Attacks with physical access

UNINTENTIONAL DAMAGES (ACCIDENTAL)

Unintentional modifications

Source code

Configuration data

Test data

Deployment data

Documentation

Back up data modification

Perturbation of environment

Erroneous use of administration of devices and systems

Development environment

Integration environment

Testing environment

Production environment

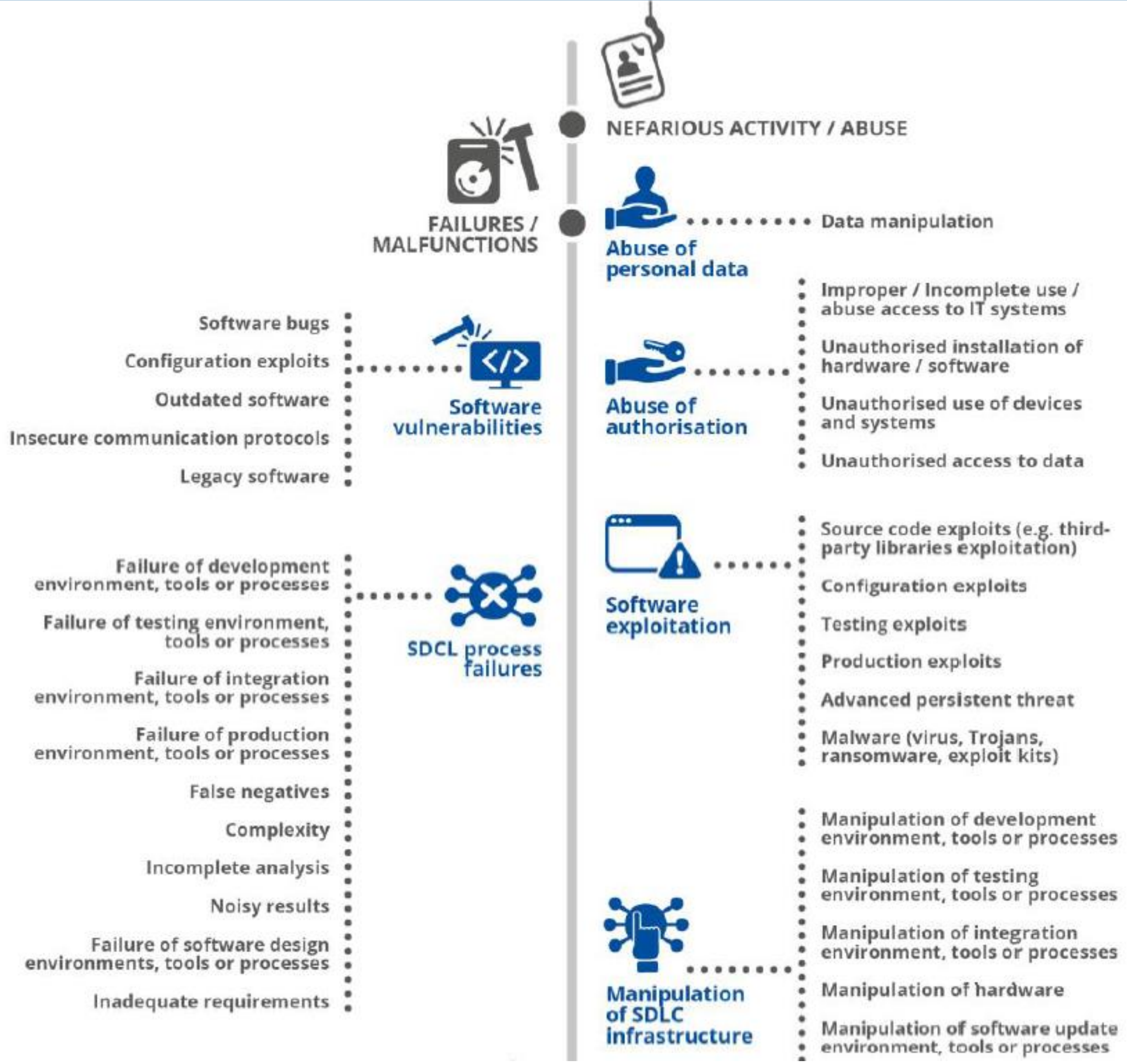
Damage caused by a third party

Discontinued third-party prodcuts/ services for development

Information leakage

Data disclosure

Threat Taxonomy



NEFARIOUS ACTIVITY / ABUSE

FAILURES / MALFUNCTIONS

Software vulnerabilities
- Software bugs
- Configuration exploits
- Outdated software
- Insecure communication protocols
- Legacy software

SDCL process failures
- Failure of development environment, tools or processes
- Failure of testing environment, tools or processes
- Failure of integration environment, tools or processes
- Failure of production environment, tools or processes
- False negatives
- Complexity
- Incomplete analysis
- Noisy results
- Failure of software design environments, tools or processes
- Inadequate requirements

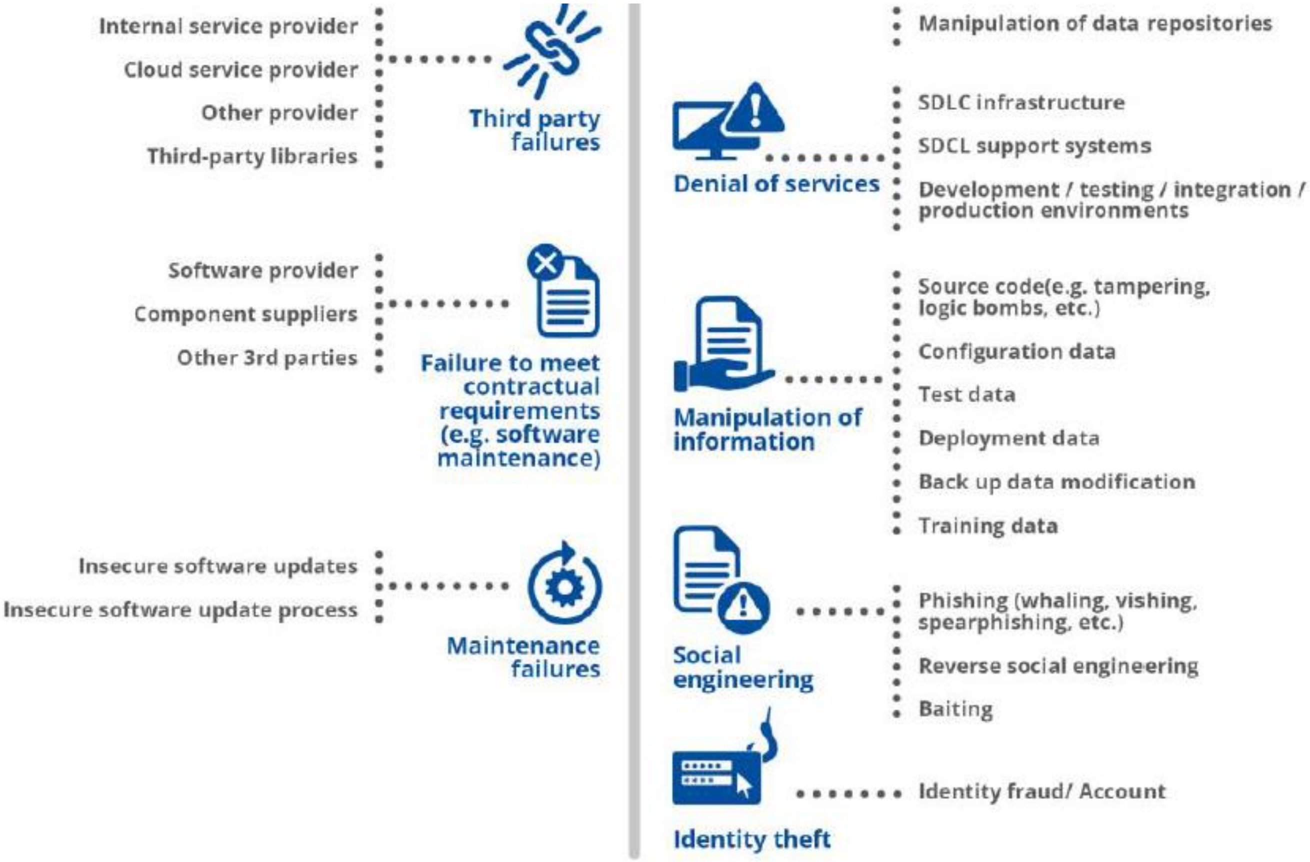
Abuse of personal data
- Data manipulation

Abuse of authorisation
- Improper / Incomplete use / abuse access to IT systems
- Unauthorised installation of hardware / software
- Unauthorised use of devices and systems
- Unauthorised access to data

Software exploitation
- Source code exploits (e.g. third-party libraries exploitation)
- Configuration exploits
- Testing exploits
- Production exploits
- Advanced persistent threat
- Malware (virus, Trojans, ransomware, exploit kits)

Manipulation of SDLC infrastructure
- Manipulation of development environment, tools or processes
- Manipulation of testing environment, tools or processes
- Manipulation of integration environment, tools or processes
- Manipulation of hardware
- Manipulation of software update environment, tools or processes

Threat Taxonomy



Third party failures
- Internal service provider
- Cloud service provider
- Other provider
- Third-party libraries

Failure to meet contractual requirements (e.g. software maintenance)
- Software provider
- Component suppliers
- Other 3rd parties

Maintenance failures
- Insecure software updates
- Insecure software update process

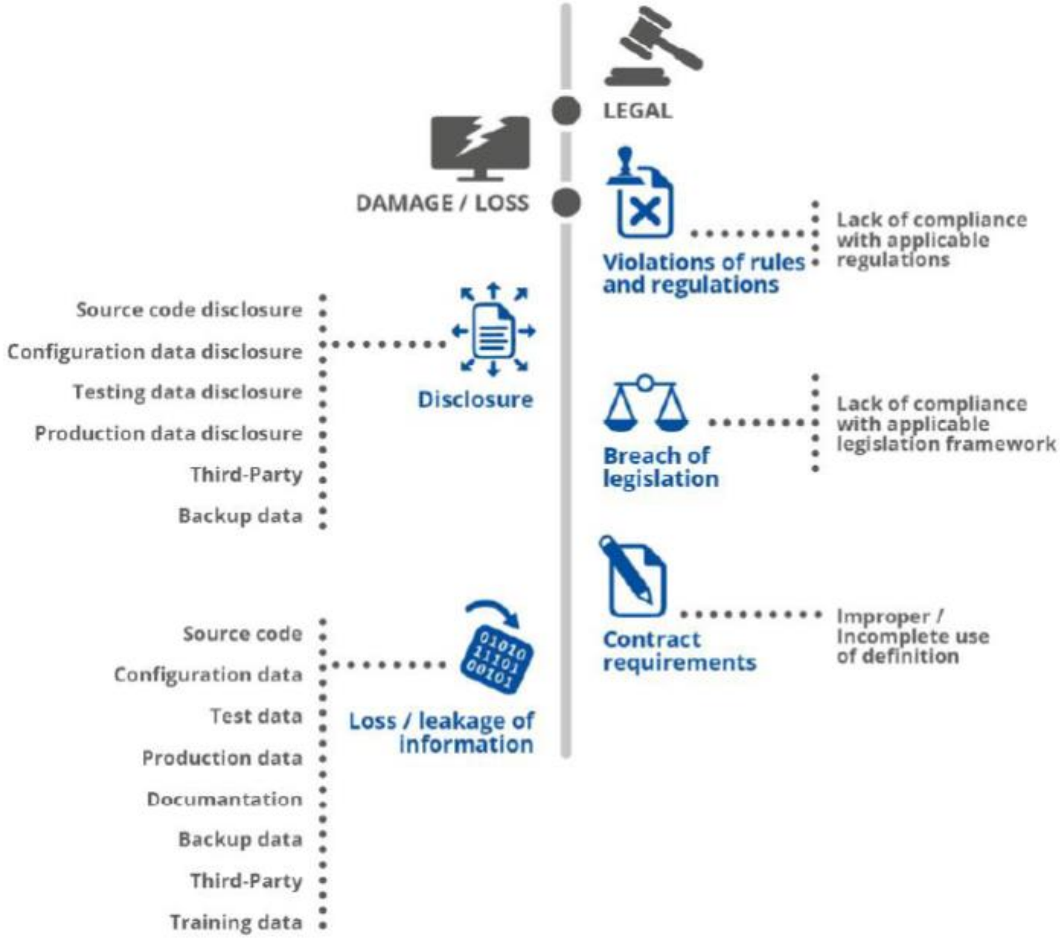
Denial of services
- Manipulation of data repositories
- SDLC infrastructure
- SDCL support systems
- Development / testing / integration / production environments

Manipulation of information
- Source code(e.g. tampering, logic bombs, etc.)
- Configuration data
- Test data
- Deployment data
- Back up data modification
- Training data

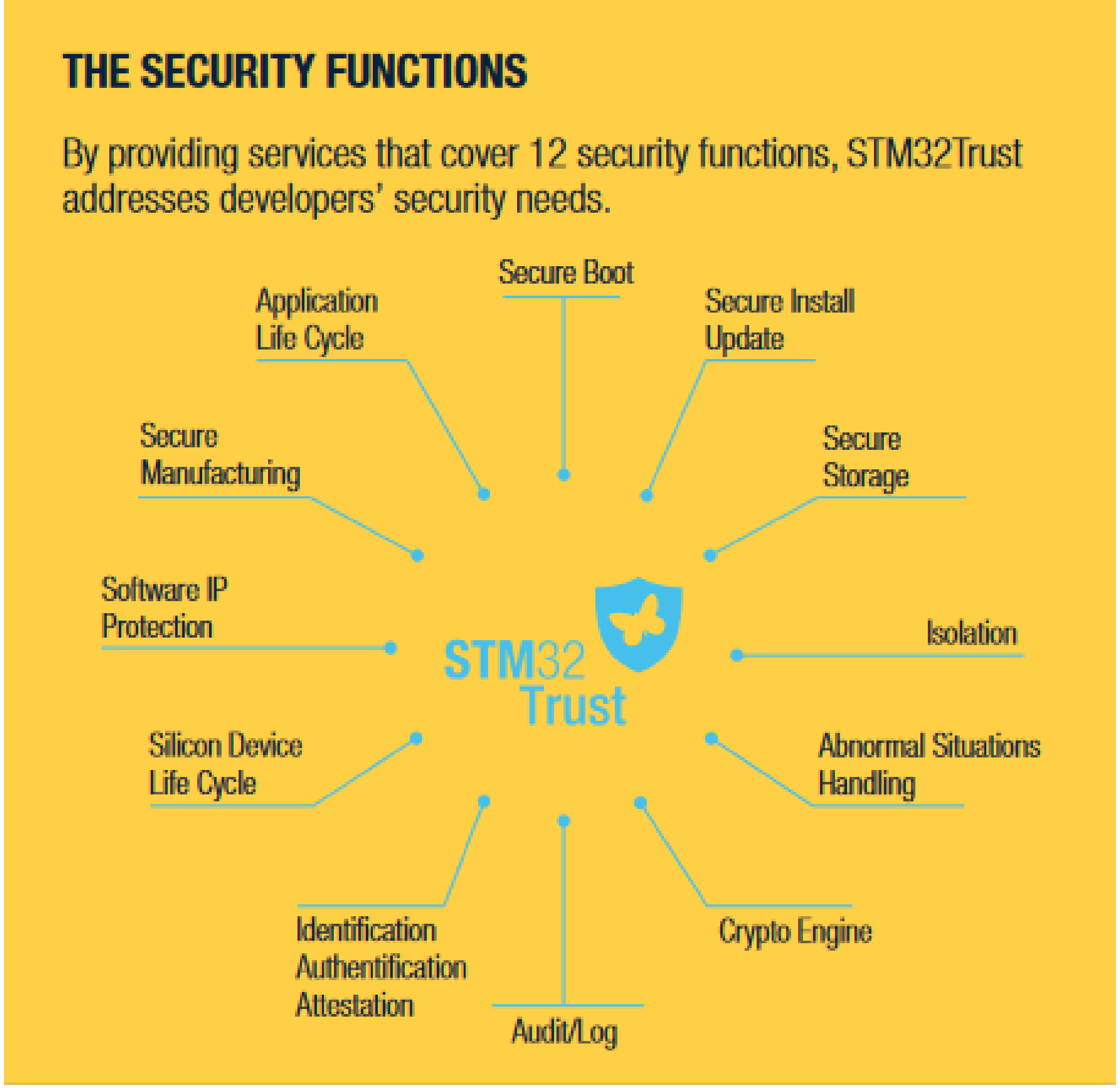
Social engineering
- Phishing (whaling, vishing, spearphishing, etc.)
- Reverse social engineering
- Baiting

Identity theft
- Identity fraud/ Account

Threat Taxonomy


LEGAL
DAMAGE / LOSS
Violations of rules and regulations
Lack of compliance with applicable regulations
Source code disclosure
Configuration data disclosure
Testing data disclosure
Production data disclosure
Third-Party
Backup data
Disclosure
Breach of legislation
Lack of compliance with applicable legislation framework
Contract requirements
Improper / Incomplete use of definition
01010
11101
00101
Source code
Configuration data
Test data
Production data
Documantation
Backup data
Third-Party
Training data
Loss / leakage of information

Συναρτήσεις Ασφάλειας (STM32)



THE SECURITY FUNCTIONS

By providing services that cover 12 security functions, STM32Trust addresses developers' security needs.

Secure Boot

Secure Install Update

Secure Storage

Isolation

Abnormal Situations Handling

Crypto Engine

Audit/Log

Identification Authentification Attestation

Silicon Device Life Cycle

Software IP Protection

Secure Manufacturing

Application Life Cycle

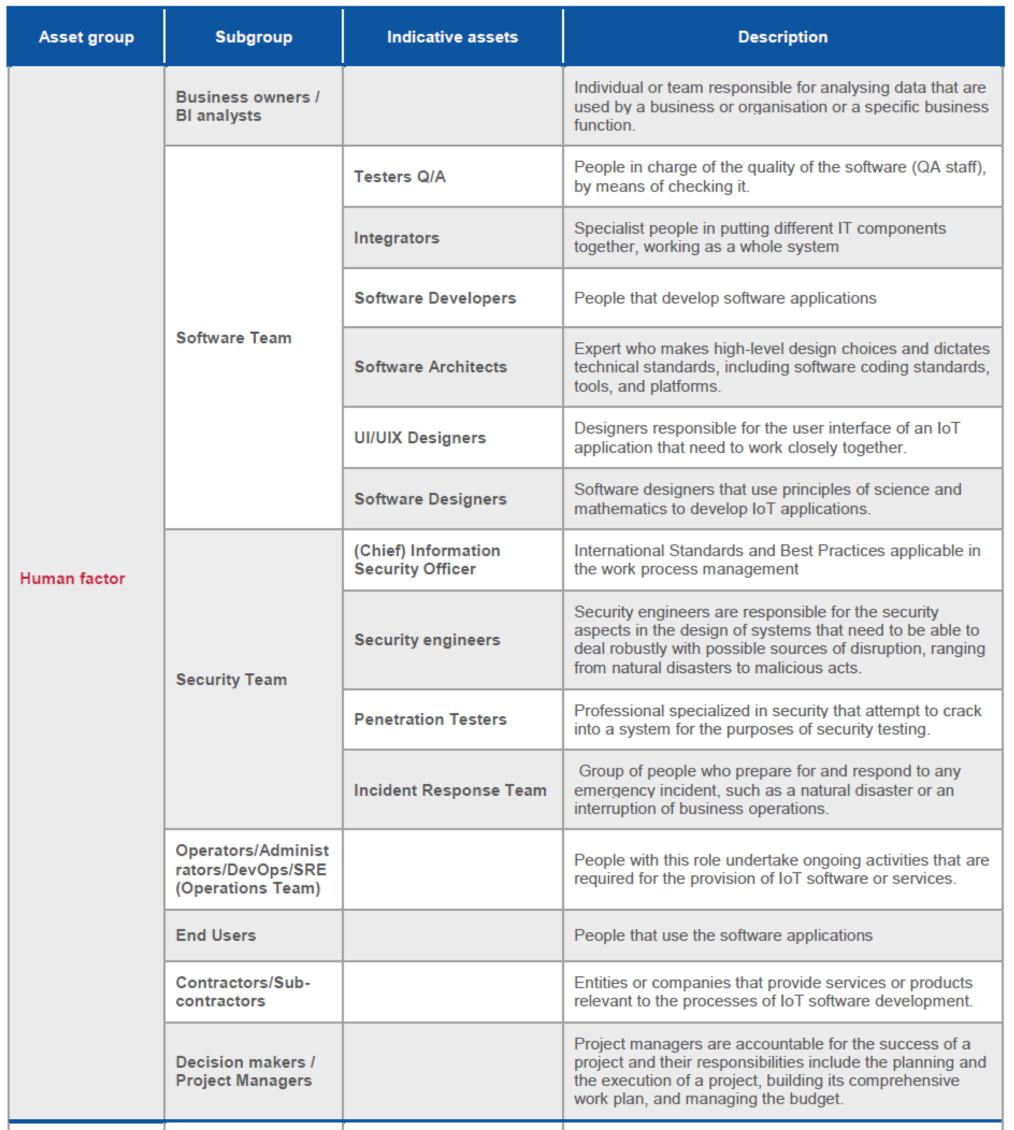
Συναρτήσεις Ασφάλειας (STM32)

- Secure boot: Ability to ensure the authenticity and integrity of an embedded application
- Secure Install/Update: Installation or update of firmware with initial integrity and authenticity checks before programming and execution
- Secure Storage: Ability to securely store secrets like data or keys
- Isolation: Isolation between trusted and non-trusted parts of an application
- Abnormal situation handling: Ability to detect abnormal situations (both hardware and software) and to take adapted decisions such as removing secret data
- Crypto Engine: Ability to process cryptographic algorithms, as recommended by security assurance schemes
- Audit/Log: Keep trace of security events in an unchangeable way
- Identification / Authentication / Attestation: Unique identification of a device and/or software, and ability to detect its authenticity, inside the device or externally
- Silicon Device Lifecycle: Control states to securely protect silicon device assets through a constrained path
- Software IP Protection: Ability to protect a section or the whole software package against external or internal reading. Can be multi-tenant
- Secure Manufacturing: Initial device provisioning in unsecured environment with overproduction control. Possibility to personalize secure components
- Application Life Cycle: Define unchangeable incremental states to securely protect application states and assets

Bibliography

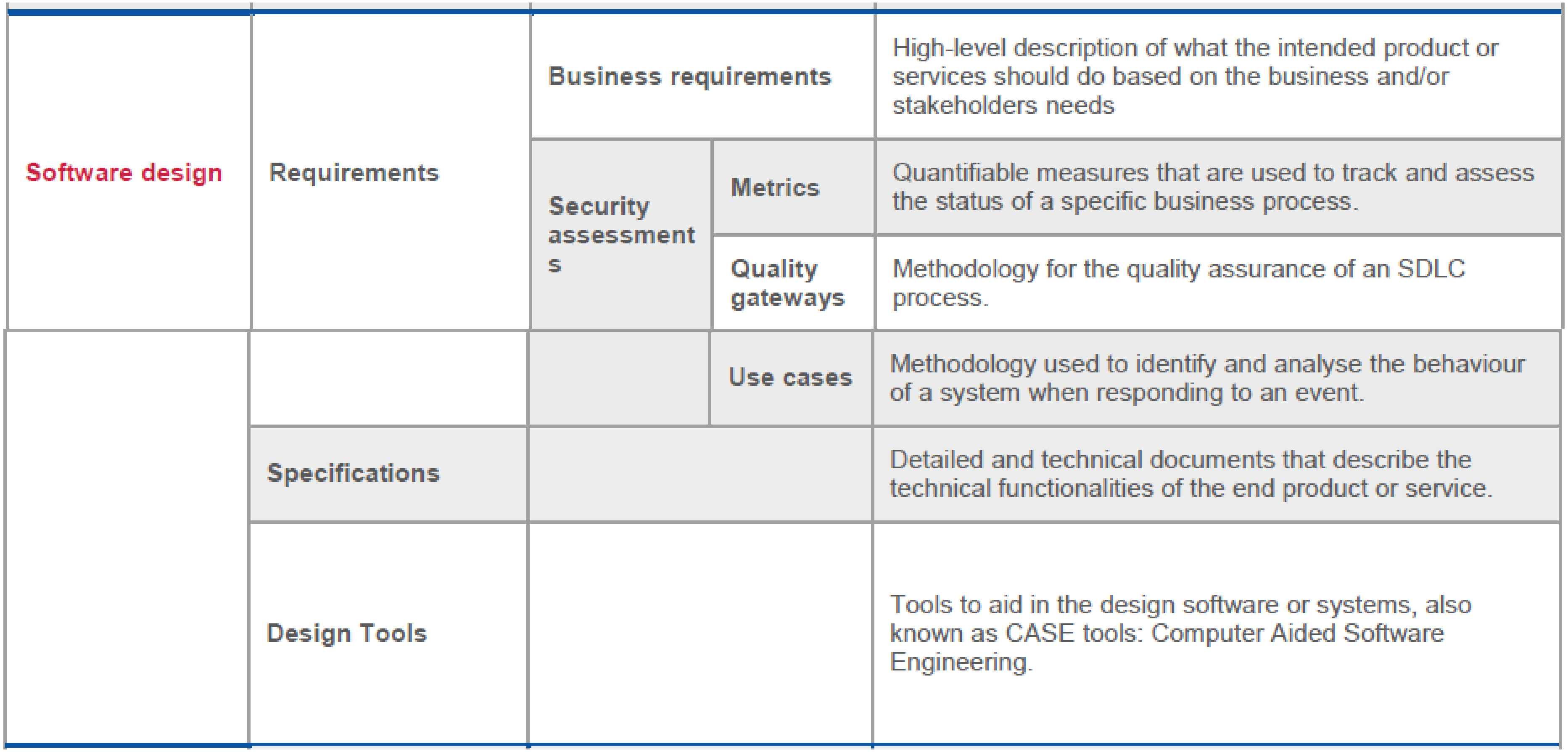
- European Union Agency for Cybersecurity (ENISA)
  - https://www.enisa.europa.eu/publications/good-practices-for-security-of-iot-1
  - https://www.enisa.europa.eu/publications/guidelines-for-securing-the-internet-of-things

Human Factor



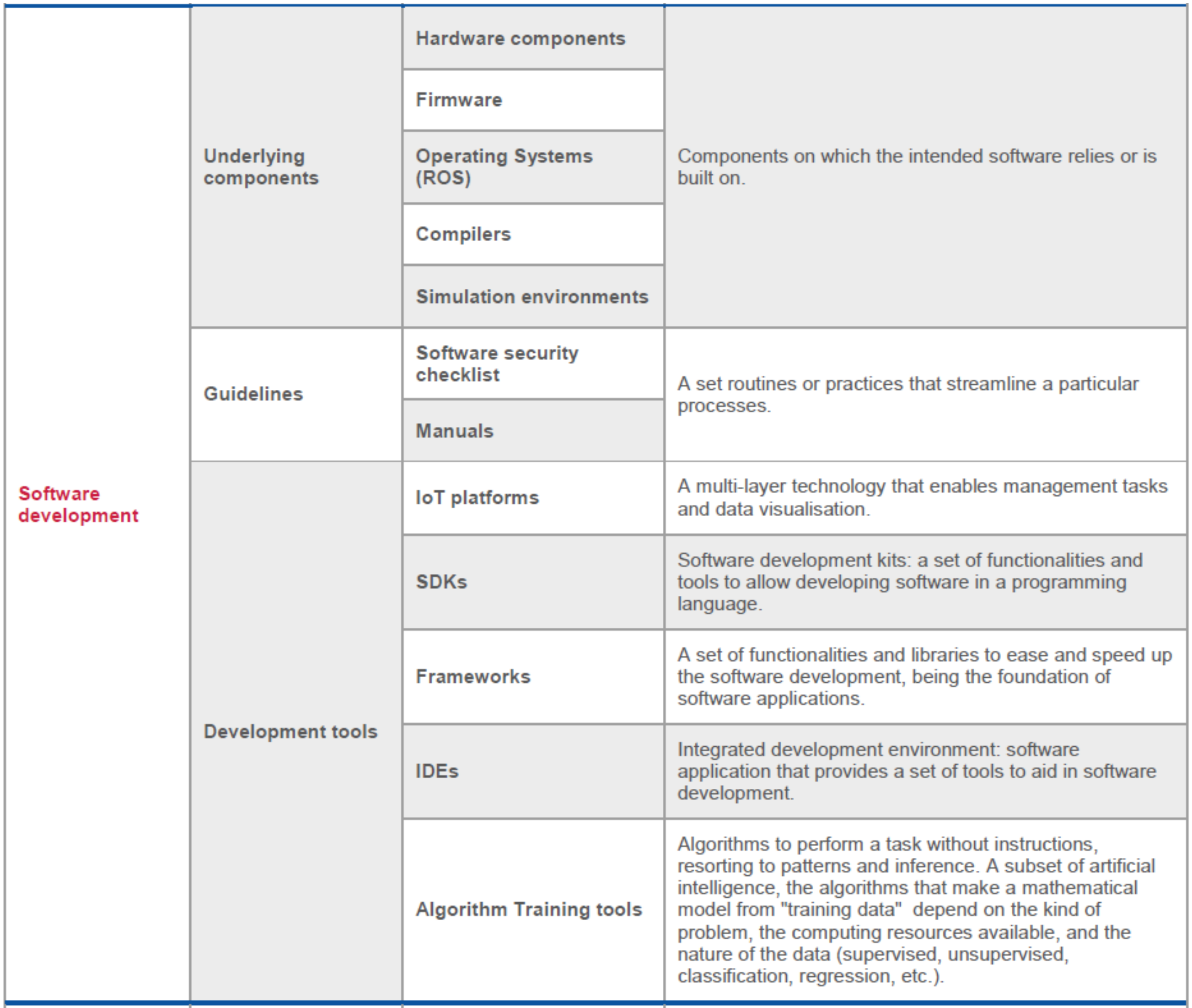
Asset group	Subgroup	Indicative assets	Description
Human factor	Business owners / BI analysts		Individual or team responsible for analysing data that are used by a business or organisation or a specific business function.
	Software Team	Testers Q/A	People in charge of the quality of the software (QA staff), by means of checking it.
		Integrators	Specialist people in putting different IT components together, working as a whole system
		Software Developers	People that develop software applications
		Software Architects	Expert who makes high-level design choices and dictates technical standards, including software coding standards, tools, and platforms.
		UI/UIX Designers	Designers responsible for the user interface of an IoT application that need to work closely together.
		Software Designers	Software designers that use principles of science and mathematics to develop IoT applications.
	Security Team	(Chief) Information Security Officer	International Standards and Best Practices applicable in the work process management
		Security engineers	Security engineers are responsible for the security aspects in the design of systems that need to be able to deal robustly with possible sources of disruption, ranging from natural disasters to malicious acts.
		Penetration Testers	Professional specialized in security that attempt to crack into a system for the purposes of security testing.
		Incident Response Team	Group of people who prepare for and respond to any emergency incident, such as a natural disaster or an interruption of business operations.
	Operators/Administrators/DevOps/SRE (Operations Team)		People with this role undertake ongoing activities that are required for the provision of IoT software or services.
	End Users		People that use the software applications
	Contractors/Sub-contractors		Entities or companies that provide services or products relevant to the processes of IoT software development.
	Decision makers / Project Managers		Project managers are accountable for the success of a project and their responsibilities include the planning and the execution of a project, building its comprehensive work plan, and managing the budget.

Software Design



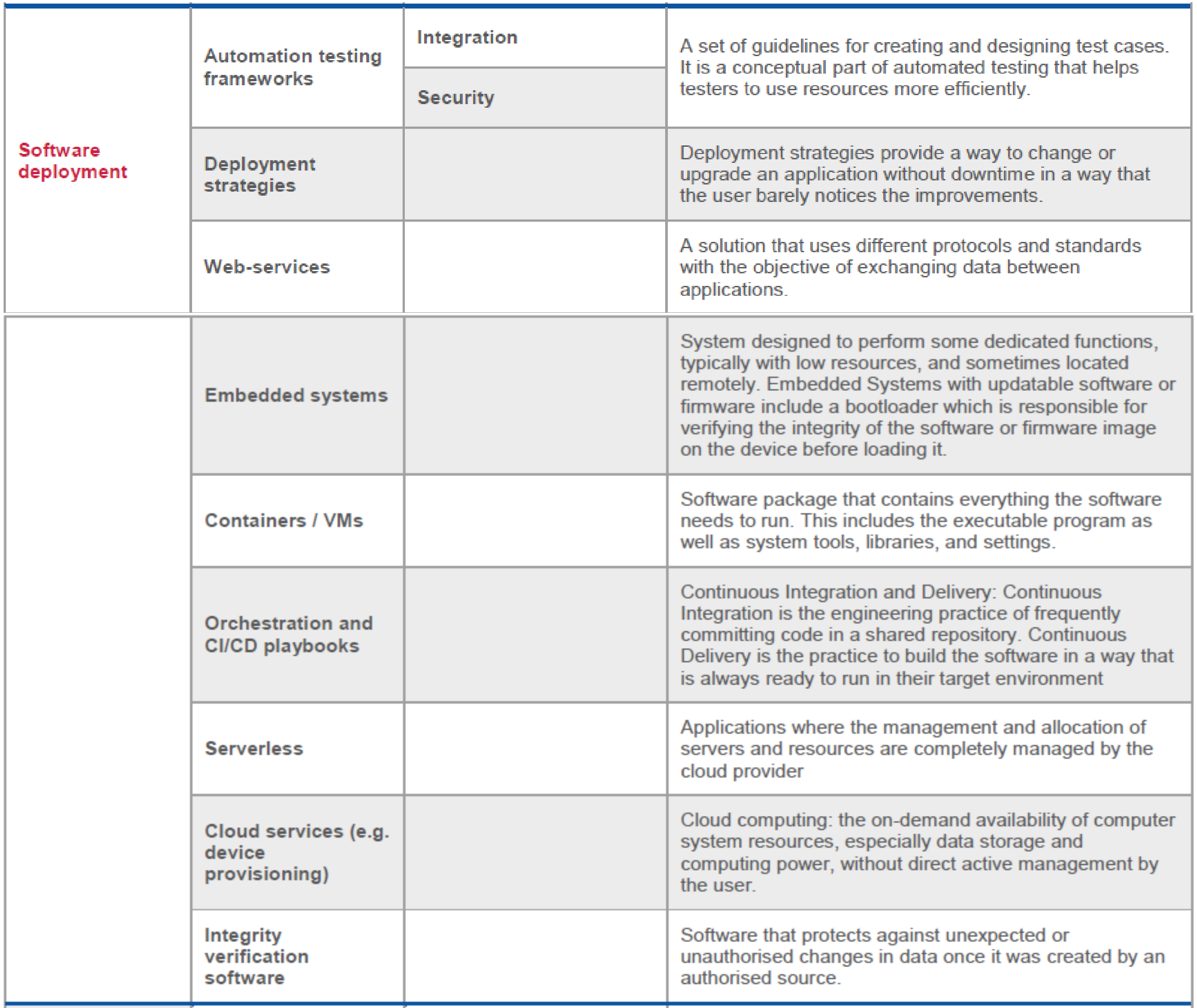
Software design	Requirements	Business requirements		High-level description of what the intended product or services should do based on the business and/or stakeholders needs
		Security assessments	Metrics	Quantifiable measures that are used to track and assess the status of a specific business process.
			Quality gateways	Methodology for the quality assurance of an SDLC process.
			Use cases	Methodology used to identify and analyse the behaviour of a system when responding to an event.
	Specifications			Detailed and technical documents that describe the technical functionalities of the end product or service.
	Design Tools			Tools to aid in the design software or systems, also known as CASE tools: Computer Aided Software Engineering.

Software Development



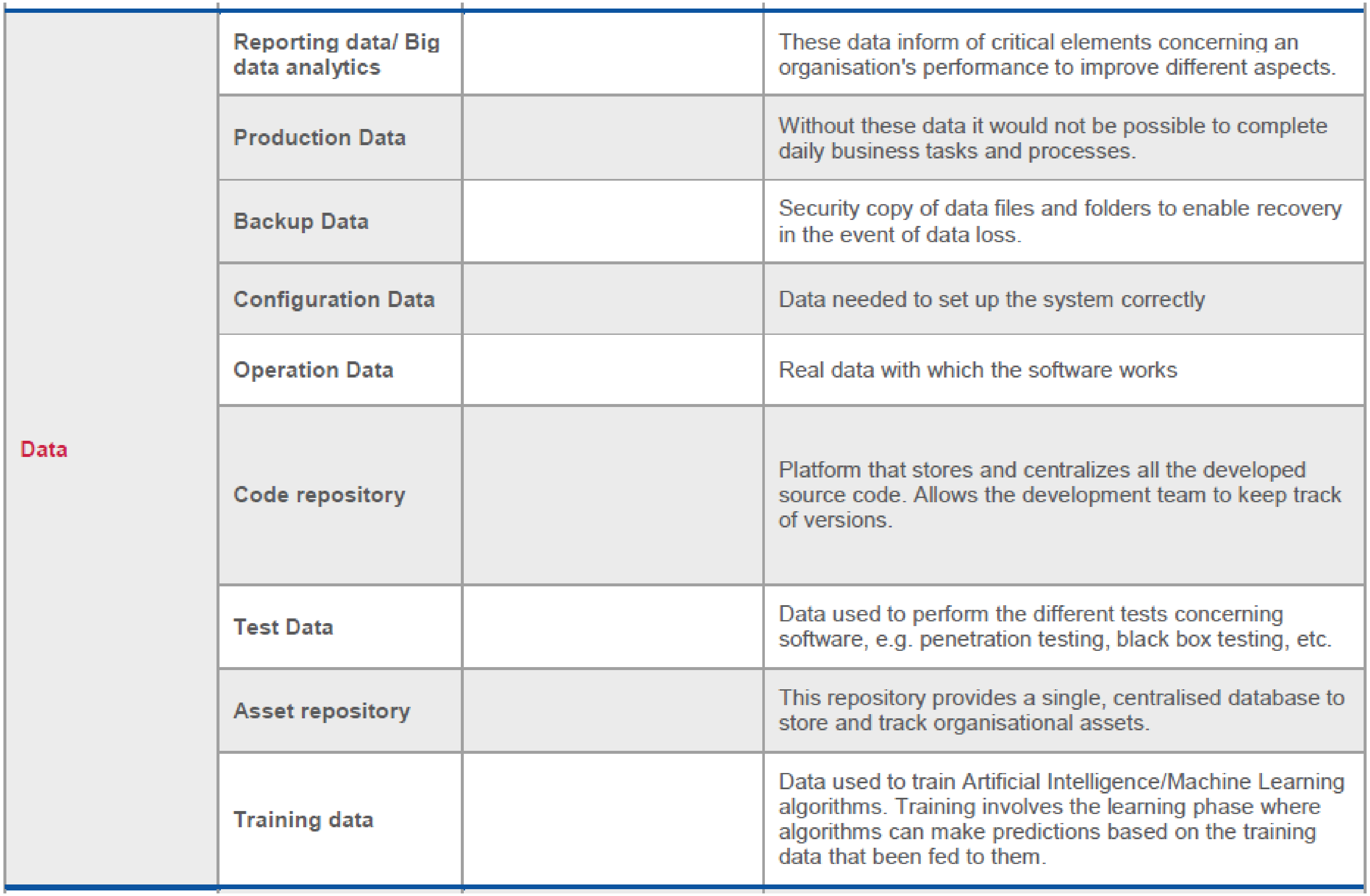
Software development	Underlying components	Hardware components	Components on which the intended software relies or is built on.
		Firmware	
		Operating Systems (ROS)	
		Compilers	
		Simulation environments	
	Guidelines	Software security checklist	A set routines or practices that streamline a particular processes.
		Manuals	
	Development tools	IoT platforms	A multi-layer technology that enables management tasks and data visualisation.
		SDKs	Software development kits: a set of functionalities and tools to allow developing software in a programming language.
		Frameworks	A set of functionalities and libraries to ease and speed up the software development, being the foundation of software applications.
		IDEs	Integrated development environment: software application that provides a set of tools to aid in software development.
		Algorithm Training tools	Algorithms to perform a task without instructions, resorting to patterns and inference. A subset of artificial intelligence, the algorithms that make a mathematical model from "training data" depend on the kind of problem, the computing resources available, and the nature of the data (supervised, unsupervised, classification, regression, etc.).

Software Deployment



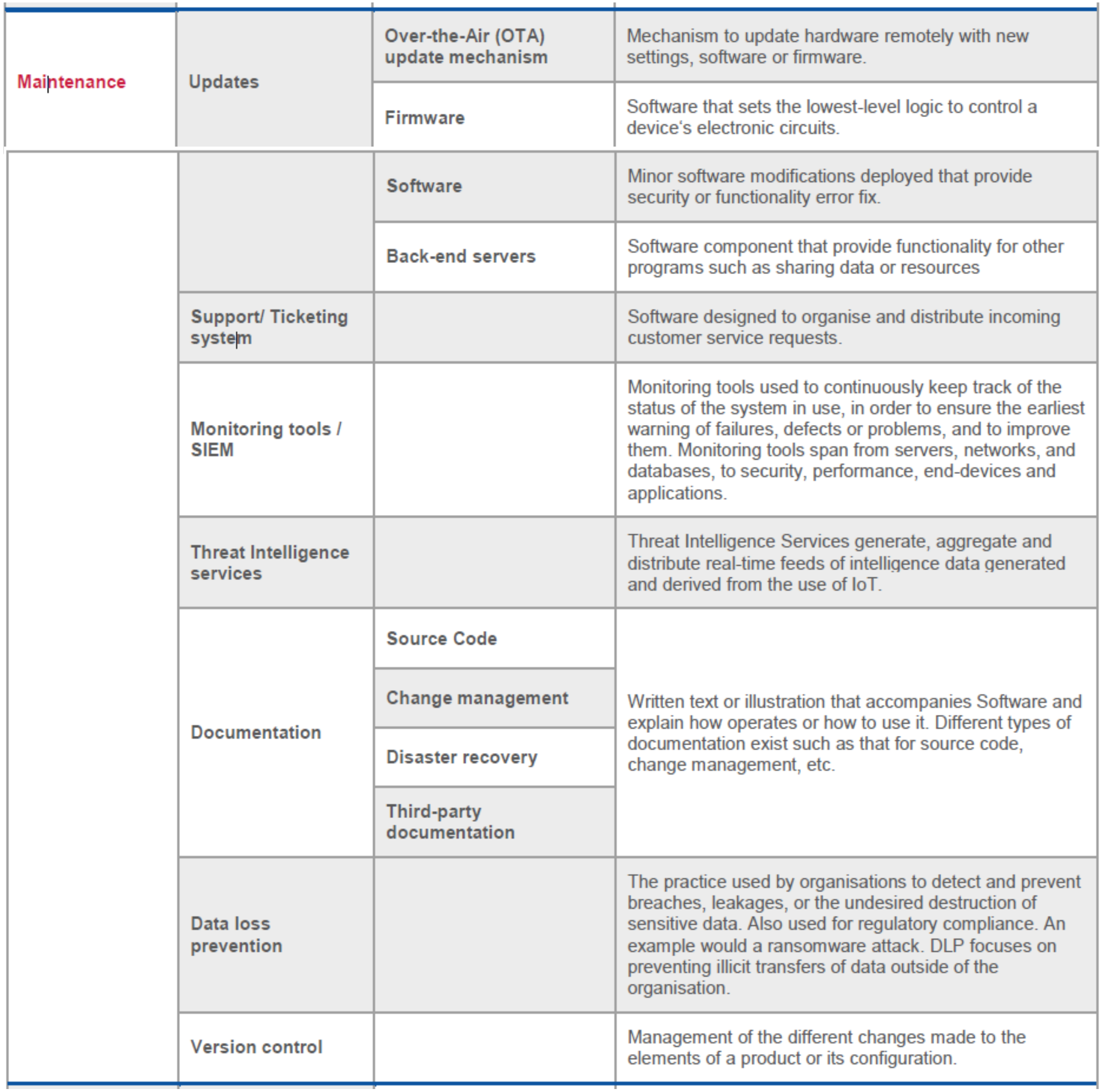
Software deployment	Automation testing frameworks	Integration	A set of guidelines for creating and designing test cases. It is a conceptual part of automated testing that helps testers to use resources more efficiently.
		Security	
	Deployment strategies		Deployment strategies provide a way to change or upgrade an application without downtime in a way that the user barely notices the improvements.
	Web-services		A solution that uses different protocols and standards with the objective of exchanging data between applications.
	Embedded systems		System designed to perform some dedicated functions, typically with low resources, and sometimes located remotely. Embedded Systems with updatable software or firmware include a bootloader which is responsible for verifying the integrity of the software or firmware image on the device before loading it.
	Containers / VMs		Software package that contains everything the software needs to run. This includes the executable program as well as system tools, libraries, and settings.
	Orchestration and CI/CD playbooks		Continuous Integration and Delivery: Continuous Integration is the engineering practice of frequently committing code in a shared repository. Continuous Delivery is the practice to build the software in a way that is always ready to run in their target environment
	Serverless		Applications where the management and allocation of servers and resources are completely managed by the cloud provider
	Cloud services (e.g. device provisioning)		Cloud computing: the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.
	Integrity verification software		Software that protects against unexpected or unauthorised changes in data once it was created by an authorised source.

Data



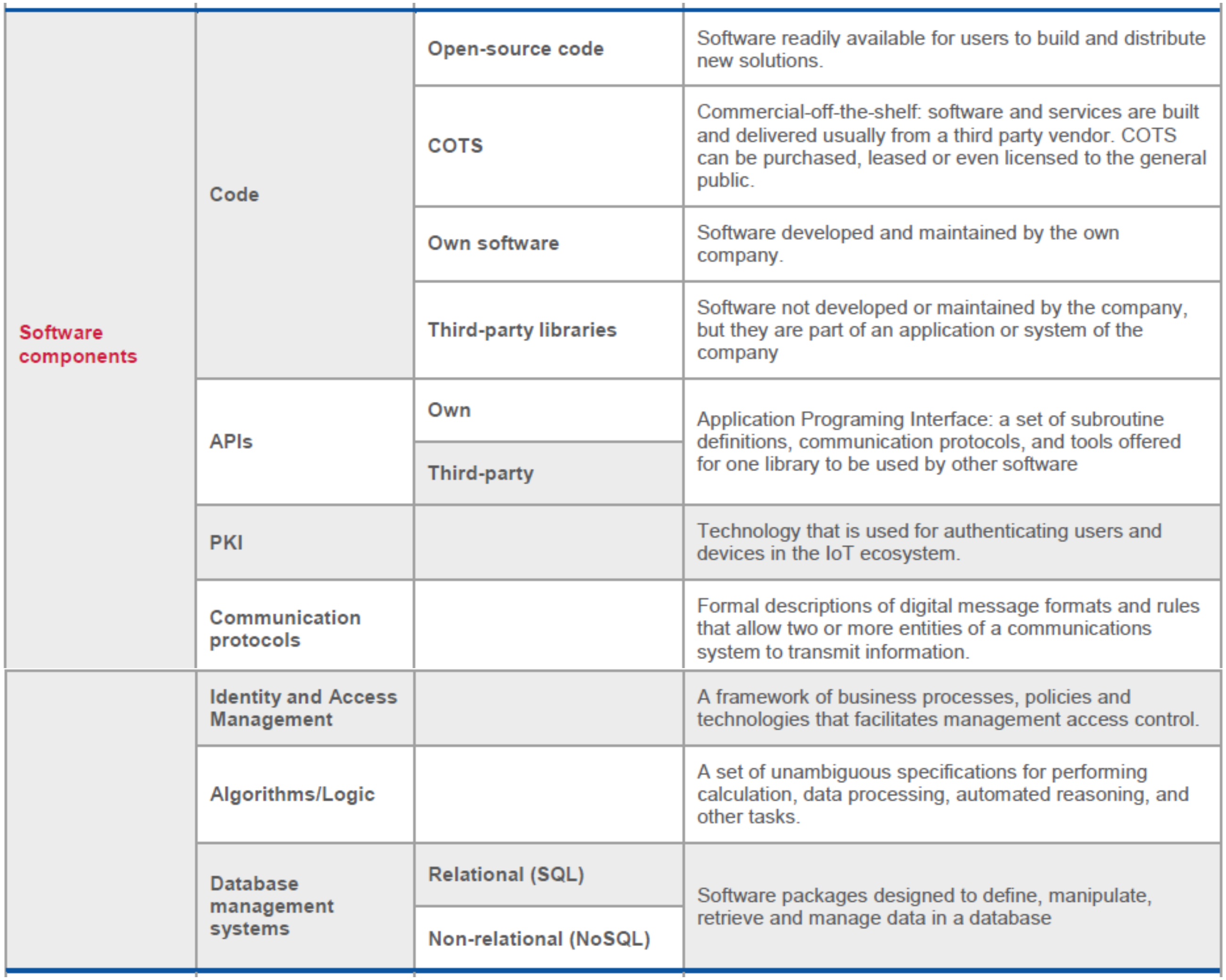
Data	**Reporting data/ Big data analytics**		These data inform of critical elements concerning an organisation's performance to improve different aspects.
	Production Data		Without these data it would not be possible to complete daily business tasks and processes.
	Backup Data		Security copy of data files and folders to enable recovery in the event of data loss.
	Configuration Data		Data needed to set up the system correctly
	Operation Data		Real data with which the software works
	Code repository		Platform that stores and centralizes all the developed source code. Allows the development team to keep track of versions.
	Test Data		Data used to perform the different tests concerning software, e.g. penetration testing, black box testing, etc.
	Asset repository		This repository provides a single, centralised database to store and track organisational assets.
	Training data		Data used to train Artificial Intelligence/Machine Learning algorithms. Training involves the learning phase where algorithms can make predictions based on the training data that been fed to them.

Maintenance



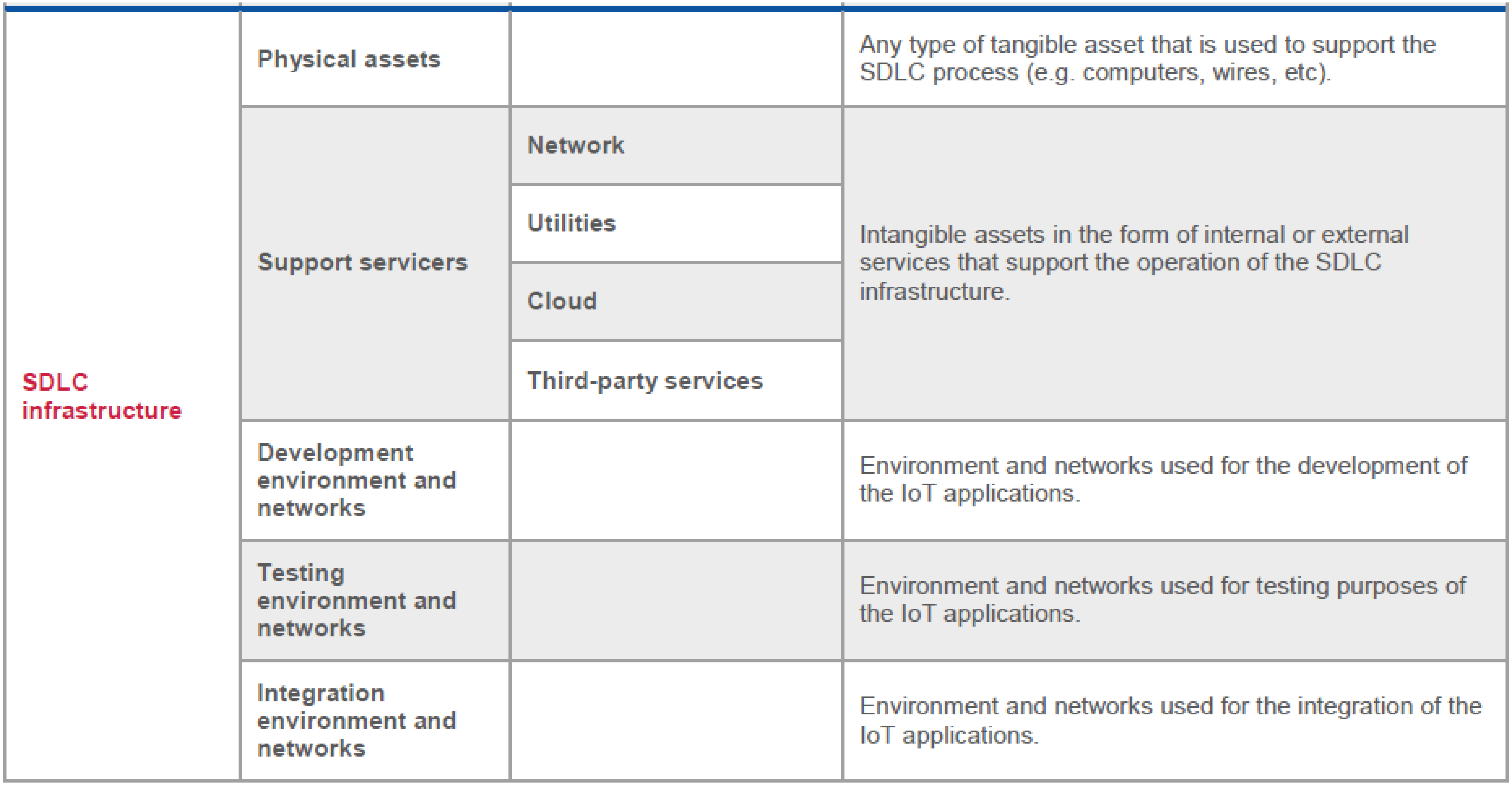
Maintenance	Updates	Over-the-Air (OTA) update mechanism	Mechanism to update hardware remotely with new settings, software or firmware.
		Firmware	Software that sets the lowest-level logic to control a device's electronic circuits.
		Software	Minor software modifications deployed that provide security or functionality error fix.
		Back-end servers	Software component that provide functionality for other programs such as sharing data or resources
	Support/ Ticketing system		Software designed to organise and distribute incoming customer service requests.
	Monitoring tools / SIEM		Monitoring tools used to continuously keep track of the status of the system in use, in order to ensure the earliest warning of failures, defects or problems, and to improve them. Monitoring tools span from servers, networks, and databases, to security, performance, end-devices and applications.
	Threat Intelligence services		Threat Intelligence Services generate, aggregate and distribute real-time feeds of intelligence data generated and derived from the use of IoT.
	Documentation	Source Code	Written text or illustration that accompanies Software and explain how operates or how to use it. Different types of documentation exist such as that for source code, change management, etc.
		Change management	
		Disaster recovery	
		Third-party documentation	
	Data loss prevention		The practice used by organisations to detect and prevent breaches, leakages, or the undesired destruction of sensitive data. Also used for regulatory compliance. An example would a ransomware attack. DLP focuses on preventing illicit transfers of data outside of the organisation.
	Version control		Management of the different changes made to the elements of a product or its configuration.

Software Components



Software components	Code	Open-source code	Software readily available for users to build and distribute new solutions.
		COTS	Commercial-off-the-shelf: software and services are built and delivered usually from a third party vendor. COTS can be purchased, leased or even licensed to the general public.
		Own software	Software developed and maintained by the own company.
		Third-party libraries	Software not developed or maintained by the company, but they are part of an application or system of the company
	APIs	Own	Application Programing Interface: a set of subroutine definitions, communication protocols, and tools offered for one library to be used by other software
		Third-party	
	PKI		Technology that is used for authenticating users and devices in the IoT ecosystem.
	Communication protocols		Formal descriptions of digital message formats and rules that allow two or more entities of a communications system to transmit information.
	Identity and Access Management		A framework of business processes, policies and technologies that facilitates management access control.
	Algorithms/Logic		A set of unambiguous specifications for performing calculation, data processing, automated reasoning, and other tasks.
	Database management systems	Relational (SQL)	Software packages designed to define, manipulate, retrieve and manage data in a database
		Non-relational (NoSQL)	

SDLC Infrastructure



SDLC infrastructure	Physical assets		Any type of tangible asset that is used to support the SDLC process (e.g. computers, wires, etc).
	Support servicers	Network	Intangible assets in the form of internal or external services that support the operation of the SDLC infrastructure.
		Utilities	
		Cloud	
		Third-party services	
	Development environment and networks		Environment and networks used for the development of the IoT applications.
	Testing environment and networks		Environment and networks used for testing purposes of the IoT applications.
	Integration environment and networks		Environment and networks used for the integration of the IoT applications.