Towards an Open, Secure, Decentralized and Coordinated Fog-to-Cloud Management Ecosystem

D2.4 Security/Privacy Requirements and Features

Project Number	**730929**
Start Date	**01/01/2017**
Duration	**36 months**
Topic	**ICT-06-2016 - Cloud Computing**

Work Package	**WP2, Technology survey, business models and architectural definition**
Due Date:	*M4*
Submission Date:	*02/05/2017*
Version:	*1.0*
Status	*Final*
Author(s):	*Jens Jensen, Cheney Ketley, Shirley Crompton (STFC)* *Antonio Salis (TISCALI)* *Matja Cankar, Miha Stopar, Jolanda Modic (XLAB)* *Eva Marín Tordera, Sarang Kahvaza, Sergi Sánchez López (UPC)* *Rosa M. Badia (BSC)*
Reviewer(s)	*Andrea Bartoli, Laura Val (WOS)* *MEB (SIXSQ)*

Project co-funded by the European Commission within the H2020 Programme		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	10.04.2017	Unedited version captured from the cloud	Ketley, Crompton, Jensen, Salis, Cankar, Stopar, Modic, Marín Tordera, Kahvaza, Sánchez López, Badia
0.2	11.04.2017	Basic edits completed, some restructuring and additional content in the threats section.	J Jensen
0.3	12.04.2017	Much restructuring and connecting content; deleting or filling in empty sections; executive summary, conclusion. Sent for internal review.	J Jensen
0.4	20.04.2017	Addressing most comments from 1st reviewer (section numbers remain). Sent to WP2 for general comments.	MEB (rev.), J Jensen (ed)
0.5	28.04.2017	Addressing comments from 2nd round of internal review, viz. UPC, XLAB, Tiscali, WOS, STFC, TUBS. Moved and fixed refs. Added components description.	Marín Tordera , Stopar, Salis, Val, Crompton, Jukan (reviewers), J Jensen (ed)

Table of Contents

List of figures

List of tables

_Toc472584842

_Toc472584842

Executive Summary

While still in its infancy, current IoT deployments often have security weaknesses that have been exploited, from “hackers” who are exploring the system to malicious cybercriminals. Moreover, these exploits have been covered widely in the press and tends to give IoT a bad name.

It is clear that mF2C must do better: without a comprehensive cross-infrastructure approach to security, the outcome of the project will see little practical use and have little chance of surviving beyond the end of the project. As has become (good) practice in FP7 and H2020 projects (having learned from experiences in earlier projects), security is designed in from the proposal, rather than added as an afterthought.

This document describes the background (privacy, data protection, protocols and cryptography) behind securing a distributed infrastructure. We then look at the implications for mF2C, from the point of view of the architecture as it is currently understood, the software components and use cases. No single security “solution” fits all applications – or devices – or budgets – so it must be possible to select the right level and enforce it across the infrastructure. For each such application it is necessary to understand the threats, and the associated risks to the infrastructure. In many situations, the weaker link is the end user, so usability is important, as is the motivation to implement security, which is it not just seen as a hurdle, but an essential part of the service. User protection starts with privacy and enables users to control and monitor how their data is used; with better transparency, users should feel empowered rather than forced to share their data.

This deliverable is submitted at the end of month 4 of the project, so it is necessarily early days – many technical things may change. Nevertheless, its scope is to set out security for the rest of the project, in order that future evaluations and plans are built on it.

1. Introduction

This document serves two main purposes. First, it introduces components of mF2C IoT security, by outlining the requirements and the wider landscape in which mF2C evolves. It defines some general IT security practices and frameworks and secure software development methodologies, and also looks specifically at their applications to IoT and the expected development in mF2C in particular.

Secondly, it also casts its net more widely by surveying other relevant developments in IoT security; these may not be immediately useful for mF2C but are intended to serve as input to future versions of this deliverable.

Some issues may depend on future choices: they may rely on the choice of implementation of a protocol, or they may depend on the results of tests. In this case, we have highlighted these issues but merely as "placeholders" for future work.

This document is structured into seven main sections:

- This section (1) gives the Introduction and describes the aim of this deliverable.
- Section 2 covers the background information, the constraints and requirements into which mF2C fits, as well as general technical background.
- Section 3 covers the security requirements based on the architecture as it is currently understood. It also looks at the (proposed) infrastructure, aiming to identify security goals and the associated measures of success (and KPIs, if relevant.) In particular, this section contains a proposed security framework which aligns with the architecture.
- Section 4: Many mF2C partners have existing (or emerging) software or hardware components or technologies, which they wish to (re)use for mF2C. Section 4 briefly outlines these software components and lists the existing security implementation for the component and what, if anything, might be missing in order to integrate into a secure mF2C implementation.
- Section 5 covers the Use Cases as they are currently understood.
- Section 6 describes the challenges and goals of mF2C security, particularly in going beyond current IoT and cloud SoTA.
- Section 7 aims for a "todo/next steps" view of mF2C security, highlighting opportunities for innovation, summarising the challenges to be addressed and not least the tests and/or simulations that need to be run, and recommendations for best practices as seen through the current plans for mF2C.

1.1. Purpose

The objectives of this deliverable are:

- Identify background security requirements for mF2C, based on the principle of secure development and operations; with a specific focus on privacy.
    - In particular, identify current best practices and relevant organisations and projects, in order to not duplicate efforts.
- Provide background information
    - for mF2C developers, on best practices for development of secure-by-design software;
    - for mF2C operators, on best practices of operations of secured infrastructure and

interfacing it to the outside world;

- Analyse the proposed Architecture and Use Cases, in order to identify specific security requirements and propose solutions.
- Identify potential future directions for mF2C.

2.2. Glossary of Acronyms

Some of the more common abbreviations like "US" for United States or "IBM" are not included here; some abbreviations that are used only in one location in the deliverable ("PBC") and/or whose expansion is not relevant to the understanding ("SIM card", "QR code") are also not included. When appropriate, the context is explained in parentheses.

Acronym	Definition
ACL	Access Control List
API	Application Programming Interface (software engineering)
APT	Advanced Persistent Threat (section 2.1)
ARM	www.arm.com
AUP	Acceptable Use Policy
BLE	Bluetooth Low Energy
CC	Common Criteria
CSA	Cloud Security Alliance
CSIRT	Computer Security Incident Response Team
CSP	Cloud Service Provider
DDoS	Distributed Denial of Service [attack]
DNS	Domain Name System
DoS	Denial of Service [attack]
EAL	Evaluation Assurance Level
FP	Framework Programme (EU)
GDPR	General Data Protection Regulation (EU)
GFD	Grid Forum Document, see OGF
GRE	Generic Routing Encapsulation
ICO	Information Commissioner's Office (UK)
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers (www.ieee.org)
IGTF	Interoperable Global Trust Federation (www.igtf.net)
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
ISMS	Information Security Management System
ISO	International Standardization Organization
ISP	Internet Service Provider
IT	Information Technology
LPT	Licensed Penetration Tester
LTE	Long Term Evolution [phone data communication] (4G)
M2M	Machine-to-machine [communications]
MAC	Media Access Control [address] (networking, IEEE 802), Message Authentication Code (cryptography)
MITM	Man-in-the-middle [attack] (cryptography)
NIST	National Institute of Standards and Technology (US Dept of Commerce)

NP	Non-Polynomial (computational complexity class of problems currently considered computationally intractable)
OGF	Open Grid Forum, www.ogf.org
PoC	Proof of Concept
PP	Protection Profile
QoP	Quality of Protection (QoS for security SLOs)
QoS	Quality of Service
RFC	Request For Comment (www.rfc-editor.org)
RSA	Rivest-Shamir-Adleman (cryptosystem)
RTOS	Real-Time Operating System
SAR	Security Assurance Requirements
SCADA	Supervisory Control And Data Acquisition
SDN	Software Defined Networking
SFR	Security Functional Requirements
SHDSL	Symmetrical High-speed Digital Subscriber Line
SIEM	Security Information and Event Management
SLA	Service Level Agreement
SLO	Service Level Objective
SME	Small or Medium sized Enterprise
SSAE	Statement on Standards for Attestation Engagements (American Institute of Certified Public Accountants)
ST	Security Target
SON	Self-Organising Network (section 3.3.2)
SQL	Structured Query Language (databases)
SSL	Secure Sockets Layer
TLS	Transport Layer Security (RFCs 2246, 4346, 5246)
TPM	Trusted Platform Module
UC	Use Case
USB	Universal Serial Bus
VM	Virtual Machine
WG	Working Group
WS	Web Services

Table 1. Acronyms and abbreviations

2. Background and General Requirements

This section lists the "background" constraints and requirements, i.e. those describing the scenarios into which mF2C must fit, mainly in the areas of: threats, security and privacy requirements, standards for security and privacy, legal constraints, and technology. The main focus is on Layer 2, "the edge" – Layers 1 (smart agents) and 0 (cloud) tend to use more established technologies, such as IP networking, or to have been studied extensively (e.g. privacy of data in clouds) already. However, we also obviously need to study how the layers interact (see section 3), e.g. that Layer 1 could monitor and gateway connections from Layer 2, or in some cases Layer 2 could connect to Layer 0.

2.1 External Threats

Not all deployments of an IoT infrastructure will have the same threats. This section covers some of the more common threats - likely to affect most IoT deployments - as well as some of the more exotic that would affect only a few.

As a starting point, we start with the threats identified by the CSA for cloud services [CSA2016]. Building on earlier work from 2010, the Top Threats WG has now identified twelve threats which we summarise here. Their relevance to mF2C lies not just in their applications to cloud but also to the other Layers; we discuss briefly their relevance to the rest of the mF2C layers and point to further work in this document.

Of course, a *threat* can turn into an *attack*.

CSA also looked at Microsoft's STRIDE as a means of categorising threats - STRIDE being:

- Spoofing (of identity)
- Tampering (of data)
- Repudiation
- Information Disclosure
- Denial of service
- Elevation of privilege

We shall use both of these. We use the CSA threats to identify the threats to an mF2C IoT infrastructure, and we use STRIDE to assess the impact of each threat. The impact of the threat can then lead to a risk assessment.
The following diagram shows at a glance CSA's mapping of threats to impact category:

Table 1: CSA's STRIDE classification of CSA cloud threats

	1. Data breaches	2. Weak credentials	3. Insecure APIs	4. System/Appl.	5. Account Hijacking	6. Malicious insiders	7. APTs	8. Data loss	9. Insufficient due diligence	10. Abuse/nefario	11. DoS	12. Shared tech issue
Spoofing		X		X	X	X			X			
Tamper		X	X	X	X	X			X			
Repudiation		X	X	X	X			X	X			

Info Leakage	X	X	X	X	X	X	X		X			X
DoS		X		X	X			X	X	X	X	
Elev. privilege		X	X	X	X		X		X			X

[CSA1] Data breaches

mF2C will without a doubt be used to handle personal or sensitive data; if we were to exclude this, we would severely limit the usability of the work, and even if we do not explicitly handle personal data, the fog could still be used to glean personal information about its participants, for example by detecting their position and movements.

Other data could include application-specific data, or sensor measurements, or other proprietary data which is distributed across the deployment but needs to be protected for intellectual property reasons.

In the GDPR (section 2.4.1), the processor could be liable for the breach.

[CSA2] Weak identity/credential/access management

Any secure distributed infrastructure requires its entities to authenticate to each other, whether using peer-to-peer credentials management or credentials issued by one or more trusted authorities. This requirement can apply to:

- automated entities like agents, which may be hosted on different physical parts,
- the physical parts of the infrastructure like the environments hosting the agents and the applications,
- “fixed” parts of the deployment, like the market place, or a particular cloud-hosted service,
- as well as the end users that make use of edge devices and their applications.

There are several aspects of this threat:

- Cryptography – that the protocols are weak due to inadequate design or poor implementation, or limitations of the device;
- Device uniqueness (each device is unique, so no device impersonates another);
- Device persistence – that each device is the same device every time it is seen.
- Personalisation – that the device belongs to a real person, whose name and identity may be verified.

[CSA3] Insecure APIs

APIs are obviously essential to connecting services together in a SOA, as well as connecting applications to the environments and frameworks in which they are hosted. Some frameworks like MQTT were designed as protocols to enable IoT devices to communicate with each other, with the assumption that the channel or environment in which they operate is secured by other means, or that security is not needed.

[CSA4] System and applications vulnerabilities

As in any piece of software, vulnerabilities can hide sometimes for a long period of time. Moreover, as the mF2C software stack ranges potentially from the cloud data centre over the Internet to smart agents, through to the fog to microagents and perhaps over a hardware protocol to sensors - there is a considerable stack to review. Moreover, some devices are dependent on firmware or libraries to aid application development, with a provenance outside of mF2C. A comprehensive security review may well not be feasible.

[CSA5] Account hijacking

In a practical deployment of mF2C, a user's account could be taken over by a malicious attacker. The hijacking can have impact not only on the person directly affected but also, through the authorisations held by this person, can have devastating influences on anything they can control.

[CSA6] Malicious insiders

This threat refers to a trusted employee - or contractor - who is acting maliciously, whether because they are disgruntled, have been bribed, or, like the spy stories of old, think they are acting correctly but are passing information to an imposter. It should be noted that public CSPs have typically done extensive vetting of its trusted personnel.

[CSA7] Advanced Persistent Threats (APTs)

APTs include a range of threats that are typically stealthy, or are trying to be stealthy, and aim to either surreptitiously steal data or to lie dormant and undetected (and perhaps spreading) until they are needed. An interesting example is the USB attack where, rather than providing a memory stick with a virus on it, the USB protocol itself has been compromised, so can attack well before the anti-virus gets a chance to run.

[CSA8] Data loss

Data loss is a risk to any device or infrastructure that deals with data: for any data that needs storage or preservation, there is a risk of loss. This deliverable does not contain a data-specific analysis of risks - beyond the very basic risk of loss of confidentiality. A more thorough analysis could be done in future work, if needed.

An edge device may need to send precious data to a higher layer (see 3.1) because it has limited storage capacity. In this case, it needs assurance that the data has been received and archived before it deletes its own copy (i.e. it essentially needs a receipt – as data integrity is important, the receipt may include a checksum.)

[CSA9] Insufficient due diligence

A system is only as strong as its weakest link, and many devices need to be patched and updated from time to time in order to remain protected against threats. One concern for IoT is that many edge

devices are released with poor drivers or weak protocols. Another major concern is that IoT devices may be less likely to be patched as they are out "in the field", the user doesn't know or doesn't care, there are no updates available, updates can't be verified, the user does not change the default password, or the infrastructure cannot afford the downtime.

Also the end user can be a weak link, e.g. where they use the same password with several different services, or they use insecure passwords. It has long been known that passwords are bad solutions to end user security; for an overview, see [Kan2015]. In general, users may be unwilling or unable to comply with strong security requirements, which imply that usability becomes important.

[CSA10] Abuse and nefarious use of services

Any infrastructure that offers services to end users can be abused; by sharing data for which it was not intended, for doing calculations that are not authorised, for sending or receiving data which it should not access. Every infrastructure will have an AUP, and every country will have its laws against misuse of computing infrastructure (as well as specific laws on particular types of data). An infrastructure will need to monitor itself for unauthorised activity; for some types like propagation of malware it may be easier to detect, whereas it may be more difficult to monitor nefarious activity inside applications.

[CSA11] Denial of Service

Denial of Service (DoS) and Distributed DoS (DDoS) attacks have been covered extensively in the press as well as in academic research: as IoT is distributed and contains many devices, a compromised infrastructure that can be instructed to target a particular entity on the Internet can do serious damage. While DoS "attacks" can be due to poorly programmed applications or misconfigured software, the predominant case is the botnet, where a large number of devices (sometimes called "zombies") are controlled – without proper authorisation – by some person who may later direct the botnet to attack a specified address.

Another increasingly common but unrelated DoS attack is Ransomware, where an attacker has infected a person's computer and started encrypting files on the computer; once the person notices the attack, their computer is no longer working and they need to pay a ransom to the attacker in order to get the key to recover their data.

[CSA12] Shared Technology Issues

Shared technology is originally a typical cloud issue, as it denotes the risks arising from the need to isolate users from each other. However, it could also apply to IoT scenarios, where the users' applications need to co-exist on the same infrastructure and in particular share networks and bandwidth. Less capable devices may be more likely to be dedicated to a single user, but if they are not, they will also be less likely to offer the isolation features or virtualisation features that protect one user from the other.

2.1.1. STRIDE Assessment

As mentioned above, we use the STRIDE classification to assess the impact of a threat being exploited. In turn, the impact assessment should lead to a risk assessment framework (probably for a future deliverable).

Spoofing Identity (CSA threats 2, 4, 5, 6, 9)

The British anti-fraud organisation CIFAS has reported a sharp increase in identity fraud to 53% of all detected fraud cases. In 86% of cases this was carried out through online attacks [cifasfrd].

Although these statistics relate to the use of non-IoT devices as well as smartphones and tablets, it is reasonable to expect that this problem will not reduce. Identity spoofing can lead to loss of personal or sensitive data, elevation of privileges in the infrastructure, as well as damage to reputation.

The entity immediately affected is the end user, but it can also affect the service provider if the user repudiates the activities. Note that also non-human entities can be spoofed; usually this is done to fool the human user into thinking they are talking to a legitimate service, so once again the human is the victim.

Tampering with Data (CSA threats 2-6, 9)

Edge devices are particularly vulnerable to being tampered with. Not only are they physically within reach of malevolent actors it is difficult to monitor them for intrusion.

Military devices have features that make it difficult to tamper with them. However consumer devices have very little protection. Furthermore consumer devices have common equipment (e.g. the processor) installed across many brands of device and that common equipment has articles easily available on the web describing how to get root access.

The impact of this exploit varies according to what data is tampered with and where; in general it is either data owner or the entity processing the data (or relies on the results) that is impacted.

Repudiation (CSA threats 2-5, 8, 9)

Repudiation refers to the case where the customer denies having performed an action that they have done, or conversely claims to have performed an action that they haven't, or they do not deny that the action has been done or not, but deny being the user who has done it, perhaps by claiming that their account has been compromised.

The impact of this can be that users do not pay for their services, or they escape the consequences of an illegal or nefarious action. This in turn can impact on the service provider who may be affected financially or their reputation may suffer.

Information Disclosure (all CSA threats except 8, 10, 11)

A basic data security models looks at confidentiality, integrity, and availability (see next item) of data. Information should normally be available to authorised parties – these are normally the data processors but can obviously include other exceptionally authorised parties such as auditors or a court of law. Information disclosure refers to the unauthorised disclosure of information, and, as can be seen from the matrix, a lot of threats can, when exploited, lead to such a disclosure.

The consequences of such as disclosure depend on the type of data and the type of entity to whom it was released. Cryptographers model the impact from best case scenarios where an "honest but curious" individual "inadvertently" came across the information and perhaps notified relevant authorities, to a worst case scenario where organised crime, a foreign unfriendly state, or an unscrupulous competitor exploits the leaked information, leading to loss of life, severe economic loss, embarrassment, etc.

Like the data tampering, the entity impacted, and the precise impact, depends on what data is disclosed and to whom.

Denial of Service (CSA threats 2, 4, 5, 8-11)

Misconfigured edge devices have demonstrated very threatening capabilities when they become recruited into botnets.

In December 2016 the dynamic DNS supplier, Dyn, came under a heavy DDoS (Distributed Denial of Service) attack from 300,000 devices that took it offline for some time [dynorg]. The attack was under the control of a Mirai botnet and utilised devices such as webcams, DVRs, routers and printers (see Annex 5).

The entity impacted is primarily the service provider, as their services are being consumed by a botnet rather than legitimate users; but also legitimate users who rely on the services – and are unable to get them – are impacted. The consequences are typically a financial loss.

In addition to DDoS attacks, botnets are also used for spamming, sniffing traffic, keylogging, spreading new malware, installing advertisement add-ons, Google AdSense abuse, attacking IRC chat networks, manipulating online polls/games, and mass identity theft [bothoneynet].

Elevation of Privilege (CSA threats 2-5, 7, 9, 12)

As we have seen, some threats can lead to an unauthorised elevation of privilege. This elevation in turn can lead to other threats being realised, such as data leakage or nefarious use of the infrastructure – it depends on the privileges being exploited.

The process of tampering with a device is known as "rooting" i.e. to get access to the root account on Unix-like operating systems.

Once root has been achieved it is then possible to use this device as a base to get access to further devices and even to get access to the overall application as a trusted administrator account.

Some types of attack possible through rooting are as follows:

- Man-in-the-middle attack
- Impersonation attack
- Access to private key of the device
- Access to encrypted data or communication that uses the private key

In the context of mF2C project the impersonation attack is the most dangerous.

The Philips Hue is a set of devices that includes led lightbulbs that can be controlled to create mood effects plus a bridge device that connects the lightbulbs via the ZigBee protocol and mesh networking. it also connects onwards to the Internet and allows control from Amazon Echo Dot and an Apple iPhone app.

[hueoflynn] describes how to "get root" on the Philips Hue bridge device.

[hueseger] describes how to enable the hidden WiFi feature on the processor of the bridge. This could make possible "drive-by" WiFi attacks that could in principle compromise misconfigured smartphones as they pass by [hueoflynn2].

Documents describing the hidden WiFi feature are readily available from the manufacturer on its datasheet [atmeldatasheet] and as official FCC submissions [huefcc] describing its radio features. The hidden WiFi antenna is shown in the photo of the last page of this document [huefccwifi].

The entity affected by privilege elevation is generally the legitimate user, as their device can be tampered with and their data stolen; if used as a botnet or other nefarious purposes, it can also impact service providers.

2.2. Technology Constraints

This section lists technological constraints that should be taken into account during the development of the mF2C infrastructure.

2.2.1. Cryptography

As many IoT devices are resource-constrained, the cryptographic algorithms running on these devices need to be specially tailored - the paradigm is called *Lightweight Cryptography*. Lightweight Cryptography aims to provide secure and efficient end-to-end communication between low-end devices such as RFID tags, sensors, contactless smart cards, and health-care devices.

Lightweight Symmetric Key Cryptography

Block ciphers CLEFIA [CLEFIA2007] and PRESENT [PRESENT2007] have been well-studied and are ready to be used in practical systems. ECRYPT II eSTREAM [eSTREAM] announced in 2008 a portfolio of new stream ciphers from which Grain v1, MICKEY v2, and Trivium are suitable for resource-constrained devices.

Research on lightweight hash functions [Aumasson2010] is relatively new area and there is a lack of dedicated lightweight hash functions which could be adopted in real-world systems.

Lightweight Public Key Cryptography

For key distribution [Aranha2010] presented the implementation of elliptic curve cryptography for the sensor platform MICAz Mote. [Oliveira2011] presented how security in wireless sensor networks can be bootstrapped using an authenticated identity-based non-interactive protocol based on Pairing-Based Cryptography (PBC) and presented TinyPBC, an efficient implementation of PBC primitives for an 8-bit processor.

Existing Lightweight Cryptography libraries

The RELIC toolkit [RELIC] provides varies algorithms for multiple-precision integer arithmetic, bilinear maps, elliptic curves, prime and binary field arithmetic, and cryptographic protocols. It is portable to a wide variety of platforms and provides a high level of customization (inclusion of desired components, various optimizations for different platforms).

WolfSSL [wolfSSL] is a small, fast, portable implementation of TLS/SSL for embedded devices.

Crypto-avr-lib [AvrCryptoLib] is aimed to be used in IoT and contains a set of implementations of different cryptographic primitives for AVR 8-bit microcontrollers.

WiseLib [Baumgartner2010] has been written for network embedded devices. It implements elliptic curve over prime fields.

TinyECC [Liu2008] was made for running on TinyOS and it supports various elliptic curve cryptography algorithms

Implementing Lightweight Cryptography in mF2C

Zero knowledge proofs (ZKP) are cryptographic proofs of possession of a secret which do not reveal any information about the secret, or in practice reveal negligible information about the secret, even if the communication channel is open to an eavesdropper. Typically based on NP complexity problems (i.e. currently considered computationally intractable), the most common example is using RSA public/private keys, where the public key is used to send a challenge which can only be answered if the other party has the private key. However, most ZKP are computationally intensive and thus not suitable for lightweight devices; others require a large communication overhead (Hamiltonian paths in graphs, for example.)

While designing new cryptographic primitives is out of scope for the project (and wouldn't be good practice either), the library described in 5.1.3 which includes cryptographic primitives for zero-knowledge protocols is planned to be tested in IoT environment and optimized for resource-constrained devices. The potential applications of zero-knowledge protocols in IoT are discussed in 5.1.3.

2.2.2. Blockchain

Blockchain is not a lightweight cryptographic technology (particularly not when it includes the "proof

of work"), but given the current popularity of blockchains for irreversible logs[1] or distributed ledgers, and "digital contracts," someone is bound to ask how mF2C will use blockchains.

While the project should not use blockchains just for the sake of it, we should also recognise the ongoing research and developments in the field, and apply them appropriately. Indeed, CSA has a blockchain working group with which we are already engaging.

2.2.3. Trusted Computing Platform/Environment

Trusted computing is a paradigm which aims to enforce the computers and other devices to consistently behave in expected ways. This is usually achieved by hardware which either has encryption keys inaccessible from the outside or provides an isolated environment which ensures that sensitive data cannot be extracted from outside this environment.

Naturally, having the guarantee that the devices behave as expected is desired for any system. But as mentioned above, to achieve this, a specialized hardware or architecture is needed (there are software-based approaches, but are prone to attacks as demonstrated in [Castelluccia2009], [Kovah2012], and [Wurster2005]). Many devices used in IoT systems are cheap and resource-constrained and thus it might not be realistic to expect that IoT maintainers would be willing to replace them with devices which possess (expensive compared to the price of simple sensors) specialized hardware, like TPM [TPM] or even ARM TrustZone [ARM]. Furthermore, solutions like TPM have high complexity and do not scale to embedded systems [Norman2013] [Owusu2013]. There are solutions for low-end devices, like TyTAN [Brasser2015] which provides a hardware-assisted dynamic root of trust and remote attestations. However, TyTAN requires a hardware component providing features like memory access control enforcement based on the code that aims to access a data region. Furthermore, TyTAN uses the FreeRTOS operating system [FreeRTOS] with extensions that are not publicly available. Another solution for establishing a dynamic root of trust in a remote embedded low-end device is SMART [SMART2012], but again it requires hardware modifications of the existing micro-controller units.

Therefore, for a project which is not focused exclusively on the security architectures for low-end devices, it is difficult to expect to integrate the existing state-of-the-art security architectures into its platform. On the other hand software on the more advanced devices can be made more secure using ARM TrustZone [ARM] (if the device supports TrustZone). Also, recently ARM equipped new low-end Cortex-M [ARMCortex] processors with TrustZone and it is to be expected that the support for TrustZone will be extended to further (low-end) devices.

2.3. Applications and Usability

Not all applications will have equal security requirements. While it is a good principle to make services *secure by design*, the fact that some security features may make services more expensive, or more difficult to use, suggests that applications and agents - and end users - should have a means of

[1] Note that irreversible logs do not need proof of work, so that is *a priori* the more lightweight use of blockchains.

specifying the required level. For example, if users need to be equipped with hardware security tokens for high level of assurance authentication (like one time passwords), there is obviously both an extra cost and usability implications.

In other words, the requirements for application security are:

- If security comes at a cost, there should be a means for entities to specify the required security level (i.e. a QoP SLO)
  - There should be an automated (i.e. machine readable and negotiable API)
  - For any application that involves humans, there should be a means for them to manage their security setting
- Whether the entity is automated or human, the setting should be simple enough to be easy enough to use, and should be sufficiently broad to cover most of the intended uses.
- Default settings should be at a relatively high level, not with security turned off.
- For the workplace, it should be possible to configure the application security to comply with workplace IT security policies. As a simple example, compare the security settings tab in Internet Explorer. Its settings are designed to make it easy for "normal" people - who just "want the Internet to work" - to get sensible trade-offs between security settings and the Internet "working", plus with the ability for expert users to customise the levels.

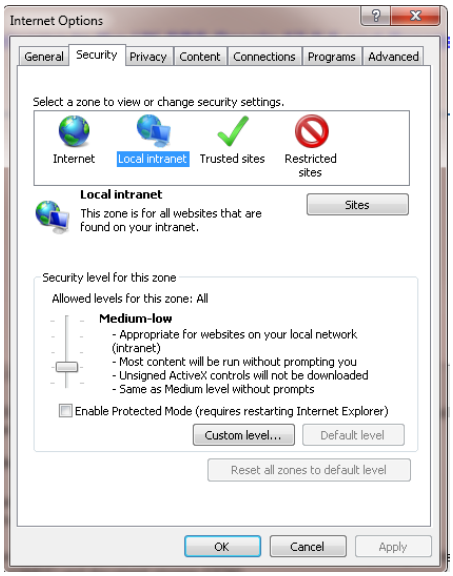

Figure 1: Internet Explorer - usable security

Also note that it comes with the distinction between the local subnet(s) and the wider Internet as a whole, which is either discovered or (in the workplace) configured by administrators.

It would make sense to run usability tests with users (sometimes called "UX" for User eXperience); it is good practice when developing security for non-technical users, because the end user, too, is a threat (see CSA threat 9).

2.4. Privacy Requirements

Privacy must be introduced from the perspective of the end user, who ultimately needs to trust the infrastructure. It is the end user who is covered by data protection legislation, and, following the GDPR (described below), user data, as well as metadata in the infrastructure pertaining to the user and their activities, are in scope.

2.4.1. Privacy, GDPR, Consent, Transparency

Privacy is an important security objective for mF2C, not just because it is a concern within IoT, but also because we need to implement the General Data Protection Regulation (GDPR, see also Annex 7).

As an overview of topics – many of which are covered in a little more detail in the following sections – the following privacy-related questions should be considered:

- How users understand and set their privacy levels; how users opt in/out of sharing.
- Privacy by design – how the system is designed to support privacy goals.
    - Safe storage – e.g. encrypted "at rest"
    - "Safe" access control lists (ACLs), e.g. default to owner-only.
    - "Safe revocation," e.g. the device is disabled (or it wipes its data) unless it is connected within a given time period to a central authority.
- Consent - how users reflect their understanding by giving consent for data processing, how can this consent be given in an IoT environment with multiple and large volumes of data sensed in different contexts over wide periods of time.
    - How consent can be targeted (e.g. allow use of medical data for research purposes but not for profit.)
    - How consent for particular purposes is recorded, e.g. the user has consented to one use, but when another use is proposed, the user has to be asked again.
        - Facilitating this process also needs to happen without violating the user's privacy.
    - How they can revoke their consent.
    - How they can see what their data has done.
    - How consent can be maintained
- Delegation of access to data
- Anonymised use of data, (safe) aggregation
    - Resistance to de-anonymization
- Society
    - Safe use of data
    - "Data leaks as the new norm"
    - Safeguards, penalties and enforcement mechanisms, particularly in an international environment, or unknown cloud locations.
        - It should be possible to see where the data goes – i.e. in which jurisdiction.
        - When applicable, it should be possible to select geographically restricted services (by region, country), as a data security requirement (e.g. data may be tagged with its restriction)
    - In particular, safeguards for individuals – could the data be misused by the state, e.g. in a repressive regime.

The significance of privacy

Privacy is considered by many, in both developed and developing countries, to be a "nice-to-have". It is willingly given up so that national security forces may better prevent terrorism. However it is far more valuable than a "nice-to-have".

Privacy allows us to protect ourselves by creating boundaries around sensitive information that may cause us harm. The information might in itself be sensitive e.g. sexuality, or the use of it might later become sensitive e.g. medical history relevant to employment – this is particularly relevant in the case of IoT where devices monitor location, activity, etc. Breaches of privacy can be embarrassing or costly, or both, so protection of sensitive data needs to be taken seriously.

Examples of privacy breaches of different scale

The following four anecdotes illustrate harm being done to an individual by organisations of different sizes - from local employer to State.

1. Some employers are known to use RFID tracking devices to track the location of staff within a warehouse e.g. for picking goods for parcel despatch. When two staff spend too much time close together they are probably talking and a supervisor is sent to investigate. In addition some warehouses employ staff on short-term contracts to avoid the restrictions of employment law, in particular unfair dismissal regulations, meaning the staff involved could lose their jobs for talking too much. Related technologies are available for parents to track their children, or their children's use of mobile phones/Internet; but could of course be misused if installed without the target's consent.

RFID devices are sold [verotrack] specifically to track staff and so improve efficiency which is a legitimate use of them but some employers abuse these devices.

2. The relationship between individuals and commercial interests is also under attack. In the USA, President Trump has repealed the privacy regulations put in place by the previous administration. Internet Service Providers (ISPs) such as Verizon, Comcast and AT&T, will no longer be bound by the privacy regulations[2]. This means they will be able to perform "deep packet inspection" on Internet traffic, monitoring their customers financial and health information so that their identity can be sold for highly-targeted advertising [wpostprivbill]. Not only is this financially rewarding for ISPs, it alters the balance of power between the individual and the businesses. Furthermore it is possible that future deregulation of the US market by the FCC would allow ISPs to rate-limit website access. The websites could, in principle, be held to ransom by the ISPs for financial or political gain [fortunepol].

3. In Britain the privacy regulator, the Information Commissioner's Office, has fined eleven major charities during 2017 for breaching the privacy of individuals to target those most likely to make more donations. The charities involved include Oxfam, Cancer Research UK, WWF-UK, and Great Ormond Street Hospital Children's Charity. The regulator found that the charities secretly pooled data from various sources and traded personal data with other charities to target new and lapsed donors. Furthermore, some charities had hired companies to profile the wealth of their targets by checking incomes, property values, lifestyles and in a few cases those most likely to be persuaded to leave money in their wills on death. Some charities shared information with charities from different parts of the charity sector [bbccharprv].

4. Many countries now log all access to the Internet. The ISP is required to keep the logs for one to two years typically. This has become law in Britain during 2016. The metadata contains information about the account holder, what websites or Internet addresses were accessed, time and location, but

[2] In the UK, the Regulation of Investigatory Powers Act (2000) also gives extensive powers to intercept and use CSP's data and metadata about their customers, the main difference being that use of intercepted data requires a warrant and is mainly restricted to public bodies such as national security, law enforcement, or similar regulatory bodies.

not what content was accessed. [bbcinvpow] [isprev]

The danger is of course that the state has to be trustworthy. In Turkey following the failed coup attempt in 2016, literally overnight a considerable number of people were reassessed as being threats to the State and were imprisoned. [polrevturkey] Thousands have been imprisoned, including 192 journalists as of January 2017 [wikipjrn], while 5583 academics have been dismissed from their jobs, their passports cancelled, with a lifetime ban on public sector employment and eviction from public housing [concernaca].

Another situation that has developed in Turkey is an acceleration towards using a full suite of mass surveillance tools. These tools have been used to de-anonymise individuals on a large scale by harvesting passwords for websites that do not use SSL to protect them in transit, then attempt to reuse the password for other sites that are more sensitive such as an email provider [forbprocera].

The impact of IoT devices on privacy

IoT devices open up new ways to gather information. They can accelerate the invasion of privacy. The following table shows some examples. [Zieg2014]. (See also Annex 7)

Table 2: IoT privacy threats

Threat	Explanation
Identification	Associating a persistent identifier with someone in some context that violates privacy.
Tracking	Determining someone's position in time and space e.g. smartphone location
Profiling	Compiling information dossiers about someone to correlate with other information sources to infer private information.
Privacy-violating interaction and presentation	Disclosing private information to an unintended audience when conveying through a public medium e.g. someone observes video over the shoulder of another
Lifecycle transitions	A device that discloses private information when it changes its control state e.g. sensitive photos found on used smartphones.
Inventory attacks	The collection of information about personal things e.g. possession of a medicine.
Linkage	Combining data sources to infer personal data that might or might not be accurate

The following is a list of the top 10 privacy risks according to the owasp privacy project [owasp10priv].

- P1 Web Application Vulnerabilities

- P2 Operator-sided Data Leakage
- P3 Insufficient Data Breach Response
- P4 Insufficient Deletion of personal data
- P5 Non-transparent Policies, Terms and Conditions
- P6 Collection of data not required for the primary purpose
- P7 Sharing of data with third party
- P8 Outdated personal data
- P9 Missing or Insufficient Session Expiration
- P10 Insecure Data Transfer

Anonymisation and informed consent

Anonymisation is often used to achieve privacy compliance and so make data processing possible. However the data can sometimes be de-anonymised by correlating with other datasets that reveal the identity. Large amounts of data from numerous users/devices make this process more feasible. Differential privacy may be an option.

It is possible to violate the rights of an entire group of people without knowing the identity of any one individual, e.g. through discrimination against any member of a racial or religious group.

There is also a risk to individuals in scenarios where they do not have much choice. A person may be likely to accept some IoT-based invasion of privacy demanded by the employer, e.g. the forced installation of an app on the individual's smartphone to use GPS for location versus time tracking. A similar case arises with insurance: the user will pay a high insurance premium for their car insurance unless they agree to have their driving (including location) monitored by a device. With the increased use of "smart" meters and devices, users may be financially penalised for guarding their privacy. One should also be aware of regulations, strengthened by GDPR, that recognise situations where users are forced to give their consent.

One option that has been explored for personal data – data "about me" and data "by me" – is to tag data with sufficient metadata that should then be honoured by the processors [Kirkham2003].

Automation of privacy protection

In order to implement privacy features for the end users of mF2C services, policies should be defined that apply to the infrastructure and enable devices to enforce privacy regulations. In order to scale, this should be automated as much as possible.

A problem with the automation of privacy protection is that it is not possible with current technology to assess what is private or not without human intervention – the most obvious solution is for users to tag their own data, but this is not always possible; the next possibility is to build in features in smart and micro agents (see also [Kirkham2003]. An additional complication is that there are multiple regulations in multiple geographic regulatory zones, so some geographic awareness may be required.

GDPR impact

The EU General Data Protection Regulation (GDPR) comes into force in 2018. It has strong protections for privacy.

The main rights for individuals under the GDPR, according to the British regulator (ICO), will be [icoprep]:

- subject access
- to have inaccuracies corrected
- to have information erased
- to prevent direct marketing
- to prevent automated decision-making and profiling
- data portability

Note that the regulator's interpretation of what needs to be done for GDPR compliance is slightly different to what is commonly thought, particularly in the area of Privacy By Design.

The main problem areas for mF2C are in the volatility of the network (some parts may be inaccessible when access is required) and the dispersal of data over a lot of systems. Another important area of concern is the use of technology that is privacy-intrusive such as facial recognition equipment or biometrics. These would require a Data Protection Impact Assessment to be performed [icopia].

See Annex 7 for an analysis of how mF2C would be affected by the GDPR.

The EU group Working Party Article 29 (WP29) has published guidelines on the following topics [euwp29]:

- Data portability
- Data Protection Officers
- Identifying a controller's or processor's lead supervisory authority

Identifying the lead supervisory authority will be difficult for mF2C, due to the distributed and heterogeneous nature of the (proposed) mF2C infrastructure and the diversity of applications – the controller may differ from one application to the next.

The ICO has identified these areas as priorities for creating regulatory guidelines [icowhen]:

- Administrative fines
- High risk processing and Data Protection Impact Assessments
- Certification
- Profiling
- Consent
- Transparency
- Notification of personal data breaches
- Tools for international transfers

mF2C must investigate all of these areas in more depth and provide tools and procedures to implement them.

Also a procedure to "onboard" individuals and organisations will be required.

Machine to machine

The GDPR has provisions regarding algorithmic privacy, intended to protect data subjects typically in insurance markets.

mF2C could conceivably have two IoT devices exchanging data containing private information.

There is a need for the data subject to be protected, but there is also a need for machine to machine (M2M) transactions to be free to operate without worries about privacy. Anonymising data takes it out of the scope of privacy regulations but it is not always possible to do that.

2.5. Relevant Standards and Best Practices

It makes sense to build as much as possible on existing work and existing standards, partly to avoid duplicating effort, partly to make use of guidance derived from other's practical experiences, but also to enable interoperation between mF2C and the world outside. The purpose of this section is to summarise the most important that are relevant to mF2C *security*. For a more general introduction to IoT standards, see D2.1, section 3.4.

2.5.1. Cloud Security Alliance

As the name suggests, the CSA brings together the cloud services industry in order to develop and promote best practices for cloud services - several of which may be relevant to mF2C's use of cloud. However, CSA also has an IoT working group [CSA] whose main focus is to document the use cases and develop guidance for securing IoT.

2.5.2. ENISA

The European Union Agency for Network and Information Security (ENISA) provides guidance for IT security across a wide range of topics. ENISA's working groups deliver reports on the current state and best practices in their fields; in IoT the focus areas are smart cars, homes, airports, and cities.

2.5.3. NIST

The US National Institute of Standards and Technology (NIST) is an entity under the US Department of Commerce which works on the implementation of US federal IT security requirements, as well as the standards and interoperation required for commerce and e-commerce. While not technically a standards body, it has developed a wide range of documents that support IT security over a wide range of activities, including cloud. As an example, NIST SP800-53 defines security controls whereas SP800-39 defined information security risk.

2.5.4. OGF

The Open Grid Forum is formally the home of the Interoperable Global Trust Federation (IGTF) which has defined technology-agnostic profiles for levels of assurance for large scale academic collaborations as well as best practices for running security infrastructure services.

OGF also provides the WS-Agreement and WS-AgreementNegotiation profiles which have been used by several European projects (e.g. SLA@SOI, OPTIMIS, Contrail) to implement automated selection of services based on advertised security features and automated negotiation of security SLOs (vs cost, typically).

Potentially also of interest might be the NSI WG which has looked previously at routing information via peer nodes, although they have not been particularly interested in the security aspects beyond basic authentication and authorisation.

2.5.5. OASIS

OASIS provides security guidance in several areas relevant - or potentially relevant - to the mF2C security design and implementation:

- Privacy and identity management,
- IoT and M-to-M (machine to machine),
- Emergency management

Other areas may also be of interest, depending on the precise requirements of the mF2C use cases; these can include the Cloud, Security, Healthcare, or Web Service (including service orchestration).

2.5.6. ISO/IEC JTC1 SC27

Subcommittee 27 is the IT security group which also has liaison to SC38 (cloud and distributed computing, see D2.1 section 3.4.1) as well as WG10 (IoT, see D2.1, section 3.4.2.) See section 2.6 and Annex 8.

See also Annex 3 List of IoT security frameworks.

2.6. Security Culture

Today nearly every business activity is supported by electronic processing of data, so data is the most valuable company asset. The information Security and related Privacy aspects is an argument that too often has been perceived and managed as secondary and disjoint in respect of primary company business.

Despite the need to protect critical data assets, ensuring its confidentiality, integrity and availability, the countermeasure typically put in place by companies is a limited collection of procedures with some firewall, antivirus and some access control.

Recurrent studies highlight that major threats suffered by SMEs are clustered in the following:

- Traditional IT-security, including network/systems security, malware and spam management,
- Physical security, including fire/water/smoke damages, access management and assets theft,
- End-users, including ignorance/negligence in terms of company policies and awareness on real threats.

The most common security incidents are data loss, with higher percentage of employee mistakes against malicious software, and minor cases of hardware or systems failure.

Despite all this evidence, most senior managers still focus only on technical aspects of information security, devaluing the "human factor" and forgetting about employee education, competence and awareness on information security, which helps in creating and sustaining a security culture in the company.

The real great opportunity is to consider information security, including Privacy and other applicable laws on data and systems, as a great chance of competitive advantage, aligned with business objectives.

Information Security Culture

Given this scenario, the first question is how a security culture could be created and sustained, who would lead this process, and which is the expected process to follow?

First of all, the need for a security culture has to start from the very top of every company or organization, typically with CEO involvement. He/she needs to understand the assets to be protected, the information security program, and relevance of this to reach the business objectives.

So top management is the main sponsor and continuous promoter, which fosters the information security strategy. Then the strategy must be propagated and clearly communicated at all levels of the organization. Finally moving to the operational side, the security strategy defined should be translated and integrated in the daily activities.

So to summarize, the following are the main steps to follow to establish a security culture in an organization and ensuring the effectiveness and success of the process:

1. **Establish Ground level**

   A first assessment must be performed on **people**, **processes** and **technologies**, so evaluating personnel and managers' knowledge, education and awareness about information security and its importance. Some technical testing could be performed to collect raw data about vulnerabilities, to be used to inform and motivate people and fix the issues.

2. **Educate management**

   This should be done as soon as possible so that they could contribute to the identification

of main issues and in scope definition, and the continuous support for the necessary changes in the organizations

3. Define vision and objectives

Top management must define the desired vision, the level of information security and objectives, aligned with the company business objectives, so that information security becomes a part of every process and not just a separate, often disregarded, process, for present and future plans and activities, thus considering security objectives in every product/service from the very beginning

4. Risk analysis

A detailed evaluation of the business context and environment need to be performed to determine the risks and opportunities that need to be addressed to ensure the expected outcomes, prevent or reduce undesired effects and achieve continual improvement. According to findings, a list of actions to address these risks and opportunities should be defined, implement all them and evaluate the effectiveness of these actions.

5. Create policies, procedures with roles and responsibilities

Management expresses its vision and desired direction of the organization through policies, procedures and regulations, which should be created on the basis of the required level of information security. These instructions should be endorsed and followed by the management (leading by example). All policies and procedures need to be communicated throughout the whole organization, and periodically checked for validity. Also relevant roles and responsibilities should be established, with a Security Committee in charge of evaluating the most relevant issues that can arise and suggest and track the necessary countermeasures

6. Educate personnel

The personnel should be educated about the policies, procedures and regulations. Dedicated training should be held using real world examples and personal experience. The policies should be explained with reasoning behind, so that people can understand why these should be followed.

7. Audits

Internal audits should be made regularly. Personnel's knowledge and education should be evaluated some months after security training have been completed, also raw data collection and usable **metrics** are to be used to assess the current level of security. The audit should cover aspects like all software installed in workstations cross-checking with allowed licenses. The results of the audit should be used to pin-point problem areas and

spot possible paths for improvement

8. Develop

The management and maintenance of the information security culture should be developed based on the results from the audit. User training should be tailored to meet specific security concerns; processes should be developed towards a more secure way of working, and in line with best practices like ITIL

9. Maintain

The achieved information security culture should be maintained according to the PDCA model, where you **P**lan, **D**o, **C**heck, **A**ct. Awareness campaigns, continuous education are some of the recommended ways to sustain this culture at the appropriate level.

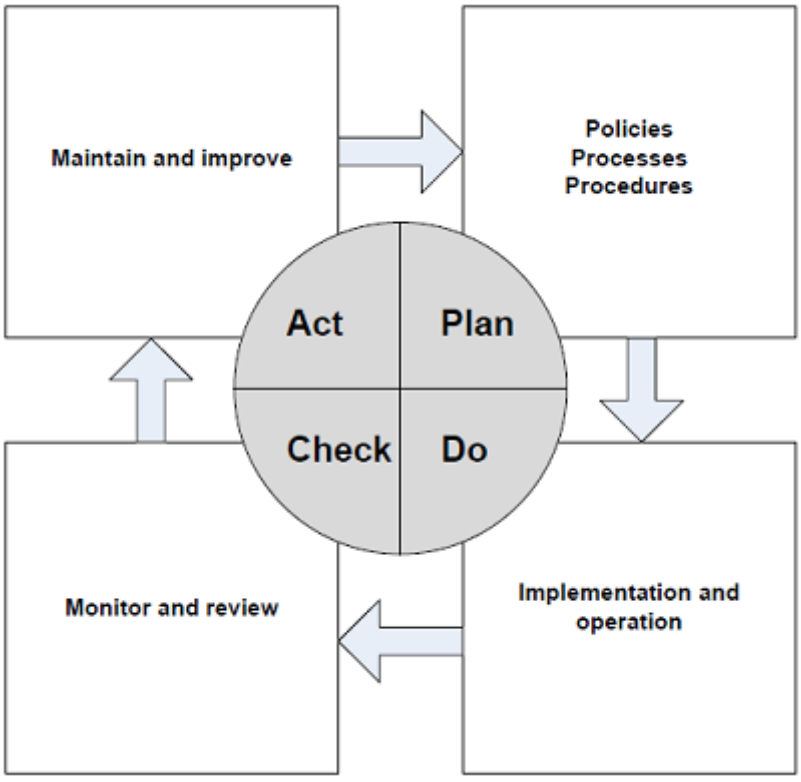

Figure 2: Security process

Information Security Management System

An Information Security Management System (ISMS) is a set of policies concerned with information security management or IT related risks. The governing principle behind an ISM is that an organization should design, implement and maintain a coherent set of policies, processes and systems to manage risks to its information assets, thus ensuring acceptable levels of information security risk.

The **ISO/IEC 27001** standard, is a well-known specification for an ISMS. The objective of the standard is to provide a model for establishing, implementing, operating, monitoring, reviewing, maintaining, and improving ISMS. The ISO/IEC 27001 standard covers management responsibility, internal audits and ISMS improvements; It also presents and defines a set of objectives and controls to be used when improving information security. By complying with the ISO/IEC 27001 standard the organization can ensure the security issues are being addressed in a consistent, repeatable and auditable manner. The ISO/IEC 27001 certificate reassures internal and external stakeholders that the information security issues are being managed compliant with the norm.

2.6.1. Security by Design

Security is a critical attribute of any hardware and software system, which must be considered and included in all steps of their lifecycle. It's absolutely recommended to consider a list of security requirements and best practices as implicit requirements of every development. A detailed analysis of all foreseen managed assets (data, applications, users, roles, etc.) must be performed and security controls are to be implemented accordingly. Software architects are responsible for constructing their design to adequately cover risks from both typical usage and from extreme attacks, as brute force or injection attacks, and frauds. Security architecture refers to the fundamental pillars: the application must provide controls to protect the confidentiality of information, integrity of data, and provide access to the data when it is required, and only to the authorized users. So when starting a new application or re-factoring an existing one, an architect should consider each functional feature and consider worst case scenarios, like:

- Is the process managing this feature as safe as possible?
- If I were an attacker, how would I (ab)use this feature?
- Is the feature required to be on by default? Could I limit the risks of misuse?

The best system architecture design and detailed design documents contain security discussion in each and every feature, how the risks are going to be mitigated, and what was actually done during coding. Security architecture starts on the day the business requirements are modelled, and never finishes until the last copy of your application is decommissioned. Security is a life-long process, not a one shot accident.

Security Principles

The following are the recommended security principles, mostly taken from OWASP Development Guide [SecByDesign] :

Minimize attack surface area
Every feature that is added to an application adds a certain amount of risk to the overall application. The aim for secure development is to reduce the overall risk by reducing the attack surface area. For example, a web application implements online help with a search function. The search function could be vulnerable to SQL injection attack, so the recommendation is to use a centralized data validation routine, to reduce dramatically the chance of SQL injection.

Establish secure defaults

By default, every end-user experience should be secure, and it should be up to the user reduce their security, if they are allowed to do it. For example by default passwords aging and complexity should be enabled, but users might be allowed to turn these features off, simplifying the use of the application but increasing their risk.

Principle of Least privilege
It recommends that accounts have the least amount of privilege required to perform their business processes. This includes user rights, resource permissions such as CPU limits, memory, network, file system permissions. For example if a middleware server only requires access to the network, read access to a database table and ability to write to a log, only these should abilities should be granted.

Defence in depth
This principle suggests that where one control would be reasonable, more controls that approach risks in different fashions are better. Controls, when used in depth, can make severe vulnerabilities very difficult to exploit and unlikely to occur. For example a flawed administrative interface is unlikely to be vulnerable to anonymous attack if it correctly gates access to production management networks, checks for administrative user authorization, and logs all access.

Fail securely
Applications regularly fail to process transactions for many reasons. How they fail can determine if an application is secure or not. So additional care is required.

Don't trust services
Many organizations use the processing capabilities of third party partners, who quite often have differing security policies and posture than you. It's unlikely that you can influence or control any external third party, whether they are home users or major suppliers or partners. For example a loyalty program provider provides data that is used by Internet banking, providing the number of reward points and a small list of potential redemption items. However the data should be checked to ensure that it is safe to display to end users, and that the reward points are a positive number, and not improbably large.

Separation of duties
A key fraud control is separation of duties. Certain roles have different levels of trust than normal users. In particular administrators are different to normal users, and as general rule administrators should not be users of the application. For example an administrator should be able to turn the system on or off, set password policy but should not be able to log on to the storefront as a super privileged user, such as being able to "buy" goods on behalf of other users.

Avoid security by obscurity
Security through obscurity is a weak security control, and nearly fails when it is the only control, just to say that security of key systems should not be reliant upon keeping details hidden.

Keep security simple
Attack surface area and simplicity go hand in hand. Certain software engineering fads prefer overly complex approaches to what would otherwise be relatively straightforward and simple code. Developers should avoid the use of double negatives and complex architectures when a simpler approach would be faster and simpler.

Fix security issues correctly
Once a security issue has been identified, it is important to develop a test for it, and to understand the root cause of the issue. When design patterns are used, it is likely that the security issue is widespread

amongst all code bases, so developing the right fix without introducing regressions is essential. For example, a user has found that they can see another user's balance by adjusting their cookie. The fix seems to be relatively straightforward, but as the cookie handling code is shared among all applications, a change to just one application will trickle through to all other applications. The fix must therefore be tested on all affected applications.

2.6.2. Privacy by design

Privacy by design is an approach, that can be combined with security by design, to projects that promotes privacy and data protection compliance from the start. In particular the expected goal is to embed privacy measures and privacy enhancing technologies (PETs) directly into the design of information technologies and systems.

Privacy by design is regarded as a multifaceted concept, involving various technological and organizational components, which implement privacy and data protection principles in systems and services.

The General Data Protection Regulation (GDPR, see section 2.4.1) for the first time addresses data protection by design as a legal obligation for data controllers and processors, making an explicit reference to data minimization and the possible use of pseudonymisation. On top of this, it introduces the obligation of data protection by default, going a step further into stipulating the protection of personal data as a default property of systems and services.

Taking a "privacy by design" approach is an essential tool in minimizing privacy risks and building trust. Designing projects, processes, products or systems with privacy in mind at the outset can lead to benefits which include:

- Potential problems are identified at an early stage, when addressing them will often be simpler and less costly.
- Increased awareness of privacy and data protection across an organization.
- Organizations are more likely to meet their legal obligations
- Actions are less likely to be privacy intrusive and have a negative impact on individuals.

2.6.3. Training and Skills

News reporting cyber-attacks involving the theft of data, unauthorized access or damage to commercial and critical systems continue to dominate the headlines, and this highlights the need for organizations to hire IT professionals with the highest preparation and awareness on systems vulnerabilities, the ability to cope with such attacks, on the basis of appropriate education, training and experience.

When trying to define security management complete set of required expertise, the leader in certifying information security professionals is the **I**nternet **S**ecurity **C**onsortium, with its **CISSP** certification. (ISC)2 [(ISC)2] defined a set of ten domains of information security that is known as the **Common Body of Knowledge** (CBK). Every manager and senior specialist must understand and be well versed in the following areas:

1. Access Control
2. Application Development Security
3. Business Continuity and Disaster Recovery Planning
4. Cryptography
5. Information Security Governance and Risk Management
6. Legal regulations, investigations, and compliance
7. Operations Security
8. Physical and Environmental Security
9. Security Architecture and Design
10. Telecommunications and Network Security

Competence Framework

The competitiveness of European industry is dependent on both the effective use of ICT for industrial and business processes and the knowledge, skills and competences of existing and new employees.

Some European standards have been elaborated with the aim of implementing and promoting high standards among ICT professionals/ practitioners and ICT end users.

e-Skills certification plays a crucial role as substantial effort is currently being made by the e-Skills community and involved stakeholders to establish a common European Framework for ICT e-Skills and competences in Europe. ICT skills/competence frameworks are an important prerequisite for ICT competence development and related quality assurance for recognition and transferability of qualifications.

The extensive work done in the last 10 years, brought to the following related standards:

- **CEN CWA 16458**:2012 (European ICT professional profile)
- **e-CF 3.0** (EN16234:2016) (e-Competence Framework – a common European framework for ICT professionals in all industry sectors)

This European Standard provides a reference of 40 competences as required and applied at the Information and Communication Technology (ICT) business related workplace, using a common language for competences, skills and proficiency levels that can be understood across Europe. As the first sector-specific implementation of the European Qualifications Framework (EQF), this European Standard aligns its proficiency levels to the EQF learning levels. This European Standard was created for application by:

- ICT service, user and supply organizations,
- ICT professionals, managers and human resource (HR) departments,
- vocational education institutions and training bodies including higher education,
- social partners (trade unions and employer association), professional associations, accreditation, validation and assessment bodies,
- market analysts and policy makers, and other organizations and stakeholders in public and

private sectors.

Within the 6 different groups of fields of expertise, the following information security profiles are defined:

- ICT Security Manager
- ICT Security Specialist

Security Certifications

Professional certifications are a very powerful way of proving a high level of competence, supported by practice and a follow-up for continuous improvement programme.

Most popular and considered certification schemes are based on ISO/IEC 17024 "General requirements for bodies operating certification of persons" standard, which guarantee a reliable, repeatable, internationally recognized, evaluation process of certified persons.

When speaking about information security oriented certifications we need to make a distinction between them, as they can be grouped according to the kind of competence they certify.

Basically there are the following main fields:

- **Organization based certifications**, they include IT governance, business continuity management, IT service management, or a mix of organization and technology
- **Technology based certifications**, that can be split into the following:
  - Vendor specific,
  - Vendor neutral
- **Product based certifications**

Certifications are often useful because they provide a means of comparing one participant to another, and they provide a means of communicating a security assurance level concisely. For example, in providing services for bioinformatics, the customer generally does not understand security, and does not want to understand security – they just want to know the system is secure, and use it.

Annex 8 lists some of the most relevant security classifications.

2.6.4. Proactive Mentality

Since the increasing attack surface areas and the relevant number of undisclosed vulnerabilities that hackers can exploit to adversely affect data, systems and networks, it's highly recommended to put in place a preventive and proactive defensive approach that leverage on:

- Organizational security (IT security governance)
- Operational security (continuous monitoring)

The IT Security Governance (sometimes called **GRC** – **G**overnance, **R**isk, **C**ompliance) aims at setting up

a team devoted to information security, that promote and lead the protection of the infrastructure with a risk based approach. The high level approach works matching the following:

- Risk analysis and treatment
- Operational controls
- Periodic Internal audits and vulnerability assessments

Where evidences collected from the risk analysis should match with operational security metrics and findings from internal audits, with a treatment that aims at continual security improvement.

At the lower level daily operational tasks must be performed with the highest attention and care to the security policies fulfilment, the prompt software updates and security patches on base software and application as needed.

Then a 24x7 Computer Security Incident Response team (**CSIRT**) need to be setup to provide a reliable and trusted single point of contact for reporting computer security incidents. CSIRT provides the means for reporting incidents and for disseminating important incident-related information. CSIRT serves to raise awareness among its customers of computer security issues, and provides information for secure protection of critical computing infrastructure and equipment against potential organized computer attacks. Organizations must share in the responsibility of coordinating their response efforts with other similar institutions. Gathering intelligence information from all sources is a critical part of information infrastructure protection, then process and correlate event data with security information and event management (**SIEM**) software products for real-time analysis of security alerts generated by network hardware and applications. Networking in a trusted environment and sharing incident information and detection and response techniques can play an important role in identifying and correcting weaknesses.

2.7. Legal Constraints

mF2C is anticipated to be used across many national boundaries including outside of Europe. It is therefore necessary to take into account the local laws and regulations in force in each country. Cryptography is of particular interest to mF2C because it is integral to the security operation of the system.

There are several classes of restrictions:

- National security restrictions
    - Export restrictions vary from country to country. There is little harmonisation and the lists change without notice.
        - British export controls [ukcommctrl]
        - United States export controls [usaear]
        - Chinese export controls as seen by Britain [ukcommchin] and in original form, Chinese only [chinexpctrl]
    - restrictions on access to content (read or write)
    - restrictions on access to individuals or groups

- Criminal law e.g. fraud
- Civil law e.g. business contracts
- Non-security regulatory e.g. medical data
- Security regulatory e.g. GDPR (see section 2.4.1)

We are interested in two categories

- data being moved by users e.g. banned material, banned sites, software considered to be munitions
- the mechanisms used by mF2C e.g. cryptography both import and export

For example, in China hardware and software products with encryption as their core function are regulated by the Office of State Commercial Encryption Administration (OSCCA) and their use has to be approved. Encrypted phones and faxes require approval for their import and so encrypted instant messaging clients are likely to as well. This impacts on mF2C by setting a precedent as to what must be licensed before use [oscca]. Reusing existing licensed components should be alright but importing new encryption software or using end-to-end encryption could be problematic.

A partial list of encryption import restrictions by country [wikipimpcrypt]

In general, countries experiencing political instability or have repressive regimes are likely to ban cryptography.

Many countries have laws requiring decryption by the user, on demand, by law enforcement or national security forces. This impacts on mF2C, particularly in a decentralised model where the encrypting side may not be accessible at that time. This is a very large topic and will not be covered fully here.

2.7.1. Geo-location impacts

There are several types of regulatory regimes that overlap. However the detail of the regulations varies from country to country.

- Privacy (e.g. GDPR, EU-US Privacy Shield)
- Business regulatory (eg medical, financial, Part11, Sarbanes-Oxley)
- National security (eg crypto)

On privacy alone each country has different laws. [privpolctry]

Internally to the USA there is no overall privacy law. Each state has its own laws and some federal organisations have incorporated privacy rules into their regulatory frameworks e.g. HIPAA. [wikipusprv]

The Safe Harbour Framework was introduced in the US to allow easier commercial interactions with Europe which has a strong privacy framework. But the safe harbour framework collapsed when the EU Court of Justice ruled it invalid following the revelations by Edward Snowden. It was later replaced by

Privacy Shield. [ecprivshld]

2.7.2. Audit and audit trail

To achieve regulatory compliance for several quite different regulatory regimes it will be necessary for mF2C to create some kind of audit trail of activities.

The audit trail will require a human and machine audit process with auditors to interpret it.

Note that it will be compulsory for private data within the scope of GDPR to be demonstrated to be in compliance unless it is anonymised. Early encryption is considered to be adequate as anonymisation.

2.7.3. Civil discovery in the legal arena

Civil discovery is known as Disclosure in the UK legal system. Both forms are often used in complex civil litigation cases.

In US law civil discovery is wide-ranging because it is not intended to be restricted to finding information that is relevant but instead is an exploration to find information that might be relevant. Large corporations in the US frequently have their email and other systems trawled. It is quite common for "Do not delete" instructions to be issued by management every few months. mF2C may find that it has data or documents (e.g. scans of paper documents) in its possession. These would have to be identified, retained and then forwarded.

The use of discovery can be "gamed" to put a large financial burden on an opponent by requesting thousands of documents of little relevance to the case. This could impact mF2C since, although a lot of the process can be automated, the resources requested might be scattered across a large number of devices. This may seem a contrived example, but if the purpose is to "game" the legal process it would be plausible.

2.8. Vulnerability Management

The growth of the number, distribution and heterogeneity of connected devices, has been increasing the threats, with an emerging new era of cybercrime that pose new questions in terms of information security.

New vulnerabilities are discovered each day and the speed at which these new threats are created makes securing critical assets even trickier.

The solution is to quickly immunize the infrastructure from these threats by eliminating their foundation: vulnerabilities. A vulnerability can be defined as a defect or bug that allows an external entity to directly or indirectly influence the availability, reliability, confidentiality or integrity of a system, application, or data.

New vulnerabilities appear daily because of software flaws, faulty configuration of applications, and human error. When discovered, these can be exploited, resulting in erratic program behaviour, illicit

network entry, privacy violations, and interrupted business operations.

What is needed is a strategy that could address vulnerability exposure, elimination and control in a systematic way. This strategy requires vulnerability management practices.

A Vulnerability Management strategy has to consider the different vectors (Network, web, mobile, wireless, endpoint) that are currently used, combined together, by attackers.

So a successful vulnerability management process is based on the following objectives:

1. Discover and categorize your assets

In order to manage vulnerabilities, you must understand what assets you have in your network and then test to find any vulnerabilities that exist. This is done by creating and continuously maintaining a database of all IP devices attached to your network. Scanning is most often done by focusing on a particular IP or range of addresses, therefore, organizing your database by IPs is more effective.

2. Identify assets based on business risk

Now that you have a big-picture view of your assets, where they reside, and how they are categorized, it's time to prioritize. It is important to isolate critical assets that have a direct impact on business risk — such as a database that contains social security numbers or credit card information.

3. Scan for vulnerabilities

Scanning is the foundational process for finding and fixing web and network vulnerabilities. Traditional vulnerability scanners are isolated from each other, each collecting their own set of vulnerabilities, resulting in a data overload. Scan results should be consolidated and normalized into a unified repository.

4. Prioritize vulnerabilities

Traditional vulnerability management solutions often produce thousands of "high severity" vulnerabilities for the operations staff to remediate. This scan data overload leads to confusing priorities and complicates remediation efforts. Prioritization based on previously defined critical assets, exploit types, and business risk, among other things, can help reduce this overload.

5. Generate attack paths to high-risk assets

Attack paths reflect the ability to understand not only where the critical assets are, but also what the topography around those assets looks like considering vulnerabilities, exploits, network configurations, and potential attacker patterns. This will help define exposure points that should be locked down along with any other areas of the network that could lead an adversary to your critical data.

6. Remediate. Patch. Monitor.

As these areas have been defined, they should be shared with other constituents. Strong reporting at all levels within the organization is required for risk reporting, trending, compliance efforts, remediation efforts, and overall business risk. The data discovered by scanning, consolidating, prioritizing, and modelling attack paths should be translated into tangible remediation tasks for IT Operations through service desk tools or patch management.

7. Validate

Validation is extremely important and often overlooked. Since remediation responsibilities usually fall on a different team than information security, remediation validation is an important step for closing the loop. These validation efforts should output to a report, comparing new results with original results, to ensure the vulnerabilities have been addressed.

2.8.1. Penetration Testing

The Penetration Test is a recurrent step in the Vulnerability Management process, that aims at conducting a serial of methodical and repeatable tests, to determine the characteristics of all devices, and work through all the different web application and/or system/network vulnerabilities. In particular the **goals** of penetration testing are to determine whether and how a malicious user can gain unauthorized access to assets that affect the fundamental security of the system, files, logs, etc., and confirm that the applicable controls, such as scope, vulnerability management, methodology, and network segmentation are in place.

Annex 1 contains a list of penetration tests/processes.

Penetration Testing Methodologies list

The cornerstone of a successful penetration test is the methodology involved in devising it. The underlying methodology should help the tester by providing a systematic approach to the testing pattern. The consistency, accuracy, and efficiency of the test must be met and should be up to the mark of the testing methodology. This does not mean that the entire framework should be restrictive, however.

The following are two important types of penetration testing methodologies:

1. Proprietary methodologies

2. Open-source and public methodologies

Proprietary Methodologies There are many organizations that work on penetration testing and who offer services and certifications. These network-security organizations have their own methodologies that are kept confidential. Examples of some proprietary methodologies are:

- IBM
- Foundstone
- EC-Council Licensed Penetration Tester (LPT)

Open-Source and Public Methodologies There is a wide range of methodologies that are publicly available. Anyone can use these methodologies. The following diagram illustrates a typical methodology.

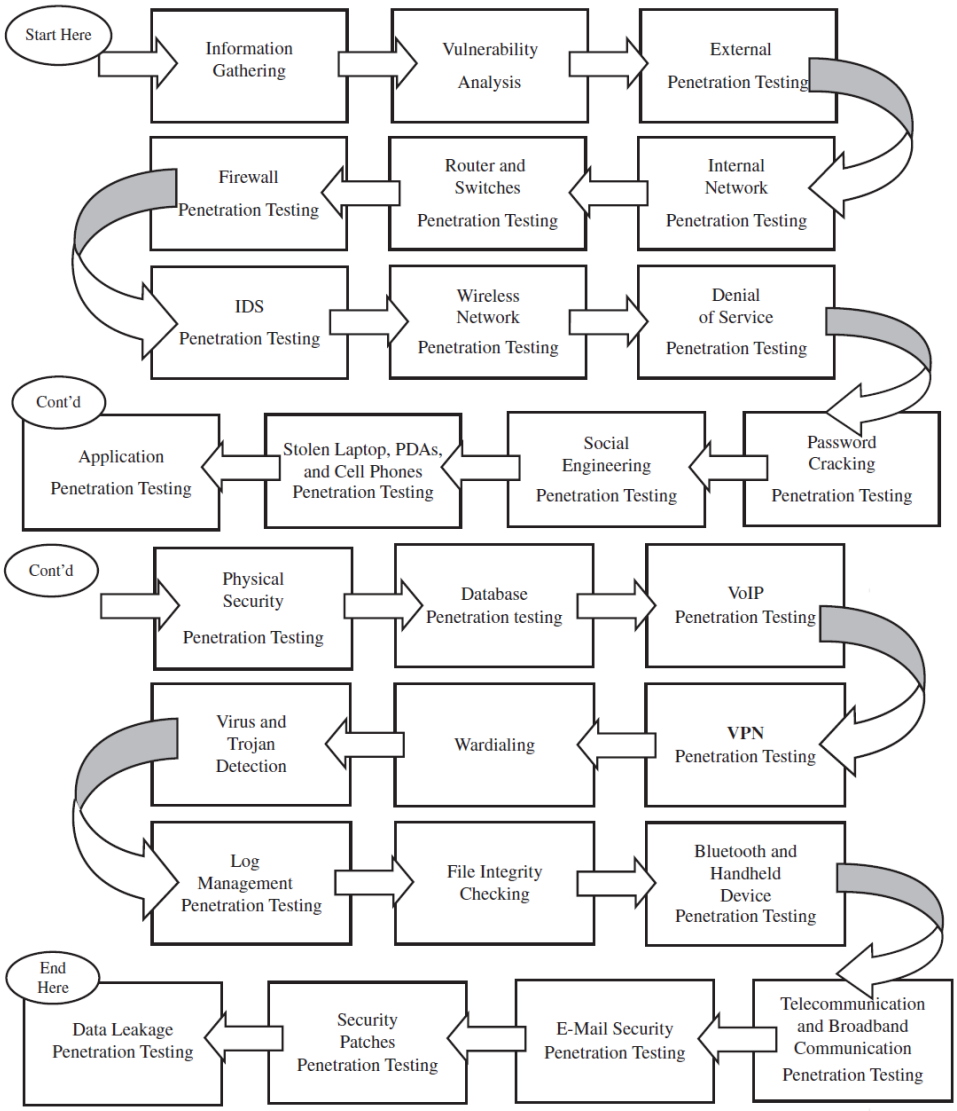

Figure 3: penetration testing

source: Penetration Testing Procedures & Methodologies, EC Council Press

The following methodologies can be accessed online:

- ***OSSTMM***: OSSTMM is the Open-Source Security Testing Methodology Manual, compiled by Pete Herzog. OSSTMM is a standard set of penetration tests to achieve security metrics. It is considered to be a de facto standard of the highest level of testing, and it ensures high

consistency and remarkable accuracy.

- ***CISSP***: CISSP is a certification program governed by the International Information Systems Security Certifications Consortium [(ISC)2]. It aims at maintaining high management-level information and network security.
- ***CISA***: The Certified Information Systems Auditor program is sponsored by ISACA and is accepted worldwide.
- ***CHECK***: This methodology tries to spot all the vulnerabilities of a system that may cause the loss of sensitive information stored on that system.
- ***OWASP***: OWASP is the Open Web Application Security Project, which is an open-source methodology. It provides a set of tools and a knowledge base, which help in protecting Web applications and services. It is beneficial for system architects, developers, vendors, consumers, and security professionals who might work on designing, developing, deploying, and testing the security of Web applications and Web services.
- ***PCI-DSS***: PCI-DSS is a security methodology defined by VISA and oriented to guarantee the security of electronic transactions made by credit cards. It includes a public penetration testing procedure to be executed on a regularly basis by all card holder processors.

3. mF2C IoT Security Requirements

This section looks at the security implications of the architecture – and requirements arising from the architecture – such as the need to connect heterogeneous devices dynamically to data centres (cloudy or otherwise), as well as requirements on the architecture arising from the other requirements (e.g. gateway devices). It is based on a snapshot of the architecture as of the end of March 2017.

The basic summary of the architecture as of the end of March 2017 is the following:

- There are essentially three layers:
    - Cloud layer, or Layer 0,
    - Layer 1 (smart agents, middle)
    - Layer 2 (microagents and edge devices, bottom layer). In turn, edge devices may interface to sensors and actuators, etc.

  The distinctions between layers may be somewhat arbitrary but, in terms of physical entities, one should think of a pyramid: lower layers contain large numbers of "micro" devices and sensors (potentially interfacing to humans), and higher layers contain few very capable data centres.
- Applications run through agents - cloud, smart, and micro - and can essentially be instantiated anywhere; however, control communication is hierarchical between layers.
- In contrast, data communication is peer-to-peer.
- As a guideline, any activity is handled by the lowest level that is capable of handling it, i.e. as close to the edge as is feasible.

Even if this is not 100% accurate, it is sufficient for the purposes of this document.

3.1. mF2C Architecture – Requirements and Attacks

The mF2C architecture handles clouds, fogs – deploying smart and micro agents – and sensors. In order to provide security to all different components in the architecture, the first action is to identify the security requirements, challenges, and issues brought by the hierarchical architecture. To that end, we highlight all security requirements on the different mF2C components individually, that is cloud, fog and edge devices [Verma2011] [Sri2010] [Ali2015] and [Morsy2016].

In addition, we attempt to provide references to the CSA threats, as described in section 2.1). For each requirements we aim to identify the threat that is being addressed, and for the attacks, the threat that is being exploited. This mapping is necessarily a bit fuzzy as the handling of the threats will depend in practice on implementations and deployments: the aim here is mainly to emphasise that requirements arise from the need to deal with the identified threats. Note that the same control can mitigate against different threats in different contexts or in different layers – "access control" protects the user, but it also protects the infrastructure. Note also that several requirements can mitigate against more than one threat. Sometimes a given type of attack can also arise from more than one threat.

1. *Cloud Security*: In the cloud side, main security requirements are:

Secure storage [CSA threats 1, 4, 6, 7, 9, 12]: All data stored at cloud must be encrypted and shared

only with authorized users.

User and device authentication and authorization [CSA threats 2, 3, 4, 5, 9]: All devices and users must be authenticated to access to the cloud, to prevent undesired information disclosure to unauthorized users.

Key management [CSA threats 2, 5, 7]: A key management mechanism to handle key distribution to users and devices is mandatory to encrypt messages and thus provide secure communication.

Identity management [CSA threats 2, 5, 10]: Users, services, servers, clouds and all entities must have a unique identity to be recognizable by the system and parties. Identity must not disclose user private information.

Policy management [CSA threats 2, 4]: well-structure policies for security provisioning must be defined by cloud.

Logging protection mechanisms [CSA threats 1, 2]: A secure password-based or other type of logging strategy needed to protect user private information.

Access control [CSA threats 1, 2, 10]: A well-secure access must be defined for users to prevent hackers and attackers to access to the infrastructure.

Trust [CSA threats 5, 6]: Cloud service providers must be trustable enough for users to store their data in the infrastructure.

Data protection [CSA threats 1, 2, 9]: All data processing, aggregation, storing must be encrypted and protected from unauthorized users.

Application programming interface security [CSA threat 3]: Software application communication can be defined by a set of protocols and standards through Internet. Cloud APIs provide all the infrastructure, platform and software service levels communication: i) Platform as a Service API provides access to the service; ii) Software as a Service provides the software application API connection with cloud, and; iii) Infrastructure as a Service provides access and management to resources such as network and VMs.

Web application security [CSA threats 3, 4]: Some critical applications, such as banking must have a high secure web quality to avoid attackers to gather any user information.

Federation of security among multi clouds [CSA threats 7, 9]: When multiple clouds are federated or some services from different clouds are needed, their security requirements must be federated

Heterogeneity [CSA threats 4, 7, 12]: When different service providers deliver a huge amount of services using different technologies, the heterogeneity problem arises, such as no security compatibility at software and hardware levels.

Integrity [CSA threats 8]: This refers to data and system integrity. Information can only be changed

in an authorized manner. Integrity provides accurate and reliable information between cloud components.

Confidentiality and privacy [CSA threats 1]: Access must be restricted to those authorized to view the data. It prevents user private information disclosure.

Availability [CSA threats 11]: This requirement means that cloud and network systems work properly without interruption, problems or possible bugs.

We revisited security requirements in the cloud. In the next step after analysing all the requirements, we discuss security attacks and threads in the cloud environment. The more critical security threads, according to [Kazim2015], are:

Backdoor channel attacks [CSA threats 4]: attackers take remote access to the compromised system. Attackers take control over victim's resource by using backdoor channel then attacker can use victims to launch zombie attack, even they can disclose private victim's information.

Malware injection [CSA threats 7]: Hackers can inject malware application, services, or virtual machines into the cloud system or datacentres to interrupt the whole system.

Virtualization attack [CSA threats 7, 12]: There are two types of virtualization attacks including VM escape and rootkit in hypervisor. In VM escape, attacker runs a program in a VM and breaks the isolation layer in order to run with hypervisor's root privileges instead with the VM privileges which allows the attacker to interact with the hypervisor. Therefore, attacker gets access to the host OS and other VMs running on the physical machine.

Rootkit in hypervisor [CSA threats 4, 7]: VM-based rootkit initiates a hypervisor compromising the existing host OS to a VM. In reality Host OS does not exist, however, the new guest OS assume that it is running as the host OS with the corresponding control over resources. Hypervisor produce a channel to execute unauthorized code into the system which attacker get control over running VM on the host machine and activities manipulation on the system.

Denial of service [CSA threats 11]: Attackers can affect the availability of the cloud and prevent legitimate users to access to cloud by jamming or flooding requests to the server.

Man in the middle attack [CSA threats 2, 3]: If a secure channel between cloud and users is broken, attackers are able to access data exchange or even datacentre.

Metadata spoofing [CSA threats 3, 4, 5]: An attacker can modify web service's description languages where descriptions of services are stored.

Malicious insider [CSA threats 6]: Person who is an employee in the cloud organization can use their privileges to disclose private information.

Phishing attack [CSA threats 3, 5]: Attackers can manipulate the web link and redirect users to a fake one to get user's private information. An attacker may use cloud services to host a phishing

attack site to hijack accounts and services of other users in cloud.

SQL injection [CSA threats 3, 4]: Attackers can inject malicious data into the SQL and get the private information or interrupt the whole SQL.

Sniffer attack [CSA threats 3, 9]: The attacker tries to read the content of a network packet, or to derive partial information (e.g. number of letters in a password).

Zombie attack (DoS/DDoS) [CSA threats 11]: Through the Internet, an attacker tries to flood the victim by sending requests from innocent hosts (normal host not the fake one) in network. There are 2 types of Zombie attack; the first is when an attacker floods a large number of requests via a zombie (innocent host) to affect availability of cloud services. The second case is when a huge number of requests overloads cloud to be exhausted which can cause DoS and DDoS attack. DDoS is a type of DoS attack where multiple compromised systems are used to target a single system causing a DoS attack.

Spoofing attack [CSA threats 3, 5]: This occurs when an attacker impersonates to be a legitimate cloud user with the intention of stealing sensitive information or launching the attack to the whole cloud system.

The cloud privilege is high computation, storage, and network therefore we have to be able to provide a high security. There are many cloud security solutions in the market that can be applied, however there are so many challenges still unsolved.

2. *Layer 1 security (smart agents)*: In its initial conception, fog should bring more privacy —as a consequence of its proximity to end-users. On the other hand, its distributed nature requires that fog computing face not only the security challenges inherited from the cloud (shifted from cloud to the edge), but some other inherent to fog computing. First, fog computing brings virtualization closer to the users, thus fog computing must also deal with security issues related to the virtualization environment as it usually happens in cloud computing. Second, recognising the distributed strategy adopted by fog computing, authentication at different levels turns into one of the main security challenges in fog – identity management of entities is the key. Indeed, the fact that fog computing can shift some computational capabilities, data analysis, data aggregation, data filtering and storage to edge devices, drives the edge of the network to handle private, sensitive or confidential information — such as, personal information. Thus, secure communications must be granted in order to guarantee data privacy at the edge of the network. Third, there is a high heterogeneity in the devices at the edge—nodes, servers, gateways, access points, etc.—, what makes the design of an architecture granting security provisioning a hard challenge. Security management in fog is a high challenge due to their distributed nature. Data security and secure communication must be applied for interconnecting fog-cloud structure. Some of the fog security requirements include [Alrawais2017] [Chiang2016] [Yi2015].

Authentication [CSA threats 2, 4]: All components, such as fog nodes, fog servers, gateways, etc. need to be authenticated. Authentication allows only authorized components to communicate and achieve data.

Privacy [CSA threats 1]: Fog user's private information must be anonymous or confidential. Confidential information can be shared only to the authorized components.

Access controls [CSA threats 4, 9, 10]: Access control must be defined in fogs components to restrict unauthorized users achieve critical information.

Data protection [CSA threats 1, 2]: All data processing, communication and storage at fog must be encrypted to be protected against attackers.

Secure gateway [CSA threats 4]: All gateways must be protected against attackers by a well-defined security strategy and protocol.

Intrusion detection [CSA threats 4]: A well-structure intrusion detection mechanism must be defined for the fog system.

Virtualization security [CSA threats 12]: Fog inherits some security challenges such as virtualization security from cloud and brings them next to the users. A security mechanism must be defined to protect from these virtualization attacks.

Identity management [CSA threats 2]: Fog users, devices, servers must have unique identities to be recognizable. A secure identity management must be defined and implemented for both Layer 2 and Layer 1 (and these may differ due to the different capabilities.)

Integrity [CSA threats 8]: It means both, data and system integrity. Information can only be changed in an authorized manner. It provides accurate and reliable information between fog components.

Confidentiality [CSA threats 1]: Access must be restricted to those authorized to view the data. It prevents user private information disclosure. It assures that only authenticated users can access information.

Availability [CSA threats 11]: All network and fog systems must be available and work properly without interruption, problems or possible bugs.

Some important attacks in fogs are:

Man in the middle [CSA threats 2, 3]: If a secure channel between fog nodes, users, and servers is broken an attacker will be able to access data exchange.

Virtualization attack [CSA threats 4, 7]: Fog inherits virtualization attack same as cloud environment due to their similar characteristics. This attack is already described in the cloud section in details.

DoS/DDoS attack [CSA threats 11]: Attackers can affect the availability of fogs and prevent legitimate users from accessing to fog servers. These attacks are described at the cloud part.

Malware injection [CSA threats 7]: Hackers can inject malware data, services, or virtual machines into the fog system to interrupt the whole system.

Gateway attack [CSA threats 3, 4]: An attacker can get control over gateways to disclose fog information, interrupt the fog system, use that gateway to launch zombie attacks, etc.

Spoofing attack [CSA threats 5]: This occurs when an attacker impersonates to be a legitimate fog device, user, or server to steal data or launch attack to the fog system.

Due to its mobility nature, one of the main security issues in fog is secure mobility and a secure handover. Although, many challenges are yet unsolved in the fog security area, unfortunately, most of the cloud security solutions cannot be applied to the fog scenarios due to their low computation and storage capabilities and high mobility.

3. *Edge security (micro agents)*: this layer includes edge devices and dumb sensors. Edge devices even with computing capacity may suffer of low computation, storage and network capabilities. Taking into account this consideration the most important security challenges, according to [Nia2016], are:

Authentication and authorization [CSA threats 2, 3]: All edge devices must be authenticated to communicate to fog, cloud and to each other.

Access control [CSA threats 4, 9]: a well-structure access policy must be defined to the edge devices.

Secure bootstrapping mechanism [CSA threats 2, 4]: A secure authenticated registration and initialization for bootstrapping edge devices must be defined.

Data security [CSA threats 1, 3]: Data must be encrypted for communication between edge devices.

Identity management [CSA threats 4]: All devices should have a unique identity which must be kept secure from unauthorized users. See section 3.3.1.

Integrity [CSA threats 8]: The meaning is the same as previously described in cloud and fog; it provides accurate and reliable information between edge devices.

Availability [CSA threats 11]: The network and edge devices must be available and properly work without interruption, problems or possible bugs.

Confidentiality [CSA threats 1]: Access must be restricted to those unauthorized to view the data. It prevents user private information disclosure. It assures only authenticated user can access information.

Some vital attacks in this layer are –see [Nia2016]–, the following

Hardware attack [CSA threat 7]: It's a malicious modification of an integrated circuit. An attacker

can access to data and software running on the integrated circuit.

Non-network side channel attack [CSA threat 7]: Edge nodes may indulge some critical information under normal operation; it causes privacy issues.

Denial of service attack [CSA threat 11]: There are 3 types of attacks: 1. Battery draining: edge nodes have a small battery with limited energy capacity. An attacker may disable battery and cause node failure. 2. Sleep deprivation: an attacker sends a set of legitimate requests to the power-battery limited energy capacity edge node to interrupt the device. 3. Outage attack: it happens when an edge node stops performing normal operations. It causes devices stop functioning.

Physical attacks/ tampering [CSA threat 7]: The attacker with a physical access may get valuable information, tamper with the circuit, modify programming and change the operating system because edge devices host in the environment where physical access may be possible.

Node replication attack [CSA threats 5, 10]: The attacker replicates node identification and enters a new malicious node to the system. It affects network performance.

Camouflage attack [CSA threats 4, 10]: Attackers hide an authorized edge node or insert a counterfeit edge node to catch, modify or redirect packets.

Corrupted/malicious node [CSA threats 4, 7]: Attackers take access to the network by corrupting a legitimate edge node or by injecting a malicious node to the system to access to other nodes.

Tracking [CSA threats 1, 2]: A fixed RFID tag has a unique identifier that can be read by nearby unauthorized readers, therefore attackers use a large number of RFID readers of this unique identifier.

Inventorying [CSA threats 1, 2]: An attacker can obtain a manufacturer code and product code and other valuable information that are attached to the RFID tags

Tag cloning [CSA threat 2]: Attackers impersonate RFID tags to get access to private information

Counterfeiting [CSA threat 2]: Attackers manipulate tags by modifying their identity.

Eavesdropping [CSA threats 1]: Attackers intercept, read, and save message for future analysis to launch more attacks.

Due to the edge devices' characteristics, cloud solutions or even fog security solutions cannot be always applied to them, and new solutions should be designed.

To summarize all security requirements and attacks at different levels, we present the tables below.

Table 3: Security requirements by architectural layer

Layer 0 security Requirements	Layer 1 security Requirements	Layer 2 security Requirements
Secure storage	Security management	Authentication and authorization
User and device authentication and authorization	Authentication and authorization	Access control
Key management	Access control	Secure bootstrapping mechanism
Identity management	Data protection	Data security
Policy management	Secure communication	Identity management
Logging protection mechanism	Secure gateway	Integrity
Access control	Intrusion detection	Availability
Trust	Virtualization security	Confidentiality and privacy
Data security and protection	Identity management	
API security	Integrity	
Web application security	Confidentiality and privacy	
Federation of security among multi clouds	Availability	
Heterogeneity		
Integrity		
Confidentiality and privacy		
Availability		

Table 4: Security attacks by architectural layer

Layer 0 security attacks	Layer 1 security attack	Layer 2 Attack
Backdoor channel attacks	Man in the middle	Hardware attack
Malware injection	Virtualization attack	Non-network side channel attack
Denial of service (DoS)	DoS/Ddos attack	DoS/DDoS attack
Man in the middle	Malware injection	Physical attack/tampering
Metadata spoofing	Spoofing attack	Node replication attack
Malicious insider	Gateway attack	Camouflage attack

Phishing attack	Most of the cloud attack can be happen in Layer 1 (Layer 1 inherits security challenges from cloud)	Corrupted/malicious node
SQL injection		Tracking node
Sniffer attack		Inventorying attack
Zombie attack (DoS/DDoS)		Tag cloning
Virtualization attack		Counterfeiting
Spoofing attack		Eavesdropping

It must be highlighted that besides the unsolved security issues from the seed cloud and fog computing models, new F2C specific security challenges will come up. Thus, proposing a solution for F2C undoubtedly requires a strong background on security aspects both in the cloud and fog scenarios. Moreover, we must consider that, although traditional cloud security protocols may theoretically provide some security to fog computing systems, the constraints on processing capacities of the edge devices undoubtedly limit the efficiency of such existing protocols. On the other hand, security initiatives designed for fog computing cannot be applied to cloud due to they are designed for edge device for limited capabilities and cannot meet the huge amount of processing and storage cloud requirements. In addition, the design of secure fogs and clouds with existing security solutions without considering the coordinated nature of F2C (interoperability, heterogeneity, etc.), may cause additional security problems when considering the whole set of resources envisioned in F2C.

3.2. Security Components

We have discussed the requirements from the architecture but not yet the components that implement the security. This, of course, could change between now and the first release, but it is our best guess at the time of writing. Devices may come with existing security frameworks (e.g. smart phones) and capabilities (TPM based devices) or without (Arduino), but mF2C will need to develop and test security components that link everything together.

We do it from the bottom up. We use the terms “controller” to essentially house the communications code/library, and “gearbox” to denote access to higher level features.

3.2.1. Layer 2:

Edge devices are generally physical devices and will generally need physical security. While the boundary to layer 1 is a bit fuzzy, we may expect edge devices to not come with a pre-existing security framework.

mF2C framework: can be **source code** to compile into a micro agent, or a **library** for devices that support this. Providing source code would be advantageous as it would enable compile time optimisations that could reduce code size: for example, we could avoid linking in a TLS library for a device which has a private network link so doesn’t need TLS, or is not capable of using TLS.

Controller library: enables a device to

- establish its identity (see section 3.3.1);
- discover the security capabilities (might be known at compile time) of:
  - itself
  - its communication networks and
  - peers;
- discover its peers and, when applicable, agents or services in higher layers, and
  - verify their security
  - discover their security capabilities
- listen for incoming connections

Gearbox: gearbox functionality may be available in this layer, or could be requested from services in Layer 1 by communicating a request.

3.2.2. Layer 1

Layer 1 devices are expected to be "smarter" and more computationally capable than Layer 2. They may come with existing security models which mF2C may wish to integrate.

mF2C framework: we expect to consist of the following:

- libraries to link into applications (so made available for each relevant platform, e.g. smart phones, Raspberry Pi, etc.)
- can optionally also be a *framework* to aid the prototyping and development of mF2C applications
- standalone agents which do not implement any use case directly, but serve to keep mF2C together – these agents would monitor services, connect Layer 2 to Layer 0 or provide gearbox features for Layer 2 (or Layer 1)

Controller: a Layer 1 controller would have the same high level functionality as a Layer 2 controller, but may of course implement more sophisticated algorithms. In addition, it would:

- enable secure communications with both Layers 0 and 2, and facilitate communications between these layers, including "smart" communications where data can be cached, monitored, filtered, sorted, receipted, etc.
- discover and update infrastructure-wide security policies
- enable a smart agent to select required security levels for a given application, data set, or communication for a given set of peers[3].
- this security level could be established by policy from "outside", e.g. mandated at compile time

[3] While service selection should be done according to the required security level, it makes sense to also optimise for cost or speed, etc.; see section 4.1.2

or by trusted Layer 0 services (in order to implement overall infrastructure security)

Gearbox security functionality would comprise services to:

- implement fog-wide security by coordinating
- evaluate actions against security policies
- orchestrate edge devices
- self-organise smart agents and/or level 1 devices – some applications may require at least one agent, at most one agent, or precisely one agent – to fulfil a particular role or task. See section 3.3.2.
- securely discover gearbox services and applications in the cloud and invoke them, discover and access data in the cloud

3.2.3. Layer 0

mF2C framework would comprise:

- "cloudy" services, instantiated on demand
- libraries (and potentially frameworks) for development of applications
- mF2C-specific applications, providing monitoring or gearbox functionality
- market place for applications and services
- policy management services for runtime policy updates
- when applicable, CASB services

Controller would provide the same features as Layer 1 (with, obviously, a cloud agent *in lieu* of a smart agent.)

Gearbox: consists of libraries and cloud agents that implement:

- Policy repository and policy administration point
- Global identity services, e.g. certification authority
- Security services such as timestamping, monitoring, centralised logging and accounting
- Cross-fog coordination, e.g. in detecting and handling security incidents. Sharing incidents information (securely) and coordinating reactions to incidents – automatically – could lessen the impact of an attack, and improve overall security in the infrastructure.

3.3. Discussion of Selected Special Requirements

The purpose of the following section is to discuss some of the more challenging requirements and highlight how they could be solved or the type of research or development that may be needed to address them.

3.3.1. Identity

In most cases, it is essential that every entity (human, agent, host, service) be able to securely authenticate itself. The background to this is the following list of requirements/features – not all of

these would be required in every context, so the choice of algorithms/protocols would need to be flexible:

Uniqueness is generally one of the fundamental requirements for any distributed infrastructure: each participating entity, whether a server, an agent, a human, etc. is assigned a unique identity with which it can represent itself to the infrastructure, and, perhaps, communicate securely with other entities.

Uniqueness, as identified in the requirements, is essential for many things, including logging, auditing, peer-to-peer communications, preventing MITM attacks, as well as detecting and reacting to malicious or compromised devices and detecting systematic errors in data.

Persistence: that it is the same entity every time that connects, thus enabling continuity and ownerships.

Traceability: that, based on the presented identity, the entity can be identified against a real-life identity (person, physical device)

Verifiability: that the identity can be verified as being that of a legitimate participant in the infrastructure, or more specifically, that it can be verified as belonging to the particular entity that is presenting it as their identity.

Secrecy: that the identity is based on knowledge that is kept secret, or is communicated only over fully trusted channels, to avoid a malicious entity snooping the identifier and impersonating the legitimate entity.

For example, uniqueness could be achieved by randomly generating an identifier (if the device has access to randomness); with N random bits, the probability of an accidental collision is $2^{-N/2}$, so could be made as small as desired (at the cost of requiring more bits.) If this identifier is stored, it would obviously provide persistence. But it would need to be kept secret, or it would be very easy to impersonate the legitimate device.

Machine identity

In the on-boarding of an entity – a new edge device, say – the device can either have a factory-set identity (à la MAC address), or it can contact a central (infrastructure-specific) authority to obtain an identity (and then remember this identity for the rest of its functional life). The former is not always possible, and the latter may be problematic if the device is participating in several infrastructures, or if the device needs to function before it can be assigned its identity.

A third option is a randomly generated identity: if an identity is specified by, say, a 64 bit integer, then the chances of *impersonating* a given entity – i.e. by guessing its integer – is obviously 2^{-64} or less than 10^{-19}. The probability of two devices generating the same identity is similarly 2^{-32}, or less than 10^{-9}.

This approach has the advantage that the probabilities can be made arbitrarily small – at the cost of adding more bits to the identity – but also requires a good source of randomness to ensure all identities are equally likely and cannot be guessed by an adversary.

The fourth option is to ignore identity altogether and live with the fact that entities cannot be uniquely identified.

Human identity

Humans, too, can be represented in the infrastructure via their devices: most people have mobile phones, and in some use cases will have personal devices, or devices that are personalised to them through a log in or the device otherwise recognising the identity of the person, typically through biometrics, or using a smart card.

In some cases it will be necessary for humans to have a legally binding representation to the system, where the representation to the infrastructure can be *traced* to the person's real-life identity, perhaps in a way which can be audited or presented as evidence in a court of law. (In the more common cloud usage scenarios, this strong uniqueness and traceability is not usually required; a link to credit card will suffice.)

3.3.2. Self-organising system

One of the mooted features of a mF2C architecture is a self-organised network (SON): In a simple example, a group of (distributed) peers could elect one to fulfil a particular role, for an application for which it is important that there is one and only one holder of that role at any given time (it could be a database that the others synchronise with, a "central" discovery service, a logging service, or it could relay information to/from the outside.) The differences to the more traditional diversified roles – where a dedicated entity is deployed to fulfil a particular purpose are:

- All peers are in principle able to fulfil the special role (e.g. they all have Gearbox functionality); but they elect to only have one use it.
- If the peer with the special role dies, the rest of the system will:
    - Discover that the role has disappeared, and
    - Rerun the election process.
- Similarly, if the peer with the special role needs to leave the distributed infrastructure, it can either just leave and let the others discover it, or it can notify the others to rerun the election process, or it can itself nominate another to take on the role.
- If a new peer joins the distributed infrastructure, it can discover who holds the role.
- If the peer with the special role becomes temporarily unavailable, and the rest of the system thinks it has died and then rerun the election process, and the original holder of the role then comes back, then, too, there is a discovery and resolution process wherein the data held by the two holders of the role is consolidated and one hands over to the other.

It follows that a SON must have

- Resource discovery and brokerage *(bootstrapping)*
- Self-organisation and evolution *(execution and evolution)*
- Fault monitoring and mitigation strategies

… and many of these are true for all distributed systems but as mentioned above, SONs pose special

challenges.
The potential advantages of SON include that they can improve application performance in a cost-effective manner by having the network set itself up and organise its own resources, rather than relying on a central authority – which may be temporarily unreachable – to sort things out. SON can be seen as an adaptive functionality where the network detects changes and based on that make decisions to overcome the effects of these changes. From other point of view, network nodes work cooperatively to response to the changes in order to achieve certain objectives. SON has ability to minimize human involvement in network processing to efficiently make planning, deployment, and maintenance activities, and can reduce operational and capital expenditure and achieve network capacity, coverage, and service quality optimization. SON merges network planning, configuration, and optimization into a unified atomization process with minimal human intervention. Some of the SON characteristics are scalability, stability, agility, the support to large-scale networks, inexpensive network deployment, simple Internet connectivity feature, time-synchronized, self-organized, self-healing, self-configured, and data transition hop-by-hop. The benefits of SON are coverage, mobility, scalability, heterogeneity, compatibility. SON is able to provide services such as internet access, video conferencing, voice communication, and facilitate the data transferring between networks [Aliu13] [Peng13] .

However, this self-organized system suffers from security challenges and issues such as [Dorri2015] [Alazani2016] [Wen2015] [Singh201] [Bayou2015] :

- SON may be more vulnerable to network and data security attacks due to the hop-by-hop and distributed structure, such as, DoS, MAC spoofing, eavesdropping, Sybil, malicious injection.
- Some security challenges and issues that must be solved in SON include availability authentication, data confidentiality, integrity, non-repudiation, secure routing protocols, security with QoS (security provisioning in SON increment packet time delivery and processing time in each node therefore providing QoS and security is highly challengeable) , and cluster-based security.
- So many security challenges are unsolved such as insecure radio channel, network structure vulnerability, node attack vulnerability (once a node is attacked due to their hop by hop characteristic bring threats to the whole system), mistrust between nodes, and routing attacks.
- After the analysis of security issues and challenges above, we can classify attacks in SON into: black hole attack, worm hole attack, byzantine attack, spoofing attack, routing attack, resource consumption attack, session hijacking, DoS, impersonation attack, modification attack, fabrication attack, man in the middle attack, gray hole attack, and traffic analysis attack.

mF2C needs to decide whether SON is worth pursuing – as mentioned above, a typical use is when there is a need for one and only one entity to hold a particular role; but there are sometimes alternative approaches that could achieve the same benefits.

3.3.3. DDoS and Botnets

As DDoS and compromised devices (such as in botnets) have garnered a lot of press attention, mF2C will need to have defences against botnets, both from within (mF2C devices are compromised, or malicious devices are connecting to mF2C) or from outside (through the Internet.) See Annex 5.

Inbound and outbound

There are two main methods for controlling botnets:

- Looking for compromised devices
- Controlling entry, routing, and exit with a gateway/router

Inbound botnet traffic will consist of botnet commands or traffic scanning for devices to compromise. The scanning traffic can come from anywhere on the Internet since it is often compromised devices that look for new devices to compromise. The botnet commands can sometimes be spotted using a sniffer such as Snort at the entry point i.e. gateway.

Outbound botnet traffic will consist of DDoS traffic usually, or traffics canning for devices to compromise. If we know a DDoS is in progress it can be blocked at the gateway.

What would be useful is a website similar to the email blacklists site so that DDoS information could be shared. Spamhaus offers a Botnet Controller List, but we need something more comprehensive [spamhbcl].

One problem with using a gateway is that we cannot always force a device to use the gateway; another is that the gateway may not be able to inspect the packet or distinguish botnet attacks from legitimate traffic (e.g. the "Slashdot effect.") Or, of course, the gateway itself may be overwhelmed. Nevertheless, blocking communications at the gateway, when one is used, should be explored as an option.

Current research is looking at machine learning to detect unusual activities. It is still an open question whether this is a useful approach. For example, if communications are encrypted, it is not possible to perform deep inspection, and a DDoS attack may be indistinguishable from very busy activity in the IoT infrastructure, e.g. a reaction to an emergency.

Operational security discussion

How can we detect external threats, do we look for signs of attack? Do we look at performance over time both network and service performance to help identify and attack.

4. External Components

In order to ensure that mF2C doesn't start from scratch, the project partners have proposed software and hardware that the project can build upon. The purpose of this section is to briefly cover these components, but highlighting existing security features (if any), gaps that may need to be filled if/when integrating these components (if known), and suggestions for improving security, if needed. If the security of the proposed component is unknown, we ask the questions here.

4.1. Software

In this section we look at existing software which may be useful in building mF2C infrastructures.

4.1.1. dataClay (BSC)

dataClay is a distributed storage platform that allows to store, share, and integrate data in a controlled and flexible way. It can be seen as an evolution of a database system, adapted to the context of multiple independent data owners and consumers, each of them having access to different subsets of the data. From the point of view of the software (either applications/services or middleware/platforms) that accesses the data, data is stored and managed following the object-oriented paradigm, that is, every piece of data is an object that may be part of another complex object, such as a collection. Treating everything as objects, applications/platforms can indistinctly deal with transient or persistent data, and access data regardless of its location within the distributed platform, since dataClay transparently deals with these lower-level data management details.

In order to store and access data in dataClay, users must be authenticated by means of a user identifier and a password. Each user has access to a set of classes and, for each class, to a subset of its attributes, methods and objects. dataClay guarantees that a user cannot access (part of) an object or execute a method for which the user is not authorized. Also, in dataClay data cannot be explored or arbitrarily queried. Instead, users can only execute methods associated to each object (and only the ones they can access), which provides a certain degree of security if they are designed to ensure that sensitive data is not returned as a result. However, several security issues regarding storage should be considered in order to provide a secure mF2C implementation.

First, data at rest as well as communications, both within dataClay and from dataClay to the application/middleware, should be encrypted to protect data from external threats, taking into account that microagents have limited resources and may not be able to perform complex computations. The existence or complexity of the encryption could depend on the sensitivity of the data.

Second, since the mF2C infrastructure has a high degree of dynamicity due to resources that join and leave, mF2C should be able to ensure that sensitive data is not sent to an insecure device. One solution to this threat could be the assignment of a trust level to each device according to its characteristics (e.g. computational capacity to encrypt/decrypt, known owner, open ports …). In this way, mF2C would be able to guarantee that sensitive data reaches only trusted devices. This could be implemented through tagging devices, data, or both, with a security level semantics. Data tagging was explored in the TAS3 project [Kirkham2003].

Finally, all the security management required for the data should be integrated into the whole mF2C identity and access control management in order to simplify both the security management in the infrastructure and the usability of mF2C by users.

4.1.2. WS-Agreement

WS-Agreement (GFD.192) and WS-Agreement Negotiation (GFD.193) are WS-* protocols so could use standard web services security, as well as basic HTTP security, so it becomes necessary to describe what the implementation(s) support.

In previous (framework 6 and 7) projects, WS-Agreement has been used to negotiate security features (e.g. SLA@SOI, Contrail, OPTIMIS). In the cloud industry today, there "Cloud Access Security Brokers" and "Cloud Security Gateways" are hot (or at least warm) topics, but as far as we are concerned, any requirement for dynamically negotiating security features – when required by the applications – has been investigated by the earlier projects[4].

4.1.3. XLAB's authenticator

The library aims to implement existing cryptographic primitives for building anonymous credential systems. While there are some technologies that realise such systems (Microsoft's U-Prove [BRA2010], IBM Idemix [IBM2012]), currently there are no open source libraries providing the aforementioned primitives and much less the libraries which were tested on the IoT infrastructures.

An example scenario where such library could be used is anonymous authorization system for cars to parking lot gates without disclosing any information other than the fact that the car is authorized to use the parking lot.

Another example is anonymous authentication for IoT devices that are sending the data to some central component where the data is being processed. Anonymous authentication implies that the data is anonymized and while the central component can process the data, it cannot determine which device corresponds to which data.

One of the main objectives is to make the library easily buildable for various platforms, including the platforms used in IoT. For this reason the library is being written in Go which supports cross compilation [GOLANG]. Another reason for using Go is its suitability for developing fast, scalable servers which is of great importance for IoT infrastructures containing huge number of devices which need to regularly authenticate and communicate to some central component. The library uses grpc [GRPC] for communication and thus clients (software deployed on IoT devices) can be written in any other programming language since grpc works (using code generation from services and types descriptors) across languages.

[4] We also note that some of the FP7 projects that have been reusing work from earlier projects found that it required substantial amounts of effort; however, the work should now have matured. Also, as the FP6/7 projects have now finished, sustainability needs to be considered.

The library aims to provide implementation of the anonymous credential systems based on zero-knowledge proofs, such as [CAM2001a] [CAM2001b] which allow anonymous yet accountable transactions between users and organizations (users can anonymously prove assertions about themselves). Furthermore, in such systems user can reveal only selected properties of the credential (for example user's date of birth). Unlike anonymous credentials systems based on blind signatures [CHA1983] (such as U-Prove), systems based on zero-knowledge proofs enable credentials that make multiple showings unlinkable and only a single credential is required for a user.

The library will support different algorithms for:

- signing
- making cryptographic commitments
- proving equality (of two committed values), inequality (for example an attribute is less than or greater than a constant), set membership (for example attribute is in the set of given values)
- verifiable encryption

One of the priorities is to make the library suitable for IoT devices. While on 8-bit (like Arduino/AVR) and 16-bit architectures the library written in Go would have issues at least due to garbage collection and limited address space, Go become more viable on large processors like ARM Cortex series. The smallest ARM processors are Cortex-M 32-bit microcontrollers (run at low clock speeds and have a small amount of system RAM) which are gaining the importance in IoT because of their low-cost and energy-efficiency (furthermore, some of the M series are already equipped with TrustZone [ARM]). As it is expected that the number of these chips that will be used in the next years could be greater than the number of ARM chips currently deployed in smartphones and tablets, the library aims to provide optimizations for ARM microcontrollers.

4.1.4. COMPSs programming model

COMPSs [servicess] is a programming framework that provides developers with a sequential, infrastructure-agnostic programming model, to ease the development of distributed, high-performing applications. Applications developed in the COMPSs programming model are automatically instrumented to invoke a runtime toolkit that splits the application into computing units (tasks), finds the data dependencies among tasks and orchestrates their parallel execution on top of a distributed platform, guaranteeing the sequential consistency of the application.

BSC has recently developed a flavour of the COMPSs runtime designed to support mobile devices, which is known as COMPSs-Mobile[compssmobile].

The execution model of this version is based on applications that begin their execution in a mobile device and are able to offload the execution of CPU intensive tasks to external nodes in a cloud.

Due to the distributed nature of this execution model, COMPSs-Mobile has been recently extended with security features, including Single Sign-On. To support a wider range of security frameworks, these mechanisms are implemented in a generic way via GSSAPI (RFC2743).

COMPSs applications are mapped into a master-worker architecture, where the master orchestrates the execution of the whole application and offloads tasks to be executed in the worker nodes, between other features. In security terms, the master (the mobile device) is the client and authenticates on behalf of the user, using a previously obtained Kerberos credential. Worker nodes that receive incoming connections must be able to authenticate themselves, in this case with a Kerberos keytab but typically with an X.509 certificate, and must be able to validate the client credential of the user.

With regard communications, COMPSs-Mobile runtime already encapsulates all the network interactions within a communication layer component; therefore this is the only component that needs to be modified to secure COMPSs-Mobile communications, and the application remains agnostic of this. To achieve secrecy, the messages are encrypted. This implies also increasing the size of the messages, which will depend on the actual size of the token selected as base to split the messages.

4.1.5. Platforms and Security Frameworks Overview

An IoT platform has come to be loosely defined as a broad range of software technologies that join together different components to enable the interaction between the Internet and the “things” i.e. edge devices. Platforms sit between the device sensors and the data networks. It allows the deployment of applications, remote data collection, secure connectivity and device or sensor management. Annex 2 provides a list of IoT platforms.

Security frameworks come with ready-made security so that developers do not move away from best practice or simply forget to put it in.

See Annex 3 for a list of IoT security frameworks.

4.2. Future Security Evaluations

We note that other than the specific hardware described under Use Cases plus a few suggestions being aired at the kick-off meeting, we have little information about the hardware and platforms that will be used to develop the mF2C infrastructure(s).

Some hardware, however, is known: we have a list of (suggested) devices from Intel, and the Nuvlabox from SixSQ; the Use Cases below suggest additional hardware. We suggest that, in addition, it will be useful to experiment with prototyping devices such as the Arduinos, and general microcontroller-based devices, as one can easily prototype devices with a specific purpose, including testing security of an IoT infrastructure. Moreover, as these devices are easy to prototype, an attacker will be able to build one, too, and will try to connect them to the fog, so it makes sense to test this scenario.

Another hardware-specific question is how applications and agents are implemented for the hardware (e.g. cross compiled) and deployed (e.g. onboarding, how does the device or agent get its identity) on to the infrastructure. Emulation of the hardware may be relevant for simulations, in particular because they will enable us to scale beyond a smallish number of devices, but emulation alone is likely to be insufficient; it will be necessary to test the actual hardware in prototype or production infrastructures.

Future documents will thus need to discuss these questions in further depth, evaluate the security of

the hardware, platforms, and CSPs that were actually used, plus look at the combined infrastructure as a whole. In addition to the IoT devices (use cases and prototyping) mentioned, these targets can include:

- Cloud platforms: OpenNebula, OpenStack,
  - Evaluations of their security depend on how they are deployed.
- Abstractions of cloud platforms: SixSQ's Slipstream
  - An abstraction helps mitigate many risks but by definition adds another layer, which in turn raises security questions.
- Cloud CSPs – the primary focus is usually the data centre; some providers such as Azure[5], Google[6], and Amazon[7] have certified their data centre using SSAE-16 or similar (there are many different types of data centre certifications) , whereas others are "in the process" of being certified, or say nothing about security in their public information. Some CSPs specialise in secure data services, such as Dropbox.
- Device platforms
  - Cisco
  - Android Things/Google Brillo
- Protocols (also see protocol description under Use Cases)
  - MQTT
  - XMPP, AMQP, AllJOyn, Weave, Co-AP, OIC, ZigBee, Z-Wave, oBix, Licas….
- Policy engines
  - A policy engine might be used to implement GearBox functionality – e.g. service orchestration, business level security policies, etc. Annex 6 provides a list of policy engines.

[5] https://www.microsoft.com/en-us/TrustCenter/Compliance/SOC

[6] https://cloud.google.com/security/compliance (accessed April 2017)

[7] https://aws.amazon.com/compliance/soc-faqs/

5. Use Cases

The definition of use cases will be finalised in WP5 and therefore this section cannot collect all aspects of the security and privacy that are attributed to each use case. However, the specifics of the devices and environment, in which they will operate, can be identified and analysed for security and privacy issues. This section will present the basic parameters of the use cases that are crucial for understanding the security and privacy requirements and potential flaws and attack surface.

5.1. UC1 – smart cities

The first UC is "owned" by WorldSensing: smart cities.

5.1.1. Identification of relevant protocols in Smart City

A number of fundamental challenges are specific to Smart Cities; addressing these challenges is the key to their unhindered development. The lack of standardized wireless solutions (i.e., Multi-hop Networks, etc.) has attracted significant academic interest in the last decade. This research has sparked the development and commercialization of a large number of proprietary solutions, which, on consequence, has resulted in technology fragmentation; an obstacle for the Smart City market in which complex heterogeneous systems of sensors and actuators need to form a homogeneous communicating entity.

Major standardization bodies are well-aware of the lack of standardized solutions, and are well underway in answering it. The IEEE802.15.4 standard, for example, has been at the forefront for providing a physical layer which offers a healthy trade-off between power consumption, communication range and data rate. Beyond this low layer, the Internet Engineering Task Force (IETF) has been adapting Internet protocols to "constrained" networks of low-power devices. IETF standardization efforts most applicable to the Smart City space are 6LoWPAN, an adaptation layer enabling even the smallest device to be IPv6-compliant and appear as regular hosts on the Internet, and RPL, a routing protocol allowing the creation of multi-hop meshes. More recently, the IETF 6TiSCH working group was created to define how to build "umbrella networks" taking advantage of the unprecedented performance of IEEE802.15.4e TSCH. However, despite the introduction of novel protocols, the main commercial solutions for Smart City applications are following introduced.

5.1.2. Commercial communication protocols for the Use Case #1

Many communication technologies are well known such as WiFi, Bluetooth, ZigBee and 2G/3G/4G cellular, but there are also several new emerging networking options such as Thread as an alternative for home automation applications, and Whitespace TV technologies being implemented in major cities for wider area IoT-based use cases. Depending on the application, factors such as range, data requirements, security and power demands and battery life will dictate the choice of one or some form of combination of technologies. In the proposed Use Case #1, the following communication protocol will be taking into account:

Bluetooth: An important short-range communications technology is of course Bluetooth, which has become very important in computing and many consumer product markets. It is expected to be a

cornerstone for wearable products in particular, again connecting to the IoT albeit probably via a smartphone in many cases. The new Bluetooth Low-Energy (BLE) – or Bluetooth Smart, as it is now branded – is a significant protocol for IoT applications. Importantly, while it offers similar range to Bluetooth it has been designed to offer significantly reduced power consumption.

Zigbee: ZigBee, like Bluetooth, has a large installed base of operation, although perhaps traditionally more in industrial settings. ZigBee PRO and ZigBee Remote Control (RF4CE), among other available ZigBee profiles, are based on the IEEE802.15.4 protocol, which is an industry-standard wireless networking technology operating at 2.4GHz targeting applications that require relatively infrequent data exchanges at low data-rates over a restricted area and within a 100m range such as in a home or building.

WiFi: WiFi connectivity is often an obvious choice for many developers, especially given the pervasiveness of WiFi within the home environment within LANs. It requires little further explanation except to state the obvious that clearly there is a wide existing infrastructure as well as offering fast data transfer and the ability to handle high quantities of data.

Cellular: Any IoT application that requires operation over longer distances can take advantage of GSM/3G/4G cellular communication capabilities. While cellular is clearly capable of sending high quantities of data, especially for 4G, the expense and also power consumption will be too high for many applications, but it can be ideal for sensor-based low-bandwidth-data projects that will send very low amounts of data over the Internet. A key product in this area is the SparqEE range of products, including the original tiny CELLv1.0 low-cost development board and a series of shield connecting boards for use with the Raspberry Pi and Arduino platforms.

Sigfox: An alternative wide-range technology is Sigfox, which in terms of range comes between WiFi and cellular. It uses the ISM bands, which are free to use without the need to acquire licenses, to transmit data over a very narrow spectrum to and from connected objects. The idea for Sigfox is that for many M2M applications that run on a small battery and only require low levels of data transfer, then WiFi's range is too short while cellular is too expensive and also consumes too much power. Sigfox uses a technology called Ultra Narrow Band (UNB) and it is only designed to handle low data-transfer speeds of 10 to 1,000 bits per second. It consumes only 50 microwatts compared to 5000 microwatts for cellular communication, or can deliver a typical stand-by time 20 years with a 2.5Ah battery while it is only 0.2 years for cellular.

LoRaWAN: similar in some respects to Sigfox and Neul, LoRaWAN targets wide-area network (WAN) applications and is designed to provide low-power WANs with features specifically needed to support low-cost mobile secure bi-directional communication in IoT, M2M and smart city and industrial applications. Optimized for low-power consumption and supporting large networks with millions and millions of devices, data rates range from 0.3 kbps to 50 kbps.

Table 5: UC1 Protocols

	Standard	Frequency	Range	Data Rate

Bluetooth	Bluetooth 4.2 core specification	2.4GHz (ISM)	50-150m (Smart/BLE)	1Mbps (Smart/BLE)
Zigbee	IEEE802.15.4	2.4GHz	10-100m	250kbps
WiFi	Based on 802.11n (most common usage in homes today)	2.4GHz and 5GHz bands	Approximately 50m	600 Mbps maximum, but 150-200Mbps is more typical, depending on channel frequency used and number of antennas (latest 802.11-ac standard should offer 500Mbps to 1Gbps)
Cellular	GSM/GPRS/EDGE (2G), UMTS/HSPA (3G), LTE (4G)	900/1800/1900/2100 MHz	35km max for GSM; 200km max for HSPA	35-170kps (GPRS), 120-384kbps (EDGE), 384Kbps-2Mbps (UMTS), 600kbps-10Mbps (HSPA), 3-10Mbps (LTE)
Sigfox	Sigfox	900MHz	30-50km (rural environments), 3-10km (urban environments)	10-1000bps
LoRaWAN	LoRaWAN	Sub-GhZ	2-5km (urban environment), 15km (suburban environment)	0.3-50 kbps.

Identification of relevant devices:

- Layer 0: The cloud backend
- Layer 1: LoadSensing gateway
- Layer 2: LoadSensing device, Mobile device, Raspberry pi, Odroid, Camera WiFi, DoS-Sensing device, etc.

5.1.3. Lifecycle of device:

Defining policies for the lifecycle of the devices in a smart city network is usually based on the lifetime of the physical device – it gets damaged, or the battery runs out – or the functionality gets superseded. It is necessary to consider the time a device or group of devices can support the security features that the smart city network needs. The main issue in this use case is that the LoadSensing device relies on a battery.

5.1.4. Testing the use case

The use case will be based on the commercial communication protocols are have available at the moment. Considering the long list of commercial communication protocols for smart cities available these days, this PoC will be based in three communication protocols: Bluetooth, WiFi and LoRaWAN. Hopefully, and with the aim to increase the quality of the experimentations, the use case will also include: Zigbee and Sigfox.

Table 6: UC1 - expected use of protocols

Commercial solution	Adopted in the Use Case	Optional
Bluetooth	X	
Zigbee		X
Z-Wave		
6LowPAN		
Thread		
WiFi	X	
Cellular	X	
NFC		
Sigfox		X
Neul		
LoRaWAN	X	

5.1.5. Privacy and private data management

Privacy is a main concern in Smart Cities; for example, in typical applications as Smart Grid the knowledge about energy consumption in real time is a must to achieve with more efficient energy generation system, however, such information not only represent an opportunity but it opens the door to security threats and have a negative impact on privacy issues. In this context, the anonymization of the collected information through innovative solutions, such as the Shamir Secret Sharing scheme, are of paramount importance to divide the requests and collect information (i.e., with the use of an anonymous routing protocol, etc.). The aim of the privacy-related protocols implemented in this use case is to guarantee privacy while maintaining high quality of service.

5.2. UC2 – smart boats

The second mF2C UC is submitted by XLAB:

5.2.1. Identification of relevant protocols:

Sentinel device is defined by the following protocols and communication paths:

- BLE connectivity to the sensor devices.
- WiFi connectivity for Router communication (optional)
- 2G/3G communication to the server
- Optional - data plan sharing:
    - Connection is possible through router or mobile phone data plan.
    - If it would be requested, more Sentinel devices could share the same data plan.
- Own special protocol for communication to lower the overhead.
- Requirement for communication:
    - Security with small footprint - For example the Sentinel has just a few Kbs transmission per month. In the season a charter boat generates approximately 3MB of transfer per month (off season around 600kB).

5.2.2. Identification of relevant devices:

- Layer 0: The cloud backend
- Layer 1: Sentinel device, Sentinel Router, Mobile Device
- Layer 2: Sentinel remote sensor device, 3rd party devices connected over NMEA2000 protocol (depth sensors, wind sensor, output control unit, engine). 3rd party devices must be supported in Sentinel device.

5.2.3. Lifecycle of device:

- The device is mounted on the vessel, connected to the battery and to the sensors.
- After mounting the user sends request for the activation of the device and SIM card. The QR code from the device needs to be scanned with Sentinel App and sent to the main server. This activation activates the SIM card in the device, which enables the connectivity of the device, and network operator starts charging for the SIM card.
- After activation Sentinel device initializes, connects to the cloud and starts sending the data.

- The Sentinel device can be controlled through the link to the cloud.
- NOTE: The SIM cards are for two zones EU or World. If EU card goes out of EU the communication goes to sleep.

5.2.4. Testing the use case

- Software components have unit tests which perform the automatic testing of the viability of the solution.
- Hardware testing is still manual and currently we do not have an automatic testing/integration procedure.

5.2.5. UC-specific security discussion, including DDoS if applicable

- Sentinel has his own communication protocol which does not integrate encryption or strong security. The system relies on the operator security - all communication is passed through private Access Point Name (APN).
- Sentinel Routers could be a part of the DDoS attack if the ports are not secured or closed - attacker could access the router from the GSM/Operator network and start scanning / attacking device. Closing the ports saves the issue.

5.2.6. Simulation of devices (if applicable)

- Sentinel devices can be simulated. Simulators run everything inside the Server and pretend they are traveling on predefined routes.
- We have also demonstration board for “hands-on” simulation with installed sensors.

5.2.7. Privacy and private data management

Sentinel device collects users data and a boat location. This data are sensible and needs to be stored in a proper way to avoid misuse. For mF2C project a test set data (if needed) will be anonymised in the same way as for the testing and/or developing new data fusion functionalities.

The special care of the private data transmission and handling data should be considered in cases where the devices can share data plan or the communication is routed through other devices. In this case, if it will be supported by the mF2C project, the low footprint cryptography should be integrated, to limit possibilities of man in the middle attack performed on a device sharing the connectivity.

5.3. UC3 – smart hubs

UC3 is Tiscali’s “smart hubs” – the final version of the Smart Fog-Hub Service will be deployed at the Cagliari Elmas airport, so we in principle we could also consider physical security (once more is known about the airport), although one would expect the airport would be reluctant to share security information with mF2C.

5.2.8. Identification of relevant protocols:

The Smart Fog Hub Use Case will use the following protocols:

- BLE connectivity to the sensor devices and beacons(optional).
- WiFi connectivity between the Smart Fog Hub and edge devices
- Fiber/SHDSL link between Cloud and Smart Fog Hub, with LTE(4G) optional backup,
- Requirement for communication:
    - Security - devices in the edge should be authenticated
    - Bandwidth – since machine learning algorithms are to be run, load balancing between the Smart Fog Hub and Cloud would be needed, so a medium/high transfer rate is requested

5.2.9. Identification of relevant devices:

- Layer 2 - **Edge sensors**, including smartphones, laptops, tablets, any other IoT device with WiFi connection; most of them will be data generators, some could have some computing power and potentially could offer/share data and eventually also computing resources
- Layer 2 - **LE Beacons** (optional),
- Layer 1 - which is basically composed by the Fog-Hub, that will perform the role of data collector, power provider for the fog layer processing, with fast links to Layer 0 and WiFi link with Layer 2. It will be configured with some resiliency capabilities, at least for stored data and fast reboot/recover.
- Layer 0 – **Tiscali Cloud**, based on an OpenStack instance that will provide scalable computing power for massive data processing, offering resources in case of need to the Smart Fog Hub

5.2.10. Lifecycle of device:

Devices in Layer 0 and Layer 1 are fixed and permanent, while at Layer 2 a multitude of edge devices will enter and exit the scope.

5.2.11. UC-specific security discussion, including DDoS if applicable

- The airport environment is particularly exposed to untrusted devices
- Since we accept all kind of devices in the scope we expect possibility of any kind of security threats, including network scanning, data leaks, MITM and DDoS, etc.
- The Smart Fog Hub should be fine-tuned in security perspective in order to preserve itself and the Cloud, different policies for Cloud, Fog and the edge should be considered

5.2.12. Simulation of devices (if applicable)

- Currently we have not foreseen the use of simulation of devices, even if it could be possible for testing purposes, especially for edge devices

5.2.13. Privacy and private data management

Depending on the level of engagement different class of private data could be collected and managed, basically personal information and information of related devices. For the purpose of this Use Case most of data will be anonymized, otherwise a suitable data encrypt could be used. In case of higher engagement an appropriate informative page will be presented and agreed by the user prior of use

the service.

6. Conclusion - Challenges and Goals

Clearly this document is only the first security-related deliverable, it can only outline the current state and the perceived future directions; there is nothing specific to evaluate (other than the Architecture). Consequently, there must be future work, in greater depth in the specific areas that turn out to be required for mF2C. This work could be documented both in separate, specific documents, in publications, and in future deliverables. This future work must describe:

- Actual requirements – much of the requirements described here are general cloud/distributed/IoT requirements; not all will apply to every deployment.
- Implementation – how mF2C implements the security requirements, and which trade-offs are made to do so;
- Results of tests – how the actual physical devices perform, e.g. with lightweight cryptography, and in realistic conditions with "proper" networking;
    - In particular, the architecture invites a use of "hybrid" crypto where lower layers do lightweight crypto but communicate "up" through relatively private channels, and stronger security is added by higher layers.
- Implementation of vulnerability management and results of penetration testing, as applicable;
- Results of simulations – how simulations are designed to test particular security features
- Operational security – how the infrastructure is monitored and how security incidents are dealt with; learning best practices for operations.
    - In particular, using machine learning to detect abnormal behaviour is an area of ongoing research; there are mixed opinions on the usefulness of this approach.
    - However, automating as much as possible of incident handling is clearly useful to be able to react in a timely fashion, and to improve scalability.

Some security objectives are important enough for the success of mF2C – required for user trust and meeting overall security objectives – that they need a separate written assessment which can be based on both simulations and testing:

- Privacy – in particular, GDPR compliance [icogdpr]
    - The right to be informed
    - The right of access
    - The right to rectification
    - The right to erasure
    - The right to restrict processing
    - The right to data portability
    - The right to object
    - Rights in relation to automated decision making and profiling.
- Coping with attacks, including DDoS, MITM, spoofing, sniffing, malware injection, or other active and passive attacks.
    - Combining simulation technologies with real-life device, a sandbox approach can be useful to isolate testing on networks, in order to not trigger alarms or violate network

AUPs.

- Usability – if the target audience finds security features hard to use, they may circumvent or bypass the features, or may be put off using the IoT.
- Trust – why should the user trust the infrastructure? Policies must be defined to cover the deployments, to give users the assurance that their data is safe.
- Scalability – the need for also security to scale to a realistic number of devices (if necessarily only in simulation)

7. Outlook

The availability of broadband wireline and wireless connections and exponential growth of connected devices creates new business opportunities. Telco Providers are particularly interested in exploring the new ecosystems in order to spot new value added services to be offered on top of basic connectivity.

At the same time there are compliance requirements in EU and UK on smart energy that demand Energy Suppliers adopt methods and tools to enable the end-user to easily monitor their energy usage. In this scenario ISPs are expected to play a relevant role for the management and transmission of data at customer premises. So ISPs and Telco are very interested in home automation and smart energy/heating/water/gas metering services as they could extend current offering of Data, VoIP, Video Streaming & other online services.

The adoption and success of these new services depends on quality, security, trust and privacy of management and transmission of data, so these services must guarantee at least the following:

- **Data Privacy**, customers ask for fulfilment of data privacy regulations, otherwise the service would be dismissed,
- **Device Authentication,** each device should be recognized as it is and managed accordingly,
- **Information Security**, in terms of confidentiality and integrity of managed information, this is mandatory in case of the information managed and transmitted are related to service usage and metering, in case of lack of guarantee of integrity chance of frauds could prevent any commercial use. Access is not guaranteed because of the volatility of the network connections.

So the new picture is composed of two main domains:

- The **Trusted**, which include systems and networks, cloud and virtualized services provided by the ISP/Telco
- The **Untrusted**, which is composed by an ever increasing number of mobile/edge devices, with an increasing computing power

The second one is characterized by a huge percentage of vulnerable devices, about 70% [HPE2014], mainly due to poor quality software drivers, and use of weak/unsecure protocols to connect objects. Moreover these devices can be hacked in several ways, with chance to modify MAC address/IMEI/other unique identifier, making the authentication process very "*unreliable*": objects in the edge would be botnets, or real devices with fake identities, with possibilities of some varieties of MITM attacks.

The foreseen research areas would answer to the question: "**How to manage all devices at Fog level, connected to a trusted domain (cloud), as a separate layer, with the ability to dynamically serve as a trusted, decision-making instance for enforcing the required policy management strategy?**"

As a requirement to classify information, such as management data, user data, and usage data, in order to apply relevant policies, a labelling model tailored on Fog environments can be derived. Then the assigned labels and attributes can determine the selection of additional security and privacy controls

based on the information's protection requirements. In this perspective reputation and/or consensus algorithms on objects at the edge, as used by latest IPS/antiDDoS advanced appliances, should be verified in order to guarantee the optimal classification and tagging of devices.

The security and privacy framework moreover have to provide a set of security specific data processing functions, including encryption to ensure confidentiality, versioning and signing to guarantee data integrity, pseudo-anonymization and anonymization algorithms on labelled data to prevent excesses in user profiling, key management for cryptographic operations.

Also advanced use of Block Chaining could enforce confidentiality and authenticity of managed information that could be particularly impacting in Fog environments.

ISO/IEC 29115 (Information technology -- Security techniques -- Entity authentication assurance framework) and NIST SP 800-63 (Digital Authentication Guideline) will be used for the determination of the best Level of Assurance achievable.

Making authorization decisions, policy validation and enforcement in arbitrary distributed systems, such as complex Fog infrastructures, is far more complex than in traditional centralized models, so classic access control models, as role-based access control, are not adequate to deal with the high dynamics of the foreseen Fog infrastructures. So novel approaches, including Risk-aware Access Control and Next Generation Access Control need to be analysed and adapted for the adoption in Fog environments.

In case of need of reliable rating and billing of service usage reporting, accounting for all users and services should be guaranteed by the security framework. Accounting data have to be transmitted reliably, unmodified, confidentially, avoiding repudiation by users, with ways to verify all these.

Finally in order to detect service misuse and attacks in Fog environment, audit trails must be generated by providing consistent logging features that facilitates automated processing, aggregation, and correlation of security-related system and service events.

References

References are divided into two parts; the first being publications (usually cited as [AuthorYear]), and the second part is the web links.

[Alazani2016] S. Alanazi, K, Saleem, J. Al-Muhtadi, A. Derhab, "Analysis of denial service impact on data routing in mobile eHealth wireless Mesh Network", Mobile Information Systems Volume 2016 (2016)

[Ali2015] M. Ali, S, U. Khan, A. V. Vasilakos, "Security in cloud computing", Information Sciences 305 (2015) 357-383

[Aliu13] Oh G. Aliu, A. Imran, M. A. Imran, B. Evans, "A survey of self organisation in future cellular networks", IEEE communications survey and tutorials 15 (1), 336-361, 2013.

[Alrawais2017] Al Alrawais, A. Alhothaily, C. Hu, X. Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues", IEEE Internet Computing (Volume: 21, Issue: 2, Mar.-Apr. 2017)

[Aranha2010] D. Aranha, R. Dahab, J. Lopez, and L. Oliveira, "Efficient implementation of elliptic curve cryptography in wireless sensors," Advances in Mathematics of Communications, vol. 4, no. 2, pp. 169–187, 2010.

[Aumasson2010] J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, "Quark: A Lightweight Hash." in CHES 2010, no. 6225 in LNCS, pp. 1–15, Springer-Verlag, 2010.

[Baumgartner2010] Baumgartner, Tobias, Ioannis Chatzigiannakis, Sándor Fekete, Christos Koninis, Alexander Kröller, and Apostolos Pyrgelis. "Wiselib: A generic algorithm library for heterogeneous sensor networks." In Wireless Sensor Networks, pp. 162-177. Springer Berlin Heidelberg, 2010.

[Bayou2015] L. Bayou, D. Espes, N. Cuppens-Boulahia, "Security Issue of WirelessHART Based SCADA Systems", Proceedings International Conference on Risks and Security of Internet and Systems CRiSIS 2015: Risks and Security of Internet and Systems pp 225-241

[BRA2010] Brands, 2010. Brands, S. (2010). U-Prove technology overview. Technical report, Microsoft Corporation.

[Brasser2015] F. Brasser et al. Tytan: Tiny trust anchor for tiny devices. In ACM/EDAC/IEEE DAC'15.

[Castelluccia2009] Castelluccia, C., Francillon, A., Perito, D., Soriente, C. (2009). On the Difficulty of Software-Based Attestation of Embedded Devices. In Proceedings of the 16th ACM conference on Computer and communications security (CCS'09), pp. 400-409. ACM.

[Chiang2016] M. Chiang, T. Zhang, "Fog and IoT: An Overview of Research Opportunities", IEEE Internet of Things Journal (Volume: 3, Issue: 6, Dec. 2016)

[CLEFIA2007] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA." in Proceedings of Fast Software Encryption – FSE'07 (A. Biryukov, ed.), no. 4593 in LNCS, pp. 181–195, Springer-Verlag, 2007.

[CSA2016] J-M Brook, S Field, D Shackleford (eds): CSA Top Threats WG: Treacherous 12: Cloud Computing Top Threats 2016, CSA (2016)

[Dorri2015] A. Dorri, S. R. Kamel, E. Kheyrkha, "Security Challenges in Mobile Ad Hoc Networks: A Survey" International Journal of Computer Science & Engineering Survey (IJCSES) Vol.6, No.1, February 2015

[CAM2001a] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-

transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, Advances in Cryptology - EUROCRYPT 2001, volume 2045 of Lecture Notes in Computer Science, pages 93–118. Springer Berlin / Heidelberg, 2001.

[CAM2001b] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, Security in Communication Networks, volume 2576 of Lecture Notes in Computer Science, pages 268–289. Springer Berlin / Heidelberg, 2003.

[CHA1983] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald Rivest, , and Alan Sherman, editors, Advances in Cryptology Proceedings of Crypto, volume 82, pages 199–203. Plenum Publishing, 1983.

[IBM2012] Security Team, IBM Research, 2012. Security Team, IBM Research (2012). Specification of the Identity Mixer Cryptographic Library, version 2.3.4. Technical report, IBM Research, Zürich.

[Kan2015] C Kannelopoulos, N Liampotis, N van Dijk, P Solagna (eds): Analysis of user community and service provider requirements, AARC DJRA1.1, https://aarc-project.eu/wp-content/uploads/2015/10/AARC-DJRA1.1.pdf (2015)

[Kazim2015] M. Kazim, S. Y. Zhu, “A survey on top security threats in cloud computing”, International Journal of Advanced Computer Science and Applications(IJACSA), 6(3), 2015

[Kirkham2003] T Kirkham, S Winfield, S Ravet, S Kellomäki: *The Personal Data Store Approach to Personal Data Security*, IEEE Security & Privacy vol 11, Issue 5, pp. 12-19 (2003), DOI: 10.1109/MSP.2012.137

[Kovah2012] Kovah, X., Kallenberg, C., Weathers, C., Herzog, A., Albin, M., Butterworth, J. (2012). New Results for Timing-Based Attestation. In Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP’12). pp. 239-253. IEEE.

[Liu2008] An Liu, and Peng Ning. "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks", in Information Processing in Sensor Networks, 2008. IPSN'08. International Conference on, pp. 245-256. IEEE, 2008.

[Morsy2016] M. A. Morsy, J. Grundy, I. Miller, “An Analysis of the Cloud Computing Security Problem”, arXiv:1609.01107v1.

[Nia2016] A. M Nia, N. Jha, “A Comprehensive Study of Security of Internet-of-Things”, IEEE Transactions on Emerging Topics in Computing (Volume: PP, Issue: 99)

[Norman2013] J. Noorman, P. Agten, W. Daniels, R. Strackx, A. Van Herrewege, C. Huygens, B. Preneel, I. Verbauwhede, and F. Piessens. Sancus: Low-cost trustworthy extensible networked devices with a zero-software trusted computing base. In USENIX Security Symposium. USENIX Association, 2013.

[Oliveira2011] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvea, M. Scott, D. F. Camara, J. Lopez, and R. Dahab. TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks. Computer Communications, 4(2):169–187, 2011.

[Owusu2013] E. Owusu, J. Guajardo, J. McCune, J. Newsome, A. Perrig, and A. Vasudevan. OASIS: On achieving a sanctuary for integrity and secrecy on untrusted platforms. In ACM Conference on Computer & Communications Security (CCS). ACM, 2013

[Peng13] M. Peng, D. Liang, Y. Wei, J. Li, H. Chen, “Self-Configuration and Self-Optimization in LTE-Advanced Heterogeneous Networks”, IEEE Communications Magazine , Volume: 51 Issue: 5

[PRESENT2007] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, “PRESENT: An Ultra-Lightweight Block

Cipher." in CHES 2007, no. 4727 in LNCS, pp. 450–466, Springer-Verlag, 2007.

[Singh201] N. Singh, G. Chhabra, K. P. Singh, H. Saini, "A secure authentication in multi-operator domain (SAMD) for wireless Mesh Network", Proceeding of the International conference on data engineering and communication technology, advanced in intelligent systems and computing (ICDECT 2016)

[SMART2012] K. Eldefrawy et al. SMART: Secure and minimal architecture for (establishing a dynamic) root of trust. In NDSS'12

[Sri2010] S. Srinivasamurthy, D. Q. Liu, "Survey on Cloud Computing Security", Presented at 2nd IEEE International Conference on Cloud Computing Technology and Science, Indianapolis, USA.

[Verma2011] A. Verma, S. Kaushal, "Cloud computing Security Issues and Challenges: A Survey", International Conference on Advances in Computing and Communications ACC 2011: Advances in Computing and Communications pp 445-454.

[Wen2015] K. Wen, "Research on Wireless-based Intrusion Detection in Mesh Network security system", Proceedings of the 4th International Conference on Information Technology and Management Innovation (ICITMI 2015)

[Wurster2005] Wurster, G., Van Oorschot, P., Somayaji, A. (2005). A generic attack on checksumming-based software tamper resistance. In Proceedings of the 2005 IEEE Symposium on Security and Privacy (SP'05). pp. 127-138. IEEE.

[Yi2015] S. Yi, Z. Qin, Q. Li, "Security and Privacy Issues of Fog Computing: A Survey", Wireless Algorithms, Systems, and Applications. WASA 2015. Lecture Notes in Computer Science, vol 9204. Springer, Cham.

[Zieg2014] Jan Henrik Ziegeldorf, Oscar Garcia Morchon, and Klaus Wehrle. "Privacy in the Internet of Things: threats and challenges.", Security and Communication Networks 7.12 (2014): 2728-2742

Web references

[ARM] ARM TrustZone. https://developer.arm.com/technologies/trustzone
[ARMCortex] https://www.arm.com/products/processors/cortex-m/
[atmeldatasheet] http://www.atmel.com/Images/Atmel-42223%E2%80%93SAM-R21_Datasheet.pdf
[avast] https://press.avast.com/avast-exposes-internet-of-things-attack-risk-in-barcelona-home-of-mobile-world-congress-2017
[AvrCryptoLib] https://wiki.das-labor.org/w/AVR-Crypto-Lib/en
[bbccharprv] http://www.bbc.co.uk/news/technology-39502258
[bbcinvpow] http://www.bbc.co.uk/news/uk-34713435
[botenigma] http://www.enigmasoftware.com/top-10-botnet-threats-in-the-united-states/
[bothoneynet] https://www.honeynet.org/book/export/html/50
[CC-ISO15408] https://en.wikipedia.org/wiki/Common_Criteria
[chinexpctrl] http://exctrl.mofcom.gov.cn/
[cifasfrd] https://www.cifas.org.uk/secure/contentPORT/uploads/documents/160706_cifas_fraudscape_ONLINE.pdf
[concernaca] http://concernedscientists.org/2017/01/over-5500-academics-arrested-in-turkey-since-rule-of-law-suspended/
[CSA] http://www.cloudsecurityalliance.org/
[dynorg] https://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/
[ecprivshld] http://ec.europa.eu/justice/data-protection/files/factsheets/factsheet_eu-us_privacy_shield_en.pdf

[eSTREAM] http://www.ecrypt.eu.org/stream/.
[euwp29] http://ec.europa.eu/newsroom/just/item-detail.cfm?item_id=50083
[forbprocera] https://www.forbes.com/sites/thomasbrewster/2016/10/25/procera-francisco-partners-turkey-surveillance-erdogan/#4a5bda5e4434
[fortunepol] http://fortune.com/2017/03/29/white-house-trump-internet-privacy/
[FreeRTOS] RTOS available under GPL (http://www.freertos.org)
[GOLANG] https://golang.org/doc/install/source#environment
[GRPC] http://www.grpc.io/
[huefcc] https://apps.fcc.gov/oetcf/eas/reports/ViewExhibitReport.cfm?mode=Exhibits&RequestTimeout=500&calledFromFrame=Y&application_id=BGxrt14HR5P%2B6fn5P8ASjw%3D%3D&fcc_id=O3M324131201801
[huefccwifi] https://apps.fcc.gov/eas/GetApplicationAttachment.html?id=2693103
[hueoflynn] https://www.youtube.com/watch?v=hi2D2MnwiGM
[hueoflynn2] https://www.blackhat.com/us-16/briefings.html#a-lightbulb-worm
[hueseger] https://medium.com/@rxseger/enabling-the-hidden-wi-fi-radio-on-the-philips-hue-bridge-2-0-42949f0154e1
[icogdpr] https://ico.org.uk/for-organisations/data-protection-reform/overview-of-the-gdpr/
[icopia] https://ico.org.uk/media/for-organisations/documents/1595/pia-code-of-practice.pdf
[icoprep] https://ico.org.uk/media/1624219/preparing-for-the-gdpr-12-steps.pdf
[icowhen] https://ico.org.uk/for-organisations/data-protection-reform/guidance-what-to-expect-and-when/
[(ISC)2] (ISC)2, http://library.ahima.org/doc?oid=107038
[isprev] http://www.ispreview.co.uk/index.php/2016/11/controversial-new-uk-internet-snooping-bill-approved-mps.html
[ivezic] http://ivezic.com/cyber-physical-systems-security/iot-security/internet-of-things-iot-security-guidelines-and-frameworks/
[miraianalysis] https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html
[miraiturnkey] https://www.sentryo.net/the-mirai-iot-botnet-a-publically-available-turn-key-threat-2/
[onionpubl] https://www.onion-router.net/Publications.html
[oscca] http://www.freshfields.com/en/global/Digital/China_rules_on_encryption/
[owasp10priv] https://www.owasp.org/index.php/OWASP_Top_10_Privacy_Risks_Project
[owaspframework] https://www.owasp.org/index.php/IoT_Framework_Assessment
[polrevturkey] https://policyreview.info/articles/news/internet-and-recent-coup-attempt-turkey/423
[privpolctry] http://privacypolicies.com/blog/privacy-law-by-country/
[RELIC] D. F. Aranha and C. P. L. Gouvea, “RELIC is an Efficient LIbrary for ˆ Cryptography,” https://github.com/relic-toolkit/relic.
[SecByDesign] https://en.wikipedia.org/wiki/Security_by_design
[shodan] https://www.shodan.io/explore
[shodandflt] https://www.shodan.io/search?query=%22default+password%22
[schneierpol] https://www.schneier.com/blog/archives/2017/02/security_and_pr.html
[stdspaving] https://www.standardsuniversity.org/e-magazine/march-2016/iot-security-standards-paving-the-way-for-customer-confidence/
[TPM] Trusted Platform Module (TPM) Summary, https://trustedcomputinggroup.org/trusted-platform-module-tpm-summary/
[ukcommchin] https://www.bis.doc.gov/index.php/enforcement/oee/220-eco-country-pages/1040-china-export-control-information
[ukcommctrl] https://www.bis.doc.gov/index.php/regulations/commerce-control-list-ccl
[usaear] https://www.bis.doc.gov/index.php/regulations/export-administration-regulations-ear
[verotrack] http://vero.solutions/solutions/verotrack/tracking-people/

[w3ruleng] https://www.w3.org/2005/rules/wg/wiki/List_of_Rule_Systems
[wikipgdpr] https://en.wikipedia.org/wiki/General_Data_Protection_Regulation
[wikipimpcrypt]
https://en.wikipedia.org/wiki/Restrictions_on_the_import_of_cryptography#Status_by_country
[wikipjrn] https://en.wikipedia.org/wiki/List_of_arrested_journalists_in_Turkey
[wikiptor] https://en.wikipedia.org/wiki/Tor_(anonymity_network)#Weaknesses
[wikipusprv] https://en.wikipedia.org/wiki/Information_privacy_law#United_States
[wolfSSL] https://github.com/wolfSSL/wolfssl
[wpostprivbill] https://www.washingtonpost.com/business/economy/house-sends-bill-rolling-back-internet-privacy-protections-to-trump/2017/03/28/db704ca4-13d5-11e7-9e4f-09aa75d3ec57_story.html?utm_term=.8a9e4441ab96

Annex 1: Penetration testing

Many security tests will need to be performed; this Annex illustrates the process by describing in more detail the penetration testing (section 2.8)

Penetration tests

There are three **types** of penetration tests:

- black-box,
- white-box,
- grey-box.

In a black-box assessment, the client provides no information prior to the start of testing. In a white-box assessment, the entity may provide the penetration tester with full and complete details of the network and applications. For grey-box assessments, the entity may provide partial details of the targeted systems. Typically white-box and grey-box assessments yield more accurate results and provide a more comprehensive test of the security posture of the environment than a pure black-box assessment.

The **process** for performing a penetration test must be determined before testing the networking devices and system vulnerabilities. The penetration testing process include the following procedures:

- Defining the scope (list of all target devices)
- Performing the penetration test (it's up to the tester guarantee the applications, networks and systems are not vulnerable to a security risk that could allow unauthorized access)
- Reporting and delivering results, with list of prioritized vulnerabilities and risks, information about each device's vulnerabilities, and recommendations for repairing found vulnerabilities and provide technical information on how to fix vulnerabilities found in the system.

The following activities must be **fulfilled** in a penetration test:

- Appointing a qualified penetration tester, who follows rules of Non Disclosure Agreement,
- Clear definition of main parameters of the test, such as objectives, limitations, and justifications of the procedures,
    - Choosing a suitable set of tests that balances costs and benefits,
    - Following a methodology with proper planning and documentation,
    - Documenting results and recommendations carefully and making comprehensible for the client,
    - Availability of the tester to answer any query regarding the test.

Penetration Testing Techniques

There are different techniques that are commonly used, they differentiate in the way they expose the attacker, ranging from a stealth attack (no visibility/impact) to a very aggressive attack:

- Passive scanning, carried out during the start of an external penetration test and provides information on the configuration of a system by using public domain sources. This include:
    - Network mapping & OS fingerprinting: provide an overview of the configuration of the entire network being tested, and these techniques are designed to specify different types of services present on the target system
    - Spoofing: the act of using one machine to pretend to be another. This technique is used in both internal and external penetration testing to access computers that are configured to reply only to specific computers
    - Network sniffing: used to capture data as it travels across a network. This is usually performed as a part of internal penetration testing, as it is very easy to capture packets from within a network

- Traffic Analysis can be used to identify the most important controlling servers and their physical location. This could make them vulnerable to physical tampering and root compromise. One proposed response to traffic analysis is to use Tor or I2P/Freenet. However these tools only obfuscate the connections between end-user and server despite their use of cryptographic tools. By using correlation attacks and statistical analysis it is possible to recover the access patterns [onionpubl] [wikiptor].
- Trojan attack: Trojans are malicious code that are usually sent to a network as email attachments or transferred via chat rooms. A penetration test attempts to send specially crafted Trojans to a network
- Brute force attack: a brute force attack is the most commonly known password cracking method; the attacker basically tries to use all possible character combinations to crack the password effectively. It can overload a system and possibly stop it from responding to legal requests.
- A related type of attack probes the hardware directly. For example, for lightweight devices performing RSA operations, security researchers have derived information about the private key by measuring the power consumption of the device.

Penetration Testing Strategies

Penetration tests can be conducted using different strategies:

- External penetration testing is mainly done on servers, core software, and other infrastructure components. It is a conventional method of penetration testing. It normally starts from an external entry point, but some internal entry point could be selected (e.g. Guests wireless)
- Internal security assessment: The internal security assessment offers a clear view of the site's security. Internal security assessments have a methodology similar to external penetration testing.
- Application security assessment: Application security assessment has a methodology similar to external penetration testing.
- Network security assessment: The network security assessment identifies risks and vulnerabilities that may harm network and security policies. It also provides information that is needed to make network security decisions.

- Wireless/remote-access security assessment: Wireless/remote-access security assessment deals with the security risks associated with wireless devices. Some of the wireless devices that are under security threat are 802.11 wireless networking and Internet access through broadband. Precautions must be taken so that the architecture, design, and deployment of such solutions are secure.
- Telephony security assessment: Telephony security assessment deals with the security issues of voice technologies. Penetration testers may attempt to exploit the PBXs to route calls at the target's expense or check mailbox deployment and security, voice over IP (VoIP) integration, unauthorized modem use, and associated risks.
- Social engineering assessment: it is a technique used by attackers to exploit the human vulnerabilities within a network. Social engineering is a procedure where the weaknesses and the amicability of people are exploited. Testers may use techniques such as eavesdropping, dumpster diving, cracking employee passwords through guessing, and trying to memorize access codes by observing people.
- Hardware assessment: if a lightweight device is taken out of its normal habitat and probed in a laboratory, it can become more difficult for it to protect its secrets. Conversely, the same types of tests are used in testing for security certifications; compare NIST's [FIPS140-2].

Annex 2: List of IoT platforms

Important features expected from an IoT platform

- Device management and Integration support
- Information security
- Data collection protocols
- Data analytics
- Benchmarks
- Edge analytics
- Support for IoT context
- Handling out-of-order processing

Primarily Cloud platforms

Table 7: List of cloud platforms

Name	Description
Amazon AWS IoT	Safe and reliable network. Supports huge amount of messages. Connects to Amazon S3, Amazon Machine Learning, Amazon Elastic Compute Cloud, Amazon Lambda, Amazon DynamoDB
Microsoft IoT pack for Raspberry Pi 2	Gets users started quickly. Uses software developed by Adafruit. Integrates with Windows 10 IoT core.
IBM BlueMix	Create and expose enterprise APIs to BlueMix Transform and synchronise data Securely connect to many environments

Primarily software platforms

Table 8: List of software IoT platforms

Name	Opensource?	Description
IoT-ignite		Android-based
Android Things		From Google

XMPRO Agile IoT Platform		Heterogenous connectivity. Machine-learning. BPM and workflow. Azure/AWS/Sap Hanna
Aeris IoT platform		Integrated connectivity, analytic and application suite
ClearBlade Novi		
Ayla Networks		
KAA	Yes. Apache 2.0	10KB ram, footprint. Guaranteed data delivery. Transport security. Efficient serialisation.
GE Predix		Industrial Big Data, Software-defined machines
Oracle Integrated Cloud		Error detection and repair, Rich connectivity, Point and click
Carriots		PaaS-based, Rest API, Device management, Rule engine, Listeners and Triggers, Access to 3rd party APIs
Salesforce IoT cloud		Uses Thunder, Heroku Connect, Integrates into Salesforce.com
Cisco IoT system		4G, 3G, 2G, WiFi, small cell networks combined into Cisco Virtualised Packet Core Scales quickly because virtualised
IBM Watson		Speech to text/ text to speech, Visual recognition, Concept insights, Tradeoff analytics
ThingWorx		Predictive analytics, User interfaces with mashups, Event-driven execution engine, Device management, Intelligence tools
Microsoft Azure IoT suite		Analytics, Database, Power BI, Applications, Notification hubs, Mobile
Amazon AWS IoT platform		

RTI Connext DDS		Industrial, Doesn't use message brokers
OpenSensors		Publishes sensor data on open-access basis for free via message broker, Provides public and private infrastructure, Supported by Tim Berners-Lee
Verizon ThingSpace		Rest API, Connect and check devices, Tracks lifecycle of device, Provides a simulator, Hosts application servers at Verizon data centre
2lemetry IoT Analytics platform		Acquired by AWS IoT, Real-time analytics Apache Storm, Integrates with ThingWorx, Salesforce Heroku; supports MQTT, CoAP, STOMP, M3DA
Appcelerator		Real-time analytics Titanium; supports MQTT, HTTP
Bosch IoT suite		MQTT, CoAP, STOMP, AMQP
Ericsson Device Connection Platform		CoAP
Everything IoT smart Products platform		MQTT, CoAP, WebSockets
IBM IoT Foundation Device Cloud		MQTT, HTTPS
PLAT.ONE		MQTT, SNMP
ParStream IoT Analytics Platform		Acquired by Cisco; Real-time analytics; Batch analytics; ParStream DB; Claims 3 to 4 million rows per second throughput
Connect2.me		Middleware platform
Xively PaaS enterprise IoT platform		HTTP, HTTPS, Sockets, Websocket, MQTT
Cayenne arduino platform		Gui arduino development toolset

Primarily hardware platforms

Table 9: List of IoT hardware platforms

Name	Open source?	
NuvlaBox		http://sixsq.com/services/nuvla/
Arduino	Yes	
Raspberry-pi	yes	
Intel Galileo		Very light linux distro plus Arduino environment
Tibbo Project System		Programmable and configurable.; Two programming languages Tibbo C and Tibbo Basic

Annex 3: List of IoT security frameworks

Primarily implemented in software

Table 10: List of software-based IoT security frameworks

Name	Description
Verizon Enterprise Solutions	Offers a security credentialing service in addition to existing security
Symantec embedded security	Protects embedded operating systems; QNX, Windows embedded, Linux; Protects automotive, industrial control systems, ATM, Point of sale, medical devices.; Security, management, analytics
Bitdefender box	protection for home devices, smartphones
Kramba security	Protects the electronic control unit (ECU) of cars
Gemalto	Secure hardware tokens
Digicert IoT solutions	Encryption. Signing of software
Trustwave	Finds weaknesses in software to harden, develop and test.

Primarily implemented in hardware

Table 11: List of hardware IoT security platforms

Infineon technologies	Hardware-based device integrity checks, authentication, secure key management

Primarily documentation-based security frameworks - e.g. checklists

Two lengthy lists of well-known security frameworks [ivezic] [schneierpol]

A simplified overview of security topics [stdspaving]

Important features expected in a security framework

For comparison with the list in section 0, this lists the required features of a (not necessarily IoT) security framework [owaspframework] (modified slightly):

Table 12: General security framework - required features

Feature

Secure key management and entity identities, secure onboarding process
Communications encryption
Storage encryption
Strong logging
Automatic updates (e.g. trusted patches, see next items)
Version reporting
Update verification
Cryptographic ID (of devices or other entities, e.g. keys)
No default passwords
Strong local authentication
Offline security features
Configurable root trust store (e.g. certificates)
Device and owner authentication (i.e. a personal device is used by its owner)
Transfer of ownership handling
Defensive capabilities
Secure M2M
Secure web interface
Use established protocols
Ability to use hardware security features
Ability to use multi-factor authentication if available
Location-aware permissions
Tracks and contains data from tainted sources
Features disabled by default
Written in a type-safe programming language
No hard-coded secrets
Device monitoring and management
Usability
Scalability

Annex 4: List of protocols for data collection

Table 13: List of data protocols

Name
MQTT
SNMP
AMQP
XMPP
CoAP
DDS
WebSockets
HTTP
HTTPS
STOMP
M3DA

Annex 5: botnets and the Mirai botnet

There are numerous types of botnet as shown in Figure 4[botenigma].

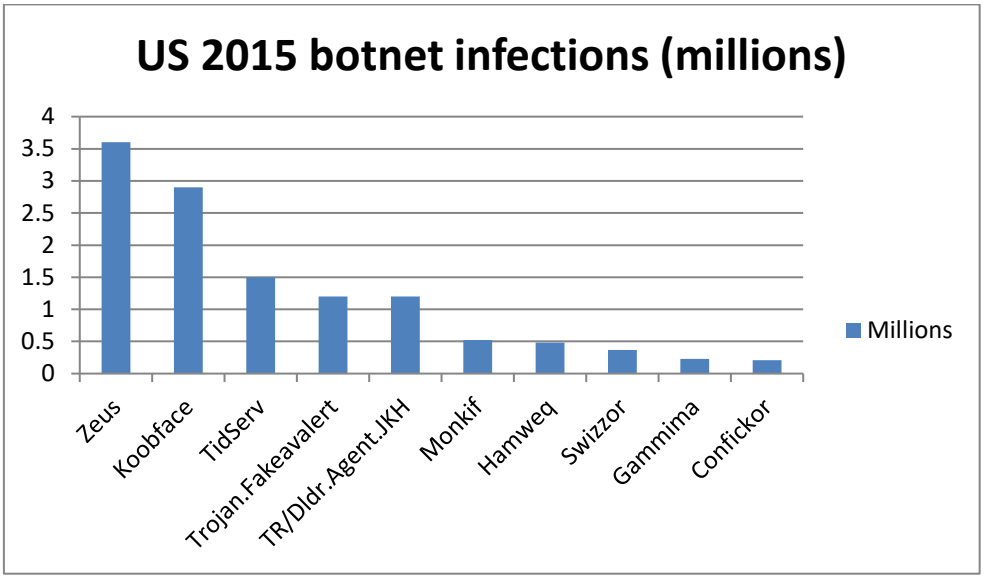

Figure 4: US 2015 botnet infections

The security researcher Avast in a publicity release noted that there are 5.3 million smart devices in Spain that are vulnerable and 493 thousand in Barcelona alone [avast].

A list of individual devices that are vulnerable to many types of botnet can be extracted at the Shodan website [shodan]. The reason these devices are visible despite being behind a home broadband router firewall is because uPNP makes temporary holes in the firewall to allow incoming connections from the owner.

As an example, we look at the Mirai botnet.

Mirai operates by searching for devices with default passwords that are known. These then recruit into the botnet more devices with default passwords. On a command from the botnet controller the devices send packets of random data that are very large. These overwhelm the target by causing it to try to unpack the data from large numbers of devices.

The standard DDoS protection is to look for packets from known botnet addresses and drop the packets. But the Mirai botnet overcame this protection by encapsulating its attack inside General Route Encapsulation (GRE) protocol packets which had randomised addresses. This made the DDoS protection ineffective.

The list of vulnerable devices is hardcoded into Mirai and targets mostly consumer webcams. This could easily be modified to attack SCADA devices, for example, which even now still have default passwords in some cases [miraiturnkey] . Mirai is also able to remove competing worms and trojans from its vulnerable device. It also blocks attempts to login by disabling telnet, ssh and http.

[miraianalysis]

Below is a list of devices known to be vulnerable to the Mirai botnet due to default passwords that are well known. From this it can be seen that the type of device includes webcams, DVRs, voip phones, printers and routers – and that the default passwords generally are not very strong. (The list was taken from the source code for Mirai.)

Table 14: List of devices vulnerable to Mirai

Username/Password	Manufacturer
admin/123456	ACTi IP Camera
root/anko	ANKO Products DVR
root/pass	Axis IP Camera, et. al
root/vizxv	Dahua Camera
root/888888	Dahua DVR
root/666666	Dahua DVR
root/7ujMko0vizxv	Dahua IP Camera
root/7ujMko0admin	Dahua IP Camera
666666/666666	Dahua IP Camera
root/dreambox	Dreambox TV receiver
root/zlxx	EV ZLX Two-way Speaker?
root/juantech	Guangzhou Juan Optical
root/xc3511	H.264 – Chinese DVR
root/hi3518	HiSilicon IP Camera
root/klv123	HiSilicon IP Camera
root/klv1234	HiSilicon IP Camera
root/jvbzd	HiSilicon IP Camera
root/admin	IPX-DDK Network Camera
root/system	IQinVision Cameras, et. al
admin/meinsm	Mobotix Network Camera
root/54321	Packet8 VOIP Phone, et. al
root/00000000	Panasonic Printer
root/realtek	RealTek Routers
admin/1111111	Samsung IP Camera
root/xmhdipc	Shenzhen Anran Security Camera

admin/smcadmin	SMC Routers
root/ikwb	Toshiba Network Camera
ubnt/ubnt	Ubiquiti AirOS Router
supervisor/supervisor	VideoIQ
root/<none>	Vivotek IP Camera
admin/1111	Xerox printers, et. al
root/Zte521	ZTE Router

Annex 6: List of policy engines

Also known as business rules engines, business rules management system or business rules and decision management systems.

There is another list at this URL [w3ruleng]

Table 15: List of policy engines

Name	Description	Open source?
Sandvine Policy Engine	Network policy enforcement only	no
Congress	Mostly intended for cloud provisioning. Uses Datalog declarative language	yes
OASIS XACML also Apache openAZ	attribute-based access control Uses XML	yes
Unified Rule Engine	generic opencog rule engine Uses Pattern Matcher Difficult syntax	yes
Gandalf Decision Engine	scoring centralised as decision as a service. Rest api	yes freemium model
Rule-reactor	small footprint, runs at the client. Uses javascript.	yes
Drools	rete trees for decisions centralised on an application server uses java	yes license apache
OpenRules	business rules and decision management system. uses java	yes freemium
Easy Rules	simple easy to use api uses java	yes MIT license

OpenL Tablets	table based	
Windows Workflow Foundation	uses .Net 3.5	
Haley Expert Rules	web api uses C# and java	
Inrule	uses .Net	
DTRules	uses java	yes
Flexrule	rete tree	
Tibco Business rules		no

Annex 7: List of GDPR impacts

Following is a summary of GDPR impacts taken from the British regulator's (the Information Commissioner's Office - ICO) documentation [icoprep]

Their relevance to mF2C is shown. Some of the impact will be on mF2C users and some on mF2C systems.

Table 16: List of GDPR impacts

Activity	Description	Impact on mF2C
Awareness	Make decision makers and key people aware the law is changing	Implementing the GDPR could have significant resource implications, especially for larger and more complex organisations.
Information you hold	You should document what personal data you hold, where it came from and who you share it with. You may need to organise an information audit	Correcting inaccurate personal information - a log of what is held, where it came from and where it has been sent to is required. Effective policies and procedures are required and they must be documented.
Communicating privacy information	You should review your current privacy notices	Amend the privacy notice to explain the legal basis for processing the data, retention periods and that

		people have a right to complain.
Individual's rights	Check your procedures to ensure they cover all the rights that individuals have, including how you would delete personal data or provide data electronically	Data portability features to provide the data electronically are required. This data may be scattered across mF2C systems and be in device-specific formats.
Subject access requests	Update your procedures and plan how you will handle requests within the new timescales and provide any additional information	Human intervention is required at this point. The timescales for responding are shortened to 30 days. The grounds for refusing are changing but will require policies and procedures in place to document why it is being refused. This data may be scattered across mF2C systems.
Legal basis for processing data	Look at the various types of data processing you carry out, identify your legal basis for carrying it out and document it.	The legal basis for processing will have to be explained in the privacy notice and also in the subject access request. The legal basis needs to be documented for accountability checks. The legal basis may not be clear. People will have a stronger right to have their data deleted where you use consent as your legal basis for processing.
Consent	Review how you are seeking, obtaining and recording consent	Consent has to be verifiable. The controller has to demonstrate that it was given and so an audit trail is required.
Children	Put systems in place to verify individuals' ages and to gather parental or guardian consent	Consent has to be written in words that children can understand. Also the same language they speak. A parent's or guardian's consent is

		required. Ages must be verified.
Data breaches	Have the right procedures in place to detect, report and investigate a personal data breach. Only data breaches where the individual is likely to suffer some form of damage, such as through identity theft or a confidentiality breach need be reported.	Affected people have to be informed which may be difficult to do. Detecting may be difficult to do. All organisations will now have to follow this rule . Management of the actual data breach is necessary. Systems are geographically widely spread. Assess what data is within the scope.
Data Protection by Design and Data Protection Impact Assessments	Note that you do not always have to carry out a PIA (= DPIA) – a PIA is required in high-risk situations. It has always been good practice to adopt a privacy by design approach and to carry out a privacy impact assessment as part of this	mF2C may not know what situations the data will be involved in and may not know whether or not it is high-risk. mF2C would be required to consult with the ICO if it might be a high-risk situation. Can this process be pushed out to the user?
Data Protection Officers	Appoint someone to take responsibility for data protection compliance	A DPO would be required when activities involve the regular and systematic monitoring of data subjects on a large scale. The DPO must take responsibility for your data protection compliance and have the knowledge, support and authority to do so effectively
International	Determine which data protection supervisory authority you come under.	The EU WP29 has produced guidelines explaining this. This will impact on mF2C.

Annex 8: List of relevant security classifications

Some of these prove only a theoretical knowledge of organizational and technical security, while others require also a hands-on experience, or examinations or audits by a trusted party; some others require a periodic verification, training update and continuous improvement tasks.

The following are some of most popular certifications:

- ISO
  - Lead Auditor ISO/IEC 27001
  - Lead Auditor ISO 22301 Business Continuity management
- ISACA
  - CISA – Certified Information Systems Auditor
  - CISM – Certified Information Security Manager
  - CGEIT – Certified in the Governance of Enterprise IT
  - CRISC – Risk and Information Systems Control certification
- ISC
  - CISSP – Certified Information Systems Security Professional
- CSA
  - CCSK – Certificate of Cloud Security Knowledge
  - STAR – Security Trust & Assurance Registry
- CompTIA
  - Security+
  - CASP – CompTIA Advanced Security Practitioner
- PCI-DSS
  - QSA – Qualified Security Assessor
- OSSTMM
  - OPSA – OSSTMM Professional Security Analyst
  - OPSE - OSSTMM Professional Security Expert
- EC COUNCIL
  - CEH – Certified Ethical Hacker
  - LPT – Licensed Penetration Tester
  - ECSA – EC-Council Certified Security Analyst
- CISCO
  - CCIE Security
  - CCDP Security
  - CCNP Security

Security Products Certifications

The **Common Criteria** [CC-ISO15408] for Information Technology Security Evaluation is an international standard (**ISO/IEC 15408**) for computer security certification.

Common Criteria is a framework in which computer system users can specify their security functional and assurance requirements (SFRs and SARs respectively) through the use of Protection Profiles (PPs),

vendors can then implement and/or make claims about the security attributes of their products, and testing laboratories can evaluate the products to determine if they actually meet the claims. In other words, Common Criteria provides assurance that the process of specification, implementation and evaluation of a computer security product has been conducted in a rigorous and standard and repeatable manner at a level that is commensurate with the target environment for use.

Common Criteria evaluations are performed on computer security products and systems. So main point is:

Target Of Evaluation (TOE) – the product or system that is the subject of the evaluation.

The evaluation serves to validate claims made about the target. To be of practical use, the evaluation must verify the target's security features. This is done through the following:

Protection Profile (PP) – a document, typically created by a user or user community, which identifies security requirements for a class of security devices (for example, smart cards used to provide digital signatureshttps://en.wikipedia.org/wiki/Digital_signature, or network firewalls) relevant to that user for a particular purpose. Product vendors can choose to implement products that comply with one or more PPs, and have their products evaluated against those PPs. In such a case, a PP may serve as a template for the product's ST (Security Target, as defined below), or the authors of the ST will at least ensure that all requirements in relevant PPs also appear in the target's ST document. Customers looking for particular types of products can focus on those certified against the PP that meets their requirements.

Security Target (ST) – the document that identifies the security properties of the target of evaluation. The ST may claim conformance with one or more PPs. The TOE is evaluated against the SFRs (Security Functional Requirements. Again, see below) established in its ST, no more and no less. This allows vendors to tailor the evaluation to accurately match the intended capabilities of their product. This means that a network firewall does not have to meet the same functional requirements as a database management system, and that different firewalls may in fact be evaluated against completely different lists of requirements. The ST is usually published so that potential customers may determine the specific security features that have been certified by the evaluation.

Security Functional Requirements (SFRs) – specify individual security functions which may be provided by a product. The Common Criteria presents a standard catalogue of such functions. For example, a SFR may state how a user acting a particular role might be authenticated. The list of SFRs can vary from one evaluation to the next, even if two targets are the same type of product. Although Common Criteria does not prescribe any SFRs to be included in an ST, it identifies dependencies where the correct operation of one function (such as the ability to limit access according to roles) is dependent on another (such as the ability to identify individual roles).

The evaluation process also tries to establish the level of confidence that may be placed in the product's security features through quality assurance processes:

Security Assurance Requirements (SARs) – descriptions of the measures taken during development

and evaluation of the product to assure compliance with the claimed security functionality. For example, an evaluation may require that all source code is kept in a change management system, or that full functional testing is performed. The Common Criteria provides a catalogue of these, and the requirements may vary from one evaluation to the next. The requirements for particular targets or types of products are documented in the ST and PP, respectively.

Evaluation Assurance Level (EAL) – the numerical rating describing the depth and rigor of an evaluation. Each EAL corresponds to a package of security assurance requirements (SARs, see above) which covers the complete development of a product, with a given level of strictness. Common Criteria lists seven levels, with EAL 1 being the most basic (and therefore cheapest to implement and evaluate) and EAL 7 being the most stringent (and most expensive). Normally, an ST or PP author will not select assurance requirements individually but choose one of these packages, possibly 'augmenting' requirements in a few areas with requirements from a higher level. Higher EALs do not necessarily imply "better security", they only mean that the claimed security assurance of the TOE has been more extensively verified.

Most PPs and most evaluated STs/certified products have been for IT components (e.g. firewalls, smart cards, operating systems). Common Criteria certification is sometimes specified for IT procurement.