Final Project - Draft Report

Custos

the watcher who will guard the watchers

Chosen template: Physical Computing and the Internet of Things
8.1: *Secure IoT Device Management in a Safety-Critical Smart Environment*

Course: CM3070

Date of submission: 04/08/2025

Author: J. A. Ferguson, London, UK

The final product, following the template **“Physical Computing and the Internet of Things - 8.1: Secure IoT Device Management in a Safety-Critical Smart Environment”** will allow for the secure integration and continuous monitoring of a modular smart agriculture system. The problem is that smart tech is deployed with convenience in mind, not security. We look to change this by developing a system to integrate smart devices and enforce security policy. Agriculture is one of the oldest forms of human technology, and successful agriculture requires certain conditions to produce viable foods. Variable moisture conditions, unstable temperatures, and lack of wired electricity has made it hard to deploy technology to monitor environmental conditions. However, in the last decade, internet connected devices have become cheaper and more resilient. This is evident by the boom in smart home technology such as the video doorbell, the smart fridge, the heating and lighting we can control just with our voice. In 2025 we can easily conceive that in the next five years, most farms will have some kind of monitoring or automation system deployed in situ.

We might also conceive of a threat to this system. Agriculture qualifies as a safety-critical domain, especially in the context of food security and climate instability. In

2024, the Labour Party of the UK wrote that "food security is national security."[1] When viewed in the context of the Russia- Ukraine war, this statement clearly indicates why an effort must be made to protect the agriculture industry from cyberattack. Existing systems are hugely expensive, complex proprietary technology developed by large corporations, universities or through government grants. This poses a barrier to the small projects, the home farmers, the kitchen gardeners who will be connecting their food production to the internet, but may not be doing so securely. Most contemporary agriculture does not have an easily accessible and safe platform for introducing internet connected devices to monitor and control the conditions of the land.

This project proposes a secure and modular platform for microcontroller driven devices to be added as modules into a system, with each device monitored and its behaviour logged and visualised for the user. It will be possible to remove a device from the network remotely at any time, either manually or automatically using policy based triggers. The focus of other products is to monitor telemetry from sensors that read the conditions of the land itself, this project monitors the sensors themselves, taking our title "Custos" from the Latin for "guardian" or "watcher". The concept of this project is to create the concept for a system that works at any size, and the minimum viable product is in my kitchen garden.

This project addresses this in two ways - firstly by building a secure system, and secondly by monitoring the connected devices for anomalies that might indicate compromise.

The Custos platform will consist of:

- A set of microcontrollers simulating sensor and actuator devices
- A central MQTT broker for message transport
- A backend service to evaluate telemetry and enforce policy
- A dashboard to visualise device state and trigger alerts

The proof of concept model will be for my small garden, with 1–2 sensor nodes streaming data, live dashboard visualisation, and alert generation for anomalous behaviour (e.g. silence, extreme values).

Custos takes a different approach to other internet connected agriculture products, and a different approach to a lot of IoT in general. It monitors the devices instead of focusing

[1] Cupriak, A., 2024. Labour manifesto: ‘Food security is national security’. *Farmers Guide*. Available at: <https://www.farmersguide.co.uk/business/labour-manifesto-food-security-is-national-security/> [Accessed 15 June 2025].

on the agriculture. This will hopefully allow for some early detection of failures, and allow people to fix misconfiguration in addition to protecting from cyber attack or failure.

Literature Review

For the last two decades, the world has been populated with internet connected devices; starting with the smart phone, moving through the household from toothbrushes to vacuum cleaners, from wristwatches to lighting. The proliferation of such devices has enabled data to flow in new and unexpected ways, not restricted to the home but also allowed for increased automation and monitoring of the natural environment, notably in the agriculture industry. Previously, I have written (in the CM3040 module) on how small scale smart technology can be introduced into everyday gardening. In this paper, I am investigating how IoT can be integrated into existing designs, to build new and harden existing smart agriculture solutions in a security conscious way. This is an urgent topic because if we rely on internet connected technologies to grow our food, we must ensure that the supply of our food is not vulnerable to a catastrophic technology induced failure.

The smart devices that make up the internet of things (commonly referred to as IoT devices, or just IoT) have begun to transform the agriculture industry, with the potential for a similar impact upon society as the plough made thousands of years ago. This new age of farming was coined “Agriculture 4.0” by a global consulting firm (De Clercq, Vats and Biel, 2018) as part of research conducted in the last decade,[2] describing a “food security” issue in terms of the correlation of worldwide population growth and the subsequent increasing demand for food, posing this as an issue to be solved by the technology sector.[3] Although many technologies are mentioned, including; drone technology, artificial intelligence, 3D printing, automated hydroponics, vertical farming, and data analytics - the Internet of Things is not identified as something that is foundational to the success of Agriculture 4.0. Rather, IoT is included in the long list, as if it was but another module, interchangeable with AI.[4] This early article is indicative of the ongoing situation, where commercial solutions offer technology to an industry under pressure to adapt and adopt, where farmers are being promised efficiency and profits, using the threat to food security as a weapon without considering how the technologies themselves could be weaponised.

[2] De Clercq, M., Vats, A. and Biel, A., 2018. Agriculture 4.0: The Future of Farming Technology. [online] Available at: <https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2021/apr/agriculture-4-0-the-future-of-farming-technology.pdf>. p.5.

[3] Ibid. p.12.

[4] Ibid. p. 16.

Technical or cyber security is not marketed as the highest priority for the target audience for many of the consumers of these IoT devices, but the devices themselves are well placed to be secured. In agriculture, many of the IoT devices are sensors that are able to take measurements, record and store data, and send them to a central server (Singh et al., 2022).[5] The sensors are high precision, can be calibrated to defined tolerances, and will report their findings in real time.[6] These are all shared features with good security monitoring tools, such as those deployed in secure buildings, and indeed those already used to protect farms and crops against pests and other physical threats.[7] Therefore, there is an opportunity to build some strong foundations for the development of a secure network, though it is clear that this must be as automated and user friendly as possible, as much of the online literature in the smart agriculture sector references only the most basic security tips; such as recommending the use of passwords.[8] This is not to say that there are no complex IoT farming systems, quite the opposite, research published in China detail both the the improvements that precision feeding offers to the farm's output efficiency and cost while also tracking the performance of the IoT platform in terms of usability of the system and packet loss (Xia et al., 2023).[9]

By tracking the error rate of the precision equipment against the baseline error recorded when humans did not use technology, the researchers found that the IoT system performed at last as well as humans at delivering a quantity of feed to animals.[10] However, the system was severely limited, while there were a great deal of child nodes providing either monitoring data or controlling servo motors, the parent node - a handheld Personal Digital Assistant (PDA) - was a bottleneck for the data.[11] This suggests that while usability in the form of a handheld device was prioritised for the research in order to record accurate data, in the event that two child nodes attempted to send data to the PDA at the same time, packet loss became inevitable and so therefore the data could not be recorded.[12] This causes significant concern for a system that prioritises security of the system as information must have good integrity, and must be recorded accurately for fear of triggering a security alert or an alarm. This also highlights

[5] Singh, R., Singh, R., Gehlot, A., Akram, S.V., Priyadarshi, N. and Twala, B., 2022. Horticulture 4.0: Adoption of Industry 4.0 Technologies in Horticulture for Meeting Sustainable Farming. *Applied Sciences*, [online] 12(24), p.12557. https://doi.org/10.3390/app122412557.
[6] Ibid.
[7] Anon. 2025. *Smart Farming: How Technology is Enhancing Agricultural Security*. [online] Available at: <https://www.agritecture.com/blog/smart-farm-security-technology> [Accessed 11 June 2025].
[8] Ibid.
[9] Xia, J., Xu, J., Zeng, Z., Lv, E., Wang, F., He, X. and Li, Z., 2023. Development of a Precision Feeding System with Hierarchical Control for Gestation Units Using Stalls. *Applied Sciences*, [online] 13(21), p.12031. https://doi.org/10.3390/app132112031.
[10] Ibid. p.4.
[11] Ibid. p.5. Figure 10.
[12] Ibid. p.5.

the need for a number of human machine interfaces, to prevent the kind of system failure described above - it is clear from this source that a single PDA will not suffice for a complex operation. With a number of administration nodes and personal devices, comes the need for secure user authentication and role based policy access that come along with it.

As the number of nodes increase, so does the visible attack surface, increasing the risk of compromise from device hijacking and the risk of data loss as described in (Alruwaili et al., 2024).[13] It is therefore vital to properly secure the network that the devices operate within using standard methods, but also enforce rigorous security methods for the devices, the users, and the gateway that acts as interface between the user and the device infrastructure.[14] There are a number of different access control frameworks that have been proposed; including “physically secure”[15] “unclonable functions” (PUFs) that are hardcoded into the devices themselves,[16] and access control policies that are able to meet the demands of a non-static network of devices with multiple users and potentially different ways to view data travelling across the network.[17] The data travelling across the network may be traffic like simple sensor readings being updated in real time, or it could be commands issued by the users that control various robotic functions such as the delivery of animal food,[18] irrigation or camera position.[19] Therefore, I pose that the risk associated with device hijacking directly relates to the ability of the land to continue to produce efficiently. As a result, the system must offer security, observability, and also a level of alerting for when errors or anomalies are detected.

Security is an issue for IoT, which is often by definition exposed to the open internet, or running numerous devices and services on smaller local networks, which makes it

[13] Alruwaili, O., Mohammed Alotaibi, F., Tanveer, M., Chaoui, S. and Armghan, A., 2024. PSAF-IoT: Physically Secure Authentication Framework for the Internet of Things. *IEEE Access*, [online] 12, pp.78549–78561. https://doi.org/10.1109/ACCESS.2024.3407353. p.1.

[14] Ibid. p.4.

[15] Ibid. p.1

[16] Alruwaili, O., Tanveer, M., Alotaibi, F.M., Abdelfattah, W., Armghan, A. and Alserhani, F.M., 2024. Securing the IoT-enabled smart healthcare system: A PUF-based resource-efficient authentication mechanism. *Heliyon*, [online] 10(18), p.e37577. https://doi.org/10.1016/j.heliyon.2024.e37577. p.4.

[17] Chaudhry, S.A., Yahya, K., Al-Turjman, F. and Yang, M.-H., 2020. A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems. IEEE Access, [online] 8, pp.139244–139254. https://doi.org/10.1109/ACCESS.2020.3012121. p.2.

[18] Xia, J., Xu, J., Zeng, Z., Lv, E., Wang, F., He, X. and Li, Z., 2023. Development of a Precision Feeding System with Hierarchical Control for Gestation Units Using Stalls. *Applied Sciences*, [online] 13(21), p.12031. https://doi.org/10.3390/app132112031. p 5.

[19] Singh, R., Singh, R., Gehlot, A., Akram, S.V., Priyadarshi, N. and Twala, B., 2022. Horticulture 4.0: Adoption of Industry 4.0 Technologies in Horticulture for Meeting Sustainable Farming. *Applied Sciences*, [online] 12(24), p.12557. https://doi.org/10.3390/app122412557. p.2

necessary to consider the difficulty of updating device software or firmware.[20] This offers inherent weaknesses when relying on IoT devices, which may have persiant and unpatched vulnerabilities. (Chaudhry et al., 2020) notes that alongside this risk of vulnerability exploitation, the limited processing resource available to the microcontrollers commonly used in IoT devices expose these systems to catastrophic denial of service attacks.[21] These devices are often powered by internally small batteries, as IoT can support very low power consumption, however this limits the resilience that these devices offer, as described in (Deva Shahila et al., 2024).[22] The paper demonstrates why "effective safeguards" are needed when developing IoT products, and poses a "System on Chip"[23] approach to designing hardware for IoT networks that are at risk of tamper, or compromise. However, despite the clear need for such secure hardware and firmware in order to deploy hardened and resilient IoT, this is a novel approach that does not address the reality that many IoT devices are already on the market. Therefore reinstallation of a new system architecture onto a wide range of devices for the sake of security is simply not realistic - if only because it would be very inconvenient.

Therefore, policy based access control is the most reasonable solution for both user access and for device control. By ensuring that access is granted and continuously validated it will allow safe and secure transmission of data, and will also trigger alerts in the event of a policy based failure. This kind of security feature will prevent several prominent cyber attacks including physical capture, malicious device deployment and on path attacks.[24] We can rely on rule evaluation in order to set the triggers for such alerting or trigger more severe security features. Typically in an agricultural internet of things network we will see monitoring for temperature, moisture levels, or humidity within certain parameters. It is possible to adapt the same rules to communicate fluctuations in network traffic or to monitor for unauthorised access. (Chaudhry et al., 2020) proposes improvements to the existing "lightweight access control and key agreement (LACKA-IoT)" system that enforces an access control policy during the registration phase, and also provides continual security analysis to authenticate the

[20] García-García, J.I., Marín-Aragón, D., Maciá, H. and Jiménez-Cantizano, A., 2021. Viñamecum: A Computer-Aided Method for Diagnoses of Pests and Diseases in the Vineyard. Applied Sciences, [online] 11(10), p.4704. https://doi.org/10.3390/app11104704. P.2.

[21] Chaudhry, S.A., Yahya, K., Al-Turjman, F. and Yang, M.-H., 2020. A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems. *IEEE Access*, [online] 8, pp.139244–139254. https://doi.org/10.1109/ACCESS.2020.3012121. p.1.

[22] Deva Shahila, D.F., Suresh, Dr.L.P., Aruna Jeyanthy, P., Stephen, V. and R M, A., 2024. Designing and Analyzing Secure SoC Architecture for IoT Devices. In: *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*. [online] pp.1893–1899. https://doi.org/10.1109/ICCPCT61902.2024.10673314. p.5.

[23] Ibid. p.2

[24] Chaudhry, S.A., Yahya, K., Al-Turjman, F. and Yang, M.-H., 2020. A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems. IEEE Access, [online] 8, pp.139244–139254. https://doi.org/10.1109/ACCESS.2020.3012121. p.3.

legitimacy of device to device communication.[25] In the event a device or agent fails a control check, it would be possible to trigger an alert on the network, **or indeed to trigger an on-device security control that would isolate the device from the network.**

While rule based policy and automation are key features for secure IoT environments, these systems must be readable and accessible by people. Ideally these people would not have to be IT experts, or network administrators to operate the systems properly. Other IoT systems (without emphasis on security) employ visual interfaces in order to display data, and control functionality. In (Xia et al., 2023) there is an emphasis on how a graphical user interface helps a person to select an animal to be fed, by touching a screen and remotely trigger the delivery of food,[26] while in (Huet et al., 2022) the environmental controls of a beehive are displayed and updated in real time on a dashboard.[27] While the first example shows a snapshot of the current moment at any time, the second dashboard shows change over time in the form of a graph,m and includes an algorithmic projection of what the expected future reading would ideally be.[28] This kind of visual feedback is particularly useful, as while a system might be operating within accepted parameters, a person might be able to simply identify by sight anomalies in the readings too subtle for the system - or indeed recognise if the system's parameters are incorrectly configured or malfunctioning.

The literature here has sought to address general IoT security principles, authentication protocols, hardware security, and IoT's broad applications into agriculture. There is a generally accepted view in the literature that monitoring, control and prediction of data is a positive step forward and treated as if this is a step into the future, as "Agriculture 4.0". However this is often without reference to secure data handling, despite discussion of technology failures by many of the researchers. One of the key issues I wish to face is the trade off between security and convenience for smaller or less well off farmers who choose to adopt IoT technology. Instead of considering IoT devices as single function objects, as only environmental sensors or control panels, my project looks to integrate these devices into a network where each node actively participates in the security of the system.

[25] Ibid. p.5.

[26] Xia, J., Xu, J., Zeng, Z., Lv, E., Wang, F., He, X. and Li, Z., 2023. Development of a Precision Feeding System with Hierarchical Control for Gestation Units Using Stalls. *Applied Sciences*, [online] 13(21), p.12031. https://doi.org/10.3390/app132112031. p.5.

[27] Huet, J.-C., Bougueroua, L., Kriouile, Y., Wegrzyn-Wolska, K. and Ancourt, C., 2022. Digital Transformation of Beekeeping through the Use of a Decision Making Architecture. Applied Sciences, [online] 12(21), p.11179. https://doi.org/10.3390/app122111179. p.16.

[28] Ibid. Fig. 11.

Design Plan

Template: CM3070 Physical Computing and the Internet of Things - 8.1 Secure IoT Device Management in a Safety-Critical Smart Environment

Domain and design

The project sets out to design and develop a modular, policy based system for onboarding, controlling, and monitoring IoT devices into an agricultural setting. The primary goal is to deliver a system that is able to maintain operational security, as agriculture is fast becoming a safety critical environment in terms of the networked technology that is in use. This includes elements for device authentication and user registration while maintaining the expected usability of existing systems without compromising safety and security on the network. An important element will be the delivery of policy based access control, which is intended to allow a central polity engine to enforce specific rules, through real time analysis of the users and devices across the network. Real time decision making is really important for agricultural technology, as the right decision made too late can have devastating consequences for food production, for the safety of animals, and could even create risk to human life.

“Custos” addresses a gap in the agricultural space, where my research suggests that secure technology is the preserve of either academia, or expensive proprietary technology. Therefore for both convenience and also to serve the market, Custos makes use of readily available low cost hardware, while also exploring options for connectivity in agricultural or rural areas where a stable internet connection cannot be guaranteed for each node in the network. The project is aimed at non-technical users, who require an intuitive user interface that is easily to navigate in adverse weather conditions or periods of high stress or exhaustion. While the project does not expect farm workers to be non-technical, the design must be considered for the largest possible audience, across multiple generations. Therefore, the system must be able to give good visibility into the current status, and an operational overview of device status, network traffic, and sensor readings that enable the detection of errors or anomalies. Resilience is also crucial for the system design, as internet connected devices are under increasing pressure from unauthorised access and misuse. It is therefore really important that the system is resilient to misuse, cyber attack, and is able to operate in non optimal states including poor weather and network degradation.

Structure and architecture

The project uses Espressif ESP32 microcontrollers as sensor nodes, and Raspberry Pi computers to serve frontend and host backend infrastructure. The ESP32s will gather and transmit device telemetry and sensor readings to a central processing unit - the raspberry pi. The Raspberry Pi will then send the data to a backend server, which may be in the cloud, or could be served locally. Sensor nodes are connected to a MQTT (Message Queuing Telemetry Transport) broker, which is then connected to the backend. Transmitting data via MQTT protocols is fast and efficient, and while the protocols themselves do not have built in security, the MQTT packets do support passwords, and the published specification recommends using TLS across the network (it actually specifies SSL, but this term is depreciated and refers to the same underlying security feature). The back end infrastructure should ideally be cloud based, will receive and store data, and will analyse the data while enforcing the rules and policies that trigger alerts and automation. Finally, the dashboard will present a front end user interface - either a security monitoring tool such a Splunk, or an open source system such as Grafana. The front end will display a visual representation of the real time status of the system and also offer functionality to trigger system actions.

Components and technology

ESP32 microcontrollers serve as the primary perception layer of the system, the interface between the network and the physical environment that another on board system (such as the Pimoroni GROW, or the system that I built in a previous project) will monitor - note this system is not part of the design for this project. These low power devices are remarkably cheap and widely available, making them a good choice for a project where nodes may need replacement, and a modular and scalable system such as Custos.

The ESP32 boards support both WiFi and Bluetooth, however does not natively support LoRa (Long Range) broadcast frequencies, which would offer better performance over distance. Depending on the outcomes of the prototyping phase, it may be more suitable to switch to LoRa or LoRaWan in order to support long range communication. The ESP32 devices will be pre programmed to collect sensor data and also collect device health metrics, and push the reading to the MQTT broker using a defined syntax called a “topic”. To test the devices resilience to tampering, denial of service, or unusual readings, the devices can be installed with programs that simulate these issues and produce synthetic data.

Communication and network layer

MQTT is the messaging protocol of choice, as it is already widely used in IoT products, is implemented over WiFi, and is supported by the native hardware of the devices we are using. We might alternatively consider CoAP (Constrained Application Protocol), though this is considerably newer (published in 2014 as opposed to MQTT in 1999) and less widely adopted, therefore may pose a problem to the modularity of the system. Ideally, we would create a broker that would be able to handle multiple protocols, however this is not in scope for this project. Using the MQTT protocol for device to device and device to broken communications, we will define a “topic” structure - defining the expected data structure for communication that will be published by the nodes and received by the broker. The project will structure data for transmission in a hierarchy, stating the network name, device type, unique device ID, and the telemetry data.

The communication is managed between nodes using a publish and subscribe (pub/sub) model where devices publish messages to defined topics, and other devices or services listen for changes to the topics they are subscribed to. Authentication and validation is achieved using TLS across the network for encrypted communication between nodes, and credential authorisation will be enforced for all clients, devices and backend services. The aim is to ensure that only the authorised devices and users can publish and subscribe to data across the network. The MQTT protocol is significantly more resource efficient than communicating using HTTP, a key consideration when choosing the messaging protocol. This will also allow us to consider upgrading the data link and network later later, potentially to move away from WiFi to a lower power network, such as LoRa mentioned earlier. This would help conserve battery life in devices, as well as extending the range of the network, though it may impact the amount of data that would be able to travel across the network with devices transmitting simultaneously.

Add diagram

Backend and policy

The policies are set and enforced by a central service on the backend, handling data ingest and analysis. The backend also determines policy based access control, and provides commands in the event of rules being broken. Policies are defined in rule files which allow for flexible and adaptable configurations, to handle data sent by different nodes. When handling rule evaluation, the backend will have triggers set for alerting the user via the frontend, or cutting the data connection with a device, that will be contained within separate files or hardcoded into the system.

Examples for rules might include a user alert if the temperature on certain sensor reaches above a certain threshold. This would allow for monitoring of the Custos system in the event of extreme temperature fluctuation. Additionally in the event that no data is received from a publishing node for a certain period of time, the Node device status would be updated in the dashboard as potentially degraded. In the event that extremely anomalous data is published to a topic, we may consider a rule that disconnects the publishing node entirely to prevent data integrity compromise.

Onboarding and offboarding

The process of onboarding a new device, and disconnecting a device safely from the network…
Add diagrams!

Dashboard

It is important that the system includes a user-friendly dashboard that includes visualisations of the data within the network. The focus will be on simplicity and usability for non-technical users, and will prominently display key metrics clear alerting and the device health across the network. Metrics may include; last connection time, battery health, and a graph that shows change over time for a user defined metric. Users may be able to select the period of time that the graph shows data. This would provide the visual feedback to help users understand the data they are provided, and also help with anomaly detection missed by the system, or not defined by the rules. The dashboard would also be an interface to onboard new devices into the network and revoke device access.

Add diagram

Control

The system includes the ability to send control commands to devices. This is essential for security, as it is important to be able to isolate compromise node from the network. In the context of Agriculture a malfunctioning IoT device might cause damage or risk to human life. This will be possible to complete from the dashboard.

Workplan

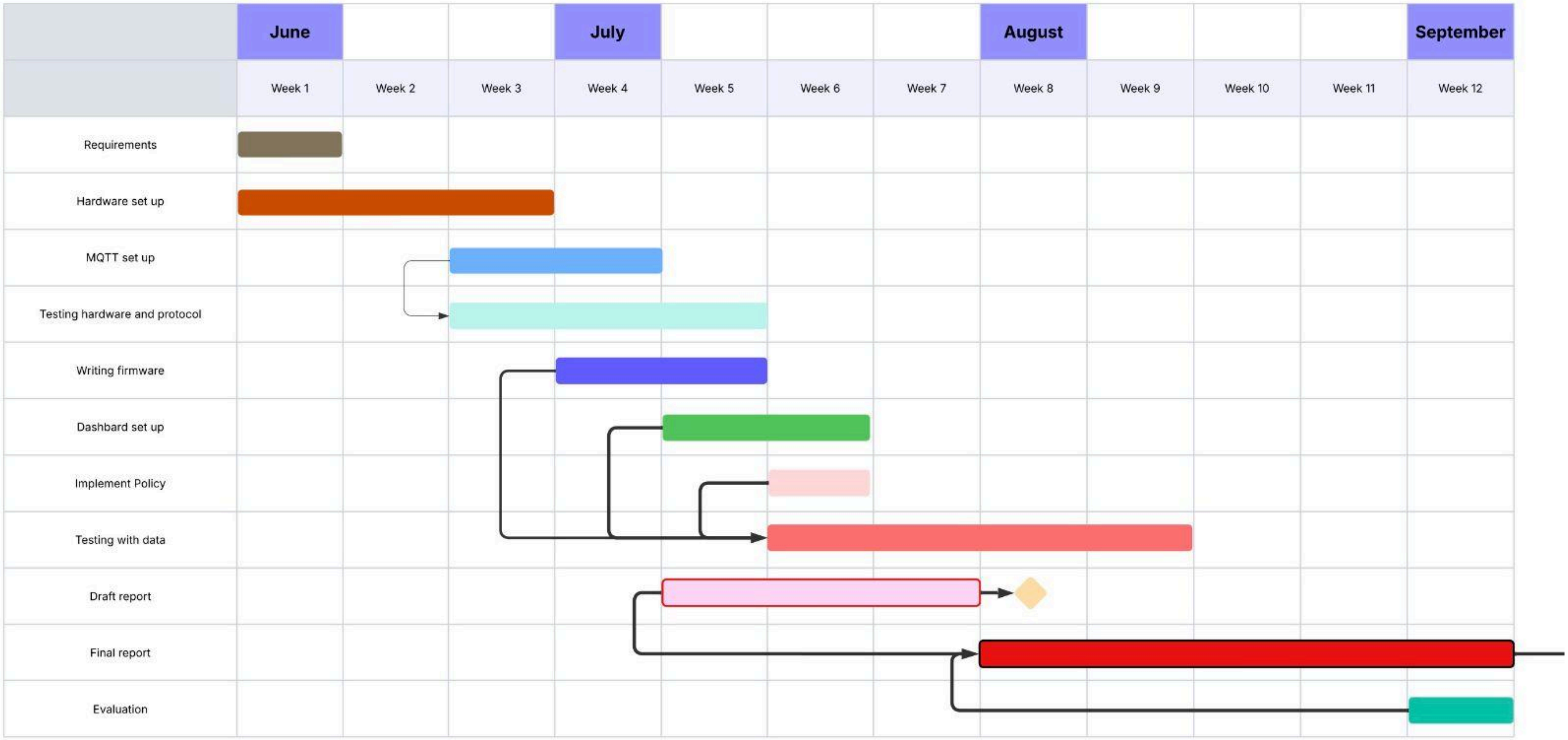
Week starting:	
June 16 Week 1	Finalising the requirements for the system Flashing and testing microcontrollers Connecting sensors
June 23 Week 2	Testing batteries, power sources Writing drafts for the policy logic
June 30 Week 3	Setting up MQTT + broker Testing on microcontrollers
July 7 Week 4	Write failing tests Writing firmware for microcontrollers to enable data collection and transmission
July 14 Week 5	Setting up dashboard and frontend Testing data Write first draft of report
July 21 Week 6	Implementing policy Testing logic Second draft of report
July 28 Week 7	Write failing tests Test system for errors Proofread report, check references
August 4 Week 8	Draft report due Create fault injections with synthetic data
August 11 Week 9	Test system at fault

August 18 Week 10	Write first draft of final report
August 25 Week 11	Finalise codebase
September 1 Week 12	Make and document final evaluation Do exam
September 8 Week 13	Second draft of report Proofread
September 15 Week 14	Submission Deadline

Gantt Chart:

(see next page)

Custos - CM3040 Project 8.1
Secure IoT Device Management in a Safety-critical Smart Environment



Feature Prototype:

This is very much the minimum viable product, even for a prototype. We focus on setting up the lightweight messaging protocol, and some basic alerts and tracking features for the dashboard. The dashboard can ‘onboard’ new devices by subscribing to their messages - however there is no validation at this point that the data is valid, instead relying on the trustworthiness of the cloud broker. The prototype aims to demonstrate that support for real-time telemetry which is updated and tracked in a graph from iot devices is possible using a lightweight messaging protocol. It provides a user friendly dashboard with visualisations, monitoring of connected nodes and alerting to indicate the user where there may be an error. The system currently relies on a simulation script written in c++ which provides synthetic data that mimics a sensor node, in the final version of this project this will not rely on synthetic data, it will be running actual nodes. In deviation from the previous plan we are using esp8266 microcontrollers as hardware; one which will publish the synthetic data, and another that is subscribed to receive the data. Both microcontrollers have visual alerting in the form of LEDs to alert the user to the system status. The project publishes to a cloud-based broker,. and serves a locally hosted web dashboard. However, I expect the final iteration of the project to run a local broker that publishes to a dashboard service, such as Splunk or Grafina, as opposed to relying on a dashboard that I coded myself.

Prototype Architecture

The prototype system comprises of

- Publishing Node - Sensor Simulation
- Subscriber Node - Physical Data Check
- Web Dashboard - Visual Interface

The nodes and the dashboard are connected to the cloud MQTT broker - in this case hosted with HiveMQ, which relies on WebSockets to provide authentication and security.

The first microcontroller, the publishing node, is implemented with a C++ script for arduino called “publisherTest.ino”. The second, similarly, runs “subscriberTest.ino”.

The publishing node:
The publishing node features not only the publication of the synthetic data which is essentially a random float to a certain topic, but also features two LED alert signals. The

first alert will flash red in the event that the hardware has not connected to the local Wi-Fi, and will stay red continuously in the event that it is not possible to connect to the cloud broker. The second LED will light continuously blue so long as there is a connection to the broker and so long as data is being published. The publisher connects to the MQTT broker using TLS and publish only credentials. Messages are sent in JSON structure, with a key value pair, along with a timestamp. Messages are published at a regular interval, just like real data. The synthetic data was the first real test of the MQTT messaging system, and has been very useful for debugging as I didn’t need to worry about a sensor or actuator failing.

The subscriber node:
This is considerably more simple - it's simply listens for data that is published on the topic the node is subscribed to, and will blink a yellow LED for each packet of data that it receives. This is subscribed to the topic published by the other node, and is connected to the broker, using subscribe only credentials. The blinking LED acts as a feedback loop, showing that the data is being published, being received by the broker, and being recovered by the subscriber. It indicates the network is functional, and gives a visual alert when there is an issue.

The web dashboard:
A browser based front and system, written in HTML JavaScript and little else. it is intentionally simple and connects to the broker using secure web sockets and in its default configuration, is subscribed to the same topics as the subscriber node. It is possible to subscribe to further topics on the dashboard. The dashboard includes a graph that visualises the data (received by the broker) from the publishing node, recorded as temperature over time. There is a visual alert that triggers when the difference between consecutive data points is greater than a user defined number. The browser will also alert the user in the event that data is no longer being received, and will record the length of the data outage in a chart. To give the user an indication of the amount of information loving through the network I've included a graph logging the number of messages published per second. I've also included a CSV export to allow the user to download all telemetry captured by the dashboard. There is a limitation here as the dashboard currently does not record any data and when the dashboard is closed all data from the session is lost .

Evaluation

The prototype has been evaluated to meet some of the most basic requirements of the system, and to assist with the debugging process. I think this succeeds on the basis that it demonstrates the technical feasibility, and responsiveness of the proposed system, as well as offering options for observability, and monitoring over time.

- The synthetic sensor was published to the broker, and received by both subscriber and dashboard.
- The subscriber LED behaviour is consistent with expected flow, demonstrating fault detection that is *not* on the dashboard. This offered validation and resilience.
- The dashboard visualises data in real time, and alerts trigger when thresholds are breached or data is missing, proving the alert logic functions as designed.
- MQTT message delays and outages were simulated by literally pulling the plug on the publishing node. These interruptions were detected by both the dashboard and the subscriber node.
- The dashboard works, and functions on Chrome for desktop and mobile.
- The CSV export functionality works, but can be improved by pulling from a database.
- User interface elements such as threshold controls and alert text trigger as expected, and look good too!
- The prototype demonstrates real-time data ingestion, alerting on both outage and data anomalies, and includes physical hardware feedback.
- The system is modular, extensible to new sensor types and topics.

This section sets out how we will evaluate the development and prototyping of the system. The methodology incorporates evaluation criteria for “state-of-the-art” IoT device management systems, with a focus on security and usability. This has been developed using iterative development with periodic assessment and (some) personal reflections that we have captured to drive improvement.

1. Objectives

The evaluation phase is designed to answer the following questions:

- Does the system provision, authenticate, monitor, and manage devices?

- Are the communications reliable?
- Can the system detect anomalous behaviour?
- How resilient, and user-friendly is the prototype?
- Can the dashboard provide data in near real-time?

2. Criteria

We will assess the function, security, performance, ease of use, and extendability.

2.1 Functions

- Sensor readings must be captured and transmitted.
- MQTT message exchange happens, and has correct topics.
- The backend must correctly parse, store, and analyse data.
- Dashboard must display accurate, timely information.
- Provisioning workflow must authenticate new devices and issue keys.

2.2 Security

- Devices must not be able to join without undergoing a provisioning process.
- Communication must be encrypted.
- Credentials must be stored securely and never transmitted in plain text.
- Some resilience to common attacks (e.g. replay, spoofing, DoS).

2.3 Performance

- MQTT throughput under various device loads.
- Backend processing time for analysing data.
- Time taken for a new device to come online.
- System responsiveness (data generation and dashboard interface).

2.4 Usability

- Ease of onboarding (noting the time, any errors, any human intervention).
- Clarity of dashboard interface, any controls (disconnecting a device).
- Properly displayed logs and alerts.

2.5 Extendability

- Ease of adding additional sensors and new rules.
- Number of devices supported before performance drops off.
- Modularity of system architecture (for cloud / local deployment).

3. Evaluation

3.1 Proof of Concept

An end-to-end system will be deployed, including:

- At least two nodes transmitting temperature or moisture, and uptime.
- A Raspberry Pi configured with:
    - Local MQTT broker (e.g., Mosquitto),
    - Backend server (Python/Node.js service),
    - (Optional cloud MQTT broker + cloud backend test.)
    - Backend database (e.g., SQLite for prototype, PostgreSQL for full deployment).
- Dashboard connected to backend and broker (Grafana).

It will also have been exposed to basic stress testing using synthetic data using test scripts that simulate sensor behaviour, also testing using malformed or unexpected data, and some .

3.2 Security Testing

- Onboarding tested with invalid credentials (and replay attack ?)
- Attempt to connect unauthenticated device to broker.
- TLS and password-based protection on MQTT will be checked using a tool like Wireshark (is this realistic for prototype?).
- Endpoints tested for request validation and access control.
- Device impersonation and message tampering will be tried.

3.3 Performance

- A script will simulate 10+ concurrent nodes to test broker and backend.
- Network latency will be logged and compared against target thresholds (< 1s MQTT roundtrip, < 3s backend-to-dashboard latency).
- Backend data speeds will be measured in events per second (EPS).

3.4 User Testing & Feedback

- Usability testing with testers (two family members) to perform device onboarding, dashboard interpretation with data and also an alert.
- Feedback asked for:
    - clarity of instructions,
    - preference for the interface,
    - feeling of security.

3.5 Evaluation Reporting

All results will be documented with:

- Summary tables for metrics (latency, provisioning time, etc.).
- Diagrams showing system architecture and data flow.
- Screenshots of the dashboard during normal and alert states.
- Commentary on lessons idetifed and observed limitations.

4. Tools and Techniques

- Wireshark to analyse MQTT traffic and ensure TLS is enforced.
- Grafana for data visualisation and dashboarding.
- Python for backend logging and analysis.
- Curl for manual testing of backend endpoints.
- Bluetooth debugging tools – to simulate provisioning /onboarding attempts.
- Custom test scripts (also Python) to generate synthetic dat traffic and evaluate scaling.

5. Evaluation Schedule (from 04/08 until submission)

Week	Task
1	04/08 Unit testing of backend code and MQTT handling
2	11/08 Initial provisioning and functional tests
3	18/08 Security validation and TLS checks
4	25/08 Stress and performance testing
5	31/08 User testing and final evaluations
6	07/09 Documentation and summary of evaluation findings

6. Success Metrics for prototype

Metrics

- MQTT message latency less than 2 seconds
- Device onboarding success rate close to 90% (9 times out of 10)
- Dashboard update latency less than 5 seconds
- 10 simulated devices supported
- Unauthenticated provisioning attempts blocked

7. Reflection and Future Work

Evaluation will also consider how well the system met its original objectives and where future improvements could be made. This could include:

- Integration of more complex device policies (e.g. anomaly detection).
- Deployment to an actual network.

- Real-world user study with a greater number of testers (that aren't all family).
- Development of mobile app companion for provisioning.

Implementation

The next iteration of Custos will begin to incorporate a dedicated backend, including a database, and ideally a hardware MQTT broker. I will expand the number of topics provided by synthetic data and use this to develop and test a cloud hosted dashboard, before flashing real sensor firmware and deploying in my garden. I need to address power management and hardware protection for physical units and iterate the security features in the code itself.

The project is coming along nicely, and I expect to be able to evaluate the final project deliverable on the performance of the features I have implemented for this prototype, but in a live environment, with the ability to store and perhaps even query past data. The prototype offers a secure and reasonably functional smart device monitoring system, offering user feedback both using an online dashboard and also in the physical world. I am excited to take the next steps, and see if I am able to deploy a more complex communication system.

This section outlines the current stage of implementation, details the techniques used, discusses key challenges addressed, and presents the path forward.

System Overview

Custos is built around a series of interconnected modules:

- **Sensor Nodes**: ESP32 microcontrollers
- **Communication Layer**: MQTT broker (currently hosted on Raspberry Pi)
- **Backend**: Custom Python-based service with support for data processing and security enforcement
- **Database**: Planned integration with SQLite for local storage; cloud migration planets
- **Dashboard**: Prototype dashboard with Grafana and optional web UI for real-time visibility and control

Each of these components is designed to function independently and interoperate using a standardized message format.

Microcontroller and Synthetic Data

The ESP32 firmware currently used is a hybrid implementation: in lieu of real sensors, it generates **synthetic data** streams to simulate environmental conditions and device states. This synthetic data follows a defined topic schema for MQTT publishing:

```
custos/device/<device_id>/telemetry
custos/device/<device_id>/status
custos/device/<device_id>/alert
```

Each synthetic message includes:

- A device UUID
- Timestamp (UTC ISO format)
- Sensor data (temperature, humidity, voltage, etc.)
- Device status (online/offline)
- Alert flags (e.g., overvoltage, tampering)

This synthetic setup allows for rapid iteration of the backend and dashboard components without needing to fully deploy hardware units during development.

MQTT Broker and Messaging

The MQTT broker is currently hosted on a local Raspberry Pi using **Mosquitto**, configured to use username-password authentication. While this implementation does not yet enforce TLS encryption, support is in place to upgrade the stack to use certificates during deployment.

ESP32 devices connect using lightweight MQTT libraries (e.g. PubSubClient), and publish synthetic telemetry every 10 seconds. The broker uses topic filters and message retention policies to facilitate both real-time updates and state restoration upon device reconnection.

Planned upgrades include:

- **Hardware-based broker appliance** with secure key provisioning
- **QoS 1** support for at-least-once delivery
- **Broker-side logging and auditing**

Backend Service

A custom Python-based backend has been developed to:

- **Subscribe to MQTT topics**
- **Parse and validate incoming messages**
- **Log structured JSON payloads**
- **Perform real-time rule evaluation**
- **Trigger alerts or actions (e.g., LED blink, dashboard push)**

The backend uses paho-mqtt for broker communication and is designed as a service daemon using asyncio and Python's built-in queue and threading libraries to manage concurrency. Future versions will migrate to FastAPI and use Redis for message caching.

Planned feature expansions:

- Modular plugin system for different rule sets
- Real-time alerting engine with email/SMS integration
- Historical data query endpoints

Security Enforcement

A lightweight security framework is built into the firmware and backend logic:

- **Device authentication**: each device is issued a unique ID and pre-shared key

- **Message signing**: each message is SHA256 hashed with HMAC using the pre-shared key
- **Backend validation**: backend checks message origin and integrity before processing

Although the current prototype operates on test credentials and unencrypted channels, the next phase will include:

- BLE-based provisioning to inject keys
- TLS enforcement over MQTT and REST endpoints
- Policy-based access control for dashboard interactions

Dashboard and Visual Feedback

A Grafana-based dashboard connects to a local time-series database (currently simulated via CSV files) and displays:

- Real-time telemetry graphs (e.g., temperature over time)
- Device status indicators
- Triggered alerts
- Uptime logs

Example panels include:

- A multi-line chart showing data from multiple ESP32s
- A table of last known device states and alert levels
- A heat map of network activity over 24 hours

The dashboard is customizable and future work will migrate to a hosted instance of Grafana or a custom frontend using React and D3.js for full control.

In addition to the digital interface, ESP32 devices also provide **physical feedback** using onboard LEDs. For example:

- **Yellow blink**: message sent
- **Green solid**: connected and idle
- **Red flash**: alert raised

This provides at-a-glance confirmation of device state, especially useful in outdoor deployments.

Deployment and Hardware Design

The current physical deployment is limited to the lab, but the following steps are planned for full rollout:

1. **Deploy MQTT broker to cloud VPS**
2. **Flash firmware to physical ESP32s with real sensors (e.g., DHT22, soil moisture, vibration sensors)**
3. **Design and 3D print weatherproof casings**
4. **Integrate solar panel charging system and battery voltage monitoring**
5. **Test BLE provisioning routine in the field**

Attention will be paid to **power management**, using deepSleep modes in ESP32 and **watchdog timers** to handle connection drops.

Visual Results and Sample Outputs

Screenshots and sample outputs will be included in the full report, but current highlights include:

- Telemetry chart showing simulated temperature spikes
- Status overview table of 5 virtual devices

- MQTT message log from Mosquitto
- CLI logs from the backend daemon parsing and validating incoming messages

These outputs validate the overall architecture and demonstrate a working proof-of-concept.

Bibliography

Alruwaili, O., Mohammed Alotaibi, F., Tanveer, M., Chaoui, S. and Armghan, A., 2024a. PSAF-IoT: Physically Secure Authentication Framework for the Internet of Things. *IEEE Access*, [online] 12, pp.78549–78561. https://doi.org/10.1109/ACCESS.2024.3407353.

Alruwaili, O., Tanveer, M., Alotaibi, F.M., Abdelfattah, W., Armghan, A. and Alserhani, F.M., 2024b. Securing the IoT-enabled smart healthcare system: A PUF-based resource-efficient authentication mechanism. *Heliyon*, [online] 10(18), p.e37577. https://doi.org/10.1016/j.heliyon.2024.e37577.

Anon. 2022a. *Ukraine war spotlights agriculture sector's vulnerability to cyber attack*. [online] Cisco Talos Blog. Available at: <https://blog.talosintelligence.com/ukraine-and-fragility-of-agriculture/> [Accessed 10 June 2025].

Anon. 2022b. *Why Ukraine's agriculture sector could be set to attract cyberattacks*. [online] euronews. Available at: <https://www.euronews.com/next/2022/08/24/the-war-in-ukraine-has-threatened-its-vital-agriculture-now-it-could-be-crippled-by-a-cybe> [Accessed 10 June 2025].

Anon. 2025a. *BS 8646:2023 | 30 Jun 2023 | BSI Knowledge*. [online] Available at: <https://knowledge.bsigroup.com/products/use-of-autonomous-mobile-machinery-in-agriculture-and-horticulture-code-of-practice> [Accessed 10 June 2025].

Anon. 2025b. *CEMA - European Agricultural Machinery - Agriculture 4.0*. [online] Available at: <https://www.cema-agri.org/priorities/agriculture-4-0> [Accessed 11 June 2025].

Anon. 2025c. *Guidance to support safe and secure use of crop robots in farming*. [online] BSI. Available at: <https://www.bsigroup.com/en-GB/insights-and-media/media-centre/press-releases/2023/june/guidance-safe-secure-use-crop-robots-farming/> [Accessed 10 June 2025].

Anon. 2025d. *Smart Farming: How Technology is Enhancing Agricultural Security*. [online] Available at: <https://www.agritecture.com/blog/smart-farm-security-technology> [Accessed 11 June 2025].

Anon. 2025e. *The Complete Guide to Smart Farming & Agriculture*. [online] Available at: <https://smartertechnologies.com/guides/the-complete-guide-to-smart-agriculture-farming/> [Accessed 10 June 2025].

Anon. n.d. *12 Climate-smart technologies that could transform the way we grow food*. [online] World Economic Forum. Available at: <https://www.weforum.org/stories/2023/10/12-climate-smart-technologies-that-could-transform-the-way-we-grow-food/>.

Anon. n.d. *How the Russian invasion of Ukraine has further aggravated the global food crisis*. [online] consilium.europa.eu. Available at: <https://www.consilium.europa.eu/en/infographics/how-the-russian-invasion-of-ukraine-has-further-aggravated-the-global-food-crisis/>.

Whiting, A., 2019. The UK senses a smart farming revolution. *Transform Industry*. Available at: <https://transformindustry.com/sensors/the-uk-senses-a-smart-farming-revolution/> [Accessed 10 June 2025].

Chaudhry, S.A., Yahya, K., Al-Turjman, F. and Yang, M.-H., 2020. A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems. *IEEE Access*, [online] 8, pp.139244–139254. https://doi.org/10.1109/ACCESS.2020.3012121.

Cupriak, A., 2024. Labour manifesto: ‘Food security is national security’. Farmers Guide. Available at:

<https://www.farmersguide.co.uk/business/labour-manifesto-food-security-is-national-security/> [Accessed 15 June 2025].

Anon., 2025. *Shielding the supply: Cybersecurity in food and agriculture*. [online] Cybersecurity Guide. Available at: <https://cybersecurityguide.org/industries/food-and-agriculture/> [Accessed 10 June 2025].

De Clercq, M., Vats, A. and Biel, A., 2018. Agriculture 4.0: The Future of Farming Technology. [online] Available at: <https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2021/apr/agriculture-4-0-the-future-of-farming-technology.pdf>.

Deva Shahila, D.F., Suresh, Dr.L.P., Aruna Jeyanthy, P., Stephen, V. and R M, A., 2024. Designing and Analyzing Secure SoC Architecture for IoT Devices. In: *2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*. [online] pp.1893–1899. https://doi.org/10.1109/ICCPCT61902.2024.10673314.

García-García, J.I., Marín-Aragón, D., Maciá, H. and Jiménez-Cantizano, A., 2021. Viñamecum: A Computer-Aided Method for Diagnoses of Pests and Diseases in the Vineyard. *Applied Sciences*, [online] 11(10), p.4704. https://doi.org/10.3390/app11104704.

Gerlée, K., 2023. IoT in agriculture – 6 smart farming examples. *Freeeway*. Available at: <https://freeeway.com/iot-in-agriculture-6-smart-farming-examples/> [Accessed 10 June 2025].

Holmberg, A., 2024. Food security in light of the war in Ukraine: food studies meets defence studies. *Defence Studies*, [online] 24(4), pp.543–558. https://doi.org/10.1080/14702436.2024.2378793.

Huet, J.-C., Bougueroua, L., Kriouile, Y., Wegrzyn-Wolska, K. and Ancourt, C., 2022. Digital Transformation of Beekeeping through the Use of a Decision Making Architecture. *Applied Sciences*, [online] 12(21), p.11179. https://doi.org/10.3390/app122111179.

Adamson, J., 2023. TuberScan undergoes UK testing. *Crop Production Magazine*. Available at: <https://www.cpm-magazine.co.uk/news/tuberscan-undergoes-uk-testing/> [Accessed 10 June 2025].

Joseph, M., Matthew, B., Lee, W., Jim, M., Richard, G. and Edwin, H., 2025. *Research - Tuberscan: Using deep learning and data science to improve agronomy decisions in the potato industry*. [online] Available at: <https://www.harper-adams.ac.uk/research/project/1285/tuberscan-using-deep-learning-and-data-science-to-improve-agronomy-decisions-in-the-potato-industry> [Accessed 10 June 2025].

Leal Filho, W., Fedoruk, M., Paulino Pires Eustachio, J.H., Barbir, J., Lisovska, T., Lingos, A. and Baars, C., 2023. How the War in Ukraine Affects Food Security. *Foods*, [online] 12(21), p.3996. https://doi.org/10.3390/foods12213996.

Lewis, V., 2023. *BSI release guidance on use of AMM in agriculture and horticulture*. [online] News from AA Farmer. Available at: <https://s45854.p1721.sites.pressdns.com/news/bsi-release-guidance-on-use-of-amm-in-agriculture-and-horticulture.html> [Accessed 10 June 2025].

Lin, F., Li, X., Jia, N., Feng, F., Huang, H., Huang, J., Fan, S., Ciais, P. and Song, X.-P., 2023. The impact of Russia-Ukraine conflict on global food security. *Global Food Security*, [online] 36, p.100661. https://doi.org/10.1016/j.gfs.2022.100661.

Plexal, n.d. *CYBER SECURITY AND OUR FOOD SYSTEM*. [online] Available at: <https://www.plexal.com/wp-content/uploads/2022/05/Cyber-security-and-our-food-system-4.pdf>.

Shahila, D.F.D., Prathaban, B.P., Stephen, V., Jeyanthy, P.A. and Evangelin, D.L., 2023. Novel Biometric ATM User Authentication and Real-Time Secure Clickbait for Multi-Bank Transactions. In: *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*. [online] pp.1–7. https://doi.org/10.1109/ICCEBS58601.2023.10448889.

Shalimov, A., 2023. *IoT in Agriculture: 9 Technology Use Cases for Smart Farming (and Challenges to Consider)*. [online] Eastern Peak - Technology Consulting & Development Company. Available at:

<https://easternpeak.com/blog/iot-in-agriculture-technology-use-cases-for-smart-farming-and-challenges-to-consider/> [Accessed 10 June 2025].

Sharp, N., 2025. *7 types of agricultural sensors driving the smart farming revolution*. [online] Available at: <https://www.escatec.com/blog/7-types-of-agricultural-sensors-driving-the-smart-farming-revolution> [Accessed 10 June 2025].

Singh, R., Singh, R., Gehlot, A., Akram, S.V., Priyadarshi, N. and Twala, B., 2022a. Horticulture 4.0: Adoption of Industry 4.0 Technologies in Horticulture for Meeting Sustainable Farming. *Applied Sciences*, [online] 12(24), p.12557. https://doi.org/10.3390/app122412557.

Singh, R., Singh, R., Gehlot, A., Akram, S.V., Priyadarshi, N. and Twala, B., 2022b. Horticulture 4.0: Adoption of Industry 4.0 Technologies in Horticulture for Meeting Sustainable Farming. *Applied Sciences*, [online] 12(24), p.12557. https://doi.org/10.3390/app122412557.

Torres-Sanchez, R., Zafra, M.T.M., Soto-Valles, F., Jiménez-Buendía, M., Toledo-Moreo, A. and Artés-Hernández, F., 2021. Design of a Distributed Wireless Sensor Platform for Monitoring and Real-Time Communication of the Environmental Variables during the Supply Chain of Perishable Commodities. *Applied Sciences*, [online] 11(13), p.6183. https://doi.org/10.3390/app11136183.

Welsh, C., 2024. Russia, Ukraine, and Global Food Security: A Two-Year Assessment. [online] Available at: <https://www.csis.org/analysis/russia-ukraine-and-global-food-security-two-year-assessment> [Accessed 10 June 2025].

Xia, J., Xu, J., Zeng, Z., Lv, E., Wang, F., He, X. and Li, Z., 2023. Development of a Precision Feeding System with Hierarchical Control for Gestation Units Using Stalls. *Applied Sciences*, [online] 13(21), p.12031. https://doi.org/10.3390/app132112031.

Changelog
Cut parts of the lit review